SOFTWARE SYSTEMS LAB (COP 701

2020MCS2444: Aayush Kumar Singh

November 2020

1 CACHE SIMULATOR: PART ONE

1.1 EVALUATE CPI:

In this part we calculated the Cycles per Instruction(CPI) for a set of benchmarks on X86 with configuration given below:

CPU Models: TimingSimpleCPU, DerivO3CPU and MinorCPU.
Cache levels: Two levels.
Unified caches: L1 data and L1 instruction caches, unified L2 cache.
Size: 128KB L1 data cache, 128KB L1 instruction cache, 1MB L2 cache.-
Associativity: Two-way set-associative L1 caches, Direct-mapped L2 cache.
Block size: 64 bytes (applied to all caches).
Block replacement policy: Least Recent Used policy (LRU)

For each benchmark, CPI is calculated using the following formula:
Let L1ICM, be L1 Instruction Cache Miss Number
L1DCM, be L1 Data Cache Miss Number
L2CM, be L2 Cache Miss Number
L1 miss penalty = 6 cycles
L2 miss penalty = 50 cycles
N, be number of instructions
then,

CPI $= 1 + \frac{(L1ICM+L1DCM)*6+(L2CM)*50}{N}$

Benchmarks used:

1. 401.bzip2
2. 429.mcf
3. 456.hmmer
4. 458.sjeng
5. 470.lbm

In this part each benchmark was run for **5*10^8 instructions.**

Each benchmark took 45*mins* approx to execute the given number of instructions above on a single CPU type. The respective cache miss number was obtained from **stats.txt** file.

CPI FOR BENCHMARK ONE: 401.bzip2

i. TimingSimpleCPU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
117807	622	117256	1.01315

ii. DerivO3PU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
5103	651	5480	1.00062

iii. MinorCPU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
2.09813e+06	1340	740463	1.09924

CPI FOR BENCHMARK TWO: 429.mcf

i. TimingSimpleCPU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
15283	476	15588	1.00175

ii. DerivO3PU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
629	729	980	1.00011

iii. MinorCPU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
1.03462e+07	916	6.572e+06	1.78137

CPI FOR BENCHMARK THREE: 456.hmmer

i. TimingSimpleCPU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
1766	1272	3026	1.00034

ii. DerivO3PU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
895	1544	1617	1.00019

iii. MinorCPU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
144966	1775	7462	1.00251

CPI FOR BENCHMARK FOUR: 458.sjeng

i. TimingSimpleCPU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
1300	336	1635	1.00018

ii. DerivO3PU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
526	512	798	1.00009

iii. MinorCPU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
1.67207e+07	5475	8.38099e+06	2.03881

CPI FOR BENCHMARK FIVE: 470.lbm

i. TimingSimpleCPU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
10866	610	11475	1.00129

ii. DerivO3PU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
953	951	1552	1.00018

iii. MinorCPU

L1ICache Misses	L1DCache Misses	L2Cache Misses	CPI
1.40702e+07	1023	7.35711e+06	1.90456

1.2 PART TWO: OPTIMIZE CPI FOR EACH BENCHMARK

In this we determined the optimal cache configuration for each of the 5 benchmark i.e, the one which generates lowest CPI.
To do this each benchmark was run by varying the following things:

i. L1 Data Cache Size [64kB 128kB 256kB]
ii. L1 Instruction Cache Size [64kB 128kB 256kB]
iii. L2 Cache Size [1MB 2MB 4MB 8MB]
iv. Block Size/ Cache line size = [16 32 64]
v. Associativity of L1 cache = [1 2 4]
vi. Associativity of L2 cache [1 2 4]

Everything else was set to it's default value. Since, there are total of 972 configurations possible with above variations, here we just present the ones which have best among them. Each benchmark was run for $5 * 10^7$ instructions.

