



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# ΠΑΡΑΛΛΗΛΕΣ ΥΠΟΛΟΓΙΣΤΙΚΕΣ ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΣΚΟΠΟΣ – ΠΡΟΟΔΟΣ – ΑΠΟΤΕΛΕΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΔΡΑΣΕΙΣ

ΚΕΟ-3:

Σπυρίδων Λυκοθανάσης<sup>α</sup>

Αίγλη Κορφιιάτη<sup>α</sup>

Παναγιώτης Αλεφραγκής<sup>β</sup>

Νοε 11, 2015, Χανιά



<sup>α</sup>Computer and Informatics Engineering Dept., Technological Educational Institute of Western Greece, Greece



<sup>β</sup>Computer Engineering and Informatics Department, University of Patras, Greece  
Pattern Recognition Laboratory, Director: S. Likothanassis. Professor, e-mail: likothan@ceid.upatras.gr

# ΠΕΡΙΕΧΟΜΕΝΑ

*1. ΣΚΟΠΟΣ ΔΡΑΣΗΣ 3.1 ΚΑΙ 4.1*

*2. ΠΡΟΟΔΟΣ*

*3. ΑΠΟΤΕΛΕΣΜΑΤΑ*

*4. ΕΦΑΡΜΟΓΗ ΕΝΟΣ ΥΒΡΙΔΙΚΟΥ DMSM ΠΑΡΑΛΛΗΛΟΥ  
MONTE CARLO PDE ΕΠΙΛΥΤΗ*

*5. ΕΦΑΡΜΟΓΗ: ΜΕΘΟΔΟΙ ΧΑΛΑΡΩΣΗΣ ΣΤΗ ΔΙΕΠΑΦΗ  
ΣΕ ΠΑΡΑΛΛΗΛΕΣ ΠΛΑΤΦΟΡΜΕΣ*



## ***1. ΣΚΟΠΟΣ ΔΡΑΣΗΣ 3.1 ΚΑΙ 4.1***

Η Δράση 3 αναφέρεται:

- γενικά στη χρήση σύγχρονων υπολογιστικών αρχιτεκτονικών στην υλοποίηση των αριθμητικών μεθόδων
- του αντίστοιχου επιστημονικού λογισμικού, που θα αναπτύσσεται στα πλαίσια της Δράσης 2 (ιδιαίτερα 2.2 και 2.3) στην ανάπτυξη της πλατφόρμας λογισμικού της Δράσης 4.
- 
- Η υλοποίησή της πραγματοποιείται από δύο διαφορετικές δράσεις (3.1 και 3.2), ανάλογα με την κατηγορία αρχιτεκτονικών που χρησιμοποιούνται.
- 
- Η ΚΕΟ3 έχει αναλάβει την δράση 3.1: Υλοποίηση σε Clusters, Grids και Cloud.

Αντικείμενο της συγκεκριμένης δράσης αποτελεί η αποδοτική υλοποίηση των αριθμητικών μεθόδων για ασυνεχή προβλήματα πολλαπλών πεδίων σε σύγχρονες παράλληλες αρχιτεκτονικές.

Αρχικά η υλοποίηση έλαβε χώρα σε επίπεδο συστοιχιών υπολογιστών (Clusters).

Η υλοποίηση των μεθόδων αφορά:

- τόσο σε ομοιογενή, συμμετρικά σχήματα όπου γίνεται χρήση ενός συγκεκριμένου προγραμματιστικού μοντέλου ανταλλαγής μηνυμάτων (MPI),
- όσο και σε ετερογενή, μη συμμετρικά σχήματα, όπου ο παραλληλισμός των μεθόδων επιτελείται σε πολλαπλά επίπεδα.
- Σύμφωνα με αυτά τα συνεργατικά σχήματα εφαρμόστηκε συνδυασμός μοντέλων ανταλλαγής μηνυμάτων (MPI) με μοντέλα προγραμματισμού κοινής μνήμης (OpenMP, Pthreads) για την περισσότερο ευέλικτη αξιοποίηση των επεξεργαστών πολλαπλών πυρήνων που περιέχονται σε κάθε κόμβο του Cluster.



- ▶ Κατά το επόμενο στάδιο η υλοποίηση έγινε σε περιβάλλον Cloud μέσω χρήσης υπηρεσιών διαδικτύου (web Services).
- ▶ Το λογισμικό αξιοποιεί με καλύτερο τρόπο τους διαθέσιμους υπολογιστικούς πόρους.
- ▶
- ▶ Ιδιαίτερα κατά την περίπτωση όπου υποβάλλονται μέσω του Cloud αιτήσεις για παράλληλη εκτέλεση από περισσότερους του ενός χρήστες.
- ▶ Για το λόγο αυτό στη δράση 4.1 υλοποιήθηκε το μοντέλο Λογισμικό σαν Υπηρεσία (SaaS – Software as a Service).
- ▶ Δίνει τη δυνατότητα ενός φιλικού προς το χρήστη διαδικτυακού γραφικού περιβάλλοντος που μπορεί να χρησιμοποιηθεί όχι μόνο από σταθμούς εργασίας αλλά και από φορητές συσκευές.

- ▶ Βασικοί στόχοι της Δράσης 4.1:
- ▶ σχεδίαση μιας αρχιτεκτονικής η οποία έθεσε τις βάσεις για μια ενοποιημένη προσέγγιση αντιμετώπισης των σύνθετων προβλημάτων
- ▶ αναγκαιότητα σχεδίασης/ανάπτυξης λογισμικού με δυνατότητα ενσωμάτωσης σύγχρονων υπολογιστικών συστημάτων
- ▶ ανάπτυξη ενός λειτουργικού περιβάλλοντος επίλυσης σύνθετων προβλημάτων, το οποίο ενσωματώνει μεθόδους / λογισμικό της Δράσης 3
- ▶ επιτρέπει την εκμετάλλευση των σύγχρονων υπολογιστικών συστημάτων
- ▶ και διευκολύνει την αποδοτική χρήση των λογισμικών μονάδων.
- ▶ Η αρχιτεκτονική αυτή προσαρμόστηκε σε ήδη υπάρχουσα πλατφόρμα ανοικτού λογισμικού (FEniCS)

- ▶ Το περιβάλλον FEniCS αποτελεί και την πλατφόρμα αξιολόγησης των μεθόδων επίλυσης σύνθετων ΜΔΕ και την επικύρωσή τους σε σημαντικά προβλήματα της Περιβαλλοντικής Μηχανικής και της Ιατρικής ( Πρόβλημα υφαλμύρισης και όγκων εγκεφάλου) .
- ▶ Αρχικά έγινε χρήση υπηρεσίας Cloud, από τον ΩΚΕΑΝΟ (υπήρξαν προβλήματα λόγω αναβάθμισης – εγκαταλείφθηκε).
- ▶ Στη συνέχεια εγκαταστάθηκε Cloud υποδομή (με την εγκατάσταση Xen Server της Citrix) στο Εργαστήριο Αναγνώρισης Προτύπων του Τμήματος Μηχανικών Η/Η και Πληροφορικής του Πανεπιστημίου Πατρών.
- ▶ Αποτελείται από μία Intel(R) Xeon(R) CPU E3-1220@3.10GHz με 4 cores και 16GB RAM.
- ▶ Το λειτουργικό το οποίο τρέχει πάνω στο cloud είναι το UBUNTU Server Edition.
- ▶ Η διαδικασία η οποία και χρησιμοποιήθηκε για την δημιουργία των cloud VM αναφέρεται στα παραδοτέα.

- ▶ Στη συγκεκριμένη δράση η υλοποίηση έγινε σε περιβάλλον Cloud μέσω χρήσης υπηρεσιών διαδικτύου (web services).
- ▶ Το παρεχόμενο λογισμικό είναι σε θέση να αξιοποιεί με καλύτερο τρόπο τους διαθέσιμους υπολογιστικούς πόρους, ιδιαίτερα κατά την περίπτωση όπου υποβάλλονται μέσω του Cloud αιτήσεις για παράλληλη εκτέλεση από περισσότερους του ενός χρήστες.
- ▶
- ▶ Στη δράση 4.1 υλοποιήθηκε το μοντέλο Λογισμικό σαν Υπηρεσία (SaaS – Software as a Service):
- ▶
- 1. δυνατότητα ενός φιλικού προς το χρήστη διαδικτυακού γραφικού περιβάλλοντος
- 2. Χρησιμοποιείται από σταθμούς εργασίας
- 3. αλλά και από φορητές συσκευές.

