

# Ετήσια Τεχνική Έκθεση

Έτος 2012



## ΘΑΛΗΣ – Πολυτεχνείο Κρήτης

Πλατφόρμα προηγμένων μαθηματικών μεθόδων και λογισμικού για την επίλυση προβλημάτων πολλαπλών πεδίων (multi physics, multidomain) σε σύγχρονες υπολογιστικές αρχιτεκτονικές: Εφαρμογή σε προβλήματα Περιβαλλοντικής Μηχανικής και Ιατρικής (MATENVMED- MIS 379416)

### Δράση 4.1

Συγκερασμός αριθμητικών μεθόδων και λογισμικού



## Περιεχόμενα

<b>1</b>	<b>Σκοπός</b>	<b>3</b>
<b>2</b>	<b>Μεθοδολογία</b>	<b>4</b>
2.1	Υλοποίηση Περιβάλλοντος Επίλυσης Προβλημάτων (ΠΕΠ) . . . .	4
2.2	Αξιοποίηση Σύγχρονων Υπολογιστικών Συστημάτων και Σύγχρο- νων Μέσων Αποθήκευσης) . . . . .	6
2.2.1	Χρήση Σύγχρονου Υλικού για την Επίλυση Μεγάλων Αραιών Γραμμικών Συστημάτων . . . . .	6
2.2.2	Αλγόριθμοι Out-of-Core & Μνήμες Flash . . . . .	7
<b>3</b>	<b>Συνεργασίες</b>	<b>8</b>
<b>4</b>	<b>Παραδοτέα</b>	<b>9</b>
<b>5</b>	<b>Μελλοντικές Δράσεις</b>	<b>9</b>

## 1 Σκοπός

Τα τελευταία χρόνια αποτελεί κοινή διαπίστωση των εμπλεκομένων με την σχεδίαση και ανάπτυξη λογισμικού για ευρείας κλίμακας επιστημονικούς υπολογισμούς η αναγκαιότητα για την ανάπτυξη:

- Ενός λειτουργικού περιβάλλοντος επίλυσης προβλημάτων (Problem Solving Environment - PSE) το οποίο, εστιαζόμενο στα ιδιαίτερα χαρακτηριστικά της κλάσης των υπό μελέτη προβλημάτων, διευκολύνει την αποδοτική χρήση (και επανάχρηση) των λογισμικών μονάδων που είτε προϋπάρχουν είτε αναπτύσσονται.
- Μιας καινοτόμου πρακτικής ανάπτυξης λογισμικού η οποία θα μας επιτρέψει να εκμεταλλευτούμε τις δυνατότητες των σύγχρονων υπολογιστικών μηχανών.

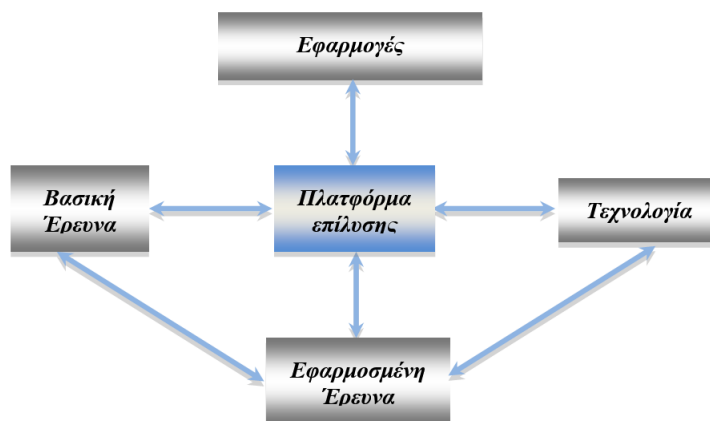
Η παρούσα δράση έχει ως στόχο να σχεδιάσει με σαφήνεια μια αρχιτεκτονική αναφοράς, η οποία φιλοδοξεί να θέσει τις βάσεις για μια ενοποιημένη προσέγγιση αντιμετώπισης των σύνθετων προβλημάτων που απασχολούν το έργο MATENVMED, καλύπτοντας τις παρακάτω αναγκαιότητες.

