



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

**Αξιολόγηση ποιότητας εφαρμογών σε υπηρεσιοστρεφείς
υποδομές πραγματικού χρόνου**

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Φώτης Α. Αίσωπος

Αθήνα, Ιούλιος 2012



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Αξιολόγηση ποιότητας εφαρμογών σε υπηρεσιοστρεφείς υποδομές πραγματικού χρόνου

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Φώτης Α. Αίσωπος

Συμβουλευτική Επιτροπή : Θεοδώρα Α. Βαρβαρίγου
Γεώργιος Στασινόπουλος
Ανδρέας Γ. Σταφυλοπάτης

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την

.....
Θ.Α. Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

.....
Γ. Στασινόπουλος
Καθηγητής Ε.Μ.Π.

.....
Α. Γ. Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

.....
Σ. Παπαβασιλείου
Αν. Καθηγητής Ε.Μ.Π.

.....
Δ. Ασκούνης
Καθηγητής Ε.Μ.Π.

.....
Ι. Ψαρράς
Ομ. Καθηγητής Ε.Μ.Π.

.....
Ε. Βαρβαρίγος
Καθηγητής Πανεπιστημίου
Πατρών

Αθήνα, Ιούλιος 2012

.....
Φώτης Α. Αίσωπος

Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Φώτης Α. Αίσωπος, 2012.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

ΠΡΟΛΟΓΟΣ

Η διδακτορική διατριβή που παρουσιάζεται εκπονήθηκε από το Δεκέμβριο του 2007 μέχρι και τον Ιούνιο του 2012, στο εργαστήριο Τηλεπικοινωνιών του τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής, στη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Η διατριβή αυτή αποτέλεσε αφορμή για να ασχοληθώ με εξαιρετικά ενδιαφέροντα επιστημονικά θέματα, που αφορούν κυρίως στους τομείς της προδιαγραφής, του σχεδιασμού, της μοντελοποίησης και του ελέγχου υπηρεσιοστρεφών υποδομών και με εφοδίασε με πολύτιμη εμπειρία και γνώσεις.

Θα ήθελα να ευχαριστήσω θερμά την καθηγήτριά μου κ. Θεοδώρα Βαρβαρίγου για το ενδιαφέρον που έδειξε, για τις πολύτιμες συμβουλές της και την ιδιαίτερη στήριξη που μου παρείχε κατά την διάρκεια αυτής της πορείας μου, καθώς επίσης και τους καθηγητές της τριμελούς συμβουλευτικής επιτροπής κ.κ. Γεώργιο Στασινόπουλο και Ανδρέα Γ. Σταφυλοπάτη. Επίσης, θα ήθελα να ευχαριστήσω όλους τους συναδέλφους μου στην ερευνητική ομάδα με τους οποίους συνεργάστηκα άπογα όλα αυτά τα χρόνια. Ιδιαίτερες ευχαριστίες ωστόσο θα ήθελα να απευθύνω στους στενούς μου συνεργάτες Κωνσταντίνο Τσερπέ, Μαγδαληνή Καρδαρά, Θάνο Παπαοικονόμου, Γεώργιο Παπαδάκη, Βρεττό Μουλό, Βίκυ Ανδρόνικου και Μανώλη Σάρδη με τους οποίους μοιραστήκαμε πολλές ώρες ερευνητικής εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω μέσα από την καρδιά μου την οικογένειά μου και την Ανθή μου για την πίστη τους σε εμένα και τη στήριξή τους σε όλες τις δύσκολες στιγμές.

Φώτης Α. Αίσωπος

Ιούνιος 2012

Πίνακας περιεχομένων

Περίληψη	xi
Abstract	xv
1 Εισαγωγή	1
1.1 Υπηρεσίες στο διαδίκτυο	1
1.2 Υπηρεσιοστρεφής Αρχιτεκτονική	3
1.2.1 Γενικά για τις υπηρεσίες σε SOA	5
1.2.2 Διαδικτυακές Υπηρεσίες	7
1.3 Σχήματα παροχής υπηρεσιών	13
1.3.1 Η Εφαρμογή ως Υπηρεσία	14
1.3.2 Η Πλατφόρμα ως Υπηρεσία	16
1.3.3 Η Υποδομή ως Υπηρεσία	17
1.4 Καινοτομία - Συνεισφορά	18
1.5 Οργάνωση κειμένου	20
2 Πλατφόρμες Software as a Service	22
2.1 Υπολογιστικό πλέγμα	23
2.1.1 Ορισμός Πλέγματος	23
2.1.2 Χαρακτηριστικά	25
2.1.3 Ιστορικά στοιχεία	26
2.1.4 Παράμετροι Σχεδίασης Πλέγματος	32
2.1.5 Αρχιτεκτονική Ανοιχτών Υπηρεσιών Πλέγματος	37
2.1.6 Πλατφόρμες Ανοικτού Κώδικα	45
2.2 Το Υπολογιστικό Νέφος	47
2.2.1 Από το υπολογιστικό πλέγμα στο υπολογιστικό νέφος	47
2.2.2 Ορισμός Νέφους	48
2.2.3 Χαρακτηριστικά	49
2.2.4 Μοντέλα Ανάπτυξης και Παροχής	51
2.2.5 Το πλαίσιο SPI	52
2.2.6 Πλατφόρμες Ανοικτού Κώδικα	53
2.3 Πλατφόρμες με ανοιχτές αγορές εφαρμογών ως υπηρεσίες	55
2.3.1 Positive Spaces	55
2.3.2 SocIoS	58
3 Ποιότητα Υπηρεσίας	62
3.1 Η έννοια της Ποιότητας Υπηρεσίας	63
3.2 Service Level Agreement	64
3.3 Μοντέλο της Ποιότητας Υπηρεσίας	68
3.4 Ποιότητα Υπηρεσίας σε περιβάλλοντα SaaS	70
3.4.1 WS-Agreement	70
3.4.2 Εφαρμογές SaaS σε αγορές υπηρεσιών και παράμετροι ποιότητας	75
4 Αξιολόγηση Ποιότητας Υπηρεσίας σε SaaS	88
4.1 Αξιολόγηση όρων χαμηλού επιπέδου	89
4.2 Ποιότητα Εμπειρίας του χρήστη	90

4.3	Η μεριά του πελάτη	91
4.3.1	Χρήση πληρεξουσίας οντότητας	92
4.3.2	Χρήση Σημασιολογιών	94
4.3.3	Αξιολόγηση ποιότητας κατά τη διαδικασία εξεύρεσης μιας υπηρεσίας	95
4.3.4	Σχήματα συνεργατικής αξιολόγησης ποιότητας από τους πελάτες	98
4.3.5	Αξιολόγηση ποιότητας κατά τη Σύνθεση Υπηρεσιών σε ροές εργασίας .	100
4.4	Η μεριά του παρόχου.....	101
4.4.1	Αντιστοίχιση παραμέτρων υψηλού και χαμηλού επιπέδου	103
4.4.2	Βελτιστοποίηση λειτουργίας παρόχου σε πραγματικό χρόνο	105
4.4.3	Προσαρμογή των SLAs βάσει της συμπεριφοράς του παρόχου	106
4.4.4	Ενοικίαση πόρων	107
5	Πολυδιάστατη Συνεργατική Αξιολόγηση Ποιότητας Υπηρεσίας από το μέρος του Πελάτη	109
5.1	Κύκλος ζωής υπηρεσίας.....	113
5.1.1	Δημοσίευση Υπηρεσίας	113
5.1.2	Ανακάλυψη Υπηρεσίας.....	114
5.1.3	Επιλογή Υπηρεσίας.....	115
5.1.4	Κλήση Υπηρεσίας (δέσμευση) και Παρακολούθηση.....	117
5.1.5	Αξιολόγηση Υπηρεσίας	118
5.2	Μοντέλο πρόβλεψης αξιολογήσεων	118
5.2.1	Μοντελοποίηση του προβλήματος	121
5.2.2	Πρόβλεψη αξιολόγησης με χρήση συνεργατικών τεχνικών φιλτραρίσματος και επιλογή υπηρεσίας.....	122
5.3	Αρχιτεκτονική Υλοποίησης	126
5.3.1	Αλληλεπιδράσεις κατά την αξιολόγηση και επιλογή υπηρεσίας.....	128
5.3.2	Αλληλεπιδρούσες οντότητες.....	129
5.3.3	Ροή των δεδομένων.....	130
5.4	Πειράματα	134
5.4.1	Περιβάλλον αξιολόγησης	134
5.4.2	Μέθοδος Αξιολόγησης του προτεινόμενου μηχανισμού.....	135
5.4.3	Εκτεταμένη αξιολόγηση με προσομοιωμένο σύνολο δεδομένων	139
5.4.4	Πρακτική Επαλήθευση	149
5.5	Συμπεράσματα	156
6	Αξιολόγηση και Δυναμική Διαχείριση Πόρων από το μέρος του Παρόχου	160
6.1	Μοντέλο αξιολόγησης και διαχείρισης πόρων.....	164
6.1.1	Μοντελοποίηση προβλήματος.....	164
6.1.2	Απλοποίηση παραμέτρων για μία διάσταση.....	165
6.2	Αναγωγή στο πρόβλημα του σακιδίου.....	167
6.2.1	Ορισμός του προβλήματος του σακιδίου.....	167
6.2.2	Αναγωγή του προβλήματος διαχείρισης πόρων	168
6.2.3	Εναλλακτικοί αλγόριθμοι για την επίλυση του προβλήματος Knapsack .	175
6.2.4	Το πρόβλημα σε πολλές διαστάσεις	176
6.3	Αρχιτεκτονική υλοποίησης και αξιολόγησης	179
6.3.1	Επίπεδο διαχείρισης πόρων	179
6.3.2	Υπηρεσία Ιστορικού και Εκτίμησης Παραβιάσεων	180
6.3.3	Σχήμα διαχείρισης πόρων στο μεσολογισμικό GRIA	181

6.3.4	Υλοποίηση νέου διαχειριστή πόρων.....	182
6.4	Πειράματα	187
6.4.1	Μέθοδος Αξιολόγησης.....	187
6.4.2	Πειραματικά δεδομένα.....	188
6.4.3	Αποτελέσματα πειραμάτων	192
6.5	Συμπεράσματα	198
7	Συμπεράσματα και μελλοντικές ερευνητικές κατευθύνσεις.....	202
7.1	Σύνοψη και Συμπεράσματα.....	202
7.2	Ερευνητικές κατευθύνσεις και μελλοντική εργασία	206
	Γλωσσάριο	209
	Βιβλιογραφικές Αναφορές.....	211

Ευρετήριο Σχημάτων

Σχήμα 1: Δομή Υπηρεσιοστρεφούς Αρχιτεκτονικής.....	4
Σχήμα 2: Υπηρεσιοστρεφές μεσολογισμικό του ερευνητικού έργου +Spaces, με διασυνδεδεμένες διαδικτυακές υπηρεσίες	5
Σχήμα 3: Χρήση SOAP Πρωτοκόλλου σε εφαρμογές	12
Σχήμα 4: Αρχιτεκτονική τριών επιπέδων του OGSA.....	43
Σχήμα 5: Αρχιτεκτονική GRIA.....	46
Σχήμα 6: Πλαίσιο λειτουργίας υπολογιστικού νέφους.....	53
Σχήμα 7: Ροή δεδομένων στο +Spaces	55
Σχήμα 8: Αρχιτεκτονική της πλατφόρμας του +Spaces	57
Σχήμα 9: Ροή δεδομένων στο SocIoS	59
Σχήμα 10: Αρχιτεκτονική της πλατφόρμας του SocIoS	60
Σχήμα 11: Η ποικιλία των μοντέλων SLA.....	65
Σχήμα 12: Εννοιολογικό μοντέλο πρότυπης διεπίπεδης υπηρεσίας WS-Agreement.....	73
Σχήμα 13: Χρήση της πλατφόρμας του SocIoS για εύρεση χρηστών/δεδομένων από τα κοινωνικά δίκτυα και επεξεργασία τους.....	76
Σχήμα 14: Διάγραμμα συστατικών λογισμικού SocIoS Middleware.....	77
Σχήμα 15: Δομή SocIoS SLA Manager.....	78
Σχήμα 16: Στάδια επεξεργασίας εφαρμογής 3D Rendering	79
Σχήμα 17: Παράδειγμα γράφου ν-γραμμμάτων "home_phone" για την εξαγωγή χαρακτηριστικών ανάλυσης.....	84
Σχήμα 18: Εξαγωγή χαρακτηριστικών ομοιότητας γράφων μηνυμάτων	85
Σχήμα 19: Γράφημα συσχέτισης διανυσμάτων 2-διαστάσεων. Ο κύκλος δείχνει την "γειτνίαση" του διανύσματος v_i , με διανύσματα με έντονη συσχέτιση.....	112
Σχήμα 20: Γενική απεικόνιση της αρχιτεκτονικής αναφοράς	120
Σχήμα 21: Γενική Αρχιτεκτονική Υλοποίησης Μηχανισμού SR.....	127
Σχήμα 22: Ακολουθιακό διάγραμμα αλληλεπιδράσεων κατά την Αξιολόγηση Υπηρεσίας	128
Σχήμα 23: Ακολουθιακό διάγραμμα αλληλεπιδράσεων κατά την Επιλογή Υπηρεσίας	128
Σχήμα 24: Διεπαφές πελάτη GRIA για εξεύρεση και αξιολόγηση υπηρεσιών	135
Σχήμα 25: Η αλληλεπίδραση των οντοτήτων του συστήματος για την αξιολόγηση και παροχή συστάσεων	137
Σχήμα 26: Διακριτά βήματα του Μοντέλου της Υπηρεσίας Συστάσεων	138
Σχήμα 27: Κατανομή Weibull με $\alpha=12$, $\beta=4,2$, μέση τιμή=4.....	145
Σχήμα 28: Πραγματικές τιμές και μέση τιμή για την p_1 (αριστερά)/ Τιμές πρόβλεψης και μέση τιμή για την p_1 (δεξιά).....	147
Σχήμα 29: Πραγματικές τιμές και μέση τιμή για την p_2 (αριστερά)/ Τιμές πρόβλεψης και μέση τιμή για την p_2 (δεξιά).....	147
Σχήμα 30: Πραγματικές τιμές και μέση τιμή για την p_3 (αριστερά)/ Τιμές πρόβλεψης και μέση τιμή για την p_3 (δεξιά).....	147
Σχήμα 31: Πραγματικές τιμές και μέση τιμή για την p_4 (αριστερά)/ Τιμές πρόβλεψης και μέση τιμή για την p_4 (δεξιά).....	148

Σχήμα 32: Πειραματική διαδικασία (Φάση A).....	152
Σχήμα 33: Πειραματική διαδικασία (Φάση B).....	155
Σχήμα 34: Μέσο ποσοστιαίο σφάλμα στις 35 περιπτώσεις σύγκρισης αξιολογήσεων..	156
Σχήμα 35: Το πρόβλημα διαχείρισης πόρων μοντελοποιημένο ως το κλασικό Knapsack πρόβλημα	168
Σχήμα 36: Μια νεοεισερχόμενη εργασία j_i ζητά περισσότερους πόρους από τους διαθέσιμους.....	169
Σχήμα 37: Εκτέλεση της εργασίας με την υψηλότερη αναλογία p_i / w_i στο σύστημα. Οι υπόλοιπες περιμένουν στην ουρά	170
Σχήμα 38: Μια νέα εργασία j_{R+1} εισέρχεται στην ουρά, με υψηλότερη προτεραιότητα από την εργασία j_{i-1} που ήδη εκτελείται.....	171
Σχήμα 39: Δεσμεύονται πόροι για την εργασία j_{R+1} , η εργασία δουλειά j_{i-1} πάει στην ουρά	172
Σχήμα 40: Παραδείγματα ασυσχέτιστων, χαλαρά συσχετισμένων και στενά συσχετισμένων δεδομένων βαρών και κερδών.....	175
Σχήμα 41: Διεπαφές υπηρεσίας ιστορικού και Resource Manager Plugin.....	183
Σχήμα 42: Ροή δεδομένων κατά την υποβολή μιας νέας εργασίας	186
Σχήμα 43: Συνολική απεικόνιση αρχιτεκτονικής αξιολόγησης από τον πελάτη και τον πάροχο με όλα τα εμπλεκόμενα στοιχεία	186
Σχήμα 44: Ποσοστιαία χρησιμοποίηση πόρων με το μετα-δρομολογητή σακιδίου, κέρδος = 16557 (1 ^η Επανάληψη)	193
Σχήμα 45: Ποσοστιαία χρησιμοποίηση πόρων με υλοποίηση πολιτικής FCFS μετα-δρομολόγησης, κέρδος = 12540 (1 ^η Επανάληψη)	193
Σχήμα 46: Ποσοστιαία χρησιμοποίηση πόρων με τον απλό δρομολογητή του GRIA, κέρδος = 11670 (1 ^η Επανάληψη).....	193
Σχήμα 47: Ποσοστιαία χρησιμοποίηση πόρων με το μετα-δρομολογητή σακιδίου, κέρδος = 16244 (2 ^η Επανάληψη)	194
Σχήμα 48: Ποσοστιαία χρησιμοποίηση πόρων με υλοποίηση πολιτικής FCFS μετα-δρομολόγησης, κέρδος = 14866 (2 ^η Επανάληψη)	194
Σχήμα 49: Ποσοστιαία χρησιμοποίηση πόρων με τον απλό δρομολογητή του GRIA, κέρδος = 11198 (2 ^η Επανάληψη).....	194
Σχήμα 50: Ποσοστιαία χρησιμοποίηση πόρων με το μετα-δρομολογητή σακιδίου, κέρδος = 11843 (3 ^η Επανάληψη)	195
Σχήμα 51: Ποσοστιαία χρησιμοποίηση πόρων με υλοποίηση πολιτικής FCFS μετα-δρομολόγησης, κέρδος = 10301 (3 ^η Επανάληψη)	195
Σχήμα 52: Ποσοστιαία χρησιμοποίηση πόρων με τον απλό δρομολογητή του GRIA, κέρδος = 4259 (3 ^η Επανάληψη).....	195

Ευρετήριο Πινάκων

Πίνακας 1: Η βασική δομή του WSDL	9
Πίνακας 2: Η βασική δομή ενός απλού tModel.....	11
Πίνακας 3: Η δομή ενός SOAP μηνύματος.....	13
Πίνακας 4: Απλοποιημένο σχήμα της βασικής δομής του WS-Agreement	72
Πίνακας 5: SLA Template στο GRIA για υπηρεσία 3D Rendering	81
Πίνακας 6: SLA Template κατηγορίας Gold στο SocIoS για υπηρεσία ανάλυσης συναισθημάτων	86
Πίνακας 7: Μέσες βαθμολογίες ενός τυχαίου πωλητή του e-Bay	119
Πίνακας 8: Επέκταση του σχήματος του WS-Agreement με αξιολογήσεις στους όρους του (παρουσιάζεται μόνο το τροποποιημένο στοιχείο GuaranteeTerm)	133
Πίνακας 9: Οι όροι του SLA για την υπηρεσία που επιτελεί 3D video rendering	140
Πίνακας 10: Μέση αξιολόγηση της ποιότητας 5 υπηρεσιών από 7 animators με βάση 4 παραμέτρους ποιότητας	141
Πίνακας 11: Τα στοιχεία των δέκα κατανομών Weibull που χρησιμοποιούνται στην προσομοίωση	145
Πίνακας 12: Μέσα ποσοστιαία σφάλματα πρόβλεψης.....	148
Πίνακας 13: Πραγματικές και προβλεπόμενες βαθμολογίες για τον Animator 1	153
Πίνακας 14: Μήτρα συσχετίσεων για τον animator 1. Το $\text{corr}(1, X)$ δείχνει τον συντελεστή συσχέτισης μεταξύ animator 1 και animator X. Οι τιμές κυμαίνονται από [- 1, 1], με το 0 να συνεπάγεται μηδενική συσχέτιση και τα 1 και -1 υψηλή συσχέτιση....	154
Πίνακας 15: Ένα παράδειγμα JSDL εγγράφου Περιγραφής Εργασίας του GRIA.....	185
Πίνακας 16: Εργασίες 3D Rendering υπο εκτέλεση. Τα ονόματα των εργασιών δείχνουν τη διαμόρφωσή των αρχείων (αριθμός αντικειμένων, φώτων, κλπ.).	189
Πίνακας 17: Κατηγορίες τιμολόγησης κόστους παραβίασης.....	191
Πίνακας 18: Σύνοψη των πειραματικών αποτελεσμάτων	196

Ευρετήριο Εξισώσεων

Εξίσωση 1: Ορισμός του συντελεστή συσχέτισης του Pearson για μια παράμετρο	123
Εξίσωση 2: Πρόβλεψη αξιολόγησης για μια παράμετρο με βάση τους συντελεστές συσχέτισης	124
Εξίσωση 3: Ευκλείδεια νόρμα διαφοράς σταθμισμένου διανύσματος πρόβλεψης από το ιδανικό.....	126
Εξίσωση 4: Συνάρτηση πυκνότητας πιθανότητας Weibull	142
Εξίσωση 5: Πρώτο μέτρο του κέρδους εκτέλεσης για την εργασία j_i	173
Εξίσωση 6: Κόστος που εξοικονομεί ο πάροχος από επιτυχή εκτέλεση της εργασίας j_i	173
Εξίσωση 7: Συνολική συνάρτηση κέρδους για μια εργασία j_i	174
Εξίσωση 8: Υπολογισμός πολυδιάστατης αναλογίας κέρδους / βάρους για μια εργασία	177
Εξίσωση 9: Ταξινομημένος πίνακας προτεραιοτήτων με τις αναλογίες κερδών / βάρη	178
Εξίσωση 10: Η Markovian ιδιότητα για την ουρά του μετα-δρομολογητή.....	190

Περίληψη

Με την εμφάνιση των υπηρεσιοστρεφών τεχνολογιών και υποδομών και την υιοθέτηση των ηλεκτρονικών συμβάσεων μεταξύ παρόχων υπηρεσιών και πελατών, εμφανίστηκε η ανάγκη ελέγχου και επικύρωσης της προσφερόμενης ποιότητας καθ' όλη τη διάρκεια του κύκλου ζωής μιας υπηρεσίας. Η παρακολούθηση και η αξιολόγηση αυτής γίνεται είτε στην πλευρά του πελάτη, βάσει της ποιότητας της πρότερης εμπειρίας του, χρησιμοποιώντας μηχανισμούς για την κατάταξη και επιλογή των εναλλακτικών υπηρεσιών, ή στην πλευρά του παρόχου, με δυναμική αναδιαμόρφωση των υπηρεσιών και την αντίστοιχη ανακατανομή των πόρων, προκειμένου να βελτιστοποιηθούν οι παράμετροι ποιότητας με βάση τις εγγυήσεις του. Το τελευταίο, αρχικά απαιτεί ενός είδους "μετάφραση" των ποιοτικών παραμέτρων υψηλού επιπέδου που είναι πιο κοντά στην αντίληψη του πελάτη, σε όρους χαμηλού επιπέδου που σχετίζονται με τη διαδικασία διαχείρισης των πόρων. Η δυναμική ανακατανομή πόρων με βάση την παρακολούθηση και αξιολόγηση της ποιότητας μπορεί να οδηγήσει σε βελτιστοποίηση της αξιοποίησής τους, όπως και των κερδών του παρόχου.

Η παρούσα διατριβή εστιάζει στα συστήματα παροχής "Εφαρμογής ως Υπηρεσία" και επιδιώκει την εισαγωγή και ανάλυση καινοτόμων μηχανισμών για την αποτελεσματική αξιολόγηση της παρεχόμενης ποιότητας. Έτσι, αρχικά παρουσιάζονται όλες οι τεχνολογίες αιχμής και οι συνήθεις μεθοδολογίες αξιολόγησης της ποιότητας υπηρεσιών

σε τέτοια περιβάλλοντα, με έμφαση στις δύο πτυχές του προβλήματος: την αξιολόγηση από το μέρος του πελάτη, που συνεπάγεται τη δημιουργία συστημάτων φήμης για τους παρόχους ώστε να διευκολύνει τη σωστή επιλογή υπηρεσίας, και την αξιολόγηση από το μέρος του παρόχου, με συνέπεια την καλύτερη διαχείριση και βελτίωση της μελλοντικής ποιότητας παροχής.

Στο πλαίσιο που τέθηκε παραπάνω, το πρώτο σκέλος της παρούσας ερευνητικής εργασίας ασχολείται με το πρόβλημα επιλογής μιας υπηρεσίας από μια πληθώρα διαθέσιμων εφαρμογών που διατίθενται για τον πελάτη. Αξιοποιώντας την εμπειρία των πελατών που εμφανίζουν παρόμοια συμπεριφορά, προτείνεται ένα μοντέλο συνεργατικής αξιολόγησης των παρεχόμενων υπηρεσιών, με βαθμολόγηση των παρόχων και χρήση τεχνικών συσχέτισης, ώστε να προβλεφθεί επιτυχώς η μελλοντική αξιολόγηση υπηρεσιών που δεν έχουν χρησιμοποιηθεί από τους πελάτες. Έτσι, δομείται ένα σύστημα φήμης παρόχων που υποστηρίζει τους χρήστες κατά τη διαδικασία επιλογής υπηρεσίας με την παροχή προσωποποιημένων συστάσεων, με βάση το ιστορικό αξιολόγησής τους και τη συσχέτιση ανάμεσά τους. Η ενσωμάτωση του συστήματος αυτού σε μια υπηρεσία συστάσεων και η χρήση του από ένα λογισμικό πελάτη σε πειράματα με προσομοιωμένα αλλά και πραγματικά δεδομένα, καταδεικνύουν την επιτυχή υλοποίησή του και τη βελτιωμένη υποστήριξη της επιλογής υπηρεσίας, βάσει της πρότερης αξιολόγησης.

Στο δεύτερο σκέλος της διατριβής, παρουσιάζεται ένας μηχανισμός δυναμικής διαχείρισης πόρων για περιβάλλοντα παροχής "Εφαρμογής ως Υπηρεσία", με στόχο τη βέλτιστη αξιοποίησή τους κάτω από μεγάλο φόρτο εργασίας. Χρησιμοποιώντας την ελαστικότητα των σύγχρονων εικονικοποιημένων υποδομών, το προτεινόμενο μοντέλο υπαγορεύει ότι οι πόροι του συστήματος θα πρέπει να αξιοποιηθούν πλήρως από τις

εισερχόμενες σε αυτό εργασίες, ακόμα και στην περίπτωση που δεν πληρούνται οι απαιτήσεις τους, με χρήση τεχνικών διαχείρισης ρίσκου. Οι τεχνικές αυτές χρησιμοποιούν την πιθανότητα παραβίασης των ηλεκτρονικών συμφωνιών, που προκύπτει από το ιστορικό εκτελέσεων, για την αξιολόγηση της παρεχόμενης ποιότητας ανά πάσα στιγμή και την καλύτερη δυνατή κατανομή των πόρων. Το πρόβλημα της κατανομής ανάγεται στο γνωστό πρόβλημα του σακιδίου και μια ευριστική λύση σχεδιάζεται και υλοποιείται σε ένα υπάρχον υπηρεσιοστρεφές σύστημα. Τα αποτελέσματα της δεύτερης δέσμης πειραμάτων με ένα σύνολο εισερχόμενων εργασιών υψηλής επιβάρυνσης στο σύστημα αυτό, δείχνουν βελτιστοποίηση των κερδών του παρόχου υπηρεσιών λόγω της καλύτερης αξιοποίησης της υπολογιστικής υποδομής, ως συνέπεια του μηχανισμού αξιολόγησης και διασφάλισης της ποιότητας με τη δυναμική διαχείριση των πόρων.

Λέξεις κλειδιά: Υπηρεσιοστρεφείς Υποδομές, Εφαρμογή ως Υπηρεσία, Αξιολόγηση Ποιότητας, Τεχνικές Συσχέτισης, Συστήματα Φήμης, Εικονικοποίηση, Δυναμική Ανακατανομή Πόρων, Τεχνικές Διαχείρισης Ρίσκου, Πρόβλημα Σακιδίου

Abstract

With the emergence of service provisioning technologies and infrastructures and the adoption of Service Level Agreements between service providers and customers, the need to control and validate the offered quality has appeared throughout the service lifecycle. This monitoring and evaluation is performed either on the client side, using the customer's quality of experience and employing mechanisms for service classification and selection, or on the provider side, dynamically reconfiguring the service and allocating resources accordingly, in order to optimize the quality metrics guaranteed. The latter, of course, initially requires mapping of the high-level quality parameters, which are closer to the customer perception, to low-level computing terms related to the resource management process. Dynamic resource allocation based on quality monitoring and evaluation can lead to optimizing resource utilization as well as the provider's profits.

This thesis focuses on "Software as a Service" systems and seeks to introduce and analyze innovative mechanisms for the effective evaluation of service quality. So, initially it presents the current state-of-the-art technology and methodologies regarding the evaluation of the provided quality of service in service oriented environments, illustrating that the problem is two-fold: first the evaluation from the customer side,

which entails the creation of reputation systems for the providers, in order to serve service selection, and second the evaluation from the provider side, resulting in better management and improving the future quality of provision.

In that frame, the first part of the current research work tackles the problem of selecting a service from a plethora of available ones for a customer. Exploiting the experience of users that present similar behavior, a collaborative evaluation model of the provided services is proposed, using correlation techniques and rating the providers, to successfully predict future assessment of services that have not been used by customers yet. Thus, a reputation system for providers is developed that supports users in the service selection process by providing personalized recommendations, based on evaluation history and correlations between them. The integration of this system in a recommendation service and its connection with a client software, to conduct experiments with simulated and real datasets, indicates the success of its implementation and the improvement of service selection support, based on prior evaluation.

The second part of the thesis presents a dynamic resource management mechanism for "Software as a Service" systems in order to maximize resource utilization under a heavy load. Employing the elasticity of virtualized infrastructures, the proposed model dictates that system resources must be fully exploited by incoming jobs, even if they do not satisfy their requirements completely, using risk management techniques. Those techniques use the violation probability for Service Level Agreements, apparent from the service history, to evaluate the quality offered at any time and allocate resources in an optimum way. The resource allocation problem is deducted to the well known Knapsack Problem and a heuristic solution is designed and implemented as part of an existing

service oriented system. The second set of experiments, using a dataset of resource demanding incoming jobs, results in the optimization of the service provider profit, thanks to the optimum utilization of computing resources, as a result of the evaluation and quality assurance mechanism via dynamic resource management.

Keywords: Service Oriented Infrastructures, Software as a Service, Quality Evaluation, Correlation Techniques, Reputation Systems, Virtualization, Dynamic Resource Allocation, Risk Management Techniques, Knapsack Problem

1

Εισαγωγή

Το τρέχον κεφάλαιο στοχεύει στην εισαγωγή του αναγνώστη στην έννοια των υπηρεσιών του παγκόσμιου ιστού και των υπηρεσιοστρεφών αρχιτεκτονικών. Αρχικά λοιπόν, παρατίθεται μια παράγραφος που αναλύει τις σύγχρονες δομές στο διαδίκτυο και την ανάγκη που δημιουργήθηκε για τη χρησιμοποίηση υπηρεσιών διαδικτύου σε εμπορικές και μη εφαρμογές. Εν συνεχεία εισάγεται η έννοια της υπηρεσιοστρεφούς αρχιτεκτονικής και παρουσιάζονται συνοπτικά τα πρωτόκολλα SOAP, UDDI και η γλώσσα WSDL. Αναφέρονται τα διάφορα υπάρχοντα σχήματα υπηρεσιών, με βάση το τι παρέχεται σαν υπηρεσία κάθε φορά, όπως Software as a Service, Platform as a Service και Infrastructure as a Service, με έμφαση στο πρώτο που θα αποτελέσει και το αντικείμενο ανάλυσης της παρούσας διατριβής. Τέλος αναφέρονται οι συνεισφορές της διατριβής, αναδεικνύοντας την καινοτομία της, και η οργάνωση του παρόντος κειμένου.

1.1

Υπηρεσίες στο διαδίκτυο

Κατά την τελευταία δεκαετία, ο τομέας των υπηρεσιών έχει καταστεί ο μεγαλύτερος και ταχύτερα αναπτυσσόμενος τομέας των επιχειρήσεων στον κόσμο. Για τη συνέχιση αυτής της ανάπτυξης, οι υπηρεσίες χρειάζεται να γίνουν πιο ευρέως και εύκολα διαθέσιμες,

όπως επίσης και να συμβάλλουν στην αύξηση της παραγωγικότητας. Με δεδομένη την τεράστια ανάπτυξη του διαδικτύου, ήδη από το τέλος του προηγούμενου αιώνα, έγινε σαφές ότι η έξυπνη χρήση της τεχνολογίας της πληροφορικής μπορεί να βοηθήσει σημαντικά στην επίτευξη των παραπάνω στόχων.

Η εμφάνιση της ιδέας για το «Διαδίκτυο των Υπηρεσιών» (Internet of Services - IoS) τα τελευταία χρόνια έχει ως σκοπό να επεκτείνει δραματικά την έννοια του αρχικού διαδικτύου, ως μια απλή δομή διασύνδεσης βάσεων στατικών δεδομένων που υπήρχε μέχρι πρόσφατα, και να εισαγάγει τη δυνατότητα δυναμικής εξυπηρέτησης. Ήδη με την εμφάνιση της ηλεκτρονικής μάθησης (e-Learning), της ηλεκτρονικής διακυβέρνησης (e-Government) και παρόμοιων τεχνολογιών, το διαδίκτυο άρχισε να παίρνει τελείως διαφορετική μορφή εξυπηρετώντας καθημερινές ανάγκες των πολιτών, ενώ με την εμφάνιση των κοινωνικών δικτύων (Social Networks), η δυναμική των υπηρεσιών αυτών καθίσταται πλέον τεράστια. Διάφορες εταιρείες και ερευνητικά ιδρύματα έχουν αρχίσει να εξερευνούν τις διάφορες πτυχές του τομέα των υπηρεσιών, για να καθορίσουν ποιες μπορούν να παρασχεθούν ως η επόμενη γενιά υπηρεσιών του διαδικτύου και, εφόσον μπορούν συνδυαστούν, να προσφερθούν ίσως και σαν υπηρεσίες προστιθέμενης αξίας.

Στο παραπάνω επιχειρηματικό μοντέλο, η συμμετοχή όλων των ειδών "παικτών", από μεγάλους παρόχους υπηρεσιών έως και απλούς οικιακούς χρήστες, αναμένεται να δημιουργήσει μια ανοιχτή αγορά υπηρεσιών. Σε μια τέτοια αγορά, η ζήτηση και η προσφορά καθορίζουν την πολυπλοκότητα των υπηρεσιών που παρέχουν τις βασικές λειτουργικότητες, οι οποίες δυνητικά μπορούν να συνδυαστούν ώστε να δημιουργήσουν μεγαλύτερες ροές εργασίας (workflows). Η διάκριση μεταξύ παροχών υπηρεσιών και πελατών σε μια τέτοια δαιδαλώδη αγορά δεν είναι τόσο σαφής, όσο είναι στον

πραγματικό κόσμο, δημιουργώντας την ανάγκη να οργανωθεί όσο το δυνατόν αποτελεσματικότερα η παροχή των υπηρεσιών στα πλαίσια της τεράστιας ανταγωνιστικότητας, αλλά και να αξιολογηθούν διάφορα κριτήρια στη διαδικασία επιλογής των υπηρεσιών αυτών.

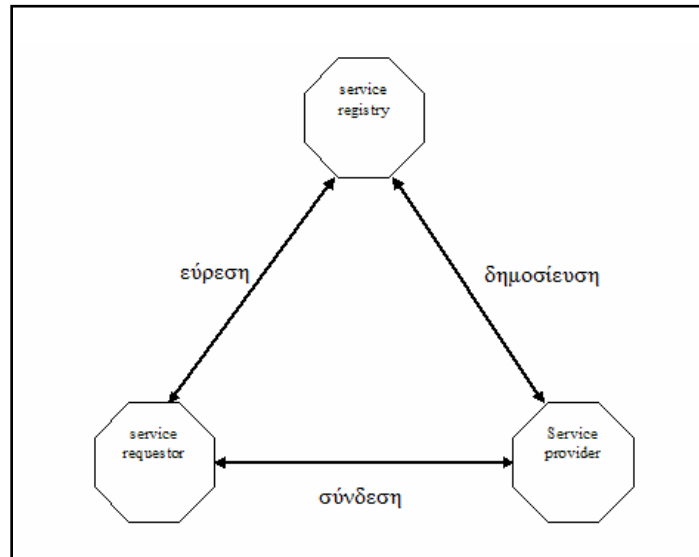
1.2 Υπηρεσιοστρεφής Αρχιτεκτονική

Στην προηγούμενο παράγραφο αναφέρθηκαν τύποι απλών υπηρεσιών που αποκτούν πρόσθετη αξία όντας διαθέσιμες στο διαδίκτυο. Η παροχή μιας εφαρμογής ως υπηρεσία είναι μια σχετικά απλή διαδικασία και επιπρόσθετα αποδίδει στην εφαρμογή πρόσθετα λειτουργικά χαρακτηριστικά. Τα χαρακτηριστικά αυτά όμως δεν δημιουργούν μια υπηρεσιοστρεφή αρχιτεκτονική (Service Oriented Architecture - SOA), αφού η διαφορά τους είναι πιο μεγάλη.

Το πρότυπο υπηρεσιοστρεφούς αρχιτεκτονικής είναι ένα σχεδιαστικό μοντέλο με κύριο χαρακτηριστικό την ενσωμάτωση λογικής εφαρμογών μέσα σε υπηρεσίες που θα αλληλεπιδρούν μέσω συγκεκριμένων επικοινωνιακών πρωτοκόλλων. Πιο συγκεκριμένα, οι υπηρεσιοστρεφείς αρχιτεκτονικές είναι ένα αρχιτεκτονικό είδος, των οποίων ο στόχος είναι να επιτευχθεί ένα είδος "χαλαρής" σύνδεσης μεταξύ αλληλεπιδρώντων λογισμικών. Κάθε υπηρεσία σε τέτοιες αρχιτεκτονικές είναι μια μονάδα εργασίας που γίνεται διαθέσιμη από έναν πάροχο, ώστε να παρέχει τα επιθυμητά τελικά αποτελέσματα στον καταναλωτή υπηρεσίας. Βάσει αυτού, η υιοθέτηση σε μια εφαρμογή μιας SOA δομής σημαίνει αυτόματα την αποδοχή κάποιων σχεδιαστικών αρχών και πρόσθετων τεχνολογιών ως βασικού τμήματος του τεχνικού περιβάλλοντός της.

Σε επίπεδο σχεδιασμού συστημάτων, η χρήση της τεχνολογίας των υπηρεσιών διαδικτύου (Web Services) οδήγησε στην ευρύτερη υιοθέτηση της υπηρεσιοστρεφούς

αρχιτεκτονικής. Οι βασικοί ρόλοι και λειτουργίες στην αρχιτεκτονική αυτή παρουσιάζονται στο Σχήμα 1 (Menychtas, 2009):



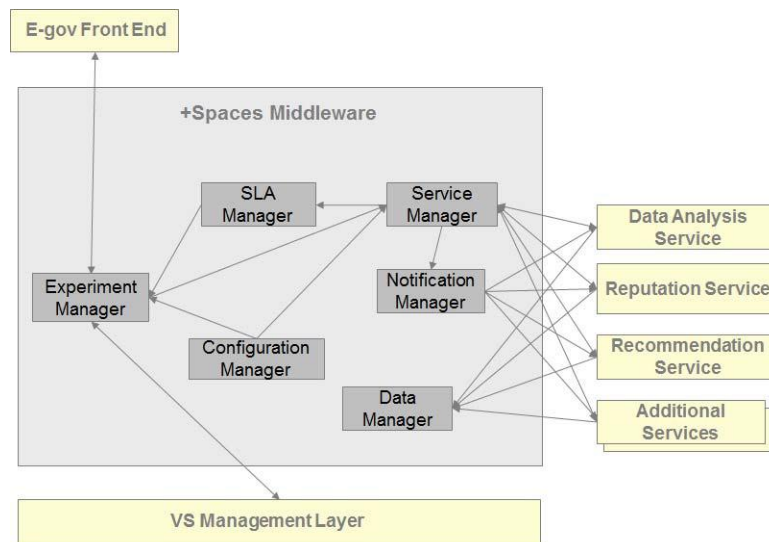
Σχήμα 1: Δομή Υπηρεσιοστρεφούς Αρχιτεκτονικής

Η αρχιτεκτονική αυτή υποδεικνύει μια σχέση εξυπηρετητή-πελάτη (server-client) ανάμεσα στον πάροχο υπηρεσιών (service provider, που παίζει το ρόλο του server) και τον πελάτη - ζητούντα την υπηρεσία (service requestor, που παίζει το ρόλο του client). Ο service provider είναι αυτός που παρέχει την υπηρεσία δεχόμενος μηνύματα και κλήσεις από τους service requestors. Είναι επίσης υπεύθυνος για τη δημιουργία της περιγραφής της υπηρεσίας (service description) και τη δημοσίευσή της σε κάποιο κατάλογο-οδηγό υπηρεσιών (Universal Description, Discovery and Integration - UDDI). Ο service requestor αναζητά μια συγκεκριμένη υπηρεσία μέσω της περιγραφής της σε κάποιο κατάλογο υπηρεσιών (service registry) και στη συνέχεια καλεί την επιθυμητή υπηρεσία. Ο κατάλογος υπηρεσιών φέρνει ουσιαστικά τις δύο πλευρές, client και server, σε επαφή.

Η συνέχεια αφορά μόνο τις δύο συμμετέχουσες μονάδες (service requestor και service provider), χωρίς συνήθως να παρεμβαίνουν εξωτερικές οντότητες.

1.2.1 Γενικά για τις υπηρεσίες σε SOA

Οι υπηρεσίες τείνουν να γίνουν τμήματα της εφαρμογής αθροιστικά σχηματίζοντας το περιβάλλον της. Δεν αποτελούν απλά ένα κομμάτι της, αλλά έχουν χαρακτηριστικά που τις μετατρέπουν σε μέρος μιας αρχιτεκτονικής προσανατολισμένης στις υπηρεσίες (SOA). Ένα από τα χαρακτηριστικά αυτά είναι η αυτονομία από άλλες υπηρεσίες. Αυτό σημαίνει ότι κάθε υπηρεσία είναι υπεύθυνη για το δικό της εύρος λειτουργίας, περιορίζοντας και εξειδικεύοντάς την σε συγκεκριμένες επαγγελματικές χρήσεις. Αυτός ο σχεδιασμός έχει ως αποτέλεσμα τη δημιουργία ανεξάρτητων μονάδων, δυναμικά συνδεδεμένων μεταξύ τους με κάποιο πρότυπο πλαίσιο επικοινωνίας, όπως για παράδειγμα η αρχιτεκτονική του Σχήματος 2 μιας πλατφόρμα υπηρεσιών e-Government (Kardara, Fuchs, Aisopos, Papaoikonomou, Tserpes, & Varvarigou, 2011):



Σχήμα 2: Υπηρεσιοστρεφές μεσολογισμικό του ερευνητικού έργου +Spaces, με διασυνδεδεμένες διαδικτυακές υπηρεσίες

Εξαιτίας αυτής της ανεξαρτησίας που απολαμβάνουν οι υπηρεσίες στο πλαίσιο αυτό, η προγραμματιστική λογική που κάθε υπηρεσία χρησιμοποιεί, δεν χρειάζεται να προσαρμόζεται σε συγκεκριμένη πλατφόρμα ή τεχνολογία. Έτσι, ένα κύριο χαρακτηριστικό των υπηρεσιών μιας τέτοιας αρχιτεκτονικής, είναι ο πολυμορφισμός των μερών της με ταυτόχρονη ομαλή συνεργασία.

Ο κύκλος ζωής των υπηρεσιών σε SOA, ο οποίος αναλύεται διεξοδικά στο Κεφάλαιο 5, είναι γενικά προδιαγεγραμμένος και εμπεριέχει τα παρακάτω στάδια:

- Δημοσίευση υπηρεσίας: Ο πάροχος υπηρεσιών θέτει μια δημιουργηθείσα υπηρεσία δημόσια διαθέσιμη, ώστε να μπορεί να ανακαλυφθεί από τους ενδιαφερόμενους πελάτες.
- Ανακάλυψη υπηρεσίας: Ο πελάτης ανακαλύπτει την επιθυμητή υπηρεσία μέσω συγκεκριμένων πρωτοκόλλων και μητρώων, που θα παρουσιασθούν παρακάτω.
- Διαπραγμάτευση και επιλογή υπηρεσίας: Ο πελάτης διαπραγματεύεται με τον πάροχο τις παραμέτρους κλήσης και τη χρέωση της υπηρεσίας. Κατόπιν συμφωνίας, ο πελάτης επιλέγει τη συγκεκριμένη υπηρεσία για χρησιμοποίηση.
- Κλήση υπηρεσίας: Ο πελάτης καλεί την υπηρεσία δίνοντας δεδομένα εισόδου και περιμένοντας την απόκριση και τα ζητούμενα αποτελέσματα.
- Παρακολούθηση υπηρεσίας: Ο πάροχος (ή και ο πελάτης) μπορούν να παρακολουθούν καθ' όλη τη διάρκεια της κλήσης τη χρησιμοποίηση των πόρων (ή τις τιμές των παραμέτρων κλήσης)
- Αξιολόγηση υπηρεσίας: Μετά το πέρας της κλήσης και σύμφωνα με την παρακολούθηση κατά τη διάρκειά της, οι δύο συμβαλλόμενοι μπορούν να αξιολογήσουν κατά πόσο τηρήθηκε η προαναφερθείσα συμφωνία.

1.2.2 Διαδικτυακές Υπηρεσίες

Στην προαναφερθείσα αρχιτεκτονική, ο πιο γνωστός τύπος υπηρεσιών που έχει επικρατήσει είναι οι λεγόμενες διαδικτυακές υπηρεσίες XML Services, γνωστές ως *Web Services*. Μία διαδικτυακή υπηρεσία είναι μια καλά καθορισμένη συνάρτηση η οποία είναι προσπελάσιμη μέσω του διαδικτύου. Οι υπηρεσίες αυτές επικοινωνούν μέσω συγκεκριμένων πρωτοκόλλων Internet (κυρίως HTTP), ενώ για την αποστολή και παραλαβή δεδομένων χρησιμοποιούν XML αρχεία συγκεκριμένης δομής, όπως θα δούμε παρακάτω. Η ύπαρξη ενός κοινού συντακτικού (XML) επιτρέπει στις υπηρεσίες να απαλλαγθούν από τις τυπικές ενέργειες που χρειάζονται για την επικοινωνία. Αυτές τις αναλαμβάνουν πλέον οι εκάστοτε εξυπηρετητές που φιλοξενούν τις υπηρεσίες. Έτσι οι υπηρεσίες μπορούν εύκολα να γραφτούν, να αντικατασταθούν, να αναπτυχθούν και να συντηρηθούν. Η τεχνολογία των διαδικτυακών υπηρεσιών καθορίζει μια κοινή πλατφόρμα επικοινωνίας με την οποία μπορούν να δημιουργηθούν διασυνδέσεις σε διάφορες συναρτήσεις. Επιπρόσθετα, είναι συνηθισμένο μια υπηρεσία να λειτουργεί και ως πελάτης (client/requestor) και ως πάροχος (provider) υπηρεσίας. Αναλόγως λοιπόν με τη δραστηριότητα της υπηρεσίας κάθε στιγμή, μετατρέπεται από το ένα στο άλλο.

Η "παραγωγή" τέτοιων υπηρεσιών απαιτεί αρχικά την περιγραφή της υπηρεσίας αναλυτικά (πχ. με ένα WSDL ή ένα WADL έγγραφο κατάλληλα διαρθρωμένο) και τη δυνατότητα μεταφοράς ενός XML εγγράφου (πχ. με SOAP μηνύματα) μέσω του πρωτοκόλλου HTTP. Οι υπηρεσίες που χρησιμοποιούν SOAP, ήταν αυτές που αναπτύχθηκαν πρώτα και υιοθετήθηκαν ευρύτατα, λόγω της ασφάλειας που παρέχουν. Ωστόσο, λόγω "βάρους" που προσδίδει η επιπρόσθετη ασφάλεια και της πολυπλοκότητάς τους, προσφάτως αναπτύχθηκαν νέες "ελαφρές" υπηρεσίες τύπου REST

(Representational State Transfer), οι οποίες συχνά περιγράφονται από ένα WADL έγγραφο και μπορούν να κληθούν απλά με μία HTTP GET αίτηση μέσω ενός συγκεκριμένου URL. Παρόλη την απλότητά τους ωστόσο, οι RESTful Web Services κρίνονται μη ασφαλείς για τις περισσότερες εμπορικές εφαρμογές, ενώ δεν μπορούν να υποστηρίξουν επαρκώς αναγκαίες λειτουργικότητες, όπως τη διαχείριση Service Level Agreements. Επιπροσθέτως, το WADL δεν έχει τη δυναμικότητα του WSDL, το οποίο θα παρουσιαστεί παρακάτω. Η παρούσα εργασία εξετάζει κυρίως διαδικτυακές υποδομές που χρησιμοποιούν SOAP, οι οποίες παρέχουν όλες τις επιθυμητές δυνατότητες.

Η ευρεία αποδοχή του μοντέλου των Web Services είχε ως αποτέλεσμα την ανάγκη πρόσθετων τεχνολογιών βασισμένων σε αυτές και τη δημιουργία καινούργιων προτύπων. Κάποια χαρακτηριστικά τέτοια πρότυπα, βασισμένα σε Web Services, είναι τα παρακάτω:

- **WS-Security:** Μια δυναμική επέκταση του SOAP για την εφαρμογή μεγαλύτερης ασφάλειας σε Web Services, με τη χρήση κρυπτογράφησης και υπογεγραμμένων πιστοποιητικών.
- **WS-Addressing:** Ένα πρότυπο μηχανισμών που επιτρέπουν στις υπηρεσίες να ανταλλάξουν πληροφορίες που αφορούν ηλεκτρονικές διευθύνσεις.
- **WS-Policy:** Ένα πρότυπο που χρησιμοποιείται ώστε οι υπηρεσίες να διαφημίσουν την πολιτική ασφάλειας ή ποιότητάς τους και οι πελάτες να καθορίσουν τις απαιτήσεις τους.
- **WS-Notification:** Ένα επεκτάσιμο πρότυπο διεπαφών και πρωτοκόλλων για την αποστολή μηνυμάτων μεταξύ υπηρεσιών κατά τα πρότυπα του σχήματος «συνδρομής - ειδοποίησης».

- **WS-Agreement:** Ένα επεκτάσιμο πρότυπο διεπαφών και μοντέλων για τη δημιουργία ηλεκτρονικών συμβολαίων, που θα αναλυθεί διεξοδικά παρακάτω.

1.2.2.1 *Web Services Description Language (WSDL)*

Οι διαδικτυακές υπηρεσίες χρειάζεται να ορίζονται με συγκεκριμένο τρόπο έτσι ώστε να μπορούν να εντοπιστούν και χρησιμοποιηθούν από τους πελάτες ή άλλες υπηρεσίες και εφαρμογές. Για αυτό το σκοπό δημιουργήθηκε από τον οργανισμό W3C, μια γλώσσα περιγραφής γνωστή ως *Web Services Description Language (WSDL)*. Η γλώσσα WSDL καθορίζει ένα XML έγγραφο με μια συγκεκριμένη δομή που περιέχει όλες τις πληροφορίες σχετικά με τα δεδομένα εισόδου και εξόδου, τις μεθόδους και ότι άλλο χρειάζεται να γνωρίζει κάποιος για την ακριβή χρήση μιας διαδικτυακής υπηρεσίας. Η γενική δομή ενός τέτοιου εγγράφου είναι η παρακάτω:

```
<?xml version="1.0"?>
<definitions name="...">
  <documentation>
    ...
  </documentation>
  <types>
    ...
  </types>
  <message name="...">
    <part name="body" element="..."/>
  </message>
  <portType name="...">
    <operation name="...">
      <input message="..."/>
      <output message="..."/>
    </operation>
  </portType>
  <binding name = "..." >
    <operation name="..." >
      <input>
        ...
      </input>
      <output>
        ...
      </output>
    </operation>
  </binding>
  <service name="..." >
    <port name="..." binding="..."/>
  </service>
</definitions>
```

Πίνακας 1: Η βασική δομή του WSDL

Όπως φαίνεται στον Πίνακα 1, η περιγραφή της υπηρεσίας βασικά εμπεριέχει ένα προαιρετικό πλαίσιο ελεύθερου κειμένου (documentation), όπου ο προγραμματιστής μπορεί να γράψει βοηθητικά σχόλια, τον ορισμό μερικών νέων τύπων (types) και διεπαφών (portType) που πιθανόν χρησιμοποιεί και τον καθορισμό των στοιχειωδών υπηρεσιών που παρέχονται. Για κάθε υπηρεσία (service), καθορίζεται μια συγκεκριμένη διεπαφή (port), δεμένη με μια συγκεκριμένη ορισθείσα λειτουργία (operation) με καθορισμένα στοιχεία εισόδου (input) - εξόδου (output).

1.2.2.2 Universal Description, Discovery and Integration (UDDI)

Το UDDI βασίζεται στο αρχικό όραμα μιας «μετα-υπηρεσίας» για τον εντοπισμό των υπηρεσιών διαδικτύου, επιτρέποντας ερωτήματα σε αποθήκες μετα-δεδομένων (Januszewski, Mooney, Harrah, Lee, Munter, & Von Riegen, 2002). Προσφέρει στην αγορά των διαδικτυακών υπηρεσιών ένα μοντέλο για τη δημιουργία δυναμικών και διαλειτουργικών μητρώων υπηρεσιών σε XML, ιδιαίτερα χρήσιμο τόσο για ιδιωτικές όσο και για δημόσια προσβάσιμες υπηρεσίες. Δημιουργήθηκε υπό την αιγίδα του οργανισμού για την Προώθηση των Προτύπων Δομημένων Πληροφοριών (Organization for the Advancement of Structured Information Standards - OASIS), επιτρέποντας στους παρόχους να δημοσιεύουν καταλόγους των υπηρεσιών και να ανακαλύπτουν ο ένας τον άλλον, καθορίζοντας λεπτομερώς πώς οι υπηρεσίες ή οι εφαρμογές λογισμικού αλληλεπιδρούν μέσω του διαδικτύου.

Το UDDI χρησιμοποιεί μηνύματα SOAP για την αποθήκευση νέων υπηρεσιών, άλλα και για να παρέχει πρόσβαση στα WSDL έγγραφα που περιγράφουν τις διαδικτυακές υπηρεσίες που περιλαμβάνονται στον κατάλογο του. Οι υπηρεσίες καταχωρούνται στα μητρώα UDDI σαν WSDL περιγραφές, μοντελοποιώντας τις

διεπαφές και την υλοποίησή τους. Μια υπηρεσία συγκεκριμένης λειτουργικότητας μπορεί στη συνέχεια να αναζητηθεί χρησιμοποιώντας το API του UDDI και ένα tModel, μια XML περιγραφή των συνδέσεων που χρησιμοποιούνται στο UDDI, που μπορεί επίσης να λαμβάνει υπόψη τα στοιχεία του WSDL. Η βασική δομή ενός πολύ απλού tModel φαίνεται στον παρακάτω πίνακα:

```
<tModel tModelKey="...">
  <name>
    ...
  </name>
  <description>
    ...
  </description>
  <overviewDoc>
    <description>
      ...
    </description>
    <overviewURL>
      ...
    </overviewURL>
  </overviewDoc>
</tModel>
```

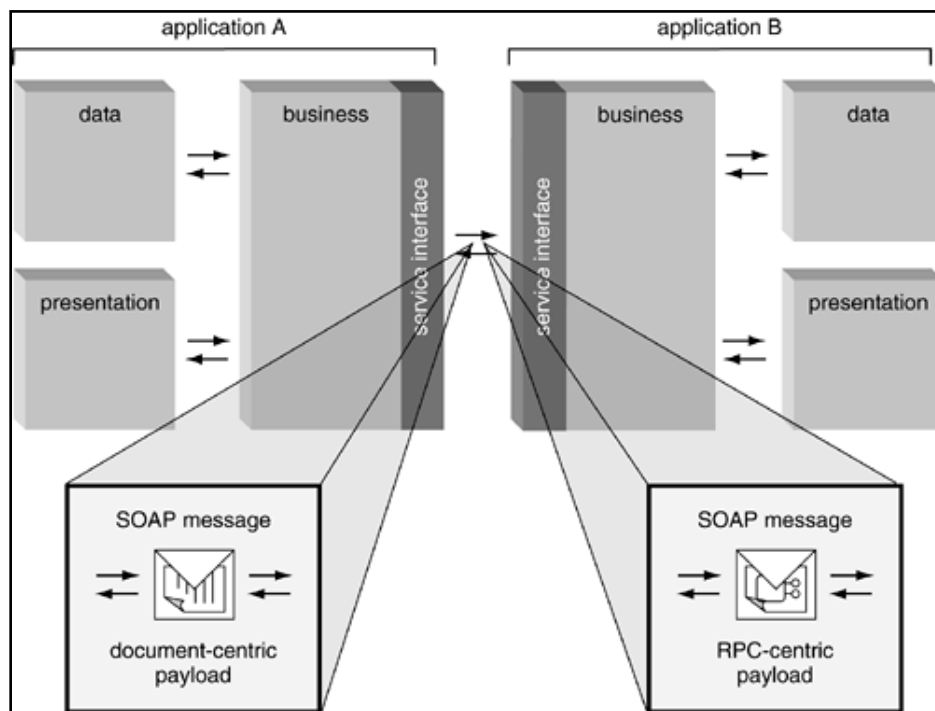
Πίνακας 2: Η βασική δομή ενός απλού tModel

Κάθε tModel έχει ένα μοναδικό κλειδί, ένα όνομα και μια περιγραφή. Οι πληροφορίες για την συγκεκριμένη υπηρεσία υπεισέρχονται κάτω από το στοιχείο overviewDoc, όπου παρατίθεται η περιγραφή (description) και ένα link στο WSDL (overviewURL).

1.2.2.3 Simple Object Access Protocol (SOAP)

Αν και αρχικά είχε θεωρηθεί ως η τεχνολογία που θα γεφυρώσει το κενό μεταξύ ανόμοιων πλατφορμών βασισμένων σε RPC (Remote Procedure Call) επικοινωνία, το SOAP έχει εξελιχθεί στο πιο ευρέως χρησιμοποιούμενο πρότυπο επικοινωνίας για τη χρήση υπηρεσιών διαδικτύου. Μετά από αυτή την εξέλιξη γίνεται πολλές φορές η παράφραση του ακρωνύμιου SOAP από Simple Object Access Protocol σε Service-Oriented Architecture (ή Application) Protocol (Kyriazis, 2007).

Το πρωτόκολλο SOAP διαμορφώνει ένα πρότυπο μήνυμα που αποτελείται από ένα XML έγγραφο ικανό να περιγράψει δεδομένα όπως RPC κλήσεις. Το μήνυμα αυτό μεταφέρεται μεταξύ των υπηρεσιών και των εφαρμογών, χρησιμοποιώντας κυρίως το HTTP πρωτόκολλο δικτύου. Με τον τρόπο αυτό ολοκληρώνεται το πλαίσιο λειτουργίας και επικοινωνίας στην SOA δομή, αφού με την βοήθεια της περιγραφής WSDL είναι εφικτή η επικοινωνία και συνεργασία οποιωνδήποτε υπηρεσιών στο δίκτυο. Στο παρακάτω σχήμα βλέπουμε τη χρήση του πρωτοκόλλου σε εφαρμογές είτε για προτυποποιημένη RPC επικοινωνία είτε για γενική χρήση μεταφοράς μηνύματος:



