

Jared Raiola

CS4290 Project 3

Report

7/30/20

Tracing Experiments:

The intent of using different cache coherence protocols is to maintain up to date copies of data in each cache and avoid resource contention while maintaining a fast run time, high cache to cache transfer count and keeping cache misses to a minimum. With this in mind, each experiment was run to find the best performing protocol(s).

Experiment 1:

	Run Time (Cycles)	Cache Misses	Cache Accesses	Silent Upgrades	Cache-to- Cache Transfers
MSI	317	7	12	0	4
MOSI	217	7	12	0	5
MESI	317	7	12	0	4
MOESI	217	7	12	0	5

In experiment 1, each protocol performed about the same across the board. The MOSI and MOESI protocols performed the best with the lowest run times and the highest transfer count. This means that the trace in this experiment needs to take advantage of the function of the owner state, as the cache in the owner state has the most up-to-date copy, claiming the responsibility for supplying data and updating memory, regardless of whether the caches have the same data or not, focusing on reducing writebacks.

Experiment 2:

	Run Time (Cycles)	Cache Misses	Cache Accesses	Silent Upgrades	Cache-to- Cache Transfers
MSI	2367	30	104	0	7
MOSI	1167	30	104	0	19
MESI	2267	30	104	1	8
MOESI	975	31	104	1	22

In experiment 2, the MOSI and MOESI protocols performed the best. MOSI had the second lowest runtime with more cache transfers than MSI and MESI, and MOESI had the lowest runtime with an extra cache miss and the highest cache transfer count. This means that this trace needs to take advantage of the owner state, which focuses on reducing the amount of writebacks made.

Experiment 3:

	Run Time (Cycles)	Cache Misses	Cache Accesses	Silent Upgrades	Cache-to- Cache Transfers
MSI	3723	56	200	0	20
MOSI	3723	56	200	0	20
MESI	2607	48	200	8	23
MOESI	2607	48	200	8	23

In experiment 3, the best performing protocols were MESI and MOESI. These protocols had the lowest run time and lowest cache misses, with the highest transfers and made use of silent upgrades. This means that the trace needs to take advantage of the exclusive state, in order to reduce bus transactions by performing upgrades silently since the exclusive state recognizes that the block is exclusively owner by that cache and only needs to be modified in that cache.

Experiment 4:

	Run Time (Cycles)	Cache Misses	Cache Accesses	Silent Upgrades	Cache-to- Cache Transfers
MSI	2265	27	60	0	5
MOSI	1869	29	60	0	11
MESI	1549	20	60	2	5
MOESI	951	20	60	2	11

In experiment 4, the best performing protocol was MOESI. This protocol had the lowest run time and lowest cache misses, with the highest transfers and made use of silent upgrades. This means that the trace needs to take advantage of the exclusive state and the owner state, in order to reduce bus transactions and number of writebacks made.

Experiment 5:

	Run Time (Cycles)	Cache Misses	Cache Accesses	Silent Upgrades	Cache-to- Cache Transfers
MSI	1661	21	37	0	5
MOSI	1261	21	37	0	9
MESI	1561	21	37	0	6
MOESI	1161	21	37	0	10

In experiment 5, the MOSI and MOESI protocols performed the best. MOSI had the second lowest runtime with more cache transfers than MSI and MESI, and MOESI had the lowest runtime with an extra cache miss and the highest cache transfer count. This means that this trace needs to take advantage of the owner state, which focuses on reducing the amount of writebacks made.

Experiment 6:

	Run Time (Cycles)	Cache Misses	Cache Accesses	Silent Upgrades	Cache-to- Cache Transfers
MSI	7775	87	747	0	12
MOSI	6975	87	747	0	20
MESI	5235	65	747	22	15
MOESI	4435	65	747	22	23

In experiment 6, the best performing protocol was MOESI. This protocol had the lowest run time and lowest cache misses, with the highest transfers and made use of silent upgrades. This means that the trace needs to take advantage of the exclusive state and the owner state, in order to reduce bus transactions and number of writebacks made.

Experiment 7:

	Run Time (Cycles)	Cache Misses	Cache Accesses	Silent Upgrades	Cache-to- Cache Transfers
MSI	6459	79	952	0	17
MOSI	5359	79	952	0	28
MESI	3993	55	952	24	17
MOESI	3991	37	952	22	28

In experiment 7, the best performing protocol was MOESI. This protocol had the lowest run time and lowest cache misses, with the highest transfers and made use of silent upgrades. This means that the trace needs to take advantage of the exclusive state and the owner state, in order to reduce bus transactions and number of writebacks made.

Experiment 8:

	Run Time (Cycles)	Cache Misses	Cache Accesses	Silent Upgrades	Cache-to- Cache Transfers
MSI	9447	110	800	0	18
MOSI	8447	110	800	0	28
MESI	6739	91	800	19	26
MOESI	5639	91	800	19	37

In experiment 8, the best performing protocol was MOESI. This protocol had the lowest run time and lowest cache misses, with the highest transfers and made use of silent upgrades. This means that the trace needs to take advantage of the exclusive state and the owner state, in order to reduce bus transactions and number of writebacks made.

Architecting A System:

When determining which protocol would be best in a system that placed equal importance on every provided program, it is necessary to take a look at each state in each protocol and what it does. M: the modified state, recognizes that this current cache is the only one that holds the copy of the data and the copy is dirty. O: the owned state, recognizes that the cache has the most up-to-date copy of the data and is responsible for supplying and updating memory, regardless of if other caches have a copy of the data or not. E: the exclusive state, recognizes that the current cache has the only copy of the data and it's the most up-to-date copy of the data. S: the shared state, recognizes that the data has multiple copies across multiple caches and all memory is up to date. I: the invalid state, recognizes that the cache either does not have a copy of the data or that the copy of the data is dirty. With all of these states in mind, we now can focus on our goal of reducing the run time of the protocol, reducing the amount of cache misses and reducing the amount of writebacks and bus transactions by use of cache transfer and silent upgrades respectively. When regarding the data in the **Tracing Experiments** portion of this report, we can see that across the board, no matter whether the trace program must take advantage of reducing writebacks or reducing bus transactions, the protocol that combines all of these features, MOESI, performs best across the board for all programs. Since its performance level is the best for every single experiment tested, it's the most obvious choice to use in system that places equal emphasis on all of these features.

Limitations of The Simulator:

One of the major limitations of the simulator is the fact that it is a singular programming not running in parallel with itself in a threaded fashion. This means that it does operations sequentially, that a real-life system would in parallel. This affects performance as it cannot truly test multiple cache accesses to memory at the same time. Another major limitation of the simulator is the assumption that data replies are only sent when you are responsible for supplying the data. This means that if data is wrongly sent, there is no safety net to recognize that the incorrect data was sent into memory.

Enhancements:

This simulator could be enhanced by implementing multithreading, this would allow multiple threads to act as different caches and accurately test reading and writing into memory at the same time. Another enhancement that could be made to the simulator is to test for wrong data being supplied by caches.