2018 计算机系统设计实验报告

湖南大學

HUNAN UNIVERSITY

实验名称：测量 FFT 程序执行时间

姓　　名：孟祥炜

学　　号：201607020301

专业班级：智能 1602

一、 实验目标

测量 FFT 程序运行时间，确定其时间复杂度。

二、 实验要求

- 采用 C/C++编写程序
- 根据自己的机器配置选择合适的输入数据大小 n，至少要测试多个不同的 n (参见思考题)
- 对于相同的 n，建议重复测量 30 次取平均值作为测量结果 (参见思考题)
- 对测量结果进行分析，确定 FFT 程序的时间复杂度
- 回答思考题，答案加入到实验报告叙述中合适位置

三、 实验内容

FFT 算法对应的 C++代码已经给出（这里就不复制粘贴了）

fft 蝶形算法要求 N 是 2 的整数次幂，当输入不为 2 的整数次幂时，比如输入 1000，那么对于 1001~1024 的点，就会按 0 处理,导致结果的不准确。当 n 越大，频率分辨率越大，但时间分辨率会减小。所以 N 取适当的值即可。

在这里我们取 N=16、32、64、128、256、512

删去所给代码中输出的部分，对快速傅里叶变换运算部分进行计时，同时根据实验要求，对相同的 n，重复测量 30 次，取平均值作为测量结果。
（原因，每次的 FFT 过程不同，因此每次运行的时间都不相同，取 30 次的平均值是为了减小偶然误差的影响）
使用 rdtscp 工具通过时间戳的记录进行精准计时。（好像这个工具和 perf 工具是用的同样的方法）

得到以下表格：

N	16	32	64	128	256	512
Time(ns)	36544	97792	373115	580312	1600488	3255520

（取 30 次平均值）
假设材料所给的时间复杂度公式是正确的。

$$a*n*logn+\frac{b}{3}*n+\sqrt{2}*c*logn+d$$

借助 matlab 对参数 a、b、c、d 进行求解
//保留四位小数
a=-158.7644
b=2362.5180
c=4186.8269
d=15640.7302
（右图）

```
ans =

   1.0e+05 *

  -0.001587643569160
   0.256251804648907
  -0.418682693245918
   1.564073023994374
```

将解带回时间复杂度方程，观察理论计算结果（下图）

```
ans =

   1.0e+06 *

   0.046071315455187
   0.108286892025406
   0.286848260056457
   0.693020751587862
   1.544254571277347
   3.265289209597742
```

绘制表格，进行对比

理论时间(ns)	46071	108289	286848	693021	1544255	3265289
实际时间(ns)	36544	97792	373115	580312	1600488	3255520
时间误差	26.07%	10.73%	23.12%	19.42%	3.51%	0.30%

最大误差为 26.07%，误差较大，不能够验证该公式的正确性。

在我自己运行调试的过程中，发现尽管已经取了 30 次作为平均值，但是每次运行的结果依然差距较大，最大的时候达到 2~3 倍，这可能是导致理论时间与实际时间有较大误差的原因。经过调试，发现，当取到 10000 次的时候，多次实际运行时间基本维持在±5~10%的误差范围，因此修改程序，改取 10000 次平均值，得到下表

N	16	32	64	128	256	512
Time(ns)	37374	82695	190307	424069	930303	2109737

（取 10000 次平均值）

同样使用 matlab 求解参数

得到结果

a=644.5265
b=-5818.2419
c=15178.9488
d=-61184.3992
（右图）

```
ans =

   1.0e+04 *

   0.064452646436133
  -0.581824192836997
   1.517894878149636
  -6.118439920019193
```

将解代回时间复杂度方程，观察理论计算结果
（下图）

```
ans =

   1.0e+06 *

   0.034899771818395
   0.087209964008468
   0.190988920018819
   0.418330250529287
   0.934046023759115
   2.109010069865914
```

绘制表格进行对比

理论时间(ns)	34900	87210	190989	418330	934046	2109010
实际时间(ns)	37374	82695	190307	424069	930303	2109737
时间误差	6.62%	5.46%	0.36%	1.35%	0.40%	0.03%

考虑到所测量的实际运行时间有±5~10%的误差，且时间误差最大为%6.62，可以认为实际时间与理论时间相同

为了进一步验证所给时间复杂度的计算式，我们采用得到的公式对 n=1024 和 n=2048 进行预测
根据上面的参数得到理论实践计算式为

T(n)= 644.5265*n*logn-1939.4130*n+21466.2753*logn-61184.3992

可得 T(1024)=4791171　T(2048)=10722720　（取整的结果）
测得实际运行时间为 t(1024)=4504990　t(2048)=10598353
误差分别为 6.35%和 1.17%，属于可接受的误差。

