DTCC

数 / 造 / 未 / 来

第十二届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2021

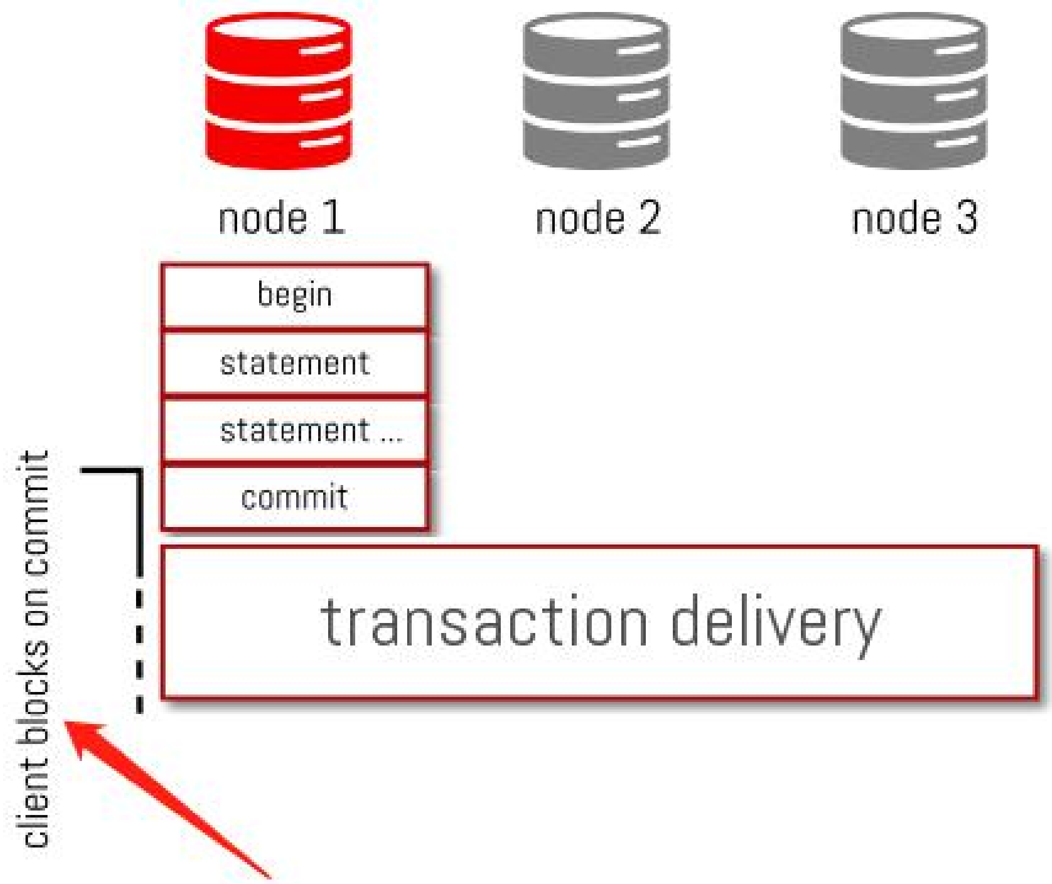
2021 年 10 月 18 日 - 20 日 | 北京国际会议中心

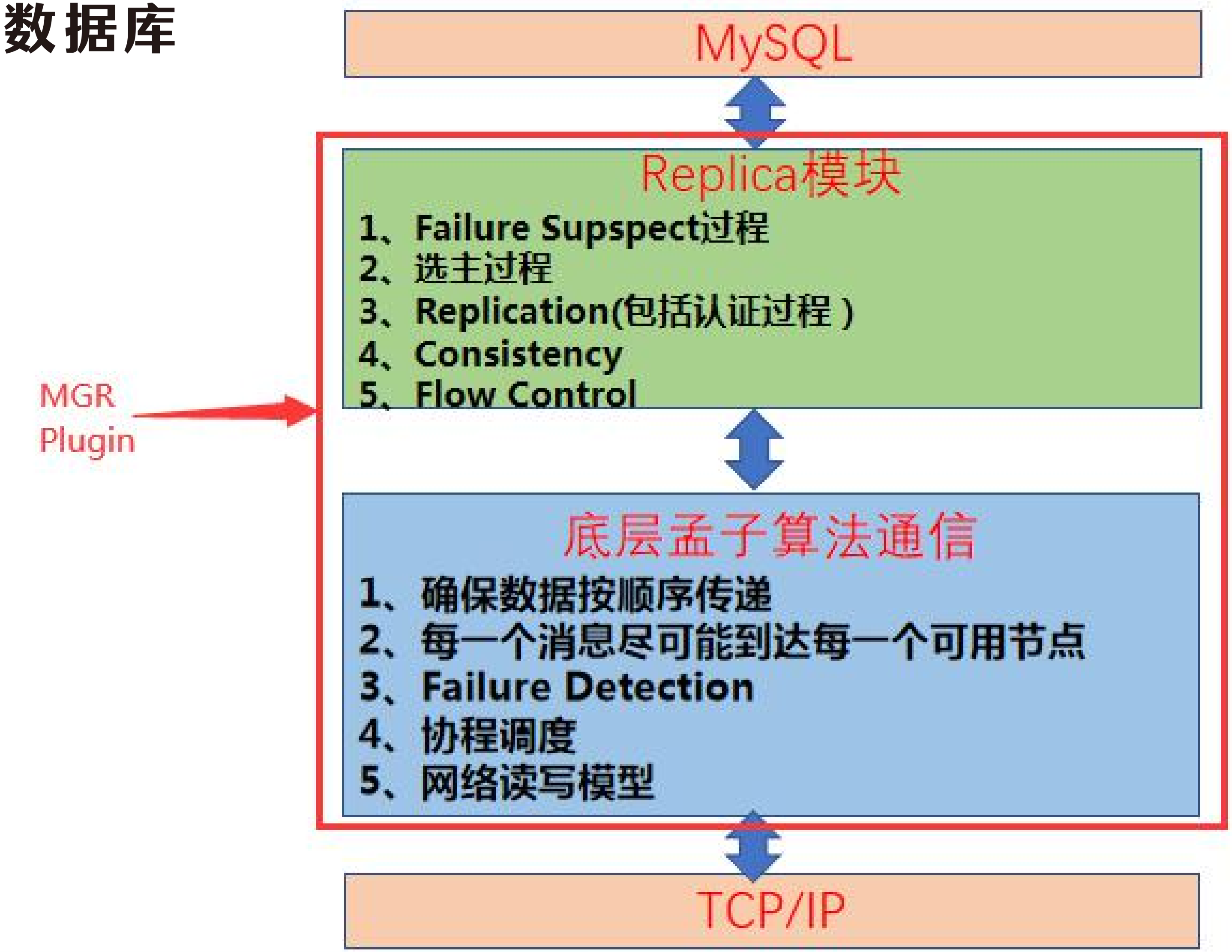
IT168.com ChinaUnix.net ITPUB

MySQL高可用组件MGR之深度分析

万里数据库
王斌

MySQL高可用组件MGR


node 1
node 2
node 3
begin
statement
statement ...
commit
transaction delivery
client blocks on commit



MySQL

Replica模块

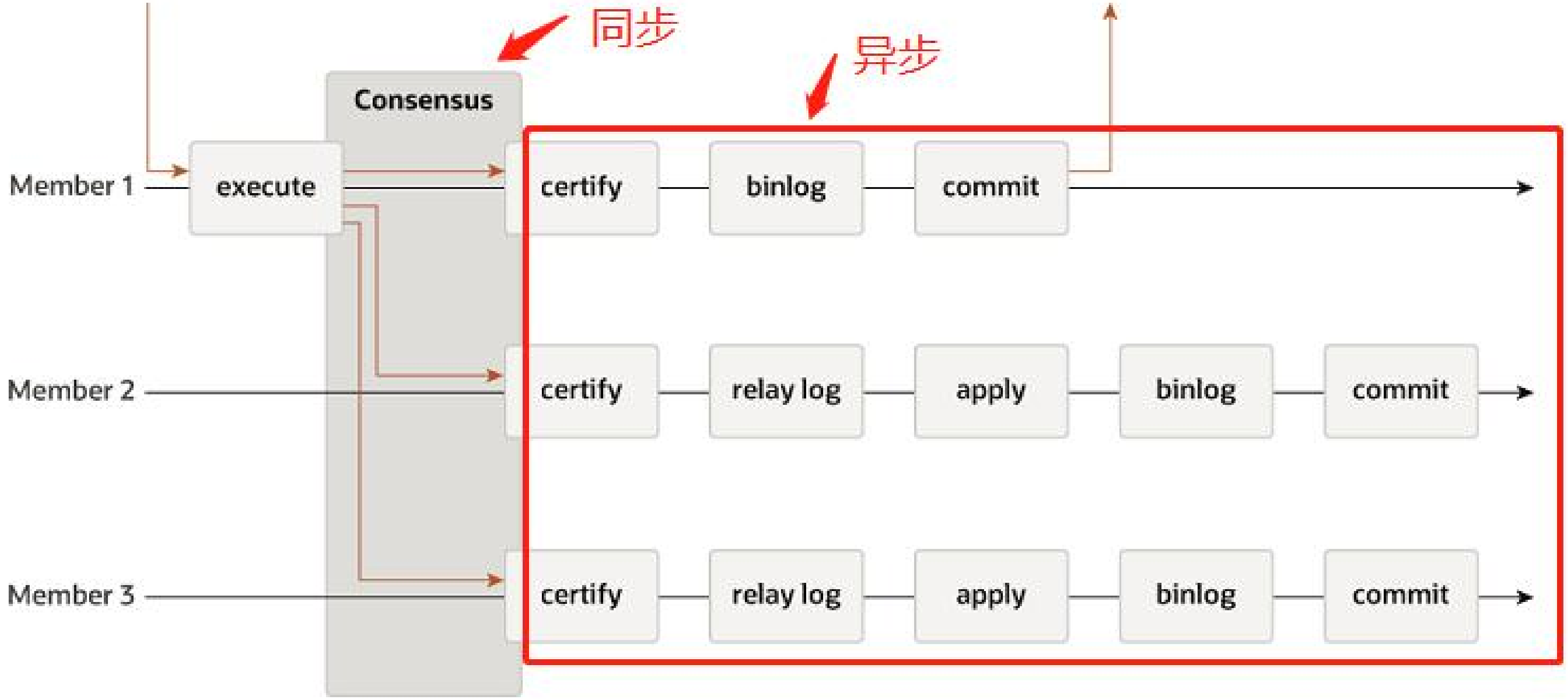
1、Failure Supspect过程
2、选主过程
3、Replication(包括认证过程)
4、Consistency
5、Flow Control

