ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΟΙΚΟΝΟΜΙΚΑ ΜΟΝΤΕΛΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΟΡΩΝ ΣΕ ΥΠΟΛΟΓΙΣΤΙΚΑ ΠΛΕΓΜΑΤΑ (GRID)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΕΩΡΓΙΟΣ Τ. ΚΑΚΟΥΡΑΣ

Επιβλέπων : Τσανάκας Παναγιώτης
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2006

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΟΙΚΟΝΟΜΙΚΑ ΜΟΝΤΕΛΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΟΡΩΝ ΣΕ ΥΠΟΛΟΓΙΣΤΙΚΑ ΠΛΕΓΜΑΤΑ (GRID)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΕΩΡΓΙΟΣ Τ. ΚΑΚΟΥΡΑΣ

Επιβλέπων : Τσανάκας Παναγιώτης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 13η Νοεμβρίου 2006.

.....	.....	.....
Τσανάκας Παναγιώτης	Κοζύρης Νεκτάριος	Πεκμεστζή Κιαμάλ
Καθηγητής Ε.Μ.Π.	Επίκουρος Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2006

.....

ΓΕΩΡΓΙΟΣ Τ. ΚΑΚΟΥΡΑΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Γεώργιος Τ. Κακούρας, 2006

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Η συνεχής ανάπτυξη των Υπολογιστικών Πλεγμάτων τα τελευταία χρόνια, έχει εξαπλώσει τη χρήση τους σε πολλούς τομείς τόσο επιστημονικούς όσο και εμπορικούς. Οι συμμετέχοντες σε ένα πλέγμα έχουν διαφορετικούς στόχους, στρατηγικές και απαιτήσεις, ενώ οι πόροι που μοιράζονται στο πλέγμα είναι ετερογενείς με τη διαθεσιμότητα και το φορτίο τους να ποικίλλουν δυναμικά με το χρόνο.

Σκοπός της παρούσας διπλωματικής εργασίας είναι η έρευνα για το πως οικονομικές αρχές μπορούν να χρησιμοποιηθούν για την αποτελεσματική διαχείριση των πόρων ενός πλέγματος. Αναλύεται το είδος αρχιτεκτονικής πλέγματος όπου οι προμηθευτές πόρων ανταγωνίζονται για να παρέχουν την καλύτερη υπηρεσία στους καταναλωτές των πόρων, οι οποίοι επιλέγουν πόρους βασιζόμενοι στο κόστος χρήσης αυτών και στην ποιότητα υπηρεσίας (QoS) που επιθυμούν. Παρουσιάζεται η λειτουργική δομή μιας τέτοιας οικονομίας και τα οικονομικά μοντέλα που μπορούν να εφαρμοστούν σε για τη διαχείριση των πόρων ενός υπολογιστικού πλέγματος.

Εν συνεχεία προσομοιώνεται ένα υπολογιστικό πλέγμα από γεωγραφικά διεσπαρμένους ετερογενείς πόρους, όπου ένας και περισσότεροι χρήστες συναγωνίζονται για την χρήση των διαμοιρασμένων πόρων, έχοντας κάθε φορά διαφορετικές απαιτήσεις όσον αφορά τον χρόνο στον οποίο θέλουν να ολοκληρωθεί η εκτέλεση των εργασιών τους και το ποσό που είναι διατεθειμένοι να πληρώσουν. Από τα αποτελέσματα της προσομοίωσης εξάγονται χρήσιμα συμπεράσματα για την λειτουργία του οικονομικού μοντέλου που προσομοιώνεται.

Λέξεις Κλειδιά: Υπολογιστικό πλέγμα (GRID), διαχείριση πόρων, οικονομικά μοντέλα

Abstract

The continuous development of Grid Computing during the last years, has expanded its use in many fields not only scientific but also commercial. Grid participants have different goals, policies and requirements, while the resources shared are heterogeneous with their cost and availability to vary in time.

The scope of this thesis is the research of how economic principles can be used for the effective management of Grid resources. A Grid architecture is studied where the providers of resources compete in order to provide the better service for their consumers, who buy resources based on their cost and quality of service (QoS) requirements. The functional structure of such economy and various economic models that can be applied in grid resource management are presented.

A worldwide grid is simulated, consisting of geographically dispersed heterogeneous resources, where one and more users compete for the use of shared resources, for various scenarios of budget and deadline constrains. The simulation results give useful conclusions over the simulated economic model.

Keywords: Grid, Grid computing, resource management, economic models

Στους γονείς μου.

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ τον επιβλέποντα της Διπλωματικής μου Εργασίας, Καθηγητή Ε.Μ.Π.
κ. Παναγιώτη Τσανάκα για την καθοδήγησή του και την άψογη συνεργασία μας.

Πίνακας περιεχομένων

1 Εισαγωγή 21

1.1 Αντικείμενο της διπλωματικής 21

1.2 Οργάνωση τόμου 23

2 Το GRID 25

2.1 ΓΕΝΙΚΑ ΓΙΑ ΤΟ GRID 25

2.1.1 Ορισμός 25

2.1.2 Κίνητρα Εμφάνισης των Grid 27

2.1.3 Άλλοι ορισμοί 28

2.1.4 Σύγκριση με παρόμοιες τεχνολογίες 29

2.2 ΔΥΝΑΤΟΤΗΤΕΣ ΤΟΥ GRID 30

2.2.1 Εκμετάλλευση των υποχρησιμοποιούμενων πόρων. 30

2.2.2 Παράλληλη υπολογιστική επεξεργασία 32

2.2.3 Συμμετοχή σε εικονικές οργανώσεις (VOs) και κοινή χρήση πόρων. 33

2.2.4 Πρόσβαση και σε άλλα είδη πόρων 34

2.2.5 Εξισορρόπηση των πόρων 35

2.2.6 Επιλεκτική παραχώρηση πόρων 35

2.2.7 Καλύτερη Διαχείριση 36

2.2.8 Αξιοπιστία 37

2.3 ΕΙΔΗ ΠΟΡΩΝ 39

2.3.1 Υπολογιστική ισχύς 39

2.3.2 Αποθήκευση δεδομένων 39

2.3.3 Επικοινωνία- Εύρος ζώνης 40

2.3.4 Λογισμικό και άδειες χρήσης 41

2.3.5 Ειδικός εξοπλισμός - Πολιτικές διαμοιρασμού του 41

2.4 ΕΙΔΗ GRID – ΕΜΠΟΡΙΚΑ ΠΑΡΑΔΕΙΓΜΑΤΑ 42

2.4.1 Computing grids 42

2.4.2 Data grids – Πλέγματα δεδομένων *45*

2.4.3 Collaborative grids – Συνεργατικά πλέγματα *46*

2.4.4 Govermental Grids – Κυβερνητικά πλέγματα *47*

2.4.5 Utility grids - Πλέγματα αξιοποίησης / χρήσης (όχι κτήσης) τεχνολογικής υποδομής *47*

2.4.6 Enterprise optimization grids – Grid επιχειρηματικής βελτιστοποίησης .. *48*

2.5 ΤΡΟΠΟΙ ΑΝΑΠΤΥΞΗΣ ΕΜΠΟΡΙΚΩΝ GRID 48

2.5.1 Departmental grids *49*

2.5.2 Intergrids *49*

2.5.3 ExtraGrids *49*

2.6 ΑΝΑΛΥΣΗ ΠΛΕΟΝΕΚΤΗΜΑΤΩΝ, ΕΜΠΟΡΙΚΗΣ ΧΡΗΣΗΣ GRID. ΔΕΙΚΤΕΣ ΕΚΤΙΜΗΣΗΣ ΑΥΤΩΝ 50

2.6.1 Επιχειρηματικά οφέλη *51*

2.6.2 Τεχνικά- Τεχνολογικά Πλεονεκτήματα *54*

2.7 ΠΡΟΣΠΑΘΕΙΕΣ ΠΡΟΤΥΠΟΠΟΙΗΣΗΣ GRID 55

2.7.1 Global Grid Forum (GGF) *55*

2.7.2 EGA (Enterprise Grid Alliance) *55*

2.7.3 Open Grid Forum (OGF) *56*

3 Οικονομικά Μοντέλα Διαχείρισης Πόρων σε Υπολογιστικά Πλέγματα. 57

3.1 Γιατί οι παραδοσιακές προσεγγίσεις απόδοσης πόρων αποτυγχάνουν. 58

3.2 Ανάγκη για βασισμένη σε οικονομικές αρχές διαχείριση πόρων Grid. 59

3.3 Οφέλη μιας « Grid – Οικονομίας » 62

3.4 Δομή μιας « Grid – Οικονομίας » 64

3.4.1 Χαρακτηριστικά μιας « Grid – Οικονομίας » *64*

3.4.2 Grid Layers *65*

3.4.3 Λειτουργική Δομή μιας « Grid – Οικονομίας » *67*

3.5 Οικονομικά μοντέλα διαχείρισης πόρων πλέγματος 70

3.5.1 Μοντέλο αγοράς προϊόντων (Σταθερή τιμολόγηση ή προσαρμοσμένη στην προσφορά/ζήτηση) *71*

3.5.2 Μοντέλο ανακοινωμένων τιμών (Posted Price Model) *73*

3.5.3 Μοντέλο – διαπραγματεύσεων (Bargaining Model) *74*

3.5.4 Μοντέλο αποδοχής προσφοράς / σύμβασης (Tender/Contract-Net Model) *75*

3.5.5 Auction Model - Μοντέλο δημοπρασίας *78*

3.5.6 Βασισμένο σε προσφορές μοντέλο καταμερισμού πόρων - Bid-based Proportional Resource Sharing Model *84*

3.5.7 Συνεταιριστικό πρότυπο ανταλλαγής - Cooperative Bartering Model..... *85*

3.5.8 Μονοπώλιο/ολιγοπώλιο..... *85*

3.6 Τιμολόγηση, χρεώσεις και μηχανισμοί πληρωμής σε μια Grid – Οικονομία. 86

3.6.1 Πώς καθορίζεται η τιμή ενός πόρου ? *86*

3.6.2 Στοιχεία υπηρεσιών που καταγράφονται και χρεώνονται *88*

3.6.3 Μηχανισμοί πληρωμής *89*

4 Προσομοιώσεις Διαχείρισης Πόρων σε Υπολογιστικά Πλέγματα με τη χρήση του GridSim[41] 91

4.1 Εισαγωγή 91

4.2 Άλλα εργαλεία προσομοίωσης 92

4.3 GridSim 93

4.3.1 Δυνατότητες του GridSim *93*

4.3.2 Οντότητες στο GridSim *94*

4.4 Σχεδιασμός προσομοιώσεων 96

4.4.1 Εισαγωγή..... *96*

4.4.2 Ροή εργασιών που εκτελεί ο Μεσίτης πόρων πλέγματος - Grid Resource Broker κατά τη διάρκεια της προσομοίωσης..... *98*

5 Σχεδιασμός προσομοίωσης «Οικονομίας-Grid» ακολουθώντας στρατηγική βελτιστοποίησης κόστους.....101

5.1 Χαρακτηριστικά πλέγματος που προσομοιώνεται. 102

5.1.1 Χαρακτηριστικά Πόρων..... *102*

5.1.2 Κόστος χρήσης πόρων *103*

5.1.3 Εργασίες προς εκτέλεση..... *107*

5.2 Προσομοιώσεις βάσει απαιτήσεων των χρηστών. Καθορισμένος Προϋπολογισμός και Προθεσμία 108

5.2.1 Ένας χρήστης υποβάλλει εργασίες στο πλέγμα..... 108

5.2.2 Πολλοί χρήστες υποβάλλουν ταυτόχρονα εργασίες στο πλέγμα..... 115

6 Επίλογος.....129

7 Βιβλιογραφία.....131

8 ΠΑΡΑΡΤΗΜΑ. Screenshots – Κώδικας.....134

Ευρετήριο Σχημάτων

Σχήμα 2.1 Υπολογιστικό Πλέγμα - Grid 26

Σχήμα 2.2 Εκμετάλλευση των υποχρησιμοποιούμενων πόρων 31

Σχήμα 2.3 Εκτέλεση πολλαπλών αντιγράφων μιας εργασίας στο πλέγμα 38

Σχήμα 2.4 Τρόποι ανάπτυξης εμπορικών Grid 48

Σχήμα 3.1 Αλληλεπίδραση χρηστών – μεσαζόντων - πόρων 61

Σχήμα 3.2 Διαστρωμάτωση Αρχιτεκτονικής Υπολογιστικού Πλέγματος 66

Σχήμα 3.3 «Οικονομία Grid» 69

Σχήμα 3.4 Μοντέλο αγοράς προϊόντων 72

Σχήμα 3.5 Μοντέλο ανακοινωμένων τιμών 73

Σχήμα 3.6 Μοντέλο διαπραγματεύσεων 75

Σχήμα 3.7 Μοντέλο αποδοχής προσφοράς / σύμβασης 76

Σχήμα 3.8 Μοντέλο δημοπρασίας 79

Σχήμα 5.1 Αριθμός Gridlets που εκτελούνται για διαφορετικά όρια προϋπολογισμού, με καθορισμένη κάθε φορά την προθεσμία 110

Σχήμα 5.2 Αριθμός Gridlet που εκτελούνται για διαφορετική κάθε φορά προθεσμία, με καθορισμένο κάθε φορά το όριο προϋπολογισμού. 113

Σχήμα 5.3 Αριθμός Gridlet που εκτελούνται για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν συμμετέχουν 6 καταναλωτές πόρων στο πλέγμα. 116

Σχήμα 5.4 Αριθμός Gridlet που εκτελούνται για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν συμμετέχουν 15 καταναλωτές πόρων στο πλέγμα. 118

Σχήμα 5.5 Αριθμός Gridlet που εκτελούνται για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν συμμετέχουν 23 καταναλωτές πόρων στο πλέγμα. 121

Σχήμα 5.6 Gridlets που εκτελεί ο 1ος χρήστης για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν στο πλέγμα συμμετέχουν 1, 6 ,15, 23 καταναλωτές πόρων. 123

Σχήμα 5.7 Μέσος χρόνος ολοκλήρωσης επεξεργασίας Gridlet, για διαφορετικά όρια προϋπολογισμού, όταν συμμετέχουν 1, 6, 15, 23 καταναλωτές πόρων στο πλέγμα 128

Ευρετήριο Πινάκων

Πίνακας 4.1 Στρατηγικές Βελτιστοποίησης διαχείρισης στο GridSim 97

Πίνακας 5.1 Χαρακτηριστικά των CPU που οργανώνονται ως πόροι 102

Πίνακας 5.3 Χαρακτηριστικά των πόρων που συμμετέχουν στο πλέγμα που προσομοιώνουμε (α) 105

Πίνακας 5.4 Χαρακτηριστικά των πόρων που συμμετέχουν στο πλέγμα που προσομοιώνουμε (β) 106

Πίνακας 5.6 Εργασίες που εκτελούνται στο πλέγμα. 107

Πίνακας 5.7 Προσομοιωμένες τιμές Προϋπολογισμού - Προθεσμίας 108

Πίνακας 5.8 Αριθμός Gridlets που εκτελούνται για διαφορετικά όρια προϋπολογισμού, με καθορισμένη κάθε φορά την προθεσμία 110

Πίνακας 5.9 Αριθμός Gridlet που εκτελούνται για διαφορετική κάθε φορά προθεσμία, με καθορισμένο κάθε φορά το όριο προϋπολογισμού. 113

Πίνακας 5.10 Ρυθμός αύξησης των Gridlets με με καθορισμένο κάθε φορά το όριο προϋπολογισμού 114

Πίνακας 5.11 Αριθμός Gridlet που εκτελούνται για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν συμμετέχουν 6 καταναλωτές πόρων στο πλέγμα. 116

Πίνακας 5.12 Αριθμός Gridlet που εκτελούνται για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν συμμετέχουν 15 καταναλωτές πόρων στο πλέγμα. 117

Πίνακας 5.13 Αριθμός Gridlet που εκτελούνται για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν συμμετέχουν 23 καταναλωτές πόρων στο πλέγμα. 120

Πίνακας 5.15 Χρόνος ολοκλήρωσης επεξεργασίας Gridlet για κάθε χρήστη, όταν συμμετέχουν 1 και 6 καταναλωτές πόρων στο πλέγμα, για διαφορετικά όρια προϋπολογισμού. 124

Πίνακας 5.16 Χρόνος ολοκλήρωσης επεξεργασίας Gridlet για κάθε χρήστη, όταν συμμετέχουν 15 καταναλωτές πόρων στο πλέγμα, για διαφορετικά όρια προϋπολογισμού. 125

Πίνακας 5.17 Χρόνος ολοκλήρωσης επεξεργασίας Gridlet για κάθε χρήστη, όταν συμμετέχουν 23 καταναλωτές πόρων στο πλέγμα, για διαφορετικά όρια προϋπολογισμού. 126

Πίνακας 5.18 Μέσος χρόνος ολοκλήρωσης επεξεργασίας Gridlet, για διαφορετικά όρια προϋπολογισμού, όταν συμμετέχουν 1, 6, 15, 23 καταναλωτές πόρων στο πλέγμα..... 127

1

Εισαγωγή

Στο εισαγωγικό αυτό κεφάλαιο παρουσιάζεται το αντικείμενο της παρούσας διπλωματικής εργασίας και τα κεφάλαια που θα ακολουθούν.

1.1 Αντικείμενο της διπλωματικής

Η συνεχής ανάπτυξη των Υπολογιστικών Πλεγμάτων – Grid τα τελευταία χρόνια μας επιτρέπει να θεωρήσουμε ότι αποτελούν την πιο πρόσφατη και πλήρη εξέλιξη μεταξύ παρόμοιων αναπτυσσόμενων τεχνολογιών–υλοποιήσεων όπως τα κατανεμημένα υπολογιστικά συστήματα (distributed computing), το Διαδίκτυο (WWW), τα δίκτυα ομότιμων οντοτήτων (peer-to-peer) και τις τεχνολογίες εικονικοποίησης πόρων (virtualization technologies).

Σε ένα Grid πολλά ετερογενή συστήματα που διαθέτουν πόρους όπως υπολογιστές, δίκτυα, βάσεις δεδομένων και επιστημονικά όργανα συνδέονται μεταξύ τους, με σκοπό την δημιουργία ενός μεγάλης κλίμακας και ευρέως διασκορπισμένου υπολογιστικού συστήματος. Οι χρήστες του πλέγματος μπορούν να απολαμβάνουν όλους αυτούς τους συσσωρευμένους πόρους υπό τη μορφή υπηρεσιών.

Οι πόροι που αποτελούν ένα Grid είναι ετερογενείς και η διαθεσιμότητα και το φορτίο τους ποικίλλει δυναμικά με το χρόνο. Μπορεί να υπόκεινται σε διαφορετικές οργανώσεις καθεμία με τις δικές της πολιτικές διαχείρισης και μοντέλα χρήσης. Παράλληλα οι συμμετέχοντες στο πλέγμα (ιδιοκτήτες των πόρων, χρήστες των πόρων) έχουν διαφορετικούς σκοπούς, στρατηγικές και απαιτήσεις. Όλα αυτά τα έμφυτα χαρακτηριστικά του πλέγματος κάνουν την διαχείριση των πόρων μια ιδιαίτερα δύσκολη πρόκληση.

Στην παρούσα διπλωματική εργασία εξετάζουμε πως οικονομικές αρχές μπορούν να χρησιμοποιηθούν για την αποτελεσματική διαχείριση των πόρων ενός πλέγματος.

Ερευνούμε γιατί οι παραδοσιακές προσεγγίσεις αποτυγχάνουν και γιατί η υιοθέτηση οικονομικών μοντέλων είναι αυτή που μπορεί να κάνει ένα υπολογιστικό πλέγμα ελκυστικό, για όλους τους συμμετέχοντες σε αυτό.

Περιγράφουμε το είδος αρχιτεκτονικής πλέγματος όπου οι προμηθευτές πόρων (ιδιοκτήτες) ανταγωνίζονται για να παρέχουν την καλύτερη υπηρεσία στους καταναλωτές των πόρων (χρήστες), οι οποίοι επιλέγουν πόρους βασιζόμενοι στο κόστος χρήσης αυτών και στην ποιότητα υπηρεσίας (QoS) που επιθυμούν.

Παρουσιάζουμε την λειτουργική δομή μιας τέτοιας οικονομίας, τα οικονομικά μοντέλα που μπορούν να εφαρμοστούν σε μια τέτοια αρχιτεκτονική πλέγματος καθώς και άλλες παραμέτρους που σχετίζονται με αυτά όπως: τιμολόγηση, χρέωση, μηχανισμούς πληρωμής.

Εν συνεχεία προσομοιώνουμε ένα υπολογιστικό πλέγμα από γεωγραφικά διεσπαρμένους ετερογενείς πόρους. Με τη χρήση του πακέτου εργαλείων προσομοίωσης πλέγματος GridSim προσομοιώνουμε σενάρια όπου ένας και περισσότεροι χρήστες συναγωνίζονται για την δέσμευση πόρων και την εκτέλεση των εργασιών τους, έχοντας κάθε φορά διαφορετικές απαιτήσεις όσον αφορά τον χρόνο στον οποίο θέλουν να ολοκληρωθεί η εκτέλεση των εργασιών τους και το ποσό που είναι διατεθειμένοι να πληρώσουν. Χρησιμοποιώντας μεγάλο αριθμό διαφορετικών σεναρίων απαιτήσεων βγάζουμε χρήσιμα συμπεράσματα για την λειτουργία του οικονομικού μοντέλου που προσομοιώνουμε.

1.2 Οργάνωση τόμου

Στο παρόν πρώτο κεφάλαιο, αναλύεται το αντικείμενο και ο σκοπός της διπλωματικής εργασίας και παρουσιάζονται εν συντομία τα περιεχόμενα κάθε κεφαλαίου.

Στο δεύτερο κεφάλαιο επεξηγείται η τεχνολογία των υπολογιστικών πλεγμάτων – Grid. Συγκρίνουμε το Grid με παρόμοιες τεχνολογίες, παρουσιάζουμε τις δυνατότητές του και τα είδη των πόρων που μπορούν να μετέχουν σε αυτό. Έπειτα εκθέτουμε εμπορικά παραδείγματα υπολογιστικών πλεγμάτων, τους τρόπους ανάπτυξης εμπορικών grid καθώς και τα πλεονεκτήματα τους.

Στο τρίτο κεφάλαιο στοιχειοθετούμε την ανάγκη για βασισμένη σε οικονομικές αρχές διαχείριση πόρων και περιγράφουμε την δομή και λειτουργία μιας «Grid - Οικονομίας». Αναλύουμε διάφορα οικονομικά μοντέλα διαχείρισης πόρων πλέγματος και εξετάζουμε και σχετικές με αυτά παραμέτρους όπως: τιμολόγηση, χρεώσεις και μηχανισμοούς πληρωμής.

Στο τέταρτο κεφάλαιο παρουσιάζουμε το πακέτο εργαλείων προσομοιώσεων πλέγματος GridSim, τις δυνατότητές του και τον τρόπο λειτουργίας του. Συγκρίνουμε με άλλα εργαλεία προσομοιώσεων και εξηγούμε την επιλογή του.

Στο πέμπτο κεφάλαιο συναντάμε την προσομοίωση μιας «Οικονομίας-Grid» ακολουθώντας όπου ακολουθείται στρατηγική βελτιστοποίησης κόστους για τους χρήστες του πλέγματος. Παρουσιάζονται τα στοιχεία των πόρων που αποτελούν το πλέγμα, τα χαρακτηριστικά των χρηστών που θέλουν να χρησιμοποιήσουν πόρους και οι απαιτήσεις των εργασιών που αυτοί θέλουν να εκτελέσουν. Για ένα μόνο χρήστη στο πλέγμα προσομοιώνουμε 96 διαφορετικά σενάρια απαιτήσεων προθεσμίας-προυπολογισμού και από τα αποτελέσματα βγάζουμε συμπεράσματα για την λειτουργία του οικονομικού μοντέλου που χρησιμοποιούμε. Έπειτα προσομοιώνουμε το ίδιο πλέγμα με πολλούς χρήστες να συναγωνίζονται για τη δέσμευση των πόρων.

Στο έκτο και τελευταίο κεφάλαιο παρουσιάζεται μια σύνοψη της διπλωματικής και των συμπερασμάτων τα οποία εξήχθησαν. Τέλος, πραγματοποιούνται προτάσεις για περαιτέρω έρευνα και επέκταση των όσων παρουσιάζονται στην παρούσα διπλωματική έργασία.

2

To GRID

2.1 ΓΕΝΙΚΑ ΓΙΑ ΤΟ GRID

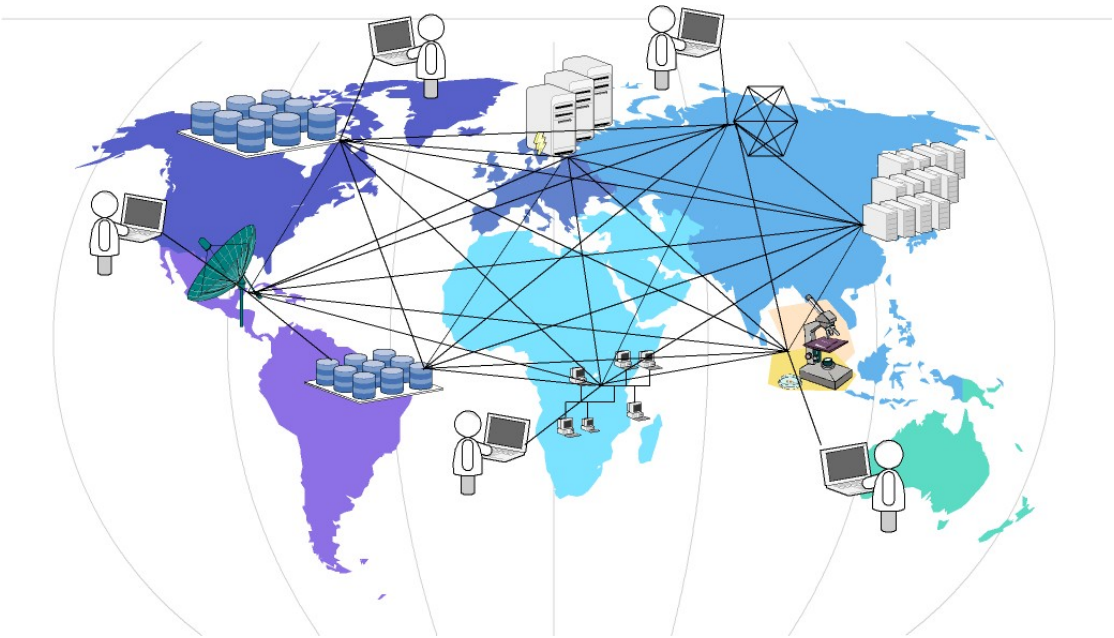
2.1.1 Ορισμός

Το Grid (Δίκτυο Κατανεμημένης Υπερ-υπολογιστικής Ισχύος) είναι μια αρχιτεκτονική διαμοιρασμού εφαρμογών/πόρων που δίνει την ικανότητα σε συνδεδεμένα ετερογενή συστήματα και εφαρμογές να μοιράζονται με διαφάνεια υπολογιστικούς και αποθηκευτικούς πόρους.

Στόχος αυτής της αρχιτεκτονικής είναι από μεγάλο αριθμό διαφορετικών συστημάτων που συνδέονται μεταξύ τους πάνω από προηγμένα δίκτυα , να δημιουργηθεί ένα απλό, εικονικό, ενοποιημένο σύστημα. Το εικονικό, ενιαίο αυτό σύστημα δίνει σε χρήστες και εφαρμογές πρόσβαση σε υπολογιστικούς πόρους, συσκευές και υπηρεσίες.

Οι πόροι που διαμοιράζονται σε μια αρχιτεκτονική Grid μπορούν να είναι ετερογενείς (να έχουν υλοποιηθεί σε διαφορετικές πλατφόρμες, αρχιτεκτονικές υλικού/λογισμικού, γλώσσες προγραμματισμού). Μπορεί να βρίσκονται σε

απομακρυσμένες γεωγραφικά τοποθεσίες και να ανήκουν σε διαφορετικές διαχειριστικές περιοχές (administrative domains). Σε σύγκριση με τον Παγκόσμιο Ιστό που είναι μια υπηρεσία για τη διανομή πληροφοριών μέσω του Διαδικτύου, θα μπορούσαμε να ισχυριστούμε ότι το Grid είναι μια υπηρεσία για τη διανομή υπολογιστικής δύναμης, αποθηκευτικών χώρων και άλλων πόρων μέσω του Διαδικτύου.



Σχήμα 2.1 Υπολογιστικό Πλέγμα - Grid

Υπάρχουν εκατοντάδες πλεγμάτων που εγκαθίστανται σε ποικίλες επιστημονικές και εμπορικές υλοποιήσεις. Πλέγματα Grid συναντάμε στις ακόλουθες βιομηχανίες:

- Αεροδιαστημική και αυτοκινητοβιομηχανία (για συνεργατικό σχεδιασμό και μοντελοποίηση).
- Αρχιτεκτονική (εφαρμοσμένη μηχανική και κατασκευές).
- Ηλεκτρονική (σχεδιασμός και δοκιμές).
- Ενέργεια (για την εξερεύνηση πετρελαίου και το φυσικού αερίου).
- Οικονομικά / ασφαλιστικά / ακίνητη περιουσία (τίτλοι και μεσιτεία — ειδικά για την ανάλυση αποθεμάτων/χαρτοφυλακίων και τη διαχείριση κινδύνου).
- Βιολογικές επιστήμες (ιδιαίτερα στα φαρμακευτικά είδη).

- Ανάλυση (επιχειρησιακές αναλύσεις, μαθηματικά και στατιστικές).
- Κατασκευές (ομαδικός συνεργατικός σχεδιασμός, διαχείριση διαδικασιών).
- Τραπεζικά και χρηματοπιστωτικά.
- Υποστήριξη αποφάσεων / ανάσυρση δεδομένων (ειδικότερα στις οικονομικές αγορές).
- Δημιουργία ψηφιακού περιεχομένου (παραδείγματος χάριν, για τη δημιουργία ειδικών εφέ).
- Δοκιμές (π.χ. δοκιμή φόρτωσης περιβάλλοντος λογισμικού).
- Εφαρμοσμένη μηχανική (αυτοματισμοί ηλεκτρονικού σχεδιασμού - EDA, αυτοματισμοί μηχανολογικού σχεδιασμού – MDA);
- Παιχνίδια (Internet, intranet).
- ΜΜΕ / ψυχαγωγία (για την παραγωγή ψηφιακού animation).

2.1.2 Κίνητρα Εμφάνισης των Grid

Την τελευταία δεκαετία η παραγωγή ταχύτερου υλικού (hardware) και πιο έξυπνου λογισμικού, οδήγησε στην μεγάλη αύξηση της παραγωγής προϊόντων σύγχρονης τεχνολογίας και στην αύξηση της ταχύτητας των δικτύων. Πιο συγκεκριμένα, στον επιστημονικό χώρο, η διαθεσιμότητα γρήγορων υπολογιστικών πόρων επέτρεψε στους επιστήμονες να διευρύνουν τις προσομοιώσεις και τα πειράματά τους και να λάβουν υπόψιν τους περισσότερες παραμέτρους από ποτέ.

Τα γρήγορα δίκτυα τους έδωσαν τη δυνατότητα να ανταλλάζουν τις μετρήσεις των οργάνων τους και τα αποτελέσματα των πειραμάτων τους, με συνεργάτες σε όλο τον κόσμο σχεδόν στιγμιαία. Ερευνητικοί οργανισμοί αρχίσαν τότε να αναπτύσσουν προγράμματα που διευκολύνουν τη δημιουργία τέτοιων συνεργασιών, για να μπορέσουν να αντιμετωπισθούν μεγάλης κλίμακας επιστημονικά προβλήματα. Συνεπεία αυτών των μεγάλων συνεργασιών και της αυξανόμενης υπολογιστικής δύναμης, ήταν τα δεδομένα που παράγονταν και αναλύονταν στα πλαίσια τέτοιων προγραμμάτων να είναι και ογκώδη και ευρέως διεσπαρμένα. Επομένως, οι προκλήσεις τέτοιων περιβαλλόντων περιστρέφονταν γύρω από τα δεδομένα –τη

διαχείριση πρόσβασης σε αυτά, τη διανομή τους, την επεξεργασία και την αποθήκευση τους-.

Αυτές οι προκλήσεις ήταν που παρακίνησαν για τη δημιουργία μιας υπολογιστικής υποδομής με την συνένωση ευρέως διανεμημένων πόρων όπως βάσεις δεδομένων, κεντρικοί υπολογιστές αποθήκευσης, υψηλής ταχύτητας δίκτυα, υπερυπολογιστές και συστάδες υπολογιστών για την επίλυση των μεγάλης κλίμακας προβλημάτων, οδηγώντας σε αυτό που είναι ευρέως γνωστό ως Grid computing.

2.1.3 Άλλοι ορισμοί

Ο όρος Grid computing χρησιμοποιήθηκε μεταφορικά στις αρχές του 1990 για περιγράψει την προσπάθεια να γίνει η υπολογιστική δύναμη πολύ εύκολα προσβάσιμη σε αντιστοιχία με το πλέγμα ηλεκτρικής ενέργειας, που παρέχει σταθερά, αξιόπιστα και με διαφανή τρόπο ηλεκτρική ενέργεια ανεξάρτητα από την πηγή αυτής. Δεδομένου ότι υπάρχει μεγάλος αριθμός προγραμμάτων σε όλο τον κόσμο που ασχολείται με την ανάπτυξη πλεγμάτων Grid για διαφορετικούς σκοπούς σε διάφορα στάδια, οι ορισμοί του πλέγματος που έχουν εμφανιστεί είναι πολλοί:

Ο σαφής ορισμός ενός υπολογιστικού πλέγματος - Grid δίνεται από τον Ian Foster στο άρθρο του "What is the Grid? A Three Point Checklist" [1]. Τα τρία σημεία σύμφωνα με τον Foster που χαρακτηρίζουν ένα υπολογιστικό πλέγμα είναι:

- Οι υπολογιστικοί πόροι δεν διαχειρίζονται κεντρικά.
- Χρησιμοποιούνται ανοιχτά πρότυπα.
- Επιτυγχάνεται σημαντική ποιότητα υπηρεσιών.

Οι Plaszczak/Wellner ορίζουν το Grid ως " τεχνολογία που επιτρέπει την εικονική συνάθροιση πόρων, την κατ'απαίτηση διάθεση πόρων και τον διαμοιρασμό υπηρεσιών (πόρων) μεταξύ οργανισμών." [2]

Σύμφωνα με την IBM, « Grid είναι η δυνατότητα, χρησιμοποιώντας ένα σύνολο ανοικτών προτύπων και πρωτοκόλλων, να αποκτήσει κανείς πρόσβαση σε εφαρμογές και δεδομένα, επεξεργαστική ισχύ, αποθηκευτική ικανότητα και μια απέραντη σειρά

άλλων υπολογιστικών πόρων μέσω του Διαδικτύου. Το Grid είναι ένα είδος παράλληλου και κατανεμημένου συστήματος που κάνει δυνατή την συλλογή, συνάθροιση και διαμοιρασμό πόρων (οι οποίοι βρίσκονται διεσπαρμένοι σε πολλαπλές διαχειριστικές περιοχές) με βάση την διαθεσιμότητα, την χωρητικότητα, την απόδοση, και το κόστος τους καθώς και με βάση τις απαιτήσεις ποιότητας υπηρεσιών των χρηστών. [3]

Μια ιδέα πολύ παλαιότερη εμφανίστηκε στο MIT το 1965 από τον Fernando Corbató. Ο Fernando και οι άλλοι σχεδιαστές του λειτουργικού Multics οραματίστηκαν μια εγκατάσταση υπολογιστών που θα λειτουργούσαν "όπως μια επιχείρηση ηλεκτρικής ενέργειας ή μια επιχείρηση ύδρευσης". [4]

Ο Rajkumar Buyya ορίζει το Grid σαν " ένα τύπος παράλληλου και κατανεμημένου συστήματος που κάνει δυνατή την εν ώρα εκτέλεσης συλλογή, συνάθροιση και διαμοιρασμό πόρων από γεωγραφικά απομακρυσμένους αυτόνομους πόρους με βάση την διαθεσιμότητα, την χωρητικότητα, την απόδοση, και το κόστος τους καθώς και με βάση τις απαιτήσεις ποιότητας υπηρεσιών των χρηστών. [5]

Το CERN, ένας από τους μεγαλύτερους οργανισμούς που χρησιμοποιεί τεχνολογίες Grid, χαρακτηρίζει το Grid ως : "μια υπηρεσία για το διαμοιρασμό υπολογιστικής ισχύς και αποθηκευτική ικανότητας διαμέσου του Διαδικτύου." [6]

Το πρόγραμμα Globus καθορίζει το πλέγμα ως «μια υποδομή που κάνει δυνατή την ενσωματωμένη, συνεργατική χρήση υπολογιστών, δικτύων, βάσεων δεδομένων, και επιστημονικών οργάνων που ανήκουν και διοικούνται από πολλούς διαφορετικούς οργανισμούς. [7]

2.1.4 Σύγκριση με παρόμοιες τεχνολογίες

Το Grid (Δίκτυο Κατανεμημένης Υπερ-υπολογιστικής Ισχύος) μπορεί να θεωρηθεί ως η πιο πρόσφατη και πλήρης εξέλιξη μεταξύ παρόμοιων αναπτυσσόμενων τεχνολογιών–υλοποιήσεων όπως τα κατανεμημένα υπολογιστικά συστήματα (distributed computing), το Διαδίκτυο (WWW), τα δίκτυα ομότιμων οντοτήτων (peer-to-peer) και οι τεχνολογίες εικονικοποίησης πόρων (virtualization technologies).

Σε σύγκριση με το Διαδίκτυο, έτσι και στο grid η πολυπλοκότητα της δομής του παραμένει κρυμμένη από το χρήστη ο οποίος απολαμβάνουν μια απλοποιημένη και διαφανή υπηρεσία. Ενώ όμως το Διαδίκτυο καθιστά ικανή μόνο την επικοινωνία μεταξύ χρηστών του και την πρόσβαση σε πληροφορία, στο grid είναι δυνατή επιπλέον η πλήρης συνεργασία και ο διαμοιρασμός πόρων με σκοπό την επίτευξη κοινών στόχων.

