

GEM5 Installation

Building Gem5 (Platform Ubuntu):

- Pre-requisites:
> `sudo apt install build-essential git m4 scons zlib1g
zlib1g-dev libprotobuf-dev protobuf-compiler libprotoc-
dev libgoogle-perftools-dev python-dev python`
- Clone the Gem5 repository
> `git clone https://gem5.googlesource.com/public/gem5`
- Build the opt file
> `scons build/ARM/gem5.opt -j<number of cores>`

```
root@vm-cda5106:/opt/gem5# scons build/ARM/gem5.opt -j4
root@vm-cda5106:/opt/gem5# clear
root@vm-cda5106:/opt/gem5# echo "Build Gem5.opt for ARM"
Build Gem5.opt for ARM
root@vm-cda5106:/opt/gem5# scons build/ARM/gem5.opt -j4
scons: Reading SConscript files ...

You're missing the gem5 style or commit message hook. These hooks help
to ensure that your code follows gem5's style rules on git commit.
This script will now install the hook in your .git/hooks/ directory.
Press enter to continue, or ctrl-c to abort:
Warning: Your compiler doesn't support incremental linking and lto at the same time, so lto is being disabled.
To force lto on anyway, use the --force-lto option. That will disable partial linking.
Info: Using Python config: /usr/bin/python2.7-config
Checking for C header file Python.h... yes
Checking for C library python2.7... yes
Checking for C library pthread... yes
Checking for C library dl... yes
Checking for C library util... yes
Checking for C library m... yes
Checking for accept(0,0,0) in C++ library None... yes
Checking for zlibVersion() in C++ library z... yes
```

Figure 1: Building Gem5 opt file for ARM

Testing the build Gem5: (SE Mode)

- Run the following command from the Gem5 directory:
> `./build/ARM/gem5.opt configs/example/arm/starter_se.py --
cpu="minor" tests/test-progs/hello/bin/arm/linux/hello`

The output will be “Hello World” on the screen

```
akash@vm-cda5106:/opt/gem5$ sudo ./build/ARM/gem5.opt configs/example/arm/starter_se.py --cpu="minor" tests/test-progs/hello/bin/arm/linux/hello
gem5 Simulator System. http://gem5.org
gem5 is copyrighted software; use the --copyright option for details.

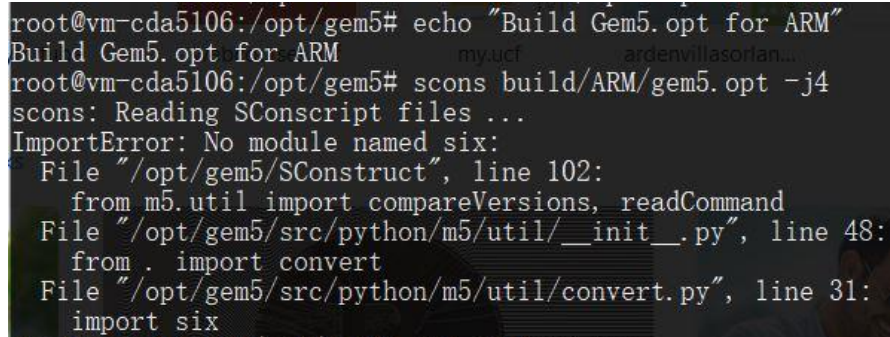
gem5 compiled Sep 21 2019 01:08:26
gem5 started Sep 24 2019 03:05:13
gem5 executing on vm-cda5106, pid 49253
command line: ./build/ARM/gem5.opt configs/example/arm/starter_se.py --cpu=minor tests/test-progs/hello/bin/arm/linux/hello

info: 1. command and arguments: ['tests/test-progs/hello/bin/arm/linux/hello']
Global frequency set at 1000000000000 ticks per second
warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (1024 Mbytes)
warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (1024 Mbytes)
0: system.remote_gdb: listening for remote gdb on port 7000
info: Entering event queue @ 0. Starting simulation...
warn: CP14 unimplemented crn[14], opc1[7], crm[15], opc2[7]
Hello world!
exiting with last active thread context @ 24087000
```