MGR Plugin

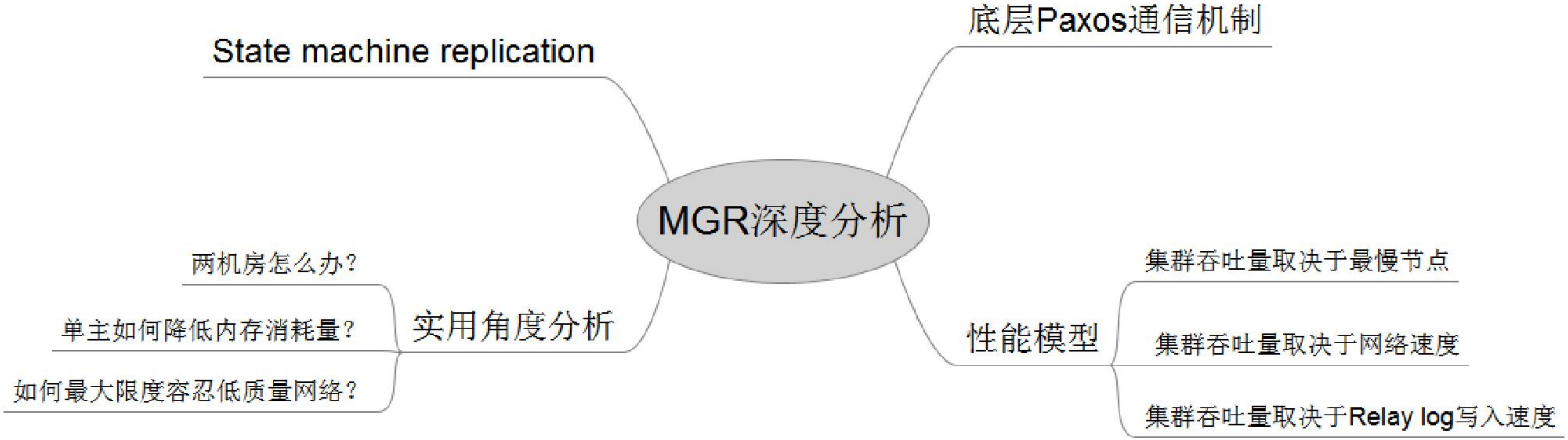
底层孟子算法通信

1、确保数据按顺序传递
2、每一个消息尽可能到达每一个可用节点
3、Failure Detection
4、协程调度
5、网络读写模型

TCP/IP


同步
异步
Consensus
Member 1
execute
certify
binlog
commit
Member 2
certify
relay log
apply
binlog
commit
Member 3
certify
relay log
apply
binlog
commit

深度分析内容



- MGR深度分析
  - State machine replication
  - 底层Paxos通信机制
  - 实用角度分析
    - 两机房怎么办?
    - 单主如何降低内存消耗量?
    - 如何最大限度容忍低质量网络?
  - 性能模型
    - 集群吞吐量取决于最慢节点
    - 集群吞吐量取决于网络速度
    - 集群吞吐量取决于Relay log写入速度

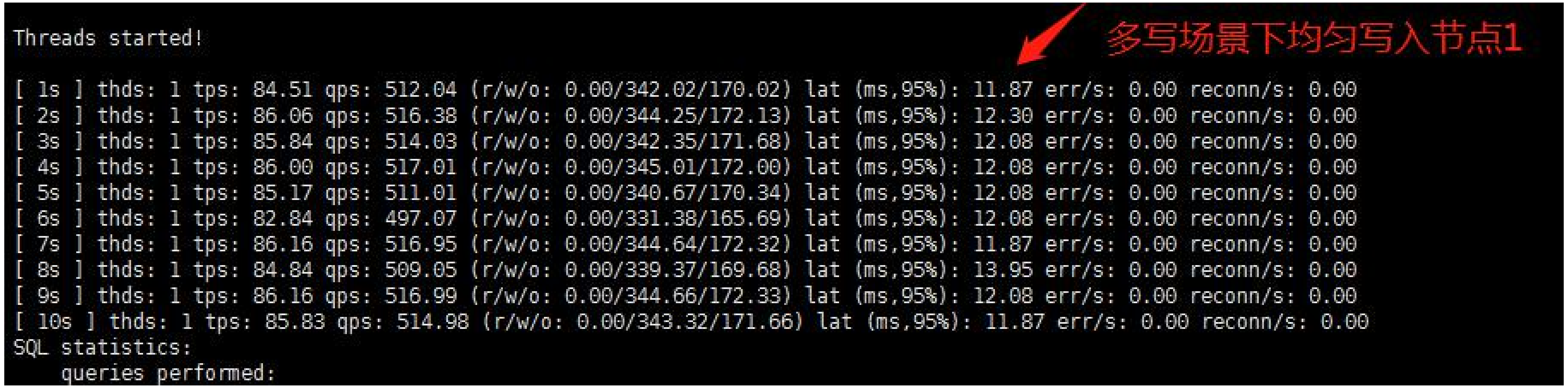
底层Paxos通信机制

参 考 孟 子 算 法

A rotating leader protocol for multi-site systems

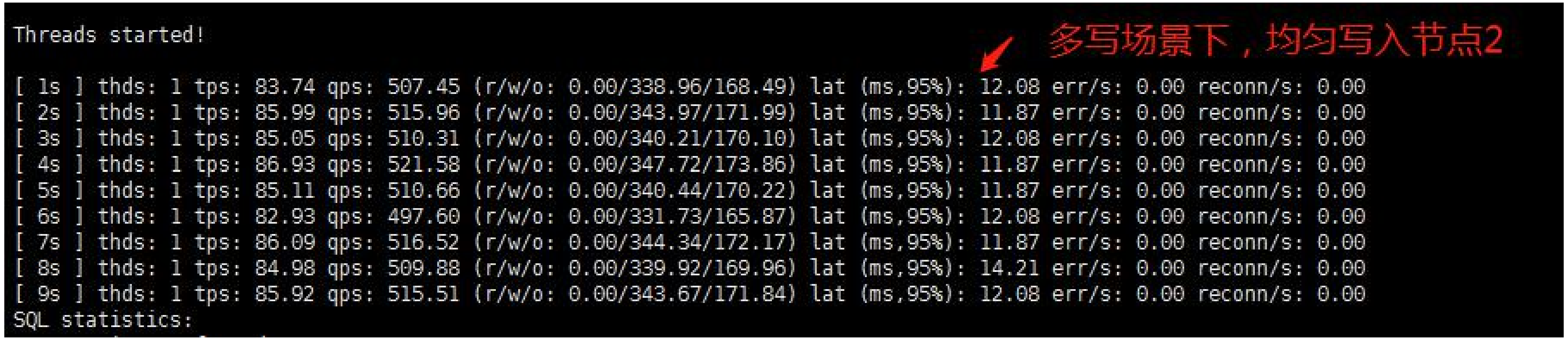
为Wide Area Networks而生的Paxos算法

两个节点之间RTT=10ms



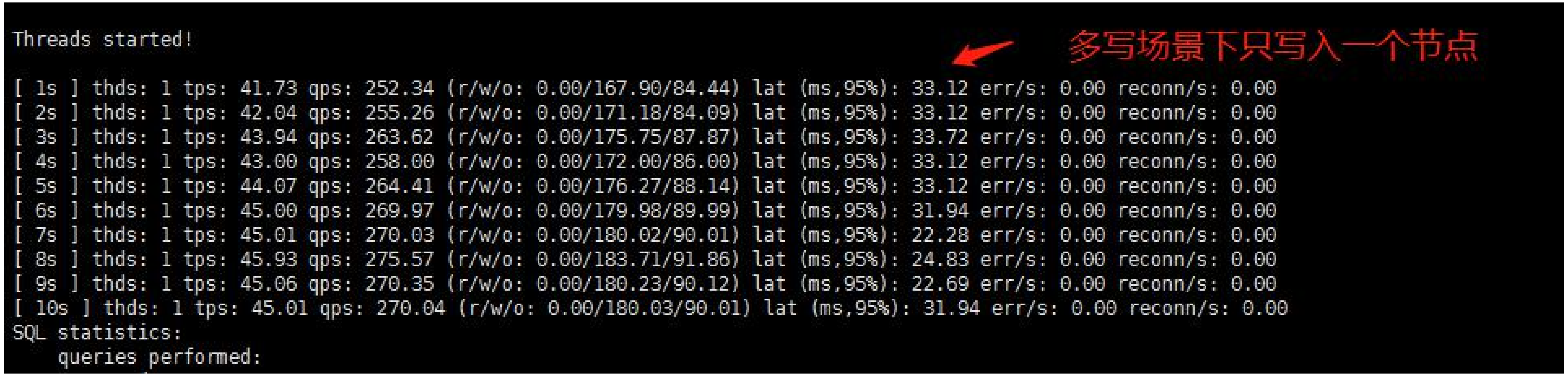
多写场景下均匀写入节点1

```
Threads started!

[ 1s ] thds: 1 tps: 84.51 qps: 512.04 (r/w/o: 0.00/342.02/170.02) lat (ms,95%): 11.87 err/s: 0.00 reconn/s: 0.00
[ 2s ] thds: 1 tps: 86.06 qps: 516.38 (r/w/o: 0.00/344.25/172.13) lat (ms,95%): 12.30 err/s: 0.00 reconn/s: 0.00
[ 3s ] thds: 1 tps: 85.84 qps: 514.03 (r/w/o: 0.00/342.35/171.68) lat (ms,95%): 12.08 err/s: 0.00 reconn/s: 0.00
[ 4s ] thds: 1 tps: 86.00 qps: 517.01 (r/w/o: 0.00/345.01/172.00) lat (ms,95%): 12.08 err/s: 0.00 reconn/s: 0.00
[ 5s ] thds: 1 tps: 85.17 qps: 511.01 (r/w/o: 0.00/340.67/170.34) lat (ms,95%): 12.08 err/s: 0.00 reconn/s: 0.00
[ 6s ] thds: 1 tps: 82.84 qps: 497.07 (r/w/o: 0.00/331.38/165.69) lat (ms,95%): 12.08 err/s: 0.00 reconn/s: 0.00
[ 7s ] thds: 1 tps: 86.16 qps: 516.95 (r/w/o: 0.00/344.64/172.32) lat (ms,95%): 11.87 err/s: 0.00 reconn/s: 0.00
[ 8s ] thds: 1 tps: 84.84 qps: 509.05 (r/w/o: 0.00/339.37/169.68) lat (ms,95%): 13.95 err/s: 0.00 reconn/s: 0.00
[ 9s ] thds: 1 tps: 86.16 qps: 516.99 (r/w/o: 0.00/344.66/172.33) lat (ms,95%): 12.08 err/s: 0.00 reconn/s: 0.00
[ 10s ] thds: 1 tps: 85.83 qps: 514.98 (r/w/o: 0.00/343.32/171.66) lat (ms,95%): 11.87 err/s: 0.00 reconn/s: 0.00
SQL statistics:
    queries performed:
```



