

并行方案测试

王建鑫

2019 年 3 月 1 日

1 测试方案

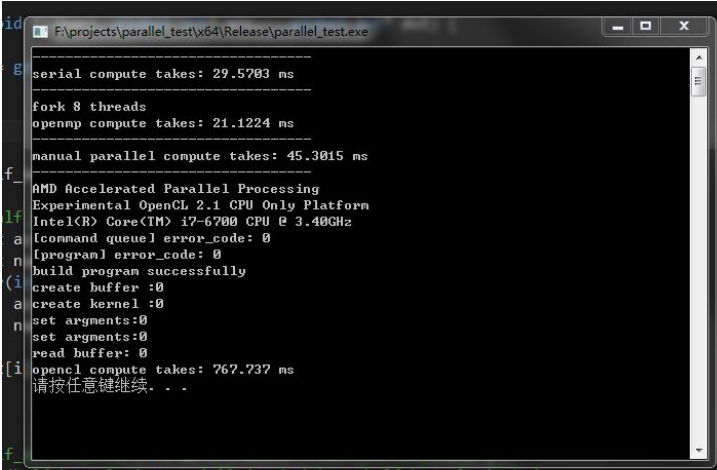
选择最为典型的可以并行计算的操作：均值滤波，作为基准对比多种并行加速方案性能。

并行加速方案有：

- openmp
- 手动并行
- opengl

编写多种并行加速方案代码。不断改变均值滤波的窗口尺寸以增加计算量，对比不同并行方案的性能。

窗口直径为5： 当窗口直径为5像素时，结果如图??。



```
id
g
serial compute takes: 29.5783 ms
-----
fork 8 threads
openmp compute takes: 21.1224 ms
-----
manual parallel compute takes: 45.3815 ms
-----
f
AMD Accelerated Parallel Processing
Experimental OpenCL 2.1 CPU Only Platform
lf
Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
a
[command queue] error_code: 0
n
[program] error_code: 0
n
build program successfully
(i
create buffer :0
a
create kernel :0
n
set arguments:0
n
set arguments:0
n
read buffer: 0
[i
opencl compute takes: 767.737 ms
请按任意键继续. . .
f
(half_kernel_size <= j && j < height - half_kernel_size) {
```