- Ενός λειτουργικού περιβάλλοντος επίλυσης προβλημάτων (Problem Solving Environment - PSE) το οποίο, εστιαζόμενο στα ιδιαίτερα χαρακτηριστικά της κλάσης των υπό μελέτη προβλημάτων, διευκολύνει την αποδοτική χρήση (και επανάχρηση) των λογισμικών μονάδων που είτε προϋπάρχουν είτε αναπτύσσονται.
- Μιας καινοτόμου πρακτικής ανάπτυξης λογισμικού η οποία θα μας επιτρέψει να εκμεταλλευτούμε τις δυνατότητες των σύγχρονων υπολογιστικών μηχανών.

Όπως φαίνεται στην Εικόνα 1, η πλατφόρμα επίλυσης προβλημάτων αποτελεί κεντρικό σημείο του έργου, καθώς καλείται να ενοποιήσει και να συγκεράσει τις ερευνητικές προσπάθειες που αναπτύσσονται στις υπόλοιπες δράσεις του έργου. Κατά το έτος 2012 βασικός στόχος της παρούσας δράσης ήταν η επισκόπηση της υπάρχουσας κατάστασης σε ότι αφορά τις αρχιτεκτονικές ενδιαφέροντος του έργου, αλλά και ήδη υπάρχουσες πλατφόρμες επίλυσης προβλημάτων.

Το υπόλοιπο της παρούσης Τεχνικής Έκθεσης είναι οργανωμένο ως εξής: Στην παράγραφο 2 παρουσιάζουμε τα βασικά στοιχεία της μεθοδολογίας που ακολουθήσαμε. Στην παράγραφο 4 αναφερόμαστε στα παραδοτέα που παρήχθησαν στα πλαίσια της δράσης. Στην παράγραφο 3 περιγράφουμε τις συνεργασίες που αναπτύχθηκαν, ενώ η παράγραφος 5 ολοκληρώνει την έκθεση με τη συζήτηση των μελλοντικών δραστηριοτήτων στα πλαίσια της δράσης.





Εικόνα 1: Συνολική αρχιτεκτονική MATENVMED

## 2 Μεθοδολογία

### 2.1 Υλοποίηση Περιβάλλοντος Επίλυσης Προβλημάτων (ΠΕΠ)

Κατά το έτος 2012 πραγματοποιήθηκε επισκόπηση των ήδη υπάρχουσών λύσεων στον τομέα των περιβαλλόντων επίλυσης προβλημάτων για PDEs. Αποφασίστηκε να εξαντληθεί κάθε περιθώριο αξιολόγησης και αξιοποίησης υπαρχόντων περιβαλλόντων επίλυσης προβλημάτων (ΠΕΠ). Με τον τρόπο αυτό, θα είναι εφικτό να επικεντρωθεί η προσπάθεια της ομάδας στο ερευνητικό αντικείμενο του έργου, δηλαδή την υποστήριξη εφαρμογών MDMP (Multi-Domain, Multi-Physics), έναντι της εξαρχής υλοποίησης ενός νέου ΠΕΠ. Βασικά κριτήρια κατά τη φάση αξιολόγησης των υπαρχόντων ΠΕΠ ήταν τα ακόλουθα:

- Υποστήριξη επίλυσης μερικών διαφορικών εξισώσεων κατάλληλων για τις απαιτήσεις των κεντρικών προβλημάτων του έργου.
- Ανοιχτού κώδικα, ώστε να είναι δυνατή η ανάπτυξη της απαιτούμενης υποστήριξης για τις εφαρμογές και τις πλατφόρμες ενδιαφέροντος του έργου. Με άδεια χρήσης που δε θα απαγορεύει τις απαιτούμενες από το έργο δραστηριότητες.
- Ευρεία και ενεργή κοινότητα ανάπτυξης και υποστήριξης.
- Δυνατότητα χρήσης τεχνολογίας (βιβλιοθηκών) αιχμής στο επίπεδο της γραμμικής άλγεβρας.
- Απλή διεπαφή χρήστη που θα επιτρέπει γρήγορη ανάπτυξη και εύκολο πειραματισμό.

- Δυνατότητα χρήσης ως ανεξάρτητο τμήμα λογισμικού στα πλαίσια μεγαλύτερων εφαρμογών.

