



ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΑΝΕΡΓΟΦΟΡΙΚΗΣ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΔΙΠΛΩΜΑ ΕΙΔΙΚΕΥΣΗΣ (MSc)
στα ΠΑΝΕΡΓΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Διαχείριση Κατανεμημένης Εκτέλεσης Εφαρμογών
σε Υπολογιστικά Πλέγματα.

Τα Περιβάλλοντα
“Sun One Grid Engine”
& “Globus Toolkit”

Κουτουμάνος Ευάγγελος

M3020008

ΑΘΗΝΑ, ΦΕΒΡΟΥΑΡΙΟΣ 2004



ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΚΑΤΑΛΟΓΟΣ



0 000000 520959



ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ
ΒΙΒΛΙΟΘΗΚΗ
εισ. 76280
Αρ.
ταξ.

**ΜΕΤΑΠΤΥΧΙΑΚΟ ΔΙΠΛΩΜΑ ΕΙΔΙΚΕΥΣΗΣ (MSc)
στα ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Διαχείριση Κατανεμημένης Εκτέλεσης Εφαρμογών
σε Υπολογιστικά Πλέγματα.**

**Τα Περιβάλλοντα
“Sun One Grid Engine”
& “Globus Toolkit”**

Κουτουμάνος Ευάγγελος

M3020008



Επιβλέπων Καθηγητής: Θεόδωρος Αποστολόπουλος

Εξωτερικός Κριτής: Ιωάννης Μήλης

**ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

ΑΘΗΝΑ, ΦΕΒΡΟΙΟΥ ΑΡΙΟΣ 2004





Περίληψη Εργασίας

Η διπλωματική εργασία που παρατίθεται στη συνέχεια συντάχθηκε από τον Κουτουμάνο Ευάγγελο, φοιτητή του Μεταπτυχιακού Προγράμματος Σπουδών στα Πληροφοριακά Συστήματα, του τμήματος Πληροφορικής του Οικονομικού Πανεπιστημίου Αθηνών.

Επιβλέπων καθηγητής ήταν ο κος Θεόδωρος Αποστολόπουλος, καθηγητής του Οικονομικού Πανεπιστημίου.

Ο τίτλος της εργασίας είναι "Διαχείριση κατανεμημένης εκτέλεσης εφαρμογών σε υπολογιστικά πλέγματα. Τα περιβάλλοντα "Sun One Grid Engine" και "Globus Toolkit" ". Από τον τίτλο προκύπτει ότι σκοπό της εργασίας αποτελεί η μελέτη της λειτουργίας των περιβαλλόντων "Sun One Grid Engine" και "Globus Toolkit", η εγκατάστασή τους και η δημιουργίας κάποιας εφαρμογής που θα χρησιμοποιεί τους πόρους και τις δυνατότητες ενός υπολογιστικού πλέγματος. Για το σκοπό αυτό χρησιμοποιήθηκαν 3 υπολογιστές του Εργαστηρίου Συστημάτων Υπολογιστών και Επικοινωνιών του Οικονομικού Πανεπιστημίου Αθηνών.

Ο όρος υπολογιστικό πλέγμα διατυπώθηκε στα μέσα της δεκαετίας του 1990 με σκοπό να περιγράψει μια κατανεμημένη υπολογιστική υποδομή, η δημιουργία, της οποίας, απέβλεπε στην αξιοποίηση της σε υψηλού επιπέδου επιστημονικές έρευνες. Ωστόσο, σήμερα, χρησιμοποιείται, καταχρηστικά, για να περιγράψει πολύπλοκες υπολογιστικές υποδομές που αφορούν σε διάφορους τομείς από προηγμένα δίκτυα μέχρι τεχνητή νοημοσύνη.

Η πραγματική έννοια του όρου υπολογιστικό πλέγμα αντιστοιχεί σε μια εικονική μηχανή που σε φυσικό επίπεδο δομείται σε ένα δίκτυο γεωγραφικά κατανεμημένων υπολογιστικών πόρων.

Οι βασικές λειτουργίες ενός υπολογιστικού πλέγματος είναι οι εξής:

1. Εκμετάλλευση αναξιοποίητων υπολογιστικών πόρων
2. Παράλληλη αξιοποίηση επεξεργαστικής ισχύος
3. Υποστήριξη λειτουργίας εικονικών οργανισμών
4. Εξισορρόπηση φορτίου
5. Ταυτόχρονη διαχείριση διαφορετικών έργων

Οι βασικοί τύποι ενός υπολογιστικού πλέγματος ανάλογα με τον αριθμό των υπολογιστών που περιλαμβάνουν είναι οι εξής:

1. Καθολικό Υπολογιστικό Πλέγμα (Global Grid)
2. Επιχειρησιακό Υπολογιστικό Πλέγμα (Enterprise Grid)
3. Υπολογιστικό Πλέγμα Συστοιχίας (Cluster Grid)

Το κείμενο μας περιλαμβάνει ανάλυση του υπολογιστικού πλέγματος και του τρόπου λειτουργίας του.

Στο τμήμα που αφορά στα περιβάλλοντα υπολογιστικών πλεγμάτων με τα οποία ασχοληθήκαμε περιλαμβάνεται εκτενής αναφορά στον τρόπο λειτουργία τόσο του Sun One Grid Engine όσο και του Globus Toolkit.

To Sun One Grid Engine, όπως μαρτυρά και η ονομασία του, αποτελεί την πρόταση της εταιρίας Sun για την ανάπτυξη υπολογιστικών πλεγμάτων. Η αναφορά μας σ' αυτό περιέχει την επεξήγηση του τρόπου λειτουργίας και συνεργασίας των δομικών στοιχείων ενός πλέγματος που υλοποιείται με βάση το Sun One Grid Engine. Δηλαδή, αναλύεται ο τρόπος που οργανώνονται οι υπολογιστές ενός πλέγματος ανάλογα με το ρόλο που τους αποδίδεται (master, submit, administration ή execution host). Επίσης, παρουσιάζεται ο τρόπος διαχείρισης των ουρών στις οποίες οργανώνει το περιβάλλον τις εργασίες που υποβάλλονται σ' αυτό προς εκτέλεση.

Γίνεται αναφορά στην έκδοση Sun Grid Engine Enterprise Edition (SGEEE) που αποτελεί αναβαθμισμένη εκδοχή του Sun One Grid Engine η οποία χρησιμοποιείται σε υπολογιστικά πλέγματα επιχειρησιακά ή καθολικά.

Το Globus είναι το αποτέλεσμα της κοινής προσπάθειας μίας κοινοπραξίας, που ονομάζεται Globus Alliance, στην οποία συμμετέχουν η IBM, η Microsoft, η DAPRA, η NASA, το υπουργείο ενέργειας και το εθνικό ίδρυμα επιστημών των Η.Π.Α. Είναι μια συλλογή από τμήματα λογισμικού που έχουν σχεδιασθεί με τέτοιο τρόπο ώστε να υποστηρίζουν την ανάπτυξη των εφαρμογών που θα εκτελούνται σε υψηλής απόδοσης κατανεμημένα υπολογιστικά περιβάλλοντα. Κάθε ένα από τα τμήματα λογισμικού παρέχει μια κύρια υπηρεσία όπως η επικύρωση του χρήστη, η διαχείριση των πόρων, η συλλογή πληροφοριών για το σύστημα, η ανακάλυψη λαθών, η επικοινωνία και η απομακρυσμένη πρόσβαση στα δεδομένα. Ο τρόπος δόμησης και λειτουργίας των υπηρεσιών αναλύεται εκτενώς στο κείμενο μας.

Η εργασία περιλαμβάνει ανάλυση της αρχιτεκτονικής Open Grid Service Architecture (OGSA) η οποία παρουσιάστηκε τον Ιούνιο του 2002 στο Global Grid Forum και υιοθετήθηκε για πρώτη φορά για την υλοποίηση του Globus Toolkit 3. Οι δημιουργοί της αρχιτεκτονικής αυτής αποσκοπούν στον ορισμό κανόνων που πρέπει να διέπουν ένα περιβάλλον που υλοποιεί υπολογιστικά πλέγματα ώστε στα πλαίσια λειτουργίας του να είναι δυνατός ο συγκερασμός διαφορετικών τεχνολογιών πληροφορικής.

Στο πρακτικό τμήμα της εργασίας περιλαμβάνεται η αναφορά στην εγκατάσταση του Sun One Grid Engine και του Globus Toolkit 3 η οποία υλοποιήθηκε στο Εργαστήριο Συστημάτων Υπολογιστών και Επικοινωνιών του Οικονομικού Πανεπιστημίου Αθηνών. Η αναφορά περιέχει τα ευρήματα και τις παρατηρήσεις που έγιναν κατά τη φάση εγκατάστασης των δύο εργαλείων, σχετικά με τη δυσκολία που εμφάνισαν, το χρόνο που απαίτησαν και τις διαφορετικές τεχνολογίες που χρησιμοποιήθηκαν.

Επίσης, υλοποιήσαμε μια εφαρμογή χρησιμοποιώντας τις δυνατότητες του Globus toolkit και το υπολογιστικό πλέγμα που δημιουργήθηκε κατά τη φάση εγκατάστασης του. Η εφαρμογή αποτελεί μια υπηρεσία πλέγματος που παρέχει στο χρήστη τη δυνατότητα να κάνει μαθηματικές πράξεις μεταξύ δύο αριθμών. Από το πρακτικό μέρος της εργασίας καταλήξαμε σε ορισμένα χρήσιμα συμπεράσματα και παρατηρήσεις που αφορούν στα δύο περιβάλλοντα.

Πρώτη παρατήρηση αποτελεί το γεγονός ότι η εγκατάσταση του Globus Toolkit είναι πολύ πιο δύσκολη και χρονοβόρα σε σχέση με τη διαδικασία εγκατάστασης του Sun One Grid Engine. Η δυσκολία έγκειται στο συνδυασμό των πολλών διαφορετικών τεχνολογιών που απαιτούνται να ρυθμιστούν ώστε να ολοκληρωθεί η εγκατάσταση. Η έκταση της χρονικής διάρκειας της διαδικασίας σχετίζεται με τα μεγάλα χρονικά διαστήματα που απαιτεί το Globus ώστε να χτίσει το υπό ανάπτυξη πλέγμα.

Μέσα από την εφαρμογή που υλοποιήθηκε επιβεβαιώθηκε και το γεγονός ότι το Globus, ως το πρώτο εργαλείο που υιοθέτησε τις αρχές της OGSA, συνδυάζει πολλές και ετερόκλητες τεχνολογίες. Το γεγονός αυτό ανοίγει ένα παράθυρο έρευνας των δυνατοτήτων που υπάρχουν ώστε μέσα στις τεχνολογίες που συνδυάζει να ενταχθούν και οι τεχνολογίες άλλων εργαλείων ανάπτυξης υπολογιστικών πλεγμάτων όπως είναι το Sun One Grid Engine.

The following master thesis was composed by Vagelis Koutoumanos, post graduate student of the MSc programme in Informatin Systems, Athens University of Economics and Business.

Executive Summary

The following master thesis was composed by Vagelis Koutoumanos, post graduate student of the MSc programme in Informatin Systems, Athens University of Economics and Business.

Mr. Theodoros Apostolopoulos, professor of the Athens University of Economics and Business, was the supervisor.

The title of the project is "Distributed applications' management execution in grid computing: The environments "Sun One Grid Engine" and "Globus Toolkit" ". The title reveals the scope of the project, which is to study the installation and the functioning of "Sun One Grid Engine" and "Globus Toolkit". Also, the project contains the implementation of a Grid Service. In order to achieve our scope we used 3 computers which are located at the Communications and Computer Systems Laboratory (ccslab).

The term Grid Computing was formulated in the middle of the '90s in order to describe a distributed computational infrastructure, which was developed for high level scientific research. However, today, it is used for the description of complicated computational infrastructures which are related to many sections of computer science such as advanced networks or artificial intelligence.

The real meaning of the term Grid Computing corresponds to a virtual machine which in physical level is structured in a network of geographically distributed computational resources.

The following pages contain the main results of the project. The first part of the thesis is the theoretical part, which describes the basic concepts of Grid Computing. The second part is the practical part, which describes the implementation of the Grid Service. The third part is the conclusion of the project.



The basic functions of Grid Computing are the following:

1. Exploiting underutilized resources
2. Parallel CPU capacity
3. Virtual resources and virtual organizations for collaboration
4. Load Balancing
5. Simultaneous management of different projects

The basic types of a grid are the followings:

1. Global Grid
2. Enterprise Grid
3. Cluster Grid

The next sections of text include a detailed reference in the functioning of Sun One Grid Engine and Globus Toolkit.

One Grid Engine is the resolution for the construction of a grid by Sun company. We refer to the functioning and the interaction between the components of a grid which is structured on Sun One Grid Engine. We analyze the way that the hosts of a grid are organized in relation to their type (master, submit, administration or execution host) and the management of queues into which the jobs of the grid are organized. We, also, refer to Sun Grid Engine Enterprise Edition which is an upgrade edition of Sun One Grid Engine and it is used for enterprise or global grids.

The Globus Toolkit is a Globus Alliance research effort in which IBM, Microsoft, DAPRA, NASA, the ministry of energy and the US Institution of Science take part. It comprises a set of components that implement basic services for security, resource allocation, resource management and communication. The functioning and the structure of Globus Toolkit are analyzed in our text.

This master thesis, also, includes a detailed analysis of Open Grid Service Architecture (OGSA) which was introduced at June of 2002 in the Global Grid Forum and it was first adopted by Globus Toolkit. The developers of the OGSA are intended to determine the rules to which a grid should adhere in order to compromise different computer technologies.

The remainder of this report includes an analysis of the Sun One Grid Engine and Globus Toolkit installation which took place at the Communications and Computer Systems Laboratory (ccslab). The analysis refers to the findings and the observations about the difficulty of the installations, the time which we spent for them and the differentiation of the technologies which we used.

We, also, implemented an application for the grid which was constructed by the Globus Toolkit. This application, in fact, is a grid service. A user can use it in order to make the basic mathematical operations (addition, subtraction, multiply and division). That practical part of the project helped us to draw useful conclusions about the Sun One Grid Engine and Globus Toolkit.

Our first observation is that the installation of Globus Toolkit is more difficult and time-consuming process than the installation of Sun One Grid Engine. The difficulty consists in the combination of different technologies which is demanded for the completion of the installation. The extent time which was needed for the installation, is due to the long periods in which the Globus builds the grid.

The application verified that the Globus Toolkit, as the first tool which adopted the OGSA issues, compromises many and heterogeneous technologies. This can provide motivation to the scientific community to try a combination between Globus Toolkit and some others toolkits which are used for the development of grid such as Sun One Grid Engine.

Finally, we propose for further research, the study and the building of a theory in order to make easier the implementation of scalable algorithms which are essential for the parallel exploitation of the grid's CPU capacity.

Πληροφορική Εργασίας	9
Επεξηγητικό Έγχρωμα	12
Ευρετήριο Πλήκτρο	13
Εγκατάσταση του προγράμματος	14
1. Καταργητικό Πλέγμα	16
1.1 Σύνταξη Αριθμητικών Εξισώσεων	16
1.2 Τα Μεταβατικά σύμβολα και τα αριθμητικά	17
1.3 Η αριθμητική παραγωγή	18
1.4 Η αριθμητική διαδικασία	19
1.5 Η λειτουργία της αριθμητικής παραγωγής	20
1.6 Η αριθμητική εργασία	21
1.7 Τα σταθερά Βαρύτητα	22
1.8 Η αριθμητική παραγωγή	23
2. Λογισμικό	24
2.1 Η αριθμητική παραγωγή	25
2.2 Η αριθμητική εργασία	26

Πίνακας Περιεχομένων

Περίληψη Εργασίας **1**

Executive Summary..... **5**

Πίνακας Περιεχομένων **9**

Ευρετήριο Σχημάτων..... **12**

Ευρετήριο Πινάκων..... **13**

Ευχαριστίες..... **14**

1. Υπολογιστικό Πλέγμα..... **16**

1.1 Γενική Αναφορά **16**

1.2 Το Υπολογιστικό πλέγμα στην πράξη **17**

 1.2.1 Εκμετάλλευση αναδιοποίητων υπολογιστικών πόρων17

 1.2.3 Εικονικοί πόροι και οργανισμοί19

 1.2.4 Πρόσβαση σε σύνθετους πόρους.....19

 1.2.5 Αξιοπιστία.....20

 1.2.6 Εξισορρόπηση φορτίου20

 1.2.7 Διαχείριση Έργων.....21

 1.2.8 Πολιτικές διαχείρισης εργασιών.....21

1.3 Τύποι υπολογιστικών πλεγμάτων **23**

 1. Υπολογιστικό Πλέγμα Συστοιχίας23

 2. Επιχειρησιακό Υπολογιστικό Πλέγμα23

 3. Καθολικό Υπολογιστικό Πλέγμα24

1.4 Αρχιτεκτονική..... **25**

 1. Βαθμίδα Πρόσβασης26

 2. Βαθμίδα Διαχείρισης26

3. Υπολογιστική Βαθμίδα.....	26
2. Sun One Grid Engine.....	28
2.1 Γενικά	28
2.2 Τρόπος λειτουργίας	29
2.2.1 Παράδειγμα λειτουργίας.....	29
2.2.2 To Sun Grid Engine στην πράξη.....	30
2.3 Αρχιτεκτονική.....	32
2.4 Προγραμματισμός και Έλεγχος Εργασιών	37
2.5 Ασφάλεια	38
2.6 Υψηλή διαθεσιμότητα.....	38
2.7 Οι πολιτικές του SGEEE.....	39
3. The Globus Toolkit.....	43
3.1 Γενικά	43
3.2 Η μορφή του Globus.....	44
3.3 Τα δομικά στοιχεία του Globus	47
3.3.1 Διαχείριση Πόρων.....	47
3.3.2 Επικοινωνίες.....	49
3.3.3 Πληροφορία	51
3.3.4 Ασφάλεια	52
3.3.5 Έλεγχος κατάστασης	55
3.3.6 Απομακρυσμένη πρόσβαση σε αρχεία	55
3.3.7 Διαχείριση εκτελέσιμων διαδικασιών.....	56
3.3.8 Κιβώτιο Υπηρεσιών (Service Container).....	56

4. Open Grid Service Architecture.....	58
4.1 Περιγραφή της Αρχιτεκτονικής.....	59
4.1.1 Πόροι.....	59
4.1.2 Υπηρεσίες Ιστού & OGSI	60
4.1.3 Δημιουργικές Υπηρεσίες OGSA.....	64
5. ΥΛΟΠΟΙΗΣΗ	69
5.1 Εγκατάσταση Sun Grid Engine	70
5.2 Εγκατάσταση του Globus Toolkit 3	75
5.3 Παρατηρήσεις Εγκατάστασης.....	79
5.4 Εφαρμογή	81
6. Συμπεράσματα & Περαιτέρω Έρευνα	93
Βιβλιογραφία	96

Ευρετήριο Σχημάτων

Σχήμα 1 – Κατηγορίες υπολογιστικών πλεγμάτων	24
Σχήμα 2 - Αρχιτεκτονική υπολογιστικού πλέγματος.....	25
Σχήμα 3 – Διάταξη υπολογιστών σύμφωνα με SGE	34
Σχήμα 4 – Ροή Εργασιών σε SGE πλέγμα	36
Σχήμα 5 – Ροή Εργασιών σε SGE και SGEEE	41
Σχήμα 6 – Η κλεψύδρα του Globus	45
Σχήμα 7 – Διαχείριση πόρων στο Globus.....	48
Σχήμα 8 - Nexus	50
Σχήμα 9 - OGSA.....	59
Σχήμα 10 - OGSI.....	62
Σχήμα 11 – Δημιουργικές Υπηρεσίας OGSA	65
Σχήμα 12 – Τύποι Υπηρεσιών Πυρήνα	66
Σχήμα 13 – Ένδειξη Λειτουργίας NFS	72
Σχήμα 14 – Οθόνη πελάτη	90

Ευρετήριο Πινάκων

Πίνακας 1 – Πίνακας πληροφοριών sge_master	33
Πίνακας 2 – Βήματα Εγκατάστασης SGE	74
Πίνακας 3 - Βήματα Εγκατάστασης Globus Toolkit 3	78

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά:

Τον Καθηγητή του τμήματος Πληροφορικής του Οικονομικού Πανεπιστημίου Αθηνών, κο Θεόδωρο Αποστολόπουλο για την υποστήριξή του και την καθοδήγησή του κατά τη διάρκεια της εκπόνησης της παρούσας εργασίας.

Τον υποψήφιο διδάκτορα Γεώργιο Οικονόμου για τις συμβουλές και την πολύτιμη βοήθεια που μου προσέφερε.

Κεφάλαιο 1^ο

Υπολογιστικό Πλέυρα

1. Υπολογιστικό Πλέγμα

1.1 Γενική Αναφορά

Το υπολογιστικό πλέγμα (GRID) αποτελεί, τα τελευταία χρόνια, έναν ιδιαίτερα σημαντικό τομέα στο χώρο της πληροφορικής. Ο όρος υπολογιστικό πλέγμα διατυπώθηκε στα μέσα της δεκαετίας του 1990 ώστε να προσδιορίσει μια κατανεμημένη υπολογιστική υποδομή η οποία χρησιμοποιούνταν σε υψηλού επιπέδου επιστημονικές έρευνες. Σήμερα, ο όρος χρησιμοποιείται, καταχρηστικά, για να περιγράψει υποδομές που αφορούν σε διάφορους τομείς όπως από προηγμένα δίκτυα μέχρι τεχνητή νοημοσύνη [13].

Η πραγματική έννοια του όρου υπολογιστικό πλέγμα αντιστοιχεί σε μια εικονική μηχανή που σε φυσικό επίπεδο δομείται σε ένα δίκτυο γεωγραφικά κατανεμημένων υπολογιστικών πόρων.

Σε επίπεδο λογισμικού αυτή η εικονική μηχανή υλοποιείται από ένα στρώμα ενδιάμεσου λογισμικού (middleware) πάνω από το λειτουργικό σύστημα κάθε φυσικής μηχανής που συμμετέχει.

Το υπολογιστικό πλέγμα, όπως κάθε εικονική μηχανή, εκθέτει στο χρήστη του ένα σύνολο δυνατοτήτων πρόσβασης στη λειτουργικότητά του. Αυτές οι δυνατότητες δίνονται συνήθως με τη μορφή είτε εντολών διαχείρισης των εφαρμογών του, στα πλαίσια αυτής της εικονικής μηχανής, είτε βιβλιοθηκών με ρουτίνες που μπορεί άμεσα να χρησιμοποιήσει στον κώδικα των εφαρμογών του κατά την ανάπτυξη ή την τροποποίησή τους.

Μια ενδιαφέρουσα δυνατότητα προκύπτει και από την ολοκλήρωση της λειτουργικότητας του υπολογιστικού πλέγματος με υπηρεσίες ιστού (web services). Το πλέγμα, ως υπηρεσία ιστού πλέον, προβάλλει ως η επικρατούσα τάση όσον αφορά στην πρόσβαση του τελικού χρήστη σε αυτό.

