



Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών Συστημάτων
Πρόγραμμα Μεταπτυχιακών Σπουδών
" Διδακτική της Τεχνολογίας & Ψηφιακά
Συστήματα "

“A Unified Benchmark Framework for Cloud Deployed Applications
Tool in IaaS: The Web Tool”

Συγγραφέας: Σταμπολτάς Χριστόφορος

Διπλωματική Εργασία υποβληθείσα στο Τμήμα Ψηφιακών
Συστημάτων του Πανεπιστημίου Πειραιώς ως μέρος των απαιτήσεων
για την απόκτηση Μεταπτυχιακού Διπλώματος Ειδίκευσης στα
Δικτυοκεντρικά Συστήματα

Επιβλέπων Καθηγητής: Μαρίνος Θεμιστοκλέους

Πειραιάς, Απρίλιος 2014

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στο Πανεπιστήμιο Πειραιά, στο Τμήμα Ψηφιακών Συστημάτων, κατά το έτος 2013.

Η ολοκλήρωση της μεταπτυχιακής αυτής εργασίας θα ήταν αδύνατη χωρίς την πολύτιμη υποστήριξη του καθηγητή μου, Αν. Καθηγητή των Ψηφιακών Συστημάτων Κου Μαρίνου Θεμιστοκλέους. Του εκφράζω ένα βαθύ ευχαριστώ για όλη τη βοήθεια που μου προσέφερε. Χρωστάω επίσης, ένα πολύ μεγάλο ευχαριστώ στο Λέκτορα Καθηγητή Κυριαζή Δημοσθένη για την υποστήριξή του, την καθοδήγησή του, την κατανόηση του και την προσφορά των μηχανημάτων βάσει των οποίων αυτή η εργασία ήρθε εις πέρας. Επίσης, χρωστάω ένα μεγάλο ευχαριστώ στον κύριο Γιώργο Κουσιούρη, ο οποίος με κατεύθυνε ορθώς με σκέψη άρτια και επιστημονική.

Επίσης θα ήθελα, από βάθους καρδιάς να ευχαριστήσω το φίλο, συνάδερφο και συνεργάτη Έλτον Κεβάνη, του οποίου η βοήθεια και η παρακίνηση ήταν θεμελιώδης για την άριστη συνεργασία μας, για την ολοκλήρωση αυτής της εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την υπομονή και τη συμπαράσταση τους καθ' όλη τη διάρκεια των σπουδών μου, ειδικότερα δε κατά την εκπόνηση της διπλωματικής μου εργασίας.

Πίνακας Περιεχομένων

Περίληψη	8
Η ανάγκη της συγκριτικής αξιολόγησης	8
Η ανάγκη της συγκριτικής αξιολόγησης στην πληροφορική	9
Βασικοί Ορισμοί Υπολογιστικού Νέφους	10
Ορισμός Υπολογιστικού Νέφους	10
Ορισμός του Cloud Computing σύμφωνα με το US NIST	12
Βασικά Λειτουργικά Χαρακτηριστικά Cloud Computing:	12
Μοντέλα Υπηρεσιών	14
Μοντέλα Ανάπτυξης Υπολογιστικών Νεφών	19
Σκοπός των Υπολογιστικών Νεφών	22
Ανοικτά Ερευνητικά Θέματα	23
Συμπεράσματα	27
Κεφάλαιο 3 Ανάγκη ύπαρξης Εργαλείου Συγκριτικής Αξιολόγησης Υπολογιστικών Νεφών	28
Ανάγκη	28
Το εργαλείο συγκριτικής αξιολόγησης που έχουμε αναπτύξει	30
Βασικοί Πάροχοι Υπολογιστικών Νεφών (IaaS)	31
Amazon Web Services	31
Τύποι Εικονικών Μηχανών και κόστη	32
Rackspace Cloud	35
Τύποι Εικονικών Μηχανών και κόστη	36
Επιδόσεις 1 Flavor Class	36
Επιδόσεις 2 Flavor Class	38
Vcloud	39
Κόστη	40
Ερευνητικές – Εμπορικές Διεργασίες αντίστοιχες με το εργαλείο που έχουμε αναπτύξει (State of the Art)	41
CloudHarmony	41
Cloudsleuth	43
Openbenchmarking	47
Συμπέρασμα - Τι διαφορετικό προσφέρει το δικό μας εργαλείο	49
ΤΟ ΕΡΓΑΛΕΙΟ	50
Ολοκληρωμένη Λύση Εφαρμογής	50

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Ανάλυση της Εφαρμογής	52
Αρχιτεκτονική	52
Client-Server	52
Περιγραφή Λειτουργικότητας	61
Υλοποίηση	69
Θεμελιώδεις Κλάσεις	69
Μία γενική αναφορά στον τρόπο λειτουργίας των κλάσεων	72
DB Test	72
Web Serving	78
Map-Reduce / Hadoop Test	82
ΣΤΙΓΜΙΟΤΥΠΑ ΕΡΓΑΛΕΙΟΥ	92
DB Bench	93
Map Reduce	97
Web Serving	102
Reports	105
Μελλοντική Μελέτη	109
Βιβλιογραφικές Αναφορές	110

Πίνακας Εικόνων

Εικόνα 1	Νέφη Υπολογιστών	14
Εικόνα 2	Ιδιωτικά Συστήματα ΥΝ	19
Εικόνα 3	Νέφη Κοινοτήτων	20
Εικόνα 4	Δημόσια Νέφη	20
Εικόνα 5	Υβριδικά Νέφη	21
Εικόνα 6	Ρόλοι οντοτήτων και αλληλεπίδραση μεταξύ τους στα Υπολογιστικά Νέφη	24
Εικόνα 7	Amazon Web Services[188]	31
Εικόνα 8	Rackspace	35
Εικόνα 9	Vcloud	39
Εικόνα 10	Vmcloud Κόστη	40
Εικόνα 11	CloudHarmony	41
Εικόνα 12	Αρχική Σελίδα	43
Εικόνα 13	Επιλογές	43
Εικόνα 14	Αποτελέσματα	43
Εικόνα 15	CloudSleuth Αρχική	45
Εικόνα 16	Cloudsleuth Availability	45
Εικόνα 17	Cloudsleuth Performance Analyzer	46
Εικόνα 18	Openbenchmarking.org	47
Εικόνα 19	Openbenchmarking Αρχική	48
Εικόνα 20	Openbenchmarking Επιλογή Δοκιμής	48
Εικόνα 21	Ολοκληρωμένη Λύση	52
Εικόνα 22	Μοντελο Client-Server[31]	53
Εικόνα 23	Java EE[33]	55
Εικόνα 24	The MVC Model[25]	57
Εικόνα 25	Spring MVC[36]	58
Εικόνα 26	3-Tier Application[37]	59
Εικόνα 27	Twitter Bootstrap	61
Εικόνα 28	SE Συνάρτηση	63
Εικόνα 29	Περιγραφή Λειτουργικότητας	68
Εικόνα 30	MVC	69
Εικόνα 31	General Classes	70
Εικόνα 32	Db Controller	76

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Εικόνα 33 DB ClassesWeb Serving Test	77
Εικόνα 34 Web Serving Controller	80
Εικόνα 35 Web Serving Classes	81
Εικόνα 36 Map Reduce Controller	83
Εικόνα 37 Dfsio Map Reduce Classes	84
Εικόνα 38 MrBenchController	85
Εικόνα 39 NNBenchmark Classes	86
Εικόνα 40 Teeragen CController Classes	87
Εικόνα 41 Report CController	90
Εικόνα 42 Reports Classes	91
Εικόνα 43 Αρχικό Μενού	92
Εικόνα 44 Db Bench Step 1	93
Εικόνα 45 Wrong Input	94
Εικόνα 46 DB Step 2	95
Εικόνα 47 DB Step 3	95
Εικόνα 48 DB Final	96
Εικόνα 49 Map Reduce Step 1	97
Εικόνα 50 Map Reduce Step3	99
Εικόνα 51 Map Reduce Step 2	98
Εικόνα 52 Map Reduce Final MRBENCH	100
Εικόνα 53 Map Reduce Final DFSIO	100
Εικόνα 54 NNBENCH Write	101
Εικόνα 55 TERAGEN Outputs	101
Εικόνα 56 Web Serving Step 1	102
Εικόνα 57 Web Serving Step2	103
Εικόνα 58 Web Serving Step3	103
Εικόνα 59 Web Serving Final	103
Εικόνα 60 Waiting Screen	104
Εικόνα 61 Welcome Screen	105
Εικόνα 62 Reports Form	106
Εικόνα 63 Reports Select Test	107
Εικόνα 64 Reports Final	108

ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

Τον 21ο αιώνα η ανάπτυξη των ICT τεχνολογιών δημιούργησε το όραμα της «υπολογιστικής ωφέλειας» (computer utility) σύμφωνα με το οποίο η υπολογιστική ωφέλεια θα γίνει η 5η δημόσια ωφέλεια (μετά το νερό, τον ηλεκτρισμό, το αέριο και το τηλέφωνο), υπό την έννοια της εύκολης πρόσβασης οποιουδήποτε και οπουδήποτε σε αυτές. Διάφορα υπολογιστικά μοντέλα προτάθηκαν, ερευνήθηκαν και υλοποιήθηκαν προς αυτή την κατεύθυνση όπως cluster, grid, utility, services computing και άλλα. Ο όρος utility computing αναφέρεται στην παροχή υπολογιστικών πόρων και υπηρεσιών κατά ζήτηση και αποτελεί μία πιο γενική έννοια, η οποία μπορεί να υλοποιηθεί/εξασφαλιστεί από διάφορες τεχνολογίες.

Το cluster είναι ένας τύπος παράλληλου και κατανεμημένου συστήματος που αποτελείται από ένα σύνολο διασυνδεδεμένων αυτόνομων υπολογιστών που λειτουργούν σαν ένας ενιαίος υπολογιστικός πόρος.

Το grid είναι ένας τύπος παράλληλου και κατανεμημένου συστήματος που επιτρέπει το διαμοιρασμό, την επιλογή και την πρόσμιξη γεωγραφικά κατανεμημένων αυτόνομων πόρων με δυναμικό τρόπο κατά ζήτηση ανάλογα με τη διαθεσιμότητα, την απόδοση και το κόστος καθώς και βάση των απαιτήσεων ποιότητας (Quality of Service) των χρηστών.

Στην παρούσα διπλωματική εργασία θα αντιμετωπίσουμε και θα προσπαθήσουμε στο καλύτερο δυνατό σημείο να επιλύσουμε το πρόβλημα της ύπαρξης ενός αξιόπιστου εργαλείου για την μέτρηση εφαρμογών που εκτελούνται σε ένα εικονικό μηχάνημα (Virtual Machine , VM) σε ένα σύστημα παρόχου υλικού σύννεφου στο στρώμα Υποδομή ως Υπηρεσία (IaaS). Έπειτα , αφού προσδιορίσουμε τις ανάγκες που επιτάσσουν την ύπαρξη και άρα τη δημιουργία ενός τέτοιου εργαλείου θα παρουσιάσουμε τη λύση που

αποφασίσαμε να σχεδιάσουμε και να υλοποιήσουμε. Θα αναλύσουμε τη δομή της , τα λειτουργικά της μέρη , καθώς και τον τρόπο που αυτά συνδέονται ως τελικό στάδιο.

ΚΕΦΑΛΑΙΟ 2 Συγκριτική Αξιολόγηση – Υπολογιστικά Νέφη

Περίληψη

Στο παρόν κεφάλαιο θα παρουσιάσουμε αναλυτικά το σκοπό για τον οποίο δημιουργήσαμε το παρόν εργαλείο βάσει των αναγκών και των απαιτήσεων που καλύπτει. Θα καλύψουμε τα ανοικτά πλαίσια στην επιστημονική και εμπορική έρευνα που έχουν οδηγήσει στην ύπαρξή του , όπως επίσης και την αντίστοιχη βιβλιογραφία που έχει θεμελιωθεί για το σκοπό αυτό. Έπειτα θα προσπαθήσουμε να αναλύσουμε με μεθοδικότητα και επιστημονική προσέγγιση την παραπάνω βιβλιογραφία ως προς το πόσο καλύπτει τα κενά που έχουν προκύψει και δικαιολογούν τη δημιουργία του εργαλείου μας. Έπειτα θα περιγράψουμε τα ανοικτά θέματα που δικαιολογούν ερευνητική έρευνα πέραν της δικής μας και θα αναλύσουμε τα συμπεράσματα αυτής της προσέγγισης.

Η ανάγκη της συγκριτικής αξιολόγησης

Η συγκριτική αξιολόγηση των συστημάτων πληροφορικής είναι η διαδικασία της αξιολόγησης των επιδόσεών τους αλλά και άλλων μη λειτουργικών χαρακτηριστικών με άλλα συστήματα ή προ-συμφωνημένα με τη βιομηχανία πρότυπα. Παραδοσιακά η κύρια

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

χρήση της συγκριτικής αξιολόγησης έχει ως θεμελιώδη στόχο να διευκολύνει κατά ένα μεγάλο ποσοστό την τεκμηρίωση συμβάσεων υπολογιστικών συστημάτων μέσω της δημοσίευσης των επαληθευμένων αποτελεσμάτων από τρίτα και ανεξάρτητα μέρη. Σκοπό έχει την αντικειμενική καταγραφή προτερημάτων καθώς και ελλείψεων αυτών των προτύπων με άμεσο στόχο τη βελτίωση τους.

Οι στόχοι της συγκριτικής αξιολόγησης είναι:

1. Να καθορίσει τι και πού χρήζει ανάγκης βελτιώσεων
2. Να αναλύσει τον τρόπο με τη βοήθεια άλλων διαδικασιών ή/και άλλων οργανισμών προκειμένου να επιτύχουν υψηλά επίπεδα απόδοσης
3. Να χρησιμοποιήσουν οι εμπλεκόμενοι οργανισμοί τις εξαγόμενες πληροφορίες προκειμένου να βελτιώσουν τις επιδόσεις τους υιοθετώντας κατάλληλες μεθόδους βελτίωσης.

Ο πρωταρχικός σκοπός της συγκριτικής αξιολόγησης είναι φυσικά η βελτίωση της ανταγωνιστικής θέσης της εταιρείας.

Η ανάγκη της συγκριτικής αξιολόγησης στην πληροφορική

Η έννοια της συγκριτικής αξιολόγησης δεν είναι κάτι το καινούργιο ή πρωτοποριακό στον τομέα της πληροφορικής και γενικότερα στον τομέα της Τεχνολογίας Πληροφορικών (Information Technology IT). Με τη βοήθεια εργαλείων αξιολόγησης, συνήθως αναφερόμαστε σε ένα πρόγραμμα ή πιο γενικά σε μία ομάδα προγραμμάτων, που χρησιμοποιούνται για την αξιολόγηση μίας «λύσης» σε σχέση με μία άλλη. Από τις αρχές του 1970, οι τεχνικές συγκριτικής αξιολόγησης, έχουν εφαρμοσθεί στην πληροφορική για την μέτρηση της αποδοτικότητας των υπολογιστών, υπολογιστικών συστημάτων και δικτύων.

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Προκειμένου να προχωρήσουμε σε βάθος στο εργαλείο που έχουμε αναπτύξει πρέπει αρχικά να βάλουμε ως βάση κάποιες αρχές και όρους που έχουν σχέση με τη συγκριτική αξιολόγηση. Η παρακάτω ορολογία θα χρησιμοποιηθεί κατά τη διάρκεια αυτής της παρουσίασης. Μια συγκριτική αξιολόγηση (benchmark) περιέχει ένα πλήρες σετ από προδιαγραφές που απαιτούνται για να καταστεί δυνατή η αξιολόγηση. Οι προδιαγραφές αυτές περιλαμβάνουν:

- Ένα σενάριο
- Τα κριτήρια αξιολόγησης
- Τις μετρήσεις αξιολόγησης
- Τα αποτελέσματα της συγκριτικής αξιολόγησης

Αναλόγως του πλαισίου που θα οριστεί, πλέον ο όρος συγκριτική αξιολόγηση θα καταδείχνει μία συγκεκριμένη δοκιμή: «η συγκριτική αξιολόγηση δύο εικονικών μηχανών με 4-core 2Gb μνήμη και Ubuntu περιβάλλον, που στο πρώτο εξετάζουμε το throughput σε MongoDB και στο δεύτερο MySQL».

Βασικοί Ορισμοί Υπολογιστικού Νέφους

Ορισμός Υπολογιστικού Νέφους

Αν κάνουμε αυτή την ερώτηση σε διαφορετικούς ανθρώπους θα διαπιστώσουμε ότι δεν υπάρχει μια απλή απάντηση. Το υπολογιστικό νέφος είναι η παροχή υπολογιστικού χώρου με τη μορφή υπηρεσίας κι όχι προϊόντος, όπου κοινόχρηστοι πόροι, λογισμικό και πληροφορίες παρέχονται σε υπολογιστές και άλλες συσκευές διαμέσου ενός δικτύου (συνήθως το Διαδίκτυο).

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Ο όρος «Cloud», πιθανότατα προκύπτει από τα γνωστά σκίτσα που έχει κατά καιρούς ζωγραφίσει καθένας από εμάς προσπαθώντας να αναπαραστήσει στο χαρτί το Διαδίκτυο. Με το σκίτσο του «Cloud» συνήθως προσπαθούμε να περιγράψουμε ένα απομακρυσμένο σύνολο αξιόπιστων υπηρεσιών στον οποίο και στηριζόμαστε, χωρίς όμως να μας ενδιαφέρει το πώς λειτουργεί αυτό στα ενδότερα του. Όπως ακριβώς συμβαίνει και με το ηλεκτρικό ρεύμα όπου ο καταναλωτής ασχολείται μόνο με που βρίσκεται μια πρίζα και όχι με το πώς παράγεται ή μεταφέρεται η ηλεκτρική ενέργεια. [13].

Με τον όρο Υπολογιστικό Νέφος (ΥΝ) (Cloud Computing) εννοούμε την πρόσβαση σε υπολογιστές και τη λειτουργικότητα τους δια μέσω του διαδικτύου ή ενός τοπικού δικτύου. Οι χρήστες του Cloud ζητούν την πρόσβαση από ένα σύνολο υπηρεσιών διαδικτύου, οι οποίες διαχειρίζονται τους διαθέσιμους υπολογιστικούς πόρους (μπορεί να είναι υπολογιστές, δίκτυο, αποθηκευτικός χώρος, λειτουργικά συστήματα, περιβάλλοντα ανάπτυξης εφαρμογών αλλά και εφαρμογές). Όταν αποδοθεί ένα τμήμα των πόρων σε κάποιο χρήστη αυτό είναι ατομικά αφιερωμένο σε αυτό το χρήστη μέχρι αυτός να απελευθερώσει τη χρήση του. Ονομάζεται Cloud Computing γιατί ο χρήστης δεν μπορεί στην πραγματικότητα να δει να προσδιορίσει και να καταλάβει το που βρίσκονται ακριβώς οι υποδομές που χρησιμοποιεί ή τον εξοπλισμό που φιλοξενεί τις υπηρεσίες που έχει ζητήσει και του έχει δοθεί η άδειά ανά χρησιμοποιήσει. Σχηματικά θα μπορούσαμε να πούμε ότι οι πόροι ανασύρονται από ένα Cloud (σύννεφο – νέφος) πόρων όταν αποδοθούν σε ένα χρήστη και αυτές επιστρέφουν σε αυτό όταν απελευθερωθούν. Ένα Cloud είναι ένα σύνολο μηχανημάτων και διαδικτυακών υπηρεσιών που υλοποιούν το Cloud Computing.

Το Cloud Computing παρέχει υπολογιστική ισχύ, λογισμικό, πρόσβαση σε δεδομένα, και υπηρεσίες αποθηκευτικού χώρου, χωρίς να απαιτείται από τον τελικό χρήστη η φυσική τοποθεσία και παραμετροποίηση του συστήματος που τους παρέχει αυτές τις υπηρεσίες.

Ορισμός του Cloud Computing σύμφωνα με το US NIST

Το Cloud Computing είναι ένα μοντέλο που επιτρέπει την εύκολη, on-demand (τη στιγμή που ζητείται) πρόσβαση μέσω δικτύου σε ένα “κοινό ταμείο” από παραμετροποιήσιμους υπολογιστικούς πόρους (π.χ. Δίκτυα, servers, αποθηκευτικό χώρο, εφαρμογές και υπηρεσίες) οι οποίοι μπορούν πολύ εύκολα να παρακολουθηθούν και να αποδοθούν με πολύ μικρή παρέμβαση της διαχείρισης, ή αλληλεπίδρασης από τον πάροχο των υπηρεσιών. Αυτό το μοντέλο του Cloud προάγει τη διαθεσιμότητα και απαρτίζεται από πέντε βασικά χαρακτηριστικά, τρία μοντέλα παροχής-παράδοσης της υπηρεσίας και τέσσερα μοντέλα υλοποίησης του.

Βασικά Λειτουργικά Χαρακτηριστικά Cloud Computing:

- On-demand self-service. Ο καταναλωτής μπορεί να ζητήσει μονομερώς τις υπολογιστικές δυνατότητες, όπως ο χρόνος που θα χρησιμοποιήσει στον server και το μέγεθος του αποθηκευτικού χώρου που θα χρησιμοποιήσει μέσω δικτύου αυτόματα χωρίς να απαιτείται καμία ανθρώπινη αλληλεπίδραση με τον πάροχο της εκάστοτε υπηρεσίας.
- Ubiquitous network access. (“από παντού δικτυακή πρόσβαση”) Οι Δυνατότητες αυτές είναι προσπελάσιμες από παντού δια μέσω δικτύου και διαδεδομένων στάνταρντ και μηχανισμών πράγμα που επιτρέπει την χρήση κι πρόσβαση τους από

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

πολλές και ετερόκλητες πλατφόρμες χρήστη (π.χ. Κινητά τηλέφωνα, φορητούς υπολογιστές, PDA).

- **Location independent resource pooling.** Οι υπολογιστικοί πόροι του παρόχου συγκεντρώνονται σε ένα κοινό σύνολο χρησιμοποιώντας ένα μοντέλο πολλών ενοικιαστών, με διαφορετικές φυσικούς και εικονικούς πόρους οι οποίοι αποδίδονται πολλές φορές δυναμικά μετά από την απαίτηση του χρήστη. Ο καταναλωτής γενικά δεν έχει κανένα έλεγχο και γνώση για την ακριβή τοποθέτηση του παρεχόμενου πόρου, αλλά μπορεί να δύναται να προσδιορίσει σε ένα πιο αφηρημένο επίπεδο την τοποθεσία όπως η χώρα η πόλη ή το συγκεκριμένο data-center. Παραδείγματα τέτοιων πόρων είναι αποθηκευτικός χώρος, επεξεργασία, μνήμη, εύρος ζώνης δικτύου, και Virtual Machines.
- **Rapid elasticity.** οι πόροι που χρησιμοποιεί ένας καταναλωτής μπορούν να μεταβάλλονται ελαστικά και γρήγορα, ώστε να ακολουθούν τη ζήτηση και τις ανάγκες του τελευταίου
- **Measured Service.** Τα συστήματα Cloud αυτόματα ελέγχουν και βελτιστοποιούν τη χρήση των υπολογιστικών πόρων χρησιμοποιώντας κάποια μετρητικά συστήματα σε κάποιο από τα επίπεδα της αφαίρεσης που εισάγουν, κατάλληλο για την συγκεκριμένη παρεχόμενη υπηρεσία (αποθηκευτικού χώρου, υπολογιστικής ισχύος, εύρους ζώνης, ενεργού αριθμού χρηστών κλπ.). Η χρήση των πόρων μπορεί να παρακολουθηθεί, ελεγχθεί και να αναφερθεί ότι παρέχει διαφάνεια και για τις δύο πλευρές, τελικού χρήστη – καταναλωτή και παρόχου της χρησιμοποιούμενης υπηρεσίας.



Εικόνα 1 Νέφη Υπολογιστών

Μοντέλα Υπηρεσιών

- Λογισμικό ως Υπηρεσία (Software as a Service (SaaS): Η δυνατότητα που παρέχεται στον καταναλωτή είναι η χρήση των εφαρμογών του παρόχου οι οποίες εκτελούνται σε μία υποδομή νέφους υπολογιστών. Οι εφαρμογές είναι προσβάσιμες από διάφορες συσκευές – πελάτες (client devices) , είτε μέσω μιας πολύ ελαφριάς πλατφόρμας τύπου πελάτη, (πχ ενός περιηγητή ιστοσελίδων) , είτε τέλος από μία εφαρμογή που λειτουργεί αποκλειστικά ως πελάτης για τη διασύνδεση με το χρήστη. Ο καταναλωτής δε διαχειρίζεται ή έχει πρόσβαση στην υποκείμενη τεχνολογία λειτουργίας του νέφους , όσον αφορά το δίκτυο, τους εξυπηρετητές , το λειτουργικό σύστημα , μεθόδους αποθήκευσης ή ακόμα και μεμονωμένες δυνατότητες της εφαρμογής , με την πιθανή εξαίρεση τη δυνατότητα

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

ο χρήστης να είναι σε θέση να διαχειριστεί συγκεκριμένες εξ' ορισμού ρυθμίσεις της εφαρμογής. Εκτός από την ανάπτυξη/προσαρμογή της εφαρμογής, αυτός ο ρόλος προσδιορίζει επίσης ποιοι είναι οι όροι που μπορούν να ρυθμιστούν από τους τελικούς καταναλωτές της εφαρμογής:

- Παράμετροι φόρτου εργασίας: διαμορφώσιμες παράμετροι με τις οποίες ένας πελάτης θέλει να εκτελέσει μια υπηρεσία (π.χ. αριθμός χρηστών, ανάλυση μιας εικόνας σε εφαρμογές πολυμέσων κ.λπ.)
- Ποιότητα των παραμέτρων υπηρεσίας (Quality of Service-QoS): η αναμενόμενη έξοδος της υπηρεσίας, αυτή που χρησιμοποιείται για να εκφράσει τα επίπεδα QoS (π.χ. χρόνος απόκρισης, πλαίσια ανά δευτερόλεπτο μιας μετάδοσης πολυμέσων κ.λπ.). Αυτό είναι επίσης γνωστό ως Βασικοί Δείκτες Απόδοσης (Key Performance Indicators-KPIs)
- Πλατφόρμα ως Υπηρεσία (Platform as a Service (PaaS)): Η πλατφόρμα ως υπηρεσία ή αλλιώς γνωστό και ως cloudware είναι η συνέχεια του SaaS. Παρέχει μια cloud πλατφόρμα εφαρμογών για εταιρείες ή ιδιώτες που κατασκευάζουν λογισμικό για τους ίδιους είτε για τρίτους. Το PaaS παρέχει όλους τους πόρους που απαιτούνται για να δημιουργηθούν εφαρμογές και υπηρεσίες μέσω του Internet, χωρίς να πρέπει να κατεβάσει ή να εγκαταστήσει λογισμικό (μέσω γλωσσών προγραμματισμού, βιβλιοθήκες, βάσεις δεδομένων, υπηρεσίες και εφαρμογές που υποστηρίζονται από τους παρόχους). Οι υπηρεσίες PaaS περιλαμβάνουν την σχεδίαση εφαρμογών, την ανάπτυξη, τον έλεγχο, την εγκατάσταση και την φιλοξενία εφαρμογών. Έχει σχεδιαστεί για να χρησιμοποιείται από πολλούς χρήστες ταυτόχρονα και παρέχει αυτόματες διευκολύνσεις για ταυτόχρονη διαχείριση, κλιμάκωση, ανακατεύθυνση και ασφάλεια. Αυτός ο ρόλος ενσωματώνει

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

ενέργειες όπως η μοντελοποίηση της εφαρμογής, η πρόβλεψη απόδοσής της, η παρακολούθηση, αξιολόγηση των γεγονότων και υλοποίηση διορθωτικών ενεργειών (π.χ. αύξηση των πόρων). Σε αυτήν την διαδικασία μπορεί να χρησιμοποιήσει τις πληροφορίες που παρέχονται από τον υπεύθυνο για την ανάπτυξη εφαρμογής μέσω μιας περιγραφής του τμήματος λογισμικού (π.χ. σε γλώσσα XML (W3 Schools)). Ο πελάτης δεν διαχειρίζεται ή ελέγχει τις υποδομές όπως το δίκτυο, τους διακομιστές, τα λειτουργικά συστήματα παρά μόνο την εφαρμογή που θα ανεβάσει στο νέφος. Πάροχοι PaaS μπορούν να χρησιμοποιούν APIs, ή λογισμικό πύλης εγκατεστημένο στον υπολογιστή του πελάτη. Η Force.com και η GoogleApps είναι παραδείγματα PaaS. Οι προγραμματιστές πρέπει να γνωρίζουν ότι αυτή τη στιγμή, δεν υπάρχουν πρότυπα για τη διαλειτουργικότητα ή τη δυνατότητα μεταφοράς δεδομένων στο σύννεφο. Ορισμένοι πάροχοι δεν θα επιτρέψει το λογισμικό που δημιουργήθηκε από τους πελάτες τους να μετακινηθούν από την πλατφόρμα του παρόχου.

- Υποδομή ως Υπηρεσία (Infrastructure As a Service (IaaS)): Η υποδομή ως υπηρεσία ή αλλιώς γνωστή και ως υλικό (Hardware as a Service-HaaS) προσφέρει το υλικό (σε αντίθεση με τα SaaS και PaaS τα οποία παρέχουν εφαρμογές), έτσι ώστε η επιχείρηση να μπορεί να βάζει ότι θέλει σε αυτό. Η εταιρεία ή ο ιδιώτης μπορεί να υπενοικιάσει υποδομή ανάλογα με τις απαιτήσεις εκείνης της χρονικής στιγμής, αντί να προβεί στην αγορά εξοπλισμού (υπολογιστικού, δικτυακού, κλπ). Ο χρήστης δεν διαχειρίζεται/ ελέγχει τη βασική υποδομή cloud, αλλά έχει τον έλεγχο των λειτουργικών συστημάτων, της αποθήκευσης, και τις αναπτυσσόμενες εφαρμογές και ενδεχομένως περιορισμένο έλεγχο της επιλογής εξαρτημάτων δικτύωσης (π.χ. firewalls υποδοχής). Σημαντικό πλεονέκτημα του IaaS είναι η δυνατότητα μεταφοράς εικονικών μηχανών από το ιδιόκτητο περιβάλλον της εταιρείας ή του

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

ιδιώτη στο cloud, με συνοπτικές διαδικασίες καθώς και ότι πολλοί χρήστες μπορούν να χρησιμοποιήσουν τον εξοπλισμό ταυτόχρονα όπως το Amazon Web Services παρέχει το virtual instanceAPI διακομιστή) για να ξεκινήσει, να σταματήσει η πρόσβαση και να ρυθμίσει ο χρήστης virtual servers και αποθήκευσης τους. Στην επιχείρηση, το cloud computing επιτρέπει σε μια εταιρεία να πληρώνουν μόνο για όσο ικανότητας όπως απαιτείται, και να φέρει περισσότερο σε απευθείας σύνδεση, το συντομότερο, όπως απαιτείται. Επειδή αυτό το pay-for-what-you-use μοντέλο μοιάζει με τον τρόπο ηλεκτρικού ρεύματος, καυσίμων και νερού καταναλώνονται, είναι μερικές φορές αναφέρεται ως υπολογιστική χρησιμότητα

- Καταναλωτής/πελάτης: μια οντότητα που χρησιμοποιεί μια εφαρμογή που προσφέρεται ως υπηρεσία. Ο καταναλωτής έρχεται σε επαφή με τον προμηθευτή PaaS που έχει καταστήσει αυτό το SaaS διαθέσιμο και ζητά αυτό το λογισμικό για μια συγκεκριμένη διαμόρφωση (παραμέτρους φόρτου εργασίας). Επιπλέον απαιτεί ορισμένα επίπεδα ποιότητας της υπηρεσίας (QoS), όπως αυτή υπολογίζεται από τους συγκεκριμένους βασικούς δείκτες απόδοσης (KPIs) της εφαρμογής. Ο προμηθευτής PaaS έρχεται έπειτα σε επαφή με έναν προμηθευτή IaaS και ζητά τους πόρους υλικού που έχει προβλέψει ότι θα ικανοποιήσουν τις ανάγκες του καταναλωτή. Μετά από μια επιτυχή διαπραγμάτευση, η εφαρμογή (SaaS) εκκινείται (από το PaaS) στις υποδομές που παρέχονται από το IaaS.

Όλες οι προαναφερθείσες ενέργειες επικυρώνονται τυπικά μέσω των Συμφωνιών Επιπέδων Υπηρεσιών (Service Level Agreements-SLAs) μεταξύ των συμβαλλόμενων μερών. Το SLA περιγράφει τη συμφωνία για αυτούς τους όρους και διατάξεις και δεσμεύει νομικά τους εμπλεκόμενους παρόχους. Χαρακτηριστικά, υπάρχουν δύο SLAs, ένα μεταξύ του καταναλωτή και του προμηθευτή PaaS (Application SLA, A-SLA),

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

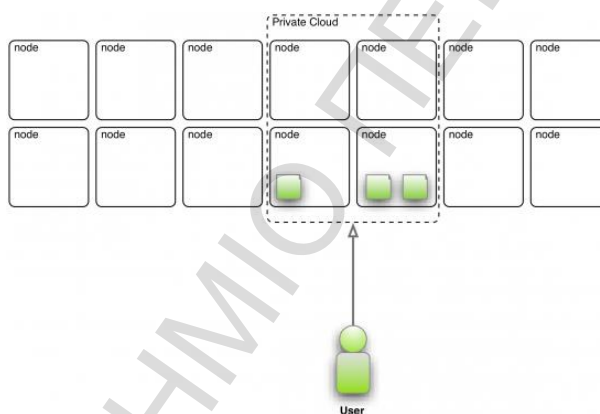
που αναφέρονται στους όρους εφαρμογής (παράμετροι φόρτου εργασίας και επίπεδα τιμών των παραμέτρων υπηρεσίας), και ένα μεταξύ των προμηθευτών PaaS και IaaS (Technical SLA, T-SLA), που εκφράζονται με όρους επιπέδων εικονικών υπολογιστικών πόρων.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Μοντέλα Ανάπτυξης Υπολογιστικών Νεφών

• Ιδιωτικά Συστήματα Υπολογιστικών Νεφών (Private Cloud).

