Machine Problem – 3
ECE 506 – Architecture of Parallel Computers
Due: Tuesday, 1st December, 2015

Name: Parth Bhogate
NCSU ID: 200108628

For each of the following experiments, the number of processors is taken as 4 and the address trace file used in longTrace. The L1 cache implements WTNA policy, while the L2 cache implements the WBWA policy. Also, L2 cache implements the MESI cache coherence protocol.

1. Varying Cache Size:

Block Size: 64B
L1 Cache Size: 256kB
L1 Cache Associativity: 4
L2 Cache Size: 512 kB
L2 Cache Associativity: 8

Parameter	Processor 0	Processor 1	Processor 2	Processor 3	Average
L1 Read Misses	5798	5829	5803	5832	5815.5
L1 Write Misses	46	43	42	39	42.5
L2 Coherence Misses	2014	2034	2008	2020	2019
Back Invalidations from L2 to L1	1979	2004	1980	1984	1986.75
L1 Cache Fills	5798	5829	5803	5832	5815.5
L1 Miss Rate	4.69%	4.79%	4.59%	4.68%	4.68%

Block Size: 64B
L1 Cache Size: 512kB
L1 Cache Associativity: 4
L2 Cache Size: 1 MB
L2 Cache Associativity: 8

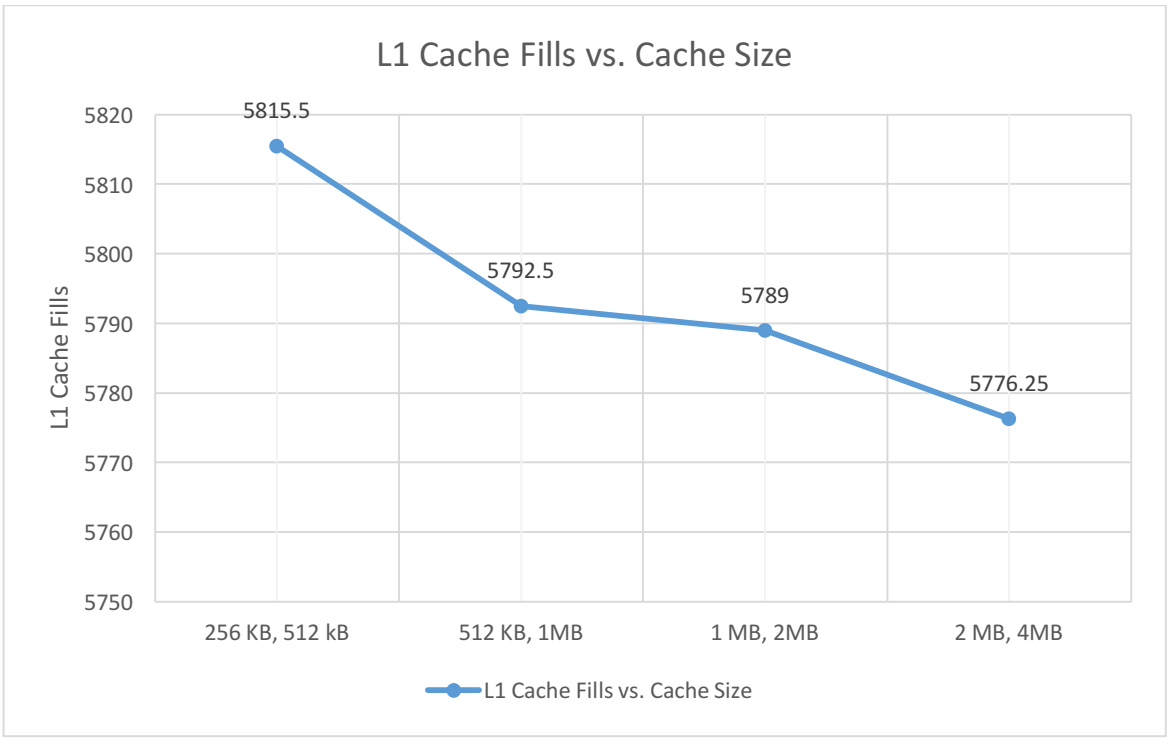
Parameter	Processor 0	Processor 1	Processor 2	Processor 3	Average
L1 Read Misses	5774	5808	5780	5808	5792.5
L1 Write Misses	46	43	42	39	42.5
L2 Coherence Misses	2014	2034	2008	2020	2019
Back Invalidations from L2 to L1	1978	2001	1979	1983	1985.25
L1 Cache Fills	5774	5808	5780	5808	5792.5
L1 Miss Rate	4.67%	4.77%	4.57%	4.66%	4.66%