Each benchmark took approx 4 mins to run for those many instructions. Then we extracted the miss numbers to calculate CPI like we did in previous section. After calculating the CPI we plotted respective graph to analyse the CPI by varying combination of factors and keeping other factors constant at maximum that is in our range.

i. For benchmark: 401.bzip2

L1I Cache Size	L1D Cache Size	L2 Cache Size	Block Size	Assoc. of L1	Assoc. of L2	CPI
256	256	8	64	4	4	1.12802
127	256	8	64	4	4	1.12802
64	256	8	64	4	4	1.12802
256	127	8	64	4	4	1.12803
127	127	8	64	4	4	1.12803
64	127	8	64	4	4	1.12803
256	64	8	64	4	4	1.12803
127	64	8	64	4	4	1.12803
64	64	8	64	4	4	1.12803
256	256	8	64	2	4	1.12803
127	256	8	64	2	4	1.12803
64	256	8	64	2	4	1.12803
256	256	8	64	1	4	1.12803
256	127	8	64	2	4	1.12804
127	127	8	64	2	4	1.12804
64	127	8	64	2	4	1.12804
127	256	8	64	1	4	1.12804
256	64	8	64	2	4	1.12805
64	256	8	64	1	4	1.12805
127	64	8	64	2	4	1.12805
64	64	8	64	2	4	1.12805
256	127	8	64	1	4	1.1281
127	127	8	64	1	4	1.12811
64	127	8	64	1	4	1.12811
256	64	8	64	1	4	1.12818
127	64	8	64	1	4	1.12819
64	64	8	64	1	4	1.1282
127	256	8	64	4	2	1.12859
256	256	8	64	4	2	1.12859
64	256	8	64	4	2	1.12859
127	127	8	64	4	2	1.1286
256	127	8	64	4	2	1.1286
64	127	8	64	4	2	1.1286
127	64	8	64	4	2	1.12861
256	64	8	64	4	2	1.12861
64	64	8	64	4	2	1.12861
256	256	8	64	2	2	1.12861

We can observe that changing the size of L1 Instruction cache has no effect on CPI. Reducing the size of L1 Data Cache slightly increases the CPI, Reducing the associativity of L1 or L2 cache increases the CPI. Similarly, reducing the size of L2 cache increases the CPI.

ii. For benchmark: 429.mcf

L1I Cache Size	L1D Cache Size	L2 Cache Size	Block Size	Assoc. of L1	Assoc. of L2	CPI
64	127	8	64	2	4	1.02155
127	127	1	64	2	2	1.02155
256	256	2	64	2	1	1.02155
256	256	2	64	2	2	1.02155
256	256	2	64	2	4	1.02155
256	256	2	64	4	1	1.02155
256	256	2	64	4	2	1.02155
64	127	8	64	4	4	1.02155
64	127	8	64	4	2	1.02155
64	127	8	64	4	1	1.02155
127	127	1	64	2	4	1.02155
64	127	8	64	2	2	1.02155
64	127	8	64	2	1	1.02155
256	256	2	64	4	4	1.02155
64	127	4	64	4	4	1.02155
64	127	4	64	4	2	1.02155
64	127	4	64	4	1	1.02155
64	127	4	64	2	4	1.02155
64	127	4	64	2	2	1.02155
127	127	4	64	2	1	1.02155
127	127	8	64	2	4	1.02155
127	127	8	64	2	2	1.02155
127	127	8	64	2	1	1.02155
127	127	4	64	4	4	1.02155
127	127	4	64	4	2	1.02155
127	127	4	64	4	1	1.02155
127	127	4	64	2	4	1.02155
127	127	4	64	2	2	1.02155
64	127	4	64	2	1	1.02155
127	127	2	64	4	4	1.02155
127	127	2	64	4	2	1.02155
127	127	2	64	4	1	1.02155
127	127	2	64	2	4	1.02155
127	127	2	64	2	2	1.02155
127	127	2	64	2	1	1.02155
127	127	1	64	4	4	1.02155
127	127	1	64	4	2	1.02155
127	64	8	64	4	4	1.02155
256	64	2	64	4	1	1.02155

The above configuration along with many more gave optimal in this benchmark and similarly like prev benchmark there was a slight increase in CPI when size of L1 Data Cache was reduced along with L2 cache.