1.2 Το Υπολογιστικό πλέγμα στην πράξη

1.2.1 Εκμετάλλευση αναξιοποίητων υπολογιστικών πόρων

1.2.1.1 Επεξεργαστική ισχύς

Η πιο απλή χρήση ενός υπολογιστικού πλέγματος είναι η αξιοποίηση της επεξεργαστικής ισχύος των μηχανημάτων που το συνθέτουν. Παρέχει τη δυνατότητα να εκτελεσθεί μια εφαρμογή, η οποία είναι εγκατεστημένη σε έναν υπολογιστή του, σε κάποιον από τους υπόλοιπους υπολογιστές που το συνθέτουν, επειδή ο πρώτος πρέπει να εξυπηρετήσει μεγάλο φόρτο εργασίας ενώ ο δεύτερος είναι αδρανής (idle). Για την υλοποίηση αυτής της διαδικασίας απαιτείται να πληρούνται δυο προϋποθέσεις. Η πρώτη είναι ότι η εφαρμογή πρέπει να είναι δυνατό να εκτελεσθεί σε διαφορετικό μηχάνημα δίχως να προκαλείται υπερβολικό αιωρούμενο φορτίο (overhead) στο πλέγμα. Η δεύτερη προσδιορίζει ότι είναι αναγκαίο το εναλλακτικό μηχάνημα να ικανοποιεί οποιαδήποτε απαίτηση υλικού ή λογισμικού που απαιτεί η διαδικασία εκτέλεση της συγκεκριμένης εφαρμογής.

1.2.1.2 Αποθηκευτικά μέσα

Άλλη χρήση του Grid σχετίζεται με την αξιοποίηση του αποθηκευτικού χώρου κάποιων μηχανημάτων του πλέγματος ο οποίος δεν χρησιμοποιείται. Το πλέγμα που διαχειρίζεται ενιαία όλο τον αποθηκευτικό χώρο των μηχανημάτων, συχνά ονομάζεται και υπολογιστικό πλέγμα δεδομένων (data grid). Τα πλέγματα αυτά παρέχουν τη δυνατότητα να ενοποιηθεί ο αχρησιμοποίητος αυτός χώρος στα πλαίσια μιας μεγάλης εικονικής αποθήκης δεδομένων, η οποία είναι δυνατό να χρησιμοποιηθεί από όλα τα στοιχεία που συνθέτουν το πλέγμα. Ο αποθηκευτικός χώρος κάθε μηχανήματος μπορεί να είναι είτε η μνήμη, είτε δευτερεύοντα αποθηκευτικά μέσα όπως σκληροί δίσκοι ή άλλα μέσα μόνιμης αποθήκευσης. Η διαχείριση δευτερευόντων αποθηκευτικών μέσων επιτυγχάνεται από διάφορα συστήματα τα οποία έχουν

αναπτυχθεί όπως Andrew File System (AFS), Network File System (NFS), Distributed File System (DFS), General Parallel File System (GPFS) τα οποία παρέχουν διαφορετικούς βαθμούς απόδοσης, και ποικίλα χαρακτηριστικά ασφάλειας και αξιοποιησίας.

Ο βαθμός χρήσης της συνολικής αποθηκευτικής ικανότητας των μηχανημάτων του πλέγματος, μπορεί να αυξηθεί με τη χρήση ενός ενοποιημένου συστήματος διαχείρισης αρχείων (file system) το οποίο θα αξιοποιήσει με ενιαίο τρόπο όλες τις αποθηκευτικές τους δυνατότητες. Επίσης, το ενοποιημένο αυτό σύστημα μπορεί να προσφέρει ένα απλό ομοιόμορφο χώρο ονομάτων (namespace). Έτσι καθίσταται ευκολότερο για τους χρήστες του συστήματος να χρησιμοποιούν δεδομένα που είναι αποθηκευμένα στο πλέγμα χωρίς να αναφέρουν την ακριβή θέση τους μέσα σε αυτό.

1.2.2 Παράλληλη αξιοποίηση επεξεργαστικής ισχύος

Βασικό χαρακτηριστικό της τεχνολογίας του Grid Computing είναι η δυνατότητα που παρέχει για παράλληλη αξιοποίηση της επεξεργαστικής ισχύος. Η χρήση της δυνατότητας αυτής απαιτεί την ανάπτυξη εφαρμογών οι οποίες δομούνται σε αλγόριθμους που μπορούν να επιμεριστούν σε ανεξάρτητα εκτελέσιμα τμήματα. Αυτά αποτελούν υπο-εφαρμογές κάθε μια από τις οποίες εκτελείται σε διαφορετικό μηχάνημα. Μάλιστα, όσο πιο ανεξάρτητα είναι τα τμήματα αυτά τόσο πιο πολύ αυξάνεται η κλιμάκωση (scalability) της εφαρμογής. Για παράδειγμα, μια απόλυτα κλιμακωτή εφαρμογή ολοκληρώνεται 10 φορές πιο γρήγορα εάν έχει χωρισθεί σε 10 τμήματα και χρησιμοποιηθούν για την εκτέλεσή της 10 επεξεργαστές σε σχέση με το χρόνο που θα απαιτηθεί αν χρησιμοποιηθεί ένας.

Στην προσπάθεια επίτευξη της πλήρους κλιμάκωσης παρουσιάζονται πολλά προβλήματα. Το κυριότερο είναι ότι οι περισσότεροι αλγόριθμοι είναι δυνατό να διαχωριστούν σε συγκεκριμένο αριθμό τμημάτων. Ένα άλλο πρόβλημα παρουσιάζεται στις περιπτώσεις που τα τμήματα δεν είναι εντελώς ανεξάρτητα.

Για παράδειγμα, εάν όλα τα τμήματα μιας εφαρμογής απαιτείται να διαβάζουν και να γράφουν από ένα αρχείο ή μια βάση δεδομένων, το όριο πρόσβασης του αρχείου ή της βάσης αποτελεί και το όριο κλιμάκωσης της εφαρμογής.

1.2.3 Εικονικοί πόροι και οργανισμοί

Σπουδαία λειτουργία του υπολογιστικού πλέγματος αποτελεί και η δυνατότητα που παρέχει να συνεργασθούν πολλοί χρήστες, διαφορετικών οργανισμών, μεταξύ τους. Αυτοί μπορούν να οργανωθούν δυναμικά στα πλαίσια εικονικών οργανισμών οι οποίοι διαμοιράζουν τους πόρους τους συγχωνεύοντας τους σε ένα μεγαλύτερο υπολογιστικό πλέγμα. Αυτό έχει τη δυνατότητα να υλοποιήσει τις πολιτικές διαμοιρασμού οι οποίες πρέπει να ισχύουν σε κάθε οργανισμό. Οι πολιτικές αυτές καθορίζουν επακριβώς το ποιοι πόροι διαμοιράζονται, τον τρόπο που γίνεται αυτό και ποιοι χρήστες μπορούν να έχουν πρόσβαση σε αυτούς.

1.2.4 Πρόσβαση σε σύνθετους πόρους

Πέρα από την πρόσβαση σε επεξεργαστές και αποθηκευτικά μέσα ένα υπολογιστικό πλέγμα παρέχει πρόσβαση σε σύνθετους υπολογιστικούς πόρους, όπως σε προηγμένο εξοπλισμό υλικού ή εφαρμογές λογισμικού που απαιτούν άδεια χρήσης.

Παραδείγματα πρόσβασης σε σύνθετους πόρους:

1. Αν ένας χρήστης χρειάζεται να αυξήσει το μέγεθος χρήσης του διαδίκτυου με σκοπό τη δημιουργία μιας μηχανής αναζήτησης, έχει τη δυνατότητα να το πραγματοποιήσει αν χρησιμοποιήσει τα μηχανήματα του πλέγματος που έχουν δικιά τους σύνδεση στο διαδίκτυο. Με αυτό τον τρόπο, η συνολική δυνατότητα αναζήτησης πολλαπλασιάζεται.

2. Έστω ότι η εκτέλεση μιας εργασίας απαιτεί ένα πρόγραμμα, η άδεια χρήσης του οποίου είναι ιδιαίτερα ακριβή. Μέσω του πλέγματος μπορούν όλοι οι χρήστες του να το χρησιμοποιούν εφόσον υπάρχει εγκατεστημένο σε ένα μόνο μηχάνημά του.

1.2.5 Αξιοπιστία

Πολλές φορές σε υπολογιστικά συστήματα που λειτουργούν για την εκτέλεση κρίσιμων εφαρμογών χρησιμοποιείται υλικό υψηλού κόστους (διπλοί επεξεργαστές, ειδικές πηγές ενέργειας κ.α.) ώστε να αυξηθεί η αξιοπιστία του με αποτέλεσμα το κόστος υλοποίησης και δημιουργίας να είναι, συχνά, απαγορευτικό. Η τεχνολογία του υπολογιστικού πλέγματος αντιμετωπίζει το πρόβλημα της αξιοπιστίας με πολύ μικρότερο κόστος. Αυτό το επιτυγχάνει μέσω της δυνατότητας που έχει να αντικαταστήσει εύκολα και άμεσα ένα μηχάνημα ή μια συσκευή που μπορεί να εμφανίσουν δυσλειτουργία.

1.2.6 Εξισορρόπηση φορτίου

Η τεχνολογία του υπολογιστικού πλέγματος παρέχει τη δυνατότητα εξισορρόπησης του φορτίου που δημιουργείται από τη ροή των εργασιών που καλείται να εξυπηρετήσει. Για τη διαδικασία αυτή χρησιμοποιεί δύο μεθόδους, ανάλογα με την κατάσταση που επικρατεί στους υπολογιστές του:

- Σε περιπτώσεις που παρατηρείται απρόσμενη αύξηση του φόρτου εργασίας ενός υπολογιστή τη διαμοιράζει σε ανενεργούς υπολογιστές, να επιτύχει αφενός ισόρροπη χρήση όλων των πόρων του.
- Στην περίπτωση που όλοι οι υπολογιστές έχουν υψηλό φορτίο εργασίας να εκτελέσουν η εκτέλεση της εργασίας με τη χαμηλότερη προτεραιότητα είτε

αναβάλλεται, είτε ακυρώνεται ώστε να δοθεί η δυνατότητα να εκτελεσθούν υψηλής προτεραιότητας εργασίες.

1.2.7 Διαχείριση Έργων

Το υπολογιστικό πλέγμα παρέχει δυνατότητες διαχείρισης προτεραιοτήτων ανάμεσα σε διαφορετικά έργα που το χρησιμοποιούν. Ανατρέπει την ισχύουσα κατάσταση με βάση την οποία η υλοποίηση ενός έργου απαιτούσε την ύπαρξη αντίστοιχης δεσμευμένης για το σκοπό αυτό υπολογιστικής υποδομής. Το γεγονός αυτό οδηγούσε, συχνά, στην εκδήλωση του φαινομένου της μειωμένης χρήσης των πόρων αυτών ενώ κάποιο άλλο έργο χρειαζόταν επιπλέον πόρους από αυτούς που του είχαν παρασχεθεί. Η τεχνολογία του υπολογιστικού πλέγματος δίνει τη δυνατότητα η υλοποίηση διαφορετικών έργων να πραγματοποιείται με χρήση κοινών πόρων ανάλογα με τις εκάστοτε απαιτήσεις κάθε έργου.

1.2.8 Πολιτικές διαχείρισης εργασιών

Ένα υπολογιστικό πλέγμα είναι δυνατό να χρησιμοποιήσει διαφορετικές πολιτικές στον τρόπο που διαχειρίζεται την εκτέλεση των εργασιών που υποβάλλονται σ' αυτό. Η πιο απλή προσδιορίζει ότι ο χρήστης που υποβάλλει την εργασία αφού επιλέξει το μηχάνημα που θεωρεί πιο κατάλληλο για την εκτέλεσή, στη συνέχεια την αποστέλλει σ' αυτό.

Πιο προηγμένα πλέγματα περιλαμβάνουν ένα προγραμματιστή εργασιών ο οποίος προσδιορίζει αυτόματα με βάση συγκεκριμένα κριτήρια το μηχάνημα που θα εκτελεσθεί κάθε εργασία. Μάλιστα, τα μηχανήματα δεσμεύονται από το πλέγμα ώστε να μην είναι δυνατό να εκτελέσουν εργασίες που δεν υποβάλλονται από χρήστες του. Το φαινόμενο της δέσμευσης πόρων παρατηρείται κυρίως σε περιπτώσεις που η υλοποίηση ενός έργου έχει στενά

χρονικά όρια ώστε οι κατασκευαστές του να έχουν εξασφαλίσει τους απαιτούμενους για την ανάπτυξή του πόρους.

Οι παραπάνω πολιτικές εφαρμόζονται, μέχρι σήμερα, σε περιπτώσεις πλεγμάτων στα οποία ο διαμοιραζόμενος πόρος είναι οι επεξεργαστές [14]. Ωστόσο, η εφαρμογή τους και σε άλλες περιπτώσεις θα μπορούσε να βελτιστοποιήσει τη λειτουργία ενός πλέγματος. Για παράδειγμα, είναι ιδιαίτερα σημαντικό οι εργασίες να εκτελούνται κοντά στα μηχανήματα που είναι αποθηκευμένα τα δεδομένα που χρησιμοποιούν. Με τον τρόπο αυτό μειώνεται σημαντικά η κυκλοφορία (traffic) στο δίκτυο. Βέβαια το γεγονός ότι μέχρι σήμερα σύνθετες πολιτικές δεν έχουν εφαρμοσθεί οφείλεται στη μεγάλη πολυπλοκότητα που παρουσιάζουν.



1.3 Τύποι υπολογιστικών πλεγμάτων

Ένα υπολογιστικό πλέγμα είναι δυνατό να ανήκει σε μια από τις παρακάτω κατηγορίες¹ [5]:

- ✓ **Υπολογιστικό Πλέγμα Συστοιχίας (Cluster Grid)**
- ✓ **Επιχειρησιακό Υπολογιστικό Πλέγμα (Enterprise Grid)**
- ✓ **Καθολικό Υπολογιστικό Πλέγμα (Global Grid)**

1. Υπολογιστικό Πλέγμα Συστοιχίας

Το υπολογιστικό πλέγμα συστοιχίας αποτελεί την πιο απλή μορφή ενός υπολογιστικού πλέγματος. Περιλαμβάνει διάφορα υπολογιστικά συστήματα τα οποία επικοινωνούν μεταξύ τους μέσω ενός δικτύου. Τα συστήματα αυτά μπορεί να είναι κατανεμημένοι σταθμοί εργασίας και εξυπηρετητές οι οποίοι χρησιμοποιούν εκτός από τους δικούς τους και πόρους οι οποίοι βρίσκονται σε ένα κεντρικό τομέα του πλέγματος. Συνήθως, ένα cluster grid ανήκει και χρησιμοποιείται από την ομάδα ανάπτυξης ενός έργου ή από ένα τμήμα ενός οργανισμού.

2. Επιχειρησιακό Υπολογιστικό Πλέγμα

Ένα επιχειρησιακό υπολογιστικό πλέγμα διαμορφώνεται από το συνδυασμό και την συνλειτουργία πολλών υπολογιστικών πλεγμάτων συστοιχίας. Χρησιμοποιούνται από διάφορές ομάδες υλοποίησης ενός έργου ή από διάφορα τμήματα ενός οργανισμού ώστε να μπορούν να χρησιμοποιούσουν

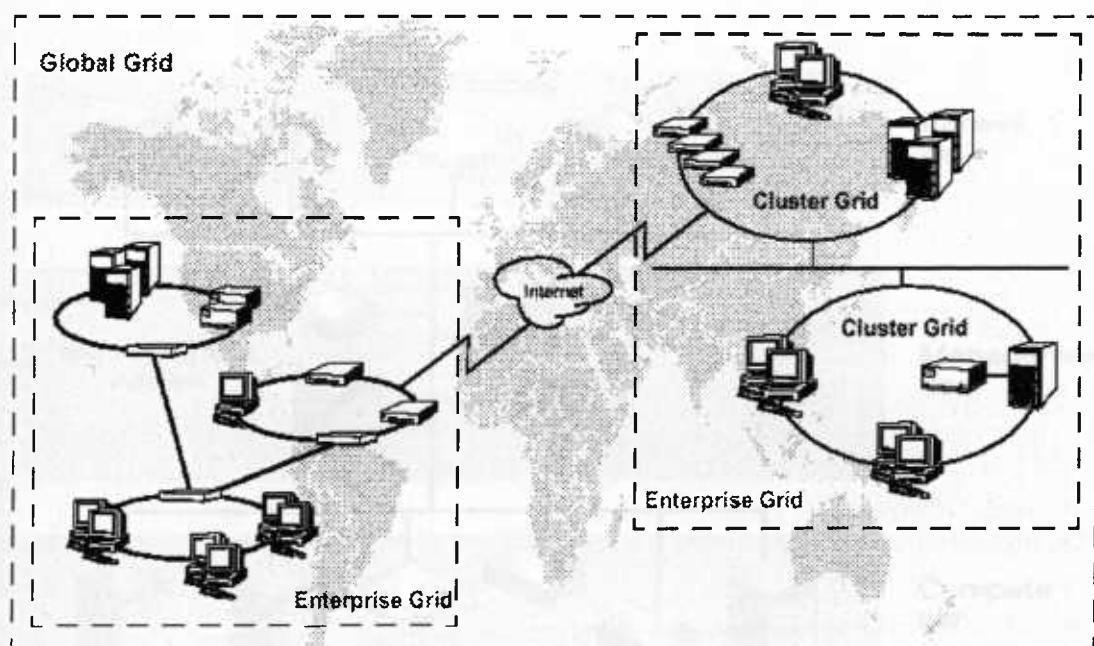
¹ Η ορολογία αυτή χρησιμοποιείται κυρίως από την εταιρία SUN. Ωστόσο και στην υπόλοιπη βιβλιογραφία η ορολογία που χρησιμοποιείται είναι είτε συνώνυμη είτε ταυτόσημη.

κοινούς υπολογιστικούς πόρους. Αυτοί ανήκουν σε διαφορετικούς τομείς διαχείρισης αλλά βρίσκονται στον ίδιο γεωγραφικό χώρο.

3. Καθολικό Υπολογιστικό Πλέγμα

Τα καθολικά υπολογιστικά πλέγματα είναι συλλογές από επιχειρησιακά υπολογιστικά πλέγματα, τα οποία απαιτείται να ακολουθούν ίδιες πολιτικές λειτουργίας και να χρησιμοποιούν ίδια πρωτόκολλα επικοινωνίας, ακόμη και αν έχουν κατασκευαστεί με διαφορετικό τρόπο. Οι υπολογιστικοί πόροι που περιλαμβάνουν μπορεί να είναι γεωγραφικά διασπαρμένοι και να είναι διαθέσιμοι σε όλους τους χρήστες του πλέγματος. Έτσι δίνουν τη δυνατότητα συνεργασίας ακόμα και σε διαφορετικούς οργανισμούς.

Στο σχήμα 1 φαίνονται οι 3 κατηγορίες υπερ-υπολογιστικών πλεγμάτων:



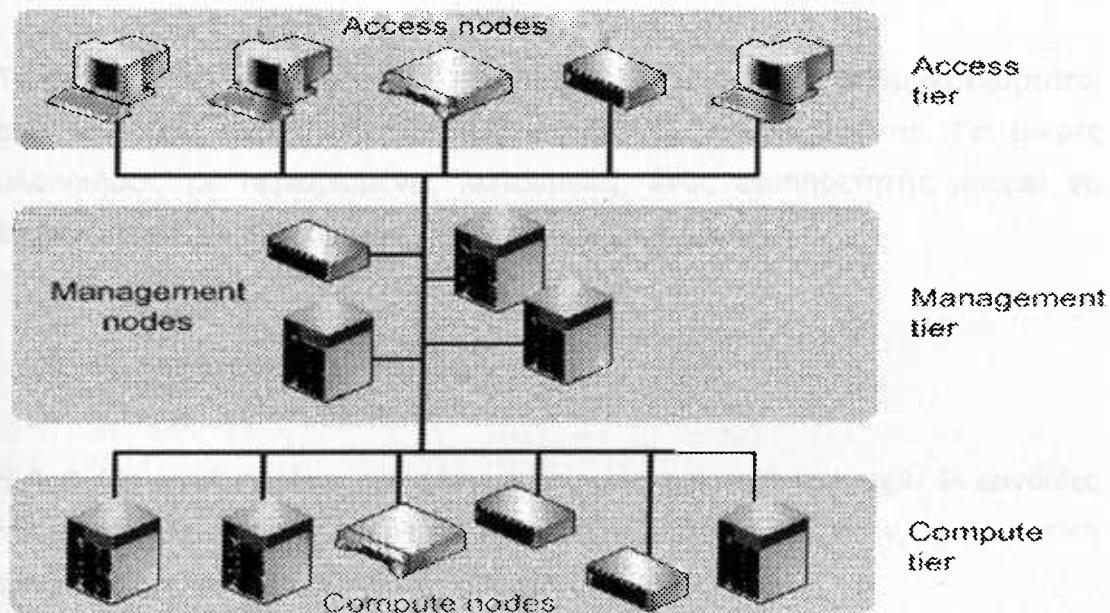
Σχήμα 1 – Κατηγορίες υπολογιστικών πλεγμάτων

1.4 Αρχιτεκτονική

Η αρχιτεκτονική ενός υπολογιστικού πλέγματος χωρίζεται σε τρεις βαθμίδες :

- ✓ **Πρόσβαση**
- ✓ **Διαχείριση**
- ✓ **Υπολογιστική**

Κάθε βαθμίδα, όπως φαίνεται στο σχήμα 2, ορίζεται από τις υπηρεσίες που παρέχει. Το επίπεδο πρόσβασης παρέχει δυνατότητες πρόσβασης στο πλέγμα. Το επίπεδο διαχείρισης παρέχει τις κύριες υπηρεσίες του πλέγματος, όπως η διαχείριση εργασιών και ο έλεγχος της κατάστασης των στοιχείων που το συνθέτουν. Το υπολογιστικό επίπεδο παρέχει την υπολογιστική ισχύ του πλέγματος και υποστηρίζει το περιβάλλον εκτέλεσης των εφαρμογών των χρηστών.



Σχήμα 2 - Αρχιτεκτονική υπολογιστικού πλέγματος

1. Βαθμίδα Πρόσβασης

Το επίπεδο πρόσβασης παρέχει υπηρεσίες πρόσβασης και αυθεντικοποίησης στους χρήστες του πλέγματος. Η μέθοδος πρόσβασης αφενός είναι δυνατό να στηρίζεται είτε σε συμβατικές τεχνολογίες όπως telnet, rlogin, ftp και ssh είτε σε τεχνολογίες διαδικτύου, αφετέρου απαιτείται να μπορεί να συνδυαστεί με διαδεδομένες τεχνολογίες αυθεντικοποίησης όπως το LDAP και το Kerberos.

2. Βαθμίδα Διαχείρισης

Η ενδιάμεση βαθμίδα είναι υπεύθυνη για την παροχή των βασικών υπηρεσιών του πλέγματος όπως διαχείριση κατανεμημένων υπολογιστικών πόρων, έλεγχος απόδοσης συστήματος και έλεγχος υλικού. Επίσης, μπορεί να διαδραματίζει τους εξής ρόλους:

- ✓ Εξυπηρετητής Αρχείων (File Server)
- ✓ Εξυπηρετητής αδειών λογισμικού

Το μέγεθος και ο αριθμός των εξυπηρετητών σ' αυτή τη βαθμίδα εξαρτάται από τον τύπο και το επίπεδο των υπηρεσιών που παρέχονται. Για μικρές υλοποιήσεις με περιορισμένες λειτουργίες, ένας εξυπηρετητής μπορεί να επιλεγεί ώστε να διαχειρίζεται τις αντίστοιχες υπηρεσίες.

