

利用GEM5评测系统性能

智能1602 201607020301 孟祥炜

一、GEM5 简介

- ▶ gem5主要用C++和python编写，大多数组件都是在BSD风格的许可下提供的。
- ▶ 它可以在完整系统模式（FS模式）下模拟具有设备和操作系统的完整系统，或者在系统调用模式（SE模式）下由模拟器直接提供系统服务的用户空间程序。
- ▶ 在CPU模型上执行Alpha, ARM, MIPS, Power, SPARC和64位x86二进制文件有不同程度的支持，包括两个简单的单个CPI模型，一个乱序模型和一个按顺序流水线模型。
- ▶ 内存系统可以灵活地由缓存和交叉开关构建。最近，Ruby模拟器已经与gem5集成，以更灵活的内存系统建模。

二、GEM5的 安 装

► 1. 安装各类库文件及编译器:

命令行输入 `sudo apt-get install mercurial scons swig gcc m4 python python-dev libgoogle-perftools-dev g++ libprotobuf-dev`
`sudo apt-get install build-essential`

► 2. 下载GEM5源码

`hg clone http://repo.gem5.org/gem5`

► 3. 编译X86架构

`scons build/X86/gem5.opt`

► 4. 运行GEM5自带的测试程序检验安装是否成功

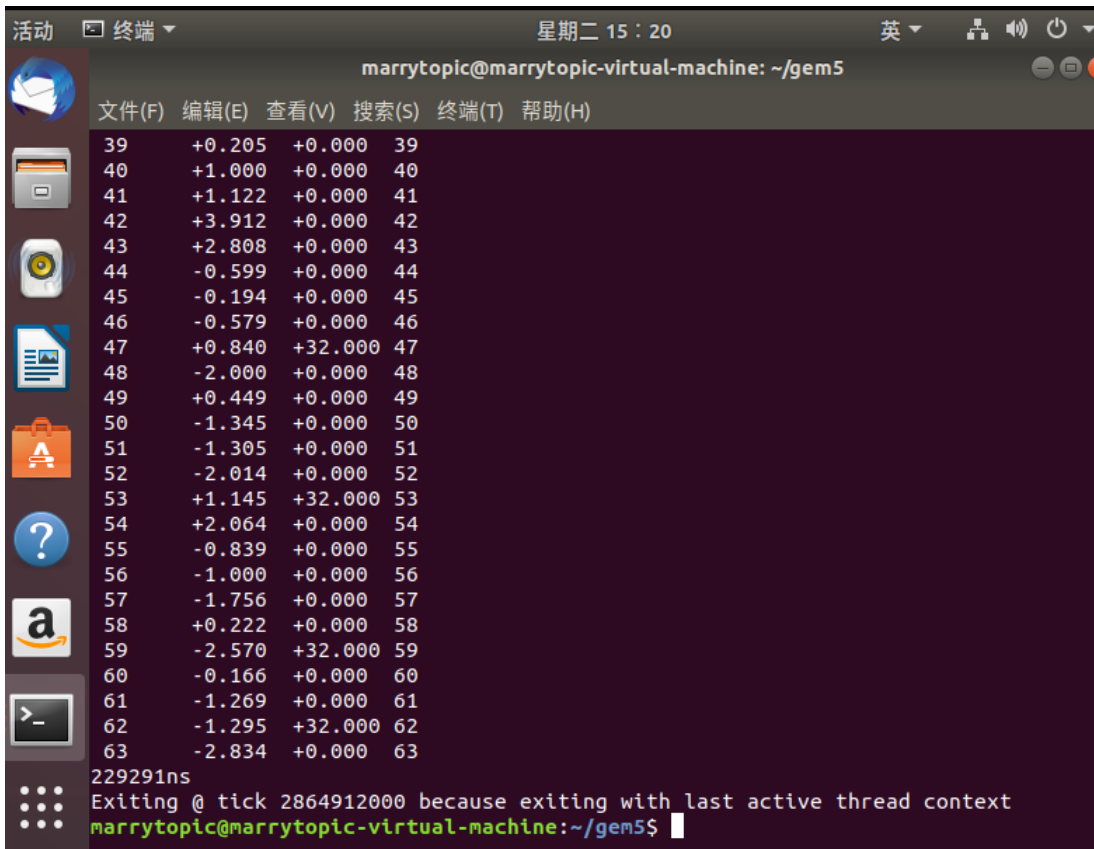
`build/X86/gem5.opt configs/example/se.py -c tests/test-progs/hello/bin/x86/linux/hello`