Όπως τα δίκτυα peer-to-peer, έτσι και το grid computing επιτρέπει στους χρήστες πρόσβαση και κοινή χρήση αρχείων. Αυτό όμως που διαχωρίζει ένα πλέγμα grid είναι ότι δεν περιορίζεται σε αρχεία αλλά επιτρέπει την κοινή πρόσβαση και χρήση και άλλων πόρων, μεταξύ πολλών χρηστών ταυτόχρονα (many-to-many sharing).

Όμοια με τις συστάδες υπολογιστών (clusters) και τις τεχνολογίες κατανεμημένων υπολογιστικών συστημάτων (distributed computing), το πλέγμα συναθροίζει υπολογιστικούς πόρους. Διαφέρει όμως με την τεχνολογία Clusters στο ότι μπορεί να συναθροίσει και πόρους οι οποίοι είναι διεσπαρμένοι τόσο διοικητικά όσο και γεωγραφικά.

Σε σύγκριση με τις τεχνολογίες εικονικοποίησης (virtualization technologies) οι οποίες επιτρέπουν την εικονικοποίηση σε ένα και μόνο σύστημα, το πλέγμα μπορεί να επεκταθεί στην εικονικοποίηση πλειάδας ετερογενών, διεσπαρμένων συστημάτων.

2.2 ΔΥΝΑΤΟΤΗΤΕΣ ΤΟΥ GRID

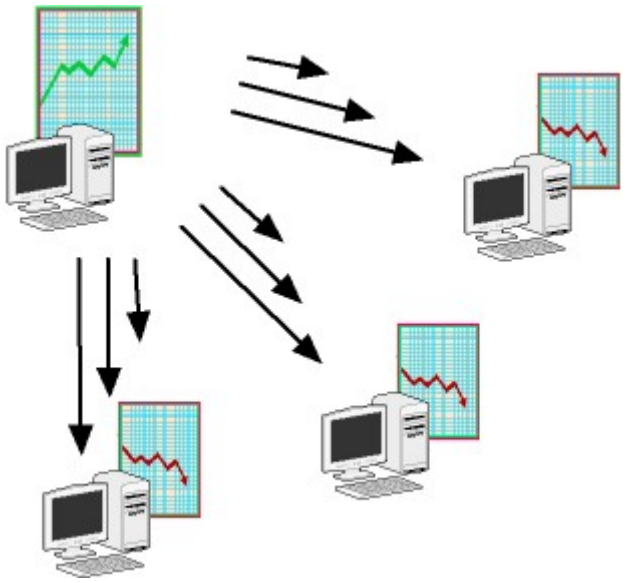
2.2.1 Εκμετάλλευση των υποχρησιμοποιούμενων πόρων.

Η συνηθέστερη χρήση του grid computing είναι η εκτέλεση μια υπάρχουσας εφαρμογής σε ένα απομακρυσμένο μηχάνημα. Το μηχάνημα στο οποίο εκτελείται η εφαρμογή μπορεί να είναι απασχολημένο αν βρεθεί σε ώρα αιχμής των δραστηριοτήτων του. Η εν λόγω εργασία αν πληροί τις εξής δύο προϋποθέσεις:

- να είναι ικανή να εκτελεστεί σε απομακρυσμένο μηχάνημα

- το απομακρυσμένο μηχάνημα να καλύπτει οποιοδήποτε εξειδικευμένο υλικό, λογισμικό ή πόρους που απαιτούνται, όπως αυτά επιβάλλονται από την προς εκτέλεση εφαρμογή

τότε μπορεί να εκτελεσθεί σε ένα υποαπασχολούμενο μηχάνημα που ανήκει στο πλέγμα.



Σχήμα 2.2 Εκμετάλλευση των υποχρησιμοποιούμενων πόρων

Ας φανταστούμε μια αερογραμμή με 90% του στόλου της στο έδαφος, μια αυτοκινητοβιομηχανία με το 40% των εγκαταστάσεων συναρμολόγησης ανενεργό, μια αλυσίδα ξενοδοχείων με το 95% των δωματίων της μη κατειλημμένα.

Έτσι και σχεδόν κάθε οργανισμός έχει στην κατοχή του τεράστια ποσά υποχρησιμοποιούμενων υπολογιστικών πόρων που είναι ευρέως διεσπαρμένα. Οι περισσότεροι υπολογιστές γραφείου είναι απασχολημένοι λιγότερο από 5% του χρόνου λειτουργίας τους ενώ ακόμα και μηχανήματα εξυπηρετητές (servers) μπορεί συχνά να είναι σχετικά μη απασχολημένοι. Με το Grid παρέχεται ένα πλαίσιο για την εκμετάλλευση όλων αυτών των υποχρησιμοποιούμενων πόρων με αποτέλεσμα να αυξάνεται αισθητά η αποδοτικότητα χρησιμοποίησης αυτών.

Εκτός από τους πόρους επεξεργασίας που υποχρησιμοποιούνται, συχνά οι υπολογιστές έχουν τεράστιο αποθηκευτικό χώρο που μένει ανεκμετάλλευτος. Το Grid μπορεί να χρησιμοποιηθεί για να αθροίσει αυτόν τον αχρησιμοποίητο αποθηκευτικό

χώρο σε ένα πολύ μεγαλύτερο εικονικό χώρο αποθήκευσης δεδομένων, που μπορεί να παρέχει βελτιωμένη απόδοση και αξιοπιστία.

Σε περιπτώσεις που μια δέσμη εργασιών πρέπει να διαβάσει μεγάλο όγκο δεδομένων, αυτά τα δεδομένα μπορούν να αντιγραφούν αυτόματα σε διάφορα στρατηγικά σημεία μέσα στο πλέγμα. Κατά συνέπεια, εάν η εργασία πρέπει να εκτελεσθεί σε ένα απομακρυσμένο μηχάνημα μέσα στο πλέγμα, τα δεδομένα είναι ήδη εκεί και δεν χρειάζεται να μεταφερθούν προς το σημείο εκείνο. Έτσι επιτυγχάνεται σαφές πλεονέκτημα απόδοσης. Επιπλέον, τα αντίγραφα των δεδομένων σε διάφορα σημεία του πλέγματος μπορούν να χρησιμοποιηθούν και ως εφεδρικά όταν τα αρχικά αντίγραφα καταστρέφονται ή δεν είναι διαθέσιμα.

2.2.2 Παράλληλη υπολογιστική επεξεργασία

Η δυνατότητα για παράλληλη επεξεργασία (parallel cpu capacity) είναι ένα από τα ελκυστικότερα χαρακτηριστικά γνωρίσματα ενός Grid. Εκτός από τη χρήση σε καθαρές επιστημονικές ανάγκες, τέτοια συσσωρευμένη υπολογιστική δύναμη δημιουργεί μια νέα εξέλιξη σε βιομηχανίες όπως ο βιοϊατρικός τομέας, η οικονομική μοντελοποίηση, η εξερεύνηση πετρελαίου, η επεξεργασία κινουμένων εικόνων, και πολλές άλλες.

Κοινό χαρακτηριστικό τέτοιων εφαρμογών είναι ότι οι εφαρμογές έχουν γραφτεί χρησιμοποιώντας αλγορίθμους που μπορούν να χωρίσουν την εργασία σε κομμάτια που μπορούν να εκτελεσθούν ανεξάρτητα.

Μια απαιτητική σε πόρους εφαρμογή μπορεί να θεωρηθεί ότι αποτελείται από «υποεργασίες» (“subjobs”), κάθε μια από τις οποίες εκτελείται σε ένα διαφορετικό μηχάνημα στο πλέγμα. Όσο λιγότερο εξαρτημένες είναι αυτές οι υποεργασίες μεταξύ τους, τόσο πιο «κλιμακωτή» μπορεί να γίνει η εφαρμογή.

Συχνά όμως εμφανίζονται εμπόδια όπως:

- ο αλγόριθμος να μπορεί να χωρίσει την εφαρμογή μόνο σε έναν περιορισμένο-συγκεκριμένο αριθμό ανεξάρτητων υποεργασιών, ανεξάρτητα από τον μέγιστο αριθμό πόρων στο πλέγμα.

- εάν οι υποεργασίες δεν είναι απολύτως ανεξάρτητες μεταξύ τους, μπορεί να προκληθούν συγκρούσεις και να περιοριστεί ο βαθμός παράλληλης επεξεργασίας.
- Παραδείγματος χάριν, εάν όλες οι υποεργασίες πρέπει να διαβάσουν και να γράψουν από το ένα κοινό αρχείο ή βάση δεδομένων, τα όρια πρόσβασης του αρχείου ή της βάσης γίνονται ο περιοριστικός παράγοντας στο βαθμό παράλληλης επεξεργασίας που μπορεί να επιτευχθεί.
- Άλλες πηγές ανταγωνισμού διεργασιών μέσα σε μια παράλληλη εφαρμογή πλέγματος είναι οι καθυστερήσεις των μηνυμάτων επικοινωνίας μεταξύ των διεργασιών, η χωρητικότητα του δικτύου, τα πρωτόκολλα συγχρονισμού, το εύρος ζώνης εισόδου-εξόδου οργάνων και συσκευών αποθήκευσης, και οι λανθάνουσες καταστάσεις που παρεμβαίνουν στις απαιτήσεις πραγματικού χρόνου.

2.2.3 Συμμετοχή σε εικονικές οργανώσεις (VOs) και κοινή χρήση πόρων.

Μια άλλη σημαντική συμβολή του Grid είναι να επιτρέπει και να απλοποιεί τη συνεργασία μεταξύ ενός ευρύτερου ακροατηρίου. Στο παρελθόν, τα κατανεμημένα υπολογιστικά συστήματα (distributed computing) πέτυχαν ως ένα ορισμένο βαθμό αυτήν την συνεργασία. Το Grid επεκτείνει αυτές τις ικανότητες σε ένα ακόμα ευρύτερο ακροατήριο, προσφέροντας πρότυπα που επιτρέπουν σε ετερογενή συστήματα να λειτουργήσουν μαζί, για να διαμορφώσουν την εικόνα ενός μεγάλου εικονικού υπολογιστικού συστήματος που διαθέτει πλειάδα πόρων.

Οι χρήστες του πλέγματος μπορούν να οργανωθούν δυναμικά σε διάφορες εικονικές οργανώσεις (virtual organizations, VOs), κάθε μια με διαφορετική πολιτική λειτουργίας. Αυτές οι εικονικές οργανώσεις μπορούν να μοιραστούν τους πόρους τους συλλογικά ως ένα ακόμα μεγαλύτερο πλέγμα, υπερσύνολο των προηγουμένων.

Η κοινή χρήση αφορά αρχικά δεδομένα υπό μορφή αρχείων ή βάσεων δεδομένων. Ένα "πλέγμα δεδομένων" μπορεί να επεκτείνει τις ικανότητες δεδομένων με διάφορους τρόπους. Καταρχάς, τα αρχεία ή οι βάσεις δεδομένων μπορούν να

εκταθούν σε πολλά συστήματα και να προσφέρουν νέες δυνατότητες σε σύγκριση με οποιοδήποτε απλό σύστημα. Μια τέτοια έκταση των δεδομένων σε διάφορα σημεία του πλέγματος μπορεί να βελτιώσει τους χρόνους μεταφοράς δεδομένων μέσω χρήσης τεχνικών απογύμνωσης (striping techniques). Τα δεδομένα μπορούν να αντιγραφούν σε διάφορα σημεία σε όλο το πλέγμα και να χρησιμεύσουν και ως αντίγραφα ασφαλείας.

Η κοινή χρήση δεν περιορίζεται μόνο σε δεδομένα, αλλά περιλαμβάνει πολλούς άλλους πόρους, όπως εξοπλισμός, λογισμικό, υπηρεσίες, άδειες χρήσης, και άλλα. Αυτοί οι πόροι εικονικοποιούνται "virtualize" για να δώσουν μια πιο ομοιόμορφη διαλειτουργικότητα μεταξύ των ετερογενών «μελών» του πλέγματος.

Οι συμμετέχοντες και οι χρήστες του πλέγματος μπορούν να είναι μέλη διάφορων πραγματικών και εικονικών οργανώσεων. Το πλέγμα μπορεί να βοηθήσει στην επιβολή κανόνων ασφάλειας μεταξύ τους και να εφαρμόσει τις πολιτικές, οι οποίες μπορούν να επιλύσουν τις προτεραιότητες τόσο για τους πόρους όσο και για τους χρήστες.

2.2.4 Πρόσβαση και σε άλλα είδη πόρων

Εκτός από τους υπολογιστικούς και αποθηκευτικούς πόρους, ένα πλέγμα μπορεί να παρέχει πρόσβαση σε άλλα είδη πόρων, ειδικό εξοπλισμό, λογισμικό, άδειες χρήσης και άλλες υπηρεσίες.

Έτσι αν ένας χρήστης επιθυμεί να αυξήσει το συνολικό εύρος ζώνης του στο Διαδίκτυο για να υλοποιήσει π.χ. μια μηχανή αναζήτησης για εξόρυξη δεδομένων, η εργασία μπορεί να χωριστεί μεταξύ των μηχανημάτων του πλέγματος που καθένα έχει ανεξάρτητη σύνδεση στο Διαδίκτυο. Μ’αυτό τον τρόπο η συνολική ικανότητα αναζήτησης πολλαπλασιάζεται, δεδομένου ότι κάθε μηχανή έχει μια χωριστή σύνδεση στο Διαδίκτυο. Εάν οι μηχανές είχαν μοιραστεί τη σύνδεση στο Διαδίκτυο, δεν θα υπήρχε αποτελεσματική αύξηση του εύρους ζώνης.

Σε ορισμένα μηχανήματα ενός Grid μπορεί να έχει εγκατασταθεί μεγάλης αξίας λογισμικό για τις ανάγκες των χρηστών. Οι εργασίες του μπορούν να σταλούν σε αυτά τα μηχανήματα εκμεταλλευόμενες πλήρως τις άδειες χρήσης λογισμικού.

Η χρήση απομακρυσμένων εκτυπωτών με προηγμένες ικανότητες χρώματος ή πολύ γρηγορότερες ταχύτητες εκτύπωσης, μέσα σε ένα τοπικό δίκτυο είναι πλέον άκρως τετριμμένη. Ένα Grid επεκτείνει την ικανότητα αυτή και σε απομακρυσμένη χρήση άλλου ειδικού εξοπλισμού π.χ. έναν υψηλής ταχύτητας, αυτοτροφοδοτούμενο εγγραφέα DVD που θα μπορούσε να χρησιμοποιηθεί για την ταχύτερη εγγραφή μεγάλων ποσοτήτων δεδομένων. Με όμοιο τρόπο ηλεκτρονικά μικροσκόπια ανίχνευσης συνδεδεμένα σε κάποια από τα μηχανήματα του πλέγματος μπορούν να χρησιμοποιούνται από απόσταση.

Μια αρχιτεκτονική Grid μπορεί να επιτρέψει ακόμα πιο περίπλοκη πρόσβαση, ενδεχομένως σε απομακρυσμένα ιατρικά διαγνωστικά και ρομποτικά εργαλεία χειρουργικών επεμβάσεων, με αμφίδρομη-επικοινωνία από απόσταση.

2.2.5 Εξισορρόπηση των πόρων

Ένα Grid συνενώνει μεγάλο αριθμό πόρων από μεμονωμένα μηχανήματα για τη δημιουργία ενός μεγαλύτερου συνολικού εικονικού πόρου. Οι εφαρμογές που είναι ειδικά σχεδιασμένες να λειτουργούν σε περιβάλλον πλέγματος, μπορεί να επιτύχουν εξισορρόπηση των πόρων στέλνοντας τις υποεργασίες για εκτέλεση σε μηχανές με χαμηλή χρησιμοποίηση.

Το χαρακτηριστικό αυτό γνώρισμα του πλέγματος είναι άκρως πολύτιμο σε καταστάσεις μέγιστης δραστηριότητας, μέσα στα τμήματα ενός μεγαλύτερου οργανισμού. Σε μια τέτοια κατάστασης πλήρους φόρτου μπορούν: να καθοδηγηθούν οι εργασίες σε σχετικά μη απασχολημένα μηχανήματα του πλέγματος ή σε περιπτώσεις που όλα τα μηχανήματα του πλέγματος είναι ήδη πλήρως απασχολημένα, οι χαμηλότερης προτεραιότητας εργασίες που εκτελούνται να ανασταλούν προσωρινά ή να ακυρωθούν ώστε να ελευθερωθούν πόροι για την εκτέλεση εργασιών υψηλότερης προτεραιότητας.

2.2.6 Επιλεκτική παραχώρηση πόρων

Στον κόσμο των επιχειρήσεων το χαρακτηριστικό αυτό του Grid είναι άκρως πολύτιμο. Συχνά σε επιχειρήσεις Κατά την υλοποίηση ενός μεγάλου έργου (project)

συμβαίνει διοικητικές αποφάσεις να αλλάζουν την προτεραιότητα αυτού και να πρέπει να ολοκληρωθεί μέσα σε μικρότερο χρονικό διάστημα.

Π.χ. αποφασίζεται ότι η παραγωγή ενός προϊόντος πρέπει να επιτευχθεί πολύ πιο γρήγορα για να βγει στην αγορά πριν τους ανταγωνιστές του (fastest time to market). Μια αρχιτεκτονική πλέγματος μπορεί υπό προϋποθέσεις να ικανοποιήσει μια τέτοια κατάσταση. Θα πρέπει το μέγεθος των εργασιών να είναι εκ των προτέρων γνωστό και αυτές να μπορούν να διαχωριστούν σε μικρότερες υποεργασίες. Το πλέγμα με σωστό προγραμματισμό μπορεί να διαθέσει μεγάλο αριθμό πόρων στο συγκεκριμένο έργο, δίνοντας του προτεραιότητα σε σχέση με άλλες εργασίες, έτσι ώστε να επιτευχθεί ο μικρότερος χρόνος ολοκλήρωσής του.

Επιλεκτική παραχώρηση των πόρων θα μπορούσε να γίνει σε ένα Grid και με άλλα κριτήρια όπως π.χ. μειώνοντας τις συγκρούσεις και ελαχιστοποιώντας τον επικοινωνιακό φόρτο μέσα στο πλέγμα, όταν οι εργασίες επικοινωνούν η μια με την άλλη, με το Διαδίκτυο ή με αποθηκευτικούς πόρους.

2.2.7 Καλύτερη Διαχείριση

Η ενοποίηση των πόρων σε ένα υπολογιστικό πλέγμα και ο χειρισμός ετερογενών συστημάτων με ενιαίο τρόπο δημιουργεί νέες δυνατότητες για την καλύτερη διαχείριση μεγαλύτερων και πιο διεσπαρμένων υποδομών. Είναι εύκολο να απεικονιστεί η χωρητικότητα και η χρησιμοποίηση, κάτι που διευκολύνει τον έλεγχο των δαπανών για υπολογιστικούς πόρους στα πλαίσια ενός μεγαλύτερου οργανισμού.

Το Grid προσφέρει δυνατότητα διαχείρισης των προτεραιοτήτων μεταξύ διαφορετικών έργων (projects) ενώ στο παρελθόν κάθε έργο έπρεπε να διαθέτει τους δικούς πόρους καθώς και τα έξοδα που εμπλέκονται μ'αυτό.

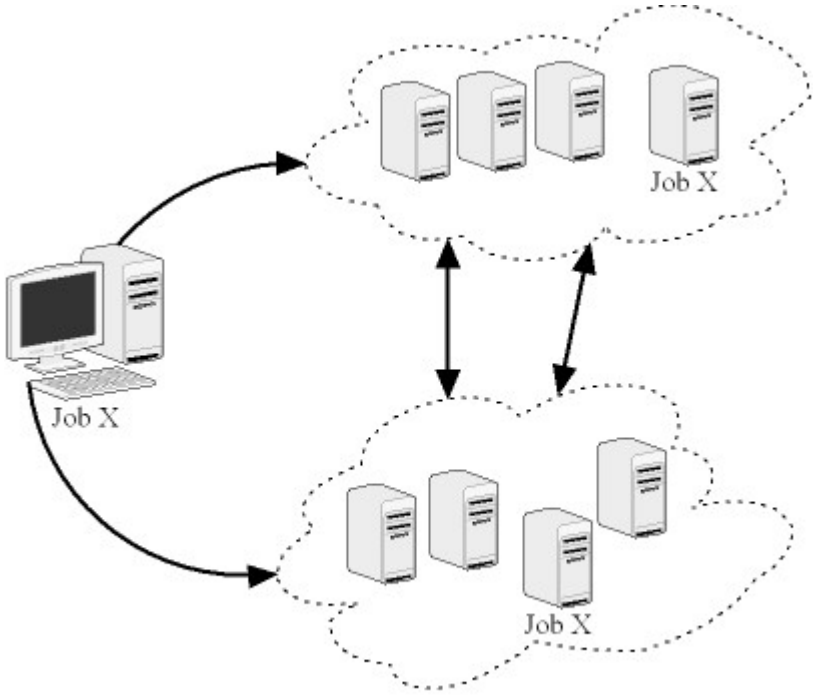
Συχνά οι πόροι που διατίθενται σε ένα έργο μπορεί να υποχρησιμοποιούνται την ώρα που κάποιο άλλο χρειάζεται περισσότερους πόρους λόγω απροσδόκητων καταστάσεων. Έχοντας λοιπόν τη συνολική εικόνα που μπορεί να προσφέρει ένα Grid, είναι ευκολότερος ο έλεγχος και η διαχείριση τέτοιων καταστάσεων .

2.2.8 Αξιοπιστία

Για να αυξήσουν την αξιοπιστία τους τα συμβατικά υπολογιστικά συστήματα χρησιμοποιούν ακριβό εξειδικευμένο υλικό. Δομούνται χρησιμοποιώντας μικροτσίπ με εφεδρικά κυκλώματα για να μπορούν να επιβεβαιώνουν τα αποτελέσματα και να εξασφαλίζουν ομαλή ανάκτηση μετά από αστοχία υλικού ή πιθανή βλάβη. Επίσης τα μηχανήματα αυτά χρησιμοποιούν ζεύγη επεξεργαστών με δυνατότητα εν λειτουργία αντικατάστασης ώστε σε περίπτωση βλάβης του ενός να μπορεί να αντικατασταθεί χωρίς την παύση λειτουργίας του άλλου.

Οι παροχές ηλεκτρικού ρεύματος και τα συστήματα ψύξης εγκαθίστανται με παρόμοιο τρόπο ώστε σε περίπτωση διακοπής παροχής ενέργειας να εκκινούν άμεσα γεννήτριες και να μην διακόπτεται η λειτουργία. Όλα τα παραπάνω μπορεί να οδηγούν σε ένα αξιόπιστο σύστημα, όμως το κόστος υλοποίησης του είναι πολύ μεγάλο.

Μια εναλλακτική προσέγγιση η οποία κερδίζει όλο και περισσότερο έδαφος, είναι αυτή που βασίζεται περισσότερο σε τεχνολογίες λογισμικού παρά σε ακριβό υλικό. Το Grid είναι μόνο το ξεκίνημα μιας τέτοιας τεχνολογίας. Τα συστήματα που ανήκουν σε ένα πλέγμα μπορούν να είναι σχετικά φθηνά και να είναι γεωγραφικά διασκορπισμένα. Κατά συνέπεια, εάν υπάρξει διακοπή ενέργειας ή κάποιο άλλο είδος αστοχίας σε μια θέση, τα άλλα μέρη του πλέγματος είναι σχεδόν απίθανο να επηρεαστούν. Το λογισμικό διαχείρισης του πλέγματος τότε μπορεί αυτόματα να υποβάλει εκ νέου τις εργασίες σε άλλα μηχανήματα στο πλέγμα όταν ανιχνευθεί μια αποτυχία.



Σχήμα 2.3 Εκτέλεση πολλαπλών αντιγράφων μιας εργασίας στο πλέγμα

Σε κρίσιμες, καταστάσεις πραγματικού χρόνου, πολλαπλά αντίγραφα σημαντικών εργασιών μπορούν να εκτελούνται σε διαφορετικές μηχανές σε όλο το πλέγμα. Έτσι αν υπάρξει μια αστοχία τα αποτελέσματα της εργασίας μπορούν να ληφθούν από ένα άλλο μηχάνημα του πλέγματος.

Με την περαιτέρω ανάπτυξη του αποκαλούμενου “autonomic computing”, που αυτόματα λύνει προβλήματα που εμφανίζονται στο πλέγμα, ακόμα και πριν γίνουν αυτά αντιληπτά από ένα χρήστη ή διαχειριστή του συστήματος, στο μέλλον θα μπορούμε να εξασφαλίσουμε αξιοπιστία με τη χρήση μόνο λογισμικού κάτω από ένα περιβάλλον Grid.

2.3 ΕΙΔΗ ΠΟΡΩΝ

Τα μηχανήματα που ανήκουν σε ένα Grid συχνά περιγράφονται με όρους όπως "κόμβοι", "πόροι", "μέλη", "δότες", "πελάτες," "οικοδεσπότες," "μηχανές". Όλοι συμβάλλουν με οποιοδήποτε συνδυασμό πόρων στο πλέγμα σαν ένα σύνολο. Μερικοί από τους πόρους μπορούν να χρησιμοποιηθούν από όλους τους χρήστες του πλέγματος ενώ για άλλους υπάρχουν περιορισμοί.

2.3.1 Υπολογιστική ισχύς

Η υπολογιστική ισχύς που παρέχεται από τους επεξεργαστές των μηχανημάτων του πλέγματος είναι ο πιο συνηθισμένος πόρος. Οι επεξεργαστές μπορούν να ποικίλλουν σε ταχύτητα, αρχιτεκτονική, πλατφόρμα λογισμικού και σε άλλους σχετικούς παράγοντες όπως μνήμη, αποθήκευση, συνδεσιμότητα. Υπάρχουν τρεις κύριοι τρόποι για την εκμετάλλευση των υπολογιστικών πόρων σε ένα Grid. Ο απλούστερος είναι η εκτέλεση μιας εφαρμογής σε ένα απομακρυσμένο μηχάνημα του πλέγματος αντί να εκτελεσθεί τοπικά. Ο δεύτερος είναι να χωριστεί η εργασία (εφόσον αυτό είναι εφικτό) σε μικρότερες υποεργασίες και αυτές να εκτελεσθούν παράλληλα σε διαφορετικούς επεξεργαστές. Τέλος είναι να τρέχει μια εργασία πολλές φορές σε διαφορετικά μηχανήματα στο πλέγμα, όταν αυτό είναι απαραίτητο.

2.3.2 Αποθήκευση δεδομένων

Ο δεύτερος πιο κοινός πόρος που χρησιμοποιείται σε ένα πλέγμα είναι ο αποθηκευτικός χώρος. Ένα πλέγμα που παρέχει την δυνατότητα ενοποίησης των αποθηκευτικών πόρων αποκαλείται "πλέγμα δεδομένων" (data grid).

Κάθε μηχανή στο πλέγμα παρέχει συνήθως μέρος της αποθηκευτικής της ικανότητας για χρήση στο πλέγμα, ακόμα κι αν αυτό είναι προσωρινό. Αυτό μπορεί να αφορά μνήμη που συνδέεται με τον επεξεργαστή ή μπορεί να είναι δευτερεύον αποθηκευτικός χώρος όπως σκληροί δίσκοι ή άλλα μέσα μόνιμης αποθήκευσης. Τα

δευτερεύοντα μέσα αποθήκευσης σε ένα πλέγμα μπορούν να χρησιμοποιηθούν με διάφορους τρόπους ώστε να αυξηθεί η συνολική χωρητικότητα, απόδοση, διανομή, και αξιοπιστία των δεδομένων.

Η συνολική χωρητικότητα μπορεί να αυξηθεί με την εκμετάλλευση των διαθέσιμων χώρων πολλών μηχανημάτων, ενοποιώντας τους κάτω από ένα κοινό σύστημα αρχείων. Ένα μοναδικό όνομα στα όρια του πλέγματος μπορεί να αποδοθεί σε κάθε στοιχείο ώστε οι χρήστες να μπορούν να χρησιμοποιήσουν π.χ. ένα αρχείο αδιαφορώντας για το που αυτό βρίσκεται αποθηκευμένο στο πλέγμα. Με παρόμοιο τρόπο διάφορες βάσεις δεδομένων μπορούν να ενοποιηθούν και να σχηματίσουν μια μεγαλύτερη και περιεκτικότερη βάση προσβάσιμη από τους χρήστες μέσω λειτουργιών ερωτημάτων.

Με τη δημιουργία πολλαπλών αντιγράφων συνόλων δεδομένων μπορεί να επιτευχθεί αυξημένη αξιοπιστία και απόδοση. Ένας ευφυής χρονοπρογραμματιστής πλέγματος μπορεί να βοηθήσει να επιλέξει σε ποιες συσκευές αποθήκευσης πρέπει να υπάρχουν αντίγραφα των δεδομένων, με βάση το πόσο συχνά αυτά χρησιμοποιούνται και από ποιους. Οι εργασίες έπειτα μπορούν να προγραμματιστούν να τρέξουν στις μηχανές που βρίσκονται «πιο κοντά» ή είναι άμεσα συνδεδεμένες με αυτές τις συσκευές αποθήκευσης.

2.3.3 Επικοινωνία- Εύρος ζώνης

Άλλος ένας σημαντικός πόρος πλέγματος είναι ή ικανότητα μετάδοσης δεδομένων. Αυτό περιλαμβάνει επικοινωνία μέσα στα όρια ενός πλέγματος ή και έξω από αυτό. Οι εσωτερικές επικοινωνίες είναι σημαντικές για την αποστολή εργασιών και των απαιτούμενων δεδομένων κόμβους του πλέγματος. Μερικές εργασίες απαιτούν την επεξεργασία μεγάλης ποσότητας δεδομένων που μπορεί να βρίσκονται σε απομακρυσμένα μηχανήματα. Το εύρος ζώνης που είναι διαθέσιμο για τέτοιες επικοινωνίες είναι κρίσιμος πόρος και μπορεί να περιορίσει τη χρησιμοποίηση του πλέγματος.

Η εξωτερική επικοινωνία με το Διαδίκτυο είναι σημαντική σε πολλές περιπτώσεις όπως π.χ. στην υλοποίηση μηχανών αναζήτησης. Τα μηχανήματα του πλέγματος μπορούν να έχουν το καθένα τη δική του σύνδεση στο Διαδίκτυο εκτός από την

μεταξύ τους σύνδεση στα όρια του πλέγματος. Όταν οι συνδέσεις αυτές δεν μοιράζονται το ίδιο επικοινωνιακό μονοπάτι, τότε προσθέτουν στο συνολικό διαθέσιμο εύρος ζώνης για πρόσβαση στο Διαδίκτυο.

Επιπλέον (εφεδρικοί) επικοινωνιακοί δίαυλοι είναι πολλές φορές απαραίτητοι για την αντιμετώπιση πιθανών βλαβών στο δίκτυο ή αυξημένη επικοινωνιακή κίνηση δεδομένων. Ένα σύστημα διαχείρισης Grid είναι ικανό να απεικονίσει την τοπολογία του δικτύου και να εντοπίσει σημεία επικοινωνιακής συμφόρησης. Αυτές οι πληροφορίες μπορούν να χρησιμοποιηθούν και όταν εξετάζονται πιθανές αναβαθμίσεις υλικού.

2.3.4 Λογισμικό και άδειες χρήσης

Στο πλέγμα κάποια μηχανήματα μπορεί να έχουν εγκατεστημένο λογισμικό που είναι πολύ ακριβό για να εγκατασταθεί σε καθένα από αυτά.

Σε ένα πλέγμα οι αντίστοιχες εργασίες στέλνονται στα μηχανήματα με το εξειδικευμένο λογισμικό και εκτελούνται εκεί. Για πολύ ακριβές εφαρμογές λογισμικού αυτό σημαίνει αισθητή μείωση των εξόδων σε ένα οργανισμό.

Ορισμένες άδειες επιτρέπουν την εγκατάσταση σε όλα τα μηχανήματα του πλέγματος αλλά περιορίζουν τον αριθμό των εγκατεστημένων εφαρμογών που μπορούν να τρέξουν ταυτόχρονα στο πλέγμα.

2.3.5 Ειδικός εξοπλισμός - Πολιτικές διαμοιρασμού του

Συχνά συναντάμε συνδεδεμένο εξοπλισμό όπως π.χ. ιατρικά μηχανήματα. Σε τέτοιες περιπτώσεις οι διαχειριστές δημιουργούν ένα τεχνητό τύπο πόρων που χρησιμοποιείται από τους χρονοπρογραμματιστές για να αναθέσουν τις εργασίες σύμφωνα με κανόνες πολιτικής ή άλλους περιορισμούς. Έτσι κάποια μηχανήματα του πλέγματος μπορεί να περιοριστούν στο να δέχονται μόνο εργασίες που αφορούν ιατρική έρευνα ή άλλα για παράδειγμα να χρησιμοποιούνται μόνο για στρατιωτικούς σκοπούς, αποκλείοντας οποιαδήποτε άλλη εργασία.

Για να μπορούν να εφαρμοσθούν τέτοιου είδους πολιτικές διαμοιρασμού θα πρέπει να επιβληθεί από τους διαχειριστές μια ταξινόμηση σε κάθε είδος εργασίας και μέσω κάποιας διαδικασίας πιστοποίησης να δίνεται πρόσβαση σε αυτούς που πρέπει.

2.4 ΕΙΔΗ GRID – ΕΜΠΟΡΙΚΑ ΠΑΡΑΔΕΙΓΜΑΤΑ.

Υπάρχουν πολλά είδη πλέγματος, από ιδιωτικά επιστημονικά/ερευνητικά υπολογιστικά πλέγματα (scientific/research "compute" grids) μέχρι πλέγματα αξιοποίησης και χρήσης (όχι κτήσης) τεχνολογικής υποδομής (utility grids).

Ως επί το πλείστον, τα σημερινά πλέγματα χρησιμοποιούνται για να συναθροίσουν την υπολογιστική δύναμη (computing grids), δεδομένα (data grids) και για συνεργατική μοντελοποίηση και σχεδιασμό (collaborative grids).

Εξελίσσονται όμως ραγδαία και νέοι τύποι πλεγμάτων με σκοπό να παρέχουν συγκεκριμένες λύσεις (όπως π.χ. βελτιστοποίηση χρησιμοποίησης επιχειρηματικών πόρων). Επίσης πολλές κυβερνήσεις θεωρούν τώρα πλέγματα, στρατηγικά σημαντικά για να διατηρήσουν την οικονομική ανταγωνιστικότητά τους στις παγκόσμιες αγορές.

2.4.1 Computing grids

Εστιάζουν στην εκμετάλλευση αχρησιμοποίητης υπολογιστικής ισχύος. Μέχρι σήμερα χρησιμοποιούνταν κυρίως από κοινότητες επιστημονικής έρευνας. Σήμερα εξελίσσονται διαρκώς και χρησιμοποιούνται για την εξερεύνηση πετρελαίου και αερίου, την ανακάλυψη νέων φαρμάκων και την δημιουργία ψηφιακού περιεχομένου. Χρησιμοποιούνται λοιπόν για να αθροίσουν τη δύναμη χιλιάδων προσωπικών αλλά και κεντρικών υπολογιστών — και δημιουργούν ένα περιβάλλον που μπορεί να παρέχει επίπεδα απόδοσης υπερυπολογιστών (αλλά με κόστος πολύ μικρότερο από την αγορά ενός υπερυπολογιστή).

2.4.1.1 ΕΡΕΥΝΗΤΙΚΑ Computing Grids

Μερικά από τα ποιό γνωστά επιστημονικά και ερευνητικά Computing Grids είναι:

Το Seti@Home Project [15] — Οι χρήστες συμμετέχουν μέσω του διαδικτύου εκτελώντας στον υπολογιστή τους ένα δωρεάν πρόγραμμα που μεταφορτώνει και αναλύει ραδιοτηλεσκόπια δεδομένα. (2.7 million years of computer time (2006)).

Η επιτυχία του SETI@Home οδήγησε στην ανάπτυξη του BOINC (Berkeley Open Infrastructure for Network Computing) [17] , μια πλατφόρμα λογισμικού που επιτρέπει την εθελοντική κατ'επιλογή συμμετοχή σε πολλαπλά ερευνητικά πλέγματα με αντικείμενο την Αστρονομία/Φυσική/Χημεία (Spinhenge@home, QuantumMonte Carlo@Home, Einstein@home, LHC@home, SETI@home), τα Μαθηματικά/ Παιχνίδια στρατηγικής (SZTAKI Desktop Grid, Riesel Sieve, Rectilinear Crossing Number, Chess960@home), τη Βιολογία/Φαρμακολογία (SIMAP, Rosetta@home, World Community Grid, Tanpaku, Predictor@home, Malariacontrol.net) και τις Γήινες Επιστήμες (BBC Climate Change Experiment, Climateprediction.net, Seasonal Attribution Project).

Στοιχεία για το BOINC:

- πάνω από 760.000 συμμετέχοντες σε 245 χώρες
- πάνω από 1.000.000 υπολογιστές
- περισσότερα από 400 TeraFLOPS (περισσότερα απ'τον υπερυπολογιστή BlueGene)
- πάνω από 12 Petabytes ελεύθερου αποθηκευτικού χώρου

Το Great Internet Mersenne Prime Search (GIMPS) [18] – είναι ένα ερευνητικό πρόγραμμα για τον υπολογισμό των Mersenne πρώτων αριθμών. (10,000 peak hosts).

Το NASA Information Power Grid (IPG) [21], συνδέει μεταξύ τους κέντρα της NASA, και μετατρέπει υπερυπολογιστές, υψηλής τεχνολογίας επιστημονικά όργανα ,και terabyte δεδομένων σε ένα ενιαίο, υπολογιστικό περιβάλλον, στη διάθεση των επιστημόνων της NASA.

Το World Community Grid [22] (415.000 hosts) είναι μια προσπάθεια για τη δημιουργία του μεγαλύτερου δημόσιου, δημόσιου computing grid για ανθρωπιστικούς

σκοπούς υποστηρίζοντας δράσεις όπως: Help Defeat Cancer, Human Proteome Folding, FightAIDS@Home .

2.4.1.2 Εμπορικά Computing Grids

Τα τελευτά χρόνια έχουν δει μεγάλη ανάπτυξη τα Εμπορικά Computing Grids (ιδιαίτερα στις βιολογικές επιστήμες και τις αγορές ενέργειας). Μερικά από αυτά παρουσιάζονται παρακάτω:

Επιστήμες της Υγείας.

