



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

**Αξιολόγηση και Βελτίωση Ποιότητας Υπηρεσίας και Εμπειρίας σε
Συστήματα Υπολογιστικού Νέφους**

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Αλέξανδρος Ε. Ψύχας

ΑΘΗΝΑ, ΜΑΡΤΙΟΣ 2022



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

Αξιολόγηση και Βελτίωση Ποιότητας Υπηρεσίας και Εμπειρίας σε Συστήματα Υπολογιστικού Νέφους

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Αλέξανδρος Ε. Ψύχας

Συμβουλευτική Επιτροπή : Θεοδώρα Βαρβαρίγου (Επιβλέπουσα)

Συμεών Παπαβασιλείου

Εμμανουήλ Βαρβαρίγος

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 17^η Μαρτίου 2022.

.....
Θεοδώρα Βαρβαρίγου
Καθ. Ε.Μ.Π.

.....
Συμεών Παπαβασιλείου
Καθ. Ε.Μ.Π.

.....
Εμμανουήλ Βαρβαρίγος
Καθ. Ε.Μ.Π.

.....
Κωνσταντίνος Τσερπές
Αν.Καθ. Χαροκόπειο Παν.

.....
Αναστάσιος Δουλάμης
Αν.Καθ. Ε.Μ.Π.

.....
Δημήτριος Ασκούνης
Καθ. Ε.Μ.Π.

.....
Ιωάννης Ψαρράς
Καθ. Ε.Μ.Π.

ΑΘΗΝΑ, Μάρτιος 2022.

Η σελίδα αυτή είναι σκόπιμα λευκή

.....
Αλέξανδρος Ε. Ψύχας

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

Copyright © Αλέξανδρος Ε. Ψύχας, 2022.

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.

Η σελίδα αυτή είναι σκόπιμα λευκή

Πρόλογος

Η διδακτορική διατριβή που παρουσιάζεται στις επόμενες σελίδες εκπονήθηκε από τον Φεβρουάριο του 2014 μέχρι τον Μάρτιο του 2022, στο εργαστήριο κατανεμημένων συστημάτων του τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής, στη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου.

Κατά τη διάρκεια της εκπόνησης της διατριβής, είχα την ευκαιρία να ασχοληθώ με αρκετά ενδιαφέροντα επιστημονικά θέματα που αφορούν κυρίως στους τομείς των συστημάτων υπολογιστικού νέφους, της αρχιτεκτονικής σχεδίασης κατανεμημένων συστημάτων, καθώς και την υλοποίηση συστημάτων λογισμικού για Cloud.

Θα ήθελα να ευχαριστήσω από τα βάθη της καρδιάς μου, την καθηγήτρια μου κ. Θεοδώρα Βαρβαρίγου για το ενδιαφέρον που έδειξε, για την πολύτιμη καθοδήγησή της και για την ιδιαίτερη στήριξη που μου παρείχε κατά την διάρκεια αυτής της πορείας μου, καθώς επίσης τους καθηγητές της τριμελούς συμβουλευτικής επιτροπής κ.κ. Συμεών Παπαβασιλείου και Εμμανουήλ Βαρβαρίγο.

Επίσης, θα ήθελα να ευχαριστήσω όλους τους συναδέλφους και μέντορες μου στην ερευνητική ομάδα με τους οποίους συνεργάστηκα όλα αυτά τα χρόνια. Ιδιαίτερες ευχαριστίες αρμόζουν στους συνεργάτες και μέντορες, Αναπληρωτή Καθηγητή Δημοσθένη Κυριαζή, Επίκουρο Καθηγητή Γεώργιο Κουσιουρή και Αντόνιο Λίτκε.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την αμέριστη και ανιδιοτελή προσφορά και στήριξή τους, καθώς και τον αδερφό μου Κωσταντίνο που μέσα από τις πράξεις του μου έδειξε τη σημαίνει σκληρή δουλειά και αφοσίωση.

“Archimedes will be remembered when Aeschylus is forgotten, because languages die and mathematical ideas do not. “Immortality” may be a silly word, but probably a mathematician has the best chance of whatever it may mean.

-- G.H. Hardy, A Mathematician's Apology. --

Αλέξανδρος Ε. Ψύχας

Μάρτιος 2022

Η σελίδα αυτή είναι σκόπιμα λευκή

Πίνακας Περιεχομένων

Περίληψη.....	1
Abstract	2
1 Εισαγωγή.....	3
1.1 Καινοτομία και συνεισφορά διατριβής.....	6
1.2 Οργάνωση κειμένου.....	8
2 Βιβλιογραφική και τεχνολογική επισκόπηση συστημάτων υπολογιστικού νέφους	12
2.1 Συμφωνία επιπέδου ποιότητας υπηρεσίας(SLA) στα υπολογιστικά νέφη(Cloud)	15
2.2 Συγκριτική αξιολόγηση(Benchmarking) υπηρεσιών υπολογιστικού νέφους.....	20
2.3 Προφίλ αναγκών και κατηγοριοποίηση λογισμικού	23
2.3.1 Προσαρμογή και πρόβλεψη μοντέλων παραλληλοποίησης εικονικών μηχανών (VM)	23
2.3.2 Ελαχιστοποίηση πτώσης απόδοσης συστήματος υπολογιστικού νέφους	24
2.3.3 Αποτελεσματικά μοτίβα κατανομής με βάση μοντέλων βέλτιστης παραλληλοποίησης.....	25
2.4 Αξιολόγηση ποιότητας υπηρεσιών υπολογιστικού νέφους προσανατολισμένες σε υπολογιστές υψηλών επιδόσεων(HPC).....	29
3 Σύλληψη αρχιτεκτονικής και δομικών εννοιών συστήματος.....	32
3.1 Ανάλυση ρόλων και στόχων συστήματος για την δημιουργία της αρχιτεκτονικής	33
3.1.1 Ανάλυση στόχων και χρηστών συστήματος	33
3.1.2 Ανάλυση δομικών στοιχείων αρχιτεκτονικής	35
3.2 Πλήρης αρχιτεκτονική συστήματος δομημένη σε λειτουργικές ομάδες.....	37
3.2.1 Σύνολο εργαλείων παρόχου IaaS(IPT).....	38
3.2.2 Σύνολο εργαλείων QoE Entity(QET).....	41
3.2.3 Σύνολο εργαλείων για Cloud Adopter(CAT).....	45
3.3 Αλληλεπίδραση μεταξύ στοιχείων του συστήματος	49
4 Υλοποίηση συστήματος αξιολόγησης και βελτίωσης υπηρεσιών υπολογιστικού νέφους.....	54
4.1 Υλοποίηση υποσυστήματος ελέγχου επιπέδου ποιότητας υπηρεσίας (SLA monitoring subsystem).	61
4.1.1 Προδιαγραφές και τυποποίηση αξιολόγησης Cloud SLA	61
4.1.2 Τεχνική υλοποίηση(λογισμικό QoE,3ALib)	73

4.2	Υλοποίηση υποσυστήματος συγκριτικής αξιολόγησης (QoE,BenchSuite)	80
4.2.1	Τυποποίηση μετρικών Benchmarking.....	80
4.2.2	Τεχνική υλοποίηση(λογισμικό QoE, Bench Suite)	85
4.3	Εργαλείο δημιουργίας προφίλ αναγκών και κατηγοριοποίησης λογισμικού.	95
4.3.1	Ανάλυση και θεωρητική προσέγγιση	95
4.3.2	Ανάλυση αρχιτεκτονικής συστήματος δημιουργίας προφίλ λογισμικού	99
4.4	Μοντέλα παρεμβολών και κατανομής πόρων.	103
4.4.1	Ανάλυση τεχνικής υλοποίηση των ANN δικτιών	106
5	Αξιολόγηση λειτουργίας και απόδοσης υποσυστημάτων	112
5.1	Αξιολόγηση λειτουργικότητας και αποδοτικότητας υποσυστήματος QoE(3ALib, Bench Suite)	113
5.1.1	Αξιολόγηση σταθερότητας διεργασιών συστήματος	113
5.1.2	Αξιολόγηση αποδοτικότητας συστήματος.....	118
5.2	Αξιολόγηση ακρίβειας μοντέλων ANN.....	120
5.2.1	Συλλογή δεδομένων και αξιολόγηση μοντέλων παρεμβολών	122
5.2.2	Αξιολόγηση αποδοτικότητας και ακρίβειας μοντέλων κατανομής πόρων ...	131
6	Συμπεράσματα και μελλοντικές ερευνητικές κατευθύνσεις	139
6.1	Πρόβλεψη απόδοσης εφαρμογών Cloud	140
6.2	Πρόβλεψη και μείωσης του κόστους λειτουργίας Cloud	141
6.3	Βελτίωση της αξιολόγησης και επιλογής παρόχου Cloud.....	142
7	Συντομογραφίες.....	145
8	Βιβλιογραφικές Αναφορές	147

Εικόνες

Εικόνα 1: Καινοτομία και όραμα έρευνας/διατριβής.....	7
Εικόνα 2: Βιβλιογραφική και τεχνολογική προσέγγιση διατριβής.	15
Εικόνα 3: Συσχέτιση μεταξύ ρόλων και στόχων συστήματος.	37
Εικόνα 4: Αρχιτεκτονική συστήματος[3].....	38
Εικόνα 5: Διαγραμμαμα αντικειμένων συστήματος.	50
Εικόνα 6: Αφαιρετικού επιπέδου διαλειτουργικότητα QoE-3ALib.....	73
Εικόνα 7: Φόρμα παραμετροποίησης εργαλείου 3ALib.	74
Εικόνα 8: Λειτουργία SLA Analytics για αξιολόγηση αυστηρότητας δημόσιων SLA.....	77
Εικόνα 9: Αποτελέσματα μηνιαίων ελέγχων SLA διαθεσιμότητας.	78
Εικόνα 10: Αποτελέσματα ελέγχου CPU_Utilization για SLA κλιμάκωσης εφαρμογής.	79
Εικόνα 11: Αφαιρετικού επιπέδου διαλειτουργικότητα QoE-Bench Suite.....	85
Εικόνα 12: Διεπαφή παραμετροποίησης και εκτέλεσης Benchmark.	86
Εικόνα 13: Απεικόνιση αποτελεσμάτων εκτελεσμένων Benchmark μέσω του QoE.	88
Εικόνα 14: Απεικόνιση ιδιωτικών εκτελεσμένων Benchmark μέσω του QoE.....	89
Εικόνα 15: Συγκεντρωτικά Analytics σε μορφή ραβδογράμματος.....	89
Εικόνα 16: Συγκεντρωτικά Analytics απόδοσης/κόστους σε μορφή ραβδογράμματος.	91
Εικόνα 17: Επεξεργασία μοντέλου κόστους υπηρεσιών.....	92
Εικόνα 18: Διαδικασία παραμετροποίησης και εκτέλεσης Benchmark.....	94
Εικόνα 19: Αρχιτεκτονική αφαιρετικού επιπέδου εργαλείου δημιουργίας προφίλ.	99
Εικόνα 20: Διεπαφή παραμετροποίησης εργαλείου εξαγωγής προφίλ.	101
Εικόνα 21: Διεπαφή κατηγοριοποίησης εφαρμογών.....	102
Εικόνα 22: Σύσταση αλγορίθμων ANN για τα μοτίβα κατανομής και παρεμβολών.	104
Εικόνα 23: Νεύρο-εξελικτική διαδικασία εκμάθησης ANN[5].	107
Εικόνα 24: Διεπαφή χρήστη για επιλογή μοντέλου[5].....	111
Εικόνα 25: Συνολικός χρόνος υπολογισμού SLA 200 χρηστών[4].	119
Εικόνα 26: Μέσος χρόνος υπολογισμού SLA ανά χρήστη[4].	120
Εικόνα 27: Κάθετη δομή της εφαρμογής CRM/ERP.....	121
Εικόνα 28: Παράδειγμα κλιμάκωσης εφαρμογής CRM/ERP.....	121
Εικόνα 29: Ποσοστιαία υποβάθμιση απόδοσης του DaCapo tomcat σε συνδυασμό όλων των Benchmarks[4].	127
Εικόνα 30: Ποσοστιαία υποβάθμιση απόδοσης του DaCapo Eclipse σε συνδυασμό όλων των Benchmarks[4].	127
Εικόνα 31: Ποσοστιαία υποβάθμιση απόδοσης του DaCapo web-proxy σε συνδυασμό όλων των Benchmarks[4].	128
Εικόνα 32: Τρισδιάστατη απεικόνιση των παραμέτρων του microservice των πελατών[5].	135
Εικόνα 33: Ραβδόγραμμα συγκριτικής απόδοσης ML αλγορίθμων.	138

Πίνακες

Πίνακας 1: Αντιστοίχιση στόχων χρηστών με λειτουργίες συστήματος για QoE Toolkit. ..	57
Πίνακας 2: Αντιστοίχιση στόχων χρηστών με λειτουργίες συστήματος για τα εργαλεία Profiling and Classification.	59
Πίνακας 3: Αντιστοίχιση στόχων χρηστών με λειτουργίες συστήματος για τα εργαλεία Interference and Mapping Models.....	60
Πίνακας 4: Ανάλυση μετρικών με βάση αναγκών χρήστη/παρόχου Cloud Service.....	66
Πίνακας 5: Ανάλυση απόδοσης VM με βάση αναγκών χρήστη/παρόχου Cloud Service. ...	83
Πίνακας 6: Σύνολο μετρικών για την δημιουργία αποτυπώματος εφαρμογής.	97
Πίνακας 7: Λίστα αξιολογημένων διεργασιών για την σταθερότητα συστήματος QoE. ...	115
Πίνακας 8: Αποτελέσματα αξιολόγησης σταθερότητας διεργασιών QoE.....	117
Πίνακας 9: Αναφορά κατηγοριοποίησης λογισμικού CRM/ERP σε Benchmarks.	123
Πίνακας 10: Διαμόρφωση μοντέλων παρεμβολών.	129
Πίνακας 11: Χαρακτηριστικά πλατφόρμας υλοποίησης μοντέλων παρεμβολών.....	130
Πίνακας 12: Παράμετροι σεναρίων χρήσης εφαρμογής CRM.	132
Πίνακας 13: Μνήμη RAM(MB) των κοντέινερ (Docker container) ανά υπηρεσία/batch..	133
Πίνακας 14: Σύνολο δεδομένων που παράχθηκε από το testbed.	134
Πίνακας 15: Αναλυτική συγκριτική αξιολόγησης MAPE.	136
Πίνακας 16: Παράμετροι του νευρωνικού δικτύου όπου επιλέχτηκαν από τον GA.....	137

Περίληψη

Το Cloud έχει μια πληθώρα από χαρακτηριστικά που είναι απαραίτητα για τους κατόχους εφαρμογών ώστε να διασφαλίσουν την ποιότητα υπηρεσίας, την σταθερότητα καθώς και την κλιμάκωσή τους. Λόγω αυτών των χαρακτηριστικών αποτελεί την ντε φάκτο λύση υποδομής για την εγκατάσταση και λειτουργία μιας πληθώρας από εφαρμογές. Η ανάπτυξη μιας εφαρμογής σε ένα περιβάλλον Cloud ωστόσο, θέτει μια σειρά από σοβαρές προκλήσεις, κυρίως όσον αφορά τη διαδικασία επιλογής παρόχου και πόρων, με βάση την αναμενόμενη ποιότητα υπηρεσίας, καθώς και τη διαχείριση των εικονικών πόρων στις εγκαταστάσεις του παρόχου. Η διατριβή αυτή επιχειρεί να προσεγγίσει και να επιλύσει αυτά τα θεμελιώδη προβλήματα που αφορούν την υιοθέτηση λύσεων Cloud. Συγκεκριμένα, επιχειρεί να αντιμετωπίσει αυτά τα ζητήματα παρέχοντας μια εξελιγμένη αρχιτεκτονική και ένα σύνολο εργαλείων λογισμικού που βοηθά τόσο τον χρήστη του Cloud όσο και τον πάροχο, όσον αφορά το προφίλ και την κατηγοριοποίηση εφαρμογών, την ανάλυση και την πρόβλεψη των επιπτώσεων παρεμβολών σε παραλληλοποιημένες εφαρμογές, την ανάλυση και διαχείριση των πόρων, τη συγκριτική αξιολόγηση και την ανάλυση και αποσαφήνιση της ποιότητας εμπειρίας σε υποδομές Cloud. Αναφορικά με τις παρεμβολές και διαχείριση πόρων αναπτύχθηκαν μοντέλα βασισμένα σε τεχνητά νευρωνικά δίκτυα, που αναλύουν δεδομένα καταγραφής για να βοηθήσουν τους παρόχους να διαχειριστούν καλύτερα τους εικονικούς πόρους. Η ποιότητα υπηρεσίας και εμπειρίας καθώς και η αξιολόγηση των Cloud υποδομών επικεντρώθηκε στην ανάπτυξη λογισμικού για την αξιολόγηση του επιπέδου ποιότητας υπηρεσίας, καθώς και στην τυποποίηση των μετρικών και διαδικασιών της συγκριτικής αξιολόγησης των Cloud υπηρεσιών.

Λέξεις κλειδιά: Υπολογιστικό Νέφος, Ποιότητα υπηρεσίας, Κατανεμημένα συστήματα

Abstract

Cloud has a number of features that are essential for application owners to ensure quality of service, stability and scalability. Because of these features it comes across as the de facto infrastructure solution for the installation and operation of a variety of applications. Developing an application in a Cloud environment, however, poses a number of serious challenges, mainly in terms of the provider and resource selection process, based on the expected quality of service, and the management of virtual resources at the provider's premises. This dissertation attempts to approach and solve these fundamental problems related to the adoption of Cloud solutions. In particular, it seeks to address these issues by providing a sophisticated architecture and set of software tools that assist both the Cloud user and the provider, in terms of application profiling and categorization, analysis and prediction of interference effects in parallel running applications, resource analysis and management, benchmarking and analysis and clarification of the quality of experience in Cloud infrastructure. In terms of interference and resource management, models based on artificial neural networks have been developed, which they analyze log data to help providers better manage virtual resources. Service quality and experience as well as the general evaluation of Cloud infrastructure focused on the development of software for the evaluation of the service level agreement quality as well as the standardization of the benchmarking processes and metrics.

Keywords: Cloud Computing, Quality of Service, Distributed Systems

1 *Εισαγωγή*

Οι περισσότερες εφαρμογές λογισμικού δημιουργούν μια σειρά από σημαντικές ερωτήσεις πριν από τη μετεγκατάστασή τους σε μια υποδομή που βασίζεται στο Cloud[1]. Πώς θα εκτελείται η εφαρμογή σε περιβάλλον πολλαπλών ενοικιαστών· ποιοι τύποι υπηρεσιών/παρόχων θα πρέπει να επιλεγούν για την υλοποίηση ή τη φιλοξενία της, σε ποιους παρόχους και με βάση ποια κριτήρια· πώς πρέπει ο πάροχος εφαρμογών να χρεώνει τους πελάτες του και με βάση ποιες παραμέτρους και εκτιμήσεις σε σχέση με το προσφερόμενο QoS (Quality of Service (ποιότητα υπηρεσίας)); Πώς μπορεί κανείς να επιλέξει σε ποιον πάροχο θα μετεγκατασταθεί, καθώς και σε ποιον τύπο υπηρεσίας αυτού του παρόχου; Είναι το προσφερόμενο QoS επαρκές για τις ανάγκες της εφαρμογής, είναι ικανοποιητικό ή και μπορεί να προβλεφθεί ή να υπολογιστεί πριν την μετεγκατάσταση της εφαρμογής; Εάν η απόδοση των εικονικών πόρων συνεχίζει να αλλάζει, τότε θα αλλάξει και ο αριθμός τους που χρησιμοποιούνται για τη διατήρηση του QoS που προσφέρεται στους πελάτες της εφαρμογής. Επιπροσθέτως, τα ζητήματα ταχύτητας και καθυστέρησης έχουν αναγνωριστεί ως το κορυφαίο εμπόδιο σε αυτόν τον τομέα, ενώ το κόστος και η αξιοπιστία είναι ο κορυφαίος και δεύτερος σημαντικότερος παράγοντας για την αξιολόγηση των υπηρεσιών Cloud [2]. Ειδικά η απόδοση μπορεί να εμποδίσει την

προσαρμογή των λύσεων Cloud για μια ποικιλία εφαρμογών (π.χ. στην περίπτωση προσομοίωσης Computational Fluid Dynamics), για τις οποίες η έγκαιρη ολοκλήρωση των διεργασιών της είναι το κλειδί για την παραγωγικότητα της εταιρείας. Τουτέστιν, σε αυτές τις περιπτώσεις, προτιμώνται εξειδικευμένες και δαπανηρές λύσεις έναντι των διαφόρων υποδομών Cloud, γεγονός που δημιουργεί πρόσθετο κόστος ή αποτρέπει ένα ευρύ σύνολο των SME(Small Medium Enterprise(μικρό-μεσαίες επιχειρήσεις)) από το να είναι ανταγωνιστικές ή ακόμη και λειτουργικές.

Μετά την ανάπτυξη της εφαρμογής στο Cloud, προκύπτουν ένα νέο σύνολο ερωτημάτων σχετικά με την κατάσταση λειτουργίας της. Ο αντίστοιχος επιλεγμένος πάροχος καλύπτει μέρος του σε σχέση με τις εγγυημένες μετρήσεις SLA(Service Level Agreement(συμφωνία επιπέδου ποιότητας υπηρεσίας)); Λαμβάνει ο πάροχος διορθωτικά ή/και προληπτικά μέτρα προκειμένου να εγγυηθεί το QoS, τη σταθερότητα των υπηρεσιών του και την QoE(Quality of Experience(ποιότητας εμπειρίας)) που προσφέρει στους πελάτες του; Διατηρούν οι πάροχοι τις ρητές συμφωνίες επιπέδου υπηρεσιών; Ωστόσο, οι εταιρείες συχνά δεν μπορούν να απαντήσουν σε αυτές τις ερωτήσεις πριν από την ανάπτυξη και εγκατάσταση των εφαρμογών στο Cloud, πράγμα που σημαίνει ότι οποιαδήποτε κακή λήψη αποφάσεων είναι πρακτικά μη αναστρέψιμη, καθώς θα δημιουργούσε επιπλέον κόστος μετάβασης και επαναδιαμόρφωση σε άλλο πάροχο. Η μετάβαση σε μια υποδομή Cloud μπορεί να αποφέρει μείωση κόστους λειτουργίας, καλύτερη απόδοση και στρατηγικά πλεονεκτήματα, αλλά θα πρέπει να γίνεται με τη δέουσα επιμέλεια και προσεκτική εξέταση.

Επιπλέον, στο επίπεδο παρόχου IaaS(Infrastructure as a Service(υποδομή ως υπηρεσία)) προκύπτει ένας σημαντικός αριθμός προκλήσεων που έχουν να κάνουν κυρίως με τη δυναμικότητα των εξεταζόμενων εφαρμογών και τον φόρτο εργασίας τους, την αδυναμία

πληροφόρησης για το εσωτερικό λογισμικό της εφαρμογής του πελάτη που εκτελείται στους εικονικούς πόρους και την ανάγκη για αναβάθμιση του διακομιστή προκειμένου να ελαχιστοποιηθεί το κόστος, διατηρώντας παράλληλα τα εγγυημένα SLA, για την πλευρά του πελάτη. Ιδιαίτερα για τα SLA, λόγω αυτών των προκλήσεων οι πάροχοι πρακτικά αναγκάζονται να εκδίδουν SLA στο βασικό επίπεδο IaaS που σχετίζεται μόνο με τη διαθεσιμότητα και όχι π.χ. σχετικά με την απόδοση των εικονικών CPU. Επιπλέον, κάθε φορά που ο πάροχος χρειάζεται να χρησιμοποιήσει εξωτερικούς πόρους, είτε λόγω περιορισμένης εσωτερικής χωρητικότητας είτε λόγω βελτιστοποίησης της λειτουργίας μιας υπηρεσίας, αντιμετωπίζουν και αυτοί τα ίδια ερωτήματα που αντιμετωπίζουν οι αρχικοί κάτοχοι εφαρμογών.

Επομένως, οι ακόλουθες προκλήσεις είναι κρίσιμες προκειμένου να απαντηθούν οι προαναφερθείσες ερωτήσεις, τόσο από την πλευρά του παρόχου του Cloud όσο και από την πλευρά του χρήστη- υιοθέτη :

- Η αστάθεια στην απόδοση των υπηρεσιών IaaS λόγω του δυναμικού φόρτου εργασίας, των υποδομών και των υπηρεσιών πολλαπλών μισθωτών δημιουργούν ένα διαρκώς μεταβαλλόμενο οικοσύστημα, που πρέπει ωστόσο να προσφέρει σταθερότητα στους πελάτες του SaaS(Software as a Service(λογισμικό ως υπηρεσία)) για συνεπή τιμολόγηση στους αντίστοιχους τελικούς χρήστες τους.
- Η άγνοια των χαρακτηριστικών των φυσικών υποδομών ή των στρατηγικών διαχείρισης πόρων του παρόχου από την πλευρά του υιοθέτη, που οδηγεί σε μη ενημερωμένη και μη βέλτιστη επιλογή παρόχου.
- Άγνοια δομών και χαρακτηριστικών εφαρμογών πελατών από την πλευρά του παρόχου που τους απαγορεύει να εφαρμόζουν ωφέλιμες στρατηγικές διαχείρισης.

- Έλλειψη κατάλληλων μηχανισμών παρακολούθησης SLA για δυνατότητα ελέγχου των Cloud SLA και σημαντικές διαφορές σε βασικές πτυχές, όπως εγγυημένες μετρήσεις για διαφορετικούς τύπους υπηρεσιών μεταξύ παρόχων για τους υιοθέτες των υπηρεσιών αυτών, με αποτέλεσμα την αδυναμία άμεσης σύγκρισης των προσφορών παρόχων σε σχέση με τις προσφερόμενες εγγυήσεις.
- Έλλειψη κατάλληλα εφαρμοσμένων μετρήσεων που μπορούν να περιγράψουν την απόδοση του παρόχου σε κατανοητό επίπεδο για τον μέσο χρήστη του Cloud, χωρίς να χρειάζεται να είναι ειδικοί στα συστήματα ή συγκεκριμένα στο Cloud Computing.
- Ετερογένεια στα API των IaaS που προκαλούν επιπρόσθετο φόρτο εργασίας για την προσαρμογή των χρηστών σε αυτά σε περίπτωση μετεγκατάστασης.

1.1 Καινοτομία και συνεισφορά διατριβής

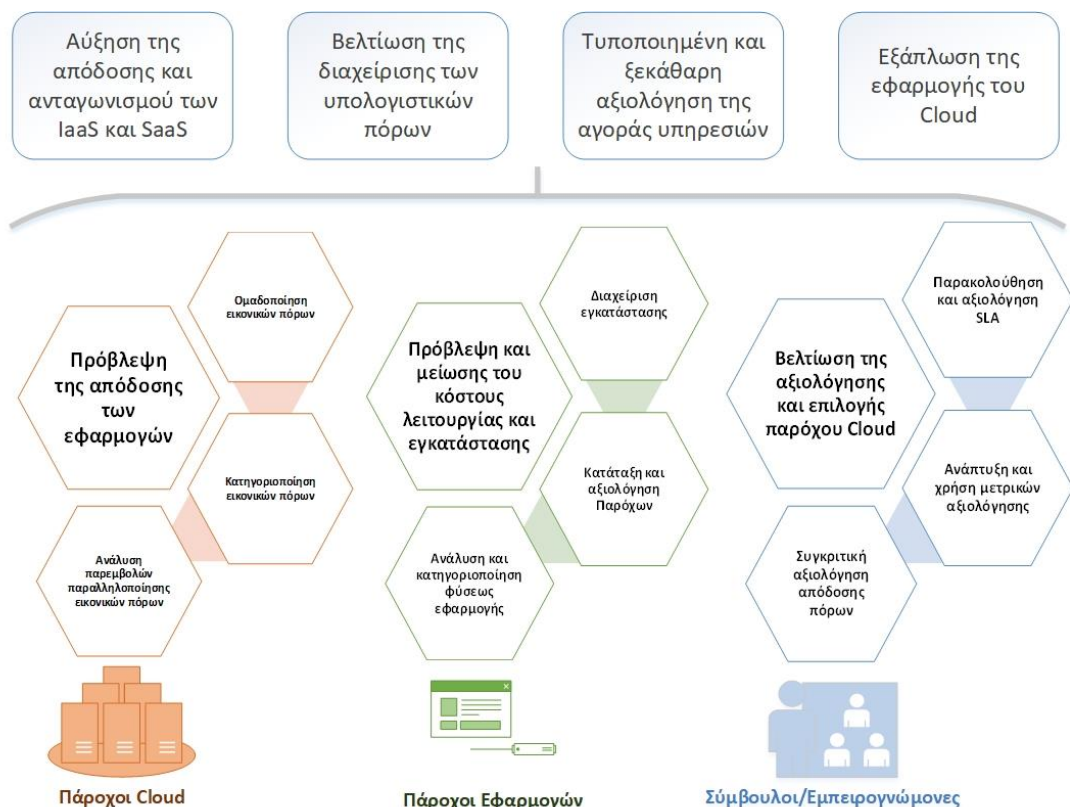
Καθώς το Cloud μετατρέπεται στην ντε φάκτο υποδομή για την εγκατάσταση και λειτουργία μιας πληθώρας εφαρμογών και λογισμικού, παρατηρείται ότι υπάρχουν προβλήματα που αφορούν όλους τους παράγοντες που το συναποτελούν (παρόχους υποδομών, χρήστες ή υιοθέτες, συμβούλων-εμπειρογνομόνων). Το όραμα της έρευνας και κατ' επέκτασης της διατριβής αυτής είναι η δημιουργία μιας πρότυπης αρχιτεκτονικής που αποτελείται από ένα συνονθύλευμα από εργαλεία λογισμικού όπου έχουν ως σκοπό την επίλυση των προαναφερθέντων προβλημάτων με απώτερο σκοπό(Εικόνα 1):

- Την αύξηση της απόδοσης και ανταγωνισμού των IaaS και SaaS
- Την βελτίωση της διαχείρισης των υπολογιστικών πόρων των χρηστών Cloud
- Την τυποποιημένη και ξεκάθαρη αξιολόγηση της αγοράς υπηρεσιών Cloud

- Την εξάπλωση της εφαρμογής του Cloud

Παράλληλα με την δημιουργία της πρότυπης αρχιτεκτονικής αναπτύχθηκαν ή εξελίχθηκαν τα κομμάτια λογισμικού που συνιστούν την αρχιτεκτονική αυτή (πλήρης ανάλυση υλοποίησης Κεφάλαιο 4). Η ανάπτυξη των εργαλείων αυτών έχουν ως απώτερο σκοπό:

- Την πρόβλεψη της απόδοσης των εφαρμογών στο περιβάλλον Cloud
- Την πρόβλεψη και μείωσης του κόστους λειτουργίας και εγκατάστασης του λογισμικού στο Cloud
- Την βελτίωση της αξιολόγησης και επιλογής παρόχου Cloud



Εικόνα 1: Καινοτομία και όραμα έρευνας/διατριβής.

Η ερευνητική δουλειά που έγινε στο πλαίσιο της διατριβής στους συγκεκριμένους τομείς τεχνολογιών Cloud είχε ως αποτέλεσμα έναν αριθμό από δημοσιεύσεις. Συγκεκριμένα στο κομμάτι της γενικής αρχιτεκτονικής του συστήματος αξιολόγησης και βελτίωσης ποιότητας υπηρεσίας στο Cloud, έγινε δημοσίευση που παρουσιάζει αυτή την πλήρη αρχιτεκτονική στο 2017 International Conference on High Performance Computing & Simulation (HPCS) υπό την αιγίδα της IEEE [3]. Για τα αναπτυχθέντα εργαλεία αξιολόγησης Cloud υπηρεσιών έγινε η αντίστοιχη δημοσίευση στο FGCS Journal [4]. Τέλος, για τα μοντέλα τεχνητών νευρωνικών δικτύων που έχουν σκοπό την βελτίωση της υπηρεσίας από την πλευρά του IaaS έγινε δημοσίευση στο συνέδριο CLOSER το 2019 [5].

1.2 Οργάνωση κειμένου

Η παρούσα διατριβή διαρθρώνεται από 8 κεφάλαια, η ακολουθία καθώς και το περιεχόμενο των οποίων εμφανίζεται ως εξής:

Το **κεφάλαιο 2** παρουσιάζει την βιβλιογραφική και τεχνολογική επισκόπηση που αφορά λογισμικό, τυποποιήσεις δεδομένων και στάνταρτ που έχουν δημιουργηθεί για την έρευνα και εξέλιξη του τομέα της πληροφορικής Cloud Computing. Συγκεκριμένα, η επισκόπηση αφορά την ανάλυση των ορισμών και των τυποποιήσεων για την συμφωνία επιπέδου υπηρεσίας(SLA), καθώς και το πώς το υπάρχον λογισμικό για την αξιολόγηση και παρακολούθηση υπηρεσιών Cloud υστερεί από μηχανισμούς για την παρακολούθηση των SLA. Επιπροσθέτως, αναλύεται σε αυτό το κεφάλαιο το πώς η χρήση λογισμικού συγκριτικής αξιολόγησης (Benchmarks) έχει εξελιχθεί και χρησιμοποιείται κατά κόρον για την ανάλυση της απόδοσης των εικονικών πόρων του Cloud. Τέλος, σημαντικό κομμάτι της επισκόπησης αυτού του κεφαλαίου αποτελεί η ερευνητική επισκόπηση των

τεχνικών και των τεχνολογικών επιτευγμάτων στον τομέα της ανάλυσης των αναγκών καθώς και των μεθόδων για την κατηγοριοποίηση των εφαρμογών.

Το **κεφάλαιο 3** παρουσιάζει τη σύλληψη της αρχιτεκτονικής καθώς και των δομικών εννοιών του συστήματος για την αξιολόγηση και την βελτίωση των Cloud συστημάτων. Αρχικά, γίνεται μια ανάλυση των ρόλων(τύπους χρηστών) καθώς και των στόχων των διαφόρων χρηστών του συστήματος ώστε να γίνουν πιο κατανοητές οι επιλογές σχεδίασης του συστήματος. Στη συνέχεια, παρατίθεται η αρχιτεκτονική, δομημένη σε λειτουργικές ομάδες και αναλύονται τα ξεχωριστά κομμάτια λογισμικού που την συντελούν καθώς και το πως συνδέονται και αλληλοεπιδρούν.

Το **κεφάλαιο 4** παρουσιάζει την υλοποίηση των εργαλείων λογισμικού που υλοποιήθηκαν ή εξελίχθηκαν στα πλαίσια της ερευνητικής διαδικασίας του διδακτορικού. Για το εργαλείο ελέγχου και αξιολόγησης ποιότητας υπηρεσίας (SLA-3ALib) παρατίθενται οι προδιαγραφές και τα στάνταρντ που τυποποίησαν τις δομές δεδομένων, καθώς και η υλοποίηση του συστήματος, παράλληλα σε αυτό το κεφάλαιο παρουσιάζεται και η τεχνική υλοποίηση καθώς και οι διεπαφές του εργαλείου. Η υλοποίηση του υποσυστήματος συγκριτικής αξιολόγησης (QoE-Bench Suite) περιέχει την τυποποίηση των μετρικών για την αξιολόγηση της απόδοσης των συστημάτων Cloud καθώς και την τεχνική υλοποίηση και διεπαφή του. Για το εργαλείο δημιουργίας προφίλ αναγκών ήταν σημαντικό να αναλυθεί ενδελεχώς το θεωρητικό υπόβαθρο καθώς οι μέθοδοι κατηγοριοποίησης των εφαρμογών χρησιμοποιείται και από τα μοντέλα παρεμβολών. Επίσης, παρατίθεται και η αρχιτεκτονική καθώς και η μεθοδολογία δημιουργίας προφίλ αναγκών. Η τεχνική υλοποίηση δεν παρατίθεται διότι δεν αποτελεί απορία της παρούσας διατριβής. Το τελευταίο υποκεφάλαιο αναλύει την τεχνική υλοποίηση των μοντέλων

παρεμβολών και κατανομής πόρων που βασίζονται σε γενετικούς αλγορίθμους δημιουργίας τεχνικών νευρωνικών δικτιών.

Το **κεφάλαιο 5** παρουσιάζει την αξιολόγηση και πειραματική διαδικασία που εξετάζει αν τα εργαλεία λογισμικού που αναπτύχθηκαν στα πλαίσια της διατριβής πετυχαίνουν τον σκοπό ανάπτυξη τους. Για τα εργαλεία λογισμικού που αφορούν την αξιολόγηση ποιότητας υπηρεσίας και SLA εξετάστηκε το κατά πόσο αυτά τα εργαλεία μπορούν να κλιμακωθούν και να εξυπηρετήσουν πολλαπλούς χρήστες αλλά και να μπορούν να συλλέγουν ένα μεγάλο όγκο δεδομένων, ώστε να πραγματοποιηθεί η αξιολόγηση στις Cloud υπηρεσίες. Ωστόσο, για να πετύχει αυτήν την λειτουργικότητα το σύστημα χρειάζεται να εκτελέσει ένας αριθμός από διεργασίες, και για το λόγο αυτό εξετάστηκε επίσης το κατά πόσο οι διεργασίες των συστημάτων αυτών διεκπεραιώνονται επιτυχώς. Η αξιολόγηση για τα μοντέλα παρεμβολών και κατανομής πόρων επικεντρώνεται στην αξιολόγηση της ακρίβειας των μοντέλων που παράγονται από αυτά τα δύο εργαλεία. Για την παραγωγή του συνόλου δεδομένων για την εκπαίδευση αλλά και την αξιολόγηση των μοντέλων κατανομής πόρων χρησιμοποιήθηκε μια εφαρμογή (CRM/ERP) και αξιολογήθηκε το κατά πόσο το μοντέλο μπορεί να αντιστοιχίσει μετρικές για την ποιότητα υπηρεσίας σε συγκεκριμένο αριθμό διαφόρων πόρων, όπως CPU, RAM. Για τα μοντέλα παρεμβολών το σύνολο δεδομένων που εξετάστηκε αποτελούταν από εκτελέσεις πολλαπλών Benchmarks παράλληλα, σε τυποποιημένους εικονικούς πόρους. Η αξιολόγηση μελέτησε το κατά πόσο το μοντέλο μπορούσε να βγάλει μοτίβα μείωσης απόδοσης λόγω παρεμβολών που δημιουργούν μείωση απόδοσης των εικονικών πόρων.

Το **κεφάλαιο 6** περιέχει την συμπερασματολογία για την έρευνα της παρούσας διατριβής. Σχολιάζεται, ακόμη, το κατά πόσο τα εργαλεία αξιολόγησης και βελτίωσης υπηρεσιών

Cloud πέτυχαν το στόχο τους, το επίπεδο ωριμότητας που έφτασαν, καθώς και το πως μπορούν να βελτιωθούν και να εξελιχθούν σε μελλοντικές ερευνητικές δραστηριότητες.

Το **κεφάλαιο 7** περιέχει τις συντομογραφίες αγγλικών όρων και το **κεφάλαιο 8** περιέχει τις βιβλιογραφικές αναφορές της παρούσας διατριβής.

2 *Βιβλιογραφική και τεχνολογική επισκόπηση συστημάτων υπολογιστικού νέφους*

Η ανάλυση της τεχνολογικής αιχμής που πραγματοποιείται στο κεφάλαιο αυτό, παρουσιάζει τη βιβλιογραφική και ερευνητική δουλειά που έγινε στους διάφορους τομείς που αφορούν την διατριβή αυτή, καθώς και εργαλεία και συστήματα που έχουν αναπτυχθεί. Ο κύριος γνώμονας για την ανάλυση αυτή είναι να δημιουργηθεί λογισμικό, που όχι μόνο παρουσιάζει ερευνητικές καινοτομίες, αλλά που παράλληλα θα είναι λειτουργικό καθώς και θα έχει χαρακτηριστικά που θα το καθιστούν χρηστικό.

Ό,τι αφορά τη τεχνολογική αιχμή, η ανάλυση που συντάχθηκε βασίζεται σε τρεις κύριους πυλώνες. Ο πρώτος πυλώνας είναι η ανάλυση για την συμφωνία επιπέδου ποιότητας υπηρεσίας (SLA) σε υπολογιστικά νέφη. Τα SLA, παρόλο που λειτουργούν στο παρασκήνιο, είναι ένα κομμάτι που υπάρχει σε οποιαδήποτε υπηρεσία που περιλαμβάνει έναν πάροχο και ένα χρήστη της υπηρεσίας αυτής. Στα πλαίσια της διατριβής αυτής, το κύριο ερευνητικό ενδιαφέρον ήταν η ανάλυση, η ψηφιοποίηση, καθώς και η αξιολόγηση των SLA που αφορούν τα υπολογιστικά νέφη.

Η αξιολόγηση της ποιότητας της συμφωνίας δίνει μια εκτίμηση για το πόσο καλά και έμπιστα οι πάροχοι υπηρεσιών Cloud τηρούν τη συμφωνία και το συμβόλαιο που υπάρχει μεταξύ παρόχου και χρήστη. Ωστόσο, δεν αναλύουν την ποιότητα της υπηρεσίας και την απόδοσή του συστήματος, διότι υπάρχει περίπτωση τα εχέγγυα που προσδιορίζονται στο SLA να καλύπτονται αλλά αυτό δεν συνεπάγεται ότι η ποιότητα της υπηρεσίας είναι καλή παρά μόνο αξιόπιστη. Ο λόγος είναι ότι τα εχέγγυα που δίνει ο πάροχος μπορεί να είναι πολύ περιορισμένα(η διαθεσιμότητα της υπηρεσίας). Για το λόγο αυτό, χρειάζεται να γίνει αξιολόγηση της απόδοσης του συστήματος ανεξαρτήτως από το SLA που έχει συμφωνηθεί.

Ο δεύτερος πυλώνας της βιβλιογραφικής ανάλυσης αφορά την συγκριτική αξιολόγηση (Benchmarking) των υπηρεσιών υπολογιστικού νέφους. Στο κομμάτι αυτό, αναλύονται οι βιβλιογραφικές προσεγγίσεις καθώς και υλοποιημένες λύσεις λογισμικού που χρησιμοποιούνται κατά κόρον. Η συγκριτική αξιολόγηση υποδομών έχει εγκαθιδρυθεί πολλά χρόνια, ωστόσο δοθείσης της ιδιαιτερότητας των Cloud υποδομών (εικονικές μηχανές VM, τεχνολογία Hyper-Threading στο Cloud) οι συμβατικές προσεγγίσεις για την συγκριτική αξιολόγηση δεν μπορούν να αποδώσουν τα μέγιστα. Η μελέτη στο κεφάλαιο 2.2 αναλύει το πώς εργαλεία καθώς και ήδη εγκαθιδρυμένες τεχνικές μπορούν να χρησιμοποιηθούν σε συνδυασμό ώστε να δημιουργηθεί μια πιο έγκυρη αξιολόγηση της απόδοσης των Cloud υποδομών.

Η αξιολόγηση της αξιοπιστίας καθώς και της απόδοσης των υπηρεσιών νέφους είναι πολύ σημαντικά στοιχεία, παρόλα αυτά η αξιολόγηση ιδιαίτερα της απόδοσης πρέπει να συνυπάρχει με την κατανόηση των αναγκών του χρήστη. Πριν την μετακόμιση του λογισμικού από τις εσωτερικές εγκαταστάσεις στο Cloud, είναι σημαντικό να αναλυθούν και να κατανοηθούν οι ανάγκες του λογισμικού ώστε να βρεθεί η βέλτιστη λύση για τον

χρήστη. Μια υπηρεσία Cloud δύναται να είναι πιο αποδοτική από κάποια άλλη και αυτό συνεπάγεται στην συντριπτική πλειοψηφία των περιπτώσεων ότι είναι πιο κοστοβόρα. Κατανοώντας τις ανάγκες του λογισμικού τους οι χρήστες μπορούν να επιλέξουν λύσεις που θα είναι αποδοτικές αλλά και με μικρότερο κόστος από σχετικά καλύτερες υπηρεσίες που δεν είναι ωστόσο αναγκαίες για την διατήρηση της ποιότητας υπηρεσίας που θέλουν οι χρήστες. Παράλληλα, η κατανόηση των αναγκών σε πόρους, δύναται να βοηθήσει και τους παρόχους των υπηρεσιών νέφους, ώστε να κάνουν καλύτερη τοποθέτηση του λογισμικού στο Cloud (π.χ σε ένα φυσικό μηχάνημα να τοποθετηθούν 2 εφαρμογές που έχουν διαφορετικές ανάγκες σε πόρους, μια εφαρμογή που χρειάζεται μεγάλη υπολογιστική ισχύ και μια εφαρμογή που χρειάζεται μεγάλη απόδοση στο δίκτυο).

Στο κεφάλαιο 2.3 αναλύεται η βιβλιογραφία για την δημιουργία προφίλ των αναγκών λογισμικού καθώς και της κατηγοριοποίησης τους, ώστε να επιτευχθούν οι στόχοι που προαναφέρθηκαν.

Τέλος, είναι σημαντικό να αναφερθεί ότι μια κύρια περίπτωση χρήσης που μελετήθηκε και χρησιμοποιήθηκε στην έρευνα αυτή είναι οι υπολογιστές υψηλών επιδόσεων (HPC). Για το λόγο αυτό παρατίθεται και μια βιβλιογραφική επισκόπηση της αξιολόγησης καθώς και εργαλείων που αφορούν την ανάλυση των HPC σε υποδομές υπολογιστικού νέφους.

