



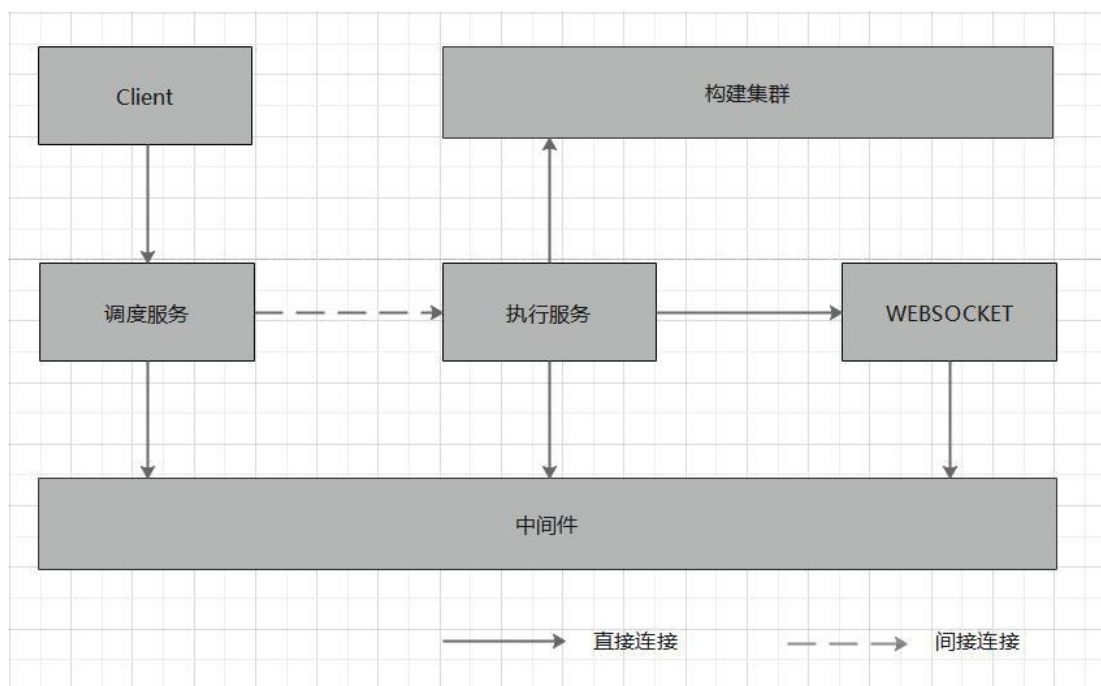
执行引擎性能测试报告

上海精鲲计算机 信息科技有限公司	撰写人
	JKSTACK
	执行引擎性能测试报告

目录

1. 测试环境部署	4
2. 测试项目	4
3. 测试环境和工具	5
3.1：测试环境	5
3.2：测试工具	5
4. 术语解释	6
5. 测试方案	7
5.1：基准测试	7
5.2：压力测试	7
5.3：压力负载均衡测试（稳定性测试）	7
6. 测试场景及测试结果分析	8
6.1：执行引擎事务处理及分析	8
6.2：构建任务事务处理数据对比及分析	15
7. 测试总结及分析建议	17
7.1：测试总结及分析建议	17
7.2：配置推荐	17

1. 测试环境部署



注：本次测试采均为单节点

2. 测试项目

2.1 执行引擎

2.2 执行引擎-构建集群

3. 测试环境和工具

3.1：测试环境

环境	操作系统	CPU	内存	磁盘	部署方式
中间件	Linux	8 核	16G	300G	容器化
构建节点	Linux	4/8 核	8/16G	600G	容器化
执行引擎	Linux	4/8 核	8/16G	300G	容器化
客户端	Linux	8 核	16G	300G	容器化