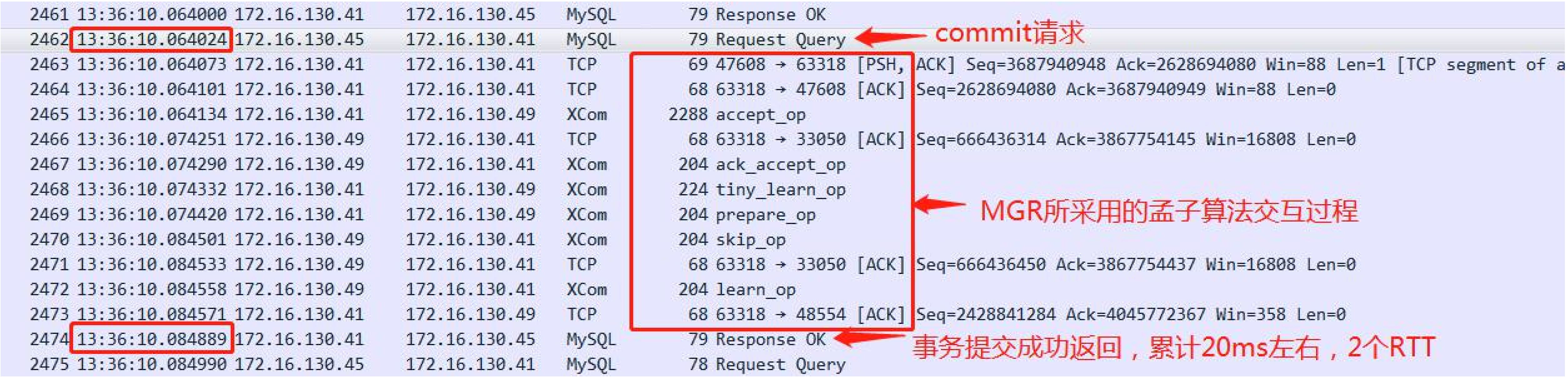
多写场景下，均匀写入节点2

```
Threads started!

[ 1s ] thds: 1 tps: 83.74 qps: 507.45 (r/w/o: 0.00/338.96/168.49) lat (ms,95%): 12.08 err/s: 0.00 reconn/s: 0.00
[ 2s ] thds: 1 tps: 85.99 qps: 515.96 (r/w/o: 0.00/343.97/171.99) lat (ms,95%): 11.87 err/s: 0.00 reconn/s: 0.00
[ 3s ] thds: 1 tps: 85.05 qps: 510.31 (r/w/o: 0.00/340.21/170.10) lat (ms,95%): 12.08 err/s: 0.00 reconn/s: 0.00
[ 4s ] thds: 1 tps: 86.93 qps: 521.58 (r/w/o: 0.00/347.72/173.86) lat (ms,95%): 11.87 err/s: 0.00 reconn/s: 0.00
[ 5s ] thds: 1 tps: 85.11 qps: 510.66 (r/w/o: 0.00/340.44/170.22) lat (ms,95%): 11.87 err/s: 0.00 reconn/s: 0.00
[ 6s ] thds: 1 tps: 82.93 qps: 497.60 (r/w/o: 0.00/331.73/165.87) lat (ms,95%): 12.08 err/s: 0.00 reconn/s: 0.00
[ 7s ] thds: 1 tps: 86.09 qps: 516.52 (r/w/o: 0.00/344.34/172.17) lat (ms,95%): 11.87 err/s: 0.00 reconn/s: 0.00
[ 8s ] thds: 1 tps: 84.98 qps: 509.88 (r/w/o: 0.00/339.92/169.96) lat (ms,95%): 14.21 err/s: 0.00 reconn/s: 0.00
[ 9s ] thds: 1 tps: 85.92 qps: 515.51 (r/w/o: 0.00/343.67/171.84) lat (ms,95%): 12.08 err/s: 0.00 reconn/s: 0.00
SQL statistics:
```



多写场景下只写入一个节点

```
Threads started!

[ 1s ] thds: 1 tps: 41.73 qps: 252.34 (r/w/o: 0.00/167.90/84.44) lat (ms,95%): 33.12 err/s: 0.00 reconn/s: 0.00
[ 2s ] thds: 1 tps: 42.04 qps: 255.26 (r/w/o: 0.00/171.18/84.09) lat (ms,95%): 33.12 err/s: 0.00 reconn/s: 0.00
[ 3s ] thds: 1 tps: 43.94 qps: 263.62 (r/w/o: 0.00/175.75/87.87) lat (ms,95%): 33.72 err/s: 0.00 reconn/s: 0.00
[ 4s ] thds: 1 tps: 43.00 qps: 258.00 (r/w/o: 0.00/172.00/86.00) lat (ms,95%): 33.12 err/s: 0.00 reconn/s: 0.00
[ 5s ] thds: 1 tps: 44.07 qps: 264.41 (r/w/o: 0.00/176.27/88.14) lat (ms,95%): 33.12 err/s: 0.00 reconn/s: 0.00
[ 6s ] thds: 1 tps: 45.00 qps: 269.97 (r/w/o: 0.00/179.98/89.99) lat (ms,95%): 31.94 err/s: 0.00 reconn/s: 0.00
[ 7s ] thds: 1 tps: 45.01 qps: 270.03 (r/w/o: 0.00/180.02/90.01) lat (ms,95%): 22.28 err/s: 0.00 reconn/s: 0.00
[ 8s ] thds: 1 tps: 45.93 qps: 275.57 (r/w/o: 0.00/183.71/91.86) lat (ms,95%): 24.83 err/s: 0.00 reconn/s: 0.00
[ 9s ] thds: 1 tps: 45.06 qps: 270.35 (r/w/o: 0.00/180.23/90.12) lat (ms,95%): 22.69 err/s: 0.00 reconn/s: 0.00
[ 10s ] thds: 1 tps: 45.01 qps: 270.04 (r/w/o: 0.00/180.03/90.01) lat (ms,95%): 31.94 err/s: 0.00 reconn/s: 0.00
SQL statistics:
    queries performed:
```



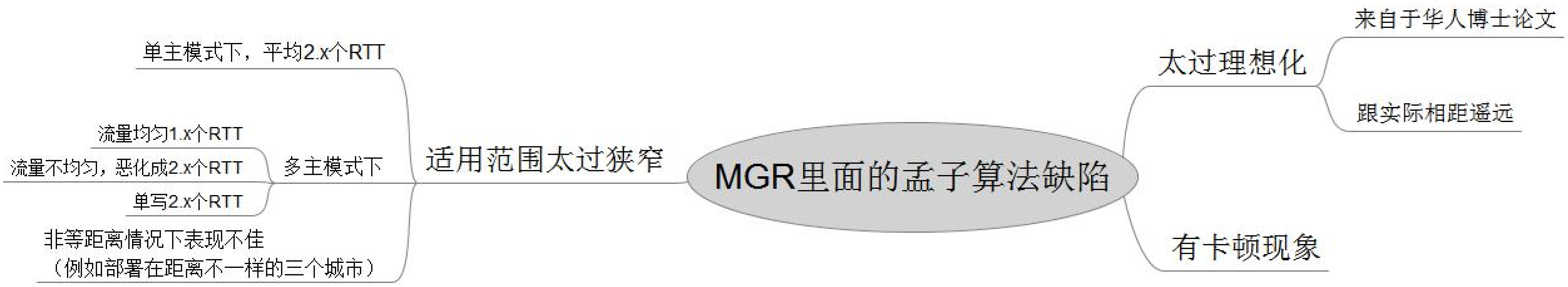
```
2461 13:36:10.064000 172.16.130.41   172.16.130.45   MySQL      79 Response OK
2462 13:36:10.064024 172.16.130.45   172.16.130.41   MySQL      79 Request Query
2463 13:36:10.064073 172.16.130.41   172.16.130.41   TCP        69 47608 → 63318 [PSH, ACK] Seq=3687940948 Ack=2628694080 Win=88 Len=1 [TCP segment of a
2464 13:36:10.064101 172.16.130.41   172.16.130.41   TCP        68 63318 → 47608 [ACK] Seq=2628694080 Ack=3687940949 Win=88 Len=0
2465 13:36:10.064134 172.16.130.41   172.16.130.49   XCom     2288 accept_op
2466 13:36:10.074251 172.16.130.49   172.16.130.41   TCP        68 63318 → 33050 [ACK] Seq=666436314 Ack=3867754145 Win=16808 Len=0
2467 13:36:10.074290 172.16.130.49   172.16.130.41   XCom      204 ack_accept_op
2468 13:36:10.074332 172.16.130.41   172.16.130.49   XCom      224 tiny_learn_op
2469 13:36:10.074420 172.16.130.41   172.16.130.49   XCom      204 prepare_op
2470 13:36:10.084501 172.16.130.49   172.16.130.41   XCom      204 skip_op
2471 13:36:10.084533 172.16.130.49   172.16.130.41   TCP        68 63318 → 33050 [ACK] Seq=666436450 Ack=3867754437 Win=16808 Len=0
2472 13:36:10.084558 172.16.130.49   172.16.130.41   XCom      204 learn_op
2473 13:36:10.084571 172.16.130.41   172.16.130.49   TCP        68 63318 → 48554 [ACK] Seq=2428841284 Ack=4045772367 Win=358 Len=0
2474 13:36:10.084889 172.16.130.41   172.16.130.45   MySQL      79 Response OK
2475 13:36:10.084990 172.16.130.45   172.16.130.41   MySQL      78 Request Query
```

commit请求

MGR所采用的孟子算法交互过程

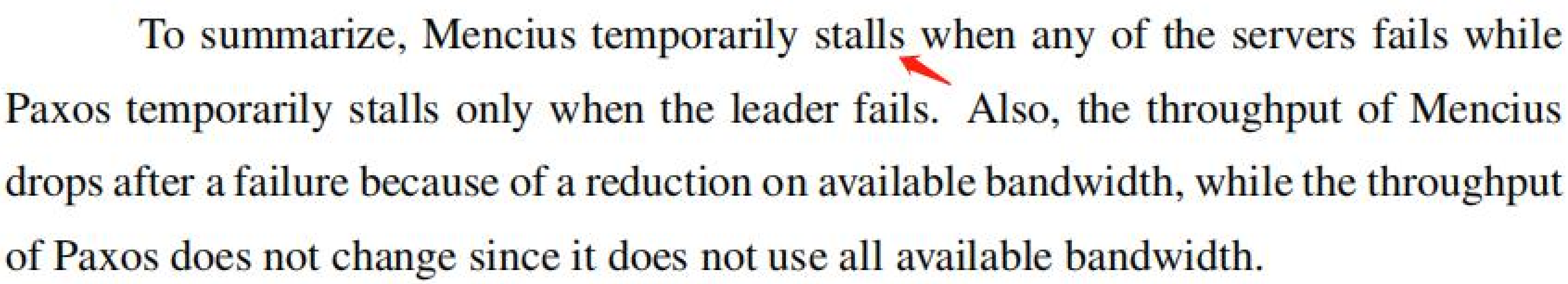
事务提交成功返回，累计20ms左右，2个RTT

MGR实现的孟子算法不适合单主场景



MGR里面的孟子算法缺陷

- 适用范围太过狭窄
  - 单主模式下，平均2.x个RTT
  - 多主模式下
    - 流量均匀1.x个RTT
    - 流量不均匀，恶化成2.x个RTT
    - 单写2.x个RTT
  - 非等距离情况下表现不佳（例如部署在距离不一样的三个城市）
- 太过理想化
  - 来自于华人博士论文
  - 跟实际相距遥远
- 有卡顿现象



To summarize, Mencius temporarily stalls when any of the servers fails while Paxos temporarily stalls only when the leader fails. Also, the throughput of Mencius drops after a failure because of a reduction on available bandwidth, while the throughput of Paxos does not change since it does not use all available bandwidth.