Οι υποδομές του νέφους υπολογιστών λειτουργούν αποκλειστικά για ένα οργανισμό. Μπορούν να διαχειρίζονται από τον οργανισμό ή από τρίτους και μπορούν να βρίσκονται στους χώρους του οργανισμού είτε όχι.

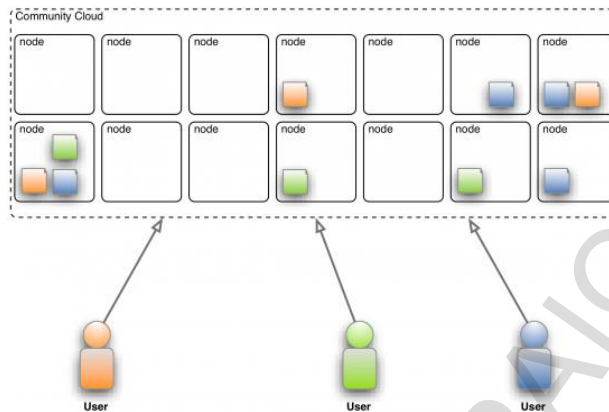


Εικόνα 2 Ιδιωτικά Συστήματα ΥΝ

• Νέφη Κοινοτήτων (Community cloud).

Οι υποδομές του νέφους υπολογιστών μοιράζονται από διάφορους οργανισμούς και υποστηρίζουν μία συγκεκριμένη κοινότητα με κοινές ανησυχίες (αποστολή, απαιτήσεις ασφάλειας, πολιτική, εκτιμήσεις συμμόρφωσης). Μπορεί να διαχειρίζονται από τον οργανισμό ή από τρίτους και μπορεί να βρίσκεται στους χώρους των οργανισμών είτε όχι.

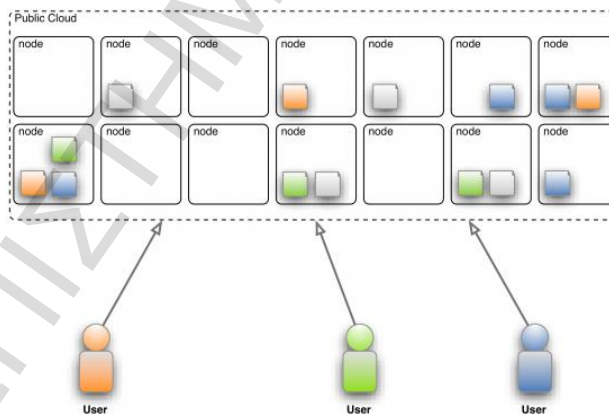
“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”



Εικόνα 3 Νέφη Κοινοτήτων

- Δημόσια Νέφη (Public Cloud).

Οι υποδομές του νέφους υπολογιστών είναι διαθέσιμες στο γενικό κοινό ή μία μεγάλη βιομηχανική ομάδα, επιχειρήσεις, ακαδημαϊκές κοινότητες, δημόσιοι οργανισμοί ή πιθανοί συνδυασμοί αυτών.

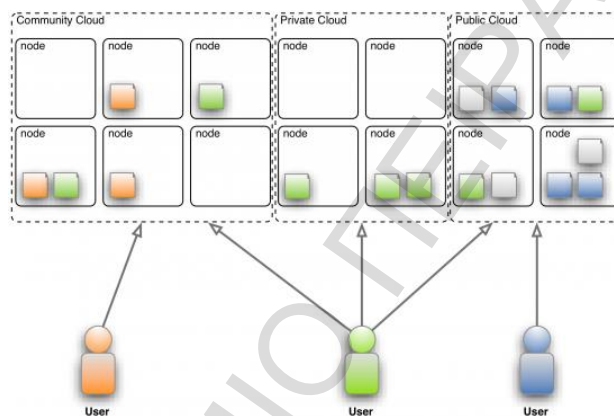


Εικόνα 4 Δημόσια Νέφη

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

• Υβριδικά Νέφη (Hybrid Cloud).

Οι υποδομές του νέφους υπολογιστών αποτελούν ένα συνονθύλευμα από δύο διακριτών τύπων νεφών (δημοσίων, κοινοτήτων, ιδιωτικών) που παραμένουν ομαδικές οντότητες αλλά είναι δεσμευμένα μεταξύ υποχρεωμένες να ακολουθούν τυποποιημένες ή αποκλειστικές τεχνολογίες που επιτρέπουν τη μεταφορά δεδομένων και εφαρμογών.



Εικόνα 5 Υβριδικά Νέφη

Σκοπός των Υπολογιστικών Νεφών

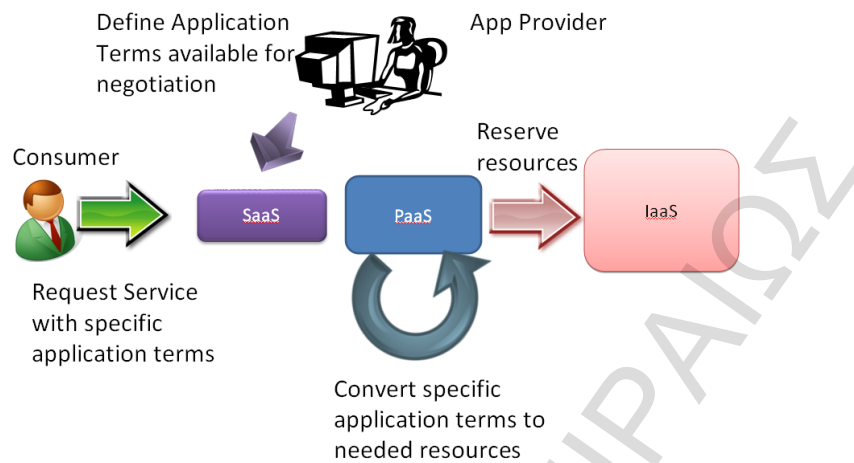
Από απόψεως υλικού (Hardware), τρεις είναι οι πτυχές που τα Νέφη Υπολογιστών καινοτομούν [33]:

1. Η ψευδαίσθηση των άπειρων υπολογιστικών πόρων , διαθέσιμοι σε πρώτη ζήτηση από τον πελάτη, εξαλείφοντας με αυτόν τον τρόπο , την ανάγκη των χρηστών να προγραμματίσουν μπροστά για τροφοδότηση νέου υλικού.
2. Η κατάργησή των εκ-των προτέρων συμφωνιών από τους χρήστες Υπολογιστικών Νεφών. Έτσι οι χρήστες/εταιρείες/οργανισμοί μπορούν να ξεκινήσουν με ένα ελάχιστο , στον απαιτήσέων τους, υλικό και προοδευτικά να αυξάνουν , πχ υπολογιστική ισχύ ή αποθηκευτικό χώρο κατά το δοκούν , χωρίς να χρειάζεται εξαρχής η σύναψη συμφωνίας.
3. Η σημαντική ιδιότητα που διακρίνει τα συστήματα Υπολογιστικών Νεφών , που είναι η πληρωμή-κατά-χρήση , δηλαδή , ο χρήστης πληρώνει τις υπηρεσίες για όσο του είναι απαραίτητες , πχ (επεξεργαστές ανά ώρα και αποθηκευτικό χώρο ανά ημέρα) και να τους αποδεσμεύει όποτε οι συγκεκριμένοι πόροι δεν είναι διαθέσιμοι και αυτοί με τη σειρά τους να διατίθενται όπου είναι απαραίτητοι, από τον πάροχο σε κάποιον άλλο πελάτη.

Ανοικτά Ερευνητικά Θέματα

Σε αυτό το σύνθετο και πολυστρωματικό οικοσύστημα, ένα από τα κρισιμότερα βήματα είναι ο ρόλος του PaaS στη μετάφραση των καθορισμένων από τον καταναλωτή παραμέτρων εφαρμογής (φόρτος εργασίας και QoS, όπως αυτοί εκφράζονται στο A-SLA) στις ιδιότητες επιπέδων των πόρων (δεδομένου ότι αυτοί απαιτούνται για να εκφραστούν στο T-SLA). Εντούτοις, η ανταλλαγή των πληροφοριών μεταξύ των οντοτήτων που βρίσκονται στα διαφορετικά στρώματα είναι δύσκολη για τεχνικούς και επιχειρηματικούς λόγους. Από τη μία πλευρά, είναι εξαιρετικά απίθανο για αυτές τις οντότητες, ειδικά για τους προμηθευτές εφαρμογής και τους παρόχους υποδομής, να εκθέσουν/αποκαλύψουν τις εσωτερικές λειτουργικές παραμέτρους και τις διαδικασίες των στρωμάτων τους σε άλλα στρώματα λόγω του κινδύνου πρόσβασης ανταγωνιστών σε αυτές τις πληροφορίες. Αφ' ετέρου, η ανταλλαγή τέτοιων πληροφοριών δεν είναι πάντα τεχνικά εφικτή. Κάθε οντότητα εστιάζει σε ένα ιδιαίτερο σύνολο παραμέτρων που μπορεί να ερμηνευθεί από αυτό το στρώμα και βασίζεται σε αυτές τις παραμέτρους τους αντίστοιχους μηχανισμούς για την παροχή υπηρεσιών. Επιπλέον, η χρήση βιβλιοθηκών ή γενικά κώδικα τρίτων (third party software) για τα οποία δεν υπάρχει γνώση της εσωτερικής δομής δυσχεραίνει την ροή της πληροφορίας, ακόμα και αν αρθούν οι προαναφερθέντες περιορισμοί. Η έλλειψη προτύπων για τα Υπολογιστικά Νέφη είναι ένα άλλο μεγάλο ζήτημα που περιορίζει τον αποτελεσματικό διαμοιρασμό των πληροφοριών μεταξύ των διαφορετικών οντοτήτων. Μερικές ενδεχόμενες προσπάθειες προς αυτήν την κατεύθυνση περιλαμβάνουν τα OCCI [16] και OVF[17].

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”



Εικόνα 6 Ρόλοι οντοτήτων και αλληλεπίδραση μεταξύ τους στα Υπολογιστικά Νέφη

Στο τωρινό οικοσύστημα Νεφών δεν υπάρχει τρόπος ή πρόβλεψη για την χαρτογράφηση αυτή από A-SLA σε T-SLA. Η συνηθέστερη μορφή είναι ένα στατικό SLA το οποίο δίνουν οι πάροχοι στους καταναλωτές και αναφέρεται σε παραμέτρους όπως η διαθεσιμότητα των πόρων. Ο καταναλωτής της υπηρεσίας πρέπει να διαλέξει από μόνος του όμως πόσους πόρους χρειάζεται για μια συγκεκριμένη εφαρμογή με συγκεκριμένο φορτίο. Ο πάροχος του εξασφαλίζει μόνο ότι οι εικονικές του μηχανές θα είναι διαθέσιμες π.χ. το 99.9% του χρόνου.

Επιπλέον, ένα σημαντικό κενό στα υπάρχοντα Νέφη είναι η έλλειψη εγγυήσεων απόδοσης πραγματικού χρόνου και ποιότητας της υπηρεσίας. Συγκεκριμένα, για τις εφαρμογές πραγματικού χρόνου (real-time), ο πάροχος πρέπει να λαμβάνει υπόψη του το βασικό χαρακτηριστικό αυτών, που είναι η πραγματοποίηση μιας πράξης από τη συγκεκριμένη εφαρμογή μέσα σε κάποιο αυστηρό χρονικό πλαίσιο.

Οι πάροχοι υπολογιστικών πόρων δίνουν πρόσβαση σε συγκεκριμένα κβαντισμένες εικονικές μηχανές (π.χ. μία εικονική μηχανή ανά πυρήνα) χωρίς να εγγυώνται όμως την πραγματική απόδοσή τους. Επιπλέον, για την υποστήριξη εφαρμογών πραγματικού χρόνου

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

είναι απαραίτητη μια λεπτομερής ανάλυση κάθε εφαρμογής και αυστηρός έλεγχος των πόρων, ώστε να ικανοποιούνται οι χρονικοί περιορισμοί που εισέρχονται λόγω του πραγματικού χρόνου. Για να γίνει αυτό όμως, χρειάζεται μια λογική προ-σχεδιασμού της κατανομής των πόρων, που καθιστά απαραίτητο ένα μοντέλο in advance reservation. Διορθώσεις κατά τη διάρκεια της πραγματικής λειτουργίας μπορούν να γίνουν, αλλά πρέπει να είναι περιορισμένης κλίμακας ώστε να μην υπάρχουν παραβιάσεις στους χρονικούς περιορισμούς.

Επιπλέον, ένα πολύ σημαντικό θέμα που υπεισέρχεται σε διαμοιραζόμενες υποδομές είναι η αλληλεπίδραση μεταξύ των εικονικών πόρων που εκτελούνται σε ένα φυσικό πόρο και ανήκουν σε διαφορετικούς χρήστες. Αυτή η αλληλεπίδραση μπορεί να ποικίλει ανάλογα με τον τρόπο δρομολόγησης, την αρχιτεκτονική του φυσικού πόρου και το είδος των εφαρμογών που εκτελούνται μέσα στις εικονικές μηχανές. Άρα ένα πάροχος Νέφους οφείλει να έχει τρόπους εύρεσης και ελαχιστοποίησης αυτής της αλληλεπίδρασης, ώστε να παρέχει την ποιότητα υπηρεσίας που υπόσχεται στους πελάτες του.

Θα μπορούσαμε να αναφέρουμε 5 βασικά ανοικτά ερευνητικά θέματα για τα Νέφη Υπολογιστών[34]:

1. Διαθεσιμότητα της υπηρεσίας. Οι οργανισμοί ανησυχούν για το αν οι υπηρεσίες θα έχουν επαρκή διαθεσιμότητα αν υλοποιηθούν σε Νέφη Υπολογιστών προκαλώντας με αυτόν τον τρόπο ανησυχία. Κατά έναν ειρωνικό τρόπο, τα υπάρχοντα προϊόντα SaaS έχουν θέσει πολύ ψηλά τον πήχη. Η αναζήτηση Google είναι άμεσα διαθέσιμη 24/7/365 όντας για πολλούς ο κυρίαρχος στον τομέα της στο Διαδίκτυο. Αν οι χρήστες ήθελαν να χρησιμοποιήσουν αυτήν την υπηρεσία και για οποιοδήποτε λόγο αυτή είχε τεθεί εκτός λειτουργίας, τότε οι τελευταίοι θα νόμιζαν ότι το Διαδίκτυο θα είχε «πέσει».

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

2. «Κλείδωμα» των δεδομένων: Πακέτα λογισμικού έχουν βελτιώσει τη διαλειτουργικότητα μεταξύ πλατφορμών , αλλά οι διεπαφές των προγραμματιστικών εφαρμογών (API's) που έχουν σχέση με τα Νέφη Υπολογιστών είναι ακόμα ιδιόκτητα ή δεν έχουν αποτελέσει ακόμα αντικείμενο ενεργού τυποποίησης. Έτσι οι πελάτες δεν μπορούν εύκολα να εξάγουν δεδομένα από προγράμματα από έναν χώρο ΥΝ προκειμένου να εκτελεστούν σε ένα δεύτερο , διαφορετικό.
3. Εμπιστευτικότητα των δεδομένων και δυνατότητα ελέγχου: Δεν υπάρχουν ουσιαστικά πολλά εμπόδια για τη δημιουργία ενός ασφαλούς περιβάλλοντος στα ΥΝ , και οποιαδήποτε από αυτά μπορούν εύκολα να ξεπεραστούν σχετικά εύκολα με τεχνολογίες όπως η κρυπτογραφημένη αποθήκευση , εικονικά τοπικά δίκτυα , φίλτρα πακέτων και άλλα συναφή. Όμως η διαδικασία αυτή είναι σε πρώιμο αλλά εξελίξιμο στάδιο χωρίς να έχουν γίνει σημαντικά βήματα προς ένα ασφαλές υλοποιημένο περιβάλλον.
4. Το φαινόμενο του «χωνιού» κατά τη διαδικασία της μεταφοράς αρχείων: Οι εφαρμογές τα τελευταία χρόνια έχουν αρχίσει να γίνονται πιο απαιτητικές στο θέμα των δεδομένων. Όταν το κόστος για τη μεταφορά 1Gb κατά μήκος του νέφους, κοστίζει από 100\$ έως 150\$, τα κόστη αυξάνονται σημαντικά, δημιουργώντας ένα φλέγον ζήτημα κατά πόσο είναι αποδοτικό μία εταιρεία/οργανισμός/ιδιώτης να χρησιμοποιεί υπηρεσίες Νέφους , όταν τα κόστη μεταφοράς των δεδομένων είναι τόσο υψηλά.
5. Συγκριτική αξιολόγηση: Κατά καιρούς έχουν δημοσιευθεί ερευνητικές μελέτες για την ανάγκη δημιουργίας εργαλείων ασφαλούς και αντικειμενικής συγκριτικής αξιολόγησης τόσο μεταξύ παρόχων ΥΝ , αλλά και εφαρμογών σε επίπεδο SaaS σε ΥΝ . Στο άρθρο των Saurabh Kumar Garg, Steve Versteeg, Rajkumar Buyya [35] , έχει

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

γίνει εκτενής αναφορά , ανάλυση και παρουσίαση ένα πολύ-επίπεδο πλαίσιο λογισμικού για συγκριτικές αξιολογήσεις. Το εργαλείο CloudCmp παρέχει μία ασφαλή προσέγγιση του τρόπου μεθόδου μετρήσεων για τη συγκριτική αξιολόγηση και σκοπό έχει να επιτευχθεί η καλύτερη αναλογία απόδοσης/κόστους για μία εφαρμογή που εκτελείται σε ένα εικονικό μηχάνημα Υπολογιστικού Νέφους. Επίσης υπάρχουν διαδικτυακές εφαρμογές όπως η CloudHarmony[37], CloudSleuth[38] και OpenBenchmarking.org[39] που στοχεύουν σε μία αντικειμενική συγκριτική αξιολόγηση.

Συμπεράσματα

Το όραμα της Πληροφορικής ως εργαλείο , πλέον μπορούμε να ισχυριστούμε με βεβαιότητα ότι αρχίζει και σχηματίζεται . Οι επιχειρήσεις μέσω Διαδικτύου έχουν αυξηθεί και αυξάνονται με γεωμετρικό αυξητικό ρυθμό. Οι απαιτήσεις του μέσου χρήστη έχει αυξηθεί αντίστοιχα και δεν μπορούν σε καμία περίπτωση να ταυτιστούν με τις απαιτήσεις του μέσου χρήστη πριν 10 χρόνια. Πριν από 2 δεκαετίες ήταν κάτι σχεδόν αδιανόητο μία εταιρεία να αναπτυχθεί και να αποκτήσει εκατομμύρια πελάτες , και αν εν τέλει τα κατάφερνε αυτό θα έπαιρνε χρόνια. Αυτό τώρα μπορεί να συμβεί σε λίγους μήνες. Από την πλευρά των Υπολογιστικών Νεφών και των παρόχων αυτών, η κατασκευή μεγάλων κέντρων δεδομένων με τη βασική χρήση τριών απλών συνιστωσών , επεξεργαστική ισχύς-αποθηκευτικός χώρος-δικτύωση , σε χαμηλά κόστη υλοποίησης , με ταυτόχρονη εφαρμογή της πολιτικής χρέωση για όσο είναι απαραίτητο (pay as you go) , έχει δημιουργήσει μία νέα αγορά. Από την πλευρά του χρήστη , μπορούμε να πούμε ότι μέσα σε λίγα και απλά βήματα , ο μέσος χρήστης μπορεί να δημιουργήσει το δικό του κέντρο δεδομένων και υπηρεσιών.

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Με την εμφάνιση του Υπολογιστικού Νέφους , μέ'σω των χαρακτηριστικών και των ιδιαιτεροτήτων που το διακρίνουν , έχουν εμφανιστεί καινούργια δεδομένα και ζητούμενα. Τα πλεονεκτήματα είναι αρκετά. Τα μειονεκτήματα ,είναι σαφώς ένα ανοικτό θέμα, μιας και σε οποιαδήποτε αρχή , δημιουργούνται απορίες, τίθενται ερωτήσεις και γενικά η δυναμική ενός νέου συστήματος προκαλεί τόσο δέος και θαυμασμό όσο και επιφύλαξη.

Στην παρούσα διπλωματική εργασία θα προσπαθήσουμε όσο το δυνατόν καλύτερα , αντικειμενικά, μέσω των επομένων κεφαλαίων , να εστιάσουμε σε ένα θέμα που θεωρούμε σημαντικό , να το αναλύσουμε και να προτείνουμε τη λύση μας.

Κεφάλαιο 3 Ανάγκη ύπαρξης Εργαλείου Συγκριτικής Αξιολόγησης Υπολογιστικών Νεφών

Ανάγκη

Οι υποδομές-ως-υπηρεσίες (IaaS) των Υπολογιστικών Νεφών εξελίσσονται σε ένα πλούσιο και ενεργό κλάδο των εμπορικών υπηρεσιών ΤΠΕ. Οι χρήστες του IaaS μπορούν να «παραγγέλλουν» υπηρεσίες " επεξεργασίας , αποθήκευσης, δικτύων και άλλων θεμελιωδών πόρων »]κατ' απαίτηση (on-demand) , δηλαδή , όταν χρειάζεται, για όσο διάστημα χρειαστεί , και να πληρώνουν μόνο για ό, τι είναι πραγματικά καταναλώνεται . Για την τελευταία πέντε χρόνια, εμπορικά IaaS Νέφη Υπολογιστών όπως το EC2 της Amazon έχουν αποκτήσει μια αυξανόμενη βάση χρηστών , από τις μικρές και μεσαίες επιχειρήσεις έως τους επιστημονικούς Χρήστες Υψηλών Αποδόσεων Υπολογιστικών Συστημάτων (HPC) . Ωστόσο , η αυξανόμενη υιοθέτηση των Υπολογιστικών Νεφών και ίσως ακόμη και τα

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

υποδείγματα τιμολόγησης εξαρτώνται από την ικανότητα των υποψηφίων χρηστών του Υπολογιστικού Νέφους για την ύπαρξη μίας συγκριτικής αξιολόγησης και να συγκρίνουν εμπορικές υπηρεσίες ΥΝ.

Η διαδικασία εφαρμογής μετρήσεων υπολογιστικών συστημάτων , αποτελεί πρωταρχικό στόχο μέτρησης της απόδοσής τους αλλά και άλλων μη μετρήσιμων χαρακτηριστικών , ώστε να είναι δυνατή η σύγκριση των ιδίων με άλλα συστήματα ή πολλές φορές και με συστήματα που ακολουθούν βιομηχανικά προσυμφωνηθέντα πρότυπα.

Τα εργαλεία μετρήσεων υπολογιστικών επιδόσεων (benchmarks), πλέον είναι ένα καθιερωμένο πρότυπο στο ερευνητικό τοπίο της υψηλής υπολογιστικής απόδοσης[7]. Οι υπηρεσίες των Υπολογιστικών Νεφών , τείνουν να έχουν αναπτυχθεί και να είναι προσανατολισμένες , σε ένα ευρύ φάσμα πιθανών εφαρμογών, πολλές από τις οποίες προσβλέπουν προς τη φιλοξενία επιχειρηματικών εφαρμογών. Καθώς η τεχνολογία των ΥΝ , γίνεται όλο και περισσότερο αποδεκτή , τόσο από επιχειρήσεις , όσο και από ιδιώτες, τόσο γίνεται και πιο επιτακτική η ανάγκη , για την δημιουργία σωστά-προσανατολισμένων εφαρμογών μετρήσεων (benchmark tools) , εργαλεία που θα προσφέρουν , σωστή και δίκαια αξιολόγηση τέτοιων συστημάτων , αλλά θα δημιουργούν και το κατώφλι από το οποίο μία τέτοια υπηρεσία θα είναι αποδεκτή και θα δύναται να βελτιστοποιηθεί. Η ποικιλία των επιλογών και των ρυθμίσεων των ΥΝ , και οι προσπάθειες που απαιτούνται για να φτάσουμε στο σημείο στο οποίο, παραδοσιακές εφαρμογές μετρήσεων μπορούν να εκτελεστούν , έχει διάφορες επιπτώσεις σχετικά με την ορθότητα της σύγκρισης - και , μάλιστα , για το ζήτημα της αναλογίας ποιότητα/κόστος (value for money). Τα εργαλεία συγκριτικών μετρήσεων των ΥΝ πρέπει να προσφέρουν δυνατότητες σύγκρισης σε όλα τα επίπεδα IaaS , καθώς επίσης πρέπει να προσφέρουν λεπτομερείς αναφορές για τον κύκλο

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

ζωής του , εν χρήσει και προς εξέταση , ΥΝ σύστημα. Τα υπάρχοντα εργαλεία μετρήσεων (benchmark tools) δεν προσφέρουν αυτήν τη δυνατότητα σύγκρισης. Πρέπει , έστω και στο ελάχιστο , να είμαστε σε θέση να κατανοήσουμε , κατά πόσο ένα «μέτριο» σύστημα ΥΝ , αποδίδει στις απαιτήσεις του χρήστη η/και της εταιρείας , σε από την άποψη του πραγματικού bandwidth , ώστε ο τελικός χρήστης να γνωρίζει με μία ελάχιστη απόκλιση , το χρόνο μεταφόρτωσης και μετεγκατάστασης δεδομένων προς/από ένα ΥΝ σύστημα , αλλά και το χρόνο επικοινωνίας μεταξύ ξεχωριστών παρόχων ΥΝ. Εν συνεχεία θα ήταν απαραίτητο να γνωρίζουμε κατά πόσο η διαθέσιμη μνήμη, επεξεργαστική ισχύς , ταχύτητα και χωρητικότητα δίσκου , είναι αυτές τις οποίες ο πάροχος δηλώνει πως παρέχει ή όχι , και αν όχι κατά πόσο αποκλίνουν από την πραγματικότητα , γεγονός το οποίο θα μπορούσε να οδηγήσει σε συμφόρηση των εικονικοποιημένων πόρων και κατ' επέκταση εκτέλεσης των εφαρμογών που βασίζονται σε αυτούς τους πόρους. Κατά τη σύναψη συμφωνίας μεταξύ παρόχου και πελάτη (ιδιώτης ή/και εταιρείας) , υπογράφεται ένα συμφωνητικό επιπέδου υπηρεσιών (Service Level Agreement – SLA). Η ανάγκη για την ύπαρξη αυτών των ανεξάρτητων εφαρμογών μετρήσεων προέρχεται από την τήρηση των προσυμφωνηθέντων υπηρεσιών , από μέρους του παρόχου , ώστε ο πελάτης να χρησιμοποιεί τις υπηρεσίες με την ελάχιστη δυνατή απόκλιση από το SLA.

Το εργαλείο συγκριτικής αξιολόγησης που έχουμε αναπτύξει

Στη διάρκεια αυτού του κεφαλαίου θα αναφερθούμε στο είδος του εργαλείου που έχουμε αναπτύξει , θα αναλύσουμε διεξοδικά τους λόγους που δικαιολογούν την ύπαρξη ενός τέτοιου εργαλείου. Επίσης θα αναφέρουμε τι έχει υλοποιηθεί μέχρι πρότινος πάνω στον τομέα αυτό στον επιστημονικό και εμπορικό τομέα , όπως και τι προσφέρει διαφορετικό από τις μέχρι τώρα λύσεις. Τέλος θα αναλύσουμε τις τεχνολογίες που

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

χρησιμοποιήσαμε για να δημιουργήσουμε αυτό το αποτέλεσμα και πώς αυτές συνδέονται μεταξύ τους , αλλά και ποιες είναι οι ανάγκες που οδήγησαν στο να επιλεγούν οι συγκεκριμένες τεχνολογίες.

Αποφασίσαμε να υλοποιήσουμε ένα εργαλείο συγκριτικής αξιολόγησης εφαρμογών που εκτελούνται σε εικονικά μηχανήματα (VM) , που έχουν δημιουργηθεί σε στο επίπεδο IaaS Υπολογιστικών Νεφών

Βασικοί Πάροχοι Υπολογιστικών Νεφών (IaaS)

Από τη γέννηση του Υπολογιστικού Νέφους , έχουν υπάρξει πολλές εταιρείες παροχής υπηρεσιών τόσο σε επίπεδο IaaS , PaaS και SaaS. Αρκετές από αυτές ήσαν διάπτοντες αστέρες με βραχύ χρόνο ζωής , άλλες όμως λόγω συγκεκριμένων επιχειρηματικών κινήσεων αλλά βασικά λόγω οράματος και συνέχειας , έγιναν κατεστημένο και κινητήρια δύναμη των υπολοίπων. Στην παράγραφο αυτή θα καταδείξουμε ποιοι είναι οι βασικότεροι πάροχοι και βασικές διαφορές ως προς την επιχειρηματική τους στρατηγική , ως προς το τι προσφέρουν.

Amazon Web Services



Εικόνα 7 Amazon Web Services[188]

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Η εταιρεία Amazon Web Services (AWS) είναι μία συλλογή απομακρυσμένων υπηρεσιοστρεφών υπηρεσιών , συνθέτοντας έτσι μία υπολογιστική πλατφόρμα Υπολογιστικού Νέφους που προσφέρονται μέσω Διαδικτύου από την Amazon.com. Οι πιο γνωστές από αυτές τις υπηρεσίες είναι η Amazon EC2 και Amazon S3. Η υπηρεσία αυτή φημίζεται ότι παρέχει μεγάλη υπολογιστική ικανότητα (πιθανότατα πολλούς εξυπηρετητές) πολύ πιο γρήγορα και οικονομικά από μία απλή υλοποίηση φυσικών εξυπηρετητών. Η AWS βρίσκεται σε 8 γεωγραφικές «περιοχές»: Ανατολικές Ηνωμένες Πολιτείες (Βόρεια Βιρτζίνια) , όπου βρίσκεται και η πλειοψηφία των AWS εξυπηρετητών, Δυτικές Ηνωμένες Πολιτείες (Βόρεια Καλιφόρνια , Όρεγκον) , Βραζιλία (Σάο Πάολο) , Ευρώπη (Ιρλανδία) , Νότια Ασία (Σιγκαπούρη) , Ανατολική Ασία (Τόκιο) και Αυστραλία (Σύδνεϋ). Υπάρχει επίσης και μία υπηρεσία με ονομασία GovCloud που είναι εγκατεστημένη στις ΗΠΑ και συγκεκριμένα στο βορειοδυτικό τμήμα , που προβλέπεται για την κυβέρνηση των ΗΠΑ. Κάθε μία από τις προαναφερθείσες περιοχές έχει πολλαπλές «Ζώνες Διαθεσιμότητας» (Availability Zones) , δηλαδή διακριτά κέντρα δεδομένων που παρέχουν υπηρεσίες AWS. Οι ζώνες διαθεσιμότητας απομονώνονται μεταξύ τους για την αποφυγή διακοπής υπηρεσιών , με φόβο εξάπλωσης τους και στις λοιπές. Αρκετές υπηρεσίες είναι διαθέσιμες σε αυτές τις ζώνες (πχ S3 , DynamoDB) ενώ άλλες μπορούν να ρυθμιστούν για να κλωνοποιούνται για να αποφεύγονται φαινόμενα μη άρνησης υπηρεσίας ή «πτώσης» των υπηρεσιών.

Τύποι Εικονικών Μηχανών και κόστη

Παρακάτω παρατίθενται οι τύποι των εικονικών μηχανών που μπορεί ένας πελάτης να ενοικιάσει για όσο το χρειάζεται (pay as you go) με τα συγκεκριμένα χαρακτηριστικά καθώς και το κόστος ενοικίασης για κάθε ένα από αυτά.

