

COMP/ELEC 526: Project Description

Vijay S. Pai

Fall 2003

1 Academic Information

Group signup. September 24. Signup in class, confirm by email.

Proposal. October 6, 1:00 PM. Hard copy, 1-2 pages. 5% of semester grade.

Oral Progress Report. November 3, in class. 10 minutes. 5% of semester grade.

Project Talk. To be scheduled. Tentatively last week of class. 30 minutes. 10% of semester grade.

Project Report. Officially due last day of class. Hard copy, detailed. 20% of semester grade.

2 Project Description

The project is to do some original research in parallel computers or to revalidate data found in a published paper under a different set of test conditions (e.g., different applications or system configurations). Note that this is not simply library-style research or a literature survey; the project must provide some interesting contribution to the field. You will be graded on how well you define your problem, survey previous work, design and conduct experiments, and present your results. The project should be done in groups of two or three.

3 Group

You should decide on your partners by September 24, when there will be a signup sheet in class. You will be spending a lot of time with your partners and all members of a group will get the same grade, so choose your partners carefully.

4 Proposal

I strongly encourage you to talk to me before submitting the proposal, as all project topics must be approved by me. Ideally, there should be no surprises in your proposal, as your earlier discussion with me should have helped you avoid proposing an unacceptable project. The proposal should be about 1-2 pages long and should include the following.

1. A description of what you are going to study and why this study is interesting or important.
2. A brief summary of previous work in the chosen area and how it relates to your project. You should reference at least three technical papers. The text book points to many papers. Common sources for such papers are the Proceedings of the International Symposium on Computer Architecture (ISCA), Proceedings of the conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Proceedings of the International Symposium on High Performance Computer Architecture (HPCA), Proceedings of the International Symposium on Microarchitecture (MICRO), Proceedings of the International Conference on Parallel Processing, Proceedings of the International Conference on Supercomputing, Supercomputing, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, ACM Transactions on Computer Systems, IEEE Computer. All Rice students have access to many of these publications online through the ACM Digital Library and IEEE Digital Library. Many of these publications are also available for reference at Fondren Library. I encourage you to talk to me for additional suggestions.

3. A description of how you plan to achieve your goals (e.g., what methods you will use to evaluate your ideas) and a schedule of when you expect to complete the various sub-tasks. This is the most important part of the proposal. Be as specific as you can. This part will allow you and me to assess the success probability of your project early, when radical changes can still be made.

5 Oral Progress Report

Each group will give a 10-minute progress report in class on November 3. This would be a good time to introduce peers to your topic, discuss changes to goals or methods, current progress, accomplishments so far, etc. This is a good time to get feedback and take some early questions from peers. If you would like to make some brief reading material on your project available to your peers and instructor before this, you may send a URL out on the class mailing list.

Please try to schedule a large fraction of your computational experiments before the oral progress report, since educational computing resources tend to be stressed in the last few weeks of class.

6 Project Talks

Project presentations will take place during the last week of class. Each group will give a conference-style talk of approximately 25 minutes (with length to be adjusted according to the schedule), with roughly 5 minutes for questions at the end of the talk. *All group members should deliver part of the talk.* The talk should give *highlights* of the final report, including the problem, motivation, results, conclusions, and possible future work. Do not try to cram everything you have done into the allotted time. Time limits will be announced in advance and will be strictly enforced so that all groups have a chance to present. *It is very important for you to practice your talk several times before you give it to the class.* Practicing is the best way to estimate the length of the talk and to use your time most effectively. Please see the links on the web page for information on how to give a bad talk, as well as how to prepare a good conference talk.

7 Final Report

Reports should consist of an abstract, body, and optional appendices, much like a conference paper. The abstract should summarize the contributions of the report in one or two paragraphs. The body should be formatted double spaced with 12 point font and should not exceed 6000 words. The body of a typical paper will contain sections on introduction, related work, one or more sections explaining the problem studied, experimental methodology, results, conclusions, and bibliography of works cited. Figures and illustrations are highly encouraged and will not count against any length limitations. Additional supporting material of any length can be put in appendices. I will read the body and only skim appendices. Excessively long documents will be penalized.

I expect a well-written document. Remember the best research is not much use if it is not communicated properly. Additionally, it is essential to cite prior work when using or expanding upon results or ideas that have been previously published. If you are not familiar with common citation formats, please contact the instructor.

8 Resources Available

The following describes some of the available resources. This is not an exhaustive list, so talk to me if you need something not on the list.

