Gem5 - Overview

MARS Lab, Dept. of CSE, IIT Guwahati

Introduction

- Full system computer architecture simulator.
- Developed by **University of Michigan** (M5) and **University of Wisconsin Madison** (GEMS).
- M5: Simulation Framework, GEMS: Memory System Model
- Encompassing System-level architecture as well as processor microarchitecture.

Why GEM5?

- **C++** and **python** based.
- Flexible (CPU and Memory Models)
- Modular (Object-oriented)
- Event driven (Objects schedules their own events)
- Architecture support: ALPHA, **ARM**, MIPS, Power, **X86**, and many more.
- Vast Community Support: https://www.gem5.org/ask-a-question/

System Modes

- Full System (FS Mode)
  - Simulates a complete system, including devices and an operating system
  - Use Case: Benchmarking applications
- Syscall Emulation (SE Mode)
  - System services provided by the simulator
  - Use Case: OS Fast-boot

CPU Models

- AtomicSimple
- TimingSimple
- InOrder
- O3

For more: https://www.gem5.org/documentation/general_docs/cpu_models/SimpleCPU

In-Order	Out-of-Order
Default 5-stage pipeline (IF,ID,EX,MEM,WB)	Default 7-stage pipeline (IF,ID,Rename,Issue,Ex,WB,Commit)
Configured to model different pipeline stages.	Simulate superscalar architectures

Memory Models

- Classic
- **Ruby** (Cache Coherency)

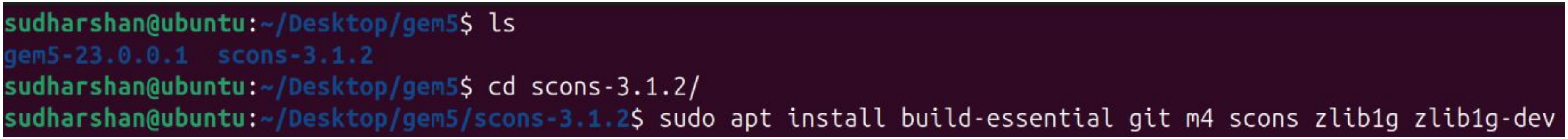
Gem5 Installation in Ubuntu (24.04 LTS)

1) Download gem5-23.0.0.1 and scons-3.1.2 from the below link

   https://drive.google.com/drive/folders/1ILU5UNamAhpKj5YVb99LOepEi8d2mqTs

2) Keep gem5-23.0.0.1 and scons-3.1.2 in one directory.
3) Run the below commands to build gem5

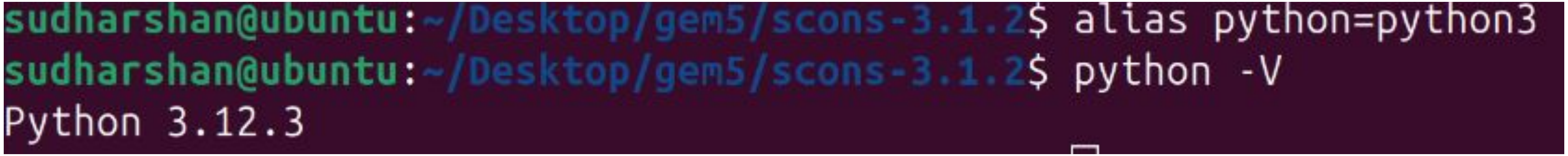
   command-1: “ sudo apt install build-essential git m4 scons zlib1g zlib1g-dev ” command in scons-3.1.2 directory



```
sudharshan@ubuntu:~/Desktop/gem5$ ls
gem5-23.0.0.1  scons-3.1.2
sudharshan@ubuntu:~/Desktop/gem5$ cd scons-3.1.2/
sudharshan@ubuntu:~/Desktop/gem5/scons-3.1.2$ sudo apt install build-essential git m4 scons zlib1g zlib1g-dev
```

