Εαρινό εξάμηνο 2019-2020

Υπολογιστική νέφους & υπηρεσιών

**Προσομοίωση CloudSim**

**Άσκηση 1η**

|  |  |
| --- | --- |
| D:\Users\Harrys\Desktop\cloud-computing.jpg | Ασημακόπουλος Χαράλαμπος  ΑΜ: 141098  Email: [xarhsasi@gmail.com](mailto:xarhsasi@gmail.com) / [cs141098@uniwa.gr](mailto:cs141098@uniwa.gr) |

Περιεχόμενα

[Γενικά 3](#_Toc38818293)

[Εκτέλεση Εργασίας – Σενάρια cloudsim 3](#_Toc38818294)

[Αποτελέσματα / Συμπεράσματα 10](#_Toc38818295)

# Γενικά

Στην εργασία 1, παραμετροποιήσαμε το λογισμικό cloudsim σε περιβάλλον ανάπτυξης Intellij και αναλύσαμε λεπτομερώς τις δυνατότητες τις οποίες μας παρέχει. Κατανοήσαμε τη λειτουργικότητα και τη διασύνδεση των επι μέρους συστημάτων σε ένα cloud based σύστημα και πώς αυτά μεταξύ τους αλληλεπιδρούν. Τέλος, εμβαθύναμε σε τεχνικές όπως space sharing / time sharing και συγκρίναμε τα αποτελέσματα τους μέσα απο το περιβάλλον προσομοίωσης.

# Εκτέλεση Εργασίας – Σενάρια cloudsim

Εκτελώντας το έκτο παράδειγμα παίρνουμε τα παρακάτω αποτελέσματα.

1. Πόσα Datacenters δημιουργούνται;

**Απάντηση:**

Συνολικά δημιουργούνται 2 Datacenters.

1. Πόσοι Hosts δημιουργούνται σε κάθε Datacenter;

**Απάντηση:**

Σε κάθε Datacenter δημιουργούνται απο δύο Hosts.

1. Πόσους επεξεργαστές (PEs) έχει ο κάθε Host;

**Απάντηση:**

Ο πρώτος Host έχει έναν τετραπύρηνο επεξεργαστή ενώ ο δεύτερος έχει έναν διπύρηνο επεξεργαστή.

1. Τι άλλα χαρακτηριστικά έχει ο κάθε HOST(mips, μνήμη, κτλπ);

**Απάντηση:**

Κάθε Host έχει το δικό του μοναδικό ID, τη διαθέσιμη RAM απο τον οποία αποτελείται,

το δικό του προκαθορισμένο εύρος ζώνης(Bandwidth), το χώρο αποθήκευσης (Storage) και τους επεξεργαστές που έχει διαθέσιμους.

1. Τι άλλα χαρακτηριστικά έχει το κάθε Datacenter;

**Απάντηση:**

Το Datacenter περιγράφεται απο τη δική του αρχιτεκτονική (arch), το λειτουργικό σύστημα που έχει εγκατεστημένο (os), το Virtual Machine Monitor (VMM), τους Host που περιέχει, το time zone του συστήματος, το κόστος πεξεργασίας στο συγκεκριμένο datacenter, το κώστος χρησιμοποίησης μνήμης στο συγκεκριμένο datacenter, το κόστος για το χώρο αποθήκευσης καθώς και το κώστος για το διαθέσιμο εύρος ζώνης. Επιπλέον χαρακτηρίζεται απο το δικό του μοναδικό όνομα.

1. Πόσες VM δημιουργούνται;

**Απάντηση:**

Συνολικά δημιουργούνται 20 Vms.

1. Τι χαρακτηριστικά έχει η κάθε μία;

**Απάντηση:**

Κάθε VM έχει το συνολικό μέγεθος εικόνας που θα έχει (10000mb), τη μνήμη του (512mb), την αρχιτεκτονική του συστήματος του, το διαθέσιμο bandwidth το οποίο θα έχει, τον αριθμό των επεξεργαστών καθώς και το όνομα του VMM.

1. Με τι αλγόριθμο γίνεται η προσπάθεια ανάθεσής τους σε κάποιον Host; (πως αποφασίζεται δηλαδή αν και σε ποιον Host θα τρέξει μία VM;)

**Απάντηση:**

Η προσπάθεια ανάθεσης σε ενός VM σε έναν HOST γίνεται σύμφωνα με το host που έχει τους λιγότερους δεσμευμένους επεξεργαστές τη χρονική στιγμή που γίνεται η ανάθεση. Σε περίπτωση που δε βρεθεί διαθέσιμος επεξεργαστής ελέγχει όλους τους Host μέχρις ότου βρεθέι, αν δε βρεθεί τότε τερματίζει.