## 2. ΠΡΟΟΔΟΣ ΦΥΣΙΚΟΥ ΑΝΤΙΚΕΙΜΕΝΟΥ

- ▶ 1. Μελέτη σχετικής βιβλιογραφίας
- ▶ 2. Εγκατάσταση και δοκιμαστική λειτουργία Πλατφόρμας FEniCS.
- ▶ 3. Εγκατάσταση Εικονικού Cluster σε OpenStack (Τμήμα Μηχανικών Πληροφορικής – ΤΕΙ Δ. Ελλάδας)
- ▶ 4. Παράλληλη υλοποίηση της μεθόδου Monte Carlo σε Cluster
- ▶ 5. Προετοιμασία FEniCS για παράλληλη υλοποίηση των IR Μεθόδων (π.χ. σύνδεση με βιβλιοθήκες, κ.λ.π.)

- ▶ 6. Παράλληλη υλοποίηση της IR GEO στη FEniCS.
- ▶ 7. Παράλληλη υλοποίηση της IR ROB στη FEniCS.
- ▶ 8. Σύγχρονη / ασύγχρονη υλοποίηση της IR GEO.
- ▶ 9. Υλοποίηση Μεθόδων Χαλάρωσης στη Διεπαφή σαν Υπηρεσία (IRaaS).



# ΑΠΟΤΕΛΕΣΜΑΤΑ – ΔΡΑΣΗ 3.1

## 1. Σειριακή και Παράλληλη Υλοποίηση της IR – GEO

- ▶ Η μέθοδος *IR – GEO*, υλοποιήθηκε στην *FEniCS* – πρώτο βήμα για τη δημιουργία ενός ολοκληρωμένου Περιβάλλοντος Επίλυσης Προβλημάτων (Π.Ε.Π.) πολλαπλών πεδίων/ πολλαπλών φυσικών.
- ▶ Μία από τις κύριες προκλήσεις της υλοποίησης ήταν η πρόσβαση στις τιμές των διεπαφών και η επιβολή νέων ανανεωμένων τιμών στις διεπαφές.
- ▶ Για την αξιολόγηση της συγκεκριμένης υλοποίησης, χρησιμοποιούνται δύο διαφορετικά προβλήματα ΜΔΕ.
- ▶ Επίσης, έγινε μία παράλληλη υλοποίηση *GEO IR* στη *FEniCS*, με βάση το *RabbitMQ* για την ανταλλαγή μηνυμάτων και η σύγκριση της απόδοσής της με την σειριακή υλοποίηση.
- ▶ Τα αποτελέσματα αποδεικνύουν τη χρησιμότητα της παράλληλης υλοποίησης στην κατεύθυνση της επίλυσης μεγάλων προβλημάτων πολλαπλών πεδίων/ πολλαπλών φυσικών.

## 2. Κατανεμημένη Υλοποίηση της IR –ROB

- ▶ Η μέθοδος ROB είναι μία IR μέθοδος που χρησιμοποιεί Robin συνθήκες για τη λείανση των τιμών στις διεπαφές.
- ▶ Η ROB υλοποιήθηκε στην πλατφόρμα FEniCS και χρησιμοποιεί το RabbitMQ για την ανταλλαγή μηνυμάτων.
- ▶ Μειώνει σημαντικά το συνολικό χρόνο εκτέλεσης.
- ▶ Επίσης, μειώνει σημαντικά την επιβάρυνση της επικοινωνίας, καθώς απαιτείται μόνο ένας περιορισμένος αριθμός μηνυμάτων ανάμεσα στους κόμβους που χειρίζονται γειτονικά πεδία.
- ▶ Η πειραματική αξιολόγηση καταδεικνύει την επεκτασιμότητα και αποτελεσματικότητά της για την επίλυση μεγάλων προβλημάτων πολλαπλών πεδίων/ πολλαπλών φυσικών.
- ▶ Πειραματικά αποτελέσματα σε ένα πρόβλημα μοντέλο με μεταβαλλόμενη διακριτοποίηση δείχνουν επιταχυνόμενο ρυθμό σύγκλισης, με τη ROB να μειώνει τον αριθμό επαναλήψεων κατά μέσο όρο στο μισό συγκρινόμενη με τη GEO IR ενώ την ίδια στιγμή έχει συγκρίσιμο χρόνο εκτέλεσης.

### 3. Εφαρμογή ενός Υβριδικού Παράλληλου Monte Carlo PDE Solver σε Ορθογώνια Πολλαπλά Πεδία

- ▶ Αξιοποίηση της απόδοσης ενός cluster πολλαπλών πυρήνων (εκτέλεση ενός profiling της ακολουθιακής υλοποίησης για τον εντοπισμό hotspots στον κώδικα που θα μπορούσαν να οδηγήσουν σε μείωση του χρόνου εκτέλεσης, αν γίνει παράλληλη υλοποίηση αποτελεσματικά.)
- ▶ Η ανάλυση απόδοσης βασίστηκε στην υλοποίηση των αλγορίθμων που προτείνονται στο Vavalis et al [4].
- ▶ Λύνει την εξίσωση Poisson με Dirichlet συνοριακές συνθήκες για 2D και 3D υπέρ- ορθογώνια πεδία-επέκταση.

- ▶ Η βασική υλοποίηση [4] ήδη υποστηρίζει παραλληλοποίηση διαμοιραζόμενης μνήμης του Monte Carlo και του Laplace επιλυτή με χρήση Pthreads.
- ▶ Εστιάσαμε στην άμεση υλοποίηση μιας υβριδικής παράλληλης υλοποίησης με χρήση OpenMPI για επικοινωνία μεταξύ διεργασιών.
- ▶ Υλοποιήθηκε απλό master/worker σχήμα για να κατανείμει τον υπολογιστικό φόρτο του Monte Carlo πυρήνα ανάμεσα στις διεργασίες.
- ▶ Η master διεργασία υλοποιεί τους πυρήνες της παρεμβολής και του Laplace επιλυτή για κάθε υποπεδίο τοπικά με χρήση threads, αφού είχε πρώτα λάβει την απαιτούμενη εκτίμηση των σχετικών κόμβων από τις άλλες διεργασίες.
- ▶ Η λογική του βασικού κώδικα [4] μένει σχετικά απaráλλακτη.
- ▶ Η υβριδική υλοποίηση καταφέρνει να επιδείξει σημαντική επιτάχυνση, ενώ διατηρεί την ποιότητα της λύσης σε δείγματα συνόλων δεδομένων εισόδου

# ΑΠΟΤΕΛΕΣΜΑΤΑ – ΔΡΑΣΗ 4.1

*IRaaS, ένα βασισμένο στο cloud περιβάλλον επίλυσης προβλημάτων πολλαπλών πεδίων/ πολλαπλών φυσικών*

- ▶ *Βασίζεται στην παράλληλη υλοποίηση της GEO και ROB IR με χρήση της FEniCS και το RabbitMQ.*
- ▶ *Η διαδικασία της επίλυσης των προβλημάτων πολλαπλών πεδίων/ πολλαπλών φυσικών πραγματοποιείται στην cloud υποδομή, σαν SaaS εφαρμογή.*
- ▶ *Οι χρήστες: (i) ορίζουν σύνθετες ΜΔΕ πολλαπλών φυσικών, (ii) επιλέγουν τους κατάλληλους επιλυτές για τα πεδία και τη GEO/ROB IR μέθοδο για τις διεπαφές και (iii) έχουν πρόσβαση στην υπολογισμένη λύση στο συνολικό πρόβλημα.*
- ▶ *Παράλληλα, ένας μηχανισμός αποδοτικής κατανομής και διαχείρισης πετυχαίνει τρεις σημαντικούς στόχους.*
- ▶ *Κάνει κατανομή του φόρτου στους ελάχιστους δυνατούς πόρους, λύνει το πρόβλημα σε έναν σχεδόν βέλτιστο χρόνο εκτέλεσης και η κατανομή πόρων πραγματοποιείται αυτόματα στο παρασκήνιο, χωρίς την παρέμβαση του χρήστη.*
- ▶ *υπόψη η ποικιλομορφία των υποπροβλημάτων η εφαρμογή μπορεί να υπολογίσει και να παρέχει τα βέλτιστα VMs (σε πλήθος και δυνατότητες) σύμφωνα με τους διαθέσιμους πόρους, το μέγεθος του συνολικού προβλήματος και το μέγεθος των υποπροβλημάτων του.*

