



# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ  
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**3<sup>RD</sup> PARTY BROKER**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Γιώργος Ε. Δρυμάκης

**Επιβλέπων:** Θεοδώρα Βαρβαρίγου  
Αν. Καθηγήτρια Ε.Μ.Π.

Αθήνα, Απρίλιος 2006





# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ  
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**3<sup>RD</sup> PARTY BROKER**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Γιώργος Ε. Δρυμάκης

**Επιβλέπων:** Θεοδώρα Βαρβαρίγου  
Αν. Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή επιτροπή την 00 Φεβρουαρίου 2006

.....  
Θεοδώρα Βαρβαρίγου  
Αν. Καθηγήτρια Ε.Μ.Π.

.....  
Εμμ. Πρωτονοτάριος  
Καθηγητής Ε.Μ.Π.

.....  
Γεώργιος Στασινόπουλος  
Καθηγητής Ε.Μ.Π.

Αθήνα, Απρίλιος 2006



.....  
Γιώργος Ε. Δρυμάκης

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

Copyright © Γιώργος Ε. Δρυμάκης, 2004

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω την καθηγήτρια μου και τον υπευθυνο αυτής της διπλωματικής για την ευκαιρια που μου δωσανε να ασχοληθω με ενα grid application, οπως επείσης και το ιδρυμα για την παροχη υλικου και μέσων για την αποπεράτωση αυτης της διπλωματικης. Τους υπόλοιπους που με την αμεριστη εμπιστοσύνη και υπομονή τους με βοήθησαν και θα εινα διπλα μου θα τους δείχνω την ευγνωμοσήνη μου καθημερινά.

## Περίληψη

Ο σκοπός της διπλωματικής εργασίας ήταν η βελτίωση ενός ολοκληρωμένου γραφικού περιβάλλοντος που θα χρησιμοποιείτε σαν εφαρμογή χρήστη για αποστολή εργασιών σε πολυπλέγματα.

Έγινε αναφορά σε μια ερευνητική εφαρμογή, την GRIA, που αναπτύχθηκε για επαγγελματική χρήση και πιο συγκεκριμένα για το web service της.

Εντοπίστηκαν σημεία που θα μπορούσαν να βελτιωθούν για την ευκολότερη και ταχύτερη χρήση του δικτυακού τόπου και δημιουργήθηκαν παράλληλες ιστοσελίδες που εξυπηρετούν την αυτόματη δρομολόγηση και αποστολή εργασιών προς πολυπλέγματα.

Για την δημιουργία και την εξέλιξη αυτών των παραπάνω ιστοσελίδων χρησιμοποιήθηκαν τεχνολογίες Java, JSP, Applet και της βάσης δεδομένων MySQL.

Η εργασία αυτή κάνει τον ήδη εύχρηστο δικτυακό τόπο της GRIA client ταχύτερο και ευκολότερο στην χρήση του.

## Λέξεις κλειδιά

Πολύπλεγμα, Grid, GRIA, γραφικό περιβάλλον, πρόγραμμα πελάτη, γραμμή εντολών, XML, γλώσσα προγραμματισμού, JAVA, JRE, Applet, JSP, Tomcat, MySQL, στατιστικά, αυτόματη δρομολόγηση, φόρτος εργασίας

## **Abstract**

The aim of this thesis is the improvement of a graphic environment, meant to be used for submitting work under grid technology. A research application, GRIA, is examined which was developed for professional use and more specifically for its web service.

Points that could be improved were located for an easier and automatic run and forwarding of those works towards grid technology.

For the creation and the evolution of the above sites, the technology of Java, JSP, Applet and MySQL data base were used.

This thesis makes the easy-to-use Internet site of GRIA client web service even faster and easier.

## **Keywords**

GRID, GRIA, application interface, client, command line, XML, JAVA, JRE, Applet, JSP, Tomcat, MySQL, statistics, automatic scheduling, workload



## Περιεχόμενα

Ευχαριστίες .....	6
Λέξεις κλειδιά .....	7
Abstract .....	8
Keywords .....	8
Περιεχόμενα .....	9
ΕΙΣΑΓΩΓΗ .....	11
Πρόλογος .....	12
Η αρχή της απομακρυσμένης επεξεργασίας .....	12
Σημαντικότητα του GRID .....	13
Τα πρώτα βήματα του GRID .....	13
Στόχος του GRID .....	14
Τα οφέλη του GRID .....	14
Το GRID σήμερα .....	15
Τι προσφέρει το Grid .....	16
Επαγγελματικές υλοποιήσεις του GRID .....	16
Προϋποθέσεις για ένα Grid .....	17
Εισαγωγή στο GRIA .....	18
Επαγγελματική χρήση του GRIA .....	19
Ποιότητα των Υπηρεσιών (QoS) .....	20
Το τρέχων GRIA .....	21
Επιλέγοντας την Java .....	21
Παροχέας (Supplier) .....	23
Πελάτης (Client) .....	24
Χρήση του Client .....	25
• gria open .....	26
• gria tender .....	26
• gria upload .....	28
• gria run .....	28
• gria download .....	29
• gria finish .....	30
• gria show .....	30
• gria browse .....	30
Δυσκολίες που παρουσιάζονται στο Client .....	30
Λύσεις των μειονεκτημάτων .....	32
Εξελίξεις της καινούριας εφαρμογής .....	34
Ο μεσολαβητής .....	38
Ελλείψεις του μεσολαβητή .....	40
Η τεχνολογία και τα εργαλεία που χρησιμοποιήθηκαν .....	41
Λειτουργία του μεσολαβητή .....	42
Open Account .....	43
Tender .....	45
Upload .....	47
Run .....	48
Download .....	49
Show – Browse .....	49
Finish .....	50
Επιπρόσθετα χαρακτηριστικά του μεσολαβητή .....	51
Βάση δεδομένων .....	51
Cheapest solution .....	53
Fastest solution .....	56
Workload .....	57
Upload στον μεσολαβητή .....	60
Upload στον παροχέα .....	61
Run .....	62
Εγκατάσταση του μεσολαβητή σε διάφορα λειτουργικά συστήματα .....	62
Αποτελέσματα χρήσης του μεσολαβητή .....	63
Ευχρηστία .....	64
Ασφάλεια .....	66
Διαχείριση .....	67
Στατιστικά .....	67
Αυτόματη δρομολόγηση .....	68
Συμπεράσματα .....	68



## ΕΙΣΑΓΩΓΗ

Σκοπός της διπλωματικής εργασίας ήταν η ανάπτυξη ενός αυτοματοποιημένου, ταχύτερου και ευκολότερου τρόπου αποστολής εργασιών σε πολυπλέγματα έτσι ώστε να γίνουν προσιτές οι τεχνολογίες πολυπλέγματος και σε ποιο απλούς χρήστες υπολογιστών.

Αναλυτικότερα έγινε αναφορά στις τεχνολογίες πολυπλέγματος και πιο συγκεκριμένα στο ερευνητικό πρόγραμμα GRIA που εξετάζει τις επαγγελματικές χρήσεις του πολυπλέγματος. Στην δικτυακή έκδοση του συγκεκριμένου προγράμματος προστέθηκε ένα κομμάτι που αυτοματοποιεί κάποιες διαδικασίες που ήταν υποχρεωμένος ο χρήστης να ολοκληρώσει, έτσι ώστε να αποφευχθούν τυχών λάθη και καθυστερήσεις κατά την εκτέλεση του. Για την αποπεράτωση του επίμαχου κομματιού γνωστό και ως 3<sup>rd</sup> party broker χρησιμοποιήθηκαν τεχνολογίες όπως, java για γλώσσα προγραμματισμού, jsp για την δημιουργία δυναμικών ιστοσελίδων, MySQL για βάση δεδομένων και όλα αυτά ήταν σε έναν υπολογιστή που είχε έναν εξυπηρετητή Tomcat.

Επίσης χρησιμοποιήθηκαν ανάλογα προγράμματα για την χρήση αυτών των τεχνολογιών. Το πρόγραμμα IntelliJ της IDEA για τον προγραμματισμό σε java και Dreamweaver της MACROMEDIA για την δημιουργία των δυναμικών σελίδων.

## **Πρόλογος**

Το πολύπλεγμα είναι ένα είδος κατανεμημένης χρήσης υπολογιστών που περιλαμβάνει την κοινή χρήση υπολογιστή, εφαρμογής, δεδομένων, χώρο αποθήκευσης και πόρων δικτύου με ασφαλή, ομοιόμορφο αξιόπιστο και συντονισμένο τρόπο ανάμεσα σε δυναμικά και γεωγραφικά διασκορπισμένους οργανισμούς. Ο σκοπός των πολυπλεγμάτων είναι η γρηγορότερη αποπεράτωση κάθε είδους υπολογισμών με αποτέλεσμα την άμεση εφαρμογή τους σε εταιρίες και υπηρεσίες που έχουν ανάγκη την ταχύτητα, ασφάλεια, αξιοπιστία και ποιότητα και όχι τα υπολογιστικά μέσα.

## **Η αρχή της απομακρυσμένης επεξεργασίας**

Η ιδέα της χρησιμοποίησης αδέσμευτων πόρων γεννήθηκε στις αρχές της δεκαετίας του '70 όταν οι υπολογιστές συνδέθηκαν σε δίκτυα για πρώτη φορά. Έτσι έγιναν πειραματικά προγράμματα όπως το Creeper-Reaper που είναι προγονός του ARPAnet και αργότερα του Internet.

Πιο συγκεκριμένα η γνωστή εταιρία Xerox το '73 εγκατέστησε το πρώτο δίκτυο Ethernet και δρομολογήθηκε η πρώτη προσπάθεια για κατανεμημένη υπολογιστική εφαρμογή. Οι επιστήμονες John F. Shoch και Jon A. Hupp δημιούργησαν ένα «σκουλήκι» (worm) το οποίο έτρεχε σε κάθε μηχανήμα του δικτύου και χρησιμοποιούσε τους αδέσμευτους πόρους για ωφελίμους σκοπούς.

Αργότερα ο Richard Crandall, επιστήμονας της Apple έβαζε αδρανείς δικτυωμένους υπολογιστές να εργάζονται με σκοπό να κάνουν υπολογισμούς και να συνδυάζουν τα αποτελέσματα με άλλους υπολογιστές στο δίκτυο.

Έτσι έγιναν τα πρώτα βήματα της κατανεμημένης επεξεργασίας.

## **Σημαντικότητα του GRID**

Οι εταιρίες για να επιτύχουν τους στόχους τους και να αναπτυχθούν εξαρτώνται από υπερβολικά υψηλές επεξεργαστικές δυνατότητες. Για να ικανοποιηθεί όμως η απαίτηση τους από επεξεργαστική ισχύ πρέπει ρισκάρουν μια επένδυση μεγάλου χρηματικού ποσού σε μηχανήματα και προσωπικό το οποίο πολλές εταιρίες αποφεύγουν να κάνουν. Έπειτα η συνεχής ανάπτυξη των υπολογιστικών υλικών απαρχαιώνεται σε πολύ μικρό χρονικό διάστημα με συνέπεια όποια αγορά και αν γίνει μετά από καιρό θεωρείτε ξεπερασμένη. Λύση για το πρόβλημα είναι τα πολυπλέγματα τα οποία δεν βρίσκονται στον χώρο της κάθε εταιρίας και έτσι δεν χρειάζεται να τα αγοράσουν, να τα συντηρούν και να τα αναβαθμίζουν αφού κατά κάποιο τρόπο θα τα νοικιάζουν.

## **Τα πρώτα βήματα του GRID**

Μετά το '90 όπου και το παγκόσμιο δίκτυο (internet) ωρίμασε δημιουργήθηκαν προγράμματα για την επίτευξη ενός σκοπού. Το κυριότερο και το πιο διαδεδομένο ακόμα και σήμερα είναι το SETI@home (Search for Extra Terrestrial Intelligence) το οποίο ξεκίνησε το '99 και από τότε χρησιμοποιεί 4,5 εκατομμύρια προσωπικούς υπολογιστές περίπου σήμερα για να επεξεργάζεται σήματα τα οποία τα λαμβάνουν κεραιές από το διάστημα με σκοπό να βρουν κάποιο σήμα το οποίο να προέρχεται από νοήμον ζωή εκτός της γης. Αυτό το πρόγραμμα σημείωσε και μια άλλη πρωτοπορία, αφού εκατομμύρια άνθρωποι εγκατέστησαν το πρόγραμμα στους υπολογιστές τους για την επίτευξη ενός σκοπού, ο εθελοντισμός ήταν τεράστιος.

Ένα δεύτερο που αξίζει να σημειωθεί είναι το λεγόμενο το dnet γνωστό και ως distributed.net, το οποίο χρησιμοποιούσε χιλιάδες υπολογιστές για να σπάσουν κρυπτογραφημένους κωδικούς.

## **Στόχος του GRID**

Με την έννοια Grid computing εννοούμε ένα υπολογιστικό πρότυπο το οποίο μπορεί να παρομοιαστεί με ένα πολύπλεγμα ηλεκτρικής ισχύος. Με την διαφορά ότι παράγει επεξεργαστική ισχύ και όχι ηλεκτρική και μπορεί να την διαθέσει σε όποιον του ζητηθεί. Σήμερα μέσω internet μέρος του πολυπλέγματος μπορεί να γίνει κάθε υπολογιστής συνδεδεμένος στο διαδίκτυο και να διατεθεί η επεξεργαστική του ισχύς σε όλο τον κόσμο.

## **Τα οφέλη του GRID**

Στην ουσία το grid computing πρόκειται για έναν παγκόσμιο καταμερισμό εργασίας. Όταν για να ολοκληρωθεί μια ιατρική έρευνα από έναν μόνο υπολογιστή, απαιτούνται εκατοντάδες χρόνια μαθηματικών υπολογισμών, είναι φυσικό επακόλουθο πως η τεχνολογία θα πρέπει να δώσει κάποια άλλη λύση. Εκεί λοιπόν που ένας υπολογιστής θα έπρεπε να πραγματοποιήσει εκατομμύρια μαθηματικούς υπολογισμούς, είναι πολύ πιο εύκολο και λιγότερο χρονοβόρο για εκατομμύρια υπολογιστές να πραγματοποιήσουν μόνο έναν υπολογισμό. Η λογική του grid computing στηρίζεται στις αρχές των συνδέσεων peer-to-peer. Τα system resources ενός μεγάλου αριθμού υπολογιστών, ελέγχονται από έναν κεντρικό υπολογιστή και εργάζονται παράλληλα δημιουργώντας κατά συνέπεια έναν υπέρ-υπολογιστή. Το ένα χέρι νίβει το άλλο

Με την εποχή όπου "640 KB είναι όλη η μνήμη που θα χρειαστεί ποτέ μια εφαρμογή" (Μπιλ Γκέιτς 1981) να έχει περάσει ανεπιστρεπτί και τις έρευνες για το καλό της ανθρωπότητας (έρευνες κατά του ιού AIDS και του καρκίνου) να απαιτούν την επεξεργασία τεράστιων ποσοτήτων δεδομένων, η εφαρμογή του grid computing ίσως να αποτελεί τη μοναδική λύση. Σύμφωνα, μπορεί "ο δικός μου υπολογιστής Celeron με 128 MB μνήμης" όντως να μην κάνει τη μεγάλη διαφορά σε μια πολύπλοκη έρευνα, αλλά σίγουρα 1 εκατομμύριο τέτοιοι "φτωχοί" υπολογιστές θα βοηθήσουν πολύ. Αυτό είναι το πνεύμα του grid computing σε τελική ανάλυση.

## **Το GRID σήμερα**

Πολλές εταιρίες αξιοποιούν τα πλεονεκτήματα της σημερινής δομής του Grid, γιατί αν και σε ερευνητικό επίπεδο το Grid προσφέρει αποδοτικά οφέλη στους χρήστες του, χωρίς να περιορίζονται σε ένα σύστημα που θα χρειάζεται συνεχώς αναβαθμίσεις.

Τα βασικά προβλήματα των εταιριών είναι θέματα ασφάλειας, διαβάθμισης και ουσίας ως προς την ποιότητα των υπηρεσιών. Η λύση που χρειάζονται, οι συνδυασμένες συσκευές, που αποτελούν το πολύπλεγμο σε ένα επαγγελματικό Grid είναι σύμφωνα με τις προδιαγραφές του GGF (Global Grid Forum).

Χρησιμοποιώντας τα πρότυπα αυτά, οι συνδυασμένες συσκευές τους επιτρέπουν να κινηθούν σύμφωνα με την συμβατότητα με τις υπάρχουσες νέες τεχνολογίες Grid, αλλά και να υιοθετήσουν καινούριες καθώς αυτές εξελίσσονται, προσφέροντας νέα πλεονεκτήματα και ευκολίες.

Χρησιμοποιώντας στο έπακρο τις δυνατότητες των μηχανημάτων της μια εταιρία μπορεί να αποφύγει περιττά έξοδα. Αυτό επιτυγχάνετε με την χρήση περισσευόμενων πόρων κάθε υπολογιστή της όταν βρίσκετε σε άεργη κατάσταση στο δίκτυο της εταιρίας. Αυτή η περισσευόμενη ισχύς μπορεί να διανεμηθεί στις ανάγκες της εταιρίας ακόμη και αν βρίσκετε σε γεωγραφικά απομακρυσμένες περιοχές μέσω του διαδικτύου. Το πιο γνωστό στοιχείο που θα μπορούσε να εκμεταλλευτεί είναι από τους επιτραπέζιους υπολογιστές που χρησιμοποιούν μόνο το 10% της επεξεργαστικής του ισχύς. Ακόμα παραπέρα θα μπορούσε με τις απαραίτητες εφαρμογές αυτή η περισσευόμενη επεξεργαστική ισχύς να διανεμηθεί και σε άλλες εταιρίες και να λειτουργήσει σαν έσοδο για την κάθε εταιρία που δίνει επεξεργαστική ισχύ.

## **Τι προσφέρει το Grid**

Το κυριότερο που προσφέρει το Grid είναι η ταχύτητα των αποτελεσμάτων. Το πλεόνασμα επεξεργαστικής ισχύος που προκύπτει από μια Grid πλατφόρμα μπορεί να βοηθήσει έναν οργανισμό για την ταχύτερη εύρεση αποτελεσμάτων επιταχύνοντας έτσι τις έρευνες, όπως για παράδειγμα ιατρικές έρευνες που είναι ζωτικής σημασίας.

Το χαμηλό κόστος υπολογισμού είναι ακόμα ένα προτέρημα του Grid. Δίνει λύση σε εταιρίες που έχουν ανάγκη από επεξεργαστική ισχύ χωρίς να χρειαστεί να επενδύσουν μεγάλα χρηματικά ποσά σε υλικό (hardware), απλά αγοράζοντας επεξεργαστική ισχύ από άλλες εταιρίες που διαθέτουν τον ανάλογο εξοπλισμό και διανείμουν ισχύ.

Συνεπεία των παραπάνω είναι και ο προσανατολισμός των εταιριών για καλύτερα αποτελέσματα αφού πλέον η ταχύτητα και το κόστος δεν είναι τόσο σημαντικός παράγοντας. Πλέον οι εταιρίες αναθεωρούν τις ανάγκες τους και αναζητούν καλύτερα και ποιοτικότερα αποτελέσματα.

## **Επαγγελματικές υλοποιήσεις του GRID**

Αναμφίβολα οι τομείς που έχει ζήτηση το Grid είναι επαγγελματικοί και ερευνητικοί που έχουν τεράστιες ανάγκες από υπολογιστικές δυνατότητες.

Σκεπτόμενοι τις ανάγκες αυτών θα προσεγγίζουμε αυτήν την εργασία.

Βασισμένοι στους πόρους μιας εταιρίας όπου είναι οι άεργοι προσωπικοί υπολογιστές που λειτουργούν πάντα πολύ κάτω από τις δυνατότητές τους και στην παρουσία ενός cluster που μπορεί ενδεχομένως να υπάρχει συμβάλουν στην δημιουργία ενός εσωτερικού Grid, που θα παράγει επεξεργαστική ισχύ για την εταιρία ή το ίδρυμα ή ακόμα να είναι ένα μέσο εσόδων προς την εταιρία αν χρησιμοποιείτε από άλλους. Με αυτόν τον τρόπο θα ικανοποιεί τις ανάγκες της μια εταιρία και ακόμα και ανάγκες άλλων εταιριών, με τρόπο γρήγορο και ασφαλή.