Figure 2: SE Mode "Hello World"

Possible Errors and solutions:

1. Module 'six' not found



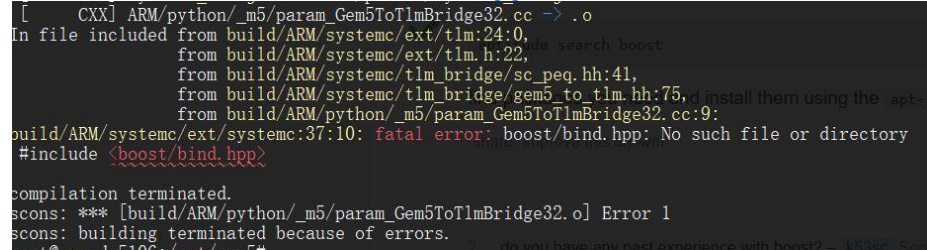
```
root@vm-cda5106:/opt/gem5# echo "Build Gem5.opt for ARM"
Build Gem5.opt for ARM
root@vm-cda5106:/opt/gem5# scons build/ARM/gem5.opt -j4
scons: Reading SConscript files ...
ImportError: No module named six:
  File "/opt/gem5/SConstruct", line 102:
    from m5.util import compareVersions, readCommand
  File "/opt/gem5/src/python/m5/util/__init__.py", line 48:
    from . import convert
  File "/opt/gem5/src/python/m5/util/convert.py", line 31:
    import six
```

Figure 3: No module named six

Solution: Module 'six' is a python module can be installed using 'pip'

```
> pip install six
```

2. Header file not found 'boost/bind.hpp'



```
[CXX] ARM/python/_m5/param_Gem5ToTlmBridge32.cc -> .o
In file included from build/ARM/systemc/ext/tlm:24:0:
    from build/ARM/systemc/ext/tlm.h:22,
    from build/ARM/systemc/tlm_bridge/sc_peq.hh:41,
    from build/ARM/systemc/tlm_bridge/gem5_to_tlm.hh:75:
    from build/ARM/python/_m5/param_Gem5ToTlmBridge32.cc:9:
build/ARM/systemc/ext/systemc:37:10: fatal error: boost/bind.hpp: No such file or directory
#include <boost/bind.hpp>
compilation terminated.
scons: *** [build/ARM/python/_m5/param_Gem5ToTlmBridge32.o] Error 1
scons: building terminated because of errors.
```

Figure 4: Header file missing 'boost/bind.hpp'

Solution: Use apt-get to install the package

```
> apt-get install libboost-all-dev
```

Simulation of Full-System steps for Gem5 (Platform Ubuntu):

Architecture: ARMv8 (64-bit), HPI (In-order)

- Get system disk images, binaries/kernel, DTB files

```
> wget http://www.gem5.org/dist/current/arm/aarch-system-20170616.tar.xz
```
- Extract the files, in a new folder

```
> mkdir fullsystem
> cd fullsystem
> tar xvfJ aarch-system-20170616.tar.xz
```
- Define the environment variable to the disks

```
> echo "export FS_PATH=/path_to_aarch-system-20170616_dir/"
>> ~/.bashrc
> source ~/.bashrc
```

Note: By default, the scripts have path stated as "M5_PATH", defining it as something else require to change the variable in the simulation scripts.

Testing simple FS simulation Gem5: (Minor CPU Mode)

