Machine Problem 2

Cache Coherence Protocols in a SMP system

Kashyap Ravichandran, (kravich2), ECE 506

Introduction:

Three coherence protocols, namely MSI, MESI, and Dragon, was implemented for a single level cache. Metrics like number of reads and writes, number of read misses and write misses, number of writes backs, number of cache to cache transfers, number of interventions and invalidations, number of BusRdx issued and number of flushes were observed. Three experiments were performed by varying either the cache size or the associativity or the block size and the other two parameters were kept a constant. The results were tabulated, and few key trends were displayed graphically for better understanding. The trends were also briefly discussed, and the cause was identified.

Experiments:

Varying the size of the cache and keeping the associativity and block size a constant.

The following configuration were followed throughout for this experiment:
Cache Size: 1MB
Block Size: 64B

a) MSI Protocol

An important trend that we could see as we increase the cache size is that the miss rate reduces initially but then saturates when the size is too big. Another important trend is that BusRdx doesn't change with increasing cache size. This is obvious as a BusRdx doesn't depend on the cache configuration but on the protocol employed. Another important trend that can possibly be affected by the size of the cache is the fact that interventions, invalidations and flushes doesn't vary with size. This can be mainly attributed to the fact that there are more memory location right now and as a result less contention which actually saturates interventions, invalidations and the flushes metrics.

Cache Size: 256K

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5775	5805	5771	5813
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.67%	4.77%	4.57%	4.66%
6	Writebacks toMemory	254	235	278	234
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	6745	6740	6774	6761
9	Interventions	68	47	81	63
10	Invalidations	2014	2034	2008	2020
11	Flushes	113	92	120	88

12	BusRdX issued	716	700	725	714

Cache Size: 512K

Sr. No.	**Statistics**	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5757	5792	5756	5796
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.65%	4.76%	4.55%	4.65%
6	Writebacks toMemory	190	170	205	171
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	6663	6662	6686	6681
9	Interventions	71	47	82	63
10	Invalidations	2014	2034	2008	2020
11	Flushes	116	92	121	88
12	BusRdX issued	716	700	725	714

Cache Size: 1M

Sr. No.	**Statistics**	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5752	5781	5752	5796
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.65%	4.75%	4.55%	4.65%
6	Writebacks toMemory	170	155	186	171
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	6638	6636	6663	6661
9	Interventions	71	48	82	63
10	Invalidations	2014	2034	2008	2020
11	Flushes	116	93	121	88
12	BusRdX issued	716	700	725	714

Cache Size: 2M

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5750	5779	5751	5789
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.65%	4.75%	4.55%	4.64%
6	Writebacks toMemory	166	143	176	152
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	6632	6622	6652	6655
9	Interventions	71	48	82	64
10	Invalidations	2014	2034	2008	2020
11	Flushes	116	93	121	89
12	BusRdX issued	716	700	725	714

b) MESI Protocol

All trends explained earlier apply to this too. We can see that between MSI and MESI there is a significant reduction in BusRdx that is issued by a processor. And memory transaction has reduced drastically because of cache to cache transfer. The read and the write misses are expected to be constant when we compare configurations between MSI and MESI protocols. There is an increase in the number of interventions as we have included a new non-sharing state. The invalidation is a constant across both MSI and MESI which was expected.

Cache Size: 256K

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5775	5805	5771	5813
3	Number of Writes	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.67%	4.77%	4.57%	4.66%
6	Writebacks toMemory	254	235	278	234
7	Cache-to-Cache Transfers	4405	4441	4406	4411
8	Memory Transaction	1663	1640	1685	1675
9	Interventions	1468	1432	1469	1479
10	Invalidations	2014	2034	2008	2020
11	Flushes	113	92	120	88
12	BusRdX issued	39	41	42	39

Cache Size: 512K

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5757	5792	5756	5796
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.65%	4.76%	4.55%	4.65%
6	Writebacks toMemory	190	170	205	171
7	Cache-to-Cache Transfers	4392	4431	4396	4400
8	Memory Transaction	1594	1572	1607	1606
9	Interventions	1466	1429	1465	1474
10	Invalidations	2014	2034	2008	2020
11	Flushes	116	92	121	88
12	BusRdX issued	39	41	42	39

Cache Size: 1M

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5752	5781	5752	5790
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.65%	4.75%	4.55%	4.64%
6	Writebacks toMemory	170	155	186	157
7	Cache-to-Cache Transfers	4389	4422	4393	4395
8	Memory Transaction	1572	1555	1587	1591
9	Interventions	1464	1428	1464	1474
10	Invalidations	2014	2034	2008	2020
11	Flushes	116	93	121	89
12	BusRdX issued	39	41	42	39

Cache Size: 2M

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5750	5779	5751	5789
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39

5	Total Miss Rate	4.65%	4.75%	4.55%	4.64%
6	Writebacks toMemory	166	143	176	152
7	Cache-to-Cache Transfers	4387	4420	4392	4394
8	Memory Transaction	1586	1543	1577	1586
9	Interventions	1464	1428	1464	1474
10	Invalidations	2014	2034	2008	2020
11	Flushes	116	93	121	89
12	BusRdX issued	39	41	42	39

c) Dragon

Since the protocol does not have invalid state, we don't have invalidation in the dragon protocol. Compared to the protocols observed earlier we could see that the number of memory transaction is too high. This is mainly attributed to the fact that we don't have any cache to cache transfers. There are lesser number of intervention as we increase the size of the cache, this can be attributed to the fact that increase in cache size would reduce the contention for a particular block.