Ήδη υπάρχουν αρκετές υλοποιήσεις για αποθήκευση δεδομένων που δίνουν πρόσβαση σε χρήστες να έχουν πρόσβαση σε αυτά όποτε και από όπου θέλει ο καθένας γρήγορα και με ασφάλεια.

### **Προϋποθέσεις για ένα Grid**

Για να λειτουργήσει αποδοτικά ένα Grid σε μια εταιρία που το χρησιμοποιεί για ωφέλιμους δικούς της σκοπούς ή για παροχή υπολογιστικής ισχύς σε άλλες εταιρίες πρέπει να πληρεί κάποιες προϋποθέσεις.

Το κυριότερο που πρέπει να έχει ακολουθήσει είναι τα πρότυπα του GGF (Global Grid Forum) που έχει αναλάβει την εποπτεία όλων των ερευνητικών προγραμμάτων. Αυτά τα πρότυπα είναι σημαντικότερα για την επέκταση του εκάστοτε Grid και την συνεργασία του με άλλα. Με αυτόν τον τρόπο είναι εύκολο να πραγματοποιηθούν εξελίξεις τα επόμενα χρόνια πάνω στα πολυπλέγματα και την ενσωμάτωση νέων τεχνολογιών για μεγιστοποίηση υπολογιστικών δυνατοτήτων. Παράλληλα μέσω διαφόρων άλλων διαδικασιών, αφού πρόκειται για εταιρίες, δίνει δυνατότητες διαχείρισης σε υπευθύνους υπαλλήλους εταιριών για χρέωση πελατών και προσθήκη εξωτερικών πολυπλεγμάτων. Τα παραπάνω πρέπει φυσικά να γίνονται αποτελεσματικά, ηλεκτρονικά και με απόλυτη ασφάλεια για να αποφευχθούν επιθέσεις από εξωτερικούς χρήστες ή κερδοσκόπους. Συνέπεια για την αποφυγή ανεπιθύμητων επιθέσεων είναι η ανάπτυξη του τομέα τη ασφάλειας. Σε ευαίσθητα συστήματα σαν τα πολυπλέγματα που κινούνται σημαντικά δεδομένα, οι ανταγωνιστικές εταιρίες πρέπει να είναι ασφαλή τα δεδομένα τους και οι εφαρμογές τους. Αυτό επιτυγχάνετε μέσω κάποιων μηχανισμών και την ταυτοποίηση των χρηστών που εξουσιοδοτεί ορισμένους συγκεκριμένους χρήστες ενώ παράλληλα να αποκλείει στους υπόλοιπους την πρόσβαση στις υπολογιστικές υπηρεσίες. Παράλληλα στους εξουσιοδοτημένους χρήστες υπάρχει το απόρρητο μεταξύ τους και απόλυτος διαμοιρασμός των δεδομένων τους. Σε αντίθετη περίπτωση αμφισβητείται η τεχνολογία και οι υπηρεσίες του πολυπλέγματος που παρέχονται και τελικά θα απορριφθεί από τον επαγγελματικό χώρο.

Επίσης τεράστια σημασία έχει η χρηστικότητα της εφαρμογής. Σε κάθε εταιρία το προσωπικό που απασχολείτε με την εφαρμογή του πολυπλέγματος θέλει μια εφαρμογή η οποία να μην χρειάζεται ιδιαίτερες γνώσεις υπολογιστών πέρα από τις βασικές. Έτσι η εφαρμογή πρέπει να είναι σταθερή, δηλαδή χωρίς πολλά προβλήματα που να χρειάζεται κάποιος ειδικός για την αποκατάσταση της κάθε φορά που δημιουργείτε κάποια βλάβη. Παράλληλα η εφαρμογή να μην έχει υπολογιστικές απαιτήσεις και έχει γνωστό και εύχρηστο περιβάλλον. Παραπέρα οι συνηθισμένες κινήσεις που γίνονται για την εκτέλεση μιας εργασίας να είναι απλοποιημένες και όσο το δυνατόν λιγότερο χρονοβόρες. Δηλαδή να υπάρχει κάποιος μηχανισμός που να υπολογίζει τις ανάγκες της κάθε διεργασίας και να προτείνει λύσεις.

## **Εισαγωγή στο GRIA**

Το GRIA (Grid Resources for Industrial Applications) δηλαδή Πόροι Πολυπλέγματος για Βιομηχανικές Εφαρμογές, είναι μια ερευνητική μελέτη για χρήση πολυπλεγμάτων σε εμπορικές και βιομηχανικές εφαρμογές. Αυτό που ξεχωρίζει το GRIA από τις υπόλοιπες μελέτες πάνω σε πολυπλέγματα είναι το ότι προορίζετε για επαγγελματική χρήση με συνέπεια την απλοποίηση της χρήσης του προγράμματος και το υψηλό επίπεδο ασφάλειας που πρέπει να έχει.

Βασικός στόχος είναι η μερική αυτοματοποίηση του web service GRIA έτσι ώστε να γίνονται ταχύτερα ευκολότερα και πιο αποδοτικά οι δουλειές που θα αποστέλλονται σε πολυπλέγματα. Με αυτόν τον τρόπο αποφεύγονται κάποιες κινήσεις και κάποιοι υπολογισμοί που ήταν ο χρήστης αναγκασμένος να κάνει με αποτέλεσμα να επιτυγχάνετε η απλοποίηση και επιταχύνετε η διαδικασία αποπεράτωσης μιας δουλειάς από την αρχή μέχρι το τέλος της.

## Επαγγελματική χρήση του GRIA

Οι βασικοί χρήστες του προγράμματος είναι επαγγελματίες που επιθυμούν ταχύτητα, ευχρηστία, αποδοτικότητα και ασφάλεια. Αυτοί επιθυμούν να αγοράζουν υπολογιστικούς πόρους ενώ σε άλλες περιπτώσεις να τους πουλάνε. Αυτοί οι πόροι μπορεί να είναι πολυπλέγματα γενικού σκοπού όπως ένα απλό cluster (συστοιχία υπολογιστών) ή οποιοδήποτε άλλο σύστημα παροχής υπολογιστικής ισχύος. Είναι πολύ κοινό παράδειγμα εταιρίες να χρησιμοποιούν καθημερινά εφαρμογές με ανάγκες υψηλής υπολογιστικής ισχύος και να τις ικανοποιούν με κάποιο είδος cluster το οποίο σε φορές περισσείας ισχύος να πωλείται η υπολογιστική ισχύ του σε εξωτερικούς πελάτες σαν υπηρεσία.

Οι φάσεις που περνάει το GRIA σε κάθε χρήση του από κάποιον πελάτη είναι ως εξής :

Αρχικά ο χρήστης ελέγχει και υπολογίζει το μέγεθος και το φόρτος εργασίας (workload) της δουλειάς που επιθυμεί να εκτελεστεί, έπειτα ελέγχει και τακτοποιεί τους διαθέσιμους πόρους που είναι κατάλληλοι για την συγκεκριμένη δουλειά, διαπραγματεύεται και εξασφαλίζει τους πόρους που επιλέγει για χρήση. Τέλος εκτελεί την υπηρεσία και παίρνει τα αποτελέσματα.

Όταν ένας πελάτης ελέγχει τους διαθέσιμους πόρους και διαπραγματεύεται, δεν τους δεσμεύει εκείνη την στιγμή και αυτό είναι ένα σημαντικό προτέρημα της σχεδίασης του συστήματος. Τα GRIA συστήματα δεν εκτελούνται στα ίδια τα clusters γιατί χρειάζονται απλά να έχουν κάποιες πληροφορίες για τους πόρους για να μπορούν να διαπραγματεύονται και να πραγματοποιούν συμφωνίες. Οι πληροφορίες που χρειάζονται τα συστήματα είναι η τιμή και η υπηρεσία που διατίθεται από τον κάθε πόρο.

Περιπτώσεις του GRIA είναι σε δύο εταιρίες που το χρησιμοποιούν για διαφορετικά είδη αποτελεσμάτων. Η CESI χρησιμοποιεί εφαρμογές δομικής ανάλυσης σε ενεργειακές μελέτες, ενώ η KINO χρησιμοποιεί εφαρμογές για κινηματογραφικές παραγωγές και λείανση (rendering) εικόνων. Και οι δυο

εταιρίες είναι πελάτες της Dolphin που είναι παροχέας υψηλού επιπέδου υπολογιστικού υλικού που εξυπηρετεί τις παραπάνω εταιρίες.

Περιπτώσεις όπως η προηγούμενη γεννάει την ανάγκη για ασφαλή και κρυπτογραφημένη μεταφορά αρχείων και πιστοποίηση και έλεγχος της των εμπλεκομένων.

### **Ποιότητα των Υπηρεσιών (QoS)**

Όταν αποστέλλετε μια εργασία με το GRIA ορίζονται και οι απαιτήσεις του πελάτη, όπως ο χρόνος εκτέλεσης, η τιμή που είναι διατεθειμένος να πληρώσει και η ποιότητα του αποτελέσματος που επιθυμεί. Αυτές οι απαιτήσεις ορίζουν την ποιότητα των υπηρεσιών του συστήματος. Έπειτα το σύστημα λαμβάνοντας υπ' όψη τις απαιτήσεις του πελάτη και τους διαθέσιμους πόρους διαπραγματεύεται με τον πελάτη έτσι ώστε να επιτευχθεί ένα αποδεκτό QoS.

Για να επιτευχθεί μια τέτοια συμφωνία είναι απαραίτητο να προβλεφθεί ο συνολικός χρόνος εκτέλεσης της εργασίας σε κάθε πόρο και η χωρητικότητα από την πλευρά του παροχέα. Ο υπολογισμός του φόρτου εργασίας (workload) γίνεται από την μεριά του πελάτη και δηλώνετε στο σύστημα μαζί με το μέγεθος της δουλειάς. Αυτό όμως μπορεί να απλοποιηθεί με μια εφαρμογή που ο πελάτης απλά θα πρέπει να δηλώνει την τοποθεσία της δουλειάς του στον υπολογιστή του και οι υπολογισμοί να γίνονται αυτόματα και στην συνέχεια αποστέλλονται στο σύστημα. Έτσι οι απαραίτητοι αυτοί υπολογισμοί θα γίνονται ταχύτερα και αποδοτικότερα χωρίς λάθη. Αργότερα κατά την διαπραγμάτευση του πελάτη και του παροχέα θα μπορούσαν να απλοποιηθούν κάποιες δημοφιλής επιλογές των πελατών όπως η φθηνότερη και η ταχύτερη αποπεράτωση μιας εργασίας.

Στην πρώτη περίπτωση της φθηνότερης επιλογής έχει δημιουργηθεί ένας έλεγχος κατά την διαπραγμάτευση για να επιστρέφεται μόνο το αποτέλεσμα με την χαμηλότερη τιμή.

Πιο πολύπλοκο είναι κατά την επιλογή της ταχύτερης αποπεράτωσης. Εκεί πρέπει να ληφθούν υπ' όψη οι υπολογιστικές δυνατότητες του κάθε πόρου και αποστέλλεται ανάλογος φόρτος εργασίας σε κάθε πόρο.

### **Το τρέχων GRIA**

Εξετάζοντας τις ανάγκες σε ένα επαγγελματικό Grid με εμπορικό σκοπό προκύπτουν ανάγκες για την υλοποίηση του και η σημερινή μορφή του που είναι χωρισμένο σε δύο επίπεδα, τον παροχέα και τον χρήστη. Ο παροχέας είναι κάθε οργανισμός που προσφέρει υπολογιστική ισχύ υψηλού επιπέδου για απαιτητικές εφαρμογές όπως προέχουμε πει. Οι χρήστες είναι οι πελάτες των παροχέων όπως οι εταιρίες που χρησιμοποιούν τις απαιτητικές εφαρμογές για εμπορικούς σκοπούς όπως δομικές μελέτες και εξομάλυνση εικόνων. Είναι σαφώς ότι η επικοινωνία των δύο μερών του GRIA, του χρήστη, πελάτη (client), και του παροχέα (service provider, supplier) πρέπει να γίνεται με καθορισμένο τρόπο, που ορίζει το GGF, που περιλαμβάνει ασφάλεια, αξιοπιστία και λειτουργικότητα. Εμείς όμως θα ασχοληθούμε με το μέρος του πελάτη και θα προσπαθήσουμε να γίνει ακόμα πιο λειτουργικό και ταχύτερο από το ήδη υπάρχων.

### **Επιλέγοντας την Java**

Η ανάπτυξη και η εξέλιξη των δύο εφαρμογών έγιναν με την γλώσσα προγραμματισμού την Java. Η επιλογή της ήταν καθαρά λόγω των πλεονεκτημάτων που παρουσιάζει σε διαδικτυακές εφαρμογές. Η Sun που είναι η κατασκευάστρια εταιρία της Java έχει παρουσιάσει αυτήν την γλώσσα προγραμματισμού σαν απλή, αντικειμενοστραφή, κατανεμημένη, ερμηνευόμενη, εύρωστη, ασφαλής, αρχιτεκτονικά ουδέτερη, μεταμορφομένη, υψηλής απόδοσης, πολυνηματική και ασφαλής γλώσσα. Η Java παρέχει ασφάλεια των εφαρμογών της και έχει δυνατότητες προστασίας του απορρήτου που εκμεταλλεύονται πλήρως από το GRIA.

Από την γέννηση της η Java δημιουργήθηκε για διαδικτυακές εφαρμογές και αυτό της δίνει εντυπωσιακό προβάδισμα απέναντι στις άλλες γλώσσες που η χρήση τους πάνω σε δίκτυα δεν γίνεται με τόσο απλό, γρήγορο και ασφαλή τρόπο. Ειδικά όταν δημιουργείτε μια επαγγελματική εφαρμογή με τις απαιτήσεις του GRIA είναι αναγκαία η χρήση της. Επίσης ο προγραμματισμός τριών επιπέδων επιτρέπει στην κάθε εφαρμογή να ελίσσεται και να μπορεί πολύ εύκολα να μετασχηματίζονται σε web services που είναι πολύ εύχρηστα απλά και ταχύτατα και αναγκαία σήμερα στις περισσότερες εταιρίες.

Επίσης πολύ βασικό προτέρημα της Java είναι η πολυνηματικότητα της που εκμεταλλεύεται από το GRIA αυτή της η ιδιότητα. Με αυτόν τον τρόπο και τα web services μπορεί να εκτελείτε το GRIA σε έναν υπολογιστή και πολλοί χρήστες μέσω δικτύου να εκμεταλλεύονται την εφαρμογή αυτή.

Απλουστεύεται ακόμα περισσότερο η Java αφού μπορεί κάθε χρήστης της να χρησιμοποιεί έτοιμες ρουτίνες, βιβλιοθήκες (libraries) που είναι δημιουργημένες ήδη και να προστίθενται σε αυτές καινούριες τεχνολογίες και εξελίξεις που είναι σημαντικά για Grid εφαρμογές αφού είναι ακόμα σε ανάπτυξη με σκοπό την βελτιστοποίηση του.

Το σημαντικότερο προτέρημα της Java είναι ότι είναι platform independent, δηλαδή κάθε εφαρμογή σε Java εκτελείτε σε κάθε υπολογιστή ανεξαρτήτου λειτουργικού συστήματος που είναι εγκατεστημένο αρκεί να υπάρχει το JRE (Java Realtime Enviroment) που περιέχει τον JVM (Java Virtual Machine) και τις βασικές βιβλιοθήκες τις Java. Συνεπώς κάθε εταιρία που θα εγκαταστήσει την εφαρμογή δεν χρειάζεται να αλλάξει το λειτουργικό της σύστημα αφού η εφαρμογή GRIA θα εκτελείτε στο περιβάλλον της Java.

## Παροχέας (Supplier)

Έχει δημιουργηθεί μια εφαρμογή που εκμεταλλεύεται τους πόρους ενός υπολογιστικού συστήματος για την εκτέλεση εργασιών που απαιτούν υψηλή υπολογιστική ισχύ. Ο παροχέας όπως και όλη η εφαρμογή GRIA είναι γραμμένη σε Java και μπορεί να εκτελεστεί σε οποιοδήποτε λειτουργικό σύστημα, ο παρόν παροχέας εκτελείτε σε Linux χωρίς αυτό να περιορίζει την εφαρμογή αλλά το συγκεκριμένο λειτουργικό σύστημα εκτός του ότι είναι δωρεάν και χρησιμοποιείτε ευρέως από την ερευνητική κοινότητα και μπορεί ένας έμπειρος χρήστης να το παραμετροποιήσει ανάλογα με τις απαιτήσεις της κάθε εφαρμογής. Η τρέχουσα έκδοση της GRIA είναι η δεύτερη πολύ βελτιωμένη σε σχέση με την πρώτη ενώ αναμένετε και η τρίτη έκδοση της.

Αυτή η εφαρμογή έχει δυνατότητες διαχείρισης φόρτου εργασίας καθώς και οικονομικών συναλλαγών, που προς το παρόν εκτελούνται εικονικά αφού οι υπηρεσίες είναι δωρεάν. Οι επικοινωνία αυτή γίνεται μέσω του πρωτοκόλλου SOAP (Simple Object Access Protocol)

Επειδή η εφαρμογή είναι ακόμα σε εξέλιξη οι υπηρεσίες που υποστηρίζονται είναι ακόμα ελάχιστες. Αρχικά δημιουργήθηκε μια δοκιμαστική υπηρεσία με το όνομα <http://gria.org/application/test> για να καλύψει ανάγκες έρευνας για την ομαλή αποστολή και λήψη εργασιών. Εκτός από την δοκιμαστική αυτή υπηρεσία έχουν δημιουργηθεί υπηρεσίες που χρησιμοποιούνται για μερική επαγγελματική χρήση όπως η υπηρεσία για εξομάλυνση εικόνων με το όνομα <http://gria.org/application/render> που είναι η πιο λειτουργική επαγγελματική λειτουργία αφού εταιρίες όπως η KINO αποστέλλει εργασίες παίρνοντας τα επιθυμητά αποτελέσματα. Πιο συγκεκριμένα στέλνει αρχεία με κατάληξη .rib και .bat τα οποία είναι σε ζευγάρια με το ίδιο όνομα, συμπίεζεται κάθε rib και bat με το ίδιο όνομα σε ένα αρχείο και αποστέλλεται ένα ένα. Η τελευταία υπηρεσία που υποστηρίζεται είναι με το όνομα <http://gria.org/application/india> που χρησιμοποιείτε από εταιρίες όπως η CESI για την δομική μελέτη αρχιτεκτονικών έργων, αλλά αυτή η υπηρεσία δεν υποστηρίζεται πλήρως.

## Πελάτης (Client)

Το δεύτερο μέρος της εφαρμογής είναι και αυτό που θα ασχοληθούμε στην παρούσα εργασία. Εμείς ασχοληθήκαμε με την έκδοση 2.0.2 ενώ τώρα βρίσκετε ήδη στην έκδοση 3.0. Η πλευρά του πελάτη είναι αυτή που έρχεται σε επαφή με όποιον θέλει να στείλει μια εργασία με υψηλές υπολογιστικές απαιτήσεις και είναι το σπουδαιότερο κομμάτι όλης της εφαρμογής. Η εφαρμογή αυτή έχει γραφτεί σε Java, με όλα τα προτερήματα που συνεπάγονται, και βρίσκετε στην δεύτερη έκδοσή της. Αποτελείται εκτός από τις βασικές βιβλιοθήκες της Java από 33 άλλες βιβλιοθήκες. Η σημαντικότερη για την εργασία αυτήν είναι το API της GRIA. Το γεγονός ότι είναι γραμμένη σε Java απαιτεί μόνο την παρουσία του JRE 1.4.2.04 στο μηχάνημα που θα εκτελείτε η εφαρμογή, ανεξαρτήτου λειτουργικού, αν και η παρούσα εργασία έχει υλοποιηθεί σε Windows XP. Η GRIA εκτελείται σε γραφικό περιβάλλον αλλά και σε γραμμές εντολών μέσα από DOS για παράδειγμα.