3.2：测试工具

- Postman 调试
- Jmeter 用户并发数
- Docker 起多台OS
- python 脚本

3.3：其他

日志级别：ERROR

4. 术语解释

术语	解释
TPS (每秒处理事物数)	在每秒时间内系统可处理完毕的事务数。
异常率	本次测试中出现的错误率，即响应失败数/总请求数
并发数	测试时同时系统发出事务请求的数量，并发线程数用以模拟同时与系统建立连接的用户。
Received KB/src	每秒从服务器端接收到的数据量
Sent KB/src	每秒从客户端发送的请求的数量
QPS	每秒的响应请求数，也即是最大吞吐能力

5. 测试方案

5.1: 基准测试

- 在小并发下，看服务的处理情况，作为后续对比基础。

5.2: 压力测试

- 高并发查看服务器的吞吐量

5.3: 压力负载均衡测试（稳定性测试）

- 对软件多次测试，长时间运行，是否正常运行
- 软件长时间对日常的用户数进行操作运行，观察系统内存占用率是否越来越大，可用内存是否减少，内存是否溢出，饱和运算内存是否占用过大、是否溢出

6. 测试场景及测试结果分析

6.1：执行引擎事务处理及分析

6.1.1 测试结果

* 4C8G 配置:

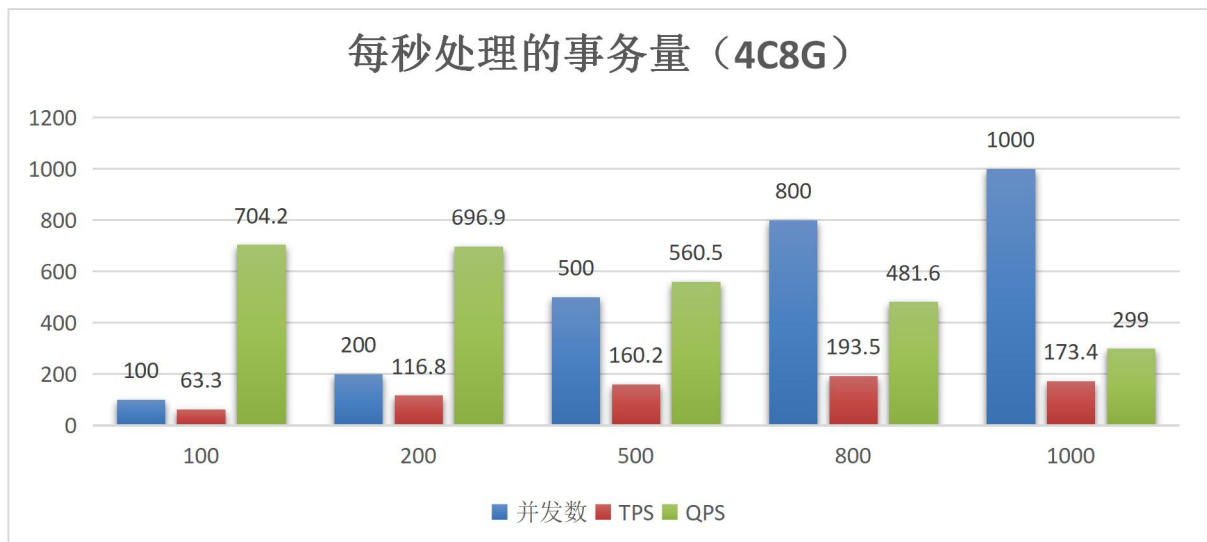
- ⇒ 根据基准测试和压力测试，可以看出执行引擎调度服务异步最高可支持 800/s 吞吐量 (TPS) 达到最大值 193.5/sec 事务成功率 100%;
- ⇒ 系统稳定性: 每隔 1 秒增加 100/s, 执行 20 次, 共计 2000 样本,均可保证系统正常处理请求。