# ΔΗΜΟΣΙΕΥΣΕΙΣ

1. **A. Korfiati, P. Tsompanopoulou, S. Likothnassis**, “Serial and Parallel Implementation of the Interface Relaxation Method GEO”, Numerical Analysis Conference (NumAn 2014), September 2–5, Chania, Greece, 2014.
2. **P. Alefragis, A. Spyrou and S. Likothanassis**, “Application of a hybrid parallel Monte Carlo PDE Solver on rectangular multi-domains” Numerical Analysis Conference (NumAn 2014), September 2–5, Chania, Greece, 2014.
3. **Aigli Korfiati, Niki Sfika, Konstantis Daloukas, Christos Alexakos, Panagiota Tsompanopoulou and Spiros Likothanassis**, “IRaaS: A Cloud Implementation of an Interface Relaxation Method for the Solution of PDEs”, 2015 International Conference of Parallel and Distributed Computing (ICPDC'15), part of World Congress on Engineering 2015 (WCE 2015). (Certificate of Merit).
4. **Niki Sfika, Aigli Korfiati, Christos Alexakos, Spiros Likothanassis, Konstantis Daloukas and Panagiota Tsompanopoulou**, “Dynamic Cloud Resources Allocation on Multidomain/ Multiphysics Problems”, The 3rd International Conference on Future Internet of Things and Cloud (FiCloud 2015), 24–26 August 2015, Rome, Italy.
5. **A. Korfiati, K. Daloukas, P. Alefragis, P. Tsompanopoulou and S. Likothanassis**, “An Ansynchronpus Interface Relaxation Method for Multi-domain / Multi-Physics Problems”, International Conference on Numerical Analysis Applications and Methods (ICNAAM), Rhodes, Greece, Sept. 23–29, 2015.



# Άλλα Αποτελέσματα

1. *Ο κώδικας των παράλληλων υλοποιήσεων των μεθόδων και η Εφαρμογή IRaaS.*
2. *Υπό συγγραφή 2 επιστημονικά άρθρα για υποβολή σε διεθνή επιστημονικά περιοδικά.*
3. *Μία δημοσίευση σε συνεργασία με μέλη του εργαστηρίου Α.Π.: N. Sfika, A.Korfiati, C. Alexakos, S. Likothanassis, “Workflow coordinated resources allocation for Big Data analytics in the cloud”, 2nd Workshop on “New Methods and Tools for Big Data (MT4BD-2015)”, part of 11th International Conference on Artificial Intelligence Applications and Innovations (AIAI 2015), 14–17 September 2015, Bayonne/Biarritz – France.*
4. *Μια Μ.Δ.Ε. με τίτλο: «Σχεδιασμός και υλοποίηση προηγμένων μαθηματικών μεθόδων για την επίλυση προβλημάτων πολλαπλών πεδίων σε σύγχρονες υπολογιστικές αρχιτεκτονικές», Α. Κορφιάτη, Μ.Π.Ε. “Επιστήμη και Τεχνολογία Υπολογιστών”, του Τμήματος Μηχανικών Η/Υ και Πληροφορικής του Πανεπιστημίου Πατρών.*
5. *Σε εξέλιξη μια Διδακτορική διατριβή, με θέμα: «Σχεδιασμός και υλοποίηση ολοκληρωμένου περιβάλλοντος επίλυσης προβλημάτων σε αρχιτεκτονικές υψηλής επίδοσης και εφαρμογές», Α. Κορφιάτη, Μ.Π.Ε. “Επιστήμη και Τεχνολογία Υπολογιστών”, του Τμήματος Μηχανικών Η/Υ και Πληροφορικής του Πανεπιστημίου Πατρών.*



# Υβριδική DMSM Παράλληλη υλοποίηση Monte Carlo PDE Επιλυτή

# Περιγραφή Σειριακού Αλγορίθμου

- Η υλοποίηση παράλληλης διαμοιραζόμενης μνήμης προέρχεται από G. Sarailidis and M. Vavalis<sup>1</sup>.
- Ψευδοκώδικας:

*Υπολόγισε τις συντεταγμένες των κόμβων σε όλες τις υποπεριοχές  
Για κάθε κόμβο*

*Παρήγαγε μία εργασία*

*Ανέθεσε την εργασία σε ένα νήμα*

*Εκτέλεσε Monte Carlo να εκτιμήσεις την τιμή του κόμβου*

*Χρήσιμοποίησε 2D|3D Interpolation για την εύρεση νέων ορίων*

*Για κάθε περιοχή*

*Παρήγαγε μια εργασία*

*Ανέθεσε την εργασία σε ένα νήμα*

*Λύσε το υποπρόβλημα με χρήση επιλυτή Laplace*

*Παρήγαγε τα αποτελέσματα*

1. G. Sarailidis and M. Vavalis, "Hybrid Solvers for Elliptic PDEs on rectangular multi-domains in 2 and 3 dimensions", submitted to Scientific Computing, 2013

# Χρονική ανάλυση βασικού αλγορίθμου

- Πραγματοποιήθηκε ανάλυση απόδοσης για τα ποσοστά του χρόνου εκτέλεσης στα 3 τμήματα της
  - Αλγόριθμος Monte Carlo (MC)
  - Αλγόριθμος Interpolation (I)
  - Επιλυτής Laplace (LS)

| Walks   | 5K  |     |     | 20K |     |    | 80K |    |    |
|---------|-----|-----|-----|-----|-----|----|-----|----|----|
| Problem | MC  | I   | LS  | MC  | I   | LS | MC  | I  | LS |
| 1       | 60% | -   | 5%  | 79% | -   | 4% | 95% | -  | 5% |
| 2       | 50% | -   | 10  | 76% | -   | 3% | 90% | -  | 5% |
| 3       | 74% | -   | 9%  | 88% | -   | 4% | 92% | -  | 4% |
| 4       | 80% | 13% | 7%  | 85% | 11% | 4% | 88% | 5% | 7% |
| 5       | 70% | 20% | 10% | 83% | 9%  | 8% | 90% | 6% | 4% |
| 6       | 78% | 12% | 10% | 84% | 10% | 6% | 94% | 3% | 1% |
| 7       | 74% | 10% | 6%  | 81% | 11% | 8% | 90% | 6% | 4% |

## Υβριδική Παραλληλοποίηση κατανεμημένης/ διαμοιραζόμενης μνήμης σε εικονική συστάδα υπολογιστών

- Στόχος η χρήση υπολογιστικών πολυπύρηνων διασυνδεδεμένων υπολογιστών με χρήση MPI για την επικοινωνία ανάμεσα στις διεργασίες.
- Χρήση μοντέλου διαχειριστή/εργαζόμενου για την παραλληλοποίηση του MC.
- Η διεργασία διαχειριστή υλοποιεί τις I και LS.

