

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
«ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΛΗΡΟΦΟΡΙΚΗΣ»

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

Τίτλος μεταπτυχιακής διατριβής	ΠΡΟΣΟΜΟΙΩΣΗ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΝΕΦΩΝ
Όνοματεπώνυμο φοιτητή	ΣΠΑΝΟΥΔΑΚΗΣ ΠΑΥΛΟΣ
Πατρώνυμο	ΕΜΜΑΝΟΥΗΛ
Αριθμός μητρώου	ΜΠΣΠ14082
Επιβλέπων	ΧΡΗΣΤΟΣ ΔΟΥΛΗΓΕΡΗΣ, ΚΑΘΗΓΗΤΗΣ

Ημερομηνία παράδοσης: Ιούλιος 2017

Περιεχόμενα

1	Εισαγωγή	5
2	Προσομοιωτής Cloudsim.....	6
2.1	Αρχιτεκτονική του Cloudsim.....	6
2.2	Επεκτάσεις του Cloudsim	7
2.3	Χαρακτηριστικά & Δυνατότητες.....	7
2.4	Πειράματα και αποτελέσματα	7
2.4.1	Εκτέλεση 1ης προσομοίωσης.....	7
2.4.2	Εκτέλεση 2ης προσομοίωσης.....	9
2.4.3	Εκτέλεση 3ης προσομοίωσης.....	11
2.4.4	Συμπεράσματα και αποτελέσματα	14
3	Προσομοιωτής Cloudreports.....	16
3.1	Πειράματα και αποτελέσματα	16
3.2	Συμπεράσματα	20
4	Προσομοιωτής Cloudsurf	21
4.1	Χαρακτηριστικά & δυνατότητες.....	21
4.2	Λειτουργίες.....	21
4.3	Πειράματα και αποτελέσματα	22
5	Προσομοιωτής iCanCloud	23
5.1	Χαρακτηριστικά και δυνατότητες	23
5.2	Αρχιτεκτονική του iCanCloud	23
5.3	Πειράματα και αποτελέσματα προσομοιωτή iCanCloud.....	25
5.3.1	Καθορισμός παραμέτρων.....	25
5.3.2	Δημιουργία αρχείων ρυθμίσεων.....	29
5.3.3	Αρχεία αποτελεσμάτων.....	29
5.3.4	Διαγράμματα αποτελεσμάτων.....	31
5.4	Συμπεράσματα	34
6	Προσομοιωτής CloudAnalyst.....	35
6.1	Χαρακτηριστικά και δυνατότητες	35
6.2	Πειράματα και αποτελέσματα του CloudAnalyst	38
6.2.1	1° Πείραμα.....	38
6.2.2	Αποτελέσματα	39

6.2.3	Συμπεράσματα	40
6.3	2 ^ο Πείραμα.....	40
6.3.1	Παράμετροι πειράματος	41
6.3.2	Αποτελέσματα πειράματος	42
6.3.3	Συμπεράσματα	44
7	Συμπεράσματα – Σύνοψη.....	45
8	Αναφορές.....	46
9	Κατάλογος εικόνων	48
10	Παράρτημα εγχειριδίων χρήσης προσομοιωτών	49

ΠΡΟΣΟΜΟΙΩΣΗ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΝΕΦΩΝ

Περίληψη

Η υπολογιστική νέφους είναι ένας τύπος υπολογιστικής βασισμένη στο ιντερνέτ που παρέχει διαμοιραζόμενους πόρους και δεδομένα σε υπολογιστές και άλλες συσκευές κατ' απαίτηση. Είναι ένα μοντέλο για ενεργοποίηση ευρείας, κατ' απαίτηση πρόσβαση σε ένα κοινόχρηστο χώρο από ρυθμιζόμενους υπολογιστικούς πόρους (π.χ δίκτυα υπολογιστών, εξυπηρετητές, αποθηκευτικό χώρο, εφαρμογές και υπηρεσίες), τα οποία μπορούν γρήγορα να παρέχουν υπηρεσίες με ελάχιστη προσπάθεια διαχείρισης. Λύσεις υπολογιστικής νέφους και αποθήκευσης παρέχουν στους χρήστες και τις επιχειρήσεις διάφορες δυνατότητες για αποθήκευση και επεξεργασία των δεδομένων τους είτε ιδιωτικά είτε σε κέντρα δεδομένων, που πιθανόν στεγάζονται μακριά από το εύρος των χρηστών σε απόσταση που μπορεί να κυμαίνεται από μια πόλη σε ολόκληρο τον κόσμο. Η υπολογιστική νέφους βασίζεται στο διαμοιρασμό των πόρων για να επιτύχει συνοχή και οικονομία παρόμοια με μια χρησιμότητα μέσω ενός δικτύου ηλεκτρικής ενέργειας. Στη παρούσα μεταπτυχιακή διατριβή θα γίνει ανάλυση μερικών από τους πιο πρόσφατους και δημοφιλείς προσομοιωτές υπολογιστικής νέφους (Cloudsim [1], Cloudsurf [4], Cloudreports [3], iCanCloud [5,19], Cloudanalyst [3,6]). Θα γίνει καταγραφή των χαρακτηριστικών και των δυνατοτήτων τους καθώς και η σκοπιμότητα του κάθε προσομοιωτή. Στη συνέχεια θα γίνουν προσομοιώσεις (πειράματα) ξεχωριστά για τον κάθε προσομοιωτή σύμφωνα με κάποιες προδιαμορφωμένες έρευνες έτσι ώστε να γίνει σύγκριση και επιβεβαίωση των πιθανών ιδίων αποτελεσμάτων που θα παραχθούν. Τέλος θα γραφτούν εγχειρίδια χρήσης για τον κάθε προσομοιωτή ώστε να κατανοηθεί η δυνατότητα εγκατάστασης και χρήσης τους.

Λέξεις κλειδιά: Υπολογιστική νέφους, κέντρα δεδομένων, υπολογιστικοί πόροι

Abstract

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications and services), which can be rapidly provisioned and released with minimal management effort. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in either privately owned, or third-party data centers that may be located far from the user—ranging in distance from across a city to across the world. Cloud computing relies on sharing of resources to achieve coherence, similar to a utility over an electricity network. In this postgraduate dissertation will be analyzed some of the most recent and popular cloud simulators (Cloudsim [1], Cloudsurf [4], Cloudreports [3], iCanCloud [5,19], Cloudanalyst [3,6]). It will be recorded the features and the capabilities as well as the feasibility of each simulator. Then will be done simulations (experiments) individually for each simulator according to some preformed surveys, so that become comparison and confirmation same possible results which will be produced. In conclusion will be written manuals of each simulator so that be understood the installation and use capability.

Keywords: Cloud computing, data centers, computing resources

1 Εισαγωγή

Η υπολογιστική νέφος είναι μια από τις αναδυόμενες τεχνολογίες που με την ευκολία πρόσβασης που διαθέτει και με την ευρεία δυνατότητα εφαρμογής, επιτρέπει στους πελάτες να προσελκύνονται σε αυτό και συνεπώς δημιουργεί θέματα που πρέπει να ξεπεραστούν σε αυτό το τομέα. Η ανάπτυξη πραγματικού νέφους για δοκιμές ή για εμπορική χρήση είναι δαπανηρή χρηματικά. Αυτό το μειονέκτημα εξαλείφθηκε με τη χρήση των προσομοιωτών νέφους. Υπάρχουν πολλά οφέλη από τη χρήση προσομοιωτών ιδιαίτερα στο νέφος όπως i) **Ελάχιστο κόστος:** Αφού είναι μόνο λογισμικό κοστίζει πολύ λιγότερο συγκριτικά με το υλικό ii) **Επαναληπτικό και ελεγχόμενο:** μπορούμε να επαναλάβουμε περισσότερες από μια φορές μέχρι να πάρουμε το επιθυμητό αποτέλεσμα. iii) **Περιβάλλον:** Παρέχει εξέλιξη για διαφορετικά σενάρια κάτω από διαφορετικό φόρτο και κόστος μέτρησης [14]. Ο προσομοιωτής υπολογιστικού νέφους [8] βοηθά στη μοντελοποίηση διαφόρων ειδών εφαρμογών νέφους δημιουργώντας κέντρα δεδομένων, εικονικά μηχανήματα και πολλές χρησιμότητες οι οποίες μπορούν να προστεθούν για τη διαχείριση του, επιπλέον κάνοντας το πολύ εύκολο στην ανάλυση του. Μέχρι σήμερα πολλοί προσομοιωτές νέφους έχουν προταθεί και είναι διαθέσιμοι για χρήση. Αυτοί οι προσομοιωτές δημιουργήθηκαν για ειδικό σκοπό, έχοντας ποικίλα χαρακτηριστικά εξυπηρετώντας διαφορετικό σκοπό κάθε ένας από αυτούς ανάλογα τις ανάγκες του χρήστη.

Η βασική ιδέα της υπολογιστικής νέφους που διαφέρει από το grid computing είναι οι μοναδικές υπηρεσίες και η εικονικοποίηση. Οι υπηρεσίες κατηγοριοποιούνται σαν i) Software as a service (SaaS), το οποίο ασχολείται με εφαρμογές, διαχείριση και διεπαφές χρήστη. Π.χ. Microsoft office 365 [9], Salesforce [10], Dropbox [11] κτλ. ii) Platform as a Service (PaaS), το οποίο παρέχει περιβάλλον για ανάπτυξη εφαρμογής, εικονικές μηχανές κτλ, iii) infrastructure as a Service (IaaS) το οποίο παρέχει το υπολογιστικό υλικό, εικονικό χώρο, εικονική υποδομή πχ Amazon EC2 [12] Eucalyptus [13].

Για ανάπτυξη και ανάλυση οποιουδήποτε περιβάλλοντος υπολογιστικής νέφους με τη βοήθεια των προσομοιωτών χρειάζεται να κατανοήσουμε τους προσομοιωτές νέφους με τα υπέρ και τα κατά τους. Οι πιο πρόσφατοι προσομοιωτές είναι: 1) *Cloudsim* [1] πρόκειται για το πιο δημοφιλές εργαλείο προσομοίωσης διαθέσιμο περιβάλλον υπολογιστικής νέφους, 2) *CloudAnalyst* [15] βασίζεται στην αρχιτεκτονική του *cloudsim* έχοντας επιπρόσθετα χαρακτηριστικά, 3) *GreenCloud* [16] πρόκειται για προσομοιωτή που αποτελεί επέκταση του NS-2 [17], διαφέρει από τον *Cloudsim* και δημιουργήθηκε ειδικά για περιβάλλοντα εύρεσης ενέργειας, 4) *MDCSim* [18] δημιουργήθηκε για σχεδίαση και ανάλυση μεγάλης κλίμακας πολλαπλών επιπέδων κέντρων δεδομένων. Επιτρέπει τη μέτρηση ισχύος και την ανάλυση κάθε επιπέδου της τριστρωματικής (3-layer) αρχιτεκτονικής, τροποποιώντας κάθε στρώμα χωρίς να επηρεάσει τα υπόλοιπα, 5) *iCanCloud* [19] σχεδιάστηκε για να βελτιστοποιήσει τα μειονεκτήματα των προσομοιωτών όπως το *GreenCloud* και *Cloudsim*. Είναι πολύ ευέλικτος και το μοναδικό του χαρακτηριστικό είναι ότι ο χρήστης μπορεί να προσαρμόσει το πυρήνα του hypervisor (VMM) που με τη σειρά του είναι ο πυρήνας του *iCanCloud*, 6) *Network Cloudsim* [20] επεκτείνει τα χαρακτηριστικά του *Cloudsim* καθώς υποστηρίζει επικοινωνία μεταξύ των στοιχείων των εφαρμογών και διάφορων δικτυακών στοιχείων, 7) *EMUSIM* [21] δεν είναι απλά ένας προσομοιωτής; Παρέχει παράλληλα προσομοίωση και μίμηση μιας εφαρμογής νέφους. Σχεδιάστηκε για SaaS εφαρμογές έχοντας εντατική επεξεργαστική ισχύ και οι οποίες είναι πολύ δαπανηρές χρηματικά για πραγματική ανάπτυξη, 8) *Groudsim* [22] είναι μια ξεχωριστή πλατφόρμα προσομοίωσης για την υπολογιστική νέφους και δικτύων και σχεδιάστηκε για τη προσομοίωση επιστημονικών εφαρμογών, 9) *DCsim* [23] προσομοιωτής βασικά αναπτύχθηκε για προσομοίωση εικονικών κέντρων δεδομένων που αναπτύχθηκαν στο IaaS μοντέλο, έχει πολλαπλούς διασυνδεδεμένους εξυπηρετητές και ο κάθε εξυπηρετητής έχει το δικό του προγραμματιστή CPU και πολιτική διαχείρισης πόρων 10) *MR-CloudSim* [24] ο εν λόγω προσομοιωτής αποτελεί επέκταση του *Cloudsim*. Ο *cloudsim* προσομοιωτής δεν διαθέτει δυνατότητα για επεξεργασία αρχείων, καθώς το κόστος και ο χρόνος σχετίζονται με αυτό. Στον *MR-cloudsim* έγιναν κάποιες αλλαγές σε μερικές κλάσεις του *cloudsim* καθώς το *map reduce* εφαρμόστηκε στην αρχιτεκτονική. Αυτός ο προσομοιωτής είναι απαραίτητος για εργασία σχετική με επεξεργασία δεδομένων που χρησιμοποιούν το πρωτόκολλο *map reduce*, 11) Ο προσομοιωτής *Smartsim* [25] δημιουργήθηκε αποκλειστικά για το mobile cloud computing και είναι ο μόνος προσομοιωτής που υποστηρίζει προσομοίωση διάφορων εφαρμογών του mobile cloud computing. Έχει χαρακτηριστικά για μοντελοποίηση εφαρμογών νέφους για κινητά τηλέφωνα, 12) Τέλος ο προσομοιωτής *SimIC* [26] χρησιμοποιείται για προσομοίωση διάφορων inter-cloud δραστηριοτήτων. Πρόκειται για έναν ιδιαίτερο προσομοιωτή που είναι σχεδιασμένος πάνω στο *SimJava* πακέτο. Ο *SimIC* χρησιμοποιεί τον αλγόριθμο ICMS για inter-cloud προγραμματισμό από τον οποίο εξαρτώνται το μεγαλύτερο μέρος των κατανεμημένων παραμέτρων. Επίσης δίνει τη λειτουργία as-pay-as-you-go.

Η παρούσα μεταπτυχιακή διατριβή έχει ως στόχο την ανάλυση μερικών από τους πιο πρόσφατους και δημοφιλής προσομοιωτές υπολογιστικής νέφους (*Cloudsim* [1], *Cloudsurf*[4], *Cloudreports* [3], *iCanCloud*[5,19], *Cloudanalyst*[3,6]). Θα γίνει καταγραφή των χαρακτηριστικών και των δυνατοτήτων τους καθώς και η σκοπιμότητα του κάθε προσομοιωτή. Στη συνέχεια θα γίνουν προσομοιώσεις (πειράματα) ξεχωριστά για τον κάθε προσομοιωτή σύμφωνα με κάποιες προδιαμορφωμένες έρευνες (surveys) ώστε να γίνει σύγκριση και επιβεβαίωση των πιθανών ιδίων αποτελεσμάτων που θα παραχθούν. Τέλος θα γραφτούν εγχειρίδια χρήσης για τον κάθε προσομοιωτή ώστε να κατανοηθεί η δυνατότητα εγκατάστασης και χρήσης τους.

2 Προσομοιωτής Cloudsim

Το cloudsim [1] είναι ένα εργαλείο προσομοίωσης σε java με δυνατότητα επέκτασης που επιτρέπει τη μοντελοποίηση και τη προσομοίωση συστημάτων υπολογιστικού νέφους και περιβαλλόντων παροχής εφαρμογών. Το cloudsim υποστηρίζει τόσο συστήματα όσο και τη συμπεριφορά μοντελοποίησης στοιχείων υπολογιστικού νέφους όπως data centers, virtual machines, και πολιτικές παροχής πόρων. Εφαρμόζει γενικές τεχνικές παροχής εφαρμογών που μπορούν να επεκταθούν με εύκολη και περιορισμένη προσπάθεια. Υποστηρίζει τη μοντελοποίηση και τη προσομοίωση των περιβαλλόντων υπολογιστικής νέφους που αποτελείται τόσο από μονά όσο και από διαδικτυακά νέφη (federation clouds) . Επιπλέον εκθέτει προσαρμοσμένες διεπαφές για την εφαρμογή των πολιτικών και παρέχει τεχνικές για τη κατανομή των εικονικών μηχανών κάτω από διαδικτυακά σενάρια υπολογιστικής νέφους. Αρκετοί ερευνητές από οργανισμούς όπως τα εργαστήρια HP των ΗΠΑ, χρησιμοποιούν το cloudsim στην έρευνα παροχής πόρων νέφους και αποτελεσματικής διαχείρισης πόρων κέντρων δεδομένων. Η χρησιμότητα του cloudsim επιδεικνύεται από μια μελέτη περιπτώσεων συμπεριλαμβάνοντας δυναμική παροχή υπηρεσιών εφαρμογών στα υβριδικά περιβάλλοντα νέφους.

Τα κύρια πλεονεκτήματα της χρήσης του cloudsim περιλαμβάνουν: i) *αποτελεσματικότητα χρόνου*: απαιτεί πολύ λιγότερη προσπάθεια και χρόνο να εφαρμόσει cloud-based εφαρμογές παρέχοντας έλεγχο περιβάλλοντος καθώς και ii) *προσαρμοστικότητα & δυνατότητα εφαρμογής*: οι developers μπορούν να μοντελοποιήσουν και να τεστάρουν την απόδοση των υπηρεσιών των εφαρμογών τους σε ετερογενή περιβάλλοντα νέφους (Amazon EC2, Microsoft Azure) με λίγο προγραμματισμό και προσπάθεια ανάπτυξης. Το κύριο μειονέκτημα του cloudsim είναι η έλλειψη γραφικού περιβάλλοντος χρήστη (GUI). Παρόλα αυτά χρησιμοποιείται ακόμα σε πανεπιστήμια και βιομηχανίες για προσομοίωση αλγορίθμων βασισμένων στο νέφος.

2.1 Αρχιτεκτονική του Cloudsim

Η αρχικές κυκλοφορίες του Cloudsim χρησιμοποιούσαν το Simjava σαν ξεχωριστό συμβάν προσομοίωσης μηχανής που υποστηρίζει αρκετές κύριες λειτουργίες, όπως επεξεργασία συμβάντων, δημιουργία οντοτήτων συστημάτων νέφους (services, host, data center, broker, VMs) , επικοινωνία μεταξύ των συστατικών μερών, και διαχείριση ρολογιού προσομοίωσης. Στην παρούσα κυκλοφορία του cloudsim, το επίπεδο Simjava έχει μετακινηθεί για να επιτραπούν μερικές προηγμένες λειτουργίες που δεν υποστηρίζονται από αυτό. Το επίπεδο προσομοίωσης του cloudsim παρέχει υποστήριξη για μοντελοποίηση και προσομοίωση εικονικών βασισμένων στο νέφος περιβαλλόντων κέντρων δεδομένων περιλαμβάνοντας ειδικές διεπαφές διαχείρισης για εικονικών μηχανών, μνήμη, αποθηκευτικό χώρο και εύρος ζώνης. Τα βασικά θέματα όπως παροχή εξυπηρετητών σε εικονικές μηχανές, διαχείριση εκτέλεσης εφαρμογής και παρακολούθηση κατάσταση δυναμικών συστημάτων, διαχειρίζονται από αυτό το επίπεδο. Ένας πάροχος νέφους, που θέλει να μελετήσει την αποτελεσματικότητα διαφορετικών πολιτικών ως προς την παροχή εξυπηρέτησης στα εικονικά μηχανήματα, χρειάζεται να εφαρμόσει τις στρατηγικές του σε αυτό το επίπεδο. Τέτοια εφαρμογή μπορεί να γίνει επεκτείνοντας προγραμματιστικά τη κεντρική λειτουργικότητα τροφοδοσίας σε εικονικά μηχανήματα. Υπάρχει μια σαφής διαφορά σε αυτό το επίπεδο που σχετίζεται με την παροχή εξυπηρέτησης σε εικονικά μηχανήματα. Αυτό το επίπεδο επίσης δείχνει τις λειτουργίες που ένας προγραμματιστής εφαρμογών νέφους μπορεί να επεκτείνει για να εκτελέσει πολύπλοκα φορτία και μελέτη απόδοσης εφαρμογών. Το κορυφαίο επίπεδο στη στοίβα του cloudsim είναι το *User-code layer* που δείχνει τις βασικές οντότητες για τους εξυπηρευτές (αριθμός εικονικών μηχανών, τις προδιαγραφές τους, κτλ) εφαρμογές (αριθμός εργασιών και οι απαιτήσεις τους), εικονικά μηχανήματα, αριθμό χρηστών και τύποι εφαρμογών τους και πολιτικές προγραμματισμού broker. Επεκτείνοντας τις βασικές οντότητες σε αυτό το επίπεδο ένας προγραμματιστής εφαρμογών νέφους μπορεί να εκτελέσει τις παρακάτω δραστηριότητες: 1) Να παράγει ένα σύνολο αιτημάτων κατανομής φόρτου, και ρυθμίσεων παραμέτρων εφαρμογών ,2) Να μοντελοποιεί σενάρια διαθεσιμότητας νέφους και να εκτελεί ισχυρά τεστ βασισμένα σε πολύπλοκες ρυθμίσεις παραμέτρων ,3) εκτέλεση πολύπλοκων εφαρμογών παρέχοντας τεχνικές για νέφη.

2.2 Επεκτάσεις του Cloudsim

Το cloudsim [27] από μόνο του δεν διαθέτει γραφικό περιβάλλον, επεκτάσεις όπως το Cloud Reports προσφέρουν GUI (Graphic User Interface) για Cloudsim προσομοιωτές. Επιπλέον το CloudsimEx [27] επεκτείνει το CloudSim προσθέτοντας map reduce δυνατότητες προσομοίωσης καθώς και παράλληλες προσομοιώσεις. Τέλος το Cloud2Sim [27] επεκτείνει το CloudSim για να εκτελείται σε πολλαπλούς καταναμημένους server, αξιοποιώντας το Hazelcast το οποίο είναι ένα καταναμημένο framework εκτέλεσης.

2.3 Χαρακτηριστικά & Δυνατότητες

Το Cloudsim [28] υποστηρίζει τη μοντελοποίηση και τη προσομοίωση μεγάλης κλίμακας υπολογιστικά περιβάλλοντα νέφους περιλαμβάνοντας κέντρα δεδομένων, πάνω σε ένα μονό υπολογιστικό κόμβο. Παρέχει μια αυτοδύναμη πλατφόρμα για τη μοντελοποίηση των νεφών, κανονισμό υπηρεσιών, παροχή και καταμερισμό πολιτικών. Υποστηρίζει τη προσομοίωση συνδέσεων δικτύου μεταξύ των προσομοιωμένων τμημάτων του συστήματος. Διαθέτει ικανότητα προσομοίωσης ενωμένων (federated) περιβαλλόντων νέφους. Παρέχει διαθεσιμότητα εικονικής μηχανής που βοηθά στη δημιουργία και διαχείριση πολλαπλών ανεξάρτητων και παράλληλα φιλοξενούμενων εικονικών υπηρεσιών σε ένα κομβικό κέντρο δεδομένων. Παρέχει ευελιξία εναλλαγής μεταξύ κοινόχρηστου χώρου και κατανομής κοινόχρηστου χρόνου επεξεργασίας πυρήνων σε εικονικές υπηρεσίες.

2.4 Πειράματα και αποτελέσματα

Προκειμένου να κατανοήσουμε τη λειτουργία του Cloudsim διεξήγαμε μια σειρά από πειράματα [2]. Η εκτέλεση των πειραμάτων έγινε στο Eclipse IDE προκειμένου να εφαρμόσουμε τον κώδικα μας σε java. Σε κάθε πείραμα δώσαμε διαφορετικούς παραμέτρους και εξάγαμε διαφορετικά αποτελέσματα. Από τα πειράματα μας φαίνεται ο «κύκλος ζωής» του προσομοιωτή μας. Πιο συγκεκριμένα οι φάσεις από την έναρξη έως τη λήξη είναι οι εξής: Έναρξη → Αρχικοποίηση πακέτου cloudsim → Δημιουργία κέντρων δεδομένων → Δημιουργία broker¹ → Δημιουργία εικονικών μηχανών → Δημιουργία cloudlets² → Έναρξη προσομοίωσης → Τέλος προσομοίωσης → Εξαγωγή αποτελεσμάτων → Τέλος.

