CS622A

Advanced Computer Architecture

Assignment III

Multi-core Cache Simulator

GROUP 24

Aman Aryan
20111009

Mani Kant Kumar
20111030

Instructor:
Dr. Mainak Chaudhuri

1 Problem

In this assignment, we will develop a simulator to model a multi-core cache hierarchy supporting directory-based cache coherence with the following specifications:

- Number of cores: 8
- Cache coherence protocol: directory-based MESI
- L1 cache: 32 KB, 8-way, 64-byte block size, LRU

  S state blocks can be replaced silently from L1 caches

  8 in number
- L2 cache: 4 MB, 16-way, 64-byte block size, LRU, 8 banks (each bank is 512 KB), inclusive

2 Instructions to run the code

2.1 To generate machine access traces:

- gcc -O3 -static -pthread prog1.c -o prog1
- ./prog1 8
- make obj-intel64/addrtrace.so
- ../../../pin -t obj-intel64/addrtrace.so – ./prog1 8

2.2 To run the simulator:

- g++ start.cpp -o start
- ./start addrtrace.out

3 Simulation results

Program	No. of simulated cycles	L1 accesses	L1 misses	L2 misses
prog1.c	151165686	140526222	6884861	6510410
prog2.c	2926850	2532135	246654	79627
prog3.c	10864143	9623814	656336	107325
prog4.c	1214763	1065497	131963	70584

Table 3.1

Messages Received by L1	prog1.c	prog2.c	prog3.c	prog4.c
GET	21977	5206	3475	93
GETX	82	28	22	37
PUT	6163350	180758	590444	66051
PUTX	741294	65947	65982	65962
INV	16824	47	192	38
ACK	11599	42	87	38

Table 3.2

Messages Received by L2	prog1.c	prog2.c	prog3.c	prog4.c
GET	6163350	180758	590444	66051
GETX	741294	65947	65982	65962
SWB	21977	5206	3475	93
ACK	82	28	22	37
EWB	6866040	242513	652138	127827

Table 3.3

3.1 List of messages received by the L1 caches:

- GET : When there is read request and dirty bit is 1, the GET message will be forwarded to owner from L2 home bank.
- GETX : When there is write request and dirty bit is 1, the GETX message will be forwarded to owner from L2 home bank.
- PUT : L1 cache gets this PUT message as reply of GET request.
- PUTX : L1 cache gets this PUTX message as reply of GETX request.
- INV (Invalidation) : When there are more than one sharers of block and write permission is requested, then L2 home will send invalidation to all sharers. Moreover, when a block from L2 is evicted, to maintain inclusion these block must be invalidated in L1 cache if present.
- ACK (Acknowledgement) : In case when there are more than one sharers and GETX is requested these sharers will be invalidated and will send acknowledgement to requester L1 cache.

3.2 List of messages received by the L2 cache banks:

- GET : When there is read request in input trace and block is not available in L1 cache, it requester send GET request to L2 home bank.
- GETX : When there is write request in input trace and block is not available in L1 cache, the requester will send GETX request to L2 home bank.
- SWB (Sharing Write Back) : In case of read request and dirty bit =1 the owner will send this SWB message to L2 home bank.
- ACK : In case of write request and dirty bit =1 the owner will send this ACK message to L2 home bank.
- EWB (Evict Write Back) : When a dirty block from L1 is evicted, this message must be passed to L2 home bank to update the state of block in order to maintain inclusion and cache coherence.

3.3 Observations and Explanations :

- The count of PUT messages received by L1 caches is equal to the count of GET messages received by L2 caches : Whenever there is read request in trace file and the block is not available in L1 cache, then L1 will forward this request to L2 home bank and in return the requester must receive PUT message from home bank or owner. So the count is same.

- The count of PUTX messages received by L1 caches is equal to the count of GETX messages received by L2 caches : Whenever there is write request in trace file and the block is not available in L1 cache, then L1 will forward GETX to L2 home bank and L1 will finally receive PUTX from owner or L2 home bank depending upon the dirty bit.

- The count of GET messages received by L1 caches is equal to the count of SWB messages received by L2 caches : Whenever there is read request in trace file and dirty bit is 1, L2 home bank will forward the GET message from requester to the owner which in turn must receive the Sharing Write Back messages since the owner is in M state.

- The count of GETX messages received by L1 caches is equal to the count of ACK messages received by L2 caches : Whenever there is write request in trace file and dirty bit is 1, then L2 home bank will forward the GETX message from requester to the owner which in turn receives Acknowledgement messages.