运行程序之后, 如果打印出 “hello,world”, 即表示GEM5安装成功

三、使用GEM5测试多线程FFT程序

- ▶ GEM5支持两种模拟模式，全系统模式 (FS) 和 系统调用模式 (SE)
- ▶ 由于我们只是测试FFT程序的性能，因此使用 SE模式即可

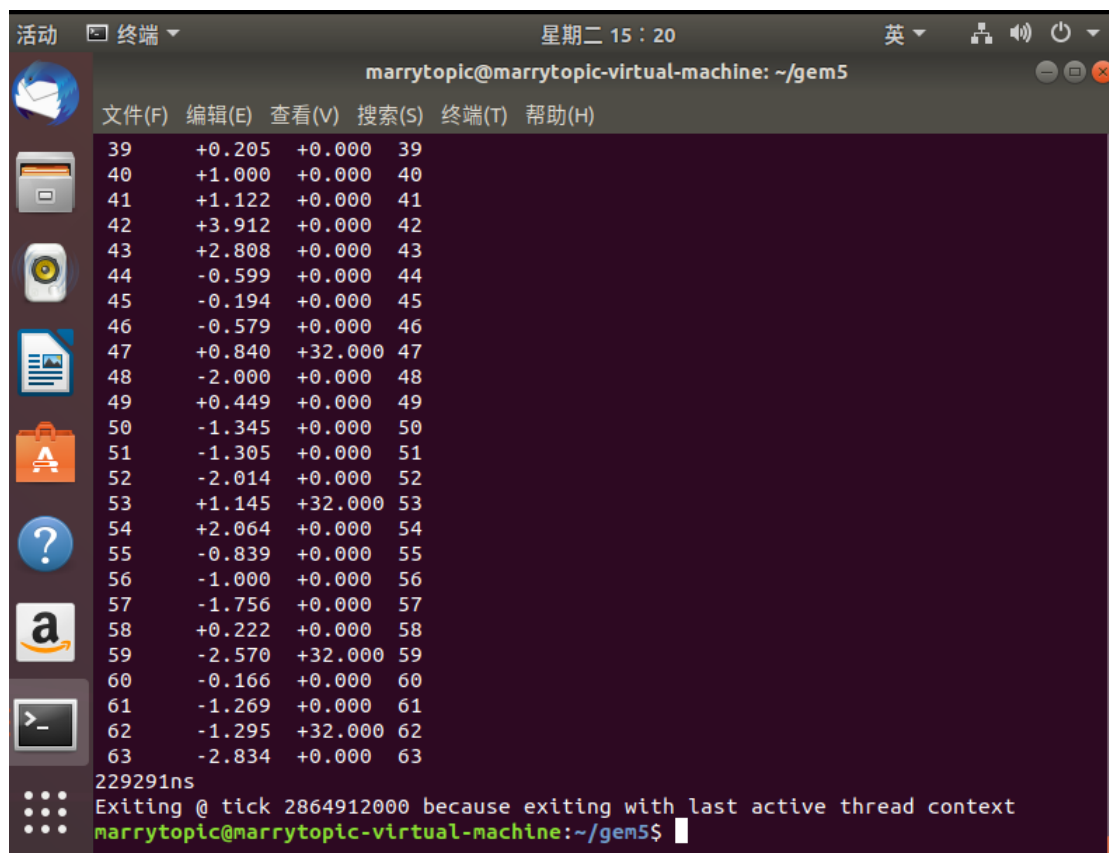
- ▶ FFT程序在GEM5上的运行结果



```
marrytopic@marrytopic-virtual-machine: ~/gem5
39 +0.205 +0.000 39
40 +1.000 +0.000 40
41 +1.122 +0.000 41
42 +3.912 +0.000 42
43 +2.808 +0.000 43
44 -0.599 +0.000 44
45 -0.194 +0.000 45
46 -0.579 +0.000 46
47 +0.840 +32.000 47
48 -2.000 +0.000 48
49 +0.449 +0.000 49
50 -1.345 +0.000 50
51 -1.305 +0.000 51
52 -2.014 +0.000 52
53 +1.145 +32.000 53
54 +2.064 +0.000 54
55 -0.839 +0.000 55
56 -1.000 +0.000 56
57 -1.756 +0.000 57
58 +0.222 +0.000 58
59 -2.570 +32.000 59
60 -0.166 +0.000 60
61 -1.269 +0.000 61
62 -1.295 +32.000 62
63 -2.834 +0.000 63
229291ns
Exiting @ tick 2864912000 because exiting with last active thread context
marrytopic@marrytopic-virtual-machine: ~/gem5$
```