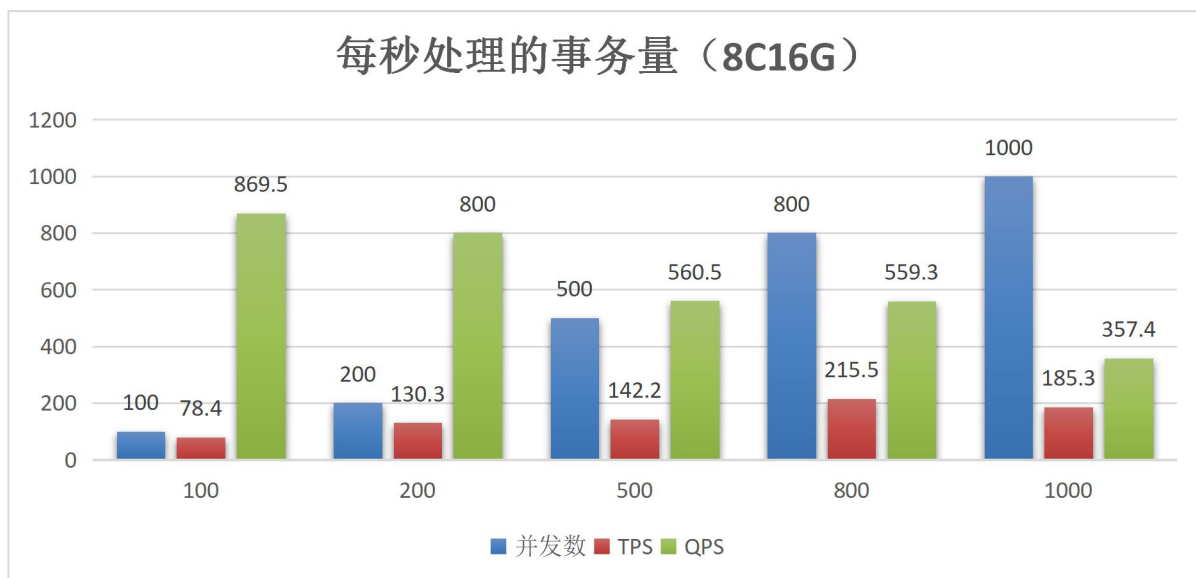
* 8C16G 配置:

- ⇒ 根据基准测试和压力测试，执行引擎调度服务最高可支持1000/s，事务成功率 100% 此配置下TPS和QPS有所提升;
- ⇒ 系统稳定性测试: 每隔 1 秒增加 200/s, 共计 4000 样本,均可保证系统正常处理请求。