3. Υπολογιστική Βαθμίδα

Η βαθμίδα αυτή παρέχει στο πλέγμα την υπολογιστική του ισχύ. Οι εργασίες που υποβάλλονται στις ανώτερες βαθμίδες κατευθύνονται στην υπολογιστική ώστε να εκτελεσθούν σε κάποιο από τους κόμβους της.

Κεφάλαιο 2^ο

Sun One Grid Engine

2. Sun One Grid Engine

2.1 Γενικά

Το περιβάλλον One Grid Engine αποτελεί την πρόταση της εταιρίας Sun για τη διαχείριση κατανεμημένων υπολογιστικών πλεγμάτων. Σκοπός της λειτουργίας του είναι η βελτιστοποίηση της χρήσης του υλικού και του λογισμικού ενός ετερογενούς υπολογιστικού περιβάλλοντος στα πλαίσια του οποίου αλληλεπιδρούν πολλά υπολογιστικά συστήματα. Σε αυτά κατανέμει το φόρτο εργασίας, που καλείται να εξυπηρετήσει το περιβάλλον, με στόχο την αύξηση της παραγωγικότητα τους και τη μεγιστοποίηση του πλήθους των εργασιών τις οποίες είναι δυνατό να εκτελέσουν.

Το Sun One Grid Engine (SGE), όπως είναι η πλήρης ονομασία του, για την επίτευξη των επιδιώξεων του συλλέγει και συνθέτει τους διαθέσιμους υπολογιστικούς πόρους παρέχοντας έναν απλό, σταθερό και αξιόπιστο τρόπο πρόσβασης στους χρήστες. Επίσης, παρέχει πληροφορίες και στατιστικά δεδομένα σχετικά με τον τρόπο επεξεργασίας των υπό εκτέλεση εργασιών, στοιχεία τα οποία αφενός καθιστούν ευκολότερο τον έλεγχο του βαθμού χρησιμοποίησης των πόρων, ο οποίος επιτυγχάνεται, αφετέρου προσδιορίζουν τον τρόπο με τον οποίο είναι δυνατό να αυξηθεί ο βαθμός αυτός.

2.2 Τρόπος λειτουργίας

Ας δούμε τον τρόπο λειτουργίας του SGE μέσα από ένα παράδειγμα.

2.2.1 Παράδειγμα λειτουργίας

Ο τρόπος λειτουργίας του μπορεί εύκολα να γίνει κατανοητός αν παραλληλιστεί με τον τρόπο λειτουργίας ενός μεγάλου υποκαταστήματος μιας τράπεζας. Ας εξετάσουμε το εξής υποθετικό σενάριο [1]:

Στο χώρο υποδοχής πελατών του υποκαταστήματος υπάρχουν πολλοί πελάτες με διαφορετικές απαιτήσεις, οι οποίοι περιμένουν να εξυπηρετηθούν. Ένας από αυτούς θέλει να κάνει ανάληψη ενός μικρού ποσού από το λογαριασμό του. Ο επόμενος έχει συνάντηση με κάποιο από τους επενδυτικούς συμβούλους της τράπεζας ώστε να συζητήσουν σχετικά με μια μεγάλη επένδυση την οποία σκοπεύει να κάνει. Ο πελάτης, που προηγείται των δύο προαναφερόμενων, θέλει να κάνει αίτηση για δάνειο όπως και οι 8 πελάτες που βρίσκονται μπροστά του. Έστω, ότι τη συγκεκριμένη μέρα, η τράπεζα έχει πολλούς υπαλλήλους που είναι κατάλληλοι να εξυπηρετήσουν τον πελάτη που θέλει να κάνει ανάληψη μετρητών από το λογαριασμό, αλλά μόνο έναν ή δύο υπαλλήλους που μπορούν να εξυπηρετήσουν τις πολλές αιτήσεις για δάνεια. Αποτέλεσμα της κατάστασης αυτής είναι πολλοί πελάτες να πρέπει να περιμένουν αρκετή ώρα ώστε να εξυπηρετηθούν.

Εάν το SGE ήταν ο διευθυντής της τράπεζας θα οργάνωνε τον τρόπο εξυπηρέτησης ως εξής:

- ✓ Τη στιγμή που φθάνουν στο υποκατάστημα οι πελάτες καλούνται να δηλώσουν τα στοιχεία τους και τις απαιτήσεις τους ενώ καταγράφεται και ο χρόνος άφιξης τους.

- ✓ Με βάση τις πληροφορίες αυτές, οι πελάτες που εξυπηρετούνται είναι εκείνοι, οι ανάγκες των οποίων, μπορούν να ικανοποιηθούν από κάποιο διαθέσιμο υπάλληλο, έχουν την υψηλότερη προτεραιότητα και ο χρόνος αναμονής τους είναι ο μεγαλύτερος.
- ✓ Ένας υπάλληλος είναι δυνατό να εξυπηρετήσει ταυτόχρονα περισσότερους από έναν πελάτες.
- ✓ Μεταξύ δύο υπαλλήλων που μπορούν να εξυπηρετήσουν ένα πελάτη επιλέγεται ο υπάλληλος με το μικρότερο φόρτο εργασίας.
- ✓ Η τράπεζα έχει τη δυνατότητα να προσδιορίσει συγκεκριμένες πολιτικές εξυπηρέτησης π.χ. Παρέχεται προνομιακή εξυπηρέτηση σε εταιρικούς πελάτες, επειδή είναι η πιο επικερδής κατηγορία πελατών. Διασφαλίζεται ότι μια συγκεκριμένη κατηγορία πελατών εξυπηρετείται αποτελεσματικά γιατί κατά το παρελθόν είχαν διατυπωθεί παράπονα. Διασφαλίζεται ότι οι πελάτες που έχουν συγκεκριμένη ώρα προκαθορισμένη συνάντηση εξυπηρετούνται εγκαίρως.
- ✓ Σύμφωνα με τις πολιτικές αυτές το σύστημα δίνει προτεραιότητα εξυπηρέτησης στους αντίστοιχους πελάτες. Η πιστή εφαρμογή τους ελέγχεται από το σύστημα.

2.2.2 To Sun One Grid Engine στην πράξη

Με βάση το προηγούμενο σενάριο λειτουργίας μιας τράπεζας, σε ένα υπολογιστικό πλέγμα που χρησιμοποιεί το SGE η εργασία αντιστοιχεί στον πελάτη, ο χώρος υποδοχής, στον χώρο που αποθηκεύονται οι υπό εκτέλεση εργασίες και οι ουρές που βρίσκονται σε κάποια μηχανήματα, στους υπαλλήλους. Παρόμοια και οι απαιτήσεις εκτέλεσης κάθε εργασίας, όπως

διαθέσιμη μνήμη, ταχύτητα επεξεργασίας, διαθέσιμες άδειες λογισμικού κλπ, μπορούν να διαφέρουν και μόνο συγκεκριμένες ουρές να είναι δυνατό να εξυπηρετήσουν μια εργασία.

Σε ένα περιβάλλον που χρησιμοποιείται το SGE η διαχείριση των εργασιών και των διαθέσιμων πόρων γίνεται ως εξής:

- ✓ Ένας χρήστης ο οποίος εναποθέτει στο υπολογιστικό περιβάλλον μια εργασία, καθορίζει το σύνολο των απαιτήσεων που πρέπει να ικανοποιούνται ώστε να εκτελεσθεί. Ταυτόχρονα, καταγράφεται η ταυτότητα του χρήστη και ο χρόνος εναπόθεσης της εργασίας.
- ✓ Τη στιγμή που μια ουρά καθίσταται διαθέσιμη ώστε να εξυπηρετήσει μια νέα εργασία, το SGE καθορίζει ποιες εργασίες είναι κατάλληλες ώστε να αποδοθούν στην ουρά αυτή και στη συνέχεια αποδίδει εκείνη με τη μεγαλύτερη προτεραιότητα ή το μεγαλύτερο χρόνο αναμονής.
- ✓ Οι ουρές του SGE επιτρέπουν την ταυτόχρονη εκτέλεση διαφορετικών εργασιών.
- ✓ Στόχο του SGE αποτελεί η απόδοση νέων εργασιών στη λιγότερο φορτωμένη και στην πιο κατάλληλη ουρά.

2.3 Αρχιτεκτονική

Το Sun One Grid Engine υιοθετεί τις αρχές της Διαχείρισης Κατανεμημένων Πόρων (Distributed Resource Management – DRM). Το γεγονός αυτό του αποδίδει όλα τα χαρακτηριστικά του DRM. Δηλαδή, οι βασικότερες λειτουργίες του είναι το batch queuing, η εξισορρόπηση φορτίου, ο υπολογισμός στατιστικών στοιχείων εκτέλεσης εργασιών, η δυνατότητα ακύρωσης και ανάκτησης εργασιών.

Τα δομικά στοιχεία που συνθέτουν την αρχιτεκτονική και αποτελούν τον πυρήνα λειτουργίας του είναι τα εξής:

- ✓ Master host: Ένας από τους υπολογιστές του πλέγματος επιλέγεται να είναι ο master host του SGE. Αυτός διαχειρίζεται όλες τις αιτήσεις των χρηστών και λαμβάνει αποφάσεις για τον προγραμματισμό εκτέλεσης των αιτούμενων εργασιών και την επιλογή του κόμβου στον οποίο θα εκτελεσθούν. Η λειτουργία του βασίζεται σε δύο daemons τον master (*sge_master*) και τον scheduler (*sge_schedd*).
 - *Sge_master*: Διαδραματίζει το σπουδαιότερο ρόλο στη διαδικασία διαχείρισης και προγραμματισμού των ενεργειών του cluster. Διατηρεί και ενημερώνει πίνακες με πληροφορίες σχετικά με τους hosts, τις εργασίες, τις ουρές εργασιών, το φορτίο του συστήματος και τους χρήστες. Λαμβάνει αποφάσεις προγραμματισμού εκτέλεσης εργασιών από το *sge_schedd* και αιτήσεις δράσης από τους *sge_execd* των διαφόρων execution hosts. Παράδειγμα της μορφής του πίνακα που διατηρεί ο *sge_master* αποτελεί ο πίνακας 1 [1].

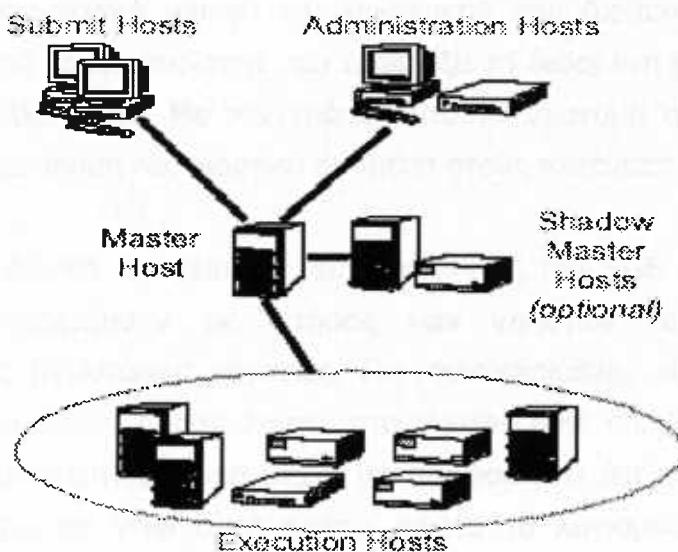
	Host 1	Host 2	Host 3
Arch	Linux	Sol 7	Sol 7
Load	0.4	1.3	1.3
State	Idle	Idle	Full

Πίνακας 1 – Πίνακας πληροφοριών sge_master

- Sge_schedd: Αποφασίζει σχετικά με τον προγραμματισμό εκτέλεσης των εργασιών. Πιο συγκεκριμένα, καθορίζει ποια εργασία θα αποσταλεί σε ποια ουρά και αποστέλλει τον ορισμό αυτό στον sge_master ώστε να τον υλοποιήσει.
- ✓ Execution hosts: Οι κόμβοι οι οποίοι επιλέγονται από το master host να εκτελούν τις εργασίες. Η λειτουργία τους βασίζεται στον execution daemon sge_execd.
- Sge_execd: Είναι υπεύθυνος για τις ουρές εργασιών στον αντίστοιχο host και για την εκτέλεση των εργασιών που περιλαμβάνουν.
- ✓ Submit host: Τα μηχανήματα που υποβάλλουν αιτήσεις εκτέλεσης εργασιών, ενώ ταυτόχρονα τις διαχειρίζονται και τις ελέγχουν.
- ✓ Administration: Οι διαχειριστές του SGE χρησιμοποιούν κάποιους κόμβους διαχείρισης ώστε να πραγματοποιούν τις αλλαγές που θέλουν στον τρόπο λειτουργίας του πλέγματος, όπως αλλαγές στις παραμέτρους του DRM, προσθήκη νέων κόμβων ή νέων χρηστών.
- ✓ Shadow master host: Εκτός από το master host είναι δυνατό άλλα μηχανήματα να ορισθούν ως shadow master hosts. Σκοπό αυτών των μηχανημάτων αποτελεί ο συνεχής έλεγχος του master και η αυτόματη και διαφανής αντικατάστασή του στην περίπτωση που

προκύψει κάποιο πρόβλημα στο master. Με τον τρόπο αυτό οι εργασίες που εκτελούνται στο cluster δεν επηρεάζονται από την αποτυχία του master host.

Στο παρακάτω σχήμα 2 [5] φαίνεται η διάταξη των στοιχείων του SGE.



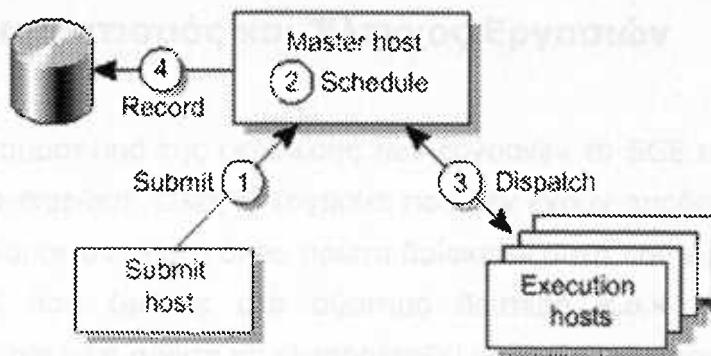
Σχήμα 3 – Διάταξη υπολογιστών σύμφωνα με SGE

- ✓ **Ουρές:** Η ουρά ορίζεται ως ένας χώρος συγκέντρωσης μιας κατηγορίας εργασιών οι οποίες είναι δυνατό να εκτελεσθούν σε ένα συγκεκριμένο υπολογιστή του πλέγματος, ταυτόχρονα. Η ουρά αποδίδει στις εργασίες που περιλαμβάνει συγκεκριμένα χαρακτηριστικά όπως για παράδειγμα πότε είναι δυνατό να μεταναστεύσουν. Καθ' όλη τη διάρκεια του χρόνου ύπαρξή τους, οι εργασίες εξαρτώνται από την ουρά στην οποία περιέχονται. Η εξάρτηση αυτή επηρεάζει την κατάσταση κάθε εργασίας π.χ. αν η λειτουργία μιας ουράς διακοπεί, ταυτόχρονα διακόπτεται και η εκτέλεση όλων των εργασιών που περιλαμβάνει. Η απόθεση μιας εργασίας σε μια ουρά πραγματοποιείται με βάση τις προδιαγραφές

σε υπολογιστικούς πόρους (π.χ. μνήμη, λειτουργικό σύστημα, διαθέσιμο λογισμικό κ.α.) που πρέπει να πληρούνται ώστε να εκτελεσθεί. Δηλαδή, αφού ληφθεί από το master host μια εργασία εξετάζονται οι απαιτήσεις που πρέπει να ικανοποιούνται ώστε να εκτελεσθεί αποτελεσματικά και στη συνέχεια αποστέλλεται σε μια ουρά η οποία βρίσκεται στον κατάλληλο, με βάση τόσο τα χαρακτηριστικά υλικού και λογισμικού που διαθέτει όσο και του βαθμού χρησιμοποίησης που εμφανίζει τη δεδομένη χρονική στιγμή, execution host. Με τον τρόπο αυτό το σύστημα αποσκοπεί στην εξισορρόπηση του φορτίου ανάμεσα στους execution hosts.

Τα παραπάνω δομικά στοιχεία της αρχιτεκτονικής του SGE αλληλεπιδρούν ώστε να εξυπηρετήσουν τις αιτήσεις των χρηστών του συστήματος εκτελώντας τις αντίστοιχες εργασίες. Πιο συγκεκριμένα, οι εργασίες που πρέπει να εκτελεστούν υποβάλλονται στο master host και διατηρούνται σε προσωρινό αποθηκευτικό χώρο μέχρι να αποφασισθεί ότι είναι έτοιμες να εκτελεσθούν. Για να γίνει αυτό πρέπει πρώτα το λογισμικό του SGE να αποφασίσει ότι οι διαθέσιμοι υπολογιστικοί πόροι ανταποκρίνονται στις απαιτήσεις κάθε εργασίας, αφού οι απαιτήσεις αυτές μπορεί να διαφέρουν πάρα πολύ και μόνο συγκεκριμένοι υπολογιστές να έχουν τη δυνατότητα να τις ικανοποιούν. Στη συνέχεια, η εργασία με την υψηλότερη προτεραιότητα αποστέλλεται στον αντίστοιχο execution host ώστε να εκτελεσθεί.

Η τυπική ροή των εργασιών σε ένα πλέγμα που υλοποιείται με βάση το SGE φαίνεται στο σχήμα 4 [6].



Σχήμα 4 – Ροή Εργασιών σε SGE πλέγμα

1. Submission: Η αίτηση ενός χρήστης υποβάλλεται μέσω του submit host και στη συνέχεια αποστέλλεται στο master host.
2. Scheduling: O master host ορίζει το host στον οποίο θα εκτελεσθεί η εργασία.
3. Execution: Αφού προοδιορισθεί ο host στον οποίο θα εκτελεσθεί μια εργασία, ο master τη στέλνει σ' αυτόν. Αυτός αποθηκεύει την εργασία σε μια βάση δεδομένων με πληροφορίες για τις εργασίες και ξεκινάει τη διαδικασία εκτέλεσης της.
4. Record: Όταν ολοκληρωθεί η εκτέλεση της εργασίας ο execution host ενημερώνει το master και διαγράφει την εργασία από την αντίστοιχη βάση δεδομένων. Ο master με τη σειρά του ενημερώνει μια βάση δεδομένων στην οποία καταχωρούνται πληροφορίες σχετικά με την εκτελεσμένες εργασίες.

2.4 Προγραμματισμός και Έλεγχος Εργασιών

Για τον προγραμματισμό της εκτέλεσης των εργασιών το SGE εφαρμόζει τον κανόνα *first-in-first-out*. Όλες οι εργασίες που δεν έχουν αποδοθεί σε κάποια ουρά βρίσκονται σε μια λίστα όπου πρώτη βρίσκεται αυτή που έφθασε πρώτη, δεύτερη αυτή που έφθασε στο σύστημα δεύτερη κ.ο.κ. Επιδίωξη του συστήματος αποτελεί η πρώτη να εξυπηρετηθεί πρώτη κατά σειρά. Για το λόγο αυτό αναζητά κατάλληλη ουρά ώστε να την αποστείλει. Ωστόσο, πολλές φορές ο διαχειριστής του συστήματος αποδίδει σε κάποιες εργασίες μεγαλύτερο βαθμό προτεραιότητας σε σχέση με άλλες ώστε να εκτελεσθούν πιο γρήγορα. Αφού ο διαχειριστής αποδώσει τους βαθμούς προτεραιότητας τότε ταξινομούνται οι εργασίες από αυτή με το μεγαλύτερο βαθμό προς αυτή με το μικρότερο. Αν στη λίστα υπάρχουν 2 ή περισσότερες εργασίες με τον ίδιο βαθμό προηγείται αυτή που είναι περισσότερη ώρα στη λίστα.

Η παραπάνω διαδικασία πολλές φορές, ιδιαίτερα σε περιπτώσεις που ένας χρήστης υποβάλλει μια σειρά από εργασίες την ίδια στιγμή, οδηγεί σε καθυστέρηση εκτέλεσης των εργασιών πολλών χρηστών. Για το λόγο αυτό, συχνά, χρησιμοποιείται η διαδικασία του *προγραμματισμού ίσου διαχωρισμού* (*equal-share-scheduling*) με βάση την οποία οι εργασίες που αποστέλλονται από ένα χρήστη του οποίου ήδη εκτελείται μια εργασία τοποθετούνται στο τέλος της λίστας.

Το SGE με στόχο την αύξηση της αξιοπιστίας παρέχει ένα ολοκληρωμένο μηχανισμό ελέγχου της διαδικασίας εκτέλεσης μιας εργασίας. Ο μηχανισμός αυτός, ο οποίος ονομάζεται *checkpointing jobs*, έχει μορφή εργασιών ελέγχου και κύριος στόχος του είναι να προσφέρει δυνατότητα επανεκτέλεσης μιας εργασίας σε περίπτωση που η εκτέλεσή της αποτύχει για οποιοδήποτε λόγο. Ο έλεγχος που πραγματοποιεί, γίνεται με την αντιγραφή ανά τακτά χρονικά διαστήματα της κατάστασης εκτέλεσης μιας εργασίας σε συγκεκριμένα αρχεία. Τα αρχεία αυτά, τα οποία ονομάζονται αρχεία επανεκκίνησης,

χρησιμοποιούνται στις περιπτώσεις που απαιτείται η επανεκτέλεσης κάποιας εργασίας, για να διαπιστωθεί σε πιο σημείο είχε φθάσει η εξέλιξη της διαδικασίας εκτέλεσης της, ώστε η επανεκκίνησης να έχει ως αφετηρία το σημείο αυτό.

2.5 Ασφάλεια

Για να είναι δυνατός ο έλεγχος πρόσβασης στο περιβάλλον το master host διατηρεί και επεξεργάζεται πληροφορίες σχετικά με την καταλληλότητα και την αξιοπιστία των submit και administration hosts. Δηλαδή, συντηρεί μια λίστα με πληροφορίες που αφορούν στις προδιαγραφές ασφάλειας των υπολογιστών του πλέγματος. Με βάση αυτή τη λίστα προσδιορίζει ποίοι από αυτούς είναι δυνατό να διαδραματίζουν συγκεκριμένους ρόλους και ποίοι όχι. Για παράδειγμα, με στόχο να αυξήσει την ασφάλεια του συστήματος μπορεί να προσδιορίσει ότι administration hosts μπορεί να είναι μόνο όσοι υπολογιστές βρίσκονται σε ένα ασφαλισμένο με φυσικά μέσα χώρο.

2.6 Υψηλή διαθεσιμότητα

Το SGE είναι δυνατό να λειτουργεί με τέτοιο τρόπο ώστε να ελαχιστοποιούνται οι περιπτώσεις αποτυχίας του συστήματος και αδυναμίας εξυπηρέτησης των υποβαλλόμενων από τους χρήστες εργασιών. Το γεγονός αυτό καθίσταται εφικτό μέσω της δυνατότητας που παρέχεται να προσδιορισθεί ένας ή περισσότεροι υπολογιστές ως shadow master host. Αυτοί στις περιπτώσεις που η λειτουργία του master host διακόπτεται, τον αντικαθιστούν άμεσα και με διαφάνεια, με αποτέλεσμα να μην επηρεάζεται η εκτέλεση των εργασιών οι οποίες έχουν ήδη δρομολογηθεί σε κάποιο υπολογιστή του πλέγματος.

