

ABSTRACT

The increased use of the grid (computing) during the last years led to the creation of numerous tools which facilitate users to complete faster and easier than before several functions, such as sending applications to the grid etc. The question raised is: which resource a user should choose to send an application in order to have more possibilities for the application to be executed faster. This is the reason why the idea of ranking grid's performance was developed. User executes benchmarks to grid's resources in order to get an idea of each resource's performance. However, a user rarely knows their application's requirements so as to choose the resource which gets the best possible results. The case for a user to send an application (or part of it) to all the resources and expect the results in order to compare them is rejected due to the fact that such a choice is time consuming. The present thesis suggests the functioning of a system capable of resembling an application's performance in case this has already been executed to any of the grid's resources. The latter is achieved by creating a rank function. This is created by finding the correlation between the benchmarks and the applications which have already been executed to the grid's resources. In this case there is no need for an application to be executed to any offered resource but only to a small percentage of them.

Via a wizard, users can create the rank function by themselves or, otherwise, it can be automatically created via the system. Then, the rank function is executed to the resources the user has chosen for their application to be evaluated. Function's form, as well as other information, is saved as an xml file which is used for depicting the results in a graphical form. Users can see the information included in the xml file mentioned, through a graph. At the same time users can also edit the information via an editor. The system is designed in respect of dividing the logical unit from the graphical one in order to be easily expandable and comprehensible. At the same time this way of designing allows the system to change the mathematical model used for the rank function's architecture without changing the core algorithm. The system was operated as an expansion of g Eclipse research program. The initial structure and function of the specific framework were reserved. g Eclipse is a tool facilitating the grid's use and meets the needs of different kinds of users. The operation is part of the g Eclipse benchmark framework which includes various features from the GridBench. GridBench is a tool developed by University of Cyprus and is a pioneering tool for ranking grid sites.

ΠΕΡΙΛΗΨΗ

Η μεγάλη αύξηση της χρήσης του υπολογιστικού πλέγματος (grid) τα τελευταία χρόνια οδήγησε στη δημιουργία πληθώρα εργαλείων που διευκολύνουν το χρήστη να πραγματοποιεί πιο γρήγορα και εύκολα διάφορες λειτουργίες του πλέγματος όπως η αποστολή εργασιών/εφαρμογών, η παραλαβή των αποτελεσμάτων κ.ά. Έτσι το ερώτημα που τίθεται είναι σε ποιό πόρο αξίζει να στείλει ο χρήστης την εφαρμογή ώστε να έχει περισσότερες πιθανότητες να εκτελεστεί πιο γρήγορα. Για το λόγο αυτό, αναπτύχθηκε η ιδέα της μέτρησης επίδοσης στο υπολογιστικό πλέγμα. Ο χρήστης εκτελεί κάποια benchmarks στους πόρους του υπολογιστικού πλέγματος και αποκτά μια εικόνα για τις δυνατότητες κάθε πόρου. Παρόλα αυτά σπάνια γνωρίζει τί είδους απαιτήσεις έχει η εφαρμογή του, ώστε να μπορεί να επιλέξει τους πόρους που έχουν τα καλύτερα αποτελέσματα. Η περίπτωση να στείλει μια εφαρμογή (ή και κομμάτι αυτής) σε όλους τους πόρους και να περιμένει το αποτέλεσμα ώστε να κάνει στη συνέχεια σύγκριση μεταξύ αυτών, απορρίπτεται λόγω του μεγάλου χρονικού κόστους. Στην παρούσα διπλωματική εργασία προτείνεται η υλοποίηση ενός συστήματος το οποίο είναι ικανό να προσομοιώσει την επίδοση μιας εφαρμογής εάν είχε όντως εκτελεστεί σε κάποιο πόρο του υπολογιστικού πλέγματος. Αυτό επιτυγχάνεται βρίσκοντας τη συσχέτιση μεταξύ των benchmarks και των εφαρμογών που έχουν ήδη εκτελεστεί στους πόρους του υπολογιστικού πλέγματος δημιουργώντας μια συνάρτηση βαθμολόγησης. Σε αυτή τη περίπτωση δεν χρειάζεται να εκτελεστεί η εφαρμογή σε κάθε πόρο αλλά σε ένα μικρό ποσοστό αυτών.

Ο χρήστης με τη βοήθεια ενός οδηγού (wizard) μπορεί είτε να δημιουργήσει μόνος του τη συνάρτηση βαθμολόγησης είτε να δημιουργηθεί αυτόματα αυτή από το σύστημα και η οποία στη συνέχεια εφαρμόζεται στους πόρους που έχει επιλέξει ο χρήστης προκειμένου να αξιολογήσει μια εφαρμογή. Η μορφή της συνάρτησης καθώς και κάποιες επιπλέον πληροφορίες αποθηκεύονται σε ένα xml αρχείο το οποίο χρησιμοποιείται στη συνέχεια για να απεικονιστούν τα αποτελέσματα σε μορφή γραφικής παράστασης. Ο χρήστης μπορεί μέσω ενός γραφικού περιβάλλοντος να δει τις πληροφορίες που φέρει το xml αρχείο καθώς και να το επεξεργαστεί μέσω ενός συντάκτη (editor). Η σχεδίαση του συστήματος πραγματοποιήθηκε με γνώμονα το διαχωρισμό της λογικής μονάδας από αυτό της γραφικής ώστε το σύστημα να είναι εύκολα επεκτάσιμο και κατανοητό, καθώς και στη δυνατότητα να μπορεί να τροποποιηθεί ή και να αλλάξει τελείως το μαθηματικό μοντέλο που χρησιμοποιείται για τη δημιουργία της συνάρτησης βαθμολόγησης χωρίς να απαιτούνται αλλαγές στο βασικό αλγόριθμο. Το σύστημα υλοποιήθηκε ως μια επέκταση του λογισμικού συστήματος g Eclipse διατηρώντας την υπάρχουσα δομή και λειτουργία του συγκεκριμένου πλαισίου εργασίας. Το g Eclipse αποτελεί ένα εργαλείο που διευκολύνει τη χρήση του υπολογιστικού πλέγματος και καλύπτει τις ανάγκες διαφορετικών κατηγοριών χρηστών. Η υλοποίηση εντάσσεται στο πλαίσιο εργασίας της μέτρησης επίδοσης του g Eclipse το οποίο έχει πολλά χαρακτηριστικά από το GridBench, ένα εργαλείο το οποίο αναπτύχθηκε στο πανεπιστήμιο Κύπρου και είναι πρωτοπόρο στο θέμα της μέτρησης επίδοσης στο υπολογιστικό πλέγμα.

Σύστημα Πρόβλεψης Επίδοσης Εφαρμογών Υπολογιστικού

Πλέγματος στην πλατφόρμα g Eclipse

Κυριάκος Κατσαρής

Η Διατριβή αυτή

Υποβλήθηκε προς Μερική Εκπλήρωση των

Απαιτήσεων για την Απόκτηση

Τίτλου Σπουδών Master

σε Προηγμένες Τεχνολογίες Πληροφορικής

στο

Πανεπιστήμιο Κύπρου

Συστήνεται προς Αποδοχή

από το Τμήμα Πληροφορικής

Ιούνης, 2009

ΣΕΛΙΔΑ ΕΓΚΡΙΣΗΣ

ΕΥΧΑΡΙΣΤΙΕΣ

Ολοκληρώνοντας τη διπλωματική εργασία θα ήθελα να ευχαριστήσω το κ.Δικαίακο για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον ζήτημα όπως είναι η μέτρηση επίδοσης στο υπολογιστικό πλέγμα καθώς και για τις ιδέες και τα σχόλια του σχετικά με την ανάπτυξη του συστήματος.

Επίσης νιώθω την ανάγκη να ευχαριστήσω την οικογένεια μου για τη διαρκή τους συμπαράσταση όλα αυτά τα χρόνια της φοίτησης μου τόσο προπτυχιακό όσο και στο μεταπτυχιακό πρόγραμμα καθώς και τους φίλους μου Βασίλη, Ορέστη, Λεωνίδα και Νίκο για την ψυχολογική τους υποστήριξη όποτε και αν τη χρειάστηκα

Επίσης θα ήταν μεγάλη παράλειψη να μην αναφέρω την πολύτιμη βοήθεια που μου προσέφεραν όλα τα παιδιά του εργαστηρίου HPCL όποτε και αν τη χρειάστηκα.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Κεφάλαιο 1: Εισαγωγή.....	1
1.1 Ορισμός του προβλήματος	1
1.2 Γενική περιγραφή διπλωματικής εργασίας.....	2
1.2 Δομή της διπλωματικής εργασίας.....	3
Κεφάλαιο 2 : Μέτρηση επίδοσης στο υπολογιστικό πλέγμα	4
2.1 Μέτρηση επίδοσης (benchmarking) στο υπολογιστικό πλέγμα (grid).....	4
2.2 Όραμα και δυσκολίες στο υπολογιστικό πλέγμα (grid).....	4
2.3 Μέτρηση επίδοσης στο υπολογιστικό πλέγμα (Grid Benchmarking).....	6
2.4 Κατηγορίες στη μέτρηση επίδοσης στο υπολογιστικό πλέγμα.....	6
2.5 Κατηγορίες των benchmarks	7
2.6 Dwarfs	8
2.7 GridBench.....	9
Κεφάλαιο 3 : Το g Eclipse, ένα εργαλείο για χρήση του υπολογιστικού πλέγματος.....	10
3.1 Σχετικά με το περιβάλλον Eclipse	11
3.2 Το g Eclipse, ένα εργαλείο για χρήση του υπολογιστικού πλέγματος	13
3.3 Το πλαίσιο εργασίας του g Eclipse	14
3.4 Μετρήσεις Επιδόσεων στο υπολογιστικό πλέγμα με την χρήση του g Eclipse.....	15
3.5 Ερμηνεία των GBDL αρχείων	16
3.5.1 Δημιουργία των GBDL αρχείων εντός του g Eclipse	17
3.6 Πλαίσιο εργασίας της μέτρησης επίδοσης στο εργαλείο g Eclipse.....	19
Κεφάλαιο 4: Ανάλυση και σχεδίαση συστήματος SiteRank	21
4.1 Σκοπός και στόχοι του συστήματος	21
4.2 Γενική ροή συστήματος.....	22
4.3 Δημιουργία συνάρτησης βαθμολόγησης από το χρήστη.....	23
4.3.1 Ροή Εργασίας «δημιουργία συνάρτησης βαθμολόγησης από το χρήστη»	23
4.4 Δημιουργία συνάρτησης βαθμολόγησης από το σύστημα	25

4.4.1 Ροή Εργασίας «δημιουργία συνάρτησης βαθμολόγησης από το σύστημα».....	25
4.4.2 Βήματα του αλγορίθμου για τη δημιουργία της συνάρτησης βαθμολόγησης από το σύστημα.....	26
4.4.3 Αλγόριθμος δημιουργίας της συνάρτησης βαθμολόγησης	28
4.5 Αρχιτεκτονική συστήματος εντός του πλαισίου μέτρησης απόδοσης στο g Eclipse (benchmarking framework).....	30
4.5.1 SiteRank Framework (Filtering – Aggregation – Rank model)	31
4.5.2 SiteRank View	32
4.5.2.1 Βήματα γραφικής απεικόνισης.....	32
4.5.2.2 Οδηγός (wizard) δημιουργίας συνάρτησης βαθμολόγησης από το χρήστη και από το σύστημα.....	33
4.5.2.3 Περίπτωση δημιουργίας συνάρτησης αξιολόγησης από τον χρήστη	35
4.5.2.4 Περίπτωση δημιουργίας συνάρτησης αξιολόγησης από σύστημα.....	37
4.6 Δημιουργία XML αρχείου μετά την ολοκλήρωση του οδηγού	40
4.6.1 Πληροφορίες σχετικά με την xml γλώσσα	40
4.7 Σχεδίαση και ανάλυση της όψεως (view).....	42
4.8 Υλοποίηση συστήματος	44
4.8.1 Auto Model Creation	45
4.8.2 XML file Creation.....	46
4.8.3 XML files.....	47
4.8.4 Wizard	47
4.8.5 Process XML file	48
4.8.6 Rank Analysis	48
4.8.7 Database.....	49
Κεφάλαιο 5 : Πειράματα	51
5.1 Χρήση του dwarf SciMark	51
5.2 Χρήση του dwarf nbody	58
Κεφάλαιο 6 : Επίλογος.....	63

6.1 Σύνοψη	63
6.2 Συμπεράσματα	63
6.3 Μελλοντικές Επεκτάσεις	64
Αναφορές.....	66
Παράρτημα	69

ΚΑΤΑΛΟΓΟΣ ΜΕ ΠΙΝΑΚΕΣ

Πίνακας 1	Micro Benchmarks	51
Πίνακας 2	Πόροι του V.O SEE.....	52
Πίνακας 3	Πόροι που έχουν εκτελέσει το dwarf SciMark.....	52
Πίνακας 4	Πόροι που έχουν εκτελέσει όλα τα micro benchmarks.....	53
Πίνακας 5	Πόροι για δειγματοληψία και αξιολόγηση αντίστοιχα.....	55
Πίνακας 6	Πραγματικές και υπολογισθείσες τιμές	57
Πίνακας 7	Πόροι που έχουν εκτελέσει το dwarf nbody	58
Πίνακας 8	Πόροι που έχουν εκτελέσει όλα τα micro benchmarks	59
Πίνακας 9	Πόροι για δειγματοληψία και αξιολόγηση αντίστοιχα.....	59
Πίνακας 10-	Πραγματικές και υπολογισθείσες τιμές	65

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2.1 – Δομή του υπολογιστικού πλέγματος.....	5
Εικόνα 2.2 - Κατηγορίες των benchmarks.....	8
Εικόνα 2.3 - Αρχιτεκτονική GridBench.....	9
Εικόνα 2.4 - Αρχιτεκτονική GridBench.....	10
Εικόνα 3.1 - Δομή βυσμάτων.....	12
Εικόνα 3.2 - Δομή Eclipse.....	12
Εικόνα 3.3 - Περιπτώσεις χρηστών g Eclipse.....	14
Εικόνα 3.4 - Δομή του g Eclipse.....	15
Εικόνα 3.5 - Δομή γλώσσας GBDL.....	16
Εικόνα 3.6 - Σελίδες GBDL οδηγού.....	18
Εικόνα 3.7 - Γραφικό περιβάλλον g Eclipse.....	18
Εικόνα 3.8 - Γραφική απεικόνιση benchmarks.....	19
Εικόνα 3.9 - Αρχιτεκτονική του g Eclipse Benchmark.....	20
Εικόνα 4.1 - Βήματα υλοποίησης συστήματος.....	22
Εικόνα 4.2 - Εφαρμογή συνάρτησης βαθμολόγησης σε πόρους.....	25
Εικόνα 4.3 - Διαδικασία δημιουργίας συνάρτησης αξιολόγησης	27
Εικόνα 4.4- Παράδειγμα πίνακα συσχέτισης.....	29
Εικόνα 4.5 - Αρχιτεκτονική συστήματος.....	31
Εικόνα 4.6 - Δυνατές επιλογές του χρήστη	32
Εικόνα 4.7 - Βήματα του οδηγού	33
Εικόνα 4.8 -Δομή του οδηγού	34
Εικόνα 4.9 - Σελίδα 1 του οδηγού	34
Εικόνα 4.10-Σελίδα δημιουργίας συνάρτησης από τον χρήστη	35
Εικόνα 4.11- Σελίδα δημιουργίας συνάρτησης από τον χρήστη	36
Εικόνα 4.12- Σελίδα επιλογής πόρων για αξιολόγηση	36
Εικόνα 4.13- Σελίδα επιλογής εφαρμογής	37
Εικόνα 4.14- Σελίδα επιλογής πόρων δειγματοληψίας	37
Εικόνα 4.15- Σελίδα επιλογής πόρων για αξιολόγηση	38
Εικόνα 4.16- Σελίδα εμφάνισης αποτελεσμάτων	39
Εικόνα 4.17- Εμφάνιση λάθους	39
Εικόνα 4.18- Στοιχεία που αποτελούν το xml αρχείο.....	40
Εικόνα 4.19- Σχέση μεταξύ στοιχείων της xml γλώσσας	42
Εικόνα 4.20- Διαθέσιμες πληροφορίες στον χρήστη.....	43
Εικόνα 4.21- Γραφική απεικόνιση αποτελεσμάτων	44
Εικόνα 4.22- Συστατικά που αποτελούν το σύστημα SiteRank	44
Εικόνα 4.23- Σχέση κλάσεων.....	45
Εικόνα 4.24- Βήματα υλοποίησης συνάρτησης βαθμολόγησης	46
Εικόνα 4.25- Σχέση κλάσεων	47
Εικόνα 4.26- Βήματα υλοποίησης οδηγού	48
Εικόνα 4.27- Σχέση κλάσεων	49
Εικόνα 4.28- Σχήμα βάσης δεδομένων	50

Εικόνα 5.1- Επιλογή πόρων δειγματοληψίας	53
Εικόνα 5.2- Επιλογή πόρων για εφαρμογή της συνάρτησης αξιολόγησης	54
Εικόνα 5.3- Παρουσίαση νεοδημιουργηθείσας συνάρτησης βαθμολόγησης.....	54
Εικόνα 5.4- Γραφική απεικόνιση αποτελεσμάτων	56
Εικόνα 5.5- Πραγματικές τιμές της εφαρμογής στους πόρους.....	56
Εικόνα 5.6- Γραφική απεικόνιση πραγματικών τιμών των πόρων.....	57
Εικόνα 5.7- Γραφική απεικόνιση υπολογισθέντων τιμών των πόρων.....	58
Εικόνα 5.8- Γραφική απεικόνιση αποτελεσμάτων.....	60
Εικόνα 5.9- Πραγματικές τιμές της εφαρμογής στους πόρους.....	60
Εικόνα 5.10- Γραφική απεικόνιση πραγματικών τιμών των πόρων.....	61
Εικόνα 5.11- Γραφική απεικόνιση υπολογισθέντων τιμών των πόρων.....	62

Κεφάλαιο 1

Εισαγωγή

1.1 Ορισμός του προβλήματος

Η μεγάλη ανάπτυξη του διαδικτύου, η τεχνολογική εξέλιξη των προσωπικών υπολογιστών (PCs) και η ανάπτυξη του κατάλληλου ενδιάμεσου λογισμικού (middleware) και εφαρμογών έχει δημιουργήσει μια νέα δυναμική στην κλασική έννοια του όρου "υπολογιστικό περιβάλλον". Ο συνδυασμός των παραπάνω δίνει τη δυνατότητα του κατανεμημένου γεωγραφικά διαμοιρασμού πόρων όπως η υπολογιστική ισχύς, ο αποθηκευτικός χώρος, το ψηφιακό περιεχόμενο και άλλα επιστημονικά όργανα (π.χ. αισθητήρες, τηλεσκόπια). Ουσιαστικά, ένας ερευνητής, που βρίσκεται συνεχώς συνδεδεμένος σε δίκτυο υψηλών ταχυτήτων, με τη χρήση του κατάλληλου λογισμικού μπορεί να μοιράζεται την υπολογιστική ισχύ των υπολογιστών του, τον αποθηκευτικό του χώρο και τους άλλους πόρους του εργαστηρίου του με χιλιάδες άλλους ερευνητές στον κόσμο. Ο διαμοιρασμός αυτός μπορεί να γίνει με ομοίωμα p2p, ασφαλή και κατανεμημένο τρόπο σε παγκόσμιο επίπεδο.

Οι νέες αυτές μέθοδοι, γνωστές ως τεχνολογίες "πλέγματος υπολογιστικών συστημάτων - Grid" [4], αποτελούν σήμερα την τεχνολογία αιχμής σε παγκόσμιο επίπεδο για την ικανοποίηση, μεταξύ άλλων, υψηλών απαιτήσεων σε υπολογιστική ισχύ και χώρους αποθήκευσης δεδομένων. Το μοντέλο υλοποίησης υπέρ-υπολογιστικών υποδομών που επικρατεί σήμερα είναι ο συνδυασμός προσωπικών υπολογιστών σε συστοιχίες (cluster computing) διασυνδεδεμένες με δίκτυα υψηλών ταχυτήτων, σε αντίθεση με το παλαιότερο μοντέλο των μεγάλων σε μέγεθος και ασύμφορων υπερ-υπολογιστών λόγω του υψηλού κόστους αγοράς, εγκατάστασης, συντήρησης, λειτουργίας και αναβάθμισης.

Με τη χρήση κατάλληλων εργαλείων ο χρήστης μπορεί πλέον να στείλει μια εργασία/εφαρμογή στο υπολογιστικό πλέγμα η οποία και θα εκτελεστεί σε κάποιο πόρο επιστρέφοντας το αποτέλεσμα όταν αυτό είναι διαθέσιμο. Γνωρίζοντας την ετερογένεια του υπολογιστικού πλέγματος κυρίως ως προς το υλικό, ένα ερώτημα που εύκολα προκύπτει είναι σε ποιόν πόρο θα εκτελεστεί γρηγορότερα η εργασία/εφαρμογή του χρήστη. Μια λύση που προτείνεται είναι αυτή της χρήσης κάποιων υπηρεσιών (services) όπως αυτή της παρακολούθησης (monitor service), που παρέχεται από την ίδια την υποδομή του υπολογιστικού πλέγματος. Δυστυχώς όμως η λύση αυτή δεν είναι επαρκής γιατί οι πληροφορίες που δίνει η υπηρεσία αυτή δεν είναι αρκετές και πολλές φορές δεν είναι καν αξιόπιστες.

Για αυτό το λόγο αναπτύχθηκε η έννοια της μέτρησης επίδοσης στο υπολογιστικό πλέγμα (grid benchmarking) [3]. Benchmarks που χρησιμοποιήθηκαν στο παρελθόν μπορούν πλέον να επεκταθούν και να προσαρμοστούν προκειμένου να αξιοποιηθούν στο υπολογιστικό πλέγμα μετρώντας κάθε πτυχή αυτού.

Εκτελώντας επομένως τα benchmarks ο χρήστης μπορεί να συγκρίνει τα αποτελέσματα και να αποφασίσει ποιος πόρος καλύπτει περισσότερο τις ανάγκες του ώστε να στείλει σε αυτό την εφαρμογή του. Το ιδανικότερο, όπως γίνεται εύκολα αντιληπτό, θα ήταν ο χρήστης να εκτελούσε ολόκληρη την εφαρμογή του (ή έστω ένα κομμάτι αυτής – demo) και να κάνει στη συνέχεια τις απαραίτητες συγκρίσεις ώστε να γνωρίζει στο μέλλον ποιο πόρο να επιλέξει. Το κόστος όμως να εκτελεστούν ολόκληρες εφαρμογές, ακόμα και κομμάτια αυτών, σε όλους τους πόρους είναι τεράστιο σε σχέση με το να εκτελεστούν benchmarks χαμηλού επιπέδου.

1.2 Γενική περιγραφή διπλωματικής εργασίας

Στην παρούσα διπλωματική εργασία προτείνεται η υλοποίηση ενός συστήματος που χρησιμοποιεί benchmarks χαμηλού επιπέδου, τα οποία και έχουν εκτελεστεί σε κάθε πόρο του υπολογιστικού πλέγματος, προκειμένου να δώσουν στο σύστημα τη δυνατότητα πρόβλεψης της τιμής του αποτελέσματος μιας εφαρμογής εάν αυτή είχε πραγματικά εκτελεστεί.

Συγκριμένα με τη βοήθεια στατιστικών μεθόδων δημιουργείται δυναμικά μια συνάρτηση βαθμολόγησης η οποία εκτελείται στη συνέχεια σε κάθε επιλεγθέντα πόρο δίνοντας κατά προσέγγιση τη τιμή του αποτελέσματος της εφαρμογής στον πόρο αυτό [12]. Αναλυτικότερα η συνάρτηση αυτή δέχεται ως ορίσματα μετρικές οι οποίες προέρχονται από χαμηλού επιπέδου benchmarks και συνδυάζονται κατάλληλα δημιουργώντας εν τέλει ένα γραμμικό μοντέλο. Το γραμμικό αυτό μοντέλο είναι ουσιαστικά το αποτέλεσμα της συσχέτισης μεταξύ της επίδοσης μιας συγκεκριμένης εφαρμογής με τις μετρικές χαμηλού επιπέδου. Εν τέλει δίνεται η δυνατότητα στο χρήστη του υπολογιστικού πλέγματος να γνωρίζει την πιθανή επίδοση της εφαρμογής του σε κάποιον πόρο έχοντας απλά εκτελέσει τα χαμηλού επιπέδου benchmarks σε αυτόν.

Το σύστημα παρέχει ένα γραφικό τρόπο στο χρήστη για να δημιουργηθεί αυτή η συνάρτηση βαθμολόγησης και τα αποτελέσματα που προκύπτουν παρουσιάζονται με τη μορφή γραφικής παράστασης ώστε να είναι εύκολη και άμεση η κατανόηση τους. Το σύστημα αναπτύχθηκε ως μια επέκταση του λογισμικού συστήματος g Eclipse διατηρώντας έτσι την υπάρχουσα δομή και λειτουργία του συγκεκριμένου πλαισίου εργασίας [9]. Το g Eclipse αποτελεί ένα εργαλείο που διευκολύνει τη χρήση του υπολογιστικού πλέγματος για διαφορετικές κατηγορίες χρηστών και είναι με τη σειρά του μια επέκταση του προγραμματιστικού εργαλείου Eclipse. Πρέπει να αναφερθεί ότι το εργαλείο g Eclipse ενσωματώνει πολλά χαρακτηριστικά καθώς και το σύστημα που αναπτύχθηκε στην διπλωματική εργασία, από το GridBench [7], ένα εργαλείο το οποίο αναπτύχθηκε στο πανεπιστήμιο Κύπρου και είναι πρωτοπόρο στο θέμα της μέτρησης επίδοσης στο υπολογιστικό πλέγμα.

Όσον αναφορά την υλοποίηση πρέπει να τονιστεί ο διαχωρισμός της λογικής μονάδας από αυτές των γραφικών και των δεδομένων εξασφαλίζοντας με αυτό τον τρόπο την εύκολη επέκταση του συστήματος στο μέλλον καθώς και στην καλύτερη κατανόηση του. Για να επιτευχθεί αυτό δημιουργήθηκε μια απλή xml γλώσσα η οποία περιγράφει τη συνάρτηση βαθμολόγησης, καθώς και κάποιες άλλες πληροφορίες επιτυγχάνοντας την επικοινωνία μεταξύ του μοντέλου απεικόνισης και αυτού της λογικής.

Τέλος η σχεδίαση του συστήματος έγινε με τέτοιο τρόπο ώστε να μπορεί να τροποποιηθεί ή να αλλάξει τελείως το μαθηματικό μοντέλο που χρησιμοποιείται για τη δημιουργία της συνάρτησης βαθμολόγησης ώστε να προστίθενται εύκολα και άμεσα νέοι μέθοδοι και διαδικασίες χωρίς να απαιτούνται αλλαγές στο βασικό αλγόριθμο.

1.2 Δομή της διπλωματικής εργασίας

Η δομή των κεφαλαίων που ακολουθήθηκε στη παρούσα διπλωματική εργασία είναι η εξής : Στο δεύτερο κεφάλαιο γίνεται αναφορά γενικά στη μέτρηση επίδοσης στο υπολογιστικό πλέγμα και επικεντρώνεται στη θεωρία της μέτρησης επίδοσης (είδη, δομές κτλ) καθώς και στη αρχιτεκτονική του υπολογιστικού πλέγματος. Το τρίτο κεφάλαιο αναφέρεται στο εργαλείο g Eclipse και στο πλαίσιο εργασίας του, καθώς και γενικά στο προγραμματιστικό περιβάλλον Eclipse. Το τέταρτο κεφάλαιο αναφέρεται διεξοδικά στον σχεδιασμό, την ανάλυση και τέλος την υλοποίηση του συστήματος. Το πέμπτο κεφάλαιο παρέχει το θεωρητικό υπόβαθρο που είναι απαραίτητο για κάποιες έννοιες στατιστικής που χρειάζονται σε κάποιους από τους αλγορίθμους. Στο έκτο κεφάλαιο παρέχονται κάποια παραδείγματα ως περιπτώσεις χρήσης όπου καταδεικνύεται η ικανότητα του συστήματος να επιστρέφει ορθά αποτελέσματα. Τέλος έχουμε τις απαιτούμενες αναφορές και το παράρτημα.

Κεφάλαιο 2

Μέτρηση επίδοσης στο υπολογιστικό πλέγμα

Στο κεφάλαιο αυτό θα περιγραφθεί λεπτομερώς η έννοια της μέτρησης επίδοσης στο υπολογιστικό πλέγμα, οι διαφορές από την κλασική έννοια της μέτρησης επίδοσης καθώς και οι κατηγορίες αυτών. Τέλος θα παρουσιαστεί το εργαλείο Gridbench το οποίο χρησιμοποιείται ως βάση για το σύστημα που θα περιγραφεί σε άλλα κεφάλαια.

2.1 Μέτρηση επίδοσης (benchmarking) στο υπολογιστικό πλέγμα (grid)

Τα benchmarks είναι ειδικά σχεδιασμένα προγράμματα που χρησιμοποιούνται για τον έλεγχο επίδοσης και συμπεριφοράς των υπολογιστικών συστημάτων κάτω από συγκεκριμένες και ελεγχόμενες συνθήκες [1]. Η μέτρηση επίδοσης θεωρείται η διαδικασία της εκτέλεσης κάποιων Benchmarks σε ένα συγκεκριμένο σύστημα και περιλαμβάνει τα βήματα της εκτέλεσης, μέτρησης και αναφοράς των αποτελεσμάτων. Τα benchmarks χρησιμοποιούνται, τόσο από τους κατασκευαστές συστημάτων προκειμένου να σχεδιάζουν συστήματα τα οποία θα αποκρίνονται καλύτερα σε αυτά [2], όσο και από τους χρήστες ως ένα μέσο αξιολόγησης των συστημάτων τους. Προφανώς κανένα benchmark δεν γίνεται να δώσει ακριβείς προβλέψεις σε σχέση για παράδειγμα με την κανονική εκτέλεση ενός προγράμματος. Παρόλα αυτά οι πληροφορίες που μπορούν να αντληθούν είναι πολύ σημαντικές και χρήζουν μελέτης. Επιπλέον υπάρχουν κάποιες οδηγίες που πρέπει να ακολουθούνται όταν δημιουργούνται και εκτελούνται κάποια benchmarks και παρουσιάζονται παρακάτω [3]

- Τα benchmarks πρέπει να είναι «δίκαια». Δεν πρέπει να είναι μεροληπτικά ως προς μια συγκεκριμένη υλοποίηση ή ένα σύστημα.
- Τα benchmarks πρέπει να είναι εύρηστα. Ένα εύρηστο και διαθέσιμο benchmark γίνεται πιο εύκολα αποδεκτό από τους χρήστες.
- Οι προδιαγραφές (specifications) των benchmarks πρέπει να είναι γνωστές και εύκολα κατανοητές από τους χρήστες. Είναι σημαντικό να γίνεται εύκολα αντιληπτό τί είναι αυτό που μετράει το εκάστοτε benchmark δηλαδή ποιες είναι οι μετρικές (metrics) που φέρει.
- Το κόστος της διαχείρισης ενός benchmark, το οποίο περιλαμβάνει την εκτέλεση, την αποθήκευση αποτελεσμάτων και την εμφάνιση αυτών, να είναι όσο το δυνατό μικρό.
- Τα benchmarks πρέπει να είναι «ρεαλιστικά». Τα benchmarks πρέπει να προσομοιώνουν όσο είναι δυνατό μια πραγματική κατάσταση που θα βρεθεί το σύστημα που μελετάται.

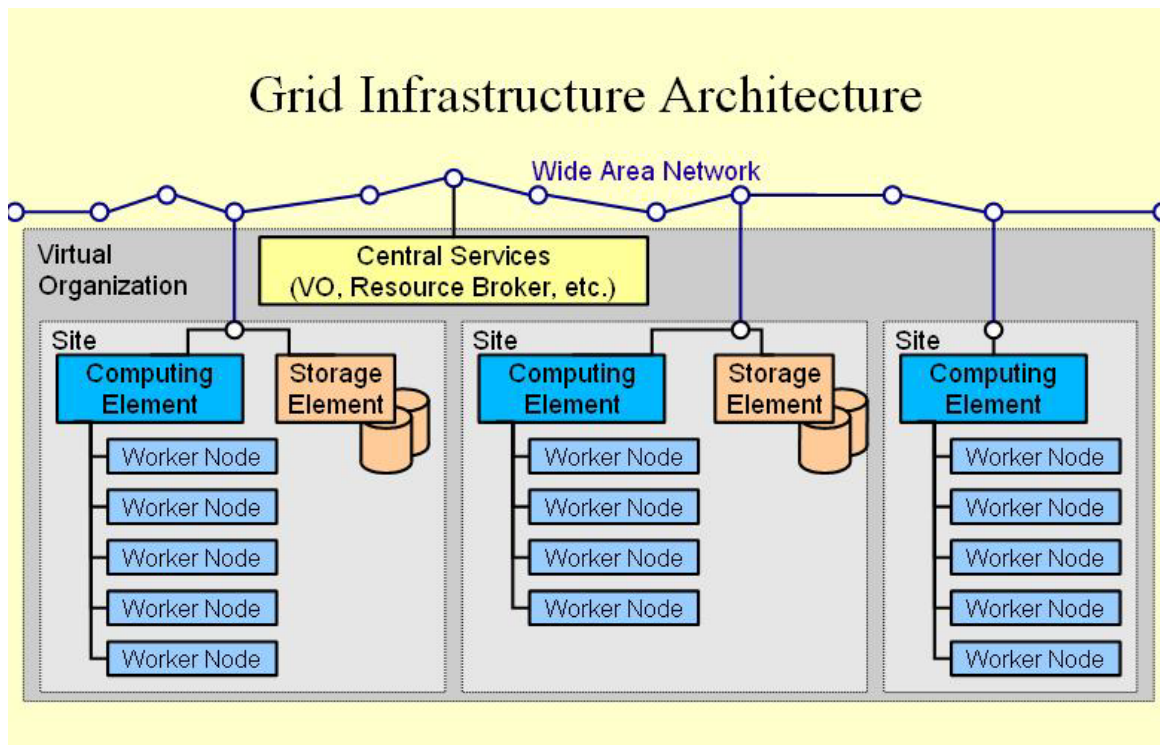
2.2 Όραμα και δυσκολίες στο υπολογιστικό πλέγμα (grid)

Ο όρος GRID (σε ελεύθερη απόδοση "Υπολογιστικό Πλέγμα") [4] περιλαμβάνει το σύνολο της υπάρχουσας υποδομής (υλικό και λογισμικό) καθώς και των υπηρεσιών για τη δημιουργία ενός ενιαίου υπολογιστικά περιβάλλοντος, που αν και γεωγραφικά

κατανεμημένο, εμφανίζεται με τρόπο διαφανή σε όλους τους χρήστες, σαν ένα ενιαίο σύνολο υπολογιστικών πόρων, μια συμπαγής υπολογιστική πλατφόρμα.

Το GRID προσφέρει ένα αξιόπιστο και ομοιόμορφο τρόπο πρόσβασης σε όλους τους διαθέσιμους υπολογιστικούς πόρους για όλη την κοινότητα των χρηστών του. Διασυνδέει ετερογενή υπολογιστικά περιβάλλοντα, με όμοια ή διαφορετική φιλοσοφία και υπηρεσίες, δημιουργώντας επιπλέον, νέα σύνολα υπηρεσιών με αυξημένες υπολογιστικές δυνατότητες και νέους τρόπους αξιοποίησης των ποικίλων πόρων που διαμοιράζει.

