



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Εργαλεία ανάλυσης απόδοσης και καταγραφή
παραμέτρων εκτέλεσης δοκιμίων αναφοράς σε
Υπολογιστικά Νέφη**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΟΚΚΟΣΟΥΛΗ ΜΙΑΤΙΑΔΗ

Επιβλέπων : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2017

Η σελίδα αυτή είναι σκόπιμα λευκή.



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

**Εργαλεία ανάλυσης απόδοσης και καταγραφή
παραμέτρων εκτέλεσης δοκιμών αναφοράς σε
Υπολογιστικά Νέφη**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΟΚΚΟΣΟΥΛΗ ΜΙΑΤΙΑΔΗ

Επιβλέπων : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

(Υπογραφή)

.....

Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....

Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....

Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2017

(Υπογραφή)

.....

ΚΟΚΚΟΣΟΥΛΗΣ ΜΙΛΤΙΑΔΗΣ

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

© 2017 – All rights reserved

Περίληψη

Όλο και περισσότερες εταιρείες εξετάζουν σοβαρά το ενδεχόμενο να χρησιμοποιήσουν τις υπηρεσίες ενός υπολογιστικού νέφους ως μέρους των λειτουργιών τους, τόσο ως προμηθευτές όσο και ως καταναλωτές. Υπάρχει βέβαια και η άλλη πλευρά, η οποία περιλαμβάνει πελάτες και προμηθευτές οι οποίοι εκφράζουν την ανασφάλεια τους σχετικά με τη διαθεσιμότητα, δέσμευση, νομοθεσία, απόδοση, ασφάλεια μίας πλατφόρμας υπολογιστικού νέφους. Ένα από τα σημαντικά κριτήρια που λαμβάνεται υπόψη από τους πελάτες στην απόφασή τους να εμπιστευτούν υποδομές υπολογιστικού νέφους για τις εφαρμογές και τα δεδομένα τους είναι όντως η απόδοση και η αξιοπιστία.

Συνεπώς το ερώτημα που τίθεται είναι αν υπάρχουν ήδη διαθέσιμα προς τους διαχειριστές υπολογιστικών υποδομών τα απαραίτητα εργαλεία για τον συνεχή έλεγχο και αξιολόγηση της απόδοσης των παρεχόμενων υπηρεσιών. Η απόδοση μετριέται με βάση συγκεκριμένες μετρικές που αφορούν τους παρεχόμενους πόρους (για παράδειγμα προσφερόμενη υπολογιστική ισχύ, ή προσφερόμενη μνήμη, κτ) και συγκρίνεται με στόχους που έχουν οριστεί εκ των προτέρων ως μέρος ενός συμβολαίου με τον εκάστοτε πάροχο (Service Level Agreement – SLA).

Στη παρούσα διπλωματική εργασία εξετάζεται η ενοποίηση της βιβλιοθήκης Availability Auditor Library (3ALib) και η προσθήκη της ως επέκταση σε μία ήδη υπάρχουσα βιβλιοθήκη που ονομάζεται Cloud Benchmark tool (Ε. Κεβάνη, 2014). Η ερευνητική υπόθεση που τίθεται προς διερεύνηση στα πλαίσια της διπλωματικής είναι ότι η βιβλιοθήκη 3ALib μπορεί να προσφέρει ένα ενιαίο περιβάλλον εργασίας για το χρήστη / πελάτη του υπολογιστικού νέφους για να παρακολουθεί τα επίπεδα διαθεσιμότητας που ορίζονται από τον πάροχο. Η ενιαία διεπαφή που διατίθεται στον χρήστη/πελάτη κρύβει τις τεχνικές λεπτομέρειες του πίσω μέρους της εφαρμογής και συνολικά η εφαρμογή ολοκληρώνεται με τη βιβλιοθήκη Cloud Benchmark tool.

Λέξεις Κλειδιά: <<Service Level Agreement – SLA, Υπολογιστικά Νέφη, PaaS, IaaS, SaaS, Cloud Performance Benchmarking, 3ALib>>

Η σελίδα αυτή είναι σκόπιμα λευκή.

Abstract

More and more companies are seriously considering the possibility to use the services of a cloud computing as part of their functions, both as suppliers and as consumers. Of course there is another side, which includes customers and suppliers who express their uncertainty about the availability, commitment, legislation, performance and security of a cloud computing platform. One of the important criteria to be considered by the customers in their decision to trust cloud computing infrastructure for applications and data performance and reliability are indeed.

Therefore the question that arises is whether there are already available to the computing infrastructure managers the necessary tools for the continuous monitoring and evaluation of service performance. Performance is measured based on specific metrics related to the provided resources (for example offered computing power, or offered memory, etc.) and compared with objectives set in advance as part of a contract with the provider (Service Level Agreement - SLA) .

In this dissertation work we examine the implementation of Availability Auditor Library (3ALib) as an extension of an existing library that has been developed and is called Cloud Benchmark tool ((E. Κεβάνη, 2014)). The research hypothesis put up for investigation in this research work is that the library 3ALib can provide a single interface for the user / client cloud computing to monitor availability levels specified by the provider. The single interface available to the user / client hides the technical details of the back of the application and overall the application is completed with the Cloud Benchmark tool library.

Keywords: << Service Level Agreement – SLA, Cloud Computing, PaaS, IaaS, SaaS, Cloud Performance Benchmarking, 3ALib >>

Η σελίδα αυτή είναι σκόπιμα λευκή.

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία διενεργήθηκε στο Εθνικό Μετσόβιο Πολυτεχνείο, στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών κατά το ακαδημαϊκό έτος 2016-2017.

Θα ήθελα να ευχαριστήσω θερμά την Καθηγήτρια του Τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής, κα. Θεοδώρα Βαρβαρίγου για την ευκαιρία που μου έδωσε μέσα από την παρακολούθηση του επιλεγόμενου μαθήματος «Δικτυακός Προγραμματισμός» να ασχοληθώ με το ενδιαφέρον θέμα του υπολογιστικού νέφους (cloud computing) κι επίσης για την καθοδήγηση και τις συμβουλές της.

Η εκπόνηση και ολοκλήρωση της παρούσης διπλωματικής εργασίας δεν θα ήταν εφικτή χωρίς την πολύ σημαντική και πολύπλευρη υποστήριξη και βοήθεια, τόσο σε θεωρητικό όσο και σε τεχνικό επίπεδο, του διδάκτορα και ερευνητή του ΕΠΙΣΕΥ (Ερευνητικό Πανεπιστημιακό Ινστιτούτο Συστημάτων Επικοινωνιών και Υπολογιστών) του Ε.Μ.Π. κου. Γιώργου Κουσιουρή, καθώς επίσης και της υποψήφιας διδάκτορος κας Αθανασίας Ευαγγελινού, στους οποίους οφείλω ιδιαίτερες ευχαριστίες.

Πίνακας περιεχομένων

1	Εισαγωγή	1
2	Βιβλιογραφία (State of the art)	5
2.1	Υπολογιστικό Νέφος.....	5
2.2	Μοντέλα Διαχείρισης Απόδοσης στο ΥΝ.....	7
2.2.1	<i>Επίπεδα Παροχής Υπηρεσιών</i>	8
2.2.2	<i>Παράμετροι και Μετρικές των Επιπέδων Εξυπηρέτησης (SLAs) στο ΥΝ</i>	10
2.3	Πλατφόρμα-ως-υπηρεσία (PaaS) και Υποδομή-ως-υπηρεσία (IaaS)	12
2.3.1	<i>Amazon EC2</i>	15
2.4	Σύγκριση των Επιπέδων Εξυπηρέτησης ανάμεσα σε παρόχους υπηρεσιών ΥΝ.....	16
2.5	Τιμολόγηση Επιπέδων Εξυπηρέτησης.....	17
2.6	Εργαλεία Συγκριτικής Αξιολόγησης Υπολογιστικών Νεφών.....	19
2.6.1	<i>Cloud Benchmark tool</i>	20
2.7	Επέκταση του Cloud Benchmark Tool: 3ALib.....	24
3	Αρχιτεκτονική 3ALib – Cloud Benchmark tool	27
3.1	Εισαγωγή – μεθοδολογία και πλαίσιο εργαλείων ARTIST	27
3.2	Σουίτα Εργαλείων ARTIST	28
3.2.1	<i>Περιπτώσεις Εφαρμογής (Use cases)</i>	30
3.3	3ALib Αρχιτεκτονική – Διαγράμματα Κλάσεων και Ακολουθίας	31
3.4	Διασύνδεση 3ALib με την Πλατφόρμα Cloud Benchmark tool	35
3.5	Μεθοδολογία Διασύνδεσης 3ALib με την Πλατφόρμα Cloud Benchmark tool.....	39
3.5.1	<i>Spring web MVC framework</i>	39
4	Μελέτη Περίπτωσης Καταγραφών 3ALib	46
4.1	Γενικές Εντολές	46
4.2	Ερωτήματα.....	47
5	Συμπεράσματα – Μελλοντικές Επεκτάσεις	50
6	Αναφορές και Βιβλιογραφία	52

1

Εισαγωγή

Το cloud computing είναι μία από τις μεγάλες τάσεις στον κλάδο της πληροφορικής σήμερα. Η περιοχή εξακολουθεί να είναι νέα και διαρκώς αναπτυσσόμενη και πολλές εταιρείες προσαρμόζουν τις λειτουργίες τους σε αυτό το πρότυπο, τόσο ως προμηθευτές όσο και ως καταναλωτές. Η υπηρεσία υπολογιστικού νέφους που προσφέρεται από τους προμηθευτές ποικίλλει, αλλά κοινή συνισταμένη όλων είναι η παροχή πρόσβασης από απόσταση σε υπολογιστικούς πόρους. Οι πόροι αυτοί μπορεί να είναι διακομιστές, χώροι αποθήκευσης, εφαρμογές και πολλά άλλα.

Το πλεονέκτημα του Υπολογιστικού Νέφους (ΥΝ) είναι ότι ένας οργανισμός που έχει ανάγκη της εγκατάστασης μίας εφαρμογής μπορεί να το κάνει αυτό χωρίς να χρειάζεται να προβεί σε αγορά και συντήρηση υλικού και λογισμικού υπό την ιδιοκτησία του. Αντί αυτού η φιλοξενία της εφαρμογής στο ΥΝ σημαίνει ότι η αντίστοιχη υπηρεσία προσφέρει την δυνατότητα της αυτόματης προσαρμογής στη τρέχουσα κίνηση και με αυτόν τον τρόπο, η αύξηση ή η μείωση του αριθμού των χρηστών είναι διαχειρίσιμη.

Υπάρχει μία ευρέως χρησιμοποιούμενη κατηγοριοποίηση των μοντέλων υπηρεσιών η οποία τα καθιστά διαφορετικά με βάση το επίπεδο εικονικής υποδομής που παρέχεται. Τα τρία πιο συχνά αναφερόμενα μοντέλα είναι το Λογισμικό-, Πλατφόρμα-, και Υπηρεσία- ως-Υπηρεσία. Λογισμικό-ως-Υπηρεσία (Software-as-a-Service, SaaS) είναι ένα κομμάτι του λογισμικού που διατίθενται στους πελάτες μέσω του Διαδικτύου. Η εφαρμογή τρέχει στην υποδομή του παρόχου ΥΝ και είναι συνήθως προσβάσιμη στους χρήστες μέσω ενός web browser. Οι

χρήστες της υπηρεσίας δεν ασχολούνται με την εγκατάσταση του λογισμικού ή την ενημέρωσή τους και συνήθως οι χρήστες χρεώνονται σε μηνιαία βάση ή σύμφωνα με κάποιο άλλο σχέδιο τιμολόγησης προπληρωμένου χρόνου (pay-as-you-go).

Το μοντέλο Πλατφόρμα-ως-Υπηρεσία (Platform-as-a-Service, PaaS) επιτρέπει στο χρήστη να αναπτύξει εφαρμογές με γλώσσες προγραμματισμού και εργαλεία που διατίθενται από τον προμηθευτή της πλατφόρμας. Αυτές οι εφαρμογές μπορεί στη συνέχεια να τρέχουν από την υποδομή της πλατφόρμας που ο χρήστης δεν διαχειρίζεται ή ελέγχει άμεσα. Το μοντέλο Υποδομή-ως-Υπηρεσία (Infrastructure-as-a-Service, IaaS) παρέχει στους πελάτες υπολογιστικούς πόρους, όπως επεξεργαστική ισχύ, χώρο αποθήκευσης και δικτυακές υπηρεσίες. Οι πόροι αυτοί μπορούν να χρησιμοποιηθούν για την ανάπτυξη και εγκατάσταση λογισμικού και ο πελάτης είναι πλήρως υπεύθυνος για τη διαχείριση των λειτουργικών συστημάτων και για όποιο άλλο λογισμικό εγκαθίσταται.

Από την άλλη πλευρά υπάρχουν πολλές ενστάσεις σχετικά με τη χρήση του Υπολογιστικού Νέφους και απόψεις σχετικά με το γιατί δεν θα πρέπει να χρησιμοποιείται. Οι ενστάσεις μπορούν να βασίζονται σε πραγματικά προβλήματα αλλά ορισμένα από αυτά μπορούν να ξεπεραστούν ή να μην μη σχετίζονται απόλυτα με το ΥΝ. Μερικές από τις πιο κοινές ενστάσεις κατά της χρήσης του ΥΝ είναι:

- η διαθεσιμότητα
- η δέσμευση
- η αδειοδότηση του λογισμικού
- νομοθεσία και νομικά ζητήματα
- απόδοση
- ασφάλεια

Για παράδειγμα η διαθεσιμότητα είναι πολύ κρίσιμη σε πολλά συστήματα και η αίσθηση της απώλειας του ελέγχου, μπορεί να αποθαρρύνει την μετάβαση σε πλατφόρμες υπολογιστικού νέφους. Ακόμη και μεγάλοι πάροχοι είχαν διακοπή της λειτουργίας τους που μπορεί να διαρκέσουν ακόμα και μέρες (πχ η Amazon, στο κέντρο δεδομένων της στη North Virginia). Βέβαια σοβαρές διακοπές της λειτουργίας των συστημάτων ΥΝ παρουσιάζονται σπάνια και από την άλλη πλευρά τίθεται το αντίστροφο ερώτημα: οι υποδομές που τρέχουν εντός του οργανισμού-πελάτη είναι ανθεκτικές και αποδοτικές;

Ο τρόπος αξιολόγησης είναι η πιο συχνή ένσταση ως προς τη χρήση του ΥΝ, αλλά χωρίς να είναι αρκετά τεκμηριωμένη. Μία άποψη είναι ότι οι πάροχοι εγγυώνται ότι πραγματοποιούν τις απαραίτητες αναλύσεις για την εξασφάλιση της μέγιστης απόδοσης των υποδομών και ότι υπάρχουν ευρέως δοκιμασμένες τεχνικές λύσεις που μπορούν να χρησιμοποιηθούν. Η

βελτιστοποίηση της υπολογιστικής ισχύς για την εκτέλεση συναλλαγών και εφαρμογών, της ανάγνωσης και εγγραφής δεδομένων αποτελούν παραδείγματα τομέων όπου γνωστές τεχνικές μέτρησης της απόδοσης μπορεί να χρησιμοποιηθούν.

Από την άλλη πλευρά, η αξιολόγηση της απόδοσης των πλατφόρμων που διατίθενται ως επί το πλείστον θα εξαρτάται από τη φήμη των παρόχων και των πραγματικών αποτελεσμάτων τους στην πράξη, παραδεχόμενοι ότι οι αποκλίσεις από τα συμφωνημένα επίπεδα απόδοσης (SLAs) ενδεχομένως να μη μπορούν να αποκαλυφθούν στο κοινό. Η υπεράσπιση αυτής της προσέγγισης για την αξιολόγηση της απόδοσης πηγάζει από την άποψη ότι η αξιολόγηση της απόδοσης απαιτεί τον ορισμό συγκεκριμένων μετρικών για να μπορεί να συγκριθεί ποσοτικά ή ποιοτικά ανάμεσα στους παρόχους.

Έχοντας υπόψη το παρακάτω πλαίσιο:

1. Η νεφο-υπολογιστική ορίζει το μοντέλο για την παροχή κατά ανάγκη πρόσβασης σε μια δεξαμενή κοινών υπολογιστικών πόρων, με την ελάχιστη παρέμβαση του παρόχου.
2. Καθώς αυξάνεται ο αριθμός των καταναλωτών που τείνουν προς τη χρήση νεφο-υπολογιστικών υπηρεσιών, θεωρείται ότι ένας συγκεκριμένος φόρτος εργασίας αποστέλλεται προς ένα σύννεφο παρόχων. Ένα Συμφωνημένο Επίπεδο Υπηρεσιών (SLA) μεταξύ των καταναλωτών και των παρόχων γίνεται υψίστης σημασίας για να διασφαλιστεί ότι η ποιότητα των υπηρεσιών διατηρείται σε ικανοποιητικά επίπεδα, και ανεξάρτητα από τη δυναμική φύση του νεφο-υπολογιστικού περιβάλλοντος .
3. Η σύμβαση για το Συμφωνημένο Επίπεδο Υπηρεσιών (SLA) περιέχει μια εξήγηση της συμφωνηθείσας υπηρεσίας, τις παραμέτρους του επιπέδου των υπηρεσιών, και τις εγγυήσεις σχετικά με την ποιότητα των υπηρεσιών, ρυθμίσεις, χρόνου και τρόπου αποκατάστασης για όλες τις περιπτώσεις παραβάσεων των παραπάνω όρων.

Στη παρούσα διπλωματική εργασία εξετάζεται η ενσωμάτωση της βιβλιοθήκης Availability Auditor Library (3ALib) ως επέκταση μίας ήδη υπάρχουσας βιβλιοθήκης που έχει αναπτυχθεί στο Τμήμα Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς και ονομάζεται Cloud Benchmark tool (E. Κεβάνη, 2014). Η ερευνητική υπόθεση που τίθεται προς διερεύνηση στα πλαίσια της διπλωματικής είναι ότι η βιβλιοθήκη 3ALib μπορεί να προσφέρει ένα ενιαίο περιβάλλον εργασίας για το χρήστη / πελάτη του υπολογιστικού νέφους για να παρακολουθεί τα επίπεδα διαθεσιμότητας που ορίζονται από τον πάροχο. Η ενιαία διεπαφή που διατίθεται στον χρήστη/πελάτη κρύβει τις τεχνικές λεπτομέρειες του πίσω μέρους της εφαρμογής και συνολικά η εφαρμογή ολοκληρώνεται με τη βιβλιοθήκη Cloud Benchmark tool .

Επομένως η εργασία αυτή έχει την εξής δομή:

- Το κεφάλαιο 2 κάνει μία καταγραφή των πρακτικών και των μοντέλων διαχείρισης SLAs όπως εφαρμόζονται στις τωρινές υποδομές ΥΝ. Παρουσιάζει επίσης τις βασικές λειτουργίες της βιβλιοθήκης 3ALib και της βιβλιοθήκης Cloud Benchmark tool.
- Το κεφάλαιο 3 παρουσιάζει την υλοποίηση της αρχιτεκτονικής για τη διασύνδεση των δύο βιβλιοθηκών βάσει της τεχνολογίας Spring (Java).
- Στο κεφάλαιο 4 χρησιμοποιείται η βιβλιοθήκη 3ALib για τη διασύνδεση σε πάροχο υπολογιστικού νέφους (Amazon WS) τη προβολή και αναζήτηση SLA αποτελεσμάτων.
- Το κεφάλαιο 5 ολοκληρώνει την εργασία με τα συμπεράσματα και μελλοντικές επεκτάσεις της.

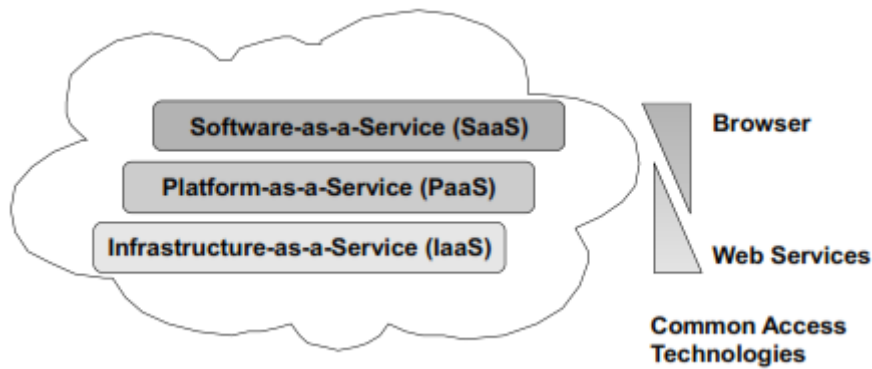
2

Βιβλιογραφία (State of the art)

2.1 Υπολογιστικό Νέφος

Σε μία έκθεση του οίκου Gartner ((Brodkin, J., 2008)) το υπολογιστικό νέφος ορίζεται ως μία υπηρεσία με μαζικά επεκτάσιμες δυνατότητες πληροφορικής οι οποίες παρέχονται σε εξωτερικούς πελάτες χρησιμοποιώντας τις τεχνολογίες του Διαδικτύου. Η υπηρεσία αυτή διαθέτει μοναδικά χαρακτηριστικά που απαιτούν την αξιολόγηση του κινδύνου σε τομείς όπως η ακεραιότητα των δεδομένων, η ανάκτηση και προστασία προσωπικών και απόρρητων δεδομένων, καθώς και την αξιολόγηση των νομικών ζητημάτων σε τομείς όπως ο αυτόματος εντοπισμός νέων υπηρεσιών (e-discovery), η συμμόρφωση με τους κανονισμούς, και η πραγματοποίηση ελέγχων. Υπηρεσίες YN με τις παραπάνω δυνατότητες προσφέρουν μεγάλοι πάροχοι όπως η Amazon (EC2), Microsoft (Azure) και η Google (App Engine).