Σχήμα 3: Χρήση SOAP Πρωτοκόλλου σε εφαρμογές

Το πρωτόκολλο SOAP καθορίζει ένα XML έγγραφο με μια συγκεκριμένη δομή που μέσα του θα εμπεριέχονται οι πληροφορίες. Η βασική δομή ενός τέτοιου εγγράφου φαίνεται παρακάτω:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    ...
  </soap:Header>
  <soap:Body>
    ...
    <soap:Fault>
      ...
    </soap:Fault>
  </soap:Body>
</soap:Envelope>

```

Πίνακας 3: Η δομή ενός SOAP μηνύματος

Εσωτερικά του "σώματος" (Body) του εγγράφου, ενσωματώνεται η πληροφορία που θέλουμε να μεταφέρουμε από τη μία υπηρεσία στην άλλη. Η τεχνολογία που χρησιμοποιείται για απομακρυσμένη κλήση υπηρεσιών είναι η SOAP-RPC, (επέκταση της τεχνολογίας RPC), η οποία εισάγει στο SOAP έγγραφο τις απαιτούμενες πληροφορίες, σύμφωνα και με το WSDL της υπηρεσίας που θα κληθεί και στέλνει το μήνυμα. Επιπλέον, όπως φαίνεται και στο Σχήμα 3, εκτός από την χρήση που περιγράφηκε, το SOAP μπορεί να μεταφέρει οποιαδήποτε άλλη πληροφορία ενσωματωμένη σε XML δομή, μέσα στο "φάκελο" του SOAP μηνύματος, αρκεί βέβαια ο παραλήπτης να γνωρίζει πώς να αποκωδικοποιήσει το έγγραφο αυτό.

1.3

Σχήματα παροχής υπηρεσιών

Στις πλατφόρμες που ακολουθούν το παραπάνω σχεδιαστικό μοντέλο, μπορούμε να διακρίνουμε διάφορα ξεχωριστά σχήματα, με βάση το τι ακριβώς παρέχεται ως υπηρεσία και σε ποιον τελικό χρήστη αυτή απευθύνεται. Έτσι, διακρίνουμε τις περιπτώσεις όπου σαν υπηρεσία παρέχεται το λογισμικό μιας απλής εφαρμογής (Software as a Service), μια προγραμματιστική πλατφόρμα ή και ένα ολόκληρο λειτουργικό σύστημα στο οποίο μπορεί να δημιουργήσει ο χρήστης την εφαρμογή του (Platform as a Service), ή η υλικοτεχνική υποδομή απευθείας (Infrastructure as a Service). Οι περιπτώσεις αυτές

παρουσιάζονται παρακάτω, με πρώτη αυτήν του Software as a Service που είναι το πιο ενδιαφέρον σχήμα, στο οποίο και εστιάζει η παρούσα διατριβή.

1.3.1 Η Εφαρμογή ως Υπηρεσία

Το Software as a Service (SaaS) είναι λογισμικό που παρέχεται μέσω του διαδικτύου και είναι διαθέσιμο στον τελικό χρήστη όπως και όταν χρειάζεται. Ως εκ τούτου, θα μπορούσε να χαρακτηριστεί ως «λογισμικό κατά παραγγελία». Η χρέωση του πελάτη μπορεί να είναι είτε με βάση τη χρήση, είτε με βάση ένα μοντέλο συνδρομής, ή ακόμη και δωρεάν, αν ο πάροχος μπορεί να βγάλει κέρδος από άλλες πηγές εκτός από το χρήστη, όπως η διαφήμιση ή οι πωλήσεις λιστών πελατών.

Το Software as a Service είναι μια ραγδαία αναπτυσσόμενη αγορά, όπως αναφέρεται και σε πρόσφατες εκθέσεις που προβλέπουν έναν συνεχή διψήφιο ρυθμό ανάπτυξης (Williams, 2010), κάτι που υποδεικνύει ότι το SaaS σύντομα θα γίνει δημοφιλές στους περισσότερους οργανισμούς στο διαδίκτυο. Τα βασικά χαρακτηριστικά των SaaS συστημάτων είναι τα ακόλουθα (Kerpes, 2011):

- Παρέχουν πρόσβαση μέσω web σε εμπορικό λογισμικό.
- Όλο το λογισμικό είναι διαχειρίσιμο από ένα κεντρικό σημείο.
- Το λογισμικό παρέχεται σύμφωνα με ένα μοντέλο "ένα προς πολλούς".
- Οι τελικοί χρήστες δεν χρειάζεται να απασχολούνται με τις αναβαθμίσεις του λογισμικού και του κώδικα .
- Οι διεπαφές προγραμματισμού της εφαρμογής (Application Programming Interfaces - APIs) που παρέχονται, επιτρέπουν την ενοποίηση μεταξύ διαφορετικών κομματιών του λογισμικού.

Οι περιπτώσεις που το SaaS συνήθως χρησιμοποιείται είναι οι παρακάτω:

- Εφαρμογές που απαιτούν μεγάλη υπολογιστική ισχύ (πχ. εφαρμογές 3D rendering) ή πολύπλοκες συνδέσεις μεταξύ διαφορετικών λογισμικών και δεν είναι εύκολο να εγκατασταθούν και να συντηρηθούν τοπικά.
- Εφαρμογές στις οποίες υπάρχει σημαντική αλληλεπίδραση μεταξύ αυτού που τις φιλοξενεί και του έξω κόσμου. Για παράδειγμα, ένα λογισμικό για e-mail newsletter.
- Εφαρμογές που απαιτούν διαδικτυακή ή κινητή πρόσβαση. Ένα παράδειγμα θα ήταν ένα κινητό λογισμικό διαχείρισης πωλήσεων.
- Λογισμικά τα οποία πρέπει να χρησιμοποιηθούν μόνο για μια σύντομη ανάγκη.
- Λογισμικά όπου οι αιχμές ζήτησης αυξάνονται σημαντικά σε ορισμένες στιγμές, όπως για παράδειγμα φορολογικές υπηρεσίες ή υπηρεσίες χρέωσης που χρησιμοποιούνται μία φορά το μήνα.

Ενώ το SaaS προσφέρει πολλά πλεονεκτήματα όπως τη δυνατότητα πρόσβασης από οποιαδήποτε θέση και την ευκολία κλιμάκωσης και συντήρησης, συχνά εγείρονται ορισμένες ανησυχίες για την ασφάλεια, ειδικά για χρήστες που επιθυμούν υψηλά επίπεδα ασφαλείας και ελέγχου, κάτι που είναι βέβαια στην ευθύνη του παρόχου. Έτσι γίνεται σαφές το πόσο μεγάλη σημασία έχει η δημιουργία σχέσεων εμπιστοσύνης αλλά και η αποτίμηση της φερεγγυότητας του κάθε παρόχου, συνδυαστικά από τους διάφορους πελάτες.

1.3.2 Η Πλατφόρμα ως Υπηρεσία

Το Platform as a Service (PaaS) είναι ένας συνδυασμός μιας πλατφόρμας ανάπτυξης και μιας στοίβας λογισμικών, που προσφέρεται ως υπηρεσία κατά απαίτηση (Biswas, 2011). Παρέχει τις υποδομές επί των οποίων οι προγραμματιστές λογισμικού μπορούν να δημιουργήσουν νέες εφαρμογές ή να επεκτείνουν ήδη υπάρχουσες, χωρίς το κόστος και την επιβάρυνση που επιβάλλει η αγορά και η διαχείριση του υποκείμενου υλικού και λογισμικού και η συντήρησή τους. Με άλλα λόγια, παρέχει όλη την υποδομή υποστήριξης για να μπορέσει ο χρήστης της υπηρεσίας να αναπτύξει δικές του εφαρμογές. Τα βασικά χαρακτηριστικά που παρέχουν τα PaaS συστήματα είναι τα ακόλουθα:

- Υπηρεσίες για την ανάπτυξη, φιλοξενία, δοκιμή και συντήρηση εφαρμογών στο ίδιο διασυνδεδεμένο περιβάλλον ανάπτυξης.
- Όλες τις διαφορετικές υπηρεσίες που απαιτούνται για την εκπλήρωση της διαδικασίας ανάπτυξης εφαρμογών.
- Διαδραστικές αρχιτεκτονικές πολλαπλών χρηστών που αναπτύσσουν πολλές εφαρμογές παράλληλα.
- Επεκτάσιμο λογισμικό που μπορεί να χρησιμοποιηθεί για την εξισορρόπηση φόρτου εργασίας.
- Ενοποίηση με διαδικτυακές υπηρεσίες και βάσεις δεδομένων μέσω κοινών προτύπων.
- Υποστήριξη της συνεργασίας ομάδων ανάπτυξης λογισμικού - κάποιες PaaS πλατφόρμες περιλαμβάνουν εργαλεία σχεδιασμού έργου και επικοινωνίας.

Εκτός από τα IT τμήματα των επιχειρήσεων που το χρησιμοποιούν για να συνδιαμορφώσουν τις δικές τους εφαρμογές, οι χρήστες του PaaS περιλαμβάνουν ανεξάρτητους προμηθευτές λογισμικού (ISV), καθώς επίσης και εκείνους που αναπτύσσουν εξειδικευμένες εφαρμογές για συγκεκριμένους σκοπούς. Ενώ ως τώρα για την ανάπτυξη εφαρμογών απαιτούνταν υλικοτεχνική υποδομή, λειτουργικά συστήματα, βάσεις δεδομένων, διακομιστές Web, και άλλα λογισμικά, με το μοντέλο PaaS μόνο η γνώση για την διασύνδεσή τους είναι απαραίτητη.

1.3.3 Η Υποδομή ως Υπηρεσία

Το σχήμα "υποδομή ως υπηρεσία" (Infrastructure as a Service - IaaS) παρέχει τις υλικοτεχνικές υποδομές υπολογιστών - συνήθως ένα περιβάλλον εικονικοποίησής τους - ως υπηρεσία. Αυτό περιλαμβάνει διακομιστές, μονάδες αποθήκευσης, δικτύου αλλά και λειτουργικά συστήματα. Στο IaaS, αντί να αγοράζει διακομιστές, λογισμικό, χώρους αποθήκευσης για δεδομένα ή εξοπλισμό δικτύου, ο τελικός χρήστης ενοικιάζει αυτούς τους πόρους ως μία κατά απαίτηση υπηρεσία. Σε γενικές γραμμές, οι πλατφόρμες IaaS μπορούν να είναι δημόσιες ή ιδιωτικές υποδομές ή και συνδυασμός των δύο. Τα βασικά χαρακτηριστικά τους είναι τα παρακάτω:

- Οι πόροι της υλικοτεχνικής υποδομής παρέχονται ως υπηρεσία.
- Επιτρέπουν τη δυναμική κλιμάκωση της χρήσης των πόρων.
- Έχουν μεταβλητό κόστος, μέσω μοντέλων τιμολόγησης, ανάλογα της χρησιμότητας του κάθε πόρου.
- Σε γενικές γραμμές περιλαμβάνουν πολλαπλούς χρήστες, οι οποίοι χρησιμοποιούν από ένα μόνο μέρος του υλικού.

Το IaaS χρησιμοποιείται γενικά από οργανισμούς που έχουν την ανάλογη τεχνογνωσία, αλλά όχι και την εσωτερική υλικοτεχνική υποδομή. Έτσι, νοικιάζουν την απαιτούμενη υποδομή από τους IaaS παρόχους και φορτώνουν τις απαραίτητες βιβλιοθήκες, τις εφαρμογές και τα δεδομένα τους, τα οποία διαμορφώνουν μετά οι ίδιοι. Μια δημοφιλής χρήση του IaaS είναι για παράδειγμα η φιλοξενία ιστοσελίδων, όπου δεν είναι επιθυμητό η εσωτερική υποδομή ενός οργανισμού να επιβαρυνθεί με το έργο αυτό, αλλά προτιμάται να αφεθεί ελεύθερη να διαχειριστεί από την επιχείρηση.

Ένα σημαντικό πράγμα που πρέπει να σημειωθεί εδώ είναι ότι η γραμμή μεταξύ PaaS και IaaS γίνεται όλο και πιο θολή με την εξέλιξη των υπαρχόντων τεχνολογιών. Υπάρχει σημαντική επικάλυψη μεταξύ PaaS, SaaS, και IaaS, και με τις ραγδαίες αλλαγές που γίνονται σε αυτόν τον τομέα, οι ορισμοί είναι ακόμα υπό διαμόρφωση. Στην πραγματικότητα, η ίδια υπηρεσία μπορεί να ταξινομηθεί σε ένα από τα τρία σχήματα ανάλογα με το ποιος κάνει την κατηγοριοποίηση – ένας προγραμματιστής, ο διαχειριστής του συστήματος ή ο υπεύθυνος του οργανισμού που την παρέχει.

1.4 Καινοτομία - Συνεισφορά

Όπως προαναφέρθηκε, η παρούσα διατριβή εστιάζει στις υπηρεσιοστρεφείς αρχιτεκτονικές και πιο συγκεκριμένα στα συστήματα Software as a Service. Εφόσον το μέλλον ήδη προδιαγράφει μια ανοιχτή αγορά υπηρεσιών για εμπορικές εφαρμογές, με πολλούς παρόχους και ακόμα περισσότερους τελικούς χρήστες, όπου ο καθένας μπορεί να χρησιμοποιεί μία ή περισσότερες υπηρεσίες, εγείρονται διάφορες προκλήσεις που πρέπει να αντιμετωπιστούν όσον αφορά την ποιότητα των παρεχόμενων υπηρεσιών.

Συγκεκριμένα, οι πελάτες επιθυμούν να αξιολογήσουν την ποιότητα των παρεχόμενων υπηρεσιών, πιθανόν ανταλλάσσοντας πληροφορίες είτε μεταξύ τους είτε με

κάποιο οργανισμό κοινής εμπιστοσύνης, ώστε να τους βοηθήσει στην καλύτερη δυνατή επιλογή μεταξύ εναλλακτικών υπηρεσιών στο μέλλον και να τους σώσει από επιπλέον επιβαρύνσεις, κρατώντας τους κατά το δυνατόν ικανοποιημένους. Από την άλλη πλευρά, οι πάροχοι υπηρεσιών επιθυμούν να βελτιστοποιήσουν τη χρησιμοποίηση των πόρων τους και να κρατήσουν όσο το δυνατόν περισσότερο ικανοποιημένους τους τελικούς χρήστες των υπηρεσιών τους, με ταυτόχρονη επίτευξη του μέγιστου κέρδους. Έτσι λοιπόν, η παρούσα διατριβή στοχεύει στην αξιολόγηση και βελτιστοποίηση της ποιότητας υπηρεσίας από δύο πλευρές: από τη μεριά του παρόχου και τη μεριά του πελάτη.

Στο μέρος του πελάτη, αφού ερευνούμε διάφορα υπαρκτά συστήματα αξιολόγησης (Reputation) από τους χρήστες, παρουσιάζουμε μια καινοτόμα μεθοδολογία συνεργατικής και πολυδιάστατης βαθμολόγησης των παρόχων, με βάση κοινά κριτήρια ποιότητας υπηρεσίας από τους πελάτες, τα οποία αποτελούν και τις διαστάσεις. Η τελική αξιολόγηση κάθε παρόχου υπηρεσιών χρησιμοποιεί τις βαθμολογίες αλλά και τις νοητές συνδέσεις που υπάρχουν μεταξύ των διαφόρων τελικών χρηστών, τους οποίους και ομαδοποιεί σε γκρουπ προτιμήσεων. Χρησιμοποιώντας αυτήν την ανάλυση, μπορεί να δομηθεί ένα ολοκληρωμένο σύστημα συστάσεων (Recommendation), που προσωποποιεί την αξιολόγηση ενός νέου παρόχου για κάθε χρήστη και τον καθοδηγεί στην επιλογή μεταξύ εναλλακτικών υπηρεσιών. Παρουσιάζονται κάποια αρχικά αποτελέσματα αυτού του μηχανισμού, τροφοδοτώντας το με ένα μεγάλο σύνολο προσομοιωμένων δεδομένων, ενώ στη συνέχεια αναλύονται οι συσχετίσεις πραγματικών χρηστών και η λειτουργία του προκύπτοντος συστήματος αξιολόγησης και επιλογής υπηρεσίας επικυρώνεται πρακτικά.

Στο μέρος του παρόχου, παρουσιάζουμε έναν δυναμικό μηχανισμό μετά-δρομολόγησης εργασιών, που στοχεύει στην βελτιστοποίηση της χρησιμοποίησης των υπολογιστικών του πόρων και τη μεγιστοποίηση του κέρδους του. Ο μηχανισμός αυτός ανάγει το πρόβλημα της διαχείρισης των πόρων σε μια παραλλαγή του γνωστού προβλήματος του σακιδίου (Knapsack), θεωρώντας πολλές διαστάσεις με βάση τους πολλαπλούς υπολογιστικούς πόρους, τη χρησιμοποίηση των οποίων θέλουμε να βελτιστοποιήσουμε. Λαμβάνοντας υπόψη το ιστορικό χρήσης της υπηρεσίας, αλλά και τις δεσμεύσεις που υπάρχουν στα ηλεκτρονικά συμβόλαια με τους πελάτες, χρησιμοποιείται μια ευριστική μέθοδος συνδυασμένη με τεχνικές διαχείρισης ρίσκου για τον καθορισμό και την αξιολόγηση της ποιότητας υπηρεσίας σε κάθε περίπτωση, ώστε η βέλτιστη αξιοποίηση της υπολογιστικής υποδομής να συμπίπτει με τη μεγιστοποίηση του κέρδους του φορέα παροχής υπηρεσιών. Η πειραματική υποδομή για την αξιολόγηση του μηχανισμού αυτού χρησιμοποιεί ένα προσαρμοσμένο υπηρεσιοστρεφές μεσολογισμικό, που επιτρέπει ευελιξία διαχείρισης σε πραγματικά επιχειρηματικά σενάρια.

1.5

Οργάνωση κειμένου

Το παρόν έγγραφο αποτελείται από επτά (7) κεφάλαια, τα οποία ουσιαστικά παρουσιάζουν και αναλύουν το αντικείμενο της διδακτορικής διατριβής.

Το Κεφάλαιο 2 κάνει μια ανασκόπηση στην εξέλιξη των υπηρεσιοστρεφών συστημάτων Software as a Service, ξεκινώντας από το Διαδίκτυο και το Υπολογιστικό Πλέγμα και περνώντας στο Υπολογιστικό Νέφος και άλλες συναφείς πλατφόρμες που παρέχουν εφαρμογές ως υπηρεσίες.

Το Κεφάλαιο 3 εισάγει τον αναγνώστη στην έννοια της Ποιότητας μιας διαδικτυακής υπηρεσίας. Αρχικά αναλύεται η έννοια της Ποιότητας Υπηρεσίας σε

υπηρεσιοστρεφή περιβάλλοντα και αναφέρεται ο ορισμός των συμφωνιών σε επίπεδο υπηρεσίας (Service Level Agreements), ως η πιο σχετική μορφή ηλεκτρονικών συμβολαίων. Στη συνέχεια παρουσιάζεται το πρότυπο WS-Agreement καθώς και παραδείγματα εφαρμογών για τις οποίες αυτό χρησιμοποιείται, σε ανοικτές αγορές SaaS υπηρεσιών με τις αντίστοιχες παραμέτρους ποιότητας υψηλού και χαμηλού επιπέδου.

Το Κεφάλαιο 4 αναλύει τις υπάρχουσες μεθόδους αξιολόγησης και εξασφάλισης Ποιότητας Υπηρεσίας, διακρίνοντας δύο πλευρές που κάνουν την εν λόγω αξιολόγηση: την πλευρά του παρόχου και αυτή του καταναλωτή - πελάτη της υπηρεσίας. Στη συνέχεια παρουσιάζεται η ερευνητική πρόοδος όσον αφορά τα παραπάνω μέρη, με την αναφορά όλων των τεχνολογιών αιχμής (state of the art) και των αντίστοιχων μελετών.

Το Κεφάλαιο 5 παρουσιάζει μια καινοτόμα συνεργατική και πολυδιάστατη μέθοδο αξιολόγησης Ποιότητας Υπηρεσίας SaaS παρόχων από το μέρος των καταναλωτών καθώς και την αντίστοιχη αρχιτεκτονική υλοποίησης. Στο τέλος περιλαμβάνει εκτενή πειράματα αξιολόγησης της μεθόδου αυτής με προσομοιωμένα σύνολα δεδομένων και αντίστοιχη πρακτική επαλήθευση με πραγματικά δεδομένα.

Το Κεφάλαιο 6 επικεντρώνεται στο μέρος του παρόχου υπηρεσιών, εισάγοντας έναν μηχανισμό αξιολόγησης και διασφάλισης Ποιότητας Υπηρεσίας με δυναμική διαχείριση των πόρων του και χρήση πληροφοριών ιστορικού και ευριστικών συναρτήσεων για αποτίμηση της μελλοντικής συμπεριφοράς. Τα πειράματα εδώ κάνουν χρήση προσαρμοσμένων υπάρχοντων συστημάτων SaaS και εικονικών μηχανών.

Τέλος, στο Κεφάλαιο 7 περιλαμβάνεται η σύνοψη και τα συμπεράσματα που εξήχθησαν κατά την εκπόνηση της διδακτορικής διατριβής καθώς και θέματα μελλοντικής εργασίας και επέκτασης των ερευνητικών αποτελεσμάτων.

2

Πλατφόρμες Software as a Service

Ένα καταναμημένο περιβάλλον είναι ένα σύστημα που αποτελείται από ένα σύνολο διασυνδεδεμένων πόρων, με ένα στρώμα εικονικοποίησης που εφαρμόζεται πάνω τους για να διασφαλίσει τη διαλειτουργικότητα και την επεκτασιμότητα τους, προσφέροντας ένα μοναδικό σημείο αλληλεπίδρασης με τον πελάτη. Τα τελευταία χρόνια πυρετώδεις συζητήσεις λαμβάνουν χώρα στην ερευνητική κοινότητα σχετικά με τις δυνατότητες των διαφόρων καταναμημένων συστημάτων υπηρεσιών, ιδιαίτερα μετά την εμφάνιση του υπολογιστικού νέφους ως επιχειρηματική λύση στο διαδίκτυο, όταν το υπολογιστικό πλέγμα είχε μόλις αρχίσει να καθιερώνεται στην ίδια αγορά.

Το υπολογιστικό νέφος, όπως θα δούμε παρακάτω, επιτρέπει τη διάκριση μεταξύ του επιπέδου ανάπτυξης των εφαρμογών και των σχετικών πόρων. Επομένως, είναι δυνατό να παρέχει στον τελικό χρήστη μια πλατφόρμα για την ανάπτυξη εφαρμογών (PaaS), μια υποδομή (IaaS) ή ακόμα και την ίδια την εφαρμογή (SaaS), όλα αυτά μέσω διαδικτυακών υπηρεσιών και εύχρηστων διεπαφών. Το γεγονός ότι οι πόροι υπολογισμού και αποθήκευσης είναι εικονικοποιημένοι σαν διαδικτυακοί πόροι

διευρύνει την γκάμα των πελατών, καθιστώντας τις υπηρεσίες υπολογιστικού νέφους διαθέσιμες σε διαφόρων ειδών χρήστες. Το υπολογιστικό πλέγμα από την άλλη πλευρά, είναι περισσότερο χρήσιμο για την αντιμετώπιση των αναγκών μεγαλύτερων οργανισμών. Ο καταναλωτής των υπηρεσιών του πλέγματος συμφωνεί να χρησιμοποιήσει το υπολογιστικό πλέγμα χωρίς να έχει ιδιαίτερη δικαιοδοσία επί του ελέγχου της εφαρμογής, η οποία παρέχεται αποκλειστικά και είναι προσβάσιμη από ένα φορέα παροχής υπηρεσιών ή / και από τον προγραμματιστή της εφαρμογής.

Αν και αυτά τα χαρακτηριστικά παραδείγματα κατανεμημένων συστημάτων παρουσιάζουν διαφορές στον τρόπο που διαχειρίζονται τον έλεγχο των πόρων τους από τους πελάτες των υπηρεσιών τους και στο επίπεδο της χρηστικότητας που ο καθένας εκθέτει προς τους πελάτες, η αρχή παραμένει η ίδια: είναι και τα δύο υπηρεσιοστρεφή συστήματα τα οποία, όσον αφορά τη διαχείριση των πόρων, προσπαθούν να εικονικοποιήσουν τους υποκείμενους πόρους τους, παρέχοντας έναν ενιαίο τρόπο για την πρόσβαση και διαχείρισή τους. Έτσι, το μοντέλο Software as a Service είναι αυτό που αποτελεί την καλύτερη κοινή προσέγγιση για εφαρμογές που παρέχονται από αυτά τα συστήματα.

2.1

Υπολογιστικό πλέγμα

2.1.1 Ορισμός Πλέγματος

Το υπολογιστικό πλέγμα (Grid) είναι ένα είδος κατανεμημένης χρήσης υπολογιστικής υποδομής, που περιλαμβάνει κοινή χρήση υπολογιστή, εφαρμογής, δεδομένων, αποθήκευσης και πόρων δικτύου δυναμικά, ανάμεσα σε γεωγραφικά διασκορπισμένους οργανισμούς. Οι τεχνολογίες πλέγματος κατάφεραν να αλλάξουν τον τρόπο που οι

διάφορες επιχειρήσεις αντιμετωπίζουν και επιλύουν πολύπλοκα υπολογιστικά προβλήματα στο διαδίκτυο. Η προοπτική για κοινή χρήση πόρων έχει γίνει πραγματικότητα σε πολλούς τομείς, καθώς το υπολογιστικό πλέγμα παραμένει ακόμα μια νέα λογική με εξέλιξη ακόμη σε προτυποποίηση και τεχνολογία.

Ο όρος Grid περιλαμβάνει το σύνολο της υποδομής (υλικό και λογισμικό), καθώς και όλων των απαραίτητων διαδικτυακών υπηρεσιών (Web Services) για τη δημιουργία ενός ενιαίου (αλλά γεωγραφικά διεσπαρμένου) υπερ-υπολογιστικού περιβάλλοντος. Καθώς υπάρχουν πολλοί ορισμοί που αλληλοσυμπληρώνονται και δεν αλληλοαναιρούνται, παρακάτω αναφέρεται ο τεχνικός ορισμός του Grid έτσι όπως δίνεται από την IBM:

«Grid είναι η δυνατότητα, με τη χρήση ενός συνόλου από ανοικτά πρότυπα και πρωτόκολλα, της απόκτησης πρόσβασης σε εφαρμογές, δεδομένα, επεξεργαστική ισχύ, χώρο αποθήκευσης δεδομένων και μίας τεράστιας ποικιλίας από υπολογιστικούς πόρους που διατίθενται στο Internet. Το Grid είναι ένα είδος παράλληλου και κατανεμημένου συστήματος που δίνει τη δυνατότητα να μοιραζόμαστε, να επιλέγουμε και να συγκεντρώνουμε πόρους, που κατανέμονται σε πολλαπλά administrative domains, βασιζόμενοι στην διαθεσιμότητα των πόρων τους, τη χωρητικότητα, την επίδοση, το κόστος και τις απαιτήσεις Ποιότητας Υπηρεσίας (Quality of Service) που καθορίζονται από το χρήστη.»

Έτσι, ο όρος Grid περιλαμβάνει το σύνολο της διασυνδεδεμένης υποδομής μέσω δικτύων υψηλών ταχυτήτων, καθώς και τις απαραίτητες υπηρεσίες για τη δημιουργία ενός ενιαίου υπερ-υπολογιστικού περιβάλλοντος, που αν και κατανεμημένο, εμφανίζεται με τρόπο ενιαίο σε όλους τους χρήστες του. Αποτελεί ένα ενιαίο σύνολο υπολογιστικών

πόρων, μια συμπαγή - αλλά κατανεμημένη - υπολογιστική πλατφόρμα που διασυνδέει ετερογενή υπολογιστικά περιβάλλοντα, με όμοια ή διαφορετική φιλοσοφία και υπηρεσίες, δημιουργώντας νέα σύνολα υπηρεσιών με αυξημένες υπολογιστικές δυνατότητες και νέους τρόπους αξιοποίησης των διαφορετικών πόρων τους οποίους διαμοιράζει.

2.1.2 Χαρακτηριστικά

Ο όρος "Grid computing" καταδεικνύει ένα υπολογιστικό πρότυπο που μοιάζει με ένα πλέγμα ηλεκτρικής ισχύος αποτελούμενο από πολλούς πόρους που παράγουν ισχύ για ένα κοινό απόθεμα, που μπορούν να το χρησιμοποιούν οι πελάτες σύμφωνα με τις ανάγκες τους. Αυτό από την πλευρά της υπολογιστικής ισχύος μπορεί να περιορίζεται στις εγκαταστάσεις μιας εταιρίας, σε μια πόλη ή να είναι ευρύτερο χάρη σε ένα εκτεταμένο δίκτυο. Όμοια στην περίπτωση του υπολογιστικού πλέγματος, οι πόροι μπορούν να κατανέμονται τοπικά, είτε να είναι διασπαρμένοι παγκόσμια, εκμεταλλευόμενοι την πρόσβαση μέσω του διαδικτύου. Έτσι, γίνονται συνεχείς προσπάθειες να καθοριστούν τα πρότυπα που επιτρέπουν κοινή χρήση υπολογιστικών πόρων και αποθήκευσης δεδομένων, όσο το δυνατόν ευρύτερα και χωρίς περιορισμούς.

Τα Grids λοιπόν ενοποιούν μέσω ηλεκτρονικών δικτύων υπολογιστικούς, αποθηκευτικούς και άλλους πόρους κατανεμημένους σε τοπική, εθνική και διεθνή κλίμακα. Βάσει αυτού, διακρίνονται με τα εξής χαρακτηριστικά (Kyriazis, 2007):

- Επιτρέπουν το διαμοιρασμό των πόρων σε πολλαπλούς χρήστες διαφορετικών κοινοτήτων με ετερογενή πεδία εφαρμογών και γεωγραφική κατανομή. Ένα Grid μπορεί να στηρίζεται σε ένα τοπικό δίκτυο (campus LAN), μητροπολιτικό δίκτυο MAN, εθνικής εμβέλειας δίκτυο (WAN) ή και διεθνούς κάλυψης δίκτυο όπως το

Ευρωπαϊκό Ερευνητικό Δίκτυο GEANT και το Αμερικανικό Abilene ανάλογα με τις απαιτήσεις των εφαρμογών και τις υπάρχουσες δικτυακές υποδομές.

- Απαιτούν ασφαλή πρόσβαση μέσω ενός ενδιάμεσου λογισμικού (middleware) με έμφαση στο λογισμικό ανοικτού κώδικα - open source. Τα Grids επεκτείνουν την φιλοσοφία του ανοικτού λογισμικού σε ανοικτά υπολογιστικά συστήματα, με περιορισμούς μόνο όσο αφορά την ασφάλεια και τη διαθεσιμότητα πόρων για την κάλυψη συγκεκριμένων αναγκών.
- Παρουσιάζουν μεγάλη δυνατότητα κλιμάκωσης, με ιδιαίτερα περιορισμένη αρχική επένδυση. Οι αρχιτεκτονικές Grid μπορεί να αποτελέσουν σημαντικό εργαλείο για την υπέρβαση του ψηφιακού χάσματος στον κόσμο, σε μια ήπειρο, σε μία χώρα (κέντρο - περιφέρεια), ή σε έναν οργανισμό (campus).
- Ενοποιούν μέσω δικτύων Internet / Intranet υπολογιστικές, αποθηκευτικές και άλλες ηλεκτρονικές εγκαταστάσεις με ετερογενείς τεχνολογικές υλοποιήσεις με στόχο την παροχή ολοκληρωμένων Ηλεκτρονικών Υπηρεσιών (e-Services). Η ενοποίηση υλοποιείται με χρήση ενός επιπρόσθετου στρώματος μεσολογισμικού (middleware) που αναλαμβάνει το διαμοιρασμό των πόρων πάνω από το δίκτυο με τα παραπάνω χαρακτηριστικά.

2.1.3 Ιστορικά στοιχεία

2.1.3.1 Ανάγκη για υπολογιστική ισχύ - Από το Διαδίκτυο στο Υπολογιστικό Πλέγμα

Η προοπτική του υπολογιστικού πλέγματος άρχισε να εξερευνάται στους ακαδημαϊκούς χώρους πολλά χρόνια πριν ο επιχειρηματικός κόσμος αντιληφθεί τις δυνατότητες που θα του πρόσφερε η μετάβαση σε ένα σύστημα κατανεμημένων υπολογισμών. Πανεπιστήμια και ερευνητικά ιδρύματα είναι παραδοσιακά αυτοί που εισάγουν νέες ιδέες και

προοπτικές στα θέματα που αφορούν τις υπολογιστικές υποδομές. Αυτό ισχύει τόσο στην περίπτωση του διαδικτύου (Internet), όσο και στην περίπτωση του Παγκόσμιου ιστού (web), ο οποίος αναπτύχθηκε στο ευρωπαϊκό ερευνητικό εργαστήριο του CERN. Ωστόσο οι αρχικές ιδέες που σχετίζονταν με τον όρο Grid, απέχουν αρκετά από τις σημερινές υλοποιήσεις.

Η επιστημονική κοινότητα είναι λοιπόν αυτή, η οποία πρώτη ανακαλύπτει τα πλεονεκτήματα των νέων τεχνολογιών καθώς και τρόπους χρησιμοποίησής τους για την επίλυση προβλημάτων. Υπάρχει πληθώρα τέτοιων προβλημάτων τα οποία μπορούν να επιλυθούν πολύ καλύτερα με τη βοήθεια της προηγμένης τεχνολογίας υπολογιστών. Πολλά από αυτά τα προβλήματα έχουν το κοινό χαρακτηριστικό ότι είναι ιδιαίτερα απαιτητικά σε υπολογιστική ισχύ. Αυτό σημαίνει ότι όσο πιο μεγάλη υπολογιστική ισχύ διαθέτει κάποιος, τόσο πιο ακριβής είναι η απάντηση στο πρόβλημα του οποίου επιχειρείται η επίλυση. Ισχύει ακόμη ότι ορισμένα προβλήματα δε μπορούν καν να επιλυθούν χωρίς την επαρκή υπολογιστική δύναμη. Το θέμα των υπολογιστικά απαιτητικών εργασιών αποτελεί συνεχώς τα τελευταία 40 χρόνια την κύρια δύναμη που δίνει ώθηση στην εξέλιξη των υπολογιστικών δομών. Οι προσπάθειες για την επίλυση των προβλημάτων αυτών οδήγησαν στην ανάπτυξη αρχικά των υπολογιστών και στη συνέχεια των υπερυπολογιστών.

2.1.3.2 Οι πρόγονοι του πλέγματος

Από τους μεγάλους υπολογιστικούς πόρους της IBM της δεκαετίας του 1960 και τους υπερυπολογιστές Cray στις δεκαετίες 1970 και 1980, σταδιακά περάσαμε σε αρχιτεκτονικές συμμετρικής πολυεπεξεργασίας SMP και μαζικής παράλληλης επεξεργασίας MPP που περιλάμβαναν πολλούς επεξεργαστές, οι οποίοι είχαν τη

δυνατότητα να λειτουργούν παράλληλα. Για την αποτελεσματικότερη χρήση τους, οι προγραμματιστές εφαρμογών έπρεπε να δομήσουν των κώδικά τους έτσι ώστε να επιτρέπει την παράλληλη εκτέλεσή του. Εν συνεχεία εμφανίστηκαν οι συστάδες υπολογιστών (clusters), δηλαδή πολλοί υπολογιστές διασυνδεδεμένοι μεταξύ τους, που κατάφεραν να διαδοθούν χάρη στο χαμηλότερο κόστος τους. Τα clusters έκαναν χρήση προγραμματιστικών μοντέλων όπως το MPI (Message Passing Interface) και το PVM (Parallel Virtual Machine), κατανεμημένων συστημάτων αρχείων όπως το NFS (Network File System) και γρήγορων δικτυακών συνδέσεων όπως Ethernet και Myrinet, έτσι ώστε να αποφεύγεται η μείωση της απόδοσης λόγω αργής επικοινωνίας μεταξύ των υπολογιστών που αποτελούν κάθε συστάδα.

Ωστόσο, η αρχιτεκτονική των cluster συστημάτων αρχικά παρουσίαζε ορισμένα βασικά προβλήματα στην περίπτωση της επικοινωνίας μεταξύ clusters που χρησιμοποιούν υποδομές διαφορετικών οργανισμών (Menychtas, 2009):

- Τα clusters τυπικά αποτελούνταν από ίδιες ή παρόμοιες μηχανές. Δύο clusters διαφορετικών αρχιτεκτονικών ήταν δύσκολο να ενωθούν σε ένα ενιαίο σύστημα. Διαφορές στους compilers, τα εργαλεία, τις βιβλιοθήκες και στη δομή των αρχείων δύο διαφορετικών λειτουργικών συστημάτων αποτελούσαν προβλήματα για την ενοποίηση.
- Τυπικές παράμετροι για δικτυακές συνδέσεις μεγάλων αποστάσεων απεδείχθησαν μη ικανοποιητικές για την τεχνολογία των clusters. Καθυστέρηση, χαμηλή ρυθμοαπόδοση (throughput), τυχαίες συμφορήσεις του δικτύου (τυπικό για το TCP/IP) και σφάλματα δικτύου ήταν κάποιοι από τους παράγοντες που συνέβαλαν σε αυτό.

- Δεν υπήρχε η απαιτούμενη τεχνολογία για να αντιμετωπιστούν τα προβλήματα ασφαλείας που προέκυπταν. Συναρτήσεις με χαμηλή ασφάλεια, που ήταν κατάλληλες για εσωτερικά συστήματα που συνήθως προστατεύονταν από firewalls, ήταν τελείως ακατάλληλες για περιβάλλον ανοικτού δικτύου.
- Θέματα διαχείρισης και πολιτικής έδωσαν επίσης μια νέα διάσταση στο πρόβλημα. Οι διάφοροι οργανισμοί έπρεπε να καθορίσουν τις συνθήκες κάτω από τις οποίες θα μοιράζονταν τους πόρους τους μέσω του δικτύου. Ακόμη όμως και αν καθοριζόταν μια τέτοια πολιτική, με την υπάρχουσα τεχνολογία δεν υπήρχαν τα απαραίτητα τεχνικά μέσα για τον αποτελεσματικό έλεγχο και επιβολή της.

Στα επόμενα χρόνια, η σχετική ερευνητική δραστηριότητα συνεχίστηκε, με αποτέλεσμα την δημιουργία του *NSFNET* το 1986, ενός δικτύου στα 56Kbps που συνέδεε τα πέντε NSF κέντρα υπερ-υπολογιστών και λίγο αργότερα, τη δημιουργία του Condor [1988] από το πανεπιστήμιο του Wisconsin. Το σύστημα αυτό (Condor High Throuput Computing, 2012) είναι ένας "διαχειριστής φόρτου εργασίας" (workload manager), με δυνατότητες παρακολούθησης και διαχείρισης πόρων, δρομολόγησης εργασιών και αποτελεί το πρώτο πρόγραμμα με κατεύθυνση προς τη λογική των υπηρεσιών πλέγματος.

2.1.3.3 Η εξέλιξη και καθιέρωση του πλέγματος

Τα εν συνεχεία εμφανιζόμενα δικτύων υψηλών ταχυτήτων και η προκύπτουσα ανάγκη για μεγάλη επεξεργαστική ισχύ οδήγησε σε έντονη ερευνητική δραστηριότητα με αποτελέσματα τα προγράμματα Legion [1993], SRB [1997], Globus [1998]. Το πρώτο βασίζεται στην ιδέα του "εικονικού υπολογιστή" (virtual computer): όλοι οι πόροι, συνδεδεμένοι μεταξύ τους, εμφανίζονται στον χρήστη ως μία εικονική μηχανή, με

αρκετά μειονεκτήματα όμως, όπως η πολύπλοκη υλοποίηση και η μικρή αποδοτικότητα. Το SRB (Storage Resource Broker) ήταν μια πλατφόρμα διαχείρισης αποθηκευτικών πόρων που βοήθησε πολύ στην ανάπτυξη των Grid τεχνολογιών, αφού αντιμετώπισε τα προβλήματα μεταφοράς δεδομένων σε ένα περιβάλλον πλέγματος. Το Globus είναι πλέον μια από τις πιο γνωστές πλατφόρμες Grid, και παρουσιάζεται αναλυτικά στο τέλος της ενότητας, στις πλατφόρμες ανοικτού κώδικα. Τέλος, με την ανάπτυξη των Web Services [2001] που παρουσιάστηκαν παραπάνω, και την εγκαθίδρυση της Αρχιτεκτονικής Ανοιχτών Υπηρεσιών Grid (Open Grid Service Architecture - OGSA) [2002] ως κύριας αρχιτεκτονικής της Grid τεχνολογίας, οι τεχνολογίες πλέγματος καθιερώθηκαν ως ένα χρησιμότερο εργαλείο στον τομέα των υπολογισμών υψηλής απόδοσης (High Performance Computing - HPC).

Το γεγονός ότι οι πρώτες υπηρεσίες υστερούσαν σε αξιοπιστία, ασφάλεια και απόδοση, όπως επίσης και σε λειτουργικότητα υψηλότερου επιπέδου, είχε ως αποτέλεσμα την εμφάνιση διαφόρων τεχνολογιών που έλυναν τα προβλήματα αυτά. Έτσι οδηγηθήκαμε στην εμφάνιση προτύπων όπως του WS-Security και του WSRF (Web Services Resource Framework). Το τελευταίο χρησιμοποιείται για τη δημιουργία υπηρεσιών διαδικτύου, οι οποίες προσφέρουν στους μετέχοντες στο Grid πρόσβαση σε πόρους. Μία υπηρεσία δικτύου μπορεί να αντιστοιχεί σε πολλούς πόρους (WS-Resources), ενώ ένας πόρος μπορεί να είναι προσπελάσιμος από διάφορες διαδικτυακές υπηρεσίες. Οι πόροι αυτοί είναι πλέον δυναμικοί, δηλαδή οι ιδιότητές τους μπορούν να αλλάζουν. Μπορούν να δημιουργούνται και να καταστρέφονται κατ' απαίτηση και η πρόσβαση σε αυτούς είναι δυνατή μόνο μέσω μιας τέτοιας υπηρεσίας.

2.1.3.4 Οι Γενιές των Συστημάτων Πλέγματος

Η πρώτη γενιά συστημάτων πλέγματος (1st Generation Grids ή *1G Grids*) ουσιαστικά αποτελούνταν από τοπικούς στοιχειώδεις "μετα-υπολογιστές" (metacomputers) (Catlett, 2004). Οι υποδομές αυτές υποστήριζαν βασικές λειτουργίες όπως το κατανεμημένο σύστημα αρχείων, πάνω στο οποίο χτίστηκαν νέες κατανεμημένες εφαρμογές με ειδικά προσαρμοσμένα δικτυακά πρωτόκολλα και το "sitewide single sign-on", δηλαδή ένα μοναδικό σημείο όπου ο χρήστης δίνει τα προσωπικά στοιχεία του (username / password) για όλα τα κατανεμημένα αυτά συστήματα. Οι υποδομές σταδιακά επεκτάθηκαν και έγινε προσπάθεια δημιουργίας "μετα-κέντρων" (metacenters), τα οποία διερεύνησαν θέματα ολοκλήρωσης μεταξύ διαφορετικών κέντρων. Τα Grids πρώτης γενιάς, ωστόσο, ήταν εντελώς προσαρμοσμένα σε συγκεκριμένα πειράματα και απλά αποτέλεσαν απόδειξη της ιδέας (proof-of-concept).

Η δεύτερη γενιά συστημάτων πλέγματος (*2G Grids*) εμφανίστηκε με προγράμματα όπως το Condor, το I-WAY (που αποτέλεσε την αρχή του Globus) και το Legion (που αναφέρθηκε παραπάνω και αποτέλεσε την αρχή του Anaki), όπου νέες υπηρεσίες ενδιάμεσου λογισμικού και πρωτοκόλλων επικοινωνιών αποτέλεσαν τη βάση για την ανάπτυξη κατανεμημένων εφαρμογών και υπηρεσιών. Τα Grids 2ης γενιάς ουσιαστικά έδωσαν τα βασικά δομικά στοιχεία, αλλά η χρήση τους απαιτούσε σημαντική προσπάθεια "customization", ώστε να καλυφθούν σημαντικά κενά. Οι ανεξάρτητες αυτές προσπάθειες χρήσης συστημάτων 2ης γενιάς που περιείχαν πολλές "κατά απαίτηση" επεκτάσεις λογισμικού, κατέστησε την διαλειτουργικότητά τους προβληματική, κάτι που ήρθε να λύσει η επόμενη γενιά.

Αξιοποιώντας την πρότερη εμπειρία τόσο από τις 2 πρώτες γενιές, όσο και τις τεχνολογίες των διαδικτυακών υπηρεσιών, ξεκίνησαν οι προσπάθειες για την 3η γενιά Grids (*3G Grids*), που βασίστηκαν στην Αρχιτεκτονική Ανοιχτών Υπηρεσιών Πλέγματος, που θα παρουσιασθεί παρακάτω. Το πρότυπο αυτό είναι το πιο σημαντικό για τις τεχνολογίες του Grid, αφού περιγράφει τις δυνατότητες του συστήματος αυτού να αναλύσει τη λειτουργία του σε όλα τα επίπεδά του. Στην Αρχιτεκτονική Ανοιχτών Υπηρεσιών Grid, μια σειρά από προδιαγραφές κοινών και ανοιχτών διεπαφών υποστηρίζουν τη διαλειτουργικότητα, ανεξάρτητα των ανεπτυγμένων υπηρεσιών. Η εκδοθείσα προδιαγραφή Open Grid Services Infrastructure (OGSI) είναι ο θεμέλιος λίθος της αρχιτεκτονικής. Με την εισαγωγή προτυποποιημένων τεχνικών προδιαγραφών, η 3η γενιά Grid επιτάχυνε τον ανταγωνισμό και την επίτευξη διαλειτουργικότητας όχι μόνο μεταξύ εφαρμογών και εργαλείων αλλά, κυρίως, μεταξύ διαφορετικών υλοποιήσεων βασικών υπηρεσιών του πλέγματος.

2.1.4 Παράμετροι Σχεδίασης Πλέγματος

Κατά τη διαδικασία σχεδιασμού ενός πλέγματος, όπως και άλλων καταναμημένων συστημάτων, πρέπει να ληφθούν υπόψη ορισμένες παράμετροι, που διαδραματίζουν σημαντικότατο ρόλο στο σχεδιασμό του Grid συστήματος. Οι κύριες παράμετροι σχεδίασης ενός πλέγματος είναι οι παρακάτω:

- **Απαιτήσεις ασφάλειας:** Τα δεδομένα και οι πόροι του συστήματος πρέπει να προστατεύονται από καταστροφή, κλοπή και γενικά από ενέργειες που στοχεύουν να βλάψουν. Τα τείχη προστασίας (firewalls) υπήρξαν μια πρώτη προσέγγιση για αυτό το πρόβλημα, η οποία όμως υστερούσε σε ευελιξία και μείωνε τη συνολική απόδοση του συστήματος. Με την εμφάνιση των προτύπων WS-Security και WS-

Agreement που αναφέρθηκαν στην εισαγωγή, γίνεται χρήση πιστοποιητικών, και επιβάλλεται η διαπραγμάτευση για την πραγματοποίηση της επικοινωνίας και κλήσης μιας υπηρεσίας του πλέγματος.

- **Ευαισθησία δεδομένων:** Αν τα διαχειριζόμενα δεδομένα είναι ευαίσθητα, τότε πέραν των ανωτέρω λύσεων, προκρίνεται ο καθορισμός ασφαλών καναλιών μεταφοράς δεδομένων με τη χρήση των προτύπων SSL (Secure Socket Layer) και κρυπτογράφησης κατά την επικοινωνία.
- **Μέγιστες απαιτήσεις σε επεξεργαστική ισχύ:** Με την εμφάνιση του πλέγματος κατέστη δυνατό κάθε επιχείρηση να χρησιμοποιεί την πλεονάζουσα επεξεργαστική ισχύ για πολλαπλές εφαρμογές, όπως επίσης και να διαθέτει επεξεργαστική ισχύ σε τρίτους αποκομίζοντας κέρδος. Επιπλέον, έχει τη δυνατότητα να μην αγοράσει επεξεργαστική ισχύ που θα ανταποκρίνεται στις μέγιστες απαιτήσεις της, και μόνο τις ώρες που οι απαιτήσεις φτάνουν στο μέγιστο να καταφεύγει σε προσφορές επεξεργαστικής ισχύς από τρίτους, για να καλύψει τις βραχυπρόθεσμες απαιτήσεις της.
- **Αποθήκευση δεδομένων:** Στην επίπεδη αρχιτεκτονική του πλέγματος όλοι οι κόμβοι είναι ισοδύναμοι, συνεπώς η αποθήκευση των δεδομένων ακολουθεί την ίδια λογική και άρα οποιοδήποτε δεδομένο μπορεί να τοποθετηθεί οπουδήποτε. Το τι θα αποθηκευτεί σε μια συγκεκριμένη τοποθεσία καθορίζεται από το σύστημα αποθήκευσης, τη διαθέσιμη χωρητικότητα, το ρυθμό δεδομένων, το κόστος αποθήκευσης, το χρόνο ζωής των δεδομένων και φυσικά την πολιτική. Η χωρητικότητα είναι ο συνολικός χώρος ενός συστήματος αποθήκευσης και θα πρέπει να ανταποκρίνεται στις μέγιστες απαιτήσεις για αποθηκευτικό χώρο. Ο

χρόνος απόκρισης συνήθως καθορίζει το μέσο που θα χρησιμοποιηθεί ανάλογα αν απαιτείται μικρός χρόνος ή όχι. Η ρυθμοαπόδοση μπορεί να αυξηθεί χρησιμοποιώντας πολλά στιγμιότυπα του ζητούμενου αντικειμένου, έτσι ώστε να επιτευχθεί αθροιστικά ο ρυθμός παράδοσης των δεδομένων. Ο χρόνος ζωής των δεδομένων, τέλος, ευνοεί συνήθως τα συστήματα αποθήκευσης σε ταινίες. Τα πρωτόκολλα που χρησιμοποιούνται εδώ και που διαχειρίζονται την αποθήκευση των δεδομένων συνήθως λειτουργούν με σχετικά αδιαφανή τρόπο.

- **Διαθέσιμο εύρος ζώνης στο διαδίκτυο:** Εφόσον το πλέγμα στηρίζεται στο διαδίκτυο, το διαθέσιμο εύρος ζώνης (bandwidth) που διαθέτει, αποτελεί σημαντικότατο παράγοντα. Όταν κάποιος αγοράζει πρόσβαση σε πόρους, πρέπει να υπάρχει αρκετό εύρος ζώνης, για να προωθηθούν οι εργασίες του μέσω του διαδικτύου. Μια εφαρμογή που τρέχει πάνω σε Grid θα πρέπει να μεταφέρει μαζί της το περιβάλλον λειτουργίας της, γεγονός που, στη περίπτωση που τρέχουν πολλές μικρές εφαρμογές μαζί, καθιστά το μέγεθος των μεταφερόμενων δεδομένων (εφαρμογή, βιβλιοθήκες, αρχεία) αρκετά μεγάλο. Μάλιστα, στην περίπτωση που τα αποτελέσματα μιας έστω και μικρής εφαρμογής παράγουν μεγάλο όγκο αποτελεσμάτων, τα πράγματα χειροτερεύουν. Σε κάθε περίπτωση, ο σχεδιασμός της αρχιτεκτονικής δεδομένων εδώ πρέπει να λαμβάνει υπόψη τα είδη των συνδέσεων των διαφόρων χρηστών, που βρίσκονται γεωγραφικά και την αναμενόμενη κίνηση στο πλέγμα, ενώ θα πρέπει συχνά να γίνεται και ενδιάμεση προσωρινή αποθήκευση επιπλέον αντιγράφων των δεδομένων (caching).
- **Υπάρχοντες πόροι:** Μια επίσης σημαντική παράμετρος για το πλέγμα είναι η δημιουργία ενός είδους ευρετηρίου πόρων καθώς και της ισχύος τους, ώστε να

διευκολύνεται η δυνατότητα επαναχρησιμοποίησης των υπαρχόντων πόρων που ειδάλλως είναι ανενεργοί για μεγάλο διάστημα. Τέτοιοι πόροι είναι υπολογιστές γραφείου (συνήθως διαθέσιμοι τις νυχτερινές ώρες, υψηλό κόστος συντήρησης ανά μηχανήμα), clusters (συνήθως διαθέσιμα 24 ώρες το εικοσιτετράωρο, αξιόπιστα και ασφαλή), ή συστήματα συμμετρικής πολυεπεξεργασίας που μένουν αναξιοποίητα για μεγάλα χρονικά διαστήματα. Έτσι, με τη χρήση του κατάλληλου συστήματος χρονοπρογραμματιστή (scheduler) μπορούν να αντιστοιχιστούν οι εργασίες στους πόρους που είναι καταλληλότεροι για την εκτέλεσή τους και είναι ανενεργοί την τρέχουσα στιγμή της αίτησης.

- **Πόροι ειδικού σκοπού:** Πόροι οι οποίοι δημιουργήθηκαν ειδικά για την εκτέλεση μιας εργασίας ή για μια κατηγορία από εργασίες μπορεί να αποτελούνται από ειδικά μηχανήματα, όπως συσκευές μετρήσεων, ειδικού σκοπού λογισμικό και ειδικού σκοπού υλικό. Παραδοσιακά αυτοί οι πόροι θα γίνονταν προσβάσιμοι μέσω ειδικών πρωτοκόλλων ή ειδικά διαμορφωμένων δικτύων. Με την έλευση του Grid computing εμφανίστηκαν διάφορα εργαλεία, που επιτρέπουν τη δημιουργία νέων υπηρεσιών κατάλληλων για σύνδεση με αυτούς τους πόρους. Βέβαια τα εργαλεία αυτά δεν έχουν φτάσει στο επίπεδο ωριμότητας που απαιτείται, ώστε να είναι δυνατό αυτό για όλους τους πόρους. Έτσι μπορούν να χρησιμοποιηθούν resource wrappers ή minihosting περιβάλλοντα, επιτρέποντας την πρόσβαση στον πόρο μέσω των υπαρχόντων ιδιόκτητων διασυνδέσεων ή και ενδιάμεσες υπηρεσίες Grid που προωθούν μέσω ειδικού μηχανισμού τις απαραίτητες πληροφορίες στον πόρο.

- **Μεταφερσιμότητα:** Μια παράμετρος στο σχεδιασμό του πλέγματος είναι η μετατροπή εφαρμογών προηγούμενων συστημάτων έτσι ώστε πλέον να λειτουργούν και να αξιοποιούν το Grid περιβάλλον. Η εύκολη λύση είναι να γραφτεί για κάθε τέτοια εφαρμογή ένας αντίστοιχος Grid wrapper κώδικας, ο οποίος θα επιτρέπει τη λειτουργία της στο νέο περιβάλλον και θα αξιοποιεί τα ιδιαίτερα χαρακτηριστικά της. Παρόλα αυτά όμως, για να αξιοποιήσει η εφαρμογή καλύτερα το Grid σύστημα, είναι καλύτερο να προσαρμόζεται ο ίδιος ο κώδικας της ξαναγράφοντας μέρος του. Έτσι, για απαιτήσεις όπως η ασφάλεια, χρειάζεται νέος σχεδιασμός, ενώ απαιτήσεις όπως η απόδοση γίνονται λιγότερο περιοριστικές στο περιβάλλον του Grid. Ορισμένες υπάρχουσες εφαρμογές μπορούν να εκτελεστούν σε ένα περιβάλλον Grid με τη βοήθεια διεπαφών, που κάνουν χρήση ήδη καθορισμένων προτύπων. Υπάρχουν όμως και άλλες, η μεταφορά των οποίων δεν είναι δυνατή ακόμη και με την παραπάνω μέθοδο, οπότε και χρειάζεται να γραφτεί μια ειδική διεπαφή, βασισμένη στην τεχνολογία πλέγματος.
- **Δυνατότητες συνεργασίας:** Τέλος, μια άλλη σημαντική παράμετρος είναι η δυνατότητα που προσφέρεται στους χρήστες για συνεργασία με άλλους χρήστες και οντότητες του συστήματος. Είναι λοιπόν σημαντικό να υπάρχει η δυνατότητα καθορισμού πολιτικών διαπραγμάτευσης για την επίτευξη της συνεργασίας, με πρωτόκολλα και κανονισμούς, οι οποίοι εξασφαλίζουν την ασφάλεια της όλης διαδικασίας, καθώς και η δυνατότητα δημιουργίας δυναμικών εικονικών οργανισμών (Virtual Organizations - VOs) ως δομών εμπιστοσύνης και δυναμικής διαχείρισής τους. Η έννοια του εικονικού οργανισμού εξετάζεται

περαιτέρω στη συνέχεια της διατριβής, στο κεφάλαιο αξιολόγησης της Ποιότητας Υπηρεσίας από τον πελάτη.

2.1.5 Αρχιτεκτονική Ανοιχτών Υπηρεσιών Πλέγματος

Μετά τα πρώτα χρόνια δημιουργίας και ωρίμασης των τεχνολογιών πλέγματος, αναζητήθηκαν τρόποι να ευθυγραμμιστεί το Grid computing με τις τεχνολογίες διαδικτυακών υπηρεσιών (Foster, Kesselman, Nick, & Tuecke, 2002). Αυτό έγινε επιθυμητό για να αξιοποιηθούν κάποιες σημαντικές ιδιότητες των Web Services: η περιγραφή και η ανακάλυψη υπηρεσίας, η αυτόματη παραγωγή κώδικα του πελάτη και του διακομιστή από τις περιγραφές των υπηρεσιών, η δέσμευση των περιγραφών υπηρεσιών σε διαλειτουργικά πρωτόκολλα δικτύου, η συμβατότητα με τα αναδυόμενα ανοικτά πρότυπα υπηρεσιών υψηλότερου επιπέδου, οι υπάρχουσες υπηρεσίες και τα εργαλεία και, φυσικά, η ευρεία εμπορική υποστήριξη. Αυτή η ευθυγράμμιση και ο συνδυασμός των τεχνολογιών πλέγματος και των διαδικτυακών υπηρεσιών οδήγησε στην Αρχιτεκτονική Ανοιχτών Υπηρεσιών Πλέγματος (Open Grid Services Architecture - OGSA).

Ο όρος *Αρχιτεκτονική* στο OGSA υποδηλώνει μια καλά καθορισμένη δέσμη βασικών διεπαφών από τις οποίες μπορούν να κατασκευαστούν ενδιαφέροντα συστήματα, και το *Ανοιχτών* χρησιμοποιείται για να δείξει την επεκτασιμότητα, την ουδετερότητα των κατασκευαστών πλέγματος και τη δέσμευσή τους σε μια κοινή διαδικασία τυποποίησης. Η αρχιτεκτονική αυτή χρησιμοποιεί τη γλώσσα περιγραφής υπηρεσιών διαδικτύου (WSDL), για τη δημιουργία αυτο-περιγραφόμενων υπηρεσιών και διαλειτουργικών πρωτοκόλλων, με επεκτάσεις που επιτρέπουν την υποστήριξη πολλαπλών συντονιζόμενων διεπαφών και τη διαχείριση αλλαγών.

Το OGSA αξιοποίησε την υπάρχουσα εμπειρία πάνω στον καθορισμό WSDL περιγραφών για μια Grid υπηρεσία, όπου ένα (ενδεχομένως παροδικό) στιγμιότυπο υπηρεσίας υποστηρίζει αξιόπιστες και ασφαλείς κλήσεις (όταν απαιτείται), στη διαχείριση κύκλου ζωής, στη διαχείριση των πολιτικών του παρόχου πλέγματος, στη διαχείριση διαπιστευτηρίων (credentials), αλλά και στην εικονικοποίηση (virtualization). Το OGSA ορίζει επίσης διεπαφές για την ανακάλυψη υπηρεσιών πλέγματος και τη δημιουργία εφήμερων στιγμιότυπων Grid υπηρεσιών. Το αποτέλεσμα είναι ένα κατανεμημένο σύστημα παροχής υπηρεσιών βασισμένο σε πρότυπα, που υποστηρίζει τη δημιουργία εξελιγμένων υπηρεσιών που απαιτούνται στις σύγχρονες επιχειρήσεις και τα υπολογιστικά περιβάλλοντα οργανισμών.

2.1.5.1 Απαιτήσεις Αρχιτεκτονικής OGSA

Οι απαιτήσεις που ανέκυψαν για τα περιβάλλοντα πλέγματος ήταν και αυτές που καθόρισαν σε μεγάλο βαθμό τα χαρακτηριστικά της αρχιτεκτονικής OGSA. Οι απαιτήσεις αυτές παρουσιάζονται παρακάτω (Foster, et al., 2005).

2.1.5.1.1 Διαλειτουργικότητα και υποστήριξη δυναμικών και ετερογενών περιβαλλόντων

Σε γενικές γραμμές, τα περιβάλλοντα πλέγματος τείνουν να είναι ετερογενή και κατανεμημένα, φιλοξενούμενα σε μια ποικιλία από πλατφόρμες (π.χ. J2EE, .NET), λειτουργικά συστήματα (π.χ. Unix, Linux, Windows, ενσωματωμένα συστήματα), συσκευές (π.χ. υπολογιστές, εργαλεία, αισθητήρες, συστήματα αποθήκευσης, βάσεις δεδομένων, δίκτυα) και υπηρεσίες, που παρέχονται από διάφορους κατασκευαστές. Επιπλέον, τα περιβάλλοντα αυτά συχνά προορίζονται να είναι μακράς διάρκειας και δυναμικά, και ως εκ τούτου μπορούν να εξελιχθούν με τρόπους που δεν είχε αρχικά προβλεφθεί. Το OGSA πρέπει να επιτρέπει τη διαλειτουργικότητα μεταξύ αυτών των

διαφορετικών, ετερογενών και κατανεμημένων πόρων και υπηρεσιών, καθώς και να μειώνει την πολυπλοκότητα διαχείρισης των ετερογενών συστημάτων.

2.1.5.1.2 Διαμοιρασμός των πόρων στους οργανισμούς

Το πλέγμα δεν είναι ένα μονολιθικό σύστημα, αλλά συχνά αποτελείται από πόρους που ανήκουν και ελέγχονται από διάφορους οργανισμούς. Ένας βασικός σκοπός του OGSA είναι να υποστηρίξει το διαμοιρασμό και την αξιοποίηση των πόρων σε διαχειριστικούς τομείς, είτε διαφορετικές μονάδες εργασίας εντός μιας επιχείρησης. Έξυπνοι μηχανισμοί απαιτούνται για να παρέχουν ένα πλαίσιο που μπορεί να υποστηρίξει τους συνεργαζόμενους χρήστες, τα αιτήματα, τους πόρους, τις πολιτικές και τις συμφωνίες με βάση οργανωτικά όρια.

2.1.5.1.3 Βελτιστοποίηση

Η απαίτηση αυτή αναφέρεται σε τεχνικές κατανομής των πόρων για την ικανοποίηση των καταναλωτών και των απαιτήσεων του παρόχου. Η βελτιστοποίηση εφαρμόζεται τόσο για τους παρόχους υπηρεσιών πλέγματος (server side) όσο και για τους καταναλωτές (client side) πόρων και υπηρεσιών. Μια συνηθισμένη περίπτωση server side βελτιστοποίησης είναι η βελτιστοποίηση χρήσης των πόρων, που θα αναλυθεί διεξοδικά στο κεφάλαιο αξιολόγησης από την πλευρά του παρόχου. Η βελτιστοποίηση με βάση τη ζήτηση απαιτεί τη διαχείριση διαφόρων ειδών φόρτου εργασίας, συμπεριλαμβανομένων των απαιτήσεων του συνολικού φόρτου εργασίας, οι οποίες μπορεί να είναι δύσκολο να προβλεφθούν.

2.1.5.1.4 Διασφάλιση Ποιότητας Υπηρεσίας

Υπηρεσίες όπως η εκτέλεση εργασιών και οι υπηρεσίες δεδομένων πρέπει να παρέχουν τη συμφωνημένη ποιότητα (η έννοια της ποιότητας παρουσιάζεται αναλυτικότερα στο αντίστοιχο κεφάλαιο). Βασικές διαστάσεις ποιότητας περιλαμβάνουν (χωρίς να περιορίζονται σε αυτές) τη διαθεσιμότητα, την ασφάλεια και την απόδοση. Οι προσδοκίες ποιότητας θα πρέπει να εκφράζονται με μετρήσιμους όρους σε ηλεκτρονικά συμβόλαια, όπως θα δούμε παρακάτω.

2.1.5.1.5 Εκτέλεση εργασιών

Το OGSA πρέπει να παρέχει δυνατότητα διαχείρισης της εκτέλεσης εργασιών που ορίζονται από το χρήστη σε όλη τη διάρκεια ζωής τους. Λειτουργίες όπως ο χρονοπρογραμματισμός, ο έλεγχος και ο χειρισμός σφαλμάτων στο χρόνο εκτέλεσης των εργασιών πρέπει να υποστηρίζεται, ακόμη και όταν η εργασία κατανέμεται σε μεγάλο αριθμό ετερογενών πόρων.

2.1.5.1.6 Υπηρεσίες δεδομένων

Μια βασική απαίτηση είναι η αποτελεσματική πρόσβαση και διακίνηση τεράστιων ποσοτήτων δεδομένων, που είναι χρήσιμη σε όλο και περισσότερα πεδία της επιστήμης και της τεχνολογίας. Επιπλέον, ο διαφανής διαμοιρασμός δεδομένων είναι σημαντικός, όπως για παράδειγμα η επίτρεψη πρόσβασης σε πληροφορίες που είναι αποθηκευμένες σε βάσεις δεδομένων οι οποίες είναι ανεξάρτητα διαχειρίσιμες. Στον επιχειρηματικό κόσμο, η αρχειοθέτηση και η διαχείριση των δεδομένων είναι οι βασικότερες απαιτήσεις.

2.1.5.1.7 Ασφάλεια

Η ασφαλής διαχείριση απαιτεί έλεγχο της πρόσβασης στις υπηρεσίες μέσω ισχυρών πρωτοκόλλων ασφαλείας και σύμφωνα με την αντίστοιχη πολιτική. Για παράδειγμα, η απόκτηση των εφαρμογών και η παροχή τους μέσω ενός Grid συστήματος μπορεί να απαιτεί έλεγχο ταυτότητας και εξουσιοδότηση. Επίσης, η ανταλλαγή πόρων από τους χρήστες απαιτεί κάποιου είδους μηχανισμού απομόνωσής τους. Επιπροσθέτως, απαιτούνται πρότυποι και ασφαλείς μηχανισμοί για την προστασία των συστημάτων πλέγματος και την υποστήριξη του ασφαλούς διαμοιρασμού των πόρων μεταξύ των διαχειριστικών περιοχών.

2.1.5.1.8 Μείωση των διαχειριστικών δαπανών

Η πολυπλοκότητα της διαχείρισης κατανεμημένων ετερογενών συστημάτων μεγάλης κλίμακας αυξάνει τα έξοδα διαχείρισης και τον κίνδυνο του ανθρώπινου λάθους. Έτσι, γίνεται σαφές ότι απαιτείται υποστήριξη για δυνατότητες διαχείρισης εικονικοποιημένων πόρων, καθώς και αυτοματοποίηση των διαχειριστικών εργασιών.

2.1.5.1.9 Δυνατότητες Κλιμάκωσης

Ένα περιβάλλον πλέγματος μεγάλης κλίμακας μπορεί να προσφέρει προστιθέμενη αξία, όπως τη δραστική μείωση του χρόνου αναμονής των εργασιών στην ουρά εκτέλεσης, εκμεταλλευόμενο τη χρήση μεγάλου αριθμού πόρων, επιτρέποντας έτσι νέες υπηρεσίες. Ωστόσο, η μεγάλη κλιμάκωση του συστήματος μπορεί να παρουσιάσει προβλήματα, δεδομένου ότι δημιουργεί νέες απαιτήσεις για τη διαχείριση των υποδομών. Έτσι, η αρχιτεκτονική διαχείρισης πρέπει να μπορεί δυνητικά να προσαρμόζεται σε χιλιάδες διαθέσιμους πόρους ποικίλης φύσης.

2.1.5.1.10 Διαθεσιμότητα

Η υψηλή διαθεσιμότητα γίνεται δυνατή συνήθως σε συστήματα ανεκτικά σε σφάλματα ή σύνθετα συστήματα συστάδων. Λόγω της ευρείας χρήσης των πληροφοριακών συστημάτων για την παροχή βασικών δημόσιων υποδομών και υπηρεσιών, ένας αυξανόμενος αριθμός συστημάτων απαιτούνται να λειτουργήσουν σε υψηλά επίπεδα διαθεσιμότητας. Εφόσον οι τεχνολογίες πλέγματος επιτρέπουν τη διαφανή πρόσβαση σε ένα ευρύτερο σύνολο πόρων, μπορούν να χρησιμοποιηθούν ως δομικό στοιχείο για την επίτευξη σταθερών και αξιόπιστων συστημάτων εκτέλεσης εργασιών. Όμως, λόγω της ανομοιογένειας των συστατικών του πλέγματος, κάποια από τα οποία μπορεί να είναι αναξιόπιστα, παρουσιάζονται προβλήματα διαθεσιμότητας. Συνεπώς, απαιτούνται μηχανισμοί αποκατάστασης αποτυχιών και διαχείρισης σφαλμάτων, έτσι ώστε η ορθή λειτουργία ενός συστήματος πλέγματος να μπορεί να ανακτηθεί γρήγορα σε ειδικές περιπτώσεις αποτυχίας ή ασυνεπειών.

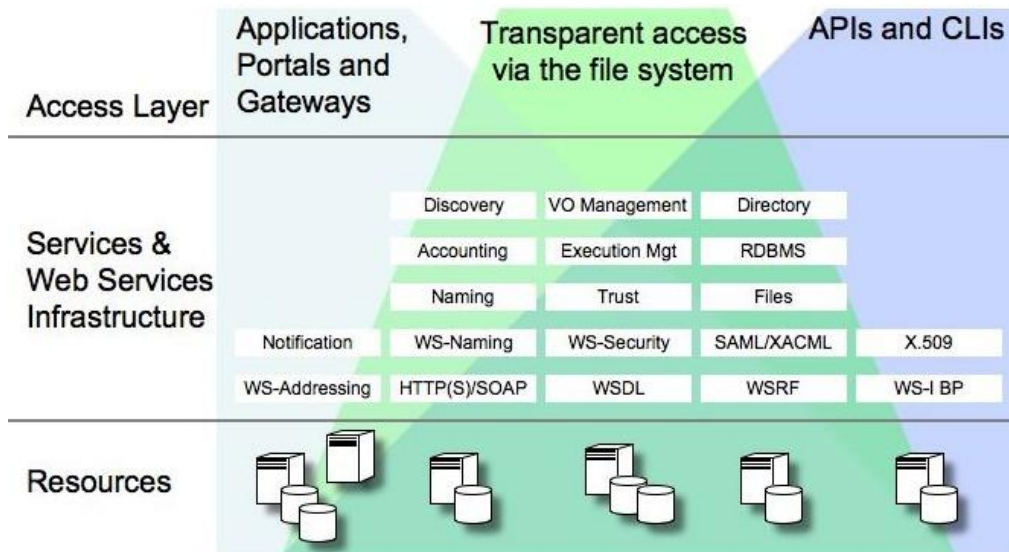
2.1.5.1.11 Ευκολία Χρήσης και Επεκτασιμότητα

Ο χρήστης θα πρέπει να είναι σε θέση να χρησιμοποιήσει το OGSA "καλύπτοντας" την πολυπλοκότητα του περιβάλλοντος, όταν αυτό απαιτείται. Τα εργαλεία της αρχιτεκτονικής θα πρέπει όσο το δυνατόν περισσότερο να μπορούν να διαχειριστούν το περιβάλλον αντί για το χρήστη και να παρέχουν αφαιρετικότητα στο επιθυμητό επίπεδο. Τέλος, δεν είναι δυνατόν να προβλέψουμε τις πολλές και ποικίλες ανάγκες που θα έχουν οι χρήστες του συστήματος. Ως εκ τούτου, οι μηχανισμοί και οι πολιτικές πρέπει να υλοποιηθούν μέσω επεκτάσιμων και αντικαταστάσιμων συστατικών στοιχείων λογισμικού, προκειμένου να επιτραπεί στο OGSA να εξελίσσεται με την πάροδο του

χρόνου και να επιτρέπει στους χρήστες να κατασκευάσουν τους δικούς τους μηχανισμούς και τις πολιτικές που ανταποκρίνονται σε συγκεκριμένες ανάγκες.

2.1.5.2 Επίπεδα της Αρχιτεκτονικής

Το OGSA είναι το πρότυπο μιας κλασικής αρχιτεκτονική ενδιάμεσου λογισμικού σχεδιασμένη έτσι ώστε να ταιριάζει σε παραδοσιακά μοντέλα κατακευματισμένων συστημάτων τριών επιπέδων , όπως φαίνεται στο σχήμα που ακολουθεί (OGSA Primer, 2008):



Σχήμα 4: Αρχιτεκτονική τριών επιπέδων του OGSA

Τα επίπεδα που διακρίνονται είναι τα εξής:

- **Επίπεδο πρόσβασης:** Το επίπεδο πρόσβασης είναι το υψηλότερο της αρχιτεκτονικής OGSA. Το επίπεδο αυτό παρέχει κατάλληλες διεπαφές στους χρήστες ώστε να αλληλεπιδρούν με το πλέγμα μέσω μιας εφαρμογής ειδικού σκοπού, μιας διαδικτυακής πύλης, ενός συνόλου γραφικών διεπαφών ή και μέσω ενσωμάτωσης των διεπαφών αυτών στο τοπικό λειτουργικό σύστημα που

φιλοξενεί το πλέγμα. Με το πέρασμα του χρόνου και όσο αναπτύσσονται υπηρεσίες πλέγματος στο χαμηλότερο επίπεδο, καινούργιες εφαρμογές και διεπαφές θα κάνουν την εμφάνισή τους, χρησιμοποιώντας μια ή και περισσότερες Grid υπηρεσίες.

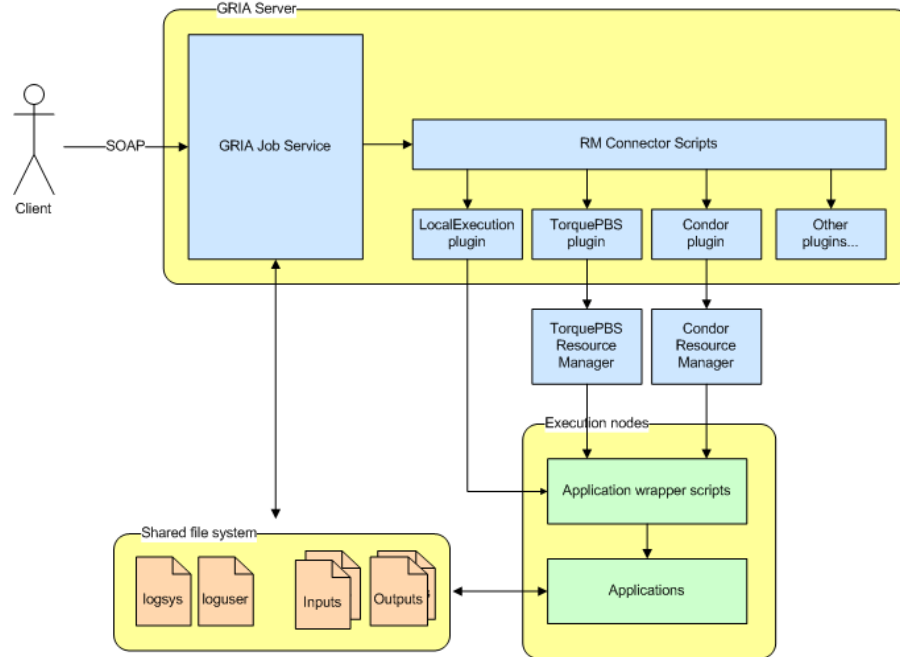
- **Επίπεδο υπηρεσιών:** Το επίπεδο υπηρεσιών παρέχει ένα τυποποιημένο σύνολο διεπαφών, πάνω στο οποίο βασίζεται η υλοποίηση του επιπέδου πρόσβασης και είναι ουσιαστικά ο πυρήνας του OGSA. Μία πολύ βασική θεώρηση του OGSA είναι ότι όλοι οι πόροι του συστήματος αντιστοιχίζονται με υπηρεσίες. Έτσι η υποδομή OGS (Open Grid Services Infrastructure) χρησιμοποιεί συγκεκριμένες, γνωστές υπηρεσίες δικτύου και πρωτόκολλα όπως τα XML και WSDL για να δημιουργήσει επαφές και διασυνδέσεις κάθε Grid υπηρεσίας με τους πόρους του συστήματος.
- **Επίπεδο πόρων:** Το επίπεδο πόρων είναι το πιο χαμηλό επίπεδο από τα τρία και παρέχει τους φυσικούς ή εικονικούς πόρους του πλέγματος. Ως πόρος (resource) σε ένα Grid μπορεί να θεωρηθούν διάφορα στοιχεία, από τον επεξεργαστή ενός υπολογιστή, μέχρι και το τμήμα υπολογιστών μιας μεγάλης εταιρίας. Επίσης εκτός από υπολογιστικές μονάδες, συμμετέχουν και μονάδες αποθήκευσης, βάσεις δεδομένων, δικτυακές υποδομές κλπ. Αυτοί είναι κυρίως οι φυσικοί πόροι. Εκτός από αυτούς έχουμε και τους εικονικούς, πχ. όταν κάποια ενδιάμεσα συστήματα (middleware) χρησιμοποιώντας τους φυσικούς πόρους προσφέρουν στοιχειώδεις υπηρεσίες ή συνενωμένους φυσικούς πόρους στο παραπάνω επίπεδο.

2.1.6 Πλατφόρμες Ανοικτού Κώδικα

2.1.6.1 GRIA

Το λογισμικό GRIA (Grid Resources for Industrial Applications) είναι ένα υπηρεσιοστρεφές μεσολογισμικό πλέγματος, σχεδιασμένο για να υποστηρίζει συνεργασίες επιχείρησης με επιχείρηση (Business-to-Business - B2B), με την παροχή υπηρεσιών σε ένα ασφαλές, διαλειτουργικό και ευέλικτο πλαίσιο (Aisopos, Tserpes, Panousopoulos, Phillips, Kardara, & Varvarigou, 2009). Το GRIA χρησιμοποιεί πρωτόκολλα διαδικτυακών υπηρεσιών σύμφωνα με βασικές προδιαγραφές διαλειτουργικότητας και είναι ανοιχτού κώδικα (το μεγαλύτερο μέρος του λογισμικού είναι LGPL). Με τη χρήση επιχειρηματικών μοντέλων και σημασιολογιών επιτρέπεται στους παρόχους υπηρεσιών και τους χρήστες να ανακαλύψουν ο ένας τον άλλον και να διαπραγματευτούν τους όρους για την πρόσβαση σε εφαρμογές οι οποίες είναι διαθέσιμες ως υπηρεσίες GRIA (Software as a Service).

Το GRIA παρέχει διάφορα πακέτα λογισμικού, όλα σχεδιασμένα για την αντιμετώπιση συγκεκριμένων αναγκών του επιχειρηματικού κόσμου, που αποτελεί και το βασικό χρήστη του. Παρέχει μια πλήρη εργαλειοθήκη για τη διαχείριση των υπηρεσιών πλέγματος σε επίπεδο εφαρμογής αλλά και πόρων, χρησιμοποιώντας πρωτόκολλα διαχείρισης συμφωνιών και χρέωσης των πελατών, με ταυτόχρονη εφαρμογή περιορισμών στους πόρους. Η βασική αρχιτεκτονική του, εστιάζοντας στο κομμάτι της διαχείρισης πόρων το οποίο και θα μας απασχολήσει ιδιαίτερα παρακάτω, φαίνεται στο Σχήμα 5 (GRIA: Service Oriented Collaborations for Industry and Commerce project, 2012):



Σχήμα 5: Αρχιτεκτονική GRIA

Όπως φαίνεται στο σχήμα, το GRIA διαθέτει μια ειδική υπηρεσία (Job Service) για την εκτέλεση διαφόρων εργασιών από τους πελάτες που ζητούν μια συγκεκριμένη εφαρμογή (Application). Η υπηρεσία αυτή αναθέτει την εκτέλεση στους διάφορους διαχειριστές πόρων (Resource Managers - RMs) που εισάγονται στο σύστημα με τη μορφή plugin, και δρομολογούν την εργασία είτε προς τοπική εκτέλεση (με χρήση του LocalExecution plugin), είτε προς κατανεμημένη εκτέλεση (TorquePBS, Condor plugins), με κλήση τοπικών script που λειτουργούν ως Application wrappers.

2.1.6.2 Globus

Το πιο γνωστό και διαδεδομένο σύστημα διαχείρισης υπηρεσιών πλέγματος είναι το GLOBUS, που αναπτύχθηκε στο Argonne National Lab στο πανεπιστήμιο του Berkeley. Το GLOBUS προτυποποίησε πρωτοκόλλα για τη ασφάλεια, μεταφορά δεδομένων, ανακάλυψη πόρων και εκτέλεση εργασιών. Η αρχιτεκτονική του GLOBUS λειτουργεί σε

τέσσερα λογικά επίπεδα: το επίπεδο Εφαρμογών, το επίπεδο Υπηρεσιών υψηλού Επιπέδου και Εργαλείων, το επίπεδο Υπηρεσιών Πυρήνα και το επίπεδο Τοπικών Υπηρεσιών. Τις βασικές Υπηρεσίες Πυρήνα αποτελούν οι υπηρεσίες:

- Globus Security Interface (**GSI**)
- Metacomputing Directory Service (**MDS**)
- Globus Resource Access Manager (**GRAM**)

Η τελευταία (GRAM) είναι και αυτή που διαχειρίζεται την πρόσβαση στους πόρους, χρησιμοποιώντας διάφορους τοπικούς Resource Managers (LSF, Condor, NQE) όπως και στη λογική του GRIA που είδαμε παραπάνω.

2.2 Το Υπολογιστικό Νέφος

2.2.1 Από το υπολογιστικό πλέγμα στο υπολογιστικό νέφος

Από τις αρχές αυτής της δεκαετίας, οι κατανεμημένες αρχιτεκτονικές που χτίστηκαν κάτω από διαδικτυακές υπηρεσίες του Παγκοσμίου Ιστού, όπως η Αρχιτεκτονική Ανοικτών Υπηρεσιών Πλέγματος (OGSA), εισήγαγαν μια νέα γενιά συστημάτων που έχουν κατασκευαστεί ως ένα σύνολο στοιχειωδών υπηρεσιών, ενδεχομένως από διαφορετικούς παρόχους, διασυνδεδεμένων μεταξύ τους προκειμένου να προσφέρουν μια ενιαία υπηρεσία στους πελάτες. Τα περιβάλλοντα πλέγματος ξεκίνησαν σαν μία πολλά υποσχόμενη τεχνολογία η οποία ενσωμάτωνε ετερογενείς υπολογιστικούς πόρους μεγάλης κλίμακας σε ένα κοινό σύστημα διαχείρισης. Οι πόροι αυτοί πιθανόν να άνηκαν στον ίδιο ή σε διαφορετικούς παρόχους (και αντίστοιχες γεωγραφικές τοποθεσίες) και η διαχείρισή τους επιτυγχανόταν μέσω των υπηρεσιοστρεφών αρχιτεκτονικών και υλοποιήσεων.

Ωστόσο, η διαφημιστική εκστρατεία γύρω από το υπολογιστικό πλέγμα απέτυχε να δημιουργήσει μια εκτεταμένη νέα γενιά βιώσιμων εμπορικών παρόχων Grid Computing. Ένα βασικό μειονέκτημα των υπηρεσιών πλεγμάτων ήταν η έλλειψη πλήρους απομόνωσης μεταξύ των χρηστών, που δημιουργούσε προβλήματα στην εγγύηση της παρεχόμενης υπηρεσίας, την ασφάλεια αλλά κυρίως τις διαφορετικές απαιτήσεις του περιβάλλοντος κάθε εφαρμογής. Έτσι, δύο εφαρμογές με διαφορετικές ανάγκες π.χ. σε λειτουργικό σύστημα, βιβλιοθήκες ή εκδόσεις λογισμικού δεν μπορούσαν να εκτελούνται στον ίδιο φυσικό πόρο.

Παρόλα αυτά, η έννοια των κεντρικών εγκαταστάσεων με υπολογιστικές και αποθηκευτικές υποδομές που λειτουργούν από τρίτους εντάχθηκε σε ένα νέο πρότυπο υπολογιστικών συστημάτων, το υπολογιστικό νέφος (Cloud Computing) (Foster, Yong Zhao Raicu, & Lu, 2008). Το Cloud Computing περιλαμβάνει παροχή μέσω διαδικτύου όχι μόνο δυναμικά κλιμακωτών πόρων, αλλά και αφαιρετικών και πλήρως εικονικοποιημένων. Μέσω της τεχνικής της εικονικοποίησης, κάθε χρήστης είναι εντελώς απομονωμένος από τους υπόλοιπους και μπορεί να χειρίζεται κατά το δοκούν τη διαμόρφωση του περιβάλλοντος εκτέλεσης των εφαρμογών του. Έτσι, άτομα ή επιχειρήσεις μπορούν να έχουν πρόσβαση σε ελαστικούς υπολογιστικούς πόρους ή λογισμικό με πληρωμή ανάλογα με τη χρήση, χωρίς την ανάγκη κεφαλαιακής επένδυσης εξαρχής ή μετέπειτα συντήρησης, αλλά και με περισσότερη ασφάλεια, όλα αυτά συνήθως κατόπιν αντίστοιχων ηλεκτρονικών συμφωνητικών.

2.2.2 Ορισμός Νέφους

Το υπολογιστικό νέφος αναφέρεται σε εφαρμογές και υποδομές που προσφέρονται ως υπηρεσίες μέσω διαδικτύου, από διάφορα κέντρα δεδομένων σε όλο τον κόσμο, που

συλλογικά αναφέρονται ως το "νέφος". Η ιδέα του νέφους απλοποιεί τις πολλές συνδέσεις δικτύου και συστημάτων υπολογιστών που συμμετέχουν στις online υπηρεσίες, παρέχοντας αφαιρετικά τους πόρους ως πλήρως εικονικοποιημένους, παρά το γεγονός ότι μπορεί να ανήκουν σε διαφορετικές υποδομές/τοποθεσίες. Κάθε χρήστης με σύνδεση στο διαδίκτυο τυπικά μπορεί να έχει πρόσβαση σε ένα δημόσια δημοσιευμένο υπολογιστικό νέφος και τις υπηρεσίες που παρέχει. Δεδομένου ωστόσο ότι οι εν λόγω υπηρεσίες συχνά συνδέονται, οι χρήστες μπορούν να μοιράζονται πληροφορίες μεταξύ πολλών συστημάτων και με άλλους χρήστες.

Ο τεχνικός ορισμός του Cloud από τον οργανισμό NIST (National Institute of Standards and Technology), δίνεται παρακάτω (Mell & Grance, 2011):

«Το Υπολογιστικό Νέφος είναι ένα μοντέλο για τη πραγματοποίηση εύκολης, ανεμπόδιστης και κατά παραγγελία πρόσβασης μέσω δικτύου σε μια κοινή ομάδα διαμορφώσιμων πόρων υπολογισμού (π.χ., δίκτυα, κεντρικοί υπολογιστές, αποθηκευτικοί χώροι, εφαρμογές, και υπηρεσίες) που μπορούν να είναι διαθέσιμοι γρήγορα και με την ελάχιστη διαχειριστική προσπάθεια ή αλληλεπίδραση φορέων παροχής υπηρεσιών.»

Όπως φαίνεται και από τον πιο πάνω ορισμό, το υπολογιστικό νέφος έχει αρκετές ομοιότητες με το υπολογιστικό πλέγμα, του οποίου μπορεί να θεωρηθεί ως εξέλιξη.

2.2.3 Χαρακτηριστικά

Δεδομένου ότι έχουν αναλυθεί ήδη διεξοδικά τα χαρακτηριστικά του πλέγματος στις προηγούμενες παραγράφους, κάνουμε εδώ μια σύντομη αναφορά στα βασικά χαρακτηριστικά ενός υπολογιστικού νέφους:

- Αυτο-εξυπηρέτηση κατά απαίτηση: Δυνατότητες μονομερούς καθορισμού από τον καταναλωτή των υπολογιστικών πόρων που χρειάζεται, όπως για παράδειγμα

- του χρόνου εξυπηρέτησης και του χώρου αποθήκευσης, αυτόματα χωρίς ανθρώπινη αλληλεπίδραση με τον κάθε πάροχο υπηρεσίας.
- Πρόσβαση μέσω διαδικτύου: Οι δυνατότητες αυτές παρέχονται μέσω δικτύου και μπορούν να προσπελαστούν μέσω τυποποιημένων μηχανισμών που προωθούν τη χρήση από ετερογενείς πλατφόρμες πελάτη (π.χ., κινητά τηλέφωνα, tablets, φορητούς υπολογιστές και σταθμούς εργασίας).
 - Συγκέντρωση πόρων: Οι υπολογιστικοί πόροι του παρόχου συγκεντρώνονται για να εξυπηρετήσουν πολλούς καταναλωτές, με τη χρήση ενός μοντέλου πολλαπλών ενοικιαστών, με διαφορετικούς φυσικούς και εικονικούς πόρους που ανατίθενται και ανακατανέμονται σε αυτούς δυναμικά με βάση τις ανάγκες τους. Έτσι προσδίδεται μια αίσθηση ανεξαρτησίας θέσεων, δεδομένου ότι ο πελάτης δεν έχει συνήθως κανένα έλεγχο ή γνώση για την ακριβή τοποθεσία των παρεχόμενων πόρων (αποθήκευσης, επεξεργασίας, μνήμης ή εύρους ζώνης δικτύου), αλλά μπορεί να είναι σε θέση να προσδιορίζει τη θέση σε ένα υψηλότερο πιο αφαιρετικό επίπεδο (π.χ. χώρα, περιοχή, ή κέντρο δεδομένων).
 - Ταχεία ελαστικότητα: Οι πόροι που χρησιμοποιεί και απελευθερώνει ο καταναλωτής μπορούν να μεταβάλλονται ελαστικά και γρήγορα, σε ορισμένες περιπτώσεις αυτόματα, ώστε να ακολουθούν τη ζήτηση και τις ανάγκες του. Για τον καταναλωτή, η παροχή πόρων συχνά φαίνεται να είναι απεριόριστη και μπορεί να μεταβληθεί σε οποιαδήποτε ποσότητα ανά πάσα στιγμή.
 - Καταγραφή χρησιμοποίησης πόρων: Τα συστήματα υπολογιστικών νεφών ελέγχουν και βελτιστοποιούν τη χρήση των πόρων αυτόματα, με υπηρεσίες καταγραφής χρησιμοποίησης σε ένα αφαιρετικό επίπεδο ανάλογα με το είδος της

υπηρεσίας (π.χ. επίπεδο αποθήκευσης, επεξεργασίας, εύρους ζώνης, ή ενεργών λογαριασμών χρηστών). Η χρήση των πόρων παρακολουθείται και ελέγχεται, παρέχοντας διαφάνεια για τον πάροχο, αλλά και για τον καταναλωτή της υπηρεσίας που χρησιμοποιείται.

2.2.4 Μοντέλα Ανάπτυξης και Παροχής

Ανάλογα με τον τρόπο που ένα περιβάλλον νέφους αναπτύσσεται και παρέχονται οι υπηρεσίες του, διακρίνουμε τα παρακάτω μοντέλα:

- **Private Cloud:** Η υποδομή του υπολογιστικού νέφους παρέχεται για αποκλειστική χρήση από ένα και μόνο οργανισμό που αποτελείται από πολλούς καταναλωτές (π.χ. επιχειρηματικές μονάδες). Μπορεί να ανήκει και να είναι διαχειρίσιμο από τον ίδιο τον οργανισμό, από έναν τρίτο, ή από κάποιο συνδυασμό αυτών, και μπορεί να αναπτύσσεται μέσα στις εγκαταστάσεις του οργανισμού.
- **Community Cloud:** Η υποδομή του υπολογιστικού νέφους παρέχεται για αποκλειστική χρήση από μια συγκεκριμένη κοινότητα καταναλωτών από οργανισμούς με κοινές ανησυχίες (π.χ. ίδια αποστολή, ίδιες απαιτήσεις ασφάλειας, πολιτικές και ανάγκες συμμόρφωσης με κανονισμούς). Μπορεί να ανήκει και να είναι διαχειρίσιμο από έναν ή περισσότερους από τους οργανισμούς της κοινότητας, από έναν τρίτο, ή από κάποιο συνδυασμό αυτών, και μπορεί να αναπτύσσεται μέσα στις εγκαταστάσεις των οργανισμών.
- **Public Cloud:** Η υποδομή του υπολογιστικού νέφους παρέχεται για ανοικτή χρήση από το ευρύ κοινό. Μπορεί να ανήκει και να είναι διαχειρίσιμο από επιχειρηματικούς, ακαδημαϊκούς, ή κυβερνητικούς οργανισμούς, ή από κάποιο συνδυασμό αυτών. Αναπτύσσεται στις εγκαταστάσεις του Cloud παρόχου.

- **Hybrid Cloud:** Η υποδομή του υπολογιστικού νέφους είναι ένας συνδυασμός από δύο ή περισσότερες διακριτές υποδομές νέφους (Private, Community ή Public) που παραμένουν ξεχωριστές οντότητες, αλλά συνδέονται μεταξύ τους με τυποποιημένες τεχνολογίες που επιτρέπουν τη φορητότητα των δεδομένων και των εφαρμογών (π.χ. cloud bursting για εξισορρόπηση φορτίου μεταξύ νεφών).

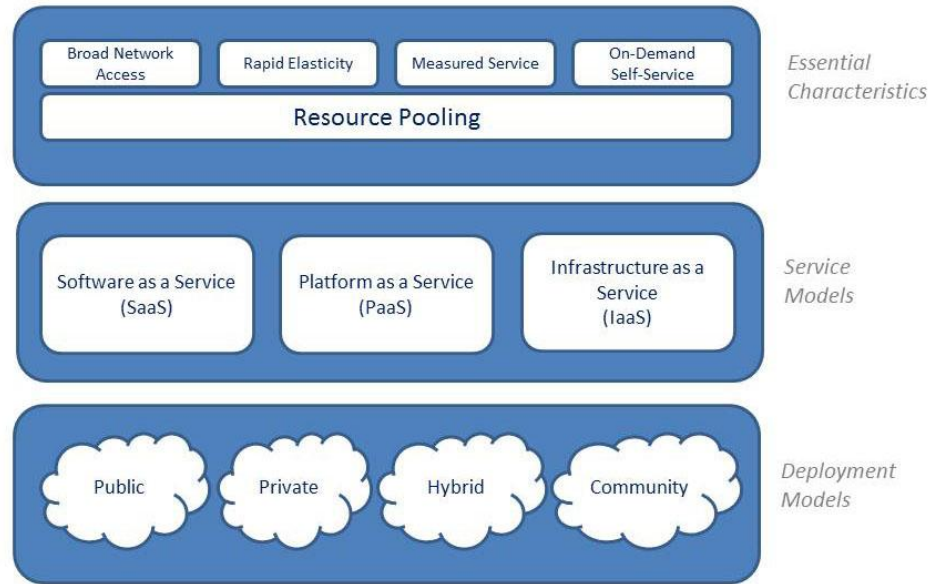
2.2.5 Το πλαίσιο SPI

Ένα συμφωνηθέν κοινό πλαίσιο για την περιγραφή υπηρεσιών του υπολογιστικού νέφους είναι το SPI (Software Platform Infrastructure), το οποίο περιγράφει τα τρία βασικά σχήματα παροχής μέσα από το νέφος: Software as a Service, Platform as a Service και Infrastructure as a Service. Τα υπολογιστικά νέφη είναι σε θέση να υλοποιούν οποιοδήποτε από τα αυτά τρία σχήματα παροχής υπηρεσιών, ανάλογα με τον προσανατολισμό του παρόχου και τις ανάγκες των καταναλωτών των υπηρεσιών Cloud, παρέχοντας τις αντίστοιχες εγγυήσεις λειτουργίας. Κάποια χαρακτηριστικά παραδείγματα γνωστών υπηρεσιών που ακολουθούν τις τρεις περιπτώσεις είναι τα παρακάτω (Picarello, 2011):

- **Software as a Service:** Google Docs, Zoho Docs
- **Platform as a Service:** Google App Engine
- **Infrastructure as a Service:** Amazon EC2, S3

Τα σχήματα αυτά αναλύθηκαν στην εισαγωγή της διατριβής και, όπως αναφέρθηκε, εμείς θα εστιάσουμε σε θέματα διαχείρισης και επιλογής του μοντέλου SaaS.

Έτσι, συμπερασματικά, το συνολικό πλαίσιο λειτουργίας ενός υπολογιστικού νέφους με βάση τα διαθέσιμα σχήματα παροχής, τα μοντέλα ανάπτυξης και τα βασικά χαρακτηριστικά που αναφέρθηκαν, φαίνεται στο Σχήμα 6:



Σχήμα 6: Πλαίσιο λειτουργίας υπολογιστικού νέφους

Ένα σημαντικό θέμα που πρέπει να αναφερθεί τέλος σε σχέση με το πλαίσιο SPI, είναι ότι παρά τα τεράστια οφέλη της τεχνολογίας του Cloud Computing, κάποια από τα παραπάνω σχήματα παροχής, ιδιαίτερα στα Public Clouds, θέτουν αρκετά προβλήματα προς εξέταση στους τομείς της ακεραιότητας των δεδομένων αλλά και της ασφάλειας, ζητήματα που αντιμετωπίζονται με συγκεκριμένες προσεγγίσεις από τα διάφορα περιβάλλοντα ανοικτού κώδικα.

2.2.6 Πλατφόρμες Ανοικτού Κώδικα

2.2.6.1 OpenNebula

Το OpenNebula Cloud ήταν προϊόν ενός ερευνητικού έργου που στοχεύει στην ανάπτυξη προτύπων λύσεων στη βιομηχανία για την κατασκευή και διαχείριση εικονικοποιημένων κέντρων δεδομένων και υποδομών των επιχειρήσεων με τη μορφή ενός υπολογιστικού νέφους. Το OpenNebula είναι ένα κατακευματισμένο σύστημα διαχείρισης εικονικών μηχανών που επιτρέπει την πλήρη εικονικοποίηση μιας υποδομής. Επίσης, προσφέρει τη

δυνατότητα αναπόσπαστης διαχείρισης των εικονικών υπηρεσιών του υπολογιστικού του νέφους, συμπεριλαμβανομένου του φόρτου δικτύου, και της διαχείρισης των εικονικών περιβαλλόντων.

Το OpenNebula διαθέτει plugins για το Amazon Elastic Compute Cloud (Amazon EC2) που επιτρέπουν την ταυτόχρονη ανάπτυξη εικονικών μηχανών σε μια τοπική υποδομή αλλά και στο Amazon EC2. Η βασική αρχιτεκτονική του αποτελείται από δύο επίπεδα: ένα επίπεδο υπηρεσιών και ένα επίπεδο υποδομών. Σαν λογισμικό αποτελείται από διάφορα κομμάτια (modules) με σχεδιασμό που επιτρέπει διαλειτουργικότητα, κάνοντας δυνατή τη σύνδεση με άλλες πλατφόρμες (πχ. με το Nimbus), αλλά και την προσθήκη διεπαφών (πχ. για το Amazon EC2).

2.2.6.2 *Nimbus Cloud*

Το Nimbus είναι ένα από τα πιο γνωστά εργαλεία ανοικτού κώδικα που παρέχει μια υποδομή νέφους στην επιστημονική κοινότητα, το οποίο ξεκίνησε χρησιμοποιώντας ένα μέρος του Globus toolkit (Globus core), που παρουσιάστηκε παραπάνω, για την επεξεργασία των αιτήσεων πελατών και την πιστοποίηση των χρηστών. Το Nimbus μπορεί να χρησιμοποιηθεί αυτόνομα ή και σε συνδυασμό με υπηρεσίες Globus, για τη διαχείριση μιας εικονικής συστάδας υπολογιστών.

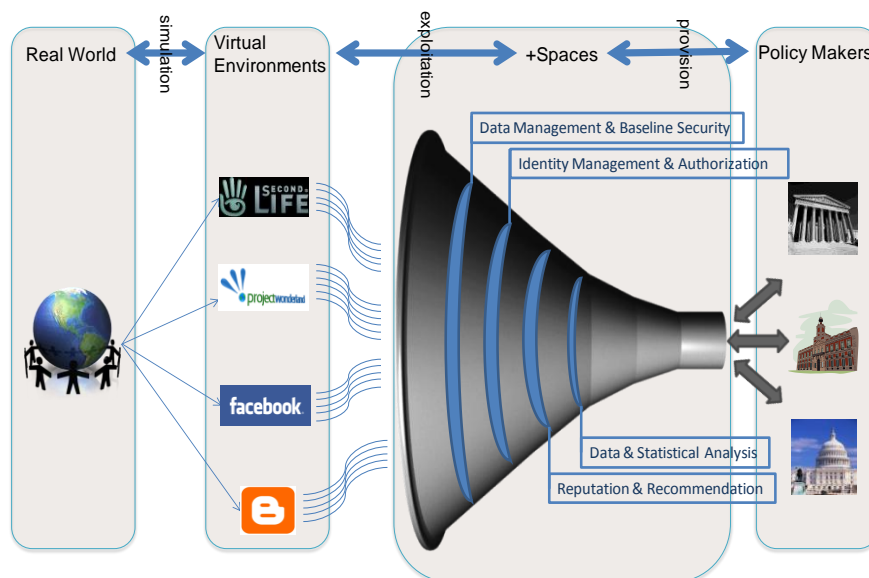
Το Nimbus πλέον σήμερα επικεντρώνεται περισσότερο στην παροχή Υποδομής σαν Υπηρεσία (IaaS) (Nimbus: cloud computing for science, 2012). Ωστόσο, διάφορες εμπορικές παραλλαγές του (για παράδειγμα IPRO Nimbus, Nimbus Blue) επικεντρώνονται ακόμα στο Software as a Service, για εξειδικευμένες εφαρμογές (πχ. e-Capture, e-Review).

2.3

Πλατφόρμες με ανοιχτές αγορές εφαρμογών ως υπηρεσίες

2.3.1 Positive Spaces

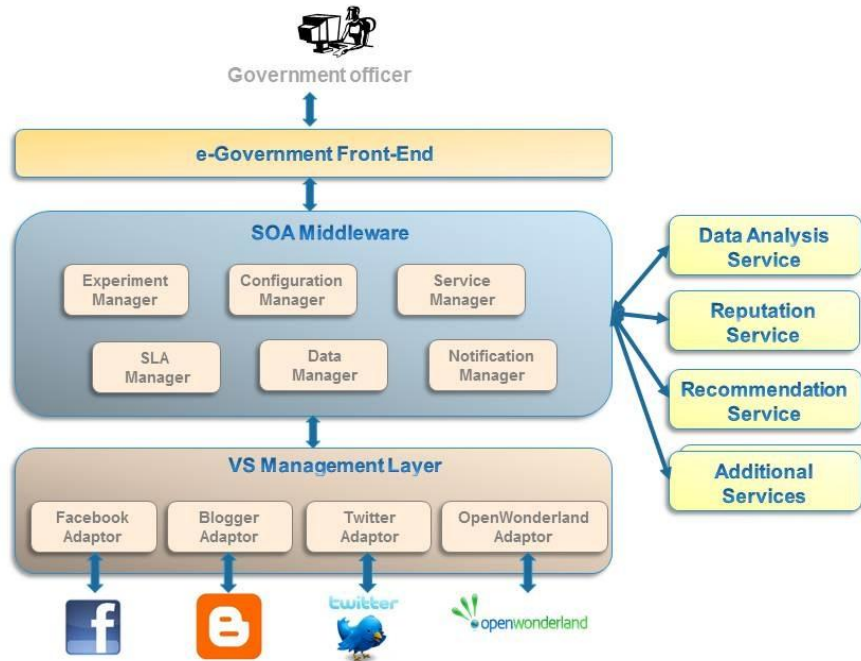
Η πλατφόρμα του ευρωπαϊκού έργου Positive Spaces (Policy Simulation in Virtual Spaces - +Spaces) έχει ως στόχο να παρέχει στους υπεύθυνους χάραξης πολιτικών τα μέσα για έλεγχο των διαφόρων νομοθετικών πλαισίων με e-Government εφαρμογές σε εικονικά περιβάλλοντα (Virtual Environments) τριών διαστάσεων, σε συνδυασμό με υπηρεσίες συγκέντρωσης και ανάλυσης δεδομένων. Τα δεδομένα που προέρχονται από τα περιβάλλοντα αυτά, παρουσιάζονται εν συνεχεία με κατάλληλο τρόπο στον τελικό χρήστη ώστε να συναχθούν συμπεράσματα για τις επιπτώσεις των πολιτικών που δοκιμάζονται (Kardara, Fuchs, Kosta, Aisopos, Tserpes, Varvarigou, 2012). Το έργο στοχεύει επίσης στην αύξηση της συμμετοχής στη διαδικασία αυτή συμπεριλαμβάνοντας στα εικονικά περιβάλλοντα αυτά διάφορους ιστοτόπους κοινωνικής δικτύωσης.



Σχήμα 7: Ροή δεδομένων στο +Spaces

Στο Σχήμα 7, φαίνεται μια επισκόπηση της ροής των δεδομένων στο +Spaces. Το τελευταίο στάδιο πριν τα δεδομένα παραδοθούν στον τελικό χρήστη είναι οι λειτουργίες αξιολόγησης και σύστασης (Reputation & Recommendation), καθώς και η λειτουργία στατιστικής ανάλυσης δεδομένων (Data & Statistical Analysis). Αυτές περιλαμβάνουν εργασίες όπως την αξιολόγηση των χρηστών στα εικονικά περιβάλλοντα που συμμετέχουν στις εφαρμογές, την παροχή συστάσεων όσον αφορά τις εφαρμογές αυτές, αλλά και τη διενέργεια συσχετίσεων των δεδομένων και δημιουργίας γραφημάτων, ώστε να παρουσιασθούν σαφώς τα αποτελέσματα στον τελικό χρήστη. Οι λειτουργίες αυτές προσθέτουν «νοημοσύνη» στην πλατφόρμα, με την επεξεργασία των δεδομένων από τα εικονικά περιβάλλοντα προκειμένου να οδηγήσουν σε συμπεράσματα για τις διάφορες πολιτικές, και παρέχονται ως επί πληρωμή υπηρεσίες ανάλυσης (Kardara, Fuchs, Kosta, Aisopos, Tserpes, & Varvarigou, 2012). Έτσι διαμορφώνεται μια ανοικτή αγορά SaaS υπηρεσιών, όπου νέες υπηρεσίες ανάλυσης δεδομένων μπορούν να προστεθούν και να γίνουν διαθέσιμες στο χρήστη της πλατφόρμας (Aisopos, et al., 2012).

Η βασική αρχιτεκτονική της πλατφόρμας φαίνεται παρακάτω στο Σχήμα 8. Όπως φαίνεται στην απεικόνιση της αρχιτεκτονικής, το +Spaces λειτουργεί ως μια ενδιάμεση οντότητα μεταξύ των εικονικών περιβαλλόντων (3D εικονικών κόσμων και κοινωνικών δικτύων) και του επιπέδου εφαρμογών e-Government. Η πλατφόρμα επιτρέπει στους τελικούς χρήστες (Government officer) την ανάπτυξη και διαχείριση τέτοιων εφαρμογών μέσω μιας γραφικής διεπαφής (Front-End), με την οποία παρέχεται η πρόσβαση στο μεσολογισμικό και, μέσω αυτού, στις υπηρεσίες ανάλυσης, βάσει ενός σχήματος "συνδρομής" επί πληρωμή, συνοδευόμενης από συγκεκριμένες συμφωνίες παροχής:



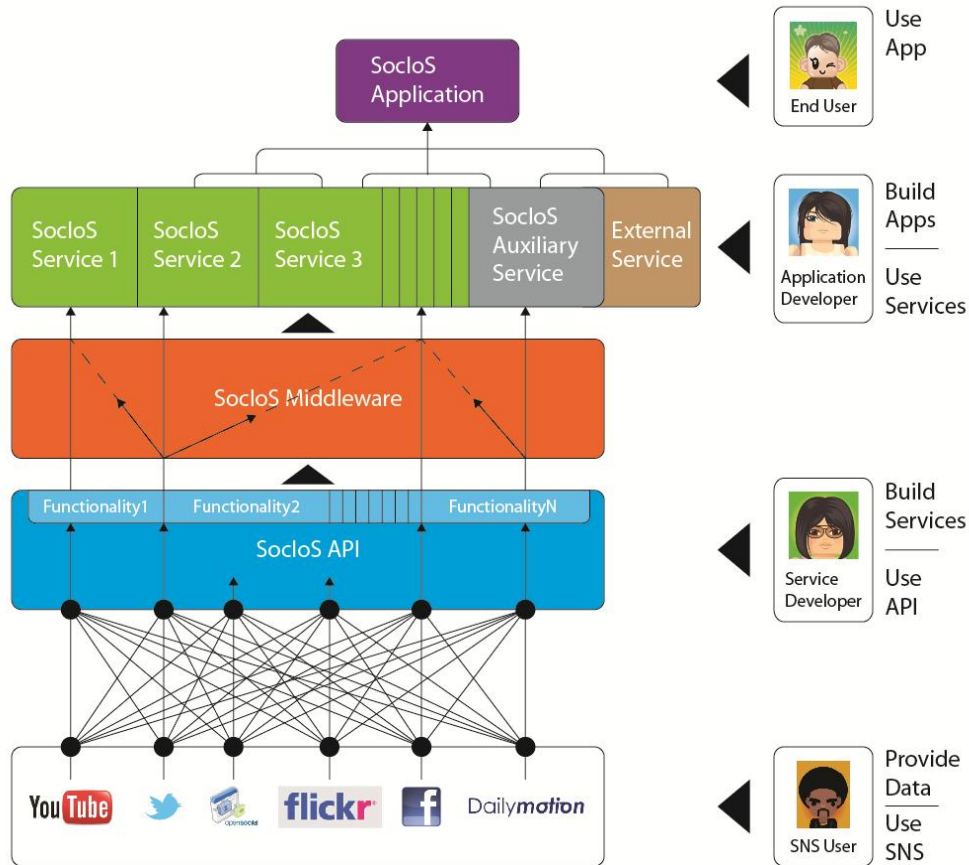
Σχήμα 8: Αρχιτεκτονική της πλατφόρμας του +Spaces

Μια βασική ιδέα της αρχιτεκτονικής αυτής είναι η υποστήριξη εικονικών περιβαλλόντων μέσω προσαρμογέων (Adaptors) που χρησιμοποιούν τις προγραμματιστικές διεπαφές τους (API). Ενιαία πρόσβαση σε αυτούς τους προσαρμογείς παρέχεται μέσω του στρώματος διαχείρισης εικονικών χώρων (VS Management Layer - VSML), που λειτουργεί ως επίπεδο αφαίρεσης πάνω από αυτούς. Ένα υπηρεσιοστρεφές μεσολογισμικό (SOA Middleware) χειρίζεται την ροή της επικοινωνίας και των δεδομένων μεταξύ των τελικών χρηστών, του VSML και των διαφόρων ενσωματωμένων και εξωτερικών υπηρεσιών ανάλυσης δεδομένων που διατίθενται μέσω αυτού και χρεώνουν τη χρήση των πόρων τους ανά κλήση. Εκτός από αυτές τις τρεις υπηρεσίες ανάλυσης, η πλατφόρμα είναι ανοικτή σε προσθήκη περισσότερων υπηρεσιών παρόμοιας ή διαφορετικής λειτουργικότητας, ώστε να παρέχει ένα ευρύ φάσμα εργαλείων.

2.3.2 SocIoS

Η πλατφόρμα του ευρωπαϊκού έργου SocIoS (Exploiting Social Networks for Building the Future Internet of Services) στοχεύει στην παροχή των εργαλείων για τη δημιουργία εφαρμογών που αξιοποιούν τη δυναμική και το περιεχόμενο των ιστοτόπων κοινωνικής δικτύωσης (Social Networking Sites - SNSs) (Tserpes, Papadakis, Kardara, Papaikononou, Aisopos, Sardis, & Varvarigou, 2012). Αυτά τα εργαλεία προσφέρονται ως διαδικτυακές υπηρεσίες SOAP, κάνοντας χρήση των ανοικτών APIs και των μεταδεδομένων των SNSs, προκειμένου να αντληθούν πληροφορίες και δεδομένα που "ζουν" μέσα σε αυτά από κάθε υπηρεσία για δικούς της λόγους.

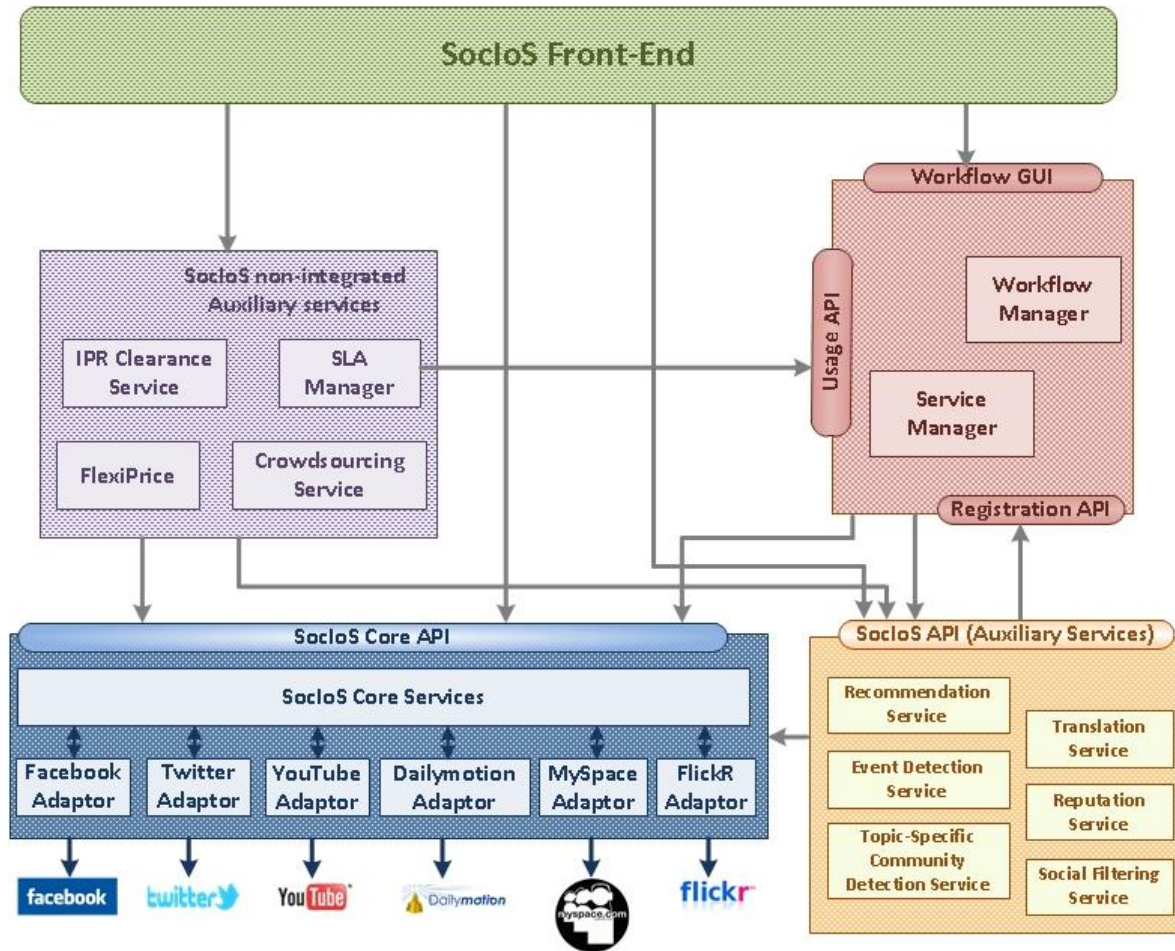
Η ροή των δεδομένων από τα κοινωνικά δίκτυα προς τον τελικό χρήστη μέσω των διαφόρων υπηρεσιών του SocIoS δείχνεται παρακάτω στο Σχήμα 9, μαζί με τα διάφορα επίπεδα χρήσης της πλατφόρμας (Papadakis, Tserpes, Sardis, Kardara, Papaikononou, & Aisopos, 2012). Πέρα από τον τελικό χρήστη, που είναι αυτός ο οποίος τελικά θα συγκεντρώσει τα δεδομένα που αντλούνται προς όφελος του, διακρίνεται ο απλός χρήστης του κοινωνικού δικτύου (SNS User), ο οποίος πρέπει να συναινέσει στη χρησιμοποίηση και επεξεργασία του περιεχόμενου που έχει δημοσιοποιήσει (User Created Content), καθώς και οι εμπλεκόμενοι προγραμματιστές (Service Developer και Application Developer). Ο Service Developer χρησιμοποιεί το ανοικτό API που παρέχει το SocIoS, για την δημιουργία και προσθήκη διαθέσιμων μέσω της πλατφόρμας υπηρεσιών Software as a Service, που είτε αντλούν περιεχόμενο από τα υποκείμενα κοινωνικά δίκτυα, είτε επεξεργάζονται τα δεδομένα που ανακτούνται κατόπιν πληρωμής. Ο Application Developer είναι αυτός που μπορεί να συνθέσει τις διαθέσιμες διαδικτυακές υπηρεσίες σε μια πιο σύνθετη εφαρμογή για τον τελικό χρήστη.



Σχήμα 9: Ροή δεδομένων στο SocIoS

Γίνεται σαφές λοιπόν ότι η πλατφόρμα επιτρέπει τη δημιουργική σύνθεση των προαναφερθέντων υπηρεσιών σε ροές εργασίας εφαρμογών προς κατανάλωση από τον τελικό χρήστη, αλλά και την παροχή απλών εφαρμογών απευθείας σε αυτόν με ένα ανοικτό μητρώο υπηρεσιών, δημιουργώντας μια δυναμική αγορά SaaS. Οι συνθέσεις αυτές μαζί με τα μητρώα υπηρεσιών και τις διάφορες διαχειριστικές οντότητες που χρειάζεται η πλατφόρμα για να λειτουργήσει, παρέχονται και ελέγχονται από το μεσολογισμικό στρώμα του SocIoS (SocIoS Middleware).

Μια πιο λεπτομερής εικόνα της αρχιτεκτονικής της πλατφόρμας του SocIoS φαίνεται παρακάτω στο Σχήμα 10:



Σχήμα 10: Αρχιτεκτονική της πλατφόρμας του SocIoS

Τα εννοιολογικά στοιχεία που εμφανίζονται στην αρχιτεκτονική είναι τα εξής:

- SocIoS Front-End: Η γραφική διεπαφή του SocIoS για τη σύνδεση των τελικών χρηστών, που παρέχει όλη τη λειτουργικότητα της πλατφόρμας για τη χρήση των υπηρεσιών του SocIoS και τη σύνθεσή τους σε ροές εργασίας (Workflow GUI).
- SocIoS non-integrated Auxiliary services: Ενσωματωμένες βοηθητικές υπηρεσίες της πλατφόρμας που αναλαμβάνουν διαχειριστικές και επικουρικές λειτουργίες της χρήσης των δεδομένων (πχ. ξεκαθάρισμα δικαιωμάτων χρήσης περιεχομένου, τιμολόγηση δεδομένων με κλειστή άδεια χρήσης από το FlexiPrice Service).

- SocIoS Core API (Core Services): Μια συνάθροιση μεθόδων που παρέχονται από τις διεπαφές (APIs) των υποκείμενων κοινωνικών δικτύων. Το SocIoS Core API παρέχει μια πρότυπη καθολική διεπαφή για τη συλλογή των μεθόδων και των αντικειμένων των SNS APIs. Αυτό επιτρέπει στους προγραμματιστές να δημιουργήσουν πολλαπλά στιγμιότυπα της ίδιας λειτουργικότητας σε διάφορες υπηρεσίες κοινωνικής δικτύωσης από μία και μοναδική κλήση μεθόδου.
- SocIoS API (Auxiliary Services): Οι βοηθητικές αυτές υπηρεσίες δεν αποτελούν μέρος του συνόλου των βασικών λειτουργιών που περιγράφονται από το SocIoS Core API. Είναι πιο πολύπλοκες, συνδυάζοντας τα δεδομένα με τους κοινωνικούς γράφους των χρηστών, ώστε να παρέχουν μια σύνθετη ανάλυση για διάφορους λόγους (αξιολόγηση εγκυρότητας SNS χρηστών/δεδομένων, ανίχνευση γεγονότων, μετάφραση, ανάλυση συναισθημάτων). Μπορούν να φιλοξενοούνται στην πλατφόρμα ή να παρέχονται από εξωτερικούς φορείς επί πληρωμή.
- SocIoS Middleware: Η λειτουργικότητα που η πλατφόρμα παρέχει στο επίπεδο του SocIoS (Core) API, παρέχεται σε επίπεδο υπηρεσιών διαδικτύου (Web Services), έτσι ώστε να καταστεί πιο εύχρηστη στο πλαίσιο της ανοιχτής αγοράς του SocIoS. Το μεσολογισμικό αναλαμβάνει την παροχή και διαχείριση των υπηρεσιών που θα αναπτυχθούν στο SocIoS, είτε διατηρώντας μητρώα υπηρεσιών στη διαχειριστική οντότητα Service Manager, είτε με τον ορισμό ροών εργασίας στο Workflow Manager, ή τέλος με ενσωματωμένες βοηθητικές υπηρεσίες που ελέγχουν την ποιότητα των υπηρεσιών (SLA Manager).

3

Ποιότητα Υπηρεσίας

Στο παρόν κεφάλαιο, ο αναγνώστης εισάγεται στην βασική έννοια της Ποιότητας Υπηρεσίας σε υπηρεσιοστρεφή περιβάλλοντα. Ακόμη και αν η έννοια αυτή παραμένει σχετικά ασαφής, η πρωταρχική σημασία της ποιότητας είναι αδιαμφισβήτητη τόσο για τις επιχειρήσεις-παρόχους όσο και για τους πελάτες. Έτσι, η έρευνα μαζί με την αγορά πάντα προσπαθούσαν να μοντελοποιήσουν την ποιότητα, προκειμένου να παρέχουν τα μέσα για να αξιολογηθεί. Σε ορισμένους τομείς και εφαρμογές, η ποσοτικοποίηση της ποιότητας έχει επιτευχθεί, αλλά οι ακριβείς απαιτήσεις δεν είναι εύκολο ακόμα να οριστούν από τους πελάτες (Parasuraman, Zeithaml, & Leonard, 1985), (Takeuchi & Quelch, 1983) και ως εκ τούτου το όλο μοντέλο της υπηρεσίας πληροί συνήθως μόνο τις προϋποθέσεις του φορέα παροχής υπηρεσιών.

Αυτό ισχύει συχνά όταν πρόκειται για ηλεκτρονικές υπηρεσίες, και ιδιαίτερα σε υπηρεσιοστρεφή περιβάλλοντα, όπως αυτά που χρησιμοποιούν SOA αρχιτεκτονικές. Η συνεχώς μεταβαλλόμενη φύση των τεχνολογιών αυτών, η πιθανή «έλλειψη εξοικείωσης με τεχνικούς όρους» των καταναλωτών και η πολύπλοκη ιεραρχία των σύγχρονων επιχειρήσεων, συχνά καθιστώντας δύσκολο να αναγνωρισθεί ποιός είναι ο τελικός

χρήστης, συμβάλλουν ώστε να γίνεται το πρόβλημα της μοντελοποίησης της ποιότητας εξαιρετικά δύσκολο.

3.1

Η έννοια της Ποιότητας

Υπηρεσίας

Σε γενικές γραμμές, όλα τα υπηρεσιοστρεφή καταναμημένα περιβάλλοντα υποστηρίζουν τη δυναμική παροχή κλιμακούμενων και εικονικοποιημένων πόρων σαν υπηρεσία στο διαδίκτυο. Στο Software as a Service, που αυτό που μας ενδιαφέρει περισσότερο εδώ, αυτό γίνεται μέσω μιας εφαρμογής που παρέχεται ως υπηρεσία. Σε τέτοια συστήματα, οι υπολογιστικοί πόροι μπορούν να παρασχεθούν εξωτερικά ως μια εικονικοποιημένη "ομάδα" πόρων, σαν να υπήρχε από κάτω μια ενιαία υποδομή με συγκεντρωμένες τις δυνατότητες των επί μέρους διασυνδεδεμένων υποδομών του κάθε περιβάλλοντος. Οι πελάτες των υπηρεσιών μπορούν να εκτελέσουν διάφορες εφαρμογές με κλήσεις των αντίστοιχων υπηρεσιών, συνοδευόμενες από μια σειρά ειδικών απαιτήσεων με συγκεκριμένο κόστος. Για κάθε αίτηση, ο πάροχος μπορεί να εκχωρήσει δυναμικά το μέγεθος της μνήμης, της επεξεργαστικής ισχύος ή του χώρου αποθήκευσης σε μια συγκεκριμένη εργασία πελάτη και άρα οι επιδόσεις και οι δυνατότητες των παρεχόμενων υπηρεσιών μπορούν να διαφέρουν ανάλογα με τη διαμόρφωσή τους κάθε φορά.

Εκμεταλλευόμενοι την παραπάνω ιδιότητα (ελαστικότητα), οι πάροχοι είναι σε θέση να διατηρήσουν υψηλή τη χρησιμοποίηση των πόρων τους, για να παρέχουν στους τελικούς χρήστες την απαιτούμενη Ποιότητα Υπηρεσίας (Quality of Service - QoS). Η έννοια του QoS γίνεται διαφορετικά αντιληπτή στον πάροχο (πχ. υψηλή δέσμευση πόρων για μια εργασία) και στον πελάτη (πχ. γρήγορη απόκριση, αξιοπιστία/ακρίβεια

αποτελεσμάτων) μιας υπηρεσίας. Έτσι, γίνεται σαφές ότι συγκεκριμένα μέτρα απόδοσης πρέπει να θεσπιστούν για τη διασφάλιση της αποτελεσματικότητας και της αποδοτικότητας της αλυσίδας παροχής υπηρεσιών, δεδομένου ότι όπως αναφέρθηκε και στην εισαγωγή, οδεύουμε σε μια γιγαντιαία υπερπροσφορά πληροφοριών και υπηρεσιών στο διαδίκτυο, που αν μη τι άλλο επιβάλλει την αξιολόγησή τους και το διαχωρισμό των πιο χρήσιμων και των περιττών ή αναξιόπιστων. Τα μέτρα αυτά απόδοσης μπορούν να ταξινομηθούν είτε ως ποιοτικά (π.χ. μέτρο ικανοποίησης του πελάτη της υπηρεσίας) ή ως ποσοτικά (π.χ. μέτρα που εξαρτώνται άμεσα από το κόστος ή το κέρδος που ορίζεται στο τιμολόγιο παροχής υπηρεσιών) (Beamon 1998).

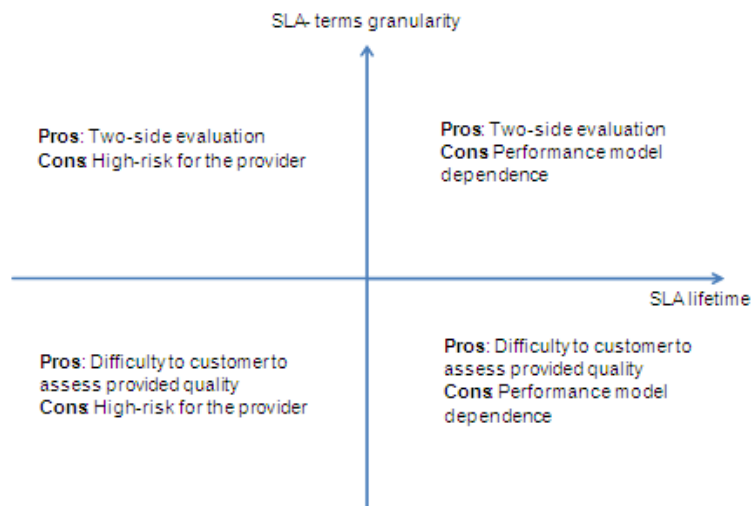
Η ποιότητα των υπηρεσιών και οι επιδόσεις τους καταγράφονται και αξιολογούνται σε μια μορφή ηλεκτρονικής σύμβασης, που ονομάζεται Service Level Agreement (SLA). Οι όροι αυτού του είδους της σύμβασης, που παρουσιάζεται αναλυτικά στην επόμενη ενότητα, καθορίζουν - ρητά ή εμμέσως- τους πόρους που πρόκειται να διατεθούν από τον πάροχο για τον καταναλωτή της υπηρεσίας, ενώ επίσης προσδιορίζουν και το χρονικό διάστημα που η συμφωνία αυτή θα είναι ενεργή, το κόστος των πόρων, καθώς και ρήτρες διασφάλισης των δύο πλευρών από πιθανές παραβιάσεις προθεσμιών.

3.2 *Service Level Agreement*

Στις αρχές του 2000, ένας νέος όρος επινοήθηκε, ο οποίος έχει ίσως περισσότερο από οποιοδήποτε άλλον σχέση με την Ποιότητα Υπηρεσίας και τα υπηρεσιοστρεφή συστήματα, αυτός της Συμφωνίας σε Επίπεδο Υπηρεσίας (Service Level Agreement - SLA) (Tian, Gramm, Naumowicz , Ritter, & Schiller, 2003), (Sahai, Machiraju, Sayal, Jin & Casati, 2002), (Keller & Ludwig, 2002). Φυσικά, ο ίδιος ο όρος είχε καθοριστεί

πριν από πολύ καιρό ως «... μια συμφωνία μεταξύ του παρόχου της υπηρεσίας και των πελατών της που ποσοτικοποιεί την ελάχιστη ποιότητα των υπηρεσιών που ανταποκρίνεται στις απαιτήσεις» (Hiles, 1994).

Με άλλα λόγια, τα SLAs είναι η επισημοποίηση μιας αποδείξιμης, ποσοτικής αξιολόγησης του QoS πάνω στην οποία ο πάροχος και ο καταναλωτής-πελάτης μπορούν να διαπραγματευτούν και να συμφωνήσουν. Κατόπιν συμφωνίας, ο πάροχος δεσμεύεται να εξασφαλίσει τους απαραίτητους πόρους, έτσι ώστε να ικανοποιηθούν οι απαιτήσεις του καταναλωτή. Ο καταναλωτής είναι τότε σε θέση να αξιολογήσει τα αποτελέσματα της υπηρεσίας σε αντιδιαστολή με το σύνολο των όρων του SLA. Υπάρχουν διάφορα μοντέλα διαπραγμάτευσης για να πραγματοποιηθεί αυτή η διαδικασία. Οι κύριοι άξονες διαφοροποίησης των μοντέλων αυτών είναι η διάρκεια ζωής και η αποκρυστάλλωση των όρων του SLA σε μετρήσιμα μεγέθη (βλέπε παρακάτω σχήμα). Η διαφοροποίηση αυτή αφορά κυρίως την πλευρά (πάροχος ή πελάτης) που πρέπει να κάνει την αξιολόγηση, αλλά και την πλευρά που είναι πιο επιρρεπής σε παραβιάσεις του SLA (συνήθως αυτές οι πλευρές είναι οι δύο αντίθετες).



Σχήμα 11: Η ποικιλία των μοντέλων SLA

Στο Σχήμα 11, όσο δεξιά προχωράμε στον άξονα «SLA lifetime», τόσο περισσότερο ο πάροχος πρέπει να εξασφαλίζει πόρους του για τον πελάτη, ενώ όσο πάνω προχωράμε στον άξονα «granularity», τόσο πλησιάζουμε στη χρήση όρων που είναι πιο κοντά στην κατανόηση του πελάτη.

Σε γενικές γραμμές, η χρήση SLA με μεγάλη διάρκεια ζωής απαιτεί την υιοθέτηση σχημάτων κράτησης πόρων εκ των προτέρων από τον πάροχο και την εκτίμηση των μελλοντικών απαιτήσεων, για την εξασφάλιση της βέλτιστης κατανομής των πόρων, για την ελαχιστοποίηση των παραβιάσεων. Επιπροσθέτως, οι περιπτώσεις στις οποίες έχει νόημα για τον πελάτη να κρατάει μακροπρόθεσμα SLAs (όπως αναφέρονται και χρησιμοποιούνται εδώ: (Aisopos, Tserpes, Kardara, Panousopoulos, Phillips, & Salamouras, 2010)) είναι μόνο αυτές όπου η εφαρμογή το επιτρέπει (π.χ. παροχή υπηρεσιών υπολογιστικού νέφους, ως μέρος μιας ροής επιχειρηματικών εργασιών (Patel, Ranabahu, & Sheth, 2009)). Αν αυτό συμβαίνει, τότε το πιθανότερο είναι ότι οι όροι του SLA θα προσανατολίζονται στους πόρους που διατίθενται, είτε περιγράφοντας άμεσα την ποσότητα και το είδος των πόρων που θα ανατεθούν στον πελάτη κάθε φορά που καλεί την υπηρεσία ή σε μια συγκεκριμένη χρονική περίοδο (π.χ. 4 CPUs), ή γενικά περιγράφοντας την παροχή πόρων από τον πάροχο (π.χ. διαθεσιμότητα και αξιοπιστία).

Εκτός από τον καθορισμό του QoS που πρέπει να παραδοθεί από τον πάροχο, τα SLAs μπορούν επίσης να επιβάλλουν περιορισμούς στους πόρους που ο καταναλωτής επιτρέπεται να χρησιμοποιήσει. Αν ο πελάτης υπερβεί τη χρήση που ορίζεται στο SLA, τότε υπόκειται στους όρους που αφορούν την παραβίαση. Τέτοια μοντέλα που υποστηρίζονται τόσο από περιβάλλοντα όπως το μεσολογισμικό GRIA (Boniface,

Phillips, Sanchez-Macian Perez, & SurrIDGE, 2007) που εστιάζουν στη διαχείριση της χωρητικότητας, όσο και σε περιβάλλοντα Infrastructure as a Service (Waheed Iqbal, 2009), (Amazon: Service level agreement for ec2, 2008).

Τα βραχυπρόθεσμα SLAs από την άλλη, απαιτούν μηχανισμούς παρακολούθησης και αξιολόγησης σε πραγματικό χρόνο, καθώς και σχήματα (σχεδόν) βέλτιστης κατανομής πόρων συστήματος. Μια από τις βασικές προκλήσεις για τον πάροχο εδώ είναι η παρακολούθηση σε πραγματικό χρόνο της χωρητικότητας και των πόρων, ώστε να αποφασίζει δυναμικά εάν νέα SLAs μπορούν να γίνουν αποδεκτά με βάση τον τρέχοντα φόρτο εργασίας. Προτάσεις πάνω σε τέτοιους μηχανισμούς και συστήματα έχουν γίνει από το έργο IRMOS (Cucinotta, Checconi, Abeni, & Palopoli, 2010; Addis, Zlatev, Mitchell, & Boniface, 2009) ωστόσο, οι πιο ολοκληρωμένες προτάσεις προέρχονται από τη δουλειά των (Raimondi, Skene, & Emmerich, 2008). Σε αντίθεση με τα μακροπρόθεσμα, τα βραχυπρόθεσμα SLAs εμποτίζουν μια εγγενή δυσκολία στον πάροχο, όσον αφορά την επαρκή διαχείριση και παρακολούθηση των πόρων του, λόγω των στενών χρονικών περιθωρίων.

Ο άλλος άξονας της διαφοροποίησης που αναφέρθηκε πιο πάνω αφορά στην κατανόηση και τη δυνατότητα αξιολόγησης των όρων ενός SLA. Το μοντέλο της Ποιότητας Υπηρεσίας που θα αναφερθεί παρακάτω μπορεί να είναι είτε πιο κοντά στο χαμηλό επίπεδο, με όρους που προσανατολίζονται στους πόρους, ή πιο κοντά στο υψηλό επίπεδο, με όρους που προσανατολίζονται στην εφαρμογή. Για κάθε επιλογή, μια ποικιλία γλωσσών και μηχανισμών έχουν αναπτυχθεί, με μερικούς από τους οποίους να χρησιμοποιούνται ευρέως. Το πιο ευρέως αποδεκτό πλαίσιο για την περιγραφή των πόρων είναι το Πλαίσιο Περιγραφής Πόρων (Resource Description Framework - RDF)

(Lassila & Swick, 1999), σκοπός του οποίου είναι να καθορίσει ένα μηχανισμό για την περιγραφή των πόρων, που δεν κάνει υποθέσεις για ένα συγκεκριμένο πεδίο εφαρμογής, ούτε καθορίζει a priori τη σημασιολογία του κάθε τομέα εφαρμογής.

Από την άλλη πλευρά, υπάρχουν και γλώσσες για συγκεκριμένους τομείς (Domain Specific Modeling Languages - DSML) για SLAs, όπως η Platform-Independent Component Modeling Language (PICML) που προτάθηκε από τη δουλειά των Balasubramanian, Balasubramanian, Parsons, Gokhale, & Schmidt (2007) και το μοντέλο που προτείνουν οι Skene, Raimondi, & Emmerich (2009). Η μοντελοποίηση της ποιότητας επιτυγχάνεται με τη χρήση DSML αναφέρεται συνήθως ως Quality of Business (QoBiz) και οι ποσοτικοποιημένοι όροι που εκφράζουν την ποιότητα στο πλαίσιο αυτό καλούνται επίσης μετρικές QoBiz (Moorsel, 2001).

3.3

Μοντέλο της Ποιότητας

Υπηρεσίας

Όπως αναλύθηκε στις προηγούμενες παραγράφους, οι Συμφωνίες σε Επίπεδο Υπηρεσίας αποτελούν ένα θεμελιώδες εργαλείο για τη ρύθμιση της σχέσης πελάτη-παρόχου, τον καθορισμό και την ποσοτικοποίηση του παρεχόμενου επιπέδου της Ποιότητας Υπηρεσίας. Μια σειρά από όρους που καθορίζονται με βάση τη συγκεκριμένη εφαρμογή περιλαμβάνονται στο "έγγραφο" ενός SLA, έτσι ώστε να μπορέσει ο πελάτης να καθορίσει την εξυπηρέτηση των αναγκών του με βάση τις ικανότητες του παρόχου της υπηρεσίας. Μόλις επιτευχθεί συμφωνία, και οι δύο μπορούν να παρακολουθούν την ποιότητα της παρεχόμενης υπηρεσίας (Kardara, Konstanteli, Aisopos, Andronikou, &

Varvarigou, 2009) και εφόσον κάποιος όρος παραβιάζεται από κάποια πλευρά, η σύμβαση προβλέπει ρήτρες αποζημίωσης, που περιλαμβάνονται επίσης στο SLA.

Οι αδυναμίες που εμφανίζονται εδώ σχετίζονται με τη δυσκολία της κάθε πλευράς στην παρακολούθηση και αξιολόγηση του QoS σύμφωνα με το SLA, όταν οι όροι είναι πιο κοντά στην κατανόηση της άλλης πλευράς. Γλώσσες όπως το RDF είναι κοινά αποδεκτές από τους παρόχους για περιγραφή του μοντέλου Ποιότητας Υπηρεσίας, αλλά το γεγονός ότι δεν επιτρέπουν την εξωτερική πρόσβαση στους πόρους, σημαίνει ότι ο πελάτης δεν μπορεί να επικυρώσει την παρεχόμενη ποιότητα χρησιμοποιώντας τους όρους ενός τέτοιου SLA. Από την άλλη πλευρά, η υιοθέτηση γλωσσών DSML για την αναπαράσταση του SLA συνεπάγεται ότι ο πάροχος πρέπει να διαθέτει ένα μοντέλο υψηλής απόδοσης, κάτι στο οποίο δεν μπορούμε να στηριχτούμε όπως δείχνει η εμπειρία, εκτός κι αν επικεντρωθούμε σε συγκεκριμένες υλοποιήσεις και εφαρμογές.

Στα πλαίσια της παρούσας διατριβής υιοθετούμε τη γενική περίπτωση, όπου τα SLAs μπορεί να περιέχουν όρους χαμηλού (πχ. ποσότητα μνήμης), αλλά και υψηλού επιπέδου (πχ. προθεσμία απόκρισης, ποιότητα αποτελέσματος) που είναι κατανοητοί και στις δύο πλευρές. Για την αξιολόγησή τους, στην πρώτη περίπτωση ο πελάτης πρέπει να βασιστεί στην εμπειρία του, όπως θα περιγραφεί και στο κεφάλαιο που αναφέρεται στην αξιολόγηση ποιότητας υπηρεσίας, ενώ στη δεύτερη ο πάροχος επίσης χρησιμοποιώντας επίσης την εμπειρία του θα πρέπει να "μεταφράσει" τις υψηλού επιπέδου απαιτήσεις σε συγκεκριμένες απαιτήσεις πόρων (κάνοντας μια όσο το δυνατόν πιο ακριβή εκτίμηση).

3.4

Ποιότητα Υπηρεσίας σε περιβάλλοντα SaaS

Στην ενότητα αυτή παρουσιάζεται το WS-Agreement που αποτελεί ένα κοινά αποδεκτό αφαιρετικό μοντέλο για ηλεκτρονικές συμβάσεις σε SaaS, βασισμένο στα πρότυπα των διαδικτυακών υπηρεσιών (οικογένεια WS-Standards), και ακολούθως παραδείγματα εφαρμογών που μπορούν να παρασχεθούν ως υπηρεσίες SaaS σε μια ανοικτή αγορά υπηρεσιών, ώστε να γίνουν πιο κατανοητοί οι όροι υψηλού επιπέδου και η αντιστοίχισή τους σε πόρους, στα πλαίσια των SLAs τέτοιων εφαρμογών.

3.4.1 WS-Agreement

Το πρότυπο Web Services Agreement (A. Andrieux, 2011), ή πιο απλά WS-Agreement, είναι ένα επεκτάσιμο πρωτόκολλο διαδικτυακών υπηρεσιών που δημιουργήθηκε από τον οργανισμό Open Grid Forum για την επίτευξη συμφωνίας μεταξύ ενός παρόχου υπηρεσιών και ενός καταναλωτή των υπηρεσιών αυτών, χρησιμοποιώντας μια επεκτάσιμη XML γλώσσα για τον καθορισμό της δομής της συμφωνίας, καθώς και των προτύπων συμφωνιών για τη διευκόλυνση της ανακάλυψης και διαπραγμάτευσης μεταξύ των δύο.

Ο στόχος του WS-Agreement είναι να καθορίσει μια γλώσσα και ένα κοινό πρωτόκολλο για τη διαφήμιση των δυνατοτήτων ενός παρόχου υπηρεσιών και τη δημιουργία συμφωνιών που βασίζονται σε δυναμικές προσφορές, καθώς και για την επιβεβαίωση της συμμόρφωσης με τους όρους της συμφωνίας σε πραγματικό χρόνο. Μια συμφωνία μεταξύ ενός καταναλωτή υπηρεσιών και του παρόχου ορίζει έναν ή περισσότερους όρους σε επίπεδο υπηρεσιών, τόσο ως έκφραση των αναγκών του

καταναλωτή και όσο και ως διαβεβαίωση από τον πάροχο υπηρεσιών σχετικά με τη διαθεσιμότητα των πόρων ή/και την ποιότητα των υπηρεσιών. Για παράδειγμα, μια συμφωνία μπορεί να παρέχει διαβεβαιώσεις για το μέγιστο χρόνο απόκρισης και τη διαθεσιμότητα της υπηρεσίας. Εναλλακτικά, μπορεί να παρέχει διαβεβαιώσεις σχετικά με τη διαθεσιμότητα πόρων όπως της μνήμης, του χώρου αποθήκευσης, CPU, κτλ. Για να αποκτήσει αυτές τις διαβεβαιώσεις για την ποιότητα των υπηρεσιών, ο καταναλωτής ή οντότητα πελάτη που ενεργεί για λογαριασμό του πρέπει να επικυρώσει ένα ηλεκτρονικό συμβόλαιο παροχής υπηρεσιών με τον πάροχο, ή όποια άλλη οντότητα ενεργεί για λογαριασμό του παρόχου υπηρεσιών. Εφόσον οι όροι της συμφωνίας σχετίζονται με τον ορισμό του τύπου της υπηρεσίας, ο ορισμός αυτός πρέπει να είναι μέρος των όρων ή να συμφωνηθεί πριν από τη δημιουργία του ηλεκτρονικού συμβολαίου.

Η προδιαγραφή του WS-Agreement παρέχει ένα σχήμα για τον καθορισμό της συνολικής δομής του εγγράφου μιας τέτοιας συμφωνίας. Όπως φαίνεται στον Πίνακα 4, ένα WS-Agreement ορίζει πληροφορίες σχετικά με τα μέρη μεταξύ των οποίων συνάπτεται η συμφωνία, τη διάρκεια της συμφωνίας αυτής και ένα σύνολο από όρους. Οι όροι μπορεί να περιλαμβάνουν έναν ή περισσότερους όρους σχετικά με την εφαρμογή που παρέχεται ως υπηρεσία και ορισμένους όρους εγγυήσεων, προσδιορίζοντας στόχους ποιότητας σε επίπεδο υπηρεσιών και επιχειρηματικές αξίες που σχετίζονται με αυτούς τους στόχους. Η δομή αυτή είναι δυναμική και πλήρως επεκτάσιμη, εφόσον μπορούν να προστεθούν επιπλέον πληροφορίες και διαφορετικού τύπου όροι σε οποιοδήποτε μέρος του προτύπου, παραδείγματα των οποίων θα δούμε παρακάτω.

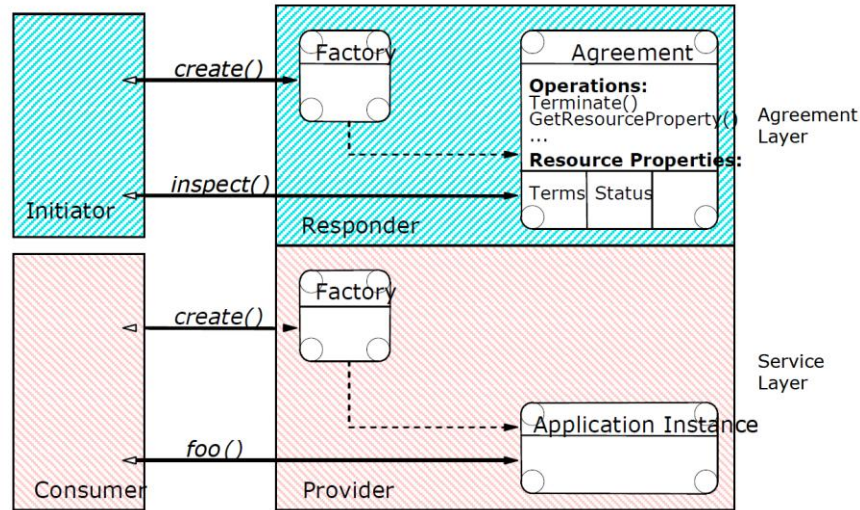
```

