**CSE 590 COMPUTER ARCHITECTURE PROJECT 2**

Chitravardhini Bellamkonda |50420023 | Chitrava

* **Problem Statement**

Cache parameters influence the performance of a processor to a great extent. In this project you are asked to tune various cache parameters and analyze performance in each case.The architecture you will have to analyze is the X86 architecture.

* **Introduction and Tools**

In this project we would be simulating various cache situations with different sizes to find out the impact of each case scenario on the overall system performance (memory fetching and CPI)

For this we will be using Gem5 Simulator over an x86 architecture. This particular simulator is setup on the metallica server, and we will be using the same for it.

There are two modes of simulating Gem5: SE (System Call Emulation) mode and FS (Full System) mode. For this project, we will use the SE mode. For this we are provided with system script at path: $GEM5\_DIR/configs/example/se.py on the server.

The following benchmarks are provided by the instructor to perform the experiments and binary files are provided in respective folders for execution:

1. 401.bzip2

2. 429.mcf

3. 456.hmmer

4. 458.sjeng

5. 470.lbm

Final execution Steps:

Step 1: set up env vars using the commands in the problem statement

Step 2: execute the below command with proper arguments

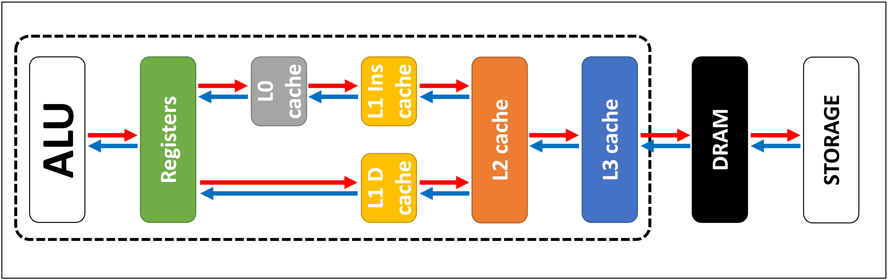
**time /util/gem5/build/X86/gem5.opt -d /home/csgrad/chitrava/COA/401/l1d\_assoc /util/gem5/configs/example/se.py -c /util/gem5/benchmark/401.bzip2/src/benchmark -o /util/gem5/benchmark/401.bzip2/data/input.program -I 1000000 --caches --l2cache --l1d\_assoc=4 --l1d\_size=64kB --l2\_size=1MB --l1i\_size=64kB --l1i\_assoc=8 --l2\_assoc=8**

We have used an automated python script to derive all the stats for the experiments.

Before we dive into the details of the experiment let us have a look at little terminology.

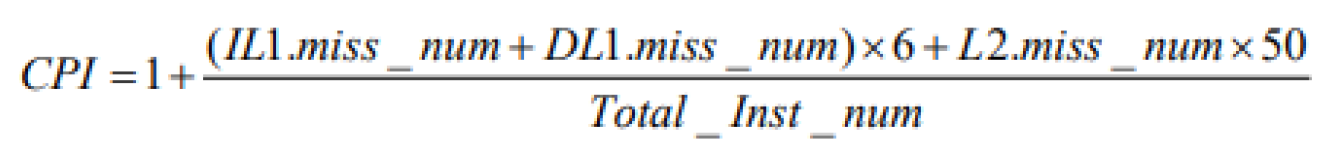
* **Terminology**

1. Cache: Cache is high-speed memory unlike the main memory within the processor. It lies within the processor die and is very close to the processor indeed reducing the latency and faster data fetching speeds. This memory is costlier than main memory as it lies within the dye and when compared to CPU registers it is more economical. This acts as a buffer between main memory (RAM) and the processor. This holds the frequently used data by the CPU and there are many types as can be seen from the picture mentioned below.