Στην παρακάτω εικόνα φαίνεται η δομή του Grid



Εικόνα 2.1 – Δομή του υπολογιστικού πλέγματος (Πηγή εικόνας : Tsouloupas G, Dikaiakos MD [6])

Η δυναμική συλλογή πόρων που αναφέρθηκε προηγουμένως είναι ομαδοποιημένη σε εικονικούς οργανισμούς (Virtual Organizations). Αυτοί αποτελούνται από μεγάλο αριθμό ετερογενών πόρων (υλικό και λογισμικό) τα οποία είναι κατανεμημένα σε διαφορετικούς διαχειριστικούς οργανισμούς και είναι συνδεδεμένα μέσω του διαδικτύου. Η πρόσβαση στους πόρους αυτούς επιτυγχάνεται μέσω ενδιάμεσου λογισμικού (middleware) που υπάρχει εγκατεστημένο στο υπολογιστικό πλέγμα και το οποίο λειτουργεί είτε ως μια Διεπαφή Προγραμματισμού Εφαρμογών για τους προγραμματιστές, είτε ως ένα μέσο επικοινωνίας με τους χρήστες [3].

Η υποδομή υπολογιστικού πλέγματος (grid infrastructure) συμπεριλαμβάνει γεωγραφικά κατανεμημένες τοποθεσίες (grid sites) οι οποίες ανήκουν σε διαφορετικούς διαχειριστικούς οργανισμούς, επικοινωνούν μέσω του διαδικτύου και υποστηρίζουν ένα ή και περισσότερους εικονικούς οργανισμούς. Μια τέτοια τοποθεσία (grid site) αποτελείται, όπως φαίνεται και στην εικόνα 2.1, από συστατικές υπολογιστών που ονομάζονται worker nodes, ένα κεντρικό υπολογιστή που διαχειρίζεται τις εργασίες των χρηστών και ονομάζεται Computing Element και

πιθανώς ένα χώρο αποθήκευσης (Storage Element). Όλα αυτά είναι διασυνδεδεμένα μέσω τοπικού δικτύου υψηλής ταχύτητας. Επίσης κάθε τοποθεσία περιλαμβάνει κάποιες τοπικές υπηρεσίες οι οποίες αφορούν τη συμμετοχή αυτής σε μια υποδομή υπολογιστικού πλέγματος. Τέλος η υποδομή υπολογιστικού πλέγματος συμπεριλαμβάνει κάποιες κεντρικές υπηρεσίες που παρέχουν πληροφορίες σχετικά με την αποστολή και τη διαχείριση εργασιών (job submission, job control) κτλ .

Η ορθή αποστολή μιας εργασίας ενός χρήστη στο υπολογιστικό πλέγμα, περιλαμβάνει μια σειρά βημάτων όπου η πιθανότητα για αναμονή, καθυστέρηση ή σφάλμα είναι μεγάλη. Έτσι το ζήτημα της ποιότητας των υπηρεσιών αποκτά ιδιαίτερη σημασία ειδικά αν αξιοποιηθεί το υπολογιστικό πλέγμα εμπορικά, όπου οι προσφερόμενοι πόροι θα διατίθενται έναντι κάποιας τιμής. Πλέον γίνεται αντιληπτό ότι η μέτρηση απόδοσης (benchmarking) αποκτά ιδιαίτερη σημασία.

2.3 Μέτρηση επίδοσης στο υπολογιστικό πλέγμα (Grid Benchmarking)

Σε αυτή την ενότητα θα γίνει αναφορά στα προβλήματα και τις δυσκολίες που προκύπτουν από τη μέτρηση επίδοσης στο υπολογιστικό πλέγμα. Ορίζουμε ως μέτρηση επίδοσης στο υπολογιστικό πλέγμα (Grid Benchmarking), τη χρήση των κατάλληλων benchmarks με σκοπό τον επαρκή και περιεκτικό χαρακτηρισμό της επίδοσης διαφόρων πτυχών της υποδομής του υπολογιστικού πλέγματος [4].

Τα παραδοσιακά benchmarks δεν μπορούν απευθείας να εκτελεστούν στο υπολογιστικό πλέγμα, επειδή υπάρχει ένα επιπλέον σημαντικό κόστος από τη χρήση των υπηρεσιών του πλέγματος καθώς επίσης και λόγω των διαφόρων ζητημάτων που προκύπτουν από τη δομή και την αρχιτεκτονική της υποδομής του υπολογιστικού πλέγματος.

Για παράδειγμα η επίδοση του υπολογιστικού πλέγματος μπορεί να επηρεαστεί από την επίδοση των μεμονωμένων στοιχείων που την αποτελούν. Από την επίδοση του υλικού, την επίδοση του δικτύου, την αξιοπιστία του ενδιάμεσου λογισμικού που χρησιμοποιείται, την επίδοση βιβλιοθηκών και υπηρεσιών καθώς και από προβλήματα που προκύπτουν από την διαχειριστική αρχή.

Λόγω όλων αυτών των παραγόντων είναι επιτακτική η ανάγκη τα ίδια benchmarks να εκτελούνται πολλές φορές και να διεξάγεται στατιστική ανάλυση προκειμένου τα αποτελέσματα να έχουν στατιστική σημαντικότητα και έτσι η εξαγωγή των αποτελεσμάτων να έχει νόημα. Επίσης τα benchmarks πρέπει να ορίζονται και να περιγράφονται με τέτοιο τρόπο ώστε να μπορεί να εξασφαλιστεί η μεταφορά τους σε οποιοδήποτε ενδιάμεσο λογισμικό με ελάχιστο κόστος [3].

2.4 Κατηγορίες στη μέτρηση επίδοσης στο υπολογιστικό πλέγμα

Το τελευταίο καιρό αναπτύχθηκαν πολλά ερευνητικά προγράμματα τα οποία προσπαθούν να επικεντρωθούν σε διάφορες πτυχές της μέτρησης απόδοσης στο υπολογιστικό πλέγμα με την χρήση διαφόρων εργαλείων και τεχνικών. Θα αναφερθούν κάποιες από αυτές [3]

- Μέτρηση επίδοσης στην υποδομή του υπολογιστικού πλέγματος.

Σε αυτή την περίπτωση επικεντρωνόμαστε στο ανώτερο επίπεδο αρχιτεκτονικής της υποδομής του πλέγματος και χρησιμοποιούμε benchmarks τα οποία προσομοιώνουν workloads προκειμένου να ελέγξουν την συμπεριφορά του συστήματος σε ακραίες καταστάσεις.

- Μέτρηση επίδοσης στις υπηρεσίες του υπολογιστικού πλέγματος. Προκειμένου να ελεγχθούν οι υπηρεσίες του πλέγματος έχουν αναπτυχθεί benchmarks τα οποία μετρούν το χρόνο απόκρισης αυτών. Για παράδειγμα δημιουργούνται ακολουθίες ερωτήσεων στο σύστημα πληροφοριών (information system) και αξιολογούν το χρόνο απόκρισης.

2.5 Κατηγορίες των benchmarks

Όπως γίνεται εύκολα αντιληπτό ο σχεδιασμός των benchmarks πρέπει να γίνεται με τέτοιο τρόπο ώστε να μπορεί να εξασφαλιστεί ο έλεγχος της επίδοσης όλης της υποδομής του υπολογιστικού πλέγματος. Από τις CPUs των υπολογιστών, μέχρι τις υπηρεσίες, το δίκτυο κτλ. Με βάση αυτά μπορεί να γίνει ο διαχωρισμός των benchmarks σε τρεις κατηγορίες. Κάθε κατηγορία παρέχει εν τέλει πληροφορίες για διαφορετικές πτυχές της υποδομής του υπολογιστικού πλέγματος [3].

- Micro benchmarks

Σε αυτή την κατηγορία απομονώνουμε κάποια οντότητα του πλέγματος και τη μετράμε σαν να ήταν τελείως αυτόνομη σε σχέση με ολόκληρη την υποδομή. Για παράδειγμα μετράμε την επίδοση των επεξεργαστών (πχ σε πράξεις κινητής υποδιαστολής ανά δευτερόλεπτο).

- Micro kernels

Σε αυτή την κατηγορία μετράμε την επίδοση ενός συνόλου οντοτήτων που εντάσσονται στην υποδομή του πλέγματος. Για παράδειγμα η μέτρηση της επίδοσης στην παράλληλη επικοινωνία των CPUs στα πλαίσια μια τοποθεσίας (grid site).

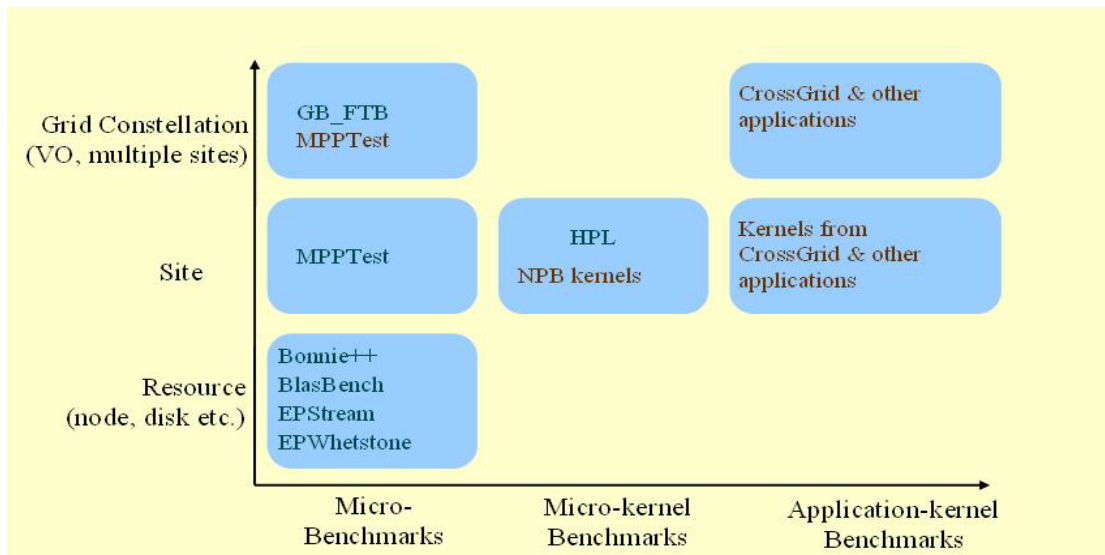
- Application kernels και Grid applications [20]

Σε αυτή την περίπτωση μετράμε την επίδοση μιας οντότητας του πλέγματος σε όσο το δυνατόν περισσότερο ρεαλιστικές συνθήκες. Για παράδειγμα εκτελούμε ένα κομμάτι (demo) μιας πραγματικής εφαρμογής και μετράμε τα δευτερόλεπτα που θέλει για να εκτελεστεί.

Ο διαχωρισμός που φαίνεται στην εικόνα 2.2 εξασφαλίζει τη μέτρηση επίδοσης σε

- Ανεξάρτητους πόρους, όπως CPUs, δίσκους αποθήκευσης κτλ
- Τοποθεσίες (sites), δηλαδή συλλογές πόρων που είναι διασυνδεδεμένες μέσω τοπικού δικτύου (και ανήκουν στην ίδια διαχειριστική αρχή).
- Το ενδιάμεσο λογισμικό, το οποίο είναι λογισμικό που παρέχει την πρόσβαση στους διαμοιραζόμενους πόρους και δίνει την αίσθηση ενός ενιαίου συστήματος.

Στην παρούσα διπλωματική εργασία αναπτύχθηκαν και ενσωματώθηκαν στο σύστημα benchmarks που ανήκουν στην κατηγορία Micro Benchmarks. Λεπτομερής αναφορά για κάθε benchmark που αναπτύχθηκε, υπάρχει στο παράρτημα.



Εικόνα 2.2 – Κατηγορίες των benchmarks (Πηγή εικόνας : Dikaiakos MD [3])

2.6 Dwarfs

Ένα dwarf είναι μια αλγοριθμική μέθοδος η οποία ακολουθεί ένα συγκεκριμένο μοτίβο στον υπολογισμό και την επικοινωνία [5]. Τα dwarfs είναι εμπνευσμένα από τον Phil Colella ο οποίος θεωρεί ότι υπάρχουν 7 υπολογιστικές μέθοδοι οι οποίες θα είναι σημαντικές για την επιστήμη και τη μηχανική των υπολογιστών τα επόμενα χρόνια.

Τα 7 dwarfs αποτελούν 7 αντίστοιχες κλάσεις όπου κάθε κλάση χαρακτηρίζεται από την ομοιότητα στον υπολογισμό. Προγράμματα που ανήκουν σε κάποια κλάση μπορεί να υλοποιούνται διαφορετικά αλλά υπάρχει η πεποίθηση ότι βασικές αρχές και ιδιότητες θα παραμείνουν αναλλοίωτες στον χρόνο. Παρακάτω παρατίθενται οι κατηγορίες αυτές.

- Dense Linear Algebra
- Sparse Linear Algebra
- Spectral Methods
- N-Body Methods
- Structured Grids
- Unstructured Grids
- MapReduce
- Combinational Logic
- Graph Traversal
- Dynamic Programming
- Backtrack and Branch-and-Bound
- Graphical Models
- Finite State Machines

Επομένως εκτελώντας ένα dwarf και παίρνοντας αποτελέσματα για αυτό, καλύπτουμε μια συγκεκριμένη κατηγορία από αντίστοιχες της κατηγορίας αυτής εφαρμογές. Με αυτό τον τρόπο δεν απαιτείται η εκτέλεση κάθε εφαρμογής ξεχωριστά αλλά ενός dwarf (της κατηγορίας αυτής) παίρνοντας περίπου ίδια αποτελέσματα.

2.7 GridBench

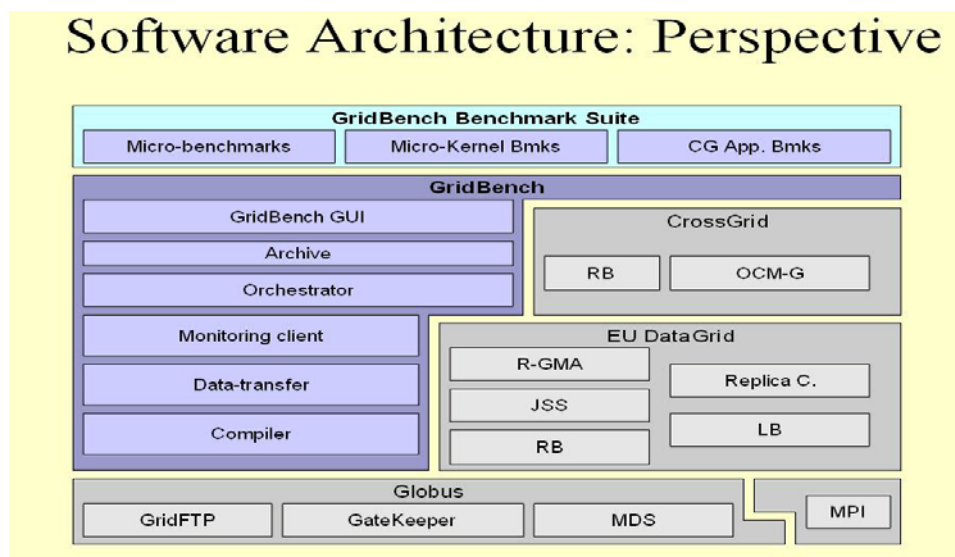
Το GridBench αποτελεί ένα εργαλείο για το σχεδιασμό, την προσαρμογή και εκτέλεση benchmarks στο υπολογιστικό πλέγμα [6]. Παρέχει μια πύλη (portal) για την εκτέλεση των benchmarks, τη συλλογή και την ανάλυση των αποτελεσμάτων. Η ανάλυση και η ερμηνεία των αποτελεσμάτων είναι το ίδιο σημαντική με τη συλλογή αυτών και το GridBench (το οποίο δημιουργήθηκε στο Πανεπιστήμιο Κύπρου στα πλαίσια του ερευνητικού προγράμματος CrossGrid) προσφέρει ένα πολύ χρήσιμο πλαίσιο εργασίας (framework) για τη επίτευξη αυτών. Μια περιγραφική γλώσσα χρησιμοποιείται (GBDL) για την ρύθμιση και προσδιορισμό των benchmarks (περισσότερες πληροφορίες για την γλώσσα αυτή θα δοθούν σε παρακάτω κεφάλαιο). Διάφορα πειράματα (εκτελέσεις benchmarks) μπορούν να πραγματοποιηθούν πολλές φορές και ειδικά εργαλεία για ανάλυση των αποτελεσμάτων βοηθούν στην ερμηνεία αυτών.

2.7.1 Αρχιτεκτονική του GridBench

Το GridBench αποτελείται από δύο συνιστώσες [7]. Το GridBench Benchmark Suite και το GridBench Framework. Το GridBench Benchmark Suite περιλαμβάνει τα micro benchmarks, micro kernel benchmarks και application benchmarks τα οποία είναι προσεχτικά επιλεγμένα ώστε να εξασφαλίζουν την καλύτερη δυνατή περιγραφή της επίδοσης του υπολογιστικού πλέγματος. Το GridBench framework παρέχει όλες τις απαραίτητες λειτουργίες που απαιτούνται για τον ορισμό και την εκτέλεση των benchmarks όπως επίσης και όσον αναφορά τη συλλογή, αποθήκευση και ανάλυση των αποτελεσμάτων.

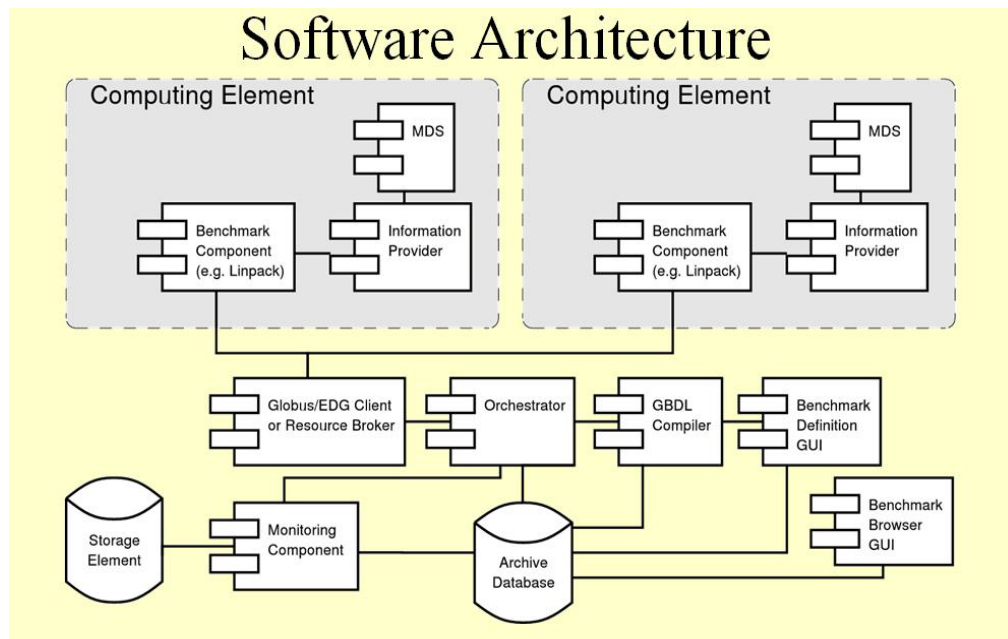
Τέλος η σχεδίαση του πλαισίου εργασίας έγινε με τέτοιο τρόπο ώστε να είναι ανεξάρτητο από οποιοδήποτε ενδιάμεσο λογισμικό του υπολογιστικού πλέγματος και άρα εύκολη η προσαρμογή του σε οποιαδήποτε υποδομή.

Στην παρακάτω εικόνα παρουσιάζεται η αρχιτεκτονική του λογισμικού



Εικόνα 2.3 – Αρχιτεκτονική GridBench (Πηγή εικόνας : Tsouloupas G, Dikaiakos MD [6])

Το πλαίσιο εργασίας όπως αναφέρθηκε αποτελείται από διάφορες υπό-μονάδες που αφορούν την εκτέλεση, συλλογή και αποθήκευση δεδομένων καθώς και τη υπομονάδα που αφορά την εμφάνιση των αποτελεσμάτων (GUI). Το τελευταίο επίπεδο (το Globus) αφορά το ενδιάμεσο λογισμικό για το οποίο είναι υλοποιημένο, αλλά το οποίο μπορεί να προσαρμοστεί ανάλογα με την κάθε υποδομή του πλέγματος. Μια διαφορετική όψη του συστήματος φαίνεται στην παρακάτω εικόνα, όπου υπάρχει σαφέστερος διαχωρισμός των λειτουργιών της κάθε μονάδας καθώς και η σχέση μεταξύ αυτών.



Εικόνα 2.4 – Αρχιτεκτονική GridBench (Πηγή εικόνας : Tsouloupas G, Dikaiakos MD [6])

Κεφάλαιο 3

Το g Eclipse, ένα εργαλείο για χρήση του υπολογιστικού πλέγματος

Στο κεφάλαιο αυτό θα παρουσιαστεί το περιβάλλον Eclipse καθώς και το πλαίσιο εργασίας g Eclipse, το οποίο αποτελεί ένα εργαλείο για χρήση του υπολογιστικού πλέγματος το οποίο βασίζεται στο περιβάλλον του Eclipse.

3.1 Σχετικά με το περιβάλλον Eclipse

Το Eclipse [8] αποτελεί ένα ολοκληρωμένο περιβάλλον με το οποίο μπορούμε να αναπτύξουμε και να εκτελέσουμε κώδικα (Software Development Kit). Το περιβάλλον αυτό καθώς και ότι απαιτείται για την ορθή λειτουργία του, είναι ελεύθερης διανομής και ελεύθερου κώδικα βοηθώντας στη διάδοση του, τόσο στους χρήστες όσο και στις εταιρίες, μετατρέποντας τελικά το Eclipse σε ένα δυναμικό και συνεχώς αναπτυσσόμενο εργαλείο.

Παρόλο που το Eclipse έχει γραφτεί στη γλώσσα προγραμματισμού Java, η χρήση του δεν περιορίζεται σ' αυτή. Υπάρχουν διαθέσιμα (ή σχεδιάζονται) τμήματα για προσάρτηση στο Eclipse για την ανάπτυξη εφαρμογών σε γλώσσες προγραμματισμού όπως οι C/C++ και η COBOL. Το Eclipse μπορεί να χρησιμοποιηθεί επίσης και για άλλους τύπους εφαρμογών που δε σχετίζονται με την ανάπτυξη λογισμικού, όπως για παράδειγμα συστήματα διαχείρισης περιεχομένου (Content Management Systems).

Το πλαίσιο εργασίας (framework) του Eclipse [18] σχεδιάστηκε ως μια ανοιχτή πλατφόρμα με σκοπό την ανάπτυξη πλούσιων εφαρμογών «rich client applications». Η κεντρική ιδέα της αρχιτεκτονικής του Eclipse που συνέβαλε σημαντικά στην ανάπτυξη του, είναι τα βύσματα (plugins), μια ιδιαίτερη αρχιτεκτονική λογισμικού η οποία συμβάλει σε μια αρθρωτή και σταδιακή σχεδίαση που έχει ως στόχο να εμπλουτίσει την πλατφόρμα ανάπτυξης (IDE).

Κάθε βύσμα χρησιμοποιεί και βελτιώνει τη λειτουργικότητα των άλλων βυσμάτων, μέσω ενός μηχανισμού, που είναι υπεύθυνο για τον καθορισμό των συσχετίσεων μεταξύ των διαφορετικών βυσμάτων καθώς και για το πώς και πότε αυτά θα εκτελεστούν. Επιπλέον το πλαίσιο εργασίας του Eclipse βασίζεται στη λειτουργία των «προεκτάσεων» (extension points) [18], τα οποία καθορίζουν τον τρόπο με τον οποίο θα επεκταθεί η υπάρχουσα λειτουργικότητα, σχεδιάζοντας έτσι επεκτάσιμα συστατικά λογισμικού με καλά ορισμένες διεπαφές.

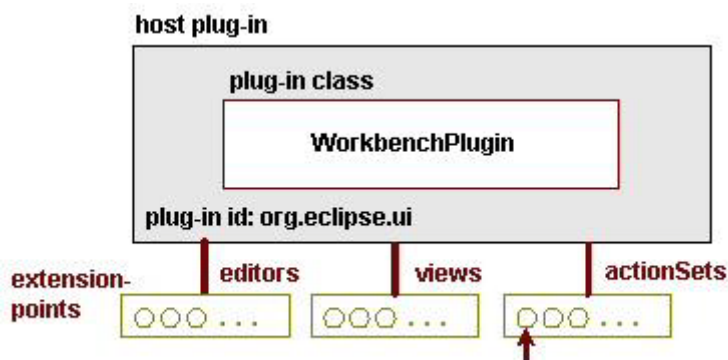
Στον πυρήνα της πλατφόρμας του Eclipse βρίσκεται ένας φορτωτής (plug in loader). Η επιπρόσθετη λειτουργία στο Eclipse πραγματοποιείται από τα βύσματα που φορτώνονται κατά την αρχικοποίηση του πλαισίου (runtime) ή όταν το πλαίσιο τα χρειαστεί. Τα βύσματα αυτά μπορούν να αφαιρεθούν ή να αντικατασταθούν ανάλογα με τις εφαρμογές. Η αναφερθείσα αρχιτεκτονική διευκολύνει επομένως την επαρκή τεκμηρίωση της, η οποία βοηθά στην καλύτερη κατανόηση της άρα εν γένει και στην καλύτερη χρήση της.

Συνοψίζοντας, επειδή τα πάντα στο Eclipse είναι διακριτά τμήματα που ενσωματώνονται, με βάση τους μηχανισμούς που προαναφέρθηκαν, δίνεται η δυνατότητα σε όσους αναπτύσσουν εργαλεία να προσφέρουν επεκτάσεις στο Eclipse και να δημιουργήσουν τελικά ένα συνεπές και ολοκληρωμένο περιβάλλον για τους χρήστες.

Το γραφικό περιβάλλον του Eclipse είναι οργανωμένο σε όψεις (views), οδηγούς (wizards), συντάκτες (editors) κτλ. Τα στοιχεία αυτά παρέχουν τη βασική λειτουργικότητα που απαιτείται προκειμένου να είναι εύκολη η ενσωμάτωση νέων στοιχείων γραφικών (GUI) στην υπάρχουσα πλατφόρμα, δημιουργώντας επιπλέον τις προοπτικές (perspective), οι οποίες ουσιαστικά αποτελούν μια συλλογή από όψεις, συντάκτες κτλ, που απαιτούνται για να εκτελέσουν ένα συγκεκριμένο στόχο όπως για παράδειγμα η ανάπτυξη ενός προγράμματος.

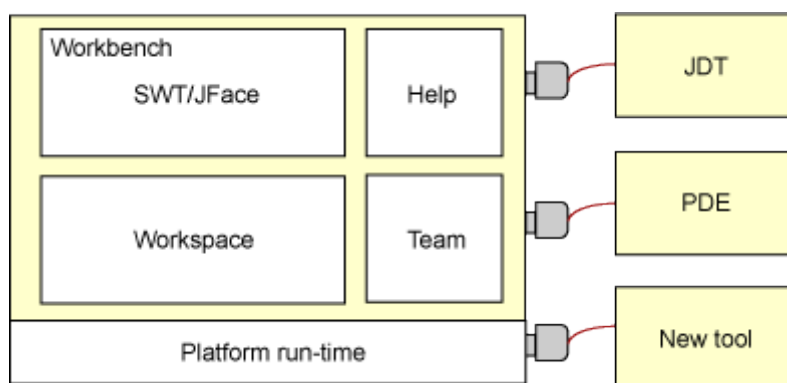
Το Eclipse προσφέρεται με προκαθορισμένες προοπτικές για την εξερεύνηση των πόρων, την ανάπτυξη java εφαρμογών, την ανάπτυξη βυσμάτων και τη διόρθωση (debugging) [18]. Η μορφή και η θέση των προοπτικών μπορούν να καθοριστούν σύμφωνα με τις ανάγκες του εκάστοτε χρήστη. Προφανώς νέες προοπτικές μπορούν να δημιουργηθούν στα πλαίσια των βυσμάτων που δημιουργούνται.

Στην εικόνα 3.1 φαίνεται πώς μέσω των προεκτάσεων αποκτούμε πρόσβαση σε δομές όπως συντάκτες, όψεις κτλ.



Εικόνα 3.1 – Δομή βυσμάτων (Πηγή εικόνας : Eclipse [8])

Οτιδήποτε αναπτύσσεται πάνω στην πλατφόρμα του Eclipse, όπως φαίνεται στην εικόνα 3.2



Εικόνα 3.2 – Δομή Eclipse (Πηγή εικόνας : Eclipse [8])

3.2 Το g Eclipse, ένα εργαλείο για χρήση του υπολογιστικού πλέγματος

Το g Eclipse [9] είναι ένα ελεύθερο λογισμικό που έχει δημιουργηθεί για χρήστες του υπολογιστικού πλέγματος και βασίζεται στην πλατφόρμα και την αρχιτεκτονική του Eclipse. Επεκτείνει τη λειτουργικότητα του Eclipse και παρέχει επεκτάσεις που επιτρέπουν στο χρήστη να έχει πρόσβαση σε υφιστάμενες υποδομές του υπολογιστικού πλέγματος.

Το g Eclipse παρέχει λειτουργίες που διευκολύνουν το χρήστη στην αποστολή εργασιών στο υπολογιστικό πλέγμα (grid job), στη διαχείριση ενός grid site και στην ανάπτυξη εφαρμογών για το υπολογιστικό πλέγμα. Ένας χρήστης του υπολογιστικού πλέγματος μπορεί να έχει διακριτούς ρόλο η ανάλογα με τους στόχο η που εξυπηρετεί. Το g Eclipse αποτελεί ένα εργαλείο το οποίο παρέχει τις απαιτούμενες προοπτικές με βάση τους ρόλους των χρηστών του πλέγματος και οι οποίες φαίνονται παρακάτω [10] :

Προοπτική Χρήστη

Οι χρήστες του Grid έχουν περιορισμένη τεχνογνωσία σε σχέση με τα υπολογιστικά πλέγματα. Διαμέσου της «Προοπτικής Χρήστη» παρέχεται η δυνατότητα στο χρήστη να εκτελέσει εφαρμογές στο υπολογιστικό πλέγμα.

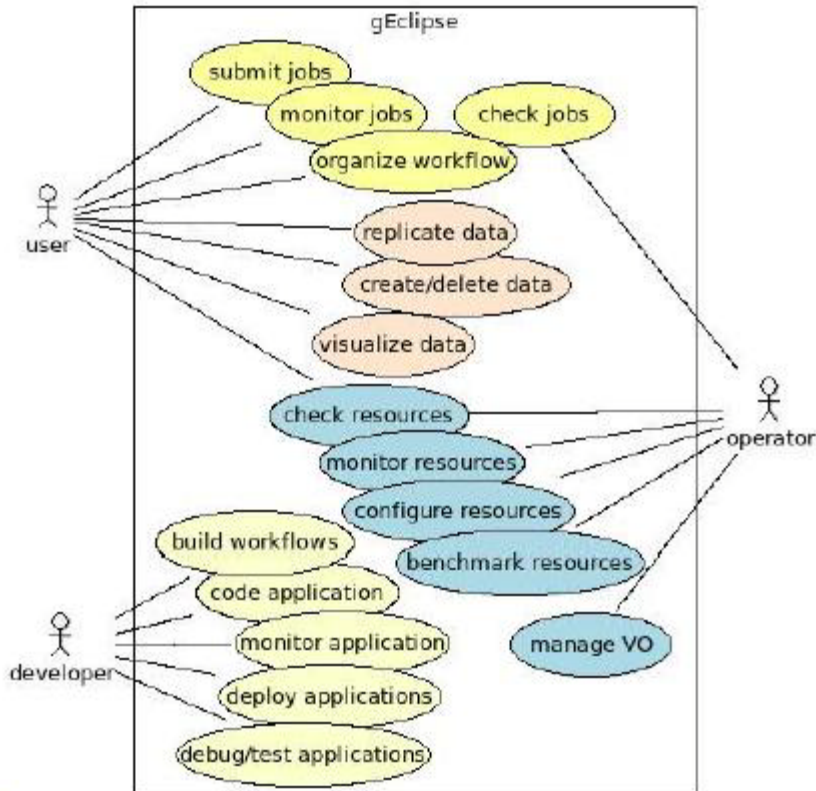
Παράλληλα δίνεται η δυνατότητα παρακολούθησης της προόδου εκτέλεσης της εφαρμογής καθώς και διαχείρισης αρχείων δεδομένων στο υπολογιστικό πλέγμα.

Προοπτική Χειριστή

Οι χειριστές υποδομών υπολογιστικού πλέγματος κατέχουν την τεχνογνωσία διαχείρισης τέτοιων υποδομών. Η «Προοπτική Χειριστή» πρέπει να παρέχει την ευχέρεια στους χειριστές τέτοιων υποδομών να διαχειρίζονται τοπικούς πόρους καθώς και πόρους που ανήκουν στο Virtual Organization στο οποίο ανήκουν.

Προοπτική Προγραμματιστή

Οι προγραμματιστές εφαρμογών για το υπολογιστικό πλέγμα κατέχουν την απαραίτητη τεχνογνωσία για ανάπτυξη εφαρμογών σε τέτοιο περιβάλλον αλλά δεν αναμένεται από αυτούς γνωρίζουν εις βάθος την τεχνολογία πίσω από το υπολογιστικό πλέγμα ή τις υποδομές που το στηρίζουν. Η «Προοπτική Προγραμματιστή» επιδιώκει να προσφέρει στους προγραμματιστές τα κατάλληλα εργαλεία για την ανάπτυξη εφαρμογών και εξουδετέρωση τυχών σφαλμάτων στο υπολογιστικό πλέγμα.



Εικόνα 3.3 – Περιπτώσεις χρηστών g Eclipse (Πηγή εικόνας : g Eclipse[9])

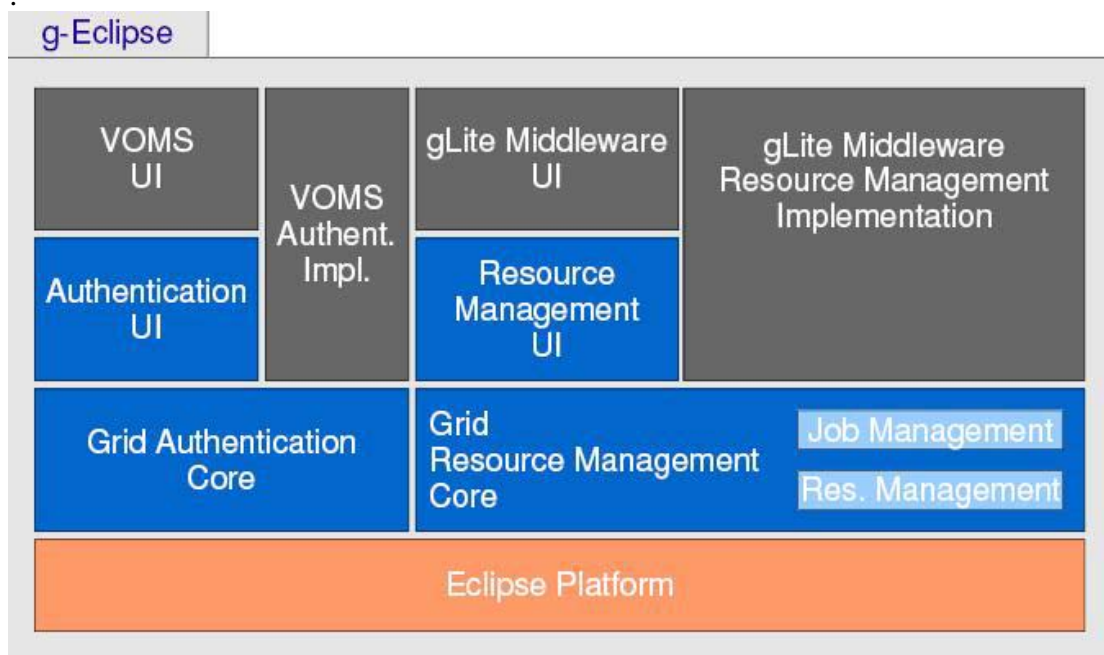
3.3 Το πλαίσιο εργασίας του g Eclipse

Εκτός από τους ρόλους που περιγράφηκαν προηγουμένως, οι χρήστες του πλαισίου εργασίας του g Eclipse (g eclipse framework) [11] μπορούν να χωριστούν σε δύο κατηγορίες ανάλογα με τη χρήση που κάνουν. Οι περισσότεροι από τους χρήστες χρησιμοποιούν το g Eclipse ως ένα γραφικό εργαλείο εντός του Eclipse και χρησιμοποιούν τα γραφικά στοιχεία που προσφέρονται διότι αποκρύπτουν επιτυχώς την πολυπλοκότητα του υπολογιστικού πλέγματος (δηλαδή το χρησιμοποιούν ως ένα γραφικό περιβάλλον - GUI).

