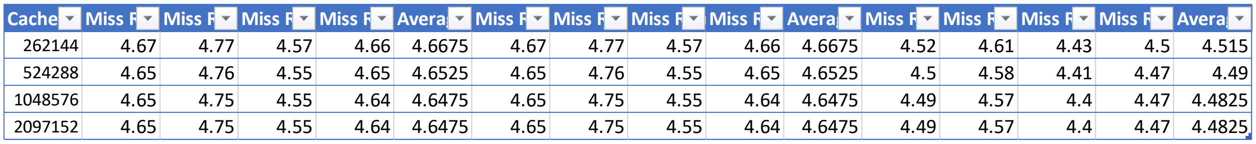
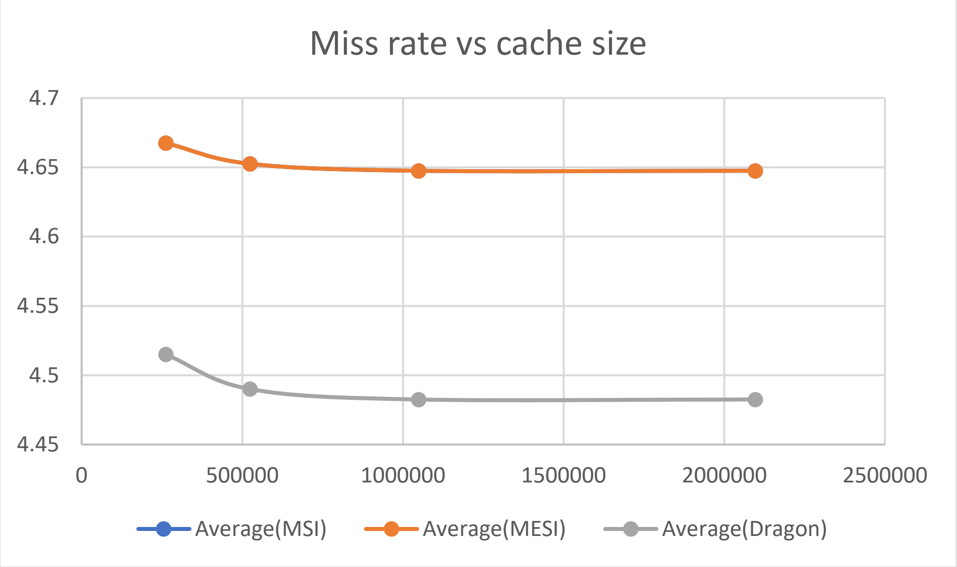
Machine Problem 2 – Coherence Protocols

Effects of cache size



Cache ▾	Miss R ▾	Miss R ▾	Miss R ▾	Miss R ▾	Averag ▾	Miss R ▾	Miss R ▾	Miss R ▾	Miss R ▾	Averag ▾	Miss R ▾	Miss R ▾	Miss R ▾	Miss R ▾	Averag ▾
262144	4.67	4.77	4.57	4.66	4.6675	4.67	4.77	4.57	4.66	4.6675	4.52	4.61	4.43	4.5	4.515
524288	4.65	4.76	4.55	4.65	4.6525	4.65	4.76	4.55	4.65	4.6525	4.5	4.58	4.41	4.47	4.49
1048576	4.65	4.75	4.55	4.64	4.6475	4.65	4.75	4.55	4.64	4.6475	4.49	4.57	4.4	4.47	4.4825
2097152	4.65	4.75	4.55	4.64	4.6475	4.65	4.75	4.55	4.64	4.6475	4.49	4.57	4.4	4.47	4.4825