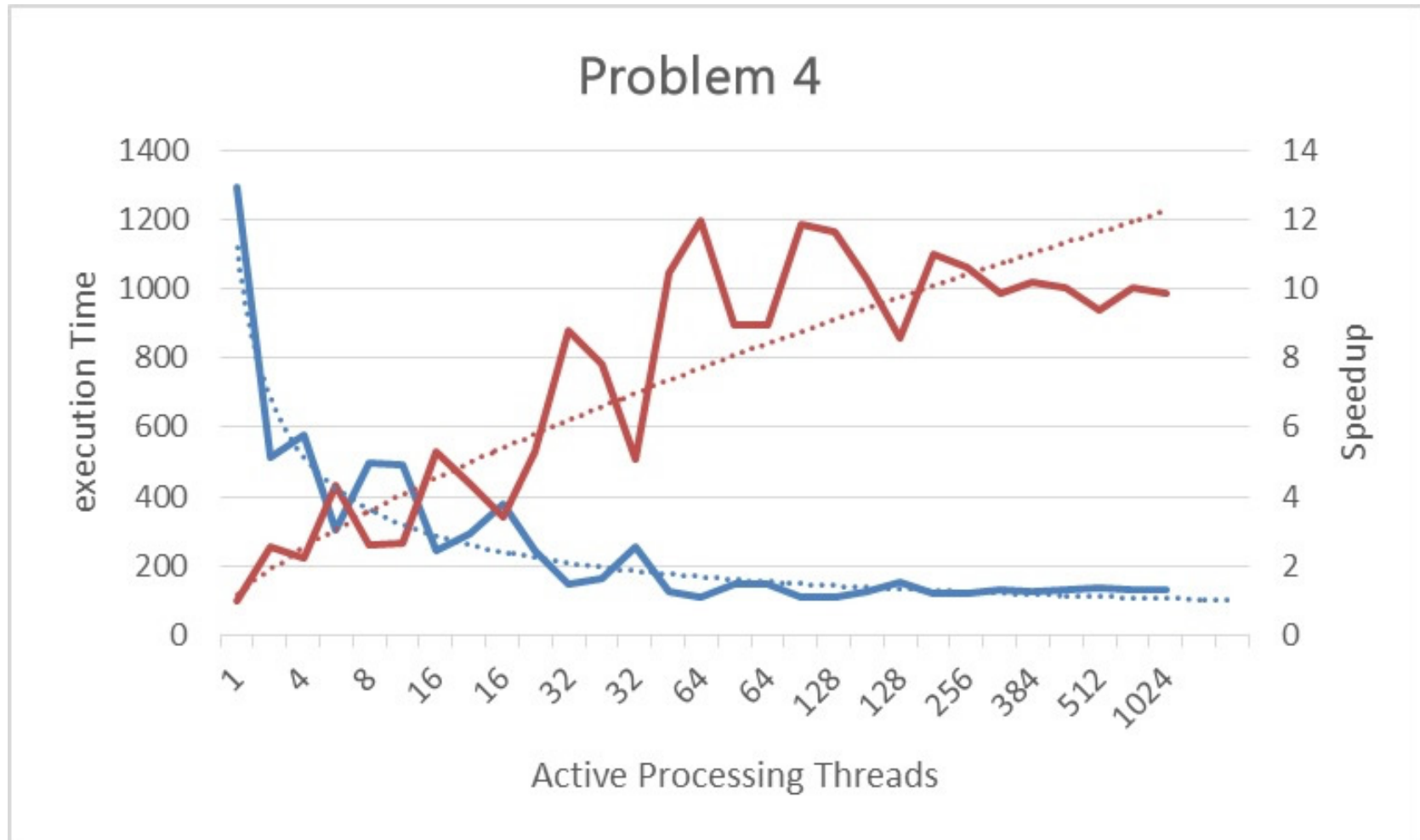
# Πειραματική εγκατάσταση

- Υλοποιήθηκε μια εικονική παράλληλη συστάδα με 8 υπολογιστές με 4 vCPU η κάθε μία πάνω από μία εγκατάσταση OpenStack Cloud.
- Για όλα τα διαθέσιμα προβλήματα δημιουργήσαμε διάφορους συνδυασμούς
  - Από μέγιστο πλήθος νημάτων ανά εικονική μηχανή {1,2,4,6,8}
  - Το συνολικό πλήθος διεργασιών εργάτη {1,2,4,8,16,32,64}
  - το πλήθος των τυχαίων διαδρομών ανά κόμβο στον MC {20K, 40K, 80K, 160K}

# Πειραματικά αποτελέσματα

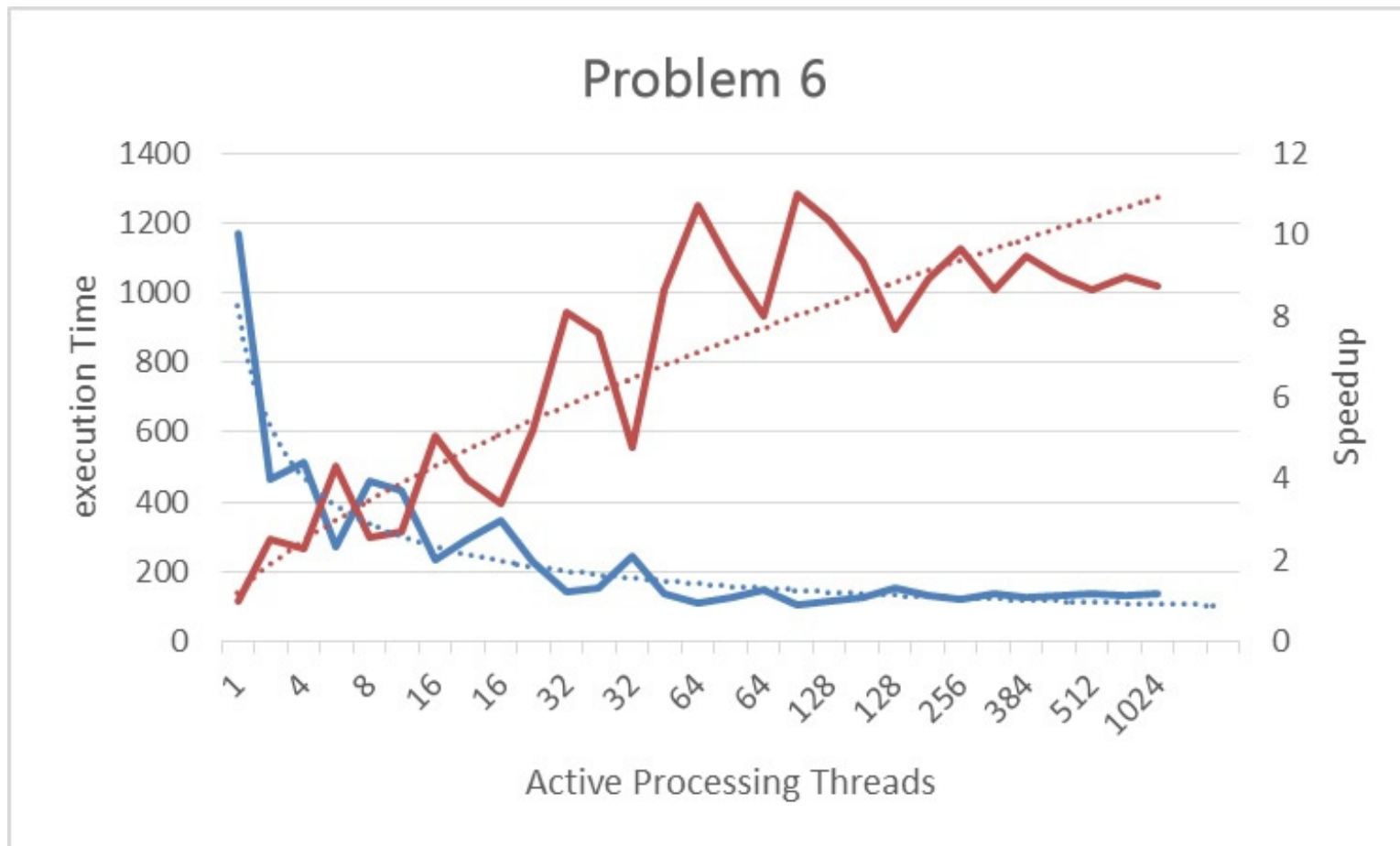
| Problem             | 1    |     |     |     |     |     |     | 2           |      |      |       |       |       |       |
|---------------------|------|-----|-----|-----|-----|-----|-----|-------------|------|------|-------|-------|-------|-------|
| np                  | 1    | 2   | 4   | 8   | 16  | 32  | 64  | 1           | 2    | 4    | 8     | 16    | 32    | 64    |
| nt.<br>Walks        | 1    | 2   | 4   | 6   | 6   | 8   | 8   | 1           | 2    | 4    | 6     | 6     | 8     | 8     |
| 40K                 | 26   | 8   | 6   | 3   | 3   | 4   | 3   | 26          | 9    | 7    | 4     | 3     | 3     | 4     |
| 80K                 | 49   | 14  | 11  | 5   | 4   | 6   | 5   | 50          | 14   | 10   | 6     | 4     |       | 5     |
| 160K                | 98   | 24  | 19  | 9   | 6   | 6   | 8   | 99          | 25   | 19   | 9     | 6     | 7     | 9     |
| Problem             | 3    |     |     |     |     |     |     | 4           |      |      |       |       |       |       |
| 40K                 | 25   | 10  | 5   | 3   | 3   | 2   | 3   | 538         | 183  | 120  | 70    | 62    | 68    | 77    |
| 80K                 | 40   | 26  | 10  | 4   | 3   | 4   | 4   | 931         | 298  | 170  | 88    | 80    | 86    | 90    |
| 160K                | 99   | 38  | 18  | 8   | 5   | 6   | 6   | 1777        | 511  | 295  | 124   | 109   | 122   | 129   |
| Problem             | 5    |     |     |     |     |     |     | 6           |      |      |       |       |       |       |
| 40K                 | 534  | 183 | 120 | 70  | 62  | 68  | 74  | 494         | 176  | 423  | 78    | 72    | 75    | 77    |
| 80K                 | 946  | 298 | 170 | 88  | 180 | 86  | 90  | 847         | 273  | 164  | 97    | 82    | 92    | 94    |
| 160K                | 1757 | 511 | 295 | 124 | 109 | 122 | 129 | 1550        | 464  | 292  | 135   | 106   | 121   | 130   |
| Problem             | 7    |     |     |     |     |     |     | Problem Set |      |      |       |       |       |       |
| 40K                 | 365  | 176 | 109 | 78  | 72  | 75  | 77  | 2008        | 745  | 610  | 306   | 277   | 295   | 312   |
| 80K                 | 713  | 273 | 164 | 97  | 82  | 92  | 94  | 3576        | 1196 | 699  | 385   | 335   | 371   | 382   |
| 160K                | 1416 | 464 | 292 | 135 | 105 | 121 | 130 | 6796        | 2037 | 1230 | 544   | 446   | 505   | 541   |
| Problem Set Speedup |      |     |     |     |     |     |     | Min         | 2.81 | 2.03 | 6.33  | 6.86  | 6.59  | 6.42  |
|                     |      |     |     |     |     |     |     | Avg         | 3.08 | 4.17 | 8.85  | 10.60 | 9.53  | 9.12  |
|                     |      |     |     |     |     |     |     | Max         | 3.34 | 5.13 | 11.48 | 14.62 | 12.81 | 11.92 |