Cache Size: 256K

Sr. No.	**Statistics**	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5635	5646	5644	5652
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	3	2	2	0
5	Total Miss Rate	4.52%	4.61%	4.43%	4.50%
6	Writebacks toMemory	226	243	232	234
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	5864	5891	5878	5886
9	Interventions	1405	1396	1398	1430
10	Invalidations	0	0	0	0
11	Flushes	3	9	6	9
12	BusRdX issued	0	0	0	0

Cache Size: 512K

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5601	5610	5610	5617
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	3	2	2	0
5	Total Miss Rate	4.50%	4.58%	4.41%	4.47%
6	Writebacks toMemory	128	127	135	141
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	5732	5739	5747	5758
9	Interventions	1400	1388	1389	1418
10	Invalidations	0	0	0	0
11	Flushes	3	9	6	6
12	BusRdX issued	0	0	0	0

Cache Size: 1 M

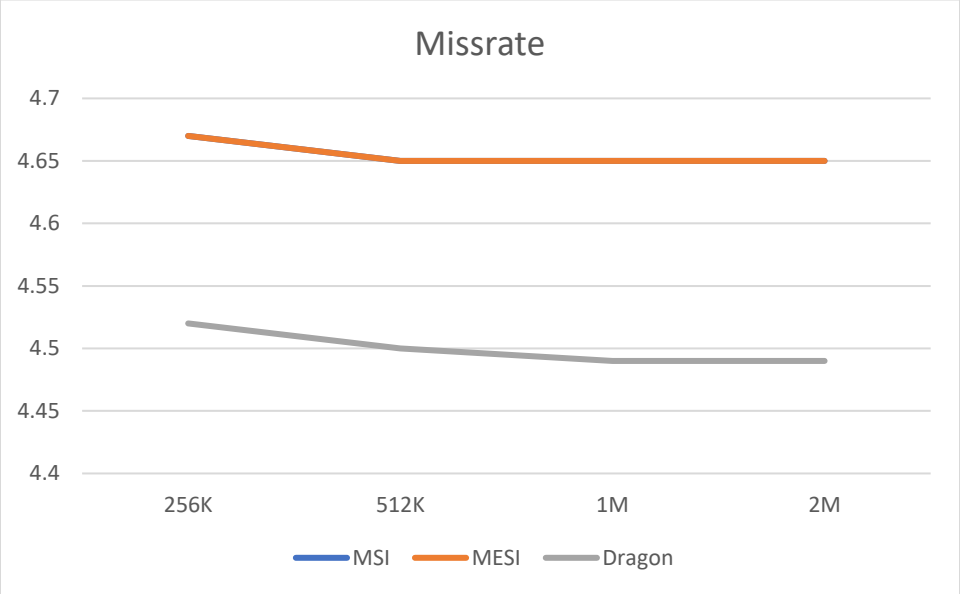
Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5595	5604	5604	5611
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	3	2	2	0
5	Total Miss Rate	4.49%	4.57%	4.40%	4.47%
6	Writebacks toMemory	107	110	105	123
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	5705	5716	5711	5734
9	Interventions	1398	1387	1387	1417
10	Invalidations	0	0	0	0
11	Flushes	3	9	6	6
12	BusRdX issued	0	0	0	0

Cache Size: 2 M

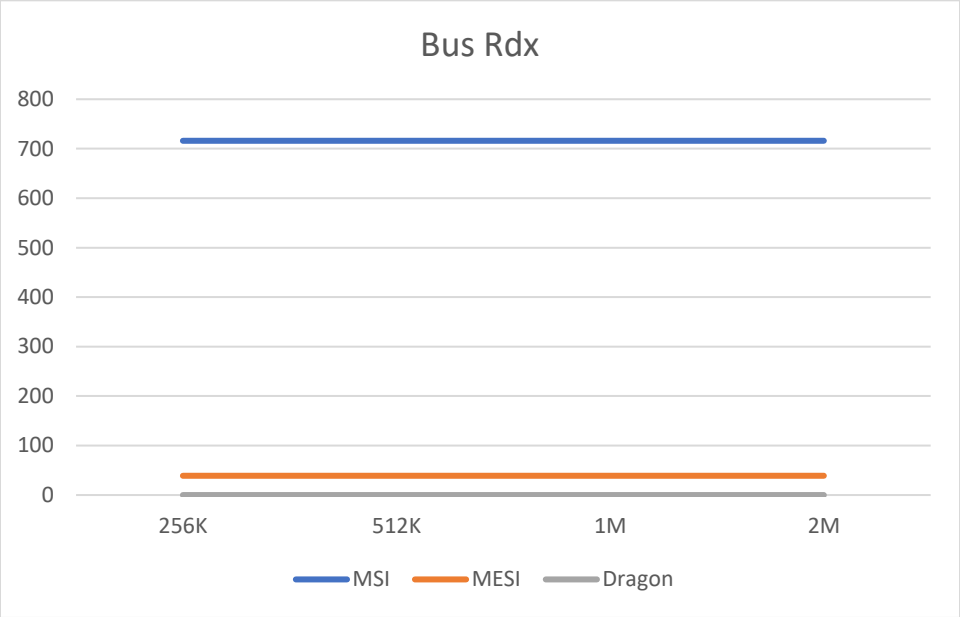
Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5591	5603	5601	5608
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	3	2	2	0
5	Total Miss Rate	4.49%	4.57%	4.40%	4.47%
6	Writebacks toMemory	102	105	98	121
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	5696	5710	5701	5729
9	Interventions	1396	1387	1387	1416
10	Invalidations	0	0	0	0
11	Flushes	3	9	6	6
12	BusRdX issued	0	0	0	0

