**Ontology VBFT共识算法性能测试报告**

TPS（Transaction per Second）每秒交易处理笔数，反映了系统在同一时间内能处理业务的最大能力，是区块链的核心性能指标之一。Apache JMeter是Apache组织开发的基于Java的压力测试工具， 本文使用JMeter对Ontology 0.8.2 版本做压测，将测试过程及结果记录如下。

1. 测试环境

每个节点为微软云虚拟机，共7个共识节点，硬件配置如下：

1. CPU：单节点8CPU，具体信息如下：
   1. 名称：Intel(R) Xeon(R) CPU E5-2673 v3
   2. 主频：2.40GHz
   3. 缓存大小：30720 KB
   4. 核数：8
2. 内存：单节点27G
3. 硬盘：固态硬盘，大小50G，读写速率限制不超过25MB/s

软件配置如下：

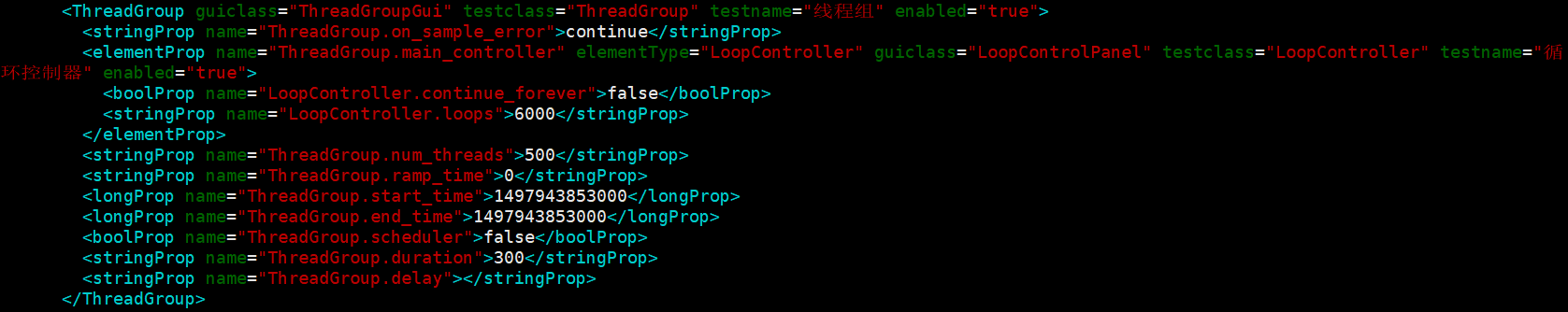
1. 系统信息：ubuntu 16.04.4 LTS
2. Go环境：go1.9.3 linux/amd64
3. Ontology参数

Ontology版本为0.8.2

启动命令：

./ontology --maxtxinblock 120000 --gaslimit 0 --rest --localrpc --disableeventlog --loglevel 2

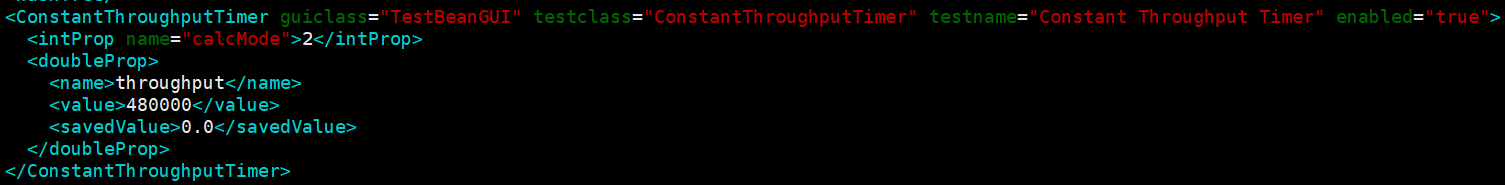
1. 测试步骤
   * 1. 使用go-sdk构造一批不同的ONT转账交易，确保其hash不一样，每笔交易的转账数额为1；
     2. 启动ontology测试网络，共7个节点；
     3. 查询交易发送的目标账户的余额，并记录；
     4. 使用JMeter将这批交易发送到测试网络上，配置500个线程发送，设置固定吞吐量定时器控制发送速率；记录开始发送交易时间。
     5. 查看节点日志，通过log中numtx观察落账交易数量，出现第一个非空块时记录时间，发送完毕后，连续出现三个以上的空块时，可认为交易已经处理完毕，取最后一个非空块的时间作为落账结束时间；
     6. 查询交易发送的目标账户的余额，并记录；
     7. 计算余额差值，除以测试时间，即可得TPS。
2. Jmeter配置
   * + 1. 线程数与发送次数配置：