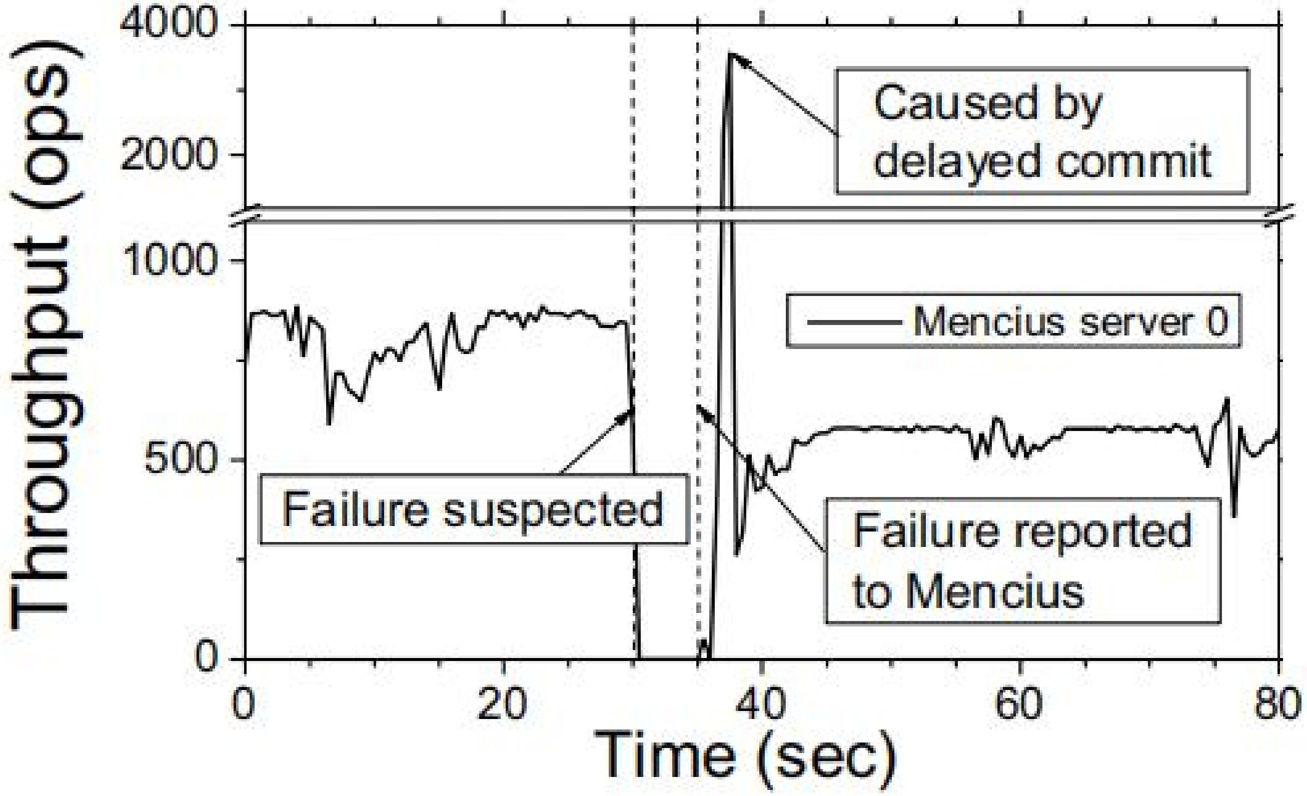
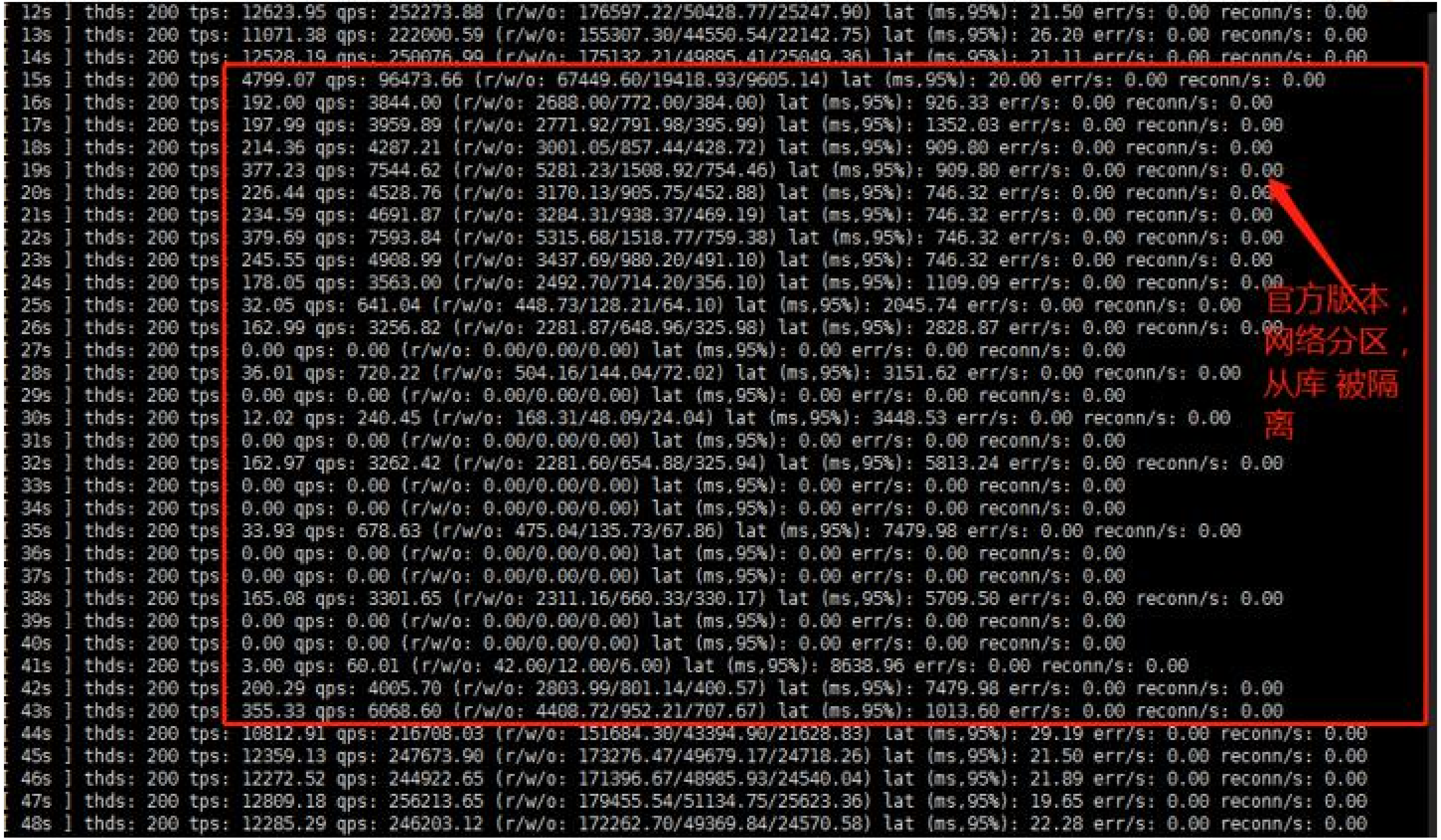
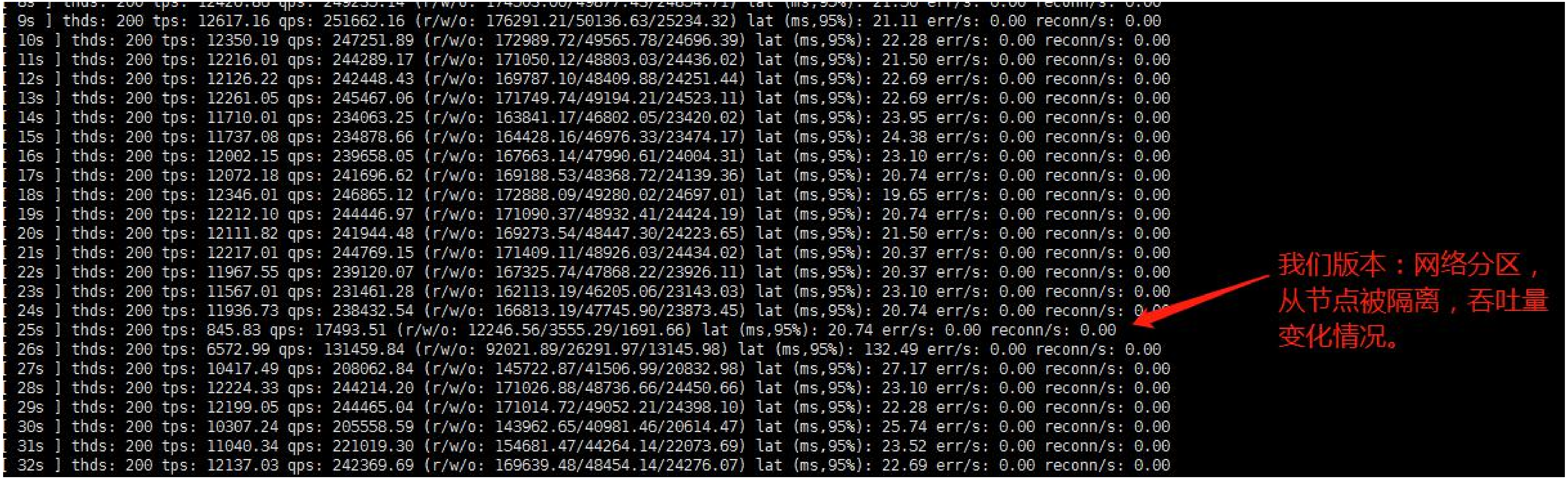


Figure 5.9: The throughput of Mencius when a server crashes



```
[ 12s ] thds: 200 tps: 12623.95 qps: 252273.88 (r/w/o: 176597.22/50428.77/25247.90) lat (ms,95%): 21.50 err/s: 0.00 reconn/s: 0.00
[ 13s ] thds: 200 tps: 11071.38 qps: 222000.59 (r/w/o: 155307.30/44550.54/22142.75) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 14s ] thds: 200 tps: 12528.19 qps: 250076.99 (r/w/o: 175132.21/49895.41/25049.36) lat (ms,95%): 21.11 err/s: 0.00 reconn/s: 0.00
[ 15s ] thds: 200 tps: 4799.07 qps: 96473.66 (r/w/o: 67449.60/19418.93/9605.14) lat (ms,95%): 20.00 err/s: 0.00 reconn/s: 0.00
[ 16s ] thds: 200 tps: 192.00 qps: 3844.00 (r/w/o: 2688.00/772.00/384.00) lat (ms,95%): 926.33 err/s: 0.00 reconn/s: 0.00
[ 17s ] thds: 200 tps: 197.99 qps: 3959.89 (r/w/o: 2771.92/791.98/395.99) lat (ms,95%): 1352.03 err/s: 0.00 reconn/s: 0.00
[ 18s ] thds: 200 tps: 214.36 qps: 4287.21 (r/w/o: 3001.05/857.44/428.72) lat (ms,95%): 909.80 err/s: 0.00 reconn/s: 0.00
[ 19s ] thds: 200 tps: 377.23 qps: 7544.62 (r/w/o: 5281.23/1508.92/754.46) lat (ms,95%): 909.80 err/s: 0.00 reconn/s: 0.00
[ 20s ] thds: 200 tps: 226.44 qps: 4528.76 (r/w/o: 3170.13/905.75/452.88) lat (ms,95%): 746.32 err/s: 0.00 reconn/s: 0.00
[ 21s ] thds: 200 tps: 234.59 qps: 4691.87 (r/w/o: 3284.31/938.37/469.19) lat (ms,95%): 746.32 err/s: 0.00 reconn/s: 0.00
[ 22s ] thds: 200 tps: 379.69 qps: 7593.84 (r/w/o: 5315.68/1518.77/759.38) lat (ms,95%): 746.32 err/s: 0.00 reconn/s: 0.00
[ 23s ] thds: 200 tps: 245.55 qps: 4908.99 (r/w/o: 3437.69/980.20/491.10) lat (ms,95%): 746.32 err/s: 0.00 reconn/s: 0.00
[ 24s ] thds: 200 tps: 178.05 qps: 3563.00 (r/w/o: 2492.70/714.20/356.10) lat (ms,95%): 1109.09 err/s: 0.00 reconn/s: 0.00
[ 25s ] thds: 200 tps: 32.05 qps: 641.04 (r/w/o: 448.73/128.21/64.10) lat (ms,95%): 2045.74 err/s: 0.00 reconn/s: 0.00
[ 26s ] thds: 200 tps: 162.99 qps: 3256.82 (r/w/o: 2281.87/648.96/325.98) lat (ms,95%): 2828.87 err/s: 0.00 reconn/s: 0.00
[ 27s ] thds: 200 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 28s ] thds: 200 tps: 36.01 qps: 720.22 (r/w/o: 504.16/144.04/72.02) lat (ms,95%): 3151.62 err/s: 0.00 reconn/s: 0.00
[ 29s ] thds: 200 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 30s ] thds: 200 tps: 12.02 qps: 240.45 (r/w/o: 168.31/48.09/24.04) lat (ms,95%): 3448.53 err/s: 0.00 reconn/s: 0.00
[ 31s ] thds: 200 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 32s ] thds: 200 tps: 162.97 qps: 3262.42 (r/w/o: 2281.60/654.88/325.94) lat (ms,95%): 5813.24 err/s: 0.00 reconn/s: 0.00
[ 33s ] thds: 200 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 34s ] thds: 200 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 35s ] thds: 200 tps: 33.93 qps: 678.63 (r/w/o: 475.04/135.73/67.86) lat (ms,95%): 7479.98 err/s: 0.00 reconn/s: 0.00
[ 36s ] thds: 200 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 37s ] thds: 200 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 38s ] thds: 200 tps: 165.08 qps: 3301.65 (r/w/o: 2311.16/660.33/330.17) lat (ms,95%): 5709.50 err/s: 0.00 reconn/s: 0.00
[ 39s ] thds: 200 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 40s ] thds: 200 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 41s ] thds: 200 tps: 3.00 qps: 60.01 (r/w/o: 42.00/12.00/6.00) lat (ms,95%): 8638.96 err/s: 0.00 reconn/s: 0.00
[ 42s ] thds: 200 tps: 200.29 qps: 4005.70 (r/w/o: 2803.99/801.14/400.57) lat (ms,95%): 7479.98 err/s: 0.00 reconn/s: 0.00
[ 43s ] thds: 200 tps: 355.33 qps: 6068.60 (r/w/o: 4408.72/952.21/707.67) lat (ms,95%): 1013.60 err/s: 0.00 reconn/s: 0.00
[ 44s ] thds: 200 tps: 10812.91 qps: 216708.03 (r/w/o: 151684.30/43394.90/21628.83) lat (ms,95%): 29.19 err/s: 0.00 reconn/s: 0.00
[ 45s ] thds: 200 tps: 12359.13 qps: 247673.90 (r/w/o: 173276.47/49679.17/24718.26) lat (ms,95%): 21.50 err/s: 0.00 reconn/s: 0.00
[ 46s ] thds: 200 tps: 12272.52 qps: 244922.65 (r/w/o: 171396.67/48985.93/24540.04) lat (ms,95%): 21.89 err/s: 0.00 reconn/s: 0.00
[ 47s ] thds: 200 tps: 12809.18 qps: 256213.65 (r/w/o: 179455.54/51134.75/25623.36) lat (ms,95%): 19.65 err/s: 0.00 reconn/s: 0.00
[ 48s ] thds: 200 tps: 12285.29 qps: 246203.12 (r/w/o: 172262.70/49369.84/24570.58) lat (ms,95%): 22.28 err/s: 0.00 reconn/s: 0.00
```