iii. For benchmark: 456.hmmer

L1I Cache Size	L1D Cache Size	L2 Cache Size	Block Size	Assoc. of L1	Assoc. of L2	CPI
256	256	8	64	4	4	1.01975
256	256	4	64	4	4	1.01975
256	256	4	64	4	2	1.01975
256	256	4	64	4	1	1.01975
256	256	2	64	4	4	1.01975
256	256	2	64	4	2	1.01975
256	256	2	64	4	1	1.01975
256	256	1	64	4	4	1.01975
256	256	1	64	4	2	1.01975
256	256	8	64	4	1	1.01975
256	256	8	64	4	2	1.01975
127	256	2	64	4	4	1.01975
127	256	2	64	4	2	1.01975
127	256	2	64	4	1	1.01975
127	256	8	64	4	4	1.01975
127	256	8	64	4	2	1.01975
127	256	1	64	4	4	1.01975
127	256	1	64	4	2	1.01975
127	256	8	64	4	1	1.01975
127	256	4	64	4	4	1.01975
127	256	4	64	4	2	1.01975
127	256	4	64	4	1	1.01975
256	256	8	64	2	4	1.01975
256	256	8	64	2	2	1.01975
256	256	1	64	2	2	1.01975
256	256	8	64	2	1	1.01975
256	256	1	64	2	4	1.01975
256	256	4	64	2	4	1.01975
256	256	4	64	2	2	1.01975
256	256	4	64	2	1	1.01975
256	256	2	64	2	1	1.01975
256	256	2	64	2	2	1.01975
256	256	2	64	2	4	1.01975
64	256	4	64	4	4	1.01975
64	256	8	64	4	4	1.01975
64	256	4	64	4	2	1.01975
64	256	4	64	4	1	1.01975
64	256	2	64	4	4	1.01975

The above configs along with some more configs gave optimal CPI for this benchmark. The first slight increase in CPI was at configs

64	256	2	64	2	4	1.01976

iv. For benchmark: 458.sjeng

L1I Cache Size	L1D Cache Size	L2 Cache Size	Block Size	Assoc. of L1	Assoc. of L2	CPI
64	256	2	64	4	2	1.07928
127	256	2	64	4	1	1.07928
256	256	8	64	4	1	1.07928
127	256	1	64	4	2	1.07928
127	256	1	64	4	1	1.07928
64	256	8	64	4	4	1.07928
64	256	8	64	4	2	1.07928
64	256	8	64	4	1	1.07928
64	256	4	64	4	4	1.07928
64	256	4	64	4	2	1.07928
64	256	4	64	4	1	1.07928
64	256	2	64	4	4	1.07928
127	256	1	64	4	4	1.07928
64	256	2	64	4	1	1.07928
64	256	1	64	4	4	1.07928
64	256	1	64	4	2	1.07928
64	256	1	64	4	1	1.07928
256	256	2	64	4	1	1.07928
256	256	2	64	4	2	1.07928
256	256	2	64	4	4	1.07928
256	256	4	64	4	1	1.07928
256	256	4	64	4	2	1.07928
256	256	4	64	4	4	1.07928
256	256	8	64	4	4	1.07928
127	256	8	64	4	4	1.07928
127	256	8	64	4	2	1.07928
127	256	8	64	4	1	1.07928
127	256	2	64	4	2	1.07928
127	256	4	64	4	4	1.07928
127	256	4	64	4	2	1.07928
127	256	4	64	4	1	1.07928
256	256	1	64	4	1	1.07928
256	256	1	64	4	2	1.07928
127	256	2	64	4	4	1.07928
256	256	1	64	4	4	1.07928
256	256	8	64	4	2	1.07928

The above configs along with some more configs gave optimal CPI for this benchmark. The first slight increase in CPI was at configs