Σήμερα σχεδόν κάθε σημαντική φαρμακευτική επιχείρηση χρησιμοποιεί computing grids για την ανακάλυψη νέων φαρμάκων. Η Johnson & Johnson [23], είναι ο πρωτοπόρος στον κόσμο κατασκευαστής των προϊόντων υγειονομικής περίθαλψης, καθώς επίσης και πάροχος σχετικών υπηρεσιών υγείας, φαρμάκων, ιατρικών συσκευών και διαγνωστικών μηχανημάτων. Χρησιμοποιεί μια καινοτόμα αρχιτεκτονική computing grid που επιτρέπει την άκρως οικονομικότερη υλοποίηση των παραπάνω αναφερόμενων δραστηριοτήτων της. Όμοια η Novartis AG [24] είναι ένας από τους παγκόσμιους ηγέτες στην ανακάλυψη φαρμάκων για προστασία της υγείας και την θεραπεία ασθενειών. (Σημειώνεται ότι και οι δυο είναι στην λίστα των 50 “most admired companies” του Fortune [25] για το 2006).

Οικονομική - Χρηματιστηριακή Αγορά

Οι εταιρίες αυτού του χώρου έχουν τους τρόπους να μειώνουν τους χρόνους επεξεργασίας των στοιχείων τους και να ανοίγουν νέες επιχειρηματικές ευκαιρίες στο χώρο τους, με τη χρήση των Grid. Η Wachovia Corporation [26] μια από τις μεγαλύτερες επιχειρήσεις οικονομικών υπηρεσιών στον κόσμο εκμεταλλεύεται το Grid για να παρέχει σε 13,4 εκατομμύρια νοικοκυριά και επιχειρήσεις μια ευρεία σειρά τραπεζικών εργασιών, διαχείρισης ενεργητικού, διαχείρισης πλούτου και προϊόντα και υπηρεσίες εταιρικών και τραπεζικών επενδύσεων. Όμοια η JPMorgan Chase & Co [28]. είναι παγκόσμια εταιρία οικονομικών υπηρεσιών με ενεργητικό $1,3 τρισεκατομμυρίων και λειτουργία σε περισσότερες από 50 χώρες. Η εταιρία είναι ηγέτης στις τραπεζικές επενδύσεις, οικονομικές υπηρεσίες για καταναλωτές,

τραπεζικές εργασίες μικρών επιχειρήσεων, οικονομική επεξεργασία δοσοληψιών και διαχείριση πλούτου.

Αγορά Ενέργειας

Οι Ενεργειακές Αγορές (π.χ.: εξερεύνηση πετρελαίου και αερίου) έχουν αρχίσει να χρησιμοποιούν τα grid για απαιτητικές σε υπολογισμούς σεισμολογικές εφαρμογές και εφαρμογές απεικόνισης. Με αυτές οι επιχειρήσεις αναλύουν γεωλογικά δεδομένα και το που πρέπει να κάνουν γεωτρήσεις για πετρέλαιο/αέριο.

Η επιχείρηση πετρελαίου της Shell [28] που εκμεταλλεύεται τα πλεονεκτήματα του Grid είναι ένας από τους κύριους παραγωγούς πετρελαίου και φυσικού αερίου της Αμερικής, εμπορεύεται φυσικό αέριο και βενζίνη και κατασκευάζει και πετροχημικά.

Και η BP [29] που είναι μια από τις μεγαλύτερες στον κόσμο επιχειρήσεις ενέργειας, χρησιμοποιεί αρχιτεκτονικές Grid και παρέχει στους πελάτες της καύσιμα για μεταφορά, ενέργεια για τη θέρμανση και φωτισμό, λιανικές υπηρεσίες και τα προϊόντα πετροχημικών για καθημερινές χρήσεις.

2.4.2 Data grids – Πλέγματα δεδομένων

Εστιάζουν στην ανάλυση δεδομένων. Αυτού του είδους τα πλέγματα συγκεντρώνουν πόρους για την εξερεύνηση βάσεων δεδομένων με σκοπό να ενισχύσουν δραστηριότητες: υποστήριξης αποφάσεων και ανάσυρσης δεδομένων. Τα data grids είναι αρκετά δημοφιλή στην οικονομική κοινότητα και χρησιμοποιούνται για την διενέργεια οικονομικών αναλύσεων και αναλύσεων χαρτοφυλακίων (financial and portfolio analysis).

Έτσι αν ένα πληροφοριακό σύστημα μπορεί να παραγάγει μια ανάλυση αποθεματικού σε λιγότερο χρόνο (με την χρήση data grid), κατόπιν η εταιρία και ο πελάτης ωφελούνται με το να εκτιμούν την εμπορική δυναμική τους γρηγορότερα από τους πελάτες άλλων εταιριών. Με αυτό το πλεονέκτημα, οι πελάτες μπορούν να μεγιστοποιήσουν τα κέρδη τους και οι εταιρίες να αποκτήσουν συγκριτικό πλεονέκτημα έναντι των ανταγωνιστών τους.

Ο γιγάντιος τραπεζικός οργανισμός HSBC [30] [31] χρησιμοποιεί αριθμό πλεγμάτων Grid σε όλο τον κόσμο (3,000 CPUs - 1,500 servers - 16 grids) που συνδέονται μεταξύ τους και χρησιμοποιούνται κυρίως από τα τμήματα επενδύσεων του τραπεζικού οργανισμού. Οι αναλύσεις παραγώγων και η αναλύσεις κινδύνου απαιτούν μεγάλα ποσά υπολογιστικής δύναμης που μόνο τα data grids μπορούν να προσφέρουν, με μικρό συγκριτικά κόστος .

Η Deutsche Bank [32] χρησιμοποιεί Data grids για την ανάλυση κινδύνου χαρτοφυλακίων, ενώ το παράρτημά της στη Νέα Υόρκη εξετάζει καθημερινά πάνω από 1000 σενάρια για την ανάλυση της παγκόσμιας αγοράς και παρουσιάζει τα αποτελέσματά στο τέλος της ημέρας.

2.4.3 Collaborative grids – Συνεργατικά πλέγματα

Εστιάζουν στην απεικόνιση και τη διανομή μεγάλων, απαιτητικών σε γραφικά αρχείων μεταξύ ομάδων που συνεργάζονται (και είναι συχνά γεωγραφικά διασκορπισμένες). Αυτές ομάδες συνεργασίας χρησιμοποιούν τα πλέγματα για να σχεδιάσουν πολύπλοκα μοντέλα ή εικόνες σε πραγματικό χρόνο.

Η βιομηχανία κατασκευών χρησιμοποιεί την απέραντη δύναμη επεξεργασίας του πλέγματος για να υποστηρίξει την μοντελοποίηση και το σχεδιασμό σύνθετων προϊόντων (αυτό είναι ιδιαίτερα εμφανές στις αυτοκινητικές και αεροδιαστημικές βιομηχανίες όπου τα πλέγματα χρησιμοποιούνται ως μέσα ενίσχυσης για την σύνθετη εργασία του σχεδιασμού ή για να παρέχουν ικανότητες συνεργασίας μεταξύ των τμημάτων σε μια αλυσίδα σχεδιασμού).

Τόσο η Boeing Company[33] όσο και η Airbus[34], οι μεγαλύτερες αεροδιαστημικές βιομηχανίες (παράγουν εμπορικά και στρατιωτικά αεροσκάφη, ηλεκτρονικά και τμήματα από αμυντικά συστήματα, πυραύλους, δορυφόρους κ.α.) χρησιμοποιούν Collaborative Grids, για να ελαττώσουν τους χρόνους σχεδιασμού, ανάπτυξης υλοποίησης. Σχεδιαστές σε όλα τα μέρη του κόσμου αναπτύσσουν π.χ διαφορετικά μέρη του αεροπλάνου, και χρειάζεται να μοιράζονται συνεχώς πληροφορίες καθώς δουλεύουν για την τελική συναρμολόγηση. Με παρόμοιο τρόπο χρησιμοποιούν τα grid και ορισμένες αυτοκινητοβιομηχανίες.

2.4.4 Govermental Grids – Κυβερνητικά πλέγματα

Οι κυβερνήσεις αντιμετωπίζουν τα ίδια προβλήματα με τους εμπορικούς χρήστες: ζητήματα εξελιξιμότητας συστημάτων, αξιοπιστία συστημάτων, περιορισμό δαπανών για αγορά εξοπλισμών, υποεκμετάλλευση υπολογιστικών πόρων. Η τεχνολογία πλέγματος μπορεί να βοηθήσει τις κυβερνήσεις να βελτιώσουν τη χρησιμοποίηση των πόρων τους και να μειώσουν τις δαπάνες.

Κυβερνήσεις περισσότερο τεχνολογικά ώριμες αναγνωρίζουν πια τη στρατηγική σημασία του πλέγματος, ως μέσο για την διατήρηση της ανταγωνιστικότητας τους στις παγκόσμιες αγορές. Πολλές μεγάλες κυβερνήσεις επενδύουν σε προγράμματα πλέγματος με συνέπεια τις καινοτόμες ανακαλύψεις που θα οδηγήσουν σε προϊόντα τα οποία θα χρησιμοποιηθούν με την πώλησή τους στις παγκόσμιες αγορές.

Γνωστά κυβερνητικά πλέγματα είναι αυτά των Ηνωμένων Πολιτειών, της Ινδίας και της Μεγάλης Βρετανίας.

2.4.5 Utility grids - Πλέγματα αξιοποίησης / χρήσης (όχι κτήσης) τεχνολογικής υποδομής.

Αυξάνονται συνεχώς στις εμπορικές αγορές. Το όφελος που προσφέρουν είναι ότι επιτρέπουν στους χρήστες πληροφοριακών υποδομών να ρυθμίσουν τις δαπάνες, ή να τις μεταχειριστούν έτσι ώστε να ρυθμίσουν τις συνολικές μηνιαίες ή ετήσιες δαπάνες. Εάν υπάρχει ανάγκη για περισσότερη υπολογιστική ισχύ ή αποθήκευση οι διαχειριστές πληροφοριακών υποδομών μπορούν να αποκτήσουν τους πόρους που χρειάζονται, της στιγμή που τους χρειάζονται — χωρίς να πρέπει απαραιτήτως να αγοραστεί νέος εξοπλισμός. Μια επιχείρηση μπορεί για να πληρώσει μόνο για την υπολογιστική δύναμη που χρειάζεται σε ένα πάροχο υπηρεσιών πλέγματος, αποφεύγοντας να αγοράσει νέα συστήματα για να διευθετήσει έτσι καταστάσεις εμφανίζονται απαιτήσεις πέραν του μέγιστου φόρτου εργασιών.

2.4.6 Enterprise optimization grids – Grid επιχειρηματικής βελτιστοποίησης

Κάνουν ακριβώς ότι δηλώνει το όνομά τους — βοηθούν τις επιχειρήσεις να αυξήσουν τη χρησιμοποίηση των πόρων τους μέσω ενός βελτιστοποιημένου σχεδίου πλέγματος. Αυτά τα πλέγματα εστιάζουν στην παροχή αυξανόμενων πόρων υπολογισμού και καλύτερης χρησιμοποίησης των συστημάτων αποθήκευσης για τις επιχειρήσεις που δοκιμάζουν να αυξήσουν την αποδοτικότητα του κεφαλαίου που επενδύουν σε νέα συστήματα υπολογιστών και αποθήκευσης.

2.5 ΤΡΟΠΟΙ ΑΝΑΠΤΥΞΗΣ ΕΜΠΟΡΙΚΩΝ GRID

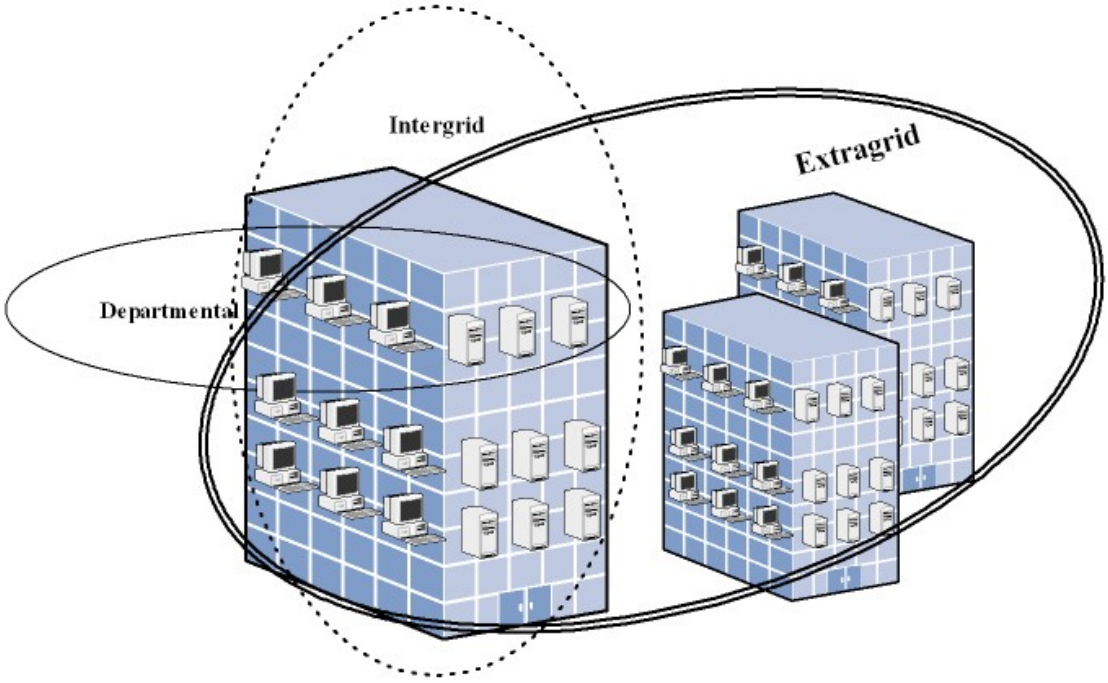
Τα Grid μπορούν να υλοποιηθούν σε διάφορα μεγέθη, που κυμαίνονται από μόνο μερικά μηχανήματα στα όρια ενός τμήματος, μέχρι ομάδες μηχανημάτων που οργανώνονται ιεραρχικά και εκτείνονται σε όλο τον κόσμο.

Γενικότερα τα υπολογιστικά πλέγματα αναπτύσσονται:

Σε επίπεδο τμημάτων (departmental grids)

Σε μια ολόκληρη επιχείρηση (intergrids)

Μεταξύ πολλαπλών επιχειρήσεων ή οργανώσεων (extragrids).



Σχήμα 2.4 Τρόποι ανάπτυξης εμπορικών Grid

2.5.1 Departmental grids

Τα departmental grids αναπτύσσονται στα όρια ενός τμήματος μιας επιχείρησης. Αποτελούν το απλούστερο είδος πλέγματος όχι μόνο λόγω της έκτασης τους αλλά και λόγω του ότι συχνά τα μηχανήματα που το συνθέτουν βασίζονται στην ίδια αρχιτεκτονική υλικού/λογισμικού. Συνήθως τέτοιες υλοποιήσεις δεν απαιτούν την ύπαρξη πολιτικών διαμοιρασμού ή ειδική πρόνοια για προστασία.

2.5.2 Intergrids

Τα Inter-departmental grids είναι πλέγματα που ενοποιούν τους υπολογιστικούς και αποθηκευτικούς πόρους μιας επιχείρησης και τους διαθέτουν σε όλα τα τμήματά της.

Όλο και περισσότερες επιχειρήσεις αναπτύσσουν πλέγματα μεταξύ των τμημάτων τους - επιτρέποντάς τους να μοιραστούν δεδομένα και να συνεργαστούν ευκολότερα ενώ παράλληλα παρέχουν στην επιχείρηση πρόσβαση σε πρόσθετους και αχρησιμοποίητους υπολογιστικούς πόρους.

Απαιτούν την ύπαρξη πολιτικών διαμοιρασμού που ρυθμίζουν: το είδος των εργασιών που επιτρέπεται να εκτελούνται και το πότε, τις προτεραιότητες μεταξύ των τμημάτων ή μεταξύ των εργασιών για την χρήση των πόρων.

Σε τέτοια πλέγματα που εκτείνονται σε πολλά τμήματα πρέπει να ληφθεί μέριμνα για προστασία, π.χ. σε περιπτώσεις η πρόσβαση σε ευαίσθητα δεδομένα ενός τμήματος δεν πρέπει να είναι δυνατή από εργασίες άλλων τμημάτων.

Στον εμπορικό κόσμο, τα περισσότερα grid οργανώνονται πίσω από την εταιρικά τείχη προστασίας (corporate firewalls) σε υλοποιήσεις που περιλαμβάνουν ένα ή περισσότερα τμήματα της επιχείρησης.

2.5.3 ExtraGrids

Οι περισσότερο εντατικές εφαρμογές σε όλο τον κόσμο συναντώνται στην τεχνική εφαρμοσμένη μηχανική και τις επιστημονικές ερευνητικές κοινότητες, γι’αυτό δεν πρέπει να αποτελεί έκπληξη το γεγονός ότι μέχρι σήμερα, ένα μεγάλο μέρος της ανάπτυξης των grid έχει εστιαστεί στη δημιουργία compute grids για την υποστήριξη σύνθετων επιστημονικών υπολογισμών και προσομοιώσεων.

Η μεγάλη πλειοψηφία των extragrids σήμερα είναι compute/data grids που χρησιμοποιούνται για επιστημονικούς και ερευνητικούς σκοπούς. Τα Extragrids για την παροχή υποδομών δεδομένων/συνεργασίας/υπολογιστικών πόρων μεταξύ επιχειρησιακών συνεργατών δεν είναι ιδιαίτερα εξαπλωμένα. Αιτία της καθυστέρησης της εξάπλωσής τους είναι ότι λίγες επιχειρήσεις είναι πρόθυμες να αφήσουν τον οποιονδήποτε (ακόμη και έμπιστους επιχειρησιακούς συνεργάτες τους) να αποκτήσουν πρόσβαση σε συστήματα που περιέχουν ζωτικής σημασίας και ευαίσθητες πληροφορίες για την λειτουργία της επιχείρησης — ακόμα κι αν οι επιχειρησιακοί συνεργάτες έμπιστοι και τα εσωτερικά επιχειρησιακά συστήματα και δεδομένα μπορούν να γίνουν ασφαλή.

Σε "ExtraGrids" που εκτείνονται σε μεγάλες γεωγραφικές εκτάσεις συναντάμε συχνά μισθωμένους επικοινωνιακούς διαύλους μεταξύ των εγκαταστάσεων του πλέγματος. Επίσης χρησιμοποιούνται τεχνικές VPN tunneling ή άλλες τεχνολογίες για να συνδέσουν μέσω του Διαδικτύου τους κόμβους του πλέγματος.

Οι απαιτήσεις για ασφάλεια στα extragrids είναι ιδιαίτερα αυξημένες καθώς ξεπερνιούνται τα φυσικά όρια οποιαδήποτε εγκατάστασης και είναι ιδιαίτερα επικίνδυνο να εμφανιστούν φαινόμενα επιθέσεων ή κατασκοπείας.

2.6 ΑΝΑΛΥΣΗ ΠΛΕΟΝΕΚΤΗΜΑΤΩΝ, ΕΜΠΟΡΙΚΗΣ ΧΡΗΣΗΣ GRID. ΔΕΙΚΤΕΣ ΕΚΤΙΜΗΣΗΣ ΑΥΤΩΝ.

Η εμπορική χρήση υποδομών πλέγματος κερδίζει συνεχώς έδαφος με την εξάπλωσή τους στο χώρο των επιχειρήσεων. Οι εταιρίες που έχουν ήδη αναπτύξει υποδομές grid, εκμεταλλεύονται τα πλεονεκτήματα της τεχνολογίας αυτής αποκτώντας έτσι συγκριτικό πλεονέκτημα έναντι των ανταγωνιστών τους.

Η αξιολόγηση νέων τεχνολογιών στον επιχειρηματικό κόσμο γίνεται με πολύ αυστηρά κριτήρια. Βασικότερο αυτών είναι ότι οποιαδήποτε επένδυση σε νέες τεχνολογίες θα πρέπει να αποφέρει κέρδη στην εταιρία. Έτσι και για τις υποδομές πλέγματος είναι απαραίτητο να αποδεικνύεται ότι βελτιώνουν ουσιαστικά τα επιχειρηματικά αποτελέσματα και κέρδη. Αυτό γίνεται με την ύπαρξη μηχανισμών και δεικτών σύμφωνα με τους οποίους μπορούν με πρακτικό τρόπο να εκτιμηθούν τα οφέλη μιας υποδομής grid.

Κάποιοι από τους δείκτες εκτίμησης πλεονεκτημάτων είναι:

- Επίδοση των εφαρμογών πλέγματος και του όγκου δεδομένων που χρησιμοποιούν ή διαχειρίζονται.
- Απόδοσης επί της Επένδυσης- ROI(Return on Investment) & Χρόνος Απόσβεσης (Break Even).
- Συνολικό Κόστος Ιδιοκτησίας - TCO (Total Cost of Ownership).
- Επιπρόσθετη Επιχειρηματική Αξία.

Τα οφέλη που προσδίδει μια υποδομή grid σε μια επιχείρηση είναι επιχειρηματικά αλλά και τεχνολογικά.

2.6.1 Επιχειρηματικά οφέλη

Επιτάχυνση επιχειρηματικών αποτελεσμάτων.

Ανάλογα με το είδος της επιχείρησης αυτό μπορεί να σημαίνει: ταχύτερη παραγωγή προϊόντων, ταχύτερη επίλυση προβλημάτων, ταχύτερη υλοποίηση ή ανάπτυξη υπηρεσιών. Τα παραπάνω μεταφράζονται σε μείωση του χρόνου που τα προϊόντα ή οι υπηρεσίες μιας επιχείρησης φθάνουν στην αγορά (time to market), κάτι που είναι ιδιαίτερα πολύτιμο μεταξύ ανταγωνιστικών εταιριών.

Αύξηση της παραγωγικότητας των εργαζομένων

Μια υποδομή grid απελευθερώνει τους τελικούς χρήστες δίνοντάς τους πρόσβαση σε υπολογιστικούς, αποθηκευτικούς και άλλους πόρους όταν τους χρειάζονται. Επιπλέον δίνεται η δυνατότητα στους υπαλλήλους να κινηθούν εύκολα και γρηγορότερα από ποτέ μεταξύ των διαφόρων φάσεων σχεδιασμού/παραγωγής προϊόντων, ερευνητικών έργων και άλλων.

Ενδυνάμωση της συνεργασίας και προώθηση λειτουργικής ευελιξίας

Εκτός από την συνάθροιση υπολογιστικών πόρων ένα πλέγμα φέρνει κοντά και τους ανθρώπους μιας εταιρίας κάνοντας ευκολότερη την μεταξύ τους συνεργασία. Τα ευρέως διασκορπισμένα τμήματα και οι επιχειρήσεις δημιουργώντας εικονικές οργανώσεις μπορούν να μοιραστούν μεταξύ τους δεδομένα και πόρους.

Κλιμακωτές IT υποδομές

Μια υποδομή πλέγματος επιτρέπει την αποτελεσματική κλιμάκωσή της ώστε να ικανοποιούνται οι μεταβλητές επιχειρησιακές απαιτήσεις: Δημιουργούνται λοιπόν εύκαμπτες, ελαστικές λειτουργικές υποδομές. Έτσι είναι δυνατόν να διευθετηθούν οι γρήγορες διακυμάνσεις των απαιτήσεων/αναγκών των πελατών. Οι υπολογιστικοί και αποθηκευτικοί πόροι αντιδρούν στιγμιαία και ανταποκρίνονται στις ανάγκες που προκύπτουν.

Αύξηση του οφέλους των επενδύσεων IT.

Με την χρήση τεχνολογικής υποδομής οργανωμένης σε πλέγμα, μια επιχείρηση μπορεί να πετύχει μεγιστοποίηση χρησιμοποίησης των πόρων της, ενοποιώντας τους σε ένα ενιαίο εικονικό σύστημα. Έτσι καταφέρνει να εκμεταλλευτεί σχεδόν πλήρως τα υπολογιστικά συστήματα στα οποία έχει ήδη επενδύσει αγοράζοντάς τα ενώ μπορεί να αποφύγει δαπάνες σε νέα αν δεν προσδίδουν σημαντικό όφελος.

Τα υπολογιστικά συστήματα μιας επιχείρησης αυξάνονται συνεχώς και μαζί τους αυξάνεται και το κόστος διαχείρισης τους. Η αρχιτεκτονική πλέγματος επιτρέπει τη

διαχείριση όλων αυτών των συστημάτων ως μια ενιαία οντότητα οδηγώντας σε δραστική μείωση του διαχειριστικού κόστους.

Πρόσβαση σε υπολογιστικούς πόρους υπό τη μορφή υπηρεσιών

Στις σύγχρονες επιχειρήσεις όλο και περισσότερες εταιρικές εφαρμογές παρέχονται ως κατανεμημένες δικτυακές υπηρεσίες. Πρέπει να διευκρινιστεί όμως η σχέση μεταξύ αρχιτεκτονικής πλέγματος (Grid Architecture), αρχιτεκτονικής προσανατολισμένης προς τις υπηρεσίες (SOA - Service-Oriented Architecture) και ροής επιχειρησιακής διαδικασίας (Business Process Flow).

- Grid Architecture - Βασικός ρόλος της αρχιτεκτονικής πλέγματος είναι η αύξηση της χρησιμοποίησης των πόρων μιας επιχείρησης.
- SOA (Service-Oriented Architecture) — Παρότι τα grids μπορούν να λειτουργήσουν χωρίς την ύπαρξη αρχιτεκτονικής προσανατολισμένης προς τις υπηρεσίες (SOA), είναι ιδιαίτερα επιθυμητή η ύπαρξή της.

  Με το συνδυασμό Grid-SOA γίνεται δυναμική ανάθεση πόρων στις διάφορες δικτυακές υπηρεσίες και ξεπερνιούνται προβλήματα επικοινωνίας προγραμμάτων και μεταφοράς δεδομένων — οδηγούμαστε έτσι σε βελτιωμένες διαδικασίες, αυξημένη παραγωγικότητα, μεγαλύτερη ευελιξία, και μικρότερο υπολογιστικό κόστος.
- Business Process Flow — Όταν μια επιχείρηση λειτουργεί με αρχιτεκτονική SOA μπορεί να αρχίσει να τρέχει τις επιχειρησιακές της διαδικασίες με διαφανή τρόπο πάνω από το στρώμα των πληροφοριακών συστημάτων της.

Τα πληροφοριακά συστήματα που χρησιμοποιούν αρχιτεκτονική βασισμένη σε υπηρεσίες Διαδικτύου πάνω από αρχιτεκτονική grid αποτελούν την ιδανικότερη υποδομή για την διαχείριση και την εκτέλεση επιχειρησιακών διαδικασιών.

2.6.2 Τεχνικά- Τεχνολογικά Πλεονεκτήματα

Οι υποδομές πλέγματος παρέχουν ένα αριθμό τεχνικών και τεχνολογικών πλεονεκτημάτων ανεξάρτητα με το οικονομικό όφελος που αυτά μπορούν να προσδώσουν.

- Εδραιώνεται η διαχείριση του φόρτου εργασιών, παρέχεται δυνατότητα εκτέλεσης εφαρμογών υψηλών απαιτήσεων και μειώνονται οι κύκλοι ζωής των διεργασιών.
- Δημιουργούνται αξιόπιστες υποδομές και σε ένα βαθμό αυτοδιαχειριζόμενες, υπό την έννοια ότι είναι σε θέση να αντιδρούν και να ανταποκρίνονται σε αλλαγές συνθηκών και απαιτήσεων.
- Αυξάνεται η πρόσβαση σε δεδομένα και η συνεργασία. Συνενώνονται σε ομάδες τα δεδομένα και διανέμονται σε όλο το εύρος του πλέγματος. Υποστηρίζονται μεγάλες διεπιστημονικές συνεργασίες και ενδυναμώνεται η συνεργασία μεταξύ οργανισμών και επιχειρήσεων.
- Δημιουργείται ελαστική υποδομή με υψηλά επίπεδα διαθεσιμότητας.
- Εξισορροπείται ο φόρτος εργασίας.
- Δυνατότητα αποκατάστασης μετά από λειτουργικές αστοχίες.
- Προκύπτουν κλιμακούμενες υποδομές που μπορούν εύκολα να επεκταθούν για να ικανοποιήσουν περισσότερους χρήστες αλλά και πιο πολύπλοκες επιχειρησιακές απαιτήσεις.

2.7 ΠΡΟΣΠΑΘΕΙΕΣ ΠΡΟΤΥΠΟΠΟΙΗΣΗΣ GRID

2.7.1 Global Grid Forum (GGF)

Λαμβάνοντας υπόψη την μεγάλη ανάπτυξη υλικολογισμικού πλέγματος, η ανάπτυξη «προτύπων» είναι σημαντική για να εξασφαλίσει τη διαλειτουργικότητα μεταξύ διαφορετικών προϊόντων και εφαρμογών Grid. Τα τελευταία χρόνια, οι προσπάθειες για την τυποποίησης πλέγματος καθοδηγούνται από το Global Grid Forum (GGF)[35].

Το Global Grid Forum (GGF) είναι η κοινότητα χρηστών, υπεύθυνων για ανάπτυξη και προμηθευτών που οδηγούν την παγκόσμια προσπάθεια προτυποποίησης για τα Grid. Οι δύο κύριες λειτουργίες του GGF είναι η ανάπτυξη προτύπων σχετικά με τα Grid και η δημιουργία κοινοτήτων εντός της συνολικής κοινότητας Grid προσπαθώντας να καλύψει την ευρύτερη δυνατή συμμετοχή φορέων τόσο από τον ακαδημαϊκό κόσμο όσο από τη βιομηχανία. Το GGF διαμορφώνει ανοικτά πρότυπα σύμφωνα με την IETF (Internet Engineering Task Force). Τα πρότυπα OGSA, OGSI και JSDL δημιουργήθηκαν από το GGF.

Το πρότυπο Open Grid Services Architecture (OGSA) περιγράφει μια αρχιτεκτονική για ένα προσανατολισμένο προς τις υπηρεσίες υπολογιστικό περιβάλλον grid (service-oriented grid computing) κατάλληλο για επιχειρήσεις αλλά και για επιστημονική χρήση. Βασίζεται σε διάφορες άλλες Τεχνολογίες υπηρεσιών Ιστού, ειδικότερα τις WSDL και SOAP.

Αυτό που ο OGSA προσπαθεί να κάνει, είναι να εναρμονίσει την εργασία που γίνεται - κυριώς από ακαδημαϊκούς - για την ανάπτυξη του πακέτου εργαλείων Globus με τις αποκαλούμενες «Υπηρεσίες Ιστού» (Web Services) που ωθούν οι βιομηχανίες.

2.7.2 EGA (Enterprise Grid Alliance)

Ο EGA (Enterprise Grid Alliance)[36] είναι μαζί με τον GGF (Global Grid Forum) οι δυο μεγαλύτεροι οργανισμοί που ασχολούνται με την προτυποποίηση του grid computing. Ο EGA αναπτύσσει επιχειρηματικές λύσεις grid και επιταχύνει την

επέκταση του grid computing στις επιχειρήσεις. Εξετάζει τα εμπόδια που οι συναντούν οι οργανισμοί στη χρησιμοποίηση επιχειρηματικών grid και προτείνει ανοιχτές διαλειτουργικές λύσεις. Μέχρι σήμερα εστίαζε κυρίως στις ανάγκες των επιχειρηματικών χρηστών και συνέβαλε στην εξοικείωση των επιχειρήσεων με τα grid.

2.7.3 Open Grid Forum (OGF)

Τον Ιούλιο του 2006 ανακοινώθηκε η σύμπραξη του GGF με την EGA (Enterprise Grid Alliance) για το σχηματισμό ενός νέου οργανισμού με ονομασία "Open Grid Forum" (OGF). [20]

Ο νέος αυτός οργανισμός συγκεντρώνει τις δύο κορυφαίες οργανώσεις προτυποποίησης grid της βιομηχανίας, για να επιταχύνει την κυρίαρχη υιοθέτηση των grid παγκοσμίως. Ο OGF θα ωφεληθεί από την εμπειρία του EGA στο χώρο των επιχειρήσεων και θα εστιάσει σε βραχυπρόθεσμα πρακτικά αποτελέσματα που συνδέονται με την ανοικτή, συνεργατική προσέγγιση του GGF στην έρευνα πλέγματος, στις γνωστές του πρακτικές και στην εμπειρία του στην ανάπτυξη προτύπων .

Στο εξής λοιπόν

GGF (Global Grid Forum)+ EGA (Enterprise Grid Alliance)=OGF (Open Grid Forum).

3

Οικονομικά Μοντέλα Διαχείρισης Πόρων σε Υπολογιστικά Πλέγματα.

Το Grid κάνει δυνατή τη συγκέντρωση και το διαμοιρασμό εκατομμυρίων πόρων που μπορεί να είναι γεωγραφικά διεσπαρμένοι σε όλο τον κόσμο ανήκοντας σε διαφορετικούς οργανισμούς ή διαχειριστικές περιοχές. Περιλαμβάνει ετερογενείς πόρους (προσωπικούς υπολογιστές, τερματικά, συστάδες υπολογιστών, υπερυπολογιστές), συστήματα διαχείρισης λειτουργίας (λειτουργικά συστήματα, συστήματα αναμονής), πολιτικές και εφαρμογές (επιστημονικές, εφαρμοσμένης μηχανικής, εμπορικές) με ποικίλες απαιτήσεις (ΚΜΕ, I/O, μνήμης, χωρητικότητας δικτύου). Οι πόροι ανήκουν σε διαφορετικές οργανώσεις καθεμία με τις δικές της: πολιτικές διαχείρισης, μοντέλα χρήσης και χρέωσης για διαφορετικούς χρήστες σε διαφορετικές στιγμές. Η διαθεσιμότητα των πόρων και το φορτίο τους ποικίλλει δυναμικά με το χρόνο.

Σε ένα τέτοιο περιβάλλον πλέγματος, οι παραγωγοί (ιδιοκτήτες των πόρων) και οι καταναλωτές (χρήστες των πόρων) έχουν διαφορετικούς σκοπούς, στόχους, στρατηγικές, και σχέδια ανάλογα με την προσφορά-και-ζήτηση πόρων. Επιπλέον τόσο οι πόροι όσο και οι τελικοί χρήστες είναι γεωγραφικά διανεμημένοι σε διάφορα σημεία του κόσμου σε διαφορετικές χρονικές ζώνες.

Η διαχείριση των πόρων είναι πρόκληση που προκύπτει από τα έμφυτα χαρακτηριστικά του πλέγματος και πρέπει γίνεται με βιώσιμα, εξελικτικά πρότυπα και πολιτικές που θα αναδείξουν και προωθήσουν ως πολύτιμες τις υπηρεσίες του πλέγματος, ώστε να είναι ελκυστικό για όλους τους συμμετέχοντες σε αυτό. Μόνο έτσι θα μπορέσει το Grid να προωθηθεί ως επικρατούσα τάση και να οδηγήσει σε μια βιομηχανία υπολογιστών προσανατολισμένη στην παροχή υπηρεσιών στο χρήστη, με βάση τις απαιτήσεις του.

3.1 Γιατί οι παραδοσιακές προσεγγίσεις απόδοσης πόρων αποτυγχάνουν.

Για τις αρχιτεκτονικές διαχείρισης των πόρων έχουν προταθεί διάφορες προσεγγίσεις όπως συγκεντρωμένη, αποκεντρωμένη, και ιεραρχική αρχιτεκτονική. Αυτές αντιμετωπίζουν διαφορετικά ζητήματα αποδοτικότητας, σταθερότητας και εξελιξιμότητας των πλεγμάτων.

Ταξινομούνται σε:

- Παραδοσιακές προσεγγίσεις (control theory)
- Οικονομικές προσεγγίσεις βασισμένες σε κανόνες της αγοράς (economics).

Στη διαχείριση της πολυπλοκότητας που εμφανίζεται σε μεγάλης κλίμακας πλέγματα, οι παραδοσιακές προσεγγίσεις είναι ακατάλληλες δεδομένου ότι προσπαθούν να βελτιστοποιήσουν το μέτρο απόδοσης του συνολικού συστήματος. Χρησιμοποιούν συγκεντρωτικές πολιτικές που χρειάζονται τις πλήρεις πληροφορίες της κατάστασης του πλέγματος και κοινή διαχειριστική πολιτική ή χρησιμοποιούν αποκεντρωμένες πολιτικές βασισμένες στη συναίνεση μεταξύ των συμβαλλόμενων μερών. Εξαιτίας όμως της πολυπλοκότητας του πλέγματος είναι δύσκολο να καθοριστεί μια αποδεκτή σε όλο το σύστημα μήτρα απόδοσης και μια κοινή πολιτική διαχείρισης. Τα στοχαστικά λοιπόν μοντέλα απόκρισης απόδοσης του πλέγματος παραμένουν αόριστα, καθιστώντας το δύσκολο να καθορίσουν τυπικά, θεωρητικούς μηχανισμούς

ελέγχου. Επομένως, οι ιεραρχικές και αποκεντρωμένες προσεγγίσεις είναι καταλληλότερες για την απόδοση των πόρων και την διαχείριση λειτουργίας του όλου συστήματος. Μέσα σε αυτές τις προσεγγίσεις, συναντάμε διάφορα οικονομικά μοντέλα για τη διαχείριση και τη ρύθμιση προσφοράς-και-ζήτησης για πόρους.

Στις οικονομικές προσεγγίσεις γίνεται χρήση πραγματικών οικονομικών προτύπων και στρατηγικών που τα δανειζόμαστε από τον κόσμο της αγοράς. Όπως σε μια πραγματική οικονομία χρησιμοποιείται ένα βιώσιμο πρότυπο για την ανταλλαγή και τη ρύθμιση των πόρων, αγαθών και υπηρεσιών με τη χρήση νομίσματος, έτσι και σε ένα grid μπορεί να αντιμετωπισθεί με οικονομικούς όρους.

Σε ένα τέτοιο οικονομικό μοντέλο οι ιδιοκτήτες των πόρων διαθέτουν (πωλούν) τη χρήση των πόρων τους σε χρήστες οι οποίοι πληρώνουν για την πρόσβαση στους πόρους, με βάση το είδος αυτών και το χρόνο χρήσης τους. Δημιουργείται έτσι ένα πλαίσιο που προσφέρει ένα κίνητρο στους ιδιοκτήτες των πόρων για διάθεση και διανομή των πόρων τους. Παράλληλα παρακινεί τους χρήστες να αγοράσουν υπηρεσίες και να υλοποιήσουν τις συναλλαγές τους ανάλογα με τις απαιτήσεις ποιότητας των υπηρεσιών (QoS) που ζητούν.