Διαπιστώθηκε η διαθεσιμότητα μεγάλου αριθμού ΠΕΠ, με αρκετά κοινά μεταξύ τους χαρακτηριστικά. Πρόκειται τυπικά για αναπτυξιακές (ενίοτε και ερευνητικές) προσπάθειες μεγάλης κλίμακας. Κατά κανόνα είναι γραμμένα σε C/C++ και τα μη εμπορικά ΠΕΠ είναι ανοιχτού κώδικα. Παρά τον μεγάλο τους αριθμό, δε διαπιστώθηκαν μεγάλες διαφορές στις βασικές τους ιδέες. Πολλά μπορούν να αξιοποιήσουν τον παραλληλισμό στο επίπεδο της γραμμικής άλγεβρας, αλλά και με χρήση MPI ή PosixThreads. Κάποια πιο σύγχρονα περιβάλλοντα είναι επίσης ικανά να αξιοποιήσουν επιταχυντές με χρήση OpenCL.

Τα σημαντικότερα εμπορικά πακέτα που εξετάστηκαν είναι τα ακόλουθα:

- Matlab PDE Toolbox<sup>1</sup>
- COMSOL<sup>2</sup>
- Abaqus<sup>3</sup>
- FlexPDE<sup>4</sup>

Δυστυχώς το γεγονός ότι τα εμπορικά πακέτα είναι κλειστού κώδικα και συνήθως δεν παρέχουν επαρκώς ευέλικτο API για τον έλεγχο της συμπεριφοράς τους και τη χρήση τους από εξωτερικές εφαρμογές μας ανάγκασε να τα απορρίψουμε εξ αρχής.

Επίσης εξετάζεται μια σειρά από μη εμπορικά ΠΕΠ:

- Implicit Parallel Accurate Reservoir Simulator (IPARS)<sup>5</sup>
- Chombo<sup>6</sup>
- deal.II<sup>7</sup>[4]
- FEniCS<sup>8</sup>[10]
- FiPy<sup>9</sup>[6]

<sup>1</sup><http://www.mathworks.com/products/pde>

<sup>2</sup><https://www.comsol.com>

<sup>3</sup><http://www.3ds.com/products-services/simulia/products/abaqus>

<sup>4</sup><http://www.pdesolutions.com>

<sup>5</sup><http://csm.ices.utexas.edu/lparsWeb/iparsintro.html>

<sup>6</sup><https://commons.lbl.gov/display/chombo>

<sup>7</sup><http://www.dealii.org>

<sup>8</sup><http://fenicsproject.org>

<sup>9</sup><http://www.ctcms.nist.gov/fipy>



- HiFlow<sup>310</sup>[2]
- freeFEM++<sup>11</sup>[7]
- OpenModelica<sup>12</sup>[5]
- Overture<sup>13</sup>[8]
- pdelib2<sup>14</sup>
- PETSc<sup>15</sup>[3]

## 2.2 Αξιοποίηση Σύγχρονων Υπολογιστικών Συστημάτων και Σύγχρονων Μέσων Αποθήκευσης)

### 2.2.1 Χρήση Σύγχρονου Υλικού για την Επίλυση Μεγάλων Αραιών Γραμμικών Συστημάτων

Η αποδοτική επίλυση συστημάτων γραμμικών εξισώσεων με αραιούς πίνακες είναι θεμελιώδους σημασίας για τις υπολογιστικές επιστήμες, αφού τέτοιου είδους συστήματα απαντώνται κατά την επίλυση προβλημάτων του πραγματικού κόσμου (real world problems). Πιο συγκεκριμένα, στα προβλήματα πολλαπλών πεδίων (multidomain/multiphysics problems) αραιοί πίνακες προκύπτουν κατά την διακριτοποίηση μερικών διαφορικών εξισώσεων (ανεξάρτητα της μεθόδου διακριτοποίησης). Έτσι, τα λογισμικά για την προσομοίωση προβλημάτων πολλαπλών μοντέλων φυσικής ενσωματώνουν και μεθόδους για την επίλυση αραιών συστημάτων.

Η διαθέσιμη, κάθε φορά, τεχνολογία υλικού επηρεάζει τις μεθόδους υπολογισμού, πολύ δε περισσότερο, όταν οι υπολογισμοί απαιτούν πολλούς πόρους και ενέχουν χρονικούς περιορισμούς. Όσο αφορά τους υπολογισμούς με αραιούς πίνακες φαίνεται ότι μπορεί να ωφεληθούν σημαντικά από τις επιδόσεις που προσφέρουν οι σύγχρονες GPUs, οι συνεπεξεργαστές (coprocessors) ακόμη και οι multi-core επεξεργαστές. Η επόμενη GPU της NVidia (αρχιτεκτονικής Kepler) αναμένεται ότι θα μπορεί να επιτύχει μέχρι και 1.66Tflops (K40, single GPU), ενώ ο συνεπεξεργαστής Xeon Phi της Intel 1.2Tflops. Οι πιο πρόσφατοι εμπορικοί επεξεργαστές γενικού σκοπού ενσωματώνουν μέχρι και 8 cores και επιτυγχάνουν μέχρι και 1/2 Tflor.

