ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
http://www.cslab.ece.ntua.gr

Διπλωματικές Εργασίες
Ακαδημαϊκό έτος 2021-22

Ι. Παράλληλα Συστήματα

1 Πολλαπλασιασμός αραιού πίνακα με διάνυσμα (SpMV)

Ο υπολογιστικός πυρήνας του πολλαπλασιασμού αραιού πίνακα με διάνυσμα (SpMV) χρησιμοποιείται ευρέως σε παράλληλες εφαρμογές μεγάλης κλίμακας. Ωστόσο, λόγω της αλγοριθμικής του φύσης, δεν αξιοποιεί επαρκώς την υπολογιστική ισχύ των σύγχρονων επεξεργαστών. Οι παρακάτω εργασίας εστιάζουν στην βελτιστοποίηση του με σε διαφορετικές αρχιτεκτονικές με τη χρήση των κατάλληλων προγραμματιστικών μοντέλων.

1.1 Βελτιστοποίηση του υπολογιστικού πυρήνα πολλαπλασιασμού αραιού πίνακα με διάνυσμα (SpMV) σε FPGAs

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 ή άλλων προγραμματιστικών μοντέλων υψηλού επιπέδου, καθώς και 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: Παναγιώτης Μπάκος, pmpakos@cslab.ece.ntua.gr
Νικέλα Παπαδοπούλου, nikela@cslab.ece.ntua.gr, 210-772-2279
Γεώργιος Γκούμας, goumas@cslab.ece.ntua.gr, 210-772-2402

2 Συνελικτικά Νευρωνικά Δίκτυα (CNNs))

2.1 Μελέτη επίδοσης και βελτιστοποίηση των συνελικτικών νευρωνικών δικτύων σε σύγχρονες αρχιτεκτονικές

Ένα από τα ιδιαίτερα χαρακτηριστικά των συνελικτικών νευρωνικών δικτύων είναι ότι αποτελούνται από 7 εμφωλευμένα loops, στα οποία μπορεί κανείς να εφαρμόσει πολλές από τις συνήθεις τεχνικές βελτιστοποίησης loops (permutation, tiling, κ.λ.π.), ωστόσο ο χώρος των δυνατών βελτιστοποιήσεων και παραμέτρων που προκύπτει είναι μεγάλος. Ταυτόχρονα, η επιλογή των κατάλληλων βελτιστοποιήσεων είναι καθοριστική για την επίδοσή τους, αφού αποτελούν το μόνο τρόπο αποδοτικής αξιοποίησης του συστήματος μνήμης κάθε αρχιτεκτονικής. Πρόσφατα, στη βιβλιογραφία, έχουν προταθεί εναλλακτικές προσεγγίσεις και συστήματα λογισμικού για την αποδοτική εξερεύνηση αυτού του χώρου των βελτιστοποιήσεων και παραμέτρων.

Στην παρούσα εργασία θα μελετήσουμε την επίδοση των συνελικτικών νευρωνικών δικτύων σε σύγχρονες αρχιτεκτονικές (CPUs, GPUs), τις διάφορες τεχνικές βελτιστοποίησης, και τις τεχνικές αναζήτησης των κατάλληλων βελτιστοποιήσεων και παραμέτρων, επεκτείνοντάς τες κατάλληλα.

Σχετική Βιβλιογραφία:

- Li, R., Xu, Y., Sukumaran-Rajam, A., Rountev, A., Sadayappan, P. (2021, April). Analytical characterization and design space exploration for optimization of CNNs. In Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (pp. 928-942).
- Zheng, L., Jia, C., Sun, M., Wu, Z., Yu, C. H., Haj-Ali, A., ... Stoica, I. (2020). Ansor: Generating high-performance tensor programs for deep learning. In 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20) (pp. 863-879).

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

3 Αποδοτική χρήση και προγραμματισμός GPGPU

Οι σύγχρονες μονάδες επεξεργασίας γραφικών ή κάρτες γραφικών (GPUs) έχουν εξελιχθεί από το να είναι χρήσιμες μόνο για συγκεκριμένες λειτουργίες, σε ισχυρά εργαλεία γενικής χρήσης (GPGPUs) ικανά να υποστηρίξουν μια πολύ μεγαλύτερη ποικιλία προβλημάτων, παρέχοντας μια ταχύτερη και πιο ενεργειακά αποδοτική εναλλακτική λύση σε σχέση με τους κανονικούς επεξεργαστές (CPUs).

3.1 Υλοποιηση multi-GPU L3 BLAS βιβλιοθηκης σε OpenCL

Οι BLAS (Basic Linear algebra subprograms) αποτελούν μια συγκεκριμένη ομάδα απο ρουτίνες γραμικής άλγεβρας, που χρησιμοποιούνται σε πολλά επιστημονικά προβληματά. Συγκεκριμενα οι Level-3 BLAS, που αφορούν πράξεις μεταξύ πινάκων, ειναι υπολογιστικά απαιτητικές και συνήθως καθορίζουν την συνολική επίδοση των εφαρμογών που τις χρησιμοποιούν, με αποτέλεσμα η παραλληλοποίησή τους να είναι ένα πολύ σημαντικό και όχι απλό πρόβλημα. Συγκεκριμενα στις GPUs, λόγω της ξεχωριστής μνήμης τους, το πρόβλημα γίνεται αρκετά πιο περίπλοκο, καθώς μια ολοκληρωμένη υλοποίηση (διαφανής για το χρήστη) πρέπει να περιλαμβάνει και τις μεταφορές δεδομένων, οι οποίες μπορούν να επικαλυφθούν με τους υπολογισμούς. Οι ευρέως χρησιμοποιούμενες multi-GPU Level-3 BLAS χρησιμοποιούν το μοντέλο CUDA. Αυτό καθιστά αδύνατη τη χρήση τους σε συστήματα άλλων αρχιτεκτονικών επιταχυντών, καθώς και σε ετερογενή συστήματα. Σκοπός αυτής της διπλωματικής ειναι:

- Η εξερεύνηση και δοκιμή βιβλιοθηκών για single GPU Level-3 BLAS με OpenCl backend.
- Η ενσωμάτωση κάποιας τέτοιας βιβλιοθήκης ως backend στο CoCoPeLia αντί του αντίστοιχου CUDA backend.

Σχετικά Μαθήματα: Συστήματα Παράλληλης Επεξεργασίας.

Σχετική Βιβλιογραφία:

1. https://docs.nvidia.com/cuda/
2. https://www.khronos.org/opencl/
3. ”BLASX: A High Performance Level-3 BLAS Library for Heterogeneous Multi-GPU Computing”
4. ”CoCoPeLia: Communication-Computation Overlap Prediction for Efficient Linear Algebra on GPUs”

Επικοινωνία: Αναστασιάδης Πέτρος, panastas@cslab.ece.ntua.gr

4 Πρόβλεψη επίδοσης παράλληλων εφαρμογών

4.1 Μοντελοποίηση επίδοσης παράλληλων εφαρμογών

Η υλοποίηση και βελτιστοποίηση παράλληλων εφαρμογών σε μοντέρνες αρχιτεκτονικές υψηλής επίδοσης είναι ιδιαίτερα κοστοβόρα και απαιτητική. Δεδομένου ότι το προς παραλληλοποίηση πρόβλημα μπορεί να έχει εναλλακτικούς αλγορίθμους και κάθε αλγόριθμος διαφορετικές προσεγγίσεις παραλληλοποίησης, η λογική “trial and error”, δηλαδή η υλοποίηση όλων των διαφορετικών εκδόσεων και η πειραματική αξιολόγηση της επίδοσής τους δεν είναι ο πλέον ενδεδειγμένος τρόπος. Στόχος της συγκεκριμένης διπλωματικής εργασίας είναι η κατάστρωση μοντέλων πρόβλεψης επίδοσης παράλληλων προγραμμάτων χωρίς να είναι αναγκαία η υλοποίησή τους. Στο πλαίσιο της εργασίας ο φοιτητής θα έχει στη διάθεσή του σειριακές υλοποιήσεις των εφαρμογών υπό εξέταση τις οποίες θα αναλύσει και στη συνέχεια θα καταστρώσει μοντέλα πρόβλεψης επίδοσης υποστηριζόμενα από μετροπρογράμματα (benchmarks). Ο έλεγχος των μοντέλων θα γίνει με σύγκριση των πραγματικών παράλληλων υλοποιήσεων που και αυτές θα είναι μέρος της διπλωματικής εργασίας.

Σχετικά Μαθήματα: Συστήματα Παράλληλης Επεξεργασίας

Επικοινωνία: Γεώργιος Γκούμας, nikela@cslab.ece.ntua.gr, 210-772-2402
Νικέλα Παπαδοπούλου, nikela@cslab.ece.ntua.gr, 210-772-2495

4.2 Τεχνικές προβολής της επίδοσης παράλληλων εφαρμογών σε υπολογιστικά συστήματα μεγάλης κλίμακας

Η 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 ή συστήματα με διαφορετικά αρχιτεκτονικά χαρακτηριστικά 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. Στη βιβλιογραφία έχουν προταθεί τεχνικές για την κατάστρωση μοντέλων προβολής τόσο της επίδοσης, σε όρους χρόνου εκτέλεσης, όσο και των χαρακτηριστικών που καθορίζουν την επίδοση μιας εφαρμογής (π.χ. FLOPs, bytes, memory footprint), ως συναρτήσεων του αριθμού των πυρήνων/επεξεργαστών και του μεγέθους της εισόδου των εφαρμογών. Στην παρούσα διπλωματική θα μελετήσουμε την επέκταση αυτών των τεχνικών σε τρεις κατευθύνσεις: α) στην μοντελοποίηση χαρακτηριστικών εφαρμογών που η είσοδός τους είναι πολυπαραμετρική (π.χ. γράφοι αντί πινάκων), β) στη μοντελοποίηση του χρόνου εκτέλεσης ως συνάρτηση των παραπάνω χαρακτηριστικών, και γ) στην παραμετροποίηση αυτών των μοντέλων για να καταστεί εφικτή η μεταφορά τους από ένα σύστημα σε ένα άλλο.