Πρέπει να σημειωθεί ότι η έκδοση JRE που χρησιμοποιήθηκε δεν είναι η τελευταία γιατί παρουσιαζόντουσαν προβλήματα με την 1.4.2.05, δικαιολογημένα γιατί κατά την βελτίωση της Java και τις νέες εκδόσεις κάποιες εντολές θυσιάζονται στον βωμό της ταχύτητας και της βελτίωσης, έτσι καταργούνται κάποιες και δημιουργούνται νέες. Αυτό συνεπάγεται τη βελτίωση της GRIA και σε πιο χαμηλό επίπεδο όπως είναι η γλώσσα προγραμματισμού της.

Αρχικά για να σταλεί μια εργασία προς εκτέλεση σε κάποιο πολύπλεγμα πρέπει να δημιουργηθούν κάποια certificates τα οποία τοποθετούνται σε τοπικό φάκελο του υπολογιστή που τρέχει η GRIA. Ο φάκελος με το όνομα conf περιέχει αρχεία τύπου xml που περιέχουν πληροφορίες για την λειτουργία της εφαρμογής. Πληροφορίες όπως την τοποθεσία των πιστοποιητικών βρίσκονται σε αυτόν τον φάκελο. Η χρήση των πιστοποιητικών κάνει την εφαρμογή περισσότερο ασφαλή από εξωτερικούς χρήστες που επιθυμούν να προσβάλουν την εφαρμογή και τον χρήστη της. Οι βιβλιοθήκες και τα πιστοποιητικά είναι τα βασικά που χρειάζεται η εφαρμογή του client για να τρέξει. Πιο συγκεκριμένα πρέπει οι βιβλιοθήκες, δηλαδή τα 33 αρχεία jar και ο περαιτέρω κώδικας να είναι στον φάκελο lib σε μορφή κλάσεων.



Μετά την εγκατάσταση της εφαρμογής ό κάθε χρήστης είναι έτοιμος να αποστείλει εργασία σε κάποιο πολύπλεγμα με την παρακάτω σειρά.

1. `gria open`
2. `gria tender`
3. `gria upload`
4. `gria run`
5. `gria download`
6. `gria finish`

Μπορούν επίσης να χρησιμοποιηθούν οποιαδήποτε στιγμή και οι παρακάτω εντολές για τον έλεγχο της ροής των εργασιών.

7. `gria show`
8. `gria browse`

Βέβαια στις καινούριες εκδόσεις έχουν προστεθεί και άλλες εντολές. Κάθε επικοινωνία μεταξύ του παροχέα και του πελάτη γράφεται σε ένα αρχείο με το όνομα `gria.state`. Αυτές οι επικοινωνίες είναι τεσσάρων ειδών `conversation`, οι οποίες σηματοδοτούν και μια διαφορετική εντολή του πελάτη στον παροχέα και αντίστροφα. Πιο ειδικά οι `conversations` είναι γραμμένες στον κώδικα της GRIA σε ξεχωριστές κλάσεις με τα ονόματα `AccountConversation`, `AllocationConversation`, `DataConversation` και `JobConversation`. Όπως θα καταλάβουμε καλύτερα παρακάτω αυτές οι κλάσεις μπορούν να κληθούν μόνο σειριακά, δηλαδή πρέπει πάντα να έχει κληθεί η προηγούμενη κλάση για να κληθεί και η επόμενη.

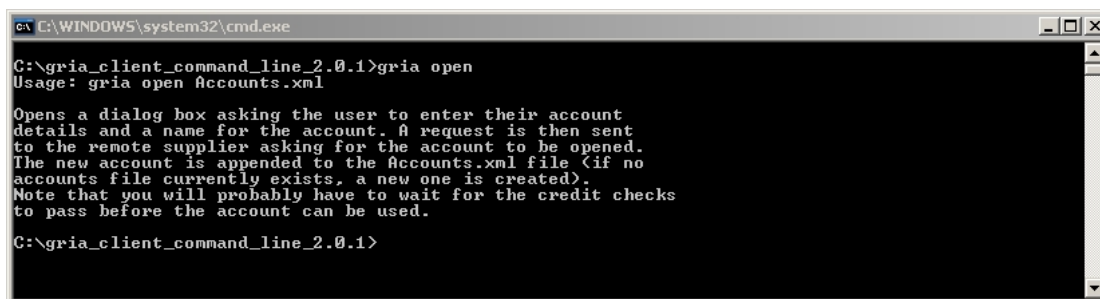
## **Χρήση του Client**

Παρακάτω θα αναλύσουμε πώς ένας χρήστης της GRIA μπορεί να αποστείλει μια εργασία σε ένα πολύπλεγμα και να λάβει τα αποτελέσματα. Σκοπός της ανάλυσης είναι να εντοπίσουμε τα σημεία που χρειάζονται βελτίωση έτσι ώστε κάθε χρήστης να μπορεί ευκολότερα και ταχύτερα να αποστέλλει μια εργασία σε ένα πολύπλεγμα. Όπως προέχουμε αναφέρει όλες οι κινήσεις πρέπει να γίνουν

σειριακά το οποίο καθιστά ευκολότερο στη χρήση του το πρόγραμμα. Η περιγραφή των βημάτων είναι η παρακάτω.

- **gria open**

Είναι η πρώτη εντολή που πρέπει να εκτελεστεί από τον χρήστη για να δημιουργήσει έναν λογαριασμό τον οποίο θα χρησιμοποιήσει για να εκτελέσει και τις υπόλοιπες λειτουργίες του client. Για να εκτελεστεί αυτή η εντολή ο χρήστης δίνει όλα του τα προσωπικά του στοιχεία καθώς επίσης και τον τρόπο χρέωσης των υπηρεσιών που παρέχει ο supplier και επιθυμεί να λάβει. Αυτό το αποτέλεσμα καταγράφεται στο αρχείο gria.state που σε περίπτωση πρώτης χρήσης του προγράμματος δημιουργείται εκείνη την στιγμή. Είναι αναγκαία η δημιουργία ενός account για την δυνατότητα μετάβασης στο επόμενο βήμα. Πρέπει να σημειωθεί ότι μπορεί ο κάθε χρήστης να έχει δημιουργήσει όσα accounts επιθυμεί σε όσους suppliers θέλει. Αυτό γίνεται με την εκτέλεση της εντολής gria open Account.xml όπου η τελευταία παράμετρος είναι για να ορίσει το όνομα του αρχείου που θα περιέχει της αναγκαίες πληροφορίες. Στην παρακάτω εικόνα φαίνεται πως απαιτείται να συντάσσεται η εντολή για την δημιουργία ενός account.



```
C:\WINDOWS\system32\cmd.exe
C:\gria_client_command_line_2.0.1>gria open
Usage: gria open Accounts.xml

Opens a dialog box asking the user to enter their account
details and a name for the account. A request is then sent
to the remote supplier asking for the account to be opened.
The new account is appended to the Accounts.xml file (if no
accounts file currently exists, a new one is created).
Note that you will probably have to wait for the credit checks
to pass before the account can be used.

C:\gria_client_command_line_2.0.1>
```

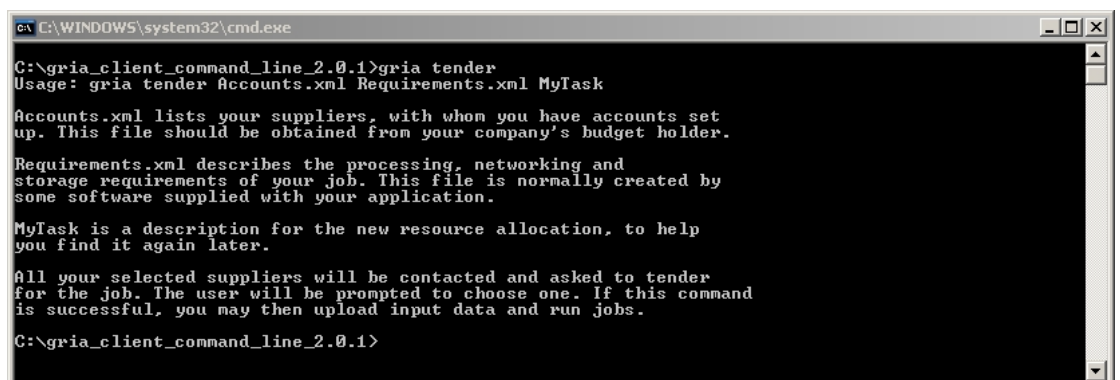
- **gria tender**

Το επόμενο βήμα μετά από την δημιουργία ενός account είναι να επικοινωνήσει με τον κάθε παροχέα που έχει account και να αποστείλει τα χαρακτηριστικά της δουλειάς που επιθυμεί να αποστείλει όπως επίσης και τα χρονικά περιθώρια για την εκτέλεση της εργασίας. Η απαιτούμενη μνήμη, ο χώρος, και οι ελάχιστοι δυνατοί πόροι που απαιτούνται είναι τα χαρακτηριστικά που αποστέλλει, ο φόρτος εργασίας και τέλος την υπηρεσία που επιθυμεί να χρησιμοποιήσει είναι τα στοιχεία που αποστέλλει κατά

εκτέλεση της εντολής tender. Όλα τα παραπάνω στοιχεία γράφονται σε ένα αρχείου τύπου xml με το όνομα Requirements.xml και έχει την παρακάτω μορφή.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <requirements xmlns:ns1=
"http://www.gria.org/org/gria/serviceprovider/commontypes/messaging"
max-num-concurrent-resources-per-resource-alloc="0" valid="true"
num-parallel-resources-per-job="0">
- <ns1:max-upload-data-volume>
<ns1:bytes>4986264</ns1:bytes>
</ns1:max-upload-data-volume>
- <ns1:max-download-data-volume>
<ns1:bytes>49862640</ns1:bytes>
</ns1:max-download-data-volume>
<resource-allocation-start>2005-03-09T15:25:46.352Z</resource-allocation-start>
<resource-allocation-end>2005-04-09T15:25:38.352Z</resource-allocation-end>
<service-name>http://gria.org/application/test</service-name>
- <ns1:max-store-data-volume>
<ns1:bytes>54848904</ns1:bytes>
</ns1:max-store-data-volume>
</requirements>
```

Οι παραπάνω υπολογισμοί είναι χρονοβόροι και εφ' όσον τους πραγματοποιεί κάποιος άνθρωπος είναι εύκολο να γίνουν και λάθη τα οποία κοστίζουν χρονικά αλλά, χρηματικά αλλά μπορεί επίσης αν γίνουν εσφαλμένα να μην οδηγήσουν καν στην αποπεράτωση της εργασίας.



```
C:\WINDOWS\system32\cmd.exe
C:\gria_client_command_line_2.0.1>gria tender
Usage: gria tender Accounts.xml Requirements.xml MyTask

Accounts.xml lists your suppliers, with whom you have accounts set
up. This file should be obtained from your company's budget holder.

Requirements.xml describes the processing, networking and
storage requirements of your job. This file is normally created by
some software supplied with your application.

MyTask is a description for the new resource allocation, to help
you find it again later.

All your selected suppliers will be contacted and asked to tender
for the job. The user will be prompted to choose one. If this command
is successful, you may then upload input data and run jobs.

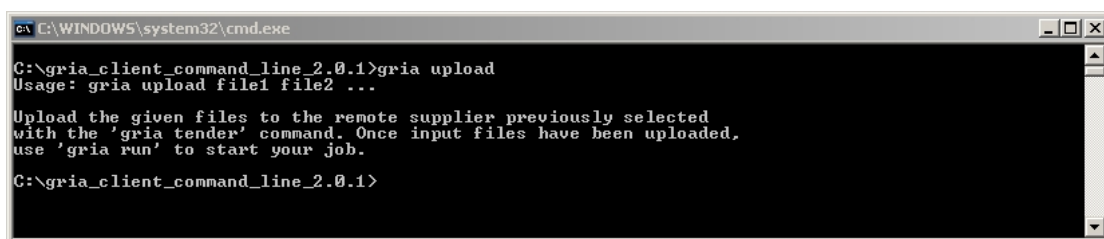
C:\gria_client_command_line_2.0.1>
```

Στην παραπάνω εικόνα φαίνεται πως πρέπει να εκτελείται η συγκεκριμένη εντολή. Πρέπει να δηλώνεται το xml αρχείο που περιέχει τους suppliers που έχει ανοικτεί κάποιο account. Έπειτα δηλώνεται το όνομα του xml αρχείου που περιέχει πληροφορίες για την εργασία που επιθυμείτε να αποσταλεί και τέλος ένα

όνομα που περιγράφει την εργασία. Στην συνέχεια ο κάθε παροχέας που έχει γίνει η επικοινωνία στέλνει με την σειρά του απάντηση με τα ανάλογα στοιχεία, όπως κόστος, χρόνος αποπεράτωσης και γενικότερα τις παρούσες δυνατότητες του.

- **gria upload**

Σαν συνέχεια των παραπάνω κινήσεων είναι η αποστολή των αρχείων που ευθυμεί ο χρήστης να επεξεργαστεί στον παροχέα που έχει ήδη επιλέξει. Κατά την εκτέλεση αυτής της εντολής δημιουργούνται data conversation που χρησιμοποιούνται σαν είσοδος για την εκτέλεση της εργασίας. Πρέπει ο χρήστης να συμπιέσει τα δεδομένα που επιθυμεί να αποστείλει για επεξεργασία και να τα στείλει με την εντολή `gria upload file1 file2 ...` όπου `file1` και `file2` είναι τα συμπιεσμένα αρχεία που αποστέλλει. Τα συμπιεσμένα αρχεία αυτά περιέχουν επίσης και πληροφορίες για το πώς θα επεξεργαστούν. Παρακάτω είναι ένα παράδειγμα για το upload.



```
C:\WINDOWS\system32\cmd.exe
C:\gria_client_command_line_2.0.1>gria upload
Usage: gria upload file1 file2 ...

Upload the given files to the remote supplier previously selected
with the 'gria tender' command. Once input files have been uploaded,
use 'gria run' to start your job.
C:\gria_client_command_line_2.0.1>
```

Πρέπει να τονιστεί ότι κατά την εκτέλεση αυτής της εντολής πρέπει ο χρήστης να ορίσει τα αρχεία που επιθυμεί να αποστείλει και σε περιπτώσεις που είναι ο αριθμός των συμπιεσμένων αρχείων μεγάλος, συνεπάγεται πολλές γραμμές που ορίζουν τα συμπιεσμένα αρχεία.

- **gria run**

Το επόμενο βήμα είναι η εντολή εκτέλεσης της εργασίας. Αυτό επιτυγχάνετε με την εντολή `gria run http://gria.org/application.test work.xml –input test.zip –output result.zip`. Οι παραπάνω παράμετροι είναι σημαντικοί για την εκτέλεση της εργασίας και ορίζουν τον τύπο της υπηρεσίας, πληροφορίες για την συγκεκριμένη εργασία και δήλωση των αρχείων εισόδου και ονομασία των αρχείων εξόδου ώστε να τα λάβουμε μετά την αποπεράτωση της εργασίας. Παρακάτω είναι ένα παράδειγμα για την εκτέλεση της εργασίας.

```
C:\WINDOWS\system32\cmd.exe
C:\gria_client_command_line_2.0.1>gria run
Usage: gria run http://gria.org/application/INDIA Work.xml --input file1 --input scene.dat --output
result

Run a job on a remote machine (selected using 'gria tender') using the
named application.
The input files must have been previously uploaded with 'gria upload'.
The outputs can then be downloaded with 'gria download'.

Work.xml is a file describing the processing requirements of this job.
It comes from the same place as the Requirements.xml file.
C:\gria_client_command_line_2.0.1>
```

Οι τελευταίες πληροφορίες που χρειάζονται αναγράφονται σε ένα αρχείου τύπου xml που έχει την παρακάτω μορφή και συνηθίζεται να ονομάζεται Work.xml.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <spec xmlns:ns1="http://www.gria.org/org/gria/serviceprovider/commontypes/messaging">
- <ns1:work>
    <ns1:std-CPU-seconds>5</ns1:std-CPU-seconds>
  </ns1:work>
- <ns1:min-physical-memory>
    <ns1:bytes>1000</ns1:bytes>
  </ns1:min-physical-memory>
- <ns1:max-output-volume>
    <ns1:bytes>1000</ns1:bytes>
  </ns1:max-output-volume>
    <ns1:num-processors>3</ns1:num-processors>
  <ns1:arguments>test.txt result.txt</ns1:arguments>
</spec>
```

- **gria download**

Το τελευταίο βήμα είναι η λήψη των δεδομένων που επεξεργαστήκαν από τον παροχέα που εστάλησαν, σύμφωνα με την ονομασία που δόθηκε για αποτέλεσμα κατά την εκτέλεση της εντολής gria run. Παρακάτω είναι ένα παράδειγμα της εντολής gria download.

```
C:\WINDOWS\system32\cmd.exe
C:\gria_client_command_line_2.0.1>gria download
Usage: gria download data [filename] ...

Download from the given data conversation on a remote supplier to the
named local file. If no local filename is given, one is created using
the name of the data conversation.

Data sets previously created using 'upload' or 'run' can be referred to
by their short name, but data sets created elsewhere must be imported
first, so that the system knows about them. Eg:

    gria import http://example.com/gria/dataservice#52 output.dat
    gria download output.dat
C:\gria_client_command_line_2.0.1>
```

- **gria finish**

Τελικά πρέπει να εκτελεστεί η εντολή finish για να σβηστούν πληροφορίες για την δουλειά η οποία τελείωσε και να ελευθερωθούν πόροι των συστημάτων. Οι πληροφορίες που σβήνονται είναι στο αρχείο gria.state.

- **gria show**

Η παρούσα εντολή μπορεί να χρησιμοποιηθεί ανά πάσα στιγμή και όποτε επιθυμεί ο κάθε χρήστης για να ελέγξει και να παρουσιάσει τις τρέχουσες εργασίες, λειτουργίες και λογαριασμούς που έχει χρησιμοποιήσει ή ενεργοποιήσει ο χρήστης στο παρελθόν και δεν τις έχει σβίσει, δηλαδή δεν έχει κάνει gria finish.

- **gria browse**

Η συγκεκριμένη εντολή είναι ίδια με την παραπάνω εντολή με την διαφορά ότι η gria browse εμφανίζει σε γραφικό περιβάλλον με την χρήση Java Swing τις εργασίες, λειτουργίες και λογαριασμούς που έχει χρησιμοποιήσει ο χρήστης και βρίσκονται σε εξέλιξη εκείνη την στιγμή. Παράλληλα στο παραθυρικό περιβάλλον μπορούν να χρησιμοποιηθούν εντολές όπως gria download και gria finish.

### **Δυσκολίες που παρουσιάζονται στο Client**

Από την εγκατάσταση μέχρι την χρήση της εφαρμογής παρουσιάστηκαν κάποια προβλήματα όπως παρουσιάζονται κατά την αναλυτική περιγραφή της εφαρμογής. Αρχικά παρουσιάστηκαν προβλήματα κατά την εγκατάσταση της εφαρμογής. Βέβαια υπήρχαν εγχειρίδια για την εγκατάσταση και εκτέλεση του GRIA αλλά ήταν ελλιπέστατα και χρειάστηκε να επικοινωνήσουμε με την ομάδα που είναι υπεύθυνη για την δημιουργία και ανάπτυξη της εφαρμογής, οι οποίοι μας κατατόπισαν για της ρυθμίσεις που έπρεπε να γίνουν για εγκατασταθεί και να λειτουργήσει η εφαρμογή. Το κυριότερο ήταν τα certificates, Έπρεπε τα αρχεία πιστοποίησης να τοποθετηθούν σε κάποιο μέρος του υπολογιστεί και να δηλωθούν σε ένα αρχείο τύπου conf που βρίσκετε μέσα στον φάκελο της εφαρμογής. Μετά από αυτήν την ρύθμιση έπρεπε να ορίσουμε και να δηλώσουμε

στο λειτουργικό μας σύστημα μεταβλητές και παραμέτρους για να επιτύχουμε την ομαλή λειτουργία της εφαρμογής. Έπειτα από της παραπάνω ρυθμίσεις και κάποιες άλλες όπως η καταγραφή των δραστηριοτήτων του χρήστη είμαστε έτοιμοι να τρέξουμε την εφαρμογή μας.

