

Parallel Network-RAM Project Status Report October 15th, 2003

John Oleszkiewicz

Time Issues

Problem 1 – It is unlikely I will get my TA hours alleviated

- In order for this to happen, the following must occur:
 - My hours will have to be cut to $\frac{1}{4}$ time.
 - Someone else will have to take my place.
 - This other person must be capable of doing the job and have the time to do it (CSE 260 grading is time intensive)
 - To fund this new person, more than \$2,000 will have to be spent.
 - In addition, I will need more than $\frac{1}{4}$ time pay – more money spent.

Problem 2 – It is unlikely I will have much time to work on this problem during the winter break

- I have several commitments this winter break that I can not fulfill at any other time. These include:
 - Preparing for my wedding
 - Beginning the search for a job
 - Visiting relatives for the holidays
- Whatever time I have left over after these commitments could possibly be spent working on the project

Time Available

- Assuming no change in TA hours and no work done during the winter break, I have the following amount of time available before February:
 - October, November, December
 - Approximately 8 weeks
 - Can possibly work 10 hours per week
 - January
 - Approximately 4 weeks
 - 40 hours+ per week

Tasks

The following are tasks that must be completed before February if we are to publish a paper. They are arranged roughly in order of difficulty:

Implement a realistic network-RAM model

- Current centralized model is not very realistic – requires prior knowledge of process memory requirements
- Suggested replacement – distributed algorithm
 - Server sits on each PE (CPU)
 - As memory is overallocated, the server will allocate network RAM as appropriate
 - This will require design and implementation work
 - Little design work has been done for a distributed manager
 - Little implementation been done in this direction
 - This will also require the most feedback so that design and implementation do not go in the wrong direction.

Replace current memory access pattern generator with faster one

- Memory access pattern generator is bottleneck of simulator
 - Current system: at every memory reference generated by process, check page number and determine if reference results in page fault or not.
 - Memory references generated every few nanoseconds of simulated time.
 - Many processes run for thousands of seconds
 - New proposed system: determine time until next page fault based on a probability distribution (possibly exponential) that is influenced by amount of memory overallocated and amount paged in already.
 - How to validate model created for realism? References in the literature to something like this?
- Current version of the simulator is limited to proofs-of-concept only because of speed.
- Once replaced, we should be able to run real trace data.

Fix small inaccuracies and assumptions in paging model

- Small inaccuracies in the implementation of the paging model have been cataloged and need to be fixed.
- Not difficult – just time-consuming

Investigate communication patterns of CFD application and use data as a parameter to the simulator

- Investigation already a part of a term project in another class.
- Should be able to apply trace data from term project directly to our simulation.

Write the paper

- Some writing work already done this summer.
- Writing work started includes
 - Introduction to the problem
 - Introduction to background material/related work
 - Information about the simulator and the models it uses
- Many references gathered on parallel job schedulers.

Proposed Timeline

This proposed timeline assumes no change in TA workload and no time spent during winter break on the project.

If setbacks occur, it will be possible to shift some tasks into January, since more time is available to work then.

October

Design and specify the distributed network RAM algorithm.
Investigate CFD application communication pattern.

November

Implement the distributed network RAM algorithm.
Replace memory access pattern generator.

December

Fix paging model inaccuracies.

Early January

Finish up changes, start debugging work.

Late January

Run simulations and gather results.
Write paper.

February

Finish paper.
Submit paper.