Project Profile for ***Group 10*** Date: Sep 23rd, 2019

Original/Revised: Original

I(a.) If you have installed ***gem5*** and ***PARSEC*** on your laptop discuss briefly:

- When and how you tested the installation, what was the input, did you run a test case:

  Installed on: Sep 16th, 2019

  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>
    ```

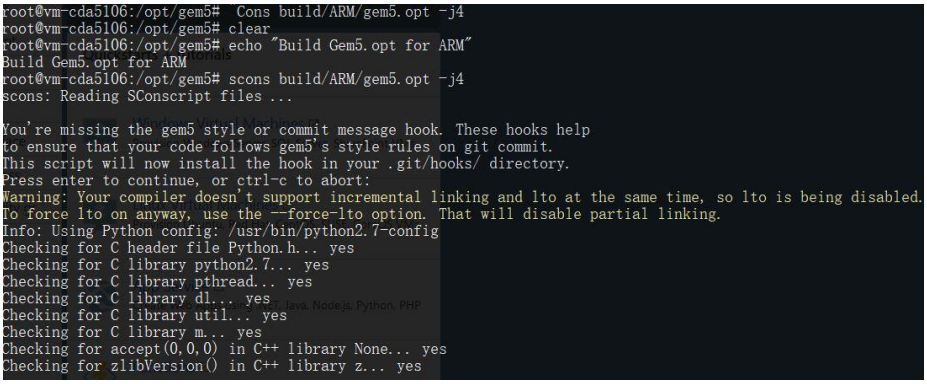

Figure 1: Building Gem5 opt file for ARM

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/starte
r_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 assign
ed (1024 Mbytes)
warn: DRAM device capacity (8192 Mbytes) does not match the address range assign
ed (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"

Errors encountered:

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
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
gem5 is copyrighted software; use the --copyright option for details.

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

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_emm.arm64 Filepath  binaries/boot_emm.arm64  Paths  <generator object <gen
expr> at 0x7f908456daf0>
Filename:  boot_emm.arm Filepath  binaries/boot_emm.arm  Paths  <generator object <genexpr
> at 0x7f908456daf0>
Global frequency set at 1000000000000 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
  ```

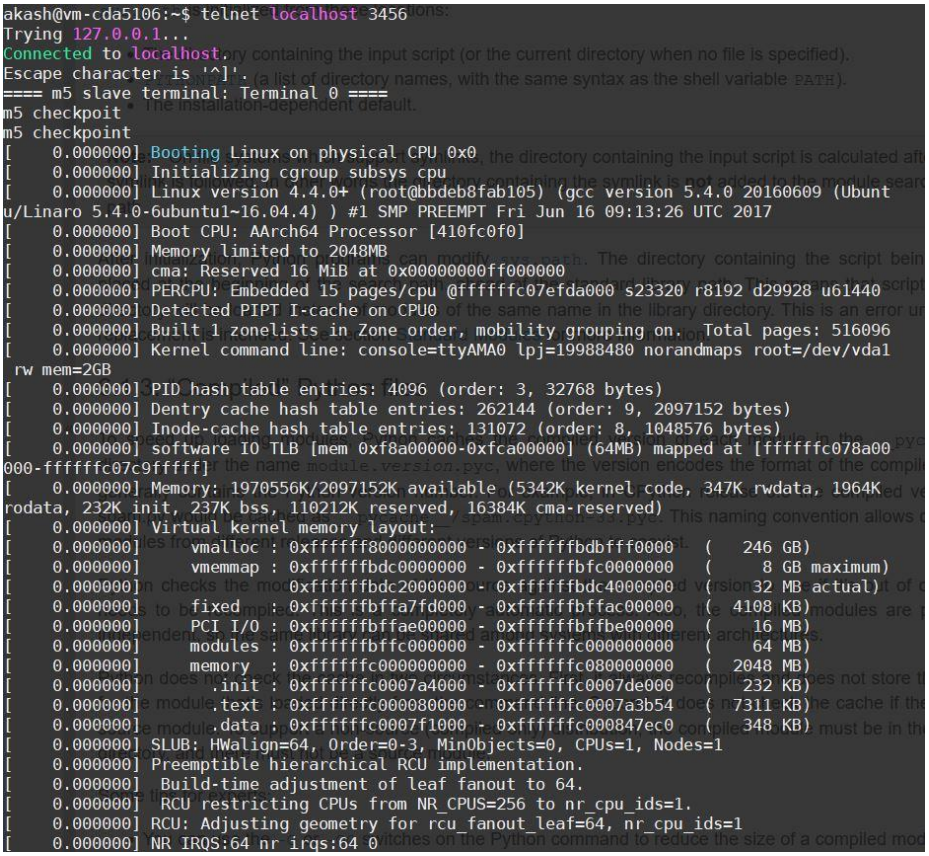

Figure 6: Telnet to the FS simulation

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-patch.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/latest-5/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 direcetory 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

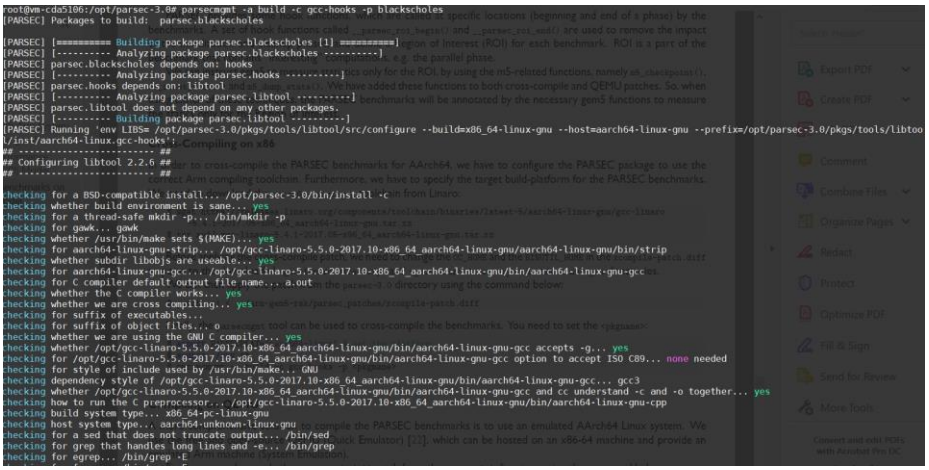

Figure 7: Building PARSEC BlackScholes

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
```