2.7 Οι πολιτικές του SGEEE

Πέρα από το One Grid Engine (SGE) η εταιρία Sun έχει δημιουργήσει και το Sun Grid Engine Enterprise Edition (SGEEE) [3], [4]. Η διαφορά τους έγκειται στο ότι το SGE Enterprise Edition χρησιμοποιείται σε δομές που υλοποιούν επιχειρησιακά ή καθολικά υπολογιστικά πλέγματα. Το SGEEE περιέχει όλες τις λειτουργίες και τα χαρακτηριστικά του SGE. Δηλαδή, αποτελεί αναβαθμισμένη έκδοση του Sun Grid Engine. Η κοινή βάση του αποδεικνύεται από το γεγονός ότι το SGE μπορεί να μετατραπεί σε SGEEE αναβαθμίζοντας το master host.

To Sun Grid Engine Enterprise Edition εφαρμόζει συγκεκριμένες πολιτικές οι οποίες υπαγορεύουν τον τρόπο με τον οποίο πρέπει να γίνει η κατανομή των υπολογιστικών πόρων στα έργα ή στις ομάδες που χρησιμοποιούν το πλέγμα. Οι πολιτικές αυτές, που ουσιαστικά είναι και η βασική διαφορά ανάμεσα σε SGE και SGEEE, είναι οι εξής:

1. Πολιτική Share Based ή Share Tree Based.

Η πολιτική *share based* ή *share tree based* κατανέμει ένα ποσοστό των υπολογιστικών πόρων σε κάθε χρήστη ή σε κάθε έργο, για τις ανάγκες του οποίου χρησιμοποιείται το υπολογιστικό πλέγμα. Αν το άθροισμα των πόρων που χρησιμοποιεί κάποιος χρήστης ή κάποιο έργο σε κάποιο συγκεκριμένο χρονικό διάστημα ξεπεράσει το ποσοστό που αρχικά του είχε αποδοθεί, θα πρέπει οι επιπλέον πόροι να επιστραφούν στους υπόλοιπους χρήστες. Δηλαδή, αν για το έργο A δεν χρησιμοποιούνται όλοι οι πόροι που του έχουν αποδοθεί, είναι δυνατό να χρησιμοποιηθεί μέρος τους για την εκτέλεση του έργου B. Όταν όμως το για το έργο A χρησιμοποιηθούν όλοι οι πόροι που του αντιστοιχούν θα πρέπει το έργο B να αποδώσει στο A ένα τμήμα των πόρων που χρησιμοποίησε. Το ποσοστό των πόρων που πρέπει να επιστρέψει το B εξαρτάται από το χρονικό διάστημα που αξιοποίησε τους πόρους του A. Γενικά

ισχύει ότι όσο περισσότερο χρονικό διάστημα χρησιμοποίησε τους πόρους του Α τόσο μικρότερο ποσοστό πρέπει να επιστρέψει.

2. Πολιτική Functional

Η πολιτική *functional* είναι παρόμοια με τη *share tree based*. Η μοναδική διαφορά τους έγκειται στο γεγονός ότι *functional* ορίζει πως το δανειζόμενο μέρος των πόρων, που έχει δανεισθεί κάποιος από τους χρήστες ή από τις διαφορετικές ομάδες έργων που χρησιμοποιούν το πλέγμα, δεν απαιτείται να επιστραφεί. Δηλαδή, η πολιτική αυτή «ξεχνά», ουσιαστικά, το παρελθόν των διοσοληψιών πόρων μεταξύ των χρηστών του πλέγματος.

3. Πολιτική Override

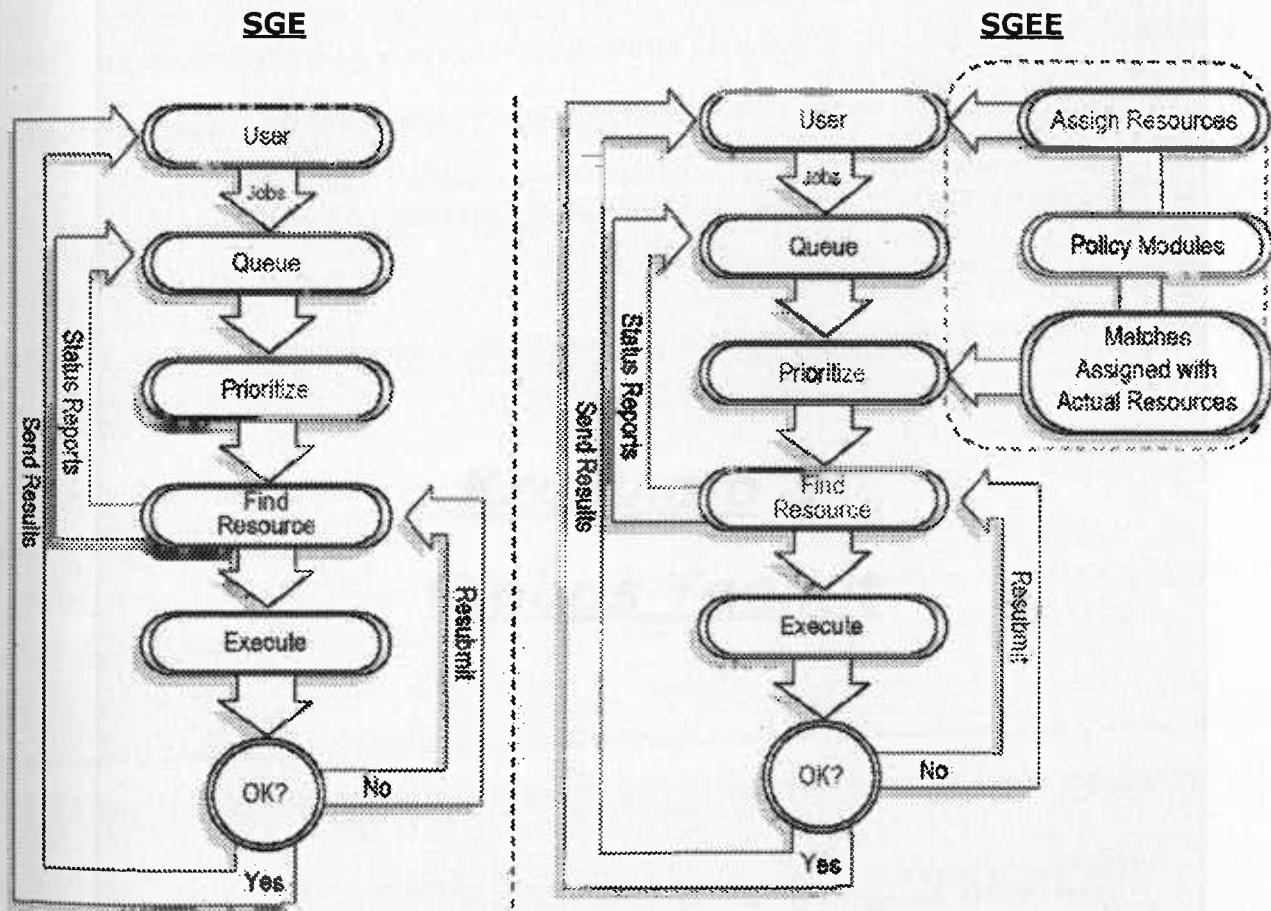
Η πολιτική *override* καθορίζει ότι ο διαχειριστής του συστήματος προσδιορίζει το ποσοστό των υπολογιστικών πόρων που δικαιούται να εκμεταλλευτεί κάθε χρήστης. Δηλαδή, χωρίς κανένα αυτοματισμό και ανάλογα με τις εκάστοτε απαιτήσεις των χρηστών επανακαθορίζεται και η δυνατότητα χρήσης των υπολογιστικών πόρων από αυτούς. Η πολιτική αυτή παρουσιάζει το πλεονέκτημα ότι η ανακατανομή των πόρων δε αφορά μόνο σε χρήστες, ομάδες χρηστών ή συγκεκριμένα έργα που χρησιμοποιούν τις υπηρεσίες του πλέγματος αλλά και σε μεμονωμένες εργασίες ή ομάδες εργασιών οι οποίες έχουν υποβληθεί στο πλέγμα.

4. Πολιτική Deadline

Η πολιτική *deadline* χρησιμοποιείται όταν η εκτέλεση μιας εργασία πρέπει να τελειώσει πριν από κάποια συγκεκριμένη χρονική στιγμή. Ο διαχειριστής του συστήματος προσδιορίζει το επιπλέον ποσοστό των υπολογιστικών πόρων που πρέπει να αποδοθεί προς χρήση στη διαδικασία εκτέλεσης της εργασίας ώστε αυτή να τερματιστεί εγκαίρως. Κίνητρο εφαρμογής αυτής της πολιτικής για μια εταιρία ανάπτυξης εφαρμογών πληροφορικής θα μπορούσε, για παράδειγμα,

να αποτελέσει ο κίνδυνος αθέτησης του συμφωνηθέντος χρόνου παράδοσης μιας εφαρμογής.

Στο σχήμα 5 φαίνεται η ροή των εργασιών στο SGE και στο SGEEE [4].



Σχήμα 5 – Ροή Εργασιών σε SGE και SGEEE

Κεφάλαιο 3^ο
Globus Toolkit

3. The Globus Toolkit

3.1 Γενικά

To Globus αποτελεί το αποτέλεσμα της κοινής προσπάθειας μίας κοινοπραξίας στην οποία συμμετέχουν η IBM, η Microsoft, η DAPRA, η NASA, το υπουργείο ενέργειας και το εθνικό ίδρυμα επιστημών των Η.Π.Α.. Η προσπάθεια αυτή έχει στόχο να καταστήσει δυνατή τη δημιουργία υπολογιστικών πλεγμάτων που θα παρέχουν διεισδυτική, αξιόπιστη και συνεπή πρόσβαση σε υψηλής απόδοσης υπολογιστικούς πόρους, ανεξάρτητα από την γεωγραφική κατανομή των πηγών και των χρηστών.

To Globus είναι μια συλλογή από τμήματα λογισμικού που έχουν σχεδιασθεί με τέτοιο τρόπο ώστε να υποστηρίζουν την ανάπτυξη των εφαρμογών που θα εκτελούνται σε υψηλής απόδοσης κατανεμημένα υπολογιστικά περιβάλλοντα. Κάθε ένα από τα τμήματα λογισμικού παρέχουν μια κύρια υπηρεσία όπως η επικύρωση του χρήστη, η κατανομή των πηγών, η συλλογή πληροφοριών για το σύστημα, η ανακάλυψη λαθών, η επικοινωνία και η απομακρυσμένη πρόσβαση στα δεδομένα.

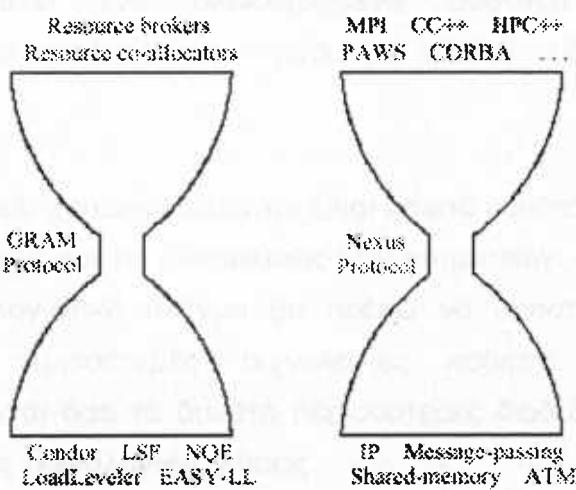
3.2 Η μορφή του Globus

Δομικό στοιχείο του Globus είναι το Globus Metacomputing Toolkit το οποίο ορίζει τις βασικές υπηρεσίες και τις προϋποθέσεις οι οποίες πρέπει να ικανοποιούνται για την υλοποίηση ενός υπολογιστικού πλέγματος. Ο όρος metacomputing [8], περιγράφει και τη μορφή του εργαλείου. Ο όρος αυτός χαρακτηρίζει τις εφαρμογές οι οποίες αποσκοπούν στην υλοποίηση εικονικών υπολογιστών υψηλών επιδόσεων οι οποίοι συνθέτονται από γεωγραφικά κατανεμημένες πηγές που συνδέονται και επικοινωνούν μέσω δικτύων υψηλών ταχυτήτων.

Ο σχεδιασμός του Globus Metacomputing Toolkit στηρίζεται στις εξής αρχές [7]:

- ✓ Το εργαλείο περιλαμβάνει ένα σύνολο από δομικά στοιχεία τα οποία υλοποιούν βασικές υπηρεσίες ασφάλειας, καταμερισμού πηγών, διαχείρισης πόρων και επικοινωνίας. Δηλαδή, προσφέρει ένα σύνολο υπηρεσιών από τις οποίες μπορεί οι σχεδιαστές συγκεκριμένων εργαλείων ή εφαρμογών να χρησιμοποιήσουν ορισμένες ή όλες, ανάλογα με τις ανάγκες τους. Με τον τρόπο αυτό ανταποκρίνεται στη βασική αρχή των υπολογιστικών πλεγμάτων που προβλέπει ότι αυτά θα πρέπει να υποστηρίζουν μεγάλο πλήθος εφαρμογών και προγραμματιστικών μοντέλων.
- ✓ Το εργαλείο διαχωρίζει τις παρεχόμενες υπηρεσίες σε τοπικές και σε καθολικές για κάθε ένα από τα σύνολα υπηρεσιών που περιλαμβάνει (ασφάλεια, επικοινωνία, διαχείριση πόρων κλπ). Οι τοπικές χρησιμοποιούνται για τη διευκόλυνση στην ανάπτυξη μιας εφαρμογής ενώ οι καθολικές τοποθετούνται πάνω από τις τοπικές. Με τον τρόπο αυτό καθίσταται εύκολη η μεταβολή των καθολικών ή δημιουργία νέων αφού δεν επηρεάζει την υποκείμενη υποδομή που

αποτελείται από τις τοπικές. Η αρχιτεκτονική στην οποία βασίζεται η επικοινωνία των τοπικών και των καθολικών υπηρεσιών με τις εφαρμογές που τις χρησιμοποιούν έχει τη μορφή κλεψύδρας. Στο «λαιμό» της κλεψύδρας μια καλά ορισμένη διεπαφή παρέχει ομοιόμορφη πρόσβαση σε διαφορετικές υλοποιήσεις τοπικών υπηρεσιών. Παράλληλα με βάση τη διεπαφή αυτή ορίζονται και οι καθολικές υπηρεσίες. Αυτή αποτελεί και το σημείο εισόδου ενός υπολογιστή στο πλέγμα. Μόνη προϋπόθεση που πρέπει να ικανοποιεί είναι να παρέχει τις υπηρεσίες που είναι ορισμένες στο λαιμό της κλεψύδρας. Στο παρακάτω σχήμα φαίνεται η μορφή αυτή για τις υπηρεσίες επικοινωνιών και διαχείρισης πόρων του Globus [7]:



Σχήμα 6 – Η κλεψύδρα του Globus

- ✓ Οι διεπαφές ορίζονται με τέτοιο τρόπο ώστε να διαχειρίζονται την ετερογένεια των τοπικών και των καθολικών υπηρεσιών. Ονομάζονται ημιδιαφανείς διεπαφές, προσφέροντας δομημένους μηχανισμούς μέσω των οποίων διάφορα εργαλεία και εφαρμογές μπορούν να ανακαλύψουν και να ελέγξουν τη μορφή του υποκείμενου συστήματος. Αυτή η ημιδιαφάνεια έχει σημαντικά

πλεονεκτήματα γιατί δίνει τη δυνατότητα στις καθολικές υπηρεσίες να αντιλαμβάνονται τα χαρακτηριστικά των τοπικών με αποτέλεσμα να προσαρμόζουν τη συμπεριφορά του στις υποκείμενες υπηρεσίες.

- ✓ Οι υπηρεσίες πληροφορίας είναι αναπόσπαστο τμήμα του εργαλείου. Τα υπολογιστικά πλέγματα λειτουργώντας υπό ένα καθεστώς αστάθειας, καθώς ο βαθμός χρήσης τους και η διαθεσιμότητα των πηγών αλλάζει, οι υπολογιστές και τα δίκτυα παρουσιάζουν δυσλειτουργίες, παλιά συστήματα αφαιρούνται, νέα προστίθενται και τα υπάρχοντα ανανεώνονται, απαιτείται να ανακαλύπτουν δυναμικά και κάθε στιγμή την κατάσταση που επικρατεί στους κόλπους τους. Για το λόγο αυτό το Globus metacomputing toolkit περιλαμβάνει ένα ολοκληρωμένο σύστημα συλλογής και επεξεργασίας πληροφοριών μέσω του οποίου ελέγχει τη λειτουργία του.
- ✓ Το εργαλείο χρησιμοποιεί όταν είναι εφικτό υφιστάμενα πρότυπα για τις διεπαφές και τις υλοποιήσεις των υπηρεσιών. Το γεγονός ότι σε ένα υπολογιστικό πλέγμα θα πρέπει να υποστηρίζονται όσο το δυνατό περισσότερες τεχνολογίες καθιστά απαραίτητο να υιοθετούνται όσο το δυνατό περισσότερες διαδεδομένες και κοινά αποδεκτές τεχνολογικές λύσεις.



Η χρήση του GRAM API προσδιορίζει ότι στοπίζεται την απομείωση των πόρων. Σε ποσό να καρρέσουν με δύσκολη της γνώσης καθαρόμασιν πόρουν ΡΕΣ (Resources Extraction Language), διάλογο, ή RSL γνήσιας ποικιλίας και την ακριβότερη μέσω των εφάρμογών θεωρήσεων χρήσης των πόρων των

3.3 Τα δομικά στοιχεία του Globus

Στη συνέχεια παρουσιάζονται τα δομικά στοιχεία του Globus Metacomputing Toolkit:

3.3.1 Διαχείριση Πόρων

Το Globus είναι δομημένο με βάση μια πολυεπίπεδη αρχιτεκτονική. Στο κατώτερο επίπεδό της βρίσκεται ο διαχειριστής κατανομής πόρων, που ονομάζεται GRAM (Globus Resource Allocation Manager), ο οποίος παρέχει τις τοπικές υπηρεσίες για τη διαχείριση των υπολογιστικών πόρων του πλέγματος. Κάθε GRAM είναι υπεύθυνο για ένα σύνολο πόρων που λειτουργεί με βάση την ίδια πολιτική κατανομής. Δηλαδή, ένα πλέγμα που υλοποιείται με το Globus μπορεί να περιέχει περισσότερα από ένα GRAMs, καθένα από τα οποία είναι υπεύθυνα για ένα συγκεκριμένο σύνολο υπολογιστικών πόρων.

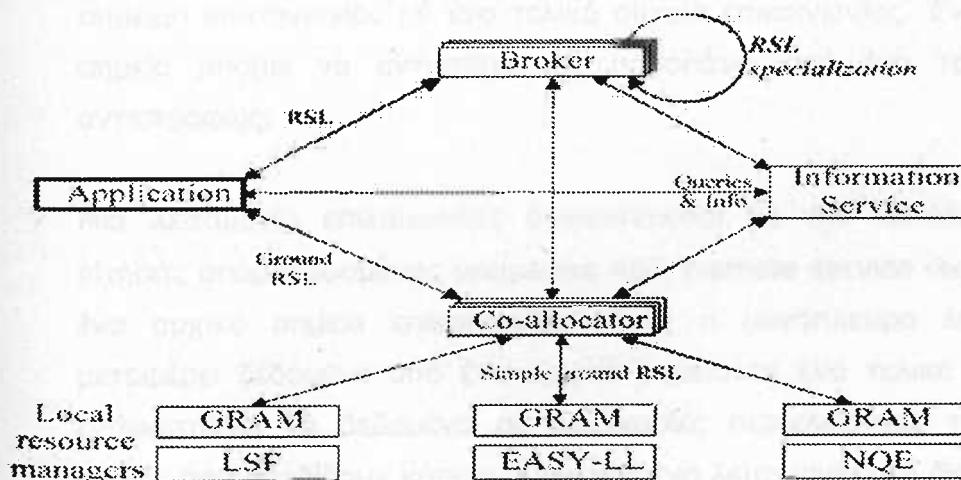
Κάθε GRAM προσφέρει μια πρότυπη διεπαφή στα συστήματα διαχείρισης πόρων. Έτσι, οι εφαρμογές και τα εργαλεία του υπολογιστικού πλέγματος μπορούν να εκφράσουν την κατανομή των πηγών και τη διαδικασία διαχείρισης των αιτήσεων χρήσης τους, σε όρους ενός πρότυπου API, ενώ κάθε τμήμα του πλέγματος δεν δεσμεύεται σχετικά με το εργαλείο διαχείρισης των πηγών που χρησιμοποιεί. Το GRAM μπορεί να συνλειτουργήσει συσχετιζόμενο με αρκετά διαφορετικά εργαλεία. Τα σημαντικότερα από αυτά είναι το Network Queuing Environment (NQE), το EASY – LL, το Load Sharing Facility (LSF), το LoadLeveler και το Condor [9].

Η χρήση του GRAM API προσδιορίζει ότι οι αιτήσεις για χρήση των πόρων θα πρέπει να εκφράζονται με όρους της γλώσσας καθορισμού πόρων RSL (Resource Specification Language). Δηλαδή, η RSL χρησιμοποιείται ώστε να εκφρασθούν μέσω των εφαρμογών οι αιτήσεις χρήσης των πόρων του

πλέγματος. Οι αιτήσεις αυτές λαμβάνονται από ένα πλήθος μεσιτών (*brokers*) πόρων οι οποίοι ουσιαστικά διαβάζουν τις αιτήσεις που είναι γραμμένες σε γλώσσα RSL και προσδιορίζουν τα τμήματα των πόρων και τους αντίστοιχους GRAMs, που μπορούν να τις εξυπηρετήσουν.

Το τελικό βήμα για την ολοκλήρωση της διαδικασίας κατανομής των πόρων είναι να διαιρεθεί η αίτηση που είναι γραμμένη σε RSL σε ένα σύνολο υπο-αιτήσεων, καθεμία από τις οποίες, θα αποσταλεί στον κατάλληλο GRAM. Στο σημείο αυτό, σε περιπτώσεις που απαιτούνται υψηλές αποδόσεις, κρίνεται αναγκαίο να υπάρχει μια διαδικασία ταυτόχρονης κατανομής των ίδιων πηγών σε περισσότερες από μια αιτήσεις. Το Globus ανταποκρίνεται στην απαίτηση αυτή περιλαμβάνοντας ένα *συν-κατανεμητή* (*co-allocator*) πηγών. Αυτός χωρίζει τις αιτήσεις που είναι γραμμένες σε RSL σε κομμάτια, τα κατανέμει στους GRAMs και εναρμονίζει τις επιστρεφόμενες τιμές. Μέχρι σήμερα έχουν υλοποιηθεί δυο τέτοιοι συν-κατανεμητές. Ο ένας προβλέπει ότι εάν οποιαδήποτε από τις πηγές που απαιτούνται δεν είναι διαθέσιμη τότε η προσπάθεια *συν-κατανομής* (*co-allocation*) αποτυγχάνει. Ο δεύτερος επιτρέπει τμήματα των αιτήσεων RSL να μεταβάλλονται μέχρι να πραγματοποιηθεί η συν-κατανομή.

Στο σχήμα 7 αποτυπώνεται η διαδικασία διαχείρισης των πόρων [9].



