

Adding and Evaluating Data Prefetching Strategies in JIAJIA

Group Member:

Xinghua An (anxh@cs.cmu.edu)

Ting Liu (tiv@andrew.cmu.edu)

Project Description:

Over the past decade, software Distributed Shared Memory systems have been extensively studied to provide a good compromise between programmability of shared memory multiprocessors and hardware simplicity of message-passing multicomputers. However, software DSMs suffer from the high communication and coherence-induced overheads caused by the high-level implementation and large granularity of coherence. So a variety of mechanisms such as multiple-write protocol, lazy lease consistency, and partial hardware support, have been proposed to minimize remote communication and hide communication latency.

JIAJIA [1] is software DSM with a lock-based cache coherency protocol for scope consistency [2]. Performance measurements with widely-accepted benchmarks such as SPLASH2 program suite and NAS Parallel Benchmark indicate that JIAJIA reaches higher speedup and absolute execution time compared to other successful software DSMs such as CVM. This should be contributed to its simplicity and efficiency of lock-based cache coherency protocol and its distinctive feature of home-based memory organization which requires no diffs generating for home pages.

Despite those great features, we unfortunately found that there were still something unsatisfiable in JIAJIA. For example, MM or FFT needs to read a matrix in both row and column order. Suppose that the whole matrix is stored uniformly in all computers of a local network in row order. So the process on one computer can access a row of data without remote accessing operation. But it has to wait for data stored in the memory of other computers when it tries to read a column of data. This unsymmetry of accessing time is caused by the unsymmetry of data distribution. As we know, overlapping computations and communications is an efficient way (without rearranging data distribution in all computers) to reduce communication overhead and thus avoid unnecessary wait for remote data. Unfortunately, we didn't find those well-known software DSMs (i.e. CVM, SHRIMP and ADSM) support this feature. So we plan to add prefetching caches in JIAJIA and then evaluate how this technique hides memory latency by exploiting the overlap of processor computations with data accesses. Although introducing data prefetching will incur new complexity in cache coherency protocol and then might totally diminish its advantage, it's not a serious problem in JIAJIA. Because JIAJIA is a scope-consistency-based software DSM, the maintenance occurs only in the boundary of two scopes (collective function `jia_barrier()` in JIAJIA). This desirable feature will greatly simplify our new prefetching-supported cache coherency protocol, which is a key part in our work.

And another important work we should do is to adopt proper data prefetching strategy [3][4] for JIAJIA. Since software DSMs are widely used by scientific applications which read data in some particular patterns such as *sequential* and *stride*, we aim our prefetching algorithm on improve the efficiency of scientific applications.

Plan of Attack:

Phase I *Understanding the project:*

Scratch related papers in this area (including memory consistency model, cache coherency protocols in software DSMs, and data prefetching strategies) to get a deeper understanding of those concepts. Look inside the source code of JIAJIA and try to do some simple modifications.

Phase II *Implementation:*

Propose the new cache coherency protocol for scope consistency model supporting data prefetching, select correct data prefetching strategy and then implement it on JIAJIA.

Phase III *Testing and evaluation:*

Use benchmark programs such as SPLASH2 and some other scientific applications (MM, FFT or LU) to test the performance of JIAJIA with support of data prefetching. Evaluate the experiment result.

Phase IV *Conclusion:*

Draw conclusions. Finish the project report.

Schedule:

DATE	ANTICIPATED ACHIEVEMENTS
10/20 – 10/26	Read related papers and get familiar with tools
10/27 – 10/30	Look inside source code of JIAJIA
10/31 – 11/15	Implement data prefetching strategy.
11/16 – 11/25	Modify cache coherency protocol
11/26 – 11/27	Write test programs (MM, FFT)
11/28 – 12/5	Test our system
12/5 – 12/10	Write project report and poster

Milestone:

The design and implementation should be done before Nov.26th.

Literature Search:

1. Weiwu Hu, Weisong Shi, Zhimin Tang and Ming Li. A Lock-based Cache Coherence for Scope Consistency. *Journal of Computer Science and Technology*, Vol 13, No. 2, pp. 97-110, 1998.
2. Weiwu Hu, Weisong Shi, and Zhimin Tang. A Framework for Memory Consistency Models. *Journal of Computer Science and Technology*, Vol 13, No.2, pp. 110-124, March, 1998.
3. S.P. Vanderwiel and D. J. Lilja, Data Prefetch Mechanisms, *ACM Computing Survey*, vol.32, no. 2, pp. 174 C 199. Jun 2000.
4. J. W. C. Fu and J. H. Patel, Stride directed prefetching in scalar processors. In *Proceedings of the 25th International Symposium on Microarchitecture*, pp. 102-110, 1992.

(more papers are not listed)

Resources Needed:

Testing benchmark programs.

JIAJIA source code (available at <http://www.ict.ac.cn/chpc/dsm/dist.html>)

Getting Started:

We have already read papers on memory consistency models, cache coherency protocols and data prefetching, and we got some preliminary ideas about the design and implementation.