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Ομάδα Στιγμιότυπων	Τύπος Στιγμιότυπων (Ονομασία)
Γενικού σκοπού	m1.small m1.medium m1.large m1.xlarge m3.medium m3.large m3.xlarge m3.2xlarge
Βελτιστοποίηση ως προς την υπολογιστική ισχύ	c1.medium c1.xlarge c3.large c3.xlarge c3.2xlarge c3.4xlarge c3.8xlarge cc2.8xlarge
Βελτιστοποίηση ως προς τη μνήμη	m2.xlarge m2.2xlarge m2.4xlarge cr1.8xlarge
Βελτιστοποίηση ως προς την αποθήκευση	hi1.4xlarge hs1.8xlarge i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge
Micro Στιγμιότυπα	t1.micro
GPU Στιγμιότυπα	cg1.4xlarge g2.2xlarge

Παρακάτω παρατίθεται ένας πίνακας με τα κόστη των στιγμιότυπων για τη Ζώνη της Ευρώπης (Ιρλανδία) με λειτουργικό σύστημα Linux.[κόστη→Απρίλιος 2014]. Όπως είναι φυσικό αντίστοιχα κόστη υπάρχουν και για τις υπόλοιπες ζώνες. Θεωρούμε σημαντικό να μεταφράσουμε κάποιους όρους για να γίνει πιο άνετα η κατανόηση του παρακάτω πίνακα.

vCPU	Πλήθος Εικονικών Επεξεργαστών
ECU	Πραγματική Επεξεργαστική Ισχύς
Memory (GiB)	Μνήμη
Instance Storage (GB)	Χωρητικότητα

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage	
General Purpose - Current Generation					
m3.medium	1	3	3.75	1 x 4 SSD	\$0.077 per Hour
m3.large	2	6.5	7.5	1 x 32 SSD	\$0.154 per Hour
m3.xlarge	4	13	15	2 x 40 SSD	\$0.308 per Hour
m3.2xlarge	8	26	30	2 x 80 SSD	\$0.616 per Hour
General Purpose - Previous Generation					
m1.small	1	1	1.7	1 x 160	\$0.047 per Hour
m1.medium	1	2	3.75	1 x 410	\$0.095 per Hour
m1.large	2	4	7.5	2 x 420	\$0.190 per Hour
m1.xlarge	4	8	15	4 x 420	\$0.379 per Hour
Compute Optimized - Current Generation					
c3.large	2	7	3.75	2 x 16 SSD	\$0.120 per Hour
c3.xlarge	4	14	7.5	2 x 40 SSD	\$0.239 per Hour
c3.2xlarge	8	28	15	2 x 80 SSD	\$0.478 per Hour
c3.4xlarge	16	55	30	2 x 160 SSD	\$0.956 per Hour
c3.8xlarge	32	108	60	2 x 320 SSD	\$1.912 per Hour
Compute Optimized - Previous Generation					
c1.medium	2	5	1.7	1 x 350	\$0.148 per Hour
c1.xlarge	8	20	7	4 x 420	\$0.592 per Hour
cc2.8xlarge	32	88	60.5	4 x 840	\$2.250 per Hour
GPU Instances - Current Generation					
g2.2xlarge	8	26	15	60 SSD	\$0.702 per Hour
GPU Instances - Previous Generation					
cg1.4xlarge	16	33.5	22.5	2 x 840	\$2.360 per Hour
Memory Optimized - Current Generation					
r3.large *	2	TBD	15	1 x 32 SSD	\$0.195 per Hour
r3.xlarge *	4	TBD	30.5	1 x 80 SSD	\$0.390 per Hour
r3.2xlarge *	8	TBD	61	1 x 160 SSD	\$0.780 per Hour
r3.4xlarge *	16	TBD	122	1 x 320 SSD	\$1.560 per Hour
r3.8xlarge *	32	TBD	244	2 x 320 SSD	\$3.120 per Hour
Memory Optimized - Previous Generation					
m2.xlarge	2	6.5	17.1	1 x 420	\$0.275 per Hour
m2.2xlarge	4	13	34.2	1 x 850	\$0.550 per Hour
m2.4xlarge	8	26	68.4	2 x 840	\$1.100 per Hour
cr1.8xlarge	32	88	244	2 x 120 SSD	\$3.750 per Hour
Storage Optimized - Current Generation					
i2.xlarge	4	14	30.5	1 x 800 SSD	\$0.938 per Hour

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

i2.2xlarge	8	27	61	2 x 800 SSD	\$1.876 per Hour
i2.4xlarge	16	53	122	4 x 800 SSD	\$3.751 per Hour
i2.8xlarge	32	104	244	8 x 800 SSD	\$7.502 per Hour
hs1.8xlarge	16	35	117	24 x 2048	\$4.900 per Hour
Storage Optimized - Previous Generation					
hi1.4xlarge	16	35	60.5	2 x 1024 SSD	\$3.100 per Hour
Micro Instances					
t1.micro	1	Variable	0.615	EBS Only	\$0.020 per Hour

Rackspace Cloud



Εικόνα 8 Rackspace

Η εταιρεία Rackspace Cloud είναι ένα σύνολο προϊόντων και υπηρεσιών υπηρεσιοστρεφών υπηρεσιών Υπολογιστικών Νεφών, με έδρα τις ΗΠΑ. Από τις υπηρεσίες που προσφέρει οι πιο γνωστές είναι οι : Cloud Sites για υπηρεσίες ιστοσελίδων, Cloud Files για υπηρεσίες αποθήκευσης δεδομένων, Cloud Servers για εικονικούς ιδιωτικούς εξυπηρετητές, ισορροπιστές φόρτου εργασίας, βάσεις δεδομένων, δημιουργία αντιγράφων ασφαλείας και παρακολούθησης εργασιών (monitoring). Η εταιρεία είναι, κατά τα

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

λεγόμενά της, ο παγκόσμιος ηγέτης στην υβριδική τεχνολογία Υπολογιστικών Νεφών. Είναι ο ιδρυτής της ανοικτής πλατφόρμας Υπηρεσιών Νέφους Openstack. Το Openstack είναι ένα έργο όπως ονομάζεται – cloud computing – που έχει ως στόχο την «πανταχού παρούσα ανοικτού κώδικα πλατφόρμα υπηρεσιοστρεφών ΥΝ. Ενεργεί κυρίως ως υποδομή-ως-υπηρεσία (IaaS) , είναι δωρεάν και το λογισμικό ανοικτού κώδικα λειτουργεί κάτω από τους όρους του Apache Foundation και τους όρους χρήσης του. Παρομοίως όπως και η AWS λειτουργεί με «ζώνες». Η εταιρεία διατηρεί 9 data centers σε 6 γεωγραφικές περιφέρειες , Σικάγο (ΗΠΑ) , Ντάλλας (ΗΠΑ) , Βόρεια Βιρτζίνια (ΗΠΑ) , Λονδίνο (Ηνωμένο Βασίλειο) , Χονγκ-Κονγκ (Κίνα) , Σύδνεϋ (Αυστραλία). Η εταιρεία Rackspace στην πολιτική χρέωσης αλλά και διάθεσης των εικονικών μηχανών ακολουθεί τα παρακάτω. Παραθέτουμε τον παρακάτω πίνακα σύμφωνα με τις τρέχουσες χρεώσεις (Απρίλιος 2014).

Τύποι Εικονικών Μηχανών και κόστη

Επιδόσεις 1 Flavor Class

Καταλληλότερο για web servers, πολλαπλής επεξεργασίας, δικτυακές συσκευές, μικρές βάσεις δεδομένων, και πιο γενικού φόρτου εργασίας χρήσης υπολογιστές.

Χαρακτηριστικά:

- Χαμηλή έως μέτρια Μνήμη
- Εξαιρετικής απόδοσης, RAID 10-protected SSD αποθηκευτικό χώρο
- 10-Gigabit Δικτύωση
- Από 1–8 vCPUs*
- Σκληροί Δίσκοι με χαρακτηριστικά: I/O up to ~35,000 4K random read IOPS και ~35,000 4K random write IOPS.

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Χρεώσεις Επιδόσεις 1 Flavor Class για εικονικά μηχανήματα με Λειτουργικό Σύστημα Linux

1GB	2GB	4GB	8GB
Performance	Performance	Performance	Performance
1GB RAM	2GB RAM	4GB RAM	8GB RAM
1 vCPU1	2 vCPUs1	4 vCPUs1	8 vCPUs1
20GB SSD system disk2	40GB SSD system disk2	40GB SSD system disk2	40GB SSD system disk2
—	20GB SSD data disk3	40GB SSD data disk3	80GB SSD data disk3
200Mb/s network bandwidth4	400Mb/s network bandwidth4	800Mb/s network bandwidth4	1,600Mb/s network bandwidth4
INFRASTRUCTURE SERVICE LEVEL	INFRASTRUCTURE SERVICE LEVEL	INFRASTRUCTURE SERVICE LEVEL	INFRASTRUCTURE SERVICE LEVEL
MANAGED CLOUD SERVICE LEVELS	MANAGED CLOUD SERVICE LEVELS	MANAGED CLOUD SERVICE LEVELS	MANAGED CLOUD SERVICE LEVELS
\$0.16/Hr.	\$0.20/Hr.	\$0.28/Hr.	\$0.44/Hr.

Επιδόσεις 2 Flavor Class

Καταλληλότερο για web servers, πολλαπλής επεξεργασίας, δικτυακές συσκευές, μικρές βάσεις δεδομένων, και πιο γενικού φόρτου εργασίας χρήσης υπολογιστές.

Χαρακτηριστικά:

- Μέτρια έως υψηλή Μνήμη
- Εξαιρετικής απόδοσης, RAID 10-protected SSD αποθηκευτικό χώρο
- 10-Gigabit Δικτύωση
- Από 4–32 vCPUs*
- Σκληροί Δίσκοι με χαρακτηριστικά: I/O up to ~80,000 4K random read IOPS και ~70,000 4K random write IOPS.
- Αποκλειστική μίσθωσης 120 ρυθμισμένη εικονική μηχανή (Flavor) για πιο σταθερή και αποδοτική λειτουργία.

Χρεώσεις Επιδόσεις 2 Flavor Class για εικονικά μηχανήματα με Λειτουργικό Σύστημα Linux

15GB	30GB	60GB	90GB	120GB
Performance	Performance	Performance	Performance	Performance
15GB RAM	30GB RAM	60GB RAM	90GB RAM	120GB RAM
4 vCPUs¹	8 vCPUs¹	16 vCPUs¹	24 vCPUs¹	32 vCPUs¹
40GB SSD system disk²	40GB SSD system disk²	40GB SSD system disk²	40GB SSD system disk²	40GB SSD system disk²
150GB SSD data disk³	300GB SSD data disk³	600GB SSD	900GB SSD	1,200GB SSD

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

		(2 x 300GB) data disk3	(3 x 300GB) data disk3	(4 x 300GB) data disk3
1,250Mb/s network bandwidth4	2,500Mb/s network bandwidth4	5,000Mb/s network bandwidth4	7,500Mb/s network bandwidth4	10,000Mb/s network bandwidth4
MANAGED CLOUD SERVICE LEVELS	MANAGED CLOUD SERVICE LEVELS	MANAGED CLOUD SERVICE LEVELS	MANAGED CLOUD SERVICE LEVELS	MANAGED CLOUD SERVICE LEVELS
\$0.85/Hr.	\$1.70/Hr.	\$3.40/Hr.	\$5.10/Hr.	\$6.80/Hr.

Vcloud



Εικόνα 9 Vcloud

Η Vcloud είναι μία πρωτοβουλία από την εταιρεία VMware, η οποία επιτρέπει στους πελάτες της να μετακυλήσουν τις εργασίες τους από ένα εσωτερικό Υπολογιστικό

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Νέφος σε ένα εξωτερικό. Ο στόχος της πρωτοβουλίας είναι να παρέχει τη δύναμη των υπηρεσιοστρεφών υπηρεσιών Νέφους που επιτρέπεται μέσα από την εικονικοποίηση. Η αρχιτεκτονική του VCloud βασίζεται στην τεχνολογία vShield[192] που ενεργεί ως προεπιλεγμένη πύλη του δικτύου. Αυτή η εικονική πύλη υλοποιείται σε έναν ελεύθερο υποδοχέα (host) του συστήματος και διοχετεύει τις υπηρεσίες του σε αυτό το host ή και σε άλλους υλοποιώντας με αυτόν τον τρόπο την έννοια του υβριδικού μοντέλου IaaS. Ένας πελάτης χρησιμοποιώντας το δικό του εξειδικευμένο υλικό έχει τη δυνατότητα να εκτελεί ένα ιδιωτικό νέφος έχοντας όμως τη δυνατότητα να κινείται σε δημόσια νέφη κατά το δοκούν. Έτσι ο χρήστης έχει πολύ μεγάλη ευελιξία γιατί όταν εξωτερικοί παράγοντες οδηγούν σε εξαιρετικές απαιτήσεις για αποθηκευτικό χώρο ή/και υπολογιστική ισχύ , τότε μπορεί να χρησιμοποιεί πόρους του νέφους για τις ανάγκες του. Βάσει αυτής της λειτουργικότητας δεν νοείται εικονική μηχανή (VM) αλλά , μία μέθοδος έξοδος προς το δημόσια νέφος μέσω λογισμικού.

Κόστη

PRODUCT TITLE	License Price (US ONLY**)	1 Year Support & Subscription	
VMware vCloud Suite Standard Foundational solution with infrastructure provisioning, health and capacity, security and extensibility to public clouds	USD 4,995.00	Basic	USD 1,049.00
		Production	USD 1,249.00
VMware vCloud Suite Advanced Solution for governed, enterprise cloud, with <u>capacity management</u> , cloud management extensibility and increased automation	USD 7,495.00	Basic	USD 1,574.00
		Production	USD 1,874.00
VMware vCloud Suite Enterprise Comprehensive solution for resilient, compliant and secure private cloud with <u>disaster recovery</u> automation, visibility into application performance, and regulatory compliance	USD 11,495.00	Basic	USD 2,414.00
		Production	USD 2,874.00

Εικόνα 10 Vmcloud Κόστη

Ερευνητικές – Εμπορικές Διεργασίες αντίστοιχες με το εργαλείο που έχουμε αναπτύξει (State of the Art)

Εδώ θα αναφέρουμε εργαλεία που έχουν αναπτυχθεί και είναι διαθέσιμα. Συγκεκριμένα εργαλεία συγκριτικής αξιολόγησης Υπηρεσιοστρεφών Υπηρεσιών Νέφους.

CloudHarmony



Εικόνα 11 CloudHarmony

Το cloudharmony παρέχει αντικειμενική ανάλυση των επιδόσεων μέσα από συγκριτική αξιολόγηση παρόχων Υπηρεσιών Νέφους. Η πρόθεση των δημιουργών είναι να αποτελέσει ανεξάρτητη πηγή , μη-προκατειλημμένη και αντικειμενική για τις μετρήσεις των επιδόσεων. Το εν λόγω εργαλείο δεν έχει καμία σχέση με την κυριότητα από κάποιον και δε χρηματοδοτείται από κάποια εταιρεία. Τα κριτήρια αξιολόγησης που προβλέπονται από το εργαλείο CloudHarmony εμπίπτουν σε 3 κατηγορίες:

1. Performance Benchmarking
2. Network Benchmarking
3. Uptime Monitoring

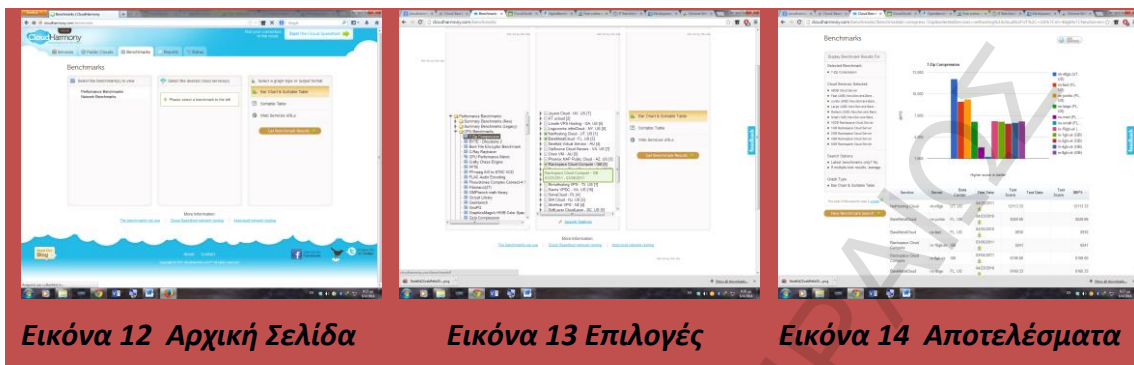
“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Οι δημιουργοί του εργαλείου δηλώνουν ότι υπάρχουν 6 ερωτήματα τα οποία πρέπει να τεθούν κατά την εξέταση των απαιτήσεων ενός συγκριτικού τεστ αξιολόγησης. Απαντώντας σε αυτές τις ερωτήσεις ο χρήστης, του παρέχεται μία καλύτερη κατανόηση σχετικά με την εγκυρότητα και τη δυνατότητα εφαρμογής των απαιτήσεων.

1. Ποια είναι η ερώτηση που πρέπει να απαντήσει η συγκριτική αξιολόγηση; Για παράδειγμα θέλουμε να δούμε αν μία υπηρεσία είναι 10 φορές πιο γρήγορη από την ίδια όταν εκτελούνται σε 2 διαφορετικές εικονικές μηχανές;
2. Ποια είναι η πραγματική μέτρηση; Η ταχύτητα δικτύου αφορά πραγματικά το δίκτυο ή στην απάντηση υπεισήλθαν και εξωτερικοί παράγοντες (πχ ταχύτητα δίσκου).
3. Είναι μία αντικειμενική μέτρηση και εξάγει συμπεράσματα και αποτελέσματα απαντώντας επακριβώς το ερώτημα που τέθηκε;
4. Είναι το επίπεδο αναφοράς επί ίσοις όροις; Υπάρχουν παράγοντες που παρέχουν αθέμιτο πλεονέκτημα σε μία από τις δύο ή περισσότερες εφαρμογές που δοκιμάζονται;
5. Ήταν τα δεδομένα που αναφέρθηκαν με ακρίβεια; Η συγκριτική αξιολόγηση οδηγεί συχνά σε μεγάλα σύνολα δεδομένων. Συνοψίζοντας δεδομένα συνοπτικά μπορεί να οδηγήσει σε σφάλματα. Αποτελέσματα που χρήζουν περαιτέρω μελέτης μπορεί να παραληφθούν ή ακόμα να γίνουν λάθος μαθηματικοί υπολογισμοί.
6. Έχει σημασία για αυτόν που εκτελεί τη συγκριτική αξιολόγηση; Το τελικό ερώτημα που τίθεται είναι, υποθέτοντας ότι τα αποτελέσματα είναι έγκυρα, πραγματικά τι σημαίνουν;

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Στις παρακάτω εικόνες παρατίθενται υποδείγματα συγκριτικών αξιολογήσεων με το εν λόγω εργαλείο.



Ενώ το εργαλείο που παρουσιάζει η ομάδα υλοποίησης του CloudHarmony είναι αρκετά διαδραστικό και δίνει στο χρήστη τη δυνατότητα να εκματαλευτεί μία πλειάδα από συγκριτικές αξιολογήσεις, εντούτοις έπειτα από λίγη ώρα ενσχόλησης, ο χρήστης διαπιστώνει ότι η αξιολόγηση δε γίνεται εκείνη τη στιγμή, αλλά εξάγονται αποτελέσματα συγκρίσεων μπορεί και 3 με 4 χρόνια πίσω. Θεωρούμε λοιπόν ότι ο δοκιμή δε γίνεται εκείνη τη στιγμή, αλλά παίρνουμε αποτελέσματα από τεστ που έχουν εκτελεστεί και απλά γίνεται μία ανάκτηση σε πραγματικό χρόνο των αποτελεσμάτων. Επίσης ο χρήστης δε γνωρίζει τον τύπο των εικονικών μηχανημάτων που γίνεται η συγκριτική αξιολόγηση (περιβάλλον IaaS).

Cloudsleuth

Η εταιρεία Cloudsleuth προσφέρει τη δυνατότητα απεικόνισης των χρόνων απόκρισης και διαθεσιμότητας για τους παρόχους υπηρεσιών ΥΝ, αποτυπώνοντας την πληροφορία γραφικά στον παγκόσμιο χάρτη. Η παρακάτω εικόνα δείχνει ένα παράδειγμα του χρόνου απόκρισης υπηρεσιών PaaS και IaaS για την Ευρώπη τις τελευταίες έξι(6)ώρες για τον πάροχο Google App Engine. Το εργαλείο αυτό έχει κυκλοφορήσει από την εταιρεία

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Compuware. Αναπτύσσει μία ανεξάρτητη άποψη για το πώς μία καλά δομημένη εφαρμογή στο Υπολογιστικό Νέφος εκτελείται. Παρέχει μια εικόνα του πόσο αποτελεσματικά τα διανεμόμενα μέρη μίας εφαρμογής συνεργάζονται. Το Cloudsleuth μπορεί να δώσει απτά αποτελέσματα στο χρήστη ως προς το κατά πόσο μία εφαρμογή έχει απόδοση καλύτερη στον περιηγητή ιστοσελίδων Microsoft Explorer ή Mozilla Firefox. Επικεντρώνεται στις πλατφόρμες GoGrid, Microsoft Azure, or Google App Engine όσον αφορά τα Υπολογιστικά Νέφη.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Click to go back, hold to see history

CLOUDSLEUTH
DECODING THE MYSTERIES OF THE CLOUD

Brought to you by **Compuware**

Applications Companies Forums Blogs About

Resources, Companies, Services Search

SPOTLIGHT
It's here. Our new app:
OUTAGE ANALYZER (beta)
Track third-party web service outages and instantly know their cause and impact. It's free - try it now. >>

Global Provider View
Compare IaaS and PaaS performance >>

CDN Performance Analyzer
Visualize the effects of CDN performance >>

EXPERT CLOUD PERSPECTIVES
Breaking Down Hybrid Cloud Planning
There's no doubt that hybrid cloud planning is a complex process. Most companies enter the hybrid cloud space wanting the freedom of the open...
read more
Posted by: Dawn Athnam 4.15.2013

Apps for cloud computing
One trend that's gained popularity is the availability of cloud hosted applications designed for business. These applications are commonly housed...
read more
Posted by: Dawn Athnam 2.4.2013

What happens to your cloud after you die?
With digital data moving to the cloud, we are freeing up storage space and making the management of

LATEST CLOUD RESOURCES
GMO Cloud Case Studies
Posted by: GMO Cloud America, Inc.
See GMO Cloud's existing and well-satisfied customers and read how they use GMO Cloud Hosting...
read more

Q&A with Level One Sites
Posted by: eApps
Level One Sites is the leading provider of software solutions for Reebok/CrossFit franchise...
read more

NEWEST CONTRIBUTING PARTNER

PARTNERS
Browse cloud solutions here!

CONNECTRIA
HOSTING

ROSETTA

eApps

city|cloud
Your data center in a browser

Cable&Wireless
Worldwide

DAJUBEO
Cloud Managed

enomaly
elastic computing

CLOUDSLEUTH NEWSLETTER
Sign up for our monthly cloud information
Name: updates.

Εικόνα 15 CloudSleuth Αρχική

Global Provider View Compare the end-user experience of PaaS and IaaS providers from around the world.

Metric: **Response Time** **Availability** Time Frame: **6 hour** Locations: **World** View: [Icons]

Cloud Providers / Google App Engine

City	Availability
Mesa	100 %
Los Angeles	100 %
Frankfurt	100 %
Copenhagen	100 %
Paris	100 %
Berlin	100 %
Bern	100 %
London	100 %
Istanbul	100 %
Oslo	100 %
Bangalore	100 %
Mumbai	100 %
Helsinki	100 %
Singapore	100 %
Hong Kong	100 %
Tokyo - NTT	100 %
Sao Paulo	100 %
Tokyo - KDDI	100 %
Buenos Aires	100 %
Sydney	100 %
Shanghai	0.00 %

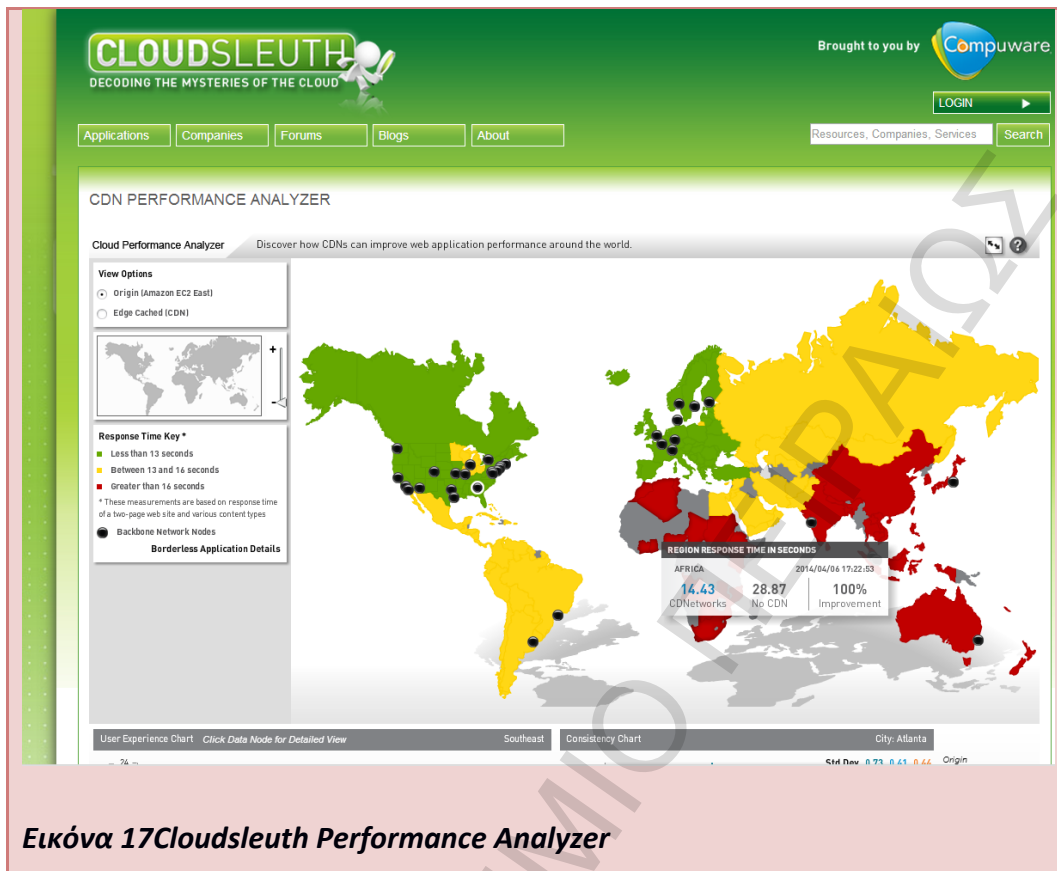
06 Apr 2014 11:29 To 06 Apr 2014 17:29 (GMT-0300)

Backbone Network: ● > 99 (%) ● 99-95 (%) ● < 95 (%)

Powered by **Compuware GOMEZ**

Εικόνα 16 CloudSleuth Availability

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”



Εικόνα 17 Cloudsleuth Performance Analyzer

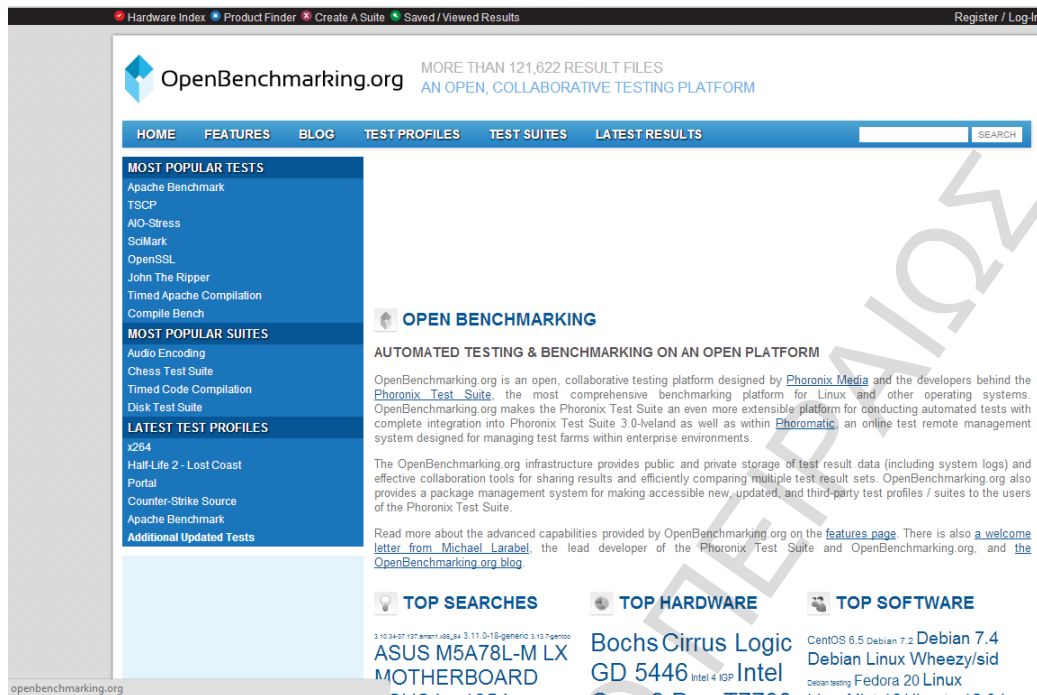
Openbenchmarking



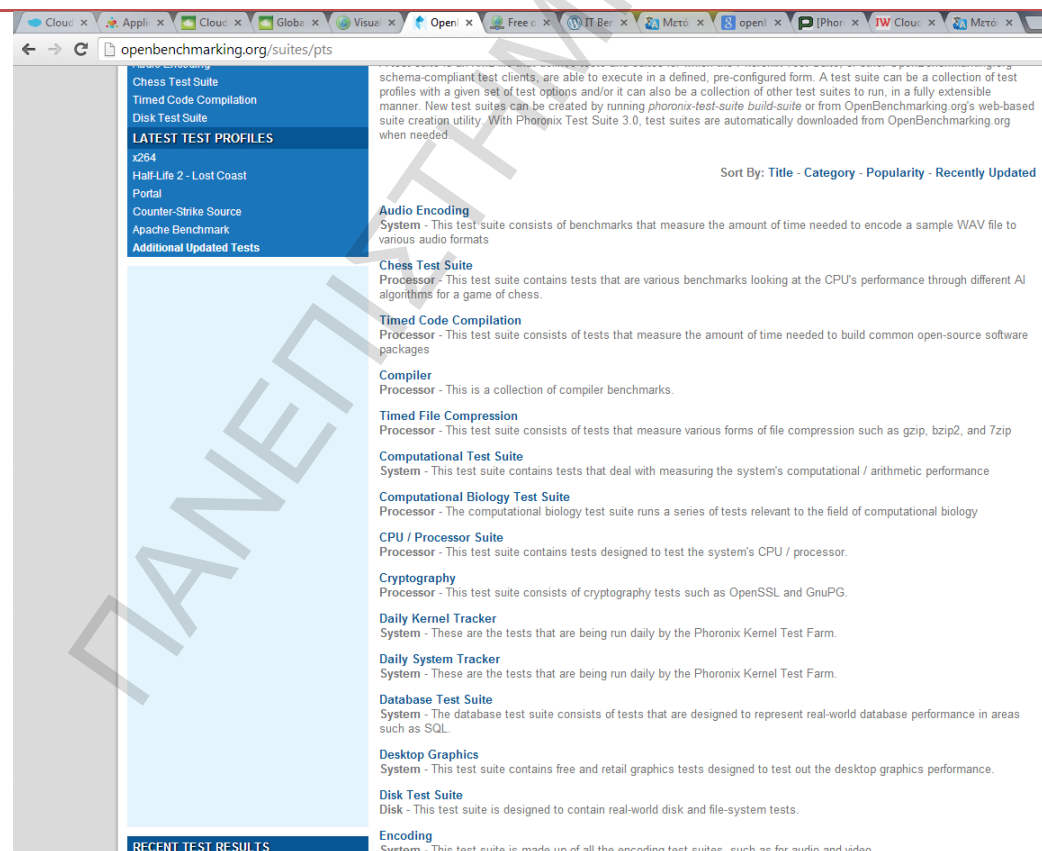
Εικόνα 18 Openbenchmarking.org

Η ιστοσελίδα openbenchmarking.org προσφέρει πληροφορίες από διάφορα συγκριτικά τεστ που έχουν πραγματοποιηθεί σε παρόχους ΥΝ με βάση το Phoronix Test Suite. Τα αποτελέσματα αυτών των τεστ είναι διαθέσιμα και αφορούν κυρίως τους παρόχους AWS και Rackspace. Είναι ένα έργο που παρέχει μια τεράστια κοινότητα-θόλο συνεχώς αυξανόμενων δεδομένων δοκιμών και άλλες πληροφορίες για το υλικό (όπως αρχεία καταγραφής του συστήματος, αρχεία καταγραφής αποτελεσμάτων, και αρχεία ρυθμίσεων), που το καθιστά άψογα εύκολο για συγκριτική αξιολόγηση των επιδόσεων πολλών εικονικών μηχανών σε επίπεδο IaaS, με λειτουργικό σύστημα Linux, BSD, και άλλα λειτουργικά συστήματα. Με το OpenBenchmarking.org είναι επίσης εύκολο να γίνουν δοκιμές υλικού και επιτρέπει τη σύγκριση των αποδόσεων ενός δεδομένου προ-ρυθμισμένου λογισμικού / υλικού έναντι άλλων. Ολοκληρωμένες δυνατότητες αναζήτησης κάνουν εύκολη την αναζήτηση σχετικών πληροφοριών από το αποθετήριο που διαθέτει, ενώ το σύστημα σύγκρισης που είναι όμοια με εικονικό καλάθι αγορών στο διαδίκτυο εξασφαλίζει με ακρίβεια αποτελέσματα κοιτάζοντας τις πιο σημαντικές πτυχές της απόδοσης για τις ανάγκες του χρήστη ή του οργανισμού.

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”



Εικόνα 19 Openbenchmarking Αρχική



Εικόνα 20 Openbenchmarking Επιλογή Δοκιμής

Συμπέρασμα - Τι διαφορετικό προσφέρει το δικό μας εργαλείο

Έπειτα από την παρουσίαση των υλοποιήσεων – εργαλείων που υπάρχουν στο διαδίκτυο, είναι δόκιμο να αναφέρουμε για ποιο λόγο δημιουργήθηκε το δικό μας εργαλείο.

Όπως είδαμε και στις προηγούμενες ενότητες, αλλά αν και ο χρήστης θελήσει να ασχοληθεί και να αφιερώσει λίγο χρόνο, εξερευνώντας λίγο περισσότερο τις παραπάνω υλοποιήσεις θα παρατηρήσει ότι υπάρχουν κάποια βασικά κενά. Αρχικά ο χρήστης δεν έχει τη δυνατότητα να εκτελέσει οποιουδήποτε είδους συγκριτική αξιολόγηση μεταξύ εικονικών μηχανών που έχει ρυθμίσει ο ίδιος. Δεν μπορεί να επέμβει στο υλικό του υπολογιστή με κανέναν τρόπο, πόσο μάλλον δε στις ρυθμίσεις της καθεαυτό εφαρμογής, προκειμένου να εκτελεστεί η δοκιμή. Αν παραδείγματος χάριν επιθυμεί να εκτελεστεί μία δοκιμή, σε διαφορετικά μηχανήματα με την ίδια βάση δεδομένων, με τον ίδιο αριθμό καταχωρήσεων, τότε αυτό είναι αδύνατο. Επίσης, ένα μεγάλο μειονέκτημα είναι ότι, οι δοκιμές δεν εκτελούνται on-the-fly (εκείνη τη στιγμή) στα πιο πολλά εργαλεία. Έχει τη δυνατότητα να δει γραφικές αναπαραστάσεις περασμένων ημερών-μηνών-ετών και να τις συγκρίνει.

Το δικό μας εργαλείο, δίνει πλήρη αυτονομία στο χρήστη. Δίνει τη δυνατότητα να επιλέξει σε ποιον πάροχο θα γίνει η αξιολόγηση, τι είδους εικονικό μηχανήμα θα δημιουργηθεί με συγκεκριμένες από το χρήστη ρυθμίσεις. Έπειτα, ο χρήστης δίνει όλες τις παραμέτρους της δοκιμής, μία φορά. Όταν το μηχανήμα θα δημιουργηθεί και θα ξεκινήσει το λειτουργικό σύστημα, τότε με αυτοματοποιημένες διαδικασίες (scripts) το λειτουργικό αναλαμβάνει να μεταφορτώσει και να εγκαταστήσει προγράμματα, βιβλιοθήκες αλλά και συγκεκριμένες ενημερώσεις. Με την ολοκλήρωση αυτής της διαδικασίας, εκτελείται η

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

δοκιμή, χρησιμοποιώντας τις ρυθμίσεις του χρήστη. Ο ίδιος από τη στιγμή που θα επιλέξει να εκτελεστεί η δοκιμή, δεν είναι υποχρεωμένος να κάνει καμία άλλη ενέργεια, παρά να περιμένει να ολοκληρωθεί. Όταν γίνει αυτό, τότε με γραφικές αναπαραστάσεις μπορεί να αξιολογήσει τα αποτελέσματα της συγκριτικού αξιολόγησης που επέλεξε να κάνει. Επίσης αν το επιθυμεί μπορεί από προηγούμενα αποθηκευμένα αξιολογικά κριτήρια, να εμφανίσει με τη μορφή γραφημάτων τα αποτελέσματα.