因此认为，所给的时间复杂度计算公式是正确的。

四、 测试平台

CPU：i7-6500U

内存：DDR3 1GB

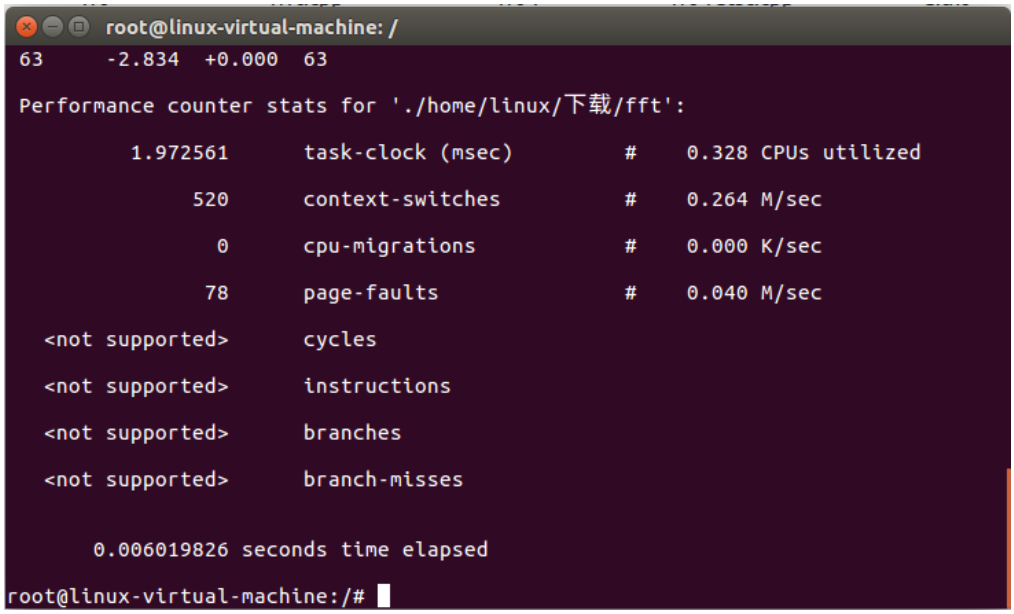
操作系统：ubuntu-16.04.5 LTS

备注：虚拟机环境运行（Vmware Workstation Pro 15.0）

五、 测试记录

对于时间复杂度公式的讨论在第三部分。

对于所给的 C++代码，使用 perf 工具进行追踪并输出



（由于运行环境为虚拟机，因此后面几项数据显示<not supported>）
（这个代码是直接使用原代码进行的测试，也就是 n=30，所以这里的运行时间和前面分析时间复杂度时的数据不同（事实上当 n=30 时，每次 perf 的时间相差甚远，这里随机截了一张图，以显示 perf 工具的监视结果））

六、 分析和结论

从测试记录来看，FFT 程序的执行时间随数据规模增大而增大，且所给出的时间复杂度计算公式符合实际情况，因此 FFT 程序执行时间为

$T(n)= a*n*\log n+\frac{b}{3}*n+\sqrt{2}*c*\log n+d$

根据测试结果，代入参数，得

T(n)= 644.5265*n*logn-1939.4130*n+21466.2753*logn-61184.3992（时间单位:ns）

T(n)=O(nlogn)

七、 思考题

1. 分析 FFT 程序的时间复杂度，得到执行时间相对于数据规模 n 的具体公式

   T(n)= 644.5265*n*logn-1939.4130*n+21466.2753*logn-61184.3992

   （时间单位:ns）

2. 根据上一点中的分析，至少要测试多少不同的 n 来确定执行时间公式中的未知数？

   四个变量中数量级最大的达到了 10^4，因此至少 n=10^4，事实上也正是在 n=10^4 的条件下才求得了误差较小的解

3. 重复 30 次测量然后取平均有什么统计学的依据？

   因为每次运行同样的程序，由于代码内存在函数迭代以及大量的计算，使得运行的时间并不相同。重复 30 次测量然后取平均能够减小极端数据对正常数据的影响，同时又不会损失样本数据。（不过在这个实验里 30 次似乎还不足以使结果稳定下来）