



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΕΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

<http://www.cs-lab.ece.ntua.gr>

Διπλωματικές Εργασίες

Ακαδημαϊκό έτος 2019-20

Μέρος I

Αρχιτεκτονική - Παράλληλα Συστήματα

1 Επέκταση εξομοιωτή χρονοδρομολόγησης εργασιών για υπολογιστικά συστήματα υψηλής επίδοσης σε Python

Τα υπολογιστικά συστήματα υψηλής επίδοσης (High Performance Computing clusters[1]) είναι ευρέως διαδεδομένα και συχνά χρησιμοποιούνται για την επίλυση πολύπλοκων προβλημάτων σε ποικίλες ερευνητικές περιοχές όπως η πρόγνωση και η μοντελοποίηση των καιρικών φαινομένων καθώς και η διερεύνηση της ακολουθίας του ανθρώπινου γονιδιώματος. Οι ερευνητές υποβάλλουν στο σύστημα τις εργασίες τους και αυτό αναλαμβάνει να τις εκτελέσει. Η κατάλληλη χρονοδρομολόγηση των εργασιών αποτελεί καθοριστικό παράγοντα για την αποτελεσματική χρήση των υπολογιστικών πόρων αλλά και την εξοικονόμηση στην κατανάλωση χρόνου και ενέργειας. Το βασικό λογισμικό που συνθέτει μια τέτοια υπολογιστική υποδομή, ονομάζεται διαχειριστής πόρων (resource manager) και περιλαμβάνει έναν χρονοδρομολογητή εργασιών (job scheduler). Ο resource manager αναλαμβάνει να διαμοιράσει τους υπολογιστικούς πόρους στις αντίστοιχες εργασίες. Ο job scheduler επικοινωνεί με τον resource manager προκειμένου να πληροφορηθεί για τις ουρές (queues), τα φορτία των υπολογιστικών κόμβων (nodes) και την διαθεσιμότητα των πόρων, ώστε να πάρει αποφάσεις για τη χρονοδρομολόγηση εργασιών. Πληθώρα λογισμικών ανεπτυγμένα κατά MPI[2] (πασίγνωστο πρότυπο παράλληλου προγραμματισμού μέσω ανταλλαγής μηνυμάτων μεταξύ των διεργασιών) εκτελούνται καθημερινά σε υπολογιστικά συστήματα υψηλής επίδοσης. Πρόσφατα συγγράφηκε μια πρώτη έκδοση ενός ανοικτού κώδικα λογισμικού[3] σε Python που αξιοποιεί το εργαλείο mpi-run[4] (υπεύθυνο για την εκτέλεση εφαρμογών MPI) και αναπτύσσει μια εξομοίωση χρονοδρομολογητή εργασιών πάνω από τον Torque[5] resource manager. Σκοπός της διπλωματικής αυτής είναι η βελτιστοποίηση του Python κώδικα με (i) την υλοποίηση διάφορων υπάρχοντων και νέων αλγορίθμων χρονοδρομολόγησης για HPC

συστήματα, (ii) την υποστήριξη περισσότερων εφαρμογών/benchmarks, (iii) τη γενικότερη αναβάθμιση του κώδικα και των επιλογών για το χρήστη.

Σχετικά Μαθήματα: Συστήματα Παράλληλης Επεξεργασίας

Σχετική Βιβλιογραφία:

1. <https://en.wikipedia.org/wiki/Supercomputer>
2. https://en.wikipedia.org/wiki/Message_Passing_Interface
3. <https://github.com/nikosT/MPI-Job-Scheduler>
4. <https://www.open-mpi.org/doc/v4.0/man1/mpirun.1.php>
5. <https://en.wikipedia.org/wiki/TORQUE>

Επικοινωνία: Νικόλαος Τριανταφύλλης, ntriantafyl@cslab.ece.ntua.gr

Γεώργιος Γκούμας, goumas@cslab.ece.ntua.gr, 210-772-2402

2 Αξιολόγηση και σύγκριση αλγορίθμων χρονοδρομολόγησης με ή χωρίς τεχνικές co-scheduling

Ο διαχειριστής πόρων (resource manager[1]) αποτελεί το βασικό λογισμικό που συνθέτει ένα υπολογιστικό σύστημα υψηλής επίδοσης (High Performance Computing clusters[2]). Οι ερευνητές υποβάλλουν τις εργασίες τους και αυτός αναλαμβάνει να τις εκτελέσει, διαμοιράζοντας τους υπολογιστικούς πόρους, κατάλληλα, στις αντίστοιχες εργασίες. Ο job scheduler (υπομήμα του resource manager) επικοινωνεί με τον resource manager προκειμένου να πληροφορηθεί για τις ουρές (queues), τα φορτία των υπολογιστικών κόμβων (nodes) και την διαθεσιμότητα των πόρων, ώστε να πάρει αποφάσεις για τη χρονοδρομολόγηση εργασιών. Υπάρχουν διάφοροι αλγόριθμοι στην κατηγορία των space-sharing αλγορίθμων χρονοδρομολόγησης, όπως ο First Come First Served (FCFS), ο Shortest Job First (SJF), ο Longest Job First (LJF), ο Backfilling κ.α. Οι αλγόριθμοι αυτοί -στα υπολογιστικά σύστημα υψηλής επίδοσης- δεσμεύουν, συνήθως, πόρους στο επίπεδο του κόμβου. Εξ' ορισμού η επιλογή ανάθεσης πόρων στον επίπεδο κόμβου (δεδομένου ότι οι κόμβοι περιλαμβάνουν ολοένα περισσότερα και μεγαλύτερα εξαρτήματα υλικού πια) είναι αντιπαραγωγική όσον αφορά το throughput συστήματος, την κατανάλωση ενέργειας και κόστους. Μελέτες δείχνουν πως το co-scheduling[3], δηλαδή η ανάθεση πόρων στο επίπεδο του πυρήνα (κι άρα η εκτέλεση διαφορετικών εφαρμογών ταυτόχρονα στον ίδιο κόμβο), οδηγεί σε αποτελεσματικότερη χρήση των υπολογιστικών πόρων. Από την άλλη πλευρά, η επιλογή των εφαρμογών που θα εκτελεστούν ταυτόχρονα λαμβάνει σημαντικό ρόλο για την επίδοση που θα πετύχουν λόγω των race conditions που θα αναπτυχθούν ανάλογα με τους πόρους που ζητά η εκάστοτε εφαρμογή. Στη διπλωματική αυτή περιλαμβάνεται η μελέτη και αξιολόγηση αλγορίθμων χρονοδρομολόγησης με χρήση υπάρχοντων benchmarks. Συγκεκριμένα, θα μελετηθεί (i) η κλιμακωσιμότητα των benchmarks σ' ένα cluster, (ii) η επίδοση για διαφορετικού είδους αναθέσεις πόρων, (iii) τα race conditions που αναπτύσσονται σε διάφορα είδη εφαρμογών (π.χ. memory bounded, compute bounded), (iv) η αξιολόγηση και σύγκριση αλγορίθμων χρονοδρομολόγησης για τα συγκεκριμένα benchmarks με ή χωρίς τεχνικές co-scheduling.

Σχετικά Μαθήματα: Συστήματα Παράλληλης Επεξεργασίας

Σχετική Βιβλιογραφία:

1. <https://en.wikipedia.org/wiki/TORQUE>
2. <https://en.wikipedia.org/wiki/Supercomputer>
3. http://www.cslab.ntua.gr/~ntriantafyl/stuff/HPC_Job_Scheduling.pdf

Επικοινωνία: Νικόλαος Τριανταφύλλης, ntriantafyl@cslab.ece.ntua.gr
Γεώργιος Γκούμας, goumas@cslab.ece.ntua.gr, 210-772-2402

3 Διαχείριση Πόρων σε Συστήματα Μεγάλης Κλίμακας

Καθώς η εκτέλεση υπηρεσιών όπως web service, email, streaming, επιστημονικές, υπολογιστικές και office εφαρμογές μεταφέρεται σε συστήματα μεγάλης κλίμακας, τα ζητήματα της απομονωμένης πρόσβασης σε πόρους και της παροχής ικανοποιητικής ποιότητας υπηρεσίας (Quality of Service) γίνονται ολοένα πιο πολύπλοκα. Για τη διατήρηση του Service Level Agreement (SLA) σε εφαρμογές υψηλής προτεραιότητας με αυστηρές απαιτήσεις οι διαχειριστές των συστημάτων καταφεύγουν σε λύσεις που μειώνουν την πιθανότητα μειωμένης επίδοσης των εφαρμογών λόγω του ανταγωνισμού για κοινούς πόρους, αλλά ταυτόχρονα μειώνουν τη χρησιμοποίηση των πόρων του συστήματος και οδηγούν σε αυξημένο λειτουργικό κόστος. Για την αντιμετώπιση των ζητημάτων αυτών προτείνονται τεχνικές απομόνωσης πόρων (Last Level Cache - Intel CMT CAT, Memory Bandwidth, Core isolation), τεχνικές χαρακτηρισμού των εφαρμογών ως προς τους κρίσιμους για την εκτέλεση τους πόρους με σκοπό τη συνεκτέλεση εφαρμογών με συμπληρωματικές απαιτήσεις για πόρους, και τεχνικές on-the-fly εντοπισμού μειωμένης επίδοσης. Στόχος της παρούσας εργασίας είναι η σύγκριση διαφόρων τεχνικών εικονικοποίησης και η εφαρμογή τεχνικών διαχείρισης και απομόνωσης πόρων για την αποδοτική συνεκτέλεση εφαρμογών σε ένα σύστημα μεγάλης κλίμακας. Συγκεκριμένα, η εργασία περιλαμβάνει (α) σύγκριση τεχνικών εικονικοποίησης (VM, Docker containers, Kubernetes) ως προς τις ομοιότητες, διαφορές και δυνατότητες που παρέχουν για την εκτέλεση εφαρμογών σε συστήματα μεγάλης κλίμακας, (β) μελέτη τεχνικών χαρακτηρισμού εφαρμογών ως προς τη χρησιμοποίηση πόρων με χρήση τεχνικών Machine Learning, (γ) ανάπτυξη εργαλείων παρακολούθησης της επίδοσης εφαρμογών με συνδυασμό τεχνικών Machine Learning και χρήση hardware performance counters, καθώς και αναζήτηση εναλλακτικών μεθόδων όταν η πρόσβαση στους performance counters είναι περιορισμένη ή αδύνατη και (δ) υλοποίηση τεχνικών απομόνωσης και διαχείρισης πόρων σε συστήματα διαχείρισης εικονικοποίησης (Openstack, Kubernetes).

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών, Αρχιτεκτονική Υπολογιστών, Λειτουργικά Συστήματα

Σχετική Βιβλιογραφία:

1. Novaković, Dejan, et al. "Deepdive: Transparently identifying and managing performance interference in virtualized environments." Presented as part of the 2013 USENIX Annual Technical Conference (USENIXATC 13). 2013.
2. Gan, Yu, et al. "Seer: Leveraging Big Data to Navigate the Complexity of Performance Debugging in Cloud Microservices." Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. ACM, 2019.
3. Delimitrou, Christina, and Christos Kozyrakis. "Quasar: resource-efficient and QoS-aware cluster management." ACM SIGARCH Computer Architecture News. Vol. 42. No. 1. ACM, 2014.