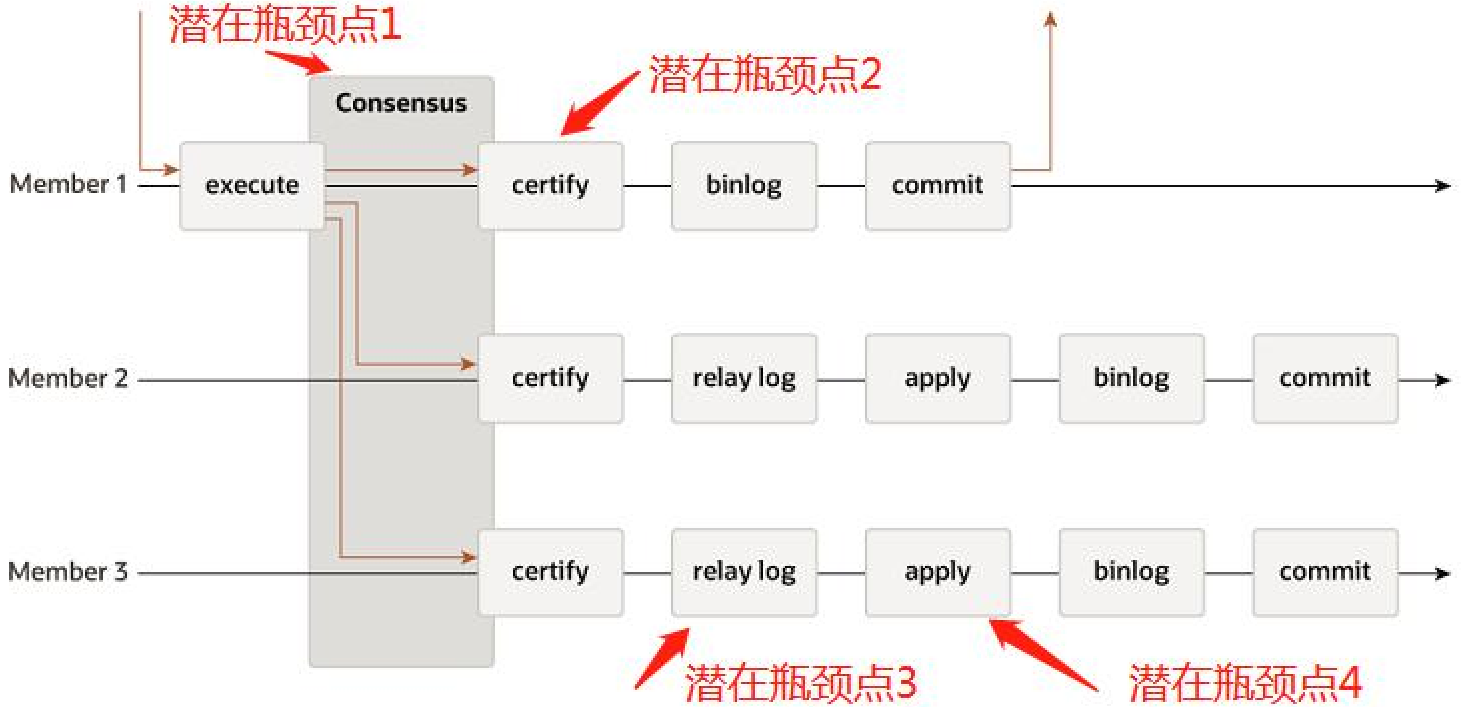
官方版本，网络分区，从库 被隔离

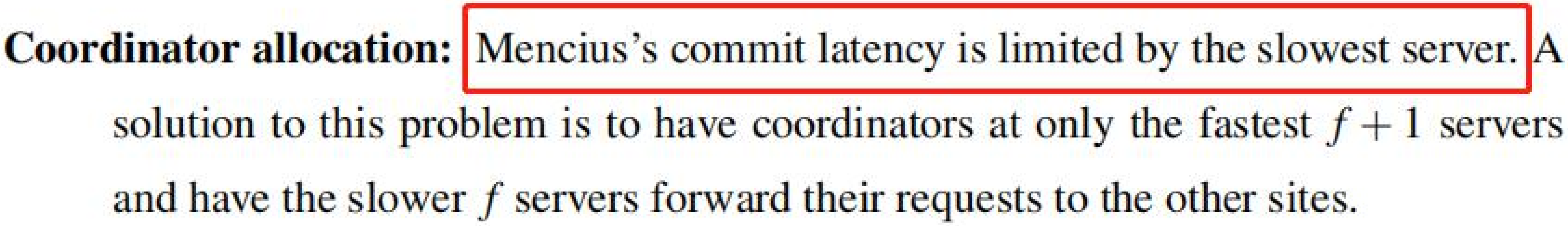


```
[ 9s ] thds: 200 tps: 12617.16 qps: 251662.16 (r/w/o: 176291.21/50136.63/25234.32) lat (ms,95%): 21.11 err/s: 0.00 reconn/s: 0.00
[ 10s ] thds: 200 tps: 12350.19 qps: 247251.89 (r/w/o: 172989.72/49565.78/24696.39) lat (ms,95%): 22.28 err/s: 0.00 reconn/s: 0.00
[ 11s ] thds: 200 tps: 12216.01 qps: 244289.17 (r/w/o: 171050.12/48803.03/24436.02) lat (ms,95%): 21.50 err/s: 0.00 reconn/s: 0.00
[ 12s ] thds: 200 tps: 12126.22 qps: 242448.43 (r/w/o: 169787.10/48409.88/24251.44) lat (ms,95%): 22.69 err/s: 0.00 reconn/s: 0.00
[ 13s ] thds: 200 tps: 12261.05 qps: 245467.06 (r/w/o: 171749.74/49194.21/24523.11) lat (ms,95%): 22.69 err/s: 0.00 reconn/s: 0.00
[ 14s ] thds: 200 tps: 11710.01 qps: 234063.25 (r/w/o: 163841.17/46802.05/23420.02) lat (ms,95%): 23.95 err/s: 0.00 reconn/s: 0.00
[ 15s ] thds: 200 tps: 11737.08 qps: 234878.66 (r/w/o: 164428.16/46976.33/23474.17) lat (ms,95%): 24.38 err/s: 0.00 reconn/s: 0.00
[ 16s ] thds: 200 tps: 12002.15 qps: 239658.05 (r/w/o: 167663.14/47990.61/24004.31) lat (ms,95%): 23.10 err/s: 0.00 reconn/s: 0.00
[ 17s ] thds: 200 tps: 12072.18 qps: 241696.62 (r/w/o: 169188.53/48368.72/24139.36) lat (ms,95%): 20.74 err/s: 0.00 reconn/s: 0.00
[ 18s ] thds: 200 tps: 12346.01 qps: 246865.12 (r/w/o: 172888.09/49280.02/24697.01) lat (ms,95%): 19.65 err/s: 0.00 reconn/s: 0.00
[ 19s ] thds: 200 tps: 12212.10 qps: 244446.97 (r/w/o: 171090.37/48932.41/24424.19) lat (ms,95%): 20.74 err/s: 0.00 reconn/s: 0.00
[ 20s ] thds: 200 tps: 12111.82 qps: 241944.48 (r/w/o: 169273.54/48447.30/24223.65) lat (ms,95%): 21.50 err/s: 0.00 reconn/s: 0.00
[ 21s ] thds: 200 tps: 12217.01 qps: 244769.15 (r/w/o: 171409.11/48926.03/24434.02) lat (ms,95%): 20.37 err/s: 0.00 reconn/s: 0.00
[ 22s ] thds: 200 tps: 11967.55 qps: 239120.07 (r/w/o: 167325.74/47868.22/23926.11) lat (ms,95%): 20.37 err/s: 0.00 reconn/s: 0.00
[ 23s ] thds: 200 tps: 11567.01 qps: 231461.28 (r/w/o: 162113.19/46205.06/23143.03) lat (ms,95%): 23.10 err/s: 0.00 reconn/s: 0.00
[ 24s ] thds: 200 tps: 11936.73 qps: 238432.54 (r/w/o: 166813.19/47745.90/23873.45) lat (ms,95%): 20.74 err/s: 0.00 reconn/s: 0.00
[ 25s ] thds: 200 tps: 845.83 qps: 17493.51 (r/w/o: 12246.56/3555.29/1691.66) lat (ms,95%): 20.74 err/s: 0.00 reconn/s: 0.00
[ 26s ] thds: 200 tps: 6572.99 qps: 131459.84 (r/w/o: 92021.89/26291.97/13145.98) lat (ms,95%): 132.49 err/s: 0.00 reconn/s: 0.00
[ 27s ] thds: 200 tps: 10417.49 qps: 208062.84 (r/w/o: 145722.87/41506.99/20832.98) lat (ms,95%): 27.17 err/s: 0.00 reconn/s: 0.00
[ 28s ] thds: 200 tps: 12224.33 qps: 244214.20 (r/w/o: 171026.88/48736.66/24450.66) lat (ms,95%): 23.10 err/s: 0.00 reconn/s: 0.00
[ 29s ] thds: 200 tps: 12199.05 qps: 244465.04 (r/w/o: 171014.72/49052.21/24398.10) lat (ms,95%): 22.28 err/s: 0.00 reconn/s: 0.00
[ 30s ] thds: 200 tps: 10307.24 qps: 205558.59 (r/w/o: 143962.65/40981.46/20614.47) lat (ms,95%): 25.74 err/s: 0.00 reconn/s: 0.00
[ 31s ] thds: 200 tps: 11040.34 qps: 221019.30 (r/w/o: 154681.47/44264.14/22073.69) lat (ms,95%): 23.52 err/s: 0.00 reconn/s: 0.00
[ 32s ] thds: 200 tps: 12137.03 qps: 242369.69 (r/w/o: 169639.48/48454.14/24276.07) lat (ms,95%): 22.69 err/s: 0.00 reconn/s: 0.00
```

