Multi-core and Multi-processor Architecture (CS423A), Spring 2023
Indian Institute of Technology Kanpur
Assignment 3

REPORT

3

Members: Manish (190477), Sarthak Rout (190772)
Group Number: 10
Date: April 22, 2023

Contents

1 Program 1

- Number of simulated cycles: 67921032
- Number of L1 cache accesses: 141980189
- Number of L1 cache misses: 8567836
- Number of UPGRADE misses: 1732
- Number of L2 misses: 6738623

1.1 L1 Message Counts

L1 Message	Count
GET	**25421**
GET_X	**58**
INVAL_ACK	**20197**
NACK	**1453530**
INVAL	**20197**
PUT	**46182**
PUT_E	**6093929**
PUT_X	**690281**
WB_ACK	**5662447**
INVAL_L2	**1125947**

1.2 L2 Message Counts

L2 Message	Count
GET	**6141135**
GET_X	**2122683**
UPGR	**20104**
SWB	**25418**
SIMPL_WB	**5662447**
OT	**58**
INVAL_L2_REPLY	**1094411**

1.3 Behaviour of the Program

# Threads Shared	1	2	3	4	5	6	7	8
1	0.05%	0.01%	0.26%	4.50%	19.86%	33.96%	24.10%	17.26%

- Almost all the blocks are shared by 4-8 threads
- In the program, we can see that we are initialising the data by the main process and then breaking it into 8 parts. Every thread access data_size/2 access(read then write) starting from tid*(data_part) and accessing in an Arithmetic progression with difference = (tid+1) and wrapping around the data if necessary.

1.4 Observations

- There are a lot INVAL_L2 which means that there is a lot of eviction happening in L2 means the working set doesn't fit into the L2 evenly.
- GETX msgs forwarded to L1 are very less means all GETX are resolved at L2. If it is resolved at L2 means that it will not be present at L2 or It will be in the SHARED state(invalidations will be sent for this case). If it is in the SHARED state means that a lot of INVAL msgs will be sent to L1 which is not the case as there are a not much of INVAL msgs at L1.
- OT will be very less as a location is only written by a single thread only.
- There are lot of SIMPL_WB means that a lot of blocks which are modified are evicted from L1.

2 Program 2

- Number of simulated cycles: 1243625
- Number of L1 cache accesses: 2515385
- Number of L1 cache misses: 10558
- Number of UPGRADE misses: 1477
- Number of L2 misses: 1392

2.1 L1 Message Counts

L1 Message	Count
GET	**1509**
GET_X	**19**
INVAL_ACK	**490**
NACK	**580**
INVAL	**490**
PUT	**2338**
PUT_E	**875**
PUT_X	**1681**
WB_ACK	**886**
INVAL_L2	**0**

2.2 L2 Message Counts

L2 Message	Count
GET	**3214**
GET_X	**1714**
UPGR	**546**
SWB	**1509**
SIMPL_WB	**886**
OT	**19**
INVAL_L2_REPLY	**0**

2.3 Behaviour of the Program

# Threads Shared	1	2	3	4	5	6	7	8
2	0.53%	12.52%	24.84%	62.09%	0.01%	0	0	0.01%

- Almost all blocks are shared by more than 1 thread.
- What the program code is doing is that it divides the data into 8(num_of_threads) parts. For every thread, it read and writes the data for its data blocks and reads the 1 left neighbour and 2 right neighbours' data.

2.4 Observations

- There are zero INVAL_L2 which means that there is no eviction happening in L2 means the working set fits into the L2 evenly.
- SWB is sent when we have written our data and someone else is reading from it. This is quite happening in this program.
- OT will be very less as a location is only written by a single thread only.

3 Program 3

- Number of simulated cycles: 4146299
- Number of L1 cache accesses: 9509426
- Number of L1 cache misses: 28396
- Number of UPGRADE misses: 18140
- Number of L2 misses: 1393

3.1 L1 Message Counts

L1 Message	Count
GET	**8594**
GET_X	**17**
INVAL_ACK	**7764**
NACK	**5530**
INVAL	**7764**
PUT	**8706**
PUT_E	**1009**
PUT_X	**8898**
WB_ACK	**1099**
INVAL_L2	**0**

3.2 L2 Message Counts

L2 Message	Count
GET	**9725**
GET_X	**5241**
UPGR	**9177**
SWB	**8594**
SIMPL_WB	**1099**
OT	**17**
INVAL_L2_REPLY	**0**

3.3 Behaviour of the Program

# Threads Shared	1	2	3	4	5	6	7	8
3	0.54%	0.08%	0	0	0	0	0	99.38%

- All the blocks are shared by all the threads
- In the program data is divided into num_of_threads and every thread picks a unique data part and reads and writes to it.