<wsag:Agreement AgreementId="xs:string">
  <wsag:Name>
    xs:string
  </wsag:Name>
  <wsag:Context xs:anyAttribute>
    <wsag:AgreementInitiator>xs:anyType</wsag:AgreementInitiator>
    <wsag:AgreementResponder>xs:anyType</wsag:AgreementResponder>
    <wsag:ExpirationTime>xs:DateTime</wsag:ExpirationTime>
    <xs:any/>
  </wsag:Context>
  <wsag:Terms>
    <wsag:ServiceDescriptionTerm>
      wsag:ServiceDescriptionTermType
    </wsag:ServiceDescriptionTerm>
    <wsag:ServiceReference>
      wsag:ServiceReferenceType
    </wsag:ServiceReference>
    <wsag:ServiceProperties>
      wsag:ServicePropertiesType
    </wsag:ServiceProperties>
    <wsag:GuaranteeTerm>
      wsag:GuaranteeTermType
    </wsag:GuaranteeTerm>
  </wsag:Terms>
</wsag:Agreement>

```

Πίνακας 4: Απλοποιημένο σχήμα της βασικής δομής του WS-Agreement

Η διαδικασία δημιουργίας μιας συμφωνίας ξεκινά συνήθως με ένα προκαθορισμένο πρότυπο συμφωνίας (Agreement Template), το οποίο προσδιορίζει κάποιες προσαρμόσιμες πτυχές του συμβολαίου και τους κανόνες που πρέπει να ακολουθούνται για τη δημιουργία του, τους οποίους αποκαλούμε περιορισμούς δημιουργίας συμφωνίας. Τα πρότυπα συμφωνίας επίσης ορίζονται από τις προδιαγραφές του WS-Agreement με μια δομή συναφή αυτής μιας συμφωνίας. Η δημιουργία μιας συμφωνίας μπορεί να ξεκινήσει από την πλευρά του καταναλωτή υπηρεσίας ή από την πλευρά της παροχής, αφού το πρωτόκολλο προβλέπει διεπαφές που επιτρέπουν συμμετρικούς τρόπους δημιουργίας. Με βάση τις συμμετρικές αυτές προσεγγίσεις, ένα εννοιολογικό μοντέλο για την αρχιτεκτονική υπηρεσιών του WS-Agreement, με έμφαση στις διεπαφές που ορίζονται παρουσιάζεται στο Σχήμα 12 (A. Andrieux, 2011):



Σχήμα 12: Εννοιολογικό μοντέλο πρότυπης διεπίπεδης υπηρεσίας WS-Agreement

Το παραπάνω μοντέλο έχει δύο επίπεδα όπως φαίνεται και στο σχήμα, που περιγράφονται παρακάτω. Σε κάθε περίπτωση όμως, λόγω των πολλαπλών δυνατοτήτων και της επεκτασιμότητας όσον αφορά το σχεδιασμό ενός συστήματος WS-Agreement, αποφάσεις σχετικές με το συγκεκριμένο τομέα και τις συγκεκριμένες εφαρμογές πρέπει να λαμβάνονται, όσον αφορά τη σύνθεση της λειτουργίας και το σχεδιασμό διεπαφών.

3.4.1.1 Επίπεδο Συμφωνίας

Το επίπεδο συμφωνίας (Agreement Layer) παρέχει υπηρεσιοστρεφείς διεπαφές που μπορούν να χρησιμοποιηθούν για τη δημιουργία, την αναπαράσταση και τον έλεγχο των συμφωνιών, με χρήση και των υπηρεσιών που υλοποιούνται στο επίπεδο υπηρεσίας. Το επίπεδο συμφωνίας έχει πάντα τους ακόλουθους τύπους διεπαφών:

- Μια διεπαφή παραγωγής συμφωνιών (Agreement Factory) για τη δημιουργία μιας συμφωνίας από ένα αρχικό σύνολο όρων. Επιστρέφει μια τελική αναφορά (Endpoint Reference - EPR) σε μια υπηρεσία συμφωνιών. Η διεπαφή αυτή εκθέτει επίσης τις αποδεκτές πρότυπες συμφωνίες για τη δημιουργία της τελικής

συμφωνίας. Η σύζευξη της συμφωνίας με τη συγκεκριμένη υπηρεσία στην οποία αναφέρεται πρέπει να περιγράφεται στη συμφωνία, και μπορεί να πάρει εναλλακτικές μορφές:

- Η υπηρεσία μπορεί να αναφέρεται στη συμφωνία ως μέρος των όρων της (και συνεπώς μπορεί να αποτελέσει αντικείμενο διαπραγμάτευσης).
- Η υπηρεσία μπορεί να δημιουργηθεί ανά συμφωνία, δηλαδή το επίπεδο συμφωνίας να έχει τον έλεγχο της δημιουργίας του στιγμιότυπου της υπηρεσίας, με τη συμφωνία να περιγράφει τη συμπεριφορά της νεοσυσταθείσας υπηρεσίας.
- Η υπηρεσία μπορεί να δημιουργηθεί εξωτερικά, αλλά να φέρει συγκεκριμένα αναγνωριστικά που επιτρέπουν τη σύζευξη με μια συγκεκριμένη συμφωνία. Για παράδειγμα, μια συμφωνία σχετικά με το εύρος ζώνης του δικτύου μπορεί να αναφέρεται σε συγκεκριμένα μετα-δεδομένα δικτύου (όπως πεδία στις κεφαλίδες των μηνυμάτων) ως ένας τρόπος να δηλώσει εγγυήσεις για συγκεκριμένο φόρτο δικτύου.
- Μια διεπαφή συμφωνίας, με τις λειτουργίες ελέγχου της κατάστασης εκτέλεσης και των μετα-δεδομένων της συμφωνίας. Επιπλέον μέθοδοι διαχείρισης σχετικές με συγκεκριμένες λειτουργίες μπορούν να προστεθούν σε αυτήν.

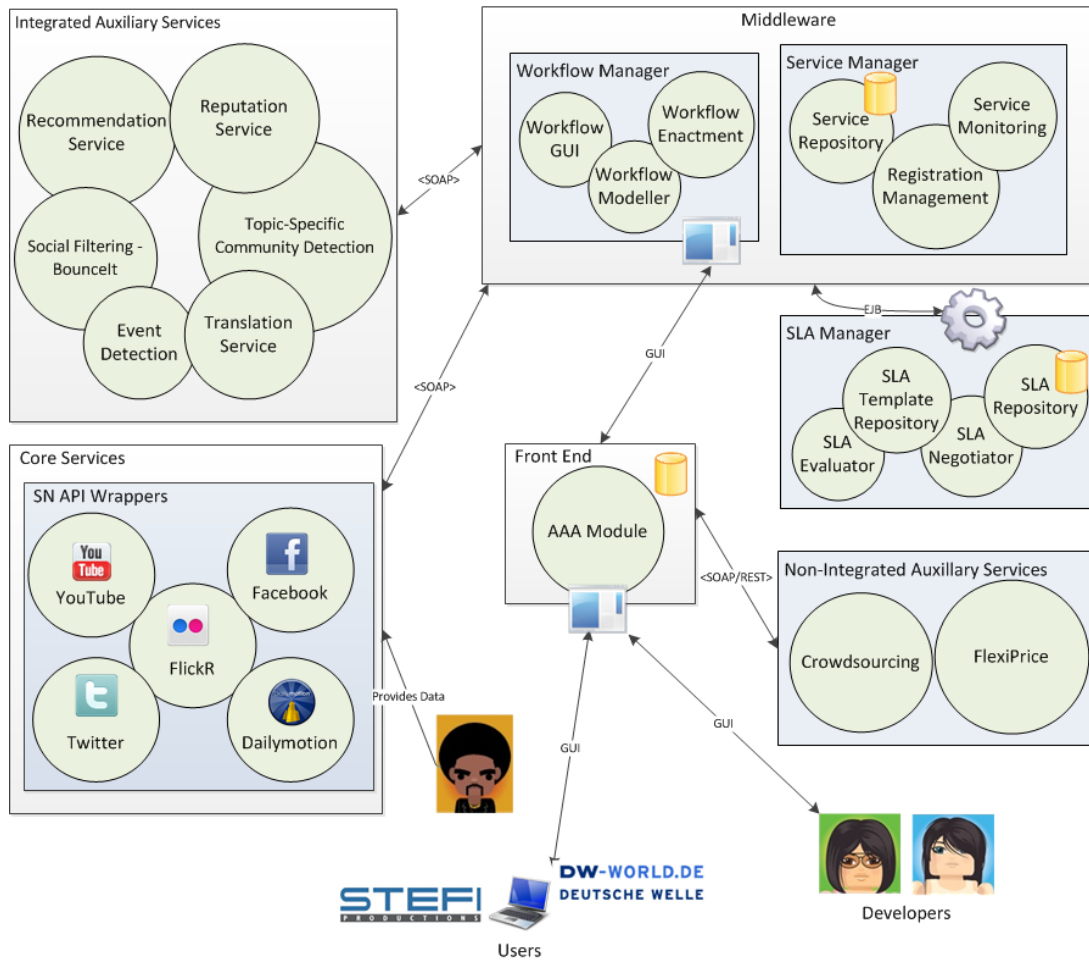
3.4.1.2 Επίπεδο Υπηρεσίας

Το επίπεδο υπηρεσίας αποτελεί το στρώμα της υπηρεσίας που είναι περισσότερο προσανατολισμένο στην εφαρμογή, σύμφωνα με το πρότυπο του WS-Agreement. Η παρεχόμενη υπηρεσία μπορεί προαιρετικά να παρέχει μια διεπαφή διαδικτυακής υπηρεσίας: Για παράδειγμα, οι υπολογιστικές εργασίες μπορούν να εικονικοποιούνται ως

στιγμιότυπα διαδικτυακών υπηρεσιών με πρόσθετες διεπαφές. Άλλες υπηρεσίες μπορούν να μην έχουν μια υπηρεσιοστρεφή αναπαράσταση: Η διαθεσιμότητα του δικτύου μπορεί να θεωρηθεί ως μια κατηγορία υπηρεσίας χωρίς αναπαράσταση διαδικτυακής υπηρεσίας, αλλά μπορεί να είναι χρήσιμη στη διαχείριση των παρεχόμενων χαρακτηριστικών ποιότητας υπηρεσίας, μέσω συμφωνιών που ορίζονται στο επίπεδο πάνω από το επίπεδο υπηρεσίας. Οι διεπαφές σε αυτό το επίπεδο είναι συγκεκριμένες στη λειτουργικότητα, και δεν χρειάζεται να μεταβληθούν, με την εισαγωγή του επιπέδου συμφωνίας.

3.4.2 Εφαρμογές SaaS σε αγορές υπηρεσιών και παράμετροι ποιότητας

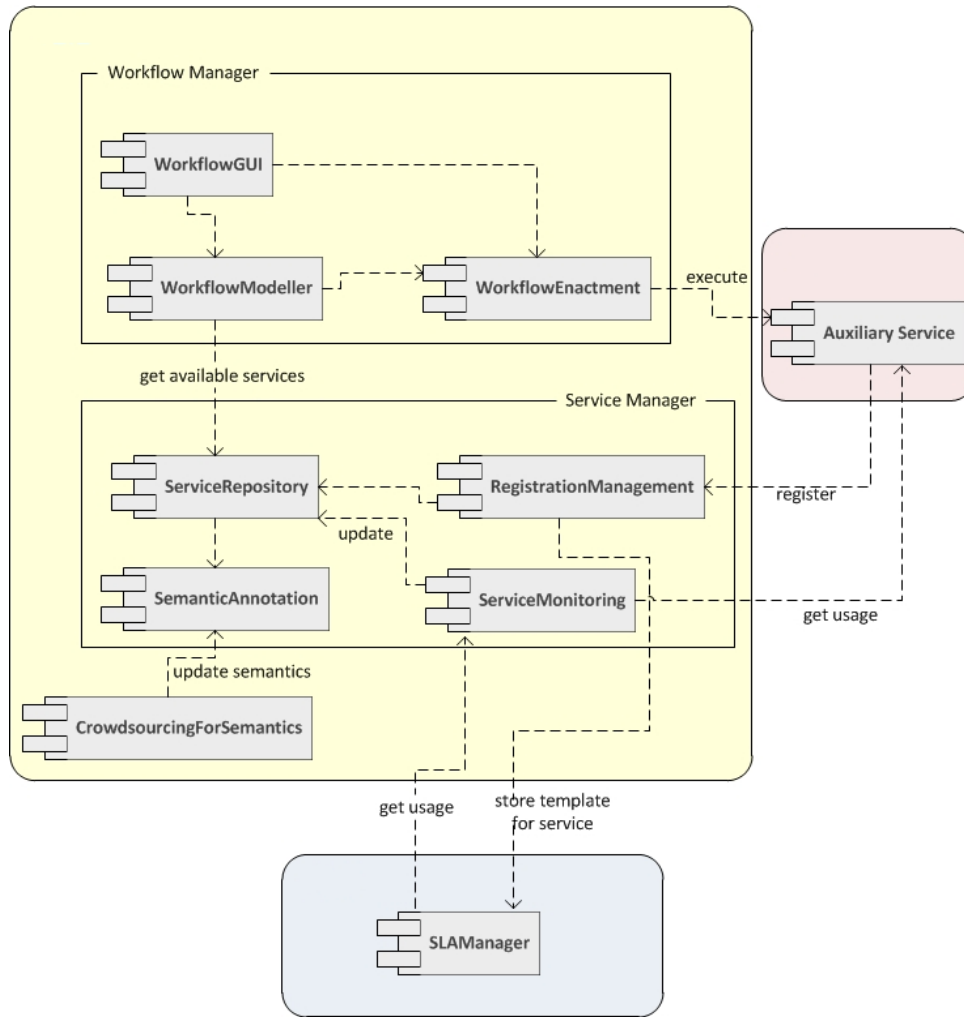
Σε ανοικτές αγορές υπηρεσιών SaaS, όπως στην περίπτωση της πλατφόρμας του SocIoS που παρουσιάστηκε παραπάνω, οι εφαρμογές που παρέχονται ως υπηρεσίες SOAP και οι παράμετροι ποιότητας για κάθε μια διαφέρουν, ανάλογα με τον τελικό χρήστη της πλατφόρμας. Στην περίπτωση του SocIoS οι τελικοί χρήστες που συνδέονται στο Front-End GUI μέσω ενός AAA (Authorization, Authentication, Accounting) Module, ποικίλουν από δημοσιογραφικούς οργανισμούς, που επιθυμούν άντληση ειδήσεων και δεδομένων πολυμέσων από τα κοινωνικά δίκτυα και ανάλυσή τους, μέχρι εταιρίες casting και 3D video animators, που χρησιμοποιούν τα δίκτυα αυτά για εύρεση ηθοποιών ή τοποθεσιών, και δημιουργία διαφημιστικών σποτ (Σχήμα 13).



Σχήμα 13: Χρήση της πλατφόρμας του SocIoS για εύρεση χρηστών/δεδομένων από τα κοινωνικά δίκτυα και επεξεργασία τους

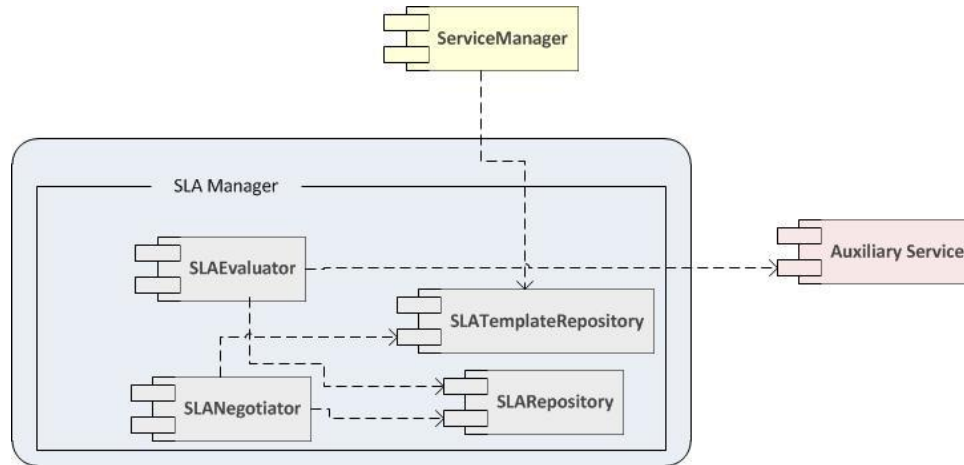
Με βάση τη ζητούμενη εφαρμογή ανάλυσης ή επεξεργασίας δεδομένων κάθε φορά, οι παράμετροι ποιότητας που καταγράφονται στα SLAs των τελικών χρηστών μπορεί να αναφέρονται σε υψηλό επίπεδο (εφαρμογής) ή χαμηλό επίπεδο (πόρων). Όπως φαίνεται στο παρακάτω διάγραμμα συστατικών λογισμικού (Σχήμα 14), οι διάφορες βοηθητικές υπηρεσίες (Integrated Auxiliary Services) ελέγχονται από το εργαλείο διαχείρισης του Service Manager που καταγράφει και μοιράζεται τη χρησιμοποίηση (με την κλήση της μεθόδου get usage) αλλά και την απόδοσή τους με το SLA Manager, που

παρακολουθεί και επικυρώνει τους όρους των συμφωνιών SLA, που έχουν προκύψει κατά τη διαδικασία εγγραφής (registration) των υπηρεσιών.



Σχήμα 14: Διάγραμμα συστατικών λογισμικού SocIoS Middleware

Η εσωτερική αρχιτεκτονική του SLA Manager του SocIoS φαίνεται παρακάτω. Σημειώτεον ότι τα SLAs και οι μέθοδοι που τα διαχειρίζονται ακολουθούν το πρότυπο του WS-Agreement, το οποίο έχει επεκταθεί σε αυτήν την περίπτωση, για να περιλαμβάνει επιπλέον στοιχεία και πληροφορίες (πχ. όροι χρήσης Disclaimer).



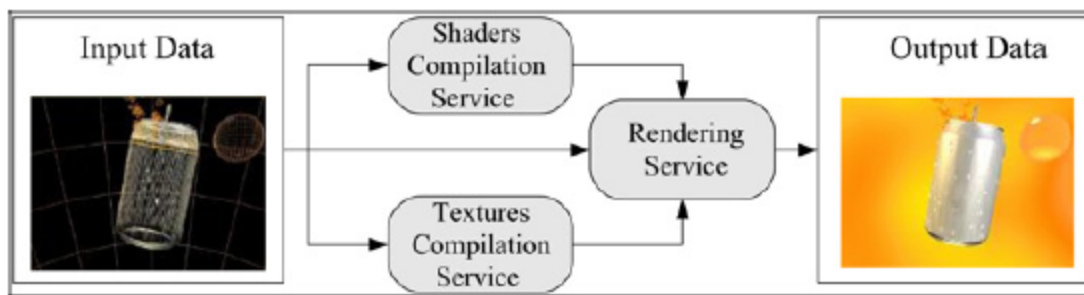
Σχήμα 15: Δομή SocIoS SLA Manager

Στις ακόλουθες παραγράφους, παρουσιάζονται παραδείγματα εφαρμογών ως υπηρεσίες που χρησιμοποιούνται σε ένα τέτοιο περιβάλλον, ώστε να γίνουν πιο κατανοητές οι παράμετροι υψηλού και χαμηλού επιπέδου που υπάρχουν στα SLAs και η σημασία τους. Στο πρώτο παράδειγμα (3D Rendering) έχουν περισσότερο νόημα όροι σε επίπεδο πόρων (T-SLA - Technical SLA), ενώ στο δεύτερο (Ανάλυση συναισθημάτων σε κοινωνικά δίκτυα) όροι σε υψηλό επίπεδο εφαρμογής (A-SLA - Application SLA).

3.4.2.1 3D Rendering

Η εφαρμογή τρισδιάστατης απόδοσης (3D Rendering) χρησιμοποιείται από επαγγελματίες που χρησιμοποιούν εξειδικευμένες εφαρμογές animation, με λεπτομερείς περιγραφές χώρου, αντικειμένων και σκιάσεων προκειμένου να παραχθούν εικόνες και βίντεο ανώτερης ποιότητας για την παραγωγή διαφημιστικών ή και άλλα επιχειρηματικά πεδία (πχ. σχεδίαση αρχιτεκτονικής κτηρίων). Δεδομένου ότι χρησιμοποιείται πληθώρα δεδομένων και η επεξεργασία απαιτεί σημαντικό φόρτο εργασίας από τον πάροχο της εφαρμογής, εμφανίζονται προκλήσεις ποιότητας και απόδοσης.

Μια γενική εικόνα της λειτουργίας μιας τέτοιας εφαρμογής φαίνεται στο Σχήμα 16 (Menychtas, 2009), με τα διακριτά στάδια επεξεργασίας που χρειάζονται για την τελική απόδοση. Σαν είσοδος στην εφαρμογή απόδοσης (renderer) δίνονται διάφορα χαρακτηριστικά μιας απεικόνισης, ή ενός συνόλου απεικονίσεων (στην περίπτωση ενός βίντεο), που περιγράφουν λεπτομέρειες φωτισμού και υφής, οι οποίες παρέχονται αρχικά στις αντίστοιχες υπηρεσίες, που προωθούν το αποτέλεσμα της επεξεργασίας τους στην τελική υπηρεσία Rendering. Ωστόσο, όσον αφορά τον τελικό χρήστη που δεν ενδιαφέρεται για τις λεπτομέρειες λειτουργίας, υπηρεσία είναι μόνο η συνολική εφαρμογή 3D Rendering.



Σχήμα 16: Στάδια επεξεργασίας εφαρμογής 3D Rendering

Η εφαρμογή 3D Rendering που θα χρησιμοποιηθεί και παρακάτω στα πειράματά μας βασίζεται σε μια διανομή του 3Delight® renderer (3Delight RenderMan - compliant renderer, 2008). Το 3Delight είναι μια γνωστή εμπορική εφαρμογή για 3D Rendering με μια ανοιχτή άδεια. Η παροχή μιας τέτοιας εφαρμογής ως υπηρεσία επιτυγχάνεται με τη χρήση περιγραφών που ακολουθούν τη μορφή RenderMan Bytestream Interface (RIB) (RenderMan Interface Specification v3.2, 2005), μια πλήρη προδιαγραφή των απαιτούμενων διεπαφών μεταξύ των μοντέλων περιγραφών και των renderers. Πιο αναλυτικά, η διαδικασία 3D Rendering επεκτείνεται με την προσθήκη ενός ενδιάμεσου

βήματος, όπου ένα αρχείο RIB παράγεται για να περιγράψει τη σκηνή που θα αποδοθεί. Τα αρχεία σε μορφή RIB είναι σημαντικά μικρότερα σε μέγεθος και ως εκ τούτου πιο εύκολο να ανταλλάσσονται μέσω του δικτύου. Από την άλλη πλευρά, είναι πιο δύσκολα διαχειρίσιμα, εφόσον είναι πολλά και έχουν μεγάλο αριθμό διασυνδέσεων. Κατόπιν του βήματος αυτού οι καθορισμένοι renderers είναι σε θέση να ανακατασκευάσουν και να αποδώσουν σωστά την εικόνα χρησιμοποιώντας τα ενδιάμεσα αρχεία RIB.

Για να δώσουμε μια ιδέα του φόρτου εργασίας που παράγει μια τέτοια επεξεργασία, τρέξαμε ένα από τα απλά παραδείγματα που είναι διαθέσιμα μέσω του διαδικτύου για το 3Delight, δίνοντας ως είσοδο ένα αρχείο "cornell.rib" και μια προκαθορισμένη ανάλυση (320x240) για μια μόνο απεικόνιση σε ένα τοπικό cluster. Χρησιμοποιώντας μια εικονική μηχανή Xen με 1 επεξεργαστή (2.8GHz) και 200 KB μνήμη, 10 δευτερόλεπτα χρειάστηκαν για να ολοκληρωθεί η συγκεκριμένη απόδοση.

Γίνεται σαφές ωστόσο ότι με βάση τους πόρους που αποδίδονται σε κάθε νέα εργασία μπορούμε ανάλογα να προσφέρουμε διαφορετική ποιότητα υπηρεσίας στον πελάτη. Για παράδειγμα μέγιστη δέσμευση πόρων (πχ. μνήμη) πιθανόν να οδηγεί σε ολοκλήρωση της εργασίας (και άρα απόκριση της υπηρεσίας) γρηγορότερα. Επίσης, ανάλογα με την πολυπλοκότητα του παρεχόμενου 3D μοντέλου και τους συνολικά διαθέσιμους πόρους, κάποιοι renderers μπορεί να επιστρέψουν βίντεο με χαμένα καρέ ή λάθη στον ήχο. Συνεπώς οι παράμετροι ποιότητας που πρέπει να συμφωνηθούν για μια τέτοια υπηρεσία είναι κυρίως χαμηλού επιπέδου (επίπεδο πόρων), με βάση την εμπειρία αλλά και τη χρέωση που επιθυμεί ο τελικός χρήστης, ο οποίος χρειάζεται να έχει μια γενική γνώση της εφαρμογής. Ένα παράδειγμα τέτοιου SLA δίδεται παρακάτω:


```

<?xml version="1.0"?>
<slaTemplate>
  <label>Sample SLA Template</label>
  <description>Limits instantaneous use to 2 CPUs</description>
  <billingPeriod>
    <years>0</years>
    <months>0</months>
    <days>1</days>
    <hours>0</hours>
    <minutes>0</minutes>
    <seconds>0</seconds>
  </billingPeriod>
  <signingFee>10.00</signingFee>
  <subscriptionFee>10.00</subscriptionFee>
  <currency>EUR</currency>
  <startTime>
    <year>2010</year>
    <month>6</month>
    <dayOfMonth>1</dayOfMonth>
  </startTime>
  <endTime>
    <year>2011</year>
    <month>6</month>
    <dayOfMonth>1</dayOfMonth>
  </endTime>
  <permittedServices/>
  <constraints>
    <constraint type='INSTANTANEOUS'>
      <metric type='RESOURCE'>
        <uri>http://www.gria.org/sla/metric/resource/cpu</uri>
        <description>CPU</description>
        <units type='DECIMAL'>
          <instantaneous>CPU</instantaneous>
        </units>
      </metric>
      <bound>LE</bound>
      <private>>false</private>
      <limit>2.0</limit>
      <contention>1.0</contention>
      <repeating>>false</repeating>
    </constraint>
  </constraints>
  <pricingTerms>
    <pricingTerm type='INSTANTANEOUS_INCREASE'>
      <description>creation charge</description>
      <lowerBound>0</lowerBound>
      <upperBound>-1</upperBound>
      <price>100</price>
      <metric type='ACTIVITY'>
        <uri>http://www.gria.org/sla/metric/activity/job</uri>
        <description>job</description>
        <units type='DECIMAL'>
          <instantaneous>job</instantaneous>
        </units>
      </metric>
    </pricingTerm>
  </pricingTerms>
</slaTemplate>

```

Πίνακας 5: SLA Template στο GRIA για υπηρεσία 3D Rendering

Στο υπόδειγμα ενός προτύπου SLA (SLA Template) του μεσολογισμικού GRIA που δόθηκε, το οποίο βασίζεται στο WS-Agreement, προτείνεται ο περιορισμός της στιγμιαίας χρήσης των επεξεργαστών (CPUs) μέχρι δύο από την εφαρμογή. Όπως φαίνεται στην αρχή του εγγράφου, καθορίζεται η περίοδος χρέωσης (ανά ημέρα), η αρχική χρέωση του πελάτη και η διάρκεια ζωής της συμφωνίας. Εν συνεχεία τίθεται ο προαναφερθείς περιορισμός και καθορίζεται η χρέωση για κάθε νέα δραστηριότητα (υποβολή εργασίας), αφού το T-SLA αυτό είναι μακράς διάρκειας.

3.4.2.2 Ανάλυση συναισθημάτων σε κοινωνικά δίκτυα

Μια ιδιαίτερα χρήσιμη υπηρεσία που εμφανίζεται στο αναδυόμενο «Διαδίκτυο των Υπηρεσιών» είναι αυτή της ανάλυσης συναισθημάτων (Sentiment Analysis) στα μέσα κοινωνικής δικτύωσης. Για να δοθεί μια ιδέα της τεράστιας δυναμικής τους, το Twitter που αποτελεί ένα από τα πιο γνωστά κοινωνικά δίκτυα περιλαμβάνει περίπου 180 εκατομμύρια χρήστες που δημοσιεύουν περισσότερα από 1 δισεκατομμύρια μηνύματα (tweets) την εβδομάδα. Ένα μεγάλο μέρος αυτών των μηνυμάτων αντιπροσωπεύει υποκειμενικές πληροφορίες, που περιέχουν απόψεις και συναισθήματα πάνω σε διάφορα θέματα ενδιαφέροντος. Έτσι, τέτοια δίκτυα μπορούν να παρέχουν πολύτιμες πληροφορίες για μια σειρά από θέματα που κυμαίνονται από τον αντίκτυπο ενός προϊόντος έως και τις απόψεις τις κοινωνίας για θέματα πολιτικής. Είναι σαφές λοιπόν η χρησιμότητα της ανάλυσης για παράδειγμα από δημοσιογράφους ή εταιρίες στατιστικής για την εξαγωγή χρήσιμων πληροφοριών και συμπερασμάτων.

Το περιεχόμενο των κοινωνικών δικτύων ωστόσο θέτει σοβαρές προκλήσεις για τη δυνατότητα εφαρμογής μιας τέτοιας ανάλυσης, λόγω των εγγενών χαρακτηριστικών τους. Τέτοια χαρακτηριστικά είναι το μέγεθος του κειμένου (σύντομο κείμενο ελεύθερης

μορφής), οι νεολογισμοί που χρησιμοποιούνται (ανεπίσημο κείμενο), ο θόρυβος (ανορθόγραφο κείμενο) και η πολυγλωσσία. Συνεπώς μια υπηρεσία ανάλυσης αυτού του περιεχομένου απαιτεί ιδιαίτερη πολυπλοκότητα (Aisopos, Papadakis, Tserpes, & Varvarigou, 2012) και μάλιστα σε εκτενή σύνολα μηνυμάτων (πιθανώς και εκατομμυρίων), κάτι που συνεπάγεται ανάγκη μεγάλης υπολογιστικής ισχύος, ώστε η ανάλυση αυτή να είναι αξιόπιστη.

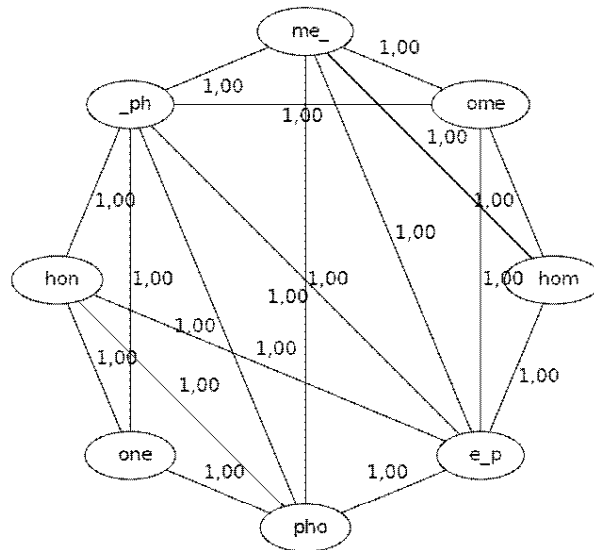
Στα πλαίσια μιας υπηρεσίας Sentiment Analysis, η ανάλυση επικεντρώνεται στον εντοπισμό συναισθηματικής "πόλωσης" μεμονωμένων μηνυμάτων, όπου πληθώρα τέτοιων μηνυμάτων σχετικών με ένα συγκεκριμένο θεματικό πεδίο (πχ. πολιτική) κατατάσσεται σε κατηγορίες με βάση τις παρακάτω διαιρέσεις (ανά περίπτωση):

- Δυαδική κατάταξη: Ταξινόμηση ενός συνόλου μηνυμάτων σε δύο κατηγορίες συναισθηματικής πόλωσης (θετικά, αρνητικά).
- Γενική κατάταξη: Ταξινόμηση ενός συνόλου μηνυμάτων σε τρεις κατηγορίες συναισθηματικής πόλωσης (θετικά, αρνητικά, ουδέτερα).

Για την κατάταξη των μηνυμάτων στις κατηγορίες που αναφέρονται παραπάνω, είναι απαραίτητη η εξαγωγή των διαφόρων χαρακτηριστικών του κάθε μηνύματος και η τροφοδότησή τους στην ανοικτού κώδικα βιβλιοθήκη της Weka που μπορεί να υλοποιήσει διαφόρους αλγορίθμους κατάταξης (Naive Bayes, Support Vector Machines, C4.5) για την προαναφερθείσα ανάλυση.

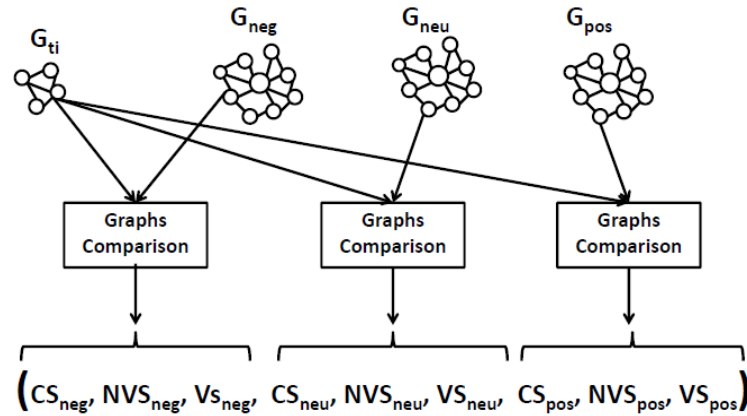
Μια κατηγορία χαρακτηριστικών είναι αυτά που βασίζονται μόνο στο περιεχόμενο ενός μηνύματος (Content-based features). Τέτοια μπορεί να είναι ο αριθμός των ειδικών χαρακτήρων ή ορισμένων σημείων στίξης, καθώς και των κεφαλαίων γραμμάτων σε ένα μήνυμα. Μια πιο προηγμένη ομάδα χαρακτηριστικών που

χρησιμοποιείται σε αυτό το πεδίο είναι αυτή που προκύπτει με την τεχνική των γράφων ν-γραμμάτων (Aisopos, Papadakis, & Varvarigou, 2011). Το σύνολο των ν-γραμμάτων ενός μηνύματος περιλαμβάνει όλες τις συμβολοσειρές μήκους ν μέσα στο κείμενό του. Ο γράφος ν-γραμμάτων συνδέει επιπλέον με βάρη όλα τα γειτονικά ν-γράμματα, με τις ακμές αυτές να υποδηλώνουν το ποσοστό συνύπαρξής τους σε ένα μήνυμα. Για παράδειγμα για τη φράση "home_phone" εξάγεται ο παρακάτω γράφος:



Σχήμα 17: Παράδειγμα γράφου ν-γραμμάτων "home_phone" για την εξαγωγή χαρακτηριστικών ανάλυσης

Έτσι, κατασκευάζοντας το γράφο για κάθε tweet (G_{ti}) και συγκρίνοντας τον με τους συνολικούς γράφους θετικών (G_{pos}), αρνητικών (G_{neg}) ή ουδέτερων (G_{neu}) μηνυμάτων που δημιουργούνται από ένα training set, με βάση διάφορα κριτήρια ομοιότητας (ομοιότητα συνοχής - CS, ομοιότητα τιμών - VS, κανονικοποιημένη ομοιότητα τιμών - NVS) εξάγονται πιο σύνθετα χαρακτηριστικά για τους αλγόριθμους κατάταξης.



Σχήμα 18: Εξαγωγή χαρακτηριστικών ομοιότητας γράφων μηνυμάτων

Η μέθοδος των γράφων ν-γραμμμάτων εισάγει ιδιαίτερα εμπειριστατωμένα χαρακτηριστικά στην ανάλυση, ωστόσο μια ακριβέστερη κατάταξη των συναισθημάτων χρειάζεται και χαρακτηριστικά που βασίζονται στο κοινωνικό πλαίσιο ενός μηνύματος (Social context-based features). Τέτοια μπορεί να είναι ο αριθμός προηγούμενων θετικών ή αρνητικών μηνυμάτων του συγγραφέα ενός tweet και των φίλων του, ο αριθμός θετικών ή αρνητικών μηνυμάτων για το συγκεκριμένο θέμα, ή ο αριθμός των αναφορών (mentions) σε αυτό το μήνυμα. Μια ολοκληρωμένη υπηρεσία ανάλυσης συναισθημάτων, με τάξη αποτελεσματικότητας πάνω από 95%, χρησιμοποιεί και τις δύο αυτές κατηγορίες χαρακτηριστικών (Aisopos F., Papadakis, Tserpes, & Varvarigou, 2012), καθιστώντας την πολυπλοκότητα κατάταξης ιδιαίτερα υψηλή και την όλη διαδικασία χρονοβόρα. Σε μια τέτοια περίπτωση υπηρεσίας, η υπό διαπραγμάτευση ποιότητα υπηρεσίας έγκειται στην αξιοπιστία κατάταξης (χρήση χαρακτηριστικών περιεχομένου ή/και κοινωνικού πλαισίου) καθώς και στο χρόνο απόκρισης. Εδώ εμφανώς παράμετροι υψηλού επιπέδου όπως οι προηγούμενες έχουν περισσότερο νόημα από ότι θα είχαν όροι επιπέδου πόρων. Παρακάτω ακολουθεί ένα παράδειγμα αντίστοιχου SLA της πλατφόρμας SocIoS:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:AgreementOffer xmlns:ns2="http://schemas.ggf.org/graap/2007/03/ws-
agreement" xmlns:ns4="http://positivespaces2.grid.ece.ntua.gr/wsagreement"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ns2:AgreementTemplateType">
  <ns2:Name>SocIoS Services Gold Offer</ns2:Name>
  <ns2:Context>
    <AgreementInitiator xsi:type="ns2:AgreementInitiatorIdentifierType">
      <Reference xmlns:xs="http://www.w3.org/2001/XMLSchema"
xsi:type="xs:string">NTUA</Reference>
    </AgreementInitiator>
    <ExpirationTime>2013-05-20T19:06:06.093+03:00</ExpirationTime>
    <ns4:Disclaimer>
      By consenting to this service level agreement, you will be given access to
      the service offered through the SocIoS prototype, provided by the partners
      of the FP7 SocIoS consortium funded under Grant Agreement No 257774.
    </ns4:Disclaimer>
    <ns4:templateCategory>Gold</ns4:templateCategory>
  </ns2:Context>
  <ns2:Terms>
    <ns2:All>
      <ServiceDescriptionTerm ServiceName="SentimentAnalysis"
Name="contextFeaturesUsed">
        <ns4:contextFeatures>true</ns4:contextFeatures>
      </ServiceDescriptionTerm>
      <ServiceDescriptionTerm ServiceName="SentimentAnalysis"
Name="nGramsSizeOfn">
        <ns4:nGramsSize>4</ns4:nGramsSize>
      </ServiceDescriptionTerm>
      <GuaranteeTerm>
        <ns2:QualifyingCondition
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xsi:type="xs:string">day</ns2:QualifyingCondition>
        <ns2:ServiceLevelObjective>
          <CustomServiceLevel>
            <ns4:numberOfTweets>1000000</ns4:numberOfTweets>
          </CustomServiceLevel>
        </ns2:ServiceLevelObjective>
        <BusinessValueList>
          <Reward>
            <AssessmentInterval>
              <Count>500000</Count>
            </AssessmentInterval>
            <ValueUnit>EUR</ValueUnit>
            <ValueExpression
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xsi:type="xs:int">500</ValueExpression>
          </Reward>
          <Penalty>
            <ValueUnit>EUR</ValueUnit>
            <ValueExpression
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xsi:type="xs:int">300</ValueExpression>
          </Penalty>
        </BusinessValueList>
      </GuaranteeTerm>
    </ns2:All>
  </ns2:Terms>
</ns2:AgreementOffer>

```

Πίνακας 6: SLA Template κατηγορίας Gold στο SocIoS για υπηρεσία ανάλυσης συναισθημάτων

Το υπόδειγμα προτύπου A-SLA στην πλατφόρμα του SocIoS ακολουθεί και επεκτείνει το πρότυπο WS-Agreement, για μια τέτοια υπηρεσία ανάλυσης. Σε αυτό το SLA Template, αρχικά παρέχονται οι γενικές πληροφορίες της συμφωνίας (όνομα, πάροχος υπηρεσίας, διάρκεια), εν συνεχεία φαίνονται επιπλέον πεδία που προστέθηκαν ως επέκταση (Disclaimer, templateCategory) και τέλος οι όροι περιγραφής της υπηρεσίας (ServiceDescriptionTerms) και οι όροι χρέωσης. Εφόσον η συμφωνία είναι κατηγορίας Gold, προτείνεται ανάλυση με χαρακτηριστικά και κοινωνικού πλαισίου και περιεχομένου, όπου στους γράφους n -γραμμάτων τίθεται $n=4$ για μέγιστη απόδοση, έναντι αντίστοιχου αντιτίμου (Reward).

4

Αξιολόγηση Ποιότητας Υπηρεσίας σε SaaS

Σε αυτό το κεφάλαιο, παρουσιάζονται όλες οι τεχνολογίες και μεθοδολογίες αιχμής που ασχολούνται με την αξιολόγηση της Ποιότητας Υπηρεσίας σε κατανεμημένα Software as a Service περιβάλλοντα, μπαίνοντας πλέον στην ουσία της παρούσας διατριβής. Η αξιολόγηση της ποιότητας τέτοιων υπηρεσιών έχει αποτελέσει για πολύ καιρό τώρα κεντρικό θέμα στην έρευνα από διάφορες σκοπιές. Εδώ, μελετώνται λύσεις που έχουν προταθεί σε διάφορες καταστάσεις ως προς το θέμα της αξιολόγησης της ποιότητας των υπηρεσιών, εξηγώντας την έννοια της "ποιότητας της εμπειρίας" και το πώς χρησιμοποιείται για την αξιολόγηση, και περιγράφοντας τις τεχνικές που έχουν προταθεί για την πλευρά των πελατών, καθώς και για την αντίστοιχη πλευρά του παρόχου υπηρεσιών (Aisopos, Kardara, Moulos, Papaoikonomou, Tserpes, & Varvarigou, 2012).

4.1

Αξιολόγηση όρων χαμηλού

επιπέδου

Σε υπηρεσιοστρεφή περιβάλλοντα παροχής εφαρμογών ως υπηρεσίες, ο πάροχος υπηρεσιών συνήθως δεν είναι προσανατολισμένος μόνο σε μια συγκεκριμένη εφαρμογή. Συνεπώς είναι κατά πάσα πιθανότητα απρόθυμος να αποδεχτεί ένα μοντέλο απεικόνισης της ποιότητας υπηρεσίας υψηλού επιπέδου. Στις περισσότερες περιπτώσεις οι όροι του SLA είναι σε επίπεδο πόρων, κάτι το οποίο δημιουργεί μια σημαντική αδυναμία στη διαδικασία αξιολόγησης: Ο τυπικός κύκλος ζωής ενός SLA σε SaaS περιλαμβάνει την παροχή, τη διαπραγμάτευση, τη συμφωνία, την παρακολούθηση και την αξιολόγηση. Σε όλον αυτό τον κύκλο ζωής και τα δύο μέρη έχουν πλήρη έλεγχο του SLA και των όρων του, εκτός από το στάδιο της παρακολούθησης. Μόνο ο πάροχος έχει πρόσβαση στους πόρους του και επιβεβαιώνει απευθείας τη χρήση τους σύμφωνα με ό,τι έχει υποσχεθεί στο SLA. Αυτό δημιουργεί ένα ζήτημα εμπιστοσύνης, αφού μόνο ο πάροχος μπορεί να αξιολογήσει πραγματικά την ποιότητα, αφήνοντας τον καταναλωτή εκτός παιχνιδιού (Tserpes, Kyriazis, Menychtas, & Varvarigou, 2008).

Μια πρόταση για την εισαγωγή ενός αξιόπιστου τρίτου οργανισμού έχει γίνει από κάποιους, όπως για παράδειγμα από τους Molina-Jimenez, Shrivastava, Crowcroft, & Gevros (2004), Overton (2002), Keller & Ludwig (2002), όπου η παρακολούθηση και αξιολόγηση των υπηρεσιών μπορεί να παρέχεται από μια εξειδικευμένη οντότητα. Το ζήτημα στην προκειμένη περίπτωση είναι αν υπάρχει κάποιος τρίτος που ο πελάτης θα εμπιστευόταν για αυτή τη διαδικασία και που την ίδια στιγμή ο πάροχος θα του έδινε πρόσβαση στους πόρους του. Αυτή ακριβώς είναι η αδυναμία μιας κατά τα άλλα ιδανικής λύσης. Ιδανική επειδή η αντιστοίχιση μεταξύ QoS και QoBiz θα γινόταν από

έναν άλλο οργανισμό που εγγυάται ότι αυτό το κάνει σωστά και δίκαια, αλλά δεν είναι βιώσιμη, διότι είναι σχεδόν αδύνατο να βρεθεί μια τέτοια αξιόπιστη τρίτη οντότητα.

Λαμβάνοντας υπόψη τα παραπάνω, η κοινή πρακτική όσον αφορά την αξιολόγηση του μοντέλου ποιότητας είναι συχνά με χρήση όρων SLA χαμηλού επιπέδου, αφήνοντας τον πελάτη να στηριχτεί στη δική του εμπειρία για την επιβεβαίωση της ειλικρίνειας του παρόχου. Η παρακάτω ενότητα αναλύει περαιτέρω το θέμα.

4.2 Ποιότητα Εμπειρίας του χρήστη

Τα κοινά υπηρεσιοστρεφή συστήματα επιζητούν τη χρήση μοντέλων σε επίπεδο πόρων για παρακολούθηση της ποιότητας, ωστόσο, υπάρχουν αρκετοί ερευνητές ισχυρίζονται το αντίθετο. Στις ερευνητικές δουλειές που εκπόνησαν οι Tserpes, Kyriazis, Menychtas, Litke, Christogiannis, & Varvarigou (2008), Goiri, Julià, Fitó, Macías, & Guitart (2010), Gu, Nahrstedt, Yuan, Wichadakul, & Xu (2002) και Lamanna, Skene, & Emmerich (2003), αναφέρεται ότι η υιοθέτηση μιας κοινής "γλώσσας" για τον καθορισμό της ποιότητας μεταξύ του πελάτη και του παρόχου είναι αναγκαία. Ο μόνος τρόπος για να συμβεί αυτό είναι η χρησιμοποίηση όρων επιπέδου εφαρμογής (μετρικών QoBiz, ή όπως μερικές φορές ονομάζονται "μετρικές σε επίπεδο υπηρεσίας"), μεταθέτοντας το βάρος στον πάροχο να τις μεταφράσει σε πιο απτές παραμέτρους σε επίπεδο πόρων. Υπάρχουν διάφοροι τρόποι για αυτή τη "μετάφραση" και οι περισσότεροι από αυτούς αφορούν την παραγωγή ενός ενδιάμεσου μοντέλου απόδοσης που μπορεί να δημιουργηθεί χρησιμοποιώντας τεχνικές ασαφούς λογικής (π.χ. . (Varvarigou, Tserpes, Kyriazis, Silvestri, & Psimogiannos, 2010)) ή ημι-στατική συσχέτιση QoS-QoBiz όπως παρουσιάζεται από τους Kim, Lee, Lee, Lee, LYU, & Choi (2008).

Παρόλα αυτά, στις περισσότερες των περιπτώσεων, ο πελάτης πρέπει να συνήθως να στραφεί στη δικιά του εμπειρία, βάσει της εξειδίκευσής του, προκειμένου να αξιολογήσει την παρεχόμενη ποιότητα. Η διαδικασία αυτή σχετίζεται με την υποκειμενική αντίληψη του πελάτη για την ποιότητα. Αυτό συνήθως αναφέρεται ως Ποιότητα Εμπειρίας του χρήστη (Quality of user Experience - QoE) και παρόλο που δεν μπορεί να έχει κάποια επίπτωση στο ίδιο το SLA, έχει άμεση επίπτωση στην αξιολόγηση της φήμης του παρόχου από τον πελάτη. Ωστόσο η αξιολόγηση του παρόχου είναι μοναδική για τον πελάτη που την διεξάγει, εφόσον οι απαιτήσεις καθώς και η εμπειρία στον τομέα ποικίλλει από χρήστη σε χρήστη (Li-yuan, Wen-an, & Jun-de, 2006). Επιπλέον, δεν υπάρχει τρόπος να ενσωματωθούν QoE όροι στα SLAs, καθώς δεν υπάρχει άμεση αντιστοιχία για την μετατροπή τους σε όρους επιπέδου πόρων.

Ως εκ τούτου, υπάρχουν αρκετές προτάσεις για τη δημιουργία μηχανισμών αξιολόγησης με βάση το ιστορικό, που καταγράφουν τη φήμη των παρόχων και παρέχουν υποστήριξη αποφάσεων για τη μελλοντική επιλογή υπηρεσιών. Όμοια, υπάρχουν μια σειρά προτάσεων και εφαρμογών που σχετίζονται με την αντικειμενική χρήση της QoE αξιολόγησης από περισσότερους από έναν πελάτες της υπηρεσίας. Οι πιο σημαντικές παρουσιάζονται στις επόμενες παραγράφους.

4.3

Η μεριά του πελάτη

Αξιοποιώντας την έννοια του QoE, οι πελάτες των υπηρεσιών μπορούν να αξιολογήσουν το QoS που προσφέρεται από ανταγωνιστικούς παρόχους οι ίδιοι, ή μέσω συστημάτων υποστήριξης τρίτων οργανισμών, που κάνουν συστάσεις με βάση τις προηγούμενες κλήσεις (πιθανόν από διαφορετικούς πελάτες) της κάθε υπηρεσίας, καθώς και τις

προτιμήσεις του κάθε πελάτη. Έτσι, σε γενικές γραμμές η παρακολούθηση του QoS από την πλευρά του πελάτη γίνεται μέσω:

- Ενός ενδιάμεσου που παρεμβάλλεται στην επικοινωνία μεταξύ παρόχου και του καταναλωτή υπηρεσιών και συλλέγει στοιχεία σχετικά με την απόδοση των υπηρεσιών. Οι πληροφορίες που αναλύονται εδώ τείνουν να είναι πιο αντικειμενικές, ωστόσο αυτή η λύση μπορεί να είναι ακριβή αν αυτό γίνεται σε μεγάλη κλίμακα, κι επίσης μπορεί να υπάρχουν προβλήματα καθυστερήσεων.
- Συστημάτων δημιουργίας φήμης (Reputation) τρίτων οργανισμών, στα οποία όλοι οι χρήστες συμμετέχουν και τα ανατροφοδοτούν. Οι χρήστες τέτοιων συστημάτων πρέπει να είναι σε θέση να αξιολογήσουν με κάποιο τρόπο το QoS που λαμβάνουν, με την αξιοπιστία των συμμετεχόντων εδώ να είναι το πιο σημαντικό θέμα που πρέπει να επιλυθεί.

Φυσικά, δεδομένου ότι μία τρίτη υπηρεσία εμπλέκεται στη διαδικασία αξιολόγησης / επιλογής, προκύπτουν θέματα εμπιστοσύνης και ασφάλειας, τα οποία επίσης πρέπει να αντιμετωπιστούν, όπως θα συζητηθεί παρακάτω.

4.3.1 Χρήση πληρεξούσιας οντότητας

Οι Maximilien & Singh (2002) παρουσίασαν για πρώτη φορά το εννοιολογικό μοντέλο μιας υπηρεσίας Reputation, χρησιμοποιώντας μια πληρεξούσια οντότητα (Web Service Agent Proxy - WSAP) για τους πελάτες των διαδικτυακών υπηρεσιών. Σε αυτό το μοντέλο, όλες οι επικοινωνίες με τα μητρώα UDDI για την εύρεση μιας υπηρεσίας συμβαίνουν μέσω του WSAP. Με αυτό τον τρόπο το WSAP, το οποίο λειτουργεί εξωτερικά από τα μητρώα και τις οντότητες που χρησιμοποιούνται για Reputation,

καταγράφει οποιαδήποτε ανατροφοδότηση του πελάτη και βοηθά στην εύρεση των κατάλληλων παρόχων, με την εκμάθηση και διάδοση της γνώσης του σε εξωτερικούς οργανισμούς και άλλα WSAPs. Έτσι, αυτή η πληρεξούσια οντότητα υλοποιεί τελικά ένα σύστημα φήμης, διευκολύνοντας τους πελάτες κατά τη διάρκεια της διαδικασίας επιλογής υπηρεσίας.

Μια παρόμοια προσέγγιση προσανατολισμένη σε πληρεξούσιο (proxy) ακολουθήθηκε επίσης από τους Lo & Wang (2007), για τη δημιουργία ενός πλαισίου αξιολόγησης και επιλογής (CosmosQoS) διαδικτυακών υπηρεσιών με βάση την ποιότητα. Ο στόχος ήταν πάλι η καλύτερη υποστήριξη των αιτούντων για υπηρεσίες, έτσι ώστε να μπορούν να επιλέγουν τις πιο ενδεδειγμένες, βάσει των QoS απαιτήσεών τους, χρησιμοποιώντας ένα πιο συγκεντρωτικό μηχανισμό αξιολόγησης της φήμης των υπηρεσιών, με έναν καθολικό proxy. Οι Cao, Huang, Wang, & Gu (2009) πρότειναν ένα μοντέλο αξιολόγησης βασισμένο στο QoS και τις προτιμήσεις των χρηστών που ονομάζεται Q-WSEM (QoS based Web Service Evaluation Model), εισάγοντας ένα εξωτερικό Κέντρο Εξυπηρέτησης για την υποστήριξη της αξιολόγησης της ποιότητας των διαδικτυακών υπηρεσιών. Η οντότητα αυτή συλλέγει QoS πληροφορίες για τις υποψήφιες διαδικτυακές υπηρεσίες που μπορούν να παρέχουν την απαιτούμενη λειτουργικότητα στους χρήστες και στη συνέχεια αποθηκεύει και ενημερώνει τις πληροφορίες QoS στα μητρώα UDDI. Παρέχει επίσης τη δυνατότητα εισαγωγής περιορισμών από το χρήστη και διαμόρφωσης με βάση τις προτιμήσεις του, και εκτελεί το φιλτράρισμα και την αξιολόγηση των υπηρεσιών, λαμβάνοντας υπόψη ποιοτικά και ποσοτικά κριτήρια.

4.3.2 Χρήση Σημασιολογιών

Μια σημαντική καινοτομία του μοντέλου WSAP που αναφέρθηκε παραπάνω ήταν η χρήση σημασιολογιών (semantics) για να περιγραφούν με πιο συγκεκριμένο τρόπο τα χαρακτηριστικά των διαδικτυακών υπηρεσιών. Semantics χρησιμοποιήθηκαν επίσης και από τους Wang, Vitvar, Kerrigan, & Toma (2006) για τη δημιουργία ενός μοντέλου επιλογής υπηρεσίας, βασισμένο σε μια QoS οντολογία που καθορίστηκε με τη χρήση του Web Services Modeling Ontology (WSMO), το οποίο φιλτράρει τις διαθέσιμες υπηρεσίες χρησιμοποιώντας μετρικές ποιότητας σε ένα συνδυαστικό αλγόριθμο επιλογής. Οι Muñoz Frutos, Kotsiopoulos, Vaquero Gonzalez, & Rodero Merino (2009) επέκτειναν το παραπάνω σε ένα εννοιολογικό μοντέλο ποιότητας, το οποίο διαμορφώθηκε ως μια QoS οντολογία με έναν αλγόριθμο επιλογής, ενώ οι Chaari, Badr, & Biennier (2008) εφάρμοσαν νέες οντολογικές αντιλήψεις και επέκτειναν την προδιαγραφή του WS-Policy (Bajaj, Box, & Chappell, 2006) που αναφέρθηκε και στην εισαγωγή, προκειμένου να συμπεριλάβουν και πολιτικές ποιότητας, ώστε να παρέχουν QoS υποστήριξη κατά την επιλογή των υπηρεσιών.

Οι Maximilien & Singh (2004) διερεύνησαν περαιτέρω την επιλογή και τη δημιουργία εμπιστοσύνης με αυτόνομα Web Services, προτείνοντας μια προσέγγιση με πολλαπλούς agents που χρησιμοποιούν QoS μετρικές, με μια οντολογία QoS που επιτρέπει το ταίριασμα των υπηρεσιών σημασιολογικά και δυναμικά. Ένας απλός αλγόριθμος δημιουργήθηκε για αυτό το ταίριασμα, έτσι ώστε να εξισορροπούνται οι πολιτικές των καταναλωτών με τις διαφημιζόμενες πολιτικές των παρόχων υπηρεσιών, με βάση μη λειτουργικές ιδιότητες, όπως η φήμη της ποιότητας μιας υπηρεσίας.

Οι Wang & Vassileva (2007) επέκτειναν τη δημιουργία εμπιστοσύνης κατά την επιλογή διαδικτυακών υπηρεσιών βασιζόμενοι σε χαρακτηριστικά όπως το QoS, το κόστος, κ.λπ., έχοντας ένα κεντρικό μητρώο για τη συλλογή και αποθήκευση των πληροφοριών QoS και ένα τρίτο μέρος για να εποπτεύει την όλη διαδικασία. Τέλος, οι (Serhani, Dssouli, Hafid, & Sahraoui, 2005) παρουσίασαν μια αρχιτεκτονική βασισμένη σε έναν ενδιάμεσο (broker), έχοντας έναν QoS broker να εκτελεί μια σημασιολογική επαλήθευση της περιγραφής των παραμέτρων ποιότητας κάθε υπηρεσίας, συγκρίνοντας τις τιμές των μετρικών με τις παρουσιαζόμενες, για τον έλεγχο της συμμόρφωσης των υπηρεσιών και την αξιολόγηση και επιλογή της κατάλληλης υπηρεσίας για τον πελάτη.

4.3.3 Αξιολόγηση ποιότητας κατά τη διαδικασία εξεύρεσης μιας υπηρεσίας

Η αξιολόγηση της ποιότητας των υπηρεσιών - και συνεπώς η επιλογή της κατάλληλης από τον πελάτη - μπορεί να ενσωματωθεί στη διαδικασία ανακάλυψης της επιθυμητής υπηρεσίας. Οι Ma, Wang, Li, Xie, & Liu (2008) συνδύασαν τεχνολογίες σημασιολογιών και Προγραμματισμό Περιορισμών (Constraint Programming - CP) για την ανακάλυψη υπηρεσίας λαμβάνοντας υπόψη περιορισμούς QoS. Το πλαίσιο ανακάλυψης αποτελείται από τρία στρώματα: Το πρώτο στρώμα είναι το σημασιολογικό (Semantic Matchmaking Layer), το οποίο χρησιμοποιεί Description Logic (DL) reasoners για να διασφαλίσει ότι τα QoS δεδομένα μιας διαφημιζόμενης διαδικτυακής υπηρεσίας είναι σημασιολογικά συμβατά με τις ανάγκες του χρήστη. Πάνω από αυτό, υπάρχει το στρώμα του Προγραμματισμού Περιορισμών, όπου οι περιορισμοί ποιότητας αντιμετωπίζονται ως πρόβλημα ικανοποίησης περιορισμών. Τέλος, υπάρχει το στρώμα επιλογής QoS, που επεξεργάζεται τη λίστα υποψηφίων υπηρεσιών που παρέχονται από το CP στρώμα, για να επιστραφεί η καλύτερη προσφορά σε μια απαίτηση QoS.

Οι García, Ruiz, Ruiz-Cortés, Martín-Díaz, & Resinas (2007) πρότειναν ένα υβριδικό μηχανισμό αναζήτησης υπηρεσίας, ο οποίος καθορίζει μια διαδικασία ανακάλυψης n σταδίων. Κάθε στάδιο εξυπηρετείται από τον καταλληλότερο μηχανισμό. Ειδικότερα, η ανακάλυψη με βάση την ποιότητα βασίζεται σε Constraint Programming, ενώ η λειτουργική ανακάλυψη γίνεται με DL reasoners.

4.3.3.1 Κεντρικές υποδομές αποθήκευσης των πληροφοριών QoS

Ένα βασικό πρόβλημα, όταν χρησιμοποιούνται μετρικές QoS κατά την ανακάλυψη των διαδικτυακών υπηρεσιών είναι το που αυτές οι πληροφορίες αποθηκεύονται και αξιολογούνται με βάση τις απαιτήσεις των χρηστών. Πολλοί ερευνητές, όπως ο Blum (2004) και οι Xu, Martin, Powley, & Zulkernine (2007) βασίστηκαν στα tModels του UDDI που παρουσιάστηκαν παραπάνω, για να συμπεριλάβουν QoS πληροφορίες σε κεντρικά μητρώα. Ο Ran (2003) εισήγαγε ένα νέο μοντέλο για την εξεύρεση υπηρεσιών με την προσθήκη ενός νέου στοιχείου που ονομάστηκε Πιστοποιητής Ποιότητας Υπηρεσίας (QoS Certifier). Αυτή η οντότητα πιστοποίησης επαληθεύει το διαφημιζόμενο QoS μιας διαδικτυακής υπηρεσίας. Αν η διαδικασία είναι επιτυχής, οι πάροχοι λαμβάνουν ένα αναγνωριστικό πιστοποίησης, που τους επιτρέπει να εγγραφούν στο μητρώο UDDI, ώστε οι πληροφορίες QoS για τις υπηρεσίες τους να μπορούν να ανακτηθούν από εκεί.

Οι Zhang, Zheng, & Lyu (2010) πρότειναν μια διαφορετική προσέγγιση, παρουσιάζοντας μια μηχανή αναζήτησης υπηρεσιών που ονομάζεται WSEXPRESS, η οποία κάνει τόσο την αξιολόγηση ποιότητας όσο και την επιλογή υπηρεσίας. Η διαδικασία αναζήτησης αποτελείται από τρία μέρη: Μη λειτουργική αξιολόγηση, λειτουργική αξιολόγηση και κατάταξη υπηρεσιών με βάση την ποιότητα. Για το μέρος

που αφορά τη λειτουργικότητα, το WSEXPRESS εκτελεί μια προ-επεξεργασία των WSDL όλων των διαθέσιμων υπηρεσιών, για τον εντοπισμό των χρήσιμων όρων (π.χ. παραμέτρων εισόδου / εξόδου) και στη συνέχεια εκτελεί έναν υπολογισμό της ομοιότητας με το ερώτημα του χρήστη. Για το μη-λειτουργικό μέρος, τεχνικές υπολογισμού χρησιμότητας παρέχουν μια λίστα των υποψήφιων Web Services που ικανοποιούν τις απαιτήσεις ποιότητας. Τέλος, τα αποτελέσματα αυτών των σταδίων συνδυάζονται σε μια λίστα υπηρεσιών που ικανοποιεί τόσο τις λειτουργικές όσο και τις μη λειτουργικές απαιτήσεις του χρήστη.

4.3.3.2 Κατανεμημένες υποδομές αποθήκευσης των πληροφοριών QoS

Οι Schlosser, Sintek, Decker, & Nejd (2002) πρότειναν μια ισότιμη (Peer to Peer - P2P) υποδομή αντί του συγκεντρωτικού μοντέλου του UDDI. Οι κόμβοι στο P2P δίκτυο οργανώνονται σε μια δομή βασισμένη σε υπερ-κύβους, με αυτούς που περιλαμβάνουν παρόμοιες υπηρεσίες να ομαδοποιούνται σε συστάδες λειτουργικότητας. Στην ιδανική περίπτωση, το P2P δίκτυο θα πρέπει να είναι σε θέση να δρομολογήσει μια αναζήτηση ενός χρήστη για μια υπηρεσία σε εκείνους τους κόμβους που μπορούν να απαντήσουν το συγκεκριμένο ερώτημα.

Μια παρόμοια P2P προσέγγιση ακολουθούν οι Vu, Hauswirth, & Aberer (2005), όπου όλες οι οντολογικές απεικονίσεις για τις εισόδους / εξόδους μιας διαδικτυακής υπηρεσίας σε ένα αίτημα, κατηγοριοποιούνται σε ομάδες οντοτήτων βάσει της σημασιολογικής ομοιότητάς τους. Κάθε σημασιολογική περιγραφή μιας υπηρεσίας (σε διαφήμιση ή σε αίτημα) συνδέεται με ένα διάνυσμα πολλαπλών κλειδιών που ονομάζεται "χαρακτηριστικό διάνυσμα", το οποίο χρησιμοποιείται για την κατανομή των

διαφημιζόμενων υπηρεσιών στα P2P μητρώα. Η σημασιολογική ανακάλυψη των υπηρεσιών σε αυτό το μοντέλο βασίζεται στα λεγόμενα φίλτρα Bloom (Bloom filters).

4.3.4 Σχήματα συνεργατικής αξιολόγησης ποιότητας από τους πελάτες

Οι οντολογικές προσεγγίσεις που αναφέρονται παραπάνω σχεδίασαν τα μοντέλα τους με βάση μόνο τα πρότυπα SLA που προσφέρουν οι πάροχοι και τα QoS χαρακτηριστικά στα έγγραφα WSDL, χωρίς να λαμβάνονται υπόψη άλλες απόψεις καταναλωτών. Τα περισσότερα reputation συστήματα που βασίζονται στην αξιολόγηση της ποιότητας από τη μεριά του πελάτη, ωστόσο, συλλέγουν και συνδυάζουν τις αξιολογήσεις από πολλούς χρήστες, λειτουργώντας εντελώς ανεξάρτητα από τους παρόχους υπηρεσιών, χωρίς οι τελευταίοι να γνωρίζουν την ύπαρξή τους.

Οι Wang, Lee, & Ho (2007) παρουσίασαν μια μέθοδο που λαμβάνει υπόψη όχι μόνο τα αντικειμενικά στοιχεία που περιγράφονται από τους παρόχους υπηρεσιών, αλλά και τις υποκειμενικές αξιολογήσεις από τους πελάτες που χρησιμοποιούν τις υπηρεσίες. Σε αυτή την εργασία, μια οντότητα που κάνει συστάσεις βασισμένες στην ποιότητα συγκεντρώνει QoS πληροφορίες από ένα μητρώο UDDI κι επίσης συλλέγει τις εκτιμήσεις των χρηστών, για τη σωστή υποστήριξη στην επιλογή των υπηρεσιών μέσω ενός γενετικού αλγορίθμου (Genetic algorithm - GA). Υποκειμενικές πληροφορίες μελετήθηκαν επίσης και από τους Ding Li, & Zhou (2008), που εργάστηκαν πάνω στην επιλογή διαδικτυακών υπηρεσιών με βάση τη φήμη σε ένα περιβάλλον πλέγματος, χρησιμοποιώντας έναν αξιολογητή που συλλέγει και ερμηνεύει τις παρατηρήσεις των πελατών για να τους βοηθήσει στην επιλογή του "καλύτερου" παρόχου υπηρεσιών από έναν κατάλογο.

Δεδομένου ότι η αξιοπιστία των υποκειμενικών αξιολογήσεων ποιότητας του πελάτη μπαίνει τώρα στο παιχνίδι, τα Συστήματα Υποστήριξης Αποφάσεων (Decision Support) θεωρούνται τα καταλληλότερα για την παροχή συστάσεων στη διαδικασία επιλογής υπηρεσιών. Οι Wu, Chang, & Thomson (2005) παρουσίασαν ένα τέτοιο σύστημα αποφάσεων με βάση την ποιότητα για κατανεμημένες αρχιτεκτονικές διαδικτυακών υπηρεσιών, χρησιμοποιώντας ένα Decision Support Systems Module (DSS Module) για την επιλογή του καταλληλότερου φορέα παροχής υπηρεσιών. Το ενσωματωμένο DSS Module σε αυτή τη λύση περιλαμβάνει μια οντότητα ανάλυσης ποιότητας και ένα μοντέλο εμπιστοσύνης, που χρησιμοποιούν ιστορικά στοιχεία από μια βάση QoS δεδομένων, παράγοντας ως έξοδο μετρικές αξιοπιστίας για τις εμπλεκόμενες υπηρεσίες, ώστε να παρέχει την τελική λίστα κατάταξης των υπηρεσιών. Οι Tong & Zhang (2006) πρότειναν έναν παρόμοιο αλγόριθμο αποφάσεων πολλαπλών χαρακτηριστικών για επιλογή υπηρεσιών βάσει του QoS, χρησιμοποιώντας μετρικές ποιότητας που διαφημίζονται από τους παρόχους και υπολογίζονται στα μητρώα.

Οι Vu, Hauswirth, & Aberer (2005) παρουσίασαν μια προσέγγιση για την ποιοτική κατάταξη και επιλογή, η οποία χρησιμοποιεί τεχνικές εμπιστοσύνης και αξιολόγησης φήμης για την πρόβλεψη της μελλοντικής ποιότητας της υπηρεσίας. Οι ιδιότητες της ποιότητας των διαδικτυακών υπηρεσιών περιγράφονται από μια QoS οντολογία που ορίζεται στο WSMO και μια τεχνική πρόβλεψης της μελλοντικής συμμόρφωσης της ποιότητάς τους από παρελθοντικά δεδομένα, ενώ την ίδια ώρα επιτελείται διαχείριση της εμπιστοσύνης των συμβαλλόμενων. Η αξιοπιστία της ανατροφοδότησης από τους χρήστες είναι γενικά ένα σημαντικό πρόβλημα στην ηλεκτρονική επιχειρηματικότητα, στην περίπτωση που χρησιμοποιούνται υποκειμενικές

πληροφορίες, όπως αναφέρεται στα ανωτέρω. Τα Reputation συστήματα πρέπει να είναι σε θέση να φιλτράρουν τις κακόβουλες αναφορές. Οι ίδιοι ερευνητές (Vu, Hauswirth, & Aberer, 2005) πρότειναν ένα μοντέλο που βασίζεται σε αξιόπιστους πράκτορες (agents). Αυτοί παράγουν πάντα αξιόπιστες αναφορές QoS που μπορούν να χρησιμοποιηθούν για την αξιολόγηση της συμπεριφοράς των άλλων χρηστών.

Ο μηχανισμός αξιολόγησης ποιότητας που θα παρουσιαστεί στο παρακάτω κεφάλαιο βασίζεται σε αυτήν ακριβώς τη λογική, με ένα Σύστημα Υποστήριξης Αποφάσεων που εκτελεί συνεργατικές τεχνικές φιλτραρίσματος σε πολλές διαστάσεις πάνω στις QoS αξιολογήσεις των πελατών μιας διαδικτυακής υπηρεσίας, ώστε να παρέχει ένα μηχανισμό επιλογής για Software as a Service. Ο μηχανισμός αυτός, όπως θα αναλυθεί, φιλτράρει τις κακόβουλες αναφορές που πιθανόν υπάρχουν και "προσωποποιεί" τη φήμη του παρόχου για κάθε πελάτη.

4.3.5 Αξιολόγηση ποιότητας κατά τη Σύνθεση Υπηρεσιών σε ροές εργασίας

Μια ειδική περίπτωση της διαδικασίας επιλογής υπηρεσίας είναι αυτή στην οποία έχουμε Σύνθεση Υπηρεσιών (Service Composition), με συνδυασμό μιας σειράς στοιχειωδών υπηρεσιών, συχνά από διαφορετικούς παρόχους, που θα διασυνδεθούν μεταξύ τους και θα παρέχουν νέες εξατομικευμένες υπηρεσίες. Σε αυτή την περίπτωση, η επιλογή των στοιχειωδών υπηρεσιών σχετίζονται συνήθως με από άκρη σε άκρη (end-to-end) QoS απαιτήσεις. Ως εκ τούτου, η σύνθεση πρέπει να γίνεται με βάση την ποιότητα για να βρεθεί ο καλύτερος συνδυασμός υπηρεσιών, τέτοιος ώστε οι συγκεντρωτικές τιμές QoS να ανταποκριθούν σε αυτές τις end-to-end απαιτήσεις.

Οι Alrifai & Risse (2010) πρότειναν μια αποτελεσματική λύση Σύνθεσης Υπηρεσιών με βάση την ποιότητα, με χρήση γραμμικού προγραμματισμού MILP, για να

βρεθεί η βέλτιστη διάσπαση των καθολικών περιορισμών QoS σε τμηματικούς περιορισμούς και να αναζητηθούν οι καλύτερες τμηματικές διαδικτυακές υπηρεσίες που ικανοποιούν αυτούς τους περιορισμούς. Οι Kyriazis, Tserpes, Menychtas, Sarantidis, & Varvarigou (2009) παρουσίασαν μια αντίστοιχη προσέγγιση για την επιλογή υπηρεσιών με QoS κριτήρια, με διάφορες αντιστοιχίσεις των εφαρμογών στις στοιχειώδεις διαδικασίες μιας ροής εργασίας σε υπηρεσίες πλέγματος, ώστε να μεγιστοποιηθεί το όφελος των χρηστών όσον αφορά το προσφερόμενο επίπεδο ποιότητας. Αυτό επιτυγχάνεται μέσω μονάδων εποπτείας, χαρτογράφησης των ροών εργασίας και υποστήριξης αποφάσεων, ελέγχοντας τις διαδικασίες ορισμού της ροής εργασίας, αντιστοιχίζοντάς τις σε στιγμιότυπα υπηρεσιών και αξιολογώντας πληροφορίες από το ιστορικό σχετικά με τη συμπεριφορά των παρόχων.

4.4

Η μεριά του παρόχου

Στις προηγούμενες παραγράφους αναφερθήκαμε το ζήτημα της αξιολόγησης των υπηρεσιών από τη σκοπιά του καταναλωτή, παρουσιάζοντας τεχνικές κατάταξής τους με βάση την παρεχόμενη ποιότητα, και τη χρήση αυτής της αξιολόγησης για την ανάπτυξη Συστημάτων Υποστήριξης Αποφάσεων που βοηθούν τους καταναλωτές στην επιλογή των καλύτερων για αυτούς υπηρεσιών.

Όσον αφορά αυτήν την αξιολόγηση, ο πελάτης δεν χρειάζεται να γνωρίζει τις εσωτερικές διεργασίες που λαμβάνουν χώρα από την πλευρά του παρόχου σε επίπεδο διαχείρισης υπηρεσίας (Service Level Management - SLM) (Sturm, Morris, & Jander, 2000) εφόσον παρέχεται η υπηρεσία στα υποσχόμενα επίπεδα. Η παρούσα ενότητα έχει σκοπό να εξετάσει την αξιολόγηση των υπηρεσιών από την πλευρά του παρόχου, διερευνώντας με ποιους τρόπους οι πάροχοι υπηρεσιών μπορούν να αξιολογούν τις

υπηρεσίες τους και να χρησιμοποιούν τα αποτελέσματα της αξιολόγησης για να αποκομίζουν επιχειρηματικά οφέλη. Για έναν πάροχο, το να μπορεί να υπόσχεται και να προσφέρει ένα υψηλό επίπεδο QoS μεταφράζεται σε ένα σημαντικό πλεονέκτημα σε σχέση με τους άλλους παρόχους που προσφέρουν παρόμοιες ανταγωνιστικές υπηρεσίες.

Στα καταναμημένα υπηρεσιοστρεφή περιβάλλοντα Software as a Service (πχ. υπολογιστικά πλέγματα και υπολογιστικά νέφη), οι Συμφωνίες σε Επίπεδο Υπηρεσίας, όπως έχει ήδη συζητηθεί στα προηγούμενα, έχουν καθιερωθεί ως το πρότυπο εργαλείο για τον καθορισμό των εγγυήσεων ποιότητας που υπόσχεται ο πάροχος στον καταναλωτή, με τη διαχείριση των υπηρεσιών στα εν λόγω συστήματα να γίνεται όλο και περισσότερο καθοδηγούμενη από τα SLAs. Ένα σημαντικό βήμα προς αυτή την κατεύθυνση ήταν η δημοσίευση του μοντέλου Web Service Level Agreement (WSLA) (Keller & Ludwig, 2002), με τη θέσπιση προδιαγραφών για τον καθορισμό και την παρακολούθηση των SLAs των διαδικτυακών υπηρεσιών, που έθεσε τη βάση για το σχεδιασμό πολλών υπηρεσιοστρεφών αρχιτεκτονικών, ενώ επίσης ενσωματώθηκε και σε υφιστάμενα μοντέλα διαχείρισης. Οι Debusmann & Keller (2003) ερεύνησαν πώς το WSLA μπορεί να προσαρμοστεί για έναν πάροχο υπηρεσιών που χρησιμοποιεί ένα παραδοσιακό σύστημα διαχείρισης, όπως το Common Information Model (Force, 1999).

Προκειμένου να προσελκύσουν περισσότερους πελάτες και να βελτιώσουν τη φήμη τους, οι πάροχοι υπηρεσιών πρέπει όχι μόνο να προσφέρουν SLAs με καλύτερους όρους, αλλά κυρίως να είναι σε θέση να παρέχουν τη συμφωνημένη ποιότητα. Για συστήματα που βασίζονται σε SLA, αυτό μεταφράζεται σε μια σειρά προκλήσεων:

- Δυναμικές αποφάσεις για το αν νέα SLAs μπορούν να συμφωνηθούν με βάση την τρέχουσα χωρητικότητα του συστήματος.

- Δυναμική διαπραγμάτευση των παραμέτρων του SLA, κατά τη δημιουργία νέων συμφωνιών.
- Συνεχή παρακολούθηση των πόρων και των εφαρμογών που τρέχουν στο σύστημα.
- Αξιολόγηση του αποτελέσματος της παρακολούθησης των παραμέτρων του SLA, προκειμένου να αποφασιστεί κατά πόσον οι όροι αυτοί έχουν εκπληρωθεί.
- Σε περίπτωση που δεν έχουν εκπληρωθεί οι όροι ενός SLA, δυναμικές ενέργειες και μέτρα που λαμβάνονται για την αντιμετώπιση της παραβίασης.

4.4.1 Αντιστοίχιση παραμέτρων υψηλού και χαμηλού επιπέδου

Μια βασική απαίτηση για την αντιμετώπιση των προαναφερθέντων προκλήσεων είναι η κατάλληλη αντιστοίχιση των όρων υψηλού επιπέδου με τις χαμηλού επιπέδου παραμέτρους. Οι παράμετροι χαμηλού επιπέδου, ή μετρικές πόρων όπως ορίζονται από τους Tian, Gramm, Naumowicz, Ritter, & Schiller (2003), ανακτώνται απευθείας από το σύστημα διαχείρισης των πόρων με διάφορους μετρητές χρήσης. Ο συνδυασμός διαφόρων μετρικών πόρων σύμφωνα με ένα συγκεκριμένο αλγόριθμο ορίζει μια σύνθετη μετρική ή μια παράμετρο υψηλού επιπέδου. Παραδείγματα σύνθετων μετρικών είναι ο μέγιστος χρόνος απόκρισης μιας υπηρεσίας, η μέση διαθεσιμότητα της υπηρεσίας, ή το ελάχιστο διαθέσιμο εύρος ζώνης μιας υπηρεσίας. Παραδείγματα μετρικών πόρων είναι ο χρόνος που το σύστημα λειτουργεί, η υπολογιστική ισχύς που διατίθεται σε μια κλήση, ή ο αριθμός κλήσεων μιας υπηρεσίας. Οι παράμετροι ενός πρότυπου SLA παρέχουν για τις σύνθετες μετρικές (ή και για τις μετρικές πόρων) ένα εύρος τιμών που κάθε μια μπορεί να γίνει αποδεκτή από κάθε πελάτη. Έτσι, ουσιαστικά, κάθε παράμετρος του SLA

καθορίζει τη συμφωνηθείσα απόδοση της υπηρεσίας σε σχέση με μια συγκεκριμένη μετρική.

Μια λανθασμένη αντιστοίχιση μεταξύ των παραμέτρων χαμηλού και υψηλού επιπέδου σημαίνει ότι ο πάροχος είτε θα διαθέσει περισσότερους πόρους από ό,τι απαιτείται για μια συγκεκριμένη δουλειά, προκειμένου να διασφαλιστεί ότι το SLA θα εκπληρωθεί, είτε θα δέχεται περισσότερα SLAs από ότι μπορεί να αντέξει, κάτι που οδηγεί σε παραβιάσεις. Οι Rosenberg, Platzer, & Dustdar (2006) προσδιόρισαν κάποια σημαντικά QoS χαρακτηριστικά και παρουσίασαν τεχνικές αντιστοίχισης για το συνδυασμό μετρικών πόρων με σύνθετες μετρικές, προκειμένου να σχηματίσουν τις παραμέτρους SLA για ένα συγκεκριμένο τομέα, χωρίς ωστόσο να λαμβάνουν υπόψη τα αποτελέσματα παρακολούθησης της χρήσης των πόρων.

Η παρακολούθηση και αξιολόγηση των υπηρεσιών βοηθούν τον πάροχο στη συλλογή στατιστικών στοιχείων σχετικά με την απόδοση των υπηρεσιών, που μπορούν μακροπρόθεσμα να αναλυθούν. Η γνώση που αποκτάται από την ανάλυση αυτή βοηθάει τον πάροχο να κάνει μια καλύτερη αντιστοίχιση μεταξύ παραμέτρων υψηλού και χαμηλού επιπέδου κι έτσι να βελτιώσει την αξιοποίηση των πόρων του και να ελαχιστοποιήσει τις παραβιάσεις SLA. Οι Emeakaroha, C., Brandic, Maurer, & Dustdar (2010) προτείνουν ένα πλαίσιο για τη διαχείριση των αντιστοιχίσεων των χαμηλού επιπέδου μετρικών πόρων σε SLAs υψηλού επιπέδου, αναλύοντας την παρακολούθηση της χρήσης των μετρικών επιπέδου πόρων. Επίσης, οι Tserpes, Kyriazis, Menychtas, & Varvarigou (2008) ανέπτυξαν ένα μηχανισμό που παρακολουθεί τις επιδόσεις του φορέα παροχής υπηρεσιών καθ' όλη τη διάρκεια της ζωής του και χρησιμοποιεί τα ιστορικά

στοιχεία, προκειμένου να καθοριστούν καλύτερα οι υψηλού επιπέδου παράμετροι σε σχέση με τις μετρικές των πόρων.

4.4.2 Βελτιστοποίηση λειτουργίας παρόχου σε πραγματικό χρόνο

Εκτός από τη μακροπρόθεσμη ανάλυση, η αποτελεσματική παρακολούθηση και ο εντοπισμός παραβιάσεων μπορεί να βοηθήσουν τον πάροχο να βελτιστοποιήσει τη λειτουργία του κατά τη διάρκεια του χρόνου εκτέλεσης, χρησιμοποιώντας ένα δυναμικό μηχανισμό διαχείρισης των πόρων που επιτρέπει την ανακατανομή και τη μεταφορά των εφαρμογών που εκτελούνται.

Οι Ferretti, Ghini, Panzieri, Pellegrini, & Turrini (2010) εισήγαγαν μια αρχιτεκτονική ενδιάμεσου λογισμικού που δίνει τη δυνατότητα δυναμικής διαμόρφωσης, διαχείρισης και βελτιστοποίησης των πόρων και υπηρεσιών του υπολογιστικού νέφους με βάση το SLA. Η προτεινόμενη αρχιτεκτονική περιλαμβάνει έναν εξισορροπιστή φορτίου που κατανέμει τον υπολογιστικό φόρτο ανάμεσα στους πόρους της πλατφόρμας, παρακολουθώντας το παρεχόμενο QoS, προκειμένου να διασφαλιστεί η βέλτιστη χρήση των πόρων. Αν ανιχνευθεί απόκλιση από την υποσχεθείσα ποιότητα που θα μπορούσε να οδηγήσει σε παραβίαση του SLA, η πλατφόρμα αναδιαρθρώνεται δυναμικά, προκειμένου να χρησιμοποιήσει πρόσθετους πόρους. Ομοίως, εάν ανιχνευθεί κατά τη διάρκεια του χρόνου εκτέλεσης ότι το SLA μπορεί να εκπληρωθεί με λιγότερους πόρους από εκείνους που είχαν χορηγηθεί αρχικά, πάλι εκτελείται αναδιάρθρωση της πλατφόρμας, προκειμένου να απελευθερωθούν οι μη χρησιμοποιηθέντες πόροι.

Οι Lodi, Panzieri, Ross, & Turrini (2007) καθόρισαν μια αρχιτεκτονική ενδιάμεσου λογισμικού που επιτελεί μια ποιοτική ομαδοποίηση των διακομιστών εφαρμογών με βάση το SLA. Η τεχνολογία του διακομιστή εφαρμογών, χωρίς την

ποιοτική υποστήριξη, προσφέρει μηχανισμούς ομαδοποίησης και εξισορρόπησης φορτίου για την ικανοποίηση των απαιτήσεων επεκτασιμότητας και διαθεσιμότητας. Προκειμένου να ενισχυθεί ο εν λόγω μηχανισμός με τη διαχείριση του SLA, η προαναφερθείσα αρχιτεκτονική υποστηρίζει δυναμική διαχείριση των πόρων, με τους διακομιστές εφαρμογών να μπορούν να αλλάζουν δυναμικά την ποσότητα των κατανεμηθέντων πόρων που διατίθενται για τις εφαρμογές που φιλοξενούνται σε αυτούς κατά απαίτηση, έτσι ώστε να ανταποκρίνονται στις QoS απαιτήσεις που τίθενται σε επίπεδο εφαρμογής.

4.4.3 Προσαρμογή των SLAs βάσει της συμπεριφοράς του παρόχου

Εκτός από τη δυναμική ανακατανομή των πόρων, η συλλογή δεδομένων παρακολούθησης κατά το χρόνο εκτέλεσης μπορεί να χρησιμοποιηθεί από τους παρόχους και για τη δυναμική προσαρμογή των προτύπων SLAs που προτείνουν στους πελάτες με βάση την απόδοση των πόρων. Οι Spillner & Schill (2009) προτείνουν έναν καινοτόμο μηχανισμό που συγκεντρώνει τα αποτελέσματα της παρακολούθησης και τα ανατροφοδοτεί πίσω στο μητρώο υπηρεσιών για την αναπροσαρμογή των τιμών των μη λειτουργικών ιδιοτήτων στις περιγραφές των υπηρεσιών, ανάλογα με τη συμπεριφορά κατά το χρόνο εκτέλεσης της υπηρεσίας. Χρησιμοποιώντας αυτές τις ενημερωμένες τιμές, οι αντίστοιχοι περιορισμοί στα πρότυπα SLAs ευθυγραμμίζονται με ημι-αυτόματο τρόπο με τους επιχειρηματικούς στόχους.