我们版本：网络分区，从节点被隔离，吞吐量变化情况。

性能模型


潜在瓶颈点1
Consensus
潜在瓶颈点2
Member 1
execute
certify
binlog
commit
Member 2
certify
relay log
apply
binlog
commit
Member 3
certify
relay log
apply
binlog
commit
潜在瓶颈点3
潜在瓶颈点4

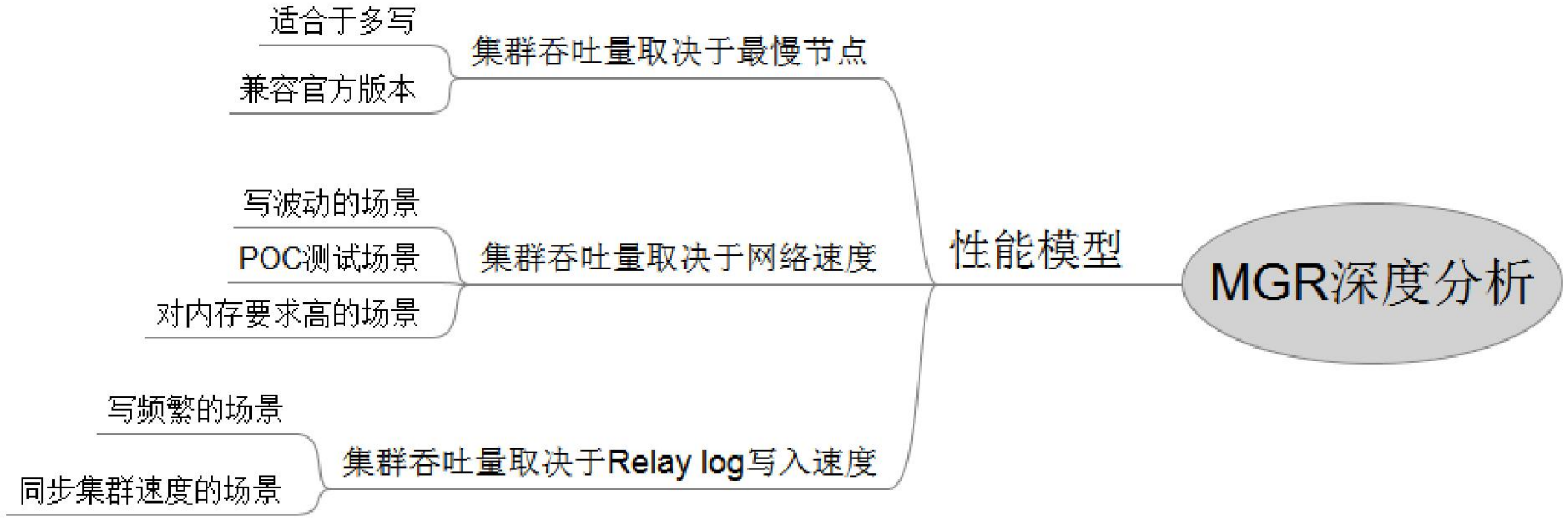


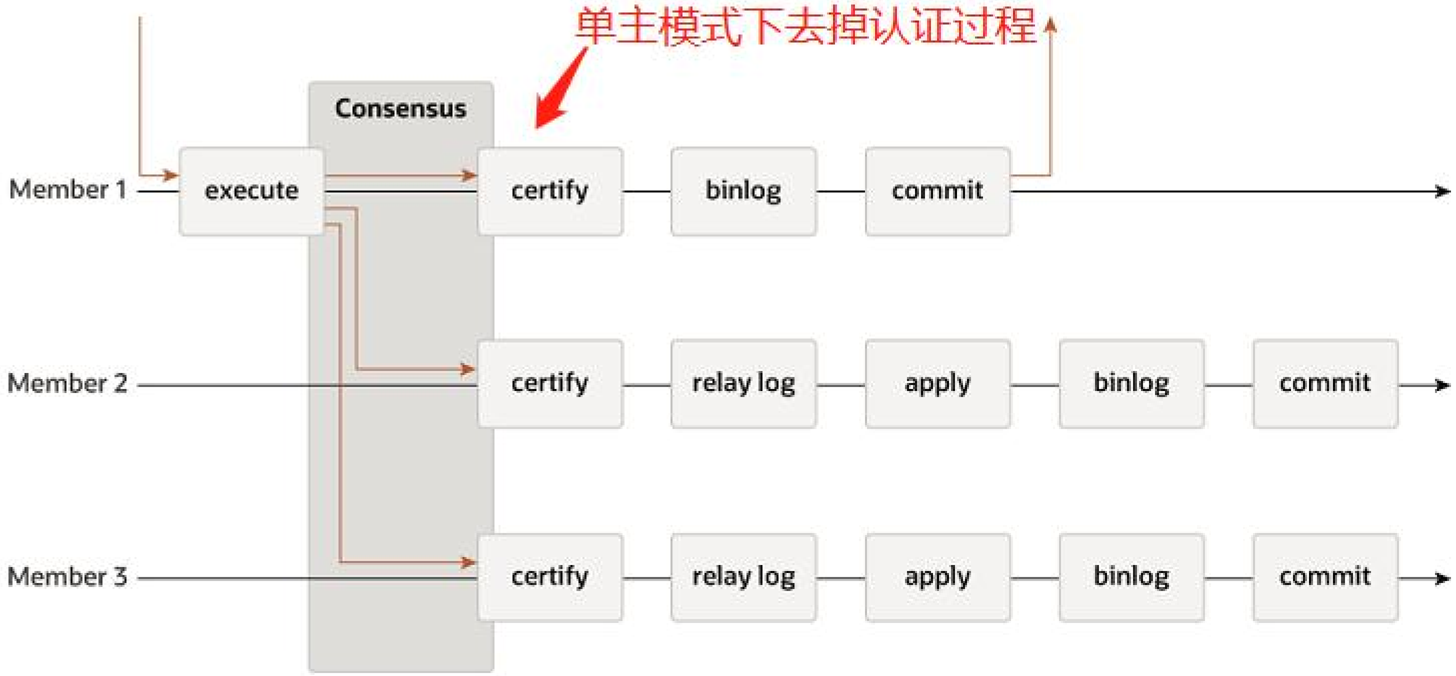
Coordinator allocation: Mencius’s commit latency is limited by the slowest server. A solution to this problem is to have coordinators at only the fastest $f+1$ servers and have the slower f servers forward their requests to the other sites.