3.2 Ανάγκη για βασισμένη σε οικονομικές αρχές διαχείριση πόρων Grid.

Σε ένα πλέγμα όπου οι πάροχοι διαθέτουν τους πόρους τους και οι χρήστες πληρώνουν για να αγοράσουν υπηρεσίες, η ανάγκη για μια προσανατολισμένη σε οικονομικούς όρους διαχείριση των πόρων προέρχεται από απαντήσεις σε ερωτήματα όπως:

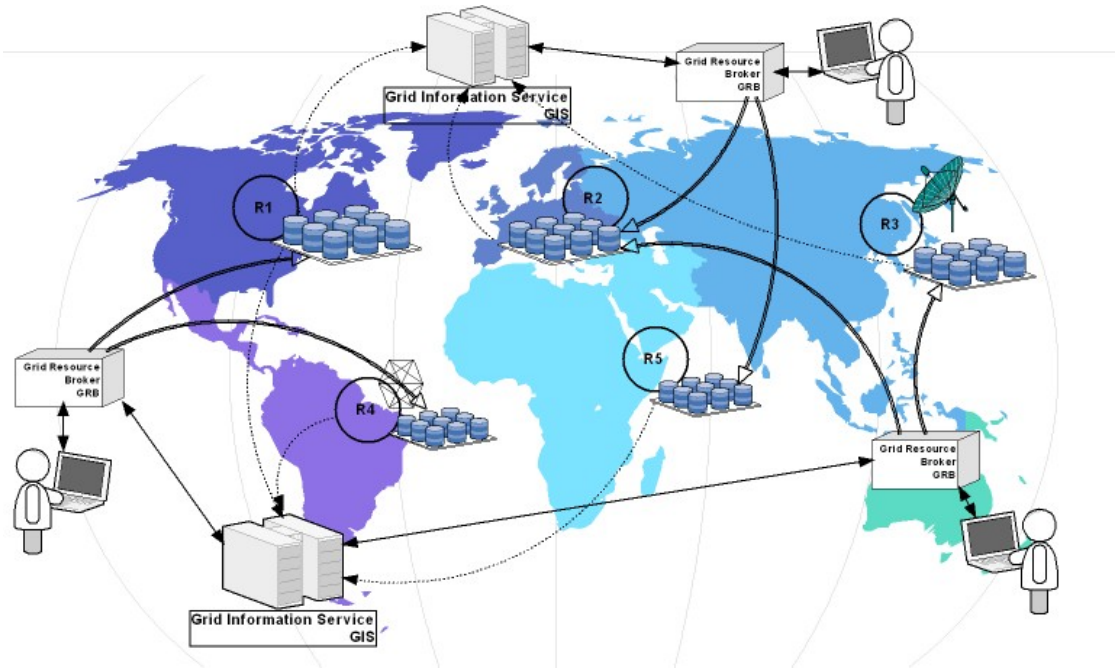
- Τι παρακινεί τους ιδιοκτήτες πόρων να συμβάλλουν με τη διάθεση των πόρων τους στο πλέγμα;
- Είναι δυνατό να έχουμε πρόσβαση σε όλους τους πόρους του πλέγματος, αν συμβάλουμε και εμείς με τη διάθεση των δικών μας πόρων;
- Αν όχι, πώς μπορούμε να έχουμε πρόσβαση σε όλους τους πόρους πλέγματος;

- Εάν έχουμε πρόσβαση στους πόρους μέσω συνεργασίας, μας επιτρέπεται να λύσουμε εμπορικά προσωπικά μας προβλήματα;
- Οι ιδιοκτήτες των πόρων θα πρέπει να χρεώνουν την ίδια ή διαφορετική τιμή για τους διαφορετικούς χρήστες;
- Είναι το κόστος πρόσβασης το ίδιο για τις ώρες αιχμής και τις μη ώρες αιχμής;
- Πώς μπορούν οι ιδιοκτήτες των πόρων να μεγιστοποιήσουν το κέρδος τους;
- Πώς μπορούν οι χρήστες να λύσουν τα προβλήματά τους με το ελάχιστο κόστος;
- Πώς μπορεί ένας χρήστης να πάρει την υψηλή προτεραιότητα από άλλους;
- Εάν ο χρήστης χαλαρώσει την προθεσμία μέχρι την οποία απαιτεί τα αποτελέσματα, μπορεί να μειωθεί το κόστος;

Ένα οικονομικό μοντέλο διαχείρισης πόρων μπορεί να δώσει λύσεις σε τέτοια ερωτήματα τη στιγμή που άλλα μοντέλα αδυνατούν. Μια Αγορά Grid (Grid Market) εκτείνεται γύρω από:

- τους ιδιοκτήτες των πόρων (producers) ως πωλητές που διαθέτουν τους πόρους τους και ανακοινώνουν τις τιμές των υπηρεσιών που προσφέρουν,
- τους καταναλωτές (consumers) ως αγοραστές που επιθυμούν να αγοράσουν υπηρεσίες.
- μερικούς μεσάζοντες (μεσίτες/brokers)

Ο χρήστης-καταναλωτής διατυπώνει τις αιτήσεις των εφαρμογών του σε ένα μεσίτη (broker) ο οποίος ανακαλύπτει τους κατάδηλους πόρους για την εκπλήρωση των απαιτήσεων του πελάτη του. Ο μεσίτης μετά από διαπραγματεύσεις επιλέγει ποια ή ποιες υπηρεσία θα αγοράσει για λογαριασμό του πελάτη του. Έπειτα φροντίζει για την εκτέλεση της εργασίας και για την επιτυχή ολοκλήρωσή της.



Σχήμα 3.1 Αλληλεπίδραση χρηστών – μεσαζόντων - πόρων

Η χρήση ανταγωνιστικών οικονομικών προτύπων για την διαχείριση των πόρων είναι επιβεβλημένη, αφού διαφορετικοί πάροχοι πόρων και καταναλωτές έχουν διαφορετικούς σκοπούς, στόχους, στρατηγικές, και απαιτήσεις που ποικίλλουν με το χρόνο. Σε τέτοια πλέγματα οι καταναλωτές πόρων υιοθετούν τη στρατηγική να λύνουν τα προβλήματά τους με το χαμηλότερο κόστος μέσα σε ένα απαραίτητο χρονικό πλαίσιο και οι προμηθευτές υιοθετούν τη στρατηγική της καλύτερης πιθανής απόδοσης της επένδυσή τους. Η αξιολόγηση των χρηστών για τη χρησιμότητα μιας υπηρεσίας που δύναται να παρασχεθεί είναι χρονικά εξαρτώμενη από την εικόνα του πλέγματος κάθε στιγμή.

Προκειμένου να αυξηθεί η αξία χρησιμότητας και ποιότητας των υπηρεσιών που παραδίδεται στους συμμετέχοντες (παρόχους, χρήστες) του πλέγματος, πρέπει να παρέχονται τα ακόλουθα σε ένα τέτοιο οικονομικό μοντέλο:

1. ένα μέσο για να μπορούν να εκφράσουν οι συμμετέχοντες τις απαιτήσεις, τις αποτιμήσεις και τους στόχους τους,
2. τακτικές χρονοπρογραμματισμού που θα μεταφράσουν τις παραπάνω (απαιτήσεις) σε πολιτικές κατανομής πόρων.

3. μηχανισμοί για να επιβάλλουν την επιλογή και την κατανομή των διαφορετικών υπηρεσιών. Αυτοί οι μηχανισμοί θα πρέπει να προσαρμόζονται στις αλλαγές της διαθεσιμότητας πόρων οποιαδήποτε χρονική στιγμή.

Επιπλέον για να είναι η αγορά υγιής και ανταγωνιστική θα πρέπει να προβλέπονται στην όλη υποδομή πλέγματος μηχανισμοί συντονισμού που θα βοηθήσουν την αγορά να φτάσει σε μια τιμή ισορροπίας προσφοράς-ζήτησης, όπου η παροχή μιας υπηρεσίας θα είναι ίση με την ποσότητα αυτής που απαιτείται.

3.3 Οφέλη μιας « Grid – Οικονομίας »

Με τη χρήση οικονομικών αρχών ως βάση για τις πολιτικές και τους μηχανισμούς κατανομής των πόρων δημιουργείται ένα ελκυστικό περιβάλλον για την περαιτέρω ανάπτυξη των υπολογιστικών πλεγμάτων.

Οι οικονομίες είναι διαισθητικά εύκολο να γίνουν κατανοητές, και μπορούν να αναλυθούν μέσω ενός ιδιαίτερου σώματος της υπάρχουσας θεωρίας που το δανειζόμαστε από τις αγορές-προϊόντων. Οι περισσότεροι άνθρωποι π.χ. καταλαβαίνουν αρκετά γρήγορα ότι μια αύξηση στην αξία ενός πόρου του πλέγματος θα καταστείλει την απαίτηση για αυτό.

Άλλα οφέλη μιας Grid – Οικονομίας είναι τα εξής:

- Συμβάλλει στην οικοδόμηση ενός μεγάλης κλίμακας πλέγματος δεδομένου ότι παρακινεί τους ιδιοκτήτες πόρων να συμβάλουν τους (μη απασχολημένους) πόρους τους, για να τους χρησιμοποιήσουν άλλοι και να ωφεληθούν
- Παρέχει μια δίκαιη βάση για την πρόσβαση στους πόρους πλέγματος για τον καθέναν.
- Βοηθά στην ρύθμιση του ισοζυγίου προσφοράς-ζήτησης για πόρους.

- Προσφέρει οικονομικό τον κίνητρο για τους χρήστες να καθυστερούν την απαίτηση ολοκλήρωσης προβλημάτων χαμηλής προτεραιότητας (ώστε να επιτύχουν μικρότερο κόστος). Ενθαρρύνει όμως τη λύση πρώτα των χρονικά κρίσιμων προβλημάτων.
- Αφαιρεί την ανάγκη για έναν κεντρικό συντονιστή (κατά τη διάρκεια της διαπραγμάτευσης).
- Προσφέρει ομοιόμορφη επεξεργασία όλων των πόρων. Δηλαδή επιτρέπει τις εμπορικές συναλλαγές κάθε είδους πόρων όπως υπολογιστική δύναμη, μνήμη, αποθηκευτικών χώρων, εύρος ζώνης δικτύων, δεδομένα, συσκευές, όργανα.
- Επιτρέπει στους χρήστες να εκφράσουν τις απαιτήσεις και τους στόχους τους.
- Βοηθά στην ανάπτυξη πολιτικών κατανομής πόρων που είναι επικεντρωμένες στο χρήστη παρά στο όλο σύστημα.
- Προσφέρει έναν αποδοτικό μηχανισμό για την κατανομή και τη διαχείριση των πόρων.
- Βοηθά στην οικοδόμηση ενός ιδιαίτερα εξελικτικού συστήματος αφού η διαδικασία λήψης αποφάσεων διανέμεται σε όλους χρήστες και ιδιοκτήτες των πόρων.
- Παρέχει ευελιξία στην περίπτωση αποτυχιών και υψηλή προσαρμοστικότητα κατά τη διάρκεια αλλαγών.
- Τέλος, τοποθετεί τη δύναμη τόσο στα χέρια των ιδιοκτητών και όσο και των χρηστών των πόρων. Δίνεται και στους δυο η δυνατότητα να πάρουν αποφάσεις που θα μεγιστοποιήσουν τη χρησιμότητα και το κέρδος τους.

3.4 Δομή μιας « Grid – Οικονομίας »

3.4.1 Χαρακτηριστικά μιας « Grid – Οικονομίας »

Σε ένα παγκόσμιο περιβάλλον πλέγματος, οι ικανότητες που μια υποδομή πρέπει να υποστηρίζει (άσχετα με το αν χρησιμοποιείται οικονομικό μοντέλο ή όχι) είναι:

- Μακρινή αποθήκευση και/ή τήρηση αντιγράφων δεδομένων
- Δημοσίευση σε κατάλογο κάθε συνόλου δεδομένων που χρησιμοποιείται, με καθολικό όνομα και πλήρη λίστα των χαρακτηριστικών του.
- Ασφάλεια - έγκριση πρόσβασης και επικύρωση.
- Διαφανή πρόσβαση σε απομακρυσμένους πόρους (δεδομένα και υπολογιστικοί πόροι).
- Δημοσίευση των υπηρεσιών που είναι διαθέσιμες στο πλέγμα και του κόστους πρόσβασης σε αυτές.
- Εύρεση των κατάλληλων συνόλων δεδομένων από τα καθολικά ονόματα ή τις ιδιότητές τους.
- Ανακάλυψη των κατάλληλων υπολογιστικών πόρων.
- Χαρτογράφηση και χρονοπρογραμματισμός των εργασιών
- Υποβολή, έλεγχος, καθοδήγηση της εκτέλεσης των εργασιών.
- Μετακίνηση κώδικα/δεδομένων μεταξύ υπολογιστών χρηστών και διανεμημένων πόρων.
- Εφαρμογή της απαιτούμενης ποιότητας υπηρεσιών (QoS).
- Μέτρηση και χρέωση της χρήσης των πόρων.

Μια οικονομική προσέγγιση στο Grid εισάγει όμως διάφορα νέα ζητήματα που πρέπει να αντιμετωπιστούν σε σχέση με αυτά που ήδη υπάρχουν. Στην υποδομή μιας « Grid – Οικονομίας » πρέπει επιπλέον να προβλέπονται τα εξής :

- Ένας κατάλογος πληροφοριών και αγοράς (Information and Market directory) για την κοινοποίηση των διαθέσιμων οντοτήτων (πόρων) πλέγματος.

- Μοντέλα για τον καθορισμό της αξίας των πόρων
- Σχέδια τιμολόγησης των πόρων και μηχανισμοί κοινοποίησης αυτών
- Οικονομικά πρότυπα και πρωτόκολλα διαπραγμάτευσης
- Μεσολαβητές για να ενεργήσουν ως ρυθμιστική αρχή για τον καθορισμό της αξίας των πόρων, των προτύπων νομίσματος, και του χειρισμού κρίσεων.
- Μηχανισμοί τιμολόγησης, χρέωσης και πληρωμής
- Συστήματα που δρουν ως μεσίτες και φροντίζουν για τον προγραμματισμό εκτέλεσης των εργασιών με βάση τις αιτήσεις ποιότητας υπηρεσιών (QoS) των χρηστών.

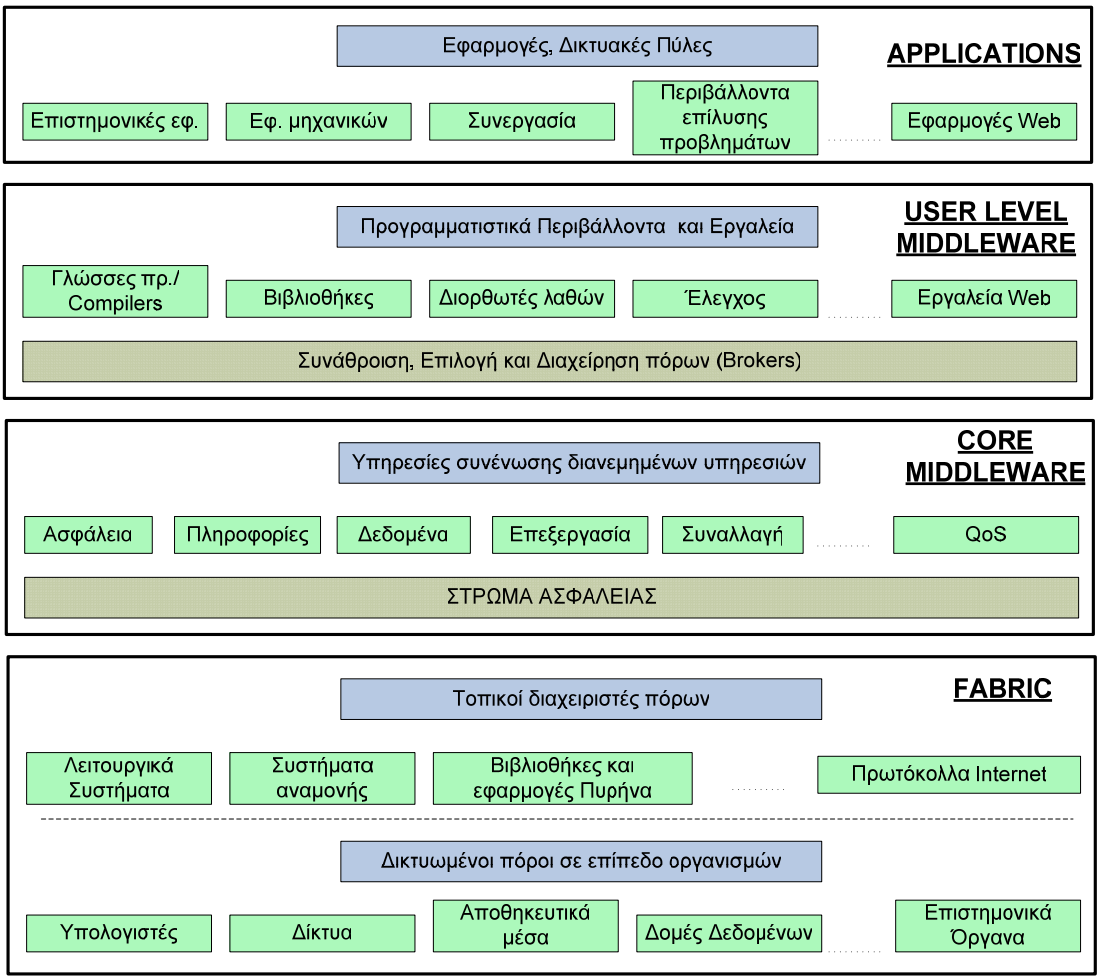
3.4.2 Grid Layers

Τα διάφορα τμήματα πλέγματος που παρέχουν τις ανωτέρω υπηρεσίες οργανώνονται σε στρώματα (Layers). Κάθε στρώμα στηρίζεται στις υπηρεσίες που προσφέρονται από το χαμηλότερο στρώμα και επίσης αλληλεπιδρά - συνεργάζεται με τα συστατικά του ίδιου επιπέδου.

Η στοίβα υλικολογισμικού μιας χαρακτηριστικής αρχιτεκτονικής πλέγματος, αποτελείται από τέσσερα στρώματα:

fabric, core middleware, user-level middleware, and applications and portals layers.

1. δομικό (fabric layer)
2. βασικού υλικολογισμικού (core middleware layer)
3. υλικολογισμικού σε επίπεδο χρήστη (user-level middleware layer)
4. εφαρμογών και δικτυακών πυλών (application and portal layers)



Σχήμα 3.2 Διαστρωμάτωση Αρχιτεκτονικής Υπολογιστικού Πλέγματος

Το Grid Fabric Layer αποτελείται από πηγές υπολογιστικών πόρων , δίκτυα, αποθηκευτικά μέσα και επιστημονικά εργαλεία. Οι πηγές υπολογιστικών πόρων μπορεί να απαρτίζονται από πολλές αρχιτεκτονικές όπως συστοιχίες, υπερυπολογιστές, εξυπηρετητές και απλούς υπολογιστές που τρέχουν διάφορα λειτουργικά συστήματα. Τα επιστημονικά όργανα όπως τα τηλεσκόπια και τα δίκτυα αισθητήρων παρέχουν δεδομένα σε πραγματικό χρόνο που μπορούν να διαβιβαστούν άμεσα για επεξεργασία ή να αποθηκευτούν σε μια βάση δεδομένων.

Το Core Grid middleware layer προσφέρει τις υπηρεσίες όπως απομακρυσμένη διαχείριση, όμοιο-κατανομή των πόρων, πρόσβαση σε αποθηκευτικούς χώρους, καταχώρηση και ανακάλυψη πληροφοριών, ασφάλεια, και πτυχές της ποιότητας της υπηρεσίας (QoS) όπως η δέσμευση πόρων και οι εμπορικές συναλλαγές.

Οι υπηρεσίες αυτές αφαιρούν την πολυπλοκότητα και την ετερογένεια του fabric layer και παρέχουν μια ενιαία μέθοδο για την πρόσβαση σε διανεμημένους πόρους.

Το user-level middleware layer του πλέγματος χρησιμοποιεί τις διεπαφές που παρέχονται από προηγούμενο στρώμα για να παρέχει τις υψηλότερου επιπέδου αφαιρέσεις και υπηρεσίες. Αυτές περιλαμβάνουν τα προγραμματιστικά περιβάλλοντα ανάπτυξης εφαρμογών, εργαλεία προγραμματισμού και τους μεσάζοντες (brokers) για τη διαχείριση των πόρων και τον προγραμματισμό της εκτέλεσης των εφαρμογών.

Οι εφαρμογές και οι πύλες πλέγματος αναπτύσσονται χρησιμοποιώντας τα προγραμματιστικά περιβάλλοντα, τις διεπαφές, τη μεσιτεία και τις υπηρεσίες διαχείρισης πόρων του προηγούμενου στρώματος.

Μια εφαρμογή π.χ., όπως η προσομοίωση παραμέτρων θα απαιτούσε υπολογιστική δύναμη, πρόσβαση σε απομακρυσμένα σύνολα δεδομένων, και μπορεί να χρειαζόταν να αλληλεπιδράσει και με επιστημονικά όργανα.

Οι πύλες πλέγματος προσφέρουν εφαρμογές-υπηρεσίες μέσω δικτύου, όπου οι χρήστες μπορούν να υποβάλουν και να συλλέξουν τα αποτελέσματα των εργασιών τους από μακρινούς πόρους μέσα στον Ιστό.

3.4.3 Λειτουργική Δομή μιας « Grid – Οικονομίας »

Οι δυο βασικοί συμμετέχοντες στις Οικονομίες Grid είναι οι προμηθευτές των πόρων (GSPs - Grid Service Providers – Πάροχοι Υπηρεσιών Πλέγματος) και οι καταναλωτές των πόρων (GRCs - Grid Resource Consumers – Καταναλωτές πόρων πλέγματος).

Οι μεσάζοντες/μεσίτες (GRBs – Grid Resource Brokers – Μεσάζοντες Πόρων Πλέγματος) δρουν ως αντιπρόσωποι των καταναλωτών που εκφράζουν τις απαιτήσεις τους. Οι GTSs - Grid Trade Servers είναι εκπρόσωποι των παρόχων και διαπραγματεύονται για λογαριασμό τους την τιμή των πόρων. Επίσης κατευθύνουν το

σύστημα χρέωσης (accounting system) για την καταγραφή των πόρων που χρησιμοποιούνται και την αντίστοιχη χρέωση που πρέπει να υποβληθεί.

Πάροχοι και καταναλωτές έχουν διαφορετικές προσδοκίες και στρατηγικές όταν συμμετέχουν σε ένα πλέγμα. Σε μια «Οικονομία Grid», οι καταναλωτές των πόρων υιοθετούν τη στρατηγική επίλυσης των προβλήματά τους όσο το δυνατόν πιο φτηνά μέσα σε ένα καθορισμένο χρονικό πλαίσιο. Οι προμηθευτές των πόρων υιοθετούν τη στρατηγική της καλύτερης δυνατής επιστροφής της επένδυσής τους σε τηλεπικοινωνιακές υποδομές. Προσπαθούν να μεγιστοποιήσουν τη χρησιμοποίηση των πόρων τους με την προσφορά ανταγωνιστικού κόστους πρόσβασης σε υπηρεσίες. Οι καταναλωτές έχουν την δυνατότητα να επιλέγουν τους παρόχους που καλύπτουν καλύτερα τις απαιτήσεις τους.

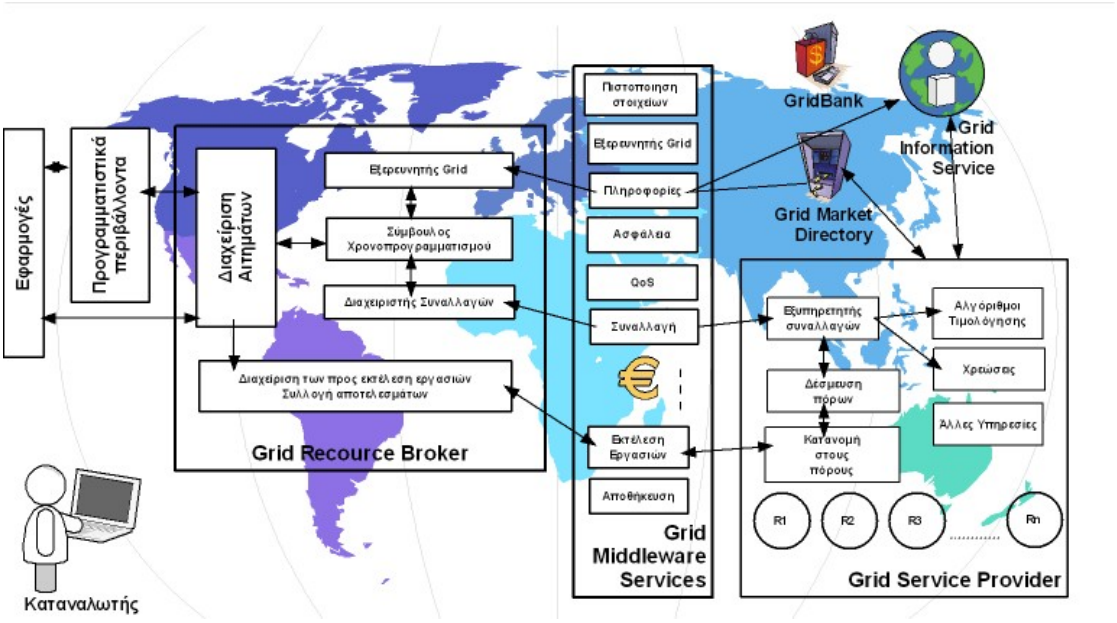
Οι GSPs - Grid Service Providers – Πάροχοι Υπηρεσιών Πλέγματος καταχωρούν τις υπηρεσίες που προσφέρουν με όλα τα χαρακτηριστικά τους σε μέσα σε μια ή περισσότερες Υπηρεσίες πληροφοριών πλέγματος GIS - Grid information services. Παράλληλα ανακοινώνουν τις τιμές των υπηρεσιών τους σε ένα κατάλογο αγοράς πλέγματος GMD - Grid Market Directory με σκοπό να προσελκύσουν χρήστες.

Χρησιμοποιούν επεξηγηματική γλώσσα για τον καθορισμό των υπηρεσιών που προσφέρουν, τον καθορισμό του κόστους και των ειδικών σεναρίων τιμολόγησης όπως: διαφορετικές τιμές ανάλογα με το χρόνο χρήσης ή με την ώρα της ημέρας (φθηνότερες υπηρεσίες σε ώρες μη-αιχμής).

Οι χρήστες αναπτύσσουν την εφαρμογή τους ως κατανεμημένη εφαρμογή με τη χρήση εργαλείων ανάπτυξης λογισμικού. Έπειτα καθορίζουν τις απαιτήσεις της εφαρμογής τους και την ποιότητα υπηρεσίας που επιθυμούν στο μεσίτη των πόρων πλέγματος.

Ο μεσίτης των πόρων πλέγματος χρησιμοποιώντας την υπηρεσία πληροφοριών πλέγματος GIS ανακαλύπτει τους διαθέσιμους πόρους και τα χαρακτηριστικά τους. Ψάχνοντας στον κατάλογο αγοράς πλέγματος Grid market directory προσδιορίζει τις τιμές των πόρων και επιλέγει αυτούς που ανταποκρίνονται στις ανάγκες του χρήστη.

Ελέγχει αν ο χρήστης έχει τις απαραίτητες πιστωτικές μονάδες και αν είναι εξουσιοδοτημένος να χρησιμοποιήσει το πλέγμα. Ακολούθως χρονοπρογραμματίζει την εκτέλεση της εφαρμογής του χρήστη και φροντίζει για την επιτυχή περάτωση της, παρακολουθώντας συνεχώς την ποιότητα των παρεχόμενων υπηρεσιών που έχει συμφωνηθεί. Όταν η εκτέλεση ολοκληρωθεί συλλέγει τα αποτελέσματα και τα προωθεί στο χρήστη. Το σύστημα χρέωσης - accounting system υπολογίζει το ποσό που πρέπει να πληρωθεί από το χρήστη και αυτό καταβάλλεται ανάλογα με το μηχανισμό πληρωμής που χρησιμοποιείται.



Σχήμα 3.3 «Οικονομία Grid»

Συνοψίζοντας:

Ένας Πάροχος Υπηρεσιών Πλέγματος GSPs έχει επαφή με:

- Το Grid Market Directory: (GMD), όπου ανακοινώνουν τις υπηρεσίες που προσφέρει και οι τιμές τους με σκοπό να προσελκύσει καταναλωτές.
- Τον Grid Trade Server (GTS), που τον εκπροσωπεί κατά τις διαπραγματεύσεις για την πώληση υπηρεσιών. Στοχεύει στην μέγιστη χρησιμοποίηση των πόρων του παρόχου και στην μεγιστοποίηση των κερδών του. Μπορεί να

χρησιμοποιήσει διάφορα οικονομικά μοντέλα (αναλύονται σε επόμενη ενότητα) για την πώληση των πόρων.

- Τις Πολιτικές τιμολόγησης (Pricing Policies), που καθορίζουν τις τιμές που οι πάροχοι θέλουν να χρεώσουν τους χρήστες. Μπορεί να χρησιμοποιηθούν διάφορες πολιτικές οι οποίες μπορεί να ποικίλλουν από χρήστη σε χρήστη ή με το χρόνο. Μπορεί ακόμα να κατευθύνονται από την προσφορά-και-ζήτηση πόρων όπως σε μια πραγματική οικονομία.
- Τους μηχανισμούς λογαριασμών και χρέωσης (Resource Accounting and Charging components), που είναι υπεύθυνοι για την καταγραφή χρήσης πόρων και την αντίστοιχη χρέωση που έχει συμφωνηθεί μεταξύ Grid Resource Brokers και Grid Trade Server.

Ένας Καταναλωτής πόρων πλέγματος - Grid Resource Consumer έχει επαφή με:

- Τον μεσάζοντά του Grid Resource Broker, στον οποίο διατυπώνει τις απαιτήσεις των εφαρμογών του, την ποιότητα υπηρεσιών που επιθυμεί, το ποσό που είναι διαθέσιμος να διαθέσει, το χρονικό περιθώριο μέσα στο οποίο θέλει να έχει ολοκληρωθεί η εργασία του.
- Τον Μηχανισμό πληρωμής - Payment Mechanism που χρησιμοποιείται για να καταβάλει το ποσό που χρεώθηκε.

3.5 Οικονομικά μοντέλα διαχείρισης πόρων πλέγματος

Όπως στη συμβατική αγορά, η κοινότητα των καταναλωτών αντιπροσωπεύει την ζήτηση ενώ η κοινότητα των παρόχων των πόρων αντιπροσωπεύει την προσφορά. Οι πάροχοι GSPs με τη χρήση συστημάτων λογισμικού διαθέτουν τους πόρους τους και μαζί με τη χρήση Grid Trading Services (GTS) κάνουν δυνατή την εμπορική εκμετάλλευσή τους ικανοποιώντας την εκτέλεση αιτημάτων που κατευθύνονται από

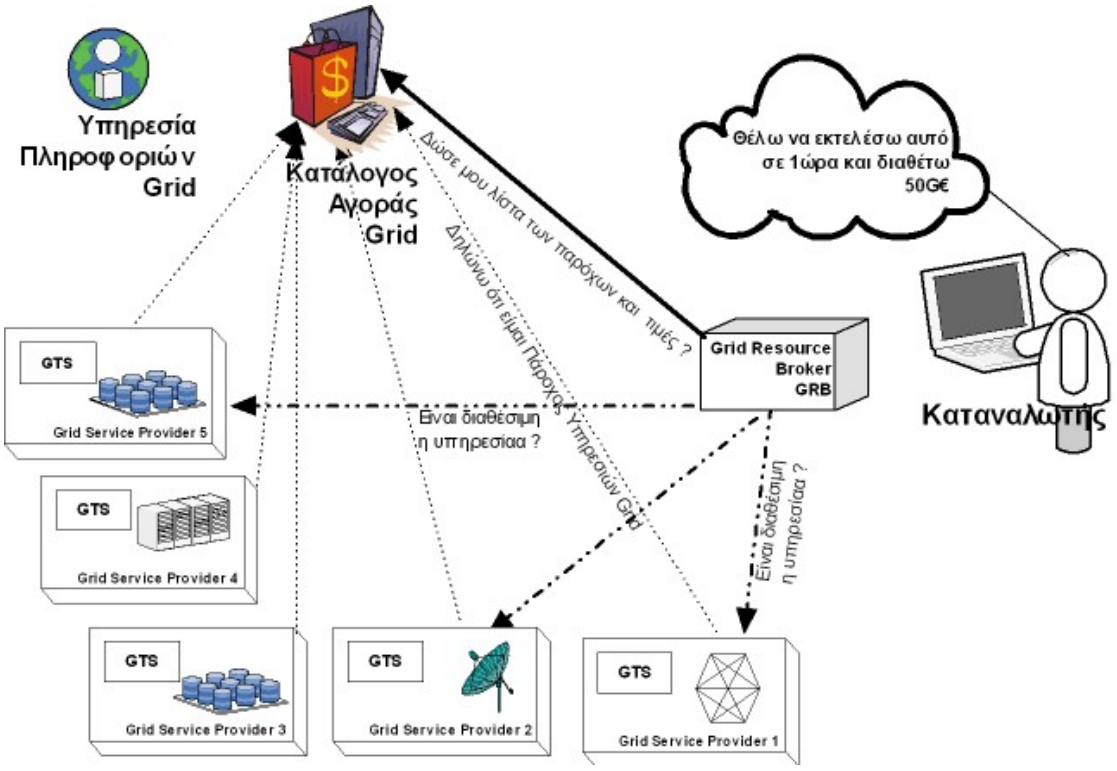
τους μεσίτες (GRBs). Στην αλληλεπίδραση μεταξύ GRBs και GSPs κατά τη διάρκεια του εμπορίου πόρων μεσολαβεί ένα Grid Market Directory (GMD) Κατάλογος αγοράς πλέγματος (GMD) . Χρησιμοποιούνται διάφορα οικονομικά μοντέλα ή πρωτόκολλα συναλλαγής που αποφασίζουν για την τιμή πρόσβασης της υπηρεσίας.

Υπάρχουν πολυάριθμα οικονομικά μοντέλα για την διαχείριση πόρων πλέγματος, που περιλαμβάνουν μικροοικονομικές και μακροοικονομικές αρχές δανεισμένες από την πραγματική αγορά.

- Μοντέλο αγοράς προϊόντων (Commodity Market Model)
- Μοντέλο ανακοινώσιμων τιμών (Posted Price Model)
- Μοντέλο διαπραγματεύσεων (Bargaining Model)
- Μοντέλο υποβολή προσφορών/υπογραφής συμβάσεων (Tendering/Contract-Net Model)
- Μοντέλο δημοπρασίας (Auction Model)
- Μοντέλο αναλογικής απόδοσης πόρων βασισμένο σε προσφορές (Bid-based Proportional Resource Sharing Model)
- Μοντέλο Συνεταιριστικής ανταλλαγής (Cooperative Bartering Model)
- Μονοπώλιο και ολιγοπώλιο (Monopoly and Oligopoly)

3.5.1 Μοντέλο αγοράς προϊόντων (Σταθερή τιμολόγηση ή προσαρμοσμένη στην προσφορά/ζήτηση)

Στο μοντέλο αγοράς προϊόντων, οι ιδιοκτήτες των πόρων διευκρινίζουν την τιμή της υπηρεσίας που προσφέρουν και χρεώνουν τους χρήστες σύμφωνα με το ποσό πόρων που καταναλώνουν. Η πολιτική τιμολόγησης μπορεί να καθοριστεί από διάφορες παραμέτρους και μπορεί να είναι σταθερή ή μεταβλητή ανάλογα με την προσφορά και τη ζήτηση των πόρων. Γενικά, οι υπηρεσίες διατιμώνται κατά τέτοιο τρόπο ώστε να διατηρείται η ισορροπία προσφοράς και ζήτησης.



Σχήμα 3.4 Μοντέλο αγοράς προϊόντων

Στο μοντέλο σταθερής τιμολόγησης, μόλις καθοριστεί η τιμή των πόρων για μια ορισμένη περίοδο, παραμένει ίδια ανεξάρτητα από την ποιότητα υπηρεσιών. Δεν επηρεάζεται σημαντικά από την ζήτηση.

Σε ένα μοντέλο προσφοράς και ζήτησης, οι τιμές αλλάζουν πολύ συχνά βασισμένες στις αλλαγές προσφοράς και ζήτησης.

