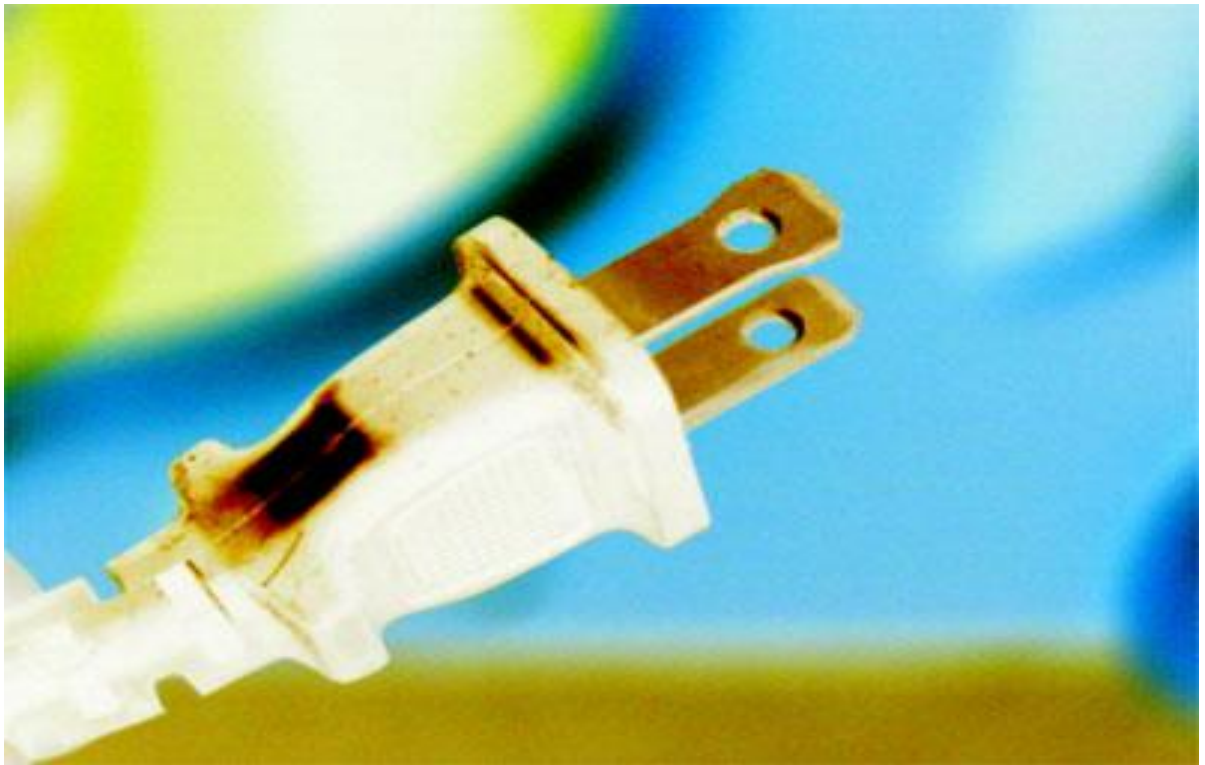


Πανεπιστήμιο Μακεδονίας  
ΔΠΜΣ Πληροφοριακά Συστήματα  
Δίκτυα Υπολογιστών  
Καθηγητής: Α.Α Οικονομίδης

**Παρουσίαση και ταξινόμηση αλγορίθμων εξισορρόπησης φορτίου σε  
συστήματα “Σύννεφου” και ομότιμα δίκτυα**

**Presentation and classification of load Balancing Algorithms in  
clouds and P2P**



**Ψαρά Κωνσταντινιά Α.Μ 08/13**

**Μάιος 2014**

## Περιεχόμενα

1.Περίληψη.....	3
2.Εισαγωγή.....	4
3.Υπολογιστική σύννεφου “Cloud computing”.....	5
4.Εξισορρόπηση φορτίου σε “cloud”.....	7
5.Αλγόριθμοι εξισορρόπησης φορτίου σε “cloud”.....	7
5.1Αλγόριθμος Round Robin.....	7
5.2Αλγόριθμος DLBDM.....	7
5.3Αλγόριθμος Min-Min.....	8
5.4Αλγόριθμος Max-Min.....	8
5.5Αλγόριθμος Biased Random Sampling.....	8
5.6Αλγόριθμος OLB.....	9
5.7Αλγόριθμος Particle Swarm Optimization.....	9
6.Συγκριτικός πίνακας αλγορίθμων εξισορρόπησης φορτίου σε “cloud”.....	10
7.Συστήματα peer to peer.....	10
7.1. Γενικά.....	11
7.2 Κατηγορίες ομότιμων συστημάτων.....	11
8.Εξισορρόπηση φορτίου σε ομότιμα συστήματα.....	12
9.Αλγόριθμοι εξισορρόπησης φορτίου σε ομότιμα συστήματα.....	12
9.1Αλγόριθμοι Round Robin and Randomized.....	12
9.2 Αλγόριθμος Central Manager.....	13
9.3Αλγόριθμος Threshold.....	13
9.4 Αλγόριθμος Central Queue.....	14
9.5 Αλγόριθμος Local Queue.....	15

<b>10. Συγκεντρωτικός πίνακας αλγορίθμων εξισορρόπησης φορτίου σε συστήματα “peer to peer” .....</b>	<b>15</b>
<b>11. Συμπεράσματα.....</b>	<b>16</b>
<b>12.Βιβλιογραφία.....</b>	<b>17</b>