Εν κατακλείδι, το εργαλείο μας, δίνει στο χρήστη πλήρη ανεξαρτησία και ελευθερία για τον τρόπο, το χρόνο και τα χαρακτηριστικά κάθε αξιολόγησης.

ΤΟ ΕΡΓΑΛΕΙΟ

Στο κεφάλαιο αυτό θα παρουσιάσουμε την εφαρμογή που έχουμε υλοποιήσει. Συγκεκριμένα. Θα αναλύσουμε την αρχιτεκτονική και τις τεχνολογίες που ακολουθήσαμε και χρησιμοποιήσαμε, έπειτα θα περάσουμε σε μία λεπτομερή περιγραφή της λειτουργικότητας της εφαρμογής, ενώ τέλος θα επεξηγήσουμε αναλυτικά πως υλοποιήθηκε η εν λόγω εφαρμογή.

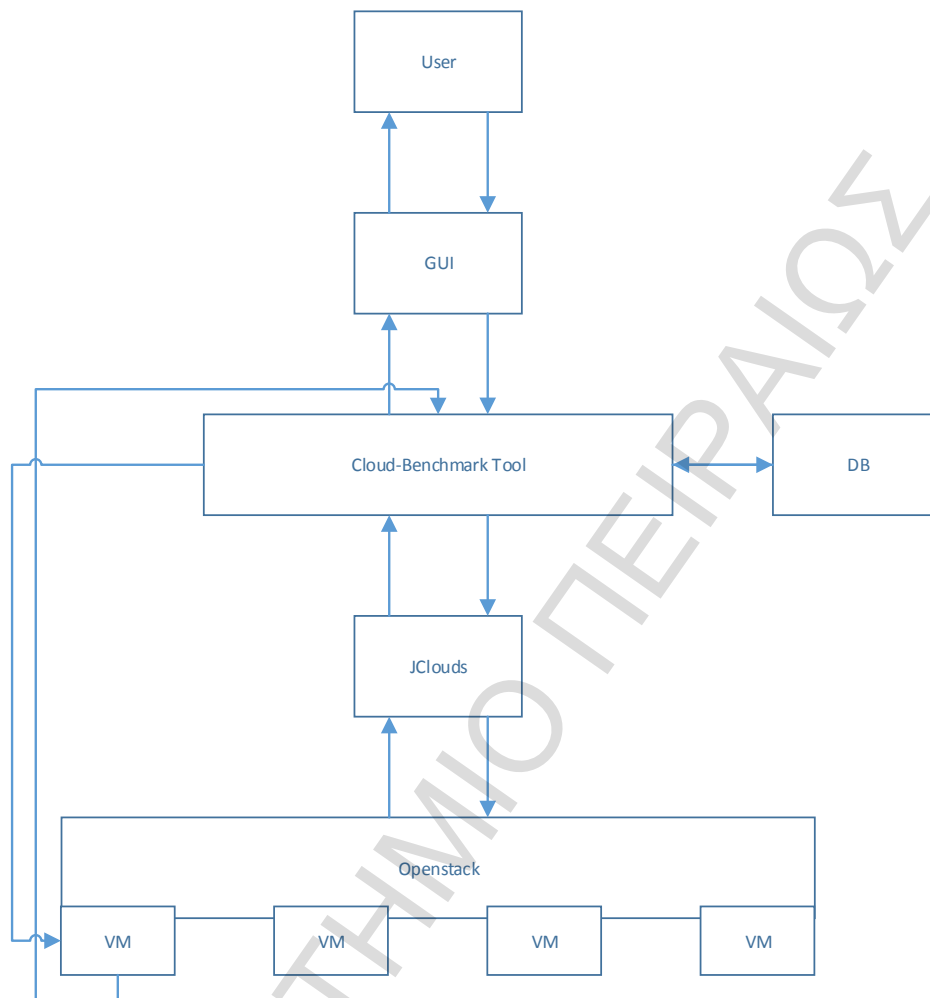
Ολοκληρωμένη Λύση Εφαρμογής

Στο παρακάτω σχεδιάγραμμα παρουσιάζουμε με ποιον τρόπο λειτουργεί η εφαρμογή που έχουμε δημιουργήσει. Πιο συγκεκριμένα ο χρήστης με την είσοδό του στο Γραφικό περιβάλλον της εφαρμογής (GUI), μπορεί να εκτελέσει κάποιο από τα συνολικά τρία (3) τεστ. Με την επιλογή κάποιου, το εργαλείο που βρίσκεται στο backend (Cloud Benchmark Tool), χρησιμοποιεί σαν ενδιάμεσο το open-source πρόγραμμα JClouds, που είναι υπεύθυνο για να επικοινωνεί με τον αντίστοιχο cloud provider.

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Όταν δημιουργηθεί ένα εικονικό μηχάνημα(VM) στον provider(στην υλοποίησή μας Openstack) , πλέον η επικοινωνία γίνεται μόνο μεταξύ VM και Tool. Με την επιτυχή ολοκλήρωση ενός test , τα αποτελέσματα παρουσιάζονται στην οθόνη του χρήστη , παράλληλα όμως αποθηκεύονται και σε μία βάση δεδομένων , όπου και διατηρούνται για περαιτέρω πιθανή μελέτη. Εδώ θα πρέπει να κάνουμε σαφές , πως το Tool είναι σε θέση να εκτελεί πολλαπλά test , τόσο με συγκεκριμένη συχνότητα αλλά και με επαναλήψεις κάθε φορά (πχ ανά τρεις ώρες να εκτελούνται 5 τεστ για DB). Στο χρήστη όμως εμφανίζονται τα αποτελέσματα κάθε φορά του τελευταίου τεστ. Επίσης, παρέχεται στον χρήστη η δυνατότητα να δει αποτελέσματα προηγούμενων τεστ που έχει εκτελέσει ώστε να μπορεί να αναλύσει τα αποτελέσματα αυτά σε καλύτερο βάθος συγκρίνοντας τα με διαφορές παραμέτρους.

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”



Εικόνα 21 Ολοκληρωμένη Λύση

Ανάλυση της Εφαρμογής

Σκοπός μας στο κεφάλαιο αυτό είναι να παρουσιάσουμε αναλυτικά την αρχιτεκτονική (τεχνολογίες) πίσω από την εφαρμογή μας, όπως επίσης τη λειτουργικότητα (how to use) και την υλοποίηση (programming – UML Modelling) της.

Αρχιτεκτονική

Client-Server

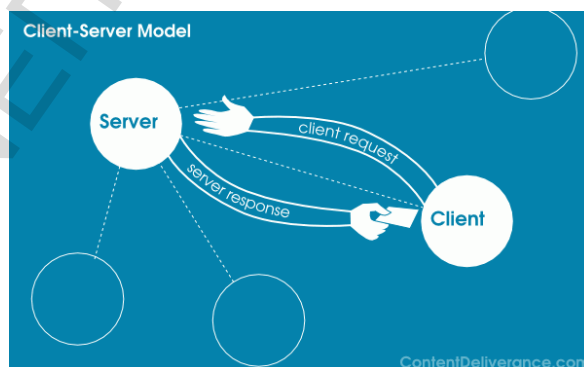
Αποτελεί μία δομή καταμεμημένων εφαρμογών, όπου οι διεργασίες και ο φόρτος κατανομονται ανάμεσα σε παρόχους υπηρεσιών (servers) και χρήστες υπηρεσιών (clients)[30]. Αποτελείται στην ουσία από δυο πλευρές που αλληλεπιδρούν με μια λογική

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

αιτήματος-απάντησης , καθώς ο client κάνει αίτηση για μια ενέργεια ή μια υπηρεσία από το server και αναμένει τη σχετική απάντησή του.

Η αρχιτεκτονική αυτή απεικονίζει τη σχέση μεταξύ δύο υπολογιστικών προγραμμάτων , στην οποία το ένα πρόγραμμα , το πρόγραμμα client , ζητάει υπηρεσίες από το άλλο πρόγραμμα , το πρόγραμμα server.

Η αρχιτεκτονική αυτή απεικονίζει τη σχέση μεταξύ δυο υπολογιστικών προγραμμάτων, στην οποία το ένα πρόγραμμα, το client πρόγραμμα, ζητάει υπηρεσίες από το άλλο πρόγραμμα, το server πρόγραμμα. Βασικές δικτυακές λειτουργίες όπως η ανταλλαγή δεδομένων, η πρόσβαση στο Web και η πρόσβαση σε βάσεις δεδομένων, στηρίζονται στο client-server μοντέλο. Είναι σημαντικό να καταλάβουμε ότι η σχέση μεταξύ του πελάτη(client) και του διακομιστή (server) είναι μια σχέση εντολοδότησης/απόκρισης. Σε κάθε συναλλαγή , ο client ξεκινά την αίτηση και ο server ανταποκρίνεται ανάλογα . Ένας server δεν μπορεί να ξεκινήσει διάλογο με τους clients. Δεδομένου ότι ο client και ο server είναι οντότητες λογισμικού που μπορεί να βρίσκονται σε οποιαδήποτε hardware υλοποίηση .



Εικόνα 22 Μοντελο Client-Server[31]

Υπάρχουν διαφορετικά λειτουργικά μοντέλα στην αρχιτεκτονική client-server . Αναφορικά , θα τονίσουμε πως υπάρχει ξεχωριστή υλοποίηση για κάθε εφαρμογή . Αρχικά

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

υπάρχει το μοντέλο των στατικών ιστοσελίδων (HTML) , όπου ο client στέλνει ένα αίτημα για μια HTML ιστοσελίδα και ο server επιστρέφει την απάντηση (σελίδα) και κλείνει η σύνδεση μεταξύ των δύο. Εν συνεχεία ένα δεύτερο μοντέλο που ακολουθείται είναι εκείνο κατά το οποίο , ο client στέλνει στο server ένα αίτημα (POST) , και ο δεύτερος δεν στέλνει ολόκληρη τη σελίδα , αλλά στέλνει πίσω μία απάντηση έχοντας καταναλώσει αντίστοιχη υπολογιστική ισχύ (πχ σε μία ιστοσελίδα αναζήτησης). Τρίτο μοντέλο αρχιτεκτονικής είναι , το λεγόμενο server-side μοντέλο , όπου η απάντηση στον client παράγεται δυναμικά. Ως παράδειγμα τέτοιων τεχνολογιών είναι οι HyperText Preprocessor (PHP) , Java Server Pages

(JSP), Active Server Pages (ASP), Server Side Includes (SSI). Χαρακτηριστικό της τελευταίας τεχνολογίας είναι ότι η προγραμματιστική λογική δεν εμφανίζεται (πχ JAVA Programming), αλλά αφού έχουν εκτελεστεί οι επεξεργασίες στον Server , οι πληροφορίες αποστέλλονται στον Client με τη μορφή HTML. Σαφώς στην αρχιτεκτονική Client- Server , υπάρχουν πλεονεκτήματα και μειονεκτήματα στην εφαρμογή της , αλλά η ανάλυση τους φεύγει από το scope της παρούσας διπλωματικής.

Java EE

Κατά τη διάρκεια της υλοποίησης της εφαρμογής , αποφασίσαμε να εντάξουμε την JAVA EE (Java Enterprise Edition-J2EE) , ως το κύριο εργαλείο ανάπτυξης κώδικα. Αυτό οφείλεται σε πολλά από τα χαρακτηριστικά που τη διακρίνουν. Αναλυτικότερα, η JAVA EE , είναι το πρότυπο για community-driver enterprise software [23]. Η Java EE διαφέρει από την βασική έκδοση της Java από το γεγονός ότι προσθέτει βιβλιοθήκες οι οποίες παρέχουν λειτουργικότητα ώστε να αναπτυχθεί διαδικτυακό, πολυμερές λογισμικό σε Java, βασισμένο σε αυτοτελή μέρη που τρέχουν σε application servers.

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Αποτελεί μία συλλογή από τις προδιαγραφές για την ανάπτυξη επιχειρηματικών εφαρμογών. Οι προδιαγραφές αυτές ορίζονται από την εταιρεία Sun® , που περιγράφουν υπηρεσίες, διεπαφές προγραμματισμού εφαρμογών (API's) και πρωτόκολλα. Οι 13 βασικές τεχνολογίες που συνθέτουν την J2EE [32] είναι οι: JDBC , JNDI , EJBs , RMI , JSP , servlets της Java , XML , JMS , Java IDL , JTS , JTA , JavaMail και JAF . Η λογική πίσω από αυτό είναι ότι ο πάροχος προμηθεύει έναν τρόπο υλοποίησης των παραπάνω διεπαφών , μέσω κλάσεων που ορίζουν τα αντίστοιχα interfaces.



Εικόνα 23 Java EE[33]

Τα πιο περίπλοκα (προγραμματιστικά) συστήματα είναι κατανομημένα. Δυστυχώς υπάρχει μία δυσκολία στην ανάπτυξη τέτοιων συστημάτων χωρίς τη χρήση JAVAEE, η οποία μπορεί να «κρύβει» τις πολυπλοκότητες ενός προς-υλοποίηση συστήματος (scalability with services like JMS, distributed transactions, distributed scope management, κλπ) και ουσιαστικά ο προγραμματιστής να εστιάζει στο επιχειρηματικά προγραμματιστικό κομμάτι και όχι στο τεχνικά προγραμματιστικό (implementing via interfaces vs code from scratch).

Model-View-Controller

Στον αντικειμενοστραφή προγραμματισμό , το μοντέλο MVC , (model-view-controller) , είναι μία μεθοδολογία ή ένα πρότυπο σχεδιασμού για τη δημιουργία επιτυχημένων και αποδοτικών UI (user interfaces) σε σχέση με τα μοντέλα δεδομένων από

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

τα οποία εξαρτώνται. Η τεχνολογία αυτή χρησιμοποιείται ευρέως στην ανάπτυξη προγραμμάτων με γλώσσες προγραμματισμού όπως Java, Smalltalk, C και C++.

Στην βάση του , το MVC [34], όπως έπειτα έγινε και με τα υπόλοιπα frameworks που επηρέασε ήταν αυτό που θα αποκαλούσαμε Ξεχωριστή Παρουσίαση (Separated Presentation). Η ιδέα πίσω από αυτό το μοντέλο είναι να υπάρχει πλήρης διαχωρισμός των αντικειμένων που συμμετέχουν στην εφαρμογή (Domain objects) σε σχέση με αυτά που θα παρουσιαστούν στην οθόνη (presentation objects). Τα Domain objects πρέπει να είναι εντελώς αυτόνομα χωρίς καμία αναφορά στα presentation objects, θα πρέπει να είναι σε θέση να υποστηρίζουν πολλαπλές παρουσιάσεις στην οθόνη και πολλές φορές ταυτόχρονα. Αυτή η προσέγγιση ήταν ένα σημαντικό κομμάτι της κουλτούρας των χρηστών/επαγγελματιών του Unix.

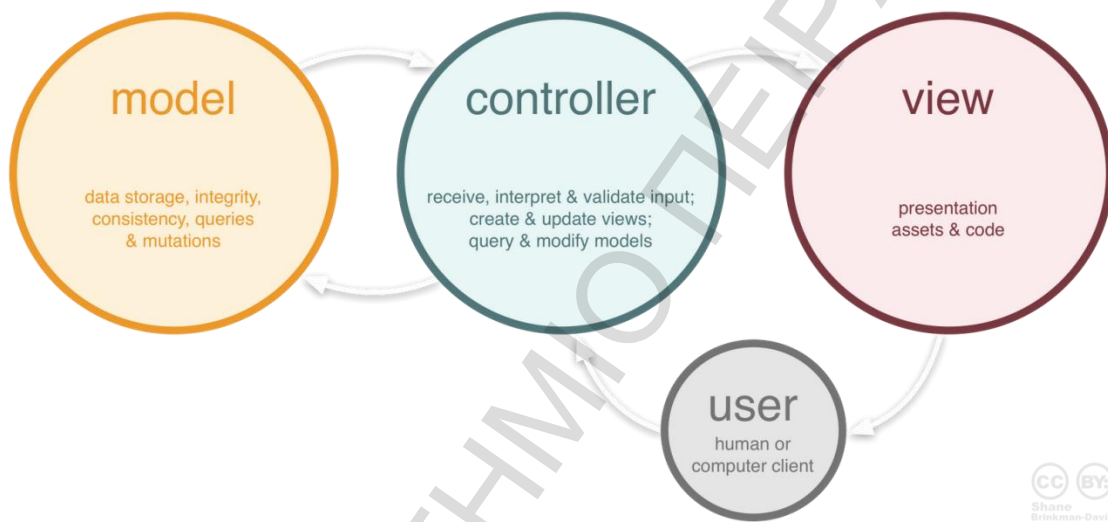
Το πρότυπο MVC έχει προτιμηθεί από πολλούς επαγγελματίες προγραμματιστές ως το πιο reusable code εργαλείο, για την επαναχρησιμοποίηση αντικειμένων κώδικα μέσω ενός λογικού μοτίβου, γεγονός το οποίο μπορεί να μειώσει δραστικά το χρόνο συγγραφής εφαρμογών που χρησιμοποιούν διεπαφές χρήστη.

Το model-view-controller πρότυπο, προτείνει τρεις βασικές δομές/συνιστώσες οι οποίες πρέπει να χρησιμοποιηθούν κατά την ανάπτυξη λογισμικού.

- Το Μοντέλο (Model). Αντιπροσωπεύει τη λογική δομή των δεδομένων σε μία εφαρμογή καθώς και την high-level class που συνδέεται με αυτό. Αυτό το μοντέλο δεν παρέχει καμία πληροφορία για το περιβάλλον εργασίας του χρήστη (User Interface).

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

- Την γραφική άποψη(View). Μία συλλογή από κλάσεις που αντιπροσωπεύουν τα στοιχεία που θα αναπαρασταθούν στο UI (πχ buttons, boxes κλπ).
- Τον Ελεγκτή (Controller). Αντιπροσωπεύει την κλάση που χρησιμοποιείται για τη σύνδεση και των επικοινωνιών των δύο παραπάνω δομικών στοιχείων.



Εικόνα 24 The MVC Model[25]

Spring MVC Framework

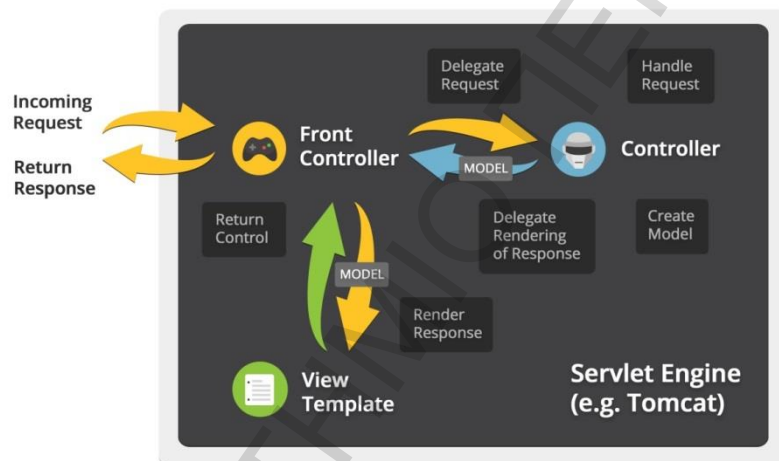
Στην παράγραφο αυτή θα αναφέρουμε βασικά χαρακτηριστικά του Framework στο οποίο βασιστήκαμε για να αναπτύξουμε το front-end της εργασίας μας. Αυτό είναι το Spring MVC Framework[35]. Είναι ένα framework σχεδιασμένο πάνω σε έναν DispatcherServlet ο οποίος αποστέλλει αιτήσεις προς τους διαχειριστές (handlers) , με αντίστοιχες δυνατότητες ρύθμισης τόσο για το mapping όσο και για την ανάλυση , τις τοπικές ρυθμίσεις, την ημέρα/ώρα καθώς επίσης και για χαρακτηριστικά μεταφόρτωσης αρχείων.

Ο default handler βασίζεται στους σχολιασμούς(Annotations) @Controller και @RequestMapping , προσφέροντας έτσι ένα φάσμα ευέλικτων μεθόδων χειρισμού. Με το Spring MVC Framework , μπορούμε να χρησιμοποιήσουμε κάθε αντικείμενο κλάσης ως

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

εντολή ή ως αντικείμενο φόρμας. Δε χρειάζεται να εφαρμόσουμε κάποιο συγκεκριμένο interface ή κλάση.

Η διαδικασία απεικόνισης στο Spring MVC Framework είναι εξίσου ευέλικτη. Ένας Controller είναι υπεύθυνος για το mapping των δεδομένων αποστέλλοντας τα στο αντίστοιχο view name (στην εργασία μας τα αντίστοιχα .jsp).

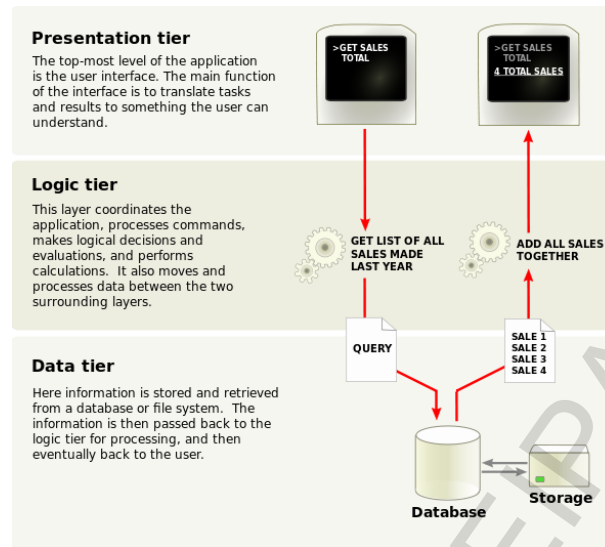


Εικόνα 25 Spring MVC[36]

3-Tier Applications

Στην τελευταία παράγραφο της αρχιτεκτονικής που ακολουθήσαμε θα αναφερθούμε στην 3-tier αρχιτεκτονική που ακολουθήσαμε στην πορεία της διπλωματικής μας εργασίας και για ποιο λόγο προχωρήσαμε με αυτή τη τεχνολογία.

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”



Εικόνα 26 3-Tier Application[37]

Η τεχνολογία της 3-tier application βασίζεται στα παρακάτω:

Αρχιτεκτονική Client-Server

Κάθε επίπεδο (Παρουσίασης (Presentation), Λογικής (Logic), Δεδομένων (Data)) πρέπει και είναι ανεξάρτητο και να μην εκθέτει τυχόν εξαρτήσεις που έχει από την εφαρμογή.

Επίπεδα που δε σχετίζονται δεν πρέπει να επικοινωνούν.

Τυχόν αλλαγές στην πλατφόρμα εκτέλεσης της εφαρμογής πρέπει να επηρεάζουν μόνο το συγκεκριμένο επίπεδο το οποίο αφορούν οι αλλαγές.

Με την εισαγωγή του μεσαίου στρώματος (Business Logic Tier) , ο client χειρίζεται μόνο τη λογική παρουσίασης (presentation logic). Αυτό σημαίνει ότι λόγω της ελάχιστης επικοινωνίας που πρέπει να υπάρχει μεταξύ client και μεσαίου στρώματος , η

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

εφαρμογή client γίνεται όλο και πιο “ελαφριά”. Για παράδειγμα , ένα πρόγραμμα περιήγησης στο Διαδίκτυο που επιτρέπει την πλοήγηση και την αναζήτηση πληροφοριών , σχεδόν άμεσα χωρίς καθυστέρηση.

Όσο και περισσότεροι χρήστες έχουν πρόσβαση σε ένα 3-tier σύστημα, το τελευταίο μπορεί να γίνει σαφώς πιο επεκτάσιμο , μιας όλο και περισσότερο μεσαία επίπεδα μπορούν να προστεθούν ώστε να εξασφαλισθεί καλύτερη απόδοση του συστήματος.

Η ασφάλεια , βασικός παράγοντας εύρυθμης λειτουργίας ενός συστήματος, όσον αφορά την 3-tier τεχνολογία , είναι σαφώς βελτιωμένη μιας και το μεσαίο στρώμα προστατεύει το στρώμα της ΒΔ.

Το μοναδικό μειονέκτημα το οποίο μπορεί να θεωρηθεί ότι έχει ένα σύστημα υλοποιημένο με 3-tier τεχνολογία είναι ότι, πρόσθετες βαθμίδες αυξάνουν την πολυπλοκότητα του συστήματος.

Bootstrap front-end framework

Για την καλύτερη απεικόνιση και πιο πλούσια γραφικά αναπαράσταση όλου του framework αποφασίσαμε να εντάξουμε το bootstrap twitter[38]. Είναι μία δωρεάν συλλογή από εργαλεία για την δημιουργία ιστοσελίδων. Περιέχει πρότυπα τόσο HTML όσο και CSS, για κουμπιά , φόρμες ,και άλλα στοιχεία που βοηθούν στη διεπαφή με το χρήστη.



Εικόνα 27 Twitter Bootstrap

Περιγραφή Λειτουργικότητας

Στο σημείο αυτό θα παρουσιάσουμε τον τρόπο με το οποίο ο χρήστης χρησιμοποιεί την εφαρμογή μας.

Reports

Σε πρώτο στάδιο ο χρήστης, είναι σε θέση αν δεν θέλει να εκτελέσει κάποιο test , να ανατρέξει σε κάποιο ή κάποια από τα προηγούμενα test προκειμένου να μελετήσει τα αποτελέσματά τους μέσω πινάκων ή να εξάγει κάποια γραφήματα με αυτά. Πρέπει να επιλέξει την επιλογή Settings→Reports όπου και θα εμφανιστεί μπροστά του μία φόρμα προς συμπλήρωση. Υποχρεωτικά πρέπει να επιλέξει την ημερομηνία έναρξης του test (Start Date) και αν επιθυμεί μπορεί να επιλέξει να του εμφανιστούν test που πραγματοποιήθηκαν μέσα σε κάποιες μέρες (End Date). Αν πάλι επιθυμεί test για μία ημέρα δεν έχει πάρα να επιλέξει στην επιλογή End Date , την ίδια ημερομηνία με την Start Date. Έπειτα ο χρήστης καλείται να καταχωρίσει τον τύπο του VM Instance πού επιθυμεί, βάσει του οποίου έγινε το test. Στο σημείο αυτό είναι αναγκαίο να αναφέρουμε ότι με την

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

ολοκλήρωση της φόρμας και την υποβολής της, παράγονται δύο αποτελέσματα. Το πρώτο είναι μία γραφική παράσταση, η οποία για κάθε τεστ που έχει πραγματοποιηθεί, εμφανίζει αν οι συγκεκριμένες ρυθμίσεις ήταν αποδοτικές βάσει ενός κριτηρίου, το οποίο έχουμε συγκεκριμενοποιήσει για κάθε τεστ. Συγκεκριμένα όταν έχει εκτελεστεί ένα benchmark που αφορά βάση δεδομένων, ως κρίσιμο παράγοντα θεωρούμε το OVERALLThroughput. Throughput είναι το κόστος των επιτυχημένων ανταλλαγών πληροφορίας διαμέσου ενός καναλιού επικοινωνίας [22]. Για το κομμάτι του web-serving έχουμε λάβει ως κρίσιμο παράγοντα το ρυθμό επεξεργασίας Kb/sec. Στα test του map/reduce λόγω του ότι εκτελούνται 4 διαφορετικά, κάθε ένα από αυτά έχει το δικό του κρίσιμο παράγοντα. Έτσι για το NN Benchmark έχουμε το Average Exec Time (open_read, delete, write_close για κάθε ξεχωριστό αρχείο), για το DFSIO έχουμε το Test Execute Time, για το Teragen το compute base splits, ενώ τέλος για το MRBench, το AvgTime.

Έτσι, όταν υποβληθεί η φόρμα, εκτελείται το αντίστοιχο query και συλλέγονται οι αντίστοιχες πληροφορίες. Αυτές, για κάθε τεστ που ικανοποιεί τις παραμέτρους, και για κάθε δείκτη, δημιουργείται αντίστοιχη γραφική παράσταση (chart). Έτσι ο χρήστης γνωρίζει ποιο τεστ ήταν αποδοτικότερο τη συγκεκριμένη ημερομηνία.

Παράλληλα με την γραφική παράσταση έχουμε επιλέξει να υλοποιήσουμε, έναν μετρητή απόδοσης τεστ. Για να εκφράσουμε καλύτερα την απόδοση μιας υπηρεσίας, σκεφτήκαμε να χρησιμοποιήσουμε μία συνάρτηση που θα μπορέσει να αποδώσει καλύτερα τα παρακάτω:

- Να δέχεται και να αποτυπώνει καλύτερα το workload ενός τεστ
- Να δέχεται και να λαμβάνει υπόψη το κόστος χρήσης ενός συγκεκριμένου workload από ένα VM
- Να αποτυπώνει καλύτερα την απόδοση για το συγκεκριμένο workload

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

- Να παραμετροποιούνται πολύ περισσότερο κάποιες μετρικές βάσει επιλογών του χρήστη.
- Αν η διαίσθηση “καλύτερο και ακριβότερο” VM θα έχει καλύτερη απόδοση.

Έχοντας υπόψη τα παραπάνω σημεία, σκεφτήκαμε να θεωρήσουμε κάποια στοιχεία σαν “θετικούς παράγοντες” (όπως το workload) και σαν αρνητικό θεωρήσαμε το κόστος. Περαιτέρω, θα πρέπει να γίνει κανονικοποίηση για να μπορούμε να εξάγουμε σωστά και όσο το δυνατό πιο λεπτομερή αποτελέσματα. Για παράδειγμα, αν ο χρήστης θεωρήσει σα φόρτο 90% το μέσο χρόνο ολοκλήρωσης του τεστ για μία ΒΔ και 10% το κόστος, και αν οι δύο παραπάνω τιμές είναι 523 sec και 0.60 (€/ώρα) * 8.7 ώρες = 5.23€, καταλαβαίνουμε ότι αυτά τα δύο ποσά είναι εκ διαμέτρου ανόμοια. Επιλέξαμε να εφαρμόσουμε κανονικοποίηση 1-2 και για τις δύο τιμές.

Καταλήξαμε λοιπόν στην παρακάτω συνάρτηση απόδοσης ενός VM για ένα δεδομένο workload.

$$SE = \frac{1}{w_1 * delay + w_2 * Cost}$$

Εικόνα 28 SE Συνάρτηση

SE= Service Efficiency

w1: weight 1 (ποσοστό φόρτου 1)

w2: weight 2(ποσοστό φόρτου 2)

delay: Μέσος Όρος συγκεκριμένου δείκτη τεστ

cost: Το κόστος του αντίστοιχου VM που έχει επιλέξει ο χρήστης στη φόρμα

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Στην περίπτωση που ο χρήστης επιθυμεί να εκτελέσει κάποιο από διαθέσιμα Test , στην αρχική οθόνη έχει τη δυνατότητα να επιλέξει μεταξύ των:

- DB
- Web Serving
- Map Reduce

Τα υπόλοιπα test είναι για μελλοντική υλοποίηση.

Κάθε test έχει τα δικά του χαρακτηριστικά που πρέπει να συμπληρωθούν από το χρήστη και σε αυτά θα αναφερθούμε αναλυτικά. Υπάρχουν και κοινά βήματα τα οποία κρίνουμε σκόπιμο να τα εξετάσουμε συνολικά με το πέρας των ξεχωριστών χαρακτηριστικών.

DB

Με την επιλογή της εκτέλεσης του test των βάσεων δεδομένων , ο χρήστης ουσιαστικά οδηγείται μέσω τριών βημάτων να δώσει το configuration του test. Στο πρώτο βήμα καλείται να επιλέξει μία Βάση Δεδομένων από τις ακόλουθες

- MongoDB
- MySQL
- Postgre
- Cassandra

Έπειτα δίνει τον αριθμό των thread που θα εκτελούνται στο test , ενώ έπειτα καταχωρεί το target του test. Τέλος επιλέγει από ένα πτυσσόμενο μενού το φορτίο (workload). Οι επιλογές που έχει είναι τέσσερις (4). Τρεις (3) προκαθορισμένες και μία (1) χειροκίνητη.

Συγκεκριμένα:

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

- Light Workload
- Medium Workload
- Heavy Workload
- Very Heavy Workload

Αξίζει να σημειωθεί στο σημείο αυτό , ότι σε κάποια από τα πεδία που συμπληρώνει ο χρήστης επιτρέπονται μόνο αριθμητικοί , και συγκεκριμένα ακέραιοι αριθμοί , και έχει προβλεφθεί το validation για αυτό το σκοπό.

Τα βήματα 2 και 3 είναι κοινά και για τα τρία τεστ και θα περιγραφούν σε μία κοινή παράγραφο.

Map Reduce

Στο τεστ αυτό , το οποίο ομοίως εκτελείται σε τρία βήματα , το πρώτο είναι και το πιο απλό και λιγότερο παραμετροποιήσιμο. Ο χρήστης στο βήμα αυτό καλείται απλά και μόνο να επιλέξει ποιου είδους τεστ θα εκτελεστεί. Τα τεστ είναι τέσσερα και είναι όπως κάτωθι:

- TestDFSIO
- NNbench
- Teragen
- MrBench

Ομοίως όπως και στην προηγούμενη παράγραφο τα βήματα 2 και 3 θα περιγραφούν παρακάτω.

Web Serving

Το τεστ αυτό αφορά μετρήσεις Web Servers. Στο πρώτο , από ομοίως τρία βήματα , ο χρήστης καλείται να επιλέξει σε ποιον Server θα πραγματοποιηθεί το τεστ , ενώ

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

ακολουθως πρέπει να συμπληρώσει τον αριθμό των threads , requests (πόσα ταυτόχρονα αιτήματα) και concurrent clients (πόσοι ταυτόχρονοι συνδεδεμένοι clients στον server).