Σε γενικές γραμμές, όταν αυξάνεται η ζήτηση ή μειώνεται η προσφορά, οι τιμές αυξάνονται έως ότου υπάρχει η ισοζύγιο μεταξύ προσφοράς - ζήτησης. Τα σενάρια τιμολόγησης σε ένα Μοντέλο αγοράς προϊόντων μπορούν να βασιστούν σε:

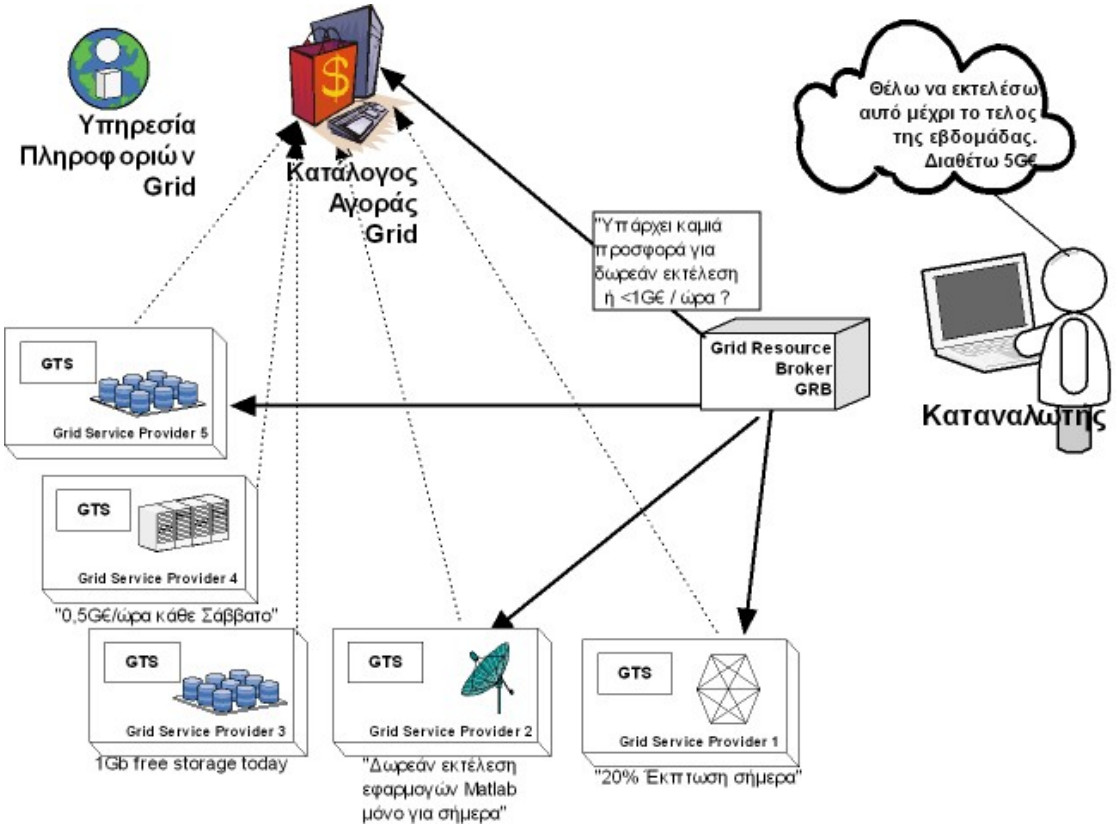
- Σταθερή χρέωση
- Διάρκεια χρήσης (χρόνος)
- Συνδρομή
- Βασισμένη στην προσφορά και ζήτηση

3.5.2 Μοντέλο ανακοινωμένων τιμών (Posted Price Model)

Το Μοντέλο ανακοινωμένων τιμών είναι παρόμοιο με το μοντέλο αγοράς προϊόντων, μόνο που σ'αυτό ο GSP δεν διαφημίζει ειδικές προσφορές με στόχο την προσέλκυση (νέων) καταναλωτών για να κατακτήσει μερίδιο της αγοράς ή για να παρακινήσει τους χρήστες να χρησιμοποιήσουν φτηνότερες χρονοσχισμές για τις εργασίες τους.

Σε αυτήν την περίπτωση, οι μεσίτες δεν χρειάζεται να διαπραγματευτούν άμεσα με τους GSPs για την τιμή, αλλά να χρησιμοποιήσουν τις ήδη ανακοινωμένες προσφορές για υπηρεσίες δεδομένου ότι είναι γενικά φτηνότερες έναντι των κανονικών τιμών. Οι προσφορές σίγουρα θα έχουν περιορισμούς χρήσης, αλλά μπορούν να είναι ελκυστικές για μερικούς χρήστες.

Παραδείγματος χάριν, κατά τη διάρκεια των περιόδων διακοπών, η απαίτηση για πόρους είναι πιθανό να περιοριστεί και οι GSPs μπορούν να ανακοινώσουν ελκυστικές προσφορές υπηρεσιών στοχεύοντας στο να προσελκύσουν χρήστες και να αυξήσουν την χρησιμοποίηση των πόρων τους.



Σχήμα 3.5 Μοντέλο ανακοινωμένων τιμών

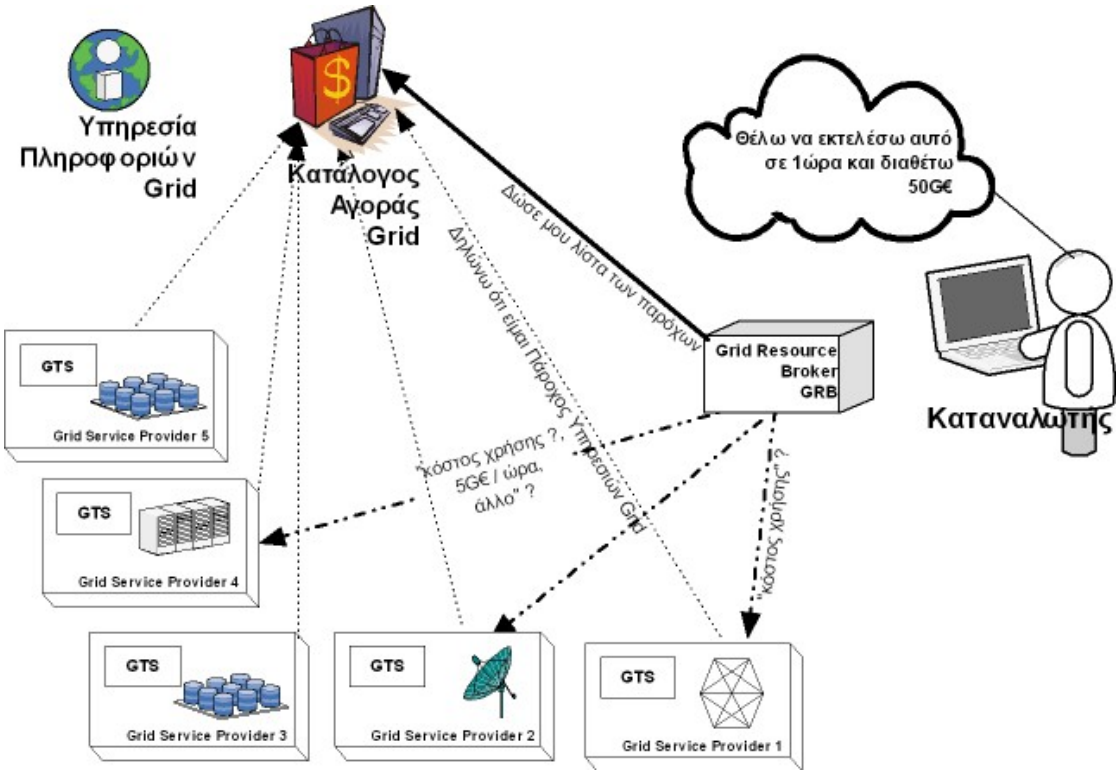
Οι δραστηριότητες που συσχετίζονται συγκεκριμένα με το πρότυπο ανακοινωμένων τιμών εκτός από εκείνους αφορούν το μοντέλο αγοράς προϊόντων είναι:

- Οι φορείς παροχής υπηρεσιών πλέγματος (GSPs) ανακοινώνουν τις ειδικές προσφορές και τους σχετικούς όρους τους κ.λπ. στον κατάλογο αγοράς πλέγματος, Grid Market Directory.
- Ο μεσίτης εξετάζει το GMD για να προσδιορίσει εάν οποιαδήποτε από τις υπάρχουσες προσφορές, ικανοποιεί τις απαιτήσεις του.
- Ο μεσίτης ερευνά (GSP) για τη διαθεσιμότητα των ανακοινωμένων υπηρεσιών. (δεν ανακοινώνεται πότε μια προσφορά υπηρεσιών αγοράστηκε και δεσμεύτηκε από κάποιον μεσίτη)
- Τα υπόλοιπα βήματα είναι παρόμοια με εκείνα που επισημαίνονται στο πρότυπο αγοράς προϊόντων.

3.5.3 Μοντέλο – διαπραγματεύσεων (Bargaining Model)

Στα προηγούμενα μοντέλα, οι μεσίτες καταβάλλουν τις τιμές πρόσβασης, οι οποίες καθορίζονται από τους GSPs. Στο μοντέλο διαπραγματεύσεων, οι μεσίτες των πόρων διαπραγματεύονται με τους GSPs για χαμηλότερη τιμή πρόσβασης και υψηλότερη διάρκεια χρήσης. Καθένας από τους μεσίτες, GSPs έχει τους δικούς του στόχους και διαπραγματεύονται έως ότου επιτευχθούν αυτοί. Οι μεσίτες συνήθως αρχίζουν προσφέροντας μια πολύ χαμηλή τιμή, ενώ οι GSPs με μια πολύ υψηλή. Και οι δύο διαπραγματεύονται έως ότου φθάσουν σε μια αμοιβαία αποδεκτή τιμή ή μέχρι ένας από τους να μην είναι πρόθυμος να διαπραγματευτεί περαιτέρω.

Η όλη διαπραγμάτευση καθοδηγείται από τις απαιτήσεις χρηστών (π.χ., χαλαρό περιθώριο χρόνου ολοκλήρωσης). Οι μεσίτες παίρνουν ρίσκο και διαπραγματεύονται για τις όσο το δυνατόν φτηνότερες τιμές και δύνανται να απορρίψουν ακριβούς GSPs. Κάτι τέτοιο μπορεί να οδηγήσει σε χαμηλότερη χρησιμοποίηση πόρων. Τότε οι GSPs προθυμοποιούνται να μειώσουν την τιμή αντί να διατηρούν τους πόρους τους αχρησιμοποίητους.



Σχήμα 3.6 Μοντέλο διαπραγματεύσεων

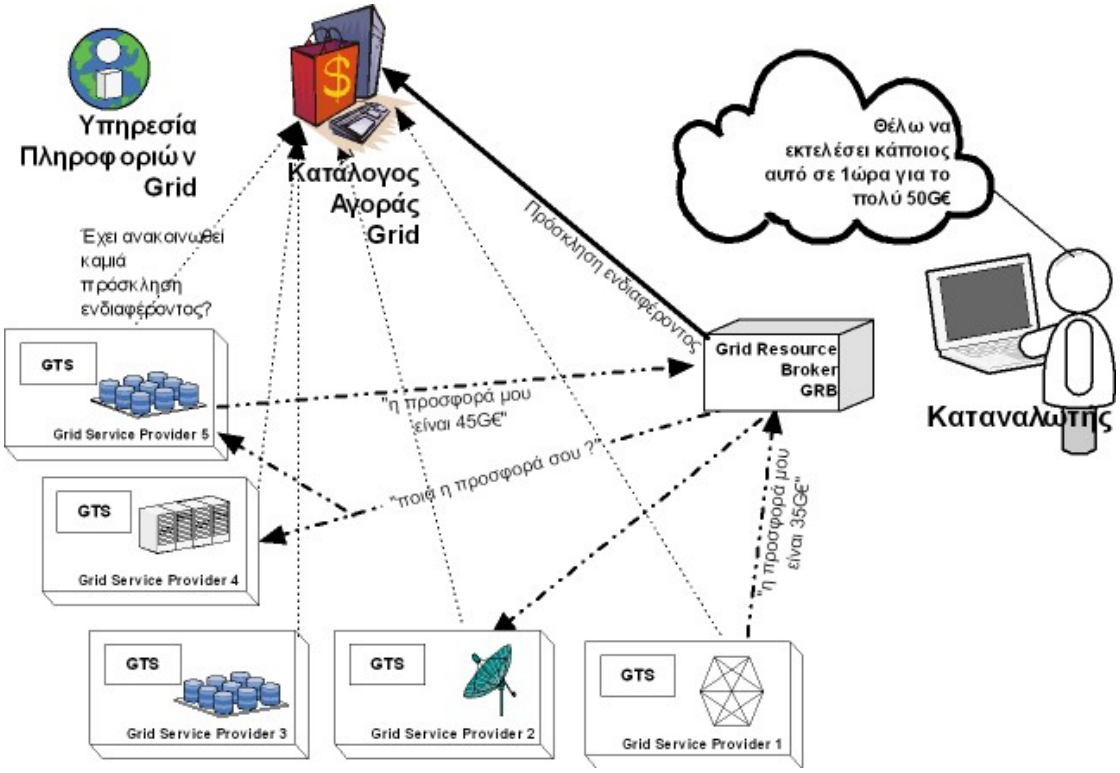
Οι μεσίτες και οι GSPs χρησιμοποιούν αυτό το πρότυπο όταν δεν προσδιορίζονται σαφώς οι τιμές των υπηρεσιών και η προσφορά-ζήτηση της αγοράς. Οι χρήστες μπορούν να διαπραγματευτούν για μια χαμηλότερη τιμή με το σκεπτικό δημιουργίας καλών σχέσεων για μελλοντικές συνεργασίες

3.5.4 Μοντέλο αποδοχής προσφοράς / σύμβασης (Tender/Contract-Net Model)

Το Tender/Contract-Net model είναι ένα από τα ευρύτατα χρησιμοποιημένα πρότυπα για τη διαπραγμάτευση υπηρεσιών σε διανεμημένη περιβάλλοντα επίλυσης προβλημάτων.

Βασίζεται στον μηχανισμό υλοποίησης συμβάσεων που χρησιμοποιείται από τις επιχειρήσεις για να ελέγχουν αυτόματα την ανταλλαγή αγαθών και υπηρεσιών. Βοηθά στην εύρεση ενός αρμόδιου πάροχου υπηρεσιών που θα απασχοληθεί σε έναν δεδομένο στόχο.

Ο χρήστης/μεσίτης που ζητά την εκπλήρωση ενός αιτήματός του αποκαλείται Διαχειριστής ενώ ο πάροχος του πόρου που μπορεί να ικανοποιήσει το αίτημα λέγεται Πιθανός Ανάδοχος



Σχήμα 3.7 Μοντέλο αποδοχής προσφοράς / σύμβασης

Από την προοπτική του διαχειριστή, η διαδικασία είναι:

- Ο μεσίτης Consumer (Broker) ανακοινώνει τις απαιτήσεις του (χρησιμοποιώντας ένα πρότυπο συναλλαγής) και καλεί τους GSPs να δώσουν τις προσφορές τους
- Οι ενδιαφερόμενοι GSPs αξιολογούν την αγγελία και αποκρίνονται υποβάλλοντας τις προσφορές τους.
- Ο Broker αξιολογεί και αναθέτει τη σύμβαση στον πιο κατάλληλο GSP.
- Ο μεσίτης και ο GSP επικοινωνούν ιδιαιτέρως και χρησιμοποιείται ο πόρος.

Το περιεχόμενο του προτύπου διαπραγμάτευσης που χρησιμοποιείται για την αναγγελία της εργασίας περιλαμβάνει, τον παραλήπτη (χρήστης), προδιαγραφές απαιτήσεων επιλεξιμότητας (π.χ., Linux, x86, 1024MB μνήμη), παρουσίαση

έργου/υπηρεσιών, προαιρετική τιμή που ο χρήστης είναι πρόθυμος να επενδύσει, προδιαγραφές της προσφοράς (τι πρέπει να περιλαμβάνει η προσφορά), προθεσμία αποδοχής προσφορών.

Από την προοπτική ενός αναδόχου/GSP, η διαδικασία είναι:

- Λήψη αναγγελιών/αιτημάτων για παροχή υπηρεσιών (έστω μέσω του GMD).
- Αξιολόγηση την δυνατότητας παροχής της αιτούμενης υπηρεσίας.
- Απόκριση με την κατάθεση προσφοράς.
- Παράδοση της υπηρεσίας εάν η προσφορά γίνει αποδεκτή.
- Έκθεση των αποτελεσμάτων και χρέωση του μεσίτη/χρήστη σύμφωνα με την συμφωνηθείσα προσφορά.

Το πλεονέκτημα αυτού του μοντέλου είναι ότι αν αποτύχει ο επιλεγμένος GSP να παρέχει ικανοποιητικές υπηρεσίες, ο μεσίτης μπορεί να αναζητήσει άλλους GSPs.

Μειονέκτημα όμως είναι το ότι μια εργασία μπορεί να αποδοθεί σε ένα λιγότερο ικανό πάροχο GSP αν ένας πιο ικανός GSP είναι απασχολημένος την ώρα της αποδοχής σύμβασης.

Άλλος ένας περιορισμός είναι ότι ο διαχειριστής GRB δεν έχει υποχρέωση να ενημερώσει τους πιθανούς αναδόχους ότι μια συμφωνία πραγματοποιήθηκε.

Μερικές φορές ένας διαχειριστής ίσως δεν λάβει προσφορές για διαφόρους λόγους, όπως:

- Οι πιθανοί ανάδοχοι είναι απασχολημένοι με άλλες εργασίες.
- Ένας πιθανός ανάδοχος είναι ενεργός αλλά κατατάσσει το συγκεκριμένο αίτημα κάτω από άλλα που έχει υπόψιν του.
- Κανένας ανάδοχος, ακόμα και οι ενεργοί δεν είναι σε θέση να προσφέρει την υπηρεσία που ζητείται.

Το μοντέλο συμβάσεων μπορεί να υλοποιηθεί και χωρίς διαδικασία διαπραγμάτευσης, όπου ο επιλεγμένος GSP αποκρίνεται με αποδοχή ή άρνηση στο αίτημα. Αυτή η ικανότητα μπορεί να απλοποιήσει την διαδικασία και να βελτιώσει την αποδοτικότητα ορισμένων υπηρεσιών.

3.5.5 Auction Model - Μοντέλο δημοπρασίας

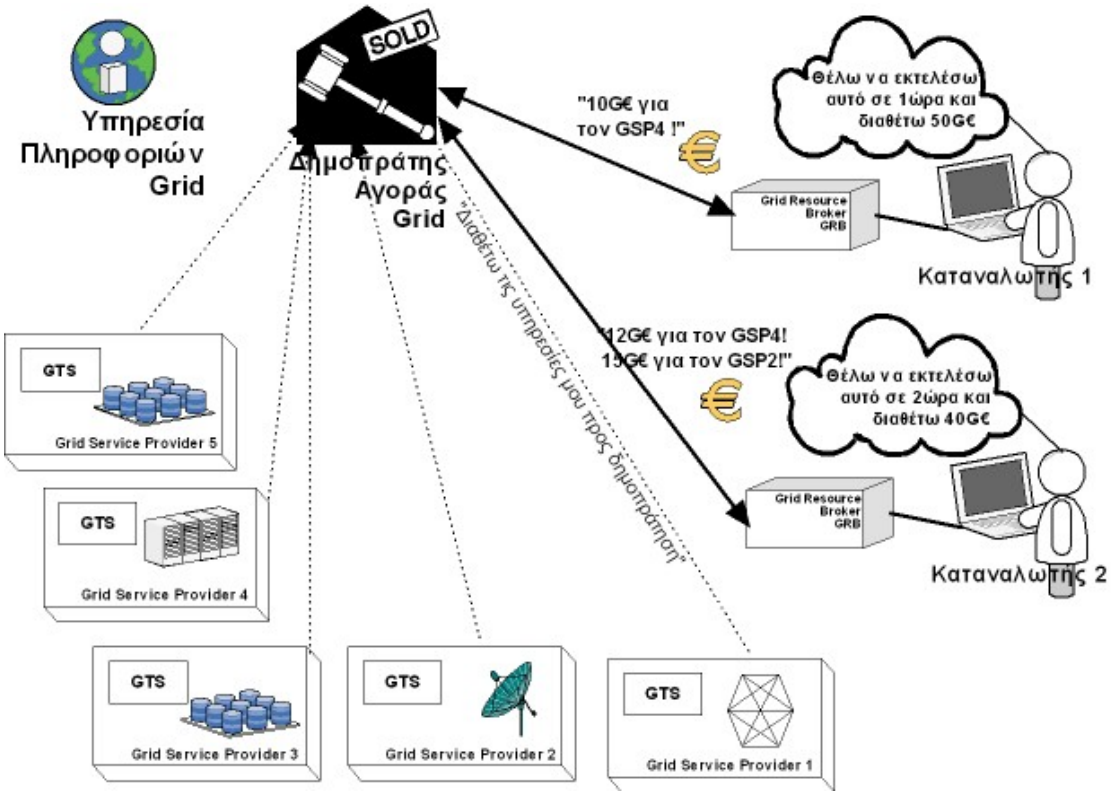
Το μοντέλο δημοπρασίας [37] υποστηρίζει διαπραγμάτευση ενός-με-πολλούς, δηλαδή μεταξύ του παρόχου υπηρεσιών (πωλητής) και πολλών καταναλωτών (αγοραστές), και περιορίζει τη διαπραγμάτευση σε μια απλή μεταβλητή (π.χ. αξία). Ο δημοπράτης θέτει τους κανόνες της δημοπρασίας, οι οποίοι είναι αποδεκτοί από καταναλωτές και προμηθευτές. Οι δημοπρασίες χρησιμοποιούν τις δυνάμεις αγοράς για να διαπραγματευτούν την αξία μιας υπηρεσίας.

Στον πραγματικό κόσμο, οι δημοπρασίες χρησιμοποιούνται εκτενώς, ιδιαίτερα για την πώληση αγαθών/στοιχείων μέσα σε μια καθορισμένη διάρκεια. Οι τρεις βασικοί φορείς που συμμετέχουν στις δημοπρασίες είναι: ιδιοκτήτες των πόρων, δημοπράτες (μεσολαβητές), και αγοραστές. Πολλές πύλες ηλεκτρονικού εμπορίου όπως το Amazon.com και το eBay.com λειτουργούν ως μεσολαβητές (δημοπράτες). Οι ρόλοι τόσο των αγοραστών όσο και των πωλητών μπορούν να αυτοματοποιηθούν.

Τα βήματα που περιλαμβάνονται στη διαδικασία μιας δημοπρασίας είναι:

- Οι GSPs αναγγέλλουν τις υπηρεσίες τους και καλούν για προσφορές.
- Οι μεσίτες δίνουν τις προσφορές τους (και μπορούν να δουν τι προσφέρουν άλλοι καταναλωτές εάν η δημοπρασία είναι ανοιχτή).
- Συνεχίζονται οι προσφορές από τους μεσίτες μέχρι να μην είναι κανείς άλλος διαθέσιμος να δώσει προσφορά ή ο δημοπράτης σταματά αν δεν έχει ικανοποιηθεί η τιμή ελάχιστης προσφοράς.
- Ο GSP προσφέρει την υπηρεσία σε αυτόν που κερδίζει.
- Ο καταναλωτής χρησιμοποιεί τον πόρο.

Σε μια δημοπρασία η οποία εκτείνεται στα όρια του Διαδικτύου, είναι άκρως δελεαστική η αυτοματοποίηση της διαδικασίας, όπου όλα τα συμβαλλόμενα μέρη μπορούν να καθορίσουν τις στρατηγικές για τους πράκτορες τους και αυτοί να μπορούν να συμμετέχουν σε πολλαπλάσιες δημοπρασίες για να επιτύχουν βελτιστοποίηση των αντικειμενικών τους λειτουργιών.



Σχήμα 3.8 Μοντέλο δημοπρασίας

Οι δημοπρασίες μπορούν να ταξινομηθούν σε τέσσερις τύπους:

- Αγγλική δημοπρασία - English Auction (first-price open cry)
- Δημοπρασία μεγαλύτερης σφραγισμένης προσφοράς - First-price sealed-bid auction
- Δημοπρασία Vickrey - Vickrey (Second-price sealed-bid) auction
- Ολλανδική δημοπρασία – Dutch Auction
- Διπλή δημοπρασία - Double Auction (Continuous)

Όλες οι ανωτέρω δημοπρασίες διαφέρουν από την άποψη του εάν εκτελούνται ως ανοικτές ή κλειστές δημοπρασίες και το τι θα πληρώσει ο υψηλότερος πλειοδότης. Στις ανοικτές δημοπρασίες, οι πράκτορες που υποβάλλουν προσφορές μπορούν να ξέρουν την αξία των προσφορών άλλων πρακτόρων. Τους δίνεται έτσι η ευκαιρία να προσφέρουν ανταγωνιστικές προσφορές. Στις κλειστές δημοπρασίες, οι προσφορές των συμμετεχόντων δεν αποκαλύπτονται

3.5.5.1 Αγγλική δημοπρασία - English Auction (first-price open cry)

Όλοι οι ενδιαφερόμενοι είναι ελεύθεροι να αυξήσουν τις προσφορές τους και να υπερβούν τις προσφορές άλλων. Όταν κανένας από τους πλειοδότες δεν είναι πρόθυμος να αυξήσει την τιμή άλλο, η δημοπρασία τελειώνει και ο υψηλότερος πλειοδότης κερδίζει την υπηρεσία στην τιμή της προσφοράς του.

Σε ένα τέτοιο μοντέλο, το βασικό ζήτημα είναι πώς GRBs αποφασίζουν το ποσό της προσφοράς τους. Ο GRB έχει ένα κρυφό ποσό (καθορίζεται από το χρήστη) και αναπτύσσει μια στρατηγική για το πως θα κάνει μια σειρά προσφορών η οποία προκύπτει συναρτήσει του κρυφού ποσού που διαθέτει, των προβλέψεων για το πως θα κινηθούν οι προσφορές άλλων αγοραστών και του ιστορικού παλαιότερων προσφορών.

Ο GRB αποφασίζει το κρυφό ποσό που θα διαθέσει βασισμένος στις απαιτήσεις του χρήστη (χρονική προθεσμία και χρήματα που προθυμοποιείται να επενδύσει για την επίλυση του προβλήματος).

Στις private value English auctions, η κυρίαρχη στρατηγική του GRB είναι να προσφερθεί πάντα ένα μικρό ποσό "υψηλότερο" από την τρέχουσα υψηλότερη προσφορά, και να σταματά όταν φτάνει στο κρυφό ποσό που έχει διαθέσιμο.

Στις correlated value auctions, δημοπρασίες κλιμακωτής αξίας, οι πολιτικές είναι διαφορετικές και επιτρέπουν στον δημοπράτη να αυξάνει την τιμή κατά ένα σταθερό ποσοστό ή στο ποσοστό που επιθυμεί.

Αυτοί που δεν ενδιαφέρονται να δώσουν άλλες προσφορές μπορούν να το δηλώσουν ανοιχτά (open-exit) χωρίς τη δυνατότητα επανεισόδου. Οι πληροφορίες αυτές βοηθούν τους άλλους ενδιαφερόμενους να προσαρμόσουν τις εκτιμήσεις τους και να ρυθμίσουν αντίστοιχα το ύψος των προσφορών τους.

3.5.5.2 Δημοπρασία μεγαλύτερης σφραγισμένης προσφοράς - First-price sealed-bid auction

Κάθε ενδιαφερόμενος υποβάλλει μια προσφορά χωρίς γνώση των άλλων προσφορών.

Ο υψηλότερος πλειοδότης κερδίζει τη δημοπρασία στην τιμή της προσφοράς του. Σε αυτήν την περίπτωση η στρατηγική που ακολουθεί ο μεσίτης είναι συνάρτηση του κρυφού διαθέσιμου ποσού και των προβλέψεων για τις εκτιμήσεις των άλλων ενδιαφερομένων.

Η βέλτιστη στρατηγική είναι να προσφέρει κανείς λιγότερο από την πραγματική αξία και ίσως κερδίσει. Πάραυτα όλα εξαρτώνται από τις προσφορές των άλλων ενδιαφερομένων.

3.5.5.3 Δημοπρασία Vickrey - Vickrey (Second-price sealed-bid) auction

Κάθε πλειοδότης υποβάλλει μια προσφορά χωρίς γνώση των άλλων προσφορών. . Ο υψηλότερος πλειοδότης κερδίζει την υπηρεσία στην τιμή που προσφέρει ο δεύτερος υψηλότερος πλειοδότης. Η αρχιτεκτονική και οι στρατηγικές εφαρμογής είναι παρόμοιες με το ContractNet/Tender μοντέλο που παρουσιάστηκε παραπάνω.

3.5.5.4 Ολλανδική δημοπρασία – Dutch Auction

Ο δημοπράτης ξεκινά την δημοπρασία με μια υψηλή τιμή και την μειώνει σταδιακά έως ότου ένας ενδιαφερόμενος αγοράζει στην τρέχουσα τιμή. Το μοντέλο είναι παρόμοιο με τη δημοπρασία first-price sealed-bid auction επειδή και στις δύο περιπτώσεις η προσφορά μετράει μόνο εάν είναι η υψηλότερη, και δεν αποκαλύπτεται καμία σχετική πληροφορία κατά τη διάρκεια της διαδικασίας.

Από πλευράς στρατηγικής του μεσίτη, η ολλανδική δημοπρασία είναι παρόμοια με την English (first-price sealed-bid auction).

Η βασική διαφορά μεταξύ τους είναι ότι στην αγγλική η δημοπρασία ανοίγει με μικρή τιμή και σταδιακά αυτή αυξάνεται έως ότου πέσει η ζήτηση ενώ, στην ολλανδική δημοπρασία ανοίγει ψηλά με μεγάλη τιμή και μειώνεται σταδιακά έως ότου αυξηθεί η ζήτηση και γίνει εφάμιλλη της προσφοράς.

Τα πρωτόκολλα για μια ολλανδική δημοπρασία εξελίσσονται ως εξής:

η δημοπρασία προσπαθεί να βρει την αγοραστική αξία ενός αγαθού/υπηρεσίας ξεκινώντας από μια τιμή πολύ υψηλότερη από την αναμενόμενη αγοραστική αξία, έπειτα σταδιακά μειώνεται η τιμή μέχρι ένας από τους αγοραστές να αποδεχθεί την τιμή. Το ποσοστό μείωσης της τιμής εξαρτάται από τον δημοπράτη και ίσως έχουν μια ελάχιστη τιμή κάτω από την οποία δεν συνεχίζουν. Αν η τιμή πέσει στο όριο αυτό χωρίς να υπάρχουν αγοραστές η δημοπρασία τερματίζεται.

Από άποψη πραγματικού χρόνου, η ολλανδική δημοπρασία είναι αποδοτικότερη αφού ο δημοπράτης μπορεί να μειώσει την τιμή κατά ένα στρατηγικό/σημαντικό ποσοστό και ο πρώτος πλειοδότης να κερδίσει.

3.5.5.5 Διπλή δημοπρασία - Double Auction (Continuous)

Είναι ένα από τα πιο διαδεδομένους θεσμούς ανταλλαγής στην αγορά, οι ρίζες του οποίου ξεκινούν από την αρχαία Αίγυπτο και τη Μεσοποταμία. Στην πραγματικότητα, είναι το πρωτεύον οικονομικό μοντέλο για τις εμπορικές συναλλαγές δικαιοσύνης, προϊόντων, και παραγώγων στις χρηματιστηριακές αγορές (π.χ., NASDAQ).

Στο διπλό μοντέλο δημοπρασίας double auction model, διαταγές αγορών (προσφορές) και διαταγές πώλησης (αιτήματα) μπορούν να υποβληθούν οποιαδήποτε στιγμή κατά τη διάρκεια της περιόδου των συναλλαγών.

Εάν οποιαδήποτε στιγμή υπάρχουν ανοικτές προσφορές και αιτήματα που βρίσκονται σε αντιστοιχία ή είναι συμβατά από την άποψη της τιμή και των απαιτήσεων (π.χ., ποσότητα αγαθών ή μετοχών), μια συναλλαγή εκτελείται αμέσως.

Σε αυτήν την δημοπρασία οι διαταγές ταξινομούνται από την υψηλότερη στις χαμηλότερες για να προκύψουν τα σχεδιαγράμματα προσφοράς και ζήτησης. Από τα σχεδιαγράμματα, η μέγιστη ποσότητα που ανταλλάσσεται μπορεί να καθοριστεί με το ταίριασμα αναγγελιών πώλησης (αρχίζοντας από τη χαμηλότερη τιμή και ανεβαίνοντας) με τις προσφορές για αγορά (αρχίζοντας από την υψηλότερη τιμή και κατεβαίνοντας).

Έχουν αναπτυχθεί μηχανισμοί πρακτόρων βασισμένοι σε λογισμικό για να αυτοματοποιήσουν τη διπλή δημοπρασία, για εμπορικές συναλλαγές αποθεμάτων με ή χωρίς ανθρώπινη παρέμβαση.

Το μοντέλο διπλής δημοπρασίας έχει πολύτιμα χαρακτηριστικά για χρήση σε υπολογιστικά πλέγματα.

Οι μεσίτες μπορούν εύκολα να δημοσιεύουν αιτήματα ανάλογα με τα διαθέσιμα χρήματα, την προθεσμία, την πολυπλοκότητα της εργασίας, την στρατηγική χρονοπρογραμματισμού, και τα χαρακτηριστικά των πόρων. Οι GSPs μπορούν να δημοσιεύσουν αναγγελίες πώλησης βασισμένες στο τρέχον φορτίο, και στην τρέχουσα ζήτηση καθώς και τους περιορισμούς στις τιμές.

Αυτά μπορούν να υποβληθούν στους πράκτορες της αγοράς πλέγματος GMD – Grid Market Directory που παρέχουν συνεχή εκκαθάριση ή αντιστοίχηση υπηρεσιών. Δεδομένου ότι οι προσφορές εκκαθαρίζονται συνεχώς, τόσο οι GRBs όσο και οι GSPs μπορούν να λάβουν στιγμιαίες αποφάσεις με λιγότερο υπολογιστικό κόστος και πολυπλοκότητα.

Στις δημοπρασίες μπορεί να εμφανιστούν ανεπιθύμητα φαινόμενα όπως:

- συνεργίας (εάν οι πλειοδότες συντονίζουν τις τιμές προσφοράς τους έτσι ώστε οι προσφορές να παραμένουν τεχνητά χαμηλές)
- παραπλανητικοί δημοπράτες στην περίπτωση δημοπρασίας vickey
- παραπλανητικοί πλειοδότες,
- αντίθετες με τη λογική προσφορές με στόχο την κερδοσκοπία, κ.λπ.

3.5.6 Βασισμένο σε προσφορές μοντέλο καταμερισμού πόρων - Bid-based Proportional Resource Sharing Model

Τα βασισμένα στην αγορά αναλογικά συστήματα διαχείρισης πόρων, είναι αρκετά δημοφιλή στα συνεργατικά περιβάλλοντα επίλυσης προβλημάτων όπως τις συστάδες υπολογιστών (clusters (in single administrative domain)).

Σε αυτό το μοντέλο, το ποσοστό του μεριδίου των πόρων που διατίθεται στην εφαρμογή ενός χρήστη, είναι ανάλογο με την αξία της προσφοράς του σε σύγκριση με τις προσφορές άλλων χρηστών.

Στους χρήστες διατίθενται πιστωτικές μονάδες ή κουπόνια, τα οποία μπορούν να χρησιμοποιήσουν για να έχουν πρόσβαση στους πόρους. Η αξία κάθε πιστωτικής μονάδας εξαρτάται από τη ζήτηση των πόρων και την αξία που άλλοι χρήστες αποδίδουν στον πόρο κατά την διάρκεια της χρήσης αυτού.

Π.χ. δυο χρήστες επιθυμούν την πρόσβαση σε ένα πόρο με παρόμοιες απαιτήσεις. Ο χρήστης Α είναι πρόθυμος να ξοδέψει 2 κουπόνια ενώ ο Β 4 κουπόνια.

Σε αυτή την περίπτωση ο Α χρήστης παίρνει το $\frac{2}{4+2} = \frac{1}{3}$ ενώ ο Β χρήστης παίρνει τα $\frac{4}{4+2} = \frac{2}{3}$ του διαθέσιμου πόρου, που είναι ανάλογο στην αξία που και οι δυο χρήστες απέδωσαν στον πόρο για την εκτέλεση της εφαρμογής τους.

Αυτό μπορεί να είναι ένας καλός τρόπος για τη διαχείριση μιας κοινής δεξαμενής πόρων μέσα σε μια οργάνωση ή όταν οι πόροι ανήκουν σε πολλαπλές οντότητες (όπως τα τμήματα ενός πανεπιστήμιου), όπου μπορούν να έχουν κατανομή των πιστωτικών τους μονάδων ανάλογα με την επένδυση που έκαναν. Μπορούν να διευκρινίσουν το πόσο των πιστωτικών τους μονάδων που είναι πρόθυμοι να διαθέσουν, για την εκτέλεση των εργασιών τους.

Παραδείγματος χάριν, ένας χρήστης μπορεί καθορίσει μικρό αριθμό πιστωτικών μονάδων για τις μη-διαλογικές εργασίες batch και τις υψηλές πιστώσεις για τις διαλογικές εργασίες με υψηλούς χρόνους απόκρισης.

Σε ένα περιβάλλον κοινών πόρων, οι GSPs μπορούν να χρησιμοποιήσουν αυτό το πρότυπο για την προσφορά μιας QoS υπηρεσίας στους πελάτες που πληρώνουν περισσότερο.

3.5.7 Συνεταιριστικό πρότυπο ανταλλαγής - Cooperative Bartering Model

Μια κοινότητα οντοτήτων μοιράζεται η μια με την άλλη πόρους για να δημιουργήσουν ένα συνεργατικό υπολογιστικό περιβάλλον. Εκείνοι που συμβάλλουν τους πόρους τους σε μια κοινή δεξαμενή μπορούν να πάρουν πρόσβαση σε όλους τους πόρους της δεξαμενής αυτής.

Ένα εξελιγμένο μοντέλο μπορεί να εφαρμοστεί για να αποφασίζει τι μερίδιο των πόρων μπορούν οι συνεισφέροντες να χρησιμοποιήσουν. Ένας χρήστης μπορεί να κερδίζει πόντους όταν μοιράζεται τους πόρους του, τους οποίους μπορεί να χρησιμοποιήσει όταν θελήσει αυτός.

3.5.8 Μονοπώλιο/ολιγοπώλιο

Στα προηγουμένως αναφερθέντα πρότυπα έχουμε υποθέσει μια ανταγωνιστική αγορά όπου διάφοροι GSPs και οι μεσίτες/οι καταναλωτές καθορίζουν την τιμή αγοράς.

Εντούτοις, υπάρχουν οι περιπτώσεις όπου ένας μόνος GSP εξουσιάζει την αγορά και είναι ο μόνος πάροχος μιας συγκεκριμένης υπηρεσίας. Στην οικονομική θεωρία αυτό το μοντέλο είναι γνωστό ως μονοπώλιο.

Οι χρήστες δεν μπορούν να επηρεάσουν τις τιμές των υπηρεσιών και πρέπει να επιλέξουν την υπηρεσία στην τιμή που δίνεται από το μοναδικό GSP που μονοπωλεί την αγορά πλέγματος. Όσον αφορά την τεχνική υλοποίηση αυτού του προτύπου, ο GSP απλώς ενημερώνει το GMD για τις τιμές των υπηρεσιών του και οι μεσίτες ενημερώνονται χωρίς οποιαδήποτε δυνατότητα να διαπραγματευτούν οι τιμές.