1. Από ποιες παραμέτρους καθορίζεται το πόσες VM μπορεί να φιλοξενήσει κάθε Host;

**Απάντηση:**

Η παράμετρος οπου καθορίζει το πόσες VM μπορεί να φιλοξενήσει κάθε HOST είναι οι διαθέσιμοι επεξεργαστές, η διαθέσιμη μνήμη καθώς ,ο τύπος αρχιτεκτονικής καθώς και ο διαθέσιμος χώρος που υπάρχει σε κάθε Host.

1. Πόσες VMs θα τρέξουν τελικά και σε ποιον Host η κάθε μία (και γιατί);

**Απάντηση:**

Συνολικά θα τρέξουν 6 VM’s για κάθε datacenter στους Hosts:

*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 #0*

*0.1: Broker: VM #2 has been created in Datacenter #2, Host #0*

*0.1: Broker: VM #3 has been created in Datacenter #2, Host #1*

*0.1: Broker: VM #4 has been created in Datacenter #2, Host #0*

*0.1: Broker: VM #5 has been created in Datacenter #2, Host #1*

Τα VM’s απο 6-11 θα υποστηριχθούν στο DataCenter με id 3.

*0.2: Broker: VM #6 has been created in Datacenter #3, Host #0*

*0.2: Broker: VM #7 has been created in Datacenter #3, Host #0*

*0.2: Broker: VM #8 has been created in Datacenter #3, Host #0*

*0.2: Broker: VM #9 has been created in Datacenter #3, Host #1*

*0.2: Broker: VM #10 has been created in Datacenter #3, Host #0*

*0.2: Broker: VM #11 has been created in Datacenter #3, Host #1*

Οι άλλες 8 δεν θα τρέξουν επειδή οι Hosts δεν έχουν διαθέσιμη μνήμη και διαθέσιμους πόρους αρχιτεκτονικής.

Έτσι τα μηνύματα που παίρνουμε είναι:

*[VmScheduler.vmCreate] Allocation of VM #6 to Host #0 failed by RAM*

*[VmScheduler.vmCreate] Allocation of VM #6 to Host #1 failed by MIPS*

*[VmScheduler.vmCreate] Allocation of VM #7 to Host #0 failed by RAM*

*[VmScheduler.vmCreate] Allocation of VM #7 to Host #1 failed by MIPS etc…*

1. Πόσα Cloudlets δημιουργούνται;

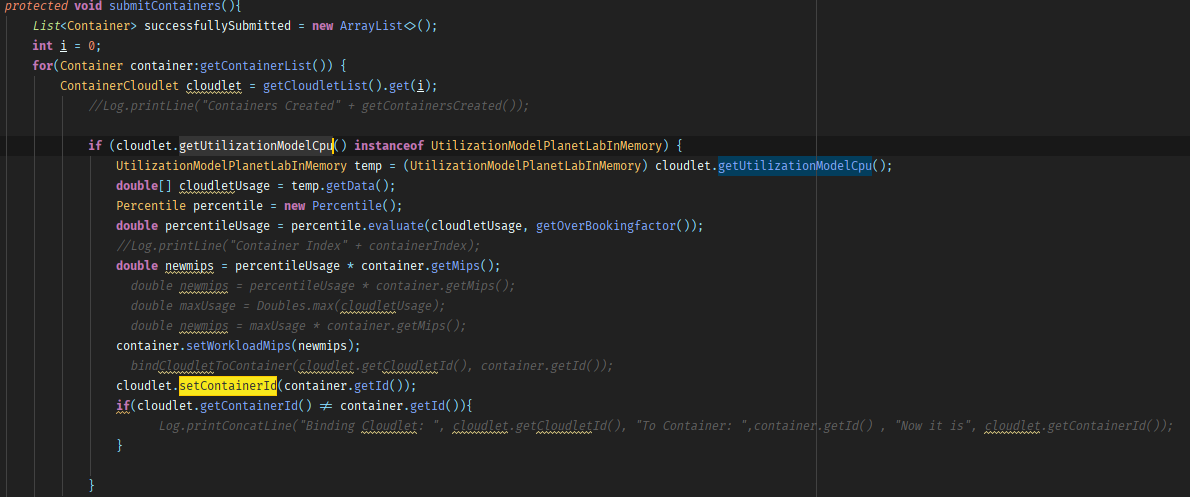
**Απάντηση:**

Συνολικά δημιουργούνται 40 Cloudlets.

