Rusu Wu
Programming assignment #1
11694764
10/13/2020

Programming assignment #1

1. Find CPI from equation

Parameter setting:

L1 Data cache and L1 Instruction cache: 2-way set-associative, 64-byte block, 1024 sets

(2*64*1024=128KB).

L2 cache: 1MB cache directed mapped, 64-byte block.

Instruction count: 100000000

$$CPI = 1 + \frac{5*(IL1_{misses} + DL1_{misses}) + 50*(L2_{misses})}{Total\_instrctions}$$

For the benchmark *401.bzip2,*

system.cpu.icache.overall_misses::total 521

system.cpu.dcache.overall_misses::total 2434558

system.l2.overall_misses::total 2219608

CPI=2.2316

For the benchmark *429.mcf,*

system.cpu.icache.overall_misses::total 486

system.cpu.dcache.overall_misses::total 97288 system.l2.overall misses::total 92093

CPI=1.0509

For the benchmark *456.hmmer,*

system.cpu.icache.overall_misses::total 1340

system.cpu.dcache.overall_misses::total 40149

system.l2.overall_misses::total 6354

CPI=1.0053

For the benchmark *458.sjeng,*

system.cpu.icache.overall_misses::total 2183

system.cpu.dcache.overall_misses::total 8298580

system.l2.overall misses::total 8297070

CPI=5.5636

For the benchmark *470.lbm,*

system.cpu.icache.overall_misses::total 481

system.cpu.dcache.overall_misses::total 1533680

system.l2.overall misses::total 1534158

CPI=1.8438

2. Find optimal configuration

1) Associativity

Parameter setting:

L1 Data cache and L1 Instruction cache, 64-byte block, 1024 sets (2*64*1024=128KB).

L2 cache: 1MB , 64-byte block.

Instruction count: 100000000

	Associativity					
L1 data cache	1	1	2	2	4	8
L1 instruction cache	1	1	2	2	4	8
L2 cache	1	2	1	2	4	8

For (2:2:1),we could find the results at step 2.

For the benchmark *401.bzip2,*

Associativity L1D: L1I: L2	1:1:1	1:1:2	2:2:1	2:2:2	4:4:4	8:8:8
L1I misses	552	552	521	521	521	521
L1D misses	2498810	2498810	2434558	2434558	2409637	2383224
L2 misses	2217963	2209931	2219608	2209931	2201488	2198791
CPI	2.2339	2.2299	2.2316	2.2267	2.2213	2.2186

As shown in the graph, for L1I, the overall misses do decrease when change the associativity from 1 to 2. However, as L1I associativity increasing , the overall misses keep staying at 521. Therefore, the optimal associativity of L1I is 2. For L1D and L2, the more assiciativity, the less misses.

For the benchmark *429.mcf,*

Associativity L1D: L1I: L2	1:1:1	1:1:2	2:2:1	2:2:2	4:4:4	8:8:8
L1I overall misses	489	489	486	486	486	486
L1D overall misses	109044	109044	97288	97288	95692	95211
L2 overall misses	92621	90647	92093	90647	90180	89666
CPI	1.0518	1.0508	1.0509	1.0502	1.0499	1.0496

The optional-associativity is 2-way for L1I, 8-way for both L1D and L2.

For the benchmark *456.hmmer,*

Associativity L1D: L1I: L2	1:1:1	1:1:2	2:2:1	2:2:2	4:4:4	8:8:8
L1I overall misses	3643	3643	1340	1340	1315	1314
L1D overall misses	67164	67164	40149	40149	33643	33664
L2 overall misses	6332	6329	6354	6329	6329	6329

CPI	1.0067	1.0067	1.0053	1.0052	1.0049	1.0049

Similar to *401.bzip2*, b, the optimal configuration is 4-way for both L1I, L1D and L2.