- Execute a simple FS simulation:
> ./build/ARM/gem5.opt
configs/example/arm/starter_fs.py --cpu="minor" -
--num-cores=1 --disk-image=\$FS_PATH/disks/linaro-
minimal-aarch64.img

```
root@vm-cda5106:/opt# ls
fullsystem gem5
root@vm-cda5106:/opt# cd gem5 && ls
binaries directories, i.e. the path to the aarch-system-20170616
root@vm-cda5106:/opt/gem5# $GEM5/build/ARM/gem5.opt /opt/gem5/configs/example/arm/starter_
fs.py --cpu="minor" --num-cores=1 --disk-image=$FS_PATH/disks/linaro-minimal-aarch64.img
gem5 Simulator System... http://gem5.org/20170616.tar.xz
gem5 is copyrighted software; use the --copyright option for details.
ATHS/path to aarch-system-20170616 dir/" >> ~/.bashrc
gem5 compiled Sep 21 2019 01:08:26
gem5 started Sep 22 2019 18:04:39
gem5 executing on vm-cda5106, pid 15672
command line: /opt/gem5/build/ARM/gem5.opt /opt/gem5/configs/example/arm/starter_fs.py --c
pu=minor --num-cores=1 --disk-image=/opt/fullsystem/disks/linaro-minimal-aarch64.img
/opt/configs/example/arm/starter_fs.py --help
Filename: vmlinux.vexpress_gem5_v1_64.20170616 Filepath binaries/vmlinux.vexpress_gem5_v
1_64.20170616 Paths <generator object <genexpr> at 0x7f908456db40>
Filename: /opt/fullsystem/disks/linaro-minimal-aarch64.img Filepath /opt/fullsystem/disk
s/linaro-minimal-aarch64.img Paths <generator object <genexpr> at 0x7f908456daf0>
Filename: boot_emmm.arm64 Filepath binaries/boot_emmm.arm64 Paths <generator object <gen
expr> at 0x7f908456daf0>
Filename: boot_emmm.arm Filepath binaries/boot_emmm.arm Paths <generator object <genexpr
> at 0x7f908456daf0>
in the system console associated with a TCP port. In order to reach the
Global frequency set at 1000000000 ticks per second
warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (2048 M
```

Figure 5: FS mode, CPU="minor"

- Create a new session and attach the terminal to the simulation using the following command

```
> telnet localhost 3456
akash@vm-cda5106:~$ telnet localhost 3456
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^['.
==== m5 slave terminal: Terminal 0 ====
m5 checkpoint * The installation-dependent default.
m5 checkpoint
[ 0.000000] Booting Linux on physical CPU 0x0, the directory containing the input script is calculated at
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Linux version 4.4.0+ (root@bbdeb8fab105) (gcc version 5.4.0 20160609 (Ubuntu
u/Linaro 5.4.0-6ubuntu1~16.04.4) ) #1 SMP PREEMPT Fri Jun 16 09:13:26 UTC 2017
[ 0.000000] Boot CPU: AArch64 Processor [410fc0f0]
[ 0.000000] Memory limited to 2048MB
[ 0.000000] cma: Reserved 16 MiB at 0x00000000ff000000. The directory containing the script bel
[ 0.000000] PERCPU: Embedded 15 pages/cpu @fffffc07efda0000 s23320 r8192 d29928 u61440
[ 0.000000] Detected PIPT I-cache on CPU0
[ 0.000000] Built 1 zonelists in Zone order, mobility grouping on. Total pages: 516096
[ 0.000000] Kernel command line: console=ttyAMA0 lpj=19988480 norandmaps root=/dev/vda1
rw mem=2GB
[ 0.000000] PID hash table entries: 4096 (order: 3, 32768 bytes)
[ 0.000000] Dentry cache hash table entries: 262144 (order: 9, 2097152 bytes)
[ 0.000000] Inode-cache hash table entries: 131072 (order: 8, 1048576 bytes)
[ 0.000000] software IO TLB [mem 0xf8a00000-0xfca00000] (64MB) mapped at [fffffc078a00
000-fffffc07c9ffff]
[ 0.000000] Memory: 1970556K/2097152K available (5342K kernel code, 347K rwddata, 1964K
rdata, 232K init, 237K bss, 110212K reserved, 16384K cma-reserved)
[ 0.000000] Virtual kernel memory layout:
[ 0.000000] vmalloc: 0xfffff80000000000 - 0xfffffbd000000000 ( 246 GB)
[ 0.000000] vmemmap: 0xfffffbd000000000 - 0xfffffbd000000000 ( 8 GB maximum)
[ 0.000000] checks the mod 0xfffffbd000000000 - 0xfffffbd000000000 (version 32 MB actual) it
[ 0.000000] to fixed mpt: 0xfffffbd000000000 - 0xfffffbd000000000 ( 4108 KB) modules are
[ 0.000000] PCI I/O: 0xfffffbd000000000 - 0xfffffbd000000000 ( 16 MB)
[ 0.000000] modules: 0xfffffbd000000000 - 0xfffffbd000000000 ( 64 MB)
[ 0.000000] memory: 0xfffffbd000000000 - 0xfffffbd000000000 ( 2048 MB)
[ 0.000000] init: 0xfffffbd000000000 - 0xfffffbd000000000 ( 232 KB)
[ 0.000000] module text: 0xfffffbd000000000 - 0xfffffbd000000000 ( 7311 KB)
[ 0.000000] module data: 0xfffffbd000000000 - 0xfffffbd000000000 ( 348 KB)
[ 0.000000] SLUB: Hwalign=64, Order=0-3, MinObjects=0, CPUs=1, Nodes=1
[ 0.000000] Preemptible hierarchical RCU implementation.
[ 0.000000] Build-time adjustment of leaf fanout to 64.
[ 0.000000] RCU restricting CPUs from NR_CPUS=256 to nr_cpu_ids=1.
[ 0.000000] RCU: Adjusting geometry for rcu fanout leaf=64, nr_cpu_ids=1
[ 0.000000] NR_IRQS:64 nr_irqs:64 0
```