Σχήμα 7 – Διαχείριση πόρων στο Globus

3.3.2 Επικοινωνίες

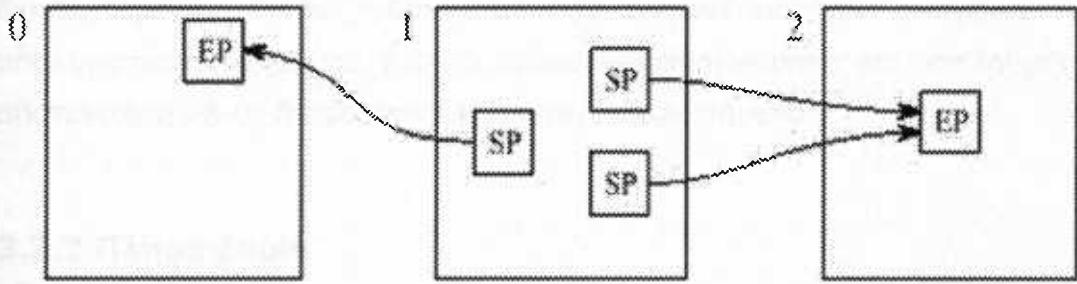
Οι υπηρεσίες επικοινωνιών που περιλαμβάνει το Globus παρέχονται από μια βιβλιοθήκη που ονομάζεται Nexus. Το Nexus ορίζει ένα χαμηλού επιπέδου API, το οποίο χρησιμοποιείται να υποστηρίξει ένα πλήθος βιβλιοθηκών και γλωσσών υψηλότερου επιπέδου. Επίσης, οι υπηρεσίες αυτές χρησιμοποιούνται και για την υλοποίηση άλλων εφαρμογών του Globus.

Οι ανάγκες επικοινωνίας των υπολογιστικών πλεγμάτων είναι πολύ υψηλές και περιλαμβάνουν από μεταφορά μηνυμάτων από σημείο σε σημείο μέχρι αναξιόπιστες επικοινωνίες multicast. Το γεγονός αυτό καθιστά απαραίτητη τη δυνατότητα χρήσης πολλών, αν όχι όλων, των τεχνολογικών επικοινωνίας που υπάρχουν. Γι' αυτό το Nexus ορίζει μια διεπαφή που είναι σχεδιασμένη με τέτοιο τρόπο ώστε να υποστηρίζει ένα πλήθος πρωτοκόλλων επικοινωνίας χαμηλότερου επιπέδου και να παρέχει τη δυνατότητα χρήσης εργαλείων υψηλότερου επιπέδου. Η διεπαφή αυτή ονομάζεται Nexus.

Η επικοινωνία στο Nexus ορίζεται σύμφωνα με 2 βασικές έννοιες:

- ✓ Μια σύνδεση επικοινωνίας σχηματίζεται με τη ένωση ενός αρχικού σημείου επικοινωνίας με ένα τελικό σημείο επικοινωνίας. Ένα αρχικό σημείο μπορεί να αντιστοιχεί σε παραπάνω από ένα τελικά και αντιστρόφως.
- ✓ Μια λειτουργία επικοινωνίας αρχικοποιείται με την κατάθεση μιας αίτησης απομακρυσμένης υπηρεσίας RSR (remote service request) σε ένα αρχικό σημείο επικοινωνίας. Αυτή η μονόπλευρη λειτουργία μεταφέρει δεδομένα από ένα αρχικό σημείο σε ένα τελικό και τότε ενσωματώνει τα δεδομένα σε διαδικασίες περικλείοντας τα τελικά σημεία που αποδίδουν κάποια συγκεκριμένη λειτουργία στη διαδικασία.

Το παρακάτω σχήμα 8 δείχνει το μηχανισμό επικοινωνίας που ορίζει το Nexus. Στο σχήμα φαίνονται τρεις διαδικασίες, τρεις συνδέσεις επικοινωνίας και τρία αρχικά σημεία. Στη διαδικασία 1 αντιστοιχούν τα αρχικά σημεία ενώ τα τελικά στις διαδικασίες 0 και 2 [10].



Σχήμα 8 - Nexus

Οι συνδέσεις επικοινωνίας μπορούν να σχεδιασθούν με διάφορες μεθόδους ανάλογα με τις απαιτήσεις απόδοσης που πρέπει να ικανοποιούν. Οι μέθοδοι αυτές δεν αφορούν μόνο στα πρωτόκολλα επικοινωνίας αλλά και σε άλλες έννοιες όπως η ασφάλεια, η αξιοπιστία και η ποιότητα των υπηρεσιών. Μια εφαρμογή μπορεί να επιλέξει διαφορετικές μεθόδους σε καθεμία από τις συνδέσεις που χρησιμοποιεί. Για παράδειγμα, μια εφαρμογή, για την εκτέλεση της οποίας κάποιες επικοινωνίες πρέπει να είναι αξιόπιστες και κάποιες άλλες γρήγορες, μπορεί να δημιουργήσει δύο συνδέσεις ανάμεσα σε δύο διαδικασίες, με τη μια να προσφέρει υψηλή αξιοπιστία παρουσιάζοντας, όμως, χαμηλή ταχύτητα μεταφοράς και την άλλη να παρέχει υψηλές ταχύτητες αλλά αναξιόπιστη μεταφορά δεδομένων.

Οι δυνατότητες αυτές που προσφέρει το Globus για την επιλογή διαφορετικών μεθόδων δεν θα ήταν χρήσιμη αν το εργαλείο δεν πρόσφερε ένα ολοκληρωμένο μηχανισμό συλλογής και επεξεργασίας πληροφοριών σχετικά με τις τεχνολογίες δικτύων και τα πρωτόκολλα που χρησιμοποιούνται από ένα υπολογιστικό πλέγμα. Οι πληροφορίες αυτές δίνουν τη δυνατότητα σε

εφαρμογές και εργαλεία να ανακαλύψουν τις διαθέσιμες μεθόδους και να επιλέξουν εκείνες που ανταποκρίνονται καλύτερα στους στόχους τους.

Το Nexus αποτυπώνει τον τρόπο που οι υπηρεσίες του Globus χρησιμοποιούν ημιδιαφανείς διεπαφές αφού μια εφαρμογή έχει τη δυνατότητα να εκφράσει όλες τις λειτουργίες της με όρους ενός απλού ομοιόμορφου API, διαχειριζόμενη με τον τρόπο αυτό την ετερογένεια των στοιχείων του υπολογιστικού πλέγματος. Έτσι, οι εφαρμογές είναι φορητές και εκτελούμενες αποτελεσματικά σε διαφορετικά υπολογιστικά συστήματα.

3.3.3 Πληροφορία

Η δυναμική φύση των υπολογιστικών πλεγμάτων συνεπάγεται ότι τα εργαλεία στα οποία στηρίζουν τη λειτουργία τους και οι εκτελούμενες εφαρμογές πρέπει να έχουν τη δυνατότητα να προσαρμόζονται στις αλλαγές που συντελούνται στη δομή και στην κατάσταση του συστήματος. Για το λόγο αυτό το Globus περιλαμβάνει το Metacomputing Directory Service (MDS) το οποίο είναι σχεδιασμένο να παρέχει ένα περιβάλλον πλούσιο σε πληροφορίες που αναφέρονται στην κατάσταση των τμημάτων του πλέγματος σε κάθε χρονική στιγμή και οι οποίες είναι συνεχώς διαθέσιμες προς χρήση. Αυτό αποθηκεύει και παρέχει πρόσβαση σε πληροφορίες σχετικά με την αρχιτεκτονική, το λειτουργικό σύστημα και το μέγεθος της μνήμης ενός υπολογιστή, το bandwidth του δικτύου και την καθυστέρηση που αυτό παρουσιάζει, τα διαθέσιμα πρωτόκολλα επικοινωνίας και την αντιστοιχία μεταξύ των διευθύνσεων IP και του δικτύου.

Το MDS παρέχει ένα σύνολο από εργαλεία και APIs για την ανακάλυψη και τη δημοσίευση των πληροφοριών σχετικά με τη δομή και την κατάσταση του υπολογιστικού πλέγματος. Χρησιμοποιεί το API και τη μέθοδο παρουσίασης των δεδομένων που ορίζονται από το πρωτόκολλο LDAP (Lightweight Directory Access Protocol). Το LDAP ορίζει ένα ιεραρχικό, δενδροειδούς

μορφής χώρο ονομάτων (*namespace*) το οποίο ονομάζεται δένδρο καταλόγου πληροφορίας (*directory information tree*) και έχει μορφή κατανεμημένης υπηρεσίας με βάση την οποία αυθαίρετα τμήματα υποδένδρων μπορούν να αντιστοιχηθούν σε διαφορετικούς εξυπηρετητές. Έτσι, η απαιτούμενη τοπική υπηρεσία που υποστηρίζει το MDS είναι ένας LDAP εξυπηρετητής (ή μια πύλη σε ένα άλλο LDAP εξυπηρετητή, εάν πολλά τμήματα του πλέγματος μοιράζονται ένα εξυπηρετητή) και οι λειτουργίες που απαιτούνται για την ενημέρωσή του με πληροφορίες σχετικά με την τρέχουσα κατάσταση του τμήματος του πλέγματος στο οποίο αντιστοιχεί. Η καθολική υπηρεσία MDS είναι το σύνολο όλων των εξυπηρετητών.

Τη διαδικασία συλλογής και επεξεργασίας των πληροφοριών του πλέγματος το Globus την έχει ενσωματώσει στη λειτουργία όλων των δομικών του στοιχείων. Κάθε υπηρεσία του είναι υπεύθυνη για την παραγωγή πληροφοριών οι οποίες μπορεί να είναι χρήσιμες στους χρήστες, στις εφαρμογές που εκτελούνται αλλά και στις υπόλοιπες υπηρεσίες. Για παράδειγμα κάθε GRAM έχει ενσωματωμένη μια λειτουργία που λέγεται *GRAM reporter* η οποία είναι υπεύθυνη για τη συλλογή και τη δημοσίευση πληροφοριών σχετικά με τον τύπο των πηγών που διαχειρίζονται, τη διαθεσιμότητά τους, τις τεχνολογίες που χρησιμοποιούν κ.λ.π.

3.3.4 Ασφάλεια

Η υποδομή της ασφάλειας η οποία αναπτύχθηκε για την αρχική μορφή του Globus σχετίστηκε μόνο με την επίλυση του προβλήματος της αυθεντικοποίησης. Δηλαδή, της διαδικασίας με βάση την οποία μια οντότητα αναγνωρίζει την ταυτότητα μιας άλλης οντότητας. Η επικέντρωση μόνο σε αυτό το πρόβλημα οφείλεται στο γεγονός ότι η επίλυση του θα δημιουργήσει τη βάση για την επίλυση των υπολοίπων προβλημάτων της ασφαλείας όπως η εξουσιοδότηση και η κρυπτογράφηση.

Οι λύσεις για την αυθεντικοποίηση στα υπολογιστικά πλέγματα πρέπει να ανταποκρίνονται σε δύο προβλήματα που δεν προσδιορίζονται από τις συνηθισμένες λύσεις που χρησιμοποιούνται. Το πρώτο είναι η υποστήριξη της τοπικής ετερογένειας. Οι πόροι του πλέγματος χρησιμοποιούνται από διάφορες οντότητες, καθεμία από τις οποίες έχει διαφορετικό πλαίσιο διαχείρισης (*administrative domain*). Καθένα από αυτά τα πλαίσια έχει τις δικές του απαιτήσεις σε σχέση με τον τρόπο απόδειξης της αυθεντικότητας και απόδοσης εξουσιοδότησης. Συνεπώς, τα πλαίσια αυτά έχουν και διαφορετικές τοπικές λύσεις, πολιτικές και μηχανισμούς ασφάλειας, όπως είναι το σύστημα Kerberos και το Secure-Shell, τα οποία θέτουν περιοριστικούς φραγμούς σχετικά με τις επιλογές που πρέπει να γίνουν αναφορικά με την ασφάλεια ολόκληρου του πλέγματος.

Το δεύτερο πρόβλημα που πρέπει να αντιμετωπιστεί από τις λύσεις ασφαλείας του Globus είναι η ανάγκη να υποστηρίζονται πλαίσια ασφαλείας για κάθε δυνατή επικοινωνία. Επειδή σε ένα υπολογιστικό πλέγμα η εκτέλεση μιας λειτουργίας περιλαμβάνει την επικοινωνία και συνλειτουργία πολλών εφαρμογών που εκτελούνται σε διαφορετικούς υπολογιστές απαιτείται οι κανόνες ασφαλείας να εφαρμόζονται για οποιαδήποτε επικοινωνία εγκαθίσταται μεταξύ δύο διαδικασιών.

Αυτά τα δύο πρόβληματα μορφοποιούνται και την πολιτική ασφαλείας του Globus. Αυτή καθορίζει ότι για να αντιμετωπισθεί το πρόβλημα της ετερογένειας σε κάθε πηγή του πλέγματος διατηρείται μια τοπική ταυτότητα κάθε χρήστη. Ενώ για την επίλυση του δεύτερου προβλήματος προβλέπει ότι ένας χρήστης πρέπει αποδεικνύει την αυθεντικότητά του μόνο μια φόρα στα πλαίσια μιας εφαρμογής την οποία εκτελεί. Αφού γίνει η απόδειξη αυτή δημιουργείται ένα πιστοποιητικό το οποίο επιτρέπει στις διαδικασίες που προέρχονται από το συγκεκριμένο χρήστη να χρησιμοποιούν τους πόρους χωρίς άλλη παρέμβαση από το χρήστη.

Η πολιτική ασφαλείας του Globus υλοποιείται μέσω της *υποδομής ασφαλείας του Globus GSI (Globus Security Infrastructure)*. Το GSI έχει σχεδιασθεί, όπως και τα υπόλοιπα στοιχεία του Globus, με τέτοιο τρόπο ώστε διάφορες καθολικές υπηρεσίες να βρίσκονται πάνω από μια απλή τοπική.

Ο εσωτερικός σχεδιασμός του GSI δίνει έμφαση στο σημαντικό ρόλο που διαδραματίζουν τα πρότυπα στον προσδιορισμό των υπηρεσιών και των εργαλείων του πλέγματος. Διάφορα από τα προβλήματα τα οποία πρέπει να επιλύσει το GSI δεν υποστηρίζονται από ένα υπάρχον σύστημα ασφαλείας. Παρ' όλα αυτά, η δυνατότητά του να αλληλεπιδρά με άλλα συστήματα, να ανεξαρτητοποιείται από χαμηλού-επιπέδου μηχανισμούς και να εκμεταλλεύεται των υπάρχοντα κώδικα οφείλεται στο γεγονός ότι όλοι οι αλγόριθμοι ασφαλείας έχουν γραφεί με βάση το πρότυπο GSS (Generic Security Service). Το πρότυπο αυτό ορίζει μια πρότυπη διαδικασία και ένα API για την απόκτηση πιστοποιητικών (password ή certificates), για την αμοιβαία απόδειξη της αυθεντικότητας (μεταξύ client – server), για την κρυπτογράφηση και την αποκρυπτογράφηση. Το GSS είναι ανεξάρτητο από κάθε μηχανισμό ασφαλείας και μπορεί να χρησιμοποιηθεί από διαφορετικές μεθόδους ασφαλείας.

Το GSI υποστηρίζει 2 μηχανισμούς οι οποίοι είναι αποδεκτοί από τη διεπαφή του GSS. Ο πρώτος είναι ένα σύστημα κωδικών ο οποίος υλοποιεί τον τύπο αυθεντικοποίησης rlogin του Unix. Τα πλεονεκτήματά του είναι ότι εύκολο να υλοποιηθεί και δεν επηρεάζεται από εξωτερικούς παράγοντες. Ο δεύτερος μηχανισμός χρησιμοποιεί το την κρυπτογράφηση δημόσιου κλειδιού και βασίζεται στο πρωτόκολλο αυθεντικοποίησης που ορίζεται από το SSL. Αυτή η υλοποίηση έχει το πλεονέκτημα ότι παρέχει υψηλότερα επίπεδα ασφαλείας και συνλειτουργεί με διάφορες διαδεδομένες υπηρεσίες, όπως το LDAP και το HTTP.

3.3.5 Έλεγχος κατάστασης

Το Globus περιλαμβάνει την υπηρεσία Heartbeat Monitor (HBM) η οποία παρέχει απλούς μηχανισμούς για τον έλεγχο της κατάστασης ενός κατανεμημένου συνόλου διαδικασιών. Στην αρχιτεκτονική του HBM περικλείονται μια διεπαφή πελάτη (*client-interface*) και ένα API για συλλογή δεδομένων. Η διεπαφή επιτρέπει σε μια διαδικασία να εκτελεσθεί μαζί με την υπηρεσία HBM, η οποία αναμένει να λάβει σήματα κανονικής λειτουργίας από τη διαδικασία. Αν δεν λάβει ένα από τα σήματα που αναμένει, η HBM προσπαθεί να αντιληφθεί αν το σφάλμα οφείλεται στη διαδικασία ή σε κάποιο τμήμα του δικτύου ή σε κάποιο υπολογιστή. Το API επιτρέπει σε άλλη διαδικασία να συλλέξει πληροφορίες σχετικά με την κατάσταση της εκτελούμενης διαδικασίας. Αυτές οι πληροφορίες μπορούν να χρησιμοποιηθούν για τη δημιουργία μηχανισμών ανίχνευσης και διόρθωσης σφαλμάτων. Η υπηρεσία HBM χρησιμοποιείται για τον έλεγχο της κατάστασης βασικών δομικών στοιχείων του Globus όπως το GRAM και το MDS.

3.3.6 Απομακρυσμένη πρόσβαση σε αρχεία

Η υπηρεσία του Globus που παρέχει τη δυνατότητα πρόσβασης σε απομακρυσμένα αρχεία ονομάζεται *καθολική πρόσβαση σε δευτερεύουσες αποθήκες* (*Global Access to Secondary Storage (GASS)*). Η υπηρεσία αυτή επιτρέπει προγράμματα που χρησιμοποιούν την πρότυπη βιβλιοθήκη I/O της C να ανοίξει και να επεξεργασθεί αρχεία που βρίσκονται σε απομακρυσμένους υπολογιστές. Τα αρχεία όταν ανοιχθούν, αντιγράφονται σε προσωρινά τοπικά αρχεία ώστε αφενός να επιτρέπεται η επεξεργασία τους χωρίς συνεχή σύνδεση αφετέρου να αποφεύγεται η συνεχής μεταφορά του ίδιου αρχείου.

3.3.7 Διαχείριση εκτελέσιμων διαδικασιών

Το Globus περιλαμβάνει μια ακόμη υπηρεσία, την Globus Executable Management (GEM), η οποία υποστηρίζει τη δημιουργία, την ταυτοποίηση και την τοποθέτηση εκτελέσιμων αρχείων σε ετερογενή περιβάλλοντα. Παρέχει μηχανισμούς προσαρμογής των χαρακτηριστικών ενός υπολογιστή, που συμμετέχει σε ένα υπολογιστικό πλέγμα, με τις απαιτήσεις εκτέλεσης ενός αρχείου.

3.3.8 Κιβώτιο Υπηρεσιών (Service Container)

Τα δομικά αυτά στοιχεία του Globus που ουσιαστικά αποτελούν τις βασικές υπηρεσίες που παρέχει κάθε υπολογιστικό πλέγμα που αναπτύσσεται με βάση το Globus τα διαχειρίζεται το *κιβώτιο των υπηρεσιών υπολογιστικού πλέγματος* (*Grid Service Container*). Αυτό εκτός από τη διαχείριση των υπηρεσιών (προσδιορισμό κύκλου ζωής, αντιστοίχιση υπηρεσιών υποβαλλόμενων αιτήσεων) υλοποιώντας τις αρχές της OGSA, που θα παρουσιάσουμε στο επόμενο κεφάλαιο, διαμορφώνει μια υποδομή πάνω στην οποία είναι δυνατόν να δομηθούν εφαρμογές που ανεξάρτητα από τις τεχνολογίες που αξιοποιούν έχουν τη δυνατότητα να εκμεταλλευτούν τις υπηρεσίες του υπολογιστικού πλέγματος.

Κεφάλαιο 4^ο

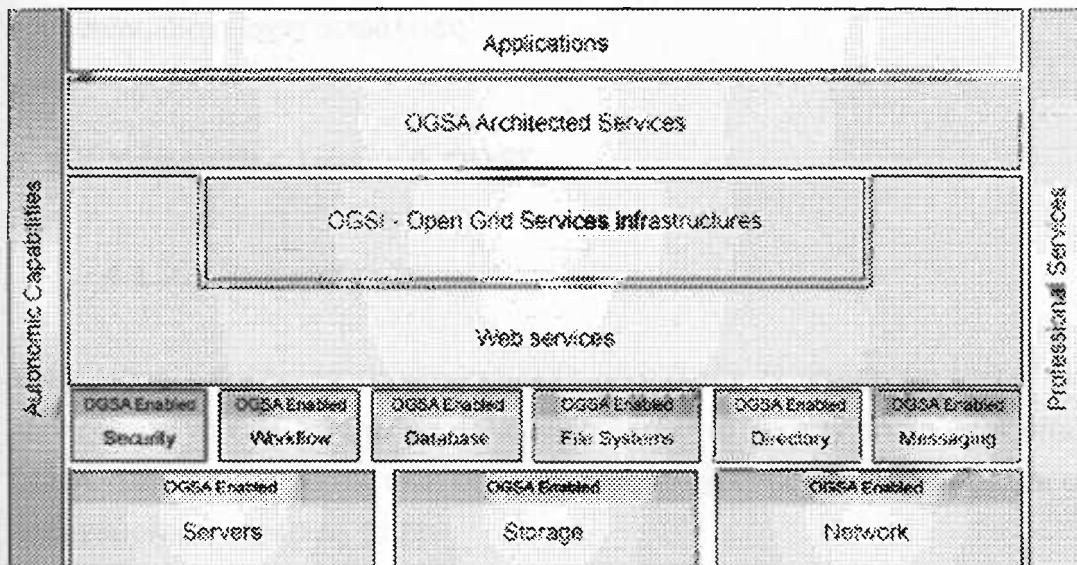
Open Grid Service Architecture

4. Open Grid Service Architecture

Η ανάγκη για την υλοποίηση ενός ολοκληρωμένου τρόπου λειτουργίας ενός υπολογιστικού πλέγματος, ανεξάρτητα από τη διαφορετική φιλοσοφία των τεχνολογιών που συνυπάρχουν στα πλαίσια του, οδήγησε στην δημιουργία μιας αρχιτεκτονικής που ονομάστηκε Open Grid Service Architecture. Η αρχιτεκτονική αυτή, η οποία παρουσιάστηκε τον Ιούνιο του 2002 στο Global Grid Forum και υιοθετήθηκε για πρώτη φορά για την υλοποίηση του Globus Toolkit 3, στοχεύει στα εξής:

1. Διαμόρφωση ενιαίου τρόπου διαχείριση υπολογιστικών πόρων οι οποίοι είναι κατανεμημένοι σε ετερογενείς πλατφόρμες.
2. Διαμόρφωση μιας κοινής βάσης για αυτόνομες λύσεις διαχείρισης πόρων. Επειδή ένα υπολογιστικό πλέγμα περιλαμβάνει πολλούς υπολογιστικούς πόρους, διαφορετικών τεχνολογιών, οι οποίοι αλληλεπιδρούν, είναι απαραίτητη η ύπαρξη δυνατότητας αυτοδιαχείρισης τους.
3. Ορισμός ανοιχτών, δημοσιευμένων διεπαφών. Για την αλληλεπίδραση των διάφορων πόρων, τα πλέγματα πρέπει να δημιουργούνται με βάση συγκεκριμένες διεπαφές και πρωτόκολλα.
4. Εκμετάλλευση των πρότυπων τεχνολογιών. Οι δημιουργοί του OGSA χρησιμοποίησαν, όσο το δυνατό περισσότερο, πρότυπες τεχνολογίες. Θεμέλιο τη αρχιτεκτονικής αποτελούν οι υπηρεσίες ιστού (Web Services).