<sup>10</sup><http://www.hiflow3.org>

<sup>11</sup><http://www.freefem.org/ff++>

<sup>12</sup><https://www.openmodelica.org>

<sup>13</sup><http://www.overtureframework.org>

<sup>14</sup><http://www.wias-berlin.de/software/pdelib/about.jsp?lang=1>

<sup>15</sup><http://www.mcs.anl.gov/petsc>



Επερχόμενες τεχνολογίες υλικού όπως το disaggregated server rack υπόσχονται να προσφέρουν νέα περιβάλλοντα για την εκτέλεση υπολογισμών. Η αρχιτεκτονική αυτή προτείνει την αντικατάσταση του κλασσικού server rack, ως συνόλου από διακριτούς εξυπηρετητές, σε σύνολα από διαμοιραζόμενους επεξεργαστές, μνήμη και μονάδες αποθήκευσης συνδεδεμένα μεταξύ τους με οπτικό δίκτυο.

## 2.2.2 Αλγόριθμοι Out-of-Core & Μνήμες Flash

Για την αντιμετώπιση των αυξημένων απαιτήσεων σε μνήμη στις άμεσες μεθόδους, έχουν προταθεί αλγόριθμοι που χρησιμοποιούν δευτερεύουσα μνήμη (αλγόριθμοι out-of-core). Οι περισσότερες βιβλιοθήκες λογισμικού για την επίλυση αραιών συστημάτων με άμεσες μεθόδους (π.χ. INTEL MKL PARDISO, MUMPS, PASTIX, TAUCS) περιλαμβάνουν και τέτοιους αλγορίθμους.

Τα τελευταία χρόνια οι μνήμες flash χρησιμοποιούνται ως αποθηκευτικό μέσο σε ενσωματωμένα συστήματα, κινητά τηλέφωνα, φορητούς υπολογιστές και εξυπηρετητές, έχοντας σαφή πλεονεκτήματα έναντι των παραδοσιακών μαγνητικών δίσκων. Οι εξαιρετικές επιδόσεις τους μας δίνουν κίνητρο να ερευνήσουμε την δυνατότητα αξιοποίησής τους από αλγορίθμους που χρησιμοποιούν εξωτερική μνήμη για την επίλυση αραιών γραμμικών συστημάτων.

Οι μνήμες flash ανήκουν στην κατηγορία των ηλεκτρικά εγγράψιμων και επαναπρογραμματιζόμενων μνημών (EEPROM – electrically erasable programmable read only memory). Οι μνήμες αυτές είναι μη πτητικές (non-volatile) δηλαδή διατηρούν τα περιεχόμενά τους και μετά την διακοπή της τροφοδοσίας τους με ηλεκτρικό ρεύμα. Οι μνήμες flash χρησιμοποιούνται για την μόνιμη αποθήκευση δεδομένων σε ενσωματωμένα συστήματα, κινητές συσκευές και τα τελευταία χρόνια, ως αποθηκευτικά μέσα σε προσωπικούς υπολογιστές και εξυπηρετητές. Η ευρεία διάδοσή τους οφείλεται στις εξαιρετικές επιδόσεις τους σε σύγκριση με τους παραδοσιακούς μαγνητικούς δίσκους, την αντοχή τους σε δονήσεις (shock resistant) και την χαμηλή κατανάλωση ρεύματος.

Υπάρχουν δύο τύποι μνημών flash οι NAND και οι NOR που λαμβάνουν την ονομασία τους από τους αντίστοιχους τύπους λογικών πυλών που χρησιμοποιούν. Οι μνήμες NOR διευθυνσιοδοτούνται σε επίπεδο byte και χρησιμοποιούνται κυρίως για την αποθήκευση κώδικα, ενώ οι NAND διευθυνσιοδοτούνται σε επίπεδο σελίδας (page) και χρησιμοποιούνται για την αποθήκευση δεδομένων. Κάθε κυψέλη (cell) της μνήμης flash μπορεί να αποθηκεύσει ένα (SLC) ή περισσότερα (MLC) bits ανάλογα με τις διαφορετικές καταστάσεις που μπορεί να υποστηρίξει ένα τρανζίστορ. Τα cells οργανώνονται σε σελίδες και οι σελίδες σε μπλοκ. Το τυπικό μέγεθος μια σελίδας κυμαίνεται από 512 Bytes ως 16KB, ενώ ενός μπλοκ από 32KB ως 512KB. Στη συνέχεια του εγγράφου με τον όρο μνήμη flash θα αναφερόμαστε στην κατηγορία NAND.