4. Zhang, Xiao, et al. "CPI 2: CPU performance isolation for shared compute clusters." Proceedings of the 8th ACM European Conference on Computer Systems. ACM, 2013.
5. Intel CMT - CAT

Επικοινωνία: Γιάννης Παπαδάκης, ypap@cslab.ece.ntua.gr
Κωστής Νίκας, knikas@cslab.ece.ntua.gr, 210-772-4159

4 Σχεδιασμός και υλοποίηση δυναμικών μηχανισμών διαμοιρασμού μοιραζόμενων πόρων σε σύγχρονα πολυπύρηννα υπολογιστικά συστήματα με τη χρήση ενισχυτικής μάθησης

Στους σύγχρονους επεξεργαστές, οι συνεκτελούμενες διεργασίες ανταγωνίζονται για τους μοιραζόμενους πόρους του συστήματος, όπως η κρυφή μνήμη του τελευταίου επιπέδου (last-level cache) και το διαθέσιμο εύρος ζώνης πρόσβασης στην μνήμη (memory bandwidth), με αποτέλεσμα συχνά μία διεργασία να επηρεάζει την απόδοση των άλλων. Οι πλέον σύγχρονοι επεξεργαστές παρέχουν υποστήριξη σε επίπεδο υλικού (hardware) και λογισμικού (runtime) για την επίβλεψη της χρησιμοποίησης (monitoring) των πόρων αυτών. Επιπλέον, παρέχουν υποστήριξη για τον διαμοιρασμό των πόρων αυτών (partitioning). Η παρούσα διπλωματική θα μελετήσει την υλοποίηση μηχανισμών και πολιτικών διαμοιρασμού των μοιραζόμενων πόρων ενός πολυπύρηνου συστήματος με τη χρήση τεχνικών ενισχυτικής μάθησης (reinforcement learning), εξετάζοντας την επίτευξη διαφορετικών στόχων ανάλογα με τη χρήση του συστήματος (όπως την αύξηση της συνολικής επίδοσης του συστήματος ή τη διατήρηση υψηλής επίδοσης για μία ή περισσότερες εφαρμογές). Η ενισχυτική μάθηση επιτρέπει την αυτοματοποιημένη υλοποίηση τέτοιων στόχων, με τη χρήση agents λογισμικού που διερευνούν έναν χώρο παραμέτρων (exploration) και αξιοποιούν υπάρχουσα γνώση (exploitation) και λαμβάνουν τις κατάλληλες αποφάσεις για την επίτευξη του τελικού στόχου. Στη συγκεκριμένη εργασία, οι αντίστοιχες παράμετροι αφορούν τους μοιραζόμενους πόρους του συστήματος και η υπάρχουσα γνώση αφορά την επίδοση των εφαρμογών υπό συγκεκριμένες συνθήκες εκτέλεσης.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών, Αρχιτεκτονική Υπολογιστών, Λειτουργικά Συστήματα, Εργαστήριο Λειτουργικών Συστημάτων, Τεχνικές Μηχανικής Μάθησης

Σχετική Βιβλιογραφία:

1. Lo, David, et al. "Heracles: Improving resource efficiency at scale." ACM SIGARCH Computer Architecture News. Vol. 43. No. 3. ACM, 2015.
2. Nikas, Konstantinos, et al. "DICER: Diligent Cache Partitioning for Efficient Workload Consolidation." Proceedings of the 48th International Conference on Parallel Processing. ACM, 2019.
3. Closed-Loop Network Automation for Optimal Resource Allocation via Reinforcement Learning

Επικοινωνία: Κωστής Νίκας, knikas@cslab.ece.ntua.gr, 210-772-4159
Νικέλα Παπαδοπούλου, nikela@cslab.ece.ntua.gr, 210-772-2495
Βασίλης Καρακώστας, vkarakos@cslab.ece.ntua.gr, 210-772-4133

5 Τεχνικές πρόβλεψης και προβολής της επίδοσης παράλληλων εφαρμογών σε υπολογιστικά συστήματα μεγάλης κλίμακας

Η μοντελοποίηση και πρόβλεψη της επίδοσης των παράλληλων εφαρμογών που εκτελούνται σε υπερυπολογιστές είναι κρίσιμη για το σχεδιασμό των συστημάτων επόμενης γενιάς. Ένα από τα σημαντικότερα ερωτήματα που καλούνται να απαντήσουν τα διάφορα μοντέλα πρόβλεψης είναι η επίδοση των εφαρμογών σε συστήματα μεγαλύτερης κλίμακας από τα υπάρχοντα, δηλαδή η προβολή της επίδοσης των εφαρμογών (performance extrapolation). Το συγκεκριμένο ερώτημα γίνεται επιτακτικό καθώς βρισκόμαστε στη φάση της μετάβασης από την εποχή των επιδόσεων της τάξης των PetaFLOPS στην εποχή των επιδόσεων της τάξης των ExaFLOPS, που συνοδεύεται από σημαντική αύξηση του πλήθους των πυρήνων και των κόμβων των υπερυπολογιστών. Στη βιβλιογραφία έχουν προταθεί πολλές τεχνικές μοντελοποίησης και πρόβλεψης της επίδοσης, που στηρίζονται είτε σε αναλυτικά μοντέλα είτε σε εμπειρική μοντελοποίηση. Επιπλέον, από τις τεχνικές αυτές, ορισμένες έχουν επίγνωση της αρχιτεκτονικής ή/και της εφαρμογής, ενώ άλλες αντιμετωπίζουν με γενικό τρόπο την αρχιτεκτονική ή τις εφαρμογές. Για την εφαρμογή τους, απαιτούν τη συλλογή κάποιας πληροφορίας, στο χρόνο μεταγλώττισης ή στο χρόνο εκτέλεσης, και χρησιμοποιούν πολλαπλά εργαλεία για την εξαγωγή και τη χρήση τέτοιας πληροφορίας, όπως προσομοιωτές (simulators) και εργαλεία συλλογής ίχνους (tracing tools). Η παρούσα διπλωματική έχει δύο μέρη. Στο πρώτο μέρος, θα μελετηθεί η σχετική βιβλιογραφία και τα υπάρχοντα εργαλεία προσομοίωσης, για την επιλογή της κατάλληλης μεθοδολογίας για την προβολή της επίδοσης σε συστήματα μεγαλύτερης κλίμακας. Στο δεύτερο μέρος, θα αναπτυχθούν μοντέλα πρόβλεψης επίδοσης παράλληλων εφαρμογών, θα ενσωματωθούν σε εργαλεία προσομοίωσης και θα εξεταστούν πειραματικά ως προς την αποτελεσματικότητά τους να προβάλλουν την επίδοση εφαρμογών σε συστήματα μεγαλύτερης κλίμακας.

Σχετικά Μαθήματα: Συστήματα Παράλληλης Επεξεργασίας

Επικοινωνία: Νικέλα Παπαδοπούλου, nikela@cslab.ece.ntua.gr, 210-772-2495

Γεώργιος Γκούμας, goumas@cslab.ece.ntua.gr, 210-772-2402

6 Υλοποίηση του πολλαπλασιασμού αραιού πίνακα με διάνυμα σε συστήματα κατανεμημένης μνήμης με χρήση του προγραμματιστικού μοντέλου MPI

Ο υπολογιστικός πυρήνας πολλαπλασιασμού αραιού πίνακα με διάνυμα (SpMV) συναντάται σε πληθώρα επιστημονικών εφαρμογών που εκτελούνται σε υπολογιστικά συστήματα μεγάλης κλίμακας και συχνά αναλώνει σημαντικό μέρος του συνολικού χρόνου εκτέλεσης. Στόχος της παρούσας διπλωματικής είναι η υλοποίηση του πυρήνα SpMV σε συστήματα κατανεμημένης μνήμης με χρήση του προγραμματιστικού μοντέλου MPI που χρησιμοποιείται ευρέως για την παραλληλοποίηση εφαρμογών μεγάλης κλίμακας. Αρχικά θα μελετηθούν υπάρχουσες υλοποιήσεις από διαδεδομένες βιβλιοθήκες λογισμικού όπως το PETSc (<https://www.mcs.anl.gov/petsc/>) και στη συνέχεια θα γίνουν βελτιστοποιήσεις με χρήση κατάλληλων τεχνικών κατανομής εργασίας και συμπίεσης δεδομένων.

Σχετικά Μαθήματα: Συστήματα Παράλληλης Επεξεργασίας

Σχετική Βιβλιογραφία:

1. Lewis, John G., and Robert A. Van de Geijn. "Distributed memory matrix-vector multiplication and conjugate gradient algorithms." Supercomputing'93: Proceedings of the 1993 ACM/IEEE Con-

ference on Supercomputing. IEEE, 1993.

2. Vastenhouw, Brendan, and Rob H. Bisseling. "A two-dimensional data distribution method for parallel sparse matrix-vector multiplication." *SIAM review* 47.1 (2005): 67-95.
3. Çatalyürek, ÜmiT. V., Cevdet Aykanat, and Bora Uçar. "On two-dimensional sparse matrix partitioning: Models, methods, and a recipe." *SIAM Journal on Scientific Computing* 32.2 (2010): 656-683.
4. PETSc.

Επικοινωνία: Αθηνά Ελαφρού, athena@cslab.ece.ntua.gr, 210-772-2133

Γεώργιος Γκούμας, goumas@cslab.ece.ntua.gr, 210-772-2402

7 Βελτιστοποίηση του υπολογιστικού πυρήνα πολλαπλασιασμού αραιού πίνακα με δiάνυσμα (SpMV) σε FPGAs με τη χρήση του προγραμματιστικού μοντέλου OpenCL

Ο υπολογιστικός πυρήνας του πολλαπλασιασμού αραιού πίνακα με δiάνυσμα (SpMV) χρησιμοποιείται ευρέως σε παράλληλες εφαρμογές μεγάλης κλίμακας. Ωστόσο, λόγω της αλγοριθμικής του φύσης, δεν αξιοποιεί επαρκώς την υπολογιστική ισχύ των σύγχρονων επεξεργαστών. Στην παρούσα διπλωματική εργασία, θα μελετηθεί η υλοποίηση και η βελτιστοποίηση του συγκεκριμένου υπολογιστικού πυρήνα σε επαναδιαμορφούμενες αρχιτεκτονικές (FPGAs), που επιτρέπουν στον προγραμματιστή τη δημιουργία υλικού εξειδικευμένου στην εφαρμογή (application-specific). Συγκεκριμένα, θα μελετηθεί η επίδοση βασικών υλοποιήσεων του SpMV για FPGAs με τη χρήση του προγραμματιστικού μοντέλου της OpenCL, εναλλακτικά σχήματα αποθήκευσης αραιών πινάκων και θα εφαρμοστούν τεχνικές βελτιστοποίησης του υπολογιστικού πυρήνα με στόχο την επίτευξη της μέγιστης δυνατής επίδοσης στις συγκεκριμένες αρχιτεκτονικές.

Σχετικά Μαθήματα: Συστήματα Παράλληλης Επεξεργασίας, Ψηφιακά Συστήματα VLSI

Επικοινωνία: Νικέλα Παπαδοπούλου, nikela@cslab.ece.ntua.gr, 210-772-2279

Γεώργιος Γκούμας, goumas@cslab.ece.ntua.gr, 210-772-2402

8 Υλοποίηση βιβλιοθήκης για τον πολλαπλασιασμό αραιού πίνακα με δiάνυσμα σε ετερογενή συστήματα με τη χρήση του προγραμματιστικού μοντέλου OmpSs

Το προγραμματιστικό μοντέλο OmpSs (<https://pm.bsc.es/ompss>) αποτελεί μια επέκταση του προγραμματιστικού μοντέλου OpenMP, που υποστηρίζει ετερογενείς αρχιτεκτονικές, μεταξύ των οποίων και επαναδιαμορφούμενες αρχιτεκτονικές (FPGAs). Η παρούσα διπλωματική θα στηριχθεί σε υπάρχουσες υλοποιήσεις του υπολογιστικού πυρήνα αραιού πίνακα με δiάνυσμα (SpMV) για ετερογενείς αρχιτεκτονικές και τις ενσωματώσει σε μία βιβλιοθήκη λογισμικού, βασισμένη στο προγραμματιστικό μοντέλο OmpSs, που θα αξιοποιεί με βέλτιστο τρόπο τους διαθέσιμους υπολογιστικούς πόρους ετερογενών συστημάτων, επιλέγοντας στο χρόνο μεταγλώττισης ή/και εκτέλεσης τους καταλληλότερους

υπολογιστικούς πυρήνες για την εκτέλεση εφαρμογών, με επίγνωση τόσο της αρχιτεκτονικής, όσο και των χαρακτηριστικών της εφαρμογής και του δεδομένου αραιού πίνακα.

