



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΜΣ «ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΚΑΙ ΥΠΗΡΕΣΙΕΣ»

Προσομοίωση σε Περιβάλλον Κινητής Νεφούπολογιστικής

Διπλωματική Εργασία

Φοιτητής

Νικόλαος Παναγόπουλος

Επιβλέπων Καθηγητής

Αν. Καθηγητής Άγγελος Ρούσκας

Πειραιάς, Δεκέμβριος 2016

Ευχαριστίες

Θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες στον καθηγητή Άγγελο Ρούσκα για την υποστήριξή του και καθοδήγηση που μου παρείχε για την ολοκλήρωση της παρούσας διπλωματικής εργασίας, όπως επίσης τον υποψήφιο διδάκτορα Ευθύμιο Οικονόμου για την βοήθεια που μου παρείχε. Ιδιαίτερες ευχαριστίες θέλω να απευθύνω στη φίλη μου Θεώνη για την αμέριστη συμπαράσταση, υπομονή και ενθάρρυνσή της. Τέλος θέλω να ευχαριστήσω την οικογένειά μου για την απρόσκοπτη και ανιδιοτελή συμπαράστασή τους.

Περίληψη

Στις μέρες μας οι κινητές συσκευές είναι πανταχού παρούσες στην καθημερινότητα των ανθρώπων και οι απαιτήσεις των εφαρμογών των κινητών συσκευών για διαθέσιμους πόρους αυξάνονται διαρκώς. Παρόλα αυτά, οι διαθέσιμοι πόροι είναι περιορισμένοι. Η κινητή νεφοϋπολογιστική αντιμετωπίζει το πρόβλημα της έλλειψης των διαθέσιμων πόρων των κινητών συσκευών με την αποφόρτιση δεδομένων και επεξεργαστικής ισχύος προς το υπολογιστικό νέφος. Στην εξέλιξη της σύγκλισης των τεχνολογιών, τα Cloudlet αποτελούν ένα σημαντικό συνδετικό στοιχείο στην client-cloud ιεραρχία.

Η προσομοίωση και δοκιμή των cloudlets σε πραγματικές συνθήκες είναι δύσκολο να πραγματοποιηθεί, οπότε για το σκοπό αυτό χρησιμοποιούνται εργαλεία προσομοίωσης. Σκοπός της παρούσας διπλωματικής εργασίας ήταν η μελέτη του Cloudlet-based Cloud Computing και η απόπειρα προσομοίωσης ενός αντίστοιχου περιβάλλοντος στο πλαίσιο μίας όσο το δυνατόν ρεαλιστικής περίπτωσης χρήσης. Για λόγους ευχρηστίας επιλέχθηκε το εργαλείο ανοιχτού κώδικα Cloud Analyst για την εκτέλεση σεναρίων προσομοίωσης ενός περιβάλλοντος όπου χρήστες με κινητές συσκευές δημιουργούν αιτήματα προς εφαρμογές, τα οποία εξυπηρετούνται είτε από ένα απομακρυσμένο Data Center μόνο είτε από το Data Center και ένα μικρότερο Cloudlet είτε από το Data Center και περισσότερα Cloudlets. Κατά την εκτέλεση των σεναρίων έγινε σύγκριση των αλγορίθμων επιλογής Data Center, που διαθέτει το εργαλείο CloudAnalyst, μαζί με έναν ακόμα αλγόριθμο, που υλοποιήθηκε στο πλαίσιο της εργασίας, προκειμένου να εξαχθούν συμπεράσματα σχετικά με το κατά πόσον η ύπαρξη των Cloudlets μπορεί να βελτιώσει τους χρόνους επεξεργασίας αιτημάτων και απόκρισης των Data Center και να αναβαθμίσει την εμπειρία του χρήστη.

Abstract

Nowadays mobile devices are ubiquitous in everyday life and the request of mobile applications for available resources is constantly increasing. However, the actual available resources are limited. Mobile Computing addresses the problem of lack of resources by offloading data and processing power from the Cloud or the mobile device to the Cloudlet, a small “cloud-in-a-box” located closer to the user. In the evolution of the convergence of technologies, Cloudlet an important connecting element in client-cloud hierarchy.

The simulation and testing of Cloudlets in real conditions is difficult to achieve, the use of simulation tools for this purpose is mandatory. The aim of this thesis was to study the Cloudlet-based Cloud Computing and attempt to simulate a relevant environment of a realistic use case. For reasons of simplicity we selected Cloud Analyst, an open-source simulation tool, in order to perform test scenarios in an environment where users with mobile devices create requests for applications, which are served by either a remote Data Center only or from the Data Center and a smaller Cloudlet either from the Data Center and more Cloudlets. During the simulation we had the opportunity to test the Service Broker Policies featured in CloudAnalyst, as well as to implement another Service Broker algorithm, in order to investigate whether the existence of Cloudlets can improve Data Centers’ processing and response times and enhance the overall user experience.

Περιεχόμενα

Ευχαριστίες.....	i
Περίληψη.....	ii
Abstract	iii
Περιεχόμενα.....	iv
Κατάλογος εικόνων	vi
Εισαγωγή	1
1 Mobile Cloud Computing.....	2
1.1 Ορισμός	2
1.2 Αρχιτεκτονική	2
1.3 Περιορισμοί και Πλεονεκτήματα	5
1.4 Εφαρμογές Κινητής Νεφούπολογιστικής.....	6
1.4.1 Mobile learning.....	6
1.4.2 Mobile commerce	6
1.4.3 Mobile healthcare	7
1.4.4 Mobile gaming.....	7
1.4.5 Social networking	8
1.5 Καθυστέρηση μετάδοσης (delay).....	9
1.6 Fog Computing (FC)	10
1.7 Mobile Edge Computing (MEC)	12
1.8 Cloudlets/mDc.....	13
2 Cloud Simulators	17
2.1 CloudSim.....	17
2.2 CloudAnalyst.....	18
2.3 GreenCloud.....	19
2.4 iCanCloud.....	19
2.5 SmartSim	19
2.6 MR-CloudSim	20
2.7 SimIC.....	20
2.8 Κριτήρια Αξιολόγησης.....	20
3 CloudAnalyst.....	23

3.1	Πλεονεκτήματα χρήσης.....	23
3.2	Βασικά συστατικά στοιχεία.....	24
3.3	Δρομολόγηση των αιτημάτων των χρηστών.....	28
3.4	Υπολογισμός της καθυστέρησης μετάδοσης.....	29
3.5	Αλγόριθμοι επιλογής DataCenter	30
3.6	Υλοποίηση αλγορίθμου στο CloudAnalyst.....	32
4	Μελέτη περίπτωσης χρήσης και προσομοίωση κινητών εφαρμογών σε περιβάλλον με Cloudlets.....	35
4.1	Εισαγωγή.....	35
4.2	Σενάρια προσομοίωσης	35
4.3	Σενάριο 1	38
4.3.1	Παράμετροι προσομοίωσης.....	38
4.3.2	Αποτελέσματα προσομοίωσης	39
4.4	Σενάριο 2	40
4.4.1	Παράμετροι προσομοίωσης.....	41
4.4.2	Αποτελέσματα προσομοίωσης	42
4.5	Σενάριο 3	44
4.5.1	Παράμετροι προσομοίωσης.....	44
4.5.2	Αποτελέσματα προσομοίωσης	45
5	Συμπεράσματα	49
	Αναφορές	50

Κατάλογος εικόνων

Εικόνα 1-1: Η αρχιτεκτονική του MCC	3
Εικόνα 1-2: Service-Oriented Αρχιτεκτονική στο CC	4
Εικόνα 1-3: Η καθυστέρηση στο MC	9
Εικόνα 1-4: Η επίδραση του IoT στην Οικονομία μέχρι το 2025	10
Εικόνα 1-5: Αρχιτεκτονική MEC [16]	12
Εικόνα 1-6: Η αρχιτεκτονική Mobile-Cloudlet-Cloud Computing [21]	14
Εικόνα 1-7: Hyperscale DC ανά τον κόσμο	15
Εικόνα 1-8: Hyper Scale Data Center	16
Εικόνα 2-1: Η αρχιτεκτονική του CloudSim	18
Εικόνα 2-2: Η αρχιτεκτονική του CloudAnalyst	19
Εικόνα 2-3: Συγκριτικός πίνακας προσομοιωτών Cloud. Πηγή: Suryateja, Pericherla S. "A Comparative Analysis of Cloud Simulators." International Journal of Modern Education and Computer Science (IJMECS) 8.4 (2016): 64.	21
Εικόνα 3-1: Οι περιοχές του Cloud Analyst πάνω στον παγκόσμιο χάρτη	24
Εικόνα 3-2: Οι πίνακες παραμετροποίησης των Delay & Bandwidth ανά περιοχή	25
Εικόνα 3-3: Ρύθμιση Βάσης Χρηστών	25
Εικόνα 3-4: Τα πεδία για την ρύθμιση των παραμέτρων για τα Data Center στον CloudAnalyst	26
Εικόνα 3-5: Το διάγραμμα ροής για την δρομολόγηση των αιτημάτων των χρηστών	28
Εικόνα 3-6: Ο αλγόριθμος Service Proximity Service Broker	30
Εικόνα 4-1: Γραφική απεικόνιση της προσομοίωσης του σεναρίου 1 στο Cloud Analyst	38
Εικόνα 4-2: Γραφική απεικόνιση της προσομοίωσης του σεναρίου 2 στο Cloud Analyst	41
Εικόνα 4-3: Γραφική απεικόνιση της προσομοίωσης του σεναρίου 3 στο Cloud Analyst	44
Εικόνα 4-4: Φόρτος αιτημάτων στα Cloudlets με Service Proximity Service Broker	47
Εικόνα 4-5: Φόρτος αιτημάτων στα Cloudlets με Service Proximity Round Robin Service Broker	48

Εισαγωγή

Σκοπός της παρούσας διπλωματικής εργασίας είναι η μελέτη του Cloudlet-based Mobile Computing και η απόπειρα προσομοίωσης ενός αντίστοιχου περιβάλλοντος στο πλαίσιο μίας όσο το δυνατόν ρεαλιστικής περίπτωσης χρήσης, με την βοήθεια ενός εργαλείου προσομοίωσης.

Η ευελιξία και η διάδοση της τεχνολογίας του Cloud Computing σε συνδυασμό με την ραγδαία αύξηση κινητών συσκευών, εφαρμογών και υπηρεσιών οδήγησε στην εφαρμογή της ιδέας του Cloud στα δίκτυα κινητών επικοινωνιών. Σταδιακά, οι προκλήσεις όσον αφορά την καθυστέρηση διάδοσης της πληροφορίας στα απομακρυσμένα DataCenters, καθώς και οι αυξημένες απαιτήσεις των χρηστών κινητών συσκευών τόσο ως προς την επεξεργαστική ισχύ όσο και ως προς τον ρυθμό μετάδοσης, έφεραν πιο κοντά στην πραγματικότητα την ιδέα της δημιουργίας DataCenters (Cloudlets) μικρής κλίμακας και χαμηλού κόστους κοντά στους χρήστες, με γνώμονα την αποφόρτιση (offloading) δεδομένων και επεξεργαστικής ισχύος τόσο από τις συσκευές όσο και από τα απομακρυσμένα DataCenters κατά περίπτωση, καθώς και την αύξηση του ρυθμού μετάδοσης δεδομένων (bandwidth) σε επίπεδα τοπικών δικτύων (LAN).

Στην παρούσα εργασία γίνεται αρχικά μία συνοπτική περιγραφή των τεχνολογιών του Cloud Computing, του Mobile Cloud Computing και των Cloudlets ως προς τα γενικά χαρακτηριστικά, την αρχιτεκτονική και τις εφαρμογές.

Κατόπιν, περιγράφονται συνοπτικά τα διαθέσιμα εργαλεία προσομοίωσης ενός περιβάλλοντος Cloud Computing με έμφαση στο εργαλείο CloudAnalyst, το οποίο είναι και αυτό που επιλέχθηκε για την παρούσα εργασία. Το εργαλείο περιγράφεται αναλυτικά ως προς τα συστατικά του στοιχεία, τις διαθέσιμες στον χρήστη παραμέτρους και τους ενσωματωμένους αλγορίθμους, βάσει των οποίων το εργαλείο επιλέγει κατά την προσομοίωση το DataCenter, στο οποίο θα αποστείλει το επόμενο αίτημα προς επεξεργασία. Στο πλαίσιο αυτό έγινε επιπλέον η υλοποίηση ενός επιπλέον αλγορίθμου επιλογής DataCenter.

Η κεντρική ενότητα της εργασίας αποτελείται από την μελέτη μίας περίπτωσης χρήσης με ένα απομακρυσμένο DataCenter και ένα ή περισσότερα μικρότερα Cloudlets, καθώς και την προσομοίωσή της με την βοήθεια του εργαλείου CloudAnalyst. Η προσομοίωση έγινε με την εκτέλεση διαφορετικών σεναρίων, ώστε να εξεταστούν οι δυνατότητες του εργαλείου, να συγκριθούν οι διάφορες πολιτικές επιλογής των DataCenters, που θα επεξεργαστούν συγκεκριμένα αιτήματα, να αξιολογηθεί εν γένει η χρήση των Cloudlets βοηθητικά μαζί με τα παραδοσιακά απομακρυσμένα DataCenters –όσο κάτι τέτοιο είναι εφικτό στο συγκεκριμένο περιβάλλον προσομοίωσης-, να παρουσιαστούν τα αποτελέσματα και να εξαχθούν τα σχετικά συμπεράσματα.

1 Mobile Cloud Computing

Η ερευνητική πρόοδος που έχει σημειωθεί τα τελευταία χρόνια στον τομέα των ψηφιακών επικοινωνιών και δικτύων, καθώς επίσης και οι ολοένα μεγαλύτερες απαιτήσεις των χρηστών από εφαρμογές μέσω κινητών συσκευών, έχει οδηγήσει στην ανάπτυξη ενός καινοτόμου τομέα της Κινητής Νεφοϋπολογιστικής (Mobile Cloud Computing-MCC), που είναι συνδυασμός των Mobile Computing, Cloud Computing και των Ασύρματων Δικτύων. Καθώς οι φορητές συσκευές αποκτούν ολοένα και μεγαλύτερο ενδιαφέρον στην καθημερινότητα των ανθρώπων ως το αποτελεσματικότερο και πιο εύχρηστο εργαλείο επικοινωνίας, δίνουν την δυνατότητα στον χρήστη να ενσωματώσει πλούσια εμπειρία από διάφορες υπηρεσίες που του προσφέρονται μέσω των εφαρμογών, είτε αξιοποιώντας τις δυνατότητες της συσκευής τοπικά είτε χρησιμοποιώντας την επεξεργαστική ισχύ κάποιου απομακρυσμένου server μέσω κάποιου ασύρματου δικτύου. Ο όρος MCC παρουσιάστηκε λίγο αργότερα από την ιδέα του CC, προσελκύνοντας το επιχειρηματικό ενδιαφέρον ως μια κερδοφόρα επιχειρηματική επιλογή που μειώνει την ανάπτυξη του κόστους λειτουργίας των φορητών εφαρμογών, δίνει την δυνατότητα στους χρήστες να έχουν πληθώρα εμπειριών μέσω εφαρμογών χαμηλού κόστους αλλά ταυτόχρονα δίνει τη δυνατότητα στους ερευνητές ως μια καλά υποσχόμενη λύση για Green Computing [1].

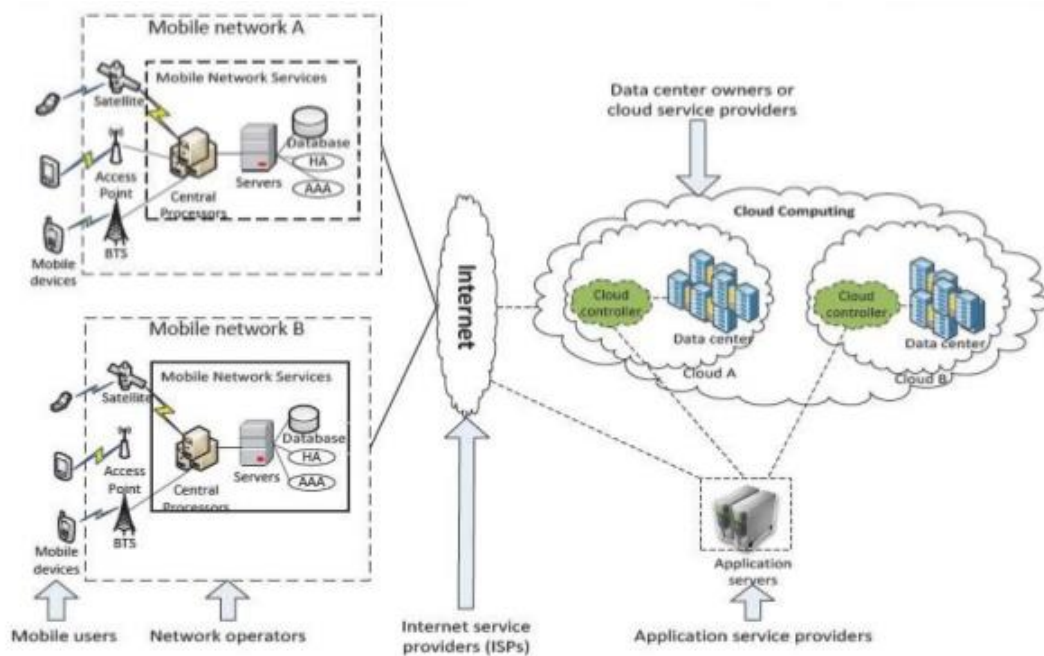
1.1 Ορισμός

Με απλά λόγια η Κινητή Νεφοϋπολογιστική είναι μια υποδομή όπου τόσο η αποθήκευση όσο και η επεξεργασία των δεδομένων γίνεται μακριά από την φορητή συσκευή σε πανίσχυρες κεντροποιημένες πλατφόρμες εντός των υπολογιστικών νεφών. Ουσιαστικά το MCC επεκτείνει τις επεξεργαστικές δυνατότητες των φορητών συσκευών και επιπλέον προσφέρει σε αυτές χώρους για την αποθήκευση δεδομένων.

1.2 Αρχιτεκτονική

Η βασική αρχιτεκτονική του MCC αναπαρίσταται στο Σχήμα 1, όπου οι φορητές συσκευές είναι διασυνδεδεμένες στα δίκτυα της κινητής τηλεφωνίας μέσω των σταθμών βάσης, οι οποίες πραγματοποιούν και ελέγχουν τις ζεύξεις και τις λειτουργικές διεπαφές μεταξύ του δικτύου και των συσκευών. Τα αιτήματα των κινητών χρηστών και οι πληροφορίες (ID και location) μεταδίδονται στους κεντρικούς επεξεργαστές που είναι συνδεδεμένοι με τους servers που παρέχουν τις υπηρεσίες δικτύου. Στο σημείο αυτό οι πάροχοι προσφέρουν στους χρήστες υπηρεσίες AAA, μέσω του HA και των συνδρομητικών δεδομένων, που είναι αποθηκευμένα στις βάσεις τους. Τα δεδομένα αυτά παραδίδονται μέσω του Internet στο Cloud. Στο Cloud

υπάρχουν οι cloud controllers, οι οποίοι επεξεργάζονται τα δεδομένα αυτά για να μπορέσουν να προσφέρουν τις επιθυμητές υπηρεσίες, οι οποίες υλοποιούνται με τη λογική της βοηθητικής επεξεργασίας, της εικονικοποίησης και της προσανατολισμένης στις υπηρεσίες αρχιτεκτονικής (π.χ. web/application/database servers).



