CS622A

Advanced Computer Architecture

Assignment 3

Multi Level Cache With Directory-Based Coherence Protocol Simulator

---

GROUP 9

Mohit Kumar
20111034

Rohit Katariya
20111050

Instructor:
Dr. Mainak Chaudhuri

December 02, 2020

1 Implementation Details

1.1 Messages Used in the Input Queue of L1 cache

1. **PUTE**
   - This message is generated when Core sends an GET request to directory .
   - The sender is the first sharer to the requested block.
   - Then, PUTE message reply is generated.
2. **PUTX**
   - This message is generated when Core sends an GETX request to directory .
   - Then ,PUTX message reply is generated .
3. **PUT**
   - This message is generated when Core sends an GET request to directory .
   - The sender is not the first sharer to the requested block.
   - Then ,PUT message reply is generated .
4. **GET**
   - This message is generated when Core sends an GET request to directory.
   - The directory finds out that some other Core has kept this requested block in M mode .
   - Then,directory will forward the GET request to current owner of this block and also send the PUT message reply to the requested core.
   - The current owner will change its mode from M to S and writes back to memory.
5. **GETX**
   - This message is generated when Core sends an GETX request to directory.

- The directory finds out that some other Core has kept this requested block in M mode .
- Then,directory will forward the GETX request to current owner of this block and also send the PUTX message reply to the requested core.
- The current owner will change its mode from M to I.

6. **UPGR-ACK**

   - This message is generated when Core has block in S mode and it wants the block in M mode .
   - Then,core will send the Upgrade Message to the directory.
   - Then,directory will reply with UPGR-ACK message to the requested core.

7. **INV** & **INV-ACK**

   - This message is generated in two case:
   - (a) When some core request the block in GETX mode .
     (b) Then, directory find outs the mode of the requested block . If the mode is S . It will send the INV to all the sharer of the requested block.
     (c) All the sharer will reply the INV-ACK to the requested core.
     (d) Directory will also send PUTX to the requested core .
   - (a) When some core has block in S mode and it send the UPGRADE request to the directory .
     (b) Then,directory will find out the all the sharer of the requested block and sends the INV to them .
     (c) All the sharer will reply with the INV-ACK to the requested core.
     (d) Directory will also send UPGR-ACK to the requested core.

1.2 Messages Used in the Input Queue of L1 cache

1. **GET**

   - When some core want to access the block in S mode ,it will send the GET request to the directory .

- Depending upon whether it is first sharer or not , it responds with PUTE or PUT message respectively .

2. **GETX**

   - When some core want to access the block in M mode , it will send the GETX request to the directory .
   - Then ,directory will respond with PUTX message to the requested core .

3. **SWB**

   - When some core want to access the block in S mode ,it will send the GET request to the directory .
   - Then, directory will find out the requested block has current owner or not .
   - If it is so,it will send the GET request to the current owner of the requested block.
   - The current owner will forward the block to requester and changes its mode from M to S and writes back to memory .
   - This write back has been symbolised as SWB.

4. **EvictWB**

   - All the write back to the memory except SWB has been symbolized as EvictWB.
   - It is done so that memory has latest copy whenever there is eviction of cache block from M state .

5. **UPGRADE**

   - When some core has block in S mode and wants it in M mode .
   - It sends UPGRADE message to directory .

6. **ACK**

   - When some core sends GETX request to directory .
   - The directory finds out that there is some owner of the requested block.

- The directory sends GETX to current owner of the requested block.
- The current owner forward the block to requested core and send only ACK to the directory .

2 Simulation Results

2.1 Stimulated Cycle

Program	**Stimulated Cycles**
Prog. 1	130436745
Prog. 2	2515810
Prog. 3	9442493
Prog. 4	1062912

2.2 L1 Cache Statistics

2.2.1 Program 1

Core	**L1 Accesses**	**L1 Hits**	**L1 Miss**	**L1 Upgrade Misses**
Core 0	33568897	32420975	1147912	10
Core 1	13631729	12910373	721351	5
Core 2	13631701	12877580	754118	3
Core 3	13631704	12713761	917938	5
Core 4	13631643	12812011	819630	2
Core 5	13631673	12779277	852394	2
Core 6	13631661	12746508	885150	3
Core 7	13631794	12844905	786880	9