Κοινή ομάδα (Βήματα 2 και 3)

Με το πέρας του πρώτου βήματος της παραμετροποίησης, ο χρήστης καλείται στο να συμπληρώσει την επανληπτικότητα του τεστ (Measurement Parameters) και σε ποιον πάροχο θα εκτελεστεί το τεστ (Target Cloud Provider). Συγκεκριμένα, στο βήμα Measurement Parameters , ο χρήστης υποχρεούται να συμπληρώσει την ημερομηνία έναρξης του τεστ (Date) , την ώρα (Time) , τη συχνότητα (Centenary) και τέλος την επαναλήπτικότητα (Frequency). Στο τρίτο βήμα επιλέγει σε ποιον cloud provider θα γίνει το τεστ. Οι διαθέσιμοι είναι :

- Rackspace
- Openstack
- Amazon WS
- VCloud

Στην παρούσα διπλωματική εργασία όλα τα τεστ και οι μετρήσεις που θα παρουσιαστούν , πραγματοποιήθηκαν σε μία private υλοποίηση cloud OpenStack.

Με την επιλογή του αντίστοιχου provider , εμφανίζονται τα πεδία υποχρεωτικής συμπλήρωσης που αφορούν τα διαπιστευτήρια του χρήστη στο VM.

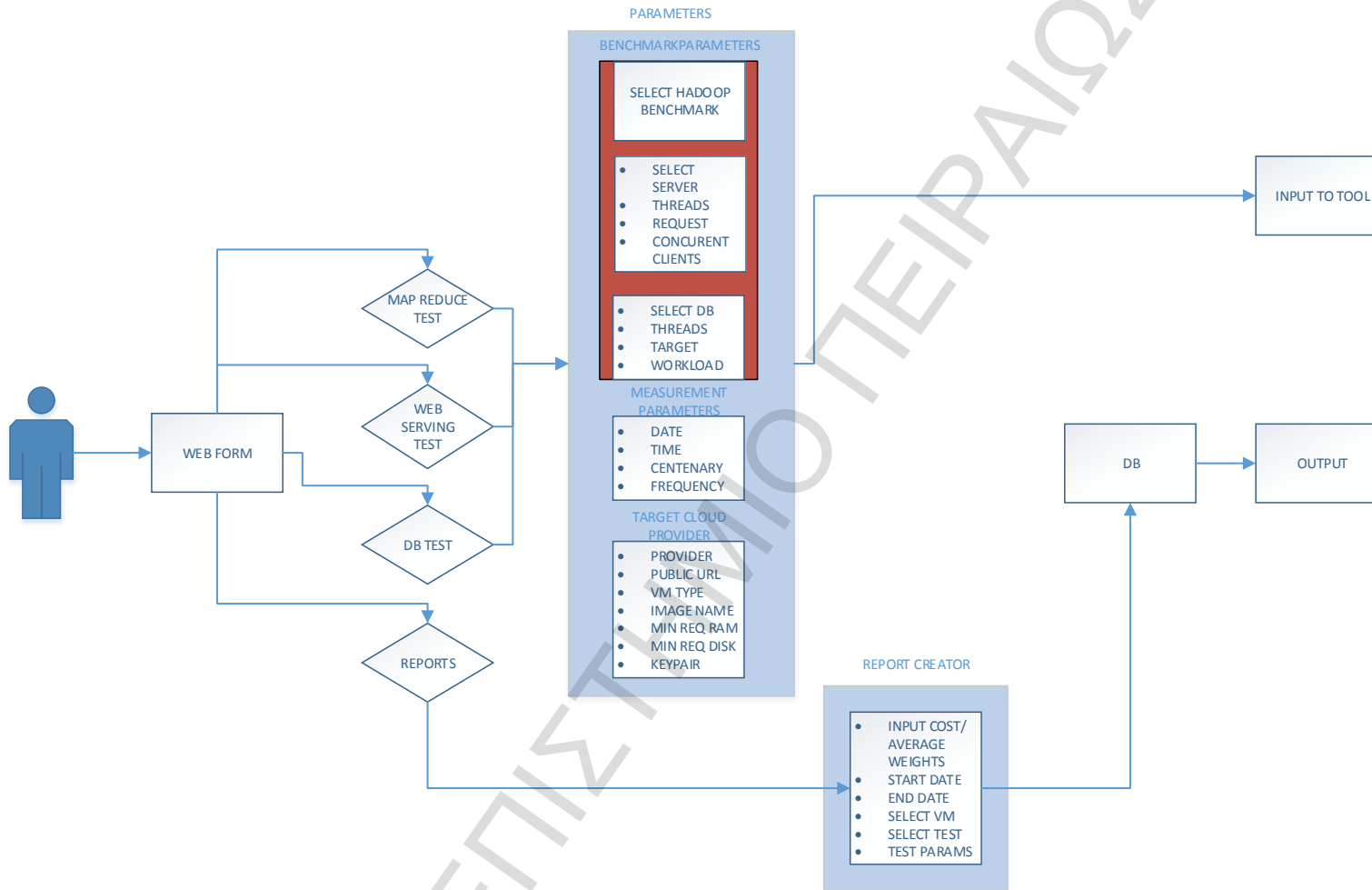
Έπειτα , ο χρήστης συμπληρώνει μία public URL , που θα χρησιμοποιεί για τη σύνδεση του με το εικονικό μηχάνημα , ενώ έπειτα επιλέγει σε ποια VM Type θα πραγματοποιηθεί το τεστ. Εδώ θα θέλαμε να τονίσουμε , πως ανάλογα του κάθε provider εμφανίζονται και οι αντίστοιχες υλοποιήσεις VM Type. Εν συνεχεία , ο χρήστης

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

πληκτρολογεί το όνομα του λειτουργικού συστήματος με το οποίο θα ξεκινήσει το VM με τα αντίστοιχα πεδία ελάχιστης μνήμης (Minimum Required Ram) και ελάχιστη χωρητικότητα σκληρού δίσκου (Minimum Required Disk Space) τα οποία θα δεσμευτούν κατά την στιγμή δημιουργίας του VM. Τέλος για να μπορέσει να πραγματοποιηθεί η ασφαλής επικοινωνία μεταξύ του VM που θα εκτελεστούν τα τεστ και του υπολογιστή που εκτελείται η εφαρμογή , πρέπει να εφαρμοσθεί SSH επικοινωνία των δύο. Στην τελευταία επιλογή ο χρήστης επιλέγει αν υπάρχει από προηγούμενα τεστ (Use your keypair) ,κλειδί (keypair) ή επιθυμεί τη δημιουργία νέου (Create new).

Ανεξάρτητα με το είδος του τεστ που θα εκτελεστεί , όταν ο χρήστης , έχει ολοκληρώσει την καταχώρισή του σε όλα τα πεδία , έχει τη δυνατότητα να επιλέξει το button Finish , προκειμένου να ξεκινήσει το τεστ. Για λόγους ασφαλείας αν κάποιο από τα πεδία δεν έχει συμπληρωθεί , τότε η εφαρμογή δεν επιτρέπει στο χρήστη να προχωρήσει παρακάτω. Έτσι πρέπει να ελέγξει τις καταχωρίσεις του και να επανέλθει ξανά στο τελικό βήμα.

Ακολουθεί σχηματική αναπαράσταση , η οποία επεξηγεί τη ροή ακολουθίας εκτέλεσης του τεστ.



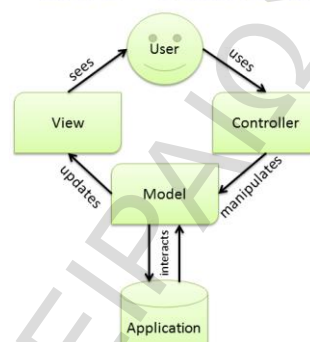
Εικόνα 29 Περιγραφή

Υλοποίηση

Στο παρόν κεφάλαιο θα μελετήσουμε τις κλάσεις που συμμετέχουν στο κομμάτι του frontend του GUI, καθώς επίσης και τον τρόπο που αυτές, όπως και οι μέθοδοί τους συνδέονται μεταξύ τους προκειμένου να παραχθεί το τελικό αποτέλεσμα στην οθόνη του χρήστη.

Η διπλανή εικόνα δείχνει συνοπτικά πως συνδέονται στο μοντέλο MVC τα τρία μέρη (έχει πραγματοποιηθεί αναλυτική περιγραφή σε προηγούμενο κεφάλαιο).

MVC – Data flow



Εικόνα 30 MVC

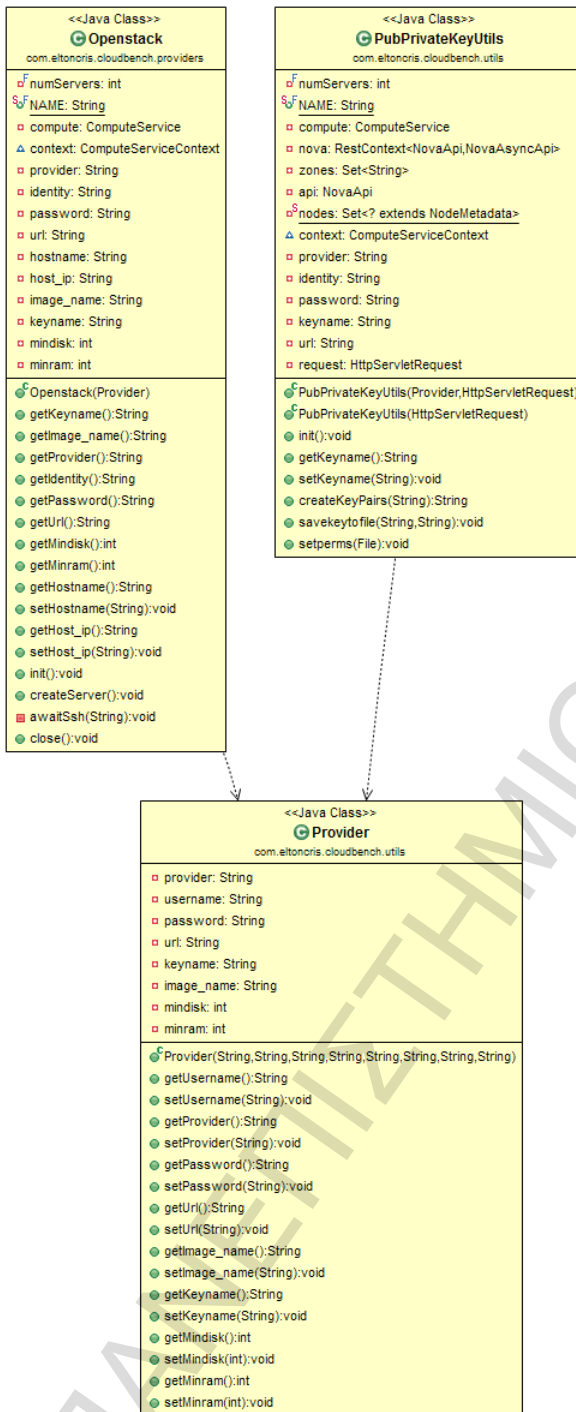
Για την πληρέστερη επεξήγηση των χρησιμοποιούμενων κλάσεων και των μεθόδων τους, μπορούμε να οργανώσουμε την περιγραφή των παραπάνω σε δύο βασικές κατηγορίες:

- Θεμελιώδεις/Κοινές κλάσεις για οποιοδήποτε τεστ
- Συγκεκριμένες κλάσεις για κάθε τεστ ξεχωριστά

Θεμελιώδεις Κλάσεις

Θα ξεκινήσουμε αρχικά με την περιγραφή των βασικών-θεμελιωδών κλάσεων και θα προχωρήσουμε έπειτα στην επεξήγηση κάθε τεστ ξεχωριστά ακολουθούμενα με τη σχεδιαστική αναπαράσταση των κλάσεων και των μεθόδων τους.

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”



Εικόνα 31 General Classes

και του υπολογιστή (benchmark data outputs) , πρέπει υποχρεωτικά να αναπτυχθεί μία ασφαλής επικοινωνία μεταξύ τους, είτε υπάρχουν ήδη keypairs είτε πρέπει να δημιουργηθούν καινούργια. Αυτή η πληροφορία δίνεται πάλι από το χρήστη μέσω της

Στην διπλανή εικόνα παρουσιάζουμε τις τρεις βασικές κλάσεις που συμμετέχουν για τη δημιουργία (spawning) ενός VM. Αρχικά η κλάση provider περιέχει όλες τις πληροφορίες για την αυτή τη διαδικασία. Συγκεκριμένα δέχεται τον τύπο του provider, credentials του χρήστη για το συγκεκριμένο provider , καθώς και τι τύπος θα είναι το VM που θα γίνει spawn . Δεδομένου ότι στην υλοποίηση μας έχουμε δημιουργήσει Openstack private Cloud , η κλάση Provider καλεί την Openstack κλάση στην οποία περνάνε όλα τα απαραίτητα δεδομένα για το spawning του μηχανήματος και μέσω της μεθόδου init() της κλάσης Openstack , το εικονικό μηχάνημα δημιουργείται.

Παράλληλα , επειδή θα υπάρχει ανταλλαγή αρχείων μεταξύ του VM

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

φόρμας που έχει συμπληρώσει αρχικά. Η κλάση PubPrivateKeyUtils είναι υπεύθυνη για τη διαδικασία αυτή και με την ολοκλήρωσή της, έχει γίνει ανταλλαγή των δημοσίων κλειδιών και των δύο μηχανημάτων.

Στην εφαρμογή που έχουμε αναπτύξει ο χρήστης μπορεί να εκτελέσει τρία (3) διαφορετικά τεστ, όπως έχει ήδη αναφερθεί. Κρίνεται σκόπιμο στο σημείο αυτό να αναλύσουμε τα είδη των τεστ.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Μία γενική αναφορά στον τρόπο λειτουργίας των κλάσεων

Όταν μία σελίδα καλείται, αν έχει γίνει map, τότε αμέσως καλείται ο Model (Model) που αντιστοιχεί στην mapped σελίδα και αρχικοποιούνται οι μεταβλητές που χρησιμοποιούνται. Εδώ αξίζει να τονίσουμε ότι κατά την έκταση προγραμματισμού του framework χρησιμοποιήθηκαν οι μέθοδοι Get και Set, βασικές μέθοδοι για να είναι δυνατή η μεταφορά των τιμών των μεταβλητών μεταξύ σελίδων. Όταν ο Controller (Controller) έχει τρέξει τον κώδικα που του αναλογεί για τη συγκεκριμένη Mapped σελίδα, τότε προωθεί στην επόμενη αν η συμπλήρωση είναι ορθή και επιτυχής, ή θα γίνει reload της σελίδας σε πιθανή αποτυχία ορθής συμπλήρωσης της φόρμας. Στην εφαρμογή μας γίνονται πραγματοποιούνται δύο είδους έλεγχοι. Ο πρώτος είναι κατά τη διάρκεια συμπλήρωσης της φόρμας μέσω τεχνολογίας Javascript όπου εκεί γίνονται έλεγχοι για σωστή συμπλήρωση πεδίου (αν απαιτείται αριθμητική τιμή) και ο δεύτερος είναι όταν η φόρμα έχει συμπληρωθεί μεν αλλά μερικώς (πχ δεν έχει συμπληρωθεί το πεδίο threads). Αν φυσικά η συμπλήρωση είναι σωστή τότε εμφανίζεται (View) η αντίστοιχη σελίδα με τα αποτελέσματα, ενώ χωρίς ο χρήστης να γνωρίζει έχει ταυτόχρονα πραγματοποιηθεί insert sql query για να αποθηκευθούν τα αποτελέσματα στη βάση

DB Test

Για το κομμάτι των Βάσεων Δεδομένων (DB test) χρησιμοποιούμε το framework YCSB που είναι δημιουργημένο για εκτενή cloud serving stores tests[15] στις παρακάτω DB

- PNUTS
- BigTable
- HBase
- Hypertable

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

- Azure
- Cassandra
- CouchDB
- Voldemort
- MongoDB
- Infinispan
- Dymomite
- Redis
- GemFire
- GigaSpaces XAP
- DynamoDB

Στο framework που έχουμε αναπτύξει χρησιμοποιούμε τις MongoDB, MySQL , Postgre και τέλος Cassandra Βάσεις Δεδομένων. Στην εικόνα που ακολουθεί παρουσιάζουμε τις κλάσεις που συμμετέχουν σε αυτό το τεστ. Θα εξηγήσουμε αναλυτικά τον τρόπο με τον οποίο συνδέονται .Ο χρήστης επιλέγει να εκτελέσει ένα τεστ βάσεων δεδομένων. Με αυτή την ενέργεια φορτώνεται η σελίδα db.jsp η οποία έχει γίνει mapped βάσει συγκεκριμένου Controller (DbController.java). Η λειτουργία του Controller στην προκειμένη περίπτωση είναι η εξής. Κάθε φορά που ο server (Apache TomCat 7.0) δέχεται requests για τη συγκεκριμένη σελίδα db.jsp , αρχικοποιούνται οι μεταβλητές της κλάσης DB. Αυτό γίνεται γιατί στη φόρμα που συμπληρώνει ο χρήστης , ο κώδικας περιέχει την παρακάτω γραμμή

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

db.jsp

```
form:form commandName="dbentity" id="form2" enctype="multipart/form-data"  
name="myForm" cssClass="form-horizontal">
```

Η περιγραμμένη περιοχή αποδίδει ένα command name στη φόρμα με όνομα dbentity. Ο Controller λαμβάνει το όνομα της φόρμας , το συγκρίνει με το ModelAttribute και στο επόμενο βήμα δημιουργεί ένα Object dbentity της κλάσης DB. Αυτό φαίνεται στην παρακάτω γραμμή κώδικα στην περιγραμμένη περιοχή.

DbController.java

```
public String db(@ModelAttribute("dbentity") DB dbentity,  
BindingResult result, HttpServletRequest request,  
HttpServletRequest response) throws IOException {
```

Βασικός μας στόχος είναι , ότι κάθε τεστ που εκτελείται, και πιο συγκεκριμένα οι παράμετροι που έχει καταχωρίσει ο χρήστης να αποθηκεύονται σε μία ΒΔ , σε περίπτωση που χρειαστεί το τεστ να επαναληφθεί ή να γίνει πιο αναλυτική μελέτη των output σε σχέση με τα input.

Με την δημιουργία του αντικειμένου dbentity , οι μεταβλητές

```
database threads target operationcounts recordcount date time centenary  
freq vmtype username password keypairname imagename ram minDisk  
provider  
url novakeys novakeys providersum
```

που ουσιαστικά είναι οι μεταβλητές της φόρμας που έχει συμπληρώσει ο χρήστης , μέσω των Get and Set μεθόδων , τους αποδίδονται ως τιμές οι τιμές της φόρμας και πλέον

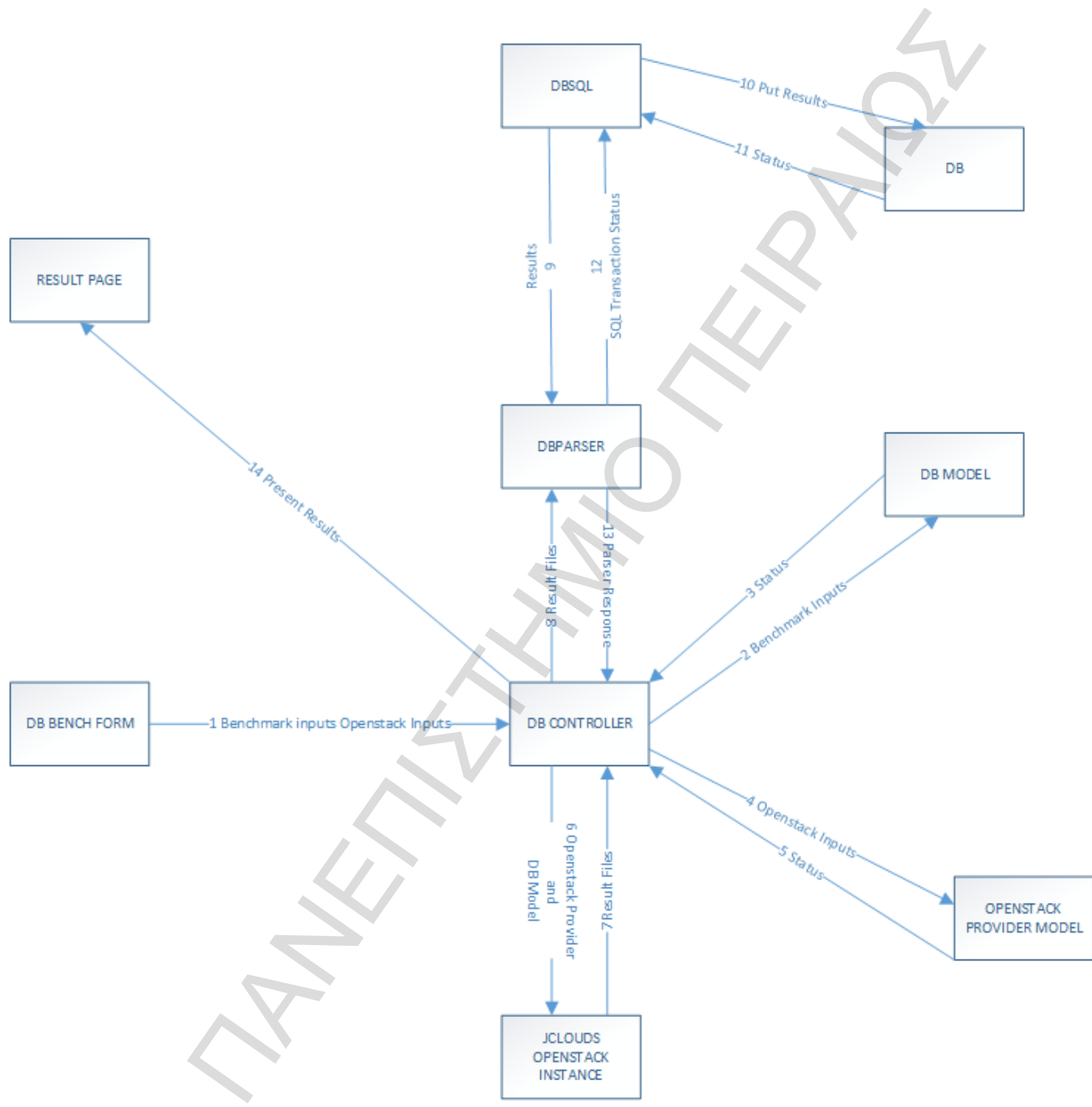
“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

μπορούν να χρησιμοποιηθούν περαιτέρω. Εν συνεχεία , επειδή πρέπει να αποδοθούν και γραφικά τα αποτελέσματα του Benchmark στο χρήστη , εκτελείται ένα select query που ανακτά τα αποτελέσματα που έχουν αποθηκευθεί στη ΒΔ και μέσω της εντολής

```
model.addAttribute("****", ****);
```

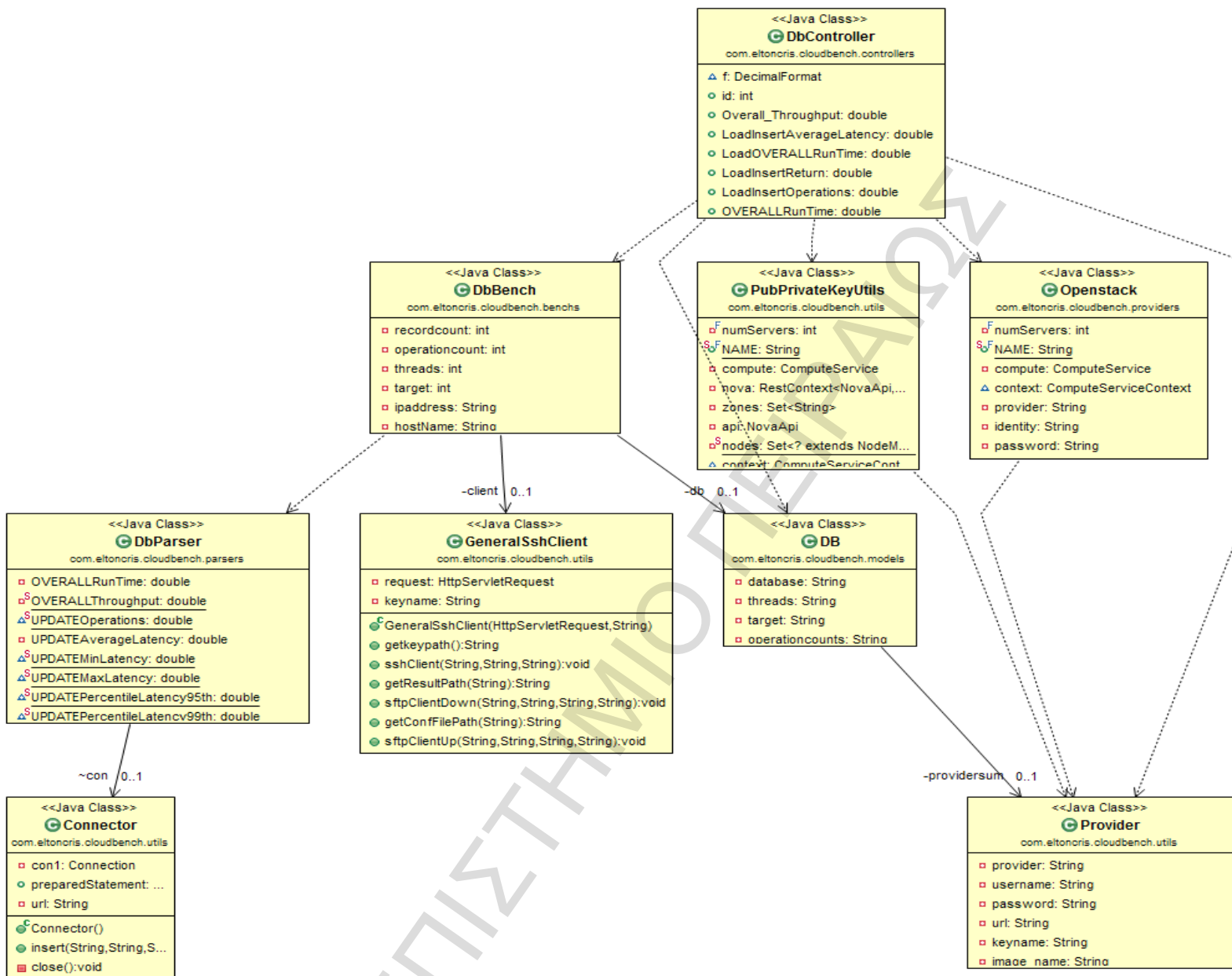
αποδίδεται ένα attribute στην ανακτούμενη πληροφορία προκειμένου να αναπαρασταθεί στην σελίδα success.jsp, όπου και παρουσιάζονται τα αποτελέσματα με τη μορφή JavaScript Charts ή απλά αποδίδεται η τιμή τους αν δεν είναι απαραίτητη κάποια γραφική αναπαράσταση.

Ακολουθεί το class diagram και οι συσχετίσεις των κλάσεων καθώς επίσης και το διάγραμμα ροής εκτέλεσης του τεστ.



Εικόνα 32 Db Controller

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”



Εικόνα 33 DB Classes Web Serving Test

Web Serving

Στη διαδικασία εκτέλεσης Web Serving test έχουμε επιλέξει το weighttp. Είναι ένα ελαφρύ και μικρό εργαλείο συγκριτικής αξιολόγησης για web servers. Έχει σχεδιαστεί για να είναι πολύ γρήγορο και εύκολο στη χρήση και υποστηρίζει ένα μικρό κομμάτι του HTTP πρωτοκόλλου ώστε να είναι απλό. Το συγκεκριμένο tool υποστηρίζει multithreading ώστε να χρησιμοποιεί σε την τεχνολογία επεξεργαστών πολλαπλών πυρήνων, καθώς και ασύγχρονα i/o για ταυτόχρονα requests σε ένα thread.[16].

Η λογική όπως και στο προηγούμενο test είναι ακριβώς η ίδια.

```
Web_serving.jsp

<form:form commandName="webserving" id="form2" enctype="multipart/form-
data" name="myForm" cssClass="form-horizontal">
    Web Serving Controller

@RequestMapping(value = "/web_serving")
public String web_serving(@ModelAttribute("webserving"))
```

Όταν ο χρήστης επιλέξει να εκτελέσει web serving test, ο αντίστοιχος controller (WebServingController) ενεργοποιείται και δημιουργείται ένα αντικείμενο του αντίστοιχου model (WebServing). Όταν ο χρήστης επιλέξει να ξεκινήσει το test (Finish Button), πραγματοποιείται ένας έλεγχος ώστε αν η φόρμα έχει συμπληρωθεί σωστά να οδηγηθεί ο χρήστης στη σελίδα των αποτελεσμάτων, αλλιώς να επιστρέψει πάλι στο αρχικό μενού. Αυτό γίνεται με τον έλεγχο της τιμής των threads για το test. Συγκεκριμένα αν η τιμή τους είναι null τότε ο controller οδηγεί το χρήστη να συμπληρώσει ξανά τη φόρμα. Σε διαφορετική περίπτωση κάνει προώθηση στη σελίδα success2. Αυτή η σελίδα δεν υπάρχει, απλά είναι μία mapped σελίδα προκειμένου εκείνη τη στιγμή να κληθούν οι κλάσεις για πραγματοποιηθούν συνδέσεις με τη ΒΔ για να ανακτηθούν τα αντίστοιχα outputs των αποτελεσμάτων και να οπτικοποιηθούν προς το χρήστη με τις μεθόδους:

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

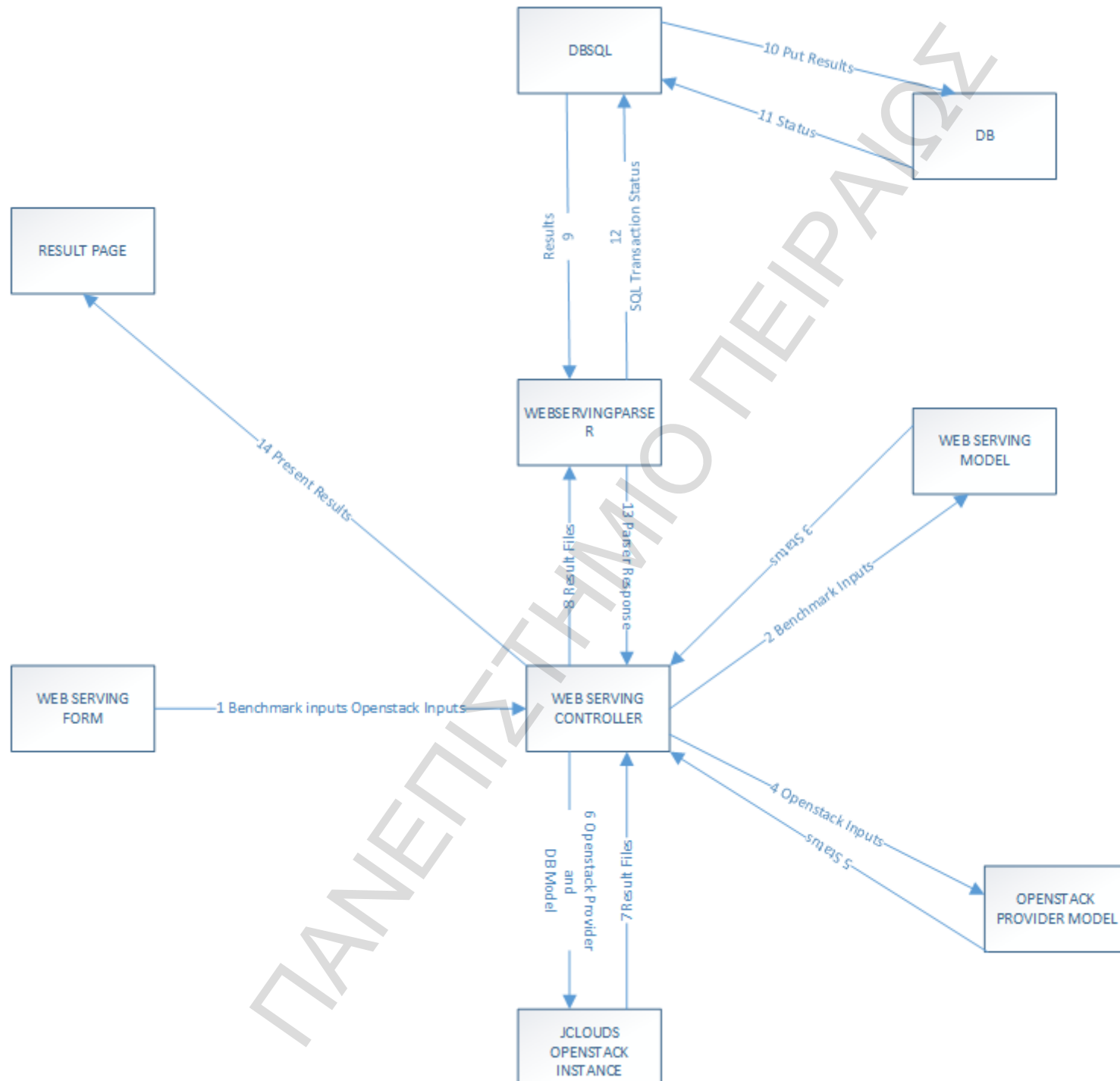
```
model.addAttribute("total_sec",total_sec );
model.addAttribute("success",success*100 );
model.addAttribute("seconds", seconds);
model.addAttribute("millisec", millisec);
model.addAttribute("microsec", microsec);
model.addAttribute("kbpersec", kbpersec);
model.addAttribute("totalreq", totalreq);
model.addAttribute("startedreq", startedreq);
model.addAttribute("donereq", donereq);
model.addAttribute("succeededreq", succeededreq);
model.addAttribute("failedreq", failedreq);
model.addAttribute("erroredreq", erroredreq);
model.addAttribute("traffichttp", traffichttp);
model.addAttribute("traffictotal", traffictotal);
model.addAttribute("trafficdata", trafficdata);
model.addAttribute("conc_req", conc_req);
```

Παρακάτω ακολουθεί ένα τμήμα του αρχείου success.jsp το οποίο είναι και υπεύθυνο να απεικονίσει τα αποτελέσματα στην οθόνη.

```
<div class="span3">
  <div class="chart" data-percent="${total_sec}">
    <font size="1"> ${seconds} sec ${millisec} millisec
      ${microsec} microsec </font>
  </div>
  <div class="chart-bottom-heading">
    <span class="label label-info">Overall Run Time</span>
  </div>
</div>
<div class="span3">
  <div class="chart" data-percent="${kbpersec}">${kbpersec}</div>
  <div class="chart-bottom-heading">
    <span class="label label-info">Throughput</span>
  </div>
</div>
```

Η πιο πάνω επιλεγμένη μεταβλητή, με τη μέθοδο `addAttribute`, συσχετίζει ένα String με όνομα `kbpersec` με την τιμή `kbpersec`, που έχει ανακληθεί στο προηγούμενο query. Έτσι όταν θα φορτώσει η εν λόγω σελίδα, θα περάσει η τιμή της μεταβλητής `kbpersec` του Controller στην σελίδα. Αυτή είναι και συνοπτικά και η βασική ροή του μοντέλου MVC.