Για έναν φορέα παροχής υπηρεσιών, είναι σημαντικό όχι μόνο να ικανοποιήσει τη ζήτηση των καταναλωτών του για Ποιότητα Υπηρεσίας υψηλού επιπέδου, αλλά και να εκπληρώσει τους δικούς του επιχειρηματικούς στόχους, ελαχιστοποιώντας με οικονομικά αποδοτικό τρόπο τις επιπτώσεις των παραβιάσεων του SLA. Οι Bucu,

Chang, Luan, Ward, Wolf, & Yu (2004) παρουσίασαν τις προδιαγραφές σχεδιασμού ενός συστήματος διαχείρισης SLA σε ένα Utility Computing SaaS περιβάλλον, με βάση τους επιχειρηματικούς στόχους, ονόματι SAM. Το SAM είναι ένα καταναμημένο σύστημα διαχείρισης SLA που βοηθάει έναν Utility Computing πάροχο στη μείωση του οικονομικού κινδύνου των παραβιάσεων που γίνονται στο επίπεδο της Ποιότητας Υπηρεσίας, εκπληρώνοντας τις απαιτήσεις του SLA, σε συμφωνία πάντα με τις επιχειρηματικές ανάγκες του παρόχου.

4.4.4 Ενοικίαση πόρων

Μια ειδική περίπτωση στο πρόβλημα της μέτρησης και κατανομής των πόρων είναι αυτή που ο ρόλος του παρόχου είναι διπλός: μπορεί να ενεργεί είτε ως ιδιοκτήτης των πόρων ή ως ενοικιαστής. Προκειμένου να ικανοποιήσουν τις ανάγκες των πελατών τους, όλο και περισσότερες επιχειρήσεις χρησιμοποιούν εξωτερικούς πόρους σε συνδυασμό με δικούς τους, όντας ταυτόχρονα πάροχοι και πελάτες. Αυτό δημιουργεί μια πρόσθετη απαίτηση στο μηχανισμό παρακολούθησης και αξιολόγησης: ο πάροχος θα πρέπει τώρα να παρακολουθεί το QoS που παρέχεται και από εξωτερικούς πόρους εκτός των δικών του, καθώς και να λαμβάνει υπόψη του το πρόσθετο κόστος για την ενοικίαση αυτών.

Ένα δυναμικό σχήμα κατανομής των πόρων με βάση την παρακολούθηση θα μπορούσε στην παραπάνω περίπτωση να βοηθήσει τον πάροχο στην επίτευξη της βελτιστοποίησης της χρήσης των πόρων και τη μεγιστοποίηση των κερδών του. Για παράδειγμα, αν κατά τη διάρκεια του χρόνου εκτέλεσης, η χρησιμοποίηση των πόρων του φορέα παροχής υπηρεσιών είναι χαμηλή, τότε θα μπορούσε να εξετάσει τη δυνατότητα να ελευθερώσει κάποιους πόρους για τη μείωση του κόστους, χωρίς να παραβιάσει εν τω μεταξύ κάποιο από τα ενεργά SLAs. Ένα παρόμοιο μοντέλο δυναμικής

ανακατανομής με τη χρήση μιας ευριστικής συνάρτησης και μιας τεχνικής διαχείρισης ρίσκου, θα παρουσιαστεί στο Κεφάλαιο 6 της παρούσας διατριβής, ως μια καινοτόμα μέθοδος αξιολόγησης της Ποιότητας Υπηρεσίας από την πλευρά του παρόχου υπηρεσιών.

5

Πολυδιάστατη

Συνεργατική Αξιολόγηση

Ποιότητας Υπηρεσίας από

το μέρος του Πελάτη

Η μέθοδος αξιολόγησης που θα παρουσιαστεί στο παρόν κεφάλαιο εκμεταλλεύεται την έννοια της Ποιότητας της Εμπειρίας που παρουσιάστηκε στα προηγούμενα, προκειμένου να καταστεί δυνατό για τους πελάτες να αξιολογούν τις διάφορες υπηρεσίες και να καθιστούν αυτές τις αξιολογήσεις προς χρήση για την παροχή συστάσεων προς άλλους πελάτες. Έτσι, χρησιμοποιώντας τις αναφορές αυτές, οι πελάτες ομαδοποιούνται ανάλογα με τις ομοιότητές τους, προκειμένου να προβλέψουμε την αξιολόγησή τους για υπηρεσίες που δεν έχουν χρησιμοποιήσει μέχρι τώρα. Συσχετίζοντας τις υπάρχουσες αναφορές από διάφορους πελάτες, είναι δυνατόν να παρέχουμε συστάσεις σε πελάτες με παρόμοιες προτιμήσεις με άλλους, έτσι ώστε να υποστηριχθούν στη διαδικασία επιλογής των υπηρεσιών με διασφάλιση των κριτηρίων που θέτουν για την ποιότητα.

Η μέθοδος που χρησιμοποιεί το προτεινόμενο αυτό μοντέλο επιλογής υπηρεσίας (Tserpes, Aisopos, Kyriazis, & Varvarigou, 2010) αποτελείται από δύο βήματα: Το πρώτο βήμα περιλαμβάνει τη δημιουργία ενός μηχανισμού που συλλέγει τις αξιολογήσεις των καταναλωτών (ratings) όσον αφορά τις QoS παραμέτρους τις οποίες τα δύο μέρη είχαν συμφωνήσει πριν από την κλήση μιας υπηρεσίας. Το δεύτερο χρησιμοποιεί συνεργατικές τεχνικές φιλτραρίσματος (collaborative filtering techniques) και ειδικότερα υπολογίζει τη συσχέτιση ανάμεσα στις αξιολογήσεις των πελατών με χρήση του συντελεστή συσχέτισης του Pearson. Τα δύο αυτά βήματα προϋποθέτουν μια μικρή τροποποίηση στην διαχείριση του κύκλου ζωής των υπηρεσιών, που αναλύεται διεξοδικά παρακάτω.

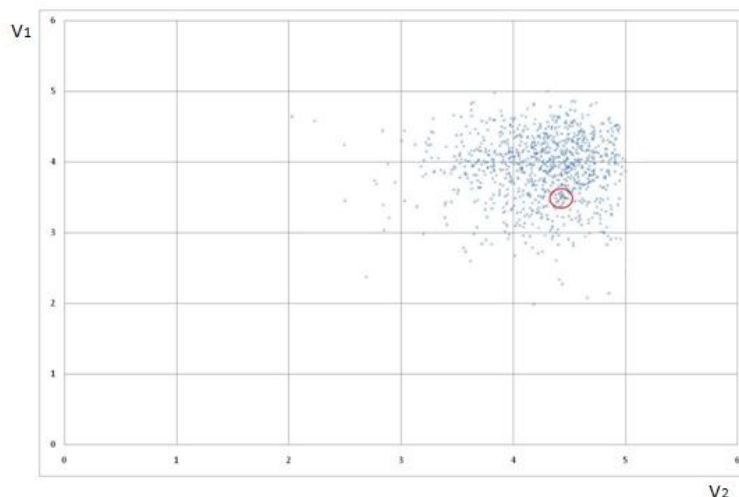
Η πειραματική υλοποίηση του μοντέλου βασίζεται σε μια SaaS υποδομή που ακολουθεί τις αρχές ενός υπηρεσιοστρεφούς περιβάλλοντος (Tserpes, Aisopos, Kyriazis, & Varvarigou, 2011). Στα περιβάλλοντα SOA, συνήθως όλοι οι πάροχοι υπηρεσιών καταγράφουν τις υπηρεσίες τους σε ένα service broker και οι διάφοροι καταναλωτές αναζητούν και ανακαλύπτουν τις επιθυμητές υπηρεσίες μέσω αυτού. Το ρόλο αυτόν του broker αναλαμβάνει ένα μητρώο διαδικτυακών υπηρεσιών, συνοδευόμενο συχνά από μηχανισμούς υποστήριξης για τη διαχείρισή τους. Για κάθε μία από τις διαθέσιμες περιγραφές διαδικτυακών υπηρεσιών, οι πάροχοι επισυνάπτουν μια πρότυπη συμφωνία SLA, ως ένα είδος προσφοράς που καθορίζει τους όρους της κλήσης της υπηρεσίας και ποσοτικοποιεί το επίπεδο ποιότητας υπηρεσίας που προσφέρουν. Μια υπηρεσία καλείται εκ μέρους των καταναλωτών μόνον αφού έχουν συμφωνηθεί οι όροι του SLA μεταξύ των δύο πλευρών. Οι όροι του SLA περιέχουν το σύνολο των παραμέτρων που καθορίζουν την παρεχόμενη ποιότητα. Μόλις οι όροι αυτοί συμφωνηθούν, οι πάροχοι

παρέχουν πρόσβαση στους πόρους τους μέσω της διεπαφής της υπηρεσίας και παρακολουθούν τις QoS παραμέτρους. Η όλη αυτή διαδικασία αναφέρεται ως "διαπραγμάτευση του SLA" και μπορεί να πραγματοποιηθεί με τη βοήθεια του service broker. Σε περίπτωση που συμβεί κάποια παραβίαση των όρων, οι ρήτρες SLA τίθενται σε ισχύ για το αντίστοιχο μέρος που την προκάλεσε.

Στην τρέχουσα περίπτωση, η προτεινόμενη υποδομή επιτρέπει στους παρόχους υπηρεσιών να εγγράφουν τις διαδικτυακές υπηρεσίες τους σε ένα μητρώο UDDI. Οι καταναλωτές υπηρεσιών μπορούν να ανακαλύψουν στη συνέχεια τις υπηρεσίες που θέλουν και να τις καλέσουν με βάση τη λειτουργική περιγραφή τους, η οποία με τη σειρά καταγράφεται χρησιμοποιώντας τη γλώσσα περιγραφής διαδικτυακών υπηρεσιών (WSDL) που παρουσιάστηκε στην εισαγωγή. Με τον τερματισμό της χρήσης μιας υπηρεσίας, οι καταναλωτές παρέχουν μια αξιολόγηση που υποδηλώνει το επίπεδο ικανοποίησής τους από κάθε παράμετρο QoS που περιέχεται στο SLA. Οι βαθμολογίες αυτές αποθηκεύονται στο service broker ο οποίος διατηρεί το μητρώο των υπηρεσιών και των SLAs, δημιουργώντας ένα προφίλ τόσο για τον παρόχο όσο και για τους καταναλωτές. Με τη χρήση των συνεργατικών τεχνικών φιλτραρίσματος, το σύστημα βρίσκει καταναλωτές των οποίων τα διανύσματα αξιολόγησης εμφανίζουν μεγάλη συσχέτιση και εφαρμόζει ένα μηχανισμό συστάσεων (Service Recommendation - SR), προτείνοντας υπηρεσίες που ικανοποιούν τις μη λειτουργικές απαιτήσεις τους.

Η χρήση μεθόδων συσχέτισης σε Recommendation συστήματα δεν είναι καινούργια ιδέα. Σε γενικές γραμμές, με την ευρεία εξάπλωση του διαδικτύου και την άνθιση του ηλεκτρονικού εμπορίου, αλγόριθμοι συνεργατικού φιλτραρίσματος χρησιμοποιήθηκαν εκτενώς σε συστήματα συστάσεων. Επιπλέον, οι τεχνικές συσχέτισης

είναι από τις πρώτες επιλογές των μηχανικών λογισμικού για την ανάπτυξη συστημάτων υποστήριξης αποφάσεων, δεδομένου ότι είναι αξιόπιστες, γρήγορες και εύκολες να υλοποιηθούν. Μια γενική εικόνα της έννοιας της συσχέτισης μεταβλητών πολλών διαστάσεων παρουσιάζεται στο Σχήμα 19, με ένα απλό παράδειγμα διανυσμάτων του Ευκλείδειου χώρου 2-διαστάσεων $V_i = [v_1, v_2]$, όπου $1 \leq v_1, v_2 \leq 5$, αντίστοιχο με την περίπτωση αξιολόγησης SLA δύο παραμέτρων. Στο σχήμα αυτό, μόνο τα τελικά σημεία των διανυσμάτων επισημαίνονται στο χώρο, τα οποία και προέρχονται από τα πειράματα που θα πραγματοποιηθούν παρακάτω, ενώ όλα τα διανύσματα θεωρείται ότι έχουν την αρχή τους στο σημείο $O(0,0)$:



Σχήμα 19: Γράφημα συσχέτισης διανυσμάτων 2-διαστάσεων. Ο κύκλος δείχνει την "γειτνίαση" του διανύσματος v_i , με διανύσματα με έντονη συσχέτιση

Όπως μπορεί να παρατηρηθεί στο Σχήμα 19, τα σημεία στη "γειτονιά" ενός τυχαίου διανύσματος v_i , υποδεικνύουν διανύσματα με παρόμοιες τιμές για τις δύο διαστάσεις. Αυτά τα γειτονικά διανύσματα, καθώς και όσα είναι παράλληλα του v_i , συνδέονται λογικά μέσω γραμμικής σχέσης και μπορεί να χρησιμοποιηθούν για την εξαγωγή πολύτιμων πληροφοριών, όπως θα δείξουμε παρακάτω.

Εδώ, προκειμένου να αξιολογηθεί η υλοποιηθείσα τεχνική, η παρούσα αναφορά παρουσιάζει αρχικά μια προσομοίωση του μοντέλου σε ένα υπηρεσιοστρεφές περιβάλλον όπου τεχνητά σύνολα δεδομένων χρησιμοποιούνται ως σύνολα αξιολόγησης για μια υπηρεσία 3D Rendering. Στη συνέχεια παρουσιάζεται μια πρακτική επαλήθευση, με εισαγωγή πραγματικών αξιολογήσεων από animators που χρησιμοποιούν και βαθμολογούν μια σειρά από υπηρεσίες με τρεις κατηγορίες συμβολαίων (Gold, Silver και Bronze), δεχόμενοι συστάσεις από τον υπό εξέταση μηχανισμό.

5.1 Κύκλος ζωής υπηρεσίας

Η μέθοδος αξιολόγησης που θα παρουσιασθεί αναλυτικά παρακάτω προϋποθέτει την εισαγωγή δύο αλληλένδετων διαδικασιών ενδιάμεσα στα στάδια του κύκλου ζωής μιας υπηρεσίας σε περιβάλλοντα SaaS: Η πρώτη είναι η διαδικασία βαθμολόγησης της ποιότητας μιας υπηρεσίας μετά από κάθε κλήση της από τον τελικό χρήστη και η δεύτερη είναι η παροχή συστάσεων (SR μηχανισμός) σε αυτόν κατά το στάδιο επιλογής υπηρεσίας, ως αποτέλεσμα της ανάλυσης συσχετίσεων που παρέχει το μοντέλο αξιολόγησης. Στην παράγραφο αυτή παρουσιάζονται αναλυτικά τα στάδια του κύκλου ζωής μιας τέτοιας διαδικτυακής υπηρεσίας, με έμφαση στις αλλαγές που το SR μοντέλο επιφέρει. Όπως είναι αναμενόμενο, ο SR μηχανισμός επηρεάζει το στάδιο επιλογής υπηρεσίας, όπου παρεμβαίνει συστήνοντας υπηρεσίες και επίσης εισάγει ένα νέο στάδιο, αυτό της αξιολόγησης/βαθμολόγησης των υπηρεσιών.

5.1.1 Δημοσίευση Υπηρεσίας

Η δημοσίευση μιας υπηρεσίας είναι η εναρκτήρια φάση του κύκλου ζωής. Οι πάροχοι μπορούν να καταχωρούν τις υπηρεσίες τους σε μητρώα που ανήκουν σε service brokers,

υποβάλλοντας μια περιγραφή των ιδιοτήτων τους. Έτσι, οι υπηρεσίες γίνονται διαθέσιμες στους καταναλωτές όταν αυτοί τις ζητούν ακολουθώντας μια πατέντα δημοσίευσης / συνδρομής. Παρόλο που αρκετές προσεγγίσεις έχουν προταθεί για τη μοντελοποίηση των ιδιοτήτων των υπηρεσιών, ώστε να μπορούν εύκολα να καταχωρηθούν (με την πιο σημαντική ίσως να είναι η Οντολογία Μοντελοποίησης Υπηρεσιών (Web Service Modelling Ontology), κανένα μοντέλο δεν έχει υιοθετηθεί ευρέως από τις αγορές διαδικτυακών υπηρεσιών που να περιλαμβάνει τις μη λειτουργικές ιδιότητες. Αυτό οφείλεται κυρίως στο γεγονός ότι οι μη λειτουργικές ιδιότητες είναι προσανατολισμένες στην εφαρμογή, γεγονός που καθιστά πρακτικά αδύνατη την απαρίθμηση ή ακόμη και την ποσοτικοποίησή τους. Αντίθετα, το μοντέλο για την καταγραφή των λειτουργικών χαρακτηριστικών που έχει καθιερωθεί τα τελευταία χρόνια όπως προαναφέρθηκε είναι η γλώσσα WSDL.

5.1.2 Ανακάλυψη Υπηρεσίας

Μόλις οι υπηρεσίες έχουν δημοσιευθεί σε ένα μητρώο, οι καταναλωτές μπορούν να εγγραφούν και να ανακτήσουν εκείνες που ανταποκρίνονται στις ανάγκες τους. Για να επιτευχθεί αυτό, υποβάλλουν τις απαιτήσεις τους με τη μορφή ενός συνόλου ιδιοτήτων που θα επιθυμούσαν να έχουν οι υπηρεσίες. Το μητρώο UDDI αποτελεί την πιο κοινή υλοποίηση μητρώου υπηρεσιών. Οι διαδικτυακές υπηρεσίες καταγράφονται στα μητρώα UDDI ως περιγραφές WSDL για τη μοντελοποίηση των διεπαφών και της υλοποίησής τους. Οι λειτουργικές ιδιότητες κάθε υπηρεσίας μπορούν στη συνέχεια να αναζητηθούν χρησιμοποιώντας την αντίστοιχη διεπαφή του UDDI και ένα tModel, μια XML περιγραφή των διεπαφών που χρησιμοποιούνται στο UDDI, που μπορεί επίσης να συμπεριλαμβάνει τις WSDL έννοιες.

Το αποτέλεσμα αυτού του ερωτήματος από τον καταναλωτή επιστρέφει μια λίστα συνδέσμων WSDL, που αντιστοιχούν στις υπηρεσίες που ταιριάζουν με τα λειτουργικά κριτήρια. Ωστόσο, ο καταναλωτής πρέπει να κάνει μια επιλογή από τη λίστα αυτή με βάση διαφορετικά κριτήρια. Τα κριτήρια αυτά είναι οι μη λειτουργικές απαιτήσεις ή απαιτήσεις QoS, που σε συνδυασμό με το κόστος των υπηρεσιών, θα οδηγήσει στην τελική απόφαση για το ποια υπηρεσία θα κληθεί. Η διαδικασία αυτή περιγράφεται στην συνέχεια.

5.1.3 Επιλογή Υπηρεσίας

Για να επιλέξουν οι καταναλωτές μία υπηρεσία από ένα σύνολο υπηρεσιών που πληρούν τις λειτουργικές απαιτήσεις τους, πρέπει να φιλτράρουν τη λίστα αυτή ανάλογα με τις απαιτήσεις ποιότητας υπηρεσίας και πιθανόν να πληρώσουν για αυτό. Σε αυτό το στάδιο, ο πάροχοι και ο καταναλωτής διαπραγματεύονται την ποιότητα που θα παρασχεθεί, έτσι ώστε ο δεύτερος να καταλήξει στην περισσότερο κατάλληλη υπηρεσία. Στα περιβάλλοντα αναφοράς, το QoS ορίζεται και καταμετράται μέσω SLAs, συνεπώς αυτό το στάδιο του κύκλου ζωής μπορεί να ενσωματώσει το υπο-στάδιο της διαπραγμάτευσης SLA. Η "διαπραγμάτευση του SLA" είναι μια διαδικασία που επιτρέπει σε ένα φορέα παροχής υπηρεσιών και ένα καταναλωτή υπηρεσιών να συμφωνήσουν σχετικά με τους όρους υπό τους οποίους οι πόροι του παρόχου θα χρησιμοποιηθούν από τον καταναλωτή κατά τη διάρκεια της κλήσης μιας υπηρεσίας. Το αποτέλεσμα αυτής της διαδικασίας είναι μια συμφωνία SLA που χρησιμοποιείται για την παρακολούθηση και εφαρμογή των συμφωνηθέντων όρων.

Το επικρατούν μοντέλο διαπραγμάτευσης είναι το μοντέλο "διακριτής προσφοράς" (discrete offer model). Η ιδέα πίσω από αυτό το μοντέλο διαπραγμάτευσης

είναι ότι ο πάροχος δημιουργεί σταθερές προσφορές που παρουσιάζονται σε ένα είδος προ-συμφώνου, το SLA Template, το οποίο περιέχει όλες τις λεπτομέρειες που καθορίζουν την προσφορά, εκτός από τα στοιχεία του καταναλωτή. Ως εκ τούτου, ένας πάροχος μπορεί να δημιουργήσει για παράδειγμα τρία πρότυπα SLAs για μια υπηρεσία, το καθένα με διαφορετικούς όρους ποιότητας και ρήτρες χρέωσης. Με τον τρόπο αυτό, ο πάροχος μπορεί να κλιμακώσει την παραδοθείσα ποιότητα σε τρία διακριτά επίπεδα: "Gold" / "Silver" / "Bronze". Εάν οι καταναλωτές επιθυμούν να καλέσουν τη συγκεκριμένη υπηρεσία, θα πρέπει να επιλέξουν ένα από αυτά τα SLA Templates και να ενημερώσουν τον πάροχο. Ο τελευταίος παράγει στη συνέχεια το πλήρες έγγραφο του SLA οι λεπτομέρειες του οποίου (όπως οι παράμετροι QoS) αποστέλλονται στον καταναλωτή. Οι όροι του SLA ανακτώνται στη συνέχεια από το λογισμικό που ασχολείται με την παρακολούθηση της χρήσης της υπηρεσίας και της εφαρμογής των όρων της συμφωνίας.

Συνοψίζοντας τα παραπάνω, στο στάδιο της επιλογής υπηρεσίας, η λίστα των διαθέσιμων υπηρεσιών που παράγονται στο προηγούμενο στάδιο παρουσιάζεται στους καταναλωτές με τη μορφή ενός καταλόγου προτύπων SLA που συνδέονται με αυτές τις υπηρεσίες. Οι καταναλωτές αξιολογούν τις προσφορές και επιλέγουν ένα από τα πρότυπα SLA. Με αυτόν τον τρόπο, καταφέρνουν όχι μόνο να επιλέξουν μια συγκεκριμένη υπηρεσία που ικανοποιεί τις λειτουργικές απαιτήσεις τους, αλλά και εκείνη που είναι πιο κοντά στις μη λειτουργικές απαιτήσεις. Ο μηχανισμός που προτείνεται προσθέτει επιπλέον εργασίες σε αυτό το σημείο της διαχείρισης του κύκλου ζωής των υπηρεσιών. Αυτές περιλαμβάνουν τη χρήση ενός SR μηχανισμού που επιτελεί μια πρόβλεψη αξιολόγησης για λογαριασμό του καταναλωτή, ώστε να τον βοηθήσει

στην επιλογή του κατάλληλου προτύπου SLA ενός συγκεκριμένου παρόχου από ένα σύνολο εναλλακτικών λύσεων, με βάση τη συλλογική εμπειρία των καταναλωτών, όπως θα συζητηθεί σε μεγαλύτερη λεπτομέρεια παρακάτω.

5.1.4 Κλήση Υπηρεσίας (δέσμευση) και Παρακολούθηση

Όταν οι καταναλωτές καταλήξουν σε ένα πρότυπο SLA, ανακοινώνουν την απόφασή τους στον πάροχο. Στη συνέχεια, ο τελευταίος δημιουργεί ένα έγγραφο SLA το οποίο περιέχει όλες τις απαραίτητες πληροφορίες για την εκτέλεση και παρακολούθηση της υπηρεσίας, όπως εξηγήθηκε στην προηγούμενη παράγραφο. Παράλληλα, ο πάροχος κάνει τους απαραίτητους πόρους διαθέσιμους για την συμφωνημένη χρονική περίοδο κατά την οποία το SLA θα παραμείνει ενεργό. Οι καταναλωτές μπορούν τώρα να καλέσουν την υπηρεσία εντός της ανωτέρω προθεσμίας, με χρήση των δεδομένων κλήσης που δίδονται από τον πάροχο. Η διαδικασία αυτή είναι επίσης γνωστή ως "δέσμευση υπηρεσίας" (service binding).

Οι καταναλωτές πρέπει να διασφαλίσουν ότι δεν θα παραβιάσουν το μέρος του SLA που αντιστοιχεί σε αυτούς. Αν το κάνουν, τότε ο πάροχος θα ελέγξει αμέσως τους πόρους του και θα θέσει τις ρήτρες SLA σε ισχύ. Από την άλλη πλευρά, οι πάροχοι πρέπει να επίσης να συμμορφώνονται με τους όρους του SLA που αφορούν το μέρος τους. Ωστόσο, δεν υπάρχει αντικειμενικός τρόπος για τους καταναλωτές να γνωρίζουν ότι το έκαναν, δεδομένου ότι οι ίδιοι οι πάροχοι κατέχουν και ελέγχουν τους πόρους οι οποίοι παρέχονται, προσθέτοντας ένα είδος μεροληψίας στις μετρήσεις. Οι καταναλωτές μπορούν να βασίζονται μόνο στην εμπειρία τους για να κριθεί η ποιότητα των παρεχόμενων υπηρεσιών.

5.1.5 Αξιολόγηση Υπηρεσίας

Το στάδιο αξιολόγησης των υπηρεσιών είναι μία προσθήκη στην τυπική διαχείριση του κύκλου ζωής που επιβάλλεται από τη χρήση του SR μηχανισμού. Ως εκ τούτου, κατά τη λήξη της ισχύος ενός SLA, ζητείται στους καταναλωτές από το service broker να βαθμολογήσουν τους όρους του SLA υποδεικνύοντας την ικανοποίησή τους για τα επίπεδα QoS που παραδόθηκαν από τον πάροχο. Αυτό επιτρέπει τη δημιουργία ενός ιστορικού βαθμολογιών από πολλούς καταναλωτές που μπορεί να χρησιμοποιηθεί για να αξιολογήσει τη φήμη των παρόχων καθώς και να καταστεί δυνατή η σύσταση υπηρεσιών (ουσιαστικά προτύπων SLA) κατά το στάδιο επιλογής υπηρεσίας.

5.2

Μοντέλο πρόβλεψης

αξιολογήσεων

Η ιδέα της μεθόδου πρόβλεψης αξιολόγησης κατά το στάδιο επιλογής υπηρεσίας βασίζεται στην υπόθεση ότι σε ένα σύνολο από πελάτες με δυνητικά πολύ μεγάλο αριθμό, είναι δυνατό να αναγνωριστούν ομοιότητες στη "συμπεριφορά" διακριτών ομάδων. Αυτό επιτρέπει το σχηματισμό υπο-ομάδων που μπορούν να εξεταστούν σε τοπικό επίπεδο και να αναγνωριστεί ένα γενικό πλαίσιο που χαρακτηρίζει τη συμπεριφορά της κάθε μιας από αυτές. Με τον τρόπο αυτό καθίσταται δυνατό να προβλεφθεί σε μεγάλο βαθμό η συμπεριφορά ενός πελάτη με τον προσδιορισμό της γενικής συμπεριφοράς της υπο-ομάδας.

Σε μια ανοικτή Software as a Service αγορά υπηρεσιών, ένας μηχανισμός υποστήριξης λήψης αποφάσεων για την επιλογή των υπηρεσιών μπορεί να λειτουργεί με παρόμοιο τρόπο όπως με αυτόν του συστήματος ανάδρασης της γνωστής σελίδας

πωλήσεων του e-Bay. Η ιστοσελίδα αυτή παρέχει μια γενική αξιολόγηση για κάθε πωλητή ενός αντικειμένου με βάση τις βαθμολογίες άλλων χρηστών-πελατών για αυτόν για τέσσερις παραμέτρους, δημιουργώντας έτσι μια δημόσια προβαλλόμενη «φήμη», πχ. για έναν τυχαίο χρήστη παίρνουμε:

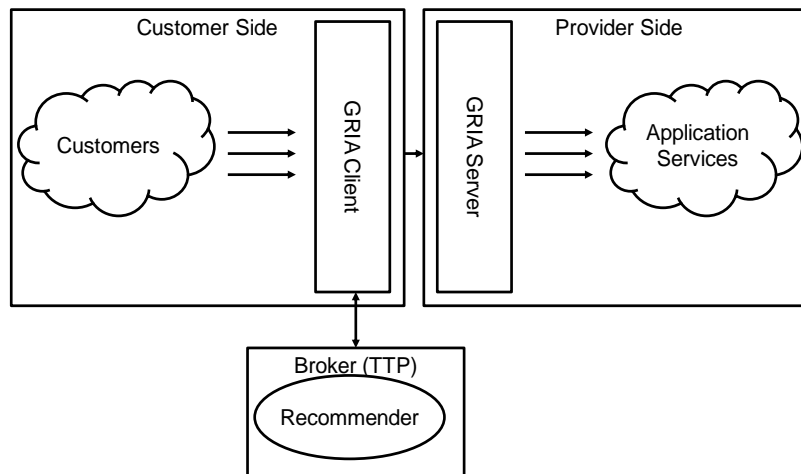
Criteria	Average Rating	Number of Ratings
Item as described	★★★★★	78058
Communication	★★★★☆	77914
Shipping time	★★★★★	81947
Shipping and handling charges	★★★★☆	78132

Πίνακας 7: Μέσες βαθμολογίες ενός τυχαίου πωλητή του e-Bay

Σε αυτό το σύστημα (όπως και στην προαναφερθείσα αγορά υπηρεσιών που εξετάζεται εδώ), κάθε πελάτης στηρίζεται στην άποψη των άλλων πελατών, όταν θέλει να αγοράσει ένα αντικείμενο (υπηρεσία).

Η κύρια διαφορά είναι ότι στη δική μας περίπτωση ο πελάτης δεν ξέρει τη μέση βαθμολογία της υπηρεσίας προκειμένου να την επιλέξει, αλλά η εφαρμογή πελάτη συλλέγει τις βαθμολογίες-αξιολογήσεις και τις χρησιμοποιεί για να υπολογίσει μια πρόβλεψη όσον αφορά την αξιολόγηση που θα έκανε ο ίδιος ο πελάτης. Τη λειτουργία αυτή μπορεί εύκολα να την αναλάβει ένας ενδιάμεσος που ενεργεί ως ένα έμπιστο τρίτο μέρος στην όλη διαδικασία. Ο ρόλος μιας τέτοιας ενδιάμεσης οντότητας είναι σημαντικότερος γιατί βοηθά στην αποφυγή ενός πολύ βασικού μειονεκτήματος των συστημάτων ανατροφοδότησης: ο σχεδιασμός της πολιτικής αξιολόγησης σε ένα σύστημα παρόμοιο του e-Bay επηρεάζει σοβαρά το επίπεδο εμπιστοσύνης και αποτελεσματικότητάς του και ως εκ τούτου συστήματα στα οποία οι προαναφερθείσες βαθμολογίες παραμένουν μυστικές περιορίζουν επιτηδευμένες αξιολογήσεις και αποδεικνύονται πιο αποτελεσματικά (Masclat & Rénard, 2008).

Το προαναφερθέν μοντέλο αποτελεί τη βάση για τη δημιουργία ενός μηχανισμού που λειτουργεί ως σύστημα συστάσεων κατά την επιλογή υπηρεσιών, βασισμένο σε συνεργατικές τεχνικές φιλτραρίσματος και ιδιαίτερα σε μεθόδους συσχέτισης (Pennock, Horvitz, Lawrence, & Giles, 2000). Αυτή η τεχνική υποστήριξης αποφάσεων για την επιλογή υπηρεσιών υλοποιείται ως υποστηρικτικός μηχανισμός σε ένα πραγματικό περιβάλλον που δίνει τη δυνατότητα παροχής υπηρεσιών σε μεγάλη κλίμακα. Μια γενική απεικόνιση του αυτού του περιβάλλοντος αναφοράς φαίνεται στο Σχήμα 20. Η απεικόνιση αυτή παρουσιάζει τη λειτουργία μιας Υπηρεσίας Συστάσεων (Recommender Service) για την υποστήριξη των πελατών σε ένα συγκεκριμένο μεσολογισμικό (GRIA) όπως θα εξηγηθεί στην παράγραφο που περιγράφει τα πειράματα, ωστόσο η αντίστοιχη αρχιτεκτονική είναι ομοίως λειτουργική και σε άλλα SaaS περιβάλλοντα.



Σχήμα 20: Γενική απεικόνιση της αρχιτεκτονικής αναφοράς

Η διαδικασία επιλογής των υπηρεσιών μπορεί να υποστηριχθεί από ένα σύστημα συστάσεων μέσω των κατάλληλων διεπαφών με τον πελάτη. Ωστόσο, όπως εξηγήθηκε στα προηγούμενα, το σύστημα αυτό βασίζεται σε συνεργατικές τεχνικές φιλτραρίσματος, οι οποίες συνεπάγονται την ανάγκη μιας βάσης συγκέντρωσης και ανάλυσης των

δεδομένων που προέρχονται από τους διάφορους πελάτες. Αυτός ο ρόλος αναλαμβάνεται από μια τρίτη οντότητα κοινής εμπιστοσύνης (Trusted Third Party - TTP) που συλλέγει όλα τα δεδομένα που χρειάζονται από τους πελάτες, προκειμένου να προχωρήσει στις ανάλογες συστάσεις. Τέλος, όταν η επιλογή των επιθυμητών υπηρεσιών ολοκληρώνεται και η ροή εργασίας είναι έτοιμη να εκτελεστεί, ο διακομιστής διεκπεραιώνει τις αιτήσεις για νέες εργασίες με την ανάθεση των κατάλληλων πόρων που υπάρχουν στην υποκείμενη υποδομή σε κάθε εργασία που είναι προγραμματισμένη να τρέξει.

Η προαναφερθείσα διαδικασία εξηγείται εν συντομία, καθώς στην πράξη υπεισέρχονται διάφορα θέματα διαχείρισης της ασφάλειας, της εμπιστοσύνης και των συμφωνιών SLA, αλλά θεωρούνται εκτός της εμβέλειας της παρούσας μεθόδου αξιολόγησης. Εδώ εστιάζουμε στην περιγραφή του πώς ένα σύστημα συστάσεων θα μπορούσε να προσφέρει υποστήριξη λήψης αποφάσεων, χρησιμοποιώντας συνεργατικές τεχνικές φιλτραρίσματος. Στη συνέχεια, το πρόβλημα τίθεται στη θεωρητική του διάσταση, έτσι ώστε να απεικονιστούν καλύτερα το όλο πλαίσιο της αξιολόγησης που ακολουθεί και το πιο σημαντικό, η συνεργατική τεχνική φιλτραρίσματος που προτείνεται.

5.2.1 Μοντελοποίηση του προβλήματος

Έστω ότι έχουμε ένα σύνολο l πελατών ενός υπηρεσιοστρεφούς περιβάλλοντος C_a , όπου $a \in [1, l]$. Ο κάθε ένας από αυτούς μπορεί να χρησιμοποιήσει ένα σύνολο n διαθέσιμων (εναλλακτικών) υπηρεσιών της ίδιας λειτουργικότητας, που αναπαρίσταται ως S_{ai} , όπου $i \in [1, n]$. Για κάθε υπηρεσία, ο πελάτης και ο πάροχος έχουν ένα μοναδικό

SLA, το οποίο καθορίζει m διαφορετικές παραμέτρους ποιότητας. Για κάθε πελάτη και υπηρεσία, αυτές οι παράμετροι αναπαρίστανται ως P_{aij} , όπου $j \in [1, m]$.

Κάθε φορά που μια υπηρεσία (S_{ai}) καλείται από έναν πελάτη (C_a), ο πελάτης πρέπει να αξιολογεί την παρεχόμενη ποιότητα, έτσι ώστε να δημιουργηθεί ένα ιστορικό αξιολογήσεων που θα χρησιμοποιηθεί στο μέλλον κατά τη διαδικασία επιλογής υπηρεσίας. Αυτή η αξιολόγηση συνίσταται απλά στη βαθμολόγηση κάθε παραμέτρου του SLA, που υποδεικνύει την ικανοποίηση του από κάθε μία από τις παραμέτρους ποιότητας, με βάση την εμπειρία και τις προτιμήσεις του. Επομένως, η αξιολόγηση m παραμέτρων κάθε υπηρεσίας S_{ai} μπορεί να αναπαρασταθεί σαν ένα διάνυσμα βαθμολογιών: $E_{ai} = [r_1, r_2, \dots, r_m]$, όπου το r_v δείχνει τη βαθμολογία για τη v -ιοστή παράμετρο του SLA από τον C_a . Στην περίπτωση που περισσότερες από μια βαθμολογίες έχουν παραχθεί για την S_{ai} από τον C_a , οι μέσες τιμές των βαθμολογιών τίθενται ως στοιχεία του διανύσματος αξιολόγησης: $E_{ai} = [\bar{r}_1, \bar{r}_2, \dots, \bar{r}_m]$.

5.2.2 Πρόβλεψη αξιολόγησης με χρήση συνεργατικών τεχνικών φιλτραρίσματος και επιλογή υπηρεσίας

Ας υποθέσουμε τώρα ότι δύο πελάτες C_a και C_u έχουν χρησιμοποιήσει τουλάχιστον μία κοινή υπηρεσία, το οποίο σημαίνει σύμφωνα με τους παραπάνω συμβολισμούς ότι: $S_a \cap S_u = I \neq \emptyset$. Επομένως, για κάθε κοινή υπηρεσία οι αξιολογήσεις μπορούν να συγκριθούν, έτσι ώστε να γίνει αντιληπτό πόσο μοιάζουν οι απαιτήσεις των πελατών κι επίσης πόσο παρόμοια είναι τα επίπεδα ικανοποίησής τους. Αυτό είναι ιδιαίτερα χρήσιμο, γιατί αν δύο πελάτες έχουν μεγάλη πιθανότητα σύγκλισης για κάθε υπηρεσία του I , τότε θα βαθμολογήσουν πιθανότατα με τον ίδιο τρόπο τις υπηρεσίες που

υπάρχουν στο σύνολο I' , όπου $I \cap I' = \emptyset$ (Breese, Heckerman, & Kadie, 1998). Με όσους πιο πολλούς άλλους πελάτες εμφανίζει ένας παρόμοιος αξιολογήσεις, τόσο μεγαλύτερη πιθανότητα υπάρχει να προβλέψουμε τις αξιολογήσεις του για το αντίστοιχο σύνολο I' . Έχοντας ως αναφορά έναν πελάτη C_a , για τον οποίο θέλουμε να προβλέψουμε την αξιολόγησή του για μια συγκεκριμένη υπηρεσία, ακολουθούμε τα παρακάτω 3 βήματα (Αλγόριθμος I):

1. Μετράμε πόσο όμοια είναι τα διανύσματα αξιολογήσεών του με τους υπόλοιπους πελάτες που έχουν χρησιμοποιήσει τις ίδιες υπηρεσίες, υπολογίζοντας ένα συντελεστή συσχέτισης για κάθε στοιχείο του διανύσματος αξιολόγησης. Αυτό αποτελεί δείκτη για το πόσο ισχυρή γραμμική σχέση έχει ο πελάτης αυτός με κάθε έναν από τους υπόλοιπους.
2. Χρησιμοποιώντας τους πελάτες με την μεγαλύτερη συσχέτιση (πελάτες γειννίας), υπολογίζουμε μια πρόβλεψη της βαθμολογίας του για μια συγκεκριμένη παράμετρο, από έναν σταθμισμένο συνδυασμό των βαθμολογιών των πελατών γειννίας.
3. Επαναλαμβάνουμε τα βήματα 1) και 2) για κάθε παράμετρο του SLA της ζητούμενης υπηρεσίας για τον πελάτη αυτό.

Στο πρώτο βήμα του Αλγορίθμου I, υπολογίζεται η ομοιότητα μεταξύ των δύο πελατών για κάθε παράμετρο του SLA, με χρήση του συντελεστή συσχέτισης του Pearson, που ορίζεται παρακάτω:

$$c_{au} = \frac{\sum_{i=1}^h (r_{ai} - \bar{r}_a) \times (r_{ui} - \bar{r}_u)}{\sqrt{\sum_{i=1}^h (r_{ai} - \bar{r}_a)^2 \times \sum_{i=1}^h (r_{ui} - \bar{r}_u)^2}}$$

Εξίσωση 1: Ορισμός του συντελεστή συσχέτισης του Pearson για μια παράμετρο

Στην Εξίσωση 1, το $h = |I_{au}|$, αναπαριστά τον πλήθος των υπηρεσιών που έχουν αξιολογηθεί από τον C_a και τον C_u (το σύνολο των υπηρεσιών που αξιολογήθηκαν κι από τους δύο υποδηλώνεται εδώ ως I_{au} και το συμπληρωματικό του ως I'_{au} για λόγους αναγνωσιμότητας), το r_{ai} είναι η βαθμολογία του C_a για το συγκεκριμένο όρο του SLA

της υπηρεσίας S_i και το $\bar{r}_a = average(r_{a1j}, r_{a2j}, \dots, r_{ahj})$ είναι η μέση βαθμολογία του πελάτη C_a για τη συγκεκριμένη παράμετρο j όλων των υπηρεσιών στο I_{au} . Ο συντελεστής συσχέτισης του Pearson καθορίζει το βαθμό "αναλογικότητας" δύο μεταβλητών. Στην περίπτωση που ο συντελεστής αυτός είναι μηδέν, τότε υπάρχει μηδαμινή γραμμική συσχέτιση μεταξύ των δύο μεταβλητών.

Έχοντας υπολογίσει το συντελεστή συσχέτισης c_{au} για κάθε παράμετρο μεταξύ των πελατών, μπορούμε να παράγουμε προβλέψεις για τον πελάτη στην κάθε παράμετρο της υπο διερεύνηση υπηρεσίας S_{ai} , βάσει της σταθμισμένης μέσης τιμής των αποκλίσεων από τη μέση τιμή των υπολοίπων πελατών για την παράμετρο αυτή:

$$p_{ai} = \bar{r}_a + \frac{\sum_{u=1}^g (r_{ui} - \bar{r}_u) \times c_{au}}{\sum_{u=1}^g c_{au}}$$

Εξίσωση 2: Πρόβλεψη αξιολόγησης για μια παράμετρο με βάση τους συντελεστές συσχέτισης

όπου g είναι ο αριθμός των πελατών γειννίας του C_a που έχουν χρησιμοποιήσει την υπηρεσία S_{ai} και p_{ai} η προβλεπόμενη βαθμολογία μιας παραμέτρου της S_{ai} για τον πελάτη C_a . Η πρόβλεψη αυτή μπορεί να οριστεί σε ένα γενικότερο πλαίσιο ως p_{aij} , όπου στην περίπτωση αυτή το j καθορίζει τη συγκεκριμένη παράμετρο της υπηρεσίας S_{ai} .

Η παραπάνω διαδικασία επαναλαμβάνεται για κάθε παράμετρο ποιότητας του SLA της υπηρεσίας S_i , ώστε να υπολογιστούν προβλέψεις για όλες τις βαθμολογίες - αξιολογήσεις. Σαν αποτέλεσμα προκύπτει ένα διάγραμμα:

$$PR_{ai} = \begin{bmatrix} P_{ai1} \\ P_{ai2} \\ \vdots \\ P_{aim} \end{bmatrix}$$

Το διάνυσμα αυτό εμπεριέχει την πρόβλεψη για την αξιολόγηση του πελάτη C_a για μια συγκεκριμένη υπηρεσία με προκαθορισμένες απαιτήσεις ποιότητας που περιλαμβάνονται στο SLA.

Συγκρίνοντας τα διάφορα διανύσματα πρόβλεψης PR για τις υπηρεσίες του ίδιου τύπου, σύμφωνα με την επιχειρηματική πολιτική του πελάτη, μπορούμε να συμπεράνουμε ποια συγκεκριμένη υπηρεσία πρέπει να χρησιμοποιήσει. Η αναφερθείσα επιχειρηματική πολιτική μπορεί για παράδειγμα να καθορίζει σε ποιους συγκεκριμένους όρους του SLA πρέπει να δοθεί έμφαση ή ότι η επικρατούσα υπηρεσία πρέπει να είναι αυτή με τις πιο υψηλές τιμές στους όρους. Για να παραμετροποιήσουμε αυτήν την πολιτική χρησιμοποιώντας το συγκεκριμένο μοντέλο του προβλήματος που παρουσιάστηκε παραπάνω, τα προκύπτοντα διανύσματα πρέπει να συγκριθούν σταθμίζοντας κάθε παράμετρο e με ένα βάρος w_e , έτσι ώστε να καθορίσουμε τη σημασία κάθε όρου του SLA, σύμφωνα με τις προτιμήσεις του πελάτη. Η σύγκριση αυτή λαμβάνει χώρα χρησιμοποιώντας ένα διάνυσμα αναφοράς, το ιδανικό διάνυσμα αξιολόγησης R_{opt} . Το R_{opt} αναπαριστά την περίπτωση στην οποία ο πελάτης είναι πλήρως ικανοποιημένος, κι επομένως οι τιμές για όλες τις διαστάσεις του R_{opt} είναι οι μέγιστες:

$$R_{opt} = \max \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix}$$

Η Ευκλείδεια νόρμα της διαφοράς μεταξύ του σταθμισμένου διανύσματος πρόβλεψης WPR_{ai} και του διανύσματος αναφοράς R_{opt} χρησιμοποιείται σαν μέτρο

σύγκρισης μεταξύ των υπηρεσιών, για να επιλεγεί η καταλληλότερη για τον πελάτη C_a . Η νόρμα αυτή παρέχει το μήκος (μέγεθος) της διαφοράς μεταξύ της πρόβλεψης και της ιδανικής περίπτωσης για κάθε υπηρεσία S_i :

$$\begin{aligned} diff_{ai} = \|R_{opt} - WPR_{ai}\| &= \left\| \max \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix} - \begin{bmatrix} w_1 * p_{ai1} \\ w_2 * p_{ai2} \\ \vdots \\ w_m * p_{aim} \end{bmatrix} \right\| = \left\| \begin{bmatrix} \max r_1 - w_1 * p_{ai1} \\ \max r_2 - w_2 * p_{ai2} \\ \vdots \\ \max r_m - w_m * p_{aim} \end{bmatrix} \right\| = \left\| \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \right\| \Rightarrow \\ \Rightarrow diff_{ai} &= \sqrt{b_1^2 + \dots + b_m^2} \end{aligned}$$

Εξίσωση 3: Ευκλείδεια νόρμα διαφοράς σταθμισμένου διανύσματος πρόβλεψης από το ιδανικό

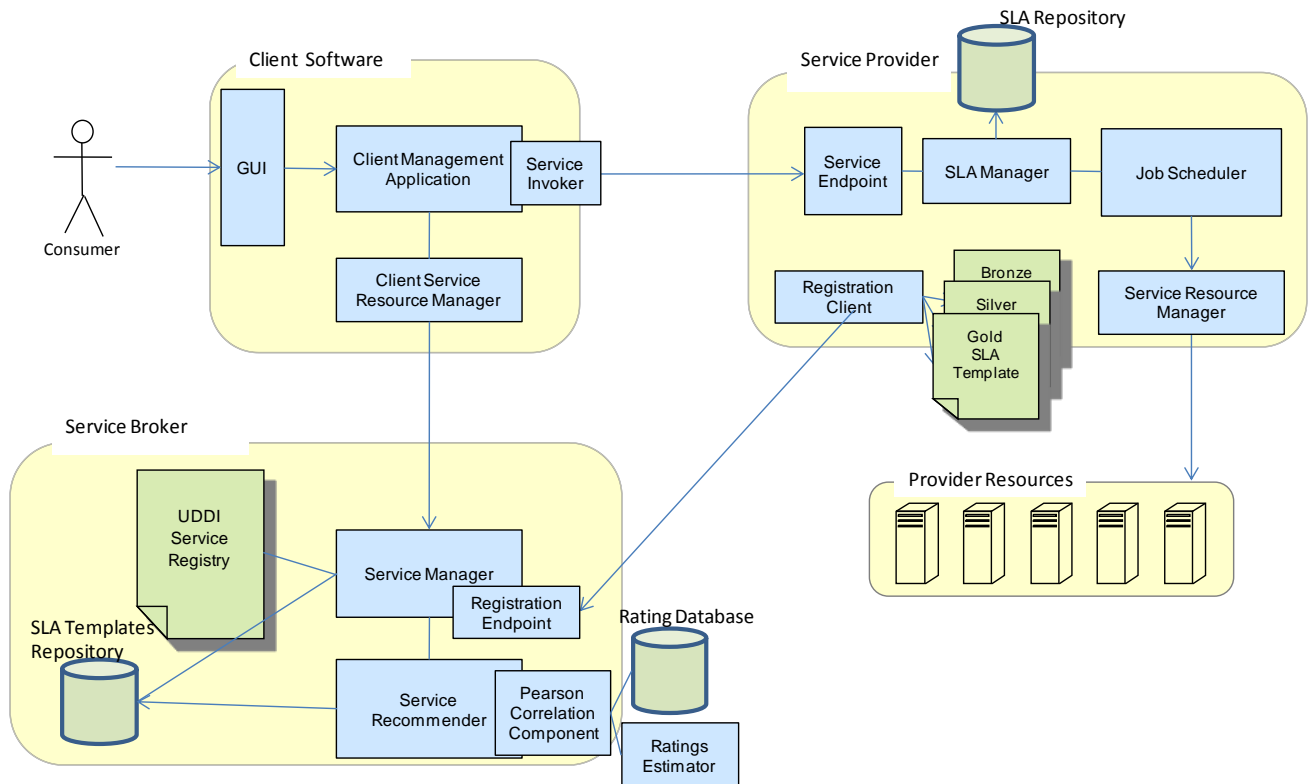
Έχοντας το παραπάνω μέτρο, η μικρότερη διαφορά $diff_{ai}$ αναδεικνύει την καλύτερη υπηρεσία S_{ai} για να επιλεγεί από τον πελάτη C_a . Στα επόμενα, με σημείο αναφοράς το παρουσιασθέν μοντέλο, υλοποιούμε την προαναφερθείσα μέθοδο αξιολόγησης και επιλογής υπηρεσίας σε έναν αλγόριθμο, ο οποίος ενθυλακώνεται σε μια βοηθητική διαδικτυακή υπηρεσία που αποτελεί μέρος του κύκλου ζωής των SaaS υπηρεσιών στο περιβάλλον αναφοράς. Η αρχιτεκτονική που θα παρουσιασθεί είναι ανεξάρτητη της πλατφόρμας, ακολουθώντας τους γενικούς υπηρεσιοστρεφείς κανόνες.

5.3

Αρχιτεκτονική Υλοποίησης

Η παράγραφος αυτή παρουσιάζει τη γενική αρχιτεκτονική υλοποίησης για την υποστήριξη του SR μηχανισμού που αναλύθηκε (Σχήμα 21). Η αρχιτεκτονική είναι σχεδιασμένη βάσει των αρχών SOA και εστιάζει σε δύο στάδια της διαχείρισης του κύκλου ζωής των υπηρεσιών, όπως αναφέρθηκε: στα στάδια Επιλογής και Αξιολόγησης Υπηρεσίας. Έτσι, παρέχεται μια πιο λεπτομερής εικόνα για το πώς ο μηχανισμός SR λειτουργεί στο πλαίσιο μιας SOA υποδομής, προκειμένου να υποστηρίξει τους καταναλωτές στην επιλογή των κατάλληλων υπηρεσιών. Για λόγους αναγνωσιμότητας

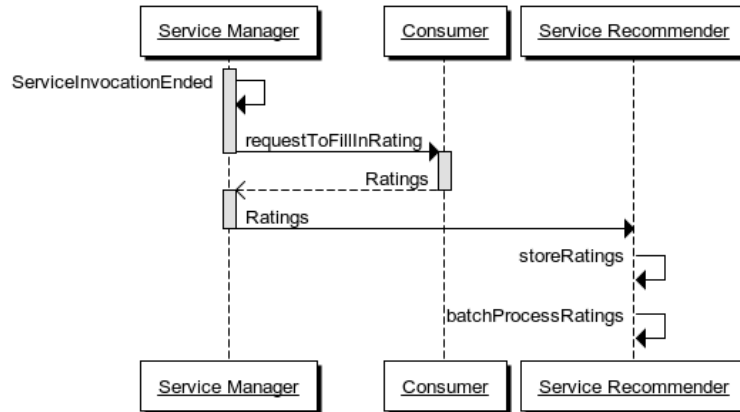
και σαφήνειας, η περιγραφή του συστήματος εδώ δεν εστιάζει στη δημοσίευση των υπηρεσιών και στις φάσεις παρακολούθησης που είναι έξω από το πλαίσιο της αξιολόγησης από το μέρος του πελάτη.



Σχήμα 21: Γενική Αρχιτεκτονική Υλοποίησης Μηχανισμού SR

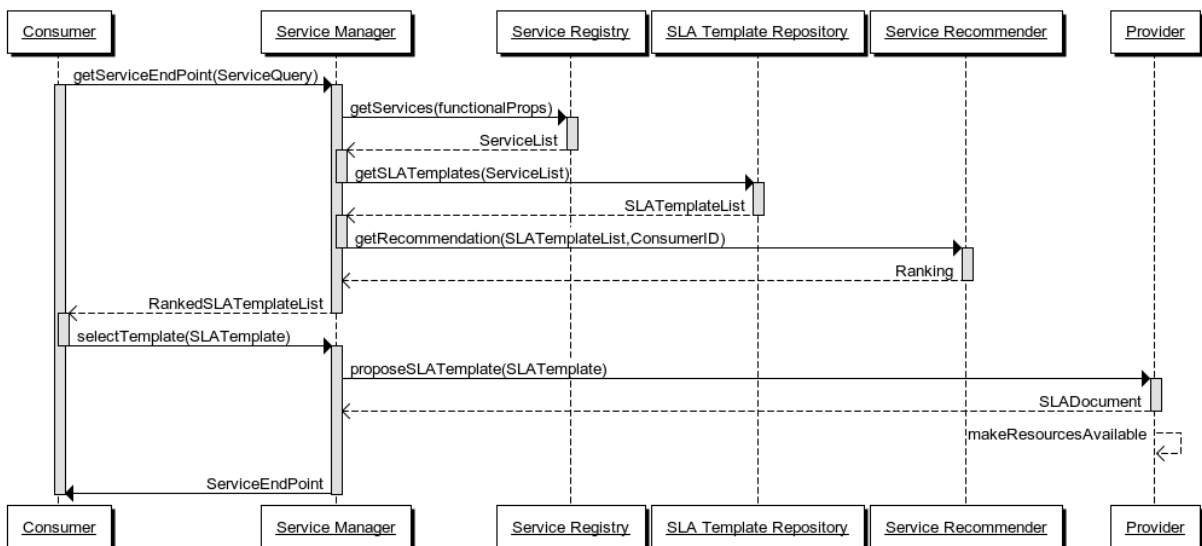
5.3.1 Αλληλεπιδράσεις κατά την αξιολόγηση και επιλογή υπηρεσίας

Το Σχήμα 22 παρουσιάζει τις αλληλεπιδράσεις μεταξύ των στοιχείων της αρχιτεκτονικής κατά τη φάση Αξιολόγησης Υπηρεσίας, μετά το πέρας της κλήσης:



Σχήμα 22: Ακολουθιακό διάγραμμα αλληλεπιδράσεων κατά την Αξιολόγηση Υπηρεσίας

Ακολούθως, το Σχήμα 23 παρουσιάζει την ακολουθία των αλληλεπιδράσεων μεταξύ των στοιχείων της αρχιτεκτονικής στη φάση της Επιλογής Υπηρεσίας:



Σχήμα 23: Ακολουθιακό διάγραμμα αλληλεπιδράσεων κατά την Επιλογή Υπηρεσίας

Στη συνέχεια παρουσιάζουμε συνοπτικά τα εμπλεκόμενα μέρη. Ως γενικό σχόλιο, αξίζει να αναφερθεί ότι όλες οι οντότητες που απεικονίζονται στο Σχήμα 22 και στο Σχήμα 23, εκτός από τον πάροχο (Provider) και τον καταναλωτή (Consumer), ανήκουν συνήθως στον ενδιάμεσο (Service Broker), όπως φαίνεται και στη συνολική αρχιτεκτονική που δόθηκε στο Σχήμα 21.

5.3.2 Αλληλεπιδρούσες οντότητες

Οι οντότητες που αλληλεπιδρούν στην παραπάνω ακολουθία είναι οι εξής:

- **Consumer:** Η οντότητα του καταναλωτή εδώ, χάριν απλότητας, συμπεριλαμβάνει και τον τελικό χρήστη των υπηρεσιών και το λογισμικό πελάτη που παρέχει διεπαφές για την έκφραση των απαιτήσεων του τελικού χρήστη κατά την Ανακάλυψη Υπηρεσίας και για το χειρισμό της διαδικασίας διαπραγμάτευσης του SLA.
- **Service Manager:** Η βασική οντότητα που διαχειρίζεται τις αιτήσεις από τους καταναλωτές και καλεί τους παρόχους για λογαριασμό τους. Ο Service Manager επικοινωνεί με τις υπηρεσίες του service broker, προκειμένου να παρέχει μια διεπαφή εξυπηρέτησης για τον καταναλωτή.
- **Service Registry:** Ένα μητρώο στο οποίο καταχωρούνται οι περιγραφές των υπηρεσιών από τους παρόχους, ώστε να καταστούν διαθέσιμες στους καταναλωτές, κατά τη φάση ανακάλυψης των υπηρεσιών. Συνήθως αυτή τη δουλειά την αναλαμβάνει ένα μητρώο UDDI.
- **SLA Template Repository:** Μια βάση δεδομένων που αποθηκεύει τα πρότυπα SLAs για κάθε υπηρεσία που καταχωρούν οι πάροχοι. Για μια υπηρεσία μπορεί να υπάρχουν πολλαπλά SLA Templates που ουσιαστικά αντιπροσωπεύουν

- διαφορετικές προσφορές των παρόχων. Οι όροι στα πρότυπα SLAs μιας υπηρεσίας συνήθως διαφέρουν ως προς την ποιότητα που υπόσχονται και το σχετικό κόστος και μπορεί να κατηγοριοποιηθούν συνήθως σε τρεις διακριτές προσφορές: Gold, Silver και Bronze SLA.
- **Provider:** Η οντότητα του παρόχου εδώ, επίσης χάριν απλότητας, αποτελείται από τις διεπαφές του παρόχου υπηρεσιών στη διαδικασία διαπραγμάτευσης του SLA.
 - **Service Recommender:** Η οντότητα που βάσει μίας λίστας προτύπων SLAs και ενός αναγνωριστικού καταναλωτή, παρέχει μια λίστα κατάταξης των SLAs με βάση όλου του προηγούμενου ιστορικού για τις παραμέτρους QoS που αφορούν αυτά τα SLAs. Κάθε φορά που ο χρόνος ισχύος του SLA λήγει (δηλαδή σταματά η πρόσβαση στους πόρους του παρόχου για τον καταναλωτή), ο καταναλωτής καλείται να δηλώσει την ικανοποίησή του από την παρεχόμενη ποιότητα και να βαθμολογήσει τις QoS παραμέτρους που είχε συμφωνήσει με τον πάροχο (βλ. Σχήμα 22). Αναλύοντας τη σχέση όλων των καταναλωτών και των βαθμολογιών τους, η οντότητα είναι σε θέση να προτείνει μια κατάταξη των διαθέσιμων υπηρεσιών στον κάθε καταναλωτή.

5.3.3 Ροή των δεδομένων

Όπως απεικονίζει το Σχήμα 23, η ροή ξεκινά όταν η πλευρά του καταναλωτή κάνει μια αναζήτηση (*serviceQuery*) στο service broker για μια υπηρεσία. Θεωρούμε ότι ο καταναλωτής είναι εγγεγραμμένος στη βάση του service broker πριν από την πραγματοποίηση μιας τέτοιας κλήσης (η διαδικασία εγγραφής δεν περιγράφεται εδώ, μην προσφέροντας κάτι στην όλη ανάλυση). Η αναζήτηση περιλαμβάνει τις λειτουργικές

ιδιότητες (*functionalProps*) της υπό εύρεση υπηρεσίας, σύμφωνα με τις προδιαγραφές ενός ερωτήματος UDDI, που θα μεταδοθεί στη συνέχεια στο μητρώο UDDI.

Σε ένα τυπικό σενάριο εγγραφής στο UDDI, ο πάροχος δημοσιεύει μια υπηρεσία, την καταχωρεί στο μητρώο και ορίζει ένα πρότυπο δέσμευσης με τεχνικές πληροφορίες για την υπηρεσία. Το πρότυπο δέσμευσης παρέχει μια αναφορά σε ένα ή περισσότερα tModels, τα οποία αντιπροσωπεύουν αφηρημένες διεπαφές της διαδικτυακής υπηρεσίας. Τα tModels μπορεί να δημοσιεύονται μοναδικά από τον πάροχο, με πληροφορίες σχετικά με τις διεπαφές και τις αναφορές URL στο έγγραφο WSDL της υπηρεσίας του. Ως εκ τούτου, ο σκοπός της αναζήτησης από τον πελάτη είναι να βρει μια υλοποίηση μιας γνωστής διεπαφής. Με άλλα λόγια, ο πελάτης έχει ένα αναγνωριστικό tModel και επιδιώκει πρότυπα δέσμευσης με αναφορά σε αυτό το tModel.

Τα αποτελέσματα της αναζήτησης είναι μια λίστα με υπηρεσίες (*serviceList*), δηλαδή ένας κατάλογος αναγνωριστικών διαδικτυακών υπηρεσιών (*bindingTemplate IDs*) που επιστρέφονται στην οντότητα διαχείρισης των υπηρεσιών. Στη συνέχεια, ο Service Manager μετατρέπει το *serviceList* σε ένα XML και το προωθεί στην αποθήκη Προτύπων SLA (SLA Template Repository), έτσι ώστε να ανακτήσει τον κατάλογο των SLATemplates (*SLATemplateList*) που συνδέονται με αυτές τις διαδικτυακές υπηρεσίες. Εν συνεχεία, το *SLATemplateList* αναμεταδίδεται προς την οντότητα SR, μαζί με το *consumerID* που καθορίζει μοναδικά τον καταναλωτή που κίνησε τη διαδικασία. Ο SR μηχανισμός αναμένεται να απαντήσει με ένα *rankedSLATemplateList*, δηλαδή τη λίστα των δοθέντων SLA Templates, κατατασσόμενων με βάση την προηγούμενη αξιολόγηση παρόμοιων καταναλωτών για τις πιθανές υπηρεσίες.

Ακολουθώντας, οι καταναλωτές χρησιμοποιούν το *rankedSLATemplateList* προκειμένου να εξετάσουν τις επιλογές τους. Μπορούν είτε να λάβουν υπόψη ή να αγνοήσουν τη σύσταση του SR μηχανισμού στην επιλογή του προτύπου SLA (και άρα της υπηρεσίας) της προτίμησής τους. Μόλις το επιλέξουν, ο Service Manager το προτείνει στον πάροχο και εφόσον αυτός το δεχτεί, αποστέλλει την τελική συμφωνία (*SLADocument*). Με τον τερματισμό του SLA, είτε λόγω παραβίασης της σύμβασης ή λόγω λήξης της διάρκειας ζωής του SLA, ο Service Manager ζητάει από τον καταναλωτή να αξιολογήσει την ποιότητα υπηρεσίας, σύμφωνα με τις παραμέτρους που ορίζονται στο SLA (Σχήμα 22). Αυτό γίνεται βαθμολογώντας κάθε μία από τις QoS παραμέτρους, εκφράζοντας την πεποίθηση του κατά πόσο τα κατώτατα όρια που είχε υποσχεθεί ο πάροχος διατηρήθηκαν. Οι βαθμολογίες (Ratings) στη συνέχεια προωθούνται στην SR οντότητα, που τις αποθηκεύει και τις επεξεργάζεται, προκειμένου να είναι σε θέση να παρέχει ακόμη πιο αποτελεσματικές συστάσεις.