Μετά το πέρας των αρχικών ρυθμίσεων και εκτελώντας την εφαρμογή παρατηρήσαμε το περιβάλλον εκτέλεσης της. Το περιβάλλον γραμμής εντολών είναι πλέον κατά πολύ ξεπερασμένο και δεν συνιστάται για επαγγελματικές εφαρμογές. Μπορεί ένα τέτοιο περιβάλλον να εξυπηρετεί προχωρημένους χρήστες έτσι ώστε να έχουν καλύτερο έλεγχο και να αντιλαμβάνονται τυχόν λάθη αλλά για τους χρήστες τους οποίους προσδιορίζεται το πρόγραμμα δεν πλησιάζει καθόλου το ιδανικό. Σήμερα όπου όλες οι σύγχρονες επαγγελματικές και βιομηχανικές εφαρμογές τρέχουν σε παραθυρικά περιβάλλοντα όπως επίσης και τα περισσότερα λειτουργικά συστήματα ο κάθε χρήστης που δεν είναι τόσο εξειδικευμένος και έμπειρος χρειάζεται ένα περιβάλλον οικείο και εύκολο στην χρήση του. Αν δεν αλλάξει η παρούσα κατάσταση του GRIA client είναι σχεδόν βέβαιο ότι το project οδηγείτε σε αποτυχία διότι πολύ απλά σε πολύ ελάχιστο χρονικό διάστημα μπορεί να δημιουργηθεί εφαρμογή που πιο εύκολο και οικείο interface.

Επόμενο μειονέκτημα της εφαρμογής και ίσως το πιο σημαντικό είναι ότι πρέπει ο κάθε χρήστης που επιθυμεί να αποστείλει μια εργασία σε ένα cluster πρέπει να συντάξει δύο αρχεία τύπου xml τα οποία ως γνωστό είναι λίγο ιδιότροπα στην σύνταξή τους. Τα συγκεκριμένα αρχεία περιέχουν πληροφορίες πολύτιμες όπως είναι ημερομηνίες και ώρες για την έναρξη και αποπεράτωση της εργασίας, απαιτούμενο μέγεθος που θα χρειαστεί για την επεξεργασία του, η ζητούμενη υπηρεσία καθώς επίσης το φόρτος εργασίας και η ποιότητα του αποτελέσματος. Είναι εύκολα κατανοητό ότι με το παραμικρό λάθος και ενώ ο χρήστης θα πιστεύει ότι έχει δημιουργήσει τα xml σωστά να μην τελειώσει η εργασία μέσα στον απαιτούμενο χρόνο, να τελειώσει με λάθος αποτελέσματα, να μην τελειώσει ποτέ ή ακόμα η χρέωση που θα γίνει στον χρήστη να μην είναι η σωστή.

Εκτός από τα συντακτικά λάθη που μπορούν να δημιουργηθούν υπάρχει περίπτωση ο χρήστης να μην υπολογίσει σωστά το φόρτος εργασίας, του οποίου ο

υπολογισμός είναι πολύ σημαντικός και δεν υπάρχει ενσωματωμένος τρόπος υπολογισμού του, με αποτέλεσμα να έχουμε πάλι λάθος εκτίμηση της εργασίας με συνέπεια και καθυστερημένα αποτελέσματα σε λάθος χρόνο ή υπερβολική χρέωση του χρήστη από τον παροχέα που έχει επιλέξει. Θα πρέπει δηλαδή να ενσωματωθεί μια μικρή εφαρμογή που θα υπολογίζει το φόρτος εργασίας, workload κατά προσέγγιση αυτόματα σύμφωνα με μαθηματικούς υπολογισμούς.

Άξιο αναφοράς είναι επίσης το γεγονός ότι σε κάποιες περιπτώσεις σφάλματος ακόμα και σε λάθη που απλά υπήρχε έλλειψη δικτύου οι γραμμές λάθους που εμφάνιζε ήταν υπερβολικά πολλές και πολύ δύσκολα να εντοπιστεί το λάθος βάση του αποτελέσματος, του excerption, που έβγαζε και κάποιος μη έμπειρος χρήστης και χωρίς εξοικείωση σε περιβάλλοντα Grid δεν μπορεί να εντοπίσει το λάθος.

Ίσως το σημαντικότερο απ' όλα, αν λάβουμε υπ' όψη το είδος των πελατών που είναι επαγγελματίες που ζητούν ευχρηστία, αξιοπιστία και χαμηλό κόστος, είναι το ότι στην συγκεκριμένη περίπτωση θα πρέπει η εφαρμογή του client να εγκατασταθεί σε όλους τους υπολογιστές των χρηστών που θα έχουν πρόσβαση και υποχρέωση αποστολής εργασιών σε παροχής. Αυτό σημαίνει ότι εκτός από την δυσκολία εγκατάστασης της εφαρμογής πρέπει να εκπαιδευτούν οι χρήστες στην συγκεκριμένη εφαρμογή του client. Με προέκταση του προβλήματος σε ανάγκες για περισσότερα πιστοποιητικά και έλλειψη του ελέγχου των κινήσεων του κάθε χρήστη από την ίδια την εταιρία. Επίσης αφήνετε στην κρίση του κάθε υπαλλήλου την επιλογή του παροχέα που επιλέγει για την εκτέλεση μιας εργασίας.

### **Λύσεις των μειονεκτημάτων**

Εκτός από τα παραπάνω προβλήματα που αναφέραμε και αναλύσαμε παραπάνω υπάρχουν και άλλα προβλήματα που η εξάλειψή ή η βελτίωση τους μπορούν να οδηγήσουν στην βελτιστοποίηση της εφαρμογής και την εδραίωση της στους επαγγελματικούς χώρους. Ένα παράδειγμα είναι ότι οι παροχείς δεν είναι όλοι γνωστοί στους πελάτες, δεν υπάρχει πουθενά κάποιος τρόπος που να έχει τους



παροχείς έτσι ώστε οι πελάτες να μπορούν να επιλέξουν μέσα από μεγάλη γκάμα τον καλύτερο που τους ταιριάζει, σαν αποτέλεσμα οι πελάτες και οι χρήστες της κάθε εφαρμογής. Αυτό γεννά την ανάγκη για δημιουργία ενδιάμεσων εταιριών που θα γνωρίζουν όλους τους παροχείς με τις αναγκαίες πληροφορίες του κάθε παροχέα.

Είναι πολύ εύκολο να κατανοηθεί γιατί κάποιος ενδιάμεσος μεσολαβητής είναι αναγκαίος αν σκεφτούμε ότι υπάρχουν πολλά παρόμοια παραδείγματα σε άλλους τομείς της ζωής μας που ήδη είναι σε λειτουργία, όπως οι διαμεσολαβητές των εισιτηρίων, οι πράκτορες. Αυτοί είναι ο ενδιάμεσος κρίκος μεταξύ των πελατών και των εταιριών που εκτελούνε ταξίδια, όλων των ειδών, από ακτοπλοϊκά έως και αεροπορικά. Η δουλειά τους είναι όταν ένας πελάτης τους ζητά κάποιο εισιτήριο για κάποιον προορισμό να τον προμηθεύουν με πληροφορίες για τον τρόπο, το κόστος, και την ποιότητα μεταφοράς τους και τέλος με επιλογή του πελάτη να γίνετε η αγορά του εισιτηρίου μέσω του πράκτορα ο οποίος πληρώνετε βάση των εισιτηρίων που πουλάει. Αλλιώς ο καθένας πελάτη θα ήταν αναγκασμένος να πηγαίνει σε κάθε ναυτιλιακή ή αεροπορική εταιρία και να ζητά πληροφορίες για να αγοράσει ένα εισιτήριο, πράγμα χρονοβόρο και αντιοικονομικό και για τον πελάτη αλλά και για τις εταιρίες που θα αναγκάζονταν να προσλάβουν περισσότερο προσωπικό για την εξυπηρέτηση των πελατών. Έτσι για να επανέλθουμε στην εργασία μας, ένα ενδιάμεσο πρόγραμμα θα κάνει την δουλειά του πράκτορα που θα έχει πληροφορίες για όλους τους παροχείς και θα μπορεί να τις προμηθεύει σε κάθε πελάτη που τις αναζητά, χωρίς να είναι αναγκασμένος ο πελάτης να ελέγχει έναν έναν, αν τους γνωρίζει όλους, τους παροχείς για να βρει τον καταλληλότερο για την εκτέλεση της εργασίας του.

Όπως αναφέρθηκε πριν το περιβάλλον που χρησιμοποιείτε για την αποστολή μιας εργασίας είναι ξεπερασμένο και καθόλου φιλικό για τον χρήστη, user friendly. Έτσι πρέπει να σχεδιαστεί ένα άλλο περιβάλλον πιο φιλικό, απλοϊκό και εύκολο τρόπο για τις αποστολές των εργασιών του κάθε χρήστη με ένα interface που θα βοηθούσε και θα διευκόλυνε την δουλειά. Ένας τρόπος για να ξεφύγουμε από command line χωρίς να επιβαρύνουμε τους υπολογιστές με Java Swings, να χρησιμοποιείτε το πρόγραμμα σε τοπικά δίκτυα και χωρίς εγκατάσταση του σε κάθε υπολογιστή που πρέπει να τρέχει είναι το internet.

Το πρόγραμμα του μεσολαβητή θα έχει δυνατότητες scheduling που με αυτόν τον τρόπο ο κάθε χρήστης θα μπορεί να χρησιμοποιεί και αυτόματα να επιλέγονται οι παροχές ή οι παροχές που επιθυμεί για να κάνει την δουλειά του είτε γρήγορα είτε φθηνά χωρίς να μπαίνει σε χρονοβόρες και πολύπλοκες διαδικασίες που χρειάζονται αν πρέπει να σπάσει την δουλειά και να την στείλει σε πολλούς παροχείς. Αυτοί οι αυτοματοποιημένοι τρόποι εξασφαλίζουν γρηγορότερο, ευκολότερο και πιο σίγουρο μέσο για την διεκπεραίωση των εργασιών κάθε χρήστη.

Παράλληλα πολύ καθοριστική είναι η χρήση υπολογισμού του φόρτου εργασίας, ένα εργαλείο που ονομάζεται workload estimator και υπολογίζει το φόρτο κάθε εργασίας που επιθυμεί να στείλει κάποιος χρήστη σε παροχέα χωρίς να το στείλει. Ο μεσολαβητής υπολογίζει το workload βάσει ενός αλγορίθμου που θα αναλύσουμε παρακάτω. Χρήσιμο και καθοριστικό εργαλείο αν σκεφτούμε ότι βάση αυτού υπολογίζεται ο χρόνος διεκπεραίωσης μιας εργασίας και το κόστος της.

Δεν πρέπει να αμελήσουμε το σπουδαιότερο ζήτημα που είναι η ασφάλεια των δεδομένων της κάθε εταιρίας. Αυτά τα δεδομένα που μεταφέρονται μέσω του διαδικτύου σε εξυπηρετητές και παροχείς είναι περιουσία της κάθε εταιρίας. Μέσω των πρωτοκόλλων SOAP που καθιστούν τις μεταφορές ασφαλείς από άλλες ανταγωνιστικές εταιρίες και κακόβουλων χρηστών του διαδικτύου που οι επιθέσεις τους τα τελευταία χρόνια είναι πολλές. Εκτός από το πρωτόκολλο SOAP υπάρχουν και τα πιστοποιητικά ασφαλείας που τοποθετούν το GRIA στην κορφή σε σχέση με άλλα προγράμματα Grid.

### **Εξελίξεις της καινούριας εφαρμογής**

Μετά από την εκτενή αναφορά των προβλημάτων που έχει ήδη υπάρχουσα εφαρμογή, προκύπτουν δύο τρόποι για την υλοποίηση και βελτίωση της. Η πρώτη σκέψη θα μπορούσε να είναι ένας διαμεσολαβητής σαν μια άλλη ξεχωριστή εταιρία όπου κάθε εταιρία χρήστης θα συνδεόταν μαζί του μέσω internet, και η

άλλη υλοποίηση της είναι να βρίσκετε μέσα στην εταιρία χρήστη και ο κάθε υπάλληλος της να επικοινωνεί με τον διαμεσολαβητή μέσω του τοπικού δικτύου που τώρα πια όλες οι εταιρίες έχουν. Η εφαρμογή του διαμεσολαβητή θα είναι σε έναν υπολογιστή εξυπηρετητή που θα έχει εγκατεστημένη την εφαρμογή client της GRIA και θα τρέχει μόνο εκεί και όχι σε κάθε υπολογιστή που θα χρησιμοποιούν την GRIA.

Η παρούσα έκδοση του GRIA σαν web service μπορεί να λειτουργήσει και με τους δύο τρόπους. Είναι μια εφαρμογή που μπορεί να τρέξει και σε ένα τοπικό δίκτυο αλλά και σε έναν απομακρυσμένο εξυπηρετητή. Η εφαρμογή αυτή μπορεί να τρέξει και να εξομοιώσει το GRIA client σε παραθυρικό περιβάλλον χωρίς καμιά εγκατάσταση σε κάποιον υπολογιστή εκτός από τον εξυπηρετητή φυσικά, που θα τρέχει την εφαρμογή.

Η εφαρμογή GRIA web service εξομοιώνει πλήρως τον client αλλά και πάλι χρειάζεται βελτιώσεις. Κάθε χρήστης μπορεί να μπει στην εφαρμογή χρησιμοποιώντας ένα όνομα χρήστη και έναν κωδικό και ανάλογα τα δικαιώματα του να μπορεί να ανοίξει λογαριασμό σε κάποιον παροχέα ή ακόμα να εισάγει καινούριους χρήστες που θα μπορούν να χρησιμοποιούν το GRIA client. Το περιβάλλον είναι φιλικό στον χρήστη, αφού πλέον οι περισσότεροι είναι αξιωμένοι με το περιβάλλον του internet και πλέον υπάρχει αλληλεπίδραση μεταξύ του χρήστη και της εφαρμογής.

Ανακεφαλαιώνοντας ο μεσολαβητής έχει κύρια χαρακτηριστικά όπως αναφέραμε με ιδιαίτερη έμφαση στην ευκολία χρήσης και λειτουργικότητας του GRIA client, επίσης πρόσβαση απ' όλους τους χρήστες με έμφαση στην ασφάλεια και διαχωρισμό των κινήσεων που επιτρέπεται να κάνει ο κάθε χρήστης, καθώς και έλεγχο του καθένα αλλά και έλεγχο των κινήσεων προς τους παροχείς. Με αυτόν τον τρόπο μπορούν πολύ εύκολα να γίνετε συλλογή στατιστικών και χρήση τους για ενδοεταιρικούς σκοπούς.

Τελικά ο μεσολαβητής επικεντρώθηκε στην αποδέσμευση του τελικού χρήστη από το GRIA client και αντικατέστησε την δύσκολη εγκατάσταση και το δύσχρηστο περιβάλλον από μια εφαρμογή προσιτή και εύχρηστη από τον χρήστη

και χωρίς να χρειάζεται εγκατάσταση. Αν αναλογιστούμε τις ήδη υπάρχουσες υλοποιήσεις σε Grid εφαρμογές, παρατηρούμε ότι οι μέχρι τώρα εφαρμογές προσανατολιζόντουσαν για ερευνητικούς και εκπαιδευτικούς σκοπούς με αποτέλεσμα χρήση τους μόνο από έμπειρους χρήστες. Το GRIA επιδίωξε να ξεπεράσει το ερευνητικό επίπεδο και να φτάσει να χρησιμοποιείται από απλούς χρήστες, όπου ως ένα σημείο αυτό το πέτυχε. Αν εξαιρέσουμε το αντιαισθητικό και δύσχρηστο command line, η εφαρμογή είναι πλήρης για να μπορεί ένας έμπειρος χρήστης να αποστείλει μια εργασία που επιθυμεί και να λάβει πίσω τα αποτελέσματα. Αυτό ξεπεράστηκε με την χρήση του μεσολαβητή που δημιουργήθηκε ένα παραθυρικό περιβάλλον πιο λειτουργικό και ευκολότερο στην χρήση του από το command line. Ακόμα και όλες οι διαδικασίες που έπρεπε να ακολουθήσει κάποιος για μια εργασία, γίνονται μέσω μεσολαβητή με εύκολο και γρήγορο τρόπο, χωρίς να γράφει ο χρήστης γραμμές εντολών και χωρίς να γράφει xml. Ένα σημαντικό μειονέκτημα του GRIA client είναι και η δημιουργία των xml που όπως είπαμε είναι πολύ εύκολο να κάνει κάποιος λάθος και να δημιουργήσει προβλήματα στην εταιρία και στον εαυτό του. Πλέον τα xml δημιουργούνται αυτόματα με χρήση τρόπων αλληλεπίδρασης του χρήστη και του προγράμματος. Ο χρήστης συμπληρώνει φόρμες με στοιχεία και χαρακτηριστικά της δουλειάς με σκοπό την δημιουργία των xml αυτόματα χωρίς συντακτικά λάθη τουλάχιστον.

Μια ακόμα σημαντική παράμετρος του μεσολαβητή είναι και ο τρόπος που έχουν πρόσβαση οι χρήστες στην εφαρμογή. Όπως είπαμε η εφαρμογή GRIA client αποτελείται από 33 jar αρχεία που είναι και το API της εφαρμογής. Άρα η εγκατάσταση του GRIA client δεν είναι υποχρεωτική σε όλους τους υπολογιστές, αρκεί λοιπόν να τρέχει η GRIA client σε έναν κεντρικό υπολογιστή εξυπηρετητή που μπορούν όλοι οι χρήστες να έχουν πρόσβαση μέσα από την εταιρία ή ακόμα και σε μερικές περιπτώσεις να έχουν πρόσβαση και εκτός εταιρίας. Αυτό θα γίνεται μέσω του τοπικού δικτύου που θα έχει η εταιρία χωρίς επιπλέον απαιτήσεις. Με συνέπεια να φαίνεται η πιο λειτουργική λύση να είναι η μετατροπή του GRIA client σε web service με δυνατότητες πρόσβασης στην εφαρμογή και μέσα από τοπικό δίκτυο αλλά και από κάποιον απομακρυσμένο υπολογιστή εκτός εταιρίας. Έτσι κάθε χρήστης μπορεί να χρησιμοποιεί τον μεσολαβητή για να αποστέλλει και να λαμβάνει εργασίες σε ή από κάποιον παροχέα.

Τελευταία αλλά καθόλου αδικημένα το θέμα ασφαλείας είναι πολύ σημαντικό για χρήση του μεσολαβητή από εταιρίες. Αφού ο μεσολαβητής δημιουργείται βάση του GRIA client δεν μπορεί παρά να πληρεί τα χαρακτηριστικά ασφαλείας για να μπορεί να λειτουργήσει και να χρησιμοποιηθεί από εταιρίες με επιτυχία και ασφάλεια από κακόβουλες επιθέσεις εξωτερικών χρηστών ή ακόμα και από ανταγωνιστικές εταιρίες.

Παράλληλα το GRIA έχει δώσει μεγάλη έμφαση στην προτυποποίηση και ενσωμάτωση νέων χαρακτηριστικών και τεχνολογιών που το κάνουν αποδεκτό και δίνουν την δυνατότητα να αλληλεπιδρά με κάθε είδους άλλες τεχνολογίες. Όπως αναφέραμε πριν για τον προγραμματισμό τριών επιπέδων μπορεί να είναι συμβατό με μεταγενέστερες τεχνολογίες και πρότυπα. Είναι ένα σημαντικό στοιχείο που πρέπει να το έχει ο μεσολαβητής γιατί οι εκδόσεις του GRIA αλλάζουν ταχύτατα και δύσκολα μπορεί ακολουθήσει. Δυνατότητες όπως η δρομολόγηση και ο υπολογισμός φόρτου εργασίας δεν εξαρτώνται από το API του GRIA άρα και από της αλλαγές που γίνονται σε κάθε έκδοση.

