6.895 Theory of Parallel Systems	September 8, 2003

Project Proposal
New Cache-Oblivious Algorithms

Group Members: Zhang Jiahui and Neel Kamal

1 Background

In our project, we try to propose several cache oblivious algorithms that use cache effectively. These algorithms cover a wide area from "Large Integer Arithmetic" to "Dynamic Programming" and from "Computational Geometry" to "Graph Algorithms". There are exiting algorithms for these problems, but when the "size of the problem" becomes larger, traditional versions will encounter more and more cache misses, and therefore, degrade the performance tremendously. We believe with our new cache oblivious versions, performance will be largely improved with less cache misses, and there is no need to turn any parameters, since they are cache OBLIVIOUS.

2 Project Goals

In this project, we plan to achieve the following goals:

1. Design the Cache Oblivious algorithm for "Large Integer" arithmetic as well as their applications, such as the famous RSA encryption algorithm, matrix multiplication of large integers, and etc. Also, we may extend it to deal with high precision floating point numbers.
2. Design the Cache Oblivious versions of some "Dynamic Programming"problems, such as "Longest Common Sequence".
3. Design the Cache Oblivious versions of some "Computational Geometry" problems. (Detailed topics will be decided later.)
4. Design the Cache Oblivious versions of some "Graph Algorithm", such as "Floyd algorithm for pair-wise shortest path", "Connectivity of a Graph", and etc.
5. We shall design the above algorithms with theoretical analysis of their upper and lower bonds of Cache misses.
6. Also, we plan to come out with the multithreaded version of the above algorithms leading to parallelization.

3 What We Propose

We propose cache oblivious versions of some widely used algorithms with theoretical performance analysis. We also propose multi threaded variations of each algorithm and hence come out with a good parallel version.

4 Justifying the load

It is a 2-student-group based project, and we believe the load we propose is neither to heavy nor too light. Since we will explore several areas of computer science, come out with a number of cache oblivious algorithms for various popular problems, conduct theoretical analysis of the performance (bonds of cache misses) of our proposed ones, and finally parallelize them. Of course we are not doing something totally new, but our proposed cache oblivious algorithms are all original. One member of the group may be in charge of design and analysis of the cache oblivious algorithms, and the other member may help in the analysis and extend them into parallel versions.

5 Backup plan

I think we may have some difficulties when doing the theoretical analysis of the performance in terms of cache misses for cache-oblivious versions of some complex algorithms in computational geometry or graph algorithms. So what we are going to do is that we will try our best first, and in case we really can not figure out, we may pick some not-so-difficult algorithms in these areas to make them cache oblivious.

6 References

- Cache-Oblivious Algorithms by Matteo Frigo, Charles E. Leiserson, Harald Prokop, and Sridhar Ramachandran. In the 40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA.

- Cache-Oblivious Algorithms by Harald Prokop. Master's Thesis, MIT Department of Electrical Engineering and Computer Science. June 1999.

- Minicourse on Multithreaded Programming by Charles E. Leiserson and Harald Prokop. July 1998.

- Introduction to algorithms By Thomas H. Corman, Chatles E. Leisersion, Ronald L. Ricest, Clifford Stein.