Figure 6: Telnet to the FS simulation

PARSEC Installation

Compiling PARSEC Benchmarks:

- Download PARSEC files and extract

```
> wget http://parsec.cs.princeton.edu/download/3.0/parsec-3.0.tar.gz
> tar -xvzf parsec-3.0.tar.gz
```
- To enable the benchmark to work on ARM, cross compiling on x86 is to be enabled, for this a few patches are applied to enable it to generate static binaries. Get the arm-rsk repository from github:

```
> wget https://github.com/arm-university/arm-gem5-rsk.git
```

STEP1: From the Parsec3.0 directory apply the static-path.diff

```
> patch -p1 < ../arm-gem5-rsk/parsec_patches/static-path.diff
```

STEP2: To recognize the ARM AArch64 architecture, replace the config.guess and config.sub files.

```
> mkdir tmp; cd tmp
> wget -O config.guess 'http://git.savannah.gnu.org/gitweb/?p=config.git;a=blob_plain;f=config.guess;hb=HEAD'
> wget -O config.sub 'http://git.savannah.gnu.org/gitweb/?p=config.git;a=blob_plain;f=config.sub;hb=HEAD'
> cd /opt/parsec-3.0
> find . -name "config.guess" -type f -print -execdir cp {} config.guess_old \;
> find . -name "config.guess" -type f -print -execdir cp /opt/tmp/config.guess {} \;
> find . -name "config.sub" -type f -print -execdir cp {} config.sub_old \;
> find . -name "config.sub" -type f -print -execdir cp /opt/tmp/config.sub {} \;
```

- Download the aarch64-linux-gnu toolchain from Linaro:

```
> wget https://releases.linaro.org/components/toolchain/binaries/1.1-at90/aarch64-linux-gnu/gcc-linaro-5.5.0-2017.10-x86\_64\_aarch64-linux-gnu.tar.xz
```
- Change the CC_HOME and the BINUTIL_HOME in the xcompile-patch.diff to point to the downloaded <gcc-linaro directory> (/opt/ gcc-linaro-5.5.0-2017.10-x86_64_aarch64-linux-gnu) and <gcc-linaro directory>/aarch64-linux-gnu directories.
- Go to the parsec-3.0 direcotory and apply the xcompile-patch.diff

```
> patch -p1 < ../arm-gem5-rsk/parsec_patches/xcompile-patch.diff
```
- Add the following env variable to *.bashrc* file and source it

```
> export PARSECPLAT="aarch64-linux"
```