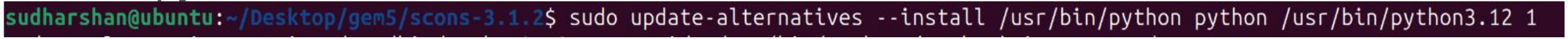
Command-2: “ alias python=python3 ”

Command-3: “ python -V ”



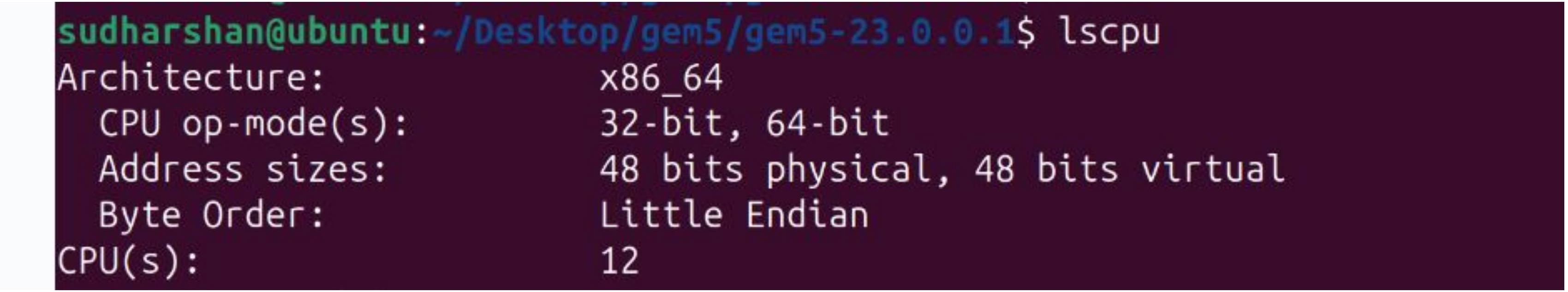
```
sudharshan@ubuntu:~/Desktop/gem5/scons-3.1.2$ alias python=python3
sudharshan@ubuntu:~/Desktop/gem5/scons-3.1.2$ python -V
Python 3.12.3
```

Command-4: “ sudo update-alternatives --install /usr/bin/python python /usr/bin/python3.12 1 ”



```
sudharshan@ubuntu:~/Desktop/gem5/scons-3.1.2$ sudo update-alternatives --install /usr/bin/python python /usr/bin/python3.12 1
```

Command-5: Now go to gem5-23.0.0.1 directory and type “ lscpu ”. Here you can see the no.of CPUs in your system.



```
sudharshan@ubuntu:~/Desktop/gem5/gem5-23.0.0.1$ lscpu
Architecture:             x86_64
  CPU op-mode(s):         32-bit, 64-bit
  Address sizes:          48 bits physical, 48 bits virtual
  Byte Order:             Little Endian
CPU(s):                   12
```