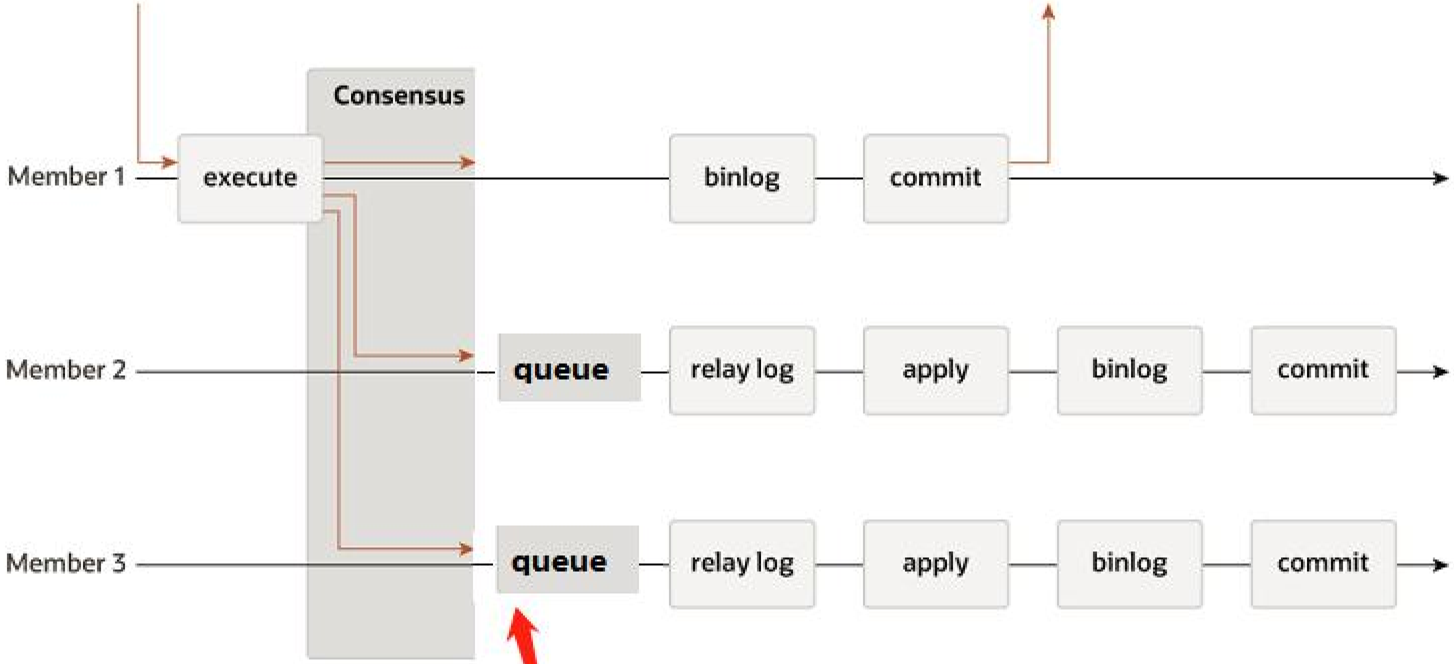
MGR采用了最慢节点性能模型

在POC测试场景下，

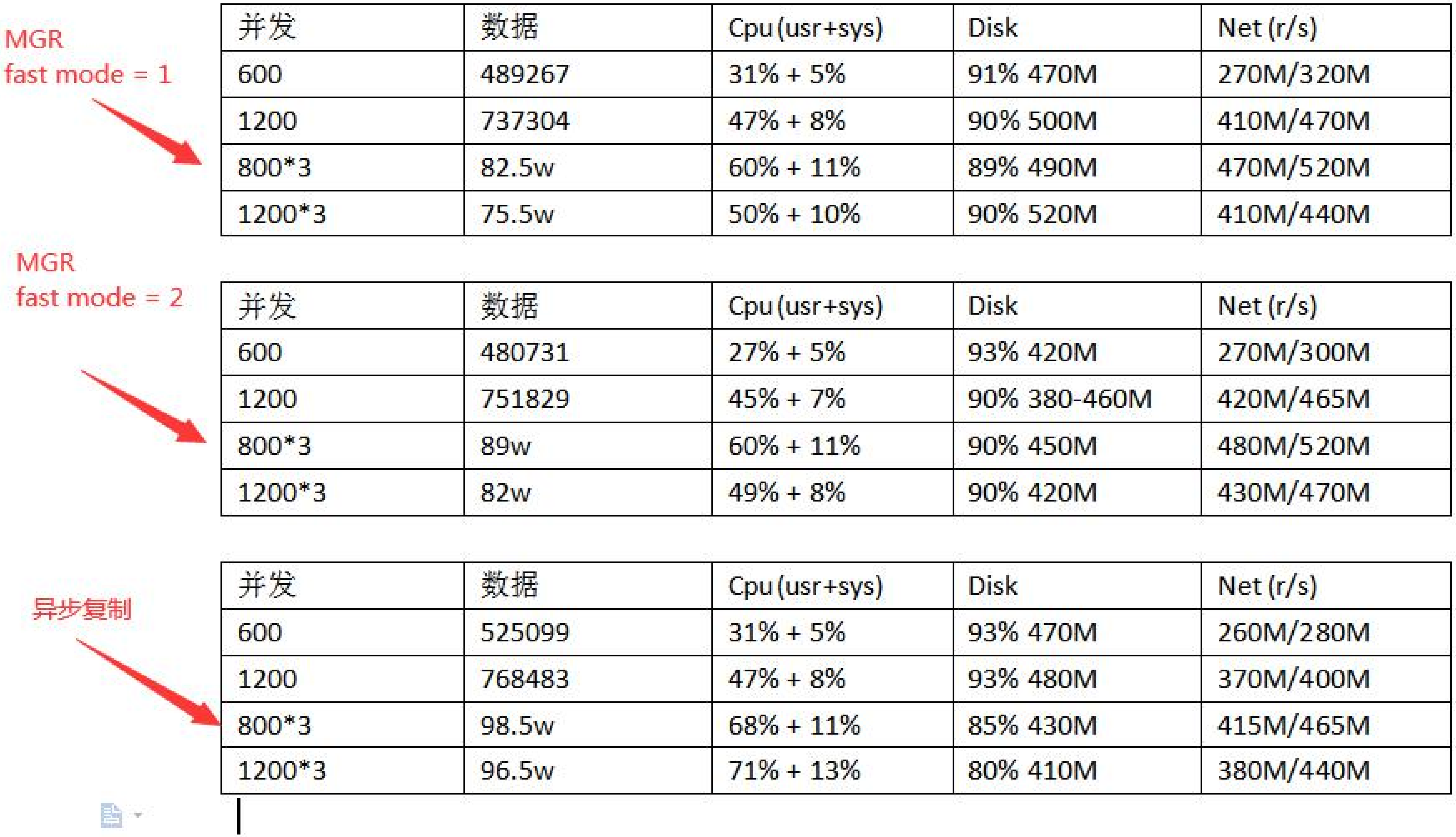
最慢节点性能模型是没有前途的


MGR深度分析
性能模型
集群吞吐量取决于最慢节点
适合于多写
兼容官方版本
集群吞吐量取决于网络速度
写波动的场景
POC测试场景
对内存要求高的场景
集群吞吐量取决于Relay log写入速度
写频繁的场景
同步集群速度的场景


单主模式下去掉认证过程
Consensus
Member 1
execute
certify
binlog
commit
Member 2
certify
relay log
apply
binlog
commit
Member 3
certify
relay log
apply
binlog
commit



不再需要认证数据库
事务信息放入到内存queue中去



MGR
fast mode = 1

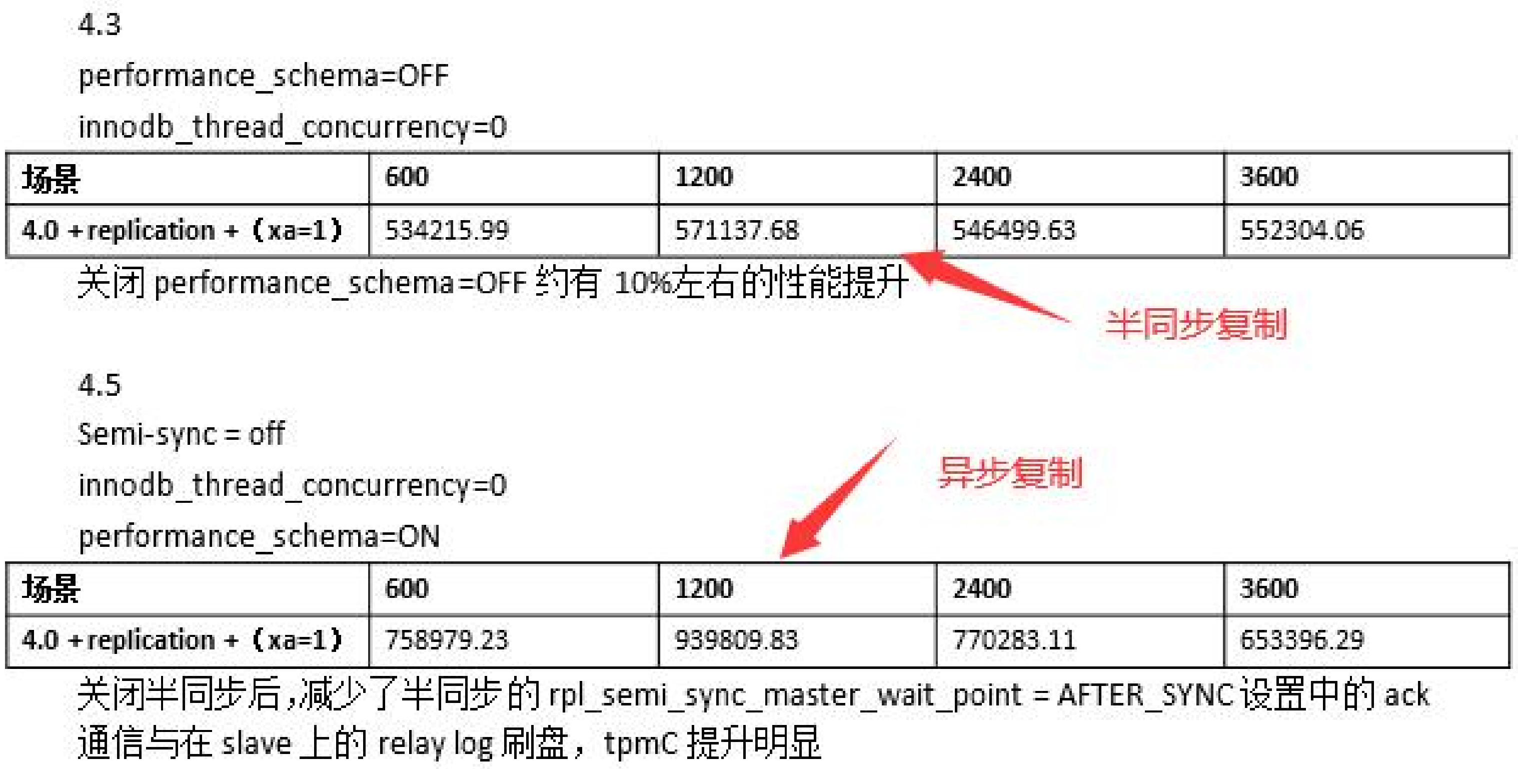
并发	数据	Cpu (usr+sys)	Disk	Net (r/s)
600	489267	31% + 5%	91% 470M	270M/320M
1200	737304	47% + 8%	90% 500M	410M/470M
800*3	82.5w	60% + 11%	89% 490M	470M/520M
1200*3	75.5w	50% + 10%	90% 520M	410M/440M

MGR
fast mode = 2

并发	数据	Cpu (usr+sys)	Disk	Net (r/s)
600	480731	27% + 5%	93% 420M	270M/300M
1200	751829	45% + 7%	90% 380-460M	420M/465M
800*3	89w	60% + 11%	90% 450M	480M/520M
1200*3	82w	49% + 8%	90% 420M	430M/470M

异步复制

并发	数据	Cpu (usr+sys)	Disk	Net (r/s)
600	525099	31% + 5%	93% 470M	260M/280M
1200	768483	47% + 8%	93% 480M	370M/400M
800*3	98.5w	68% + 11%	85% 430M	415M/465M
1200*3	96.5w	71% + 13%	80% 410M	380M/440M



4.3

performance_schema=OFF

innodb_thread_concurrency=0

场景	600	1200	2400	3600
4.0 +replication +（xa=1）	534215.99	571137.68	546499.63	552304.06

关闭 performance_schema=OFF 约有 10%左右的性能提升

半同步复制

4.5

Semi-sync = off

innodb_thread_concurrency=0

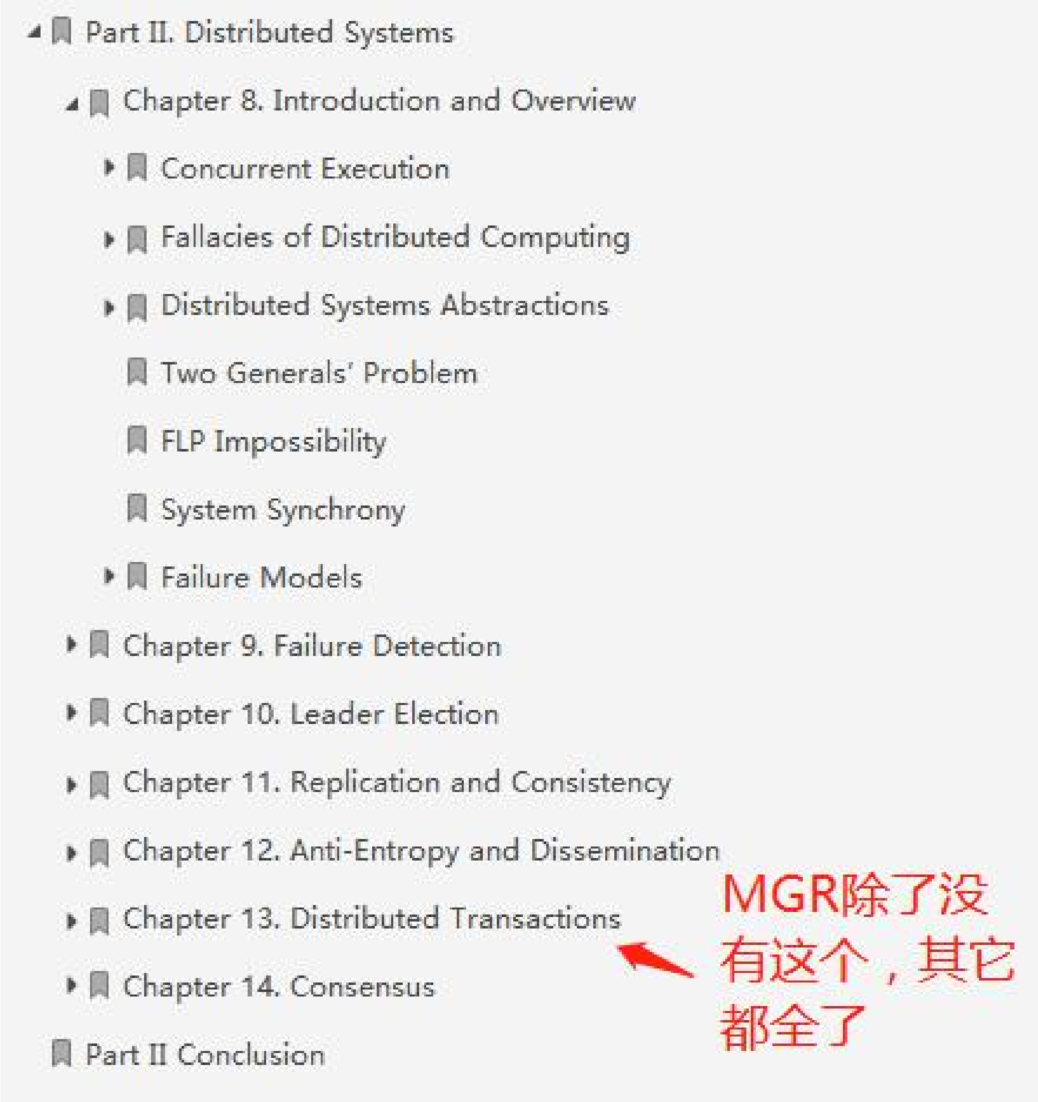
performance_schema=ON

异步复制

场景	600	1200	2400	3600
4.0 +replication +（xa=1）	758979.23	939809.83	770283.11	653396.29