Σχετική Βιβλιογραφία:

- Calotoiu, A., Beckinsale, D., Earl, C. W., Hoefler, T., Karlin, I., Schulz, M., Wolf, F. (2016, September). Fast multi-parameter performance modeling. In 2016 IEEE International Conference on Cluster Computing (CLUSTER) (pp. 172-181). IEEE.
- Ritter, M., Calotoiu, A., Rinke, S., Reimann, T., Hoefler, T., Wolf, F. (2020, May). Learning cost-effective sampling strategies for empirical performance modeling. In 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS) (pp. 884-895). IEEE.

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

5 Δρομολόγηση Εφαρμογών και Διαχείριση Πόρων σε Υπολογιστικά Κέντρα

Οι υπολογιστικές υποδομές των Υπολογιστικών Κέντρων (Datacenters) χρησιμοποιούνται για την ταυτόχρονη εκτέλεση εφαρμογών. Η κατάλληλη χρονοδρομολόγηση και η διαχείριση των κοινόχρηστων πόρων του συστήματων αποτελούν καθοριστικούς παράγοντες για την αποτελεσματική χρήση των υπολογιστικών πόρων και την εξοικονόμηση χρόνου και ενέργειας.

5.1 Διαχείριση Πόρων σε Συστήματα Μεγάλης Κλίμακας

Καθώς η εκτέλεση πολλών τύπων υπηρεσιών μεταφέρεται σε συστήματα μεγάλης κλίμακας, η πρόκληση της διατήρησης υψηλής ποιότητας υπηρεσίας συνεχώς μεγαλώνει. Η απουσία αποδοτικών λύσεων διαμοιρασμού των κοινόχρηστων πόρων οδηγεί τους Cloud Service Providers στην απομόνωση ολόκληρων servers για την εκτέλεση εφαρμογών με αυστηρούς περιορισμούς για την επίδοση τους.

Αυτό όμως οδηγεί στην υποχρησιμοποίηση αυτών των πόρων και την αύξηση του λειτουργικού κόστους. Για την αντιμετώπιση των ζητημάτων αυτών προτείνονται τεχνικές διαχείρισης των κοινόχρηστων πόρων (Last Level Cache - Intel CMT CAT, Memory Bandwidth, Core isolation), τεχνικές χαρακτηρισμού των εφαρμογών ως προς τους κρίσιμους πόρους με σκοπό τη συνεκτέλεση εφαρμογών με συμπληρωματικές απαιτήσεις για πόρους, και τεχνικές εντοπισμού μειωμένης επίδοσης κατά το χρόνο εκτέλεσης.

5.1.1 Διαχείριση πόρων σε Kubernetes clusters με χρήση resource managers της Intel

Η χρήση containers και του Kubernetes ως πλατφόρμας διαχείρισης τους φαίνεται να επικρατεί στη βιομηχανία τα τελευταία χρόνια έναντι της χρήσης VMs, επομένως η μελέτη των επιλογών που προσφέρονται για την αποδοτική εκτέλεση πολλών containers σε ένα υπολογιστικό κόμβο είναι αναμφισβήτητα η επόμενη πρόκληση. Η Intel αναπτύσσει ήδη Resource Managers (Intel Telemetry Aware Scheduling, CRI Resource Manager, Workload Collocation Agent, Platform Resource Manager, CPU Manager for Kubernetes) προς αυτή την κατεύθυνση.
Σχετική Βιβλιογραφία:

1. Intel Resource Director Technology
2. Intel Telemetry Aware Scheduling

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

5.1.2 Διαχείριση πόρων σε serverless πλατφόρμες

Το υπολογιστικό μοντέλο του serverless computing έχει κερδίσει τα τελευταία χρόνια το έντονο ενδιαφέρον τόσο της ερευνητικής κοινότητας όσο και της βιομηχανίας, επειδή οι υπολογιστικοί πόροι χρησιμοποιούνται μόνο όταν υπάρχει ανάγκη από τις εφαρμογές, ελαχιστοποιώντας με αυτό τον τρόπο τη κατανάλωση πόρων από άεργες εφαρμογές. Συνεπώς, αυτό το υπολογιστικό μοντέλο προσφέρει σημαντικά οφέλη τόσο για τους χρήστες, οι οποίοι χρεώνονται για μόνο για τους πόρους που χρησιμοποιούν, όσο και για τους παρόχους υπηρεσιών, οι οποίοι είναι σε θέση να κάνουν καλύτερη διαχείριση των διαθέσιμων υπολογιστικών τους πόρων. Σκοπός της παρούσας διπλωματικής είναι η εξοικείωση με πλατφόρμες για serverless computing και η βελτιστοποίησή τους με την χρήση τεχνικών διαχείρισης πόρων προς την επίτευξη διαφόρων στόχων, όπως την ελαχιστοποίηση των αργών αρχικοποιήσεων (cold startups) και την αύξηση της χρησιμοποίησης των υπολογιστικών πόρων χωρίς να επηρρεάζεται η επίδοση των εφαρμογών.
Σχετική Βιβλιογραφία:

1. vHive: Open-Source Framework for Serverless Experimentation

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

5.2 Βέλτιστη αξιοποίηση υπολογιστικών πόρων σε συστήματα μεγάλης κλίμακας

Σε συστήματα μεγάλης κλίμακας υψηλών υπολογιστικών επιδόσεων, οι αλγόριθμοι δρομολόγησης εργασιών, όπως ο Back-Filling, για να λάβουν αποφάσεις, αξιοποιούν την πληροφορία που τους παρέχουν οι χρήστες, οι οποίοι, καθώς υποβάλουν την εργασία τους, αιτούνται τους απαραίτητους υπολογιστικούς πόρους (κόμβους, πυρήνες, μνήμη, επιταχυντές) και παρέχουν μια εκτίμηση για το χρόνο

ολοκλήρωσης των εργασιών τους. Όσο πιο ακριβής είναι αυτή η πληροφορία, τόσο καλύτερη είναι η αξιοποίηση του συστήματος (throughput) και η ικανοποίηση των χρηστών (χαμηλοί χρόνοι αναμονής). Ωστόσο, καθώς οι εφαρμογές που εκτελούνται συχνά περιλαμβάνουν χιλιάδες γραμμές κώδικα και χρησιμοποιούν αρκετές επιπλέον βιβλιοθήκες και υπολογιστικά πακέτα, οι χρήστες δεν είναι πάντα σε θέση να εκτιμήσουν σωστά την επίδοση της εφαρμογής τους και τον αναμενόμενο χρόνο εκτέλεσής της. Έτσι, υποβάλουν εκτιμήσεις που είναι ανακριβείς ως προς τους ζητούμενους πόρους και καταλήγουν σε σπατάλη πόρων (π.χ. ο χρήστης θα μπορούσε να είχε λάβει αντίστοιχο χρόνο εκτέλεσης με λιγότερους υπολογιστικούς πόρους). Σε σχέση με τους χρόνους εκτέλεσης, οι χρήστες κατά κανόνα υπερεκτιμούν τον αναμενόμενο χρόνο εκτέλεσης της εφαρμογής τους.

Στην παρούσα διπλωματική, θα μελετήσουμε την επίδραση των εκτιμήσεων των χρηστών στην επίδοση του συστήματος και θα επεκτείνουμε υπάρχοντες αλγορίθμους χρονοδρομολόγησης για συστήματα μεγάλης κλίμακας με δυνατότητες διάδρασης με το χρήστη, για την εκτίμηση και επιλογή των κατάλληλων πόρων σε σχέση με την εργασία του χρήστη, με στόχο τη βέλτιστη αξιοποίηση των πόρων του συστήματος και τη μεγιστοποίηση της απόδοσης του συστήματος.

Επικοινωνία: Νικέλα Παπαδοπούλου, nikela@cslab.ece.ntua.gr
Γεώργιος Γκούμας, goumas@cslab.ece.ntua.gr, 210-772-2402

5.3 Χρονοδρομολόγηση εφαρμογών σε υπολογιστικά συστήματα υψηλής επίδοσης

Τα υπολογιστικά συστήματα υψηλής επίδοσης (High Performance Computing clusters -HPC clusters) 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. Το βασικό λογισμικό που συνθέτει μια τέτοια υπολογιστική υποδομή, ονομάζεται διαχειριστής πόρων (resource manager) και περιλαμβάνει έναν χρονοδρομολογητή εργασιών (job scheduler). Ο διαχειριστής πόρων αναλαμβάνει να διαμοιράσει τους υπολογιστικούς πόρους στις αντίστοιχες εργασίες. Ο χρονοδρομολογητής εργασιών επικοινωνεί με τον διαχειριστή πόρων 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.

5.3.1 Μελέτη και αξιολόγηση αλγορίθμων χρονοδρομολόγησης MPI εργασιών σε πραγματικό περιβάλλον διαχειριστή πόρων

Ο διαχειριστής πόρων (resource manager) αυτός αναλαμβάνει να εκτελέσει τις διάφορες εργασίες, διαμοιράζοντας τους υπολογιστικούς πόρους κατάλληλα. Ο job scheduler (υποτμήμα 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. Οι αλγόριθμοι χρονοδρομολόγησης -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 τη ρυθμαπόδοση (throughput) του συστήματος, την κατανάλωση ενέργειας και κόστους. Μελέτες δείχνουν πως το co-scheduling, 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. Σκοπός της διπλωματικής αποτελεί η πειραματική μελέτη και αξιολόγηση αλγορίθμων χρονοδρομολόγησης με χρήση υπαρχόντων benchmarks σε πραγματικό περιβάλλον διαχειριστή πόρων. 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/Slurm_Workload_Manager`
3. `https://en.wikipedia.org/wiki/Supercomputer`
4. `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

5.3.2 Ανάπτυξη αλγορίθμων χρονοδρομολόγησης MPI εργασιών σε προσομοιωτή διαχειριστή πόρων (SLURM simulator)

Υπάρχουν διάφοροι αλγόριθμοι στην κατηγορία των space-sharing αλγορίθμων χρονοδρομολόγησης σε ένα σύστημα υψηλής επίδοσης (HPC), όπως ο First Come First Served (FCFS), ο Shortest Job First (SJF), ο Longest Job First (LJF), ο Backfilling κ.α. Σκοπός της διπλωματικής είναι η ανάπτυξη αλγορίθμων χρονοδρομολόγησης MPI εργασιών σε προσομοιωτή διαχειριστή πόρων που να αποσκοπούν στην καλύτερη εκμετάλλευση των πόρων του συστήματος (resource management) σε συνδυασμό με τη μίμηση ενός πραγματικού HPC Cluster κι άρα πιο έγκυρη αξιολόγηση των αλγορίθμων αυτών στον χρονοδρομολογητή εργασιών (job scheduler).