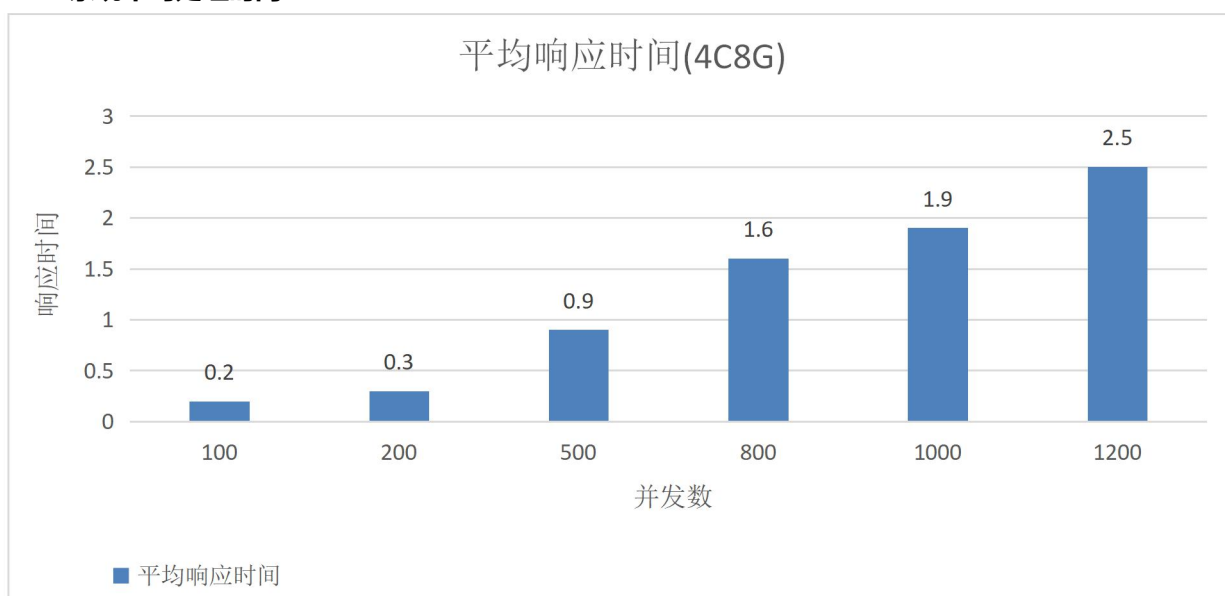
6.1.2 测试数据如下

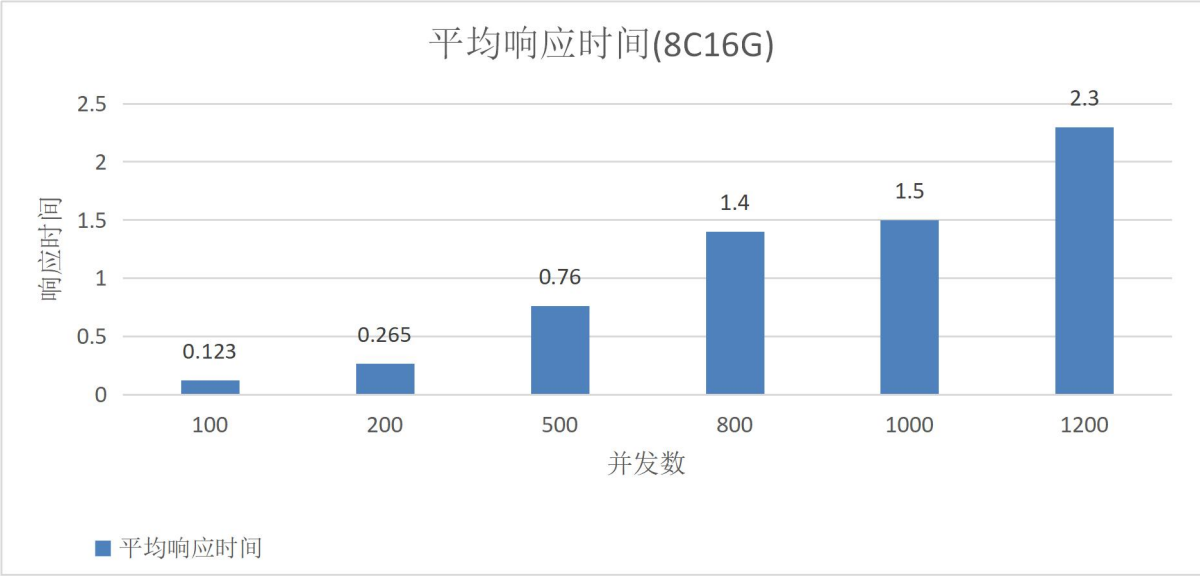
* 系统每秒处理的事务量-调度服务





* 系统平均处理时间





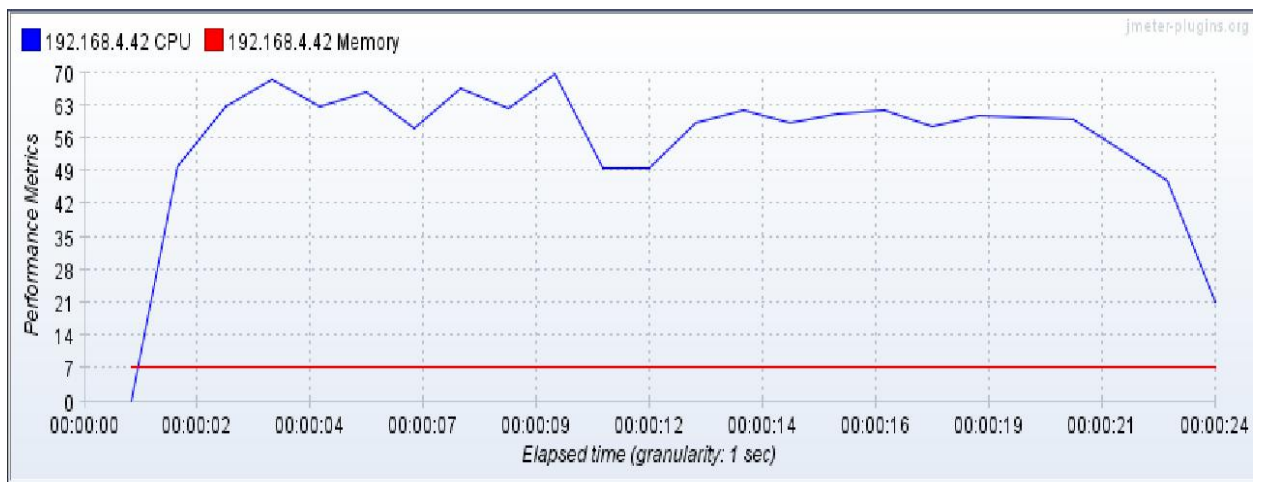
8C16G 平均响应时间

说明：系统在越高的压力下，系统平均处理请求的时间也长。4C8G和8C16G效果不明显