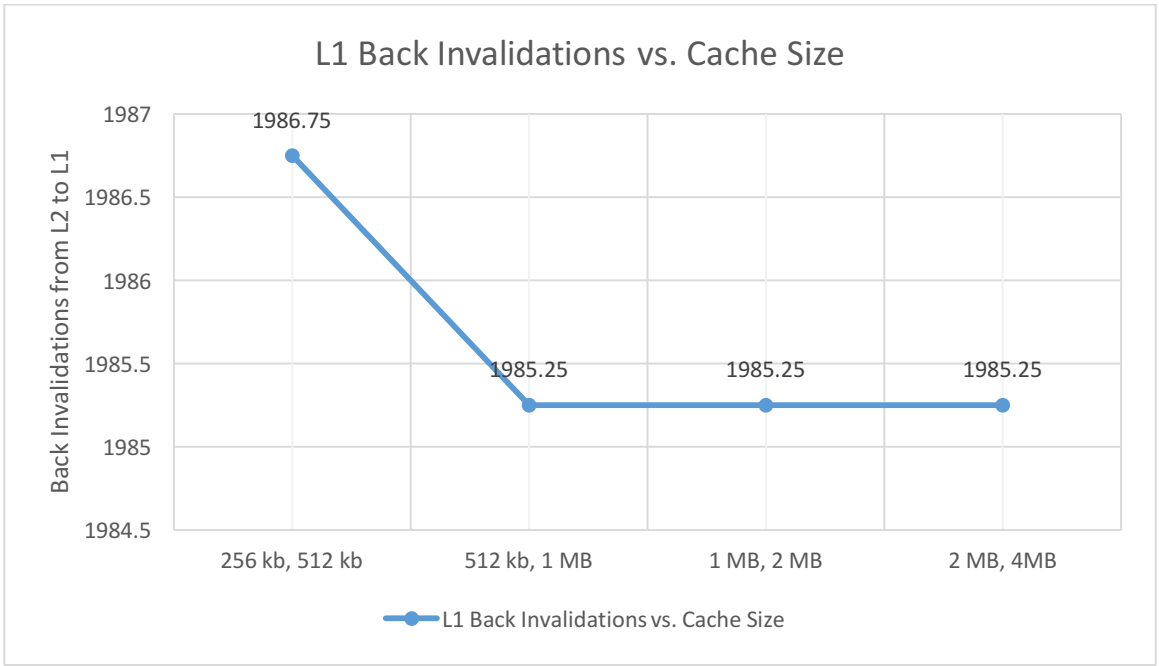
Block Size: 64B
L1 Cache Size: 1 MB
L1 Cache Associativity: 4
L2 Cache Size: 2 MB
L2 Cache Associativity: 8

Parameter	Processor 0	Processor 1	Processor 2	Processor 3	Average
L1 Read Misses	5771	5805	5775	5805	5789
L1 Write Misses	46	43	42	39	42.5
L2 Coherence Misses	2014	2034	2008	2020	2019
Back Invalidations from L2 to L1	1978	2001	1979	1983	1985.25
L1 Cache Fills	5771	5805	5775	5805	5789
L1 Miss Rate	4.67%	4.77%	4.57%	4.66%	4.67%