2.4.1 Εκτέλεση 1ης προσομοίωσης

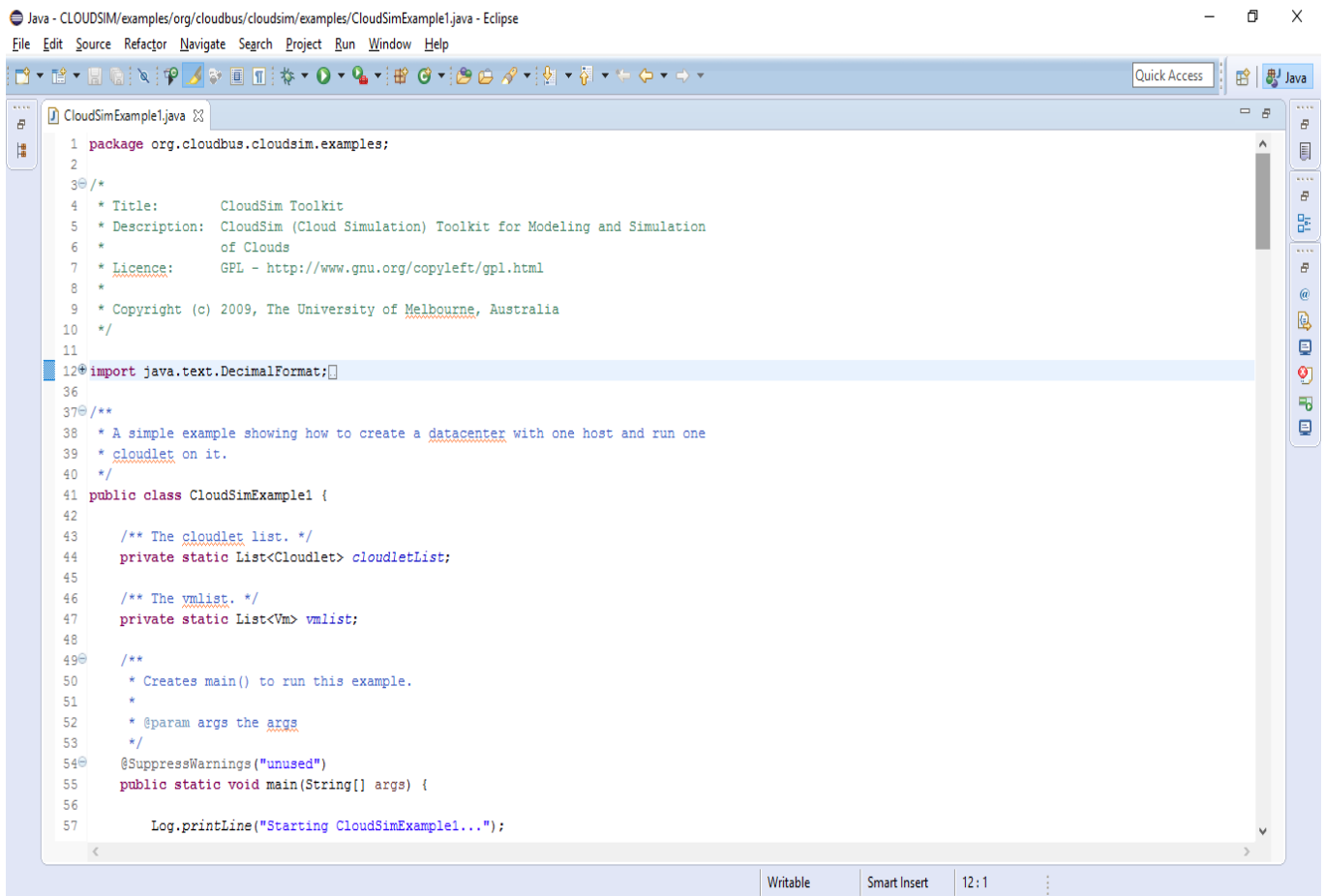
Για να γίνει μια εκτίμηση της απόδοσης του νέφους, έγινε προσομοίωση σε λειτουργικό σύστημα windows 7 basic (64-bit), με επεξεργαστή I3, 2.40GHZ ταχύτητα και μνήμη 3GB. Για τη δοκιμή της 1^{ης} προσομοίωσης χρειάστηκε:

- 1 data center
- 1 host
- 1 cloudlet

Η εικόνα 1 μας δείχνει τον κώδικα στο cloudsim εκτελεσμένο σε java καθώς και τα αντίστοιχα αποτελέσματα που προκύπτουν φαίνονται στην εικόνα 2.

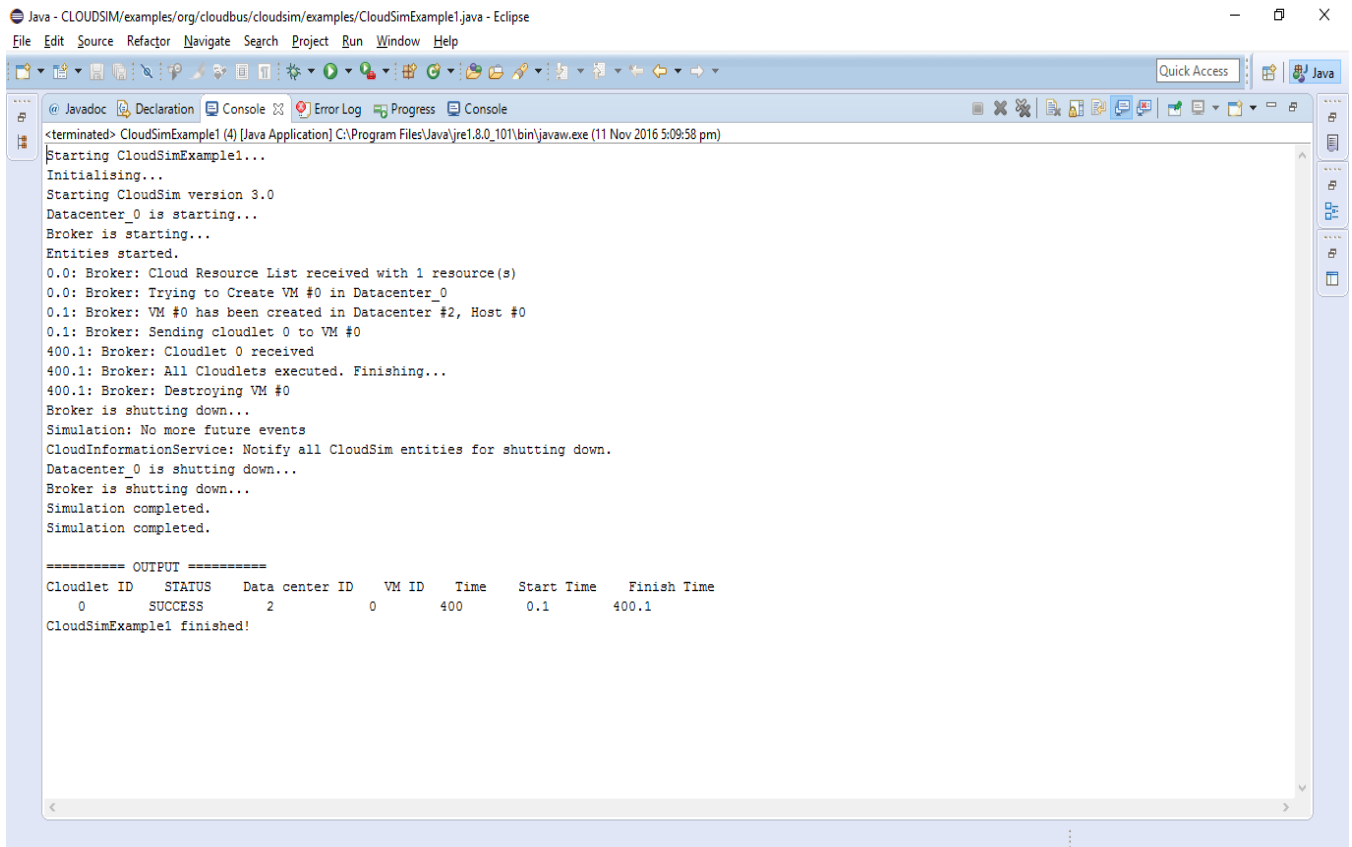
¹ Πρόκειται για μια οντότητα που διαχειρίζεται τη χρήση, απόδοση και παράδοση των υπηρεσιών νέφους και διαπραγματεύεται τις σχέσεις μεταξύ των παρόχων και καταναλωτών της υπολογιστικής νέφους.

² Πρόκειται για ένα αρχιτεκτονικό στοιχείο που προκύπτει από τη σύγκλιση του mobile computing (IoT) και του cloud computing. Αναπαριστά το μεσαίο στοιχείο της ιεραρχίας της τριστρωματικής αρχιτεκτονικής mobile or IoT device --- cloudlet --- cloud. Ο στόχος του είναι να φέρει το cloud πιο κοντά.



```
1 package org.cloudbus.cloudsim.examples;
2
3 /**
4  * Title:      CloudSim Toolkit
5  * Description: CloudSim (Cloud Simulation) Toolkit for Modeling and Simulation
6  *             of Clouds
7  * Licence:    GPL - http://www.gnu.org/copyleft/gpl.html
8  *
9  * Copyright (c) 2009, The University of Melbourne, Australia
10 */
11
12 import java.text.DecimalFormat;
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37 /**
38  * A simple example showing how to create a datacenter with one host and run one
39  * cloudlet on it.
40  */
41 public class CloudSimExample1 {
42
43     /** The cloudlet list. */
44     private static List<Cloudlet> cloudletList;
45
46     /** The vm list. */
47     private static List<Vm> vmList;
48
49     /**
50      * Creates main() to run this example.
51      *
52      * @param args the args
53      */
54     @SuppressWarnings("unused")
55     public static void main(String[] args) {
56
57         Log.println("Starting CloudSimExample1...");
58     }
59 }
```

Εικόνα 1. Κώδικας προσομοίωσης με δοκιμή 1 datacenter με 1 host και εκτέλεση 1 cloudlet



```
Java - CLOUDSIM/examples/org/cloudbus/cloudsim/examples/CloudSimExample1.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java
@ Javadoc Declaration Console Error Log Progress Console
<terminated> CloudSimExample1 (4) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (11 Nov 2016 5:09:58 pm)
Starting CloudSimExample1...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID   STATUS   Data center ID   VM ID   Time   Start Time   Finish Time
0            SUCCESS   2                0       400    0.1          400.1
CloudSimExample1 finished!
```

Εικόνα 2.Αποτελέσματα 1^{ης} προσομοίωσης

2.4.2 Εκτέλεση 2ης προσομοίωσης

Στη συνέχεια έγινε και μια δεύτερη προσομοίωση στο ίδιο data center με τις εξής αλλαγές:

- 1 datacenter
- 2 hosts
- 2 cloudlets

Τα cloudlets εκτελούνται σε VM με διαφορετικά MIPS (million instructions per second). Τα cloudlets θα πάρουν διαφορετικούς χρόνους για να ολοκληρώσουν την εκτέλεση βασισόμενη στην απόδοση του VM. Οι εικόνες 3 & 4, μας δείχνουν τον κώδικα καθώς και τα αποτελέσματα που παράγονται.

```

Java - CLOUDSIM/examples/org/cloudbus/cloudsim/examples/CloudSimExample3.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java
CloudSimExample3.java
114     long fileSize = 300;
115     long outputSize = 300;
116     UtilizationModel utilizationModel = new UtilizationModelFull();
117
118     Cloudlet cloudlet1 = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel, utilizationModel, utilizationModel);
119     cloudlet1.setUserId(brokerId);
120
121     id++;
122     Cloudlet cloudlet2 = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel, utilizationModel, utilizationModel);
123     cloudlet2.setUserId(brokerId);
124
125     //add the cloudlets to the list
126     cloudletList.add(cloudlet1);
127     cloudletList.add(cloudlet2);
128
129     //submit cloudlet list to the broker
130     broker.submitCloudletList(cloudletList);
131
132
133     //bind the cloudlets to the vms. This way, the broker
134     // will submit the bound cloudlets only to the specific VM
135     broker.bindCloudletToVm(cloudlet1.getCloudletId(), vm1.getId());
136     broker.bindCloudletToVm(cloudlet2.getCloudletId(), vm2.getId());
137
138     // Sixth step: Starts the simulation
139     CloudSim.startSimulation();
140
141
142     // Final step: Print results when simulation is over
143     List<Cloudlet> newList = broker.getCloudletReceivedList();
144
145     CloudSim.stopSimulation();
146
147     printCloudletList(newList);
    
```

Εικόνα 3.Κώδικας προσομοίωσης με δοκιμή 1 datacenter με 2 hosts και εκτέλεση 2 cloudlets

```

Java - CLOUDSIM/examples/org/cloudbus/cloudsim/examples/CloudSimExample3.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java
@ Javadoc Declaration Console Error Log Progress Console
<terminated> CloudSimExample3 (1) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (12 Nov 2016 1:25:43 pm)
Starting CloudSimExample3...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #1
0.1: Broker: Sending cloudlet 0 to VM #0
0.1: Broker: Sending cloudlet 1 to VM #1
80.1: Broker: Cloudlet 1 received
160.1: Broker: Cloudlet 0 received
160.1: Broker: All Cloudlets executed. Finishing...
160.1: Broker: Destroying VM #0
160.1: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID   STATUS   Data center ID   VM ID   Time   Start Time   Finish Time
1            SUCCESS   2                1       80     0.1          80.1
0            SUCCESS   2                0       160    0.1          160.1
CloudSimExample3 finished!
    
```

Εικόνα 4.Αποτελέσματα 2ης προσομοίωσης

2.4.2.1 Συμπεράσματα 1ης και 2ης προσομοίωσης

Σε αυτές τις προσομοιώσεις παρατηρήσαμε διαφορετικά αποτελέσματα εκτελώντας διαφορετικούς κώδικες στους οποίους πήραμε πληροφορίες σχετικά με τα cloudlets, status, datacenter ID, VM ID, start time, finish time. Αλλάζοντας τον αριθμό των host των datacenter και των cloudlets παρατηρήσαμε τη διαφορά. Στην 1^η προσομοίωση ο broker δημιούργησε 1 VM στο data_center και χρειάστηκε περισσότερο χρόνο προκειμένου να εκτελεστεί 1 cloudlet σε αυτό (400 MIPS). Στη 2^η προσομοίωση δημιουργήθηκαν 2 VM στο data_center και ο χρόνος που χρειάστηκε για την εκτέλεση των 2 cloudlets σε 2 VMs είναι 80 και 160 MIPS αντίστοιχα σε κάθε VM.

2.4.3 Εκτέλεση 3ης προσομοίωσης

Στην επόμενη προσομοίωση προσαρμόσαμε το δικό μας προγραμματιστικό αλγόριθμο στις ανάλογες τοποθεσίες των ήδη υπαρχών τμημάτων κώδικα. Για τη προσομοίωση εφαρμόστηκε το πρόγραμμα Eclipse IDE όπου έγιναν οι ανάλογες διαμορφώσεις στον κώδικα του cloudsim. Ο αλγόριθμος που θα εφαρμόσουμε είναι ο **Shortest Job First**³ και ο σκοπός μας είναι με βάση τον αλγόριθμο να εκτελεστούν μια σειρά από cloudlets. Τα cloudlets που έχουν μικρότερο αριθμό εντολών (length) εκτελούνται πρώτα. Με αποτέλεσμα ένα VM που διαθέτει ένα συγκεκριμένο αριθμό από MIPS θα χρειαστεί λιγότερο χρόνο για να εκτελεστεί ένα cloudlet με μικρό αριθμό εντολών. Ο χρόνος εκτέλεσης προκύπτει διαιρώντας το μέγεθος εντολών του Cloudlet / αριθμό MIPS *εικονικού μηχανήματος* όπως φαίνεται και στον παρακάτω τύπο **Execution time= length(cloudlet)/Mips(Vm)**.

Τα κομμάτια του κώδικα του cloudsim που διαμορφώσαμε ήταν τα εξής αρχεία:

- ✓ CloudsimExample6.java
- ✓ DatacenterBroker.java

2.4.3.1 Κώδικας εκτέλεσης 3ης προσομοίωσης

Ξεκινώντας από το αρχείο **CloudsimExample6.java** εφαρμόζουμε την εξής τεχνική:

Line 134: Αλλάζουμε τον αριθμό των VM που δημιουργούνται και αποστέλλονται στον broker από 20 σε 1 από τη λίστα των VM.

```
131 //Fourth step: Create VMs and Cloudlets and send them to broker
132 vmlist = createVM(brokerId,20); //creating 20 vms
133 cloudletList = createCloudlet(brokerId,40); // creating 40 cloudlets
```

```
131 //Fourth step: Create VMs and Cloudlets and send them to broker
132 vmlist = createVM(brokerId,1); //creating 20 vms
133 cloudletList = createCloudlet(brokerId,40); // creating 40 cloudlets
```

Lines 93-94: Δημιουργούμε τη κλάση random (η οποία δηλώνεται αρχικά) για τη παραγωγή τυχαίων αριθμών (ID) cloudlet. Στη συνέχεια προσθέτουμε στη παράμετρο length +rObj.nextInt(2000). Αυτή η μέθοδος χρησιμοποιείται για να πάρει μια ψευδοτυχαία ομοιόμορφα κατανομημένη ακέραια τιμή συμπεριλαμβανομένου

³ Πρόκειται για ένα αλγόριθμο χρονοπρογραμματισμού της KME με βάση τη μικρότερη διάρκεια εκτέλεσης μιας διεργασίας και βασίζεται στη διάρκεια του επόμενου ξεσπάσματος μιας διεργασίας. Η διεργασία με το μικρότερο χρόνο ξεσπάσματος εκτελείται πρώτη. Σε περίπτωση που δύο διεργασίες έχουν ίδιο χρόνο ξεσπάσματος επιλέγεται ο αλγόριθμος με βάση σειρά άφιξης (FCFS).

του 0 και τη συγκεκριμένη τιμή που επιλέγεται από αυτή τη τυχαία συχνότητα παραγωγής αριθμών. Ο αριθμός 2000 είναι το όριο για το τυχαίο αριθμό που θα επιστραφεί ο οποίος πρέπει να είναι θετικός.

```

81
82 //cloudlet parameters
83 long length = 1000;
84 long fileSize = 300;
85 long outputSize = 300;
86 int pesNumber = 1;
87 UtilizationModel utilizationModel = new UtilizationModelFull();
88
89 Cloudlet[] cloudlet = new Cloudlet[cloudlets];
90
91 for(int i=0;i<cloudlets;i++){
92     cloudlet[i] = new Cloudlet(i, length, pesNumber, fileSize, outputSize, utilizationModel, utilizationModel);
93     // setting the owner of these Cloudlets
94     cloudlet[i].setUserId(userId);
95     list.add(cloudlet[i]);
96 }
97
98
99
100

```

```

83 //cloudlet parameters
84 long length = 1000;
85 long fileSize = 300;
86 long outputSize = 300;
87 int pesNumber = 1;
88 UtilizationModel utilizationModel = new UtilizationModelFull();
89
90 Cloudlet[] cloudlet = new Cloudlet[cloudlets];
91
92 for(int i=0;i<cloudlets;i++){
93     Random rObj = new Random();
94     cloudlet[i] = new Cloudlet(i, (length + rObj.nextInt(2000)), pesNumber, fileSize, outputSize, utilizationModel, utilizationModel);
95     // setting the owner of these Cloudlets
96     cloudlet[i].setUserId(userId);
97     list.add(cloudlet[i]);
98 }
99
100 return list;

```

Line 68: Τέλος αλλάζουμε τη πολιτική προγραμματισμού των VM από **timeshared** σε **Spaceshared** προκειμένου να γίνει προγραμματισμός πολλαπλών νημάτων ταυτόχρονα σε πολλούς πυρήνες.

```

67 for(int i=0;i<vms;i++){
68     vm[i] = new Vm(i, userId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared());
69     //for creating a VM with a space shared scheduling policy for cloudlets:
70     //vm[i] = Vm(i, userId, mips, pesNumber, ram, bw, size, priority, vmm, new CloudletSchedulerSpaceShared());

```

```

67 for(int i=0;i<vms;i++){
68     vm[i] = new Vm(i, userId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerSpaceShared());
69     //for creating a VM with a space shared scheduling policy for cloudlets:
70     //vm[i] = Vm(i, userId, mips, pesNumber, ram, bw, size, priority, vmm, new CloudletSchedulerSpaceShared());

```

Στη συνέχεια πηγαίνουμε στο επόμενο αρχείο **DatacenterBroker.java** για να κάνουμε τις ανάλογες παραμετροποιήσεις.

```

335     /**
336     * Submit cloudlets to the created VMs.
337     *
338     * @pre $none
339     * @post $none
340     */
341     protected void submitCloudlets() {
342         int vmIndex = 0;
343
344
345
346         for (Cloudlet cloudlet : getCloudletList()) {
347             Vm vm;
348             // if user didn't bind this cloudlet and it has not been executed yet
349             if (cloudlet.getVmId() == -1) {
350                 vm = getVmsCreatedList().get(vmIndex);
351             } else { // submit to the specific vm
352                 vm = VmList.getById(getVmsCreatedList(), cloudlet.getVmId());
353                 if (vm == null) { // vm was not created

```

Line:341: Πηγαίνουμε στη protected μέθοδο `submitCloudlets()` όπου προσθέτουμε τις παρακάτω γραμμές κώδικα.

List<Cloudlet> sortList = new ArrayList<Cloudlet>(); // Δημιουργία μιας ταξινομημένης λίστας sortList όπου θα είναι περιέχει συχνότητα των cloudlet. Επειδή η λίστα είναι ένα interface δημιουργούμε την class ArrayList.

ArrayList<Cloudlet> tempList = new ArrayList<Cloudlet>(); //Δημιουργία tempList.

for(Cloudlet cloudlet : getCloudletList()) { tempList.add(cloudlet); // Προσθήκη cloudlet από τη CloudletList στη tempList.

}

int totalCloudlets = tempList.size (); // προσθήκη του συνολικού μεγέθους των cloudlets στην tempList

for(int i=0;i<totalCloudlets;i++) //Μέτρηση του συνολικού αριθμού cloudlets

{

Cloudlet smallestCloudlet = tempList.get(0); //Εισαγωγή του cloudlet μικρότερου μεγέθους στη tempList

for(Cloudlet checkCloudlet: tempList) //Πρόκειται να ελέγξουμε τα cloudlet στην templist

{

if(smallestCloudlet.getCloudletLength()>checkCloudlet.getCloudletLength())

//Συνθήκη για εύρεση του cloudlet με το μικρότερο length

{

smallestCloudlet=checkCloudlet; // Επιλογή του μικρότερου Cloudlet

}

```

    }

    sortList.add(smallestCloudlet); // προσθήκη του μικρότερου cloudlet στη sortList.
    tempList.remove(smallestCloudlet); // διαγραφή του μικρότερου cloudlet από τη
tempList και τοποθέτηση στη sortList.
}

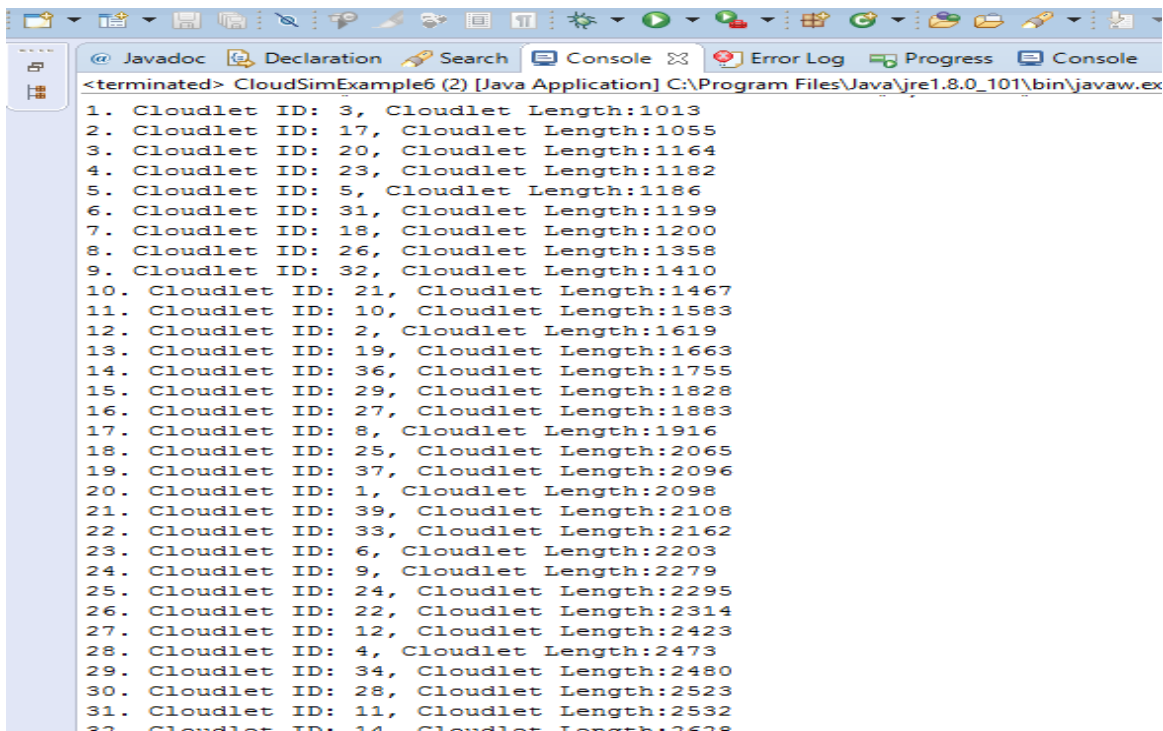
int count=1; //αύξων μετρητής cloudlet με αρχικοποίηση στο 1.
for(Cloudlet printCloudlet : sortList) //έλεγχος αν η λίστα είναι ταξινομημένη ή όχι.
{
    Log.println(count + ". Cloudlet ID: " + printCloudlet.getCloudletId() + ", Cloudlet
Length:" + printCloudlet.getCloudletLength()); //Εκτύπωση στην οθόνη τα αποτελέσματα από
count, Cloudlet ID, Cloudlet Length.

    count++; //Αύξηση μετρητή.
} for (Cloudlet cloudlet : sortList) {

```

2.4.4 Συμπεράσματα και αποτελέσματα

Όπως παρατηρούμε από τους παρακάτω πίνακες το κομμάτι του κώδικα που προσθέσαμε εμφανίζει τα παρακάτω αποτελέσματα, καθώς ο broker παράγει 40 cloudlets με τυχαία σειρά και σύμφωνα με τον αλγόριθμο που χρησιμοποιήσαμε SJF εμφανίζει το Length από το μικρότερο προς το μεγαλύτερο.