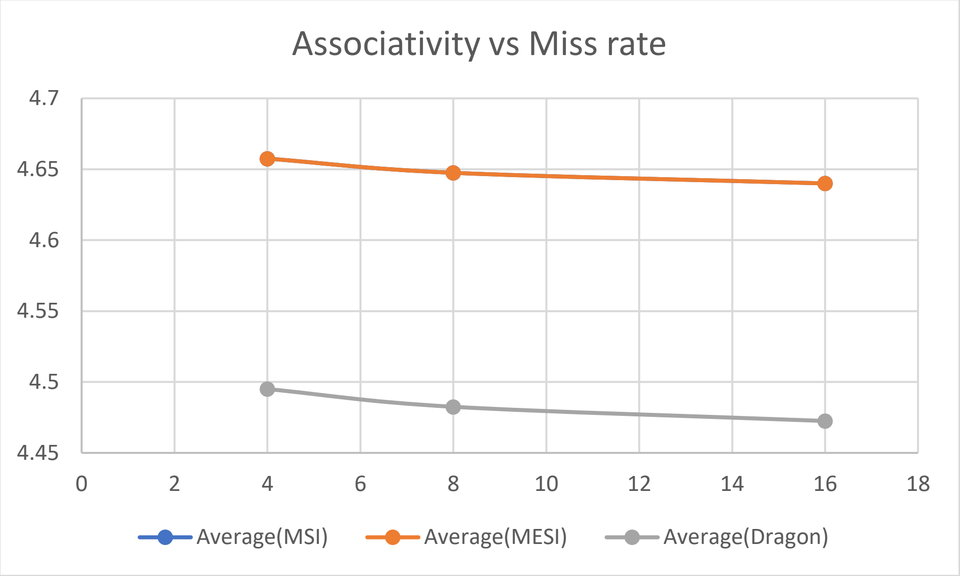


The miss rates for MSI and MESI are identical but we see that the miss rate for Dragon is slightly lower. This can be attributed to the fact that none of the existing values are ever invalidated in Dragon while both MSI and MESI invalidate blocks if they have been written into by another processor. Therefore, MSI and MESI would have a higher number of coherence misses.

It is further observed that the miss rates decrease as cache size increases which can be attributed to the elimination of the capacity misses as the cache size increases.

Effects of cache associativity

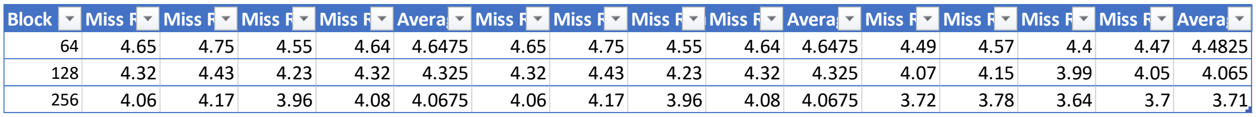
Associ	Miss R	Miss R	Miss R	Miss R	Avera	Miss R	Miss R	Miss R	Miss R	Avera	Miss R	Miss R	Miss R	Miss R	Avera
4	4.66	4.76	4.56	4.65	4.6575	4.66	4.76	4.56	4.65	4.6575	4.5	4.59	4.41	4.48	4.495
8	4.65	4.75	4.55	4.64	4.6475	4.65	4.75	4.55	4.64	4.6475	4.49	4.57	4.4	4.47	4.4825
16	4.64	4.74	4.54	4.64	4.64	4.64	4.74	4.54	4.64	4.64	4.48	4.56	4.39	4.46	4.4725