4.1 Περιγραφή της Αρχιτεκτονικής



Σχήμα 9 - OGSA

Όπως φαίνεται και στο σχήμα 9 η αρχιτεκτονική OGSA αποτελείται από τέσσερα επίπεδα, ξεκινώντας από κάτω τα εξής:

1. Πόροι
2. Υπηρεσίες ιστού & OGSI
3. Δημιουργικές Υπηρεσίες OGSA
4. Εφαρμογές υπολογιστικού πλέγματος

4.1.1 Πόροι

Οι υπολογιστικοί πόροι, που αποτελούν το βασικό στοιχείο ενός πλέγματος, διαχωρίζονται σε φυσικούς και λογικούς. Στους φυσικούς πόρους περιλαμβάνονται εξυπηρετητές, αποθηκευτικά μέσα και δίκτυα. Πάνω από τους φυσικούς υπάρχουν οι λογικοί. Αυτοί ομογενοποιούν τους φυσικούς και

ουσιαστικά τους κάνουν διαθέσιμους στα υπόλοιπα στοιχεία που συνθέτουν το πλέγμα. Λογικοί πόροι είναι τμήματα λογισμικού που υλοποιούν εργασίες όπως σύστημα διαχείρισης αρχείων, διαχείριση βάσεων δεδομένων και ροής εργασιών, διαχείριση ασφάλειας.

4.1.2 Υπηρεσίες Ιστού & OGSI

4.1.2.1 Υπηρεσίες Ιστού

Το δεύτερο επίπεδο της αρχιτεκτονικής αποτελούν οι υπηρεσίες ιστού (web service) και το OGSI. Αυτά αποτελούν τη βασική ίδέα στην οποία βασίζεται το OGSA. Σύμφωνα με αυτή όλοι οι πόροι ενός πλέγματος αντιμετωπίζονται σαν παρεχόμενες υπηρεσίες. [12]

Οι υπηρεσίες ιστού είναι μια νέα γενιά των εφαρμογών ιστού. Είναι ανεξάρτητες, περιγράφουν των εαυτό τους, είναι τμηματικές εφαρμογές που μπορούν να δημοσιευθούν, να βρεθούν, και να κληθούν από τον ιστό. Εκτελούν τις λειτουργίες, οι οποίες μπορούν να είναι οτιδήποτε, από τα απλά αιτήματα έως περίπλοκες επιχειρησιακές διαδικασίες. Μόλις επεκταθεί μια υπηρεσία ιστού, άλλες εφαρμογές (και άλλες υπηρεσίες ιστού) μπορούν να ανακαλύψουν και να καλέσουν την δημοσιευμένη υπηρεσία.

Η βασική πλατφόρμα των υπηρεσιών ιστού είναι η XML συν το HTTP. Το HTTP είναι ένα πανταχού παρόν πρωτόκολλο, που τρέχει σχεδόν παντού στο διαδίκτυο. Η XML παρέχει μια μεταγλώσσα στην οποία μπορείτε να γράψετε τις εξειδικευμένες γλώσσες για να εκφράσετε τις σύνθετες αλληλεπιδράσεις μεταξύ των χρηστών και των υπηρεσιών ή μεταξύ των συστατικών μιας σύνθετης υπηρεσίας. Πίσω από την πρόσωψη ενός κεντρικού υπολογιστή δικτύου, το μήνυμα XML μετατρέπεται σε ένα αίτημα ενδιάμεσου λογισμικού (middleware) και τα αποτελέσματα επιστρέφονται πίσω σε XML. Τα πράγματα

βέβαια δεν είναι τόσο απλά, υπάρχει ένα σύνολο εργαλείων και πρωτοκόλλων που κάνουν τη παραπάνω πλατφόρμα πραγματική πλατφόρμα.

Η πλήρους λειτουργίας πλατφόρμα υπηρεσιών ιστού μπορεί να θεωρηθεί ως XML συν το HTTP συν το SOAP συν το WSDL συν το UDDI. Σε πιο υψηλά επίπεδα, κάποιος μπορεί επίσης να προσθέσει τεχνολογίες όπως XAML, XLANG, XKMS, και XFS -- υπηρεσίες που δεν γίνονται αποδεκτές παγκοσμίως ως υποχρεωτικές.

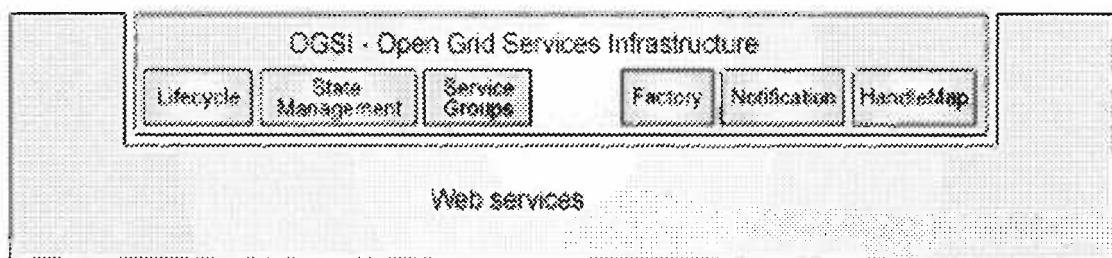
4.1.2.2 Open Grid Service Infrastructure (OGSI)

Το OGSI (Open Grid Services Infrastructure) είναι μια προδιαγραφή που χρησιμοποιείται για τον ορισμό των υπηρεσιών του πλέγματος. Χρησιμοποιεί μηχανισμούς που αξιοποιούνται για την υλοποίηση υπηρεσιών ιστού όπως η XML (Extensible Markup Language) και WSDL (Web Service Description Language) ώστε να προσδιορίσει πρότυπες διεπαφές και τρόπους αλληλεπίδρασης μεταξύ όλων των υπολογιστικών πόρων. Πιο συγκεκριμένα, το OGSI επεκτείνει τις υπηρεσίες ιστού παρουσιάζοντας διεπαφές και συμβάσεις που αφορούν σε δύο βασικούς τομείς.

1. Οι υπηρεσίες ενός υπολογιστικού πλέγματος παρουσιάζουν μια δυναμική και συνήθως βραχυπρόθεσμη μορφή. Δηλαδή, μια υπηρεσία κάθε χρονική στιγμή είναι δυνατό να μεταλλάσσεται ανάλογα με την εργασία που καλείται να εξυπηρετήσει, με τον τρόπο που οι πόροι συντονίζονται και με τη μεταβολή της κατάστασης του συστήματος. Για το λόγο αυτό, οι υπηρεσίες του πλέγματος χρειάζονται διεπαφές ώστε να διαχειρίζονται τη δημιουργία τους, τον κύκλο ζωής τους και την παύση της λειτουργίας τους.
2. Οι υπηρεσίες ενός πλέγματος έχουν χαρακτηριστικά και δεδομένα μέσω των οποίων επικοινωνούν. Αυτό είναι παρόμοιο με την παραδοσιακή δομή των αντικειμένων στον αντικειμενοστραφή προγραμματισμό σύμφωνα με την οποία αυτά έχουν χαρακτηριστικά και δεδομένα.

Αντίστοιχα, οι υπηρεσίες ιστού πρέπει να διαμορφώνονται με τέτοιο τρόπο ώστε να υποστηρίζουν την ανταλλαγή δεδομένων με τις υπηρεσίες υπολογιστικού πλέγματος.

Οι διεπαφές και οι συμβάσεις που το OGSI περιλαμβάνει, όπως φαίνονται και στο σχήμα 10, είναι οι εξής:



Σχήμα 10 - OGSI

- ✓ **Εργοστασίο (Factory):** Οι υπηρεσίες πλέγματος που υλοποιούν αυτή τη διεπαφή παρέχουν δυνατότητα δημιουργίας νέων υπηρεσιών. Τα εργοστάσια μπορούν να δημιουργούν είτε προσωρινές υπηρεσίες περιορισμένων λειτουργιών, όπως ένας προγραμματιστής (scheduler) υλοποιεί μια υπηρεσία ώστε να παρουσιάσει την εκτέλεση συγκεκριμένης εργασίας, είτε υπηρεσίες με μεγαλύτερο χρόνο ζωής όπως ένα τοπικό αντίγραφο ενός συχνά χρησιμοποιούμενου συνόλου δεδομένων. Βέβαια, δεν δημιουργούνται όλες οι υπηρεσίες δυναμικά. Για παράδειγμα, κάποια μπορεί να είναι το αποτέλεσμα της ύπαρξης ενός φυσικού πόρου.
- ✓ **Κύκλος ζωής (Life cycle):** Επειδή οι υπηρεσίες υπολογιστικού πλέγματος μπορεί να είναι προσωρινές, μια υπηρεσία έχει συγκεκριμένη διάρκεια ζωής. Αυτός ο χρόνος ζωής μπορεί να επεκταθεί ανάλογα με τις απαιτήσεις των στοιχείων που εξαρτώνται ή διαχειρίζονται την υπηρεσία αυτή. Ο μηχανισμός του κύκλου ζωής διαμορφώθηκε έτσι ώστε να εμποδίσει τις υπηρεσίες να σπαταλήσουν πόρους χωρίς να είναι

απαραίτητη για το σκοπό αυτό η ύπαρξη ενός μεγάλης κλίμακας, κατανεμημένου συλλέκτη σκουπιδιών-υπηρεσιών (scavenger).

- ✓ **Διαχείριση Κατάστασης (State Management):** Το OGSI προσδιορίζει ένα πλαίσιο παρουσίασης της κατάστασης μιας υπηρεσίας που ονομάζεται Δεδομένα Υπηρεσίας (Service Data) και ένα μηχανισμό για την εξέταση ή την τροποποίηση της που ονομάζεται Βρες/Θέσεις τα Δεδομένα Υπηρεσίας (Find/SetServiceData). Επίσης, το OGSI απαιτεί αφενός ένα ελάχιστο σύνολο χαρακτηριστικών κατάστασης στα Στοιχεία Δεδομένων Υπηρεσίας (Service Data Elements) τα οποία κάθε υπηρεσία του πλέγματος πρέπει να υποστηρίζει, αφετέρου όλες οι υπηρεσίες να υλοποιούν το μηχανισμό Βρες/Θέσεις τα Δεδομένα Υπηρεσίας (Find/SetServiceData).
- ✓ **Ομάδες Υπηρεσίας (Service Groups):** Οι ομάδες υπηρεσίες είναι συλλογές υπηρεσιών οι οποίες έχουν συγκεκριμένη ένδειξη και χρησιμοποιούνται για κάποιο σαφώς καθορισμένο σκοπό. Για παράδειγμα, μπορούν να χρησιμοποιηθούν για τη συλλογή των υπηρεσιών που απεικονίζουν τους πόρους σε ένα συγκεκριμένο κόμβο του πλέγματος.
- ✓ **Γνωστοποίηση (Notification):** Οι πληροφορίες κατάστασης (Δεδομένα Υπηρεσίας) μιας υπηρεσίας αλλάζουν καθ' όλη τη διάρκεια λειτουργίας του πλέγματος. Πολλές αλληλεπιδράσεις που συμβαίνουν μεταξύ των υπηρεσιών απαιτούν δυναμικό έλεγχο της αλλαγής της κατάστασης. Η Γνωστοποίηση εφαρμόζει ένα παραδοσιακό παράδειγμα δημοσίευσης/προσυπογραφής (publish/subscribe) για την εφαρμογή του ελέγχου αυτού. Οι υπηρεσίες πλέγματος υποστηρίζουν μια διεπαφή με ονομασία Πηγή Γνωστοποίησης (NotificationSource) που επιτρέπει σε άλλες υπηρεσίες να μαθαίνουν και να προσυπογράφουν τις αλλαγές κατάστασης μιας υπηρεσίας.

- ✓ **Χάρτης Χειρισμού (HandleMap):** Κάθε φορά που τα εργοστάσια χρησιμοποιούνται για τη δημιουργία μιας νέας υλοποίησης μια υπηρεσίας, επιστρέφουν την ταυτότητα του νέου στιγμιότυπου της. Αυτή η ταυτότητα αποτελείται από δύο τμήματα:
 1. Χειριστήριο Υπηρεσίας Υπολογιστικού Πλέγματος (Grid Service Handle (GSH))
 2. Αναφορά Υπηρεσίας Υπολογιστικού Πλέγματος (Grid Service Reference (GSR))

Ένα Χειριστήριο (GSH) παραπέμπει πάντα στην αντίστοιχη υπηρεσία, ενώ μια Αναφορά (GSR) μπορεί να αλλάξει κατά τη διάρκεια του χρόνου ζωής της αντίστοιχης υπηρεσίας. Η διεπαφή Χάρτη Χειρισμού (HandleMap) παρέχει τον τρόπο με τον οποίο θα μπορεί το σύστημα να λάβει μια Αναφορά (GSR) για μια υπηρεσία εφόσον δηλώσει το αντίστοιχο Χειριστήριο (GSH).

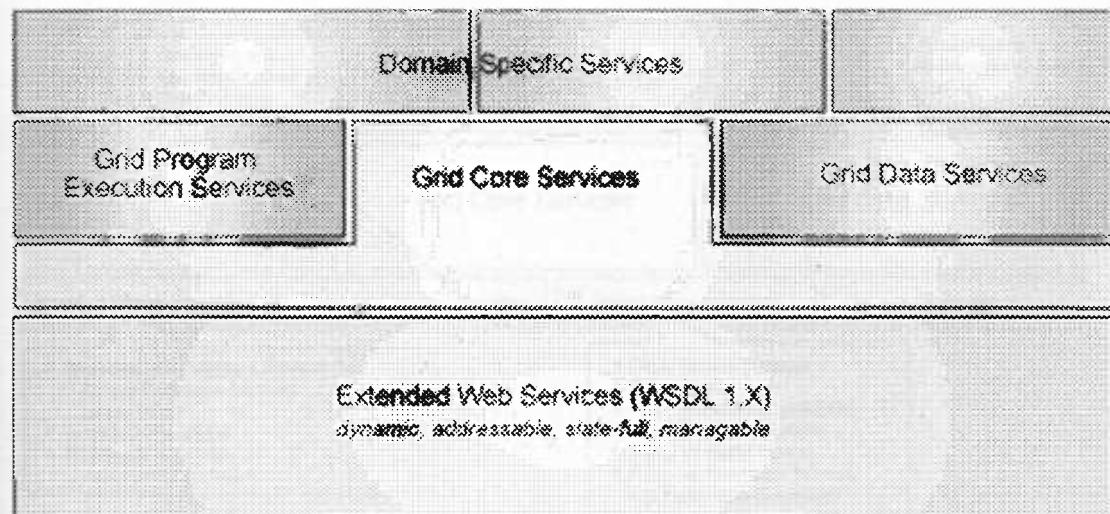
4.1.3 Δημιουργικές Υπηρεσίες OGSA

Το επίπεδο των υπηρεσιών ιστού παρέχει την υποδομή που αποτελεί τη βάση διαμόρφωσης του επόμενου επιπέδου των δημιουργικών υπηρεσιών του OGSA². Αυτές αποτελούν ένα πλούσιο σύνολο υπηρεσιών πλέγματος και μπορούν να κατηγοριοποιηθούν ως εξής (σχήμα 11):

- ✓ **Υπηρεσίες Πυρήνα Υπολογιστικού Πλέγματος (Grid Core Services)**
- ✓ **Υπηρεσίες εκτέλεσης προγραμμάτων Υπολογιστικού Πλέγματος (Grid Program Execution Services)**

² To Global Grid Forum ασχολείται, σήμερα, με την υλοποίηση αυτών των υπηρεσιών.

- ✓ **Υπηρεσίες Δεδομένων Υπολογιστικού Πλέγματος (Grid Data Services)**
- ✓ **Υπηρεσίες Ειδικού-Πλαισίου (Domain-Specific Services)**



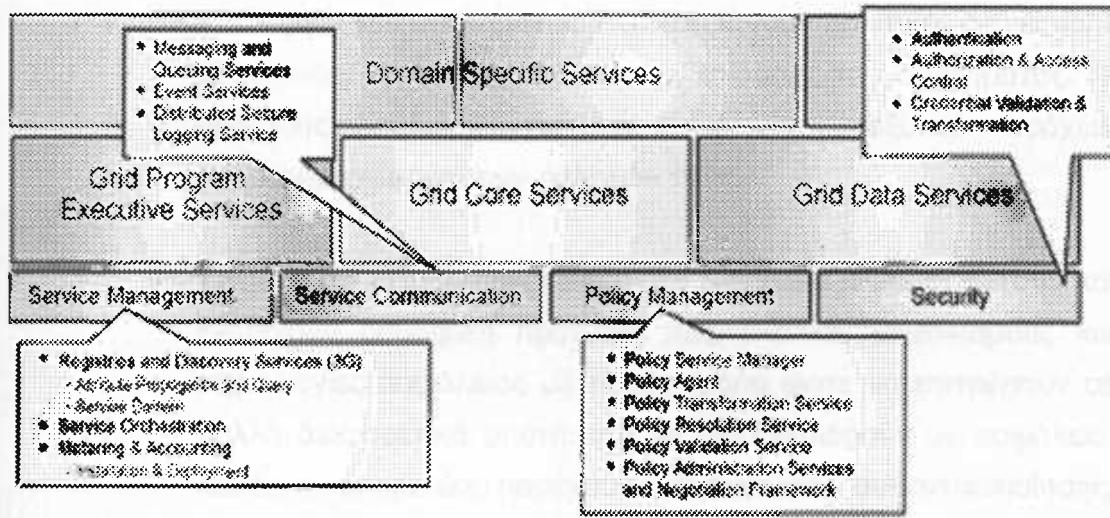
Σχήμα 11 – Δημιουργικές Υπηρεσίας OGSA

Ο τρεις πρώτες κατηγορίες αποτελούν την περιοχή της τρέχουσας δουλειάς και έρευνας το Global Grid Forum. Στόχο της δουλειάς αυτής αποτελεί η ολοκλήρωση και η ωρίμανση των υπηρεσιών αυτών ώστε στη συνέχεια, αφού έχει διαμορφωθεί η βάση, να προσδιοριστούν οι υπηρεσίες Ειδικού-Πλαισίου.

Οι βασικές προδιαγραφές που θα διέπουν τις υπηρεσίες των τριών πρώτων κατηγοριών αναλύονται στη συνέχεια.

1. Υπηρεσίες Πυρήνα Υπολογιστικού Πλέγματος: Η ονομασία τους προέρχεται από το γεγονός ότι αυτές θα τις εκμεταλλευτούν όλες οι υπηρεσίες υψηλότερου επιπέδου υποστηρίζοντας είτε προγράμματα εκτέλεσης εργασιών, είτε πρόσβασης σε δεδομένα, είτε υπηρεσίες ειδικής περιοχής.

Οι υπηρεσίες πυρήνα διαχωρίζονται με βάση τέσσερις διαφορετικούς τύπους (σχήμα 12):



Σχήμα 12 – Τύποι Υπηρεσιών Πυρήνα

- Υπηρεσία Διαχείρισης (Service Management):** παρέχει λειτουργίες οι οποίες διαχειρίζονται τις υπηρεσίες που αναπτύσσονται στα πλαίσια του κατανεμημένου πλέγματος. Αυτοματοποιεί τη διαχείρισή τους και συντελεί στην ομαλή λειτουργία του πλέγματος παρέχοντας πληθώρα μηχανισμών εγκατάστασης, συντήρησης, ελέγχου και επίλυσης βλαβών.
- Υπηρεσία Επικοινωνίας (Service Communication):** παρέχει ένα μεγάλο αριθμό λειτουργιών οι οποίες υποστηρίζουν τις βασικές μεθόδους επικοινωνίας μεταξύ των υπηρεσιών του πλέγματος. Υποστηρίζουν διάφορα πρότυπα επικοινωνίας τα

οποία επιτρέπουν την αποτελεσματική δια-υπηρεσιακή επικοινωνία.

- **Υπηρεσία Διαχείρισης Πολιτικής (Policy Management):** δημιουργεί ένα γενικό πλαίσιο για την ανάπτυξη, τη διοίκηση και τη διαχείριση πολιτικών και συμφωνιών που σχετίζονται με τη λειτουργία του συστήματος. Περιλαμβάνουν πολιτικές διαχείρισης ασφάλειας, κατανομής πόρων και επίδοσης του συστήματος. Οι όροι τους προσδιορίζονται βάσει συμφωνίας μεταξύ των παρόχων των υπηρεσιών και των πελατών τους.
- **Υπηρεσίες Ασφάλειας (Security Services):** υποστηρίζουν και ενοποιούν δημοφιλή πρότυπα, πρωτόκολλα, μηχανισμούς και τεχνολογίες ασφάλειας με τέτοιο τρόπο ώστε να επιτρέπουν σε πολλά διαφορετικά συστήματα να αλληλεπιδρούν με ασφάλεια. Αυτές οι υπηρεσίες παρέχουν μηχανισμούς αυθεντικοποίησης, εξουσιοδότησης, μεταφοράς πιστοποιητικών κ.λ.π.

2. Υπηρεσίες Εκτέλεσης Προγραμμάτων Υπολογιστικού Πλέγματος:

Ενώ το OGSI και οι υπηρεσίες πυρήνα είναι εφαρμόσιμες σε κάθε κατανεμημένο υπολογιστικό σύστημα. Η κατηγορία των υπηρεσιών εκτέλεσης προγραμμάτων, όμως, αφορά μόνο στο πρότυπο του υπολογιστικού πλέγματος της κατανεμημένης εκτέλεσης εργασιών, η οποία υποστηρίζει υψηλής απόδοσης υπολογισμούς, παράλληλη επεξεργασία και κατανεμημένη συνεργασία πόρων.

3. Υπηρεσίες Δεδομένων Υπολογιστικού Πλέγματος: Οι υπηρεσίες αυτές διαμορφώνουν τα δεδομένα που χρησιμοποιούνται από το πλέγμα. Παρέχουν μηχανισμούς πρόσβασης σε πληροφορίες διαφορετικών τύπων όπως για παράδειγμα το περιεχόμενο βάσεων δεδομένων, αρχείων, κειμένων.

Κεφάλαιο 5^ο
Υλοποίηση

5. ΥΛΟΠΟΙΗΣΗ

Το πρακτικό τμήμα της εργασίας είναι δυνατό να διαχωριστεί σε τρεις τομείς οι οποίοι προσδιορίζονται με βάση τα δύο εργαλεία τα οποία χρησιμοποιήθηκαν και την εφαρμογή-υπηρεσία που αναπτύχθηκε :

1. Εγκατάσταση του Sun Grid Engine

2. Εγκατάσταση του Globus Toolkit 3

3. Δημιουργία Υπηρεσίας πλέγματος

Στη συνέχεια παρατίθεται η διαδικασία που ακολουθήθηκε για την εγκατάσταση των δύο εργαλείων και για την ανάπτυξη τής εφαρμογής. Το κείμενο αυτό δεν αποτελεί εγχειρίδιο εγκατάστασης του κάθε εργαλείου αλλά μια περιγραφή της.