Η υπάρχουσα υλοποίηση του μεσολαβητή έχει δυνατότητα να ορίζονται από κάποιον χρήστη οι δυνατές κινήσεις άλλων χρηστών, σε αντίθεση με το GRIA client που μπορούσε οποιοσδήποτε χρήστης να εκτελέσει όποια εντολή ήθελε μέσα από το command line. Με αυτόν τον τρόπο οι χρήστες περιορίζονται και δημιουργούνται ομάδες χρηστών που μπορούν να εκτελέσουν ορισμένες εντολές του GRIA. Υπάρχει βέβαια μια ομάδα χρηστών που μπορούν να δίνουν δικαιώματα ή να τα αναιρούν. Πιο συγκεκριμένα υπάρχουν χρήστες τύπου διαχειριστές που μπορούν να εκτελέσουν οποιαδήποτε εντολή του GRIA μέσω του μεσολαβητή και επίσης να προσθέσουν ή να διαγράψουν καινούριους χρήστες και ακόμα να τους ρυθμίσουν τα δικαιώματά τους που μπορεί να είναι από δικαιώματα διαχειριστή μέχρι και χωρίς κανένα δικαίωμα αλλά να μπορεί να επιβλέπει κινήσεις. Έχει επιλεχθεί επίσης η δημιουργία νέων accounts, σε καινούριους παροχείς, να γίνετε μόνο από διαχειριστές όπως επίσης και η απομάκρυνση από την λίστα των παροχέων αν τελικά είναι ασύμφοροι.

Επίσης ένα ακόμα μειονέκτημα που εξαλείφθηκε με τον μεσολαβητή είναι το logging. Αυτή η δυνατότητα δεν υπάρχει στο GRIA και έπρεπε ο κάθε χρήστης να σημειώνει μόνος του τιμές, παροχές, υπηρεσίες και ότι άλλο τον ενδιέφερε. Ακόμα δεν μπορούσε να κρατηθεί ιστορικό κινήσεων άρα ούτε στατιστικά για τις κινήσεις. Ο μεσολαβητής αποθηκεύει κάθε κίνηση κάθε χρήστη σε μια βάση δεδομένων που μπορούν να έχουν πρόσβαση μόνο οι διαχειριστές.

Ένα ακόμα προτέρημα του μεσολαβητή σε σχέση με την εφαρμογή GRIA, είναι ότι αν και πλέον εκτελούν τις ίδιες εντολές, με μεγάλη διαφορά στην επαφή με τον χρήστη είναι ότι ο μεσολαβητής δεν χρειάζεται εγκατάσταση πέρα από την αναγκαία ύπαρξη ενός internet browser. Και εφ' όσον ο μεσολαβητής τρέχει δικτυακά δεν εξαρτάται από το λειτουργικό σύστημα του κάθε χρήστη.

## **Ο μεσολαβητής**

Όπως αναφέραμε παραπάνω ο μεσολαβητής είναι ένα web service που επικοινωνεί με το API του GRIA, και έχει γραφτεί σε Java και πιο συγκεκριμένα σε Java server pages, JSP. Η χρήση της τεχνολογίας jsp επιλέχθηκε μεταξύ και των τεχνολογιών asp και php για την αιτία ότι και το client μέρος του GRIA όπως και ολόκληρο το πρόγραμμα είναι γραμμένο σε Java και έτσι η επικοινωνία του μεσολαβητή και του API της GRIA θα γινόταν πολύ ευκολότερα.

Για δημιουργηθεί λοιπόν η εφαρμογή του μεσολαβητή δεν χρειάζεται να αλλαχθεί το παραμικρό στον κώδικα που ήδη υπήρχε. Έπρεπε μόνο να κατανοηθεί πως λειτουργεί σε επίπεδο API. Σημαντική βοήθεια είναι ένα πλήρες JavaDoc, δηλαδή μια αναλυτική περιγραφή του κώδικα, έτσι ώστε εύκολα μπορούσε να κατανοηθεί ο κώδικας και τρόπο πρέπει να το χρησιμοποιήσουμε κατάλληλα.

Καθώς το GRIA είναι γραμμένο σε Java έχει διακριτά στοιχεία τα οποία το ένα κληρονομεί το άλλο. Με το API που δημιουργείτε εκτελούνται οι εντολές του μέσω των jsp. Κάθε jar αρχείο από τα 33 που έχει το API περιέχει δεκάδες κλάσεις και μπορεί να χρησιμοποιηθεί μεμονωμένα με συνέπεια να δημιουργηθεί

ο μεσολαβητής χωρίς να αλλάξει τίποτα από το ήδη κώδικα της GRIA. Η μοναδική προϋπόθεση είναι να υπάρχει εγκατεστημένο στο μηχάνημα που θα εκτελείτε ο μεσολαβητής το JRE (Java Runtime Environment) για την λειτουργία του. Αξιοποιώντας πλήρως τα πλεονεκτήματα του αντικειμενοστραφούς προγραμματισμού δημιουργούνται νέες κλάσεις που καλούν άλλες κλάσεις από το API της GRIA για τον χειρισμό αρχείων xml όπως και για την επικοινωνία μέσω πρωτοκόλλου SOAP.

Με την χρήση των υπαρχουσών κλάσεων και την ενσωμάτωση καινούριων δημιουργήθηκε μια καινούρια εφαρμογή, ο μεσολαβητής, με καινούρια στοιχεία και με απόλυτη συμβατότητα, εφόσον τα στοιχεία κληρονομούνται από τις βιβλιοθήκες της GRIA.

Αν και η σωστή χρήση των στοιχείων δεν προεξοφλούσε την επιτυχία μιας σωστής λειτουργικότητας, χρειαζόταν σωστή σχεδίαση πριν την υλοποίηση. Χρειάστηκε τα στοιχεία του client να προσαρμοστούν με τέτοιο τρόπο έτσι ώστε μαζί με νέα ώστε να υπάρχει εύκολη λειτουργία, σε εξυπηρετητές, ασφαλή επικοινωνία και ελεγχόμενη.

Ο μεσολαβητής μπορεί να στηθεί σε έναν εξυπηρετητή συνδεδεμένο στο internet έτσι οποιοσδήποτε με πρόσβαση στο διαδίκτυο θα μπορεί να τον χρησιμοποιήσει ή να υπάρχει μέσα στην ίδια εταιρία που θα χρησιμοποιούν τον client σε κάποιον εξυπηρετητή της εταιρίας και να λειτουργεί σαν πύλη (gateway) στον οποίο θα στέλνουν οι χρήστες τις εργασίες τους. Και στις δυο περιπτώσεις το μόνο που θα χρειάζονται οι χρήστες είναι ένας δικτυακός πλοηγός (web browser) εγκατεστημένος στους υπολογιστές τους, κάτι που όλα τα λειτουργικά συστήματα έχουν.

Στην πρώτη περίπτωση που ο μεσολαβητής είναι σε κάποιον εξυπηρετητή στο διαδίκτυο, κάθε εταιρία θα έχει και έναν διαχειριστή που θα μπορεί να ορίζει τους παροχείς και να προσθαφαιρεί χρήστες. Στην άλλη περίπτωση που ο μεσολαβητής είναι στον χώρο της εταιρίας θα μπορούν να υπάρχουν πάνω από ένας διαχειριστής και να κατανέμετε η δουλειά σε αυτούς για την προσθαφαίρεση χρηστών και παροχέων.

Το σημαντικότερο όμως όλων είναι ότι ο client δεν χρειάζεται να είναι εγκατεστημένος σε κανέναν υπολογιστή παρά μόνο σε αυτόν που θα είναι ο μεσολαβητής και οποιοσδήποτε επιθυμεί να στείλει κάποια εργασία θα το κάνει μέσω του πλοηγού του, χωρίς καμία υποχρέωση για εγκατάσταση κάποιου λογισμικού πέρα του πλοηγού.

### **Ελλείψεις του μεσολαβητή**

Αν και υπάρχουν πολλά στοιχεία που βελτιώνουν τον GRIA client υπάρχουν και άλλα στοιχεία που μπορούν πολύ εύκολα να ενσωματωθούν με την ήδη υπάρχουσα υλοποίηση του μεσολαβητή για να γίνει ακόμα λειτουργικότερος και φιλικότερος ακόμα και για τον πιο άπειρο χρήστη.

Καθώς ο μεσολαβητής είναι μια σύγχρονη δικτυακή τοποθεσία μπορούν πολύ εύκολα να ενσωματωθούν κομμάτια στον ήδη υπάρχοντα κώδικα. Αυτά τα κομμάτια δεν είναι άλλα από μερικές σελίδες jsp και μερικές κλάσεις. Κάποια στοιχεία που θα βελτίωναν τον μεσολαβητή είναι η δρομολόγηση εργασιών και ο υπολογισμός του φόρτου εργασίας.

Η δρομολόγηση των εργασιών είναι ένα σημαντικό εργαλείο αν σκεφτούμε ότι αναφερόμαστε σε εταιρίες που πολλές φορές ο χρόνος και οι ημερομηνίες γίνονται πειστικές και επιθυμούν να τελειώνει μια εργασία το συντομότερο δυνατόν. Θα υπάρχουν τριών ειδών δρομολογήσεων σε πρώτη φάση. Η χειροκίνητη δρομολόγηση θα είναι όπως γίνετε και στην παρούσα υλοποίηση του μεσολαβητή. Η φθηνότερη και ταχύτερη δρομολογήσεις θα είναι οι καινούριες αυτόματες που με μορφή wizard θα οδηγούν τον κάθε χρήστη στις προεπιλεγμένες επιλογές.

Ο υπολογισμός του φόρτου εργασίας θα γίνετε μέσω μιας κλάσης που θα προστεθεί στον κώδικα του μεσολαβητή και θα κάνει κατά προσέγγιση



υπολογισμό του workload. Αυτό θα γίνει με έναν αλγόριθμο που θα αναλυθεί παρακάτω.

## **Η τεχνολογία και τα εργαλεία που χρησιμοποιήθηκαν**

Καθώς το GRIA είναι εξ' ολοκλήρου γραμμένο σε Java και ο μεσολαβητής πολύ σωστά επιλέχθηκε να γραφεί σε Java Server Pages. Άρα συνεχίζοντας αυτές τις τεχνολογίες θα υλοποιήσουμε τις βελτιώσεις του μεσολαβητή σε jsp και Java.

Για την δημιουργία καινούργιων jsp χρησιμοποιήθηκε ένα πρόγραμμα της εταιρίας Macromedia το Dreamweaver η έκδοση MX που είναι ικανό να δημιουργήσει κάθε είδους δυναμικές και στατικές σελίδες. Στην δημιουργία των κλάσεων και του applet χρησιμοποιήθηκε ένα πρόγραμμα της IDEA το IntelliJ που είναι από τα καλύτερα εργαλεία ανάπτυξης κώδικα σε γλώσσα Java. Το ίδιο μπορεί να δημιουργήσει και jsp αλλά δεν προτείνεται λόγω των resources που χρησιμοποιεί από τον υπολογιστή.

Επίσης χρησιμοποιήθηκε η βάση δεδομένων που ήδη έχει χρησιμοποιηθεί στον μεσολαβητή χωρίς καμία τροποποίηση. Δεν χρειάστηκε καμία αλλαγή αφού όλα τα στοιχεία που χρειάζονται είναι ήδη στην βάση και δεν χρειάζεται προσθήκη κανενός. Διατηρούνται έτσι όλα τα στοιχεία και χωρίς καμία προσθήκη στην βάση δεδομένων προσθέτουμε δεδομένα κάθε φορά που γίνετε αυτόματη δρομολόγηση. Χρησιμοποιείται βάση δεδομένων MySQL, αρχικά γιατί είναι σταθερή χωρίς προβλήματα στην εγκατάσταση αλλά και στην αποθήκευση σημαντικών και ευαίσθητων δεδομένων των εταιριών και επίσης είναι open source και δωρεάν σε αυτήν την έκδοση που την χρησιμοποιήσαμε. Ακόμα ένα προτέρημα της MySQL είναι ότι μπορεί να τρέξει σε όλα τα λειτουργικά συστήματα όπως άλλωστε και όλα τα περιφερειακά προγράμματα που χρησιμοποιούνται για την εκτέλεση του μεσολαβητή. Επίσης έχει βγει και έκδοση της μου υποστηρίζει και ελληνικούς χαρακτήρες

Ακόμα ένας web server είναι αναγκαίος στο μηχάνημα που είναι εγκατεστημένο το client κομμάτι της GRIA για τις δοκιμές του μεσολαβητή. Σε αυτόν τον τομέα ποικίλουν οι επιλογές, εμείς όμως επιλέξαμε το πιο διαδεδομένο και δωρεάν πρόγραμμα τον Apache Tomcat της Jakarta που έχει την δυνατότητα να τρέχει και σε Windows και σε Linux αλλά και σε Mac. Ο web server είναι το πρόγραμμα που θα εγκατασταθεί στον εξυπηρετητή και θα μεταγλωττίζει τα jsp και θα τα παρουσιάζει σαν σελίδες html.

### **Λειτουργία του μεσολαβητή**

Όπως είπαμε η τεχνολογία που έχει χρησιμοποιηθεί στον μεσολαβητή είναι τα jsp πράγμα που βοηθάει πολύ στην βελτίωση του και προσθήκη νέων χαρακτηριστικών. Πρέπει όμως να κατανοηθεί ο τρόπος λειτουργίας του μεσολαβητή για να μπορέσουμε με επιτυχία να τον τροποποιήσουμε έτσι ώστε να γίνει ακόμα λειτουργικότερος.

Κατά την είσοδο στον δικτυακό τόπο του μεσολαβητή γίνεται εισαγωγή του ονόματος χρήστη και κωδικού του χρήστη που εισέρχεται. Το κατάλληλο jsp ελέγχει μέσω της βάσης δεδομένων τις μεταβλητές αν υπάρχουν αρχικά και στην συνέχεια τα δικαιώματα που έχει ο χρήστης έτσι ώστε να αποκλείσει λειτουργίες σε περίπτωση που δεν είναι ο διαχειριστής. Όπως είπαμε οι διαχειριστές είναι οι μοναδικοί χρήστες που έχουν πρόσβαση σε όλες τις λειτουργίες του μεσολαβητή.

Στην συνέχεια όταν γίνει η είσοδος του χρήστη βρίσκετε στην πρώτη σελίδα του τόπου του μεσολαβητή.

[Home](#)  
[Administration Home](#)  
[New Account](#)  
[Tender](#)  
[Upload Data](#)  
[Set Job](#)  
[Check Job](#)  
[Download Data](#)  
[Check Current Conversations](#)  
  
[Logout](#)

**Hello Administrator !!!**

[Bug Report](#)

Όταν ένας χρήστης μπαίνει στην αρχική σελίδα του μεσολαβητή μπορούμε να διαπιστώσουμε ότι δεν είναι όλες οι επιλογές ενεργοποιημένες. Αυτό συμβαίνει για τον λόγο ότι δεν έχει σταλεί καμιά δουλειά για επεξεργασία ακόμα και βρίσκετε στο στάδιο να σταλεί. Πριν επιχειρηθεί να σταλεί κάποια δουλειά σε κάποιον παροχέα πρέπει πρώτα να δημιουργηθεί λογαριασμός στον συγκεκριμένο παροχέα. Έπειτα μπορεί να γίνει η επικοινωνία μεταξύ του παροχέα και το μεσολαβητή για να ανταλλάξουν τις απαραίτητες πληροφορίες. Παρακάτω θα αναλύσουμε τις λειτουργίες του παροχέα και στην συνέχεια θα αναφέρουμε τις βελτιώσεις και τον τρόπο που επιτεύχθηκαν.

## **Open Account**

Όπως αναφέραμε στο κομμάτι client του GRIA έτσι και στον μεσολαβητή πρέπει να δημιουργηθεί κάποιος λογαριασμός σε κάποιον παροχέα. Αυτό γίνεται με την εντολή `gria open` που στο συγκεκριμένο κομμάτι μπορεί να την εκτελέσει μόνο διαχειριστής που έχει εισαχθεί επιτυχώς στον δικτυακό τόπο του μεσολαβητή.

Όπως φαίνεται και στην παραπάνω εικόνα για να δημιουργηθεί ένας λογαριασμός σε κάποιον παροχέα θα πρέπει να συμπληρωθεί η παραπάνω φόρμα. Το βασικότερο όλων των πεδίων είναι το URL του παροχέα ώστε να επικοινωνήσει ο μεσολαβητής μαζί του. Τα υπόλοιπα πεδία είναι πεδία λεπτομερειών του διαχειριστή, χρήστη ή της εταιρίας που εργάζεται. Αυτή η σελίδα εξαφανίζεται όταν καλείτε το newaccount.jsp και μετά το τέλος της συμπλήρωσης των πεδίων και πατώντας το Submit καλείτε να εκτελεστεί του επόμενο jsp το openaccount.jsp, όπου κατά την εκτέλεσή του εμφανίζεται μήνυμα επιτυχίας ή αποτυχίας της δημιουργίας λογαριασμού στον παροχέα που επιλέξαμε.

## Tender

Στο συγκεκριμένο στάδιο είναι το σημείο που θα τροποποιήσουμε για να προσθέσουμε κάποια στοιχεία για την διευκόλυνση του χρήστη. Αναλύοντας το συγκεκριμένο βήμα, θα κατανοηθούν ευκολότερα οι ελλείψεις.

Home  
Administration Home  
New Account  
Tender  
Upload Data  
Set Job  
Check Job  
Download Data  
Check Current Conversations  
Logout

Description:

Allocation

ResourceAllocationStart: 2004-10-14

ResourceAllocationEnd: 2004-10-14

MaxStoreDataVolume: 10000000

MaxDownloadDataVolume: 10000000

MaxUploadDataVolume: 1000000

MaxNumConcurrentResourcesPerResourceAlloc: 1

MaxWork: 0.0

MinPhysicalMemory: 0

MinRelativeMachinePerformance: 1

NumParallelResourcesPerJob: 1

ServiceName: http://gria.org/GRIATestApp

Submit Reset

OCT 2004

S	M	T	W	T	F	S
				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Today Cancel

Bug Report

Κάνοντας tender ο χρήστης υποχρεούται να συμπληρώσει μια φόρμα με τα στοιχεία τις δουλειάς του, στο setrequirements.jsp. Αυτά είναι χρονικά περιθώρια, δεδομένα μεγέθους της εργασίας, απαιτούμενος χώρος στον παροχέα, η υπηρεσία που επιθυμεί ο χρήστης, καθώς επίσης και το φόρτος εργασίας που πρέπει ο χρήστης να το υπολογίσει μόνος του. Μετά την σωστή συμπλήρωση των δεδομένων καλείτε το επόμενο jsp, το sendrequirements.jsp, δημιουργείτε το Requirements.xml που έχουμε αναφέρει νωρίτερα και γίνεται η επικοινωνία μεταξύ του μεσολαβητή και του παροχέα επιστρέφοντας ο δεύτερος δεδομένα όπως κόστος και χρονικά περιθώρια εκτέλεσης της εργασίας.

Η απάντηση του παροχέα οι των παροχέων φαίνεται κάπως έτσι, μόνο που στην συγκεκριμένη εικόνα έχει γίνει επικοινωνία με έναν μόνο παροχέα.