Σύμφωνα με τον ορισμό του YN, ένα τέτοιο σύστημα προσφέρει δυναμική επέκταση πόρων για την παροχή μίας υπηρεσίας μέσω του Διαδικτύου και ως εκ τούτου υπόσχεται πολλά οικονομικά οφέλη που θα διανεμηθούν μεταξύ των εμπλεκόμενων. Ανάλογα με τον τύπο των πόρων που παρέχονται από YN, μπορούν να οριστούν διακριτά στρώματα (Εικόνα 1).



Εικόνα 1: στρώματα υπολογιστικού νέφους και τεχνολογίες πρόσβασης (Jensen, M. ; Horst Gortz; Schwenk, J. ; Gruschka, N. ; Iacono, L.L., 2009)

Το πιο χαμηλό στρώμα παρέχει βασικές υποδομές και συστατικά όπως επεξεργαστική ισχύ, μνήμη και αποθήκευση, και ονομάζονται Υποδομές-ως-Υπηρεσία (IaaS). Η υποδομή Elastic Compute Cloud (EC2) της Amazon είναι ένα σημαντικό παράδειγμα για το μοντέλο IaaS. Στην κορυφή του IaaS, επιτρέπει την φιλοξενία εφαρμογών σε ένα συγκεκριμένο περιβάλλον ειδικά διαμορφωμένο για τις υπηρεσίες αυτές. Το μοντέλο αυτό ονομάζεται Πλατφόρμα-ως-Υπηρεσία με χαρακτηριστικό παράδειγμα την πλατφόρμα App Engine της Google. Το πιο πάνω επίπεδο, Εφαρμογές-ως-Υπηρεσία, προσφέρει στους χρήστες έτοιμες προς χρήση εφαρμογές. Για την πρόσβαση σε όλες τις παραπάνω υποδομές του ΥΝ απαιτούνται δύο ειδών τεχνολογίες: υπηρεσίες ιστού είναι απαραίτητες για την πρόσβαση σε υπηρεσίες IaaS, ενώ ο περιηγητής ιστού είναι χρήσιμος για τη πρόσβαση σε εφαρμογές SaaS. Η πρόσβαση σε υπηρεσίες PaaS γίνεται και με τους δύο τρόπους (Jensen, M. ; Horst Gortz; Schwenk, J. ; Gruschka, N. ; Iacono, L.L., 2009).

Όλα τα παραπάνω μοντέλα υπόσχονται μείωση των κεφαλαιακών δαπανών (κόστος επένδυσης), και περιλαμβάνει μείωση του κόστους του υλικού στο στρώμα IaaS και μειωμένο κόστος αδειών σε όλα τα στρώματα. Ειδικά το στρώμα IaaS εξασφαλίζει για ένα πελάτη ένα εικονικό κέντρο δεδομένων με τη μέγιστη δυνατή απόδοση και βέλτιστη αξιοποίηση των διαθέσιμων πόρων . Επιπλέον, εξασφαλίζονται μειώσεις των λειτουργικών δαπανών (OpEx) λόγω της χρήσης λιγότερου υλικού, αδειών και ενημερωμένων εκδόσεων προγραμμάτων.

Μία σύγχρονη υποδομή ΥΝ περιλαμβάνει πέντε βασικά χαρακτηριστικά (Takabi, H;Joshi, J.B.D.;Gail-Joon Ahn, 2010): αυτό εξυπηρετήση κατά απαίτηση, πανταχού πρόσβαση στο δίκτυο, διάθεση πόρων ανεξάρτητα από την τοποθεσία τους, ταχεία ελαστικότητα, και μετρήσιμη παροχή υπηρεσίας. Η ταχεία ελαστικότητα επιτρέπει τη γρήγορη κλιμάκωση ή

αποκλιμάκωση της διάθεσης πόρων για μία υπηρεσία. Μετρήσιμη παροχή υπηρεσιών σημαίνει την εφαρμογή συγκεκριμένων επιχειρησιακών μοντέλων και βοηθούν τους παρόχους να διαχειριστούν και να βελτιστοποιήσουν τη χρήση των υπολογιστικών πόρων μέσω εργαλείων αυτοματοποιημένης κατανομής πόρων, εξισορρόπησης φορτίου, και καταμέτρησης.

Στο YN εφαρμόζονται επιπρόσθετα τα εξής μοντέλα υλοποίησης: δημόσια, ιδιωτικά, υβριδικά και YN για μία συγκεκριμένη κοινότητα χρηστών. Τα YN δημόσιας χρήσης είναι διαθέσιμα στο ευρύ κοινό ενώ τα ιδιωτικής χρήσης YN είναι διαθέσιμα για αποκλειστική χρήση των εκάστοτε οργανισμών. Υπάρχουν επίσης YN που είναι αφιερωμένα σε μία ορισμένη κοινότητα ή ομάδα.

2.2 Μοντέλα Διαχείρισης Απόδοσης στο YN

Παρά το γεγονός ότι ο ευρύς τομέας του Cloud Computing έχει αλλάξει με την πάροδο του χρόνου, οι υπηρεσίες που προσφέρονται έχουν διαιρεθεί σε τρεις μεγάλες κατηγορίες: Υποδομή ως Υπηρεσία (Infrastructure as a Service – IaaS), Πλατφόρμα ως Υπηρεσία (Platform as a Service – PaaS) και Λογισμικό ως Υπηρεσία (Software as a Service) (Oracle, 2011).

1) Υποδομή ως Υπηρεσία

Οι πάροχοι τέτοιου είδους υπηρεσιών, όπως AWS (Amazon Web Services), προσφέρουν έναν εικονικό server και αποθηκευτικό χώρο, καθώς και διεπαφές προγραμματισμού εφαρμογών (Application Programming Interfaces – APIs) προκειμένου να επιτρέπεται να γίνονται αιτήσεις από άλλα προγράμματα ή/και ανταλλαγή δεδομένων. Έτσι επιτρέπεται στους χρήστες να μεταφέρουν φόρτους εργασίας προς εκτέλεση σε ένα εικονικό μηχάνημα. Οι χρήστες έχουν έναν προσυμφωνημένο χώρο αποθήκευσης και μπορούν να ξεκινήσουν / σταματήσουν / έχουν πρόσβαση και να ρυθμίσουν το εικονικό μηχάνημα και τον αποθηκευτικό χώρο, αναλόγως του τι χρειάζονται ακριβώς. Συνήθως οι πάροχοι IaaS προσφέρουν διαφόρων μεγεθών instances καθώς και instances που μπορεί να είναι είτε βελτιστοποιημένα ως προς τη διαθέσιμη μνήμη είτε ως προς την υπολογιστική ισχύ, ακόμα και instances προσαρμοσμένα στις ανάγκες διαφόρων φόρτων εργασίας (workloads).

2) Πλατφόρμα ως Υπηρεσία

Οι πάροχοι σε αυτή την περίπτωση φιλοξενούν εργαλεία ανάπτυξης στις υποδομές τους. Οι χρήστες έχουν πρόσβαση σε αυτά τα εργαλεία μέσω του Internet χρησιμοποιώντας APIs και web portals. Ο συγκεκριμένος τύπος υπηρεσίας χρησιμοποιείται γενικά για ανάπτυξη λογισμικού και πολλοί πάροχοι θα φιλοξενήσουν το λογισμικό μετά την ανάπτυξή του. Τυπικοί πάροχοι αυτού του είδους είναι: το Force.com της Salesforce.com, το AWS Elastic Beanstalk και το Google App Engine.

3) Λογισμικό ως Υπηρεσία

Το Λογισμικό ως Υπηρεσία είναι ένα μοντέλο που παρέχει εφαρμογές λογισμικού, οι οποίες συχνά αποκαλούνται web services (διαδικτυακές υπηρεσίες ή υπηρεσίες web). Για παράδειγμα, το Microsoft Office 365 είναι μια προσφορά Λογισμικού ως Υπηρεσία που αφορά λογισμικό παραγωγικότητας και υπηρεσίες email. Οι χρήστες έχουν πρόσβαση σε τέτοιες υπηρεσίες από οποιαδήποτε τοποθεσία χρησιμοποιώντας έναν Η/Υ ή μια κινητή συσκευή που έχει πρόσβαση στο Internet.

2.2.1 Επίπεδα Παροχής Υπηρεσιών

Ένα SLA¹ ορίζει το επίπεδο παροχής υπηρεσιών και διατυπώνεται ρητά στη σύμβαση ανάμεσα στον πελάτη και τον πάροχο (Takabi, H.;Joshi, J.B.D.;Gail-Joon Ahn, 2010). Καταγράφει την κοινή αντίληψη ανάμεσα στα δύο μέρη σχετικά με υπηρεσίες, προτεραιότητες, ευθύνες, εγγυήσεις και ρήτρες. Έτσι στο ΥΝ τα επίπεδα παροχής υπηρεσιών που διατυπώνονται ελέγχουν τη χρήση των υπολογιστών πόρων. Ως εκ τούτου, το κύριο ζήτημα για το ΥΝ είναι να χτίσει ένα νέο στρώμα για να υποστηρίξει τη δυναμική διαπραγμάτευση μίας σύμβασης και να παρακολουθεί την εφαρμογή των κανόνων της σύμβασης (E. Aljournah, F. Al-Mousawi et al, 2015). Βασικές πτυχές προσδιορισμού του Επιπέδου Εξυπηρέτησης μίας Υπηρεσίας είναι οι παρακάτω:

- Ακρίβεια: Η έννοια της Ακρίβειας νοιάζεται για το ποσοστό σφάλματος σε μία υπηρεσία. Ο προσδιορισμός του μέσου αριθμού των σφαλμάτων σε μια δεδομένη χρονική περίοδο είναι εφικτός.

¹ Service Level Agreement

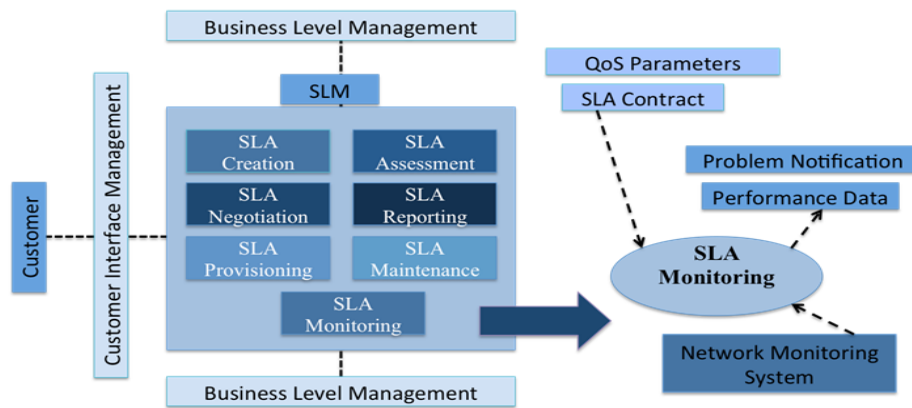
- Διαθεσιμότητα: η Διαθεσιμότητα ορίζεται από το λόγο του χρόνου εργασίας της υπηρεσίας. Ένας άλλος ορισμός μπορεί να πηγάζει από τη στιγμή που η υπηρεσία μπορεί να χρησιμοποιηθεί σε σχέση με το συνολικό χρόνο. Η πτυχή αυτή συγκρίνει το μέσο χρόνο σε σχέση με τις αποτυχίες της υπηρεσίας. Έτσι, δεν θίγει τη φήμη του νέφους αυξάνοντας ή μειώνοντας την εμπιστοσύνη του χρήστη. Η διαθεσιμότητα συνήθως υπολογίζεται από την πιθανότητα ότι το σύστημα θα είναι εξοπλισμένο όταν απαιτείται. Είναι πιθανό να προσδιορίσει:
 - την απόκριση του συστήματος, όταν εμφανίζεται κάποια δυσλειτουργία
 - τον χρόνο που απαιτείται για να προσδιορίσει μια αποτυχία
 - τη περίοδο που απαιτείται για να επανακάμψει από μια δυσλειτουργία
 - αν αφιερώνεται χρόνος για την κάλυψη των δυσλειτουργιών
 - τον μη-παραγωγικό χρόνο που απαιτείται για την εφαρμογή διορθωτικών επεμβάσεων.

- Χωρητικότητα είναι ο αριθμός των ταυτόχρονων αιτημάτων στα οποία καλείται να ανταποκριθεί η υπηρεσία σε ένα καθορισμένο χρονικό διάστημα. Είναι πιθανό να καθορίσει το μέγιστο αριθμό ταυτόχρονων αιτημάτων στα οποία καλείται να διαχειριστεί μια υπηρεσία σε ένα ορισμένο χρόνο.

- Το Κόστος αφορά τη ζήτηση υπηρεσιών ΥΝ. Είναι πιθανό να προσδιορίσει:
 - Το κόστος ανά ζήτηση
 - Το κόστος με βάση τον όγκο των πληροφοριών
 - Ζώνες διαφοροποίησης του κόστους που σχετίζονται με τις ώρες αιχμής της προσφερόμενης υπηρεσίας.

- Η καθυστέρηση (Latency) νοιάζεται για το μεγαλύτερο δυνατό στιγμιότυπο του χρόνου ανάμεσα στην εμφάνιση μιας ζήτησης και το σημείο τερματισμού της ζήτησης.

Η ολοκληρωμένη διαδικασία διαχείρισης των διαφόρων Επιπέδων Εξυπηρέτησης από την αρχή μέχρι την αξιολόγηση διακρίνεται σε τρία επίπεδα: α) διαχείριση σε επιχειρησιακό επίπεδο, β) διαχείριση σε επίπεδο υπηρεσιών, και γ) διαχείριση σε επίπεδο δικτύου. Η διαχείριση σε επίπεδο υπηρεσιών αποτελείται από διάφορες λειτουργίες που περιλαμβάνουν τη δημιουργία SLA, διαπραγμάτευση, παροχή, παρακολούθηση, συντήρηση, υποβολή αναφορών και την αξιολόγηση όπως αυτές παρουσιάζονται στην Εικόνα 2.



Εικόνα 2: Διαχείριση και αδιάκοπη παρακολούθηση του Προσφερόμενου Επιπέδου Εξυπηρέτησης (E. Aljoumah, F. Al-Mousawi et al, 2015)

Η δυνατότητα εποπτείας και αδιάκοπης παρακολούθησης είναι μια σημαντική απαίτηση για ένα και αφορά τη δυνατότητα του παρόχου των υπηρεσιών και του πελάτη να μπορούν να παρατηρήσουν και να διαχειρίζονται τη συμπεριφορά της υπηρεσίας που σχετίζονται με το SLA. Εναλλακτικά μπορεί να απασχολούν ένα έμπιστο τρίτο μέρος για να το πράξουν. Χωρίς την απαίτηση αυτή, θα ήταν αδύνατο για ένα πελάτη να δηλώσει ότι υπάρχει παραβίαση του SLA. Ως εκ τούτου, οι όροι της σύμβασης μπορεί να αγνοηθούν από τον πάροχο των υπηρεσιών. Το πρόβλημα που αντιμετωπίζουν οι μηχανικοί κατά την παρακολούθηση της συμμόρφωσης ενός SLA είναι ο προσδιορισμός ανάλογων μετρικών απόδοσης. Το SLA πρέπει να είναι σχεδιασμένο για να εγγυάται τη δυνατότητα της αδιάκοπης παρακολούθησης, και την ελαχιστοποίηση της πιθανότητας χαμηλής συμμόρφωσης (E. Aljoumah, F. Al-Mousawi et al, 2015).

2.2.2 Παράμετροι και Μετρικές των Επιπέδων Εξυπηρέτησης (SLAs) στο YN

Οι μετρικές αξιοποιούνται στο πλαίσιο της διαδικασίας παρακολούθησης, βελτίωσης της διαδικασίας ενός λογισμικού, της πολιτικής απασχόλησης των επιχειρήσεων, και κυρίως κάθε τομέα ο οποίος απαιτεί τη συλλογή πληροφοριών για να επιβεβαιωθεί εάν οι στόχοι επιτυγχάνονται ή όχι. Οι παράμετροι μίας σύμβασης επιπέδου εξυπηρέτησης (SLA) προσδιορίζονται από μια ομάδα μετρικών. Αυτές οι μετρικές αποφασίζουν τις ποσότητες των δεδομένων που πρέπει να συλλεχθούν για να επιβεβαιώσουν εάν οι SLA παράμετροι έχουν επιτευχθεί. Οι ακόλουθες στρατηγικές προτείνονται για να περιγράψουν τις SLA παραμέτρους με έναν τρόπο που απλοποιεί τη διαχείριση Επιπέδων Εξυπηρέτησης. Λαμβάνοντας υπόψη ότι τα SLAs συνήθως συμφωνούνται μετά από διαπραγμάτευση, οι στρατηγικές αυτές ισχύουν τόσο για τους παρόχους όσο και τους πελάτες (E. Aljoumah, F. Al-Mousawi et al, 2015).

Παράμετρος	Περιγραφή
CPU capacity	Η επεξεργαστική ισχύ (CPU speed) που εκχωρείται σε μία εικονική μηχανή (VM)
Memory size	Η μνήμη που εκχωρείται σε μία εικονική μηχανή
Boot time	Ο χρόνος που απαιτείται για την εκκίνηση της μηχανής
Storage	Αποθηκευτικός χώρος που εκχωρείται σε μία εικονική μηχανή
Scale up	Μέγιστος επιτρεπόμενος αριθμός εικονικών μηχανών
Scale down	Ελάχιστος επιτρεπόμενος αριθμός εικονικών μηχανών
Scale up time	Ο χρόνος που απαιτείται για την αύξηση του αριθμού των μηχανών
Scale down time	Ο χρόνος που απαιτείται για την μείωση του αριθμού των μηχανών
Auto scaling	Δηλώνει την δυνατότητα αυτόματης αύξησης του αριθμού των μηχανών ή όχι
Max number can be configured on physical server	Ο μέγιστος επιτρεπόμενος αριθμός εικονικών μηχανών που μπορούν να εκτελεστούν σε ένα φυσικό μηχάνημα
Availability	Ο εγγυημένος χρόνος διαθεσιμότητας μιας υπηρεσίας
Response time	Ο εγγυημένος χρόνος εκτέλεσης μίας υπηρεσίας και επιστροφής αποτελεσμάτων

Πίνακας 1: Μετρικές SLA για IaaS υποδομές (E. Aljoumah, F. Al-Mousawi et al, 2015)

Παράμετρος	Περιγραφή
Time and effort to create new application environment	Χρόνος και προσπάθεια που απαιτείται για τη δημιουργία ενός νέου περιβάλλοντος εφαρμογής
Time to redeploy application	Χρόνος που απαιτείται για αναδιάταξη και επαναφόρτωση της εφαρμογής
Time to promote application into a new lifecycle phase	Χρόνος που απαιτείται για την προώθηση μιας εφαρμογής σε επόμενη φάση κύκλου ζωής
Time and effort required integrating business process, event processor – creating a complex app	Χρόνος και προσπάθεια που απαιτείται για την ενσωμάτωση επιχειρηματικής διαδικασίας – δημιουργία σύνθετης εφαρμογής

Πίνακας 2: Μετρικές SLA για PaaS υποδομές (<https://dzone.com/articles/paas-performance-metrics>)

2.3 Πλατφόρμα-ως-υπηρεσία (PaaS) και Υποδομή-ως-υπηρεσία (IaaS)

PaaS (Platform-as-a-Service)

Το PaaS (Platform-as-a-Service) είναι μια κοινόχρηστη και ελαστικά επεκτάσιμη πλατφόρμα για εφαρμογές που προσφέρεται σαν μια ιδιωτική ή δημόσια υπηρεσία ΥΝ. Επιτρέπει στις επιχειρήσεις να ενσωματώσουν τις υπάρχουσες εφαρμογές τους σε μία κοινόχρηστη (shared), κοινή αρχιτεκτονική, καθώς επίσης και να αναπτύξουν καινούργιες εφαρμογές που αξιοποιούν τις κοινόχρηστες (shared) υπηρεσίες που παρέχονται από την πλατφόρμα. Η πλατφόρμα PaaS προσφέρει εξοικονόμηση κόστους μέσω της τυποποίησης και της υψηλότερης αξιοποίησης της κοινόχρηστης πλατφόρμας σε πολλαπλές εφαρμογές. Επίσης, παρέχει μεγαλύτερη ευελιξία μέσω της ταχύτερης ανάπτυξης εφαρμογών αξιοποιώντας κοινόχρηστες υπηρεσίες που βασίζονται σε πρότυπα και επεκτείνονται ελαστικά ανάλογα με την ζήτηση.

Η πλατφόρμα PaaS περιλαμβάνει βάση δεδομένων σαν υπηρεσία (Database-as-a-Service) βασισμένη στην βάση δεδομένων ενός παρόχου καθώς επίσης και middleware υπηρεσίες (Middleware-as-a-Service).

