Last Modified: 29 April 2014

IS2202 - Computer Systems Architecture

Scalability and Programming

Bonus assignment 3

-Submitted by

Tamilselvan Shanmugam tamsha@kth.se

1. Scalable Multiprocessors

1. NUMA Vs UMA architectures

UMA: All the processors are connected to memory with a common interconnect. UMA has the problem of bus traffic when scaling.

NUMA: Every processor has its own memory bank. Accessing other processors memory is possible via interconnect. NUMA's problem is the varying access time to different parts of memory. Accessing memory of own processor is faster than accessing memory of farthest processor.

Programming in NUMA involves lot of work. In general NUMA gives better performance when it comes to scaling[1].

2. Bus Based Vs Directory Based Coherence protocol

Bus Based: All the memory transactions are broadcasted in the bus, so that all the processors receive the message. Simple to implement.

Directory Based: A dedicated directory is maintained in memory that says which processors share the cache line. Based on the directory information, point to point messages will be sent to that particular cache.

Why Directory Based: In Bus Based protocol, as the number of processor goes up, they want to use the common shared bus for memory transaction which creates bottleneck; other processors may wait longer to get the bus to broadcast. Directory based protocol avoids broadcasting mechanism by sending point to point messages. This is the key to choose directory protocol for scaling[2].

3. Overflow handling

DIRiB: This mechanism can keep track of "i" number of processors that share the cache line. i.e Until "i" number of processors DIRiB mechanism sends point to point message. Beyond the count "i", it sets overflow bit and start broadcasting the message. Works well if the data is shared less or shared heavily[3].

DIRiNB: Similar to DIRiB it can hold "i" number of processors. When overflowing, it evicts any one copy from of the cache form a processor and holds new requested processor information. By doing such, DIRiNB will never send broadcast message.

DIRiCVr: Until "i" number of shares, it sends point to point message. Beyond "i" DIRiCVr groups processors and store their state. If anyone processor from a group has copy, then point to point message is sent to all the group members. Less traffic than DIRiB[3].

4. Data Structure of SCI:

It is Doubly Linked list consisting of "forward" and "backward" links. Traversing through the linked list gives all the cache which has the copy of data.

Where it is stored:

The doubly linked list is stored in the cache of every processor. When a data is brought in or evicted, linked list is modified.

2. Programming multiprocessors

5. Message passing Vs Shared memory

Message passing: communication is established by sending message to other threads. . Sending thread no need to wait for any event.

Receiving thread either wait forever to receive a message or wait until some defined time to read the message.

Shared memory: Memory is shared across the threads. To avoid data corruption, semaphore or synchronizations mechanisms are used to protect the shared memory access. A thread has to wait until it gets the shared memory access.

6. Not shared between threads

Program counter: *Not shared*

Thread should have distinct program counter to execute its own instructions.

Stack pointer: *Not shared*

When context switch happens, thread has to store its present state to resume when it wakes up. So stack pointer should be unique.

File descriptor: *Shared*

A thread can access file opened by other threads. So it can be shared.

Virtual to physical memory mappings: *Shared*

All threads share virtual to physical memory mappings[4].

7. No false sharing

Normally false sharing will not occur in MPI framework. It sends and receives message between processes than caching block of data.

Pthread and OpenMP do caching data in blocks and have chances of false sharing.

8. Architectural problem of Red-Black

Every iteration has to sweep the data twice. This adds the cache miss to double fold.

Increasing in number of core increases communication between the cores that limits the parallelism because of bus bandwidth constrains.

References:

[1] N. Manchanda and K. Anand, "Non-Uniform Memory Access (NUMA) New York University {nm1157, ka804} @cs.nyu.edu."

[2] Samaher Al-Hothali, Safeeullah Soomro, Khurram Tanvir, and Ruchi Tuli, "Snoopy and Directory Based Cache Coherence Protocols: A Critical Analysis," *J. Inf. Commun. Technol.*, vol. Vol. 4, No. 1, (Spring 2010) 01–10.

[3] "High Performance Computer Architecture." [Online]. Available: http://www.cs.nyu.edu/courses/fall04/G22.2243-001/lectures/lect13-4up.pdf. [Accessed: 19-Apr-2014].

[4] "Operating Systems Notes." [Online]. Available: http://www.personal.kent.edu/~rmuhamma/OpSystems/Myos/threads.htm. [Accessed: 20-Apr-2014].