[Home](#)  
[Administration Home](#)  
[New Account](#)  
[Tender](#)  
[Upload Data](#)  
[Set Job](#)  
[Check Job](#)  
[Download Data](#)  
[Check Current Conversations](#)

Select	Account URL	Price	MaxNumConcurrent ResourcesPer Resource Alloc	Λ D:
<input checked="" type="radio"/>	<a href="http://vinsanto.it-innovation.soton.ac.uk:443/axis/services/AccountService#1625">http://vinsanto.it-innovation.soton.ac.uk:443/axis/services/AccountService#1625</a>	0.01	1	10

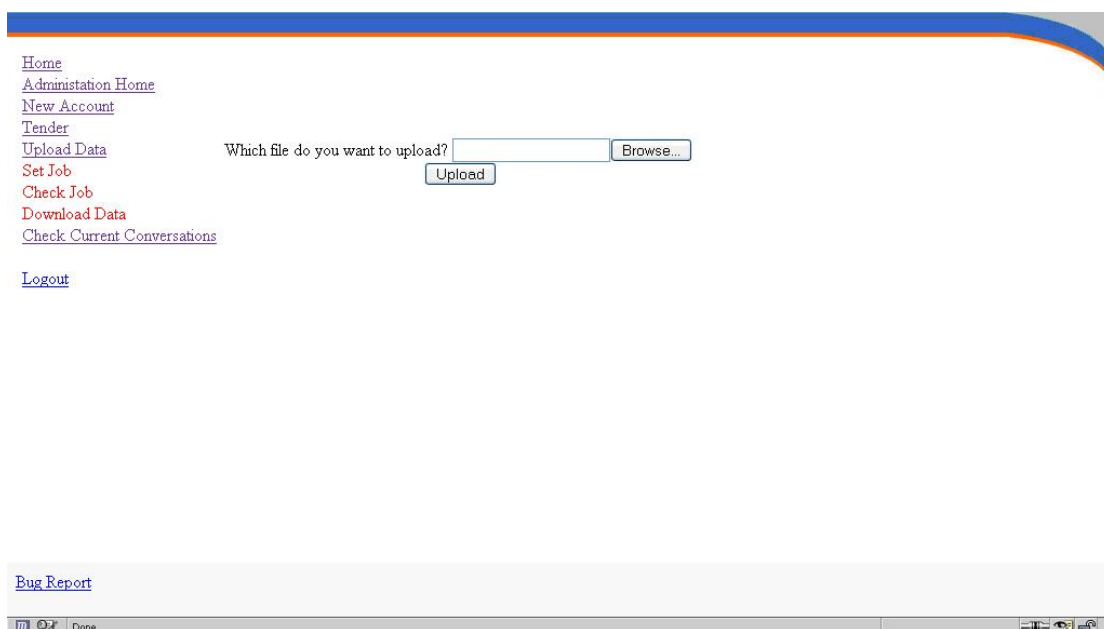
[Logout](#)



Σε περίπτωση που υπήρχαν και άλλοι παροχείς θα γινόταν επιλογή κατά την κρίση του χρήστη για τον βέλτιστο παροχέα και θα γινόταν η επιβεβαίωση χώρου με το `confirmallocation.jsp`, και τελικά θα εμφανιζόταν μήνυμα επιτυχημένης ή αποτυχημένης ενέργειας.

## Upload

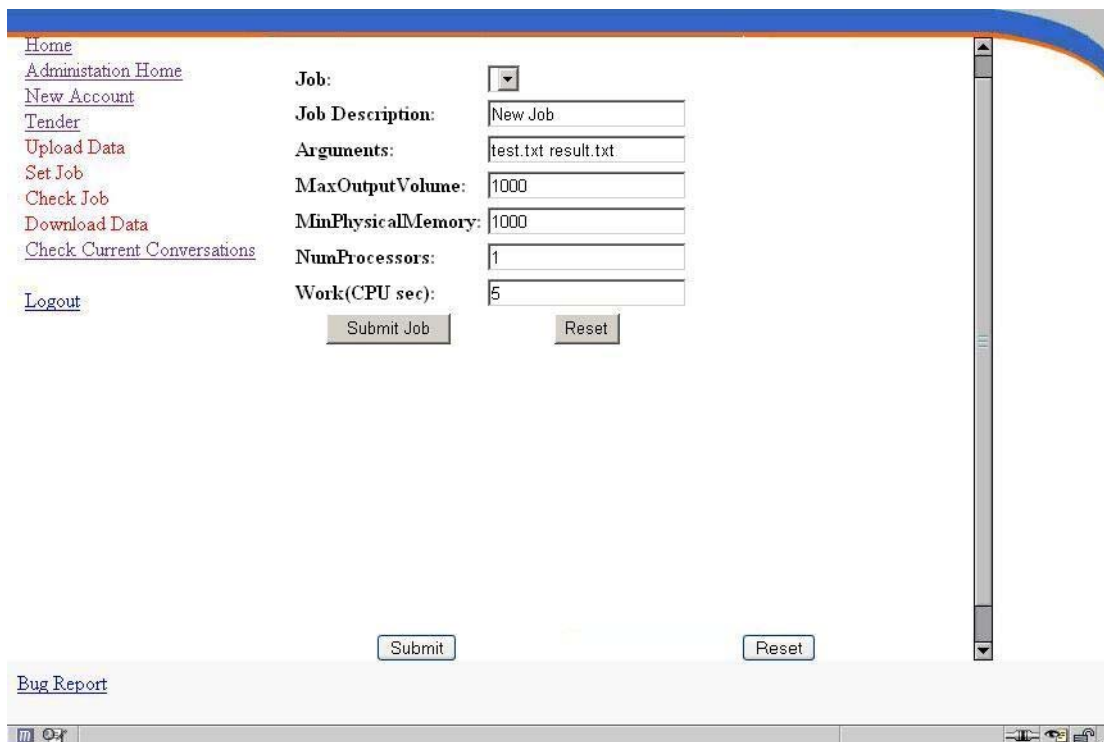
Η συνέχεια που ακολουθείτε και στο GRIA client είναι το upload των δεδομένων που είναι για επεξεργασία. Πριν από οποιαδήποτε κίνηση πρέπει τα αρχεία να συμπίεστούν κατάλληλα και κάθε ομάδα ξεχωριστά για να αποσταλούν. Στο setupload.jsp ορίζει ο χρήστης το αρχείο που επιθυμεί να στείλει στον παροχέα και έπειτα το selectupload.jsp καλεί ένα servlet που βρίσκεται στην θέση /servlet/UpLoadIt και αποστέλλει το αρχείο στον επιλεγμένο παροχέα. Αυτό φυσικά πρέπει να επαναληφθεί για όσα αρχεία χρειάζονται να σταλούν στους παροχείς.



Προς την αποστολή του αρχείου από τον χρήστη στον παροχέα, το αρχείο περνάει από τον μεσολαβητή και μετά στον παροχέα. Το αρχείο γράφεται σε μια ορισμένη τοποθεσία στον μεσολαβητή και έπειτα αποστέλλεται στον ζητούμενο παροχέα.

## Run

Έχοντας αποστείλει όλα τα αρχεία που πρέπει να επεξεργαστούν, το επόμενο βήμα είναι να δοθεί εντολή για την εκτέλεση της επεξεργασίας στον παροχέα.



The screenshot shows a web application interface with a navigation menu on the left and a job submission form on the right. The navigation menu includes links for Home, Administration Home, New Account, Tender, Upload Data, Set Job, Check Job, Download Data, Check Current Conversations, and Logout. The job submission form contains the following fields and controls:

Job:	<input type="text" value="New Job"/>
Job Description:	<input type="text" value="New Job"/>
Arguments:	<input type="text" value="test.txt result.txt"/>
MaxOutputVolume:	<input type="text" value="1000"/>
MinPhysicalMemory:	<input type="text" value="1000"/>
NumProcessors:	<input type="text" value="1"/>
Work(CPU sec):	<input type="text" value="5"/>

Below the form are two buttons: "Submit Job" and "Reset". At the bottom of the page, there are two more buttons: "Submit" and "Reset". A "Bug Report" link is visible in the footer area.

Για την εκκίνηση της εργασίας πρέπει να δημιουργηθεί ένα αρχείο τύπου xml όπως αναφέραμε και παραπάνω για το GRIA client. Στο setjob.jsp υπάρχει η παραπάνω φόρμα για την αυτόματη δημιουργία του Work.xml. Έπειτα από την συμπλήρωση των δεδομένων που είναι υποχρεωτικά για την εκκίνηση της επεξεργασίας καλείτε το sendjob.jsp που επικοινωνεί με τον παροχέα και ξεκινάει η εργασία.



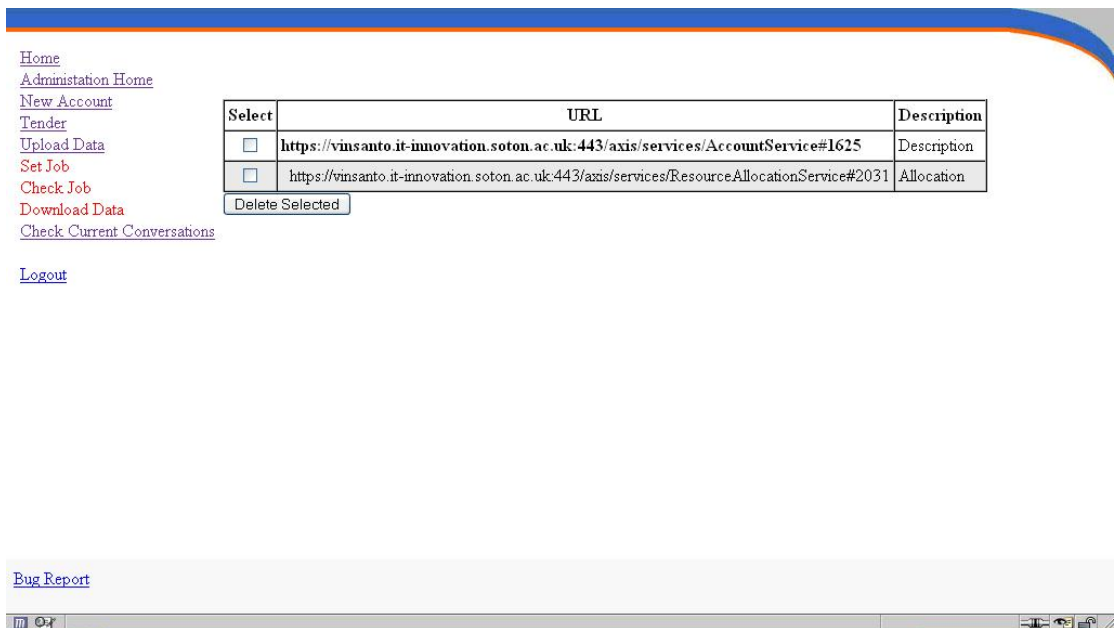
## Download

Καθώς έχουν αποσταλεί τα δεδομένα και έχουν επεξεργαστεί από τον παροχέα με τα στοιχεία που του έχει δώσει ο χρήστης, έπεται η λήψη των επεξεργασμένων δεδομένων από τον χρήστη. Αυτό γίνεται μέσω του μεσολαβητή από την σελίδα download.jsp.

Στην συγκεκριμένη σελίδα ο χρήστης επιλέγει από τα δεδομένα που υπάρχουν στον λογαριασμό του στον παροχέα που έχει αποστείλει τα δεδομένα, είτε αυτά είναι δεδομένα εισόδου είτε εξόδου, και στην συνέχεια από την επόμενη σελίδα την downloaddata.jsp μεταφέρει τα αρχεία αυτά στον υπολογιστή του.

## Show – Browse

Σε οποιαδήποτε φάση μπορεί να βρίσκετε ο χρήστης του μεσολαβητή μπορεί να ελέγξει τις εργασίες που βρίσκονται σε λειτουργία. Μέσω τις σελίδας conversations.jsp μπορούμε εμφανίσουμε τις διαπραγματεύσεις μεταξύ χρήστη και παροχέα, την περιγραφή τους αλλά και την κατάσταση στην οποία βρίσκονται.



The screenshot displays a web application interface. On the left side, there is a vertical menu of navigation links: [Home](#), [Administration Home](#), [New Account](#), [Tender](#), [Upload Data](#), [Set Job](#), [Check Job](#), [Download Data](#), [Check Current Conversations](#), and [Logout](#). The main content area features a table with the following structure:

Select	URL	Description
<input type="checkbox"/>	<a href="https://vinsanto.it-innovation.soton.ac.uk:443/axis/services/AccountService#1625">https://vinsanto.it-innovation.soton.ac.uk:443/axis/services/AccountService#1625</a>	Description
<input type="checkbox"/>	<a href="https://vinsanto.it-innovation.soton.ac.uk:443/axis/services/ResourceAllocationService#2031">https://vinsanto.it-innovation.soton.ac.uk:443/axis/services/ResourceAllocationService#2031</a>	Allocation

Below the table, there is a button labeled "Delete Selected". At the bottom of the interface, there is a link for [Bug Report](#) and a standard Windows taskbar at the very bottom.

Όπως φαίνεται και στην παραπάνω εικόνα ο χρήστης κατά την εμφάνιση των διαπραγματεύσεων μπορεί να διαγράψει όποια επιθυμεί με σειρά όμως που είναι δυνατό αυτό να γίνει.

## **Finish**

Τέλος πρέπει ο χρήστης να ενημερώσει τον παροχέα για την λήξη των εργασιών έτσι ώστε να αποδεσμευτούν οι πόροι. Αυτή η διακοπή των συζητήσεων, conversation, διακόπτονται μέσω της σελίδας conversations.jsp που μπορεί ο χρήστης, με δικαιώματα διαχειριστή, να διαγράψει όποια συζήτηση επιθυμεί. Φυσικά με σειρά που είναι δυνατόν αυτό να γίνει. Για παράδειγμα δεν μπορεί να διαγραφεί μια συζήτηση εισόδου ενώ υπάρχει ήδη συζήτηση για την εκτέλεση της εργασίας και είναι σε λειτουργία.

Ένα όχι αναγκαστικά μειονέκτημα του GRIA είναι ότι όλες οι ενεργές συζητήσεις καταγράφονται σε ένα αρχείο με το όνομα gria.state. Σε αυτό το αρχείο έχουν πρόσβαση όλοι οι χρήστες που σημαίνει ότι οποιοσδήποτε χρήστης θα μπορεί να δημιουργήσει μια δουλειά και να στείλει άλλος τα δεδομένα, ή ακόμα κάποιος να ακυρώσει μια δουλειά άλλου. Αυτό θα μπορούσε να παρακαμφτεί αν μπορούσαμε να μην ορίζεται αυτόματα το όνομα αυτό αλλά με επιλογή του κάθε χρήστη ή αυτόματα σε κάθε χρήστη ξεχωριστά.

Μετά απ' όσα αναφέραμε είναι κατανοητό ότι δεν χρειάζεται καμία εγκατάσταση σε κανέναν υπολογιστή χρήστη. Οι μοναδικές εγκαταστάσεις γίνονται στον εξυπηρετητή, server, που χρειάζεται το GRIA, έναν Tomcat, το JRE και την βάση δεδομένων MySQL. Έτσι κάθε χρήστης μπορεί να βελτιώσει την απόδοση του χρησιμοποιώντας τον browser του και μόνο. Όμως για λόγους ασφαλείας δημιουργήθηκαν δικλίδες ασφαλείας που αποτρέπουν στους χρήστες να κάνουν λάθη, όπως για παράδειγμα να πάνε σε ένα βήμα χωρίς να έχουν υλοποιήσει το προηγούμενο, δηλαδή δεν μπορεί να κάνει κάποιος upload χωρίς να έχει κάνει tender. Αυτό το απαγορεύει ο εξυπηρετητής από το δίκτυο της εταιρίας χωρίς να επιβαρύνει τον παροχέα με τέτοιου είδους εύκολα ανιχνεύσιμα λάθη.

## **Επιπρόσθετα χαρακτηριστικά του μεσολαβητή**

Όπως αναφέραμε παραπάνω υπάρχουν κάποιοι τρόποι βελτίωσης του μεσολαβητή και την ευκολότερη χρήση του από τους απλούς χρήστες. Η διαδικασία για την είσοδο δεδομένων που αφορούν τα δεδομένα που είναι προς επεξεργασία από κάποιον παροχέα συχνά είναι πολύπλοκα και χρονοβόρα. Ένα νέο εργαλείο χωρίς να παρεμβαίνει στην μέχρι τώρα λειτουργία του μεσολαβητή είναι και η δρομολόγηση των εργασιών.

Ο τρόπος που ακολουθείτε μέχρι τώρα για την αποστολή μια εργασίας από τον μεσολαβητή περιγραφικέ παραπάνω. Δίνουμε την δυνατότητα στον κάθε χρήστη να αυτοματοποιήσει την αποστολή της εργασίας του με τον ακόλουθο τρόπο.

Θέλοντας να κάνει tender ένας χρήστης του εμφανίζεται να επιλέξει να θέλει να ακολουθήσει τον αυτοματοποιημένο τρόπο ή τον χειροκίνητο που μέχρι τώρα είναι σε λειτουργία.

Οι αυτόματες δρομολογήσεις είναι 2 ειδών, η φθηνότερη και η γρηγορότερη. Επιλεχθήκαν να γίνουν αυτές γιατί θεωρούμε ότι είναι αυτές που θα επιθυμούσαν περισσότερο οι πελάτες εταιρίες. Η πρώτη δρομολόγηση βρίσκει τον οικονομικότερο παροχέα και στέλνει αυτόματα εκεί την δουλειά. Ενώ στην δεύτερη δρομολόγηση παίρνει τις ταχύτερες επεξεργασίας όλων των παροχέων και στέλνει αρχεία ανάλογα το φόρτο εργασίας του κάθε αρχείου. Έτσι επιτυγχάνεται της ταχύτερης αποπεράτωσης εργασίας.

## **Βάση δεδομένων**

Η αρχική ανάγκη της βάσης ήταν για τη δημιουργία χρηστών. Αυτοί καταγράφονται σε έναν πίνακα με ονομασία user και από εκεί επαληθεύονται τα ονόματα των χρηστών και οι κωδικοί τους. Επίσης περιέχει πληροφορίες για τους χρήστες όπως τηλέφωνα, ηλεκτρονικό ταχυδρομείο, τόπος παραμονής, το επίπεδο της ασφαλείας τους και άλλα. Μέσα από της σελίδες του μεσολαβητή μπορεί ένας

χρήστης με δικαιώματα διαχειριστή να επικοινωνήσει με τη βάση να δημιουργήσει καινούριο χρήστη, να διαγράψει κάποιον και ακόμα να κάνει αλλαγές στον κάθε χρήστη και να αλλάξει και το επίπεδο ασφάλειάς του.

Καθώς η βάση δεδομένων υπάρχει για λόγους ασφαλείας και στατιστικών, ακόμα και οι αυτόματες δρομολογήσεις καταγράφονται σε αυτήν. Εκτός από τον πίνακα των χρηστών υπάρχουν και πίνακες για την καταγραφή των επιλογών τους και καταχωρήσεων που γίνονται στις φόρμες που συμπληρώνει ο ίδιος ή συμπληρώνονται αυτόματα από τον μεσολαβητή. Οι πίνακες open, tender και run είναι οι πίνακες που καταγράφονται οι εκάστοτε επιλογές των χρηστών για τις αντίστοιχες κινήσεις τους.