IaaS (Infrastructure-as-a-Service)

Περιλαμβάνει μια πλήρη γκάμα υπολογιστικών διακομιστών, αποθήκευσης, δικτύωσης, λογισμικό virtualization, λειτουργικά συστήματα, και το λογισμικό διαχείρισης που απαιτείται για το πρότυπο IaaS. Παρέχει όλο το υλικό της υποδομής και τα τμήματα λογισμικού που απαιτούνται για την υποστήριξη των διαφορετικών απαιτήσεων των εφαρμογών.

Υποστηρίζει συγκέντρωση των πόρων, ελαστική επεκτασιμότητα, ταχεία ανάπτυξη εφαρμογών και υψηλή διαθεσιμότητα. Η μοναδική ικανότητα να παραδίδονται εφαρμογές υποστηρίζοντας εγγενώς τις αρχές του virtualization και της διαχείρισης με ενσωματωμένες τεχνολογίες υπολογισμού, αποθήκευσης και δικτύωσης, επιτρέπει την ταχεία ανάπτυξη και την αποτελεσματική διαχείριση δημόσιων και ιδιωτικών IaaS.

Σύγκριση PaaS και IaaS

Για πολλές επιχειρήσεις το βασικό ζήτημα στην επιλογή μεταξύ της δημιουργίας ενός PaaS ή της χρησιμοποίησης ενός προσφερόμενου IaaS, είναι το κατά πόσον μια τυποποιημένη, επαναχρησιμοποιήσιμη και κοινόχρηστη (shared) πλατφόρμα θα κάνει αυτό που θέλουν οι

επιχειρήσεις να παρέχουν στους πελάτες τους. Ένα IaaS προσφέρει τις βασικές ικανότητες υπολογισμού, αποθήκευσης και δικτύωσης, με συνέπεια να είναι περισσότερο ευέλικτο.

Όμως το IaaS απαιτεί από τους χρήστες να παρέχουν τα υπόλοιπα, συμπεριλαμβανομένων των εφαρμογών, του middleware και των βάσεων δεδομένων, με αποτέλεσμα να υπάρχει μεγαλύτερο κόστος ανάπτυξης, χρόνου και ανομοιογένειας. Για πολλές επιχειρήσεις ένα ιδιωτικό PaaS είναι μια φυσική στρατηγική που ωφελεί τους χρήστες καθώς επίσης και τον πάροχο υπηρεσιών πληροφορικής.

Ένα PaaS δίνει στους χρήστες έναν τυποποιημένο, επαναχρησιμοποιήσιμο και κοινόχρηστο σημείο εκκίνησης για την ανάπτυξη εφαρμογών, παρέχοντας έναν γρηγορότερο και απλούστερο τρόπο ανάπτυξης εφαρμογών με επαρκή ευελιξία. Από την πλευρά των τμημάτων πληροφορικής η χρησιμοποίηση μιας υποδομής PaaS σημαίνει μεγαλύτερη διαχειριστικότητα, ασφάλεια, συνέπεια και έλεγχο.

Εξέλιξη των επιχειρήσεων στο cloud computing

Οι περισσότερες επιχειρήσεις εξελίσσουν την τρέχουσα υποδομή πληροφορικής τους για να υιοθετήσουν περισσότερα cloud χαρακτηριστικά με την πάροδο του χρόνου, αντί να μετακινήσουν άμεσα τα πάντα στο cloud. Η τεχνολογία εξελίσσεται και αναπτύσσεται ραγδαία, έχοντας σαν αποτέλεσμα οι επιχειρήσεις να πρέπει να αλλάξουν τις πολιτικές και τις διεργασίες τους. Σε πολλές περιπτώσεις θα πρέπει να είναι διαθέσιμες οι τεχνικές υποδομές των κτιρίων που στεγάζουν τις επιχειρήσεις, έτσι να μπορούν να γίνουν άμεσα οι απαραίτητες οργανωτικές αλλαγές που απορρέουν από την εφαρμογή των καινούργιων τεχνολογικών αλλαγών.

Για πολλά κέντρα δεδομένων κατά την μετάπτωση σε ένα ιδιωτικό cloud το πρώτο βήμα είναι η ενοποίηση, μεταβαίνοντας από αποκλειστικά περιβάλλοντα σε κοινόχρηστες και ελαστικά επεκτάσιμες πλατφόρμες και υποδομές. Οι εφαρμογές σιλό που λειτουργούν σε αποκλειστικό middleware, βάσεις δεδομένων, διακομιστές και αποθηκευτικά μέσα είναι σχεδιασμένες για ένα συγκεκριμένο φόρτο εργασίας, έτσι δεν υπάρχει εγγενώς πλεονάζουσα διαθεσιμότητα και πόροι που να μπορούν να χρησιμοποιηθούν για αυτές τις εφαρμογές. Κάθε σιλό μπορεί να περιλαμβάνει 12 ετερογενή συστήματα από πολλαπλούς προμηθευτές, οδηγώντας σε μεγάλη πολυπλοκότητα και υψηλά διαχειριστικά κόστη. Με την μετάβαση σε μια ενοποιημένη αρχιτεκτονική με τυποποιημένες, κοινόχρηστες υπηρεσίες, μπορεί να πραγματοποιηθεί σημαντική εξοικονόμηση κόστους.

Η ενοποίηση μπορεί να γίνει είτε στο επίπεδο του IaaS, συνήθως αξιοποιώντας την τεχνολογία των εικονικών διακομιστών (server virtualization technology), είτε στο επίπεδο του PaaS, μέσω της τυποποίησης και της εδραίωσης σε μια ενιαία βάση δεδομένων ή / και σε μια middleware αρχιτεκτονική. Η ενοποίηση στο επίπεδο του PaaS προσφέρει περισσότερη

αξία, επειδή μειώνει την ετερογένεια των χρησιμοποιούμενων λογισμικών, πράγμα το οποίο είναι ο πραγματική κινητήρια δύναμη της πολυπλοκότητας και του κόστους, πέρα από την αύξηση της αξιοποίησης των πόρων της τεχνολογίας πληροφορικής. Η ενοποίηση στο επίπεδο του IaaS μπορεί επίσης να προσφέρει υψηλότερη απόδοση με τον διαμοιρασμό του υλικού, αλλά δεν κάνει τίποτα για να μειώσει την πολυπλοκότητα των χρησιμοποιούμενων λογισμικών που εκτελούνται πάνω από το υλικό. Μια πρόσφατη έρευνα έδειξε ότι η ενοποίηση στο επίπεδο του middleware και της βάσης δεδομένων ήταν πιο δημοφιλής από την ενοποίηση στο επίπεδο του υπολογιστικού διακομιστή και των αποθηκευτικών μέσων. Ορισμένοι πελάτες βέβαια κάνει ενοποίηση και στα δύο επίπεδα.

Ενώ το virtualization είναι μια σημαντική τεχνολογία για το cloud computing, είναι επίσης σημαντικό να κατανοήσουμε ότι μια άλλη τεχνολογία που ονομάζεται ομαδοποίηση (clustering) είναι συμπληρωματική και επίσης πολύ σημαντική. Το virtualization είναι ένας πολύ καλός τρόπος για να διαμοιράζεται το υλικό και να επιτρέπεται η εύκολη ανάπτυξη εφαρμογών. Η ομαδοποίηση (clustering) είναι κρίσιμη για ενεργοποίηση της κλιμάκωσης πέρα από μια απλή φυσική μηχανή και για την ανοχή στα σφάλματα

Το επόμενο βήμα στην εξέλιξη είναι η προσθήκη αυτό-εξυπηρέτησης (self-service), αυτόματης κλιμάκωσης και χρέωσης. Ένας εργαζόμενος μπαίνει στην ηλεκτρονική πύλη των εργαζομένων, ταυτοποιείται επιτυχώς, κάνει αίτηση για ένα εικονικό μηχάνημα με ένα ορισμένο ποσό CPU, μνήμης και δίσκου, παίρνει μια εικόνα (image) μιας βάσης δεδομένων ή ενός middleware και στην συνέχεια πατάει «υποβολή». Εάν ο ρόλος και τα δικαιώματα αυτού του εργαζομένου του επιτρέπουν να έχει αυτό το ποσό πόρων πληροφορικής, τότε αυτά αυτόματα του τροφοδοτούνται χωρίς να εμπλακεί κάποιος διαχειριστής (administrator) από το τμήμα πληροφορικής (IT). Εάν όχι, τότε η αίτησή του μπορεί να δρομολογηθεί για έγκριση σε έναν διευθυντή (manager), είτε του τμήματος πληροφορικής (IT) είτε όχι.

Με την εφαρμογή αυτής της φαινομενικά απλής διαδικασίας θα είναι έτοιμο και θα λειτουργεί ένα στιγμιότυπο μιας PaaS Πλατφόρμας. Αφού αυτό το στιγμιότυπο έχει τεθεί στην παραγωγή, το σύστημα κάνει διαχείριση των πόρων με βάση μια πολιτική για την αυτόματη προσαρμογή της χωρητικότητας. Κάθε μήνα η επιχειρηματική μονάδα του εργαζομένου χρεώνεται με μια εσωτερική επιβάρυνση με βάση το ποσό των πόρων πληροφορικής που καταναλώνει αυτό το στιγμιότυπο. Για να γίνουν όλα αυτά πραγματικότητα θα πρέπει η επιχείρηση να έχει καθορισμένες πολιτικές και διαδικασίες, και η τεχνολογία θα πρέπει να είναι σε θέση να τις εφαρμόσει.

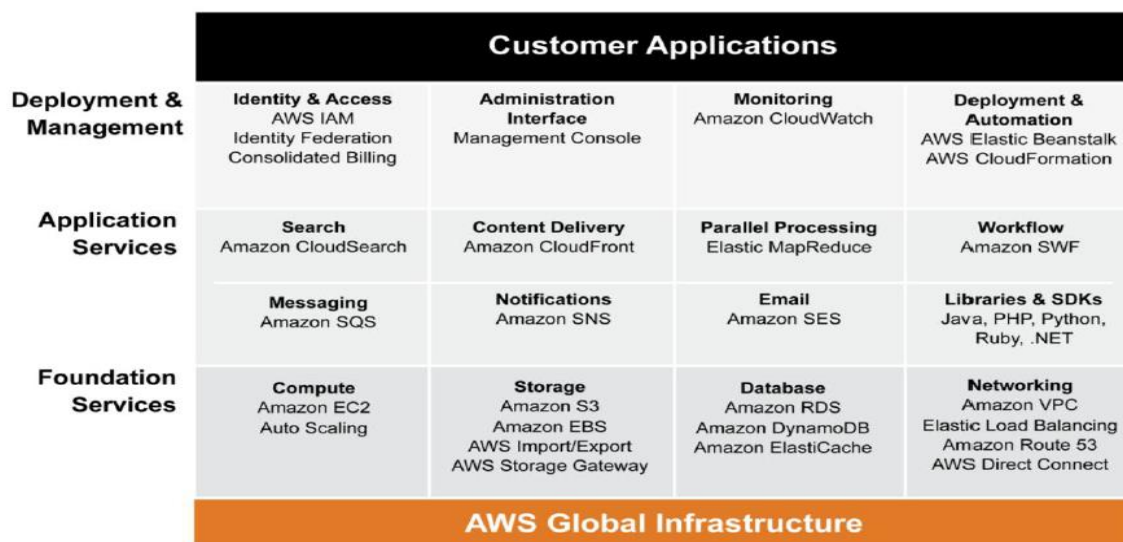
Πρέπει να επισημανθεί ότι δεν ωφελούνται όλες οι επιχειρήσεις από την αυτό-εξυπηρέτηση (self-service) και την ελαστική επεκτασιμότητα, αλλά πολλές όμως ωφελούνται, συνεπώς οι επιχειρήσεις το υπολογίζουν αυτό και κάνουν την κίνησή τους ή όχι προς αυτή την κατεύθυνση. Ορισμένες επιχειρήσεις δεν είναι έτοιμες να εφαρμόσουν πλήρως την αυτό-

εξυπηρέτηση, δεδομένου ότι πρέπει να καθοριστούν πολλές πολιτικές και διαδικασίες, και μπορεί να προτιμούν μοντέλα χρέωσης πληρωμής κατά την χρήση (pay-per-use). Επίσης, μπορεί να υπάρχουν και άλλες προκλήσεις όπως η δια-επιχειρησιακή οργανωτική υποστήριξη, η δημιουργία του επιχειρηματικού μοντέλου και του μοντέλου χρηματοδότησης, και διάφορα πολιτιστικά (cultural) θέματα.

2.3.1 Amazon EC2

Στα τέλη της δεκαετίας του 1990 και στις αρχές της δεκαετίας του 2000 η Amazon αναδυόταν ως η κορυφαία εταιρεία στο χώρο του ηλεκτρονικού εμπορίου (Perry, R., & Hendrick, S., 2012). Οι εσωτερικές επιχειρηματικές απαιτήσεις επέβαλλαν στην Amazon να χτίσει μια υποδομή εφαρμογών που θα υποστήριζε μαζική κλίμακα και αξιοπιστία στους ακόλουθους τομείς: υπολογισμοί, παράλληλη επεξεργασία, αποθήκευση, διαχείριση περιεχομένου, διαχείριση δεδομένων (σχεσιακές και μη σχεσιακές βάσεις δεδομένων), επεξεργασία συναλλαγών, ανταλλαγή μηνυμάτων, ουρές μηνυμάτων, πληρωμές, ασφάλεια, παρακολούθηση και διαχείριση.

Οι αντικρουόμενοι και συχνά ανταγωνιστικοί στόχοι που είχαν τεθεί για τις τεχνολογίες πληροφορικής που έπρεπε να χρησιμοποιηθούν και περιελάμβαναν την επεκτασιμότητα και τον έλεγχο του κόστους κατεύθυναν την Amazon στο δρόμο των υπηρεσιών. Οι υπηρεσίες που δημιουργήθηκαν μέσα σε αυτή την διαδικασία μετασχηματισμού της πληροφοριακής υποδομής έθεσαν τα οριστικά θεμέλια για τα AWS. Στην (Perry, R., & Hendrick, S., 2012) παρουσιάζονται οι βασικές υπηρεσίες που παρέχονται από τα AWS.



Εικόνα 3: Amazon Web Services (Perry, R., & Hendrick, S., 2012)

Ο προσανατολισμός της Amazon στην ελάττωση του κόστους από τις λειτουργίες σε μεγάλη κλίμακα για το ηλεκτρονικό εμπόριο, την οδήγησε προς τον προσανατολισμό στις υπηρεσίες και στην έκθεση όλων των πόρων της ως επεκτάσιμες και αναλώσιμες υπηρεσίες. Αυτή η κίνηση της Amazon επίσης εξασφάλισε ότι η κουλτούρα ανάπτυξης εφαρμογών θα είναι ευθυγραμμισμένη με τις σύγχρονες τεχνικές ανάπτυξης λογισμικού και αυτό είχε ως αποτέλεσμα μία πλατφόρμα που ήταν ευέλικτη, ευκίνητη και επεκτάσιμη.

Από την σκοπιά του πελάτη των AWS αυτή το API του δίνει την δυνατότητα να αναπτύξει ποιο εύκολα εφαρμογές που χρησιμοποιούν AWS, ποιο εύκολα να θέσουν σε παραγωγή εφαρμογές που προϋπήρχαν στα AWS και ποιο εύκολα να δημιουργήσουν υβριδικές εφαρμογές που διαμοιράζονται μεταξύ των AWS και ιδιωτικών κέντρων δεδομένων.

2.4 Σύγκριση των Επιπέδων Εξυπηρέτησης ανάμεσα σε

παρόχους υπηρεσιών YN

Ο κάθε χρήστης /πελάτης καλείται να αξιολογήσει τα προσφερόμενα επίπεδα εξυπηρέτησης των επώνυμων παρόχων και με βάση αυτά να προχωρήσει στη σύναψη μίας συνεργασίας με έναν ή περισσότερους από αυτούς:

SLA παράμετρος / πάροχος YN	Amazon EC2	Microsoft Azure Storage	Rackspace Cloud Servers	Dell Boomi	Google Cloud Storage
Κατηγορία υπηρεσιών	IaaS	PaaS	IaaS	SaaS	PaaS
Μέθοδος διασύνδεσης με την υποδομή του παρόχου	Χειροκίνητη διασύνδεση				
Εγγυημένος Χρόνος Διαθεσιμότητας	<99.95%	<99.9%	100% εξαιρώντας τις περιόδους συντήρησης	<99.9% Η υπηρεσία δεν είναι διαθέσιμη 1'/εβδομάδα για λόγους συντήρησης	>= 99.9%
Περιγραφή SLA	Προσδιορισμός εκ των προτέρων των όρων και προϋποθέσεων				
Σύμβαση	Κείμενο σύμβασης SLA που δίνεται από τον πάροχο				
Διαχείριση και Εποπτεία Υπηρεσίας	Επιτρέπεται η χρήση Συστημάτων Εποπτείας & Διαχείρισης Τρίτων υπό τους όρους της Amazon	Υπηρεσίες Διαχείρισης παρέχονται από τον πάροχο. Επιτρέπεται η χρήση Συστημάτων Εποπτείας & Διαχείρισης	Ο πάροχος πραγματοποιεί εκ των προτέρων εποπτεία της υποδομής, συντήρηση/ αναβάθμιση λειτουργικών	Ο πάροχος προσφέρει αναβαθμίσεις των συστημάτων, προγρ/μένη και έκτακτη συντήρηση	Επιτρέπεται η χρήση Συστημάτων Εποπτείας & Διαχείρισης Τρίτων

		Τρίτων	συστημάτων, και συντήρηση εφαρμογών		
--	--	--------	---	--	--

Πίνακας 3: κριτήρια αξιολόγησης των επιπέδων εξυπηρέτησης μεταξύ παρόχων (E. Aljoumah, F. Al-Mousawi et al, 2015)

2.5 Τιμολόγηση Επιπέδων Εξυπηρέτησης

Το συμφωνημένο επίπεδο εξυπηρέτησης μεταξύ του πελάτη και του παρόχου υπηρεσιών ΥΝ διαμορφώνει τη νόμιμη σχέση μεταξύ των δύο εμπλεκόμενων μερών. Η συμφωνία αυτή προστατεύει τα δικαιώματα και των δύο πλευρών κάτω από οποιαδήποτε περίσταση. Το συμφωνημένο επίπεδο εξυπηρέτησης παρέχει επομένως επαρκή γνώση σχετικά με τους πόρους των προσφερόμενων υπηρεσιών και χαρακτηριστικά όπως η ποιότητα και η τιμή των παρεχόμενων υπηρεσιών.

Η Διαχείριση του Επιπέδου Εξυπηρέτησης πέρα από τη διαχείριση των πολλών πόρων, περιλαμβάνει επίσης την τιμολόγηση. Στρατηγικές τιμολόγησης είναι οι τρόποι που εφαρμόζει ο εκάστοτε πάροχος για να καθορίσει το κόστος των υπηρεσιών σύμφωνα με την ζήτηση της κάθε υπηρεσίας και τον εξοπλισμό που απαιτείται για τη παροχή της υπηρεσίας. Οι πάροχοι υπηρεσιών χρησιμοποιούν την επιρροή τους και την επωνυμία (brand name) τους για να εφαρμόσουν ένα μηχανισμό έξυπνης τιμολόγησης που θα αυξήσει τα κέρδη τους. Υπάρχουν διάφορες στρατηγικές τιμολόγησης που καθορίζουν τις τελικές τιμές των υπηρεσιών.

Για παράδειγμα, οι χρεώσεις των υπηρεσιών μπορεί να υπολογιστούν με βάση την κορύφωση και αποκορύφωση της διεκπεραίωσης των αιτημάτων των χρηστών, τη ζήτηση για υπηρεσίες, τη διαθεσιμότητα της υπηρεσίας, και τις τιμές χρέωσης, ακόμη και αν είναι σταθερή ή δυναμική η παροχή υπηρεσιών. Οι περισσότεροι από τους παρόχους ΥΝ χρησιμοποιούν ένα από τα τρία βασικά μοντέλα τιμολόγησης τα οποία είναι: α) τιμή βάσει προσφοράς (bid model), β) σταθερές τιμές και γ) δυναμική τιμολόγηση. Οι πάροχοι φαίνεται να προτιμούν τη δυναμική τιμολόγηση επειδή πρέπει να αυξάνουν το κόστος της υπηρεσίας με σταθερό τρόπο για να μεγιστοποιήσουν τα έσοδα τους.

Δυναμική τιμολόγηση σημαίνει τη συνεχή μεταβολή των χρεώσιμων τιμών των υπηρεσιών. Οι τιμές συνεχώς μεταβάλλονται με βάση την προσφορά και τη ζήτηση των υπηρεσιών. Αν η ζήτηση για υπηρεσίες αυξάνεται ή η προσφορά των υπηρεσιών μειώνεται η τιμή θα ανέβει. Και εάν η ζήτηση για υπηρεσίες μειώνεται ή η παροχή υπηρεσιών αυξάνεται, η τιμή θα πάει κάτω.