Οι πληροφορίες που το Service Recommender θα πρέπει να διατηρήσει, εκτός από τις βαθμολογίες, είναι οι όροι αξιολόγησης του SLA, η ταυτότητα του καταναλωτή και οι λεπτομέρειες των διαδικτυακών υπηρεσιών (συμπεριλαμβανομένου του παρόχου). Για την απλοποίηση αυτής της διαδικασίας, ο Service Manager μεταδίδει στο SR ολόκληρο το έγγραφο SLA ενσωματώνοντας τις αξιολογήσεις των καταναλωτών. Για το σκοπό αυτό, στα πλαίσια αυτού του μοντέλου, δημιουργήθηκε μια ελαφρώς τροποποιημένη έκδοση του WS-Agreement, που περιλαμβάνει ένα νέο χαρακτηριστικό "rating" κάτω από κάθε στοιχείο "ServiceTerm" ή "Guarantee Term" της προδιαγραφής WS-Agreement, όπως φαίνεται για παράδειγμα στον Πίνακα 8:

```

<xs:complexType name="GuaranteeTermType">
  <xs:complexContent>
    <xs:extension base="wsag:TermType">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0"
name="ServiceScope" type="wsag:ServiceSelectorType"/>
        <xs:element minOccurs="0"
ref="wsag:QualifyingCondition"/>
        <xs:element ref="wsag:ServiceLevelObjective"/>
        <xs:element name="BusinessValueList"
type="wsag:BusinessValueListType"/>
      </xs:sequence>
      <xs:attribute name="Obligated"
type="wsag:ServiceRoleType"/>
      <xs:attribute name="rating" type="xs:integer"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Πίνακας 8: Επέκταση του σχήματος του WS-Agreement με αξιολογήσεις στους όρους του (παρουσιάζεται μόνο το τροποποιημένο στοιχείο GuaranteeTerm)

Έτσι, η βάση δεδομένων του SR, αποθηκεύει όλη τη συμφωνία WS-Agreement μαζί με τις βαθμολογίες.

Τα οφέλη από την επέκταση του WS-Agreement αντί της χρησιμοποίησης ενός πολύ απλούστερου μοντέλου δεδομένων για το μηχανισμό SR είναι πολλαπλά:

- Το WS-Agreement είναι μια παγκόσμια προδιαγραφή, που ευρέως υιοθετείται από πολλούς φορείς παροχής υπηρεσιών.
- Υπάρχουν ελάχιστες απαιτήσεις επεξεργασίας από το λογισμικό πελάτη και περιορίζονται στην προσθήκη της αξιολόγησης σε κάθε όρο SLA.
- Ο SR μηχανισμός υιοθετεί το σύστημα καταχώρησης που χρησιμοποιείται ήδη στο WS-Agreement (το Agreement-Initiator ID είναι το αναγνωριστικό του καταναλωτή).
- Οι λεπτομέρειες της υπηρεσίας (π.χ. το αναγνωριστικό του παρόχου) περιλαμβάνονται ήδη στο έγγραφο, οπότε δεν μένει παρά η αντιστοίχισή τους σε ένα στοιχείο του UDDI με απλή διασταύρωση.

5.4

Πειράματα

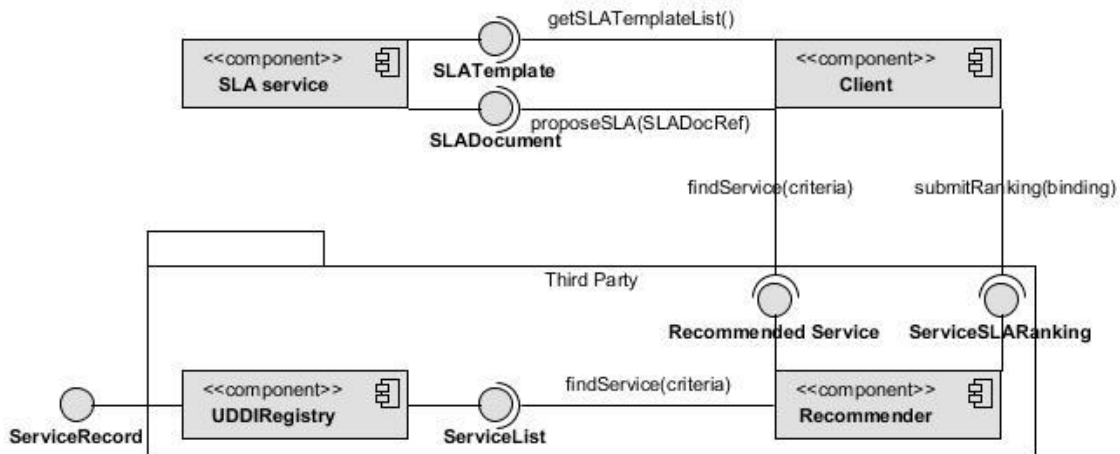
Αυτή η ενότητα περιγράφει τα αποτελέσματα που προκύπτουν από την εφαρμογή, τη δοκιμή και την παρατήρηση του προαναφερθέντος μηχανισμού ως ένα σύστημα συστάσεων στην αρχιτεκτονική αναφοράς. Έτσι, στα παρακάτω εξηγούμε πώς το σύστημα συστάσεων αξιολογήθηκε σε μια αρχιτεκτονική που ακολουθεί το μοντέλο που παρουσιάστηκε, τον τρόπο με τον οποίο δημιουργήθηκε το σύνολο των πειραματικών δεδομένων και τέλος, παρουσιάζεται μια διεξοδική αξιολόγηση των αποτελεσμάτων.

5.4.1 Περιβάλλον αξιολόγησης

Το περιβάλλον αξιολόγησης της προτεινόμενης μεθόδου βασίζεται στις αρχές της αρχιτεκτονικής που περιγράφεται από τους (Snelling & Anjomshoaa, 2007), όντας ένα επιχειρηματικά προσανατολισμένο περιβάλλον παροχής υπηρεσιών με γνώμονα τα SLAs. Ειδικότερα, είναι ένα Software as a Service περιβάλλον χτισμένο πάνω στο μεσολογισμικό GRIA, στο οποίο οι εφαρμογές διατίθενται για πρόσβαση από τους πελάτες μέσω ειδικών διεπαφών. Το GRIA αντιμετωπίζει τις μη λειτουργικές απαιτήσεις όπως η ασφάλεια και η διαχείριση των SLAs, που στο GRIA βασίζονται στο πρότυπο WS-Agreement, με τη χρήση εξελιγμένων μηχανισμών στον πάροχο, όσο και στην πλευρά των πελατών. Ωστόσο, ακόμα κι αν το GRIA επιλέχθηκε ως η αρχιτεκτονική αναφοράς, η γενική μεθοδολογία που ακολουθούμε είναι ανεξάρτητη του μεσολογισμικού και θα μπορούσε να εφαρμοστεί στα περισσότερα περιβάλλοντα SaaS.

Στα πλαίσια της γενικής αρχιτεκτονικής αναφοράς του συστήματος που απεικονίζεται παραπάνω στο Σχήμα 20, οι πελάτες χρησιμοποιούν ειδικές διεπαφές που παρέχονται από το λογισμικό πελάτη GRIA. Ο πελάτης GRIA προσφέρει την απαραίτητη λειτουργικότητα, ώστε να καθοδηγεί το χρήστη σε όλο τον κύκλο ζωής κατά

την υποβολή μιας νέας εργασίας, δηλαδή στην ανακάλυψη και η επιλογή των υπηρεσιών, στη διαπραγμάτευση των όρων του SLA και στην κλήση της τελικής υπηρεσίας από το μέρος του πελάτη. Προσφέρει επίσης ένα γραφικό περιβάλλον στον χρήστη, αλλά μπορεί να είναι προσβάσιμος και μέσω μιας δικτυακής πύλης ή οποιουδήποτε άλλου γραφικού περιβάλλοντος. Επίσης, προσθέτοντας και το στάδιο της Αξιολόγησης Υπηρεσίας στο τέλος, ο πελάτης GRIA υλοποιεί και την εν λόγω λειτουργικότητα, με προσθήκη βαθμολογιών σε κάθε όρο του SLA μετά το πέρας της κλήσης.



Σχήμα 24: Διεπαφές πελάτη GRIA για εξεύρεση και αξιολόγηση υπηρεσιών

5.4.2 Μέθοδος Αξιολόγησης του προτεινόμενου μηχανισμού

Το πρώτο βήμα για την αξιολόγηση του μηχανισμού είναι η δημιουργία ενός μεγάλου συνόλου πειραματικών δεδομένων. Ωστόσο, λόγω των αχανών συνθηκών στην αγορά που αναφερόμαστε, αλλά και των περιορισμών ασφάλειας και εμπιστευτικότητας που τίθενται από πραγματικούς επιχειρηματικούς πελάτες, δεν υπάρχει επαρκής αριθμός διαθέσιμων πελατών και παρόχων, προκειμένου να δημιουργηθεί ένα ογκώδες πραγματικό σύνολο πειραματικών δεδομένων. Ως εκ τούτου, το σύνολο των δεδομένων που χρησιμοποιήθηκε στα αρχικά πειράματά μας δημιουργήθηκε με τέτοιο τρόπο ώστε

να προσομοιώνεται όσο το δυνατόν περισσότερο ένα ρεαλιστικό σενάριο. Το σενάριο περιλάμβανε την παροχή διαφόρων υπηρεσιών που χρησιμοποιούν το μεσολογισμικό GRIA σε χρήστες που επιθυμούν να δημιουργήσουν 3D video animation. Ο πελάτης καλεί μια διαδικτυακή υπηρεσία για τη μετατροπή ενός παραμετροποιημένου προσχέδιου (wireframe) σε ένα επιθυμητό καρέ βίντεο και στη συνέχεια σε ένα ολοκληρωμένο video stream. Για τη δημιουργία του συνόλου δεδομένων, οι αξιολογήσεις ενός μικρού αριθμού πραγματικών επιχειρηματικών πελατών καταγράφηκαν για μια σειρά υπηρεσιών και συγκεκριμένων παραμέτρων.

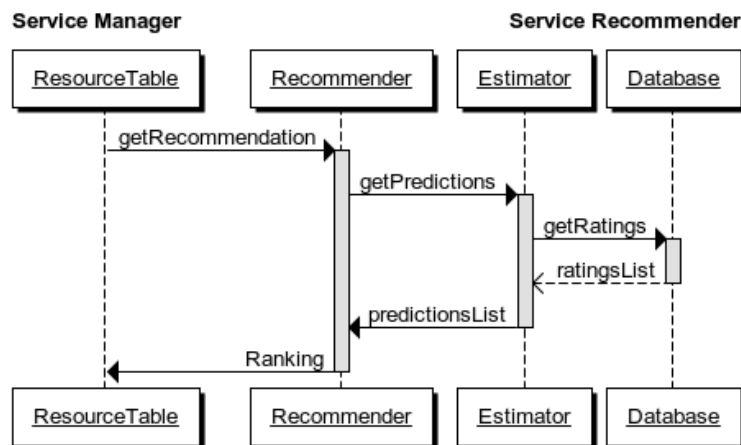
Εκτεταμένες αξιολογήσεις δημιουργήθηκαν χρησιμοποιώντας το εργαλείο StatAssist της Mathwave (StatAssist, Intelligent Analytics, 2010), παράγοντας ένα σύνολο δεδομένων κατάλληλου μεγέθους. Ένα μέρος του συνόλου αυτού χρησιμοποιήθηκε ως σύνολο αξιολόγησης, ώστε να μετρηθεί η απόκλιση των προβλεπόμενων από τις πραγματικές (ρεαλιστικά παραγόμενες) τιμές αλλά και από τις μέσες τιμές, οι οποίες χρησιμοποιούνται ως τιμές καθολικής αξιολόγησης για όλους τους πελάτες σε συστήματα τύπου e-Bay. Κατόπιν της αξιολόγησης της λειτουργίας της πειραματικής διάταξης για το προσομοιωμένο σύνολο, σαν πρακτική επαλήθευση, ζητήθηκε από τους επιχειρηματικούς πελάτες η κλήση και αξιολόγηση διαφόρων υπηρεσιών με τα αντίστοιχα SLAs, ώστε να συγκριθούν οι πραγματικές βαθμολογίες τους με τις προβλέψεις του SR συστήματος για αυτούς και να αξιολογηθεί η καταλληλότητα των συστάσεων από τον SR μηχανισμό.

Η δημιουργηθείσα Υπηρεσία Συστάσεων (βλ. Σχήμα 20) υλοποιήθηκε με χρήση της γλώσσας C++, δεδομένου ότι χρειάζεται ένα ελαφρύ λογισμικό με γρήγορη απόκριση για αυτούς τους υπολογισμούς. Ο υπολογισμός των συντελεστών συσχέτισης

Pearson για l πελάτες και n διαθέσιμες υπηρεσίες έχει χρονική πολυπλοκότητα $O(n + l^2)$ (Zhang, Bai, & Gao, 2009) και δεδομένου ότι υπολογίζουμε τους συντελεστές αυτούς για m ποιοτικές παραμέτρους, η συνολική πολυπλοκότητα εκτοξεύεται σε $O(m * (n + l^2))$, οπότε γίνεται σαφής η ανάγκη ενός γρήγορου λογισμικού. Η SR υπηρεσία καλεί περιοδικά την κλάση Estimator (Εκτιμητής), η οποία παρέχει δύο βασικές μεθόδους:

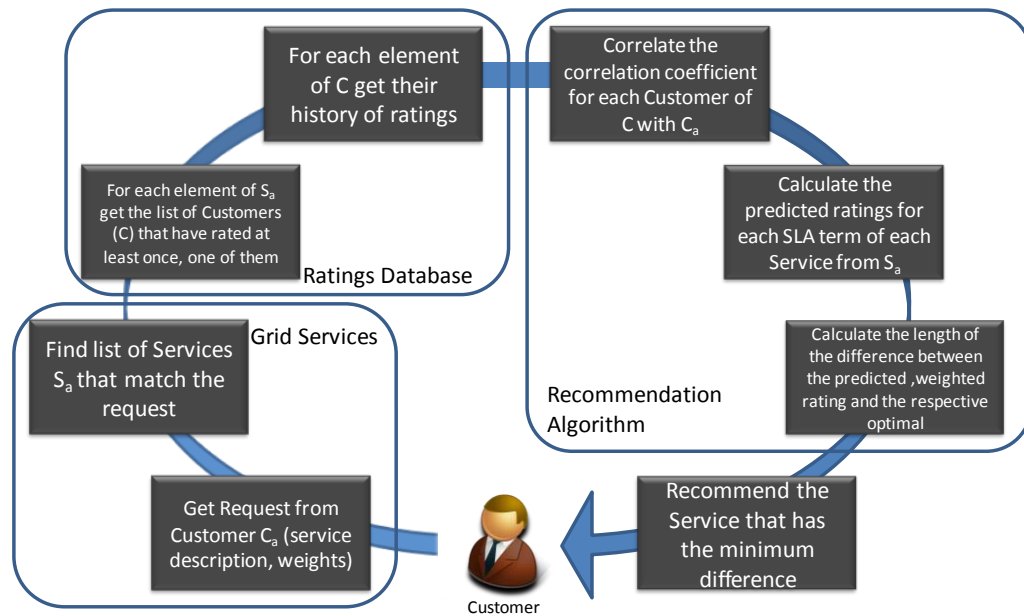
- `getRatings ()`: Μέθοδος που συγκεντρώνει όλες τις βαθμολογίες για τις κοινόχρηστες υπηρεσίες των πελατών, που αποθηκεύονται στη βάση δεδομένων του συστήματος συστάσεων.
- `getPredictions ()`: Μέθοδος που κάνει προβλέψεις για τις βαθμολογίες ενός πελάτη σε υπηρεσίες που δεν έχει καλέσει στο παρελθόν, βάσει των τεχνικών συσχέτισης με βαθμολογίες άλλων πελατών που χρησιμοποιεί ο SR μηχανισμός.

Μετά την ανάκτηση των αποτελεσμάτων πρόβλεψης για κάθε διαθέσιμη υπηρεσία, μια Υπηρεσία Συστάσεων συγκρίνει και κάνει μια σύσταση στον πελάτη:



Σχήμα 25: Η αλληλεπίδραση των οντοτήτων του συστήματος για την αξιολόγηση και παροχή συστάσεων

Η όλη υλοποίηση έγινε στα πλαίσιο του παρακάτω μοντέλου σύστασης, το οποίο συνοψίζει τα βήματα που περιγράφηκαν από την οπτική του συστήματος συστάσεων:



Σχήμα 26: Διακριτά βήματα του Μοντέλου της Υπηρεσίας Συστάσεων

Όπως φαίνεται στο Σχήμα 26, η όλη διαδικασία ξεκινά όταν ένας πελάτης (έστω C_a) ζητά ένα συγκεκριμένο τύπο διαδικτυακής υπηρεσίας. Η αίτηση αυτή συνοδεύεται από την περιγραφή της υπηρεσίας (π.χ. υπηρεσία ανάλυσης συναισθημάτων σε ανεπίσημο κείμενο) και τις ειδικές απαιτήσεις που εκφράζονται μέσα από τα βάρη που δίνονται για τους όρους του SLA (π.χ. απαιτήσεις για ελάχιστη ποσοστιαία ορθότητα ανάλυσης και ελάχιστες απαιτήσεις για το χρόνο απόκρισης). Αυτό γίνεται μέσω των κατάλληλων διεπαφών του GRIA για την επιλογή υπηρεσιών. Το επόμενο βήμα στον κύκλο ζωής υπηρεσιών του GRIA είναι η εξεύρεση υπηρεσίας, το οποίο περιλαμβάνει την ανακάλυψη αυτών των τύπων υπηρεσιών που ταιριάζουν με την αρχική αίτηση. Όλα αυτά λαμβάνουν χώρα μέσα από μια σειρά αλληλεπιδράσεων μεταξύ του λογισμικού πελάτη και του ενδιάμεσου broker.

Με το πέρας των παραπάνω διαδικασιών που λαμβάνουν χώρα στην υποδομή του GRIA, καλείται το σύστημα συστάσεων, ως μέρος του broker κοινής εμπιστοσύνης (TTP - Σχήμα 20). Η TTP οντότητα, τηρεί ένα αρχείο όλων των προηγούμενων αξιολογήσεων των χρηστών. Χρησιμοποιώντας αυτό το αρχείο, η Υπηρεσία Συστάσεων επιλέγει τα διανύσματα αξιολόγησης μόνο των πελατών που έχουν αξιολογήσει έστω μία από τις διαθέσιμες υπηρεσίες, τουλάχιστον μία φορά. Στη συνέχεια, ο μηχανισμός συστάσεων που ανήκει στην TTP οντότητα χρησιμοποιεί την παρεχόμενη λίστα βαθμολογιών προκειμένου να βρει το συντελεστή συσχέτισης των αξιολογήσεων του C_a με τους άλλους πελάτες (που οι αξιολογήσεις τους περιλαμβάνονται στη λίστα). Αυτός ο συντελεστής είναι ιδιαίτερα χρήσιμος, προκειμένου να ανατεθούν βάρη στις γνώμες των άλλων πελατών για το ζητούμενο τύπο υπηρεσίας και να παρασχεθεί μια πρόβλεψη για κάθε διαθέσιμη υπηρεσία αυτού του τύπου. Αυτή η πρόβλεψη παράγεται ως ένα διάνυσμα βαθμολογιών για τις παραμέτρους κάθε υπηρεσίας. Το διάνυσμα με το μικρότερο μέτρο διαφοράς από την ιδανική βαθμολογία του πελάτη C_a αναδεικνύει την υπηρεσία που θα ικανοποιεί τις απαιτήσεις του με τον καλύτερο τρόπο. Το αποτέλεσμα αυτό παρουσιάζεται απευθείας στον πελάτη του GRIA.

5.4.3 Εκτεταμένη αξιολόγηση με προσομοιωμένο σύνολο δεδομένων

5.4.3.1 Σύνολο πειραματικών δεδομένων προσομοίωσης

Τα δεδομένα που συγκεντρώθηκαν βασίστηκαν όπως προαναφέρθηκε σε ένα πραγματικό επιχειρηματικό σενάριο, το οποίο αφορά την εφαρμογή 3D Rendering εικόνας και βίντεο. Η σχέση ανάμεσα στους πελάτες (3D animators) και τους παρόχους υπηρεσιών διέπεται από SLAs, οι όροι των οποίων απεικονίζονται στον Πίνακα 9 για το πλαίσιο αυτών των πειραμάτων. Αυτό που ενδιαφέρει στην προκειμένη περίπτωση είναι οι "QoS

Όροι", που αντιπροσωπεύουν τους όρους σε επίπεδο εφαρμογής, μέσω των οποίων ο 3D animator εκφράζει τις απαιτήσεις της κάθε εργασίας. Εν προκειμένω, αποφύγαμε τη χρήση όρων SLA σε επίπεδο πόρων και επικεντρωθήκαμε σε όρους σχετικούς με την εφαρμογή με σκοπό την απλοποίηση της αξιολόγησης του QoS και τη διευκόλυνση των animators. Για παράδειγμα, δεν χρησιμοποιήθηκε ο αριθμός των CPUs (όπως στον Πίνακα 5) ή η ποσότητα μνήμης που ο animator επιτρέπεται να χρησιμοποιεί, επιτρέποντας το προεπιλεγμένο διαχειριστή πόρων του GRIA να τα καταναίμει.

Γενικοί Όροι	QoS Όροι	Όροι Τιμολόγησης
Parties	Renderer	Normal Price
Start Time	Resolution	Compensations for all the QoS Terms
End Time	Frame Rate	
	Codec	

Πίνακας 9: Οι όροι του SLA για την υπηρεσία που επιτελεί 3D video rendering

Για τους σκοπούς των πειραμάτων αυτών, δημιουργήσαμε πέντε διαφορετικές υπηρεσίες 3D Rendering. Η διαφοροποίηση μεταξύ τους είχε κυρίως να κάνει με το λογισμικό (AIR®, 3Delight, AC3D renderers) και την τεχνική (χρήση της προδιαγραφής RenderMan Interface Bytestream (RIB)) που χρησιμοποιήθηκε για το 3D Rendering, καθώς και την υπολογιστική υποδομή που παραχωρήθηκε στην κάθε υπηρεσία. Επτά διαφορετικοί 3D animators χρησιμοποίησαν τις πέντε υπηρεσίες 3D Rendering και αξιολόγησαν την κάθε μία, βαθμολογώντας κάθε όρο του SLA με βάση την Ποιότητα Υπηρεσίας που έλαβαν. Η βαθμολογίες είχαν εύρος 1-5, με το 1 να αντιπροσωπεύει την κακή και το 5 την καλύτερη δυνατή ποιότητα. Οι τελικοί χρήστες επανέλαβαν την κλήση των υπηρεσιών αρκετές φορές (από 20 έως και 30) και ο μέσοι όροι όλων των αξιολογήσεων υπολογίστηκαν, όπως παρουσιάζονται στον Πίνακα 10.

Παράμετροι	Renderer	Resolution	Frame Rate	Codec
Υπηρεσία 1	4.3	3.9	3.7	4.1
Υπηρεσία 2	4.5	4.1	3.9	4.3
Υπηρεσία 3	4.4	4.2	3.9	4.3
Υπηρεσία 4	4.4	4.1	4.0	4.3
Υπηρεσία 5	4.4	4.1	4.3	4.4

Πίνακας 10: Μέση αξιολόγηση της ποιότητας 5 υπηρεσιών από 7 animators με βάση 4 παραμέτρους ποιότητας

Ωστόσο, για την πλήρη αξιολόγηση του προτεινόμενου μοντέλου, ένας μεγάλος αριθμός βαθμολογιών πελατών θα ήταν χρήσιμος, αλλά μια τέτοια εκτεταμένη χρήση των υπηρεσιών δεν ήταν ιδιαίτερα εύκολο να επιτευχθεί στα πλαίσια των πειραμάτων μας. Δεδομένου ότι η εφαρμογή στήθηκε για ερευνητικούς σκοπούς κι όχι για εμπορική χρήση, περιορίζεται σημαντικά η δυνατότητα ενασχόλησης περισσότερων τελικών χρηστών. Για αυτό το λόγο αποφασίστηκε να γίνει μια προσομοίωση ενός μεγάλου αριθμού αξιολογήσεων πελατών, παίρνοντας σαν βάση το υπάρχον σύνολο δεδομένων.

5.4.3.1.1 Η κατανομή Weibull

Το μοντέλο που χρησιμοποιήθηκε για τη δημιουργία ενός τεχνητού συνόλου δεδομένων βασίζεται στην κατανομή Weibull (ReliaSoft, 2006). Η κατανομή Weibull προσφέρει μια συνεχή συνάρτηση πυκνότητας πιθανότητας που χρησιμοποιείται συνήθως στις ακόλουθες περιπτώσεις:

- Για την αναπαράσταση της διαθεσιμότητας υπολογιστών σε υπολογιστικά περιβάλλοντα μεγάλης κλίμακας, όπως μεγάλες επιχειρήσεις, το διαδίκτυο και τα υπολογιστικά πλέγματα.
- Για την αναπαράσταση της κατανομής των χρόνων παρασκευής και απόκρισης στη μηχανική βιομηχανίας.
- Στην ανάλυση μηχανικής αξιοπιστίας και αστοχιών (η πιο κοινή χρήση).

Τα παραπάνω κριτήρια μπορούν να αντιστοιχιστούν σε κάποιες συνήθεις παραμέτρους που χρησιμοποιούνται στα SLAs υπηρεσιών που παρέχονται από ένα SaaS περιβάλλον: τη διαθεσιμότητα, το χρόνο απόκρισης και την αξιοπιστία των αποτελεσμάτων. Δεδομένου λοιπόν ότι η κατανομή Weibull μπορεί να περιγράψει τη μακροπρόθεσμη συμπεριφορά ενός SaaS συστήματος στο χρόνο, με βάση τις κλήσεις των υπηρεσιών του από πολλούς χρήστες, θεωρούμε ότι είναι ικανή να αναπαραστήσει ικανοποιητικά και την κατανομή των αξιολογήσεων των πελατών για τους συνήθεις QoS όρους, όπως για παράδειγμα η αξιοπιστία/διαθεσιμότητα των υπηρεσιών, ο χρόνος απόκρισης, η ακρίβεια των αποτελεσμάτων κλπ.

Η συνάρτηση πυκνότητας πιθανότητας της κατανομής Weibull είναι η ακόλουθη:

$$f(x) = \frac{\alpha}{\beta} \left(\frac{x-\gamma}{\beta}\right)^{\alpha-1} e^{-\left(\frac{x-\gamma}{\beta}\right)^\alpha}$$

Εξίσωση 4: Συνάρτηση πυκνότητας πιθανότητας Weibull

όπου στην περίπτωσή μας το ολοκλήρωμα $\int_{x1}^{x2} f(x)dx$ αναπαριστά την πιθανότητα η μέση βαθμολογία ενός πελάτη για μια παράμετρο μιας συγκεκριμένης υπηρεσίας να βρίσκεται μεταξύ των τιμών $x1$ και $x2$.

Οι παράμετροι της κατανομής Weibull όπως φαίνονται στην Εξίσωση 4 είναι οι εξής:

- α : παράμετρος σχήματος ($a > 0$)
- β : παράμετρος κλιμάκωσης ($b > 0$)
- γ : παράμετρος τοποθέτησης (με $\gamma = 0$ παίρνουμε την κατανομή Weibull 2 παραμέτρων)

Η παράμετρος α ορίζει το σχήμα της κατανομής, καθορίζοντας πόσο οξεία είναι η καμπύλη κατανομής πάνω από τη μέση τιμή, γύρω από την οποία βρίσκονται οι περισσότερες βαθμολογίες, και τη διακύμανσή αυτών. Επομένως, αυξάνοντας το α , η διακύμανση της κατανομής μειώνεται και οι βαθμολογίες συγκεντρώνονται γύρω από τη μέση τιμή. Η παράμετρος β ορίζει την κλιμάκωση της καμπύλης, καθορίζοντας τη μέση τιμή των βαθμολογιών που αναφέρθηκε, ενώ η παράμετρος γ καθορίζει την θέση της κατανομής στον άξονα των X . Εφόσον στα πειράματά μας θέλουμε την κατανομή να ξεκινά από την αρχή των αξόνων, στα θεωρούμε ότι $\gamma = 0$, έχοντας την κατανομή Weibull 2 παραμέτρων. Θέτοντας διαφορετικές τιμές στις 2 παραμέτρους μιας Weibull, μπορούν να παραχθούν διαφορετικές κατανομές κάθε φορά, με συγκεκριμένο μέσο όρο και διακύμανση.

5.4.3.1.2 Παραγωγή προσομοιωμένου συνόλου δεδομένων

Στο πλαίσιο της πρώτης δέσμης πειραμάτων, αποφασίστηκε να δημιουργήσουμε ένα μεγάλο προσομοιωμένο σύνολο δεδομένων που αφορά αξιολογήσεις πελατών (συνολικού εύρους 1-5) με κατανομές Weibull, οι οποίες θα έχουν ίδιο μέσο όρο (m) και διακύμανση ($\text{Var}(X) = E[(X-m)^2]$) με τις πραγματικές κατανομές βαθμολογιών που προέκυψαν παραπάνω. Αυτό χρειάζεται για να είναι αυτή η προσομοίωση έγκυρη, ώστε να διατηρηθεί η υποκειμενικότητα των αξιολογήσεων των 3D animators, κάτι που εξασφαλίζεται παράγοντας τεχνητά σύνολα δεδομένων με βάση τις πραγματικές βαθμολογίες.

Ωστόσο, δεν έχει βρεθεί καμία αποτελεσματική μαθηματική λύση για το πρόβλημα του υπολογισμού των παραμέτρων α και β της κατανομής Weibull, με δεδομένη τη μέση τιμή και τη διακύμανση. Συνεπώς, η μέθοδος που ακολουθήθηκε για

τη δημιουργία των προαναφερθέντων κατανομών για κάθε όρο του SLA, ήταν ο πειραματισμός με διάφορες τιμές για τις παραμέτρους a και b , έως ότου προσεγγιστούν οι επιθυμητές τιμές μέσου όρου και διακύμανσης. Έτσι, με χρήση των πραγματικών δεδομένων από τις αξιολογήσεις των υπηρεσιών 3D Rendering (Πίνακας 10) καθορίστηκαν οι τιμές των παραμέτρων της κατανομής όπως περιγράφεται παραπάνω, για να παραχθεί μια ρεαλιστική προσομοίωση για την αξιολόγηση της μεθόδου.

Εδώ πρέπει να αναφερθεί πάλι ότι σε πραγματικά σύνολα δεδομένων είναι δυνατόν να υπάρξουν ψεύτικες ή και επιτηδευμένες αξιολογήσεις/βαθμολογίες με σκοπό να επηρεάσουν (θετικά συνήθως) τη φήμη ενός παρόχου υπηρεσιών και συνολικά τα αποτελέσματα της ανάλυσης. Σε μια πραγματική αγορά υπηρεσιών ή και προϊόντων αυτό είναι σύνηθες φαινόμενο, με πιο προφανές παράδειγμα το παρεμφερές σύστημα αξιολόγησης του e-Bay, όπου οι περισσότεροι πωλητές έχουν επιτηδευμένες βαθμολογίες που προσεγγίζουν τις ιδανικές. Οι κακόβουλες αξιολογήσεις μπορούν να συνεισφέρουν σημαντικά στη συνολική φήμη μιας υπηρεσίας, όταν χρησιμοποιείται μόνο η μέση αξιολόγηση, όπως συμβαίνει και στο προαναφερθέν σύστημα. Ωστόσο, με χρήση της προτεινόμενης συνεργατικής μεθόδου, μακροπρόθεσμα οι αξιολογήσεις αυτές θα έχουν μεγάλη απόκλιση από τις περισσότερες βαθμολογίες και μηδαμινή γραμμικότητα με τα διανύσματα αξιολογήσεων άλλων πελατών, συνεπώς η συνεισφορά τους στην πρόβλεψη των αξιολογήσεων για κάθε παράμετρο αναμένεται να είναι μηδαμινή (οι συντελεστές συσχέτισης θα προσεγγίζουν το μηδέν).

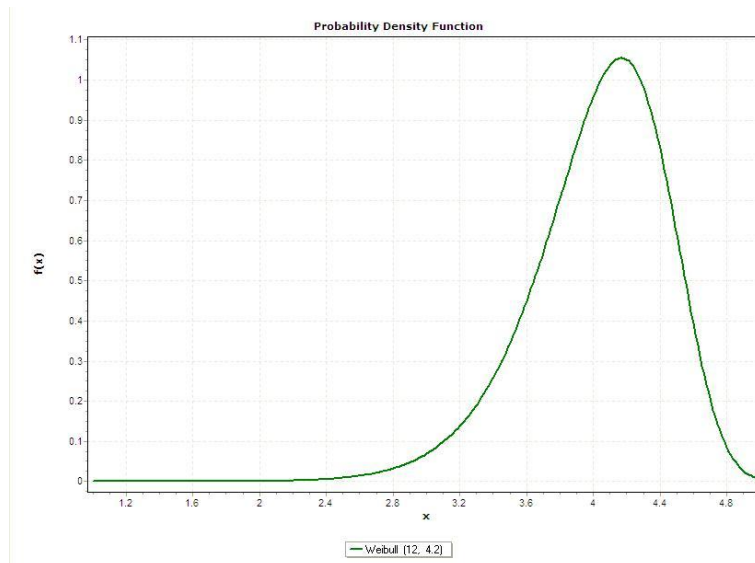
Όπως αναφέρθηκε έχουμε 5 διαφορετικούς παρόχους που παρέχουν τον ίδιο τύπο 3D Rendering υπηρεσίας. Ο σκοπός των παρακάτω πειραμάτων είναι να προβλεφθούν οι αξιολογήσεις των πελατών για τέσσερις (4) παραμέτρους ποιότητας: p_1 , p_2 , p_3 και p_4 .

Για κάθε παράμετρο του SLA λοιπόν, χρησιμοποιώντας τις μέσες τιμές και διακυμάνσεις των πραγματικών αξιολογήσεων του Πίνακα 10, παρήχθησαν πέντε (5) κατανομές Weibull που αντιπροσωπεύουν τις κατανομές της βαθμολογίας χρηστών για τους πέντε διαφορετικούς παρόχους υπηρεσιών (20 κατανομές συνολικά). Ωστόσο, δεδομένου ότι κάποιες από τις κατανομές έχουν σχεδόν ακριβώς τον ίδιο μέσο όρο και διακύμανση, στην πραγματικότητα προκύπτουν δέκα (10) διαφορετικές κατανομές Weibull, των οποίων οι παράμετροι φαίνονται παρακάτω:

Κατανομή #	α	β	Μέση τιμή
1	12	4.45	4.3
2	15.5	4.6	4.5
3	16	4.4	4.4
4	16	4.5	4.4
5	10	4.1	3.9
6	11	4.3	4.1
7	11.5	4.4	4.2
8	9.5	3.9	3.7
9	10	4.1	3.9
10	12	4.2	4

Πίνακας 11: Τα στοιχεία των δέκα κατανομών Weibull που χρησιμοποιούνται στην προσομοίωση

Για να γίνει περισσότερο κατανοητή στον αναγνώστη η μορφή αυτών των κατανομών, παρουσιάζεται η γραφική παράσταση της τελευταίας του πίνακα:



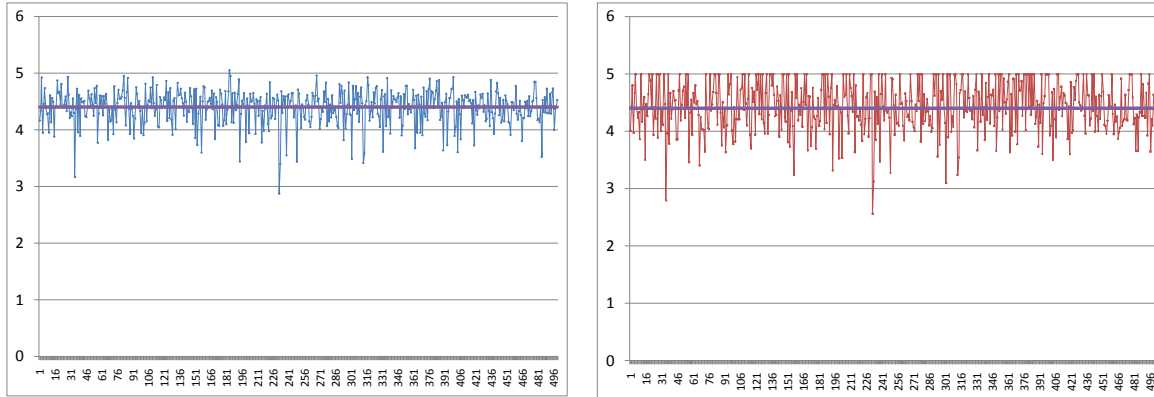
Σχήμα 27: Κατανομή Weibull με $\alpha=12$, $\beta=4,2$, μέση τιμή=4

Οι παραχθείσες βαθμολογίες για κάθε παράμετρο ποιότητας, ακολουθούν επίσης κατανομή Weibull και ανάμεσα στις διάφορες υπηρεσίες. Με χρήση αυτών των κατανομών το πρόβλημα μπορεί να αναλυθεί σε μια πιο ρεαλιστική κλίμακα. Τα αποτελέσματα αυτής της ανάλυσης παρουσιάζονται στην παράγραφο που ακολουθεί.

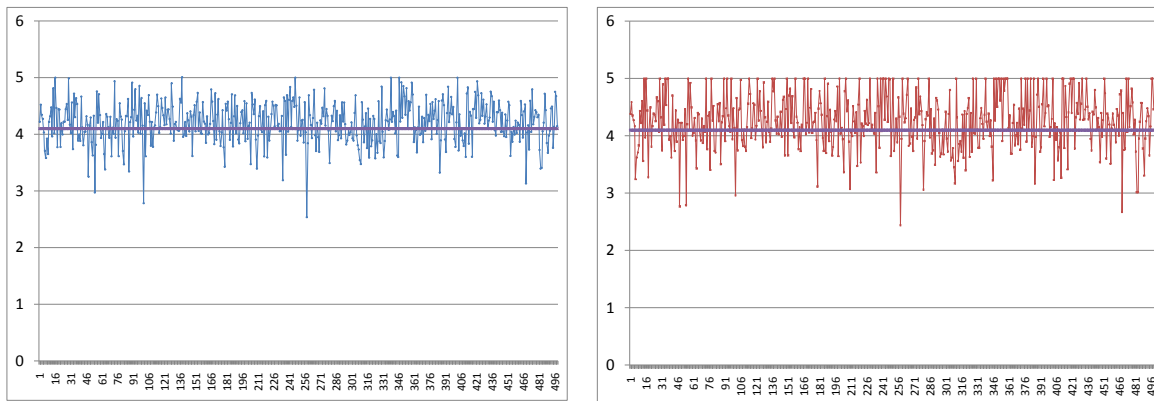
5.4.3.2 Αξιολόγηση αποτελεσμάτων

Με βάση τις κατανομές που προαναφέρθηκαν, δημιουργήθηκαν τεχνητές βαθμολογίες 500 διαφορετικών πελατών, ως αξιολόγηση των 4 παραμέτρων ποιότητας για τις 5 διαφορετικές υπηρεσίες. Οι προβλέψεις της αξιολόγησης με χρήση της συνεργατικής μεθόδου έγιναν για την πέμπτη κατά σειρά υπηρεσία, για κάθε έναν από τους 500 πελάτες και τις 4 παραμέτρους ποιότητας. Ο υπολογισμός της πρόβλεψης για κάθε πελάτη υλοποιήθηκε με χρήση του αλγορίθμου συσχέτισης του Pearson, με όλες τις άλλες αξιολογήσεις, έτσι ώστε να συσχετιστεί κάθε πελάτης με τους υπόλοιπους 499, με βάση τις προηγούμενες αξιολογήσεις του για τις υπόλοιπες 4 υπηρεσίες.

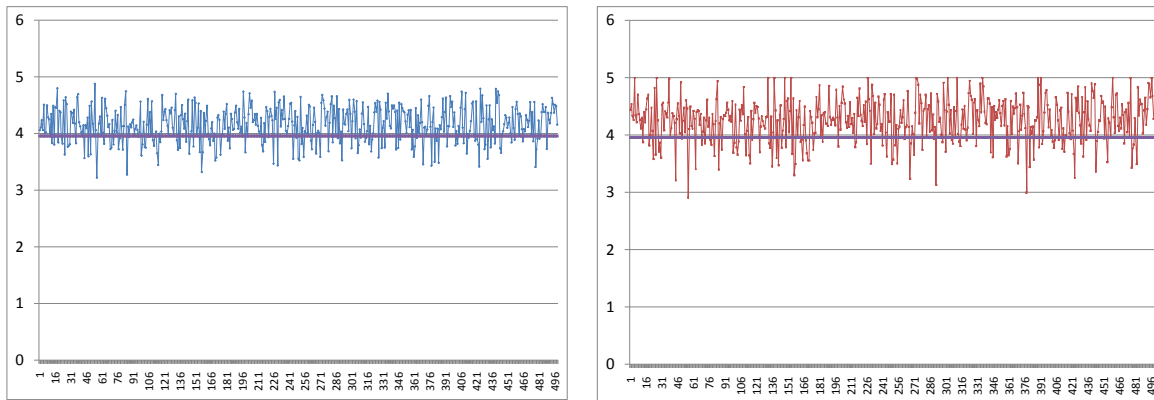
Τα αποτελέσματα που παρουσιάζονται στα παρακάτω σχήματα συγκρίνουν τις τιμές πρόβλεψης με τις πραγματικές τιμές καθώς και τη μέση τιμή όλων των πελατών, η οποία είναι αυτή που χρησιμοποιείται στα περισσότερα υπάρχοντα μοντέλα αξιολόγησης. Οι μπλε γραμμές παρακάτω απεικονίζουν τις ρεαλιστικές τιμές που παρήχθησαν με την κατανομή Weibull, οι κόκκινες γραμμές τις βαθμολογίες που προβλέφθηκαν με την παρουσιαζόμενη μέθοδο και οι μωβ το μέσο όρο των αξιολογήσεων για κάθε μεταβλητή:



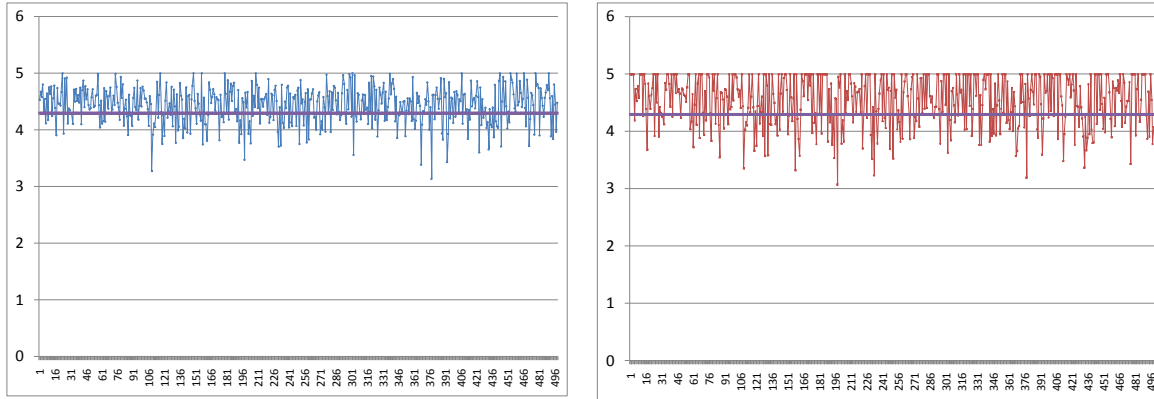
Σχήμα 28: Πραγματικές τιμές και μέση τιμή για την p1 (αριστερά)/ Τιμές πρόβλεψης και μέση τιμή για την p1 (δεξιά)



Σχήμα 29: Πραγματικές τιμές και μέση τιμή για την p2 (αριστερά)/ Τιμές πρόβλεψης και μέση τιμή για την p2 (δεξιά)



Σχήμα 30: Πραγματικές τιμές και μέση τιμή για την p3 (αριστερά)/ Τιμές πρόβλεψης και μέση τιμή για την p3 (δεξιά)



Σχήμα 31: Πραγματικές τιμές και μέση τιμή για την p4 (αριστερά)/ Τιμές πρόβλεψης και μέση τιμή για την p4 (δεξιά)

Όπως εκ πρώτης όψεως φαίνεται στα παραπάνω σχήματα, στις περισσότερες περιπτώσεις οι προβλέψεις προσεγγίζουν ικανοποιητικά τις πραγματικές αξιολογήσεις, δείχνοντας πως η μέθοδος λειτουργεί σωστά, ωστόσο σποραδικά μπορούν να εντοπιστούν κάποιες αποκλίσεις από τις αναμενόμενες τιμές.

Τα μέσα ποσοστιαία σφάλματα (Mean Percent Error - MPE) των παραπάνω προβλέψεων αλλά και της μέσης τιμής για κάθε παράμετρο φαίνονται στον Πίνακα 12:

Παράμετρος	p1	p2	p3	p4
<i>MPE collaborative filtering μεθόδου</i>	4,4%	4,1%	4,3%	4,5%
<i>MPE μέσης τιμής</i>	5,5%	7,3%	7,3%	6,7%

Πίνακας 12: Μέσα ποσοστιαία σφάλματα πρόβλεψης

Όπως βλέπουμε, το προτεινόμενο μοντέλο πρόβλεψης αξιολόγησης με βάση τη συσχέτιση φαίνεται να είναι σαφώς αποτελεσματικότερο από την εναλλακτική λύση της μέσης τιμής. Οι υπάρχουσες αποκλίσεις μπορεί να δικαιολογηθούν και από το γεγονός ότι το σύνολο δεδομένων που χρησιμοποιήθηκε είναι τυχαίοι αριθμοί που δημιουργούνται με το εργαλείο StatAssist, έτσι ώστε να ακολουθούν την κατανομή Weibull, αλλά προφανώς κάποιες φορές μπορεί να αποκλίνουν από την επιλεγμένη κατανομή λόγω της τυχαιότητάς τους.

Με την αύξηση του αριθμού των παρόχων υπηρεσιών, η μέση βαθμολογία των χρηστών προσεγγίζει σαφώς περισσότερο τη μέση τιμή της κατανομής, έτσι ώστε τα δεδομένα να γίνονται πιο ρεαλιστικά. Επιπλέον, με την αύξηση των πελατών του δείγματος που χρησιμοποιείται σε αυτά τα πειράματα, οι προβλέψεις είναι πιο ακριβείς, και η συμβολή πιθανών επιτηδευμένων βαθμολογιών μειώνεται, αφού εμφανίζονται "ομάδες χρηστών", σύμφωνα με τις προτιμήσεις των πελατών, οι οποίες είναι ιδιαίτερα χρήσιμες για την παρούσα ανάλυση. Με την ομαδοποίηση των πελατών ανάλογα με τις προτιμήσεις τους και τον ανάλογο χειρισμό αυτών των ομάδων, καταφέρνουμε να προβλέψουμε τις αξιολογήσεις τους με πιο αποτελεσματικό και ακριβή τρόπο, καθώς οι χρήστες ιδίων ομάδων αναμένεται να βαθμολογήσουν με παρόμοιο τρόπο και να μην επηρεαστούν από κακόβουλες αξιολογήσεις ή δεδομένα έξω από την ομάδα τους.

5.4.4 Πρακτική Επαλήθευση

Σε επέκταση των προηγούμενων πειραμάτων, και ως πρακτική επαλήθευση, εκτελέσαμε κάποια πειράματα με τους πραγματικούς τελικούς χρήστες για να επιβεβαιώσουμε τη σωστή υποστήριξη αποφάσεων στο χρήστη από τον SR μηχανισμό. Η πειραματική διάταξη περιελάμβανε τις 5 υπηρεσίες 3D Rendering που στήθηκαν παραπάνω στο μεσολογισμικό GRIA, που ενεργούν ως διακριτοί πάροχοι υπηρεσιών. Τα πρότυπα SLAs που δημιουργήθηκαν για τους σκοπούς των πειραμάτων (ένα απλό παράδειγμα SLA Template του GRIA για 3D Rendering φαίνεται στον Πίνακα 5) περιέλαβαν εδώ όλες τις λεπτομέρειες για τους παρόχους υπηρεσιών και την εφαρμογή του αλγόριθμου 3D απόδοσης. Επίσης, η διάρκεια ζωής τους είχε οριστεί σε μία μόνο εκτέλεση της εργασίας (σε αντίθεση με καθορισμό της χρονικής περιόδου στο προηγούμενο παράδειγμα), χωρίς περιορισμούς από την πλευρά των καταναλωτών, έτσι ώστε οι animators να φέρουν εις

πέρασ τις εργασίες τους χωρίς να ανησυχούν για παραβιάσεις SLA. Στο GRIA καθορίστηκε να επιτρέπεται στην εκτέλεση να συνεχίζει, ακόμη και σε περίπτωση παραβίασης SLA, εφόσον ο καταναλωτής θα είναι αυτός που αξιολογεί κι όχι ο πάροχος ή ο service broker.

Όσον αφορά τους όρους που αντιπροσωπεύουν την ποιότητα στα SLAs, αποφύγαμε όπως και στα προηγούμενα πειράματα χρήση των πόρων χαμηλού επιπέδου. Αντ' αυτού, έμπειροι animators - χρήστες των υπηρεσιών του SocIoS (βλ. Ενότητα 3.4.2) προσδιόρισαν μια σειρά από μετρικές ποιότητας που υιοθετήθηκαν ως οι QoS παράμετροι που θα αξιολογηθούν: χρόνος απόκρισης, διαθεσιμότητα, σφάλματα πλαισίων και αναλογία τιμής / ποιότητας. Ένας δεύτερος λόγος για την επιλογή των όρων αυτών είναι ότι μπορεί να μετρηθούν και από τις δύο πλευρές, εφόσον μπορούν να ελεγχθούν και από την πλευρά του παρόχου με τους κατάλληλους μηχανισμούς. Οι ευρέως χρησιμοποιούμενες αυτές παράμετροι ποιότητας εξηγούνται ως εξής:

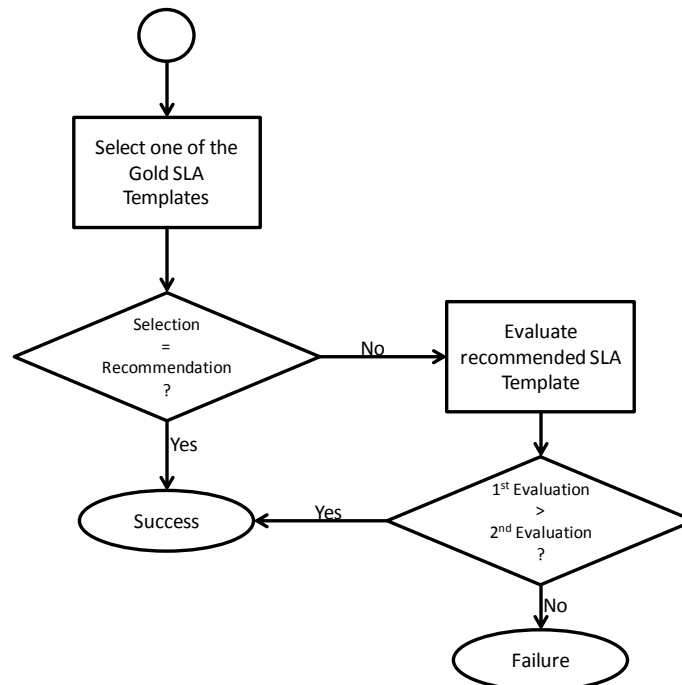
- **Χρόνος απόκρισης:** Ο χρόνος που χρειάζεται στην υπηρεσία για να αποδώσει ένα βίντεο και να στείλει το αποτέλεσμα στον πελάτη.
- **Διαθεσιμότητα:** Η αναλογία των περιπτώσεων που η υπηρεσία ήταν διαθέσιμη όταν ζητήθηκε με αυτές που δεν αποκρινόταν.
- **Frame Rate:** Ανάλογα με την πολυπλοκότητα του μοντέλου 3D, κάποιοι renderers μπορεί να επιστρέψουν βίντεο με χαμένα/κατεστραμμένα πλαίσια ή λάθη στον ήχο
- **Αναλογία τιμής / ποιότητας:** Ένα μέτρο για να εκτιμηθεί αν η ποιότητα των υπηρεσιών άξιζε την τιμή για τον καταναλωτή

Βάσει αυτών, για κάθε έναν από τους πέντε παρόχους υπηρεσιών που ορίζονται στα τρέχοντα πειράματα, δημιουργήθηκαν τρία πρότυπα SLAs με διαφορετικές παραμέτρους, έτσι ώστε να μπορούν να κατηγοριοποιηθούν σε Gold, Silver και Bronze προσφορές. Ως εκ τούτου, $5 \times 3 = 15$ διαφορετικές προσφορές δημιουργήθηκαν. Κάθε ένας από τους επτά (7) animators κλήθηκε να χρησιμοποιήσει και να αξιολογήσει τις διαφορετικές προσφορές (στα πλαίσια αυτών των πειραμάτων αξιολογείται και συστήνεται κάθε διαφορετικό SLA Template, παρά κάθε υπηρεσία). Τους επετράπη να χρησιμοποιήσουν οποιαδήποτε δεδομένα εισόδου θέλουν, με τον περιορισμό να δώσουν τα ίδια δεδομένα για κάθε τύπο προτύπου SLA. Επιπλέον, ζητήθηκε η παραγωγή βίντεο να μην είναι μεγαλύτερη από 3 λεπτά. Αυτός ο περιορισμός επιβλήθηκε λόγω της ανάγκης να εξασφαλιστεί ότι η υποδομή δεν θα επιβαρύνεται τόσο από τις κλήσεις ώστε να δημιουργήσει προβλήματα έξω από τον έλεγχο των παρόχων. Μετά την κλήση κάθε υπηρεσίας, ένα ηλεκτρονικό έντυπο (built-in στον πελάτη GRIA) παρουσιάστηκε στους animators ζητώντας τους να βαθμολογήσουν κάθε μία από τις παραμέτρους QoS στο SLA σε μία κλίμακα από 1 έως 5 (όπου 1 = μη ικανοποιητική και 5 = απόλυτα ικανοποιητική). Η κλίμακα αυτή είναι ισοδύναμη με μια τυπική ψυχομετρική κλίμακα Likert πέντε επιπέδων. Με την ολοκλήρωση αυτής της διαδικασίας, ο SR μηχανισμός δημιούργησε μια δέσμη με αξιολογήσεις και συσχετίσεις στη βάση δεδομένων του.

Στη συνέχεια, ένας (1) από τους animators κλήθηκε να χρησιμοποιήσει και να αξιολογήσει υπό τους ίδιους όρους όπως οι προηγούμενοι, μόνο 10 από τις διαθέσιμες προσφορές υπηρεσιών (μόνο Bronze και Silver προσφορές). Ακολούθως, κλήθηκε να επιλέξει ένα από τα υπόλοιπα 5 πρότυπα Gold SLA, με βάση την εμπειρία που συνέλλεξε από τη χρήση των ίδιων υπηρεσιών κάτω από διαφορετικά SLAs. Το νόημα πίσω από

αυτό το βήμα ήταν ο animator να σχηματίσει γνώμη για κάθε πάροχο και στη συνέχεια να χρησιμοποιήσει αυτή τη γνώμη για να τον αξιολογήσει σε μια πιο απαιτητική υπόθεση. Μετά την εκτέλεση της υπηρεσίας, ο animator κλήθηκε να αξιολογήσει το SLA. Στη συνέχεια, η επιλογή συγκρίθηκε με τη σύσταση που το SR παράγει χρησιμοποιώντας τους συντελεστές συσχέτισης Pearson. Εάν η επιλεγμένη υπηρεσία, δεν ταίριαζε με τη συνιστώμενη, ο animator κλήθηκε να καλέσει την τελευταία με την ίδια διάταξη (δεδομένα εισόδου και ρυθμίσεις), και στη συνέχεια να την αξιολογήσει.

Αυτό επαναλήφθηκε 7 φορές, κάθε φορά τοποθετώντας έναν άλλο από τους 7 animators στη θέση να επιλέξει μία από τις 5 προσφορές παροχής υπηρεσιών με βάση την εμπειρία του. Πριν από κάθε επανάληψη χρειάστηκε επαναρύθμιση των υπηρεσιών και αξιολόγηση όλων των SLAs από τους υπόλοιπους έξι πελάτες, ώστε ο τρέχων πελάτης να είναι ο μόνος που δεν έχει αξιολογήσει τα Gold SLAs. Το παρακάτω σχήμα απεικονίζει γραφικά το πειραματικό σενάριο κατά την πρώτη φάση:



Σχήμα 32: Πειραματική διαδικασία (Φάση Α)

Τα αποτελέσματα έδειξαν ότι σε 4 από τις 7 περιπτώσεις, ο animator επέλεξε την ίδια υπηρεσία με αυτή που ο SR μηχανισμός είχε προτείνει, ενώ σε 2 από τις 3 υπόλοιπες περιπτώσεις, οι αξιολογήσεις της συνιστώμενης υπηρεσίας από τους animators ήταν πιο κοντά στην βέλτιστες (από το SR) από αυτές στην αρχική επιλογή τους. Αυτό το αποτέλεσμα, επιβεβαιώνει ότι η εμπειρία των άλλων είναι χρήσιμη στη μοντελοποίηση της συνολικής εμπειρία του καταναλωτή. Ωστόσο, αυτό δεν σημαίνει ότι η επιλογή των καταναλωτών είναι η βέλτιστη, ως εκ τούτου, πρέπει επιπλέον να συγκριθούν οι προβλέψεις με τις αξιολογήσεις σε περισσότερες περιπτώσεις για να επιβεβαιώσουμε την επιτυχή σύζευξη μεταξύ QoE και QoS.

Στον Πίνακα 13, παρουσιάζονται οι βαθμολογίες του Animator 1 για τα Silver και Bronze SLAs και οι αντίστοιχες προβλεπόμενες τιμές για τα Gold SLAs από τον SR μηχανισμό, όπου τα SLAs με τον ίδιο αριθμό συνδέονται με τον ίδιο πάροχο υπηρεσιών:

	Bronze SLA Templates					Silver SLA Templates					Gold SLA Templates				
	SLA T1	SLA T2	SLA T3	SLA T4	SLA T5	SLA T1	SLA T2	SLA T3	SLA T4	SLA T5	SLA T1	SLA T2	SLA T3	SLA T4	SLA T5
QoS1	3	3	2	2	4	4	5	4	3	3	3,09416	3,82675	2,99450	2,58325	3,60851
QoS2	4	3	3	3	4	4	5	3	4	3	3,60179	3,18210	2,76509	2,42643	3,77873
QoS3	4	1	4	2	5	5	4	3	2	3	2,59940	3,16046	3,75547	1,83593	3,60407
QoS4	4	4	4	2	4	5	5	5	2	3	4,83578	4,69683	4,52513	1,18022	3,15624

Πίνακας 13: Πραγματικές και προβλεπόμενες βαθμολογίες για τον Animator 1

Θα περίμενε κανείς ότι η προηγούμενη εμπειρία του animator με τον πάροχο θα πρέπει να απεικονίζει την αναμενόμενη συμπεριφορά του όταν επιλέγει μια νέα προσφορά υπηρεσίας. Πράγματι, αυτό συμβαίνει στις περισσότερες περιπτώσεις, με τις προβλεπόμενες βαθμολογίες να είναι κοντά στις προηγούμενες για τον ίδιο πάροχο. Μια ενδιαφέρουσα περίπτωση ωστόσο παρουσιάζεται στο κελί που επισημαίνεται παρακάτω,

στο οποίο ακόμη και αν ο καταναλωτής έχει βαθμολογήσει τον πάροχο ψηλά για αυτήν την QoS παράμετρο (4 και 5) η προβλεπόμενη τιμή είναι σημαντικά χαμηλή (2,599404).

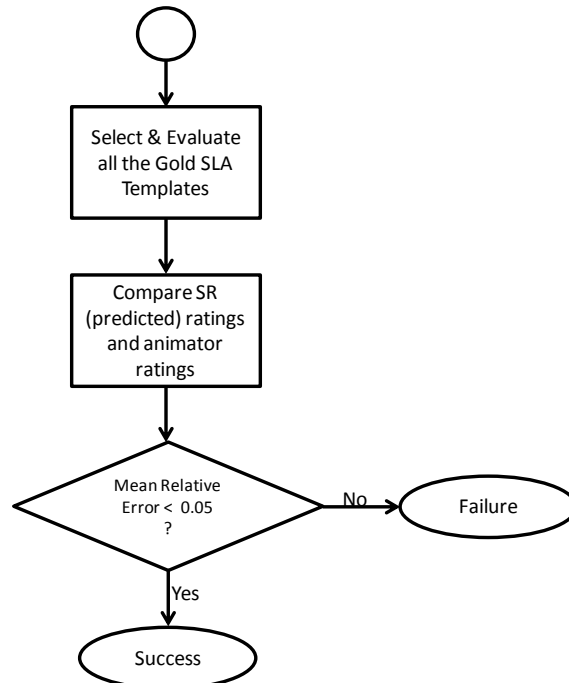
Αυτή η φαινομενική αποτυχία του συστήματος μπορεί να εξηγηθεί, αν παρατηρήσουμε τον Πίνακα 14, που παρουσιάζει τις συσχετίσεις μεταξύ του animator 1 και των υπολοίπων για κάθε μία από τις παραμέτρους QoS:

Παράμετρος	corr(1,2)	corr(1,3)	corr(1,4)	corr(1,5)	corr(1,6)	corr(1,7)
QoS1	0,707214	0,596371	0,342097	0,50703	0,652981	0,097988
QoS2	0,297398	0,210172	0,278793	0,874886	0,118371	-0,09058
QoS3	0,539732	0,218017	0,117871	0,30567	0,043466	0,801567
QoS4	0,118937	-0,380590	-0,308930	0,786246	-0,224880	0,299490

Πίνακας 14: Μήτρα συσχετίσεων για τον animator 1. Το corr(1, X) δείχνει τον συντελεστή συσχέτισης μεταξύ animator 1 και animator X. Οι τιμές κυμαίνονται από [-1,1], με το 0 να συνεπάγεται μηδενική συσχέτιση και τα 1 και -1 υψηλή συσχέτιση.

Για την παράμετρο QoS 3, στην οποία η αποτυχία εμφανίστηκε, ο καταναλωτής που επηρεάζει περισσότερο τον animator 1 είναι ο animator 7. Ο τελευταίος αξιολόγησε τη συγκεκριμένη υπηρεσία (SLA1) για το Gold SLA και τη συγκεκριμένη παράμετρο QoS (QoS3) αρκετά χαμηλά, με βαθμολογία 2. Ως εκ τούτου, η γνώμη του σταθμίζεται περισσότερο από τους υπόλοιπους για την εκτίμηση της αξιολόγησης του animator 1, οδηγώντας σε μια χαμηλή προβλεπόμενη τιμή. Στην πραγματικότητα, η αξιολόγηση που έδωσε ο animator 1 όταν εκτέλεσε τη συγκεκριμένη υπηρεσία με το Gold SLA ήταν 3, τιμή πιο κοντά στην πρόβλεψη παρά στη μέση δική του βαθμολογία.

Μετά την ανάλυση αυτή, προχωρήσαμε με την εξέταση του κατά πόσον η συνιστώμενη επιλογή ήταν στην πραγματικότητα η βέλτιστη (Φάση B, Σχήμα 33):

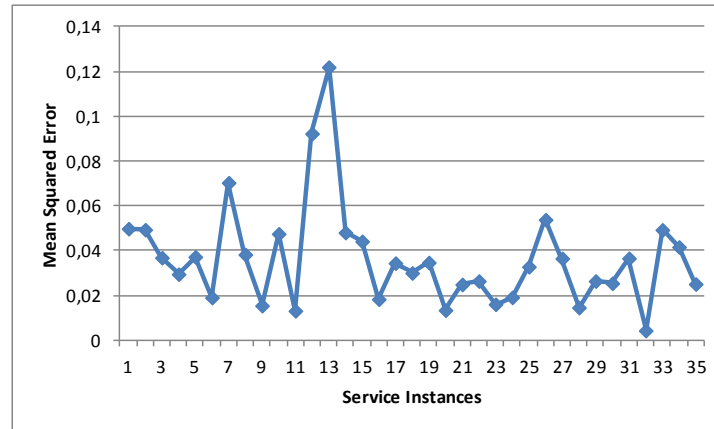


Σχήμα 33: Πειραματική διαδικασία (Φάση Β)

Στη φάση αυτή, ο animator ενήργησε ως επιλογέας κάθε φορά, καλούμενος επίσης να εκτελέσει τις υπόλοιπες προσφορές υπηρεσιών (3 ή 4 ανάλογα με το αν αρχική επιλογή του ταίριαζε με τη συνιστώμενη) υπό τους ίδιους όρους. Συνεπώς, 35 διανύσματα αξιολογήσεων παρήχθησαν (για 5 Gold SLA Templates από 7 animators) ώστε να συγκριθούν με τα 35 διανύσματα πρόβλεψης από τον SR μηχανισμό.

Τα αποτελέσματα της σύγκρισης έδειξαν ότι μόνο σε 4 περιπτώσεις (11,4% των συνολικών) από τις 35 η τιμή του μέσου ποσοστιαίου σφάλματος υπερέβη το όριο του 0,05 (Σχήμα 34). Στις περιπτώσεις που το σφάλμα αυτό είναι μικρότερο του 5%, ουσιαστικά η πρόβλεψη είναι απολύτως σωστή, αφού με στρογγυλοποίηση των τιμών (εφόσον οι αξιολογήσεις του πελάτη είναι ακέραιες) οι προβλέψεις ταυτίζονται με τις πραγματικές βαθμολογίες. Εξετάζοντας τα αποτελέσματα του πειράματος από την άποψη της αρχιτεκτονικής, είναι σημαντικό να αναφέρουμε ότι ο SR μηχανισμός έχει προσθέσει μια επιβάρυνση στον χρόνο απόκρισης ενός μητρώου UDDI. Η πολυπλοκότητα του

αλγορίθμου πρόβλεψης θα μπορούσαν να είναι αποτρεπτικός παράγοντας για την εφαρμογή αυτού του μηχανισμού, ωστόσο οι υπολογισμοί (και κυρίως οι λειτουργίες συσχέτισης) μπορούν να πραγματοποιούνται σε προηγούμενο χρόνο της διαδικασίας σύστασης και επιλογής υπηρεσίας, βελτιώνοντας σε μεγάλο βαθμό την απόδοση του συστήματος.



Σχήμα 34: Μέσο ποσοστιαίο σφάλμα στις 35 περιπτώσεις σύγκρισης αξιολογήσεων

5.5

Συμπεράσματα

Στα πλαίσια της παρουσιαζόμενης μεθόδου, καταδείχθηκε η δυνατότητα παροχής μηχανισμών υποστήριξης λήψης αποφάσεων σε χρήστες περιβαλλόντων SaaS με χρήση της εμπειρίας άλλων πελατών που έχουν χρησιμοποιήσει τις διαθέσιμες υπηρεσίες. Αυτό επιτυγχάνεται με την υλοποίηση ενός μηχανισμού συνεργατικού φιλτραρίσματος και την παροχή του ως υπηρεσία, ως μέρος της διαχείρισης του κύκλου ζωής των υπηρεσιών σε ένα υπάρχον υπηρεσιοστρεφές περιβάλλον. Η εφαρμογή του απαιτεί την επέκταση της τυπικής διαδικασίας Επιλογής Υπηρεσίας και την εισαγωγή ενός νέου σταδίου στον κύκλο ζωής: τη φάση Αξιολόγηση Υπηρεσίας. Εκτελώντας πειράματα στο περιβάλλον αυτό, αναδείχθηκε η αναμενόμενη συνεισφορά των συνεργατικών τεχνικών φιλτραρίσματος στην επιλογή από τον πελάτη της υπηρεσίας που καλύπτει τις

απαιτήσεις του πιθανότατα με βέλτιστο τρόπο. Η ανταλλαγή των δεδομένων μεταξύ των διαφόρων πελατών έγινε με μεσολάβηση μιας έμπιστης τρίτης οντότητας ως ενδιάμεσου.

Η κύρια αξίωση που έχει ο παρών μηχανισμός από κάθε τέτοιο υπηρεσιοστρεφές περιβάλλον είναι να επιτρέπει τη διαμόρφωση εικονικών οργανισμών (VOs), ως δομές εμπιστοσύνης των συμβαλλόντων όπου η διαχειριστική οντότητα του VO ενεργεί ως οντότητα κοινής εμπιστοσύνης. Η έννοια του εικονικού οργανισμού είναι καθοριστική, καθώς σύμφωνα με τον ορισμό του, οι μετέχοντες σε έναν εικονικό οργανισμό συνεργάζονται μεταξύ τους για να εξυπηρετήσουν τα κοινά τους συμφέροντα. Αυτό βέβαια στην πράξη ενέχει ζητήματα που σχετίζονται με την αξιοπιστία της διαχειριστικής οντότητας. Η οντότητα αυτή πρέπει να συλλέγει μέσω ενός συστήματος βαθμολόγησης τις αξιολογήσεις των διαφόρων χρηστών αφού αυτοί καλέσουν διάφορες υπηρεσίες. Εφόσον οι χρήστες αυτοί αξιολογούν μόνο τους όρους του SLA στους οποίους συμφώνησαν για την κλήση κάθε υπηρεσίας (οι οποίοι είναι και αυτοί που τους είναι κατανοητοί), ο μηχανισμός βασίζεται στην αξιολόγηση κάθε όρου (διάστασης) ξεχωριστά. Συνεπώς, η αξιολόγηση μιας υπηρεσίας από έναν πελάτη συνίσταται στην παροχή ενός διανύσματος αξιολογήσεων, παρά μιας μόνο βαθμολογίας (πολυδιάστατη). Υπολογίζοντας το βαθμό συσχέτισης των πελατών στον εικονικό οργανισμό με τη χρήση των διανυσμάτων αξιολογήσεων, μπορεί να παραχθεί μια έγκυρη σύσταση για μια συγκεκριμένη υπηρεσία επιθυμητής απόδοσης με βάση τις προτιμήσεις των πιο "κοντινών" χρηστών στον ερωτώντα.

Μια άλλη προσέγγιση θα ήταν να υπολογιστούν οι συσχετίσεις μόνο με τους χρήστες που χρησιμοποιούν κοινούς όρους SLA, παρά κοινές υπηρεσίες. Το πρόβλημα εδώ θα ήταν ο διαφορετικός βαθμός σημαντικότητας που έχει ένας όρος SLA για τους

πελάτες όταν αναφέρεται σε διαφορετικές υπηρεσίες. Παραδείγματος χάρη, ο χρόνος απόκρισης σαν παράμετρος ενός SLA έχει διαφορετικό βάρος όταν ο τύπος υπηρεσίας είναι το "3D Rendering" ή η "Ανάλυση συναισθημάτων σε κοινωνικά δίκτυα", με βάση βεβαίως και το ποιος είναι ο τελικός χρήστης (πχ. 3D animator ή δημοσιογράφος αντίστοιχα) και πόσο επείγουσα είναι η κλήση της υπηρεσίας, που μπορεί να συνδέεται με τη γρηγορότερη ανάκτηση των αποτελεσμάτων σε σχέση με τον ανταγωνισμό. Ωστόσο, η στάθμιση ενός όρου SLA με βάση τη σημασία του σε ένα συγκεκριμένο είδος υπηρεσίας στο οποίο και χρησιμοποιείται, έχει νόημα στην περίπτωση που ο πελάτης έχει επίγνωση του συγκεκριμένου τύπου υπηρεσίας που καλείται, κάτι που δεν ισχύει πάντα. Συνεπώς, η προσέγγιση που προτιμήθηκε εδώ είναι η χρήση όρων του SLA από τύπους υπηρεσιών που χρησιμοποιούν και οι δύο πελάτες που περιλαμβάνονται κάθε φορά στην ανάλυση συσχέτισης.

Μια από τις κύριες συνεισφορές της παρουσιασθείσας μεθόδου, είναι ότι η επιρροή των κακόβουλων χρηστών που επιδιώκουν να χειραγωγήσουν το σύστημα αξιολόγησης, ώστε πιθανώς να παραποιήσουν τα αποτελέσματα της ανάλυσης προς όφελός τους, ουσιαστικά ελαχιστοποιείται από το μοντέλο υποστήριξης αποφάσεων, εφόσον οι αξιολογήσεις τους σταθμίζονται αναλόγως. Αυτοί οι χρήστες, εμφανίζοντας αποκλίσεις από τις μέσες τιμές αλλά και μηδαμινή γραμμικότητα με τα διάφορα γκρουπ πελατών, έχουν πολύ μικρές τιμές συσχέτισης με τους υπόλοιπους πελάτες. Αν αυτό δε συμβαίνει, σημαίνει ότι απλά ο κακόβουλος χρήστης αξιολογεί τις υπηρεσίες με τον ίδιο τρόπο με τους άλλους χρήστες, οπότε και πάλι το τελικό αποτέλεσμα δεν επηρεάζεται σημαντικά. Η αδυναμία της μεθόδου είναι πιθανώς η περίπτωση όπου στο σύστημα συρρέουν επιτηδευμένες αξιολογήσεις από τέτοιους χρήστες που επιθυμούν να βλάψουν

ή να προωθήσουν έναν συγκεκριμένο πάροχο. Ωστόσο, δεδομένης της ύπαρξης ενός μεγάλου αριθμού πελατών και υπηρεσιών σε ένα τέτοιο υπηρεσιοστρεφές περιβάλλον, και πάλι οι κακόβουλες βαθμολογίες θα είναι έξω από τα υπόλοιπα γκρουπ πελατών, ελαχιστοποιώντας την επίδρασή τους.

Ένα άλλο ανοικτό θέμα είναι η επίδραση του χρόνου, εφόσον οι παλιότερες και οι νεότερες αξιολογήσεις πελατών έχουν διαφορετική επίδραση στη συνολική αξιολόγηση σε δεδομένο χρόνο. Έτσι, σαν μελλοντική κατεύθυνση στην παρούσα εργασία θα έπρεπε πιθανόν να αξιολογηθεί και ο παράγοντας του χρόνου, ως ένα "βάρος" σε κάθε βαθμολογία, μειώνοντας τη συνεισφορά της σε βάθος χρόνου. Επιπροσθέτως, εμπειρίες χαμηλής ποιότητας κατά το παρελθόν θα μπορούσαν να αντισταθμιστούν με πρόσφατες καλύτερης απόδοσης και αντιστρόφως.

6

Αξιολόγηση και Δυναμική Διαχείριση Πόρων από το μέρος του Παρόχου

Η μέθοδος αξιολόγησης που παρουσιάζεται στο κεφάλαιο αυτό χρησιμοποιεί το ιστορικό εκτέλεσης εργασιών (κλήσεις της υπηρεσίας) του παρόχου καθώς και πληροφορίες δέσμευσης των πόρων σε πραγματικό χρόνο, με σκοπό τη δυναμική ανακατανομή τους σε εφαρμογές που παρέχονται ως υπηρεσίες (Aisopos, Tserpes, & Varvarigou, 2011). Έτσι οι πληροφορίες που αντλούνται από το χρόνο εκτέλεσης καθώς και η αξιολόγηση της επίδοσης των υπηρεσιών κατά το παρελθόν βοηθάνε στην καλύτερη εσωτερική οργάνωση του παρόχου και στη βελτίωση της παρεχόμενης Ποιότητας Υπηρεσίας. Η εφαρμογή του παρόντος μηχανισμού σε περιβάλλοντα Software as a Service όπως θα δούμε, απαιτεί την εικονικοποίηση (virtualization) των πόρων του παρόχου, που επιτρέπει την ενιαία και δυναμική διαχείρισή τους. Οι τεχνικές εικονικοποίησης εμφανίστηκαν με την εισαγωγή του υπολογιστικού νέφους, αλλά πρακτικά μπορούν να υλοποιηθούν και σε συστήματα πλέγματος, καθώς και σε γενικότερα περιβάλλοντα

Software as a Service, για τη δυναμική παροχή υπολογιστικών πόρων από τους παρόχους των εφαρμογών.

Σε γενικές γραμμές, στα SaaS περιβάλλοντα, με ελάχιστες εξαιρέσεις όπως πχ. στο (Georgina Gallizo, 2009), τα SLAs είναι μακροπρόθεσμα και προσανατολισμένα στους πόρους (T-SLAs). Αυτό πρακτικά σημαίνει ότι ο φορέας εγγυάται τη διάθεση μιας ορισμένης ποσότητας πόρων για την εφαρμογή που χρησιμοποιεί ο καταναλωτής, η οποία μπορεί να τους χρησιμοποιήσει ανά πάσα στιγμή σε οποιοδήποτε βαθμό κατά τη διάρκεια ζωής της συμφωνίας. Αυτό συνεπάγεται, ωστόσο, ότι δεν είναι απαραίτητο να δεσμεύονται όλοι οι συμφωνηθέντες πόροι για λογαριασμό των καταναλωτών καθ' όλη τη διάρκεια ζωής του SLA. Η παρατήρηση αυτή είναι ιδιαίτερα σημαντική, καθώς οι πάροχοι τείνουν να υιοθετούν μη ευέλικτα σχήματα διαχείρισης, όπως η πρωθύστερη κράτηση πόρων υπό το φόβο συντέλεσης παραβιάσεων του SLA.

Σε κάθε περίπτωση, εάν οι υπό διεκπεραίωση εργασίες, που υποβάλλονται από τους πελάτες όταν καλούν μια SaaS υπηρεσία, απαιτούν με βάση τα SLAs περισσότερους συνολικά πόρους από εκείνους που διατίθενται, ο πάροχος υπηρεσιών έρχεται αντιμέτωπος με ένα πρόβλημα χωρητικότητας. Αν το απόθεμα πόρων δεν αρκεί για να καλύψει τη ζήτηση, οι πάροχοι προσφεύγουν σε εξωτερική ανάθεση των εργασιών ή σε χορήγηση επιπλέον πόρων, στην περίπτωση που κρατούν επιπλέον πόρους ως εφεδρικούς. Αν και αυτό δεν αποτελεί σημαντικό πρόβλημα για μεγάλους οργανισμούς, όπως η περίπτωση του Amazon που αναφέρθηκε στα προηγούμενα, με πρακτικά απεριόριστους διαθέσιμους πόρους, για μικρομεσαίες επιχειρήσεις σημαίνει μίσθωση επιπλέον πόρων ή απλά μη εκτέλεση των εργασιών (παραβίαση του SLA). Το κόστος αυτών των εναλλακτικών λύσεων καταδεικνύει την έλλειψη ευελιξίας στην αξιολόγηση

του κινδύνου ανάληψης της διεκπεραίωσης μιας εργασίας χρησιμοποιώντας ελεύθερους πόρους, ακόμη και αν δεν επαρκούν για να καλύψουν τις απαιτήσεις ποιότητας τη συγκεκριμένη χρονική στιγμή.

Η εφαρμογή ενός μηχανισμού για αποτελεσματική κατανομή των πόρων σε πραγματικό χρόνο κατατάσσεται σε μια οικογένεια μη-ντετερμινιστικών δύσκολων προβλημάτων βελτιστοποίησης πολυωνυμικού χρόνου NP-hard (Non-deterministic Polynomial-time hard) με πολλές παραμέτρους. Στο δεύτερο αυτό σκέλος της Διατριβής παρουσιάζεται μια λύση για την επίτευξη της μέγιστης αξιοποίησης των πόρων με ταυτόχρονη διαχείριση του κινδύνου πιθανών παραβιάσεων του SLA για εκκρεμούσες εργασίες που αποφέρουν το μικρότερο κέρδος για τον πάροχο. Στα πλαίσια αυτού του μηχανισμού, γίνονται δύο βασικές παραδοχές που μπορεί να επηρεάζουν ως ένα βαθμό την εφαρμοσιμότητα της πρότασης: α) όλες οι εργασίες που υποβάλλονται μπορούν να διακοπούν και να συνεχισθούν σε μεταγενέστερο στάδιο και β) το σύνολο των πόρων είναι ομοιογενές (ή αντιμετωπίζεται ως τέτοιο).

Η προτεινόμενη λύση αντιμετωπίζει την υποκείμενη υποδομή του SaaS περιβάλλοντος ως μια ομάδα ελαστικών πόρων, με χρήση οποιασδήποτε τεχνολογίας για την εικονικοποίησή τους. Συνεπώς, οι πόροι χωρίζονται λογικά σε μικρά κομμάτια και διατίθενται για τις εισερχόμενες εργασίες. Το ποσό των χορηγούμενων πόρων ποικίλει ανάλογα με τον φόρτο που δημιουργούν οι απαιτήσεις της κάθε εργασίας. Ο όρος «φόρτος εργασίας» αναφέρεται στην ποσότητα των πόρων που πρέπει να διατεθούν προκειμένου να ικανοποιηθούν πλήρως οι QoS απαιτήσεις του πελάτη, όταν αυτός υποβάλλει μια εργασία. Ένας πιο σαφής όρος θα μπορούσε να είναι ο «απαιτούμενος φόρτος εργασίας», ο οποίος υπολογίζεται από τον πάροχο κατά την αντιστοίχιση των

απαιτήσεων υψηλού επιπέδου σε παραμέτρους χαμηλού επιπέδου. Συνεπώς, εξαρτάται από τις απαιτήσεις του πελάτη αλλά και από το χρόνο, εφόσον ο απαιτούμενος φόρτος εργασίας μειώνεται όταν ένα μέρος της εργασίας έχει ολοκληρωθεί και μπορεί να αυξηθεί όταν μια εργασία παραμένει στην ουρά προς εκτέλεση για μεγάλο χρονικό διάστημα.

Το πρόβλημα τώρα είναι πώς να μεγιστοποιηθεί η αξιοποίηση των πόρων, με ταυτόχρονη μεγιστοποίηση του κέρδους (και άρα ελαχιστοποίηση του κινδύνου παραβιάσεων SLA). Για την επίλυσή του θα καταφεύγουμε σε ένα πολύ γνωστό πρόβλημα, το οποίο παρουσιάζει αναλογίες με το δικό μας: το πρόβλημα του σακιδίου (Knapsack):

«Δοθέντος ενός συνόλου αντικειμένων (εργασιών προς υποβολή), το καθένα με ένα βάρος (φόρτος εργασίας) και μια αξία (κέρδος), καθορίστε τα αντικείμενα και τις ποσότητες που πρέπει να συμπεριληφθούν σε μια συλλογή (σύνολο εργασιών που θα εκτελεστούν / πόρων πρέπει να δεσμευτούν), έτσι ώστε το συνολικό βάρος (συνολικός φόρτος εργασίας) να είναι μικρότερο ή ίσο με ένα συγκεκριμένο όριο (χωρητικότητα υποδομής) και η συνολική αξία (συνολικό κέρδος) είναι όσο το δυνατόν μεγαλύτερη».

Λαμβάνοντας υπόψη την παραπάνω αναλογία, ανάγουμε το πρόβλημα της κατανομής των πόρων στο πρόβλημα του Knapsack και διερευνούμε τις πιθανές λύσεις. Για να καταλήξουμε στην καλύτερη δυνατή λύση, πρέπει να μοντελοποιηθεί το προκύπτον κέρδος από την εκτέλεση κάθε εργασίας. Το κέρδος στην περίπτωση μας είναι μια συνάρτηση πολλών παραμέτρων, που εξαρτάται από την πιθανότητα της παραβίασης του SLA με δεδομένο αριθμό δεσμευόμενων πόρων, το κόστος της αποζημίωσης και το κόστος λειτουργίας των υποδομών που συνάγεται η εκτέλεση μιας

εργασίας. Για να αξιολογηθεί ο προτεινόμενος μηχανισμός, υλοποιείται ένα σχήμα κατανομής πόρων στο επίπεδο μετα-χρονοπρογραμματισμού ενός κατανεμημένου συστήματος. Στη φάση της αξιολόγησης, δύο συνήθεις εναλλακτικοί μηχανισμοί κατανομής καθώς και διάφορες στρατηγικές τιμολόγησης χρησιμοποιούνται, έτσι ώστε να δοκιμαστεί το μοντέλο έναντι του πιθανού κέρδους που αποφέρει σε διαφορετικές συνθήκες.

6.1 Μοντέλο αξιολόγησης και διαχείρισης πόρων

6.1.1 Μοντελοποίηση προβλήματος

Ένα βασικό ζήτημα που προκύπτει στις υπηρεσιοστρεφείς αρχιτεκτονικές πραγματικού χρόνου, όπως αυτές που περιγράφονται παραπάνω, είναι όπως αναφέρθηκε το πρόβλημα της αποδοτικής διαχείρισης των πόρων: Δεδομένου ότι η χωρητικότητα των πόρων, έστω M , είναι περιορισμένη και πολλοί καταναλωτές υπηρεσιών του παρόχου απαιτούν πόρους βάσει συγκεκριμένων ηλεκτρονικών συμβολαίων (SLAs) που εγγυώνται τη διαθεσιμότητα, ευριστικοί αλγόριθμοι και "έξυπνες" τεχνικές απαιτούνται για να κατανέμουν τους διαθέσιμους πόρους, ικανοποιώντας τους πελάτες όσο το δυνατόν περισσότερο. Στο μοντέλο αυτό, υποθέτουμε ότι συγκεκριμένες απαιτήσεις QoS για τον όρο "slowdown" (συνήθης όρος στα SLA για το χρόνο αναμονής μιας αίτησης σε σχέση με το χρόνο εξυπηρέτησής της), αντανακλώντας την προϋπόθεση ότι όλες οι αιτήσεις εργασίας θα πρέπει να αντιμετωπίζονται ισότιμα ανεξάρτητα από τους πόρους που απαιτούνται, δεν υπάρχουν, ώστε να εξασφαλιστεί η μέγιστη δυνατή ευελιξία.

Εάν σε ένα τέτοιο μοντέλο ο συνολικός απαιτούμενος φόρτος εργασίας, έστω W , υπερβαίνει τη χωρητικότητα του συστήματος M , η πλήρης δέσμευση των πόρων έρχεται σε σύγκρουση με τη διαφύλαξη των όρων ποιότητας για όλους τους πελάτες. Μερικές από τις απαιτήσεις των πελατών δεν μπορούν να ικανοποιηθούν, ή τουλάχιστον να ικανοποιηθούν πλήρως, λόγω της έλλειψης επαρκών πόρων, με αποτέλεσμα την απώλεια κερδών μέσω παραβιάσεων SLA, που τελικά θα συμβούν. Είναι επομένως προφανές ότι χρειάζονται κατάλληλοι δυναμικοί μηχανισμοί κατανομής πόρων, με χρήση τεχνικών διαχείρισης κινδύνου για τέτοιες περιπτώσεις, προκειμένου να ελαχιστοποιηθεί αυτή η απώλεια, μεγιστοποιώντας παράλληλα την αξιοποίηση των πόρων.

6.1.2 Απλοποίηση παραμέτρων για μία διάσταση

Στα εικονικοποιημένα SaaS περιβάλλοντα αναφοράς που περιγράφονται στις προηγούμενες ενότητες, το ενιαίο σύνολο πόρων του συστήματος αποτελείται από διάφορα διαθέσιμα στοιχεία της υποδομής (μνήμη, επεξεργαστές, δίσκος, κλπ.). Ωστόσο, αρχικά θα εξετάσουμε μία μόνο "διάσταση" του συνόλου, για παράδειγμα τη μνήμη του συστήματος. Η συνολική μνήμη (χωρητικότητα) ενός τέτοιου συστήματος συμβολίζεται ως M . Ας υποθέσουμε τώρα ότι υπάρχουν R υπο εκτέλεση εργασίες που απαιτούν πόρους συστήματος, οι οποίες υποβλήθηκαν από τους καταναλωτές των υπηρεσιών, με κάθε μία να δηλώνεται ως $j_i, i \in \{1, 2, \dots, R\}$. Οι εργασίες αυτές μπορεί να αντιπροσωπεύονται από εκτελέσιμα αρχεία μαζί με αρχεία και παραμέτρους εισόδου, που παρέχονται από τον καταναλωτή. Ανάλογα με τον τύπο κάθε εργασίας, καθώς και κάποιες παραμέτρους εισόδου (προθεσμία εκτέλεσης, ποιότητα αποτελέσματος), η πλευρά του παρόχου εκτιμά τις υπολογιστικές απαιτήσεις της, όπως περιγράφεται στην αρχή του κεφαλαίου.

Ας υποθεθεί ότι κάθε εκκρεμής εργασία j_i απαιτεί ένα ορισμένο ποσό μνήμης w_i , όπως υπαγορεύεται από τους όρους του αντίστοιχου SLA του πελάτη. Η συνολική μνήμη που απαιτείται (απαιτούμενος φόρτος εργασίας) μπορεί να περιγραφεί από ένα σύνολο φορτίων εργασίας: $W = \{w_1, w_2, \dots, w_R\}$. Επιπροσθέτως, η επιτυχής εκτέλεση κάθε εργασίας παράγει ένα ορισμένο κέρδος, με βάση το κόστος της εργασίας που ορίζεται στο SLA, συνεπώς το σύνολο των κερδών μπορεί να συμβολισθεί ως $P = \{p_1, p_2, \dots, p_R\}$, δηλώνοντας το κέρδος του παρόχου εάν κάθε εργασία εκτελείται χωρίς κάποια παραβίαση του SLA. Εάν ο συνολικός απαιτούμενος φόρτος εργασίας είναι μεγαλύτερος από τη χωρητικότητα του συστήματος ($\sum w_i > M$), τότε θα πρέπει να αποφασιστεί ποιες από τις εργασίες θα εκτελεστούν σε μια δεδομένη χρονική περίοδο. Οι μακροπρόθεσμοι στόχοι είναι αφενός να επιτευχθεί η μέγιστη αξιοποίηση της διαθέσιμης μνήμης στην υποδομή, και αφετέρου να βγει το μέγιστο κέρδος από αυτές, ανάλογα με κάθε κόστος εργασίας.

Τα προβλήματα κατανομής στοιχείων με αξίες και χρονο-προγραμματισμού με οικονομικούς περιορισμούς συνήθως επιλύονται μέσω αναγωγής σε ένα γνωστό NP-πλήρες πρόβλημα, όπως το πρόβλημα του σακιδίου (Antonella Basso, 2001). Στην περίπτωση μας, προκειμένου να βρεθεί μια εφικτή και αποτελεσματική λύση στο περιγραφόμενο πρόβλημα της διαχείρισης των πόρων, αποφασίστηκε να αναχθεί στο Fractional Knapsack (παραλλαγή του παραπάνω προβλήματος), με δυνατότητα λύσης από διάφορους αποδοτικούς αλγόριθμους, μερικοί από τους οποίους παρουσιάζονται παρακάτω. Στις ακόλουθες ενότητες περιγράφονται δύο βασικές παραλλαγές του προβλήματος του σακιδίου (0/1 Knapsack και Fractional Knapsack) και η αναγωγή του προβλήματος διαχείρισης πόρων σε αυτό.

6.2

Αναγωγή στο πρόβλημα του

σακιδίου

6.2.1 Ορισμός του προβλήματος του σακιδίου

6.2.1.1 Το δυαδικό πρόβλημα

Το κλασικό δυαδικό πρόβλημα του σακιδίου (0/1 Knapsack), γνωστό και ως το πρόβλημα του «σάκου του ληστή», είναι ένα πρόβλημα συνδυαστικής βελτιστοποίησης: Δίνονται δύο σύνολα R θετικών ακεραίων, $P = \{p_1, p_2, \dots, p_R\}$ και $W = \{w_1, w_2, \dots, w_R\}$, και ένας ακέραιος M . Τα w_i μπορεί να ερμηνευθούν ως μεγέθη ή βάρη των αντικειμένων $1, 2, \dots, R$ και το M ως το μέγεθος του σακιδίου που έχει ο ληστής. Τα p_i παρέχουν τις αξίες των αντικειμένων αυτών. Παράλληλα, κάθε αντικείμενο i καταλαμβάνει χώρο w_i όταν τοποθετείται στο σακίδιο. Ο στόχος είναι να γεμίσει το σακίδιο με πολλά αντικείμενα με τέτοιο τρόπο ώστε η συνολική τους αξία να μεγιστοποιείται. Από μαθηματική άποψη, το 0/1 Knapsack μπορεί να διατυπωθεί ως:

Μεγιστοποιήστε το: $\sum p_i \delta_i$,

με την προϋπόθεση ότι: $\sum w_i \delta_i \leq M$ και $\delta_i = 0,1$.

6.2.1.2 Το κλασματικό πρόβλημα

Το κλασματικό πρόβλημα του σακιδίου (Fractional Knapsack) είναι μια παραλλαγή του κλασικού προβλήματος. Εδώ δίνονται αντικείμενα με διαφορετικές αξίες ανά μονάδα όγκου και ανώτατα ποσά, ενώ είμαστε σε θέση να λάβουμε τα κομμάτια (κλάσματα) των αντικειμένων ($\delta_i \in [0,1]$), και πρέπει να βρεθεί το πιο πολύτιμο μίγμα των αντικειμένων που ταιριάζουν σε ένα σακίδιο σταθερού όγκου. Από τη στιγμή που μπορούν να ληφθούν

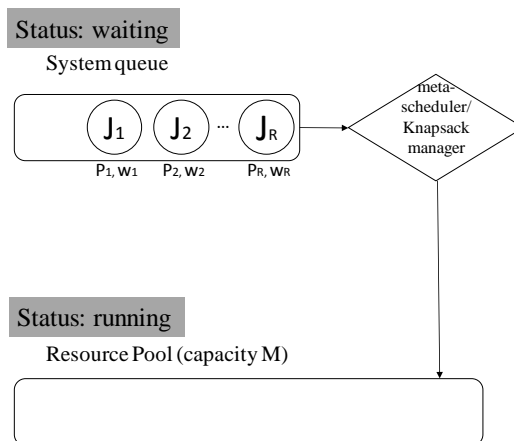
κλάσματα των αντικειμένων αντί μιας πλήρους μονάδας για κάθε ένα, ένας greedy αλγόριθμος βρίσκει τη βέλτιστη λύση: Παίρνουμε όσο το δυνατόν περισσότερο από το αντικείμενο που είναι το πιο πολύτιμο ανά μονάδα όγκου. Αν εξακολουθεί να υπάρχει χώρος, παίρνουμε όσο το δυνατόν περισσότερο από το επόμενο πιο πολύτιμο αντικείμενο. Συνεχίζουμε μέχρι το σακίδιο να είναι πλήρες. Από μαθηματική άποψη, το κλασματικό πρόβλημα μπορεί να διατυπωθεί ως εξής:

Μεγιστοποιήστε το: $\sum p_i \delta_i$,

με την προϋπόθεση ότι: $\sum w_i \delta_i \leq M$ και $\delta_i \in [0,1]$.

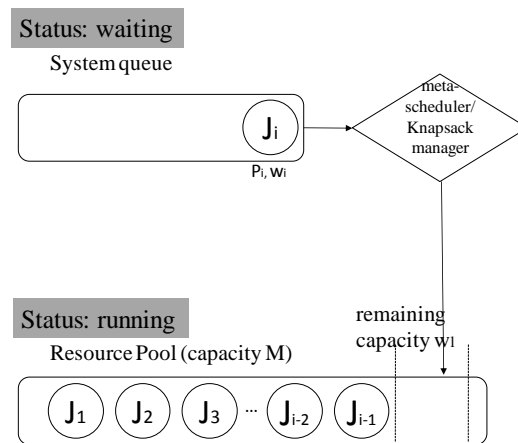
6.2.2 Αναγωγή του προβλήματος διαχείρισης πόρων

Το πρόβλημα διαχείρισης των πόρων (παράγραφος 6.1.1) μπορεί εύκολα να αναχθεί στο πρόβλημα του σακιδίου: το σύνολο της απαιτούμενης μνήμης $W = \{w_1, w_2, \dots, w_R\}$ αντιστοιχεί στο σύνολο των βαρών και η συνολική μνήμη M στο μέγεθος του σακιδίου, ενώ το σύνολο των κερδών $P = \{p_1, p_2, \dots, p_R\}$ για τις εκκρεμείς εργασίες στις αξίες των αντικειμένων, κάτι που οδηγεί σε ένα κλασικό δυαδικό πρόβλημα Knapsack (Σχήμα 35).



Σχήμα 35: Το πρόβλημα διαχείρισης πόρων μοντελοποιημένο ως το κλασικό Knapsack πρόβλημα

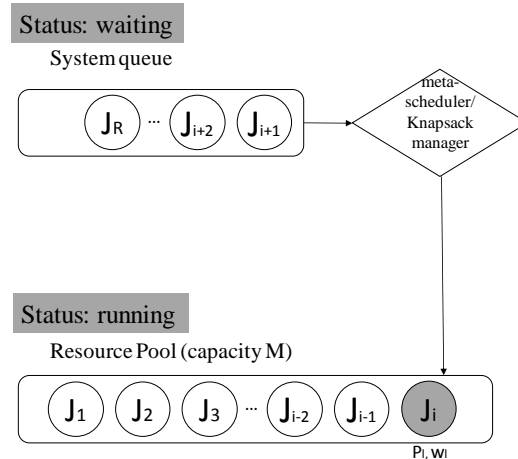
Στην περίπτωση μας, ωστόσο, η κατανομή των πόρων στις εισερχόμενες εργασίες είναι δυναμική και μπορεί να είναι κλασματική, λόγω της ελαστικότητάς τους. Για παράδειγμα, έστω ότι οι εργασίες j_1 έως και j_{i-1} τρέχουν ήδη και μια νέα εργασία j_i καταφτάνει στην ουρά του συστήματος ζητώντας πόρους. Αν η j_i απαιτεί μνήμη w_i και η διαθέσιμη μνήμη w_l είναι μικρότερη από το απαιτούμενο ποσό ($w_l < w_i$), τότε μόνο ένα τμήμα w_l της επιθυμητής ποσότητας w_i μπορεί να διατεθεί για την j_i , λαμβάνοντας υπόψη μια σοβαρή πιθανότητα η j_i να αποτύχει ή τα αποτελέσματα να μην πληρούν τις απαιτήσεις ποιότητας του πελάτη που την υπέβαλλε (Σχήμα 36).



Σχήμα 36: Μια νεοεισερχόμενη εργασία j_i ζητά περισσότερους πόρους από τους διαθέσιμους

Αυτό μας οδηγεί στην μοντελοποίηση του προβλήματος ως ένα Fractional Knapsack. Μια αποδοτική λύση του κλασματικού προβλήματος του σακιδίου όπως αναφέρθηκε είναι με έναν άπληστο (greedy) αλγόριθμο (οι διάφορες εναλλακτικές λύσεις παρουσιάζονται αναλυτικά παρακάτω). Η ιδέα είναι να υπολογιστεί για κάθε αντικείμενο (εργασία) η πυκνότητα κέρδους (αναλογία κέρδος / βάρος), και να ταξινομηθούν σύμφωνα με την αναλογία αυτή. Ένας τυπικός αλγόριθμος ταξινόμησης, στην καλή περίπτωση έχει πολυπλοκότητα $O(n \log n)$ και στη χειρότερη $O(n^2)$. Τα

αντικείμενα (εργασίες) με την υψηλότερη πυκνότητα κέρδους θα πρέπει να επιλέγονται και να προστεθούν στο σακίδιο (περιβάλλον εκτέλεσης) μέχρι που να μην μπορεί να προστεθεί το επόμενο αντικείμενο (εργασία) ολόκληρο. Τέλος προστίθεται όσο είναι δυνατόν από το επόμενο αντικείμενο (κατανομή υπόλοιπης μνήμης στην επόμενη εργασία από την ταξινομημένη ουρά):



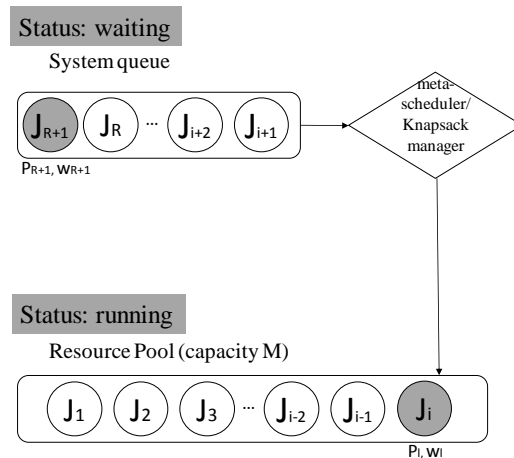
Σχήμα 37: Εκτέλεση της εργασίας με την υψηλότερη αναλογία p_i / w_i στο σύστημα. Οι υπόλοιπες περιμένουν στην ουρά

Προσαρμοσμένος στην περίπτωση μας, για την R εισερχόμενες εργασίες ο αλγόριθμος αυτός (Αλγόριθμος II) ορίζεται αναλυτικά ως εξής:

Για το σύνολο των εργασιών $J = \{j_1, j_2, \dots, j_R\}$ ακολούθησε τα βήματα:

1. Ταξιλόγησε τον πίνακα $\left[\frac{p_1}{w_1}, \frac{p_2}{w_2}, \dots, \frac{p_R}{w_R} \right]$
2. Δέσμευσε τους πόρους που ζητά η πρώτη εργασία και αφάιρέσέ την από τον πίνακα.
3. Επανάλαβε το βήμα 2 μέχρι οι απαιτήσεις της επόμενης εργασίας να μην ικανοποιούνται από τους διαθέσιμους πόρους
4. Για τις εργασίες που απομένουν, διάλεξε αυτή (j_i) με τη μεγαλύτερη αναλογία $\frac{p_i}{w_i}$ (αφού υπολογιστούν πάλι οι αξίες) και δώσε τους υπόλοιπους διαθέσιμους πόρους.

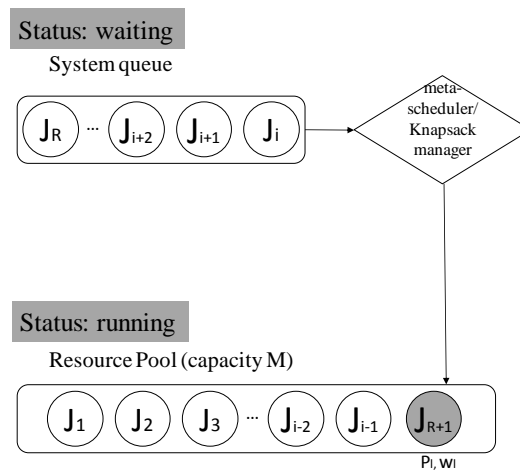
Εδώ πρέπει να σημειωθεί ότι οι αξίες (τα κέρδη που αποφέρουν οι εργασίες) θα πρέπει να υπολογιστούν εκ νέου, όπως προβλέπεται στο βήμα 4, δεδομένου ότι η συνάρτηση του "κέρδους" για μια εργασία αλλάζει με το χρόνο, όπως θα συζητηθεί στην επόμενη παράγραφο. Η συνάρτηση αυτή έχει ως ένα βαθμό εξάρτηση από την πιθανότητα παραβίασης του SLA, κι έτσι θα αλλάξει αν οι πόροι που διατίθενται είναι λιγότερο από εκείνους που απαιτούνται από την εργασία. Η πολυπλοκότητα αυτής της νέας αναδιάταξης μετά τον επαναυπολογισμό των κερδών είναι και πάλι $O(n \log n)$, χωρίς να λαμβάνεται υπόψη η πολυπλοκότητα υπολογισμού του κέρδους. Ο αλγόριθμος πρέπει να εκτελείται κάθε φορά που μια νέα εργασία εισέρχεται στην ουρά εκτέλεσης, καθώς η προτεραιότητά της μπορεί ενδεχομένως να είναι μεγαλύτερη από αυτήν μίας από τις εργασίες που ήδη εκτελούνται (Σχήμα 38), ή κάθε φορά που απελευθερώνονται πόροι μετά την ολοκλήρωση της εκτέλεσης εργασιών.



Σχήμα 38: Μια νέα εργασία j_{R+1} εισέρχεται στην ουρά, με υψηλότερη προτεραιότητα από την εργασία j_{i-1} που ήδη εκτελείται

Οι δύο συνθήκες που αναφέρθηκαν παραπάνω καθορίζουν τις συνθήκες "ενεργοποίησης" του μηχανισμού. Στην πρώτη περίπτωση, η ταξινόμηση των

προτεραιοτήτων (πυκνότητες κέρδους) θα πρέπει να γίνει ξανά, δεδομένου ότι μπορεί να είναι πιο επικερδής η κατανομή πόρων στη νέα εργασία και να τοποθετηθεί μια εργασία που ήδη εκτελείται στην ουρά σε συνθήκες "παύσης", όπως στο παράδειγμα του Σχήμα 39 (η απαιτούμενη μνήμη της νεοεισερχόμενης εργασίας w_{R+1} εδώ είναι ίση με την εναπομείνασα w_i , για λόγους απλότητας). Στη δεύτερη περίπτωση, οι απελευθερωμένοι πόροι που προκύπτουν από τον τερματισμό της εκτέλεσης εργασιών, θα πρέπει να διατίθενται για εργασίες που περιμένουν στην ουρά, προκειμένου το "σακίδιο" να είναι και πάλι όσο το δυνατόν πλήρες.



Σχήμα 39: Δεσμεύονται πόροι για την εργασία j_{R+1} , η εργασία δουλειά j_{i-1} πάει στην ουρά

6.2.2.1 Ευριστική συνάρτηση υπολογισμού του κέρδους

Ένα σημαντικό ζήτημα που ανακύπτει τώρα είναι ο υπολογισμός του κέρδους p_i για κάθε εργασία j_i . Για το σκοπό αυτό θα πρέπει να ληφθεί υπόψη το προαναφερθέν κόστος της εργασίας, το οποίο είναι μια απλή τιμή που ορίζεται στο SLA και συμβολίζεται εδώ ως PR_i , το (αμελητέο συνήθως) κόστος επεξεργασίας και εκτέλεσης στους πόρους ενός συστήματος EC_i , καθώς και η πιθανότητα να συμβεί κάποια αστοχία, όπως αναφέρθηκε παραπάνω, και οι σχετικές δαπάνες. Δεδομένου ότι η κλήση μιας υπηρεσίας βασίζεται σε

ένα SLA, μια πιθανή παραβίαση του, η οποία μπορεί επίσης να οφείλεται στο γεγονός ότι δεν επιστρέφει αποτελέσματα ικανοποιητικής ποιότητας, θα επιφέρει αποζημίωση. Η αποζημίωση αυτή είναι ένα κόστος που επίσης ορίζεται στους όρους της ηλεκτρονικής σύμβασης, και συμβολίζεται ως κόστος παραβίασης VC_i .

Φυσικά, ακόμα και αν δεσμευθούν πλήρως οι πόροι που απαιτούνται, η εφαρμογή που θα καλεστεί ως υπηρεσία μπορεί να μην λειτουργήσει σωστά ή μην καλύψει τις QoS παραμέτρους που συμφωνήθηκαν, κάτι που μπορεί να οφείλεται σε άλλους λόγους, όπως απρόβλεπτες απαιτήσεις πόρων, τυχαία λάθη κατά το χρόνο εκτέλεσης κλπ. Έτσι, η συνολική πιθανότητα παραβίαση VP_i πρέπει να εκτιμηθεί, λαμβάνοντας υπόψη το ιστορικό των αστοχιών και των παραβιάσεων SLA που παρατηρήθηκαν στο παρελθόν για κάθε εφαρμογή, καθώς και την ποσότητα των πόρων που παρέχονται στην εργασία j_i . Λαμβάνοντας υπόψη τα όλα παραπάνω, ένα πρώτο προφανές μέτρο του κέρδους εκτέλεσης της εργασίας j_i για τον πάροχο P είναι:

$$p_{i,1} = PR_i - VP_i \cdot VC_i - EC_i$$

Εξίσωση 5: Πρώτο μέτρο του κέρδους εκτέλεσης για την εργασία j_i

Δεδομένης της περιορισμένης χωρητικότητας του συστήματος, όταν πολλαπλές εργασίες που περιμένουν στην ουρά απαιτούν πόρους, συμβολίζουμε SP_i την πιθανότητα επιτυχημένης εκτέλεσης, η οποία εξαρτάται από τους πόρους που απομένουν (ικανοποιώντας πιθανόν μόνο ένα μέρος των απαιτήσεων της εργασίας): $SP_i = 1 - VP_i$. Συνεπώς, ένα άλλο μέτρο που καταδεικνύει το κέρδος p_i είναι το κόστος παραβίασης που ο πάροχος θα εξοικονομήσει, σε περίπτωση που η εργασία εκτελεστεί με επιτυχία:

$$p_{i,2} = SP_i \cdot VC_i$$

Εξίσωση 6: Κόστος που εξοικονομεί ο πάροχος από επιτυχή εκτέλεση της εργασίας j_i

Αν οι ελεύθεροι πόροι είναι μεγαλύτεροι από αυτούς που απαιτούνται από τις εκκρεμείς εργασίες, το παραπάνω μέτρο προσεγγίζει το VC_i , ειδάλλως εξαρτάται από την ποσότητα των διαθέσιμων πόρων: Όσο μεγαλύτερο ποσοστό των επιθυμητών πόρων είναι διαθέσιμο, τόσο περισσότερο κέρδος (και ως εκ τούτου προτεραιότητα, χρησιμοποιώντας τη greedy λύση) φέρει μια εργασία.

Από τη σύνθεση των δύο αυτών μέτρων του κέρδους, προκύπτει η συνολική συνάρτηση κέρδος p_i για κάθε εργασία j_i ως εξής:

$$P_{i,total} = P_{i,1} + P_{i,2} \Rightarrow$$

$$P_{i,total} = PR_i - VP_i \cdot VC_i - EC_i + SP_i \cdot VC_i$$

και άρα:

$$p_{i,total} = PR_i - 2 \cdot VP_i \cdot VC_i - EC_i + VC_i$$

Εξίσωση 7: Συνολική συνάρτηση κέρδους για μια εργασία j_i

Συνεπώς, το βήμα 1 του greedy αλγόριθμου της προηγούμενης ενότητας διαμορφώνεται ως εξής (Αλγόριθμος III):

Για το σύνολο των εργασιών $J = \{j_1, j_2, \dots, j_R\}$ ακολούθησε τα βήματα:

1. Ταξινόμησε τον πίνακα $\left[\frac{P_1}{w_1}, \frac{P_2}{w_2}, \dots, \frac{P_R}{w_R} \right]$, όπου

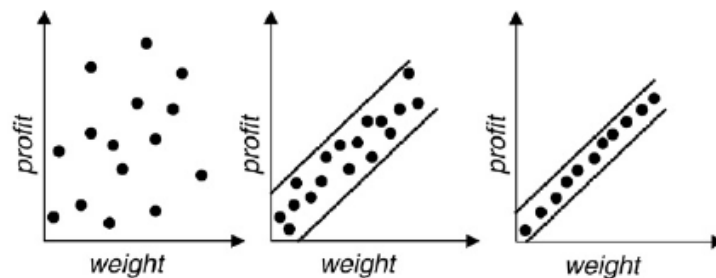
$$p_i = PR_i - 2 \cdot VP_i \cdot VC_i - EC_i + VC_i, i \in \{1 \dots R\}$$
2. Δέσμευσε τους πόρους που ζητά η πρώτη εργασία και αφάιρέσέ την από τον πίνακα.
3. Επανάλαβε το βήμα 2 μέχρι οι απαιτήσεις της επόμενης εργασίας να μην ικανοποιούνται από τους διαθέσιμους πόρους
4. Για τις εργασίες που απομένουν, διάλεξε αυτή (j_i) με τη μεγαλύτερη αναλογία $\frac{P_i}{w_i}$ (αφού υπολογιστούν πάλι οι αξίες) και δώσε τους υπόλοιπους διαθέσιμους πόρους.

Στο βήμα 4 ο επαναυπολογισμός του p_i είναι και πάλι απαραίτητος (πολυπλοκότητα $O(n)$) όπως και στον Αλγόριθμο II, δεδομένου ότι το p_i εξαρτάται από την πιθανότητα παραβίασης VP_i όπως περιγράψαμε πιο πάνω, και όταν οι ζητούμενοι πόροι δεν είναι πλήρως διαθέσιμοι αυτή η πιθανότητα αυξάνεται.

6.2.3 Εναλλακτικοί αλγόριθμοι για την επίλυση του προβλήματος Knapsack

Το πρόβλημα στο οποίο καταλήγουμε, όπως αναφέρθηκε ανήκει στην οικογένεια των NP-πλήρων προβλημάτων. Ως εκ τούτου, για να βρεθεί η βέλτιστη λύση, χρειάζεται μια βασική διερεύνηση των εναλλακτικών επιλογών. Στη βιβλιογραφία, περιπτώσεις (Litke, et al., 2007) που μελέτησαν μια παρόμοια αναγωγή με το τρέχον πρόβλημα, εξέτασαν διάφορους εναλλακτικές λύσεις, όπως αλγόριθμους Backtracking (BT), Branch and Bound (BB), Δυναμικού Προγραμματισμού (DP) και άπληστους ευριστικούς αλγόριθμους (greedy). Στην αναφερθείσα περίπτωση, η greedy τεχνική θεωρήθηκε η καλύτερη και τελικά επιλέχθηκε με σκοπό να βρεθεί μια σχεδόν βέλτιστη λύση, όπως θα γίνει και στην τρέχουσα εργασία επίσης.

Η απόδοση του εν λόγω αλγορίθμου εξαρτάται κυρίως από τα επιλεγθείσες ευριστικές συναρτήσεις, καθώς από το βαθμό συσχέτισης μεταξύ του κέρδους (profit) και του βάρους (weight), όπως για παράδειγμα σχηματοποιείται στο Σχήμα 40:



Σχήμα 40: Παραδείγματα ασυσχέτιστων, χαλαρά συσχετισμένων και στενά συσχετισμένων δεδομένων βαρών και κερδών

Στην περίπτωσή μας, είναι λογικό να υποθέσουμε ότι έχουμε να κάνουμε με χαλαρή συσχέτιση δεδομένων, αφού η τιμολογιακή πολιτική επηρεάζεται όχι μόνο από την κατανάλωση πόρων από κάθε εργασία, αλλά και άλλες παραμέτρους (διαχείριση σχέσεων με πελάτες, μεγάλες απαιτήσεις σε φόρτο εργασίας του συστήματος, καταστάσεις έκτακτης ανάγκης κλπ.) που ουσιαστικά απεικονίζονται στα SLAs.

6.2.4 Το πρόβλημα σε πολλές διαστάσεις

Μια πιο ρεαλιστική εφαρμογή του παραπάνω μοντέλου σε ένα εικονικοποιημένο σύστημα SaaS, πρέπει να λαμβάνει υπόψη ότι η διαχείριση των πόρων μιας συστάδας υπολογιστών περιλαμβάνει διάφορες διαστάσεις/πόρους (επεξεργαστές, δίσκος, μνήμη, κλπ). Συνεπώς, το βάρος w_i που αντιπροσώπευε την απαιτούμενη μνήμη από την εργασία j_i στις προηγούμενες ενότητες, θα είναι στην πραγματικότητα ένα διάνυσμα k ζητούμενων πόρων: $\vec{w}_i = [w_{i1}, w_{i2}, \dots, w_{ik}]$, ενώ η χωρητικότητα του συστήματος θα είναι ένα διάνυσμα μέγιστων διαθέσιμων τιμών: $\vec{M} = [w_{\max,1}, w_{\max,2}, \dots, w_{\max,k}]$.

Στη γνωστή βιβλιογραφία, όλες οι σχετικές ερευνητικές εργασίες αντιμετωπίζουν το πρόβλημα σε μία διάσταση, με την εξαίρεση του Paolo Campregiani (2009), όπου χρησιμοποιείται μια διαφορετική παραλλαγή του Knapsack στο επίπεδο του virtualization κάτι που δεν μπορεί να εφαρμοστεί στο υψηλό επίπεδο μεταδρομολόγησης που αναλύεται εδώ.

Δεδομένων των πολλών διαστάσεων που αντιμετωπίζουμε, στη συνέχεια ακολουθείται η κλασική προσέγγιση του Pirkul (1987) προσαρμόζοντας τη greedy λύση για το κλασματικό πρόβλημα του σακιδίου στα πολλαπλά βάρη (διαστάσεις). Σύμφωνα με αυτήν, τα αντικείμενα στο "σακίδιο" συσκευάζονται με βάση τη φθίνουσα σειρά της

αναλογίας κέρδους / βάρους, όπου "βάρους" τώρα είναι ένας αριθμός που απεικονίζει το μέτρο του διανύσματος \vec{w}_i . Η αναλογία λοιπόν στην προσέγγιση του Pirkul ορίζεται ως: $p / \sum_{j=1}^k a_j w_j$, όπου k είναι ο αριθμός των διαστάσεων και a_j ο πολλαπλασιαστής για κάθε διάσταση j , αντιπροσωπεύοντας τη σημαντικότητα κάθε βάρους. Οι πολλαπλασιαστές a_j στην τρέχουσα περίπτωση επιλέχθηκαν να διαιρούν κάθε βάρους με το μέγιστο ποσό πόρου που παρέχει το σύστημα: $a_j = 1 / w_{\max,j}$, ώστε να λαμβάνονται ισοδύναμα ποσοστά για κάθε διάσταση, αφού κάθε απαίτηση πόρων έχει ίδια σημασία στον υπολογισμό προτεραιοτήτων.

Έτσι, ο παραπάνω λόγος είναι ίσος με $p / \sum_{l=1}^k (w_l / w_{\max,l})$ και η συνολική συνάρτηση (βλ. Εξίσωση 7) γίνεται τώρα:

$$profit / weight = \frac{PR - 2 \cdot VP \cdot VC - EC + VC}{\sum_{l=1}^k (w_l / w_{\max,l})}$$

Εξίσωση 8: Υπολογισμός πολυδιάστατης αναλογίας κέρδους / βάρους για μια εργασία

Λαμβάνοντας υπόψη την ανωτέρω μεταβολή για τη greedy λύση με πολλαπλές διαστάσεις, το αναχθέν πρόβλημα και η λύση του μπορούν να ορισθούν ως εξής:

«Έχοντας ένα σύνολο k διαθέσιμων πόρων $M = \{w_{\max,1}, w_{\max,2}, \dots, w_{\max,k}\}$, κάθε εισερχόμενη εργασία j_i θα έχει ένα σύνολο απαιτήσεων $w_i = \{w_{i1}, w_{i2}, \dots, w_{ik}\}$ για τους παρεχόμενους πόρους αποφέροντας κέρδος $p_i = PR_i - 2 \cdot VP_i \cdot VC_i - EC_i + VC_i, i \in \{1 \dots R\}$ »

Για R εισερχόμενες εργασίες το πρόβλημα διαχείρισης των πόρων ανάγεται όπως εξηγήθηκε σε ένα πολυδιάστατο κλασματικό πρόβλημα του σακιδίου και επιλύεται με την ταξινόμηση του πίνακα προτεραιοτήτων A , όπως ορίζουμε τον πίνακα που περιέχει τις αναλογίες κερδών / βάρη, ο οποίος με βάση την Εξίσωση 8 είναι:

$$A = \left[\frac{PR_1 - 2 \cdot VP_1 \cdot VC_1 - EC_1 + VC_1}{\sum_{j=1}^k (w_{1j} / w_{\max,j})}, \frac{PR_2 - 2 \cdot VP_2 \cdot VC_2 - EC_2 + VC_2}{\sum_{j=1}^k (w_{2j} / w_{\max,j})}, \dots, \frac{PR_R - 2 \cdot VP_R \cdot VC_R - EC_R + VC_R}{\sum_{j=1}^k (w_{Rj} / w_{\max,j})} \right]$$

Εξίσωση 9: Ταξινομημένος πίνακας προτεραιοτήτων με τις αναλογίες κερδών / βάρη

Σημειωτέον ότι μετά το βήμα 4 του γενικού Αλγορίθμου ΙΙΙ που ορίστηκε παραπάνω, η εργασία που έχει επιλεγεί μετά από τον επαναυπολογισμό των κερδών για να εκτελεστεί, μπορεί να μη "γεμίζει" το σακίδιο. Επομένως, επιπλέον πόροι μπορούν να είναι ακόμα διαθέσιμοι για να τρέξουν και άλλες εργασίες, κι έτσι το βήμα 4 πρέπει να επαναληφθεί.

Ο προκύπτων τελικός αλγόριθμος (Αλγόριθμος ΙV) που είναι η λύση του πλήρους προβλήματος του κλασματικού πολυδιάστατου σακιδίου, στο οποίο και αναγάζεται το πρόβλημα ορίζεται ως εξής:

Για το σύνολο των εργασιών $J = \{j_1, j_2, \dots, j_R\}$ ακολούθησε τα βήματα:

1. Ταξινόμησε τον πίνακα $\langle A \rangle$ που δίνεται στην Εξίσωση 9
2. Δέσμευσε τους πόρους που ζητά η πρώτη εργασία και αφαιρέσε την από τον πίνακα.
3. Επανάλαβε το βήμα 2 μέχρι οι απαιτήσεις της επόμενης εργασίας να μην ικανοποιούνται από τους διαθέσιμους πόρους
4. Για τις εργασίες που απομένουν, αναδιάταξε τον πίνακα μετά τον επαναυπολογισμό των p_i και διάλεξε αυτή (j_i) με τη μεγαλύτερη αναλογία $\frac{P_i}{\sum_{j=1}^k (w_{ij} / w_{\max,j})}$
5. Επανάλαβε το βήμα 4 έως ότου ένας από τους υπόλοιπους διαθέσιμους πόρους εξαντληθεί, στοχεύοντας στην πλήρη εκμετάλλευσή τους.

Μετά το βήμα 5, ένας από τους διαθέσιμους πόρους έχει εξαντληθεί, οπότε τη χρονική στιγμή εκείνη η μέγιστη δυνατή χρήση των πόρων λαμβάνει χώρα. Καμία άλλη εργασία δεν μπορεί να εκτελεστεί, μέχρι μία από τις εκτελούμενες εργασίες ολοκληρωθεί και απελευθερώσει πόρους, ή μια νέα εργασία εμφανιστεί με υψηλότερη προτεραιότητα.

6.3

Αρχιτεκτονική υλοποίησης και αξιολόγησης

6.3.1 Επίπεδο διαχείρισης πόρων

Ο προτεινόμενος αλγόριθμος διαχείρισης πόρων υλοποιήθηκε ως μια λύση μεταδρομολόγησης σε ένα cluster, πάνω από το οποίο ένα λειτουργεί ένα μεσολογισμικό GRIA. Ωστόσο, στο επίπεδο διαχείρισης πόρων απαιτείται ένα πλήρως εικονικοποιημένο και ευέλικτο περιβάλλον, που να είναι σε θέση να διακόψει μια εργασία που εκτελείται ανά πάσα στιγμή (όταν συντρέχουν οι συνθήκες "ενεργοποίησης" του μηχανισμού που αναφέρθηκαν). Έτσι, αποφασίστηκε η χρήση εικονικών μηχανών (Virtual Machines - VMs) στο επίπεδο διαχείρισης πόρων και πιο συγκεκριμένα Xen VMs, ώστε να φιλοξενήσουν τα περιβάλλοντα εκτέλεσης των εργασιών. Οι εικονικές μηχανές Xen είναι πλήρως ελαστικές, αφού οι πόροι (μνήμη, επεξεργαστές κ.λπ.) που χρησιμοποιούν μπορούν να αλλάξουν δυναμικά κατά το χρόνο εκτέλεσης. Επιπλέον, τα στιγμιότυπά τους είναι δυνατό να παγώσουν και να συνεχισθούν οποτεδήποτε απαιτείται, αποθηκεύοντας τοπικά την στιγμιαία εικόνα του υπολογιστικού περιβάλλοντος. Ο χρόνος που απαιτείται για την παύση / επανενεργοποίηση των μηχανών Xen επιφέρει επιπλέον υπολογιστικό κόστος κατά τον χρονοπρογραμματισμό των εργασιών που μπορεί ενδεχομένως να επηρεάσει την απόδοση αλγορίθμου. Ωστόσο, όπως παρουσιάζεται στην ενότητα της αξιολόγησης των αποτελεσμάτων, η συνολική επίδραση της δυναμικής αναδιαμόρφωσης των Xen VMs δεν είναι κρίσιμη στην απόδοση του συστήματος.

Ως εκ τούτου, ένας μηχανισμός μετα-δρομολόγησης χτίστηκε στη βάση του μεσολογισμικού GRIA middleware, διαχειριζόμενο μια υπολογιστική συστάδα με χρήση Xen Virtual Machines. Ο μηχανισμός αυτός υλοποιεί μια άπληστη λύση του πολυδιάστατου κλασματικού προβλήματος του σακιδίου, έτσι ώστε να δίδεται η ανάλογη προτεραιότητα στις εκκρεμείς εργασίες σύμφωνα με το Αλγόριθμο IV, κάθε φορά που μια νέα εργασία φτάνει στο επίπεδο μετα-δρομολόγησης ή μια τερματιζόμενη εργασία αποδεσμεύει πόρους. Η αξιοποίηση των διαθέσιμων πόρων είναι σχεδόν βέλτιστη, ακόμη και στην περίπτωση που οι πόροι δεν καλύπτουν πλήρως τον απαιτούμενο φόρτο εργασίας, αφού το "σακίδιο" είναι συνεχώς όσο γεμάτο είναι δυνατόν. Με αυτό τον τρόπο, όταν οι απαιτήσεις υπερβαίνουν τους διαθέσιμους πόρους, τουλάχιστον μία ή και περισσότερες από τις "διαστάσεις" των πόρων θα αξιοποιηθούν πλήρως. Επιπροσθέτως, τα έσοδα του παρόχου μεγιστοποιούνται με τη διαχείριση του κινδύνου αστοχιών, έναντι των παραδοσιακών μεθόδων χρονοπρογραμματισμού και διαχείρισης των πόρων, όπως για παράδειγμα η τεχνική "First Come First Served". Το σχήμα αυτό διαχείρισης ρίσκου βασίζεται στην πιθανότητα παραβίασης του SLA, που συμβαίνει είτε εκ προθέσεως, λόγω μη παροχής των αναγκαίων πόρων, είτε ακούσια, π.χ. λόγω τυχαίας αστοχίας.

6.3.2 Υπηρεσία Ιστορικού και Εκτίμησης Παραβιάσεων

Οι πιθανότητες παραβίασης SLA που λαμβάνονται παραπάνω, εξάγονται από την Υπηρεσία Παροχής Πληροφορίας QoS (QoS Information Provisioning Service - QoS-IPS), που παρήγαγαν οι Tserpes, Kyriazis, Menychtas, & Varvarigou (2008). Η υπηρεσία QoS-IPS βασίζεται σε μια αρχιτεκτονική που ακολουθεί τις προδιαγραφές του Web Services Resource Framework και επομένως είναι μία σύνθεση από διαδικτυακές υπηρεσίες που διαχειρίζονται καταστάσεις πόρων. Η λειτουργία της είναι βασισμένη στη

χρησιμοποίηση των ιστορικών δεδομένων. Η αποθήκευση των τιμών των μετρικών επιτρέπει την πιο σύνθετη ανάλυση για τις αναφορές σε μια πιο πρόσφατη φάση ή μια σύγκριση των πραγματικών δεδομένων με τα ιστορικά συλλεχθέντα στοιχεία.

Η υπηρεσία αποτελείται από δύο θεμελιώδη συστατικά (Tserpes, 2007): την **Υπηρεσία Ιστορικού Ποιότητας** (QoS History Service) και την **Υπηρεσία Εκτίμησης Δεικτών Ποιότητας** (QoS Indices Estimator Service), αλλά αλληλεπιδρά ή μπορεί να αλληλεπιδράσει με πολλά άλλα. Μια βασική αλληλεπίδραση είναι με την υπηρεσία διαχείρισης SLA (SLA Manager) που λειτουργεί ως ενδιάμεσος για τη μεταφορά των QoS απαιτήσεων του καταναλωτή υπηρεσιών όταν τις λαμβάνει στην πρώτη φάση της διαπραγμάτευσης SLA και χρησιμοποιεί την QoS-IPS για αποθήκευση των ορίων που έθεσε ο καταναλωτής σε κάποιο όρο ποιότητας ο οποίος παραβιάστηκε κατά τη διάρκεια ζωής του SLA μέσω του QoS History Service. Από την άλλη, το QoS Indices Estimator Service υπολογίζει την πιθανότητα ο πάροχος να παραδώσει ένα συγκεκριμένο επίπεδο QoS στον καταναλωτή. Ο υπολογισμός των πιθανοτήτων βασίζεται στους διαθέσιμους πόρους και στο ιστορικό παραβιάσεων SLA που κατά το παρελθόν στην εκτέλεση εργασιών με παρόμοια χαρακτηριστικά. Κατά τα αρχικά στάδια χρήσης της υπηρεσίας, το ιστορικό θα είναι κενό, ενώ με την παρέλευση ενός εύλογου χρονικού διαστήματος "κατάρτισης" της υπηρεσίας, θα είναι πιο αξιοποιήσιμο και μεγαλύτερης σημασίας.

6.3.3 Σχήμα διαχείρισης πόρων στο μεσολογισμικό GRIA

Η αρχιτεκτονική του GRIA (Παράγραφος 2.1.6.1), καθώς και ο τρόπος που διενεργείται μια διαπραγμάτευση από την Υπηρεσία SLA (Παράγραφος 5.3.3) και παρακολουθείται η χρησιμοποίηση πόρων από τους πελάτες του, έχουν αναλυθεί ήδη σε προηγούμενες ενότητες και θα αποφευχθεί εδώ η επανάληψή τους. Θα εστιάσουμε κυρίως στη

διαχείριση των πόρων ενός cluster από την αντίστοιχη υπηρεσία του μεσολογισμικού (GRIA Job Service), η οποία είναι αυτή που διεκπεραιώνει την υποβολή των εισερχόμενων εργασιών σε αυτό, επιτρέποντας τη χρήση διαφόρων διαχειριστών πόρων (Resource Managers). Για παράδειγμα δίνεται η δυνατότητα ενσωμάτωσης του χρονοπρογραμματιστή TorquePBS (Torque Resource Manager, 2010) ή του Condor (Condor High Throughput Computing, 2012), μέσω ενός απλού μηχανισμού επιλογής.

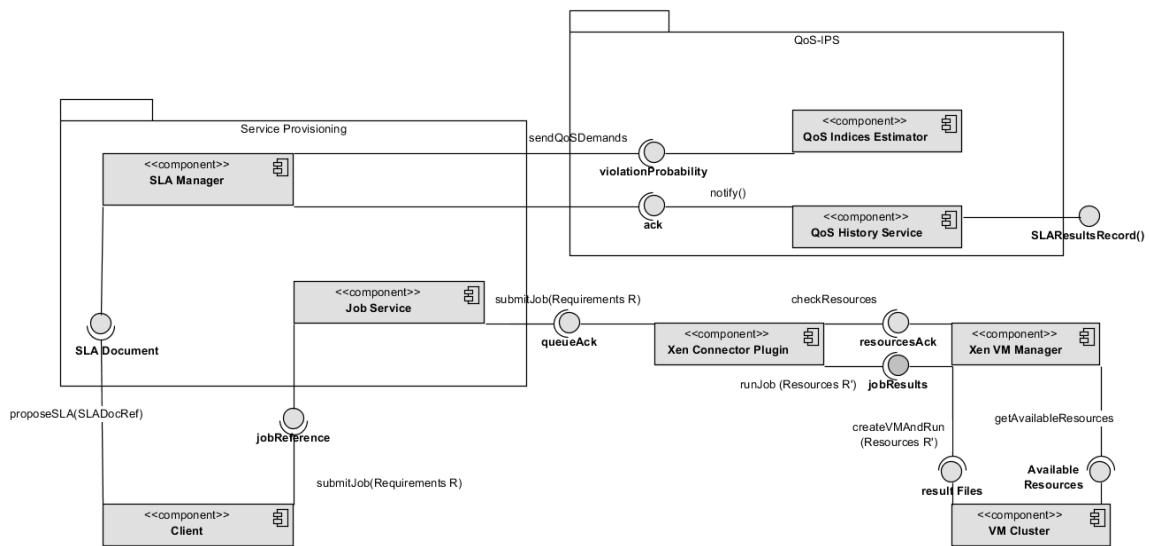
Η υπηρεσία Job Service στην πραγματικότητα δεν καλεί τους διαχειριστές πόρων άμεσα για να υποβάλει και να ελέγχει τις καταστάσεις των εκτελούμενων εργασιών. Αντ' αυτού, εισάγει ένα επιπλέον επίπεδο από RM Connector scripts, γραμμένα στη γλώσσα python: για κάθε Resource Manager, το GRIA απαιτεί ένα ξεχωριστό RM Connector Plugin. Το προεπιλεγμένο RM Connector Plugin επιτελεί τοπική εκτέλεση εργασιών, στο μηχάνημα που είναι εγκατεστημένο το μεσολογισμικό και δε χρησιμοποιεί cluster. Στο παραδοσιακό μοντέλο του GRIA, μια εργασία μπορεί να γίνει δεκτή για άμεση υποβολή ή να απορριφθεί. Δεν υπάρχει κάποιος ενδιάμεσος μηχανισμός ή ουρά ώστε να περιμένουν οι εκκρεμείς εργασίες, εφόσον οι πόροι δεν είναι διαθέσιμοι. Ως εκ τούτου, αν ο διαχειριστής πόρων δεν δέχεται περισσότερες εργασίες από αυτές που ήδη εκτελούνται, η υποβολή τους αποτυγχάνει, παραβιάζοντας πιθανόν τα υφιστάμενα SLAs.

6.3.4 Υλοποίηση νέου διαχειριστή πόρων

Προκειμένου να εφαρμοστεί ο μηχανισμός που περιγράφεται παραπάνω, προστέθηκε ένα Xen Connector Plugin που υλοποιεί μια υποτυπώδη ουρά αναμονής και εκτελεί έναν αλγόριθμο μετα-δρομολόγησης, με βάση το κλασματικό μοντέλο του σακιδίου που παρουσιάστηκε. Το νέο Plugin γράφτηκε σε γλώσσα python και υλοποιεί τις τέσσερις βασικές λειτουργίες που απαιτούνται για κάθε RM Connector Plugin του GRIA:

- **canRunJob**: Μέθοδος που ελέγχει αν μια εργασία μπορεί να εκτελεστεί
- **submitJob**: Μέθοδος υποβολής της εργασίας
- **checkJob**: Μέθοδος που ελέγχει αν η εκτέλεση έχει τερματιστεί
- **jobUsage**: Μέθοδος που ελέγχει τη χρήση των πόρων από την εργασία
- **killJob**: Μέθοδος που παύει την εκτέλεση μιας εργασίας

Η τροποποίηση αφορούσε κυρίως τις μεθόδους "canRunJob" και "submitJob", με την τελευταία να υλοποιεί τον αλγόριθμο επίλυσης του προβλήματος του σακιδίου, ιεραρχώντας την ουρά αναμονής με βάση τον Αλγόριθμο IV. Ο αλγόριθμος εκτελείται όταν μια νέα εργασία υποβάλλεται στο GRIA Job Service ή νέοι πόροι διατίθενται από τον τερματισμό εκτέλεσης μιας εργασίας. Το Plugin συνδέεται με μια μονάδα διαχείρισης εικονικών μηχανών Xen (Xen VM Manager), η οποία αποτελείται από ένα σύνολο linux shell scripts, ελέγχοντας όλες τις εικονικές μηχανές στο cluster.



Σχήμα 41: Διεπαφές υπηρεσίας ιστορικού και Resource Manager Plugin

Όταν μια νέα εργασία i υποβάλλεται (βλ. και Σχήμα 42 παρακάτω), η μονάδα Xen VM Manager ελέγχει τους διαθέσιμους πόρους του cluster μέσω του script "getAvailableResources" και επιστρέφει το αποτέλεσμα στο Plugin μετα-δρομολογητή. Ο μετα-δρομολογητής ελέγχει το διάνυσμα των απαιτήσεων R ($r1$ CPUs, $r2$ MB μνήμης, $r3$ MB δίσκου) της εργασίας από τα έγγραφα Περιγραφής Εργασίας που τη συνοδεύουν (ένα παράδειγμα δίνεται στον Πίνακα 15), καθώς και τα αντίστοιχα κόστη από το SLA του πελάτη για να δώσει προτεραιότητα στις εκκρεμείς εργασίες. Για να υπολογιστεί το συνολικό "κέρδος", θα πρέπει να εκτιμηθεί επίσης η πιθανότητα παραβίασης SLA. Για να βρεθεί αυτή η πιθανότητα, καλείται η Υπηρεσία QoS Ιστορικού, η οποία αποθηκεύει το ιστορικό όλων παραβάσεων και μπορεί να παρέχει μια εκτίμηση για την πιθανότητα παραβίασης του SLA κάθε εργασίας i , με δεδομένους τους πόρους.