For the benchmark *458.sjeng,*

Associativity L1D: L1I: L2	1:1:1	1:1:2	2:2:1	2:2:2	4:4:4	8:8:8
L1I overall misses	3342	3342	2183	2183	1659	1561
L1D overall misses	8315585	8315585	8298580	8298580	8297182	8296807
L2 overall misses	8298089	8296933	8297070	8296933	8296776	8296765
CPI	5.5650	5.5644	5.5636	5.5635	5.5633	5.5633

8-way accociativity for both L1 and L2 has the best CPI.

For the benchmark *470.lbm,*

Associativity L1D: L1I: L2	1:1:1	1:1:2	2:2:1	2:2:2	4:4:4	8:8:8
L1I overall misses	496	496	481	481	481	481
L1D overall misses	1544899	1544899	1535680	1535680	1533678	1533678
L2 overall	1534158	1534158	1534158	1534158	1534158	1534158

misses						
CPI	1.8443	1.8443	1.8438	1.8438	1.8438	1.8438

As Associativity grows more than two ways, L1I overall misses stay at 481, the two-way L1I for is enough. L1D overall misses remain the same as associativity increases from 4 to 8, so 4-way is optimal. L1D overall misses always remain at 1544899, so direct-mapped is the optimal one. (211) is the optimal associativity.

2) Block size

Parameter setting:

L1 Data cache and L1 Instruction cache: 2-way set-associative, 128KB

L2 cache: 1MB cache directed mapped

Instruction count: 100000000

Block size	16 byte	32byte	64byte	128byte

For the benchmark *401.bzip2,*

Block size	16 byte	32byte	64byte	128byte
L1I overall misses	1514	876	521	318
L1D overall misses	8179123	4339160	2434558	1537543
L2 overall misses	8006884	4147516	2219608	1260772
CPI	5.4125	3.2908	2.2316	1.7073

For the benchmark *429.mcf,*

Block size	16 byte	32byte	64byte	128byte
L1I overall misses	1411	806	486	311
L1D overall misses	307422	167083	97288	56771
L2 overall misses	302063	162089	92093	50615
CPI	1.1665	1.0894	1.0509	1.0282

For the benchmark *456.hmmer,*

Block size	16 byte	32byte	64byte	128byte
L1I overall misses	4109	2263	1340	799
L1D overall misses	149292	76773	40149	21900
L2 overall misses	23124	12033	6354	3363
CPI	1.0192	1.0100	1.0053	1.0028

For the benchmark *458.sjeng,*

Block size	16 byte	32byte	64byte	128byte
L1I overall misses	5868	3313	2183	1448
L1D overall misses	33166412	16587483	8298580	4155922
L2 overall misses	33169482	16587830	8297070	4151767
CPI	19.2434	10.1235	5.5636	3.2838

For the benchmark *470.lbm,*

Block size	16 byte	32byte	64byte	128byte

L1I overall misses	1312	767	481	309
L1D overall misses	6134474	3067285	1535680	766870
L2 overall misses	6135785	3068051	1534158	767176
CPI	4.3747	2.6874	1.8438	1.4219

For all benchmarks, with more byte of block, the CPI would reach lower, so when it increases to 128 byte of block size, we would have a much lower CPI.

3) Size allocation for L1 instruction cache and L1 data cache

L1 Data cache and L1 Instruction cache: 2-way set-associative, 64-byte block.

L2 cache: 1MB cache directed mapped, 64-byte block.

Instruction count: 100000000

L1i:L1d	64kb:64kb	128kb:128kb	128kb:256kb	256kb:128kb	256kb:256kb

For the benchmark *401.bzip2,*

L1i:L2d	64kb:64kb	128kb:128kb	128kb:256kb	256kb:128kb	256kb:256kb
L1I misses	530	521	521	521	521
L1D misses	2583321	2434558	2310063	2434558	2310063
L2 misses	2223426	2219608	2209769	2219608	2209769
CPI	2.2409	2.2316	2.2204	2.2316	2.2204

When L1i:L2d = 128kb:256kb and L1i:L2d = 256kb:256kb, it has a lowest CPI. Therefore, choose L1i:L2d = 128kb:256kb as the optimal configuration.