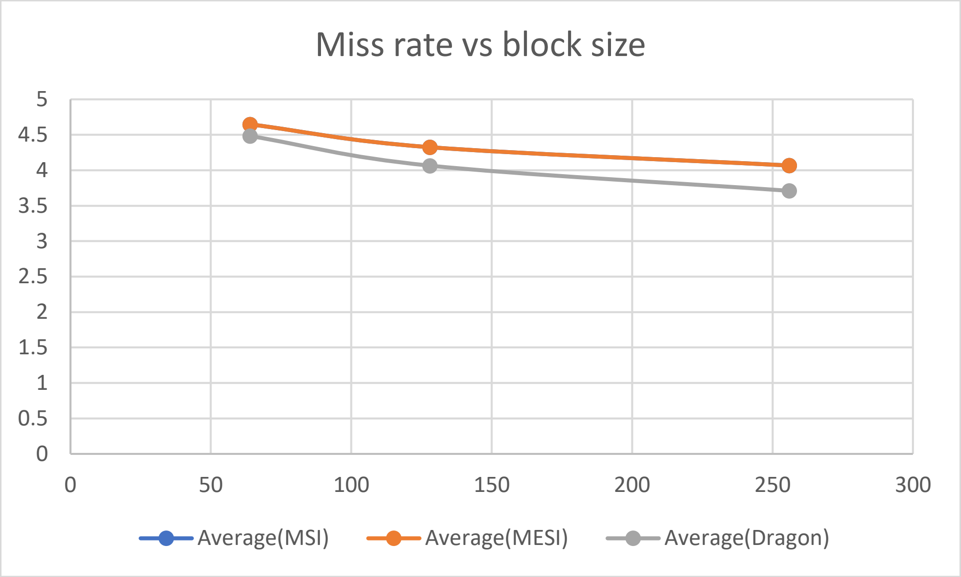
Similar observations can be made here. The overall miss rate is lower for Dragon because it avoids the coherence misses by avoiding invalidations in partner caches.

We are also able to observe a drop in miss rate as the associativity increases which is most probably caused by decreased conflict misses.

Effects of block size



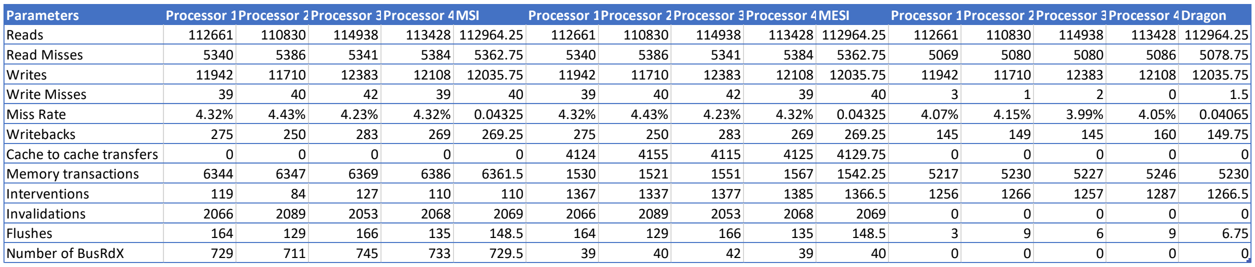
Block	Miss R	Miss R	Miss R	Miss R	Avera	Miss R	Miss R	Miss R	Miss R	Avera	Miss R	Miss R	Miss R	Miss R	Avera
64	4.65	4.75	4.55	4.64	4.6475	4.65	4.75	4.55	4.64	4.6475	4.49	4.57	4.4	4.47	4.4825
128	4.32	4.43	4.23	4.32	4.325	4.32	4.43	4.23	4.32	4.325	4.07	4.15	3.99	4.05	4.065
256	4.06	4.17	3.96	4.08	4.0675	4.06	4.17	3.96	4.08	4.0675	3.72	3.78	3.64	3.7	3.71



