Stream 测内存带宽

邹永浩

2019211168

单线程测试

直接下载 `stream.c`，不加任何参数直接编译运行的结果如下

```
zyh@WIN-HOME:~/MeasuringComputerPerformance/hw12$ gcc stream.c
zyh@WIN-HOME:~/MeasuringComputerPerformance/hw12$ ./a.out
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 10000000 (elements), Offset = 0 (elements)
Memory per array = 76.3 MiB (= 0.1 GiB).
Total memory required = 228.9 MiB (= 0.2 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 25686 microseconds.
   (= 25686 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           7831.2      0.021579     0.020431     0.026837
Scale:          7561.8      0.021742     0.021159     0.023823
Add:           11119.9      0.022196     0.021583     0.023839
Triad:         10367.6      0.023479     0.023149     0.024280
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

可以看到 Copy 速率为 `7.8GB/s` 左右，与之前作业5测试结果一致.

多线程测试

`Stream` 支持 `OpenMP`、`pthreads` 和 `MPI` 多种模式，我们使用 `OpenMP` 进行测试。

```
zyh@WIN-HOME:~/MeasuringComputerPerformance/hw12$ gcc stream.c -fopenmp -D_OPENMP
<command-line>:0:0: warning: "_OPENMP" redefined
<built-in>: note: this is the location of the previous definition
zyh@WIN-HOME:~/MeasuringComputerPerformance/hw12$ export OMP_NUM_THREADS=8
zyh@WIN-HOME:~/MeasuringComputerPerformance/hw12$ ./a.out
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 10000000 (elements), Offset = 0 (elements)
Memory per array = 76.3 MiB (= 0.1 GiB).
Total memory required = 228.9 MiB (= 0.2 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Number of Threads requested = 8
Number of Threads counted = 8
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 9283 microseconds.
   (= 9283 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           14151.7     0.012240     0.011306     0.013216
Scale:          14325.1     0.012164     0.011169     0.013047
Add:            15507.9     0.016029     0.015476     0.017753
Triad:          15473.1     0.015830     0.015511     0.016258
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

可以看到，编译选项中加入 `-fopenmp -D_OPENMP`，并且设置 `OMP_NUM_THREADS` 环境变量后，运行结果为 `14GB/s`，看来单线程确实无法发挥内存的全部速度，如果是想把内存的效率发挥到最大，还是需要编写多线程程序，充分利用多核。

参考文献

http://www.cs.virginia.edu/stream/ref.html