For the benchmark *429.mcf,*

L1I:L1D	64kb:64kb	128kb:128kb	128kb:256kb	256kb:128kb	256kb:256kb

L1I misses	486	486	486	486	486
L1D misses	104148	97288	94690	97288	94690
L2 misses	92432	92093	91795	92093	91795
CPI	1.0514	1.0509	1.0507	1.0509	1.0507

When L1i:L2d = 128kb:256kb and L1i:L2d = 256kb:256kb, it has a lowest CPI. Therefore, choose L1i:L2d = 128kb:256kb as the optimal configuration.

For the benchmark *456.hmmer,*

L1I:L2D	64kb:64kb	128kb:128kb	128kb:256kb	256kb:128kb	256kb:256kb
L1I misses	2028	1340	1319	1315	1315
L1D misses	89601	40149	6052	40149	6052
L2 misses	6368	6354	6329	6333	6329
CPI	1.0078	1.0053	1.0035	1.0052	1.0035

When L1i:L2d = 128kb:256kb and L1i:L2d = 256kb:256kb, it has a lowest CPI. Therefore, choose L1i:L2d = 128kb:256kb as the optimal configuration.

For the benchmark *458.sjeng,*

L1I:L1D	64kb:64kb	128kb:128kb	128kb:256kb	256kb:128kb	256kb:256kb
L1I misses	38546	2183	2183	1561	1561
L1D misses	8308362	8298580	8296195	8298580	8296195
L2 misses	8297654	8297070	8296906	8297059	8296897
CPI	5.5662	5.5636	5.5634	5.5635	5.5633

According to CPI,we could choose L1i:L2d = 256kb:256kb as the optimal configuration.

For the benchmark *470.lbm,*

L1I:L1D	64kb:64kb	128kb:128kb	128kb:256kb	256kb:128kb	256kb:256kb
L1I misses	490	481	481	481	481
L1D misses	1533680	1535680	1533678	1533678	1533678
L2 misses	1534158	1534158	1534158	1534158	1534158
CPI	1.8438	1.8438	1.8438	1.8438	1.8438

According to misses, we could choose L1i:L2d = 128kb:256kb as the optimal configuration.

4) L2 cache

Parameter setting:

L1 Data cache and L1 Instruction cache: 2-way set-associative, 64-byte block, 1024 sets (2*64*1024=128KB).

L2 cache: cache directed mapped, 64-byte block.

Instruction count: 100000000

L2	512 kb	1Mb	2Mb	4Mb

For the benchmark *401.bzip2,*

L2	512 kb	1Mb	2Mb	4Mb
L1I misses	521	521	521	521
L1D misses	2434558	2434558	2434558	2434558
L2 misses	2251531	2219608	2202318	2178402
CPI	2.2475	2.2316	2.2229	2.2110

For the benchmark *429.mcf,*

L2	512 kb	1Mb	2Mb	4Mb
L1I misses	486	486	486	486
L1D misses	97288	97288	97288	97288
L2 misses	94745	92093	88708	83119
CPI	1.0523	1.0509	1.0492	1.0464

For the benchmark *456.hmmer,*

L2	512 kb	1Mb	2Mb	4Mb
L1I misses.	1340	1340	1340	1340
L1D misses	40149	40149	40149	40149
L2 misses	6368	6354	6329	6329
CPI	1.0053	1.0053	1.0052	1.0052

For the benchmark *458.sjeng,*

L2	512 kb	1Mb	2Mb	4Mb
L1I misses	2183	2183	2183	2183
L1D misses	8298580	8298580	8298580	8298580
L2 misses	8297279	8297070	8296940	8296615
CPI	5.5637	5.5636	5.5635	5.5633

For the benchmark *470.lbm,*

L2	512 kb	1Mb	2Mb	4Mb

L1I misses	481	481	481	481
L1D misses	1535680	1535680	1533680	1533680
L2 misses	1534158	1534158	1534158	1534158
CPI	1.8438	1.8438	1.8438	1.8438

Commonly, changing L2 size only can affect the L2 total misses. The L2 misses of *401.bzip2, 429.mcf and 456.hmmer would decrease as* L2 size increase. Therefore, we could could choose 4MB as the optimal size for *401.bzip2, 429.mcf , 456.hmmer and458.sjeng.* However, 470.lbm would not be affected by L2 size according to analysis, so we could choose the smallest size 4MB as the optimal size.