Στην κάτωθι εικόνα (Εικόνα 2) απεικονίζονται τα ερευνητικά πεδία ενασχόλησης της παρούσας διατριβής.



Εικόνα 2: Βιβλιογραφική και τεχνολογική προσέγγιση διατριβής.

2.1 Συμφωνία επιπέδου ποιότητας υπηρεσίας(SLA) στα υπολογιστικά νέφη(Cloud)

Σύμφωνα με το [6], μερικές από τις βασικές αρχές της παρακολούθησης και αξιολόγησης υπολογιστικού νέφους, περιλαμβάνουν πτυχές όπως η χρήση μιας μεμονωμένης πλατφόρμας για όλους τους πόρους που χρησιμοποιούνται μια δεδομένη στιγμή, την παρακολούθηση του συστήματος από την άποψη της εμπειρίας του χρήστη κατά την χρησιμοποίηση του συστήματος καθώς και τη δυνατότητα ενσωμάτωσης μετρικών για την δημιουργία μιας πλήρους ανάλυσης της απόδοσης του υπολογιστικού νέφους.

Μια σύγκριση και ανάλυση μεταξύ διαφορετικών ορισμών SLA παρουσιάζεται στο [7], μαζί με κοινά προβλήματα που μπορούν να δημιουργηθούν στη διαδικασία μέτρησης που περιγράφεται στο [8]. Σύμφωνα με αυτήν την ανάλυση, η οποία οδήγησε την αρχική ανάπτυξη και χρήση των εργαλείων στα πλαίσια αυτής της έρευνας, ένας μηχανισμός δειγματοληψίας θα πρέπει να προσαρμοστεί πλήρως στο εγγυημένο KPI (Key Performance Indicator), ελέγχοντας πριν από τη δειγματοληψία τις προϋποθέσεις ανάπτυξης κάθε SLA, ελέγχοντας τη συνολική κατάσταση της υπηρεσίας και όχι μεμονωμένων στιγμών λειτουργίας της υπηρεσίας και υπολογισμός των επιπέδων KPI με βάση τους συγκεκριμένους τύπους που αναφέρονται από τους παρόχους. Ως ενδεικτικά παραδείγματα, οι τυπικές προϋποθέσεις περιλαμβάνουν εγκατάσταση σε πολλές ζώνες διαθεσιμότητας και οι αποτυχίες υπηρεσίας μπορούν να μετρηθούν μόνο όταν όλοι οι εν χρήση πόροι δεν είναι διαθέσιμοι. Ως τυπικό παράδειγμα φόρμουλας υπολογισμού διαθεσιμότητας συστήματος, οι αποτυχίες μπορούν να μετρηθούν μόνο στο συνολικό μηνιαίο ποσοστό μόνο εάν η διάρκειά τους είναι μεγαλύτερη από ένα δεδομένο διάστημα ή ένα δεδομένο ποσοστό σφάλματος στο καθορισμένο διάστημα.

Γενικά, υπάρχει ποικιλία εργαλείων παρακολούθησης [9][10], ωστόσο το κύριο ζήτημα είναι ότι, σε όλες αυτές τις περιπτώσεις υπάρχει έλλειψη προσαρμογής στους συγκεκριμένους και διαφορετικούς ορισμούς SLA (όσον αφορά τις προϋποθέσεις, τους τύπους υπολογισμού κ.λπ.) στα δημόσια συστήματα υπολογιστικού νέφους.

Στο [11], παρουσιάζεται το εργαλείο SeICSP, το οποίο υπογραμμίζει τις διαφορές στα Cloud SLAs, ακόμη και για τον ίδιο τύπο υπηρεσιών και βοηθά τους χρήστες στην επιλογή παρόχων που ταιριάζουν καλύτερα στις ανάγκες τους. Η βαθμολογία εξάγεται με βάση τα σχόλια των χρηστών από προηγούμενες εμπειρίες και από την εξέταση των αντίστοιχων SLA ως προς τη διαφάνεια. Στο [12], παρουσιάζεται το SMICloud,

προκειμένου να ταξινομηθούν οι υπηρεσίες Cloud από διάφορες οπτικές γωνίες. Αυτό δεν εμπλέκεται άμεσα με την παρακολούθηση SLA, ωστόσο αυτή η πτυχή, κυρίως με τη μορφή πιθανότητας του παρόχου να διατηρήσει το SLA, λαμβάνεται υπόψη στη διαδικασία κατάταξης. Τα στατιστικά στοιχεία παραβίασης παρόχου SLA που συγκεντρώθηκαν χρησιμοποιώντας τα εργαλεία που αναπτύχθηκαν, θα μπορούσαν να συνδυαστούν με αυτές τις προσεγγίσεις για τα αντίστοιχα μέρη που έχουν σχέση με την εμπειρία χρήστη ή την πιθανότητα παραβίασης του παρόχου SLA. Ωστόσο, οι μετρήσεις που ορίζονται από το SMICloud περιλαμβάνουν πολλές αυθαίρετες πτυχές καθώς και πληροφορίες που εισάγονται χειροκίνητα.

Για πληροφορίες παρακολούθησης σε πραγματικό χρόνο σχετικά με τη διαθεσιμότητα των υπηρεσιών, μια πολύ ενδιαφέρουσα προσέγγιση είναι το Cloudsleuth[13]. Το Cloudsleuth διαθέτει ένα εκτεταμένο δίκτυο εφαρμογών σε πολλούς παρόχους και τοποθεσίες και παρακολουθεί συνεχώς την απόδοσή τους σε σχέση με τους χρόνους απόκρισης και τις μετρήσεις διαθεσιμότητας. Ο τρόπος μέτρησης του Cloudsleuth δεν είναι προσαρμοσμένος στον ορισμό SLA κάθε παρόχου, επομένως δεν μπορεί να χρησιμοποιηθεί για αξιολόγηση αποζημίωσης. Επιπλέον, ελέγχει την απόκριση ενός διακομιστή διαδικτύου (κατάσταση 200 τύπου επιστροφής σε αίτημα GET). Επομένως, δεν μπορεί να διακρίνει μεταξύ αστοχίας λόγω μη διαθέσιμου VM (Virtual Machine) (περίπτωση ευθύνης παρόχου) ή λόγω διακοπής λειτουργίας διακομιστή εφαρμογών (ευθύνη πελάτη σε περίπτωση ανάπτυξης IaaS, ευθύνη παρόχου σε περίπτωση PaaS / SaaS) ή καθαρού σφάλματος εφαρμογής (ευθύνη πελάτη). Από την άλλη πλευρά, ακολουθεί έναν κανονικό ορισμό διαθεσιμότητας που καθιστά εφικτή τη σύγκριση υπηρεσιών από διαφορετικούς παρόχους, μια διαδικασία που δεν μπορεί να εκτελεστεί ενώ ακολουθεί τους συγκεκριμένους ορισμούς SLA, καθώς έχουν διαφορές στον τρόπο

που καθορίζουν τη διαθεσιμότητα. Επιπλέον, το CloudSleuth φαίνεται να μην είναι διαθέσιμο στο κοινό τους τελευταίους μήνες. Το CloudHarmony[14] είναι μια άλλη προσπάθεια που προσφέρει μετρήσεις διαθεσιμότητας και αναφέρει διαστήματα μη διαθεσιμότητας, ωστόσο δεν είναι σαφές πώς μετρούνται τα συνολικά επίπεδα (σύμφωνα με τον ορισμό κάθε παρόχου ή όχι) και εάν η ανάπτυξη της υπηρεσίας μέτρησης συμμορφώνεται με την SLA του συγκεκριμένου παρόχου. Επιπλέον, δεν μπορεί να χρησιμοποιείται για τις απαιτήσεις ενός μεμονωμένου χρήστη και παρόχου, δεδομένου ότι αναφέρεται και παρακολουθεί την συγκεκριμένη υπηρεσία για δοκιμές της ιστοσελίδας και να όχι για κάποιο λογαριασμό οποιουδήποτε χρήστη. Επομένως, παρακολουθεί μόνο ένα πολύ συγκεκριμένο υποσύνολο του συνολικού κέντρου δεδομένων .

Site24x7[15] προσφέρει δυνατότητες διαχείρισης SLA ως γνώμονα μεταξύ πάροχων υπηρεσιών και κατόχων εφαρμογών, ωστόσο φαίνεται ότι βασίζεται και πάλι στη διαθεσιμότητα εφαρμογών υψηλού επιπέδου και δεν είναι προσαρμοσμένη στους συγκεκριμένους ορισμούς μιας δεδομένης υπηρεσίας Cloud. Επιπλέον, δεν είναι πολύ σαφές τι συμβαίνει για πιο εξελιγμένους τύπους, όπως η κοινή περίπτωση σε πολλές δημόσιες υπηρεσίες Cloud, όπως το GAE ή το Azure, όπου το ποσοστό σφάλματος μετράται π.χ. σε ωριαία διαστήματα, στα οποία το σφάλμα πρέπει να υπερβαίνει ένα δεδομένο όριο και με διαφορετικά χρονικά όρια απόκρισης ανά τύπο λειτουργίας. Bitnami [16] προσφέρει εργαλεία για τη διαχείριση και την παρακολούθηση του Cloud, αλλά αναφέρεται κυρίως σε διεπαφές για τη χρήση υφιστάμενων υπηρεσιών παρόχου, όπως το AWS Cloudwatch .

Το Uptime Cloud Monitor[17] είναι μια εμπορική υπηρεσία για την παρακολούθηση υπηρεσιών Cloud, συμπεριλαμβανομένων προσαρμόσιμων πινάκων ελέγχου και μια ενδιαφέρουσα σειρά μετρήσεων που χρησιμοποιούνται για τη δημιουργία ειδοποιήσεων

και ειδοποιήσεων για τον χρήστη. Μια παρόμοια υπηρεσία είναι το CloudMonix [18]. Η υποστήριξη σε αυτό το στάδιο φαίνεται να περιορίζεται στο Microsoft Azure. Το Netcraft[19] είναι μια εταιρεία που παρακολουθεί την απόδοση των περιβαλλόντων φιλοξενίας από συγκεκριμένες απόψεις (π.χ. δείγματα δικτύωσης κάθε 15 λεπτά ανά κεντρικό υπολογιστή) για μια σταθερή ετήσια τιμή. Και πάλι, αυτή η προσέγγιση στη δειγματοληψία μπορεί να χρησιμοποιηθεί για σύγκριση διαφορετικών προσφορών, ωστόσο δεν προσαρμόζεται στους συγκεκριμένους ορισμούς των SLA σε δημόσια περιβάλλοντα Cloud.

Το Cloud Monitor της Akamai[20] είναι μια υπηρεσία push API σε πραγματικό χρόνο που παρέχει κρίσιμα δεδομένα συναλλαγών από την έξυπνη πλατφόρμα της Akamai. Το Cloud Monitor παρέχει πληροφορίες σε πραγματικό χρόνο για ένα ευρύ φάσμα δεδομένων που συλλέγονται, συμπεριλαμβανομένων βασικών λεπτομερειών συναλλαγών, μετρήσεων απόδοσης δικτύου και πελατών, ειδοποιήσεις ασφαλείας, στατιστικά δικτύου, δεδομένα τοποθεσίας και πληροφορίες cookie. Αν και είναι ενδιαφέρον ως πηγή πληροφοριών, τα ληφθέντα δεδομένα θα πρέπει να υποβληθούν σε επεξεργασία και να υπολογιστούν με βάση τον συγκεκριμένο ορισμό SLA κάθε παρόχου.

Zabbix[21] είναι ένα εργαλείο ανοιχτού κώδικα που μπορεί να χρησιμοποιηθεί για την παρακολούθηση των υποδομών πληροφορικής. Έρχεται με μια σειρά από ενδιαφέροντα χαρακτηριστικά, όπως σύνδεση με συγκεκριμένες υπηρεσίες και κόμβους, δημιουργία αλυσίδων εξαρτήσεων μεταξύ πόρων κ.λπ. Η πιο ενδιαφέρουσα πτυχή είναι το γεγονός ότι ο ορισμός SLA μπορεί να είναι πιο λεπτομερής από τις άλλες περιπτώσεις διαθέσιμων στο κοινό εργαλείων (συμπεριλαμβανομένων περισσότερων περίπλοκες συνθήκες σε ένα αποτυχημένο δείγμα, αν και χωρίς πλήρη ευελιξία για τον χρήστη να καθορίσει έναν

συγκεκριμένο τύπο), ωστόσο φαίνεται ότι είναι πιο κατάλληλο για τον ίδιο τον πάροχο να αυτό-παρακολουθήσει των πόρων.

Ένα από τα βασικά χαρακτηριστικά του λογισμικού που αναπτύχθηκε στα πλαίσια της διατριβής που του επιτρέπει να διακριθεί από τον ανταγωνισμό, είναι το γεγονός ότι προσαρμόζεται πλήρως σε διαφορετικούς ορισμούς παρόχων, συμπεριλαμβανομένων των απαιτούμενων ελέγχων των προϋποθέσεων ανάπτυξης ή των συνολικών εκτιμήσεων κατά τη δειγματοληψία μιας τρέχουσας υπηρεσίας Cloud. Για παράδειγμα, εάν μια υπηρεσία αποτελείται από πολλά VM ή στοιχεία, ένα τυπικό εργαλείο παρακολούθησης θα περιλαμβάνει στατιστικά στοιχεία για κάθε μεμονωμένο στοιχείο. Ωστόσο, από άποψη SLA, μόνο εάν όλοι οι χρησιμοποιημένοι πόροι Cloud δεν είναι ταυτόχρονα διαθέσιμοι, τότε το συγκεκριμένο δείγμα μπορεί να θεωρηθεί μη διαθέσιμο (σε επίπεδο περιοχής Cloud). Έτσι, τα αρχεία καταγραφής που διατηρούνται από το λογισμικό μπορούν στην πραγματικότητα να χρησιμοποιηθούν για αποζημίωση, ενώ τα αντίστοιχα αρχεία καταγραφής από δημοφιλή εργαλεία ενδέχεται να μην μπορούν να χρησιμοποιηθούν για αυτό το σκοπό.

2.2 Συγκριτική αξιολόγηση(Benchmarking) υπηρεσιών υπολογιστικού νέφους

Η ανάγκη για μια δίκαιη συγκριτική αξιολόγηση των υπηρεσιών Cloud αυξάνεται καθώς όλο και περισσότερες εταιρείες προσελκύνονται από το πρότυπο Cloud και θέλουν να μεταφέρουν την εφαρμογή τους σε αυτό. Η διαδικασία συγκριτικής αξιολόγησης μπορεί να παρέχει συγκριτικά αποτελέσματα προκειμένου να ληφθούν σωστές αποφάσεις για την μεταφορά μιας δεδομένης εφαρμογής σε Cloud περιβάλλον. Ωστόσο, η συγκριτική αξιολόγηση του Cloud συνεπάγεται αναθεώρηση των κλασικών εργαλείων και μετρήσεων

συγκριτικής αξιολόγησης. Για παράδειγμα, η κλασική CPU, η μνήμη, οι μετρήσεις του αριθμού συναλλαγών είναι ξεπερασμένες σε ένα περιβάλλον Cloud που μπορεί να κλιμακωθεί για να ικανοποιήσει (σχεδόν) οποιαδήποτε ζήτηση. Αυτό αναγνωρίζεται ευρέως στη βιβλιογραφία [22][23][24][25][26] και έχουν γίνει πολλές προσπάθειες για τον καθορισμό νέων εργαλείων και μετρήσεων για μια πιο ουσιαστική συγκριτική αξιολόγηση του Cloud.

Όσον αφορά τα εργαλεία συγκριτικής αξιολόγησης, το CloudSuite [27] και PerfKit Benchmark (PKB) [28] πρέπει να αναφερθούν. Το πρώτο έχει ορίσει ένα σύνολο οκτώ benchmark που βασίζονται σε πολύ δημοφιλή εφαρμογές του Cloud χρησιμοποιώντας ρεαλιστικούς φόρτους εργασίας (π.χ. Data Caching, Data Serving, Media Streaming) χρησιμοποιώντας πραγματικές ρυθμίσεις. Το PerfKit Benchmarker ενθυλακώνει τα δημοφιλή τεστ συγκριτικής αξιολόγησης (περίπου 30) και αυτοματοποιεί την εγκατάσταση και την εκτέλεση αυτών των τεστ σε επτά διαφορετικούς παρόχους εμπορικών Cloud (συμπεριλαμβανομένων των Google Cloud Platform, AWS και Azure). Στα πλαίσια της ερευνάς το εργαλείο που χρησιμοποιήθηκε λαμβάνει υπόψη αυτές τις προσπάθειες και ενσωματώνει ήδη ορισμένα από τα εργαλεία CloudSuite και PerfKit Benchmarker, ώστε να μπορέσει να εκμεταλλευτεί όλα τα χαρακτηριστικά των εφαρμογών αυτών, για μια πιο ακριβή μελέτη απόδοσης συστημάτων υπολογιστικού νέφους.

Από την πλευρά των μετρήσεων, BUNGEE[24][29] προτείνει μια νέα προσέγγιση για τη μέτρηση της ακρίβειας ελαστικότητας(elasticity) του Cloud, ενώ το NIST δημοσίευσε την "Περιγραφή μετρήσεων υπηρεσίας υπολογιστικού νέφους" [30] που προτείνει μια τυποποίηση του ορισμού των μετρήσεων Cloud.

Μια άλλη σχετική και επίσημη προσπάθεια έχει γίνει από την Standard Performance Evaluation Corporation (SPEC). Το 2016, η SPEC κυκλοφόρησε το πρώτο της τυποποιημένο εργαλείο μελέτης απόδοσης για συστήματα Cloud: το "SPEC Cloud_IaaS 2016"[22]. Αυτό το εργαλείο μελέτης απόδοσης ορίζει και μετρά τρεις μετρήσεις συστημάτων cloud: Επεκτασιμότητα, Ελαστικότητα και Μέσος χρόνος στιγμιαίας παροχής υπηρεσίας. Βασίζεται σε δύο φόρτους εργασίας που δημιουργεί μια συστάδα Cassandra (ή Hadoop) στο υπολογιστικό νέφος και μετρά την ποιότητα υπηρεσίας (χρησιμοποιώντας YCSB). Επίσης, σε αυτήν την περίπτωση, το αναπτυχθέν εργαλείο προσφέρει μια παρόμοια προσέγγιση, αλλά ενσωματώνει περισσότερα εργαλεία και προσφέρει υψηλότερο επίπεδο γενικευμένης χρήσης έναντι των παρόχων Cloud (οδηγίες ρύθμισης και εκτέλεσης Cloud_IaaS 2016[31] περιλαμβάνουν πολλά βήματα πολύ συγκεκριμένα για κάθε διαφορετικό πάροχο).

Εκτός από τα εργαλεία, υπάρχουν λίγες εταιρείες που προσφέρουν υπηρεσίες για την παροχή και σύγκριση αποτελεσμάτων συγκριτικής αξιολόγησης. Cloud Spectator [32] παρέχει μια υπηρεσία συγκριτικής αξιολόγησης για τους χρήστες και τους παρόχους Cloud, ωστόσο τα ελεύθερα προσβάσιμα δεδομένα περιορίζονται σε λίγους παρόχους Cloud και σε λίγες βασικές μετρήσεις (π.χ. απόδοση CPU, απόδοση ανάγνωσης / εγγραφής δίσκου και μνήμης, χρόνος απόκρισης δικτύου). Ομοίως, επίσης το ServerBear παρέχει μια ενότητα για τη συγκριτική αξιολόγηση των αποτελεσμάτων του Cloud [33]. Πριν εξαγοραστεί από την Gartner το 2015, επίσης το CloudHarmony χρησιμοποιούνταν για τη δημοσίευση ανεξάρτητων αποτελεσμάτων συγκριτικής αξιολόγησης και διαθεσιμότητας υπηρεσιών από διάφορους παρόχους Cloud.

2.3 Προφίλ αναγκών και κατηγοριοποίηση λογισμικού

Οι εφαρμογές που βρίσκονται στο ίδιο φυσικό μηχάνημα έχουν πρόσβαση και καταναλώνουν τους ίδιους φυσικούς πόρους. Η κατάσταση αυτή δύναται να δημιουργήσει τροποποιήσεις στην απόδοση των εφαρμογών, ιδιαίτερα όταν εφαρμογές με παρόμοιες ανάγκες βρίσκονται στο ίδιο μηχάνημα. Το κεφάλαιο αυτό παραθέτει μια βιβλιογραφική επισκόπηση του πώς μπορούν να ανιχνευθούν τα μοτίβα απόδοσης λόγω παραλληλοποίησης καθώς και πώς μπορεί να προσεγγιστεί το πρόβλημα αυτό για να διατηρηθεί η απόδοση ενός υπολογιστικού νέφους δημιουργώντας βελτιωμένες κατανομές των εφαρμογών.

2.3.1 Προσαρμογή και πρόβλεψη μοντέλων παραλληλοποίησης εικονικών μηχανών (VM)

Οι Jiang et al. [34], έκαναν μελέτες για το πρόβλημα της κοινής τοποθέτησης VM και της επιλογής διαδρομής στο πλαίσιο ενός κέντρου δεδομένων με δυνατότητα πολλαπλών διαδρομών, όπου οι τεχνικές διαχείρισης κίνησης μπορούν να αξιοποιηθούν για να δρομολογήσουν αυθαίρετα μεμονωμένη κίνηση VM. Επιπροσθέτως, οι συγγραφείς αντιμετώπισαν τη μετακίνηση ενός ελάχιστου αριθμού VM για την εκ νέου βελτιστοποίηση της απόδοσης της υποδομής σε κάθε νέα αποδοχή ή τερματισμό VM. Οι Verma et al. [35] παρουσίασαν ένα δυναμικό πλαίσιο πρόβλεψης και κατανομής ζήτησης πόρων σε υπολογιστικά νέφη εξυπηρέτησης πολλών μισθωτών, χρησιμοποιώντας μια δεντρική δομή περικοπής μειωμένου σφάλματος (REPTree), έναν δημοφιλή αλγόριθμο ταξινόμησης, που βασίζεται σε έναν σύστημα γρήγορης απόφασης, για την ταξινόμηση των ενοικιαστών υπηρεσιών (εφαρμογές «Λογισμικό ως υπηρεσία») βάσει ιστορικών δεδομένων, ανάλογα με το εάν οι απαιτήσεις πόρων τους θα αυξηθούν ή όχι. Αυτό

καταλήγει σε μια νέα προσέγγιση δυναμικής κατανομής πόρων χρησιμοποιώντας τον καλύτερο προσαρμοζόμενο αλγόριθμο μείωσης.

Το CloudSim παρέχει μια δωρεάν και ανοιχτή εργαλειοθήκη [36] που υποστηρίζει μοντελοποίηση συστήματος και συμπεριφοράς στοιχείων του συστήματος Cloud, όπως κέντρα δεδομένων, εικονικές μηχανές(VM) και πολιτικές παροχής πόρων. Προσομοιώνοντας τους παρόχους IaaS επεκτείνοντας την οντότητα του κέντρου δεδομένων, το CloudSim μπορεί να εξετάσει διάφορες τεχνικές παροχής για την κατανομή των VMs σε σενάρια διασυνδεδεμένων υπολογιστικών Cloud. Το CloudSim εφαρμόζει δύο διαφορετικές πολιτικές για VM που φιλοξενούνται στον ίδιο φυσικό κεντρικό υπολογιστή: πολιτική παροχής χρόνου και κοινόχρηστου χώρου, εξετάζοντας όλα τα πιθανά σενάρια με βάση τις απαιτήσεις πόρων VM.

2.3.2 Ελαχιστοποίηση πτώσης απόδοσης συστήματος υπολογιστικού νέφους

Μια πιο συγκεκριμένη μελέτη σχετικά με τα αποτελέσματα ταυτόχρονης λειτουργίας εφαρμογών παρέχεται από την μελέτη στο [22], η οποία μελετά τις βέλτιστες στρατηγικές κατανομής Cloud για αποφάσεις προγραμματισμού σε πραγματικό χρόνο και τη συντοποθέτηση VM όταν αυτές αναπτύσσονται ταυτόχρονα στον ίδιο φυσικό κόμβο. Εξετάζονται διαφορετικά επίπεδα κοινής χρήσης μνήμης, με βάση τη σύνδεση των VM που βασίζονται σε μια αρχιτεκτονική πολλαπλών πυρήνων, λαμβάνοντας υπόψη τις βαθμολογίες συγκεκριμένων σημείων αναφοράς που εκτελούνται εντός των VM. Αυτή η εργασία επιχειρεί να προβλέψει τα αποτελέσματα διαφορετικών τοποθετήσεων εφαρμογών (στον ίδιο πυρήνα, σε παρακείμενους ή μη γειτονικούς πυρήνες σε έναν φυσικό κόμβο), για την προληπτική παροχή των πόρων που απαιτούνται από αυτές τις

εφαρμογές και την επιλογή των καταλληλότερων διαμορφώσεων ελαχιστοποιώντας τα γενικά έξοδα. Τα τεχνητά νευρωνικά δίκτυα (ANN) χρησιμοποιούνται για το σκοπό αυτό, που σχεδιάζονται και βελτιστοποιούνται αυτόματα μέσω μιας καινοτόμου εξελικτικής προσέγγισης, συντονίζοντας τις παραμέτρους ενός ANN χρησιμοποιώντας έναν γενετικό αλγόριθμο.

2.3.3 Αποτελεσματικά μοτίβα κατανομής με βάση μοντέλων βέλτιστης παραλληλοποίησης

Έχουν ήδη γίνει πολλές έρευνες, σχετικά με την αποτελεσματική και ελαστική κατανομή πόρων σε συστήματα Cloud, όπως διαπιστώνεται από τη σύγκριση μεταξύ 18 αλγορίθμων που πραγματοποίησαν οι Mills et al. [37]. Ενδιαφέρουσες αρχιτεκτονικές έννοιες επισημάνθηκαν από τους Xu et al. [38], όπου ένας αλγόριθμος τοποθέτησης VM βασίζεται στη θεωρία σταθερής αντιστοίχισης και στο έργο των Maurer et al. [39], όπου προτείνεται μια προσέγγιση βασισμένη στη διαχείριση της γνώσης για δυναμική βελτιστοποίηση των υποδομών Cloud.

Μια διατύπωση του προβλήματος κατανομής πόρων στο οποίο μετέπειτα εργασίες μπορούν να επαναχρησιμοποιήσουν πόρους που είχαν εκδοθεί από προηγούμενες εργασίες έγινε στο [40], στο οποίο χρησιμοποιήθηκε ένας αλγόριθμος προσέγγισης που μπορεί να αποδώσει σχεδόν τις βέλτιστες λύσεις σε πολυωνυμικό χρόνο. Οι Chang et al. στο [41] πρότειναν ένα πρόβλημα κατανομής για εφαρμογές τηλεπικοινωνιών, όπου οι αυθαίρετες εκφράσεις λανθάνοντος χρόνου χρησιμοποιούνται για τη μοντελοποίηση των απαιτήσεων λανθάνοντος χρόνου από άκρο σε άκρο των υπηρεσιών που θα τοποθετηθούν. Μελετήθηκαν επίσης αποτελεσματικά μοντέλα για παροχή VM, κατανομή

και τιμολόγηση στα υπολογιστικά νέφη [42][43], με στόχο τη μεγιστοποίηση των εσόδων του παρόχου με βελτιστοποίηση των λειτουργιών.

Μια από τις μεγαλύτερες προκλήσεις είναι ο σωστός σχεδιασμός μιας έξυπνης λογικής διαχείρισης πόρων που λαμβάνει υπόψη τις πολύπλευρες πτυχές της διαδικασίας τοποθέτησης VM, ενώ ικανοποιεί τις διάφορες απαιτήσεις ποιότητας υπηρεσίας (QoS). Η έρευνα στο [44] εισήγαγε το ROAR (Βελτιστοποίηση Πόρων, Σύστημα Κατανομής και Προτάσεων), ένα πλαίσιο μοντελοποίησης για απλοποίηση, βελτιστοποίηση και αυτοματοποίηση αποφάσεων κατανομής πόρων Cloud για την επίτευξη στόχων QoS για εφαρμογές ιστού, συμπεριλαμβανομένης σύνθετης εφαρμογής πολλαπλών επιπέδων που διανέμεται σε διαφορετικές ομάδες διακομιστών. Το ROAR αντλεί τις κατάλληλες διαμορφώσεις πόρων Cloud για ενορχήστρωση δοκιμών σε κάθε επιλογή κατανομής πόρων, προτού χρησιμοποιήσει τα αποτελέσματα για να προτείνει μια κατανομή πόρων βελτιστοποιημένου κόστους για την επίτευξη των στόχων QoS.

Η παράμετρος ελαστικότητας εξετάζεται διεξοδικά στο [45], παρέχοντας μια πιθανολογική προσέγγιση ελέγχου που προσπαθεί να βελτιστοποιήσει την κατανομή των υπηρεσιών μέσα σε έναν πάροχο Cloud και σε υποδομή ομοσπονδιακών παρόχων, λαμβάνοντας παράλληλα υπόψη τους στόχους του επιχειρηματικού επιπέδου (π.χ. οικολογική αποδοτικότητα και κόστος). Η συγκεκριμένη λύση χρησιμοποιεί έναν ευρετικό επιλυτή χρόνου εκτέλεσης που κλιμακώνει τους καταναμημένους πόρους σύμφωνα με τη ζήτηση και τα αναμενόμενα έσοδα, καθώς και το κόστος ομοσπονδίας, παρουσιάζοντας αποτελεσματικούς χρόνους υπολογισμού. Οι ευρετικοί αλγόριθμοι χρησιμοποιούνται επίσης από τους Xiao et al. [46], ενώ παράλληλα εισήγαγαν την έννοια της «ασυμμετρίας» ως τρόπου μέτρησης της ανομοιογένειας της χρήσης των διακομιστών Cloud. Λαμβάνοντας υπόψη τους επιχειρηματικούς στόχους, όπως η κατανάλωση

ηλεκτρικής ενέργειας, οι συγγραφείς προτείνουν μια οικολογική λύση για αποτελεσματική κατανομή πόρων σε ένα περιβάλλον υπολογιστικού νέφους. Η όλη διαδικασία βελτιστοποίησης πρέπει να γίνει σε σχέση με ένα δεδομένο επίπεδο QoS για τους πελάτες.

Τέλος, οι Govindaraju και Duran [47] προτείνουν ένα πλαίσιο εξισορρόπησης φορτίου και κατανομής πόρων με γνώμονα το QoS για παρόχους Cloud IaaS, όπου το SLA σχετίζεται με τις παραμέτρους του κύκλου ζωής του VM. Το πρόβλημα κατανομής πόρων στα VM διαμορφώνεται ως βελτιστοποίηση πολλαπλών στόχων, χρησιμοποιώντας έναν γενετικό Αλγόριθμο για την εξεύρεση βέλτιστης λύσης για τη μετάλλαξη, τη διασταύρωση και την επιλογή προσομοιωμένων VM σε περιβάλλον OpenStack, με στόχο επίσης την ελαχιστοποίηση του ενεργειακού κόστους.

Τα πιο δημοφιλή εργαλεία διαχείρισης Cloud, όπως το OpenNebula, εστιάζουν στην αποτελεσματική αποσύνδεση της διαχείρισης της υπηρεσίας από την υποδομή και δυνατότητα δυναμικής κατανομής εικονικών μηχανών σε μια ομάδα φυσικών πόρων, σύμφωνα με ένα σύνολο προκαθορισμένων πολιτικών[48]. Το OpenNebula χρησιμοποιεί μια λειτουργική μονάδα που ονομάζεται Capacity Manager που προσαρμόζει την τοποθέτηση των VM έχοντας σαν βάση ένα σύνολο προκαθορισμένων πολιτικών. Ο προεπιλεγμένος προγραμματισμός χωρητικότητας εφαρμόζει μια απλή πολιτική αντιστοίχισης και υποστηρίζει περιορισμούς ενοποίησης βάση χρήστη [49]. Ως μονάδα στήριξης(backend) προγραμματισμού, το OpenNebula μπορεί επίσης να χρησιμοποιήσει το Haizea , μια αρχιτεκτονική διαχείρισης μίσθωσης ανοιχτού κώδικα που επιτρέπει στους παρόχους να μισθώνουν τους πόρους τους, αντί να παρέχουν VM που πρέπει να ξεκινούν αμέσως και μπορούν επίσης να χρησιμοποιούν πόρους δημόσιων συστημάτων Cloud (π.χ. EC2), εάν χρειάζεται. Πρέπει να σημειωθεί ότι η διαχείριση εικονικής υποδομής

OpenNebula αυτοματοποιεί τη ρύθμιση VM, υποστηρίζοντας διάφορα υποκείμενα επίπεδα εικονικοποίησης (virtualization), όπως το VMWare [50].

Το VMware χρησιμοποιεί το Cloud Capacity Manager (CCM), ένα επεκτάσιμο, ιεραρχικό, ελαστικό σύστημα κατανομής πόρων που είναι χτισμένο πάνω από τον Διαχειριστή Κατανεμημένων Πόρων (DRS) [51]. Το CCM αποτελείται από τρία ιεραρχικά επίπεδα: (α) συστάδες (μικρές ομάδες ξενιστών όπως ορίζονται από το DRS), (β) υπερσυστάδες (ομάδες συστάδων) και (γ) υπολογιστικό νέφος (μια ομάδα υπερσυστάδων). Σε επίπεδο συμπλέγματος, το VMWare DRS λαμβάνει ανεξάρτητες και τοπικές αποφάσεις για την εξισορρόπηση των φορτίων με τη μετεγκατάσταση εικονικών μηχανών χρησιμοποιώντας εκτιμήσεις ζήτησης ανά εικονική μηχανή. Επιπλέον, το CCM αναμειγνύει δυναμικά τη χωρητικότητα μεταξύ συστάδων και υπερσυστάδων σε απάντηση σε συνολικές αλλαγές στη ζήτηση, χρησιμοποιώντας μια μέτρηση που ορίζεται ως αποτελεσματική απόδοση διαχείρισης ενέργειας (emat), η οποία είναι ο λόγος του αριθμού των επιτυχημένων κεντρικών κινήσεων προς το συνολικό ποσό του χρόνου που αφιερώνεται σε όλες τις κινήσεις (επιτυχημένες και αποτυχημένες).

Τέλος, το OpenStack ακολουθεί μια πιο ανοιχτή και διαμορφώσιμη προσέγγιση, χρησιμοποιώντας το εργαλείο διαχείρισης κρατήσεων και χρηματοδοτικής μίσθωσης Blazar Virtual Machine [52]. Το Blazar DB αποθηκεύει μια δομή δεδομένων συνδεδεμένων μισθώσεων, κρατήσεων και συμβάντων και μια επεκτάσιμη ανοιχτή πηγή Resource Reservation Service διαχειρίζεται πόρους από μια κοινή ομάδα σύμφωνα με τη διαμορφωμένη ροή εργασίας. Σε επίπεδο προγραμματισμού, ένα στοιχείο που ονομάζεται υπηρεσία nova-scheduler καθορίζει σε ποιον κεντρικό υπολογιστή θα πρέπει να ξεκινήσει ένα VM[53]. Αυτό χρησιμοποιεί από προεπιλογή έναν προγραμματιστή φίλτρων, ο οποίος υποστηρίζει το φιλτράρισμα και τη στάθμιση για τη λήψη ενημερωμένων αποφάσεων

σχετικά με το πού πρέπει να δημιουργηθεί μια νέα παρουσία. Ο προγραμματιστής φίλτρων OpenStack λαμβάνει αιτήματα για πόρους, εφαρμόζοντας πρώτα δυαδικά φίλτρα για να προσδιορίσει ποιοι κεντρικοί υπολογιστές είναι κατάλληλοι για εξέταση κατά την αποστολή ενός πόρου. Οι κεντρικοί υπολογιστές που γίνονται αποδεκτοί από το φίλτρο υποβάλλονται σε επεξεργασία με διαφορετικό αλγόριθμο για να αποφασίσουν ποιοι συγκεκριμένοι κεντρικοί υπολογιστές πρέπει να χρησιμοποιηθούν για να ικανοποιήσουν αυτό το αίτημα.

2.4 Αξιολόγηση ποιότητας υπηρεσιών υπολογιστικού νέφους

προσανατολισμένες σε υπολογιστές υψηλών επιδόσεων(HPC)

Προσεγγίσεις που επιτρέπουν ενοικίαση κατ' απαίτηση [54][55][56] πόρων HPC για διαφορετικούς πελάτες που υιοθετούν το μοντέλο λειτουργίας Cloud που παρέχει πόρους κατ' απαίτηση, συμπεριλαμβανομένων των δεσμευμένων παρουσιών υπολογιστών, επισημαίνονται ως λύσεις HPC Cloud, αλλά εννοιολογικά δε διαφέρουν από τους δεσμευμένους πόρους σε κανονικές συστάδες HPC. Τα συστήματα συμπλέγματος HPC που υποστηρίζουν περιβάλλοντα καθορισμένα από τον χρήστη [57] βασίζονται σε εικονικές μηχανές ή κοντέινερ. Ενώ το μοντέλο λειτουργίας παραμένει παρόμοιο με τα υπάρχοντα συστήματα συμπλεγμάτων HPC με τα κοινά συστήματα βασισμένων στην ουρά, η δυνατότητα εκτέλεσης εργασιών εντός ενός συγκεκριμένου χρήστη ή ακόμη και ενός περιβάλλοντος που ορίζεται από τον χρήστη όσο το δυνατόν περισσότερο σε περιβάλλοντα Cloud. Τέτοια συστήματα χαρακτηρίζονται συχνά ως HPC Cloud Systems.

Το Shifter [58][59] αντιμετωπίζει το πρόβλημα της παροχής ενός καθορισμένου από το χρήστη περιβάλλοντος διαδικασίας για εργασίες HPC που βασίζονται σε κοντέινερ με έναν βελτιστοποιημένο κοινόχρηστο μηχανισμό πρόσβασης δεδομένων. Το CernVM

διαχειρίζεται με τη σειρά του τη διαδικασία δημιουργίας προσαρμοσμένων εικόνων VM και υποστηρίζει την εκτέλεσή τους σε ένα ιδιωτικό Cloud. Κανένα από αυτά τα παραδείγματα δεν υποστηρίζει τους απαιτούμενους μηχανισμούς δυναμικής διαχείρισης πόρων και κλίμακας cross-Cloud. Ωστόσο, το έργο FaST αξιολογεί γενικά τη χρήση VM για εφαρμογές HPC [62] και παρέχει στρατηγικές προγραμματισμού [60][61][64] για ενθυλακωμένες διεργασίες που στοχεύουν στη βέλτιστη χρήση κόμβων και υπολογιστικών κόμβων. Ακόμη και αφηρημένες περιγραφές εφαρμογών και διεργασιών [63] έχουν χρησιμοποιηθεί για βέλτιστο προγραμματισμό, καθώς και σενάρια μετεγκατάστασης VM σε κέντρα δεδομένων που μελετήθηκαν [65]. Ωστόσο, τα σενάρια προγραμματισμού και μετεγκατάστασης μεταξύ Cloud απαιτούν πιο εξελιγμένες στρατηγικές βελτιστοποίησης σε συνδυασμό με την αντίστοιχη τιμολόγηση και χαρακτηριστικά απόδοσης διαφορετικών παρόχων και η πραγματική εκτιμώμενη συμπεριφορά εφαρμογής.

Τα συστήματα Cloud προσφέρουν ιδιαίτερα ισχυρές εγκαταστάσεις (όπως το Amazon CCI) και μερικό έλεγχο της τοποθέτησης και της εγγύτητας για συνδεσιμότητα δικτύου για τον τομέα του HPC. Εκτός από τις δημόσιες λύσεις Cloud όπως τα παραπάνω, η βάση των τεχνολογικών λύσεων είναι λύσεις ενδιάμεσου λογισμικού (π.χ. OpenStack) ή εμπορικές πλατφόρμες όπου η απαιτητική φύση των εφαρμογών HPC (πύρινες / μνήμη , μεγάλη χωρητικότητα μνήμης ανά πυρήνα υπολογιστών, επικοινωνία χαμηλού λανθάνοντος χρόνου) παρέχεται από ειδικά σχεδιασμένους τύπου Cloud. Άλλες προσεγγίσεις είναι μηχανισμοί για την πραγματοποίηση ενός εικονικού ιδιωτικού συμπλέγματος (όπως το SabalCore [66]) σε ένα περιβάλλον Cloud. Ο χρήστης του συστήματος εικονικού συμπλέγματος δεν βλέπει ότι χρησιμοποιεί ένα εικονικό σύμπλεγμα καθώς η υποβολή και η παρακολούθηση της εργασίας είναι πανομοιότυπη με

ένα φυσικό σύστημα συμπλέγματος. Η διαφορά είναι ότι η υλική υποδομή χωρίζεται σε υποσυστήματα ανά ομάδα και ότι μπορούν να προστεθούν επιπλέον υπολογιστές διακομιστές, όπως απαιτείται. Είναι επίσης δυνατό να υπάρχει υβριδική προσέγγιση με τοπικούς πόρους HPC και ουρές / προγραμματισμό και απομακρυσμένους πόρους Cloud HPC («κόμβοι εργαζομένων») συνδεδεμένους στο σύστημα.

3

Σύλληψη αρχιτεκτονικής και δομικών

εννοιών συστήματος

Ένα από τα κύρια αντικείμενα που μελετήθηκαν και αναπτυχθήκαν στα πλαίσια της παρούσας διατριβής, ήταν η δημιουργία μίας πλήρους αρχιτεκτονικής ενός συστήματος για την αξιολόγηση αλλά και την βελτίωση της ποιότητας υπηρεσίας και εμπειρίας σε υπηρεσίες υπολογιστικού νέφους. Η πληρότητα του συστήματος έγκριτε στην πολύπλευρη προσέγγιση της ανάλυσης των υπολογιστικών νεφών. Πιο συγκεκριμένα τα σύνολα εργαλείων καθώς και των λειτουργικών ομάδων που ανήκουν, δημιουργήθηκαν ώστε να καλύψουν τις ανάγκες όλων των εν δύναμη χρηστών του συστήματος. Στο κεφάλαιο 3.1 αναλύονται όλοι οι τύποι χρηστών καθώς και οι στόχοι που θέλουν να πετύχουν, έστω να δημιουργηθούν τα σύνολα των εργαλείων αφοσιωμένα για κάθε τύπο χρήστη. Στο κεφάλαιο 3.2 παρατίθεται η πρότυπη αρχιτεκτονική του συστήματος χωρισμένη σε λειτουργικές μονάδες ανά περίπτωση χρήσης ώστε να γίνει πιο εύκολα κατανοητή η συσχέτιση μεταξύ χρηστών εργαλείων και κατ' επέκταση στόχων του συστήματος. Η αλληλεπίδραση μεταξύ των στοιχείων του συστήματος αναλύεται στο κεφάλαιο 3.3 όπου περιέχει ένα πλήρες διάγραμμα οντοτήτων του συστήματος.

Είναι σημαντικό να αναφερθεί ότι στα πλαίσια της παρούσας διατριβής δεν αναπτύχθηκαν όλα τα εργαλεία και λογισμικά όπου χρησιμοποιήθηκαν στην αρχιτεκτονική. Ωστόσο η σύλληψη, διασύνδεση και τοποθέτηση των εργαλείων αυτών ώστε να δημιουργηθεί ένα πλήρες σύστημα, αποτελεί ένα από τα βασικά αποτελέσματα της παρούσας διατριβής, όπως παρουσιάζεται και στην σχετική δημοσίευση[3].

3.1 Ανάλυση ρόλων και στόχων συστήματος για την δημιουργία της αρχιτεκτονικής

Η διαδικασία που ακολουθήθηκε ώστε να δημιουργηθεί η αρχιτεκτονική του συστήματος ήταν η ανάλυση και η δημιουργία των στόχων καθώς και των χρηστών του συστήματος, όπου μετέπειτα καθοδήγησε τη δημιουργία των κύριων λειτουργικών μονάδων. Με αυτό τον τρόπο, μπορεί να δημιουργηθεί μια συσχέτιση μεταξύ των χρηστών του συστήματος των στόχων που θέλουν να επιτύχουν αυτοί οι χρήστες και στη συνέχεια ποιο κομμάτι λογισμικού θα περιέχει την κατάλληλη λειτουργικότητα ώστε να επιτευχθούν αυτοί οι στόχοι.

3.1.1 Ανάλυση στόχων και χρηστών συστήματος

Οι στόχοι που χρησιμοποιήθηκαν σαν εφαλτήριο για την ερευνά που συντάχθηκε στα πλαίσια της διατριβής καθώς και για την δημιουργία της πλήρης αρχιτεκτονικής του συστήματος είναι οι εξής:

- Βελτίωση των πρακτικών διαχείρισης πόρων από τους παρόχους Cloud (κυρίως IaaS) που οδηγούν σε βελτιωμένη σταθερότητα των προσφερόμενων υπηρεσιών και της ποιότητας υπηρεσίας για τους τελικούς χρήστες.