Επίσης για την καλύτερη λειτουργικότητα και συλλογή πληροφοριών για στατιστικούς λόγους υπάρχει ένας πίνακας logging που καταγράφει την ημερομηνία και την ώρα για κάθε κίνηση του χρήστη από την στιγμή που θα κάνει μια επιτυχημένη είσοδο στον μεσολαβητή μέχρι που θα αποσυνδεθεί από αυτόν. Η προβολή αυτών των δεδομένων γίνεται από μια σελίδα που έχουν δικαίωμα να μπουν μόνο οι διαχειριστές και γίνεται με μορφή ραβδογράμματος

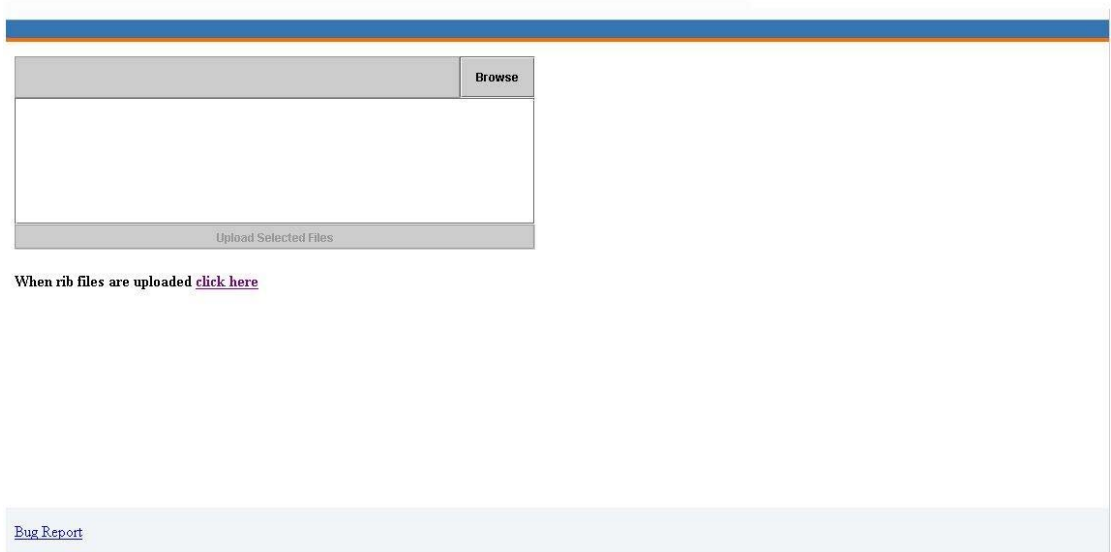
The screenshot shows a web application interface with a navigation menu on the left and a data table on the right. The navigation menu includes links for Home, Administration Home, New Account, Tender, Upload Data, Set Job, Check Job, Download Data, Check Current Conversations, and Logout. The data table has the following columns: Name, Lastname, (username), log in, log out, new account, tender, confirm, upload, download, run, check job, show, and delete. The table contains one row for a user named George Drimakis (username: gdrim) with the following values: log in: 7, log out: 5, new account: 0, tender: 1, confirm: 0, upload: 0, download: 0, run: 0, check job: 3, show: 7, and delete: 2. Below the table is a 'Back' link. At the bottom of the screenshot, there is a 'Bug Report' link and a browser status bar showing 'Done'.

Name	Lastname	(username)	log in	log out	new account	tender	confirm	upload	download	run	check job	show	delete
George	Drimakis	(gdrim)	7	5	0	1	0	0	0	0	3	7	2

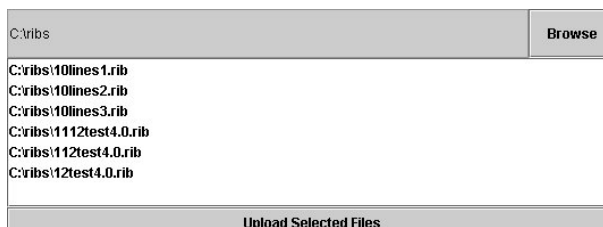
Ακόμα μπορεί να επιλέξει από την βάση να εμφανιστούν συγκεκριμένα στοιχεία σε συγκεκριμένες μόνο ημερομηνίες ή χρονικά διαστήματα.

## Cheapest solution

Κατά την επιλογή του χρήστη για φθηνή αυτόματη δρομολόγηση εμφανίζεται στην σελίδα uploadapplet.jsp που περιέχει ένα applet που ζητά να επιλέξει τον φάκελο που περιέχει τα δεδομένα που επιθυμεί να στείλει προς επεξεργασία, έτσι ώστε να αντιγραφούν όλα τα αρχεία στον εξυπηρετητή που βρίσκετε ο μεσολαβητής για να υπολογιστούν διάφορες μεταβλητές που χρειάζονται για την αποστολή των στοιχείων της εργασίας από την σελίδα setrequirements.jsp.



Επιλέγοντας τον φάκελο που είναι μέσα τα δεδομένα προς επεξεργασία, το applet αναζητά ακόμα και στους υποφακέλους του για δεδομένα με τις καταλήξεις που έχουν οριστεί και τα εμφανίζει καθώς ο χρήστης δίνει την εντολή να αποσταλούν στον μεσολαβητή.



Όταν τα αρχεία είναι στον μεσολαβητή και ο χρήστης προχωράει στην επόμενη σελίδα που είναι η `setautorequirements.jsp`. Ο μεσολαβητής υπολογίζει το `workload`, το μέγεθος και τον απαιτούμενο χώρο που χρειάζεται να υπάρχει στον μεσολαβητή των δεδομένων και συμπληρώνει την φόρμα στο `setautorequirements.jsp` αυτόματα χωρίς να αφήνει τον χρήστη να παρέμβει και να αλλάξει τα δεδομένα παρά μόνο τις ημερομηνίες που επιθυμεί.

Home  
Administration Home  
New Account  
Tender  
Upload Data  
Set Job  
Check Job  
Download Data  
Check Current Conversations  
Logout

Description: Allocation  
ResourceAllocationStart: 2004-11-17  
ResourceAllocationEnd: 2004-11-30  
MaxStoreDataVolume: 12002991  
MaxDownloadDataVolume: 10911810  
MaxUploadDataVolume: 1091181  
MaxNumConcurrentResourcesPerResourceAlloc: 1  
MaxWork(CPU Seconds): 9984  
MinPhysicalMemory: 0  
MinRelativeMachinePerformance: 1  
NumParallelResourcesPerJob: 1  
ServiceName: http://gria.org/rendering

Submit Reset

NOV 2004

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Today Cancel

Bug Report

Έπειτα μεταφέρεται στο επόμενο βήμα που είναι η εμφάνιση και ενημέρωσης του χρήστη για τον οικονομικότερο παροχέα που έχει ήδη επιλεγεί. Η επιλογή του οικονομικότερου παροχέα γίνεται έπειτα από την εκτέλεση μιας ρουτίνας στο συγκεκριμένο `jsp` και αφού έχει λάβει ο μεσολαβητής τις απαντήσεις από όλους τους παροχείς ελέγχει την τιμή που απαιτεί ο κάθε παροχέας με μια αναδρομή και αποθηκεύει προσωρινά τον παροχέα με την ελάχιστη τιμή και στο τέλος εμφανίζει τον συγκεκριμένο παροχέα.

[Home](#)  
[Administration Home](#)  
[New Account](#)  
[Tender](#)  
[Upload Data](#)  
[Set Job](#)  
[Check Job](#)  
[Download Data](#)  
[Check Current Conversations](#)

Account URL	Price	MaxNumConcurrent ResourcesPer ResourceAlloc	Λ D:
<a href="https://vinsanto.it-innovation.soton.ac.uk:443/axis/services/AccountService#1625">https://vinsanto.it-innovation.soton.ac.uk:443/axis/services/AccountService#1625</a>	0.01	1	10

[Logout](#)



Κατά την είσοδο στο επόμενο βήμα ο μεσολαβητής συμπιέζει σε κάθε αρχείο δύο αρχεία με το ίδιο όνομα και αποστέλλει τα συμπιεσμένα αρχεία στον παροχέα που εκτελεί και την εντολή gun χωρίς να το αντιλαμβάνεται ο χρήστης για την εκτέλεση της επεξεργασίας τους.

Έπειτα η λήψη των επεξεργασμένων δεδομένων γίνονται χειροκίνητα από τον χρήστη.

## Fastest solution

Παρόμοια γίνεται και κατά την επιλογή για την γρηγορότερη δρομολόγηση, με την διαφορά ότι η εργασία καταμερίζεται σε όσους παροχείς υπάρχει λογαριασμός και στέλνονται μέρη ανάλογα με το φόρτος εργασίας που μπορεί ο κάθε παροχέας να επεξεργαστεί ανά δευτερόλεπτο. Έτσι επιτυγχάνουμε την ταχύτερη δρομολόγηση χωρίς να μπαίνει στην ρουτίνα ο χρήστης και να στέλνει σε διαφορετικούς παροχείς τα αρχεία αφού τα έχει συμπιέσει πρώτα.

Κατά την λήψη των δεδομένων από τους παροχείς που περιέχουν πληροφορίες, εκτός από την τιμή, την επεξεργαστική ισχύ του κάθε cluster που αντιπροσωπεύει ο κάθε παροχέας. Αυτή η μεταβλητή, της επεξεργαστικής ισχύς, είναι ένας αριθμός που υπολογίζεται από τον παροχέα μέσω μίας κλάσης που έχει δημιουργηθεί από τους υλοποιείτες του GRIA supplier. Βάσει αυτού του αριθμού καταλαβαίνουμε πόση υπολογιστική ισχύ έχει το κάθε cluster για να αποστείλουμε ανάλογο φόρτο εργασίας. Κατά την λήψη των δεδομένων για όλους τους suppliers με μια αναδρομή αθροίζουμε την επεξεργαστική ισχύ όλων των παροχέων και αποθηκεύουμε το αποτέλεσμα σε ένα session. Παράλληλα τους επιμέρους αριθμούς που αντιπροσωπεύουν την επεξεργαστική ισχύ τους αποθηκεύουμε σε έναν πίνακα. Μαζί με το workload που κατά προσέγγιση γνωρίζουμε υπολογίζουμε τον φόρτο εργασίας που πρέπει να αποστείλουμε στον κάθε παροχέα για να επεξεργαστεί τόση ώρα όση και οι υπόλοιποι παροχείς με σκοπό την ταυτόχρονη αποπεράτωση της εργασίας που στο σύνολο στο συντομότερο δυνατό χρονικό διάστημα.

Από την παρακάτω σχέση υπολογίζεται το φόρτος εργασίας που θα αποσταλεί στον κάθε παροχέα.

$i$  = το σύνολο των suppliers

$sup\_wrkld_i$  = το φόρτος εργασίας που θα αποσταλεί στον  $i$  supplier

$sup\_proc\_pwr_i$  = επεξεργαστική ισχύς του  $i$  supplier

$total\_proc\_pwr$  = συνολική επεξεργαστική ισχύς (το άθροισμα όλων)

$total\_wrkld$  = συνολικό workload που έχουμε για αποστολή στους suppliers



$$\text{sup\_wrkld}_i = \frac{\text{sup\_proc\_pwr}_i}{\text{total\_proc\_pwr}} \text{total\_wrkld} * 100\%$$

The screenshot shows a web application interface with a blue header. On the left, there is a vertical menu with links: Home, Administration Home, New Account, Tender, Upload Data, Set Job, Check Job, Download Data, Check Current Conversations, and Logout. A 'Confirm Allocation' button is located below the menu. The main content area features a table with the following data:

%	Account URL	Price	MaxNumConcurrent ResourcesPer ResourceAlloc	Max Data'
30	https://ece.ntua.gr:763/axis/services/AccountService#3643	0.01	1	
70	https://vinsanto.it-innovation.soton.ac.uk:443/axis/services/AccountService#3643	0.05	1	

Below the table, there is a 'Bug Report' link and a status bar showing 'Done'.

Έτσι μετά τους παραπάνω αυτοματοποιημένους υπολογισμούς ο μεσολαβητής επιστρέφει την παραπάνω σελίδα στον χρήστη για να τον ενημερώσει τον χρήστη το ποσοστό του φόρτου εργασίας και τον παροχέα που θα αποστείλει τα αρχεία που είναι για επεξεργασία όπως την τιμή που ζητά ο κάθε παροχέας και στην συνέχεια όλες τις υπόλοιπες πληροφορίες που επιστρέφουν οι παροχείς στον μεσολαβητή.

## Workload

Ο υπολογισμός του φόρτου εργασίας πρέπει να αναφέρουμε ότι γίνεται -κατά προσέγγιση- με μια αναδρομή. Ο υπολογισμός γίνεται μέσω μίας κλάσης, γραμμένης σε Java, που περιέχει διεργασίες, οι οποίες όταν καλούνται επιστρέφουν αποτελέσματα όπως το μέγεθος των αρχείων, το πλήθος τους, το workload, καθώς και άλλες πληροφορίες, όπως το πλήθος των προσπελάσιμων αρχείων που χρησιμοποιήθηκαν για την εύρεση του workload και σε πια θέση βρίσκονται.

Τα αρχεία που αποστέλλονται για επεξεργασία είναι αρχεία τύπου rib και dat. Τα rib είναι αυτά που περιέχουν όλες τις πληροφορίες που χρειάζονται ισχυρή επεξεργαστική ισχύ. Ο τρόπος για τον υπολογισμό του φόρτου εργασίας επιτυγχάνεται εξομοιώνοντας την επεξεργασία αυτών των αρχείων, αλλά όχι όλων. Ο αλγόριθμος είναι ως εξής, αρχίζοντας από το πρώτο αρχείο εξομοιώνουμε την επεξεργασία του και βρίσκουμε τον φόρτο εργασίας, στην συνέχεια ελέγχουμε το αρχείο που βρίσκεται +50 θέσεις στην σειρά από το πρώτο (ομοίως εξομοιώνοντας την επεξεργασία του). Αν η διαφορά του φόρτου εργασίας μεταξύ των δύο αρχείων είναι μεγαλύτερη από 15% τότε ελέγχουμε και το αρχείο που βρίσκετε στην μέση τους (το +25). Αν το πρώτο και το +25 έχουν διαφορά μεγαλύτερη από 15% τότε ελέγχουμε και το ενδιάμεσο και ούτω κάθε εξής μέχρι να βρούμε διαφορά μικρότερη από 15%, οπότε και συνεχίζουμε με το επόμενο αρχείο, που είναι +50 θέσεις μετά από το τελευταίο αρχείο, δεύτερο, που επεξεργαστήκαμε και γίνεται η ίδια αναδρομή. Φυσικά οι παράμετροι (τιμές της διαφοράς και του βήματος ελέγχου) είναι μεταβλητές και μπορούν να αλλάξουν μέσα από το jsp που καλεί την κλάση.

Η επί τοις εκατό διαφορά μεταξύ δύο αριθμών εξαρτάται από το ποιος αριθμός είναι πρώτος και ποιος δεύτερος. Έτσι, για να είμαστε όσο το δυνατόν πιο ακριβείς στις μετρήσεις μας, κάθε φορά βρίσκουμε τον μέσο όρο των δύο διαφορών, δηλαδή της διαφοράς του πρώτου από τον δεύτερο αριθμό και της διαφοράς του δεύτερου από τον πρώτο.

Παρακάτω δίνουμε ένα παράδειγμα υπολογισμού 258 αρχείων.

--> calculating 0 - 68	Total workload in accessed ribs --> 27206
--> calculating 50 - 68	Total Accesed ribs -> 35
--> calculating 100 - 68	Number of rib files -> 258
calculating 125 - 68	Total size of ribs and bat -> 16 MB
calculating 137 - 173	
calculating 131 - 415	
calculating 128 - 2949	
calculating 126 - 415	
calculating 127 - 173	
calculating 129 - 757	
calculating 130 - 68	
calculating 134 - 757	
calculating 132 - 173	Total percent 0.99999905
calculating 133 - 2949	Total workload 78443:78443
calculating 135 - 68	Check.. total workload from array 78443
calculating 136 - 415	Check the accessed..0
calculating 143 - 757	
calculating 140 - 415	
calculating 138 - 2949	

calculating 139 - 68	TOTAL WORKLOAD.. 78443
calculating 141 - 173	
calculating 142 - 2949	workload per files.. 304
calculating 146 - 173	
calculating 144 - 68	ok workload 47466.0 of 47065 , 0.6%
calculating 145 - 415	
calculating 148 - 757	ok workload 30977.0 of 31377 , 0.4%
calculating 147 - 2949	
calculating 149 - 68	
--> calculating 150 - 415	
--> calculating 200 - 415	
--> calculating 250 - 415	
calculating 253 - 757	
calculating 251 - 173	
calculating 252 - 2949	
last calculated --> 257 - 757	

Αναλύοντας το παράδειγμα βλέπουμε ότι ο έλεγχος του πρώτου αρχείου και του 50ου δεν είχε διαφορά μεγαλύτερη από 15%, οπότε δεν ελέγχθηκε κανένα ενδιάμεσο αρχείο. Το ίδιο συνέβη και μεταξύ του 50ου και 100ου αρχείου. Όμως η διαφορά στο φόρτο εργασίας μεταξύ του 100ου και του 150ου αρχείου ήταν μεγαλύτερη από 15% με αποτέλεσμα να ελεγχθεί και το 125ο αρχείο, πρώτα με το 100ο, όπου δεν διαπιστώθηκε διαφορά μεγαλύτερη από 15%, και έπειτα με το 150ο, όπου βρέθηκε διαφορά μεγαλύτερη από 15% και σαν αποτέλεσμα ελέγχθηκαν και τα δύο αρχεία με το ενδιάμεσό τους (αν διαγράψουμε τα ψηφία μετά την υποδιαστολή, το 137ο). Το 137ο έχει διαφορά μεγαλύτερη από 15% και με το 125ο και με το 150ο, οπότε ομοίως συνεχίζουμε τους ελέγχους με τα ενδιάμεσα τους. Συνεχίζοντας με αυτόν τον τρόπο, αποθηκεύουμε σε ένα πίνακα το workload και τον αριθμό του αρχείου που αντιστοιχεί. Μετά βρίσκουμε το συνολικό sum\_workload προσθέτοντας όλα τα επί μέρους workload και επίσης βρίσκουμε τον μέσο όρο ave\_workload του κάθε αρχείου που προσπελάσαμε. Έπειτα το πολλαπλασιάζουμε με το συνολικό αριθμό των αρχείων και βρίσκουμε με προσέγγιση το συνολικό agg\_workload.

Στην πραγματικότητα όμως δεν έχουν όλα τα αρχεία το ίδιο workload και για να αποστείλουμε στους παροχείς workload ανάλογο των δυνατοτήτων τους πρέπει να βρεθούν πια αρχεία έχουν εκείνο το ποσοστό workload που επιθυμούμε να αποστείλουμε στον κάθε παροχέα. Η εύρεση του workload εκείνων των αρχείων που δεν έχουν προσπελαστεί (δηλαδή δεν έχει υπολογιστεί η τιμή τους) γίνεται υποθέτοντας ότι το workload τους είναι ίσο με τον μέσο όρο του προηγούμενου και του επόμενου αρχείου που έχει προσπελαστεί (δηλαδή έχει υπολογιστεί η workload τιμή τους). Γνωρίζοντας επομένως το πόσο workload θα πρέπει να

αποστέλλουμε στον κάθε παροχέα, αρχίζουμε να αποστέλλουμε ένα ένα τα αρχεία στον παροχέα, προσθέτοντας παράλληλα το workload του κάθε αρχείου μέχρι να η συνολική τιμή του workload ανά παροχέα να φτάσει την ζητούμενη τιμή που πρέπει να αποσταλεί (στον συγκεκριμένο παροχέα). Στην συνέχεια συνεχίζουμε εφαρμόζοντας τον ίδιο τρόπο στα επόμενα αρχεία και στους επόμενους παροχείς. Σε περίπτωση που κάποιο αρχείο περισσέψει (δεν αποσταλεί σε κάποιον παροχέα), τότε έχει προβλεφθεί προγραμματιστικά να αποσταλεί στον τελευταίο παροχέα.

### **Upload στον μεσολαβητή**

Η αντιγραφή των αρχείων στον μεσολαβητή γίνεται με ένα applet. Όπως γνωρίζουμε αυτή τεχνολογία είναι πολύ επικίνδυνη να χρησιμοποιείτε σε υπολογιστές γιατί έχει τόσες πολλές δυνατότητες που μπορεί να κάνει καταστροφικές ζημιές στα μηχανήματα που τα εκτελούν, αν κάποιο applet προέρχεται από κάποιον που θέλει να κάνει ζημία. Εάν όμως το applet προέρχεται από γνωστό δημιουργό και είναι ασφαλές μπορεί να βοηθήσει και να βελτιώσει την χρήση ενός προγράμματος όπως ο μεσολαβητής και να διευκολύνει τους χρήστες του.