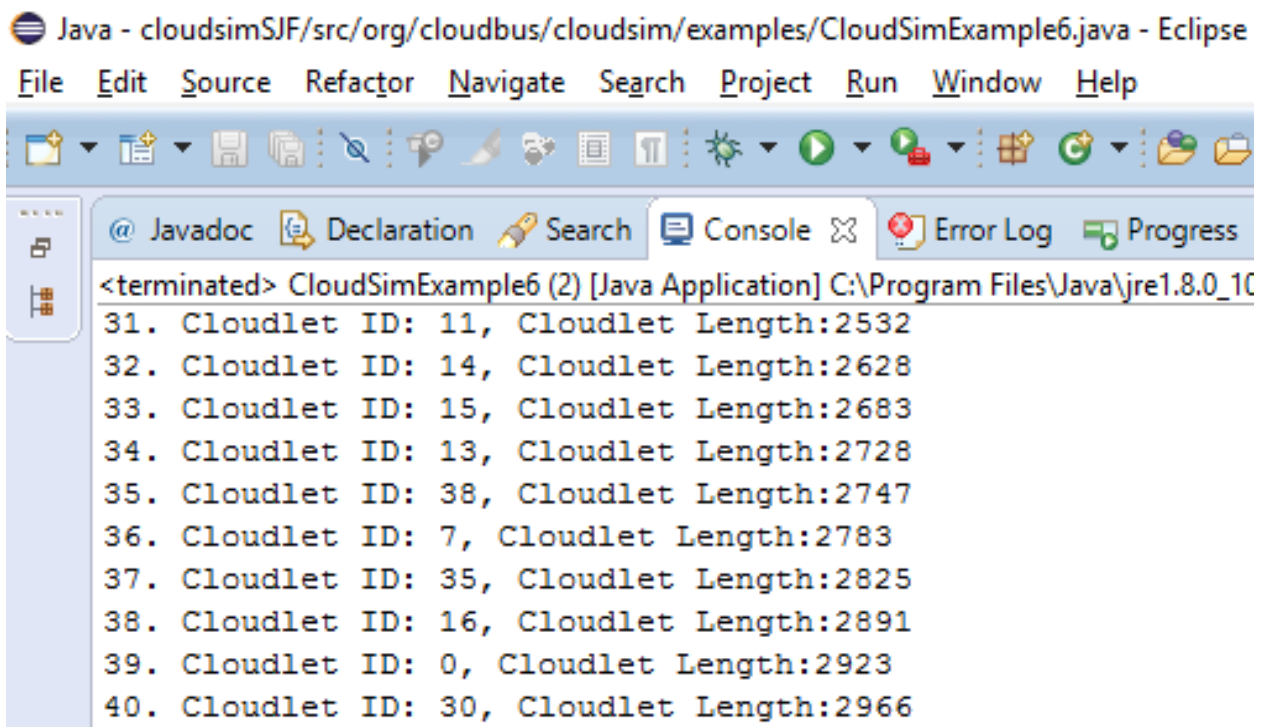


```

<terminated> CloudSimExample6 (2) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe
1. Cloudlet ID: 3, Cloudlet Length:1013
2. Cloudlet ID: 17, Cloudlet Length:1055
3. Cloudlet ID: 20, Cloudlet Length:1164
4. Cloudlet ID: 23, Cloudlet Length:1182
5. Cloudlet ID: 5, Cloudlet Length:1186
6. Cloudlet ID: 31, Cloudlet Length:1199
7. Cloudlet ID: 18, Cloudlet Length:1200
8. Cloudlet ID: 26, Cloudlet Length:1358
9. Cloudlet ID: 32, Cloudlet Length:1410
10. Cloudlet ID: 21, Cloudlet Length:1467
11. Cloudlet ID: 10, Cloudlet Length:1583
12. Cloudlet ID: 2, Cloudlet Length:1619
13. Cloudlet ID: 19, Cloudlet Length:1663
14. Cloudlet ID: 36, Cloudlet Length:1755
15. Cloudlet ID: 29, Cloudlet Length:1828
16. Cloudlet ID: 27, Cloudlet Length:1883
17. Cloudlet ID: 8, Cloudlet Length:1916
18. Cloudlet ID: 25, Cloudlet Length:2065
19. Cloudlet ID: 37, Cloudlet Length:2096
20. Cloudlet ID: 1, Cloudlet Length:2098
21. Cloudlet ID: 39, Cloudlet Length:2108
22. Cloudlet ID: 33, Cloudlet Length:2162
23. Cloudlet ID: 6, Cloudlet Length:2203
24. Cloudlet ID: 9, Cloudlet Length:2279
25. Cloudlet ID: 24, Cloudlet Length:2295
26. Cloudlet ID: 22, Cloudlet Length:2314
27. Cloudlet ID: 12, Cloudlet Length:2423
28. Cloudlet ID: 4, Cloudlet Length:2473
29. Cloudlet ID: 34, Cloudlet Length:2480
30. Cloudlet ID: 28, Cloudlet Length:2523
31. Cloudlet ID: 11, Cloudlet Length:2532
32. Cloudlet ID: 14, Cloudlet Length:2628

```

Εικόνα 5.Αποτελέσματα 3^{ης} προσομοίωσης



The screenshot shows the Eclipse IDE interface. The title bar reads "Java - cloudsimsJF/src/org/cloudbus/cloudsim/examples/CloudSimExample6.java - Eclipse". The menu bar includes "File", "Edit", "Source", "Refactor", "Navigate", "Search", "Project", "Run", "Window", and "Help". The toolbar contains various icons for file operations, search, and execution. The console window is active, displaying the following output:

```
<terminated> CloudSimExample6 (2) [Java Application] C:\Program Files\Java\jre1.8.0_10
31. Cloudlet ID: 11, Cloudlet Length:2532
32. Cloudlet ID: 14, Cloudlet Length:2628
33. Cloudlet ID: 15, Cloudlet Length:2683
34. Cloudlet ID: 13, Cloudlet Length:2728
35. Cloudlet ID: 38, Cloudlet Length:2747
36. Cloudlet ID: 7, Cloudlet Length:2783
37. Cloudlet ID: 35, Cloudlet Length:2825
38. Cloudlet ID: 16, Cloudlet Length:2891
39. Cloudlet ID: 0, Cloudlet Length:2923
40. Cloudlet ID: 30, Cloudlet Length:2966
```

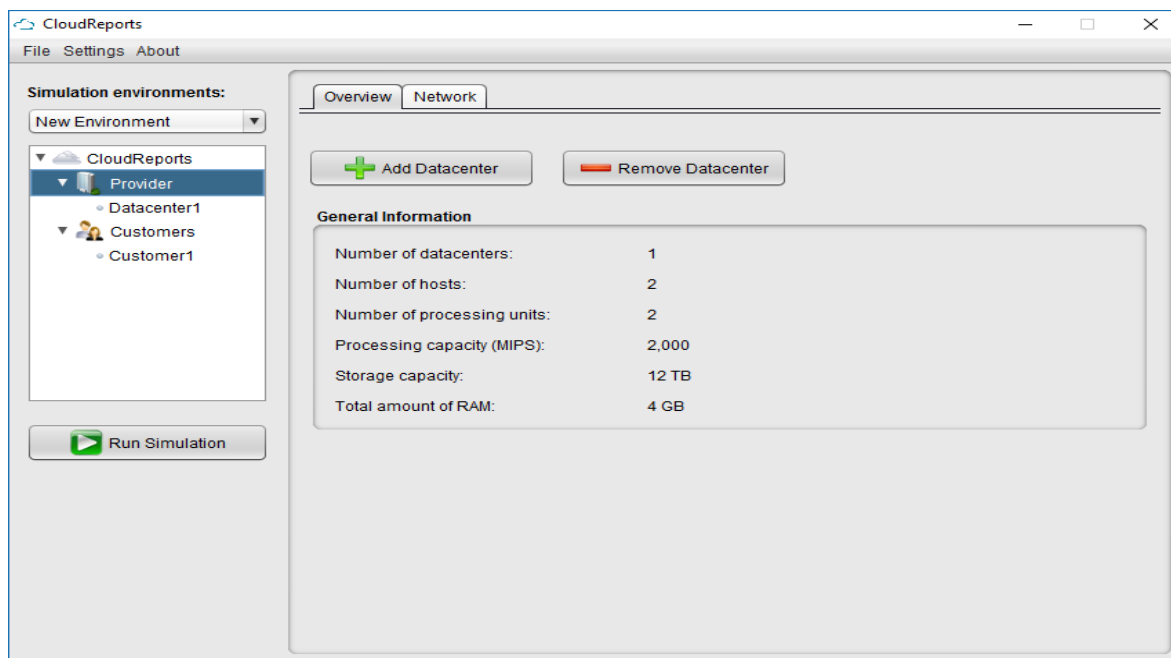
Εικόνα 6.Αποτελέσματα 3ης προσομοίωσης

3 Προσομοιωτής Cloudreports

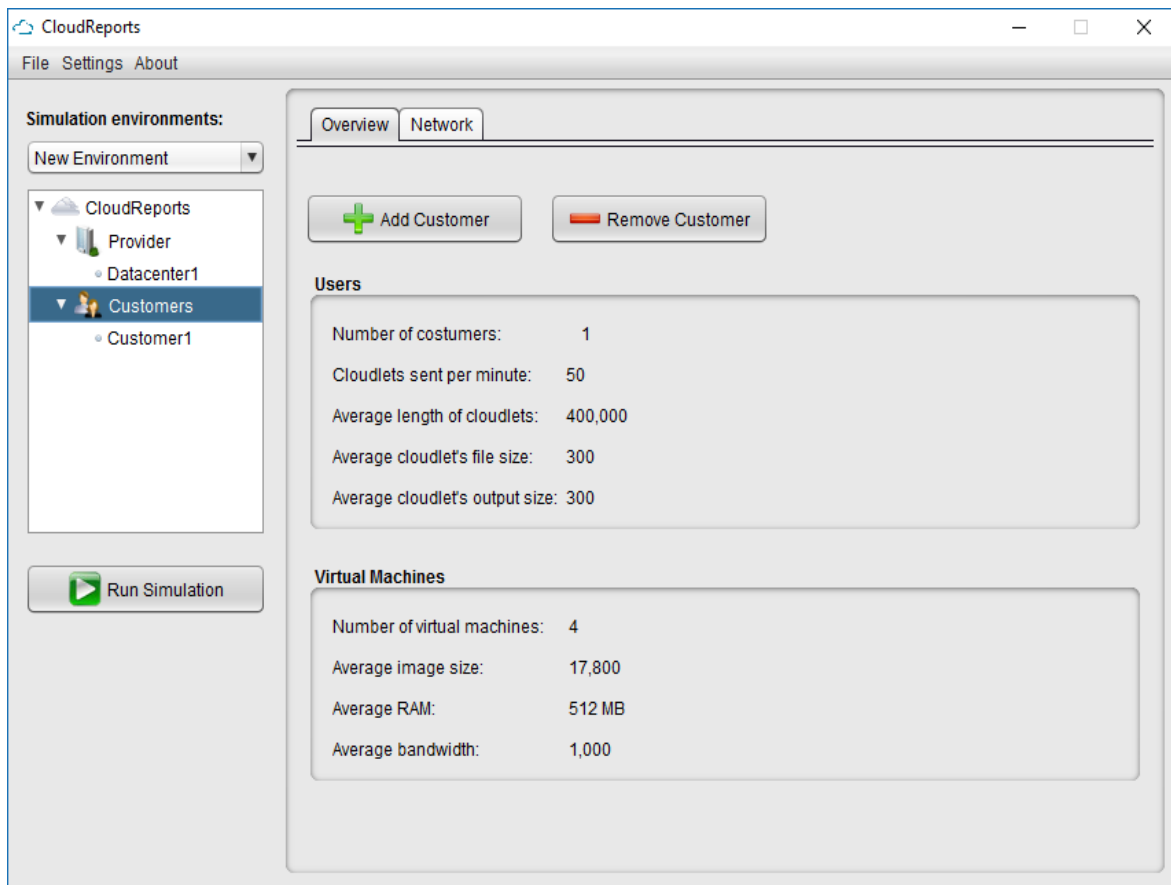
Το cloudreports [3] αποτελεί ένα εργαλείο επέκτασης του cloudsim βασιζόμενο στην αρχιτεκτονική του. Βελτιώνει τα χαρακτηριστικά του cloudsim παρέχοντας ένα γραφικό περιβάλλον προς το χρήστη, τρέχοντας πολλαπλές προσομοιώσεις την ίδια χρονική στιγμή και βελτιώνει τα αποτελέσματα της προσομοίωσης. Επιπλέον το cloudreports ωφελείται από τη χρήση του API (application programmable interface) για τη δημιουργία ειδικών επεκτάσεων για προσομοίωση ειδικών αλγορίθμων. Το cloudreports [7] απλοποιεί την δημιουργία και τη διαχείριση των περιβαλλόντων προσομοίωσης τα οποία μπορούν χειραγωγηθούν και να σωθούν για μελλοντική χρήση. Οι ερευνητές μπορούν να δημιουργήσουν πολλαπλά κέντρα δεδομένων με διαφορετική ποσότητα πόρων και να διαμορφώσουν κάθε ένα από τους hosts τους ατομικά. Επιπλέον η συμπεριφορά του client μπορεί να προσαρμοστεί ρυθμίζοντας τη ποσότητα των εικονικών μηχανών που θα αναπτυχθούν και να προσδιορίσει τα cloudlets που θα “τρέξουν” σε αυτές. Οι πόροι που απαιτούνται από κάθε εικονικό μηχάνημα είναι εντελώς προσαρμόσιμοι. Το cloudreports επιτρέπει τις προσομοιώσεις να εκτελούνται σε ομάδες, το οποίο σημαίνει ότι οι ερευνητές μπορούν να προσδιορίζουν πόσες υλοποιήσεις πρέπει να εκτελεστούν και η ποσότητα του χρόνου που θα προσομοιωθεί. Μετά την ολοκλήρωση των προσομοιώσεων το εργαλείο παράγει μια πλήρη αναφορά που αποτελείται από ένα αρχείο καταγραφής λειτουργιών και αρκετά διαγράμματα με λεπτομερής πληροφορίες που σχετίζονται με τη χρήση των πόρων, κατανομή εικονικών μηχανών, εκτέλεση cloudlet καθώς και κατανάλωση ενέργειας κέντρου δεδομένων. Επιπλέον προσθετικά αρχεία δημιουργούνται για να επιτρέψουν στα τελικά δεδομένα να εξαχθούν σε εφαρμογές όπως το matlab και octave.

3.1 Πειράματα και αποτελέσματα

Για την εκτέλεση του πειράματος όπως φαίνεται στις εικόνες 7 & 8 υποθέσαμε ότι το περιβάλλον προσομοίωσης αποτελείται από 1 datacenter, και 2 hosts. Κάθε host διαθέτει 1 CPU με υπολογιστική ισχύ 1000 MIPS, 2GB ram καθώς και 1 TB αποθηκευτικό χώρο. Σε αυτή τη προσομοίωση έχουμε 4 VM's που τρέχουν στο host με ισχύ 500,1000,500,1000 MIPS αντίστοιχα. Τα VM διαθέτουν 512 ram καθώς και 10 GB αποθηκευτικό χώρο.

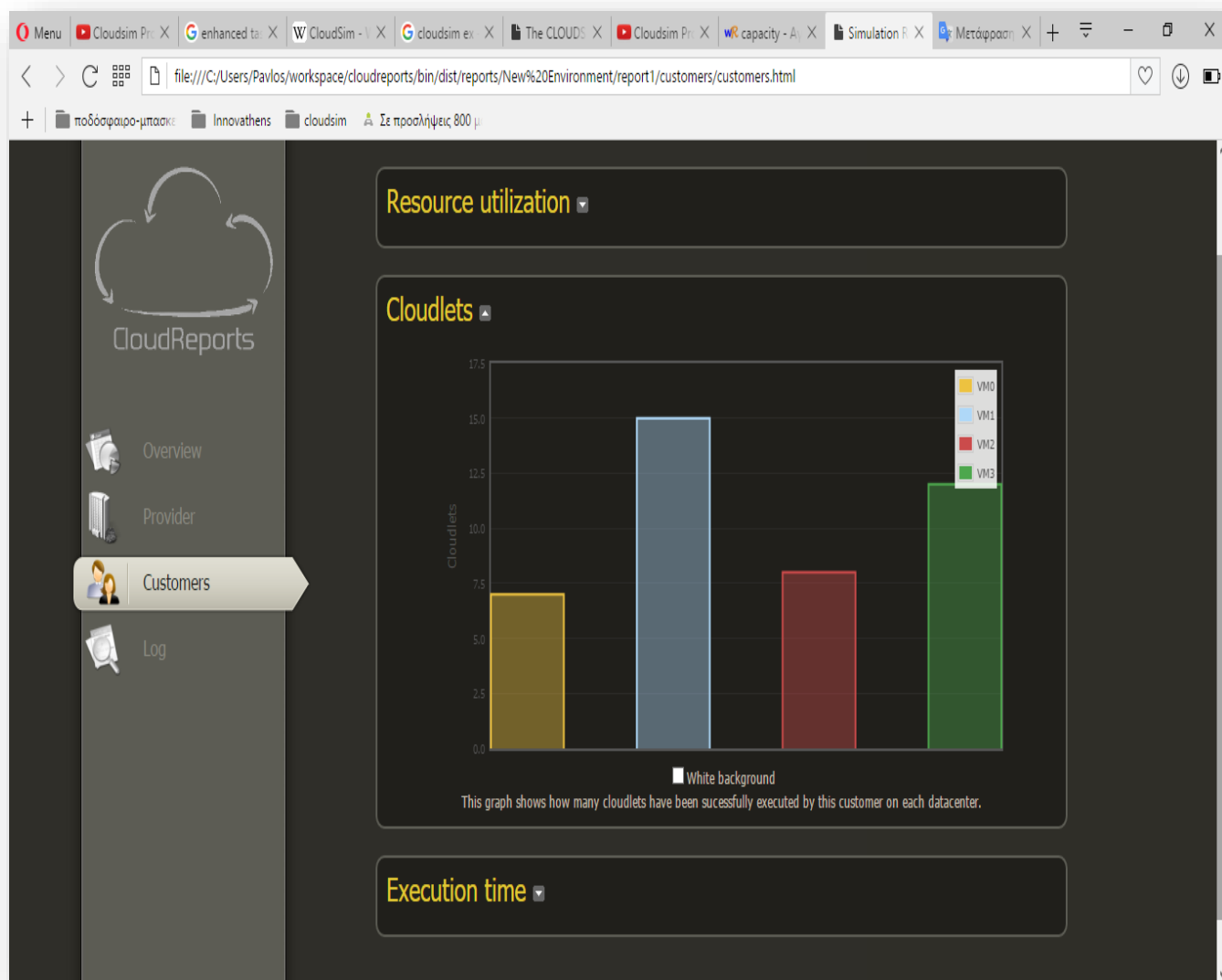


Εικόνα 7. Ρύθμιση παραμέτρων του παρόχου

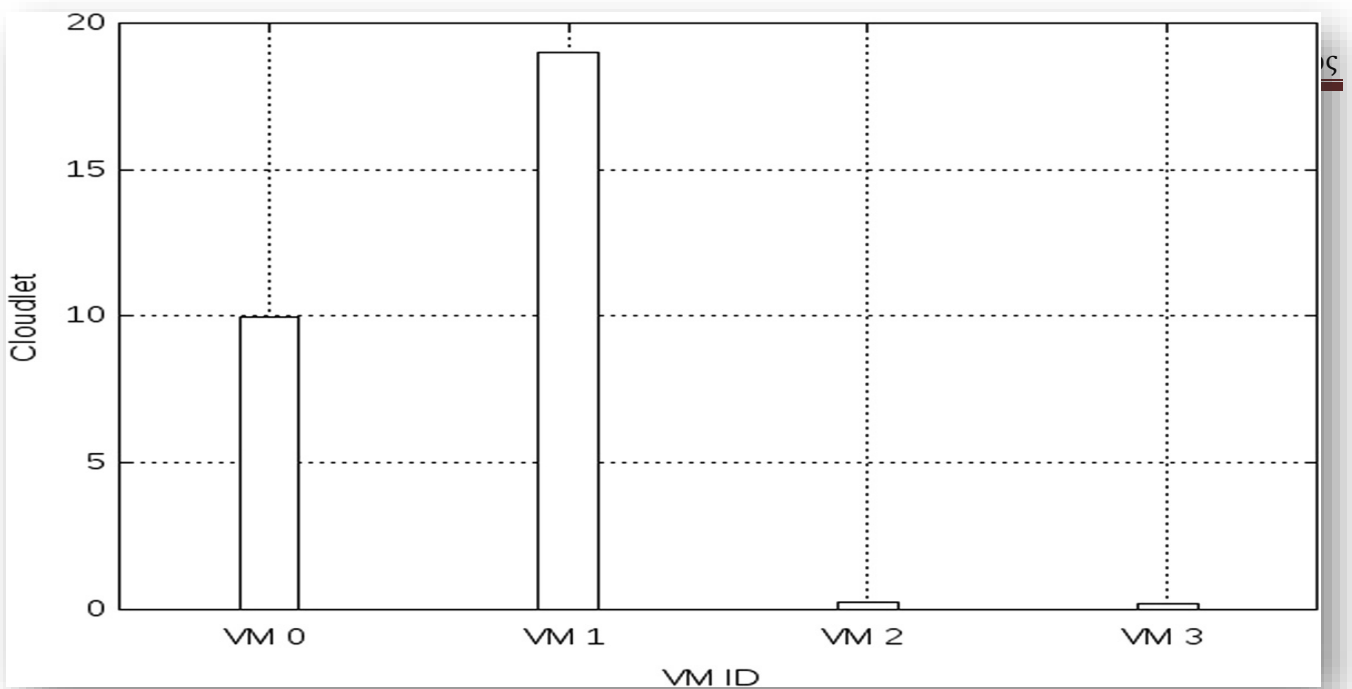


Εικόνα 8. Ρύθμιση παραμέτρων του πελάτη

Οι εικόνες 9 και 10 μας δείχνουν συγκριτικά αποτελέσματα [3] σε μορφή γραφήματος της δική μας προσομοίωσης (εικ.9) με μια ήδη υπάρχουσα (εικ.10) . Οι γραφικές παραστάσεις μας δείχνουν τον αριθμό των cloudlets που εκτελέστηκαν επιτυχώς σε κάθε VM από τον συγκεκριμένο customer στο υπάρχων datacenter. Τα αποτελέσματα παρουσιάζουν κάποιες διαφορές. Στο 1^ο VM εκτελέστηκαν σε κάθε πείραμα επιτυχώς 7 και 10 cloudlets αντίστοιχα. Στο 2^ο VM εκτελέστηκαν επιτυχώς 15 και 19 cloudlets αντίστοιχα. Στο 3^ο VM εκτελέστηκαν επιτυχώς 8 και 1 cloudlets αντίστοιχα. Στο 4^ο VM εκτελέστηκαν επιτυχώς 12 και 1 cloudlets αντίστοιχα.