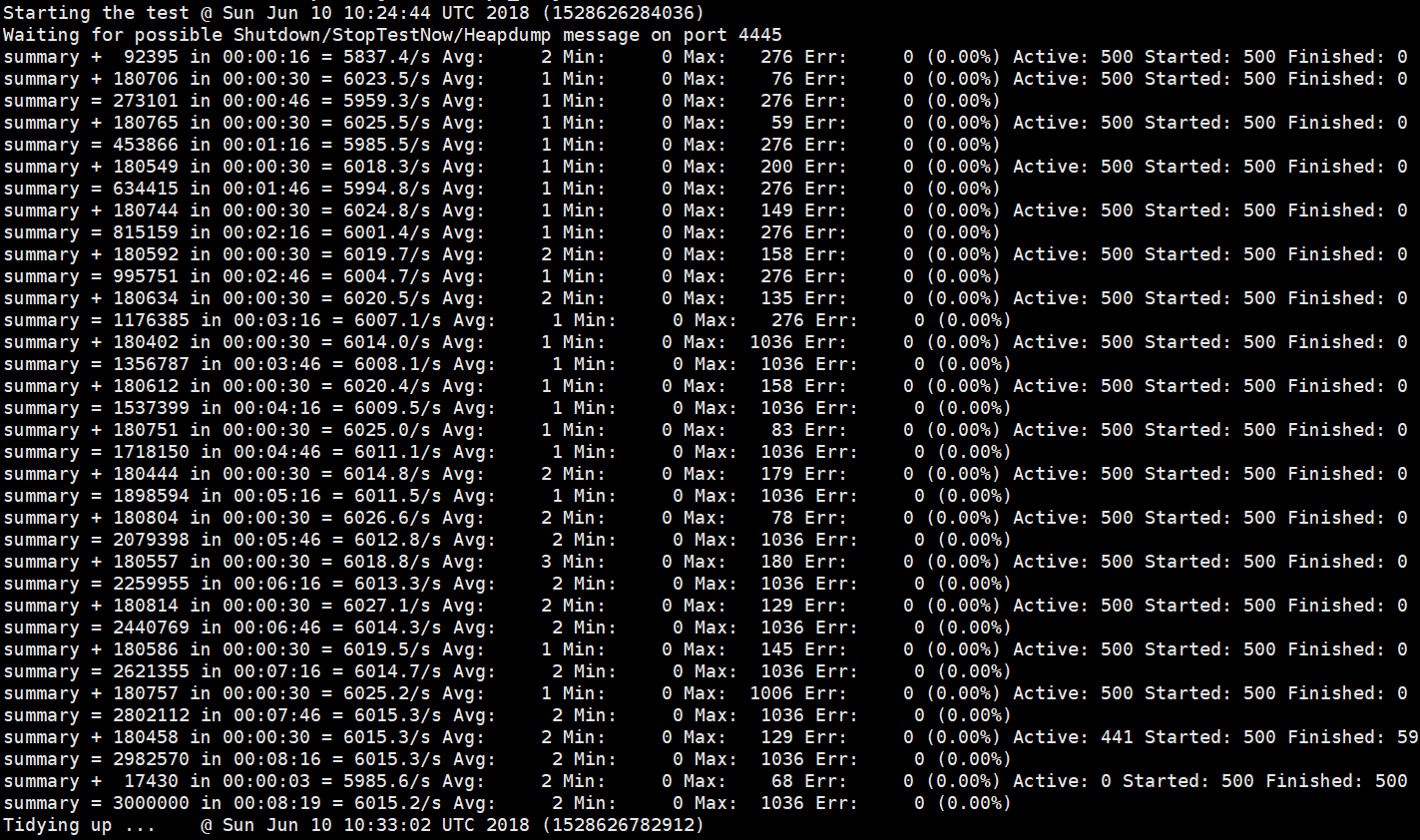
LoopController.loops=6000，ThreadGroup.num\_threads=500；前者代表一个线程发送的交易的次数，后者代表开启的线程数，二者相乘得出的值为发送的交易数，此处为3,000,000。