图 1: 窗口直径为5

其中串行计算时间为：29ms；openmp通过8线程并行，计算时间为：21ms；手动编写并行代码花费：45ms；使用OpenCL花费：767ms。

窗口直径为21： 当窗口直径为21像素时，结果如图??。

其中串行计算时间为：232ms；openmp通过8线程并行，计算时间为：102ms；手动编写并行代码花费：103ms；使用OpenCL花费：703ms。

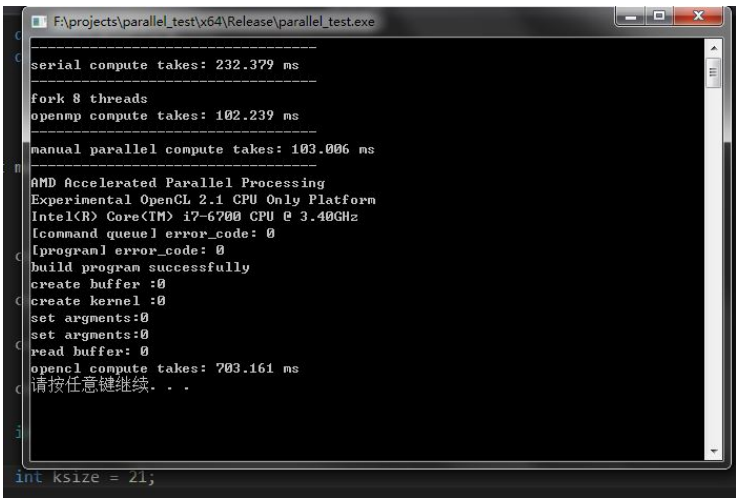


图 2: 窗口直径为21

窗口直径为51： 当窗口直径为51像素时，结果如图??。

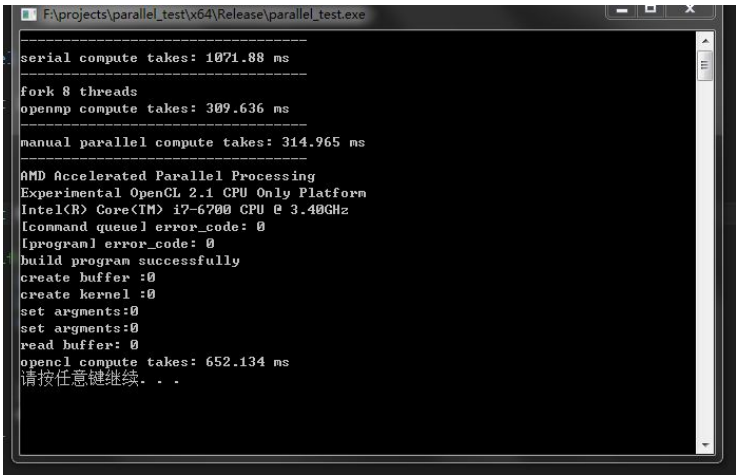
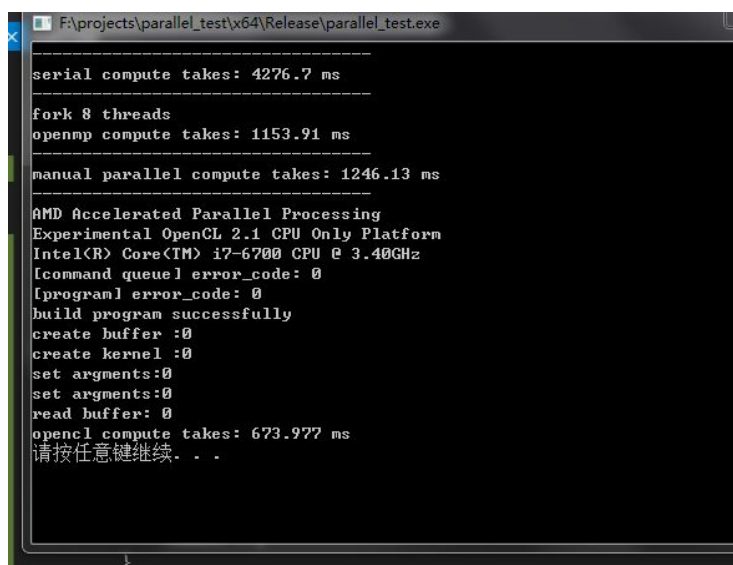


图 3: 窗口直径为51

其中串行计算时间为：1071ms；openmp通过8线程并行，计算时间为：309ms；手动编写并行代码花费：315ms；使用OpenCL花费：652ms。

窗口直径为101： 当窗口直径为101像素时，结果如图??。

其中串行计算时间为：4276ms；openmp通过8线程并行，计算时间为：

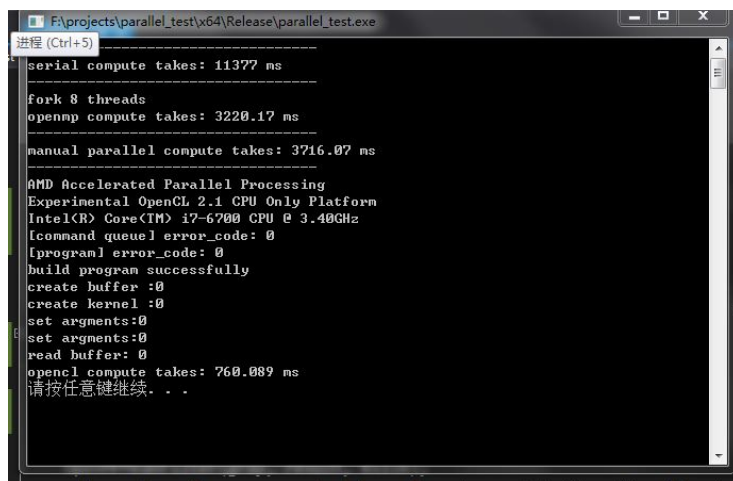


```
-----
serial compute takes: 4276.7 ms
-----
fork 8 threads
openmp compute takes: 1153.91 ms
-----
manual parallel compute takes: 1246.13 ms
-----
AMD Accelerated Parallel Processing
Experimental OpenCL 2.1 CPU Only Platform
Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
[command queue] error_code: 0
[program] error_code: 0
build program successfully
create buffer :0
create kernel :0
set arguments:0
set arguments:0
read buffer: 0
opengl compute takes: 673.977 ms
请按任意键继续. . .
```