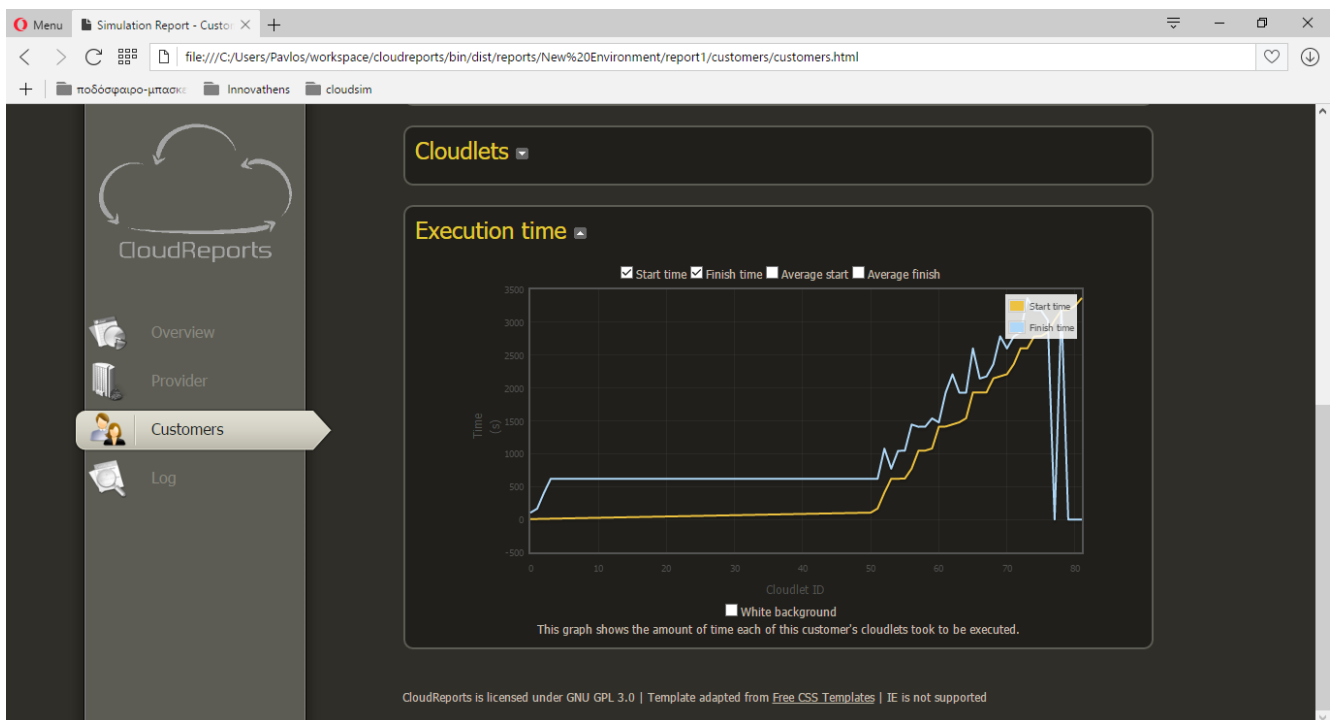


Εικόνα 9. Αριθμός Cloudlets που εκτελέστηκαν επιτυχώς από τον πελάτη σε κάθε κέντρο δεδομένων

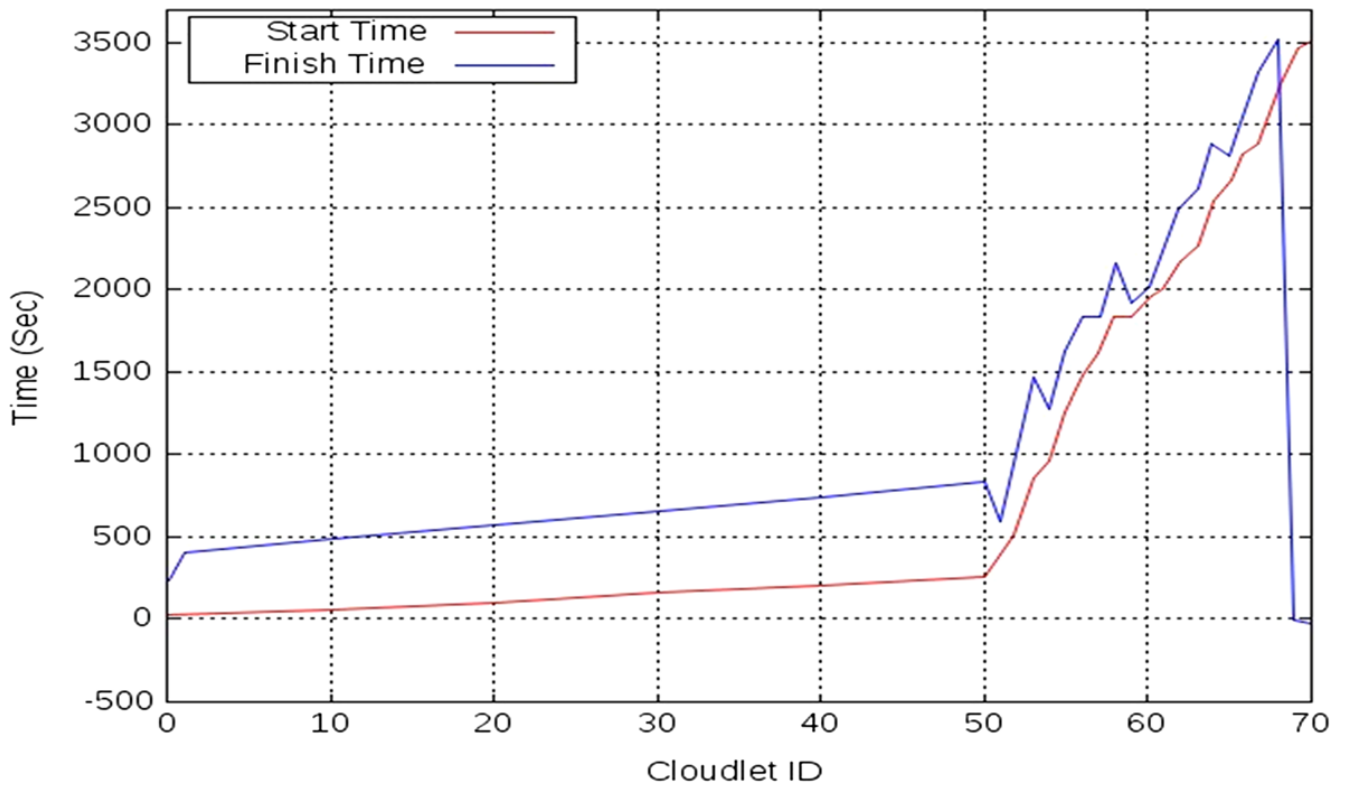


Εικόνα 10. Αριθμός Cloudlets που εκτελέστηκαν επιτυχώς από τον πελάτη σε κάθε κέντρο δεδομένων [3]

Στη συνέχεια οι εικόνες 11 & 12 μας δείχνουν το συνολικό χρόνο σε δευτερόλεπτα για το κάθε cloudlet που εκτελέστηκε επιτυχώς στο υπάρχον data center.



Εικόνα 11. Ποσότητα χρόνου εκτέλεσης των cloudlet κάθε πελάτη



Εικόνα 12. Ποσότητα χρόνου εκτέλεσης των cloudlet κάθε πελάτη [3]

3.2 Συμπεράσματα

Αυτό που παρατηρούμε είναι ότι οι χρονικές διακυμάνσεις και στα δύο διαγράμματα είναι σχεδόν ίδιες όσον αφορά το χρόνο εκτέλεσης των cloudlets, με τη μόνη διαφορά ότι στη δική μας προσομοίωση (εικόνα 11) χρειάστηκε περισσότερος χρόνος για να εκτελεστούν επιτυχώς περισσότερα cloudlets. Οι προσομοιώσεις διαφέρουν σε πολύ λίγα σημεία και αυτό οφείλεται στο χρόνο εκτέλεσης που δαπανά κάθε προσομοίωση.

4 Προσομοιωτής Cloudsurf

Το cloudsurf [4] είναι μια πλατφόρμα ανοιχτού κώδικα που επιτρέπει τη παρακολούθηση δικτυακών υποδομών δημόσιου-νέφους από την οπτική γωνία του πελάτη μέσω μη-συνεργαζόμενων προσεγγίσεων, δηλαδή χωρίς να βασίζεται σε πληροφορίες περιορισμένες στο παροχέα νέφους, ή σε οντότητες παίζοντας προνομιακό ρόλο με σεβασμό στη παροχή υπηρεσιών του νέφους. Ο λόγος της δημιουργίας του cloudsurf ήταν ότι οι πελάτες έπρεπε να αντιμετωπίζουν μια σειρά από περιορισμούς. Για παράδειγμα δεν είχαν επίγνωση πως οι δικτυακοί πόροι κατανέμονταν στις διαθέσιμες υπηρεσίες του νέφους ή δεν είχαν πληροφορίες σχετικά πως η απόδοση του δικτύου μπορεί να μεταβληθεί σταδιακά. Βεβαίως δεν ήταν ικανοί είτε να κάνουν συνειδητές επιλογές μεταξύ διαφορετικών υπηρεσιών είτε να συγκρίνουν διάφορους παρόχους. Μέσω της πλατφόρμας του cloudsurf ξεπεράστηκαν αυτοί οι περιορισμοί. Επιπλέον το cloudsurf επιτρέπει τον εύκολο διαμοιρασμό των αποτελεσμάτων παρακολούθησης μεταξύ των χρηστών, επιπλέον οδηγεί στην αποθήκευση του κόστους των πειραματισμών των χρηστών.

4.1 Χαρακτηριστικά & δυνατότητες

Μια πλατφόρμα σαν το cloudsurf έχει κάποια ιδιαίτερα χαρακτηριστικά. Αυτά είναι τα εξής:

1. Αποδοχή των μη συνεργάσιμων προσεγγίσεων παρακολούθησης

Καθώς η πλατφόρμα είχε εκτιμηθεί να είναι πλήρως προσαρμοσμένη στο πελάτη, οι προσεγγίσεις που εφαρμόζονται πρέπει να απαιτούν καμία συνεργασία οποιασδήποτε προνομιακής οντότητας. Δεδομένου ότι οι δραστηριότητες παρακολούθησης δεν βασίζονται σε αυτή τη πληροφορία, θα πρέπει κυρίως να αξιοποιηθούν ενεργές προσεγγίσεις παρακολούθησης.

2. Άνεση στη χρήση

Οι πελάτες που επωφελούνται από τις υπηρεσίες του νέφους έχουν διαφορετικά υπόβαθρα και ενδεχομένως να μην έχουν μια προηγμένη πείρα σε δραστηριότητες παρακολούθησης του δικτύου και σχετικών ζητημάτων.

3. Ευρύτητα

Καθώς οι πάροχοι προσφέρουν ένα ταχύτατα εξελισσόμενο εύρος υπηρεσιών με διαφορετικά χαρακτηριστικά και ιδιότητες, είναι καλό για τη πλατφόρμα να αναλαμβάνει τους πιο δημοφιλείς παρόχους, καθώς όλες οι επιλογές είναι διαθέσιμες για κάθε ένα από αυτούς. Επιπροσθέτως οι πελάτες μπορεί να ενδιαφέρονται για διαφορετικές ιδιότητες των δικτύων (π.χ. το ελάχιστο εγγυημένο εύρος, τη μέγιστη εφικτή διεκπεραίωση, αδράνεια). Συνεπώς η πλατφόρμα επιτρέπει στους πελάτες να εκτελέσουν τον πειραματισμό που είναι κατάλληλος για συλλογή των πτυχών του ενδιαφέροντος.

4. Προβλεψιμότητα κόστους πειραματισμού

Μη συνεργατικές προσεγγίσεις απαιτούν αλληλεπίδραση με τον πάροχο όπως κάνουν οι πελάτες. Αυτό συνεπάγεται ότι ο χρήστης της πλατφόρμας υποβάλλεται στο παράδειγμα του “pay-as-you-go”, σε συμφωνία με τους όρους συμβολαίου με τον κάθε πάροχο. Οι πειραματισμοί του δικτυακού νέφους θα μπορούσε να είναι πολύ ακριβοί (ειδικά όταν χρειάζονται επαναλαμβανόμενες αναλύσεις ή όταν η κίνηση αναπαράγεται με υψηλούς ρυθμούς) καθώς οι πάροχοι συνήθως χρεώνουν τους πελάτες όχι μόνο για υπολογισμό ή δυνατότητες μνήμης των εικονικών μηχανών(VM), αλλά επίσης για τη κίνηση που παράγεται από αυτούς. Είναι καλό το γεγονός ότι η πλατφόρμα μπορεί να είναι ικανή να εκτιμήσει το κόστος που οι πελάτες υποβάλλονται.

5. Εύκολος διαμοιρασμός αποτελεσμάτων ανάλυσης

Σύμφωνα επίσης με το κόστος των πειραματισμών, θα ήταν επιθυμητό ότι η πλατφόρμα θα παρείχε ένα εύκολο τρόπο να διαμοιράσει το αποτέλεσμα των αναλύσεων με τους ανθρώπους που ενδιαφέρεται.

4.2 Λειτουργίες

Το cloudsurf επιτρέπει την παρακολούθηση της απόδοσης του δικτύου νέφους στα συστατικά του μέρη:

1. Ενδοδουκτιακό κέντρο δεδομένων (Intra-datacenter network)

Αυτή η λειτουργία επιτρέπει στο χρήστη να συγκρίνει τη δικτυακή σύνδεση που αναπτύσσεται σε δύο εικονικά μηχανήματα που βρίσκονται στην ίδια γεωγραφική περιοχή.

2. Διαδικτυακό κέντρο δεδομένων (Inter-datacenter network)

Αυτή η λειτουργία επιτρέπει στο χρήστη να συγκρίνει τη δικτυακή σύνδεση που αναπτύσσεται σε δύο εικονικά μηχανήματα που βρίσκονται σε διαφορετικές γεωγραφικές περιοχές.

3. Δίκτυο χρήστη-νέφους (User-to-cloud network)

Αυτή η λειτουργία επιτρέπει τη συγκριτική αξιολόγηση των επιδόσεων του συνόλου των διαδρομών που διασύνδεουν το cloud των καταναλωτών στο κέντρο δεδομένων.

4.3 Πειράματα και αποτελέσματα

Το cloudsurf διαθέτει μια σειρά από προδιαμορφωμένα πειράματα:

1. Διεκπεραίωση και αδράνεια στο TCP (TCP throughput and latency)

Αυτό το πείραμα στοχεύει στην εύρεση της μέγιστης απόδοσης του TCP και της αντιληπτής αδράνειας παράγοντας υψηλού ρυθμού(1 Gbps) συνθετική TCP κίνηση.

2. Διεκπεραίωση UDP (UDP throughput)

Αυτό το πείραμα στοχεύει στην εύρεση της μέγιστης απόδοσης του UDP παράγοντας υψηλού ρυθμού (1 Gbps) συνθετική UDP κίνηση.

3. Αδράνεια ICMP (ICMP latency)

Αυτό το πείραμα στοχεύει στην εύρεση της μέγιστης αντιληπτής αδράνειας παράγοντας χαμηλού ρυθμού (10 pps) ICMP κίνησης.

5 Προσομοιωτής iCanCloud

Η αυξανόμενη πολυπλοκότητα των υπολογιστικών συστημάτων έκανε τους προσομοιωτές μια σημαντική επιλογή για σχεδιασμό και ανάλυση μεγάλων και πολύπλοκων αρχιτεκτονικών. Το iCanCloud [29] είναι μια πλατφόρμα προσομοίωσης που έχει ως στόχο τη μοντελοποίηση και τη προσομοίωση των συστημάτων υπολογιστικής νέφους, το οποίο στοχεύει σε εκείνους τους χρήστες συνεργάζονται στενά με τέτοιου είδους συστήματα. Το κύριο αντικείμενο του iCanCloud είναι να προβλέπει τις δοσοληψίες μεταξύ του κόστους και της απόδοσης μιας σειράς από εφαρμογές εκτελεσμένες σε ένα ειδικό hardware, και έπειτα να παρέχει στους χρήστες χρήσιμες πληροφορίες για τέτοια κόστη. Επομένως το iCanCloud μπορεί να χρησιμοποιηθεί από ένα μεγάλο αριθμό χρηστών, από βασικούς ενεργούς χρήστες μέχρι στους προγραμματιστές μεγάλων καταναμημένων εφαρμογών.

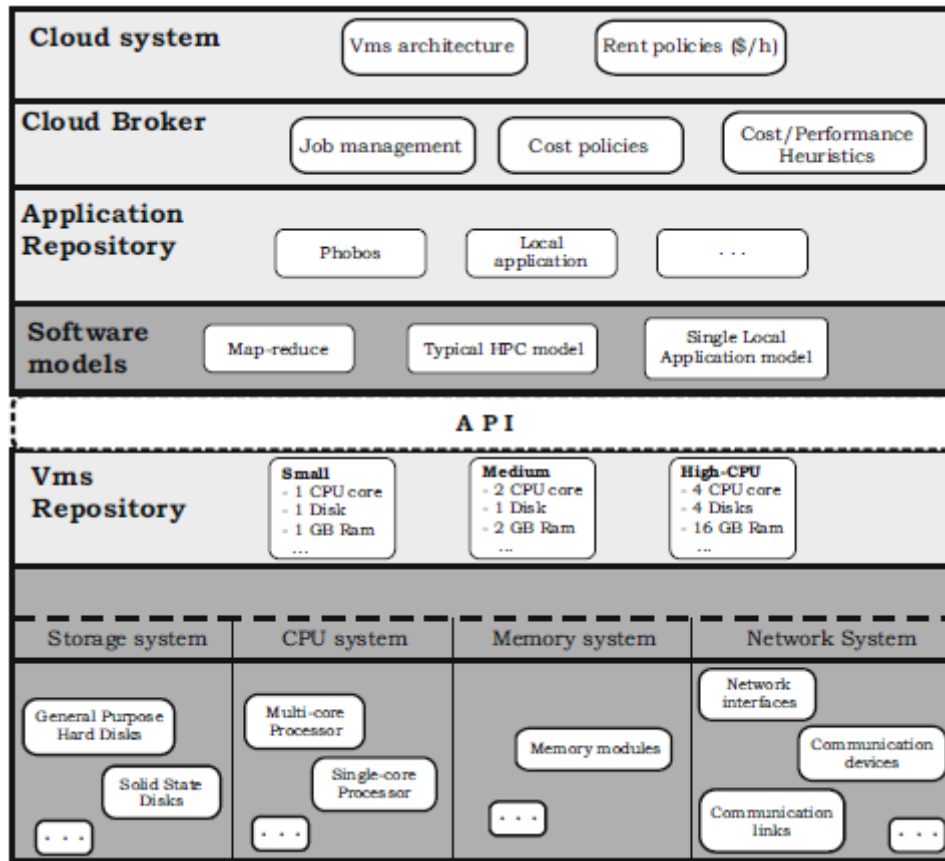
Το iCanCloud [5] framework παρέχει ένα ευέλικτο, γρήγορο και εύκολο στη χρήση εργαλείο που επιτρέπει στους χρήστες να αποκτούν αποτελέσματα γρήγορα για να τους βοηθήσει να πάρουν απόφαση σχετικά με τον αριθμό και τον τύπο των μηχανών για να χρησιμοποιήσουν, το οποίο επηρεάζει το προϋπολογισμό του χρήστη. Παρέχει ένα σύνολο από συστατικά που επιτρέπουν τη δημιουργία σεναρίων υπολογιστικής νέφους. Αυτά τα συστατικά αναπαριστούν τη συμπεριφορά των πραγματικών συστατικών που ανήκουν σε πραγματικές αρχιτεκτονικές όπως δίσκοι, δίκτυα, μνήμες, συστήματα αρχείων κτλ. Συνεπώς αυτά τα συστατικά είναι ιεραρχικά οργανωμένα στον αποθηκευτικό χώρο του iCanCloud τα οποία φτιάχνουν τη κεντρική μηχανή προσομοίωσης. Επιπλέον το iCanCloud επιτρέπει την εύκολη αλλαγή των συστατικών για ένα συγκεκριμένο χαρακτηριστικό (πχ διαφορετικοί δικτυακοί αντάπτορες μπορούν να αντικατασταθούν εύκολα).

5.1 Χαρακτηριστικά και δυνατότητες

Τα πιο ιδιαίτερα χαρακτηριστικά της πλατφόρμας προσομοίωσης iCanCloud είναι τα εξής: Υπάρχουσες και μη αρχιτεκτονικές υπολογιστικής νέφους έχουν δυνατότητα να μοντελοποιηθούν και να προσομοιωθούν. Το iCanCloud παρέχει μεθόδους για την απόκτηση της ενεργειακής κατανάλωσης κάθε δομικού στοιχείου του hardware στα συστήματα υπολογιστικής νέφους. Η μνήμη, ο αποθηκευτικός χώρος και τα υποσυστήματα του δικτύου μπορούν να μοντελοποιηθούν για να προσομοιώσουν ένα μεγάλο εύρος από αληθινά συστήματα. Το σύστημα του δικτύου μπορεί να μοντελοποιηθεί για να προσομοιώσει ένα μεγάλο εύρος από καταναμημένα περιβάλλοντα με υψηλό βαθμό λεπτομερειών. Οι χρήστες είναι ικανοί να σχεδιάσουν και να μοντελοποιήσουν πολιτικές παροχής πόρων για συστήματα νέφους με σκοπό να ισορροπήσουν τις δοσοληψίες μεταξύ απόδοσης και κατανάλωσης ενέργειας. Από τότε που η ενεργειακή κατανάλωση στα μεγάλα καταναμημένα συστήματα είναι απευθείας σχετιζόμενη με τη διαχείριση των πόρων, είναι σημαντικό να επιτρέψει στους χρήστες να προσαρμόσουν τις πολιτικές τους για να αναλύσουν την επίδραση της κατανάλωσης ενέργειας στη συνολική απόδοση του συστήματος. Προσαρμοσίμα VM μπορούν να χρησιμοποιηθούν για τη γρήγορη προσομοίωση μονό- πύρινων και πολύ-πύρινων συστημάτων. Το iCanCloud παρέχει ένα μεγάλο εύρος διαμορφώσεων για τα συστήματα αποθήκευσης, οι οποίες περιλαμβάνουν μοντελοποιήσεις για τοπικά και απομακρυσμένα συστήματα αποθήκευσης, όπως NFS, και παράλληλα συστήματα αποθήκευσης όπως παράλληλα συστήματα αρχείων και RAID συστημάτων. Παρέχει ένα φιλικό προς το χρήστη GUI για να διευκολύνει τη δημιουργία και τη προσαρμοστικότητα μεγάλων καταναμημένων μοντέλων. Το GUI είναι ιδιαίτερα χρήσιμο για: διαχείριση αποθηκευτικού χώρου προ-διαμορφωμένων VM's, διαχείριση αποθηκευτικού χώρου προ-διαμορφωμένων συστημάτων νέφους, διαχείριση αποθηκευτικού χώρου προ-διαμορφωμένων πειραμάτων, παρουσιάζοντας πειράματα από το GUI, παράγοντας γραφικές αναφορές. Παρέχει μια POSIX-based εφαρμογή και μια ευπροσάρμοστη MPI βιβλιοθήκη για μοντελοποίηση και προσομοίωση εφαρμογών. Επίσης αρκετοί μέθοδοι για μοντελοποίηση εφαρμογών μπορούν να χρησιμοποιηθούν στο icloud: χρησιμοποίηση πραγματικών εφαρμογών, χρησιμοποίηση γραφημάτων, και προγραμματισμό νέων εφαρμογών απευθείας στη πλατφόρμα προσομοίωσης. Νέα συστατικά μπορούν να προστεθούν στον αποθηκευτικό χώρο του iCanCloud για να αυξήσουν τη λειτουργικότητα της πλατφόρμας προσομοίωσης.