Miss Rate:

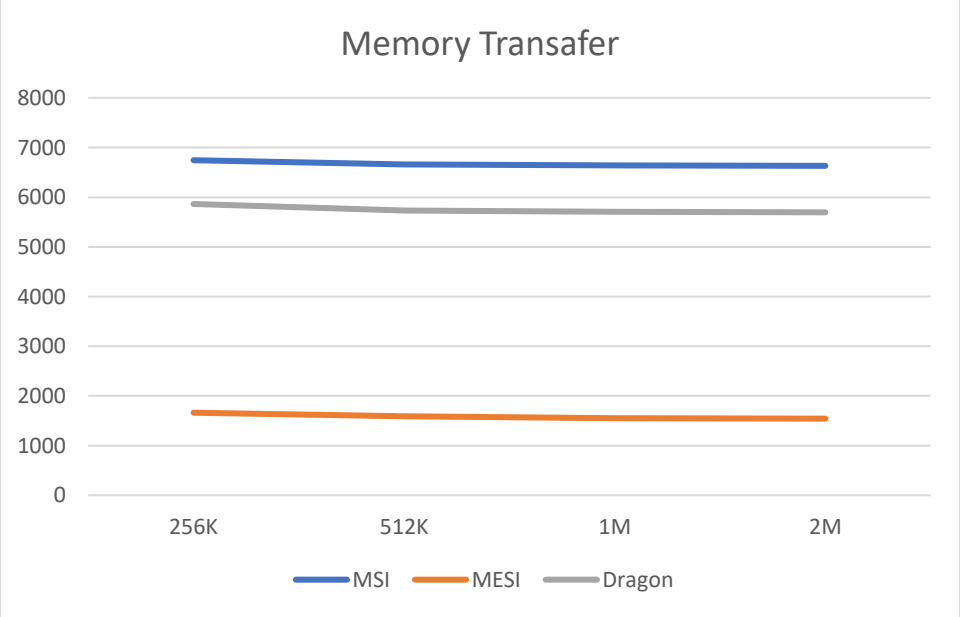
To make the graphs easily comprehensible, we are going to see the effect of the cache size, associativity and block size on one single processor.



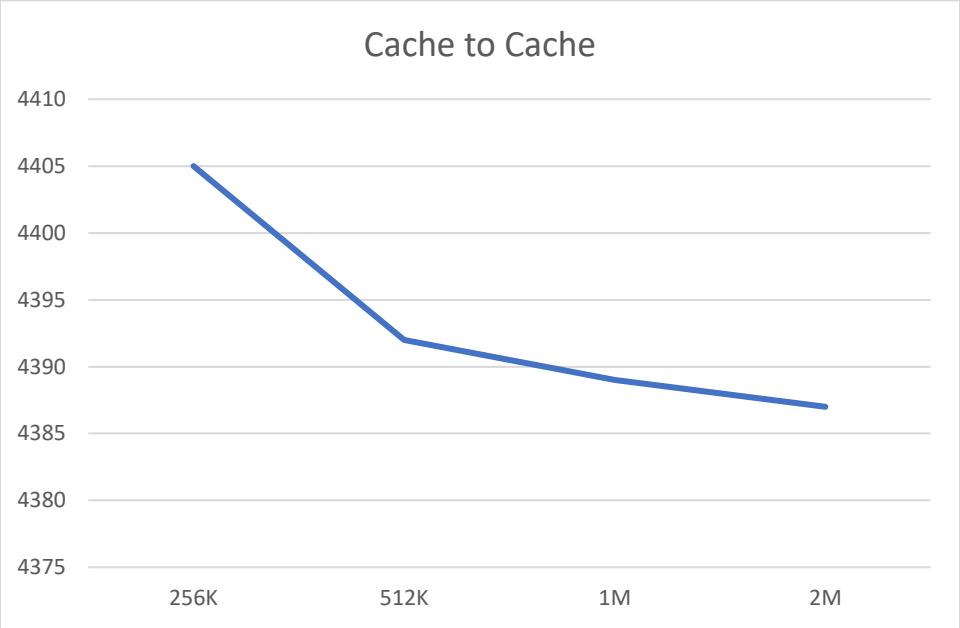
Bus Rdx:



Memory Transfer



Cache to Cache Transfer



Another important trend that we could notice here is that the number of write backs have reduced as we increase the size of the cache. This is because as we increase the size of the cache, there is less contention between for a cache space and has resulted in the reduction of write backs as the number of evictions has come down. However, the write backs incurred during flushing is still the same as the cache block still goes through all the same states.

Varying the associativity and keeping the cache size and block size a constant.