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://en.wikipedia.org/wiki/Slurm_Workload_Manager`
4. `https://github.com/ubccr-slurm-simulator/slurm_simulator`

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

6 Ανάλυση δεδομένων υπερυπολογιστικών συστημάτων

Οι διαχειριστές των σύγχρονων υπερυπολογιστικών συστημάτων έχουν τη δυνατότητα συλλογής σημαντικής πληροφορίας για τη συμπεριφορά των εφαρμογών που εκτελούνται στο σύστημα, τα αιτήματα των χρηστών και την ποιότητα υπηρεσίας που λαμβάνουν, αλλά και τη συνολική κατάσταση του συστήματος. Η συλλογή ιστορικών δεδομένων του συστήματος δίνει δυνατότητες για τη βελτίωση της χρήσης του συστήματος, αφού η επεξεργασία της μπορεί να συμβάλλει στη μείωση του χρόνου αναμονής από την πλευρά των χρηστών, στη βελτίωση των πολιτικών δέσμευσης πόρων και χρονοδρομολόγησης των εργασιών, αλλά και στην ανθεκτικότητα του συστήματος σε σφάλματα. Η παρακάτω εργασία εστιάζει στην ανθεκτικότητα του συστήματος σε σφάλματα.

6.1 Υλοποίηση συστήματος πρόβλεψης σφαλμάτων σε συστήματα υψηλής επίδοσης με τεχνικές μηχανικής μάθησης

Οι σημερινοί υπερυπολογιστές αντιμετωπίζουν συχνά σφάλματα σε καθημερινή βάση. Παρόλο που υπάρχουν πολλές προσεγγίσεις ανάκτησης (recovery), όπως το checkpoint/restart, κατά την ανάκτηση των διάφορων components από σφάλματα, χάνεται σημαντική υπολογιστική ισχύς. Τα νέα συστήματα κλίμακας exascale προβλέπεται ότι αντιμετωπίζουν ακόμα πιο ψηλά ποσοστά σφαλμάτων, λόγω του αυξημένου πλήθους των components που τα συνθέτουν. Επομένως, είναι σημαντικό να υπάρχουν καλώς ορισμένοι δείκτες σφαλμάτων (failure indicators). Τα αρχεία καταγραφής του συστήματος (logs) αποτελούνται από κείμενο που κρύβει πληροφορίες για την εκάστοτε "υγεία" του συστήματος. Συνεπώς, η αποτελεσματική πρόβλεψη σφαλμάτων του συστήματος μέσω των logs θα μπορούσε να επιτρέψει προληπτικούς μηχανισμούς ανάκτησης και άρα αύξηση της αξιοπιστίας (reliability). Στην παρούσα διπλωματική, θα μελετήσουμε την αξιοποίηση των logs και συγκεκριμένα του κειμένου που αυτά περιέχουν, θα αναλύσουμε τα κείμενα αυτά και θα αναπτύξουμε ένα σύστημα πρόβλεψης σφαλμάτων σε κόμβους του υπερυπολογιστή με τη χρήση Transformers. Για την χρήση των Transformers, η υλοποίηση θα γίνει τόσο χρησιμοποιώντας κάποιο pre-trained μοντέλο όσο και φτιάχνοντας το δικό μας Transformers μοντέλο.

Σχετική Βιβλιογραφία:

- Das, A., Mueller, F., Siegel, C., Vishnu, A. (2018, June). Desh: deep learning for system health prediction of lead times to failure in hpc. In Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing (pp. 40-51).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

Σχετικά Μαθήματα: Συστήματα Παράλληλης Επεξεργασίας, Νευρωνικά Δίκτυα και Ευφυή Υπολογιστικά Συστήματα

Επικοινωνία: Μαριάννα Τζώρτζη mtzortzi@cslab.ece.ntua.gr
Νικέλα Παπαδοπούλου, nikela@cslab.ece.ntua.gr, 210-772-2279
Γεώργιος Γκούμας, goumas@cslab.ece.ntua.gr, 210-772-2402

II. Αρχιτεκτονική

7 Αρχιτεκτονική Υπολογιστών και Μηχανική Μάθηση

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 παρακάτω εργασίες εστιάζουν τόσο στην χρήση τεχνικών μηχανικής μάθησης στην αρχιτεκτονική υπολογιστών όσο και στην αποδοτική υλοποίηση των ίδιων των αλγορίθμων.

7.1 Εφαρμογή αλγορίθμων μηχανικής μάθησης στην αρχιτεκτονική υπολογιστών

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 για τη βελτιστοποίηση της επίδοσης με στόχο βελτιστοποιήσεις στη χρήση των κρυφών μνημών (caches, prefetching), στο μηχανισμό εικονικής μνήμης (TLBs), στο μηχανισμό πρόβλεψης διακλαδώσεων (branch prediction) 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. BranchNet: A Convolutional Neural Network to Predict Hard-to-Predict Branches
4. Virtual Address Translation via Learned Page Table Indexes
5. SmartChoices: Hybridizing Programming and Machine Learning
6. 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

7.2 Απεικόνιση αλγορίθμων Βαθιάς Μάθησης στην πλατφόρμα Graphcore IPU

Τα IPUs της Graphcore (www.graphcore.ai) υπόσχονται επιδόσεις σημαντικά καλύτερες από GPUs στην επεξεργασία εφαρμογών βαθιάς μάθησης (που μπορούν να εκφραστούν ως γράφοι) με χρήση παραλληλισμού και πολλαπλών τοπικών μνημών. Τα IPUs της GraphCore προγραμματίζονται με το Poplar toolflow και υποστηρίζουν την απεικόνιση CNN (π.χ. με PyTorch). Ενδεικτικά, η εργασία αυτή θα ασχοληθεί με τα παρακάτω:

- Απεικόνιση ενός αλγορίθμου Histogram-of-Oriented-Gradients (HOG) με χρήση του προγραμματιστικού περιβάλλοντος Poplar και σύγκριση με multi-core CPUs και GPUs ως προς επιδόσεις, κλιμακωσιμότητα, κλπ.
- Αξιοποίηση πολλαπλών IPU για την επίλυση μεγαλύτερων προβλημάτων
- Αντικατάσταση του αλγορίθμου HOG με full-featured CNN και σύγκριση σε IPUs.

Σχετική Βιβλιογραφία:

1. Ενδεικτική εφαρμογή για απεικόνιση:`https://arxiv.org/abs/2006.00816`
2. Πληροφορίες για την πλατφόρμα:`https://www.graphcore.ai/hubfs/Lead%20gen%20assets/DSS8440%20IPU%20Server%20White%20Paper_2020.pdf`

Επικοινωνία:

Διονύσιος Πνευματικάτος, pnevmati@cslab.ece.ntua.gr, 6944763171,
Μηλιάδης Παναγιώτης, pmiliad@cslab.ece.ntua.gr

8 Κύρια Μνήμη Συστήματος

8.1 Αρχιτεκτονικές με ανομοιόμορφη πρόσβαση μνήμης (Non Uniform Memory Access - NUMA)

Η κύρια μνήμη στα σύγχρονα πολυεπεξεργαστικά υπολογιστικά συστήματα είναι συνήθως φυσικά κατανεμημένη σε πολλαπλούς κόμβους αλλα παραμένει λογικά ενιαία (ένας συνεχής χώρος φυσικών διευθύνσεων). Κάθε κόμβος αποτελείται από μια συστάδα επεξεργαστών συνδεδεμενων με μια τοπική μνήμη μέσω ένος κοινού διαύλου. Οι επεξεργαστές όλων των κόμβων μπορούν να προσπελάσουν τόσο την τοπική τους μνήμη όσο και όλες τις απομακρυσμένες (που ανήκουν στους υπόλοιπους κόμβους). Επομένως ο χρόνος πρόσβασης της κύριας μνήμης δεν είναι σταθερός και οι αρχιτεκτονικές αυτές ονομάζονται Ανομοιόμορφης Πρόσβασης Μνήμης ή αλλιώς Non Uniform Memory Access (NUMA).

8.1.1 Ανάλυση της επίδοσης εφαρμογών σε NUMA αρχιτεκτονικές και υλοποίηση αποτελεσματικής κατανομής και χρονοδρομολόγησης

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 σύγχρονων εφαρμογών, καθώς επίσης και η αναγνώριση προτύπων και συμπεριφορών και η κατηγοριοποίηση των εφαρμογών σε ευαίσθητες και μη-ευαίσθητες, ως

προς την συμπεριφορά τους λόγω της NUMA αρχιτεκτονικής. Επιπλέον, θα υλοποιηθεί ένας διαχειριστής πόρων και πολιτικές αποτελεσματικής χρονοδρομολόγησης πολλαπλών εφαρμογών σε συνέχεια προηγούμενης διπλωματικής εργασίας, τόσο μέσω της κατάλληλης κατανομής των κόμβων μνήμης (memory node allocation) και υπολογιστικών κόμβων (compute node allocation), όσο και μέσω κατάλληλης υποστήριξης σε επίπεδο λειτουργικου συστήματος.

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

8.2 Επίδοση και προσαρμογή εφαρμογών σε συστήματα με Non-Volatile Memories

Πρόσφατα έχουν κάνει την εμπορική εμφάνιση τους αναδυόμενες μη πτητικές τεχνολογίες μνήμης (NVM) (δηλ. PCM, STTRAM κ.λπ.) οι οποίες προσφέρουν byte-addressable πρόσβαση σε μόνιμα δεδομένα (διατηρημένα σε αστοχίες ισχύος), με καθυστέρηση πρόσβασης κοντά σε αυτή της τεχνολογίας DRAM. Είναι προσπελάσιμες με απλές load/store εντολές CPU και μπορούν να προσφέρουν χωρητικότητες της τάξης των TBs. Η μη πτητικότητα της τεχνολογίας μνήμης προσφέρει μια μοναδική ευκαιρία για τον επαναπροσδιορισμό της αυστηρής διάκρισης μεταξύ μνήμης (memory) και αποθήκευσης (storage) στη στοίβα υπολογιστών. Στόχος της παρούσας διπλωματικής είναι η εξοικείωση με τα προγραμματιστικά περιβάλλοντα για NVM μνήμες, η ανάλυση επίδοσης εφαρμογών σε αυτά τα συστήματα, και η βελτιστοποίηση εκτέλεσης τους.