# Πειραματικά αποτελέσματα

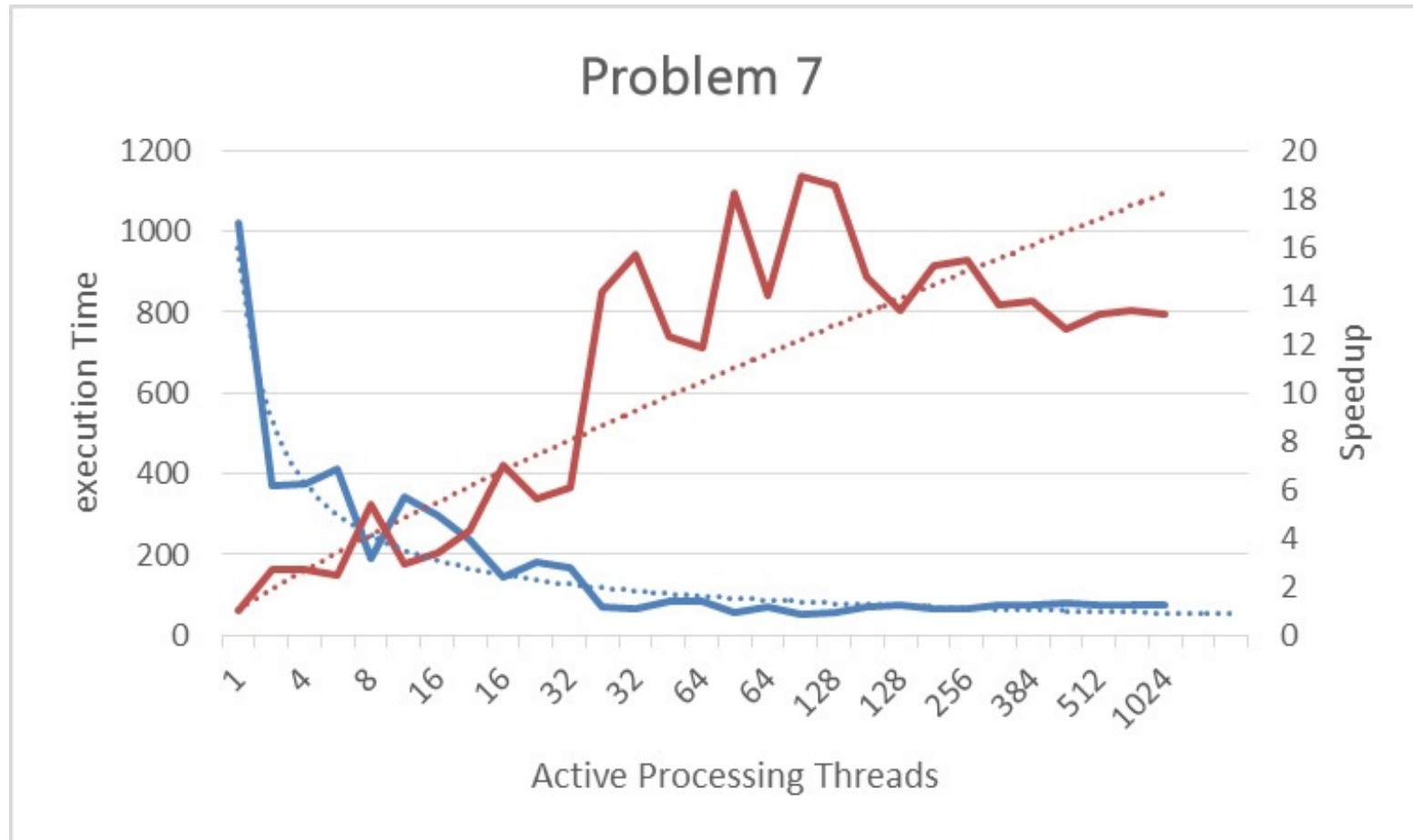




# Πειραματικά αποτελέσματα



# Πειραματικά αποτελέσματα



# Πειραματικά αποτελέσματα

- Τα καλύτερα αποτελέσματα τα είχαμε με χρήση 128 νημάτων παρά το γεγονός ότι είχαμε 32 εικονικούς επεξεργαστικούς πυρήνες
- Η αυξομείωση στην απόδοση έχει σχέση με το λόγο ανάθεση νημάτων ανά εργασία, με ιδανικό λόγο ανάθεσης 4 νήματα ανά διεργασία

# Dynamic Cloud Resources Allocation on Multidomain/Multiphysics Problems

# IRaaS Application

- The proposed cloud application is a solution environment for multiphysics/multidomain problems
  - implementing the interface relaxation methodology, and
  - utilizing cloud technologies that manage pre-existing hardware, network, operating system and applications.
- The user sets the problem's parameters, chooses the interface relaxation method (GEO/ROB) that fits better to the specific problem and finally gets the problem solution from any place and any device.
- The application dynamically allocates the minimum possible resources automatically in the background without the user's interference.