Οι μνήμες flash έχουν κάποια ιδιαίτερα χαρακτηριστικά: η ανάγνωση και η εγγραφή πραγματοποιείται σε επίπεδο σελίδας ενώ η διαγραφή σε επίπεδο μπλοκ. Η ανάγνωση είναι ταχύτερη από την εγγραφή και η εγγραφή πιο γρήγορη από την διαγραφή. Επιπλέον, μια σελίδα θα πρέπει να διαγραφεί πριν την επανεγγραφή της (δεν υποστηρίζονται in-place updates). Τέλος, η ζωή των μνημών flash περιορίζεται από έναν πεπερασμένο αριθμό κύκλων εγγραφής/διαγραφής.

Οι δίσκοι στερεάς κατάστασης (SSDs), αποθηκευτικές συσκευές που χρησιμοποιούν την μνήμη flash, είναι συμβατοί με τους μαγνητικούς σκληρούς δίσκους. Τα σημαντικότερα μέρη μιας τέτοιας συσκευής είναι τα ολοκληρωμένα κυκλώματα της μνήμης flash και ένας μικροελεγκτής που προσομοιώνει μια διαπαφή τύπου μπλοκ, όμοια με αυτή των σκληρών δίσκων, χρησιμοποιώντας το FTL (Flash Translation Layer). Το FTL αντιστοιχεί τις λογικές διευθύνσεις που χρησιμοποιούνται από τα ανώτερα στρώματα σε φυσικές στα ολοκληρωμένα της flash και υλοποιεί μηχανισμούς για την ανανέωση σελίδων εκτός θέσης (out-of-place updates), για την εξισορρόπηση φθοράς (wear leveling) και ανάκτησης σελίδων (garbage collection). Πιο συγκεκριμένα, ο μηχανισμός ανανέωσης σελίδων εκτός θέσης ικανοποιεί τα αιτήματα ανανέωσης σελίδων πραγματοποιώντας εγγραφές σε «καθαρές» σελίδες (δηλ. σελίδες που έχουν ήδη διαγραφεί) και ανανεώνει κατάλληλα τον πίνακα αντιστοίχισης λογικών και φυσικών διευθύνσεων. Ο μηχανισμός εξισορρόπησης φθοράς κατανέμει ομοιόμορφα τις εγγραφές σε όλο το μέσο, και ο μηχανισμός ανάκτησης σελίδων ανακτά σελίδες που έχουν επισημανθεί ως διαγραμμένες.

Από τα παραπάνω φαίνεται ότι τα αποθηκευτικά μέσα που χρησιμοποιούν μνήμες flash έχουν εξαιρετικές δυνατότητες αλλά και ιδιαίτερα χαρακτηριστικά. Έτσι, οι αλγόριθμοι και οι δομές δεδομένων που έχουν σχεδιαστεί για τους μαγνητικούς δίσκους φαίνεται να χρειάζονται αναθεώρηση. Μάλιστα, ένας σημαντικός αριθμός από επιστημονικές εργασίες έχει δημοσιευθεί στο πεδίο των βάσεων δεδομένων, κυρίως σε δομές ευρετηρίων (indexes) [15, 11, 1, 14, 13] και buffer managers [12, 9], που προσπαθούν να εκμεταλλευτούν τα πλεονεκτήματα της flash (random reads) και να αποσοβήσουν τις ιδιαιτερότητές της (writes πιο αργά από reads).

### 3 Συνεργασίες

Στον Πίνακα 1 συνοψίζονται οι βασικές δραστηριότητες που αναπτύχθηκαν στα πλαίσια της δράσης 4.1 κατά το 2012 και αναφέρονται οι ομάδες που συμμετείχαν.





	ΚΕΟ 1	ΚΕΟ 2	ΚΕΟ 3
Προδιαγραφές ΠΕΠ	x	x	x
Απαιτήσεις για υποστήριξη προβλημάτων MDMP	x	x	x
Επισκόπηση GPUs/FPGAs/Flash Storage		x	

Πίνακας 1: Συνεργασίες στα πλαίσια της Δράσης 4.1.