Η δεύτερη κατηγορία χρηστών χρησιμοποιεί το g Eclipse ως ένα πλαίσιο εργασίας (με τη λογική της διεπαφής προγραμματισμού εφαρμογών - API) για τη δημιουργία δικών τους εφαρμογών βασισμένη στο μοντέλο (core) του g Eclipse και όχι στα γραφικά στοιχεία (UI components).

Για αυτούς τους λόγους το g Eclipse σχεδιάστηκε ώστε να υπάρχει πλήρης διαχωρισμός μεταξύ του γραφικού μέρους από το μοντέλο, προσφέροντας ταυτόχρονα μια εύχρηστη διεπαφή προγραμματισμού εφαρμογών - API με τις αντίστοιχες προεκτάσεις (extension points).

Το g Eclipse έχει επίσης σχεδιαστεί έχοντας ως κεντρικό άξονα ένα μοντέλο ανεξάρτητο από οποιαδήποτε υλοποίηση ενδιάμεσου λογισμικού (middleware). Η υποστήριξη κάποιου συγκεκριμένου ενδιάμεσου λογισμικού (πχ gLite [19]) υλοποιείται με την επέκταση του ανεξάρτητου μοντέλου. Στην εικόνα φαίνεται η αρχιτεκτονική και η δομή του g Eclipse.



Εικόνα 3.4 – Δομή του g Eclipse (Πηγή εικόνας : g Eclipse [9])

3.4 Μετρήσεις Επιδόσεων (benchmarking) στο υπολογιστικό πλέγμα με την χρήση του g Eclipse

Οι πόροι στο υπολογιστικό πλέγμα είναι ετερογενείς όσον αναφορά τις υπολογιστικές δυνατότητες, καθώς και στην ποικιλία διαφορετικών ρυθμίσεων (που συνήθως καθορίζονται από τον διαχειριστή) έχοντας ως αποτέλεσμα διαφορετικές επιδόσεις στις εργασίες (jobs) των χρηστών. Για αυτό το λόγο οι χρήστες πρέπει να μπορούν να αξιολογούν τους υπολογιστικούς πόρους, ώστε να επιλέγουν τους καλύτερους για τις εφαρμογές που τους ενδιαφέρουν. Επομένως η ανάγκη για ένα διαδραστικό εργαλείο που θα αξιολογεί τους υπολογιστικούς πόρους είναι επιτακτική.

Οι χρήστες του υπολογιστικού πλέγματος μπορούν να πραγματοποιούν μετρήσεις επιδόσεων μέσω του g Eclipse. Το g Eclipse ενσωματώνει ένα εργαλείο το οποίο επιτρέπει την περιγραφή, δημιουργία και εκτέλεση benchmarks στο υπολογιστικό πλέγμα καθώς επίσης και τη δυνατότητα να αποθηκεύει τοπικά τα αποτελέσματα επιτρέποντας έτσι ταυτόχρονα την παρουσίαση του ιστορικού αυτών σε γραφικό περιβάλλον.

Το εργαλείο αυτό ενσωματώνει πολλά χαρακτηριστικά και ιδιότητες από το ερευνητικό έργο GridBench που αναπτύχθηκε στο Πανεπιστήμιο Κύπρου (όπως προαναφέρθηκε στο κεφάλαιο 2). Έχει λοιπόν δημιουργηθεί εντός του g Eclipse ένα πλαίσιο εργασίας, το οποίο είναι υπεύθυνο για την ορθή αποστολή και παραλαβή των αποτελεσμάτων οποιουδήποτε benchmark. Για να επιτευχθεί όμως η επικοινωνία αυτή, έχει χρησιμοποιηθεί μια περιγραφική xml γλώσσα με την οποία δημιουργούνται τα κατάλληλα xml αρχεία τα οποία καθορίζουν τη μορφή των εισόδων και των εξόδων των benchmarks, καθώς επίσης και οποιαδήποτε επιπλέον ρύθμιση είναι απαραίτητη για την ορθή αποστολή και παραλαβή των αποτελεσμάτων.

Η γλώσσα αυτή (Grid Benchmark Description Language – GBDL) [7] δημιουργήθηκε στα πλαίσια του GridBench και κάθε benchmark οφείλει να περιγράφεται μέσω αυτής προκειμένου να μπορεί να εκτελεστεί και εντός του g Eclipse.

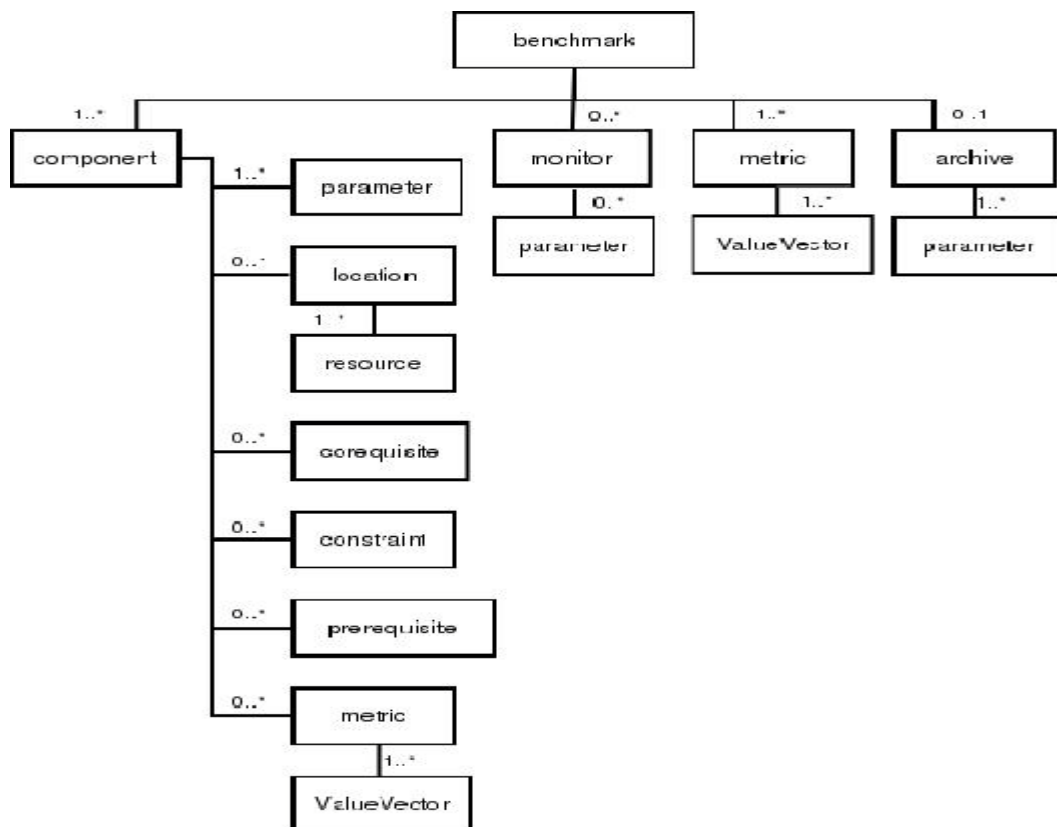
3.5 Ερμηνεία των GBDL αρχείων

Η δημιουργία και επεξεργασία των GBDL αρχείων αποτελούν την ουσία του πλαισίου εργασίας της μέτρησης επιδόσεων στο g Eclipse για αυτό και θα δοθεί μια πλήρη αναφορά στην γλώσσα. Όπως αναφέρθηκε η GBDL γλώσσα κωδικοποιεί βασικές πληροφορίες οι οποίες είναι απαραίτητες για την περιγραφή και εκτέλεση των benchmarks. Η γλώσσα έχει σχεδιαστεί ώστε :

- Η περιγραφή των benchmarks να είναι ανεξάρτητη από το υπάρχον ενδιάμεσο λογισμικό, που υποστηρίζεται από το εκάστοτε υπολογιστικό πλέγμα.
- Επιτρέπει να περιγράφονται πληροφορίες για την εκτέλεση των benchmarks ανά κάθε χρονική στιγμή μέσω του διαθέσιμου συστήματος παρακολούθησης πληροφοριών (monitoring system) του υπολογιστικού πλέγματος.
- Επιτρέπει τη συσχέτιση των benchmarks με συγκεκριμένες μετρικές (metrics)

Με βάση τα στοιχεία τα οποία καταγράφονται, μπορούμε να γνωρίζουμε τις συνθήκες αλλά και την υποδομή στην οποία εκτελέστηκαν τα benchmarks στο υπολογιστικό πλέγμα και έτσι να εξάγουμε καλύτερα αποτελέσματα για τους συγκεκριμένους πόρους και για τη συγκεκριμένη υποδομή.

Στην εικόνα 3.5 έχουμε μια υψηλού επιπέδου παρουσίαση της δομής της GBDL γλώσσας ενώ στη συνέχεια φαίνεται ένα παράδειγμα αυτής



Εικόνα 3.5 – Δομή γλώσσας GBDL (Πηγή εικόνας : Tsouloupas G, Dikaiakos MD [6])

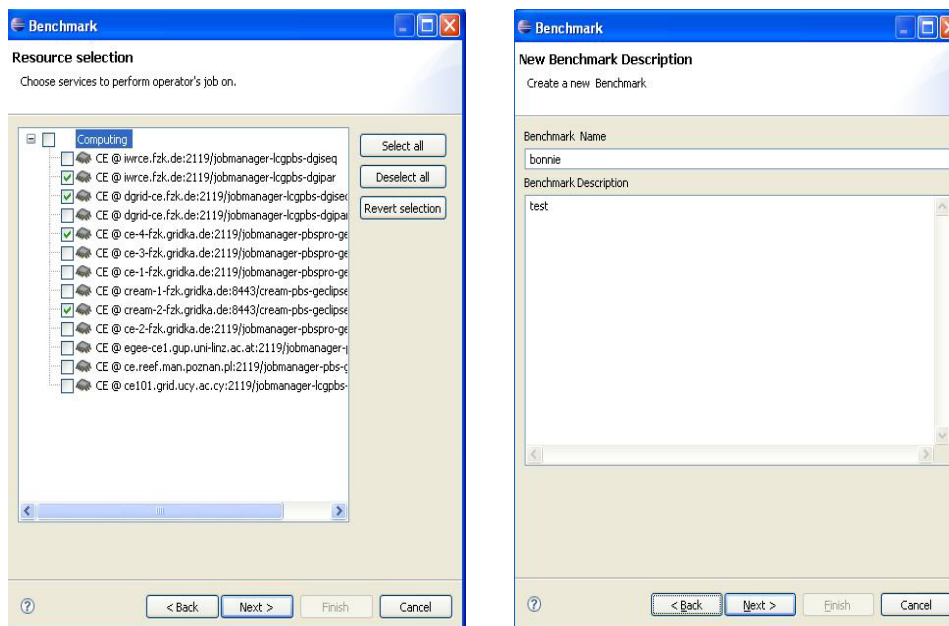
```

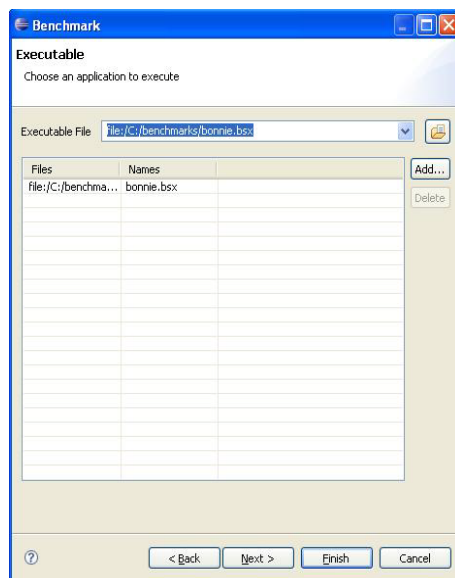
<?xml version="1.0" encoding="UTF-8"?>
<gbd1:Benchmark      xmlns:gbd1="http://www.grid.ucy.ac.cy/docs/gbd1"
description="application" name="BonnieApplication">
  <gbd1:Resource>iwrce.fzk.de</gbd1:Resource>
  <gbd1:Resource>dgrid-ce.fzk.de</gbd1:Resource>
  <gbd1:Resource>ce-3-fzk.gridka.de</gbd1:Resource>
  <gbd1:Resource>cream-1-fzk.gridka.de</gbd1:Resource>
  <gbd1:Resource>ce-2-fzk.gridka.de</gbd1:Resource>
  <gbd1:Resource>ce-3-fzk.gridka.de</gbd1:Resource>
  <gbd1:Resource>cream-1-fzk.gridka.de</gbd1:Resource>
  <gbd1:Resource>ce-5-fzk.gridka.de</gbd1:Resource>
  <gbd1:Resource>ce-6-fzk.gridka.de</gbd1:Resource>
  <gbd1:Resource>cream-1-fzk.gridka.de</gbd1:Resource>
  <gbd1:Resource>ce-4-fzk.gridka.de</gbd1:Resource>
  <gbd1:Resource>egee-ce1.gup.uni-linz.ac.at</gbd1:Resource>
  <gbd1:Resource>ce101.grid.ucy.ac.cy</gbd1:Resource>
  <gbd1:Parameter      data_type="string"      name="executable"
type="system">file:/C:/benchmarks/bonnie.bsx</gbd1:Parameter>
  <gbd1:Parameter      data_type="string"      name="stagein"
type="system">file:/C:/benchmarks/bonnie.bsx</gbd1:Parameter>
</gbd1:Benchmark>

```

3.5.1 Δημιουργία των GBDL αρχείων εντός του g Eclipse

Για τη δημιουργία των GBDL αρχείων εντός του g Eclipse [21] εκτελείται από το χρήστη ένας οδηγός (wizard). Παρακάτω φαίνονται κάποιες από τις σελίδες αυτού.

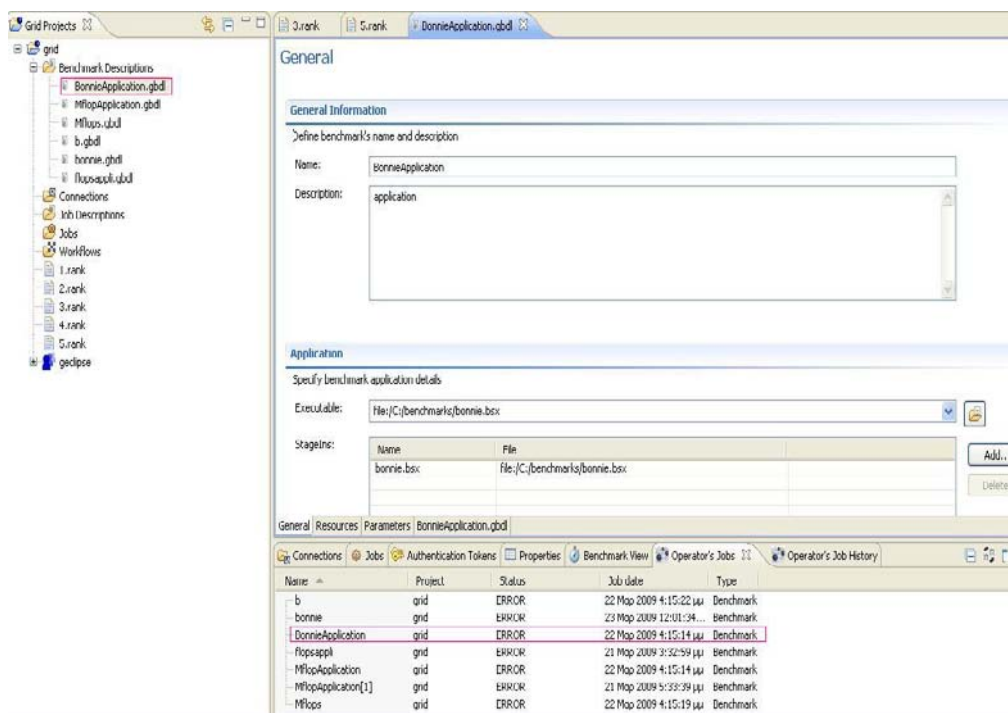




Εικόνα 3.6 – Σελίδες GBDL οδηγού

Στην πρώτη σελίδα για παράδειγμα επιλέγονται οι υπολογιστικοί πόροι που ο χρήστης επιθυμεί να αξιολογήσει, στη δεύτερη σελίδα δίνεται ένα όνομα και μια περιγραφή ενώ στην τρίτη σελίδα επιλέγεται το benchmark που θα εκτελεστεί και γενικά ότι άλλο αρχείο χρειαστεί να σταλθεί συμπληρωματικά στο υπολογιστικό πλέγμα.

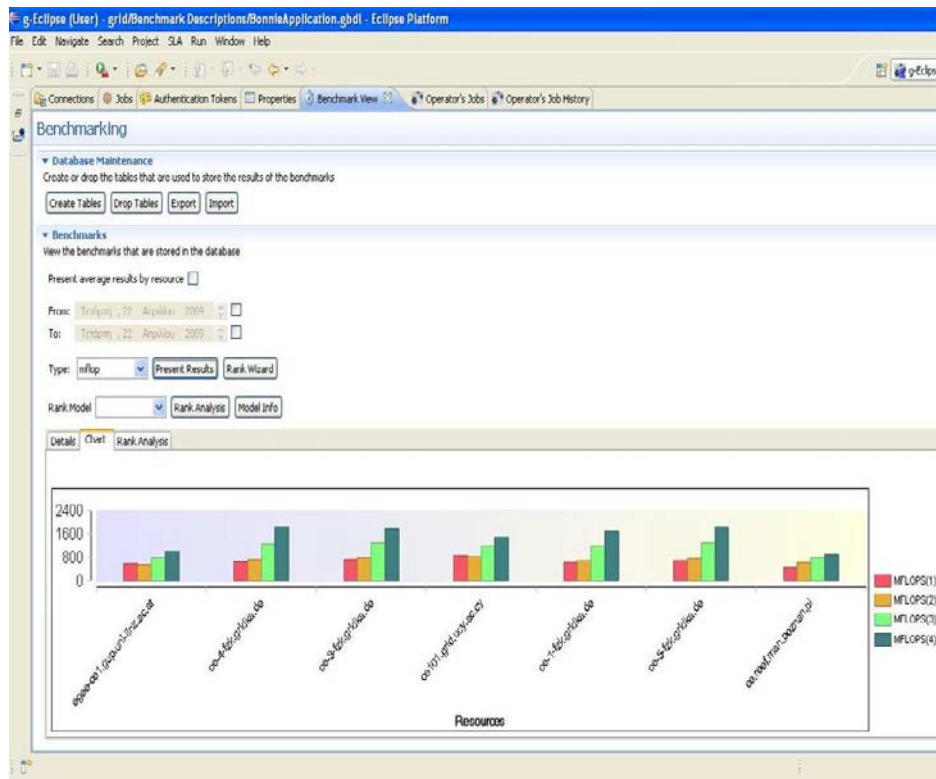
Κατά τον επιτυχή τερματισμό αυτού του οδηγού δημιουργείται ένα GBDL αρχείο όπως φαίνεται στην παρακάτω εικόνα στα αριστερά



Εικόνα 3.7 – Γραφικό περιβάλλον g Eclipse

Το GBDL αρχείο μπορεί να τροποποιηθεί μέσω του συντάκτη (editor) και φυσικά να εκτελεστεί μέσω της αποστολής του στο υπολογιστικό πλέγμα. Για περισσότερες πληροφορίες ο αναγνώστης καλείται να ανατρέξει στο user manual του g Eclipse [21].

Όπως αναφέρθηκε προηγουμένως το πλαίσιο εργασίας της μέτρησης επίδοσης επιτρέπει την παρουσίαση των αποτελεσμάτων σε γραφικό περιβάλλον ώστε να διευκολύνει την εξαγωγή και ερμηνεία των αποτελεσμάτων. Παρακάτω εμφανίζεται αυτό το γραφικό περιβάλλον όπου εμφανίζεται το ιστορικό του benchmark Mflops¹.



Εικόνα 3.8 - Γραφική απεικόνιση benchmarks

Η σχεδίαση και η αρχιτεκτονική της βάσης που χρησιμοποιήθηκε και όποιες τεχνικές λεπτομέρειες θα αναφερθούν διεξοδικά στα παρακάτω κεφάλαια όπου αυτό απαιτείται.

3.6 Το πλαίσιο εργασίας της μέτρησης επίδοσης στο υπολογιστικό πλέγμα στο εργαλείο g Eclipse

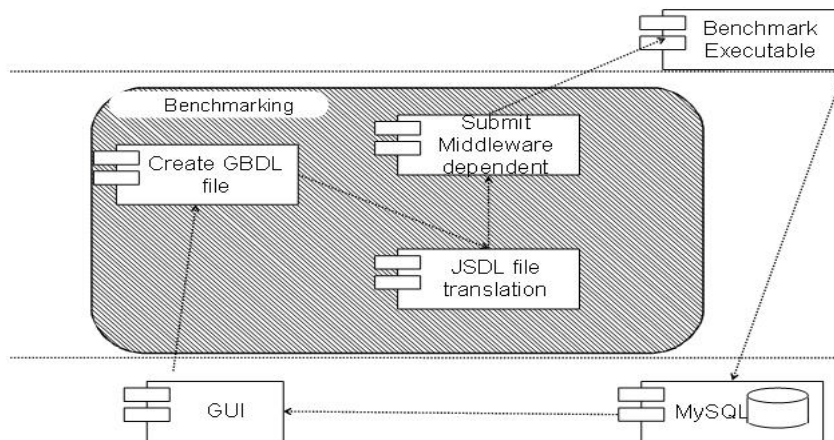
Στις προηγούμενες ενότητες παρουσιάστηκε η γλώσσα GBDL, ο τρόπος που δημιουργούνται τα αρχεία της μορφής αυτής, καθώς και το πώς εμφανίζονται τα αποτελέσματα στο χρήστη στα πλαίσια του εργαλείου g Eclipse.

Σε αυτή την ενότητα θα παρουσιαστούν τα βήματα και οι μονάδες που αποτελούν το πλαίσιο εργασίας της μέτρησης επίδοσης στο υπολογιστικό πλέγμα

Παρακάτω εμφανίζεται γραφικά η αρχιτεκτονική του πλαισίου εργασίας [7].

¹ Περισσότερα για το benchmark Mflops στο παράρτημα

g Eclipse Benchmark Framework



Εικόνα 3.9- Αρχιτεκτονική του g Eclipse Benchmark

Create GBDL file

Η πρώτη μονάδα αποτελεί τη δημιουργία του GBDL αρχείου και η οποία περιγράφηκε σε προηγούμενη ενότητα.

JSDL file Translation

Στη συνέχεια το αρχείο που δημιουργήθηκε μετατρέπεται στο Job Submission Description Language αρχείο (JSDL), (μια xml γλώσσα που χρησιμοποιείται για την περιγραφή μιας εργασίας και την οποία αντιλαμβάνεται το ενδιάμεσο λογισμικό(middleware) του υπολογιστικού πλέγματος). Για κάθε πόρο που έχει σημειωθεί στο GBDL αρχείο, ως πόρος προς μέτρηση, δημιουργείται και το αντίστοιχο JSDL αρχείο.

Submit JSDL file

Τα JSDL αρχεία αποστέλλονται αυτόματα στο υπολογιστικό πλέγμα με τη χρήση των υπάρχουσών υποδομών που έχουν αναπτυχθεί στο g Eclipse.

Benchmark Executable – MySQL

Η εργασία αυτή (το Benchmark) εκτελείται στον επιλεγθέντα πόρο και το αποτέλεσμα αυτής, επιστρέφεται πάλι με τη μορφή ενός GBDL αρχείου, οι πληροφορίες του οποίου αποθηκεύονται σε μια βάση δεδομένων.

GUI

Τέλος, τα αποτελέσματα εμφανίζονται με συγκεκριμένη μορφή (όπως φαίνονται στην εικόνα 3.8) και το οποίο επιτυγχάνεται με τις κατάλληλες επερωτήσεις στη βάση δεδομένων.

Επίσης πρέπει να αναφερθεί ό,τι υπάρχει λογικός διαχωρισμός μεταξύ του μοντέλου και της απεικόνισης των δεδομένων όπως επιβάλλεται από το πρότυπο Model–view–controller (MVC) [25], η λογική του οποίου ακολουθείται σε όλο το g Eclipse.

Κεφάλαιο 4

Ανάλυση και σχεδίαση συστήματος SiteRank

Στο κεφάλαιο αυτό θα γίνει αρχικά αναφορά στο σκοπό και στους στόχους της ανάπτυξης του συστήματος, καθώς και θα δοθεί επίσης λεπτομερώς η ανάλυση και η σχεδίαση αυτού. Κάθε δομικό κομμάτι του συστήματος θα αναλυθεί ως προς τη λειτουργικότητα και τον τρόπο ενσωμάτωσης του στο υπάρχον σύστημα της μέτρησης επίδοσης στο g Eclipse.

4.1 Σκοπός και στόχοι του συστήματος

Σκοπός του συστήματος που αναπτύχθηκε είναι η εύκολη και άμεση δημιουργία μιας μεθόδου από τον χρήστη, για την βαθμολόγηση των πόρων του υπολογιστικού πλέγματος με τη χρήση του πλαισίου εργασίας του g Eclipse.

Για παράδειγμα ο χρήστης θέλει να μετρήσει την επίδοση των πόρων σε σχέση με μια συγκεκριμένη εφαρμογή ώστε να επιλέξει τον καλύτερο δυνατό για αυτή. Η μεθοδολογία αυτή απαιτεί τη δημιουργία μιας συνάρτησης η οποία [12] :

- Δέχεται ως ορίσματα μετρικές (metrics) οι οποίες προέρχονται από χαμηλού επιπέδου benchmarks (micro benchmarks) [13].
Η επιλογή αυτών των μετρικών μπορεί να γίνει είτε αυτόματα, με βάση κάποιον αλγόριθμο, είτε από τον χρήστη. Περισσότερες λεπτομέρειες σε παρακάτω ενότητες.
- Συνδυάζονται οι μετρικές που επιλέχθηκαν σε ένα γραμμικό μοντέλο στη μορφή $R = a*metric1(\text{statistical method}) + \dots + k*metricX(\text{statistical method})$

Εν τέλει η συνάρτηση αυτή δημιουργεί μια συσχέτιση μεταξύ της επίδοσης μιας συγκεκριμένης εφαρμογής με τις μετρικές των διαφόρων micro benchmarks. Τέλος πραγματοποιείται η βαθμολόγηση των πόρων του υπολογιστικού πλέγματος και τα αποτελέσματα παρουσιάζονται σε γραφικό περιβάλλον.

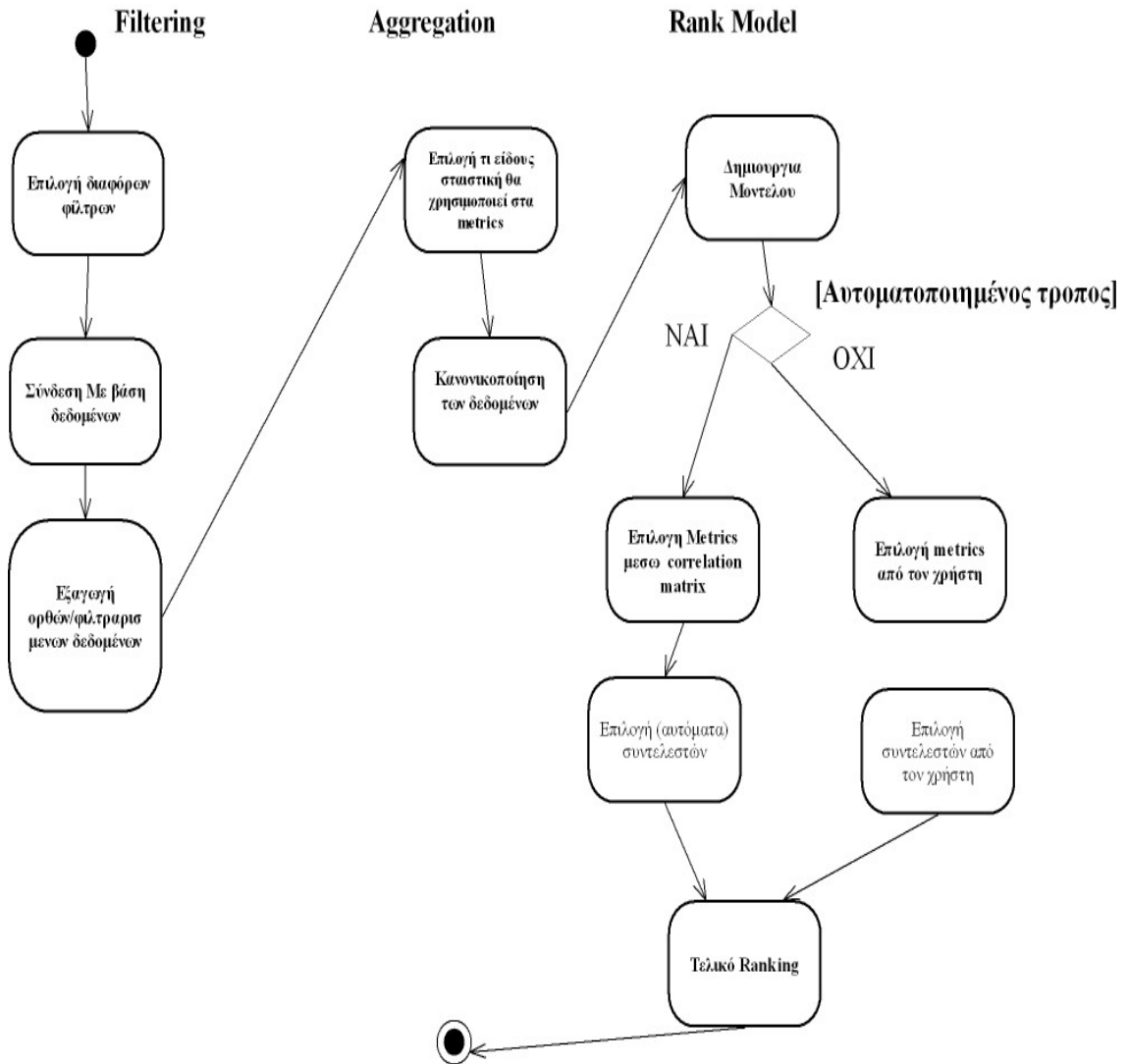
Ο χρήστης μπορεί έτσι να γνωρίζει την πιθανή επίδοση της εφαρμογής του σε κάποιον πόρο έχοντας μόνο εκτελέσει τα micro benchmarks σε αυτόν. Είναι προφανές ότι η ευκολία και η ταχύτητα να εκτελεστούν micro benchmarks στους πόρους είναι μεγαλύτερη από το να εκτελεστεί όλη η εφαρμογή (ή ακόμα και κομμάτι αυτής) σε κάθε πόρο του πλέγματος ώστε να γίνει μετά η αξιολόγηση.

Όλη αυτή η διαδικασία ενσωματώθηκε ως ένα επιπλέον εργαλείο στο υπάρχον σύστημα της μέτρησης επίδοσης του g Eclipse, δίνοντας έτσι περισσότερες πληροφορίες στους χρήστες προκειμένου να γνωρίζουν με αυξημένες πιθανότητες πού θα εκτελεστεί πιο γρήγορα η εφαρμογή που τους ενδιαφέρει. Η δημιουργία της συνάρτησης γίνεται με τη χρήση ενός οδηγού (wizard) εύκολα χωρίς να απαιτούνται ιδιαίτερες γνώσεις από το χρήστη και για την απεικόνιση των αποτελεσμάτων χρησιμοποιείται η υπάρχουσα μορφή και δομή γραφικών που προϋπήρχε στο g Eclipse ώστε να υπάρχει συμβατότητα και ομοιομορφία με αυτό.

Πολλά στοιχεία της υλοποίησης βασίζονται στο εργαλείο GridBench και το module SiteRank αυτού. Υπάρχουν όμως και κάποιες διαφορές οι οποίες θα επισημανθούν στη συνέχεια.

4.2 Γενική ροή συστήματος

Στην παρακάτω εικόνα φαίνεται το διάγραμμα ροής της διαδικασίας βαθμολόγησης των πόρων του υπολογιστικού πλέγματος, τα βήματα του οποίου υλοποιήθηκαν στα πλαίσια του συστήματος. Κάθε κομμάτι θα αναλυθεί ξεχωριστά.



Εικόνα 4.1 – Βήματα υλοποίησης συστήματος

Όπως φαίνεται και από την εικόνα προκειμένου να δημιουργηθεί το μοντέλο απαιτείται πρώτα να εκτελεστούν κάποιες απαραίτητες διαδικασίες [12].

- **Φιλτράρισμα (Filtering)**

Το φιλτράρισμα αναφέρεται στην επιλογή του χρήστη όσον αφορά το ποιά αποτελέσματα θα περιέχονται ή θα αποκλείονται από τη διαδικασία της βαθμολόγησης των πόρων μέσω της συνάρτησης που θα δημιουργηθεί. Για παράδειγμα ο χρήστης μπορεί να περιορίσει τον αριθμό των μετρικών που μπορούν να χρησιμοποιηθούν για τη δημιουργία της συνάρτησης

- Ορίζοντας κατάλληλα τις ημερομηνίες που τον ενδιαφέρουν («από» - «μέχρι»). Οποιοσδήποτε μετρικές έχουν εκτελεστεί εκτός των ημερομηνιών αυτών θα αγνοούνται από το σύστημα.
- Επιλέγοντας τον ελάχιστο αριθμό εκτελέσεων για κάθε μετρική (benchmark). Benchmarks που έχουν εκτελεστεί λιγότερες φορές από αυτή που ορίστηκε θα αγνοούνται από το σύστημα

- **Συσσωρευση(aggregation)**

Η διαδικασία αυτή προσδιορίζει το πώς «αξιοποιούνται» στατιστικά τα διάφορα αποτελέσματα που έχουν ληφθεί από τα micro benchmarks. Οι επιλογές που είναι διαθέσιμες : μέσος όρος (mean), τυπική απόκλιση (standard deviation), ελάχιστο (min), μέγιστο (max) και μέση απόκλιση (average deviation). Κάποιοι από αυτούς τους στατιστικούς όρους θα εξηγηθούν σε παρακάτω κεφάλαιο.

Στη συνέχεια αφού πραγματοποιηθούν τα βήματα αυτά οδηγούμαστε στη διαδικασία για τη δημιουργία της συνάρτησης βαθμολόγησης (Rank Model). Σε αυτή την περίπτωση έχουμε δύο επιλογές, είτε θα δημιουργήσει ο χρήστης μόνος του τη συνάρτηση είτε θα δημιουργηθεί αυτόματα από το σύστημα.

Στις παρακάτω υποενότητες θα αναλυθεί η κάθε περίπτωση ξεχωριστά και θα δοθούν τα απαιτούμενα βήματα και το πώς πρέπει να αποκρίνεται το σύστημα. Στην ενότητα που αναφέρεται στην υλοποίηση του συστήματος θα δειχθεί πως συγκεκριμένα αποκρίνεται το σύστημα σε κάθε περίπτωση.