8.2.1 Βελτιστοποίηση εφαρμογών

Εφαρμογές έντασης δεδομένων (π.χ. βάσεις) μπορούν να ξαναγραφούν ώστε να αποφύγουν το serialization / deserialization των δεδομενων τους και να κερδίσουν σε επίδοση όταν τρέχουν σε συστήματα με PMem devices. Εκτός από την ριζική αναπροσαρμογή, εφαρμογές μπορούν να ενσωματώσουν μερικώς προγραμματιστικές πρακτικές που εκμεταλλεύονται την PMem τεχνολογία για να επιταχύνουν τα IO operations (e.g. χρήση mmio και μονιμοποίηση (persistence) των δεδομένων από το χώρο χρήστη). Στον αντίποδα, η PMem εισάγει και κόστη στο IO (κυρίως στο mmio) σε σύγκριση με την προσωρινή αποθήκευση IO δεδομένων στη DRAM (page cache buffering): έχει περιορισμένο bandwidth, αυξημένο latency και ασύμμετρα κόστη στα read/write operations. Άρα οι εφαρμογές πρέπει να προσέξουν ποια δεδομένα θα τοποθετήσουν αμειγώς στη PMem και για ποια δεδομένα θα επιτρέψουν το κλασικό DRAM buffering over storage. Σκοπός αυτής της διπλωματικής είναι η μελέτη των παρα πάνω και ο πειραματισμός με την προσαρμογή νέων εφαρμογών σε συστήματα με PMem devices.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών, Εργαστήριο Λειτουργικών Συστημάτων

Σχετική Βιβλιογραφία:

1. Finding and Fixing Performance Pathologies in Persistent Memory Software Stacks
2. Persistent Memory Development Kit
3. Basic Performance Measurements of theIntel Optane DC Persistent Memory Module

Επικοινωνία: Χλόη Αλβέρτη, xalverti@cslab.ece.ntua.gr, 210-772-2279
Γεώργιος Γκούμας, goumas@cslab.ece.ntua.gr, 210-772-4133

8.2.2 Δομές δεδομένων

Στο καινούργιο προγραμματιστικό μοντέλο που οδηγεί η χρήση PMem μνήμης και στο οποίο δεσπόζει ο έλεγχος και η επιβολή της μονιμότητας των δεδομένων (data persistency) από το χώρο χρήστη, ένας θεμελιώδες εργαλείο είναι η μη πτητική στοίβα (non-volatile heap), η χρήση δλδ μιας persistent malloc. Τετοιες λειτουργίες/διεπαφές προσφέρονται από επίσημες βιβλιοθήκες (π.χ. Intel PMDK). και δίνουν τη δυνατότητα σχεδιασμού persistent δομών δεδομένων (π.χ. δέντρα, λίστες, γράφους). Οι δομές αποθηκεύονται μόνιμα σε ενα PMem storage device σε binary μορφή και προσπελαύνονται/ανανεώνονται κατευθείαν στο device). Η δυσκολία που προκύπτει ειναι η δημιουργία δομών που μπορούν να εγγυηθούν τη συνοχη (atomicity/consistency) των δεδομένων και της ιδιας της δομής μετά από ένα σφάλμα συστήματος ή μια αστοχία ισχύος (λειτουργίες που παραδοσιακά προσφέρει ένα σύστημα αρχείων στο χώρο πυρήνα). Στην παρούσα διπλωματική θα μελετήσουμε την αντίστοιχη βιβλιογραφία, θα εξοικειωθούμε με τα προγραμματιστικά εργαλεία και μοντέλα για PMem και θα επιχειρήσουμε το σχεδιασμό μιας persistent δομής (π.χ. skiplist).
Σχετική Βιβλιογραφία:

1. MOD: Minimally Ordered Durable Data structures for Persistent Memory
2. Data structure primitives on persistent memory: an evaluation
3. NV-Heaps: making persistent objects fast and safe with next-generation, non-volatile memories
4. Persistent Memory Development Kit

Επικοινωνία: Κωστής Νίκας, knikas@cslab.ece.ntua.gr, 210-772-2279

8.3 Επεξεργασία στη Μνήμη

Οι πρόσφατες εξελίξεις στην αρχιτεκτονική 3D-stack τεχνολογιών μνήμης (για παράδειγμα High-Bandwidth Memory, HBM) έχουν ανανεώσει το ενδιαφέρον για Επεξεργασία Κοντά στη Μνήμη, ή αλλιώς Near-Data-Processing (NDP). Οι NDP αρχιτεκτονικές έχουν σχεδιαστεί με στόχο να μειώσουν την κίνηση δεδομένων (data movement) μεταξύ του επεξεργαστή και της κύριας μνήμης, τοποθετώντας πυρήνες χαμηλής κατανάλωσης ενέργειας και μικρού κόστους κοντά στην κύρια μνήμη. Πρόσφατες εργασίες [1-3, 7-9, 11, 12] δείχνουν τα οφέλη των NDP αρχιτεκτονικών για παράλληλες εφαρμογές όπως graph-processing, neural networks, bioinformatics και databases. Ο στόχος αυτής της έρευνας είναι να μελετήσει εφαρμογές που επωφελούνται από NDP αρχιτεκτονικές και να αναπτύξει μηχανισμούς και προσομοιωτές για το σκοπό αυτό.

8.3.1 Υλοποίηση Προσομοιωτή (Simulator) για Near-Rank Processing χρησιμοποιώντας τους ZSim [5] και Ramulator [6].

8.3.2 Βελτιστοποίηση του Υπολογιστικού Πυρήνα Πολλαπλασιασμού Αραιού Πινακα με Διάνυσμα (SpMV) μέσω Near-Data Processing.

8.3.3 Σχεδίαση μιας NDP Αρχιτεκτονικής για την Επιτυχάνυση Εφαρμογών για Personalized Recommendation [10-12].

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών, Συστήματα Παράλληλης Επεξεργασίας

Σχετική Βιβλιογραφία:

1. Saugata Ghose et al., ”Processing-in-memory: A workload-driven perspective”, in IBM Journal of Research and Development 2019.

2. Benjamin Y. Cho et al., ”Near Data Acceleration with Concurrent Host Access”, in ISCA 2020.

3. Elliot Lockerman et al., ”Livia: Data-Centric Computing Throughout the Memory Hierarchy”, in ASPLOS 2020.

4. Po-An Tsai et al., ”Adaptive Scheduling for Systems with Asymmetric Memory Hierarchies”, in MICRO 2018.

5. Daniel Sanchez et al., ”ZSim: Fast and Accurate Microarchitectural Simulation of Thousand-Core Systems”, in ISCA 2013.

6. Yoongu Kim et al., ”Ramulator: A Fast and Extensible DRAM Simulator”, in IEEE CAL 2016.

7. Junwhan Ahn et al., ”A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing”, in ISCA 2015.

8. Youwei Zhuo et al., ”GraphQ: Scalable PIM-Based Graph Processing”, in MICRO 2019.

9. Lifeng Nai et al., ”GraphPIM: Enabling Instruction-Level PIM Offloading in Graph Computing Frameworks”, in HPCA 2017.

10. Udit Gupta et al., ”The Architectural Implications of Facebook’s DNN-based Personalized Recommendation”, in HPCA 2020.

11. Youngeun Kwon et al., ”TensorDIMM: A Practical Near-Memory Processing Architecture for Embeddings and Tensor Operations in Deep Learning”, in MICRO 2019.

12. Liu Ke et al., ”RecNMP: Accelerating Personalized Recommendation with Near-Memory Processing”, in ISCA 2020.

Επικοινωνία: Χριστίνα Γιαννούλα, cgiannoula@cslab.ece.ntua.gr

9 Μελέτη οργανώσεων πινάκων σελίδων (page tables)

Το λειτουργικό σύστημα είναι υπεύθυνο για την δέσμευση φυσικής μνήμης και για την αποθήκευση και συντήρηση των αντιστοιχίσεων των εικονικών διευθύνσεων των εφαρμογών σε φυσικές διευθύνσεις. Τη πληροφορία αυτή τη συντηρεί το λειτουργικό σε ειδικές δομές ανά εφαρμογή, τους πίνακες σελίδων (page tables). Οι δομές αυτές είναι παραδοσιακά δενδρικές (radix trees) τις οποίες προσπελαύνει ειδικός μηχανισμός υλικού (hardware page tables walkers) για να βρει τις φυσικές μεταφράσεις εικονικών διευθύνσεων των εφαρμογών κατά την εκτέλεση τους. Το βάθος του δέντρου εξαρτάταται από το μέγεθος του χώρου διευθύνσεων (address space) και ως σήμερα τα δέντρα ήταν τεσσάρων επιπέδων. Με την ολοένα αυξανόμενη χωρητικότητα των κύριων μνημών, το βάθος των δέντρων επίκειται να μεγαλώσει και να γίνει 5 επιπέδων. Το βάθος επηρρεάζει το χρόνο που απαιτείται για την ανάκτηση μιας μετάφρασης και συνεπώς την επίδοση των εφαρμογών. Η επίδρασή πολλαπλασιάζεται όταν οι εφαρμογές εκτελούνται εντός εικονικών μηχανών (virtual machines) όπου χρειάζεται η εμφολευμένη προσπέλαση (nested paging) των πινάκων σελίδων τόσο του guest οσο και του host machine. Σύγχρονες ερευνητικές δουλειές προτείνουν εναλλακτικές οργανώσεις των πινάκων σελίδων, π.χ πίνακες

κατακερματισμού αντί για δέντρα, για τη μείωση του χρόνο του page table walk. Σκοπός της παρούσας διπλωματικής είναι να προσομοιώσει και να αξιολογήσει διαφορετικές οργανώσεις page tables.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών, Λειτουργικά Συστήματα

Σχετική Βιβλιογραφία:

1. Hash don't cache the page table
2. Elastic Cuckoo Page Tables: Rethinking Virtual Memory Translation for Parallelism

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

10 Επεκτάσεις Αρχιτεκτονικής

10.1 Αρχιτεκτονική υποστήριξη για βελτίωση του χρόνου εκκίνησης και εκτέλεσης containers

Πολλοί πάροχοι υποδομών υπολογιστικού νέφους (cloud computing) παρέχουν την δυνατότητα εκτέλεσης εφαρμογών χρησιμοποιώντας το περιβάλλον των containers. Ωστόσο, σε αυτό το περιβάλλον εκτέλεσης υπάρχουν δύο αιτίες που επηρρεάζουν την απόδοση των εφαρμογών: (α) η αργή αρχικοποίηση (boot time due to cold starts) των containers (για παράδειγμα, serverless functions [2,3,4,5,1]), και (β) η αργή εκτέλεση των κλήσεων συστήματος (system calls) λόγω των επιπλέον ελέγχων που γίνονται [6] (για παράδειγμα, I/O intensive applications).