Σχετικά Μαθήματα: Συστήματα Παράλληλης Επεξεργασίας

Επικοινωνία: Νικέλα Παπαδοπούλου, nikela@cslab.ece.ntua.gr, 210-772-2279

Γεώργιος Γκούμας, goumas@cslab.ece.ntua.gr, 210-772-2402

9 Υλοποίηση και μελέτη επίδοσης εφαρμογών υψηλής έντασης σε ετερογενή συστήματα (CPU, GPU, FPGA) με τη χρήση του προγραμματιστικού μοντέλου OpenCL

Την τελευταία δεκαετία η αύξηση της πολυπλοκότητας μιας ευρείας κατηγορίας υπολογιστικών εφαρμογών και η κατάρρευση της κλιμάκωσης της ενεργειακής επίδοσης των επεξεργαστών γενικού σκοπού προκάλεσε μια στροφή προς τα ετερογενή υπολογιστικά συστήματα (heterogeneous computing). Η ενορχηστρωμένη χρήση επεξεργαστών γενικού σκοπού (CPU) και προγραμματιζόμενων επιταχυντών ειδικού σκοπού (HW accelerators) για υπολογιστικές εφαρμογές υψηλής έντασης εφαρμόζεται ήδη σε υπολογιστικά συστήματα κέντρων δεδομένων (data centers), αλλά και σε συστήματα υψηλών επιδόσεων (High Performance Computing -HPC). Στόχος η αύξηση της απόδοσης με όρους επίδοσης (performance) αλλά και κατανάλωσης ισχύος (power efficiency). Οι επαναδιαμορφούμενες αρχιτεκτονικές, γνωστές και με τον αγγλικό όρο FPGAs (Field Programmable Gate Arrays), είναι ένα ισχυρό υπολογιστικό υλικό το οποίο επιτρέπει στους σχεδιαστές τη δημιουργία συστημάτων εξειδικευμένων εφαρμογών (application-specific). Παρά τη μεγάλη ευελιξία και τη συχνή υπεροχή τους σε ενεργειακή κατανάλωση/επίδοση συγκριτικά με τους επεξεργαστές γενικού σκοπού αλλά και με άλλους επιταχυντές (π.χ GPU), ο προγραμματισμός τους με μοντέλα επιπέδου μεταφοράς καταχωρητή (RTL-based programming model) δρούσε ανασταλτικά στην ευρεία χρήση τους. Τα τελευταία χρόνια όμως αναπτύχθηκαν προηγμένα εργαλεία σύνθεσης (HLS) με στόχο την αυτοματοποίηση τη διαδικασίας σχεδίασης, και πρόσφατα άρχισε να υποστηρίζεται/ χρησιμοποιείται το προγραμματιστικό μοντέλο της OpenCL που στοχεύει στον παράλληλο προγραμματισμό ετερογενών συστημάτων (CPU, GPU, FPGA, DSP κ.α) διευκολύνοντας παράλληλα τη φορητότητα μεταξύ διαφορετικών πλατφορμών. Η εξέλιξη αυτή σε συνδυασμό με την εξέλιξη των αρχιτεκτονικών FPGA, καθιστά τις τελευταίες μια ελκυστική τεχνολογία επιταχυντών ευρείας χρήσης. Ανακοινώσεις μεγάλων εταιριών (Intel, IBM, Microsoft) για λύσεις που συνδέουν CPUs και FPGAs είναι στοιχεία που καταδεικνύουν τη στροφή αυτή.

Η παρούσα διπλωματική περιλαμβάνει εξοικείωση με το προγραμματιστικό μοντέλο της OpenCL, υλοποίηση αλγορίθμων υψηλής έντασης σε ετερογενή συστήματα (FPGA και GPU) και εφαρμογή τεχνικών βελτιστοποίησης στηριζόμενων στις ιδιαιτερότητες της κάθε αρχιτεκτονικής. Τέλος περιλαμβάνει τη σύγκριση των διάφορων υλοποιήσεων με όρους επίδοσης και κατανάλωσης ισχύος.

Σχετικά Μαθήματα: Συστήματα Παράλληλης Επεξεργασίας, Ψηφιακά Συστήματα VLSI

Επικοινωνία: Χλόη Αλβέρτη, xalverti@cslab.ece.ntua.gr, 210-772-2279

Νικέλα Παπαδοπούλου, nikela@cslab.ece.ntua.gr, 210-772-2495

Γεώργιος Γκούμας, goumas@cslab.ece.ntua.gr, 210-772-2402

10 Ανάλυση και μοντελοποίηση της επίδοσης εφαρμογών σε αρχιτεκτονικές με ανομοιόμορφη πρόσβαση μνήμης (Non Uniform Memory Access - NUMA) και υλοποίηση αποτελεσματικής κατανομής και χρονοδρομολόγησης

Ένας από τους τρόπους αύξησης της κλιμακωσιμότητας των υπολογιστικών συστημάτων που έχουν πολλαπλούς επεξεργαστές στο ίδιο σύστημα είναι μέσω των Αρχιτεκτονικών με Ανομοιόμορφη Πρόσβαση Μνήμης ή αλλιώς Non Uniform Memory Access (NUMA) systems. Η NUMA αρχιτεκτονική αποτελεί έναν σχεδιασμό συστήματος μνήμης πολυεπεξεργαστικών υπολογιστικών συστημάτων, στα οποία ο χρόνος πρόσβασης της κύριας μνήμης (RAM) εξαρτάται από την απόσταση της θέσης μνήμης που προσπελάζει ο επεξεργαστής. Σε μία NUMA αρχιτεκτονική, ένας επεξεργαστής έχει γρηγορότερη πρόσβαση σε μία τοπική μνήμη από ότι σε μία απομακρυσμένη, η οποία όμως είναι τοπική για κάποιον άλλον επεξεργαστή. Στόχος της παρούσας διπλωματικής είναι η μελέτη και η ανάλυση της επίδοσης που προκαλεί η NUMA αρχιτεκτονική κατά την εκτέλεση ή/και τη συνεκτέλεση σύγχρονων εφαρμογών, καθώς επίσης και η αναγνώριση προτύπων και συμπεριφορών, η κατηγοριοποίηση των εφαρμογών σε ευαίσθητες και μη-ευαίσθητες ως προς την συμπεριφορά τους λόγω της NUMA αρχιτεκτονικής, και η μοντελοποίηση τους. Επιπλέον, θα υλοποιηθεί εργαλείο και πολιτικές αποτελεσματικής χρονοδρομολόγησης πολλαπλών εφαρμογών, τόσο μέσω της κατάλληλης κατανομής των κόμβων μνήμης (memory node allocation) όσο και μέσω της κατάλληλης επιλογής των υπολογιστικών κόμβων (compute node allocation).

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών, Εργαστήριο Λειτουργικών Συστημάτων

Επικοινωνία: Βασίλης Καρακώστας, vkarakos@cslab.ece.ntua.gr, 210-772-4133
Νικέλα Παπαδοπούλου, nikela@cslab.ece.ntua.gr, 210-772-2495
Κωστής Νίκας, knikas@cslab.ece.ntua.gr, 210-772-4159

11 Εφαρμογή αλγορίθμων μηχανικής μάθησης στην αρχιτεκτονική υπολογιστών

Αλγόριθμοι μηχανικής μάθησης ταξινόμησης (classification) και πρόβλεψης (prediction) εφαρμόζονται πλέον κατά κανόνα σε τομείς όπως η όραση υπολογιστών, η επεξεργασία φυσικής γλώσσας κ.α, πετυχαίνοντας εντυπωσιακά αποτελέσματα. Και ενώ συχνά σχεδιάζεται εξειδικευμένο hardware για την επιτάχυνση τους, λίγες είναι προς το παρόν οι περιπτώσεις εφαρμογής/χρήσης τους για τη βελτίωση της ίδιας της επίδοσης ενός υπολογιστικού συστήματος.

Οι σύγχρονες αρχιτεκτονικές συχνά εμπλέκουν ευριστικές μεθόδους, μεθόδους πρόβλεψης/υποθετικής εκτέλεσης για τη μεγιστοποίηση της επίδοσης ενός συστήματος. Παράδειγμα μπορεί να θεωρηθεί η χρήση προανάκλησης (prefetching), που χρησιμοποιείται για την αντιμετώπιση ενός σημαντικού σημείου συμφόρησης (bottleneck) επίδοσης των σύγχρονων αρχιτεκτονικών, του κόστους προσπέλασης της κύριας μνήμης. Σκοπός της συγκεκριμένης διπλωματικής είναι η διερεύνηση της δυνατότητας εφαρμογής αλγορίθμων μηχανικής μάθησης πρόβλεψης ακολουθιών (sequence prediction) κ.α στην αρχιτεκτονική για τη βελτιστοποίηση της επίδοσης του υποσυστήματος μνήμης. Αυτό μπορεί να περιλαμβάνει βελτιστοποιήσεις στη χρήση των κρυφών μνημών (caches-prefetching), στο μηχανισμό ειδικής μνήμης (TLBs) κ.α.

Στόχος είναι αρχικά να χρησιμοποιηθεί λογισμικό μηχανικής μάθησης (π.χ pytorch) με πραγματικά δεδομένα από σύγχρονα μηχανήματα για τη μελέτη διαφορετικών μοντέλων (π.χ LSTMs). Στη συνέχεια, ανάλογα με τα συμπεράσματα του πρώτου βήματος, θα επιχειρήσουμε να αξιολογήσουμε τη δυνατότητα εφαρμογής τους σε επίπεδο μικροαρχιτεκτονικής λαμβάνοντας υπόψη την πολυπλοκότητα, το χρόνο απόκρισης και την κατανάλωση χώρου και ενέργειας.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

Σχετική Βιβλιογραφία:

1. Learning Memory Access Patterns
2. Dynamic Branch Prediction with Perceptrons
3. Virtual Address Translation via Learned Page Table Indexes
4. SmartChoices: Hybridizing Programming and Machine Learning
5. Applying Deep Learning to the Cache Replacement Problem