2.2.2 Program 2

Core	L1 Accesses	L1 Hits	L1 Miss	L1 Upgrade Misses
Core 0	735269	652777	82490	2
Core 1	263868	239251	24616	1
Core 2	262377	267763	26414	0
Core 3	263811	239197	24614	0
Core 4	262312	237700	24612	0
Core 5	262411	237794	24616	1
Core 6	262365	237753	24612	0
Core 7	196775	180355	16419	1

2.2.3 Program 3

Core	L1 Accesses	L1 Hits	L1 Miss	L1 Upgrade Misses
Core 0	2112176	1914962	197212	2
Core 1	1049361	983774	65586	1
Core 2	1051746	986152	65594	0
Core 3	1049915	984326	65589	0
Core 4	1049076	983486	65590	0
Core 5	1052652	987057	65595	0
Core 6	1048751	983174	65577	0
Core 7	1051270	985676	65594	0

2.2.4 Program 4

Core	L1 Accesses	L1 Hits	L1 Miss	L1 Upgrade Misses
Core 0	604275	529970	74303	2
Core 1	65697	57470	8227	0
Core 2	65697	57470	8227	0
Core 3	65697	57470	8227	0
Core 4	65697	57470	8227	0
Core 5	65697	57470	8227	0
Core 6	65697	57471	8226	0
Core 7	65698	57471	8226	1

2.3 L2 Cache Statistics

Program Number	L2 Accesses	L2 Hits	L2 Miss
Prog. 1	6861557	367983	6493574
Prog. 2	244136	177197	66939
Prog. 3	650157	582870	67287
Prog. 4	131768	65111	66657

2.4 Messages Received By L1 Cache

2.4.1 Program 1

Core	PUTE	PUTX	PUT	GET	GETX	UPGR-ACK	INV	INV-ACK
Core 0	424644	721200	2068	3668	35	1610	1954	3651
Core 1	718753	45	2553	3016	5	2024	2133	1270
Core 2	751170	38	2910	2686	6	2607	2578	1488
Core 3	914386	37	3515	2737	3	2713	1906	2072
Core 4	816658	51	2921	2365	4	2570	751	1906
Core 5	848953	43	3398	2929	6	2713	789	1318
Core 6	881781	48	3321	2539	2	2927	2417	2142
Core 7	782491	59	4330	3807	8	4163	2888	1569

2.4.2 Program 2

Core	PUTE	PUTX	PUT	GET	GETX	UPGR-ACK	INV	INV-ACK
Core 0	16164	65803	523	30	29	15	15	45
Core 1	24588	10	18	520	1	4	16	7
Core 2	24078	9	527	502	0	3	6	10
Core 3	24077	9	528	519	1	4	8	12
Core 4	23564	9	1039	7	1	5	6	16
Core 5	24585	11	20	831	3	4	6	8
Core 6	24076	9	527	6	1	4	28	9
Core 7	16078	331	5	5	1	3	28	6

2.4.3 Program 3

Core	PUTE	PUTX	PUT	GET	GETX	UPGR-ACK	INV	INV-ACK
Core 0	127368	65812	4032	543	29	533	556	1593
Core 1	65040	11	535	1018	0	519	557	1557
Core 2	65549	10	35	1021	1	9	531	21
Core 3	65037	10	542	508	1	519	17	530
Core 4	65037	10	543	1019	0	520	530	1594
Core 5	65548	10	37	1020	1	10	536	20
Core 6	65546	10	21	508	0	10	1567	23
Core 7	65039	10	545	510	1	519	1568	534

2.4.4 Program 4

Core	PUTE	PUTX	PUT	GET	GETX	UPGR-ACK	INV	INV-ACK
Core 0	8459	65814	30	47	28	32	15	89
Core 1	8201	12	14	7	2	3	19	8
Core 2	8201	12	14	7	1	2	8	7
Core 3	8197	12	18	4	2	3	7	6
Core 4	8197	12	18	5	1	3	8	5
Core 5	8197	12	18	5	1	3	8	6
Core 6	8197	11	18	6	0	4	35	7
Core 7	8200	11	15	4	2	3	32	4