图 4: 窗口直径为101

1153ms; 手动编写并行代码花费: 1246ms; 使用OpenCL花费: 674ms。

窗口直径为151: 当窗口直径为151像素时, 结果如图??。



```
-----
serial compute takes: 11377 ms
-----
fork 8 threads
openmp compute takes: 3220.17 ms
-----
manual parallel compute takes: 3716.07 ms
-----
AMD Accelerated Parallel Processing
Experimental OpenCL 2.1 CPU Only Platform
Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
[command queue] error_code: 0
[program] error_code: 0
build program successfully
create buffer :0
create kernel :0
set arguments:0
set arguments:0
read buffer: 0
opengl compute takes: 760.089 ms
请按任意键继续. . .
```

图 5: 窗口直径为151

其中串行计算时间为: 11377ms; openmp通过8线程并行, 计算时

间为：3220ms；手动编写并行代码花费：3716ms；使用OpenCL花费：760ms。

窗口直径为201： 当窗口直径为201像素时，结果如图??。

图 6: 窗口直径为201

其中串行计算时间为：26447ms；openmp通过8线程并行，计算时间为：5249ms；手动编写并行代码花费：6200ms；使用OpenCL花费：857ms。

汇总 将上述时间花费汇总至一个图??中。

当计算任务量逐渐增加，串行计算消耗时间明显增加。

而OpenMP则是由自动优化，优化效果明显，但其瓶颈是CPU的计算能力，当任务量增加到一定程度，其运算时间也不可避免的增加。

手动并行编程则是将任务手动的分成多线程，其瓶颈也是CPU的计算能力。当任务量逐渐增加，这种方案和OpenMP效果差不多。

OpenCL在任务量较小时，速度缓慢。当任务量巨大时，其优势明显。这种方案的瓶颈是内存读写速度，由于内存读写速度较慢，在任务量较少时严重拖慢计算速度；当计算任务巨大时，将任务分配至异构平台，可以大大减少CPU负担，减少计算时间。所以在上述各个方案计算时间明显增加时，这种方案计算时间一直保持平稳。

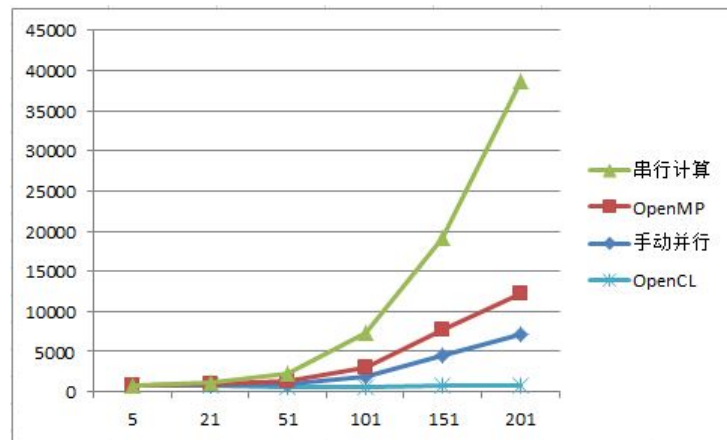


图 7: 汇总