4.3 Δημιουργία συνάρτησης βαθμολόγησης από το χρήστη

Σε αυτή την περίπτωση ο χρήστης γνωρίζει εκ των προτέρων (a priori) τα χαρακτηριστικά της εφαρμογής που θέλει να εκτελέσει στο υπολογιστικό πλέγμα καθώς και τις μετρικές που είναι περισσότερο συσχετισμένες με αυτή, επομένως τις επιλέγει μόνος δημιουργώντας τη συνάρτηση βαθμολόγησης. Είναι προφανές ότι αυτή η διαδικασία αναφέρεται σε κάποιον έμπειρο χρήστη με γνώση σχετικά με τη μέτρηση επίδοσης συστημάτων. Αυτό είναι βέβαια δύσκολο για ένα απλό χρήστη αλλά τα αποτελέσματα είναι σίγουρα πιο ακριβή. (αυτό το στοιχείο έχει αναπτυχθεί και στο GridBench)

4.3.1 Ροή Εργασίας «δημιουργία συνάρτησης βαθμολόγησης από το χρήστη»

Ο χρήστης αφού επιλέξει τις ημερομηνίες και τον ελάχιστο αριθμό εκτελέσεων των benchmarks, ώστε να ελαχιστοποιήσει το ν αριθμό των μετρικών που θα έχει

διαθέσιμες, επιλέγει τις μετρικές που τον ενδιαφέρουν, τους αντίστοιχους συντελεστές και τη στατιστική μέθοδο που θα χρησιμοποιηθεί. Η συνάρτηση βαθμολόγησης που θα δημιουργηθεί θα χρησιμοποιηθεί από το σύστημα στη συνέχεια για την γραφική απεικόνιση των αποτελεσμάτων των πόρων που επέλεξε ο χρήστης. Παρακάτω φαίνεται η ροή εργασίας υπό την μορφή βημάτων.

Ροή δημιουργίας συνάρτησης βαθμολόγησης από το χρήστη υπό τη μορφή βημάτων και εναλλακτικών ροών

1. Επιλογή ημερομηνιών
2. Επιλογή ελάχιστων εκτελέσεων των benchmarks
3. Επιλογή μετρικής
4. Επιλογή του συντελεστή κάθε μετρικής
5. Επιλογή στατιστικής μεθόδου στα αποτελέσματα κάθε μετρικής
6. Επιλογή πόρων προς αξιολόγηση
7. Γραφική απεικόνιση των αποτελεσμάτων

Εναλλακτική ροή 1 : Σφάλμα στο βήμα 1

1. Το σύστημα πρέπει να αποκρίνεται κατάλληλα στην περίπτωση λάθους εισαγωγής ημερομηνιών.

Εναλλακτική ροή 2 : Σφάλμα στο βήμα 3

1. Δεν υπάρχουν διαθέσιμες μετρικές στο σύστημα είτε λόγω της επιλογής ημερομηνίας, είτε λόγω της επιλογής ελάχιστων εκτελέσεων των μετρικών είτε λόγω μη εκτέλεσης μετρικών. Το σύστημα πρέπει να αποκρίνεται με κατάλληλο μήνυμα λάθους.

Εναλλακτική ροή 3 : Σφάλμα στο βήμα 6

1. Δεν υπάρχουν διαθέσιμοι προς αξιολόγηση πόροι. Το σύστημα πρέπει να αποκρίνεται με κατάλληλο μήνυμα λάθους.

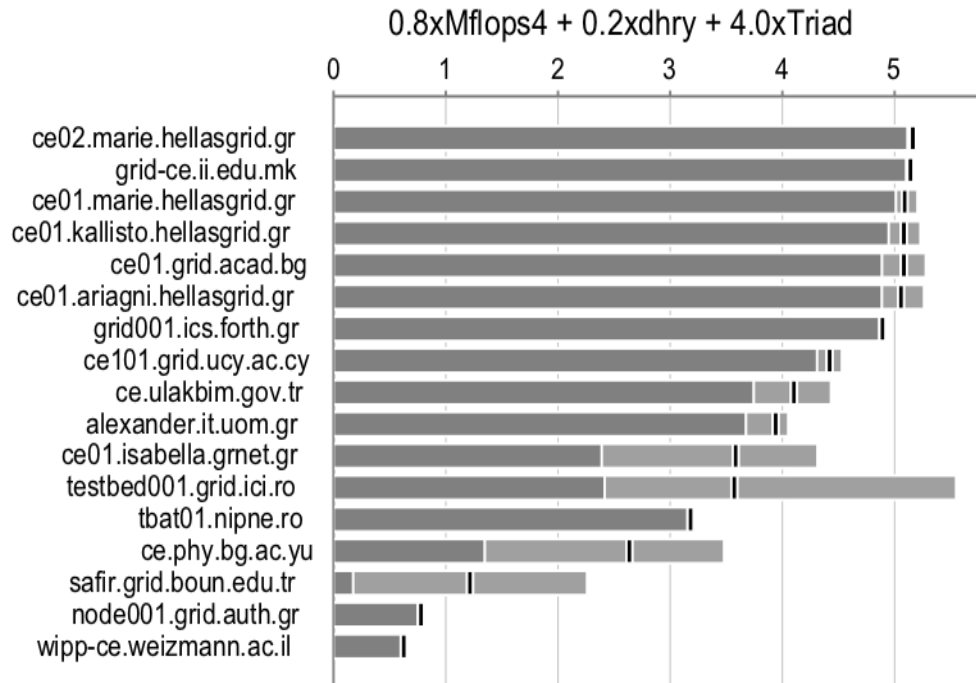
Συνάρτηση βαθμολόγησης

Μετά την ολοκλήρωση του βήματος 6 δημιουργείται το μοντέλο ή αλλιώς η συνάρτηση βαθμολόγησης. Η συνάρτηση αυτή έχει τη μορφή $R = a*metric1(statistical\ method) + \dots + k*metricX(statistical\ method)$. Κάθε (μετρική) metric αποτελεί μια επιλογή του χρήστη όπως και οι συντελεστές που του αντιστοιχούν, καθώς και η αντίστοιχη στατιστική μέθοδος. Η συνάρτηση αυτή τώρα «εκτελείται» σε κάθε πόρο και δίνει ένα αποτέλεσμα το οποίο και παρουσιάζεται γραφικά. Οι τιμές αυτές προσπαθούν να προσεγγίσουν την επίδοση της εφαρμογής εάν όντως είχε εκτελεστεί στο υπολογιστικό πλέγμα.

Παράδειγμα

Έστω ότι στον πόρο *ce101.grid.ucy.ac.cy* έχουν εκτελεστεί micro benchmarks και υπάρχουν οι μετρικές X και Y οι οποίες έχουν εκτελεστεί K φορές σε αυτόν. Έστω ότι ο χρήστης δημιούργησε τη συνάρτηση $R = 0.01*metricX(min) +$

3.1*metricY(max). Η τιμή που πρέπει να επιστρέψει η συνάρτηση είναι το αποτέλεσμα του αθροίσματος του γινομένου της ελάχιστης τιμής της μετρικής X (για K επαναλήψεις) επί τον συντελεστή 0.01 με τη μέγιστη τιμή της μετρικής Y (για K επαναλήψεις) επί τον συντελεστή 3.1. Στην παρακάτω εικόνα φαίνεται μια συνάρτηση βαθμολόγησης και η εφαρμογή της σε κάποιους πόρους.



Εικόνα 4.2 - Εφαρμογή συνάρτησης βαθμολόγησης (Πηγή εικόνας : Tsouloupas G, Dikaikos MD [12])

4.4 Δημιουργία συνάρτησης βαθμολόγησης από το σύστημα

Σε αυτή την περίπτωση ο χρήστης κάνει κάποιες ελάχιστες επιλογές και το σύστημα με βάση κάποιον αλγόριθμο δημιουργεί αυτόματα τη συνάρτηση βαθμολόγησης. Είναι προφανές ότι σε αυτή την περίπτωση δεν απαιτείται καμιά προϋπάρχουσα γνώση εκ μέρους των χρηστών αλλά τα αποτελέσματα δεν είναι πάντα ιδιαίτερα καλά και υπό συγκεκριμένες συνθήκες μπορεί και να μην υπάρχουν (αυτό το στοιχείο δεν έχει αναπτυχθεί στο GridBench).

4.4.1 Ροή Εργασίας «δημιουργία συνάρτησης βαθμολόγησης από το σύστημα»

Ο χρήστης επιλέγει αρχικά την εφαρμογή (ή application kernel), τα αποτελέσματα της οποίας υπάρχει αποθηκευμένο στη βάση δεδομένων του συστήματος, για ένα αριθμό πόρων που την έχουν ήδη εκτελέσει. Στη συνέχεια επιλέγει πόρους προς αξιολόγηση και το σύστημα δημιουργεί τη συνάρτηση την οποία μπορεί ο χρήστης και να την τροποποιήσει εάν το επιθυμεί. Τέλος η συνάρτηση αυτή χρησιμοποιείται για τη γραφική απεικόνιση των αποτελεσμάτων. Παρακάτω φαίνεται η ροή εργασίας υπό τη μορφή βημάτων.

Ροή δημιουργίας συνάρτησης βαθμολόγησης από το σύστημα υπό τη μορφή βημάτων και εναλλακτικών ροών

1. Επιλογή εφαρμογής ή application kernel τα αποτελέσματα της οποίας υπάρχουν στο σύστημα
2. Επιλογή πόρων οι οποίοι έχουν εκτελέσει την εφαρμογή που επιλέχθηκε στο προηγούμενο βήμα
3. Επιλογή πόρων προς αξιολόγηση
4. Δημιουργία και εμφάνιση μοντέλου/συνάρτησης βαθμολόγησης
5. Τροποποίηση μοντέλου/συνάρτησης βαθμολόγησης
6. Γραφική απεικόνιση των αποτελεσμάτων

Εναλλακτική ροή 1 : Σφάλμα στο βήμα 1

1. Το σύστημα πρέπει να αποκρίνεται κατάλληλα στην περίπτωση που δεν υπάρχουν αποτελέσματα οποιασδήποτε εφαρμογής

Εναλλακτική ροή 2 : Σφάλμα στο βήμα 2

1. Δεν υπάρχουν πόροι που να έχουν εκτελέσει τη συγκεκριμένη εφαρμογή και να υπάρχουν διαθέσιμα αποτελέσματα
2. Το σύστημα πρέπει να αποκρίνεται με κατάλληλο μήνυμα

Εναλλακτική ροή 3 : Σφάλμα στο βήμα 3

1. Δεν υπάρχουν διαθέσιμοι προς αξιολόγηση πόροι. Το σύστημα πρέπει να αποκρίνεται με κατάλληλο μήνυμα λάθους

Εναλλακτική ροή 4 : Σφάλμα στο βήμα 4

1. Το μοντέλο δεν μπόρεσε να δημιουργηθεί
2. Επιστροφή στο βήμα 2 για επιλογή περισσότερων/διαφορετικών πόρων
3. Εμφάνιση κατάλληλου μηνύματος από το σύστημα

4.4.2 Βήματα του αλγορίθμου για τη δημιουργία της συνάρτησης βαθμολόγησης από το σύστημα

Δειγματοληψία

Για ένα μικρό δείγμα πόρων επιλέγεται ένα σύνολο μετρικών m . Για το ίδιο δείγμα πόρων επιμετράται η επίδοση της εφαρμογής προς αξιολόγηση. Η επίδοση αυτή μπορεί να είναι σε sec, fops κτλ και ορίζεται ως α [12]. Το ποσοστό δειγματοληψίας κυμαίνεται περίπου στο 10-15% όλων των πόρων του εικονικού Οργανισμού (VO).

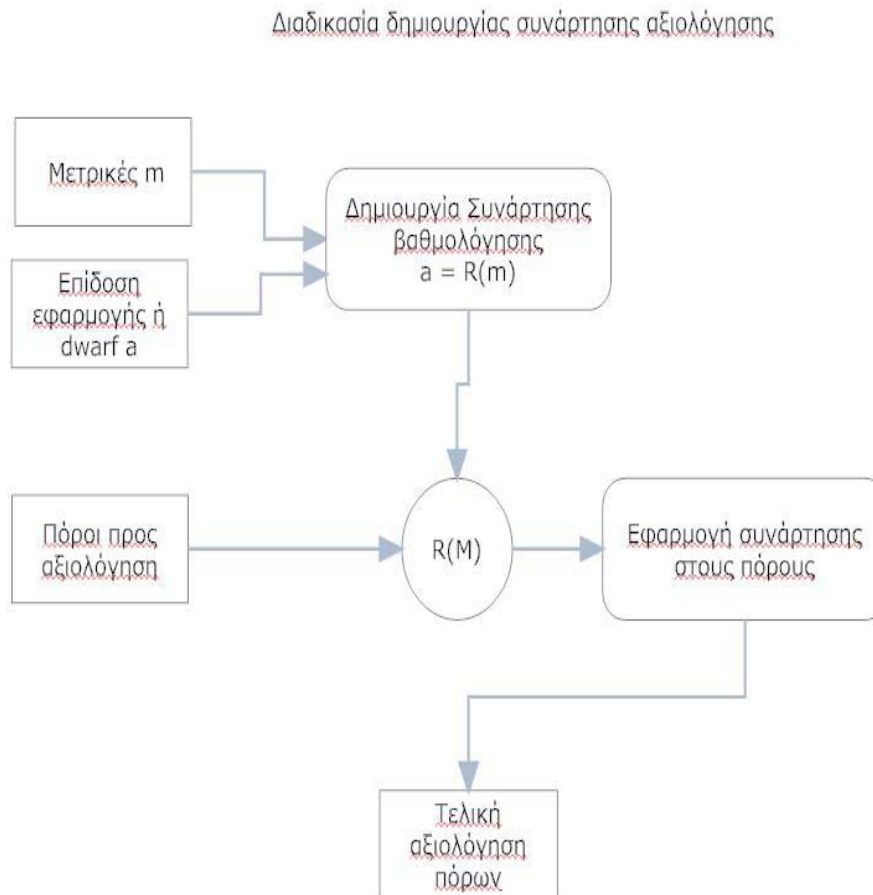
Δημιουργία Συνάρτησης βαθμολόγησης

Η συνάρτηση που δημιουργείται, η οποία ορίζεται ως R , βασίζεται στις μετρικές m και στην επίδοση της εφαρμογής a υπό την σχέση $a = R(m)$. Ουσιαστικά αυτό σημαίνει ότι επιλέγονται οι μετρικές που είναι περισσότερο συσχετισμένες με την επίδοση της εφαρμογής.

Υπολογισμός με βάση τη συνάρτηση βαθμολόγησης

Για το σύνολο των πόρων που έχει επιλέξει ο χρήστης (για αξιολόγηση), χρησιμοποιούνται οι μετρικές M και εφαρμόζεται η συνάρτηση που δημιουργήθηκε προκειμένου να γίνει εκτίμηση της επίδοσης της εφαρμογής A_{est} υπό την μορφή $A_{est} = R(M)$.

Παρακάτω παρουσιάζεται η διαδικασία της δημιουργίας και εφαρμογής της συνάρτησης βαθμολόγησης με τη μορφή σχεδιαγράμματος



Εικόνα 4.3 – Διαδικασία δημιουργίας συνάρτησης αξιολόγησης - βαθμολόγησης

4.4.3 Αλγόριθμος δημιουργίας της συνάρτησης βαθμολόγησης

Σε αυτή την υπό ενότητα θα δοθεί μια λεπτομερής περιγραφή για το πώς δημιουργείται η συνάρτηση βαθμολόγησης με τη χρήση φυσικής γλώσσας υπό μορφή βημάτων από το σύστημα.

1. Φορτώνονται από τη βάση δεδομένων όλες οι μετρικές που υπάρχουν.
2. Δημιουργείται ένας πίνακας συσχέτισης (Pearson correlation matrix) μεταξύ των μετρικών του βήματος 1 και της εφαρμογής που έχει επιλέξει ο χρήστης. (εικόνα 4.4)
 - a. Οι τιμές των μετρικών, με βάση τις οποίες θα δημιουργηθεί ο πίνακας συσχέτισης, προέρχονται από τους πόρους που επέλεξε ο χρήστης ως πόρους δειγματοληψίας.
 - b. Οι τιμές της εφαρμογής/dwarf προέρχονται από τους πόρους που επέλεξε ο χρήστης ως πόρους δειγματοληψίας.

3. Μετρικές που η συσχέτιση τους με την εφαρμογή είναι κάτω από μια συγκεκριμένη τιμή αποκόπτονται. Αυτό πραγματοποιείται γιατί ουσιαστικά κάτω από μια τιμή δεν υπάρχει καμία συσχέτιση άρα δεν υπάρχει νόημα ύπαρξης της μετρικής αυτής στη συνάρτηση.

Η τιμή αποκοπής έχει οριστεί με βάση τη θεωρία στατιστικής και πειράματα που πραγματοποιήθηκαν. Η συσχέτιση Pearson (περισσότερα στο κεφάλαιο που αφορά τη στατιστική) είναι μια τιμή που κυμαίνεται από -1 έως 1 με τιμές από >0.5 να θεωρούνται επαρκώς συσχετισμένες. Η τιμή της αποκοπής είναι αρκετά σημαντική, διότι μια μεγάλη τιμή θα έχει σαν συνέπεια να μη μπορεί να δημιουργηθεί η συνάρτηση θεωρώντας ότι δεν υπάρχει επαρκής συσχέτιση με την εφαρμογή, ενώ μια μικρή τιμή θα δημιουργεί μια συνάρτηση με πολλές μεταβλητές οι οποίες όμως δεν θα χαρακτηρίζουν σωστά τα δεδομένα.

4. Για τις μετρικές που έχουν απομείνει από το βήμα 3, δημιουργούνται πίνακες συσχέτισης μεταξύ τους και για κάθε μια από αυτές. Αυτό είναι απαραίτητο ώστε να εντοπιστούν οι μετρικές που είναι συγγραμμικές μεταξύ τους. Όπως και στο βήμα 3 η τιμή που χρησιμοποιείται ως τιμή εντοπισμού συγγραμμικότητας² είναι 15. Η συνάρτηση που χρησιμοποιείται σε αυτή την περίπτωση είναι η VIF και προκύπτει από την θεωρία στατιστικής, με τιμή όπου εμφανίζεται η συγγραμμικότητα να είναι >10 .

5. Αφού πραγματοποιήθηκε το βήμα 4, πλέον έχουν επιλεγεί οι μετρικές που θα υπάρχουν στη συνάρτηση βαθμολόγησης. Σε αυτό το βήμα πρέπει να επιλεγθούν και οι κατάλληλοι συντελεστές για την κάθε μετρική-μεταβλητή.

² Περισσότερα για το θέμα της συγγραμμικότητας στο παράρτημα, ενότητα της στατιστικής

Για την επιλογή των συντελεστών πραγματοποιείται γραμμική παλινδρόμηση με πολλές μεταβλητές.

Αναλυτικά :

- a. Για κάθε πόρο δειγματοληψίας και για κάθε μετρική που έχει επιλεγθεί από το βήμα 4 φορτώνο και όλες οι τιμές που υπάρχουν στη βάση δεδομένων.
- b. Από το βήμα a και για κάθε πόρο επιλέγεται ο μέσος όρος αυτών.
- c. Για κάθε πόρο δειγματοληψίας φορτώνονται οι τιμές της εφαρμογής και επιλέγεται ο μέσος όρος αυτών.

Κάθε πόρος δειγματοληψίας με βάση τα a, b, c αποτελεί ένα σημείο που θα χρησιμοποιηθεί για να εφαρμοστεί η γραμμική παλινδρόμηση ώστε να επιλεγθούν οι σωστοί συντελεστές.

Σε περίπτωση που οι μετρικές που επιλεγθούν από το βήμα 4 είναι περισσότερες από τα σημεία που δημιουργούνται (δηλαδή από τους πόρους δειγματοληψίας) τότε το σύστημα αφαιρεί μετρικές έως ότου η γραμμική παλινδρόμηση μπορεί να δώσει λύση. Οι μετρικές που διαγράφονται είναι αυτές που έχουν τη μικρότερη συσχέτιση ως προς την εφαρμογή.

Στην παρακάτω εικόνα φαίνεται ένας πίνακας συσχέτισης μεταξύ μιας εφαρμογής και διαφόρων μετρικών. Για παράδειγμα γίνεται αντιληπτό ότι η εφαρμογή povray είναι υψηλά συσχετισμένη με τις μετρικές Mflops3 και Mflops4. Οι υπόλοιπες μετρικές θα αποκοπούν. Στη συνέχεια θα γίνει έλεγχος συγγραμμικότητας για τις δύο μετρικές που επιλέχθηκαν όπως περιγράφηκε από τον αλγόριθμο.

	povray								
dhry	0.97	dhry							
Mflops1	0.86	0.80	Mflops1						
Mflops2	0.94	0.88	0.98	Mflops2					
Mflops3	0.98	0.95	0.94	0.98	Mflops3				
Mflops4	0.98	0.98	0.87	0.93	0.98	Mflops4			
Copy	0.84	0.84	0.69	0.76	0.80	0.79	Copy		
Scale	0.86	0.85	0.71	0.78	0.81	0.80	1.00	Scale	
Add	0.84	0.83	0.69	0.76	0.79	0.77	0.99	1.00	Add
Triad	0.83	0.82	0.67	0.74	0.78	0.75	0.99	0.99	1.00
									Triad

Εικόνα 4.4 – Παράδειγμα πίνακα συσχέτισης (Πηγή εικόνας : Tsouloupas G, Dikaiakos MD [12])

Παράδειγμα προβλήματος

Έστω ότι ο χρήστης έχει επιλέξει ως πόρους δειγματοληψίας τους

ce101.grid.ucy.ac.cy,

ce-1-fzk.gridka.de,

...

Σε κάθε πόρο από αυτούς έχουν εκτελεστεί τα ίδια benchmarks διαφορετικό αριθμό επαναλήψεων και έχουν αποθηκευτεί οι τιμές για κάθε μετρική, πχ οι μετρικές Mflop1 και Copy (>0). Επίσης έχει εκτελεστεί η εφαρμογή P, με διαφορετικό αριθμό εκτελέσεων σε κάθε πόρο (>0).

Έστω ότι οι μετρικές που επιλέγονται με βάση τα βήματα 1, 2, 3, 4 είναι οι Mflop1 και Copy. Άρα έχουμε τη δημιουργηθείσα συνάρτηση στη μορφή $k \cdot \text{Mflop1} + t \cdot \text{Copy} + c$ (σταθερά) και θέλουμε να βρούμε τους συντελεστές και τη σταθερά c.

Για παράδειγμα ο πόρος *ce101.grid.ucy.ac.cy*, έχει X διαφορετικές τιμές για τη μετρική Mflop1, K για την Copy και B για την εφαρμογή P, ενώ ο πόρος *ce-1-fzk.gridka.de* Q, Z, και M.

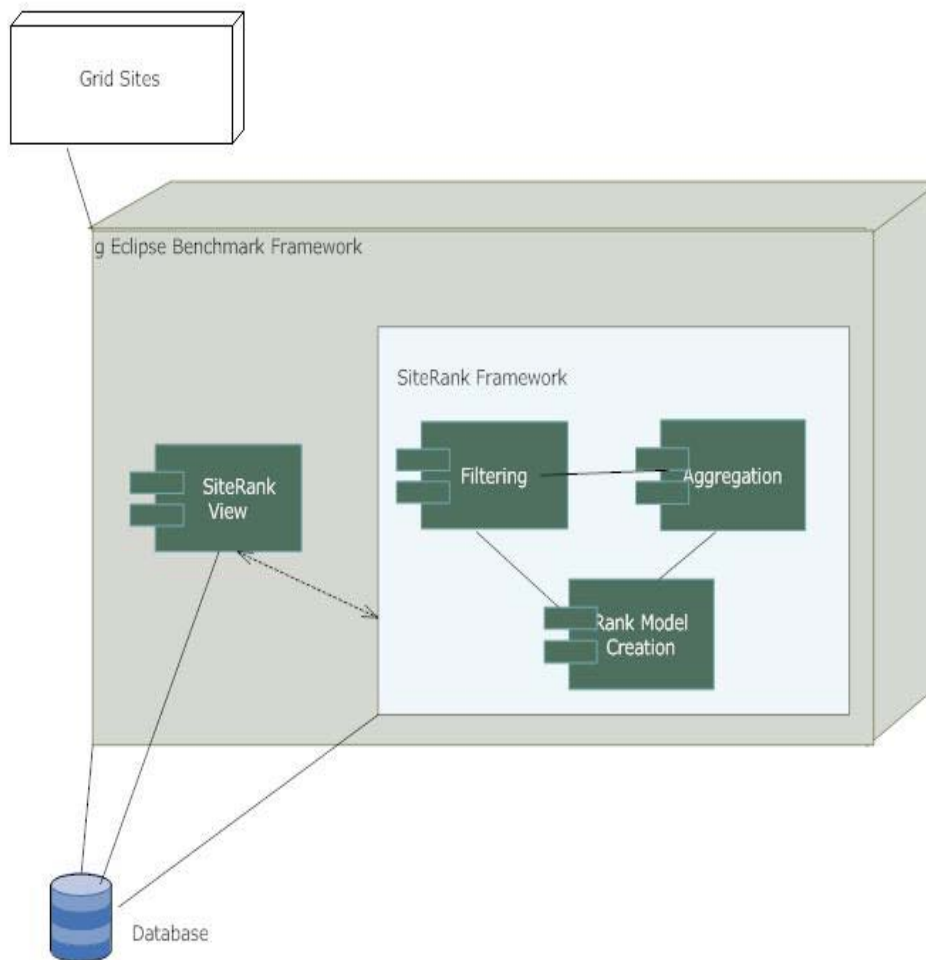
Προκειμένου να μπορεί να γίνει γραμμική παλινδρόμηση πρέπει να υπάρχει μία τιμή για κάθε μετρική και μία τιμή για την εφαρμογή, για αυτό και αποκτάται ο μέσος όρος των τιμών αυτών από κάθε πόρο. Στο συγκεκριμένο παράδειγμα έχουμε

Πόρος	Mflop1	Copy	Application
<i>ce101.grid.ucy.ac.cy</i> ,	<i>mean(X)</i>	<i>mean(K)</i>	<i>mean(B)</i>
<i>ce-1-fzk.gridka.de</i>	<i>mean(Q)</i>	<i>mean(Z)</i>	<i>mean(M)</i>
...

Έτσι μετά από αυτό κάθε πόρος αποτελεί ένα σημείο που θα χρησιμοποιηθεί στη γραμμική παλινδρόμηση όπου με τη χρήση των ελαχίστων τετραγώνων (περισσότερα στο παράρτημα στην ενότητα της στατιστικής) θα υπολογιστούν οι κατάλληλοι συντελεστές της συνάρτησης βαθμολόγησης ώστε να σχηματιστεί πλήρως αυτή. Στη συνέχεια η συνάρτηση βαθμολόγησης θα εφαρμοστεί στους πόρους που θέλει να αξιολογήσει ο χρήστης.

4.5 Αρχιτεκτονική συστήματος εντός του πλαισίου μέτρησης απόδοσης στο g Eclipse (benchmarking framework)

Στην ενότητα αυτή θα περιγραφεί η γενική αρχιτεκτονική του συστήματος εντός του πλαισίου μέτρησης επίδοσης στο g Eclipse καθώς επίσης και πώς αποκρίνεται το σύστημα κατά τη δημιουργία της συνάρτησης βαθμολόγησης.



Εικόνα 4.5 - Αρχιτεκτονική συστήματος

Όπως μπορούμε να παρατηρήσουμε στην προηγούμενη εικόνα, έχουν ενσωματωθεί εντός του πλαισίου μέτρησης επίδοσης του g Eclipse οι μονάδες αυτές που προσδίδουν την απαιτούμενη λειτουργικότητα προκειμένου να δημιουργηθεί η συνάρτηση βαθμολόγησης, να εφαρμοστεί στους πόρους και τέλος να απεικονιστούν τα αποτελέσματα με γραφικό τρόπο.

Επίσης πρέπει να επισημανθεί ότι η αρχιτεκτονική που χρησιμοποιήθηκε για τη δημιουργία του συστήματος επεκτείνει τη λειτουργικότητα του υπάρχοντος πλαισίου κρατώντας τη συμβατότητα και τα χαρακτηριστικά αυτού, αλλά ταυτόχρονα εξασφαλίζοντας και την αυτονομία του συστήματος ώστε να είναι εύκολη και άμεση κάθε αλλαγή που μπορεί μελλοντικά να απαιτηθεί.

4.5.1 SiteRank Framework (Filtering – Aggregation – Rank model)

Οι μονάδες αυτές περιέχουν όλη τη λειτουργικότητα του συστήματος. Κάθε μία αποτελεί ένα μοντέλο το οποίο καλείται από τη μονάδα SiteRank View όταν αυτό είναι απαραίτητο για την επίτευξη του ανάλογου στόχου. Κάθε μονάδα υλοποιεί κάποιο βήμα ή βήματα από αυτά που περιγράφηκαν αναλυτικά στην ενότητα «Αλγόριθμος δημιουργίας της συνάρτησης βαθμολόγησης».

4.5.2 SiteRank View

Η μονάδα αυτή περιλαμβάνει τα στοιχεία υλοποίησης που είναι απαραίτητα για την επικοινωνία χρήστη-συστήματος, προκειμένου να επιτευχθεί η δημιουργία της συνάρτησης βαθμολόγησης, καθώς και για τη γραφική απεικόνιση των αποτελεσμάτων. Αυτό πραγματοποιείται με τη χρήση ενός οδηγού (wizard) για τη δημιουργία της συνάρτησης βαθμολόγησης ενώ για τη γραφική απεικόνιση γίνεται χρήση μιας όψης (view) του g Eclipse. Η μονάδα αυτή ίσως είναι και η πιο σημαντική υπό την έννοια ότι αυτή αποτελεί την επαφή του χρήστη με το σύστημα επομένως πρέπει να έχει αυτά τα χαρακτηριστικά τα οποία θα εξασφαλίσουν την ευχρηστία στο χρήστη.

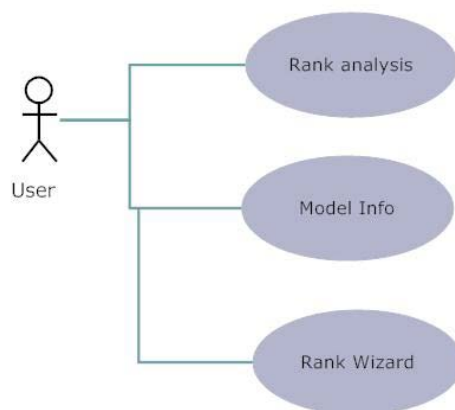
Τέλος σε κάθε περίπτωση το σύστημα οφείλει να ακολουθεί το πρότυπο λογισμικό Model-view-controller (MVC) [25]. Σύμφωνα με αυτό, η σχεδίαση χωρίζεται σε 3 κατηγορίες. Η πρώτη κατηγορία είναι το μοντέλο (model) στο οποίο περιλαμβάνονται τα δεδομένα και η λογική. Η δεύτερη κατηγορία είναι η προβολή (view) το οποίο περιλαμβάνει την παρουσίαση της εφαρμογής στον χρήστη. Τέλος έχουμε τον ελεγκτή (controller) ο οποίος ασχολείται με την επεξεργασία των δεδομένων που εισάγονται στο σύστημα και ουσιαστικά ενώνει τις δυο κατηγορίες που αναφέρθηκαν προηγουμένως.

4.5.2.1 Βήματα γραφικής απεικόνισης

Γενική ροή :

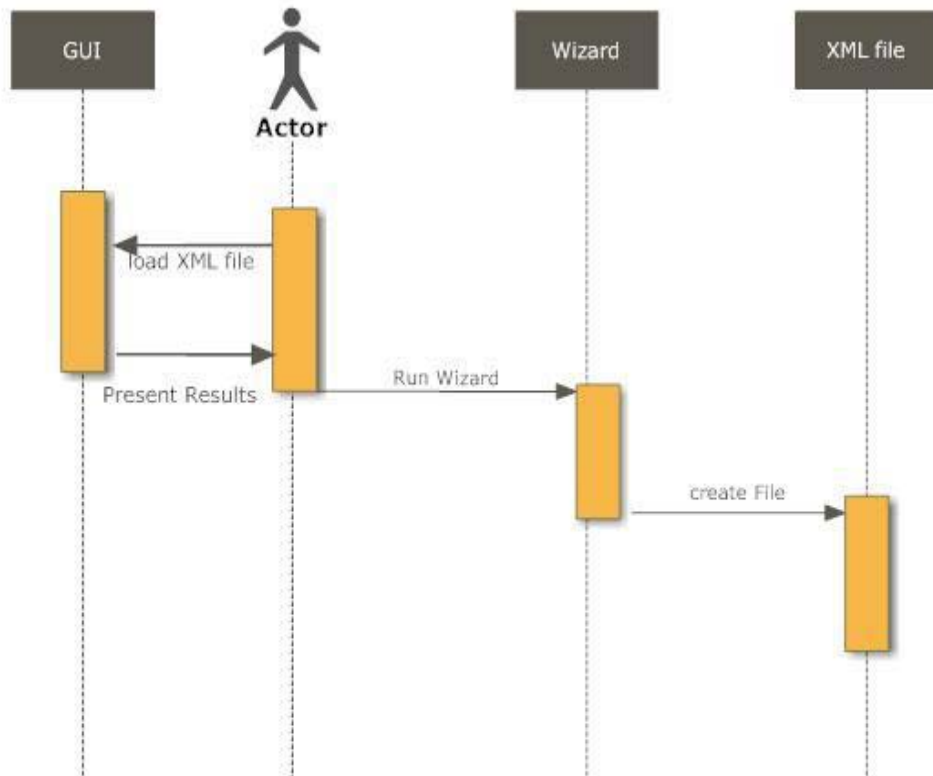
Όπως αναφέρθηκε ο χρήστης μέσω της υπάρχουσας όψης (αυτή που αναπτύχθηκε για το πλαίσιο εργασίας της μέτρησης επίδοσης στο g Eclipse) ενεργοποιεί έναν οδηγό με βάση τον οποίο θα δημιουργηθεί η συνάρτηση βαθμολόγησης. Αφού ολοκληρωθεί ο οδηγός αυτός με επιτυχία, δημιουργείται ένα αρχείο που είναι σε μορφή xml και το οποίο περιέχει πληροφορίες που αφορούν τη δημιουργηθείσα συνάρτηση καθώς και κάποιες επιπλέον πληροφορίες. Αυτό το αρχείο μπορεί να το επιλέξει ο χρήστης στη συνέχεια ώστε να γίνει η επεξεργασία του (parsing) και να απεικονιστούν τα αποτελέσματα ή να για να δοθούν επιπλέον πληροφορίες.

Γενικά οι διαθέσιμες επιλογές που έχει ο χρήστης στη διάθεση του φαίνονται στην εικόνα 4.6.



Εικόνα 4.6 – Δυνατές επιλογές του χρήστη

Στην παρακάτω εικόνα παρουσιάζονται σε μορφή διαγράμματος τα απαιτούμενα βήματα τα οποία και θα δειχθούν αναλυτικά στη συνέχεια, ξεκινώντας από την έναρξη του οδηγού (Wizard).