## 1.Περίληψη

Τα κατανεμημένα συστήματα cloud και peer to peer έχουν αρχίσει να αναπτύσσονται ραγδαία στις μέρες μας. Τα συστήματα cloud λειτουργούν ανάμεσα σε έναν client(ο προσωπικός μας υπολογιστής) και έναν απομακρυσμένο server, για να πετύχουν σωστή και αποδοτική λειτουργία καθώς διαχειρίζονται έναν μεγάλο όγκο δεδομένων είναι αναγκαία η χρήση αλγορίθμων εξισορρόπησης φορτίου. Οι αλγόριθμοι αυτοί χρησιμοποιούνται ανάλογα με το φορτίο και την τοπολογία του συστήματος για να καταφέρουν με την εξισορρόπηση φορτίου. Τα συστήματα peer to peer είναι κατάλληλα για την διανομή περιεχομένου και την κοινή χρήση των υποδομών, ένα γνωστό σε όλους μας peer to peer σύστημα είναι το Skype. Η δυναμικότητα των συστημάτων αυτών κάνει αναγκαία τη εξισορρόπηση φορτίου. Η εξισορρόπηση φορτίου πετυχαίνεται με την χρήση δυναμικών και στατικών αλγορίθμων ανάλογα με την φύση του συστήματος.

### Abstract

Nowadays distributed, cloud and peer to peer computing systems have begun to grow rapidly. Cloud systems operate between a client ( our personal computer) and a remote server, to achieve correct and efficient operation while they are managing a large volume of data is necessary the use load balancing algorithms. These algorithms are used depending on the load and the topology of the system. Peer to peer systems are suitable for content distribution and sharing of infrastructure a familiar to all of us peer to peer system is Skype. Those systems capacity makes load balancing necessary .

## 2.Εισαγωγή

Το πεδίο των παράλληλων και κατανεμημένων συστημάτων πληροφορικής έχει αλλάξει με την εμφάνιση του “cloud” .Τα συστήματα αυτά μας επιτρέπουν να έχουμε πρόσβαση και να χρησιμοποιούμε web εφαρμογές χωρίς να τις διαθέτουμε στον υπολογιστή μας ή σε κάποια άλλη συσκευή που είναι διασυνδεδεμένη με το ίντερνετ. Τα συστήματα “cloud” έχουν γίνει δημοφιλή γιατί είναι οικονομικά, έχουν μεγάλο αποθηκευτικό χώρο, μας δίνουν πρόσβαση από οποιαδήποτε συσκευή συνδεδεμένη στο διαδίκτυο και είναι πολύ ευέλικτα. Βέβαια και αυτά έχουν τα προβλήματα τους όπως η ασφάλεια και η αυξημένη πολυπλοκότητα. Το γεγονός πως όλο και περισσότεροι χρήστες χρησιμοποιούν τα συστήματα “cloud” και τα δεδομένα που χειρίζονται αυτά τα συστήματα αυξάνονται, κάνει την εξισορρόπηση φορτίου απαραίτητη για την σωστή λειτουργία τους.

Τα κατανεμημένα συστήματα peer to peer είναι σημαντικά για την διανομή περιεχομένου και την κοινή χρήση των υποδομών. Η λειτουργία τους είναι απλή, αν και η λειτουργία τους είναι σαν το κλασικό μοντέλο client server στην πραγματικότητα ο client(ο προσωπικός μας υπολογιστής) λειτουργεί και σαν τα δύο γιατί είτε στέλνει αιτήματα είτε δέχεται. Αυτή την λειτουργία την καταλαβαίνουμε από την λειτουργία του Skype. Όμως στα συστήματα αυτά η κατανομή του φορτίου δεν γίνεται ομοιόμορφα και έτσι για να μπορέσουμε να δώσουμε λύση σε αυτό το θέμα χρησιμοποιούμε την εξισορρόπηση φορτίου για να καταφέρουμε την καλύτερη κατανομή του φορτίου στους κόμβους.

### 3.Υπολογιστική σύννεφου “Cloud computing”

Η έννοια της υπολογιστικής σύννεφου (**cloud computing**) έχει αλλάξει σημαντικά το πεδίο των παράλληλων και κατανεμημένων συστημάτων πληροφορικής. Στόχος της είναι να μοιράζεται στοιχεία μεταξύ ενός client(ο κάθε χρήστης) και ενός server(ένας server που δεν ξέρουμε που είναι). Είναι ένα οικονομικό μοντέλο για την απόκτηση και διαχείριση πόρων πληροφορικής. Με πολύ απλά λόγια το (**cloud computing**) είναι μία δομή, με την οποία μας δίνεται η δυνατότητα να έχουμε πρόσβαση και να χρησιμοποιούμε web εφαρμογές χωρίς να τις διαθέτουμε στον υπολογιστή μας ή σε κάποια άλλη συσκευή που είναι διασυνδεδεμένη με το ίντερνέτ (won 2009).

Φυσικά υπάρχουν πολλά πλεονεκτήματα που έχουμε από την χρήση της τεχνολογίας αυτής, αλλά δεν μπορούμε να παραλείψουμε και τα μειονεκτήματα της, παρακάτω βλέπουμε μερικά από αυτά.

#### Πλεονεκτήματα:

- ❖ **Οικονομία.** Για την αγορά ενός λογισμικού μία εταιρία μπορεί να σπαταλήσει πολλά χρήματα, με το (**cloud**) όμως έχει την δυνατότητα να πληρώσει μόνο για την χρήση της εφαρμογής.
- ❖ **Μεγάλος Αποθηκευτικός Χώρος.** Με το (**cloud computing**) έχουμε συνήθως όσο αποθηκευτικό χώρο θα χρειαστούμε.
- ❖ **Πρόσβαση από οποιαδήποτε συσκευή διαθέτει σύνδεση στο ίντερνετ.** Τα δεδομένα που υπάρχουν σε (**cloud**) βρίσκονται σε έναν απομακρυσμένο server και είναι προσβάσιμα από οποιονδήποτε client(H/Y).