Σε αυτή τη διπλωματική εργασία, θα αναλύσουμε την εκτέλεση υπάρχοντων περιβάλλοντων εκτέλεσης containers (για παράδειγμα, Docker, gVisor, Firecracker) για να καταλάβουμε καλύτερα τις συνέπειες της εκτέλεσης εφρμογών σε docker, και θα αναγωνρίσουμε λειτουργίες που έχουν την δυνατότητα να επιταχυνθούν μέσω ειδικής υποστήριξης στο υλικό. Πιο συγκεκριμένα, θα επικεντρωθούμε σε εκείνα τα επίπεδα εικονικοποίησης που επιτρέπουν την απομόνωση εφαρμογών (e.g. SecComp [6], Namespaces). Αυτά τα επίπεδα εικονικοποίησης χρησιμοποιουν διάφορους πίνακες που χρειάζονται να δημιουργούνται, να ενημερώνονται, και να χρησιμοποιούνται από τον πυρήνα του λειτουργικού συστήματος, για να παρέχεται απομόνωση των εφσρμογών. Μετά την ανάλυση, θα επικεντρωθούμε στην ανάπτυξη ειδικής υποστήριξης σε επίπεδο υλικού και αρχιτεκτονικής με σκοπό να μειώσουμε τον χρόνο αρχικοποίησης των containers και το κόστος εκτέλεσης των system calls.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών, Εργαστήριο Λειτουργικών Συστημάτων

Σχετική Βιβλιογραφία:

1. Architectural Implications of Function-as-a-Service Computing, MICRO 2019
2. Catalyzer: Sub-millisecond Startup for Serverless Computing with Initialization-less Booting, ASPLOS 2020
3. SOCK: Rapid Task Provisioning with Serverless-Optimized Containers
4. SAND: Towards High-Performance Serverless Computing
5. Serverless in the Wild: Characterizing and Optimizing the Serverless Workload at a Large Cloud Provider, ATC 2020

6. Draco: Architectural and Operating System Support for System Call, Security MICRO 2020

7. BabelFish: Fusing Address Translations for Containers, ISCA 2020

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

11 Αρχιτεκτονική RISC-V

Η RISC-V αρχιτεκτονική είναι μια ανοικτή και επεκτάσιμη αρχιτεκτονική συνόλου εντολών που ξεκίνησε να αναπτύσσεται στο Πανεπιστήμιο του Berkeley το 2010 και από το 2016 λαμβάνει διεθνή προσοχή 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). Ως ανοικτή και επεκτάσιμη αρχιτεκτονική προσφέρεται για την έρευνα σε λειτουργικές επεκτάσεις, ενώ πολλές υλοποιήσεις ανοικτού κώδικα είναι άμεσα διαθέσιμες, άλλες απλούστερες με γραμμική in-order pipeline και άλλες μεγαλύτερων επιδόσεων με πυρήνα εκτέλεσης εντολών εκτός σειράς (out-of-order).

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

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 Protocol Buffers, MICRO 2021

2. A Hardware Accelerator for Tracing Garbage Collection

3. `https://en.wikipedia.org/wiki/RISC-V`

4. `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

11.2 Αξιόπιστα Περιβάλλοντα Εκτέλεσης

Ένα αξιόπιστο περιβάλλον εκτέλεσης (Trusted Execution Environment - TEE) είναι μία εναλλακτική λειτουργία του επεξεργαστή η οποία προσφέρει υψηλότερα επίπεδα ασφάλειας. Όταν ένας επεξεργαστής βρίσκεται σε λειτουργία ασφαλούς εκτέλεσης εγγυάται ότι ο κώδικας και τα δεδομένα μίας εφαρμογής που τρέχει εντός του TEE διατηρούνται απόρρητα (confidentiality), και προστατεύεται η ακεραιότητα (integrity) τους. Ένα TEE είναι ένα απομονωμένο περιβάλλον εκτέλεσης το οποίο παρέχει επιπλέον δικλείδες ασφαλείας σε σχέση με την ”απλή” εκτέλεση του επεξεργαστή. Ο χώρος εκτέλεσης ενός TEE -ονομάζεται enclave- προσφέρει υψηλότερη ασφάλεια στις εφαρμογές από την τυπική

ασφάλεια που προσφέρει ένα λειτουργικό σύστημα, χωρίς όμως να χάνει τα προσόντα του λειτουργικού συστήματος όπως τα system calls, I/O, διαδιεργασιακή επικοινωνία κλπ.

Τα θετικά οφέλη των TEE αντισταθμίζονται από την πιθανή μείωση της επίδοσης των εφαρμογών λόγω των παρενεργειών της πρόσθετης ασφάλειας. Το υψηλότερο κόστος των context-switches λόγω των επικείμενων TLB και Cache flushes, καθώς και το επιπλέον υπολογιστικό κόστος για integrity-confidentiality δημιουργούν την ανάγκη για εξέταση εναλλακτικών τεχνικών που να προσφέρουν ικανοποιητικά επίπεδα ασφάλειας, χωρίς την μείωση της επίδοσης. Στην συγκεκριμένη διπλωματική θα ασχοληθούμε με το Keystone TEE της αρχικτεκτονικής RISC-V, την ανάλυση της συμπεριφοράς του και τις πιθανές τεχνικές/μεθόδους βελτίωσης της επίδοσης των εφαρμογών που τρέχουν εντός ενός Keystone enclave. Οι πιθανές προεκτάσεις της παρούσας διπλωματικής μπορεί να αφορούν το Keystone Framework, το λειτουργικό σύστημα Linux ή και την υποκείμενη αρχιτεκτονική RISC-V.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών, Εργαστήριο Λειτουργικών Συστημάτων

Σχετική Βιβλιογραφία:

[1] Trusted Execution Environment (Wikipedia Page),
`https://en.wikipedia.org/wiki/Trusted_execution_environment`

[2] Keystone: An Open Framework for Architecting Trusted Execution Environments,
`http://n.ethz.ch/~sshivaji/publications/keystone_eurosys20.pdf`

Επικοινωνία: Νίκος Χ. Παπαδόπουλος, ncpapad@cslab.ece.ntua.gr
Βασίλης Καρακώστας, vkarakos@cslab.ece.ntua.gr, 210-772-4133

11.3 Σχεδιασμός, υλοποίηση, και αξιολόγηση επίδοσης προηγμένων σχημάτων κρυφής μνήμης του συστήματος διαχείρισης εικονικής μνήμης του Rocket Chip Generator

Ο Rocket Chip Generator [2] είναι ένας ανοιχτού κώδικα System-on-Chip (SoC) Generator που παράγει παραμετροποιήσιμα RISC-V SoCs. Είναι υλοποιημένος στην γλώσσα Chisel [3], η οποία καθιστά εύκολη την περιγραφή πολύπλοκων και παραμετροποιήσιμων γεννητριών για επεξεργαστικούς πυρήνες, κρυφές μνήμες και δικτύων διασύνδεσης εντός του SoC.

Στα θέματα των υποενοτήτων που ακολουθούν θα ασχοληθούμε με το σύστημα διαχείρισης εικονικής μνήμης του Rocket Chip Generator. Το σύστημα διαχείρισης εικονικής μνήμης (Memory Management Unit - MMU) παίζει διττό ρόλο στα σύγχρονα υπολογιστικά συστήματα, (i) διασφαλίζει την μνήμη του συστήματος μέσω της απομόνωσης των διεργασιών και (ii) ενισχύει την παραγωγικότητα του προγραμματιστή. Τα παραπάνω επιτυγχάνονται με την χρήση εικονικών διευθύνσεων πρόσβασης στην μνήμη αντί για φυσικών, και με τον διαχωρισμό της μνήμης σε σελίδες (συνήθως των 4KB). Με την χρήση εικονικών διευθύνσεων κάθε εφαρμογή ”νομίζει” ότι δουλεύει πάνω σε συνεχόμενες σελίδες μνήμης, ενώ στην πραγματικότητα οι σελίδες μπορεί να είναι διάσπαρτες στην φυσική μνήμη. Το σύστημα διαχείρισης εικονικής μνήμης γνωστών αρχιτεκτονικών (x86, ARM, RISC-V, κλπ) αποτελείται από την μονάδα του Page Table Walker ο οποίος είναι υπεύθυνος για την μετάφραση των διευθύνσεων από εικονικές σε φυσικές, καθώς και από τον Translation Lookaside Buffer (TLB), μία κρυφή μνήμη, στην οποία κρατούνται οι πρόσφατες εικονικές-σε-φυσικές μεταφράσεις διευθύνσεων. Στα παρακάτω θέματα θα χρησιμοποιήσουμε εργαλεία ανοικτού κώδικα για την ανάπτυξη υλικού, επιβεβαίωσης ορθής λειτουργίας του, καθώς και FPGAs (Field Programmable Gate Arrays) για την μελέτη επίδοσης του υλικού που σχεδιάσαμε.