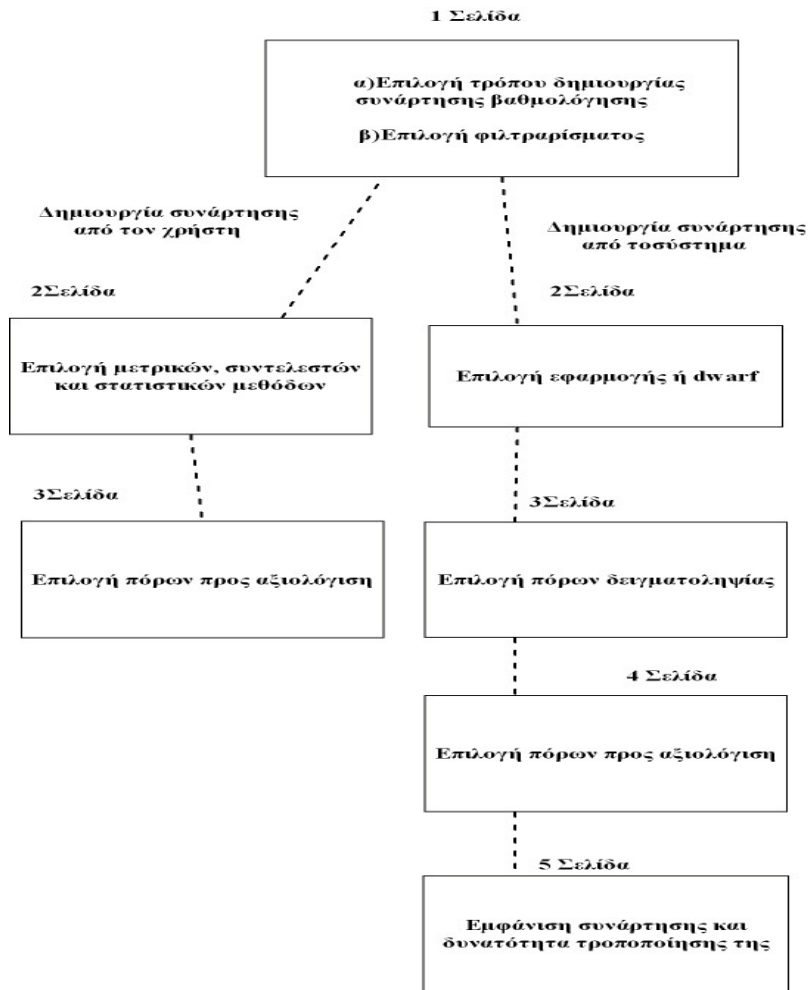


Εικόνα 4.7 – Βήματα του οδηγού

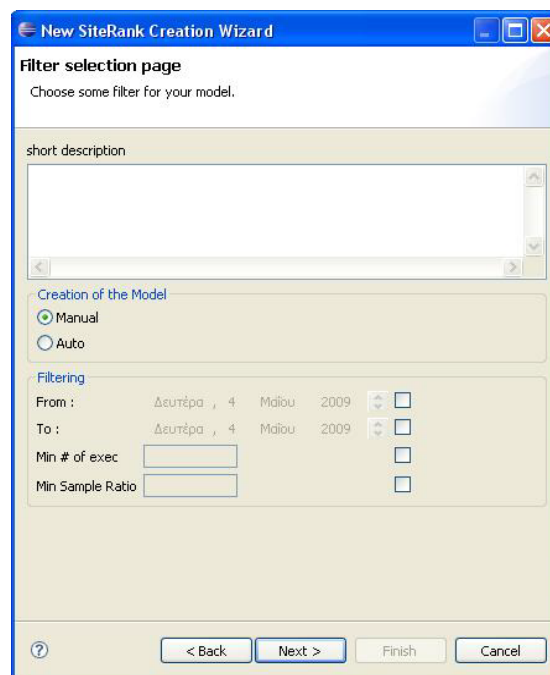
4.5.2.2 Οδηγός (wizard) δημιουργίας συνάρτησης βαθμολόγησης από το χρήστη και από το σύστημα

Στις παρακάτω εικόνες θα δειχθεί αναλυτικά η σχέση μεταξύ των διαφόρων σελίδων του οδηγού δημιουργίας της συνάρτησης βαθμολόγησης, καθώς και θα αναλυθούν τα χαρακτηριστικά της κάθε σελίδας δείχνοντας τη κάθε μία ξεχωριστά.

Στην εικόνα 4.8 φαίνεται με τη μορφή διαγράμματος η δομή των σελίδων του οδηγού.



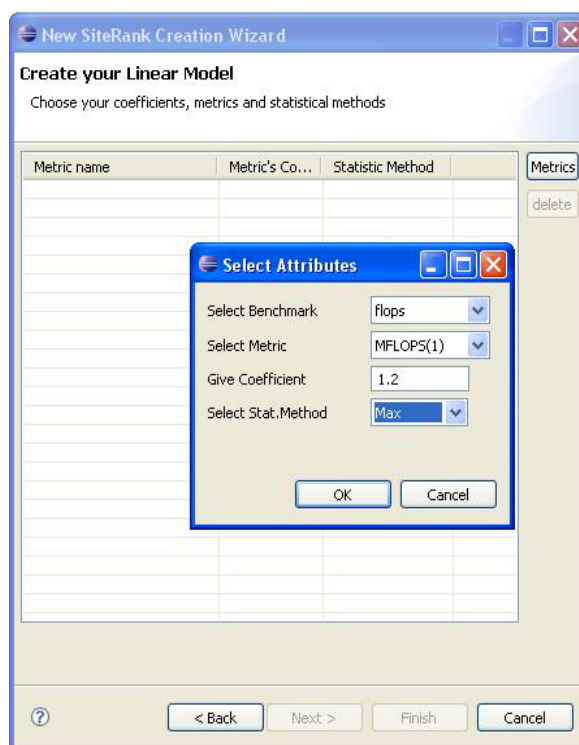
Εικόνα 4.8 - Δομή του οδηγού



Εικόνα 4.9 - Σελίδα πρώτη του οδηγού

Η παραπάνω εικόνα αποτελεί την πρώτη σελίδα που εμφανίζεται κατά την έναρξη του οδηγού. Ο χρήστης μπορεί να δώσει μια σύντομη περιγραφή που αφορά τη συνάρτηση βαθμολόγησης που θα δημιουργηθεί. Στη συνέχεια επιλέγει τον τρόπο δημιουργίας της συνάρτησης βαθμολόγησης. Στην περίπτωση που επιλέξει να δημιουργήσει μόνος του τη συνάρτηση, μπορεί να περιορίσει τις μετρικές που θέλει να έχει η συνάρτηση, με τη χρήση κάποιων φίλτρων όπως είναι η ημερομηνία, ελάχιστος αριθμός εκτελέσεων κτλ.

4.5.2.3 Περίπτωση δημιουργίας συνάρτησης αξιολόγησης από τον χρήστη

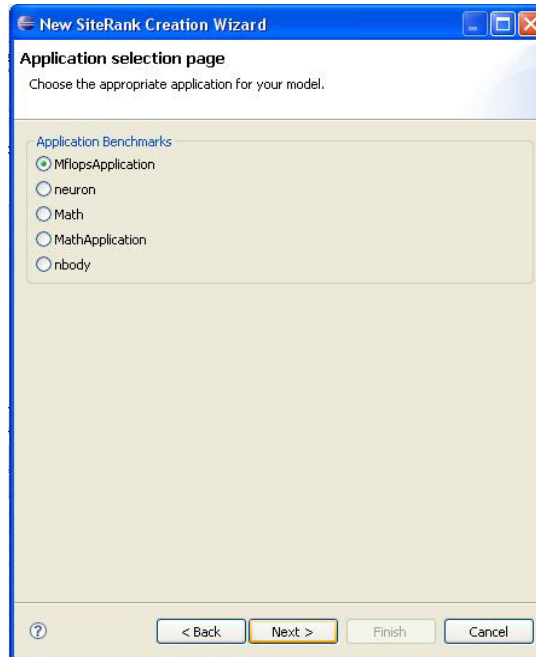


Εικόνα 4.10 - Σελίδα δημιουργίας συνάρτησης από τον χρήστη

Στη παραπάνω σελίδα ο χρήστης επιλέγει τις μετρικές τις οποίες επιθυμεί να συμπεριλάβει στη συνάρτησή του. Όπως φαίνεται και στην εικόνα 4.10 επιλέγει το κατάλληλο benchmark (το οποίο και φέρει τις μετρικές), τον συντελεστή και τη στατιστική μέθοδο.

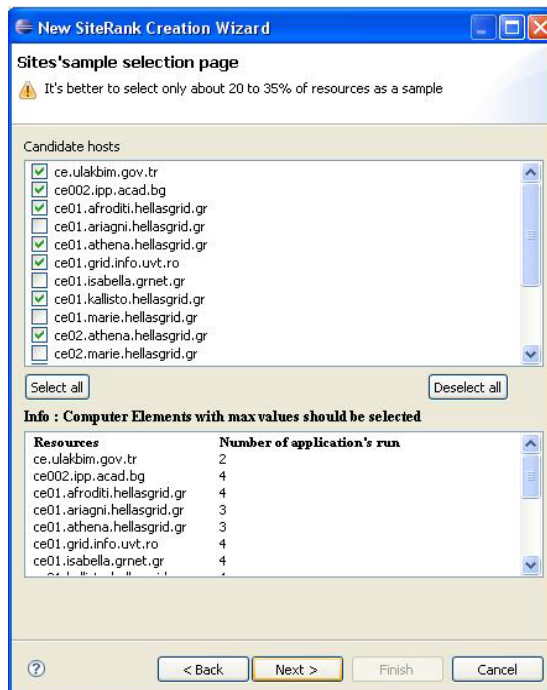
Στην παρακάτω σελίδα φαίνεται η συνάρτηση όπως δημιουργείται σταδιακά από τον χρήστη και έχει στην παρούσα φάση τη μορφή $R = 1.2 * MFLOPS(1) (Max) - 1.2 * random_seeks_speed(st\ dev)$.

4.5.2.4 Περίπτωση δημιουργίας συνάρτησης αξιολόγησης από σύστημα



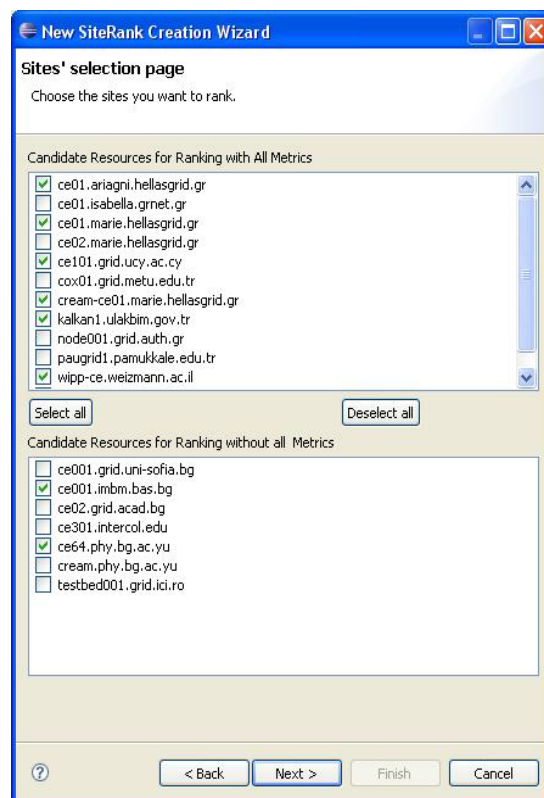
Εικόνα 4.13 - Σελίδα επιλογής εφαρμογής

Στη παραπάνω εικόνα εμφανίζονται δυναμικά όλες οι εφαρμογές ή τα dwarfs που υπάρχουν αποθηκευμένα στη βάση δεδομένων και οι οποίες έχουν εκτελεστεί σε έναν ή περισσότερους πόρους. Στην εικόνα 4.13 υπάρχουν εφαρμογές όπως η MathApplication, που σχετίζεται με μαθηματικούς υπολογισμούς, η nbody που σχετίζεται με nbody μεθόδους και τέλος μιας εφαρμογή που σχετίζεται με νευρωνικά δίκτυα. Σε περίπτωση που δεν υπάρχουν εφαρμογές διαθέσιμες ο χρήστης δεν μπορεί να συνεχίσει τη διαδικασία και εμφανίζεται κατάλληλο μήνυμα.



Εικόνα 4.14 - Σελίδα επιλογής πόρων δειγματοληψίας

Στην παραπάνω εικόνα εμφανίζονται όλοι οι πόροι οι οποίοι μπορούν να χρησιμοποιηθούν ως πόροι δειγματοληψίας όπως αυτοί περιγράφηκαν σε προηγούμενη ενότητα. Στους πόρους αυτούς έχει εκτελεστεί η εφαρμογή που επιλέχθηκε στην προηγούμενη σελίδα του οδηγού καθώς και έχουν επίσης εκτελεστεί όλα τα micro benchmarks (δηλαδή για όλους αυτούς τους πόρους είναι διαθέσιμες όλες οι μετρικές). Όποιος πόρος δεν ικανοποιεί αυτούς τους δύο περιορισμούς ταυτόχρονα, δεν εμφανίζεται στη λίστα. Στο κάτω μέρος της σελίδας εμφανίζονται οι πόροι και δίπλα πόσες φορές έχει εκτελεστεί η εφαρμογή σε αυτούς. Όπως είναι εύκολα κατανοητό πόροι που έχουν εκτελέσει πολλές φορές μια εφαρμογή έχουν περισσότερες πιθανότητες οι μετρήσεις να είναι περισσότερο αντιπροσωπευτικές. Επομένως η επιλογή ενός τέτοιου πόρου έχει περισσότερες πιθανότητες να αποτελεί ένα καλό δείγμα.

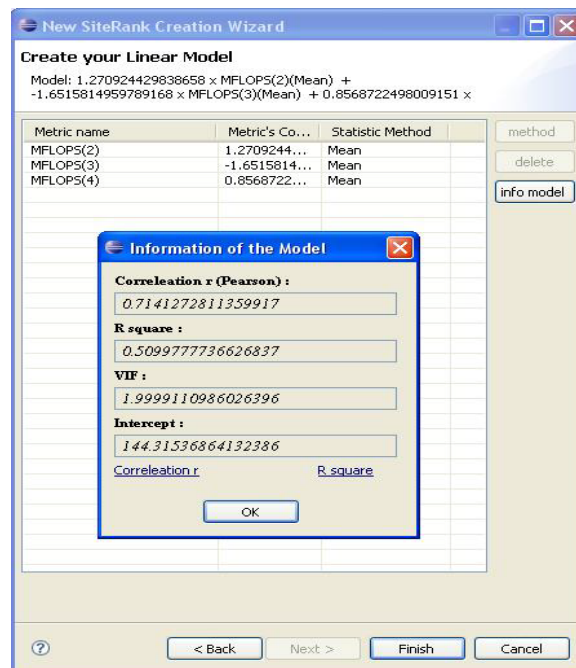


Εικόνα 4.15 - Σελίδα επιλογής πόρων για αξιολόγηση

Στην παραπάνω εικόνα παρουσιάζονται όλοι οι πόροι που είναι διαθέσιμοι από το σύστημα και ο χρήστης μπορεί να επιλέξει για να εφαρμοστεί η συνάρτηση βαθμολόγησης. Στο πάνω μέρος της σελίδας εμφανίζονται όλοι οι πόροι που έχουν όλες τις μετρικές, εκτός αυτών που επιλέχθηκαν ως πόροι δειγματοληψίας. Στο κάτω μέρος της σελίδας εμφανίζονται οι πόροι στους οποίους δεν έχουν εκτελεστεί όλα τα micro benchmarks άρα δεν υπάρχουν πληροφορίες για όλες τις μετρικές. Ο χρήστης μπορεί να επιλέξει και από τις δύο κατηγορίες τους πόρους τους οποίους επιθυμεί να αξιολογήσει την επίδοση της εφαρμογής.

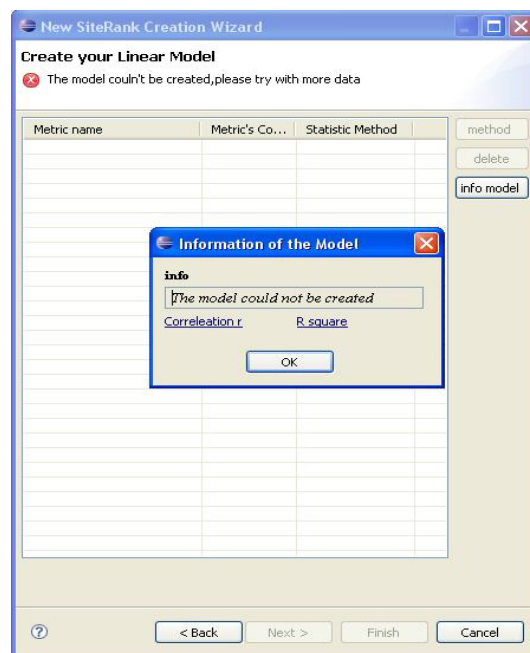
Στην παρακάτω εικόνα έχουμε την τελευταία σελίδα του οδηγού πριν τη δημιουργία του xml αρχείου. Σε αυτή εμφανίζεται η συνάρτηση που δημιουργήθηκε από το σύστημα. Ο χρήστης μπορεί να διαγράψει κάποιο όρο της συνάρτησης και να τροποποιήσει τη στατιστική μέθοδο η οποία είναι από το σύστημα πάντα ο μέσος όρος. Επιπλέον παρέχονται από το σύστημα και κάποιες πληροφορίες που αφορούν

τη δημιουργηθείσα συνάρτηση ώστε να μπορεί να κρίνει ο χρήστης πόσο καλή είναι αυτή. Περισσότερες πληροφορίες θα δοθούν στο παράρτημα στην ενότητα της στατιστικής.



Εικόνα 4.16 - Σελίδα εμφάνισης αποτελεσμάτων

Τέλος υπάρχει περίπτωση να μη μπορεί να δημιουργηθεί το μοντέλο. Αυτό οφείλεται στο γεγονός ότι με βάση τους πόρους δειγματοληψίας που επέλεξε ο χρήστης, οι τιμές των μετρικών αυτών να μην βρίσκουν κάποια συσχέτιση (Pearson correlation) με την εφαρμογή επομένως να απορρίπτονται όλες (ενότητα «Αλγόριθμος δημιουργίας της συνάρτησης βαθμολόγησης»). Σε αυτή την περίπτωση το σύστημα αποκρίνεται με κατάλληλο μήνυμα όπως φαίνεται παρακάτω.



Εικόνα 4.17 - Εμφάνιση λάθους

Για να μη συμβεί αυτή η περίπτωση υπάρχουν δύο λύσεις. Από την πλευρά του συστήματος να μειωθεί η τιμή αποκοπής ώστε να μην απορρίπτονται μετρικές που δεν έχουν μεγάλη συσχέτιση με την εφαρμογή. Αυτό όμως έχει ως συνέπεια, η συνάρτηση που δημιουργείται να μην δίνει καλά αποτελέσματα μη μπορώντας να χαρακτηρίσει επαρκώς τα στατιστικά δεδομένα. Αυτό που προτείνεται ως καλύτερη λύση είναι να επιλέγονται διαφορετικοί πόροι δειγματοληψίας από το χρήστη και να ελέγχεται η συνάρτηση για την ποιότητα της μέσω των πληροφοριών που παρέχει το σύστημα.

4.6 Δημιουργία XML αρχείου μετά την ολοκλήρωση του οδηγού

Όπως φαίνεται και από την εικόνα 4.7 επόμενο βήμα μετά την τελευταία σελίδα του οδηγού και την επιτυχή ολοκλήρωσή του, είναι η δημιουργία ενός xml αρχείου της μορφής X.rank, το οποίο περιέχει πληροφορίες σχετικά με τις επιλογές του χρήστη καθώς και για τη δημιουργηθείσα συνάρτηση βαθμολόγησης. Το αρχείο αυτό στη συνέχεια χρησιμοποιείται από το σύστημα προκειμένου να δημιουργήσει τη γραφική παράσταση καθώς και επίσης για να δίνει πληροφορίες (ως feedback) για τη συνάρτηση στο χρήστη.

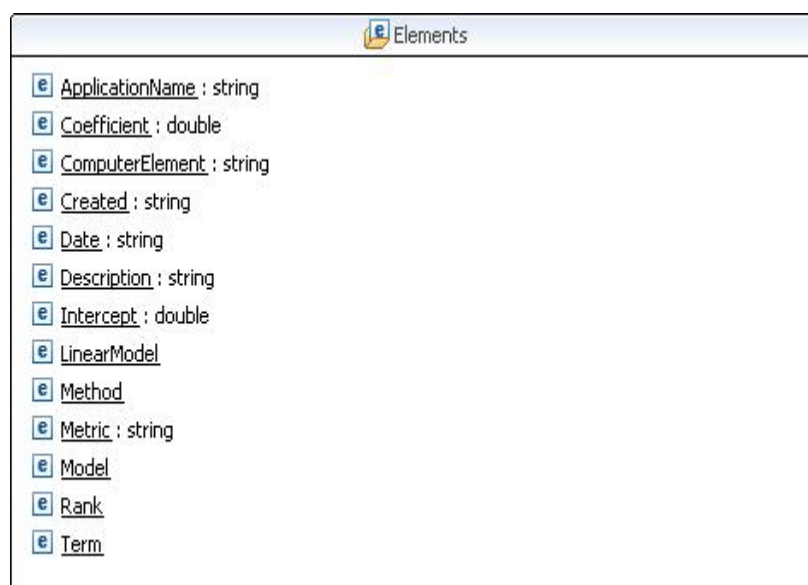
Το xml αρχείο είναι ουσιαστικά απαραίτητο για τους εξής λόγους

- Εύκολη μεταφορά από ένα εκτελέσιμο του g Eclipse σε ένα άλλο.
- Εξασφαλίζει τον πλήρη διαχωρισμό του μοντέλου από το γραφικό κομμάτι.
- Δυνατότητα δημιουργίας συντάκτη(editor) ώστε αλλαγές να γίνονται άμεσα και γρήγορα χωρίς να απαιτείται η εκτέλεση ξανά του οδηγού.

Το αρχείο αυτό σώζεται στον τρέχοντα κατάλογο που έχει επιλέξει ο χρήστης κατά την αρχική σελίδα του οδηγού με το όνομα που έδωσε.

4.6.1 Πληροφορίες σχετικά με την xml γλώσσα

Στην παρακάτω εικόνα φαίνονται τα στοιχεία της γλώσσας τα οποία και θα αναλυθούν παρακάτω.



Εικόνα 4.18 - Στοιχεία που αποτελούν το xml αρχείο

Στοιχεία που αφορούν τη συνάρτηση βαθμολόγησης :

- {Model} : Αυτό το στοιχείο περιγράφει τη συνάρτηση βαθμολόγησης. Κάθε μετρική περιγράφεται από το συντελεστή και τη στατιστική μέθοδο που θα χρησιμοποιηθεί. Τέλος υπάρχει και η σταθερά C.
- {Metric} : Στο στοιχείο αυτό γράφεται η κάθε μετρική της συνάρτησης βαθμολόγησης.
- {Method} : Στο στοιχείο αυτό γράφεται η στατιστική μέθοδος που επιλέγεται σε κάθε όρο της συνάρτησης βαθμολόγησης.
- {Coefficient} : Στο στοιχείο αυτό γράφεται ο συντελεστής από κάθε όρο της συνάρτησης βαθμολόγησης.
- {Intercept} : Στο στοιχείο αυτό γράφεται η σταθερά C, εάν υπάρχει από τη συνάρτηση βαθμολόγησης.

Στοιχεία που αφορούν «γενικές» πληροφορίες

- {ApplicationName} : Στο στοιχείο αυτό γράφεται το όνομα της εφαρμογής ή του dwarf που επέλεξε ο χρήστης.
- {ComputerElement} : Στο στοιχείο αυτό γράφεται ο πόρος που έχει επιλέξει ο χρήστης να εφαρμοστεί η συνάρτηση βαθμολόγησης.
- {Date} : Στο στοιχείο αυτό γράφεται η ημερομηνία που δημιουργήθηκε η συνάρτηση βαθμολόγησης.
- {Created} : Στο στοιχείο αυτό γράφεται εάν το μοντέλο δημιουργήθηκε από το χρήστη ή από το σύστημα αυτόματα.
- {Description} : Στο στοιχείο αυτό γράφεται τυχόν περιγραφή που έδωσε ο χρήστης για την συνάρτηση.

Ένα παράδειγμα της xml γλώσσας φαίνεται παρακάτω :

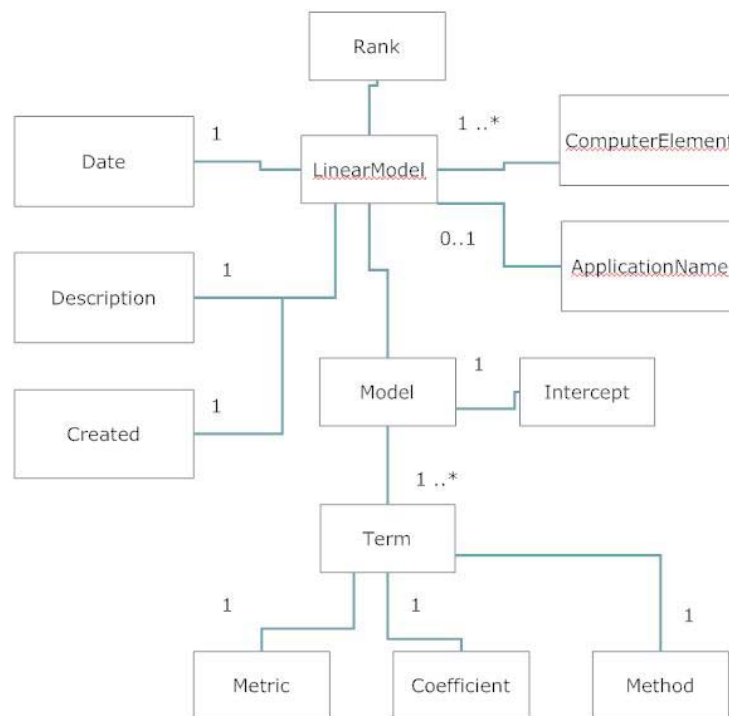
```
<?xml version="1.0" encoding="UTF-8"?>
<Rank xmlns:rank="http://www.eclipse.org/geclipse/rank">
<LinearModel>
  <Model>
    <Term>
      <Coefficient>1.270924429838658</Coefficient>
      <Metric>MFLOPS(2)</Metric>
      <Method>Mean</Method>
    </Term>
    <Term>
      <Coefficient>-1.6515814959789168</Coefficient>
      <Metric>MFLOPS(3)</Metric>
      <Method>Mean</Method>
    </Term>
  </Model>
</LinearModel>
</Rank>
```

```

</Term>
<Term>
  <Coefficient>0.8568722498009151</Coefficient>
  <Metric>MFLOPS(4)</Metric>
  <Method>Mean</Method>
</Term>
<Intercept>144.31536864132386</Intercept>
</Model>
<ComputerElement>ce001.ibm.bas.bg</ComputerElement>
<ComputerElement>ce301.intercol.edu</ComputerElement>
<Description>No description</Description>
<Date>Tue May 05 01:21:47 EEST 2009</Date>
<Created>Auto</Created>
<ApplicationName>MathApplication</ApplicationName>
</LinearModel>
</Rank>

```

Στην παρακάτω εικόνα φαίνεται η σχέση μεταξύ των στοιχείων της γλώσσας



Εικόνα 4.19 - Σχέση μεταξύ στοιχείων της xml γλώσσας

Το σύστημα για κάθε xml αρχείο που επιλέγει ο χρήστης ελέγχεται ως προς την ορθότητα της δομής της γλώσσας (όπως αυτή φαίνεται στην εικόνα 4.19) και αν αυτό είναι έγκυρο τότε θα εμφανίσει τα αποτελέσματα, αλλιώς θα εμφανίσει ένα μήνυμα λάθους. Το xml schema που χρησιμοποιείται για να ελεγχθεί η εγκυρότητα του xml αρχείου υπάρχει στο παράρτημα. Ο έλεγχος της εγκυρότητας πραγματοποιείται με τη χρήση ειδικών βιβλιοθηκών που προσφέρονται από την java (DOM, SAXP).

4.7 Σχεδίαση και ανάλυση της όψεως (view)

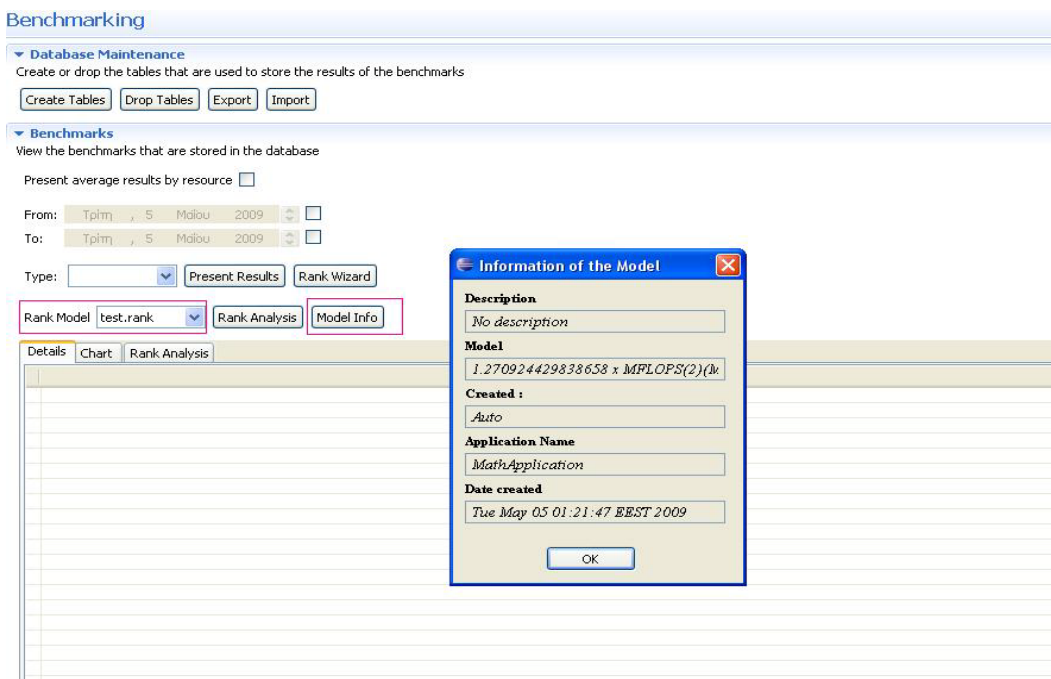
Όπως φαίνεται και στην εικόνα 4.7 στο τελευταίο βήμα έχουμε την επιλογή του αρχείου από τον χρήστη. Σε αυτό το σημείο υπάρχουν δύο διαθέσιμες επιλογές

- **Model info**

Παρουσιάζονται στο χρήστη πληροφορίες που αφορούν κυρίως τη συνάρτηση βαθμολόγησης και φαίνονται ποιες είναι παρακάτω

1. *Description* : Εμφανίζεται η περιγραφή που είχε δώσει ο χρήστης κατά τη δημιουργία της συνάρτησης βαθμολόγησης.
2. *Model* : Εμφανίζεται η συνάρτηση βαθμολόγησης .
3. *Created* : Αναφέρεται στο αν η συνάρτηση βαθμολόγησης δημιουργήθηκε από το χρήστη ή από το σύστημα.
4. *Application Name* : Αναφέρεται στην εφαρμογή ή στο dwarf που επέλεξε ο χρήστης.
5. *Date created* : Αναφέρεται στην ημερομηνία και ώρα που δημιουργήθηκε η συνάρτηση βαθμολόγησης.

Στην παρακάτω εικόνα εμφανίζονται οι πληροφορίες όπως αυτές παρουσιάζονται στο χρήστη.

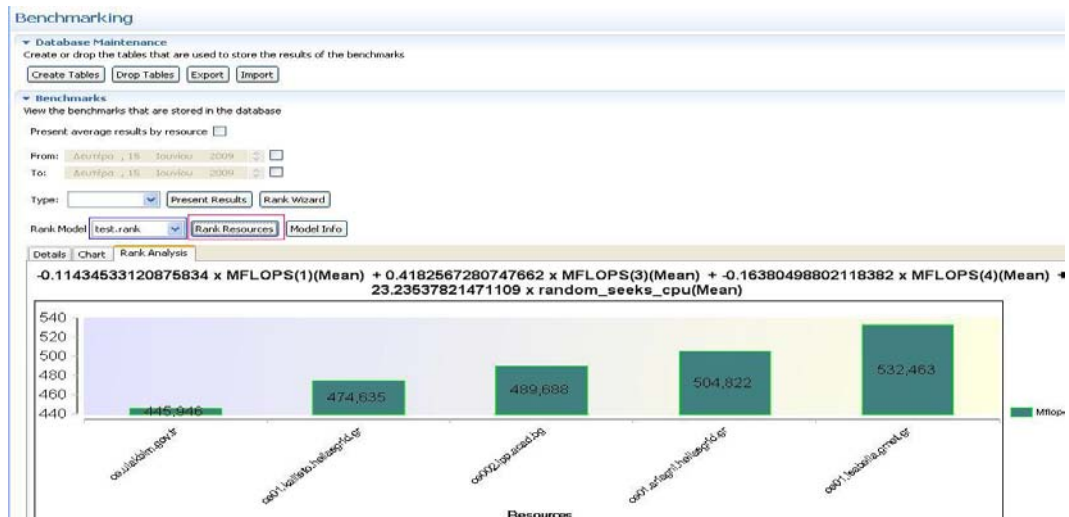


Εικόνα 4.20 - Διαθέσιμες πληροφορίες στο χρήστη

Στο πρώτο κόκκινο κουτί (αριστερά) φαίνεται το αρχείο που επέλεξε ο χρήστης και στο δεύτερο κουτί φαίνεται το κουμπί Model info που επιστρέφει τις πληροφορίες που φαίνονται στο κέντρο της εικόνας.

- **Rank Analysis**

Παρουσιάζεται η γραφική απεικόνιση με βάση τα δεδομένα που υπάρχουν στο αρχείο που επέλεξε ο χρήστης. Στην παρακάτω εικόνα εμφανίζεται η γραφική απεικόνιση.



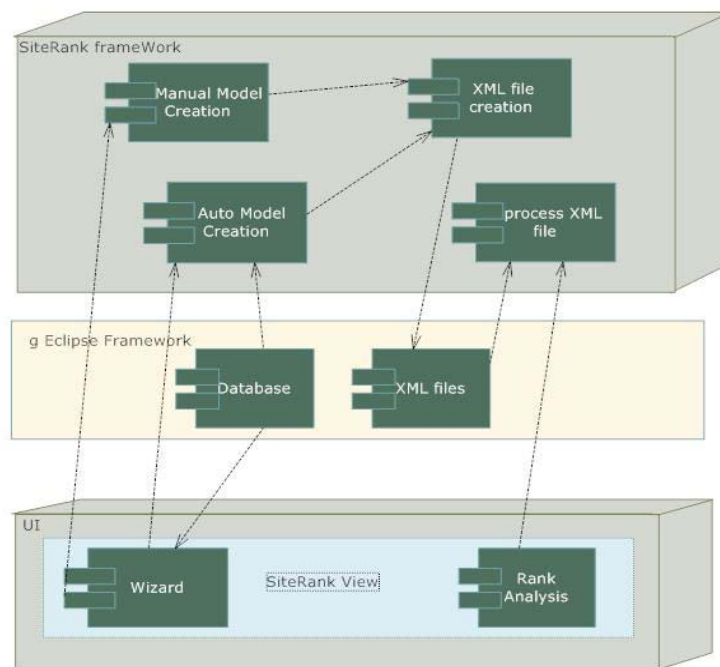
Εικόνα 4.21 - Γραφική απεικόνιση αποτελεσμάτων

Σε κόκκινο κουτάκι είναι το αρχείο που επέλεξε ο χρήστης καθώς και το κουμπί Rank Analysis που δημιουργεί τη γραφική παράσταση.

4.8 Υλοποίηση συστήματος

Στην ενότητα αυτή θα δοθούν με περισσότερη λεπτομέρεια στοιχεία που αφορούν στη σχεδίαση και τα τεχνικά χαρακτηριστικά του συστήματος. Κάθε μονάδα που η μορφή και λειτουργία περιγράφηκε σε προηγούμενες ενότητες θα εξεταστεί υπό το πρίσμα της σχεδίασης και ανάπτυξης λογισμικού.