The following configuration were followed throughout for this experiment:
Cache Size: 1MB
Block Size: 64B

a) MSI

By increasing the associativity, we reduce the number of conflicting miss in the trace file. This is visible across all three protocols. However even after increasing the associativity of the cache, we are not able to service all the write misses again across all three protocols. This is mainly because all those three misses were compulsory misses and a system can't reduce compulsory miss. The number of BusRdx that were reduced drastically between MSI and MESI due to the fact that we have a new bus transaction called Bus Upgr which reduced the bandwidth consumed by these Bus Transaction.

Associativity = 4

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5768	5797	5762	5803
3	Number ofWrites	11942	11710	12383	12108

4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.66%	4.76%	4.56%	4.65%
6	Writebacks toMemory	239	211	239	224
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	6723	6708	6726	6741
9	Interventions	69	47	81	63
10	Invalidations	2014	2034	2008	2020
11	Flushes	114	92	120	88
12	BusRdX issued	716	700	725	714

Associativity = 8

Sr. No.	**Statistics**	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5752	5781	5752	5790
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.65%	4.75%	4.55%	4.64%
6	Writebacks toMemory	170	155	186	157
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	6638	6636	6663	6661
9	Interventions	71	48	82	64
10	Invalidations	2014	2034	2008	2020
11	Flushes	116	93	121	89
12	BusRdX issued	716	700	725	714

Associativity = 16

Sr. No.	**Statistics**	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5741	5772	5741	5780
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.64%	4.74%	4.54%	4.64%
6	Writebacks toMemory	120	101	130	98
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	6577	6573	6596	6592
9	Interventions	71	48	83	64
10	Invalidations	2014	2034	2008	2020

11	Flushes	116	93	122	89
12	BusRdX issued	716	700	725	714

b) MESI

There is a significant reduction in the number of memory transaction mainly because of the fact that we have cache to cache transfer.

Associativity = 4

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5768	5797	5762	5803
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.66%	4.76%	4.56%	4.65%
6	Writebacks toMemory	239	211	239	224
7	Cache-to-Cache Transfers	4402	4434	4401	4401
8	Memory Transaction	1644	1615	1642	1665
9	Interventions	1465	1431	1465	1480
10	Invalidations	2014	2034	2008	2020
11	Flushes	114	92	120	88
12	BusRdX issued	39	41	42	39

Associativity = 8

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5752	5781	5752	5790
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.65%	4.75%	4.55%	4.64%
6	Writebacks toMemory	170	155	186	157
7	Cache-to-Cache Transfers	4389	4422	4393	4395
8	Memory Transaction	1572	1555	1587	1591
9	Interventions	1464	1428	1464	1474
10	Invalidations	2014	2034	2008	2020
11	Flushes	116	93	121	89
12	BusRdX issued	39	41	42	39

Associativity = 16

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5741	5772	5741	5780
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.64%	4.74%	4.54%	4.64%
6	Writebacks toMemory	120	101	130	98
7	Cache-to-Cache Transfers	4379	4415	4386	4386
8	Memory Transaction	1521	1499	1527	1531
9	Interventions	1463	1426	1461	1473
10	Invalidations	2014	2034	2008	2020
11	Flushes	116	93	122	89
12	BusRdX issued	39	41	42	39

c) Dragon

When we move from MESI to dragon we see that the number of memory transaction has gone up again because dragon services a miss from the cache this is because every copy of the block is clean.

Associativity = 4

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5609	5619	5619	5626
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	3	2	2	0
5	Total Miss Rate	4.50%	4.59%	4.41%	4.48%
6	Writebacks toMemory	148	149	142	158
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	5760	5770	5763	5784
9	Interventions	1400	1392	1388	1424
10	Invalidations	0	0	0	0
11	Flushes	3	9	6	6
12	BusRdX issued	0	0	0	0

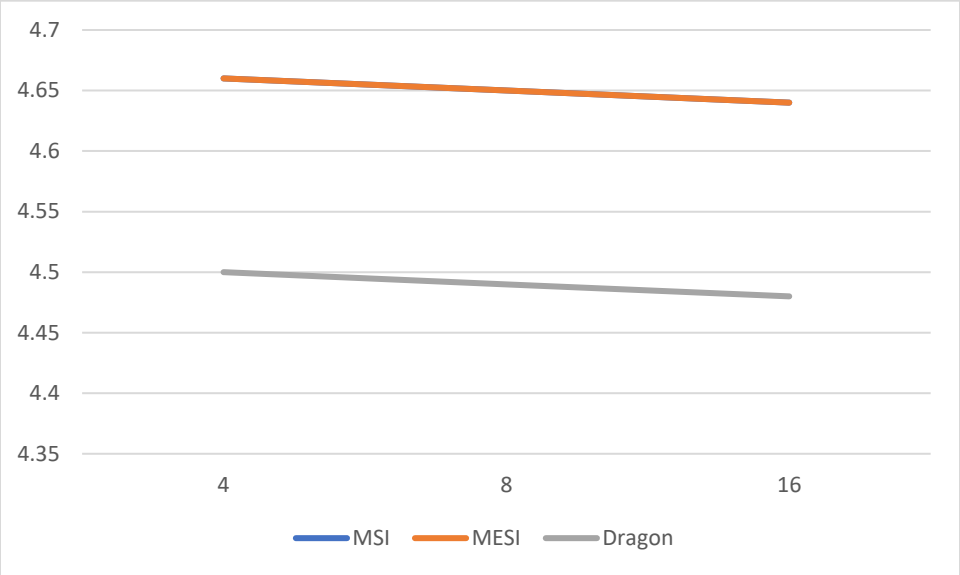
Associativity = 8

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5595	5604	5604	5611
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	3	2	2	0
5	Total Miss Rate	4.49%	4.57%	4.40%	4.47%
6	Writebacks toMemory	107	110	105	123
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	5705	5716	5711	5734
9	Interventions	1398	1387	1387	1417
10	Invalidations	0	0	0	0
11	Flushes	3	9	6	6
12	BusRdX issued	0	0	0	0