- Source the env.sh file in the parsec-3.0 directory
`> source env.sh`
- Build the PARSEC benchmark

```

nvidia-docker0:/opt/parsec-3.0# parasetool -a build -c gcc-hooks -p blacksholes
[PARSEC] Packages to build: parsec.blacksholes

[PARSEC] [===== Building package parsec.blacksholes [1] =====]
[PARSEC] ----- Analyzing package parsec.blacksholes -----
[PARSEC] parsec.blacksholes depends on hooks ----- e.g. the parallel phase
[PARSEC] ----- Analyzing package parsec.hooks -----
[PARSEC] parsec.hooks depends on libtool ----- yes only for the ROI by using the m-related functions, namely on classpath()
[PARSEC] ----- Building package parsec.libtool -----
[PARSEC] parsec.libtool does not depend on any other packages.
[PARSEC] ----- Building package parsec.libtool -----
[PARSEC] Running new tests: /usr/bin/gcc-3.0/gnattools/Libtool/src/configure --build=x86_64-linux-gnu --host=aarch64-linux-gnu --prefix=/opt/parsec-3.0/pgps/tools/Libtool
./install-gcc-hook.sh: Compiling hooks...
##
## Configuring Libtool 2.2.6 ##
##
## On cross-compiling the PARSEC benchmarks for AArch64, we have to configure the PARSEC package to use the
## Arm compiling toolchain. Furthermore, we've to specify the target build-platform for the PARSEC benchmarks.
##
checking for a BSD-compatible install... /opt/parsec-3.0/bin/install -c from Linux
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... ./bin/mkdir -p
checking for gawk... none
checking whether /usr/bin/make sets $(MAKE)... yes
checking for aarch64-linux-gnu-strip... /opt/gcc-linaro-5.5.0-2017.10-x86_64-aarch64-linux-gnu/aarch64-linux-gnu/bin/strip
checking whether arlib and as are usable... yes
checking for aarch64-linux-gnu-gcc... /opt/gcc-linaro-5.5.0-2017.10-x86_64-aarch64-linux-gnu/bin/aarch64-linux-gnu-gcc-5
checking for C compiler default output file name... a.out in the parsetool directory using the command below
checking whether the C compiler works... yes
checking whether we are cross compiling... yes
checking for fix of executables... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking for gcc-linaro-5.5.0-2017.10-x86_64-aarch64-linux-gnu/bin/aarch64-linux-gnu-gcc accepts %x... yes
checking for /opt/gcc-linaro-5.5.0-2017.10-x86_64-aarch64-linux-gnu/bin/aarch64-linux-gnu-gcc option to accept ISO C99... none needed
checking for style of include used by /usr/bin/make... GNU
checking dependency style of /usr/bin/make... GNU
checking whether /opt/gcc-linaro-5.5.0-2017.10-x86_64-aarch64-linux-gnu/bin/aarch64-linux-gnu-gcc and -c understand -c and -o together... yes
checking how to run the C preprocessor... /opt/gcc-linaro-5.5.0-2017.10-x86_64-aarch64-linux-gnu/bin/aarch64-linux-gnu-cpp
checking build system type... x86_64-pc-linux-gnu
checking host system type... aarch64-unknown-linux-gnu
checking for a sed that does not truncate output... /bin/sed
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E

```

Figure 7: Building PARSEC BlackScholes

Running PARSEC Benchmarks:

- The gem5 FS mode does not support shared directories with the host, also the distributed image used for FS simulation does not have enough free space to allocate to PARSEC binaries. So, expand the image.
> cp linaro-minimal-aarch64.img expanded-linaro-minimal-aarch64.img
Add space to the new partition
> dd if=/dev/zero bs=1G count=5 >> ./expanded-linaro-minimal-aarch64.img
> sudo parted expanded-linaro-minimal-aarch64.img
resizepart 1 100%
- Calculate the disk sector
> mkdir disk_mnt
> name=\$(sudo fdisk -l expanded-linaro-minimal-aarch64.img | tail -1 | awk -F: '{ print \$1 }' | awk -F" " '{ print \$1 }')
> start_sector=\$(sudo fdisk -l expanded-linaro-minimal-aarch64.img | grep \$name | awk -F" " '{ print \$2 }')
> units=\$(sudo fdisk -l expanded-linaro-minimal-aarch64.img | grep ^Units | awk -F" " '{ print \$8 }')
- Mount the expanded disk space to the expanded image:
> mount -o loop,offset=\$((start_sector*units)) expanded-linaro-minimal-aarch64.img disk_mnt
- Resize the file system and verify it using the 'df' command:
> resize2fs /dev/loop<X> (X is the number of the mount)
> df -h
- Copy the compiled Parsec-3.0 to the mounted image and then unmount it:

```
> cp -r /path_to_compiled_parsec-3.0_dir/
disk_mnt/home/root
> umount disk_mnt
```