Σχετική Βιβλιογραφία:

[1] RISC-V Technical Specifications,
https://riscv.org/technical/specifications/

[2] Rocket Chip Generator,
https://github.com/chipsalliance/rocket-chip/

[3] The Chisel Language,
https://www.chisel-lang.org

11.3.1 Advanced MMU Caching Techniques for the Rocket Chip Generator

Σε πολλά σύγχρονα benchmarks/workloads, η μετάφραση των εικονικών διευθύνσεων μπορεί να επιβαρύνει αισθητά την επίδοση και την ενεργειακή απόδοση του υπολογιστικού συστήματος, λόγω των αστοχιών TLB. Αναλόγως της αρχιτεκτονικής του πίνακα σελίδων, απαιτούνται 3-4 προσβάσεις στην μνήμη για την μετάφραση της εικονικής-σε-φυσική διεύθυνση. Συγκεκριμένα, σε περιβάλλοντα εικονοποίησης ο αριθμός προσβάσεων στην μνήμη μπορεί να φτάσει τις 24. Μία πρόταση στο παραπάνω πρόβλημα είναι η χρήση μεγαλύτερης χωρητικότητας -ή μεγαλύτερης συσχετιστικότητας- TLB. Όμως, το TLB βρίσκεται στο critical path του επεξεργαστή με αποτέλεσμα να επηρεάζει τον χρονισμό του: δημιουργείται ένα trade-off μεταξύ μεγέθους TLB (χαμηλότερος χρονισμός CPU) και αστοχιών TLB (χαμηλότερη επίδοση). Στην σύγχρονη βιβλιογραφία προτείνονται διάφορες αρχιτεκτονικές για caching εικονικής μνήμης, όπως τα Coalesced TLBs [1], Clustered TLBs [2], Hybrid TLB Coalescing [3], Direct Segments [4], Redundant Memory Mappings [5] και άλλα. Η παρούσα διπλωματική στοχεύει σε συνέχεια προηγούμενης διπλωματικής εργασίας στην υλοποίηση κάποιου -ή κάποιων- από τα παραπάνω σχήματα στον Rocket Chip Generator (RCG), και να εξεταστεί η επίδοση τους έναντι του vanilla TLB του RCG.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

Σχετική Βιβλιογραφία:

[1] CoLT: Coalesced Large-Reach TLBs,
https://ieeexplore.ieee.org/document/6493625

[2] Increasing TLB Reach by Exploiting Clustering in Page Translations,
http://www.cs.yale.edu/homes/abhishek/binhpham-hpca14.pdf

[3] Hybrid TLB Coalescing: Improving TLB Translation Coverage under Diverse Fragmented Memory Allocations,
https://iamchanghyunpark.github.io/papers/htc-isca2017.pdf

[4] Efficient Virtual Memory for Big Memory Servers,
https://research.cs.wisc.edu/multifacet/papers/isca13_direct_segment.pdf

[5] Redundant Memory Mappings for Fast Access to Large Memories,
http://www.cslab.ece.ntua.gr/~vkarakos/papers/isca15_redundant_memory_mappings.pdf

11.3.2 Enabling TLB Prefetching for the Rocket Chip Generator

Μία άλλη τεχνική για την βελτίωση της επίδοσης του συστήματος εικονικής μνήμης είναι το prefetching [1]. Η τεχνική αυτή βασίζεται στην εικασία ότι φέρνοντας δεδομένα από την κύρια μνήμη στην κρυφή μνήμη πριν χρειαστούν, θα μηδενιστεί η αναμονή σε μελλοντική πρόσβαση στα δεδομένα αυτά. Στην

διπλωματική αυτή, θα μελετηθεί η ανάπτυξη prefetcher για το TLB [2, 3, 4] του Rocket Chip Generator, και θα εξεταστεί η επίδοση του χρησιμοποιώντας benchmarking suites όπως το SPEC2006/SPEC2017.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

Σχετική Βιβλιογραφία:

[1] Cache Prefetching (Wikipedia Page)
`https://en.wikipedia.org/wiki/Cache_prefetching`

[2] Exploiting Page Table Locality for Agile TLB Prefetching,
`https://ieeexplore.ieee.org/document/9499825`

[3] Recency-Based TLB Preloading,
`https://courses.cs.washington.edu/courses/cse590g/00au/p117-saulsbury.pdf`

[4] Going the Distance for TLB Prefetching: An Application-driven Study,
`http://www.cse.psu.edu/~axs53/csl/papers/isca02.pdf`

[5] Inter-core cooperative TLB for chip multiprocessors,
`https://dl.acm.org/doi/abs/10.1145/1735970.1736060`

11.3.3 Enabling Configurable Page Table Walk Caches for the Rocket Chip Generator

Η μονάδα που αναλαμβάνει την μετάφραση μιας εικονικής διεύθυνσης σε φυσική ονομάζεται Page Table Walker (PTW) και είναι συνήθως υλοποιημένη στο υλικό για την ταχύτερη μετάφραση των εικονικών διευθύνσεων. Η δομή δεδομένων που χρησιμοποιείται για το mapping των εικονικών σε φυσικές διευθύνσεις ονομάζεται πίνακας σελίδων (page table) [1] και αναλόγως την αρχιτεκτονική, αποτελείται από 3 ή 4 επίπεδα. Στον Rocket Chip Generator (RCG) ο πίνακας σελίδων αποτελείται από 3 επιπεδα για το σχήμα εικονικής μνήμης Sv39 [2]. Σε περίπτωση αστοχίας TLB, η μονάδα PTW πρέπει να κάνει επομένως 3 προσβάσεις στον πίνακα σελίδων ώστε να μεταφράσει την ζητούμενη εικονική διεύθυνση σε φυσική. Για να αποφευχθούν οι κοστοβόρες προσβάσεις στην μνήμη, στον RCG έχει υλοποιηθεί μία μικρή PTW Cache [3] η οποία αποθηκεύει το mapping των πρώτων 2 επιπέδων (το mapping του 3ου επιπέδου αποθηκεύεται στο TLB). Αντικείμενο της διπλωματικής αυτής είναι η παραμετροποίηση της PTW Cache του RCG, η μελέτη υλοποίησης πιο προηγμένων σχημάτων PTW Cache [3], και η εξέταση της επίδοσης τους χρησιμοποιώντας benchmarking suites όπως το SPEC2006/SPEC2017.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

Σχετική Βιβλιογραφία:

[1] Page Table (Wikipedia Page),
`https://en.wikipedia.org/wiki/Page_table`

[2] RISC-V Page-Based 39-bit Virtual-Memory System, pages 62-64,
`https://riscv.org//wp-content/uploads/2017/05/riscv-privileged-v1.10.pdf`

[3] Translation caching: skip, don't walk (the page table),
`https://dl.acm.org/doi/10.1145/1816038.1815970`

Επικοινωνία: Νίκος Χ. Παπαδόπουλος, ncpapad@cslab.ece.ntua.gr
Βασίλης Καρακώστας, vkarakos@cslab.ece.ntua.gr, 210-772-4133

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

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

12 Αποδοτική απεικόνιση σε FPGAs

Στις μέρες μας, η χρήση των επαναπρογραμματιζόμενων αρχιτεκτονικών (FPGAs) αποτελεί σημαντική εναλλακτική πρόταση, καθώς προσφέρει τη σχεδίαση υλικού για την εκτέλεση συγκεκριμένων εφαρμογών (application-specific), με σκοπό τη βελτιστοποίηση και την επιτάχυνση του χρόνου εκτέλεσης. Αν και μπορούν να απεικονίσουν όμως οποιαδήποτε σχεδίαση υλικού, οι FPGAs έχουν ιδιαιτερότητες και «προτιμήσεις».

12.1 Αποδοτική απεικόνιση επεξεργαστών RISC-V σε FPGA

Η εργασία αυτή αφορά αφενός την συγκριτική μελέτη βασικών υπαρχόντων υλοποιήσεων RISC-V 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 περισσότεραο «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

12.2 Υλοποίηση Compiler για τη διάσπαση σχεδίασης σε πολλαπλά μικρότερα bitstreams και την αποδοτική χαρτογράφιση τους σε FPGAs

Στη σύγχρονη εποχή οι FPGAs ενσωματώνονται σε συστήματα cloud, με πιο χαρακτηριστικά παραδείγματα το F1 instance της Amazon και της Alibaba. Κάθε χρήστης μπορεί να νοικιάσει ένα σύστημα στο οποίο ενσωματόνονται FPGAs, προκειμένου να ενσωματώσει και να τρέξει τη σχεδίαση του ή τον υπολογιστικό του πυρήνα (Platform as a Service). Όμως, σπανίως ένας χρήστης αξιοποιεί πλήρως την έκταση μιας FPGA, αφήνοντας ένα μεγάλο μέρος των πόρων ανεκμετάλλευτο. Σκοπός στο cloud, είναι η πλήρης εκμετάλλευση της έκτασης ενός FPGA, δηλαδή την υπολογιστική του δύναμη, μοιράζοντας τους διαθέσιμους πόρους μεταξύ πολλών χρηστών (multi-tenant) ταυτόχρονα. Έτσι, προτείνεται η δημιουργία μικρότερων fixed-slots εντός της FPGA, όπου η σχεδίαση κάθε χρήστη θα χαρτογραφείται στο νέο περιορισμένο χώρο. Δυστυχώς, πολλές σχεδιάσεις δεν ταιριάζουν στο νέο περιορισμένο χώρο που διατίθεται από τα νέα συστήματα των παρόχων, με αποτέλεσμα την ανάγκη διάσπασης της σχεδίασης σε μικρότερα κομμάτια, όπου κάθε κομμάτι θα χαρτογραφείται σε ένα fixed-slot, ικανοποιώντας τους χωρικούς περιορισμούς που επιβάλλονται. Σκοπός της διπλωματικής εργασίας είναι η ανάπτυξη ενός compiler όπου θα δέχεται σαν είσοδο την σχεδίαση ενός χρήστη σε επίπεδο HLS ή HDL, και θα διασπά τη σχεδίαση σε μικρότερα bitstreams, με σκοπό την αποδοτική χαρτογράφιση των επιμέρους κομματιών στο νέο περιορισμένο χώρο που διατίθεται.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

Σχετική Βιβλιογραφία:

1. Yue Zha and Jing Li. 2020. Virtualizing FPGAs in the Cloud. In Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '20). Association for Computing Machinery, New York, NY, USA, 845–858. DOI:https://doi.org/10.1145/3373376.3378491
2. Y. Zha and J. Li, "Hetero-ViTAL: A Virtualization Stack for Heterogeneous FPGA Clusters," 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), 2021, pp. 470-483, doi: 10.1109/ISCA52012.2021.00044.
3. https://www.rapidwright.io/
4. https://github.com/Xilinx/RapidWright

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

12.3 Υλοποίηση context-switch και preemption μηχανισμού για υπολογιστικούς πυρήνες σε FPGAs.