Therefore, the lowest CPI can be achieved with following configurations.

	Associativity L1d:L1i:L3	Block size	Size allocation L1I:L1d	L2 cache
401.bzip2	2-way:8-way:8-way	128byte	128kb:256kb	4MB
429.mcf	2-way:8-way:8-way	128byte	128kb:256kb	4MB
456.hmmer	4-way:4-way:4-way	128byte	128kb:256kb	4MB
458.sjeng,	8-way:8-way:8-way	128byte	256kb:256kb	4MB
470.lbm	2-way:direct mapped:direct mapped	128byte	128kb:256kb	4MB

With all these optimal-configuration, the lowest CPI of each benchmark is shown as follow:

	L1I misses	L1D misses	L2 misses	CPI
401.bzip2	318	1388391	1216865	1.6779
429.mcf	311	53941	41900	1.0237
456.hmmer	795	3008	3360	1.0019
458.sjeng,	1561	8295557	8296561	5.5631
470.lbm	320	766868	767176	1.4219

3. Define cost function

The Cost Function could be defined as fellow;

$$Cost = L1\mathrm{I}_{\mathrm{asso}} + L1D_{\mathrm{asco}} + L2_{\mathrm{asso}} + 100 * \mathrm{L1}_{\mathrm{size}} + 5 * \mathrm{L2}_{\mathrm{size}} + 2 * \frac{B\mathrm{lock\ size}}{16}$$

$L1I_{\mathrm{asso}}$, $L1D_{\mathrm{asso}}$, and $L2_{\mathrm{asso}}$ represent the associativity of L1 instruction cache.

L1data cache and L2 cache.

$L1_{\mathrm{size}}$ and $L2_{\mathrm{size}}$ represent the total size of L1 and L2.

The unit of L1, L2 is MB.

Also The unit of Block size is kB.

4. Evaluation function

Defined an evaluation function as fellow:

$$\text{Evaluation Score} = \text{Cost} * \text{CPI}$$

Evaluation Score(Score) is a factor I defined for evaluating the configurations.

Since we need to lower CPI and Cost, in this evaluation function, the less Score means that the configuration has better cost performance.

However, based on the results we get from step3, we known that lowering the cost probably causes higher CPI.

Usually, the increasing of Cost would bigger than the decreasing of CPI. Therefore, I set the gap threshold as 10. It means that as the Cost increases and the CPI decreases, if the new score is no more than 10 points higher than the old score, I would say the new configuration is more optimal than the old one.

For example, as configuration upgrades and its cost increases, we know that the CPI is tern to be lower. If the old configuration's Score is 90 and the Score of a new configuration is 99, then I would say the new one is more optimal than the old one.

If the Score has a decreasing trend, it means that CPI decreases faster than the increasing of Cost. Therefore, the lower score the better as Cost grows(eg. Block size at 401.bzip2). Also, if CPI does not change at all, the lower cost one would be the optimal one(eg. 470.lbm).

The best configuration for each bench mark is highlighted for each benchmark as fellow.

