# GEM5 Simulator Research

Website Link:

<https://www.gem5.org/>

GEM 5 101

<https://www.gem5.org/documentation/learning_gem5/gem5_101/>

GEM5 Overview Video

<https://www.youtube.com/watch?v=hMj-GB8BTd4>

# GEM5 Simulator - Cache Optimisation

<https://www.youtube.com/watch?v=-riAWSuNwvU>

How to compile .c file in gem5

<http://lacasa.uah.edu/portal/Upload/tutorials/gem5/RunningPrograms-gem5.txt>

How to create Alpha binaries from SPEC2006 in gem5

<https://markgottscho.wordpress.com/2014/09/20/tutorial-easily-running-spec-cpu2006-benchmarks-in-the-gem5-simulator/>

Gem5 links for SPEC2006 binaries

<http://www.m5sim.org/SPEC_CPU2006_benchmarks#Build_the_SPEC_CPU2006_alpha_binaries>

----------------------

20200406

Gem5 Overview video

<https://www.youtube.com/watch?v=wo4b9FPEiHk>

Binaries for Spec2006

<http://www.m5sim.org/SPEC_CPU2006_benchmarks#Input_sets_and_Binaries>

Comand

./build/X86/gem5.fast configs/example/se.py --num-cpus=1 --ruby --l1d\_size=32kB --l1d\_size=32kB --l2\_size=512kB --maxinsts=100 -c tests/matmul.x86

-----------------------------

Spec Bench marks

<https://personal.utdallas.edu/~gxm112130/EE6304FA17/project1.pdf>

<https://github.com/timberjack/Project1_SPEC>

-----------------------------------------------------------------

# Getting Started with GEM5:

<https://www.gem5.org/documentation/learning_gem5/part1/building/>

### **Building gem5**

This chapter covers the details of how to set up a gem5 development environment and build gem5.

## **Requirements for gem5**

On Ubuntu, you can install all of the required dependencies with the following command. The requirements are detailed below.

sudo apt install build-essential git m4 scons zlib1g zlib1g-dev libprotobuf-dev protobuf-compiler libprotoc-dev libgoogle-perftools-dev python-dev python

**git (**[**Git**](https://git-scm.com/)**):**  
The gem5 project uses [Git](https://git-scm.com/) for version control. [Git](https://git-scm.com/) is a distributed version control system. More information about [Git](https://git-scm.com/) can be found by following the link. Git should be installed by default on most platforms. However, to install Git in Ubuntu use  
 sudo apt install git

**gcc 4.8+**  
You may need to use environment variables to point to a non-default version of gcc.  
 On Ubuntu, you can install a development environment with  
 sudo apt install build-essential

[**SCons**](http://www.scons.org/)gem5 uses SCons as its build environment. SCons is like make on steroids and uses Python scripts for all aspects of the build process. This allows for a very flexible (if slow) build system.  
 To get SCons on Ubuntu use  
 sudo apt install scons

**Python 2.7+**  
gem5 relies on the Python development libraries. To install these on Ubuntu use  
 sudo apt install python-dev

[**protobuf**](https://developers.google.com/protocol-buffers/) **2.1+**

“Protocol buffers are a language-neutral, platform-neutral extensible mechanism for serializing structured data.” In gem5, the [protobuf](https://developers.google.com/protocol-buffers/) library is used for trace generation and playback. [protobuf](https://developers.google.com/protocol-buffers/) is not a required package, unless you plan on using it for trace generation and playback.  
 sudo apt install libprotobuf-dev python-protobuf protobuf-compiler libgoogle-perftools-dev

[Boost](https://www.boost.org/) (**Optional**) :

The Boost library is a set of general purpose C++ libraries. It is a necessary dependency if you wish to use the SystemC implementation.  
 sudo apt install libboost-all-dev

## **Getting the code**

Change directories to where you want to download the gem5 source. Then, to clone the repository, use the git clone command.

git clone https://gem5.googlesource.com/public/gem5

You can now change directories to gem5 which contains all of the gem5 code

SOME Progress following these steps:

1. Built a X86 ISA simulator:
   1. Send the following command from your gem5 dir
      1. cd gem5
      2. Sudo scons build/X86/gem5.fast -j 5
2. Built Ruby simulation to allow CACHE modification
   1. Send the following command from your gem5 dir
      1. Sudo scons buid/X86/gem5.fast RUBY=True PROTOCOL=MESI\_Two\_Level -j5
         1. Where:
            1. X86 : ISA
            2. RUBY=True: invokes the rubi module in gem5
            3. PROTOCOL=MESI\_Two\_Level: allows cache coherence in multi core systems
3. Cross Compile matmult.c for gem5 with x86 ISA
   1. Send the following command from the dir containing matmul.c file
      1. gcc -static -o matmul.x86 matmul.c