```
<?xml version="1.0" encoding="UTF-8"?>
<JobDefinition xmlns="http://schemas.ggf.org/jSDL/2005/11/jSDL">
  <JobDescription>
    <JobIdentification>
      <JobName>render 1</JobName>
    </JobIdentification>
    <Application>
      <ApplicationName>render</ApplicationName>
      <POSIXApplication xmlns="http://schemas.ggf.org/jSDL/2005/11/jSDL-posix">
        <Executable>/home/ntua/griaTestApp/render/startJob.py</Executable>
      </POSIXApplication>
    </Application>
    <Resources>
      <TotalCPUCount>
        <Range>
          <LowerBound>0.0000000000</LowerBound>
          <UpperBound>2.0000000000</UpperBound>
        </Range>
      </TotalCPUCount>
      <TotalVirtualMemory>
        <Range>
          <LowerBound>0.0000000000</LowerBound>
          <UpperBound>200000000.0000000000</UpperBound>
        </Range>
      </TotalVirtualMemory>
      <TotalDiskSpace>
        <Range>
          <LowerBound>0.0000000000</LowerBound>
          <UpperBound>20000000.0000000000</UpperBound>
        </Range>
      </TotalDiskSpace>
    </Resources>
    <DataStaging name="inputImage">
```



```

<FileName>inputImage</FileName>
<CreationFlag>overwrite</CreationFlag>
<DeleteOnTermination>>true</DeleteOnTermination>
</DataStaging>
<DataStaging name="outputImage">
<FileName>outputImage</FileName>
<CreationFlag>overwrite</CreationFlag>
<DeleteOnTermination>>true</DeleteOnTermination>
</DataStaging>
</JobDescription>
</JobDefinition>