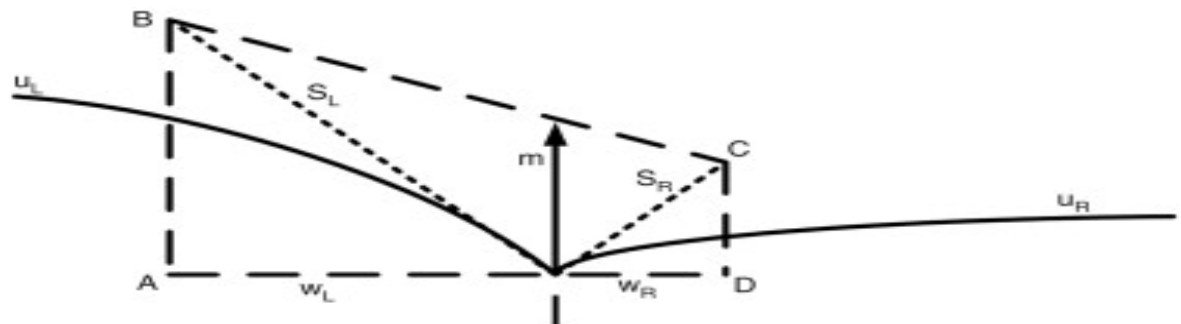
# IR methodology

- Decomposition of the PDE domain into subdomains
  - derived by the physics or for parallelization purposes
- Initial guesses on the interfaces between the subdomains
- Solution of individual sub-problems
- Computation of new values on the interfaces by particular IR methods (forcing the correct conditions for the problem)
- Iterative process until convergence is succeeded

# IR method - GEO

- New relaxed values on the interface points:
  - Add to the old ones a geometrically weighted average of the normal boundary derivatives of the adjacent subdomains.

$$u^{(k+1)}(x) = u^{(k)}(x) - \rho \left( \frac{\partial u_L^{(k)}(x)}{\partial n} - \frac{\partial u_R^{(k)}(x)}{\partial n} \right), k = 0, 1, 2, \dots$$





# IR method - ROB

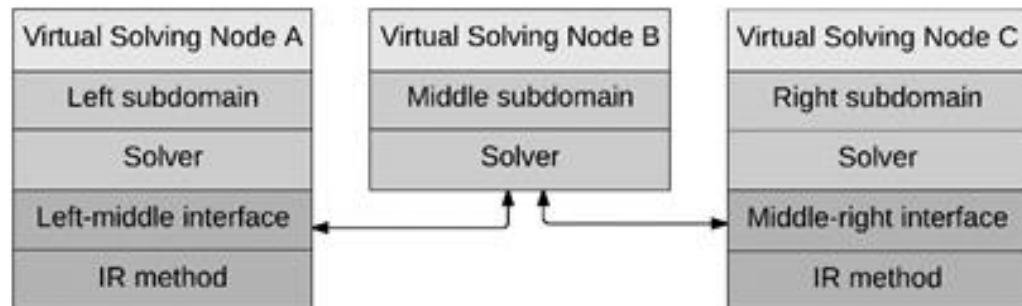
- Based on Robin interface conditions to exchange information across subdomain boundaries
- Solution of the local PDE on the subdomains using Robin conditions on the interfaces by matching a combination of Dirichlet and Neumann data from the neighboring subdomains
- The solution at each iteration  $k$  is given from the following equation:

$$Lu^{(k)} = f \quad \text{with}$$
$$-\frac{\partial u^{(k)}}{\partial x} + \lambda u^{(k)} = -\frac{\partial u_L^{(k-1)}}{\partial x} + \lambda u_L^{(k-1)} \quad \text{on domain's left interface}$$

$$\frac{\partial u^{(k)}}{\partial x} + \lambda u^{(k)} = \frac{\partial u_R^{(k-1)}}{\partial x} + \lambda u_R^{(k-1)} \quad \text{on domain's right interface}$$

# IR distributed solution

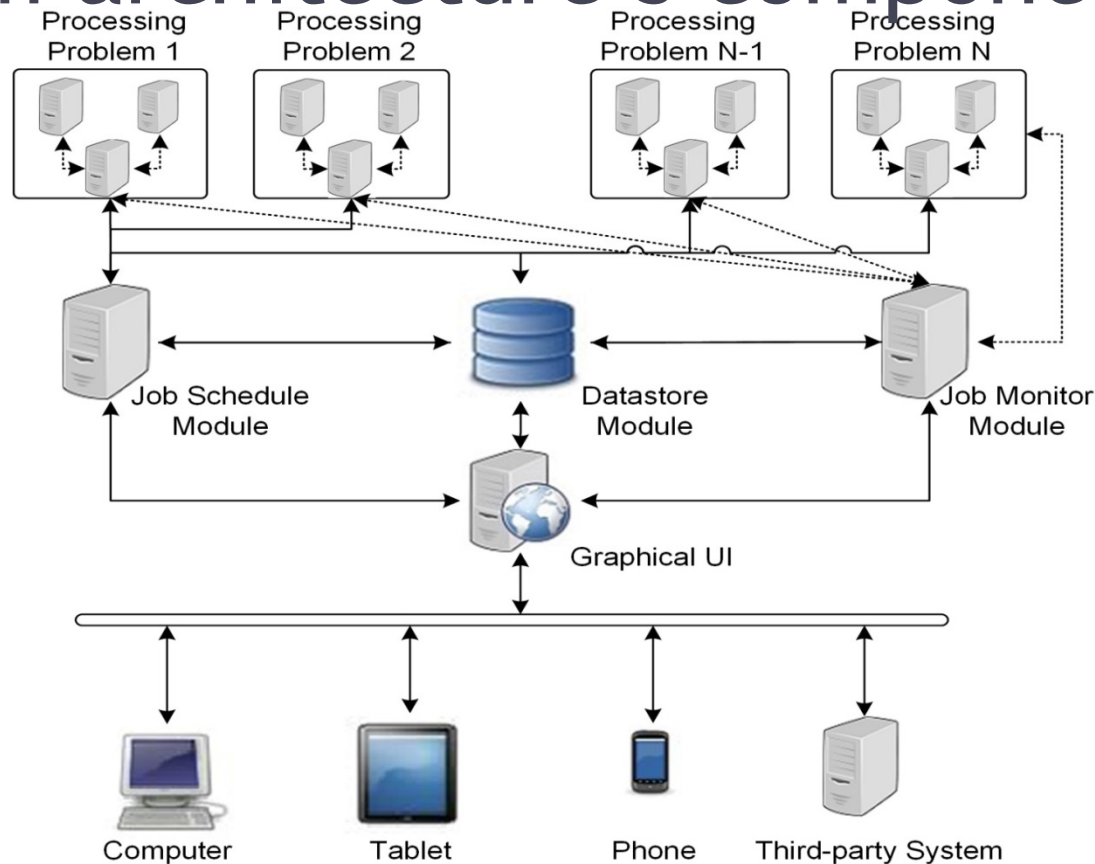
- PDE problem divided in sub-problems
- each has to be expressed as a variational problem.
- Boundary conditions and initial guesses on the interfaces between the sub-problems specified and applied.
- Solution and gradient computed.
- Get the values of the solution and the gradient on the interface points and compute the new relaxed values.
- Pass these new relaxed values back to the sub-problems as updated values of the solution on the interfaces.
- New iteration begins.



# Everything As A Service

- Infrastructure As A Service (IaaS)
  - IaaS allows clients to dynamically rent virtual machines on which they can install their own operating system (OS) and other applications.
- Platform as a Service (PaaS)
  - PaaS allows clients to develop their own system using the platform tools, without having to install and maintain these tools themselves.
- Software as a Service (SaaS)
  - With SaaS, cloud providers install, manage, and operate the software application, and the user has neither knowledge nor control of the underlying infrastructure.

# System architecture's components 1/3



# System architecture's components 2/3

- Graphical UI
  - User Authentication/Login,
  - Graphical interface for adding a new problem,
  - Graphical interface for viewing problems' progress,
  - Graphical interface for viewing results.
- Job Schedule Module
  - Orchestrates problem execution on the cloud infrastructure.
  - Allocates the resources needed by the Virtual Solver Nodes for the problem execution.
  - Deploys the eligible VMs along with their information.
  - Initiates the problems.
  - Destroys the VMs after the execution is finished.