256	256	8	64	1	2	1.07935

v. For benchmark: 470.lbm

L1I Cache Size	L1D Cache Size	L2 Cache Size	Block Size	Assoc. of L1	Assoc. of L2	CPI
64	64	8	64	2	1	1.08646
64	127	8	64	2	1	1.08646
64	127	8	64	2	2	1.08646
64	64	4	64	2	1	1.08646
64	256	4	64	2	2	1.08646
64	256	4	64	2	1	1.08646
64	256	4	64	2	4	1.08646
64	64	2	64	2	1	1.08646
64	256	2	64	2	4	1.08646
64	256	2	64	2	1	1.08646
64	127	1	64	2	4	1.08646
64	64	4	64	2	2	1.08646
64	127	1	64	2	2	1.08646
64	127	1	64	2	1	1.08646
256	256	8	64	1	1	1.08653
256	256	1	64	1	2	1.08653
256	256	8	64	1	2	1.08653
256	256	8	64	1	4	1.08653
256	256	4	64	1	4	1.08653
256	256	4	64	1	2	1.08653
256	256	2	64	1	2	1.08653
256	256	2	64	1	4	1.08653
256	256	1	64	1	4	1.08653
127	256	4	64	1	4	1.08653
127	256	2	64	1	4	1.08653
127	256	2	64	1	2	1.08653
127	256	1	64	1	2	1.08653
127	256	1	64	1	4	1.08653
127	256	8	64	1	4	1.08653
127	256	8	64	1	2	1.08653
127	256	8	64	1	1	1.08653
127	256	4	64	1	2	1.08653
64	256	8	64	1	1	1.08653
64	256	8	64	1	4	1.08653
64	256	8	64	1	2	1.08653
64	256	1	64	1	2	1.08653
64	256	1	64	1	4	1.08653
64	256	2	64	1	2	1.08653
64	256	2	64	1	4	1.08653

In this benchmark optimal values are obtained for less value of L1 Instruction Cache Size.

Now, we can visualise the effect of different factors on CPI with help of some of the graphs. These graphs were made using GNUPlot on different values obtained during previous step on benchmark 1: 401.bzip2