Η δυναμική τιμολόγηση έχει μια επιρροή στη διαπραγμάτευση των επιπέδων εξυπηρέτησης μεταξύ των καταναλωτών και των παρόχων επειδή η τιμή μπορεί να αλλάξει κατά τη διάρκεια της διαπραγματευτικής διαδικασίας. Αλλά είναι σημαντικό να τονιστεί ότι όταν ένα επίπεδο εξυπηρέτησης έχει εγκριθεί, η τιμή της εν λόγω υπηρεσίας θα παραμείνει σταθερή καθόλη την υπόλοιπη διάρκεια εφαρμογής της σχετικής σύμβασης. Η διαμόρφωση ενός συμβολαίου εξυπηρέτησης και ειδικότερα η τιμή που καθορίζεται επηρεάζει τη συνεργασία του συγκεκριμένου παρόχου με τον συγκεκριμένο καταναλωτή-πελάτη. Η τιμή μπορεί να αλλάξει αν ο ίδιος πάροχος συνεργάζεται με διαφορετικούς πελάτες.

Η τιμή υπολογίζεται σύμφωνα με ορισμένα κριτήρια που μπορεί να είναι απλά ή σύνθετα, ανάλογα με τη διαφοροποίηση και τη ποσότητα των παραμέτρων. Απλά κριτήρια θα λάβουν υπόψη τους μερικές παραμέτρους. Ωστόσο σύνθετες λειτουργίες θα λάβουν υπόψη τους πολλές παραμέτρους. Αυτές οι παράμετροι μπορεί να είναι μετρήσεις σχετικά με την εσωτερική ή την εξωτερική κατάσταση του παρόχου. Περιπτώσεις που αντλούνται από τις εσωτερικές καταστάσεις περιλαμβάνουν τα τρέχοντα φορτία εργασιών στις προσφερόμενες υπηρεσίες και ιστορικά δεδομένα. Παράμετροι που αντλούνται από το εξωτερικό περιβάλλον σχετίζονται με την πραγματική κατάσταση της αγοράς, αλλά είναι δύσκολο να μετρηθούν.

Μια κρίσιμη παράμετρος στο σχεδιασμό της τιμολόγησης μίας υπηρεσίας σχετίζεται με τη χρήση της τρέχουσας υπηρεσίας. Ένας πάροχος είναι δύσκολο να εγγυηθεί τη διαθεσιμότητα των πόρων για τη μελλοντική παροχή υπηρεσιών και ζήτηση υπηρεσιών. Αυτός είναι ο λόγος για τον οποίο οι περισσότεροι από τους μεγαλύτερους παρόχους ΥΝ, όπως το Amazon EC2, Microsoft Azure, Dell Boomi, και Google Cloud εγγυώνται ένα επίπεδο διαθεσιμότητας της τάξης του 99,9%. Αυτό σημαίνει ότι ο μέγιστος επιτρεπόμενος χρόνος μη διαθεσιμότητας σε ετήσια βάση είναι 8 ώρες, 45 λεπτά και 55 δευτερόλεπτα. Ένα πιο αυστηρό προσφερόμενο επίπεδο διαθεσιμότητας (πχ 99,99%) σημαίνει πολύ λιγότερο χρόνο ανοχής για τη μη συμμόρφωσή τους.

Προσφερόμενο Επίπεδο Διαθεσιμότητας	Μέγιστος Επιτρεπόμενος Χρόνος μη Διαθεσιμότητας σε ετήσια βάση (ω-ώρες, λ-λεπτά, δ- δευτερόλεπτα)
99,9%	8ω 45λ 55δ
99,99%	52λ 35,7δ
99,999%	5λ 15,6δ
99,9999%	31,6δ
99,99999%	3,2δ

Πίνακας 4: Μέγιστος ετήσιος επιτρεπόμενος χρόνος Μη Διαθεσιμότητας υπηρεσιών (https://en.wikipedia.org/wiki/High_availability)

Μια άλλη βασική παράμετρος για τον υπολογισμό της τιμής των υπηρεσιών είναι ο κίνδυνος. Σε ορισμένες περιπτώσεις η συμφωνία επιπέδου υπηρεσιών στο ΥΝ έχει καθορισμένες προθεσμίες ή ακραίες υποχρεώσεις, ώστε οι διαπραγματευτές και από τις δύο πλευρές θα πρέπει να βρουν μια λύση για την κάλυψη της αστικής ευθύνης. Έτσι, μία ασφάλεια πριμοδότησης πρέπει να συμμετάσχει στην τιμή.

Το βασικό κόστος των υπηρεσιών, όπως του υλικού και λογισμικού που πρέπει να αγοραστεί, το κόστος αποθηκευτικού χώρου, και το κόστος συντήρησης είναι μια άλλη σημαντική παράμετρος των χρεώσεων. Ορισμένοι πάροχοι υπηρεσιών μπορεί να αντιμετωπίσουν επιχειρηματικά προβλήματα στο ξεκίνημα τους, και έτσι αποφασίζουν να προσφέρουν τις υπηρεσίες τους σε χαμηλότερο κόστος από το βασικό κόστος. Αλλά αυτό δεν μπορεί να διαρκέσει μακροπρόθεσμα, διότι το επιτυχημένο επιχειρηματικό μοντέλο θα πρέπει να είναι επικερδές.

2.6 Εργαλεία Συγκριτικής Αξιολόγησης Υπολογιστικών Νεφών

Έχοντας υπόψη τα παραπάνω οι χρήστες του ΥΝ μπορούν να ενεργοποιήσουν διάφορες υπηρεσίες ΥΝ κατ' απαίτηση και για όσο διάστημα χρειαστεί έτσι ώστε να πληρώνουν μόνο για τους πόρους και τον χρόνο χρήσης τους. Η αυξανόμενη υιοθέτηση των υπηρεσιών ΥΝ και η δυνατότητα διαπραγμάτευσης των χρεώσιμων τιμών προς όφελος των καταναλωτών εξαρτώνται από την ικανότητα των τελευταίων να μπορούν να πραγματοποιούν συγκριτικές αξιολογήσεις των προσφερόμενων υπηρεσιών (Ε. Κεβάνη, 2014).

Λαμβάνοντας υπόψη τις παραμέτρους και μετρικές αξιολόγησης της απόδοσης των προσφερόμενων υπηρεσιών ΥΝ που παρουσιάσαμε στην §2.2.2, η υλοποίηση και χρήση σχετικών εργαλείων μετρήσεων υπολογιστικών επιδόσεων (benchmarks). Τα benchmarks είναι νέα εργαλεία που θα προσφέρουν αξιολόγηση υποδομών ΥΝ βάσει προτύπου και καθορίζουν το ελάχιστο πλαίσιο αποδοχής μίας υποδομής καθώς και τα περιθώρια της βελτίωσης της.

Τα εργαλεία μετρήσεων υπολογιστικών επιδόσεων ΥΝ ελέγχει κατά πόσο μία προσφερόμενη υπηρεσία ΥΝ λειτουργεί βάσει του Συμφωνηθέντος Επιπέδου Εξυπηρέτησης (SLA), για παράδειγμα (L. Gillam, B. Li, J. O'Loughlin, 2012) :

- Έχει ο χρήστης στη διάθεση του το συμφωνημένο bandwidth; Αν όχι ποια είναι η απόκλιση έτσι ώστε να μπορεί να υπολογίσει την επίπτωση σε τυχόν μεταφορά δεδομένων ή αρχείων
- Έχουν οι εικονικές μηχανές που έχει δημιουργήσει τη συμφωνημένη μνήμη, επεξεργαστική ισχύ , ταχύτητα και χωρητικότητα δίσκου; Αν όχι, ποια είναι η

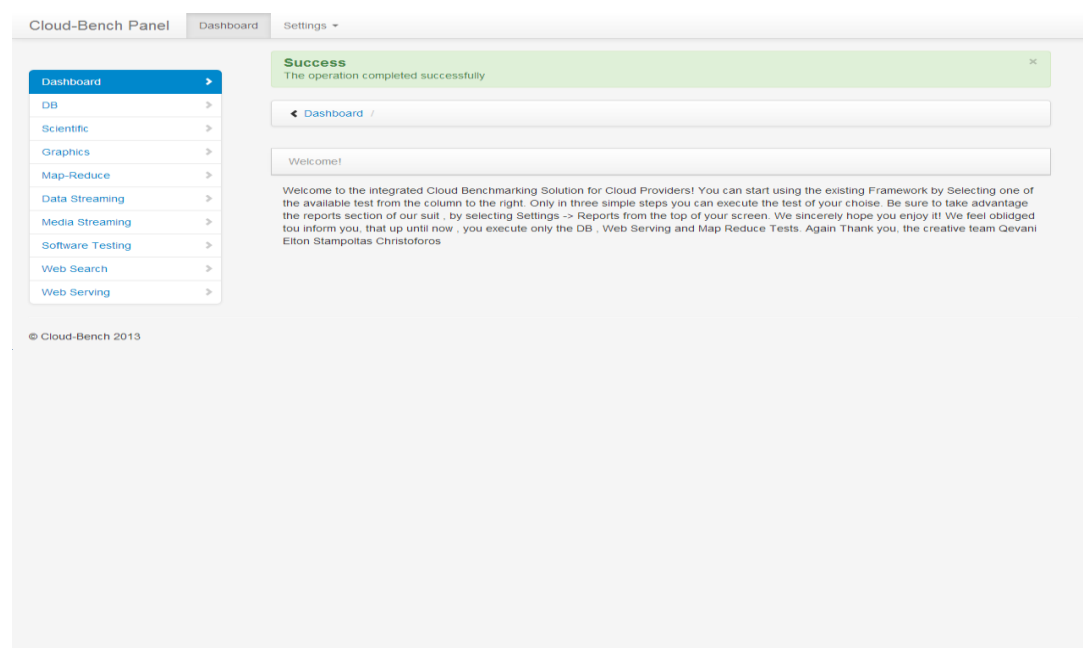
απόκλιση έτσι ώστε ο χρήστης να γνωρίζει αν μπορεί να εκτελέσει κάποιες νέες εφαρμογές και σε ποια χρονική στιγμή αυτό είναι εφικτό.

- Έχει ο χρήστης τη δυνατότητα να γνωρίζει ποιες εικονικές μηχανές είναι διαθέσιμες και ποιες όχι; Μπορεί να πραγματοποιήσει μία συγκριτική αξιολόγηση μεταξύ επιλεγμένων εικονικών μηχανών (εφόσον διαθέτουν τα ίδια λειτουργικά χαρακτηριστικά, για παράδειγμα την ίδια εφαρμογή ή βάση δεδομένων);
- Οι χρήστες έχουν στη διάθεση τους γραφικές αναπαραστάσεις για την διευκόλυνση της αξιολόγησης των αποτελεσμάτων;

2.6.1 Cloud Benchmark tool

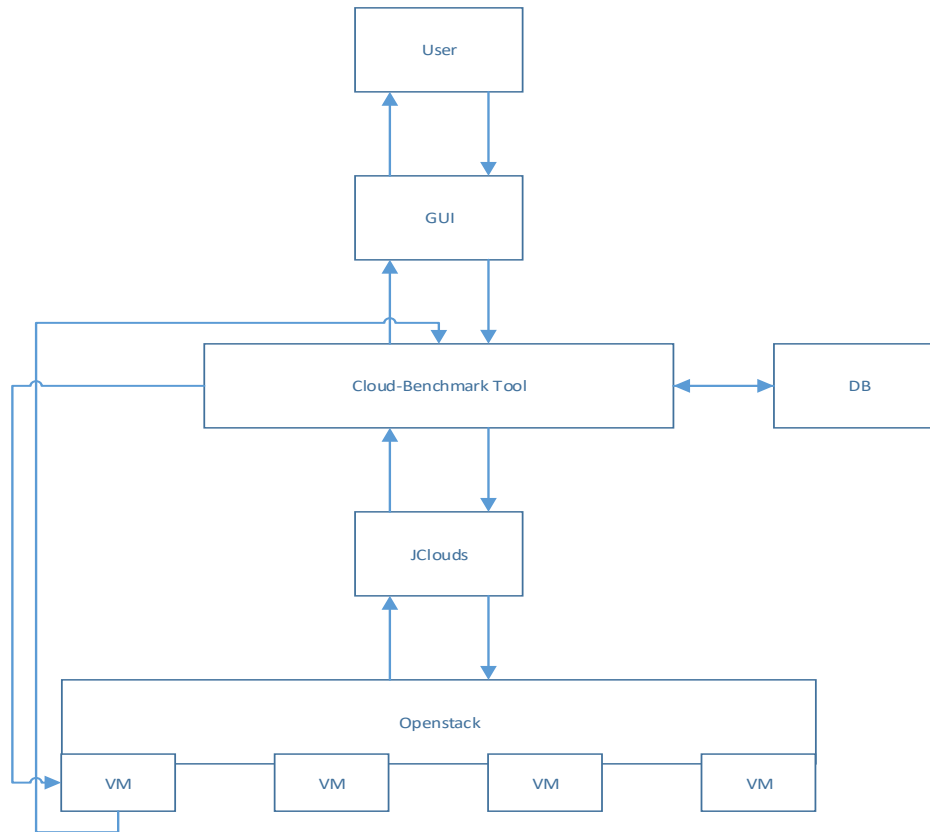
Το Cloud Benchmark tool είναι ένα εργαλείο μετρήσεων υπολογιστικών επιδόσεων ΥΝ (Ε. Κεβάνη, 2014). Το εργαλείο διαθέτει ένα κοινό γραφικό περιβάλλον από το οποίο μπορεί να εκτελέσει συγκεκριμένες δοκιμές αξιολόγησης.

Με την επιλογή κάποιας δοκιμής, το εργαλείο λαμβάνει όλα τα δεδομένα από το χρήστη και καλεί τη προγραμματιστική βιβλιοθήκη JClouds της Apache για την δρομολόγηση της δοκιμής στον αντίστοιχο πάροχο ΥΝ. Η βιβλιοθήκη αναλαμβάνει τη δημιουργία της απαιτούμενης εικονικής μηχανής και κατόπιν όλη μεταγενέστερη επικοινωνία γίνεται μηχανής και του εργαλείου. Στην εξέλιξη της δοκιμής οι μετρήσεις παρουσιάζονται στην οθόνη του χρήστη και ταυτόχρονα αποθηκεύονται σε μία βάση δεδομένων για την εξαγωγή αναφορών.



Εικόνα 1: Cloud Benchmark Tool – γραφικό περιβάλλον (Ε. Κεβάνη, 2014)

Ο χρήστης έχει την ευελιξία της ταυτόχρονης εκτέλεσης πολλαπλών δοκιμών (ανοίγοντας διαφορετικές σελίδες της εφαρμογής) ενώ είναι δυνατόν να ανατρέξει σε ιστορικό προηγούμενων δοκιμών.



Εικόνα 2: *Cloud Benchmark Tool* (X. Σταμπολτάς, 2014)

Οι κύριες δοκιμές που μπορεί ο χρήστης να εκτελέσει μέσα από το γραφικό περιβάλλον του εργαλείου είναι (X. Σταμπολτάς, 2014): α) DB, β) Web Serving, γ) Map Reduce.

DB

Επιτρέπει τη συγκριτική αξιολόγηση εικονικών μηχανών που εκτελούν την ίδια βάση δεδομένων. Ο χρήστης επιλέγει τη βάση δεδομένων του σεναρίου:

- MongoDB
- MySQL
- Postgre
- Cassandra

Και κατόπιν ορίζει τον αριθμό των νημάτων (threads) για την προσομοίωση της ταυτόχρονης εξυπηρέτησης πολλαπλών αιτημάτων και το υποτιθέμενο τρέχον φόρτο (workload) των εικονικών μηχανών:

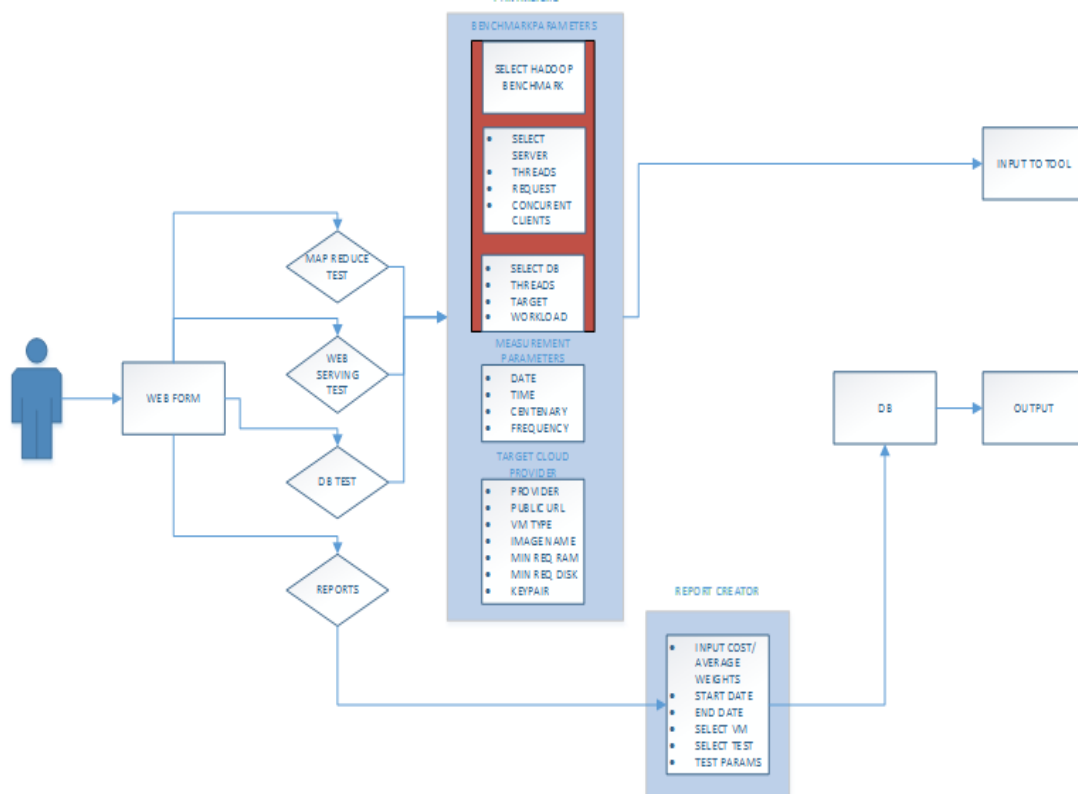
- Χαμηλός φόρτος (Light Workload)
- Μεσαίος φόρτος (Medium Workload)
- Υψηλός φόρτος (Heavy Workload)
- Πολύ υψηλός φόρτος (Very Heavy Workload)

Map Reduce

Σκοπός της δοκιμής αυτής είναι η αξιολόγηση της απόδοσης μίας εγκατάστασης Map Reduce με βάση την πραγματοποίηση τυποποιημένων δοκιμών (benchmarks) όπως: TestDFSIO, NNBench, Teragen, MrBench.

Web Serving

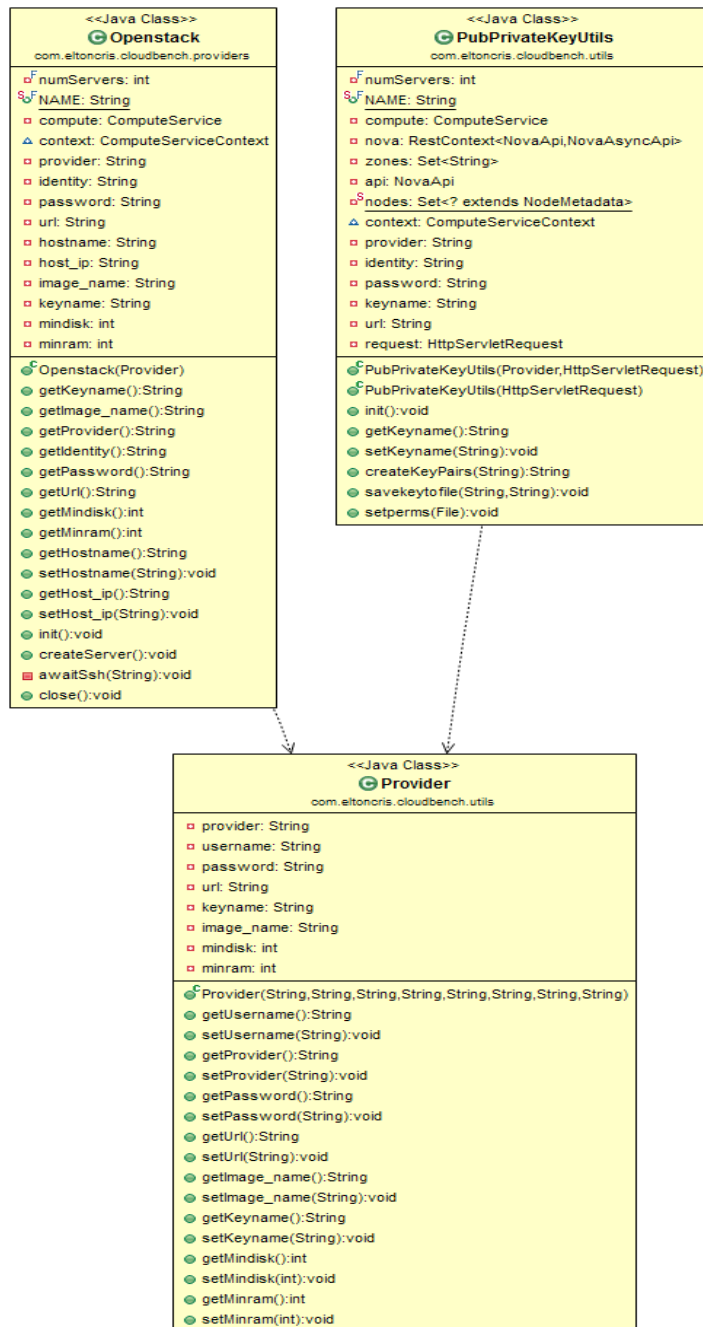
Σκοπός της δοκιμής αυτής είναι η αξιολόγηση της απόδοσης Web διακομιστών. Ο χρήστης επιλέγει τον web server και ορίζει τις παραμέτρους του σεναρίου όπως ο αριθμός των νημάτων (threads), αριθμός ταυτόχρονων αιτημάτων και πελατών.