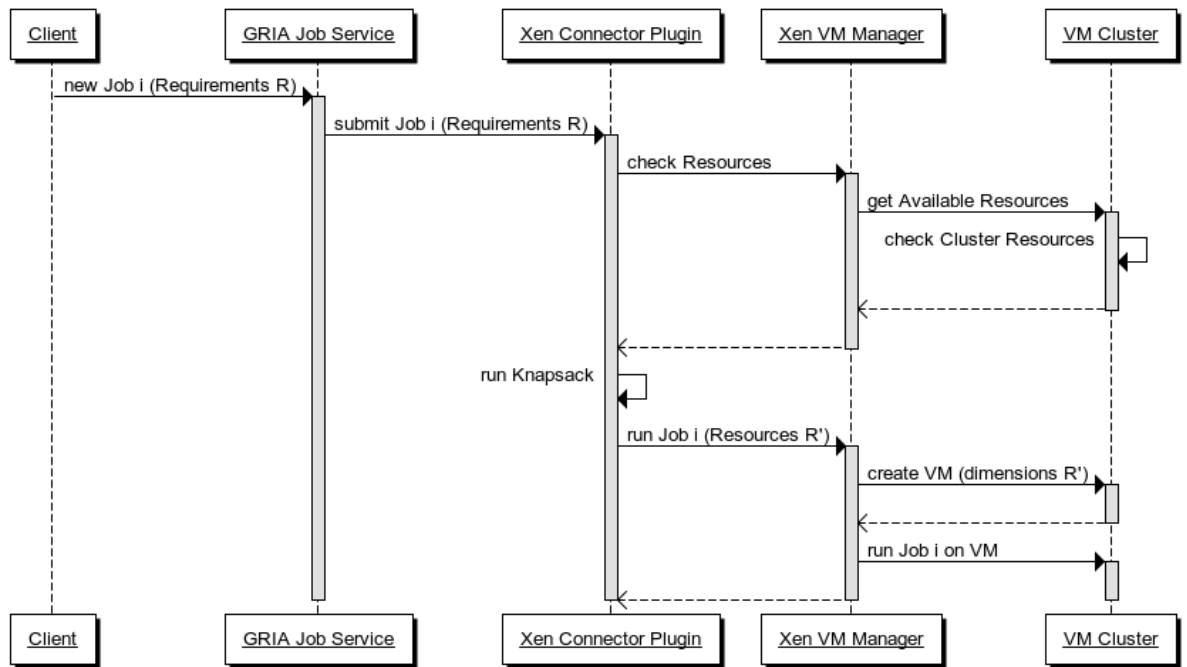
```

Πίνακας 15: Ένα παράδειγμα JSDL εγγράφου Περιγραφής Εργασίας του GRIA

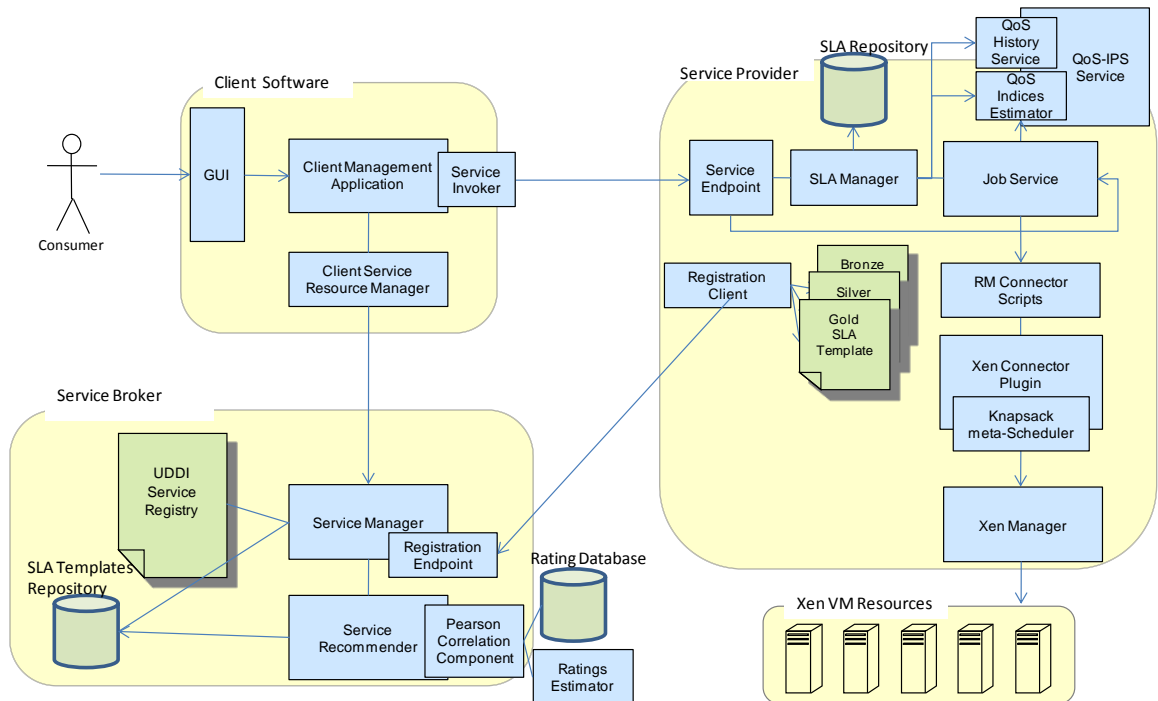
Μετά την εφαρμογή του αλγορίθμου, οι εργασίες ταξινομούνται κατά σειρά προτεραιότητας και αποστέλλονται στο μονάδα Xen VM Manager για να εκτελεστούν αναλόγως. Ο Xen VM Manager καθορίζει τις διαστάσεις R' ($r1'$ CPUs, $r2'$ MB μνήμης, $r3'$ MB δίσκου) των εικονικών μηχανών που εκτελούνται, κατανέμοντας τους πόρους με αυτόν τον τρόπο για κάθε εργασία μέσω διαμόρφωσης της εικονικής μηχανής Xen που φιλοξενεί την εφαρμογή η οποία θα τρέξει.

Κλασματική (όταν το R' δεν είναι ίσο με R) ή και πλήρης δέσμευση πόρων μπορεί έτσι να επιβληθεί, με τον καθορισμό των διαστάσεων R' στο αντίστοιχο Virtual Machine, κατά την υποβολή της εργασίας. Κάθε μία από τις εικονικές μηχανές Xen χρησιμοποιεί μια εγκατάσταση του wrapper της εφαρμογής, το οποίο είναι ένα ουσιαστικά ένα script που εκτελεί την απαιτούμενη εφαρμογή, ενώ τα δεδομένα εισόδου και εξόδου αυτόματα μεταφέρονται από και προς το περιβάλλον του VM Manager μέσω Secure Shell (SSH) σύνδεσης.

Η όλη ροή των δεδομένων που περιγράφηκε απεικονίζεται αναλυτικά παρακάτω στο Σχήμα 42:



Σχήμα 42: Ροή δεδομένων κατά την υποβολή μιας νέας εργασίας



Σχήμα 43: Συνολική απεικόνιση αρχιτεκτονικής αξιολόγησης από τον πελάτη και τον πάροχο με όλα τα εμπλεκόμενα στοιχεία

Το Σχήμα 43 παρουσιάζει μια συνολική επισκόπηση όλων των στοιχείων του συστήματος που εμπλέκονται στην αξιολόγηση και διαχείριση πόρων, συμπεριλαμβάνοντας και το λογισμικό αξιολόγησης από το μέρος των πελατών, που παρουσιάστηκε στο προηγούμενο κεφάλαιο για λόγους πληρότητας.

6.4

Πειράματα

6.4.1 Μέθοδος Αξιολόγησης

Για την αξιολόγηση του ανωτέρω μηχανισμού, η απόδοση του μεσολογισμικού GRIA με ενσωματωμένα τα στοιχεία της αρχιτεκτονικής αναφοράς, που υλοποιούν τον αλγόριθμο του σακιδίου, πρέπει να συγκριθεί με υφιστάμενες αρχιτεκτονικές και μεθόδους. Αρκετές πολιτικές χρονοδρομολόγησης μελετήθηκαν (Mor-Harchol Balter, 2002), όπως το Shortest-Remaining-Processing-Time-First (SRPT), Longest-Remaining-Processing-Time-First (LRPT), Round-Robin, ή το πιο σύνηθες First-Come, First-Served (FCFS) σενάριο. Τα περισσότερα κρίθηκαν ακατάλληλα λόγω της μεγάλης επιβάρυνσης επαναδρομολόγησης. Ο μηχανισμός μετα-δρομολόγησης συγκρίνεται με κάποια από αυτά τα εναλλακτικά σενάρια για να επισημανθεί η αποτελεσματικότητά του.

Η πειραματική διάταξη στο επίπεδο πόρων, αποτελείται από ένα Xen Virtual Machine cluster, που χρησιμοποιεί τρεις (3) διαφορετικούς υπολογιστές (τον GRIA server και άλλους 2 διασυνδεδεμένους) με περιορισμένους πόρους, για τη δημιουργία συνθηκών υπερφόρτωσης. Κάθε εικονική μηχανή που τρέχει λειτουργεί σε σύστημα Ubuntu 9.04 και έχει εγκατεστημένη μια εφαρμογή 3D Rendering, που βασίζεται σε μια διανομή του 3Delight renderer. Ο πάροχος GRIA προσφέρει την ανωτέρω εφαρμογή ως υπηρεσία, συνενώνοντας ένα αριθμό ενδιάμεσων στοιχειωδών υπηρεσιών για τη

συλλογή των δεδομένων, τον έλεγχο της εγκυρότητας, την κλήση της εφαρμογής και την παράδοση του αποτελέσματος.

Οι εργασίες προς υποβολή επιλέγονται ανάλογα με την πολυπλοκότητά τους, που επηρεάζεται κυρίως από τους ακόλουθους παράγοντες:

- Αριθμός αντικειμένων.
- Φωτισμός (τοπικός, σημειακός, άμεσος).
- Υλικά (πρότυπα, εικόνες, κινούμενα σχέδια).
- Τροποποιητές (πλέγμα, συμπαγές υλικό).

Έτσι, δημιουργούνται σκηνές προς απόδοση με βάση όλες τις δυνατές επιλογές για την πολυπλοκότητα, ώστε να δοκιμαστεί και να συγκριθεί η απόδοση του παρόχου σε διαφορετικά σχήματα μετα-δρομολόγησης και διαχείρισης πόρων.

6.4.2 Πειραματικά δεδομένα

Δοθείσας της παραπάνω πειραματικής διάταξης και των παραμέτρων 3D απόδοσης, δημιουργήθηκαν 30 εργασίες 3D Rendering προς υποβολή στην υπηρεσία του renderer, με κάθε μία να αποτελείται από 50 αρχεία RIB, που προέρχονται από 30 αντίστοιχα animations που διαρκούν μερικά δευτερόλεπτα και είχαν διαφορετικές ρυθμίσεις, ή – με άλλα λόγια – διαφορετικό συνδυασμό ενός αριθμού αντικειμένων, φωτισμού, υλικών και τροποποιητών. Σε κάθε περίπτωση όμως, παράγουν έναν αξιοσημείωτο φόρτο εργασίας, ώστε να δημιουργήσουν συνθήκες έλλειψης πόρων για την αξιολόγηση του προτεινόμενου μηχανισμού. Οι εργασίες αυτές, μαζί με τις απαιτήσεις στα SLA και μετρημένα όρια των ελεγχόμενων παραμέτρων ποιότητας, παρουσιάζονται στον Πίνακα 16 της επόμενης σελίδας.

A/A	Job Description (50rib files / 2 sec animation)	SLA Threshold CPU (sec)	Benchmarked CPU (sec)	Reserved Disk Capacity (MB)	Benchmarked Disk Utilization (MB)	Requested Minimum Memory (MB)	Benchmarked Minimum Memory Utilization (MB)
1	Ob-01	2,22	2,08	19,05	17,06	128	8,40
2	Ob-01-light-s	6,25	6,54	19,10	17,54	128	9,10
3	Ob-01-light-o	4,51	4,46	19,10	15,71	128	9,10
4	Ob-01-light-d	5,38	5,54	19,05	17,24	128	9,60
5	Ob-01-material-s	2,22	2,02	19,05	16,99	128	9,60
6	Ob-01-light-s- material-s	4,50	4,00	19,10	15,41	128	10,00
7	Ob-01-light-o- material-s	4,58	4,27	19,10	22,72	128	9,80
8	Ob-01-light-d- material-s	5,49	5,24	19,10	22,46	128	10,00
9	Ob-01-modifier- lat	2,88	2,79	19,10	19,34	128	10,50
10	Ob-01-light-s- modifier-lat	136,40	159,40	141,55	167,62	128	18,70
11	Ob-01-light-o- modifier-lat	64,24	58,67	141,75	124,73	128	19,60
12	Ob-01-light-d- modifier-lat	112,47	83,80	142,30	169,77	128	19,60
13	Ob-01-material- image	5,14	4,76	19,30	19,72	128	9,30
14	Ob-01-light-s- material-image	9,10	8,81	19,30	20,23	128	8,70
15	Ob-01-light-o- material-image	7,33	7,13	19,30	16,70	128	8,90
16	Ob-01-light-d- material-image	8,37	7,92	19,30	17,42	128	9,80
17	Ob-01-material- an	24,24	24,59	19,30	16,75	128	10,90
18	Ob-01-light-s- material-an	33,72	32,57	159,30	167,92	128	21,40
19	Ob-01-light-o- material-an	31,35	28,29	158,95	178,94	128	22,60
20	Ob-01-light-d- material-an	32,05	29,31	159,15	148,48	128	22,80
21	Ob-01-light-s- modifier-lat- material-an	186,10	102,30	286,55	331,02	128	35,60
22	Ob-01-light-o- modifier-lat- material-an	86,18	80,25	204,05	193,30	128	1,60
23	Ob-01-light-d- modifier-lat- material-an	112,40	115,94	201,85	241,20	128	35,80
24	Ob-02	2,29	2,49	26,55	23,33	128	13,90
25	Ob-02-light-s	8,37	7,76	26,60	26,75	128	12,80
26	Ob-02-light-o	5,38	5,09	26,60	23,97	128	13,50
27	Ob-02-light-d	6,74	7,27	26,60	24,19	128	13,70
28	Ob-02-material-s	2,53	2,60	26,60	28,85	128	13,70
29	Ob-02-light-s- material-s	8,61	8,83	26,65	29,54	128	14,10
30	Ob-02-light-o- material-s	5,52	5,84	26,65	24,76	128	13,70

Πίνακας 16: Εργασίες 3D Rendering υπο εκτέλεση. Τα ονόματα των εργασιών δείχνουν τη διαμόρφωσή των αρχείων (αριθμός αντικειμένων, φώτων, κλπ.).

Προκειμένου να καθοριστούν τα σχετικά όρια που φαίνονται στον παρακάτω πίνακα, μερικές εμπειρικές μέθοδοι χρησιμοποιήθηκαν που έχουν αποδειχθεί πολύ ακριβείς για εργασίες που υπέβαλε ο τελικός χρήστης στο παρελθόν. Έτσι λοιπόν, ο απαιτούμενος χρόνος CPU υπολογίστηκε με βάση τις προηγούμενες εκτελέσεις εργασιών με παρόμοια χαρακτηριστικά. Η απαιτούμενη χωρητικότητα δίσκων είναι το σύνολο του μεγέθους του αρχείου εισαγωγής, συν το μέγεθος του τελικού αρχείου (αυτό είναι γνωστό αφού εξαρτάται από την ανάλυση), συν το μέγεθος των σχετικών shaders. Η απαιτούμενη μνήμη τέθηκε ως «χαλαρό» όριο, δεδομένου ότι δεν υπήρξε κανένα παράδειγμα οποιασδήποτε εργασίας σε αντίστοιχα μεγέθη που απαιτεί περισσότερη μνήμη από 128 MB. Στα όρια αυτά, όπως και στις μετρήσεις που θα ακολουθήσουν, δεν συμπεριλαμβάνονται οι πόροι για να τρέξει το λειτουργικό σύστημα του περιβάλλοντος εκτέλεσης που είναι αμελητέοι σε σχέση με τους διατιθέμενους για εκτέλεση εφαρμογών.

Όσον αφορά το χρόνο μεταξύ των αφίξεων δύο εργασιών, επιλέχθηκε να ακολουθήσει μια απλή κατανομή Gaussian και συνεπώς η ουρά του συστήματος μπορεί να μοντελοποιηθεί ως μια αλυσίδα Markov, δεδομένου ότι ισχύει η ιδιότητα Markovian (Markovian property). Η ιδιότητα αυτή απαιτεί η πιθανότητα κάθε πιθανής μελλοντικής κατάστασης των εργασιών στην ουρά X_{n+1} (θεωρούμε σχηματικά ότι κατά την εκτέλεση του αλγορίθμου μετα-δρομολόγησης όλες οι εργασίες μπαίνουν στην ουρά και επαναταξινομούνται) να εξαρτάται μόνο από την τρέχουσα κατάσταση της ουράς X_n :

$$\Pr(X_{n+1} = x \mid X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = \Pr(X_{n+1} = x \mid X_n = x_n)$$

Εξίσωση 10: Η Markovian ιδιότητα για την ουρά του μετα-δρομολογητή

Για να ελεγχθεί η απόδοση του αλγορίθμου σε σχέση με τις υπάρχουσες τακτικές μετα-δρομολόγησης, χρειάστηκαν διάφορες πολιτικές τιμολογήσεων και απαιτήσεων,

όσον αφορά το κόστος εργασίας, κόστος παραβίασης και τις προθεσμίες εκτέλεσης. Τα κόστη εργασίας και παραβίασης τέθηκαν με βάση την κανονική κατανομή, για τη δημιουργία των δεδομένων εισόδου. Όσον αφορά τη σχέση μεταξύ τους, τρεις κατηγορίες τιμολόγησης ορίστηκαν, όπως φαίνεται στον Πίνακα 17:

<i>Violation Cost (VC) Category</i>	SMALL	MEDIUM	LARGE
<i>Mean value in the VC distribution / Mean value in the Job Cost distribution</i>	2/7	3,5/7	5/7

Πίνακας 17: Κατηγορίες τιμολόγησης κόστους παραβίασης

Για την επικύρωση του μηχανισμού με διαφορετικές πολιτικές τιμολόγησης, τρεις επαναλήψεις της διαδικασίας αξιολόγησης επιχειρήθηκαν. Κατά τη διάρκεια της πρώτης επανάληψης τα κόστη των εισερχόμενων εργασιών ενέπιπταν στη μικρή κατηγορία, ενώ κατά τη διάρκεια της δεύτερης στη μεγάλη κατηγορία. Στην Τρίτη επανάληψη, τα SLAs των εισερχόμενων εργασιών χωρίστηκαν σε 3 ομάδες με μικρά, μεσαία και μεγάλα κόστη παραβίασης (μεικτή περίπτωση).

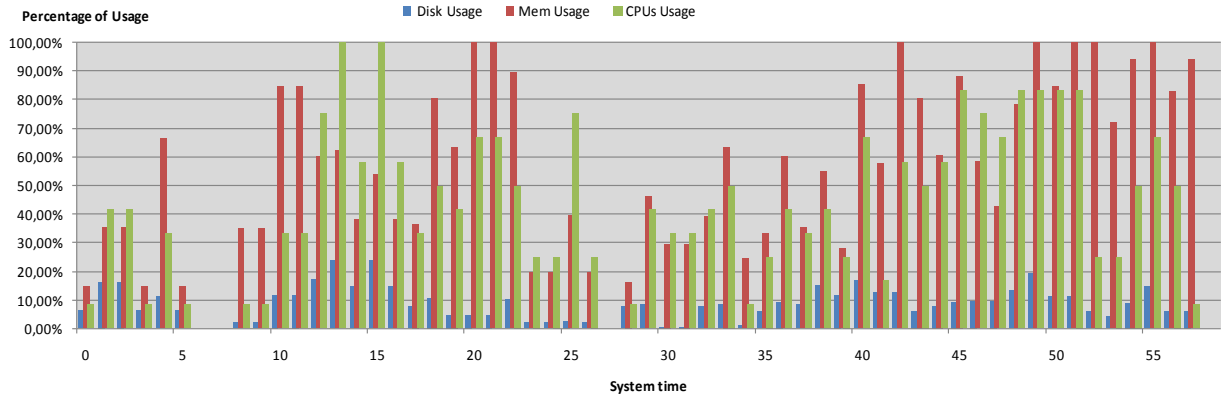
Τέλος, τα πειράματα έλαβαν αρχικά υπόψη 3 διαστάσεις, όσον αφορά τους απαιτούμενους πόρους που κατανέμονται με τον αλγόριθμο του σακιδίου: το χώρο στο δίσκο, τη μνήμη και τον αριθμό των επεξεργαστών (CPUs). Επιπρόσθετα, εφόσον σαν διάσταση θεωρήσαμε τα CPUs που απαιτούνται κάθε συγκεκριμένη στιγμή, για να γίνουν πιο κατανοητά τα πειράματα, στον αλγόριθμο μετα-δρομολόγησης συμπεριελήφθη και μια τέταρτη διάσταση για το χρόνο CPU που απομένει για την ολοκλήρωση μιας εργασίας, ώστε να ενσωματωθεί μια SRPT πολιτική στη χρονοδρομολόγηση: δεδομένου ότι οι εισερχόμενες εργασίες έχουν συγκεκριμένες προθεσμίες, ο χρόνος CPU που μένει για να ολοκληρωθεί μια εργασία είναι ένα άλλο

μέτρο του απαιτούμενου φόρτου εργασίας (εργασίες κοντά στην ολοκλήρωση πρέπει να είναι επίσης υψηλής προτεραιότητας, σε μια προσπάθεια να αποσυμφορηθεί η ουρά).

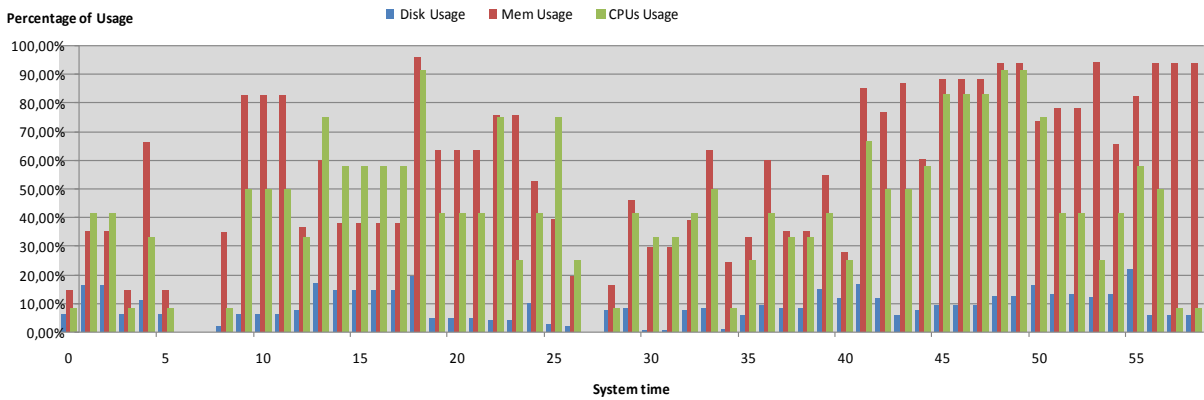
Ο φόρτος εργασίας λοιπόν, όπως αναφέρεται στην αρχή, είναι ευμετάβλητος: όταν ο χρόνος που υπολείπεται από την προθεσμία της εργασίας, για παράδειγμα, είναι μικρότερος από το υπόλοιπο του CPU χρόνου που ζητείται από την εργασία, ο αριθμός των επεξεργασιών που απαιτεί η εργασία τη συγκεκριμένη στιγμή πρέπει να αυξηθεί προκειμένου να καλύψει την προθεσμία, δεδομένου φυσικά ότι η εφαρμογή μπορεί να παραλληλοποιηθεί. Συνεπώς, ο απαιτούμενος φόρτος εργασίας για κάθε εργασία στην ουρά πρέπει να υπολογίζεται εκ νέου κάθε φορά που τρέχει ο αλγόριθμος σακιδίου.

6.4.3 Αποτελέσματα πειραμάτων

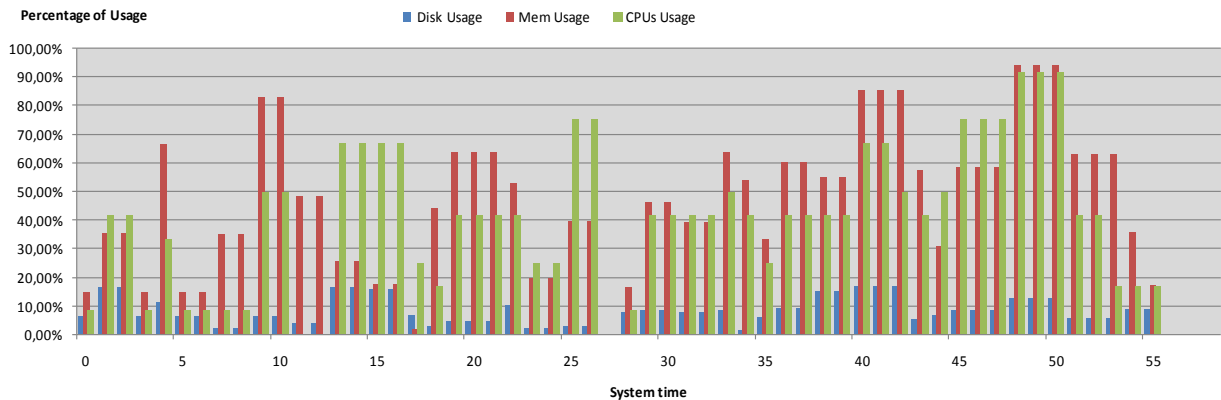
Όπως αναφέρθηκε, τα πειράματα χρησιμοποιούν 30 εργασίες 3D απόδοσης που εκτελούνται στο Xen VM cluster σε τρεις επαναλήψεις. Στα διαγράμματα που παρουσιάζονται παρακάτω (Σχήμα 44 –Σχήμα 52), απεικονίζεται η χρησιμοποίηση πόρων της υπολογιστικής συστάδας και το συνολικό κέρδος που αποκτάται από τον πάροχο, το οποίο μετράται σε νομισματικές μονάδες (υπολογιζόμενο από τις τιμές των εργασιών που εκτελέστηκαν επιτυχώς και το κόστος παραβίασης των υπολοίπων SLAs). Σε κάθε επανάληψη, το ίδιο σύνολο εργασιών υποβλήθηκε στο σύστημα που χρησιμοποιεί τη λύση κλασματικού σακιδίου, καθώς και στο πρωτότυπο σύστημα GRIA, συνδεδεμένο με το Xen VM cluster με έναν απλό μετα-δρομολογητή χωρίς την ύπαρξη ουράς, αλλά και στο GRIA με εφαρμογή FCFS πολιτικής μετα-δρομολόγησης στην ουρά του RM Plugin.



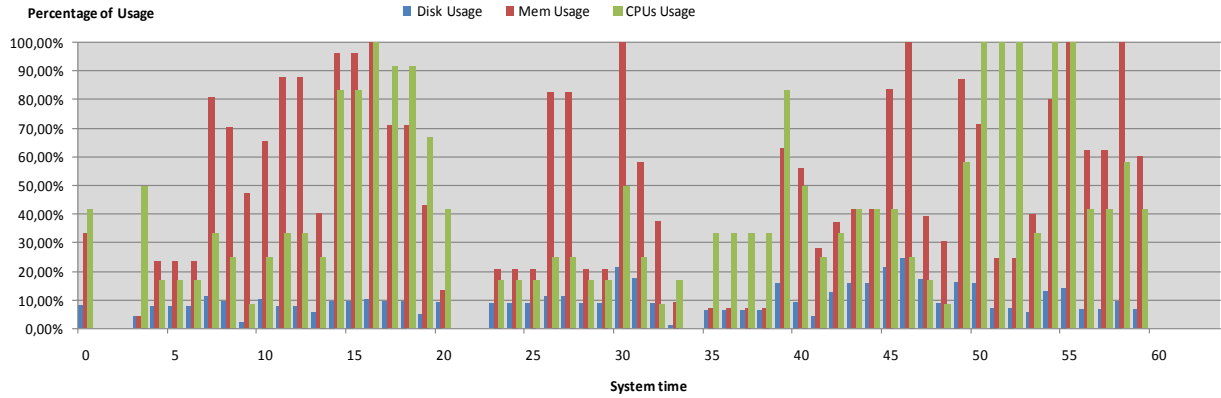
Σχήμα 44: Ποσοστιαία χρησιμοποίηση πόρων με το μετα-δρομολογητή σακιδίου, κέρδος = 16557 (1^η Επανάληψη)



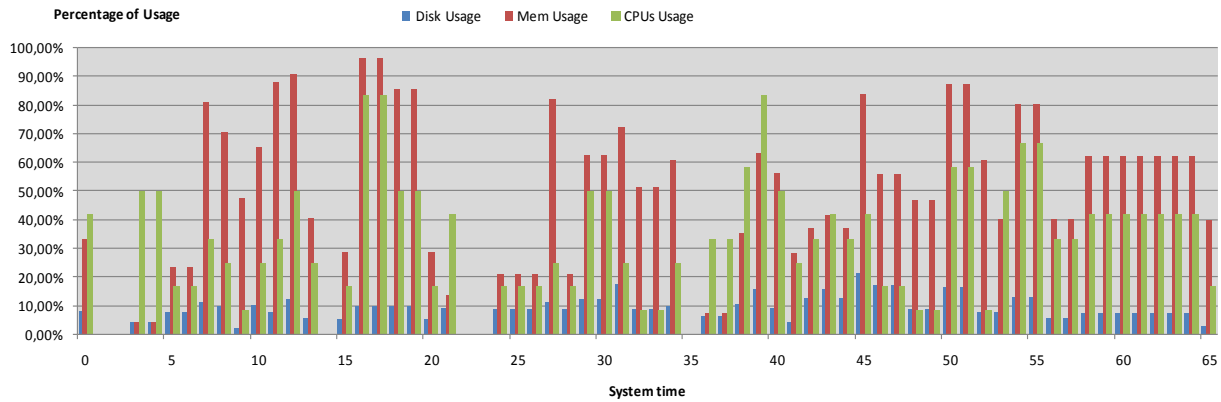
Σχήμα 45: Ποσοστιαία χρησιμοποίηση πόρων με υλοποίηση πολιτικής FCFS μετα-δρομολόγησης, κέρδος = 12540 (1^η Επανάληψη)



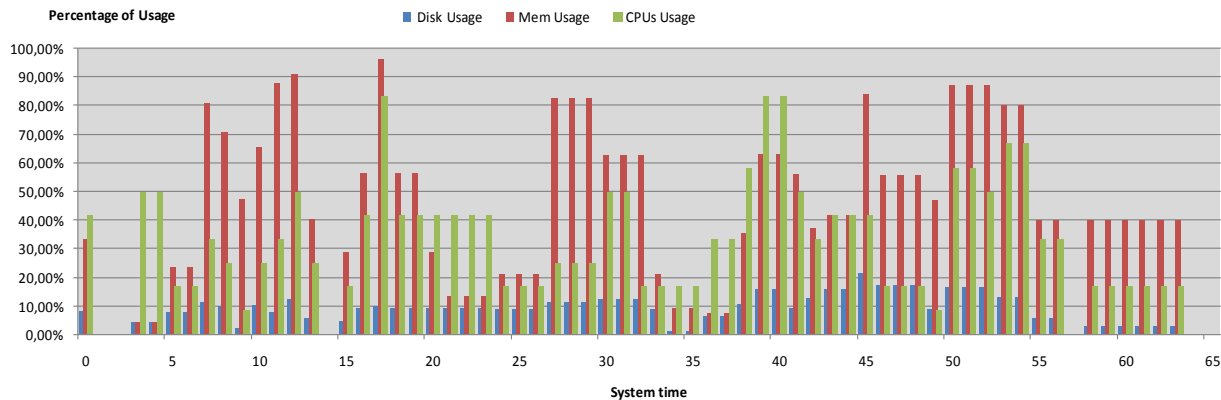
Σχήμα 46: Ποσοστιαία χρησιμοποίηση πόρων με τον απλό δρομολογητή του GRIA, κέρδος = 11670 (1^η Επανάληψη)



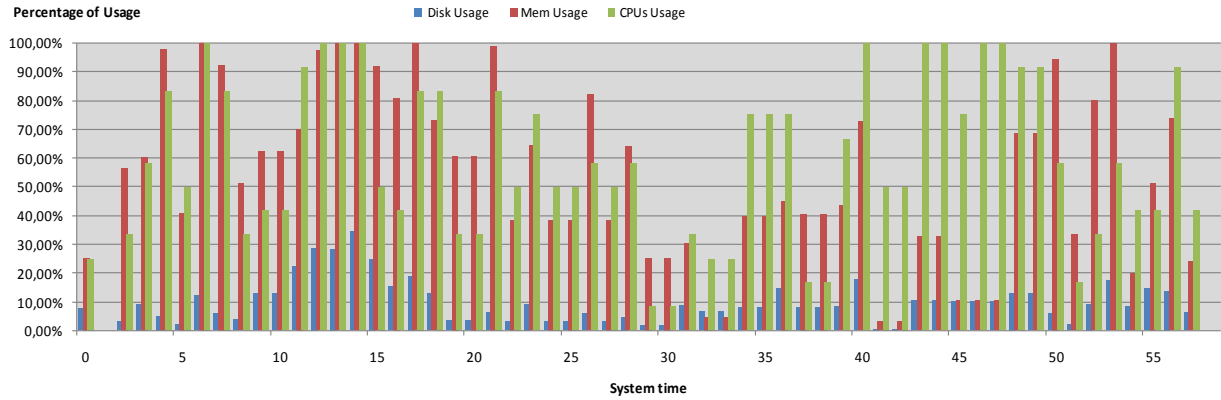
Σχήμα 47: Ποσοστιαία χρησιμοποίηση πόρων με το μετα-δρομολογητή σακιδίου, κέρδος = 16244 (2^η Επανάληψη)



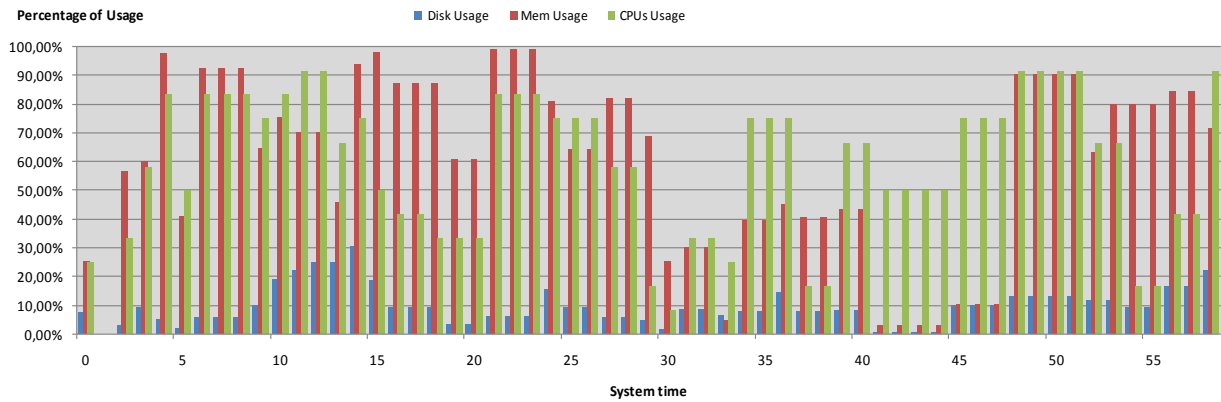
Σχήμα 48: Ποσοστιαία χρησιμοποίηση πόρων με υλοποίηση πολιτικής FCFS μετα-δρομολόγησης, κέρδος = 14866 (2^η Επανάληψη)



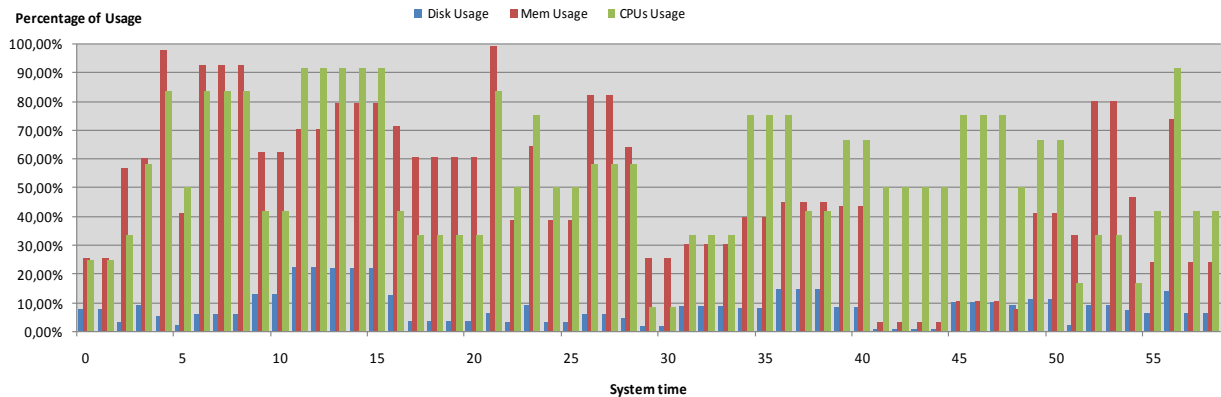
Σχήμα 49: Ποσοστιαία χρησιμοποίηση πόρων με τον απλό δρομολογητή του GRIA, κέρδος = 11198 (2^η Επανάληψη)



Σχήμα 50: Ποσοστιαία χρησιμοποίηση πόρων με το μετα-δρομολογητή σακιδίου, κέρδος = 11843 (3^η Επανάληψη)



Σχήμα 51: Ποσοστιαία χρησιμοποίηση πόρων με υλοποίηση πολιτικής FCFS μετα-δρομολόγησης, κέρδος = 10301 (3^η Επανάληψη)



Σχήμα 52: Ποσοστιαία χρησιμοποίηση πόρων με τον απλό δρομολογητή του GRIA, κέρδος = 4259 (3^η Επανάληψη)

Τα δύο εναλλακτικά συστήματα με τα οποία συγκρίνεται ο μηχανισμός δεν χρησιμοποιούν την τεχνική διαχείρισης κινδύνου, άρα δεσμεύουν μόνο τους πόρους που ικανοποιούν πλήρως τις απαιτήσεις των εργασιών. Σημειωτέον ότι αν οι διαθέσιμοι πόροι είναι σημαντικά μικρότεροι από το ποσό που απαιτείται, στην πραγματικότητα η δέσμευσή τους δεν αξίζει, αφού η πιθανότητα της αποτυχίας της εργασίας ή παραβίασης του SLA είναι εξαιρετικά υψηλή. Για το λόγο αυτό τέθηκε ένα χαμηλότερο όριο της τάξης του 25%. Επιπλέον, για εφαρμογές όπως το 3Delight, υπάρχουν αυστηρά καθορισμένες ελάχιστες απαιτήσεις για μνήμη που, σε περίπτωση που δεν πληρούνται οδηγούν σε σίγουρη αποτυχία και, επομένως, οι πόροι δεν δεσμεύονται σε μια τέτοια περίπτωση. Οι πόροι αυτοί μπορούν να διατεθούν για άλλες εργασίες στην ουρά ή απλά να διατηρούνται ελεύθεροι ως εφεδρικοί για την ασφάλεια του συστήματος διαχείρισης.

	Fractional knapsack meta-scheduler	GRIA simple meta-scheduler, no queue	FCFS policy meta-scheduler
1st Iteration			
<i>Job success rate</i>	83,3%	66,7%	70%
<i>Mean disk utilization</i>	7,4%	6,8%	6,7%
<i>Mean memory utilization</i>	49,8%	39,1%	45%
<i>Mean CPU utilization</i>	37,5%	35,5%	36,9%
<i>Total payoff</i>	16.557	11.670	12.540
2nd Iteration			
<i>Job success rate</i>	86,7%	73,3%	83,3%
<i>Mean disk utilization</i>	8%	8,4%	8,4%
<i>Mean memory utilization</i>	42,4%	41,9%	45,5%
<i>Mean CPU utilization</i>	32%	29,8%	29,9%
<i>Total payoff</i>	16.244	11.198	14.866
3rd Iteration			
<i>Job success rate</i>	86,7%	70%	86,7%
<i>Mean disk utilization</i>	9,7%	7,8%	9,7%
<i>Mean memory utilization</i>	51,1%	46,7%	59,1%
<i>Mean CPU utilization</i>	54,7%	50,7%	53,2%
<i>Total payoff</i>	16.816	12.894	16.598

Πίνακας 18: Σύνοψη των πειραματικών αποτελεσμάτων

Μια συνολική σύνοψη των αποτελεσμάτων των πειραμάτων που διενεργήθηκαν φαίνεται στον Πίνακα 18. Αναλύοντας τα αποτελέσματα των πειραμάτων και τα παραπάνω διαγράμματα χρησιμοποίησης πόρων, χρήσιμα συμπεράσματα μπορούν να εξαχθούν σχετικά με τον εφαρμοζόμενο μηχανισμό μετα-δρομολόγησης. Πρώτον, το συνολικό κέρδος του παρόχου με το μετα-δρομολογητή σακιδίου είναι σημαντικά μεγαλύτερο από αυτό του απλού GRIA ή της FCFS χρονοδρομολόγησης, όπως φαίνεται παραπάνω. Το ίδιο συμβαίνει και για το μέσο ποσοστό επιτυχίας εκτέλεσης μιας εργασίας, χωρίς παραβιάσεις SLA, καθώς και για την αξιοποίηση των πόρων του cluster. Το γεγονός αυτό υποδεικνύεται επίσης και από την οξύτητα των αλλαγών στα ποσοστά χρησιμοποίησης ανά το χρόνο (Σχήμα 44, Σχήμα 47 και Σχήμα 50), λόγω της συχνής αναδιανομής των πόρων κάθε φορά που συντρέχουν οι συνθήκες «ενεργοποίησης» του αλγορίθμου, όπως εξηγήθηκαν παραπάνω.

Η χρησιμοποίηση ορισμένων πόρων μπορεί να είναι μερικές φορές μεγαλύτερη με χρήση FCFS μετα-δρομολόγησης, κυρίως λόγω του πρόωρου επιτυχούς τερματισμού κάποιων εργασιών με χρήση του αλγορίθμου σακιδίου και της καθυστέρησης αφίξεων νέων. Ωστόσο, υπό τον ίδιο φόρτο εργασίας, η χρησιμοποίηση πόρων του μετα-δρομολογητή σακιδίου είναι κοντά στην ιδανική (αυτό μπορεί να παρατηρηθεί στο χρονικό σημείο όπου εισέρχεται η πρώτη εργασία απαιτεί πόρους περισσότερους από τους διαθέσιμους), αλλά και το καταλυτικό για την αξιολόγηση μέτρο του συνολικού κέρδους είναι πάντα καλύτερο.

Τέλος, οι χρονισμοί των αποτελεσμάτων χρησιμοποίησης δείχνουν να εισάγεται μια μικρή καθυστέρηση από την εκτέλεση του αλγορίθμου μετα-δρομολόγησης κατά τη

διανομή των πόρων στις εκκρεμείς εργασίες, η οποία, ωστόσο, δεν επηρεάζει σημαντικά τη συνολική απόδοση. Οι επιπτώσεις στην απόδοση γενικότερα, λόγω της δυναμικής ανακατανομής πόρων, μελετήθηκαν επίσης από τον (Wang Zhikui, 2007), σε συνδυασμό με τη συχνότητα ανακατανομής. Σε αυτή τη μελέτη, έχοντας πολλαπλά φορτία εργασίας για συναλλαγές που φιλοξενούνται σε εικονικές μηχανές Xen, η συνολική απώλεια CPU χωρητικότητας (που αφορά τους επεξεργαστές που παρέχονται κατά τη διάρκεια μιας χρονικής περιόδου), φτάνει το 5%, ως συνέπεια της δυναμικής ανακατανομής. Αντίστοιχα λοιπόν και στα δικά μας πειράματα, η πρόσθετη επιβάρυνση υπολογίζεται ότι δεν υπερβαίνει το ποσοστό αυτό.

6.5

Συμπεράσματα

Ο μηχανισμός που παρουσιάστηκε στο παρόν κεφάλαιο, μαζί με τα πειράματα που διενεργήθηκαν, είχαν στόχο να δείξουν πώς μια υπηρεσιοστρεφής υποδομή, που βασίζεται στις αρχές του Software as a Service, μπορεί να μεγιστοποιήσει το κέρδος του παρόχου, χρησιμοποιώντας ένα συνδυασμό των παρεχόμενων υπηρεσιών και αρχιτεκτονικές προσανατολισμένες στις τεχνολογίες εικονικοποίησης. Στα πλαίσια των πειραμάτων αυτών λοιπόν, δημιουργήσαμε μια ευριστική συνάρτηση που αξιολογεί την προτεραιότητα κάθε εργασίας στο χρόνο εκτέλεσης με βάση τη διαθεσιμότητα των πόρων, τον κίνδυνο της αποτυχίας εκτέλεσης και παραβίασης του SLA, το κόστος της παραβίασης και το συνολικό κέρδος από την επιτυχή εκτέλεση.

Δεδομένης της δυναμικότητας με την οποία η προτεραιότητα για την εκτέλεση μιας εργασίας αλλάζει, η κατανομή πόρων σε αυτή μπορεί ανά πάσα στιγμή να εμπίπτει σε μία από αυτές τις κατηγορίες: πλήρη διάθεση απαιτούμενων πόρων, μερική διάθεση απαιτούμενων πόρων, μηδενική διάθεση πόρων. Η μετάβαση από τη μία κατηγορία στην

άλλη δεν είναι πάντα απλή, αφού κάποιες εργασίες μπορεί να μην είναι δυνατόν να διακόπτονται. Για το λόγο αυτό, οι πόροι που διατίθενται εικονικοποιούνται, κάτι που σημαίνει ότι κάτω από την εποπτεία μιας εικονικής μηχανής, οι παράμετροι και τα δεδομένα μιας εργασίας μπορεί να παγώνουν, να αποθηκεύονται και να επανενεργοποιούνται οποιοδήποτε χρονική στιγμή.

Η προτεινόμενη λύση εκμεταλλεύθηκε αυτό το γεγονός προκειμένου να χρησιμοποιήσει το μέρος των πόρων που παραμένει ελεύθερο, ακόμα κι αν αυτό δεν ικανοποιεί πλήρως τον απαιτούμενο φόρτο εργασίας από τις πιο «σημαντικές» εργασίες που παραμένουν στην ουρά. Επιπλέον, η εικονικοποίηση της υποδομής επιτρέπει τη μεγαλύτερη δυνατή κατάτμηση των διαθέσιμων πόρων, με αύξηση του συντελεστή ελαστικότητάς τους. Η τελευταία παρατήρηση οδηγεί σε μια πολύ ευέλικτη τεχνολογικά λύση, οδηγεί επίσης όμως και στο συμπέρασμα ότι στην πράξη, η βελτιστοποίηση της συνάρτησης προτεραιότητας που αναφέρθηκε είναι ένα NP-hard πρόβλημα που παρουσιάζει ομοιότητες με το κλασματικό πρόβλημα του σακιδίου. Ως εκ τούτου, το πρόβλημα μοντελοποιήθηκε ως ένα πρόβλημα κλασματικού σακιδίου και χρησιμοποιήθηκε ένας άπληστος αλγόριθμος για την επίλυσή του.

Επιπλέον, το κεφάλαιο αυτό πρότεινε μια αρχιτεκτονική υλοποίησης και αξιολόγησης του μοντέλου, που συνδυάζει ένα μεσολογισμικό, που παρέχει ένα SOA περιβάλλον, και μια εικονικοποιημένη υποδομή. Βάσει αυτού, δημιουργήθηκαν πειραματικά σύνολα δεδομένων για την αξιολόγηση και επικύρωση του μηχανισμού (σύνολα εργασιών και τα αντίστοιχα SLAs με μεταβλητά κόστη και απαιτήσεις πόρων) μέσω πειραμάτων. Η σύγκριση με υπάρχοντα σχήματα αξιολόγησης ποιότητας

υπηρεσίας και διαχείρισης πόρων έδειξε ελπιδοφόρα αποτελέσματα για τη μεγιστοποίηση του κέρδους και την αξιοποίηση των διαθέσιμων πόρων υποδομής.

Τα αποτελέσματα δείχνουν ότι αν η ευριστική συνάρτηση προτεραιότητας που εκτιμά το κέρδος μπορεί να μοντελοποιήσει σωστά το πραγματικό κέρδος που εξάγεται από τα επιχειρηματικά μοντέλα των παρόχων υπηρεσιών, τότε η λύση αυτή μπορεί να είναι αξιοποιήσιμη. Ωστόσο, υπάρχουν ακόμα παράγοντες που πρέπει να συνεκτιμηθούν, όπως το κόστος συντήρησης της υψηλής χρησιμοποίησης των πόρων, έστω κι αν ο χρόνος λειτουργίας σε βάθος χρόνου μπορεί να ελαττωθεί. Το τελευταίο ισχύει ιδιαίτερα στην περίπτωση όπου οι ομάδες εισερχόμενων εργασιών φτάνουν στο σύστημα κατά κύματα, όμως, στην περίπτωση μιας πιο ομαλής κατανομής της εισερχόμενης ροής αυτό δεν είναι αλήθεια και ο παρών μηχανισμός θα πρέπει να διερευνηθεί περαιτέρω και προς αυτή τη μεριά.

Αξίζει, τέλος, να σημειωθεί ότι γίνεται η ακόλουθη υπόθεση: Παρόλο που δεν ενδιαφερόμαστε για την παραλληλοποίηση των εργασιών που υποβάλλονται, στην πραγματικότητα αυτή είναι μια κρίσιμη παράμετρος, αν θέλουμε να καλύψουμε όλες τις περιπτώσεις, εφόσον η εικονικοποίηση δημιουργεί ένα φαινομενικά ενιαίο σύνολο πόρων. Για παράδειγμα, αν το σύνολο των υπολογιστικών πόρων αποτελείται από έναν αριθμό διαφορετικών μηχανών και μια εργασία απαιτεί περισσότερους πόρους από αυτούς που ο καλύτερος εξυπηρετητής μπορεί να παρέχει, τότε προκύπτει το ζήτημα πώς το στρώμα εικονικοποίησης θα χειριστεί την εργασία χωρίς να τη σπάσει σε μικρότερες μονάδες. Ωστόσο, αυτό είναι ένα θέμα που αφορά περισσότερο το ίδιο το στρώμα εικονικοποίησης των πόρων και θεωρείται εκτός του θέματος της παρούσας διατριβής, η

οποία αντιμετωπίζει το πρόβλημα στο επίπεδο μετα-δρομολόγησης. Εξακολουθεί όμως να παραμένει ένα θέμα που μπορεί να μελετηθεί περαιτέρω στη μελλοντική έρευνα.

7

Συμπεράσματα και μελλοντικές ερευνητικές κατευθύνσεις

Στο τελευταίο αυτό κεφάλαιο συνοψίζονται οι ερευνητικές κατευθύνσεις και τα αποτελέσματα της αξιολόγησης των μηχανισμών που παρουσιάστηκαν, για να εξαχθούν κάποια γενικά συμπεράσματα. Τέλος, αναφέρονται οι μελλοντικές κατευθύνσεις της έρευνας όσον αφορά την αξιολόγηση και διασφάλιση ποιότητας υπηρεσιοστρεφών συστημάτων που παρέχουν εφαρμογές ως υπηρεσίες και αναφέρονται κάποια άμεσα σχέδια επέκτασης της παρούσας εργασίας και περαιτέρω αξιολόγησης των αρχιτεκτονικών που παρουσιάστηκαν.

7.1 Σύνοψη και Συμπεράσματα

Η αξιολόγηση της Ποιότητας Υπηρεσίας σε υπηρεσιοστρεφή περιβάλλοντα αποτελεί αναμφισβήτητα ένα σημαντικότερο πεδίο έρευνας, ειδικά όταν μιλάμε για επιχειρηματικούς οργανισμούς. Στη σύγχρονη εποχή του Internet of Services οι ανοιχτές

αγορές εφαρμογών Software as a Service αναμένεται να παίξουν έναν πρωταγωνιστικό ρόλο στο διαδικτυακό χάρτη. Οι πελάτες και οι πάροχοι των υπηρεσιών αυτών πρέπει να συμφωνούν σε απτές και μετρήσιμες παραμέτρους που θα επιτρέψουν την διμερή αξιολόγηση της ποιότητας και από τις δύο πλευρές, κατά την παράδοση του αποτελέσματος. Ωστόσο, αυτό δεν είναι πάντα εφικτό, κυρίως διότι οι τα σύγχρονα περιβάλλοντα παροχής υπηρεσιών είναι ανεξάρτητα από την εφαρμογή (π.χ. υποδομές Cloud) και βασίζονται σε καταναμημένες υποδομές. Συνεπώς, είναι δύσκολο να χαρτογραφηθεί το μοντέλο αξιολόγησης επιδόσεων μιας υποδομής σε ειδικές παραμέτρους ποιότητας για κάθε εφαρμογή.

Στα περισσότερα από τα συστήματα που εμπίπτουν σε αυτή την κατηγορία των υπηρεσιοστρεφών πλατφορμών, μια συνήθης πρακτική είναι η χρήση παραμέτρων επιπέδου πόρων, για τη μοντελοποίηση του συμφωνηθέντος μεταξύ των δύο πλευρών επιπέδου ποιότητας. Αυτές μπορούν εύκολα να παρακολουθούνται από τον πάροχο και είναι ανεξάρτητες από την εφαρμογή. Το πρόβλημα στην περίπτωση αυτή είναι η δυσκολία του πελάτη στην παρακολούθηση και αξιολόγηση της ποιότητας, ειδικά όταν μιλάμε για έναν απλό χρήστη. Διάφορες ερευνητικές μελέτες υποστηρίζουν ότι λύση αποτελεί ένα σχήμα αξιολόγησης της φήμης του παρόχου από τους πελάτες με βάση τη δική του εμπειρία αλληλεπιδράσεων ή σε συνεργασία με άλλους πελάτες.

Σε αυτήν την κατεύθυνση, το πρώτο ερευνητικό σκέλος της παρούσας διδακτορικής διατριβής εισήγαγε ένα σύστημα αξιολόγησης της φήμης των παρόχων από μια Τρίτη οντότητα κοινής εμπιστοσύνης, το οποίο χρησιμοποιεί την έννοια της Ποιότητας Εμπειρίας του καταναλωτή για την πρόβλεψη της μελλοντικής απόδοσης. Στο πλαίσιο αυτό χρησιμοποιήθηκε μια γνωστή τεχνική συνεργατικού φιλτραρίσματος, για

τη δημιουργία συσχέτισης μεταξύ των καταναλωτών υπηρεσιών. Το τελικό αποτέλεσμα της ερευνητικής εργασίας συνίσταται σε ένα σύστημα αξιολόγησης ποιότητας SaaS παρόχων και παροχής συστάσεων, ενσωματωμένο σε έναν Service Broker που χρησιμοποιεί ένα μητρώο UDDI, με σκοπό να υποστηρίξει τους πελάτες κατά το στάδιο επιλογής υπηρεσιών.

Τα πειράματα που διενεργήθηκαν για την επικύρωση του προαναφερθέντος συστήματος, αρχικά χρησιμοποίησαν ένα εκτεταμένο σύνολο ρεαλιστικά προσομοιωμένων δεδομένων, ενώ στη συνέχεια η υπηρεσία συστάσεων χρησιμοποιήθηκε από πραγματικούς τελικούς χρήστες για να επαληθευτεί η ορθότητα των συστάσεων. Η υπηρεσία προς αξιολόγηση αφορούσε μια εφαρμογή 3D απόδοσης, η οποία και αποτελεί μέρος πραγματικών επιχειρηματικών σεναρίων, ενώ η πειραματική διάταξη αποτελούνταν από ένα υπάρχον διαδεδομένο μεσολογισμικό πλέγματος, συνδεδεμένο με το Service Broker. Τα αποτελέσματα των πειραμάτων ήταν ενθαρρυντικά, ωστόσο παραμένει ανοικτή ακόμα η δυνατότητα εκτενέστερης αξιολόγησης με πραγματικούς χρήστες ως ένα stress test, που αναμένεται να βελτιώσει περαιτέρω τη λειτουργία του Service Recommender.

Από την άλλη μεριά, η αξιοποίηση των δυνατοτήτων των εικονικοποιημένων υποδομών για ελαστική κατανομή των πόρων είναι ένα από τα σύγχρονα θέματα που απασχολούν την έρευνα όλο και περισσότερο τον τελευταίο καιρό. Το δεύτερο σκέλος της παρούσας διατριβής που παρουσιάστηκε, διερευνά το πρόβλημα κάτω από ένα διαφορετικό πρίσμα, αυτό της αξιολόγησης της ποιότητας εφαρμογής από τον ίδιο τον πάροχο υπηρεσιών και της μεγιστοποίησης του κέρδους του. Σε ένα επιχειρηματικό περιβάλλον το κέρδος είναι στενά συνδεδεμένο με την αξιοποίηση των πόρων και την

ποιότητα που παρέχεται προς τον πελάτη. Συνεπώς, η αξιολόγηση της ποιότητας από την πλευρά του παρόχου και η συνεπαγόμενη ορθολογική διαχείριση των πόρων του, συμβάλλει στην καλύτερη αξιοποίηση της υποδομής του και την αύξηση του κέρδους.

Ο μηχανισμός που παράχθηκε έλαβε υπόψη του όλα τα δηλωθέντα κόστη στο ηλεκτρονικό συμβόλαιο μεταξύ του πελάτη και του παρόχου, καθώς και το ιστορικό των εκτελέσεων, για να εξάγει τις πιθανότητες παραβίασης του SLA και το σχετικό ρίσκο. Ανάγοντας το πρόβλημα κατανομής των εικονικοποιημένων πόρων (CPUs, μνήμη, δίσκος) σε μια παραλλαγή του γνωστού προβλήματος του σακιδίου και χρησιμοποιώντας μια ευριστική συνάρτηση υπολογισμού του κέρδους και έναν άπληστο αλγόριθμο για την επίλυση, παρουσιάστηκε μια σχεδόν ιδανική (near optimal) λύση στο επίπεδο της μεταδρομολόγησης εργασιών, η οποία υλοποιήθηκε στην αρχιτεκτονική αναφοράς.

Τα πειράματα χτίστηκαν στο ίδιο μεσολογισμικό με πριν και αφορούσαν πάλι την εφαρμογή 3D Rendering, ως μια υπηρεσία με υψηλές απαιτήσεις πόρων. Έτσι δόθηκε η πλήρης, από άκρο σε άκρο (end-to-end) αρχιτεκτονική υλοποίησης για τη διμερή αξιολόγηση της ποιότητας από τον πελάτη και τον πάροχο, και των συνεπαγόμενων συνεπειών (βελτίωση επιλογών υπηρεσίας και αποδοτικότερη αξιοποίηση υποδομών αντίστοιχα). Ως σύνολα πειραματικών δεδομένων δημιουργήθηκαν διάφορες εργασίες 3D απόδοσης υψηλών απαιτήσεων και υποβλήθηκαν σε τρία συνολικά εναλλακτικά συστήματα δρομολόγησης, κάτω από τα οποία υπήρχε η ίδια υπολογιστική υποδομή. Τα αποτελέσματα έδειξαν σαφή βελτίωση τόσο της χρησιμοποίησης των πόρων, όσο και του συνολικού κέρδους στο τέλος όλων των εκτελέσεων.

7.2

Ερευνητικές κατευθύνσεις και μελλοντική εργασία

Είναι γενικά αποδεκτό ότι η κατεύθυνση της έρευνας προς την οποία η αξιολόγηση των υπηρεσιών κλίνει είναι η χρήση της συνεργατικής νοημοσύνης για τη διασφάλιση της ποιότητας παροχής, ως η μόνη φαινομενικά αντικειμενική και βιώσιμη μέθοδος. Επιπλέον, ο ορισμός των παραμέτρων τείνει προς το υψηλό επίπεδο, που θα καθορίζει με ακρίβεια την ποιότητα, και από την άποψη των καταναλωτών υπηρεσιών θα βελτιώσει σε μεγάλο βαθμό την τεχνική αξιολόγηση. Έτσι θα δοθεί η δυνατότητα για το σχηματισμό τομέων αξιολόγησης, περιορίζοντας την ποσότητα των υπηρεσιών που κάθε καταναλωτής πρέπει να αξιολογήσει, καθώς και για τη δημιουργία δικτύων εμπιστοσύνης μεταξύ των ανθρώπων που καταλαβαίνουν (και τελικά καταναλώνουν) ορισμένους τύπους παρεχόμενων υπηρεσιών.

Αυτό, φυσικά, προϋποθέτει την ανάπτυξη αποτελεσματικών μηχανισμών και σημασιολογιών που βοηθούν στην μετατροπή αυτών των παραμέτρων υψηλού επιπέδου σε υπολογιστικούς όρους χαμηλού επιπέδου, πιο χρήσιμων στη διαδικασία διαχείρισης πόρων στο πλαίσιο της παροχής υπηρεσιών. Αυτή η μετατροπή, που μέχρι τώρα γίνεται κυρίως εμπειρικά, αποτελεί μια από τις μεγαλύτερες προκλήσεις στον τομέα της παροχής εφαρμογών ως υπηρεσίες και μπορεί να διευκολύνει ιδιαίτερα παρόχους με περιορισμένη διαθεσιμότητα υπολογιστικών πόρων, ώστε να συμμετέχουν δυναμικά σε ανοικτές αγορές υπηρεσιών.

Επιπροσθέτως, η κατεύθυνση της έρευνας στα υπηρεσιοστρεφή περιβάλλοντα γενικά, κινήθηκε από τα τελευταία χρόνια πέρα από τα υπολογιστικά συστήματα υψηλής απόδοσης (High Performance Computing) προς την εφαρμογή στρώματος

εικονικοποίησης στους πόρους, ώστε να παρέχεται στον τελικό χρήστη μια ενιαία και ευέλικτη εικονική υποδομή, οποιονδήποτε διαστάσεων που μεταβάλλεται ανάλογα με τις ανάγκες του. Έτσι περάσαμε από τα συστήματα πλέγματος στα περιβάλλοντα υπολογιστικού νέφους, αλλά και γενικά στην παροχή υπηρεσιών με εικονικοποιημένους πόρους. Σε αυτό το νέο τοπίο, η ορθολογική διαχείριση των πόρων είναι καταλυτικός παράγοντας για τη βελτίωση της ποιότητας παροχής και με πυρήνα αυτή θα προχωρήσει η έρευνα τα επόμενα χρόνια.

Βάσει των παραπάνω, η μελλοντική εργασία που αναμένεται ως συνέχεια της τρέχουσας διατριβής, θα αφορά μια εκτενέστατη αξιολόγηση των παρουσιασθέντων μηχανισμών και αρχιτεκτονικών σε ένα πραγματικό επιχειρηματικό περιβάλλον, όπως αυτό του ευρωπαϊκού έργου SocIoS. Σε αυτό το πλαίσιο, μπορεί να αξιολογηθεί το σύστημα συστάσεων που παρουσιάστηκε σε μια ανοικτή αγορά υπηρεσιών SaaS με εκατοντάδες πραγματικούς τελικούς χρήστες των διατιθέμενων εφαρμογών, κάτι που αναμένεται να συντελέσει στη δημιουργία γκρουπ προτιμήσεων και να βελτιώσει τις προσωποποιημένες συστάσεις προς τους χρήστες. Οι συστάσεις αυτές και η δημιουργία φήμης για κάθε πάροχο υπηρεσιών κρίνεται απαραίτητη, εφόσον σε μια ανοικτή αγορά μπορεί να υπάρχει πληθώρα εναλλακτικών λύσεων για κάθε τύπο υπηρεσίας.

Δεδομένου ωστόσο ότι αρκετές από τις βοηθητικές υπηρεσίες (Auxiliary Services) φιλοξενούνται στην πλατφόρμα του SocIoS, τίθεται το ζήτημα της αξιολόγησης της ποιότητας παροχής μέσα στην πλατφόρμα και της σωστής αξιοποίησης των πόρων. Έτσι, μια άλλη μελλοντική εργασία αφορά την αξιολόγηση του μηχανισμού διαχείρισης πόρων από τον πάροχο σε μια τέτοια πλατφόρμα, για υπηρεσίες με υψηλές απαιτήσεις πόρων, όπως αυτή της Ανάλυσης Συναισθήματος στα κοινωνικά δίκτυα. Μια

τέτοια υπηρεσία επεξεργάζεται εκατοντάδες χιλιάδες μηνύματα (tweets) και ένας απλός server δεν μπορεί να την εξυπηρετήσει σε καμία περίπτωση. Βέβαια, η εφαρμογή του παρουσιασθέντος μηχανισμού αξιολόγησης απαιτεί εικονικοποίηση των πόρων, αντίστοιχη με τη λογική ενός Cloud συστήματος.

Τέλος, όπως αναφέρθηκε και στα προηγούμενα, η ακριβής υλοποίηση μιας τέτοιας λύσης μετα-δρομολόγησης μπορεί να απαιτεί παραλληλοποίηση των εργασιών που υποβάλλονται, κάτι το οποίο θα πρέπει να μελετηθεί στο επίπεδο εικονικοποίησης για κάθε εφαρμογή (για την υπηρεσία Ανάλυσης Συναισθημάτων κρίνεται απλό, μέσω αντίστοιχης κατανομής των μηνυμάτων). Επίσης παραμένει ως μελλοντική εργασία να μοντελοποιηθεί η επιπλέον επιβάρυνση που δημιουργείται από την αναστολή και την επανενεργοποίηση των εικονικών μηχανών. Αυτή η επιβάρυνση έχει ληφθεί υπόψη στους υπολογισμούς που ενσωματώθηκαν στις μετρήσεις για τα πειράματα παραπάνω, ωστόσο ένα μοντέλο πρόβλεψης ή εκτίμησής της θα ήταν ιδιαίτερα χρήσιμο για τη συνολική αξιολόγηση του όλου μηχανισμού.

Γλωσσάριο

Στον παρακάτω πίνακα παρατίθενται οι όροι που χρησιμοποιήθηκαν στη Διατριβή:

+SPACES	Policy Simulation in Virtual Spaces
AAA	Authorization Authentication Accounting
API	Application Programming Interface
B2B	Business-to-Business
BB	Branch and Bound
BT	Backtracking
CP	Constraint Programming
CPU	Central Processing Unit
DP	Dynamic Programming
DSML	Domain Specific Modeling Languages
DSS	Decision Support Systems
EPR	EndPoint Reference
FCFS	First-Come, First-Served
GEANT	Gigabit European Advanced Network Technology
GA	Genetic Algorithm
GRAM	Globus Resource Access Manager
GRIA	Grid Resources for Industrial Applications
GSI	Globus Security Interface
GUI	Graphical User Interface
HPC	High Performance Computing
HTTP	HyperText Transfer Protocol
IaaS	Infrastructure as a Service
IoS	Internet of Services
IP	Internet Protocol
ISV	Independent Software Vendor
LAN	Local Area Network
LGPL	Lesser General Public License
LRPT	Longest Remaining Processing Time First
LSF	Load Sharing Facility
MAN	Metropolitan Area Network
MDS	Metacomputing Directory Service
MPE	Mean Percent Error
MPI	Message Passing Interface
MPP	Massively Parallel Processing
NP-HARD	Non-deterministic Polynomial-time hard
NSFNET	National Science Foundation Network

NIST	National Institute of Science and Technology
NFS	Network File System
NQE	Network Queuing Environment
OASIS	Organization for the Advancement of Structured Information Standards
OGSA	Open Grid Service Architecture
OGSI	Open Grid Services Infrastructure
P2P	Peer to Peer
PaaS	Platform as a Service
PBS	Portable Batch System
PICML	Platform-Independent Component Modeling Language
PVM	Parallel Virtual Machine
Q-WSEM	QoS based Web Service Evaluation Model
QoBiz	Quality of Business
QoE	Quality of user Experience
QoS	Quality of Service
RAM	Random Access Memory
RDF	Resource Description Framework
REST	Representational State Transfer
RIB	RenderMan Bytestream Interface
RMs	Resource Managers
RPC	Remote Procedure Call
SaaS	Software as a Service
SLA	Service Level Agreements
SLM	Service Level Management
SMP	Symmetric Multi-Processing
SNSs	Social Networking Sites
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOCIOS	Exploiting SOCIAL Networks for Building the Future Internet Of Services
SOI	Service Oriented Infrastructure
SPI	Software Platform Infrastructure
SR	Service Recommendation
SRB	Storage Resource Broker
SRPT	Shortest-Remaining-Processing-Time-First
SSH	Secure Shell
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TTP	Trusted Third Party
UDDI	Universal Description, Discovery and Integration
URL	Uniform Resource Locator
VC	Violation Cost
VM	Virtual Machine
VP	Violation Probability
VSML	Virtual Spaces Management Layer
W3C	World Wide Web Consortium
WADL	Web Application Description Language
WAN	Wide Area Network
WS	Web Service
WSAP	Web Service Agent Proxy
WSDL	Web Services Description Language
WSMO	Web Services Modeling Ontology
WSRF	Web Services Resource Framework
XML	Extensible Markup Language

Βιβλιογραφικές Αναφορές

- 3Delight RenderMan - compliant renderer. 2008, available at: <http://www.3delight.com>
- Addis, M., Zlatev, Z., Mitchell, W., & Boniface, M. (2009). *Modelling interactive real-time applications on service oriented infrastructures*. 2009 NEM Summit – Towards Future Media Internet. St Malo.
- Aisopos, F., Kardara, M., Moulos, V., Papaoikonomou, A., Tserpes, K., & Varvarigou, T. (2012). Service Quality Model Evaluation. In *Achieving Real-Time in Distributed Computing: From Grids to Clouds*, pp. 59-73. IGI Global.
- Aisopos, F., Kardara, M., Senger, P., Klinger, R., Papaoikonomou, A., Tserpes, K., et al. (2012). *e-Government and Policy Simulation in Intelligent Virtual Environments*. WEBIST - 8th International Conference on Web Information Systems and Technologies, pp. 129-135. Porto, Portugal
- Aisopos, F., Papadakis, G., & Varvarigou, T. (2011). *Sentiment analysis of social media content using N-Gram graphs*. 3rd ACM SIGMM international workshop on Social media (WSM '11), pp. 9-14. Scottsdale, Arizona, USA: ACM, New York.
- Aisopos, F., Papadakis, G., Tserpes, K., & Varvarigou, T. (2012). *Content vs. Context for Sentiment Analysis: a comparative analysis over microblogs*. 23rd ACM Conference on Hypertext and Social Media (HT'12). Milwaukee, USA: ACM, New York.

Aisopos, F., Papadakis, G., Tserpes, K., & Varvarigou, T. (2012). *Textual and contextual patterns for sentiment analysis over microblogs*. 21st international conference companion on World Wide Web, pp. 453-454. Lyon, France.

Aisopos, F., Tserpes, K., & Varvarigou, T. (2011). Resource management in software as a service using the knapsack problem model. *International Journal of Production Economics*. Avail. online 16 December 2011. ISSN 0925-5273, 10.1016/j.ijpe.2011.

Aisopos, F., Tserpes, K., Kardara, M., Panousopoulos, G., Phillips, S., & Salamouras, S. (2010). Information exchange in business collaboration using Grid technologies. *Identity in the Information Society* , 2, pp. 189-204.

Aisopos, F., Tserpes, K., Panousopoulos, G., Phillips, S. C., Kardara, M., & Varvarigou, T. (2009). Building Business Trust: Inter-organizational information sharing supported by Grid technologies. *eChallenges e-2009 Conference*. Istanbul.

Alrifai, M., & Risse, T. (2010). Efficient QoS-aware service composition. In M. Alrifai, T. Risse, M. Calisti, M. Walliser, S. Brantschen, & M. a. Herbstritt (Eds.), *Emerging Web services technology* (volume III, pp. 75-87). Whitestein Series in Software Agent Technologies and Autonomic Computing, Birkhäuser Basel.

Amazon. (2008). *Service level agreement for ec2*. Ανάκτηση από:

<http://aws.amazon.com/ec2-sla>

Andrieux A., Czajkowski K., Dan A., Keahey K., Ludwig H., Nakata T., Pruyne J., Rofrano J., Tuecke S., Xu M.. *Web Services Agreement Specification (WS-Agreement)*. 2011. <http://www.gridforum.org/documents/GFD.192.pdf>

- Bajaj, S., Box, D., & Chappell, D. (2006). *Web services policy framework (WS-Policy)*.
Ανάκτηση από: <http://www-128.ibm.com/developerworks/library/specification/wspolfram/>
- Balasubramanian, K., Balasubramanian, J., Parsons, J., Gokhale, A., & Schmidt, D. C. (2007). A platform-independent component modeling language for distributed real-time and embedded systems. *Journal of Computer and System Sciences*, 73(2), pp. 171-185.
- Basso, A., Peccati, L. (2001). Optimal resource allocation with minimum activation levels and fixed costs, *European Journal of Operational Research*, 131 (3): 536-549.
- Beamon, Benita M. (1998). Supply chain design and analysis: Models and methods. *International Journal of Production Economics* 55, no. 3: 281-294.
- Biswas, S. (2011, February 15). *Cloud Computing For Dummies: SaaS, PaaS, IaaS And All That Was*. Ανάκτηση από: <http://www.cloudtweaks.com/2011/02/cloud-computing-for-dummies-saas-paas-iaas-and-all-that-was/>
- Blum, A. (2004, June 4). *UDDI as an extended Web services registry: Versioning, quality of service, and more*. SOA & WOA: Article .
- Boniface, M. J., Phillips, S., Sanchez-Macian Perez, A., & SurrIDGE, M. (2007). *Dynamic service provisioning using GRIA SLAs*. NFPSLA-SOC'07. Vienna.
- Breese, J., Heckerman, D., & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *14th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 43-52. Madison.
- Buco, M. J., Chang, R. N., Luan, L. Z., Ward, C., Wolf, J. L., & Yu, P. S. (2004). *Utility computing SLA management based upon business objectives*.

- Cao, J., Huang, J., Wang, G., & Gu, J. (2009). *QoS and preference based Web service evaluation approach*. Eighth International Conference on Grid and Cooperative Computing, pp. 420-426.
- Catlett, C. (2004). 3G Grids - "Third Generation Grids". In: culture.publications.news.
- Chaari, S., Badr, Y., & Biennier, F. (2008). *Enhancing Web service selection by QoS-based ontology and WS-policy*. ACM Symposium on Applied Computing, SAC '08, pp. 2426-2431. Fortaleza, Ceara, Brazil: ACM.
- Condor High Throuput Computing, *The Condor Project*. 2012.
<http://www.cs.wisc.edu/condor/>.
- Cucinotta, T., Checconi, F., Abeni, L., & Palopoli, L. (2010). *Self-tuning schedulers for legacy real-time applications*. EuroSys '10, 5th European Conference on Computer Systems, pp. 55-68. Paris, France: ACM.
- Debusmann, M., & Keller, A. (2003). *SLA-driven management of distributed systems using the common information model*. IFIP/IEEE International Symposium on Integrated Management. Kluwer Academic Publishers.
- Ding, Q., Li, X., & Zhou, X. H. (2008). *Reputation based service selection in Grid environment*. International Conference on Computer Science and Software Engineering, 3, pp. 58-61.
- Emeakaroha, C. V., Brandic, I., Maurer, M., & Dustdar, S. (2010). *Low level metrics to high level SLAs - LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments*. IEEE 3rd International Conference on Cloud Computing.

- Ferretti, S., Ghini, V., Panzieri, F., Pellegrini, M., & Turrini, E. (2010). QoS-aware clouds. IEEE 3rd International Conference on Cloud Computing.
- Force, D. M. (1999, June). *Common information model (CIM) version 2.2*. Specification.
- Foster, I., Kesselman, C., Nick, J., & Tuecke, S. (2002, July). The physiology of the grid: An open grid services architecture for distributed systems integration. Ανάκτηση από: Globus Project.
- Foster, I., Kishimoto, H., Savva, A., Berry, D., Djaoui, A., Grimshaw, A., et al. (2005). *The Open Grid Services Architecture, Version 1.0*. Global Grid Forum.
- Foster, I., Yong Zhao Raicu, I., & Lu, S. (2008). *Cloud computing and Grid computing 360-degree compared*. Grid Computing Environments Workshop, GCE.
- Gallizo, G., Kubert, R., Oberle, K., Menychtas, A., Konstanteli, K. (2009). Service Level Agreements in Virtualized Service Platforms. eChallenges 2009. Istanbul, Turkey.
- García, J. M., Ruiz, D., Ruiz-Cortés, A., Martín-Díaz, O., & Resinas, M. (2007). *An hybrid, QoS-aware discovery of Semantic Web services using constraint programming*. 5th International Conference on Service-Oriented Computing, pp. 69-80. Vienna. Austria
- Goiri, Í., Julià, F., Fitó, J. O., Macías, M., & Guitart, J. (2010). Resource-level QoS metric for CPU-based guarantees in cloud providers. In *Economics of Grids, clouds, systems, and services, LNCS 6296*, pp. 34-47.
- GRIA: Service Oriented Collaborations for Industry and Commerce project, University of Southampton IT Innovation Centre. 2012. <http://www.gria.org/>.

- Gu, X., Nahrstedt, K., Yuan, W., Wichadakul, D., & Xu, D. (2002). An XML-based quality of service enabling language for the Web. *Journal of Visual Languages & Computing* , 13(1), pp. 61-95.
- Harchol-Balter, M., Sigman, K., Wierman, A. (2002). Understanding the slowdown of large jobs in an M/GI/1 system. *SIGMETRICS Perform. Eval.* pp. 9-11.
- Hiles, A. N. (1994). Service level agreements: Panacea or pain? *The TQM Magazine* , 6(2), pp. 14-16.
- Januszewski K., Mooney E., Harrah R., Lee S., Munter J, & Von Riegen C. (2002). UDDI Version 3 Features List, OASIS. http://uddi.org/pubs/uddi_v3_features.htm
- Kardara, M., Fuchs, O., Aisopos, F., Papaoikonomou, A., Tserpes, K., & Varvarigou, T. (2011). *A Service Oriented Architecture Enabling Policy Simulation in Virtual Spaces*. Third International Conference on Games and Virtual Worlds for Serious Applications, pp. 236 - 243. Athens: IEEE Computer Society.
- Kardara, M., Fuchs, O., Aisopos, F., Papaoikonomou, A., Tserpes, K., & Varvarigou, T. (2012). *A SOA based architecture for capturing public opinion in Virtual Spaces*. International Journal of Interactive Worlds (IJIW), Special Issue on “Serious Games and Interactive Worlds”. IBIMA.
- Kardara, M., Fuchs, O., Kosta, E., Aisopos, F., Tserpes, K., & Varvarigou, T. (2012). *Policy testing in virtual environments: addressing technical and legal challenges*. International Journal of Electronic Government Research (IJEGR), IGI Global.
- Kardara, M., Konstanteli, K., Aisopos, F., Andronikou, V., & Varvarigou, T. (2009). *A Subscription and Notification Mechanism for dynamic SLA Evaluation in GRIA*. eChallenges e-2009 Conference. Istanbul.

- Keller, A., & Ludwig, H. (2002). *The WSLA framework: Specifying and monitoring of service level agreements for Web services*. Research report, IBM.
- Kepes, B. (2011, March 9). *Understanding the Cloud Computing Stack: SaaS, PaaS, IaaS*. Ανάκτηση από:
http://broadcast.rackspace.com/hosting_knowledge/whitepapers/Understanding-the-Cloud-Computing-Stack.pdf
- Kim, H. J., Lee, D. H., Lee, J. M., Lee, K. H., Lyu, W., & Choi, S. G. (2008). *The QoE evaluation method through the QoS-QoE correlation model*. Fourth International Conference on Networked Computing and Advanced Information Management, NCM '08, pp. 719-725.
- Kyriazis, D. (2007). *Grid technologies with emphasis in dynamic workflows and quality of service*. Unpublished doctoral dissertation, National Technical University of Athens, Greece.
- Kyriazis, D., Tserpes, K., Menychtas, A., Sarantidis, I., & Varvarigou, T. (2009). Service selection and workflow mapping for Grids: An approach exploiting quality-of-service information. *Concurrency and Computation: Practice & Experience*, 21(6), pp.739-766.
- Lamanna, D., Skene, J., & Emmerich, W. (2003). *SLAng: A language for defining service level agreements*. The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03), pp. 100- 106.
- Lassila, O., & Swick, R. R. (1999). *Resource description framework (RDF) model and syntax*. Ανάκτηση από <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>

- Li-Yuan, L., Wen-An, Z., & Jun-De, S. (2006). *The research of quality of experience evaluation method in pervasive computing environment*. 1st International Symposium on Pervasive Computing and Applications, pp. 178-182.
- Litke, A., Skoutas, D., Tserpes, K., Varvarigou, T. (2007). Efficient Task Replication and Management for Adaptive Fault Tolerance in Mobile Grid environments. *Future Generation Computer Systems* 23, no. 2: pp. 163-178.
- Lo, N., & Wang, C.-H. (2007). *Web services QoS evaluation and service selection framework - A proxy-oriented approach*. IEEE Region 10 Conference TENCON 2007, pp. 1-5.
- Lodi, G., Panzieri, F., Ross, D., & Turrini, E. (2007, March). SLA-driven clustering of QoS-aware application servers. *IEEE Transactions On Software Engineering*, 33(3).
- Ma, Q., Wang, H., Li, Y., Xie, G., & Liu, F. (2008). *A semantic QoS-aware discovery framework for Web services*. IEEE International Conference on Web Service, ICWS '08, pp. 129-136.
- Masclat, D., & Pénard, T. (2008). *Is the eBay Feedback System Really Efficient? An Experimental Study*. University of Caen.
- Maximilien, E. M., & Singh, M. P. (2002). Conceptual model of web service reputation. *ACM SIGMOD Record*, 31(4), pp. 36 - 41.
- Maximilien, E. M., & Singh, M. P. (2004). *Toward autonomic Web services trust and selection*. 2nd International Conference on Service Oriented Computing, ICSOC '04, pp. 212-221. New York, NY: ACM.
- Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing*. Special Publication, National Institute of Standards and Technology.

- Menychtas, A. (2009). *Architectures and Mechanisms to evaluate quality of service in Service Oriented Architectures*. Unpublished doctoral dissertation, National Technical University of Athens, Greece.
- Molina-Jimenez, C., Shrivastava, S., Crowcroft, J., & Gevros, P. (2004). *On the monitoring of contractual service level agreements*. First IEEE International Workshop on Electronic Contracting, pp. 1- 8.
- Moorsel, A. V. (2001). *Metrics for the Internet age: Quality of experience and quality of business*. 5th Performability Workshop.
- Muñoz Frutos, H., Kotsiopoulos, I., Vaquero Gonzalez, L. M., & Rodero Merino, L. (2009). *Enhancing service selection by semantic QoS*. 6th European Semantic Web Conference on the Semantic Web: Research and Applications, pp. 565 - 577. Heraklion: LNCS.
- Nimbus: cloud computing for science. (2012, April 4). Ανάκτηση από <http://www.nimbusproject.org/about/>
- OGSA Primer (2008). Ανάκτηση από: *Open Grid Forum*. http://www.gridforum.org/OGSA_Primer
- Overton, C. (2002). On the theory and practice of Internet SLAs. *Journal of Computer Resource Measurement*, pp. 32–45.
- Papadakis, G., Tserpes, K., Sardis, M., Kardara, M., Papaoikonomou, A., & Aisopos, F. (2012). Social media meta-API: leveraging the content of social networks. *21st international conference companion on World Wide Web (WWW2012 Companion)* pp. 271-274. Lyon: ACM New York, USA.

- Parasuraman, A., Zeithaml, V. A., & Leonard, L. (1985). A conceptual model of service quality and its implications for future research. *The Journal of Marketing* , 49(4), 41-50.
- Patel, P., Ranabahu, A., & Sheth, A. (2009). *Service level agreement in cloud computing*. OOPSLA 2009, 24th ACM SIGPLAN. ACM.
- Pennock, D., Horvitz, E., Lawrence, S., & Giles, L. (2000). Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. *Sixteenth Conference on Uncertainty in Artificial Intelligence*. pp. 473-480. Morgan Kaufm.
- Picarello, F. (2011, April 20). Business, Cloud Computing – Just Right for Small. *Cadence Management Advisors*.
- Pirkul, H. (1987). A Heuristic Solution Procedure for the Multiconstraint Zero-One Knapsack Problem. Chap. 34 in *Naval Research Logistics*, pp. 161–172.
- Raimondi, F., Skene, J., & Emmerich, W. (2008). *Efficient online monitoring of Web-service SLAs*. 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering. Atlanta, GA: ACM.
- Ran, S. (2003). *A model for Web services discovery with QoS*. SIGecom Exch. pp. 1-10. ACM.
- ReliaSoft Corporation. (2006). *The Weibull Distribution*. Ανάκτηση από http://www.weibull.com/LifeDataWeb/the_weibull_distribution.htm.
- RenderMan Interface Specification v3.2. Pixar. 2005. https://renderman.pixar.com/products/rispec/rispec_pdf/RISpec3_2.pdf.
- Rosenberg, F., Platzer, C., & Dustdar, S. (2006). *Bootstrapping performance and dependability attributes of Web service*. ICWS.

- Sahai, A., Machiraju, V., Sayal, M., Jin, L. J., & Casati, F. (2002). *Automated SLA monitoring for Web services*. HP.
- Schlosser, M., Sintek, M., Decker, S., & Nejdl, W. (2002). *A scalable and ontology-based P2P infrastructure for Semantic Web services*. Second International Conference on Peer-to-Peer Computing (P2P'02), p. 104.
- Serhani, M., Dssouli, R., Hafid, A., & Sahraoui, H. (2005). *A QoS broker based architecture for efficient Web services selection*. IEEE International Conference on Web Services, ICWS 2005, pp. 113- 120.
- Skene, J., Raimondi, F., & Emmerich, W. (2009). Service-level agreements for electronic services. *IEEE Transactions on Software Engineering*, 99, 288-304.
- Snelling, D., & Anjomshoaa, A. (2007). *NextGRID Architectural Concepts*. CoreGRID Symposium, Rennes, France.
- Spillner, J., & Schill, A. (2009). *Dynamic SLA template adjustments based on service property monitoring*. IEEE Conference on Cloud Computing.
- StatAssist, *Intelligent Analytics*. (2009). Ανάκτηση από Mathwave, data analysis & simulation: <http://www.mathwave.com/products/assist.html>
- Sturm, R., Morris, W., & Jander, M. (2000). *Foundations of service level management*. SAMS Publishing.
- Takeuchi, H., & Quelch, J. (1983). Quality is more than making a good product. *Harvard Business Review*.
- Tian, M., Gramm, A., Naumowicz, T., Ritter, H., & Schiller, J. (2003). *A concept for QoS integration in Web service*. Fourth International Conference on Web Information Systems Engineering Workshops (WISEW'03), pp. 149-155. Rome. Italy

Tong, H., & Zhang, S. (2006). *A fuzzy multi-attribute decision making algorithm for Web services selection based on QoS*. IEEE Asia-Pacific Conference on Services Computing, APSCC '06. , pp. 51-57.

Torque Resource Manager. *Cluster Resources Inc.* 2010.

<http://www.clusterresources.com/products/torque-resource-manager.php>.

Tserpes, K. (2007). *Quality and Service Level Agreements Evaluation in Service Oriented Architectures*. Unpublished doctoral dissertation, National Technical University of Athens, Greece.

Tserpes, K., Aisopos, F., Kyriazis, D., & Varvarigou, T. (2010). Service selection decision support in the Internet of services. In J. Altmann, & O. Rana (Ed.), *GECON 2010. LNCS 6296*, (pp. 16-33). Springer.

Tserpes, K., Aisopos, F., Kyriazis, D., & Varvarigou, T. (2011). *A recommender mechanism for service selection in service-oriented environments*. Future Generation Computer Systems. ISSN 0167-739X, DOI: 10.1016/j.future.2011.11.003

Tserpes, K., Kyriazis, D., Menychtas, A., & Varvarigou, T. (2008). A novel mechanism for provisioning of high-level quality of service information in Grid environments. *European Journal of Operational Research*, 191(3), pp. 1113-1131.

Tserpes, K., Kyriazis, D., Menychtas, A., Litke, A., Christogiannis, C., & Varvarigou, T. (2008). *Evaluating quality provisioning levels in service oriented business environments*. 12th International IEEE Enterprise Distributed Object Computing Conference, pp. 309-315.

Tserpes, K., Papadakis, G., Kardara, M., Papaioikonomou, A., Aisopos, F., Sardis, M., & Varvarigou, T. (2012, October). An Ontology for Social Networking Sites

Interoperability. 4th International Conference on Knowledge Engineering and Ontology Development (KEOD '12). Barcelona, Spain

Varvarigou, T., Tserpes, K., Kyriazis, D., Silvestri, F., & Psimogiannos, N. (2010). A study on the effect of application and resource characteristics on the QoS in service provisioning environments. *International Journal of Distributed Systems and Technologies*, 1(1), 55-75.

Vu, L. H., Hauswirth, M., & Aberer, K. (2005). *Towards P2P-based Semantic Web service discovery with QoS support*. Workshop on Business Processes and Services (BPS). Nancy.

Vu, L.-H., Hauswirth, M., & Aberer, K. (2005). *QoS-based service selection and ranking with trust and reputation management*. *CoopIS; DOA; ODBASE. On the move to meaningful Internet systems*. Springer.

Waheed Iqbal, M. D. (2009). *SLA-driven adaptive resource management for Web applications on a heterogeneous compute cloud*. Lecture Notes in Computer Science (LNCS).

Wang, H.-C., Lee, C.-S., & Ho, T.-H. (2007). Combining subjective and objective QoS factors for personalized Web service selection. *Expert Systems with Applications*, 32(2), 571-584.

Wang, X., Vitvar, T., Kerrigan, M., & Toma, I. (2006). A QoS-aware selection model for semantic Web Services. In A. Dan, & W. Lamersdorf (Ed.), *ICSOC 2006. LNCS 4294*, (pp. 390–401). Springer.

- Wang, Y., & Vassileva, J. (2007). *A review on trust and reputation for Web service selection*. 27th International Conference on Distributed Computing Systems Workshops, ICDCSW '07, (p. 25).
- Wang, Z., Xiaoyun, Z., Pradeep, P., Sharad, S. (2007). Capacity and Performance Overhead in Dynamic Resource Allocation to Virtual Containers. *10th IFIP/IEEE Symposium on Integrated Management*. Munich, Germany.
- Williams, A. (2010, July 26). SaaS Market Grows Spectacularly to the Detriment of the IT Kings. Ανάκτηση από <http://www.readwriteweb.com/cloud/2010/07/sass-providers-challenge-the-k.php>
- Wolski, R. (2003). Ανάκτηση από CS290I Lecture notes -- Globus: "The" Grid Programming Toolkit: <http://www.cs.ucsb.edu/~rich/class/cs290I-grid/notes/Globus/>
- Wu, C., Chang, E., & Thomson, P. (2005). A decision support system for QoS-enabled distributed Web services architecture. In H. Hess, L. Franquelo, A. Malinowski, & M. Chow (Ed.), *Industrial Electronics Society Conference, IECON 2005*.
- Xu, Z., Martin, P., Powley, W., & Zulkernine, F. (2007). *Reputation-enhanced QoS-based Web services discovery*. IEEE International Conference on Web Services, ICWS 2007, (pp. 249-256).
- Zhang, F., Bai, L., & Gao, F. (2009). *A User Trust-Based Collaborative Filtering Recommendation Algorithm*. Lecture Notes in Computer Science, 5927, pp. 411-424.
- Zhang, Y., Zheng, Z., & Lyu, M. R. (2010). *WSExpress: A QoS-aware search engine for Web services*. IEEE International Conference on Web Services (ICWS), pp. 91-98.