Στην παρακάτω εικόνα φαίνονται τα συστατικά που αποτελούν το σύστημα.



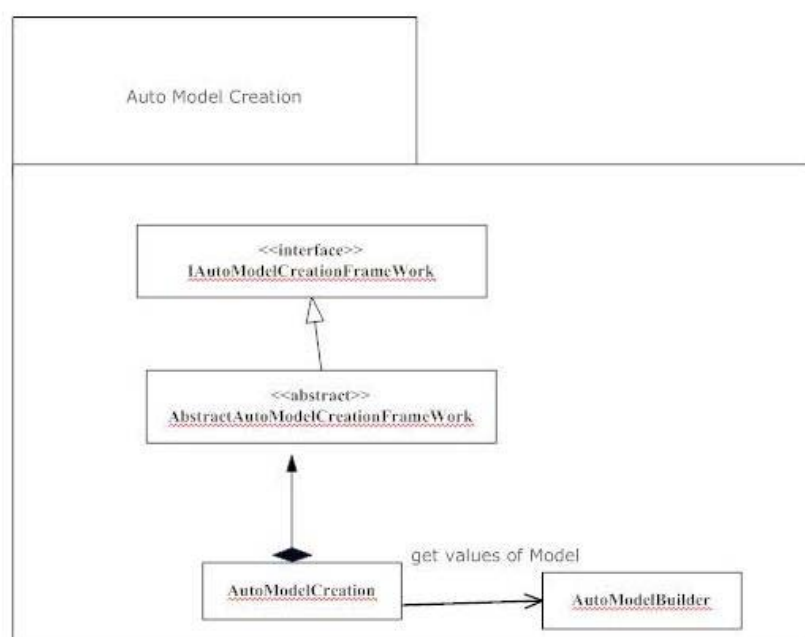
Εικόνα 4.22 - Συστατικά που αποτελούν το σύστημα SiteRank

Σύμφωνα με τη γενική αρχιτεκτονική που δόθηκε στην εικόνα 4.5 τα συστατικά Wizard και Rank Analysis ανήκουν στην μονάδα SiteRank View ενώ τα συστατικά Manual Model Creation, Auto Model Creation, XML file creation και process XML file ανήκουν στις μονάδες Filtering, aggregation και Rank Model.

4.8.1 Auto Model Creation

Το συστατικό αυτό ίσως είναι το πιο πολύπλοκο αλλά και το σημαντικότερο του συστήματος και αυτό διότι περιέχει τον αλγόριθμο για τη δημιουργία της συνάρτησης βαθμολόγησης.

Στην παρακάτω εικόνα φαίνονται οι κλάσεις που το υλοποιούν και η μεταξύ τους σχέση.



Εικόνα 4.23 - Σχέση κλάσεων

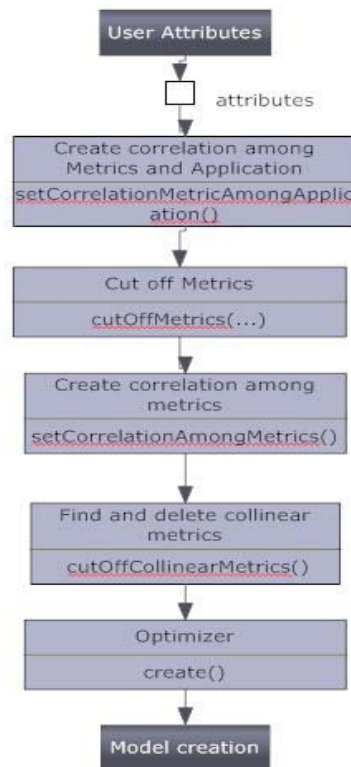
AbstractAutoModelCreationFrameWork: Η κλάση αυτή περιέχει την περισσότερη υλοποίηση που αφορά τη δημιουργία της συνάρτησης βαθμολόγησης. Περιλαμβάνει μεθόδους που είναι απαραίτητες για τη συσχέτιση των μετρικών με την εφαρμογή, των μετρικών μεταξύ τους, καθώς και για την αποκοπή μετρικών που είναι συγγραμμικοί ή δεν έχουν συσχέτιση με την εφαρμογή. Επίσης αυτή η κλάση επικοινωνεί με τη βάση δεδομένων προκειμένου να παίρνει τις τιμές για κάθε μετρική όταν χρειαστεί. Στην εικόνα 4.24 φαίνονται τα βήματα αυτά.

AutoModelCreation: Σκοπός αυτής της κλάσης είναι απλά η υλοποίηση κάποιων αφηρημένων (abstract) μεθόδων που έχουν δηλωθεί ως τέτοιες στην αφηρημένη κλάση. Οι μέθοδοι αυτοί σχετίζονται με το ποιό μαθηματικό μοντέλο θα χρησιμοποιεί σε κάθε φάση του αλγορίθμου. Η λογική αυτή χρησιμοποιήθηκε για δύο λόγους

1. Καλύτερος διαχωρισμός μεταξύ του αλγορίθμου και των μαθηματικών μοντέλων.

- Εύκολη αλλαγή των μαθηματικών μοντέλων χωρίς αλλοίωση του γενικού αλγορίθμου.

Για παράδειγμα μπορεί κάποιος να θέλει να αλλάξει τον τρόπο που υπολογίζεται η συσχέτιση (εδώ γίνεται χρήση του Pearson correlation) ή τον τρόπο που υπολογίζεται η γραμμική παλινδρόμηση. Το μόνο που έχει να κάνει είναι υπέρβαση των μεθόδων αυτών χωρίς να χρειάζεται να αλλάξει κάτι άλλο στην υλοποίηση του αλγορίθμου, ο οποίος προφανώς παραμένει ίδιος.

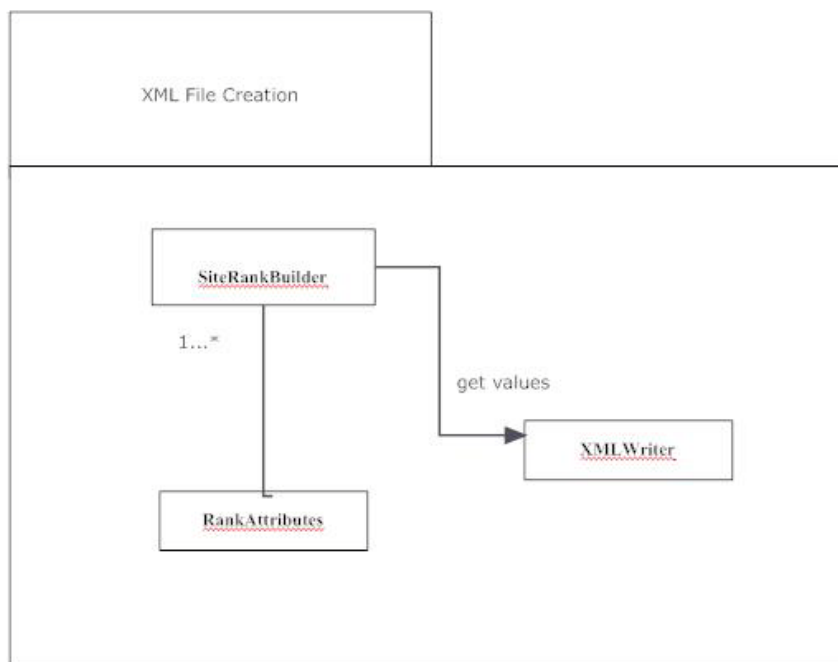


Εικόνα 4.24 - Βήματα υλοποίησης για δημιουργία συνάρτησης βαθμολόγησης

AutoModelBuilder: Σκοπός αυτής της κλάσης είναι να λάβει τα τελικά αποτελέσματα του αλγορίθμου δηλαδή τη συνάρτηση βαθμολόγησης.

4.8.2 XML file Creation

Το συστατικό αυτό είναι υπεύθυνο για τη δημιουργία των xml αρχείων. Η κύρια κλάση (SiteRankBuilder) δέχεται τη δημιουργηθήσα συνάρτηση βαθμολόγησης καθώς και κάθε επιπλέον πληροφορία και στη συνέχεια δημιουργείται το xml αρχείο (XMLWriter) για την εγκυρότητα του οποίου χρησιμοποιείται το κατάλληλο xml σχήμα. Στην παρακάτω εικόνα φαίνεται η σχέση μεταξύ των κλάσεων αυτών.



Εικόνα 4.25 - Σχέση κλάσεων

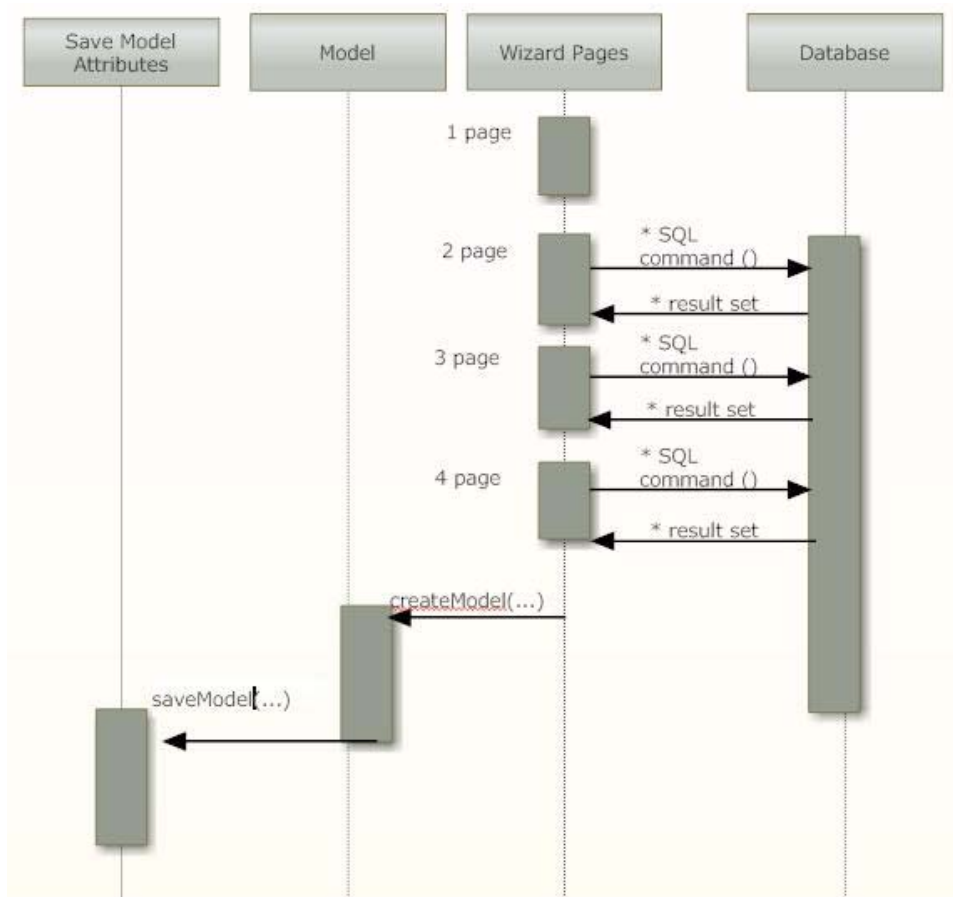
Η κλάση SiteRankBuilder αποτελεί μια κλάση επικάλυψης (wrapper class) υπό την έννοια ότι συγκεντρώνει πληροφορίες που προέρχονται από το λογικό κομμάτι του συστήματος (συνάρτηση βαθμολόγησης) αλλά και από το γραφικό κομμάτι (δηλαδή πληροφορίες που έδωσε ο χρήστης σε διάφορες σελίδες του οδηγού).

4.8.3 XML files

Αυτό το συστατικό περιλαμβάνει όλους τους απαραίτητους μηχανισμούς για την αποθήκευση και επεξεργασία αρχείων εντός του συστήματος αρχείων του Eclipse.

4.8.4 Wizard

Το συστατικό αυτό περιέχει όλες τις κλάσεις οι οποίες είναι απαραίτητες για τη γραφική απεικόνιση του οδηγού, η λειτουργικότητα του οποίου περιγράφηκε σε προηγούμενες ενότητες. Στην παρακάτω εικόνα παρουσιάζεται ο τρόπος που επικοινωνούν τα στοιχεία μεταξύ τους εντός του οδηγού.



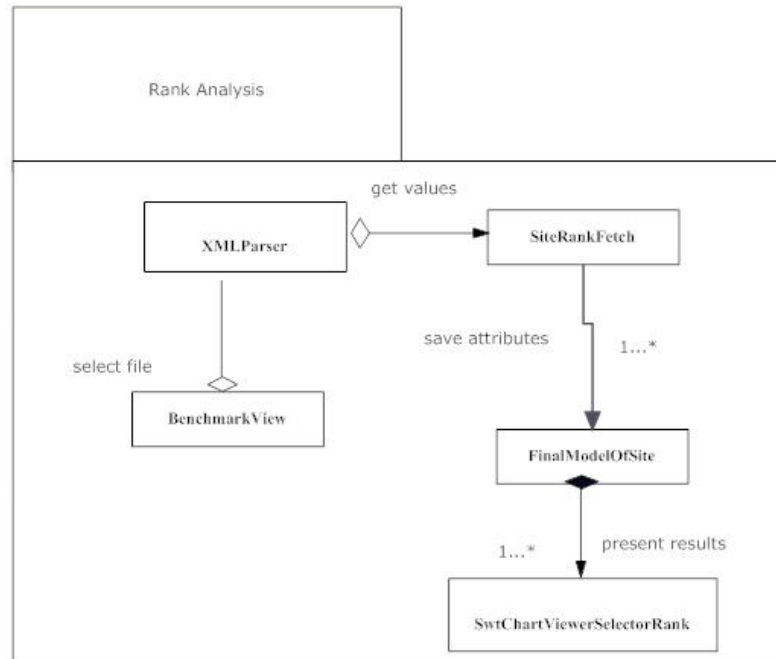
Εικόνα 4.26 - Βήματα υλοποίησης οδηγού

4.8.5 Process XML file

Το συστατικό αυτό αναπτύχθηκε ως ένα ξεχωριστό κομμάτι το οποίο χρησιμοποιείται προκειμένου να επεξεργαστεί το xml αρχείο (που επέλεξε ο χρήστης) και να πάρει τα δεδομένα του, ώστε να χρησιμοποιηθούν είτε στην γραφική απεικόνιση είτε προς πληροφόρηση του χρήστη για τη συνάρτηση βαθμολόγησης.

4.8.6 Rank Analysis

Το συστατικό αυτό περιλαμβάνει όλες τις κλάσεις που είναι απαραίτητες για την ορθή απεικόνιση των αποτελεσμάτων στην υπάρχουσα όψη της μέτρησης επίδοσης στο g Eclipse. Στην παρακάτω εικόνα παρουσιάζεται η σχέση των κλάσεων που είναι απαραίτητες για την απεικόνιση.



Εικόνα 4.27 - Σχέση κλάσεων

BenchmarkView: Αυτή η κλάση προϋπήρχε από το σύστημα μέτρησης επίδοσης στο g Eclipse. Αποτελεί την όψη που περιέχει όλες τις δυνατές ενέργειες που μπορεί να κάνει ο χρήστης (έναρξη οδηγού, εμφάνιση πινάκων με αποτελέσματα, απεικόνιση γραφικών παραστάσεων κτλ).

XMLParser: χρήση του συστατικού Process XML file.

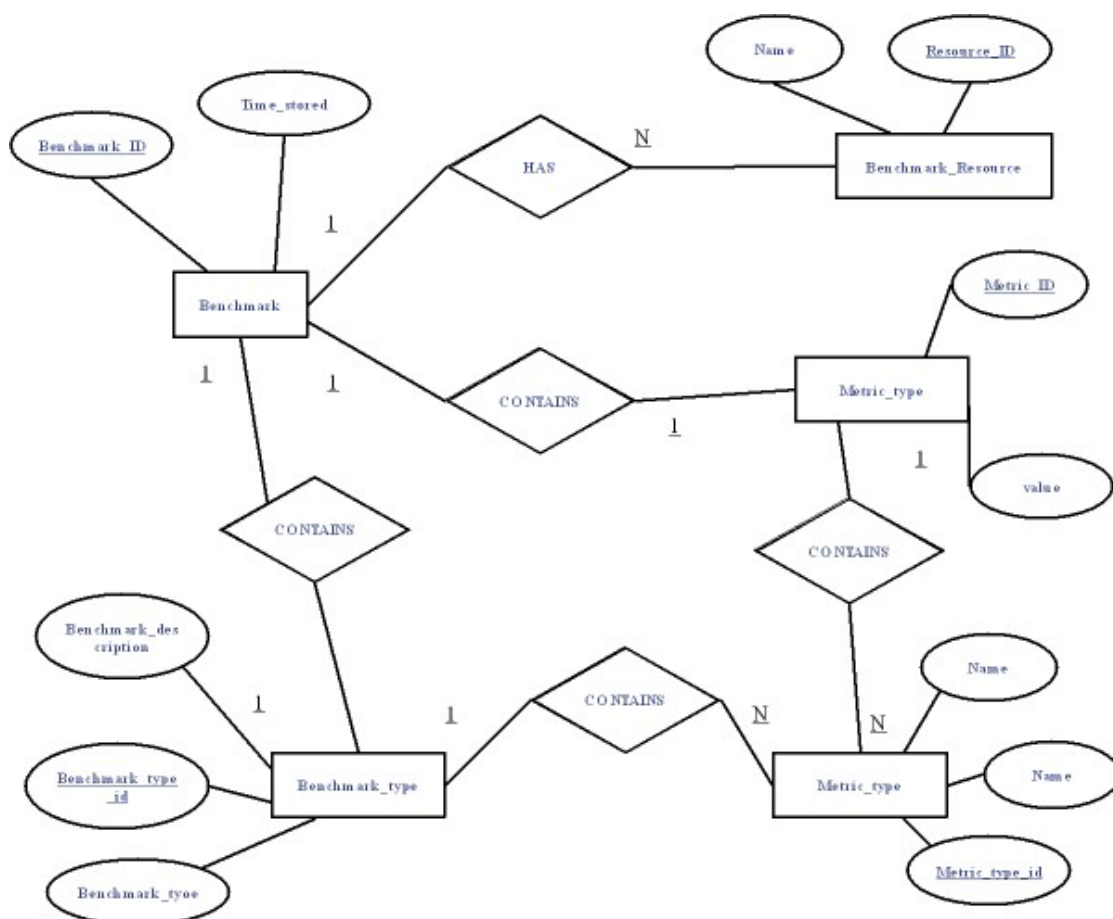
SiteRankFetch: Αποτελεί ένα νήμα (thread) το οποίο έχει σκοπό να μεταφέρει τις απαιτούμενες πληροφορίες στη κλάση SwtChartViewerSelectorRank η οποία και είναι υπεύθυνη να καλέσει το πακέτο λογισμικού για τη δημιουργία της γραφικής παράστασης.

FinalModelofSite: Η κλάση αυτή δημιουργεί ένα αντικείμενο για κάθε πόρο (που έχει επιλεγεί για μέτρηση από τον χρήστη) στον οποίο και εφαρμόζεται η συνάρτηση βαθμολόγησης. Οι πληροφορίες αυτές μεταφέρονται στην

SwtChartViewerSelectorRank: ώστε να γίνει η γραφική απεικόνιση

4.8.7 Database

Στην παρακάτω εικόνα παρουσιάζεται το σχήμα της βάσης δεδομένων που χρησιμοποιείται προκειμένου να αποθηκεύονται τα αποτελέσματα που επιστρέφονται από την εκτέλεση των benchmarks στους διάφορους πόρους του υπολογιστικού πλέγματος.



Εικόνα 4.28 - Σχήμα βάσης δεδομένων

Όλη η βάση μοντελοποιήθηκε με μια σειρά κλάσεων οι οποίες και αναλαμβάνουν την επικοινωνία με αυτή (σύνδεση, ερωτήσεις κτλ).

Καθένα από τα συστατικά που περιγράφηκαν περιέχεται σε διάφορα βύσματα/ plugins και καταλόγους με πληθώρα άλλων κλάσεων που είναι απαραίτητες για την ορθή λειτουργία. Η δομή των βυσμάτων θα δοθεί στο παράρτημα

Κεφάλαιο 5

Πειράματα

Στο κεφάλαιο αυτό θα δοθούν κάποια παραδείγματα από εφαρμογές (και dwarfs) που εκτελέστηκαν στο υπολογιστικό πλέγμα και με βάση τα αποτελέσματα που αποκτήθηκαν δημιουργήθηκε η συνάρτηση βαθμολόγησης η οποία χρησιμοποιήθηκε στη συνέχεια για τη γραφική απεικόνιση των αποτελεσμάτων.

5.1 Χρήση του dwarf SciMark

Το dwarf αυτό καλύπτει τη κατηγορία που σχετίζεται με τον υπολογισμό μαθηματικών πράξεων από ένα πόρο. Το dwarf επιστρέφει Mflops. Παραπάνω πληροφορίες για αυτό στο παράρτημα.

Αρχικά θα δοθεί μια λεπτομερής κατάσταση του περιβάλλοντος που πραγματοποιήθηκαν τα πειράματα ώστε να υπάρχει πλήρη γνώση του συστήματος τη χρονική αυτή στιγμή.

Micro Benchmarks :

Στο σύστημα έχουν εκτελεστεί τα εξής micro benchmarks με τις αντίστοιχες μετρικές που φέρουν :

Benchmark	Μετρικές
Bonnie	Input_block_cpu, random_seeks_speed, output_block_speed, Input_char_speed, random_seeks_cpu, output_rewrite_speed, Input_char_cpu, Input_block_speed, filezise, output_char_cpu, Output_rewrite_cpu, output_block_cpum, output_char_speed
Flops	MFLOPS(1), MFLOPS(2), MFLOPS(3), MFLOPS(4)
dhry	dhry
Stream	Copy, Scale, Add, Triad
memsize	Completion, max_mem

Πίνακας 1 - Micro Benchmarks

Με βάση αυτές τις μετρικές το σύστημα θα προσπαθήσει να δημιουργήσει τη συνάρτηση βαθμολόγησης

Προκειμένου να υπάρχουν αρκετοί διαθέσιμοι πόροι για τα πειράματα έγινε χρήση του εικονικού οργανισμού (Virtual Organization) SEE(South East Europe)[17]. Οι πόροι είναι οι εξής :

Όλοι οι πόροι του V.O

ce01.afroditi.hellasgrid.gr ce02.marie.hellasgrid.gr ce002.ipp.acad.bg kalkan1.ulakbim.gov.tr ce301.intercol.edu yildirim.grid.boun.edu.tr ce02.marie.hellasgrid.gr tbit01.nipne.ro cs-grid1bgu.ac.il grid01.erciyes.edu.tr ce101.grid.ucy.ac.cy ce01.kallisto.hellasgrid.gr wipp-ce.weizmann.ac.il ce01.isabella.grnet.gr grid001.ics.forth.gr CE.grid.openu.ac.il cream-ce01.marie.hellasgrid.gr	ce01.ariagni.hellasgrid.gr ce02.grid.acad.bg grid-ce.ii.edu.mk testbed001.grid.ici.ro ce01.grid.info.uvt.ro ce001.grid.uni-sofia.bg ce01.athena.hellasgrid.gr paugrid1.pamukkale.edu.tr cox01.grid.metu.edu.tr ituce.grid.itu.edu.tr ce.ulakbim.gov.tr ce64.phy.bg.ac.yu node001.grid.auth.gr ce01.marie.hellasgrid.gr ce-atlas.phy.bg.ac.yu ce02.athena.hellasgrid.gr
--	---

Πίνακας 2 – Πόροι του V.O SEE

Πόροι που έχουν εκτελέσει την εφαρμογή/dwarf SciMark

ce.ulakbim.gov.tr ce002.ipp.acad.bg ce01.afroditi.hellasgrid.gr ce01.ariagni.hellasgrid.gr ce02.marie.hellasgrid.gr kalkan1.ulakbim.gov.tr ce01.isabella.grnet.gr ce01.marie.hellasgrid.gr ce02.athena.hellasgrid.gr	ce101.grid.ucy.ac.cy paugrid1.pamukkale.edu.tr cream-ce01.marie.hellasgrid.gr cox01.grid.metu.edu.tr yildirim.grid.boun.edu.tr ce02.marie.hellasgrid.gr ce01.kallisto.hellasgrid.gr ce01.athena.hellasgrid.gr ce01.grid.info.uvt.ro
--	---

Πίνακας 3 - Πόροι που έχουν εκτελέσει το dwarf SciMark

Πόροι που έχουν όλες εκτελέσει όλα τα micro benchmarks

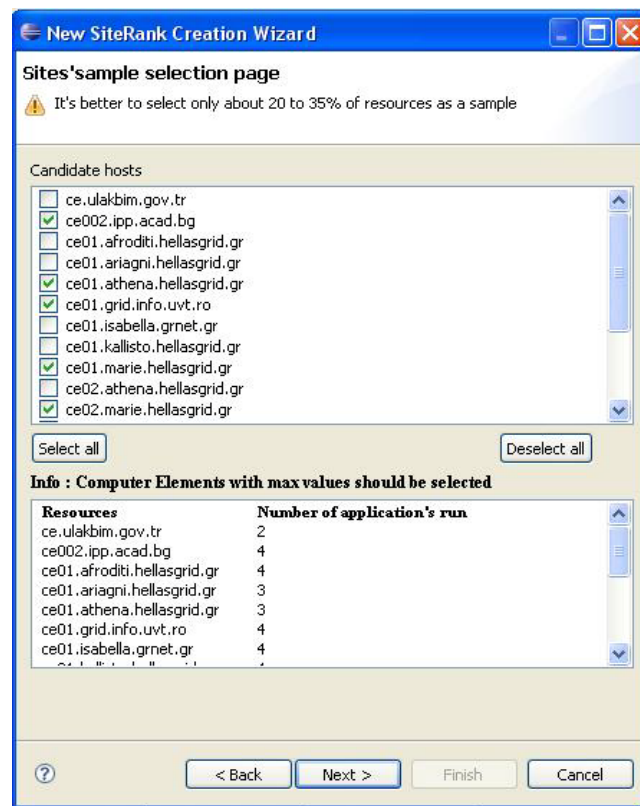
ce.ulakbim.gov.tr ce002.ipp.acad.bg ce01.afroditi.hellasgrid.gr ce01.ariagni.hellasgrid.gr ce02.marie.hellasgrid.gr kalkan1.ulakbim.gov.tr ce01.isabella.grnet.gr ce01.marie.hellasgrid.gr ce02.athena.hellasgrid.gr	ce01.grid.info.uvt.ro ce101.grid.ucy.ac.cy paugrid1.pamukkale.edu.tr cream-ce01.marie.hellasgrid.gr cox01.grid.metu.edu.tr yildirim.grid.boun.edu.tr ce02.marie.hellasgrid.gr ce01.kallisto.hellasgrid.gr ce01.athena.hellasgrid.gr
--	---

Πίνακας 4 – Πόροι που έχουν εκτελέσει όλα τα micro benchmarks

Η **μεθοδολογία** που ακολουθήθηκε στο πείραμα είναι η εξής :

Για τους πόρους που έχουν εκτελέσει το dwarf *SciMark* [22] το αποτέλεσμα του οποίου είναι διαθέσιμο (και γνωστό), θα χρησιμοποιηθεί ένα υποσύνολο ως σύνολο δειγματοληψίας (απαραίτητο για τη δημιουργία της συνάρτησης βαθμολόγησης) ενώ στους υπόλοιπους πόρους θα εφαρμοστεί η δημιουργηθείσα συνάρτηση βαθμολόγησης ώστε στο τέλος να συγκριθούν οι τιμές που θα προκύψουν ως προς τις πραγματικές τιμές.

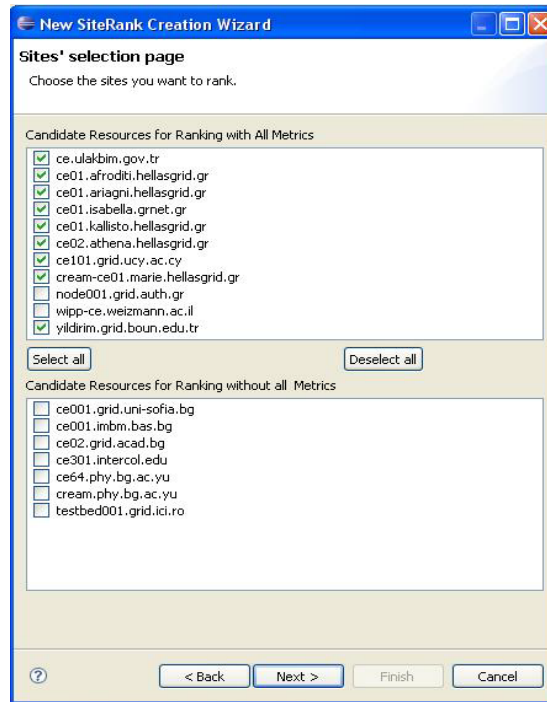
Στις παρακάτω εικόνες θα δειχτεί η δημιουργία της συνάρτησης βαθμολόγησης



Εικόνα 5.1 – Επιλογή πόρων δειγματοληψίας

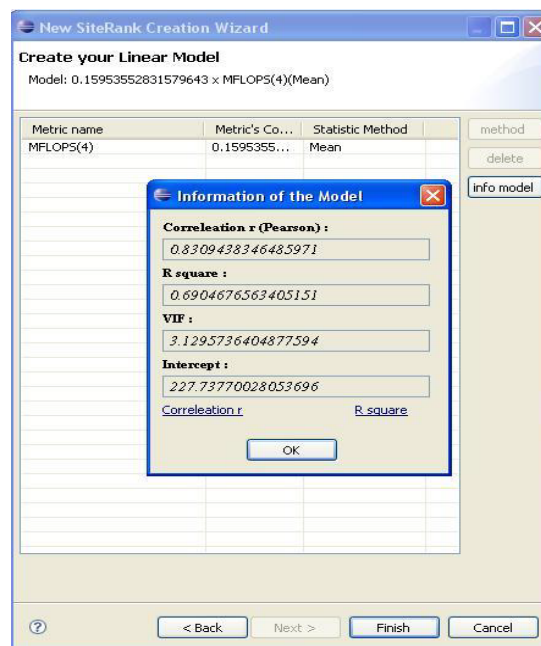
Σε αυτή τη σελίδα επιλέχθηκαν οι πόροι δειγματοληψίας. Ένα στοιχείο που μπορεί να βοηθήσει στην καλύτερη επιλογή των πόρων είναι ο αριθμός φορών που έχει εκτελεστεί η εφαρμογή σε κάθε έναν από αυτούς.

Στην παρακάτω εικόνα επιλέγονται οι πόροι προς εφαρμογή της συνάρτησης αξιολόγησης



Εικόνα 5.2 – Επιλογή πόρων για εφαρμογή της συνάρτησης αξιολόγησης

Τέλος στην παρακάτω εικόνα παρουσιάζεται την συνάρτηση που δημιουργήθηκε



Εικόνα 5.3 – Παρουσίαση νεοδημιουργηθείσας συνάρτησης βαθμολόγησης

Στον παρακάτω πίνακα παρουσιάζονται οι πόροι δειγματοληψίας καθώς και οι πόροι όπου θα εφαρμοστεί η συνάρτηση βαθμολόγησης.

Επιλεγθέντες πόροι Δειγματοληψίας Πόροι προς αξιολόγηση

ce002.ipp.acad.bg ce01.athena.hellasgrid.gr ce01.grid.info.uvt.ro ce01.marie.hellasgrid.gr ce02.marie.hellasgrid.gr cox01.grid.metu.edu.tr paugrid1.pamukkale.edu.tr ce02.marie.hellasgrid.gr	ce101.grid.ucy.ac.cy cream-ce01.marie.hellasgrid.gr yildirim.grid.boun.edu.tr ce01.kallisto.hellasgrid.gr ce01.afroditi.hellasgrid.gr ce01.ariagni.hellasgrid.gr ce01.isabella.grnet.gr ce02.athena.hellasgrid.gr
--	--

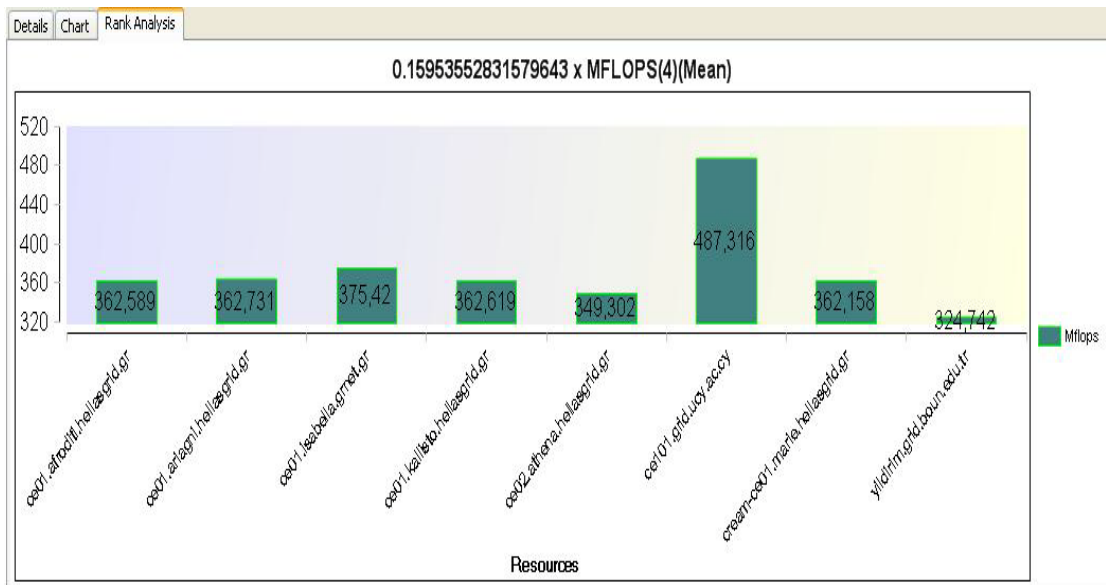
Πίνακας 5 - Πόροι για δειγματοληψία και αξιολόγηση αντίστοιχα

Παρακάτω παρουσιάζεται και το αρχείο .rank που δημιουργήθηκε :

```
<?xml version="1.0" encoding="UTF-8"?>
<Rank xmlns:rank="http://www.eclipse.org/geclipse/rank">
<LinearModel>
  <Model>
    <Term>
      <Coefficient>0.15953552831579643</Coefficient>
      <Metric>MFLOPS(4)</Metric>
      <Method>Mean</Method>
    </Term>
    <Intercept>227.73770028053696</Intercept>
  </Model>
  <ComputerElement>ce01.afroditi.hellasgrid.gr</ComputerElement>
  <ComputerElement>ce01.ariagni.hellasgrid.gr</ComputerElement>
  <ComputerElement>ce01.isabella.grnet.gr</ComputerElement>
  <ComputerElement>ce01.kallisto.hellasgrid.gr</ComputerElement>
  <ComputerElement>ce02.athena.hellasgrid.gr</ComputerElement>
  <ComputerElement>ce101.grid.ucy.ac.cy</ComputerElement>
  <ComputerElement>cream-ce01.marie.hellasgrid.gr</ComputerElement>
  <ComputerElement>yildirim.grid.boun.edu.tr</ComputerElement>
  <Description>No description</Description>
  <Date>Sat May 09 00:47:23 EEST 2009</Date>
  <Created>Auto</Created>
  <ApplicationName>MathApplication</ApplicationName>
</LinearModel>
</Rank>
```

Αποτελέσματα

Στην παρακάτω εικόνα φαίνονται τα αποτελέσματα όπως προκύπτουν από την χρήση της συνάρτησης βαθμολόγησης $0.15953552831579643 * MFLOPS(4)(Mean)$. Το αποτέλεσμα είναι σε *Mflops*.