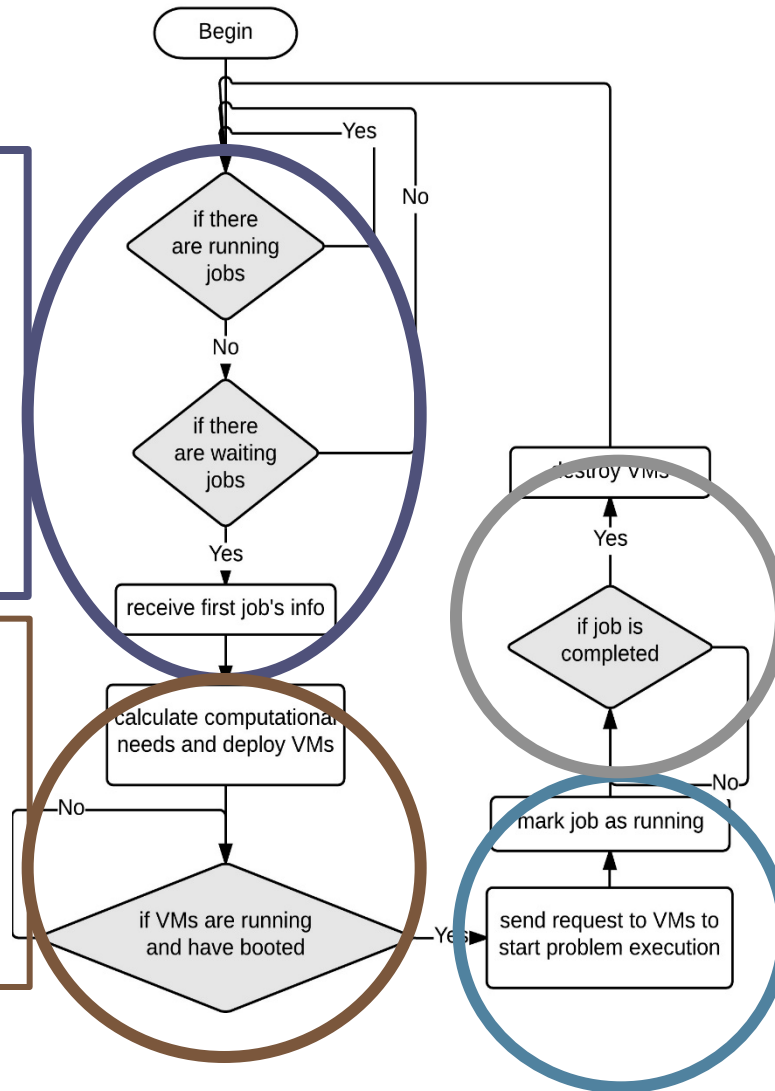
# System architecture's components 3/3

- Job Monitor Module is an Advanced Message Queue Server (AMQS):
  - Handles the communication between the entities of the system.
  - Is based on the Advanced Message Queuing Protocol (AMQP):
    - Connects systems and manages the information and messages exchange between them.
    - In the present work RabbitMQ has been used.
- DataStore Module accessed from all the system components:
  - Stores the input data, the results and the intermediate data of the users' problems.
- Virtual Solving Nodes created by the Job Schedule Module in order to start and perform the problem execution:
  - Inform the Job Monitor Module about the state/progress of the problem.
  - Send their final results back to the Job Monitor Module.

# Job Schedule Module's process

- Checks the DataStore to obtain the required information about the job queue.
- If the job queue is empty, it remains idle.
- If there are pending jobs, it selects the first job in chronological order from the queue and initiates the process for the problem solving

- Calculates computational resources depending on the size of the input data.
- Creates the VMs (based on pre-existing templates including the software required for the problem execution) that will act as the Virtual Solving Nodes.

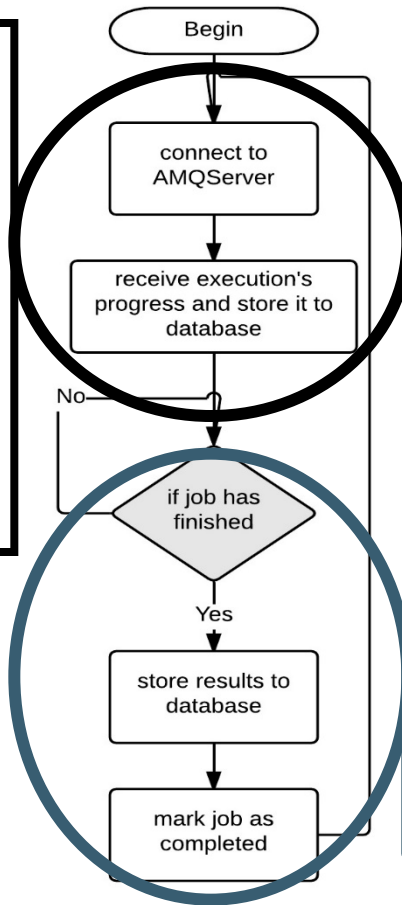


- When the problem has been executed, the Job Schedule Module destroys the created VMs and moves to the execution of a new job in the queue.

- Initiates problem solving process.

# Job Monitor Module's process

- The Virtual Solving Nodes execution progress is monitored by the Job Monitor Module.
- With RabbitMQ, it receives the current iteration of the problem being solved and sends this information to the DataStore for publishing to the users/systems.



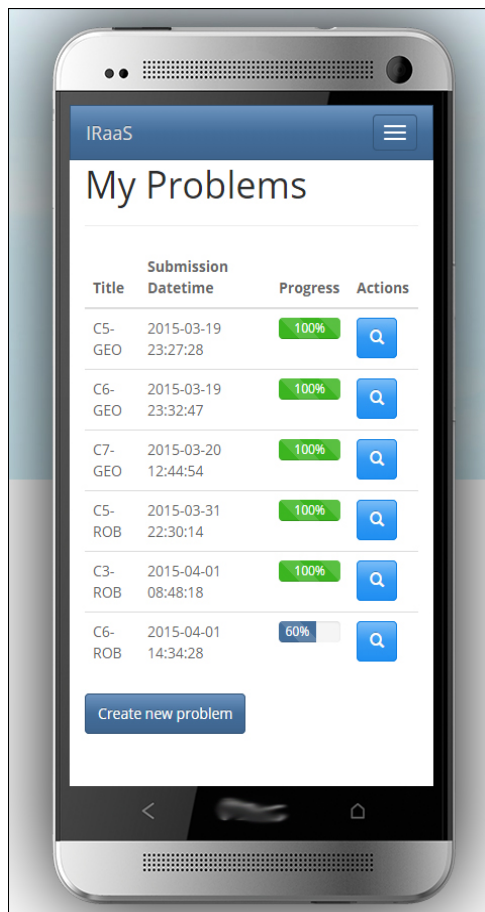
- When the job is finished, it marks the problem as completed and stores the solution in the DataStore.



# Application Interfaces

- The proposed platform is empowered with two interfaces in order to interoperate with the rest ecosystem.
  - The first is an **HTML graphical interface** used by users for submitting processing requests, evaluating their progress and accessing results.
  - The second refers to third-party systems and it is an **Application Programmable Interface (API)** that can be invoked by HTTP requests over Internet.

# Graphical interface designed for many devices



- The graphical user interface is simple and follows **responsive design guidelines** in order to meet the desired level of user friendliness.
- The responsive design is a collection of techniques applied in the HTML code in order for the **application's screens to automatically be adapted to the screen resolution** of the user's device, even in small devices such as tablets and smart phones.
- Thus, the proposed application can be easily used from anywhere with the condition that users have **a mobile or non-mobile device with web browser and internet.**

# Results

- Elliptic problem:

$$Lu(x, y) \equiv -\nabla^2 u(x, y) + \gamma^2 u(x, y) = f(x, y), (x, y) \in \Omega$$
$$u(x, y) = u^b(x, y), (x, y) \in \partial\Omega$$

with  $f(x, y)$ , and  $u^b(x, y)$ , selected such that the true solution is:

$$u(x, y) = e^{y(x+4)} x(x-1)(x-0.7)y(y-0.5)$$

- The test infrastructure comprises a server with an Intel(R) Xeon(R) CPU E3-1220@3.10GHz with 4 cores and 16GB RAM. As a result, our experiments are restricted to a small range of input datasets.