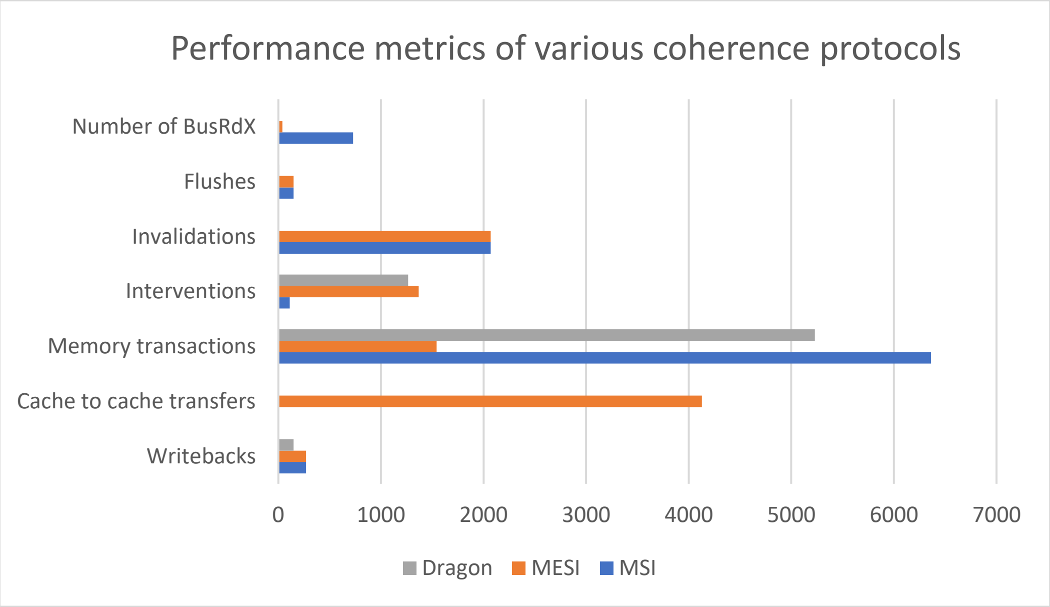
The miss rate drops slightly as the block size increases which can be attributed to the cache being better able to leverage spatial locality through increased block sizes.

Further we still observe that Dragon still has lower miss rates due to it being able to avoid the coherence misses.

Comparison between different protocols (Cache size 1MB, Associativity 8, Block size 128B)



Parameters	Processor 1	Processor 2	Processor 3	Processor 4	MSI	Processor 1	Processor 2	Processor 3	Processor 4	MESI	Processor 1	Processor 2	Processor 3	Processor 4	Dragon
Reads	112661	110830	114938	113428	112964.25	112661	110830	114938	113428	112964.25	112661	110830	114938	113428	112964.25
Read Misses	5340	5386	5341	5384	5362.75	5340	5386	5341	5384	5362.75	5069	5080	5080	5086	5078.75
Writes	11942	11710	12383	12108	12035.75	11942	11710	12383	12108	12035.75	11942	11710	12383	12108	12035.75
Write Misses	39	40	42	39	40	39	40	42	39	40	3	1	2	0	1.5
Miss Rate	4.32%	4.43%	4.23%	4.32%	0.04325	4.32%	4.43%	4.23%	4.32%	0.04325	4.07%	4.15%	3.99%	4.05%	0.04065
Writebacks	275	250	283	269	269.25	275	250	283	269	269.25	145	149	145	160	149.75
Cache to cache transfers	0	0	0	0	0	4124	4155	4115	4125	4129.75	0	0	0	0	0
Memory transactions	6344	6347	6369	6386	6361.5	1530	1521	1551	1567	1542.25	5217	5230	5227	5246	5230
Interventions	119	84	127	110	110	1367	1337	1377	1385	1366.5	1256	1266	1257	1287	1266.5
Invalidations	2066	2089	2053	2068	2069	2066	2089	2053	2068	2069	0	0	0	0	0
Flushes	164	129	166	135	148.5	164	129	166	135	148.5	3	9	6	9	6.75
Number of BusRdX	729	711	745	733	729.5	39	40	42	39	40	0	0	0	0	0



Here we have a comparison between various relevant parameters.

We can see that the highest number of exclusive bus reads is for the MSI protocol. Since the MESI protocol can issue a BusUpgrade instead of a BusRdX when a write occurs in the shared state the number of BusRdX is lower but is made up by the BusUpgrade transactions.

The number of flushes is the same for MSI and MESI because flushes occur whenever we switch from the Modified state which should be equal for both.

Since Dragon doesn't support the invalid state we see that the number of invalidations for this protocol is 0.

We are also able to notice a significantly higher number of interventions for both MESI and Dragon. We are able to observe this because MSI does this only from modified to shared which is significantly lower and points to the relevance of significant amount of accesses to a shared variable in the program which are not written to.

MESI does not suffer from as many memory transactions because it allows cache to cache transfers which decrease the overall number of memory accesses required. Since MSI does not allow this mechanism all its values need to be fetched from memory and therefore pays a heavy penalty in terms of significantly higher number of memory transactions.