5.2 Αρχιτεκτονική του iCanCloud

Η κύρια ιδέα ενός συστήματος νέφους είναι να παρέχει στους χρήστες μια ψευδο-προσαρμόσιμη υποδομή όπου μπορούν να εκτελέσουν εξειδικευμένο υλικό. Η αρχιτεκτονική του iCanCloud σχεδιάστηκε βασισμένη σε αυτή την αρχή για να μοντελοποιεί πλήρης υποδομές νέφους πάνω από τις οποίες άλλες υπηρεσίες μπορούν να σχεδιαστούν και να αναπτυχθούν, από μονές εφαρμογές (Software as a Service, SaaS) σε μια πλατφόρμα ανάπτυξης (Platform as a Service, PaaS). Η εικόνα 13 μας δείχνει τα επίπεδα αρχιτεκτονικής του iCanCloud.



Εικόνα 13. Επίπεδα αρχιτεκτονικής του iCanCloud [5]

Η πλατφόρμα προσομοίωσης μπορεί να χωριστεί σε δύο διαφορετικά τμήματα. Από τη μια πλευρά το τμήμα που παρέχει συστατικά για μοντελοποίηση και προσομοίωση στοιχείων υλικού και λογισμικού: iCanCloud core engine (dark grey). Από την άλλη πλευρά το τμήμα που περιλαμβάνει τα μοντέλα για διαχείριση του συστήματος νέφους, όπως εικονικές μηχανές και πολιτικές κόστους οι οποίες πρέπει να καθορίζονται από τους χρήστες (light grey). Το κάτω μέρος της αρχιτεκτονικής αποτελείται από το επίπεδο hardware. Αυτό το επίπεδο είναι υπεύθυνο για την μοντελοποίηση των φυσικών τμημάτων του συστήματος όπως δίσκοι, μνήμη, δίκτυα επικοινωνίας και επεξεργαστές. Χρησιμοποιώντας αυτά τα μοντέλα, ολόκληρα κατανεμημένα συστήματα μπορούν να μοντελοποιηθούν και να προσομοιωθούν. Με τη σειρά του αυτό το τμήμα αποτελείται από τέσσερις ομάδες, όπου κάθε μια ανταποκρίνεται σε ένα συγκεκριμένο βασικό σύστημα: αποθηκευτικός χώρος, cpu, μνήμη και συστήματα δικτύου. Το ανώτερο επίπεδο είναι ο αποθηκευτικός χώρος του VM και περιέχει ένα σύνολο από εικονικές μηχανές που έχουν καθοριστεί προηγουμένως από τον χρήστη. Ο προσομοιωτής iCanCloud παρέχει αρκετά μοντέλα εικονικών μηχανών που υπάρχουν σε ένα αρκετά γνωστό δημόσιο σύννεφο από την Amazon. Επιπλέον οι χρήστες μπορούν να προσθέσουν, να επεξεργαστούν η και να διαγράψουν ένα εικονικό μηχάνημα από αυτό τον αποθηκευτικό χώρο. Στο σύστημα νέφους το εικονικό μηχάνημα είναι το πιο σχετικό συστατικό. Με παρόμοιο τρόπο στο iCanCloud ένα εικονικό μηχάνημα είναι ένα δομικό στοιχείο για τη δημιουργία συστημάτων νέφους. Το κλειδί αυτής της πλατφόρμας προσομοίωσης είναι η σπονδυλώτητα της, η οποία επιτρέπει τη δημιουργία σύνθετων μονάδων χρησιμοποιώντας άλλες μονάδες. Για αυτό το λόγο η βασική ιδέα του iCanCloud αποτελείται από τη χρήση των εικονικών μηχανών για τη κατασκευή ολόκληρων υπολογιστικών συστημάτων νέφους. Στο νέφος οι εικονικές μηχανές είναι υπεύθυνες για την απόκρυψη των λεπτομερειών υλικού, παρέχοντας στους χρήστες μια λογική προβολή που αντιστοιχεί με τις απαιτήσεις των χρηστών. Συνεπώς τα μοντέλα των εικονικών μηχανών που ορίζονται σε αυτό το επίπεδο χρησιμοποιούν τα στοιχεία του υλικού που ορίζονται στο κατώτερο στρώμα.

Το επόμενο επίπεδο που ονομάζεται επίπεδο λογισμικού περιλαμβάνει τα μοντέλα εφαρμογών που παρέχονται από την iCanCloud μηχανή. Αυτά τα μοντέλα χρησιμοποιούνται για τη μοντελοποίηση της συμπεριφοράς ενός ευρέου φάσματος εφαρμογών. Η βασική ιδέα είναι να επιτρέπει στους χρήστες να προσαρμόσουν εκείνα τα μοντέλα για τη δημιουργία ειδικών μοντέλων εφαρμογών, τα οποία εκτελούνται σε ειδικά περιβάλλοντα που καθορίζονται από ένα σύνολο εικονικών μηχανών. Στη τωρινή έκδοση του προσομοιωτή υπάρχουν τρία μοντέλα για μοντελοποίηση εφαρμογών. Κάθε ένα από αυτά μπορεί πλήρως να προσαρμοστεί θέτοντας ένα

σύνολο από ειδικούς παραμέτρους από το χρήστη. Επιπλέον νέα μοντέλα εφαρμογών μπορούν εύκολα να προστεθούν στο σύστημα, επειδή το iCanCloud παρέχει ένα API για να διευκολύνει την ανάπτυξη των νέων μοντέλων εφαρμογών. Αυτό το API περιλαμβάνει ένα σύνολο λειτουργιών για χρησιμοποίηση των τεσσάρων συστημάτων που περιγράφηκαν προηγουμένως για το επίπεδο υλικού.

Το επίπεδο αποθηκευτικού χώρου εφαρμογής περιέχει μια συλλογή προκαθορισμένων εφαρμογών προσαρμοσμένη από τους χρήστες. Ομοίως στον αποθηκευτικό χώρο των εικονικών μηχανών. Αρχικά αυτός ο αποθηκευτικός χώρος παρέχει ένα σύνολο από προκαθορισμένα μοντέλα εφαρμογής. Αυτά τα μοντέλα θα χρησιμοποιηθούν για να ρυθμίσουν τις αντίστοιχες εργασίες που θα εκτελεστούν σε συγκεκριμένες περιπτώσεις ενός εικονικού μηχανήματος στο σύστημα.

Το κορυφαίο επίπεδο που ονομάζεται cloud broker, αποτελείται από μια μονάδα που είναι υπεύθυνη για τη διαχείριση όλων των εισερχόμενων εργασιών και τις περιπτώσεις όλων των εικονικών μηχανών όπου τέτοιες εργασίες εκτελούνται. Όταν μια εργασία τελειώνει την εκτέλεση της, αυτή η ενότητα είναι υπεύθυνη για την αποδέσμευση των εικονικών μηχανών που είναι σε αδράνεια και στη συνέχεια εκ νέου ανάθεση των διαθέσιμων πόρων στο σύστημα για να εκτελέσει τις υπόλοιπες εργασίες. Αυτή η ενότητα επίσης περιλαμβάνει πολιτικές κόστους για να αναθέσει εισερχόμενες εργασίες, και έπειτα ανάλογα με τη πολιτική που επιλέγεται, οι εργασίες θα ανατεθούν σε συγκεκριμένο instance που επιλέγεται από τον αντίστοιχο ευρετικό.

Τέλος στην κορυφή της αρχιτεκτονικής είναι το επίπεδο του συστήματος νέφους, το οποίο περιέχει την περιγραφή του συνόλου των εικονικών μηχανών που συνθέτει όλο το σύστημα νέφους καθώς και περιγραφή των πολιτικών κόστους.

5.3 Πειράματα και αποτελέσματα προσομοιωτή iCanCloud

Προκειμένου να διεξάγουμε τη παρακάτω προσομοίωση χρησιμοποιήσαμε λειτουργικό σύστημα Ubuntu linux 14.04 μέσω της εικονικής μηχανής virtual box. Δημιουργήσαμε ένα περιβάλλον υπολογιστικής νέφους με όλα τα απαραίτητα δομικά στοιχεία. Σκοπός της προσομοίωσης μας είναι η εύρεση κατανάλωσης ισχύος και ενέργειας των κέντρων καθώς και των κόμβων.

5.3.1 Καθορισμός παραμέτρων

Στο πείραμα μας δημιουργήσαμε ένα περιβάλλον νέφους καθορίζοντας τις παρακάτω παραμέτρους. Η προσομοίωση έγινε στο iCanCloudGUI για λόγους ευκολίας χρήσης.

<u>CPU</u>	<u>Disk</u>		
Name: AMD_FX_8350_8cores	Name: Maxtor_3TB		
Number of cores: 8	Read bandwidth(MB/s):146.0		
Speed in (MIPS):97125	Write bandwidth(MB/s):112.0		
Independent cores: yes	Number of devices:1		
Tick in seconds: 0.1	Seek time(s):0.0069		
Minimum speed (mips): 92500	Size per device(GB):3000		
Energy parameters:			
<u>Energy state</u>	<u>Value</u>	<u>Energy state</u>	<u>Value</u>

W	59	W	0
Off	0.0	disk_off	0
c0_p11	1.51	disk_IO	8
c0_p10	3.02	disk_active	8
c0_p9	4.53	disk_idle	5.4
c0_p8	6.04		
c0_p7	7.55		
c0_p6	9.068		
c0_p5	10.58		<u>Memory</u>
c0_p4	12.09	Name:Maxtor_3TB	
c0_p3	13.60	Read latency: 6.9E-6	
c0_p2	15.11	Write latency:6.9E-6	
c0_p1	16.62	Search latency:6.9E-6	
c0_p0	24.84	Size(mb):16384	
c0_operating_state	0.0	block size(kb):128	
c1_halt	0.0	Number of DRAM chips:8	
c2_stop_grant	0.0	Number of modules:1	
c3_sleep	0.0	<u>Energy state</u>	<u>Value</u>
c4_deeper_sleep	0.0	A	1.5
c5_enhanced_deeper_sleep	0.0	memory_off	0
c6_deep_power_down	0.0	memory_idle	0.15
		memory_read	0.399
		memory_write	0.415

memory_search 0.135

PSU

Name: Generic psu

Wattage:1000.0

Scale:1.0

<u>Energy state</u>	<u>Value</u>
psu_20	81.46
psu_50	84.72
psu_100	81.46

Node**Switches**

Name: Node_storage_c

Name:Generic_switch

CPU: AMD_FX_8350_8cores

Queue type: drop_tail_queue

Memory: Maxtor_3TB

Relay unit type: mac_relay_unitpp

Disk: Maxtor_3TB

MTU:1500

Disks:1

CSMA/CD support: yes

Psu: generic_psu

duplex_mode: yes

Hypervisor: main hypervisor

Storage_node:yes

Rack**Virtual machine**

Name:Rack_A_16

Name:VM_small

Storage rack: NO

CPU cores:1

Node type: Node_A Ram memory (mb):1024

Nodes:128 Hard disk(Gb):50

Switch type: Generic switch **Users**

Comm. Link: Ethernet 1 Gbps Name:user_cpu

Delay: 1.25E-4 cpu-intensive application:yes

Ber: 0.0 iterations:25

Per: 0.0 processing/iteration(mips):100000000

 Input(mb):25

 Output(mb):25

 Vm_small:1

Cloud

Name: Cloud_A

Data_center: small cluster

Cloud_manager:FCFS

Hypervisor:CitrixXenServer6_linux

User_cpu: 12 instances

Pre_loaded tenants: 50

Tenants distribution: normal

Mean: 12.0

Standar deviation: 5.0

Simulation time(hours):24

Repetitions(days):0

5.3.2 Δημιουργία αρχείων ρυθμίσεων

Στη συνέχεια εφόσον κάναμε τις παραπάνω ρυθμίσεις, έπειτα αναπαράγαμε τα αρχεία ρυθμίσεων τα οποία είναι απαραίτητα για την εκτέλεση της προσομοίωσης καθώς και για τη δημιουργία κάποιων αποτελεσμάτων που αφορούν τη προσομοίωση. Τα αρχεία που παράγαμε ήταν τα εξής:

- ✓ Scenario.ned⁴→ Πρόκειται για ένα αρχείο όπου αποθηκεύεται η δομή της τοπολογίας του δικτύου, καθώς περιγράφει τη λογική δομή του δικτύου που θα προσομοιωθεί στο software (omnet++).
- ✓ Omnetpp.ini⁵→ Αυτό το αρχείο περιέχει ρυθμίσεις που ελέγχουν πως η προσομοίωση εκτελέστηκε καθώς και τις τιμές των παραμέτρων.
- ✓ Run⁶→ Πρόκειται για βοηθητικό αρχείο εκτέλεσης της προσομοίωσης

5.3.3 Αρχεία αποτελεσμάτων

Έχοντας δημιουργήσει τα παραπάνω αρχεία μπορούμε να ξεκινήσουμε τη προσομοίωση μας. Πηγαίνοντας στο φάκελο από το τερματικό μας που βρίσκονται τα αρχεία ρυθμίσεων εκτελούμε την εντολή `$> ./run -u Cmdenv` και ξεκινάει η προσομοίωση μας.

```

pavlos@pavlos-VirtualBox: ~/omnetpp-4.6
Messages: created: 19162 present: 1693 in FES: 132
** Event #2224896 T=16501 Elapsed: 70.347s (1m 10s)
Speed: ev/sec=21578.1 simsec/sec=166.085 ev/simsec=129.922
Messages: created: 19162 present: 1693 in FES: 132
** Event #2287104 T=16979 Elapsed: 72.356s (1m 12s)
Speed: ev/sec=30964.7 simsec/sec=237.929 ev/simsec=130.142
Messages: created: 19162 present: 1693 in FES: 132
** Event #2342400 T=17405 Elapsed: 74.358s (1m 14s)
Speed: ev/sec=27620.4 simsec/sec=212.787 ev/simsec=129.803
Messages: created: 19162 present: 1693 in FES: 132
** Event #2383616 T=17722 Elapsed: 76.359s (1m 16s)
Speed: ev/sec=20597.7 simsec/sec=158.421 ev/simsec=130.019
Messages: created: 19162 present: 1693 in FES: 132
** Event #2486272 T=18511 Elapsed: 78.363s (1m 18s)
Speed: ev/sec=51251.1 simsec/sec=393.909 ev/simsec=130.109
Messages: created: 19162 present: 1693 in FES: 132
** Event #2556672 T=19053 Elapsed: 80.367s (1m 20s)
Speed: ev/sec=35147.3 simsec/sec=270.594 ev/simsec=129.889
Messages: created: 19162 present: 1693 in FES: 132
** Event #2603264 T=19411 Elapsed: 82.420s (1m 22s)
Speed: ev/sec=22705.7 simsec/sec=174.464 ev/simsec=130.145
Messages: created: 19162 present: 1693 in FES: 132
** Event #2651392 T=19781 Elapsed: 84.442s (1m 24s)
Speed: ev/sec=23802.2 simsec/sec=182.987 ev/simsec=130.076

```

Εικόνα 14.Εναρξη προσομοίωσης σε περιβάλλον linux

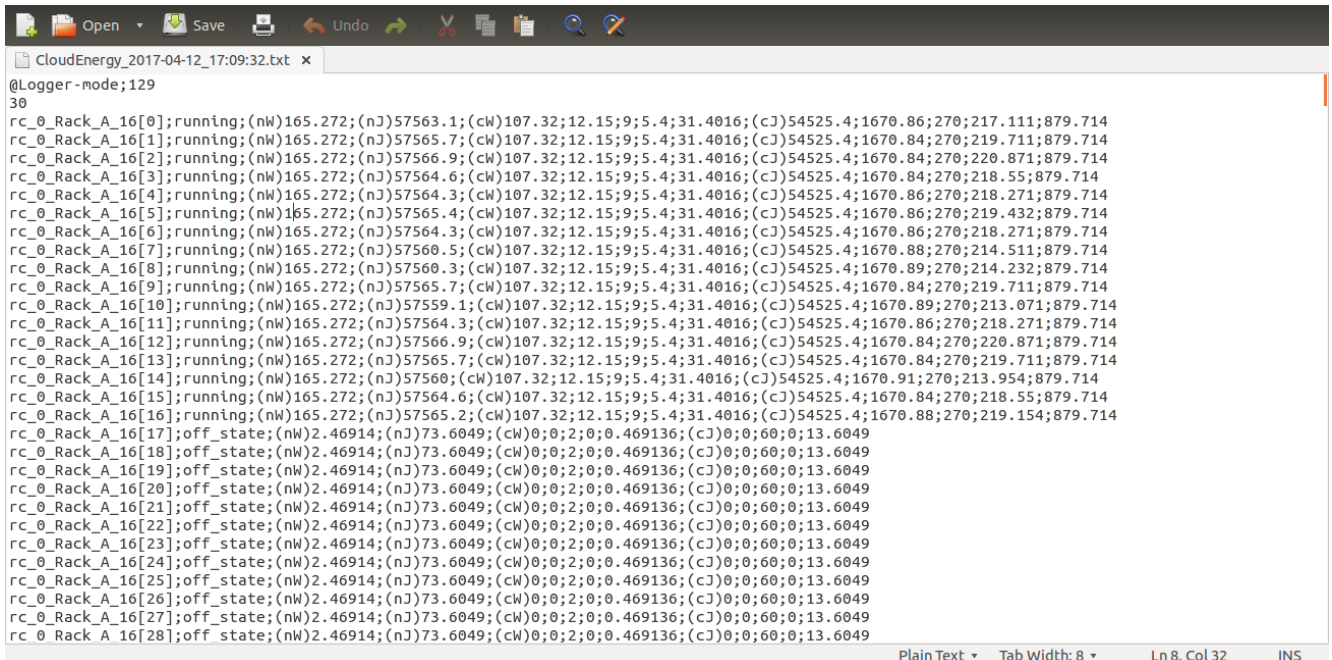
⁴ <https://www.dropbox.com/s/zdl0rpxx3ilj1y5/scenario.ned?dl=0>

⁵ <https://www.dropbox.com/s/7fhati0wbn69blu/omnetpp.ini?dl=0>

⁶ <https://www.dropbox.com/s/omqymx52leag3iv/run?dl=0>

Τα αρχεία txt των αποτελεσμάτων που παράγονται αφορούν:

- ✓ Κατανάλωση ενέργειας των datacenter ⁷(Cloudenergy:_dd/mm/yy).txt
- ✓ Χρόνους εκτέλεσης των χρηστών ⁸(Results:_dd/mm/yy).txt

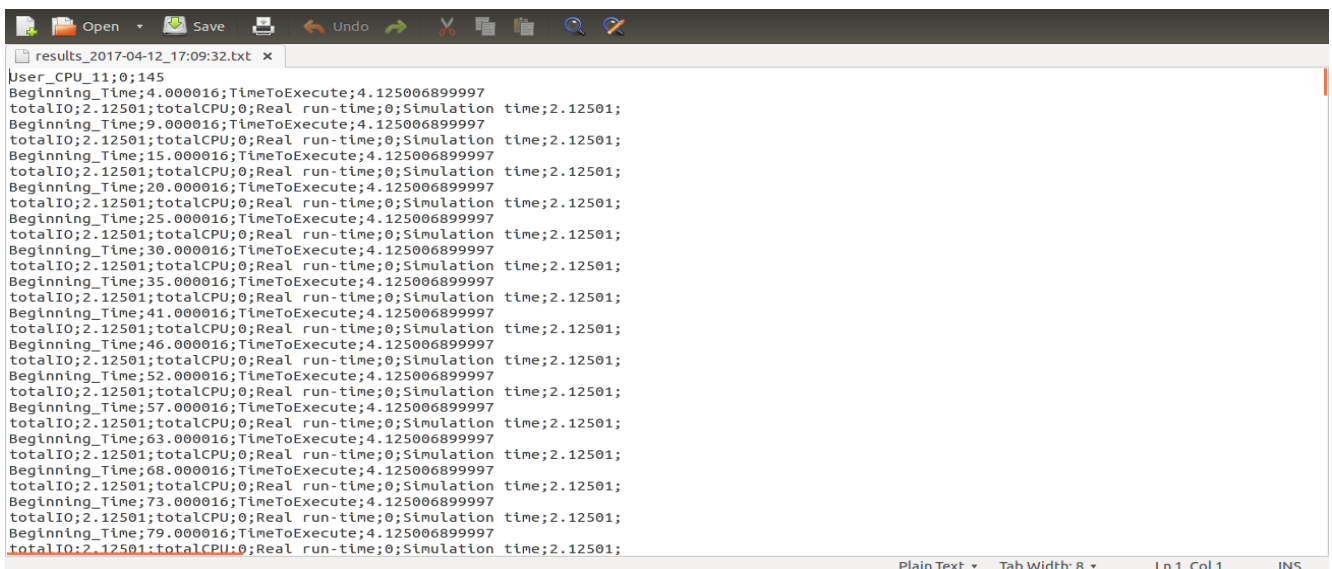


```

@Logger -mode; 129
30
rc_0_Rack_A_16[0];running;(nw)165.272;(nj)57563.1;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.86;270;217.111;879.714
rc_0_Rack_A_16[1];running;(nw)165.272;(nj)57565.7;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.84;270;219.711;879.714
rc_0_Rack_A_16[2];running;(nw)165.272;(nj)57566.9;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.84;270;220.871;879.714
rc_0_Rack_A_16[3];running;(nw)165.272;(nj)57564.6;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.84;270;218.55;879.714
rc_0_Rack_A_16[4];running;(nw)165.272;(nj)57564.3;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.86;270;218.271;879.714
rc_0_Rack_A_16[5];running;(nw)165.272;(nj)57565.4;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.86;270;219.432;879.714
rc_0_Rack_A_16[6];running;(nw)165.272;(nj)57564.3;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.86;270;218.271;879.714
rc_0_Rack_A_16[7];running;(nw)165.272;(nj)57560.5;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.88;270;214.511;879.714
rc_0_Rack_A_16[8];running;(nw)165.272;(nj)57560.3;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.89;270;214.232;879.714
rc_0_Rack_A_16[9];running;(nw)165.272;(nj)57565.7;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.84;270;219.711;879.714
rc_0_Rack_A_16[10];running;(nw)165.272;(nj)57559.1;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.89;270;213.071;879.714
rc_0_Rack_A_16[11];running;(nw)165.272;(nj)57564.3;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.86;270;218.271;879.714
rc_0_Rack_A_16[12];running;(nw)165.272;(nj)57566.9;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.84;270;220.871;879.714
rc_0_Rack_A_16[13];running;(nw)165.272;(nj)57565.7;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.84;270;219.711;879.714
rc_0_Rack_A_16[14];off_state;(nw)165.272;(nj)57560;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.91;270;213.954;879.714
rc_0_Rack_A_16[15];running;(nw)165.272;(nj)57564.6;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.84;270;218.55;879.714
rc_0_Rack_A_16[16];running;(nw)165.272;(nj)57565.2;(cw)107.32;12.15;9;5.4;31.4016;(cJ)54525.4;1670.88;270;219.154;879.714
rc_0_Rack_A_16[17];off_state;(nw)2.46914;(nj)73.6049;(cw)0;0;2;0;0.469136;(cJ)0;0;60;0;13.6049
rc_0_Rack_A_16[18];off_state;(nw)2.46914;(nj)73.6049;(cw)0;0;2;0;0.469136;(cJ)0;0;60;0;13.6049
rc_0_Rack_A_16[19];off_state;(nw)2.46914;(nj)73.6049;(cw)0;0;2;0;0.469136;(cJ)0;0;60;0;13.6049
rc_0_Rack_A_16[20];off_state;(nw)2.46914;(nj)73.6049;(cw)0;0;2;0;0.469136;(cJ)0;0;60;0;13.6049
rc_0_Rack_A_16[21];off_state;(nw)2.46914;(nj)73.6049;(cw)0;0;2;0;0.469136;(cJ)0;0;60;0;13.6049
rc_0_Rack_A_16[22];off_state;(nw)2.46914;(nj)73.6049;(cw)0;0;2;0;0.469136;(cJ)0;0;60;0;13.6049
rc_0_Rack_A_16[23];off_state;(nw)2.46914;(nj)73.6049;(cw)0;0;2;0;0.469136;(cJ)0;0;60;0;13.6049
rc_0_Rack_A_16[24];off_state;(nw)2.46914;(nj)73.6049;(cw)0;0;2;0;0.469136;(cJ)0;0;60;0;13.6049
rc_0_Rack_A_16[25];off_state;(nw)2.46914;(nj)73.6049;(cw)0;0;2;0;0.469136;(cJ)0;0;60;0;13.6049
rc_0_Rack_A_16[26];off_state;(nw)2.46914;(nj)73.6049;(cw)0;0;2;0;0.469136;(cJ)0;0;60;0;13.6049
rc_0_Rack_A_16[27];off_state;(nw)2.46914;(nj)73.6049;(cw)0;0;2;0;0.469136;(cJ)0;0;60;0;13.6049
rc_0_Rack_A_16[28];off_state;(nw)2.46914;(nj)73.6049;(cw)0;0;2;0;0.469136;(cJ)0;0;60;0;13.6049