1) For 401.bzip2

401.bzip2	L1I Asso	L1D Asso	L2 Asso	L1I Size (kB)	L1D Size (kB)	L2 Size (MB	Block Size (Byte)	Cost	CPI	Score
Associativity	1	1	1	128	128	1	64	41	2.2339	91.5899
	1	1	2	128	128	1	64	42	2.2299	93.6558
	2	2	1	128	128	1	64	43	2.2316	95.9588
	2	2	2	128	128	1	64	44	2.2267	97.9748
	4	4	4	128	128	1	64	54	2.2213	119.9502
	8	8	8	128	128	1	64	62	2.2186	137.5532
L1I and L1D Allocation	2	2	1	64	64	1	64	30.5	2.2409	68.34745
	2	2	1	128	128	1	64	43	2.2316	95.9588
	2	2	1	256	128	1	64	55.5	2.2316	123.8538
	2	2	1	128	256	1	64	55.5	2.2204	123.2322
	2	2	1	256	256	1	64	68	2.2204	150.9872
L2 Size	2	2	1	128	128	0.5	64	40.5	2.2475	91.02375
	2	2	1	128	128	1	64	43	2.2316	95.9588
	2	2	1	128	128	2	64	48	2.2229	106.6992
	2	2	1	128	128	4	64	53	2.2110	117.183
Block Size	2	2	1	128	128	1	16	37	5.4125	200.2625
	2	2	1	128	128	1	32	39	3.2908	128.3412
	2	2	1	128	128	1	64	43	2.2316	95.9588
	2	2	1	128	128	1	128	51	1.7073	87.0723

2)For the benchmark *429.mcf,*

429.mcf,	L1I Asso	L1D Asso	L2 Asso	L1I Size (kB)	L1D Size (kB)	L2 Size (MB	Block Size (Byte)	Cost	CPI	Score
Associativity	1	1	1	128	128	1	64	41	1.0518	43.1238
	1	1	2	128	128	1	64	42	1.0508	44.1336
	2	2	1	128	128	1	64	43	1.0509	45.1887
	2	2	2	128	128	1	64	44	1.0502	46.2088
	4	4	4	128	128	1	64	54	1.0499	56.6946
	8	8	8	128	128	1	64	62	1.0496	65.0752
L1I and L1D Allocation	2	2	1	64	64	1	64	30.5	1.0514	32.0677
	2	2	1	128	128	1	64	43	1.0509	45.1887
	2	2	1	256	128	1	64	55.5	1.0509	58.32495
	2	2	1	128	256	1	64	55.5	1.0507	58.31385
	2	2	1	256	256	1	64	68	1.0507	71.4476
L2 Size	2	2	1	128	128	0.5	64	40.5	1.0523	42.61815
	2	2	1	128	128	1	64	43	1.0509	45.1887
	2	2	1	128	128	2	64	48	1.0492	50.3616
	2	2	1	128	128	4	64	53	1.0464	55.4592
Block Size	2	2	1	128	128	1	16	37	1.1665	43.1605
	2	2	1	128	128	1	32	39	1.0894	42.4866
	2	2	1	128	128	1	64	43	1.0509	45.1887
	2	2	1	128	128	1	128	51	1.0282	52.4382

For the benchmark *456.hmmer,*

456.hmmer,	L1I Asso	L1D Asso	L2 Asso	L1I Size (kB)	L1D Size (kB)	L2 Size (MB	Block Size (Byte)	Cost	CPI	Score
Associativity	1	1	1	128	128	1	64	41	1.0067	41.2747
	1	1	2	128	128	1	64	42	1.0067	42.2814
	2	2	1	128	128	1	64	43	1.0053	43.2279
	2	2	2	128	128	1	64	44	1.0052	44.2288
	4	4	4	128	128	1	64	54	1.0049	54.2646
	8	8	8	128	128	1	64	62	1.0049	62.3038
L1I and L1D Allocation	2	2	1	64	64	1	64	30.5	1.0078	30.7379
	2	2	1	128	128	1	64	43	1.0053	43.2279
	2	2	1	256	128	1	64	55.5	1.0052	55.7886
	2	2	1	128	256	1	64	55.5	1.0035	55.69425
	2	2	1	256	256	1	64	68	1.0035	68.238
L2 Size	2	2	1	128	128	0.5	64	40.5	1.0053	40.71465
	2	2	1	128	128	1	64	43	1.0053	43.2279
	2	2	1	128	128	2	64	48	1.0052	48.2496
	2	2	1	128	128	4	64	53	1.0052	53.2756
Block Size	2	2	1	128	128	1	16	37	1.0192	37.7104
	2	2	1	128	128	1	32	39	1.01	39.39
	2	2	1	128	128	1	64	43	1.0053	43.2279
	2	2	1	128	128	1	128	51	1.0028	51.1428