| Problem | Problems' Details        |                  |                    |                   |                             |
|---------|--------------------------|------------------|--------------------|-------------------|-----------------------------|
|         | Discretization Parameter | Left Domain Size | Middle Domain Size | Right Domain Size | Number of Interfaces Points |
| P1      | 0.1                      | 84               | 24                 | 44                | 6                           |
| P2      | 0.05                     | 328              | 88                 | 168               | 11                          |
| P3      | 0.025                    | 1134             | 294                | 574               | 21                          |
| P4      | 0.0125                   | 4508             | 1148               | 2268              | 41                          |
| P5      | 0.00625                  | 17655            | 4455               | 8855              | 81                          |
| P6      | 0.003125                 | 69228            | 17388              | 34668             | 161                         |
| P7      | 0.0015625                | 274134           | 68694              | 137174            | 321                         |

# Results

- For the evaluation of the application, we performed a number of experiments for each test case.
- Experiments were executed with a variety of given computational resources.
- In order to balance the tradeoff between the minimum execution time and resources allocation, the best resources allocation depends exclusively on the input problems.
- We have employed the following rule for the allocation of computational resources:

*if subdomain size  $\leq 75000$ :*  
    *virtual solving node RAM = 1GB*  
    *virtual solving node processor = 1core, 2GHz*  
*else if  $75000 < \text{subdomain size} \leq 150000$ :*  
    *virtual solving node RAM = 2GB*  
    *virtual solving node processor = 1core, 2GHz*  
*else if subdomain size  $> 150000$ :*  
    *virtual solving node RAM = 4GB*  
    *virtual solving node processor = 1core, 2GHz*

# Results

- As we can observe from the next table the proposed implementation provides a significant reduction in execution time compared to the previous virtual implementation.
- The reduction factor increases further with the increase of the problem size, which is a testament of the efficiency of the proposed methodology for the solution of large-scale problems.
- In addition, the application provides an important gain in resource utilization cost for every test case. As a result, both time and resource usage reduction has been achieved.

| Problem | Comparison Analysis        |                                   |                            |                                   |
|---------|----------------------------|-----------------------------------|----------------------------|-----------------------------------|
|         | Virtual Implementation ROB | Proposed Cloud Implementation ROB | Virtual Implementation GEO | Proposed Cloud Implementation GEO |
| P1      | 16.004                     | 15.348                            | 15.495                     | 15.041                            |
| P2      | 16.281                     | 15.373                            | 16.068                     | 15.411                            |
| P3      | 17.392                     | 16.022                            | 18.140                     | 15.724                            |
| P4      | 23.634                     | 19.223                            | 25.595                     | 19.169                            |
| P5      | 45.456                     | 27.313                            | 51.808                     | 28.548                            |
| P6      | 132.179                    | 65.386                            | 166.633                    | 71.416                            |
| P7      | 1099.05                    | 265.415                           | 735.731                    | 297.122                           |

# Conclusion

- We have presented a cloud-driven application for the solution of multidomain/multiphysics problems based on the Interface Relaxation methodology.
- The users can define complex multiphysics problems, select the appropriate PDE solvers for the domains and IR method (ROB or GEO) for the interfaces and get the computed solution of the global problem.
- The resources' allocation is dynamic and is based on the computational needs of each problem.
- The proposed methodology allocates the minimum possible resources, solves the problem in a close to minimum execution time, and the resources allocation is performed automatically in the background without the user's interference.

# Παραδείγματα από εκτέλεση προβλημάτων στην IRaaS



outer rectangle

inner circle

A diagram showing a light gray circle with a black outline, centered within a larger light gray rectangle with a black outline. The text "outer rectangle" is positioned in the upper left area of the rectangle, and the text "inner circle" is centered within the circle.



# Εισαγωγή στοιχείων προβλήματος

IRaaS Problems About Logout (user)

## New Outer/Inner Rectangle Problem

Define a new problem

Problem Name \*

Problem Details \*

Outer domain XML file \*

Choose File

No file chosen

Inner domain XML file \*

Choose File

No file chosen

Iterations \*

15

Start solution process

# Προβολή Αποτελεσμάτων

IRaaS Problems About Logout (user)

## Problem Information

### Submission Datetime

2015-11-14 13:25:26

### Details

....

### Outer domain XML file

[domAfile\\_6.xml](#)

### Inner domain XML file

[domBfile\\_6.xml](#)

### Error Tolerance

1e-6

### Iterations

15

## Progress

### Status

COMPLETED

### Initialization Datetime

2015-11-14 13:27:50

15 out of 15 iterations.

100%

## Results

### Finish Datetime

2015-11-14 13:28:47

### Duration

16753 day(s) 11 hour(s) 27 minute(s) 50 second(s)

### Statistics

Inner domain vertices: 233

Inner domain interface points: 1

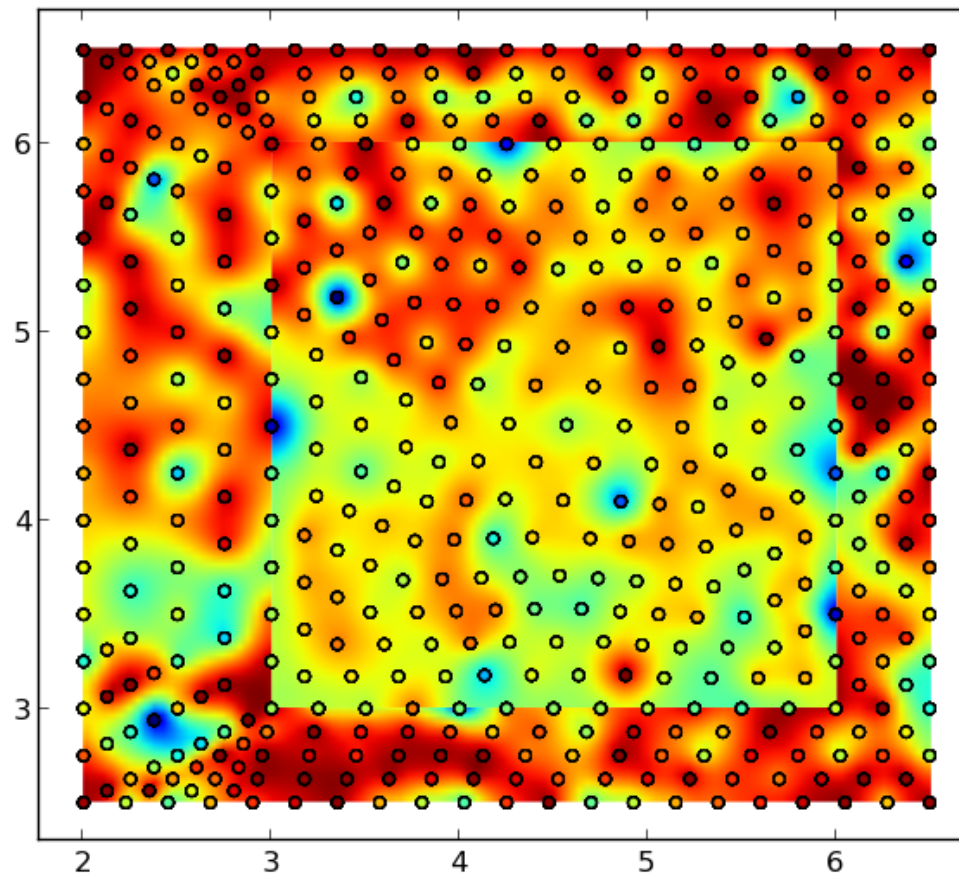
Outer domain vertices: 332

Outer domain interface points: 1

### Inner Domain

[inner\\_dom\\_iter1\\_if0.txt](#)

# Παράδειγμα αποτελέσματος από εκτέλεση



# Acknowledgements

The present research work has been partially supported by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program Education and Lifelong Learning of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALIS. Investing in knowledge society through the European Social Fund (MIS 379416)



MATENVMED

ADVANCED MATHEMATICAL METHODS AND SOFTWARE PLATFORM  
FOR SOLVING MULTIPHYSICS, MULTIDOMAIN PROBLEMS  
ON MODERN COMPUTER ARCHITECTURES:  
APPLICATIONS TO ENVIRONMENTAL ENGINEERING AND MEDICAL PROBLEMS



European Union  
European Social Fund



MINISTRY OF EDUCATION & RELIGIOUS AFFAIRS  
MANAGING AUTHORITY

Co-financed by Greece and the European Union



EUROPEAN SOCIAL FUND

# Ερωτήσεις