- Βελτίωση της δυνατότητας μέτρησης της συμπεριφοράς, την προβλεψιμότητα και τη δυνατότητα ελέγχου των υπηρεσιών Cloud, προκειμένου να αυξηθεί η εμπιστοσύνη προς αυτά τα συστήματα, επιτρέποντας έτσι τη χρήση τους σε πιο κρίσιμες εφαρμογές και ελαχιστοποιώντας το χρόνο που χρειάζονται κύριος μικρομεσαίες επιχειρήσεις (SMEs) να υιοθετήσουν λύσεις Cloud.
- Δημιουργία ενός αφαιρετικού επιπέδου πάνω από τα υπάρχοντα εργαλεία, το οποίο θα βοηθήσει στη χρήση τους με εύκολο τρόπο για ένα μεγάλο εύρος παρόχων υπηρεσιών Cloud.
- Ορισμός, υπολογισμός και χρήση μετρικών όπου θα έχουν νόημα για τους χρήστες ώστε να μπορούν να κατανοήσουν την αξιολόγηση της απόδοσης και της ποιότητας της υπηρεσίας των υπηρεσιών Cloud.
- Βελτίωση της ανταγωνιστικότητας των εφαρμογών που βασίζονται στο Cloud ελαχιστοποιώντας το κόστος ανά χρήστη, βελτιστοποιώντας τη διαδικασία επιλογής πόρων με βάση την κατηγοριοποίηση εφαρμογών και τη συγκριτική αξιολόγηση παρόχων στις ίδιες κατηγορίες για τη δημιουργία πολυεπίπεδων SLA.

Έχοντας παραθέσει τους στόχους που διέπουν και σχηματίζουν τη σύλληψη της αρχιτεκτονικής του συστήματος είναι σημαντικό να παρατεθούν και οι γενικότεροι χρήστες καθώς και οι ρόλοι που έχουν στην αρχιτεκτονική που αναπτύχθηκε. Ως αρχικοί ρόλοι υψηλού επιπέδου υποδεικνύονται οι ακόλουθες περιπτώσεις:

- **IaaS / Cloud Provider** (Πάροχος υλικών υποδομών ως υπηρεσία σε υπολογιστικά νέφη): Οντότητα που προσφέρει εικονικούς πόρους (κυρίως υπολογιστές, αποθήκευση και δίκτυο) ως υπηρεσία μέσω του Διαδικτύου στους πελάτες της. Στο πλαίσιο της διατριβής (και των εργαλείων που χρησιμοποιήθηκαν ή

αναπτυχθήκαν) επικεντρωνόμαστε κυρίως σε υπηρεσίες επιπέδου IaaS, οπότε εκτός αν αναφέρεται διαφορετικά, ο όρος Cloud αναφέρεται στο επίπεδο IaaS.

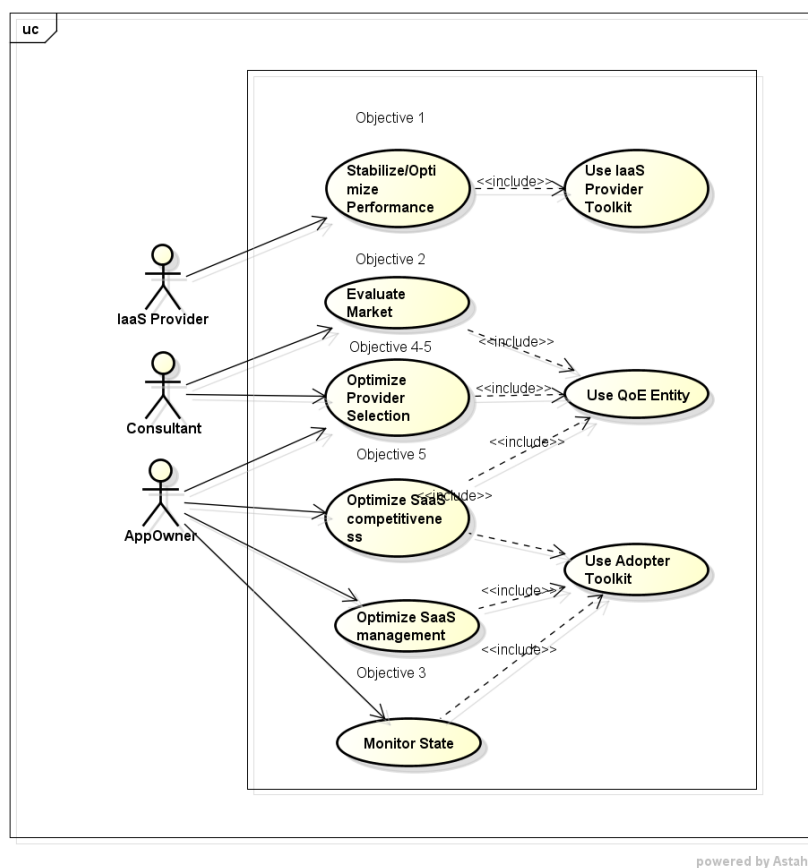
- **Application Owner** (Κάτοχος εφαρμογής): Οντότητα που διαθέτει συγκεκριμένο λογισμικό και θέλει να χρησιμοποιήσει τις υπηρεσίες Cloud προκειμένου να επιτρέψει τη λειτουργία της σε πιο δυναμικά επιχειρηματικά μοντέλα (όπως μετάβαση σε SaaS). Είναι απαραίτητο να τονιστεί ότι στο πλαίσιο των εργαλείων που χρησιμοποιήθηκαν ή αναπτυχθήκαν στην διατριβή, ο κύριος στόχος δεν είναι η αλλαγή της δομής της εφαρμογής, προκειμένου να λειτουργήσει ως SaaS (π.χ. μέσω της βοήθειας στη μετατροπή από μια μονολιθική εφαρμογή σε μια υπηρεσία προσανατολισμένη σε καταναλωτή αρχιτεκτονική κ.λπ.), αλλά το να βοηθήσει τον κάτοχο εφαρμογών στην επιλογή των κατάλληλων πόρων IaaS.
- **Consultant** (Σύμβουλος): Οντότητα που βοηθά άλλους ρόλους (όπως ο κάτοχος της εφαρμογής) κατά τη μετακόμισή τους στο Cloud και έχει ως κύριο ρόλο τη σύγκριση και την κατάταξη των υπηρεσιών Cloud για σύσταση στους πελάτες του.
- **QoE Entity** (Διαχειριστής συστήματος αξιολόγησης ποιότητας εμπειρίας): Αυτή η οντότητα εκτελεί τα εργαλεία QoE σε μια καθημερινή λειτουργία και παρέχει τα στατιστικά στοιχεία που χρειάζονται οι κάτοχοι εφαρμογών (ή οι αντίστοιχοι σύμβουλοι τους) για να επιλέξει τελικά μια κατάλληλη υπηρεσία IaaS.

3.1.2 Ανάλυση δομικών στοιχείων αρχιτεκτονικής

Με βάση την ανάλυση ρόλων καθώς και των στόχων στην προηγούμενη ενότητα, ορίστηκαν τρία βασικά εργαλεία / σετ εργαλείων. Τα εργαλεία αυτά δύναται να χρησιμοποιηθούν από τους χρήστες που ενστερνίζονται τους προαναφερθέντες ρόλους ώστε να πετύχουν τους προαναφερθέντες στόχους:

- Το σύνολο εργαλείων για Cloud Adopter (**CAT: Cloud Adopter Toolkit**)
- Το σύνολο εργαλείων για Provider IaaS (**IPT: IaaS Provider Toolkit**)
- Τα εργαλεία για QoE Entity (**QET: QoE Entity Toolkit**)

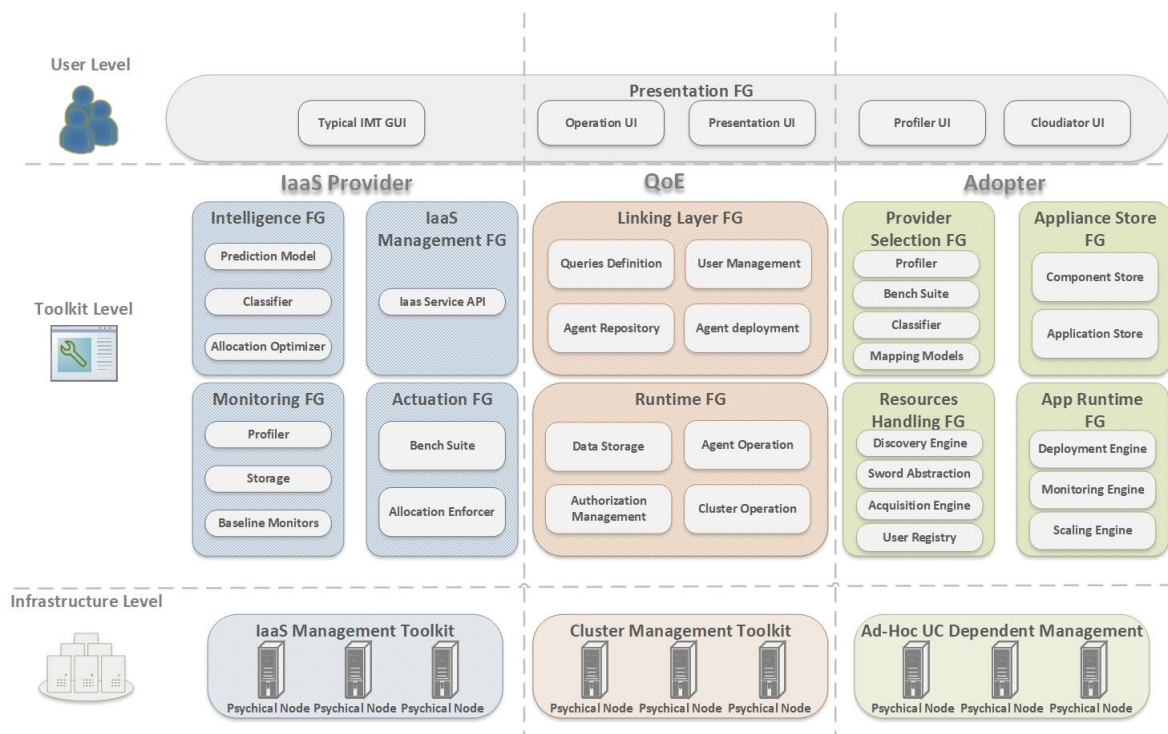
Είναι απαραίτητο να τονιστεί ότι αυτός ο διαχωρισμός έχει πραγματοποιηθεί με βάση κάθε ρόλο και οντότητα που δε χρειάζεται να ασχοληθεί με τις υπόλοιπες πτυχές εκτός από αυτές που έχουν άμεση ανάγκη και βάσει του διαχωρισμού που προβλέπεται από το μοντέλο Software Platform Infrastructure (SPI) [67], καθώς και το υποδεικνυόμενο επίπεδο διαθέσιμων πληροφοριών σε κάθε επίπεδο. Επιπλέον, η χρήση κάθε συνόλου εργαλείων είναι κυρίως ανεξάρτητη από τα άλλα σύνολα. Ένα εργαλείο μπορεί να υπάρχει σε περισσότερα από ένα σύνολα εργαλείων, με την ίδια έκδοση ή με μικρές τροποποιήσεις που είναι διαφανείς για τον χρήστη. Η αντιστοίχιση μεταξύ των ρόλων και των στόχων εμφανίζεται στην Εικόνα 3.



Εικόνα 3: Συσχέτιση μεταξύ ρόλων και στόχων συστήματος.

3.2 Πλήρης αρχιτεκτονική συστήματος δομημένη σε λειτουργικές ομάδες

Έχοντας αναλύσει το πώς οι στόχοι καθώς και οι χρήστες του συστήματος επηρεάζουν και δομούν την αρχιτεκτονική του συστήματος παραθέτετε σε αυτό το κεφάλαιο η πλήρης αρχιτεκτονική. Αναλυτικά, τα αρχιτεκτονικά μπλοκ αποτελούνται από τα ακόλουθα στοιχεία, χωρισμένα στις αντίστοιχες λειτουργικές ομάδες (Functional Groups: FG).



Εικόνα 4: Αρχιτεκτονική συστήματος[3].

3.2.1 Σύνολο εργαλείων παρόχου IaaS(IPT)

Presentation FG

Τυπικό GUI IaaS Management Tool (IMT) GUI: Αυτό το στοιχείο αναφέρεται σε τυπικά GUI που περιλαμβάνονται σε IMT όπως το OpenStack και το VMWare. Δεν προβλέπεται συγκεκριμένη επέκταση για αυτό, καθώς το σύνολο εργαλείων IPT αναμένεται να εκτελεστεί στο υπόβαθρο (daemon).

Intelligence FG

Prediction Models: Ο σκοπός αυτού του στοιχείου είναι να εξετάσει και να απεικονίσει με τη μορφή ενός μοντέλου πρόβλεψης πώς τα φυσικά μηχανήματα του IaaS Provider χειρίζονται τις διαφορετικές κατηγορίες εφαρμογών που εκτελούνται ταυτόχρονα (καθορίζονται μέσω των Benchmarks) και ποια υπολογιστική επιβάρυνση εισάγεται κατά

τη διάρκεια αυτής της εκτέλεσης λόγω της κοινής χρήσης ενός φυσικού κόμβου. Η γνώση από αυτό το στοιχείο μπορεί να προβλέψει τις παρεμβολές σε απόδοση ενός δεδομένου συνδυασμού κατηγοριών Benchmark / τύπων εφαρμογών και πρόκειται να χρησιμοποιηθεί στο Allocation Optimizer ως είσοδος.

Classifier: Αυτό το στοιχείο απαιτείται προκειμένου ο πάροχος να χαρτογραφήσει τον υπολογιστικό χαρακτήρα ενός αυθαίρετου VM σε μία από τις κατηγορίες Benchmark / εφαρμογών, για τις οποίες γνωρίζουν τις προβλεπόμενες παρεμβολές από το προηγούμενο βήμα. Ο λόγος για την ανάγκη αυτής της λειτουργικότητας είναι ότι οι πάροχοι IaaS προσφέρουν την υποδομή τους στους πελάτες τους (Adopters), αλλά δεν γνωρίζουν τον τύπο εφαρμογών που εκτελούνται εντός αυτών των VM. Επομένως, δεν μπορούν να εκμεταλλευτούν τα μοντέλα επιδόσεων που παράγονται στην αρχική φάση χωρίς να υποβληθούν σε μια τέτοια έμμεση διαδικασία χαρτογράφησης. Αυτό το στοιχείο χρησιμοποιεί επίσης πληροφορίες που προέρχονται από το στοιχείο Profiler, με τη μορφή διανυσμάτων χρήσης των υποκείμενων πόρων από τα VM, για τη σύγκριση ενός διανύσματος χρήσης αυθαίρετου εικονικού πόρου με τις κατηγορίες εφαρμογών μεμονωμένα διανύσματα ώστε να μπορέσει να δημιουργήσει μια συσχέτιση μεταξύ συνάφειας των τύπων εφαρμογών με γνώμονα την συμπεριφορά και το αποτύπωμα σε χρήση πόρων του συστήματος.

Allocation Optimizer: Έχοντας τις πληροφορίες (αντιστοιχισμένες στις κατηγορίες εφαρμογών όπως ορίζονται μέσω των αντίστοιχων Benchmark) από τις δύο προηγούμενες φάσεις, αναφορικά με το ποιες κατηγορίες εφαρμογών εκτελούνται και τι είδους παρεμβολές αυτές οι κατηγορίες προκαλούν η μία στην άλλη, ο πάροχος IaaS μπορεί τώρα να δει ολιστικά και αφηρημένα την εκχώρηση των εικονικοποιημένων πόρων στους φυσικούς κόμβους, έτσι ώστε η παρεμβολή να ελαχιστοποιηθεί. Αυτό οδηγεί σε ένα

σχέδιο κατανομής που υπαγορεύει τις κατανομές έτσι ώστε να μπορεί να επιτευχθεί σταθερότητα συμπεριφοράς.

Monitoring FG

Profiler: Αυτό το στοιχείο είναι υπεύθυνο για την εξαγωγή διανυσμάτων χρήσης των υποκείμενων μετρήσεων απόδοσης (όπως χρήση CPU, επισκέψεις cache / misses, I / O δίσκου, χρήση δικτύου κ.λπ.) και από τους εικονικούς πόρους του πελάτη που εκτελούνται στην υποδομή IaaS, καθώς και από εικονικούς πόρους που εκτελούν τα κριτήρια αναφοράς που αντιστοιχούν στις κατηγορίες εφαρμογών. Για την απόκτηση αυτών των πληροφοριών, ενδέχεται να συνεργαστεί με τις υπηρεσίες παρακολούθησης βάσης, ενώ για την εκκίνηση των αντίστοιχων σημείων αναφοράς κατά τη διαδικασία δημιουργίας προφίλ θα συνεργαστεί με το στοιχείο Bench Suite. Οι τελικές πληροφορίες (διανύσματα χρήσης π.χ. από ένα VM) προωθούνται στο στοιχείο Classifier.

Baseline Monitoring: Αυτό το στοιχείο απαιτείται για την εξαγωγή των βασικών πληροφοριών παρακολούθησης από τους φυσικούς κεντρικούς υπολογιστές του παρόχου IaaS, π.χ. ποια VM εκτελούνται και για τη συλλογή των βασικών μετρήσεων απόδοσης που απαιτούνται από τον profiler. Για τις ανάγκες αυτού του στοιχείου εργαλεία όπως το ceilometer, είτε κάποιο άλλο εργαλείο για την συλλογή δεδομένων από το φυσικό μηχάνημα μπορούν να χρησιμοποιηθούν.

Storage of Information: Αυτό το στοιχείο είναι υπεύθυνο για τη διατήρηση των πληροφοριών που αποκτήθηκαν από τη διαδικασία δημιουργίας προφίλ, έτσι ώστε να μην επαναλαμβάνονται συνεχώς. Έτσι, για παράδειγμα, τα διανύσματα των προφίλ από τις κατηγορίες εφαρμογών είναι ένας τύπος πληροφοριών που μπορεί να αποθηκευτεί. Επιπλέον, το ίδιο μπορεί να ισχύει (αν και ανανεώνεται περιοδικά) για έναν πελάτη VM,

διατηρώντας μια αντιστοίχιση μεταξύ του αναγνωριστικού προτύπου VM και της σχετικής κατηγορίας φορέα και / ή εφαρμογής.

Actuation FG

Benchmarking Suite: Αυτό το στοιχείο είναι υπεύθυνο για την εκκίνηση και την εκτέλεση των συγκριτικών αξιολογήσεων (Benchmarks) που αντιστοιχούν στις κατηγορίες εφαρμογών στο επίπεδο IaaS. Χρησιμοποιείται από το Profiler κατά την εξαγωγή των διανυσμάτων αναφοράς, για την εκκίνηση της εκτέλεσης τους.

Allocation Enforcer: Μετά την απόφαση σχετικά με το σχέδιο για την τοποθέτηση των εφαρμογών, όπως αποκτήθηκε από το Allocation Optimizer, αυτό το στοιχείο είναι υπεύθυνο για την πραγματική επικοινωνία με το IaaS Management Toolkit που χρησιμοποιήθηκε και την υποβολή αυτού του σχεδίου για εκτέλεση.

Infrastructure Level FG

IaaS Management Toolkit: Αυτό σχετίζεται με τους υποκείμενους φυσικούς πόρους που χρησιμοποιούνται αλλά κυρίως με το σύνολο εργαλείων διαχείρισης που τους χειρίζεται.

3.2.2 Σύνολο εργαλείων QoE Entity(QET)

Presentation FG

Operational UI: Το λειτουργικό περιβάλλον εργασίας εστιάζει σε πρωταρχικές λειτουργίες που σχετίζονται με τη λήψη πληροφοριών χρήστη, φόρμες εγγραφής και λειτουργικές ενέργειες όπως ενεργοποίηση σχετικών διαδικασιών παρακολούθησης / μέτρησης. Η καταγραφή της εισόδου χρήστη πρέπει να είναι δυναμική και να προσαρμόζεται στις διάφορες περιπτώσεις (π.χ. απαιτούμενες λεπτομέρειες διαμόρφωσης

ανά τύπο υπηρεσίας που χρειάζονται οι πράκτορες για την ανάπτυξη και παρακολούθηση ενός συγκεκριμένου τύπου υπηρεσίας).

Presentation UI: Το πεδίο αυτού του στοιχείου είναι η σαφής και συνοπτική παρουσίαση των αποτελεσμάτων παρακολούθησης και μέτρησης, σύμφωνα με τις αρχές της αλληλεπίδρασης ανθρώπου-υπολογιστή μέσω βελτιωμένων πτυχών οπτικοποίησης.

Linking Layer FG

User Management: Αυτό το στοιχείο είναι υπεύθυνο για τη διατήρηση πληροφοριών από την εγγραφή χρηστών, συμπεριλαμβανομένων των διαπιστευτηρίων λογαριασμού, των πληροφοριών χρέωσης και των τύπων λογαριασμού που επιλέγονται από τον τελικό χρήστη, διαχειριζόμενο τις διάφορες κατηγορίες λογαριασμών που θα καθοριστούν από το επιχειρηματικό μοντέλο QoE Entity.

Queries Definition: Αυτό το στοιχείο είναι υπεύθυνο για τον καθορισμό των αντίστοιχων ερωτημάτων που θα παρουσιαστούν στον τελικό χρήστη στο περιβάλλον εργασίας του χρήστη (presentation UI). Επομένως, πρέπει να δημιουργήσει τους απαραίτητους συνδυασμούς και ομαδοποιήσεις αποτελεσμάτων, για να παρέχει ουσιαστικές και συγκεντρωτικές πληροφορίες στον τελικό χρήστη, χρησιμοποιώντας διάφορες μετρήσεις ή / και εφαρμόζοντας λογική φιλτραρίσματος και ομαδοποίησης, για χρήση κατά την ανάκτηση δεδομένων από το στοιχείο αποθήκευσης δεδομένων του το Runtime FG.

Agent Deployment: Αυτό το στοιχείο είναι υπεύθυνο για την εκκίνηση της πακεταρισμένης έκδοσης των πρακτόρων που προβλέπονται για την συλλογή των μετρήσεις της οντότητας QoE (πράκτορες παρακολούθησης SLA και Bench Suite) σύμφωνα με τη λογική της λειτουργίας QoE. Για παράδειγμα, για κάθε τελικό χρήστη για τον οποίο θα παρακολουθείται το αντίστοιχο SLA τους, πρέπει να ξεκινήσει ένας σχετικός

πράκτορας, κατάλληλα διαμορφωμένος από την οντότητα QoE (ακολουθώντας επίσης πληροφορίες από τον τελικό χρήστη σε σχέση με ποιον πάροχο στόχου ή / και λογαριασμό) για τον αντίστοιχο πάροχο και τον τύπο υπηρεσίας που προσφέρει. Έτσι, αυτό το στοιχείο λειτουργεί επίσης ως σύνδεσμος μεταξύ της εισόδου χρήστη και των απαραίτητων εντολών για την παραμετροποίηση και λειτουργία της οντότητας QoE. Παρόμοιο είναι το πεδίο εφαρμογής στην περίπτωση των πρακτόρων Bench Suite, οι οποίοι πρέπει να ρυθμιστούν με βάση την υπηρεσία που καλείται να αξιολογήσει τον πάροχο και τον τύπο της συγκριτικής αξιολόγησης. Αυτό το στοιχείο είναι η υπηρεσία Docker REST API, σε συνδυασμό με τα κατάλληλα περιβάλλοντα(Docker) και αρχεία παραμετροποίησης(Docker Service files-Docker Secrets) για την

Agent Scheduler: Για την βελτιστοποίηση της λειτουργίας και την αυτοματοποίηση της συλλογής δεδομένων από τους πράκτορες για την επίβλεψη των SLA, καθώς και για τα Benchmarks δημιουργήθηκε το στοιχείο Agent Scheduler. Το στοιχείο αυτό έχοντας τις παραμετροποιήσεις καθώς και τους εγκατεστημένους πράκτορες από το προηγούμενο στοιχείο (Agent Deployment), μπορεί να εκτελεί περιοδικά αυτούς τους πράκτορες ώστε να κάνει την συλλογή των δεδομένων γρηγορότερη και απλούστερη, ιδιαίτερα για τους πράκτορες για τα Benchmarks που εκ φύσεως προϋποθέτουν πολλές εκτελέσεις ώστε να μπορέσει να εξαχθεί κάποιο συμπέρασμα για την απόδοση του IaaS.

Agent Repository: Αυτό το στοιχείο είναι υπεύθυνο για τη διατήρηση των εκδόσεων των πακεταρισμένων(containerized) πρακτόρων ώστε να μπορούν να είναι διαθέσιμοι και προσβάσιμοι από τα άλλα στοιχεία αυτής της λειτουργικής ομάδας (FG). Εντός της εφαρμογής κάθε πράκτορα, ενδέχεται να εκτελεστούν επιπλέον βήματα για να ολοκληρωθεί η προβλεπόμενη λειτουργικότητα του πράκτορα.

Runtime FG

Authorization Management: Αυτό το στοιχείο είναι υπεύθυνο για τον έλεγχο μεμονωμένων αιτημάτων για πρόσβαση σε δεδομένα από τη διαδικασία παρακολούθησης και κατά πόσον αυτά ταιριάζουν με τους αντίστοιχους λογαριασμούς που δημιουργήθηκαν και χρεώθηκαν μέσω του στοιχείου διαχείρισης χρήστη (User management). Αυτή η λειτουργικότητα θα πρέπει να παρέχεται είτε για ενέργειες που εκτελούνται μέσω της διεπαφής QoE είτε μέσω λειτουργιών που βασίζονται σε API, όπως και η αναμενόμενη χρήση αποτελεσμάτων συγκριτικής αξιολόγησης στις λειτουργίες προτάσεων του Provider Selection FG στο σύνολο εργαλείων Cloud Adopter.

Agent Operation: Αυτό το στοιχείο είναι υπεύθυνο για τη διατήρηση της λειτουργικής κατάστασης των πρακτόρων, όπως απαιτείται από τον στόχο τους. Συνεπώς, θα πρέπει να διασφαλίζει τη λειτουργική ακεραιότητα και τη συνέχιση της λειτουργίας για το συνολικό διάστημα που απαιτείται, καθώς και κατάλληλες διαχειριστικές ενέργειες σε περίπτωση αστοχίας. Πρόκειται για μια πτυχή που σχετίζεται κυρίως με μη λειτουργικές απαιτήσεις και μπορεί να αντιμετωπιστεί από την εσωτερική λειτουργικότητα του Cluster Management Toolkit (όπως γίνεται π.χ. στο Docker).

Data Storage: Αυτό το στοιχείο αντιπροσωπεύει το κύριο μπλοκ διαχείρισης δεδομένων της οντότητας QoE και είναι υπεύθυνο για τη διατήρηση των πρωτογενών δεδομένων που αποκτήθηκαν από τις εργασίες παρακολούθησης και συγκριτικής αξιολόγησης με δομημένο τρόπο. Αυτό το στοιχείο είναι η κύρια πηγή προς την οποία θα κατευθύνονται τα ερωτήματα από το περιβάλλον χρήστη (Presentation UI), όπως ορίζεται στο στοιχείο ορισμού ερωτημάτων (Queries Definition) του Linking Layer FG.

Cluster Operation: Αυτό το στοιχείο είναι υπεύθυνο για την έκθεση των φυσικών πόρων της οντότητας QoE και τη διαχείρισή τους προκειμένου να διασφαλιστούν πτυχές όπως κατανομημένη ανακάλυψη, εξισορρόπηση φορτίου και χειρισμός προσανατολισμένη στις υπηρεσίες. Μπορεί να βασίζεται σε λύσεις όπως το Docker Swarm API και λειτουργεί ως σύνδεσμος μεταξύ των εντολών που απαιτούνται για την ανάπτυξη του Agent και του Cluster Management FG.

Infrastructure Level FG

Cluster Management Toolkit: Αυτό σχετίζεται με τους υποκείμενους φυσικούς πόρους που χρησιμοποιούνται, αλλά κυρίως με το σύνολο των εργαλείων διαχείρισης που τους χειρίζεται (π.χ. Docker).

3.2.3 Σύνολο εργαλείων για Cloud Adopter(CAT)

Presentation FG

Clouadiator UI: Αυτό το στοιχείο παρέχει οπτικοποιημένη, βασισμένη σε REST, πρόσβαση στη λειτουργικότητα που προσφέρεται από την εφαρμογή του CAT. Συγκεκριμένα, υποστηρίζει την εγγραφή σε υπηρεσίες υπολογιστικού νέφους, καθώς και σύνδεση των λογαριασμών χρηστών για αυτά τα υπολογιστικά νέφη, πληροφορίες για τοποθεσίες, καθώς και για προσφερόμενες εικόνες και τύπους εικονικής μηχανής. Τέλος, παρέχει πρόσβαση στη δημιουργία και ανάπτυξη εφαρμογών και των υπηρεσιών τους.

Profiler UI: Αυτή η διεπαφή παρέχει οπτική χρήση για το εργαλείο Profiler που χρησιμοποιείται στην πλευρά του χρήστη που θέλει να υιοθετήσει υπηρεσίες νέφους(Cloud Adopter), για τη λήψη πληροφοριών σχετικά με το ποια διεργασία (π.χ. VM) και τη διασύνδεση δικτύου χρειάζεται για παρακολούθηση της διαδικασίας δημιουργίας προφίλ.

Provider Selection FG

Profiler: Αυτό το εργαλείο εκτελείται τοπικά για να εξαγάγει τα διανύσματα προφίλ της εφαρμογής. Η ίδια διαδικασία επαναλαμβάνεται για καθένα από τα χρησιμοποιημένα Benchmarks, εξάγοντας τα σχετικά διανύσματα για αυτά, σε συντονισμό με το στοιχείο Bench Suite.

Bench Suite: Η χρήση του στοιχείου Bench Suite σε αυτήν την περίπτωση αναφέρεται στην απόκτηση των διανυσμάτων προφίλ για τα Benchmarks στον ίδιο φυσικό κόμβο με το προφίλ της εφαρμογής. Αυτό γίνεται προκειμένου να ελαχιστοποιηθούν οι διαφορές που ενδέχεται να υπάρχουν λόγω διαφορετικών τύπων φυσικών κόμβων και που ενδέχεται να εμποδίσουν τη σωστή αναγνώριση στο στάδιο της ταξινόμησης.

Classifier: Αυτό το στοιχείο έχει την ίδια χρήση με το IPT, αλλά αυτή τη φορά για τα διανύσματα που ανακτήθηκαν κατά τη διάρκεια του τοπικού προφίλ στην πλευρά του Adopter. Το στοιχείο Classifier συγκρίνει το διάνυσμα της εφαρμογής με κάθε μεμονωμένο διάνυσμα αναφοράς (Benchmark) και αποφασίζει για το πιο παρόμοιο. Τέλος, ρωτά την οντότητα QoE για να πάρει την καλύτερη μετρημένη προσφορά υπηρεσιών (πάροχος & μέγεθος πόρου) με βάση το αποτέλεσμα της διαδικασίας ταξινόμησης. Αυτή η διαδικασία έχει ως σκοπό να μειώσει τον αριθμό των ενεργειών που απαιτούνται για τον προσδιορισμό του καταλληλότερου πάροχου και του βασικού μεγέθους πόρου για την εφαρμογή. Εάν αυτό δεν αντιστοιχιστεί σε μια ελαχιστοποιημένη, κοινή αναφορά για κατηγορίες εφαρμογών (όπως τα Benchmarks), τότε ο adopter θα πρέπει να εγκαταστήσει την εφαρμογή σε όλους τους συνδυασμούς παρόχων και μεγεθών προσφορών και στη συνέχεια να ολοκληρώσει.

Mapping Models: Αυτό το στοιχείο χρησιμοποιείται για συσχέτιση μεταξύ αναγκών σε πόρους εφαρμογής και αριθμού βασικών προσφερόμενων πόρων της υπηρεσίας του παρόχου IaaS, σε μια παρόμοια διαδικασία με αυτήν που προσδιορίστηκε στο [68].

Resource Handling FG

Discovery Engine: Αυτό το στοιχείο πραγματοποιεί προληπτικά και περιοδικά δημοσκοπήσεις όλων των γνωστών παρόχων Cloud για τη τρέχουσα έγκυρη προσφορά τους. Αυτό περιλαμβάνει τις διαθέσιμες περιοχές, τις διαθέσιμες ζώνες διαθεσιμότητας, τις εικόνες ειδώλου ανά περιοχή και τις προσφορές χρήστη και υλικού ανά περιοχή και χρήστη.

Sword Abstraction Library: Αυτό το παθητικό στοιχείο προσδίδει ένα επίπεδο αφαιρετικότητας πάνω από το API των διαφόρων παρόχων IaaS Cloud. Στον πυρήνα του είναι χτισμένο γύρω από τη βιβλιοθήκη Apache JClouds, αλλά παρέχει υποστήριξη για επιπλέον παρόχους Cloud, όπως εγκαταστάσεις FIWARE. Επιπλέον, διορθώνει πολλά σφάλματα και ελλείψεις του jClouds [69].

Acquisition Engine: Αυτό το στοιχείο εξυπηρετεί το σκοπό της απόκτησης εικονικών πόρων από παρόχους IaaS Cloud. Ενεργοποιείται με την πρόσβαση μέσω του περιβάλλοντος χρήστη του Cloudiator. Αυτό το αίτημα περιέχει τον πάροχο IaaS, το υλικό, την εικόνα και τη θέση της εικονικής μηχανής που θα δημιουργηθεί και στη συνέχεια συνδέεται με τον αντίστοιχο πάροχο χρησιμοποιώντας διαπιστευτήρια από το μητρώο Cloud του χρήστη. Μόλις αποκτήσει μια εικονική μηχανή, ο Cloudiator εγκαθιστά λογισμικό για τη διαχείριση της εικονικής μηχανής. Αυτό περιλαμβάνει λογισμικό για την ανάπτυξη εφαρμογών, καθώς και λογισμικό για την παρακολούθηση του φορτίου σε αυτό.

User Registry: Αυτό το στοιχείο διαχειρίζεται τη βάση χρηστών των χρηστών Cloudiator και τον αντίστοιχο μισθωτή τους. Για κάθε χρήστη / μισθωτή, αποθηκεύει επίσης διαπιστευτήρια Cloud που παρέχονται από τους χρήστες και τα συνδέει με το API αυτού του παρόχου Cloud.

Appliance Store FG

Component Store: Αυτό το στοιχείο λειτουργεί ως αποθετήριο για μεμονωμένα στοιχεία εφαρμογής. Αυτό μπορεί να περιλαμβάνει έναν διακομιστή ιστού, έναν εξισορροπητή φόρτωσης και μια βάση δεδομένων. Για καθένα από αυτά τα στοιχεία, αποθηκεύονται οι ακόλουθες πληροφορίες : (i) ποιες εισερχόμενες και εξερχόμενες θύρες παρέχονται και απαιτούνται από ένα στοιχείο, (ii) ένα μοναδικό όνομα και έκδοση και (iii) σενάρια χειρισμού κύκλου ζωής που βοηθούν το Application Runtime FG για τον έλεγχο της κατάστασης της εφαρμογής.

Application Store: Αυτό το στοιχείο λειτουργεί ως αποθετήριο για εφαρμογές. Οι εφαρμογές συντίθενται από στοιχεία στο χώρο αποθήκευσης. Μια εφαρμογή συνδέει εξερχόμενες θύρες από ένα στοιχείο με τις εισερχόμενες θύρες ενός άλλου στοιχείου. Μπορεί επίσης να προκαθορίσει ορισμένες επιλογές διαμόρφωσης και να θέσει περιορισμούς στον αριθμό μεμονωμένων στοιχείων.

Application Runtime FG

Deployment Engine : Αυτό το στοιχείο ενεργοποιείται κατόπιν αιτήματος χρήστη όταν πρόκειται να εγκατασταθεί μια εφαρμογή. Λαμβάνει ως παραμέτρους εισαγωγής τον αριθμό καθενός από τα στοιχεία αυτής της εφαρμογής, καθώς και τη θέση, δηλαδή το (εικονικό) μηχάνημα, όπου θα τοποθετηθεί μια συγκεκριμένη εφαρμογή. Για κάθε μία

από τις συνιστώσες, οι κύκλοι ζωής των στοιχείων υποβάλλονται σε επεξεργασία (προετοιμασία, λήψη, εγκατάσταση, ρύθμιση παραμέτρων, έναρξη).

Monitoring Engine : Αυτό το στοιχείο παρακολουθεί το φορτίο σε εικονικές μηχανές και στις εγκατεστημένες εφαρμογές τους. Αποθηκεύει αυτές τις πληροφορίες σε μια ιεραρχική αρχιτεκτονική παρακολούθησης. Οι ροές παρακολούθησης μπορούν να συγκεντρωθούν, να δειγματοληφθούν και να φιλτραριστούν.

Scalling Engine: Αυτό το στοιχείο αξιολογεί τις πληροφορίες παρακολούθησης και εφαρμόζει συγκεκριμένες ενέργειες κλιμάκωσης ή μετεγκατάστασης μόλις βρεθούν μοτίβα που καθορίζονται από το χρήστη στις ροές δεδομένων.

Infrastructure Level FG:

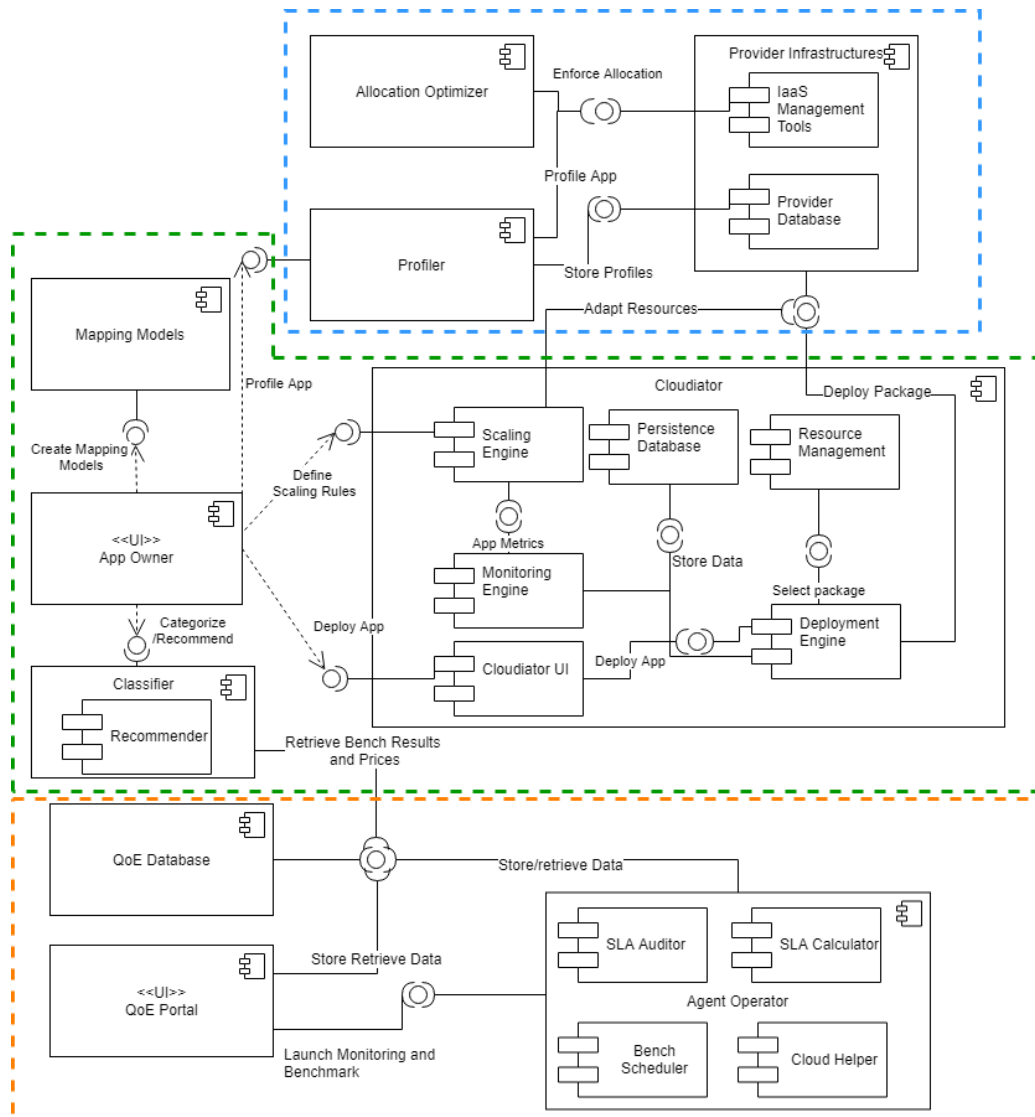
Ad Hoc UC Dependent Management: Αντιπροσωπεύει το τοπικό επίπεδο πόρων στην πλευρά του Adopter. Αυτό είναι ειδικό για κάθε περίπτωση και εξαρτάται από κάθε Σενάριο Περίπτωσης Χρήσης (UC). Αυτό το επίπεδο μπορεί να διαχειριστή με τους εσωτερικούς μηχανισμούς κάθε συστήματος UC ή μπορεί να χρειαστεί να ενσωματωθεί ένα συγκεκριμένο πρόγραμμα οδήγησης με το Resources Handling FG.

3.3 Αλληλεπίδραση μεταξύ στοιχείων του συστήματος

Όλα τα εργαλεία και τα συστατικά τους είναι ενσωματωμένα σε ένα πλήρες διάγραμμα αντικειμένων (Εικόνα 5), προκειμένου να απεικονιστούν καλύτερα όλες οι λειτουργίες και οι εξαρτήσεις των εργαλείων, όχι μόνο στο πλαίσιο των εργαλείων αλλά και ως μια ολοκληρωμένη λύση. Η λογική σύζευξη των εργαλείων μπορεί ακόμα να αναγνωριστεί στα διαγράμματα συστατικών με τα διακριτικά χρωματισμένα ορθογώνια:

- Μπλε ορθογώνιο: αντιπροσωπεύει το σύνολο εργαλείων για Provider IaaS

- Πράσινο ορθογώνιο: αντιπροσωπεύει το σύνολο εργαλείων για Cloud Adopter
- Πορτοκαλί ορθογώνιο: αντιπροσωπεύει το σύνολο εργαλείων QoE



Εικόνα 5: Διαγραμμα αντικειμένων συστήματος.

Αντικείμενα IaaS Provider

Όσον αφορά το σύνολο των εργαλείων του IaaS Provider (Εικόνα 5 - μπλε ορθογώνιο) για να πάρει ο Profiler όλα τα κατάλληλα διανύσματα χρήσης, χρησιμοποιεί το API που παρέχεται από την συστάδα εργαλείων IaaS. Όλα τα δεδομένα που προέρχονται από τον

Profiler αποθηκεύονται σε μια αποκλειστική βάση δεδομένων του παρόχου και όχι τοπικά για να διατηρηθεί η ακεραιότητα των δεδομένων, καθώς και για να είναι προσβάσιμα για απομακρυσμένη χρήση. Παρόλο που δεν παρουσιάζεται στο διάγραμμα συστατικών, επειδή ο Classifier συνδέεται λογικά και λειτουργικά με τον Profiler, αυτό το στοιχείο ανακτά τη λίστα VM που εκτελείται από τη βάση δεδομένων του παρόχου Cloud. Το στοιχείο Allocation Optimizer λαμβάνει την υπάρχουσα λίστα και κατηγορίες VM από τη βάση δεδομένων για να δοκιμάσει διάφορους συνδυασμούς (πλειάδες ή τριπλάσια των κατηγοριών VM) και στέλνει το βέλτιστο μοτίβο κατανομής στο Εργαλείο Διαχείρισης του IaaS Provider.

Αντικείμενα Cloud Adopter

Η συστάδα εργαλείων του Cloud Adopter δημιουργήθηκε για να συγκεντρώσει όλα τα προαναφερθέντα εργαλεία και λειτουργίες, για να βοηθήσει το Cloud Adopter να αξιολογήσει και να κατανοήσει τον υπολογιστικό χαρακτήρα και επίσης να αναπτύξει με τον κατάλληλο τρόπο την εφαρμογή του.

Ο Cloud Adopter μπορεί να χρησιμοποιήσει τα Mapping Models για να ορίσει συγκεκριμένους στόχους QoS που μεταφράζονται σε απαιτήσεις πόρων χαμηλού επιπέδου. Αυτές οι απαιτήσεις είναι χρήσιμες για τον χρήστη ώστε να κατανοήσει τη φύση της εφαρμογής και να δημιουργήσει ένα σταθερό σχέδιο κλιμάκωσης για την εφαρμογή του, προκειμένου να αποφευχθούν τυχόν υποβαθμίσεις στο QoS του μετά την μετοίκηση του Cloud.

Ο Cloud Adopter χρησιμοποιεί επίσης το εργαλείο Profiler και Classifier / Recommender για να κατηγοριοποιήσει την εφαρμογή του και να λάβει μια πρόταση υπηρεσίας ζητώντας από τη βάση δεδομένων QoE για την υπηρεσία με την καλύτερη απόδοση ενός

προκαθορισμένου τύπου και κατηγορίας. Προκειμένου να κλιμακωθεί σωστά χωρίς να χαθεί κανένα QoS, το εργαλείο κλιμάκωσης του Clouddiator επικοινωνεί με τη μονάδα παρακολούθησης. Εάν οι μετρήσεις που παράγονται από τη μηχανή παρακολούθησης υποδεικνύουν ότι υπάρχει ανάγκη για κλιμάκωση για τη διατήρηση του QoS, το εργαλείο κλιμάκωσης λαμβάνει τις κατάλληλες ενέργειες για να κλιμακώσει την εφαρμογή. Τέλος, για την ανάπτυξη της εφαρμογής τους σε μια προκαθορισμένη υπηρεσία, ο Κάτοχος της εφαρμογής χρησιμοποιεί το περιβάλλον εργασίας χρήστη Clouddiator που καλεί τις αντίστοιχες διεπαφές του Deployment Engine και Appliance Store, για την επιλογή του κατάλληλου πακέτου.