Επικοινωνία: Χλόη Αλβέρτη, xalverti@cslab.ece.ntua.gr, 210-772-2279

Βασίλης Καρακώστας, vkarakos@cslab.ece.ntua.gr, 210-772-4133

Κωστής Νίκας, knikas@cslab.ece.ntua.gr, 210-772-4159

12 Μελέτη, σχεδιασμός, και υλοποίηση επεκτάσεων των μηχανισμών εικονικής μνήμης

Η εικονική μνήμη αποτελεί θεμελιώδη “αφαίρεση” ενός υπολογιστικού συστήματος. Κάθε διεργασία χρησιμοποιεί το δικό της συνεχή χώρο εικονικών διευθύνσεων για να προσπελάσει τη μνήμη, ενώ το λειτουργικό σύστημα είναι υπεύθυνο για να αντιστοιχίσει τις εικονικές διευθύνσεις σε φυσικές διευθύνσεις μηχανήματος και να αποθηκεύσει/διατηρήσει/διαχειριστεί αυτές τις αντιστοιχίσεις σε πίνακες σελίδων (page tables) για κάθε διεργασία. Για να εξυπηρετήσει μια εντολή προσπέλασης μνήμης ο επεξεργαστής πρέπει να μεταφράσει την εικονική διεύθυνση (virtual address) σε φυσική (physical address). Για λόγους επίδοσης, οι επεξεργαστές χρησιμοποιούν εξειδικευμένες κρυφές μνήμες (Translation Lookaside Buffers, TLBs) για την αποθήκευση των πιο πρόσφατων μεταφράσεων. Επιπλέον, η εικονικοποίηση (virtualization) είναι μια τεχνική που επιτρέπει την ταυτόχρονη εκτέλεση πολλών εικονικών μηχανών (virtual machines ή guest VMs) σε ένα φυσικό μηχάνημα (host). Κάθε guest τρέχει το δικό του λειτουργικό σύστημα, έχοντας τη ψευδαίσθηση ότι έχει πλήρη και αποκλειστική πρόσβαση σε εικονικούς πόρους. Ειδικό λογισμικό (Virtual Machine Monitor) που τρέχει στο host είναι υπεύθυνο για την ασφαλή πρόσβαση του VM στους πραγματικούς φυσικούς πόρους του συστήματος (π.χ μνήμη). Ένας τρόπος μείωσης του κόστους της εικονικής μνήμης είναι η αύξηση του TLB reach, δηλαδή του εύρους των εικονικών διευθύνσεων που το TLB μπορεί να μεταφράσει. Ένας τρόπος αύξησης του TLB reach είναι η χρήση μεγάλων virtual-to-physical mappings. Ωστόσο η δημιουργία και η διαχείριση μεγάλων mappings, επιφέρει δυσκολίες/προκλήσεις στο memory management (π.χ fragmentation, memory bloat, unfairness, page sharing). Σκοπός της παρούσας εργασίας είναι η μελέτη αυτών των προβλημάτων και η διερεύνηση πολιτικών διαχείρισης με χρήση των huge pages σε συνέχεια προηγούμενης διπλωματικής εργασίας.

Σχετικά Μαθήματα: Εργαστήριο Λειτουργικών Συστημάτων, Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

Σχετική Βιβλιογραφία:

1. Ingens: Huge Page Support for the OS and Hypervisor
2. HawkEye: Efficient Fine-grained OS Support for Huge Pages
3. Nimble Page Management for Tiered Memory Systems
4. MEGA: overcoming traditional problems with OS huge page management

Επικοινωνία: Βασίλης Καρακώστας, vkarakos@cslab.ece.ntua.gr, 210-772-4133
Χλόη Αλβέρτη, xalverti@cslab.ece.ntua.gr, 210-772-2279

13 Σχεδιασμός και υλοποίηση μηχανισμών σε επίπεδο αρχιτεκτονικής για την εκτίμηση της ενεργά χρησιμοποιούμενης μνήμης (working set estimation)

Η έννοια της "ενεργά χρησιμοποιούμενης μνήμης" (ή αλλιώς active working set) ορίστηκε από τον Peter Denning το 1968 ως το σύνολο των δεδομένων που η κάθε διεργασία προσπελάζει σε ένα συγκεκριμένο χρονικό παράθυρο. Το active working set αποτελεί σημαντική μετρική στην οποία βασίζονται πολιτικές διαμοιρασμού πόρων (π.χ. μνήμης). Ο υπολογισμός του active working set πραγματοποιείται συνήθως στο λογισμικό, χρησιμοποιώντας υπάρχοντες μηχανισμούς του λειτουργικού συστήματος και της αρχιτεκτονικής, όπως για παράδειγμα τα access και dirty bits του πίνακα σελίδων (page table). Στόχος της διπλωματικής αυτής είναι η μελέτη, ο σχεδιασμός και η υλοποίηση νέων μηχανισμών εκτίμησης του working set σε επίπεδο αρχιτεκτονικής/υλικού.

Σχετικά Μαθήματα: Εργαστήριο Λειτουργικών Συστημάτων, Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

Επικοινωνία: Βασίλης Καρακώστας, vkarakos@cslab.ece.ntua.gr, 210-772-4133
Χλόη Αλβέρτη, xalverti@cslab.ece.ntua.gr, 210-772-2279

14 Μελέτη περιβάλλοντος ανάπτυξης και υλοποίηση επιταχυντών σε RISC-V αρχιτεκτονική

Η RISC-V αρχιτεκτονική είναι ένα σύνολο εντολών αρχιτεκτονικής (ISA) ανοιχτού κώδικου που έχει τραβήξει πρόσφατα μεγάλο ενδιαφέρον τόσο από τον ακαδημαϊκό χώρο όσο και από τον χώρο της βιομηχανίας, με κατάλληλη υποστήριξη σε όλα τα επίπεδα της υπολογιστικής στοίβας (υλικό, λειτουργικό σύστημα, βιβλιοθήκες, μεταγλωττιστές, κτλ). Στόχος της παρούσας διπλωματικής εργασίας είναι η μελέτη του περιβάλλοντος ανάπτυξης υλικού του Rocket Chip Generator που αναπτύσσεται από το Πανεπιστήμιο του Berkeley και θα εστιάσουμε στην ανάπτυξη επιταχυντών σε RISC-V αρχιτεκτονικές χρησιμοποιώντας το framework του Rocket Custom Coprocessor (RoCC).

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

Σχετική Βιβλιογραφία:

1. A Hardware Accelerator for Tracing Garbage Collection
2. <https://en.wikipedia.org/wiki/RISC-V>

3. <https://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-17.pdf>

Επικοινωνία: Βασίλης Καρακώστας, vkarakos@cslab.ece.ntua.gr, 210-772-4133
Κωστής Νίκας, knikas@cslab.ece.ntua.gr, 210-772-4159

15 Μεταφορά μετροπρογραμμάτων, πειραματική αξιολόγηση και ανάπτυξη βελτιστοποιήσεων για RISC-V development board

Η RISC-V αρχιτεκτονική είναι ένα σύνολο εντολών αρχιτεκτονικής (ISA) ανοιχτού κώδικού που έχει τραβήξει πρόσφατα μεγάλο ενδιαφέρον τόσο από τον ακαδημαϊκό χώρο όσο και από τον χώρο της βιομηχανίας, με κατάλληλη υποστήριξη σε όλα τα επίπεδα της υπολογιστικής στοίβας (υλικό, λειτουργικό σύστημα, βιβλιοθήκες, runtime systems, μεταγλωττιστές, κτλ). Στόχος της παρούσας διπλωματικής εργασίας είναι η μεταφορά εφαρμογών (porting) και η πειραματική αξιολόγηση εκτέλεσης αυτών των εφαρμογών στο "HiFive Unleashed" αναπτυξιακό board αρχιτεκτονικής RISC-V. Τέλος, θα μελετηθούν μηχανισμοί βελτιστοποίησης σε επίπεδο porting εφαρμογών / runtime systems / μεταγλωττιστή.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

Σχετική Βιβλιογραφία:

1. <https://en.wikipedia.org/wiki/RISC-V>
2. <https://www.sifive.com/boards/hifive-unleashed>

Επικοινωνία: Βασίλης Καρακώστας, vkarakos@cslab.ece.ntua.gr, 210-772-4133
Κωστής Νίκας, knikas@cslab.ece.ntua.gr, 210-772-4159

16 Αποδοτική απεικόνιση επεξεργαστών RISC-V σε FPGA

Η αρχιτεκτονική RISC-V είναι μια ανοικτή και επεκτάσιμη αρχιτεκτονική συνόλου εντολών. Πολλές υλοποιήσεις ανοιχτού κώδικα είναι διαθέσιμες, άλλες απλούστερες με γραμμική in-order pipeline και άλλες μεγαλύτερων επιδόσεων με πυρήνα εκτέλεσης εντολών εκτός σειράς (out-of-order). Οι FPGA αν και μπορούν να απεικονίσουν οποιαδήποτε σχεδίαση υλικού, έχουν ιδιαιτερότητες και «προτιμήσεις». Η εργασία αυτή αφορά αφενός την συγκριτική μελέτη βασικών υπαρχόντων υλοποιήσεων ως προς το κόστος και τις επιδόσεις τους όταν υλοποιούνται με διαφορετικές FPGA, και αφετέρου την παραμετροποίηση των εσωτερικών δομών (η την αντικατάστασή τους με άλλες ισοδύναμες) ώστε η συνολική σχεδίαση να είναι περισσότερο «φιλική» προς τις FPGA. Η εργασία συνδυάζει προσομοιώσεις για την μέτρηση επιδόσεων σε αρχιτεκτονικό επίπεδο και απεικόνιση των αρχιτεκτονικών σε FPGA με εργαλεία CAD.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

Σχετική Βιβλιογραφία:

1. <https://en.wikipedia.org/wiki/RISC-V>
2. <https://github.com/pulp-platform/riscv>
3. <https://tspace.library.utoronto.ca/handle/1807/80713>
4. <https://github.com/riscv-boom/riscv-boom>

Επικοινωνία: Διονύσιος Πνευματικάτος, pnevmati@cslab.ece.ntua.gr, 6944763171

17 Επεκτάσεις του RISC-V για near/in memory accelerators

Η αρχιτεκτονική RISC-V είναι μια ανοικτή και επεκτάσιμη αρχιτεκτονική συνόλου εντολών και ως τέτοια προσφέρεται για την έρευνα σε λειτουργικές επεκτάσεις. Η επεξεργασία δεδομένων κοντά η ακόμα και εντός της μνήμης (near/in-memory) υπόσχεται βελτιωμένες επιδόσεις και καλύτερη ενεργειακή απόδοση.

Η εργασία αυτή αφορά αφενός την δημιουργία ενός μικρού πυρήνα RISC-V ο οποίος θα είναι ο δομικός λίθος για επεξεργασία κοντά στην μνήμη για επεξεργασία μεγάλων δεδομένων (η αρχιτεκτονική αναφοράς είναι το Modrian Data Engine). Έτσι οι βασικές συναρτήσεις θα είναι ερωτήσεις βάσεων δεδομένων στις οποίες τα δεδομένα βρίσκονται στην μνήμη. Το σύστημα μνήμης θα αποτελείται από ένα (η περισσότερα) HMC modules. Συγκεκριμένα, τα βήματα της εργασίας είναι (α) η επιλογή και επέκταση ενός υπάρχοντος πυρήνα RISC-V με εντολές vector (β) ο προγραμματισμός των βασικών λειτουργιών και η επιβεβαίωση ορθής λειτουργίας με προσομοιώσεις, (γ) η ολοκλήρωσή του επεξεργαστή στο περιβάλλον FPGA+HMC της Micron και (δ) η αξιολόγηση του συνολικού συστήματος.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

Σχετική Βιβλιογραφία:

1. <https://en.wikipedia.org/wiki/RISC-V>
2. https://en.wikipedia.org/wiki/Vector_processor
3. https://en.wikipedia.org/wiki/Hybrid_Memory_Cube
4. <https://pure.tue.nl/ws/files/100178113/gagan2018dsd.pdf>
5. <https://arxiv.org/pdf/1908.02640.pdf>
6. The Mondrian Data Engine: <https://dl.acm.org/citation.cfm?id=3080233>
7. <https://www.sigarch.org/simd-instructions-considered-harmful/>
8. <https://www.youtube.com/watch?v=GzZ-8bHsD5s>

Επικοινωνία: Διονύσιος Πνευματικάτος, pnevmati@cslab.ece.ntua.gr, 6944763171

18 Enabling Processing in Memory

Modern multiprocessor systems are overwhelmingly designed to move data from main memory to the CPU cores for computation. Data intensive and irregular workloads are bottlenecked by the extensive data movement between processor and memory. As a result, there is significant latency and energy cost to the inherent data movement required in contemporary systems.