Στη σύγχρονη εποχή οι FPGAs ενσωματώνονται σε συστήματα cloud, με πιο χαρακτηριστικά παραδείγματα το F1 instance της Amazon και της Alibaba. Επιπλέον, οι χρήστες έχουν τη δυνατότητα να αξιοποιήσουν έτοιμες υλοποιημένες συναρτήσεις από διαθέσιμες βιβλιοθήκες των παρόχων στα προγράμματα τους, που έχουν σχεδιαστεί και ενσωματωθεί σε FPGAs (Software as a Service). Κάθε υπολογιστικός πυρήρας όμως μπορεί να εξυπηρετήσει μόνο έναν χρήστη (task-based), ενώ οι υπόλοιποι χρήστες πρέπει να περιμένουν στην ουρά προκειμένου να γίνει διαθέσιμος εκ νέου ο υπολογιστικός πυρήνας. Η κατάσταση αυτή οδηγεί σε μια σειρά προβλημάτων, με πιο κύρια την μη-δίκαιη κατανομή του πυρήνα μεταξύ των χρηστών. Μια λύση του παραπάνω προβλημάτος αποτελεί η δημιουργία ενός preemption και context-switch μηχανισμού, ο οποίος α) θα διακόπτει μια διεργασία, β) θα αποθηκεύει την κατάσταση της και στη συνέχεια γ) θα φορτώνει την επόμενη διεργασία της ουράς, όπως γίνεται στα λειτουγρικά συστήματα. Η ενσωμάτωση του παραπάνου μηχανισμού σε μια σχεδίαση ενός υπολογιστικού πυρήνα, επιτρέπει τη κοινή χρήση του επιταχυντή με σκοπό τη δίκαιη κατανομή του από διαφορετικούς χρήστες (ή διεργασίες). Σκοπός της διπλωματικής εργασίας είναι η ανάπτυξη ενός μηχανισμού που θα επιτρέπει τη κοινή χρήση ενός επιταχυντή, ο οποίος έχει σχεδιαστεί και υλοποιηθεί για FPGAs, από διαφορετικούς χρήστες (ή/και διεργασίες) στον άξονα του χρόνου, μέσω της υλοποίησης context-switch και preemption μηχανισμού.

Σχετικά Μαθήματα: Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

Σχετική Βιβλιογραφία:

1. Jiacheng Ma, Gefei Zuo, Kevin Loughlin, Xiaohe Cheng, Yanqiang Liu, Abel Mulugeta Eneyew, Zhengwei Qi, and Baris Kasikci. 2020. A Hypervisor for Shared-Memory FPGA Platforms. In Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '20). Association for Computing Machinery, New York, NY, USA, 827–844. DOI:https://doi.org/10.1145/3373376.3378482

2. Ahmed Khawaja, Joshua Landgraf, Rohith Prakash, Michael Wei, Eric Schkufza, and Christopher J. Rossbach. 2018. Sharing, protection, and compatibility for reconfigurable fabric with Amorphos. In Proceedings of the 13th USENIX conference on Operating Systems Design and Implementation (OSDI'18). USENIX Association, USA, 107–127.

3. Joshua Landgraf, Tiffany Yang, Will Lin, Christopher J. Rossbach, and Eric Schkufza. 2021. Compiler-driven FPGA virtualization with SYNERGY. In Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2021). Association for Computing Machinery, New York, NY, USA, 818–831. DOI:https://doi.org/10.1145/3445814.3446755

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

III. Λειτουργικά Συστήματα - Εικονικές Μηχανές

13 Persistent Memory (PMem) – Μη πτητική μνήμη

Πρόσφατα έγιναν εμπορικά διαθέσιμες μη πτητικές τεχνολογίες μνήμης (NVM) (e.g. Intel Optane NVDIMM) οι οποίες έχουν καλύτερη κλιμάκωση πυκνότητας με χαμηλότερο κόστος σε σχέση με τη τεχνολογία DRAM. Οι NVM μνήμες προσφέρουν byte-addressable πρόσβαση σε μόνιμα δεδομένα (διατηρημένα σε αστοχίες ισχύος) μέσω εντολών μνήμης CPU (loads/stores). Ο χρόνος απόκρισης (latency) και το εύρος ζώνης (bandwidth) που προσφέρουν είναι "κοντά" σε αυτα της τεχνολογίας DRAM, αλλά παραμένουν σημαντικά χειρότερα παρουσιάζοντας ταυτόχρονα ιδιαίτερα χαρακτηριστικά (π.χ. ανομοιομορφία read/write access). Η μη πτητικότητα, η επίδοση και η πυκνότητα τετοιων μνημών (μπορούν να προσφέρουν χωρητικότητες της τάξης των TBs) δίνουν τη μοναδική ευκαιρία για χρήση τους είτε ως μέσου γρήγορης αποθήκευσης (storage), είτε ως ενός έξτρα επιπέδου πιο αργής μνήμης (slow memory). Η δυνατότητα άμεσης προσπέλασης από τη CPU (direct access via load/store) και στις δύο περιπτώσεις επιτρέπει ακόμα και τον ρηξικέλευθο επαναπροσδιορισμό της αυστηρής διάκρισης μεταξύ μνήμης και αποθήκευσης στη στοίβα υπολογιστών.

13.1 Persistent memory as storage

Δεδομένου ότι ο χρόνος απόκρισης της PMem είναι τάξεις μεγεθους μικρότερος από κλασικά storage devices (e.g. flash ssds or hdds), το κόστος του λογισμικού συτήματος εμφανίζεται συχνά ως το νέο σημείο συμφόρησης της επίδοσης των εφαρμογών έντασης IO. Γι'αυτό γίνεται προσπάθεια τόσο από την ακαδημαική κοινότητα όσο και από τη βιομηχανία να βελτιστοποιηθεί το ΛΣ για PMem συσκευές (π.χ. αναπτύσσονται ειδικά συστήματα αρχείων). Η PMem είναι συνδεδεμένη στο memory bus, και η προσπέλαση της γίνεται με CPU εντολες load και store. Ετσι, το πιο δημοφιλές interface για να προσπελάσει κανείς αρχεία αποθηκευμένα σε PMem ειναι με τη κλήση συστήματος mmap() (memory-mapped IO). Με τον τρόπο αυτό η εκάστοτε εφαρμογή αποκτά πρόσβαση σε μόνιμα δεδομένα χρησιμοποιώντας απλά ένα εύρος εικονικών διευθύνσεων (direct access –DAX), το πιο σύντομο μονοπάτι πρόσβασης σε χώρο αποθήκευσης. Οι παρακάτω δύο διπλωματικές θα μελετήσουν και θα αναζητήσουν λύσεις για δύο σημαντικές πηγές κόστους επίδοσης κατά το memory-mapped IO σε PMem.

13.1.1 Σελιδοποίηση και κόστος μετάφρασης σε συστήματα με PMem storage. Μελέτη επίδοσης και υποστήριξης μεγάλων σελίδων (2MB και 1GB).

Στο memory-mapped IO ένα από τα βασικά κόστη του λογισμικού συστήματος προκύπτει από τα σφάλματα σελίδας (page faults). Το κόστος αυτό μειώνεται δραστικά με τη χρήση μεγάλων σελίδων, οι οποίες εν δυνάμει επιταχύνουν και την εκτέλεση μιας εφαρμογής μειώνοντας των αριθμό των TLB αστοχιών μετάφρασης σε επίπεδο μικροαρχιτεκοτνικής. Ωστόσο τα συστήματα αρχείων και συγκεκριμένα οι κατανεμητές τους (allocators) δεν έχουν δομηθεί/βελτιστοποιηθεί για τη χρήση μεγάλων σελίδων. Επίσης, τα συστήματα αρχείων υποφέρουν από εξωτερικό κατακερματισμό (με διαφορετικά και μόνιμα χαρακτηριστικά σε σχέση με τον κατακερματισμό μνήμης) ο οποίος δυσχεραίνει τη χρήση μεγάλων σελίδων. Στη παρούσα διπλωματική θα μελετήσουμε τη χρήσης μεγάλων σελίδων στην επίδοση εφαρμογών έντασης IO και την επίπτωση του κατακερματισμού των συστημάτων αρχείων. Στη συνέχεια θα επιχειρήσουμε τρόπους αντιμετώπισης του κατακερματισμού είτε με παρεμβάσεις στον κατανεμητή είτε με εργαλεία ανασυγκρότησης στο ερευνητικό σύστημα αρχείων NOVA.

Σχετικά Μαθήματα: Εργαστήριο Λειουργικών Συστημάτων

Σχετική Βιβλιογραφία:

1. WineFS: a hugepage-aware file system for persistent memory that ages gracefully

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

13.1.2 PMem storage σε NUMA διαταξη: τοποθέτηση δεδομένων και διεργασιών.

Το κόστος απομακρυσμένης προσπέλασης σε μια PMem NUMA διάταξη ειναι πολύ μεγαλύτερο σε σύγκριση με μια ανομοιόμορφη DRAM προσπέλαση, ειδικά για παράλληλες εφαρμογές. Σκοπός της διπλωματικής αυτής είναι να μελετήσει PMem διατάξεις με Ανομοιόμορφη Πρόσβαση Μνήμης ή αλλιώς Non Uniform Memory Access (NUMA), όπου η μη πτητική μνήμη χρησιμοποιείται και πάλι ως storage. Αφού μελετηθεί το κόστος της ανομοιμορφίας, θα επιχειρηθεί να υλοποιηθεί σε χώρο πυρήνα (και συγκεκριμένα στα πλαίσια του συστήματος αρχείων NUMA) πολιτική μεταφοράς δεδομένων και νημάτων εκτέλεσης μεταξύ NUMA κόμβων, για τη βλετιστοποίηση της επίδοσης.

Σχετικά Μαθήματα: Εργαστήριο Λειουργικών Συστημάτων

Σχετική Βιβλιογραφία:

1. NUMA-Aware Thread Migration for High Performance NVMM File Systems
2. An Empirical Guide to the Behavior and Use of Scalable Persistent Memory
3. Maximizing Persistent Memory Bandwidth Utilization for OLAP Workloads

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

13.2 Persistent memory as a memory tier

Οι μνήμες είναι ενα από τα πιο ακριβά κομμάτια των υπολογιστικών συστημάτων, και επηρρεάζουν σημαντικά την ταχύτητα εκτέλεσης σημαντικών εφαρμογών που κυριαρχούν στο υπολογιστικό νέφος (πχ. key-value stores, in-memory databases, graph analytics). Αλλά καθώς τα σύνολα δεδομένων των εφαρμογών συνεχίζουν να αυξάνονται, οι απαιτήσεις σε χωρητικότητα μνήμης αυξάνονται ακόμα περισσότερο. Ταυτόχρονα, η τεχνολογία της παραδοσιακής κύριας μνήμης (DRAM) έχει φτάσει σε ένα όριο κλιμάκωσης που περιορίζει την πυκνότητά της. Η κλιμακωσιμότητα της PMem μπορεί να δωσει λύση σε αυτό το πρόβλημα, αλλά η επίδοση της (latency και bandiwth) ειναι σημαντικά υποδεέστερη σε σχέση με τη DRAM. Έτσι εχει προταθεί η χρήση ιεαραρχιών μνημών, συνδυάζοντας αργές (PMem) και γρήγορες (DRAM) μνήμες. Σε τέτοια διατάξεις η μη πτητικότητα της PMem αγνοείται κα ιχρησιμοποιείται σαν μια πιο αργή και λιγότερο ενεργειακά κοστοβόρα μνήμη.

