Data Access Profiling in Pegasus and Improved Structure Field Regrouping
15-745 Project Proposal
March 22 2005

1 Group Information
Matt Moore <mlm1@cs.cmu.edu>
Vas Chellappa <schellap@ece.cmu.edu>

2 Project Web Page
http://cs.cmu.edu/user/mlm1/745

3 Project Description
Currently the Pegasus simulator can perform some basic profiling, and supports simulation of a full memory system. We would work on adding memory profiling to the Pegasus simulator for the purposes of producing a field affinity graph for structures. Determining both accurate cache performance and obtaining data about what items are accessed most frequently would be required to accomplish this goal.

Using this profiling information, we will attempt to perform the field reordering recommendations mentioned in Chilimbi et. al.'s paper given in Section 4.3. However, bbcache, mentioned in the given paper, used a greedy algorithm for translating the field affinity graph (which relates the contemporaneity of fields) into a structure grouping. Using a trivial example, we can show that the greedy approach to using the field affinity graph doesn’t produce optimal grouping. We would investigate the plausibility of implementing a version of the grouper that produced optimal grouping of fields, or try and find a reduction to show the problem to be NP-hard.

4 Logistics
4.1 Plan of Attack and Schedule
Week 1 :
- Find out more about Pegasus
- Importantly: Find out if Pegasus can figure out whether, given a memory address, can determine if this belongs to a structure (and if so, which one), or not. Potential major issue, so find out about this first. Talk to people Seth recommended.
- Find out about how the cache simulation is done in Pegasus
- Run the simulator, get comfortable with using it, and comfortable with the source code enough to identify parts that interest us.
- Use the new simulator (recommended by Seth)
- The regrouping problem might or might not be NP-hard. Determine this if possible. Find out optimal algorithm or good heuristic as is the case.

Week 2 :
- Come up with a preliminary design of how to go about extracting the information we need: Are we going to use an external cache sim, does pegasus already do so, do we need to modify either pegasus or the external cache sim etc.
• Start implementing the algorithm to recommend reordering.

Week 3:

• Begin main implementation: Running a program, generating traces, analyzing these traces. Output should be a weighted graph where nodes=struct fields, edges represent degree of contemporaneity of access.

• Finish implementing the algorithm to recommend reordering.

Week 4:

• Finish main implementation: Running a program, generating traces, analyzing these traces. Output should be a weighted graph where nodes=struct fields, edges represent degree of contemporaneity of access.

Week 5:

• If possible, reorder the fields in the source, and resimulate with reordered struct to find out performance (execution time/miss rate) improvement.

Week 6:

• Catch up on slack, wrap up, write report

4.2 Milestone

Our project milestone, set at the end of week 3 is to have our design of the profiling system completed, with significant progress on implementation; and determination of whether the regrouping problem is NP-hard, and complete implementation of the regrouping problem, based either on a heuristic or an optimal solution, as appropriate.

4.3 Literature Search

Our project mainly extends the capabilities of the Pegasus simulator. It is also partially based on Chilimbi et. al.’s publication:


4.4 Resources Needed

Since our project will be done mainly on the Pegasus simulator and is hardware independent, all we expect to need are the Pegasus simulator with source, and an appropriate benchmark suite (with source code) which will be determined later.

4.5 Getting Started

At this point, we have gone through an initial examination of the regrouping problem, and have demonstrated that a greedy approach is sub-optimal. We also have a slightly improved approach, which essentially illustrates the scope for an improved solution. We have also established some points of contact within the CASH group for discussion of the Pegasus simulator’s handling of the memory system.