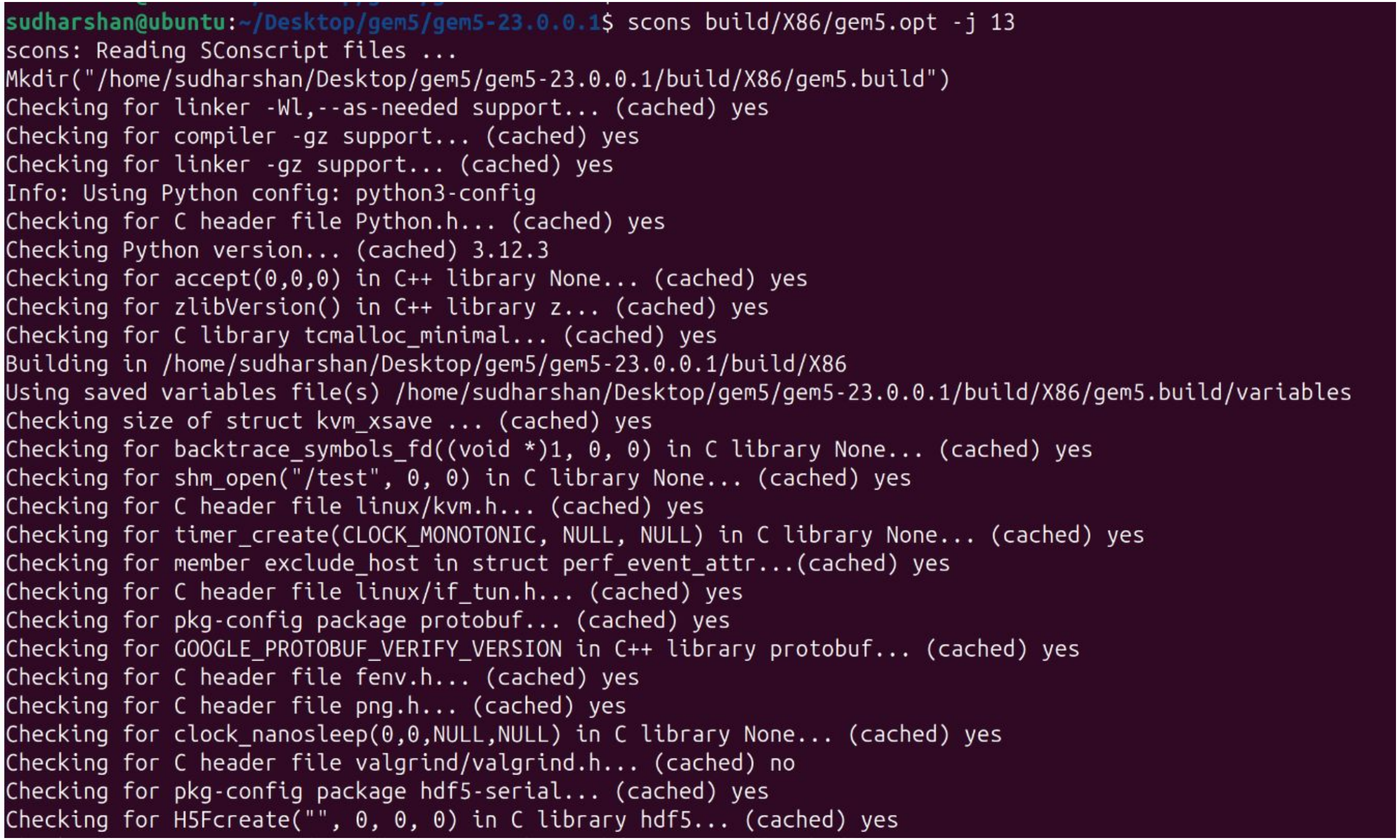
Gem5 Binary Types

The SCons scripts in gem5 currently have 3 different binaries you can build for gem5: debug, opt, and fast

- **debug:** Built with no optimizations and debug symbols.
- **opt:** This binary is build with most optimizations on, but with debug symbols included.
- **fast:** Built with all optimizations on (including link-time optimizations on supported platforms) and with no debug symbols.

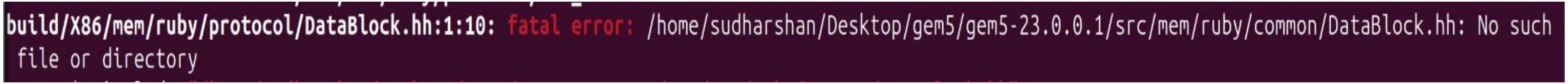
Command-6: scons build/X86/gem5.opt -j <no.of CPUs + 1>