#### Μειονεκτήματα:

- ❖ **Ασφάλεια δεδομένων.** Κάποια ευαίσθητα δεδομένα είναι λογικό να τα έχουμε σε προσωπικό σκληρό δίσκο.
- ❖ **Εξάρτηση από τον πάροχο.** Αν αυτός που παρέχει τις υπηρεσίες cloud δεν μπορεί να εξυπηρετήσει τους πελάτες του αυτοί είναι πλέον με δεμένα τα χέρια χωρίς να μπορούν να έχουν πρόσβαση σε δεδομένα και υπηρεσίες.

## 4.Εξισορρόπηση φορτίου σε “cloud”

Η εξισορρόπηση φορτίου είναι μια μέθοδος που χρησιμοποιείτε για την δικτύωση υπολογιστών με σκοπό την διανομή του φορτίου εργασίας σε πολλούς υπολογιστές. Στόχος της εξισορρόπησης φορτίου είναι η βελτίωση της απόδοσης του φορτίου μεταξύ των διάφορων πόρων (δίκτυο, σύνδεσμοι, κεντρικές μονάδες επεξεργασίας, μονάδες δίσκου) ( Ms.SHAVETA 2012).

Διάφορα θέματα του cloud computing μπορούν να λυθούν με την εξισορρόπηση φορτίου. Ένα βασικό πρόβλημα στα cloud, είναι η τυχαία άφιξη φόρτου εργασίας που επιβαρύνει τους server και άρα το σύστημα. Επίσης ένα άλλο πρόβλημα που έχουν να αντιμετωπίσουν τα συστήματα cloud είναι η διαχείριση μεγάλου όγκου δεδομένων, οπότε για την αντιμετώπιση προβλημάτων σαν αυτά είναι αναγκαία η χρήση αλγορίθμων εξισορρόπησης φορτίου. Η εξισορρόπηση φορτίου είναι ένας μηχανισμός που διανέμει το τοπικό δυναμικό φορτίο ομοιόμορφα σε όλους τους κόμβους έτσι ώστε να μην έχουμε κόμβους βαριά φορτωμένους και κάποιους άλλους σε αδράνεια. Οπότε βλέπουμε πως το ζητούμενό μας είναι η ισομερής κατανομή του φορτίου στους κόμβους του δικτύου. Για να πετύχουμε την ισομερή αυτή κατανομή χρησιμοποιούνται διάφοροι αλγόριθμοι όπως θα δούμε και στην συνέχεια της εργασίας.

Γενικά οι αλγόριθμοι εξισορρόπησης φορτίου ταξινομούνται ανάλογα με το φορτίο και την τοπολογία του συστήματος. Παρακάτω βλέπουμε αναλυτικά αυτή την ταξινόμηση.

### Ταξινόμηση ανάλογα με το φορτίο του Συστήματος.

- ❖ **Συγκεντρωτική προσέγγιση :** Ένας κόμβος είναι υπεύθυνος για τη διαχείριση της διανομής εντός του συστήματος.
- ❖ **Κατανεμημένη προσέγγιση :** Κάθε κόμβος χτίζει ανεξάρτητα δικό του φορτίο συλλέγοντας πληροφορίες από τα φορτία των άλλων κόμβων. Οι αποφάσεις λαμβάνονται σε τοπικό επίπεδο με τη χρήση των τοπικών φορέων φορτίου. Η προσέγγιση αυτή είναι περισσότερο κατάλληλη για ευρέως κατανεμημένα συστήματα όπως το cloud computing.
- ❖ **Μικτή προσέγγιση:** Ένας συνδυασμός από τις δύο παραπάνω προσεγγίσεις για να πάρουμε τα οφέλη και από τις δύο.

### Ταξινόμηση Σύμφωνα με την Τοπολογία συστήματος.

- ❖ **Στατική προσέγγιση :** Η προσέγγιση αυτή ορίζεται γενικά στο σχεδιασμό ή την υλοποίηση του συστήματος.
- ❖ **Δυναμική προσέγγιση :** Η προσέγγιση αυτή λαμβάνει υπόψη την τρέχουσα κατάσταση του συστήματος κατά τη διάρκεια της απόφασης για εξισορρόπηση φορτίου. Η προσέγγιση αυτή είναι περισσότερο κατάλληλη για ευρέως κατανεμημένα συστήματα, όπως το cloud computing.

- ❖ **Προσαρμοστική προσέγγιση :** Η προσέγγιση αυτή προσαρμόζει την κατανομή του φορτίου στις αλλαγές της κατάστασης του συστήματος, αλλάζοντας τις παραμέτρους δυναμικά και ακόμη και τους αλγόριθμους.. Η προσέγγιση αυτή είναι περισσότερο κατάλληλη για ευρέως κατανεμημένα συστήματα όπως cloud computing.

## **5.Αλγόριθμοι εξισορρόπησης φορτίου σε “cloud”**

### **5.1Αλγόριθμος Round Robin**

Αυτός ο αλγόριθμος θέλει να πετύχει την ισομερή κατανομή του φορτίου στους κόμβους. Για να το πετύχει αυτό κατανέμει τις διεργασίες μεταξύ όλων των επεξεργαστών. Κάθε δραστηριότητα κατανέμετε στον επεξεργαστή με σειρά (**round robin**). Η σειρά κατανομής της διαδικασίας διατηρείται τοπικά ανεξάρτητη από τις χορηγήσεις από απομακρυσμένους επεξεργαστές. Ο αλγόριθμος αυτός χρησιμοποιείται κυρίως σε web servers όπου τα αιτήματα http είναι παρόμοια και κατανέμονται όμοια.

Ο Round Robin κατανέμει το φορτίο σε ίσες εργασίες στους κόμβους όπου είναι και το ζητούμενο μας αλλά ο χρόνος για την επεξεργασία είναι διαφορετικός, έτσι σε κάποια χρονική στιγμή μερικοί κόμβοι είναι πολύ φορτωμένοι και κάποιοι άλλοι είναι σε αδράνεια.