Associativity = 16

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5576	5586	5586	5593
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	3	2	2	0
5	Total Miss Rate	4.48%	4.56%	4.39%	4.46%
6	Writebacks toMemory	47	43	62	57
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	5626	5631	5650	5650
9	Interventions	1395	1382	1383	1411
10	Invalidations	0	0	0	0
11	Flushes	3	9	6	6
12	BusRdX issued	0	0	0	0

Miss rate vs Associativity:



Varying the block size and keeping the associativity and the cache size a constant.

Increasing block size can reduce the number of compulsory misses. This is seen across all the three protocols. However, we can see that there is a clear increase in the number of flushes and BusRdx. Since the block size is huge, there can be a write to multiple addresses in a block that triggers so many BusRdx.

a) MSI

Block Size: 64B

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5752	5781	5752	5790
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.65%	4.75%	4.55%	4.64%
6	Writebacks toMemory	170	155	186	157
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	6638	6636	6663	6661
9	Interventions	71	48	82	64
10	Invalidations	2014	2034	2008	2020
11	Flushes	116	93	121	89
12	BusRdX issued	716	700	725	714

Block Size: 128B

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5340	5386	5341	5384
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	40	42	39
5	Total Miss Rate	4.32%	4.43%	4.23%	4.32%
6	Writebacks toMemory	275	250	283	269
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	6344	6347	6369	6386
9	Interventions	119	84	127	110
10	Invalidations	2066	2089	2053	2068
11	Flushes	164	129	166	135
12	BusRdX issued	729	711	745	733

Block Size: 256B

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5023	5070	5004	5084
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	40	42	39
5	Total Miss Rate	4.06%	4.17%	3.96%	4.08%
6	Writebacks toMemory	363	342	383	332
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	6137	6147	6167	6159
9	Interventions	166	133	192	145
10	Invalidations	2132	2157	2107	2148
11	Flushes	211	178	231	170
12	BusRdX issued	751	735	780	743

b) MESI

Block Size: 64B

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5752	5781	5752	5790
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	41	42	39
5	Total Miss Rate	4.65%	4.75%	4.55%	4.64%
6	Writebacks toMemory	170	155	186	157
7	Cache-to-Cache Transfers	4389	4422	4393	4395
8	Memory Transaction	1572	1555	1587	1591
9	Interventions	1464	1428	1464	1474
10	Invalidations	2014	2034	2008	2020
11	Flushes	116	93	121	89
12	BusRdX issued	39	41	42	39

Block Size: 128B

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5340	5386	5341	5384
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	40	42	39
5	Total Miss Rate	4.32%	4.43%	4.23%	4.32%
6	Writebacks toMemory	275	250	283	269
7	Cache-to-Cache Transfers	4124	4155	4115	4125
8	Memory Transaction	1530	1521	1551	1567
9	Interventions	1367	1337	1377	1385
10	Invalidations	2066	2089	2053	2068
11	Flushes	164	129	166	135
12	BusRdX issued	39	40	42	39

Block Size: 256B

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5023	5070	5004	5084
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	39	40	42	39
5	Total Miss Rate	4.06%	4.17%	3.96%	4.08%
6	Writebacks toMemory	363	342	383	332
7	Cache-to-Cache Transfers	3938	3943	3903	3939
8	Memory Transaction	1487	1509	1526	1516
9	Interventions	1283	1284	1321	1309
10	Invalidations	2132	2157	2107	2148
11	Flushes	211	178	231	170
12	BusRdX issued	39	40	42	39

c) Dragon

Block Size: 64B

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5595	5604	5604	5611
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	3	2	2	0
5	Total Miss Rate	4.49%	4.57%	4.40%	4.47%
6	Writebacks toMemory	107	110	105	123
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	5705	5716	5711	5734
9	Interventions	1398	1387	1387	1417
10	Invalidations	0	0	0	0
11	Flushes	3	9	6	6
12	BusRdX issued	0	0	0	0