Οι ανταγωνιστικές αγορές είναι το ένα άκρο και τα μονοπώλια είναι άλλο. Τις περισσότερες των περιπτώσεων, η κατάσταση αγοράς είναι ολιγοπώλιο, το οποίο είναι μεταξύ αυτών των δύο ακραίων περιπτώσεων: ένας μικρός αριθμός GSPs εξουσιάζει την αγορά και καθορίζει τις τιμές.

3.6 Τιμολόγηση, χρεώσεις και μηχανισμοί πληρωμής σε μια Grid – Οικονομία.

Σε ένα περιβάλλον πλέγματος οικονομικής φύσης, τόσο οι ιδιοκτήτες όσο και οι χρήστες των πόρων θέλουν να μεγιστοποιήσουν τα οφέλη τους.

Δεδομένου ότι μπορούν να υπάρξουν πολλοί GSPs που θα προσφέρουν παρόμοιες υπηρεσίες, πρέπει να έχουν μια ανταγωνιστική τιμολογιακή πολιτική προκειμένου: να προσελκύσουν χρήστες, να χρησιμοποιήσουν αποτελεσματικά τους πόρους τους, και να μεγιστοποιήσουν το κέρδος τους.

Οι πόροι που καταναλώνονται από τις εφαρμογές των χρηστών πρέπει να μετρώνται και να χρεώνονται. Θα πρέπει να προβλέπονται διάφοροι μηχανισμοί πληρωμής. Επιπλέον θα πρέπει οι χρήστες να μπορούν να αγοράσουν μονάδες πρόσβασης (access credits) στους πόρους είτε εκ των προτέρων ή μετά από τη χρήση τους.

Κάθε ένας GSP μπορεί να υλοποιήσει κάτι τέτοιο με τη χρησιμοποίηση συστημάτων όπως το QBank ή μπορεί να υπάρξει ένα παγκοσμίου εύρους Grid-wide bank -ευρέως αποκαλούμενο τράπεζα GridBank που μεσολαβεί για την πληρωμή των υπηρεσιών που χρησιμοποιούνται από το χρήστη.

3.6.1 Πώς καθορίζεται η τιμή ενός πόρου ?

Ένα απλό μοντέλο είναι το μοντέλο σταθερής τιμής, αλλά δεν καλύπτει τις περιπτώσεις που οι χρήστες απαιτούν συγκεκριμένη ποιότητα προσφερόμενων υπηρεσιών (QoS). Σε αυτήν την περίπτωση οι απαιτήσεις αλλάζουν σε σχέση με τις εφαρμογές αλλά και τη στιγμή που αυτές εκτελούνται, κάτι που οδηγεί στην ανάγκη ανάπτυξης έξυπνων πρακτόρων λογισμικού (software agents) οι οποίοι προσαρμόζονται στις αλλαγές των συνθηκών.

Τα σχέδια τιμολόγησης βασίζονται σε διάφορες παραμέτρους και μπορεί να είναι:

- Μοντέλο καθορισμένης αξίας (ίδιο κόστος για εφαρμογές και όχι QoS εξασφαλισμένη ποιότητα υπηρεσιών , όπως και στο σημερινό διαδίκτυο).
- Ανταγωνιστικά οικονομικά μοντέλα (π.χ., δημοπρασίες, συμβάσεις μετά από διαπραγμάτευση)
- Χρέωση ανάλογα με τη χρήση σε καθορισμένες ώρες της ημέρας (αιχμή, εκτός αιχμής, αργίες...)
- Χρέωση ανάλογα με την περίοδο και τη διάρκεια χρήσης (για μικρό/μεγάλο χρονικό διάστημα)
- Προσφορά και ζήτηση
- Βασισμένο σε προβλέψεις (μοντελοποίηση και πρόβλεψη των κινήσεων των ανταγωνιστών)
- Αφοσίωση των πελατών (επιβράβευση με πιστωτικές μονάδες όσων χρησιμοποιούν συχνά τις υπηρεσίες μιας εταιρίας)
- Ιστορικό δραστηριοτήτων
- Προσυμφωνημένο συμβόλαιο προσαρμοσμένο στις ανάγκες του πελάτη.
- Ημερολογιακή τιμολόγηση.
- Μαζική αγορά
- Καθορισμός των τιμών από κοινά αποδεκτούς οργανισμούς (παγκόσμιος φορέας αποδεκτός από όλους που ορίζει τις τιμές)
- Ικανότητα των πόρων όπως αξιολογείται στην κύρια αγορά
- Παροχή πόρων σε φθηνότερες τιμές για χρήση από την ακαδημαϊκή κοινότητα ή οργανισμούς κοινωνικής προσφοράς.

Θα μπορούσαμε να διαχωρίσουμε τους αγοραστές υπηρεσιών σε δύο κατηγορίες: quality-sensitive and price sensitive buyers αυτούς που επιζητούν την ποιότητα και αυτούς που θέλουν υπηρεσίες σε χαμηλή τιμή.

Η συλλογή οικονομικών στοιχείων δείχνει ότι στην περίπτωση που έχουμε χρήστες που αποζητούν την ποιότητα των προσφερόμενων υπηρεσιών, με τη χρήση συνδυαστικών μεθόδων και προσομοιώσεων μπορούμε να προβλέψουμε την πορεία της αγοράς. Όποια σενάρια τιμολόγησης και αν χρησιμοποιηθούν η αγορά σ'αυτή την περίπτωση θα φτάσει σε μια κατάσταση ισορροπίας των τιμών.

Αντίθετα όταν οι χρήστες επιζητούν πρωτίστως το χαμηλό κόστος, τότε οι περισσότερες τιμολογιακές πολιτικές οδηγούν σε ένα μεγάλο πόλεμο τιμών που δεν είναι προς όφελος της συνολικής λειτουργίας της αγοράς.

Η ιδανική λύση βρίσκεται κάπου στη μέση. Οι πολιτικές τιμολόγησης πρέπει να είναι τέτοιες ώστε: οι πάροχοι πόρων θα πρέπει να επωφελούνται από την αποτελεσματική χρήση των πόρων που παρέχουν ενώ οι καταναλωτές θα πρέπει να έχουν την δυνατότητα εξισορροπήσουν τις αιτήσεις τους μεταξύ κόστους και χρονικού πλαισίου.

3.6.2 Στοιχεία υπηρεσιών που καταγράφονται και χρεώνονται

Οι εφαρμογές των χρηστών έχουν διαφορετικές απαιτήσεις σε πόρους ανάλογα με τους υπολογισμούς που εκτελούνται και τους αλγόριθμους που χρησιμοποιούνται στην επίλυση προβλημάτων. Μερικές εφαρμογές απαιτούν εντατική χρήση ΚΜΕ , άλλες είναι εντατικές στη χρήση λειτουργιών εισόδου/εξόδου I/O ή μπορεί να είναι ένας συνδυασμός και των δύο. Το φαινόμενο αυτό πρέπει να προσμετράται κατά την χρέωση.

Επομένως, η κατανάλωση των ακόλουθων πόρων πρέπει να λογαριάζεται και να χρεώνεται:

- CPU - User time (consumed by user App.) and System time (consumed while serving user App.)
- Memory
- Maximum resident set size - page size

- Amount of memory used
- Page faults
- Storage used
- Network activity
- Signals received, context switches
- Software and Libraries accessed

Η πρόσβαση σε κάθε μια από αυτές τις οντότητες μπορεί να χρεωθεί χωριστά ή σε συνδυασμό με άλλες. Σχέδια συνδυασμένης τιμολόγησης θα πρέπει να έχουν μια μήτρα κοστολόγησης που θα παίρνει ως αίτημα πολλαπλούς πόρους για τιμολόγηση.

3.6.3 Μηχανισμοί πληρωμής

Μια αρχιτεκτονική πλέγματος βασισμένη σε οικονομικές αρχές θα πρέπει να υποστηρίζει διάφορους μηχανισμούς όπως:

- Προπληρωμένη – Πλήρωσε και χρησιμοποίησε (όπου οι χρήστες αγοράζουν μονάδες προκαταβολικά από τους GSPs ή από την Grid Bank)
- Χρησιμοποίησε και πλήρωσε αργότερα.
- Πλήρωνε ενώ χρησιμοποιείς
- Βασισμένη σε χορηγίες

Κάθε GSP μπορεί να χρεώσει τους χρήστες τους άμεσα και να χειριστεί όλα τα ζητήματα χρέωσης-πληρωμής από μόνος του. Αυτή η μέθοδος εισάγει ένα μεγάλο φορτίο τόσο για τους προμηθευτές και για τους χρήστες σε ένα μεγάλης κλίμακας περιβάλλον πλέγματος.

Αυτό μπορεί να απλοποιηθεί με την ύπαρξη μεσαζόντων όπως μιας εξελικτικής τράπεζας πλέγματος. Οι μεσίτες (brokers) που λειτουργούν για λογαριασμό των χρηστών μπορούν αυτόματα να ενημερώνουν τον πάροχο (GSPs) για τα στοιχεία του λογαριασμού του πελάτη στην τράπεζα πλέγματος (Grid Bank account details) και η χρέωση να γίνεται αυτόματα. Εναλλακτικά θα πρέπει οι χρήστες να μπορούν να χρησιμοποιήσουν άλλα ηλεκτρονικά συστήματα πληρωμών (electronic cash systems).

4

Προσομοιώσεις Διαχείρισης Πόρων σε Υπολογιστικά Πλέγματα με τη χρήση του GridSim[41] .

4.1 Εισαγωγή

Στο πραγματικό περιβάλλον μιας «Grid Οικονομίας» είναι πολύ δύσκολη η αξιολόγηση απόδοσης του μοντέλου που χρησιμοποιείται κατά τρόπο επαναλαμβανόμενο και ελεγχόμενο για διαφορετικά σενάρια. Αυτό συμβαίνει γιατί η διαθεσιμότητα των πόρων και του φορτίου σε ένα πλέγμα ποικίλλει συνεχώς από στιγμή σε στιγμή και είναι αδύνατο για ένα μεμονωμένο χρήστη μιας περιοχής να ελέγξει τις δραστηριότητες άλλων χρηστών σε άλλες διοικητικές περιοχές. Επιπλέον η πρόσβαση που έχει ένας χρήστης είναι πολύ περιορισμένη σε σύγκριση με το εύρος μιας παγκόσμιας Grid Οικονομίας αφού εκτείνεται σε περιορισμένους πόρους και περιοχές.

Προκειμένου λοιπόν να αποδειχθεί η αποτελεσματικότητα ενός οικονομικού μοντέλου διαχείρισης πόρων και των σχετικών αλγορίθμων διαχείρισης πόρων, η απόδοση πρέπει να αξιολογηθεί κάτω από διαφορετικά σενάρια όπως ποικιλία αριθμού πόρων με διαφορετικά χαρακτηριστικά, φορτίο και χρήστες με διαφορετικές απαιτήσεις (χρόνο ολοκλήρωσης των εργασιών τους, ποσό που διατίθενται να πληρώσουν).

Το πακέτο εργαλείων GridSim υποστηρίζει την μοντελοποίηση και προσομοίωση μεγάλου πλήθους ετερογενών πόρων: μηχανήματα ενός ή πολλών επεξεργαστών, κοινή και κατανεμημένης μνήμης όπως προσωπικούς υπολογιστές, τερματικά, συστάδες υπολογιστών. Η προσομοίωση μπορεί να περιλαμβάνει: πόρους που μπορεί να είναι γεωγραφικά διεσπαρμένοι και σε διαφορετικές διαχειριστικές περιοχές καθένας με τους δικούς του στόχους και κανόνες διαμοιρασμού. Χρήστες με διαφορετικές απαιτήσεις παρεχόμενης ποιότητας υπηρεσιών (QoS).

4.2 Άλλα εργαλεία προσομοίωσης

Προσομοιώσεις έχουν χρησιμοποιηθεί εκτενώς για τη διαμόρφωση και αξιολόγηση πραγματικών συστημάτων σε τομείς όπως, επιχειρησιακή ροή, εργοστασιακές κατασκευαστικές αλυσίδες, σχεδιασμός συστημάτων ηλεκτρονικών υπολογιστών. Σ'αυτό το διάστημα έχουν προκύψει για τις προσομοιώσεις σημαντικές αρχές, πρότυπα και έχουν αναπτυχθεί σχετικά εργαλεία και τεχνολογίες. Έτσι έχουμε γλώσσες προσομοίωσης (π.χ., Simscript), περιβάλλοντα προσομοίωσης (π.χ., parsec), βιβλιοθήκες προσομοίωσης (SimJava), και εφαρμογές προσομοίωσης (π.χ., προσομοιωτής δικτύων OMNet ++).

Εργαλεία που είναι διαθέσιμα για διεξαγωγή προσομοιώσεων σε υπολογιστικά περιβάλλοντα πλέγματος είναι τα: Bricks [38], MicroGrid [39], Simgrid [40], και το GridSim toolkit.

Η επιλογή του GridSim μεταξύ αυτών έγινε γιατί εστιάζει σε «Οικονομίες Grid» και περιλαμβάνει τις έννοιες των παραγωγών (ιδιοκτήτες των πόρων), των καταναλωτών (τελικοί χρήστες) και των μεσιτών που ανακαλύπτουν και διαθέτουν τους πόρους στους χρήστες.

4.3 GridSim

Το πακέτο εργαλείων GridSim παρέχει μια υποδομή για την προσομοίωση διαφορετικών κατηγοριών ετερογενών πόρων, χρηστών, εφαρμογών, μεσιτών πόρων και χρονοπρογραμματιστών. Οι μεσίτες πόρων – resource brokers εκτελούν την ανακάλυψη, την επιλογή και τη συνάθροιση των πόρων ενός διαφορετικού συνόλου διανεμημένων πόρων για έναν μεμονωμένο χρήστη. Αυτό σημαίνει, ότι κάθε χρήστης έχει τον δικό του μεσίτη πόρων του και ως εκ τούτου, μπορεί να βελτιστοποιήσει τις απαιτήσεις και τους στόχους του ιδιοκτήτη του.

4.3.1 Δυνατότητες του GridSim

Οι δυνατότητες του GridSim για την υλοποίηση προσομοιώσεων περιλαμβάνουν:

- Επιτρέπει τη μοντελοποίηση ετερογενών τύπων πόρων.
- Η δυνατότητες των πόρων μπορούν να καθοριστούν (υπό μορφή MIPS σύμφωνα με το SPEC benchmark).
- Οι πόροι μπορούν να βρίσκονται σε οποιαδήποτε χρονική ζώνη.
- Τα Σαββατοκύριακα και οι διακοπές μπορούν να χαρτογραφηθούν ανάλογα με την τοπική ώρα, δημιουργώντας έτσι χρονικές ζώνες μικρού φόρτου εργασίας.
- Οι πόροι μπορούν να δεσμευτούν προκαταβολικά για μελλοντική χρήση.

- Οι εργασίες των εφαρμογών μπορούν να είναι ετερογενείς και μπορούν να είναι εντατικές στη χρήση ΚΜΕ ή I/O.
- Δεν υπάρχει κανένα όριο στον αριθμό εργασιών εφαρμογής που μπορεί να υποβληθεί σε έναν πόρο.
- Οι πολλαπλάσιες οντότητες χρηστών μπορούν να υποβάλουν εργασίες για εκτέλεση ταυτόχρονα στον ίδιο πόρο (η επιλογή ποιων εργασιών θα εκτελέσει μπορεί να γίνει με ανταγωνιστικά οικονομικά μοντέλα).
- Η ταχύτητα δικτύων μεταξύ των πόρων μπορεί να καθοριστεί.

4.3.2 Οντότητες στο GridSim

Κατά τη διάρκεια της προσομοίωσης το GridSim δημιουργεί τις εξής οντότητες:

1. User - Χρήστης

   Κάθε αντίγραφο της οντότητας Χρήστης αντιπροσωπεύει ένα χρήστη του πλέγματος. Κάθε χρήστης μπορεί να διαφέρει από το υπόλοιπους όσον αφορά τα ακόλουθα χαρακτηριστικά:

- Τύπους εργασιών που δημιουργεί (π.χ. χρόνος εκτέλεσης εργασίας,)
- Στρατηγική βελτιστοποίησης (π.χ. ελαχιστοποίηση του κόστους, του χρόνου εκτέλεσης, ή και των δύο)
- Βαθμός δραστηριότητας (π.χ., πόσο συχνά δημιουργεί μια νέα εργασία)
- Απόλυτη προθεσμία και προϋπολογισμός (deadline and budget).

2. Broker – Μεσίτης (Μεσάζοντας)

   Κάθε χρήστης συνδέεται με ένα αντίγραφο της οντότητας μεσιτών. Ο χρήστης υποβάλλει αρχικά κάθε εργασία στον μεσίτη του και αυτός κανονίζει τις διεργασίες που πρέπει σύμφωνα με την πολιτική του χρήστη. Πριν το σχεδιασμό των διεργασιών, ο μεσίτης δυναμικά παίρνει έναν κατάλογο διαθέσιμων πόρων από την οντότητα γενικού καταλόγου (global directory entity). Επειδή κάθε μεσίτης προσπαθεί να βελτιστοποιήσει την πολιτική του χρήστη του, οι μεσίτες αναμένεται να αντιμετωπίσουν ακραίο ανταγωνισμό

για πρόσβαση στους πόρους. Οι αλγόριθμοι σχεδιασμού που χρησιμοποιούνται από τους μεσίτες πρέπει να είναι ιδιαίτερα προσαρμόσιμοι στην κατάσταση προσφοράς και ζήτησης της αγοράς.

3. Resource – Πόρος

   Κάθε περίπτωση οντότητας πόρων αντιπροσωπεύει έναν πόρο πλέγματος. Κάθε πόρος μπορεί να διαφέρει από το υπόλοιπο των πόρων όσον αφορά τα ακόλουθα χαρακτηριστικά:

   - Αριθμός επεξεργαστών.
   - Κόστος της επεξεργασίας.
   - Ταχύτητα της επεξεργασίας.
   - Εσωτερική πολιτική χρονοπρογραμματισμού διεργασιών
   - Τοπικός παράγοντας, φόρτου εργασίας.
   - Χρονική ζώνη.

   Η ταχύτητα των πόρων και ο χρόνος εκτέλεσης μιας εργασίας μπορούν να καθοριστούν με βάση πρότυπες μετρήσεις επιδόσεων όπως MIPS και SPEC.

4. Grid Information Service

   Παρέχει τις υπηρεσίες καταχώρησης των πόρων στο πλέγμα και ένα κατάλογο με όσους είναι διαθέσιμοι. Οι μεσίτες μπορούν να αναζητήσουν σε αυτό πληροφορίες για πόρους (χαρακτηριστικά, διαθεσιμότητα, τοποθεσία, ιδιοκτησία)

5. Input and Output

   Η ροή των πληροφοριών μεταξύ των οντοτήτων GridSim γίνεται μέσω των οντοτήτων Εισόδου – Εξόδου. Κάθε δικτυωμένη οντότητα GridSim έχει κανάλια I/O ή ports, που χρησιμοποιούνται για την επικοινωνία μιας οντότητας με τις δικές της οντότητες Εισόδου – Εξόδου.

4.4 Σχεδιασμός προσομοιώσεων

4.4.1 Εισαγωγή

Με τη χρήση του πακέτου εργαλείων GridSim προσομοιώνουμε ένα περιβάλλον πλέγματος. Το πλέγμα Grid που προσομοιώνεται περιέχει πολλές πηγές πόρων και κάθε χρήστης μπορεί να έχει διαφορετικές απαιτήσεις.

Κάθε χρήστης του Grid δημιουργεί ένα πείραμα-εργασία που αποτελείται από

1. Τις προδιαγραφές της εφαρμογής (ένα σύνολο Gridlets-υποεργασίες που αντιπροσωπεύουν τα μέρη της εφαρμογής που μπορούν να εκτελεστούν καθένα από μόνο του).
2. Την απαιτούμενη ποιότητα υπηρεσιών QoS (περιορισμούς προθεσμίας-deadline και προϋπολογισμού-budget, καθώς και την στρατηγική βελτιστοποίησης).

Κατά την προσομοίωση κάθε οντότητα χρηστή-User έχει ξεχωριστές απαιτήσεις σε πόρους και ποιότητα υπηρεσιών. Δημιουργεί λοιπόν την δικιά της οντότητα μεσάζοντα- Broker για την ανάθεση των Gridlets στους διάφορους πόρους.

Ο Μεσίτης πόρων πλέγματος -economic Grid Resource Broker που αναλαμβάνει την ανάθεση των εργασιών σε πόρους έχει υλοποιηθεί με τα εργαλεία του πακέτου GridSim. Υποστηρίζει την ανάθεση πόρων με στρατηγικές Περιορισμού Προθεσμίας και Προϋπολογισμού. Επίσης μπορεί να επιλέξει δυναμικά κατά το χρόνο εκτέλεσης τους κατάλληλους πόρους βασισμένος στην διαθεσιμότητα, ικανότητα και το κόστος τους, καθώς και στις απαιτήσεις ποιότητα υπηρεσιών (QoS) των χρηστών.

Ο Μεσίτης υποστηρίζει αλγορίθμους προθεσμίας-deadline και προϋπολογισμού-budget με τέσσερις διαφορετικές στρατηγικές βελτιστοποίησης:

- Κόστους - Cost.
- Χρόνου - Time.
- Κόστους_Χρόνου - Cost_time.
- Συντηρητικού χρόνου - Conservative time.

Αλγόριθμος/ Στρατηγική Βελτιστοποίησης	Περιγραφή	Χρόνος Εκτέλεσης (Προθεσμία- Deadline-D)	Κόστος Εκτέλεσης (Προϋπολογισμός- Budget-B)
Βελτιστοποίηση Κόστους	Χρησιμοποιούνται οι φτηνότεροι πόροι για να εξασφαλιστεί ότι μπορεί να επιτευχθεί η προθεσμία, ενώ παράλληλα το υπολογιστικό κόστος ελαχιστοποιείται.	Περιορίζεται από το Deadline-D που ορίζει ο χρήστης	Ελαχιστοποιείται
Βελτιστοποίηση Χρόνου	Χρησιμοποιούνται όλοι οι προσιτοί πόροι για την παράλληλη επεξεργασία των πόρων το συντομότερο δυνατόν.	Ελαχιστοποιείται	Περιορίζεται από το Budget-B που ορίζει ο χρήστης
Κόστους-Βέλτιστου Χρόνου Cost-Time Opt.	Παρόμοια με τη βελτιστοποίηση κόστους, μόνο που αν υπάρχουν πολλαπλοί πόροι με το ίδιο κόστος, εφαρμόζεται τότε στρατηγική βελτιστοποίησης χρόνου γι'αυτούς.	Ελαχιστοποιείται όταν αυτό είναι δυνατό	Ελαχιστοποιείται
Συντηρητική Βελτιστοποίηση χρόνου - Conservative time Optimisation	Παρόμοια με τη βελτιστοποίηση χρόνου, όμως εγγυάται ότι για κάθε μη επεξεργασμένη εργασία υπάρχει ένας ελάχιστος ποσό του προϋπολογισμού-ανά-εργασία.	Ελαχιστοποιείται	Περιορίζεται από το Budget-B που ορίζει ο χρήστης, αλλά για κάθε μη επεξεργασμένη εργασία προβλέπεται ένα ελάχιστο ποσό.

Πίνακας 4.1 Στρατηγικές Βελτιστοποίησης διαχείρισης στο GridSim

4.4.2 Ροή εργασιών που εκτελεί ο Μεσίτης πόρων πλέγματος - Grid Resource Broker κατά τη διάρκεια της προσομοίωσης.

1. Η οντότητα χρήστη δημιουργεί μια δοκιμασία-εργασία που περιέχει μια περιγραφή εφαρμογής (κατάλογο Gridlets που πρέπει να επεξεργαστούν) και τις απαιτήσεις των χρηστών για το μεσίτη.
2. Η υπομονάδα ανακάλυψης και συναλλαγής πόρων του Broker αλληλεπιδρά με την οντότητα GridSim GIS (Grid Information Service) για να προσδιορίσει τα στοιχεία επαφής των πόρων και αλληλεπιδρά έπειτα με τους πόρους για να υπολογίσει το τους κόστος πρόσβασης σε αυτούς. Δημιουργεί έναν κατάλογο (Broker Resource list) των πόρων με τις ιδιότητες τους, ένα κατάλογο των Gridlets που δεσμεύονται για εκτέλεση στον πόρο, και στοιχεία απόδοσης των πόρων.
3. Ο διαχειριστής διανομής εργασιών του Broker επιλέγει έναν κατάλληλο αλγόριθμο για την διανομή των Gridlets σε πόρους ανάλογα με τις απαιτήσεις χρηστών (όρια προθεσμίας και προϋπολογισμού, στρατηγική βελτιστοποίησης). Τα Gridlets που αποδίδονται σε έναν συγκεκριμένο πόρο για επεξεργασία προστίθενται στον κατάλογο Broker Resource list.
4. Για κάθε έναν από τους πόρους, ο αποστολέας (dispatcher) επιλέγει τον αριθμό Gridlets που μπορούν να οργανωθούν για εκτέλεση σύμφωνα με την πολιτική χρήσης για να αποφύγει την υπερφόρτωση των πόρων με απλές εργασίες χρηστών.
5. Ο αποστολέας (dispatcher) υποβάλλει έπειτα Gridlets στους πόρους χρησιμοποιώντας την ασύγχρονη υπηρεσία του GridSim.
6. Όταν η επεξεργασία ενός Gridlet ολοκληρωθεί, ο πόρος το επιστρέφει στο δέκτη Gridlet του Broker, ο οποίος μετρά και έπειτα ενημερώνει την παράμετρο χρόνου εκτέλεσης και τον αριθμό των διαθέσιμων πόρων. Αυτό βοηθά στην πρόβλεψη του ποσοστού κατανάλωσης πόρων για τις μελλοντικές αποφάσεις σχεδιασμού.

7. Τα βήματα, 3 –6, συνεχίζονται έως ότου υποβληθούν σε επεξεργασία όλα τα Gridlets ή ο Broker υπερβεί την προθεσμία (deadline) ή το όριο του προϋπολογισμού (budget). Έπειτα ο broker επιστρέφει έπειτα τα ενημερωμένα δεδομένα της εργασίας-πειράματος μαζί με τα επεξεργασμένα Gridlets πίσω στην οντότητα χρήστη.

5

Σχεδιασμός προσομοίωσης «Οικονομίας-Grid» ακολουθώντας στρατηγική βελτιστοποίησης κόστους.

Σε αυτή την ενότητα, παρουσιάζεται το περιβάλλον του υπολογιστικού πλέγματος που προσομοιώνεται. Δίδονται τα χαρακτηριστικά των πόρων που αποτελούν το πλέγμα και οι προς εκτέλεση εργασίες που εκφράζονται με τη μορφή Gridlets. Εξετάζονται τα εξής σενάρια όπου και στα δυο ακολουθείται στρατηγική βελτιστοποίησης κόστους:

Ένας χρήστης υποβάλλει τις εργασίες του για εκτέλεση στο πλέγμα με 96 διαφορετικούς συνδυασμούς απαιτήσεων Κόστους-Προθεσμίας.

Ένας, 6 , 15 και 23 χρήστες υποβάλλουν ταυτόχρονα τις εργασίες τους για εκτέλεση στο πλέγμα με παρόμοιες περίπου απαιτήσεις για 12 διαφορετικές απαιτήσεις Κόστους κάθε φορά.

5.1 Χαρακτηριστικά πλέγματος που προσομοιώνεται.

5.1.1 Χαρακτηριστικά Πόρων

Το Grid που προσομοιώνουμε αποτελείται από ένα αριθμό πόρων που έχουν διαφορετικά χαρακτηριστικά και απόδοση ενώ βρίσκονται διεσπαρμένοι σε διάφορα σημεία του κόσμου. Για πόρους έχουμε επιλέξει τέσσερα συστήματα που χρησιμοποιούν τα τελευταία μοντέλα Κεντρικών Μονάδων Επεξεργασίας (CPUs).

Την υπολογιστική ικανότητα αυτών των μονάδας επεξεργασίας (ΜΕ) την εκφράζουμε με βάση τα αποτελέσματα της συγκριτικής μέτρησης επιδόσεων SPEC CPU2000(INT) που δημοσιεύονται εδώ [42].

Για να μπορούν οι χρήστες να εκφράσουν τις απαιτήσεις των εφαρμογών τους σε μονάδες MIPS (million instructions per second), υποθέτουμε στην προσομοίωση ότι η εκτίμηση σε MIPS της ικανότητας κάθε μονάδας επεξεργασίας (ΜΕ) είναι ίδια με αυτήν που μετράται ως επίδοση στο SPEC CPU2000 (INT).

Hardware Vendor:	Sun Microsystems	Advanced Micro Devices	Intel(R) Corporation	IBM Corporation
Model Name:	Sun Ultra 40	ASUS A8N-SLI Deluxe, AMD Athlon (TM) 64 FX-57	Intel(R) D955XBK motherboard(3.73 GHz, Intel(R) Pentium(R) 4 processor Extreme Edition supporting Hyper-Threading Technology)	IBM eServer xSeries 236 (3.8 GHz Xeon, 2MB L2 Cache)
CPU:	AMD Opteron (TM) 256	AMD Athlon (TM) 64 FX-57	Intel(R) Pentium(R) 4 processor Extreme Edition supporting Hyper-Threading Technology(3.73 GHz, 1066 MHz bus)	Intel Xeon 3.8 (800 MHz system bus)
CPU MHz:	3000	2800	3733	3800
SPECint_base2000 =	**1836**	**1862**	**1833**	**1810**

Πίνακας 5.1 Χαρακτηριστικά των CPU που οργανώνονται ως πόροι

Κάθε πόρος του πλέγματος που προσομοιώνουμε μπορεί να περιέχει ένα ή περισσότερα μηχανήματα και κάθε μηχάνημα μία ή περισσότερες μονάδες επεξεργασίας (ΜΕ).

Επίσης για κάθε πόρο καθορίζεται:

- η γεωγραφική του θέση - εκφράζεται με παράμερο την χρονική του ζώνης (GMT).
- το baudRate - ταχύτητα επικοινωνίας της σύνδεσης σε bits/s.
- το peakLoad - το φόρτο εργασίας σε ώρες αιχμής (εφράζεται ως ποσοστό της συνολικής ικάνοτητας του πόρου, παίρνει τιμή από 0 έως 1)
- offPeakLoad – όμοια το φόρτο εργασίας σε ώρες όμως μη-αιχμής.
- holidayLoad - το φόρτο εργασίας σε περίοδο διακοπών
- επίσης μπορεί να καθοριστεί αν ο πόρος είναι ενεργός ή όχι τα Σαββατοκύριακα και κατά τη διάρκεια διακοπών.
- allocation policy – η πολιτική κατανομής που ακολουθείται από τον πόρο κατά την εκτέλεση. Μπορεί να είναι time- ή space-shared. Οι πόροι που προσομοιώνουμε λειτουργούν ως time-shared.
- το κόστος χρήσης του πόρου

5.1.2 Κόστος χρήσης πόρων

Το κόστος χρήσης κάθε μονάδας επεξεργασίας (ΜΕ) ανά μονάδα χρόνου ορίζεται ως G € (Grid Euro). Το κόστος χρήσης κάθε μονάδας επεξεργασίας (ΜΕ) σε G€ / μονάδα χρόνου, δεν αντιπροσωπεύει το κόστος επεξεργασίας και δεν μπορεί να αποτελέσει μέτρο σύγκρισης όταν οι ΜΕ έχουν διαφορετική ικανότητα MIPS.

Έτσι σε ένα υπολογιστικό πλέγμα όπως αυτό που προσομοιώνουμε, όπου συμμετέχουν πόροι με διαφορετική ικανότητα MIPS, μέτρο σύγκρισης τιμών αποτελεί το MIPS ανά G € κάθε ΜΕ.

Παράδειγμα: Έστω ότι έχουμε δυο μονάδες επεξεργασίας: την ΜΕ1 και την ΜΕ2.

Η χρήση της ΜΕ1 κοστίζει 8 G € / μονάδα χρόνου και έχει ικανότητα 900MIPS.

Η χρήση της ΜΕ2 κοστίζει 4 G € / μονάδα χρόνου και έχει ικανότητα 400MIPS.

Αν κρίνουμε από το κόστος σε G € / μονάδα χρόνου, βλέπουμε ότι η ΜΕ1 είναι ακριβότερη και μάλλον δεν συμφέρει η επιλογή της.

Όμως: για την ΜΕ1: 900/8= 225MIPS ανά G €

για την ΜΕ2: 400/4=100 MIPS ανά G € ,

που σημαίνει ότι για κάθε Grid euro που πληρώνουμε στην ΜΕ1 αυτή μας αποδίδει περισσότερα MIPS για την εκτέλεση των εργασιών μας, συνεπώς είναι συμφερότερη η επιλογή της.

Άρα λοιπόν μας ενδιαφέρουν τα MIPS ανά G € (million instructions / Grid euro) κάθε πόρου και βάσει αυτών οι brokers πρέπει να υπολογίζουν το σχετικό κόστος που απαιτείται για την επεξεργασία των Gridlets που υποβάλλουν οι χρήστες.

Στον πίνακα που ακολουθεί φαίνονται οι πόροι που αποτελούν το προς προσομοίωση πλέγμα.

Όνομα Πόρου	Λίστα Μηχανημάτων Πόρου	Αριθμός ΜΕ ανά μηχάνημα	Αριθμός ΜΕ που αποτελούν τον πόρο	Είδος ΜΕ	Εκτίμηση MIPS της ΜΕ / SPEC	Συνολική εκτίμηση MIPS πόρου	Γεωγραφική θέση	Time Zone	baud (Mbps)	Φορτίο σε ώρες αιχμής (%)	Φορτίο σε ώρες μηαιχμής (%)	Φορτίο κατά τις διακοπές (%)	Τιμή σε G€ / ΜΕ μονάδα χρόνου	Τιμή σε MIPS / G€
Resource_1	R1_Machine0	6	12	AMD Opteron (TM) 256	1836	22032	UK	0	5.5 Mbps	50	30	10	7	262.2857
	R1_Machine1	6												
Resource_2	R2_Machine0	6	6	AMD Athlon (TM) 64 FX-57	1862	11172	Italy	1	6.7 Mbps	60	30	0	8	232.75
Resource_3	R3_Machine0	4	12	Intel(R) Pentium(R) 4 (3.73 GHz, 1066 MHz bus)	1833	21996	Germany	1	7 Mbps	80	30	0	3	611
	R3_Machine1	4												
	R3_Machine2	4												
Resource_4	R4_Machine0	16	32	Intel Xeon 3.8 (800 MHz system bus)	1810	57920	Japan	9	7.5 Mbps	80	50	10	2	905
	R4_Machine1	16												
Resource_5	R5_Machine0	4	4	AMD Opteron (TM) 256	1836	7344	Italy	1	5.5 Mbps	50	20	0	6	306

Πίνακας 5.3 Χαρακτηριστικά των πόρων που συμμετέχουν στο πλέγμα που προσομοιώνουμε (α)

Όνομα Πόρου	Λίστα Μηχανημάτων Πόρου	Αριθμός ΜΕ ανά μηχάνημα	Αριθμός ΜΕ που αποτελούν τον πόρο	Είδος ΜΕ	Εκτίμηση MIPS της ΜΕ / SPEC	Συνολική εκτίμηση MIPS πόρου	Γεωγραφική θέση	Time Zone	baud (Mbps)	Φορτίο σε ώρες αιχμής (%)	Φορτίο σε ώρες μηαιχμής (%)	Φορτίο κατά τις διακοπές (%)	Τιμή σε G€ / ΜΕ μονάδα χρόνου	Τιμή σε MIPS / G€
Resource_6	R6_Machine0	8	8	AMD Opteron (TM) 256	1836	14688	Los Angeles	8	7 Mbps	50	30	10	7	262.2857
Resource_7	R7_Machine0	4	16	AMD Opteron (TM) 256	1836	29376	Washington	6	8 Mbps	40	10	0	6	306
	R7_Machine1	4												
	R7_Machine2	4												
	R7_Machine3	4												
Resource_8	R8_Machine0	6	6	AMD Athlon (TM) 64 FX-57	1862	11172	UK	0	6.5 Mbps	70	40	20	8	232.75
Resource_9	R9_Machine0	8	24	Intel(R) Pentium(R) 4 (3.73 GHz, 1066 MHz bus)	1833	43992	Finland	2	5 Mbps	60	20	0	2	916.5
	R9_Machine1	8												
	R9_Machine2	8												
Resource_10	R10_Machine0	2	2	Intel Xeon 3.8 (800 MHz system bus)	1810	3620	Germany	1	6.5 Mbps	50	10	0	1	1810

Πίνακας 5.4 Χαρακτηριστικά των πόρων που συμμετέχουν στο πλέγμα που προσομοιώνουμε (β)

5.1.3 Εργασίες προς εκτέλεση

Όσον αφορά την εφαρμογή που πρέπει να εκτελεστεί στο πλέγμα υποθέτουμε ότι αυτή αποτελείται από 500 υποεργασίες – Gridlets, που πρέπει να αποδοθούν στους πόρους. Κάθε Gridlet εκφράζεται από το μέγεθός του σε MI, το μέγεθος των δεδομένων εισόδου και εξόδου καθώς και από πλήθος άλλων παραμέτρων. Για να πλησιάζει η προσομοίωση σε εφαρμογές του πραγματικού κόσμου εισάγουμε σε όλα τα χαρακτηριστικά των εργασιών μικρές αποκλίσεις ώστε να διαφέρουν μεταξύ τους τα Gridlets.

Gridlets			
	Μέγεθος	Ελάχιστη Απόκλιση (%)	Μέγιστη Απόκλιση (%)
Gridlet	500	0	5.1
Μήκος	10000	3.2	8.3
Αρχείο	1000	5	7.1
Έξοδος	1000	4.4	6

Πίνακας 5.6 Εργασίες που εκτελούνται στο πλέγμα.

Σημειώνεται ότι λόγω της κατασκευής του GridSim το μήκος των εργασιών εκφράζεται σε όρους χρόνου που απαιτείται για να εκτελεστεί σε ένα πόρο με SPEC/MIPS rating 100. Ο χρόνος επεξεργασίας των Gridlet εκφράζεται με τέτοιο τρόπο όπου αναμένεται να χρειάζονται τουλάχιστον 100 μονάδες χρόνου. Δηλαδή το μήκος ενός Gridlet μπορεί να είναι τουλάχιστον 10000MI.