3.4 Observations

- There are zero INVAL_L2 which means that there is no eviction happening in L2 means the working set fits into the L2 evenly.

- SWB is quite high due to the write migratory nature of cache blocks.
- GETX msgs forwarded to L1 are very less means all GETX are resolved at L2. If it is resolved at L2 means that it will not be present at L2 or It will be in the SHARED state. If it is in the SHARED state means that a lot of INVAL msgs will be sent to L1 which is quite the case as there are a lot of INVAL msgs at L1.
- Mostly GET msg is forwarded to L1(91%) means reads to a block are coming after a write to a block has already happened at L1.

4 Program 4

- Number of simulated cycles: 729239
- Number of L1 cache accesses: 1066002
- Number of L1 cache misses: 6160
- Number of UPGRADE misses: 54
- Number of L2 misses: 1379

4.1 L1 Message Counts

L1 Message	Count
GET	**434**
GET_X	**35**
INVAL_ACK	**74**
NACK	**56**
INVAL	**74**
PUT	**479**
PUT_E	**922**
PUT_X	**1295**
WB_ACK	**959**
INVAL_L2	**0**

4.2 L2 Message Counts

L2 Message	Count
GET	**1404**
GET_X	**1285**
UPGR	**66**
SWB	**433**
SIMPL_WB	**959**
OT	**35**
INVAL_L2_REPLY	**0**

4.3 Behaviour of the Program

# Threads Shared	1	2	3	4	5	6	7	8
4	12.95%	87.03%	0.01%	0	0	0	0	0.01%

- Mostly all the blocks are shared by 2 threads
- In the program what we are doing is thread zero which is the main function is initializing the array and then the data is divided into 8 parts and all 8 threads reads their corresponding part.

4.4 Observations

- There are zero INVAL_L2 which means that there is no eviction happening in L2 means the working set fits into the L2 evenly.
- GETX msgs forwarded to L1 are very less means all GETX are resolved at L2. If it is resolved at L2 means that it will not be present at L2 or It will be in the SHARED state. If it is in the SHARED state means that a lot of INVAL msgs will be sent to L1 which is **not** the case as there are a very of INVAL msgs at L1. This also justifies our program logic.
- OT will be very less as a location is only written by a single thread only.
- There are very less UPGR msg means that no thread is writing the data which is sharing it with someone.

5 Appendix A: Message Glossary

- GET: Sent for a read miss from L1 to L2. Read Intervention with the same name will be forwarded by the L2 to L1 if needed.
- GET_X: Sent for a write miss from L1 to L2. Write Intervention with the same name will be forwarded by the L2 to L1 if needed.
- UPGR: Sent when L1 have a block in shared and wants to write.
- INVAL_ACK: Sent after invalidating the block by L1 to another L1 which is waiting for the Acks.
- NACK: Will be sent when our request cannot be served by the L2(directory state in pending).
- INVAL: Will be sent by L2 to all the sharers of a block to L1 when someone wants to write.
- SWB: Sent when L1 receive a GET request forwarded by L2 and L1 has block in the E/M state.
- SIMPL_WB: Sent by L1 when a block is evicted which is in M state.
- PUT: Sent by L1 or L2 for the read request coming to it
- PUT_E: Sent by L2 to L1 when there is no sharer of the block in the system
- PUT_X: Sent by L1 or L2 for the write request coming to it.
- WB_ACK: Sent by L2 after receiving a SIMPL_WB from L1.
- OT: Sent by L1 after it receive a GETX and L1 has block in E/M state
- INVAL_L2: Sent by L2 to L1 when an eviction happens(to maintain inclusivity)
- INVAL_L2_REPLY: Sent by L1 to L2 after receiving INVAL_L2 and L1 has block in E/M state.