1. Πως αποφασίζεται για κάθε Cloudlet σε ποια VM θα τρέξει;

**Απάντηση:**

Κάθε cloudlet καταχωρείται στο VM με το λιγότερο ποσοστό χρησιμοποιούμενης CPU το οποίο υπολογίζεται σύμφωνα με την αρχιτεκτονική του cloudlet.



1. Πόσα Cloudlets τελικά θα τρέξουν και σε ποια VM το κάθε ένα;

**Απάντηση:**

Θα τρέξουν και τα 40 cloudlets στα VM’s με σειρά απο 0 έως 11 όπου και είναι όλα τα διαθέσιμα VM.

1. Με τι πολιτική (space/time sharing) γίνεται η δρομολόγηση των Cloudlets στα VM στα οποία τελικά ανατέθηκαν να τρέχουν;

**Απάντηση:**

Η πολιτiκή που χρησιμοποιήθηκε για την ανάθεση των Cloudnets σε VM’s είναι TimeSharing.

1. Με τι πολιτική (space/time sharing) γίνεται η δρομολόγηση των VMs στα PEs (ή αλλιώς, η ανάθεση των PEs στα VMs) του Host στον οποίον τρέχουν;

**Απάντηση:**

Η πολιτική που χρησιμοποιήθηκε για την ανάθεση CPU’s στα VM’s είναι και αυτή TimeSharing.

1. Εξηγείστε (όσο μπορείτε λεπτομερέστερα) το τελικό μέρος των αποτελεσμάτων (output) του παραδείγματος - τι δίνει η κάθε στήλη κλπ, και ειδικότερα μεταξύ των άλλων τους χρόνους εκκίνησης περάτωσης των Cloudlets (σε άμεση συσχέτιση με τις πολιτικές δρομολόγησης που παρατηρήσατε ότι ακολουθούνται στα ερωτήματα 14. και 15.

**Απάντηση:**

Ας αναλύσουμε περαιτέρω το πρόγραμμα μας.

1. Αρχικά δημιουργούνται 2 Hosts όπου ο πρώτος έχει έναν 4πύρηνο επεξεργαστή και ο δεύτερος έναν διπύρηνο.
2. Κάθε επεξεργαστής έχει 1000 μονάδες αρχιτεκτονικής.
3. Κάθε Host έχει 2048 RAM, 1000000 storage, 10.000 bandwidth.
4. Δημιουργούνται 2 Datacenters με τα παραπάνω χαρακτηριστικά το κάθε ένα.
5. Δημιουργούνται 20 VM’s όπου το κάθε ένα απαιτεί:
6. Image Size: 10000
7. Ram: 512mb
8. Mips = 1000
9. Bandwidth: 10000
10. Processors: 1
11. Έπειτα, δημιουργούνται 40 cloudnets όπου το κάθε ένα έχει:
12. Length: 1000
13. File size = 300
14. Output size = 300
15. Processor number = 1