Block Size: 32B
L1 Cache Size: 2 MB
L1 Cache Associativity: 4
L2 Cache Size: 4 MB
L2 Cache Associativity: 8

Parameter	Processor 0	Processor 1	Processor 2	Processor 3	Average
L1 Read Misses	5754	5791	5765	5795	5776.25
L1 Write Misses	46	43	42	39	42.5
L2 Coherence Misses	2014	2034	2008	2020	2019
Back Invalidations from L2 to L1	1978	2001	1979	1983	1985.25
L1 Cache Fills	5754	5791	5765	5795	5776.25
L1 Miss Rate	4.65%	4.76%	4.56%	4.65%	4.65%


L1 Cache Fills vs. Cache Size
L1 Cache Fills
5820
5810
5800
5790
5780
5770
5760
5750
5815.5
5792.5
5789
5776.25
256 KB, 512 kB
512 KB, 1MB
1 MB, 2MB
2 MB, 4MB
L1 Cache Fills vs. Cache Size


L1 Back Invalidations vs. Cache Size
Back Invalidations from L2 to L1
1987
1986.5
1986
1985.5
1985
1984.5
1986.75
1985.25
1985.25
1985.25
256 kb, 512 kb
512 kb, 1 MB
1 MB, 2 MB
2 MB, 4MB
L1 Back Invalidations vs. Cache Size

Discussion of Trends with varying cache size:
From the graphs, we observe that the number of read misses in L1 decrease as the cache size in increased. This is due to the decrease in the number of conflict and capacity misses as a result of increased cache capacity. Coherence misses might increase with an increase in cache size, but the overall read misses decrease.

Effect of cache size on L1 cache fills:
Since the L1 cache implements Write Through No Allocate (WTNA) policy, we do not allocate a block when the L1 cache encounters a write miss. As a result, a block will be brought in to the L1 cache only when there is a read miss. As stated previously, the number of read misses in L1 decrease as the cache size of L1 and L2 is increased, and thus the number of L1 cache fills also decrease with increase in cache size. This decreasing trend can be observed in the graph above.

Effect of cache size on back invalidations issued from L2 to L1:
L2 cache in our design implements an Inclusive cache policy. Therefore, whenever a cache block is invalidated or evicted from L2 cache, the corresponding block in L1 cache is also invalidated. These invalidations are known as back invalidations.
We observe that as the L1 cache size increases, the L1 total miss rates decreases, but not by a significant amount. Due to the decrease in the L1 miss rate, the amount of interaction with L2 cache is relatively lesser. As a result, there will be fewer evictions in L2 cache and fewer bus transactions which implies that the back invalidations because of evictions and coherence invalidations in the native L2 cache will also decrease, though by a small amount only. This is reflected in the above graph.

2. Varying Block Size

Block Size: 64B
L1 Cache Size: 2 MB
L1 Cache Associativity: 4
L2 Cache Size: 4 MB
L2 Cache Associativity: 8