5.1 Εγκατάσταση Sun Grid Engine

Η εγκατάσταση του SGE πραγματοποιήθηκε σε δύο υπολογιστές του Εργαστηρίου Συστημάτων Υπολογιστών και Επικοινωνιών του Οικονομικού Πανεπιστημίου Αθηνών. Τα χαρακτηριστικά τους είναι:

1. CPU: Pentium I 100 Mhz, Ram: 64 MB
2. CPU: Pentium I 100 Mhz, Ram: 64 MB

Ο ένας διαδραμάτιζε ρόλο master host και ο άλλος ρόλο execution host. Στους δύο υπολογιστές έγινε εγκατάσταση του λειτουργικού συστήματος Linux, έκδοση Debian. Στη συνέχεια ξεκίνησε η διαδικασία εγκατάστασης του Sun One Grid Engine. Αυτή περιλαμβάνει τα εξής βήματα:

1. Λήψη του αρχείου εγκατάστασης του εργαλείου από την ιστοσελίδα της εταιρίας Sun (<http://www.sun.com/gridware>) και αποθήκευση του στο σκληρό δίσκο του υπολογιστή που προορίζεται να λειτουργεί ως master host. Η αποθήκευση πρέπει να γίνει σε συγκεκριμένο κατάλογο ο οποίος ονομάζεται κατάλογος του SGE.
2. Δημιουργία ενός χρήστη που έχει δικαιώματα διαχείρισης των υπολογιστών που συμμετέχουν στο πλέγμα. Ο χρήστης αυτός πρέπει να δημιουργηθεί σε όλους τους υπολογιστές.
3. Επειδή το Sun Grid Engine χρησιμοποιεί το πρωτόκολλο TCP για την επικοινωνία μεταξύ των υπολογιστών απαιτείται ο προσδιορισμός της αντίστοιχης θυρίδας (port) μέσω της οποίας θα γίνεται αυτή η επικοινωνία. Για το λόγο αυτό σε κάθε υπολογιστή στο αρχείο `/etc/services` προστέθηκε η παρακάτω γραμμή :

➤ `Sge_commd 536/tcp`

4. Εγκατάσταση Network File System (NFS)

Επειδή για την εγκατάσταση του Sun Grid Engine απαιτείται η χρήση αρχείων που είναι αποθηκευμένα στο master host από τους execution κρίθηκε απαραίτητη η εγκατάσταση του συστήματος NFS. Αυτό παρέχει τη δυνατότητα σε ένα μηχάνημα να προσφέρει ένα τμήμα του σκληρού δίσκου του σε ένα απομακρυσμένο μηχάνημα ώστε να το επεξεργαστεί σαν να ήταν δικό του. Η εγκατάσταση αυτή διαχωρίζεται σε 2 τμήματα:

- ✓ Εγκατάσταση NFS εξυπηρετητή
- ✓ Εγκατάσταση NFS πελάτη
 - Εγκατάσταση NFS εξυπηρετητή

Για την εγκατάσταση αυτή απαιτείται η επεξεργασία 3 αρχείων διαμόρφωσης (configuration files). Αυτά είναι τα:

➤ /etc(exports

Περιλαμβάνει μια λίστα από καταχωρήσεις καθεμία από τις οποίες προσδιορίζει τον κατάλογο ο οποίος διαμοιράζεται, με πλήρη προσδιορισμό αφενός των μηχανημάτων που μπορούν να έχουν πρόσβαση, αφετέρου των δικαιωμάτων που έχουν οι χρήστες τους σε αυτόν.

➤ /etc/host.allow
➤ /etc/host.deny

Τα 2 αυτά αρχεία χρησιμοποιούνται για να προσδιορισθεί ποια μηχανήματα έχουν δυνατότητα χρήσης των υπηρεσιών που παρέχονται μέσω του NFS. Η χρήση τους δεν είναι απαραίτητη για τη λειτουργία του NFS, ωστόσο είναι απαραίτητη για την ασφαλή λειτουργία του.

Για τη λειτουργία του το NFS χρησιμοποιεί έξι δαιμονες (οι τελευταίες εκδόσεις του Linux τους περιλαμβάνουν εξ' ορισμού):

- portmap
- mountd
- nfsd
- statd
- lockd
- rquotad

Η επιβεβαίωση της λειτουργίας του NFS εξυπηρετητή γίνεται με την εκτέλεση της εντολής **rpcinfo -p** το αποτέλεσμα της οποία θα πρέπει να είναι παρόμοιο με τον παρακάτω πίνακα (σχήμα 13)

program	vers	proto	port	
100000	2	tcp	111	portmapper
100000	2	udp	111	portmapper
100011	1	udp	749	rquotad
100011	2	udp	749	rquotad
100005	1	udp	759	mountd
100005	1	tcp	761	mountd
100005	2	udp	764	mountd
100005	2	tcp	766	mountd
100005	3	udp	769	mountd
100005	3	tcp	771	mountd
100003	2	udp	2049	nfs
100003	3	udp	2049	nfs
300019	1	tcp	830	amd
300019	1	udp	831	amd
100024	1	udp	944	status
100024	1	tcp	946	status
100021	1	udp	1042	nlockmgr
100021	3	udp	1042	nlockmgr
100021	4	udp	1042	nlockmgr
100021	1	tcp	1629	nlockmgr
100021	3	tcp	1629	nlockmgr
100021	4	tcp	1629	nlockmgr

Σχήμα 13 – Ένδειξη Λειτουργίας NFS

- Εγκατάσταση NFS Πελάτη

Για να χρησιμοποιηθεί ένας υπολογιστής ως NFS πελάτης απαιτείται να εκτελούνται σε αυτόν οι δαίμονες **portmap**, **lockd** και **statd** οι οποίοι εκτελούνται κατά την εκκίνηση του λειτουργικού Linux.

Η χρησιμοποίηση του τμήματος του NFS εξυπηρετητή από τον πελάτη γίνεται με την εκτέλεση της παρακάτω εντολής στον πελάτη (υποθέτοντας ότι ο εξυπηρετητής καλείται anduin.ccslab.aueb.gr και ο κατάλογος που θέλουμε να παραχωρήσουμε στον πελάτη είναι ο /home):

➤ `mount anduin.ccslab.aueb.gr:/home /mnt/home`

5. Εγκατάσταση του Master Host

Αποτελεί τον πυρήνα του πλέγματος από τον οποίο ελέγχεται το Sun Grid Engine. Εκτελεί τον δαίμονα `sge_qmaster`. Οι απαιτήσεις υλικού για το master εξαρτώνται από το μέγεθος της συστοιχίας (cluster) υπολογιστών που χρησιμοποιούνται. Δηλαδή, αν ο αριθμός του είναι μερικές δεκάδες 10MB ελεύθερης μνήμης είναι αρκετά. Αν, όμως, ο αριθμός τους ξεπερνά τους 1000 υπολογιστές απαιτείται 1GB ελεύθερης μνήμης. Στην παρούσα περίπτωση δεν τίθεται θέμα μνήμης αφού το μέγεθος της συστοιχίας (cluster) είναι πολύ μικρό.

Για την ολοκλήρωση της εγκατάστασης του master host, αφού οδηγηθούμε στον κατάλογο (directory) του SGE, χρησιμοποιούμε την παρακάτω εντολή ώστε να μπορέσουμε να χρησιμοποιήσουμε το αρχείο εγκατάστασης το οποίο έχει μορφή .tar.

➤ `tar -xvzf αρχείο_εγκατάστασης.tar`

Στη συνέχεια εκτελούμε το πρόγραμμα εγκατάστασης του master host.

➤ `./install_qmaster`

6. Εγκατάσταση του Execution host.

Για την εγκατάσταση του execution host απαιτείται να προσπελάσουμε από τον υπολογιστή που προορίζεται να διαδραματίσει το ρόλο αυτό τον κατάλογο SGE που βρίσκεται στο master host. Η δυνατότητα αυτή παρέχεται από το Network File System το οποίο έχει ήδη εγκατασταθεί. Αφού καταστεί δυνατή, μέσω NFS, η χρήση του καταλόγου SGE από τον execution host εκτελούμε το αντίστοιχο πρόγραμμα εγκατάστασης.

➤ `./install_exceed`

Στον πίνακα 2 παρουσιάζονται συνοπτικά τα βήματα εγκατάστασης του Sun Grid Engine.

Πίνακας Εγκατάστασης Sun Grid Engine

Βήμα	Περιγραφή
1	Λήψη του Sun Grid Engine από το διαδίκτυο
2	Δημιουργία χρήστη διαχείρισης του SGE
3	Καθορισμός της θυρίδας επικοινωνίας
4	Εγκατάσταση Network File System
5	Εγκατάσταση Master Host
6	Εγκατάσταση Execution Host

Πίνακας 2 – Βήματα Εγκατάστασης SGE

5.2 Εγκατάσταση του Globus Toolkit 3

Η εγκατάσταση του Globus Toolkit 3 πραγματοποιήθηκε σε δύο υπολογιστές του Εργαστηρίου Συστημάτων Υπολογιστών και Επικοινωνιών του Οικονομικού Πανεπιστημίου Αθηνών. Τα χαρακτηριστικά τους είναι:

1. CPU: Pentium III 650 Mhz, Ram: 256 MB
2. CPU: Pentium I 100 Mhz, Ram: 64 MB

Ο πρώτος προοριζόταν να διαδραματίσει ρόλο εξυπηρετητή και ο άλλος ρόλο πελάτη. Στους δύο υπολογιστές έγινε εγκατάσταση του λειτουργικού συστήματος Linux, έκδοση Debian. Στη συνέχεια ξεκίνησε η διαδικασία εγκατάστασης του Globus. Αξίζει να τονισθεί ότι η διαδικασία εγκατάστασης είναι ίδια για όλους τους υπολογιστές που θα συμμετάσχουν στο πλέγμα ανεξάρτητα από το ρόλο που θα διαδραματίσουν. Η διαδικασία αυτή περιγράφεται στη συνέχεια.

1. Εγκατάσταση της Java (JDK 1.4.2).

Για την εγκατάσταση της Java απαιτήθηκε η λήψη του JDK από την ιστοσελίδα της εταιρίας Sun. Στη συνέχεια έγινε αποθήκευση και αποσυμπίεση του αρχείου σε κάποιο κατάλογο του σκληρού δίσκου ο οποίος δηλώνεται ως \$JAVA_HOME. Για την ολοκλήρωση της εγκατάστασης γράφουμε στη μεταβλητή περιβάλλοντος το \$JAVA_HOME/bin. Η Java απαιτείται για τη δημιουργία των υπηρεσιών πλέγματος.

2. Εγκατάσταση Postgresql.

Απαιτήθηκε η εγκατάσταση ενός συστήματος διαχείρισης βάσεων δεδομένων που να είναι συμβατό με την τεχνολογία JDBC. Το σύστημα

που επιλέχθηκε είναι η **postgresql**. Η εγκατάστασή της σε Linux περιβάλλον γίνεται εύκολα με την εντολή

➤ **apt-get install postgresql**

Η εγκατάσταση ενός συστήματος βάσεων δεδομένων που να είναι συμβατό με την τεχνολογία JDBC χρειάζεται για τη λειτουργία ορισμένων υπηρεσιών του Globus.

3. Εγκατάσταση του εργαλείου Jakarta Ant.

Για την εγκατάσταση του εργαλείου πραγματοποιήθηκε λήψη του αρχείου από την ιστοσελίδα <http://jakarta.apache.org/ant>, αποσυμπίεση του σε κάποιο κατάλογο του σκληρού δίσκου και ορισμό του καταλόγου αυτού ως \$ANT_HOME. Επίσης, απαιτήθηκε πρόσθεση του \$ANT_HOME/bin στη μεταβλητή περιβάλλοντος PATH. Το εργαλείο αυτό αποτελεί ένα Java εργαλείο χτισίματος (Java build tool). Η λειτουργία του είναι παρόμοια με την εντολή make. Επιτρέπει στους προγραμματιστές να παρακάμψουν τα βήματα που πρέπει να εκτελεσθούν, ξεχωριστά, ώστε να δημιουργηθούν από τα πηγαίου κώδικα αρχεία τα εκτελέσιμα. Τα βήματα αυτά αναλαμβάνει να τα εκτελέσει το Ant. Τα ανεξάρτητα αυτά βήματα περιγράφονται σε ένα αρχείο το οποίο ονομάζεται αρχείο χτισίματος (build file) και προσδιορίζει πλήρως τι πρέπει να μεταγλωπιστεί, με ποιο τρόπο και με ποια σειρά. Δηλαδή, συγχωνεύει όλα τα βήματα σε ένα.

4. Εγκατάσταση του Globus Toolkit

Για την εγκατάσταση του Globus Toolkit απαιτείται η εκτέλεση των εξής βήμάτων:

- Λήψη του πακέτου εγκατάστασης από την ιστοσελίδα <http://www-unix.globus.org/toolkit/download.html>
- Άνοιγμα του αρχείου που λαμβάνεται από την ιστοσελίδα και εκτέλεση του αρχείου εγκατάστασης του Globus
 - `./install-gt3 /μονοπάτι καταλόγου εγκατάστασης`
- Απόκτηση πιστοποιητικών για κάθε υπολογιστή του πλέγματος και για κάθε χρήστη που χρησιμοποιεί τις υπηρεσίες του. Η διαδικασία αυτή απαιτεί τη ρύθμιση της υπηρεσίας πιστοποιητικών του Globus
http://gcs.globus.org:8080/gcs/setup_ca.html
 και τη λήψη από τις ιστοσελίδες
<http://gcs.globus.org:8080/gcs/usercert.html> και
<http://gcs.globus.org:8080/gcs/servicecert.html> των πιστοποιητικών για τους χρήστες και τους υπολογιστές.
- Εκτέλεση του αρχείου `./install-gt3-mmjfs` μέσω της εντολής
 - `./install-gt3-mmjfs /μονοπάτι καταλόγου εγκατάστασης`
- Στη συνέχεια απαιτείται η ρύθμιση κάποιων αρχείων που χρησιμοποιούνται για τον προσδιορισμό των χρηστών που θα έχουν πρόσβαση στις υπηρεσίες του πλέγματος.

Στον πίνακα 3 παρουσιάζονται συνοπτικά τα βήματα εγκατάστασης του Globus Toolkit.

Πίνακας Εγκατάστασης Sun Grid Engine

Βήμα	Περιγραφή
1	Εγκατάσταση της Java
2	Εγκατάσταση Postgresql
3	Εγκατάσταση Ant
4	Εγκατάσταση Globus Toolkit
4.1	Λήψη του αρχείου εγκατάστασης από το διαδίκτυο
4.2	Εγκατάσταση του Globus
4.3	Απόκτηση Πιστοποιητικών
4.4	Εκτέλεση του αρχείου ./install-gt3-mmjfs

Πίνακας 3 - Βήματα Εγκατάστασης Globus Toolkit 3

5.3 Παρατηρήσεις Εγκατάστασης

Τα συμπεράσματα και οι παρατηρήσεις που προκύπτουν από τη διαδικασία εγκατάστασης των δύο εργαλείων μπορούν να συνοψιστούν στα εξής:

- 1.** Η πολυπλοκότητα της διαδικασία εγκατάστασης του Sun One Grid Engine, αν συγκριθεί με την πολυπλοκότητα που παρουσιάζει ο τρόπος λειτουργίας του, είναι δυνατό να χαρακτηρισθεί μέτρια. Δεν απαιτείται τίποτα περισσότερο από τη χρήση του πακέτου εγκατάστασης, το οποίο διανέμεται από την ιστοσελίδα της εταιρίας Sun και την εφαρμογή των οδηγιών που το συνοδεύουν. Βέβαια η εφαρμογή τους εμφανίζει σημαντική δυσκολία. Στην περίπτωση μας χρειάστηκε και η εγκατάσταση του NFS (Network File System), ώστε να καταστεί δυνατή η χρησιμοποίηση ενός τμήματος του σκληρού δίσκου του υπολογιστή που ήταν ο εξυπηρετητής του συστήματος από τον υπολογιστή-πελάτη, η οποία αποτελεί μια σχετικά απλή διαδικασία αφού στις σύγχρονες εκδόσεις του Linux περιλαμβάνονται τα βασικά στοιχεία υλοποίησης του NFS και χρειάζονται ελάχιστες ρυθμίσεις για την ενεργοποίηση του.
- 2.** Αντίθετα από το Sun One Grid Engine κατά τη διαδικασία εγκατάστασης του Globus Toolkit 3 εμφανίστηκαν πολλά προβλήματα των οποίων η λύση ήταν χρονοβόρα. Αρχικά, παρουσιάστηκαν δυσκολίες που αφορούσαν στη ρύθμιση των εργαλείων (Java, Ant, Postgresql) που έπρεπε να εγκατασταθούν πριν ξεκινήσει η εγκατάσταση του Globus. Αφού ολοκληρώθηκε η φάση αυτή, η διαδικασία εγκατάστασης του εργαλείου εμφάνισε πολλά και σημαντικά προβλήματα. Πρώτο πρόβλημα, ήταν ότι η διαδικασία αντιγραφής και εγκατάστασης των αρχείων που ενεργοποιείται με την εκτέλεση του προγράμματος ./install-gt3 είναι εξαιρετικά χρονοβόρα. Ιδιαίτερα στον ένα υπολογιστή ο οποίος έχει επεξεργαστή Pentium I 100 Mhz διαρκούσε πολλές ώρες. Χαρακτηριστικό είναι ότι αν η διαδικασία ξεκινούσε το μεσημέρι κάποιας

μέρας ολοκληρωνόταν αργά το βράδυ. Το φαινόμενο αυτό είχε ως αποτέλεσμα κάθε φορά που κατά τη φάση εγκατάστασης εμφανιζόταν κάποιο σφάλμα που αφορούσε σε κάποια ρύθμιση των υπολογίων εργαλείων έπρεπε να διακοπεί η διαδικασία εγκατάστασης, να διορθωθεί το λάθος και στη συνέχεια να επαναληφθεί η διαδικασία. Το πρόβλημα αυτό παρουσιάστηκε και κατά την εκτέλεση του ./install-gt3-mmjfs αλλά και σε περιπτώσεις που έπρεπε να εκτελεσθούν κάποια αρχεία χτισίματος (build file). Παρά το γεγονός ότι αυτό γινόταν με τη χρήση του εργαλείου Ant η διαδικασία χτισίματος διαρκούσε αρκετές ώρες.

Σημαντικό πρόβλημα εμφανίστηκε και κατά τη διαδικασία έκδοσης και ρύθμισης των πιστοποιητικών. Όταν ξεκίνησε η διαδικασία εγκατάστασης η έκδοση των πιστοποιητικών γινόταν με την αποστολή μιας αίτησης στην ηλεκτρονική διεύθυνση ca@globus.org από την οποία στελνόταν σαν απάντηση το πιστοποιητικό. Ωστόσο, τα πιστοποιητικά αυτά είχαν ημερομηνία λήξης ισχύος την 24^η Ιανουαρίου. Η έκδοση νέων πιστοποιητικών γίνεται, πλέον, με νέο τρόπο που απαιτεί την εγκατάσταση στο πλέγμα της υπηρεσίας πιστοποιητικών του Globus (Globus Certificate Service) και εν συνεχείᾳ τη λήψη των πιστοποιητικών, χρηστών και υπολογιστών, μέσω μιας διαδικασίας που εφαρμόζεται στις [ιστοσελίδες](http://gcs.globus.org:8080/gcs/usercert.html) και [και](http://gcs.globus.org:8080/gcs/servicecert.html).

Σημαντικό ρόλο στην εμφάνιση τόσων και σημαντικών προβλημάτων διαδραμάτισε εκτός από την πληθώρα των εργαλείων που απαιτούνται να εγκατασταθούν και να συγχρονιστούν και η αδυναμία των επεξεργαστών να ανταποκριθούν γρήγορα στις εργασίες που καλούνταν να εκτελέσουν και η ελλιπής ανάλυση και περιγραφή του τρόπου εγκατάστασης που παρατηρήθηκε ακόμα και στις ιστοσελίδες του επίσημου διαδικτυακού τόπου του Globus <http://www.globus.org>.

5.4 Εφαρμογή

Στα πλαίσια της εργασίας αναπτύχθηκε και μια υπηρεσία πλέγματος. Η υπηρεσία, η οποία ονομάστηκε MathService, αποτελεί ουσιαστικά μια υπηρεσία ιστού η οποία χρησιμοποιεί τα εργαλεία και την υποδομή την οποία παρέχει το Globus Toolkit 3. Η λειτουργία της δεν διαφέρει από μια υπηρεσία πλέγματος ανωτέρου επιπέδου. Ωστόσο, δεν περιλαμβάνει χαρακτηριστικά όπως η γνωστοποίηση της κατάστασης (notification) της υπηρεσίας ή στοιχεία ασφαλείας. Η διαδικασία ανάπτυξης της εφαρμογής, η οποία υλοποιεί τις μαθηματικές πράξεις μεταξύ δύο αριθμών, περιγράφεται στη συνέχεια:

1. Το πρώτο βήμα για τη δημιουργία μιας υπηρεσίας πλέγματος είναι να προσδιορισθεί η διεπαφή της υπηρεσίας (service interface). Δηλαδή, να καθορισθεί τι είδους λειτουργίες θα παρέχει η υπηρεσία στους χρήστες τους συστήματος. Η διεπαφή αυτή ονομάζεται και τύπος λιμένα (PortType). Για τον προσδιορισμό της χρησιμοποιείται η γλώσσα WSDL (Web Service Description Language). Η δημιουργία του αρχείου σε WSDL που θα περιγράφει την υπηρεσία μπορεί να γίνει είτε με απευθείας συγγραφή του με χρήση της γλώσσας αυτής είτε με αυτόματη δημιουργία μέσω μιας Java διεπαφής. Στην εφαρμογή μας χρησιμοποιήθηκε ο δεύτερος τρόπος.
2. Δημιουργία της Java διεπαφής

```
package gt3tutorial.core.first.impl;
public interface Math
{
    public int add(int a, int b);
    public int subtract(int a, int b);
    public int multiply(int a, int b);
    public float divide(int a, int b);
}
```

Τμήμα Κώδικα 1 – Math.java

3. Αποθήκευση των παραπάνω γραμμών κώδικα σε ένα αρχείο με το όνομα Math.java σε ένα κατάλογο με όνομα \$TUTORIAL_DIR/gt3tutorial/core/first/impl.

4. Δημιουργία του WSDL κώδικα από την παραπάνω διεπαφή με την εκτέλεση των παρακάτω εντολών

```
> javac Math.java
> java org.apache.axis.wsdl.Java2WSDL
-P MathPortType
-S MathService
-I http://localhost/ogsa/services/core/
first/MathService
-n http://first.core.gt3tutorial/Math
gt3tutorial.core.first.impl.Math
```

Τώρα έχει δημιουργηθεί ένα αρχείο το οποίο ονομάζεται MathService.wsdl στον κατάλογο \$TUTORIAL_DIR

5. Για να μπορεί να χρησιμοποιηθεί το αρχείο MathService.wsdl σε περιβάλλον Globus απαιτείται να χρησιμοποιηθεί ένα εργαλείο του Globus Toolkit που ονομάζεται DecorateWSDL. Για το λόγο αυτό εκτελούμε την παρακάτω εντολή:

```
> java org.globus.ogsa.tools.wsdl.DecorateWSDL
$GLOBUS_DIRECTORY/schema/ogsi/ogsi_bindings.wsdl
MathService.wsdl2
```

Πλέον έχει δημιουργηθεί το αρχείο WSDL το οποίο περιγράφει τι μπορεί να κάνει η υπηρεσία (πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαιρεση) χωρίς όμως να έχει καθοριστεί το πώς το επιτυγχάνει.