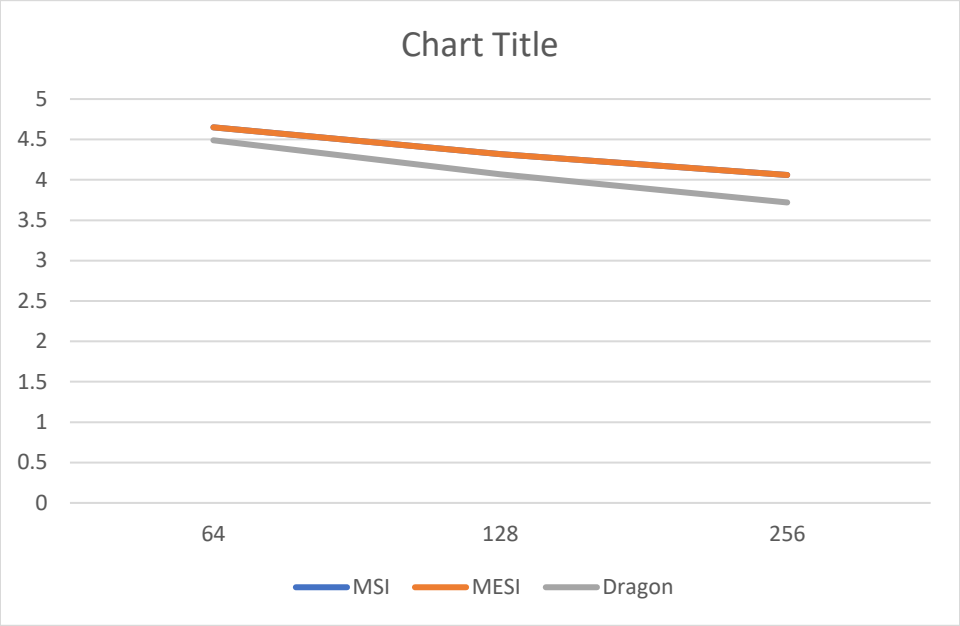
Block Size: 128B

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	5069	5080	5080	5086
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	3	1	2	0
5	Total Miss Rate	4.07%	4.15%	3.99%	4.05%
6	Writebacks toMemory	145	149	145	160
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	5217	5230	5227	5246
9	Interventions	1256	1266	1257	1287
10	Invalidations	0	0	0	0
11	Flushes	3	9	6	9
12	BusRdX issued	0	0	0	0

Block Size: 256B

Sr. No.	Statistics	0	1	2	3
1	Number of Reads	112661	110830	114938	113428
2	Number of Read Misses	4628	4633	4630	4640
3	Number ofWrites	11942	11710	12383	12108
4	Number of Write Misses	3	1	2	0
5	Total Miss Rate	3.72%	3.78%	3.64%	3.70%
6	Writebacks toMemory	198	200	185	196
7	Cache-to-Cache Transfers	0	0	0	0
8	Memory Transaction	4829	4834	4817	4836
9	Interventions	1125	1175	1143	1176
10	Invalidations	0	0	0	0
11	Flushes	3	11	9	9
12	BusRdX issued	0	0	0	0

Miss Rate vs Block Size:

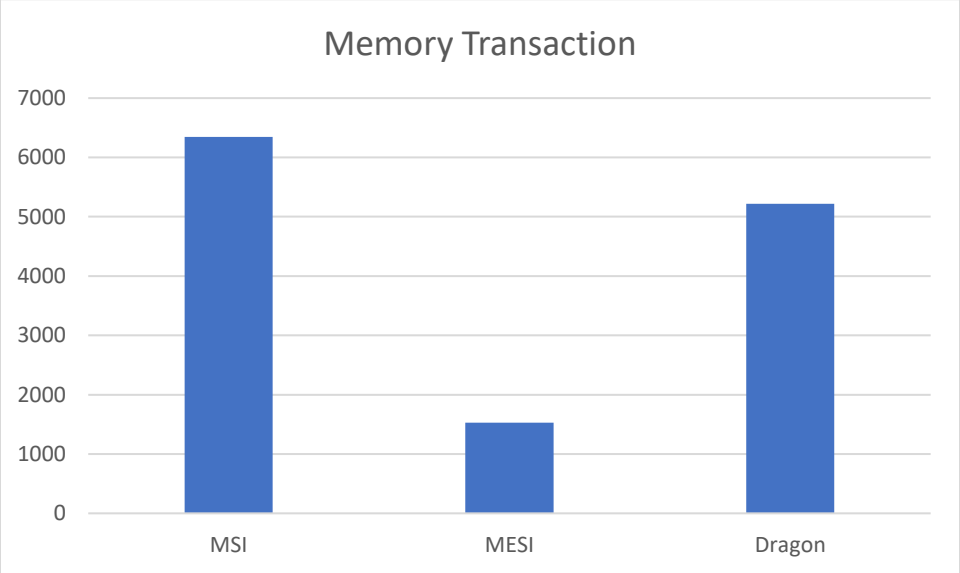


Across different Protocol:

Assume that the following configuration was used and the different metrics across all the protocols was studied too get a fair understanding of what is happening.

