

Πρόσκληση σε Δημόσια Παρουσίαση της Διδακτορικής Διατριβής του

κ. Λυμπέρη Σπύρου

Την Δευτέρα, 8 Ιουλίου 2013 και ώρα 11:00 στην αίθουσα “Σ. Ορφανουδάκης” στο Ινστιτούτο Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας (ΙΤΕ) στο Ηράκλειο, θα γίνει η δημόσια παρουσίαση και υποστήριξη της Διδακτορικής Διατριβής του υποψηφίου διδάκτορος του Τμήματος Επιστήμης Υπολογιστών κ. Λυμπέρη Σπύρου με θέμα:

“Myrmics: Ένα Κλιμακώσιμο Σύστημα Χρόνου Εκτέλεσης για Ενοποιημένα Συστήματα Διευθύνσεων”

“Myrmics: A Scalable Runtime System for Global Address Spaces”

ΠΕΡΙΛΗΨΗ

Το τέλος του ανταγωνισμού ταχύτητας των επεξεργαστών στις αρχές του αιώνα σήμανε την έναρξη της εποχής των πολυπύρηνων επεξεργαστών. Για να επωφεληθεί μια εφαρμογή από τους πολλαπλούς πυρήνες, οι προγραμματιστές πρέπει πλέον να χρησιμοποιούν παράλληλα προγραμματιστικά μοντέλα. Οι τάσεις της βιομηχανίας ημιαγωγών δείχνουν ότι οι ένα ολοκληρωμένο κύκλωμα σε μια δεκαετία θα μπορεί να ενσωματώσει εκατοντάδες πυρήνες· η αρχιτεκτονική θα είναι ετερογενής, με λίγους δυνατούς (και ενεργοβόρους) και πολλούς αδύνατους (και καλής ενεργειακής απόδοσης) πυρήνες· οι κρυφές μνήμες θα είναι λιγότερο ή και καθόλου συνεκτικές. Όσο πλησιάζει η νέα εποχή των υπερπολυπύρηνων επεξεργαστών, αποτελεί τεράστια πρόκληση να βρεθεί ένα παραγωγικό, αποδοτικό και κλιμακώσιμο σε τέτοιες αρχιτεκτονικές προγραμματιστικό μοντέλο.

Τα μοντέλα διεργασιών με εξαρτήσεις έχουν σημαντική απήχηση. Ο προγραμματιστής παρέχει ένα σειριακό πρόγραμμα, αποτελούμενο από μικρές συναρτήσεις (διεργασίες) που τρέχουν μέχρι τέλους, μαζί με πληροφορία σχετικά με το ποιά δεδομένα αυτές χρησιμοποιούν. Ένα σύστημα χρόνου εκτέλεσης αναλύει τις εξαρτήσεις, δρομολογεί και εκτελεί τις διεργασίες παράλληλα. Παρά την αυξανόμενη δημοφιλία τους, τα μοντέλα αυτά δεν είναι κλιμακώσιμα στις επερχόμενες υπερπολυπύρηνες αρχιτεκτονικές, αφού απευθύνονται κυρίως σε σημερινούς ομογενείς επεξεργαστές με συνεκτικές κρυφές μνήμες. Οι υλοποιήσεις των συστημάτων χρόνου εκτέλεσης που τα συνοδεύουν φαίνονται να μην είναι ούτε κλιμακώσιμες, ούτε κατάλληλες για ετερογενείς και λιγότερο συνεκτικές αρχιτεκτονικές.

Η εργασία μας εξερευνά τις προκλήσεις στον παράλληλο προγραμματισμό της επόμενης δεκαετίας. Ασχολούμαστε με δύο προβλήματα. Πρώτον, πώς πρέπει ένα σύστημα χρόνου εκτέλεσης να σχεδιαστεί, ώστε να κλιμακώνει σε

υπερπολυπύρηνους επεξεργαστές που θα είναι διαθέσιμοι σε δέκα χρόνια; Και δεύτερον, πώς μπορούμε να υλοποιήσουμε και να αξιολογήσουμε τέτοια συστήματα, αφού σήμερα δε διαθέτουμε τέτοιους επεξεργαστές;