Αντικείμενα QoE

Η Βάση Δεδομένων QoE είναι μια βάση δεδομένων Dockerized mongo που είναι υπεύθυνη για την αποθήκευση όλων των δεδομένων που δημιουργούνται από τα εργαλεία και απαιτούνται για τη λειτουργία τους. Ο χρήστης μπορεί να έχει πρόσβαση σε όλες τις λειτουργίες των εργαλείων μέσω του QoE Portal. Το εργαλείο διαχείρισης του QoE περιέχει τέσσερα υπό-οντότητες: δύο υπεύθυνες για τη λειτουργία του Benchmarking Suite και δύο υπεύθυνες για τη λειτουργία του 3ALib. Οι υπό-οντότητες του Benchmarking Suite είναι το BenchScheduler που είναι υπεύθυνο για την ενορχήστρωση των δοκιμών αναφοράς και το δεύτερο είναι το CloudHelper που είναι μια οντότητα που δημιουργήθηκε για να βοηθήσει τον χρήστη να διαμορφώσει νέες εκτελέσεις αξιολόγησης απόδοσης ανακαλύπτοντας σημαντικές πληροφορίες σχετικά με τις υπηρεσίες του Cloud Provider που αφορούν την απόδοση των μισθωμένων πόρων. Όσον αφορά τις υπό-οντότητες του 3ALib, ο ελεγκτής SLA είναι υπεύθυνος για την ανάλυση των υπηρεσιών παροχής Cloud όσο αφορά την διαθεσιμότητα και άλλες μετρήσεις SLA όπως (CPU-Utilization), όλα τα αρχεία καταγραφής αποθηκεύονται στη βάση δεδομένων QoE. Μετά

την επιτυχή αποθήκευση η υπό-οντότητα υπεύθυνη για την επεξεργασία των δεδομένων (Log files) υπολογίζει με βάση συγκεκριμένων παραμέτρων κατά πόσο ο Cloud Provider τήρησε την συμφωνία επιπέδου ποιότητάς της Υπηρεσίας του (SLA).

4

Υλοποίηση συστήματος αξιολόγησης και βελτίωσης υπηρεσιών υπολογιστικού νέφους.

Στο παρόν κεφάλαιο παρουσιάζονται οι προγραμματιστικές λογικές καθώς και η τεχνική υλοποίηση του συστήματος αξιολόγησης και βελτίωσης των υπολογιστικών νεφών. Πριν την παράθεση όλων των τεχνικών πληροφοριών προγραμματισμού και υλοποίησης, είναι σημαντικό να παραταθούν οι στόχοι του κάθε υπο-συστήματος καθώς και το πως λειτουργίες του συστήματος και ενέργειες χρήστη καθιστούν δυνατή την εκπλήρωση αυτών των στόχων.

QoE-3ALib-Benchmarking Suite

Η τεχνική υλοποίηση καθώς και η λειτουργία αυτών των τριών εργαλείων, είναι πλήρως αλληλένδετη, ώστε να δημιουργηθεί το QoE Toolkit (κεφάλαιο 3.2.2). Για το λόγο αυτό οι στόχοι των εργαλείων αυτών καθώς και η αλληλεπίδραση των χρηστών με αυτά είναι κοινά.

Σκοπός	Λειτουργία	Ενέργειες Χρήστη
Αυτό-αξιολόγηση	Εκτέλεση εσωτερικών Benchmarks μέσα από την διεπαφή QoE.	Οι χρήστες είναι σε θέση να εκτελέσουν ή να προγραμματίσουν νέα τεστ συγκριτικής αξιολόγησης (Benchmarks) μέσω του QoE. Στη νέα συγκριτική αξιολόγηση αναφοράς, μπορούν να υπαγορεύσουν εάν τα αποτελέσματα των δοκιμών αναφοράς πρέπει να είναι ιδιωτικά ή δημόσια.
Αξιολόγηση λύσης υπηρεσιών Cloud	Πρόσβαση σε δεδομένα αξιολόγησης Cloud μέσω του QoE.	Οι χρήστες έχουν πρόσβαση σε δεδομένα συγκριτικής αξιολόγησης και δεδομένα παρακολούθησης και αξιολόγησης SLA. Αυτά τα δεδομένα υποβάλλονται σε επεξεργασία εμφανίζονται στον χρήστη με μορφή γραφημάτων και πινάκων προκειμένου να καταστεί δυνατή η κατανόηση αυτών καθώς και να γίνουν πιο ευδιάκριτες η διαφορές των ποικίλων συστημάτων υπολογιστικού νέφους που τέθηκαν υπό αξιολόγηση.
Εκτέλεση και αξιολόγηση αναλυτικών δεδομένων	Πρόσβαση σε τυποποιημένες μετρήσεις και αναλυτικά στοιχεία	Όλα τα αποτελέσματα που παράγονται από τα εργαλεία αναλύονται και , μέσω τυποποιημένων τεχνικών μετρήσεων και συγκεντρώσεως δεδομένων , παράγονται ομογενοποιημένα δεδομένα. Οι χρήστες έχουν πρόσβαση σε αυτά

	<p>βασισμένα στα δεδομένα παρακολούθησης και συγκριτικής αξιολόγησης στην πλατφόρμα QoE.</p>	<p>τα δεδομένα μέσω του QoE.</p>
<p>Παρακολούθηση SLA</p>	<p>Εκκίνηση του πράκτορα (Monitoring Service Agent) επίβλεψης SLA.</p>	<p>Οι χρήστες μπορούν να διαμορφώσουν νέους πράκτορες παρακολούθησης SLA μέσω του QoE. Οι χρήστες εισάγουν όλες τις παραμέτρους για το νέο πράκτορα σε μια φόρμα που παρέχεται μέσω μιας απλής διεπαφής. Όταν υποβάλλουν τη φόρμα , δημιουργείται ένας νέος πράκτορας SLA.</p>
<p>Βελτιστοποίηση επιλογής υπηρεσίας Cloud.</p>	<p>Πρόσβαση σε αξιολογήσεις, κόστους σε σχέση με ποιότητα υπηρεσιών χρησιμοποιώντας την εφαρμογή QoE.</p>	<p>Οι χρήστες έχουν τη δυνατότητα μέσω μιας σειράς ερωτήσεων και ρυθμίσεων να εντοπίσουν τις ανάγκες τους (τύπος εφαρμογής που έχουν) και τους παρουσιάζονται τα κατάλληλα αποτελέσματα. Αυτή η λειτουργία εκτελείται στο QoE και έχει δημιουργηθεί για να βοηθήσει τους χρήστες στην αξιολόγηση, παρέχοντάς τους πιο εξειδικευμένα αποτελέσματα για την αξιολόγηση των</p>

		υπηρεσιών Cloud.
Σύγκριση παρόχων Υπηρεσιών Cloud	πρόσβαση σε δεδομένα αξιολόγησης Cloud μέσω της εφαρμογής QoE	Η εφαρμογή QoE περιέχει δεδομένα από την παρακολούθηση και τη συγκριτική αξιολόγηση διαφορετικών παρόχων Cloud.
Αξιολογήστε τις δυνατότητες VM	Εκτέλεση Benchmarks μέσα από την διεπαφή QoE για συγκεκριμένα VM που βρίσκονται σε κάποια Cloud υποδομή.	Οι χρήστες μπορούν να παραμετροποιήσουν μέσα από την διεπαφή του QoE καινούρια Benchmarks ώστε να αξιολογήσουν υπηρεσίες Cloud που ενδιαφέρονται να χρησιμοποιήσουν. Αυτή η διαδικασία αρχικά χρειαζόταν εκτενή παραμετροποίηση από τον χρήστη αλλά για το λόγο αυτό δημιουργήθηκε ένα σύνολο από ερωτήσεις ώστε να βοηθήσουν τον χρήστη να παραμετροποιήσει ένα καινούριο Benchmark συμπληρώνοντας μόνο κάποιες βασικές πληροφορίες.

Πίνακας 1: Αντιστοίχιση στόχων χρηστών με λειτουργίες συστήματος για QoE Toolkit.

Profiling and Classification

Στην συνέχεια ακολουθεί ο πίνακας που αφορά την αντιστοίχιση στόχων και λειτουργιών για το εργαλείο που είναι υπεύθυνο για δημιουργία προφίλ και κατηγοριοποίηση εφαρμογών. Η δημιουργία προφίλ καθώς και η κατηγοριοποίηση των εφαρμογών αποτελεί σημαντικό βήμα για την προσαρμογή τους σε περιβάλλοντα Cloud, καθώς αυτό το

εργαλείο δίνει μια επαρκή αρχική ένδειξη των αναγκών μιας εφαρμογής, ώστε να γίνει όσο τον δυνατόν πιο βέλτιστη η αρχική τοποθέτηση αυτής σε ένα περιβάλλον Cloud.

Σκοπός	Λειτουργία	Ενέργειες Χρήστη
Βελτιστοποίηση της διαχείρισης SaaS	Δημιουργία Προφίλ της εφαρμογής	Το προφίλ της εφαρμογής όταν είναι εγκατεστημένο σε ένα VM παράγει ένα αρχείο που περιέχει μετρήσεις σχετικά με τη χρήση των πόρων για μια δεδομένη εφαρμογή. Αυτό το αρχείο μπορεί να βοηθήσει τον ταξινομητή να χαρτογραφήσει την εφαρμογή σε ένα συγκεκριμένο σημείο αναφοράς. Με αυτόν τον τρόπο οι κάτοχοι εφαρμογών μπορούν να έχουν μια βασική ένδειξη των πόρων που χρειάζονται για τη λειτουργία της εφαρμογής.
Βελτιστοποίηση της ανταγωνιστικότητας SaaS	Προφίλ και ταξινόμηση της εφαρμογής	Αυτός ο στόχος προέρχεται από τον παραπάνω (Βελτιστοποίηση διαχείρισης SaaS). Η δημιουργία προφίλ και η ταξινόμηση μιας εφαρμογής μπορεί να βοηθήσει τον κάτοχο της εφαρμογής να επιλέξει την καταλληλότερη υπηρεσία, λαμβάνοντας υπόψη τις ανάγκες της εφαρμογής τους, προκειμένου να αποφευχθεί τυχόν περιττό κόστος. Για παράδειγμα, η

		επιλογή μιας υπηρεσίας πολύ δαπανηρή χωρίς να χρειάζεται πραγματικά μεγάλη πολύ υπολογιστική ισχύ ή η επιλογή μιας υπηρεσίας που δεν είναι ικανή να αντέξει το φορτίο που παράγει η εφαρμογή.
Σταθεροποίηση / Βελτιστοποίηση απόδοσης	Χρήση του ταξινομητή για κατηγοριοποίηση των VM που τρέχουν ήδη.	Ο πάροχος IaaS αποκτά πρόσβαση στη διεπαφή χρήστη των εργαλείων προφίλ και ταξινόμησης προκειμένου να ανακτήσει τα διανύσματα προφίλ της εφαρμογής που εκτελούνται στην υποδομή και στη συνέχεια, να ταξινομήσει την εφαρμογή. Ο διαχειριστής IaaS μπορεί επίσης να αποθηκεύσει την ταξινόμηση για μελλοντική χρήση.

Πίνακας 2: Αντιστοίχιση στόχων χρηστών με λειτουργίες συστήματος για τα εργαλεία Profiling and Classification.

Interference and Mapping Models

Ο πίνακας που ακολουθεί αφορά την αντιστοίχιση στόχων και λειτουργιών για τα εργαλεία που είναι υπεύθυνα για την δημιουργία μοτίβων χρήσης των πόρων του συστήματος. Αυτά τα μοτίβα είναι απαραίτητα ώστε να μπορέσουν να δημιουργηθούν συμπεράσματα για τις ανάγκες εφαρμογών σε υπολογιστικούς πόρους ώστε να πετύχουν την επιθυμητή ποιότητα υπηρεσίας.

Σκοπός	Λειτουργία	Ενέργειες Χρήστη
Αποτελεσματική κατανομή VM	Χρήση το UI του εργαλείου Interference and Mapping Models, για την εκπαίδευση ενός ANN και δεύτερον για την ανάκτηση προβλέψεων.	Οι χρήστες πρέπει πρώτα να δημιουργήσουν ένα κατάλληλο σύνολο δεδομένων για να εκπαιδεύσουν τα ANN. Μέσω της διεπαφής χρήστη μπορούν να εκπαιδεύουν και να αποθηκεύουν, για μελλοντική χρήση, το πιο κατάλληλο ANN. Αφού εκπαιδευτεί μία φορά, το αντίστοιχο μοντέλο θα παρέχει στο χρήστη προβλέψεις παρεμβολών.
Αντιστοίχιση φόρτου εργασίας και ποιότητας υπηρεσίας και υπολογιστικούς πόρους	Χρήση του εργαλείου Interference and Mapping Models για την εκπαίδευση του μοντέλου πρόβλεψης, ώστε σε μεταγενέστερο επίπεδο να μπορούν να γίνουν οι αντιστοιχίσεις φόρτου-αναγκαίων πόρων.	Οι χρήστες πρέπει πρώτα να δημιουργήσουν ένα κατάλληλο σύνολο δεδομένων για να εκπαιδεύσουν τα ANN. Μέσω της διεπαφής χρήστη θα είναι σε θέση να εκπαιδεύσουν και να αποθηκεύσουν για μελλοντική χρήση, το πιο κατάλληλο ANN. Αφού εκπαιδευτεί, το αντίστοιχο μοντέλο θα παρέχει στο χρήστη προβλέψεις σχετικά με τις παραμέτρους QoS της εφαρμογής.

Πίνακας 3: Αντιστοίχιση στόχων χρηστών με λειτουργίες συστήματος για τα εργαλεία

Interference and Mapping Models.

4.1 Υλοποίηση υποσυστήματος ελέγχου επιπέδου ποιότητας υπηρεσίας (SLA monitoring subsystem).

Στο παρόν κεφάλαιο παρουσιάζεται το πλήρες θεωρητικό και ερευνητικό υπόβαθρό που οδήγησε στην υλοποίηση των εργαλείων για την παρακολούθηση και αξιολόγηση των Cloud SLA. Πιο συγκεκριμένα παρουσιάζονται οι τυποποιημένες προδιαγραφές όπου ήταν απαραίτητες ώστε να αξιολογηθεί ένα SLA με βάση τον ορισμό του κάθε παρόχου Cloud υπηρεσιών. Επίσης παρουσιάζεται η αρχιτεκτονική και η αλληλεπίδραση των εργαλείων ώστε να επιτευχθεί η απαραίτητη λειτουργικότητα.

4.1.1 Προδιαγραφές και τυποποίηση αξιολόγησης Cloud SLA

Ο έλεγχος καθώς και αξιολόγηση ενός SLA που αφορά υπηρεσίες υπολογιστικού νέφους περιέχει ποικίλες προκλήσεις. Ότι αφορά την αξιολόγηση της απόδοσης καθώς και την επίβλεψη των συστημάτων νέφους, έχουν δημιουργηθεί μια πλήθρα εργαλείων όχι μόνο από εξωτερικούς οργανισμούς αλλά και από τους ίδιους τους παρόχους των υπηρεσιών αυτό (πχ. CloudWatch για τις υπηρεσίες νέφους της Amazon). Η ειδοποιός διαφορά μεταξύ της επίβλεψης καθώς και της αξιολόγησης πόρων υπολογιστικού νέφους και SLA μιας υπηρεσίας νέφους, είναι το ότι το SLA αποτελεί ένα συμβόλαιο μεταξύ του χρήστη της υπηρεσίας και του παρόχου αυτής. Όντας ένα συμβόλαιο και όχι μια αυστηρά ορισμένη μετρική αξιολόγησης (πχ. Latency, Response Time CPU Utilization, κλπ.) είναι αναγκαία η δημιουργία και η χρήση μιας τυποποιημένης προδιαγραφής όπου θα ορίζει τον τρόπο με τον οποίο ένα SLA όπου ορίζεται σαν ένα συμφωνητικό σε μορφή κειμένου μπορεί να ποσοτικοποιηθεί ώστε να γίνει δυνατή και η αξιολόγηση του, αλλά και η σύγκρισή του με άλλα συμβόλαια παρόχων υπηρεσιών νέφους. Στην παρούσα διατριβή

χρησιμοποιήθηκε μια τέτοια προδιαγραφή (ISO 19086- 2 [70]) ώστε να δημιουργηθεί ένα τυποποιημένο εργαλείο αξιολόγησης υπηρεσιών Cloud. Συγκεκριμένα για την ανάπτυξη του λογισμικού καθώς και την προσαρμογή των συμβολαίων SLA σε τυποποιημένο σχήμα δεδομένων χρησιμοποιήθηκαν τρία κύρια κομμάτια της προδιαγραφής:

- Η ανάλυση και ενσωμάτωση της προδιαγραφής καθώς και του μοντέλου αναφοράς αυτής στην πλατφόρμα.
- Η ανάλυση και ενσωμάτωση των μετρικών για την αξιολόγηση των υπηρεσιών Cloud.
- Η μελέτη της συμβατότητας καθώς και της εξελιξιμότητας της προδιαγραφής.

Ανάλυση προδιαγραφής

Το μοντέλο προδιαγραφών και αναφοράς δημιουργήθηκε με σκοπό την τυποποίηση του ορισμού των ρητρών SLA με τρόπο που εξυπηρετεί ολόκληρο τον κύκλο ζωής των SLA για υπηρεσίες Cloud και υπερνικά τα μειονεκτήματα των περιορισμένων υπάρχουσών προσεγγίσεων, εξαλείφοντας ασάφειες στον ορισμό και υπολογισμό των ρητρών SLA και διευκόλυνση της μέτρησης, της παρακολούθησης και της επιβολής των SLA για την επίτευξη ακεραιότητας, έτσι ώστε αυτές οι μετρήσεις να μην μπορούν να αμφισβητηθούν. Ένας άλλος στόχος ήταν να αφαιρεθούν όσο το δυνατόν περισσότερο οι ορισμοί των ρητρών SLA, έτσι ώστε να καταστεί δυνατή η εφαρμογή μετρήσεων που επιτρέπουν την άμεση συγκρισιμότητα των ρητρών SLA μεταξύ των παρόχων. Το μοντέλο αναφοράς είναι συμβατό με το ISO, χρησιμοποιώντας τις κλάσεις και τις παραμέτρους του μετρικού μοντέλου ISO 19086-2, αλλά επιπλέον επιτρέπει την παρουσίαση μιας κλάσης δειγματοληψίας για τον ορισμό της διαδικασίας δειγματοληψίας της ρήτρας SLA.

Επιπλέον, όλες οι ρήτρες SLA που ορίζονται μέσω του μοντέλου είναι κατανοητές από μηχανή. Οι κλάσεις όπου χρησιμοποιήθηκαν είναι οι κάτωθι:

Μετρικές: Το μπλοκ μετρικών αντιστοιχεί τις μέτρηση σε σχέση με τους στόχο υπηρεσίας (π.χ. διαθεσιμότητα). Κάθε μέτρηση ορίζεται μέσω τυποποιημένων ορισμών μετρικών, συμπεριλαμβανομένων των βασικών πληροφοριών που είναι απαραίτητες για την κατανόηση της μέτρησης μιας ιδιότητας που πρέπει να παρατηρηθεί.

Παράμετροι: Το μπλοκ παραμέτρων συνδέει τη μέτρηση με ένα σύνολο παραμέτρων που πρέπει να συνοδεύονται με τις μετρικές (εκφράζοντας λεπτομερώς κάθε μέτρηση). Οι παράμετροι περιλαμβάνουν τον τρόπο με τον οποίο πρέπει να εκφραστεί η μέτρηση (π.χ. float, ακέραιος αριθμός), τι πρέπει να περιμένει ο χρήστης από τη συγκεκριμένη μέτρηση του SLA και πώς διαφορετικές πτυχές ποσοτικοποιούν τις αντίστοιχες μετρήσεις.

Κανόνες: Το μπλοκ κανόνων αναφέρεται στους περιορισμούς που υπάρχουν κατά την λήψη μετρήσεων (π.χ. αριθμός ταυτόχρονων συνδέσεων για έναν αριθμό μετρήσεων χρηστών), ως στοιχεία που χρησιμοποιούνται για τον περαιτέρω περιορισμό ορισμένων τμημάτων κάθε μέτρησης και υποδεικνύουν πιθανές μεθόδους μέτρησης. Έτσι, για κάθε μέτρηση θα πρέπει να περιγράφονται οι προτεινόμενοι γενικευμένοι κανόνες της, συμπεριλαμβανομένων όλων των πιθανών περιπτώσεων. Βάσει ενός δεδομένου SLA οι κανόνες που χρησιμοποιήθηκαν είναι οι εξής:

- Ανάπτυξη σε διαφορετικές ζώνες διαθεσιμότητας
- Ενεργοποίηση συγκεκριμένων χαρακτηριστικών όπως επιλογές αναπαραγωγής
- Περιορισμός αιτημάτων σε περίπτωση μη διαθεσιμότητας
- Προγραμματισμένη διακοπή λειτουργίας συντήρησης

Ανάλυση Μετρικών

Υπάρχει μια πληθώρα από μετρικές που μπορούν δυνητικά να υπάρξουν και να εξαχθούν από ένα SLA [71]. Οι σημαντικότερες εξ αυτών είναι:

- Availability
- Elasticity
- Response Time
- CPU Utilization
- Disk Space

Ωστόσο η σημαντικότητα αυτών των μετρικών πηγάζει από τις ανάγκες των χρηστών και όχι από τα εγγέγραφα που δίνουν μέσω των SLA οι πάροχοι υπηρεσιών Cloud. Οι διαφορές μεταξύ της οπτικής γωνίας με την οποία αντιλαμβάνονται οι χρήστες και οι πάροχοι υπηρεσιών Cloud, παρατίθενται στο παρακάτω πίνακα Πίνακας 4.

Μετρικές Παρόχου Cloud Service	Μετρικές Χρήστη Cloud Service
<p>Οι πάροχοι προτιμούν γενικά τις μετρήσεις διαθεσιμότητας που δείχνουν τον Πάροχο με τον πιο θετικό τρόπο. Αυτό σημαίνει ότι τα ακόλουθα τείνουν να αποτελούν προτεραιότητες για τον Πάροχο:</p> <ul style="list-style-type: none"> • Ικανότητα Ελέγχου: Οι πάροχοι θα θελήσουν να αποφύγουν μετρήσεις που μπορεί να επηρεαστούν από παράγοντες πέραν του άμεσου ελέγχου τους, όπως η διαθεσιμότητα του δικτύου όταν δεν μπορούν να το ελέγξουν. Αντίθετα, η 	<p>Οι Χρήστες προτιμούν γενικά μετρήσεις που έχουν τα ακόλουθα χαρακτηριστικά:</p> <ul style="list-style-type: none"> • Μέτρηση στο τελικό προορισμό: Οι Χρήστες γενικά θα επιθυμούν να μετρήσουν την απόδοση στο σημείο που καταναλώνουν την υπηρεσία, χωρίς αναλύσεις ανά στοιχείο που θα μπορούσαν να υποδηλώνουν ικανοποιητική απόδοση όταν συνολικά δεν υπάρχει. • Ευθύνη αναφοράς Παρόχου: Οι Χρήστες

<p>έμφαση δίνεται στις μετρήσεις που μπορούν να μετρηθούν εξ ολοκλήρου στις εγκαταστάσεις τους. Ο ορισμός των μετρήσεων ανά στοιχείο (π.χ. αποθήκευση, υπολογισμός) είναι ένας άλλος τρόπος για να γίνουν οι μετρήσεις πιο ελεγχόμενες και προβλέψιμες.</p> <p>• Μετρησιμότητα: Οι πάροχοι θα θέλουν συνήθως να αναφέρουν τη διαθεσιμότητα βάσει των κριτηρίων που είναι ευκολότερα για τη μέτρησή τους, και ενδεχομένως επίσης που παρέχουν τη μικρότερη συγκρισιμότητα με άλλους παρόχους, για λόγους ανταγωνισμού.</p> <p>•Επίδραση: Οι μικρές αποτυχίες υπηρεσιών γενικά δεν έχουν σημαντικό αντίκτυπο στον πελάτη, και επομένως απαιτείται ένα κατώτατο όριο για να καθοριστεί εάν μια εξαίρεση υπηρεσίας προκαλεί σημαντικό αντίκτυπο. Συνήθως αυτό επιτυγχάνεται με την απαίτηση ότι η εξαίρεση υπηρεσίας παραμένει για μια καθορισμένη περίοδο. Ενδεχομένως, ο</p>	<p>γενικά θα επιθυμούν να αναφέρονται αυτόματα μετρήσεις και εξαιρέσεις από τον πάροχο, με αυτόματη επεξεργασία κυρώσεων.</p>
--	---

<p>ορισμός μπορεί να απαιτεί συνεχή εξαίρεση υπηρεσίας κατά τη διάρκεια αυτής της περιόδου, η οποία μπορεί να είναι αδύνατο να αποδειχθεί λόγω της περιοδικής φύσης των μετρήσεων.</p> <p>• Επιπτώσεις που αναγνωρίζονται από τον πελάτη-χρήστη: Ο πάροχος δεν πρέπει να τιμωρείται για μικρές εξαιρέσεις υπηρεσιών που συμβαίνουν όταν ο πελάτης δεν χρησιμοποιεί πραγματικά το σύστημα. Ο ευκολότερος τρόπος για την επίτευξη αυτού του στόχου είναι να βάλουμε τα έξοδα στον πελάτη για τον εντοπισμό εξαιρέσεων υπηρεσιών.</p>	
---	--

Πίνακας 4: Ανάλυση μετρικών με βάση αναγκών χρήστη/παρόχου Cloud Service

Η μετρική της διαθεσιμότητας (Availability) αποτελεί μία περίπτωση όπου πληροί σε μεγάλο βαθμό τα κριτήρια των Παρόχων υπηρεσιών Cloud (Πίνακας 4). Για το λόγο αυτό η διαθεσιμότητα είναι μια από τις πρώτες και κύριες μετρικές όπου έχουν εισαχθεί στα SLA των Public Cloud Παρόχων, όπως η Amazon AWS [72]. Αυτή η μετρική μπορεί να χρησιμοποιηθεί για οποιοδήποτε επίπεδο της στοίβας σύννεφων, π.χ. IaaS, PaaS και SaaS. Υπάρχει συγκριτικά περιορισμένη συμφωνία σχετικά με τον τρόπο καθορισμού αυτής της μετρικής (διαθεσιμότητας), με διαφορές που εντοπίζονται συνήθως μεταξύ του τρόπου μέτρησης και του τι εξαιρείται από τον υπολογισμό, συχνά ανάλογα με τον Πάροχο και το

επίπεδο στο οποίο βρίσκεται η υπηρεσία (VM, αποθήκευση, πλατφόρμα, βάση δεδομένων κ.λπ.). Για παράδειγμα, μπορεί να υπάρχουν μετρήσεις διαθεσιμότητας που ορίζονται από την άποψη των χρόνων απόκρισης ή από την άποψη συγκεκριμένων απαντήσεων σφαλμάτων κατά την απόπειρα χρήσης μιας υπηρεσίας Cloud (π.χ. λήψη συγκεκριμένης απόκρισης σφάλματος κατά την προσπάθεια πρόσβασης σε μια βάση δεδομένων Cloud).

Είναι δυνατό να καθοριστεί η διαθεσιμότητα για συγκεκριμένα στοιχεία διαθεσιμότητας, π.χ. για υπολογιστική διαθεσιμότητα υπηρεσίας, διαθεσιμότητα υπηρεσίας αποθήκευσης και διαθεσιμότητα δικτύου. Ωστόσο, η συνολική διαθεσιμότητα όπως φαίνεται από τον τελικό χρήστη είναι συνήθως η πιο σημαντική μετρική. Για να μπορέσει να μετρηθεί αυτή η συγκεκριμένη μετρική πρέπει να γίνουν οι εξής παραδοχές όσον αφορά τον τρόπο με τον οποίο γίνεται η συλλογή των δεδομένων:

Αποδεκτή διακοπή λειτουργίας: θεωρούμε ότι η αποδεκτή διακοπή λειτουργίας πρέπει να πληροί δύο προϋποθέσεις: πρέπει να προγραμματιστεί και να κοινοποιηθεί στον χρήστη λογικά εκ των προτέρων. Ο προγραμματισμός εκ των προτέρων για μία εβδομάδα πρέπει να είναι λογικά περιορισμένο. Ο περιορισμός του προγραμματισμένου χρόνου διακοπής στο μέγιστο 5% του συνολικού αναμενόμενου συμβατικού χρόνου πρέπει να αναμένεται.

Σαφής προδιαγραφή της διαδικασίας μέτρησης. Θεωρούμε ότι οι απαραίτητες πληροφορίες για το χρήστη που παρακολουθεί το SLA θα πρέπει να ορίζεται πλήρως στο SLA με αυστηρό τρόπο. Αυτές συμπεριλαμβάνουν τις μετρικές, την περιοδικότητα με την οποία συλλέγονται τα δεδομένα καθώς και τον τρόπο (formula) υπολογισμού της εκάστοτε μετρικής.

Κύκλος χρέωσης. Οι πάροχοι θεωρούν συνήθως ότι ο συνολικός μήνας πρέπει να λαμβάνεται υπόψη για τον υπολογισμό του διαθέσιμου χρόνου, ανεξάρτητα από το εάν οι υπηρεσίες χρησιμοποιούνται από τους Χρήστες.

Τυποποίηση Μετρικής Διαθεσιμότητας (Availability)

Η πιο σημαντική μετρική για τον χρήστη όσον αφορά τις περισσότερες υπηρεσίες Cloud είναι η διαθεσιμότητα της υπηρεσίας Cloud, εάν η υπηρεσία είναι προσβάσιμη για χρησιμοποίηση από τον τελικό χρήστη. Αυτή η μετρική μπορεί να χρησιμοποιηθεί για οποιοδήποτε επίπεδο της στοίβας των Cloud υπηρεσιών, π.χ. IaaS, PaaS και SaaS. Υπάρχει συγκριτικά περιορισμένη συμφωνία σχετικά με τον τρόπο καθορισμού αυτής της μετρικής, με διαφορές που εντοπίζονται συνήθως μεταξύ του τρόπου μέτρησης και του τι εξαιρείται από τον υπολογισμό, συχνά ανάλογα με τον Πάροχο και το επίπεδο στο οποίο βρίσκεται η υπηρεσία (VM, αποθήκευση, πλατφόρμα, βάση δεδομένων κ.λπ.). Για παράδειγμα, μπορεί να υπάρχουν μετρήσεις διαθεσιμότητας που καθορίζονται από την άποψη των χρόνων απόκρισης(response time) ή από την άποψη συγκεκριμένων απαντήσεων σφαλμάτων (error responses) κατά την απόπειρα χρήσης μιας υπηρεσίας Cloud (π.χ. απόκτηση συγκεκριμένης απόκρισης σφάλματος κατά την προσπάθεια πρόσβασης σε μια βάση δεδομένων Cloud). Είναι δυνατό να καθοριστεί η διαθεσιμότητα για συγκεκριμένα στοιχεία διαθεσιμότητας, π.χ. για υπολογιστική διαθεσιμότητα υπηρεσίας, διαθεσιμότητα υπηρεσίας αποθήκευσης και διαθεσιμότητα δικτύου. Ωστόσο, η συνολική διαθεσιμότητα όπως φαίνεται από τον τελικό χρήστη είναι συνήθως η πιο σημαντική μετρική. Για να μπορέσει να δημιουργηθεί το μοντέλο υπολογισμού της μετρικής πρέπει να ληφθούν υπόψιν οι παρακάτω διευκρινίσεις για τον τρόπο συλλογής και υπολογισμού Cloud SLA:

- Η μέτρηση πρέπει να βασίζεται σε συμφωνημένη συναλλαγή χρησιμοποιώντας συμφωνημένο πρωτόκολλο. Υπάρχουν διαφορετικές επιλογές, αλλά με διαφορετικά πλεονεκτήματα και μειονεκτήματα ανά περίπτωση (π.χ. το ICMP μπορεί να αποτελεί απειλή για την ασφάλεια, το HTTP μπορεί να περιλαμβάνει σφάλματα διακομιστή εφαρμογών, τα οποία δεν είναι ευθύνη των παρόχων IaaS κ.λπ.)
- Πρέπει να υπάρχει συμφωνημένο διάστημα μεταξύ των μετρήσεων. Αυτό υπαγορεύεται από τον Πάροχο (και ενδεχομένως από την ικανότητα του τελευταίου να ανταποκρίνεται σε αιτήματα παρακολούθησης). Σε ορισμένα επίπεδα υπηρεσίας (π.χ. αποθήκευση) χρησιμοποιούνται περιορισμοί, αλλά όχι στην περίπτωση των μετρήσεων επιπέδου IaaS. Το προτεινόμενο διάστημα μέτρησης εξαρτάται από τον ελάχιστο συνεχή χρόνο σφάλματος που έχει ορίσει ο Πάροχος. Ωστόσο, 1 δείγμα ανά λεπτό θα μπορούσε να θεωρηθεί αρκετά ακριβές.
- Ο προσδιορισμός μιας μέτρησης ως επιτυχής πρέπει να βασίζεται σε συμφωνημένο αποτέλεσμα εντός συμφωνημένης προθεσμίας.
- Το συμφωνημένο χρονικό όριο μπορεί επίσης να εξαρτάται από το διάστημα μεταξύ των μετρήσεων: $Time_limit(\text{χρονικό όριο}) = \max(\text{διάστημα μεταξύ μετρήσεων})$
- Ο προσδιορισμός μιας έγκυρης διακοπής στη διαθεσιμότητα πρέπει να βασίζεται σε μια συμφωνημένη περίοδο συνεχών μετρήσεων που υποδεικνύουν μη διαθεσιμότητα. Η τιμή που χρησιμοποιήθηκε είναι 60 δευτερόλεπτα, μια τιμή που χρησιμοποιείται από το AWS¹.

¹ <https://aws.amazon.com/compute/sla/>

- Ο στόχος διαθεσιμότητας πρέπει να συμφωνηθεί. Αυτό εξαρτάται από τον συγκεκριμένο τύπο εφαρμογής και τις απαιτήσεις της και από τον τύπο πάροχο που χρησιμοποιείται για τον υπολογισμό, επομένως δεν μπορεί να υπάρξει ένα συγκεκριμένο ποσοστό. Ωστόσο έγινε χρήση τυποποιημένων σεναρίων σφάλματος που συνήθως αντιπροσωπεύουν διαφορετικές απαιτήσεις κατηγοριών εφαρμογών. Οι τύποι παρόχου συγκριτικής αξιολόγησης έναντι αυτών των σεναρίων θα μπορούσαν να αποτελούν ένδειξη ορίου, καθώς και ένα άμεσα συγκρίσιμο χαρακτηριστικό της εγγυημένης διαθεσιμότητας του Παρόχου.

- *Διαθέσιμος χρόνος = Συνολικός χρόνος που αναμένεται συμβατικά κατά την περίοδο αναφοράς - επιτρεπόμενος χρόνος διακοπής*
- *Διαθεσιμότητα = [Διαθέσιμος χρόνος - (συνολικός χρόνος διακοπής - επιτρεπόμενος χρόνος διακοπής)] / (Διαθέσιμος χρόνος)*
- Η απλούστερη μέτρηση θα ήταν
Διαθεσιμότητα = Διαθέσιμα δείγματα / Συνολικά δείγματα, υπό την προϋπόθεση ότι τα δείγματα ακολουθούν μια ελάχιστη περίοδο

Χρησιμοποιώντας τις παραπάνω τυποποιήσεις για τον τρόπο συλλογής και υπολογισμού για την μετρική της διαθεσιμότητας(availability) για ένα Cloud SLA δημιουργήθηκε το μοντέλο για μια τυπική υπηρεσία Cloud της AWS.

```
{
  "name": "Indicative Availability (Accessibility) SLA",
  "referenceId": "ASV_001",
  "scale": "NOMINAL",
  "expression": {
    "expression": "CFA_002<PARAM_002"
  },
  "parameters": [{
    "name": "availability_limit",
```

```

"referenceId": "PARAM_002",
"unit": "%",
"parameter": "99.95"
}],
"underlyingMetrics": [ {
  "name": "CloudServiceAvailability",
  "referenceId": "CFA_002",
  "unit": "%",
  "scale": "RATIO",
  "expression": {
    "expression": "CFA_002 = ((BP_001 - UAP_001) / BP_001)"
  },
  "parameters": [ {
    "name": "billing cycle",
    "referenceId": "BP_001",
    "unit": "month",
    "parameter": "1"
  } ],
"underlyingMetrics": [ {
  "name": "CloudServiceUnavailability",
  "referenceId": "UAP_001",
  "unit": "second",
  "scale": "INTERVAL",
  "expression": {
    "expression": "UAP_001 = SUM(QDT_001)"
  },
  "underlyingMetrics": [ {
    "name": "CloudServiceUnavailability_INTERVAL",
    "referenceId": "QDT_001",
    "unit": "second",
    "scale": "INTERVAL",
    "expression": {
      "expression": "IF (QDT_001_TEMP > PARAM_001) THEN QDT_001 = QDT_001_TEMP",
      "subExpressions": [ {
        "expression": "IF (SAMPLE_001 = PARAM_002) THEN QDT_001_TEMP =
delta(SAMPLE_001.timestamp)"
      } ] },
    "parameters": [ {
      "name": "boundary_period",

```

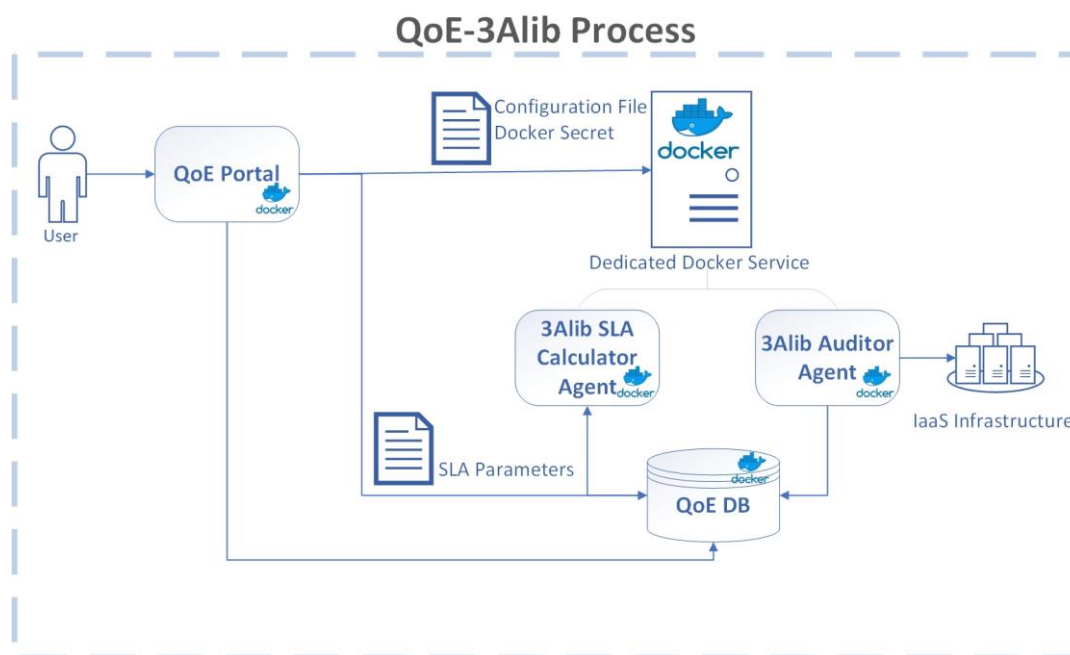
```

    "parameter": "60",
    "unit": "seconds",
    "scale": "INTERVAL",
    "referenceId": "PARAM_001"
  },
  {
    "name": "service_ping_sample_unreachable",
    "parameter": "unreachable",
    "scale": "NOMINAL",
    "referenceId": "PARAM_002"
  },
  {
    "name": "service_ping_sample_responses",
    "referenceId": "PARAM_003",
    "parameter": [
      "reachable",
      "unreachable"
    ],
    "scale": "ordinal"
  }
}],
"rules": [{
  "rule": "Services deployed in at least two availability zones",
  "note": "",
  "referenceId": "QDT_R001"
}],
"samples": [ {
  "name": "service_ping_sample",
  "referenceId": "SAMPLE_001",
  "timestamp": "the timestamp of the sample",
  "scale": "NOMINAL",
  "value": "PARAM_003",
  "protocol": "ICMP",
  "operation": "ping",
  "note": ""
} ] ] ] ] ]

```

4.1.2 Τεχνική υλοποίηση(λογισμικό QoE,3ALib)

Όπως προαναφέρθηκε στην αρχιτεκτονική ανάλυση του συστήματος (κεφάλαιο 3.2) έχει δημιουργηθεί μια πολυεπίπεδη αρχιτεκτονική, δημιουργημένη ώστε να μπορέσει να υπάρξει λειτουργικότητα στα εργαλεία, καθώς και η απαιτούμενη κατανομή και δυνατότητα κλιμάκωσής του λογισμικού.



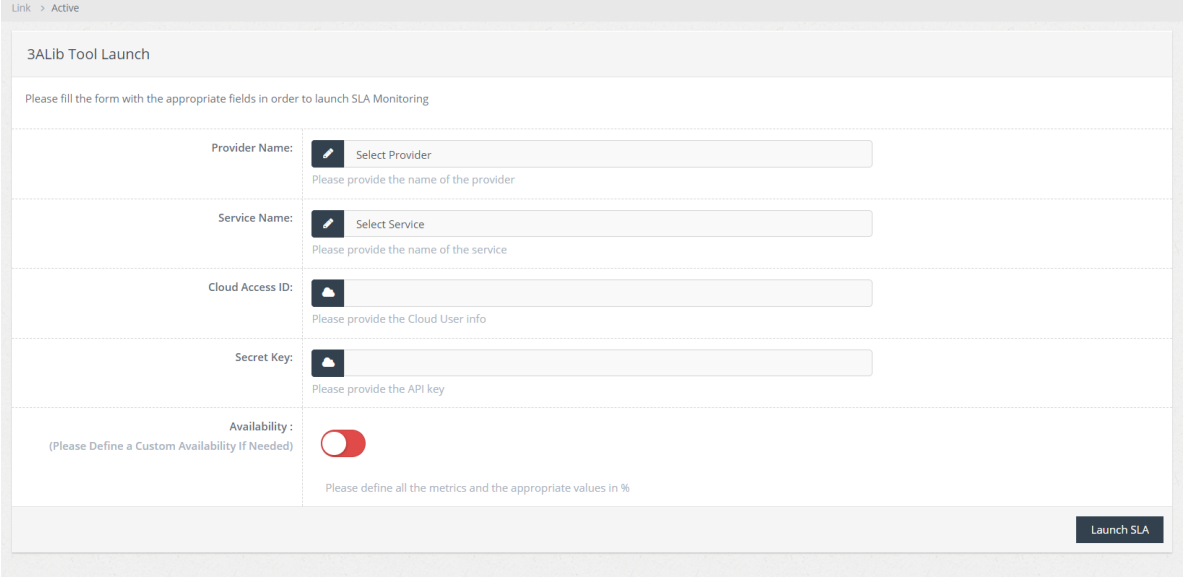
Εικόνα 6: Αφαιρετικού επιπέδου διαλειτουργικότητα QoE-3ALib.

Όλα τα κομμάτια λογισμικού που αποτελούν το συστήματα αξιολόγησης SLA, είναι κατακερματισμένα σε μικρό-υπηρεσίες (Docker Containers²). Τα Containers είναι απομονωμένα μεταξύ τους και ομαδοποιούν το δικό τους λογισμικό, βιβλιοθήκες και αρχεία διαμόρφωσης μπορούν να επικοινωνούν μεταξύ τους μέσω καλά καθορισμένων καναλιών.

² <https://www.docker.com/resources/what-container>

QoE Portal

Το QoE Portal αποτελεί την διεπαφή η οποία δύναται να χρησιμοποιηθεί από το χρήστη (SaaS) για την παραμετροποίηση ενός καινούριου Agent για την παρακολούθηση αλλά και την αξιολόγηση ενός SLA. Είναι μια διαδικτυακή διεπαφή, όπου αναπτύχθηκε σε AngularJS³ και έχει εγκατασταθεί σε ένα Docker Container, όπου λειτουργεί σαν ένας HTTP Server ώστε γρήγορα και ευκολά να μπορεί να εγκατασταθεί, καθώς και να κλιμακωθεί ανάλογος με τον φόρτο εργασίας(request load).



The screenshot shows a web form titled "3ALib Tool Launch" with a breadcrumb "Link > Active". Below the title is a instruction: "Please fill the form with the appropriate fields in order to launch SLA Monitoring". The form contains five main sections:

- Provider Name:** A dropdown menu with a cloud icon and the text "Select Provider". Below it, a placeholder text says "Please provide the name of the provider".
- Service Name:** A dropdown menu with a cloud icon and the text "Select Service". Below it, a placeholder text says "Please provide the name of the service".
- Cloud Access ID:** A text input field with a cloud icon. Below it, a placeholder text says "Please provide the Cloud User info".
- Secret Key:** A text input field with a cloud icon. Below it, a placeholder text says "Please provide the API key".
- Availability:** A toggle switch that is currently turned on (red). Below it, a placeholder text says "Please define all the metrics and the appropriate values in %".

At the bottom right of the form is a "Launch SLA" button.

Εικόνα 7: Φόρμα παραμετροποίησης εργαλείου 3ALib.