5.2 Προσομοιώσεις βάσει απαιτήσεων των χρηστών. Καθορισμένος Προϋπολογισμός και Προθεσμία

5.2.1 Ένας χρήστης υποβάλλει εργασίες στο πλέγμα

Προσομοιώνουμε ένα χρήστη ο οποίος υποβάλει αιτήματα για την εκτέλεση των 500 Gridlet με διαφορετικά κάθε φορά:

- ✓ Την προθεσμία – deadline ολοκλήρωσης των εργασιών.

  Η προθεσμία μεταβάλλεται από 150 έως 395 μονάδες χρόνου με βήμα 35.

- ✓ Τα χρήματα – budget που είναι διατεθειμένος να πληρώσει.

  Ο προϋπολογισμός μεταβάλλεται από 1400 έως 8000 με βήμα 600.

Εξετάζεται λοιπόν η Στρατηγική Βελτιστοποίησης Κόστους για 96 συνδυασμούς προϋπολογισμού – προθεσμίας όπου :

Προϋπολογισμός	Προθεσμία
1400	150
2000	185
2600	220
3200	255
3800	290
4400	325
5000	360
5600	395
6200	
6800	
7400	
8000	

Πίνακας 5.7 Προσομοιωμένες τιμές Προϋπολογισμού - Προθεσμίας

Οι απαιτήσεις του χρήστη επιβάλλουν περιορισμούς στον αριθμό των Gridlets που τελικά εκτελούνται. Για κάθε λοιπόν συνδυασμό προϋπολογισμού – προθεσμίας καταγράφουμε των αριθμό των Gridlets που εκτελούνται.

5.2.1.1 Αριθμός Gridlets που εκτελούνται για διαφορετικά όρια προϋπολογισμού, με καθορισμένη κάθε φορά την προθεσμία.

Προθε-σμία	Προϋπο-λογισμός	Gridlets που ολοκλη-ρώθηκαν
150	1400	140
	2000	191
	2600	213
	3200	224
	3800	238
	4400	242
	5000	244
	5600	246
	6200	248
	6800	243
	7400	244
	8000	244

Προθε-σμία	Προϋπο-λογισμός	Gridlets που ολοκλη-ρώθηκαν
185	1400	141
	2000	197
	2600	245
	3200	257
	3800	266
	4400	284
	5000	300
	5600	301
	6200	301
	6800	301
	7400	289
	8000	300

Προθε-σμία	Προϋπο-λογισμός	Gridlets που ολοκλη-ρώθηκαν
220	1400	144
	2000	197
	2600	253
	3200	292
	3800	303
	4400	322
	5000	331
	5600	333
	6200	333
	6800	333
	7400	333
	8000	333

Προθε-σμία	Προϋπο-λογισμός	Gridlets που ολοκλη-ρώθηκαν
255	1400	148
	2000	200
	2600	253
	3200	309
	3800	340
	4400	349
	5000	379
	5600	378
	6200	390
	6800	390
	7400	390
	8000	390

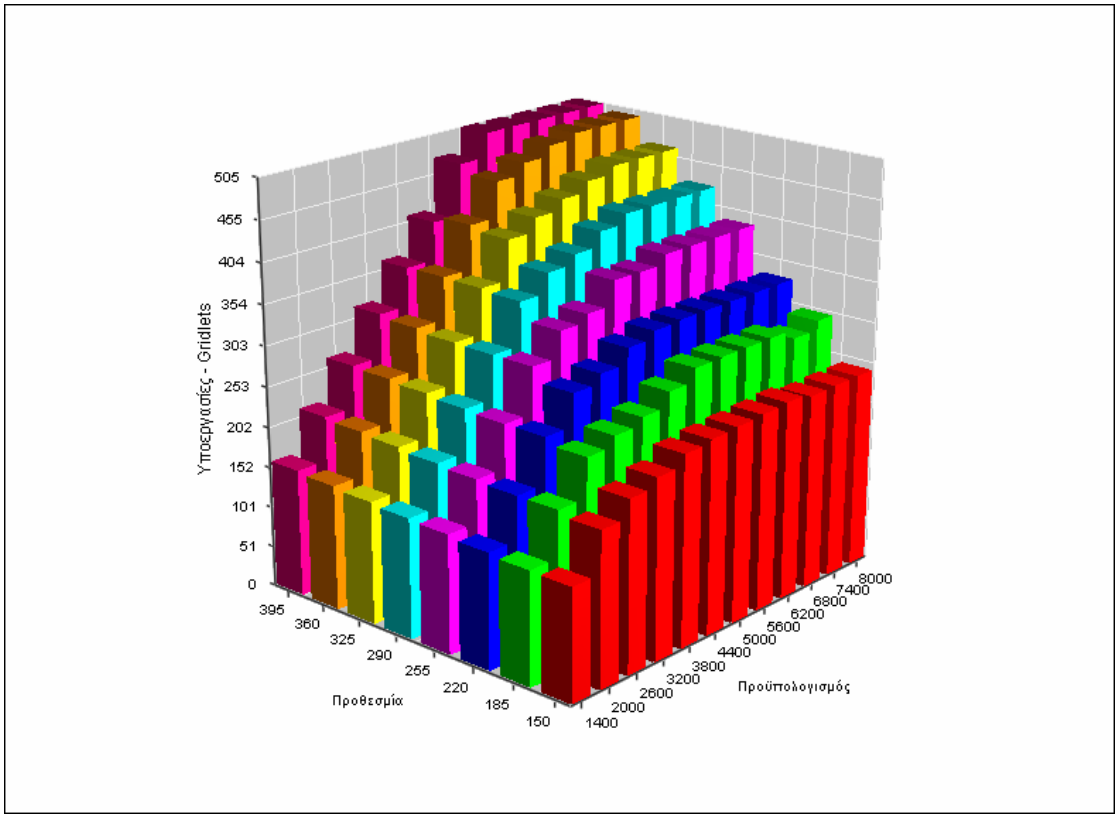
Προθε-σμία	Προϋπο-λογισμός	Gridlets που ολοκλη-ρώθηκαν
290	1400	150
	2000	203
	2600	256
	3200	309
	3800	362
	4400	386
	5000	398
	5600	416
	6200	428
	6800	428
	7400	428
	8000	428

Προθε-σμία	Προϋπο-λογισμός	Gridlets που ολοκλη-ρώθηκαν
325	1400	153
	2000	207
	2600	260
	3200	310
	3800	365
	4400	414
	5000	431
	5600	444
	6200	457
	6800	467
	7400	467
	8000	467

Προθε-σμία	Προϋπο-λογισμός	Gridlets που ολοκλη-ρώθηκαν
360	1400	156
	2000	210
	2600	263
	3200	314
	3800	366
	4400	420
	5000	464
	5600	478
	6200	491
	6800	498
	7400	498
	8000	498

Προθε-σμία	Προϋπο-λογισμός	Gridlets που ολοκλη-ρώθηκαν
395	1400	159
	2000	213
	2600	265
	3200	318
	3800	365
	4400	412
	5000	475
	5600	505
	6200	505
	6800	505
	7400	505
	8000	505

Πίνακας 5.8 Αριθμός Gridlets που εκτελούνται για διαφορετικά όρια προϋπολογισμού, με καθορισμένη κάθε φορά την προθεσμία



Σχήμα 5.1 Αριθμός Gridlets που εκτελούνται για διαφορετικά όρια προϋπολογισμού, με καθορισμένη κάθε φορά την προθεσμία

Παρατηρώντας τα αποτελέσματα των προσομοιώσεων καθώς και την τρισδιάστατη συγκριτική απεικόνισή τους εξάγουμε τα παρακάτω συμπεράσματα:

- ✓ Όταν ο χρήστης θέτει στενή προθεσμία–deadline τότε ο αριθμός των εργασιών που εκτελείται αυξάνεται όσο μεγαλώνει και το χρηματικό ποσό που διατίθεται να πληρώσει. Αυτό συμβαίνει γιατί όταν ο χρήστης διαθέτει περισσότερα χρήματα μπορεί να δεσμεύσει ακριβότερους πόρους αλλά και περισσότερους μέσα σε μικρό χρονικό διάστημα.
- ✓ Η στενή προθεσμία πάντως θέτει από μόνη της περιορισμό στο μέγιστο αριθμό gridlets που μπορούν να επεξεργαστούν. Έτσι με καθορισμένο το μικρότερο περιθώριο χρόνου (150) προκύπτει ότι με προϋπολογισμό 6200 επιτυγχάνεται ο μέγιστος αριθμός gridlets που μπορεί να επεξεργαστεί. Περαιτέρω αύξηση του προϋπολογισμού δεν οδηγεί σε αύξηση των gridlet που εκτελούνται.
- ✓ Εναλλακτικά όταν ο χρήστης κράτα μικρό το ποσό που διατίθεται να πληρώσει, τότε ο αριθμός των εργασιών που εκτελούνται αυξάνει με την χαλάρωση της προθεσμίας. Και εδώ όμως ο μικρός προϋπολογισμός δημιουργεί ένα φράγμα στον αριθμό των εργασιών που μπορούν να εκτελεσθούν.

5.2.1.2 Αριθμός Gridlet που εκτελούνται για διαφορετική κάθε φορά προθεσμία, με καθορισμένο κάθε φορά το όριο προϋπολογισμού.

Προϋπο-λογισμός	Προθε-σμία	Gridlets που ολοκλη-ρώθηκαν
1400	150	140
	185	141
	220	144
	255	148
	290	150
	325	153
	360	156
	395	159

Προϋπο-λογισμός	Προθε-σμία	Gridlets που ολοκλη-ρώθηκαν
2000	150	191
	185	197
	220	197
	255	200
	290	203
	325	207
	360	210
	395	213

Προϋπο-λογισμός	Προθε-σμία	Gridlets που ολοκλη-ρώθηκαν
2600	150	213
	185	245
	220	253
	255	253
	290	256
	325	260
	360	263
	395	265

Προϋπο-λογισμός	Προθε-σμία	Gridlets που ολοκλη-ρώθηκαν
3200	150	224
	185	257
	220	292
	255	309
	290	309
	325	310
	360	314
	395	318

Προϋπο-λογισμός	Προθε-σμία	Gridlets που ολοκλη-ρώθηκαν
3800	150	238
	185	266
	220	303
	255	340
	290	362
	325	365
	360	366
	395	365

Προϋπο-λογισμός	Προθε-σμία	Gridlets που ολοκλη-ρώθηκαν
4400	150	242
	185	284
	220	322
	255	349
	290	386
	325	414
	360	420
	395	412

Προϋπο-λογισμός	Προθε-σμία	Gridlets που ολοκλη-ρώθηκαν
5000	150	244
	185	300
	220	331
	255	379
	290	398
	325	431

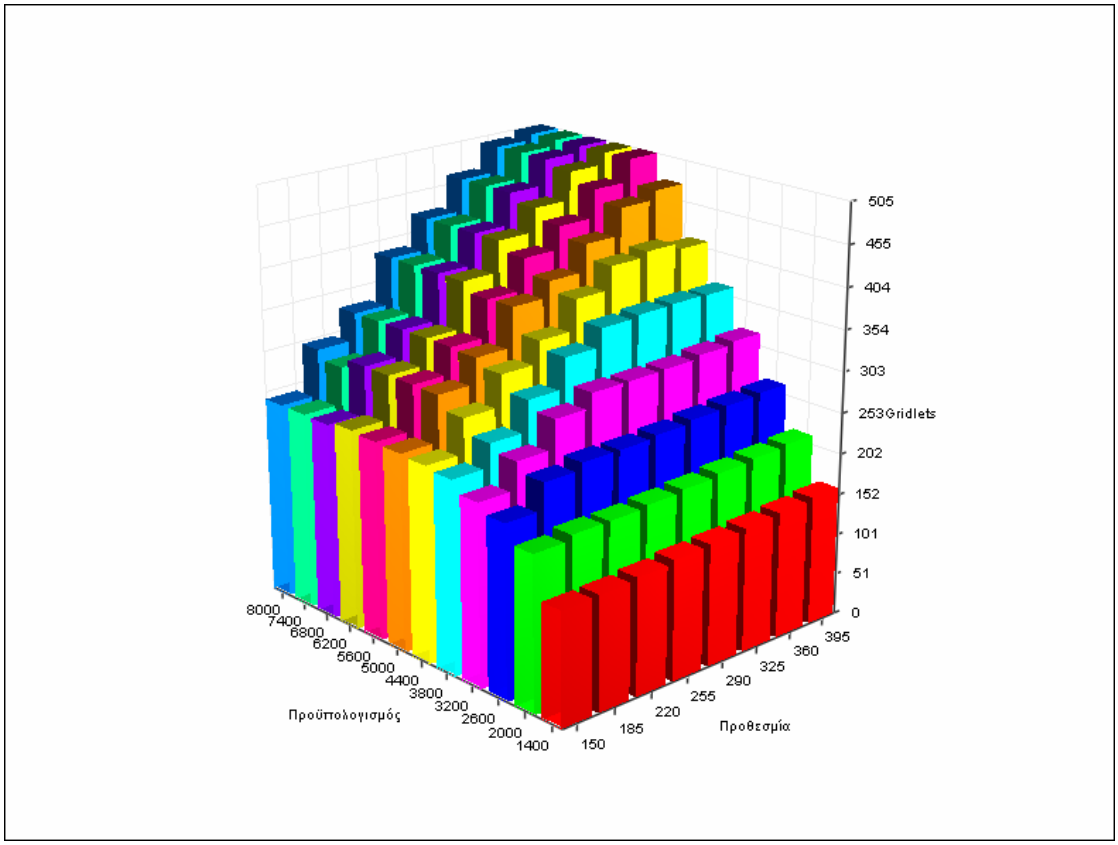
Προϋπο-λογισμός	Προθε-σμία	Gridlets που ολοκλη-ρώθηκαν
5600	150	246
	185	301
	220	333
	255	378
	290	416
	325	444

Προϋπο-λογισμός	Προθε-σμία	Gridlets που ολοκλη-ρώθηκαν
6200	150	248
	185	301
	220	333
	255	390
	290	428
	325	457

	360	464		360	478		360	491
	395	475		395	505		395	505

Προϋπο-λογισμός	Προθε-σμία	Gridlets που ολοκλη-ρώθηκαν	Προϋπο-λογισμός	Προθε-σμία	Gridlets που ολοκλη-ρώθηκαν	Προϋπο-λογισμός	Προθε-σμία	Gridlets που ολοκλη-ρώθηκαν
6800	150	243	7400	150	244	8000	150	244
	185	301		185	289		185	300
	220	333		220	333		220	333
	255	390		255	390		255	390
	290	428		290	428		290	428
	325	467		325	467		325	467
	360	498		360	498		360	498
	395	505		395	505		395	505

Πίνακας 5.9 Αριθμός Gridlet που εκτελούνται για διαφορετική κάθε φορά προθεσμία, με καθορισμένο κάθε φορά το όριο προϋπολογισμού.



Σχήμα 5.2 Αριθμός Gridlet που εκτελούνται για διαφορετική κάθε φορά προθεσμία, με καθορισμένο κάθε φορά το όριο προϋπολογισμού.

Διατηρώντας τον προϋπολογισμό και χαλαρώνοντας σταδιακά την προθεσμία , βλέπουμε πως αυξάνονται τα Gridlets που εκτελούνται.

Ας δούμε το ρυθμό αύξησης των Gridlets.

	Ελάχιστη Προθεσμία 150	**Μέγιστη Προθεσμία 395**	**Αύξηση**
Προϋπολογισμός	**Gridlets**		
1400	140	159	**19**
2000	191	213	**22**
2600	213	265	**52**
3200	224	318	**94**
3800	238	365	**127**
4400	242	412	**170**
5000	244	475	**231**
5600	246	505	**259**
6200	248	505	**257**
6800	243	505	**262**
7400	244	505	**261**
8000	244	505	**261**

Πίνακας 5.10 Ρυθμός αύξησης των Gridlets με με καθορισμένο κάθε φορά το όριο προϋπολογισμού

Παρατηρούμε ότι ο ρυθμός αύξησης των gridlets είναι σημαντικός όταν ξεκινάμε με αρκετά μεγάλο προϋπολογισμό (5000 και άνω) ενώ για προϋπολογισμό από 1400 έως 4400 παρότι αυξάνουμε την προθεσμία δεν έχουμε αντίστοιχη αύξηση των Gridlets.

Αυτό δείχνει ότι όταν ο χρήστης διατηρεί χαμηλά τον προϋπολογισμό, από τους διαθέσιμους πόρους μπορεί να χρησιμοποιήσει μόνο τους φθηνότερους οπότε όσο κι αν χαλαρώσει την προθεσμία δεν θα επιτύχει την επεξεργασία περισσότερων Gridlet.

Ενώ ξεκινώντας με σχετικά μεγάλο διαθέσιμο ποσό μπορεί να δεσμεύσει περισσότερους και γρηγορότερους πόρους οπότε σε συνδυασμό με την χαλάρωση της προθεσμίας μπορεί να εκτελέσει μεγάλο αριθμό από τα Gridlet του.

5.2.2 Πολλοί χρήστες υποβάλλουν ταυτόχρονα εργασίες στο πλέγμα

Σε αυτή την ενότητα προσομοιώνουμε με το GridSim το ίδιο υπολογιστικό πλέγμα με πολλούς χρήστες να διεκδικούν την δέσμευση των πόρων. Εκτελούμε την προσομοίωση για 6, 15, 23 χρήστες και συγκρίνουμε με την ύπαρξη ενός μόνο χρήστη στο πλέγμα.

Ακολουθούμε και εδώ στρατηγική βελτιστοποίησης κόστους. Για να μπορέσουμε να συγκρίνουμε τους χρήστες μεταξύ τους και κατανοήσουμε την όλη λειτουργία ενός τέτοιου σεναρίου θεωρούμε τους χρήστες σχεδόν με τις ίδιες απαιτήσεις και εξετάζουμε την συμπεριφορά του όλου συστήματος για σταθερή προθεσμία 360, ενώ μεταβάλλουμε τον προϋπολογισμό από 1400 έως 8000 G €.

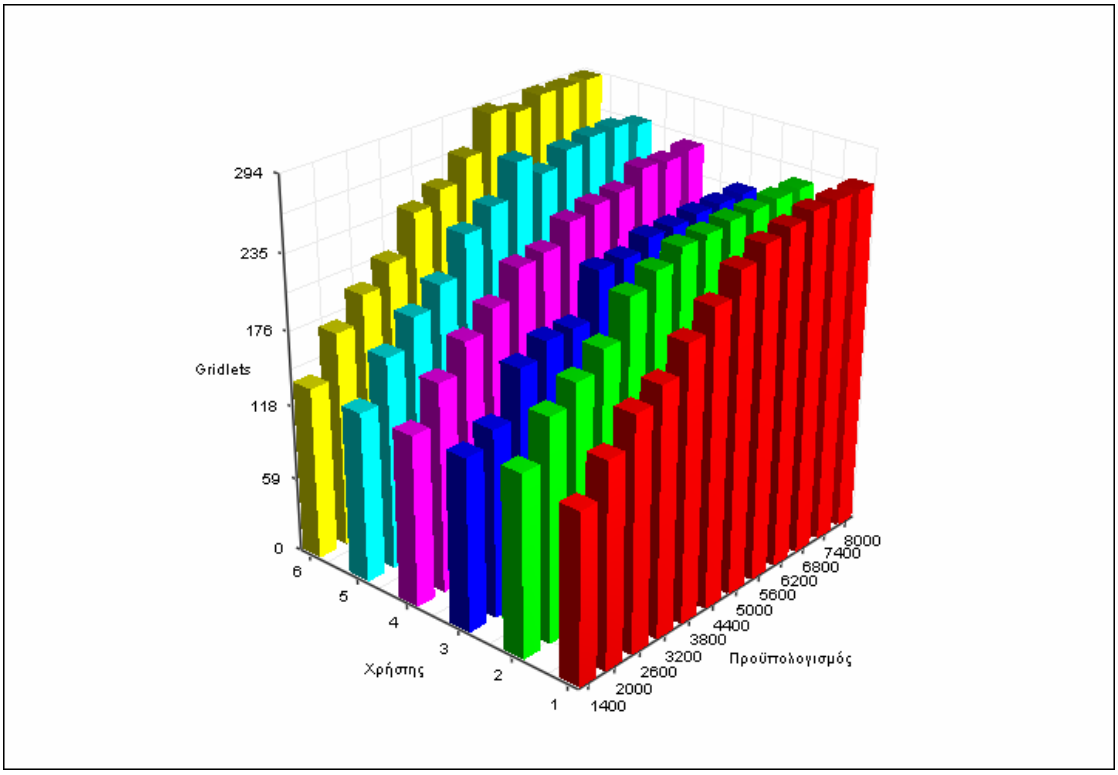
Με την εισαγωγή διαφορετικών αποκλίσεων στον αριθμό 500 των gridlet κάθε χρήστη και διαφορετικών αποκλίσεων στο μέγεθος τους, δημιουργούνται μικρές διαφορές μεταξύ των εργασιών που υποβάλλουν οι χρήστες.

5.2.2.1 Προσομοίωση με 6 χρήστες στο πλέγμα

Προθεσμία 360	Gridlets που ολοκληρώθηκαν για κάθε ένα από τους 6 χρήστες					
Προϋπολογισμός	1ος Χρήστης	2ος Χρήστης	3ος Χρήστης	4ος Χρήστης	5ος Χρήστης	6ος Χρήστης
1400	134	143	136	135	135	137
2000	160	174	147	165	168	171
2600	183	189	183	187	190	192
3200	194	204	194	203	206	208
3800	215	233	194	225	236	239
4400	232	243	229	228	249	249
5000	250	252	229	243	275	266
5600	260	253	237	248	259	293
6200	264	255	237	250	270	287
6800	268	255	239	261	273	294

7400	268	256	239	259	273	293
8000	268	256	239	261	268	293

Πίνακας 5.11 Αριθμός Gridlet που εκτελούνται για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν συμμετέχουν 6 καταναλωτές πόρων στο πλέγμα.

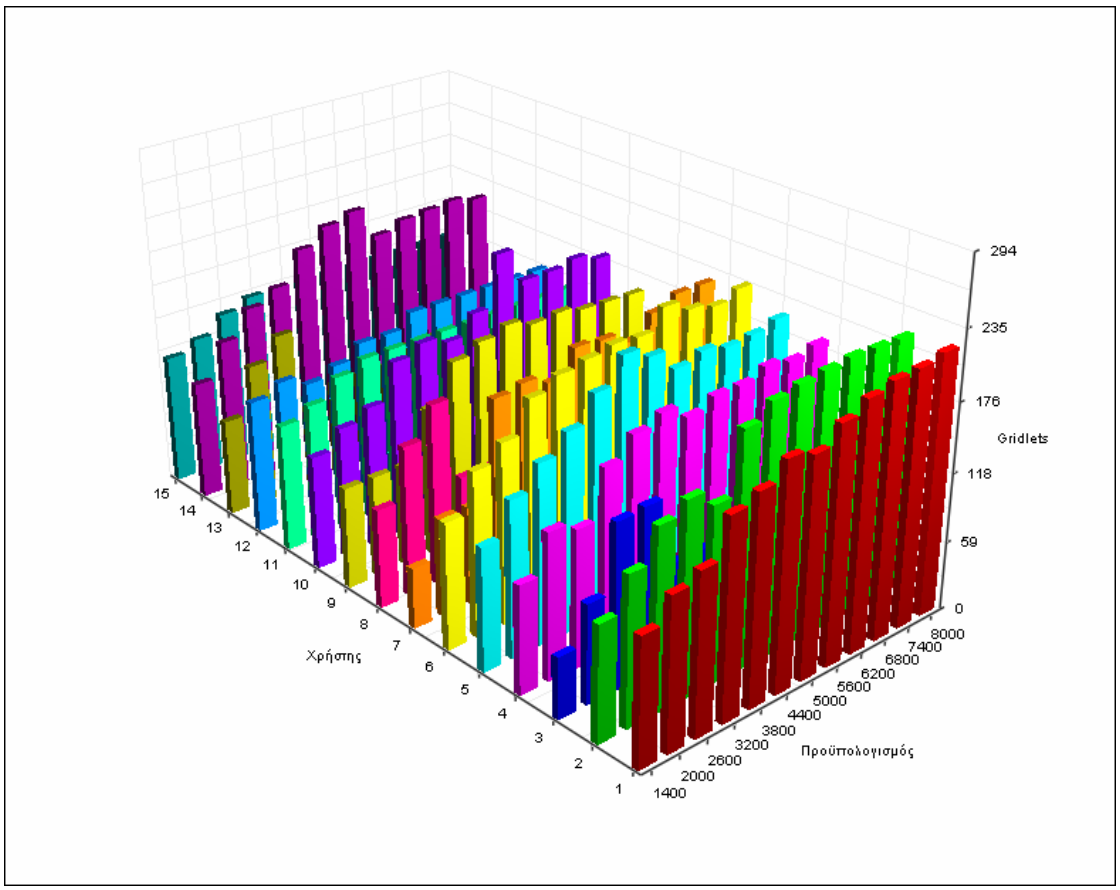


Σχήμα 5.3 Αριθμός Gridlet που εκτελούνται για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν συμμετέχουν 6 καταναλωτές πόρων στο πλέγμα.

5.2.2.2 Προσομοίωση με 15 χρήστες στο πλέγμα

Προθεσμία 360	**Gridlets που ολοκληρώθηκαν για κάθε ένα από τους 15 χρήστες**							
Προϋπολο-γισμός	**1ος Χρήστης**	**2ος Χρήστης**	**3ος Χρήστης**	**4ος Χρήστης**	**5ος Χρήστης**	**6ος Χρήστης**	**7ος Χρήστης**	**8ος Χρήστης**
1400	108	98	52	92	105	108	52	85
2000	128	126	84	124	133	140	84	129
2600	136	153	139	116	153	154	84	155
3200	168	163	141	156	169	181	166	84
3800	176	149	84	173	191	191	170	84
4400	191	199	84	181	213	195	162	84
5000	185	211	84	169	203	199	183	84
5600	200	215	84	177	185	197	181	89
6200	210	217	84	177	191	214	171	89
6800	215	219	84	184	186	204	188	89
7400	216	218	84	180	188	199	197	89
8000	220	217	84	185	193	206	197	89
Προϋπολο-γισμός	**9ος Χρήστης**	**10ος Χρήστης**	**11ος Χρήστης**	**12ος Χρήστης**	**13ος Χρήστης**	**14ος Χρήστης**	**15ος Χρήστης**	
1400	87	98	110	116	84	103	113	
2000	87	114	119	125	125	133	122	
2600	87	123	135	115	144	155	137	
3200	128	154	143	123	84	166	145	
3800	160	162	143	135	84	193	145	
4400	169	155	143	135	84	207	145	
5000	175	172	143	147	84	213	145	
5600	168	216	143	147	96	185	145	
6200	170	186	143	147	96	192	145	
6800	165	186	143	147	96	194	145	
7400	164	190	143	147	96	196	145	
8000	163	183	143	147	96	191	145	

Πίνακας 5.12 Αριθμός Gridlet που εκτελούνται για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν συμμετέχουν 15 καταναλωτές πόρων στο πλέγμα.



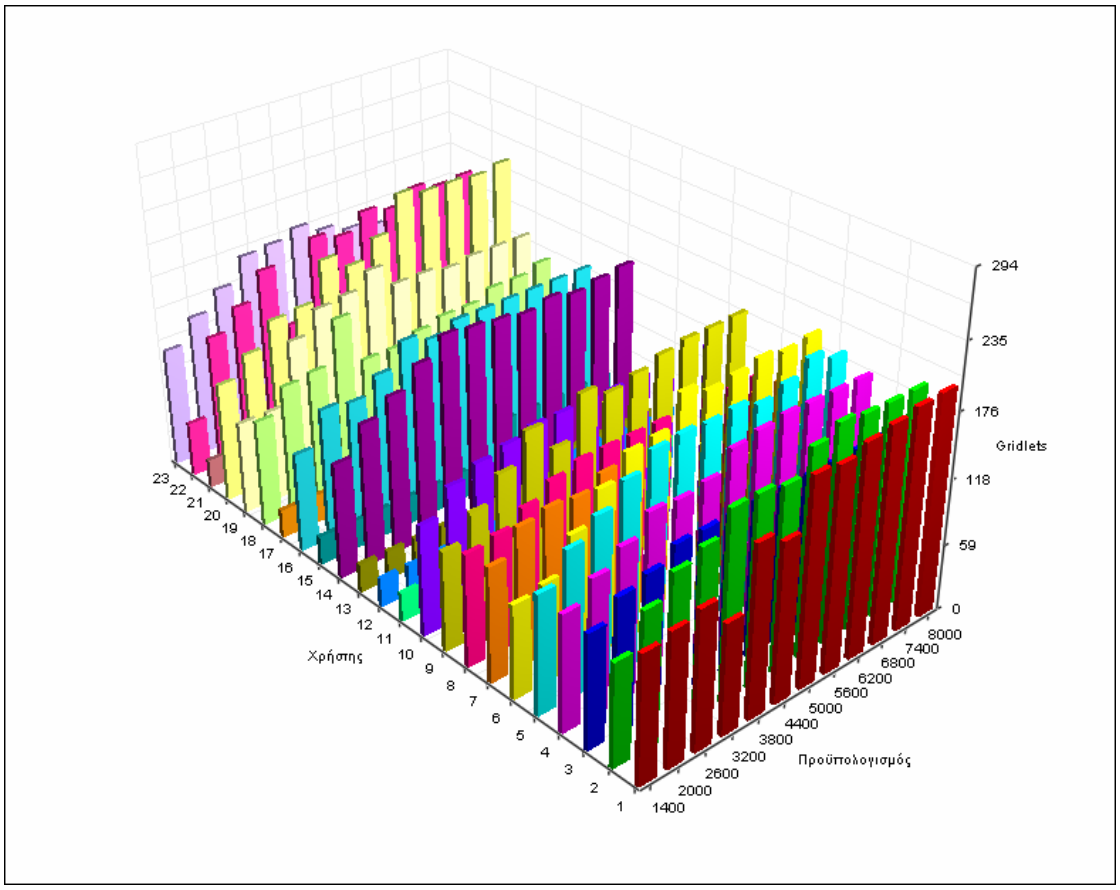
Σχήμα 5.4 Αριθμός Gridlet που εκτελούνται για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν συμμετέχουν 15 καταναλωτές πόρων στο πλέγμα.

5.2.2.3 Προσομοίωση με 23 χρήστες στο πλέγμα

Προθεσμία 360	**Gridlets που ολοκληρώθηκαν για κάθε ένα από τους 15 χρήστες**							
Προϋπολο-γισμός	**1ος Χρήστης**	**2ος Χρήστης**	**3ος Χρήστης**	**4ος Χρήστης**	**5ος Χρήστης**	**6ος Χρήστης**	**7ος Χρήστης**	**8ος Χρήστης**
1400	114	92	102	104	107	84	103	101
2000	119	122	119	121	132	89	125	105
2600	124	141	126	135	148	119	129	117
3200	99	149	138	152	166	146	126	129
3800	153	169	138	150	180	166	136	135
4400	142	169	138	154	182	168	142	135
5000	185	164	138	171	180	195	142	135
5600	182	185	138	175	182	186	142	135
6200	189	194	138	178	175	189	142	135
6800	193	190	138	177	182	193	142	135
7400	197	189	138	176	194	189	142	135
8000	198	189	138	176	184	192	142	135
Προϋπολο-γισμός	**9ος Χρήστης**	**10ος Χρήστης**	**11ος Χρήστης**	**12ος Χρήστης**	**13ος Χρήστης**	**14ος Χρήστης**	**15ος Χρήστης**	**16ος Χρήστης**
1400	90	100	26	26	26	105	26	90
2000	112	122	26	26	26	130	26	120
2600	134	131	26	26	26	144	26	116
3200	160	136	26	26	26	162	26	131
3800	134	142	26	26	26	178	26	152
4400	171	142	26	26	26	175	26	144
5000	161	142	27	26	26	172	26	152
5600	167	142	58	58	26	168	58	152
6200	174	142	58	58	44	173	58	152
6800	178	142	58	58	44	168	58	152
7400	178	142	58	58	44	172	58	152
8000	179	142	58	58	44	175	58	152

Προϋπολο-γισμός	17ος Χρήστης	18ος Χρήστης	19ος Χρήστης	20ος Χρήστης	21ος Χρήστης	22ος Χρήστης	23ος Χρήστης
1400	26	97	84	109	26	50	109
2000	26	119	84	127	26	122	131
2600	26	122	140	149	26	141	148
3200	26	161	160	151	26	166	170
3800	26	114	163	186	26	106	172
4400	26	114	177	172	26	179	179
5000	26	121	156	189	26	172	169
5600	26	126	156	221	58	187	161
6200	26	126	152	215	58	181	156
6800	26	134	155	215	58	193	156
7400	26	134	155	213	58	187	156
8000	26	140	155	217	58	189	156

Πίνακας 5.13 Αριθμός Gridlet που εκτελούνται για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν συμμετέχουν 23 καταναλωτές πόρων στο πλέγμα.



Σχήμα 5.5 Αριθμός Gridlet που εκτελούνται για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν συμμετέχουν 23 καταναλωτές πόρων στο πλέγμα.

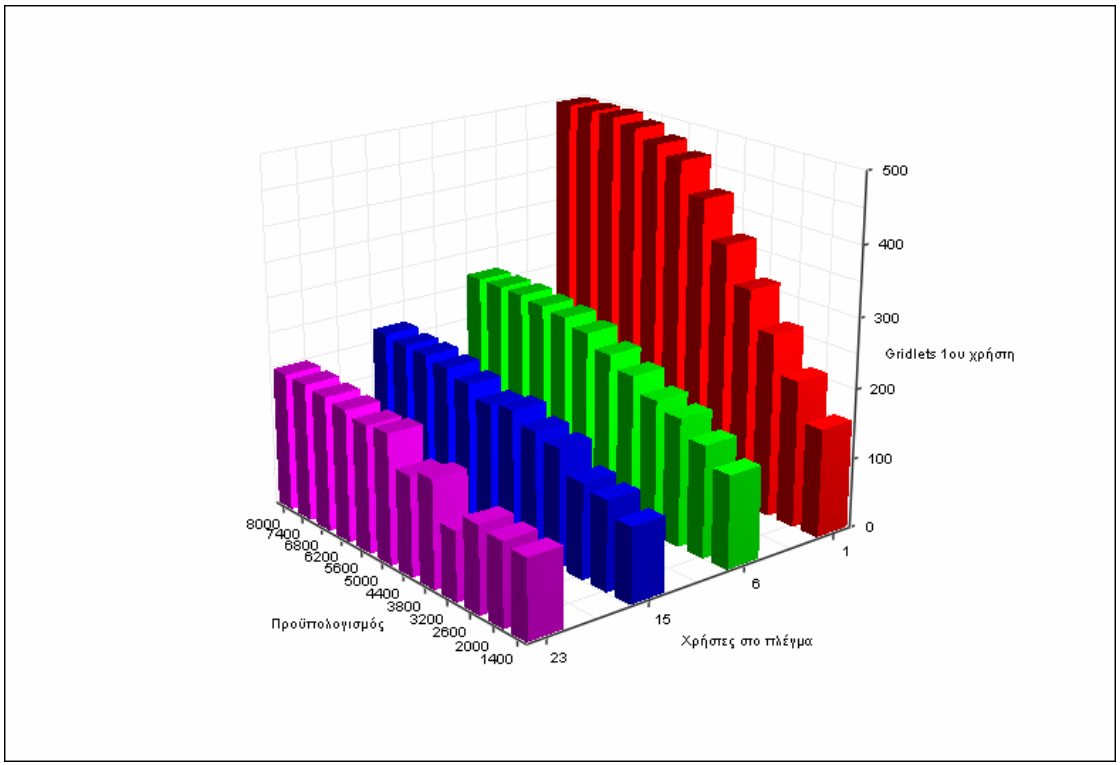
5.2.2.4 Σύγκριση απόδοσης για τον 1ο χρήστη,όταν μετέχουν και άλλοι χρήστες στο πλέγμα.

Εξετάζουμε τον αριθμό των Gridlet που μπορεί να εκτελέσει ο χρήστης για διάφορα ποσά προϋπολογισμού όταν είναι ο μόνος χρήστης στο πλέγμα σε σύγκριση με τα σενάρια όπου συναγωνίζεται άλλους 5, 14, 22 χρήστες.

Προθεσμία 360	**1 χρήστης στο**	**6 χρήστες στο**	**15 χρήστες στο**	**23 χρήστες στο**

	πλέγμα	πλέγμα	πλέγμα	πλέγμα
Προϋπολο-γισμός	Gridlets που εκτελεί ο 1ος χρήστης			
1400	156	134	108	114
2000	210	160	128	119
2600	263	183	136	124
3200	314	194	168	99
3800	366	215	176	153
4400	420	232	191	142
5000	464	250	185	185
5600	478	260	200	182
6200	491	264	210	189
6800	498	268	215	193
7400	498	268	216	197
8000	498	268	220	198

Πίνακες 5.14 Gridlets που εκτελεί ο 1ος χρήστης για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν στο πλέγμα συμμετέχουν 1, 6 ,15, 23 καταναλωτές πόρων



Σχήμα 5.6 Gridlets που εκτελεί ο 1ος χρήστης για διαφορετικό κάθε φορά όριο προϋπολογισμού, όταν στο πλέγμα συμμετέχουν 1, 6 ,15, 23 καταναλωτές πόρων

Στο διάγραμμα παρατηρούμε ότι όσο αυξάνεται ο αριθμός των χρηστών που συναγωνίζονται μέσα στο ίδιο υπολογιστικό πλέγμα για τους ίδιους πόρους, τόσο ο αριθμός των Gridlets που τελικά εκτελούνται για τον καθένα ελαττώνεται.