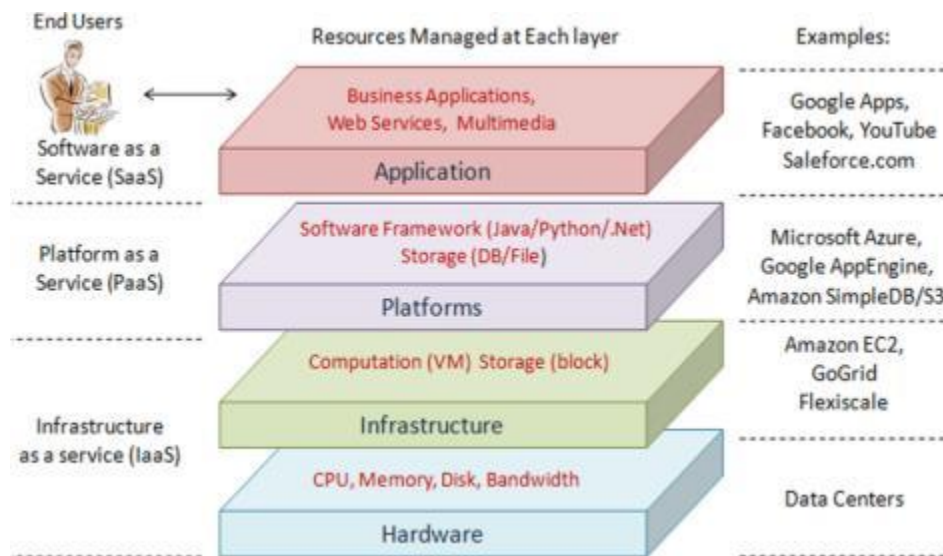
Εικόνα 1-1: Η αρχιτεκτονική του MCC

Γενικά το CC αναφέρεται σε ένα μεγάλων δυνατοτήτων κατακεντρωμένο δικτυακό σύστημα, το οποίο βασίζεται σε μια συστοιχία εξυπηρετητών (servers) μέσα στα data centers. Οι υπηρεσίες που παρέχει το Cloud ταξινομούνται βασιζόμενες σε μια λογική διαστρωμάτωσης ως εξής:

- **Data Centers layer.** Το επίπεδο αυτό παρέχει το hardware για την υποδομή του cloud. Πρόκειται για μία σειρά από servers διασυνδεδεμένους με υψηλών ταχυτήτων δίκτυα ώστε να παρέχουν υπηρεσίες πελατών.
- **IaaS (Infrastructure as a Service).** Το επίπεδο αυτό είναι ένα αυτοδιαχειριζόμενο μοντέλο για πρόσβαση, έλεγχο και απομακρυσμένη διαχείριση των υποδομών του προηγούμενου επιπέδου. Η δυνατότητα που παρέχεται στον τελικό χρήστη είναι να μπορεί να δεσμεύσει

προς χρήση επεξεργαστική ισχύ, αποθηκευτικά μέσα, δίκτυα και άλλους θεμελιώδεις υπολογιστικούς πόρους. Παραδείγματα IaaS Clouds αποτελούν τα Amazon Web Services (AWS), Cisco Metapod, Microsoft Azure, Google Compute Engine (GCE), Joyent κ.α.

- PaaS (Platform as a Service).** Το επίπεδο αυτό χρησιμοποιείται για την υποστήριξη της ανάπτυξης των εφαρμογών. Μέσω του πλαισίου αυτού ο τελικός χρήστης δεν διαχειρίζεται ούτε ελέγχει τη σχετική cloud υποδομή που συμπεριλαμβάνει τα δίκτυα, τους servers, τα λειτουργικά συστήματα ή τα αποθηκευτικά μέσα, αλλά έχει τον έλεγχο των εφαρμογών που έχουν αναπτυχθεί, και ενδεχομένως των παραμετροποιήσεων του περιβάλλοντος φιλοξενίας των εφαρμογών. Παραδείγματα PaaS αποτελούν τα Microsoft Azure, Google AppEngine, Amazon SimpleDB/S3, Apprenda κ.α.
- SaaS (Software as a Service).** Στο επίπεδο αυτό δίνεται η δυνατότητα στον τελικό χρήστη να χρησιμοποιεί τις εφαρμογές του παρόχου που τρέχουν σε μια cloud υποδομή. Οι εφαρμογές είναι προσβάσιμες από διάφορες client συσκευές μέσω ενός thin client interface, όπως ένα πρόγραμμα περιήγησης (Web browser) το οποίο επιτρέπει την πρόσβαση στον διακομιστή (server) των λογισμικών. Τα λογισμικά αποτελούν ιδιοκτησία του παρόχου. Ως αντάλλαγμα των παραπάνω διευκολύνσεων ο τελικός χρήστης καλείται να καταβάλει μίσθωμα για την χρήση τους. Παραδείγματα του επιπέδου αυτού αποτελούν τα Google Apps, Facebook, Youtube.



Εικόνα 1-2: Service-Oriented Αρχιτεκτονική στο CC

1.3 Περιορισμοί και Πλεονεκτήματα

Ο βασικός σκοπός του MCC είναι να παρέχει την κατάλληλη και τάχιστη μέθοδο στους χρήστες, ώστε να έχουν πρόσβαση και να ανακτούν τα δεδομένα από το cloud. Το CC είναι μια πολλά υποσχόμενη λύση για το Mobile Computing για πολλούς λόγους (π.χ κινητικότητα, φορητότητα και επικοινωνία) και χρησιμοποιείται για να ξεπεράσει τις δυσκολίες που υπάρχουν στο MC (π.χ διάρκεια μπαταρίας των φορητών συσκευών, επεξεργαστική ισχύ και αποθήκευση δεδομένων, αξιοπιστία) αποφέροντας μεγάλα πλεονεκτήματα στο MCC. Η μεγαλύτερη πρόκληση του MCC έγκειται στον εγγενή χαρακτήρα των φορητών συσκευών, αλλά και των ασύρματων δικτύων, όπως επίσης και στους περιορισμούς που υπάρχουν, καθιστώντας την σχεδίαση, τον προγραμματισμό και την υλοποίηση των εφαρμογών αρκετά πιο πολύπλοκες σε σύγκριση με το επίγειο υπολογιστικό νέφος. Οι κυριότεροι περιορισμοί του MCC είναι οι ακόλουθοι:

- **Περιορισμοί των φορητών συσκευών.** Παρόλο που οι φορητές συσκευές έχουν σημειώσει σημαντική πρόοδο ως προς τις δυνατότητές τους (CPU, μνήμη, αποθηκευτικοί χώροι, μέγεθος οθόνης, τεχνολογία ασυρμάτων, τεχνολογία αισθητήρων αλλά και των λειτουργικών συστημάτων), αντιμετωπίζουν προβλήματα στην επεξεργασία και υλοποίηση σύνθετων εφαρμογών όπως και στην ενεργειακή διαχείρισή τους. Σε σύγκριση των φορητών συσκευών με τους σταθερούς υπολογιστές παρατηρούμε αισθητές διαφορές μέχρι και 3 φορές στις επεξεργαστικές δυνατότητες, 8 φορές στην μνήμη, 5-10 φορές στις αποθηκευτικές δυνατότητες και 10 φορές στην απόδοση του δικτύου.
- **Ποιότητα της επικοινωνίας.** Σε σύγκριση με τα ενσύρματα δίκτυα που χρησιμοποιούν φυσικά μέσα μεταφοράς για να εγγυηθούν σταθερή και αδιάλειπτη επικοινωνία για την μεταφορά των δεδομένων, η ρυθμοαπόδοση σε ένα περιβάλλον MCC συνεχώς αλλάζει με αποτέλεσμα την μη εγγυημένη μεταφορά των δεδομένων, προκαλώντας αυξημένη καθυστέρηση. Αυτό οφείλεται κατά κύριο λόγο στην απομακρυσμένη από τους χρήστες πρόσβαση στα DC. Ενδεικτικό είναι ότι στα ασύρματα δίκτυα ο χρόνος καθυστέρησης μπορεί να φτάνει τα 200 ms, ενώ στα ενσύρματα μόλις 50 ms.
- **Διάρθρωση των εφαρμογών.** Λόγω των παραπάνω υπάρχει η ανάγκη να κατακερματίσουμε της εφαρμογές και να αξιοποιήσουμε το περιβάλλον και τις δυνατότητες του CC ώστε να τρέξουμε τμήματα των εφαρμογών που χρήζουν είτε μεγάλης επεξεργαστικής ισχύος είτε καταναλώνουν πολλή ενέργεια, με αποτέλεσμα να επηρεάσουμε την γενική απόδοση του συστήματος, αφού εξαρτόμαστε από τον χρόνο επεξεργασίας των δεδομένων μέσα στα DC και στις φορητές συσκευές, τον χρόνο απόκρισης του δικτύου, όπως επίσης τον χρόνο παράδοσης των δεδομένων.

Μέσω των εφαρμογών του MC που μεταφέρουμε στο cloud την επεξεργαστική και αποθηκευτική δυνατότητα των φορητών συσκευών, αποκτούμε τα παρακάτω πλεονεκτήματα σε συνδυασμό με τις δυνατότητες του CC.

- Επέκταση της διάρκειας της μπαταρίας των φορητών συσκευών.
- Αυξάνονται οι δυνατότητες αποθήκευσης και επεξεργασίας των δεδομένων.
- Αυξάνεται η αξιοπιστία και η ασφάλεια των δεδομένων.
- Δυναμική πρόβλεψη των πόρων που χρειάζεται ο χρήστης (dynamic provisioning).
- Επεκτασιμότητα των εφαρμογών (scalability).
- Μαζική εξυπηρέτηση χρηστών.
- Ευκολία στην ενσωμάτωση πολλαπλών υπηρεσιών από διαφορετικούς παρόχους.

1.4 Εφαρμογές Κινητής Νεφούπολογιστικής

Οι mobile εφαρμογές αποκτούν αυξανόμενο μερίδιο στην παγκόσμια αγορά ενσωματώνοντας τα πλεονεκτήματα του MCC. Στην ενότητα αυτή θα παρουσιαστούν ενδεικτικές εφαρμογές του MCC.

1.4.1 Mobile learning

Ο όρος m-learning είναι συνδυασμός του e-learning και της δυνατότητας για φορητότητα. Μπορούμε να εκμεταλλευτούμε το cloud για να αποθηκεύσουμε τα δεδομένα μας και να εκτελούμε τις περισσότερες εργασίες στο cloud μέσω των φορητών συσκευών. Με τη δυνατότητα αυτή μπορούμε να παρέχουμε καλύτερες υπηρεσίες στα άτομα που θα χρησιμοποιούν τις mobile learning εφαρμογές. Τέτοιες εφαρμογές μπορούν να θεωρηθούν οι διάφορες πλατφόρμες που χρησιμοποιούνται από τα πανεπιστήμια π.χ Coursera, όπως επίσης οι εφαρμογές του youtube και τα podcast. Οι προκλήσεις που αντιμετωπίζει το mobile learning είναι η έλλειψη σήματος, η διάρκεια της μπαταρίας της κάθε συσκευής και το μέγεθος της οθόνης και των πλήκτρων. Έτσι είναι πιο δύσκολο να γίνει σωστό mobile learning σε κινητό παρά σε κάποιο tablet. Οι κύριοι τομείς ανάπτυξης είναι η εκπαίδευση με βάση την τοποθεσία (location-based learning) και η εκπαίδευση με βάση τα κοινωνικά δίκτυα (social-networked mobile learning).

1.4.2 Mobile commerce

Το mobile commerce είναι ένα επιχειρηματικό μοντέλο όπου οι συναλλαγές γίνονται με χρήση κινητών συσκευών και εκπληρώνουν διαδικασίες που απαιτούν φορητότητα (mobile συναλλαγές και πληρωμές, αποστολή μηνυμάτων και αγορά εισιτηρίων). Τέτοιες εφαρμογές συναντάμε στους τομείς της οικονομίας, διαφήμισης και αγορών.

- Mobile financial applications: εφαρμογές του τραπεζικού συστήματος για διαχείριση χρηματοοικονομικών στοιχείων των πελατών (m-banking).
- Mobile advertising: εφαρμογές οι οποίες στέλνουν διαφημίσεις στο χρήστη λαμβάνοντας υπόψιν την τοποθεσία του και άλλες παραμέτρους.
- Mobile shopping: εφαρμογές για online shopping (Amazon, ebay κτλ.)

Οι mobile commerce εφαρμογές αντιμετωπίζουν διάφορες προκλήσεις όπως την έλλειψη σήματος, την ασφάλεια και την πολυπλοκότητα των ρυθμίσεων των συσκευών. Για το λόγο αυτό οι εφαρμογές ενσωματώνονται στο cloud για να αντιμετωπιστούν αυτά τα προβλήματα. Μια λύση είναι η χρήση μιας 3G mobile commerce πλατφόρμας που θα είναι βασισμένη στο cloud ώστε να εκμεταλλευτούμε τα πλεονεκτήματα του 3G δικτύου αλλά και της επεξεργαστικής ισχύς του cloud computing για την πιο γρήγορη επεξεργασία δεδομένων και την αύξηση της ασφάλειας[2].

1.4.3 Mobile healthcare

Η ενσωμάτωση του MCC στο χώρο της υγείας βοηθά στην παροχή ουσιαστικών αποτελεσμάτων μέσω της αποδοτικότερης μεταφοράς, αποθήκευσης, επεξεργασίας και εξασφάλισης των δεδομένων. Οι m-health εφαρμογές κάνουν χρήση των κινητών συσκευών για να συλλέξουν δεδομένα για τους ασθενείς, για την αποστολή πληροφοριών στο ιατρικό προσωπικό, σε ερευνητές αλλά και σε ασθενείς. Επίσης οι ιατροί μπορούν να παρακολουθούν ανά πάσα στιγμή σε πραγματικό χρόνο την κατάσταση των ασθενών. Ο τομέας του m-health γνωρίζει ραγδαία ανάπτυξη στις αναπτυσσόμενες χώρες, σε χώρες της Αφρικής όπου οι αποστάσεις είναι τεράστιες και το ιατρικό προσωπικό λίγο. Ενδεικτικά κάποιες επιπλέον χρήσεις των M-health εφαρμογών παρατίθενται:

- Ολοκληρωμένες υπηρεσίες παρακολούθησης της υγείας επιτρέπουν στους ασθενείς να παρακολουθούνται ανά πάσα στιγμή και οπουδήποτε μέσω ευρυζωνικών ασύρματων επικοινωνιών.
- Έξυπνο σύστημα διαχείρισης καταστάσεων έκτακτης ανάγκης μπορεί να διαχειριστεί και να συντονίζει τον στόλο των οχημάτων έκτακτης ανάγκης αποτελεσματικά και σε χρόνο κατά τη λήψη κλήσεων από ατυχήματα ή συμβάντα.
- Εφαρμογές για ανίχνευση του καρδιακού ρυθμού, την αρτηριακή πίεση και το επίπεδο του αλκοόλ ώστε να προειδοποιήσει την υγειονομική περίθαλψη

1.4.4 Mobile gaming

Το mobile gaming προσφέρει πολλά κέρδη σε εταιρίες και αναπτύσσετε ραγδαία. Στο mobile gaming οι διεργασίες που απαιτούν μεγάλη επεξεργαστική ισχύ όπως το graphic rendering

αναθέτονται στο cloud και μόνο οι διεργασίες που απαιτούν μικρή επεξεργαστική ισχύ γίνονται στην ίδια την συσκευή όπως το πάτημα πλήκτρων στη συσκευή. Πλατφόρμες που εκμεταλλεύονται την επεξεργαστική ισχύ του cloud είναι το Gaikai [3], Unity [4] και g-cluster [5]. Οι πλατφόρμες αυτές προσφέρουν in-home streaming παιχνιδιών σε διάφορες κινητές συσκευές μέσω του δικτύου του χρήστη, αλλά και cloud-based gaming. Έτσι επιτρέπεται σε συσκευές με μικρή επεξεργαστική ισχύ να τρέξουν παιχνίδια τα οποία η ίδια συσκευή δεν θα μπορούσε να παίξει. Ο βασικός τρόπος λειτουργίας είναι η ανάθεση όλων των διεργασιών στο cloud δηλαδή στους servers που φιλοξενούνται οι πλατφόρμες. Το παιχνίδι στην ουσία τρέχει στον υπολογιστή του cloud και γίνεται streaming στην συσκευή του χρήστη. Οι εντολές του χρήστη πάνε μέσω ίντερνετ στους servers, εκεί γίνεται η επεξεργασία και επιστρέφει στο χρήστη. Αυτό απαιτεί πολύ καλή και γρήγορη σύνδεση στο ίντερνετ και δεν είναι γνωστό ακόμα αν θα μπορέσει να γίνει στο μέλλον μέσω του δικτύου κινητής τηλεφωνίας.

1.4.5 Social networking

Οι εφαρμογές social networking είναι οι πιο διαδεδομένες μεταξύ των χρηστών. Τέτοιες εφαρμογές είναι εφαρμογές όπως το Facebook, Twitter, εφαρμογές επικοινωνίας όπως το Skype, Viber, Hangouts, WhatsApp αλλά και εφαρμογές ηλεκτρονικού ταχυδρομείου, όπως Yahoo mail, Gmail. Στις εφαρμογές αυτές τα περισσότερα δεδομένα αποθηκεύονται στο cloud και επιστρέφονται στο χρήστη όταν τα ζητήσει. Το cloud λόγω της μεγάλης επεξεργαστικής του ισχύς μπορεί να εκτελέσει διάφορες λειτουργίες πάνω στα δεδομένα αυτά όπως face recognition στις φωτογραφίες του χρήστη, φιλτράρισμα των δεδομένων κτλ. Αυτές οι εφαρμογές μπορούν να χωριστούν σε διάφορες κατηγορίες με βάση τις λειτουργίες που εκτελούν. Τα μοντέλα που υιοθέτησαν οι προγραμματιστές των εφαρμογών παρουσιάζονται παρακάτω.

- **Group texter.** Το μοντέλο αυτό επικεντρώνεται στην ικανότητα των εφαρμογών για αποστολή μηνυμάτων ταυτόχρονα σε μεγάλο αριθμό ατόμων.
- **Location-aware.** Το μοντέλο αυτό βασίζεται στην ικανότητα των εφαρμογών να διαχειρίζονται την τοποθεσία του χρήστη και να του προσφέρουν συμβουλές και αποτελέσματα αναζητήσεων με βάση το μέρος όπου βρίσκεται.
- **Social networker.** Το μοντέλο αυτό προσαρμόζει τις λειτουργίες των ιστοτόπων κοινωνικής δικτύωσης, όπως το Facebook, το Google+, παρέχοντας την δυνατότητα για αποστολή multimedia posts, κοινό διαμοιρασμό, instant messaging και δυνατότητα πραγματοποίησης φτηνών κλήσεων με ήχο και εικόνα.
- **Media share.** Το μοντέλο αυτό είναι προέκταση του μοντέλου group texter δίνοντας την δυνατότητα για μαζική αποστολή αρχείων ήχου, εικόνας και video. Εφαρμογές που ειδικεύονται σ' αυτές τις εργασίες είναι το Instagram, Vine, Youtube.

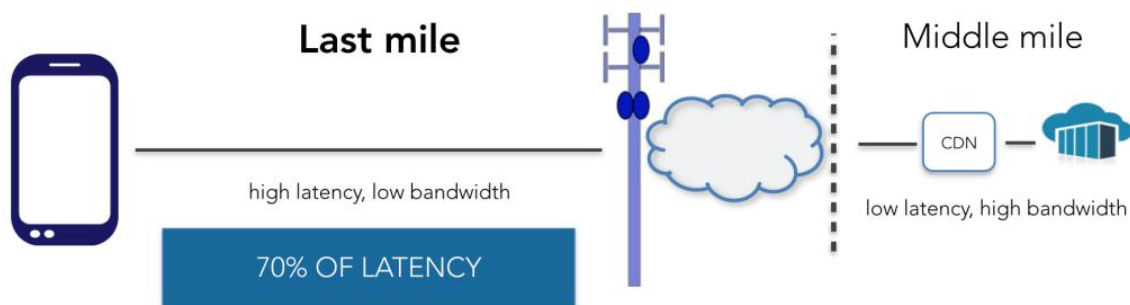
- **Social gaming.** Το μοντέλο αυτό δίνει τη δυνατότητα στους χρήστες να συναντώνται και να αλληλοεπιδρούν σε παιχνίδια. Εταιρία που δραστηριοποιείται και έχει αναπτύξει ανάλογη πλατφόρμα είναι η Zynga.

1.5 Καθυστέρηση μετάδοσης (delay)

Σε ένα διάχυτο υπολογιστικά περιβάλλον, οι κινητοί χρήστες όταν ζητάνε υπηρεσίες και πόρους χρειάζεται να έρθουν σε επικοινωνία με υποδομές, οι οποίες βρίσκονται μακριά από αυτούς εντός των υπολογιστικών νεφών. Πολλές φορές αντιμετωπίζουν προβλήματα συμφόρησης εξαιτίας του περιορισμένου εύρους ζώνης των ασύρματων ζεύξεων, προβλήματα δικτυακών αποσυνδέσεων, όπως επίσης και προβλήματα εξασθένησης του σήματος λόγω του μεταβαλλόμενου περιβάλλοντος που επικρατεί με αφορμή την κινητικότητα που τους χαρακτηρίζει. Αυτό έχει σαν αποτέλεσμα να προκαλείται καθυστέρηση (delay) στην προσπάθεια να επικοινωνήσουν με το cloud, επομένως η ποιότητα των υπηρεσιών μειώνεται σημαντικά. Είναι επιτακτική ανάγκη της τεχνολογίας να δώσει λύσεις για να μειωθεί ο χρόνος απόκρισης (latency), αλλά και απαίτηση των εταιριών που δραστηριοποιούνται στο χώρο.

Παρόλα τα πλεονεκτήματα που έχουμε με το MCC, η καθυστέρηση (delay) αλλά και η διακύμανση της καθυστέρησης (jitter) στο cloud, παράγοντες που επηρεάζουν σημαντικά εφαρμογές και δεδομένα ευαίσθητα στον χρόνο (φωνή, video), που σε μεγάλο βαθμό καταναλώνονται από κινητές συσκευές, είναι κρίσιμο θέμα στα MC, καθώς μπορεί να οδηγήσει σε δυσλειτουργία των cloud υπηρεσιών με αποτέλεσμα την μείωση των κερδών των εταιριών [6] από την απομάκρυνση των χρηστών.