“ scons build/X86/gem5.opt -j 13 ”



```
sudharshan@ubuntu:~/Desktop/gem5/gem5-23.0.0.1$ scons build/X86/gem5.opt -j 13
scons: Reading SConscript files ...
Mkdir("/home/sudharshan/Desktop/gem5/gem5-23.0.0.1/build/X86/gem5.build")
Checking for linker -Wl,--as-needed support... (cached) yes
Checking for compiler -gz support... (cached) yes
Checking for linker -gz support... (cached) yes
Info: Using Python config: python3-config
Checking for C header file Python.h... (cached) yes
Checking Python version... (cached) 3.12.3
Checking for accept(0,0,0) in C++ library None... (cached) yes
Checking for zlibVersion() in C++ library z... (cached) yes
Checking for C library tcmalloc_minimal... (cached) yes
Building in /home/sudharshan/Desktop/gem5/gem5-23.0.0.1/build/X86
Using saved variables file(s) /home/sudharshan/Desktop/gem5/gem5-23.0.0.1/build/X86/gem5.build/variables
Checking size of struct kvm_xsave ... (cached) yes
Checking for backtrace_symbols_fd((void *)1, 0, 0) in C library None... (cached) yes
Checking for shm_open("/test", 0, 0) in C library None... (cached) yes
Checking for C header file linux/kvm.h... (cached) yes
Checking for timer_create(CLOCK_MONOTONIC, NULL, NULL) in C library None... (cached) yes
Checking for member exclude_host in struct perf_event_attr...(cached) yes
Checking for C header file linux/if_tun.h... (cached) yes
Checking for pkg-config package protobuf... (cached) yes
Checking for GOOGLE_PROTOBUF_VERIFY_VERSION in C++ library protobuf... (cached) yes
Checking for C header file fenv.h... (cached) yes
Checking for C header file png.h... (cached) yes
Checking for clock_nanosleep(0,0,NULL,NULL) in C library None... (cached) yes
Checking for C header file valgrind/valgrind.h... (cached) no
Checking for pkg-config package hdf5-serial... (cached) yes
Checking for H5Fcreate("", 0, 0, 0) in C library hdf5... (cached) yes
```

If you see error like “ fatal error: /home/sudharshan/Desktop/gem5/gem5-23.0.0.1/src/mem/ruby/common/DataBlock.hh: No such file or directory ”



```
build/X86/mem/ruby/protocol/DataBlock.hh:1:10: fatal error: /home/sudharshan/Desktop/gem5/gem5-23.0.0.1/src/mem/ruby/common/DataBlock.hh: No such
 file or directory
```

Correct the path in the below files which are located in <PATH_TO_GEM5>/build/X86/mem/ruby/protocol/