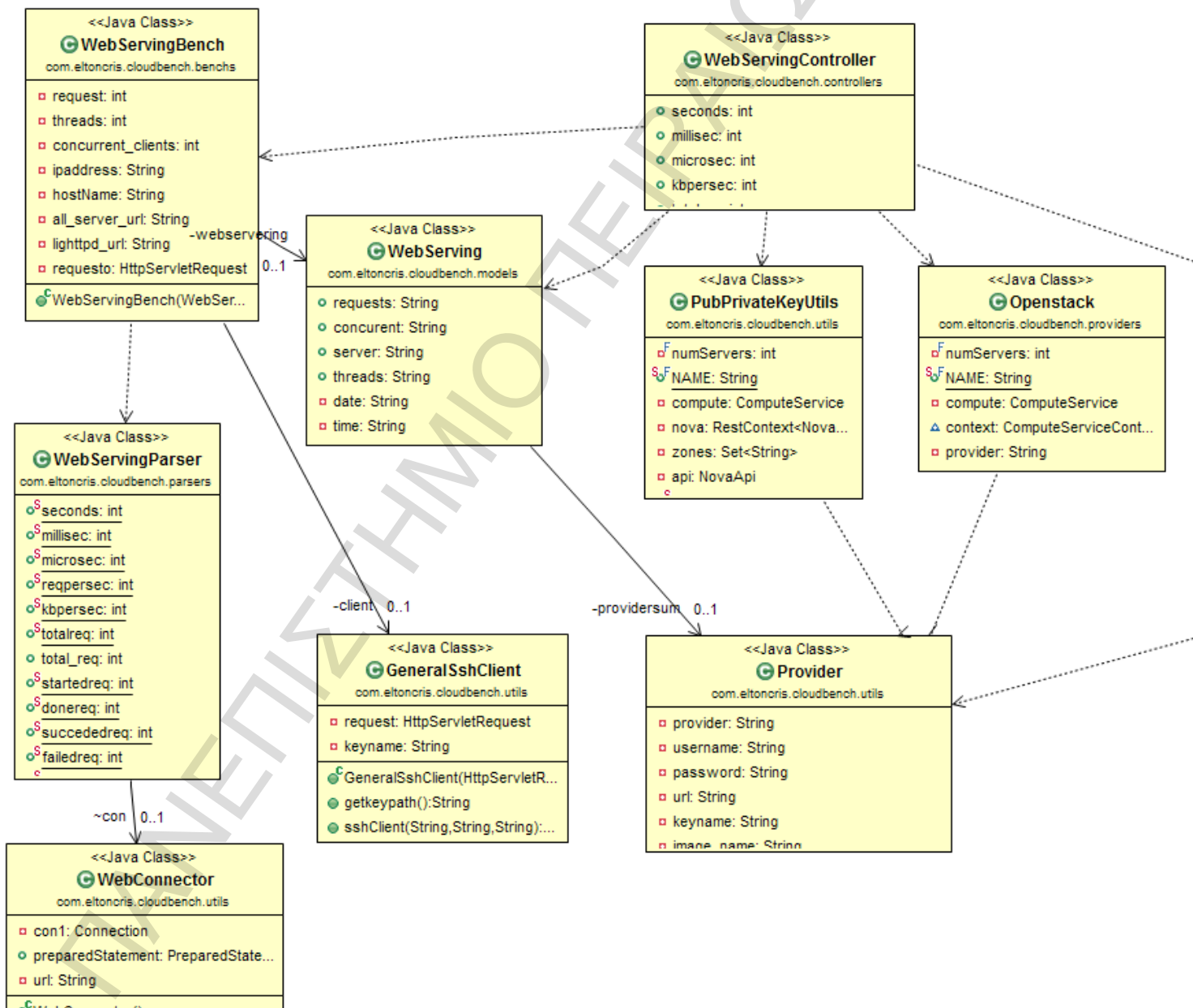
Ακολουθεί το σχεδιάγραμμα σχέσεων κλάσεων καθώς επίσης και το διάγραμμα ροής εκτέλεσης του τεστ.



Εικόνα 34 Web Serving Controller

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Εικόνα 35 Web Serving Classes



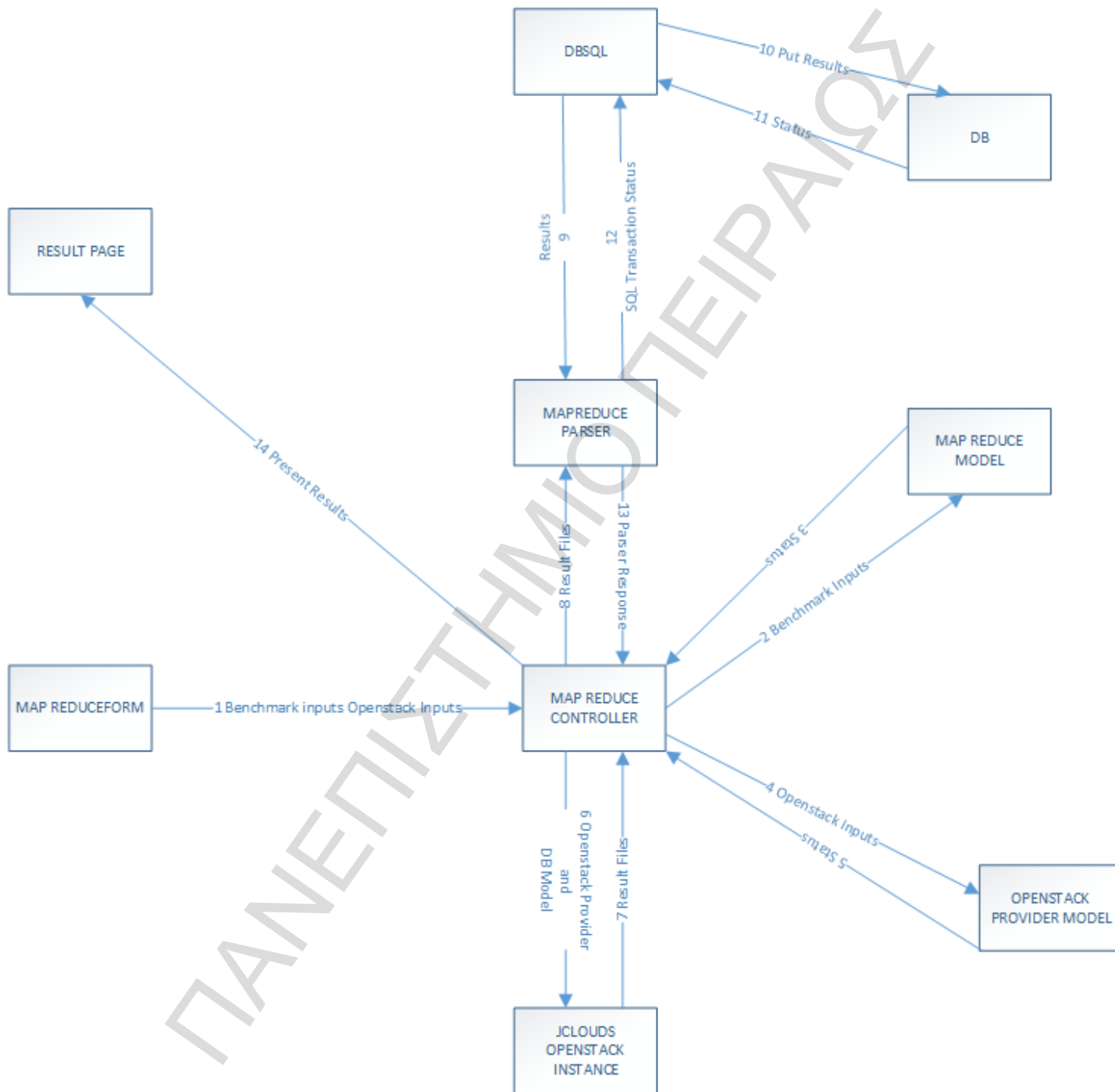
Map-Reduce / Hadoop Test

Η τελευταία επιλογή του χρήστη όσον αφορά τα τεστ είναι η επιλογή Map Reduce. Στη διαδικασία αυτή επιλέξαμε να χρησιμοποιήσουμε τα παρακάτω τεστ:

1. NNBench
2. MRBench
3. Teragen
4. Dfsio

Όπως και στα δύο προηγούμενα test , ο τρόπος υλοποίησης δεν αλλάζει, παρά μόνο ότι τα εξαγόμενα αποτελέσματα στα τεστ dfsio , nnbench και teragen παράγουν 2,3 και 3 αρχεία το καθένα ως output. Ο controller, στο τεστ αυτό έχει γραφεί ώστε να δέχεται ως όνομα ιστοσελίδας το : map_reduce , για το οποίο έχει γίνει και map. Όμως επειδή ο χρήστης έχει το δικαίωμα να επιλέξει μεταξύ τεσσάρων τεστ , γίνεται αντίστοιχη προώθηση σε τοπικές σελίδες (mapped) για την επεξεργασία του αιτήματος , και την ανάκτηση των Output από τη βάση. Έτσι αν ο χρήστης έχει επιλέξει το dfsio τεστ , προωθείται στην success. Έπειτα εκτελείται το query , ανακτώνται τα δεδομένα , γίνονται attributes και οδηγούμαστε στη σελίδα success_dfsio , όπου και οπτικοποιούνται τα αποτελέσματα.

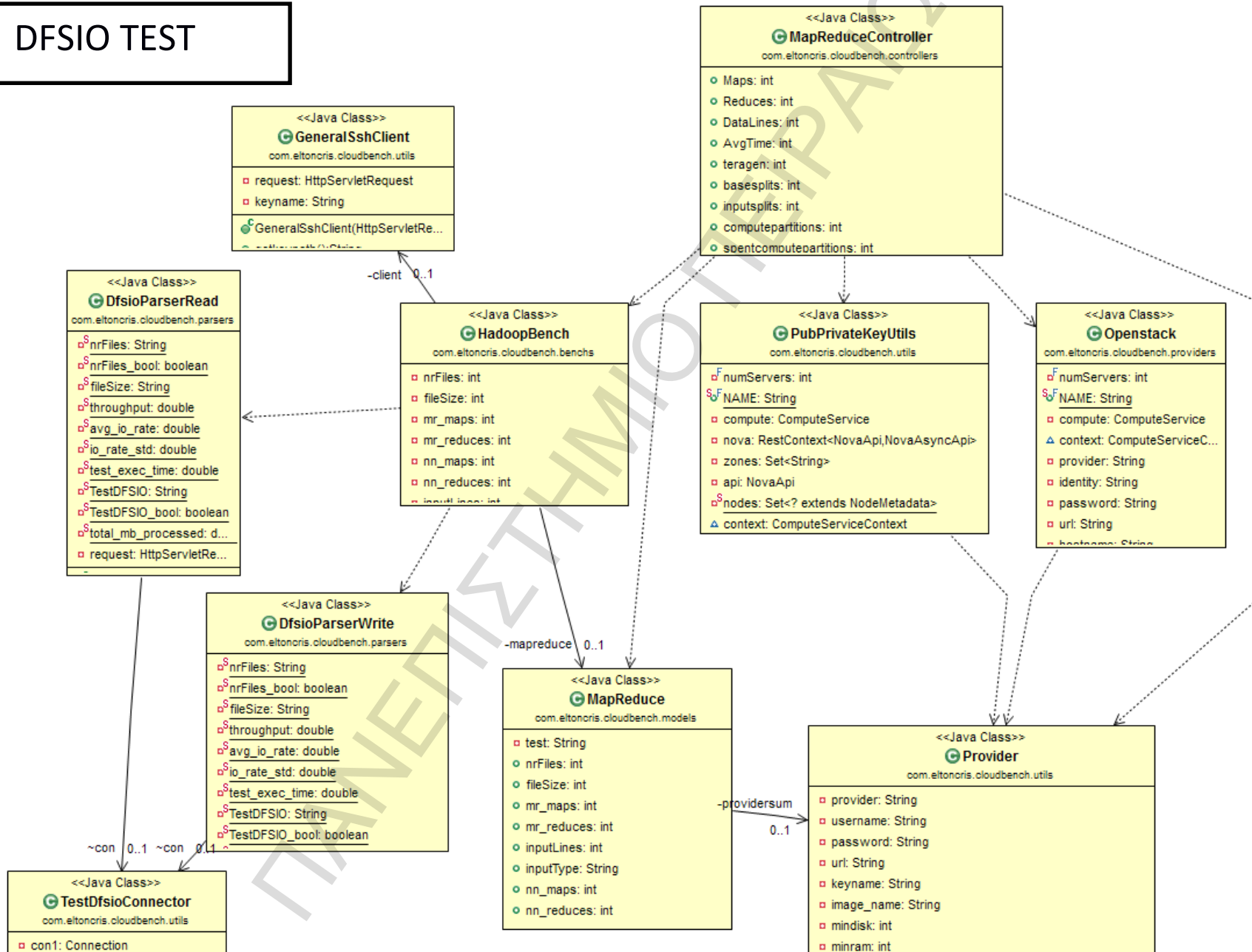
Ακολουθούν τα class diagrams για κάθε ένα από τα τέσσερα (4) τεστ ξεχωριστά καθώς επίσης και το διάγραμμα ροής εκτέλεσης του τεστ.



Εικόνα 36 Map Reduce Controller

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

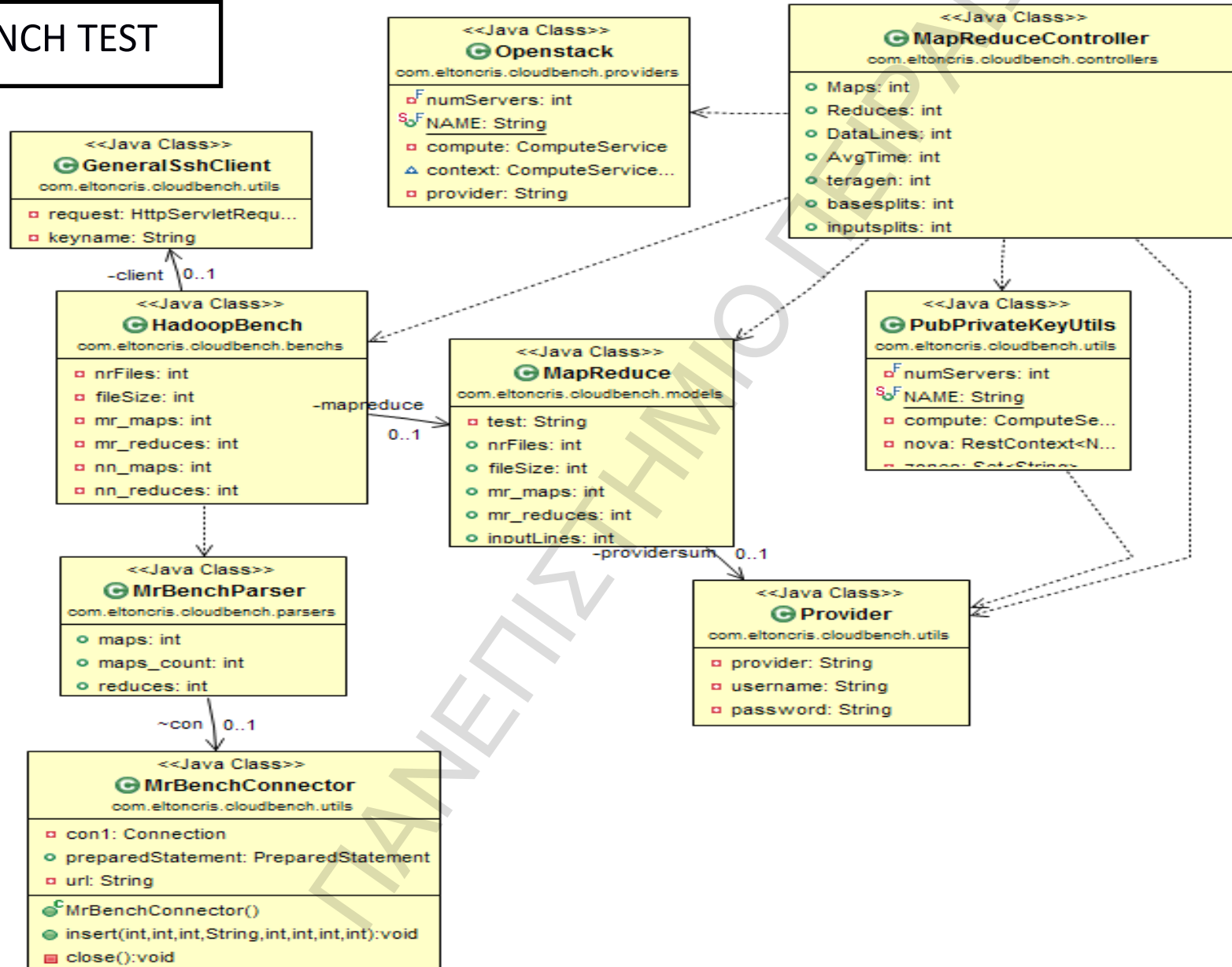
DFSIO TEST



Εικόνα 37 Dfsio Map Reduce Classes

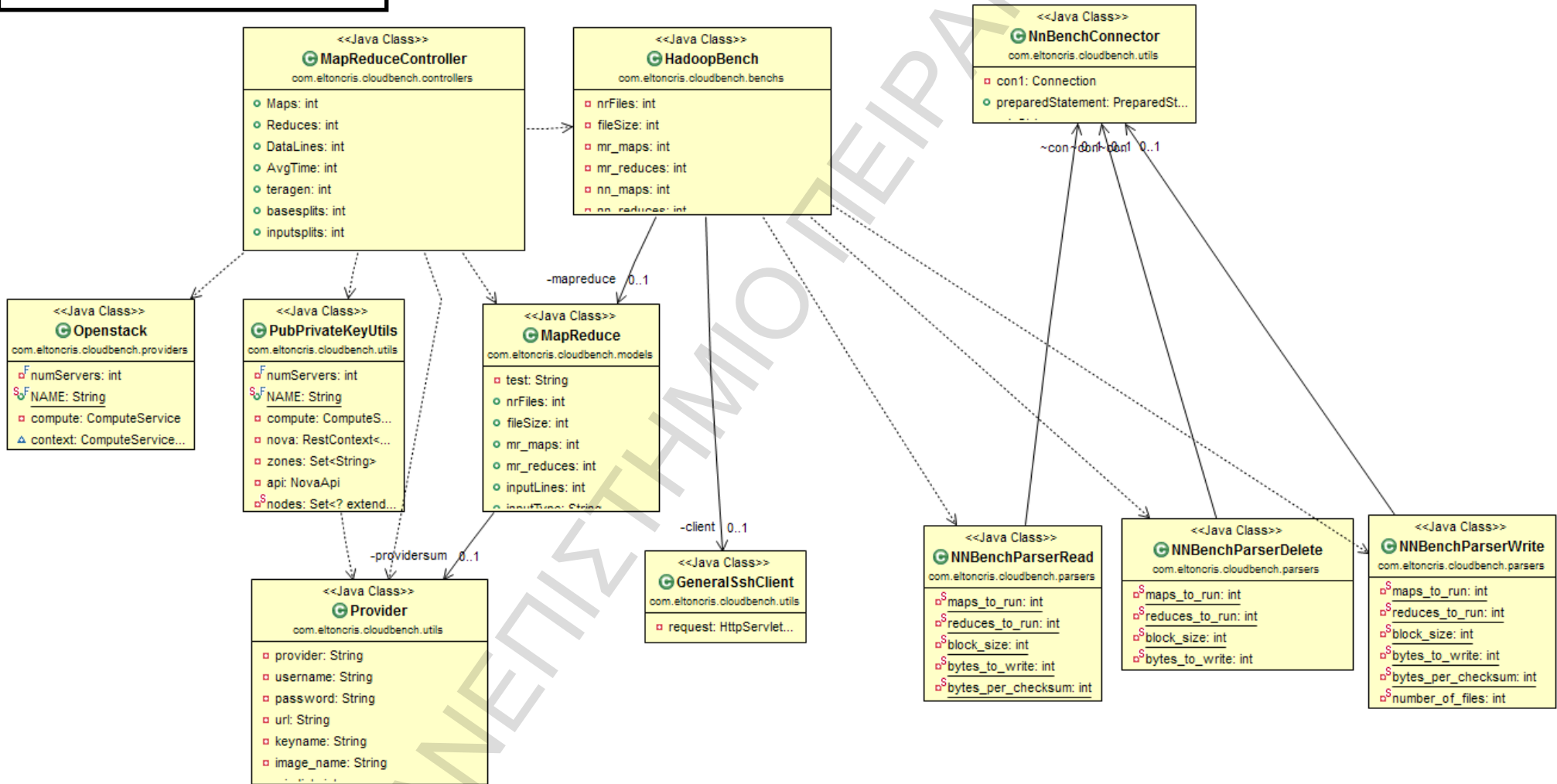
“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

MRBENCH TEST



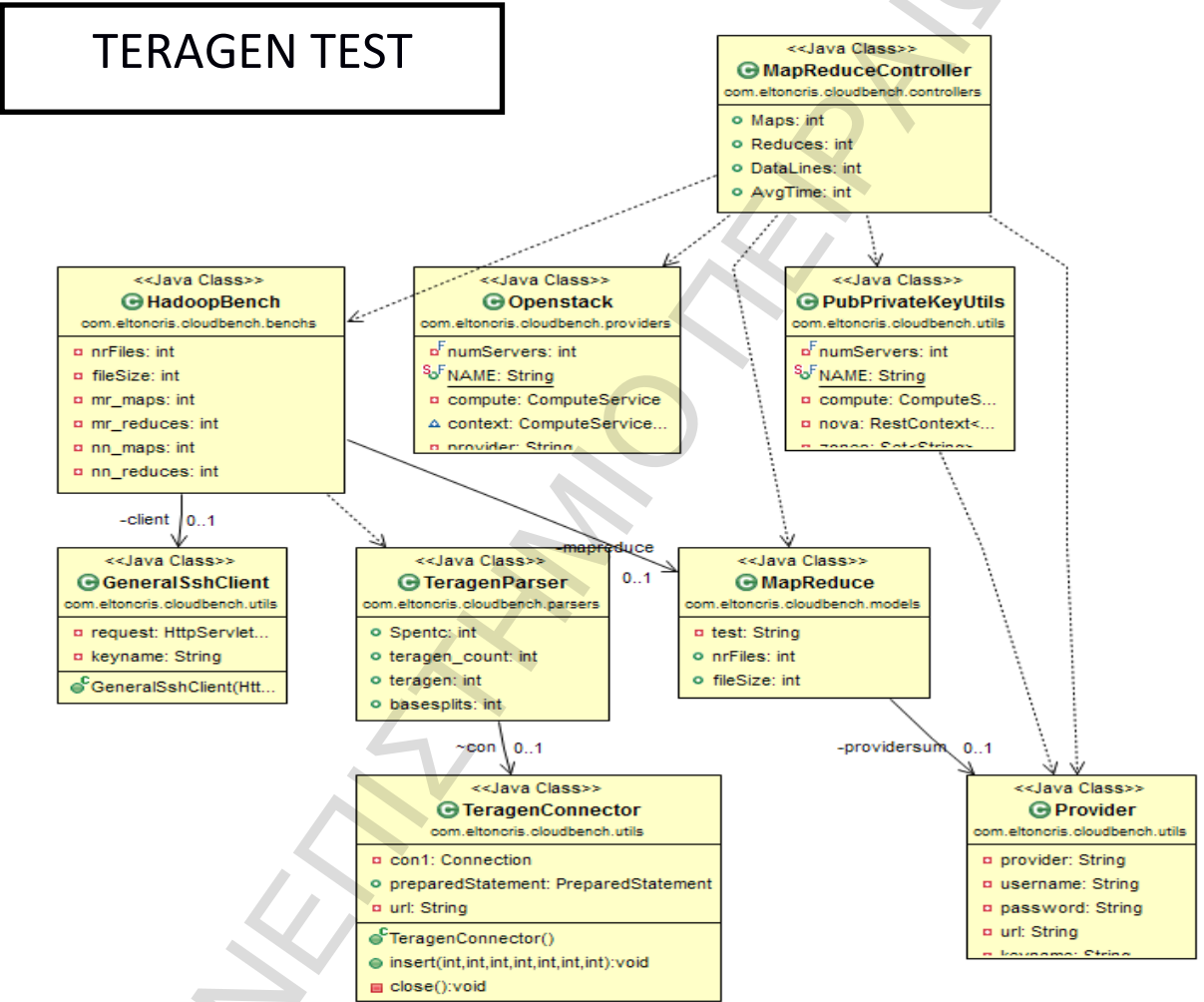
“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

NNBENCH TEST



Εικόνα 39 NNbenchmark Classes

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”



Εικόνα 40 Teragen Controller Classes

Reports

Το Framework δίνει τη δυνατότητα στο χρήστη όπως έχουμε αναφέρει , να γραφικοποιήσει τα αποτελέσματα παρελθόντων τεστ όπως επίσης και να δει στην οθόνη του αν και κατά πόσο η επιλογή του συγκεκριμένου vm για το συγκεκριμένο workload ήταν η σωστή (σε ποιο ποσοστό). Παρακάτω αναλύουμε τον Controller που είναι υπεύθυνος για τα παραπάνω.

Όταν επιλεγεί η σελίδα reports.jsp ο Controller ReportController , κάνει forward στη σελίδα results ,όταν έχει συμπληρωθεί επαρκώς η αντίστοιχη φόρμα. Τότε , πραγματοποιείται σύνδεση με τη ΒΔ για να ανακτηθούν τα δεδομένα που ζητά ο χρήστης , για να αναπαρασταθούν γραφικά , με τη μέθοδο connect. Λόγω του ότι κάποια τεστ που ανακτώνται , μπορεί να προκαλέσουν τη δημιουργία πολλαπλών γραφημάτων ανά τεστ, η κλάση PieChart , αναλαμβάνει μέσω των constructors να αναπαραστήσει μονά , διπλά , η τριπλά επαναλαμβανόμενα γραφήματα ανά τεστ. Ταυτόχρονα πρέπει να οπτικοποιηθεί η αποτελεσματικότητα του workload/vm βάσει της συνάρτησης απόδοσης SE που έχουμε αναφέρει σε προηγούμενο κεφάλαιο (3.2.2.1).

Όπως έχουμε αναφέρει η λειτουργία του Controller είναι από τις πιο σημαντικές στο μοντέλο MVC. Για τη διαδικασία δημιουργίας ενός report συμμετέχουν οι εξής κλάσεις:

- import com.eltoncris.cloudbench.utils.VmType;
- import com.eltoncris.cloudbench.utils.Normalize;
- import com.eltoncris.cloudbench.utils.PieChart;

Η κλάση VmType χρησιμοποιείται για να μπορέσει αυτόματα να αποδοθεί το κόστος κάθε Vm Instance , βάσει της επτησγής του χρήστη στη φόρμα.

Η κλάση Normalize χρησιμοποιείται για να πραγματοποιήσει την κανονικοποίηση των τιμών για τα διάφορα βάρη της SE συνάρτησης.

Η κλάση PieChart χρησιμοποιείται για την παραγωγή του γραφήματος στην τελική οθόνη του χρήστη.

Η διαδικασία της εξαγωγής του γραφήματος έχει ιδιαίτερη σημασία. Όταν ο χρήστης συγκεκριμενοποιήσει τα δεδομένα του στη φόρμα και εκτελεστεί το query, τότε είναι δυνατόν να παραχθεί ένα ή περισσότερα αποτελέσματα

Έχουμε δημιουργήσει έναν πίνακα με τα αποτελέσματα όπως φαίνεται παρακάτω.

```
List<Double> chart_table = new ArrayList<Double>();
```

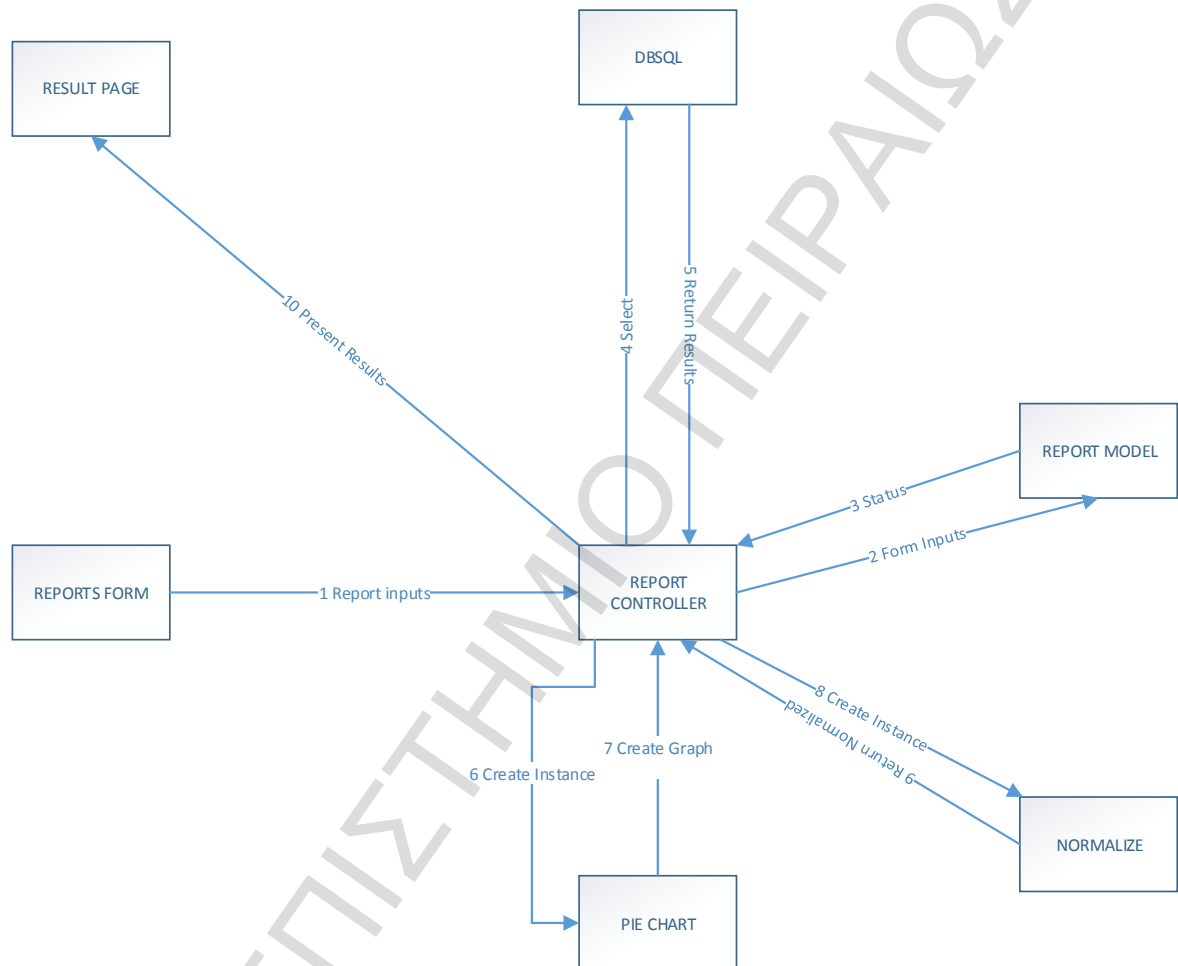
Η λίστα αυτή, είναι υπεύθυνη για να «κρατήσει» το ποσό των ιδίων αποτελεσμάτων ώστε να σχεδιαστούν τα αντίστοιχα γραφήματα.

Όταν για παράδειγμα θα σχεδιαστούν οι τα γραφήματα για ένα query σε ΒΔ θα εκτελεστεί η παρακάτω εντολή

```
PieChart demo = new PieChart("Database Results Chart", chart_table,  
chart_table.size(), "Database");
```

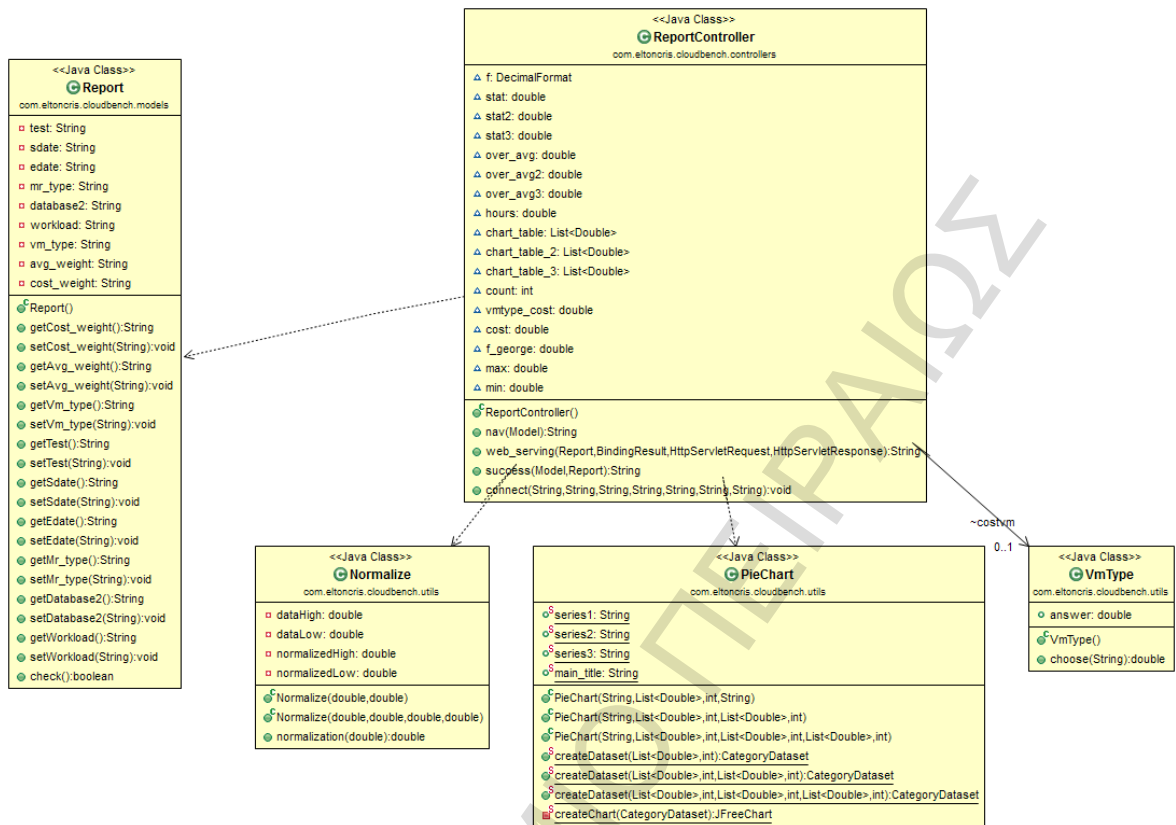
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Στην παρακάτω εικόνα παρουσιάζεται η βασική κλάση ReportsController και η συσχέτισή της με λοιπές κλάσεις καθώς και το διάγραμμα ροής της εκτέλεσης της διαδικασίας.