Προκειμένου να αντιμετωπιστεί το πρόβλημα της καθυστέρησης, βασικό στόχο στα MC αποτελεί η μείωση του χρόνου απόκρισης που παρατηρείται ως επί το πλείστον στο κομμάτι του δικτύου που βρίσκεται κοντά στο χρήστη, ουσιαστικά στην ασύρματη ζεύξη μεταξύ του χρήστη και του σταθμού βάσης (BS) ή μεταξύ του χρήστη και του WiFi AP (Wireless last mile).



Εικόνα 1-3: Η καθυστέρηση στο MC

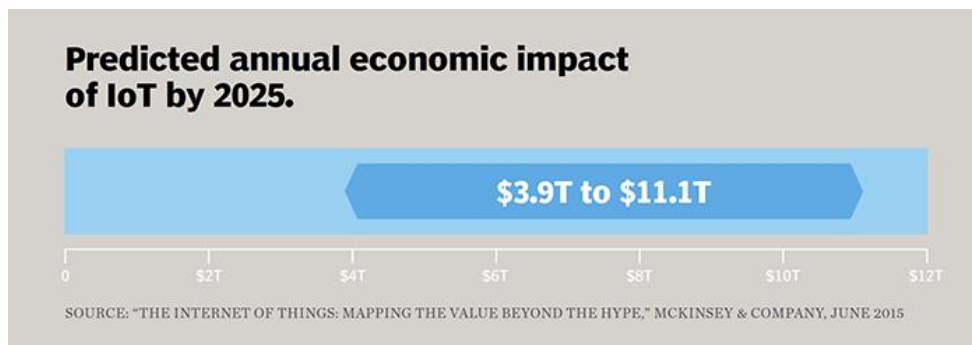
Στο υπόλοιπο κομμάτι του δικτύου (first mile και middle mile), η καθυστέρηση στο χρόνο απόκρισης οφείλεται είτε σε διασπορά των περιεχομένων γεωγραφικά είτε σε κακές πολιτικές δρομολόγησης (routing policies) και διορθώνεται με την ύπαρξη των δικτύων παροχής περιεχομένου (Content Delivery Networks-CDN). Τα CDNs διανέμουν το περιεχόμενο σε πολλαπλούς, γεωγραφικά διαφορετικούς, κόμβους και εξυπηρετούν έπειτα το περιεχόμενο από οποιοδήποτε κόμβο είναι πιο κοντά στον τελικό χρήστη, ελαχιστοποιώντας έτσι το ταξίδι πακέτων και αυξάνοντας την ταχύτητα και την αποδοτικότητα. Ανάλογα με την κυκλοφορία και τον αριθμό κόμβων, οι αλγόριθμοι του δικτύου επιλέγουν τις καλύτερες πολιτικές δρομολόγησης, ώστε να προσφέρουν τη βέλτιστη απόδοση και την αποφυγή συμφόρησης.

Στο last hop [7], σε επίπεδο δικτύου το πρόβλημα μπορεί να αντιμετωπιστεί με τη βελτίωση της απόδοσης των δικτύων ευρείας περιοχής (Wide Area Networks-WAN), ενώ σε επίπεδο πακέτων η αντιμετώπιση προσανατολίζεται στην τοποθέτηση του να φέρουμε το cloud πιο κοντά στον χρήστη.

Σε αυτήν ακριβώς την κατεύθυνση έχουν γίνει τον τελευταίο καιρό προσπάθειες από το χώρο της βιομηχανίας, όπως επίσης και τον ακαδημαϊκό χώρο, προκειμένου να γίνει ώθηση των δυνατοτήτων του CC προς την άκρη του δικτύου (Fog/Edge Computing, Mobile Edge Computing, Cloudlets). Οι βασικότερες από τις τεχνολογίες αυτές περιγράφονται συνοπτικά παρακάτω.

1.6 Fog Computing (FC)

Ο όρος Fog Computing (FC) αναφέρθηκε αρχικά από την Cisco (2011)[8] και βασίζεται στη φιλοσοφία των "νεφών", τα οποία με τη μορφή της "ομίχλης" (fog) βρίσκονται πιο κοντά στο έδαφος. Η τεχνολογία αυτή φέρνει το CC πιο κοντά στους χρήστες και συγκεκριμένα στα ασύρματα δίκτυα, προκειμένου να υποστηρίξει κατά κύριο λόγο το όραμα του Internet of Things[9].



Εικόνα 1-4: Η επίδραση του IoT στην Οικονομία μέχρι το 2025

Το FC είναι ένα σύστημα οριζόντιας αρχιτεκτονικής που διανέμει τους πόρους και τις υπηρεσίες επεξεργασίας, αποθήκευσης, ελέγχου και δικτύωσης σε οποιοδήποτε σημείο κατά μήκος της συνέχειας Cloud to Things (OpenFog Consortium 2015) [10]. Ένας μεγάλος αριθμός από διάχυτες, ετερογενείς (ασύρματες και πολλές φορές αυτόνομες) και αποκεντροποιημένες συσκευές έχουν τη δυνατότητα να επικοινωνούν και ενδεχομένως να συνεργάζονται μεταξύ τους ή με το δίκτυο για να εκτελέσουν διεργασίες αποθήκευσης και επεξεργασίας χωρίς την ανάγκη παρέμβασης τρίτου σε ένα κλειστό περιβάλλον [11].

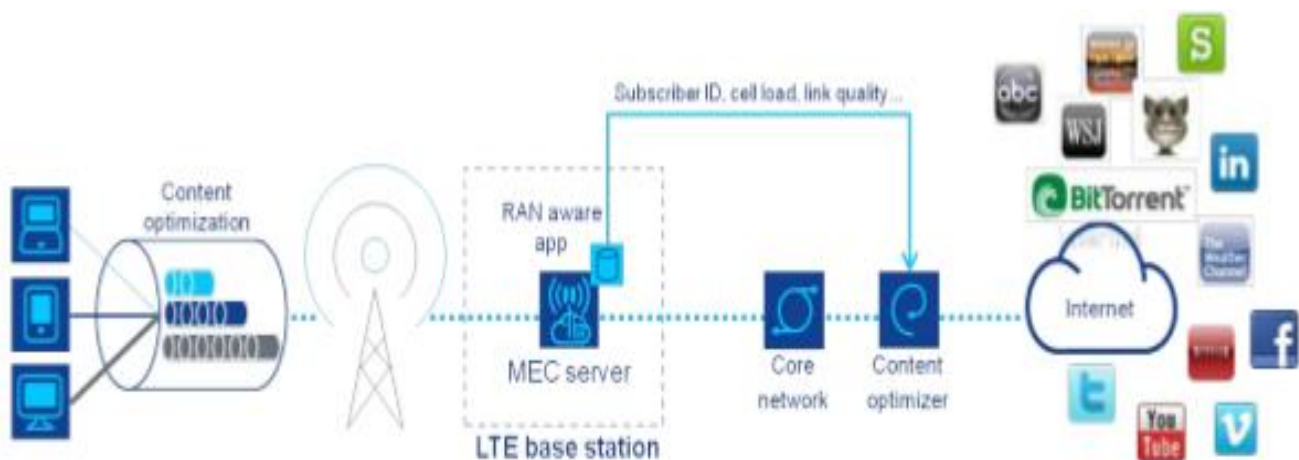
Οι FC κόμβοι (FCNs) τοποθετούνται μακριά από τα κυρίως cloud DC στο last hop επιτρέποντας χαμηλό και προβλέψιμο latency. Η κινητικότητα των συσκευών υποστηρίζεται με ανακατεύθυνση της εφαρμογής στην κινητή συσκευή και εν συνεχεία σύνδεση της συσκευής με διαφορετικό κόμβο σε μικρότερη απόσταση, σε περιπτώσεις που μία συσκευή απομακρυνθεί από την εμβέλεια του FCN, στον οποίο είναι συνδεδεμένη.

Στον παρακάτω πίνακα παρατίθενται τα βασικά χαρακτηριστικά του Fog Computing σε σχέση με το Cloud Computing [12][13].

Παράμετροι	Cloud Computing	Fog Computing
Θέσεις κόμβων διακομιστών	Εντός του Διαδικτύου	Στο άκρο του τοπικού δικτύου
Απόσταση μεταξύ Client and server	Πολλαπλά βήματα	Ένα/Μερικά βήματα
Καθυστέρηση	Υψηλή	Χαμηλή
Ασφάλεια	Αδυναμία ελέγχου τοπικά	Τοπικά Ελέγξιμο
Επίγνωση Θέσης	Όχι	Ναι
Ευαισθησία	Μεγαλύτερη πιθανότητα	Μικρότερη πιθανότητα
Γεωγραφική Κατανομή	Συγκεντρωτικά	Πυκνά και Κατανεμημένα
Αριθμός των κόμβων των διακομιστών	Λίγοι	Μεγάλος
Αλληλεπιδράσεις πραγματικού χρόνου	Δεν υποστηρίζεται πλήρως	Υποστηρίζεται
Συνδεσιμότητα στα άκρα του δικτύου	Ενσύρματα/Ασύρματα	Κυρίως Ασύρματα
Κινητικότητα	Περιορισμένη υποστήριξη	Υποστηρίζεται
Έλεγχος	Πλήρης	Μερικός
Πρόσβαση Υπηρεσιών	Μέσω του δικτύου κορμού	Στα άκρα σε φορητή συσκευή
Διαθεσιμότητα	99.99%	Εξαιρετικά Ευμετάβλητη
#χρηστών/συσκευών	Δεκάδες/Εκατοντάδες εκατομμύρια	Δεκάδες Δισεκατομμύρια
Τιμή ανά διακομιστή	\$ 1500-3000	\$ 50-200
Παραγωγή Δεδομένων	Άνθρωπος	Συσκευές
Αφετηρία Γέννησης Δεδομένων	Κεντρική Τοποθεσία	Οπουδήποτε
Κατανάλωση περιεχομένου	Τερματικές Συσκευές	Οπουδήποτε
Λογισμικό Εικονικής υποδομής	Κεντρικοί Εταιρικοί Διακομιστές	Συσκευές των χρηστών

1.7 Mobile Edge Computing (MEC)

Το MEC αναφέρθηκε πρώτη φορά από το Ευρωπαϊκό Ινστιτούτο Τηλεπικοινωνιακών Προτύπων(ETSI) το 2014 [14]. Στόχος αυτού του μοντέλου είναι να οδηγήσει τις CC δυνατότητες κοντά στο δίκτυο κινητής τηλεφωνίας (Radio Access Networks) ενσωματώνοντάς τις στις τεχνολογίες πρόσβασης 4G/5G και υποστηρίζεται από εταιρίες όπως Nokia, Huawei, IBM, Intel, NTT DoCoMo, Vodafone. Το RAN edge προσφέρει ένα περιβάλλον με εξαιρετικά χαμηλό latency και μεγάλο εύρος ζώνης (bandwidth), ενώ παράλληλα δίνει απευθείας πρόσβαση σε πραγματικού χρόνου RAN πληροφορίες, όπως οι συντεταγμένες του χρήστη και ο φόρτος της κυψέλης, οι οποίες μπορούν να προσφέρουν context-related υπηρεσίες μέσω εφαρμογών και άλλων υπηρεσιών. Η τεχνολογία αυτή δίνει τη δυνατότητα στους τηλεπικοινωνιακούς παρόχους να μπουν δυναμικά στο χώρο αυξάνοντας τα έσοδά τους και κάνοντας συνεργασίες με παρόχους υπηρεσιών και περιεχομένου, με εταιρίες που παρέχουν εφαρμογές και υπηρεσίες οπτικο ακουστικού υλικού μέσω του διαδικτύου (Over-The-Top Content) (π.χ Netflix, Youtube)[15] και με ανεξάρτητους προμηθευτές λογισμικού (independent software vendors), όπως λογισμικό για αγοραπωλησίες ακινήτων, διαχείριση στόλου οχημάτων, την παρακολούθηση των ιατρικών στοιχείων, το λιανικό εμπόριο κ.α.)



Εικόνα 1-5: Αρχιτεκτονική MEC [16]

Η τεχνολογία του MEC είναι φυσική εξέλιξη της ραγδαίας ανάπτυξης των κινητών σταθμών βάσης και της σύγκλισης της πληροφορικής με τις τηλεπικοινωνίες, δημιουργώντας νέους επιχειρηματικούς τομείς και υπηρεσίες για καταναλωτές, όπως επίσης και εταιρικούς πελάτες. Διάφοροι τομείς και περπτώσεις που κάνει χρήση η τεχνολογία αυτή είναι το Internet of Things, οι αναλύσεις βίντεο, η παράδοση περιεχομένου επαυξημένης πραγματικότητας,

βελτιστοποιημένη τοπική διανομή περιεχομένου και προσωρινή αποθήκευση των δεδομένων. Με μοναδικό τρόπο επιτρέπει σε εφαρμογές λογισμικού να αξιοποιήσουν τοπικά και σε πραγματικό χρόνο περιεχόμενο και πληροφορίες για την κατάσταση στο δίκτυο τοπικής πρόσβασης.

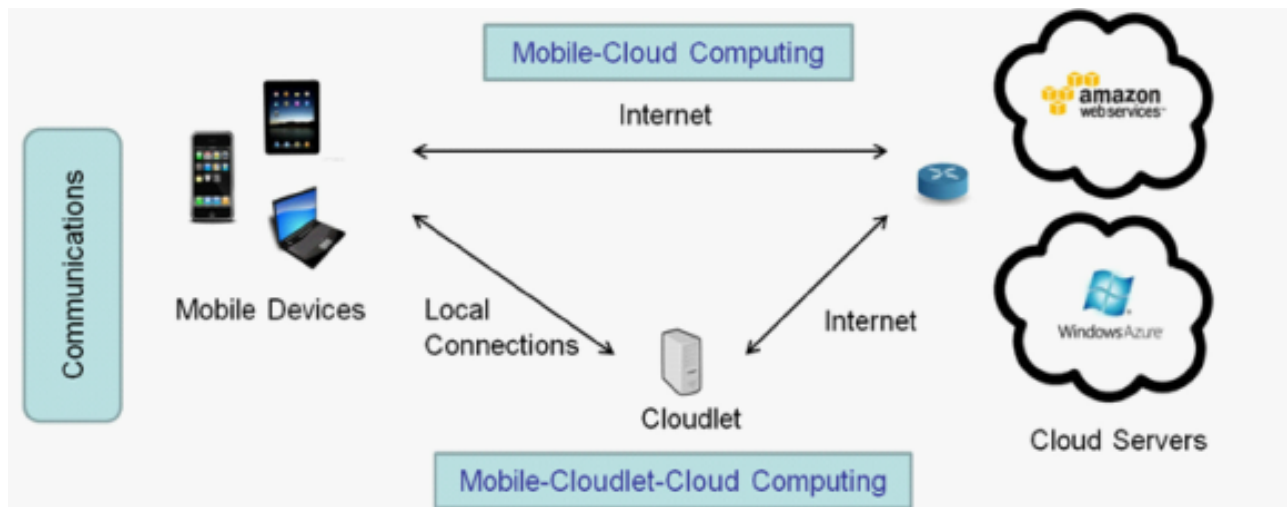
Οι πιθανοί τρόποι που μπορούμε να υλοποιήσουμε την τεχνολογία είναι:

- Μέσα στα LTE macro base station [eNB site]
- Μέσα στην multi-technology (3G/LTE)(small cells), τα οποία μπορεί να βρίσκονται είτε σε εσωτερικό χώρο είτε σε εξωτερικό [cell aggregation site]
- Μέσα στα 3G Radio Network Controller [RNC site]

1.8 Cloudlets/mDc

Τα Cloudlets αναπτύχθηκαν από το πανεπιστήμιο Carnegie Mellon [17][18] (2013) και αναπαριστούν την μεσαία βαθμίδα από μια τριπλή ιεραρχικά βαθμίδα: “mobile device - cloudlet - cloud” και έχουν ως στόχο να φέρουν το cloud πιο κοντά στο χρήστη. Αντίστοιχη πρόταση έκανε το τμήμα ερευνών της Microsoft [19], αναφέροντας την έννοια του micro DC σαν επέκταση των σημερινών hyperscale cloud DC (Microsoft Azure, Amazon Web Services). Το μοντέλο αυτό υποστηρίζεται επίσης από τους Intel, Huawei, Vodafone και από τον οργανισμό OpenEdgeComputing.org [20].

Τα Cloudlets ή microDC αποτελούν ένα ενδιάμεσο επίπεδο στην υποδομή του MCC, τα οποία παρεμβάλλονται ανάμεσα στην επικοινωνία των κινητών συσκευών με το cloud. Αποτελούνται από έναν ή περισσότερους διακομιστές μικρότερης κλίμακας δυνατοτήτων (το πολύ μέχρι 10) σε σχέση με τα παραδοσιακά DC και σχεδιάστηκαν για content caching και διάσπαση TCP connections, ώστε να επιλύσουν προβλήματα βασισμένα στο latency. Τα cloudlet είναι CDNs κόμβοι που μπορούν να βελτιώσουν την απόδοση των μηχανών αναζήτησης, των εργαλείων για παραγωγικότητα στο γραφείο, των video & audio conference όπως επίσης και μελλοντικών cloud υπηρεσιών.



Εικόνα 1-6: Η αρχιτεκτονική Mobile-Cloudlet-Cloud Computing [21]

Βασίζονται στην τεχνολογία των υπολογιστικών νεφών, ενθυλακώνουν offload code από τις φορητές συσκευές στα εικονικά μηχανήματα και μπορούν να έχουν συγκεκριμένο ρόλο και λειτουργικότητα. Χαρακτηρίζονται για την δυνατότητά τους να κρατάνε cache δεδομένα από τις φορητές συσκευές και το υπολογιστικό νέφος. Μετά την εγκατάστασή τους είναι αυτοεξυπηρετούμενα, χαρίζοντας λογική εγγύτητα στις συσκευές. Λόγω της τοποθεσίας που εγκαθίστανται (last mile) προσφέρουν χαμηλή καθυστέρηση και εύρος ζώνης που αναλογεί στο τοπικό δίκτυο πρόσβασης. Η μείωση του Latency είναι ένα από τα κυριότερα πλεονεκτήματα του Cloudlet. Εξυπηρετεί με static περιεχόμενο αμέσως τα αιτήματα που δέχεται, σταματάει την κωδικοποιημένη σύνδεση μεταξύ server και browser (SSL termination) και λειτουργεί σαν proxy χωρίζοντας την σύνδεση TCP στα δύο.

Η σύνδεση με το cloud είναι πολύ καλή, έχει επαρκείς πόρους (CPU, RAM, HDD) για εντατική επεξεργασία, ενώ δεν υπάρχει περιορισμός από αποθέματα ενέργειας. Παράλληλα εξοικονομεί μπαταρία στις κινητές συσκευές με τη δυνατότητα επεξεργασίας δύσκολων υπολογισμών αποφορτίζοντας τα Data Centers. Εξοικονομεί εύρος ζώνης με την συμπίεση του περιεχομένου, ενώ παρακολουθεί τις υπηρεσίες στο διαδίκτυο. Υπάρχει αξιοπιστία στη σύνδεση και προστατεύει εναντίον επιθέσεων άρνησης εξυπηρέτησης (Denial of Service attacks-Dos).

