Observations and results:

Results for Prog 1:

Number of simulated cycles : 69873350

Total machine accesses : 140526689

L1 cache accesses, hits and misses :

Core	L1 cache accesses	L1 cache hits	L1 cache misses	L1 cache upgrade misses
0	37765122	34673793	1147755	24926
1	14680226	11466780	728217	144105
2	14680238	11615214	728217	168201
3	14680244	10739714	932119	286509
4	14680218	11548417	823136	215587
5	14680198	11336569	859265	239014
6	14680208	11485652	888538	263552
7	14680236	11946254	786632	192722

L2 cache misses :

Bank	Misses count
0	813783
1	813666
2	813656
3	813673
4	813597
5	813749
6	813745
7	813687

L1 messages (in order of cores) :

Put	Putx	Inv	Inv_ack	Xfer	Xfer_inv
1203	1171477	3662	3662	1190	23796
1851	870467	3208	3208	1850	142647
1979	923671	3266	3266	1973	166357
2054	1216574	2685	2685	2044	285066
1935	1036786	3153	3153	1925	213826
1931	1096347	2742	2742	1921	237519

2033	1150055	3208	3208	1986	261746
2405	976947	4776	4776	2400	190337

L2 messages (in order of bank) :

Get	Getx	Swb	Upgrade	Repl_s
1913	863957	191919	191680	671622
1921	863360	192030	191788	670994
1958	863496	192065	191814	671074
1897	862914	191912	191660	670667
1930	863452	192051	191808	671079
1912	863439	191964	191723	671043
1935	863430	192115	191862	670973
1925	863673	192530	192281	670794
Total:15391	6907721	1536586	1534616	1538246

Results for Prog 2 :

Number of simulated cycles : 1809667

Total machine accesses : 2513803

L1 cache accesses, hits and misses:

Core	L1 cache accesses	L1 cache hits	L1 cache misses	L1 cache upgrade misses
0	737190	652402	82664	406
1	264501	237973	24743	371
2	262299	235578	24733	387
3	262299	235873	24749	355
4	262299	235600	24736	387
5	265777	239251	24736	358
6	262678	236110	24740	372
7	196761	178399	16536	361

L2 cache misses:

Bank	Misses count
0	8396
1	8322
2	8346
3	8326
4	8362
5	8352
6	8370
7	8431

L1 messages (in order of cores) :

Put	Putx	Inv	Inv_ack	Xfer	Xfer_inv
24	83046	29	29	21	397
15	25099	8	8	8	371
514	24605	9	9	506	287
17	25087	10	10	12	355
19	25104	10	10	9	384
18	25076	11	11	7	359
19	25093	8	8	9	370
8202	8695	8	8	8199	361

L2 messages (in order of banks):

Get	Getx	Swb	Upgrade	Repl_s
1098	29887	1488	392	28079
1101	29822	1457	371	28034
1086	29854	1448	355	28074
1087	29841	1485	391	28028
1113	29855	1493	398	28041
1110	29832	1464	368	28043
1100	29844	1459	365	28058
1133	29874	1461	357	28086
Total:8828	238809	11755	2997	22443

Results for Prog 3 :

Number of simulated cycles : 6379616

Total machine accesses : 9508868

L1 cache accesses, hits and misses :

Core	L1 cache accesses	L1 cache hits	L1 cache misses	L1 cache upgrade misses
0	2113467	1903097	198054	2450
1	1057082	978668	66305	2345
2	1051895	973452	66333	2376
3	1060012	982243	66298	2252
4	1053558	975333	66305	2321
5	1067009	988999	66330	2312
6	1057110	979231	66289	2259
7	1048736	970280	66351	2390

L2 cache misses:

Bank	Misses count
0	8537
1	8374
2	8427
3	8404
4	8458
5	8507
6	8509
7	8431

L1 messages (in order of cores) :

Put	Putx	Inv	Inv_ack	Nack	Xfer	Xfer_inv
3379	197125	35	35	0	3379	2432
17	68632	23	23	0	12	2342
17	68692	16	16	0	14	2370
23	68527	18	18	0	18	2247
20	68606	21	21	0	8	2317
22	68620	12	12	0	12	2307
19	68529	14	14	0	10	2256
19	68722	9	9	0	10	2386

L2 messages (in order of banks):

Get	Getx	Swb	Upgrade	Repl_s
433	82336	2824	2390	79449
433	82317	2801	2366	79454
433	82365	2833	2393	79461
436	82285	2634	2191	79582
447	82373	2831	2403	79478
467	82346	2660	2256	79615
412	82392	2816	2415	79505
445	82335	2721	2291	79537
Total: 3506	658749	22120	18705	636081