Η παραπάνω εικόνα απεικονίζει την φόρμα την οποία καλείται να συμπληρώσει ο χρήστης για να εκκινήσει μια καινούρια παρακολούθηση SLA. Όπως αναφέρθηκε στο κεφάλαιο 4.1.1, υπάρχει ένα σύνολο από παραμετροποιήσεις που χρειάζονται ώστε να καθίσταται δυνατή η αξιολόγηση και η παρακολούθηση(τρόπος δειγματοληψία) ενός SLA με βάση τον τρόπο που το ορίζει ένας Πάροχος υπηρεσιών Cloud.

³ <https://angular.io/>

Κατά την επιτυχή παραμετροποίηση δημιουργούνται τα απαραίτητα αρχεία ώστε να μπορέσουν να εκκινήσουν οι Agents που λειτουργούν σαν Docker services για την υπολογισμό και την παρακολούθηση του SLA. Το σχήμα δεδομένων που χρησιμοποιήθηκε για την παραμετροποίηση των services και την μεταφορά ευαίσθητων πληροφοριών όπως κωδικοί χρήστη για την πρόσβαση στις Cloud υπηρεσίες που μισθώνει είναι το παρακάτω.

```
{
  "EndpointSpec": {
    "Ports": []
  },
  "Labels": {},
  "Mode": {
    "Replicated": {
      "Replicas": 1
    }
  },
  "Name": "",
  "TaskTemplate": {
    "ContainerSpec": {
      "Mounts": [],
      "Image": "3alib_image:latest",
      "Args": [
        "java",
        "-jar",
        "3alibAuditing.jar",
        "QoEUserID",
        "providerName",
        "serviceName",
        "auditor",
        "0"
      ],
      "Secrets": [
        {
          "File": {
```

```

    "GID": "0",
    "Mode": 292,
    "Name": "QoEUserID_providerName_serviceName_containerType",
    "UID": "0"
  },
  "SecretID": "",
  "SecretName": "QoEUserID_providerName_serviceName_containerType"
}
],
"Env": [],
"Labels": {}
}
},
"UpdateConfig": {
  "Parallelism": 1,
  "Delay": 0,
  "FailureAction": "continue"
}
}

```

Το παραπάνω αρχείο κωδικοποιείται και μεταφέρεται σαν Docker secret. Με αυτόν τον τρόπο, η πληροφορία που περιέχει μπορεί να είναι προσβάσιμη μόνο από τα Docker Containers των Agent που αρχικοποιήθηκαν για την παρακολούθηση του SLA.

Μέσω της διεπαφής του QoE Portal οι χρήστες έχουν και πρόσβαση στο SLA Analytics. Τα SLA παρόχων δημόσιου Cloud αναλύονται και χρησιμοποιώντας μια τυποποιημένη μέθοδο (ISO 19086-2) κατατάσσονται. Το ISO 19086-2 είναι η οικογένεια προτύπων για Cloud SLA, αυτή η τυποποίηση χρησιμοποιείται για τη δημιουργία ενός αυστηρού σχήματος για την περιγραφή της αυστηρότητας SLA ενός δημόσιου παρόχου, προκειμένου να αξιολογηθεί και να συγκριθεί. Στο περιβάλλον χρήστη, οι χρήστες μπορούν επίσης να αποκτήσουν μια πιο εξατομικευμένη κατάταξη με βάση τις ανάγκες και τη φύση της εφαρμογής τους, χρησιμοποιώντας τους διακόπτες(Εικόνα 8) στη σελίδα που εξαιρούν και περιλαμβάνουν παραμέτρους που πρέπει να αξιολογηθούν.



Εικόνα 8: Λειτουργία SLA Analytics για αξιολόγηση αυστηρότητας δημόσιων SLA.

3ALib Auditor Agent

Έχοντας την απαραίτητη πληροφορία από το Docker secret γίνεται εκκίνηση ενός συγκεκριμένου Agent που έχει αναπτυχθεί ώστε να καλύπτει όλες της παραμέτρους για κάθε Πάροχο Cloud υπηρεσιών ξεχωριστά. Εάν ο χρήστης επιλέξει μια υπηρεσία της AWS τότε θα εκκινεί ο Agent υπεύθυνος για την παρακολούθηση αυτών των υπηρεσιών. Αυτό σημαίνει ότι θα προσαρμοστεί η δειγματοληψία, οι ζώνες διαθεσιμότητάς, το πρωτόκολλο επικοινωνίας, κλπ. Όλα τα αρχεία καταγραφής που δημιουργήθηκαν κατά την παρακολούθηση αποθηκεύονται στη βάση δεδομένων (QoE DB) ώστε να γίνει η επεξεργασία τους από 3ALib SLA Calculator.

3ALib SLA Calculator Agent

Για να καταστεί δυνατός ο υπολογισμός των SLA παραμέτρων(κατά κύριο λόγο η μετρική της διαθεσιμότητας) πρέπει να αρχικοποιηθεί ο κατάλληλος Agent. Με την ίδια λογική που επιλέγεται ο κατάλληλος Auditor, έτσι επιλέγεται και ο κατάλληλος SLA Calculator. Το σύνθημα μεσοδιάστημα που χρησιμοποιείται (χρησιμοποιούν οι πάροχοι) είναι ο ένας μήνας. Συνεπώς, ο Calculator εξάγει τα αρχεία καταγραφής του μήνα

υπολογισμού και με βάση την φόρμουλα του κάθε παρόχου, υπολογίζει το ποσοστό διαθεσιμότητας της υπηρεσίας για τον συγκεκριμένο μήνα. Μια τυπική φόρμουλα είναι η παρακάτω:

$$AP_F = 100 * \left(\frac{I_o - I_u}{I_o} \right)$$

Οπού AP_F είναι το τελικό ποσοστό διαθεσιμότητας πόρων, I_o είναι ο συνολικός χρόνος εξέτασης του SLA ενώ το I_u είναι ο συνολικός χρόνος που οι πόροι ήταν μη διαθέσιμοι.

Τα αποτελέσματα του υπολογισμού ενός SLA αποθηκεύονται στην βάση (QoE DB) και είναι προσβάσιμα από τον χρήστη μέσα από την διεπαφή του QoE portal(Εικόνα 9).


SLA Results for: testNTUAUser1

Type	Date	Provider	Service	Data Center	SLA	SLA Violation
monthly	10/2017	aws	ec2	eu-west-1	99.6%	Violation : 0.35%
monthly	10/2017	aws	ec2	us-east-1	98%	Violation : 1.95%
monthly	10/2017	aws	ec2	eu-west-1	100%	No Violation
monthly	10/2017	aws	ec2	us-east-1	100%	No Violation
monthly	10/2017	aws	ec2	eu-west-1	100%	No Violation
monthly	10/2017	aws	ec2	us-east-1	100%	No Violation
monthly	10/2017	aws	ec2	eu-west-1	100%	No Violation
monthly	10/2017	aws	ec2	us-east-1	100%	No Violation
monthly	12/2017	ULM	Compute	ULM-DC	100%	No Violation
monthly	12/2017	ULM	Compute	ULM-DC	100%	No Violation
monthly	12/2017	ULM	Compute	ULM-DC	100%	No Violation
monthly	1/2018	Cosmote	Compute	Cosmote-DC	100%	No Violation
monthly	2/2018	ULM	Compute	ULM-DC	100%	No Violation
monthly	2/2018	Cosmote	Compute	Cosmote-DC	99.98888062169313%	No Violation

Εικόνα 9: Αποτελέσματα μηνιαίων ελέγχων SLA διαθεσιμότητας.

Σημαντικό είναι να σημειωθεί ότι δεν προγραμματιστήκαν μόνο Agents για αξιολόγηση της διαθεσιμότητας, δημιουργήθηκαν επίσης Agents που αξιολογούσαν την οριζόντια κλιμάκωση μιας εφαρμογής στο Cloud. Η μετρική με την οποία προσεγγίστηκε αυτή η αξιολόγηση ήταν το ποσοστό χρήσης της CPU(CPU_Utilization). Πιο συγκεκριμένα αν το SLA περιέχει εχέγγυα για την κλιμάκωση, ένας τρόπος με τον οποίο μπορεί να

ποσοτικοποιηθεί είναι ο έλεγχος του χρόνου που το ποσοστό χρήσης της CPU παραμένει μεγαλύτερο από αυτό που ορίζει το SLA για να κλιμακωθεί η εφαρμογή και κατά συνέπεια να μειωθεί το υπολογιστικό φορτίο. Για τον προγραμματισμό και την έλεγχο αυτού του τύπου Agent χρησιμοποιήθηκε ένα ιδιωτικό Cloud σύστημα σε OpenStack⁴. Στο χρήστη τα αποτελέσματα εμφανίζονται ως ποσοστά του χρόνου(Εικόνα 10) όπου χρειάστηκε για την κλιμάκωση σε σχέση με τον εγγυημένο χρόνο.



Scale Auditing Results

Scale Results for: testNTUUser1

Date	Provider	Service	Scale Group	scale CPU Util	promised CPU Util	Status
Oct 1, 2017 5:21:25 PM	aws	ec2	test image	30%	50%	OK
Nov 22, 2018 3:07:59 PM	Testing	Compute	cda4d9bf-6558-4753-9aa7-cb615a1dfe08	0.077585509000000001%	0.000001%	Not Scaled
Nov 23, 2018 5:33:45 PM	COSMOTE	Compute	cda4d9bf-6558-4753-9aa7-cb615a1dfe08	5.68452391%	1.000000000000000001e-7%	Not Scaled

Εικόνα 10: Αποτελέσματα ελέγχου CPU_Utilization για SLA κλιμάκωσης εφαρμογής.

QoE DB

Όλα τα δεδομένα που παράγονται από τους διάφορους Agents του 3ALib αποθηκεύονται στις κατάλληλες συλλογές μιας κατανεμημένη βάση αντικειμένων MongoDB⁵. Όπως όλα τα άλλα κομμάτια λογισμικού, έτσι και η βάση λειτουργεί σαν Docker Container. Με αυτό τον τρόπο μπορεί να κατανεμηθεί ώστε να μπορεί να εξυπηρετήσει περισσότερους χρήστες.

IaaS Infrastructure

⁴ <https://www.openstack.org/>

⁵ <https://www.mongodb.com/>

Οι Cloud υπηρεσίες που χρησιμοποιήθηκαν για την ανάπτυξη καθώς και τη λειτουργία των εργαλείων είναι οι κάτωθι:

1. AWS ec2 Cloud Services
2. FIWARE(OpenStack)
3. Cosmote - Openstack
4. ULM - Openstack

4.2 Υλοποίηση υποσυστήματος συγκριτικής αξιολόγησης

(QoE,BenchSuite)

Η αξιολόγηση της ποιότητας υπηρεσίας είναι πολύ σημαντική για την εκτίμηση της φερεγγυότητας ενός IaaS Cloud παρόχου, ωστόσο για μία ολοκληρωμένη αξιολόγηση, είναι μείζονος σημασίας να αξιολογηθεί και η απόδοση των πόρων που δίνονται μέσω της εκάστοτε υπηρεσίας. Ο τρόπος με τον οποίο προσεγγίστηκε αυτή η αξιολόγηση στα πλαίσια της διατριβής, είναι μέσω τυποποιημένων συγκριτικών αξιολογήσεων (Benchmarks). Σε αυτό το κεφάλαιο αναλύεται το θεωρητικό υπόβαθρο για την αξιολόγηση Cloud υπηρεσιών, καθώς και η τεχνική υλοποίηση ώστε να επιτευχθεί η απαραίτητη λειτουργικότητα για να υλοποιηθεί η αξιολόγηση των Cloud υπηρεσιών.

4.2.1 Τυποποίηση μετρικών Benchmarking

Τα Benchmarks έχουν επικρατήσει ως ο πιο τυποποιημένος και ευρέως διαδεδομένος τρόπος για την αξιολόγηση των πόρων ενός υπολογιστικού συστήματος. Δοθείσης της ετερογένειας των εφαρμογών και του λογισμικού που έχει αναπτυχθεί και αναπτύσσεται, το άμεσα λογικό φαινόμενο ήταν να αναπτυχθεί μια πληθώρα από Benchmarks δημιουργημένα να αξιολογούν τους πόρους με διαφορετικό τρόπο, ώστε να

προσομοιώνουν με όσο τον δυνατόν μεγαλύτερη ακρίβεια διάφορους τύπους εφαρμογών. Όλο αυτό το σύνολο των ετερογενών Benchmark παράγει και διαφορετικές μετρικές απόδοσης των συστημάτων, άρρηκτα συνδεδεμένες με την φύση του Benchmark[73]. Αυτή η ανομοιογένεια της πληροφορίας που παράγουν τα διάφορα Benchmarks πρέπει να τυποποιηθεί ώστε να δημιουργηθεί ένας κοινός τρόπος αξιολόγησης, για να καταστεί δυνατή η ανάλυση της απόδοσης των πόρων ενός Cloud συστήματος. Ο τρόπος με τον οποίο προσεγγίστηκε η ανάλυση και τυποποίηση των Benchmarks ήταν μέσω της μετρικής απόδοσης εικονικών πυρήνων (Performance of Virtual Cores(PVC)).

Η απόδοση των εικονικών πυρήνων δείχνει την ικανότητα του εικονικοποιημένου πόρου (π.χ. VM) να χειρίζεται μια υπολογιστική εργασία. Αυτό δεν μπορεί να βασιστεί σε καμία μεμονομένη μέτρηση δεδομένου ότι πρόκειται για μια πολύπλοκη διαδικασία που εξαρτάται από πτυχές όπως η συχνότητα ρολογιού (CPU clock speed), η μνήμη RAM και τα μεγέθη λανθάνουσας μνήμης (cache) και η τεχνολογία αυτής. Έτσι, συνήθως η απόδοση των (εικονικών ή μη) πυρήνων βασίζεται στη χρήση δοκιμών αναφοράς, που σχετίζονται με ένα συγκεκριμένο KPI ενδεικτικό της ικανότητας του πόρου να εξυπηρετεί τον αντίστοιχο φόρτο εργασίας. Από όσα γνωρίζουμε, δεν υπάρχουν εγγυήσεις για αυτήν την πτυχή από εμπορικούς παρόχους υπηρεσιών Cloud. Σε ορισμένες περιπτώσεις παρέχεται μια βασική χωρητικότητα, ωστόσο βασίζεται σε συγκεκριμένες μετρήσεις παρόχου που είναι ασαφείς και δεν είναι συγκρίσιμες με εξωτερικές υπηρεσίες (π.χ. AWS Compute Units).

Συνήθως οι πάροχοι εγγυώνται την κατανομή του αριθμού των πυρήνων και της μνήμης RAM ενός δεδομένου εικονικού πόρου (π.χ. VM). Ωστόσο, λόγω της ενοποιημένης διαχείρισης φόρτου εργασίας, ενδέχεται να έχουν εκχωρηθεί περισσότεροι εικονικοί πυρήνες σε έναν φυσικό κόμβο από τους πραγματικά διαθέσιμους, οδηγώντας σε

αλληλοεπικάλυψη. Ακόμα κι αν αυτό δεν συμβεί, το ζήτημα των παρεμβολών VM ακόμη και όταν χρησιμοποιούνται ξεχωριστούς πυρήνες είναι επίσης ένας παράγοντας που επηρεάζει την απόδοση και την Ποιότητας Υπηρεσίας. Όπως και με την αξιολόγηση των SLA έτσι και για την απόδοση των εικονικών πόρων είναι σημαντικό να ληφθούν υπόψιν και οι δυο οπτικές (Παρόχου και Χρήστη) για το πώς αντιλαμβάνονται την απόδοση και την χρήση των εικονικών πόρων.

Απόδοση Εικονικών Πόρων Παρόχου Cloud Service	Απόδοση Εικονικών Πόρων Χρήστη Cloud Service
<p>Επιλογή μεγέθους VM από χρήστη. Ο χρήστης μπορεί να επιλέξει το μέγεθος των VM του, συνήθως από μια προεπιλογή τύπων ή σε ορισμένες περιπτώσεις καθορίζοντας το δικό του μέγεθος.</p> <p>Ενδεικτική απόδοση μεγέθους VM. Οι πάροχοι αναφέρουν την αναμενόμενη υπολογιστική ικανότητα των VM με κάποιο τρόπο (κυρίως στατικές, π.χ. υπολογιστικές μονάδες) και δεν εγγυώνται τη σταθερότητα της απόδοσης χρόνου εκτέλεσης. Σε ορισμένες περιπτώσεις, υποδεικνύουν επίσης την καταλληλότητα για έναν σκοπό.</p> <p>Σταθερότητα εμπειρικής απόδοσης. Σε πολλές περιπτώσεις, οι Χρήστες δεν</p>	<p>Σταθερότητα εμπειρικής απόδοσης. Σε πολλές περιπτώσεις, οι Χρήστες ενδιαφέρονται για τις απόλυτες τιμές απόδοσης, αλλά περισσότερο για τη σταθερότητα της εμπειρικής απόδοσης. Αυτό είναι ιδιαίτερα απαραίτητο για να δοθεί στους τελικούς χρήστες τους ένα σταθερό περιβάλλον για τις υπηρεσίες, καθώς και για να είναι σε θέση να υπολογίζουν με ακρίβεια την τιμολόγηση των υπηρεσιών που βρίσκονται σε εικονικούς πόρους (εάν οι Χρήστες είναι π.χ. SaaS Providers που νοικιάζουν υπηρεσίες σε επίπεδο IaaS).</p> <p>Αντιστοίχιση απόδοσης και κόστους σε</p>

<p>ενδιαφέρονται περισσότερο για τις απόλυτες τιμές απόδοσης, αλλά για τη σταθερότητα της απόδοσης με βάση την πραγματική χρήση. Αυτό είναι ιδιαίτερα απαραίτητο για να δοθεί στους τελικούς χρήστες τους ένα σταθερό περιβάλλον για τις υπηρεσίες, καθώς και να είναι σε θέση να υπολογίζουν με ακρίβεια την τιμολόγηση των υπηρεσιών που βρίσκονται σε εικονικούς πόρους (εάν οι Χρήστες είναι π.χ. SaaS Providers που ενοικιάζουν υπηρεσίες σε επίπεδο IaaS).</p> <p>Αντιστοίχιση απόδοσης και κόστους σε συγκεκριμένες προσφορές (π.χ. GPU βελτιωμένη για γραφικά, SSD για απόδοση αποθήκευσης I/O κ.λπ.). Οι Χρήστες χρειάζονται έναν γενικευμένο τρόπο με τον οποίο μπορούν να κατανοήσουν την ικανότητα ενός συγκεκριμένου εικονικού πόρου να χειρίζεται έναν συγκεκριμένο τύπο εφαρμογής και πώς θα μεταφραστεί σε επίπεδο KPI για τους τελικούς χρήστες τους.</p>	<p>μετρήσει στο επίπεδο της εφαρμογής.</p> <p>Οι χρήστες χρειάζονται έναν γενικευμένο τρόπο με τον οποίο μπορούν να κατανοήσουν την ικανότητα ενός συγκεκριμένου εικονικού πόρου να χειρίζεται έναν συγκεκριμένο τύπο εφαρμογής και πώς θα μεταφραστεί σε επίπεδο KPI για τους τελικούς χρήστες τους.</p>
---	--

Πίνακας 5: Ανάλυση απόδοσης VM με βάση αναγκών χρήστη/παρόχου Cloud Service.

Τυποποίηση Μετρικής Απόδοσης εικονικών Πύρινων (PVC)

Η καθορισμένη διαδικασία συγκριτικής αξιολόγησης επαναλαμβάνεται περιοδικά. Λαμβάνοντας υπόψη το δυναμικό περιβάλλον του Cloud, οποιαδήποτε διαδικασία συγκριτικής αξιολόγησης θα πρέπει να επαναλαμβάνεται περιοδικά και με τρόπο που να καλύπτει διαφορετικές ζώνες ώρας ή χρήσεις υπηρεσιών Cloud (π.χ. ώρες λειτουργίας, ώρες ψυχαγωγίας κ.λπ.). Η εκτέλεση των Benchmark πρέπει να είναι αγνωστικιστική για τον Πάροχο, εάν εκτελείται από τον Χρήστη. Πρέπει να υπάρχουν όρια απόκλισης των τιμών αναφοράς, για τα οποία η ανοχή στην απόκλιση είναι αποδεκτή.

Οι μετρικές οι οποίες εξήχθησαν για την μελέτη της απόδοσης των εικονικών πυρήνων είναι οι εξής:

- Μέση ποσοστιαία απόκλιση των αποτελεσμάτων από τη μέση τιμή για το ίδιο σημείο αναφοράς(Benchmark), τον ίδιο φόρτο εργασίας και το ίδιο μέγεθος VM θα πρέπει να είναι μικρότερο από ένα όριο σε όλες τις μετρήσεις, τουλάχιστον για τη χειρότερη περίπτωση.

$$PVC_{AVG} = 100 * \left(\frac{|M_i - AM_{AVG}|}{AM_{AVG}} \right)$$

Οπού M_i είναι μια μέτρηση ενώ AM_{AVG} είναι η μέση τιμή όλων των μετρήσεων.

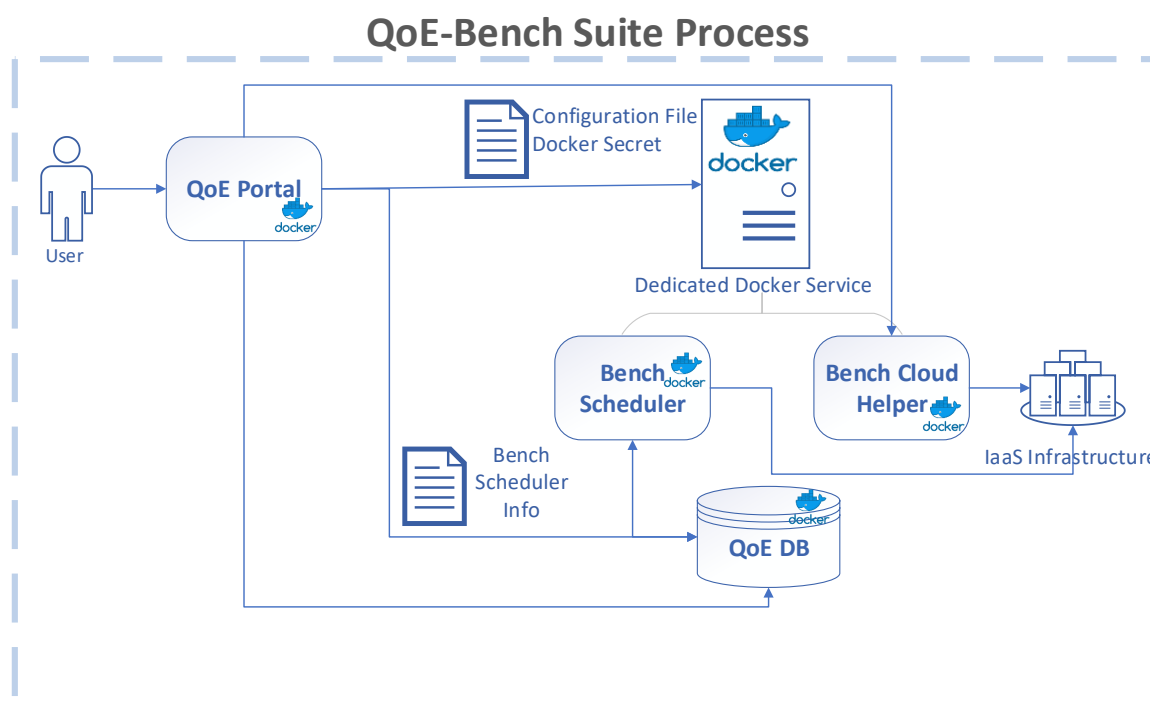
- Η απόκλιση της ελάχιστης ή μέγιστης τιμής από τη μέση τιμή για το ίδιο σημείο αναφοράς(Benchmark), τον ίδιο φόρτο εργασίας και το ίδιο μέγεθος VM δεν πρέπει να είναι μεγαλύτερη από ένα όριο.

$$PVC_{MAX} = 100 * \left(\frac{|M_{max} - AM_{AVG}|}{AM_{AVG}} \right)$$

Οπού M_{max} είναι η καλύτερη μέτρηση που έχει υπάρξει στο σύνολο δεδομένων ενώ AM_{AVG} είναι η μέση τιμή όλων των μετρήσεων.

4.2.2 Τεχνική υλοποίηση(λογισμικό QoE, Bench Suite)

Η ίδια λογική που διέπει την αρχιτεκτονική καθώς και την αλληλεπίδραση μεταξύ των στοιχείων λογισμικού QoE και 3ALib, διέπει και το QoE με το Bench Suite. Συγκεκριμένα, το QoE είναι υπεύθυνο για τη λειτουργία καθώς και για τη συλλογή, ανάλυση και παρουσίαση των δεδομένων που παράγει το Bench Suite. Από την άλλη πλευρά, το Bench Suite είναι υπεύθυνο για την επικοινωνία με τους διάφορους Cloud Παρόχους, καθώς και για την εκτέλεση των Benchmarks.

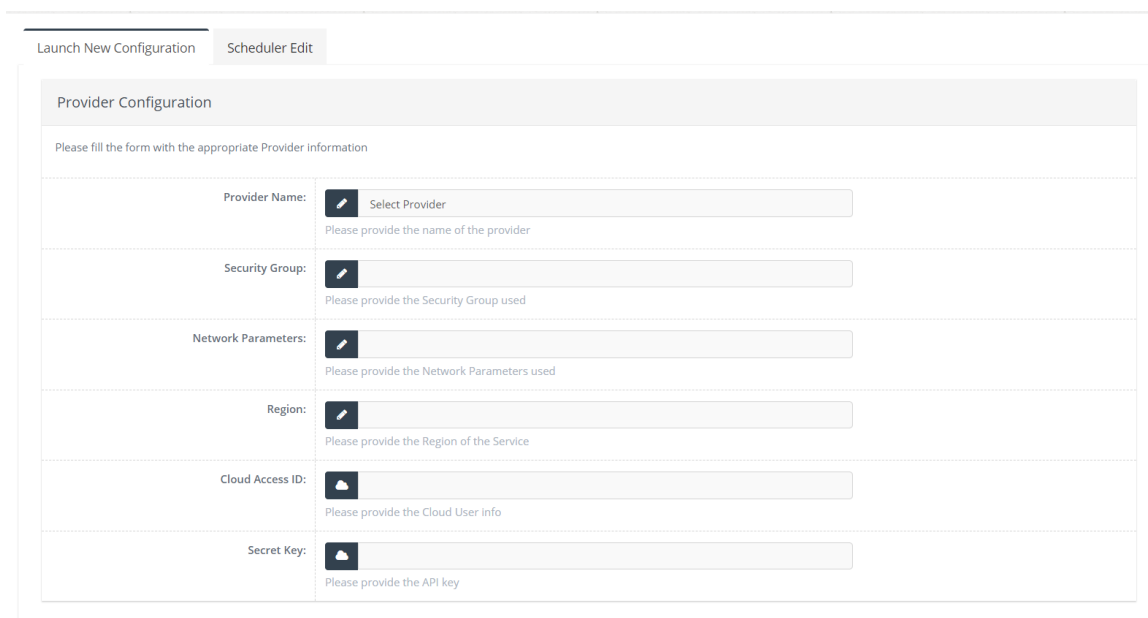


Εικόνα 11: Αφαιρετικού επιπέδου διαλειτουργικότητα QoE-Bench Suite.

QoE Portal

Προκειμένου να ξεκινήσουν ένα Benchmark, οι χρήστες πρέπει να συμπληρώσουν τη φόρμα με όλες τις κατάλληλες πληροφορίες. Αυτή η διεπαφή χρήστη έχει σχεδιαστεί ειδικά για να μπορεί να εκκινεί Benchmarks για διάφορα συστήματα. Αφού ο χρήστης αναγνωρίσει τον πάροχο Cloud, εμφανίζονται στη φόρμα τα πεδία που αντιστοιχούν στον

συγκεκριμένο πάροχο. Όταν συμπληρωθούν όλες οι πληροφορίες σχετικά με το σύστημα, οι χρήστες πρέπει να καθορίσουν το διάστημα στο οποίο θα εκτελεστούν τα Benchmarks (κάθε μέρα, μήνας, συγκεκριμένος αριθμός φορών). Επίσης, σε αυτήν τη σελίδα, υπάρχει μια ειδική καρτέλα(Εικόνα 12) για τον χρονικό προγραμματισμό των ήδη διαμορφωμένων Benchmark, αυτή η λειτουργικότητα επιτρέπει στο χρήστη να ξεκινήσει και να διαμορφώσει ήδη καθιερωμένα Benchmarks χωρίς να χρειάζεται να διαμορφώσει ξανά όλες τις παραμέτρους.



Εικόνα 12: Διεπαφή παραμετροποίησης και εκτέλεσης Benchmark.

Το σχήμα δεδομένων όπου δημιουργήθηκε ώστε να καταστήσει δυνατή την επικοινωνία με τους διάφορους Cloud Παρόχους, καθώς και το ποιο Benchmark θα εκτελεστεί, παρουσιάζεται σε Json μορφή παρακάτω.

```
{  
  "provider": {  
    "class": "benchsuite.stdlib.provider.libcloud.LibcloudComputeProvider",  
    "name": "",  
    "driver": "",
```

```

    "access_id": "",
    "secret_key": "",
    "region": "",
    "security_group": "",
    "network": "",
    "auth_url": "",
    "auth_version": "",
    "tenant": "",
    "userDomain": "",
    "projectDomain": ""
  },
  "vm": {
    "name": "",
    "image": "",
    "size": "",
    "vm_user": "",
    "platform": "",
    "key_name": "",
    "ssh_private_key": "",
    "post_create_script": ""
  }
}

```

Οι πληροφορίες που περιέχει το σχήμα αυτό είναι απαραίτητες όχι μόνο για την επικοινωνία με τον Cloud Πάροχο αλλά και για την πρόσβαση στο κατάλληλο VM στο οποίο θα εκτελεστεί το Benchmark.

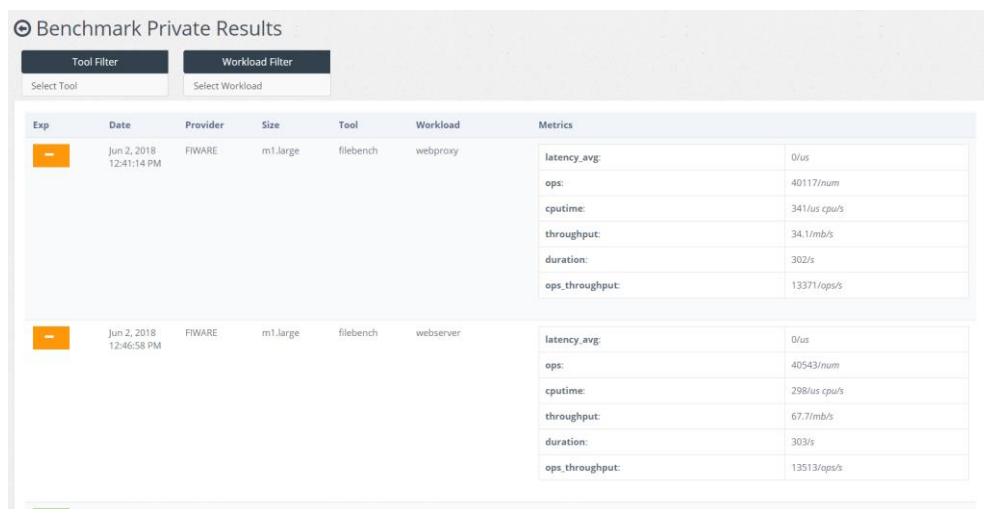
Μετά την επιτυχή εκτέλεση και παραμετροποίηση μέσω της διεπαφής QoE portal, τα αποτελέσματα των εκτελεσμένων Benchmark αποθηκεύονται στην βάση δεδομένων QoE DB για να μπορέσουν να επεξεργαστούν και να παρουσιαστούν στον χρήστη. Οι χρήστες μπορούν να δουν όλα τα Benchmarks που έχουν εκτελεστεί στους διάφορους Παρόχους Cloud, όλα αυτά τα Benchmarks συγκεντρώνονται και κατηγοριοποιούνται ανά Πάροχο, την υπηρεσία και το συγκριτικό Benchmark. Για να πραγματοποιηθεί μια πλήρης αξιολόγηση, όχι μόνο οι μετρήσεις από τα Benchmarks αλλά και οι τυποποιημένες

μετρήσεις των PVC (AVG για την σταθερότητα της μέσης απόδοσης και MAX για την σταθερότητα της μέγιστης απόδοσης), εμφανίζονται στον πίνακα. Πατώντας τα πράσινα κουμπιά με την ένδειξη «+» (Εικόνα 13) οι χρήστες μπορούν να δουν τα πλήρη αποτελέσματα σε οποιοδήποτε Benchmark και επίσης μπορούν να χρησιμοποιήσουν την γραμμή αναζήτησης για να βρουν συγκεκριμένα Benchmark.

Exp	Provider/Size	Tool	Metrics AVG		Metrics DEV		PVC AVG		PVC MAX	
-	cosmote-testbed/m1.medium	dacapo/eclipse	duration:s	1015.	duration:s	17.33	duration:s	1.41%	duration:s	2.58%
+	cosmote-testbed/m1.medium	idle/idle30	expand to see AVG		expand to see DEV		expand to see PVC AVG		expand to see PVC MAX	
+	cosmote-testbed/m1.medium	dacapo/avrora	expand to see AVG		expand to see DEV		expand to see PVC AVG		expand to see PVC MAX	
+	cosmote-testbed/m1.medium	dacapo/fop	expand to see AVG		expand to see DEV		expand to see PVC AVG		expand to see PVC MAX	
+	cosmote-testbed/m1.medium	dacapo/h2	expand to see AVG		expand to see DEV		expand to see PVC AVG		expand to see PVC MAX	
+	cosmote-testbed/m1.medium	dacapo/luindex	expand to see AVG		expand to see DEV		expand to see PVC AVG		expand to see PVC MAX	
+	cosmote-testbed/m1.medium	dacapo/jython	expand to see AVG		expand to see DEV		expand to see PVC AVG		expand to see PVC MAX	
+	cosmote-testbed/m1.medium	dacapo/lusearch	expand to see AVG		expand to see DEV		expand to see PVC AVG		expand to see PVC MAX	
+	cosmote-testbed/m1.medium	dacapo/pmd	expand to see AVG		expand to see DEV		expand to see PVC AVG		expand to see PVC MAX	

Εικόνα 13: Απεικόνιση αποτελεσμάτων εκτελεσμένων Benchmark μέσω του QoE.

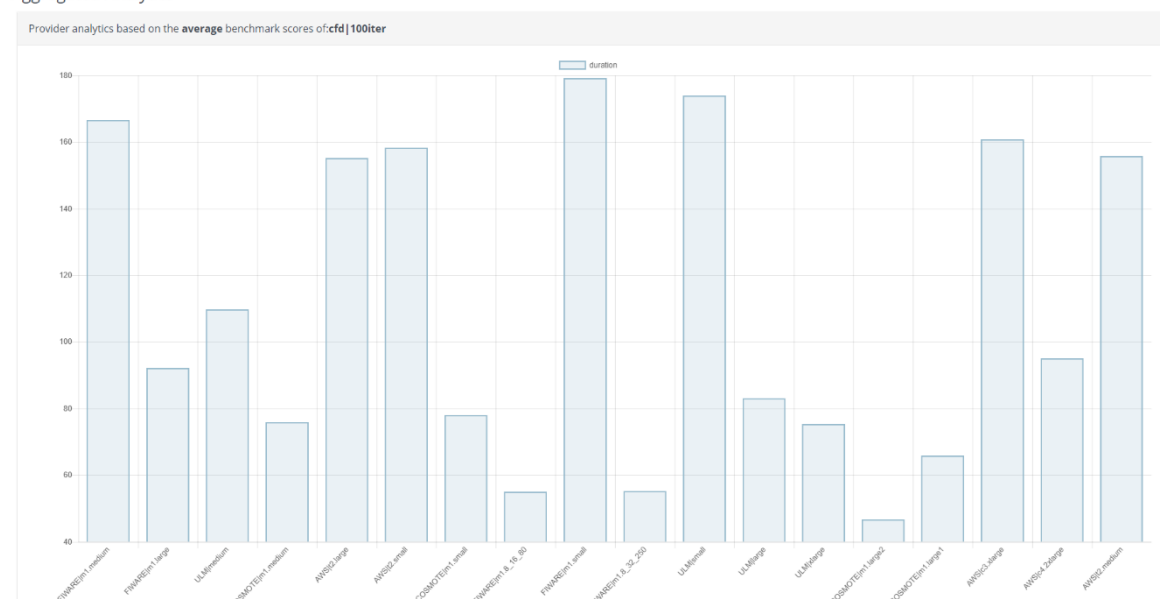
Οι χρήστες μπορούν επίσης να δουν τα προσωπικά τους αποτελέσματα κάνοντας κλικ στο κουμπί ιδιωτικών αποτελεσμάτων στη γραμμή πλοήγησης. Τα ιδιωτικά αποτελέσματα δεν είναι συγκεντρωτικά· κάθε σειρά σε αυτούς τους πίνακες παρουσιάζει μια συγκεκριμένη εκτέλεση Benchmark (Εικόνα 14). Αυτή η λειτουργικότητα αφορά κυρίως τους ιδιωτικούς Παρόχους Cloud , που δύναται να χρησιμοποιήσουν το εργαλείο για αυτό-αξιολόγηση.



Εικόνα 14: Απεικόνιση ιδιωτικών εκτελεσμένων Benchmark μέσω του QoE.

Η αναλυτική παράθεση των αποτελεσμάτων αν και σημαντική στη λεπτομερή ανάλυση της απόδοσης των διαφόρων Cloud υπηρεσιών, δεν μπορεί να παρουσιάσει μια γενικευμένη εικόνα και την συγκριτική αξιολόγηση αυτών των υπηρεσιών. Για τον λόγο αυτό, στα πλαίσια του QoE Portal δημιουργήθηκε η λειτουργικότητα των συγκεντρωτικών Analytics (Εικόνα 15).

Aggregated Analytics

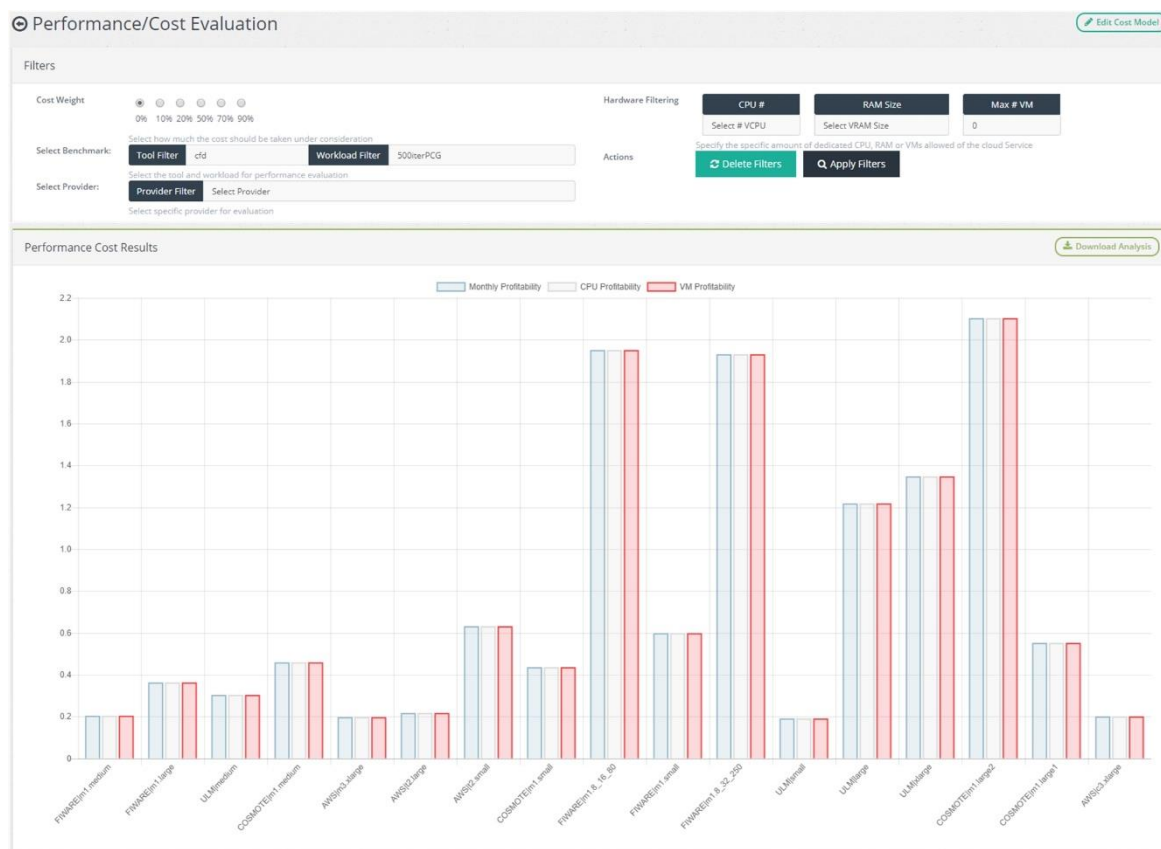


Εικόνα 15: Συγκεντρωτικά Analytics σε μορφή ραβδογράμματος.

Οι χρήστες είναι σε θέση να επιλέξουν συγκεκριμένα εργαλεία και φόρτους εργασίας και να λάβουν στατιστικά στοιχεία απόδοσης για το συγκεκριμένο σημείο αναφοράς για όλους τους παρόχους Cloud και τις υπηρεσίες τους. υπάρχουν τέσσερα γραφήματα που παράγονται στη διεπαφή χρήστη Bench analytics.

- Μέσες μετρήσεις
- Απόκλιση των μετρήσεων
- Μέσες μετρήσεις PVC (Απόδοση εικονικών πυρήνων)
- Μέγιστη απόδοση PVC

Με την λειτουργικότητα των συγκεντρωτικών Analytics, οι χρήστες μπορούν να λάβουν επίσης αναλυτικά στοιχεία απόδοσης έναντι αξιολόγησης κόστους. Χρησιμοποιώντας τα φίλτρα, οι χρήστες καθορίζουν το δείκτη αναφοράς και τον φόρτο εργασίας που τους ενδιαφέρει, καθώς και αν θέλουν αποτελέσματα μόνο για έναν συγκεκριμένο πάροχο. Ανάλογα με τις ανάγκες του χρήστη Cloud μπορεί να πραγματοποιηθεί διαφορετική εκτίμηση κόστους. Το QoE επιτρέπει στον χρήστη να καθορίσει σε τι ποσοστό θα πρέπει να ληφθεί υπόψη το κόστος (δηλαδή εάν οι χρήστες απαιτούν ισορροπημένη αξιολόγηση, το κουμπί επιλογής του ποσοστού κόστους πρέπει να μετακινηθεί στην τιμή 50%(Εικόνα 16)).



Εικόνα 16: Συγκενρωτικά Analytics απόδοσης/κόστους σε μορφή ραβδογράμματος.

Επίσης, οι χρήστες μπορούν να καθορίσουν φίλτρα υλικού (αριθμός αποκλειστικών επεξεργαστών, μέγεθος μνήμης RAM, μέγιστος αριθμός εικονικών μηχανών) προκειμένου να επιτύχουν αποτελέσματα και λύσεις υλικού που απαιτούν. Οι στατιστικές που παράγονται από το QoE περιέχουν αποτελέσματα που συγκεντρώνουν το κόστος ανά μήνα, το κόστος ανά ώρα CPU και το κόστος ανά VM βάσει του μοντέλου κόστους κάθε παρόχου. Όπως απεικονίζεται στην παρακάτω εικόνα (Εικόνα 17) στην άκρη δεξιά υπάρχει ένα κουμπί "Επεξεργασία μοντέλου κόστους". Πατώντας αυτό το κουμπί, οι χρήστες μπορούν να τροποποιήσουν το μοντέλο κόστους για κάθε πάροχο και κάθε υπηρεσία του παρόχου για αυτόν τον συγκεκριμένο λογαριασμό QoE.

Provider	Services	Performance Cost Evaluation						Action
AWS	Service Name: t2.micro	Cost Per CPU: 30	Cost Per VM: 15	Cost Per Month: 300	Max num of VM: 50	Number of CPUs: 2	RAM Size: 4	[-]
	Service Name: t2.small	Cost Per CPU: 50	Cost Per VM: 20	Cost Per Month: 400	Max num of VM: 30	Number of CPUs: 4	RAM Size: 8	[-]
	Service Name: t2.large	Cost Per CPU: 60	Cost Per VM: 30	Cost Per Month: 500	Max num of VM: 50	Number of CPUs: 8	RAM Size: 16	[-]
	Service Name: m3.xlarge	Cost Per CPU: 50	Cost Per VM: 40	Cost Per Month: 800	Max num of VM: 60	Number of CPUs: 16	RAM Size: 64	[-]
	Service Name: c3.xlarge	Cost Per CPU: 60	Cost Per VM: 50	Cost Per Month: 1000	Max num of VM: 50	Number of CPUs: 8	RAM Size: 16	[-]
	Service Name:	Cost Per CPU:	Cost Per VM:	Cost Per Month:	Max num of VM:	Number of CPUs:	RAM Size:	[+]
COSMOTE	Service Name: m1.small	Cost Per CPU: 30	Cost Per VM: 15	Cost Per Month: 300	Max num of VM: 50	Number of CPUs: 2	RAM Size: 4	[-]
	Service Name: m1.medium	Cost Per CPU: 50	Cost Per VM: 30	Cost Per Month: 400	Max num of VM: 30	Number of CPUs: 4	RAM Size: 6	[-]

Εικόνα 17: Επεξεργασία μοντέλου κόστους υπηρεσιών.