Cloudlet vs Cloud

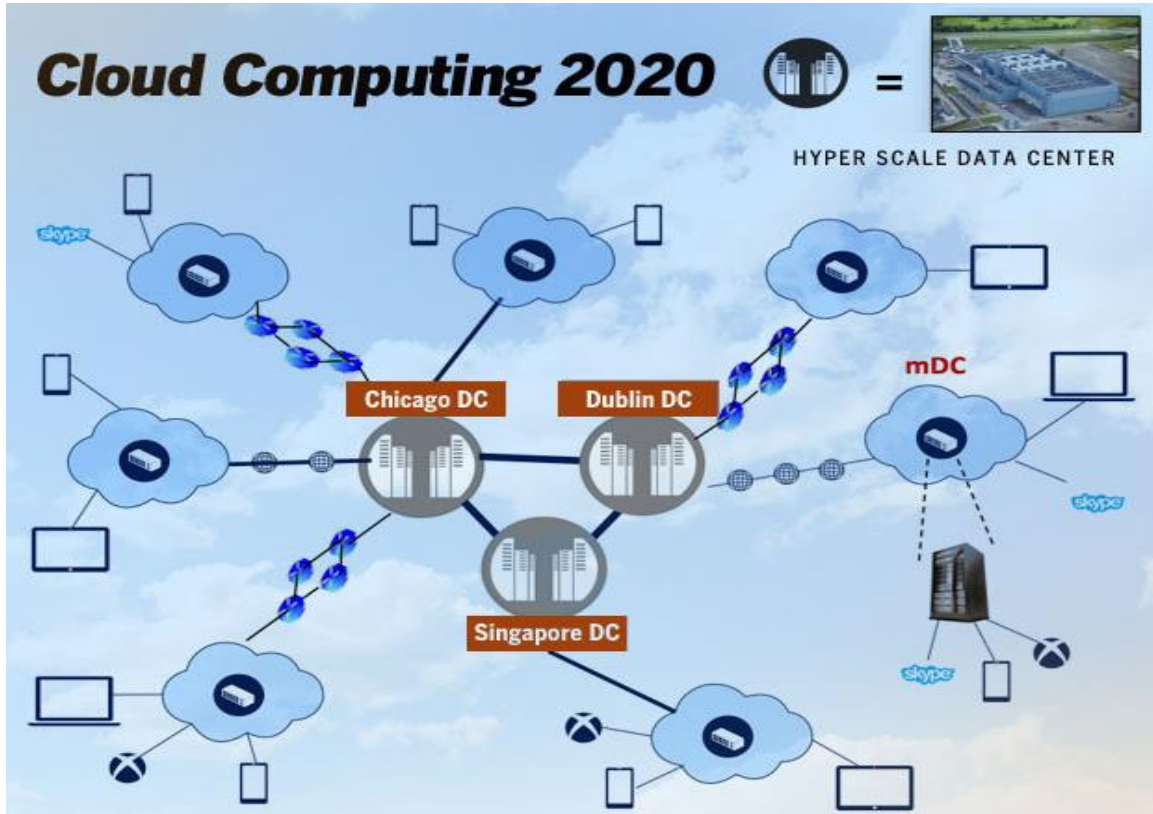
	Cloudlet	Cloud
State	Only soft state	Hard and Soft state
Management	Self-managed	Professionally administered
Environment	"Data Center in a box" at business premises	Machine room with power conditioning and cooling
Ownership	Decentralized ownership by local business	Centralized ownership by Amazon, Yahoo etc
Network	LAN latency/bandwidth	Internet latency/bandwidth
Sharing	Few users at a time	100s-1000s of users at a time



Εικόνα 1-7: Hyperscale DC ανά τον κόσμο

Η κατασκευή μεγάλης κλίμακας DC υψηλών δυνατοτήτων (Hyper Scale DC) πιο κοντά στο χρήστη θα ήταν ιδανική, αλλά απαιτούνται μεγάλα κεφάλαια για την υλοποίησή τους, ενώ ταυτόχρονα έχουν πολύ αυξημένο λειτουργικό κόστος (OpEx). Μια πιο έξυπνη προσέγγιση είναι η κατασκευή μιας πιο εκτεταμένης υποδομής από μικρότερης κλίμακας δυνατοτήτων DC (mDC cloudlets) τα οποία θα καταμεριστούν παντού, ανάλογα των αναγκών των χρηστών. Ο συνδυασμός των Hyper-scale DC και των mDC cloudlets θα προσφέρει την απαιτούμενη ώθηση σε επερχόμενες τεχνολογίες και εφαρμογές, όπως οι Εικονικοί Προσωπικοί Βοηθοί (Virtual Personal Assistants), η αύξηση της ανθρώπινης νοητικής (Human Augmentation), συστήματα

μικροηλεκτρομηχανικής (Smart Dust [22]) και η αναγνώριση αντικειμένων (Object Recognition) [23].



Εικόνα 1-8: Hyper Scale Data Center

2 Cloud Simulators

Όπως αναφέρθηκε στο προηγούμενο Κεφάλαιο, το CC είναι ένα πρότυπο που επιτρέπει σε άτομα και οργανώσεις να καταναλώσουν πόρους με την ελάχιστη ανθρώπινη παρέμβαση. Σύμφωνα με πρόσφατες έρευνες, ένας μεγάλος αριθμός εταιριών έχουν ήδη υλοποιήσει την μετάπτωση των συστημάτων τους στο cloud περιβάλλον. Παρά της μεγάλης αποδοχής που τυγχάνει από την κοινότητα της πληροφορικής, η τεχνολογία βρίσκεται σε πρωταρχικό στάδιο, οπότε υπάρχουν πολλές προκλήσεις όπως η αυτοματοποιημένη παροχή υπηρεσιών (automated service provisioning), η μετανάστευση των Εικονικών Μηχανών (Virtual Machines) (VM migration), η ενοποίηση των διακομιστών (server consolidation), η διαχείριση της ενέργειας (energy management), η ανάλυση της κίνησης (traffic analysis), η προστασία των δεδομένων (data security), καθώς και το λογισμικό που τα πλαισιώνει (software frameworks), το οποίο χρειάζεται περαιτέρω έρευνα για να σταθεροποιηθεί.

Η διεξαγωγή των ερευνών σε πραγματικό περιβάλλον cloud από μικρές εταιρίες ή ερευνητές φαντάζει ακατόρθωτη λόγω των δαπανών που συνδέονται με την εγκατάσταση και λειτουργία ενός cloud, όπως επίσης δεν είναι δυνατή η επαναλαμβανόμενη εκτέλεση πειραμάτων συγκριτικής αξιολόγησης σε περιβάλλοντα cloud πραγματικού χρόνου. Η λύση σε αυτό είναι η χρησιμοποίηση ενός προσομοιωτή, ο οποίος θα μπορέσει να προσομοιώσει ένα πραγματικό περιβάλλον cloud βοηθώντας την μοντελοποίηση των cloud εφαρμογών με αποτέλεσμα την ευκολία της ανάλυσης των αποτελεσμάτων. Μέχρι σήμερα έχουν αναπτυχθεί και χρησιμοποιούνται δεκάδες προσομοιωτές οι οποίοι διαφέρουν μεταξύ τους σε ποικίλα χαρακτηριστικά. Παρακάτω παρουσιάζονται οι πιο διαδεδομένοι εξ αυτών.

2.1 CloudSim

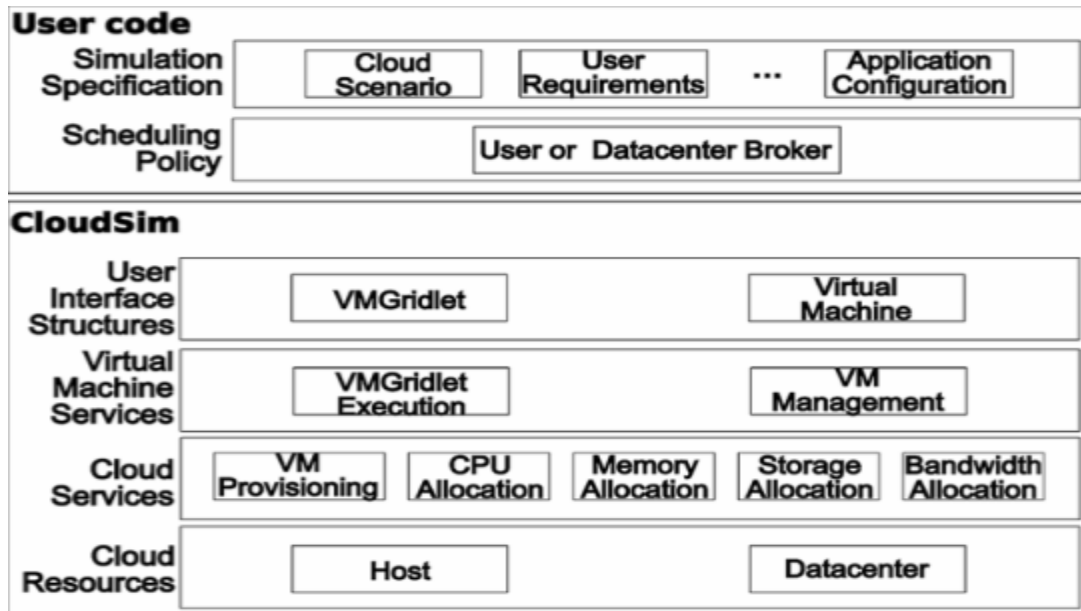
Το CloudSim είναι το πιο δημοφιλές εργαλείο προσομοίωσης που υπάρχει για μοντελοποίηση και προσομοίωση ενός CC περιβάλλοντος. Αναπτύχθηκε από τα εργαστήρια GRIDS του Πανεπιστημίου της Μελβούρνης, είναι ανοιχτού κώδικα και υλοποιήθηκε με τη χρήση της Java, ενώ βασίζεται στον πυρήνα του GridSim.

Βασικά Χαρακτηριστικά

- Υποστηρίζει την μοντελοποίηση και προσομοίωση για μεγάλης κλίμακας υπολογιστικά περιβάλλοντα.
- Αυτοδύναμη πλατφόρμα για μοντελοποίηση clouds, service brokers, provisioning και allocation policies.
- Υποστηρίζει την προσομοίωση των δικτυακών ζεύξεων μεταξύ των στοιχείων της προσομοίωσης.
- Διευκολύνει την προσομοίωση υβριδικών cloud περιβαλλόντων που περιέχουν πόρους ταυτόχρονα από δημόσιους και ιδιωτικούς τομείς.

- Παρέχει μια μηχανή εικονικοποίησης που βοηθά στην δημιουργία και διαχείριση πολλαπλών ανεξάρτητων και εξαρτημένων εικονικών υπηρεσιών σε ένα κόμβο επεξεργασίας δεδομένων.
- Ευελιξία στην εναλλαγή μεταξύ space shared και time shared κατανομής των πυρήνων επεξεργασίας σε εικονικές υπηρεσίες.

Η αρχιτεκτονική του προσομοιωτή CloudSim παρουσιάζεται στην παρακάτω εικόνα [23].

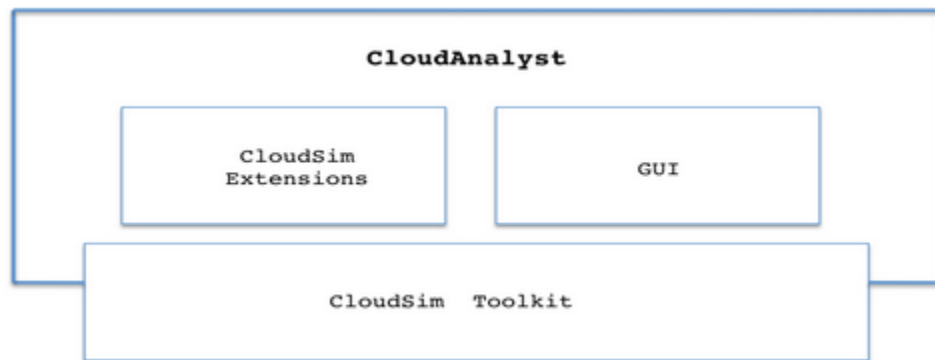


Εικόνα 2-1: Η αρχιτεκτονική του CloudSim

2.2 CloudAnalyst

Το CloudAnalyst είναι βασισμένο πάνω στο CloudSim και αναπτύχθηκε για την αξιολόγηση της απόδοσης των μεγάλης κλίμακας κατανεμημένων cloud εφαρμογών με μεγάλο φόρτο εργασιών από πολλούς χρήστες και είναι γεωγραφικά κατανεμημένα σε διάφορα κέντρα δεδομένων. Με τη χρήση του προσομοιωτή CloudAnalyst οι προγραμματιστές cloud εφαρμογών μπορούν να αξιολογήσουν την κατανομή των πόρων μεταξύ των data center, να επιλέξουν το βέλτιστο αριθμό Data Center για να εξυπηρετήσουν συγκεκριμένα αιτήματα και να μειώσουν το κόστος που σχετίζεται με αυτά. Το εργαλείο προσομοίωσης αυτό θεωρείται μοναδικό, λόγω της γραφικής παρουσίασης των αποτελεσμάτων και της ευκολίας στην επανάληψη των πειραμάτων. Στο επόμενο κεφάλαιο θα αναφερθούμε πιο διεξωδικά, αναλύοντας εκτενέστερα τις τεχνολογίες που ενσωματώνει και τα πλεονεκτήματα που το διακρίνει.

Η αρχιτεκτονική του προσομοιωτή CloudAnalyst παρουσιάζεται στην εικόνα που ακολουθεί.



Εικόνα 2-2: Η αρχιτεκτονική του CloudAnalyst

2.3 GreenCloud

Το GreenCloud είναι ένας προσομοιωτής που βασίζεται σε ανταλλαγή πακέτων και αναπτύχθηκε σε επέκταση του NetworkSimulator NS2 για περιβάλλοντα με ενεργειακό προσανατολισμό. Σχεδιάστηκε για να μπορεί να υπολογίσει την κατανάλωση ενέργειας σε οποιοδήποτε κομμάτι των DC, όπως switch, GW, link, καθώς και της επικοινωνίας μεταξύ των επιπέδων των πακέτων. Γλώσσες προγραμματισμού είναι οι C++ και OTcl. Χρειάζεται πολύ ώρα για να προσομοιώσει ένα μοντέλο καταναλώνοντας πολύ μνήμη. Είναι ιδανικός για να υπολογίζει τις ενεργειακές καταναλώσεις εντός των Data Center.

2.4 iCanCloud

Το iCanCloud σχεδιάστηκε βάση των μειονεκτημάτων των CloudSim και GreenCloud και βασίζεται πάνω στο SIMCAN. Το χαρακτηριστικό που το ξεχωρίζει είναι ότι μπορούμε να τροποποιήσουμε τον πυρήνα της κλάσης του hypervisor. Βασική γλώσσα προγραμματισμού είναι η C++ και προσφέρει μια γρήγορη, ελαστική και επεκτάσιμη δυνατότητα στο χρήστη για να μπορεί να αποφασίσει το κόστος των μηχανημάτων που θα χρειαστεί για να υλοποιήσει την cloud εφαρμογή.

2.5 SmartSim

Ο SmartSim είναι ο πρώτος προσομοιωτής που αναπτύχθηκε για να μπορεί να μοντελοποιεί και να προσομοιώνει εφαρμογές για mobile cloud computing. Υποστηρίζει το σύστημα και τη

συμπεριφορά μιας έξυπνης συσκευής (SMD) όπως μπορεί να μοντελοποιήσει την επεξεργαστική ισχύ της εφαρμογής, μνήμη που χρειάζεται, πόροι που μπορεί να χρειαστούν (provision), πόροι που χρειάζονται για επεξεργασία, αξιολόγηση χρήσης, όπως επίσης υποστηρίζει πολιτικές διαχείρισης δυναμικής επεξεργασίας. Παρόλα τα πλεονεκτήματα δεν έχει γραφικό περιβάλλον κάτι που τον καθιστά δύσχρηστο.

2.6 MR-CloudSim

Ο MR-CloudSim βασίζεται πάνω στον CloudSim και προσφέρει δυνατότητα για προσομοίωση διεργασιών MapReduce για υποστήριξη εφαρμογών BigData επεξεργασίας. Δεν υποστηρίζει επεξεργασία αρχείων, κόστος λειτουργίας και χρόνο λειτουργίας των εφαρμογών.

2.7 SimIC

Ο SimIC είναι ένας προσομοιωτής διακριτών γεγονότων και βασίζεται πάνω στο πακέτο της SimJava. Αναπτύχθηκε για να μπορεί να μοντελοποιεί και να ανταλλάζει υπηρεσίες από διαφορετικά cloud data center. Κανένα από τα παραπάνω δεν μπορούσε να προσομοιώνει αποδοτικά σενάρια με συνεργασία πολλών cloud ταυτόχρονα. Ο SimIC έχει στόχο την διαλειτουργικότητα, ευελιξία και την ελαστικότητα των υπηρεσιών καθώς μας εισάγει την έννοια της ανομοιογένειας πολλαπλών cloud χαρακτηριστικών. Χρησιμοποιεί τον αλγόριθμο Inter-Cloud Meta Scheduling (ICMS) που βασίζεται σε αρκετά καταναμημένα χαρακτηριστικά.

2.8 Κριτήρια Αξιολόγησης

Η επιλογή του κατάλληλου εργαλείου είναι μια κρίσιμη απόφαση για τον ερευνητή για την αποδοτικότερη έρευνα του εκάστοτε μοντέλου cloud. Παρακάτω γίνεται ανάλυση διάφορων χαρακτηριστικών βάση των οποίων θα γίνει και η τελική επιλογή του εργαλείου που θα χρησιμοποιήσουμε.

- **Base Platform:** Πολλοί cloud προσομοιωτές αναπτύχθηκαν με βάση υφιστάμενα πλαίσια προσομοίωσης. Τα χαρακτηριστικά των προηγούμενων συστημάτων ενσωματώνονται στο νέο.
- **Availability:** Πρέπει να γνωρίζουμε ότι οι CS είναι είτε εμπορικοί είτε ανοιχτού κώδικα. Υπάρχουν κάποιοι ανοιχτού κώδικα οι οποίοι δεν είναι ελεύθεροι για κατέβασμα.
- **Programming Language:** Είναι σημαντικό να γνωρίζουμε σε ποια γλώσσα έχει αναπτυχθεί ο κώδικας γιατί τις περισσότερες των περιπτώσεων χρειάζεται να παρέμβουμε στο framework.

- **Cost Modelling:** Το χαρακτηριστικό αυτό μας ενημερώνει αν ο προσομοιωτής μπορεί να υπολογίζει το κόστος της εφαρμογής ή αν μπορούμε να εισάγουμε την δική μας πολιτική για τον υπολογισμό του.
- **Graphical User Interface:** Η παρουσία γραφικού περιβάλλοντος βοηθά στον αποτελεσματικότερο και πιο απλό τρόπο προσομοίωσης.
- **Communication Model:** Κάποιοι ενδιαφέρονται για τα κόστη μέσα στα DC. Π.χ χρόνος χρειάζεται για τα πακέτα ή τα μηνύματα να φτάσουν από τον αρχικό εξοπλισμό ή διαδικασία μέχρι το τελικό εξοπλισμό ή διαδικασία.
- **Simulation Time:** Το χαρακτηριστικό αυτό μας δηλώνει πόσο χρειάζεται για το προσομοιωτή για να εξάγει το τελικό αποτέλεσμα. Όσο περισσότερο χρόνο χρειάζεται τόσο μη αποδοτικός είναι.
- **Energy Modelling:** Τελευταία υπάρχει μεγάλο ενδιαφέρον για τις καταναλώσεις σε ενέργεια των DC και της θερμοκρασίας τους.
- **Federation Policy:** οι cloud εφαρμογές μπορούν να τρέξουν σε ανομοιογενή clouds διαφορετικών οργανισμών.

Θα παραθέσουμε ένα πίνακα που παρουσιάζει συγκριτικά τους πιο διαδεδομένους Cloud Simulators βάση των πιο σημαντικών κριτηρίων αξιολόγησης.

Simulator	Base Platform	Availability	Prog. Language	Cost Model	GUI	Comm. Model	Simulation Time	Energy Model	Federation Model
CloudSim	SimJava	Open Source	Java	Yes	No	Limited	Seconds	Yes	Yes
CloudAnalyst	CloudSim	Open Source	Java	Yes	Yes	Limited	Seconds	Yes	Yes
GreenCloud	NS2	Open Source	C++, OTcl	No	Limited	Full	Minutes	Yes	No
iCanCloud	SIMCAN	Open Source	C++	Yes	Yes	Full	Seconds	No	No
Network CloudSim	CloudSim	Open Source	Java	Yes	No	Full	Seconds	Yes	Yes
EMUSIM	CloudSim, AEF	Open Source	Java	Yes	No	Limited	Seconds	Yes	No
GroudSim	-	Open Source	Java	No	Limited	No	Seconds	No	No
DCSim	-	Open Source	Java	Yes	No	No	Minutes	No	No
MR-CloudSim	CloudSim	Not Available	Java	Yes	No	Limited	-	Yes	Yes
SmartSim	CloudSim	Open Source	Java	Yes	No	Limited	Seconds	Yes	Yes
SimIC	SimJava	Open Source	Java	Yes	No	Limited	Seconds	Rough	Yes
SPECI	SimKit	Open Source	Java	No	No	Limited	Seconds	Rough	No
Dynamic CloudSim	CloudSim	Open Source	Java	Yes	No	Limited	Seconds	Yes	Yes
CloudSimSDN	CloudSim	Open Source	Java	Yes	Yes	Full	Seconds	Yes	Yes
secCloudSim	iCanCloud	Not Available	C++	Yes	Yes	Full	Seconds	No	No
CEPSim	CloudSim	Not Available	Java	Yes	Yes	Limited	Seconds	Yes	Yes
PICS	-	Open Source	Python	Yes	No	No	Seconds	No	No

Εικόνα 2-3: Συγκριτικός πίνακας προσομοιωτών Cloud. Πηγή: Suryateja, Pericherla S. "A Comparative Analysis of Cloud Simulators." International Journal of Modern Education and Computer Science (IJMECS) 8.4 (2016): 64.

Για λόγους ευχρηστίας και εύκολης επεκτασιμότητας, για τις προσομοιώσεις που περιγράφονται στις επόμενες ενότητες της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκε ως εργαλείο προσομοίωσης το λογισμικό Cloud Analyst. Η γραφική αναπαράσταση των αποτελεσμάτων το ξεχωρίζει έναντι των υπολοίπων και η ευκολία στην αλλαγή των παραμέτρων που χρησιμοποιούνται πραγματοποιώντας διαδοχικές μετρήσεις, μας βοηθά να συγκεντρωθούμε στα εξαγόμενα αποτελέσματα για τους λόγους της προσομοίωσης, παρά να επικεντρωθούμε σε τεχνικές δυσκολίες προγραμματισμού. Στο επόμενο κεφάλαιο θα μιλήσουμε πιο διεξοδικά για το εργαλείο προσομοίωσης CloudAnalyst.

3 CloudAnalyst

Όπως περιγράφηκε και στην προηγούμενη ενότητα, το CloudAnalyst είναι λογισμικό προσομοίωσης της τεχνολογίας cloud computing. Είναι ανοιχτού κώδικα και έχει υλοποιηθεί με την γλώσσα προγραμματισμού Java SE.