Recent architecture advances in 3D-stack memories have renewed the interest in processing-in-memory (PIM) architectures. PIM is a promising paradigm to alleviate data movement between processing units and memory by placing simple processors close to memory. Even though the PIM approach by itself is not new and has been proposed for decades, recent advances in die-stacking technologies enable the low-cost integration of a dedicated logic layer with multiple layers of DRAM arrays in a single package. Two of the most prominent 3D-stacked memory technologies are Hybrid Memory Cube (HMC)[1] and High Bandwidth Memory (HBM)[2].

Many recent works explore how various applications can benefit from PIM. Such applications include graph processing [3,4], neural networks, bioinformatics, databases, security, and data manipulation. Generally, PIM architectures can provide performance benefits for applications that are memory bound. An application is called memory bound if it exhibits one or more of the following three cases: 1) it requires more memory bandwidth than the available memory bandwidth that the memory system can provide, 2) its performance is sensitive to memory access latency, or 3) it performs irregular memory accesses (i.e. has poor temporal and spatial locality), such that it cannot benefit from three-level cache hierarchy of conventional CPU architectures.

The goal of this thesis is to understand all major sources of inefficiency that lead to data movement bottlenecks, characterize data-intensive workloads based on the source of their memory bottleneck and evaluate the benefits of executing data-intensive applications inside specialized logic in memory developing both mechanisms and simulators for this purpose.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

Σχετική Βιβλιογραφία:

1. Hybrid Memory Cube Consortium, hybridmemorycube.org
2. 25.2 A 1.2V 8Gb 8-channel 128GB/s high-bandwidth memory (HBM) stacked DRAM with effective microbump I/O test methods using 29nm process and TSV, ISSCC 2014.
3. "PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture", ISCA 2015.
4. "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing", ISCA 2015.
5. "Adaptive Scheduling for Systems with Asymmetric Memory Hierarchies", MICRO 2018.
6. "GraphPIM: Enabling Instruction-Level PIM Offloading in Graph Computing Frameworks", HPCA 2017.
7. "The Mondrian Data Engine", ISCA 2017.
8. "GraphQ: Scalable PIM-Based Graph Processing", MICRO 2019.

Επικοινωνία: Χριστίνα Γιαννούλα, christina.giann@gmail.com

Μέρος II

Λειτουργικά Συστήματα - Εικονικές Μηχανές

19 Υλοποίηση utmem (Userspace Transcendent Memory) σε ARM πλατφόρμες

Η transcendent μνήμη (transcendent memory (tmem)) αποτελεί μία προσέγγιση για καλύτερη χρήση της υποχρησιμοποιούμενης μνήμης σε ένα εικονικοποιημένο περιβάλλον. Σε τέτοιες περιπτώσεις ο host μπορεί με δυναμικό τρόπο να διαχειρίζεται αποδοτικότερα ένα tmem pool μεταξύ των εικονικών μηχανών (VMs). Ένας τέτοιος μηχανισμός φαντάζει ακόμα περισσότερο χρήσιμος σε πλατφόρμες με μειωμένους πόρους σε μνήμη, όπως πχ ένα ενσωματωμένο board. Σκοπός αυτής της εργασίας είναι η μελέτη του μηχανισμού utmem (Userspace Transcendent Memory) προηγούμενης διπλωματικής εργασίας, η τροποποίηση (porting) του για ARM πλατφόρμες και η πειραματική του αποτίμηση.

Σχετικά Μαθήματα: Λειτουργικά Συστήματα, Εργαστήριο Λειτουργικών Συστημάτων

Επικοινωνία: Στέφανος Γεράγγελος, sgerag@cslab.ece.ntua.gr

20 Υλοποίηση utmem (Userspace Transcendent Memory) ως unikernel

Η transcendent μνήμη (transcendent memory (tmem)) αποτελεί μία προσέγγιση για καλύτερη χρήση της υποχρησιμοποιούμενης μνήμης σε ένα εικονικοποιημένο περιβάλλον. Σε τέτοιες περιπτώσεις ο host μπορεί με δυναμικό τρόπο να διαχειρίζεται αποδοτικότερα ένα tmem pool μεταξύ των εικονικών μηχανών (VMs). Ταυτόχρονα, μία ενδιαφέρουσα προσέγγιση στη θεματική περιοχή του lightweight virtualization για τη μείωση του θορύβου του λειτουργικού συστήματος και περιττών εξαρτήσεων στο περιβάλλον εκτέλεσης μιας εφαρμογής είναι η δημιουργία ενός λεπτού στρώματος εξαρτήσεων (βιβλιοθήκες, λειτουργικό σύστημα) και η σύνθεση ενός ενιαίου εκτελέσιμου αρχείου της εφαρμογής (unikernel), που θα μπορεί να εκτελεστεί αυτόνομα, όπως σε ένα κοινό λειτουργικό σύστημα. Σκοπός της παρούσας εργασίας είναι η μελέτη του μηχανισμού utmem (Userspace Transcendent Memory) προηγούμενης διπλωματικής εργασίας, η ενσωμάτωση του σε κάποιο unikernel framework και η πειραματική του αποτίμηση.

Σχετικά Μαθήματα: Λειτουργικά Συστήματα, Εργαστήριο Λειτουργικών Συστημάτων

Επικοινωνία: Στράτος Ψωμαδάκης, psomas@cslab.ece.ntua.gr

21 Υποστήριξη fork() σε περιβάλλοντα unikernels

Μια ενδιαφέρουσα προσέγγιση στη μείωση του θορύβου του λειτουργικού συστήματος και περιττών εξαρτήσεων στο περιβάλλον εκτέλεσης μιας εφαρμογής είναι η δημιουργία ενός λεπτού στρώματος εξαρτήσεων (βιβλιοθήκες, λειτουργικό σύστημα) και η σύνθεση ενός ενιαίου εκτελέσιμου αρχείου της

εφαρμογής (unikernel), που θα μπορεί να εκτελεστεί αυτόνομα, όπως σε ένα κοινό λειτουργικό σύστημα. Παρά το γεγονός ότι αρκετά unikernel frameworks υποστηρίζουν POSIX λειτουργίες, υπάρχουν κάποιες από αυτές οι οποίες είναι εγγενώς δύσκολες να υποστηριχτούν, όπως η κλήση συστήματος fork(). Σκοπός αυτής της εργασίας είναι η υλοποίηση μηχανισμού προκειμένου να υποστηρίζεται η κλήση συστήματος fork() κάνοντας χρήση κάποιας διαφορετικής τεχνικής (π.χ. threads).

Σχετικά Μαθήματα: Λειτουργικά Συστήματα, Εργαστήριο Λειτουργικών Συστημάτων

Επικοινωνία: Ορέστης Λάγκας-Νικολός, olagkas@cslab.ece.ntua.gr

22 Επιτάχυνση υλικού για αποδοτική εκτέλεση εφαρμογών ως unikernels

Μια ενδιαφέρουσα προσέγγιση στη μείωση του θορύβου του λειτουργικού συστήματος και περιττών εξαρτήσεων στο περιβάλλον εκτέλεσης μιας εφαρμογής είναι η δημιουργία ενός λεπτού στρώματος εξαρτήσεων (βιβλιοθήκες, λειτουργικό σύστημα) και η σύνθεση ενός ενιαίου εκτελέσιμου αρχείου της εφαρμογής (unikernel), που θα μπορεί να εκτελεστεί αυτόνομα, όπως σε ένα κοινό λειτουργικό σύστημα. Ταυτόχρονα, η διεύρυνση της χρήσης επιταχυντών υλικού για υπολογιστικά απαιτητικά κομμάτια εφαρμογών καθιστά το υλικό περισσότερο προσβάσιμο, και άρα διαθέσιμο σε περιβάλλοντα cloud (Amazon AWS, Azure, κλπ). Στόχος της παρούσας εργασίας είναι η σχεδίαση και υλοποίηση ενός συστήματος που θα συνδυάζει την απάλειψη περιττών εξαρτήσεων της εφαρμογής από το περιβάλλον εκτέλεσης (unikernel) καθώς και την ένταξη επιτάχυνσης συγκεκριμένων υπολογιστικά απαιτητικών κομματιών της εφαρμογής. Συγκεκριμένα, η εργασία περιλαμβάνει: (α) μελέτη των διαθέσιμων frameworks για unikernels, (β) αποδελτίωση εφαρμογών που αξιοποιούν την επιτάχυνση υλικού σε GPUs/FPGAs, (γ) υλοποίηση του συστήματος που παράγει unikernels με αυτή την υποστήριξη. και (δ) πειραματική αποτίμηση του συστήματος.

Σχετικά Μαθήματα: Λειτουργικά Συστήματα, Εργαστήριο Λειτουργικών Συστημάτων

Σχετική Βιβλιογραφία:

- Unikernel frameworks:
 1. <https://github.com/cloudius-systems/osv>
 2. <http://rumpkernel.org/>
 3. <https://github.com/libos-nuse/lkl-linux>
 4. <http://cnp.neclab.eu/clickos/>
 5. <https://wiki.xenproject.org/wiki/Mini-OS>
- Acceleration:
 1. <https://www.khronos.org/ocl/>
 2. <https://www.xilinx.com/products/design-tools/software-zone/sdaccel.html>

Επικοινωνία: Κωνσταντίνος Παπαζαφειρόπουλος, krapazaf@cslab.ece.ntua.gr

23 Μελέτη λειτουργικών συστημάτων για manycore αρχιτεκτονικές επιταχυντών

Την τελευταία δεκαετία παρατηρείται επιβράδυνση στην κλιμάκωση των πολυπύρηνων (multicore) επεξεργαστών γενικού σκοπού για λόγους που σχετίζονται με την ενεργειακή πυκνότητα στα ολοένα και μικρότερου μεγέθους transistors. Μία απάντηση σε αυτήν την τάση είναι η χρησιμοποίηση ετερογενών συστημάτων και πιο συγκεκριμένα επιταχυντών. Τα λειτουργικά συστήματα γενικού σκοπού δεν είναι τα καταλληλότερα για τέτοιους είδους περιβάλλοντα, καθώς με την πληθώρα των λειτουργιών που προσφέρουν, προσθέτουν αχρείαστο θόρυβο κατά την εκτέλεση των εφαρμογών. Σε αυτό το πλαίσιο, έχουν αναπτυχθεί ελαφρύτερα συστήματα για manycore αρχιτεκτονικές με στόχο την καλύτερη κλιμακωσιμότητα των εφαρμογών. Σκοπός αυτής της διπλωματικής είναι η μελέτη λειτουργικών συστημάτων εξειδικευμένων για συσκευές επιταχυντών, η αξιολόγηση κάποιου από αυτά σε συγκεκριμένο επιταχυντή και η πιθανή επέκτασή του με βάση τα αποτελέσματα της προαναφερθείσας αξιολόγησης.