NOTE: I followed instructions from part 2 of <http://lacasa.uah.edu/portal/Upload/tutorials/gem5/RunningPrograms-gem5.txt>

1. Send a simulation command:
   1. ./build/X86/gem5.fast configs/example/se.py --num-cpus=1 --ruby --l1d\_size=32kB --l2\_size=512kB --maxinsts=1000000 -c tests/matmul.x86

NOTE: (the input parameter for matmul program is specified using --options switch):

1. Scheck statics at
   1. /gem5/m5out/stats.txt (head /gem5/m5out/stats.txt)

Notes

Gem5 Cache Memory models

1. Classic:
   1. Fast, Flexible, Easy-Configurable Memory Systems
   2. Bus/Crossbar connects the caches memories
   3. MOESI Snoopy cohercy memory Protocol
   4. Faster than Ruby
   5. Suitable for smaller number of multicore systems (less than 8 cores)
2. Ruby:
   1. MessageBuffer connect the cache memories
   2. Supports Different coherence protocols: MSI, MESI, MOESI
   3. Multicore systems

**narek@narek-VB:~/gem5$ ./build/X86/gem5.fast configs/example/se.py --num-cpus=6 --ruby --l1d\_size=8kB --l1i\_size=8kB --l2\_size=64kB --maxinsts=10000 -c matmul.x86 --options="50 50 50"**

system.ruby.hit\_latency\_hist\_seqr::total 16291

system.ruby.l1\_cntrl0.L1Dcache.demand\_accesses 3495 # Number of cache demand accesses

system.ruby.l1\_cntrl0.L1Dcache.demand\_hits 3239 # Number of cache demand hits

system.ruby.l1\_cntrl0.L1Dcache.demand\_misses 256 # Number of cache demand misses

system.ruby.l1\_cntrl0.L1Icache.demand\_accesses 13401 # Number of cache demand accesses

system.ruby.l1\_cntrl0.L1Icache.demand\_hits 13052 # Number of cache demand hits

system.ruby.l1\_cntrl0.L1Icache.demand\_misses 349 # Number of cache demand misses

system.ruby.l1\_cntrl0.mandatoryQueue.avg\_buf\_msgs 0.169400 # Average number of messages in buffer

system.ruby.l1\_cntrl0.mandatoryQueue.avg\_stall\_time 499.994987 # Average number of cycles messages are stalled in this MB

**narek@narek-VB:~/gem5$ ./build/X86/gem5.fast configs/example/se.py --num-cpus=6 --ruby --l1d\_size=16kB --l1i\_size=16kB --l2\_size=64kB --maxinsts=10000 -c matmul.x86 --options="50 50 50"**

system.ruby.hit\_latency\_hist\_seqr::total 16347

system.ruby.l1\_cntrl0.L1Dcache.demand\_accesses 3495 # Number of cache demand accesses

system.ruby.l1\_cntrl0.L1Dcache.demand\_hits 3273 # Number of cache demand hits

system.ruby.l1\_cntrl0.L1Dcache.demand\_misses 222 # Number of cache demand misses

system.ruby.l1\_cntrl0.L1Icache.demand\_accesses 13401 # Number of cache demand accesses

system.ruby.l1\_cntrl0.L1Icache.demand\_hits 13074 # Number of cache demand hits

system.ruby.l1\_cntrl0.L1Icache.demand\_misses 327 # Number of cache demand misses

system.ruby.l1\_cntrl0.mandatoryQueue.avg\_buf\_msgs 0.171080 # Average number of messages in buffer

system.ruby.l1\_cntrl0.mandatoryQueue.avg\_stall\_time 499.994937 # Average number of cycles messages are stalled in this MB

**narek@narek-VB:~/gem5$ ./build/X86/gem5.fast configs/example/se.py --num-cpus=6 --ruby --l1d\_size=64kB --l1i\_size=64kB --l2\_size=128kB --maxinsts=10000 -c matmul.x86 --options="50 50 50"**

system.ruby.hit\_latency\_hist\_seqr::total 16367

system.ruby.l1\_cntrl0.L1Dcache.demand\_accesses 3495 # Number of cache demand accesses

system.ruby.l1\_cntrl0.L1Dcache.demand\_hits 3284 # Number of cache demand hits

system.ruby.l1\_cntrl0.L1Dcache.demand\_misses 211 # Number of cache demand misses

system.ruby.l1\_cntrl0.L1Icache.demand\_accesses 13401 # Number of cache demand accesses

system.ruby.l1\_cntrl0.L1Icache.demand\_hits 13083 # Number of cache demand hits

system.ruby.l1\_cntrl0.L1Icache.demand\_misses 318 # Number of cache demand misses

system.ruby.l1\_cntrl0.mandatoryQueue.avg\_buf\_msgs 0.171222 # Average number of messages in buffer

system.ruby.l1\_cntrl0.mandatoryQueue.avg\_stall\_time 499.994933 # Average number of cycles messages are stalled in this MB