```

Εικόνα 15. Κατανάλωση ενέργειας νέφους



```

User_CPU_11;0;145
Beginning_Time;4.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;
Beginning_Time;9.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;
Beginning_Time;15.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;
Beginning_Time;20.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;
Beginning_Time;25.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;
Beginning_Time;30.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;
Beginning_Time;35.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;
Beginning_Time;41.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;
Beginning_Time;46.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;
Beginning_Time;52.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;
Beginning_Time;57.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;
Beginning_Time;63.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;
Beginning_Time;68.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;
Beginning_Time;73.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;
Beginning_Time;79.000016;TimeToExecute;4.125006899997
totalIO;2.12501;totalCPU;0;Real run-time;0;Simulation time;2.12501;

```

Εικόνα 16. Χρόνοι εκτέλεσης προσομοίωσης χρηστών

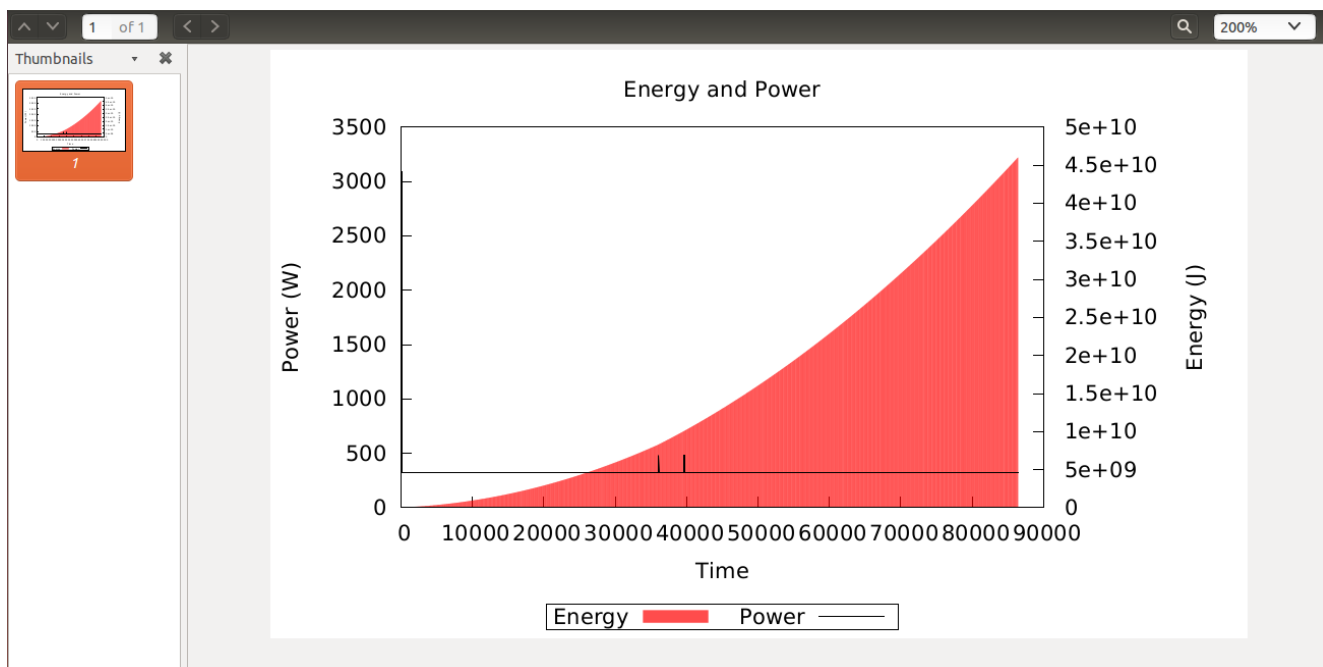
⁷ https://www.dropbox.com/s/hbzg6b694kx5wgc/CloudEnergy_2017-04-12_17_09_32.txt?dl=0

⁸ https://www.dropbox.com/s/m2cvhptd0kfbao7/results_2017-04-12_17_09_32.txt?dl=0

5.3.4 Διαγράμματα αποτελεσμάτων

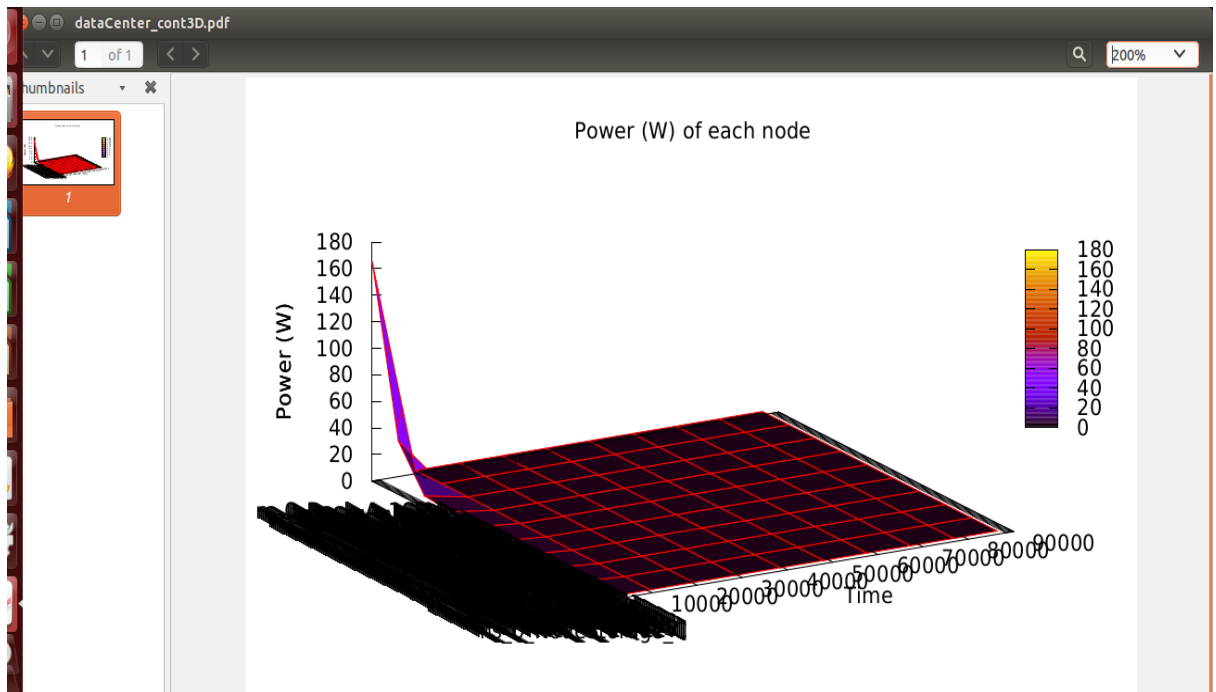
Τα αρχεία *.Gnu files* των γραφικών παραστάσεων καθώς και τα αρχεία *.dat files* προκύπτουν αναπαράγοντας το αρχείο CloudEnergy μέσω του iCanCloudGUI. Στη συνέχεια για να αναπαράγουμε τα διαγράμματα των γραφικών παραστάσεων εκτελούμε από το τερματικό μας την εντολή **bash generateCharts.sh** στο φάκελο όπου βρίσκεται το αντίστοιχο αρχείο.

Η εικόνα 17 μας δείχνει ένα διάγραμμα κατανάλωσης ισχύος και ενέργειας του datacenter κατά τη διάρκεια του χρόνου. Όπως παρατηρούμε η κατανάλωση ισχύος τη χρονική στιγμή 30.0-150.0 seconds παραμένει σταθερή στα 3086.16 Watt και στη συνέχεια από τη χρονική στιγμή 180.0-36000.0 seconds κυμαίνεται στα 318.519 Watt. Από τη χρονική στιγμή 36030.0-36120.0 second παρουσιάζει μια ελαφριά άνοδο στα 481.321 Watt. Από τα 36150.0-39600.0 seconds πέφτει ξανά στα 318.519 Watt. Από τα 39630.0-39720 second παρουσιάζει ξανά μια ελαφριά άνοδο στα 481.321 Watt. Από τα 39750.0-86400.0 seconds όπου ολοκληρώνεται η προσομοίωση παραμένει σταθερή στα 318.519 Watt. Η κατανάλωση ενέργειας (J) έχει διαρκώς ανοδική διακύμανση κατά τη διάρκεια του χρόνου.



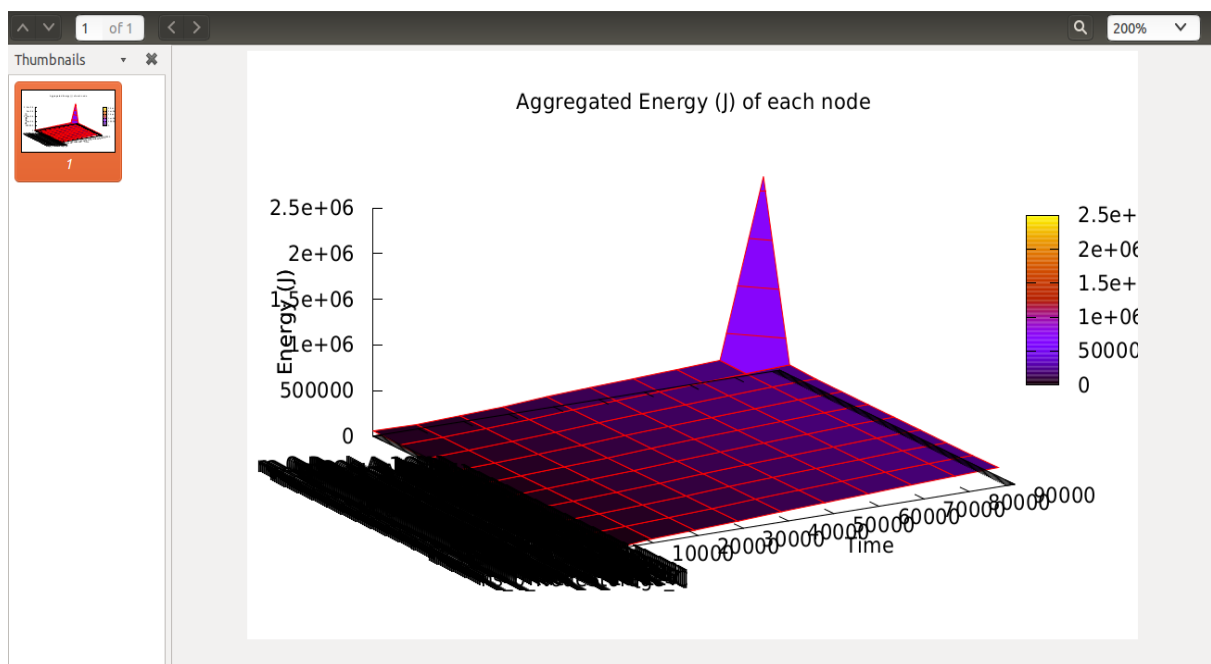
Εικόνα 17. Κατανάλωση ενέργειας και ισχύος του κέντρου δεδομένων

Η εικόνα 18 μας δείχνει σε τρισδιάστατη μορφή την κατανάλωση ισχύος (w) κάθε κόμβου στη διάρκεια του χρόνου. Όπως παρατηρούμε για τους πρώτους 16 κόμβους (σύνολο κόμβων 128) τη χρονική στιγμή 30.0 sec η κατανάλωση ισχύος βρίσκεται στα επίπεδα 165.272 (w) στη συνέχεια μειώνεται στο 2.46914 για τους υπόλοιπους κόμβους καθώς και μέχρι το τέλος της προσομοίωσης τη χρονική στιγμή 86400.0 sec.



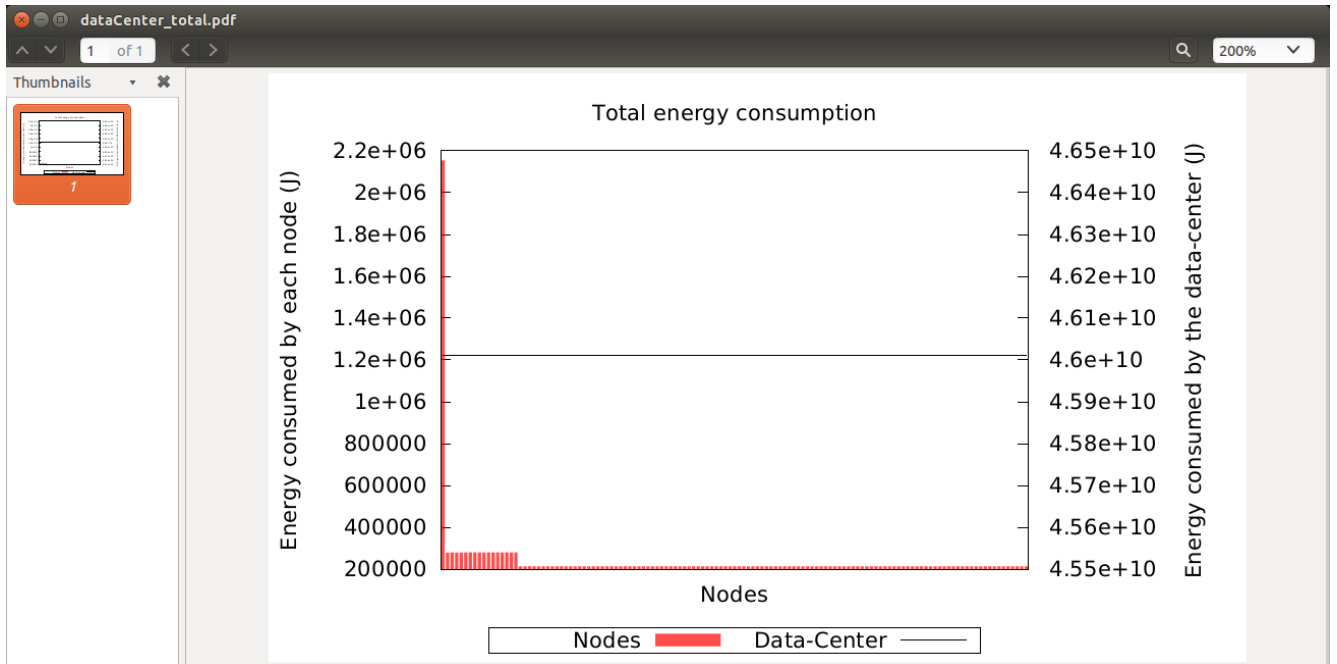
Εικόνα 18. Κατανάλωση ισχύος κάθε κόμβου

Στην εικόνα 19 παρατηρούμε την συγκεντρωτική κατανάλωση ενέργειας των κόμβων που από τη χρονική στιγμή 30.0 -86400.0 seconds παρουσιάζει ανοδικές και καθοδικές διακυμάνσεις σε τακτά χρονικά διαστήματα με την κατανάλωση ενέργειας να φτάνει στο ανώτερο επίπεδο τα 2.500.000 (J).



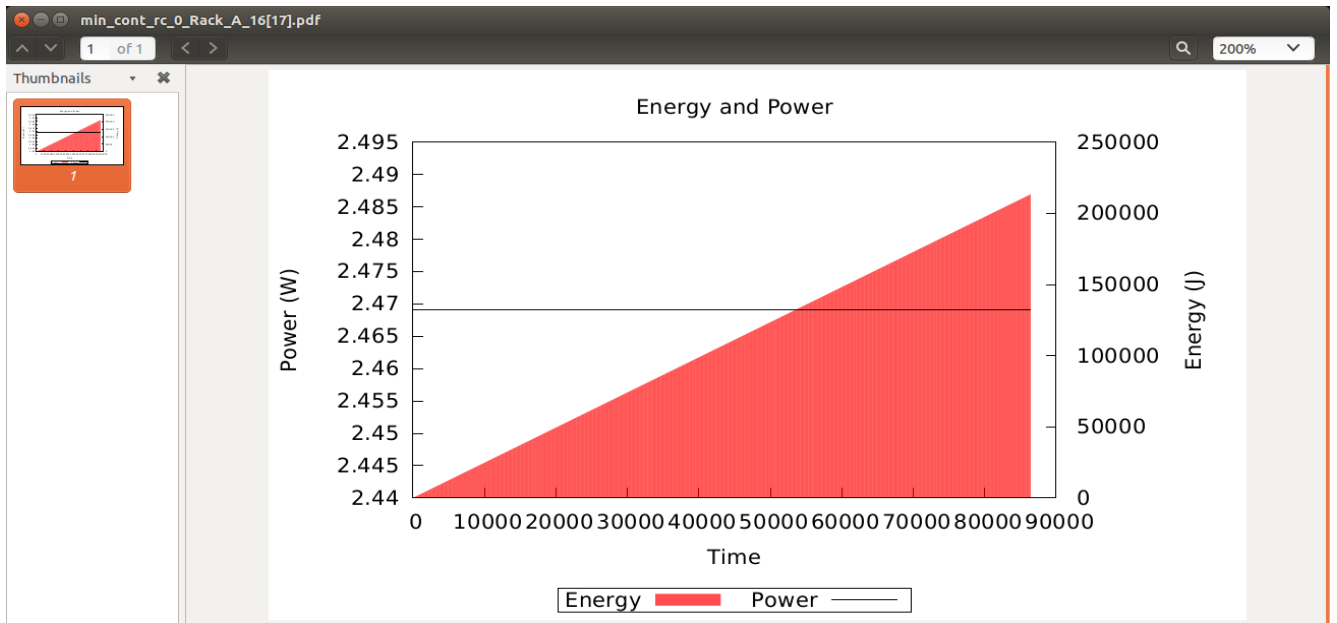
Εικόνα 19. Κατανάλωση ενέργειας κάθε κόμβου

Η εικόνα 20 μας δείχνει σε ένα συγκεντρωτικό διάγραμμα τη συνολική κατανάλωση ενέργειας από κάθε κόμβο καθώς και από το data-center. Όπως παρατηρούμε αρχικά η κατανάλωση είναι μεγαλύτερη έπειτα μειώνεται και σταθεροποιείται στα 213333.0 (J) μέχρι το τέλος της προσομοίωσης. Η κατανάλωση στο datacenter παραμένει σταθερή στα 4.60108 E10.



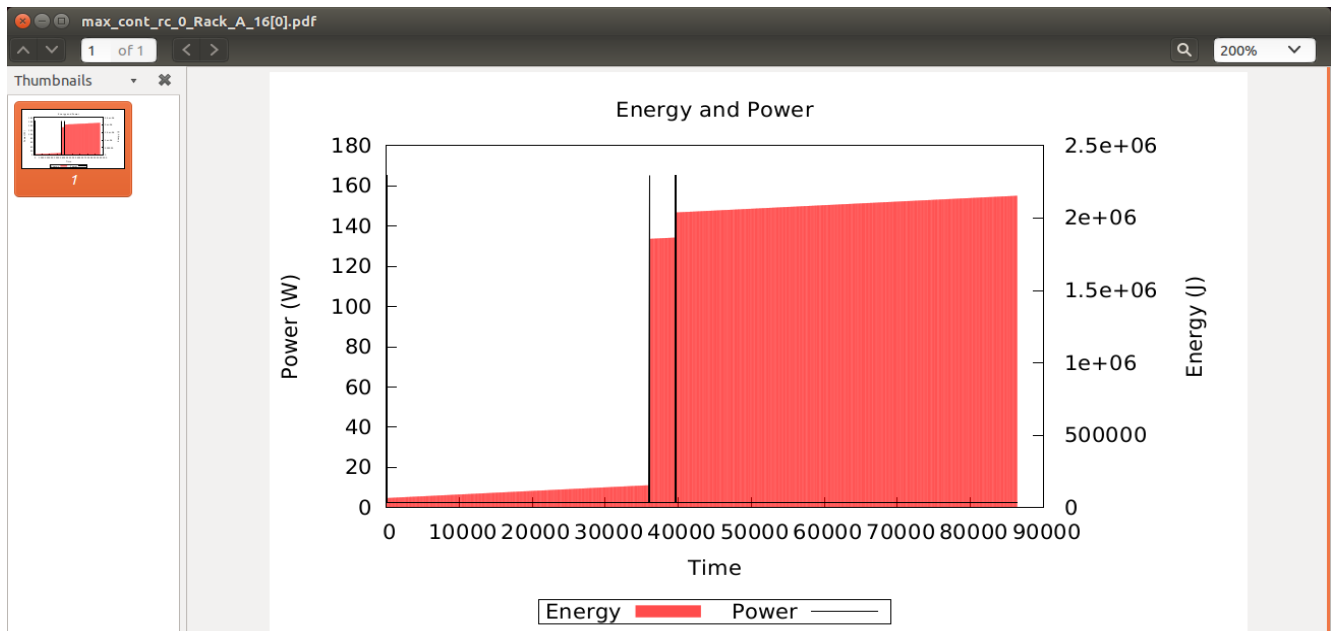
Εικόνα 20.Συνολική κατανάλωση ενέργειας

Η εικόνα 21 μας δείχνει την ελάχιστη κατανάλωση ισχύος και ενέργειας από τη χρονική στιγμή 30.0-86400.0 seconds. Η ελάχιστη ενέργεια κυμαίνεται από 73.6049-213333.0 (J). Η ελάχιστη ισχύ παραμένει σταθερή στα 2.46914 (W) .



Εικόνα 21.Ελάχιστη κατανάλωση ενέργειας και ισχύος

Η εικόνα 22 μας δείχνει τη μέγιστη κατανάλωση ισχύος και ενέργειας προς το χρόνο. Η διακύμανση στην ισχύ παραμένει σταθερή με μια αύξηση στα 165.272 (W) στην αρχή και τη μέση των χρονικών ορίων της προσομοίωσης. Η διακύμανση της ενέργειας παρουσιάζει σταθερή ανοδική πορεία από 57563.1-2153810.0 (J).



Εικόνα 22.Μέγιστη κατανάλωση ενέργειας και ισχύος

5.4 Συμπεράσματα

Στη προσομοίωση μας δημιουργήσαμε ένα εικονικό περιβάλλον υπολογιστικής νέφους δίνοντας τις ανάλογες ρυθμίσεις σύμφωνα με ένα ήδη προδιαμορφωμένο πείραμα. Εξάγαμε αποτελέσματα και γραφικές παραστάσεις που αφορούν τη κατανάλωση ενέργειας και ισχύος των κόμβων και των κέντρων δεδομένων προκειμένου να προβλεφθεί το κόστος σε ένα πραγματικό περιβάλλον υπολογιστικής νέφους.

6 Προσομοιωτής CloudAnalyst

Το CloudAnalyst αναπτύχθηκε στο εργαστήριο GRIDS του πανεπιστημίου της Μελβούρνης και σχεδιάστηκε για μερικούς ειδικούς στόχους. Διαθέτει πλεονέκτημα όπου το κύριο αντικείμενο είναι να προσομοιώσει κατανεμημένες εφαρμογές μεταξύ αρκετών κέντρων δεδομένων και ομάδων χρηστών.

Το CloudAnalyst [30] είναι ένας προσομοιωτής νέφους που προέρχεται από το cloudsim καθώς είναι εντελώς βασισμένος σε γραφικό περιβάλλον χρήστη έχοντας μερικά εκτεταμένα χαρακτηριστικά και δυνατότητες με στόχο να υποστηρίξει την αξιολόγηση των μέσων κοινωνικών δικτύων σύμφωνα με τη γεωγραφική κατανομή των χρηστών και των κέντρων δεδομένων. Σε αυτό το εργαλείο οι κοινότητες των χρηστών και των κέντρων δεδομένων που υποστηρίζουν τα κοινωνικά δίκτυα βασίζονται στη τοποθεσία τους. Παράμετροι όπως η εμπειρία του χρήστη καθώς χρησιμοποιεί την εφαρμογή κοινωνικού δικτύου και ο φόρτος στο κέντρο δεδομένων κυριαρχούν. Το CloudAnalyst χρησιμοποιείται για τη μοντελοποίηση και ανάλυση πραγματικών παγκόσμιων προβλημάτων μέσω μελέτης περιπτώσεων εφαρμογών κοινωνικής δικτύωσης ανεπτυγμένες στο νέφος.