Χρησιμοποιήθηκε applet για την αποστολή των αρχείων στον μεσολαβητή γιατί με τις σημερινές τεχνολογίες δεν υπάρχει άλλος τρόπος για τον έλεγχο και προσπέλαση του σκληρού δίσκου μέσω internet. Τα applets είναι πολύ καλά εργαλεία και μπορούν να κάνουν σχεδόν τα πάντα σε έναν υπολογιστή και γι' αυτό τον λόγο πολλά λειτουργικά δεν συνιστούν την λειτουργία του και την μπλοκάρουν.

Το συγκεκριμένο applet παίρνει σαν είσοδο το path που βρίσκονται τα αρχεία και ελέγχει τον συγκεκριμένο φάκελο καθώς και τους υποφακέλους και ανεβάζει στον μεσολαβητή τα αρχεία που του έχουμε ορίσει, χωρίς να μπαίνει στην διαδικασία ο χρήστης σε μια ρουτίνα να αποστέλλει ένα ένα τα αρχεία, πράγμα πολύ χρονοβόρο.

Έτσι χρησιμοποιώντας μια τεχνολογία όπως τα applet μειώνουμε τον χρόνο αποστολής των αρχείων στο ελάχιστο και πλέον εξαρτάτε εξολοκλήρου από την ταχύτητα του δικτύου, δηλαδή το bandwidth που έχει ο χρήστης προς τον μεσολαβητή.

### **Upload στον παροχέα**

Μετά την αντιγραφή των αρχείων στον μεσολαβητή τρέχει μια κλάση που ελέγχει τα αρχεία και κάνει τους απαραίτητους υπολογισμούς που χρειάζονται για να γίνει το tender. Έπειτα όταν βρεθούν οι παροχές που θα σταλούν τα αρχεία γίνεται το upload σε αυτούς τους.

Αυτό γίνεται αυτοματοποιημένα από τα jsp που τρέχουν συνέχεια μέχρι να αποσταλούν όλα τα αρχεία στους μεσολαβητές. Πιο συγκεκριμένα εφόσον ξέρουμε πια αρχεία θα σταλούν και σε ποιους παροχές ή παροχέα που θα σταλούν καλείτε η κλάση που τα αποστέλλει μέχρι να στείλει και το τελευταίο.

## Run

Μετά το τέλος του upload πρέπει να δοθεί η εντολή να ξεκινήσει η επεξεργασία των δεδομένων που έχουν αποσταλεί. Αυτό γίνεται αυτόματα με την ίδια εντολή του GRIA σε κάθε παροχέα αφού τα ονόματα των αρχείων και των παροχέων που έχουν αποσταλεί τα γνωρίζει ο μεσολαβητής από την βάση δεδομένων που έχουν αποθηκευτεί.

## Εγκατάσταση του μεσολαβητή σε διάφορα λειτουργικά συστήματα

Καθώς η δημιουργία του μεσολαβητή έγινε σε περιβάλλον Windows, έτσι η εξέλιξη και βελτίωσή του έγινε στο ίδιο λειτουργικό. Καθώς αναλύθηκαν τα περιφερειακά εργαλεία που χρησιμοποιήθηκαν τρέχουν σε όλα τα λειτουργικά, δοκιμάστηκε με επιτυχία και σε περιβάλλον Linux.

Η διαδικασία όμως εγκατάστασης του μεσολαβητή είναι πιο πολύπλοκη του GRIA client και αυτό οφείλεται στις παραπάνω λειτουργίες που έχουν ενσωματωθεί. Αν αναλογιστούμε όμως ότι η συγκεκριμένη εγκατάσταση πρέπει να γίνει σε έναν μόνο υπολογιστή, τον εξυπηρετητή, είναι πολύ ευκολότερη η εγκατάσταση αν αθροίσουμε τις εγκαταστάσεις που θα γινόντουσαν σε όσους υπολογιστές ήθελαν να τρέχουν την εφαρμογή. Άρα η απλοποίηση της εγκατάστασης έχει επιτευχθεί.

Η εφαρμογή GRIA θα πρέπει να εγκατασταθεί όπως επίσης και το JRE για να τρέχει στο μηχάνημα που θέλει. Η βάση δεδομένων και ο εξυπηρετητής που πρέπει να εγκατασταθούν στο μηχάνημα που θα τρέχει, είναι απαραίτητα για την χρήση του εξυπηρετητή. Άρα η πολυπλοκότητα της εγκατάστασης οφείλεται στην χρήση του μεσολαβητή. Οι παραπάνω βελτιώσεις του μεσολαβητή δεν χρειάζονται κάτι παραπάνω γιατί είναι απλά κάποιες σελίδες και κλάσεις προστεθειμένες στον μεσολαβητή. Έτσι η βελτίωση του εξυπηρετητή επιτεύχθηκε χωρίς την ανάγκη επιπρόσθετων εργαλείων και τεχνολογιών, γεγονός που μας εξυπηρετεί για την εγκατάσταση και χρήση του μεσολαβητή της GRIA.

Ένα θέμα το οποίο μπορεί να είναι λειτουργικό σε μερικές περιπτώσεις είναι ότι η βάση δεδομένων δεν χρειάζεται να βρίσκεται στον ίδιο υπολογιστή με την GRIA και τον μεσολαβητή, αλλά σε κάποιον άλλον υπολογιστή που ανήκει στο ίδιο δίκτυο αφού οι ρυθμίσεις του μεσολαβητή για την βάση είναι δυναμικές και μπορούν εύκολα να αλλαχτούν μέσω ενός jsr. Αυτό μπορεί να είναι προς όφελος μερικών εταιριών που θα τρέχουν τον μεσολαβητή στον server τους. Η βάση μπορεί να βρίσκεται σε έναν server που τρέχει την βάση δεδομένων της εταιρίας και να γίνονται κάποιες ρυθμίσεις στην βάση ευκολότερα. Ο μεσολαβητής θα επικοινωνεί με την βάση αφού οριστεί το IP του υπολογιστή που θα τρέχει η βάση και άλλα στοιχεία όπως το όνομα της και τα απαραίτητα username και password.

Έτσι με απλές κινήσεις και λιγότερο πολυπλοκότερες από την εγκατάσταση της ίδιας της εφαρμογής μπορεί να τρέξει ο μεσολαβητής σε οποιοδήποτε περιβάλλον εκτός από αυτό που χρησιμοποιήθηκε για την βελτίωση και υλοποίηση του. Όπως αναφέραμε η εγκατάστασή του σε Linux δεν είχε καμία διαφορά στις ρυθμίσεις του από τα Windows όπου δημιουργήθηκε και εξελίχθηκε ο μεσολαβητής.

### **Αποτελέσματα χρήσης του μεσολαβητή**

Στόχος του μεσολαβητή ήταν η εξάλειψη κάποιων μειονεκτημάτων που είχε η εφαρμογή. Αφού εντοπιστήκαν και καταγράφηκαν, βρέθηκαν τρόποι και μέθοδοι για την εξάλειψή τους. Κυρίως οι δυσκολίες ήταν προς τον τελικό χρήστη και όχι τόσο στην εγκατάσταση του. Οι ρυθμίσεις που έπρεπε να γίνουν για την σωστή χρήση και εκτέλεση της εφαρμογής ήταν δεκάδες και δύσκολες να γίνουν από έναν άπειρο χρήστη. Στην συνέχεια το περιβάλλον της GRIA δεν ήταν καθόλου φιλικό στον χρήστη αφού το command line έχει ξεπεραστεί πολλά χρόνια τώρα. Ένας ακόμα στόχος ήταν και η αλλαγή του με ένα άλλο περιβάλλον φιλικότερο στους τελικούς χρήστες του GRIA. Επιπροσθέτως η δημιουργία των απαραίτητων xml ήταν ένα μειονέκτημα από άποψης χρόνου και λειτουργικότητας καθώς η χειροκίνητη δημιουργία τους εγκυμονεί κινδύνους για λάθη συντακτικά και άλλα. Καθώς οι χρήστες είναι συνηθισμένοι σε εύχρηστα και ευκολότερα περιβάλλοντα

και παρόλο τις τόσες δυνατότητες που μπορεί να προσφέρει μία εφαρμογή σαν την GRIA θα μπορούσε να την άφηνε στο περιθώριο μόνο και μόνο από το θέμα ευχρηστίας.

Έτσι δημιουργήθηκε ο οδηγός για την δημιουργία του μεσολαβητή και ύστερα την βελτίωσή του. Χαρακτηριστικά όπως η ευκολία στην χρήση του και η λειτουργικότητα του ήταν ο πρωταρχικός στόχος του μεσολαβητή. Στην συνέχεια η ευκολία στην εγκατάσταση, η ασφάλεια, η συλλογή στατιστικών και ο έλεγχος χρήσης του μεσολαβητή ήταν κάποια βήματα που ήταν χρήσιμα για μια επαγγελματική εφαρμογή όπως αυτή. Στην συνέχεια προς την βελτίωση του μεσολαβητή για την αυτόματη δρομολόγηση και αυτόματο υπολογισμό του φόρτου εργασίας της εργασίας που ήθελε να αποστείλει κάθε χρήστης ήταν βελτιώσεις που καθιστούσαν τον μεσολαβητή ακόμα ευκολότερο και ταχύτερο στην χρήση του από την ίδια την εφαρμογή.

Καθώς χρησιμοποιήσαμε τον μεσολαβητή στο εργαστήριο παρατηρήσαμε ότι ακόμα και αν εγκαθιστούσαμε τον μεσολαβητή στον ίδιο υπολογιστή που επιθυμούσαμε να τρέξουμε την GRIA, ήταν ευκολότερο και προτιμότερο από την χρήση της GRIA client.

## **Ευχρηστία**

Το θέμα της ευχρηστίας χωρίζεται σε δύο τομείς. Ο τομέας της εγκατάστασης, που γίνεται μια φορά σε έναν εξυπηρετητή, και ο τομέας της χρήσης της εφαρμογής. Και στους δύο τομείς το αποτέλεσμα είναι αποδεκτό.

Βασικός στόχος του μεσολαβητή ήταν η ευκολία στην εγκατάσταση. Εγκαθιστώντας την GRIA client σε κάθε υπολογιστή που ήθελε να χρησιμοποιήσει την εφαρμογή έπρεπε να γίνει από χρήστη έμπειρο σε θέματα Grid και Java. Επίσης έπρεπε να γίνει σωστή ρύθμιση των εργαλείων που χρειαζόντουσαν και των πιστοποιητικών. Πλέον η εγκατάσταση γίνεται μία φορά σε έναν κεντρικό εξυπηρετητή που ο χρήστης που θα το εγκαταστήσει πρέπει να



έχει στοιχειώδη εμπειρία σε θέματα βάσεων δεδομένων και πιο συγκεκριμένα στην MySQL, web services, Java, JRE και Tomcat. Περισσότερη πολύπλοκη η εγκατάσταση του μεσολαβητή αλλά γίνεται μία φορά σε έναν υπολογιστή και όχι σε όσους θέλουν να χρησιμοποιήσουν την GRIA. Η χρήση του μεσολαβητή από τους χρήστες γίνεται μέσω browser, δηλαδή, Internet Explorer ή οποιονδήποτε άλλον internet browser. Έτσι αυτόματα απαλλάσσουμε τον κάθε χρήστη από την εγκατάσταση της GRIA client και οποιαδήποτε άλλης εγκατάστασης αφού όλα τα λειτουργικά συστήματα έχουν browsers ήδη εγκατεστημένα. Παράλληλα η σύνδεση με το internet για την επικοινωνία του μεσολαβητή με τους παροχείς θεωρείται αναγκαία, αλλά αυτό δεν είναι κάτι παραπάνω αφού και με την χρήση του GRIA client η σύνδεση με το διαδίκτυο ήταν υποχρεωτική. Εν ολίγοις ο χρήστης του μεσολαβητή δεν υποχρεούται να γνωρίζει θέματα Java, Grid και MySQL αφού δεν θα έρχεται σε επαφή και να ρυθμίσει τίποτα από αυτά.

Επόμενο θέμα είναι η ευχρηστία του μεσολαβητή και όχι η εγκατάσταση του αφού γίνεται μία μοναδική φορά. Η χρήση του μεσολαβητή γίνεται μέσω πλοηγών μέσω παραθυρικού περιβάλλοντος. Η δυναμική δημιουργία HTML μέσω τεχνολογίας jsp βοήθησε πολύ σε αυτόν τον τομέα. Σε κάθε σελίδα που πλοηγείτε ο χρήστης εξομοιώνει και μια εντολή της GRIA, επιπλέον υπάρχουν δικλίδες ασφαλείας που τον αποτρέπουν από λάθη. Η αλληλεπίδραση των ιστοσελίδων μέσω γραφικών και φορμών με τον χρήστη καθιστούν την χρήση εύκολη και προσιτή σε αντίθεση με την σύνταξη των εντολών που χρειαζότανε η GRIA client. Επίσης η ανάγκη για την σύνταξη αρχείων xml από τον χρήστη είναι αντικατεστημένη με την συμπλήρωση φορμών. Ο χρήστης δίνει τα απαραίτητα στοιχεία για την συμπλήρωση και δημιουργία των xml. Με αυτόν τον τρόπο αποφεύγονται πολλά λάθη που μπορούσαν εύκολα να γίνουν ακόμα και από έμπειρους χρήστες. Τα δεδομένα που συμπληρώνει ο χρήστης σε κάθε φόρμα αποθηκεύονται σε μια βάση δεδομένων και δεν χάνονται ακόμα και αν αποσυνδεθεί και συνεχίσει την δουλειά του αργότερα.

Ακόμα και με την χρήση της αυτόματης δρομολόγησης ο χρήστης πολύ εύκολα και γρήγορα οδηγείτε από το σύστημα να συμπληρώνει τις φόρμες στα αναγκαία πεδία που αδυνατεί να συμπληρώσει ο μεσολαβητής. Ο υπολογισμός του φόρτου εργασίας και η αποστολή εργασίας σε πολλαπλούς παροχείς είναι κομμάτια που

κάνουν μια τέτοια εργασία ταχύ και εύκολη αφού ο χρήστης απλά επιλέγει τον τύπο της αυτόματης δρομολόγησης, το είδος της υπηρεσίας που επιθυμεί και την ημερομηνία παραλαβής της δουλειάς.

Εν τέλει ο μεσολαβητής μαζί με την χρήση της αυτόματης δρομολόγησης έχουν αναβαθμίσει την GRIA client σε ευκολότερη και πιο εύχρηστη εφαρμογή από ότι ήταν με συνέπεια και την ευκολότερη αποδοχή της από τους επαγγελματίες χρήστες της.

### **Ασφάλεια**

Η ασφάλεια στον μεσολαβητή χωρίζεται σε δύο περιπτώσεις. Αρχικά είναι η ασφάλεια που προσφέρει η GRIA από μεριά της σε επικοινωνία του πελάτη και του παροχέα. Η ασφάλεια που έχει η GRIA πέρασε στον μεσολαβητή μέσω των API που χρησιμοποιήσαμε για την δημιουργία του. Δημιουργώντας κάθε πελάτη πιστοποιητικά ασφαλείας και η επικοινωνία μέσω πρωτοκόλλων SOAP καθιστούν την εφαρμογή και κατά συνέπεια και τον μεσολαβητή αδιάβλητο από επιθέσεις και προσπάθειες για οτιδήποτε παράνομη χρήση του. Όπως αναφέραμε και πρωτύτερα για την Java, κληρονόμησε αυτές τις δικλίδες ασφαλείας από την εφαρμογή και πέρασαν και στον μεσολαβητή.

Από την άλλη υπάρχει και ασφάλεια μεταξύ μεσολαβητή και απλού χρήστη. Η χρήση του δικτυακού τόπου γίνεται από μέσα από μια σελίδα που είναι αναγκασμένος ο κάθε χρήστης να εισάγει ένα όνομα και έναν κωδικό. Αυτά επαληθεύονται με τα δεδομένα της βάσης δεδομένων και ο χρήστης εισάγεται στον τόπο. Όταν γίνει αυτό ένα αρχείο δημιουργείτε στον εξυπηρετητή και στον χρήστη, γνωστά ως session και cookie αντίστοιχα. Μέσω αυτών γίνεται ο έλεγχος αν ο χρήστης έχει λογαριασμό στον δικτυακό τόπο του μεσολαβητή και πια τα δικαιώματά του.

## **Διαχείριση**

Δεν μπορούν όλοι οι χρήστες να κάνουν τα πάντα, γι' αυτό δημιουργήθηκε μια ομάδα χρηστών που είναι οι διαχειριστές του τόπου αυτού που έχουν το δικαίωμα να δημιουργήσουν, να αλλάξουν δικαιώματα ή ακόμα και να διαγράψουν τους υπόλοιπους χρήστες. Κάθε είσοδος ή έξοδος ακόμα κάθε επιλογή του χρήστη καταγράφεται στην βάση δεδομένων που μπορεί ο διαχειριστής να ελέγξει. Οι διαχειριστές έχουν το δικαίωμα να χρησιμοποιήσουν κάθε δυνατή επιλογή του μεσολαβητή, σε αντίθεση με τους άλλους χρήστες. Από την άλλη έχουν ευθύνες για την συντήρηση και την επίβλεψη των στατιστικών που μπορούν να βλέπουν κάθε φορά από τις σελίδες που μόνο εκείνοι έχουν το δικαίωμα να χρησιμοποιούν.

## **Στατιστικά**

Η συλλογή των στατιστικών, είναι ένα εργαλείο αποκλειστικά του μεσολαβητή, που γίνεται με την αποθήκευση, των επιλογών και των κινήσεων των χρηστών, στην βάση δεδομένων που μπορεί οποιαδήποτε στιγμή να ελέγξει ο διαχειριστής. Αυτό γίνεται για τον εντοπισμό αδυναμιών στην χρήση του μεσολαβητή και συνεπώς την περαιτέρω βελτίωσή του. Ακόμα η συλλογή των στατιστικών γίνεται και για λόγους ασφάλειας της εταιρίας από τον κάθε εργαζόμενο της που μπορεί, αν δεν ελεγχόταν, να εκτελούσε εργασίες για λογαριασμό διαφορετικό από της εταιρίας.

## **Αυτόματη δρομολόγηση**

Η αυτόματη δρομολόγηση στον μεσολαβητή είναι ένα ακόμα εργαλείο που ανεβάζουν την ευχρηστία του τόπου αφού εκτός από τον αυτόματο έλεγχο και την επιλογή του καλύτερου δυνατού παροχέα που κάνει ο μεσολαβητής κάνει και άλλους αναγκαίους υπολογισμούς. Αντιγράφοντας τα αρχεία που θέλει ο χρήστης να στείλει προς επεξεργασία στον μεσολαβητή γίνεται αυτόματα ο υπολογισμός του μεγέθους τους, ο ελεύθερος χώρος που απαιτείτε στον παροχέα ακόμα και το φόρτος εργασίας. Αυτές είναι λειτουργίες που ήταν αναγκασμένος ο χρήστης να κάνει πριν επικοινωνήσει με κάποιον παροχέα και να αποστείλει αυτά τα δεδομένα που είναι χρονοβόρα να τα βρει ένας χρήστης, ακόμα και έμπειρος. Ακόμα η αυτόματη συμπλήρωση των φορμών, η επιλογή του τελικού παροχέα ή των τελικών παροχέων και στην συνέχεια η συμπίεση και αποστολή των αρχείων είναι κινήσεις αυτοματοποιημένες που γίνονται από τον παροχέα που αν γίνονταν από έναν χρήστη θα ήταν υπερβολικά χρονοβόρες.

## **Συμπεράσματα**

Είναι εμφανές ότι η χρήση του μεσολαβητή και η χρήση της αυτόματης δρομολόγησης δεν αναβαθμίζουν απλά την GRIA client αλλά και την τοποθετούν στην κορυφή ανάμεσα σε άλλες παρόμοιες εφαρμογές. Ακόμα και ο τρόπος δημιουργίας του μεσολαβητή καθιστά δυνατόν την αναβάθμισή του και προσθήκη ακόμα περισσότερων εργαλείων για την ευκολότερη και ταχύτερη χρήση της τελικής εφαρμογής.