Cache Size: 1M
Associativity: 8
Block Size: 128 B
Processor Number: 4
Processor Considered : 0

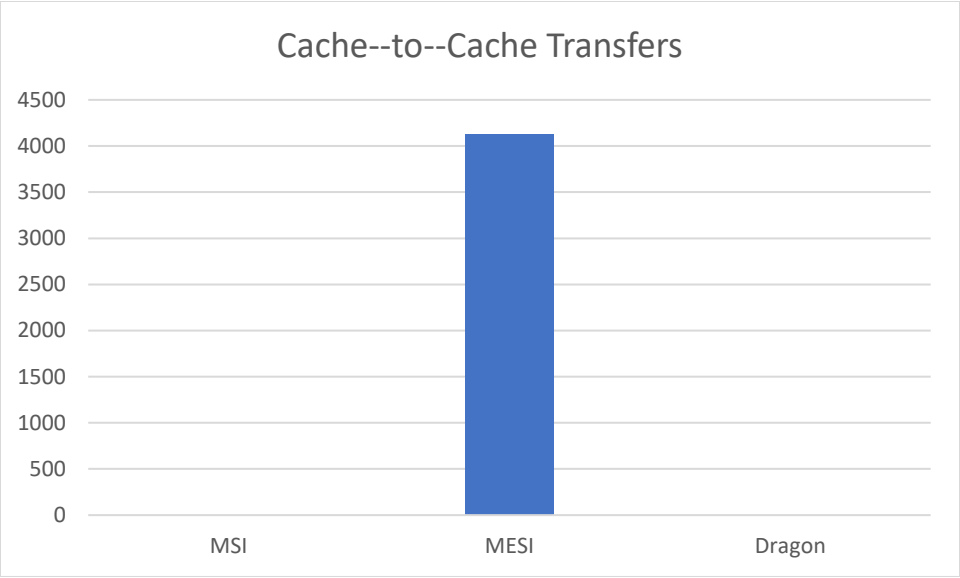
Memory Transaction:



We could see clearly that with introduction of Bus Upgrade and cache to cache protocol, there is a decrease in the number of memory transaction that is incurred by the system. With dragon it is assumed that every data block is clean and as a result a miss is serviced by the memory and not by peer caches, as a result the memory transaction has increased.

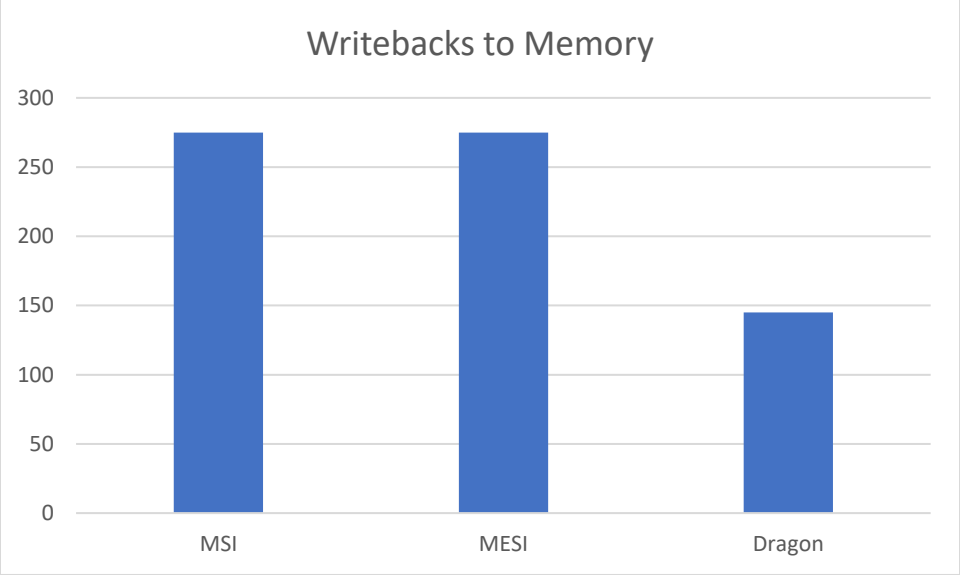
Cache to Cache:

In both MSI and dragon protocol, as mentioned earlier a miss is serviced by the main memory and not by peer caches, this as a result reduces the memory transaction significantly.



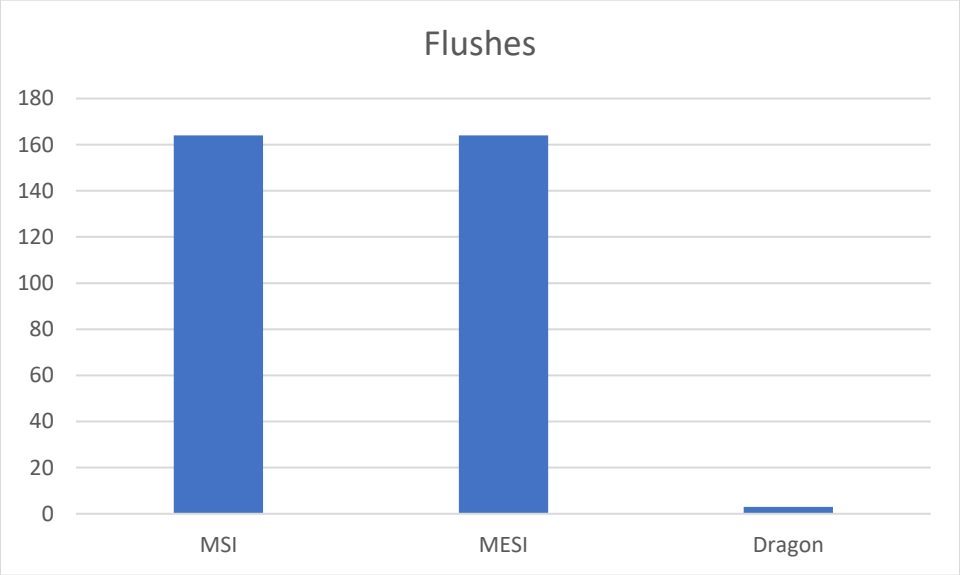
Write Backs:

The addition of an extra shared state has brought down the write backs issued quite significantly. In dragon, the protocol allows the memory to transfer data even when the block is dirty. Only when a block is evicted when it is from its Modified State or its Shared Modified state do we write that block back to the memory.