Εικόνα 3: Cloud Benchmark Tool δοκιμές (Χ. Σταμπολτάς, 2014)

Σε όλες τις παραπάνω δοκιμές ο χρήστης προσδιορίζει τις παραμέτρους των μετρήσεων (Measurement Parameters), κυρίως την ημερομηνία και ώρα έναρξης του τεστ, τη συχνότητα εκτέλεσης και τον πάροχο ο οποίος συμμετέχει στο σενάριο (Target Cloud Provider). Οι διαθέσιμοι πάροχοι είναι :

- Rackspace
- Openstack
- Amazon WS
- VCloud



Εικόνα 4: Cloud Benchmark Tool – κύριες κλάσεις (X. Σταμπολλάς, 2014)

Στο παραπάνω διάγραμμα παρουσιάζονται οι τρεις κλάσεις που υλοποιούν τη δημιουργία του εικονικού μηχανήματος και της ενημέρωσης του με τις απαραίτητες εφαρμογές για την υλοποίηση της σύγκρισης. Η κλάση `provider` αναλαμβάνει τη σύνδεση με τον πάροχο υπηρεσιών ΥΝ έχοντας ως τιμές εισόδου το όνομα του παρόχου, τους κωδικούς πρόσβασης και τον τύπο του εικονικού μηχανήματος που πρέπει να αρχικοποιηθεί. Η `Openstack` κλάση έχει όλη την απαιτούμενη πληροφορία για την αρχικοποίηση του μηχανήματος στην υποδομή του παρόχου. Τέλος αναλαμβάνει η κλάση `PubPrivateKeyUtils` τη κρυπτογραφημένη ανταλλαγή αρχείων μεταξύ του εικονικού μηχανήματος και του υπολογιστή η οποία εξασφαλίζεται με την ανταλλαγή δημοσίων κλειδιών μεταξύ των δύο μηχανημάτων.

2.7 Επέκταση του Cloud Benchmark Tool: 3ALib

Η βιβλιοθήκη `Abstract Availability Auditor (3ALib)` έχει υλοποιηθεί στη γλώσσα προγραμματισμού `Java` και παρέχει ένα αφηρημένο μηχανισμό λήψης μετρήσεων σχετικά με τη διαθεσιμότητα υπηρεσιών ΥΝ, ανεξάρτητα από το ποιο πάροχο υποστηρίζονται ή αναπτύσσονται. Επιπλέον, η βιβλιοθήκη είναι σε θέση να υπολογίσει τα πραγματικά επίπεδα διαθεσιμότητας, όπως ορίζεται στα `SLAs` των αντίστοιχων παρόχων. Η `3ALib` χρησιμοποιεί τη προγραμματιστική βιβλιοθήκη `Jclouds` της `Apache` ως μέσο για να αποκρύψει τις ιδιαιτερότητες των βιβλιοθηκών που παρέχονται από τους ίδιους τους παρόχους και τις ιδιαιτερότητες σχετικά με τη μορφή των εκάστοτε δεδομένων. Επιπλέον, υλοποιεί ειδικές κλάσεις-οδηγούς για κάθε πάροχο, προκειμένου να ενεργοποιήσουν εξ αρχής τις συγκεκριμένες προϋποθέσεις που αναφέρονται στις αντίστοιχες συμβάσεις για το Επίπεδο Εξυπηρέτησης (`SLAs`) ανά πελάτη και τη συγκεκριμένη μέθοδο υπολογισμού της διαθεσιμότητας που ακολουθείται από κάθε πάροχο (`ARTIST Consortium, 2016`).

Λαμβάνοντας υπόψη ότι ο κάθε πάροχος υπηρεσιών ΥΝ εφαρμόζει το δικό του ορισμό σχετικά με τον όρο διαθεσιμότητα στα Προσφερόμενα Επίπεδα Εξυπηρέτησης (βλ. §2.4), όταν ένας χρήστης υποψιάζεται ότι το επίπεδο διαθεσιμότητας ενός παρόχου, όπως ορίζεται στο `SLAs` τους, δεν εκπληρώνεται χρειάζεται να εξασφαλίζει αποδείξεις ότι ο πάροχος δεν τηρεί τη σύμβαση που έχει συναφθεί. Αντιμετωπίζοντας ως πρόκληση τη παραπάνω ανάγκη, δημιουργήθηκε η βιβλιοθήκη `3ALib`.

Οι στόχοι και προκλήσεις που έχει να αντιμετωπίσει η βιβλιοθήκη είναι:

1. να ενημερώνει τους χρήστες σχετικά με το γιατί οι υπηρεσίες τους δεν συμμορφώνονται με το Συμφωνημένο Επίπεδο Εξυπηρέτησης (`SLA`)

2. οι χρήστες να τηρούν λεπτομερή αρχεία καταγραφής της διαθεσιμότητας των υπηρεσιών και
3. να μπορούν να τον υπολογίζουν τα επίπεδα διαθεσιμότητας με βάση τον ορισμό των παρόχων και να αξιολογούν αποζημίωση όταν κρίνουν απαραίτητο και έχοντας τις σχετικές αποδείξεις μέσω των προαναφερθέντων αρχείων καταγραφής.

Οι βασικότερες καινοτομίες της βιβλιοθήκης περιλαμβάνουν τα εξής:

- Είναι μία βιβλιοθήκη αφηρημένων κλάσεων και μεθόδων, που σημαίνει ότι μόνο μία μέθοδος εκτίθεται στο χρήστη για να αντλήσει τις μετρήσεις επιπέδων SLA των υπηρεσιών τους ανεξάρτητα από τον συνεργαζόμενο πάροχο. Οι τεχνικές διαφορές μεταξύ των προγραμματιστικών βιβλιοθηκών των παρόχων αποκρύπτονται μέσω της χρήσης της βιβλιοθήκης Apache JClouds . Αυτό διευκολύνει σημαντικά τη χρήση της βιβλιοθήκης.
- Προσαρμόζεται με τον ορισμό του SLA του κάθε παρόχου, μια πολύ σημαντική πτυχή για κάποιον πελάτη προκειμένου να διεκδικήσει αποζημίωση από τον αντίστοιχο πάροχο. Αυτοί οι ορισμοί έχουν σημαντικές διαφορές και επιμέρους όρους και προϋποθέσεις, οι οποίες ελέγχονται ανά πάροχο στη βιβλιοθήκη, μέσω της εφαρμογής αντίστοιχων κλάσεων -οδηγών.
- Καταγράφει τα δείγματα διαθεσιμότητας που απαιτούνται ως αποδεικτικά στοιχεία προς τους παρόχους για ενδεχόμενη μη συμμόρφωση και υπολογίζει τα επίπεδα διαθεσιμότητας (σύμφωνα με τον ορισμό του παρόχου) από τα δείγματα που καταγράφει.
- Μπορεί να καθοδηγήσει τους χρήστες ως προς το κατά πόσον η χρήση τρεχουσών υπηρεσιών που είναι διαθέσιμες στο ΥΝ είναι συμβατές με συμβάσεις SLA των παρόχων, και τους συμβουλεύει στο τι απαιτείται για να γίνουν συμβατές.
- Μπορεί να χρησιμοποιηθεί για να συγκεντρωθούν σε κεντρικό σημείο (π.χ. βάση δεδομένων) στατιστικά στοιχεία από πολλούς χρήστες σχετικά με τη συμπεριφορά των υπηρεσιών ΥΝ, επιτρέποντας έτσι τη συσσώρευση στατιστικών στοιχείων και την υλοποίηση σεναρίων ανάλυσης δεδομένων και επιχειρησιακής νοημοσύνης (Business Intelligence analytics).

Λαμβάνοντας υπόψη τα παραπάνω χαρακτηριστικά της βιβλιοθήκης 3ALib, σκοπός της παρούσας πτυχιακής εργασίας είναι η επέκταση του εργαλείου Cloud Benchmark Tool με τις λειτουργίες και τα καινοτόμα χαρακτηριστικά της βιβλιοθήκης. Αυτό διευκολύνεται σημαντικά από:

1. Την αρχιτεκτονική του εργαλείου Cloud Benchmark Tool η οποία έχει υλοποιηθεί σε τεχνολογία Spring MVC Framework που είναι επέκταση της γλώσσας προγραμματισμού Java για την ανάπτυξη εφαρμογών στο διαδίκτυο. Η βιβλιοθήκη 3ALib είναι άμεσα προσβάσιμη από τις κλάσεις του εργαλείου.
2. Την ήδη υπάρχουσα παραμετροποίηση που ακολουθείται στη γραφική διεπαφή του εργαλείου σχετικά με: α) την επιλογή παρόχου για την εκτέλεση μίας δοκιμής και των σχετικών κωδικών ταυτοποίησης για την απόκτηση πρόσβασης στην υποδομή, β) την επιλογή της βάσης δεδομένων για τη καταγραφή των στατιστικών στοιχείων διαθεσιμότητας και των σχετικών κωδικών ταυτοποίησης για την απόκτηση πρόσβασης στην βάση δεδομένων. Τα παραπάνω γραφικά συστατικά μπορούν να επαναχρησιμοποιηθούν για την ταχεία επέκταση της γραφικής διεπαφής με τη λειτουργικότητα της βιβλιοθήκης 3ALib.
3. Το γεγονός ότι το εργαλείο Cloud Benchmark Tool χρησιμοποιεί ήδη τη βιβλιοθήκη Apache JClouds για την επικοινωνία με τις προγραμματιστικές βιβλιοθήκες των παρόχων.

Στη παραπάνω ενότητα, αναλύουμε την αρχιτεκτονική της υλοποίησης της βιβλιοθήκης 3ALib και τη μεθοδολογία που εφαρμόσαμε για την ενσωμάτωση της στο Cloud Benchmark Tool.

3

Αρχιτεκτονική 3ALib – Cloud Benchmark tool

3.1 Εισαγωγή – μεθοδολογία και πλαίσιο εργαλείων ARTIST

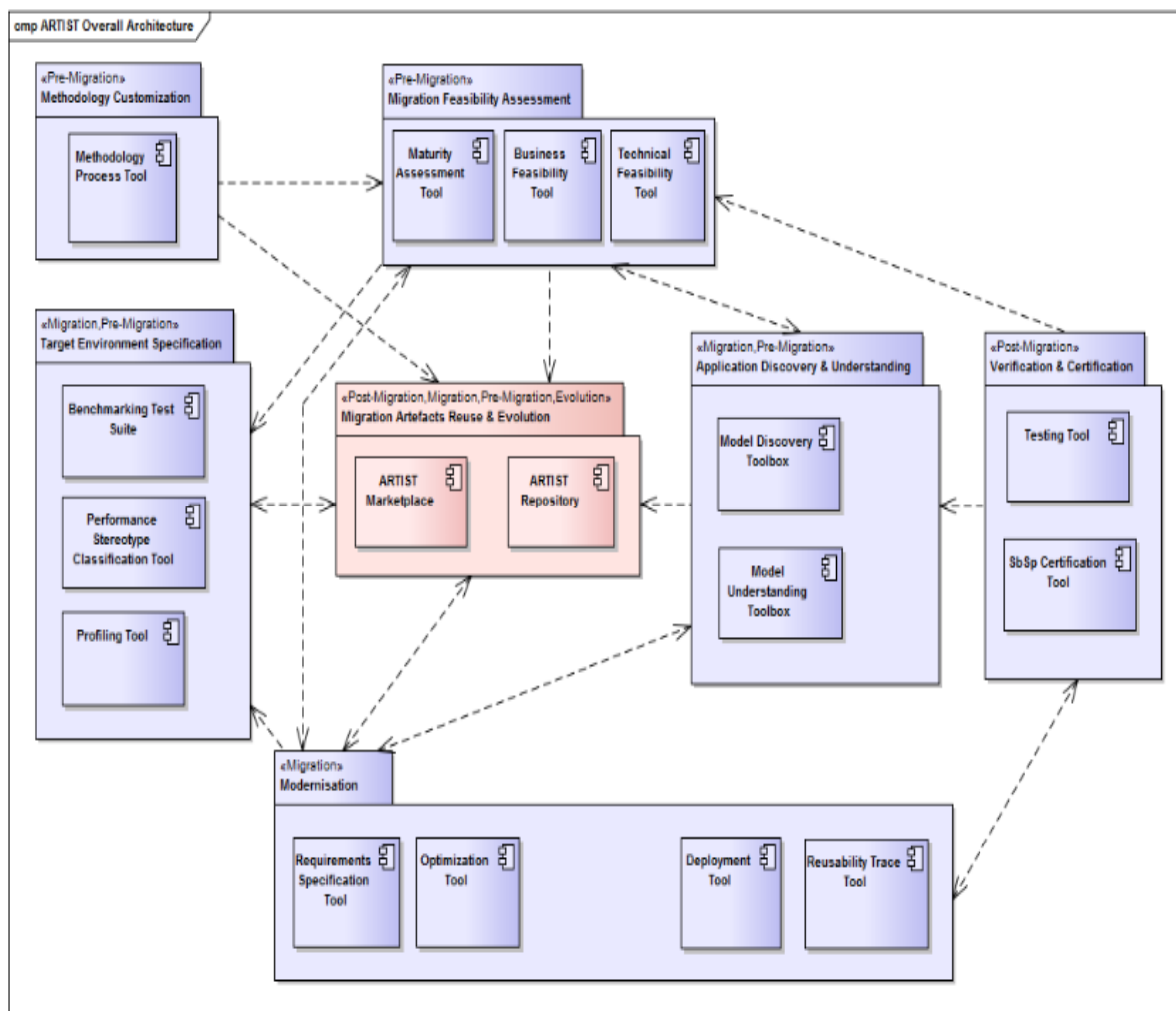
Η βιβλιοθήκη 3ALib είναι προϊόν ενός ερευνητικού προγράμματος της Ευρωπαϊκής Ένωσης για την προετοιμασία, υποστήριξη και την αύξηση της ανταγωνιστικότητας της ευρωπαϊκής βιομηχανίας λογισμικού και υπηρεσιών σε μία παγκόσμια αγορά υπηρεσιών υπολογιστικού νέφους και λογισμικού ως υπηρεσία (SaaS). Το Ευρωπαϊκό Πρόγραμμα με την ονομασία ARTIST αναπτύσσει μια σειρά από μεθόδους, εργαλεία και τεχνικές που διευκολύνουν τη μετατροπή και τον εκσυγχρονισμό των προγραμμάτων λογισμικού που έχουν ήδη δημιουργηθεί από τις επιχειρήσεις για τη μετάπτωση τους στο Υπολογιστικό Νέφος (ARTIST Consortium, 2016).

Το συγκεκριμένο Ευρωπαϊκό Πρόγραμμα δημιουργεί εργαλεία για την αξιολόγηση, σχεδιασμό, υλοποίηση και την επικύρωση της αυτοματοποιημένης εξέλιξη του μη-συμβατού με το ΥΝ λογισμικού σε ένα νέο παραδοτέο που είναι συμβατό με το μοντέλο Software-as-a-Service (SaaS) και το μοντέλο της Νεφο-Υπολογιστικής. Με έμφαση στην επαναχρησιμοποίηση κατά τη διάρκεια αυτής της μετάβασης, οι μέθοδοι και τα εργαλεία είναι αρκετά γενικευμένες ώστε να καλύπτουν και τις όποιες μελλοντικές προσπάθειες μετάπτωσης, π.χ. μετάβαση σε νέες πλατφόρμες που θα αναπτυχθούν στο μέλλον. Βασικότεροι στόχοι του είναι οι (Γ. Τ. Κουσιουρής, 2012):

1. Να αναπτύξει ένα εργαλείο συγκριτικής αξιολόγησης για την αξιολόγηση της ωριμότητας της τεχνολογίας και το επιχειρηματικό μοντέλο ενός συγκεκριμένου προϊόντος λογισμικού, συμπεριλαμβανομένου του ελέγχου των περιορισμών της τεχνολογίας που περιορίζουν την αξία του προϊόντος από την πλευρά των τεχνικών επιδόσεων, χαρακτηριστικών, διαλειτουργικότητας, συναφών δαπανών και επιλογών σχετικά με την υλοποίηση.
2. Να αναπτύξει ένα σύνολο εργαλείων που θα επιτρέψουν στις εταιρείες να αξιολογήσουν με ακρίβεια τη σκοπιμότητα, το κόστος, τις επιπτώσεις και τα οφέλη της μετάβασης από το μη-συμβατό με το ΥΝ λογισμικό σε νέα πλαίσια και πλατφόρμες (πχ SaaS), ειδικότερα όταν απαιτούνται επιπρόσθετα και μη-τεχνικές αλλαγές (πχ στο επιχειρηματικό μοντέλο) (A. Menychtas, K. Konstanteli et al, 2014).
3. Να ορίσει, αναπτύξει, δοκιμάσει, μία προσαρμόσιμη μεθοδολογία (με τη βοήθεια εργαλείου) για να βοηθήσει τις επιχειρήσεις μέσω μιας διαδικασίας που εκτελείται σε φάσεις, να πραγματοποιήσουν τη μετάπτωση του λογισμικού τους σε νέα πλαίσια και πλατφόρμες (πχ Cloud, SaaS) λαμβάνοντας υπόψη τις νέες ιδιότητες που διατίθεται μέσω αυτής της μετάπτωσης, για παράδειγμα σχεδιασμός και υλοποίηση νέας υπηρεσιο-κεντρικής αρχιτεκτονικής (SOA), η ασφάλεια, φιλοξενία πολλαπλών πελατών (multi-tenancy), η παράλληλη εξυπηρέτηση πολλαπλών αιτημάτων, η επεκτασιμότητα και ελαστικότητα, η απόδοση, αξιοπιστία, φορητότητα και η ανεξαρτησία από κατασκευαστές. Σημαντικό ρόλο παίζουν επίσης ο προσδιορισμός ενός πλαισίου Επιπέδου Εξυπηρέτησης, Ποιότητας (QoS), κα.

3.2 Σουίτα Εργαλείων ARTIST

Η βιβλιοθήκη εργαλείων ARTIST είναι δομημένη σε ενότητες οι οποίες αντιστοιχούν στις κύριες φάσεις της μεθοδολογίας ARTIST: στάδιο προ-μετάπτωσης, μετάπτωση, και στάδιο μετά την μετάπτωση. Τα πιο σημαντικά εργαλεία της βιβλιοθήκης παρουσιάζονται στη παρακάτω εικόνα. Τα διακεκομμένα βέλη αντιπροσωπεύουν τις εξαρτήσεις μεταξύ των εργαλείων (A. Menychtas, K. Konstanteli et al, 2014).



