Comparative Architectures Supervision Answer (Set 4)

Wei Song 26/04/2017

This answer provides a summary of key points in each question. It is the understanding of the supervisor, which does not necessarily always correct and does not represent the view of lecturers. It is expected to expand the key points into detailed description when similar questions are asked in exams.

Lecture 12 & 13. Chip multiprocessor

Q1. Why is a shared second-level (L2) cache typically divided into multiple banks (banked) in a chip multiprocessor?

Banked L2 allows multiple simultaneous cache requests accessing different banks.

Q2. A cache controller in a chip multiprocessor snoops the bus and observes a transaction that refers to a block that its cache contains. The block is held in State M (Modified). The bus transaction has been generated by a processor wishing to read the block. Assuming a MSI (write-back invalidate) cache coherence protocol, what actions will be taken by the cache controller?

The cache controller writes this block to memory if it is dirty and provides the data to the processor reading this block. The status of this block changes from M to S.

Q3. What optimisation does the addition of the E state to the MSI protocol provide?

Reduce the request to change state from S to M when a write is needed.

Lecture14. On-chip interconnect

Q4. For what reasons might virtual-channels be added to an on-chip network?

- Increase network throughput.
- Support QoS.
- Avoid deadlocks.

Q5. Why might an adaptive routing algorithm offer better performance than a deterministic one?

Some adaptive routing algorithms may guide traffic to reduce network congestions; therefore, increase the overall network throughput. However, adaptive algorithms do not always outperform deterministic algorithm. This benefit is network and traffic dependent.

Some adaptive algorithms are tolerant to certain amount of faults which improves network reliability.

Past exam questions

Q6. Sequential consistency offers a simple and intuitive memory consistency model. Why is it rarely supported by modern chip-multiprocessors?

Sequential consistency limits the use of a wide range of common performance optimisations used by hardware and compiler designers, e.g. write buffers, overlapping writes, non-blocking reads, compiler optimisations and in many cases introduces additional delays when accessing caches.

Q7. A 4KB, blocking, write-allocate, least-recently-used (LRU) replaced, private L1 cache with 16B lines sees the following sequence of access after reset:

0x00001000 Load
0x00001010 Store
0x00002000 Load
0x00001010 Load
0x00003000 Load
0x00001010 Store
0x00001010 Store
0x00002000 Load
0x00001000 Load
0x00002000 Load

What is the hit rate if it is (1) direct-mapped, (2) fully associative, or (3) 2-way set-associative?

Direct mapped:	MMMHMHHMMM	30%
Fully associative:	MMMHMHHHHH	60%
2-way associative:	MMMHMHHHMH	50%

Q8. How do superblock and trace scheduling differ?

A trace is a sequence of basic blocks that represents a common path in the program. Traces may have multiple side entries and exits. Superblocks eliminate side entries into a trace by using a process called tail duplication. Tail duplication creates new copies of basic blocks and redirects side entries to these copies rather than into the superblock. The removal of side entries simplifies the process of optimising the superblock since only code motion across an exit must be considered.

Q9. A large last level cache (LLC) is necessary to achieve good performance in many applications. Recent server class processors have included LLCs with capacities of 40 MB or more. Large caches such as this are constructed from numerous smaller SRAM banks.

a) **Describe an appropriate on-chip network to interconnect 32 SRAM banks to create a large LLC. The delay to access a bank should increase as we move further away from the cache controller and bus interface. The SRAM banks are square and the time taken for a signal to travel along the edge of a SRAM bank is much less than a clock cycle.**

   A mesh like network can be used. Since the signal latency for one bank is much smaller than a clock cycle, a concentrated mesh can be used. 4 or more banks are connected to a single router.

b) **To implement a set-associative LLC we may spread each set across multiple banks, i.e. each "way" of the set will be in a different bank. The different associative ways will have different access latencies depending on their distance from the cache controller. How might we optimise the placement of lines in particular banks (or ways) to minimise the cache's average access latency? Remember to consider the cost of moving lines.**

   The idea is to put the most frequently accessed lines to the banks near the controller. To avoid long delay or various delay caused by moving/swapping lines, one method is to swap a line one way near the controller when it is accessed. Gradually the lines far away from the controller are the least accessed.

c) How might the SRAM banks be efficiently interconnected so that the cache's access time is constant regardless of which bank is accessed?

Use a topology that can enforce uniform delay, such as H-net, tree, fat tree, switching network (including crossbar). Note the delay (access time) here is in the unit of cycles rather than the actual time.

d) Why might it be advantageous to be able to manage the amount of LLC used by each co-scheduled thread in a chip multiprocessor?

Improve cache fairness so one thread cannot trash the data of another thread. Also this can be used to enforce priority, quality of service, etc.