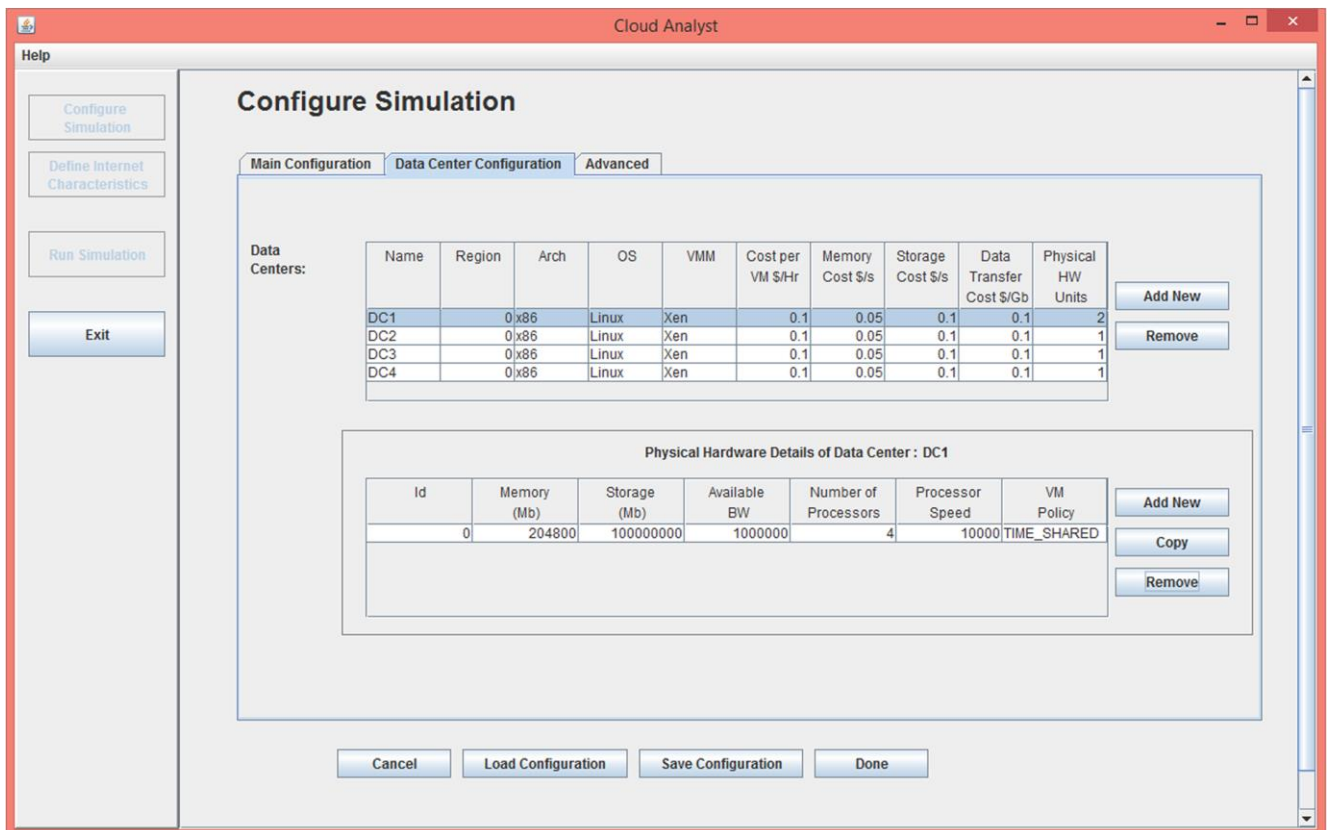
6.1 Χαρακτηριστικά και δυνατότητες

Ο τομέας του CloudAnalyst αποτελείται από τα εξής στοιχεία. *Ιντερνέτ*: μια ιδέα του πραγματικού παγκοσμίου ιντερνέτ. Κατάλληλο εύρος ζώνης και μεταφορά καθυστέρησης εισάγονται και διαμορφώνονται για όλες τις έξι περιοχές του παγκόσμιου χάρτη. *Περιοχή*: Ο κόσμος είναι χωρισμένος σε έξι διαφορετικές περιοχές όπου πάνω οι βάσεις των χρηστών (User Bases) και τα DC's ανήκουν σε μια ή περισσότερες από μια οποιαδήποτε από αυτές τις περιοχές. *Βάση χρηστών*: Δείχνει την ομάδα των χρηστών οι οποίοι λαμβάνονται υπόψη σαν μοναδικό στοιχείο στη προσομοίωση και παράγουν κίνηση για τη προσομοίωση, αυτοί μπορεί να είναι ένας χρήστης ή ένας μεγάλος αριθμός χρηστών. *Ιντερνέτ Cloudlet*: Ομαδοποίηση αιτημάτων από χρήστες. Τα αιτήματα μπορούν να ομαδοποιηθούν σαν ένα μονό ιντερνέτ cloudlet, με πληροφορίες όπως το μέγεθος της εντολής εκτέλεσης της αίτησης και των αρχείων εισόδου εξόδου. *CloudAppServiceBroker*: Στο CloudAnalyst οι έλεγχοι υπηρεσιών Broker η κίνηση μεταξύ DCs και UBs συγκρατούν την αρχή να αποφασίσει ποιες υπηρεσίες κέντρου δεδομένων αιτούνται από τη βάση του χρήστη. *Ελεγκτής κέντρου δεδομένων*: Η πιο σημαντική οντότητα στο τομέα του CloudAnalyst. Είναι το μπροστινό κομμάτι που χρησιμοποιείται από το CloudAnalyst για πρόσβαση στο κέντρο των λειτουργιών των εργαλείων του CloudSim. *VmLoadBalancer*: χρησιμοποιείται στον ελεγκτή κέντρου δεδομένων για να επιλέξει το εικονικό μηχάνημα για να αναθέσει το επόμενο cloudlet προς επεξεργασία. Υπάρχουν τρεις διαφορετικοί ισοροπητές φόρτου(load balancers) που χρησιμοποιούνται: round robin, equally spread current execution load και throttled load balancer.

Ο CloudAnalyst διαθέτει κάποιες επιπλέον δυνατότητες. *Διαθέτει γραφικό περιβάλλον χρήστη*: Εύκολο στη χρήση γραφικό περιβάλλον για δημιουργία και προβολή αποτελεσμάτων όλων των ειδών των πειραμάτων της υπολογιστικής νέφους. *Περιγραφή προσομοίωσης μέσω υψηλού βαθμού παραμετροποίησης και ευελιξία*: Το CloudAnalyst είναι εξοπλισμένο με μοντελοποιητές που έχουν υψηλό βαθμό ελέγχου πάνω στο πείραμα, μοντελοποιώντας οντότητες όπως κέντρα δεδομένων, εικονικές μηχανές, μνήμη, αποθηκευτικό χώρο και εύρος ζώνης. *Επανάληψη πειράματος*: Το CloudAnalyst μπορεί να αποθηκεύσει σενάρια προσομοίωσης και να τα επαναλάβει ξανά και ξανά μέσω παραλλαγών προσομοίωσης. Μπορεί να αποθηκεύσει τα αποτελέσματα σαν XML αρχεία, ακόμα και να αποθηκεύσει σε PDF αρχεία τα αποτελέσματα. *Αποτελεσματικές πληροφορίες*: Το CloudAnalyst παρέχει σε γραφικό περιβάλλον τα αποτελέσματα προσομοίωσης σε μορφή πινάκων και γραφημάτων, εκτός από ένα μεγάλο ποσοστό στατιστικών δεδομένων.

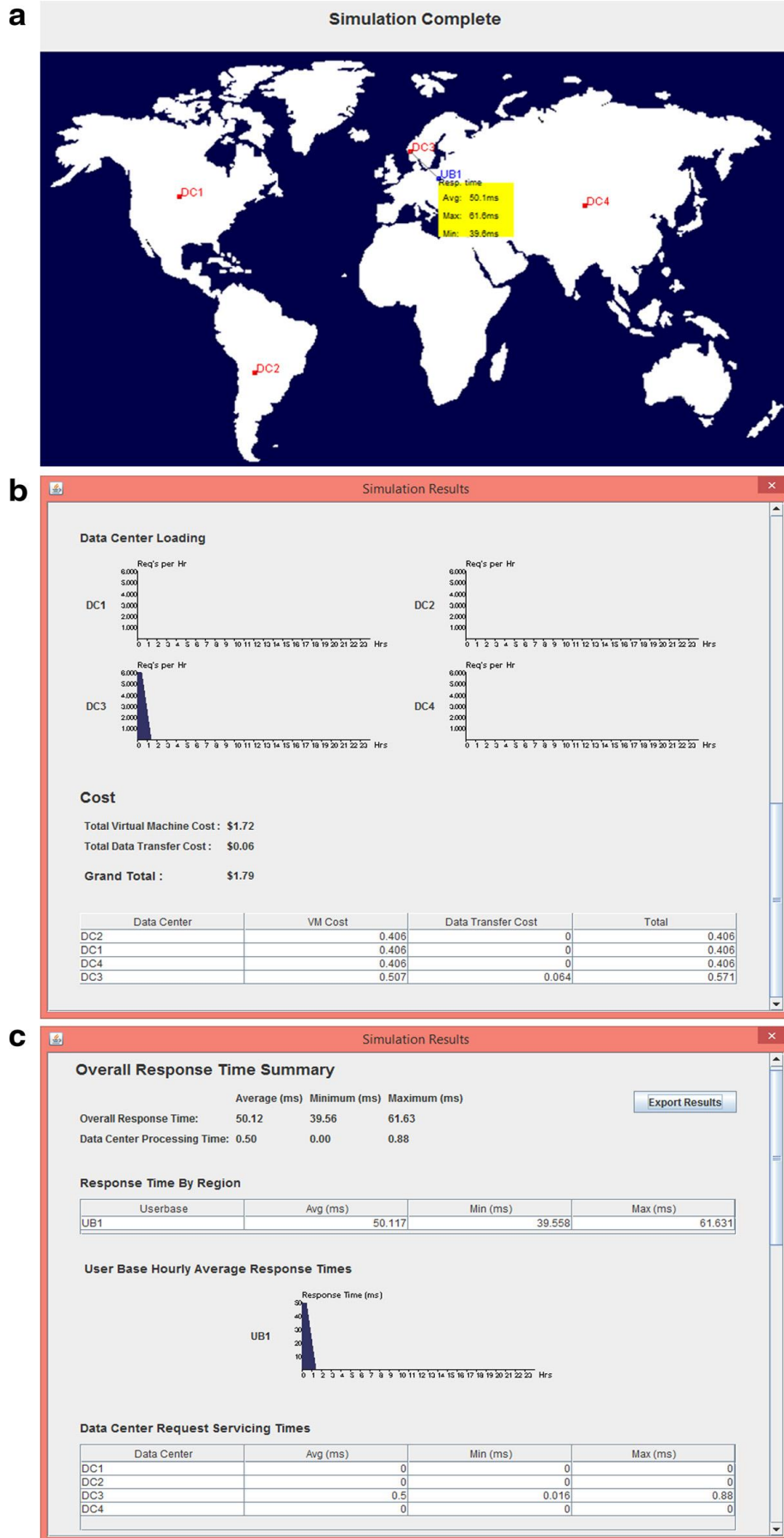
Στο CloudAnalyst περιλαμβάνονται νέα δυναμικά χαρακτηριστικά όπως εύκολη χρήση γραφικού περιβάλλοντος, ικανότητα διαχωρισμού προσομοίωσης από τον κώδικα, γρήγορη προσομοίωση και βελτιωμένα γραφικά αποτελέσματα. Η εγκατάσταση του CloudAnalyst είναι πολύ απλή και με λίγο χρόνο.

Το κύριο παράθυρο του CloudAnalyst επιτρέπει στον χρήστη να διαχειριστεί τη προσομοίωση προσδιορίζοντας τα κέντρα δεδομένων τους host, τις εικονικές μηχανές, και μια βάση χρήστη που αναπαριστά τον αριθμό των χρηστών. Η βάση χρήστη στέλνει αιτήματα που εκτελούνται σύμφωνα με μια πολιτική δρομολόγησης σε ένα συγκεκριμένο κέντρο δεδομένων.



Εικόνα 23. Πίνακας ρυθμίσεων CloudAnalyst

Όταν η προσομοίωση ολοκληρωθεί ο πίνακας εξόδου δείχνει το χρόνο απόκρισης για κάθε βάση χρήστη. Ο προσομοιωτής παρέχει μια λεπτομερή οθόνη με αποτελέσματα που περιλαμβάνουν χρόνο απόκρισης για κάθε χρήστη, αίτημα χρόνου εξυπηρέτησης για κάθε κέντρο δεδομένων και ένα αριθμό όλων των αιτημάτων που εξυπηρετούνται από το κέντρο δεδομένων. Όλα αυτά τα αποτελέσματα φαίνονται σε πίνακες και διαγράμματα για απλοποίηση της ανάλυσης όπως απεικονίζεται στην εικόνα 24.



Εικόνα 24.α) Περιοχές κέντρων δεδομένων και βάσεις χρηστών, β) Αποτελέσματα κόστους γ) Χρόνοι απόκρισης και επεξεργασίας

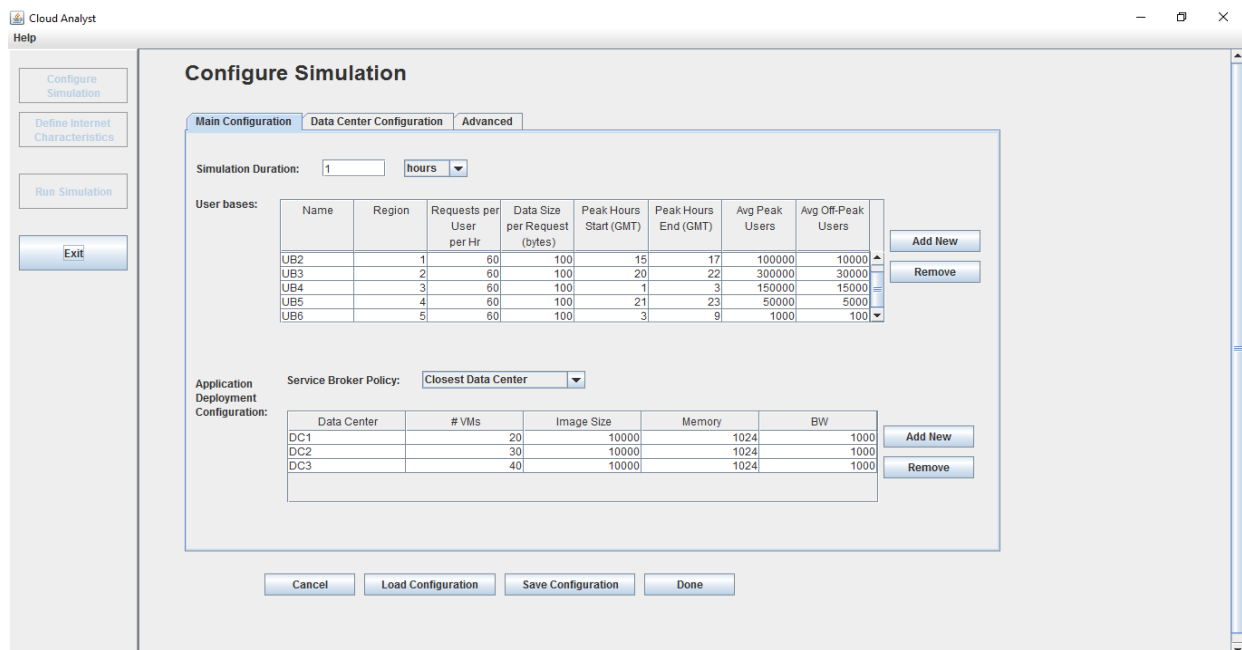
6.2 Πειράματα και αποτελέσματα του CloudAnalyst

Με το JDK του CloudAnalyst πρόκειται να παρουσιάσουμε δύο διαφορετικά πειράματα προκειμένου να δείξουμε τη λειτουργικότητα του προσομοιωτή και το σκοπό που εξυπηρετεί. Στο 1^ο πείραμα θα εξάγουμε αποτελέσματα που αφορούν τους χρόνους απόκρισης σε διάφορες γεωγραφικές περιοχές σε βάσεις των χρηστών, τους χρόνους αιτημάτων εξυπηρέτησης στα κέντρα δεδομένων και τα συνολικά κόστη των περιβαλλόντων νέφους. Στο 2^ο πείραμα θα εξετάσουμε χρόνους απόκρισης και αιτήματος υπηρεσιών καθώς και κόστους επεξεργασίας σύμφωνα με πολιτικές εξισορρόπησης φόρτου των αλγορίθμων **Round Robin**, **Equally spread current execution load** και **Throttled Load balancing**.

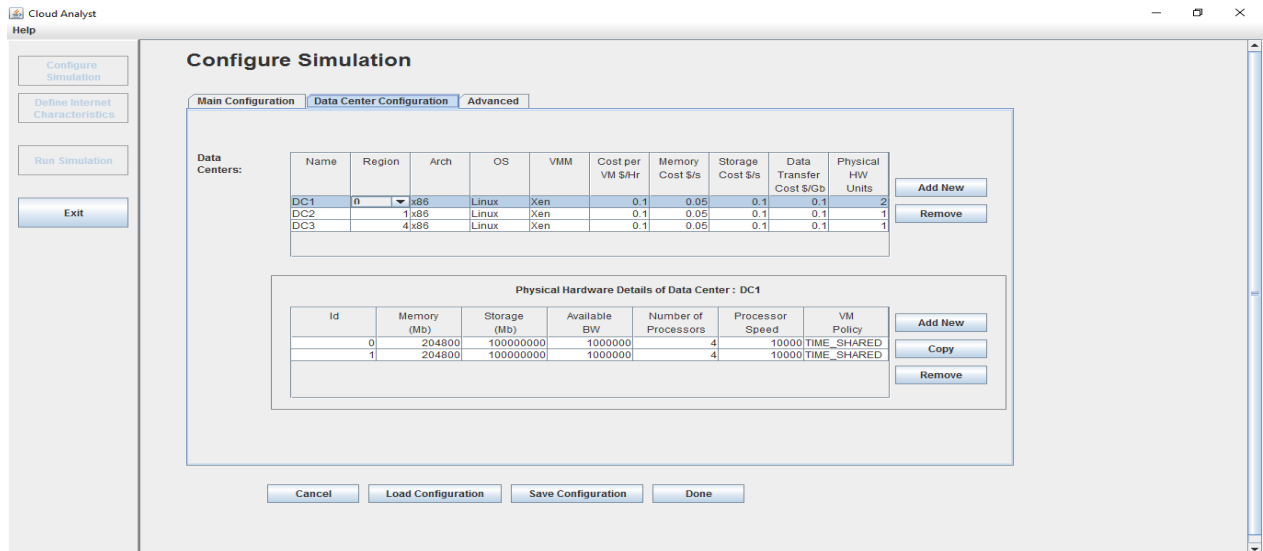
6.2.1 1^ο Πείραμα

Σε αυτό το πείραμα [3] υποθέτουμε ότι υπάρχει μια ιντερνέτ εφαρμογή που βρίσκεται σε 3 κέντρα δεδομένων στο κόσμο. Το περιβάλλον του νέφους αποτελείται από έξι βάσεις χρηστών που είναι γεωγραφικά καταναμημένες και τρία κέντρα δεδομένων με 20,30 και 40 εικονικές μηχανές (εικόνες 25 & 26). Η κίνηση της βάσης του χρήστη δρομολογείται στο κέντρο δεδομένων σύμφωνα με τη πολιτική εγγύτητας των υπηρεσιών που επιλέγει το κοντινότερο κέντρο δεδομένων όσον αφορά τη χαμηλότερη καθυστέρηση του δικτύου. Η διάρκεια της προσομοίωσης είναι 1 ώρα.

Μετά την ολοκλήρωση της προσομοίωσης το CloudAnalyst δείχνει τις περιοχές με αποτελέσματα απόκρισης χρόνου (εικόνα 27). Είναι επίσης πιθανόν να αναλύονται τα αποτελέσματα εξάγοντας ένα αρχείο PDF που περιλαμβάνει αναλυτικά γραφήματα και πίνακες. Η εικόνα 28 μας δείχνει το χρόνο αιτήματος των υπηρεσιών για κάθε κέντρο δεδομένων και η εικόνα 29 τις λεπτομέρειες κόστους αντίστοιχα.

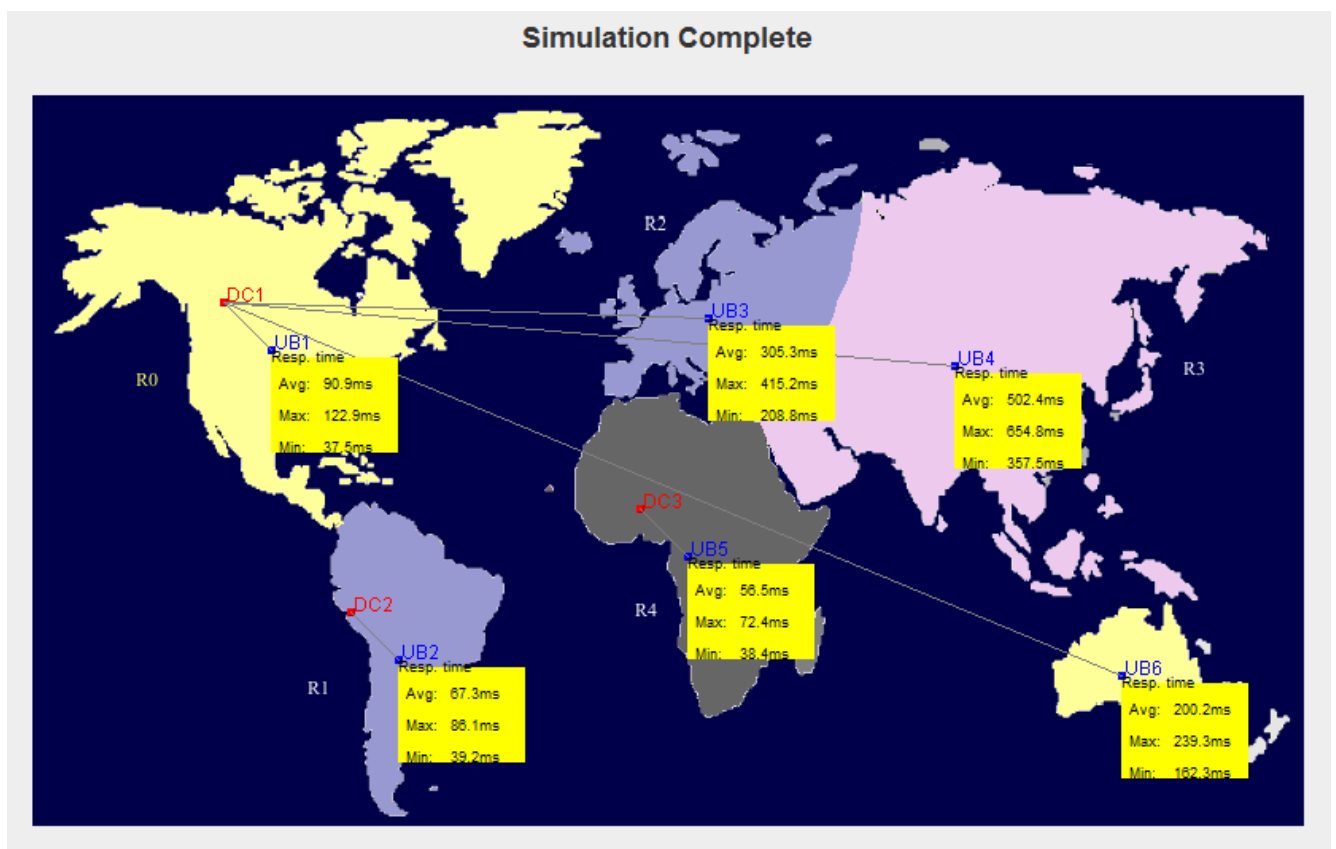


Εικόνα 25.Κύριος πίνακας ρυθμίσεων



Εικόνα 26.Ρυθμίσεις παραμέτρων βάσης χρηστών και κέντρων δεδομένων

6.2.2 Αποτελέσματα



Εικόνα 27.Αποτελέσματα χρόνων απόκρισης ανά γεωγραφική περιοχή

Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	19.441	0.031	59.259
DC2	16.508	0.743	26.81
DC3	5.831	0.998	14.239

Εικόνα 28.Χρόνοι αιτημάτων εξυπηρέτησης κέντρων δεδομένων

Data Center	VM Cost	Data Transfer Cost	Total
DC2	3.011	6.49	9.502
DC1	2.007	53.648	55.655
DC3	4.015	3.274	7.288

Εικόνα 29.Αποτελέσματα κόστους περιβαλλόντων νέφους

6.2.3 Συμπεράσματα

Τα αποτελέσματα που παράγαμε εμείς παρουσιάζουν διαφορές στους χρόνους απόκρισης σε όλες τις βάσεις χρηστών, καθώς επίσης και στους χρόνους αιτημάτων υπηρεσιών στα κέντρα δεδομένων. Η μόνη ομοιότητα βρίσκεται στο κόστος του περιβάλλοντος νέφους όπου στη δική μας περίπτωση υπολογίζεται συνολικά στα 72.45\$ από 71.32\$ όπου η διαφορά είναι σχεδόν μη αξιολογήσιμη.