Εικόνα 41 Report Controller

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

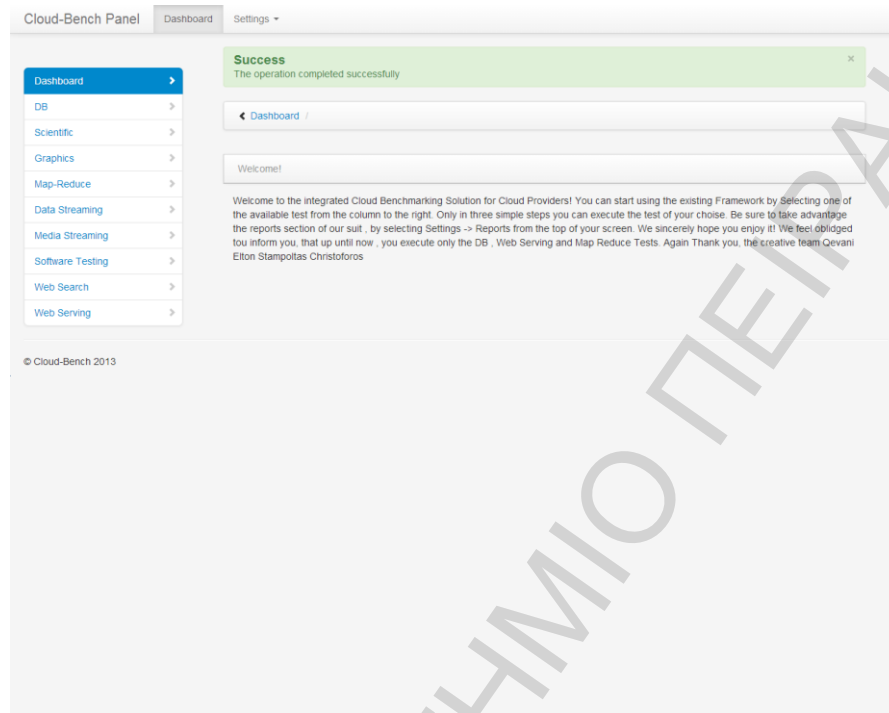


Εικόνα 42 Reports Classes

ΣΤΙΓΜΙΟΤΥΠΑ ΕΡΓΑΛΕΙΟΥ

Στο κεφάλαιο αυτό θα παρουσιάσουμε με τη βοήθεια στιγμιοτύπων , πώς ένας χρήστης μπορεί να αξιοποιήσει την εφαρμογή σε όλες τις επιλογές.

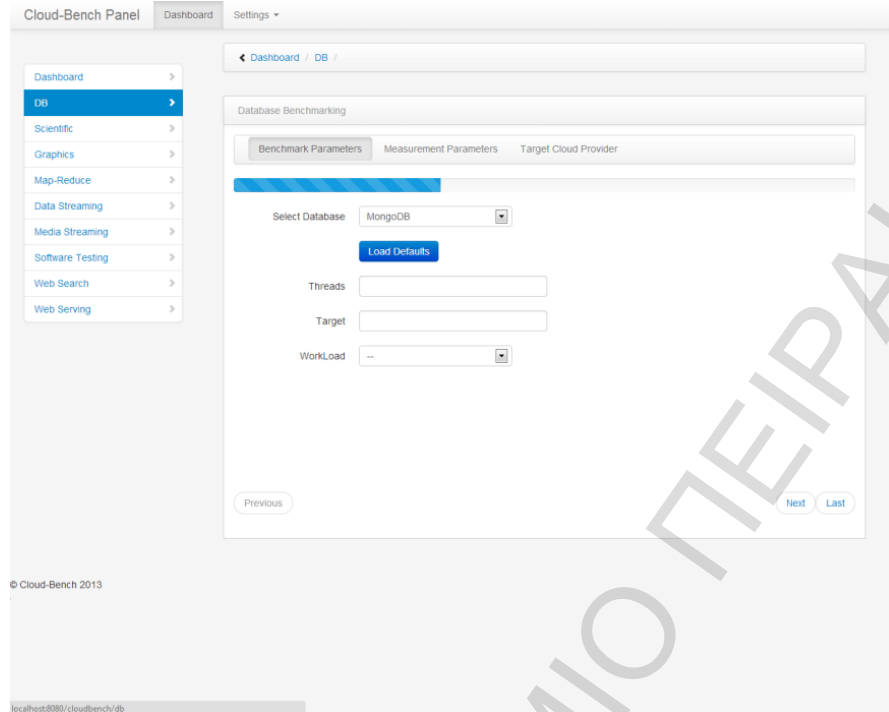
Κατά την εκκίνηση της εφαρμογής ο χρήστης βλέπει την πρώτη σελίδα (Welcome Page)



Εικόνα 43 Αρχικό Μενού

. Στην πρώτη επαφή με το εργαλείο ο χρήστης συναντά ένα περιβάλλον που ακολουθεί τα πρότυπα του twitter bootstrap. Εμφανίζεται ένα μήνυμα που καλωσορίζει το χρήστη και τον ενημερώνει για τις επιλογές που έχει καθώς για το ποια συγκριτικά τεστ έχει το δικαίωμα να επιλέξει. Ο χρήστης έχει τη δυνατότητα να επιλέξει ένα από τα διαθέσιμα τεστ ή να επιλέξει από το μενού στην κορυφή το dropdown Settings → Reports

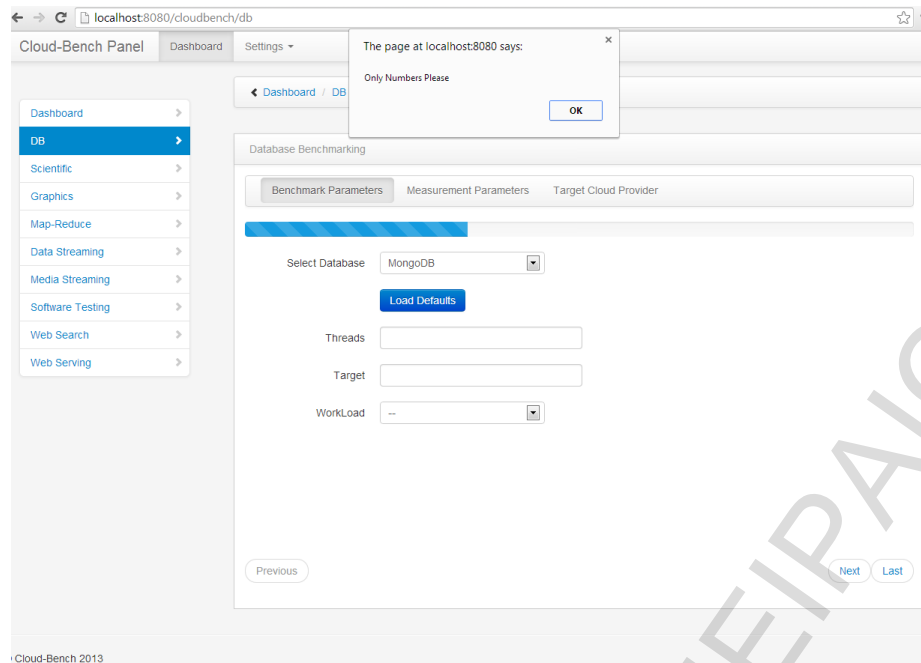
DB Bench



Εικόνα 44 Db Bench Step 1

Στην εικόνα αυτή βλέπουμε πως ο χρήστης έχει επιλέξει την εκτέλεση ενός DB Benchmark test. Πρέπει λοιπόν να συμπληρώσει όλα τα πεδία (όλα τα πεδία είναι απαραίτητα στην εφαρμογή). Συγκεκριμένα πρέπει να επιλέξει τα:

- Select Database. Επιλογή της ΒΔ από το πτυσσόμενο μενού.
- Threads. Ο αριθμός των νημάτων
- Target. Το επιθυμητό throughput.
- Workload. Το φορτίο του test. Αυτό μπορεί να το επιλέξει ο χρήστης από ένα πτυσσόμενο μενού.



Εικόνα 45 Wrong Input

Σε περίπτωση που ο χρήστης εισάγει λανθασμένες τιμές τότε ενημερώνεται για το σφάλμα αυτό. Έχουμε δώσει ιδιαίτερη βαρύτητα στο κομμάτι της αποφυγής σφαλμάτων και ο χρήστης δεν έχει το δικαίωμα να προχωρήσει στο επόμενο βήμα αν προηγουμένως δεν έχει προβεί στις απαραίτητες διορθώσεις. Εδώ αναλαμβάνει η Javascript που εκτελεί όλους τους ελέγχους που έχουμε θέσει.

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Cloud-Bench Panel Dashboard Settings

← Dashboard / DB /

Database Benchmarking

Benchmark Parameters Measurement Parameters Target Cloud Provider

Date

Time

Centenary

Frequency

Previous Next Last

© Cloud-Bench 2013

Εδώ ο χρήστης πρέπει να συμπληρώσει τα πεδία *Date*, για την ημερομηνία που επιθυμεί να γίνει το *test*, *time*, ποια συγκεκριμένη ώρα θα γίνει, *centenary*, η επαναληπτικότητα του τεστ (πχ αν 5

ώρες) και τέλος *Frequency*, η συχνότητα

Εικόνα 46 DB Step 2

Cloud-Bench Panel Dashboard Settings

You have gone full screen. Exit full screen (F11)

← Dashboard / DB /

Database Benchmarking

Benchmark Parameters Measurement Parameters Target Cloud Provider

Provider

Public URL

VM Type

Image Name

Minimum Required Ram

Minimum Required Disk Space

KeyPair

Previous Finish

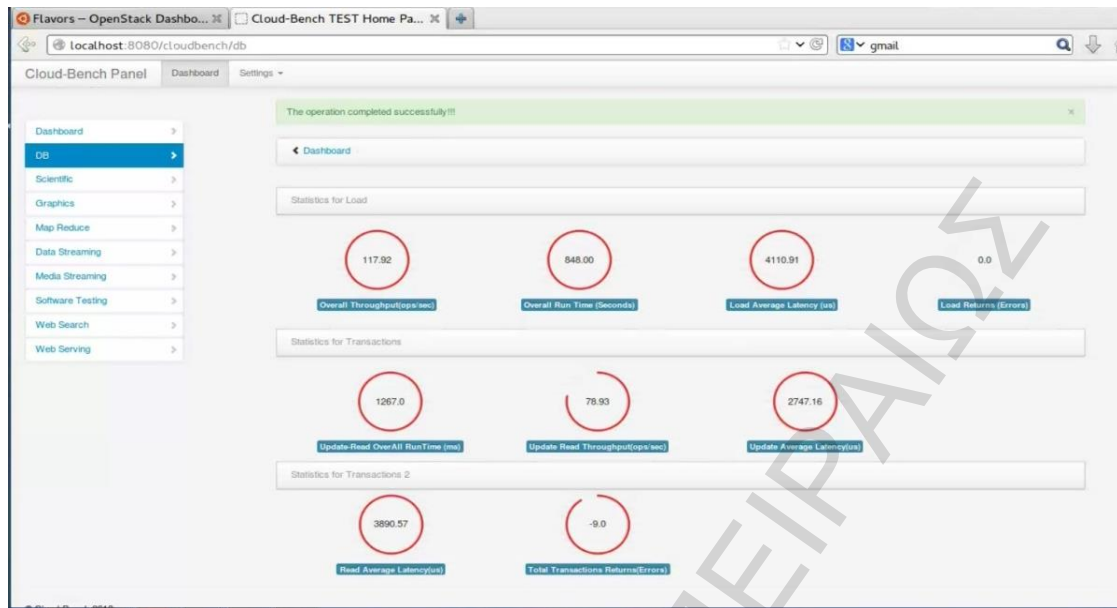
© Cloud-Bench 2013

επιλογής, δηλαδή πχ τρεις φορές στην πρώτη ώρα

Εδώ ο χρήστης συμπληρώνει τα απαραίτητα πεδία που αφορούν τον πάροχο. Στο τέλος εμφανίζεται η παρακάτω εικόνα που παρουσιάζει τα

αποτελέσματα του τεστ.

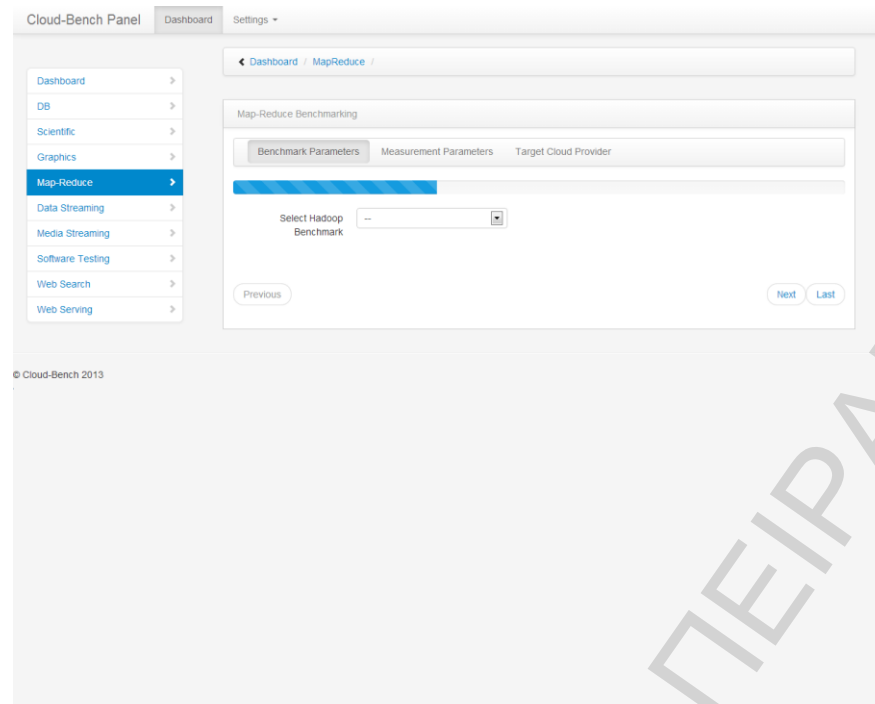
Εικόνα 47 DB Step 3



Εικόνα 48 DB Final

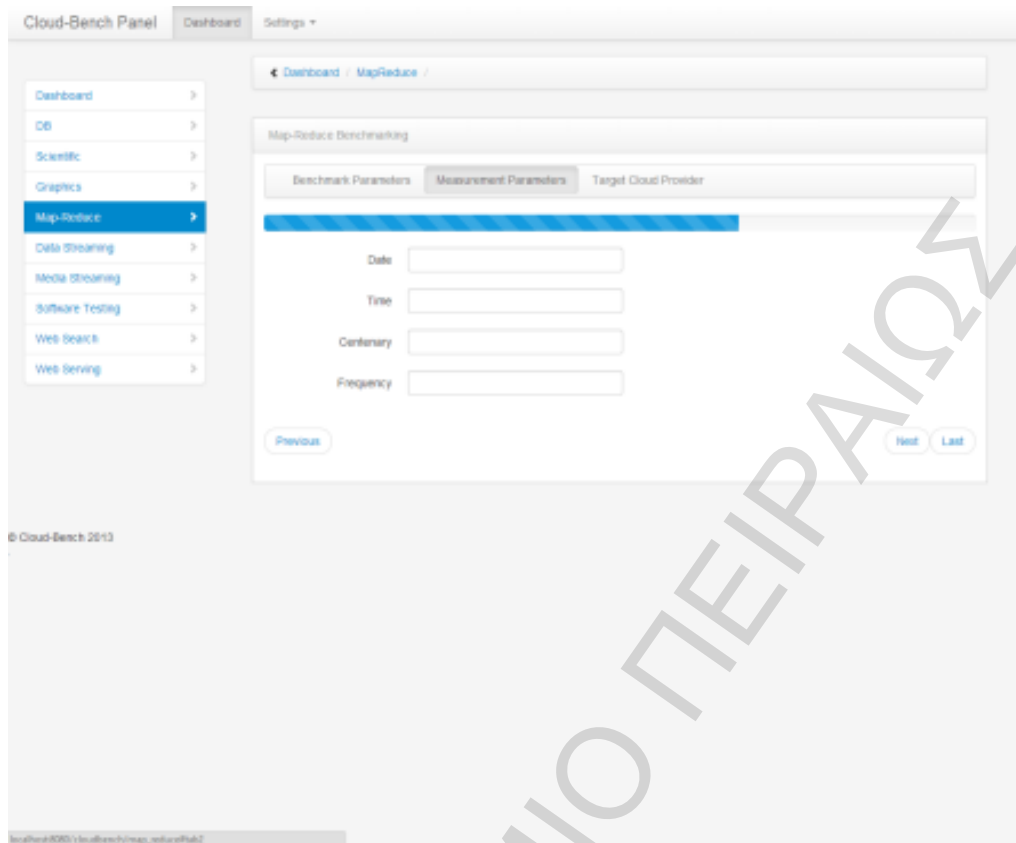
Κατά τη διάρκεια της υλοποίησης του εργαλείου, αποφασίσαμε βάσει των χαρακτηριστικών κάθε συγκριτικής αξιολόγησης να αποδώσουμε στην οθόνη, με το πέρασ κάθε δοκιμής, τα αποτελέσματα της. Σε κάθε σύγκριση το εξαγόμενο αποτέλεσμα ήταν σαφώς διαφορετικό και χρησιμοποιώντας τα ιδιαίτερα χαρακτηριστικά που προσφέρει το bootstrap, έχουμε το αποτέλεσμα της εικόνας 48. Σε κάποιες εικόνες εμφανίζεται γύρω από το αποτέλεσμα και ένας κοκκινός κύκλος. Αυτό γίνεται για να καταλάβει ο χρήστης της επί τοις εκατό (%) απόδοση του συγκεκριμένου δείκτη.

Map Reduce



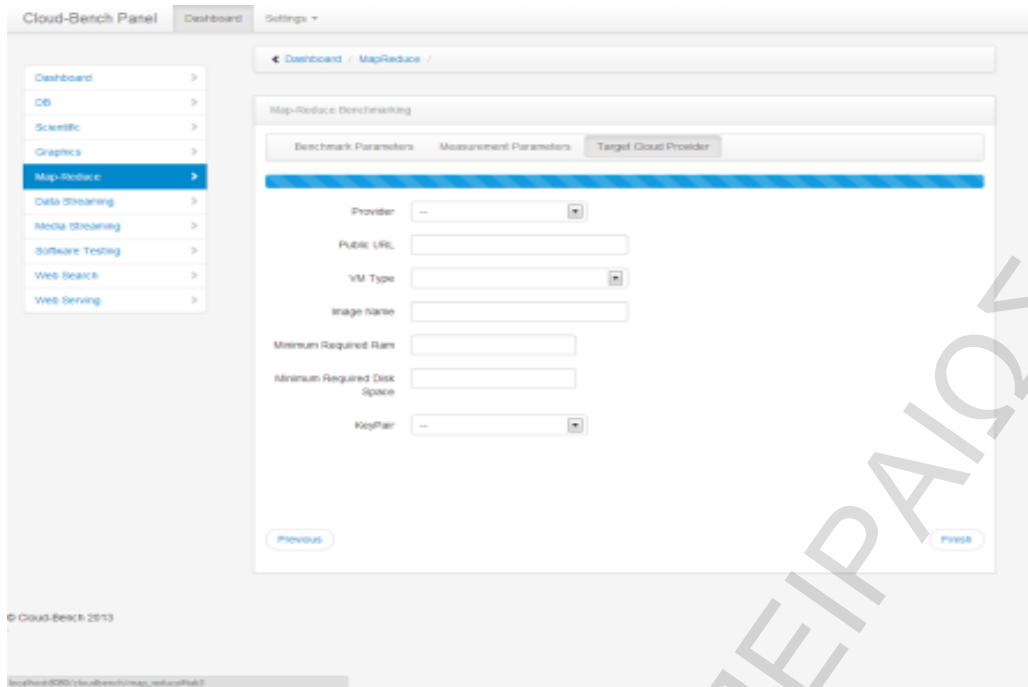
Εικόνα 49 Map Reduce Step 1

- Αρχική οθόνη. Εδώ ο χρήστης επιλέγει τον τύπου του benchmark που θα γίνει μεταξύ των TestDFSIO ,NNBench, Teragen, MrBench. Όπως βλέπουμε είναι από τα πιο διακριτικά και εύκολα μενού επιλογής λόγω των χαμηλών απαιτήσεων για ρυθμίσεις του συγκεκριμένου τεστ.



Εικόνα 50 Map Reduce Step 2

Ομοίως στο βήμα αυτό θα πρέπει να γίνουν τα βήματα που περιγράφηκαν στο αντίστοιχο βήμα για τις βάσεις δεδομένων. Καταχώριση ημερομηνίας ώρας και επαναληπτικότητας / συχνότητας



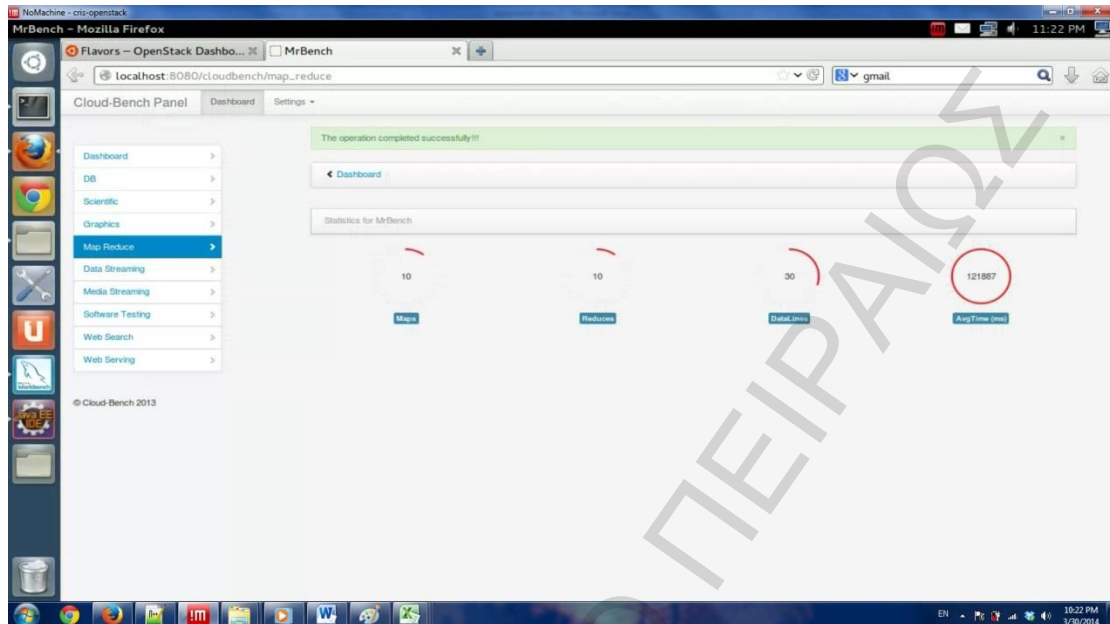
Εικόνα 51 Map Reduce Step3

Καταχώριση παραμέτρων για τον cloud provider. Εδώ για να υπάρξει επικοινωνία του μηχανήματος όπου εκτελείται το τεστ με το εικονικό μηχάνημα που θα δημιουργηθεί θα πρέπει ο χρήστης να συμπληρώσει ορθώς τα απαραίτητα πεδία.

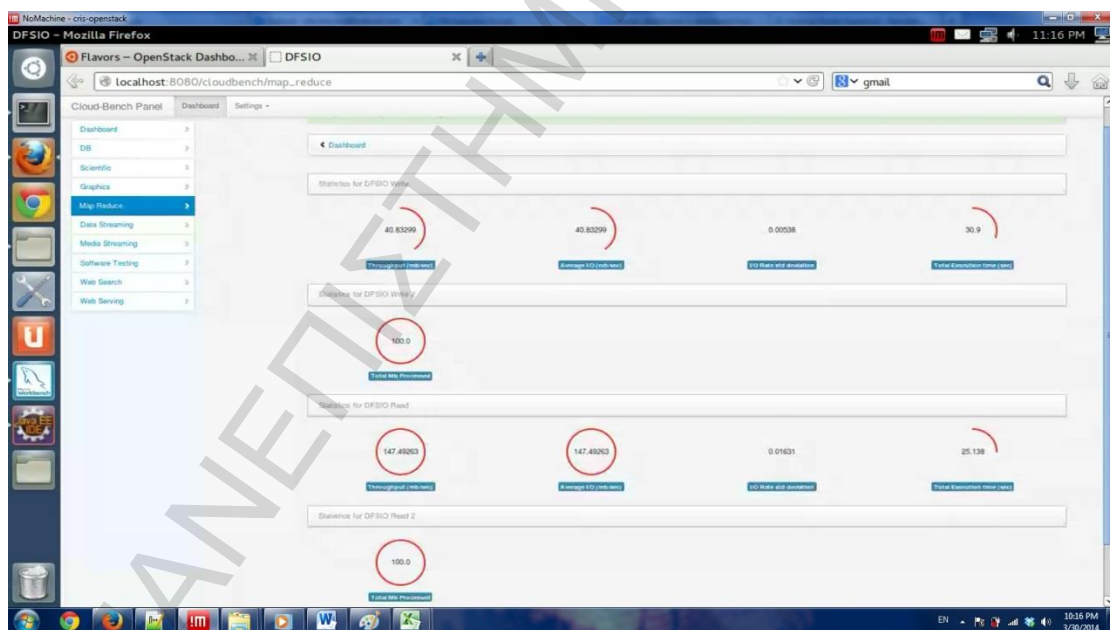
“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Λόγω του ότι τα Benchmark test που μπορούν να πραγματοποιηθούν είναι τέσσερα , οι επόμενες εικόνες παρουσιάζουν το γραφικό Output των τεστ.

MRBENCH



Εικόνα 52 Map Reduce Final MRBENCH

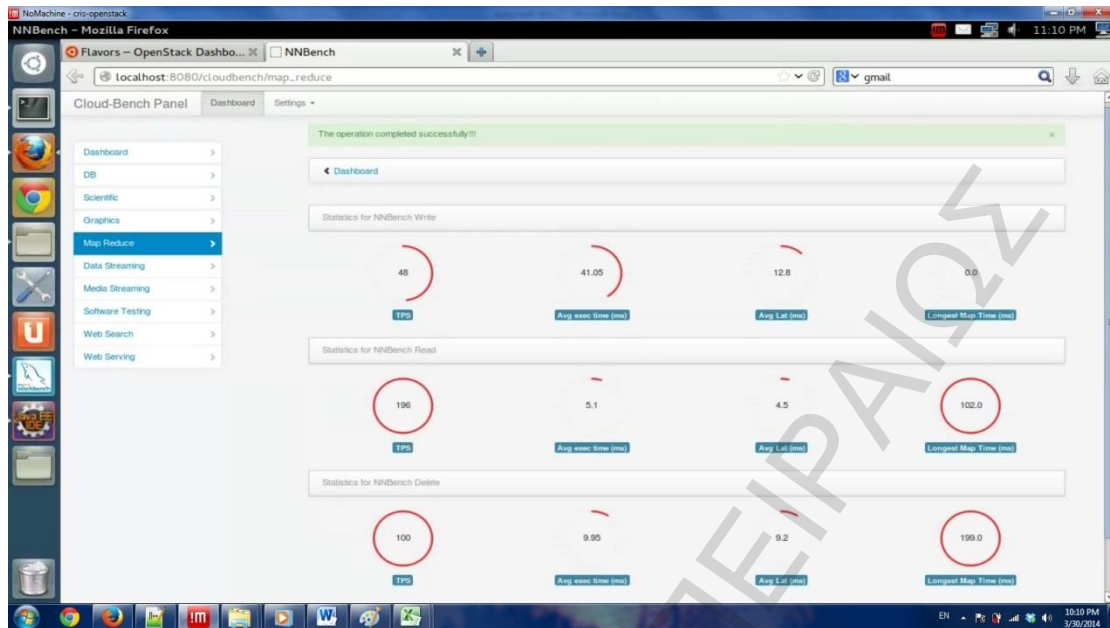


Εικόνα 53 Map Reduce Final DFSIO

Dfsio output

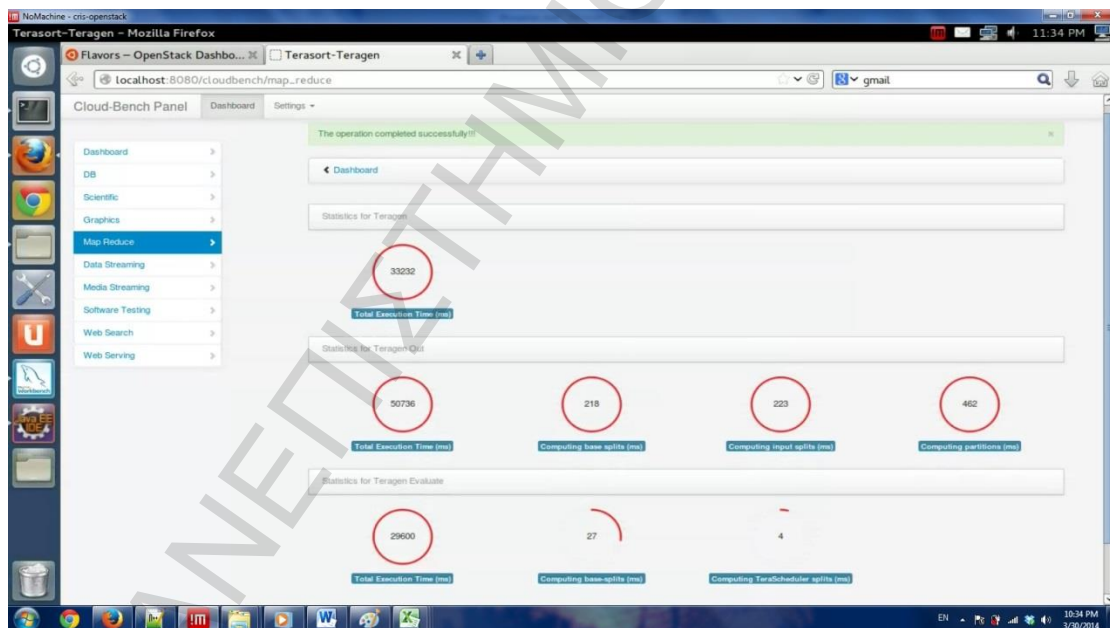
“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

NNBENCH Write



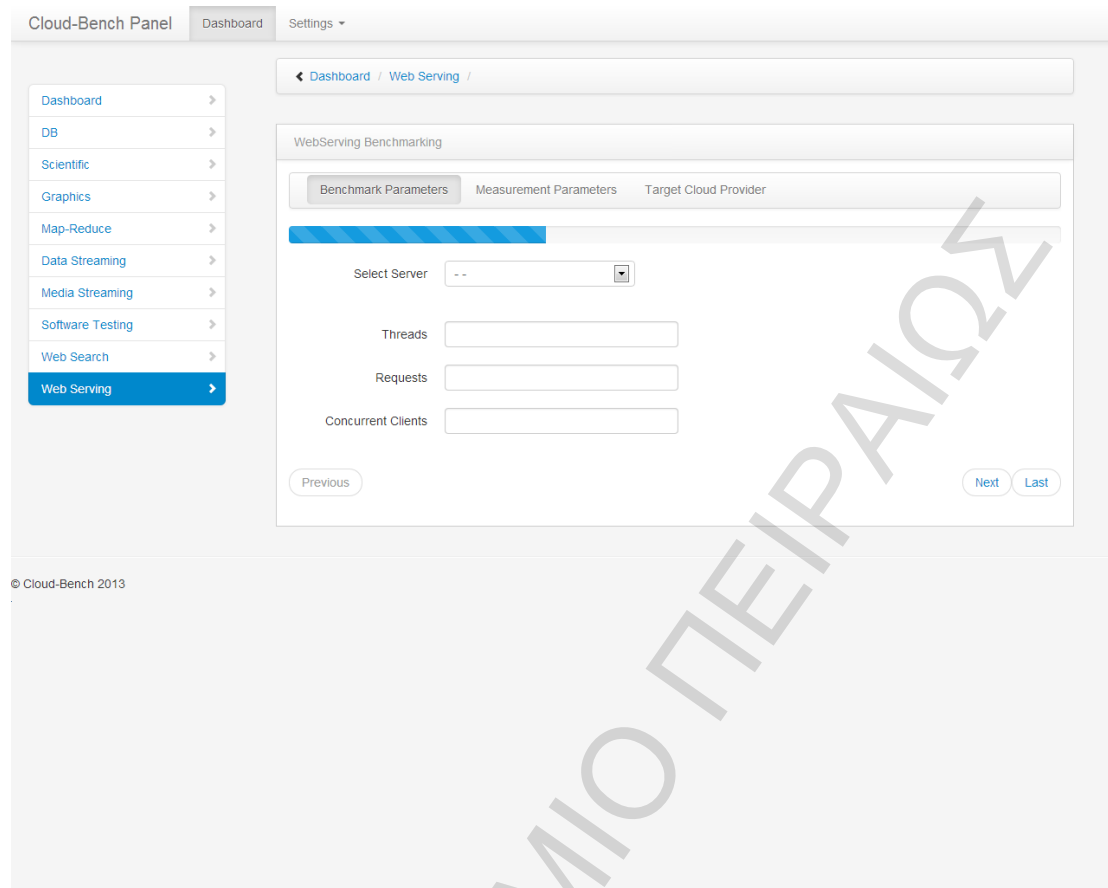
Εικόνα 54 NNBENCH Write

TERAGEN



Εικόνα 55 TERAGEN Outputs

Web Serving

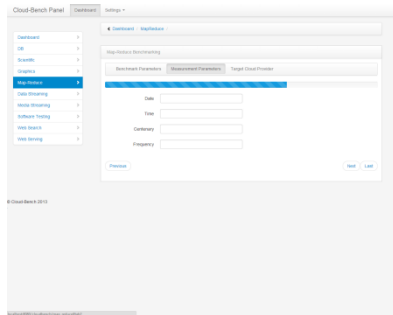


Εικόνα 56 Web Serving Step 1

Ο χρήστης πρέπει να επιλέξει από το πτυσσόμενο μενού, το Server που θα γίνει το τεστ, τα νήματα εκτέλεσης, πόσα ταυτόχρονα requests θα λαμβάνει ο server και πόσοι ταυτόχρονοι clients θα είναι συνδεδεμένοι σε αυτόν.