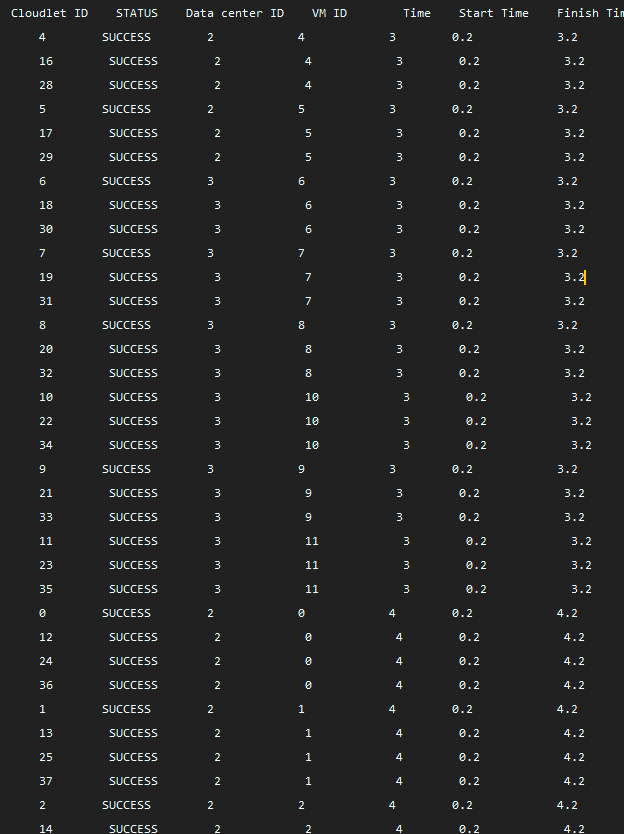
**Επομένως:**

Δημιουργήθηκαν 6 VM με 2 επεξεργαστές το κάθε ένα εκ των οποίων ο ένας είναι 4πύρηνος και ο άλλος 2πύρηνος το οποίο σημαίνει οτι σε αρχιτεκτονική Space Sharing θα καταφέρουν να «φορτωθούν» ομοιόμορφα σε κάθε VM. Οπότε κάθε VM θα έχει 40 / 12 = 3.33 cloudnets.

Οπότε ο καλύτερος δυνατός διαμοιρασμός είναι 3 cloudnets ανά VM και όσα περισσεύουν με 4 cloudnets όπως βλέπουμε στην εικόνα παρακάτω.

Το cloudnet Id είναι το μοναδικό χαρακτηριστικό κάθε cloudnet, το Status είναι η κατάσταση του όπου στην προκειμένη περίπτωση είναι όλα Success εφόσον όλα φορτώθηκαν επιτυχώς. Το DataCenter Id είναι το ID κάθε Data Center που στον οποίο ανήκει κάθε VM, το VM ID είναι το ID του κάθε VM που φορτώνει το αντίστοιχο Cloudlet και ο χρόνος είναι το χαρακτηριστικό με το οποίο αναφερόμαστε στο πόσο χρόνο χρειάστηκε για να εκτελεστεί το συγκεκριμένο process.

Παρατηρούμε οτι στα Vm’s όπου έχουν φορτωθεί 4 cloudlets χρειάζονται περισσότερο χρόνο για να εκτελεστούν όπως είναι και το αναμενόμενο.



1. Τρέξτε ξανά το παράδειγμα με τις υπόλοιπες εναλλακτικές δυνατότητες που σας παρέχονται (space/time sharing) για

α. δρομολόγηση των VMs στoυς Hosts (ανάθεση PEs)

**Απάντηση:**

Δρομολόγηση των VM’s στους Hosts με timesharing:

Αλλάζοντας μόνο τη δρομολόγηση των VM’s στους Hosts με timesharing δε παρατηρούμε καμία αλλαγή.



