Sarvagya Vaish

ECE 3056

Memory Hierarchy Design

1. Overview of the Simulation Process

Assumptions

I ran the simulations against four of the six traces given, namely bubble, merge, shuf and sieve. Stream and random were left off because they do not represent actual data sets and tend to skew the results.

Cache Configurations Considered

I considered all possible combinations of the following parameters for both L1 and L2:

```
cacheSizeInKb = [4, 8, 16, 32, 64, 128];

associativity = [1, 2, 4, 8, 16, 32];

lineSize = [16, 32, 64, 128, 256, 512, 1024];
```

Before actually running the cachesim simulation on a particular combination of L1 and L2 caches I made sure that they satisfy the 1.5 Megabits of SRAM constraint. I also checked to make sure that the block size of L2 was greater than or equal to the L1 block size.

This helped reduce the number of simulations. I ran simulations for 30383 different parameters of L1 and L2 caches. Each configuration was simulated against four traces, namely bubble, merge, shuf and sieve. In total I ran 121532 simulations. This was accomplished in under 5 hrs using 10 Amazon EC2 instances. Also, it helped to parallelize the simulations instead of running them serially on one core.

2. Simulation Results

Caches with only one level of cache (L1) and two levels of cache (L1 and L2) were simulated. The reason for this separation is that cache systems available in the market are divided up into different levels, some dedicated to single cores and some shared between all of them.

Two level caches

The following table describes the best cache configurations for different conditions considering two levels of cache:

Optimization Metric	L1 Block Size	L1 Cache Size	L1 Associativity	L2 Block Size	L2 Cache Size	L2 Associativity	Energy (μJ)	Timing (ms)
$Energy\ (joules)$	**16**	**4096**	**2**	**16**	**16384**	**16**	**291.79**	**27.16**
$Timing\ (sec)$	16	4096	2	16	131072	2	326.01	24.97
$Joules^3 * sec$	16	4096	2	16	16384	8	292.13	26.79
$Joules * sec^3$	16	4096	2	16	131072	2	326.01	24.97

The first two rows in the above figure describe the baseline metrics. To get a better picture I added the last three rows.

$Joules^3 * sec$ is used to weight energy more than timing in the optimization. By sacrificing 1.82ms on run time, the cache was able to save 33.88 μJ. This sort of cache configuration would be used in consumer electronics for example laptops, where one can sacrifice timing to be more conservative with energy.

$Joules * sec^3$ is used to weight timing more than energy in the optimization. By consuming 34.22 μJ more, the cache was able to save 2.19ms. This sort of cache configuration would be used in computers with computationally heavy workload, where one is less limited by energy.

One level caches

The following table describes the best cache configurations for different conditions considering one level of cache:

Optimization Metric	L1 Block Size	L1 Cache Size	L1 Associativity	Energy (μJ)	Timing (ms)
$Energy\ (joules)$	16	16384	4	218.65	44.38
$Timing\ (sec)$	16	4096	2	233.77	28.09

The energy consumption is actually significantly lesser than the best two level cache, 218.65 μJ as compared to 291.79 μJ. This is a 25.1% reduction in energy. But this comes at the cost of timing which increases to 44.38ms from 27.16ms (63.4% increase). This type of cache configuration would be used in devices that really need to optimize for energy and are not running heavy computational applications, for example mobile phones and tablets.

