In-Processor Message Passing

astanesc

1 Website

http://www.andrew.cmu.edu/user/astanesc/processorgc.html

2 Description

2.1 Who

This project will be performed under the watchful eye of Nathan Beckman (beckmann@cs.cmu.edu), as well Elliot Lockerman (elockerm@cs.cmu.edu).

2.2 What

This project will involve creating a message passing system built into the memory management system of a large multicore processor. Specifically, whenever a certain process needs to keep a message queue, that message queue is kept within the local cache for that processor, and when a message is sent to that processor, the information about the message is sent via hardware busses within one cache line).

2.3 So what (or why?)

The motivation behind this project is that there is a lot of inefficiency within message passing due to "pointer chasing". "Pointer chasing" is a term that describes when a program has to follow a long chain of pointers (or a short chain many times) in order to accomplish a process. Furthermore, message passing is a heavily used paradigm in modern multiprocessor systems that rely heavily on parallelism. By improving the efficiency of a fundamental building block of these systems, we can drastically improve the efficiency of the systems themselves.

3 Project Goals

75% : Memory management system is able to handle a simple single reader, single writer queue
100% : Memory management system is able to handle a single reader, multiple writer queue.
125% : Memory management system is able to handle a many reader, many writer queue
4 Milestones

4.1 First Technical Milestone
Identify exactly what hardware is needed, and have that hardware prepared so that work on upgrading the memory management system can begin. Furthermore, be well versed in what exactly building a garbage collector entails.

4.2 15-400
February 1st: Zsim memory management system operation
February 15th: AVL Tree memory management implementation is functional and demonstrably faster than traditional code
March 1st: Single reader-writer queue (with one cache line) implemented and tested through zsim
March 22nd: Support for dynamically sized queues integrated into zsim
April 5th: Support for multiple writers for the queue integrated into zsim
April 19th: Producing workloads for the queue system for testing done
May 3rd: Testing complete and queue finished

5 Literature Search
Using C++ textbooks to learn C++ as well as reading up on memory management systems

6 Resources Needed
None