2.5 Messages Received By L2 Cache Banks

Program Number	GET	GETX	SWB	EvictWB	UPGRADE	ACK
Prog. 1	6140105	721452	23747	6857425	21327	69
Prog. 2	178304	65832	2420	239216	42	37
Prog. 3	584307	65850	6147	645994	2639	33
Prog. 4	65909	65859	85	127686	53	37

3 Simulation Analysis

3.1 Miss count in each core

1. For every miss in each core, it will either send GET or GETX request to directory.

2. Depending upon state of the requested block,the directory will take further action .

3. If requested block is marked as ’I’ in directory, it will first get the block from memory or L2 cache and responds with PUTE(in case of GET request)or PUTX(in case of GETX) message .

4. If the requested block is marked as ’S’ in directory :

    - If the request is ’W’ or ’GETX’ type, then first directory will invalidate all the sharer of the block . The L2 cache will provide the data and responds with ’PUTX’ message .
    - If the request is ’R’ or ’GET’ type , then directory will ask the L2 cache to provide the data along with ’PUT’ message .

5. If the requested block is marked as M in directory :

    - If the request is ’W’ or ’GETX’ type , then directory will forward the request to the current owner of requested block. The directory also responds with ’PUTX’ message to the requester core .
    - If the request is ’R’ or ’GET’ type ,then directory will forward the request to the current owner of the requested block . The directory also responds with ’PUT’ message to the requester core .

6. Therefore, for each core : Number of Misses = (Number of PUT + Number of PUTE + Number of PUTX) message it received.

3.2 Account of Cold Misses

Program	Number of Misses in L2	Cold Misses
Prog. 1	6493574	721373
Prog. 2	66939	66008
Prog. 3	67287	66005
Prog. 4	66657	66008

- From the above table ,we can conclude :
- In Prog. 1, most of the misses are not cold misses. Due to this, the latency will increase.
- In Prog. 2 , Prog. 3 and Prog. 4 ,most of the misses are cold misses. Because of which the latency will be less as other kind of misses has not occurred too much .

3.3 Number of Sharing Write Back(SWB)

- When some core request a block by sending the GET message.
- If the directory finds out that the 'requested block has been owned by some other core (i.e in M mode) '.
- The directory will forward the 'GET' request to the current owner.
- The current owner will forward the block to requester core and also writes back in the memory. The owner core will change the mode of block from 'M' to 'S'.
- Whenever there is write back due to transition from 'M' to 'S' ,it is termed as SWB.
- Therefore, Number of SWB = (Sum of GET Request forwarded to each core).
- The SWB has occur in large amount in Prog. 1 ,2, and 3 as compared to Prog. 4 .

3.4 Number of ACK Message Received by L2 Cache Bank

- When some core request a block by sending the 'GETX' message.
- If the directory finds out that the 'requested block has been owned by some other core (i.e in M mode) '.
- The directory will forward the 'GETX' request to the current owner.
- The current owner will forward the block to requester core ,invalidate it from its private cache and send ACK to directory.
- Therfore,Number of ACK Message Received by L2 Cache Bank = (Sum of GETX Request forwarded to each core).

3.5 L1 Access by each Core

- Each program follows same trend:
- Apart from Core 0 ,rest of all core in each program has nearly same number of L1 access . This shows the even distribution of work among the rest 7 threads .
- It may be possible that Core 0 is doing the main program work also because of which L1 access of Core 0 is higher as compare to the L1 access of rest 7 core in each program .

3.6 Number of Upgrade Requests Received By L2 Cache Banks

- Number of Upgrade Request received by L2 Cache Banks is higher in Prog.1 and Prog. 3 & lower in Prog. 2 and Prog. 4.
- High number of Upgrade Request shows that initially some blocks has more than 1 sharer in S mode,then one core decided to modify the block.

3.7 Amount of PUT Reply Message Received By L1 Cache

- In Prog.1 ,2,and 3 have high number of PUT Reply message as compare to Prog. 4 .
- Possible Reason : In Prog.4 there is less amount of data read(shared) among cores at same time.