Σχετικά Μαθήματα: Λειτουργικά Συστήματα, Εργαστήριο Λειτουργικών Συστημάτων

Επικοινωνία: Στέφανος Γεράγγελος, sgerag@cslab.ece.ntua.gr

Κωστής Παπαζαφειρόπουλος, krapazaf@cslab.ece.ntua.gr

Στράτος Ψωμαδάκης, psomas@cslab.ece.ntua.gr

Μέρος III

Κατανεμημένα Συστήματα - Προχωρημένα θέματα βάσεων δεδομένων

24 Αυτόματη επιλογή και ταξινόμηση δεδομένων εισόδου βάσει χρησιμότητας για αναλυτικές εργασίες μεγάλου όγκου δεδομένων

Ενώ η βελτιστοποίηση εργασιών στον τομέα των Big Data συνήθως υλοποιείται με την αύξηση της παραλληλοποίησης και τη χρήση πλατφορμών κατανεμημένης επεξεργασίας, λίγα έχουν γίνει σχετικά με την επιλογή των πιο κατάλληλων δεδομένων από μια μεγάλη συλλογή διαθέσιμων. Πολλές αναλυτικές εργασίες είναι ιδιαίτερα ευαίσθητες στο περιεχόμενο των δεδομένων και όχι τόσο στο μέγεθός τους (π.χ., content based advertising, social network analytics). Στις εργασίες [1, 2] παρουσιάσαμε μια γενική μεθοδολογία για γρήγορη σύγκριση και ταξινόμηση πολλαπλών διαθέσιμων δεδομένων εισόδου (datasets) με βάση το αποτέλεσμα που προκαλούν όταν εφαρμόζονται σε τελεστές αναλυτικής επεξεργασίας. Στη συγκεκριμένη διπλωματική, καλείστε να υλοποιήσετε και να βελτιστοποιήσετε την επέκταση του συστήματος σε μια από τις ακόλουθες κατευθύνσεις:

- Σε δεδομένα κειμένου. Συγκεκριμένα, για τελεστές που παίρνουν σαν είσοδο 1 αρχείο κειμένου το σύστημα πρέπει να μοντελοποιεί τις διαφορές ανάμεσα σε πολλά κείμενα καθώς και να προβλέπει την έξοδο του τελεστή για οποιοδήποτε κείμενο με το ελάχιστο σφάλμα.
- Σε τελεστές που δέχονται περισσότερες της μιας εισόδους (π.χ. Join operator). Το σύστημα θα πρέπει να τροποποιηθεί ώστε να μοντελοποιεί την επίδραση που έχουν 2 dataset εισόδου καθώς και οι ομοιότητές τους στην πρόβλεψη του αποτελέσματος του τελεστή.

Επιθυμητή και είναι η υποβολή δημοσίευσης από την παραπάνω εργασία σε σχετικό workshop.

Σχετικά Μαθήματα: Προχωρημένα θέματα βάσεων δεδομένων, Κατανεμημένα Συστήματα

Σχετική Βιβλιογραφία:

1. T. Bakogiannis, I. Giannakopoulos, D. Tsumakos and N. Koziris: Apollo: A Dataset Profiling and Operator Modeling System. In Proceedings of the 2019 ACM SIGMOD/PODS.
2. I. Giannakopoulos, D. Tsumakos and N. Koziris: A Content-Based Approach for Modeling Analytics Operators. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management.

Επικοινωνία: Δημήτριος Τσουμάκος, dtsouma@cslab.ece.ntua.gr

25 Αποδοτικός συνεχής υπολογισμός ερωτημάτων σε δυναμικά δεδομένα γράφων με τη χρήση του Timely Dataflow

Η ανάλυση σε μεγάλα δεδομένα γράφων τόσο σε στατικό όσο και δυναμικό (δλδ, ο γράφος διαρκώς μεταβάλλεται με εισαγωγές διαγραφές κόμβων και ακμών) επίπεδο έχει εξελιχθεί σε πολύ σημαντικό πεδίο έρευνας και ανάπτυξης. Ένα από τα πιο σύγχρονα εργαλεία που επιτρέπουν τέτοιους

υπολογισμούς είναι το Naiad [1], που υλοποιεί το Timely-Dataflow υπολογιστικό μοντέλο. Το μοντέλο υποστηρίζει επαναληπτικούς και συνεχείς υπολογισμούς. Δίνει τη δυνατότητα επεξεργασίας ροών και εργασιών δέσμης με ταχύτητα, χρησιμοποιώντας μια νέα προσέγγιση συντονισμού που συνδυάζει ασύγχρονη και σύγχρονη εκτέλεση. Το Differential dataflow [2] εκτελεί επαναληπτικό υπολογισμό σε ροές δεδομένων με τον υπολογισμό να υφίσταται μόνο σε απόκριση προς την αλλαγή των δεδομένων. Στη συγκεκριμένη διπλωματική, καλείστε να χρησιμοποιήσετε την open-source έκδοση σε Rust (<https://github.com/frankmcsherry/timely-dataflow> <https://github.com/frankmcsherry/differential-dataflow>) και, αφού υλοποιήσετε γνωστούς αλγορίθμους γράφων (π.χ. Triangle counting, pagerank, centralities) να συγκρίνετε τη streaming εκδοχή τους στο Timely-Dataflow με το GraphX (Spark) [3].

Σχετικά Μαθήματα: Προχωρημένα θέματα βάσεων δεδομένων, Κατανεμημένα Συστήματα

Σχετική Βιβλιογραφία:

1. Derek G. Murray, Frank McSherry, Rebecca Isaacs, Michael Isard, Paul Barham, and Martín Abadi. Naiad: A timely dataflow system. In SOSP, pages 439–455, 2013.
2. Frank McSherry, Derek Gordon Murray, Rebecca Isaacs, and Michael Isard. 2013. Differential Dataflow. In Proc. Conf. on Innovative Data Systems Research (CIDR).
3. Joseph E. Gonzalez, Reynold S. Xin, Ankur Dave, Daniel Crankshaw, Michael J. Franklin, Ion Stoica. GraphX: Graph Processing in a Distributed Dataflow Framework. In USENIX Symposium on Operating Systems Design and Implementation (OSDI 14).

Επικοινωνία: Δημήτριος Τσουμάκος, dtsouma@cslab.ece.ntua.gr

Τάσος Μπακογιάννης, abk@cslab.ece.ntua.gr

26 Μελέτη και σύγκριση πολυ-συστημάτων (polystores) εκτέλεσης αναλυτικών SQL ερωτημάτων

Το Presto [1,2] είναι μια κατανεμημένη μηχανή εκτέλεσης SQL ερωτημάτων ανοιχτού κώδικα για τη διεξαγωγή διαδραστικών αναλυτικών ερωτημάτων σε πηγές δεδομένων όλων των μεγεθών που κυμαίνονται από gigabytes έως petabytes. Το Presto επιτρέπει την αναζήτηση δεδομένων σε πολλαπλές βάσεις όπως Hive, Cassandra, MySQL, MongoDB, Elasticsearch, κλπ.. Ένα ερώτημα Presto μπορεί να συνδυάσει δεδομένα από πολλαπλές πηγές, ανήκοντας στην κατηγορία των “polystores”. Στη συγκεκριμένη διπλωματική, καλείστε να μελετήσετε τις δυνατότητες του Presto και να το συγκρίνετε με τα “συγγενικά” MuSQLE [3], Impala[4] και SparkSQL [5] σχετικά με:

- βελτιστοποίηση ερωτημάτων,
- κλιμακωσιμότητα,
- απόδοση.

Σχετικά Μαθήματα: Προχωρημένα θέματα βάσεων δεδομένων, Κατανεμημένα Συστήματα

Σχετική Βιβλιογραφία:

1. R. Sethi et al., “Presto: SQL on Everything,” 2019 IEEE 35th International Conference on Data Engineering (ICDE).

2. <https://github.com/prestosql/presto>
3. V. Giannakouris, N. Papailiou, D. Tsoumakos and N. Koziris: MuSQLE: Distributed SQL Query Execution Over Multiple Engine Environments. In Proceedings of the 2016 IEEE International Conference on Big Data (BigData 2016).
4. <https://impala.apache.org/>
5. <https://spark.apache.org/sql/>

Επικοινωνία: Δημήτριος Τσουμάκος, dtsouma@cslab.ece.ntua.gr

27 Benchmarking Αλγορίθμων Consensus σε Ethereum/Hyperledger

Η τεχνολογία blockchain, που αρχικά δημιουργήθηκε για να αποτελέσει τη βάση λειτουργίας του δικτύου Bitcoin, λειτουργεί ως ένα κοινόχρηστο δημόσιο λογιστικό βιβλίο στο οποίο εγγράφονται όλες οι επιβεβαιωμένες συναλλαγές – ένα σύνολο συναλλαγών αποτελούν ένα block και το κάθε block αναφέρεται στο προηγούμενο του δημιουργώντας μια αλυσίδα [1]. Η επιβεβαίωση των συναλλαγών και η συμφωνία για τη σειρά εκτέλεσής τους γίνεται με καταναμημένο τρόπο με χρήση αλγορίθμων consensus. Οι δύο βασικές κατηγορίες τέτοιων αλγορίθμων είναι οι lottery-based (π.χ., ο Proof-of-Work του Bitcoin) και οι voting-based (π.χ., ο Byzantine Fault Tolerance του Hyperledger) [2, 3]. Καθένας από του αλγορίθμους αυτού έχουν πλεονεκτήματα και μειονεκτήματα σε σχέση με το transaction throughput που επιτυγχάνουν, την κλιμακωσιμότητά τους, την ασφάλεια που παρέχουν, ακόμα και την ενέργεια που σπαταλούν. Στόχος της διπλωματικής είναι η μελέτη διαφορετικών αλγορίθμων consensus που χρησιμοποιούνται στις πιο δημοφιλείς υλοποιήσεις Blockchain, είτε public (Ethereum [4]) είτε permissioned (Hyperledger [5]) με τη χρήση benchmarks (Blockbench [6] ή Hyperledger Caliper [7] αντίστοιχα) ως προς τις παραπάνω ιδιότητες.

Σχετικά Μαθήματα: Καταναμημένα Συστήματα

Σχετική Βιβλιογραφία:

1. Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008): 28.
2. Bano, S., Sonnino, A., Al-Bassam, M., Azouvi, S., McCorry, P., Meiklejohn, S., Danezis, G. (2017). Consensus in the age of blockchains. arXiv preprint [arXiv:1711.03936](https://arxiv.org/abs/1711.03936).
3. Hyperledger Architecture, Volume 1
4. Ethereum
5. Hyperledger
6. Dinh, Tien Tuan Anh, et al. "Blockbench: A framework for analyzing private blockchains." Proceedings of the 2017 ACM International Conference on Management of Data. ACM, 2017.
7. Hyperledger Caliper

Επικοινωνία: Κατερίνα Δόκα, katerina@cslab.ece.ntua.gr, 210-772-1175

28 Επαληθεύσιμη κατανεμημένη αποθήκευση και επεξεργασία με χρήση Blockchain

Η διαφάνεια συναλλαγών και η έλλειψη κεντρικού ελέγχου που προσφέρει η τεχνολογία blockchain έχουν οδηγήσει στην υιοθέτησή της σε πληθώρα εφαρμογών πέρα από το νόμισμα (διαχείριση ψηφιακών περιουσιακών στοιχείων, από τίτλους ιδιοκτησίας και μετοχές μέχρι ταξιδιωτικά μίλια, δημιουργία ψηφιακών ταυτοτήτων που χρησιμοποιούνται σε ψηφιακές υπογραφές, δημιουργία επαληθεύσιμης καταγραφής για κάθε είδους δεδομένο, αρχείο ή διεργασία). Η δυνατότητα που προσφέρουν κάποια blockchains για εκτέλεση έξυπνων συμβολαίων, κώδικα δηλαδή που εκτελείται αυτόματα όταν ικανοποιηθούν ορισμένες συνθήκες, έχει δώσει ακόμα μεγαλύτερη ώθηση στη χρήση τους. Παράδειγμα εφαρμογής που έχει αναπτυχθεί στο εργαστήριο είναι μια πλατφόρμα για δημιουργία πραγματικά αποκεντρωμένης υποδομής Cloud [1,2].

Σκοπός της διπλωματικής είναι να εκμεταλλευτούμε τα πλεονεκτήματα των blockchains για υποστηρίξουμε την ασφαλή κατανεμημένη αποθήκευση αρχείων και την επαληθεύσιμη εκτέλεση κατανεμημένων προγραμμάτων, ακόμα και σε συνθήκες όπου δεν εμπιστευόμαστε τους συμμετέχοντες κόμβους. Πρακτικά, θα ενισχύσουμε υπάρχοντα δημοφιλή κατανεμημένα frameworks αποθήκευσης (HDFS [3]) και επεξεργασίας (Hadoop MapReduce [4], Spark [5]) με μηχανισμούς επαλήθευσης που θα στηρίζονται στην τεχνολογία blockchain (merkle proofs [6], truebit [7]). Έτσι θα διασφαλίσουμε ότι κανένας κακόβουλος κόμβος δε θα μπορέσει να αλλάξει τα περιεχόμενα ενός κατανεμημένου αρχείου ή το αποτέλεσμα ενός κατανεμημένου υπολογισμού.