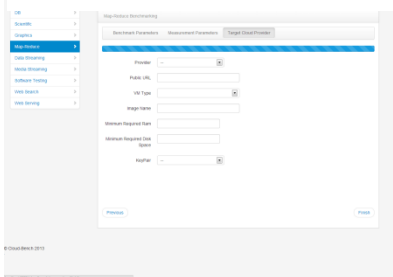
“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Οι επόμενες δύο εικόνες είναι ίδιες με του κεφ. 4.1 (δεύτερο και τρίτο βήμα).



Καταχώριση παραμέτρων ημερομηνίας, ώρας και επαναληπτικότητας / συχνότητας.

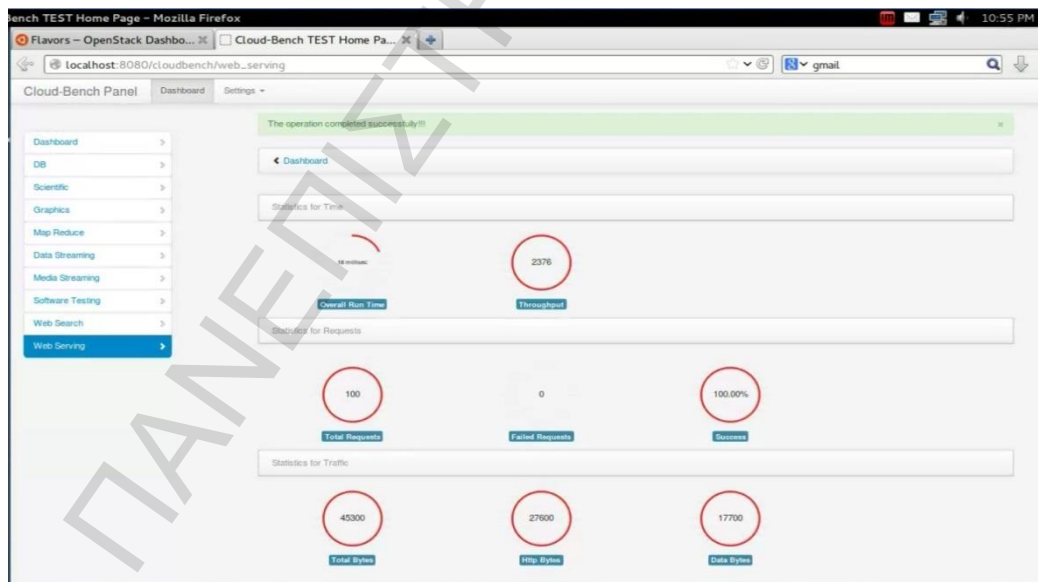
Εικόνα 57 Web Serving Step2



Καταχώριση παραμέτρων για τον cloud provider.

Εικόνα 58 Web Serving Step3

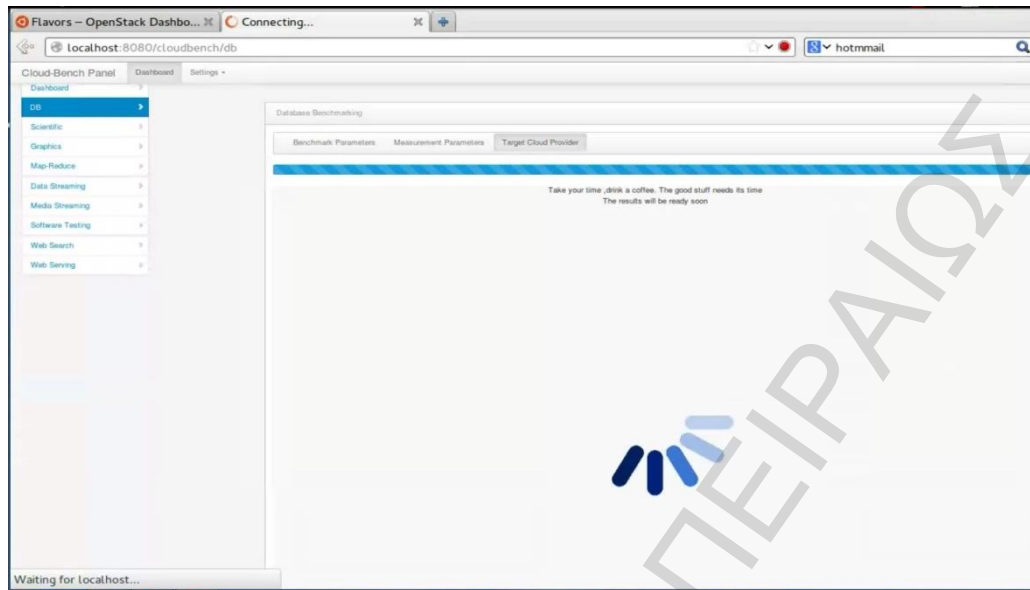
Τελικά η εμφανίζεται η οθόνη με τα αποτελέσματα.



Εικόνα 59 Web Serving Final

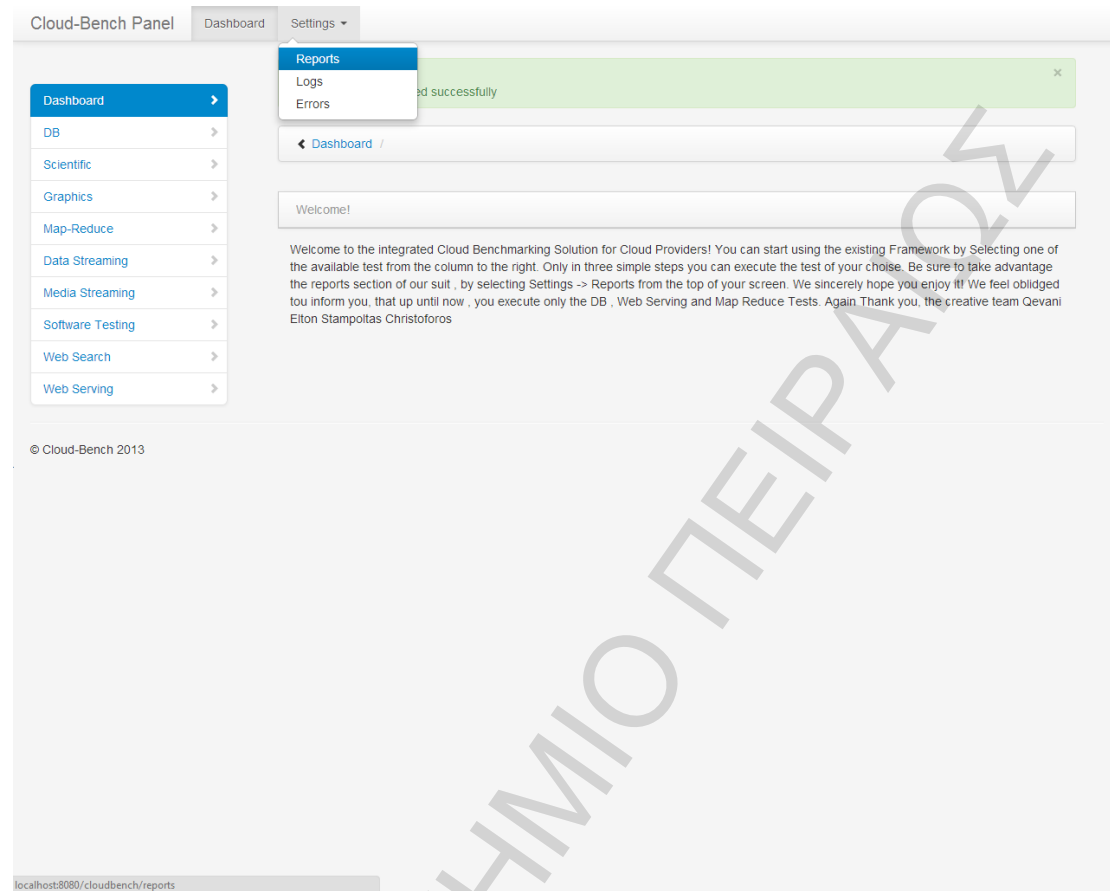
“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Κάθε τεστ χρονικά μπορεί να απαιτεί να είναι απαιτητικό, Για το λόγο αυτό έχουμε ενσωματώσει ένα γραφικό το οποίο ειδοποιεί το χρήστη ότι η διαδικασία εκτελείται . Αυτό εμφανίζεται στο παρακάτω σχήμα και είναι κοινό για όλα τα τεστ.



Εικόνα 60 Waiting Screen

Reports



Εικόνα 61 Welcome Screen

Για να μπορέσει ο χρήστης να χρησιμοποιήσει τη λειτουργία των reports πρέπει στην αρχική σελίδα να επιλέξει το drop-down menu Settings και έπειτα το Reports.

The screenshot shows the 'Reports Form' in the Cloud-Bench Panel. The form is titled 'Database Benchmarking' and contains the following elements:

- A 'Report Creator' button at the top.
- Input fields for 'Input Weights for Average' and 'Input Weights for Cost'.
- Input fields for 'Start Date' and 'End Date'.
- Dropdown menus for 'Select VM Instance' and 'Select Cloud Test'.
- 'Previous' and 'Finish' buttons at the bottom.

© Cloud-Bench 2013

Εικόνα 62 Reports Form

Έπειτα να καταχωρίσει τα αντίστοιχα βάρη για τους μέσους όρους των κρίσιμων παραγόντων και του κόστους αντίστοιχα. Επίσης πρέπει να δηλώσει μεταξύ ποιων ημερομηνιών έτρεξε το test που επιθυμεί να ανακαλέσει , καθώς επίσης και το είδος των VM που χρησιμοποιήθηκαν για αυτό το τεστ. Τέλος , επιλέγει τεστ.

Στην εικόνα που ακολουθεί ο χρήστης έχει δώσει βάρη 0.2 και 0.8 αντίστοιχα. Εδώ θα πρέπει να σημειωθεί πως ο χρήστης πρέπει να δηλώνει βάρη , πραγματικούς αριθμούς , των οποίων το άθροισμα είναι 1 μονάδα (1.0). Στην περίπτωση που γίνει κάποιο σφάλμα , ο χρήστης ενημερώνεται με Javascript alert και προχωράει στις απαραίτητες διορθώσεις.

Αναλόγως με την επιλογή του τεστ , τότε εμφανίζονται νέα πεδία τα οποία ο χρήστης συμπληρώνει . Για παράδειγμα στην παρακάτω εικόνα έχουμε επιλέξει , DB test , με MongoDB και workloada.

Cloud-Bench Panel Dashboard Settings

Database Benchmarking

Report Creator

Input Weights for Average: 0.2

Input Weights for Cost: 0.8

Start Date: 2014-03-01

End Date: 2014-03-31

Select VM Instance: m1.tiny

Select Cloud Test: Database

Select Database: MongoDB

Select Workload: Workload A

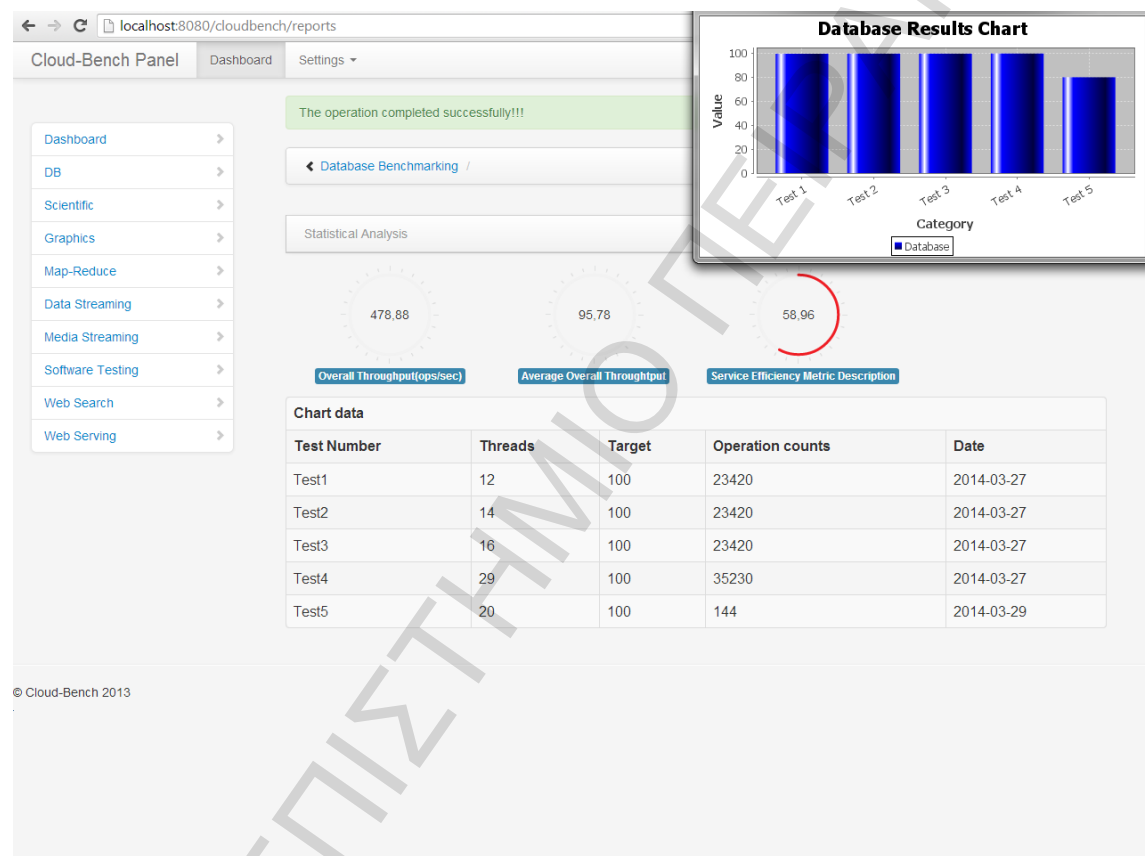
Previous Finish

© Cloud-Bench 2013

Εικόνα 63 Reports Select Test

“A Unified Benchmark Framework for Cloud Deployed Applications Tool in IaaS”

Με την ολοκλήρωση της συμπλήρωσης και την υποβολή της φόρμας , εμφανίζονται τα αποτελέσματα, όπως φαίνεται παρακάτω. Τόσο το Service Efficiency Metric Description όσο και η γραφική παράσταση για κάθε τεστ που έχει τρέξει . Τα αποτελέσματα ποικίλουν. Υπάρχει η γραφική αναπαράσταση του κάθε τεστ, (JFrame) , αλλά παρουσιάζεται αναλυτικά και ο πίνακας δεδομένων. Συγκεκριμένα εμφανίζονται τόσο οι παράμετροι της γραφικής παράστασης αλλά και δευτερεύουσες πληροφορίες που αφορούν το τεστ.



Εικόνα 64 Reports Final

Μελλοντική Μελέτη

Για το web tool front-end θα μπορούσε να χρησιμοποιηθεί τεχνολογία JQuery και Ajax για πιο γρήγορη φόρτωση των αποτελεσμάτων. Θα μπορούσε επίσης να δημιουργηθεί ένα add-on etool το οποίο θα ήταν υπεύθυνο για την εξαγωγή των αποτελεσμάτων σε ένα αρχείο με τη μορφή pdf ή xls. Στη σελίδα των αποτελεσμάτων είναι πιθανή, η ύπαρξη πιο εξειδικευμένων συναρτήσεων φόρτωσης αποτελεσμάτων για να μπορεί ο χρήστης να συγκρίνει ταυτόχρονα πολλαπλά αποτελέσματα. Στα αποτελέσματα που επιστρέφονται αμέσως μετά την εκτέλεση του τεστ, θα ήταν δυνατόν να χρησιμοποιηθούν πιο εξειδικευμένα γραφήματα που να παρουσιάζουν περισσότερα αποτελέσματα και πιο συγκεκριμένα.

Σαν βάση δεδομένων για την αποθήκευση των αποτελεσμάτων θα μπορούσε να χρησιμοποιηθεί, μη σχεσιακή βάση δεδομένων, έτσι ώστε τα αρχεία αποτελεσμάτων ή αποτελέσματα να αποθηκεύονται πιο γρήγορα και αποδοτικά.

Βιβλιογραφικές Αναφορές

1. <http://www.businessdictionary.com/definition/benchmarking.html>
2. <http://www.argi.com.my/whatispage/benchmarking.htm>
3. http://www.ict-fire.eu/fileadmin/publications/Whitepaperonbenchmarking_V2.pdf
4. P. Colella, Defining Software Requirements for Scientific Computing, 2004.
5. K. Asanovic et al., "The Landscape of Parallel Computing Research: A View from Berkeley," Electrical Engineering and Computer Sciences, University of California at Berkeley, 2006.
7. K. Asanovic et al., "A View of the Parallel Computing Landscape," Communications of the ACM, vol. 52, no. 10, pp. 56-67, 2009.
9. SPEC. (2011) Standard Performance Evaluation Corporation. [Online].
10. <http://www.spec.org/index.html>
11. EC FP7-ICT IRMOS Project. [Online]. <http://www.irmosproject.eu/>
12. EC FP7-WP9 Benchmarking. [Online]. <http://www.onelab.eu/index.php>
13. <http://cloudcomputing4u.wordpress.com/2012/05/04/%CE%B7-%CE%AD%CE%BD%CE%BD%CE%BF%CE%B9%CE%B1-%CF%84%CE%BF%CF%85-%CF%85%CF%80%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CF%83%CF%84%CE%B9%CE%BA%CE%BF%CF%8D-%CE%BD%CE%AD%CF%86%CE%BF%CF%85%CF%82-cloudcomputing/>
14. <http://nemertes.lis.upatras.gr/jspui/bitstream/10889/6594/1/thesis.pdf>
15. <http://www.cloud-competence-center.com/understanding/cloud-computing-deployment-models/>
16. <http://occi-wg.org/>
17. Open Virtualization Format Specification v1.0.0, September 2009, available at <http://xml.coverpages.org/DMTF-OVF-v10-DSP0243.pdf>
18. Debusmann, M.; Keller, A., "SLA-driven management of distributed systems using the common information model," Integrated Network Management, 2003. IFIP/IEEE Eighth International Symposium on , vol., no., pp. 563-576, 24-28 March 2003.
19. Peter Mell, Timothy Grance, "The NIST Definition of Cloud Computing", Special Publication 800-145, National Institute of Standards and Technology, September 2011
20. T. Erl, "Service-oriented Architecture: Concepts, Technology, and Design", Upper Saddle River: Prentice Hall PTR. ISBN 0-13-185858-0, 2005.
21. I. Foster, C. Kesselman, S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations" International Journal Supercomputer Applications, Vol. 15, No. 3, 2001.
22. <http://occi-wg.org/>
23. Open Virtualization Format Specification v1.0.0, September 2009, available at <http://xml.coverpages.org/DMTF-OVF-v10-DSP0243.pdf>
24. Debusmann, M.; Keller, A., "SLA-driven management of distributed systems using the common information model," Integrated Network Management, 2003. IFIP/IEEE Eighth International Symposium on , vol., no., pp. 563-576, 24-28 March 2003.

25. G. Wang, A. Chen, C. Wang, C. Fung, and S. Uczekaj, “Integrated Quality of Service (QoS) Management in Service-Oriented Enterprise Architectures”, In Proceedings of the 8th International IEEE Enterprise Distributed Object Computing Conference (EDOC), Monterey, California, September 2004.
26. Zhengting He, Cheng Peng, Aloysius Mok, "A Performance Estimation Tool for Video Applications," *rtas*, pp. 267-276, 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'06), 2006
27. Siegfried Benkner, Gerhard Engelbrecht, "A Generic QoS Infrastructure for Grid Web Services," *aict-iciw*, p. 141, Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT-ICIW'06), 2006
28. Chen, Iyer, Liu, Milojcic, Sahai, "SLA Decomposition: Translating Service Level Objectives to System Level Thresholds," *icac*, p. 3, Fourth International Conference on Autonomic Computing (ICAC'07), 2007
29. Jae W. Lee, Krste Asanovic, "METERG: Measurement-Based End-to-End Performance Estimation Technique in QoS-Capable Multiprocessors," *rtas*, pp. 135-147, 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'06), 2006
30. Peer Hasselmeyer, Bastian Koller, Lutz Schubert, Philipp Wieder: Towards SLA-Supported Resource Management. *HPCC 2006*: 743-752
31. Jarvis, S. A., Spooner, D. P., Keung, H. N., Cao, J., Saini, S., and Nudd, G. R. 2006. Performance prediction and its use in parallel and distributed computing systems. *Future Gener. Comput. Syst.* 22, 7 (Aug. 2006), 745-754.
32. Oana Florescu, Menno de Hoon, Jeroen Voeten, and Henk Corporaal. Probabilistic modelling and evaluation of soft real-time embedded systems. In Proceedings of SAMOS VI, LNCS 4017, 2006.
33. Above the Clouds: A Berkeley View of Cloud Computing
Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia
34. Above the Clouds: A Berkeley View of Cloud Computing (Michael Armbrust, Technical Report No. UCB/EECS-2009-28, February 10, 2009)
35. [21] Saurabh Kumar Garg, Steve Versteeg, Rajkumar Buyya, A framework for ranking of Cloud computing services, *Future Generation Computer Systems*, Volume 29, Issue 4, June 2013, Pages 1012-1023, ISSN 0167-739X, <http://dx.doi.org/10.1016/j.future.2012.06.006>.
36. Ang Li, Xiaowei Yang, Srikanth Kandula, and Ming Zhang. 2010. CloudCmp: comparing public Cloud providers. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (IMC '10). ACM, New York, NY, USA, 1-14. DOI=10.1145/1879141.1879143 <http://doi.acm.org/10.1145/1879141.1879143>
37. www.cloudharmony.com/
38. [www.https://cloudsleuth.net/](https://cloudsleuth.net/)
39. www.openbenchmarking.org/
40. I. Raicu, Z. Zhang, M. Wilde, I. T. Foster, P. H. Beckman, K. Iskra, and B. Clifford, “Toward loosely coupled programming on petascale systems,” in *SC*. ACM, 2008, p. 22.

41. A. Buble, L. Bulej, and P. Tuma, “Corba benchmarking: A course with hidden obstacles,” in IPDPS, 2003, p. 279.
42. P. Brebner, E. Cecchet, J. Marguerite, P. Tuma, O. Ciuhandu, B. Dufour L. Eeckhout, S. Frénot, A. S. Krishna, J. Murphy, and C. Verbrugge, “Middleware benchmarking: approaches, results, experiences,” *Concurrency and Computation: Practice and Experience*, vol. 17, no. 15, pp. 1799–1805, 2005.
43. J. Gray, Ed., *The Benchmark Handbook for Database and Transaction Systems*, 2nd ed. Morgan Kaufmann, 1993.
44. A. B. Downey and D. G. Feitelson, “The elusive goal of workload characterization,” *SIGMETRICS Performance Evaluation Review*, vol. 26, no. 4, pp. 14–29, 1999.
45. S. J. Chapin, W. Cirne, D. G. Feitelson, J. P. Jones, S. T. Leutenegger, U. Schwiegelshohn, W. Smith, and D. Talby, “Benchmarks and standards for the evaluation of parallel job schedulers,” in *JSSPP*, 1999, pp. 67–90.
46. E. Frachtenberg and D. G. Feitelson, “Pitfalls in parallel job scheduling evaluation,” in *JSSPP*, 2005, pp. 257–282.
47. A. Iosup, D. H. J. Epema, C. Franke, A. Papaspyrou, L. Schley, B. Song, and R. Yahyapour, “On grid performance evaluation using synthetic workloads,” in *JSSPP*, 2006, pp. 232–255.
48. R. Jain, Ed., *The Art of Computer Systems Performance Analysis*. John Wiley and Sons Inc., 1991.
49. J. N. Amaral, “How did this get published? Pitfalls in experimental evaluation of computing systems,” *LTES talk*, 2012.
50. A. Iosup and D. H. J. Epema, “GrenchMark: A framework for analyzing, testing, and comparing grids,” in *CCGrid*, 2006, pp. 313–320.
51. M. Oskin, F. T. Chong, and M. K. Farrens, “Hls: combining statistical and symbolic simulation to guide microprocessor designs,” in *ISCA*, 2000, pp. 71–82.
52. L. Eeckhout, S. Nussbaum, J. E. Smith, and K. D. Bosschere, “Statistical simulation: Adding efficiency to the computer designer’s toolbox,” *IEEE Micro*, vol. 23, no. 5, pp. 26–38, 2003.
53. D. Genbrugge and L. Eeckhout, “Chip multiprocessor design space exploration through statistical simulation,” *IEEE Trans. Computers*, vol. 58, no. 12, pp. 1668–1681, 2009.
54. R. H. Saavedra and A. J. Smith, “Analysis of benchmark characteristics and benchmark performance prediction,” *ACM Trans. Comput. Syst.*, vol. 14, no. 4, pp. 344–384, 1996.
55. S. Sharkawi, D. DeSota, R. Panda, R. Indukuru, S. Stevens, V. E. Taylor, and X. Wu, “Performance projection of hpc applications using speccfp2006 benchmarks,” in *IPDPS*, 2009, pp. 1–12.
56. D. Villegas, A. Antoniou, S. M. Sadjadi, and A. Iosup, “An analysis of provisioning and allocation policies for infrastructure-as-a-service clouds,” in *CCGrid*, 2012, pp. 612–619.
57. B. Schroeder, A. Wierman, and M. Harchol-Balter, “Open versus closed: A cautionary tale,” in *NSDI*, 2006.

58. A. Georges, D. Buytaert, and L. Eeckhout, “Statistically rigorous java performance evaluation,” in OOPSLA, 2007, pp. 57–76.
59. A. Vahdat, K. Yocum, K. Walsh, P. Mahadevan, D. Kostic, J. S. Chase, and D. Becker, “Scalability and accuracy in a large-scale network emulator,” in OSDI, 2002.
60. K. V. Vishwanath, A. Vahdat, K. Yocum, and D. Gupta, “Modelnet: Towards a datacenter emulation environment,” in Peer-to-Peer Computing,
61. 2009, pp. 81–82.
62. A. Iosup, N. Yigitbasi, and D. H. J. Epema, “On the performance variability of production cloud services,” in CCGRID, 2011, pp. 104–113.
63. D. H. J. Epema, “Performance analysis of cloud computing services for many-tasks scientific computing,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 6, pp. 931–945, 2011.
64. A. Iosup, M. Jan, O. O. Sonmez, and D. H. J. Epema, “On the dynamic resource availability in grids,” in GRID. IEEE, 2007, pp. 26–33.
65. D. Kondo, B. Javadi, A. Iosup, and D. H. J. Epema, “The failure trace archive: Enabling comparative analysis of failures in diverse distributed systems,” in CCGrid, 2010, pp. 398–407.
66. R. H. Arpaci-Dusseau, A. C. Arpaci-Dusseau, A. Vahdat, L. T. Liu, T. E. Anderson, and D. A. Patterson, “The interaction of parallel and sequential workloads on a network of workstations,” in SIGMETRICS, 1995, pp. 267–278.
67. R. Krebs, C. Momm, and S. Kounev, “Metrics and techniques for quantifying performance isolation in cloud environments,” in Int’l. ACM SIGSOFT conference Quality of Software Architectures (QoSA), 2012, pp. 91–100.
68. N. Huber, M. von Quast, M. Hauck, and S. Kounev, “Evaluating and modeling virtualization performance overhead for cloud environments,” in CLOSER, 2011, pp. 563–573.
69. A. Iosup and D. H. J. Epema, “Grid computing workloads,” *IEEE Internet Computing*, vol. 15, no. 2, pp. 19–26, 2011.
70. A. Iosup, O. O. Sonmez, S. Anoep, and D. H. J. Epema, “The performance of bags-of-tasks in large-scale distributed systems,” in HPDC. ACM, 2008, pp. 97–108.
71. K. Kim, K. Jeon, H. Han, S. G. Kim, H. Jung, and H. Y. Yeom, “Mrbench: A benchmark for mapreduce framework,” in ICPADS, 2008, pp. 11–18.
72. G. Wang, A. R. Butt, P. Pandey, and K. Gupta, “Using realistic simulation for performance analysis of MapReduce setups,” in HPDC Workshops, 2009, pp. 19–26.
73. M. Zaharia, D. Borthakur, J. S. Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, “Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling,” in EuroSys, 2010, pp. 265–278.
74. A. Ganapathi, Y. Chen, A. Fox, R. H. Katz, and D. A. Patterson, “Statistics-driven workload modeling for the cloud,” in ICDE Work-shops, 2010, pp. 87–92.
75. Y. Chen, A. Ganapathi, R. Griffith, and R. H. Katz, “The case for evaluating mapreduce performance using workload suites,” in MASCOTS, 2011, pp. 390–399.
76. T. A. De Ruiter and A. Iosup, “A workload model for MapReduce,” TU Delft MSc thesis, Jun 2012, [Online] Available: <http://repository.tudelft.nl/view/ir/uuid:1647e1cb-84fd-46ca-b1e1-21aaf38ef30b/>.
77. <http://repository.tudelft.nl/view/ir/uuid:1647e1cb-84fd-46ca-b1e1-21aaf38ef30b/>. Last accessed Oct 2012.

78. A. J. Oliner, A. Ganapathi, and W. Xu, “Advances and challenges in log analysis,” *Commun. ACM*, vol. 55, no. 2, pp. 55–61, 2012.
79. C. Momm and R. Krebs, “A qualitative discussion of different approaches for implementing multi-tenant saas offerings,” in *Software Engineering (Workshops)*, 2011, pp. 139–150.
80. R. Krebs, C. Momm, and S. Kounev, “Architectural concerns in multi-tenant saas applications,” in *CLOSER*, 2012, pp. 426–431.
81. P. Brebner, “Is your cloud elastic enough?: performance modelling the elasticity of infrastructure as a service (iaas) cloud applications,” in *ICPE*, 2012, pp. 263–266.
82. S. Islam, K. Lee, A. Fekete, and A. Liu, “How a consumer can measure elasticity for cloud platforms,” in *ICPE*, 2012, pp. 85–96.
83. E. Walker, “The real cost of a cpu hour,” *IEEE Computer*, vol. 42, no. 4, pp. 35–41, 2009.
84. Y.-K. Kwok and I. Ahmad, “Benchmarking and comparison of the taskgraph scheduling algorithms,” *J. Parallel Distrib. Comput.*, vol. 59, no. 3, pp. 381–422, 1999.
85. A. Antoniou and A. Iosup, “Performance evaluation of cloud infrastructure using complex workloads,” *TU Delft MSc thesis*,
86. Mar 2012, [Online] Available: <http://repository.tudelft.nl/view/ir/uuid:d8eda846-7e93-4340-834a-de3e4aa93f8b/>. Last accessed Oct 2012.
87. S. Ostermann and R. Prodan, “Impact of variable priced Cloud resources on scientific workflow scheduling,” in *Euro-Par 2012*
88. *Parallel Processing*, ser. *Lecture Notes in Computer Science*, C. Kaklamanis, T. Papatheodorou, and P. G. Spirakis, Eds., vol. 7484.
89. Springer, 2012, pp. 350–362. [Online]. Available: <http://www.springerlink.com/content/v4q338161171r42v/fulltext.pdf>
90. R. Prodan, M. Sperk, and S. Ostermann, “Evaluating high-performance computing on google app engine,” *IEEE Software*, vol. 29, no. 2, pp. 52–58, 2012.
91. D. Feitelson, “Parallel Workloads Archive,” <http://www.cs.huji.ac.il/labs/parallel/workload/>.
92. A. Iosup, H. Li, M. Jan, S. Anoep, C. Dumitrescu, L. Wolters, and D. H. J. Epema, “The grid workloads archive,” *Future Gener. Comput. Syst.*, vol. 24, no. 7, pp. 672–686, 2008.
93. Y. Guo and A. Iosup, “The Game Trace Archive,” in *NETGAMES*, 2012, pp. 1–6.
94. B. Zhang, A. Iosup, J. Pouwelse, and D. Epema, “The peer-to-peer trace archive: design and comparative trace analysis,” in *ACM CoNEXT Student Workshop*. ACM, 2010, pp. 21:1–2. [Online]. Available: <http://doi.acm.org/10.1145/1921206.1921229>
95. http://en.wikipedia.org/wiki/Amazon_Web_Services
96. <http://www.rackspace.com/about/>
97. <http://www.rackspace.com/cloud/servers/pricing/>
98. <http://en.wikipedia.org/wiki/VCloud>
99. <http://www.vmware.com/products/vsphere/features-endpoint>
100. <http://cloudharmony.com/>
101. <https://cloudsleuth.net/>