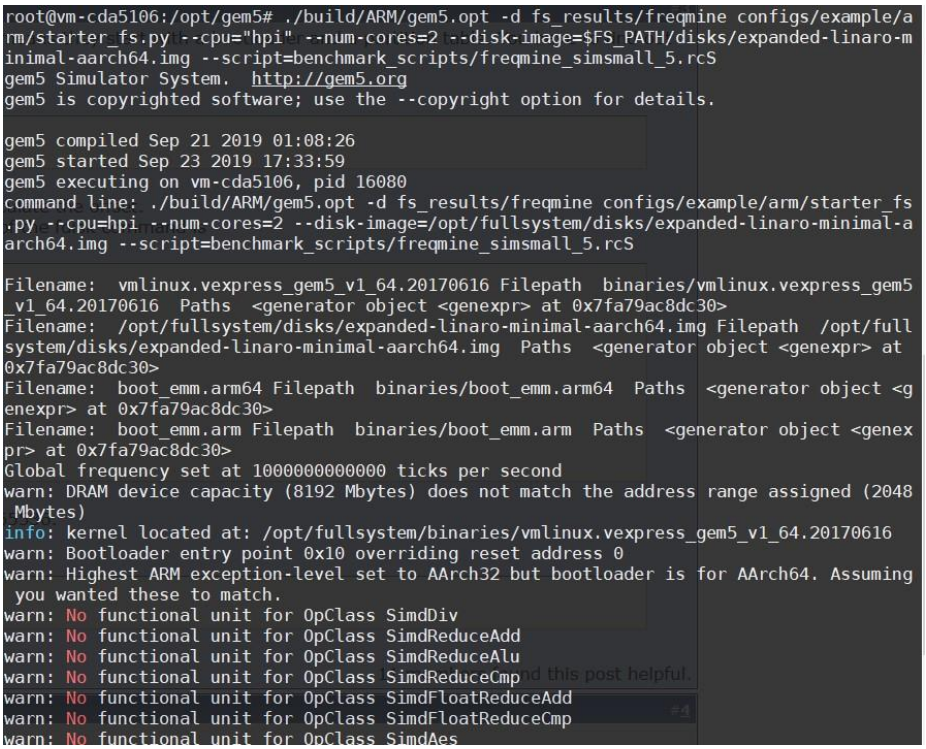

Figure 8: Executing FS Simulation

I(b.) If you used the already installed software on ***Eustis***:

When and how you tested the installation, what was the input, did you run a test case:

The problems you encountered and how you solved them.

II. The type of experiment you intend to perform. Provide enough details:

Try and vary the system cache sizes to monitor change in read/write miss rates and power consumption variation.

III. Why did you choose this particular experiment?

Memory hierarchy has been covered in class and performance and optimization was at forefront. The purpose of the experiment is to test the theory on our own.

IV. Did you find literature describing an identical or similar experiment?

No

V. The simulation mode (I strongly suggest the FS mode):

Full System (ARM V8 64Bit)

VI. The CPU and the memory models (I suggest CPU modes c or d):

High Performance In-Order (HPI)

VII. The input: specify the PARSEC benchmark name or list the code:

Freqmine

VIII. What is the running time of the test you performed?

One test Run of Freqmine with small data set took 47 mins

IX. How confident are you that the experiment you propose is feasible?

Using Microsoft Azure cloud to host the machines for simulation , so resources are not an issue.