

Lab 4 串并行矩阵相乘的比较

1.实验要求

- 编写矩阵随机生成类 MatrixGenerator 类，随机生成任意大小的矩阵，矩阵单元使用 double 存储。
- 使用串行方式实现矩阵乘法。
- 使用多线程方式实现矩阵乘法。
- 比较串行和并行两种方式使用的时间，利用第三次使用中使用过的 jvm状态查看命令，分析产生时间差异的原因是什么。

2.实验结果和代码分析

```
//通过断言的方式判断运算正确
double[][] parallelMatrix= myMatrix.parallelCompute();
double[][] serialMatrix = myMatrix.serialCompute();
assertArrayEquals(parallelMatrix,serialMatrix);
```

1.两个线程-矩阵规模在10以下

```
第一个矩阵-----
规模 7 * 6
第二个矩阵-----
规模 6 * 7
并行目标矩阵 -> -----
并行计算的执行时间是 -> 0ms
串行目标矩阵 -> -----
串行计算的执行时间是 -> 0ms
```

2.两个线程-矩阵规模在100以下

```
<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_102\bin\javaw.exe (2020年
第一个矩阵-----
规模 87 * 16
第二个矩阵-----
规模 16 * 87
并行目标矩阵 -> -----
并行计算的执行时间是 -> 3.0ms
串行目标矩阵 -> -----
串行计算的执行时间是 -> 0.0ms
```

可以看见在这个范围内串行计算性能优越

3.两个线程-矩阵规模在1000以下

```
<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_102\bin\javaw.exe
第一个矩阵 -----
规模 734 * 998
第二个矩阵 -----
规模 998 * 734
并行目标矩阵 - > -----
并行计算的执行时间是 -> 458.0ms
串行目标矩阵 - > -----
串行计算的执行时间是 -> 829.0ms
```

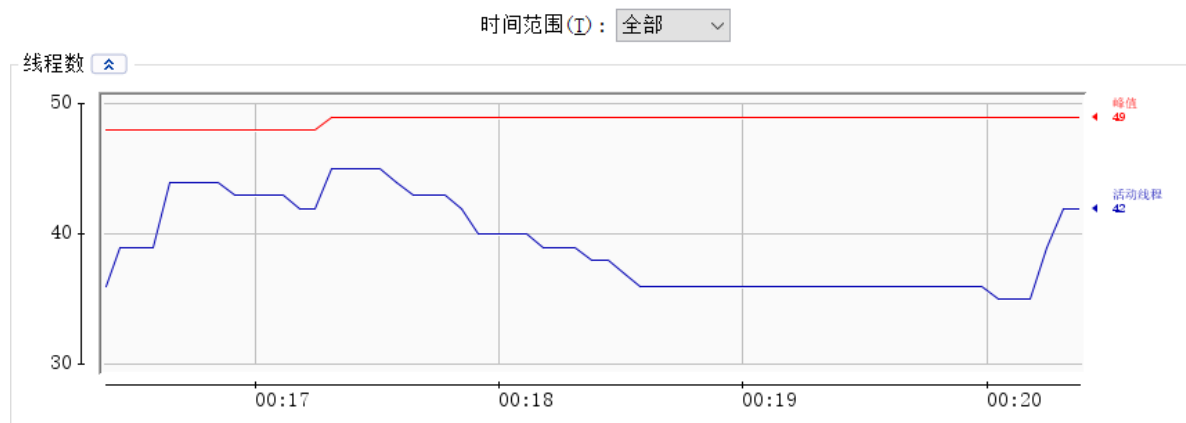
并行计算和串行计算消耗的时间拉开了明显的差距，并行计算使用的时间是串行的0.5倍。

4.四个线程-矩阵规模在1000以下

```
<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_102\bin\javaw.exe (2
第一个矩阵 -----
规模 370 * 916
第二个矩阵 -----
规模 916 * 370
并行目标矩阵 - > -----
并行计算的执行时间是 -> 46.0ms
串行目标矩阵 - > -----
串行计算的执行时间是 -> 172.0ms
```