13.2.1 Συστήματα με υβριδική ιερραρχία μνήμης (DRAM + PMem). Μελέτη τεχνικών τοποθέτησης και μεταφοράς δεδομένων μεταξύ των διαφορετικών επιπέδων μνήμης.

Σε μια υβριδική ιεραρχία αργών και γρήγορων μνημών, η τοποθέτηση των δεδομένων στα διαφορετικά επίπεδα παίζει καταλυτικό ρόλο στην τελική επίδοση. Η τοποθέτηση μπορεί να γίνει αμειγώς από το υλικό (Intel Optane memory mode), αλλά ερευνητικές εργασίες δείχνουν οτι μια τετοια προσέγγιση παρότι διαφανής ειναι μάλλον μονολιθική, χάνοντας πολλές ευκαιρίες βελτιστοποίησης. Στη παρούσα διπλωματική θα μελετήσουμε τεχνικές τοποθέτησης στα διάφορα επίπεδα μνήμης από το λογισμικό στηριζόμενοι στη σχετική βιβλιογραφία. Θα αξιολογήσουμε τη πιθανή μεταφορά τους στο χώρο του πυρήνα και θα συμπεριλάβουμε και την κατανάλωση ενέργειας ως κριτήριο τοποθέτησης (εκτός από την επίδοση).

Σχετικά Μαθήματα: Εργαστήριο Λειουργικών Συστημάτων, Αρχιτεκτονική Υπολογιστών, Συστήματα Παράλληλης επεξεργασίας

Σχετική Βιβλιογραφία:

1. HeMem: Scalable Tiered Memory Management for Big Data Applications and Real NVM
2. An Empirical Guide to the Behavior and Use of Scalable Persistent Memory

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

14 Ανάλυση και βελτιστοποίηση του μηχανισμού και των πολιτικών διαχείρισης μνήμης Automatic NUMA balancing για αρχιτεκτονικές με ανομοιόμορφη πρόσβαση μνήμης

Ένας από τους τρόπους αύξησης της κλιμακωσίμοτητας 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 προσπελαύνει 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. Το λειτουργικό σύστημα Linux παρέχει τον μηχανισμό Automatic NUMA Balancing ο οποίος μεταφέρει δυναμικά δεδομένα ανάμεσα σε κόμβους μνήμης για να μειώσει τον χρόνο πρόσβασης στην μνήμη. Στόχος της παρούσας διπλωματικής είναι η μελέτη, η ανάλυση, και η βελτιστοποίηση του μηχανισμού Automatic NUMA balancing του Linux, καθώς επίσης και η υλοποίηση αποδοτικών πολιτικών χρήσης του.

Σχετικά Μαθήματα: Εργαστήριο Λειτουργικών Συστημάτων

Σχετική Βιβλιογραφία:

1. Automatic Non-Uniform Memory Access (NUMA) Balancing, SUSE
2. Automatic NUMA Balancing, Red Hat
3. NUMA Memory Architectures and the Linux Memory System, Red Hat

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

15 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: Στράτος Ψωμαδάκης, psomas@cslab.ece.ntua.gr
Ορέστης Λάγκας-Νικολός, olagkas@cslab.ece.ntua.gr
Κωνσταντίνος Παπαζαφειρόπουλος, kpapazaf@cslab.ece.ntua.gr

16 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/opencl/`
    2. `https://www.xilinx.com/products/design-tools/software-zone/sdaccel.html`

Επικοινωνία: Κωνσταντίνος Παπαζαφειρόπουλος, kpapazaf@cslab.ece.ntua.gr
Στράτος Ψωμαδάκης, psomas@cslab.ece.ntua.gr
Ορέστης Λάγκας-Νικολός, olagkas@cslab.ece.ntua.gr

IV. Κατανεμημένα Συστήματα - Προχωρημένα θέματα βάσεων δεδομένων

17 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 διάφορες 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. Tsoumakos and N. Koziris: Apollo: A Dataset Profiling and Operator Modeling System. In Proceedings of the 2019 ACM SIGMOD/PODS.
2. I. Giannakopoulos, D. Tsoumakos 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

18 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

19 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

20 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 κατανεμημένο 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 (π.χ., Ethereum [4], Hyperledger [5]) με τη χρήση benchmarks (Blockbench [6] ή Hyperledger Caliper [7] αντίστοιχα) ως προς τις παραπάνω ιδιότητες.

Σχετικά Μαθήματα: Κατανεμημένα 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.

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

21 Μελέτη και βελτιστοποίηση του επιπέδου αποθήκευσης των Blockchain clients

Οι κόμβοι του Blockchain (κυρίως οι full nodes) αποθηκεύουν μεγάλο όγκο δεδομένων που σχετίζονται με το state του, τα transactions που έχουν γίνει, τα δεδομένα των έξυπνων συμβολαίων [1]. Συνήθως τα δεδομένα αυτά φυλάσσονται σε δενδρικές δομές αποθήκευσης (tries) που προσφέρουν γρήγορη αναζήτηση και που υλοποιούνται με τη βοήθεια κάποιου key-value store (leveldb [2], rocksdb [3]). Ωστόσο το storage layer μπορεί σε κάποιες λειτουργίες του πρωτοκόλου να αποτελεί performance bottleneck, όπως για παράδειγμα στο αρχικό sync ενός κόμβου που μόλις πρωτοεισέρχεται στο blockchain [4].

Στόχος της διπλωματικής είναι η μελέτη του workload που καλείται να εξυπηρετήσει το storage layer ενός blockchain client κατά τις διάφορες φάσεις της λειτουργίας του και η βελτιστοποίηση της απόδοσής του με χρήση διαφορετικών δομών αποθήκευσης ή/και τεχνικών caching [5].

Σχετικά Μαθήματα: Κατανεμημένα Συστήματα

Σχετική Βιβλιογραφία:

1. Getting Deep Into Ethereum: How Data Is Stored In Ethereum?
2. LevelDB
3. RocksDB
4. Why Syncing Ethereum Node Is Slow
5. Gorenflo, Christian, et al. ”Fastfabric: Scaling hyperledger fabric to 20,000 transactions per second.” 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). IEEE, 2019.

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

22 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, ονόματει 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 τεστάραμε 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 environemtns. 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

23 Αποτίμηση λειτουργίας συστημάτων επεξεργασίας μεγάλου όγκου δεδομένων πάνω σε skewed σύνολα δεδομένων.

Η ανάλυση μεγάλου όγκου δεδομένων (Big Data) με κατανεμημένα συστήματα επεξεργασίας είναι ένας από τους πιο δημοφιλείς τομείς ανάπτυξης τα τελευταία χρόνια. Τα περισσότερα κατανεμημένα συστήματα επεξεργασίας όμως, δε μπορούν να επεξεργαστούν αποδοτικά δεδομένα του παρουσιάζουν ανομοιομορφίες (skew). Συγκεκριμένα, η ανάλυση δεδομένων με χρήση συνενώσεων (joins) και συνυπολογισμών (aggregations) επηρεάζεται σημαντικά από το skew των δεδομένων, και τα εν λόγω συστήματα επεξεργασίας παρουσιάζουν συνήθως χαμηλή απόδοση σε τέτοιου τύπου ανάλυση πάνω σε skewed δεδομένα. Παρόλο που έχουν γίνει ερευνητικές προσπάθειες αντιμετώπισης του φαινομένου του data skew σε αλγοριθμικό επίπεδο και πάνω σε ειδικά συστήματα [1,2,3], τα περισσότερα από τα υπάρχοντα κατανεμημένα συστήματα επεξεργασίας δεν έχουν υλοποιήσει σχετικούς μηχανισμούς για την αντιμετώπισή του. Το Apache Spark είναι ένα από τα πιο δημοφιλή συστήματα επεξεργασίας μεγάλου όγκου δεδομένων, και παρέχει το SparkSQL[4] module για την ανάλυση δεδομένων με χρήση SQL ερωτημάτων. Σε μια προσπάθεια αντιμετώπισης του προβλήματος αυτού, ανέπτυξε πρόσφατα το adaptive execution framework [5]. Στη συγκεκριμένη διπλωματική, καλείστε να μελετήσετε το adaptive

execution framework του Spark SQL, και να αξιολογήσετε την απόδοση του συστήματος με διαφορετικά επίπεδα data skew και με διαφορετικές ρυθμίσεις του συστήματος χρησιμοποιώντας το TPC-DS benchmark.

Σχετικά Μαθήματα: Προχωρημένα θέματα βάσεων δεδομένων, Κατανεμημένα Συστήματα

Σχετική Βιβλιογραφία:

1. Y. Kwon, M. Balazinska, B. Howe, and J. Rolia. SkewTune: mitigating skew in mapreduce applications. In Proceedings of the 2012 ACM SIGMOD.

2. R. Li, M. Riedewald, and X. Deng. Submodularity of Distributed Join Computation. In Proceedings of the 2018 ACM SIGMOD.

3. W. Rödiger, S. Idicula, A. Kemper and T. Neumann. Flow-Join: Adaptive skew handling for distributed joins over high-speed networks. In Proccedings of the 2016 IEEE ICDE.

4. `https://spark.apache.org/sql/`

5. `https://databricks.com/blog/2020/05/29/adaptive-query-execution-speeding-up-spark-sql-at-runtim`
   `html`

Επικοινωνία: Ευδοκία Κασσέλα, evie@cslab.ece.ntua.gr
Ιωάννης Κωνσταντίνου, ikons@cslab.ece.ntua.gr