For the benchmark *458.sjeng,*

458.sjeng,	L1I Asso	L1D Asso	L2 Asso	L1I Size(kB)	L1D Size (kB)	L2 Size (MB	Block Size (Byte)	Cost	CPI	Score
Associativity	1	1	1	128	128	1	64	41	5.565	228.165
	1	1	2	128	128	1	64	42	5.5644	233.7048
	2	2	1	128	128	1	64	43	5.5636	239.2348
	2	2	2	128	128	1	64	44	5.5635	244.7984
	4	4	4	128	128	1	64	54	5.5633	300.4182
	8	8	8	128	128	1	64	62	5.5633	344.9246
L1I and L1D Allocation	2	2	1	64	64	1	64	30.5	5.5662	169.7691
	2	2	1	128	128	1	64	43	5.5636	239.2348
	2	2	1	256	128	1	64	55.5	5.5635	308.77425
	2	2	1	128	256	1	64	55.5	5.5634	308.7687
	2	2	1	256	256	1	64	68	5.5633	378.3044
L2 Size	2	2	1	128	128	0.5	64	40.5	5.5637	225.32985
	2	2	1	128	128	1	64	43	5.5636	239.2348
	2	2	1	128	128	2	64	48	5.5635	267.048
	2	2	1	128	128	4	64	53	5.5633	294.8549
Block Size	2	2	1	128	128	1	16	37	19.2434	712.0058
	2	2	1	128	128	1	32	39	10.1235	394.8165
	2	2	1	128	128	1	64	43	5.5636	239.2348
	2	2	1	128	128	1	128	51	3.2838	167.4738

For the benchmark *470.lbm,*

470.lbm,	L1I Asso	L1D Asso	L2 Asso	L1I Size (kB)	L1D Size (kB)	L2 Size (MB	Block Size (Byte)	Cost	CPI	Score
Associativity	1	1	1	128	128	1	64	41	1.8438	75.5958
	1	1	2	128	128	1	64	42	1.8438	77.4396
	2	2	1	128	128	1	64	43	1.8438	79.2834
	2	2	2	128	128	1	64	44	1.8438	81.1272
	4	4	4	128	128	1	64	54	1.8438	99.5652
	8	8	8	128	128	1	64	62	1.8438	114.3156
L1I and L1D Allocation	2	2	1	64	64	1	64	30.5	1.8438	56.2359
	2	2	1	128	128	1	64	43	1.8438	79.2834
	2	2	1	256	128	1	64	55.5	1.8438	102.3309
	2	2	1	128	256	1	64	55.5	1.8438	102.3309
	2	2	1	256	256	1	64	68	1.8438	125.3784
L2 Size	2	2	1	128	128	0.5	64	40.5	1.8438	74.6739
	2	2	1	128	128	1	64	43	1.8438	79.2834
	2	2	1	128	128	2	64	48	1.8438	88.5024
	2	2	1	128	128	4	64	53	1.8438	97.7214
Block Size	2	2	1	128	128	1	16	37	4.3747	161.8639
	2	2	1	128	128	1	32	39	2.6874	104.8086
	2	2	1	128	128	1	64	43	1.8438	79.2834
	2	2	1	128	128	1	128	51	1.4219	72.5169

Therefore, after the analysis, the optimal configuration of these benchmark are :

	L1I Asso	L1D Asso	L2 Asso	L1I Size(kB)	L1D Size (kB)	L2 Size (MB)	Block Size (Byte)
401.bzip2	2	2	2	64	64	1	128
429.mcf,	2	2	2	128	128	4	128
456.hmmer	2	2	2	64	64	4	128
458.sjeng,	2	2	2	64	64	0.5	128
470.lbm,	1	1	1	64	64	0.5	128