- Create a simulation script for the required package (Freqmine):

```
> cd arm-gem5-rsk/parsec_rcs
> bash gen_rcs.sh -i <simsmall/simmedium/simlarge> -p
freqmine -n <nth>
```

-n: number of threads to use for the script

- Run the simulation with the expanded image and the simulation script:

```
> ./build/ARM/gem5.opt -d fs_results/freqmine
configs/example/arm/starter_fs.py --cpu="hpi" --num-cores=2
--disk-image=$FS_PATH/disks/expanded-linaro-minimal-
aarch64.img --
script=benchmarks/scripts/freqmine simsmall 5.rcS
```

```
root@vm-cda5106:/opt/gem5# ./build/ARM/gem5.opt -d fs_results/freqmine configs/example/a
rm/starter_fs.py --cpu="hpi" --num-cores=2 --disk-image=$FS_PATH/disks/expanded-linaro-m
inimal-aarch64.img --script=benchmark_scripts/freqmine_simsmall_5.rcS
gem5 Simulator System. http://gem5.org
gem5 is copyrighted software; use the --copyright option for details.
```

```
gem5 compiled Sep 21 2019 01:08:26
gem5 started Sep 23 2019 17:33:59
gem5 executing on vm-cda5106, pid 16080
command line: ./build/ARM/gem5.opt -d fs_results/freqmine configs/example/arm/starter_fs
.py --cpu=hpi --num-cores=2 --disk-image=/opt/fullsystem/disks/expanded-linaro-minimal-a
arch64.img --script=benchmark_scripts/freqmine_simsmall_5.rcS
```

```
Filename: vmlinux.vexpress_gem5_v1_64.20170616 Filepath binaries/vmlinux.vexpress_gem5
_v1_64.20170616 Paths <generator object <genexpr> at 0x7fa79ac8dc30>
Filename: /opt/fullsystem/disks/expanded-linaro-minimal-aarch64.img Filepath /opt/full
system/disks/expanded-linaro-minimal-aarch64.img Paths <generator object <genexpr> at
0x7fa79ac8dc30>
```

```
Filename: boot_emmm.arm64 Filepath binaries/boot_emmm.arm64 Paths <generator object <g
enexpr> at 0x7fa79ac8dc30>
```

```
Filename: boot_emmm.arm Filepath binaries/boot_emmm.arm Paths <generator object <genex
pr> at 0x7fa79ac8dc30>
```

```
Global frequency set at 1000000000000 ticks per second
warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (2048
Mbytes)
```

```
info: kernel located at: /opt/fullsystem/binaries/vmlinux.vexpress_gem5_v1_64.20170616
```

```
warn: Bootloader entry point 0x10 overriding reset address 0
```

```
warn: Highest ARM exception-level set to AArch32 but bootloader is for AArch64. Assuming
you wanted these to match.
```

```
warn: No functional unit for OpClass SimdDiv
```

```
warn: No functional unit for OpClass SimdReduceAdd
```

```
warn: No functional unit for OpClass SimdReduceAlu
```

```
warn: No functional unit for OpClass SimdReduceCmp and this post helpful.
```

```
warn: No functional unit for OpClass SimdFloatReduceAdd
```

```
warn: No functional unit for OpClass SimdFloatReduceCmp
```

```
warn: No functional unit for OpClass SimdAes
```

Figure 8: Executing FS Simulation