² &GLOBUS_DIRECTORY: Ο κατάλογος που βρίσκονται τα αρχεία του Globus

6. Δημιουργία των αποκόμματων (stubs).

Τα αποκόμματα (stubs) επιτελούν όλη τη δουλειά για την επικοινωνία μεταξύ της εφαρμογής και του πλέγματος. Για τη δημιουργία τους και στον εξυπηρετητή και στον πελάτη εκτελούμε την παρακάτω εντολή:

➤ **java org.globus.ogsa.tools.wsdl.GSDL2Java
MathService.wsdl**

Τώρα στον κατάλογο \$TUTORIAL_DIR/gt3tutorial/core/first/impl υπάρχει ένα νέος κατάλογος που ονομάζεται Math και περιλαμβάνει τα αποκόμματα. Αυτά είναι κάποια Java αρχεία τα οποία θα χρησιμοποιηθούν στη συνέχεια.

7. Υλοποίηση της υπηρεσίας.

Το αρχείο υλοποίησης το δημιουργούμε μέσα στον κατάλογο \$TUTORIAL_DIR/gt3tutorial/first/impl/ και το ονομάζουμε MathImpl.java. Αρχικά περιλαμβάνει τον καθορισμό των packages και την εισαγωγή κάποιων κλάσεων OGSA οι οποίες είναι απαραίτητες.

```
package gt3tutorial.core.first.impl;
import org.globus.ogsa.impl.ogsi.GridServiceImpl;
import gt3tutorial.core.first.Math.MathPortType;
import java.rmi.RemoteException;
```

Τμήμα Κώδικα 2 – MathImpl.java (1)

Στη συνέχεια ορίζουμε την κλάση MathImpl η οποία είναι η υλοποίηση της υπηρεσίας

```
public class MathImpl extends GridServiceImpl implements  
MathPortType
```

Τμήμα Κώδικα 3 - MathImpl.java (2)

Όπως παρατηρούμε η MathImpl είναι κλάση-γόνος της GridServiceImpl κλάσης. Αυτό αποτελεί απαίτηση αφού όλες οι υπηρεσίες του πλέγματος πρέπει να αναπτύσσονται με βάση αυτή την κλάση η οποία ονομάζεται και κλάση σκελετός. Επίσης, η υπηρεσία μας, υλοποιεί μια διεπαφή που ονομάζεται MathPortType. Η διεπαφή αυτή δημιουργήθηκε στο προηγούμενο βήμα. Αποτελεί, δηλαδή, ένα απόκομμα (Stub). Το αρχείο MathPortType.java είναι παρόμοιο με την διεπαφή της υπηρεσίας που δημιουργήσαμε στο πρώτο βήμα (Math.java). Το MathPortType, όπως και το Math, προσδιορίζει τις μεθόδους που παρέχει η υπηρεσία. Διαφοροποιείται, όμως, καθώς περιέχει κώδικα που το καθιστούν απομακρυσμένη διεπαφή (remote interface).

Επόμενο στάδιο αποτελεί η συγγραφή του κατασκευαστή (constructor) της υπηρεσίας μας. Αυτός θα καλεί τον κατασκευαστή της κλάση GridServiceImpl, ο οποίος λαμβάνει την περιγραφή της υπηρεσίας.

```
public MathImpl()  
{  
super("Simple Math Service");  
}
```

Τμήμα Κώδικα 4 - MathImpl.java (3)

Στη συνέχεια υλοποιούμε τις μεθόδους που προσδιορίζονται στη διεπαφή της υπηρεσίας (πρόσθεση, αφαίρεση, πολλαπλασιασμό, διαιρεση).

```
public int add(int a, int b) throws RemoteException
{
    return a + b;
}
public int subtract(int a, int b) throws RemoteException
{
    return a - b;
}
public int multiply(int a, int b) throws RemoteException
{
    return a * b;
}
public float divide(int a, int b) throws RemoteException
{
    return a / b;
}
```

Τμήμα Κώδικα 5 - MathImpl.java (4)

Στο επόμενο πλαίσιο παρατίθεται ολόκληρο το αρχείο υλοποίησης της υπηρεσίας.

```

package gt3tutorial.core.first.impl;
import org.globus.ogsa.impl.ogs1.GridServiceImpl;
import gt3tutorial.core.first.Math.MathPortType;
import java.rmi.RemoteException;

public class MathImpl extends GridServiceImpl implements
MathPortType
{
public MathImpl()
{
super("Simple Math Service");
}
public int add(int a, int b) throws RemoteException
{
return a + b;
}
public int subtract(int a, int b) throws
RemoteException
{
return a - b;
}
public int multiply(int a, int b) throws
RemoteException
{
return a * b;
}
public float divide(int a, int b) throws
RemoteException
{
return a / b;
}
}

```

Τμήμα Κώδικα 6 - MathImpl.java (ολικό)

8. Ανάπτυξη υπηρεσίας (deployment)

Μέχρι στιγμής έχουμε γράψει όλο τον κώδικα που απαιτείται. Δηλαδή, μια διεπαφή υπηρεσίας, ένα αρχείο WSDL, μια δέσμη από αρχεία αποκόμματα (stub files), υλοποίηση της υπηρεσίας. Για το επόμενο βήμα, που είναι να αναπτύξουμε την υπηρεσία, απαιτείται όσα έχουμε δημιουργήσει να τεθούν σε λειτουργία και να τα

κάνουμε διαθέσιμα μέσω ενός υπηρεσιών πλέγματος - εξυπηρετητή ιστού (Grid Service enabled web server).

Για την ανάπτυξη ρόλο-κλειδί διαδραματίζει ένα αρχείο που ονομάζεται αφηγητής ανάπτυξης (deployment descriptor). Σ' αυτό αναφέρεται πώς ο εξυπηρετητής ιστού πρέπει να δημοσιοποιήσει την υπηρεσία πλέγματος που έχουμε δημιουργήσει (π.χ. αναφέρει ποια θα είναι η διεύθυνση ιστού της υπηρεσίας). Ο αφηγητής ανάπτυξης έχει μορφή WSDD (Web Service Deployment Descriptor). Το αρχείο αυτό είναι το εξής:

```
<?xml version="1.0"?>
<deployment name="defaultServerConfig"
xmlns="http://xml.apache.org/axis/wsdd/"
xmlns:java="http://xml.apache.org/axis/wsdd/
providers/java">
<service name="tutorial/core/first/MathService"
provider="java:RPC">
<parameter name="allowedMethods" value="*"/>
<parameter name="className"
value="gt3tutorial.core.first.impl.MathImpl"/>
</service>
</deployment>
```

Τμήμα Κώδικα 7 – Math.wsdd

Αποθηκεύουμε το αρχείο αυτό στον κατάλογο &TUTORIAL_DIR/gt3tutorial/core/first/ με όνομα Math.wsdd.

Το αρχείο αυτό προσδιορίζει τα εξής:

- Το όνομα της υπηρεσίας `<service name = "tutorial/core/first/MathService" provider="java:RPC">`. Αν συνδέσουμε το όνομα αυτό με τη βασική διεύθυνση των υπηρεσιών πλέγματος θα έχουμε την πλήρη ονομασία της υπηρεσίας μας³

<http://195.251.252.111:8080/ogsa/services/tutorial/core/first/MathService>

- Όλες οι μέθοδοι της υπηρεσίας που δηλώνονται στη διεπαφή πρέπει να είναι διαθέσιμες μέσω του εξυπηρετητή ιστού `<parameter name="allowedMethods" value="*"/>`

³ <http://195.251.252.111> είναι η διεύθυνση του server

- Η Java κλάση MathImpl παρέχει την υλοποίηση της υπηρεσίας. Έτσι όταν ο εξυπηρετητής ιστού λαμβάνει μια αίτηση για τη μέθοδο "add" γνωρίζει ότι θα πρέπει να την εξυπηρετήσει με την κλάση MathImpl.
- ```
<parameter name="className"
 value="gt3tutorial.core.first.impl.MathImpl"/>
```

9. Τελικό βήμα της διαδικασίας είναι να μεταγλωπίσουμε όλα όσα έχουμε φτιάξει και να τα εντάξουμε στον εξυπηρετητή ιστού.

- Αρχικά μεταγλωπίζουμε τα αποκόμματα (stubs).

**javac -sourcepath ./ gt3tutorial/core/first/Math/\*.java**

- Στη συνέχεια μεταγλωπίζουμε το αρχείο υλοποίησης της υπηρεσίας

**javac -sourcepath ./ gt3tutorial/core/first/impl/\*.java**

- Τώρα πρέπει να εντάξουμε όλα τα αποκόμματα σε ένα Jar αρχείο

**jar cvf Math-stub.jar gt3tutorial/core/first/Math/\*.class**

- Το ίδιο και την κλάση υλοποίησης

**jar cvf Math.jar gt3tutorial/core/first/impl/\*.class**

- Μετά δημιουργούμε ένα ειδικό τύπο Jar αρχείων που ονομάζεται GAR (Grid Archive). Με βάση αυτό το αρχείο θα δουλεύει ο εξυπηρετητής ιστού. Για την υλοποίηση του GAR δημιουργούμε ένα κατάλογο με όνομα gar/ και αντιγράφουμε τα παρακάτω αρχεία σ' αυτό.

**./gar/Math.jar  
./gar/Math-stub.jar  
./gar/server-deploy.wsdd  
./gar/schema/tutorial/MathService.wsdl**

Όπως παρατηρούμε υπάρχει ένα αρχείο που ονομάζεται server-deploy.wsdd. Αυτό είναι το Math.wsdd που δημιουργήθηκε προηγουμένως το οποίο πρέπει να μετονομαστεί σε server-deploy.wsdd για να ενταχθεί στο GAR.

- Δημιουργία του GAR

```
jar cvf Math.gar -C gar/ ./
```

- Ένταξη της υπηρεσίας στο Globus

```
ant deploy -Dgar.name=GAR_PATH
```

GAR\_PATH είναι ο κατάλογος που βρίσκεται το Math.gar

10. Για να χρησιμοποιήσουμε την υπηρεσία δημιουργούμε μια εφαρμογή πελάτη η οποία καλεί τη μέθοδο "add". Για να λειτουργήσει λαμβάνει τρία ορίσματα. Το Χειριστήριο της υπηρεσίας (Grid Service Handle) και τους δύο αριθμούς που θα προστιθένται. Για το σκοπό αυτό φτιάχνουμε ένα κατάλογο που ονομάζεται client μέσα στο gt3tutorial/core/first. Στο νέο κατάλογο δημιουργούμε ένα αρχείο που το ονομάζουμε MathClient.java. Ο κώδικας του αρχείου παρατίθεται στη συνέχεια.

```
package gt3tutorial.core.first.client;
import gt3tutorial.core.first.Math.MathServiceLocator;
import gt3tutorial.core.first.Math.MathPortType;
import java.net.URL;
public class MathClient
{public static void main(String[] args){
try{
int a = Integer.parseInt(args[1]);
int b = Integer.parseInt(args[2]);
URL GSH = new java.net.URL(args [0]);
MathServiceLocator mathService = new
MathServiceLocator();
MathPortType math = mathService.getMathService(GSH);
int sum = math.add(a,b);
System.out.println(a + " + " + b + " = " + sum);
}catch(Exception e)
{System.out.println("ERROR!"); e.printStackTrace(); } }}
```

Τμήμα Κώδικα 8 - MathClient.java

- Στη συνέχεια πρέπει να το μεταγλωπίσουμε:

**javac -sourcepath ./ gt3tutorial/core/first/client/\*.java**

- Πριν το εκτελέσουμε πρέπει να ξεκινήσουμε στον εξυπηρετητή το κιβώτιο υπηρεσιών πλέγματος (globus grid container)

**globus-start-container -p 8080**

- Εκτελούμε το πρόγραμμα στον πελάτη ζητώντας την πρόσθεση των αριθμών 48 97.

**java gt3tutorial.core.first.client.MathClient  
195.251.252.111:8080/ogsa/services/tutorial/core/first/MathService 48 97**

Το αποτέλεσμα που θα εμφανιστεί στην οθόνη του πελάτη έχει την εξής μορφή.

```
gt:/usr/local/math java
gt3tutorial.core.first.client.MathClient
re/first/MathService 48 97
- Logging is working
- Date: Tue Feb 10 16:12:45 EET 2004
- Version Apache-XML-Security-J 1.0.4
48 + 97 = 145
```

**Κεφάλαιο 6°**

**Συμπεράσματα-Περαιτέρω Έρευνα**

## **6. Συμπεράσματα & Περαιτέρω Έρευνα**

Η ολοκλήρωση της εργασίας μας κρίνεται απαραίτητο να συνοδεύεται και από μια σύντομη παρουσίαση τόσο των ευρημάτων της δουλείας μας όσο και κάποιων θεμάτων με τα οποία δεν κατέστη δυνατό να ασχοληθούμε.

Τα δύο περιβάλλοντα με τα οποία ασχοληθήκαμε αποτελούν τις πιο σύγχρονες, ίσως, προτάσεις ανάπτυξης ενός υπολογιστικού πλέγματος. Η εγκατάσταση τους στο Εργαστήριο Συστημάτων Υπολογιστών και Επικοινωνιών του Οικονομικού Πανεπιστημίου Αθηνών μας έδωσε τη δυνατότητα να εξάγουμε ορισμένα χρήσιμα συμπεράσματα σχετικά με τον τρόπο δόμησης και λειτουργίας τους.

Ο σημαντικά μικρότερος βαθμός δυσκολίας που παρουσίασε η διαδικασία εγκατάστασης του Sun One Grid Engine σε σύγκριση με το Globus Toolkit αποτελεί ένα από τα βασικά συμπεράσματα της δουλειάς μας. Η χρονική διάρκεια, σε ανθρωπούρες, που απαίτησε η εγκατάσταση του SGE κυμάνθηκε περίπου στο 1/3 του χρόνου που χρειάστηκε η εγκατάσταση του Globus Toolkit. Κρίσιμο ρόλο στην χρονική αυτή διαφορά διαδραμάτισε το γεγονός ότι για την ολοκλήρωση κάποιων σταδίων της διαδικασίας εγκατάστασης του Globus απαιτήθηκαν μεγάλα χρονικά διαστήματα κατά τα οποία γινόταν αντιγραφή αρχείων και χτίσιμο του περιβάλλοντος δίχως να υπάρχει δυνατότητα παρέμβασης από τη δική μας πλευρά.

Η δυνατότητα κλιμάκωσης ενός πλέγματος που είναι χτισμένο πάνω στην τεχνολογία του Globus είναι μεγαλύτερη σε σχέση με ένα πλέγμα που είναι χτισμένο με βάση το Sun One Grid Engine. Η διαφορά αυτή σχετίζεται άμεσα αφενός με το γεγονός ότι αποτελεί το προϊόν της συνεργασίας πολλών οργανισμών στα πλαίσια του Globus Alliance ενώ το Grid Engine είναι προϊόν της εταιρίας Sun αφετέρου με την φιλοσοφία που το χαρακτηρίζει. Έχει τέτοια δομή που μπορεί να ενσωματώσει οποιαδήποτε τεχνολογία

χρησιμοποιείται σήμερα στο χώρο της πληροφορικής. Άλλωστε αποτελεί το πρώτο εργαλείο που έχει ενσωματώσει στη δομή του τα πρότυπα της αρχιτεκτονικής Open Grid Service Architecture σκοπό δημιουργίας της οποίας αποτελεί ο συγκερασμός όλων των τεχνολογιών. Ο συγκερασμός αυτός αναδείχθηκε και από τον αριθμό των τεχνολογιών που χρησιμοποιήθηκαν, στα πλαίσια της εργασίας μας, για την υλοποίηση της απλής υπηρεσίας.

Η δυνατότητα του Globus να συνδυάζει διαφορετικές και ετερόκλητες τεχνολογίες μπορεί να αποτελέσει το εφαλτήριο για διερεύνηση του τρόπου και της χρησιμότητας συνδυασμού του με το Sun One Grid Engine στα πλαίσια λειτουργίας ενός υπολογιστικού πλέγματος.

Η ενσωμάτωση της αρχιτεκτονικής OGSA στο Globus Toolkit δίνει το ένασμα για μελέτη των δυνατοτήτων που υπάρχουν για ανάπτυξη προς αυτήν την κατεύθυνση του Sun One Grid Engine. Δηλαδή, να εξετασθεί αν είναι εφικτή η ενσωμάτωση και ποιος ο βαθμός δυσκολίας του εγχειρήματος.

Ένα άλλο σημαντικό πεδίο έρευνας αποτελεί η προσπάθεια ολοκλήρωσης της διατύπωσης των αρχών της OGSA. Όσο σαφέστερα γίνει αυτό τόσο πιο εύκολη θα γίνει η ολοκλήρωση διαφόρων τεχνολογιών στα πλαίσια των υπολογιστικών πλεγμάτων.

Πέρα από τα περιβάλλοντα τα οποία αποτέλεσαν τον πυρήνα της εργασίας μας, αξίζει να αναφερθούμε σε μια από τις βασικότερες δυνατότητες που παρέχει η λειτουργία ενός υπολογιστικού πλέγματος η οποία χρίζει έρευνας και ανάπτυξης από την επιστημονική κοινότητα. Αυτή σχετίζεται με την ικανότητά του να επεξεργάζεται μια εφαρμογή παράλληλα σε περισσότερους από έναν επεξεργαστές. Η αξιοποίηση τής δυνατότητας αυτής συνδέεται άμεσα με το βαθμό κλιμάκωσης που είναι δυνατό να εμφανίσει ο αλγόριθμος της εφαρμογής. Δηλαδή, εφόσον ο αλγόριθμος μπορεί να αποτελείται από π.χ. 10 ανεξάρτητα τμήματα η εκτέλεση της αντίστοιχης εφαρμογής σε 10 επεξεργαστές ενός υπολογιστικού πλέγματος θα πρέπει να οδηγεί σε μείωση

του χρόνου εκτέλεσης κατά 10 φορές σε σύγκριση με το χρόνο εκτέλεσης του σε ένα επεξεργαστή. Άρα, το ζήτημα της δυνατότητας κλιμάκωσης ενός αλγορίθμου και η αναζήτηση μεθόδων που θα υπολογίζουν το βαθμό που μπορεί να προσεγγίσει αποτελούν ένα πολύ ενδιαφέρον και κρίσιμο θέμα που απαιτεί εντατική έρευνα.



## **Βιβλιογραφία**

- [1].**Sun One Grid Engine Administration and User's Guide**, Sun Microsystems, Santa Clara USA, October 2002
- [2].**Sun One Grid Engine 5.3 Release Notes**, Sun Microsystems, Santa Clara USA, October 2002
- [3].**Sun ONE Grid Engine and Sun ONE Grid Engine**, Enterprise Edition Reference Manual , Sun Microsystems, Santa Clara USA, October 2002
- [4].**Sun's Grid Computing Solutions Outdistance the Competition**, An executive White Paper, Aberdeen Group, May 2002
- [5].**Sun Cluster Grid Architecture**, A technical white paper describing the foundation of Sun Grid Computing, Sun Microsystems, Santa Clara USA, May 2002
- [6].**Introduction to the Cluster Grid – Part 1**, James Coomer, Charu Chaubal, Sun BluePrints Online, August 2002
- [7].**The Globus Project: A Status Report**. I. Foster, C. Kesselman. Proc. IPPS/SPDP '98 Heterogeneous Computing Workshop, pp. 4-18, 1998.
- [8].**Globus: A Metacomputing Infrastructure Toolkit**. I. Foster, C. Kesselman. *Intl J. Supercomputer Applications*, 11(2):115-128, 1997.
- [9].**A Resource Management Architecture for Metacomputing Systems**. K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, S. Tuecke. Proc. IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing, pg. 62-82, 1998.

[10].**The Nexus Approach to Integrating Multithreading and Communication.** I. Foster, C. Kesselman, S. Tuecke, *J. Journal of Parallel and Distributed Computing*, 37:70--82, 1996.

[11].**A visual tour of Open Grid Service Architecture,** Jay Unger, Mat Haynos, August 2003, URL:  
<http://www-106.ibm.com/developerworks/grid/library/gr-visual>

[12].**The Physiology of the Grid.** Ian Foster, Carl Kesselman, Jeffrey M. Nick, Steven Tuecke, version:6/22/2002, URL:  
<http://www.globus.org/research/papers/ogsa/pdf>

[13].**The Anatomy of the Grid: Enabling Scalable Virtual Organizations.** Ian Foster, Carl Kesselman, Steven Tuecke, *International Journal Supercomputing Applications*, 15(3), 2001.

[14].**Fundamentals of Grid Computing.** Viktors Berstsis, RedBooks Paper, IBM, URL:<http://ibm.com/redbooks>

[15].**The Data Grid:Towards an Architecture for the Distributed Management and Analysis of Large Scientific Data Sets.** Chervenak, A., Foster, I., Kesselman, C., Salisbury, C. and Tuecke, S.. *J. Network and Computer Applications*, 2001.

[16].**Co-allocation Services for Computational Grids.** Czajkowski, K., Foster, I. and Kesselman, C., In *Proc. 8th IEEE Symposium on High Performance Distributed Computing*, 1999, IEEE Press.

[17].**Application Experiences with the Globus Toolkit.** Brunett, S., Czajkowski, K., Fitzgerald, S., Foster, I., Johnson, A., Kesselman, C., Leigh, J. and Tuecke, S., In *Proc. 7th IEEE Symp. on High Performance Distributed Computing*, 1998, IEEE Press, 81-89.

[18].**Grid Services for Distributed System integration.** Ian Foster, Carl Kesselman, Jeffrey M. Nick, Steven Tuecke, June 2002, IEEE Computer, 37-46

[19].**Open Grid Service Specification Version 1.0,** Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C, T. Maquire, T. Sandholm, D. Snelling, Vanderbilt, P.. Global Grid Forum, 2003,

[20].Java Sun Web Services Tutorial, URL:

<http://java.sun.com/webservices/downloads/webservicestutorial.html>

[21].Linux Online Tutorial, URL:

[http://www.tldp.org/HOWTO/NFS\\_HOWTO/intro.html](http://www.tldp.org/HOWTO/NFS_HOWTO/intro.html)

[22].Java Installation Tutorial,

URL: [http://java.sun.com/j2eee/sdk\\_1.3/install.html](http://java.sun.com/j2eee/sdk_1.3/install.html)

[23].Globus Toolkit 3, URL:<http://www.globus.org>.

[24].Sun One Grid Engine, URL:<http://www.sun.com/gridware>

[25]. The Globus Toolkit 3 Programmer's Tutorial, Borja Sotomayor, URL:  
<http://www.casa-sotomayor.net/gt3-tutorial-working/>.

[26].Step-By-Step Example for The Globus Toolkit 3.0 Alpha-2 Release. URL:  
[http://usuarios.lycos.es/javiercanolinas/step\\_by\\_step\\_guide.html](http://usuarios.lycos.es/javiercanolinas/step_by_step_guide.html)

[27]. Ant, URL:<http://jakarta.apache.org/ant>



Δωρεά



80025 75540