### **5.2Αλγόριθμος DLBDM**

Στόχος αυτού του αλγόριθμου είναι να αλληλεπιδρούν όλα τα μέρη του συστήματός μας, συμπεριλαμβανομένων και των εξισορροπητών φορτίου και των διακομιστών εφαρμογών. Βάση των πληροφοριών που έχει συλλέξει και των εσωτερικών υπολογισμών ο αλγόριθμος θα προωθήσει τις αποφάσεις στους εξισορροπητές φορτίου. Στον αλγόριθμο αυτόν δεν επηρεάζεται μόνο το φορτίο εξισορρόπησης αλλά και οι πόροι του εικονικού διακομιστή ο οποίος τώρα βρίσκεται εκτεθειμένος, έτσι παρά την προσπάθειά του για την εξισορρόπηση φορτίου αν ο DLBDM δεν έχει κατασκευαστεί με αξιοπιστία και ελαστικότητα μπορεί να οδηγήσει ακόμα και στην μη διαθεσιμότητα του συστήματος.

### **5.3Αλγόριθμος Min-Min**

Ο αλγόριθμος αυτός ξεκινάει με τις εργασίες που δεν είναι εκχωρημένες και βρίσκει τον ελάχιστο χρόνο ολοκλήρωσής τους. Συγκρίνει τις τιμές αυτές και επιλέγει την ελάχιστη τιμή τους. Τώρα βάση της ελάχιστης τιμής, δηλαδή του ελάχιστου χρόνου, το έργο προγραμματίζεται για το ανάλογο μηχάνημα, δηλαδή ένα μηχάνημα που θα λειτουργεί βάση του ελάχιστου χρόνου ολοκλήρωσης μίας διεργασίας. Τότε ο χρόνος εκτέλεσης για όλες τις άλλες εργασίες ενημερώνεται σε αυτό το μηχάνημα, επίσης προσθέτουμε τον χρόνο εκτέλεσης του ανατιθέμενου έργου και μία μία οι



διεργασίες βάση του ελάχιστου χρόνου αφαιρούνται από το μηχάνημα και ανατίθενται στους πόρους. Η δουλειά που κάνει αυτός ο αλγόριθμος είναι η κατανομή των διεργασιών στους πόρους πετυχαίνοντας βέλτιστο χρόνο, αυτό όμως είναι πάντα θεμιτό ή μήπως πρέπει να προσέχουμε και άλλα πράγματα κατά την εξισορρόπηση φορτίου;

### **5.4 Αλγόριθμος Max-Min**

Είναι σχεδόν ίδιος με τον αλγόριθμο min-min, εκτός από τα εξής: Αφού διαπιστώσει τους ελάχιστους χρόνους εκτέλεσης επιλέγεται η μέγιστη τιμή που είναι ο μέγιστος χρόνος ανάμεσα σε όλα τα καθήκοντα σε οποιουδήποτε πόρο. Τότε σύμφωνα με αυτό το μέγιστο χρονικό διάστημα, το έργο έχει προγραμματιστεί για το αντίστοιχο μηχάνημα. Άρα εδώ βλέπουμε πως ακλουθείτε η ίδια διαδικασία με τον αλγόριθμο Min-min απλά αντί για τον μικρότερο ελάχιστο χρόνο διαλέγουμε τον μεγαλύτερο.

### **5.5 Αλγόριθμος Biased Random Sampling**

Ο (M. Randles) ερεύνησε μια κατανεμημένη και επεκτάσιμη προσέγγιση εξισορρόπησης φορτίου η οποία χρησιμοποιεί τυχαία δειγματοληψία για την επίτευξη αυτό-οργάνωσης με την εξισορρόπηση του φορτίου σε όλους τους κόμβους του συστήματος. Εδώ ένα εικονικό γράφημα κατασκευάζεται, με τη συνδεσιμότητα του κάθε κόμβου (ένας διακομιστής αντιμετωπίζεται ως κόμβος) που αντιπροσωπεύει το φορτίο στο διακομιστή. Κάθε διακομιστής συμβολίζεται ως κόμβος στον γράφο, με τον καθένα να κατευθύνεται προς τους ελεύθερους πόρους του διακομιστή. Σε σχέση με τους παραπάνω αλγόριθμους εδώ βλέπουμε μία κατανεμημένη και δυναμική προσέγγιση, ο αλγόριθμος αυτός είναι ποιο αποδοτικός σε συστήματα που διαχειρίζονται μεγάλο όγκο δεδομένων.

### **5.6 Αλγόριθμος OLB**

Ο ευκαιριακός αλγόριθμος εξισορρόπησης φορτίου OLB ανεξάρτητα από το φόρτο εργασίας του κάθε κόμβου τους διατηρεί όλους απασχολημένους δηλαδή δεν λαμβάνει υπόψη το φόρτο εργασίας και την καταλληλότητα του κάθε κόμβου. Ο στόχος του είναι απλός, να κάνει την εξισορρόπηση φορτίου, αλλά για κάθε εργασία δεν υπολογίζει τον αναμενόμενο χρόνο εκτέλεσης και έτσι έχει έναν υψηλότερο χρόνο ολοκλήρωσης.

## 5.7 Αλγόριθμος Particle Swarm Optimization

Εδώ έχουμε έναν βελτιστοποιημένο αλγόριθμο που βασίζεται στην νοημοσύνη σμήνους (swarm intelligence). Ο αλγόριθμος αυτός μπορεί να αντιμετωπίσει το πρόβλημα του ποιος είναι ο καλύτερος τρόπος αναπαράστασης, σημείο ή επιφάνεια σε μία  $n$ -διάσταση. Για κάθε φορτίο μπορούμε να υπολογίσουμε την αρχική ταχύτητα του καθώς η πορεία του είναι γνωστή. Εδώ έχουμε ακόμα έναν αλγόριθμο με κατανεμημένη και δυναμική προσέγγιση που λειτουργεί πολύ διαφορετικά από τους αλγόριθμους που είδαμε παραπάνω, εδώ μπορούμε να πούμε πως έχουμε έναν ευφυή αλγόριθμο.

