

实验报告

实验名称（利用 GEM5 评测系统性能）

智能 1602 201608010609 李鹏飞

实验目标

利用 GEM5 仿真 x86-64 系统，测量 FFT 程序运行时间，与实际 x86-64 系统的测试结果进行比较。

实验要求

- * 采用 C/C++编写程序，选择合适的运行时间测量方法
- * 根据自己的机器配置选择合适的输入，保证足够长度的运行时间
- * 回答思考题，答案加入到实验报告叙述中合适位置

思考题

1. GEM5 是什么？怎么使用？
2. 利用仿真器评测系统性能的优点是什么？缺点是什么？
3. GEM5 仿真系统测得的程序性能与实际系统测得的程序性能差别大不大？可能的原因是什么？

实验内容

GEM5 的安装和使用

GEM5 的官方文档在[这里](<http://gem5.org/Documentation>), 包括了安装和使用说明。

GEM5 网站也提供了 x86-64 系统 Linux 镜像下载，可以在 GEM5 的[下载](<http://gem5.org/Download>)页面找到。

大家也可以参考[这篇博客文章](<https://blog.csdn.net/u012822903/article/details/62444286>)及其相关文章，了解如何利用 GEM5 进行 Linux 全系统模拟和运行自己的测试程序。

利用 GEM5 测试 FFT 程序在 x86-64 系统上的性能

在安装好 GEM5，并掌握如何使用 GEM5 运行自己的测试程序后，可以在 GEM5 上运行 FFT 程序，测试其性能。使用实验一中的单线程 FFT 程序，记录测试数据。

测试

测试平台

在如下机器上进行了测试：

部件	配置	备注
:----- :-----: :-----		
CPU	core i5-6500U	
内存	DDR4 4GB	
操作系统	Ubuntu 18.04 LTS	中文版

测试记录

在 GEM5 模拟器中的测试数据：

```
File Edit View Search Terminal Help
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
50000/30000000(s)
(none) / #
```

在实际系统中的测试数据：

```
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
10142/30000000(s)
```

我们发现实际系统中测试数据大约是 GEM5 中测试数据的五分之一

思考题

1.GEM5 是一个 CPU 模拟器

GEM5 模拟器提供了四个不同的 CPU 模型,两个不同的系统模型以及两个不同的内存系统模型,并且支持多种指令集(ARM、ALPHA、MIPS、Power、SPARC 和 x86),其中可以在 ARM、ALPHA 和 x86 三种架构上运行 Linux。

在这里我们利用了 linux 上的 x86 框架进行全系统仿真。

全系统仿真的步骤：

(1)新建文件夹 mountfile，用于存放挂载的文件

(2)进行系统挂载，具体命令如下：

```
sudo mount -o,loop,offset=32256 fs-image/disks/linux-x86.img /mnt
```

(3)执行 umount 操作，具体命令如下：

```
sudo umount /mnt
```

(4)重新启动 gem5 的 FS，命令如下：

```
sudo build/X86/gem5.opt configs/example/fs.py
```

(5)打开另一个终端，输入如下命令：

```
sudo ./m5term 127.0.0.1 3456
```

(6)编译源代码，等待很长时间(在这里之所以使用源代码编译是因为 gem5 中编译器版本不匹配，在本机上编译好的静态可执行文件无法正常执行)

(7)执行可执行文件

2.使用仿真模拟器测评系统性能的优缺点：

优点：

利用仿真器进行系统测评最大的优点就是省钱省力省时间。因为芯片的仿真需要大量的时间精力和金钱。如果我们生产出来的 CPU 不能达到我们的预期，那么我们除了宝贵的经验什么也没有得到。所以利用模拟器就可以避免这种情况。节省资源。

缺点：

(1)仿真器制作难度大，一款强大的仿真器结构十分复杂。因而制作难度大，功能不够完善，这就可能会造成一定的误差。

(2)仿真器的仿真时间过长，因为在仿真器中，一条指令的编译可能需要很多条指令来解释，所以大型 CPU 仿真等待时间会比较长。

3.GEM5 仿真系统测得的程序性能与实际系统测得的程序性能差别很大，从我们的实验结果中可以看出性能相差 5 倍左右，这是因为我们的模拟器在进行模拟编译的时候源代码中的一条指令需要数条指令去解释，这就导致我们的性能差别很大。

分析和结论

从测试记录来看，FFT 程序在 GEM5 上的执行时间实际系统上测得的执行时间的 5 倍，所以可以分析得到一条指令的执行大约需要 5 条指令去解释。