ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΑΡΟΥΣΙΑΣΗ / ΕΞΕΤΑΣΗ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Γλένης Απόστολος

Μεταπτυχιακός Φοιτητής

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Επόπτης Μεταπτ. Εργασίας: Καθηγητής, Μ. Κατεβαίνης

Δευτέρα, 6 Απριλίου 2015, 11:30

Αίθουσα Ε313, Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

" FT-Myrmics: Ένα σύστημα χρόνου εκτέλεσης με ανεκτικότητα σε σφάλματα για task-based προγραμματιστικά μοντέλα "

ΠΕΡΙΛΗΨΗ

Καθώς ο αριθμός των πυρήνων των μοντέρνων αρχιτεκτονικών αυξάνεται, αυξάνεται και η πιθανότητα ένας από αυτούς να παρουσιάσει κάποιο σφάλμα. Επιπλέον επειδή αυξάνει ο αριθμός πυρήνων του συστήματος αυξάνεται εμφανίζονται πρωτοποριακές αρχιτεκτονικές που δεν διαθέτουν μηχανισμούς συνοχής ανάμεσα στην κύρια και την κρυφή μνήμη, προκειμένου να μειώσουν το κόστος του συγχρονισμού. Τέλος ο μεγάλος αριθμός πυρήνων καθιστά υποχρεωτική τη χρήση του παράλληλου προγραμματισμού προκειμένου να χρησιμοποιηθούν πλήρως οι πόροι του συστήματος. Για τη διευκόλυνση του παράλληλου προγραμματισμού έχουν προταθεί αρκετά task-based μοντέλα προγραμματισμού. Το βασικό πλεονέκτημα των task-based μοντέλων προγραμματισμού είναι ότι επιτρέπουν στον προγραμματιστή να χωρίσει το πρόγραμμα τους σε ξεχωριστές εργασίες και να ορίσει μια ροή δεδομένων ανάμεσα σε αυτές. Το σύστημα χρόνου εκτέλεσης Myrmics επεκτείνει αυτή την ιδέα παραπέρα επιτρέποντας στον

προγραμματιστή να ορίσει τα όρια της μνήμης κάθε διεργασίας και χρησιμοποιώντας μια μέθοδο ανάλυσης εξαρτήσεων να παραλληλίσει αυτόματα τον υπολογισμό.

Σε αυτή τη μεταπτυχιακή διατριβή, παρουσιάζουμε το FT-Myrmics, μια επέκταση του Myrmics με υποστήριξη για αυτόματη ανοχή σε σφάλματα. Στο FT-Myrmics παρέχουμε διαφανή ανοχή σε σφάλματα., για μη-μόνιμα σφάλματα. Σαν αρχικό δείγμα επίδοσης υλοποιήσαμε το πλήρες checkpointing πάνω από το myrmics.

Πέρα από το πλήρες checkpointing εκμεταλλευόμαστε τα καλώς ορισμένα όρια μνήμης που παρέχει το Myrmics έτσι ώστε να ελαττώσουμε το απαραίτητο checkpointing. Αφού γνωρίζουμε εκ των προτέρων το ακριβές μέγεθος και τον τύπο των ορισμάτων κάθε διεργασίας μπορούμε να αποφύγουμε την αποθήκευση δεδομένων που μόνο θα διαβαστούν καθώς μπορούμε να διασφαλίσουμε ότι δεν θα γραφτούν μέσω βοηθείας από ειδικό κομμάτι hardware.

Αξιολογήσαμε το FT-Myrmics σε ένα ενδεικτικό σύνολο από μετροπρογράμματα χρησιμοποιώντας το Formic, έναν εξομοιωτή ενός επεξεργαστή 512 πυρήνων. Βρήκαμε ότι το checkpointing δημιουργεί μείωση της επίδοσης μεταξύ 1.1χ και 5χ ανάλογα με το μέγεθος του checkpoint.

Glenis Apostolos

M.Sc. Thesis

Computer Science Department

University of Crete

Master's Thesis Supervisor: Professor M. Katevenis

Monday, 06/04/2015, 11:30

Room E313, Computer Science dept., University of Crete

“FT-Myrmics : A fault tolerant runtime system for task based programming models”

ABSTRACT

As the core count of modern architectures increases, so does the probability that an individual processor of the system manifests some kind of error. Moreover as core count increases novel architectures that lack traditional cache-coherence mechanisms emerge to mitigate synchronization overheads.

Finally high core count dictate the use of parallel programming in order to take advantage of all the resources of the system. To facilitate parallel programming several task-based programming models have appeared. The main advantage of task-based programming models it that they allow the programmer to split his program into tasks and define a dataflow of operations. The Myrmics runtime system takes this approach a step further and by defining task memory footprints and using dependency analysis is able to automatically parallelize the computation automatically.

In this thesis, we present FT-myrmics an extension of the Myrmics runtime system for automatic fault tolerance. In FT-myrmics, we provide transparent fault tolerance against transient errors. As a baseline for performance, we implemented a full checkpointing solution on top of myrmics.

Apart from using full checkpointing, we take advantage of the well-defined task boundaries provided by the Myrmics programming model to avoid unnecessary checkpointing. Since we know the exact memory footprint and the type of each argument, we can avoid checkpointing read-only arguments by enforcing that they cannot be overwritten, using hardware assistance.

We evaluate the fault-tolerant FT-Myrmics system on a representative set of benchmarks, using the Formic prototype 512-core processor emulator. We find the performance overhead of distributed task checkpointing to cause slowdowns between 1.1x and 5x, depending on the size of the checkpoint.