Άλλος ένας παράγοντας που πρέπει να εξετάσουμε είναι ο χρόνος ολοκλήρωσης επεξεργασίας Gridlet για κάθε χρήστη και πόσο αυτός υπερβαίνει την καθορισμένη προθεσμία. Από τα αποτελέσματα της προσομοίωσης έχουμε για 1, 6, 15, 23 χρήστες τους παρακάτω πίνακες:

Προθε-σμία 360	**1 χρήστης στο πλέγμα**	**6 χρήστες στο πλέγμα**						
	Χρόνος παύσης εκτέλεσης Gridlets	**Χρόνος παύσης εκτέλεσης Gridlets για κάθε χρήστη**						
Προϋπολογισμός	**1ος Χρήστης**	**1ος**	**2ος**	**3ος**	**4ος**	**5ος**	**6ος**	**ΜΕΣΟΣ ΟΡΟΣ**
1400	361	467	463	466	467	464	465	465
2000	360	503	505	503	496	505	507	502
2600	377	503	503	501	504	506	505	503
3200	360	514	516	510	511	493	492	509
3800	360	487	487	485	487	488	490	487
4400	356	468	470	464	469	463	465	467
5000	408	466	466	462	468	473	471	467
5600	412	469	471	464	466	471	473	468
6200	422	467	470	462	471	470	473	468
6800	428	467	469	461	473	470	472	468
7400	428	466	470	461	471	469	472	467
8000	428	466	470	461	471	469	472	467

Πίνακας 5.15 Χρόνος ολοκλήρωσης επεξεργασίας Gridlet για κάθε χρήστη, όταν συμμετέχουν 1 και 6 καταναλωτές πόρων στο πλέγμα, για διαφορετικά όρια προϋπολογισμού.

Προθεσμία 360	**15 χρήστες στο πλέγμα**															
	Χρόνος παύσης εκτέλεσης Gridlets για κάθε χρήστη															
Προϋπολο-γισμός	**1ος**	**2ος**	**3ος**	**4ος**	**5ος**	**6ος**	**7ος**	**8ος**	**9ος**	**10ος**	**11ος**	**12ος**	**13ος**	**14ος**	**15ος**	**ΜΕΣΟΣ ΟΡΟΣ**
1400	685	609	540	605	653	667	490	670	607	691	625	624	677	643	625	628
2000	730	744	720	737	735	736	713	735	698	745	735	736	730	738	732	731
2600	758	739	737	764	740	739	727	732	671	769	756	751	738	741	754	740
3200	744	723	743	742	746	749	742	568	708	743	752	752	594	746	754	718
3800	740	704	536	740	741	743	742	580	738	735	733	701	604	743	715	699
4400	721	715	537	718	719	716	706	582	708	720	723	690	605	722	708	684
5000	710	669	546	697	720	720	704	587	714	694	723	696	604	721	711	679
5600	716	713	537	719	711	714	721	669	709	683	728	701	676	708	718	693
6200	688	703	537	721	717	711	720	668	709	695	727	700	666	709	716	691
6800	687	702	537	717	714	713	705	668	708	694	726	699	666	707	716	689
7400	686	701	537	717	712	709	705	668	710	692	725	698	666	709	716	688
8000	685	703	537	717	712	709	705	668	710	693	725	700	666	709	716	689

Πίνακας 5.16 Χρόνος ολοκλήρωσης επεξεργασίας Gridlet για κάθε χρήστη, όταν συμμετέχουν 15 καταναλωτές πόρων στο πλέγμα, για διαφορετικά όρια προϋπολογισμού.

Προθε-σμία 360	**23 χρήστες στο πλέγμα**																						
	Χρόνος παύσης εκτέλεσης Gridlets για κάθε χρήστη																						
Προϋπολο-γισμός	**1ος**	**2ος**	**3ος**	**4ος**	**5ος**	**6ος**	**7ος**	**8ος**	**9ος**	**10ος**	**11ος**	**12ος**	**13ος**	**14ος**	**15ος**	**16ος**	**17ος**	**18ος**	**19ος**	**20ος**	**21ος**	**22ος**	**23ος**	**ΜΕΣΟΣ ΟΡΟΣ**
1400	776	782	782	768	737	776	784	779	765	782	519	497	417	743	504	750	429	780	769	764	532	693	744	690
2000	870	821	800	868	792	832	800	867	809	789	519	497	417	788	504	867	429	783	769	856	532	774	798	730
2600	844	913	898	915	798	862	868	902	829	840	519	497	417	799	504	913	429	803	868	815	532	795	796	755
3200	856	887	787	887	424	859	886	862	430	884	893	836	884	881	830	892	886	878	845	891	886	881	806	828
3800	868	830	879	822	840	804	884	882	823	881	519	497	417	780	504	817	429	877	828	808	532	772	807	743
4400	794	825	866	789	793	793	869	864	795	862	519	497	417	790	504	823	429	866	781	818	532	794	791	731
5000	779	795	853	786	797	788	862	858	794	855	710	497	417	798	504	817	429	847	799	830	532	796	812	737
5600	916	909	866	918	911	915	837	911	909	827	879	857	417	915	868	813	429	820	766	922	875	919	918	840
6200	931	924	875	931	926	928	827	926	923	835	897	870	747	932	881	814	429	830	764	920	888	933	931	864
6800	931	924	881	932	931	929	827	925	924	835	898	873	747	932	885	814	429	830	775	935	890	935	931	866
7400	930	924	881	932	930	931	827	925	923	835	898	873	747	932	884	814	429	830	778	935	891	935	931	866
8000	929	923	881	932	930	928	827	924	923	835	898	873	747	932	884	814	429	830	777	935	891	935	931	866

Πίνακας 5.17 Χρόνος ολοκλήρωσης επεξεργασίας Gridlet για κάθε χρήστη, όταν συμμετέχουν 23 καταναλωτές πόρων στο πλέγμα, για διαφορετικά όρια προϋπολογισμού.

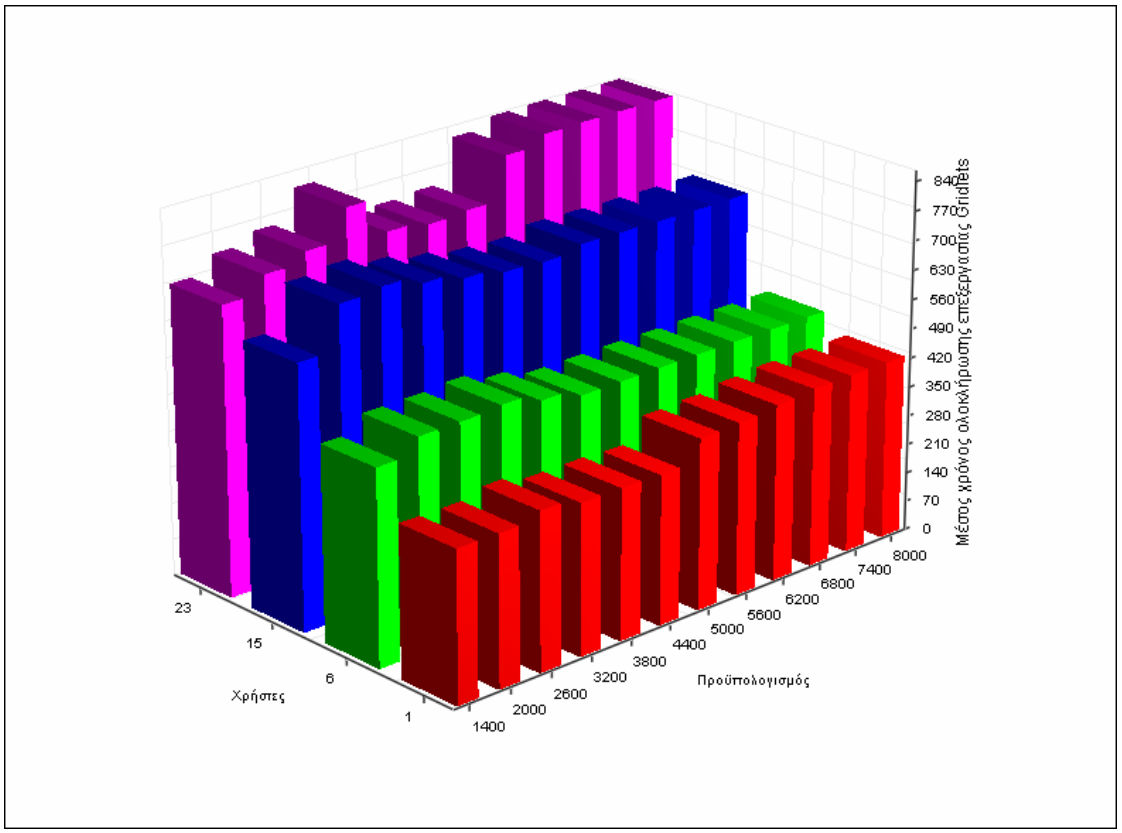
Υπολογίζοντας το μέσο χρόνο ολοκλήρωσης επεξεργασίας Gridlet κάθε σεναρίου, για διάφορες τιμές προϋπολογισμού προκύπτουν τα παρακάτω:

Προθεσμία 360	**Μέσος χρόνος ολοκλήρωσης επεξεργασίας Gridlets**			
	Χρήστες που συμμετέχουν			
Προϋπολογισμός	**1**	**6**	**15**	**23**
1400	361	465	628	690
2000	360	502	731	730
2600	377	503	740	755
3200	360	509	718	828
3800	360	487	699	743
4400	356	467	684	731
5000	408	467	679	737
5600	412	468	693	840
6200	422	468	691	864
6800	428	468	689	866
7400	428	467	688	866
8000	428	467	689	866

Πίνακας 5.18 Μέσος χρόνος ολοκλήρωσης επεξεργασίας Gridlet, για διαφορετικά όρια προϋπολογισμού, όταν συμμετέχουν 1, 6, 15, 23 καταναλωτές πόρων στο πλέγμα

Όπως φαίνεται και στο σχήμα που ακολουθεί όταν περισσότεροι από ένας χρήστες συμμετέχουν στο πλέγμα και διεκδικούν πόρους η προθεσμία εξαντλείται και καθυστερεί πολύ η ολοκλήρωση. Αυτό συμβαίνει γιατί αρχικά κάθε μεσίτης σχεδιάζει την δέσμευση πόρων που θα κάνει μέσα στα όρια της προθεσμίας. Όταν όμως έρχεται η χρονική στιγμή να δεσμεύσει ένα πόρο συμβαίνει άλλος χρήστης να τον έχει δεσμεύσει οπότε πρέπει να αναπροσαρμόσει την στρατηγική του με βάση το νέο χάρτη διαθέσιμων πόρων. Αυτή η διαδικασία είναι αναπόφευκτη και είναι ο λόγος υπέρβασης της προθεσμίας.

Σε αντίστοιχες περιπτώσεις όπου θα ακολουθείτο στρατηγικής βελτιστοποίησης χρόνου ο μεσίτης δεν θα συνέχιζε την εκτέλεση πέρα από την προθεσμία θα επεξεργάζονταν λιγότερα Gridlets και θα έμμενε ανεκμετάλλευτο μέρος του προϋπολογισμού.



Σχήμα 5.7 Μέσος χρόνος ολοκλήρωσης επεξεργασίας Gridlet, για διαφορετικά όρια προϋπολογισμού, όταν συμμετέχουν 1, 6, 15, 23 καταναλωτές πόρων στο πλέγμα

6

Επίλογος

Η εξάπλωση της χρήσης υπολογιστικών πλεγμάτων σε ευρεία κλίμακα δείχνει να πλησιάζει όλο και πιο πολύ. Η έξαρση του προγραμματιστικού φαινομένου γνωστού ως «Web 2.0» που αποτελεί την πιο δημοφιλή τάση της εποχής, μας δίνει να κατανοήσουμε πόσο σημαντική είναι η παροχή τεχνολογικών δυνατοτήτων με τη μορφή υπηρεσιών μέσω του διαδικτύου. Αν θεωρήσουμε ότι το «Web 2.0» είναι το μέλλον της ανάπτυξης εφαρμογών στο διαδίκτυο, τότε τα υπολογιστικά πλέγματα είναι η πλατφόρμα πάνω στην οποία θα αποκαλυφθούν οι δυνατότητες του. Υπολογιστικά πλέγματα που θα εστιάζουν αποκλειστικά στην υπό μορφή υπηρεσιών, παροχή υπολογιστικής δύναμης, αποθηκευτικού χώρου και άλλων πόρων.

Από την έρευνα που πραγματοποιήθηκε στην παραπάνω διπλωματική συμπεραίνουμε πως η χρήση οικονομικών μοντέλων για την διαχείριση πόρων πλέγματος είναι ιδανική για προσανατολισμένα στην παροχή υπηρεσιών Υπολογιστικά Πλέγματα.
Η δημιουργία ενός περιβάλλοντος Grid-Οικονομίας όπου οι ιδιοκτήτες των πόρων διαθέτουν ως υπηρεσία (πωλούν) τη χρήση των πόρων τους και οι χρήστες πληρώνουν για την πρόσβαση σε αυτούς, με βάση το είδος τους και το χρόνο χρήσης, μπορεί να δώσει ώθηση στην εδραίωση των υπολογιστικών πλεγμάτων και στην πλήρη εκμετάλλευση των δυνατοτήτων τους.

Η ανάλυση που έγινε για τα ποικίλα οικονομικά μοντέλα διαχείρισης πόρων, τον τρόπο λειτουργίας και τα χαρακτηριστικά τους, μπορεί να μας οδηγήσει στο πιο μοντέλο πρέπει να επιλεχθεί στην υλοποίηση μιας αρχιτεκτονικής πλέγματος ανάλογα με το ποιοι συμμετέχουν, ποιοι είναι οι στόχοι τους, ποιο το είδος των πόρων που χρησιμοποιούνται από κοινού.

Μελλοντικά μπορούν να πραγματοποιηθούν προσομοιώσεις πραγματικών υπολογιστικών πλεγμάτων και να εξεταστεί η συμπεριφορά τους για διάφορα οικονομικά μοντέλα. Ο αριθμός των χρηστών, οι απαιτήσεις τους και η διαθεσιμότητα των πόρων μπορούν να εκφραστούν βάσει πιθανοτήτων και να θεωρηθεί ότι ακολουθούν συγκεκριμένες κατανομές. Τα αποτελέσματα τέτοιων προσομοιώσεων μπορούν να χρησιμοποιηθούν για την υλοποίηση στον πραγματικό κόσμο, Υπολογιστικών Πλεγμάτων - Grid που εστιάζουν στην παροχή υπηρεσιών και λειτουργούν με βάση οικονομικές αρχές.

7

Βιβλιογραφία

[1]	Foster, I., *What is the Grid? A Three Point Checklist*, July 2002.
[2]	Plaszczak, Wellner, Jr., *Grid Computing*, 2005
[3]	IBM, *IBM Solutions Grid for Business Partners: Helping IBM Business Partners to Grid-enable applications for the next phase of e-business on demand, 2003,www.ibm.com*
[4]	F. J. Corbató, R. M. Graham, V. A. Vyssotsky, *Structure of the Multics Supervisor*, http://www.multicians.org/fjcc3.html
[5]	Buyya Rajkumar, Venugopal Srikumar, *A Gentle Introduction to Grid Computing and Technologies,* 2005
[6]	*Cern GridCafe*, http://gridcafe.web.cern.ch
[7]	*The Globus Alliance*, http://www.globus.org/
[8]	Foster, I., Kesselman, C., Tuecke, St. *The Anatomy of the Grid*, 2001.
[9]	Schnizler B., Neumann D., Weinhardt C., *Resource Allocation in Computational Grids-A Market Engineering Approach*, 2004
[10]	R Wolski, J Brevik, JS Plank, T Bryan, *Grid resource allocation and control using computational economies,* 2004

[11]	Roger Smith, *Grid Computing: A Brief Technology Analysis*, 2004, www.ctonet.org
[12]	Viktors Berstis, *Fundamentals of Grid Computing*, 2002, http://www.redbooks.ibm.com
[13]	Clabby Analytics, *The Grid Report*, 2004 Edition
[14]	Κοζύρης Ν, Πολυμενάκος Λ., Τεχνολογίες Πλέγματος (Grid) Προοπτικές και Εφαρμογές στο Η-Επιχειρείν, 2005
[15]	*Seti@Home Project* , http://setiathome.berkeley.edu
[16]	*IBM Grid computing*, http://www-1.ibm.com/grid
[17]	*BOINC - Berkeley Open Infrastructure for Network Computing, http://boinc.berkeley.edu*
[18]	*Great Internet Mersenne Prime Search (GIMPS),* http://www.mersenne.org/
[19]	Rajkumar Buyya, David Abramson, Jonathan Giddy, Heinz Stockinger, *Economic Models for Resource Management and Scheduling in Grid Computing, 2003*
[20]	Derrick Harris, *Goodbye GGF and EGA. Hello OGF,* Gridtoday5(26)
[21]	*NASA Information Power Grid,* http://www.ipg.nasa.gov
[22]	*World Community Grid, www.worldcommunitygrid.org*
[23]	*Johnson & Johnson*, http://www.jnj.com
[24]	*Novartis AG* , http://www.novartis.com/
[25]	*FortuneMagazine ,* http://money.cnn.com/magazines/fortune/
[26]	*Wachovia Corporation,* www.wachovia.com
[27]	*JPMorgan Chase & Co,* www.jpmorgan.com
[28]	*Shell,* www.shell.com
[29]	*BP,* www.bp.com
[30]	*HSBC,* www.hsbc.co.uk
[31]	*HSBC Powers Up Grid Computing Push,* BusinessWeek , June 2006
[32]	*Deutsche Bank,* www.db.com
[33]	*Boeing Company*, www.boeing.com

[34]	*Airbus*, www.airbus.com
[35]	*Global Grid Forum (GGF),* www.globalgridforum.org
[36]	*EGA (Enterprise Grid Alliance),* http://www.gridalliance.org
[37]	http://en.wikipedia.org/wiki/Auction
[38]	K. Aida, A. Takefusa, H. Nakada, S. Matsuoka, S. Sekiguchi, and U. Nagashima, *Performance evaluation model for scheduling in a global computing system, The International Journal of High Performance Computing Applications*, Vol. 14, No. 3,SagePublications,USA,2000.
[39]	H. Song, X. Liu, D. Jakobsen, R. Bhagwan, X. Zhang, K. Taura, and A. Chien, *The MicroGrid: A Scientific Tool for Modeling Computational Grids*, Proceedings of IEEE Supercomputing (SC 2000), Nov. 4-10, 2000, Dallas, USA.
[40]	H. Casanova, Simgrid: *A Toolkit for the Simulation of Application Scheduling*, Proceedings of the First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001), May 2001, Brisbane, Australia
[41]	*GridSim: A Grid Simulation Toolkit for Resource Modelling and Application Scheduling for Parallel and Distributed Computing,* The Gridbus Project, GRIDS Lab, The University of Melbourne, 2006
[42]	*SPEC CINT2000 Results Published by SPEC, http://www.spec.org/cpu2000/results/cint2000.html*

8

ΠΑΡΑΡΤΗΜΑ. Κώδικας

Παρατίθεται ενδεικτικά ο κώδικας για την προσομοίωση με το Gridsim του σεναρίου όπου συμμετέχουν 6 καταναλωτές πόρων στο πλέγμα.

```
// Prosomoiosi upologistikou plegmatos me 6 xristes na sunagwnizontai
// gia ti xrisi porwn.
//

import java.util.*;
import gridsim.*;
import gridbroker.*;

public class main
{

    public static void main(String[] args)
    {

        try
        {
```

```
            Calendar calendar = Calendar.getInstance();
            Random random;
            long seed = 11L*13*17*19*23+1;
            seed = seed * 97 + 1;

            String[] eff = {""};
            String[] efp = {""};
            String[] category = { "*.USER.GridletCompletion",
                "*.USER.TimeUtilization",   "*.USER.BudgetUtilization"
};

            String ReportWriterName = "ReportWritermain";

            // Arxikopoisi tou gridsim gia 6 xristes kai 10
resources
            GridSim.init(6,     calendar,     true,     eff,     efp,
ReportWriterName);

            // dimoiourgei tin anafora me ta apotelesmata tis
prosomoiosis
            new      ReportWriter(ReportWriterName,      6,      10,
"reportFilemain",
                category, false, null, true, 10000);


            // Arxi dimiourgias twn Entities
            LinkedList Weekends = new LinkedList();
            Weekends.add(new Integer(Calendar.SATURDAY));
            Weekends.add(new Integer(Calendar.SUNDAY));
            LinkedList Holidays = new LinkedList();

            PEList peList;                    // Lista twn monadwn
epeksergasias
            MachineList mList;        // Lista mixanimatwn
            ResourceCharacteristics resConfig;
            GridResource gridRes;     // Mia ontotita GridResource
```

```
            String[] resourceNameList = {
                "Resource_1",       "Resource_2",       "Resource_3",
"Resource_4",
                "Resource_5",       "Resource_6",       "Resource_7",
"Resource_8",
                "Resource_9",  "Resource_10",
            };

            /////////  Resource_1
            mList = new MachineList();    // Lista Mixanimatwn

            peList = new PEList();      // Lista ME
            peList.add( new PE(0, 1836) );
            peList.add( new PE(1, 1836) );
            peList.add( new PE(2, 1836) );
            peList.add( new PE(3, 1836) );
            peList.add( new PE(4, 1836) );
            peList.add( new PE(5, 1836) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(0, peList) );


            peList = new PEList();      // Lista ME
            peList.add( new PE(0, 1836) );
            peList.add( new PE(1, 1836) );
            peList.add( new PE(2, 1836) );
            peList.add( new PE(3, 1836) );
            peList.add( new PE(4, 1836) );
            peList.add( new PE(5, 1836) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(1, peList) );
```

```
            // Dimiourgia neas ontotitas xaraktiristikwn porou
            resConfig = new ResourceCharacteristics("Sun Ultra 40",
"",
                mList, 0, 0.0, 7.0);

            // Dimiourgia neas ontotitas porou
            gridRes = new GridResource("Resource_1", 5500.0,
seed*(1+1)+1,
                resConfig, 0.5, 0.3, 0.1, Weekends, Holidays);

            ///////// Resource_2
            mList = new MachineList();    // Lista Mixanimatwn

            peList = new PEList();      // Lista ME
            peList.add( new PE(0, 1862) );
            peList.add( new PE(1, 1862) );
            peList.add( new PE(2, 1862) );
            peList.add( new PE(3, 1862) );
            peList.add( new PE(4, 1862) );
            peList.add( new PE(5, 1862) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(0, peList) );


            // Dimiourgia neas ontotitas xaraktiristikwn porou
            resConfig = new ResourceCharacteristics("AMD Athlon (TM)
64 FX-57", "",
                mList, 0, 1.0, 8.0);

            // Dimiourgia neas ontotitas porou
            gridRes = new GridResource("Resource_2", 6700.0,
seed*(2+1)+1,
                resConfig, 0.6, 0.3, 0.0, Weekends, Holidays);
```

```
//////// Resource_3
mList = new MachineList();    // Lista Mixanimatwn

peList = new PEList();      // Lista ME
peList.add( new PE(0, 1833) );
peList.add( new PE(1, 1833) );
peList.add( new PE(2, 1833) );
peList.add( new PE(3, 1833) );

// Prosthiki mixanimatos sti lista
mList.add( new Machine(0, peList) );


peList = new PEList();      // Lista ME
peList.add( new PE(0, 1833) );
peList.add( new PE(1, 1833) );
peList.add( new PE(2, 1833) );
peList.add( new PE(3, 1833) );

// Prosthiki mixanimatos sti lista
mList.add( new Machine(1, peList) );


peList = new PEList();      // Lista ME
peList.add( new PE(0, 1833) );
peList.add( new PE(1, 1833) );
peList.add( new PE(2, 1833) );
peList.add( new PE(3, 1833) );

// Prosthiki mixanimatos sti lista
mList.add( new Machine(2, peList) );


// Dimiourgia neas ontotitas xaraktiristikwn porou
```

```
            resConfig    =    new    ResourceCharacteristics("Intel(R)
Pentium(R) 3.73 GHz", "",
                mList, 0, 1.0, 3.0);

            // Dimiourgia neas ontotitas porou
            gridRes    =    new    GridResource("Resource_3",    7000.0,
seed*(3+1)+1,
                resConfig, 0.8, 0.3, 0.0, Weekends, Holidays);

            ///////// Resource_4
            mList = new MachineList();    // Lista Mixanimatwn

            peList = new PEList();      // Lista ME
            peList.add( new PE(0, 1810) );
            peList.add( new PE(1, 1810) );
            peList.add( new PE(2, 1810) );
            peList.add( new PE(3, 1810) );
            peList.add( new PE(4, 1810) );
            peList.add( new PE(5, 1810) );
            peList.add( new PE(6, 1810) );
            peList.add( new PE(7, 1810) );
            peList.add( new PE(8, 1810) );
            peList.add( new PE(9, 1810) );
            peList.add( new PE(10, 1810) );
            peList.add( new PE(11, 1810) );
            peList.add( new PE(12, 1810) );
            peList.add( new PE(13, 1810) );
            peList.add( new PE(14, 1810) );
            peList.add( new PE(15, 1810) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(0, peList) );


            peList = new PEList();      // Lista ME
```

```
            peList.add( new PE(0, 1810) );
            peList.add( new PE(1, 1810) );
            peList.add( new PE(2, 1810) );
            peList.add( new PE(3, 1810) );
            peList.add( new PE(4, 1810) );
            peList.add( new PE(5, 1810) );
            peList.add( new PE(6, 1810) );
            peList.add( new PE(7, 1810) );
            peList.add( new PE(8, 1810) );
            peList.add( new PE(9, 1810) );
            peList.add( new PE(10, 1810) );
            peList.add( new PE(11, 1810) );
            peList.add( new PE(12, 1810) );
            peList.add( new PE(13, 1810) );
            peList.add( new PE(14, 1810) );
            peList.add( new PE(15, 1810) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(1, peList) );


            // Dimiourgia neas ontotitas xaraktiristikwn porou
            resConfig = new ResourceCharacteristics("Intel 3.8 GHz
Xeon, 2MB L2 Cache", "",
                mList, 0, 9.0, 2.0);

            // Dimiourgia neas ontotitas porou
            gridRes = new GridResource("Resource_4", 7500.0,
seed*(4+1)+1,
                resConfig, 0.8, 0.5, 0.1, Weekends, Holidays);

            ///////// Resource_5
            mList = new MachineList();    // Lista Mixanimatwn

            peList = new PEList();      // Lista ME
```

```
            peList.add( new PE(0, 1836) );
            peList.add( new PE(1, 1836) );
            peList.add( new PE(2, 1836) );
            peList.add( new PE(3, 1836) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(0, peList) );


            // Dimiourgia neas ontotitas xaraktiristikwn porou
            resConfig = new ResourceCharacteristics("Sun Ultra 40",
"",
                mList, 0, 1.0, 6.0);

            // Dimiourgia neas ontotitas porou
            gridRes = new GridResource("Resource_5", 5500.0,
seed*(5+1)+1,
                resConfig, 0.5, 0.2, 0.0, Weekends, Holidays);

            ///////// Resource_6
            mList = new MachineList();    // Lista Mixanimatwn

            peList = new PEList();      // Lista ME
            peList.add( new PE(0, 1836) );
            peList.add( new PE(1, 1836) );
            peList.add( new PE(2, 1836) );
            peList.add( new PE(3, 1836) );
            peList.add( new PE(4, 1836) );
            peList.add( new PE(5, 1836) );
            peList.add( new PE(6, 1836) );
            peList.add( new PE(7, 1836) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(0, peList) );
```

```
            // Dimiourgia neas ontotitas xaraktiristikwn porou
            resConfig = new ResourceCharacteristics("Sun Ultra 40",
"",
                mList, 0, 8.0, 7.0);

            // Dimiourgia neas ontotitas porou
            gridRes = new GridResource("Resource_6", 7000.0,
seed*(6+1)+1,
                resConfig, 0.5, 0.3, 0.1, Weekends, Holidays);

            ///////// Resource_7
            mList = new MachineList();    // Lista Mixanimatwn

            peList = new PEList();      // Lista ME
            peList.add( new PE(0, 1836) );
            peList.add( new PE(1, 1836) );
            peList.add( new PE(2, 1836) );
            peList.add( new PE(3, 1836) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(0, peList) );


            peList = new PEList();      // Lista ME
            peList.add( new PE(0, 1836) );
            peList.add( new PE(1, 1836) );
            peList.add( new PE(2, 1836) );
            peList.add( new PE(3, 1836) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(1, peList) );


            peList = new PEList();      // Lista ME
```

```
            peList.add( new PE(0, 1836) );
            peList.add( new PE(1, 1836) );
            peList.add( new PE(2, 1836) );
            peList.add( new PE(3, 1836) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(2, peList) );


            peList = new PEList();      // Lista ME
            peList.add( new PE(0, 1836) );
            peList.add( new PE(1, 1836) );
            peList.add( new PE(2, 1836) );
            peList.add( new PE(3, 1836) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(3, peList) );


            // Dimiourgia neas ontotitas xaraktiristikwn porou
            resConfig = new ResourceCharacteristics("Sun Ultra 40",
"",
                mList, 0, 6.0, 6.0);

            // Dimiourgia neas ontotitas porou
            gridRes   =   new   GridResource("Resource_7",   1400.0,
seed*(7+1)+1,
                resConfig, 0.4, 0.1, 0.0, Weekends, Holidays);

            ///////// Resource_8
            mList = new MachineList();    // Lista Mixanimatwn

            peList = new PEList();      // Lista ME
            peList.add( new PE(0, 1862) );
            peList.add( new PE(1, 1862) );
```

```
            peList.add( new PE(2, 1862) );
            peList.add( new PE(3, 1862) );
            peList.add( new PE(4, 1862) );
            peList.add( new PE(5, 1862) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(0, peList) );



            // Dimiourgia neas ontotitas xaraktiristikwn porou
            resConfig = new ResourceCharacteristics("AMD Athlon (TM)
64 FX-57", "",
                mList, 0, 0.0, 8.0);

            // Dimiourgia neas ontotitas porou
            gridRes   =   new   GridResource("Resource_8",   6500.0,
seed*(8+1)+1,
                resConfig, 0.7, 0.4, 0.2, Weekends, Holidays);

            ///////// Resource_9
            mList = new MachineList();    // Lista Mixanimatwn

            peList = new PEList();      // Lista ME
            peList.add( new PE(0, 1833) );
            peList.add( new PE(1, 1833) );
            peList.add( new PE(2, 1833) );
            peList.add( new PE(3, 1833) );
            peList.add( new PE(4, 1833) );
            peList.add( new PE(5, 1833) );
            peList.add( new PE(6, 1833) );
            peList.add( new PE(7, 1833) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(0, peList) );
```

```
            peList = new PEList();      // Lista ME
            peList.add( new PE(0, 1833) );
            peList.add( new PE(1, 1833) );
            peList.add( new PE(2, 1833) );
            peList.add( new PE(3, 1833) );
            peList.add( new PE(4, 1833) );
            peList.add( new PE(5, 1833) );
            peList.add( new PE(6, 1833) );
            peList.add( new PE(7, 1833) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(1, peList) );


            peList = new PEList();      // Lista ME
            peList.add( new PE(0, 1833) );
            peList.add( new PE(1, 1833) );
            peList.add( new PE(2, 1833) );
            peList.add( new PE(3, 1833) );
            peList.add( new PE(4, 1833) );
            peList.add( new PE(5, 1833) );
            peList.add( new PE(6, 1833) );
            peList.add( new PE(7, 1833) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(2, peList) );


            // Dimiourgia neas ontotitas xaraktiristikwn porou
            resConfig    =    new    ResourceCharacteristics("Intel(R)
Pentium(R) 3.73 GHz", "",
                mList, 0, 2.0, 2.0);

            // Dimiourgia neas ontotitas porou
```

```
            gridRes = new GridResource("Resource_9", 5000.0,
seed*(9+1)+1,
                resConfig, 0.6, 0.2, 0.0, Weekends, Holidays);

            ///////// Resource_10
            mList = new MachineList();    // Lista Mixanimatwn

            peList = new PEList();      // Lista ME
            peList.add( new PE(0, 1810) );
            peList.add( new PE(1, 1810) );

            // Prosthiki mixanimatos sti lista
            mList.add( new Machine(0, peList) );



            // Dimiourgia neas ontotitas xaraktiristikwn porou
            resConfig = new ResourceCharacteristics("Intel 3.8 GHz
Xeon, 2MB L2 Cache", "",
                mList, 0, 1.0, 1.0);

            // Dimiourgia neas ontotitas porou
            gridRes = new GridResource("Resource_10", 6500.0,
seed*(10+1)+1,
                resConfig, 0.5, 0.1, 0.0, Weekends, Holidays);



            // Dimiourgia ontotitwn xristwn
            GridletList glList;      // Lista Gridlets-Ergasiwn
            Experiment expt;
            UserEntity userEntity;
            int count = 0;

            ///////// User_1
            random = new Random(seed*997*(1+1)+1);
            glList = new GridletList();
```

```
            count = (int) GridSimRandom.real(500, 0.0, 0.051,
random.nextDouble());

            // loop gia dimiourgia newn gridlets
            for (int i = 0; i < count; i++) {
                double len = GridSimRandom.real(10000, 0.032, 0.083,
random.nextDouble());

                long file = (long) GridSimRandom.real(1000, 0.05,
0.071, random.nextDouble());

                long out = (long) GridSimRandom.real(1000,
0.044000000000000004, 0.06, random.nextDouble());

                // Dimiourgia neas ontotitas ergasias-gridlet
                Gridlet gl = new Gridlet(i, len, file, out);
                glList.add(gl);   // prosthiki tou gridlet sti lista
            }

            expt = new Experiment(1, glList, 1, false, 360.0, 3200.0,
                "report1.txt", resourceNameList);

            userEntity = new UserEntity("User1", expt, 1270.0,
seed*997*(1+1)+1,
                36.0, false);

            ///////// User_2
            random = new Random(seed*997*(1+2)+1);
            glList = new GridletList();
            count = (int) GridSimRandom.real(500, 0.078,
0.028999999999999998, random.nextDouble());

            // loop gia dimiourgia newn gridlets
            for (int i = 0; i < count; i++) {
                double len = GridSimRandom.real(10000, 0.071,
0.08900000000000001, random.nextDouble());
```

```
                long file = (long) GridSimRandom.real(1000, 0.086,
0.049, random.nextDouble());

                long out = (long) GridSimRandom.real(1000, 0.083,
0.053, random.nextDouble());

                // Dimiourgia neas ontotitas ergasias-gridlet
                Gridlet gl = new Gridlet(i, len, file, out);
                glList.add(gl);   // prosthiki tou gridlet sti lista
            }

            expt = new Experiment(2, glList, 1, false, 360.0, 3200.0,
                "report2.txt", resourceNameList);

            userEntity = new UserEntity("User2", expt, 1270.0,
seed*997*(1+2)+1,
                36.0, false);

            ///////// User_3
            random = new Random(seed*997*(1+3)+1);
            glList = new GridletList();
            count = (int) GridSimRandom.real(500, 0.091,
0.055999999999999994, random.nextDouble());

            // loop gia dimiourgia newn gridlets
            for (int i = 0; i < count; i++) {
                double len = GridSimRandom.real(10000,
0.08140000000000001, 0.07400000000000001, random.nextDouble());

                long file = (long) GridSimRandom.real(1000, 0.092,
0.08140000000000001, random.nextDouble());

                long out = (long) GridSimRandom.real(1000,
0.07200000000000001, 0.05, random.nextDouble());
```

```
                // Dimiourgia neas ontotitas ergasias-gridlet
                Gridlet gl = new Gridlet(i, len, file, out);
                glList.add(gl);   // prosthiki tou gridlet sti lista
            }

            expt = new Experiment(3, glList, 1, false, 360.0, 3200.0,
                "report3.txt", resourceNameList);

            userEntity = new UserEntity("User3", expt, 1270.0,
seed*997*(1+3)+1,
                36.0, false);

            ///////// User_4
            random = new Random(seed*997*(1+4)+1);
            glList = new GridletList();
            count = (int) GridSimRandom.real(500, 0.0, 0.096,
random.nextDouble());

            // loop gia dimiourgia newn gridlets
            for (int i = 0; i < count; i++) {
                double len = GridSimRandom.real(10000, 0.071, 0.062,
random.nextDouble());

                long file = (long) GridSimRandom.real(1000, 0.047,
0.037000000000000005, random.nextDouble());

                long out = (long) GridSimRandom.real(1000, 0.017,
0.065, random.nextDouble());

                // Dimiourgia neas ontotitas ergasias-gridlet
                Gridlet gl = new Gridlet(i, len, file, out);
                glList.add(gl);   // prosthiki tou gridlet sti lista
            }

            expt = new Experiment(4, glList, 0, false, 360.0, 3200.0,
                "report4.txt", resourceNameList);
```

```
            userEntity = new UserEntity("User4", expt, 1270.0,
seed*997*(1+4)+1,
                36.0, false);

            ///////// User_5
            random = new Random(seed*997*(1+5)+1);
            glList = new GridletList();
            count = (int) GridSimRandom.real(500, 0.098, 0.063,
random.nextDouble());

            // loop gia dimiourgia newn gridlets
            for (int i = 0; i < count; i++) {
                double len = GridSimRandom.real(10000,
0.027000000000000003, 0.035, random.nextDouble());

                long file = (long) GridSimRandom.real(1000, 0.015,
0.078, random.nextDouble());

                long out = (long) GridSimRandom.real(1000,
0.08900000000000001, 0.083, random.nextDouble());

                // Dimiourgia neas ontotitas ergasias-gridlet
                Gridlet gl = new Gridlet(i, len, file, out);
                glList.add(gl);   // prosthiki tou gridlet sti lista
            }

            expt = new Experiment(5, glList, 2, false, 360.0, 3200.0,
                "report5.txt", resourceNameList);

            userEntity = new UserEntity("User5", expt, 1270.0,
seed*997*(1+5)+1,
                36.0, false);

            ///////// User_6
            random = new Random(seed*997*(1+6)+1);
```

```
            glList = new GridletList();
            count = (int) GridSimRandom.real(500, 0.038, 0.077,
random.nextDouble());

            // loop gia dimiourgia newn gridlets
            for (int i = 0; i < count; i++) {
                double len = GridSimRandom.real(10000,
0.052000000000000005, 0.062, random.nextDouble());

                long file = (long) GridSimRandom.real(1000,
0.037000000000000005, 0.1, random.nextDouble());

                long out = (long) GridSimRandom.real(1000, 0.077,
0.066, random.nextDouble());

                // Dimiourgia neas ontotitas ergasias-gridlet
                Gridlet gl = new Gridlet(i, len, file, out);
                glList.add(gl);   // prosthiki tou gridlet sti lista
            }

            expt = new Experiment(6, glList, 2, false, 360.0, 3200.0,
                "report6.txt", resourceNameList);

            userEntity = new UserEntity("User6", expt, 1270.0,
seed*997*(1+6)+1,
                36.0, false);

            // ekkinisi tis prosomoiosis me to gridsim
            GridSim.startGridSimulation();
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```