Εικόνα 5.4 – Γραφική απεικόνιση αποτελεσμάτων

Επίσης παρουσιάζονται στην παρακάτω εικόνα τα πραγματικά αποτελέσματα της εκτέλεσης της εφαρμογής στους πόρους (οι πόροι σε κουτάκι)

Resource	Number of Benchmarks	Score
ce01.afroditi.hellasgrid.gr	4	364.607499...
ce01.grid.info.uvt.ro	4	441.6575
ce01.kallisto.hellasgrid.gr	4	365.21
ce301.intercol.edu	3	393.976666...
kalkan1.ulakbim.gov.tr	4	327.1575
ce101.grid.ucy.ac.cy	3	594.303333...
ce002.ipp.acad.bg	4	334.159999...
ce01.ariagni.hellasgrid.gr	3	365.57
ce.ulakbim.gov.tr	2	127.195
paugrid1.pamukkale.edu.tr	4	330.090000...
ce02.marie.hellasgrid.gr	4	352.6425
yildirim.grid.boun.edu.tr	3	330.78
cox01.grid.metu.edu.tr	3	331.92
ce02.athena.hellasgrid.gr	3	411.143333...
ce01.marie.hellasgrid.gr	4	365.205
ce01.isabella.grnet.gr	4	427.265
testbed001.grid.ici.ro	4	399.305
ce01.athena.hellasgrid.gr	3	363.766666...
cream-ce01.marie.hellasgrid.gr	3	364.926666...
ce001.ibm.bas.bg	2	395.515
ce001.grid.uni-sofia.bg	3	412.266666...
ce64.phy.bg.ac.yu	3	482.233333...
cream.phy.bq.ac.yu	1	484.49

Εικόνα 5.5 – Πραγματικές τιμές της εφαρμογής SciMark στους πόρους

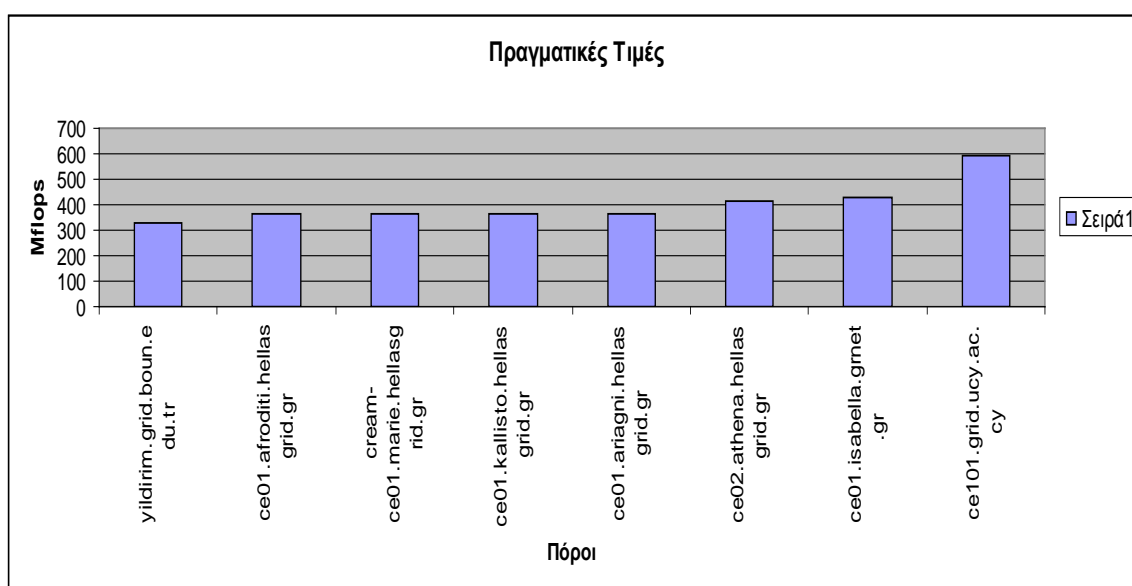
Τέλος στον παρακάτω πίνακα παρουσιάζονται τα πραγματικά αποτελέσματα από την εκτέλεση της εφαρμογής στους πόρους και δίπλα το αποτέλεσμα που προκύπτει από την εφαρμογή της συνάρτησης βαθμολόγησης (δηλαδή η εκτίμηση).

Πόροι	Πραγματική τιμή	Υπολογισθείσα τιμή
ce101.grid.ucy.ac.cy	594,303	487,386
cream-ce01.marie.hellasgrid.gr	364,92	362,158
yildirim.grid.boun.edu.tr	330,78	324,742
ce01.kallisto.hellasgrid.gr	365,21	362,619
ce01.afroditi.hellasgrid.gr	364,6	362,589
ce01.ariagni.hellasgrid.gr	365,57	362,731
ce01.isabella.grnet.gr	427,265	375,42
ce02.athena.hellasgrid.gr	411,144	349,302

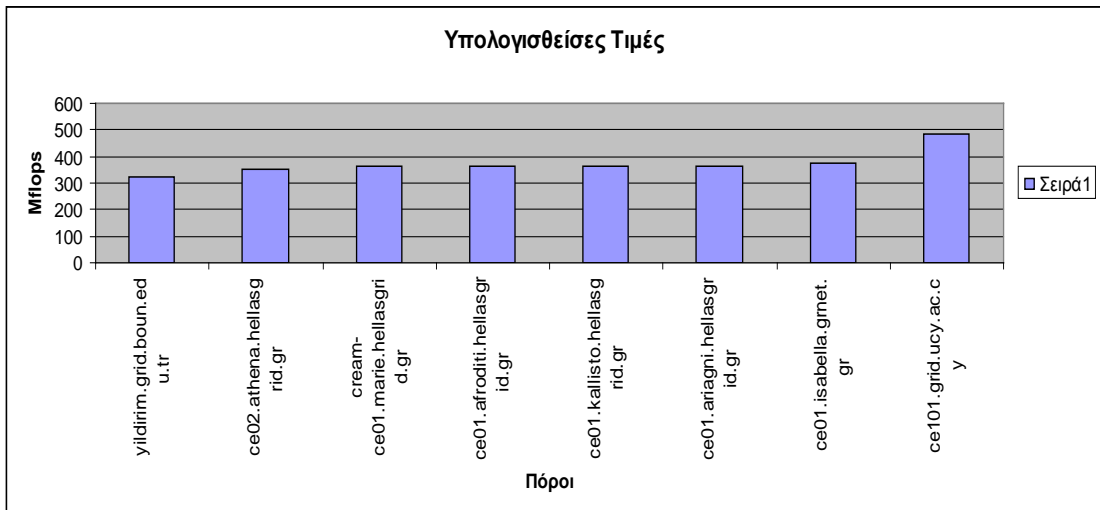
Πίνακας 6 – Πραγματικές και υπολογισθείσες τιμές

Για την αξιολόγηση των αποτελεσμάτων εφαρμόζεται Pearson Correlation και το οποίο επιστρέφει την τιμή **0,953273**. Επομένως η συσχέτιση της πραγματικής τιμής με την προκύπτουσα είναι πάρα πολύ μεγάλη. Επίσης γίνεται χρήση Kendall's Correlation η τιμή του οποίου είναι **0,619**. Η τιμή αυτή είναι χαμηλή αλλά οφείλεται στο ότι πολλά από τα αποτελέσματα είναι πολύ κοντά το ένα με το άλλο. Τέλος υπολογίζουμε και Spearman's Correlation με τιμή **0,85**.

Τέλος θα παρουσιαστούν άλλα δυο γραφήματα, το πρώτο παρουσιάζει τις πραγματικές τιμές με αύξουσα σειρά ενώ το δεύτερο τις υπολογισθείσες τιμές πάλι με αύξουσα σειρά. Όπως εύκολα μπορεί να διαπιστωθεί υπάρχουν μόνο δυο εναλλαγές στα γραφήματα και αυτό οφείλεται στο ότι τα αποτελέσματα είναι πολύ κοντά το ένα με το άλλο.



Εικόνα 5.6 – Γραφική απεικόνιση πραγματικών τιμών των πόρων



Εικόνα 5.7 – Γραφική απεικόνιση υπολογισθέντων τιμών των πόρων

Τέλος θα δειχθεί η σημασία που έχει να αποφεύγονται πόροι που έχουν ακραίες τιμές ή όταν δεν έχουν εκτελεστεί αρκετές φορές οι εφαρμογές. Θα υπολογίσουμε τη τιμή του πόρου ce.ulakbim.gov.tr (με πραγματική τιμή 127,195) η οποία και είναι 352,12. Σε αυτή την περίπτωση η συσχέτιση γίνεται 0,725715 δηλαδή όχι ιδιαίτερα καλή. Αν προσέξουμε όμως την εικόνα 5.5 θα δούμε ότι ο πόρος αυτός έχει εκτελεστεί δύο μόνο φορές και ότι η πραγματική του τιμή είναι αρκετά διαφορετική σε σχέση με τις άλλες (και ακραία τιμή και έχει εκτελεστεί λίγες φορές).

5.2 Χρήση του dwarf nbody

Micro Benchmarks: Όπως το προηγούμενο πείραμα.

Όλοι οι πόροι : Όπως στο προηγούμενο πείραμα.

Εικονικός Οργανισμός : SEE

Πόροι που έχουν εκτελέσει το dwarf nbody [24]

ce.ulakbim.gov.tr	ce101.grid.ucy.ac.cy
ce002.ipp.acad.bg	paugrid1.pamukkale.edu.tr
ce01.afroditi.hellasgrid.gr	cream-ce01.marie.hellasgrid.gr
ce01.ariagni.hellasgrid.gr	cox01.grid.metu.edu.tr
ce02.marie.hellasgrid.gr	yildirim.grid.boun.edu.tr
kalkan1.ulakbim.gov.tr	ce02.marie.hellasgrid.gr
ce01.isabella.grnet.gr	ce01.kallisto.hellasgrid.gr
ce01.marie.hellasgrid.gr	ce01.athena.hellasgrid.gr
ce02.athena.hellasgrid.gr	ce01.grid.info.uvt.ro

Πίνακας 7 - Πόροι που έχουν εκτελέσει το dwarf nbody

Πόροι που έχουν όλες εκτελέσει όλα τα micro benchmarks

ce.ulakbim.gov.tr ce002.ipp.acad.bg ce01.afroditi.hellasgrid.gr ce01.ariagni.hellasgrid.gr ce02.marie.hellasgrid.gr kalkan1.ulakbim.gov.tr ce01.isabella.grnet.gr ce01.marie.hellasgrid.gr ce02.athena.hellasgrid.gr	ce01.grid.info.uvt.ro ce101.grid.ucy.ac.cy paugrid1.pamukkale.edu.tr cream-ce01.marie.hellasgrid.gr cox01.grid.metu.edu.tr yildirim.grid.boun.edu.tr ce02.marie.hellasgrid.gr ce01.kallisto.hellasgrid.gr ce01.athena.hellasgrid.gr
--	---

Πίνακας 8 – Πόροι που έχουν εκτελέσει όλα τα micro benchmarks

Η μεθοδολογία : Όπως στο προηγούμενο πείραμα

Επιλεχθέντες πόροι Δειγματοληψίας Πόροι προς αξιολόγηση

ce002.ipp.acad.bg ce01.athena.hellasgrid.gr ce01.grid.info.uvt.ro ce01.marie.hellasgrid.gr ce02.marie.hellasgrid.gr cox01.grid.metu.edu.tr paugrid1.pamukkale.edu.tr ce02.marie.hellasgrid.gr	cream-ce01.marie.hellasgrid.gr yildirim.grid.boun.edu.tr ce01.kallisto.hellasgrid.gr ce01.afroditi.hellasgrid.gr ce01.ariagni.hellasgrid.gr ce01.isabella.grnet.gr
--	---

Πίνακας 9 - Πόροι για δειγματοληψία και αξιολόγηση αντίστοιχα

Παρακάτω παρουσιάζεται και το αρχείο .rank που δημιουργήθηκε

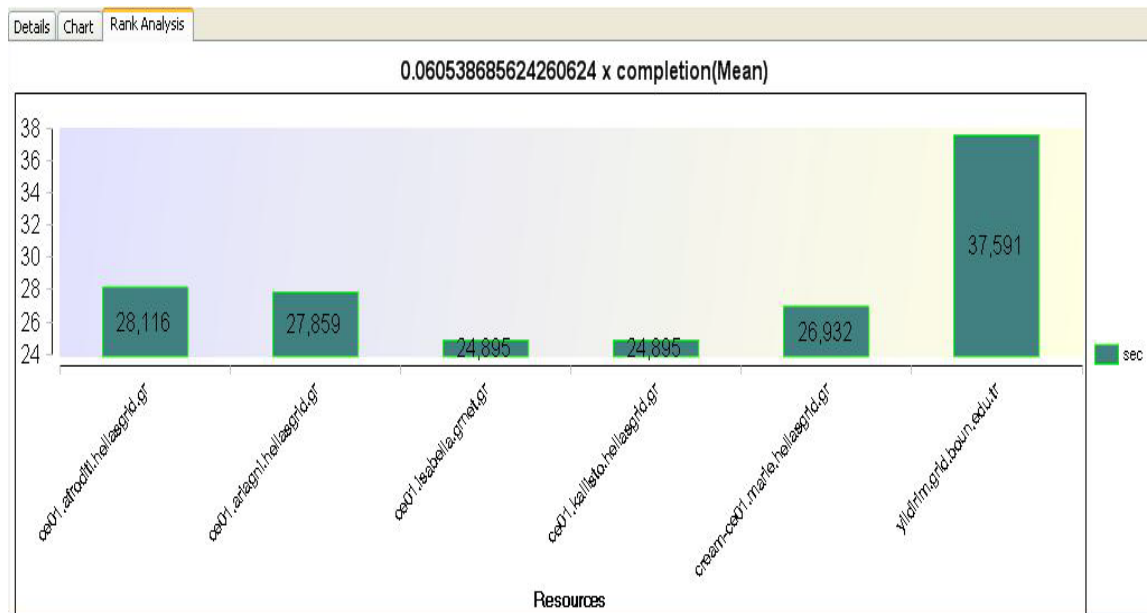
```
<?xml version="1.0" encoding="UTF-8"?>
<Rank xmlns:rank="http://www.eclipse.org/geclipse/rank">
<LinearModel>
  <Model>
    <Term>
      <Coefficient>0.060538685624260624</Coefficient>
      <Metric>completion</Metric>
      <Method>Mean</Method>
    </Term>
    <Intercept>24.894981344156676</Intercept>
  </Model>
  <ComputerElement>ce01.afroditi.hellasgrid.gr</ComputerElement>
  <ComputerElement>ce01.ariagni.hellasgrid.gr</ComputerElement>
  <Description>No description</Description>
  <Date>Sat May 09 20:01:57 EEST 2009</Date>
  <Created>Auto</Created>
  <ApplicationName>nbody</ApplicationName>
</Rank>
```

</LinearModel>

</Rank>

Αποτελέσματα

Στην παρακάτω εικόνα φαίνονται τα αποτελέσματα όπως προκύπτουν από τη χρήση της συνάρτησης βαθμολόγησης $0.060538685624260624 * \text{completion (Mean)}$. Το αποτέλεσμα είναι σε sec.



Εικόνα 5.8 – Γραφική απεικόνιση αποτελεσμάτων

Επίσης παρουσιάζονται στην παρακάτω εικόνα τα πραγματικά αποτελέσματα της εκτέλεσης της εφαρμογής στους πόρους (οι πόροι σε κόκκινο κουτάκι)

Resource	Number of Benchmarks	time
ce01.afroditι.hellasgrid.gr	3	29.683333...
node001.grid.auth.gr	4	22.507499...
ce01.grid.info.uvt.ro	4	30.907500...
ce01.kallisto.hellasgrid.gr	2	29.615000...
ce301.intercol.edu	4	23.6925
kalkan1.ulakbim.gov.tr	4	37.870000...
ce101.grid.ucy.ac.cy	3	19.083333...
ce002.ipp.acad.bg	4	25.695
ce01.ariagni.hellasgrid.gr	4	29.6475
ce.ulakbim.gov.tr	4	85.6
paugrid1.pamukkale.edu.tr	4	37.752500...
ce02.marie.hellasgrid.gr	4	25.709999...
yildirim.grid.boun.edu.tr	4	37.695
cox01.grid.metu.edu.tr	4	42.6425
ce02.athena.hellasgrid.gr	4	25.145
ce01.marie.hellasgrid.gr	4	29.6575
ce01.isabella.grnet.gr	4	22.904999...
testbed001.grid.ici.ro	4	26.185000...
ce01.athena.hellasgrid.gr	3	29.716666...
cream-ce01.marie.hellasgrid.gr	1	29.72
ce001.ibm.bas.bg	2	29.020000...
ce001.grid.uni-sofia.bg	4	24.4375

Εικόνα 5.9 – Πραγματικές τιμές της εφαρμογής nbody στους πόρους

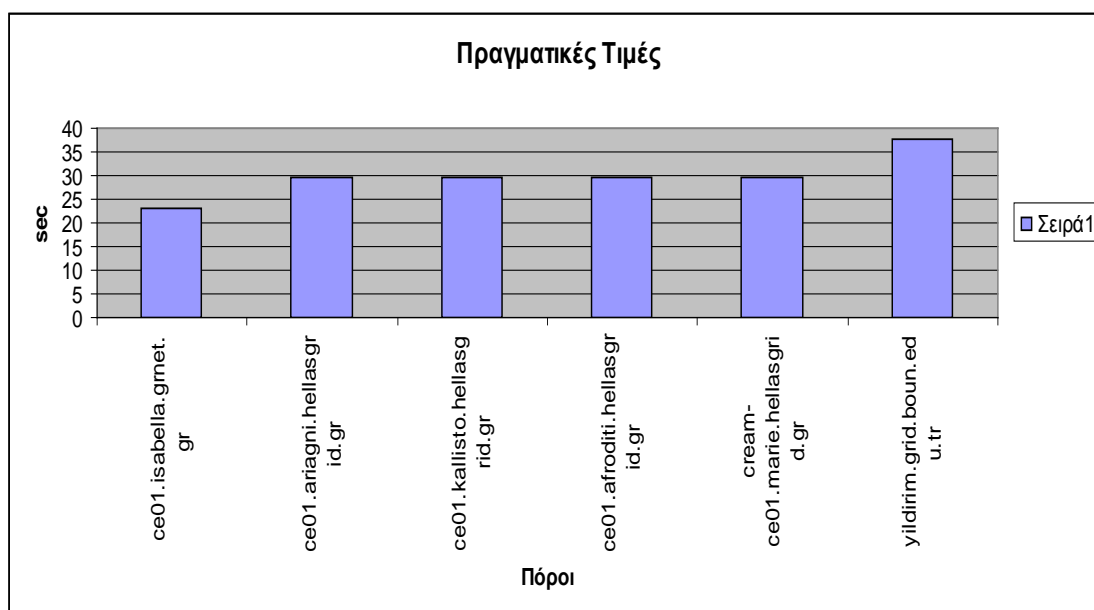
Τέλος στον παρακάτω πίνακα παρουσιάζονται τα πραγματικά αποτελέσματα από την εκτέλεση της εφαρμογής στους πόρους και δίπλα το αποτέλεσμα που προκύπτει από την εφαρμογή της συνάρτησης βαθμολόγησης

Πόροι	Πραγματική τιμή	Υπολογισθείσα τιμή
cream-ce01.marie.hellasgrid.gr	29,72	26,932
yildirim.grid.boun.edu.tr	37,695	37,591
ce01.kallisto.hellasgrid.gr	29,615	24,895
ce01.afroditi.hellasgrid.gr	364,6	362,589
ce01.ariagni.hellasgrid.gr	365,57	362,731
ce01.isabella.grnet.gr	22,904	24,895

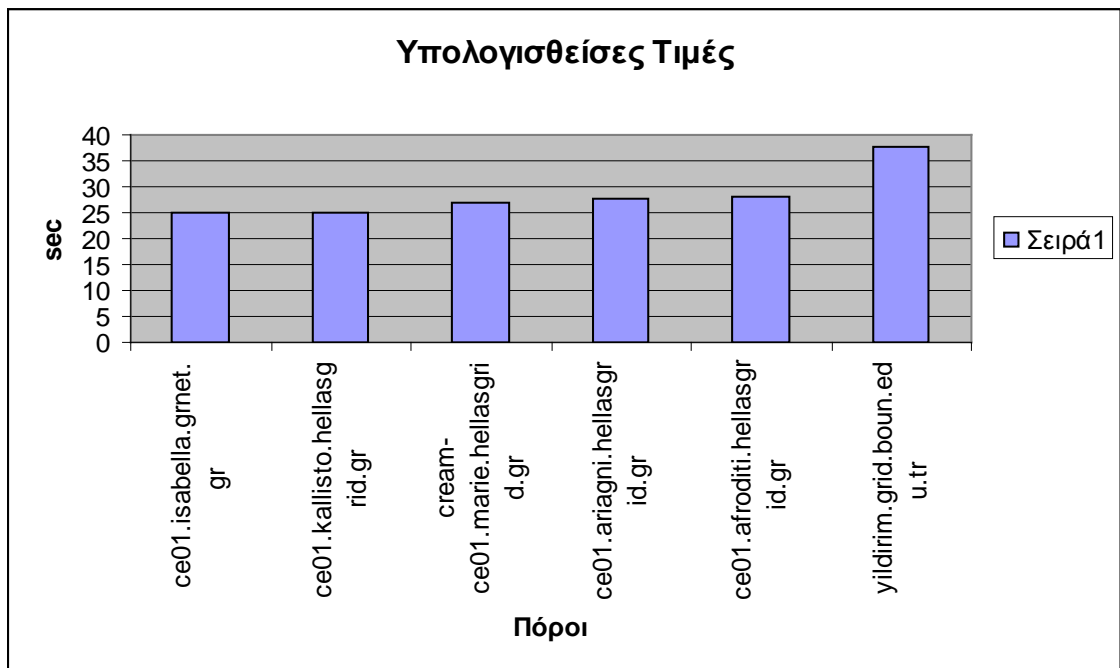
Πίνακας 10 – Πραγματικές και υπολογισθείσες τιμές

Η συσχέτιση που προκύπτει είναι (Pearson Correlation) **0,881516** και θα τη χαρακτηρίσαμε σχετικά καλή. Επίσης γίνεται χρήση Kendall's Correlation η τιμή του οποίου είναι **0.873326** καθώς και Spearman's_Correlation με τιμή **0,945615**.

Τέλος θα παρουσιαστούν άλλα δυο γραφήματα, το πρώτο παρουσιάζει τις πραγματικές τιμές με αύξουσα σειρά ενώ το δεύτερο τις υπολογισθείσες τιμές πάλι με αύξουσα σειρά



Εικόνα 5.10 – Γραφική απεικόνιση πραγματικών τιμών των πόρων



Εικόνα 5.11 – Γραφική απεικόνιση υπολογισθέντων τιμών των πόρων

Συνοψίζοντας, και στα δύο πειράματα που πραγματοποιήθηκαν με διαφορετικές εφαρμογές, το σύστημα κατάφερε να δημιουργήσει τις κατάλληλες συναρτήσεις βαθμολόγησης. Αυτό βεβαιώνεται και από τα αποτελέσματα που προέκυψαν από τη χρήση των rank correlations μεταξύ των πραγματικών και των υπολογισθεισών τιμών. Όλες οι τιμές κυμαίνονταν πάνω από 0.85 (με μέγιστο το 1).

Στη πρώτη περίπτωση (dwarf SciMark) το σύστημα επέλεξε σωστά μετρικές που σχετίζονται με Mflops (μιας και το dwarf αυτό επιστρέφει Mflops). Στη δεύτερη περίπτωση (dwarf nbody) το σύστημα επέλεξε και πάλι τη σωστή μετρική που σχετίζεται με sec (μιας και το dwarf αυτό επιστρέφει sec).

Και τα δυο πειράματα πραγματοποιήθηκαν στο ίδιο V.O (το SEE) όπου υπάρχει μεγάλος αριθμός πόρων καθώς και μεγάλη ετερογένεια μεταξύ αυτών. Θα μπορούσαμε να χαρακτηρίσουμε το V.O SEE ως ένα αντιπροσωπευτικό V.O του υπολογιστικού πλέγματος (με βάση των αριθμό των πόρων που διατίθενται και των αριθμό των εγγεγραμμένων χρηστών) δίνοντας έτσι μεγαλύτερη αξία στα αποτελέσματα των πειραμάτων.

Κεφάλαιο 6

Επίλογος

Στο κεφάλαιο αυτό γίνεται η σύνοψη των αποτελεσμάτων της διπλωματικής εργασίας δίνοντας κάποια συμπεράσματα καθώς και κάποιες αναφορές σε βελτιώσεις και μελλοντικές επεκτάσεις του συστήματος.

6.1 Σύνοψη

Στην παρούσα διπλωματική εργασία πραγματοποιήθηκε ο σχεδιασμός και η υλοποίηση ενός συστήματος που επιτρέπει στο χρήστη να υπολογίζει την επίδοση μιας εφαρμογής χωρίς να χρειαστεί να εκτελεστεί σε κάθε πόρο του υπολογιστικού πλέγματος αλλά να προσομοιώνει το αποτέλεσμα με τη χρήση μιας συνάρτησης βαθμολόγησης και των micro benchmarks.

Συγκεκριμένα αναπτύχθηκαν τα εξής :

- Ένας οδηγός (wizard) ο τερματισμός του οποίου δημιουργεί τη συνάρτηση βαθμολόγησης και η οποία αποθηκεύεται μαζί με άλλες πληροφορίες σε ένα αρχείο xml. Η δημιουργία της συνάρτησης μπορεί να γίνει είτε από το σύστημα αυτόματα είτε από τον ίδιο τον χρήστη.
- Ένα πλαίσιο εργασίας που επιτρέπει την εύκολη προσθήκη νέων μαθηματικών μοντέλων για τον υπολογισμό της συνάρτησης βαθμολόγησης.
- Ένα πλαίσιο εργασίας που αναλαμβάνει την παρουσίαση των αποτελεσμάτων σε γραφικό περιβάλλον.
- Ένας συντάκτης (editor) που δίνει τη δυνατότητα στο χρήστη να επιλέξει εκ των υστέρων και άλλους πόρους για μέτρηση.

Επιπλέον χαρακτηριστικά του συστήματος :

- Το σύστημα αναπτύχθηκε εντός του εργαλείου g Eclipse και πιο συγκεκριμένα επέκτεινε τη λειτουργία του πλαισίου εργασίας της μέτρησης επίδοσης .
- Πλήρης διαχωρισμός της λογικής του συστήματος από τη γραφική απεικόνιση και τα δεδομένα.

6.2 Συμπεράσματα

Όπως αναφέρθηκε και στην προηγούμενη υποενότητα η διπλωματική εργασία κινήθηκε σε δύο πλαίσια. Στη δημιουργία ενός συστήματος που θα επιτρέψει στο χρήστη να κάνει πρόβλεψη σε ποίο ν πόρο θα εκτελεστεί η εφαρμογή πο υ τον ενδιαφέρει πιο γρήγορα στο υπολογιστικό πλέγμα καθώς και στην ευχρηστία του συστήματος, δηλαδή η διαδικασία αυτή να γίνεται άμεσα και με κατανοητό τρόπο. Η

υλοποίηση βασίστηκε στο ήδη υπάρχον πλαίσιο εργασίας του g Eclipse και οι τεχνολογίες που χρησιμοποιήθηκαν είναι JAVA, JFaces, SAXP και γενικά ότι βιβλιοθήκες και δυνατότητες προσφέρει το πλαίσιο εργασίας του Eclipse.

Με βάση τα δύο πειράματα που έγιναν προκειμένου να διαπιστωθεί η ικανότητα του συστήματος στη πρόβλεψη της επίδοσης μιας εφαρμογής, τα αποτελέσματα ήταν αρκετά ενθαρρυντικά. Το σύστημα κατάφερε και στις δύο περιπτώσεις να δημιουργήσει κατάλληλες συναρτήσεις βαθμολόγησης με τιμές των rank correlations να φτάνουν έως και το 0.95 (με μέγιστη τιμή 1). Επομένως μπορούμε να ισχυριστούμε ότι το σύστημα είναι ικανό να προβλέψει την επίδοση μιας εφαρμογής με τη χρήση low level benchmarks. Τα πειράματα πραγματοποιήθηκαν στο V.O SEE (South East Europe) ένα αρκετά μεγάλο V.O με πληθώρα διαφορετικών πόρων, επομένως μπορούμε να ισχυριστούμε ότι τα πειράματα πραγματοποιήθηκαν σε ένα περιβάλλον που συναντά ένας χρήστης στο υπολογιστικό πλέγμα άρα τα αποτελέσματα έχουν πρακτική αξία. Τέλος αξίζει να αναφερθεί η ευκολία με την οποία δημιουργείται η συνάρτηση βαθμολόγησης μέσω ενός οδηγού χωρίς να απαιτούνται ιδιαίτερες γνώσεις από την πλευρά του χρήστη.

6.3 Μελλοντικές Επεκτάσεις

Παρόλο που έγινε προσπάθεια να δημιουργηθεί ένα ολοκληρωμένο σύστημα διαθέσιμο προς το χρήστη, υπάρχει προφανώς πληθώρα ιδεών και λειτουργιών που θα μπορούσαν να προστεθούν.

Επεκτάσεις στο μαθηματικό μοντέλο

- Το σύστημα προσπαθεί να προβλέψει τη σωστή συνάρτηση με χρήση γραμμικής παλινδρομώσεως πρώτου βαθμού (δηλαδή πρωτοβάθμια εξίσωση). Θα μπορούσε ο χρήστης (ή το σύστημα αυτόματα) να επιλέγει για παράδειγμα πολυώνυμο n βαθμού αυξάνοντας έτσι την πιθανότητα να υπάρχει καλύτερη πρόβλεψη.
- Θα μπορούσε να αναπτυχθεί μια μέθοδος για να απομακρύνονται ακραίες τιμές οι οποίες μπορούν να αλλοιώσουν σημαντικά τα αποτελέσματα.
- Θα ήταν πολύ χρήσιμο για τους χρήστες να υπήρχαν περισσότερες μετρικές αξιολόγησης που να δείχνουν πόσο καλή είναι η συνάρτηση βαθμολόγησης που δημιουργήθηκε.

Το σύστημα όπως έχει ήδη αναφερθεί μπορεί εύκολα να επεκταθεί αλλάζοντας τις μεθόδους που υλοποιούν τους μαθηματικούς υπολογισμούς μιας και είναι ανεξάρτητες από τον βασικό αλγόριθμο ο οποίος και παραμένει ίδιος.

Επεκτάσεις στο γραφικό μοντέλο

- Θα μπορούσε πιθανώς να αναπτυχθούν και άλλες σελίδες στο συντάκτη (editor) του xml αρχείου, ώστε ο χρήστης να κάνει και άλλες αλλαγές εκτός από αυτές της επιλογής πόρων.
- Θα μπορούσε να χρησιμοποιηθεί καλύτερο γραφικό πακέτο για τη γραφική παράσταση των πόρων από το πακέτο BIRT που χρησιμοποιείται.

Γενικά οποιαδήποτε γραφική προσθήκη θα μπορούσε να γίνει άμεσα και εύκολα διότι απλά αρκεί η επεξεργασία του xml αρχείου, τα δεδομένα του οποίου να ενσωματωθούν στο γραφικό πακέτο.

Γενικές επεκτάσεις στο σύστημα

- Πρέπει να προστίθενται και νέα micro benchmarks (τα οποία και φέρουν τις μετρικές) ώστε να υπάρχουν περισσότερες πιθανότητες να χαρακτηρίσουν μια εφαρμογή.
- Να αναπτυχθούν περισσότερα dwarfs ώστε να καλύπτουν όλες τις διαθέσιμες δυνατές κατηγορίες, δίνοντας μια καλή προσέγγιση για όλων των ειδών τις εφαρμογές .
- Επίσης πολύ χρήσιμο θα ήταν να δημιουργηθεί ένας χρονο-προγραμματιστής (scheduler) ο οποίος θα αναλάβει να εκτελεί micro benchmarks σε πόρους που θα επιλέγονται ανά τακτά χρονικά διαστήματα, ώστε οι μετρήσεις που παίρνονται (με βάση τις οποίες δημιουργείται η συνάρτηση βαθμολόγησης) να είναι όσο το δυνατό περισσότερο αντιπροσωπευτικές της κατάστασης του πόρου .
- Να δίνεται η δυνατότητα στη μέτρηση επίδοσης στο g eclipse να διαγράφονται benchmarks που είναι αποθηκευμένα στη βάση .
- Να μπορεί ο χρήστης τους πόρους με το καλύτερο αποτέλεσμα αυτόματα να τους ενσωματώνει σε ένα jsdl αρχείο.

Προσαρμογή στο Cloud

Όπως έχει ήδη αναφερθεί το σύστημα αναπτύχθηκε εντός του εργαλείου g Eclipse, επομένως χρησιμοποιείται εντός του υπολογιστικού πλέγματος και για συγκεκριμένα ενδιάμεσα λογισμικά (middlewares) που αυτό προσφέρει (πχ gLite). Όπως προκύπτει από την έρευνα τον τελευταίο καιρό, οδηγούμαστε προς μια νέα τεχνολογία ή αλλιώς σε μια διαφορετική μορφή από αυτή που είναι γνωστή ως τώρα το υπολογιστικό πλέγμα και αυτό είναι το cloud computing.

Το cloud computing [23] είναι μια επέκταση της ιδέας του grid computing, όπου μεγάλες/πολλές υπολογιστικές και αποθηκευτικές μονάδες διάσπαρτες γεωγραφικά γίνονται διαθέσιμες στους χρήστες μέσω του διαδικτύου με έναν ενιαίο τρόπο (βασισμένο στο δημοφιλές μοντέλο των web services) διαχείρισης και αξιοποίησης των πόρων αυτών, δίνοντας όμως ιδιαίτερη έμφαση στις δυνατότητες δυναμικής και ευέλικτης επέκτασης (ή/και συρρίκνωσης) της χρήσης των πόρων αυτών (scalability) ανάλογα με τις ανάγκες των εφαρμογών που τους χρησιμοποιούν.

Επομένως και σε αυτή τη περίπτωση η ανάγκη για μέτρηση της επίδοσης των πόρων που προσφέρονται στο χρήστη είναι επιτακτική ώστε για παράδειγμα να μπορεί να διαπιστώσει ο χρήστης αν αυτό που πληρώνει είναι αυτό που παίρνει. Το cloud όμως δεν υποστηρίζει κάποιο συγκεκριμένο ενδιάμεσο λογισμικό, ούτε στέλνονται «δουλειές-εργασίες» όπως στο grid, επομένως προκειμένου να μπορεί να ενσωματωθεί το σύστημα στο cloud πρέπει να υπάρξει διαχωρισμός από το υπάρχων πλαίσιο μέτρησης επίδοσης του g Eclipse, να δημιουργηθεί νέα ανεξάρτητη όψη (view) και ένα διαφορετικό πλαίσιο εργασίας το οποίο θα εξασφαλίζει την πρόσβαση στο cloud (πχ μέσω ssh) και να επιτρέπει την εκτέλεση των benchmarks. Τέλος θα μπορούσε να χρησιμοποιεί το τροποποιημένο (ειδικά για cloud) SiteRank

προκειμένου να μπορεί να μετρηθεί η επίδοση μιας εφαρμογής σε κάποιο cloud. Για παράδειγμα θα μπορούσαν να χρησιμοποιηθούν κάποια clouds (Amazon, Google) για δειγματοληψία και με βάση αυτά να δημιουργείται μια συνάρτηση βαθμολόγησης που θα μπορεί να χρησιμοποιηθεί σε άλλα cloud στη συνέχεια για πρόβλεψη της επίδοσης μιας εφαρμογής.