## 6. Συγκριτικός πίνακας αλγορίθμων εξισορρόπησης φορτίου σε “cloud”

Παραπάνω είδαμε κάποιους αλγόριθμους εξισορρόπησης φορτίου που είναι κατάλληλοι στο να λύσουν τα προβλήματα κατανομής φορτίου στα cloud δίκτυα. Οι αλγόριθμοι αυτοί τώρα θα συγκριθούν ανάλογα με την ταξινόμηση που είδαμε παραπάνω. Δηλαδή θα δούμε που ταξινομούνται ανάλογα με το φορτίο και την τοπολογία.

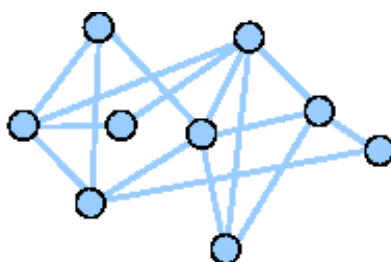
Αλγόριθμος	Συγκεντρωτική προσέγγιση	Κατανεμημένη προσέγγιση	Στατική προσέγγιση	Δυναμική προσέγγιση
Round Robin	✓		✓	
DLBDM	✓		✓	
Min-Min	✓		✓	
Max-Min	✓		✓	
Biased Random Sampling		✓		✓
OLB	✓		✓	
Particle Swarm Optimization		✓		✓

Από τον παραπάνω πίνακα παρατηρούμε πως οι αλγόριθμοι Biased Random Sampling και Particle Swarm Optimization έχουν διαφορετική ταξινόμηση από τους άλλους όπως είχαμε πει και κατά την ανάλυση τους . Οι αλγόριθμοι που είδαμε επιτυγχάνουν την εξισορρόπηση φορτίου είτε με υπολογισμό μέγιστου-ελάχιστου χρόνου είτε με το μέγεθος του φορτίου είτε με σειρά προτεραιότητας κ.α. γενικά υπάρχουν πολλοί ακόμα αλγόριθμοι που θα μπορούσαμε να αναφερθούμε αλλά δεν χωράνε όλοι σε μία εργασία

## 7.Συστήματα peer to peer

### 7.1.Γενικά

Τα συστήματα p2p είναι καταναμημένα συστήματα στα οποία οι κόμβοι έχουν ίσους ρόλους και ικανότητες και ανταλλάσσουν πληροφορίες μεταξύ τους. Τα συστήματα αυτά προσεγγίζουν διαφορετικά τον όρο client-server και αυτό γιατί ο κάθε υπολογιστής που συνδέεται σε ένα τέτοιο δίκτυο λειτουργεί και ως client και ως sever. Αυτό οφείλετε στην φύση των συστημάτων αυτών όπου δύο ή περισσότεροι υπολογιστές επικοινωνούν μεταξύ τους χωρίς κάποιον server ανάμεσά τους, αν σκεφτούμε πως λειτουργεί το «Skype» θα καταλάβουμε τι είναι τα ομότιμα συστήματα. Σύμφωνα με τον (Khaled Ragab 2011) τα ομότιμα δίκτυα θεωρούνται η πιο σημαντική εξέλιξη για την διανομή περιεχομένου και την κοινή χρήση των υποδομών. Κάθε κόμβος έχει πολλές φορές διαφορετικές δυνατότητες πόρων όπως για παράδειγμα, επεξεργασίας , αποθήκευσης , εύρους ζώνης. Έτσι είναι απαραίτητο κάθε κόμβος να έχει φορτίο ανάλογο των δυνατοτήτων του (Brighten Godfrey 2012). Λόγω του ότι από την φύση τους τα συστήματα P2P είναι δυναμικά δεν είναι εύκολο να εξασφαλιστεί ότι το φορτίο κατανέμετε ομοιόμορφα σε όλο το σύστημα. Παρακάτω βλέπουμε μια εικόνα για το πώς μοιάζει ένα δίκτυο peer to peer.



απεικόνιση ενός δικτύου peer to peer

## 7.2 Κατηγορίες ομότιμων συστημάτων

Τα P2P συστήματα χωρίζονται σε δύο κατηγορίες, τα κεντρικοποιημένα συστήματα και τα μη κεντρικοποιημένα, αδόμετα συστήματα, αναλυτικά έχουμε.

### ❖ Κεντρικοποιημένα συστήματα:

Παράδειγμα κεντρικοποιημένου συστήματος είναι το Napster που ανακάλυψε ο Shawn Fanning. Αυτού του είδους τα δίκτυα διατηρούν έναν κατάλογο-ευρετήριο σε μια κεντρική τοποθεσία ο οποίος κρατά πληροφορίες για όλα τα δεδομένα που βρίσκονται αποθηκευμένα στους κόμβους. Ο κατάλογος-ευρετήριο δημιουργείται με δύο τρόπους: είτε με τη συνεργασία των κόμβων που του παρέχουν τακτικά μια λίστα με τα δεδομένα που προσφέρουν στο δίκτυο, είτε με αναζήτηση στο δίκτυο όπως συμβαίνει με μια μηχανή αναζήτησης στο Internet.

### ❖ Μη κεντρικοποιημένα συστήματα

Τα συστήματα αυτά δε διαθέτουν κάποιον κεντρικό κατάλογο-ευρετήριο, αυτό είναι και το γεγονός που τα διαφοροποιεί από τα κεντρικοποιημένα.