3. Comparing Memory Systems

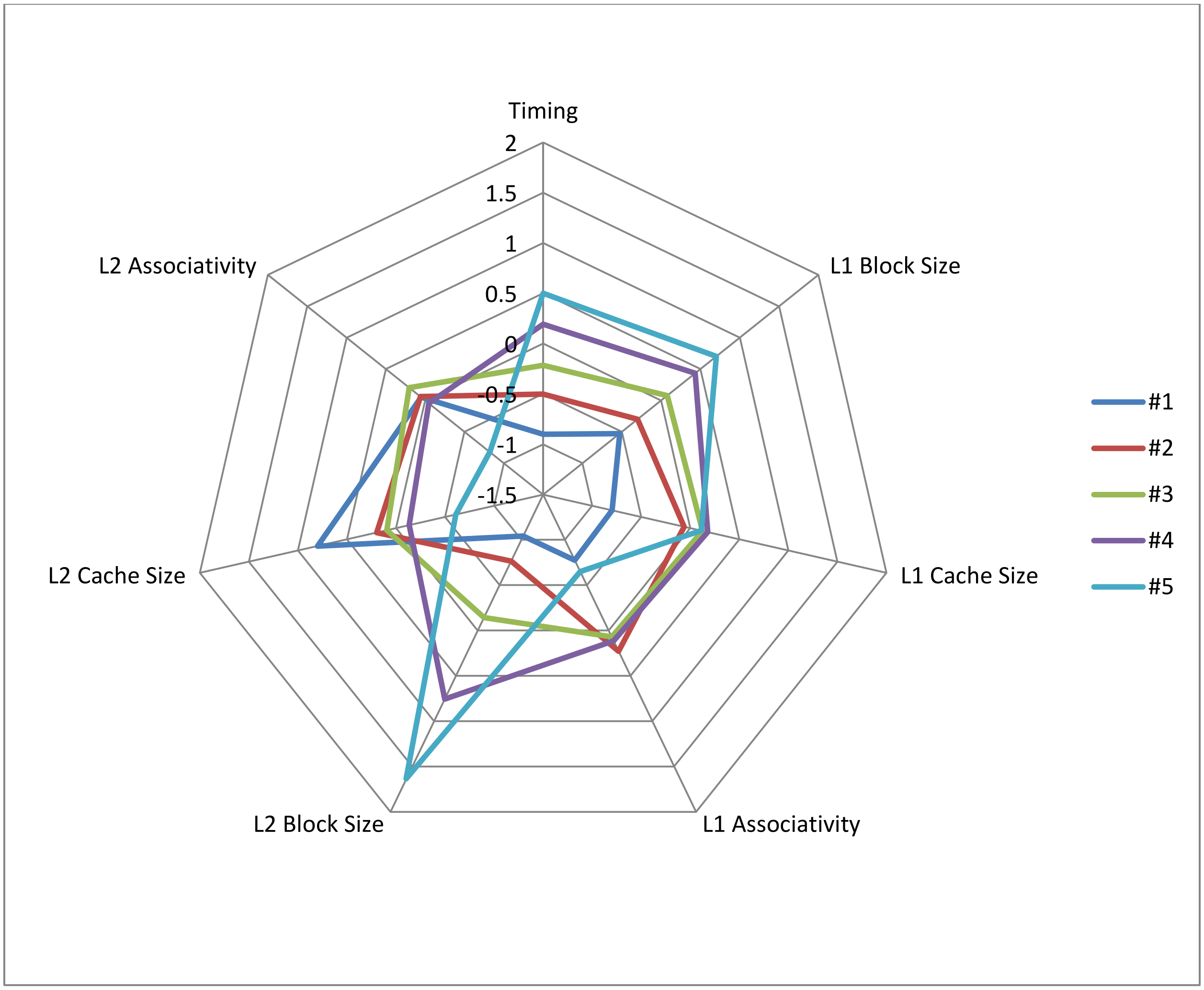
The following points are worth noting while comparing these simulation results with standard cache memory systems available in the market:

- On the Intel Core i7, L1 cache size (64K) is smaller than L2 (1MB). This was true for the simulation results too *(Ref 1)*.
- On the Intel Core i7, L1 is 8 way associative and so is L2. According to the simulations, for maximum energy optimization, the associativities are comparable too *(Ref 2)*.
- I am not 100% sure about the block sizes but from the graphs in *(Ref 3)* smaller block sizes perform better than larger ones which is consistent with the simulation results.