Bench Scheduler/Bench Cloud Helper

Για την δημιουργία των συγκεντρωτικών analytics καθώς και για την γενικότερη δημιουργία των μετρικών(π.χ. PVC metrics) αξιολόγησής των διάφορων υπηρεσιών, χρειάζονται μια πληθώρα από επιλαμβανόμενες εκτελέσεις των Benchmarks. Για την ευκολία στην αρχική παραμετροποίηση καθώς και για την επαναληψιμότητα της εκτέλεσης των Benchmarks δημιουργήθηκαν 2 κομμάτια λογισμικού.

- **Bench Scheduler:** είναι υπεύθυνος για τον καθορισμό της συχνότητας καθώς και του χρονικού διαστήματος που πρέπει να επαναλαμβάνεται η εκτέλεση του Benchmark.
- **Bench Cloud Helper:** είναι υπεύθυνος για την παραμετροποίηση και εκτέλεση των Benchmark.

Η διαδικασία για την παραμετροποίηση καθώς και την εκτέλεση των Benchmark αποτελείται από τρία βήματα, όπως απεικονίζεται στην Εικόνα 18.

Βήμα 1: Στο πρώτο βήμα, οι χρήστες τοποθετούν τις πληροφορίες του παρόχου, συγκεκριμένα πληροφορίες λογαριασμού και διαπιστευτήρια, προκειμένου να έχουν πρόσβαση στους διαθέσιμους πόρους του λογαριασμού τους. Υπάρχουν επίσης ορισμένα προαιρετικά πεδία σε περίπτωση που οι χρήστες θέλουν να καθορίσουν περισσότερες πληροφορίες για την εκκίνηση του Benchmark.

Βήμα 2: Μετά την αρχική εισαγωγή από τους χρήστες (βήμα 1), το Cloud Helper είναι σε θέση να ανακτήσει όλες τις σημαντικές πληροφορίες για τη διαμόρφωση του Benchmark. Στο βήμα 2, οι χρήστες επιλέγουν ένα συγκεκριμένο VM, μέγεθος και κατάλληλο δίκτυο VM στο οποίο θα εκτελεστεί το Benchmark. Με αυτά τα τρία πεδία, οι χρήστες είναι σε θέση να ξεκινήσουν ένα νέο Benchmark. Ωστόσο μπορούν να προστεθούν επίσης προαιρετικές παράμετροι, όπως ομάδα ασφαλείας, λειτουργικό σύστημα VM, κλπ.

Βήμα 3: Το τελευταίο βήμα για να ξεκινήσει ένα νέο Benchmark είναι να καθοριστεί το συγκεκριμένο εργαλείο και ο φόρτος εργασίας που θέλουν να εκτελέσουν. Οι χρήστες μπορούν να ξεκινήσουν πολλούς τύπους εργαλείων και φόρτους εργασίας στη διαμόρφωσή τους. Εάν οι χρήστες επιθυμούν να εκτελέσουν όλους τους φόρτους εργασίας ενός συγκεκριμένου Benchmark, μπορούν να αφήσουν κενό το τμήμα φόρτου εργασίας της διαμόρφωσης. Τέλος, οι χρήστες διαμορφώνουν τον Scheduler, ο οποίος είναι υπεύθυνος για την περιοδική εκτέλεση των δεικτών αναφοράς. Οι χρήστες καθορίζουν το διάστημα των Benchmark και επίσης εάν επιθυμούν να κάνουν τα Benchmarks δημοσία διαθέσιμα ή μόνο για ιδιωτική χρήση.

Launch New Configuration
Launch Wizard
Scheduler Edit

Step 1

Provider Configuration

Please fill the form with the appropriate Provider information

Provider Name: Please provide the name of the provider

Cloud Access ID: Please provide the Cloud User info

Secret Key: Please provide the API key

Optional Parameters
Next

Launch New Configuration
Launch Wizard
Scheduler Edit

Step 2

VM Configuration

Please fill the form with the appropriate Virtual Machine (VM) information

VM Image Name: Please select the Name of the image

VM Size: Please provide the size of the VM(hardware profile)

Network: Please provide the network to be used

Optional Parameters
Previous
Next

Launch New Configuration
Launch Wizard
Scheduler Edit

Step 3

Benchmark Definition

Please Provide all the Benchmarks you wish to run on the specific VM

Tool	Workload	Add
<input type="text" value="Select Tool"/>	<input type="text" value="Select Workload"/>	+

Schedule Configuration

Please fill the form with the appropriate Scheduling parameters

Run Period: Hours Days Weeks Please define the time period on which the tests will run

Scheduler Status: keep this Scheduler Active

Results Visibility: Who should be allowed to view these results

Previous
Bench Launch

Εικόνα 18: Διαδικασία παραμετροποίησης και εκτέλεσης Benchmark.

94

4.3 Εργαλείο δημιουργίας προφίλ αναγκών και κατηγοριοποίησης λογισμικού.

Η αξιολόγηση των Cloud υπηρεσιών είτε με βάση την ποιότητα της υπηρεσίας (SLA) είτε με βάση την απόδοση των πόρων (Benchmarking) δίνει μια πλήρη εικόνα για τις Cloud υπηρεσίες. Ωστόσο, οι μελλοντικοί καθώς και οι παρόντες χρήστες των Cloud υπηρεσιών πρέπει να κατανοήσουν τις ανάγκες του λογισμικού το οποίο εγκαθιστούν και λειτουργούν στις Cloud υποδομές. Η κατανόηση των αναγκών σε hardware των διαφόρων κομματιών λογισμικού, θα οδηγήσει στη σωστή επιλογή των Cloud υπηρεσιών για την εγκατάστασή τους. Στο κεφάλαιο αυτό αναλύεται το θεωρητικό υπόβαθρο ώστε να δημιουργηθεί το εργαλείο αυτό, καθώς και η αρχιτεκτονική και οι προγραμματιστικές λογικές, ώστε να επιτευχθεί το απαραίτητο αποτέλεσμα ακρίβεια καθώς και λειτουργικότητα.

4.3.1 Ανάλυση και θεωρητική προσέγγιση

Το εργαλείο Profiler είναι υπεύθυνο για την δημιουργία προφίλ με την μέθοδο μαύρου κουτιού, μιας νέας εφαρμογής που θα αναπτυχθεί σε περιβάλλον Cloud, προκειμένου να την κατατάξει σε μια συγκεκριμένη κατηγορία Benchmark, δηλαδή να βρει το Benchmark στο οποίο μοιάζει περισσότερο η εφαρμογή, όσον αφορά τις απαιτήσεις πόρων. Η ταξινόμηση ενός στοιχείου εφαρμογής που εκτελείται σε ένα VM απαιτεί πρώτα την ανάκτηση ενός αποτυπώματος συμπεριφοράς τυπικών γενικών κατηγοριών, κάθε μια από τις οποίες αντιπροσωπεύεται από ένα σχετικό Benchmark, το οποίο μετράτε από το εργαλείο Profiling. Ένα αποτύπωμα συμπεριφοράς συλλέγεται χρησιμοποιώντας εργαλεία παρακολούθησης νέφους (όπως το πρόσθετο Ceilometer του Openstack [74]) και

αποτελείται από ένα σύνολο μετρήσεων πόρων, όπως χρόνος CPU, MB μνήμης που χρησιμοποιείται ή αριθμός πακέτων εισόδου σε συγκεκριμένη περίοδο χρόνος.

Τα αποτυπώματα των προαναφερθέντων κατηγοριών Benchmark χρησιμοποιούνται ως είσοδος, προκειμένου να εκπαιδευτεί το εργαλείο ταξινόμησης. Επιπλέον, το αποτύπωμα της συνιστώσας εφαρμογής στο ίδιο περιβάλλον μετράτε επίσης από το εργαλείο Profiler και προωθείται στο εργαλείο Classification, για να προσδιοριστούν τα καλύτερα στερεότυπα απόδοσης, χρησιμοποιώντας την Ευκλείδεια απόσταση των αντίστοιχων διανυσμάτων αποτυπώματος.

Το εργαλείο προφίλ και ταξινόμησης έχει δύο λειτουργίες, μία για τον κάτοχο της εφαρμογής (Cloud Adaptor) και μία για τον πάροχο Cloud. Στην πρώτη λειτουργία, χρησιμεύει ως προαιρετική διαδικασία που βοηθά τον Cloud Adopter να επιλέξει την καλύτερη προσφορά υπηρεσιών όσον αφορά την αναμενόμενη απόδοση για την εφαρμογή του. Στη δεύτερη λειτουργία, χρησιμοποιείται για την κατηγοριοποίηση των VM που εκτελούνται σε μια υποδομή Cloud. Μπορεί να χρησιμοποιηθεί από έναν πάροχο Cloud για α) να γνωρίζει ποιους φόρτους εργασίας εκτελούνται στην υποδομή και β) πώς να διαχειρίζεται αποτελεσματικά αυτούς με την κατανομή πόρων σύμφωνα με τις αναμενόμενες απαιτήσεις και διακυμάνσεις.

Ανάλυση δημιουργίας αποτυπώματος εφαρμογής

Το αποτύπωμα της εφαρμογής όπως προαναφέρθηκε αποτελεί ένα πολυδιάστατο διάνυσμα, όπου κάθε όρισμα του διανύσματος αντιπροσωπεύει κάποια μετρική που αφορά μια συγκεκριμένη κατανάλωση σε πόρους (CPU, RAM, Cash hits, κλπ.). Οι μετρικές που εξάγονται από το σύστημα καλύπτουν την χρήση όλων των κύριων κομματιών υλικού που χρησιμοποιούνται για την λειτουργία κάποιου λογισμικού. Στον

παρακάτω πίνακα (Πίνακας 6) παρουσιάζονται όλες οι μετρικές που χρησιμοποιήθηκαν καθώς και οι κατηγορίες τους.

Κατηγορία	Μετρικές
CPU	<ul style="list-style-type: none"> • compute.node.cpu.user.percent • compute.node.cpu.kernel.percent • cpu_time • cpu_util • perf.cpu.cycles
RAM	<ul style="list-style-type: none"> • memory.usage • memory.resident • hardware.memory.used • hardware.memory.total
Cache Memory	<ul style="list-style-type: none"> • perf.cache.references • perf.cache.misses
Hard Disk IO	<ul style="list-style-type: none"> • disk.read.bytes.rate • disk.write.bytes.rate • disk.read.requests.rate • disk.write.requests.rate
Network	<ul style="list-style-type: none"> • network.outgoing.packets • network.outgoing.bytes.rate • network.outgoing.bytes / network.outgoing.packets • network.outgoing.packets.rate • network.incoming.packets • network.incoming.bytes.rate • network.incoming.bytes / network.incoming.packets • network.incoming.packets.rate.

Πίνακας 6: Σύνολο μετρικών για την δημιουργία αποτοπώματος εφαρμογής.

Συνολικά υπάρχουν 5 κατηγορίες μετρικών οι οποίες εξετάζονται, καθώς επίσης το σύνολο των μετρικών είναι 24. Οι κατηγορίες καλύπτουν όλα τα κύρια κομμάτια υλικού που χρησιμοποιεί μια εφαρμογή, όπως επεξεργαστή, μνήμη RAM, μνήμη Cache, καθώς και δίκτυο. Όλες αυτές οι μετρικές συνθέτουν το διάγραμμα που αντιπροσωπεύει το αποτύπωμα της εφαρμογής.

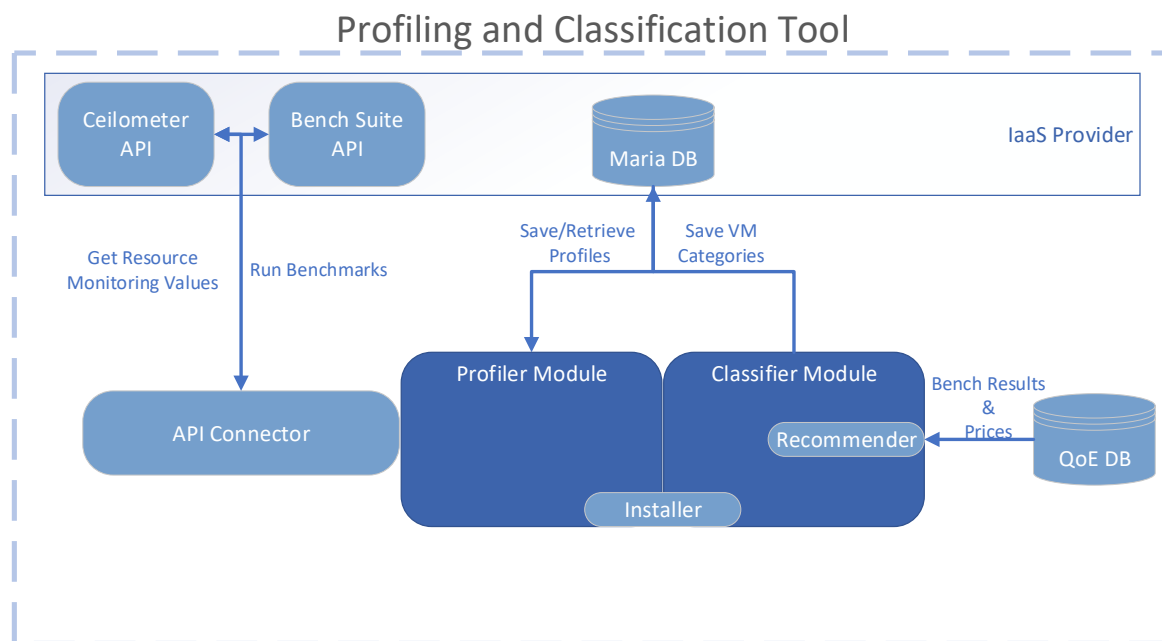
Συσχέτιση διανυσμάτων αποτυπωμάτων

Η εξαγωγή του αποτυπώματος είναι το πρώτο στάδιο για την δημιουργία του προφίλ μιας εφαρμογής ή ενός κομματιού λογισμικού. Το δεύτερο και εξίσου σημαντικό στάδιο είναι η συσχέτιση και η συμπερασματολογία πάνω σε αυτό το αποτύπωμα. Ο τρόπος με τον οποίο επιτυγχάνεται αυτό στα πλαίσια του εργαλείου που αναπτύχθηκε είναι μέσω της συσχέτισης των αποτυπωμάτων μιας εφαρμογής και αποτυπωμάτων που έχουν εξαχθεί από την εκτέλεση Benchmarks. Ο λόγος για τον οποίο δημιουργείται έτσι το προφίλ είναι διότι τα Benchmarks αποτελούν τον πιο ευρέως διαδεδομένο και εύκολο τρόπο να ελεγχθεί η απόδοση ενός συστήματος. Ο τελικός στόχος στη δημιουργία ενός προφίλ είναι η κατανόηση των υπολογιστικών αναγκών μιας εφαρμογής. Με τη συσχέτιση μεταξύ αποτυπωμάτων εφαρμογών και Benchmark έχουμε μια ενδεικτική γνώση στο πως μπορεί να αποδώσει μια εφαρμογή στα διαφορετικά Cloud συστήματα χωρίς να χρειάζεται να εγκατασταθεί και να λειτουργήσει σε αυτά, σώζοντας έτσι και χρόνο αλλά και πολύτιμους οικονομικούς πόρους.

Η συσχέτιση μεταξύ των αποτυπωμάτων γίνεται με την χρήση του αλγορίθμου k-NN (αλγόριθμος κοντινότερων γειτόνων). Πιο συγκεκριμένα βρίσκεται η ευκλείδεια ή αλλού είδους απόσταση διανυσμάτων (Manhattan, Hamming) και επιλέγεται η μικρότερη, που επί της ουσίας σημαίνει ότι η χρήση των πόρων του συστήματος για τα δυο αποτυπώματα (εφαρμογής και Benchmark) είναι παρόμοια.

4.3.2 Ανάλυση αρχιτεκτονικής συστήματος δημιουργίας προφίλ λογισμικού

Το εργαλείο δημιουργίας προφίλ αποτελείται από δυο κύρια υπό-εργαλεία, το εργαλείο για την εξαγωγή του αποτυπώματος χρήστη (profiler module) και το εργαλείο κατηγοριοποίησης λογισμικού (classification module).



Εικόνα 19: Αρχιτεκτονική αφαιρετικού επιπέδου εργαλείου δημιουργίας προφίλ.

Όπως απεικονίζεται και στην Εικόνα 19 πέραν από τα βασικά στοιχεία του λογισμικού για την λειτουργία του εργαλείου χρειάζονται οι κατάλληλες διεπαφές καθώς και βάσεις δεδομένων.

Διεπαφή προγραμματισμού εφαρμογών API Connector

Η διεπαφή που έχει δημιουργηθεί είναι απαραίτητη για την επικοινωνία και την συλλογή των αναγκαίων δεδομένων από τις Cloud υποδομές όπου τρέχουν τα Benchmarks και συλλέγονται τα αποτυπώματα χρήσης πόρων. Η διεπαφή αυτή ενορχηστρώνει δύο επικοινωνίες:

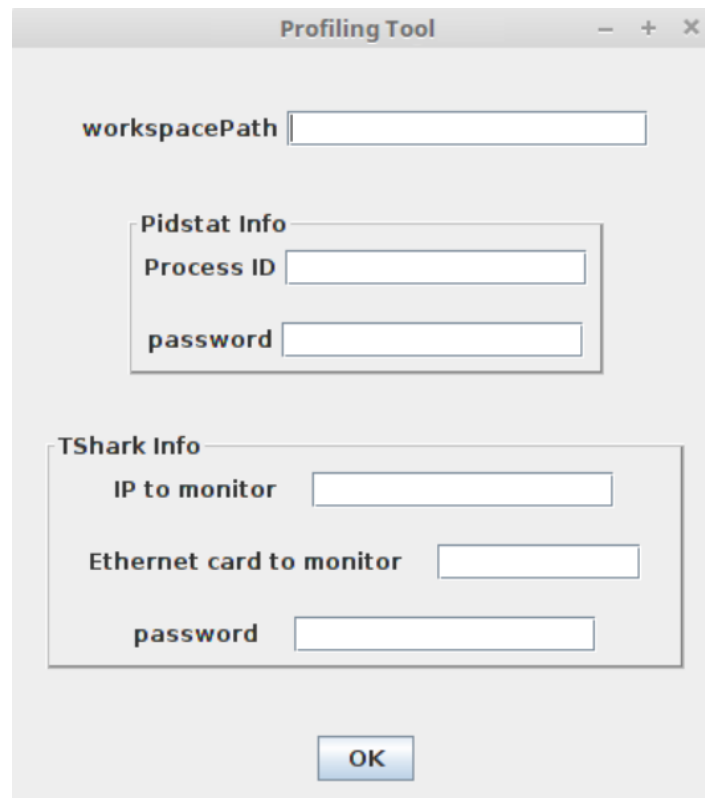
- **Ceilometer API:** Αυτή η διεπαφή είναι απαραίτητη ώστε να μπορέσουν να συλλεχθούν τα απαραίτητα στατιστικά χρήσης που συνθέτουν το αποτύπωμα της εφαρμογής.
- **Bench Suite API:** Η επικοινωνία με το Benchmarking Suite είναι αναγκαία ώστε να μπορέσουν να εκτελεστούν τα απαραίτητα Benchmarks στους εικονικούς πόρους.

Βάσεις δεδομένων

Η αποθήκευση των αποτυπωμάτων των Benchmarks, καθώς και η προσωρινή αποθήκευση των αποτυπωμάτων των εφαρμογών γίνονται σε μια Maria DB. Επικοινωνία επίσης υπάρχει και με την βάση QoE DB ώστε μετά την επιτυχή κατηγοριοποίηση μιας εφαρμογής από το υπό-εργαλείο Classifier να μπορέσει το σύστημα να προτείνει στον χρήστη μια Cloud υπηρεσία που έχει καλή συγκριτική απόδοση στη συγκεκριμένη κατηγορία Benchmark.

Profiler Module

Η διαδικασία για τη δημιουργία προφίλ μιας εφαρμογής έχει δύο μορφές, η μια μορφή είναι απευθείας σε ένα VM στο φυσικό μηχάνημα και η άλλη σε ένα απομακρυσμένο VM σε Cloud περιβάλλον Openstack. Σε ό,τι αφορά το φυσικό μηχάνημα η διαδικασία χρειάζεται την συμπλήρωση βασικών στοιχείων μέσω μιας διεπαφής (Εικόνα 20).



Εικόνα 20: Διεπαφή παραμετροποίησης εργαλείου εξαγωγής προφίλ.

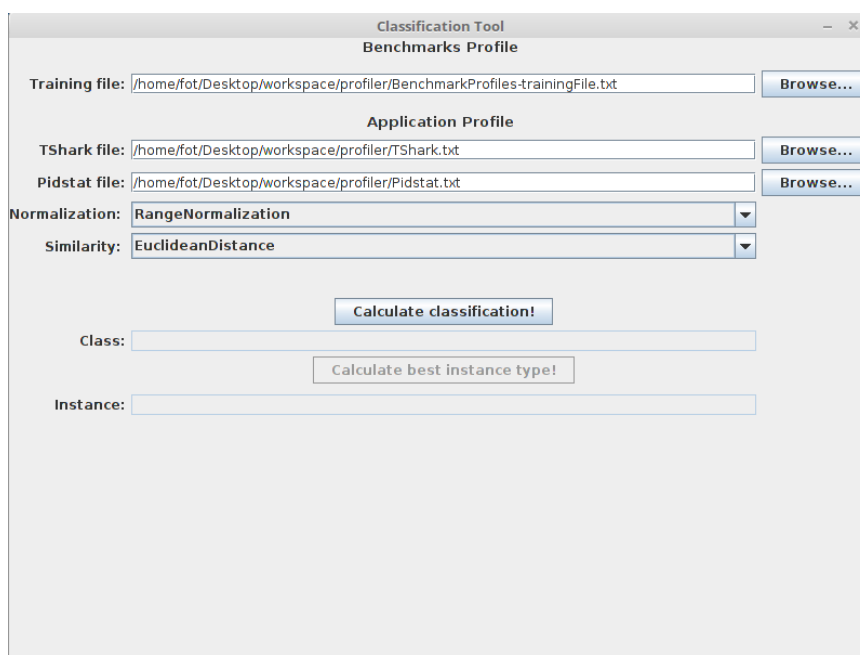
Η πρώτη πληροφορία που χρειάζεται να εισάγει ο χρήστης για την εκτέλεση της διαδικασίας δημιουργίας προφίλ είναι μια τοποθεσία στο σκληρό δίσκο για την αποθήκευση των μετρικών που τα εργαλεία παρακολούθησης δημιουργούν. Όπως παρουσιάζεται και στον Πίνακα 6, υπάρχει μια πληθώρα από μετρικές όπου εξάγονται για την δημιουργία προφίλ. Για να καταστεί δυνατή αυτή η εξαγωγή το εργαλείο προφίλ χρησιμοποιεί 2 εξωτερικά εργαλεία που έχουν δημιουργηθεί κατά κύριο λόγο για αυτή την διαδικασία. Τα δυο εργαλεία αυτά είναι το PidStat και το TShark. Το PidStat είναι υπεύθυνο για την εξαγωγή όλων των μετρικών εξαιρουμένων αυτών που αφορούν το δίκτυο, οι μετρικές που αφορούν το δίκτυο εξάγονται από το εργαλείο TShark.

Η ειδοποιός διαφορά για την εκτέλεση σε Cloud περιβάλλον είναι ότι δε χρησιμοποιούνται τα εργαλεία PidStat και TShark, αλλά εξάγονται οι μετρικές με τη

χρήση της διεπαφής Ceilometer API. Σημαντικές πληροφορίες στην διαδικασία αυτή είναι η γνώση της διεύθυνσης IP του OpenStack καθώς και τα διαπιστευτήρια(tokens) του Ceilometer API.

Classifier Module

Η διαδικασία της κατηγοριοποίησης εφαρμογών/λογισμικού έπεται της εξαγωγής προφίλ και είναι ίδια ανεξαρτήτου αν το προφίλ έχει εξαχθεί από φυσικό μηχάνημα ή από Cloud υποδομή. Χρησιμοποιώντας την διεπαφή του Classification Tool (Εικόνα 21) οι χρήστες μπορούν να κατηγοριοποιήσουν το λογισμικό σε ένα σύνολο κατηγοριών Benchmark.



Εικόνα 21: Διεπαφή κατηγοριοποίησης εφαρμογών.

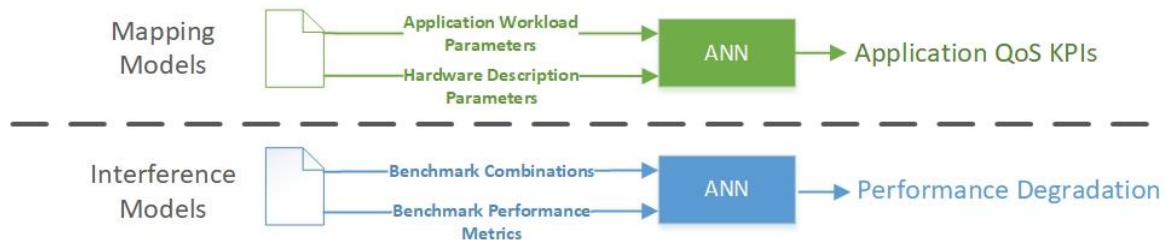
Για να ολοκληρωθεί η διαδικασία της κατηγοριοποίησης ο χρήστης θα πρέπει να εισάγει ένα σύνολο αρχείων και παραμέτρων. Το Training file περιέχει όλα τα προφίλ που εξήχθησαν από την εκτέλεση των Benchmarks και αποτελούν πολύ σημαντικά δεδομένα καθώς αποτελούν τα διανύσματα όλων των κλάσεων και κατηγοριών όπου οι εφαρμογές θα συσχετιστούν. Στην συνέχεια πρέπει να εισαχθούν από τον χρήστη τα διανύσματα

χρήσης που εξήχθησαν από τα εργαλεία PidStat και Tshark ώστε να μπορέσουν να συσχετιστούν με τα διανύσματα των Benchmarks. Τέλος, ο χρήστης έχει την επιλογή να παραμετροποιήσει το εργαλείο περαιτέρω, πιο συγκεκριμένα να επιλέξει την μέθοδο κανονικοποίησης των παραμέτρων (Normalization) των διανυσμάτων καθώς και να επιλέξει την συνάρτηση υπολογισμού απόστασης διανυσμάτων (Similarity) που θα χρησιμοποιήσει ο αλγόριθμος k-NN.

4.4 Μοντέλα παρεμβολών και κατανομής πόρων.

Όλα τα κομμάτια λογισμικού που έχουν αναλυθεί μέχρι στιγμής έχουν δημιουργηθεί για την αξιολόγηση και ανάλυση της απόδοσης των Cloud υποδομών. Ωστόσο, στα πλαίσια της παρούσας διατριβής δημιουργήθηκαν επίσης λύσεις χρησιμοποιώντας Τεχνητά Νευρωνικά Δίκτυα (Artificial Neuron Network ANN) με σκοπό τη βελτίωση της ποιότητας υπηρεσίας καθώς και την βέλτιστη τοποθέτηση εφαρμογών σε περιβάλλοντα Cloud. Η δόμηση και υλοποίηση των νευρωνικών δικτύων που δημιουργήθηκαν (μοτίβα παρεμβολών και κατανομής πόρων) έχει μια κύρια ειδοποιό διαφορά (Εικόνα 22: Σύσταση αλγορίθμων ANN για τα μοτίβα κατανομής και παρεμβολών (Εικόνα 22), τα δεδομένα που χρησιμοποιούνται για την εκπαίδευση. Στην περίπτωση των Μοντέλων Παρεμβολών, οι Πάροχοι θα πρέπει πρώτα να εγκαταστήσουν και να εκτελέσουν διάφορους συνδυασμούς Benchmark στον ίδιο φυσικό κόμβο, προκειμένου να συλλέξουν τα απαραίτητα δεδομένα. Στην περίπτωση των μοντέλων κατανομής, οι χρήστες θα πρέπει να μετρούν τις παραμέτρους QoS της εφαρμογής τους σε διαφορετικές ρυθμίσεις υλικού και διαφορετικό φόρτο εργασίας. Και οι δύο αυτές λειτουργίες εκτελούνται εξωτερικά από τους αντίστοιχους χρήστες χρησιμοποιώντας τα δικά τους εργαλεία παρακολούθησης. Η διαδικασία συλλογής δεδομένων είναι εκτός του πεδίου εφαρμογής του συγκεκριμένου

εργαλείου. Μετά τη διαδικασία εκπαίδευσης, οι πάροχοι IaaS και οι κάτοχοι εφαρμογών μπορούν να συμβουλευτούν τα αντίστοιχα εκπαιδευμένα μοντέλα και να εξάγουν προβλέψεις.



Εικόνα 22: Σύσταση αλγορίθμων ANN για τα μοτίβα κατανομής και παρεμβολών.

Μοντέλα παρεμβολών

Ο στόχος αυτού του εργαλείου είναι να αναλύσει και να προβλέψει την επίδραση μιας σειράς κρίσιμων παραμέτρων στην απόδοση των VM, όπως οι παρεμβολές που μπορεί να δημιουργήσει η ταυτόχρονη τοποθέτηση VM που τρέχουν διαφορετικής μορφής και απαιτήσεων σε φυσικούς πόρους λογισμικού στον ίδιο φυσικό κόμβο. Για το σκοπό αυτό, το εργαλείο διερευνά διάφορους συνδυασμούς τύπων φόρτου εργασίας σε VM σε σχέση με τους προαναφερθέντες παράγοντες, προκειμένου να βρει τα βέλτιστα μοτίβα κατανομής. Για όλες αυτές τις περιπτώσεις, μετράται η επίδραση στις μετρικές που εξάγουν συγκεκριμένα Benchmark που εκτελούνται εντός των VM. Στη συνέχεια, μια μέθοδος μαύρου κουτιού που βασίζεται σε γενετικά βελτιστοποιημένα ANN εισάγεται προκειμένου να διερευνηθεί η ικανότητα πρόβλεψης υποβάθμισης απόδοσης, χωρίς την πρότερη ανάγκη εκτέλεσης και συγκρίνεται με τη μέθοδο της γραμμικής παλινδρόμησης. Αυτή η «εκπαίδευση» των μοντέλων παρεμβολών καταλήγει στην πρόβλεψη των «καλύτερων» συνδυασμών Benchmark, με κριτήριο την υποβάθμιση της απόδοσης.

Μοντέλα κατανομής πόρων

Τα μοντέλα κατανομής πόρων παρέχουν μια γενικευμένη προσέγγιση μηχανικής μάθησης, για τη συσχέτιση των καθορισμένων απαιτήσεων (QoS) μιας εφαρμογής SaaS (για διάφορους φόρτους εργασίας) σε χαρακτηριστικά υλικών απαιτήσεων (όπως αυτά χρειάζονται για να εκφραστούν στο SLA με τον πάροχο IaaS). Συγκεκριμένα, το πώς ο αριθμός των πόρων που απαιτούνται για μια συγκεκριμένη τύπου εφαρμογής μπορεί να εκτιμηθεί με βάση τα αντίστοιχα χαρακτηριστικά QoS, όπως οι καθορισμένοι περιορισμοί QoS και ο συγκεκριμένος φόρτος εργασίας. Τα μοντέλα κατανομής πόρων βασίζονται σε Τεχνητά Νευρωνικά Δίκτυα στο GNU Octave⁶ και βασίζονται ξεχωριστά σε κάθε κομμάτι μιας εφαρμογής που ποικίλλουν ανάλογα με κάθε περίπτωση χρήσης καθώς και σε διαφορετικές διαμορφώσεις υλικού διαθέσιμες ανά Πάροχο Cloud υπηρεσιών.

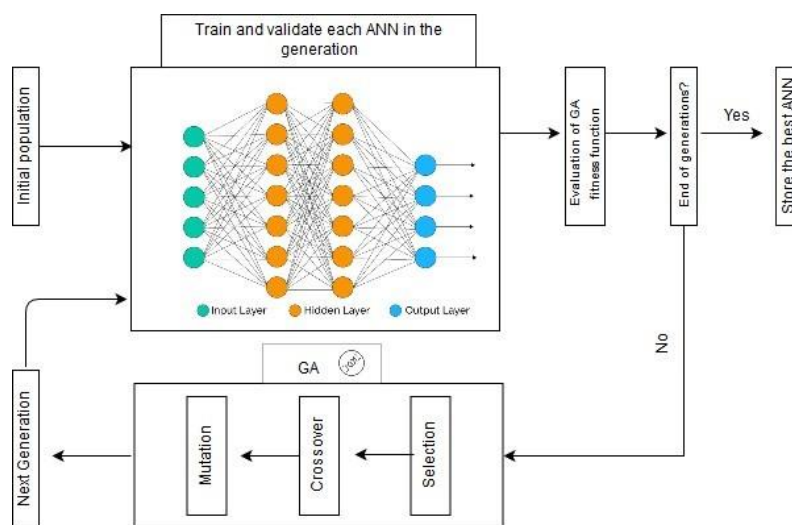
Στο παρακάτω υποκεφάλαιο αναλύεται η προσέγγιση με την οποία δημιουργήθηκε το σύστημα πρόβλεψης παρεμβολών καθώς και απόδοσης συστήματος. Πιο συγκεκριμένα, αναλύεται ο τρόπος με τον οποίο συλλέχθηκαν τα δεδομένα για τη δημιουργία των μοντέλων καθώς και πια ήταν η διαδικασία για να μπορέσουν να δημιουργηθούν τα μοντέλα μέσω της χρήσης γενετικών αλγορίθμων (GA σε ANN δίκτυα. Είναι σημαντικό να σημειωθεί ότι ο αναλυτικός τρόπος συλλογής των δεδομένων καθώς και το περιβάλλον το οποίο μελετήθηκαν και αξιολογήθηκαν αυτά τα μοντέλα παρουσιάζεται στο κεφάλαιο 5.1.1 και 5.1.2.

⁶ <https://www.gnu.org/software/octave/>

4.4.1 Ανάλυση τεχνικής υλοποίηση των ANN δικτιών

Ως πρώτο βήμα, ο διαχειριστής της εφαρμογής SaaS (Πάροχο Cloud υπηρεσιών) πρέπει να ορίσει τις παραμέτρους της εφαρμογής (φόρτος εργασίας, πόροι και KPI) για να χρησιμοποιηθούν ως είσοδοι και έξοδοι μοντέλων αντίστοιχα. Αυτή η διαδικασία είναι απαραίτητη για να προσαρμοστεί σε κάθε περίπτωση (σύνολο δεδομένων) που εξετάζεται. Τα ANN χρησιμοποιούν μη-γραμμική προσέγγιση, με λογική μαύρου κουτιού, για να συλλάβουν την εξάρτηση της εξόδου από την εφαρμοζόμενη είσοδο, όπως απεικονίζεται και στην Εικόνα 22.

Η υλοποίηση χρησιμοποιεί τη συνάρτηση «newff» της GNU Octave για την δόμηση του δικτιού ANN, η οποία βασίζεται σε δίκτυα πρόσθιας τροφοδότησης (feed forward) ανάστροφης αναδιάδοσης (back propagation) εκπαιδευμένα με τον αλγόριθμο Levenberg-Marquardt. Ο πυρήνας του μηχανισμού δημιουργίας μοντέλων αποτελείται από ένα δίκτυο ANN, βασισμένο στο εργαλείο Octave. Για τη φάση της εκπαίδευσης, το δίκτυο ANN χρειάζεται ένα αντιπροσωπευτικό σύνολο εκτελέσεων που παρέχεται από τον χρήστη των υπηρεσιών IaaS σε μορφή αρχείου Octave. Αυτό το σύνολο δεδομένων εκπαίδευσης θα χρησιμοποιηθεί για τον προσδιορισμό των βαρών του δικτύου και τον εντοπισμό σύνθετων, γραμμικών ή μη γραμμικών εξαρτήσεων της εξόδου από τις εισόδους.



Εικόνα 23: Νεύρο-εξελικτική διαδικασία εκμάθησης ANN[5].

Οι παράμετροι σχεδιασμού του ANN διαμορφώνονται από έναν γενετικό αλγόριθμο (Genetic Algorithm GA), προκειμένου να βελτιστοποιηθεί η επιλογή τους. Γενικά, η επιλογή αυτών των παραμέτρων είναι μια επίμοχθη διαδικασία, βασισμένη στην ανθρώπινη εμπειρία. Ωστόσο, δεν μπορεί να επαναληφθεί εύκολα και να εφαρμοστεί με αυτοματοποιημένο τρόπο, έτσι στην τρέχουσα προσέγγιση και για τα μοντέλα παρεμβολών καθώς και για τα μοντέλα κατανομής πόρων, χρησιμοποιήθηκαν GA. Οι παράμετροι σχεδιασμού που περιλαμβάνονται στη διαδικασία βελτιστοποίησης περιλαμβάνουν τον αριθμό των κρυφών στρωμάτων, τον αριθμό των νευρώνων ανά κρυφό στρώμα και τον τύπο των συναρτήσεων μεταφοράς κάθε στρώματος δικτύου, από τους τρεις διαθέσιμους τύπους στο Octave (tansig, logsig και pureline⁷). Όπως απεικονίζεται στην Εικόνα 23, η διαδικασία της νευρο-εξέλιξης ξεκινά με έναν αρχικό πληθυσμό τυχαίων δομές ANN. Σε κάθε γενιά, ο GA επιλέγει τα καταλληλότερα ANNs που θα διασταυρωθούν με βάση το σύστημα πιθανότητας σχετικά με την απόδοσή τους

⁷ <http://radio.feld.cvut.cz/matlab/toolbox/nnet/backpr52.html>

(διαδικασία επιλογής). Από αυτή τη διαδικασία, θα δημιουργηθούν νέες δομές ANN και μετά την απαραίτητη διαδικασία μετάλλαξης (προκειμένου να διατηρηθεί κάποια τυχαιότητα στο GA) παράγεται μια νέα γενιά ANN, έτοιμα να εκπαιδευτούν και να επικυρωθούν. Αυτή η διαδικασία συνεχίζεται μέχρι να επιτευχθεί ο προκαθορισμένος αριθμός γενεών. Είναι σημαντικό να σημειωθεί ότι, η σωστή προεπεξεργασία των δεδομένων διασφαλίζει τη συμπίληψη των ελάχιστων και μέγιστων τιμών στο υποσύνολο εκπαίδευσης, προκειμένου να αποφευχθούν οι κίνδυνοι παρέκτασης(extrapolation).

Το κριτήριο της εξελικτικής απόδοσης για τον GA, δηλαδή της συνάρτησης Fitness είναι το ποσοστό σφάλματος στο σύνολο των δεδομένων εκπαίδευσης. Ωστόσο, οι παραλλαγές με το κριτήριο απόδοσης ως το σφάλμα στο ενδιάμεσο σύνολο δεδομένων επικύρωσης είχαν δοκιμαστεί στην αρχική έκδοση του αλγορίθμου, το οποίο φαινόταν να έχει χειρότερα αποτελέσματα. Σε κάθε γενιά, τα παραγόμενα μοντέλα αξιολογούνται και επιλέγονται, με βάση την απόδοση πρόβλεψής τους στο ενδιάμεσο σύνολο επικύρωσης, ενώ η τελική επιλογή πραγματοποιείται σύμφωνα με την απόδοση πρόβλεψης στο τρίτο και τελευταίο υποσύνολο (σύνολο δεδομένων δοκιμής) των αρχικών δεδομένων. (περίπου το 30%), το οποίο δεν χρησιμοποιείται κατά την διαδικασία εκπαίδευσης.

Η διαδικασία για την δημιουργία των μοντέλων συνίσταται από τα παρακάτω βήματα:

- Δημιουργία των κατάλληλων σημείων παρακολούθησης των KPI μέσα στην δομή της εφαρμογής. Είναι σημαντικό να σημειωθεί ότι τα KPI των μοντέλων παρεμβολών διαφέρουν σε σχέση με αυτά των μοντέλων κατανομής. Στην περίπτωση των μοντέλων κατανομής αποτελούν συγκεκριμένες μετρικές που αφορούν την ποιότητα υπηρεσίας, ενώ στην περίπτωση των μοντέλων παρεμβολών αφορούν μετρικές που εξάγονται από τα Benchmarks.

- Ορισμός διαφορετικών τύπων υπολογιστικών πόρων στην περίπτωση των μοντέλων κατανομής και ορισμός του συνόλου των Benchmarks στην περίπτωση στον μοντέλων παρεμβολών.
- Συλλογή και δημιουργία των σετ δεδομένων. Όπως απεικονίζεται και στην Εικόνα 22 κάθε μοντέλο έχει την δική του σειρά από δεδομένα που τροφοδοτούνται στο μοντέλο.
- Τροφοδοσία αρχείου στον αλγόριθμο ANN μέσω της διεπαφής χρήσης του προγράμματος.
- Αρχικοποίηση και έναρξη της φάσης εκπαίδευσης και επικύρωσης για τα μοντέλα.
- Έλεγχος των αποτελεσμάτων πρόβλεψης και αποθήκευση των υποψήφιων μοντέλων.
- Χρήση μοντέλου για την πρόβλεψη νέων περιπτώσεων/συνδυασμών.

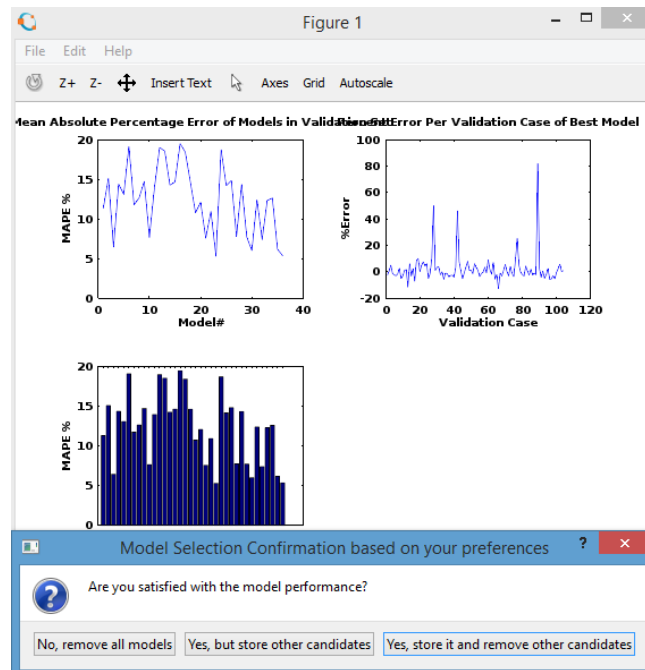
Το εργαλείο για την δημιουργία των μοντέλων αποτελείται από τρία κύρια υποσυστήματα λογισμικού. Τα Model Creator, Model Selection, ANN-Script, παράλληλα χρησιμοποιούνται και διάφορες βοηθητικές συναρτήσεις για κανονικοποίηση των δεδομένων. Η διαδικασία καθώς και η λειτουργία αυτών των scripts είναι η ακόλουθη:

- Το **Model Creator script** είναι υπεύθυνο για το τμήμα που αφορά την διεπαφής χρήστη, τη λήψη αρχείων εισόδου και διαμόρφωσης των διευθύνσεων εισόδου από τον χρήστη (π.χ. τοποθεσία αποθήκευσης των μοντέλων, το μοναδικό αναγνωριστικό των μοντέλων(ID), αριθμό των στηλών εισόδου του αρχείου csv που εισάγεται κ.λπ.). Ο χειρισμός του αρχείου καθώς και η αναγνώριση βασίζεται στο αναγνωριστικό μοντέλου(model ID), το οποίο

πρέπει να είναι μοναδικό. Επιπλέον, ο χρήστης του IaaS ερωτάται εάν το KPI είναι αύξων ή φθίνων και αν προτιμά υπερτροφοδότηση ή υποτροφοδότηση, χαρακτηριστικό που θα ληφθεί υπόψη κατά την επιλογή του τελικού μοντέλου.

- Μετά την ολοκλήρωση της ρύθμισης, εκκινείτε το **Model Selection script**, το οποίο είναι υπεύθυνο για την ανάκτηση των δεδομένων, την κανονικοποίηση τους, το χωρισμό τους σε τρία υποσύνολα (65% εκπαίδευση, 15% ενδιάμεση επικύρωση και 20% τελικό σύνολο δοκιμής) και τη ρύθμιση του τμήματος του GA που αφορά τον κώδικα πριν εκτελεστεί το ANN-script μέσα στη συνάρτηση GA. Επιπλέον, είναι υπεύθυνος για τον εντοπισμό, στο τέλος της διαδικασίας, του καλύτερου μοντέλου χρησιμοποιώντας το σύνολο δοκιμής με γνώμονα το σφάλμα πάνω σε αυτό.
- Το **ANN-script** είναι η βασική συνάρτηση που λαμβάνει μια προτεινόμενη αρχιτεκτονική ANN από το GA (chromosome) και είναι υπεύθυνη για τη δημιουργία του αντίστοιχου δικτύου με βάση τις παραμέτρους, καθώς και την εκπαίδευση και τη μέτρηση του σφάλματος του στο ενδιάμεσο σετ επικύρωσης.