## 8. Εξισορρόπηση φορτίου σε ομότιμα συστήματα

Λόγο της δυναμικής φύσης των συστημάτων P2P είναι δύσκολο να εξασφαλιστεί ότι το φορτίο κατανέμεται ομοιόμορφα σε όλο το σύστημα. Για το λόγο αυτό υπάρχει ανάγκη για εξισορρόπηση φορτίου. Οι τεχνικές εξισορρόπησης φορτίου στα συστήματα P2P θα πρέπει να είναι επεκτάσιμες και να μπορούν να αντιμετωπίσουν το μεγάλο μέγεθος των συστημάτων. Θα πρέπει να τοποθετήσουν ή να αντικαταστήσουν κοινά δεδομένα βέλτιστα μεταξύ των κόμβων διατηρώντας παράλληλα αποτελεσματικούς πίνακες επικάλυψης δρομολόγησης για να γίνει ανακατεύθυνση στον σωστό κόμβο. Οι καθυστερήσεις επικοινωνίας μεταξύ των ομότιμων κόμβων (**peers**) μεταβάλλει σημαντικά την αναμενόμενη απόδοση της εξισορρόπησης φορτίου στα συστήματα. Λόγο αυτής της καθυστέρησης η πληροφορία που έχει ένας ομότιμος κόμβος για τους άλλους ομότιμους κόμβους μπορεί να μην αντιπροσωπεύει την τρέχουσα κατάσταση. Από αυτά που είπαμε παραπάνω παρατηρούμε πως η εξισορρόπηση φορτίου είναι ένα κρίσιμο ζήτημα για την αποτελεσματική λειτουργία των συστημάτων P2P

## 9.Αλγόριθμοι εξισορρόπησης φορτίου σε ομότιμα συστήματα

Οι αλγόριθμοι εξισορρόπησης φορτίου που θα μελετήσουμε παρακάτω για τα ομότιμα συστήματα χωρίζονται στις εξής δύο κατηγορίες:

- ❖ Στατικοί αλγόριθμοι  
Στους στατικούς αλγόριθμους, τα καθήκοντα που έχουν ανατεθεί στους κόμβους πριν την έναρξη της εκτέλεσης τους δεν μπορούν να αναιρεθούν. Στόχος της στατικής μεθόδου εξισορρόπησης φορτίου είναι να μειωθεί ο συνολικός χρόνος εκτέλεσης προγραμμάτων και να ελαχιστοποιηθούν οι καθυστερήσεις επικοινωνίας.
- ❖ Δυναμικοί αλγόριθμοι  
Οι δυναμικοί αλγόριθμοι διαφέρουν από τους στατικούς στο ότι το φορτίο εργασίας κατανέμετε στους κόμβους κατά την εκτέλεση.

### 9.1 Αλγόριθμοι Round Robin and Randomized

Στον round robin οι διεργασίες κατανέμονται ομοιόμορφα στους κόμβους. Κάθε καινούρια διεργασία ανατίθεται στους κόμβους με προτεραιότητα round robin. Ο τρόπος με τον οποίο γίνεται η κατανομή των διεργασιών στους κόμβους έχει στόχο την ανεξαρτητοποίηση του κάθε κόμβου τοπικά από τους απομακρυσμένους. Ο αλγόριθμος αυτός λειτουργεί καλά σε περιπτώσεις όπου οι διεργασίες είναι περισσότερες από τους κόμβους. Ένα θετικό του round robin είναι πως οι διεργασίες μπορούν να λειτουργούν χωρίς επικοινωνία μεταξύ τους, επίσης ο round robin και ο randomized μπορούν να πετύχουν καλύτερη απόδοση από άλλους αλγόριθμους εξισορρόπησης φορτίου για συγκεκριμένες εφαρμογές ειδικού τύπου. Αλλά σε γενικές περιπτώσεις οι δύο αυτοί αλγόριθμοι δεν μπορούν να πετύχουν καλή απόδοση.

### 9.2 Αλγόριθμος Central Manager

Στον αλγόριθμο αυτό ένας κεντρικός κόμβος είναι υπεύθυνος για την διανομή των διεργασιών στους επεξεργαστές. Ο πρώτος κόμβος που επιλέγεται όταν δημιουργηθεί μία διεργασία είναι αυτός με το μικρότερο φορτίο. Ο κεντρικός κόμβος διαμοιράζει τις διεργασίες έτσι ώστε το φορτίο των κόμβων να παραμένει στο ίδιο επίπεδο για όσο περισσότερο γίνεται. Κάθε φορά που το φορτίο αλλάζει σε έναν κόμβο αυτός ενημερώνει τον κεντρικό έτσι ώστε αυτός να πάρει την απόφαση του για την εξισορρόπηση φορτίου. Το μειονέκτημα σε αυτό τον αλγόριθμο είναι ο υψηλός βαθμός επικοινωνίας μεταξύ των διεργασιών που μπορεί να οδηγήσει σε μία κατάσταση συμφόρησης. Ο αλγόριθμος αυτός αναμένετε να έχει καλύτερες επιδόσεις από παράλληλες εφαρμογές ειδικά όταν οι δυναμικές δραστηριότητες δημιουργούνται από διαφορετικούς ξενιστές.

### 9.3 Αλγόριθμος Threshold

Σε αυτό τον αλγόριθμο οι διεργασίες με την δημιουργία τους ανατίθενται αμέσως σε κόμβους. Οι κόμβοι επιλέγονται τοπικά χωρίς την αποστολή κάποιου μηνύματος. Κάθε κόμβος κρατάει αντίγραφο από το φορτίο του συστήματος. Το φορτίο κάθε κόμβου χαρακτηρίζεται από τα εξής τρία επίπεδα.