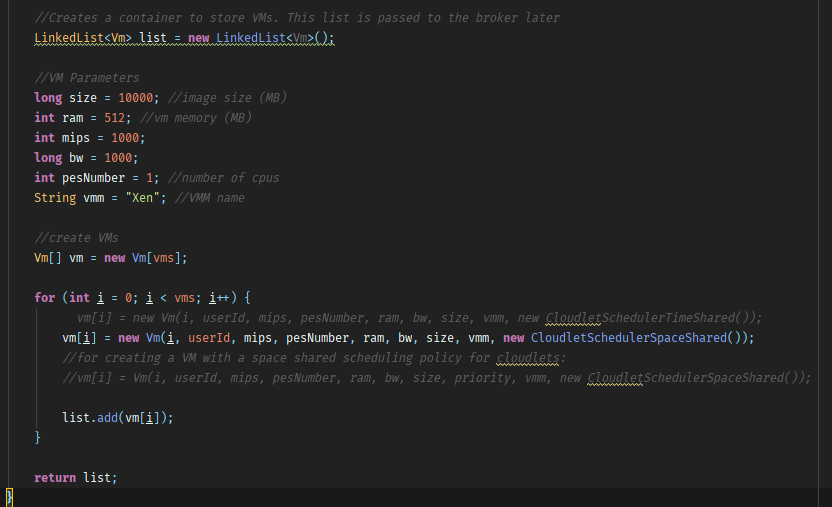
β. δρομολόγηση των Cloudlets στα VMs

**Απάντηση:**

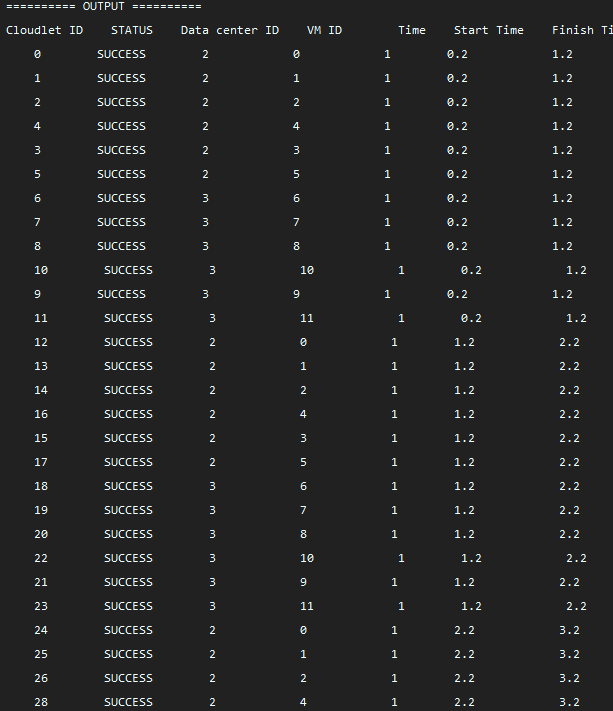
Δρομολόγηση των Cloudlets στα VM’s με space sharing:

Παρατηρούμε οτι ο χρόνος εκτέλεσης των τελικών cloudlets παραμένει ίδιος αλλά σημαιώνεται μια σημαντική ελαχιστοποίηση του χρόνου στα αρχικά cloudlets πράγμα το οποίο σίγουρα σημαίνει αύξηση απόδοσης.

Επίσης, όπως περιμέναμε με space sharing οι απόδοση των cloudlets σε VM’s γίνεται με τη σειρά, δηλαδή κάθε cloudlet εκχωρείται στο αμέσως επόμενο ελεύθερο VM’s την εκάστοτε χρονική στιγμή.



Στην παρακάτω εικόνα φαίνεται η διαφορά που αναλύσαμε παραπάνω.

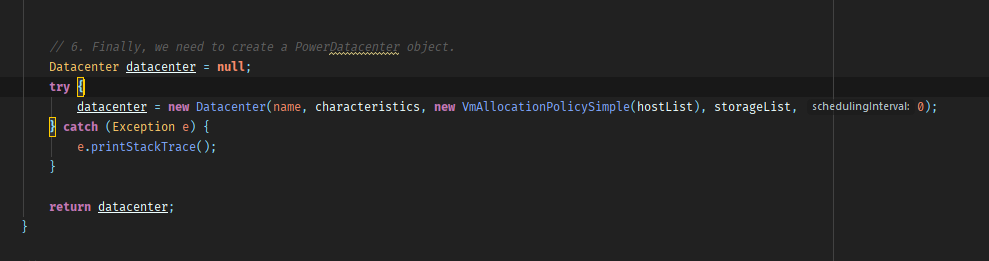


1. Σε ποια σημεία του κώδικα (υλοποίησης του simulator) και πώς θα επεμβαίνατε (α) για να αλλάξετε την πολιτική/αλγόριθμο για το 8. και (β) για να προσθέσετε κάποια άλλη δικιά σας πολιτική για τα 14. και 15.;

**Απάντηση:**

**Α)**

Για την αλλαγή του αλγορίθμου ανάθεσης σε VM’s θα τροποποιούσαμε τον παρακάτω κώδικα:



Όπου θα αλλάζαμε το VmAllocationPilicySimple σε κάποια άλλη συνάρτηση.

**Β)**

Η απάντηση δίνεται απο το ερώτημα 17. Οι αλλαγές στον κώδικα απο space sharing σε time sharing.

# Αποτελέσματα / Συμπεράσματα

Στη παρούσα εργασία είδαμε αναλυτικά το τρόπο με τον οποίο γίνεται ο διαμοιρασμός μνήμης είτε σε επίπεδο αρχιτεκτονικής επεξεργαστών είτε σε επίπεδο διαχείρισης των datacenters.

Είδαμε βελτιώσεις και αλλαγές που μπορούν να υλοποιηθούν σε συστήματα ανάλογα με τις απαιτήσεις και τις υποδομές που παρέχονται.

Μελετήσαμε την υλοποίηση σε κώδικα Java μέσω της βιβλιοθήκης CloudSim και είδαμε πώς γίνεται μια ακριβής αναπαράσταση των τεχνικών αυτών.

Κατανοήσαμε την δομή και το τρόπο διασύνδεσης όλων των επιμέρους συστημάτων και πώς αυτά αλληλεπιδρούν μεταξύ τους.