Αναφορές

1. Wreicker r, Benchmarking. Proceedings of Performance 2002 (Lecture Notes in Computer Science, vol 2459), Calzarossa MC, Tucci S(eds,). Springer: Berli, 2002; 179 -207.
2. Eigenmann, R, e, 2001. Performance Evaluation And Benchmarking with Realistic Applications. Cambridge, Mass : MIT Press
3. Marios D. Dikaiakos: Grid benchmarking: Vision, Challenges, and Current status. Concurrency and Computation: Practice and Experience 19(1): 90-99 (2007)
4. Foster, I., Kesselman, C: the Grid: Blueprint for a New Computing Infrastructure. In: Concepts and Architecture, ch. 4, pp.37-64. Elsevier, Amsterdam(2004)
5. Author not indicated. (2008) Dwarf Mine. <http://view.eecs.berkeley.edu/wiki/Dwarfs>
6. Tsouloupas G, Dikaiakos MD. GridBench: A tool for benchmarking Grids. Proceeding of the 4th International Workshop on Grid Computing(Grid2003). IEEE Computer Society Press: Los Alamitos. CA, 2003; 60-67
7. Tsouloupas, G., Dikaiakos M.D: GridBench: A Workbench for Grid Benchmarking. In: Sloot, P.M.A, Hoekstra, A.G., Priol, T., Reinefeld, A., Bubal, M. (eds) EGC 2005. LNCS, vol 3470 pp 211-225. Springer, Heidelberg(2005)
8. The Eclipse Project, <http://www.eclipse.org>
9. The g Eclipse Project, <http://www.geclipse.eu>
10. Kornmayer H, Stumpert M, Gjermundrod H. Wolniewicz P, g Eclipse A Contextualised Framework for Grid Users, Grid Resource Providers and Grid Application Developers . M Bubal : ICCS 2008 Part III, LNCs 5103 pp 399-408, 2008. Springer – Verlag Berlin Heidelberg 2008.
11. g Eclipse architecture II, Public project deliverable D1.5 of the g Eclipse project, <http://www.geclipse.eu/fileadmin/Documents/Deriverables/D1.5.pdf>
12. Tsouloupas, G Dikaiakos, MD : Grid Resource Ranking Using Low-Level Performance Measurements e-Science and Grid Computing, Second IEEE International Conference on e-Science and Grid Computing pp 70-70. IEEE Computer Society, Los Alamitos(2006)
13. Tsouloupas, G., Dikaiakos, MD : Characterization of Computational Grid Resources Using Low Level Benchmarks. In: Second IEEE International Conference on e-Science and Grid Computing pp 70-77. IEEE Computer Society, Los Alamitos (2006)

14. Allison, P.D. "Multiple Regression: a primer", page 142. Pine Forge Press: Thousand Oaks, C.A. 1999.
15. Longnecker, M.T & Ott, R.L : "A First Course in Statistical Methods", page 615. Thomson Brooks/Cole, 2004.
16. Basileios B "Statistics", page 200-214. Stamoulis 1997.
17. Enabling Grids for E-Science project, <http://www.eu-egee-org>
18. Gamma, E. Beck, K. : Contributing to Eclipse. Principles, Patterns, and Plugins. Addison-Wesley Longman, Amsterdam(2003)
19. Hemmer, F., Laure, E., Barroso Lopez, M., Di Meglio, A., Fisher, S., Guy, L., Knust, P., Prelz, F.: Middleware for the Next Generation Grid Infrastructure. In Proceeding of CHEP 2004, Intelken, Switzerland (2004), <http://glite.web.cern.ch/glite>
20. R.V der Wijngaart. Computationally intensive grid benchmarks. Global grid Forum (2003).
21. g Eclipse User Manual, Public project deliverable D2 of the g Eclipse project
22. The SciMark 2.0 project, <http://math.nist.gov/scimark2/index.html>
23. Gruman, Galen (2008-04-07). "What cloud computing really means". InfoWorld. Retrieved on 2009-01-13.
24. Nbody Benchmark
<http://shootout.alioth.debian.org/gp4/benchmark.php?test=nbody&lang=all>
25. Dubinko, Micah "XForms 1.0 Basic Profile", (2004-11-12).

Παράρτημα

Στο παράρτημα θα δοθούν διάφορες επιπλέον πληροφορίες που αφορούν την υλοποίηση, τα benchmarks και γενικά για τα εργαλεία που χρησιμοποιήθηκαν.

XML schema (rank.xsd)

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- definition of simple elements -->
  <xs:element name="Coefficient" type="xs:double"/>
  <xs:element name="Metric" type="xs:string"/>
  <xs:element name="ComputerElement" type="xs:string"/>
  <xs:element name="Description" type="xs:string"/>
  <xs:element name="Date" type="xs:string"/>
  <xs:element name="Created" type="xs:string"/>
  <xs:element name="Intercept" type="xs:double"/>
  <xs:element name="ApplicationName" type="xs:string"/>

  <!-- definition of complex elements -->

  <xs:element name="Rank">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="LinearModel" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Term">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Coefficient"/>
        <xs:element ref="Metric"/>
        <xs:element ref="Method"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="LinearModel">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Model" maxOccurs="1"/>
        <xs:element ref="ComputerElement" maxOccurs="unbounded"/>
        <xs:element ref="Description"/>
        <xs:element ref="Date"/>
        <xs:element ref="Created"/>
        <xs:element ref="ApplicationName" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

<xs:element name="Model">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Term" maxOccurs="unbounded"/>
      <xs:element ref="Intercept" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

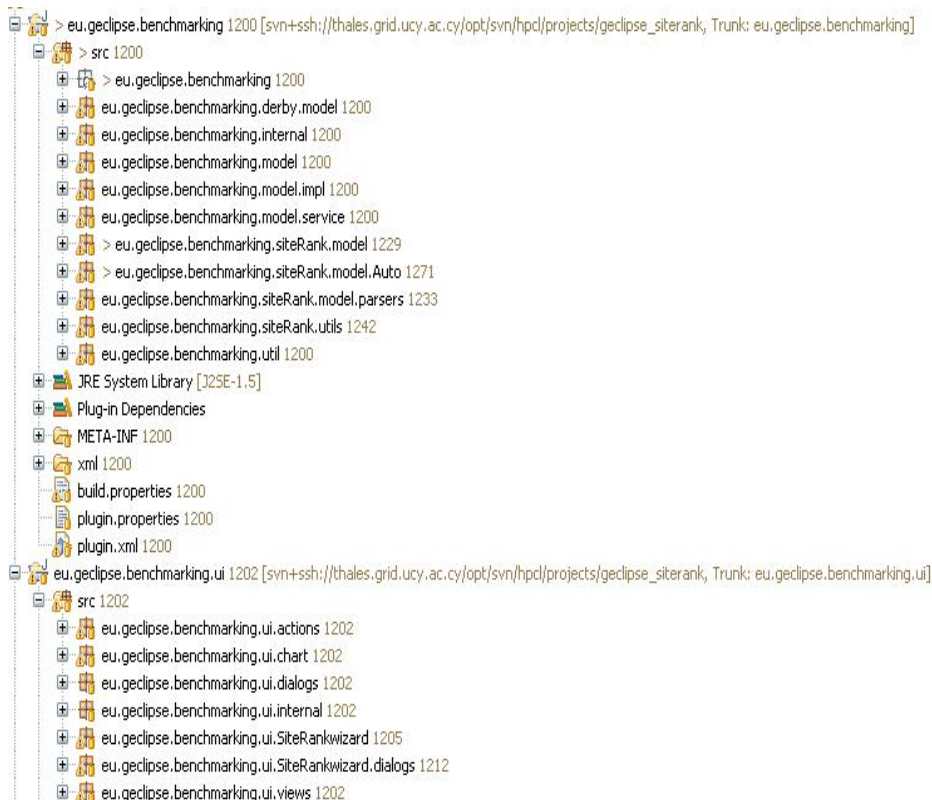
<xs:element name="Method">
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:enumeration value="Mean"/>
    <xs:enumeration value="Max"/>
    <xs:enumeration value="Min"/>
  <xs:enumeration value="St dev"/>
  <xs:enumeration value="Aver dev"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>

</xs:schema>

```

Δομή των καταλόγων – plugins

Στην παρακάτω εικόνα φαίνεται η δομή των καταλόγων εντός των βυσμάτων



Όπως φαίνεται υπάρχουν δύο βύσματα (plugins), το ένα αφορά το λογικό κομμάτι της μέτρησης επίδοσης στο g Eclipse και είναι το eu.geclipse.benchmarking ενώ το άλλο

περιλαμβάνει όλα τα απαραίτητα στοιχεία για την απεικόνιση και είναι το eu.geclipse.benchmarking.ui. Οι κατάλογοι που αφορούν το σύστημα που αναπτύχθηκε εντός της μέτρησης επίδοσης είναι κυρίως :

eu.geclipse.benchmarking.siteRank.model,
 eu.geclipse.benchmarking.siteRank.model.Auto,
 eu.geclipse.benchmarking.siteRank.model.parsers
 eu.geclipse.benchmarking.siteRank.model.utils
 eu.geclipse.benchmarking.ui.SiteRankWizard
 eu.geclipse.benchmarking.ui.SiteRankWizard.dialogs

Βέβαια υπάρχουν και υλοποιήσεις και σε άλλα αρχεία όπου κρίθηκε αυτό απαραίτητο

Εργαλεία που χρησιμοποιήθηκαν

Η γλώσσα που χρησιμοποιήθηκε για την ανάπτυξη του συστήματος ήταν

- java συνοδευόμενη από διάφορες τεχνολογίες που σχετίζονται με αυτήν, όπως Jfaces, διάφορα συστατικά (components) του Eclipse, SAXP (για xml parsing).
- Σχεσιακή βάση δεδομένων Derby Version 2 και αντίστοιχα η γλώσσα sql για επερωτήσεις. Έγινε χρήση της Derby γιατί αυτή είναι ελεύθερη για χρήση από το Eclipse Foundation.
- Για την γραφική παράσταση χρησιμοποιήθηκε το πακέτο-βύσμα BIRD γιατί αυτό ήταν ελεύθερο για χρήση από το Eclipse Foundation. Εδώ πρέπει να επισημανθεί η λιγοστή υποστήριξη που υπάρχει για αυτό το λογισμικό.
- Python 3.1 για την δημιουργία των micro benchmarks.
- Για τα μαθηματικά μοντέλα χρησιμοποιήθηκαν μέθοδοι από το JAMA και SAGA.

micro Benchmarks

Στην παρακάτω εικόνα φαίνονται κάποια benchmarks και αντίστοιχες μετρικές που αυτά φέρουν.

Factor	Metric	Delivered By
CPU	Operations per second (mixture of floating point and integer arithmetic)	EPWhetstone
CPU	Floating-Point operations per second	EPFlops
CPU	Integer operations per second	EPDhrystone
memory	sustainable memory bandwidth in MB/s (copy,add,multiply,triad)	EPStream
memory	Available physical memory in MB	EPMemsize
cache	memory bandwidth using different memory sizes in MB/s	CacheBench
Inter connect	latency, bandwidth and bisection bandwidth	MPPTest
I/O	Effective I/O bandwidth	b_eff_io

Micro benchmarks (Πηγή εικόνας : Tsouloupas G)

Mflops : Μετράει την επίδοση της CPU κάνοντας πράξεις κινητής υποδιαστολής με διαφορετικό τρόπο. Χρησιμοποιεί 8 υπομονάδες όπου κάθε υπομονάδα αποτελείται από διαφορετικούς τύπους λειτουργιών. Οι διαφορετικοί συνδυασμοί δημιουργούν ένα σετ από 4 μετρικές με διαφορετικά αποτελέσματα για κάθε μία από αυτές. Το benchmarks προσπαθεί να μεγιστοποιήσει τη χρήση του καταχωρητή προκειμένου να είναι ανεξάρτητο το αποτέλεσμα όσο είναι δυνατό από τη μνήμη του συστήματος. Συνήθως υλοποιείται στην γλώσσα C και χρόνος εκτέλεσης του κυμαίνεται κάτω από 5 λεπτά.

Dhrystone : Είναι ένα benchmarks το οποίο κάνει ακέραιους υπολογισμούς σε ένα σετ από CPU's. Είναι συνήθως γραμμένο στην γλώσσα C και ο χρόνος εκτέλεσης του κυμαίνεται κάτω από 10 sec

Stream : Το Benchmarks αυτό μετράει το μέγιστο εύρος ζώνης της μνήμης που μπορεί να χρησιμοποιηθεί (MB/sec). Το benchmark κάνει υπολογισμούς με τη χρήση 4 λειτουργιών: copy, scale, sum, και triad. Ο χρόνος εκτέλεσης είναι περίπου 10 sec

memsize: Το benchmark αυτό υπολογίζει τη χωρητικότητα της μνήμης. Αρχικά καθορίζει το μέγιστο ποσό της μνήμης που μπορεί να δεσμεύσει στην φυσική μνήμη. Το μέγεθος της φυσικής μνήμης είναι σημαντικό για εφαρμογές που «προτιμούν» να δεσμεύουν όσοι μνήμη μπορούν αποφεύγοντας την αργή swap μνήμη.

Bonnie: Το benchmarks αυτό έχει σκοπό να πραγματοποιήσει μια πληθώρα tests που σχετίζονται με το σκληρό δίσκο και το σύστημα αρχείων.

Dwarf Scimark 2.0 : Το dwarf αυτό αφορά μαθηματικούς υπολογισμούς που αφορούν κυρίως προβλήματα μηχανικής. Χρησιμοποιεί 4 διαφορετικούς υπολογισμούς που επιστρέφουν Mflops και παίρνει το μέσο όρο αυτών.

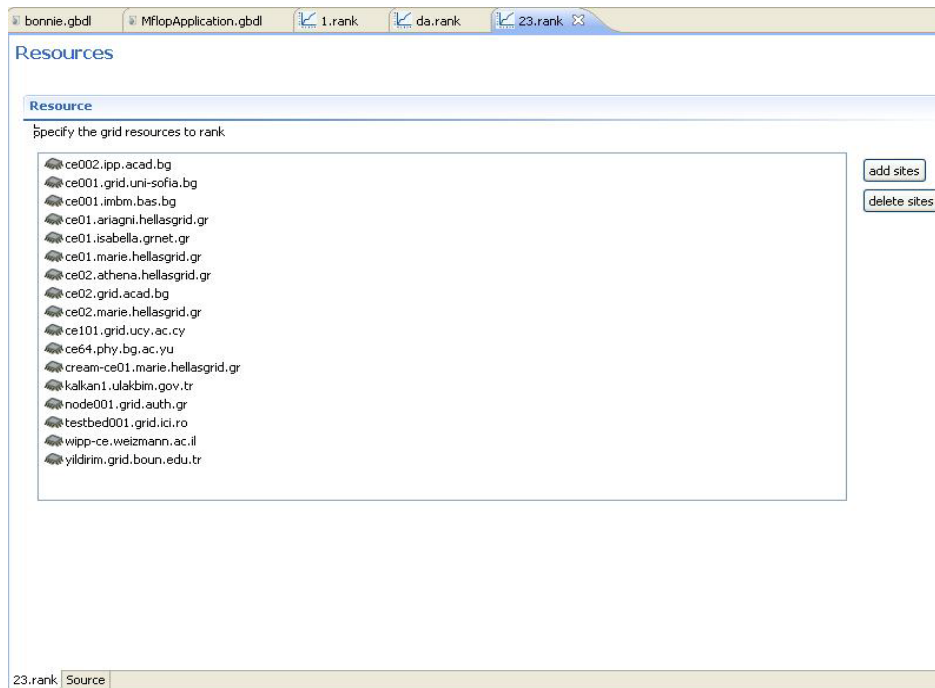
Κάνει τους εξής υπολογισμούς :

Γρήγορος μετασχηματισμός Fourier (Fast Fourier Transform), πολλαπλασιασμός αραιών πινάκων (Sparse Matrix Multiply), μέθοδο Monte Carlo και μέθοδο Gauss-Siedel

Dwarf Nbody : Το dwarf αυτό αφορά υπολογισμούς που εξαρτώνται από τις αλληλεπιδράσεις μεταξύ διαφορετικών διακριτών σημείων

Συντάκτης Editor

Δημιουργήθηκε ένας συντάκτης των xml αρχείων προκειμένου ο χρήστης να μπορεί να επιλέγει και άλλους πόρους για μέτρηση γρήγορα και εύκολα. Στην παρακάτω εικόνα φαίνεται η μορφή του



Εικόνα συντάκτη(Editor)

Θεωρητικό υπόβαθρο Στατιστικής

Παρακάτω θα παρουσιαστούν λεπτομερώς όλοι οι στατιστικοί ορισμοί και μέθοδοι που χρησιμοποιήθηκαν στο κεφάλαιο 4 κυρίως στην ενότητα «4.4.3 Αλγόριθμος δημιουργίας της συνάρτησης βαθμολόγησης»

Ορισμοί

1. Στατιστική: είναι ο κλάδος των εφαρμοσμένων μαθηματικών, ο οποίος βασίζεται σε ένα σύνολο αρχών και μεθοδολογιών για

- Το σχεδιασμό της διαδικασίας συλλογής δεδομένων (data).
- Τη συνοπτική και αποτελεσματική παρουσίαση των δεδομένων.
- Την ανάλυση και εξαγωγή συμπερασμάτων από τα δεδομένα.

Η στατιστική είναι μια ατελής επαγωγή. Από τις ιδιότητες του μέρους εξάγει συμπεράσματα για το όλον.

2. Πληθυσμός (Population - Sample Space): είναι ένα σύνολο, τα στοιχεία του οποίου εξετάζονται ως προς ένα ή περισσότερα χαρακτηριστικά τους.

3. Μεταβλητή (Variable): ενός πληθυσμού είναι το χαρακτηριστικό ως προς το οποίο εξετάζεται ο πληθυσμός. Μεταβλητές είναι δύο ειδών, ποιοτικές ή ποσοτικές. Οι ποσοτικές μεταβλητές διακρίνονται σε διακριτές και συνεχείς.

4. Συχνότητα (Frequency): μιάς τιμής χ_i της μεταβλητής X μεγέθους $n \times N$ είναι ο φυσικός αριθμός $n_i \times N$, $n_i \leq n$, που δείχνει το πλήθος εμφάνισης της τιμής χ_i . Ισχύει προφανώς: $n_1 + n_2 + \dots + n_m = n$ [16]

5. Μέτρα Διασποράς: Παρακάτω αναφέρονται διάφορα μέτρα διασποράς

1. Εύρος R: είναι η διαφορά μεταξύ της ελαχίστης παρατήρησης από την μεγίστη. Δηλαδή

$$R = X_{\max} - X_{\min}$$

2. Μέση τιμή: Η μέση τιμή είναι η καλύτερη εκτίμηση της πραγματικής τιμής. Επαναλαμβανόμενες μετρήσεις ακολουθούν τη κανονική ή Γκαουσιανή κατανομή.

3. Μέση Απόκλιση : Είναι η μέση τιμή, της απόλυτης τιμής των αποκλίσεων των τιμών της μεταβλητής από τη μέση τιμή τους.

4. Διακύμανση ή Διασπορά s^2 (Second Moment or Variation): ορίζεται από την σχέση

$$s^2 = \frac{1}{v} \sum_{i=1}^v (x_i - \bar{x})^2$$

Η διασπορά είναι η κυριότερη παράμετρος μεταβλητότητας. Όσο μικρότερη είναι, τόσο λιγότερο διαφέρουν οι τιμές των παρατηρήσεων από τη μέση τιμή τους [15].

5. Τυπική απόκλιση s (Standard Deviation): είναι η τετραγωνική ρίζα της διασποράς. Δηλαδή

$$s = \sqrt{s^2}$$

Η τυπική απόκλιση είναι το μέτρο της απόκλισης μιας τυπικής μέτρησης από τη μέση τιμή ενός συνόλου επαναλαμβανόμενων μετρήσεων. Μπορεί να αποδειχτεί ότι το 68% των μετρήσεων βρίσκονται μέσα στο διάστημα μιας τυπικής απόκλισης από τη μέση τιμή.

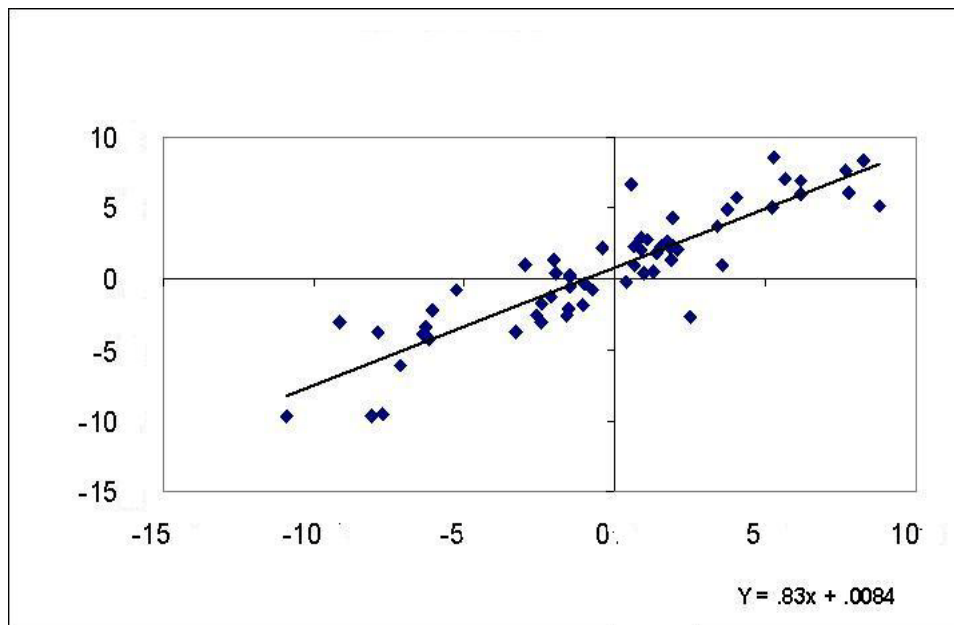
Η τυπική απόκλιση δημιουργήθηκε για το λόγο ότι η διασπορά, έχει ως μονάδα μέτρησης το τετράγωνο της αρχικής μονάδας. Λαμβάνοντας την τυπική απόκλιση, παίρνεται μέτρο διασποράς το οποίο θα έχει την ίδια μονάδα μέτρησης με το αρχικό δείγμα, όπως είναι και για όλα τα υπόλοιπα μέτρα.

6. Μέση Απόκλιση : Είναι η μέση τιμή, της απόλυτης τιμής των αποκλίσεων των τιμών της μεταβλητής από τη μέση τιμή τους $MD = \frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}|$

6. Γραμμική Παλινδρόμηση (Linear Regression): μιας εξαρτημένης μεταβλητή Y από την εξαρτημένη μεταβλητή X είναι η σχέση $y = a + \beta \cdot x$ όπου a και β είναι παράμετροι. Ο προσδιορισμός των a και β δίνει μια προσεγγιστική ευθεία, που συνδέει τις τιμές της Y δοθέντων των τιμών της X . Η ευθεία που προκύπτει λέγεται **ευθεία παλινδρόμησης της Y πάνω στην X** [16].

Η ευθεία αυτή μπορεί να κατασκευασθεί εμπειρικά ή μέσω μαθηματικών μεθόδων, όπως είναι η μέθοδος **ελαχίστων τετραγώνων**. Σκοπός είναι το άθροισμα των

τετραγώνων των κατακόρυφων αποστάσεων των σημείων (X,Y) από την ευθεία να είναι ελάχιστο. Στην παρακάτω εικόνα φαίνεται ένα παράδειγμα αυτής.



Γραμμική παλινδρόμηση

Επίσης ορίζεται η συνάρτηση γραμμικής πολλαπλής παλινδρομήσεως για $k-1$ ανεξάρτητες μεταβλητές X_1, X_2, \dots, X_{k-1} ως : $y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_{k-1} X_{k-1}$ όπου πρέπει να υπολογίσουμε k σταθερές $\alpha, \beta_1, \beta_2, \beta_{k-1}$. Μια γραμμική συνάρτηση, που προσαρμόζεται στα δεδομένα για δύο μεταβλητές αναφέρεται ως μια ευθεία γραμμή, μία γραμμική συνάρτηση για τρεις μεταβλητές είναι ένα επίπεδο, και μία γραμμική συνάρτηση για τέσσερις ή περισσότερες μεταβλητές είναι ένα υπερεπίπεδο. Στο σύστημα που αναπτύχθηκε χρησιμοποιείται κυρίως η συνάρτηση γραμμικής πολλαπλής παλινδρομήσεως όπου κάθε μεταβλητή αποτελεί μια μετρική που έχει επιλεγεί από το σύστημα (στην περίπτωση που έχουμε >1 επιλεγθέντες μετρικές).

7.Μέθοδος Ελαχίστων Τετραγώνων (Method of Least Squares) [15]: με τη μέθοδο αυτή προσδιορίζονται οι συντελεστές α και β από τους τύπους:

$$\hat{\beta} = \frac{v \sum_{i=1}^v x_i y_i - (\sum_{i=1}^v x_i)(\sum_{i=1}^v y_i)}{v \sum_{i=1}^v x_i^2 - (\sum_{i=1}^v x_i)^2}$$

Και
$$\hat{\alpha} = \bar{y} - \hat{\beta} \cdot \bar{x}$$

η ευθεία ελαχίστων τετραγώνων θα είναι η :

$$\hat{y} = \hat{\alpha} + \hat{\beta} \cdot x$$

Το $\hat{\alpha}$ είναι η τεταγμένη του σημείου στο οποίο η ευθεία τέμνει τον άξονα $y'y$ ενώ το $\hat{\beta}$, που είναι ο συντελεστής διεύθυνσης της ευθείας, εκφράζει την μεταβολή της

μεταβλητής Y όταν η μεταβλητή X μεταβληθεί κατά μια μονάδα. Με ανάλογο τρόπο υπολογίζονται όλοι οι συντελεστές στην περίπτωση πολλαπλής παλινδρόμησης.

8. Συντελεστής Γραμμικής Συσχέτισης (Linear Correlation Coefficient) [15]: είναι το μέτρο που εκφράζει τη συγκέντρωση των σημείων ενός διαγράμματος διασποράς γύρω από την ευθεία παλινδρόμησης. Ο συντελεστής **γραμμικής** συσχέτισης r δίνει ένα μέτρο του μεγέθους της **γραμμικής** συσχέτισης μεταξύ δύο μεταβλητών (ο συντελεστής γραμμικής συσχέτισης του Pearson ορίζεται ανάλογα και συμβολίζεται με ρ).

Αν X και Y δύο μεταβλητές μεγέθους n τότε ο συντελεστής γραμμικής συσχέτισης είναι ο εξής:

$$r(X, Y) = r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

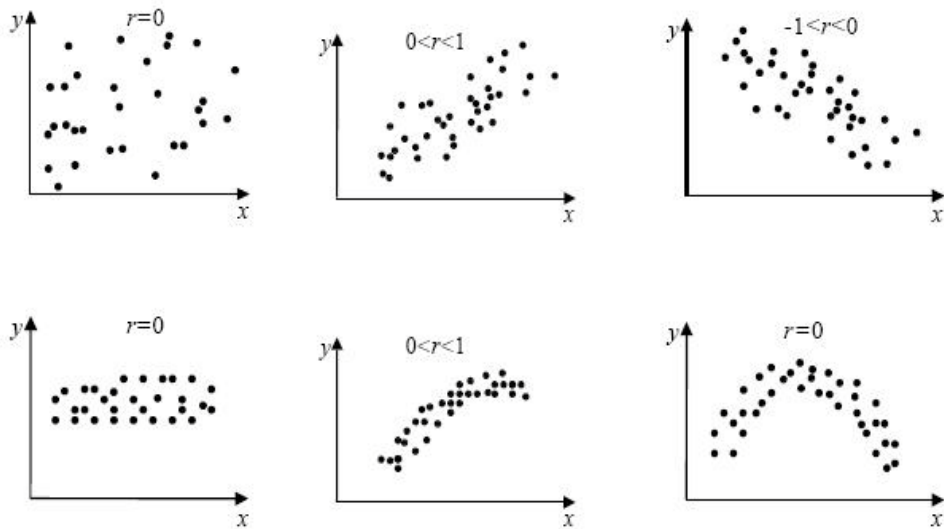
Εάν οι μέσες τιμές δεν είναι ακέραιοι αριθμοί, τότε ο συντελεστής γραμμικής συσχέτισης r δίνεται από τον τύπο:

$$r = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i) \cdot (\sum_{i=1}^n y_i)}{\sqrt{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \cdot \sqrt{n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2}}$$

Ιδιότητες του r :

1. Εάν $r = 1$ τότε έχουμε τέλεια θετική γραμμική συσχέτιση και όλα τα σημεία βρίσκονται πάνω στην ευθεία $y = \alpha + \beta \cdot x$ και $\beta > 0$. αντίστοιχα αν $r = -1$ και $\beta < 0$.
2. Εάν $r = 0$ τότε δεν υπάρχει γραμμική συσχέτιση μεταξύ των μεταβλητών X και Y . Οπότε λέμε ότι είναι **γραμμικά ασυσχέτιστες**.
3. Εάν $-0,3 \leq r < 0,3$ **δεν υπάρχει γραμμική** συσχέτιση. Αυτό, όμως, δεν σημαίνει ότι δεν υπάρχει άλλου είδους συσχέτιση μεταξύ των δύο μεταβλητών.
4. Αν $-0,5 < r \leq -0,3$ ή $0,3 \leq r < 0,5$ υπάρχει **ασθενής γραμμική** συσχέτιση.
5. Αν $-0,7 < r \leq -0,5$ ή $0,5 \leq r < 0,7$ υπάρχει **μέση γραμμική** συσχέτιση.
6. Αν $-0,8 < r \leq -0,7$ ή $0,7 \leq r < 0,8$ υπάρχει **ισχυρή γραμμική** συσχέτιση.

Στην παρακάτω εικόνα βλέπουμε την μορφή της γραφικής παράστασης ανάλογα με την τιμή του r



Γραφική παράσταση για τιμές του r

Θετικές τιμές του r δεν υποδηλώνουν, κατ' ανάγκη μεγαλύτερο βαθμό γραμμικής συσχέτισης από το βαθμό γραμμικής συσχέτισης που υποδηλώνουν αρνητικές τιμές του r . Ο βαθμός γραμμικής συσχέτισης καθορίζεται από την απόλυτη τιμή του r και όχι από το πρόσημο του r . Το πρόσημο του r καθορίζει το είδος μόνο, της συσχέτισης (θετική ή αρνητική). Πληροφορεί δηλαδή για το αν αύξηση της μιας μεταβλητής αντιστοιχεί σε αύξηση ή σε μείωση της άλλης μεταβλητής. Για παράδειγμα η τιμή $r = -0,9$ δείχνει ισχυρότερη γραμμική συσχέτιση από την τιμή $r = 0,8$ ενώ οι τιμές $r = -0,6$ και $r = 0,6$ δείχνουν ίδιο βαθμό γραμμικής συσχέτισης αλλά αντίθετο είδος.

Συσχέτιση δε σημαίνει αιτιότητα

Όταν σε μια μη πειραματική έρευνα (δειγματοληψία) δύο μεταβλητές X και Y βρίσκονται συσχετισμένες αυτό σημαίνει μόνο ότι οι μεταβλητές αυτές συνδέονται με κάποια σχέση. Δε συνεπάγεται, κατ' ανάγκη, αιτιότητα. Οι δύο μεταβλητές μπορεί βεβαία να συνδέονται με σχέση αιτιότητας, μπορεί όμως, όχι.

9. Συντελεστής Προσδιορισμού R^2 : είναι το πηλίκο

$$R^2 = \frac{SSR}{SST} \quad R \in (0,1)$$

το οποίο μπορεί να ελέγξει την αξία του απλού γραμμικού μοντέλου, το οποίο προσαρμόζεται στα δεδομένα [15]. Ο συντελεστής προσδιορισμού εκφράζει το ποσοστό της μεταβλητότητας της μεταβλητής Y που εξηγείται από τη μεταβλητή X . Όσο πιο κοντά βρίσκεται η τιμή του R^2 στην μονάδα, τόσο πιο ισχυρή γίνεται η γραμμική σχέση εξάρτησης των μεταβλητών Y και X .

Η τιμή κοντά στο μηδέν σημαίνει ότι το γραμμικό μοντέλο δεν μπορεί να εξηγήσει την διακύμανση ενώ τιμή κοντά στο 1 σημαίνει ότι όλη η διακύμανση μπορεί να εξηγηθεί.

Αυτό συμβαίνει όταν η σχέση μεταξύ της εξαρτημένης μεταβλητής και της ανεξάρτητης είναι ακριβώς γραμμική.

Το SSR είναι το άθροισμα των τετραγώνων της παλινδρόμησης και το SST είναι το συνολικό άθροισμα των τετραγώνων.

10. Συγγραμμικότητα: Ο όρος συγγραμμικότητα [16] περιγράφει ένα πρόβλημα που εμφανίζεται στην ανάλυση πολλαπλής παλινδρόμησης όταν ανεξάρτητες μεταβλητές είναι υψηλά συσχετισμένες μεταξύ τους. Σε τέτοιες περιπτώσεις δεν είναι δυνατόν να προσδιοριστούν τα επί μέρους αποτελέσματα των ανεξάρτητων μεταβλητών επί της εξαρτημένης μεταβλητής. Όταν υπάρχει συγγραμμικότητα μεταξύ ανεξάρτητων μεταβλητών σε μία εξίσωση πολλαπλής παλινδρόμησης οι συντελεστές παλινδρόμησης τείνουν να είναι αναξιόπιστοι. Συνήθως μία ή περισσότερες ανεξάρτητες μεταβλητές θα έχουν ένα συντελεστή που δεν είναι σημαντικά διαφορετικός από το μηδέν.

Ο απλούστερος τρόπος για την αντιμετώπιση του προβλήματος είναι η διαγραφή μίας ή περισσότερων μεταβλητών από τις συσχετιζόμενες μεταβλητές.

Μία μέθοδος για το εντοπισμό της συγγραμμικότητας (αυτή που γίνεται από το σύστημα) είναι με τη χρήση του Variance Inflation Factor το οποίο ορίζεται ως

$$VIF_j = \frac{1}{1 - R_j^2}$$

. Εδώ το R_j^2 είναι ο συντελεστής της πολλαπλής παλινδρόμησης ο οποίος προκύπτει από την παλινδρόμηση της ανεξάρτητης μεταβλητής X_j με όλες τις άλλες ανεξάρτητες μεταβλητές. Ο γενικός κανόνας είναι ότι για τιμή $VIF_j > 8$

υπάρχει περίπτωση για συγγραμμικότητα ενώ για τιμή $VIF_j > 12$ οι πιθανότητες είναι πολύ αυξημένες. Σε αυτή την περίπτωση διαγράφεται η μεταβλητή αυτή.

Ιδιότητα : Σε περίπτωση που υπάρχει ένας πίνακας συσχέτισης μεταξύ ανεξάρτητων μεταβλητών και δύο από αυτές έχουν υψηλή τιμή r , τότε υπολογίζοντας το τετράγωνο

της συσχέτισης αυτής, δηλαδή το R^2 και εφαρμόζοντας το στο n τύπο VIF

υπολογίζεται το κάτω όριο της τιμής VIF, μιας και το R^2 που θα υπολογιστεί από την παλινδρόμηση (ως προς όλες τις ανεξάρτητες μεταβλητές) θα είναι σίγουρα μεγαλύτερο. Το σύστημα κάνει χρήση αυτής ακριβώς της ιδιότητας προκειμένου να αφαιρέσει μεταβλητές που έχουν υψηλή συγγραμμικότητα γιατί οποιαδήποτε άλλη περίπτωση (δηλαδή ο συνδυασμός όλων των δυνατών περιπτώσεων) θα είχε μεγάλο υπολογιστικό κόστος μιας και όλα αυτά εκτελούνται δυναμικά και ο χρόνος έχει σημασία.