

Abstract

Distributed shared memory (DSM) machines are becoming an increasingly popular way to increase parallelism beyond the limits of bus-based symmetric multiprocessors (SMPs). The cache coherence protocol is an integral component of the memory system of these DSM machines, yet the choice of cache coherence protocol is often made based on implementation ease rather than performance. Most DSM machines run a fixed protocol, encoded in hardware finite state machines, so trying to compare the performance of different protocols involves comparing performance across different machines. Unfortunately, this approach is doomed since differences in machine architecture or other design artifacts can obfuscate the protocol comparison.

The Stanford FLASH (FLexible Architecture for SHared memory) multiprocessor provides an environment for running different cache coherence protocols on the same underlying hardware. In the FLASH system, the cache coherence protocols are written in software that runs on a programmable controller specialized to efficiently run protocol code sequences. Within this environment it is possible to hold everything else constant and change only the cache coherence protocol code that the controller is running, thereby making visible the impact that the protocol has on overall system performance.

This dissertation examines the performance of four full-fledged cache coherence protocols for the Stanford FLASH multiprocessor at varying machine sizes from 1 to 128 processors. The first protocol is a simple bit-vector protocol which degrades into a coarse-vector protocol for machine sizes greater than 48 processors. The second protocol is the dynamic pointer allocation protocol, which maintains the directory information in a linked list rather than a bit-vector. The third protocol is the IEEE standard Scalable Coherent Interface (SCI) protocol, with some enhancements to improve performance and some extensions so that it can function properly within the FLASH environment. The fourth protocol is a flat Cache Only Memory Architecture (COMA-F) protocol that provides automatic hardware support for replication and migration of data at a cache line granularity.

A framework is presented to discuss the data structures and salient features of these protocols in terms of their memory efficiency, direct protocol overhead, message efficiency, and general protocol scalability. The protocols are then compared when running a mix of scalable scientific applications from the SPLASH-2 application suite at different machine sizes from 1 to 128 processors. In addition to these results, more stress is placed on the memory system by running less-tuned versions of each application, as well as running each application with small processor caches to show how the relative protocol performance can change with different architectural parameters.

The results show that cache coherence protocol performance can be critical in DSM systems, with over 2.5 times performance difference between the best and worst protocol in some configurations. In addition, no single existing protocol always achieves the best overall performance. Surprisingly, the best performing protocol changes with machine size—even within the same application! Further, the best performing protocol changes with application optimization level and with cache size. There are times when each protocol in this study is the best protocol, and there are times when each protocol is the worst.

In the end, the results argue for programmable protocols on scalable machines, or a new and more flexible cache coherence protocol. For designers who want a single architecture to span machine sizes and cache configurations with robust performance across a wide spectrum of applications using existing cache coherence protocols, flexibility in the choice of cache coherence protocol is vital.