Εικόνα 5: Αρχιτεκτονική της Βιβλιοθήκης Εργαλείων της 3ALib (ARTIST Consortium, 2016)

Εργαλεία Προσδιορισμού του Περιβάλλοντος Ενδιαφέροντος

Το συγκεκριμένο πακέτο εργαλείων (Target Environment Specification Package) χρησιμοποιείται σε δεύτερο χρόνο για να παρέχει τα μοντέλα με τα βασικά χαρακτηριστικά και υπηρεσίες που προσφέρουν οι διάφοροι πάροχοι υπηρεσιών υπολογιστικού νέφους. Ο σκοπός αυτών των μοντέλων είναι διπλός (J. Goroñoigoitia, 2016):

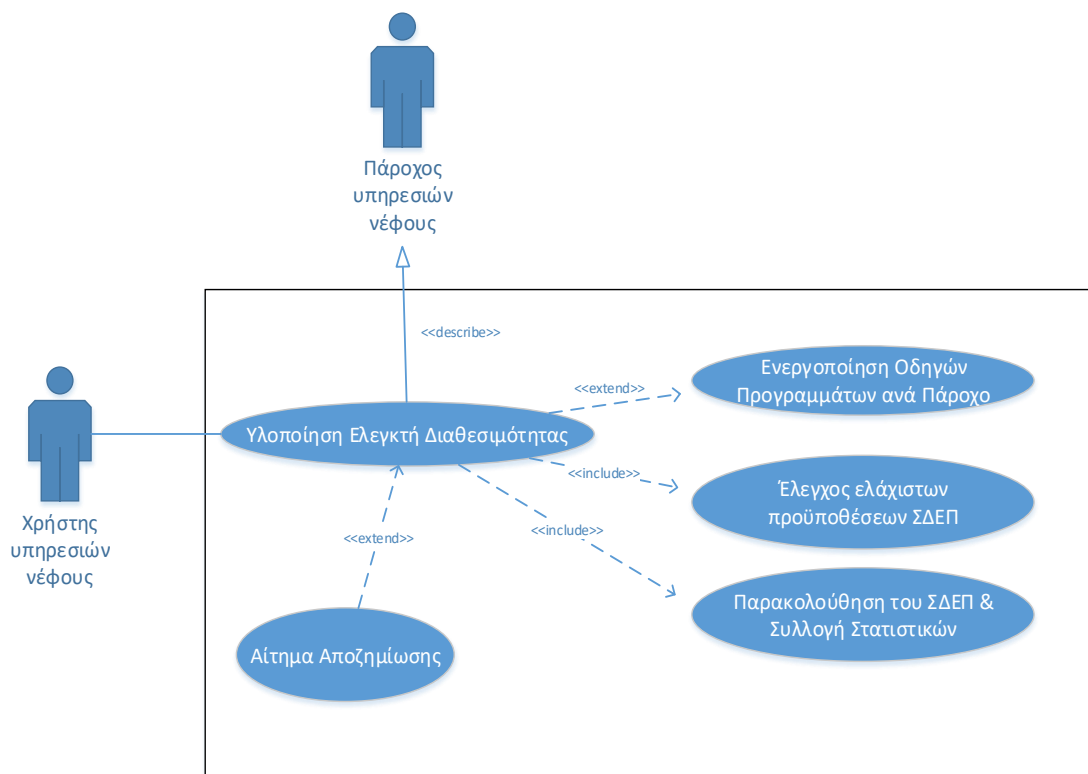
- οι χρήστες της βιβλιοθήκης μπορούν να ορίσουν τις απαιτήσεις τους προς τους εναλλακτικούς παρόχους πριν τη μετάπτωση των υπηρεσιών και υποδομών στο υπολογιστικό νέφος
- βάσει του μοντέλου, η συμπεριφορά και απόδοση των εφαρμογών και υπηρεσιών ταξινομείται και αξιολογείται. Η μέση απόδοση κατηγοριών εφαρμογών είναι διαθέσιμη στους χρήστες.
- έχοντας τα παραπάνω δεδομένα, οι χρήστες συνάπτουν συγκεκριμένο συμβόλαιο παροχής υπηρεσιών με ένα ή περισσότερους από τους παρόχους. Το συμβόλαιο

μπορεί να περιλαμβάνει συγκεκριμένες δεσμεύσεις από τη πλευρά του παρόχου (SLA) αλλά και συγκεκριμένες προϋποθέσεις από τη πλευρά του χρήστη.

3.2.1 Περιπτώσεις Εφαρμογής (Use cases)

Οι κυριότερες περιπτώσεις εφαρμογής είναι οι παρακάτω (A. Menychtas, K. Konstanteli et al, 2014):

1. Ενεργοποίηση συγκεκριμένων οδηγών προγραμμάτων και του Συμβολαίου Διασφάλισης Επιπέδου Ποιότητας (ΣΔΕΠ –SLA) ανά πάροχο υπηρεσιών υπολογιστικού νέφους (cloud service provider)
2. Έλεγχος ελάχιστων προϋποθέσεων τήρησης του ΣΔΕΠ από τη πλευρά του πελάτη
3. Παρακολούθηση του ΣΔΕΠ
4. Αίτημα αποζημίωσης - Επικοινωνία με τον πάροχο του υπολογιστικού νέφους για βελτίωση των υπηρεσιών και τήρηση του συμφωνημένου ΣΔΕΠ
5. Τροφοδότηση πληροφοριών και στατιστικών από πολλαπλούς χρήστες σχετικά με το επίπεδο ποιότητας του κάθε παρόχου



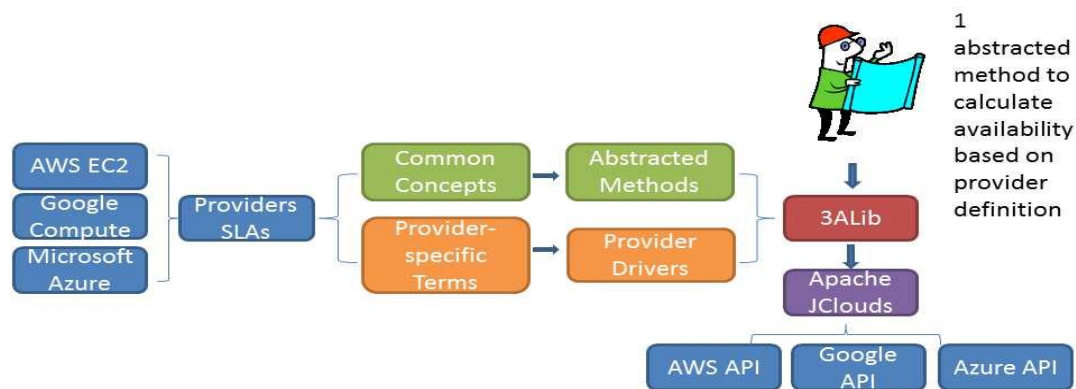
Εικόνα 6: Περιπτώσεις Χρήσης της βιβλιοθήκης 3ALib (J. Gorroñoigoitia, 2016)

3.3 3ALib Αρχιτεκτονική – Διαγράμματα Κλάσεων και

Ακολουθίας

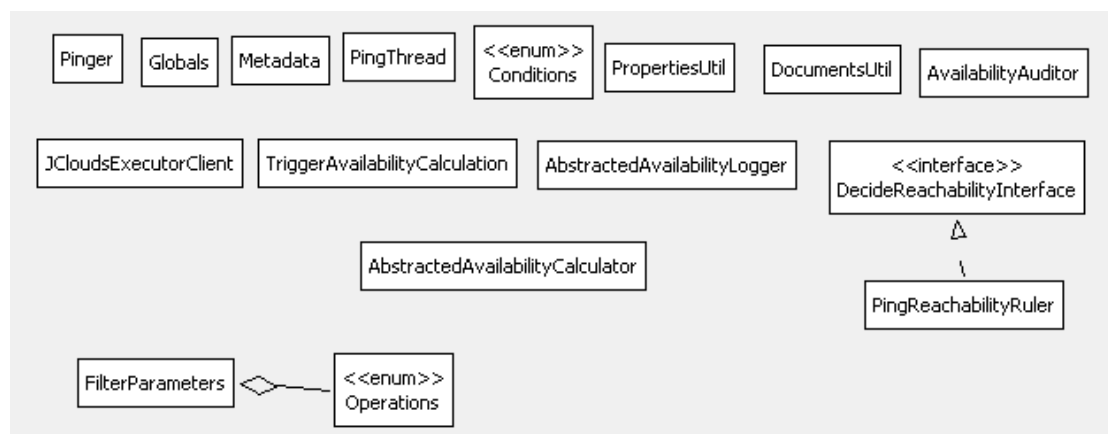
Αποτέλεσμα της παραπάνω επιστημονικής προσπάθειας μέσα από το Ευρωπαϊκό Πρόγραμμα ARTIST είναι η βιβλιοθήκη 3ALib (βλ. 2.7).

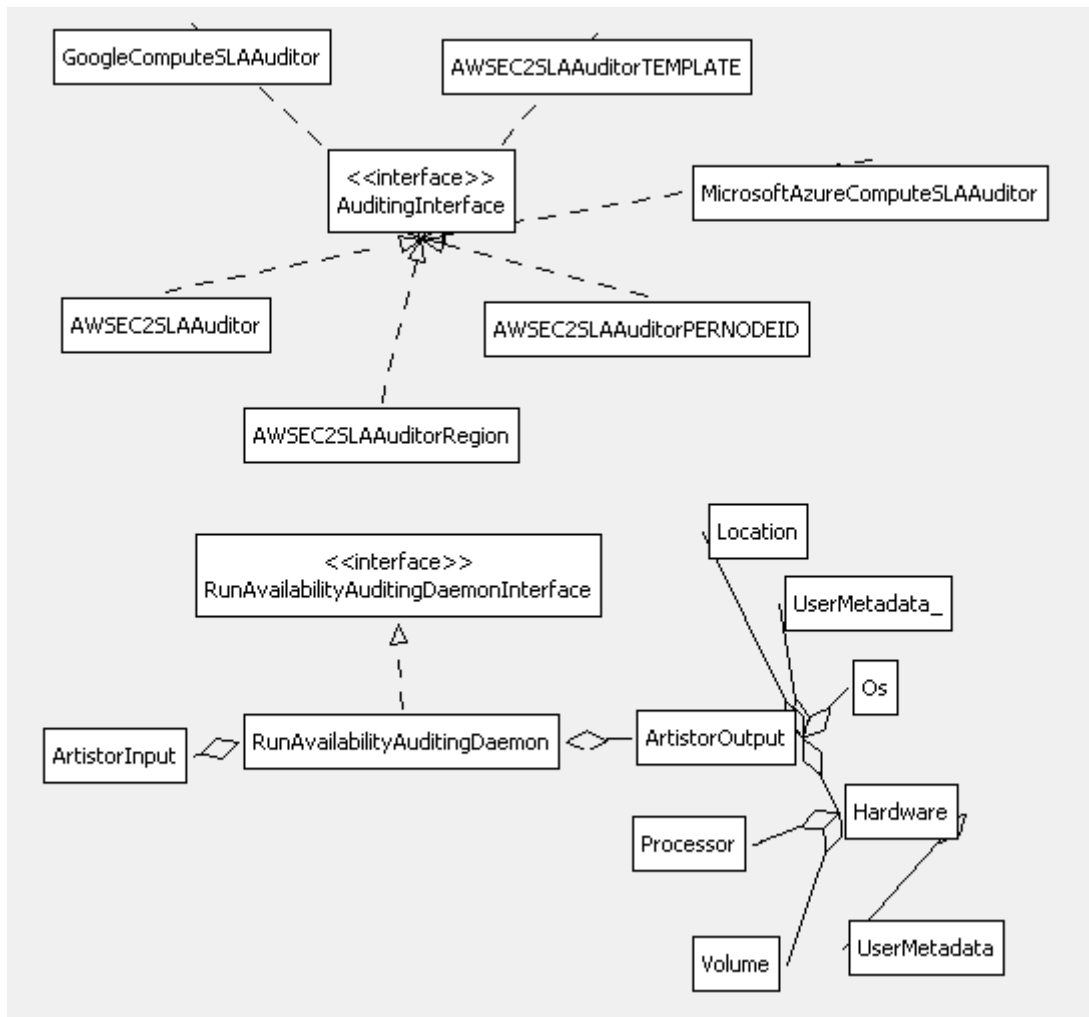
Η βασική αρχιτεκτονική της βιβλιοθήκης 3ALib φαίνεται στην Εικόνα 7. Μετά την ανάλυση των ΣΔΕΠ (SLAs) των διαθέσιμων παρόχων, οι κοινές έννοιες και όροι που αντλούνται διαμορφώνουν γενικές μεθόδους οι οποίες χρησιμοποιούνται από όλους τους παρόχους. Ειδικές περιπτώσεις αντιμετωπίζονται από ξεχωριστές μεθόδους.



Εικόνα 7: Βασική Αρχιτεκτονική της βιβλιοθήκης 3ALib (ARTIST Consortium, 2016)

Διάγραμμα κλάσεων

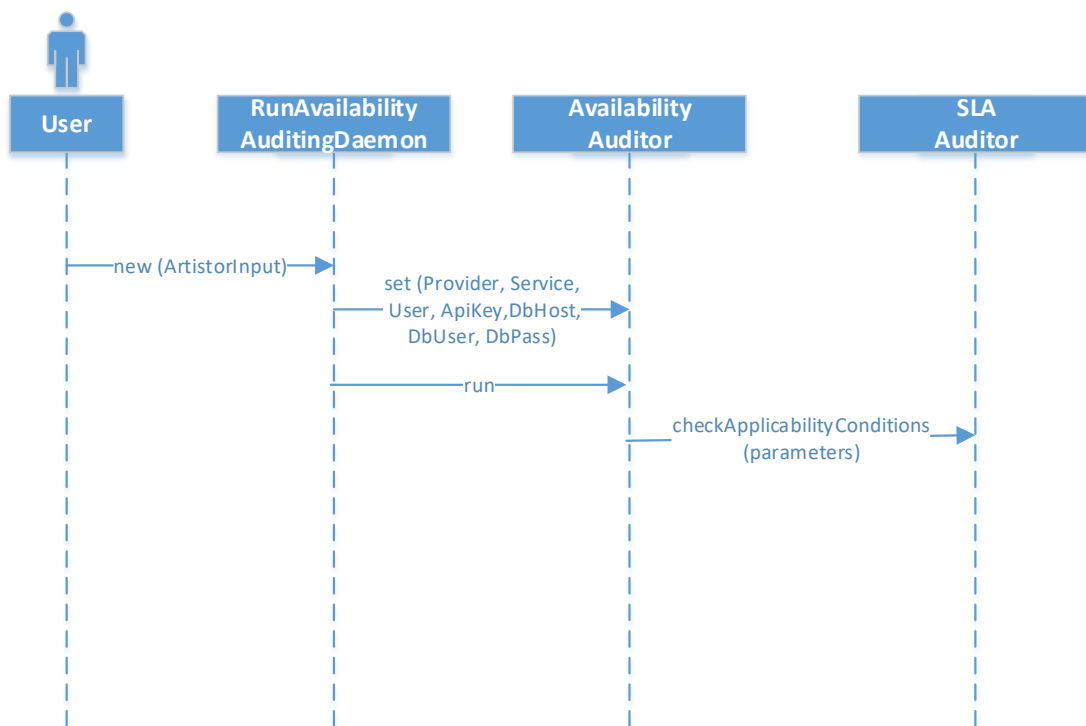




Εικόνα 8: Διάγραμμα κλάσεων της Βιβλιοθήκης 3ALib και επεκτάσεις της εφαρμογής μας

Η κλάση *Availability Auditor* χρησιμοποιεί έναν από τους ελεγκτές παρόχου που επιλέγεται από τον χρήστη. Οι επιμέρους ελεγκτές υλοποιούν την ίδια διεπαφή (*AuditingInterface*) για σκοπούς συνοχής, ωστόσο, υλοποιούν το σώμα των μεθόδων με βάση τη λογική κάθε αντίστοιχου παρόχου και του ΣΔΕΠ που τηρεί. Χρησιμοποιούν επίσης αφηρημένες μεθόδους, όπως η *JcloudsExecutorClient* (χρησιμοποιείται για την ανάκτηση πληροφοριών από τους παρόχους σχετικά με τις υπηρεσίες προς τους χρήστες και το καθεστώς τους, με βάση τη λειτουργικότητα του *Apache Jclouds*), και η *AbstractedAvailabilityLogger*. Η τελευταία είναι υπεύθυνη για να πάρει τη λεπτομερή έκθεση για την κατάσταση και την αποθήκευση της. Επιπλέον, είναι υπεύθυνη για την απόκρυψη των διαφορών / διακυμάνσεων στα αναφερόμενα μηνύματα κατάστασης από παρόχους τα οποία γενικεύει σε μία κατάσταση: ΠΡΟΣΒΑΣΙΜΗ (REACHABLE) / ΜΗ ΠΡΟΣΒΑΣΙΜΗ (NOT REACHABLE).

Τα παραπάνω απεικονίζονται στο παρακάτω διάγραμμα ακολουθίας (Εικόνα 12). Οι βασικές παράμετροι του χρήστη ορίζονται σε ένα αντικείμενο της κλάσης *ArtistorInput*.



Εικόνα 12: Διάγραμμα Ακολουθίας – Αρχικοποίηση Διεργασίας (SLA) Availability Auditing ανά Πάροχο (ARTIST Consortium, 2016)

Ορίζονται οι παράμετροι:

- Provider: ο πάροχος της υπηρεσίας υπολογιστικού νέφους (για παράδειγμα aws είναι ο κωδικός του παρόχου Amazon)
- Service: κωδικός των υπηρεσιών που διατίθενται από τον παραπάνω πάροχο (για παράδειγμα ec2 είναι ο κωδικός των υπηρεσιών Elastic Compute του παρόχου Amazon). Το ζεύγος Provider- Service προσδιορίζει μοναδικά μια υπηρεσία
- User: ο κωδικός χρήστη/συνδρομητή για τη πρόσβαση στην υπηρεσία
- ApiKey: ο κωδικός πρόσβασης ή μοναδικό κλειδί ταυτοποίησης του χρήστη για τη πρόσβαση στην υπηρεσία
- DbHost: IP διεύθυνση της βάσης δεδομένων στην οποία θα καταγράφονται τα μεταδεδομένα που θα λαμβάνονται σχετικά με τη διαθεσιμότητα των υπολογιστικών πόρων στη συνδεδεμένη υπηρεσία
- DbUser: ο κωδικός χρήστη για τη πρόσβαση στην βάση δεδομένων
- DbPass: ο κωδικός πρόσβασης για τη πρόσβαση στην βάση δεδομένων

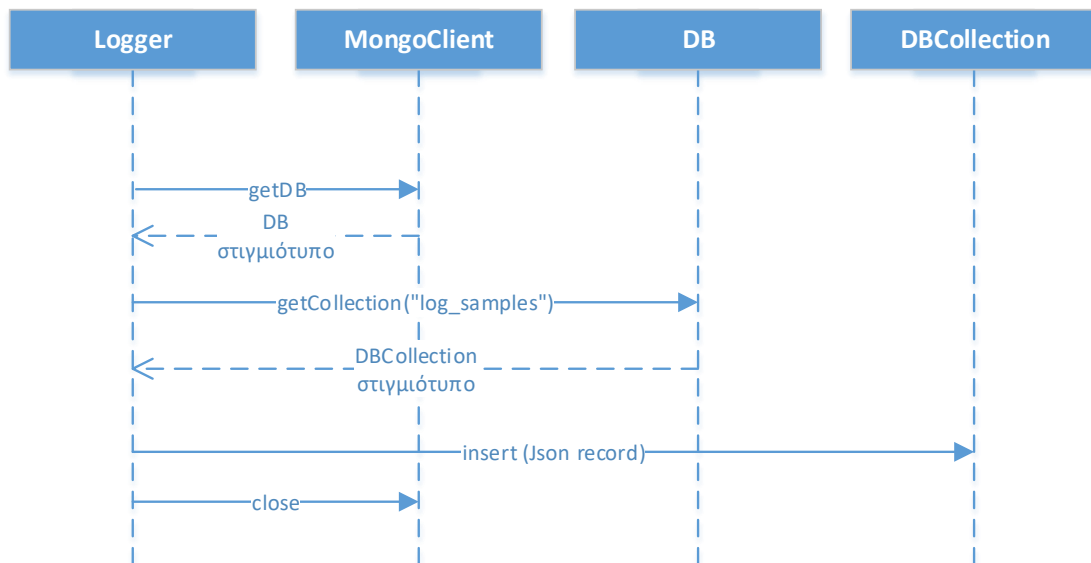
Η κλάση `RunAvailabilityAuditingDaemon` ξεκινά τη διεργασία του Ελέγχου Διαθεσιμότητας Πόρων (κλάση `AvailabilityAuditor`). Η διεργασία αυτή επιλέγει τον κατάλληλο ελεγκτή (κλάση `SLAAuditor`) για τον υπολογισμό και έλεγχο της τήρησης του ΣΔΕΠ (SLA). Για παράδειγμα, η κλάση `AWSEC2SLAAuditor` θα χρησιμοποιηθεί στη περίπτωση του παρόχου Amazon (`checkApplicabilityConditions`).

Με τη παραπάνω δομή η κλάση `RunAvailabilityAuditingDaemon` μπορεί να ξεκινήσει πολλές φορές είτε από τον ίδιο χρήστη ή από πολλούς ταυτόχρονους χρήστες. Επιπλέον, με την εκτέλεση της κάθε διεργασίας `Availability Auditing` μπορούν να ελέγχονται ταυτόχρονα είτε υπηρεσίες του ίδιου παρόχου ή υπηρεσίες εναλλακτικών παρόχων.

Ο τελικός υπολογισμός των επιπέδων τήρησης ΣΔΕΠ (SLA) γίνεται από την κλάση `AbstractedAvailabilityCalculator`, η οποία ορίζεται σε ένα αφαιρετικό επίπεδο αφού όλοι οι πάροχοι υλοποιούν τη κλάση αυτή και ακολουθούν μια παρόμοια φόρμουλα για τον υπολογισμό της διαθεσιμότητας, αλλά με μια σειρά διαφοροποιημένων παραμέτρων. Για παράδειγμα, το ελάχιστο συνεχόμενο χρονικό διάστημα για το οποίο ένας πόρος μπορεί να θεωρηθεί μη διαθέσιμος είναι ένα τέτοιο παράδειγμα. Αυτές οι παράμετροι ως επιχειρήματα, που μπορεί να αντληθεί και να χρησιμοποιηθεί για όλες τις περιπτώσεις.

Ο ελεγκτής των επιπέδων τήρησης ΣΔΕΠ (SLA) για τον εκάστοτε πάροχο καλεί τη βασική κλάση `JCloudsExecutorClient` για τον υπολογισμό της διαθεσιμότητας αναφορικά με μια σειρά υπολογιστικών μεταδεδομένων. Αυτά συσσωρεύονται στην οντότητα `Service Context` (Node metadata). Αφού ληφθούν τα μεταδεδομένα από τη λίστα των διαθέσιμων πόρων στη συνέχεια ελέγχεται η προσβασιμότητα και υπολογίζονται επιπλέον παράμετροι (`check reachability`). Κατόπιν τα μεταδεδομένα πρέπει να καταγραφούν στη βάση δεδομένων.