[](https://www.google.com/search?q=cache+memory+l1+l2&rlz=1C5CHFA_enUS984US984&hl=en&sxsrf=ALiCzsZ-HzV5Nd5eLE0rSVHmzKt_x9nOBA:1651626562359&source=lnms&tbm=isch&sa=X&ved=2ahUKEwi-5LO01MT3AhUig4kEHdA7DgYQ_AUoAXoECAEQAw&biw=1440&bih=686&dpr=2#imgrc=x1h1wt3Ev97RgM)

1. L1 Cache: Also called as primary cache, this resides within the processor and is the first location from where the processor tries to fetch the data.
2. L2 Cache: Also called as secondary cache lies within the dye of the chip but away from the processor and is common for all the computing units across the dye.
3. Cache Association: Unlike direct mapped cache, associative caches are mapped based on a particular ID, TAG and are used for avoiding frequent fetching of colliding information from the caches. These are organized based on the number of sets which can be anywhere from 2 to making it fully associative.
4. CPI: In computer architecture, cycles per instruction (aka clock cycles per instruction, clocks per instruction, or CPI) is one aspect of a processor's performance: the average number of clock cycles per instruction for a program or program fragment. It is the multiplicative inverse of instructions per cycle.[[link to reference](file:////Users/yjagilanka/Desktop/In%20computer%20architecture,%20cycles%20per%20instruction%20(aka%20clock%20cycles%20per%20instruction,%20clocks%20per%20instruction,%20or%20CPI)%20is%20one%20aspect%20of%20a%20processor's%20performance:%20the%20average%20number%20of%20clock%20cycles%20per%20instruction%20for%20a%20program%20or%20program%20fragment.%20It%20is%20the%20multiplicative%20inverse%20of%20instructions%20per%20cycle.)]

Its calculation formula is mentioned below:



1. Hit Rate and Miss Rate: Hit Rate is the no of times in which the processor performs a hit to find a specific information from the cache and finds it. Vice versa is the miss rate where processor doesn’t find the information required and misses it.

* **Brief Description/Process:**

In this particular experiment we will be varying multiple cache parameters and their sizes. The parameters being changed are as mentioned below and their ranges are also attached.

1. L1D Cache Parameter: Range 8kB ---- 128kB. Default: 64kB
2. L1D Assoc Parameter: Range 1 ---- 16 Default: 8
3. L2 Cache Parameter: Range 128kB ---- 2MB. Default: 1MB
4. L2 Assoc Cache Parameter: Range 1 ---- 16 Default: 8
5. L1I Cache Parameter: Range 8kB ---- 128kB. Default: 64kB
6. L1I Assoc Cache Parameter: Range 1 ---- 16 Default: 8

With this scenario for every benchmark, we will be having 5 test cases for each parameter and in each benchmark, we will be having 30 such experiments. Overall leading to 150 case scenarios.

* **Results Achieved in Pictorial Format:**

1. **401.bzip2**

A picture containing graphical user interface

Description automatically generated

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart, box and whisker chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

1. **429.mcf**

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generatedChart, line chart

Description automatically generatedChart, line chart

Description automatically generated

1. **456.hmmer**

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, box and whisker chart

Description automatically generatedChart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, line chart