Μετά την ολοκλήρωση της διαδικασίας, το τμήμα διεπαφής χρήστη αναλαμβάνει τον έλεγχο και παρουσιάζει τα αποτελέσματα (σφάλμα όλων των υποψηφίων μοντέλων και λεπτομερές σφάλμα του καλύτερου μοντέλου σε όλες τις περιπτώσεις επικύρωσης), ρωτώντας τον χρήστη εάν το αποτέλεσμα ήταν ικανοποιητικό, οπότε και αποθηκεύει το μοντέλο για μελλοντική χρήση, όπως απεικονίζεται στην Εικόνα 24.



Εικόνα 24: Διεπαφή χρήστη για επιλογή μοντέλου[5].

5

Αξιολόγηση λειτουργίας και απόδοσης

υποσυστημάτων

Στο παρόν κεφάλαιο παρουσιάζονται οι αξιολογήσεις της λειτουργικότητας καθώς και της απόδοσης των διαφόρων κομματιών λογισμικού που δημιουργήθηκαν στα πλαίσια της διατριβής αυτής. Όλα αυτά τα κομμάτια λογισμικού έχουν διαφορετικές λειτουργίες, αφορούν διαφορετικούς τύπους χρηστών και έχουν αναπτυχθεί για την επίλυση διαφορετικών φύσεως προβλημάτων. Δοθείσης της ετερογένειας του λογισμικού χρησιμοποιήθηκαν διαφορετικές μέθοδοι για την αξιολόγηση των υποσυστημάτων.

Πιο συγκεκριμένα για το υποσύστημα QoE(3ALib, Bench Suite) η αξιολόγηση περιλαμβάνει τη μελέτη της σταθερότητας του συστήματος καθώς και της απόδοσης κατά την κλιμάκωση των διεργασιών. Ενώ για τα υποσυστήματα ANN, για τη δημιουργία μοντέλων παρεμβολών και κατανομής πόρων, η αξιολόγηση εστίασε στην ακρίβεια πρόβλεψης.

5.1 Αξιολόγηση λειτουργικότητας και αποδοτικότητας υποσυστήματος QoE(3ALib, Bench Suite)

Στα πλαίσια της αξιολόγησης του υποσυστήματος QoE η προσοχή στράφηκε στην αξιολόγηση της σταθερότητας του συστήματος, δηλαδή το κατά πόσο οι κύριες διεργασίες που χρειάζεται να εκτελεστούν για τις κύριες λειτουργίες του συστήματος εκτελούνται με επιτυχία. Επιπροσθέτως, σημαντικός παράγοντας για την ποιότητα των συγκεκριμένων συστημάτων είναι το κατά πόσο είναι ικανά να κλιμακωθούν με απώτερο σκοπό την εξυπηρέτηση περισσότερων χρηστών.

5.1.1 Αξιολόγηση σταθερότητας διεργασιών συστήματος

Για τις ανάγκες της εκτίμησης της σταθερότητας του συστήματος επιλέχθηκαν συγκεκριμένες διεργασίες από τα δυο υποσυστήματα του συστήματος QoE(3ALib, Bench Suite). Οι διεργασίες που περιγράφονται στον παρακάτω πίνακα (Πίνακας 16) επιλέχθηκαν διότι είναι απαραίτητες για την εκτέλεση των βασικών λειτουργιών του QoE, την αξιολόγηση SLA και την εκτέλεση και συλλογή δεδομένων Benchmarks.

Διεργασία	Περιγραφή
3ALib	
Εκκίνηση διεργασίας παρακολούθησης SLA (Start a new SLA Agent)	Η πρώτη διεργασία για να μπορέσουν να συλλεχθούν τα απαραίτητα αρχεία καταγραφής για την αξιολόγηση ενός SLA. Μέσο της διεπαφής QoE UI παραμετροποιείται ένα καινούρια αίτημα για την δημιουργία ενός Docker service (SLA

	Agent) το οποίο εμπεριέχει τους κατάλληλους drivers για την επικοινωνία με την εκάστοτε υπηρεσία Cloud.
Συλλογή και αποθήκευση αρχείων καταγραφής μετρικών SLA(Collect SLA log file metrics)	Οι SLA agents που δημιουργήθηκαν από την προηγούμενη διεργασία συλλέγουν τα κατάλληλα αρχεία καταγραφής και τα αποθηκεύουν στις ορισμένες συλλογές σε μια MongoDB (QoE DB).
Υπολογισμός τελικών μετρικών SLA(Final SLA metric calculation)	Όλα τα αρχεία καταγραφής που αποθηκεύτηκαν στην QoE DB εξάγονται κατά την ολοκλήρωση της περιόδου αξιολόγησης και υπολογίζονται τα τελικά SLA metrics για την αξιολόγηση του SLA.
Bench Suite	
Εκκίνηση μιας εκτέλεσης Benchmark (single Benchmark execution)	Για την εκτέλεση ενός μοναδικού Benchmark γίνεται η κατάλληλη παραμετροποίηση από το QoE UI. Η παραμετροποίηση αυτή αφορά την εισαγωγή πληροφοριών για την σύνδεση με την εκάστοτε υπηρεσία Cloud καθώς και πληροφορίες για το συγκεκριμένο Benchmark που επρόκειτο να εκτελεστεί. Μετά την εισαγωγή των κατάλληλων πληροφοριών δημιουργείτε ένα Docker Container το οποίο μετά την επιτυχή εκτέλεση και συλλογή/αποθήκευση δεδομένων(στην QoE DB) θα διαγραφεί μόνο του.

<p>Εκκίνηση ενός ενορχηστρωτή Benchmark (Benchmark scheduler deployment)</p>	<p>Για να καταστεί δυνατή η εκτέλεση πολλαπλών Benchmarks χρειάζεται η παραμετροποίηση ενός ενορχηστρωτή Benchmark ο οποίος θα είναι υπεύθυνος για την οργάνωση και εκτέλεση ενός Benchmark πολλαπλές φορές η και την εκτέλεση πολλαπλών Benchmark. Όπως και την διεργασία της απλής εκτέλεσης Benchmark δημιουργείτε το απαραίτητο Docker Container το οποίο σε αντίθεση με την απλή εκτέλεση δεν θα διαγραφεί, αλλά θα παραμείνει ανενεργό σε περίπτωση που ο χρήστης θέλει να επαναλάβει τις εκτελέσεις.</p>
<p>Εκτέλεση πολλαπλών Benchmarks μέσω του ενορχηστρωτή (Multiple Benchmark deployment through scheduler)</p>	<p>Στην αυτοματοποιημένη διαδικασία εκτέλεσης Benchmark μέσω του ενορχηστρωτή, εκτελούνται τα ίδια βήματα όπως στην απλή εκτέλεση Benchmark με την ειδοποιό διαφορά να είναι ότι δεν δημιουργείται εκ' νέου ένα καινούριο Docker Container. Τη σύνδεση με την Cloud υπηρεσία, όπως και την συλλογή/αποθήκευση των δεδομένων, την διαχειρίζεται το Container του ενορχηστρωτή.</p>

Πίνακας 7: Λίστα αξιολογημένων διεργασιών για την σταθερότητα συστήματος QoE.

ο εξαντλητικό έλεγχο της σταθερότητας των διεργασιών πραγματοποιήθηκε διότι για να μπορέσουν να εκτελεστούν οι βασικές λειτουργίες του QoE, που είναι η αξιολόγηση της αξιοπιστίας των παρόχων IaaS μέσω της αξιολόγησης των SLA, καθώς και την

αξιολόγηση απόδοσης των IaaS μέσω Benchmarks, χρειάζεται να εγκατασταθεί ένα σύνολο από Docker Container και να επιτευχθούν επικοινωνίες και ανταλλαγές δεδομένων από διαφορετικά συστήματα(QoE, Docker, Cloud service, Mongo DB). Σε ένα τόσο ετερογενές και σύνθετο περιβάλλον είναι σημαντικό να μελετάται το κατά πόσο η βασικές λειτουργίες μπορούν να αποδίδουν χωρίς λάθη και δυσλειτουργίες. Είναι σημαντικό να αναφερθεί ότι ο εξαντλητικός έλεγχος της σταθερότητας του συστήματος δεν μπορεί να πραγματοποιηθεί με την χρήση της διεπαφής QoE UI διότι η παραμετροποίηση και η εκτέλεση του αριθμού των αιτημάτων, θα ήταν αποτρεπτικά χρονοβόρα. Επίσης, ο βασικός λόγος για την αποτυχία μιας διεργασίας μέσω της χρήσης του QoE UI είναι η λάθος παραμετροποίηση των αιτημάτων. Για τις ανάγκες των πειραμάτων επιλέχθηκαν συγκεκριμένες παραμετροποιήσεις όπου έχουν ελεγχθεί για την εγκυρότητά τους, και τα αιτήματα παράγονται αυτόματα από ένα εργαλείο παραγωγής API αιτημάτων το JMeter⁸.

Οι συγκεκριμένες Cloud υποδομές που χρησιμοποιήθηκαν για την πειραματική εκτέλεση των διεργασιών του συστήματος είναι οι ακόλουθες:

- Amazon EC2 t2.nano
- Filab 2 πυρήνες 8GB RAM (Openstack Pike)
- Τοπικό Openstack Rocky

Ο συνολικός αριθμός των εκτελέσεων της κάθε διεργασίας είναι 1000 φορές, για τις ανάγκες των διεργασιών που αφορούν την αξιολόγηση των SLA οι εκτελέσεις χωρίστηκαν ισόποσα(33% Amazon, 33% Filab, 33%Openstack) σε κάθε υποδομή διότι

⁸ <https://jmeter.apache.org/>

ήταν δυνατή η αξιολόγηση όταν οι εικονικοί πόροι ήταν σε κατάσταση αναμονής. Λόγω του αυξημένου κόστους στην χρήση των πυρήνων στην υποδομή Amazon οι διεργασίες δεν χωριστήκαν ισόποσα, το 10% των εκτελέσεων έγινε στην Amazon και το υπόλοιπο 90% διχοτομήθηκε στις άλλες υποδομές.

Διεργασία	Amazon EC2	Filab	Openstack Rocky	Συνολικό μέσο ποσοστό επιτυχίας
3ALib αξιολόγηση SLA				
Εκκίνηση διεργασίας παρακολούθησης SLA	97%	91%	98%	95%
Συλλογή και αποθήκευση αρχείων καταγραφής μετρικών SLA	93%	88%	99%	93%
Υπολογισμός τελικών μετρικών	N/A	N/A	N/A	92%
Bench Suite αξιολόγηση απόδοσης Benchmark				
Εκκίνηση μιας εκτέλεσης Benchmark	72%	76%	85%	78%
Εκκίνηση ενός ενορχηστρωτή Benchmark	N/A	N/A	N/A	89%
Εκτέλεση πολλαπλών Benchmarks μέσο του ενορχηστρωτή	63%	73%	81%	72%

Πίνακας 8: Αποτελέσματα αξιολόγησης σταθερότητας διεργασιών QoE.

Οι διεργασίες που αφορούν την αξιολόγηση SLA παρουσιάζουν καλύτερα ποσοστά επιτυχίας σε σχέση με αυτές για τα Benchmark (Πίνακας 8), λογική συνέπεια μιας και οι

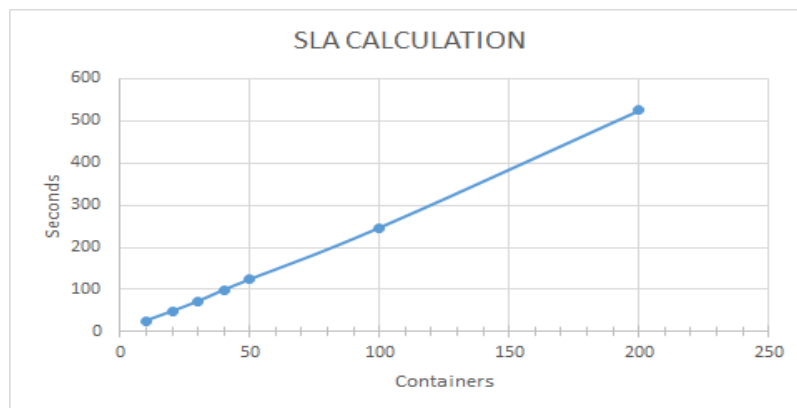
διεργασίες που αφορούν το SLA δεν προϋποθέτουν την εγκατάσταση λογισμικού στην αξιολογούμενη Cloud υπηρεσία. Είναι σημαντικό να σημειωθεί ότι κατά την εκτέλεση των πειραμάτων δεν συμπεριλήφθηκαν αποτυχημένες εκτελέσεις όπου ο λόγος αποτυχίας δεν είναι σφάλμα του συστήματος QoE όπως η διακοπή του διαδικτύου, πτώση ρεύματος, η προβλήματα στο Docker Engine .Τα πεδία των διεργασιών που περιέχουν N/A Μη Εφαρμόσιμο (Non Applicable) είναι διότι αυτές οι διεργασίες δεν προϋποθέτουν σύνδεση με τον διακομιστή της εκάστοτε Cloud Υπηρεσίας καθώς αντλούν τα δεδομένα από την βάση του QoE. Για το λόγο αυτό συνυπολογίζοντας ότι κάθε διεργασία ελέγχεται αυτοτελώς, δεν έγινε διαχωρισμός του συνόλου δεδομένων ανά πάροχο, καθώς δεν έχει σημασία για αυτές τις διεργασίες.

5.1.2 Αξιολόγηση αποδοτικότητας συστήματος.

Για το σκοπό της μελέτης αποδοτικότητας του υποσυστήματος αξιολόγησης SLA(3ALib) δημιουργήθηκαν και εκτελέστηκαν πειράματα τα οποία έχουν ως στόχο να εξετάσουν το πόσο κοστοβόρα σε υπολογιστικούς πόρους είναι οι βασικές διεργασίες του υποσυστήματος αυτού. Πιο συγκεκριμένα αυτό που εξετάστηκε ήταν το κατά πόσο ένα υπολογιστικό σύστημα μπορούσε να υποστηρίξει πολλαπλούς παράλληλους χρήστες(3ALib Agents). Όπως αναφέρθηκε και στο κεφάλαιο 4.1, κάθε 3ALib Agent είναι υπεύθυνος για την συλλογή των αρχείων καταγραφής αλλά και για τον υπολογισμό του τελικού SLA με βάση την περίοδο χρέωσης. Συνεπώς, ένας 3ALib Agent αντιπροσωπεύει ένα χρήστη του συστήματος QoE.

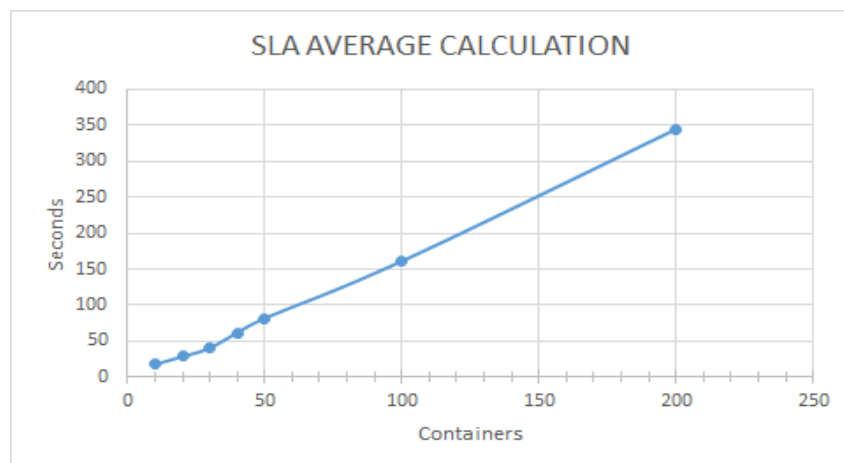
Για τις ανάγκες της εκτέλεσης των πειραμάτων χρησιμοποιήθηκε ένα σχετικά μικρό σε υπολογιστική ισχύ μηχάνημα, με 2.8GH, 3 πυρήνες 4GB of RAM, με λειτουργικό σύστημα Ubuntu Linux 16.04. Στο μηχάνημα αυτό δημιουργήθηκαν 3ALib Agent

(Docker Container) έως η απόδοση του συστήματος να γίνει μη διαχείριση από άποψη χρόνο εκτέλεσης διεργασιών υπολογισμού SLA.



Εικόνα 25: Συνολικός χρόνος υπολογισμού SLA 200 χρηστών[4].

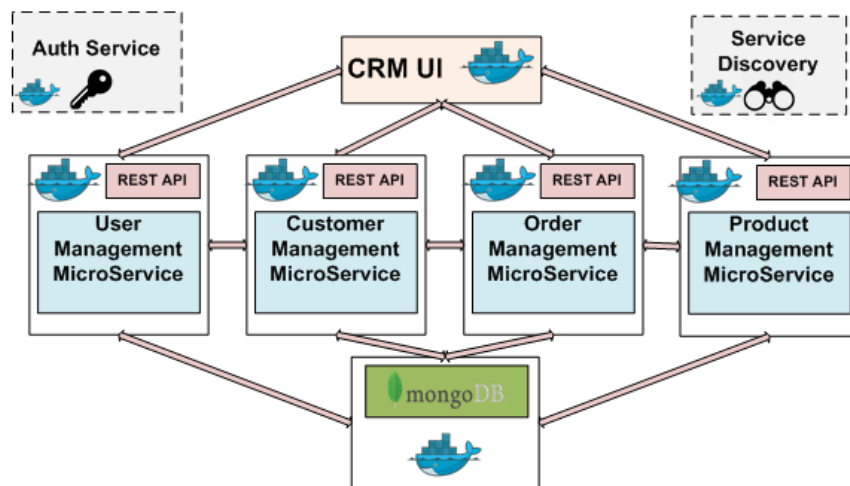
Όπως απεικονίζεται στην Εικόνα 25 ο μέγιστος συνολικός αριθμός παράλληλων SLA Agents που επιτεύχθηκε είναι 200 με συνολικό χρόνο για τον υπολογισμό του SLA να ανέρχεται στα εννέα λεπτά. Δοθείσης της περιορισμένης υπολογιστικής ισχύος του μηχανήματος καθώς και το ότι οι 200 SLA Agents τυπικά αντιπροσωπεύουν 200 χρήστες τα αποτελέσματα δείχνουν ότι οι διεργασίες αποδίδουν και μπορούν να κλιμακωθούν σε μεγάλο βαθμό για να καλύψουν τις ανάγκες μεγαλύτερου αριθμού χρηστών. Λαμβάνοντας υπόψιν ότι δεν είναι απαραίτητο αυτές οι διεργασίες να εκτελούνται παράλληλα όπως φαίνεται και στην Εικόνα 26 ο υπολογισμός ενός μοναδικού SLA ανά χρήστη χρειάζεται μόνο λίγα δευτερόλεπτα. Στην περίπτωση που υπάρχουν πολλοί παράλληλοι χρήστες ο χρόνος αυξάνεται γραμμικά πράγμα φανερώνει ότι με σωστό προγραμματισμό εκτέλεσης των διεργασιών δεν θα υπάρχουν εμφανείς καθυστερήσεις στους χρήστες.



Εικόνα 26: Μέσος χρόνος υπολογισμού SLA ανά χρήστη[4].

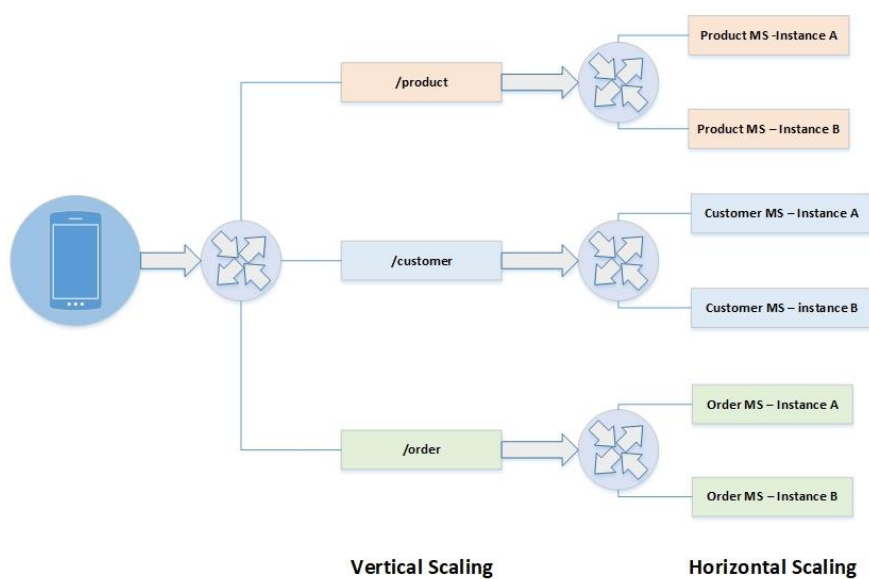
5.2 Αξιολόγηση ακρίβειας μοντέλων ANN

Όπως παρουσιάστηκε και στο κεφάλαιο 4.4 δημιουργήθηκαν δύο μοντέλα ANN, τα μοντέλα παρεμβολών και τα μοντέλα κατανομής πόρων. Για τους σκοπούς της αξιολόγησης της προσέγγισης δημιουργίας των μοντέλων ενορχηστρωθήκαν δυο διαφορετικά πειράματα αξιολόγησης χρησιμοποιώντας σαν βάση την ίδια κατανεμημένη εφαρμογή Customer Relationship Management/Enterprise Resource Planning (CRM/ERP). Η εφαρμογή CRM/ERP ακολουθεί τη σύγχρονη αρχιτεκτονική μικρό-υπηρεσιών, σύμφωνα με την οποία η λογική της εφαρμογής χωρίζεται σε πολλές ανεξάρτητες υπηρεσίες, καθεμία από τις οποίες επιφορτίζεται με την εκτέλεση μιας συγκεκριμένης εργασίας. Αυτό επιτρέπει στην εφαρμογή να είναι εύκολα επεκτάσιμη και να κατανέμεται σε πολλαπλά διαφορετικά περιβάλλοντα.



Εικόνα 27: Κάθετη δομή της εφαρμογής CRM/ERP.

Η Εικόνα 27 απεικονίζει τα στοιχεία της εφαρμογής, μαζί με τις διασυνδέσεις τους. Λόγω της φύσης της, η εφαρμογή μπορεί να κλιμακωθεί τόσο κατακόρυφα (αναθέτοντας τα στοιχεία της εφαρμογής σε μεγαλύτερα VM με περισσότερους πόρους υλικού) όσο και οριζόντια (προσθήκη περισσότερων VM προκειμένου να καταναμηθεί περαιτέρω ο φόρτος εργασίας), όπως φαίνεται στην Εικόνα 28.



Εικόνα 28: Παράδειγμα κλιμάκωσης εφαρμογής CRM/ERP.

5.2.1 Συλλογή δεδομένων και αξιολόγηση μοντέλων παρεμβολών

Προκειμένου να δημιουργηθούν τα δεδομένων για την εκπαίδευση των μοντέλων παρεμβολών, δημιουργήθηκε μια δοκιμαστική εφαρμογή βασισμένη σε scripts bash και NodeJS για να ενορχηστρώσει του εργαλείου δημιουργίας προφίλ και κατηγοριοποίησης (Profiler). Ο ενορχηστρωτής δημιουργεί κίνηση με ποικίλο φορτίο χρήστη (χρησιμοποιεί το artillery.io⁹ για αυτόν τον σκοπό), συμπληρώνει και ξεκινά το Profiler σε απομακρυσμένα VM, παρακολουθεί την εκτέλεση και, τέλος, δημιουργεί δομημένες αναφορές και σύνολα δεδομένων. Επίσης, ανακτά στατιστικές μετρήσεις (φόρτο CPU, χρήση μνήμης, χρόνος εκτέλεσης αιτήματος κ.λπ.) από κάθε μεμονωμένη υπηρεσία ERP/CRM.

Συλλογής δεδομένων κατηγοριοποίησης

Η κατηγοριοποίηση του λογισμικού και κατ' επέκταση κάθε στοιχείου της εφαρμογής CRM/ERP είναι αναγκαία διότι, για να μπορέσουν να εξαχθούν συμπεράσματα για τις παρεμβολές που μπορούν να δημιουργηθούν μεταξύ των υπηρεσιών που τρέχουν παράλληλα στους φυσικούς πόρους, πρέπει πρώτα το λογισμικό να καταταχθεί σε μια συγκεκριμένη κατηγορία. Με αυτό τον τρόπο τα μοτίβα παρεμβολών δεν είναι αναγκαίο να δημιουργήσουν καινούριο μοντέλο για κάθε καινούρια εφαρμογή που πρέπει να ελεγχθεί, αρκεί να δημιουργηθούν τα μοντέλα για κάθε κατηγορία λογισμικού. Το εργαλείο Profiler-Classification (κεφάλαιο 4.3) δίνει τη δυνατότητα της κατηγοριοποίησης του λογισμικού χρησιμοποιώντας Benchmarks. Όπως παρουσιάζεται και στο Πίνακα 9, τα κομμάτια της εφαρμογής κατηγοριοποιήθηκαν χρησιμοποιώντας

⁹ <https://artillery.io/>

μια ποικιλία από 10, 40 και 100 χρήστες (διεργασίες) ανά στοιχείο ERP/CRM. Αυτά αντιστοιχούν σε ελαφριά, μεσαία και βαριά κυκλοφορία αντίστοιχα (τα αιτήματα εκτελούνται ταυτόχρονα). Η στήλη «Κατηγορία Benchmark» αντιστοιχεί στην έξοδο του Classifier, ενώ ο «Μέσος Χρόνος Απόκρισης» είναι ο χρόνος απόκρισης της υπηρεσίας. Οι μετρήσεις στον παρακάτω πίνακα δημιουργούνται για ταυτόχρονα αιτήματα.

Εφαρμογή/ Υπηρεσία	#Διεργασίες	Μέσος Χρόνος Απόκρισης (msec)	Κατηγορία Benchmark
Users	10	141	Filebench-webproxy
	40	980	YCSB-workloadb
	100	5098	Filebench-fileserver
Products	10	1021	Filebench-webproxy
	40	6655	Filebench-fileserver
	100	14625	Filebench-fileserver
Orders	10	1412	Filebench-webproxy
	40	4767	Filebench-fileserver
	100	18507	Filebench-fileserver
Customers	10	280	YCSB-workloadd
	40	3229	YCSB-workloadd
	100	9083	DaCapo-eclipse

Πίνακας 9: Αναφορά κατηγοριοποίησης λογισμικού CRM/ERP σε Benchmarks.

Δημιουργία μοντέλων παρεμβολών

Προκειμένου να συγκεντρωθεί το σετ εκπαίδευσης και επικύρωσης για τη δημιουργία μοντέλου πρόγνωσης παρεμβολών, χρησιμοποιήθηκαν διαφορετικοί συνδυασμοί Benchmarks σε επίπεδο εφαρμογής (όπως DaCapo, YCSB και Filebench), μαζί με 6 παραλλαγές φόρτου εργασίας για κάθε περίπτωση (με αποτέλεσμα 18 διακριτά Benchmarks-workloads). Αυτές οι περιπτώσεις σχεδιάστηκαν να εκτελεστούν σε διάφορους συνδυασμούς στον φυσικό κόμβο προκειμένου να παρατηρηθεί η υποβάθμιση στις απόδοσής τους. Οι συνδυασμοί περιελάμβαναν διαφορετικές περιπτώσεις εκτέλεσης καθώς και διαφορετικά σενάρια ανάπτυξης στον κεντρικό υπολογιστή, ανάλογα με το αν οι εκχωρημένοι πυρήνες μοιράζονταν μία ή περισσότερες κρυφές μνήμες (cache memory). Αρχικά οι δοκιμές εκτελέστηκαν μεμονωμένα προκειμένου να ανακτηθεί ο χρόνος εκτέλεσης χωρίς τυχόν παρεμβολές από άλλες διεργασίες (Benchmarks). Μετά από αυτό το βήμα, οι δοκιμές ανατέθηκαν είτε σε γειτονικούς πυρήνες (κοινή χρήση κρυφής μνήμης L2 και L3) είτε σε μη γειτονικούς πυρήνες (κοινή χρήση μόνο προσωρινής μνήμης L3). Λόγω του γεγονότος ότι ο φόρτος εργασίας του κάθε Benchmark δίνετε να έχει διαφορετικό χρόνο εκτέλεσης δημιουργήθηκε ένα προκαθορισμένο χρονικό παράθυρο των 300 δευτερολέπτων για την εκτέλεση των Benchmark. Κατά τη διάρκεια των 300 δευτερολέπτων, κάθε δοκιμή εκτελείται πολλές φορές (στο εύρος εκατοντάδων φορές) λαμβάνοντας ως τελική βαθμολογία τον μέσο όρο αυτών των εκτελέσεων. Τα πειράματα συντονίστηκαν και συγχρονίστηκαν μέσω ενός προσαρμοσμένου προγράμματος συντονισμού σε Java (BCoordinator). Οι διάφοροι BClients που εκτελούνται σε κάθε VM προκειμένου να εκκινήσουν τη συγκεκριμένη δοκιμή εκτέλεσης Benchmark επικοινωνούν με τον BCoordinator μέσω πρωτοκόλλου TCP, έτσι ώστε να μπορούν να συγχρονιστούν και να διασφαλίσουν ότι η εκτέλεση της δοκιμής θα γίνει

πράγματι παράλληλα. Κατά τη σχεδίαση, λήφθηκε υπόψη ότι τα περισσότερα Benchmarks χρειάζονταν μια φάση προετοιμασίας ή οριστικοποίησης πριν ή μετά την κύρια εκτέλεσή τους. Ο χρονισμός αυτών των φάσεων δεν λήφθηκε υπόψη κατά την ανάκτηση των αποτελεσμάτων και οι BClients συγχρονίζονται για να περιμένουν έως ότου οι υπόλοιποι BClients να ολοκληρώσουν τις αντίστοιχες φάσεις. Αναλυτικά τα βήματα που ακολουθούν περιλαμβάνουν:

1. Κατά την εκκίνηση, ο BCoordinator αναμένει εισερχόμενες συνδέσεις από τους BClients.
2. Οι BClients συνδέονται μέσω υποδοχών TCP στον BCoordinator και περιμένουν ειδοποίηση, για να ξεκινήσει μια εκτέλεση.
3. Όταν ολοκληρωθεί το βήμα 2, ο BCoordinator ανακτά το αρχείο διαμόρφωσης, υποδεικνύοντας ποιος συνδυασμός δοκιμών πρέπει να εκτελεστεί και ποιος έλεγχος θα πρέπει να εκτελεστεί σε ποιο BClient.
4. Ο BCoordinator στέλνει την εντολή αρχικοποίησης σε κάθε BClient μαζί με τα ορίσματα εκτέλεσης.
5. Μόλις ολοκληρωθεί η φάση προετοιμασίας σε όλους τους BClient, περιμένουν την κύρια φάση εκτέλεσης, η οποία ξεκινά μέσω της αντίστοιχης ειδοποίησης BCoordinator. Μετά την ολοκλήρωση, τα αποτελέσματα επιστρέφονται στο στοιχείο BCoordinator προκειμένου να αποθηκευτούν τα αποτελέσματα.
6. Μετά την ανάκτηση των αποτελεσμάτων, ο BCoordinator ειδοποιεί τους BClients να προχωρήσουν στο στάδιο οριστικοποίησης/εκκαθάρισης για αυτήν την εκτέλεση.

7. Εάν υπάρχουν περισσότεροι συνδυασμοί που πρέπει να εκτελεστούν, η διαδικασία επιστρέφει στο Βήμα 4.

Ο BCoordinator υλοποιήθηκε με τη χρήση μοντέλων πολυνημάτωσης (multithreading). Για κάθε BClient ξεκινά ένα ξεχωριστό νήμα(thread) για αίτημα σύνδεσης συμβάντων από ένα BClient. Αυτό το νήμα αναλαμβάνει τη διαχείριση και την επικοινωνία με τον αντίστοιχο BClient σε όλο τον συνολικό κύκλο ζωής του πειράματος. Ο συγχρονισμός με τα άλλα νήματα του BClient πραγματοποιείται μέσω της δυνατότητας CountdownLatch της Java. Με αυτόν τον τρόπο, τυχόν τυχαίες καθυστερήσεις που ενδέχεται να εμφανιστούν κατά την εκτέλεση δεν εμποδίζουν την πραγματική ταυτόχρονη εκτέλεση των Benchmarks.

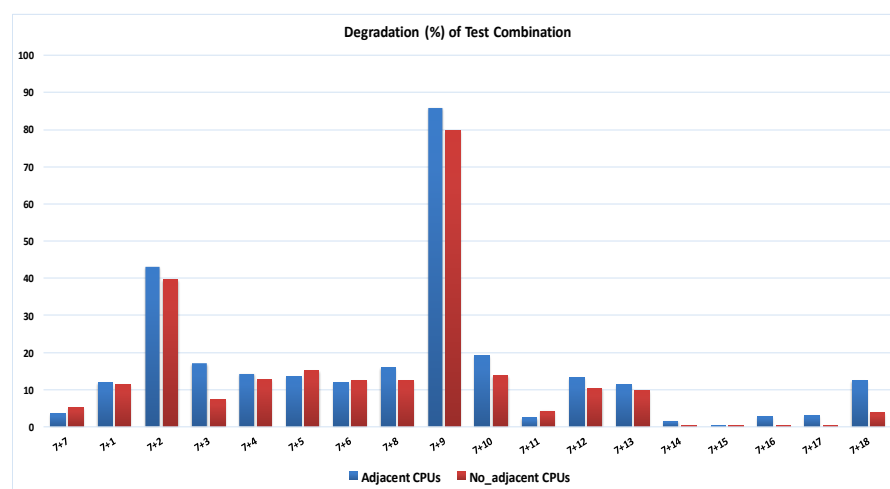
Αξιολόγηση μοντέλων παρεμβολών

Η υποβάθμιση του QoS υπολογίζεται μέσω της παρακάτω φόρμουλας. Σε αυτήν την περίπτωση, τα A και B είναι τύποι δοκιμής (Benchmarks), I είναι ο τρόπος εκτέλεσης (γειτονικοί ή μη γειτονικοί πυρήνες) και $BASE$ είναι η βαθμολογία κάθε δοκιμής όταν εκτελείται αυτόνομα.

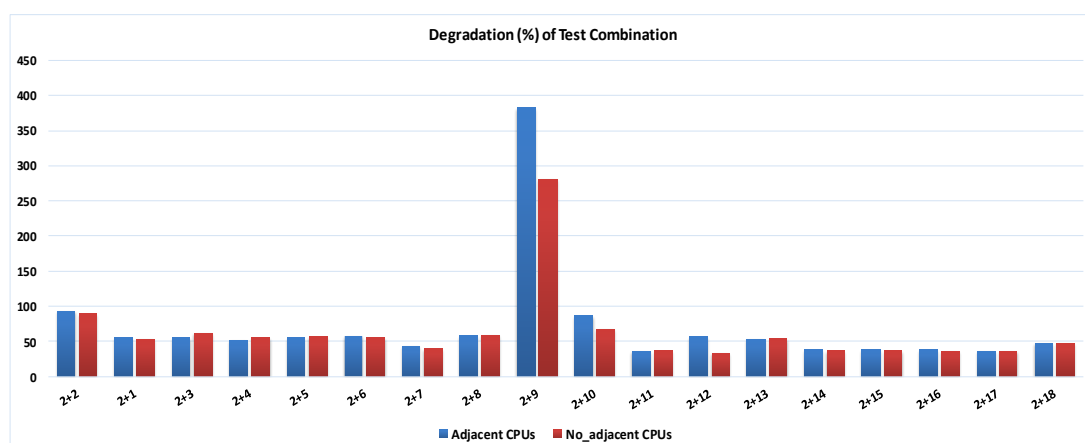
$$Degradation = 100 * \left(\frac{T_{AI} - T_{ABASE}}{T_{ABASE}} + \frac{T_{BI} - T_{BBASE}}{T_{BBASE}} \right)$$

Ενδεικτικά ποσοστά υποβάθμισης απόδοσης συγκεκριμένων κατηγοριών Benchmark απεικονίζονται στις ακόλουθες Εικόνα 29, Εικόνα 30, Εικόνα 31 . Λόγω του γεγονότος ότι ο αριθμός των φόρτων εργασίας είναι μεγάλος (18) δεν είναι εφικτό να τα συνδυάσουμε όλους τους συνδυασμούς Benchmarks σε ένα διάγραμμα. Από αυτά τα ενδεικτικά αποτελέσματα, μπορεί να προσδιοριστεί ότι η παρατηρούμενη υποβάθμιση απόδοσης μπορεί να είναι από 0 έως 360%, ανάλογα με τον τύπο των συγκριτικών Benchmarks που εκτελούνται από κοινού. Τα ποσοστά υποβάθμισης που

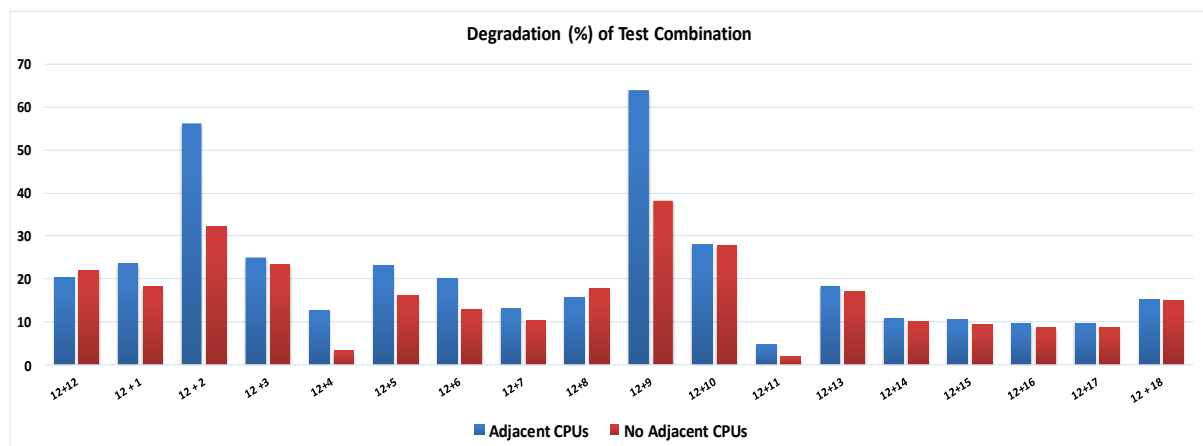
παρατηρήθηκαν δείχνουν την σημαντικότητα του να αξιολογείτε η παραλληλοποίησης των VM για την καλύτερη απόδοση και κατ' επέκταση ποιότητας υπηρεσίας για τους IaaS . Για παράδειγμα, και για τα αξιολογημένα Benchmarks είναι προφανές ότι ο χειρότερος γείτονας θα ήταν το Benchmark fileserver, ενώ για τις περιπτώσεις tomcat και web-proxy οι παρεμβολές είναι επίσης σημαντικές όταν εκτελούνται ταυτόχρονα με το Benchmark eclipse.



Εικόνα 29: Ποσοστιαία υποβάθμιση απόδοσης του DaCaro tomcat σε συνδυασμό όλων των Benchmarks[4].



Εικόνα 30: Ποσοστιαία υποβάθμιση απόδοσης του DaCaro Eclipse σε συνδυασμό όλων των Benchmarks[4].



Εικόνα 31: Ποσοστιαία υποβάθμιση απόδοσης του DaCaro web-proxy σε συνδυασμό όλων των Benchmarks[4].

Από την προαναφερθείσα διαδικασία, πραγματοποιήθηκαν 342 διαφορετικές εκτελέσεις (Πίνακας 11 διαμόρφωση πλατφόρμας) που χρησιμοποιούνται ως δεδομένα εισόδου για τον μηχανισμό πρόβλεψης παρεμβολών. Από αυτά, το 70% χρησιμοποιείται για την εκπαίδευση των μοντέλων, ενώ το 30% (103 περιπτώσεις) χρησιμοποιείται για την επικύρωση του τελικού μοντέλου. Όλες οι τιμές κανονικοποιούνται στο διάστημα $[-1, 1]$ για χρήση εντός των μοντέλων. Η διαδικασία δημιουργίας του κύριου μοντέλου ακολουθεί την υλοποίηση του [75] και βελτιστοποιεί τη δημιουργία ενός ANN που βασίζεται σε έναν GA που στοχεύει στην επιλογή βασικών παραμέτρων του δικτύου, όπως ο αριθμός των επιπέδων και οι τύποι συναρτήσεων μεταφοράς. Το Matlab χρησιμοποιείται για το σενάριο υλοποίησης, με 13 υποψήφιες συναρτήσεις μεταφοράς και μέγιστο αριθμό επιπέδων 10 και 30 νευρώνες ανά επίπεδο. Δεδομένου ότι δεν υπάρχει συγκεκριμένος κανόνας για την επιλογή του αριθμού των κρυφών στρωμάτων και νευρώνων ανά στρώμα, θέλαμε να παρέχουμε στον GA έναν (γενικά) ευρύ χώρο αναζήτησης αποφεύγοντας ωστόσο υπερβολές που θα προκαλούσαν πολύ μεγαλύτερους χρόνους εκπαίδευσης χωρίς μεγάλη βελτίωση στην απόδοση (σύμφωνα με την εμπειρία

μας με αυτό το σύνολο δεδομένων). Πειράματα με παρόμοια σύνολα δεδομένων έδειξαν ότι ο γενετικός αλγόριθμος δεν συγκλίνει ποτέ σε ANN (λύσεις) με περισσότερα κρυφά στρώματα ή περισσότερους νευρώνες ανά στρώμα από 10 και 30 αντίστοιχα. Οι είσοδοι του μοντέλου περιλαμβάνουν τους τύπους των Benchmarks που εκτελούνται, καθώς και τον τρόπο εκτέλεσης και οι έξοδοι υποδεικνύουν το ποσοστό υποβάθμισης απόδοσης κατά τους χρόνους εκτέλεσης των Benchmarks. Πραγματοποιήθηκαν δύο διαφορετικές εκτελέσεις GA, μία με 20 γενιές και 20 μέλη πληθυσμού και μία με 30 γενιές και 40 μέλη πληθυσμού. Το καλύτερο μοντέλο από κάθε διεργασία, μαζί με την τελική παραμετροποίηση από το GA και το μέσο απόλυτο ποσοστό του σφάλματος MAPE (στο ανεξάρτητο σύνολο δεδομένων επικύρωσης), απεικονίζεται στον Πίνακα 10.

<i>ANN</i>			
Αριθμός στρωμάτων/γενεών	Νευρώνες ανά στρώμα	Συναρτήσεις μεταφοράς ανά στρώμα	MAPE (%)
3/20	3-19-1	Purelin-tansig-tansig	15,8801
4/30	3-26-22-1	Satlin-tribas-purelin- purelin	18,6304
<i>Multivariate Regression</i>			
MV Regress	-0,0442 -0,0628 -0,0130 -0,6960		53,68

Πίνακας 10: Διαμόρφωση μοντέλων παρεμβολών.

Ενώ οι γενιές αυξάνονται, δεν υπάρχει καμία εγγύηση ότι το δίκτυο που προκύπτει θα βελτιωθεί εν μέρει, λόγω της εγγενούς τυχαιότητας στη διαδικασία GA. Δεδομένου επίσης του γεγονότος ότι η αύξηση των γενεών αυξάνει επίσης τον χρόνο εκπαίδευσης, μια τιμή περίπου 20 γενεών είναι επαρκής για αυτήν την περίπτωση. Η συνολική φάση

εκπαίδευσης και βελτιστοποίησης διαρκεί περίπου 1,5-2 ώρες. Το σύνολο δεδομένων εκπαίδευσης (70% του συνολικού συνόλου δεδομένων), χωρίζεται περαιτέρω σε δύο υποσύνολα, το 50% χρησιμοποιείται για τη βασική εκπαίδευση και το 20% ως επικύρωση για την ενδιάμεση εξελικτική διαδικασία.

Host OS	Ubuntu 14.04
Guest OS	Centos 6.5
Hypervisor	VMware
Benchmarks	<ol style="list-style-type: none"> 1. DaCapo Suite 2. Filebench 3. YCSB <p>(συνολικά 18 εκδόσεις με διαφορετικούς φόρτους εργασίας)</p>
Execution Mode	<ol style="list-style-type: none"> 1. Γειτονικοί πυρήνες (extended cache sharing) 2. Μη-γειτονικοί πυρήνες (minimal cache sharing)

Πίνακας 11: Χαρακτηριστικά πλατφόρμας υλοποίησης μοντέλων παρεμβολών.

Για να συγκριθεί η προσέγγιση, χρησιμοποιήθηκε ένα πολυμεταβλητό γραμμικό μοντέλο παλινδρόμησης (μέσω της συνάρτησης `mnregress` του Matlab), στο ίδιο κανονικοποιημένο σύνολο δεδομένων. Για την επικύρωση των αποτελεσμάτων χρησιμοποιήθηκε το ίδιο 30% του συνολικού συνόλου δεδομένων (δεν περιλαμβάνεται στην εκπαίδευση). Τα αποτελέσματα της διαδικασίας παλινδρόμησης ήταν σημαντικά

χειρότερα, σε σύγκριση με τη συνδυασμένη νευρογενετική προσέγγιση. Τα σφάλματα και στις δύο περιπτώσεις φαίνονται μεγάλα, ωστόσο, αυτό μπορεί επίσης να αποδοθεί στην πολυπλοκότητα του δεδομένου προβλήματος καθώς και στη δυσκολία των μοντέλων να χρησιμοποιούν αριθμητικές κατηγοριοποιήσεις για την αντιστοίχιση τύπων Benchmark και φόρτου εργασίας που μπορεί να προκαλέσουν σύγχυση στο μοντέλο. Ωστόσο, το τελευταίο βήμα είναι απαραίτητο προκειμένου να δημιουργηθεί ένα συνολικό μοντέλο που θα λαμβάνει υπόψη όλα τα σενάρια εκτέλεσης.

