In-Processor Garbage Collection

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 as one of his graduate students (who I have not been introduced to yet)

2.2 What

This project will involve creating a garbage collector built into the memory management system of a large multicore processor. Specifically, rather than a language environment keeping track of what sections of memory are allocated (and what sections can be garbage collected), the runtime would have access to extra commands provided by the system, which would enable it to offload this to the system itself.

2.3 So what (or why?)

The motivation behind this project is that there is a lot of inefficiency in modern garbage collectors, as they have to chase pointers, which requires extra time as information has to be sent back and forth across (potentially) large spaces of within processor. Furthermore, any requests to perform garbage collection are going to be inefficient as they require the processor to interrupt what it’s doing and check pointers. If the garbage collector is built into the memory management system of the processor, then the garbage collection can happen on a separate dedicated portion of the chip, which would allow the regular processor to continue on its merry way. Thus, we hope to improve the efficiency of garbage collected languages like C++ and JAVA, as well as provide a method to add garbage collection into languages like C for ”free”.

3 Project Goals

75% : Memory management system is able to do garbage collection on its own.
100% : Language compiler for a garbage collected language (such as C++) has been updated to make use of memory management system.
125% : Multiple languages now have updated compilers that can use this system.
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: Hardware is prepared for upgrading
February 15th: Work on garbage collector should have begun
March 1st: Garbage collector is finished
March 22nd: Garbage collector is tested, instruction set should be begun
April 5th: Instruction set is updated and garbage collector is complete
April 19th: Garbage collector is finished and tested, Should have identified all regions of the C++ compiler that deal with garbage collection
May 3rd: C++ Compiler upgraded

5 Literature Search
I have been looking at books and articles on garbage collection, as well as memory management systems to fully understand how to proceed

6 Resources Needed
Specialized processors are needed