Σχετικά με το πρώτο πρόβλημα, επαυξάνουμε ένα υπάρχον μοντέλο με εξαρτήσεις ώστε να υποστηρίζει φωλιάζοντα παραλληλισμό και ακανόνιστες δομές δεδομένων με δείκτες. Στη συνέχεια σχεδιάζουμε και υλοποιούμε το Mygmics, ένα σύστημα χρόνου εκτέλεσης που συνοδεύει το προγραμματιστικό αυτό μοντέλο. Το Mygmics είναι ειδικά σχεδιασμένο για μελλοντικούς, ετερογενείς, μη συνεκτικούς επεξεργαστές και κλιμακώνει χρησιμοποιώντας καινοτόμους, καταναμημένους αλγόριθμους και πολιτικές για ιεραρχική διαχείριση μνήμης, ανάλυση εξαρτήσεων και δρομολόγηση διεργασιών. Τα πειράματά μας αποκαλύπτουν ότι η κλιμακωσιμότητα του Mygmics είναι συγκριτικά καλή σε σχέση με πρότυπους, βελτιστοποιημένους κώδικες σε MPI, ενώ οι επιβαρύνσεις της αυτόματης παραλληλοποίησης παραμένουν αρκετά χαμηλές (10–30%). Επιβεβαιώνουμε ότι πολλοί από τους αλγόριθμους και πολιτικές που προτείνουμε είναι ελπιδοφόροι.

Σχετικά με το δεύτερο πρόβλημα, δημιουργούμε ένα ετερογενές πρωτότυπο σε FPGA με 520 πυρήνες, που μοντελοποιεί πιστά υπερπολυπύρηνους επεξεργαστές σύμφωνα με τις τρέχουσες προβλέψεις. Το χρησιμοποιούμε για να αξιολογήσουμε το Mygmics. Το πρωτότυπο βασίζεται στο Formic, ένα νέο τυπωμένο κύκλωμα που σχεδιάζουμε ειδικά για μεγάλες κατασκευές. Εκτιμούμε ότι το πρωτότυπό μας τρέχει κώδικα 50,000 φορές γρηγορότερα από προσομοιωτές σε λογισμικό για παρόμοιο αριθμό πυρήνων.

Επόπτης Διδακτορικής Διατριβής: Καθηγητής, Άγγελος Μπίλας

ABSTRACT

The end of the processor performance race in the start of the current century signaled the beginning of the multicore era. To harness the benefits of multiple CPU cores for a single application, programmers must now use parallel programming models. Semiconductor trends hint that processors within the next decade will manage to integrate hundreds of cores on a single chip; the architecture will be heterogeneous, with few strong (and power-hungry) cores and many weak (and power-efficient) ones; caches will be less or not at all coherent.

As the new manycore era approaches, finding a productive and efficient programming model able to scale on such architectures is a major challenge. Dependency-aware, task-based programming models have gained a significant following. The programmer provides a serial program, split into small functions (tasks) that run to completion, along with information on what data the tasks will operate. A runtime system analyzes the dependencies, schedules and executes the tasks in parallel. Despite their increasing popularity, these programming models are not ready to scale to emerging manycore architectures, as they primarily target today's homogeneous,

cache-coherent multicores. Their runtime implementations appear to be neither scalable, nor suitable for heterogeneous, less coherent architectures. Our thesis delves into the parallel programming challenges that lie ahead in the coming decade. We consider two major problems. First, how should a parallel runtime system be designed, in order to be able to scale well on a manycore processor ten years from now? And second, how can we implement and evaluate such runtime system designs, since such manycore processors are not currently available?

Towards the first problem, we enhance an existing task-based model to support nested parallelism and pointer-based, irregular data structures. We then design and implement Myrmics, a runtime system that implements this programming model. Myrmics is specifically designed to run on future, heterogeneous, non-coherent processors and to scale well using novel, distributed algorithms and policies for hierarchical memory management, dependency analysis and task scheduling. Our experimental results reveal that Myrmics scales well compared to reference, hand-tuned MPI baselines, while automatic parallelization overheads remain modestly low (10–30%). We verify that many of our proposed algorithms and policies are promising.

Towards the second problem, we create an heterogeneous 520-core FPGA prototype modeled faithfully after current predictions for manycore processors. We use it to evaluate the Myrmics runtime system. The FPGA prototype is based on Formic, a new printed circuit board we design specifically for scalable systems. We estimate that our prototype runs code 50,000 faster than software simulators for similar core counts.

Supervisor: Professor Angelos Bilas