* + - 1. 发送速率配置：

使用固定吞吐量定时器配置交易发送速率，此处为每分钟360,000，即每秒发送6,000笔交易，见下图：

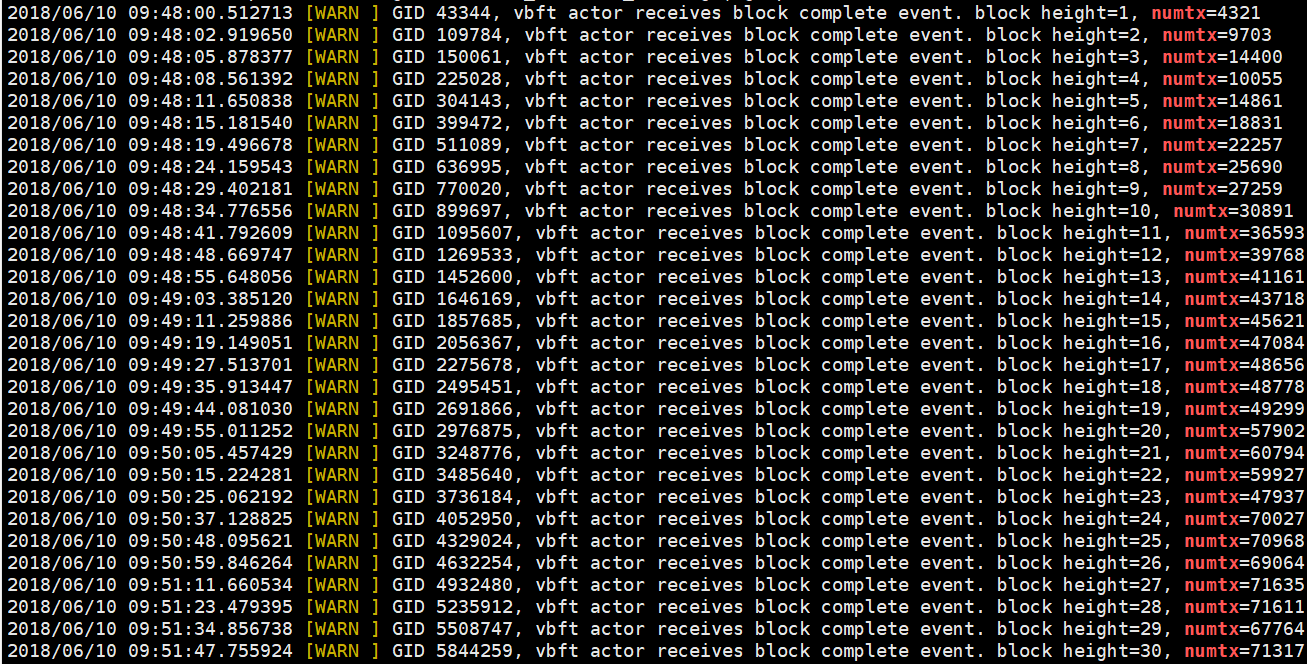


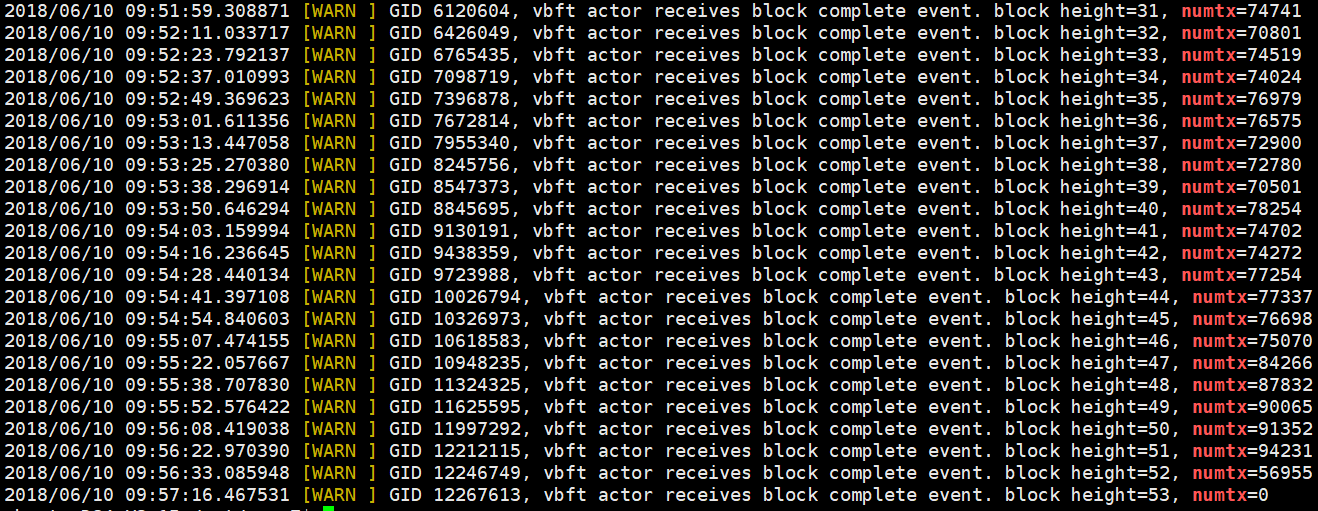
1. 交易发送情况



可以看出共发送了3,000,000笔交易，耗时00:08:19，即499秒，则交易发送速度为6012笔/s

1. 出块情况





1. 测试结果

按照上文所述步骤，进行十次测试，取十次结果的平均值，最终结果如下：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 交易数 | 发送速率 | 发送时间 | 落账成功率 | 块数 | 处理时间 | TPS | 峰值 |
| 300万 | 6000/s | 500s | 99.1% | 40 | 562s | 5341 | 5536 |

注1：TPS = 成功交易数/（完成落账时间- 开始发送交易时间）

注2：峰值即是系统稳定运行所能达到的最大TPS，算法为取落账过程中中间时段的一到两分钟之内的落账笔数，除以落账时间算得。

1. 性能分析

目前用7个节点测试，TPS达到了5300以上。

测试过程中，使用不同的发送速率，不同的交易量进行测试，测试结果TPS都达到了5000以上。最终的测试结果，也就是VBFT的峰值TPS，超过了5500，达到5536左右。

另外，本次测试发现如下问题：

* + - 1. 出块情况不稳定，多次测试过程中发现，块大小（包含的交易数）、出块间隔均不稳定；
      2. 交易量、交易发送速度越高，测出的TPS越低，所以上述测试结果为一个平均值；
      3. 观察log发现“[ERROR]server 7 verify proposal blk from 1 failed, blk 23, txs 120000, err: duplicated transaction detected”出现的较为频繁；