a) BoolVec.hh

b) DataBlock.hh

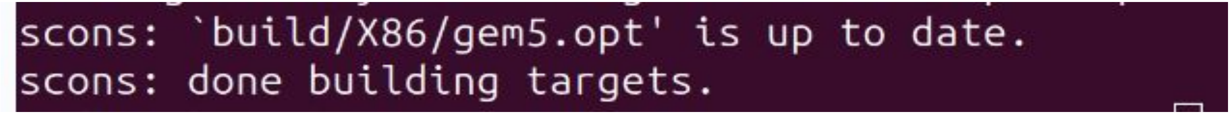
c) Set.hh

d) TBETable.hh

e) TimerTable.hh

f) WriteMask.hh

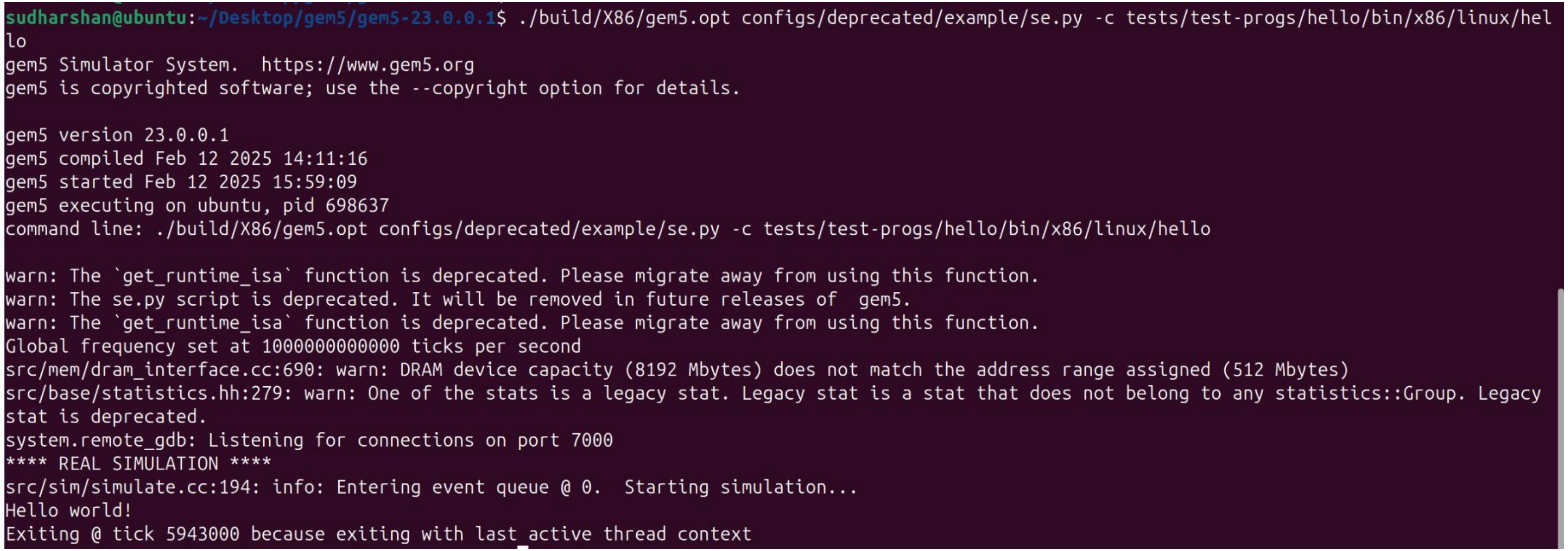
If you get scons: done building targets i.e., you have successfully build the gem5



```
scons: `build/X86/gem5.opt' is up to date.
scons: done building targets.
```

Command-7: To run Hello world! run the below command

“ ./build/X86/gem5.opt configs/deprecated/example/se.py -c tests/test-progs/hello/bin/x86/linux/hello ”



```
sudharshan@ubuntu:~/Desktop/gem5/gem5-23.0.0.1$ ./build/X86/gem5.opt configs/deprecated/example/se.py -c tests/test-progs/hello/bin/x86/linux/hel
lo
gem5 Simulator System.  https://www.gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 version 23.0.0.1
gem5 compiled Feb 12 2025 14:11:16
gem5 started Feb 12 2025 15:59:09
gem5 executing on ubuntu, pid 698637
command line: ./build/X86/gem5.opt configs/deprecated/example/se.py -c tests/test-progs/hello/bin/x86/linux/hello

warn: The `get_runtime_isa` function is deprecated. Please migrate away from using this function.
warn: The se.py script is deprecated. It will be removed in future releases of  gem5.
warn: The `get_runtime_isa` function is deprecated. Please migrate away from using this function.
Global frequency set at 1000000000000 ticks per second
src/mem/dram_interface.cc:690: warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (512 Mbytes)
src/base/statistics.hh:279: warn: One of the stats is a legacy stat. Legacy stat is a stat that does not belong to any statistics::Group. Legacy
stat is deprecated.
system.remote_gdb: Listening for connections on port 7000
**** REAL SIMULATION ****
src/sim/simulate.cc:194: info: Entering event queue @ 0.  Starting simulation...
Hello world!
Exiting @ tick 5943000 because exiting with last active thread context
```

Python in a non-default location

- If you use a non-default version of Python, (e.g., version 3.8 when 2.7 is your default), there may be problems when using SCons to build gem5.
- To fix this, you can force SCons to use your environment's Python version by running **python3 `which scons` build/X86/gem5.opt** instead of **scons build/X86/gem5.opt**.

```
python3 `which scons` build/X86/gem5.opt -j 13
```

Gem5 installation in Docker container

Using a Docker container for running gem5 is recommended due to its numerous dependencies.