* 服务器消耗性能数据对比分析



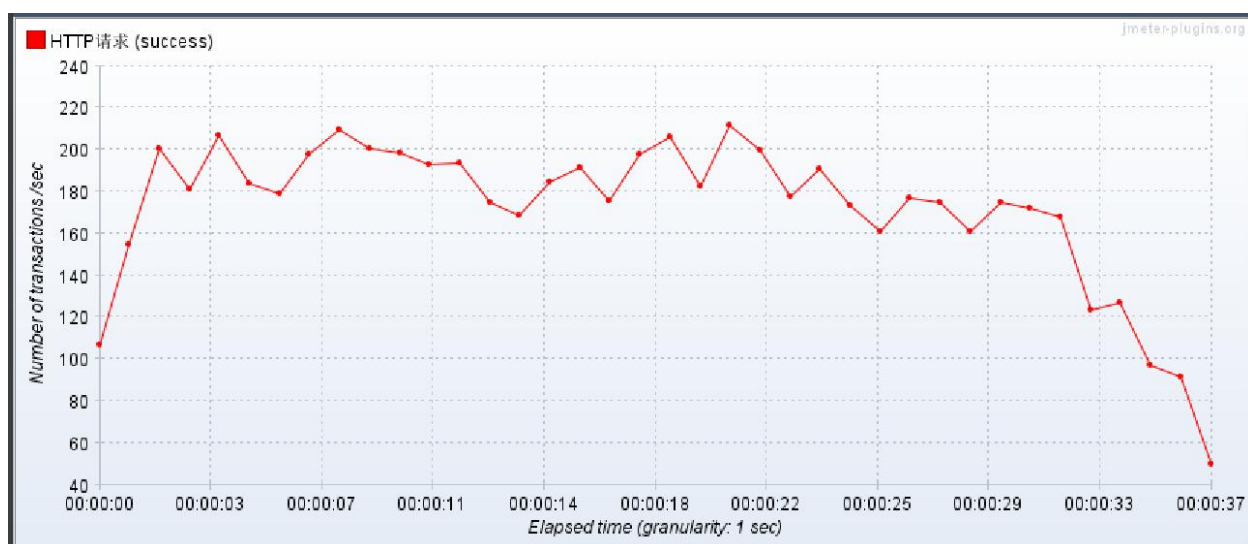
4C8G 配置 1000/s 性能图



8C16G 配置 1000/s 性能图

说明：在 1000/s 下，4C8G 消耗约占 cpu 90%，大概持续 20 秒，服务消耗 cpu 资源高。而 8C16G 消耗约占 cpu 70%，大概持续 20 秒，内存消耗也较低。此配置提升对并发数的最大值有较大提升。

