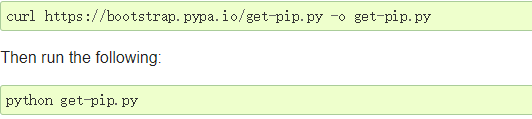
Centos7

1. 安装Mercurial ，这样可以采用hg命令来获取gem5

pip install Mercurial即可

如果python 版本中没有pip ，可以升级python或是用以下命令安装pip

yum install pyton-devel.x86\_64



本机python 2.7.5通过pip安装的Mercurial

1. hg clone <http://repo.gem5.org/gem5>直接使用

**如果要和nvmain进行混合编译的话**，请hg parent查看版本，最好使用版本12613:40c18bb90501

如果不是这个版本，则hg revert -r 40c18bb90501 --all回退到该版本，然后执行hg commit

1. 安装scons

yum install scons 默认安装了3.0.1版本

1. **编译scons build/X86/gem5.opt –j9**

会出现warning：Protocol buffer compiler (protoc) not found. Please install protobuf-compiler for tracing support 不要理会这个错误，不要安装protobuf相关包，否则会出现protoc相关的错误（原因是protoc2和protoc3的语法问题），不过最简单的方式就是不安装protobuf。

1. 运行hello示例判断gem5是否安装成功

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

1. hg clone <https://bitbucket.org/mrp5060/nvmain>

hg parent查看版本，最好使用版本373:4ee7a07f5c43

如果不是这个版本，则hg revert -r 4ee7a07f5c43 --all回退到该版本然后执行hg commit

1. 按照以下步骤安装

在user用户下 创立~/.hgrc内容为：

[ui]

username = wubing <wubin200@126.com>

[extensions]

mq =

这样就使得user可以使用hg中的高级命令qpush等等

在root用户下 创建文件~/.hgrc 内容同上，则使得root可以使用hg高级命令

if you are using a verison of gem5 cloned from

the read-only mercurial mirror, mercurial queues

can be used to apply the patch. To start, go to

the gem5 root directory and import the patch

$ hg qimport /path/to/nvmain/patches/gem5/nvmain2-XYZ

Apply the path using qpush:

$ hg qpush

You can check that the patch was applied using

qapplied:

$ hg qapplied

You can build gem5 normally at this point.

You will need mercurial queues setup to do

this. More information can be found at

http://mercurial.selenic.com/wiki/MqExtension/

In general you should add the following to ~/.hgrc:

[extensions]

mq =

and you should initialize queues in gem5 using:

$ hg qinit

To update the gem5 patch, first remove it and

delete it. It can then be imported again. In

order to save your modifications, you can create

your own patch before re-importing the nvmain

patch:

$ hg qnew mychanges.patch

$ hg qrefresh

Now your changes are saved in the "mychanges.patch"

file. Feel free to rename this :). Next re-import

NVmain:

$ hg qpop -a

$ hg qdel nvmain-XYZ

$ hg qimport /path/to/nvmain/patches/gem5/nvmain-XYZ

$ hg qpush -a

Nvmain安装完毕后：

**在gem5下configs/example中 放入se2.py 和 Mybench.py两个python**

**Mybench.py中需要修改第5行和第6行，binary\_dir以及data\_dir位置。**

采用se2.py和Mybench.py则使用/home/bing/wcn/cpu2006这个目录下的1.1版本的cpu2006。可自行将该cpu2006打包拷走 (无需编译，已经编译好了)

使用以下命令测试

./build/X86/gem5.opt --redirect-stdout --outdir=./test/perlbench ./configs/example/se2.py --cpu-type=DerivO3CPU --caches --l1d\_size=16kB --l1i\_size=16kB --l2cache --l2\_size=4MB --l1d\_assoc=2 --l1i\_assoc=2 --l2\_assoc=16 --mem-type=NVMainMemory --mem-size=4GB --nvmain-config=./nvmain/Config/2D\_DRAM\_example.config -b perlbench -I 50000000 &

如果采用se3.py和Mybench\_0.py则使用的是/home/bing/cpu2006 目录下1.2版本的cpu2006. 1.2版本的cpu2006为了测试更多的负载，还需要指定spec\_data等文件。具体细节，见附录（暂时没写）。