Results for Prog4 :

Number of simulated cycles : 869164

Total machine accesses : 1065970

L1 cache accesses, hits and misses:

Core	L1 cache accesses	L1 cache hits	L1 cache misses	L1 cache upgrade misses
0	606131	531606	74357	31
1	65690	57452	8230	6
2	65690	57447	8230	6
3	65700	57458	8231	5
4	65690	57453	8233	4
5	65690	57451	8231	4
6	65690	57444	8230	6
7	65690	57453	8230	3

L2 cache misses:

Bank	Misses count
0	8375
1	8314
2	8327
3	8314
4	8324
5	8323
6	8349
7	8392

L1 messages (in order of cores):

Put	Putx	Inv	Inv_ack	Xfer	Xfer_inv
27	74361	31	31	23	23
14	8221	9	9	7	6
19	8216	10	10	11	5
18	8218	10	10	14	4
10	8227	8	8	7	5
17	8218	9	9	6	6
18	8218	8	8	7	7
17	8216	8	8	6	4

L2 messages (in order of banks):

Get	Getx	Swb	Upgrade	Repl_s
10	16498	25	18	15987
14	16465	10	11	15952
0	16473	9	2	15953
0	16469	10	3	15950
30	16466	26	18	15956
19	16470	15	0	15960
31	16483	20	4	15984
36	16508	26	9	15995
Total: 140	131832	141	65	127737

Get: Whenever a processor requests to read a block from home, ‘get’ is sent .

Getx: Whenever a processor wants to write/modify, ‘getx’ is sent to home.

Put: ‘Put’ message is sent from home/owner of block as a reply to ‘get’ request and then both owner and home goes into shared state.

Putx: ‘Putx’ message is sent from home/owner of block as a reply to ‘getx’ request , requester becomes the new owner and goes in modified state and other sharers are invalidated.

Swb: This message is sent from owner to home in order to update the block in the home whenever there is a core requesting that same block.

Upgrade: This meassage is sent whenever a processor has replaced a particular block and then requested that block again. Here the requester changes state from invalid to modified state.

Inv: ‘Inv’ message is sent from home to the sharers of block whenever there is request to write on that block by some other processor. All the sharers change the state from shared to invalid.

Inv_ack: This message is sent from sharers to the requesting processor whenever requester wants that particular block to write/modify.

Nack: Whenever home finds that a block is in shared state and that same processor or some other processor is requesting for that block, ‘nack’ is sent from home to that processor. This may lead to livelock.

Xfer: Whenever the block is requested to perform read and home does not have it, ‘xfer_inv’ is sent from home to owner so that owner send the block to requester directly(cache to cache transfer).

Xfer_inv: Whenever the block is requested to perform write and home does not have it, ‘xfer_inv’ is sent from home to owner so that owner send the block to requester directly(cache to cache transfer).

Repl_s: This message is sent whenever a processor wants to replace a block in non-silent manner. Here the requester will go to invalid state.

L1 cache upgrade misses : If a core contains a block in shared state and it wants to perform write on it, even though L1 core contains that block as it does not have the write permissions, it is considered as a miss called as cache upgrade miss.

A buffer is implemented for ‘Miss handling table’ which helps to handle the blocks which are already being fetched, that is whenever a block is requested and it is a miss, it is inserted into Miss handling table and if similar block request is generated and the earlier process is being processed, this current request is not considered as a miss and obtained results are sent to all the requesters after the block arrives.

Observations:

1. Number of upgrade misses are lesser than the L1 cache misses which implies that if a block is in shared block then the request to write on same block is rare.

2. Number of getx in L2 cache is quite high as compared to xfer_inv implies that most of the requests are sent from owner itself and very few are cache to cache transfer.

3. As machine accesses are higher than the number of simulated cycles, we can say that there are multiple threads working in parallel.

4. The number of L1 cache accesses in core 0 for all the programs is higher than the accesses for other cores. This might be because core 0 is handling not only the shared part of the program but also other parts of the program that facilitate this shared computation.

We can verify the results from following points:

1. Sum of xfer and xfer_inv is equal to sum of swb(sharing write back) messages from L2 banks.

2. Sum of inv messages and sum of inv_ack meassages for L1 cache is same.

3. Count of xfer for any L1 core is less than sum of all put messages from rest cores which implies that the reply for each xfer is included in count of put.

4. Count of xfer_inv for any L1 core is less than sum of all putx messages from rest cores which implies that the reply for each xfer_inv is included in count of putx.