3.1 Πλεονεκτήματα χρήσης

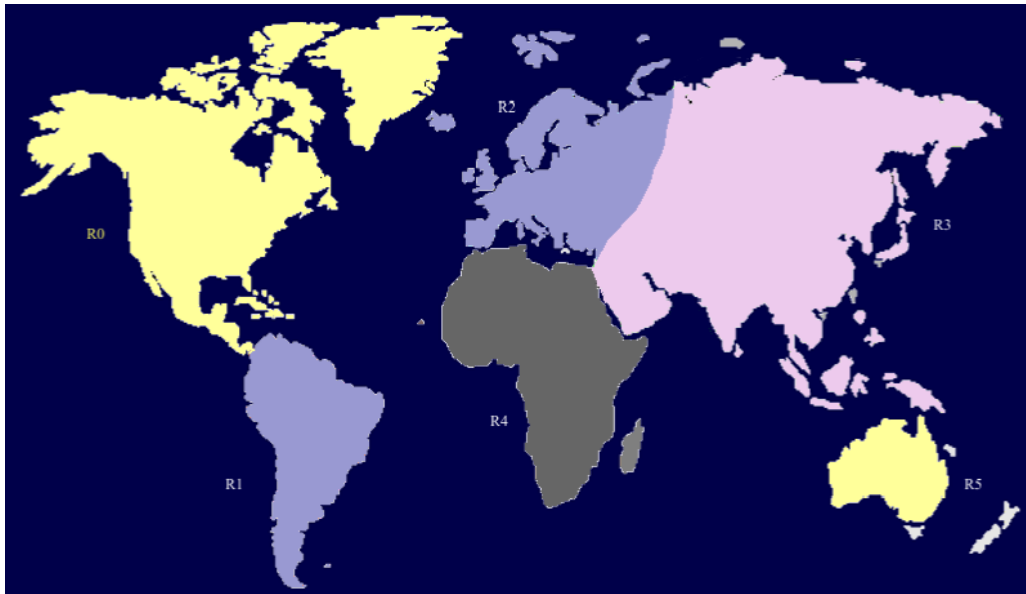
Τα βασικά πλεονεκτήματά του εργαλείου προσομοίωσης CloudAnalyst είναι:

- **Ευκολία χρήσης.** Όπως θα δούμε και παρακάτω, η ευκολία και η απλότητα του γραφικού περιβάλλοντος καθιστά την χρήση του πολύ εύκολη. Η εγκατάσταση, η παραμετροποίηση και η εισαγωγή δεδομένων γίνεται εύκολα και γρήγορα.
- **Δυνατότητα να καθορίσει μια προσομοίωση με υψηλό βαθμό παραμετροποίησης και ευελιξίας.** Σε πολλές προσομοιώσεις χρειάζεται να υπολογιστούν πολλοί διαφορετικοί παράγοντες όπως μεγάλος αριθμός κέντρων δεδομένων και εικονικών μηχανημάτων ή διαφορετικών ταχυτήτων και αιτήσεων από πολλούς χρήστες ταυτόχρονα. Το CloudAnalyst μπορεί να επεξεργαστεί υπολογισμούς σε εύρος διαφορετικών παραμέτρων και δεδομένων. Δεδομένα που μπορεί να εισάγει ο χρήστης για την προσομοίωση είναι ο αριθμός των κέντρων δεδομένων, τα εικονικά μηχανήματα που διαθέτουν, ο γεωγραφικός χώρος που βρίσκονται, η επεξεργαστική ισχύ τους, η μνήμη, ο αποθηκευτικός χώρος, το λειτουργικό σύστημα, οι αλγόριθμοι διαμοιρασμού της κίνησης, οι χρήστες, οι ποσότητα αιτημάτων τους, η ταχύτητα internet κτλπ.
- **Επανάληψη των πειραμάτων με μικρές τροποποιήσεις.** Η προσομοίωση είναι ελεγχόμενη και υπάρχει η δυνατότητα επανάληψης της εκτέλεσης των πειραμάτων. Οι ρυθμίσεις των παραμέτρων μπορούν να αποθηκευτούν (.sim) για την ορθή επανάληψη, όπως επίσης δυνατότητα αποθήκευσης υπάρχει και στα εξαγόμενα αποτελέσματα (.pdf).
- **Γραφική απεικόνιση των αποτελεσμάτων.** Τα αποτελέσματα της προσομοίωσης εμφανίζονται σε παράθυρο σε μορφή πινάκων και γραφημάτων με δυνατότητα αποθήκευσης. Η παρουσίαση των αποτελεσμάτων βοηθά την αποδοτικότερη αναγνώριση των σημαντικότερων παραμέτρων και ενισχύει την σύγκριση τους.
- **Ευκολία στην επέκταση.** Ο προσομοιωτής βασίζεται σε μια αρθρωτή σχεδίαση, ενσωματώνοντας διαφορετικές τεχνολογίες (Java SE, Java Swing, CloudSim, SimJava), η οποίες είναι πολύ εύκολο στην επέκταση με μικρές τροποποιήσεις. Επομένως μπορεί συνεχώς να εξελίσσεται.

3.2 Βασικά συστατικά στοιχεία

Τα βασικά συστατικά στοιχεία του προσομοιωτή Cloud Analyst, τα οποία μπορούν να παραμετροποιηθούν ανάλογα με τις απαιτήσεις του χρήστη, είναι τα ακόλουθα:

- **Περιοχές (Regions).** Ο Cloud Analyst χωρίζει τον παγκόσμιο χάρτη σε 6 τμήματα που αντιπροσωπεύουν την κάθε ήπειρο. Τα Data Center και οι ομάδες των χρηστών συνδέονται με τις περιοχές βοηθώντας τη διατήρηση του επιπέδου της ρεαλιστικότητας της εφαρμογής που προσομοιώνουμε.



Εικόνα 3-1: Οι περιοχές του Cloud Analyst πάνω στον παγκόσμιο χάρτη

- **Χαρακτηριστικά Internet.** Εδώ ρυθμίζονται οι παράμετροι του Internet όπως η ταχύτητα και η καθυστέρηση μεταξύ των περιοχών μοντελοποιώντας την κίνηση της δρομολόγησης ανά τον κόσμο.

Configure Internet Characteristics

Use this screen to configure the Internet characteristics.

Delay Matrix

The transmission delay between regions. Units in milliseconds

Region\Region	0	1	2	3	4	5
0	25	100	150	250	250	100
1	100	25	250	500	350	200
2	150	250	25	150	150	200
3	250	500	150	25	500	500
4	250	350	150	500	25	500
5	100	200	200	500	500	25

Bandwidth Matrix

The available bandwidth between regions for the simulated application. Units in Mbps

Region\Region	0	1	2	3	4	5
0	2,000	1,000	1,000	1,000	1,000	1,000
1	1,000	800	1,000	1,000	1,000	1,000
2	1,000	1,000	2,500	1,000	1,000	1,000
3	1,000	1,000	1,000	1,500	1,000	1,000
4	1,000	1,000	1,000	1,000	500	1,000
5	1,000	1,000	1,000	1,000	1,000	2,000

Done Cancel

Εικόνα 3-2: Οι πίνακες παραμετροποίησης των Delay & Bandwidth ανά περιοχή

- **Βάση Χρηστών.** Σε αυτό το στοιχείο ο χρήστης επιλέγει και εισάγει δεδομένα που αναπαριστούν τους χρήστες και τα αιτήματα τα οποία αποστέλλουν. Μπορούμε να παρατηρήσουμε τα πεδία που μπορεί να βάλει ο χρήστης όπως: Περιοχή χρηστών, αιτήματα χρήστη ανά ώρα, μέγεθος δεδομένου ανά αίτηση του χρήστη, ώρες αιχμής και πλήθος χρηστών σε εκείνες τις ώρες αλλά και στις υπόλοιπες ώρες της ημέρας.

Main Configuration Data Center Configuration Advanced

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	60	100	10	12	1000	100
UB2	2	60	100	3	9	1000	100
UB3	5	60	100	21	23	1000	100

Add New Remove

Εικόνα 3-3: Ρύθμιση Βάσης Χρηστών

- **Internet Cloudlet.** Ένας αριθμός αιτημάτων των χρηστών ομαδοποιούνται μέσα σε ένα Internet Cloudlet βασιζόμενο στο Grouping Factor.
- **Data center controller.** Σε αυτή την περιοχή ελέγχονται οι δραστηριότητες των κέντρων δεδομένων. Ο χρήστης μπορεί να προσθέσει κέντρα δεδομένων, να ορίσει την περιοχή όπου βρίσκονται, το κόστος καθώς και το πλήθος των εικονικών μηχανημάτων που διαθέτουν. Στην εικόνα που ακολουθεί μπορούμε να παρατηρήσουμε τα πεδία που μπορεί να βάλει ο χρήστης όπως: Περιοχή Data Center, αρχιτεκτονική, λειτουργικό σύστημα, κόστος ανά εικονικό μηχάνημα, κόστος μνήμης, κόστος μεταφοράς δεδομένων καθώς και πλήθος φυσικών υπολογιστικών μονάδων. Ο Data Center Controller διαχειρίζεται την δρομολόγηση των αιτημάτων που προέρχονται από τις ομάδες χρηστών μέσω του Internet προς τις εικονικές μηχανές.

Main Configuration | **Data Center Configuration** | Advanced

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		1 x86	Linux	Xen	0.1	0.05	0.1	0.1	4
DC3		4 x86	Linux	Xen	0.1	0.05	0.1	0.1	1

Add New
Remove

Physical Hardware Details of Data Center : DC1

Id	Memory (Mb)	Storage (Mb)	Available BW	Number of Processors	Processor Speed	VM Policy
0	204800	100000000	1000000	4	10000	TIME_SHARED
1	204800	100000000	1000000	4	10000	TIME_SHARED

Add New
Copy
Remove

Data Center	# VMs	Image Size	Memory	BW
DC1	5	10000	512	1000
DC2	5	10000	512	1000
DC3	5	10000	512	1000

Add New
Remove

Εικόνα 3-4: Τα πεδία για την ρύθμιση των παραμέτρων για τα Data Center στον CloudAnalyst

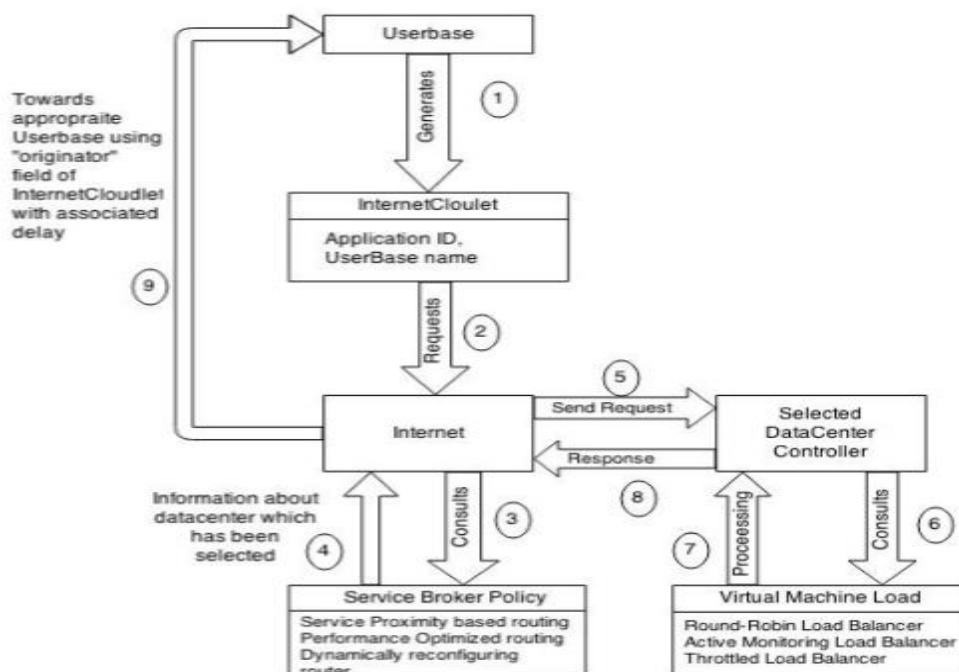
- **VMLoadbalancer.** Ο VMLoadbalancer έχει την ευθύνη της ανάθεσης του φόρτου σε κάποιο από τα εικονικά μηχανήματα ώστε να εκτελεστεί το επόμενο αίτημα των χρηστών. Ο DataCenter Controller αποφασίζει σε ποιο VM πρέπει να ανατεθούν τα αιτήματα χρησιμοποιώντας τον VM Load Balancer. Στον CloudAnalyst παρέχονται τρεις VMLoadBalancers που εκτελούν τρεις διαφορετικές διαδικασίες για την κατανομή των αιτημάτων. Οι πολιτικές αυτές είναι οι ακόλουθες
 1. **Round-Robin Load Balancer:** Στην συγκεκριμένη πολιτική χρησιμοποιείται ο αλγόριθμος Round Robin για την κατανομή των εικονικών μηχανημάτων. Ο αλγόριθμος χρησιμοποιεί μια λίστα των εικονικών μηχανημάτων όπου και διαβιβάζει τα αιτήματα των χρηστών αρχίζοντας από το πρώτο μηχάνημα με την σειρά έως το τελευταίο.
 2. **Equally Spread Current Execution Load:** Σε αυτή την πολιτική ο VMLoadbalancer επικοινωνεί περιοδικά με τα εικονικά μηχανήματα ώστε να γνωρίζει την κατάσταση στην οποία βρίσκονται και προσπαθεί να ισομοιράσει τις εργασίες μεταξύ των εικονικών μηχανών.
 3. **Throttled Load Balancer:** Σε αυτή την πολιτική το αίτημα του χρήστη προωθείται ανάλογα με το φορτίο των εικονικών μηχανημάτων. Αρχικά γίνεται μια αναζήτηση από τον VMLoadBalancer όπου αναζητεί το κατάλληλο μηχάνημα που μπορεί να υποστηρίξει το συγκεκριμένο αίτημα. Εάν δεν βρεθεί τότε το αίτημα μπαίνει σε ουρά αναμονής και όταν κάποια εικονική μηχανή γίνει διαθέσιμη τότε παίρνει όλα τα αιτήματα που βρίσκονται στην ουρά αναμονής.

- **CloudAppServiceBroker.** Ο CloudAppServiceBroker αναλαμβάνει την επιλογή της πολιτικής της κίνησης μεταξύ των κέντρων δεδομένων και των χρηστών. Το εργαλείο CloudAnalyst διαθέτει τρεις πολιτικές και αναλύονται παρακάτω.
 1. **Closest Data Center,** όπου όλη η κίνηση προωθείται στο κοντινότερο Data Center στην περιοχή των χρηστών. Συγκεκριμένα όταν ο Broker λάβει ένα αίτημα, εκείνη την στιγμή συμβουλευόντας τη λίστα με τα χαρακτηριστικά των data centers (delay,bw), επιλέγει το DC με το μικρότερο latency. Εάν υπάρχουν περισσότερα από ένα τότε επιλέγεται ένα τυχαία.
 2. **Optimize Response Time,** όπου στην συγκεκριμένη χρονική στιγμή που αποστέλλεται ένα αίτημα ανάλογα με τους χρόνους απόκρισης όλων των data centers προωθείται στο καταλληλότερο δηλαδή αυτό με το μικρότερο χρόνο απόκρισης. Δεν συνυπολογίζονται τα κόστη.
 3. **Reconfigure Dynamically with load routing,** όπου όταν ένα data center λειτουργεί κάτω από ένα όριο επιδόσεων τότε η κίνησή του μοιράζεται στα υπόλοιπα data centers. Επίσης γίνεται ανακατανομή των εικονικών μηχανημάτων ανάμεσα στα

data centers ανάλογα με την λειτουργία τους ώστε τα data centers τα οποία χρειάζονται περισσότερη επεξεργαστική ισχύ να μπορούν να έχουν στην διάθεσή τους περισσότερα εικονικά μηχανήματα.

3.3 Δρομολόγηση των αιτημάτων των χρηστών

Σε ένα cloud περιβάλλον, από την πλευρά του τελικού χρήστη ο πιο σημαντικός παράγοντας είναι η βελτιστοποίηση του κόστους και ο ρόλος του παρόχου είναι να παρέχει την κατάλληλη βοήθεια στην ανάγκη αυτή. Συνεπώς η δρομολόγηση των αιτημάτων των χρηστών είναι μια πολύ σημαντική πτυχή στα cloud.



Εικόνα 3-5: Το διάγραμμα ροής για την δρομολόγηση των αιτημάτων των χρηστών

1. Η βάση των χρηστών δημιουργεί ένα InternetCloudlet, που περιέχει το Application ID για την εφαρμογή και το όνομα της βάσης του χρήστη για να μπορέσει να δρομολογήσει απάντηση όταν του ζητηθεί (RESPONCE).
2. REQUEST στέλνεται στο Internet με μηδενική καθυστέρηση.
3. Η βαθμίδα Internet συμβουλευτεί το ServiceBroker για την επιλογή των DataCenter. Ο ServiceBroker χρησιμοποιεί μια από τις διαθέσιμες πολιτικές βάση της πληροφορίας του αιτήματος(REQUEST).
4. Ο Service Broker στέλνει τις πληροφορίες για τα επιλεγμένα Data Center στο Internet.

5. Το Internet προσθέτει τα ανάλογα χαρακτηριστικά της γραμμής (καθυστέρηση δικτύου) στο αίτημα REQUEST και το στέλνει στο DataCenterController.
6. Ο DataCenter Controller χρησιμοποιεί κάποια από τις διαθέσιμες πολιτικές για την ανάθεση των αιτημάτων προς τις εικονικές μηχανές.
7. Ο VMLoadBalancer αναθέτει τα αιτήματα των χρηστών στα VM.
8. Το επιλεγμένο DC στέλνει απάντηση (RESPONCE) στο Internet μετά την επεξεργασία του αιτήματος REQUEST.
9. Το Internet χρησιμοποιεί τις πληροφορίες από το InternetCloudlet και προσθέτει την αντίστοιχη καθυστέρηση για το RESPONSE, ενημερώνοντας την UserBase.

3.4 Υπολογισμός της καθυστέρησης μετάδοσης

Ο υπολογισμός της ολικής καθυστέρησης μετάδοσης υπολογίζεται χρησιμοποιώντας την συγκεκριμένη φόρμουλα:

$$T_{total} = T_{latency} + T_{transfer}$$

Όπου $T_{latency}$ είναι η καθυστέρηση του δικτύου και $T_{transfer}$ είναι ο χρόνος που χρειάζεται για να μεταφερθεί η πληροφορία ενός αιτήματος του χρήστη από την αφετηρία μέχρι τον προορισμό.

Η πληροφορία για την καθυστέρηση του δικτύου $T_{latency}$ υπάρχει στον πίνακα με τα χαρακτηριστικά του δικτύου (Internet characteristics), ενώ ο υπολογισμός του χρόνου μετάδοσης

1. Υπολογίζετε το εύρος ζώνης για κάθε χρήστη $BW_{peruser}$

$$BW_{peruser} = \frac{BW_{total}}{Nr}$$

Όπου BW_{total} : το διαθέσιμο εύρος ζώνης και δίνεται από τον πίνακα Internet Characteristics και N_r είναι ο αριθμός των αιτημάτων των χρηστών που μεταδίδονται.