VISUALISING TRADEOFF BETWEEN DIFFERENT DESIGN POINTS

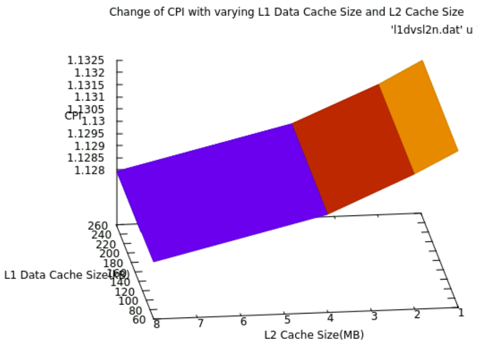

Figure 1: Fig:CPI vs L1 Data Cache Size and L2 Cache Size

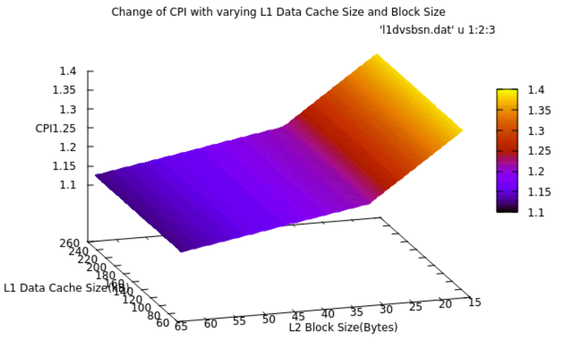

Figure 2: Fig:CPI vs L1 Data Cache Size and Block Size

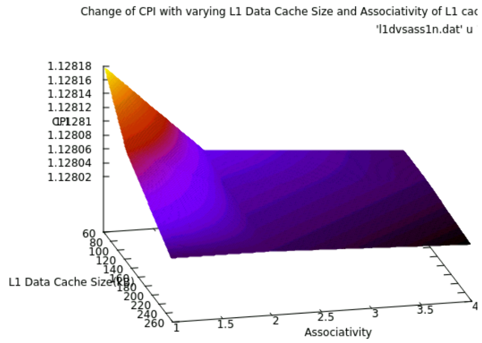

Figure 3: Fig:CPI vs L1 Data Cache Size and Associativity of L1

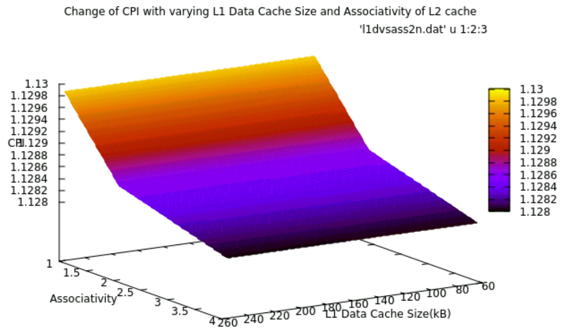

Figure 4: Fig:CPI vs L1 Data Cache Size and Associativity of L2

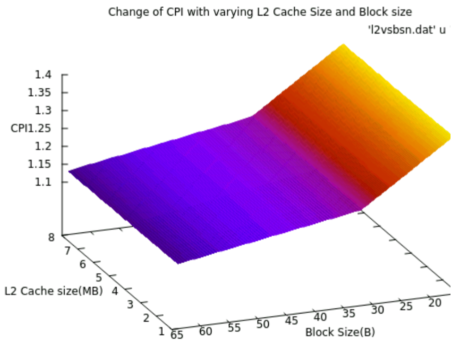

Figure 5: Fig:CPI vs L2 Cache Size and Block Size

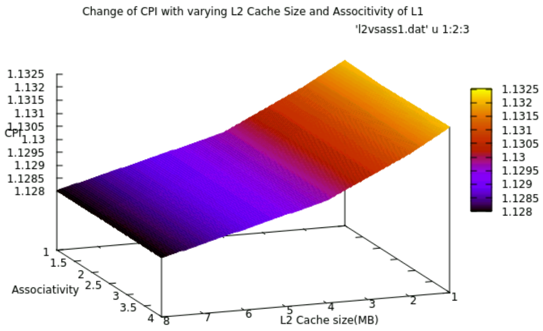

Figure 6: Fig:CPI vs L2 Cache Size and Associativity of L1

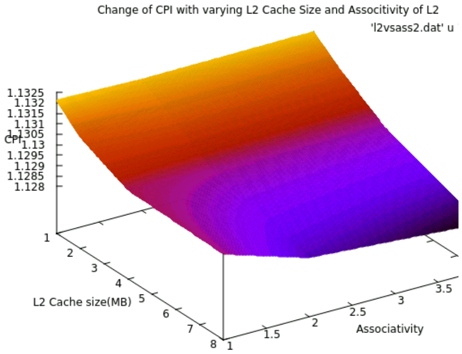

Figure 7: Fig:CPI vs L2 Data Cache and Associativity of L2

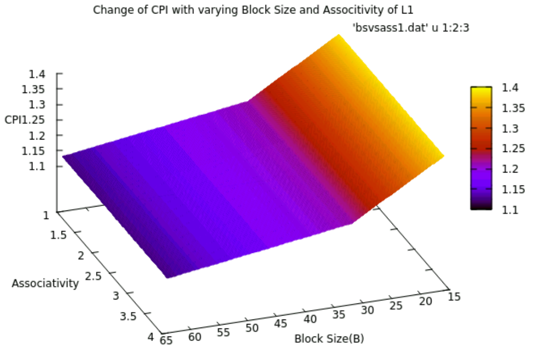

Figure 8: Fig:CPI vs Block Size and associativity of L1

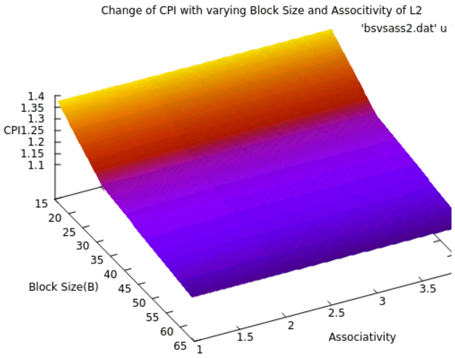

Figure 9: Fig:CPI vs Block Size and associativity of L2

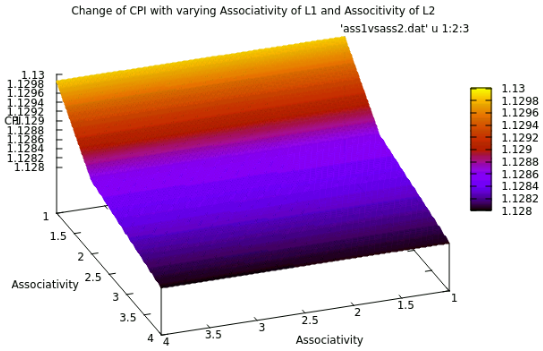

Figure 10: Fig:CPI vs Associativity of L1 and Associativity of L2