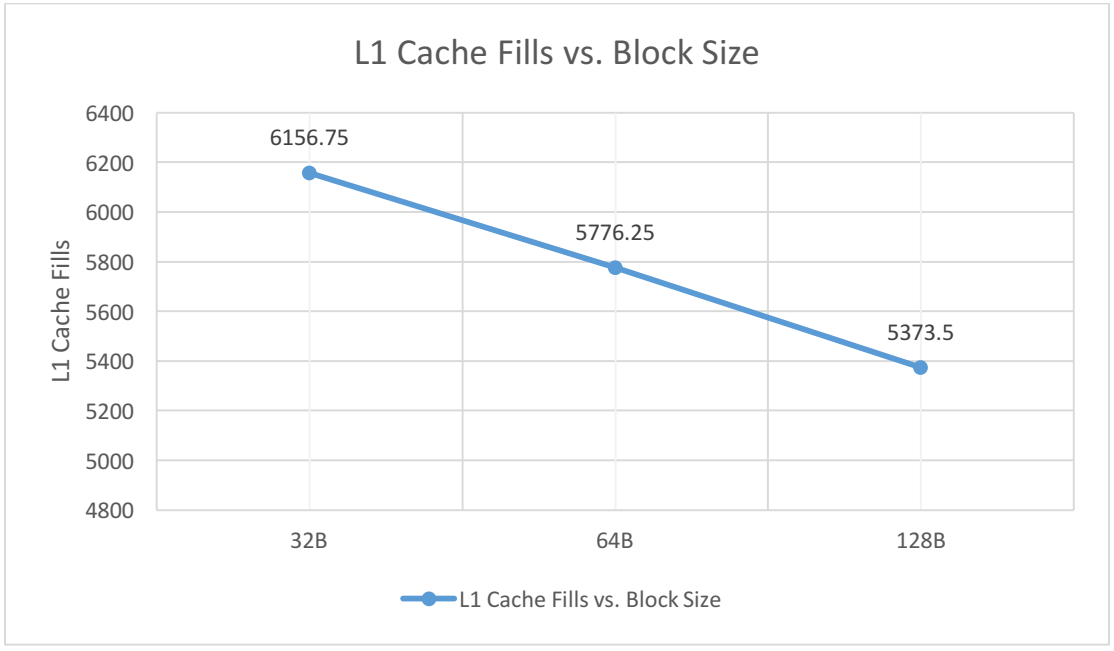
Parameter	Processor 0	Processor 1	Processor 2	Processor 3	Average
L1 Read Misses	6149	6170	6144	6164	6156.75
L1 Write Misses	49	44	44	41	44.5
L2 Coherence Misses	2006	2023	1994	2009	2008
Back Invalidations from L2 to L1	1970	1990	1965	1972	1974.25
L1 Cache Fills	6149	6170	6144	6164	6156.75
L1 Miss Rate	4.97%	5.07%	4.86%	4.94%	4.95%

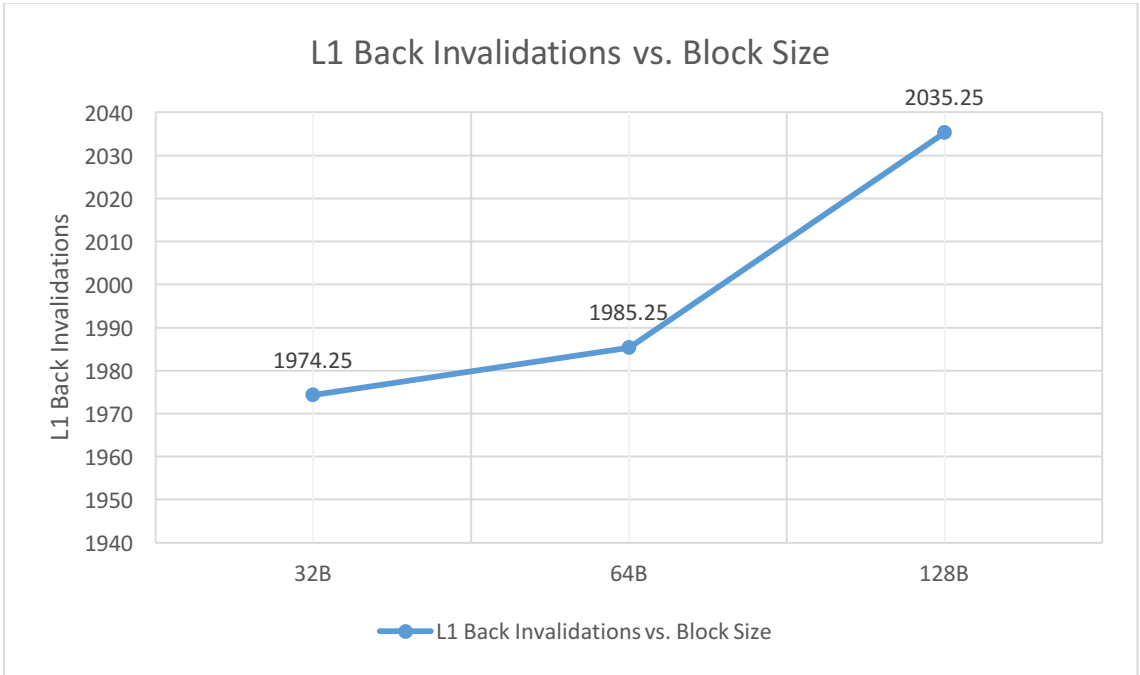
Block Size: 64B
L1 Cache Size: 2 MB
L1 Cache Associativity: 4
L2 Cache Size: 4 MB
L2 Cache Associativity: 8