2. Υπολογίζετε η καθυστέρηση μετάδοσης για κάθε αίτημα του χρήστη $T_{transfer}$.

$$T_{transfer} = \frac{D}{BW_{peruser}}$$

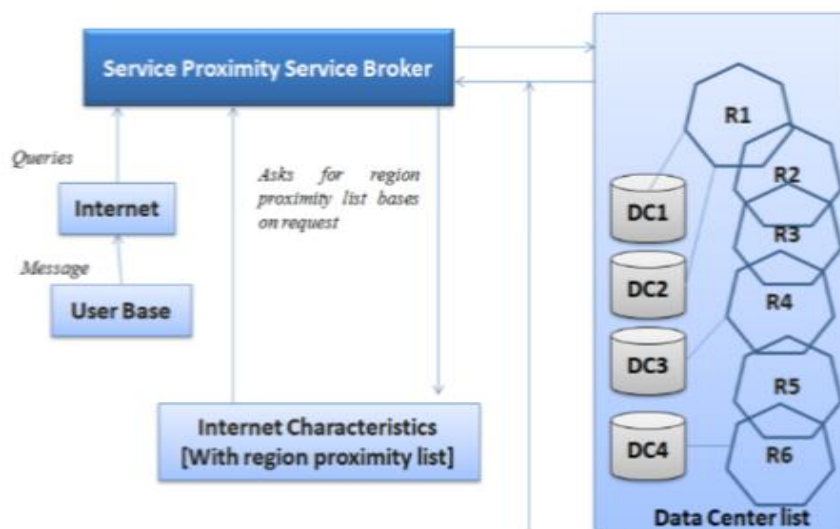
3.5 Αλγόριθμοι επιλογής DataCenter

Όσον αφορά τα κριτήρια με τα οποία γίνεται η επιλογή του datacenter, που θα εξυπηρετήσει τον χρήστη, κατά την προσομοίωση, το Cloud Analyst διαθέτει ήδη τρεις ολοκληρωμένους αλγορίθμους για τον καθορισμό της πολιτικής αυτής (Service Broker Policies). Οι αλγόριθμοι αυτοί παρουσιάστηκαν στην προηγούμενη ενότητα και περιγράφονται αναλυτικά παρακάτω:

A. Service Proximity Based Routing (Closest Data Center)

Ο αλγόριθμος αυτός αποτελεί την πιο απλή υλοποίηση από τις τρεις που υπάρχουν στον Cloud Analyst, όσον αφορά την επιλογή του Data Center που θα εξυπηρετήσει τα αιτήματα των χρηστών. Τα βήματα που ακολουθεί είναι:

1. Η περιοχή που είναι οι χρήστες καθορίζει τα αντίστοιχα πεδία του πίνακα που έχει τα χαρακτηριστικά για κάθε DC και ο ServiceProximityServiceBroker τα δέχεται.
2. Το Internet όταν δέχεται τα αιτήματα των χρηστών, ζητά από τον SPSB τα αντίστοιχα πεδία που αφορούν το αίτημα και χρησιμοποιεί την πληροφορία αυτή για να ζητήσει από το DataCenterController τον προορισμό.
3. Την περιοχή των αιτημάτων την ανακτά από τον SPSB. Τα ερωτήματα για το πιο κοντινό DC τα ανακτά πάλι από τον SPSB μέσω των χαρακτηριστικών του δικτύου. Συνεπώς δημιουργείται μια λίστα με τις περιοχές που μένουν, έκτος της περιοχής που ξεκινάνε τα αιτήματα των χρηστών, σύμφωνα με την μικρότερη δικτυακή καθυστέρηση που υπολογίζεται για κάθε περιοχή από το σημείο αναφοράς.
4. Το DC που τοποθετείται στην κορυφή της λίστας (μικρότερη καθυστέρηση) επιλέγεται από τον SPSB. Αν υπάρχουν παραπάνω DC σε μια περιοχή, τότε η επιλογή των DC γίνεται τυχαία.



Εικόνα 3-6: Ο αλγόριθμος Service Proximity Service Broker

B. Performance Optimized Routing (Optimise Response Time)

Ο αλγόριθμος αυτός διευρύνει τον Service Proximity Service Broker. Ο BestResponseTime ServiceBroker εφαρμόζει τον κανόνα για βελτιστοποιημένη επίδοση δρομολόγησης. Τα βήματα που ακολουθεί είναι:

1. Ο κατάλογος με όλα τα διαθέσιμα DC διατηρείται από τον BestResponseTime ServiceBroker.
2. Το Internet όταν δέχεται τα αιτήματα των χρηστών, ζητά από τον BRTSB τα αντίστοιχα πεδία που αφορούν το αίτημα και χρησιμοποιεί την πληροφορία αυτή για να ζητήσει από το DataCenterController τον προορισμό.
3. Λαμβάνοντας υπόψη την καθυστέρηση, εφαρμόζεται η τεχνική του SPSB και επιλέγεται το πιο κοντινό DC από τον BRTSB.
4. Η λίστα περιέχει κάθε Data Center που έχει περάσει από το BRTSB και μέσω της επανάληψης της διαδικασίας για όλα τα DC υπολογίζεται ο χρόνος απόκρισης για κάθε DC ξεχωριστά. Η διαδικασία που ακολουθείται είναι:
 - a. Ο τελευταίος καταγεγραμμένος χρόνος επεξεργασίας αναζητείται από τις ρυθμίσεις του δικτύου.
 - b. Ο χρόνος επεξεργασίας ενός συγκεκριμένου data center μηδενίζει, όταν ο τελευταίος καταχωρημένος χρόνος επεξεργασίας έχει καταγραφεί πριν ένα προκαθορισμένο όριο. Το data center μένει άπραγο όσο διαρκεί ο χρόνος του κατωφλίου.
 - c. Η τιμή που αποκομίστηκε από τα παραπάνω βήματα προστέθηκε στην καθυστέρηση του δικτύου από τον πίνακα των χαρακτηριστικών του Internet.
5. Όταν ο ελάχιστος εκτιμώμενος χρόνος επεξεργασίας αφορά το πιο κοντινό Data Center τότε επιλέγεται από τον BRTSB. Σε διαφορετική περίπτωση ο BRTSB επιλέγει το πιο κοντινό Data Center με πιθανότητα 50%.

Γ. Reconfigure Dynamically with load routing (Dynamic Service Broker Algorithm)

Ο αλγόριθμος αυτός περιέχει τα πλεονεκτήματα των προαναφερθέντων αλγορίθμων. Τα βήματα που ακολουθούνται είναι:

1. Ο DynamicServiceBroker περιέχει δύο λίστες. Μία λίστα με όλα τα data center και μία λίστα με τον χρόνο απόκρισης του κάθε data center που έχει καταγραφεί.
2. Όταν το Internet λάβει ένα μήνυμα από την User Base, τότε η user base ρωτά το DynamicServiceBroker για να λάβει απάντηση τον προορισμό του DataCenterController.

3. Για την αναγνώριση του προορισμού χρησιμοποιεί τόσο τον αλγόριθμο Best Response Time, όσο και τον Service Proximity ServiceBroker.
4. Ο DynamicServiceBroker ανανεώνει τον καταγεγραμμένο χρόνο απόκρισης ως καλύτερο, όταν ο τωρινός χρόνος είναι καλύτερος από τον προηγούμενο.

3.6 Υλοποίηση αλγορίθμου στο CloudAnalyst

Στο πλαίσιο της παρούσας διπλωματικής έγινε προσπάθεια υλοποίησης ενός επιπλέον αλγορίθμου – service broker policy, του **ServiceProximityRRServiceBroker**, ο οποίος ενσωματώθηκε στον κώδικα του Cloud Analyst.

Ο νέος αλγόριθμος βασίστηκε στον υπάρχοντα αλγόριθμο Service Proximity Service Broker Policy. Η διαφοροποίηση έγινε στον τρόπο επιλογής του DataCenter, στο οποίο θα σταλεί προς επεξεργασία το τρέχον αίτημα (ή αιτήματα) του χρήστη. Στον ήδη υλοποιημένο Service Proximity αλγόριθμο η επιλογή γίνεται τυχαία μεταξύ των DataCenters που βρίσκονται στο συγκεκριμένο region. Στον αλγόριθμο, που υλοποιήσαμε, η επιλογή του DataCenter για αποστολή του αιτήματος γίνεται με την τεχνική Round Robin, δηλαδή κάθε νέο αίτημα αποστέλλεται στο επόμενο DataCenter της λίστας. Όταν η λίστα εξαντληθεί, το επόμενο αίτημα αποστέλλεται ξανά στο πρώτο DataCenter της λίστας και τα επόμενα αιτήματα στα επόμενα DataCenters διαδοχικά. Στόχος της τεχνικής αυτής είναι να επιτευχθεί δίκαια και ομοιόμορφη σχετικά χρήση των διαθέσιμων DataCenters εντός ενός region.

Ο κώδικας του αλγορίθμου παρατίθεται παρακάτω:

ServiceProximityRRServiceBroker.java

```
package cloudsim.ext.servicebroker;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import cloudsim.ext.GeoLocatable;
import cloudsim.ext.InternetCharacteristics;
import cloudsim.ext.datacenter.DatacenterController;

public class ServiceProximityRRServiceBroker implements
CloudAppServiceBroker {

    protected Map<Integer, List<String>> regionalDataCenterIndex = null;

    //μεταβλητή που κρατάει το index στην λίστα με τα DataCenters ενός region
    private int currDC = -1;
```

```

public ServiceProximityServiceBroker(){
    regionalDataCenterIndex = new HashMap<Integer, List<String>>();
    init();
}

protected void init(){
    List<GeoLocatable> allInternetEntities = InternetCharacteristics.getInstance().getAllEntities();
    int region;

    for (GeoLocatable entity : allInternetEntities){

        if (entity instanceof DatacenterController){
            region = entity.getRegion();
            List<String> l = regionalDataCenterIndex.get(region);
            if (l == null){
                l = new ArrayList<String>();
                regionalDataCenterIndex.put(region, l);
            }
            l.add(entity.get_name());
        }
    }

}

public String getDestination(GeoLocatable inquirer) {
    List<Integer> proximityList =
InternetCharacteristics.getInstance().getProximityList(inquirer.getRegion());

    int region;
    String dcName;
    for (int i = 0; i < proximityList.size(); i++){
        region = proximityList.get(i);
        dcName = getNextDataCenter(region);
        if (dcName != null){
            return dcName;
        }

        //If it comes here, that means there are no DC's anywhere
        throw new RuntimeException("Looks like you have not configured
any Data Centers. Please check the configuration");
    }

    protected String getNextDataCenter(int region) {

        List<String> regionalList =
regionalDataCenterIndex.get(region);

        String dcName = null;
        //πηγαίνει στο επόμενο στοιχείο της λίστας (αύξηση κατά ένα)
        currDC++;

        if (regionalList != null){

```



```

        int listSize = regionalList.size(); //3
    if (listSize == 1){
        dcName = regionalList.get(0);
    }
    else {
        //αν η λίστα εξαντληθεί, πηγαίνει στην αρχή
        if (currDC >= regionalList.size()){
            currDC = 0;
        }
        //παίρνει το όνομα του DataCenter από την τιμή στη θέση που δείχνει η currDC
        dcName = regionalList.get(currDC);
    }
}

return dcName;
}
}

```

Ο παραπάνω αλγόριθμος χρησιμοποιήθηκε συγκριτικά με δύο από τους τρεις υφισταμένους αλγόριθμους Service Broker, που διαθέτει το CloudAnalyst, προκειμένου να δοκιμαστούν ορισμένα σενάρια που προσομοιώνουν συνθήκες των cloudlets και να εξαχθούν ανάλογα συμπεράσματα από τα αποτελέσματα. Η προσομοίωση των σεναρίων αυτών περιγράφεται αναλυτικά στο επόμενο κεφάλαιο.

4 Μελέτη περίπτωσης χρήσης και προσομοίωση κινητών εφαρμογών σε περιβάλλον με Cloudlets

4.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται η μελέτη μίας περίπτωσης χρήσης, σύμφωνα με την οποία ένας μεγάλος αριθμός χρηστών συγκεντρωμένος στην ίδια γεωγραφική περιοχή για ορισμένο χρονικό διάστημα δημιουργεί πλήθος αιτήσεων προς εφαρμογές που απαιτούν επεξεργαστική ισχύ από την κινητή του συσκευή. Σε ένα κινητό δίκτυο με συμβατική αρχιτεκτονική, αυτού του είδους τα αιτήματα θα εξυπηρετούνταν από κεντρικά datacenters και από τις κινητές συσκευές. Το πλήθος των χρηστών και η συχνότητα των αιτημάτων θα είχε σοβαρές συνέπειες ως προς τον χρόνο απόκρισης και την ποιότητα των υπηρεσιών, αλλά και τους πόρους των κινητών συσκευών.

Ως περίπτωση χρήσης θεωρούμε έναν γενικά πολυσύχναστο χώρο, όπως ένα αεροδρόμιο, όπου ανά διαστήματα εντός της ημέρας ένας μεγάλος αριθμός χρηστών χρησιμοποιεί την κινητή του συσκευή για εφαρμογές και υπηρεσίες που απαιτούν επεξεργαστική ισχύ μαζί με μεταφορά δεδομένων (πχ εφαρμογή αναγνώρισης προσώπου, εφαρμογή mobile check-in κ.α.).

Στο πλαίσιο της παρούσας εργασίας θεωρήσαμε ότι η συγκεκριμένη γεωγραφική περιοχή εξυπηρετείται από περισσότερα μικρά datacenters τοποθετημένα πιο κοντά στους χρήστες (Cloudlets) για την αντιμετώπιση του φόρτου της επεξεργασίας και της καθυστέρησης της κίνησης.

Το περιβάλλον αυτό προσομοιώθηκε με όσο το δυνατόν πιο ρεαλιστικό τρόπο στο εργαλείο CloudAnalyst και στη συνέχεια έγινε η εκτέλεση των σεναρίων, που περιγράφονται στις παρακάτω ενότητες.

4.2 Σενάρια προσομοίωσης

Ως πολιτική επιλογής του DataCenter, που θα δεχτεί το επόμενο ή τα επόμενα (ομαδοποιημένα) αιτήματα των χρηστών προς επεξεργασία, σε όλα τα σεναρία που ακολουθούν, χρησιμοποιήθηκαν διαδοχικά οι αλγόριθμοι ServiceProximityServiceBroker και BestResponseTimeServiceBroker, καθώς και ο ServiceProximityRRServiceBroker, που υλοποιήθηκε στο πλαίσιο της παρούσας διπλωματικής εργασίας.

Όσον αφορά τον αλγόριθμο DynamicServiceBroker, λόγω του ότι αφενός κατά τις εκτελέσεις των πειραμάτων έδινε πολύ διαφορεποιημένα αποτελέσματα (ακόμα και σε επαναληπτικές εκτελέσεις του ίδιου πειράματος) με μεγάλη απόκλιση από αυτά των άλλων δύο αλγορίθμων

του CloudAnalyst, και αφετέρου πρόκειται για «Work In Progress» (WIP), όπως αναφέρεται σε σχόλιο του πηγαίου κώδικα, επιλέξαμε να μην τον συμπεριλάβουμε στις δοκιμές.

Κάθε ένας από τους τρεις αλγορίθμους, που χρησιμοποιήθηκαν στα πειράματα, εκτελέστηκε και με τις τρεις πολιτικές ανάθεσης εργασιών στις εικονικές μηχανές κάθε DataCenter (Round Robin, Throttled, Equally Spread).

Ένα από τα πλεονεκτήματα των Cloudlets είναι η δυνατότητα σύνδεσης των χρηστών σε αυτά μέσω LAN διεπαφών, σε αντίθεση με τα απομακρυσμένα DataCenters, στα οποία οι χρήστες κατά κανόνα συνδέονται με RAN διεπαφές. Αυτό έχει αντίκτυπο στους ρυθμούς μετάδοσης και στην καθυστέρηση διάδοσης της πληροφορίας. Ως εκ τούτου, τα Cloudlets προσφέρουν πολλαπλάσιους ρυθμούς μετάδοσης (bandwidth) και παράλληλα μειωμένη καθυστέρηση (latency), λόγω και της εγγύτητάς τους ως προς τους χρήστες κινητών συσκευών. Καθώς το CloudAnalyst δεν έχει την δυνατότητα παραμετροποίησης της διεπαφής διασύνδεσης ή του φυσικού μέσου σύνδεσης, προκειμένου να προσομοιώσουμε τα ανωτέρω χαρακτηριστικά χρησιμοποιήσαμε τις παραμέτρους του bandwidth και του delay μεταξύ των περιοχών (regions), που διαθέτει το εργαλείο. Θεωρώντας ότι το απομακρυσμένο DataCenter βρίσκεται στο Region 0, ενώ τα Cloudlets και οι χρήστες βρίσκονται στο Region 2, ρυθμίσαμε τις τιμές bandwidth και delay ώστε να αναπαριστούν αφενός την αναλογία μεταξύ HSPA και 802.11n (30Mbps και 300Mbps κατά μέσο όρο, δηλαδή μία αναλογία 1/10), ενώ το delay ρυθμίστηκε σύμφωνα με πραγματικές τιμές μετά από ping εντός Ελλάδος και μεταξύ Ελλάδος και ενός DataCenter της Γερμανίας.

Bandwidth (Mbps)		
Region	2	2
0	10.000	1.000
2	1.000	10.000

Delay (ms)		
Region	0	2
0	25	120
2	120	25

Ως προς την βάση των χρηστών, όλα τα σενάρια, που περιγράφονται παρακάτω, εκτελέστηκαν δύο φορές με μία Βάση Χρηστών αποτελούμενη από 3000 χρήστες και 6000 χρήστες αντίστοιχα κατά τις ώρες αιχμής. Τα αιτήματα των χρηστών ρυθμίστηκαν σε 60 ανά ώρα και το μέγεθος κάθε αιτήματος σε 1MB. Ως ώρες αιχμής ορίστηκε ένα διάστημα τεσσάρων ωρών εντός της ημέρας. Όλες οι εκτελέσεις έγιναν για χρόνο προσομοίωσης 24 ωρών. Στους παρακάτω πίνακες παρουσιάζονται οι δύο Βάσεις Χρηστών, που χρησιμοποιήθηκαν **διαδοχικά** στις εκτελέσεις κάθε σεναρίου.

User Bases Configuration							
User Base	Region	Request per User per Hr	Data Size per Request	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg Peak Users	Avg Off-Peak Users
UB1	2	60	1 MB	18:00	22:00	3000	500
UB1	2	60	1 MB	18:00	22:00	6000	500

Ως προς την ομαδοποίηση των αιτημάτων από την πλευρά των χρηστών και από την πλευρά των DataCenters (Grouping Factors), θεωρήσαμε ότι μέχρι 100 αιτήματα χρηστών καταφτάνουν ταυτόχρονα σε ένα DataCenter προς επεξεργασία, ενώ κάθε DataCenter έχει δυνατότητα ταυτόχρονης επεξεργασίας 10 αιτημάτων.

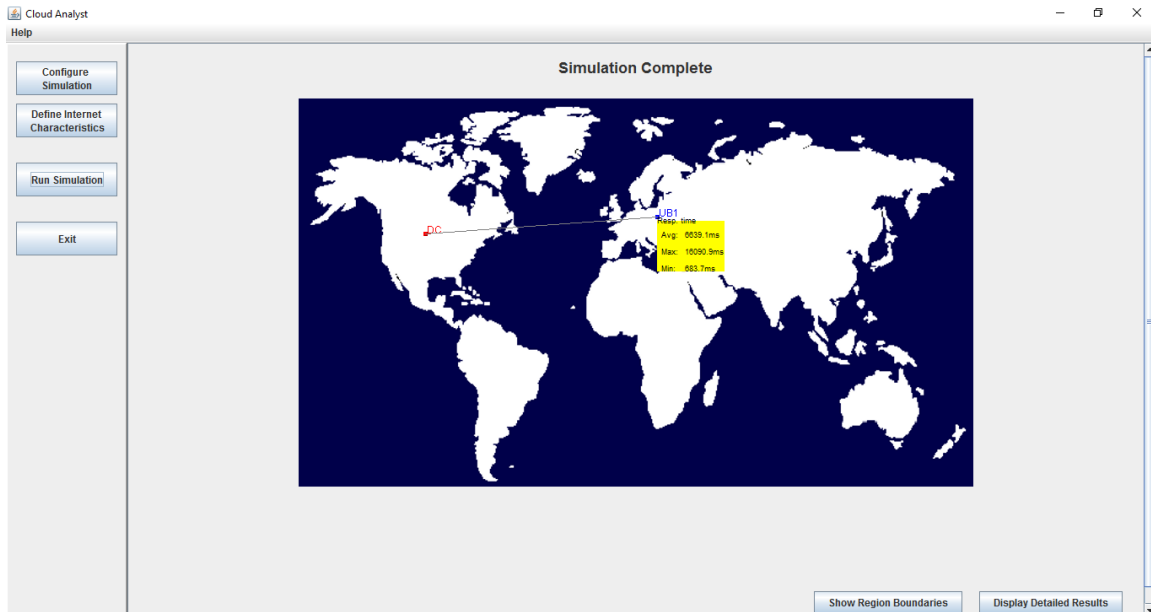
Other Configuration		
User Grouping Factor	Request Grouping Factor	Executable Instruction Length per Request
100	10	100 bytes

Τα χαρακτηριστικά των φυσικών hosts σε κάθε DataCenters παρουσιάζονται στον παρακάτω πίνακα. Το απομακρυσμένο DataCenter διαθέτει τέσσερις φυσικές μονάδες, ενώ κάθε μικρότερο Cloudlet διαθέτει από μία.

Physical Hardware Details of Physical Units (any DataCenter)					
Memory	Storage	Available BW	Number of Processors	Processor Speed	VM Policy
2GB	10TB	1Gbps	4	10000 MIPS	Time Shared

4.3 Σενάριο 1

Στο Σενάριο 1 θεωρήσαμε ότι τα αιτήματα των χρηστών εξυπηρετούνται μόνο από ένα απομακρυσμένο Data Center χωρίς την ύπαρξη τοπικών Cloudlets. Το απομακρυσμένο DataCenter και η βάση των χρηστών βρίσκονται σε διαφορετικά regions.