Description automatically generated

1. **458.sjeng**

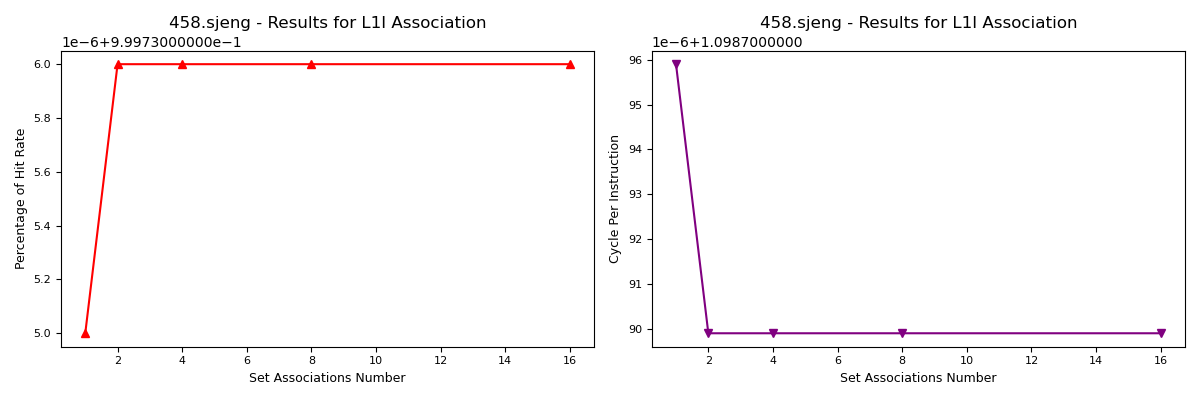
Chart

Description automatically generatedChart, line chart

Description automatically generatedChart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generatedChart, line chart

Description automatically generated

1. **470.lbm**

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generated

Chart, line chart

Description automatically generated

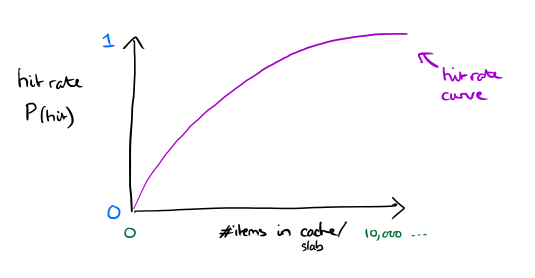
Chart, line chart

Description automatically generated

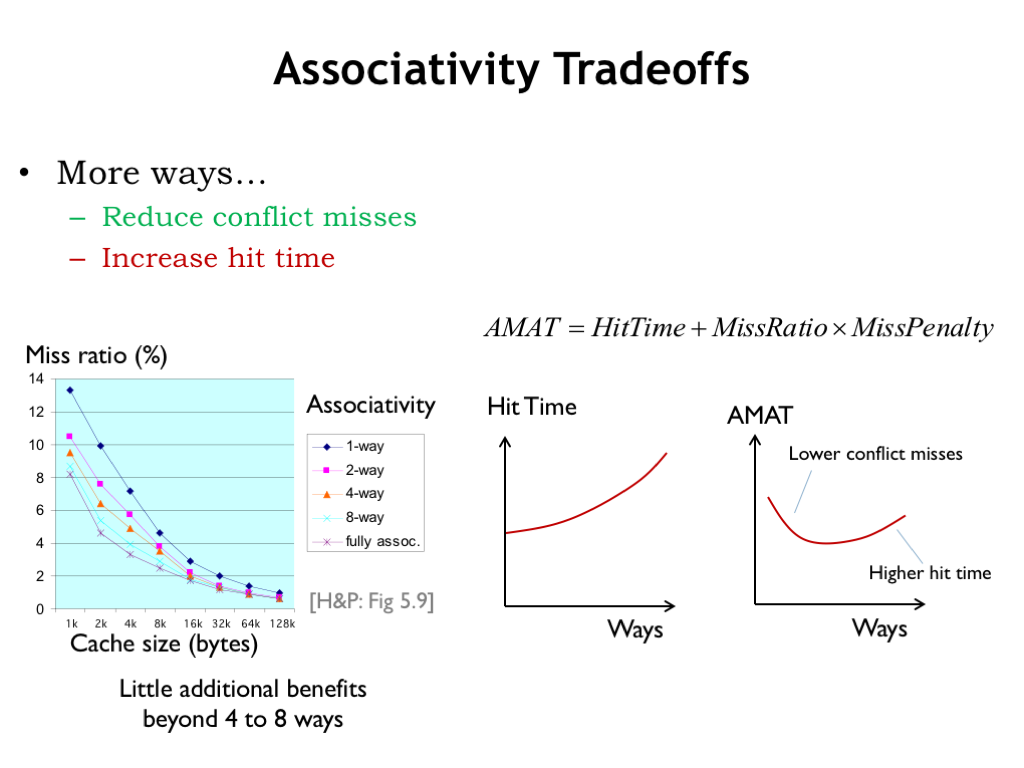
* **Inferences**

Before we go into the actual results the theoretical understanding of the impact on the performance with change in the following types of caches.