Σχετικά Μαθήματα: Κατανεμημένα Συστήματα

Σχετική Βιβλιογραφία:

1. Katerina Doka, Tasos Bakogiannis, Ioannis Mytilinis and Georgios Goumas: CloudAgora: Democratizing the Cloud. In Proceedings of the 2nd International Conference on Blockchain (ICBC), 2019, San Diego, CA, USA.
2. Tasos Bakogiannis, Ioannis Mytilinis, Katerina Doka and Georgios Goumas: Building Ad-Hoc Clouds with CloudAgora. Accepted for publication at the 38th International Symposium on Reliable Distributed Systems (SRDS 2019), October 1-4, Lyon, France.
3. Apache Hadoop HDFS
4. Apache Hadoop MapReduce
5. Apache Spark
6. Merkle proofs explained
7. <https://truebit.io/>

Επικοινωνία: Κατερίνα Δόκα, katerina@cslab.ece.ntua.gr, 210-772-1175

Ιωάννης Μυτιλήνης, gmytil@cslab.ece.ntua.gr

29 Ανάλυση Blockchain δεδομένων για εξαγωγή πληροφορίας

Στα blockchains καταγράφονται. Το bitcoin blockchain έχει ήδη μέγεθος που φτάνει στα 250GB, ενώ το ethereum blockchain είναι 140GB και μέχρι τον Νοέμβριο του 2018 είχε πάνω από 1.000.000 ενεργά έξυπνα συμβόλαια. Στα blockchains επομένως κρύβεται πολύτιμη πληροφορία που αξίζει να συγκεντρωθεί και να αναλυθεί συστηματικά. Στην πρωτογενή μορφή τους τα δεδομένα των πιο δημοφιλών blockchains (π.χ., bitcoin, Ethereum) είναι απλές καταγραφές transactions, τα οποία μπορεί να είναι είτε απλά, π.χ., αποστολή χρημάτων από έναν λογαριασμό σε έναν άλλον, είτε πιο σύνθετα, που εμπιρεύουν bytecode από έξυπνα συμβόλαια. Ως έχουν, τα δεδομένα αυτά δεν αποκαλύπτουν άμεσα και διαισθητικά πληροφορίες για τον πραγματικό κόσμο. Ωστόσο με κατάλληλη μοντελοποίηση με χρήση θεωρίας γράφων, αποθήκευση, δεικτοδότηση και επεξεργασία των δεδομένων του blockchain με μπορούμε να εξάγουμε γνώση που σχετίζεται με τη ροή χρημάτων, να αναγνωρίσουμε μοτίβα συμπεριφοράς χρηστών και να ανιχνεύσουμε πιθανές απάτες. Κάποιες πρώτες προσπάθειες για το bitcoin blockchain έχουν ήδη γίνει: [1,2,3]. Δεδομένου του μεγάλου όγκου της πληροφορίας και της ταχύτητας με την οποία παράγεται, το σύστημα για την ανάλυση των δεδομένων του bitcoin blockchain θα πρέπει αναπτυχθεί σε κατανεμημένη και κλιμακώσιμη υποδομή (π.χ. Spark [4], Flink [5]).

Σχετικά Μαθήματα: Κατανεμημένα Συστήματα

Σχετική Βιβλιογραφία:

1. Haslhofer, Bernhard, Roman Karl, and Erwin Filtz. "Ο Bitcoin Where Art Thou? Insight into LargeScale Transaction Graphs." SEMANTiCS (Posters, Demos, SuCCESS). 2016.
2. Marcin, Sergio Ivan. "Bitcoin Live: Scalable system for detecting bitcoin network behaviors in real time." (2015).
3. <https://uncharted.software/assets/louvain-clustering-for-big-data-graph-visual-analytics.pdf>
4. Apache Spark
5. Apache Flink

Επικοινωνία: Κατερίνα Δόκα, katerina@cslab.ece.ntua.gr, 210-772-1175

30 Αυτοματοποίηση χρονοδρομολόγησης και δέσμευσης πόρων για εκτέλεση εφαρμογών μεγάλης κλίμακας σε περιβάλλοντα υπολογιστικών νεφών

Τα τελευταία χρόνια η έκρηξη που σημειώνεται στην παραγωγή δεδομένων έχει δημιουργήσει αντίστοιχες ανάγκες επεξεργασίας και αξιοποίησης αυτών. Μια πληθώρα συστημάτων επεξεργασίας δεδομένων σε μεγάλη κλίμακα έχει κάνει την εμφάνισή της – συστήματα τα οποία συχνά υλοποιούν διαφορετικά προγραμματιστικά μοντέλα και παρουσιάζουν διαφορετικά χαρακτηριστικά. Οι τελευταίες εξελίξεις οδηγούν σε ένα μοντέλο όπου η υπολογιστική και προγραμματιστική πολυπλοκότητα «κρύβονται» από τον τελικό χρήστη και οι δυνατότητες των σύγχρονων αυτών συστημάτων προσφέρονται με τη μορφή υπηρεσιών που εκτελούνται πάνω από υποδομές υπολογιστικών νεφών. Με αυτό τον τρόπο, εργαλεία επεξεργασίας μεγάλου όγκου δεδομένων και μηχανικής μάθησης γίνονται

προσιτά και σε χρήστες οι οποίοι πιθανώς δεν έχουν τις τεχνικές γνώσεις ώστε να τα διαχειριστούν – καθώς η διαχείριση γίνεται από τον πάροχο των υπηρεσιών αυτών. Από την πλευρά ενός οργανισμού που παίζει αυτό το ρόλο του παρόχου υπηρεσιών, είναι επιθυμητή η δυνατότητα βέλτιστης διαχείρισης των υπολογιστικών πόρων με διπλό στόχο. Από τη μια πλευρά την ελαχιστοποίηση του κόστους και από την άλλη της τήρησης των service-level agreements (SLAs) – διμερών συμφωνιών που συνάπτονται με τους χρήστες και αφορούν το επίπεδο των παρεχόμενων υπηρεσιών.

Σε αυτό το πλαίσιο, στο εργαστήριο έχει γίνει η προσπάθεια μοντελοποίησης εφαρμογών και συστημάτων επεξεργασίας και χρήσης των παραγόμενων μοντέλων για την αυτοματοποίηση α) της χρονοδρομολόγησης και β) της εκτίμησης των απαιτούμενων υπολογιστικών πόρων για την εκτέλεσή τους σε χρόνο που δεν παραβιάζει τα SLAs. Το ζητούμενο είναι η μελέτη του προβλήματος στους εξής άξονες:

- ανάπτυξη ‘έξυπνης’ αυτοματοποιημένης μεθόδου δειγματοληψίας για την κατασκευή μοντέλων με στόχο τη μείωση του κόστους αυτής της διαδικασίας,
- διερεύνηση χρήσης διαφορετικών μοντέλων μηχανικής μάθησης για την προσέγγιση της συμπεριφοράς των εφαρμογών και των συστημάτων με βάση τις παραμέτρους που επηρεάζουν περισσότερο την επίδοσή τους,
- αξιοποίηση πόρων όπως τα spot-vms,
- εφαρμογή των παραπάνω σε Stream Processing Systems, όπου latency και throughput θέτουν σημαντικούς περιορισμούς και τα (SLAs) κινούνται σε sub-second τάξεις μεγέθους.

Σχετικά Μαθήματα: Κατανεμημένα Συστήματα

Επικοινωνία: Νίκος Χαλβαντζής, nchalv@cslab.ece.ntua.gr

Ιωάννης Κωνσταντίνου, ikons@cslab.ece.ntua.gr

31 Συγκριτική αξιολόγηση αλγορίθμων μηχανικής μάθησης σε συστήματα επεξεργασίας δεδομένων μεγάλης κλίμακας

Σήμερα, ένας τομέας επεξεργασίας δεδομένων που παρουσιάζει μεγάλο ενδιαφέρον είναι ο τομέας της μηχανικής μάθησης. Σε αυτό το πλαίσιο, χρησιμοποιείται ένα σύνολο ιστορικών δεδομένων για την εξαγωγή συμπερασμάτων πάνω σε νέα δεδομένα (κατηγοριοποίηση, πρόβλεψη τιμών, ομαδοποίηση). Ωστόσο, καθώς ο όγκος των δεδομένων συνεχώς αυξάνεται, η διαδικασία δημιουργίας των μοντέλων που είναι ικανά να εξάγουν αυτά συμπεράσματα γίνεται όλο και πιο απαιτητική και χρονοβόρα. Για το λόγο αυτό, έχει αναπτυχθεί πληθώρα κατανεμημένων συστημάτων, εξειδικευμένα και μη σε αυτόν τον τομέα. Ένα από τα πιο διαδεδομένα κατανεμημένα συστήματα γενικού σκοπού είναι το Apache Spark, το οποίο παρέχει βιβλιοθήκες αλγορίθμων μηχανικής μάθησης (MLlib και ο απόγονος της η ML). Ωστόσο, όπως αναφέρθηκε, υπάρχει και ένα σύνολο εξειδικευμένων κατανεμημένων συστημάτων στον τομέα της μηχανικής μάθησης, όπως το Google TensorFlow. Τα διαφορετικά αυτά συστήματα παρουσιάζουν σημαντικές διαφορές μεταξύ τους, τόσο σε επίπεδο αρχιτεκτονικής όσο και σε επίπεδο υλοποίησης. Για το λόγο αυτό, κρίνεται ιδιαίτερα σκόπιμος ο εντοπισμός αυτών των στοιχειωδών διαφορών που παρουσιάζουν αυτά τα συστήματα μέσω πειραματικής ανάλυσης. Η συγκριτική αυτή μελέτη μπορεί να φανεί ιδιαίτερα χρήσιμη τόσο στους προγραμματιστές που αναπτύσσουν το εκάστοτε σύστημα, για την περαιτέρω ανάπτυξη μηχανισμών που είναι ήδη ανεπτυγμένη σε κάποιο άλλο, όσο και στους χρήστες αυτών, παρουσιάζοντας τον βέλτιστο τρόπο ρύθμισης και χρήσης αυτών.

Στο εργαστήριο έχει γίνει συγκριτική μελέτη σε βάθος για την επίδοση βασικών αλγορίθμων μηχανικής μάθησης που χρησιμοποιούν τον αλγόριθμο απότομης καθόδου (Gradient Descent) για την εκπαίδευση των μοντέλων στα συστήματα Apache Spark και Google TensorFlow. Αντίστοιχες μελέτες μπορούν να υλοποιηθούν προς διάφορες κατευθύνσεις.