**narek@narek-VB:~/gem5$ ./build/X86/gem5.fast configs/example/se.py --num-cpus=6 --ruby --l1d\_assoc=4 --l1i\_assoc=4 --l1d\_size=64kB --l1i\_size=64kB --l2\_size=128kB --maxinsts=10000 -c matmul.x86 --options="50 50 50"**

system.ruby.hit\_latency\_hist\_seqr::total 16370

system.ruby.l1\_cntrl0.L1Dcache.demand\_accesses 3495 # Number of cache demand accesses

system.ruby.l1\_cntrl0.L1Dcache.demand\_hits 3284 # Number of cache demand hits

system.ruby.l1\_cntrl0.L1Dcache.demand\_misses 211 # Number of cache demand misses

system.ruby.l1\_cntrl0.L1Icache.demand\_accesses 13401 # Number of cache demand accesses

system.ruby.l1\_cntrl0.L1Icache.demand\_hits 13086 # Number of cache demand hits

system.ruby.l1\_cntrl0.L1Icache.demand\_misses 315 # Number of cache demand misses

system.ruby.l1\_cntrl0.mandatoryQueue.avg\_buf\_msgs 0.171222 # Average number of messages in buffer

system.ruby.l1\_cntrl0.mandatoryQueue.avg\_stall\_time 499.994933 # Average number of cycles messages are stalled in this MB

**narek@narek-VB:~/gem5$ ./build/X86/gem5.fast configs/example/se.py --num-cpus=6 --ruby --l1d\_assoc=4 --l1i\_assoc=4 --l1d\_size=64kB --l1i\_size=64kB --l2\_size=128kB --maxinsts=10000 -c matmul.x86 --options="50 50 50"**

system.ruby.hit\_latency\_hist\_seqr::total 16353

system.ruby.l1\_cntrl0.L1Dcache.demand\_accesses 3495 # Number of cache demand accesses

system.ruby.l1\_cntrl0.L1Dcache.demand\_hits 3267 # Number of cache demand hits

system.ruby.l1\_cntrl0.L1Dcache.demand\_misses 228 # Number of cache demand misses

system.ruby.l1\_cntrl0.L1Icache.demand\_accesses 13401 # Number of cache demand accesses

system.ruby.l1\_cntrl0.L1Icache.demand\_hits 13086 # Number of cache demand hits

system.ruby.l1\_cntrl0.L1Icache.demand\_misses 315 # Number of cache demand misses

system.ruby.l1\_cntrl0.mandatoryQueue.avg\_buf\_msgs 0.169182 # Average number of messages in buffer

system.ruby.l1\_cntrl0.mandatoryQueue.avg\_stall\_time 499.994993 # Average number of cycles messages are stalled in this MB

**narek@narek-VB:~/gem5$ ./build/X86/gem5.fast configs/example/se.py --num-cpus=6 --ruby --l1d\_assoc=4 --l1i\_assoc=4 --l1d\_size=64kB --l1i\_size=64kB --l2\_size=128kB --l3\_size=1024kB --maxinsts=100000 -c matmul.x86 --options="50 50 50"**

system.ruby.hit\_latency\_hist\_seqr::total 179368

system.ruby.l1\_cntrl0.L1Dcache.demand\_accesses 47744 # Number of cache demand accesses

system.ruby.l1\_cntrl0.L1Dcache.demand\_hits 47076 # Number of cache demand hits

system.ruby.l1\_cntrl0.L1Dcache.demand\_misses 668 # Number of cache demand misses

system.ruby.l1\_cntrl0.L1Icache.demand\_accesses 132612 # Number of cache demand accesses

system.ruby.l1\_cntrl0.L1Icache.demand\_hits 132292 # Number of cache demand hits

system.ruby.l1\_cntrl0.L1Icache.demand\_misses 320 # Number of cache demand misses

system.ruby.l1\_cntrl0.mandatoryQueue.avg\_buf\_msgs 0.373989 # Average number of messages in buffer

system.ruby.l1\_cntrl0.mandatoryQueue.avg\_stall\_time 499.998963 # Average number of cycles messages are stalled in this MB