四、单线程FFT在GEM5和本机上性能对比

► GEM5

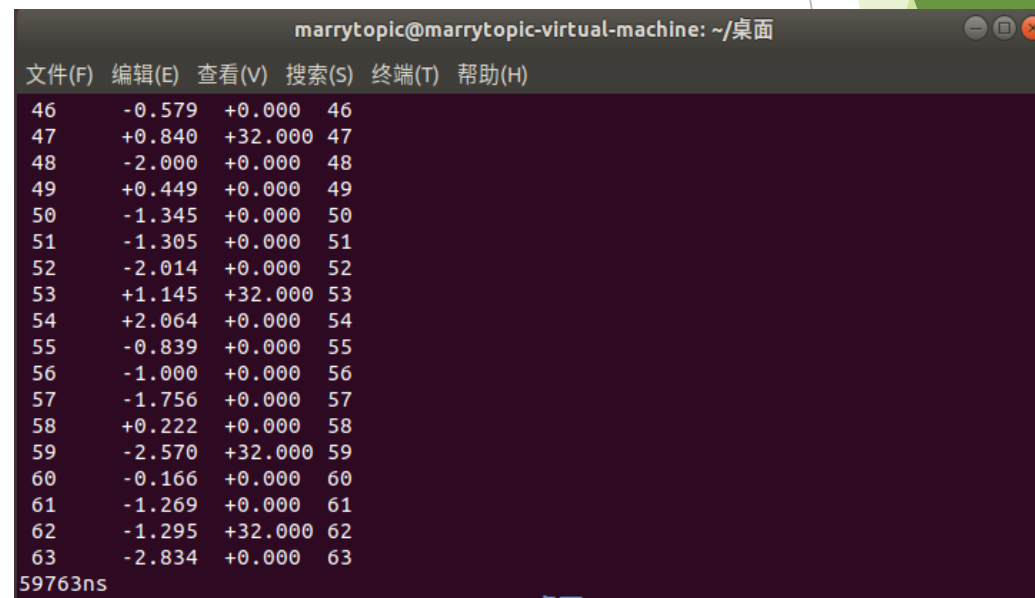


The screenshot shows a terminal window titled "marrytopic@marrytopic-virtual-machine: ~/gem5". The window displays a table of FFT performance results for various input sizes. The table has four columns: input size, real part, imaginary part, and output size. The results are as follows:

| Input Size | Real Part | Imaginary Part | Output Size |
|------------|-----------|----------------|-------------|
| 39 | +0.205 | +0.000 | 39 |
| 40 | +1.000 | +0.000 | 40 |
| 41 | +1.122 | +0.000 | 41 |
| 42 | +3.912 | +0.000 | 42 |
| 43 | +2.808 | +0.000 | 43 |
| 44 | -0.599 | +0.000 | 44 |
| 45 | -0.194 | +0.000 | 45 |
| 46 | -0.579 | +0.000 | 46 |
| 47 | +0.840 | +32.000 | 47 |
| 48 | -2.000 | +0.000 | 48 |
| 49 | +0.449 | +0.000 | 49 |
| 50 | -1.345 | +0.000 | 50 |
| 51 | -1.305 | +0.000 | 51 |
| 52 | -2.014 | +0.000 | 52 |
| 53 | +1.145 | +32.000 | 53 |
| 54 | +2.064 | +0.000 | 54 |
| 55 | -0.839 | +0.000 | 55 |
| 56 | -1.000 | +0.000 | 56 |
| 57 | -1.756 | +0.000 | 57 |
| 58 | +0.222 | +0.000 | 58 |
| 59 | -2.570 | +32.000 | 59 |
| 60 | -0.166 | +0.000 | 60 |
| 61 | -1.269 | +0.000 | 61 |
| 62 | -1.295 | +32.000 | 62 |
| 63 | -2.834 | +0.000 | 63 |

Below the table, the terminal shows the execution time: "229291ns". At the bottom, it says "Exiting @ tick 2864912000 because exiting with last active thread context" and "marrytopic@marrytopic-virtual-machine:~/gem5\$".

► 本机 (ubuntu 18.04)



The screenshot shows a terminal window titled "marrytopic@marrytopic-virtual-machine: ~/桌面". The window displays a table of FFT performance results for various input sizes. The table has four columns: input size, real part, imaginary part, and output size. The results are as follows:

| Input Size | Real Part | Imaginary Part | Output Size |
|------------|-----------|----------------|-------------|
| 46 | -0.579 | +0.000 | 46 |
| 47 | +0.840 | +32.000 | 47 |
| 48 | -2.000 | +0.000 | 48 |
| 49 | +0.449 | +0.000 | 49 |
| 50 | -1.345 | +0.000 | 50 |
| 51 | -1.305 | +0.000 | 51 |
| 52 | -2.014 | +0.000 | 52 |
| 53 | +1.145 | +32.000 | 53 |
| 54 | +2.064 | +0.000 | 54 |
| 55 | -0.839 | +0.000 | 55 |
| 56 | -1.000 | +0.000 | 56 |
| 57 | -1.756 | +0.000 | 57 |
| 58 | +0.222 | +0.000 | 58 |
| 59 | -2.570 | +32.000 | 59 |
| 60 | -0.166 | +0.000 | 60 |
| 61 | -1.269 | +0.000 | 61 |
| 62 | -1.295 | +32.000 | 62 |
| 63 | -2.834 | +0.000 | 63 |

Below the table, the terminal shows the execution time: "59763ns".

五、结果分析

- ▶ 从测量所得的数据可以看出，程序在gem5上的运行时间是在linux系统上运行时间的4倍左右。
- ▶ 可以看到，同样的程序在gem5中运行时间比在linux系统运行时间慢。
- ▶ Gem5作为一个模拟平台，程序在模拟平台上的速度肯定会比在本机系统上的慢一些，主要原因是，程序在本机上运行时，直接调用本机系统的相关链接库等文件即可执行，而在模拟平台上运行时，需要再模拟平台上调用模拟平台的文件，而模拟平台的文件的运行需要再次调用本机的相关文件执行，因此过多的调用导致运行时间较长。同时，gem5的存储、处理结构、IO等都与本机上不同，且比本机慢，因此，程序在gem5上的运行速度比在本机上的运行速度慢。

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

谢谢观看

智能1602 201607020301 孟祥炜