**Cache Data Size Increase:** Actual expected impact on increase of data cache size. It is expected that the hit rate percentage increases with the increase in the cache data size.

[](https://www.google.com/search?q=cache+size+vs+hit+rate&tbm=isch&ved=2ahUKEwj4pYvn5cT3AhXAn3IEHXAXAWcQ2-cCegQIABAA&oq=cache+size+vs+hit+rate&gs_lcp=CgNpbWcQAzIECAAQGFDSCVigEmDVE2gAcAB4AIABP4gBiAWSAQIxMZgBAKABAaoBC2d3cy13aXotaW1nwAEB&sclient=img&ei=gORxYvjlAsC_ytMP8K6EuAY&bih=686&biw=1440&rlz=1C5CHFA_enUS984US984#:~:text=is%20being%20loaded-,The%20Morning%20Paper,Scaling%20Performance%20Cliffs%20in%20Web%20Memory%20Caches%20%7C%20the%20morning%20paper,-Visit)

**Cache Associativity Increase:**

[](https://www.google.com/url?sa=i&url=https%3A%2F%2Fcomputationstructures.org%2Flectures%2Fcaches%2Fcaches.html&psig=AOvVaw3gVSA2_uAxgEsjxz1z-6O4&ust=1651717824922000&source=images&cd=vfe&ved=0CA0QjhxqFwoTCNCrvtbmxPcCFQAAAAAdAAAAABAE)

**L1 – Data Cache**

Increase in memory size: As can be inferred from the plots, with increase in L1D cache memory we can see significant increase in the hit rate and reduction in the CPI which informs that the performance of the CPU has increased overall. But to be noted is that after a certain point it tends to be constant. Assumption based on the trend.

|  |  |  |  |
| --- | --- | --- | --- |
| **L1D Cache Size** | **Impact on Hit Rate** | **Impact on Overall** | **Reason** |
| Small | Very Low | Poor Performance higher miss rate | since the data size is small, can’t rely on putting more data and fetching fast |
| Medium | Consistent | Decent Performance | Perfect spot for all levels |
| Large | Over the head | Over Head, lower miss rate | Higher Makings costs and power consumption |

Increase in Associativity: Although associativity follows the same rule but because of the spatial locality the cpi increases in turn making it slow for data fetching and more power consumption. Also, after a certain number the associativity doesn’t have larger positive impact on the CPI and hit rate.

**L1 – Instruction Cache:** Instruction Cache follows the same results as the data cache and graphs as well. But the CPI significantly reduces with increase in the data cache but lags in case of Instruction cache. This is because of more memory fetching operations being performed and hence earlier reduction in CPI. Graphs attached for reference.

Chart

Description automatically generated

Chart, line chart

Description automatically generated

**L2 – Cache**

L2 cache, This has significantly very less hits as can be seen from the data attached. But the miss rates are pretty huge because of the miss rates of higher caches or hard misses being added to the L2.

Table

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **L2 Cache Size/Associativity** | **Impact on Hit Rate** | **Impact on Overall** | **Reason** |
| Medium | Consistent | Decent Performance | Perfect spot for all levels |
| Large | High | Higher CPI, because of larger size | Higher Makings costs and power consumption, with no positive impact on performance |

**Table for Total Hit and Miss Rate Summary:** An excel object has been attached for the same which is averaged over each benchmark type.

.

Table

Description automatically generated

**Note:** Because of the scale of the axes, the impact on CPI and Hit rate isn’t much evident for L2 and also minor changes are not noticeable but are clearly seen in the data.

References:   
  
1. All lecture slides

2. Other References are linked with hyperlinks and images.