- ❖ Χαμηλά φορτισμένο
- ❖ Μέτρια φορτισμένα
- ❖ Υψηλά φορτισμένο

Στιγμιαία όλοι οι κόμβοι θεωρούνται πως είναι χαμηλά φορτισμένοι. Όταν ένας κόμβος έχει φορτίο παραπάνω από το όριο στέλνει μήνυμα στους απομακρυσμένους κόμβους με την νέα του κατάσταση. Αν η κατάσταση τοπικά δεν είναι υπερφορτωμένη η διεργασία τοποθετείτε τοπικά, αλλιώς επιλέγεται ένας χαμηλά φορτισμένος απομακρυσμένος κόμβος, αν δεν βρεθεί τέτοιος κόμβος η διεργασία παραμένει τοπικά. Ένα μειονέκτημα του αλγόριθμου αυτού είναι πως όλες οι διεργασίες τοποθετούνται τοπικά όταν οι απομακρυσμένοι κόμβοι είναι υπερφορτωμένοι. Ακόμα ένα μειονέκτημα είναι πως το φορτίο σε έναν υψηλά φορτισμένο κόμβο μπορεί να είναι πολύ υψηλότερο από ότι σε έναν άλλο υψηλά φορτισμένο κόμβο, με αποτέλεσμα να δημιουργούνται σημαντικές διαταραχές στην εξισορρόπηση φορτίου και επίσης αυξάνετε ο χρόνος εκτέλεσης μίας εφαρμογής.

### 9.4 Αλγόριθμος Central Queue

Ο αλγόριθμος αυτός αποθηκεύει νέες δραστηριότητες και ανεκπλήρωτες αιτήσεις στον κεντρικό κόμβο με προτεραιότητα FIFO ,δηλαδή όποιος έρχεται πρώτος εξυπηρετείτε και πρώτος. Μόλις δημιουργείται τώρα μία διεργασία μπαίνει στην ουρά και περιμένει να έρθει η σειρά της. Όταν το σύστημα ζητά να αρχίσει μία δραστηριότητα επιλέγεται η πρώτη, αν όμως δεν υπάρχει καμία διαθέσιμη το σύστημα μπαίνει σε κατάσταση αναμονής και περιμένει τότε θα είναι έτοιμη μία διεργασία. Έτσι βλέπουμε πως αυτός εδώ ο αλγόριθμος λειτουργεί διαφορετικά από αυτούς που παρουσιάσαμε παραπάνω και αυτό γιατί είναι δυναμικής φύσης σε αντίθεση με τους προηγούμενους. Το θετικό εδώ είναι πως οι διεργασίες ταξινομούνται και είναι άμεσα διαθέσιμες, αν όμως ο όγκος των διεργασιών είναι μεγάλος θα δημιουργηθεί συμφόρηση.

## 9.5 Αλγόριθμος Local Queue

Η βασική ιδέα αυτού του αλγόριθμου είναι η στατική κατανομή όλων των νέων διεργασιών με την μετανάστευσή τους από έναν κόμβο που το φορτίο του είναι στο κατώτερο όριο. Η παράμετρος, φορτίο, ορίζει τον ελάχιστο αριθμό των έτοιμων διεργασιών που επιχειρεί να δώσει ο διαχειριστής φορτίου. Αρχικά οι διεργασίες που δημιουργούνται στον κεντρικό κόμβο τοποθετούνται στους χαμηλά φορτισμένους κόμβους του συστήματος. Οι πρώτες παράλληλες δραστηριότητες που δημιουργούνται από τον κεντρικό κόμβο είναι συνήθως αρκετές για να διαμοιραστούν στους απομακρυσμένους κόμβους. Όλες οι άλλες δραστηριότητες που δημιουργήσε ο κεντρικός κόμβος διανέμονται τοπικά. Όταν κάποιος κόμβος γίνεται χαμηλά φορτισμένος ο διαχειριστής φορτίου προσπαθεί να φέρει διάφορες διεργασίες από απομακρυσμένους κόμβους. Για να γίνει αυτό, ο κεντρικός κόμβος στέλνει τυχαία μία αίτηση με τον αριθμό των έτοιμων διεργασιών, όταν ένας απομακρυσμένος κεντρικός κόμβος λάβει το αίτημα αυτό και συγκρίνει τις έτοιμες διεργασίες του αποστολέα με τις δικές του αν οι δικές του είναι περισσότερες στέλνει κάποιες στον αποστολέα του αιτήματος. Με την διαδικασία που είδαμε παραπάνω παρατηρούμε μία πολύ καλή προσπάθεια για εξισορρόπηση φορτίου που μπορεί να εφαρμοστεί στα συστήματα p2p. Από την άλλη βέβαια όλη αυτή η αλληλεπίδραση με τους απομακρυσμένους κόμβους ίσως οδηγήσει κάποιες φορές σε καθυστερήσεις.

## 10. Συγκεντρωτικός πίνακας αλγορίθμων εξισορρόπησης φορτίου σε συστήματα “peer to peer”

Παραπάνω παρουσιάσαμε κάποιους αλγόριθμους που πετυχαίνουν εξισορρόπηση φορτίου σε συστήματα p2p τώρα θα δούμε έναν πίνακα όπου είναι αυτοί οι αλγόριθμοι συγκεντρωμένοι και ταξινομημένοι στις κατηγορίες που αναφέραμε στο κεφάλαιο 7 δηλαδή σε στατικούς και δυναμικούς.

Αλγόριθμοι	Στατικοί	Δυναμικοί
<i>Round Robin and Randomized</i>	✓	
<i>Central Manager</i>	✓	
<i>Threshold</i>	✓	
<i>Central Queue</i>		✓
<i>Local Queue</i>		✓

Στον παραπάνω πίνακα παρατηρούμε πως οι αλγόριθμοι Round robin , Randomized, Central Manager και Threshold είναι στατικοί ενώ οι Central Queue και Local Queue είναι δυναμικοί. Αυτή την διαφοροποίηση μπορούμε να την καταλάβουμε και από την λειτουργία του κάθε αλγόριθμου όπως είδαμε στην παρουσίαση του κάθε ένα στο κεφάλαιο 7.