- Rice Simulator for ILP Multiprocessors (RSIM): This is an execution-driven simulator for modeling shared-memory multiprocessors built out of processors that aggressively exploit instruction-level parallelism (e.g., multiple issue, dynamic scheduling, speculative execution).
- Two four-processor UltraSPARC-II systems (primarily for research use in Rice Computer Architecture group, but may be available at times)
- Treadmarks (a software distributed shared memory system) for shared memory on a network of workstations.

- SPLASH and SPLASH-2 program suites, parallel versions of some NAS benchmarks, a few parallel benchmarks from the MIT Alewife group, a suite from the University of Washington, and several locally developed parallel benchmarks.

9 Some Possible Topics

Here are some possible topics you can work on. You may also work on topics not on this list.

- Explore parallel computing and parallel architectural tradeoffs in systems built of non-commodity processors. For example, many systems for networking, communication, and other such tasks exploit fine-grained parallelism in certain classes of applications.
- While traditional large-scale multiprocessors have used directory coherence protocols and CC-NUMA structures, Sun has built several large SMP systems that use unconventional snooping mechanisms. For example, the Ultra Enterprise 10000 scales to 64 UltraSPARC-2 processors using multiple address busses and a crossbar for the data network. The Fireplane (discussed in an SC 2001 paper by Charlesworth) scales to 24 UltraSPARC-3 processors using a tree-structured address broadcast and a data crossbar. Compare a system with an unconventional snooping mechanism against a directory protocol, exploring the tradeoffs in performance, programmability, and implementation cost. RSIM already provides simulation models for directory-based systems; you will need to implement a snooping system in RSIM for the comparison.
- In roughly 7 years, a single chip will be able to support a billion logic transistors. An important question that faces architects is how to utilize these transistors. One proposal is to build multiple processors on a chip, to yield a single chip multiprocessor. The design and performance of such a system may be different from a multi-chip multiprocessor. Explore reasonable designs for such a system (see Nayfeh et al. ISCA'94, Olukotun et al. ASPLOS'96, Huh et al. PACT 2001), and contrast the factors that affect performance in a single-chip and multiple-chip system.
- Many current uniprocessors support very large cache line sizes. Use of such large cache line sizes can potentially result in substantial false sharing in multiprocessors, resulting in lower performance. The use of software-based protocols can allow mitigating the effects of false sharing. For example, dirty bits per word may be maintained in software and the transfer block size may be controlled in software. Propose a scheme to reduce the effects of false sharing, and study its performance.
- Evaluate one or more latency tolerating/reducing schemes; e.g., hardware prefetching, multithreading, software prefetching, relaxed consistency models, producer-initiated communication.
- Compare various hardware cache coherence protocols, possibly including new designs that support such techniques as Reactive NUMA (Falsafi and Wood, HPCA '97), coherence prediction (Mukherjee et al, ISCA '98), multicast snooping (Bilir et al, ISCA '99), token coherence (Martin et al, ISCA '03), etc.
- Measure and analyze the performance of a real shared-memory multiprocessor on different applications using bulk measures and hardware performance counters.
- Computer architecture work relies on, and is often enabled by, good evaluation tools. RSIM is one such tool. The advantage of RSIM is that it models a lot of detail in the system. Shortcomings include slow speed, lack of sophisticated support to visualize performance measures, and lack of support for general operating system calls. Several interesting projects can be designed to address the above shortcomings of RSIM. One such project is to parallelize RSIM to improve its speed. Another project involves investigating the technique of sampling for fast performance evaluation.
- A paper in ISCA'95 characterizes the programs in the SPLASH-2 application suite. This is a detailed characterization, describing the concurrency and load balancing characteristics, the working sets, communication-to-computation ratio, data sharing patterns, and spatial locality of the applications. This characterization is extremely important for researchers using the applications to evaluate architectural issues. In particular, when

using a simulator, one can typically simulate only small problem sizes. It is difficult to determine what architectural parameters to use so that they duplicate the performance on systems with realistic parameters using realistic problem sizes. For example, what cache sizes are appropriate to assume with the smaller problem sizes? Is it appropriate to not model contention in the network? The above characterizations help answer the above questions. Your project would be to perform similar characterizations for other applications; e.g., networking or multimedia applications.

- Experimentally extend any parallel architecture research paper from any recent major conference.
- Study any of the above issues analytically by developing formal analytical performance models. This most likely requires knowledge from ELEC 428 or 429.

10 Honor code policy

As discussed above, this is a group project with groups of size 2-3. Group members may split their work as they see fit, but all grades will be assigned collectively. All related work used in preparation of the project report should be explicitly cited in its appropriate context, and plagiarism of text is strictly forbidden. Any group work suspected of violation will be reported to the Honor Council.