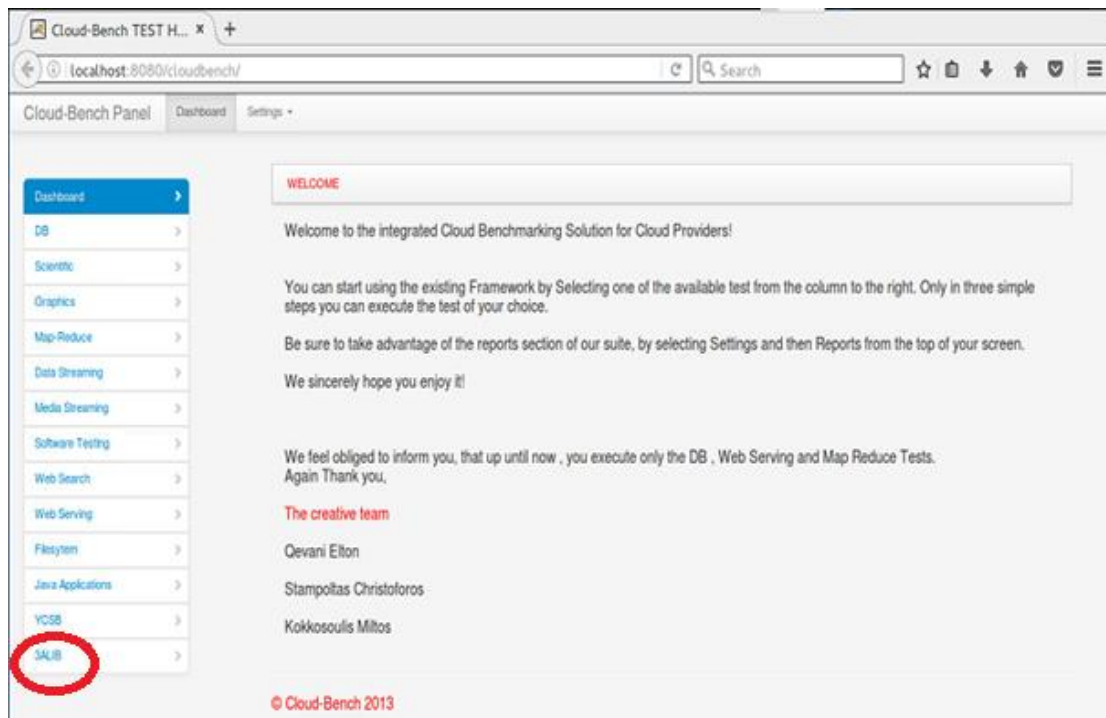
Η αποθήκευση δεδομένων βασίζεται σε μία βάση δεδομένων MongoDB στο πίσω μέρος backend που μπορεί να δημιουργηθεί από την πλευρά του χρήστη, ή μπορεί να διατεθεί από μία ανάλογη υπηρεσία σε ένα εναλλακτικό σενάριο. Ο σκοπός της χρήσης μίας λύσης NoSQL για την αποθήκευση και διαχείριση των δεδομένων είναι η επεκτασιμότητα της και ότι μπορεί να χρησιμοποιηθεί για τη παράλληλη επεξεργασία των καταγραφών (logs), το οποίο συνήθως είναι μια χρονοβόρα διαδικασία. Πάνω σε αυτά τα δεδομένα μπορούν να υλοποιηθούν περαιτέρω ερωτήματα πολλαπλών επιπέδων (analytics) με βάση προηγμένα σενάρια επιχειρησιακής νοημοσύνης (business intelligence).



Εικόνα 13: Διάγραμμα Ακολουθίας – Αποθήκευση Υπολογιστικών Μεταδεδομένων στη Βάση Δεδομένων (MongoDB) (J. Gorroñoigoitia, 2016)

3.4 Διασύνδεση 3ALib με την Πλατφόρμα Cloud Benchmark tool

Cloud Benchmark tool είναι η υφιστάμενη πλατφόρμα για τη πρόσβαση σε διαφορετικές υπηρεσίες υπολογιστικού νέφους και την εκτέλεση συγκεκριμένων διεργασιών, λήψη αποτελεσμάτων ερευνητικού σκοπού και άλλα. Για παράδειγμα, ο χρήστης έχει πρόσβαση σε απομακρυσμένη βάση δεδομένων, κόμβο της βιβλιοθήκης Map-Reduce, απομακρυσμένη διαχείριση αρχείων, εκτέλεση Java εφαρμογών και άλλα. Η λίστα των διαθέσιμων υπηρεσιών παρέχεται στη κεντρική ιστοσελίδα της πλατφόρμας. Στο τέλος της λίστας των διαθέσιμων απομακρυσμένων εργασιών έχει προστεθεί σύνδεσμος για τη πρόσβαση στην βιβλιοθήκη 3ALIB.



Εικόνα 14: *Cloud Benchmark tool*

Οι χρήστες επιλέγοντας το σύνδεσμο «3ALIB» μπορούν να ενεργοποιήσουν μία νέα διεργασία Ελεγκτή Διαθεσιμότητας (Availability Auditing) μέσω της κλάσης *RunAvailabilityAuditingDaemon*. Συνεπώς, ο χρήστης καλείται να ορίσει τις παραμέτρους της κλάσης *ArtistInput* που τροφοδοτούν την διεργασία *RunAvailabilityAuditingDaemon*.

1. Ορίζεται η βάση δεδομένων στην οποία θα καταγράφονται τα μεταδεδομένα που θα λαμβάνονται από τη συνδεδεμένη υπηρεσία. Αυτή θα είναι η *mongodb* που έχει εγκατασταθεί στον τοπικό κόμβο στον οποίο εκτελείται η πλατφόρμα *Cloud Benchmark tool*.
2. Δεν απαιτείται (στη παρούσα έκδοση) ο ορισμός κωδικού χρήστη και κωδικού πρόσβασης.

3ALIB Benchmarking

Benchmark Parameters Measurement Parameters Target Cloud Provider

Select Database: MongoDB

Load Defaults Test Button

IP: 127.0.0.1

User:

Key:

Εικόνα 15: παραμετροποίηση βάσης

Επιλέγοντας «Target Cloud Provider» ο χρήστης ορίζει τον πάροχο της υπηρεσίας υπολογιστικού νέφους. Βάσει συγκεκριμένης επιλογής ορίζεται αυτόματα και η υπηρεσία (Service) που επιλέγεται από τον συγκεκριμένο πάροχο. Ανάλογα με το ζεύγος Provider-Service το προσδιορίζει μοναδικά μια υπηρεσία επιλέγονται οι αντίστοιχες παράμετροι που θα επιτρέψουν τη σύνδεση στην υπηρεσία. Στη περίπτωση της υπηρεσίας aws-ec2 (Amazon WS) απαιτούνται οι παρακάτω παράμετροι:

- Access Key: το δημόσιο κλειδί συνδρομητή για τη πρόσβαση στην υπηρεσία
- Secret Key: το μυστικό κλειδί για τη πρόσβαση στην υπηρεσία

Benchmark Parameters Measurement Parameters Target Cloud Provider

Provider: Amazon WS

AccessKeyId

SecretKey

Public URL:

VM Type: m1.small

Image Name:

Minimum Required Ram: 1700

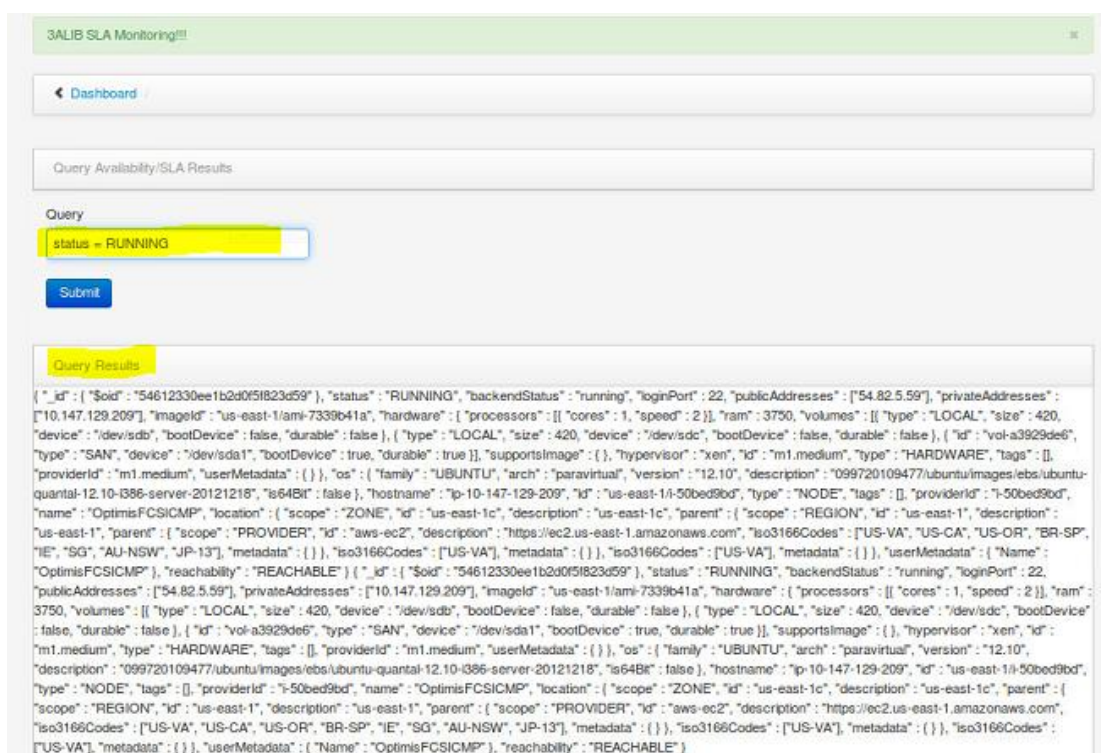
Minimum Required Disk Space: 160

KeyPair: --

Εικόνα 16: ορισμός παραμέτρων παρόχου

Δηλώνοντας τις παραμέτρους ξεκινά η εκτέλεση της διεργασίας *RunAvailabilityAuditingDaemon*. Στη φάση της παρακολούθησης (monitoring) τα μεταδεδομένα που λαμβάνονται αποθηκεύονται στη βάση δεδομένων και είναι διαθέσιμα στις υπόλοιπες υπηρεσίες και εφαρμογές της πλατφόρμας Cloud Benchmark tool, για παράδειγμα στη διαθέσιμη βιβλιοθήκη SLA Reporting. Ο χρήστης μπορεί να διακόψει την παρακολούθηση ανά πάσα στιγμή. Επιπλέον, μπορεί να ξεκινήσει μία νέα διεργασία *RunAvailabilityAuditingDaemon* ανοίγοντας μία νέα ιστοσελίδα Cloud Benchmark tool. Συνεπώς εξυπηρετούνται ταυτόχρονα αιτήματα πολλαπλών χρηστών.

Σε επόμενη σελίδα ο χρήστης μπορεί να παρακολουθεί τα αποτελέσματα που αποθηκεύονται στη βάση δεδομένων κάνοντας ερωτήματα προς τη mongodb. Για παράδειγμα η ερώτηση για το πλήθος των υπολογιστικών μηχανών οι οποίες εκτελούνται αυτή τη στιγμή (status = RUNNING) επιστρέφει τα παρακάτω αποτελέσματα.

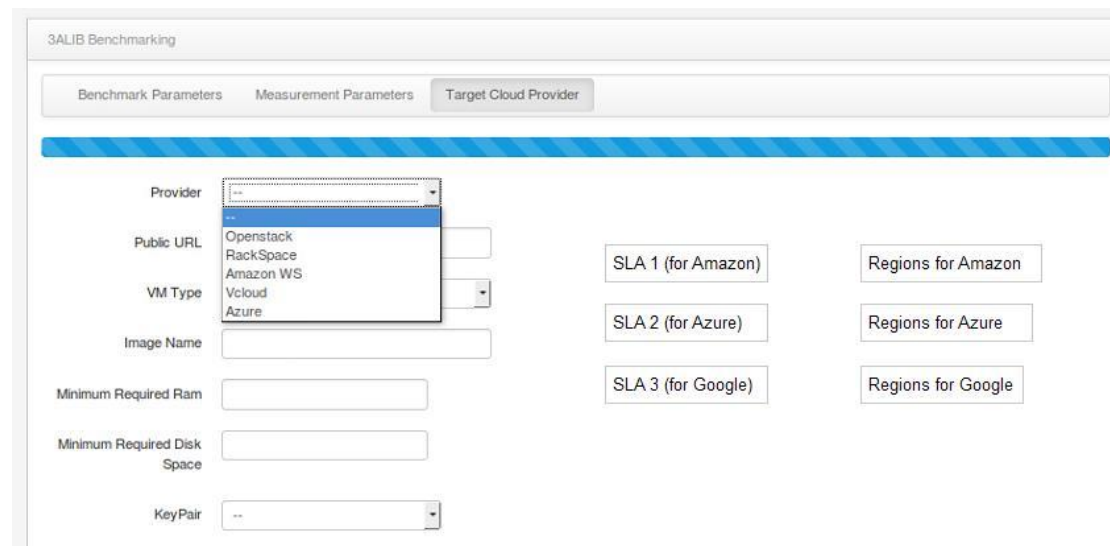


Εικόνα 17: εκτέλεση ερωτημάτων προς τη βάση

Η σύνταξη των ερωτημάτων ακολουθεί τους παρακάτω κανόνες:

1. Ερώτημα με ένα όρισμα: π.χ. [τιμή A] = 3, [τιμή B] > 3, κλπ
2. Ερώτημα με πολλαπλά ορίσματα τα οποία ενώνονται μεταξύ τους με λογική *σύζευξη*: π.χ. [τιμή A] = 3 AND [τιμή B] > 3, [τιμή A] < 3 AND [τιμή B] < 3μ
3. Ερώτημα με πολλαπλά ορίσματα τα οποία ενώνονται μεταξύ τους με λογική *διάζευξη*: π.χ. [τιμή A] = 3 OR [τιμή B] > 3, [τιμή A] < 3 OR [τιμή B] < 3μ

Η υλοποίηση έχει πραγματοποιηθεί στην μέθοδο **findDocuments** της κλάσης **DocumentsUtil**. Στη κλάση αυτή ορίζονται οι κωδικοί πρόσβασης στη βάση δεδομένων (IP διεύθυνση, κωδικός χρήστη, κωδικός πρόσβασης), το ερώτημα σύμφωνα με τη παραπάνω σύνταξη και το όνομα του αρχείου στο οποίο θα αποθηκευτούν τα αποτελέσματα. Το όνομα του αρχείου είναι δυναμικό και σχετίζεται με τον κωδικό της τρέχουσας συναλλαγής του χρήστη με την εφαρμογή. Ο κωδικός αυτός (session id) είναι μοναδικός ανά χρήστη. Κατά την εκτέλεση του ερωτήματος στον ελεγκτή *ThreeALibController* η εμφάνιση των αποτελεσμάτων γίνεται στη σχετική view (success.jsp). Η τελευταία διαβάζει τα περιεχόμενα του αρχείου το οποίο σχετίζεται με τον μοναδικό κωδικό της τρέχουσας συναλλαγής χρήστη και τα προβάλλει στην οθόνη.



Εικόνα 18: Γραφικό περιβάλλον για την επιλογή του κάθε παρόχου

3.5 Μεθοδολογία Διασύνδεσης 3ALib με την Πλατφόρμα

Cloud Benchmark tool

3.5.1 Spring web MVC framework

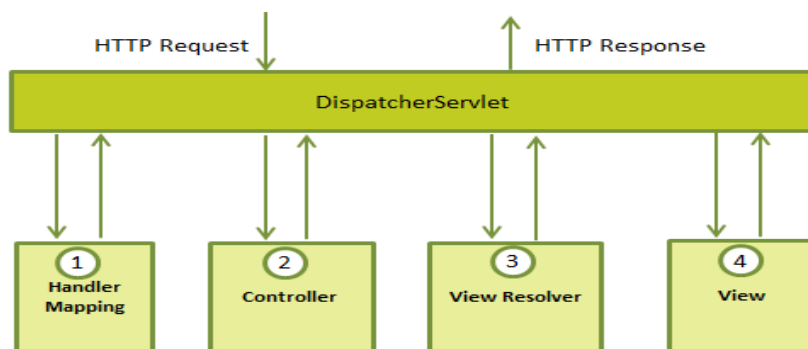
Η πλατφόρμα Cloud Benchmark tool έχει υλοποιηθεί ακολουθώντας το πρότυπο αρχιτεκτονικής Spring Model-View-Controller το οποίο προσφέρει έτοιμα συστατικά που μπορούν να χρησιμοποιηθούν για την ανάπτυξη ευέλικτων και χαλαρά συνδεδεμένων εφαρμογών διαδικτύου. Το πρότυπο MVC επιτρέπει το διαχωρισμό των διαφόρων πτυχών

της εφαρμογής (λογική των εισροών, επιχειρηματική λογική, και λογική UI), εξασφαλίζοντας παράλληλα μια χαλαρή σύνδεση μεταξύ αυτών των στοιχείων (R. Johnson, 2016).

- Η οντότητα Model ενσωματώνει τα δεδομένα της εφαρμογής.
- Η οντότητα View είναι υπεύθυνη για την απεικόνιση των δεδομένων της οντότητας Model δημιουργώντας HTML περιεχόμενο που προβάλλεται στο πρόγραμμα περιήγησης του χρήστη.
- Η οντότητα Controller είναι υπεύθυνος για την επεξεργασία των αιτημάτων των χρηστών, δημιουργεί τη κατάλληλη οντότητα Model και την τροφοδοτεί στην κατάλληλη οντότητα View για την απεικόνιση των δεδομένων.

DispatcherServlet

Η διαδικτυακή έκδοση του Spring MVC προτύπου είναι σχεδιασμένο γύρω από την οντότητα DispatcherServlet που χειρίζεται όλες τις HTTP αιτήσεις και απαντήσεις. Η ροή επεξεργασίας αιτημάτων από την οντότητα DispatcherServlet απεικονίζεται στο ακόλουθο διάγραμμα:



Εικόνα 19: DispatcherServlet (R. Johnson, 2016)

Σε κάθε εισερχόμενο HTTP αίτημα προς τον DispatcherServlet εκτελούνται τα παρακάτω βήματα (R. Johnson, 2016):

- a. Μετά τη λήψη του HTTP αιτήματος, ο DispatcherServlet συμβουλευτείται την οντότητα HandlerMapping προκειμένου να καλέσει τη κατάλληλη οντότητα Ελεγκτή (Controller).
- b. Ο ελεγκτής λαμβάνει το αίτημα και καλεί τις κατάλληλες μεθόδους παροχής υπηρεσιών (GET ή POST μέθοδοι) που έχουν υλοποιηθεί. Η επιλεγμένη μέθοδος θα θέσει τα δεδομένα που παράγονται από την ενσωματωμένη επιχειρηματική λογική σε ένα στιγμιότυπο Model και επιστρέφει το όνομα της οντότητας View στο DispatcherServlet για την απεικόνιση των δεδομένων.

- c. Η οντότητα DispatcherServlet θα συμβουλευθεί την οντότητα ViewResolver σχετικά με την οντότητα View που έχει επιλεχθεί για το HTTP αίτημα.
- d. Μόλις οριστικοποιηθεί η απαιτούμενη , Η DispatcherServlet περνά το μοντέλο δεδομένων για την άποψη που τελικά αποδίδεται στο πρόγραμμα περιήγησης.

Όλες οι παραπάνω αναφερθέντες οντότητες, δηλαδή HandlerMapping, Controller και ViewResolver είναι μέρη του Πλαισίου μίας Διαδικτυακής Εφαρμογής (WebApplicationContext). Το Πλαίσιο αυτό ορίζεται στο αρχείο web.xml της υλοποίησης της Cloud Benchmark tool εφαρμογής. Στο αρχείο δηλώνεται η κλάση που υλοποιεί την οντότητα DispatcherServlet. Επίσης δηλώνεται σε ποιου είδους http αιτήματα ανταποκρίνεται (servlet-mapping). Η δήλωση `<url-pattern>/</url-pattern>` σημαίνει ότι ανταποκρίνεται σε όλα τα αιτήματα (πχ *.jsp, *.html, κοκ) (R. Johnson, 2016).