1. Navigate to the util folder, where multiple Docker files are available.
2. Use the **all-dependencies** Docker file and install it with the following command:

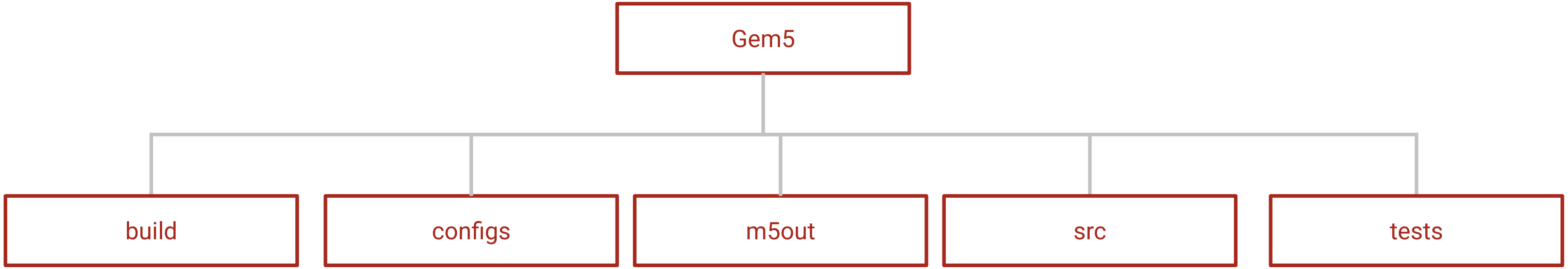
   ```
   cd <path_to_gem5/utils/dockerfiles/ubuntu-24.04_all-dependencies>

   sudo docker build . -t gem5
   ```

3. After creating the Docker image, run the following command to open a Docker container with gem5 mounted:

   ```
    docker run -it --rm -v <path_of_gem5_in_host_pc>:<path_to_be_mounted_in_docker> gem5 /bin/bash
   ```

Gem5 File Structure



build: ISA which is being created

configs: simulation configuration scripts (in python)

m5out: simulation statistics

src: gem5 source code

tests: files used for testing (examples)

Cache Files in Gem5

- src>mem>ruby>structures>CacheMemory.cc/CacheMemory.hh
- src>mem>cache>replacement_policies
- configs>common>Options.py (Default values of various cache parameters)

In the gem5 simulator, users have access to a variety of parameters to configure and customize their simulations. Use the following command to get the list of all available parameters:

Command: “sudo build/X86/gem5.opt configs/deprecated/example/fs.py **--help**”

Running Spec Benchmark

1. Download the SPEC Benchmarks file (Drive Link) and store it in a designated folder.
2. Move the `CPU-2017.py` file to the `common` directory within `gem5/configs`.
3. Update the `CPU-2017.py` configuration file to reflect the correct paths for the benchmark binaries and data.
4. Import the `CPU-2017.py` file into the `se.py` script.
5. In the latest versions of gem5, the **se mode** and **fs mode** have been deprecated, which may cause a **PID-related error**. To resolve this, comment out the relevant lines that use **PID**.
6. Execute the benchmarks using the command below and analyze the generated **stats** files.

Command:

```
./build/X86/gem5.opt -d ./<Output file path> configs/deprecated/example/se.py
--sys-clock=2GHz --mem-size=8GB --caches --l2cache -I 200000 --bench-name=ruby
```

ROI (Region of Interest):

● Specific parts of the simulation where a user wants to focus their analysis.

● Identifying and simulating only the ROI can save time and computational

resources.

–fast-forward:

● It is used to fast forward through a certain number of instructions before starting

the detailed simulation.

● Simulation can focus on ROI.

Sources

- Gem5 Documentation
- Gem5 Installation Files
- Gem5 SPEC Benchmarks
- Gem5 Community Support