* 调度服务—系统吞吐量(压力负载测试)



4C8G 配置 1000/s 吞吐量



8C16G 配置 1000/s 吞吐量

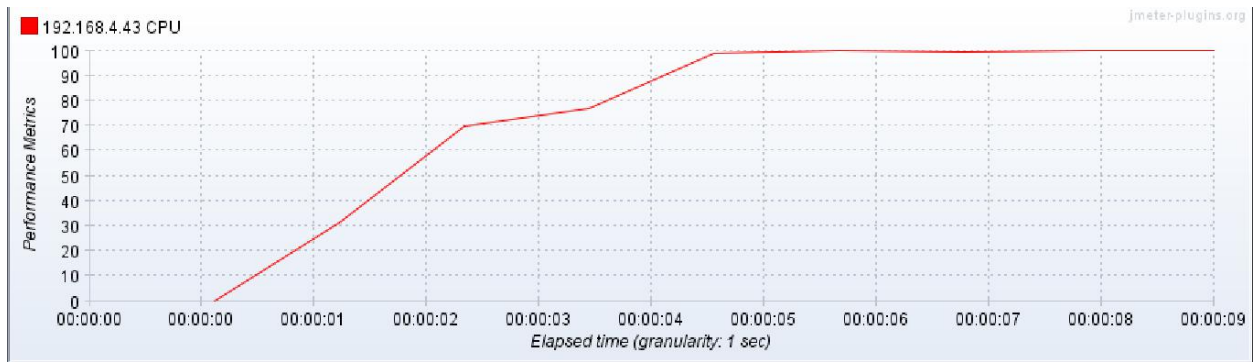
说明：1000/s 下，4C8G 的吞吐量最大，212/sec。

1000/s 下，8C16G 的吞吐量最大，246/sec。

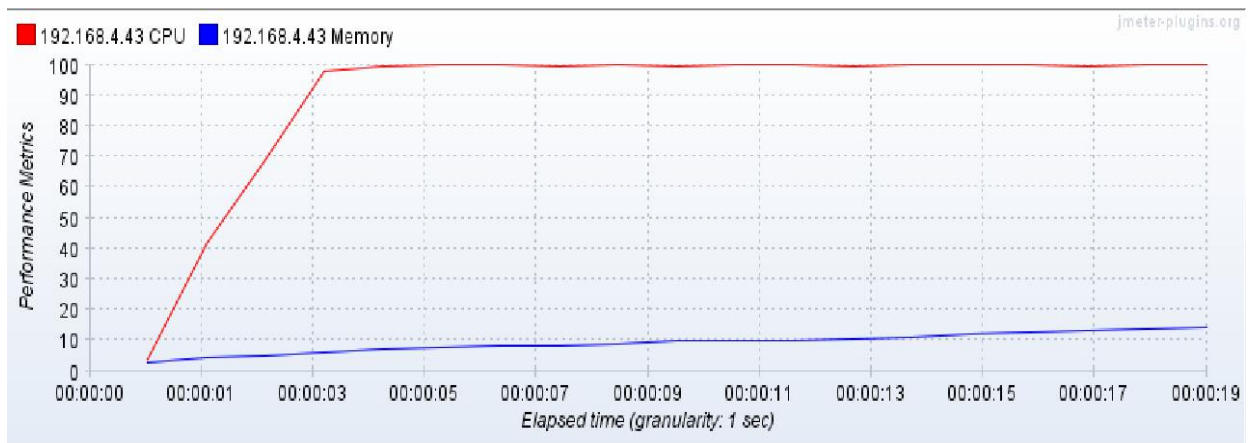
* 事务处理数据对比分析:

说明：4C8G cpu 出现短暂的满载情况，内存正常

*构建集群-服务(压力负载测试)

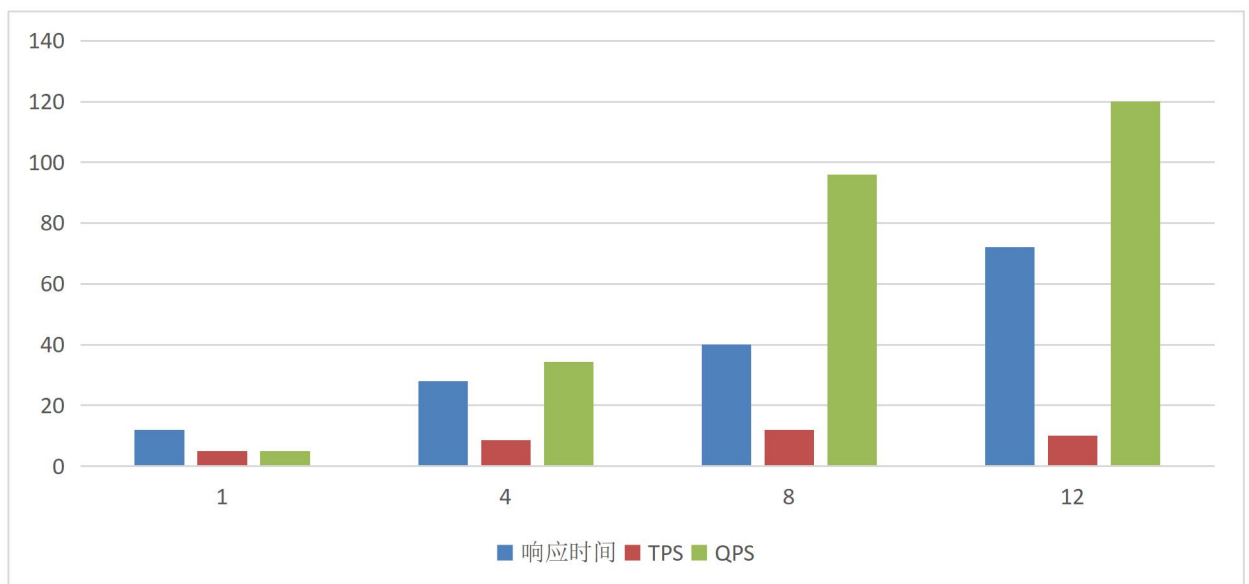


4C8G 配置 8/s 性能图



8C16G 配置 8/s 性能图

说明: 8/s 下, 4C8G 和8C16G 的配置下 CPU都是满载状态。



说明：4C8G配置下：每次并发8个构建任务的时候，构建集群的时候，cpu使用率是90%以上。完成时间为：2分钟左右。

8C16G配置下：每次并发8个构建任务的时候，构建集群的时候，cpu使用率是90%以上。完成时间为：1分钟左右。

7. 测试总结及分析建议

7.1：测试总结及分析建议

✱ 4C8G 调度任务的服务器再并发1000/s，吞吐量开始下降，构建任务再并发16次正常，高负载情况会出现任务失败的线程。处理每个构建任务耗时12秒左右。

✱ 8C16G 调度任务的服务器再并发1000/s，吞吐量开始下降，构建任务再并发30次正常，高负载情况会出现任务失败的线程。处理每个构建任务耗时12秒左右。

7.2：配置推荐

服务	并发数	最小配置	最佳配置
执行引擎-调度	<=800	4C8G	4C8G
	<=1000	4C8G	8C16G
执行引擎-执行	<=1000	4C8G	8C16G
执行引擎-构建	<=8	4C8G	4C8G
	<=16	4C8G	8C16G
	<=32	8C16G	16C32G