Σχετικά Μαθήματα: Κατανεμημένα Συστήματα

Επικοινωνία: Νίκος Προβατάς, nprov@cslab.ece.ntua.gr

Ιωάννης Κωνσταντίνου, ikons@cslab.ece.ntua.gr

32 Συγκριτική αξιολόγηση αλγορίθμων μηχανικής μάθησης σε συστήματα επεξεργασίας δεδομένων μεγάλης κλίμακας

Σήμερα, ένας τομέας επεξεργασίας δεδομένων που παρουσιάζει μεγάλο ενδιαφέρον είναι ο τομέας της μηχανικής μάθησης. Σε αυτό το πλαίσιο, χρησιμοποιείται ένα σύνολο ιστορικών δεδομένων για την εξαγωγή συμπερασμάτων πάνω σε νέα δεδομένα (κατηγοριοποίηση, πρόβλεψη τιμών, ομαδοποίηση). Ωστόσο, καθώς ο όγκος των δεδομένων συνεχώς αυξάνεται, η διαδικασία δημιουργίας των μοντέλων που είναι ικανά να εξάγουν αυτά συμπεράσματα γίνεται όλο και πιο απαιτητική και χρονοβόρα. Για το λόγο αυτό, έχει αναπτυχθεί πληθώρα κατανεμημένων συστημάτων, εξειδικευμένα και μη σε αυτόν τον τομέα. Ένα από τα πιο διαδεδομένα κατανεμημένα συστήματα γενικού σκοπού είναι το Apache Spark, το οποίο παρέχει βιβλιοθήκες αλγορίθμων μηχανικής μάθησης (MLlib και ο απόγονος της η ML). Ωστόσο, όπως αναφέρθηκε, υπάρχει και ένα σύνολο εξειδικευμένων κατανεμημένων συστημάτων στον τομέα της μηχανικής μάθησης, όπως το Google TensorFlow. Τα διαφορετικά αυτά συστήματα παρουσιάζουν σημαντικές διαφορές μεταξύ τους, τόσο σε επίπεδο αρχιτεκτονικής όσο και σε επίπεδο υλοποίησης. Για το λόγο αυτό, κρίνεται ιδιαίτερα σκόπιμος ο εντοπισμός αυτών των στοιχειωδών διαφορών που παρουσιάζουν αυτά τα συστήματα μέσω πειραματικής ανάλυσης. Η συγκριτική αυτή μελέτη μπορεί να φανεί ιδιαίτερα χρήσιμη τόσο στους προγραμματιστές που αναπτύσσουν το εκάστοτε σύστημα, για την περαιτέρω ανάπτυξη μηχανισμών που είναι ήδη ανεπτυγμένη σε κάποιο άλλο, όσο και στους χρήστες αυτών, παρουσιάζοντας τον βέλτιστο τρόπο ρύθμισης και χρήσης αυτών. Στο εργαστήριο έχει γίνει συγκριτική μελέτη σε βάθος για την επίδοση βασικών αλγορίθμων μηχανικής μάθησης που χρησιμοποιούν τον αλγόριθμο απότομης καθόδου (Gradient Descent) για την εκπαίδευση των μοντέλων στα συστήματα Apache Spark και Google TensorFlow. Αντίστοιχες μελέτες μπορούν να υλοποιηθούν προς διάφορες κατευθύνσεις.

Σχετικά Μαθήματα: Κατανεμημένα Συστήματα

Επικοινωνία: Νίκος Προβατάς, nprov@cslab.ece.ntua.gr

Ιωάννης Κωνσταντίνου, ikons@cslab.ece.ntua.gr

33 Εφαρμογές τεχνικών mechanism design (υποκλάδος της αλγοριθμικής θεωρίας παιγνίων) και deep learning για αποδοτικές online δημοπρασίες πόρων μεταξύ χρηστών σε περιβάλλοντα cloud

Στα σύγχρονα μεγάλα υπολογιστικά νέφη η δυναμική εκχώρηση πόρων σε χρήστες ή tasks χρηστών ανάλογα με την ανάγκη του χρήστη και τη διαθεσιμότητα πόρων στο δίκτυο μια δεδομένη στιγμή

εφάπτεται στο πρόβλημα του αποδοτικού resource allocation. Με την άνοδο του machine learning και την ανάπτυξη περισσότερων non-critical ή easy scalable jobs που μπορεί να τρέξει ο χρήστης σε μια cloud υπηρεσία και για να αντιμετωπιστεί το πρόβλημα των πόρων που μένουν αχρησιμοποίητοι στο AWS, η Amazon δημιούργησε την υπηρεσία Spot Instances όπου τιμές των Vms διαμορφώνονται κατά το δοκούν περιοδικά, ανάλογα με τη ζήτηση. Αυτό εμπεριέχει τον κίνδυνο κάποιος χρήστης να απωλέσει μεγάλο αριθμό των πόρων του χωρίς προειδοποίηση. Στο εργαστήριο αναπτύξαμε έναν decision component, ονόματι Game Master, που εφαρμόζεται στο Spot Instances και διενεργεί online δημοπρασίες περιοδικά με στόχο οι πόροι ενός χρήστη να αυξομειώνονται ελαστικά. Για να διασφαλίσουμε την εγκυρότητα, την τιμιότητα και τη φιλαλήθεια των παραπάνω δημοπρασιών χρησιμοποιούμε τεχνικές δανεισμένες από το mechanism design- ένας υποκλάδος της αλγοριθμικής θεωρίας παιγνίων που στόχο έχει την κατασκευή μηχανισμών που οδηγούν τους παίκτες ενός παιγνίου να παρουσιάζουν φιλαλήθη συμπεριφορά κατά τη συμμετοχή τους στο παίγνιο. Επίσης χρησιμοποιούμε μια προσέγγιση ενός deep learning min max νευρωνικού δικτύου. Καταφέρνουμε να πετύχουμε οφέλη για σωρεία διαφορετικών περιπτώσεων όπως το να πετύχουμε μεγάλα κέρδη για τον cloud vendor ή μέγιστη κοινωνική ωφέλεια για τους χρήστες (οι χρήστες μένουν στην πλειονότητά τους ευχαριστημένοι) Οι προσομοιώσεις πάνω στις οποίες τεστάρουμε τον component αφορούν στατιστικά στοιχεία από το Google Trace. Στόχος μας είναι να η δημιουργία ενός actual cluster με διαφορετικούς χρήστες, με διαφορετικά non-critical applications που συμμετέχουν στις άνωθι δημοπρασίες του Game Master περιοδικά, προσθαφαιρούνται πόροι τους και να δείξουμε πως τα κριτήρια που εμφανώς πληρούνται στις προσομοιώσεις, πληρούνται και σε actual executions environments. Παράλληλα η χρησιμοποίηση του Game Master σε περιβάλλοντα ετερογενών αρχιτεκτονικών θα μπορούσε να εξεταστεί σαν υποψήφιο θέμα.

Σχετική Βιβλιογραφία:

1. <https://aws.amazon.com/ec2/spot/>
2. L. Zhang, Z. Li and C. Wu, "Dynamic resource provisioning in cloud computing: A randomized auction approach," IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, Toronto, ON, 2014, pp. 433-441.
3. W. Shi, L. Zhang, C. Wu, Z. Li and F. C. M. Lau, "An Online Auction Framework for Dynamic Resource Provisioning in Cloud Computing," in IEEE/ACM Transactions on Networking, vol. 24, no. 4, pp. 2060-2073, Aug. 2016.
4. Dütting, Paul, Zhe Feng, Harikrishna Narasimhan, David C. Parkes and Sai Srivatsa Ravindranath. "Optimal Auctions through Deep Learning." ICML (2017).

Επικοινωνία: Κωνσταντίνος Μπιτσάκος, kbitsak@cslab.ece.ntua.gr

Ιωάννης Κωνσταντίνου, ikons@cslab.ece.ntua.gr

34 Κατασκευή machine learning agent που θα εξάγει βέλτιστα δυναμικές πληροφορίες που μέχρι τώρα μπορούσαν να εξαχθούν μόνο κατά την εκτέλεση ενός κομματιού κώδικα, κατά τη στατική LLVM ανάλυση του κώδικα (έμφαση σε OpenCL kernels για περιβάλλοντα ετερογενών αρχιτεκτονικών)

Τα τελευταία χρόνια η εκτέλεση προγραμμάτων σε ετερογενείς αρχιτεκτονικές (CPU, GPU, TPU, FPGA), δεδομένης της ύπαρξης tasks που η εκτέλεση σε κάποια από τις παραπάνω μονάδες ευνοείται είτε

χρονικά είτε σε επίπεδα power consumption, έχει γνωρίσει τεράστια άνθηση. Παραλληλοποιήσιμοι αλγόριθμοι πχ ευνοούνται από την εκτέλεση σε γραφικές μονάδες επεξεργασίας ενώ αλγόριθμοι δυναμικού προγραμματισμού εξαιτίας των πολλών memory transactions ευνοούνται από την εκτέλεση σε παραδοσιακά συστήματα CPU. Γίνεται εμφανές πως η κατασκευή ενός scheduler, που δεδομένου του source code ενός OpenCL kernel που θα παίρνει σαν είσοδο, θα βρίσκει το βέλτιστο placement του kernel σε κάποιο από τα παραπάνω devices, είναι επιτακτική. Πολλές προσπάθειες έχουν γίνει πάνω στο πρόβλημα, στην πλειονότητα των οποίων ο ml agent που παίρνει τις αποφάσεις για τον scheduler, ταϊζεται με static features από τον αρχικό κώδικα, bytes εισόδου για τον kernel, υποψήφια devices και άλλα dynamic features. Το πρόβλημα εντοπίζεται πως όσον αφορά τον αριθμό των actual εντολών που εκτελούνται δεν υπάρχει κανένας τρόπος να γνωρίζει κάποιος εκ των προτέρων παρά μόνο την ώρα της εκτέλεσης. Για αυτό static features που γίνονται extract κατά τη μετάφραση του αρχικού OpenCL κώδικα σε LLVM IR language, χρησιμοποιούνται. Καταλαβαίνει κάποιος πως η πληροφορία που συλλέγεται από αυτά τα static features χρήζει μεγάλης βελτίωσης. Πχ η στατική ανάλυση μας λέει πως βλέπει 3 memory reads. Δεν είναι σε θέση να μας πει εάν αυτά τα memory reads βρίσκονται μέσα σε ένα for loop που κάνει break, το πόσα memory reads θα εκτελεστούν εν τέλει δεδομένης μιας εισόδου. Αυτή τη στιγμή στο εργαστήριο δουλεύουμε πάνω στην κατασκευή ενός ml agent ο οποίος τρέχοντας micro-benchmarking κατά το runtime, θα μπορεί να εξάγει σχέσεις που θα προβλέπουν τον actual αριθμό των εντολών που θα εκτελεστούν κατά τη διάρκεια εκτέλεσης ενός προγράμματος. Ο agent αυτός μπορεί να επεκταθεί στην εξαγωγή διαφόρων χρήσιμων πληροφοριών από την αναπαράσταση κώδικα χρήστη σε γλώσσα μηχανής, που η στατική ανάλυση αδυνατεί να μας δώσει. Για την κατασκευή του agent θα χρησιμοποιηθούν machine learning regression τεχνικές (polynomial, elasticNet, Xgboost, προσέγγιση με νευρωνικά δίκτυα) που θα συμμετέχουν σε έναν voting ensemble system για την εύρεση της καλύτερης προσέγγισης αναπαράστασης κώδικα.

Σχετική Βιβλιογραφία:

1. C. Ferdinand, "Worst case execution time prediction by static program analysis," 18th International Parallel and Distributed Processing Symposium, 2004. Proceedings., Santa Fe, NM, USA, 2004.
2. L. T. Yang, Xiaosong Ma and F. Mueller, "Cross-Platform Performance Prediction of Parallel Applications Using Partial Execution," SC '05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing, Seattle, WA, USA, 2005, pp. 40-40.
3. L. David and I. Puaut, "Static determination of probabilistic execution times," Proceedings. 16th Euromicro Conference on Real-Time Systems, 2004. ECRTS 2004., Catania, Italy, 2004, pp. 223-230.
4. Chun, Byung-Gon, et al. "Mantis: Predicting system performance through program analysis and modeling." arXiv preprint arXiv:1010.0019 (2010).

Επικοινωνία: Κωνσταντίνος Μπιτσάκος, kbitsak@cslab.ece.ntua.gr

Κατερίνα Δόκα, katerina@cslab.ece.ntua.gr, 210-772-1175