4. Trends

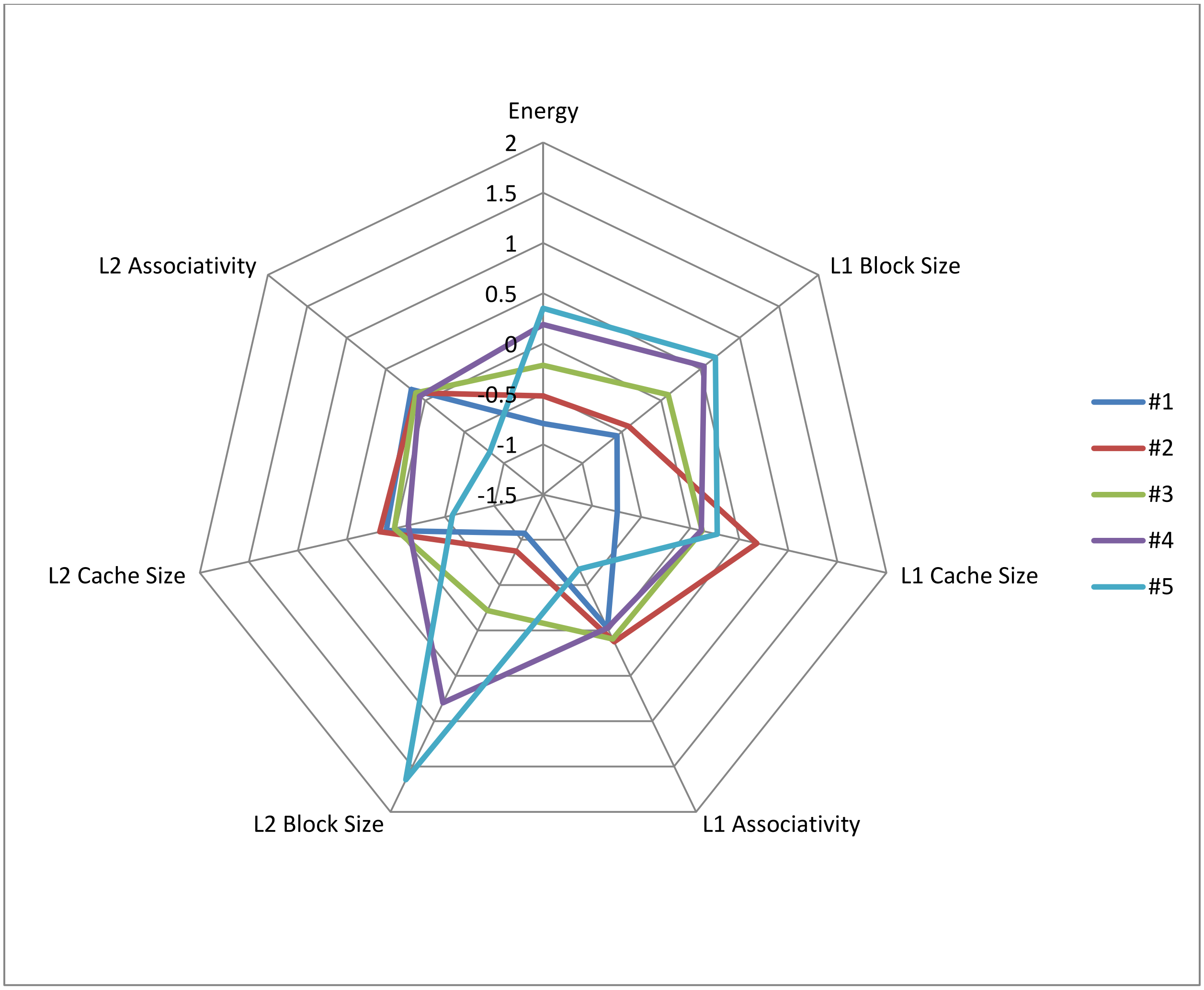
The following graphs show how varying the 6 parameters (L1 Block Size, L1 Cache Size, L1 Associativity, L2 Block Size, L2 Cache Size, L2 Associativity) for a 2 level cache affect different metrics. These plots were constructed by taking ordering the data based on the metric being considered and then taking the average of 5 equally spaced sets. The numbers in the legend of the graphs refer to the cache's ranking (i.e. 1 is better than 2 and so forth).

Trends in Timing



The main trend visible above is that timing improves as L1 cache size decreases and L2 cache size increases. Also smaller the L2 block size, the better the timing. But we also know that L2 block size needs to be bigger than L1 block size. From that I conclude that the L2 block size should ideally be close to the L1 block size for better timing. This point is also reinforced by the table on the previous page which has same block sizes for L1 and L2 under all conditions.

Trends in Energy



This graph shows similar trends in terms of cache sizes and block sizes. The only difference is the associativity. As compared to the previous graph, for better energy, L1 and L2 associativities should be comparable.

5. References

(1) http://www.legitreviews.com/article/824/1/

(2) http://techreport.com/review/15818/intel-core-i7-processors

(3) http://techreport.com/review/15818/intel-core-i7-processors/5