

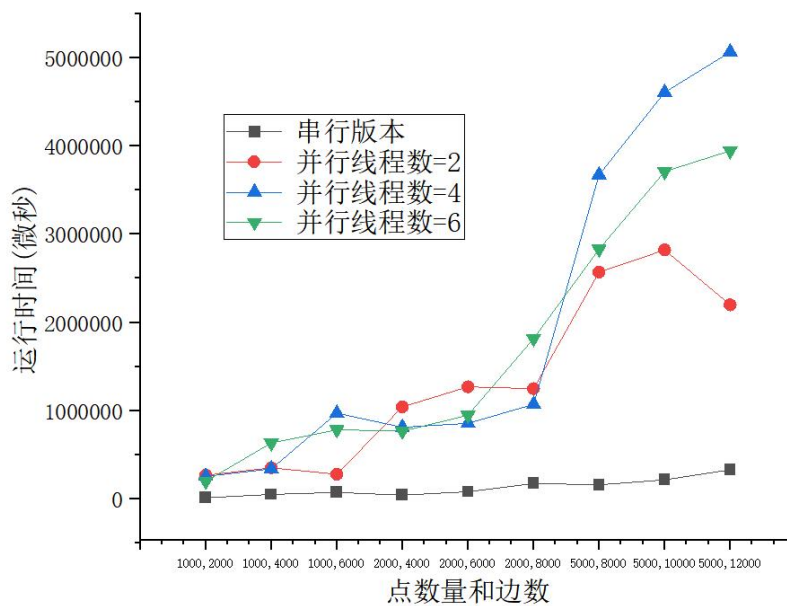
HW14-kruskal 算法并行性能分析实验报告

本实验要求实现串行与并行两种 kruskal 算法，并比较在不同并行线程数时其性能的差异。一下是本次实验的测试结果图表：

1.表格：

	单线程(串行算法)	并行, n=2	并行, n=4	并行, n=6
v=1000, e=2000	10196	265911	253200	197583
v=1000, e=4000	48961	350036	337873	632547
v=1000, e=6000	70936	278121	970008	781971
v=2000, e=4000	40092	1041920	810802	766429
v=2000, e=6000	78247	1269180	854204	945709
v=2000, e=8000	171569	1248680	1069890	1818240
v=5000, e=8000	156559	2571470	3675170	2831030
v=5000, e=10000	213739	2823130	4610220	3711920
v=5000, e=12000	328566	2199540	5067930	3948800
注：本表格中的时间数据均以微秒为单位				

2. 折线图：



3. 结果分析：

从上图可以看出，利用并行化来改进 kruskal 算法并不会提高性能，反而会使性能出现明显的下降，个人分析认为这可能是由于主线程与子线程之间频繁的传递消息并进行上锁解锁等操作会明显的拖延整体的性能。另外，随着线程数量的增多，只对并行版本而言其性能也没有很好地提升，这可能是由于因为需要共享信号以及 lock，所以随着线程的增多线程之间的竞争会加剧，从而产生较大的 overhead，从而使得性能甚至会出现下降的情况。