Εικόνα 4-1: Γραφική απεικόνιση της προσομοίωσης του σεναρίου 1 στο Cloud Analyst

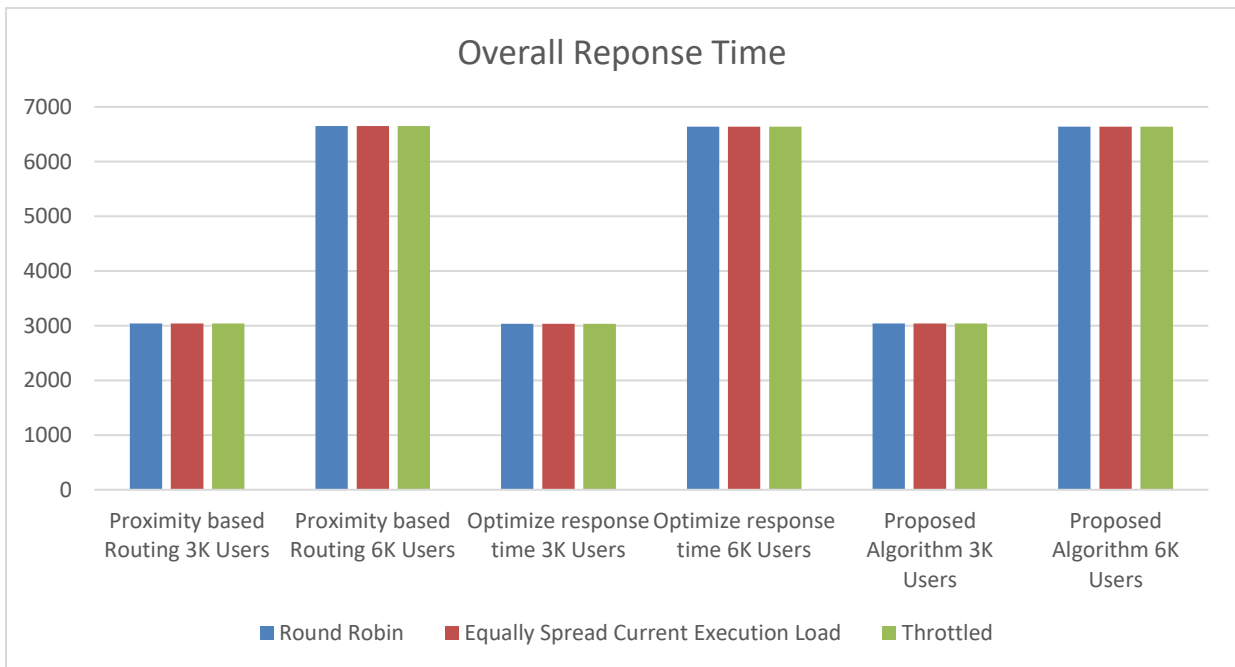
4.3.1 Παράμετροι προσομοίωσης

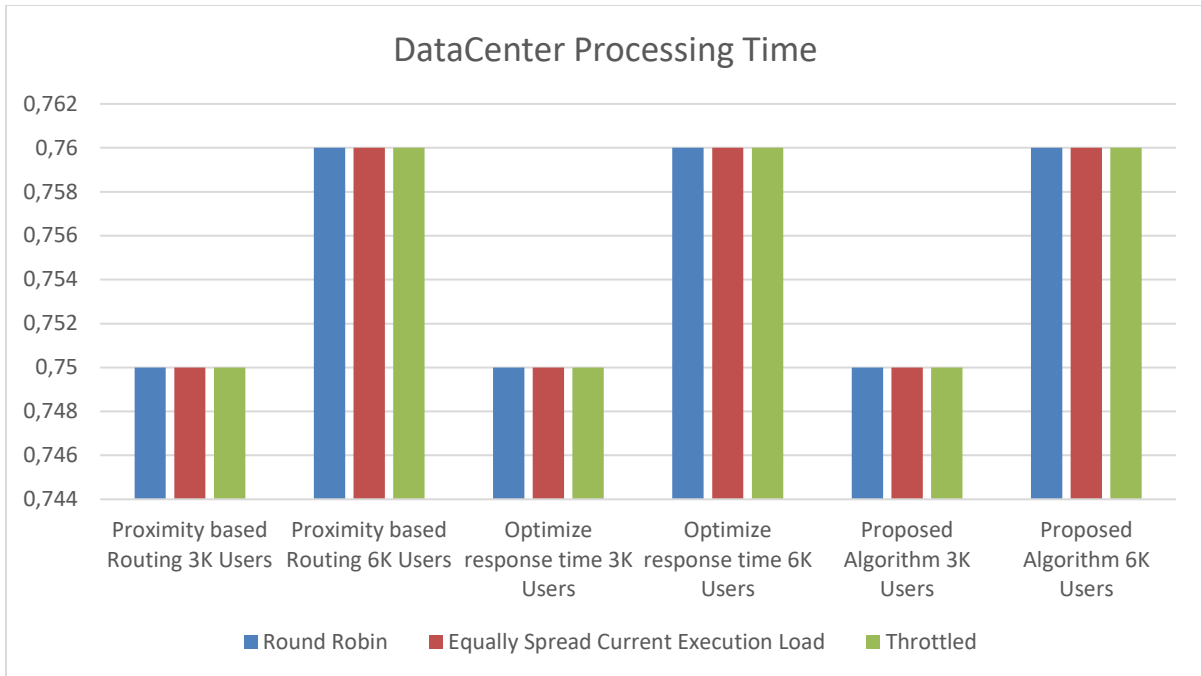
Data Center Configuration									
Name	Region	Arch	OS	VMM	Cost per VM \$/hour	Memory cost \$/s	Storage cost \$/s	Data transfer cost \$/Gb	Physical Hardware Units
DC	0	x86	Linux	Xen	0.1	0.05	0.1	0.1	2

Application Deployment Configuration				
Data Center	VMs	Image Size	Memory	Bandwidth
DC	40	10000	512	1000

4.3.2 Αποτελέσματα προσομοίωσης

VM Allocation Policy	Scenario 1	Service Broker Policy					
		Proximity based Routing		Optimize response time		Proposed Algorithm	
		3000 Users	6000 Users	3000 Users	6000 Users	3000 Users	6000 Users
Round Robin	Overall Response Time	3037,74	6649,73	3036,34	6639,06	3037,74	6639,06
	Data Center Processing Time	0,75	0,76	0,75	0,76	0,75	0,76
	VM Cost	96,03	96,03	96,03	96,03	\$96,03	96,03
Equally Spread Current Execution Load	Overall Response Time	3037,74	6649,73	3036,34	6639,06	3037,74	6639,06
	Data Center Processing Time	0,75	0,76	0,75	0,76	0,75	0,76
	VM Cost	96,03	96,03	96,03	96,03	96,03	96,03
Throttled	Overall Response Time	3037,74	6649,73	3036,34	6639,06	3037,74	6639,06
	Data Center Processing Time	0,75	0,76	0,75	0,76	0,75	0,76
	VM Cost	96,03	96,03	96,03	96,03	96,03	96,03

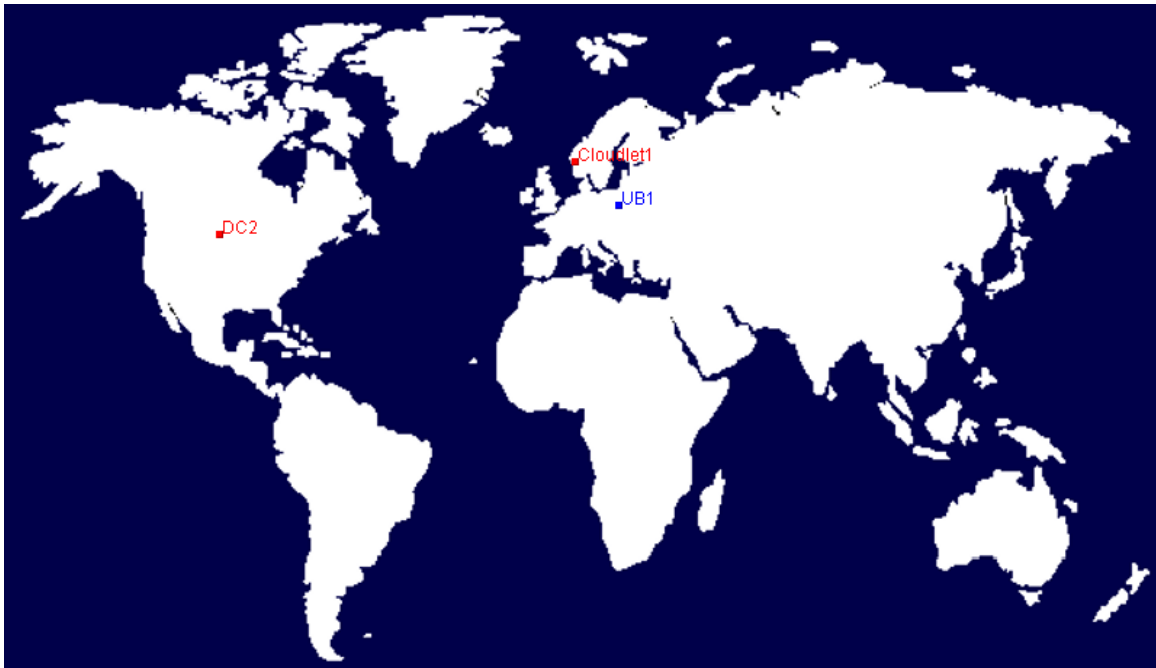




Τα αποτελέσματα του Σεναρίου 1 ήταν αναμενόμενα. Καθώς υπάρχει μόνο ένα απομακρυσμένο DataCenter, όλα τα αιτήματα των χρηστών προωθούνται σε αυτό για επεξεργασία. Ως εκ τούτου, οι διαφορετικοί αλγόριθμοι επηρεάζουν ελάχιστα έως καθόλου τον χρόνο απόκρισης και επεξεργασίας του DataCenter.

4.4 Σενάριο 2

Στο Σενάριο 2 θεωρήσαμε, εκτός από το απομακρυσμένο DataCenter, ένα επιπλέον μικρό Cloudlet στην ίδια περιοχή με την βάση των χρηστών, με στόχο να διαπιστώσουμε αν θα υπάρξει βελτίωση στον χρόνο απόκρισης, εφόσον μεγάλο μέρος των αιτημάτων θα εξυπηρετηθεί από το τοπικό Cloudlet. Το σενάριο εκτελέστηκε διαδοχικά με τις δύο Βάσεις Χρηστών, που περιγράφηκαν παραπάνω.



Εικόνα 4-2: Γραφική απεικόνιση της προσομοίωσης του σεναρίου 2 στο Cloud Analyst

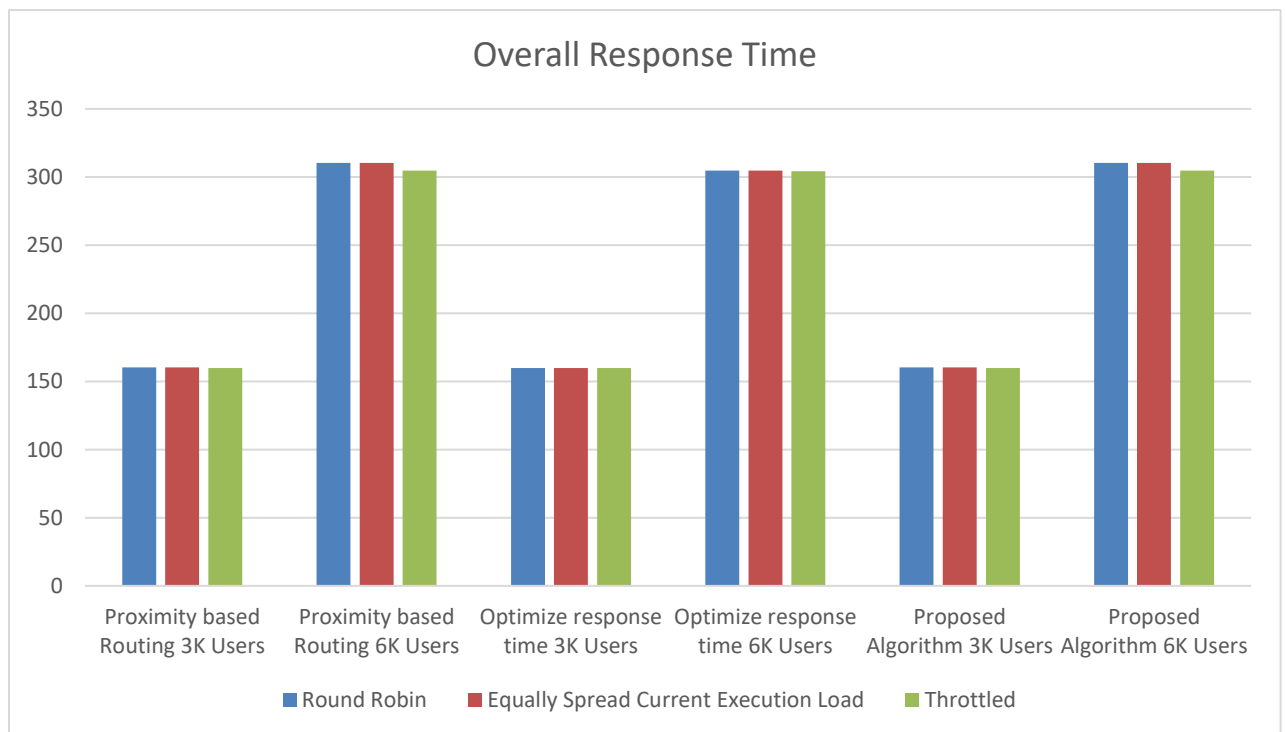
4.4.1 Παράμετροι προσομοίωσης

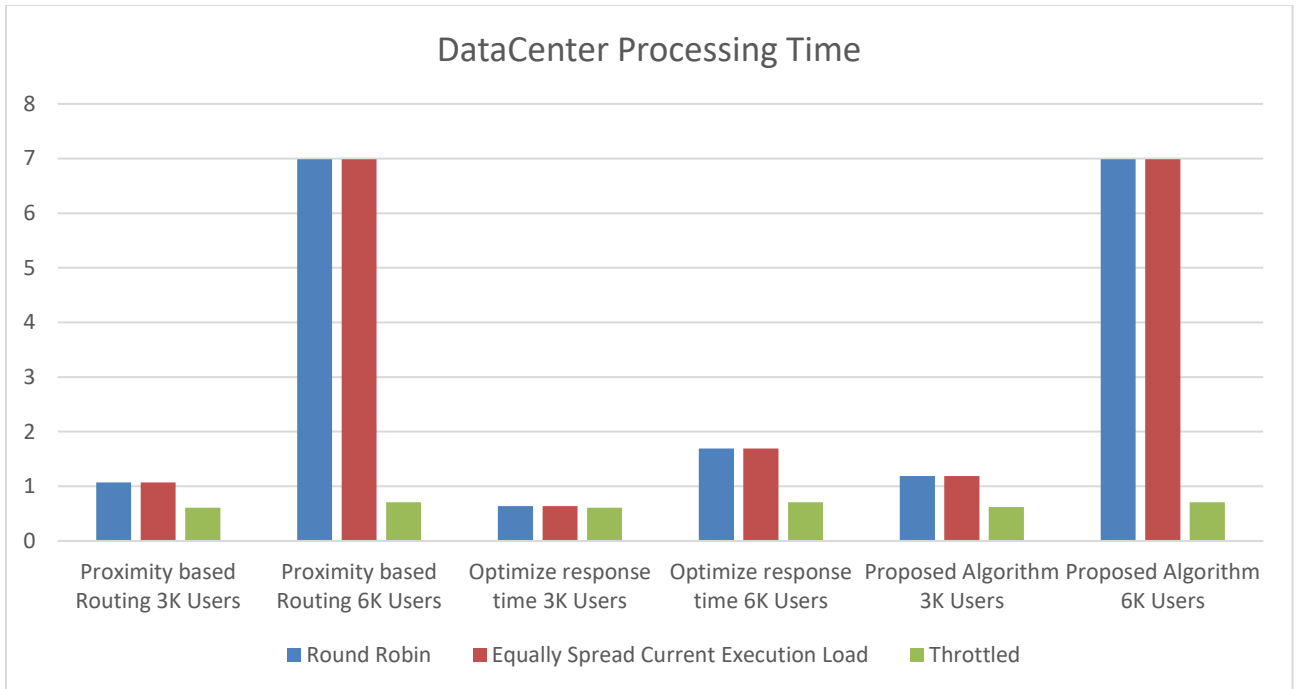
Data Center Configuration									
Name	Region	Arch	OS	VMM	Cost per VM \$/hour	Memory cost \$/s	Storage cost \$/s	Data transfer cost \$/Gb	Physical Hardware Units
DC	0	x86	Linux	Xen	0.1	0.05	0.1	0.1	2
Cloudlet1	2	x86	Linux	Xen	0.1	0.05	0.1	0.1	1

Application Deployment Configuration				
Data Center	VMs	Image Size	Memory	Bandwidth
DC	40	10000	512	1000
Cloudlet1	1	10000	512	1000

4.4.2 Αποτελέσματα προσομοίωσης

VM Allocation Policy	Scenario 2	Service Broker Policy					
		Proximity based Routing		Optimize response time		Proposed Algorithm	
		3000 Users	6000 Users	3000 Users	6000 Users	3000 Users	6000 Users
Round Robin	Overall Response Time	160,25	310,37	159,76	304,65	160,25	310,37
	Data Center Processing Time	1,07	6,99	0,64	1,69	1,19	6,99
	VM Cost	98,43	98,43	98,43	98,43	98,43	98,43
Equally Spread Current Execution Load	Overall Response Time	160,25	310,37	159,76	304,65	160,25	310,37
	Data Center Processing Time	1,07	6,99	0,64	1,69	1,19	6,99
	VM Cost	98,43	98,43	2,4	2,4	98,43	98,43
Throttled	Overall Response Time	159,81	304,8	159,75	304,39	159,81	304,8
	Data Center Processing Time	0,61	0,71	0,61	0,71	0,62	0,71
	VM Cost	98,43	98,43	98,43	98,43	98,43	98,43





Στο Σενάριο 2 παρατηρούμε ότι η ύπαρξη ενός τοπικού Cloudlet στην περιοχή των χρηστών υποδιπλασιάζει σχεδόν τους χρόνους απόκρισης και επεξεργασίας των αιτημάτων, καθώς το μεγαλύτερο μέρος των αιτημάτων προωθείται στο Cloudlet για επεξεργασία, το οποίο προσφέρει πολύ μεγαλύτερους ρυθμούς μετάδοσης δεδομένων σε σχέση με το απομακρυσμένο DataCenter, οπότε δημιουργεί αποφόρτιση και ως προς την επεξεργασία των δεδομένων και ως προς την μεταφορά τους.

Ως προς τους αλγορίθμους για την επιλογή του DataCenter, παρατηρούμε ότι ο Optimized Response Time δίνει τα καλύτερα αποτελέσματα ως προς τον συνολικό χρόνο απόκρισης, ενώ όσον αφορά τους αλγορίθμους ανάθεσης εργασιών στα VMs εντός ενός DataCenter καλύτερα φαίνεται να λειτουργεί ο Throttled. Αν και οι διαφορές σε σχέση με τους άλλους είναι της τάξεως του μισού ms, σε ένα σενάριο μεγαλύτερης κλίμακας η διαφορά θα ήταν ουσιώδης.

Ο αλγόριθμος Service Proximity Round Robin, που υλοποιήσαμε, δίνει ακριβώς ίδια αποτελέσματα με τον Service Proximity, κάτι που είναι αναμενόμενο, καθώς στο συγκεκριμένο σενάριο υπάρχει μόνο ένα τοπικό Cloudlet, οπότε και τα σχετικά αιτήματα των χρηστών –όσα δεν πηγαίνουν στο απομακρυσμένο DataCenter, θα προωθούνται σε αυτό το μοναδικό Cloudlet κάθε φορά.

4.5 Σενάριο 3

Σε συνέχεια του σεναρίου 2, στο Σενάριο 3 θεωρήσαμε το απομακρυσμένο DataCenter και επιπλέον 5 μικρά Cloudlets στην ίδια περιοχή με τις βάσεις των χρηστών, με στόχο να διερευνήσουμε κατά πόσον το πλήθος των τοπικών Cloudlets επηρεάζει ακόμα περισσότερο τους χρόνους απόκρισης και επεξεργασίας των DataCenters. Το σενάριο εκτελέστηκε διαδοχικά με τις δύο βάσεις χρηστών, που περιγράφηκαν παραπάνω.