## 11. Συμπεράσματα

Σε αυτή την εργασία παρουσιάσαμε κάποιους αλγόριθμους για εξισορρόπηση φορτίου σε συστήματα “cloud” και “peer to peer”. Οι αλγόριθμοι που είδαμε λειτουργούν ανάλογα με τα χαρακτηριστικά του κάθε συστήματος αλλά δεν είδαμε κάποιον που να λύνει όλα τα προβλήματα της εξισορρόπησης φορτίου. Κάποιου αλγόριθμοι έκαναν καλή ταξινόμηση άλλοι καλή κατανομή του φορτίου όμως δεν είναι κάποιος που τα κάνει όλα.

Τα συστήματα “cloud” τείνουν να γίνουν απαραίτητα σε όλους μας, όλοι έχουμε χρησιμοποιήσει εφαρμογές “cloud” και μπορεί να μην το ξέρουμε καν, ένα παράδειγμα τέτοιας εφαρμογής είναι το Dropbox . Σε λίγα χρόνια οι εφαρμογές cloud που χρησιμοποιούν οι απλοί χρήστες και ακόμα περισσότερο μεγάλες επιχειρήσεις θα αυξηθούν πολύ, οπότε ο όγκος των δεδομένων που θα διαχειρίζονται τα συστήματα αυτά θα αυξηθεί πολύ. Από τα παραπάνω βλέπουμε πως θα υπάρχει ανάγκη για αποδοτικότερους αλγόριθμους εξισορρόπησης φορτίου με στόχο την διαχείριση του μεγάλου όγκου δεδομένων.

Τα συστήματα peer-to-peer έχουν αναδειχθεί ως ένας δημοφιλής τρόπος για τον διαμοιρασμό τεράστιου όγκου δεδομένων. Η χρήση τους είναι συνέχεια μπροστά μας με διάσημες εφαρμογές όπως το Skype και το Nasper. Η χρήση αλγορίθμων εξισορρόπησης φορτίου βοηθάει να ξεπεράσουν τα συστήματα “peer to peer” τα προβλήματα που δημιουργούνται λόγω της δυναμικότητάς τους και να λειτουργούν αποδοτικότερα. Η χρησιμότητα των συστημάτων αυτών εξαρτάται από τεχνικές που βρίσκουν και ανακτούν τα δεδομένα. Όμως αυτές μπορεί κάποιες φορές να είναι αναποτελεσματικές οπότε και υπάρχει η ανάγκη για καινούριες (Garcia-Molina, H.).

## 12.Βιβλιογραφία

Alam Afshar, Biswas Ranjit, Naaz Sameena. (2012). *Load Balancing Algorithms for Peer to Peer and Client Server Distributed Environments*. Department of Computer Science Jamia Hamdard, New Delhi, India. International Journal of Computer Applications (0975 – 888) Volume 47– No.8.

Bharati Rajesh, Taank Chahita. (2013). *Load Balancing Algorithm for DHT Based Structured Peer to Peer System*. International Journal of Emerging Technology and Advanced Engineering. [www.ijetae.com](http://www.ijetae.com)

Chaczko Zenon, Mahadevan Venkatesh, Aslanzadeh Shahrzad, Mcdermid Christopher. (2011). *International Conference on Computer and Software Modeling*. University of Technology Sydney, Australia, Swinburne University of Technology, Australia.

Chiu Dah-Ming, Fu Tom Z. J., Wang Yongzhi. *Analysis of Load Balancing Algorithms in P2P Streaming*. Dept. of Information Engineering, the Chinese University of Hong Kong.

Doddini Probhuling L.(2013). *LOAD BALANCING ALGORITHMS IN CLOUD COMPUTING*. International Journal of Advanced Computer and Mathematical Sciences. <http://bipublication.com>.

Godfrey Brighten, Karp Richard, Lakshminarayanan Karthik, Stoica Ion, Surana Sonesh. *Load Balancing in Dynamic Structured P2P Systems*. berkeley.edu.

Gogulan R, Kalpana C, Kumar Karthick U. (2012). *Max-Min Particle Swarm Optimization Algorithm with Load Balancing for Distributed Task Scheduling on the Grid Environment*. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 1. [www.IJCSI.org](http://www.IJCSI.org)

Hirve Sumit Prof., Kshirsagar Chaitanya, Suthar Prakashchandra, Swami Vishweshwar. (2013). *Load Balancing In Structured Peer To Peer Network Using DSLS and ASH Algorithm*. International Journal of Advanced Research in Computer Science and Software Engineering. [www.ijarcsse.com](http://www.ijarcsse.com)

Kalantri Pranit, Shah Sumiran, Singhal Priyank. *Load Balancing Algorithm over a Distributed Cloud Network*. Mumbai University.

Karger David R., Ruhl Matthias. *New Algorithms for Load Balancing in Peer-to-Peer Systems*. MIT Laboratory for Computer Science Cambridge.

Katyal Mayanka, Mishra Atul. *A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment*. <http://www.publishingindia.com>

Kinger Supriya , Sidhu Amandeep Kaur.(2013). *Analysis of Load Balancing Techniques in Cloud Computing*. International Journal of Computers & Technology.

Kwon Minseok, Park Gahyun. *Distributed Tries for Load Balancing in Peer-to-Peer Systems*. Department of Computer Science SUNY Geneseo Geneseo, Department of Computer Science Rochester Institute of Technology Rochester.

Lymberopoulos Leonidas, Maglaris Vasilis, Papavassiliou Symeon. *A Novel Load Balancing Mechanism for P2P Networking*. National Technical University of Athens.

Žagar Mario, Radojević Branko. *Analysis of Issues with Load Balancing Algorithms in Hosted (Cloud) Environments*. Croatian Academic and Research Network (CARNet), Zagreb, Croatia.