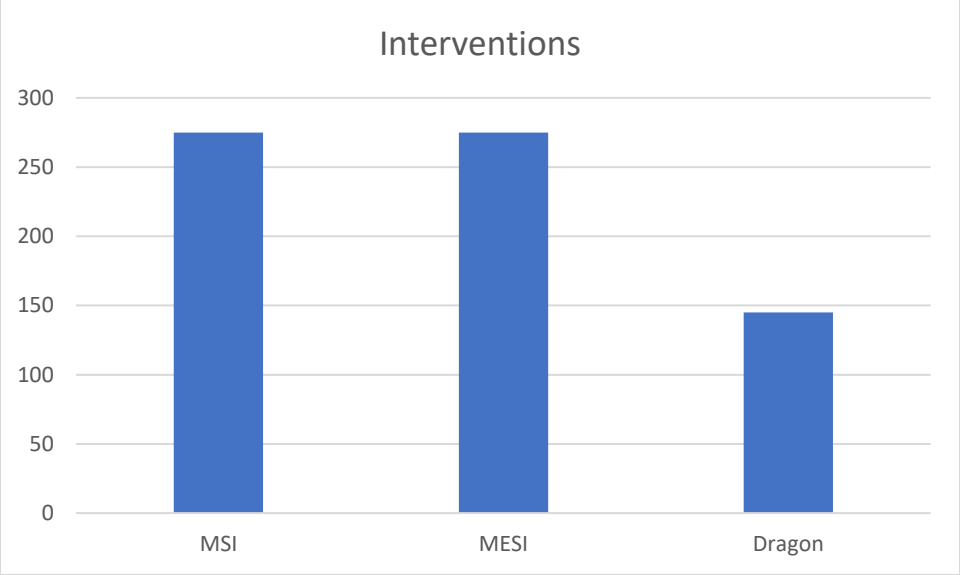
Flushes:

Lack of Bus Rdx in dragon has reduced the number of flushes significantly. MESI and MSI are almost similar protocols, with an extra exclusive state, therefore it is expected for both to have the same number of flushes.



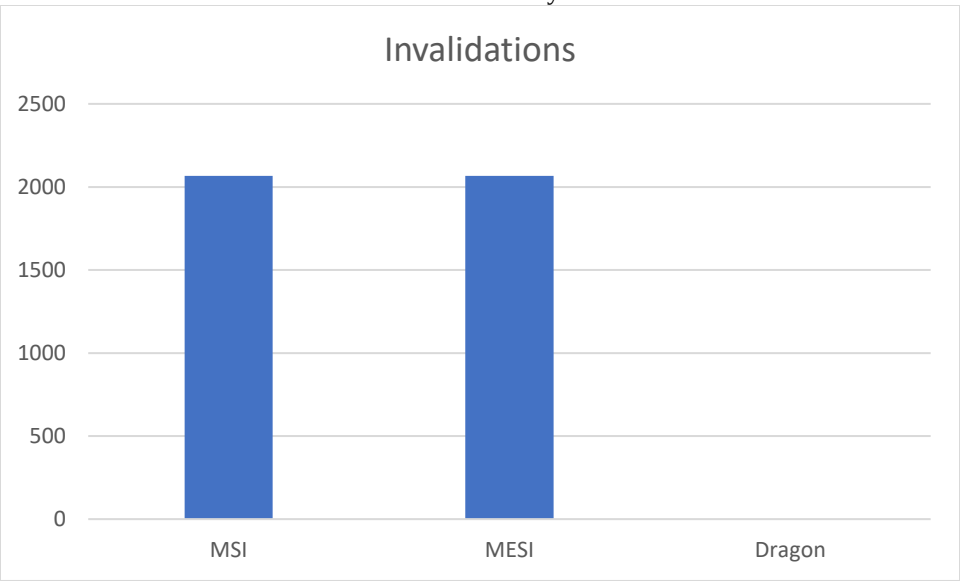
Interventions:

The lack of additional non sharing valid states in MSI can be the main reason why there are so few interventions in it. With MSI a cache block that is read for the first time would be in state shared, irrespective of the fact that the block may not be present in the remaining caches.



Invalidation

An important point to note while discussing invalidation is that, dragon doesn't have an invalid state, and as a result the number of invalidations is zero. As mentioned earlier, both MSI and MESI are similar in a lot of ways and the main difference between the two is the lack of an exclusive state. Hence, it is expected that the number of invalidation in both MSI and MESI to be the same and it is found out that they are.



BusRdx:

A state which allows dirty blocks to be shared might be the main reason why we have no BusRdx in dragon. The main function of a BusRdx is that it invalidates the blocks and the new block is modified by the requesting processor. Since we allow dirty sharing, the issuing processor gets the node after the servicing cache issues and update and moves to state "shared modified". We have implemented the simple version of MSI without its Bus Upgrade transaction. So it is expected that the number of BusRdx in MSI to be significantly greater than the number of BusRdx in MESI.