6.3 2ο Πείραμα

Σε αυτό το πείραμα [6] θα εξετάσουμε και θα συγκρίνουμε διάφορες πολιτικές που χρησιμοποιούνται για εξισορρόπηση φόρτου με τη χρήση του εργαλείου CloudAnalyst. Η εξισορρόπηση φόρτου είναι ένας μηχανισμός που αποφασίζει ποιος κόμβος από αυτούς που κάνει αίτημα θα χρησιμοποιήσει το εικονικό μηχάνημα και ποιο από τα μηχανήματα που κάνουν αίτηση θα τεθεί σε αναμονή. Η εξισορρόπηση φόρτου υποχρεούται να ελαχιστοποιήσει το κόστος του μηχανήματος και να αυξήσει το κέρδος για την υπηρεσία που προσφέρεται. Οι αλγόριθμοι που χρησιμοποιούνται στην υπολογιστική νέφους θα δώσουν μια καλύτερη εικόνα όσον αφορά το χρόνο απόκρισης, το κόστος και ο χρόνος αιτήματος υπηρεσιών του κέντρου δεδομένων. Αυτοί είναι ο **Round Robin**, ο **Equally spread current execution load** και ο **Throttled Load balancing**. Στον *Round Robin* χρησιμοποιείται η τεχνική στην οποία ο χρόνος χωρίζεται σε πολλαπλά κομμάτια. Σε κάθε κόμβο δίνεται ένα συγκεκριμένο κομμάτι χρόνου ή χρόνος με διακοπές. Σε κάθε κόμβο δίνεται ένα κβάντο και σε αυτό το κβάντο ο κόμβος θα εκτελέσει τις διεργασίες του. Οι πόροι του πάροχου υπηρεσίας παρέχονται στον αιτούμενο client στη βάση αυτής της χρονικής περιόδου.

Στον *Equally spread current execution load* (ESCE) αλγόριθμο στον οποίο ο ισορροπιστής φόρτου μεταδίδει το φορτίο σε πολλαπλές εικονικές μηχανές. Ο ισορροπιστής φόρτου διατηρεί μια ουρά από δουλειές που χρειάζεται να χρησιμοποιήσει που αυτή τη στιγμή χρησιμοποιούν τις υπηρεσίες της εικονικής μηχανής. Ο ισορροπιστής έπειτα ελέγχει διαρκώς την ουρά και τη λίστα με τα VMs. Εάν υπάρχει ένα VM διαθέσιμο που μπορεί να χειριστεί το αίτημα του κόμβου τότε το VM διατίθεται σε αυτό. Εάν επομένως υπάρχει ένα VM που είναι ελεύθερο καθώς επίσης και ένα άλλο που θέλει να απαλλαγεί από το φορτίο, τότε ο ισορροπιστής διανέμει

μερικές από τις αρμοδιότητες εκείνου του VM στο ελεύθερο έτσι ώστε να μειώσει το “overhead” του προηγούμενου VM.

Τέλος στον αλγόριθμο *Throttled Load balancing* στον οποίο ο πελάτης κάνει αίτημα στον ισορροπιστή φόρτου να βρει το κατάλληλο εικονικό μηχάνημα να εκτελέσει την απαιτούμενη διεργασία. Πιο αναλυτικά η διαδικασία ξεκινάει διατηρώντας μια λίστα με όλα τα VM. Εάν υπάρχει ταίριασμα και διαθεσιμότητα ενός μηχανήματος, τότε ο ισορροπιστής φόρτου αποδέχεται το αίτημα του πελάτη και διαθέτει το VM σε αυτόν. Εάν δεν υπάρχει διαθέσιμο κανένα VM που να ταιριάζει στα κριτήρια τότε ο ισορροπιστής φόρτου επιστρέφει τιμή -1 και το αίτημα μπαίνει σε ουρά.

6.3.1 Παράμετροι πειράματος

Οι παράμετροι που δόθηκαν για το πείραμα μας είναι για τη βάση του χρήστη όπου οι τοποθεσίες ορίστηκαν σε 6 διαφορετικές βάσεις σε 6 διαφορετικές περιοχές στον κόσμο όπως φαίνεται στην εικόνα 30. Η διάρκεια του πειράματος και για τις τρεις περιπτώσεις είναι 180 min.

Simulation Duration: min

User bases:

Name	Region	Requests per User per Hr	Data Size per Request (bytes)	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg Peak Users	Avg Off-Peak Users
UB1	0	8	1000	13	13	400000	40000
UB2	1	12	1000	15	17	100000	10000
UB3	2	8	1000	20	22	300000	30000
UB4	3	9	1000	1	3	150000	15000
UB5	4	7	1000	21	23	50000	100
UB6	5	6	1000	2	5	1000	100

Add New Remove

Εικόνα 30. Ρυθμίσεις παραμέτρων βάσης χρήστη

Στη συνέχεια τοποθετήσαμε 2 data centers για τις περιοχές R0 και R2 με τις παρακάτω παραμέτρους όπως φαίνεται στην εικόνα 31.

Data Centers:

Name	Region	Arch	OS	VMM	Cost per VM \$/Hr	Memory Cost \$/s	Storage Cost \$/s	Data Transfer Cost \$/Gb	Physical HW Units
DC1	0	x86	Linux	Xen	0.1	0.05	0.1	0.1	2
DC2	2	x86	Linux	Xen	0.1	0.05	0.1	0.1	1

Add New Remove

Εικόνα 31. Ρυθμίσεις παραμέτρων κέντρων δεδομένων

Τέλος ο αριθμός των εικονικών μηχανημάτων που διατέθηκαν για το datacenter 1 είναι 50 και για το datacenter 2 αντίστοιχα 20 όπως φαίνεται στην εικόνα 32.

Application Deployment Configuration:

Service Broker Policy:

Data Center	# VMs	Image Size	Memory	BW
DC1	20	100	1024	10
DC2	50	100	1024	1000

Add New Remove

Εικόνα 32. Επιλογή πολιτικής υπηρεσίας broker

6.3.2 Αποτελέσματα πειράματος

Στο πρώτο πείραμα εξετάσαμε αρχικά την πολιτική του αλγόριθμου **round robin** στη συνέχεια του **Equally spread current execution load(ESCEL)** και τέλος του **Throttled Load balancing** όπου στα αποτελέσματα είναι διάφορες μετρήσεις που αφορούν 1) τον χρόνο απόκρισης ανά περιοχή 2) χρόνο εξυπηρέτησης αιτήματος του κέντρου δεδομένων και το 3) κόστος της διαδικασίας.

1) Round Robin

Overall Response Time Summary			
	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	422.63	49.43	908.59
Data Center Processing Time:	131.52	0.11	333.25

[Export Results](#)

Response Time By Region			
Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	95.59	54.136	128.914
UB2	214.921	156.328	277.088
UB3	117.005	49.433	163.526
UB4	624.606	236.862	908.595
UB5	305.137	248.914	364.43
UB6	200.481	166.382	235.459

Εικόνα 33.Χρόνος απόκρισης με χρήση πολιτικής αλγορίθμου round robin

Data Center Request Servicing Times			
Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	19.437	0.111	42.721
DC2	171.719	0.255	333.245

Εικόνα 34.Χρόνος αιτήματος εξυπηρέτησης με χρήση πολιτικής αλγορίθμου Round Robin

Cost			
Total Virtual Machine Cost :	\$21.04		
Total Data Transfer Cost :	\$528.27		
Grand Total :	\$549.30		

Data Center	VM Cost	Data Transfer Cost	Total
DC2	15.025	388.823	403.848
DC1	6.01	139.444	145.454

Εικόνα 35.Συνολικό κόστος με χρήση αλγορίθμου Round Robin

2) Equally spread current execution load

Overall Response Time Summary			
	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	422.00	49.27	901.63
Data Center Processing Time:	130.75	0.11	317.39

[Export Results](#)

Response Time By Region			
Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	95.456	54.136	128.159
UB2	214.891	156.328	277.088
UB3	117.492	49.273	163.909
UB4	623.481	246.076	901.63
UB5	304.915	242.874	357.677
UB6	200.575	166.382	235.459

Εικόνα 36.Χρόνος απόκρισης με χρήση αλγορίθμου ESCEL

Data Center Request Servicing Times			
Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	19.289	0.111	40.144
DC2	170.728	0.255	317.39

Εικόνα 37.Χρόνος αιτήματος εξυπηρέτησης με χρήση πολιτικής αλγορίθμου ESCEL

Cost			
Total Virtual Machine Cost : \$21.04			
Total Data Transfer Cost : \$528.27			
Grand Total : \$549.30			

Data Center	VM Cost	Data Transfer Cost	Total
DC2	15.025	388.823	403.848
DC1	6.01	139.444	145.454

Εικόνα 38.Κόστος επεξεργασίας με χρήση αλγορίθμου ESCEL

3) Throttled Load balancing

Overall Response Time Summary			
	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	377.39	49.23	801.01
Data Center Processing Time:	65.86	0.11	247.64

[Export Results](#)

Response Time By Region			
Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	86.39	54.136	123.132
UB2	214.911	156.328	277.088
UB3	87.834	49.233	148.444
UB4	558.429	246.076	801.012
UB5	305.331	248.914	368.91
UB6	200.658	166.382	235.459

Εικόνα 39. Χρόνος απόκρισης με χρήση αλγορίθμου Throttled

Data Center Request Servicing Times			
Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	9.982	0.111	31.187
DC2	85.9	0.255	247.639

Εικόνα 40. Χρόνος αιτήματος εξυπηρέτησης με χρήση πολιτικής αλγορίθμου throttled

Cost			
Total Virtual Machine Cost :	\$21.04		
Total Data Transfer Cost :	\$528.27		
Grand Total :	\$549.30		

Data Center	VM Cost	Data Transfer Cost	Total
DC2	15.025	388.823	403.848
DC1	6.01	139.444	145.454

Εικόνα 41. Κόστος επεξεργασίας με χρήση αλγορίθμου throttled

6.3.3 Συμπεράσματα

Όπως παρατηρούμε οι χρόνοι απόκρισης καθώς και οι χρόνοι εξυπηρέτησης των αιτημάτων από τα κέντρα δεδομένων για τις πολιτικές round robin και ESCEL είναι σχεδόν ίδιοι ενώ στη πολιτική throttled είναι πολύ χαμηλότερος συγκριτικά. Το συνολικό κόστος επεξεργασίας και για τις τρεις πολιτικές είναι ακριβώς το ίδιο, επομένως η τεχνική εξισορρόπηση φόρτου δεν παίζει κανένα ρόλο στο κόστος επεξεργασίας.

7 Συμπεράσματα – Σύνοψη

Στη παρούσα διπλωματική εργασία παρουσιάσαμε μια σειρά από προσομοιωτές υπολογιστικής νέφους. Αναλύσαμε λεπτομερώς τις λειτουργίες τους, τα χαρακτηριστικά και τις δυνατότητες τους. Δημιουργήσαμε μια σειρά από προσομοιώσεις δείχνοντας τη χρησιμότητα και το σκοπό του κάθε ενός ξεχωριστά, κάνοντας πειράματα με τα εργαλεία χρήσης του κάθε προσομοιωτή. Τέλος δημιουργήσαμε εγχειρίδια χρήσης για τον κάθε προσομοιωτή ξεχωριστά ώστε οι χρήστες να κατανοήσουν με ευκολία τον τρόπο εγκατάστασης και λειτουργίας τους.

Κάνοντας την εν λόγω διατριβή διαπιστώσαμε επίσης ότι οι προσομοιωτές είναι χρήσιμα εργαλεία που βοηθούν τους χρήστες προκειμένου να επιλέξουν ένα εικονικό περιβάλλον στο οποίο θα προσαρμόσουν κάτι πραγματικό ανάλογα το σκοπό που θέλουν να εξυπηρετηθούν. Οι χρονικές διακυμάνσεις που υπάρχουν αγγίζουν όρια τιμών πραγματικού χρόνου βοηθώντας τους χρήστες στις διεξαγωγές των αποτελεσμάτων τους.

8 Αναφορές

- [1] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F De Rose and Rajkumar Buyy “Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms”, *Softw. Pract. Exper.* 2011; 41:23–50
- [2]Ranjan Kumar, G.Sahoo “Cloud computing simulation using cloudsim”, *International Journal of Engineering Trends and Technology (IJETT) – Volume 8 Number 2- Feb 2014*
- [3]Khadijah Bahwairath, Lo'ai Tawalbeh, Elhadj Benkhelifa3, Yaser Jararweh, Mohammad A. Tawalbeh “Experimental comparison of simulation tools for efficient cloud and mobile cloud computing applications” Bahwairath et al. *EURASIP Journal on Information Security (2016)* 2016:15
- [4]Valerio Persico, Antonio Montieri, Antonio Pescapè “Cloudsurf: a platform for monitoring public-cloud networks” *Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, 2016 IEEE 2nd International Forum on
- [5]A. Nuñez, J.L.Vázquez-Poletti, A. C. Caminero, J. Carretero, and I. M. Llorente “Design of a New Cloud Computing Simulation Platform”, *CCSA 2011: Computational Science and Its Applications - ICCSA 2011* pp 582-593
- [6]Tanveer Ahmed, Yogendra Singh “Analytic Study Of Load Balancing Techniques Using Tool Cloud Analyst”, Tanveer Ahmed, Yogendra Singh/ *International Journal of Engineering Research and Applications (IJERA)* Vol. 2, Issue 2,Mar-Apr 2012, pp.1027-1030
- [7]Thiago Teixeira Sá, Rodrigo N. Calheiros and Danielo G. Gomes” *CloudReports: An Extensible SimulationTool for Energy-Aware Cloud Computing Environments”* Part of the *Computer Communications and Networks* book series (CCN), pp 127-142,2014
- [8] Arif Ahmed, Abadhan Saumya Sabyasachi “Cloud Computing Simulators: A Detailed Survey and Future Direction” *IEEE International Advance Computing Conference (IACC)*, 2014
- [9] Microsoft Corp. "office365.microsoft.com/", 2011
- [10] Salesforce.com Inc. "http://www.salesforce.com/service-cloud/", 2009
- [11] Dropbox - <https://www.dropbox.com/developers>
- [12] Amazon Elastic Compute Cloud (EC2), <http://www.amazon.com/ec2/>
- [13] Enomaly Elastic Computing (2009), <http://www.enomaly.com/>
- [14] R. Buyya, R. Ranjan, R.N. Calheiros, “Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit:Challenges and Opportunities”, *Proceedings of the 7th High Performance Computing and Simulation Conference (HPCS 2009)*,Leipzig, Germany, June 21-24. IEEE Press, New York (2009).
- [15] B. Wickremasinghe, R. N. Calheiros2, R. Buyya, “CloudAnalyst: A CloudSim-based Visual Modeller for analysing Cloud Computing Environments and Applications”, *24th IEEE International Conference on Advanced Information Networking and Applications*, 2010.
- [16] D. Kliazovich, P. Bouvry, S. U. Khan, “GreenCloud: A Packetlevel Simulator of Energy-aware Cloud Computing Data Centers,” *Journal of Supercomputing*, vol. 62, no. 3, pp. 1263-1283, 2012.

- [17] Ns2 Networking simulator, <http://www.isi.edu/nsnam/ns/>
- [18] S. Lim, B. Sharma, G. Nam, E. K. Kim, and C. R. Das “MDCSim: A Multi-tier Data Center Simulation Platform” Cluster Computing and Workshops, 2009.
- [19] A. Nípez, J. L. Vázquez-Poletti, A. C. Caminero, G. G. Castañ, J. Carretero, I. M. Llorente “iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator”, Journal of Grid Computing, March 2012, Volume 10, Issue 1, pp 185-209
- [20] Garg, S. K., & Buyya, R. (2011, December). “NetworkCloudSim: modelling parallel applications in cloud simulations.” In Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on (pp. 105-113). IEEE..
- [21] R. N. Calheiros, M. A. S. Netto, C. A. F. De Rose, and R. Buyya, “EMUSIM: an integrated emulation and simulation environment for modeling, evaluation, and validation of performance of cloud computing applications,” Software-Practice and Experience, 2012.
- [22] S. Ostermann, K. Plankensteiner, R. Prodan, “Th. Fahringer, “GroudSim: An Event-Based Simulation Framework for Computational Grids and Clouds”, Euro-Par 2010 Parallel Processing Workshops Lecture Notes in Computer Science Volume 6586, 2011, pp 305-313.
- [23] M. Tighe, G. Keller, M. Bauer, H. Lutfiyya, “DCSim: A Data Centre Simulation Tool for Evaluating Dynamic Virtualized Resource Management”, 8th international conference and 2012 workshop on systems virtualization management (svm) Network and service management (cnsm), 2012.
- [24] J Jung, H Kim, “MR-CloudSim: Designing and implementing MapReduce computing model on CloudSim”, International Conference on ICT Convergence (ICTC), 2012.
- [25] M. Shiraz, A. Gani, R. H. Khokhar, E Ahmed, “An Extendable Simulation Framework for Modeling Application Processing Potentials of Smart Mobile Devices for Mobile Cloud Computing”, 10th International Conference on Frontiers of Information Technology, 2012.
- [26] S. Sotiriadis, N. Bessis, N. Antonopoulos, A. Anjum, “SimIC: Designing a new Inter-Cloud Simulation platform for integrating largescalesource management”, IEEE 27th International Conference on Advanced Information Networking and Applications, 2013.
- [27] Wikipedia: <https://en.wikipedia.org/wiki/CloudSim>
- [28] Superwits: <http://www.superwits.com/library/cloudsim-simulation-framework>
- [29] Arcos: <http://www.arcos.inf.uc3m.es/~icancloud/Home.html>
- [30] Cloudsim-blogspot: <http://cloudsim-setup.blogspot.gr/?view=classic>

9 Κατάλογος εικόνων

Εικόνα 1.Κώδικας προσομοίωσης με δοκιμή 1 datacenter με 1 host και εκτέλεση 1 cloudlet	8
Εικόνα 2.Αποτελέσματα 1 ^{ης} προσομοίωσης	9
Εικόνα 3.Κώδικας προσομοίωσης με δοκιμή 1 datacenter με 2 hosts και εκτέλεση 2 cloudlets	10
Εικόνα 4.Αποτελέσματα 2 ^{ης} προσομοίωσης	10
Εικόνα 5.Αποτελέσματα 3 ^{ης} προσομοίωσης	14
Εικόνα 6.Αποτελέσματα 3 ^{ης} προσομοίωσης	15
Εικόνα 7.Ρύθμιση παραμέτρων του παρόχου	16
Εικόνα 8.Ρύθμιση παραμέτρων του πελάτη	17
Εικόνα 9.Αριθμός Cloudlets που εκτελέστηκαν επιτυχώς από τον πελάτη σε κάθε κέντρο δεδομένων	18
Εικόνα 11.Ποσότητα χρόνου εκτέλεσης των cloudlet κάθε πελάτη	19
Εικόνα 10. Αριθμός Cloudlets που εκτελέστηκαν επιτυχώς από τον πελάτη σε κάθε κέντρο δεδομένων [3]	19
Εικόνα 12. Ποσότητα χρόνου εκτέλεσης των cloudlet κάθε πελάτη [3]	20
Εικόνα 13. Επίπεδα αρχιτεκτονικής του iCanCloud [5]	24
Εικόνα 14.Εναρξη προσομοίωσης σε περιβάλλον linux	29
Εικόνα 15.Κατανάλωση ενέργειας νέφους	30
Εικόνα 16.Χρόνοι εκτέλεσης προσομοίωσης χρηστών	30
Εικόνα 17.Κατανάλωση ενέργειας και ισχύος του κέντρου δεδομένων	31
Εικόνα 18.Κατανάλωση ισχύος κάθε κόμβου	32
Εικόνα 19. Κατανάλωση ενέργειας κάθε κόμβου	32
Εικόνα 20.Συνολική κατανάλωση ενέργειας	33
Εικόνα 21.Ελάχιστη κατανάλωση ενέργειας και ισχύος	33
Εικόνα 22.Μέγιστη κατανάλωση ενέργειας και ισχύος	34
Εικόνα 23.Πίνακας ρυθμίσεων CloudAnalyst	36
Εικόνα 24.α)Περιοχές κέντρων δεδομένων και βάσεις χρηστών, β)Αποτελέσματα κόστους γ)Χρόνοι απόκρισης και επεξεργασίας	37
Εικόνα 25.Κύριος πίνακας ρυθμίσεων	38
Εικόνα 26.Ρυθμίσεις παραμέτρων βάσης χρηστών και κέντρων δεδομένων	39
Εικόνα 27.Αποτελέσματα χρόνων απόκρισης ανά γεωγραφική περιοχή	39
Εικόνα 28.Χρόνοι αιτημάτων εξυπηρέτησης κέντρων δεδομένων	40
Εικόνα 29.Αποτελέσματα κόστους περιβαλλόντων νέφους	40
Εικόνα 30.Ρυθμίσεις παραμέτρων βάσης χρήστη	41
Εικόνα 31. Ρυθμίσεις παραμέτρων κέντρων δεδομένων	41
Εικόνα 32.Επιλογή πολιτικής υπηρεσίας broker	41
Εικόνα 33.Χρόνος απόκρισης με χρήση πολιτικής αλγορίθμου round robin	42
Εικόνα 34.Χρόνος αιτήματος εξυπηρέτησης με χρήση πολιτικής αλγορίθμου Round Robin	42
Εικόνα 35.Συνολικό κόστος με χρήση αλγορίθμου Round Robin	42
Εικόνα 36.Χρόνος απόκρισης με χρήση αλγορίθμου ESCEL	43
Εικόνα 37.Χρόνος αιτήματος εξυπηρέτησης με χρήση πολιτικής αλγορίθμου ESCEL	43
Εικόνα 38.Κόστος επεξεργασίας με χρήση αλγορίθμου ESCEL	43
Εικόνα 39. Χρόνος απόκρισης με χρήση αλγορίθμου Throttled	44

<i>Εικόνα 40. Χρόνος αιτήματος εξυπηρέτησης με χρήση πολιτικής αλγορίθμου throttled.....</i>	<i>44</i>
<i>Εικόνα 41. Κόστος επεξεργασίας με χρήση αλγορίθμου throttled</i>	<i>44</i>

10 Παράρτημα εγχειριδίων χρήσης προσομοιωτών

- i. Cloudsim: <https://www.dropbox.com/s/sthff81z27vce75/Manual%20cloudsim.docx?dl=0>
- ii. Cloudsurf: <https://www.dropbox.com/s/0lo8k5ksqmvx2ob/Manual%20Cloudsurf.docx?dl=0>
- iii. Cloudreports: <https://www.dropbox.com/s/rdt0kp7pvg6yhm4/Manual%20cloudreports.docx?dl=0>
- iv. CloudAnalyst: <https://www.dropbox.com/s/5al7a5jmh7e27x2/Manual%20cloudAnalyst.docx?dl=0>
- v. iCanCloud: <https://www.dropbox.com/s/tg8lh9b3j614atr/Manual%20iCanCloud.docx?dl=0>

Τριμελής Εξεταστική Επιτροπή

Δουληγέρης Χρήστος

Καθηγητής

(Υπογραφή)

Ψαράκης Μιχαήλ

Επίκουρος καθηγητής

(Υπογραφή)

Κοτζανικολάου Παναγιώτης

Επίκουρος καθηγητής

(Υπογραφή)