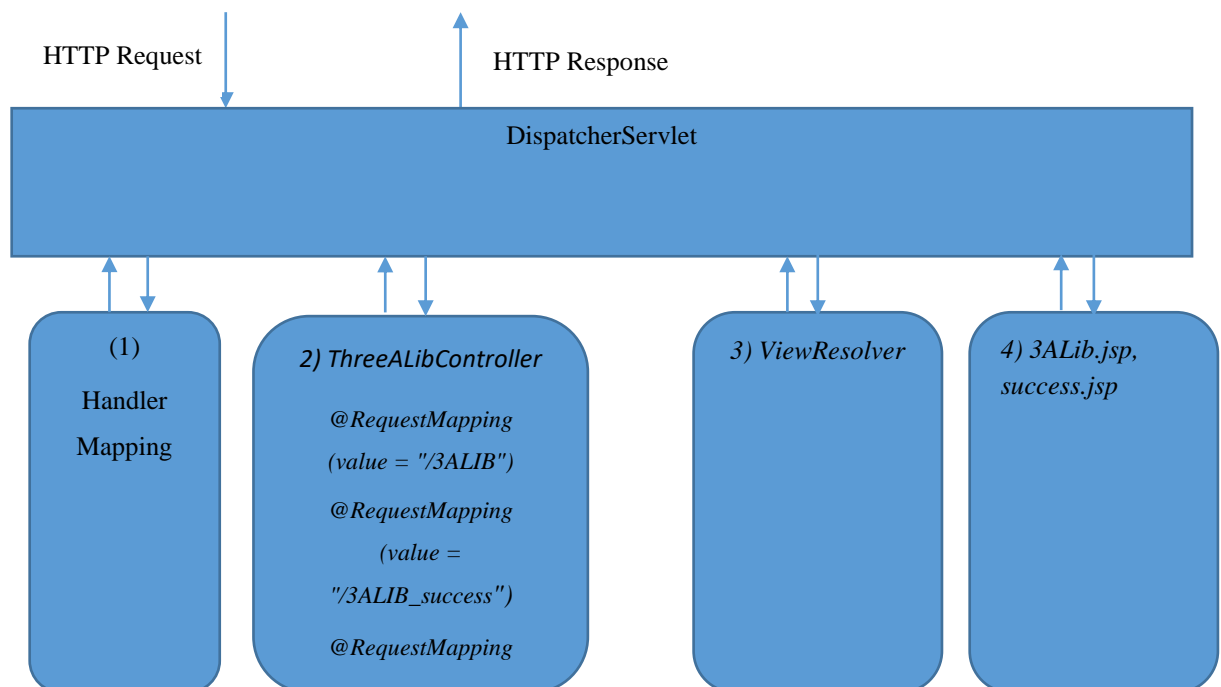
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/root-context.xml</param-value>
  </context-param>
  <listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>
  <servlet>
    <servlet-name>appServlet</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
```

```

<servlet-name>appServlet</servlet-name>
<url-pattern>/</url-pattern>
</servlet-mapping>
</web-app>

```

Επομένως για τη διασύνδεση της υπηρεσίας 3ALib πάνω στην Spring Web-MVC Αρχιτεκτονική της Cloud Benchmark tool πλατφόρμας ορίστηκαν οι παρακάτω κανόνες σύμφωνα με το σχετικό πρότυπο.



Εικόνα 20: Κανόνες Δρομολόγησης HTTP Αιτημάτων προς την υπηρεσία 3ALib μέσω του *DispatcherServlet*

Για οποιαδήποτε αιτήματα http προς την υπηρεσία 3ALib μέσω της κεντρικής ιστοσελίδας της Cloud Benchmark tool πλατφόρμας (σύνδεσμος <http://localhost:8080/3ALIB>), η οντότητα *DispatcherServlet* συμβουλεύεται την οντότητα *HandlerMapping*. Η τελευταία τα προωθεί στον Ελεγκτή *ThreeALibController* καθώς σε αυτή τη κλάση έχει δηλωθεί ο κανόνας `@RequestMapping(value = "/3ALIB")`. Ο κανόνας αυτός υλοποιείται από τη παρακάτω μέθοδο.

```

@RequestMapping(value = "/3ALIB")
public String threelib_service(@ModelAttribute("3alib") ThreeALib dbentity, BindingResult
result, HttpServletRequest request, HttpServletResponse response) throws IOException {
    (1) if (dbentity.getDatabase() != null) {
        try {
            Provider provider= dbentity.getProvidersum();
            if (provider.getProvider().equals("Amazon WS"))
                provider.setProvider("aws-ec2");
            (3) ArtistorInput artistorInput= new ArtistorInput(false, null,
provider.getProvider(), "", provider.getUsername(), provider.getPassword(),
dbentity.getDbip(), dbentity.getDbuser(), dbentity.getDbkey());
            (4) RunAvailabilityAuditingDaemon rdaemon= new
RunAvailabilityAuditingDaemon(artistorInput);
            (5) rdaemon.execute();
            request.getSession().setAttribute("3ALIB_daemon", rdaemon);
        } catch (Exception e)
            (6) return "forward:3alib_success";
        } else
            (2) return "forward:threelib_view";
    }
}

```

Η μέθοδος λαμβάνει ως παράμετρο την κλάση *ThreeALib* η οποία θα δημιουργήσει την οντότητα *Model* με δεδομένα που θα ληφθούν από τη διεπαφή με το χρήστη. Συγκεκριμένα, η μέθοδος *threelib_service* καλείται αρχικά για να αρχικοποιήσει την οθόνη στην οποία ο χρήστης μπορεί να ορίσει τις παραμέτρους της υπηρεσίας. Αυτό γίνεται στο σημείο ελέγχου (1) στον παρακάτω κώδικα. Εφόσον δεν έχει οριστεί η παράμετρος της βάσης δεδομένων σημαίνει ότι ο χρήστης πρέπει να μεταπηδήσει στην οθόνη υποβολής παραμέτρων. Αυτό γίνεται στο σημείο ελέγχου (2). Ο έλεγχος επιστρέφει στην οντότητα *HandlerMapping* για να ορίσει ποιος ελεγκτής είναι υπεύθυνος για αιτήματα *threelib_view*. Ο έλεγχος επιστρέφει στον Ελεγκτή *ThreeALibController* καθώς υλοποιεί την αντίστοιχη μέθοδο. Η μέθοδος ορίζει το όνομα της οντότητας *View* η οποία πρέπει να εμφανιστεί στον χρήστη. Σε αυτό συμβάλει η οντότητα *ViewResolver* η οποία επιστρέφει τη σελίδα *3alib.jsp*.

```
@RequestMapping(value = "/threealib_view")
```

```
public String nav(Model model) {
```

```
    return "3alib/3alib";
```

```
}
```

Κατά τη συμπλήρωση των στοιχείων της φόρμας και την υποβολή της συμπληρώνονται τα στοιχεία της οντότητας *ThreeALib (model)*. Καθώς αυτό αφορά απόκριση (http response) σε αίτημα που έχει ήδη περάσει από τον Ελεγκτή *ThreeALibController* καλείται ξανά η μέθοδος `@RequestMapping(value = "/3ALIB")`

```
public String threealib_service(@ModelAttribute("3alib") ThreeALib dbentity,...
```

για να διαχειριστεί την απόκριση προς τον χρήστη.

Το σημείο ελέγχου (1) παραπάνω επιτρέπει τη διαχείριση των δεδομένων που έχουν φορτωθεί στη κλάση *ThreeALib (model)*. Η εκτέλεση περνάει πλέον από τα σημεία (3) – αρχικοποίηση κλάσης *ArtistorInput*, (4) – Αρχικοποίηση *RunAvailabilityAuditingDaemon*, (5) έναρξη της καταγραφής στοιχείων Διαθεσιμότητας και ΣΔΕΠ, και (6) μετακίνηση στην οντότητα *View* με τίτλο "3alib_success". Στη περίπτωση αυτή τον έλεγχο αναλαμβάνει εκ νέου ο Ελεγκτής *ThreeALibController* εκτελώντας τη παρακάτω μέθοδο. Η τελευταία επιστρέφει τη σελίδα *success.jsp* για τη προβολή των υπολογιστικών δεδομένων από τη βάση δεδομένων στην οθόνη του χρήστη.

```
@ModelAttribute("3ALIB_daemon")
```

```
@RequestMapping(value = "/3alib_success") {
```

```
    return "3alib/success";
```

```
}
```


4

Μελέτη Περίπτωσης Καταγραφών 3ALib

Η βιβλιοθήκη 3ALib χρησιμοποιεί μία βάση δεδομένων MongoDB για την αποθήκευση καταγραφών (logs) με τις λεπτομερείς μετρήσεις της διαθεσιμότητας των υπηρεσιών υπό εξέταση (π.χ. Amazon EC2). Στην περίπτωση που δεν υπάρχει δυνατότητα μέτρησης της διαθεσιμότητας των υπηρεσιών (π.χ. δεν υπάρχει δικτυακή πρόσβαση, κ.λ.π.) έχουμε αναπτύξει λειτουργικότητα προσομοίωσης, κατά την οποία η βιβλιοθήκη 3ALib εισάγει υπάρχοντες μετρήσεις διαθεσιμότητας υπηρεσιών από αρχείο στην βάση δεδομένων. Στο σημείο αυτό πρέπει να επισημάνουμε ότι όταν χρησιμοποιείται η λειτουργία προσομοίωσης θα πρέπει να έχουν αφαιρεθεί από την βάση δεδομένων εγγραφές με το ίδιο μοναδικό αναγνωριστικό (id). Στην παρούσα μελέτη περίπτωσης θα ασχοληθούμε με την εξέταση των καταγραφών της βιβλιοθήκης 3ALib με την χρήση εντολών από command line.

4.1 Γενικές Εντολές

Για να εκτελέσουμε εντολές στην MongoDB πρέπει να πάμε στον φάκελο bin της εγκατάστασής της και να εκτελέσουμε την ακόλουθη εντολή:

```
mongo
```

Για να συνδεθούμε στην βάση δεδομένων πρέπει να τρέξουμε την ακόλουθη εντολή:

```
use 3alib
```


Μπορούμε να δούμε όλες τις καταγραφές της βιβλιοθήκης 3ALib εκτελώντας την ακόλουθη εντολή:

```
db.log_samples.find().pretty()
```

Στην περίπτωση που θέλουμε να διαγράψουμε όλες τις καταγραφές της βιβλιοθήκης 3ALib από την βάση δεδομένων πρέπει να εκτελέσουμε την ακόλουθη εντολή:

```
db.log_samples.remove({})
```

4.2 Ερωτήματα

Στα ερωτήματα που θα παρουσιάσουμε στην συνέχεια υπάρχουν οι ακόλουθες δύο εναλλακτικές κατά την εκτέλεσή τους:

- Παρουσίαση καταγραφών σε JSON μορφή: Εάν στο τέλος του εκάστοτε ερωτήματος που παρουσιάζει εγγραφές της βάσης δεδομένων βάλουμε την παράμετρο **pretty** τότε τα αποτελέσματα θα εμφανιστούν σε JSON μορφή. Αυτό συμβαίνει γιατί οι καταγραφές που αποθηκεύονται στην βάση δεδομένων είναι σε JSON Μορφή. Π.χ. η ακόλουθη εντολή εμφανίζει όλες τις καταγραφές σε JSON μορφή:

```
db.log_samples.find({}).pretty()
```

```
{
  "hostMetadata": true,
  "durable": true
}
{
  "supportImage": {
    "supervisor": {
      "supervisor": "dsh",
      "id": "ml.medium",
      "type": "Mikrosaipe",
      "tags": [
        "providerId": "ml.medium",
        "userMetadata": {
        }
      ]
    }
  },
  "os": {
    "family": "UBUNTU",
    "arch": "paravirtual",
    "version": "12.10",
    "description": "090720100477/ubuntui/images/ebz/ubuntui-quantal-12.10-1386-server-0811118",
    "is64bit": false
  },
  "hostname": "ip-10-147-129-209",
  "ip": "us-east-1/i-50b0d9bd",
  "type": "NODE",
  "tags": [
    "providerId": "i-50b0d9bd",
    "name": "Optimifcsc1chp",
    "location": {
      "scope": "ZONE",
      "id": "us-east-1c",
      "description": "us-east-1c",
      "parent": {
        "scope": "REGION",
        "id": "us-east-1",
        "description": "us-east-1",
        "parent": {
          "scope": "PROVIDER",
          "id": "aws-ec2",
          "description": "https://ec2.us-east-1.amazonaws.com",
          "is3106Codes": [
            "US-VA",
            "US-CA",
            "US-OR",
            "US-WP",
            "IT",
            "SO",
            "AU-AP",
            "JP-13"
          ]
        }
      },
      "metadata": {
        "is3106Codes": [
          "US-VA"
        ]
      }
    }
  },
  "is3106Codes": [
    "US-VA"
  ],
  "metadata": {
  }
}
{
  "userMetadata": {
    "name": "Optimifcsc1chp"
  },
  "reachability": "REACHABLE"
}
```

Εικόνα 21: καταγραφές σε JSON μορφή

- Πλήθος καταγραφών: Για να εμφανίσουμε το πλήθος των εγγραφών της βάσης δεδομένων για ένα συγκεκριμένο ερώτημα, αρκεί στο τέλος του ερωτήματος να βάλουμε την παράμετρο **count**. Π.χ. η ακόλουθη εντολή εμφανίζει το πλήθος όλων των καταγραφών που υπάρχουν στην βάση δεδομένων:

```
db.log_samples.find({}).count()
```

```
> db.getCollection('log_samples').find({}).count()
14
>
```

Εικόνα 22: Πλήθος καταγραφών

Στην συνέχεια παρουσιάζουμε ορισμένα ερωτήματα στις καταγραφές που υπάρχουν στην βάση δεδομένων:

1. Καταγραφές που αναφέρονται σε υπηρεσίες που εκτελούνται / είναι διαθέσιμες:

```
db.log_samples.find({"status" : "RUNNING"})
```

2. Καταγραφές που αναφέρονται σε υπηρεσίες που δεν εκτελούνται / δεν είναι διαθέσιμες:

```
db.log_samples.find({"status" : {$ne:"RUNNING"}})
```

3. Καταγραφές που αναφέρονται σε υπηρεσίες που η κατάσταση του backend τους είναι εκτελείται / διαθέσιμο:

```
db.log_samples.find({"backendStatus" : "running"})
```

4. Καταγραφές που αναφέρονται σε υπηρεσίες που η κατάσταση του backend τους δεν εκτελείται / δεν είναι διαθέσιμο:

```
db.log_samples.find({"backendStatus" : {$ne:"running"}})
```

5. Καταγραφές που αναφέρονται σε υπηρεσίες με περισσότερους από έναν πυρήνες επεξεργαστών:

```
db.log_samples.find({"hardware.processors.cores" : {$gte:1}} )
```

6. Καταγραφές που αναφέρονται σε υπηρεσίες με ταχύτητα επεξεργαστών μεγαλύτερη του ενός:

```
db.log_samples.find({"hardware.processors.speed" : {$gte:1}} )
```

7. Καταγραφές που αναφέρονται σε υπηρεσίες με μνήμη (RAM) μεγαλύτερη των 1024:

```
db.log_samples.find({"hardware.ram" : {$gte:1024}})
```

8. Καταγραφές που αναφέρονται σε υπηρεσίες που εκτελούνται σε λειτουργικό σύστημα UBUNTU της έκδοσης 12.10:

```
db.log_samples.find({'$and':{'os.family' : 'UBUNTU'},{'os.version' : '12.10'}})
```

Αντίστοιχα μπορούν να δημιουργηθούν ανάλογα ερωτήματα με βάση τις εκάστοτε επιχειρησιακές απαιτήσεις. Για να δημιουργηθούν και άλλα ερωτήματα απαιτείται πολύ καλή γνώση του JSON σχήματος των καταγραφών.

Σημείωση: Δεν παραθέτουμε αποτελέσματα εκτέλεσης των παραπάνω ερωτημάτων γιατί εκτελέστηκαν με δεδομένα προσομοίωσης τα οποία είχαν ίδιες τιμές σε όλα τα πεδία.

5

Συμπεράσματα – Μελλοντικές Επεκτάσεις

Η επέκταση του Cloud Benchmark Tool με τη βιβλιοθήκη 3ALib είναι μία καινοτομία στον συγκεκριμένο τομέα καθώς δεν έχει εμφανιστεί η βιβλιοθήκη ως μέρος ανάλογης πλατφόρμας εργαλείων συγκριτικής αξιολόγησης (benchmarking). Αναμφισβήτητα αυτό που επιτυγχάνεται είναι μία ενοποιημένη προσέγγιση στη μετάπτωση επιστημονικών και εμπορικών εφαρμογών στο υπολογιστικό νέφος έτσι ώστε:

- a. Να πραγματοποιείται ένας «εσωτερικός» έλεγχος εντός του οργανισμού αν οι υπηρεσίες μίας βάσης δεδομένων, ενός στιγμιότυπου Map Reduce, ή μίας φάρμας web servers εκτελούνται σύμφωνα με τους στόχους που έχουν τεθεί.
- b. Να πραγματοποιείται ένας «εξωτερικός» έλεγχος του παρόχου από τον οργανισμό-πελάτη αν οι παρεχόμενες υπηρεσίες εκτελούνται σύμφωνα με το συμφωνημένο SLA.

Τα παραπάνω κρίνονται απαραίτητα καθώς ερευνητικοί οργανισμοί, ακαδημαϊκά ιδρύματα και εμπορικές επιχειρήσεις χρησιμοποιούν υποδομές ΥΝ για να καλύψουν τις σύγχρονες ανάγκες λογισμικού και υπολογιστικών υποδομών. Συνεπώς καλούνται να προσαρμόσουν τα επιχειρησιακά τους μοντέλα και να προβλέπουν το σχετικό κόστος στον προϋπολογισμό τους.

Ειδικά η πραγματοποίηση δοκιμών για τη συλλογή δεδομένων από μία ενεργή συνδρομή ΥΝ (πχ Amazon Web Services) και η αποθήκευση των στατιστικών σε μία βάση δεδομένων μέσω της 3ALib δεν ήταν εφικτή στο πλαίσιο της εργασίας κατά ένα μεγάλο βαθμό. Παρόλα αυτά

πραγματοποιήθηκαν δοκιμές με ιστορικό ανάλογων δεδομένων. Συνεπώς, μελλοντικές προσπάθειες επέκτασης της εργασίας θα πρέπει να περιλαμβάνουν:

1. Την πραγματοποίηση δοκιμών συλλογής πραγματικών SLA δεδομένων από περισσότερους του ενός παρόχων
2. Την ανάλυση των παραπάνω δεδομένων σε συνδυασμό με τα αποτελέσματα δοκιμών μέτρησης της απόδοσης κάποιων από τις παραπάνω διεργασίες (βάση δεδομένων, Map Reduce, ή web server) για την άντληση επιπλέον συμπερασμάτων.

Την επέκταση του γραφικού περιβάλλοντος 3ALIB για την πραγματοποίηση αναζήτησης και εξαγωγής αναφορών (SLA reports). Η παρούσα γραφική διεπαφή δεν επιτρέπει την υποβολή συνδυαστικών ερωτημάτων (AND/OR) και δεν επιτρέπει την εξαγωγή των αποτελεσμάτων σε μία κανονικοποιημένη γραμμογράφιση προς εξαγωγή σε text /pdf αρχείο.

6

Αναφορές και Βιβλιογραφία

- A. Menychtas, K. Konstanteli et al. (2014). Software modernization and cloudification using the ARTIST migration methodology and framework. *Scientific International Journal for Parallel and Distributed Computing*, Volume 15, No 2, pp. 131-152, June 2014.
- ARTIST Consortium. (2016, 02 03). *3ALib Implementation*. Ανάκτηση από 3ALib:
http://www.artist-project.eu/sites/default/files/3ALib_Supporting%20document.pdf
- Brodkin, J. (2008). *Gartner: Seven cloud-computing security risks*. Gartner.
- E. Aljoumah, F. Al-Mousawi et al. (2015). SLA in Cloud Computing Architectures: A Comprehensive Study. *International Journal of Grid Distribution Computing*, Vol. 8, No.5, (2015), pp.7-32.
- Hogberg, D. (2012). *An Applied Evaluation and Assessment of Cloud Computing Platforms*. UMEA SWEDEN: Umea University.
- J. Gorroñoigoitia. (2016, 02 03). *ARTIST Integrated Architecture*. Ανάκτηση από 3ALib:
<http://www.artist-project.eu/node/167>
- Jensen, M. ; Horst Gortz; Schwenk, J. ; Gruschka, N. ; Iacono, L.L. (2009). On Technical Security Issues in Cloud Computing. *Cloud Computing, 2009. CLOUD '09. IEEE International Conference on* (σσ. 109-116). Bangalore: IEEE.

- L. Gillam, B. Li, J. O'Loughlin. (2012). Benchmarking Cloud Performance for Service Level Agreement Parameters. *Cloud Computing*.
- Oracle. (2011). *Oracle Cloud Computing*. Ανάκτηση 04 09, 2014, από <http://www.oracle.com/us/solutions/cloud/overview/index.html>
- Perry, R., & Hendrick, S. (2012). *The Business Value of Amazon Web Services Accelerates Over Time*. IDC.
- R. Johnson, J. H. (2016, 02 03). *Introduction to Spring Web MVC framework*. Ανάκτηση από Spring Framework Reference Documentation: <https://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html>
- Takabi, H.;Joshi, J.B.D.;Gail-Joon Ahn. (2010). Security and Privacy Challenges in Cloud Computing Environments. *Security & Privacy, IEEE* , 8(6), 24-31.
- Γ. Τ. Κουσιουρής. (2012). *Τεχνολογίες και μηχανισμοί μοντελοποίησης και πρόβλεψης απόδοσης υπηρεσιοστρεφών εφαρμογών και υποδομών*. Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου.
- Ε. Κεβάνη. (2014). *A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS: The Web Tool*. Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς.
- Χ. Σταμπολτάς. (2014). *A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS: The Web Tool*. Πανεπιστήμιο Πειραιώς: Διδακτική της Τεχνολογίας & Ψηφιακά Συστήματα Τμήμα Ψηφιακών Συστημάτων.