**narek@narek-VB:~/gem5$ ./build/X86/gem5.fast configs/example/se.py --num-cpus=6 --ruby --l1d\_assoc=4 --l1i\_assoc=4 --l1d\_size=64kB --l1i\_size=64kB --l2\_size=128kB --l3\_size=4096kB --maxinsts=10000000 -c matmul.x86 --options="50 50 50"**

system.ruby.hit\_latency\_hist\_seqr::total 11201926

system.ruby.l1\_cntrl0.L1Dcache.demand\_accesses 3129531 # Number of cache demand accesses

system.ruby.l1\_cntrl0.L1Dcache.demand\_hits 3128170 # Number of cache demand hits

system.ruby.l1\_cntrl0.L1Dcache.demand\_misses 1361 # Number of cache demand misses

system.ruby.l1\_cntrl0.L1Icache.demand\_accesses 8074141 # Number of cache demand accesses

system.ruby.l1\_cntrl0.L1Icache.demand\_hits 8073756 # Number of cache demand hits

system.ruby.l1\_cntrl0.L1Icache.demand\_misses 385 # Number of cache demand misses

system.ruby.l1\_cntrl0.mandatoryQueue.avg\_buf\_msgs 0.495536 # Average number of messages in buffer

system.ruby.l1\_cntrl0.mandatoryQueue.avg\_stall\_time 499.999978 # Average number of cycles messages are stalled in this MB

**./build/X86/gem5.fast configs/example/se.py --num-cpus=“” --ruby --l1d\_size=“” --l1d\_assoc=“” l1i\_size=“” --l1i\_assoc=“” --l2\_size= “” --l2\_assoc=“” -c matmul.x86 --options="50 50 50"**

**--num-cpus=[1,2,4,8]**

**--l1d\_size=[2KB, 4KB, 8KB, 16K, 32K, 64K]**

**--l1d\_blocksize=[16, 32, 64, 128, 256]**

**--l1d\_assoc=[]**

**--l1i\_size=[]**

**--l1i\_assoc=[]**

**--l2\_size=[]**

**--l2assoc=[]**

* 1. Cache/block size Simulate matmul.c with different data cache and block size. Assume a direct mapped cache. Change the cache sizes as 2K, 4K,8K, 16K, 32K, 64K each with different block sizes: 16, 32, 64, 128, 256. Compare and plot the data cache miss rates. (Note: when you set the block size, the block size for the first level must be no larger than that for the lower level.)

**./build/X86/gem5.fast configs/example/se.py --ruby --l1d\_size=“” --cacheline\_size=“” -c matmul.x86 --options="50 50 50"**

**--l1d\_size=[2KB, 4KB, 8KB, 16K, 32K, 64K]**

**--cacheline\_size=[16, 32, 64, 128, 256]**

* 1. Cache associativity Simulate matmul.c with different set associativities. Assume one level data cache with 8K size. Change the degrees of data cache to be 1, 2, 4, 8, 16. Compare and plot the data cache miss rates.

**/build/X86/gem5.fast configs/example/se.py --ruby --l1d\_size=“” --l1d\_assoc=“” -c matmul.x86 --options="50 50 50"**

**--l1d\_size=[8KB]**

**--l1d\_assoc=[1, 2, 4, 8, 16]**

* 1. Block replacement policy Simulate matmul.c with three different block replacement policies. Assume one level data cache with total cache sizes as 4K and 32K, each of which has degree of set associativity as 1, 4, and 32. Compare and plot the data cache miss rates.

**./build/X86/gem5.fast configs/example/se.py --ruby --l1d\_size=“” --l1d\_assoc=“” -c matmul.x86 --options="50 50 50"**

**--l1d\_size=[4kB, 32kB]**

**--l1d\_assoc=[1, 4, 32]**

* 1. Multilevel Cache Simulate the execution of matmul.c with two levels of data cache. Let the first level cache be 8K direct mapped cache, the second level be 2 way set associative caches with sizes as 8k,16k, 32k, 64k,128k. Compare and plot the local as well as the global miss rate for the second level data cache.

**./build/X86/gem5.fast configs/example/se.py --ruby --l1d\_size=“” --l2\_size= “” -c matmul.x86 --options="50 50 50"**

**--l1d\_size=[8kB]**

**--l2\_size=[8kB, 16kB, 32kB, 64kB, 128kB]**

* 1. Three types of cache misses Simulate the execution of matmul.c with one level of data cache with sizes as 8k,16k, 32k, 64k, 128k and associativity 1,2, 4, 8. Compare and plot the cache miss rate for the capacity and conflict cache misses.

**./build/X86/gem5.fast configs/example/se.py --ruby --l1d\_size=“” --l1d\_assoc=“” -c matmul.x86 --options="50 50 50"**

**--l1d\_size=[“8kB”, “16kB”, “32kB”, “64kB”, “128kB”]**

**--l1d\_assoc=[“1”, “2”, “4”, “8”]**