Parameter	Processor 0	Processor 1	Processor 2	Processor 3	Average
L1 Read Misses	5754	5791	5765	5795	5776.25
L1 Write Misses	46	43	42	39	42.5
L2 Coherence Misses	2014	2034	2008	2020	2019
Back Invalidations from L2 to L1	1978	2001	1979	1983	1985.25
L1 Cache Fills	5754	5791	5765	5795	5776.25
L1 Miss Rate	4.65%	4.76%	4.56%	4.65%	4.66%

Block Size: 128B
L1 Cache Size: 2 MB
L1 Cache Associativity: 4
L2 Cache Size: 4 MB
L2 Cache Associativity: 8

Parameter	Processor 0	Processor 1	Processor 2	Processor 3	Average
L1 Read Misses	5347	5399	5357	5391	5373.5
L1 Write Misses	46	41	42	39	42
L2 Coherence Misses	2066	2089	2053	2068	2069
Back Invalidations from L2 to L1	2030	2056	2024	2031	2035.25
L1 Cache Fills	5347	5399	5357	5391	5373.5
L1 Miss Rate	4.33%	4.44%	4.24%	4.33%	4.34%


L1 Cache Fills vs. Block Size
6400
6200
6000
5800
5600
5400
5200
5000
4800
L1 Cache Fills
6156.75
5776.25
5373.5
32B
64B
128B
L1 Cache Fills vs. Block Size


L1 Back Invalidations vs. Block Size
2040
2030
2020
2010
2000
1990
1980
1970
1960
1950
1940
L1 Back Invalidations
2035.25
1985.25
1974.25
32B
64B
128B
L1 Back Invalidations vs. Block Size

Discussion of Trends with varying block size:
As we increase the block size we observe that the number of read misses decreases. As we increase the block size, greater number of consecutive memory blocks are brought in to the cache block at a time. This effect reduces the number of read misses, and thus it serves a similar purpose like prefetching. Thus, spatial locality increases leading to a decrease in overall miss rate as well as the number of read misses.

Effect of block size on L1 cache fills:
Since the L1 cache implements Write Through No Allocate (WTNA) policy, we do not allocate a block when the L1 cache encounters a write miss. As a result, a block will be brought in (filled) to the L1 cache only when there is a read miss. As was explained previously in the general trend discussion, as the L1 and L2 block size is increased, the number of read misses in L1 decreases. Since the number of read misses decrease, the number of L1 cache fills will also decrease as is reflected in the graph above.

Effect of block size on back invalidations issued from L2 to L1:
Back invalidations from L2 to L1 are issued either when a block is evicted from L2 or when a block in L2 is invalidated on account of BusRdX or BusUpgr because of the MESI protocol. The coherence invalidations can be issued either when there is True sharing miss, or when there is a False sharing miss. True sharing misses occur when multiple threads share the same data item (byte/word). False sharing misses that occur when multiple threads share different data items located in different bytes/words located in a single cache block. As we increase the block size of a cache, the true sharing misses decrease while the false sharing misses increase. The false sharing misses increase since once a block is evicted, more bytes that are grouped together in the block are also evicted. The true sharing misses decrease because spatial locality in the program is exploited more efficiently with a larger block size. As observed in the graph above, the number of back invalidations increase with increasing block size on account of the increase in the number of false sharing coherence invalidations coming from other processors.