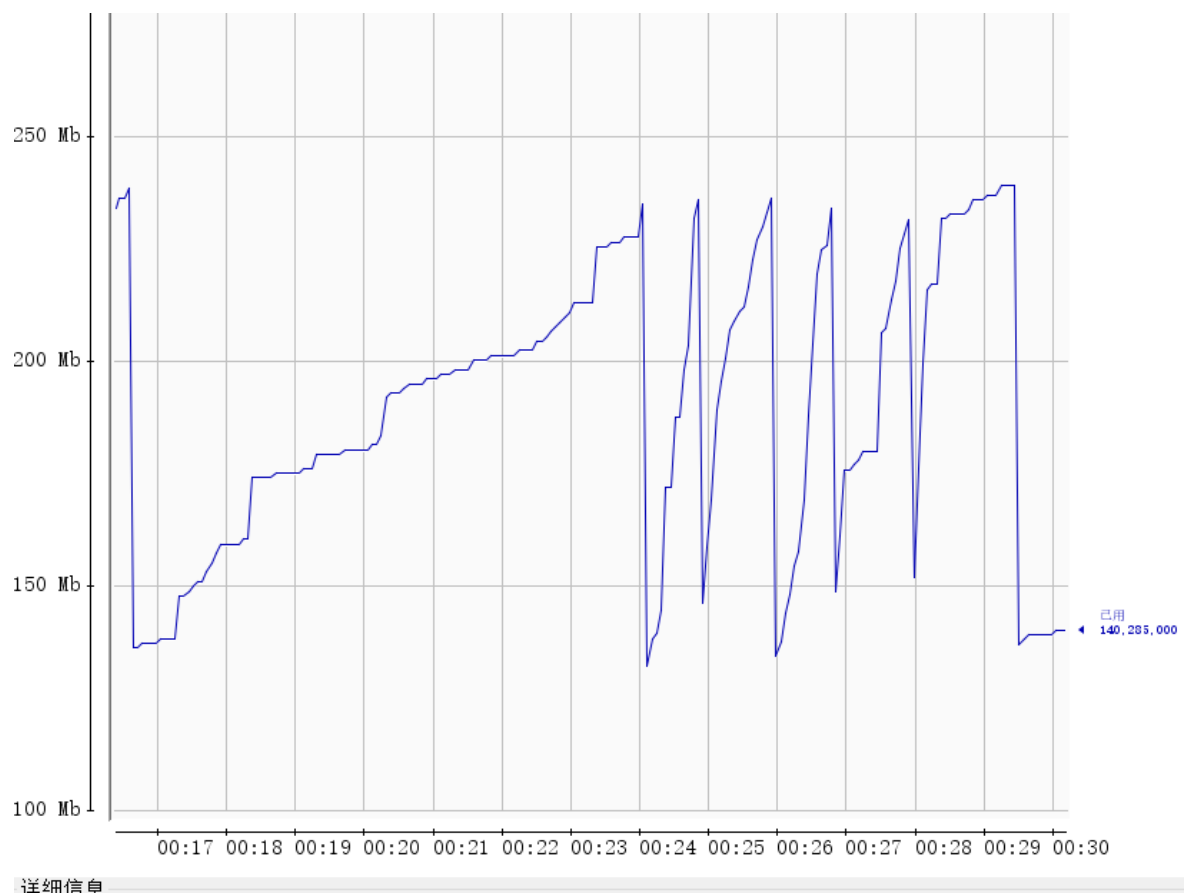
多线程并行体现了压倒性的优势。

通过gconsole窥测线程



程序run后 出现了上升趋势

内存在每一次执行完后都会发生很大的变化



最后开了八个线程执行2000规模的矩阵乘法

```
Console
<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_102\bin\javaw.exe (2020年
第一个矩阵-----
规模 1896 * 488
第二个矩阵-----
规模 488 * 1896
并行目标矩阵- > -----
并行计算的执行时间是- > 923.0ms
串行目标矩阵- > -----
串行计算的执行时间是- > 3639.0ms
```



上图为线程数量图