5.2.2 Αξιολόγηση αποδοτικότητας και ακρίβειας μοντέλων κατανομής πόρων

Ένας σημαντικός δείκτης απόδοσης για οποιαδήποτε εφαρμογή με την οποία αλληλοεπιδρούν οι χρήστες σε πραγματικό χρόνο, είναι ο χρόνος επεξεργασίας των ενεργειών του χρήστη, δηλαδή ο χρόνος που το σύστημα δεν επιτρέπει περαιτέρω αλληλεπιδράσεις έως ότου ολοκληρωθεί επεξεργασία της προηγούμενης ενέργειας. Στην περίπτωσή μας, ο χρήστης δεν αλληλεπιδρά απευθείας με τα μεμονωμένα microservices, αλλά με το UI, το οποίο με τη σειρά του δημιουργεί αιτήματα στα βασικά microservices. Έτσι, ορίσαμε ως KPI τον μέσο χρόνο επεξεργασίας κάθε μεμονωμένου βασικού microservice: Χρήστες, Προϊόντα, Πελάτες και Παραγγελίες.

Οι διαφορετικοί φόρτοι εργασίας και οι διαμορφώσεις υλικού του συστήματος είναι αυτές που επηρεάζουν κυρίως τον χρόνο επεξεργασίας κάθε microservice. Για να μπορούμε να προβλέψουμε τον χρόνο επεξεργασίας με βάση δεδομένη κίνηση και υλικό, χρησιμοποιήσαμε τα μοντέλα αντιστοίχισης. Για τα ακόλουθα πειράματα, αναπτύξαμε όλα τα microservices ως Docker κοντέινερ σε έναν μοναδικό αποκλειστικό διακομιστή.

Δημιουργία συνόλου δεδομένων

Η δημιουργία ενός συνόλου δεδομένων για τα μοντέλα κατανομής πόρων απαιτούσε τη δημιουργία αρκετών παραδειγμάτων σεναρίων πραγματικών χρηστών. Για το σκοπό αυτό, συνθέσαμε μια προσομοίωση ενός σεναρίου κοινών χρηστών στην εφαρμογή CRM (κύκλο ζωής σεναρίου):

1. Ο χρήστης συνδέεται
2. Ο χρήστης εξυπηρετεί M πελάτες
3. Κάθε πελάτης εκτελεί N παραγγελίες
4. Κάθε παραγγελία αποθηκεύεται/ενημερώνεται K φορές
5. Ο χρήστης αποσυνδέεται

Παράμετροι	Τιμές
Διάρκεια πειράματος	30, 60, 120
Κατανεμημένοι εικονικοί χρήστες	1, 10, 30, 50, 60, 90, 100, 120, 150, 200, 300
Πελάτες (M)	2, 4
Παραγγελίες ανά πελάτη (N)	2, 4
Ενημερώσεις ανά παραγγελία (K)	7

Πίνακας 12: Παράμετροι σεναρίων χρήσης εφαρμογής CRM.

Για να μπορέσει να δημιουργηθεί η απαραίτητη κίνηση για την συλλογή των μετρικών απόδοσης δημιουργήθηκε μια εφαρμογή testbed που υλοποιήθηκε με την χρήση bash scripts, GO και NodeJS. Με την χρήση αυτής της εφαρμογής μπορούν να τρέχουν τα

διάφορα σενάρια (Πίνακας 12) της εφαρμογής CRM και παράλληλα να συλλέγονται τα στατιστικά δεδομένα χρήσης πόρων (χρήση RAM, χρόνος ολοκλήρωσης αιτήματος, κα.)

	Παρτίδα Παραμετροποίησης (Execution Batch)			
Υπηρεσία	1	2	3	4
Παραγγελίες	700	896	1384	1792
Πελάτες	700	896	1384	1792
Προϊόντα	1024	1596	2048	3096
Χρήστες	700	896	1384	1792

Πίνακας 13: Μνήμη RAM(MB) των κοντέινερ (Docker container) ανά υπηρεσία/batch.

Τα σενάρια εκτελέστηκαν δημιουργώντας διαφορετικό αριθμό ταυτόχρονων χρηστών κάθε φορά. Ο Πίνακας 12 δείχνει τις παραλλαγές παραμέτρων στη διαμόρφωση των σεναρίων. Όλοι οι συνδυασμοί αυτών των τιμών (132) διαμόρφωσαν μια παρτίδα που εκτελούνταν κάτω από διαφορετική ρύθμιση υλικού κάθε φορά. Για να επιτύχουμε παραλλαγή υλικού στα πειράματα, τροποποιήσαμε τη συνολική μνήμη που έχει εκχωρηθεί σε κάθε Docker container, δημιουργώντας τέσσερις διαφορετικές διαμορφώσεις για κάθε παρτίδα.

Έτσι, το σύνολο δεδομένων για τα μοντέλα κατανομής δημιουργήθηκε εκτελώντας τις παρτίδες κύκλου ζωής (132) κάτω από τα 4 διαφορετικά προφίλ υλικού, παρέχοντας 396 παραδείγματα για κάθε τύπο υπηρεσίας. Ο Πίνακας 14 δείχνει ορισμένες ενδεικτικές τιμές για τα διάφορα microservices, όπου:

- **RAM:** Η συνολική μνήμη που εκχωρείται για το αντίστοιχο Docker container.

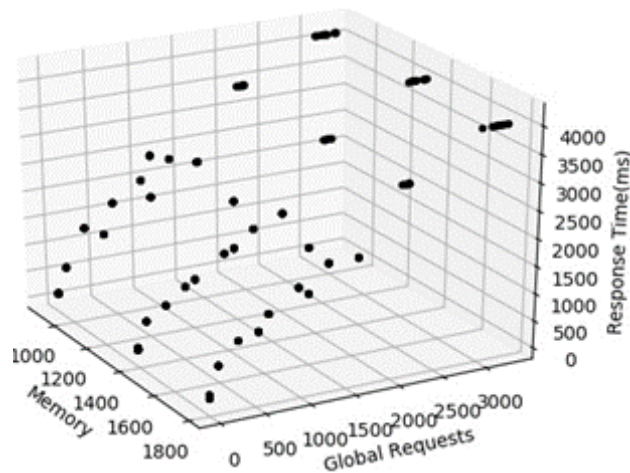
- **#Χρήστες**: Συνολικός αριθμός εικονικών χρηστών που δημιουργήθηκαν κατά την εκτέλεση του σεναρίου.
- **#Request**): Ο αριθμός των αιτημάτων που εξυπηρετούνται από το στοιχείο.
- **#RPS**(αιτήματα ανά δευτερόλεπτο): Αιτήματα που δημιουργούνται ανά δευτερόλεπτο.
- Μέσος όρος επεξεργασίας(**Avg processing**): Μέσος χρόνος επεξεργασίας κάθε microservice.

Microservices	RAM(MB)	#Χρήστες	#Αιτήματα	RPS	Avg processing (msec)
Παραγγελίες	700	10	70	3.04	23.59
Πελάτες	700	10	70	3.04	23.59
Προϊόντα	1024	10	20	0.87	104.70
Χρήστες	700	10	20	0.87	55.30
Παραγγελίες	1792	10	70	2.92	23.03
Πελάτες	1792	10	10	0.42	66.3
Προϊόντα	3096	10	20	0.83	80.10
Χρήστες	1792	10	20	0.83	37.45

Πίνακας 14: Σύνολο δεδομένων που παράχθηκε από το testbed.

Στην φάση της εκτέλεσης, χρησιμοποιήσαμε την απλούστερη δυνατή διαμόρφωση, δηλαδή μια μεμονωμένη πραγμάτωση κάθε εφαρμογής microservice. Η απόδοση του

microservice παρακολουθείται καθ' όλη τη διάρκεια του χρόνου εκτέλεσης του σεναρίου και παρέχεται ένας μέσος χρόνος επεξεργασίας ως ο τελικός γνώμονας μέτρησης της ποιότητας υπηρεσίας.



Εικόνα 32: Τρισδιάστατη απεικόνιση των παραμέτρων του microservice των πελατών[5].

Όπως παρατηρείται στα αρχικά αποτελέσματα του συνόλου δεδομένων, τα περισσότερα αιτήματα κατά τη διάρκεια του κύκλου ζωής δημιουργούνται στο microservice παραγγελιών, ενώ η αύξηση της μνήμης RAM έχει θετική επίδραση, μόνο σε προϊόντα και χρήστες. Η Εικόνα 32 απεικονίζει τις κύριες παραμέτρους ενός συνόλου δεδομένων microservices στον τρισδιάστατο χώρο.

Αξιολόγηση μοντέλων παρεμβολών

Για να επικυρώσουμε την ακρίβεια της προσέγγισης των μοντέλων χαρτογράφησης, τα συγκρίναμε με τρεις βασικές και αποτελεσματικές τεχνικές Machine Learning (ML), συγκεκριμένα με Support Vector Machine (SVMs), Random Forest (RFs) και Linear Regression (γραμμική παλινδρόμηση) (LR). Προκειμένου να επιτευχθεί το επιθυμητό επίπεδο γενίκευσης και απόδοσης πρόβλεψης, τα προαναφερθέντα μοντέλα

συντονίστηκαν χρησιμοποιώντας εξαντλητική αναζήτηση πλέγματος (grid search) σε ένα ευρύ φάσμα τιμών υπερπαραμέτρων, σε συνδυασμό με διασταυρούμενη επικύρωση k-fold ως τεχνική επικύρωσης μοντέλου. Συγκεκριμένα, για τα SVM χρησιμοποιήσαμε τον πυρήνα RBF, ρυθμίζοντας τις παραμέτρους C και gamma. Στο Random Forests ρυθμίσαμε τον αριθμό των εκτιμητών, το μέγιστο βάθος του δέντρου και τον ελάχιστο αριθμό δειγμάτων που απαιτούνται για να βρίσκονται σε έναν κόμβο φύλλου. Η Γραμμική Παλινδρόμηση, ως ένα απλό μοντέλο, αποδίδει κακώς στο δεδομένο σύνολο δεδομένων και τα αποτελέσματά της δεν θα παρουσιαστούν. Ο κώδικας για αυτά τα πειράματα γράφτηκε σε Python χρησιμοποιώντας τη βιβλιοθήκη scikit-learn.

Microservice	ANN	Random Forest	SVM
Παραγγελίες	9.6%	18.17%	20.83%
Πελάτες	6.22%	10.51%	9.33%
Προϊόντα	13.39%	13.52%	12.82%
Χρήστες	8.37%	9.38%	9.38%

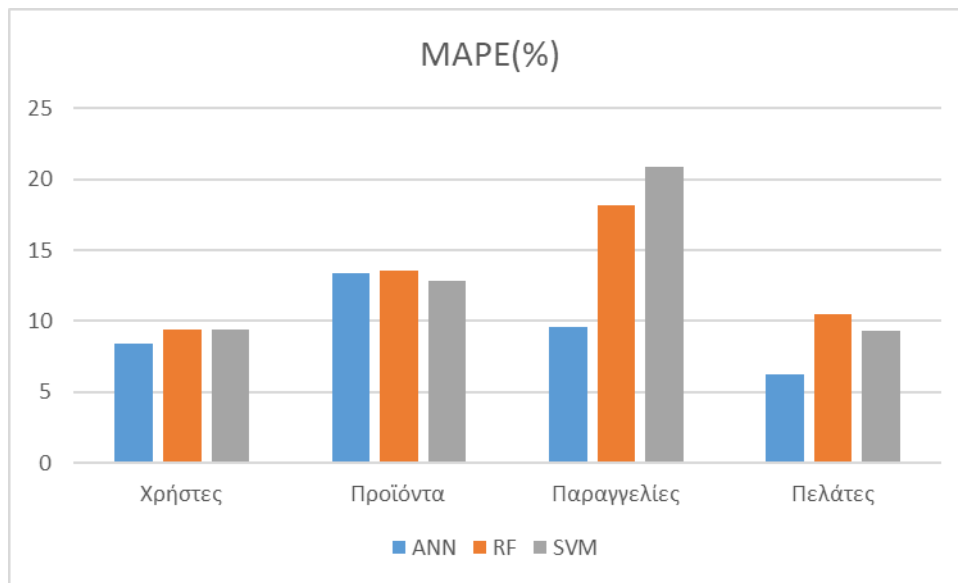
Πίνακας 15: Αναλυτική συγκριτική αξιολόγησης MAPE.

Οι παράμετροι σχεδίασης κάθε μοντέλου ANN που διαμορφώθηκαν, μετά από 30 γενιές, από τον Γενετικό Αλγόριθμο (GA) με 3 υποψήφιες συναρτήσεις μεταφοράς και μέγιστο 10 στρώματα και 30 νευρώνες ανά στρώμα απεικονίζονται στον παρακάτω Πίνακας 16.

Παραμετροποίηση του δικτύου ANN			
Υπηρεσία	Αριθμός στρωμάτων/γενεών	Νευρώνες ανά στρώμα	Συναρτήσεις μεταφοράς ανά στρώμα
Παραγγελίες	4	5-3-2-1	tansig-logsig-logsig- purelin
Πελάτες	3	5-3-1	tansig-logsig-purelin
Προϊόντα	3	5-3-1	tansig-logsig-purelin
Χρήστες	3	5-3-1	tansig-tansig-purelin

Πίνακας 16: Παράμετροι του νευρωνικού δικτύου όπου επιλέχτηκαν από τον GA.

Για να παραμείνουν απλά τα μοντέλα, κατασκευάστηκε και εκπαιδεύτηκε ένα διαφορετικό μοντέλο για κάθε διαφορετική μικρουπηρεσία της εφαρμογής CRM (δεδομένου ότι είναι ανεξάρτητες) με αποτέλεσμα καλύτερη απόδοση και ολοκληρωμένη εικόνα της συμπεριφοράς κάθε ξεχωριστού στοιχείου υπηρεσίας. Η μέτρηση επικύρωσης για αυτά τα πειράματα ήταν το μέσο απόλυτο ποσοστό σφάλματος (MAPE) κάθε μοντέλου για το ίδιο σύνολο δοκιμών.



Εικόνα 33: Ραβδόγραμμα συγκριτικής απόδοσης ML αλγορίθμων.

Μια επισκόπηση των MAPE που προκύπτουν απεικονίζεται στην Εικόνα 33, με τους αριθμούς για κάθε Υπηρεσία ξεχωριστά να παρέχονται στον Πίνακα 15. Όπως μπορεί να παρατηρηθεί στα παρεχόμενα αποτελέσματα, το ANN δίκτυο, βελτιστοποιημένο από τον GA, ξεπερνά σαφώς τον ανταγωνισμό για όλες τις υπηρεσίες, ωστόσο είναι λιγότερο αποτελεσματικές κατά τη φάση της εκπαίδευσης, λόγω του βήματος βελτιστοποίησης που εισήχθη με τη χρήση του GA.

6

Συμπεράσματα και μελλοντικές ερευνητικές κατευθύνσεις

Ο στόχος της παρούσας διατριβής ήταν η σχεδίαση αλλά και η ανάπτυξη εργαλείων λογισμικού για την αξιολόγηση αλλά και βελτίωση υπηρεσιών Cloud. Αυτά τα εργαλεία λογισμικού είναι αδύνατο να καλύψουν στην πληρότητα τους όλες τις ανάγκες για αξιολόγηση αλλά και διαχείριση των πόρων Cloud. Ωστόσο, όπως αναφέρθηκε και στο εισαγωγικό κεφάλαιο 1, υπάρχουν συγκεκριμένες ερευνητικές κατευθύνσεις για την σχεδίαση και ανάπτυξη των εργαλείων αυτών, που αφορούσαν τα κάτωθι στοιχεία ερευνάς στο Cloud:

- Την πρόβλεψη της απόδοσης των εφαρμογών στο περιβάλλον Cloud
- Την πρόβλεψη και μείωσης του κόστους λειτουργίας και εγκατάστασης του λογισμικού στο Cloud
- Την βελτίωση της αξιολόγησης και επιλογής παρόχου Cloud

Η ανάπτυξη του κάθε εργαλείου λογισμικού είχε ως στόχο να δημιουργήσει λύσεις στους προαναφερθέντες ερευνητικούς τομείς.

6.1 Πρόβλεψη απόδοσης εφαρμογών Cloud

Η πρόβλεψη της απόδοσης των εφαρμογών έχει σαν απώτερο σκοπό την δημιουργία γνώσης πάνω στις ανάγκες σε πόρους μιας εφαρμογής για να πετύχει ένα συγκεκριμένο QoS. Αυτό σημαίνει ότι ο IaaS πρέπει να έχει γνώση για το πώς μια εφαρμογή συμπεριφέρεται και αποδίδει όταν λειτουργεί με διαφορετικών δυνατοτήτων εικονικούς πόρους. Με αυτόν τον τρόπο ένας IaaS μπορεί να βελτιώσει την τοποθέτηση των εφαρμογών στους εικονικούς πόρους διατηρώντας το προκαθορισμένο QoS.

Για τους σκοπούς της συγκεκριμένης έρευνας αναπτύχθηκαν τα μοντέλα κατανομής πόρων τα οποία με την χρήση ANN εκπαιδευμένα από έναν GA είχαν ως σκοπό να εξάγουν μοτίβα συσχέτισεως μεταξύ των εικονικών πόρων CPU, RAM και μετρικών QoS. Με απώτερο σκοπό μέσα από την εκπαίδευση του ANN να μπορεί να προβλεφθεί το πόσους πόρους χρειάζεται μια εφαρμογή για να διατηρήσει ένα συγκεκριμένο QoS δοθείσης ενός συγκεκριμένου φόρτου εργασίας(αριθμό χρηστών). Για τους σκοπούς του πειραματισμού και επαλήθευσης των μοντέλων χρησιμοποιήθηκε μια πραγματική εφαρμογή CRM/ERP, η οποία αποτελούνταν από ένα σύνολο από καλά ορισμένες μικρό-υπηρεσίες. Τα αποτελέσματα επαλήθευσης ήταν πολύ ικανοποιητικά και ο επιλεγθής αλγόριθμος AI έδειχνε να αποδίδει καλύτερα από άλλες γνωστές λύσεις. Ωστόσο ο χρόνος για την παραμετροποίηση των δεδομένων απόδοσης εικονικών πόρων η επιλογή των μετρικών QoS καθώς και η εκπαίδευση του ANN αποτελούν πολύ χρονοβόρες διαδικασίες που για μελέτη περίπτωσης και διεξαγωγή έρευνας είναι εφικτές. Η πλήρης τυποποίηση της διαδικασίας για αυτήν την αξιολόγηση παρόλο που προσφέρει μια λύση στο πρόβλημα απαιτεί αρκετό χρόνο για να την καταστήσει εύκολα εφαρμόσιμη στη τεχνολογική βιομηχανία στο άμεσο μέλλον.

6.2 Πρόβλεψη και μείωσης του κόστους λειτουργίας Cloud

Η μείωση του κόστους λειτουργίας αποτελεί ένα πολυδιάστατο κομμάτι της λειτουργίας του Cloud. Υπάρχουν πολλές τεχνικές και επιχειρησιακές προσεγγίσεις που μπορούν να συντελέσουν στην διαχείριση και μείωση του κόστους. Στην παρούσα διατριβή η προσέγγιση που χρησιμοποιήθηκε ήταν η ανάπτυξη μοντέλων βασισμένα σε ANN που έχουν ως σκοπό την μελέτη των παρεμβολών που δημιουργούνται μεταξύ των εικονικών πόρων που εκτελούν παράλληλα διεργασίες διαφορετικών εφαρμογών. Ο απώτερος σκοπός αυτής της προσέγγισης είναι να μπορέσουν οι πάροχοι Cloud να ενορχηστρώσουν τις διάφορες εφαρμογές μειώνοντας τους αναγκαίους υπολογιστικούς πόρους που χρειάζονται μέσα από την βελτιωμένη απόδοση λόγω ελαχιστοποίησης των παρεμβολών.

Η μείωση της απόδοσης των εφαρμογών που δύναται να προκαλέσουν αυτές οι παρεμβολές μπορούν να αποφευχθούν αν γνωρίζουμε εκ των προτέρων ποιοι τύποι εφαρμογών προκαλούν τις μεγαλύτερες παρεμβολές απόδοσης όταν τρέχουν παράλληλα στους ιδίους εικονικούς πόρους με άλλες εφαρμογές. Για τον λόγο αυτό εκπαιδεύτηκαν τα μοντέλα παρεμβολών. Σε αντίθεση με τα μοντέλα κατανομής πόρων τα συγκεκριμένα μοντέλα για να παράξουν το σύνολο των δεδομένων δεν χρησιμοποιούν πραγματικές εφαρμογές, αλλά Benchmarks. Τα Benchmarks αυτά εκτελούνταν παράλληλα σε διαφορετικούς εικονικούς πόρους και μελετάτε η τυχόν μειώσεις απόδοσης στις μετρικές που εξάγουν. Τα μοντέλα που παρήχθησαν φανέρωσαν ότι κάποιοι συνδυασμοί Benchmark δύναται να παράξουν μεγάλες παρεμβολές και μείωση απόδοσης, γεγονός που δίνει αξία στην μελέτη που διεξήχθη. Ωστόσο η μελέτη του κατά πόσο η τοπολογία των φυσικών πυρήνων που χρησιμοποιούνται από τους εικονικούς πόρους δεν έδειξε σημαντικές διαφοροποιήσεις.

Όπως σε κάθε εκπαίδευση και επαλήθευση αλγορίθμων ML το σύνολο δεδομένων που χρειάζεται είναι μεγάλο ωστόσο η χρήση το benchmarks έκανε για την συγκεκριμένη μελέτη αρκετά πιο εύκολή την διαδικασία συλλογής δεδομένων. Όμως δοθείσης της ετερογένειας των Benchmarks τόσο στις διεργασίες που εκτελούν τόσο και στις δομές δεδομένων που χρησιμοποιούν για την εξαγωγή και παρουσίαση των μετρικών απόδοσης είναι αδύνατο να εισαχθεί αυτόματα ένα καινούριο Benchmark χωρίς πρώτα να γίνουν οι απαραίτητες ενέργειες για ομογενοποίηση του τρόπου εκτέλεσης καθώς και των μετρικών που εξάγει ώστε να μπορούν να χρησιμοποιηθούν στην περαιτέρω εκπαίδευση των μοντέλων.

6.3 Βελτίωση της αξιολόγησης και επιλογής παρόχου Cloud

Ένα από τα σημαντικότερα θέματα όπου διερευνήθηκαν στα πλαίσια της συγκεκριμένης διατριβής είναι η αξιολόγηση και επιλογή υπηρεσιών Cloud. Δοθείσης της πληθώρας από λύσεις υποδομών που προσφέρονται είτε σε ιδιωτικά είτε σε «δημόσια» Cloud συστήματα, είναι σημαντικό οι χρήστες αλλά και οι μελλοντικοί υιοθέτες να μπορούν να επιλέξουν ενημερωμένα την καλύτερη δυνατή λύση για το συγκεκριμένο λογισμικό που επρόκειτο να εγκατασταθεί και να λειτουργήσει στο Cloud. Η αξιολόγηση του Cloud καθώς και της βελτίωσης αυτής αποτελεί έναν ερευνητικό τομέα που υπάρχουν πολλές λύσεις οι οποίες πολλές φορές προέρχονται από τους ίδιους του παρόχους των υπηρεσιών αυτών. Παρόλα αυτά, οι λύσεις αυτές εστιάζουν κατά κύριο λόγο στην παρακολούθηση της απόδοσης αλλά και ενορχήστρωσης των πόρων. Η ερευνητική δουλειά και το λογισμικό που αναπτύχθηκε στα πλαίσια της διατριβής έχει σκοπό την προσέγγιση της αξιολόγησης όχι μόνο από την πλευρά της απόδοσης, αλλά και από την πλευρά της σταθερότητας και της αντιληφθείσας ποιότητας εμπειρίας των υπηρεσιών Cloud.

Συγκεκριμένα, η αξιολόγηση των συμβολαίων επιπέδου ποιότητας υπηρεσίας SLA καθώς και η τυποποίηση της αξιολόγησης απόδοσης μέσα από Benchmarks ήταν οι κύριοι στόχοι των εργαλείων λογισμικού που αναπτύχθηκαν.

Όσον αφορά την αξιολόγηση των SLA αναπτύχθηκε το κατάλληλο λογισμικό το οποίο είχε σκοπό να τυποποιήσει τη διαδικασία συλλογής δεδομένων αλλά και ελέγχου των SLA χρησιμοποιώντας συγκεκριμένα στάνταρντ[70]. Το SLA αποτελεί ένα έγγραφο το οποίο δεσμεύει τον πάροχο της υπηρεσίας με συγκεκριμένα εχέγγυα σε ότι αφορά την ποιότητα υπηρεσίας. Αυτό το έγγραφο για να μπορέσει να μετασχηματιστεί και να περιγραφεί χρησιμοποιώντας μια δομή δεδομένων χρειάζεται πρώτα να αποσαφηνιστούν όλα τα κύρια δομικά κομμάτια που το συντελούν. Όπως ο τρόπος συλλογής των δεδομένων, οι μετρικές που εγγυάται ο πάροχος της υπηρεσίας, κλπ.(αναλυτικότερα κεφάλαιο 4.1). Έχοντας αποσαφηνίσει αυτά τα στοιχεία είναι σημαντικό να υλοποιηθεί και το σύστημα το οποίο θα μπορεί να αλληλοεπιδρά με όλα τα ετερογενή συστήματα Cloud και να κάνει την αξιολόγηση με βάση τις μετρικές και τους περιορισμούς του εκάστοτε παρόχου.

Ό,τι αφορά την αξιολόγηση της απόδοσης των Cloud υπηρεσιών η χρήση των Benchmarks δεν αποτελεί μια καινούρια προσέγγιση, η έρευνα εστίασε κύριος στην τυποποίηση και εξέλιξη του τρόπου συλλογής των δεδομένων απόδοσης από τα Benchmarks και την εφαρμογή τυποποιημένων μετρικών και σοδιάζουν την αξιολόγηση της απόδοσης και της σταθερότητας αυτής (Performance of Virtual Cores κεφάλαιο 4.2.1).

Τα πειράματα που διεξήχθησαν για την αξιολόγηση των εργαλείων αυτών είχαν σκοπό να κρίνουν το κατά πόσο μπορούν αυτά τα συστήματα να εξυπηρετήσουν πολλαπλά αιτήματα από πολλαπλούς χρήστες σε διαφορετικές Cloud υποδομές, χωρίς σφάλματα. Ο

λόγος διεξαγωγής αυτού του τύπου αξιολόγησης ήταν διότι ένα για να αποδώσει ένα τέτοιο σύστημα και να μπορέσει κάποιος να συμπερασματολογήσει για το κατά πόσο μια υπηρεσία είναι πιο αποδοτική ή λιγότερο κοστοβόρα από μια άλλη, χρειάζεται μεγάλο όγκο από δεδομένα αξιολόγησης που εκτείνονται σε όλο το φάσμα των υπηρεσιών Cloud. Τα αποτελέσματα της πειραματικής διαδικασίας έδειξαν σταθερότητα (μεγαλύτερη 90% κεφάλαιο 5.1.1) δοθείσης της πολυπλοκότητας και του αριθμού των μικρό-υπηρεσιών που ενορχηστρώνονται για να επιτευχθεί η λειτουργικότητα.

Μολονότι το σύστημα αξιολόγησης SLA (QoE-3ALib) μπορεί να αξιολογήσει επιτυχώς ένα SLA μιας Cloud υπηρεσίας δεν υπάρχει κανένας τυποποιημένος μηχανισμός για την αποζημίωση των πελάτων σε περίπτωση παραβίασης του. Για το λόγο αυτό η ερευνητική δουλειά μετέπειτα από την διατριβή αυτή εστίασε στην ανάπτυξη μηχανισμών δημιουργίας, αξιολόγησης και διαχείρισης των SLA υπό την αιγίδα συστημάτων Blockchain με χρήση έξυπνων συμβολαίων (smart contracts)[76].

7 *Συντομογραφίες*

Ο παρών πίνακας παρουσιάζει αλφαβητικά το λεξιλόγιο με τις συντομογραφίες της διατριβής.

ANN	Artificial Neural Network
API	Application Programming Interface
AVG	Average
AWS	Amazon Web Services
CAT	Cloud Adopter Toolkit
CCI	Common Client Interface
CCM	Cloud Capacity Manager
CPU	Central Process Unit
CRM	Customer Relationship Management
DRS	Distributed Resource Scheduler
ERP	Enterprise Resource Planning
GAE	Google App Engine
GET	Get HTTP request
GID	Group Identification
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HPC	High performance Computing
HTTP	Hypertext Transfer Protocol (world wide web protocol)
IaaS	Infrastructure as a Service
ICMP	Internet Control Message Protocol

IEEE	Institute of Electrical and Electronics Engineers
IMT	IaaS Management Toolkit
IPT	IaaS Provider Toolkit
ISO	International Organization for Standardization
KPI	Key Performance Indicator
MAPE	Mean Absolute Percentage Error
NIST	National Institute of Standards & Technology (US)
PKB	PerfKit Benchmark
PVC	Performance of Virtual Cores
QET	Quality of Experience Toolkit
QoE	Quality of Experience
QoS	Quality of Service
RAM	Random-Access Memory
REST	Representational State Transfer
SaaS	Software as a Service
SLA	Service level Agreement
SME	Small Medium Enterprise
SPEC	Standard Performance Evaluation Corporation
SPI	Software Platform Infrastructure
SSD	Solid State Drive
TCP	Transmission Control Protocol
UID	Unique Identifier
YCSB	Yahoo Cloud Serving Benchmark

8

Βιβλιογραφικές Αναφορές

- [1] Cloud Landscape Report: Price & Performance, Internap, available at:
<http://www.internap.com/resources/bmc-price-performance/>
- [2] Diving into IT Cloud Services Report, Spiceworks, available at:
<http://www.spiceworks.com/marketing/diving-into-IT-cloud-services/report/>
- [3] Kousiouris, G., Aisopos, F., Psychas, A., Varvarigou, T., Domaschka, J., Baur, D., ... & Papper, J. (2017, July). A toolkit based architecture for optimizing cloud management, performance evaluation and provider selection processes. In 2017 International Conference on High Performance Computing & Simulation (HPCS) (pp. 224-232). IEEE.
- [4] Psychas, A., Violos, J., Aisopos, F., Evangelinou, A., Kousiouris, G., Bouras, I., ... & Stavroulas, Y. (2020). Cloud toolkit for Provider assessment, optimized Application Cloudification and deployment on IaaS. *Future Generation Computer Systems*, 109, 657-667.
- [5] Bouras, I., Aisopos, F., Violos, J., Kousiouris, G., Psychas, A., Varvarigou, T. A., ... & Stavroulas, Y. (2019). Mapping of Quality of Service Requirements to Resource Demands for IaaS. In *CLOSER* (pp. 263-270).

- [6] Best Practices for monitoring cloud infrastructure you don't own, Whitepaper, Sevone.com, available at: <https://www.sevone.com/white-paper/4-best-practices-monitoring-cloud-infrastructure-you-dont-own>
- [7] M. Wang, M. Wang, X. Meng, L. Zhang, Consolidating virtual machines with dynamic bandwidth demand in data centers, in: Proceedings of IEEE INFOCOM'11, 2011, pp. 71–75, doi: 10.1109/INFCOM.2011.5935254 .ARTIST Consortium, Deliverable D7.1 Definition and extension of performance stereotypes, ICCS/NTUA and other partners," March 2014
- [8] Garg, Saurabh Kumar, Steve Versteeg, and Rajkumar Buyya. "A framework for ranking of cloud computing services." *Future Generation Computer Systems* 29.4 (2013): 1012-1023.
- [9] Fatema, Kaniz, et al. "A survey of cloud monitoring tools: Taxonomy, capabilities and objectives." *Journal of Parallel and Distributed Computing* 74.10 (2014): 2918-2933.
- [10] Alhamazani, Khalid, et al. "An overview of the commercial cloud monitoring tools: research dimensions, design issues, and state-of-the-art." *Computing* 97.4 (2015): 357-377.
- [11] Ghosh, N.; Ghosh, S.K.; Das, S.K., "Selcsp: A Framework To Facilitate Selection Of Cloud Service Providers," *Cloud Computing, Ieee Transactions On* , Vol.3, No.1, Pp.66,79, Jan.-March 1 2015 Doi: 10.1109/Tcc.2014.232857
- [12] Garg, S.K.; Versteeg, S.; Buyya, R., "Smicloud: A Framework For Comparing And Ranking Cloud Services," *Utility And Cloud Computing (Ucc)*, 2011 Fourth

Ieee International Conference On , Vol., No., Pp.210,218, 5-8 Dec. 2011 Doi:
10.1109/Ucc.2011.36

- [13] Carnell, John. Spring microservices in action. Manning Publications Co., 2017.
- [14] Hammoud, Ahmad, et al. "On the detection of passive malicious providers in cloud federations." IEEE Communications Letters 23.1 (2018): 64-67.
- [15] Ikram, Mohammed Abdulaziz, and Farookh Khadeer Hussain. "Software as a Service (SaaS) service selection based on measuring the shortest distance to the consumer's preferences." International Conference on Emerging Internetworking, Data & Web Technologies. Springer, Cham, 2018.
- [16] Oo, Myat Nandar, and Nay Lynn. "Forensic Analysis of Residual Artifacts on Bitnami Hadoop." 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIconRus). IEEE, 2020.
- [17] Stephen, Absa, Shajulin Benedict, and RP Anto Kumar. "Monitoring IaaS using various cloud monitors." Cluster Computing 22.5 (2019): 12459-12471.
- [18] Aktas, Mehmet S. "Hybrid cloud computing monitoring software architecture." Concurrency and Computation: Practice and Experience 30.21 (2018): e4694.
- [19] Foster, Matt. "Netcraft Analysis: Online Speed Testing Tools." 25th February (2013).
- [20] <https://www.akamai.com/uk/en/what-we-do/intelligent-platform/cloud-monitor.jsp>
- [21] Olups, Rihards. Zabbix 1.8 network monitoring. Packt Publishing Ltd, 2010.
- [22] George Kousiouris, Tommaso Cucinotta, Theodora Varvarigou, "The Effects of Scheduling, Workload Type and Consolidation Scenarios on Virtual Machine

Performance and their Prediction through Optimized Artificial Neural Networks, The Journal of Systems and Software (2011).

- [23] Carsten Binnig, Donald Kossmann, Tim Kraska, and Simon Loesing. 2009. How is the weather tomorrow?: towards a Benchmark for the cloud. In Proceedings of the Second International Workshop on Testing Database Systems (DBTest '09). ACM, New York, NY, USA, Article 9 , 6 pages.
- [24] Nikolas Roman Herbst, Samuel Kounev, Andreas Weber, and Henning Groenda. 2015. BUNGEE: an elasticity Benchmark for self-adaptive IaaS cloud environments. In Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '15). IEEE Press, Piscataway, NJ, USA, 46-56.
- [25] Alexandru Iosup. 2013. IaaS cloud Benchmarking: approaches, challenges, and experience. In Proceedings of the 2013 international workshop on Hot topics in cloud services (HotTopiCS '13). ACM, New York, NY, USA, 1-2.
- [26] Rouven Krebs, Christof Momm, and Samuel Kounev. 2012. Metrics and techniques for quantifying performance isolation in cloud environments. In Proceedings of the 8th international ACM SIGSOFT conference on Quality of Software Architectures (QoSA '12). ACM, New York, NY, USA, 91-100.
- [27] Palit, Tapti, Yongming Shen, and Michael Ferdman. "Demystifying cloud Benchmarking." 2016 IEEE international symposium on performance analysis of systems and software (ISPASS). IEEE, 2016.

- [28] Kratzke, Nane, and Peter-Christian Quint. "About automatic Benchmarking of iaas cloud service providers for a world of container clusters." *Journal of Cloud Computing Research* 1.1 (2015): 16-34.
- [29] Calheiros, Rodrigo N., et al. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." *Software: Practice and experience* 41.1 (2011): 23-50.
- [30] Vaulx, Frederic de, Jeff Perdue CMU, Steve Woodward, Alan Sill TTU, Ken E. Stavinoha, Tom Rutt Fujitsu, Jenny Huang, Omar Fink SAIC and Steven J. McGee. "Cloud Computing Service Metrics Description" Nist Special Publication 500-307 *Cloud Computing Service Metrics Description Reports* (2014).
- [31] K. Mills, J. Filliben, and C. Dabrowski. Comparing VM-placement algorithms for on-demand clouds. In *Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science, CLOUDCOM '11*, pages 91–98, Washington, DC, USA, 2011. IEEE Computer Society.
- [32] H. Xu and B. Li. Anchor: A versatile and efficient framework for resource management in the cloud. *Parallel and Distributed Systems, IEEE Transactions on*, 24(6):1066–1076, 2013.
- [33] M. Maurer, I. Brandic, and R. Sakellariou. Enacting SLAs in clouds using rules. In *Proceedings of the 17th international conference on Parallel processing - Volume Part I, Euro-Par'11*, pages 455–466, Berlin, Heidelberg, 2011. Springer-Verlag.
- [34] J.W. Jiang, T. Lan, S. Ha, M. Chen, and M. Chiang. Joint VM placement and routing for data center traffic engineering. In *INFOCOM, 2012 Proceedings IEEE*, pages 2876–2880, 2012.

- [35] Verma, Manish, et al. "Dynamic resource demand prediction and allocation in multi-tenant service clouds." *Concurrency and Computation: Practice and Experience* (2016).
- [36] Calheiros, Rodrigo N., et al. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." *Software: Practice and experience* 41.1 (2011): 23-50.
- [37] K. Mills, J. Filliben, and C. Dabrowski. Comparing VM-placement algorithms for on-demand clouds. In *Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science, CLOUDCOM '11*, pages 91–98, Washington, DC, USA, 2011. IEEE Computer Society.
- [38] H. Xu and B. Li. Anchor: A versatile and efficient framework for resource management in the cloud. *Parallel and Distributed Systems, IEEE Transactions on*, 24(6):1066–1076, 2013.
- [39] M. Maurer, I. Brandic, and R. Sakellariou. Enacting SLAs in clouds using rules. In *Proceedings of the 17th international conference on Parallel processing - Volume Part I, Euro-Par'11*, pages 455–466, Berlin, Heidelberg, 2011. Springer-Verlag.
- [40] F. Chang, J. Ren, and R. Viswanathan. Optimal resource allocation in clouds. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 418–425, July 2010.
- [41] F. Chang, R. Viswanathan, and T. L. Wood. Placement in clouds for application-level latency requirements. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 327–335, 2012.

- [42] Mashayekhy, Lena, et al. "An online mechanism for resource allocation and pricing in clouds." *IEEE transactions on computers* 65.4 (2016): 1172-1184.
- [43] Hu, Xinhui, et al. "Competitive strategies for online cloud resource allocation with discounts: The 2-dimensional parking permit problem." *Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on*. IEEE, 2015.
- [44] Sun, Yu, et al. "ROAR: A QoS-oriented modeling framework for automated cloud resource allocation and optimization." *Journal of Systems and Software* 116 (2016): 146-161.
- [45] Konstanteli, Kleopatra, et al. "Elastic admission control for federated cloud services." *IEEE Transactions on Cloud Computing* 2.3 (2014): 348-361.
- [46] Z. Xiao, W. Song, and Q. Chen. Dynamic resource allocation using virtual machines for cloud computing environment. *Parallel and Distributed Systems, IEEE Transactions on*, 24(6):1107–1117, 2013.
- [47] Govindaraju, Yatheendraprakash, and Hector Duran-Limon. "A QoS and energy aware load balancing and resource allocation framework for IaaS cloud providers." *Proceedings of the 9th International Conference on Utility and Cloud Computing*. ACM, 2016.
- [48] Milojičić, Dejan, Ignacio M. Llorente, and Ruben S. Montero. "Opennebula: A cloud management tool." *IEEE Internet Computing* 15.2 (2011): 11-14.
- [49] Sotomayor, Borja, et al. "Capacity leasing in cloud systems using the opennebula engine." *Workshop on Cloud Computing and its Applications*. Vol. 3. 2008.
- [50] Sotomayor, Borja, et al. "Virtual infrastructure management in private and hybrid clouds." *IEEE Internet computing* 13.5 (2009).

- [51] Kesavan, Mukil, Ada Gavrilovska, and Karsten Schwan. "Elastic resource allocation in datacenters: gremlins in the management plane." *ELASTIC* 1 (2012): 8.
- [52] Keahey, Kate, et al. "Managing Allocatable Resources." 2019 IEEE 12th International Conference on Cloud Computing (CLOUD). IEEE, 2019.
- [53] Nanditha, A. "Network Aware OpenStack Nova Scheduler."
- [54] el Khaldi, Fouad, et al. "Cloud based hpc for innovative virtual prototyping methodology: Automotive applications." *Transportation Research Procedia* 14 (2016): 993-1002.
- [55] Kesavan, Mukil, Ada Gavrilovska, and Karsten Schwan. "Elastic resource allocation in datacenters: gremlins in the management plane." *ELASTIC* 1 (2012)
- [56] B. Koller, M. Gienger. "Enhancing High Performance Computing with Cloud Concepts and Technologies", in *Sustained Simulation Performance 2014*
- [57] Bergman, Keren, et al. "Exascale computing study: Technology challenges in achieving exascale systems." Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Tech. Rep 15 (2008).
- [58] D. M. Jacobsen, R. S. Canon. "Contain This, Unleashing Docker for HPC", "<https://www.nersc.gov/assets/Uploads/cug2015udi.pdf>", 2015
- [59] R. S. Canon, D. M. Jacobsen. "Shifter: Containers for HPC", https://cug.org/proceedings/cug2016_proceedings/includes/files/pap103.pdf, 2016
- [60] A. Brinkmann, P. Kling, F. Meyer, L. Nagel, S. Riechers, T. Süß. "Scheduling shared continuous resources on many-cores", *SPAA '14*, 2014

- [61] S. Pickartz, R. Gad, S. Lankes, L. Nagel, T. Süß, A. Brinkmann, S. Krempel. “Migration Techniques in HPC Environments”. In Proceedings of the 9th Workshop on Virtualization in High-Performance Cloud Computing (VHPC) held in conjunction with Euro-Par 2014, August 25-29, Porto, Portugal
- [62] J. Breitbart, S. Pickartz, J. Weidendorfer, A. Monti. “Viability of Virtual Machines in HPC”. ROME Workshop at Euro-Par, Grenoble, France, August 2016.
- [63] J. Breitbart, J. Weidendorfer. “Detailed Application Characterization and its Use for Effective Co-Scheduling”. Co-Scheduling of HPC Applications, Advances in Parallel Computing 29, IOS Press, 2017.
- [64] S. Pickartz, J. Breitbart, C. Clauss, S. Lankes, A. Monti. “Virtualization in HPC: An Enabler for Adaptive Co-scheduling?”. Co-Scheduling of HPC Applications, Advances in Parallel Computing 29, IOS Press, 2017.
- [65] A. Kohne, D. Pasternak, L. Nagel, O. Spinczyk. “Evaluation of SLA-based decision strategies for VM scheduling in Cloud data centers”. In Proceedings of the 3rd Workshop on CrossCloud Infrastructures & Platforms, CrossCloud 2016, London, April 2016.
- [66] Gentsch, Wolfgang. "Linux containers simplify engineering and scientific simulations in the cloud." 2014 Annual Global Online Conference on Information and Computer Technology. IEEE, 2014.
- [67] Youseff L, Butrico M, Da Silva D (2008) Toward a unified ontology of cloud computing. In: Grid computing environments workshop, 2008. GCE '08grid computing environments workshop,2008, GCE '08, pp 1–10P

- [68] G. Kousiouris, D. Kyriazis, S. V. Gogouvitis, A. Menychtas, K. Konstanteli and T. A. Varvarigou, "Translation of Application-level Terms to Resource-level attributes across the Cloud Stack Layers", Computers and Communications (ISCC), 2011 IEEE Symposium on , vol., no., pp.153-160, June 28 2011-July 1 2011
- [69] Daniel Baur and Jörg Domaschka. 2016. Experiences from building a cross-Cloud orchestration tool. In Proceedings of the 3rd Workshop on CrossCloud Infrastructures & Platforms (CrossCloud '16). ACM, New York, NY, USA, , Article 4 , 6 pages
- [70] <https://www.iso.org/standard/67546.html>
- [71] Herbst, Nikolas, et al. "Ready for rain? a view from spec research on the future of cloud metrics." arXiv preprint arXiv:1604.03470 (2016).
- [72] <https://aws.amazon.com/cloud-directory/sla/>
- [73] John, Lizy Kurian, and Lieven Eeckhout, eds. Performance evaluation and Benchmarking. CRC Press, 2018.
- [74] "OpenStack Docs: Welcome to Ceilometer's documentation!" [Online]. Available: <https://docs.openstack.org/ceilometer/latest/>. [Accessed: 23-Dec-2017].
- [75] G. Kousiouris, T. Cucinotta, and T. Varvarigou, "The Effects of Scheduling, Workload Type and Consolidation Scenarios on Virtual Machine Performance and Their Prediction Through Optimized Artificial Neural Networks," J Syst Softw, vol. 84, no. 8, pp. 1270–1291, Aug. 2011.
- [76] Kapsoulis, N., Psychas, A., Litke, A., & Varvarigou, T. (2021). Reinforcing SLA Consensus on Blockchain. Computers, 10(12), 159.