Εικόνα 4-3: Γραφική απεικόνιση της προσομοίωσης του σεναρίου 3 στο Cloud Analyst

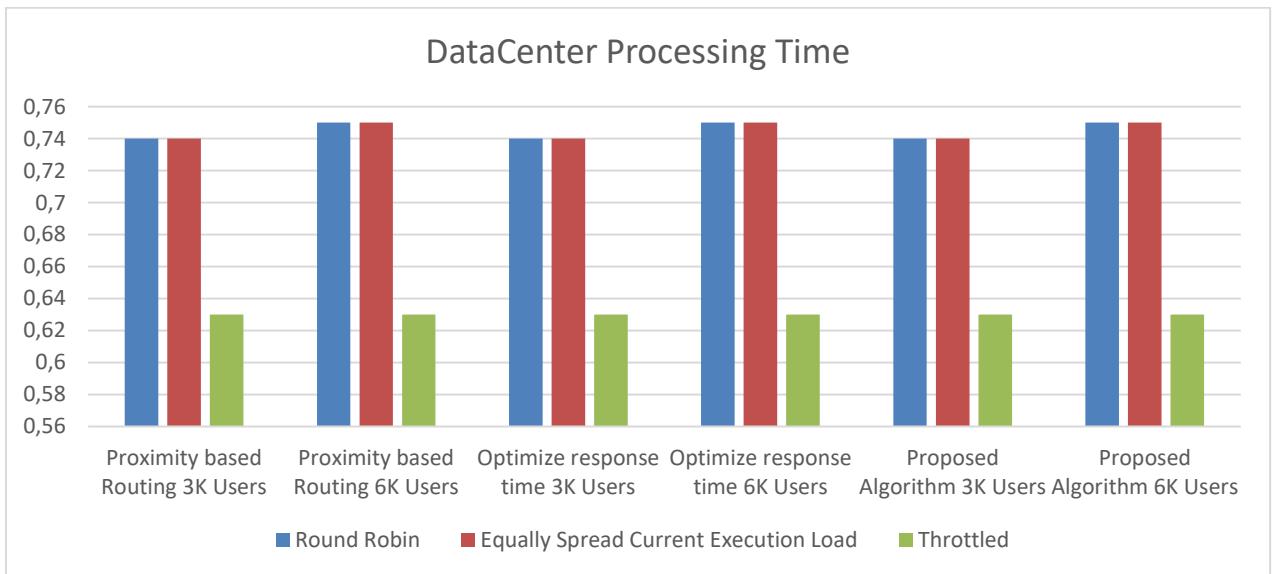
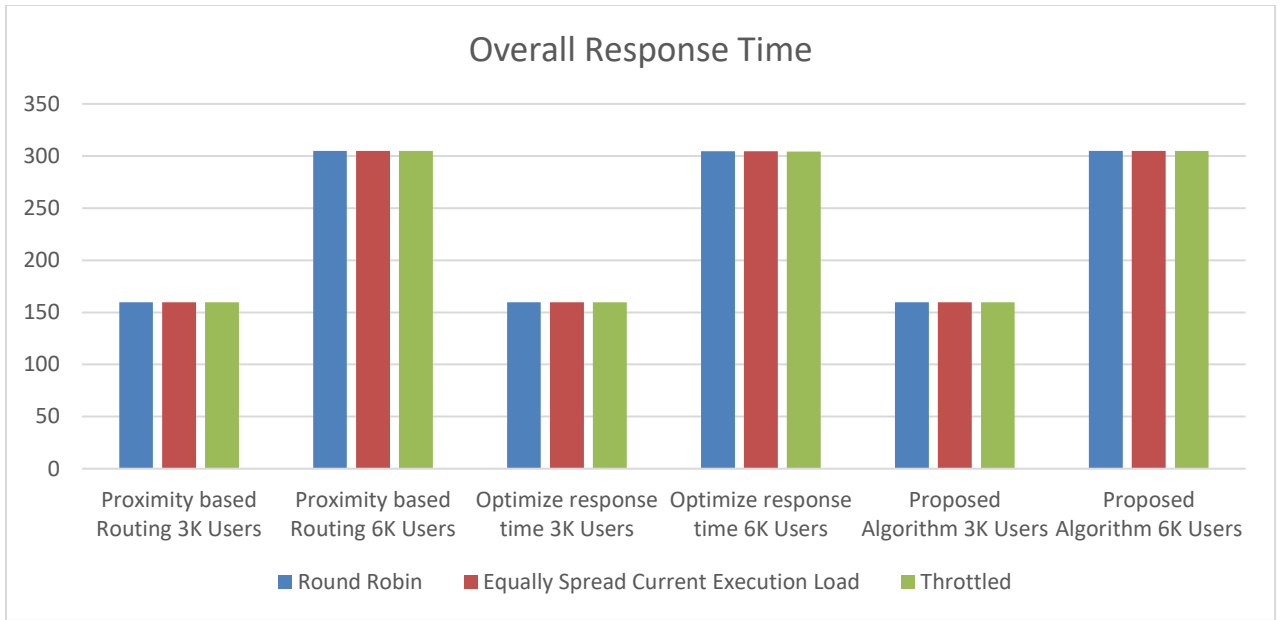
4.5.1 Παράμετροι προσομοίωσης

Data Center Configuration									
Name	Region	Arch	OS	VMM	Cost per VM \$/hour	Memory cost \$/s	Storage cost \$/s	Data transfer cost \$/Gb	Physical Hardware Units
DC	0	x86	Linux	Xen	0.1	0.05	0.1	0.1	2
Cloudlet 1	2	x86	Linux	Xen	0.1	0.05	0.1	0.1	1
Cloudlet 2	2	x86	Linux	Xen	0.1	0.05	0.1	0.1	1
Cloudlet 3	2	x86	Linux	Xen	0.1	0.05	0.1	0.1	1
Cloudlet 4	2	x86	Linux	Xen	0.1	0.05	0.1	0.1	1
Cloudlet 5	2	x86	Linux	Xen	0.1	0.05	0.1	0.1	1

Application Deployment Configuration				
Data Center	VMs	Image Size	Memory	Bandwidth
DC	40	10000	512	1000
Cloudlet1	5	10000	512	1000
Cloudlet2	5	10000	512	1000
Cloudlet3	5	10000	512	1000
Cloudlet4	5	10000	512	1000
Cloudlet5	5	10000	512	1000

4.5.2 Αποτελέσματα προσομοίωσης

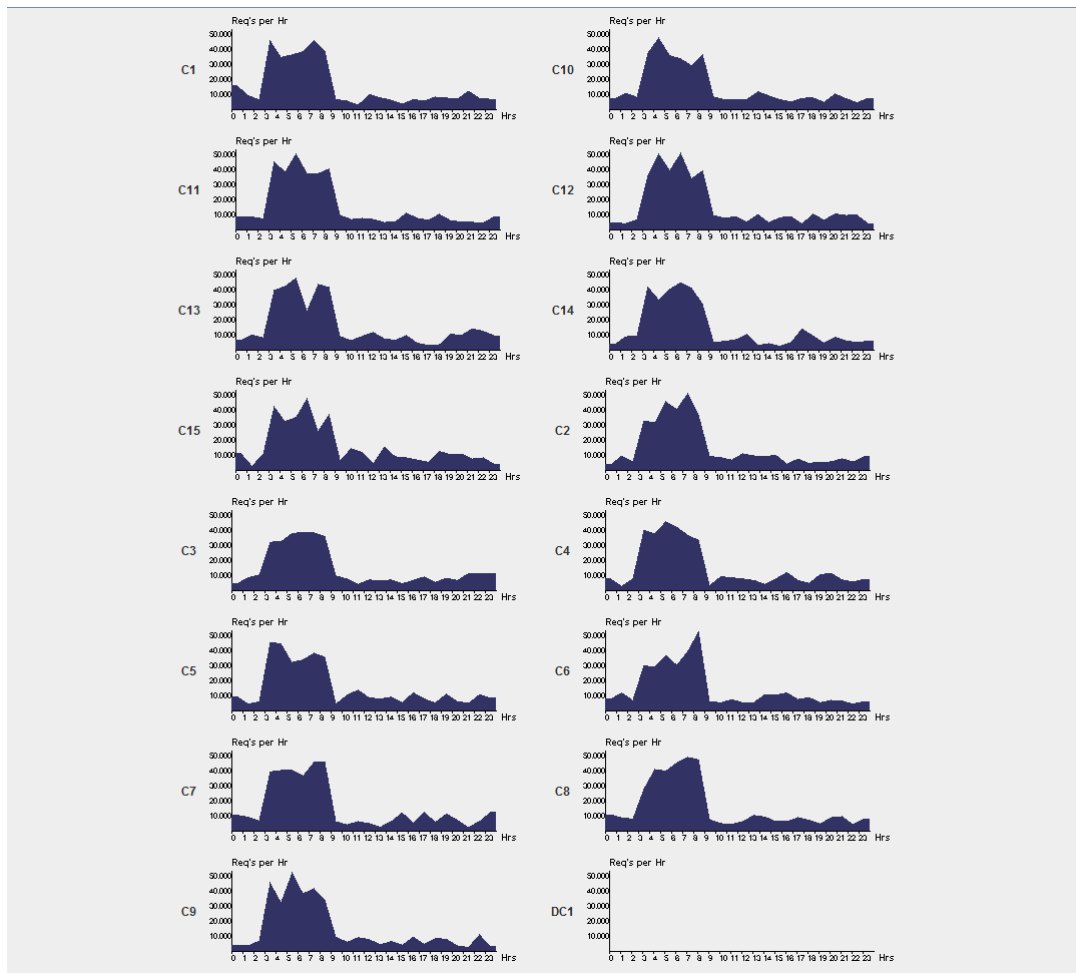
VM Allocation Policy	Scenario 3	Service Broker Policy					
		Proximity based Routing		Optimize response time		Proposed Algorithm	
		3000 Users	6000 Users	3000 Users	6000 Users	3000 Users	6000 Users
Round Robin	Overall Response Time	159,81	304,85	159,73	304,45	159,8	304,8
	Data Center Processing Time	0,74	0,75	0,74	0,75	0,74	0,75
	VM Cost	153,64	153,64	153,64	153,64	153,64	153,64
Equally Spread Current Execution Load	Overall Response Time	159,8	304,8	159,73	304,45	159,8	304,8
	Data Center Processing Time	0,74	0,75	0,74	0,75	0,74	0,75
	VM Cost	153,64	153,64	153,64	153,64	153,64	153,64
Throttled	Overall Response Time	159,76	304,77	159,69	304,42	159,76	304,77
	Data Center Processing Time	0,63	0,63	0,63	0,63	0,63	0,63
	VM Cost	153,64	153,64	153,64	153,64	153,64	153,64



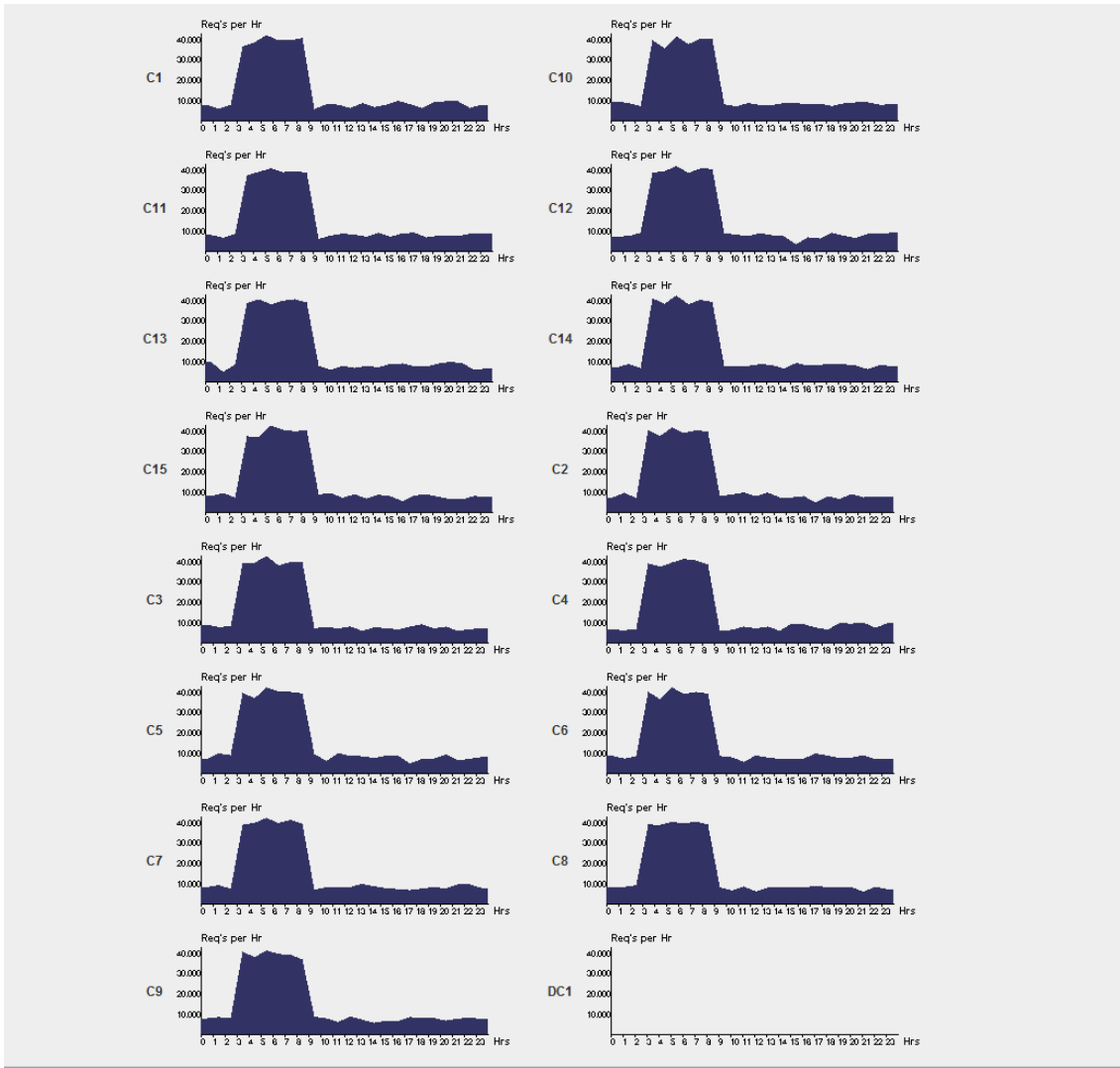
Σε σχέση με το Σενάριο 2, όπου είχαμε ένα τοπικό Cloudlet, στο Σενάριο 3 με 5 Cloudlets παρατηρούμε ελαφρά μείωση σε κάποιες περιπτώσεις ως προς τον χρόνο επεξεργασίας των αιτημάτων στα DataCenters, το οποίο μπορεί να ερμηνευτεί από την μείωση του φόρτου των αιτημάτων, που δέχεται κάθε Cloudlet, αφού πλέον αυτός κατανέμεται σε πέντε αντί για ένα Cloudlet.

Για να μελετήσουμε καλύτερα τον Round Robin Service Proximity Service Broker, που υλοποιήσαμε, προσομοιώσαμε μία παραλλαγή του σεναρίου αυτού αυξάνοντας το πλήθος των τοπικών Cloudlets σε 15 και τους χρήστες σε 10000 και 2000 για τις ώρες αιχμής και μη αιχμής αντίστοιχα, καθώς και τις ώρες αιχμής σε 6. Επιπλέον, ως προς την ομαδοποίηση αιτημάτων, αυξήσαμε το User Grouping Factor σε 1000 και το Request Grouping Factor σε 100.

Από τα παρακάτω αποτελέσματα, που αναπαριστούν τον φόρτο κάθε DataCenter κατά την διάρκεια εκτέλεσης του σεναρίου με την πολιτική επιλογής DataCenter “ServiceProximityServiceBroker” και “ServiceProximityRoundRobinServiceBroker” αντίστοιχα, στην δεύτερη περίπτωση παρατηρούμε ότι ο φόρτος των αιτημάτων κατανέμεται πιο ομοιόμορφα μεταξύ των Cloudlets. Αυτό εξηγείται από τον τρόπο λειτουργίας των δύο αλγορίθμων. Ο Round Robin, που υλοποιήσαμε, επιλέγει κάθε φορά το επόμενο DataCenter του ίδιου region, ενώ ο Service Proximity επιλέγει τυχαία μεταξύ των DataCenters του ίδιου region.



Εικόνα 4-4: Φόρτος αιτημάτων στα Cloudlets με Service Proximity Service Broker



Εικόνα 4-5: Φόρτος αιτημάτων στα Cloudlets με Service Proximity Round Robin Service Broker

5 Συμπεράσματα

Μετά από την θεωρητική μελέτη του Cloudlet-based Mobile Cloud Computing, αλλά και την προσπάθεια προσομοίωσης σχετικών σεναρίων στο CloudAnalyst, καταλήξαμε στα εξής γενικά συμπεράσματα:

1. Τα Cloudlets είναι μια αισιόδοξη προοπτική στο Mobile Cloud Computing. Όσο οι απαιτήσεις των κινητών χρηστών αυξάνονται διαρκώς και οι κινητές συσκευές έχουν φυσικούς περιορισμούς, παρόλο την εξέλιξη του hardware τους, υπάρχει η ανάγκη να φέρουμε μικρά cloud κοντά στο χρήστη με πολλαπλάσιους ρυθμούς μετάδοσης για να βοηθήσει στην μείωση του latency. Αυτό είναι ένα από τα μεγαλύτερα ζητήματα στα δίκτυα κινητών επικοινωνιών.
2. Μετά από τα ενδεικτικά σενάρια που εκτελέσαμε στον προσομοιωτή, είδαμε ότι όντως η ύπαρξη μικρών τοπικών cloudlets σε συνδυασμό με τα απομακρυσμένα data centers μειώνει τον χρόνο επεξεργασίας των αιτημάτων των χρηστών και τον συνολικό χρόνο απόκρισης του δικτύου, βελτιώνοντας την εμπειρία του χρήστη.
3. Σε small-scale σενάρια, οι διαφορές μεταξύ των αλγορίθμων επιλογής του data center στο οποίο θα σταλεί ένα αίτημα (τόσο οι αλγόριθμοι που εμπεριέχονται στο εργαλείο προσομοίωσης CloudAnalyst (Service Proximity Service Broker & Optimize Response Time) όσο και ο προτινόμενος Round Robin Service Broker) και οι αλγόριθμοι ανάθεσης εργασιών στα VMs μέσα σε κάθε data center διαφοροποιούνται ελαφρώς μεταξύ τους και οι διαφορές δεν είναι πολύ αισθητές. Τα καλύτερα αποτελέσματα τα έχει ο συνδυασμός του Optimized Response Time Service Broker με τον Throttled VM Allocation Policy.
4. Σε large scale σενάρια έχουν πιο αξιοσημείωτη διαφοροποίηση με καλύτερο τον optimized response. Θα είχε ενδιαφέρον η υλοποίηση ενός αλγορίθμου δυναμικής επιλογής datacenter συνυπολογίζοντας το φορτίο του, το bandwidth, το latency καθώς και το κόστος.
5. Το κόστος της υλοποίησης αποτελεί σημαντικό παράγοντα και θα πρέπει να γίνει ανάλυση κόστους/αξίας μια τέτοιας υλοποίησης ως προς του παρόχους και ως προς τους χρήστες. Για να είναι βιώσιμη η τεχνολογία, θα πρέπει να βρεθεί μια χρυσή τομή μεταξύ του σημείου που θα έχει ουσιαστική αξία το computation offloading & το data offloading με το πόσο αυτό θα κοστίζει σε hardware, software, network και διαχείριση. Αυτό μπορεί να γίνει με γνώμονα τις εφαρμογές, οπότε θα πρέπει να μελετηθεί και το θέμα της κατανομής των πόρων μεταξύ των cloudlets.

Αναφορές

- [1]: <http://searchcio.techtarget.com/definition/green-IT-green-information-technology>
- [2]: Yang X, Pan T, Shen J. On 3G mobile e-commerce platform based on cloud computing, 2010; 198–201.
- [3]: <https://www.gaikai.com/>
- [4]: <https://unity3d.com/unity/multiplatform>
- [5]: <http://www.gcluster.com/eng/>
- [6]: research.microsoft.com/en-us/.../hcir13-slowsearch.pdf
- [7]: http://badunetworks.com/wp-content/uploads/2014/09/WLH_0818_14.pdf
- [8]: Bonomi, Flavio, et al. "Fog computing and its role in the internet of things." *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012.
- [9]: F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in ACM SIGCOMM Workshop on Mobile cloud Computing, Helsinki, Finland, 2012, pp. 13--16.
- [10]: <http://www.openfogconsortium.org/resources/#definition-of-fog-computing>
- [11]: L.M. Vaquero, L.Rodero-Merino, "Finding your Way in the Fog: Towards a Comprehensive Definition of Fog Computing", ACM SIGCOMM Computer Comm. Review, Vol. 44, No 5, October 2014
- [12]: K.P.Saharan A.Kumar " Fog in Comparison to Cloud: A Survey", Int'l. Journal of Computer Applications (0975 – 8887) Volume 122 – No.3, July 2015
- [13]: L.M. Vaquero, L.Rodero-Merino, "Finding your Way in the Fog: Towards a Comprehensive Definition of Fog Computing", ACM SIGCOMM Computer Comm. Review, Vol. 44, No 5, October 2014
- [14]: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf
- [15]: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf
- [16]: <http://elijah.cs.cmu.edu/>
- [17]: M.Satyanarayanan, et.al., "Cloudlets: at the Leading Edge of Mobile-Cloud Convergence", 2014 6th International Conference on Mobile Computing, Applications and Services (MobiCASE)
- [18]: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7026272

[19]: <http://www.networkworld.com/article/2979570/cloud-computing/microsoft-researcher-why-micro-datacenters-really-matter-to-mobiles-future.html>

[20]: <http://openegecomputing.org/about-oec.html>

[21]: Soyata, T., Ba, H., Heinzelman, W., Kwon, M., & Shi, J. (2013). Accelerating mobile cloud computing: A survey. *Communication Infrastructures for Cloud Computing*, 175-197.

[22]: <http://www.gartner.com/newsroom/id/3412017>

[23]: <http://www.cloudiquity.com/2009/03/cloudsim-offers-cloud-computing-simulation/>