## 4 Παραδοτέα

Τα παραδοτέα της παρούσας δράσης, σύμφωνα και με το τεχνικό δελτίο του έργου, είναι τα ακόλουθα:

**Ετήσια τεχνική έκθεση περιγραφής αποτελεσμάτων:** το παρόν κείμενο.

## 5 Μελλοντικές Δράσεις

Το αμέσως επόμενο βήμα στα πλαίσια της δράσης 4.1 είναι η επιλογή κατάλληλου ΠΕΠ το οποίο θα επεκταθεί για τις ανάγκες των προβλημάτων στα οποία στοχεύει το έργο. Επίσης, το ΠΕΠ θα πρέπει να επεκταθεί κατάλληλα ώστε να μπορεί να αξιοποιήσει σύγχρονες τεχνολογίες υλικού, καθώς και απομακρυσμένη εκτέλεση (Grid / Cloud).

## Αναφορές

- [1] D. Agrawal, D. Ganesan, R. Sitaraman, Y. Diao, and S. Singh. Lazy-adaptive tree: An optimized index structure for flash devices. In *Proceedings of VLDB Endowment*, pages 361–372, 2009.
- [2] H. Anzt, W. Augustin, M. Baumann, T. Gengenbach, T. Hahn, A. Helfrich-Schkarbanenko, V. Heuveline, E. Ketelaer, D. Lukarski, A. Nestler, S. Ritterbusch, S. Ronnas, M. Schick, M. Schmidtreick, C. Subramanian, J.-P. Weiss, F. Wilhelm, and M. Wlotzka. Hiflow3: A hardware-aware parallel finite element package. In Holger Brunst, Matthias S. Müller, Wolfgang E. Nagel, and Michael M. Resch, editors, *Tools for High Performance Computing 2011*, pages 139–151. Springer Berlin Heidelberg, 2012.
- [3] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern*



- Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.
- [4] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II – a general purpose object oriented finite element library. *ACM Trans. Math. Softw.*, 33(4):24/1–24/27, 2007.
- [5] Peter Fritzson, Peter Aronsson, Håkan Lundvall, Kaj Nyström, Adrian Pop, Levon Saldamli, and David Broman. The OpenModelica Modeling, Simulation, and Software Development Environment. *Simulation News Europe*, 44(45), December 2005.
- [6] Jonathan E. Guyer, Daniel Wheeler, and James A. Warren. FiPy: Partial differential equations with Python. *Computing in Science & Engineering*, 11(3):6–15, 2009.
- [7] F. Hecht. New development in freefem++. *J. Numer. Math.*, 20(3-4):251–265, 2012.
- [8] William D. Henshaw. Overture: An object-oriented system for solving pdes in moving geometries on overlapping grids. In L. Sakell and D. D. Knight, editors, *Proceedings of the First AFOSR Conference on Dynamic Motion CFD*, pages 281–290, June 1996.
- [9] P. Jin, Y. Ou, T. Harder, and Z. Li. Ad-lru: An efficient buffer replacement algorithm for flash-based databases. *Data and Knowledge Engineering*, 72:83–102, 2012.
- [10] Anders Logg, Kent-Andre Mardal, Garth N. Wells, et al. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012.
- [11] Y. Lu, J. Li, B. Cui, and X. Chen. Log-compact r-tree: an efficient spatial index for ssd. In *Proceedings of Database Systems for Advanced Applications*, pages 202–213, 2011.
- [12] S.Y. Park, D. Jung, J.U. Kang, J.S. Kim, and J. Lee. Cflru: a replacement algorithm for flash memory. In *Proceedings of the 2006 international conference on Compilers, architecture and synthesis for embedded systems*, pages 234–241. ACM, 2006.
- [13] H. Roh, S. Park, S. Kim, M. Shin, and S.W. Lee. B+-tree index optimization by exploiting internal parallelism of flash-based solid state drives. In *Proceedings of VLDB Endowment*, pages 286–297, 2011.



- [14] N. Wang, P. Jin, S. Wan, Y. Zhang, and L. Yue. Or-tree: An optimized spatial tree index for flash-memory storage systems. *Data and Knowledge Engineering*, pages 1–14, 2012.
- [15] C.H. Wu, T.W. Kuo, and L.P. Chang. An efficient b-tree layer implementation for flash- memory storage systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 19, 2007.