关闭半同步后，减少了半同步的 rpl_semi_sync_master_wait_point = AFTER_SYNC 设置中的 ack 通信与在 slave 上的 relay log 刷盘，tpmC 提升明显

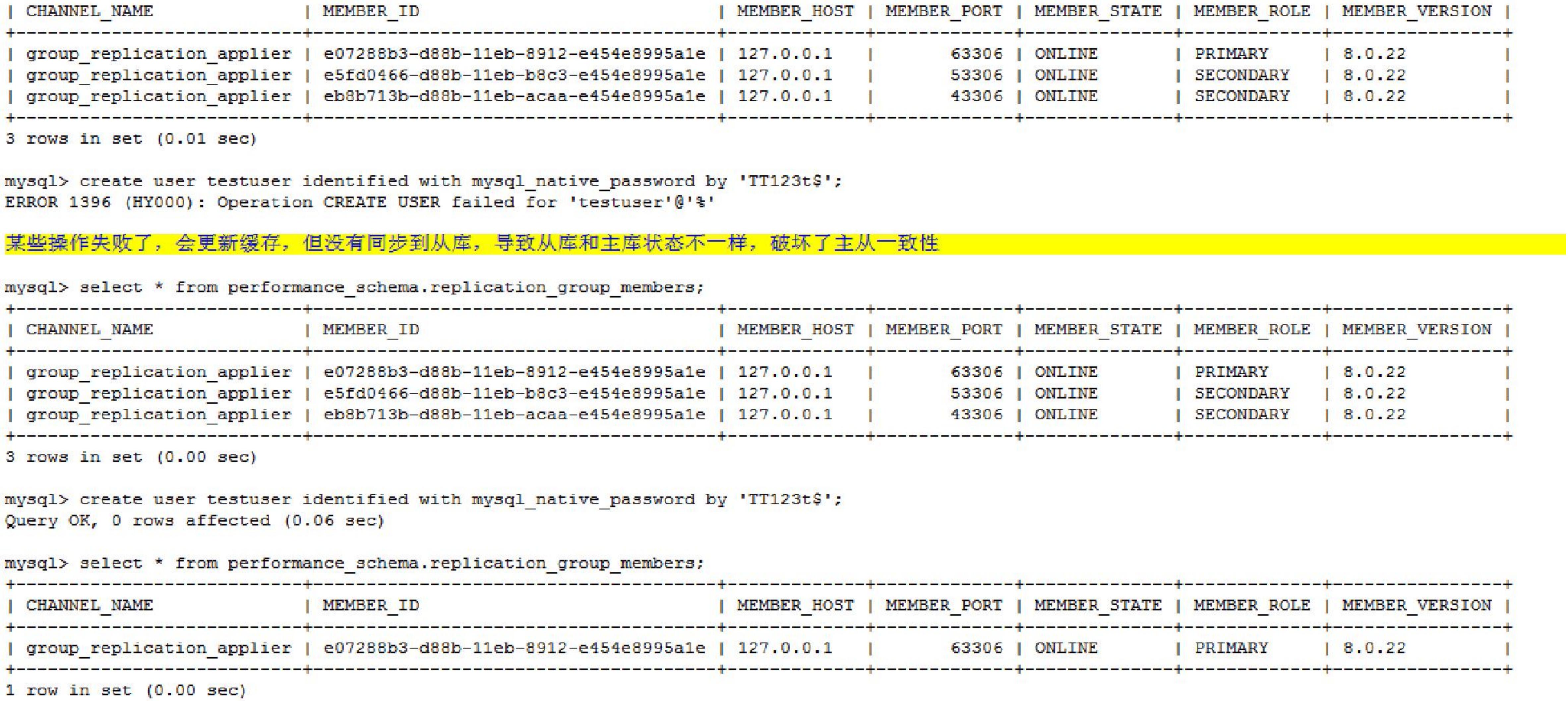
State Machine Replication



- Part II. Distributed Systems
  - Chapter 8. Introduction and Overview
    - Concurrent Execution
    - Fallacies of Distributed Computing
    - Distributed Systems Abstractions
    - Two Generals' Problem
    - FLP Impossibility
    - System Synchrony
    - Failure Models
  - Chapter 9. Failure Detection
  - Chapter 10. Leader Election
  - Chapter 11. Replication and Consistency
  - Chapter 12. Anti-Entropy and Dissemination
  - Chapter 13. Distributed Transactions
  - Chapter 14. Consensus
- Part II Conclusion

MGR除了没有这个，其它都全了

并不是每一个MySQL操作都是状态机操作



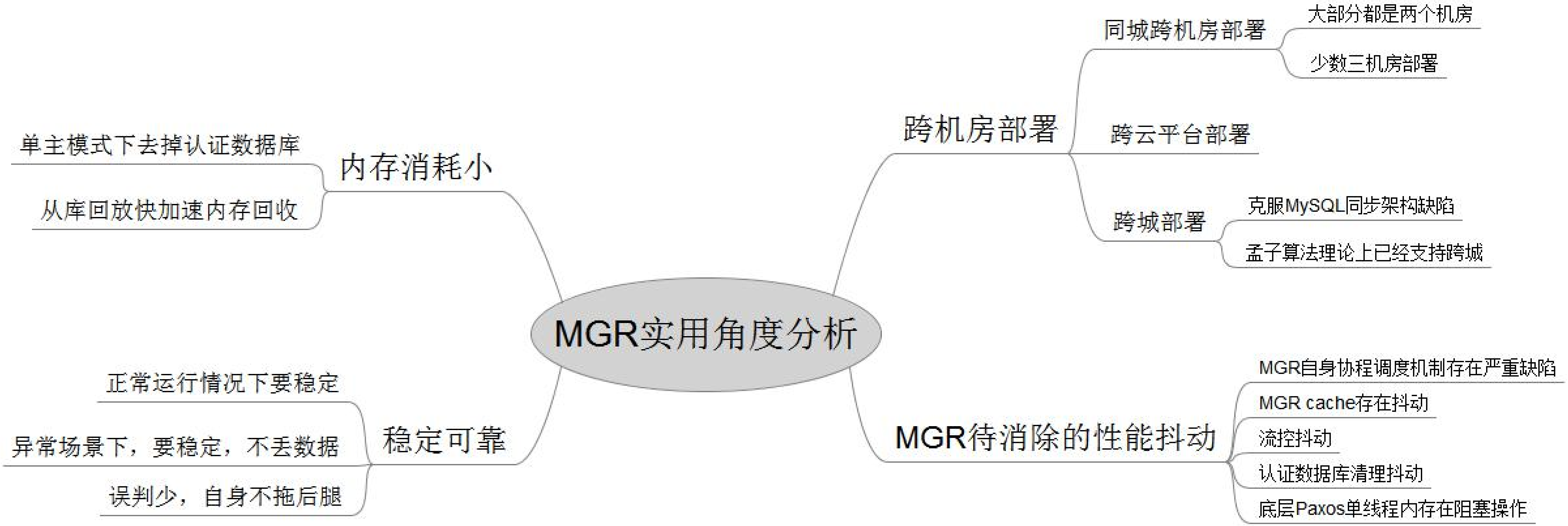
```
| CHANNEL_NAME              | MEMBER_ID                            | MEMBER_HOST | MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION |
+---------------------------+--------------------------------------+-------------+-------------+--------------+-------------+----------------+
| group_replication_applier | e07288b3-d88b-11eb-8912-e454e8995a1e | 127.0.0.1   |       63306 | ONLINE       | PRIMARY     | 8.0.22         |
| group_replication_applier | e5fd0466-d88b-11eb-b8c3-e454e8995a1e | 127.0.0.1   |       53306 | ONLINE       | SECONDARY   | 8.0.22         |
| group_replication_applier | eb8b713b-d88b-11eb-acaa-e454e8995a1e | 127.0.0.1   |       43306 | ONLINE       | SECONDARY   | 8.0.22         |
+---------------------------+--------------------------------------+-------------+-------------+--------------+-------------+----------------+
3 rows in set (0.01 sec)

mysql> create user testuser identified with mysql_native_password by 'TT123t$';
ERROR 1396 (HY000): Operation CREATE USER failed for 'testuser'@'%'
```

某些操作失败了，会更新缓存，但没有同步到从库，导致从库和主库状态不一样，破坏了主从一致性

```
mysql> select * from performance_schema.replication_group_members;
+---------------------------+--------------------------------------+-------------+-------------+--------------+-------------+----------------+
| CHANNEL_NAME              | MEMBER_ID                            | MEMBER_HOST | MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION |
+---------------------------+--------------------------------------+-------------+-------------+--------------+-------------+----------------+
| group_replication_applier | e07288b3-d88b-11eb-8912-e454e8995a1e | 127.0.0.1   |       63306 | ONLINE       | PRIMARY     | 8.0.22         |
| group_replication_applier | e5fd0466-d88b-11eb-b8c3-e454e8995a1e | 127.0.0.1   |       53306 | ONLINE       | SECONDARY   | 8.0.22         |
| group_replication_applier | eb8b713b-d88b-11eb-acaa-e454e8995a1e | 127.0.0.1   |       43306 | ONLINE       | SECONDARY   | 8.0.22         |
+---------------------------+--------------------------------------+-------------+-------------+--------------+-------------+----------------+
3 rows in set (0.00 sec)

mysql> create user testuser identified with mysql_native_password by 'TT123t$';
Query OK, 0 rows affected (0.06 sec)

mysql> select * from performance_schema.replication_group_members;
+---------------------------+--------------------------------------+-------------+-------------+--------------+-------------+----------------+
| CHANNEL_NAME              | MEMBER_ID                            | MEMBER_HOST | MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION |
+---------------------------+--------------------------------------+-------------+-------------+--------------+-------------+----------------+
| group_replication_applier | e07288b3-d88b-11eb-8912-e454e8995a1e | 127.0.0.1   |       63306 | ONLINE       | PRIMARY     | 8.0.22         |
+---------------------------+--------------------------------------+-------------+-------------+--------------+-------------+----------------+
1 row in set (0.00 sec)
```

实用角度分析

我们到底需要一个什么样的MGR？



MGR实用角度分析

- 跨机房部署
  - 同城跨机房部署
    - 大部分都是两个机房
    - 少数三机房部署
  - 跨云平台部署
  - 跨城部署
    - 克服MySQL同步架构缺陷
    - 孟子算法理论上已经支持跨城
- MGR待消除的性能抖动
  - MGR自身协程调度机制存在严重缺陷
  - MGR cache存在抖动
  - 流控抖动
  - 认证数据库清理抖动
  - 底层Paxos单线程内存在阻塞操作
- 内存消耗小
  - 单主模式下去掉认证数据库
  - 从库回放快加速内存回收
- 稳定可靠
  - 正常运行情况下要稳定
  - 异常场景下，要稳定，不丢数据
  - 误判少，自身不拖后腿

GreatSQL已经实现

大部分上述用户诉求

同城双机房部署

GreatSQL实现

基于地理标签的paxos通信机制

跨机房部署

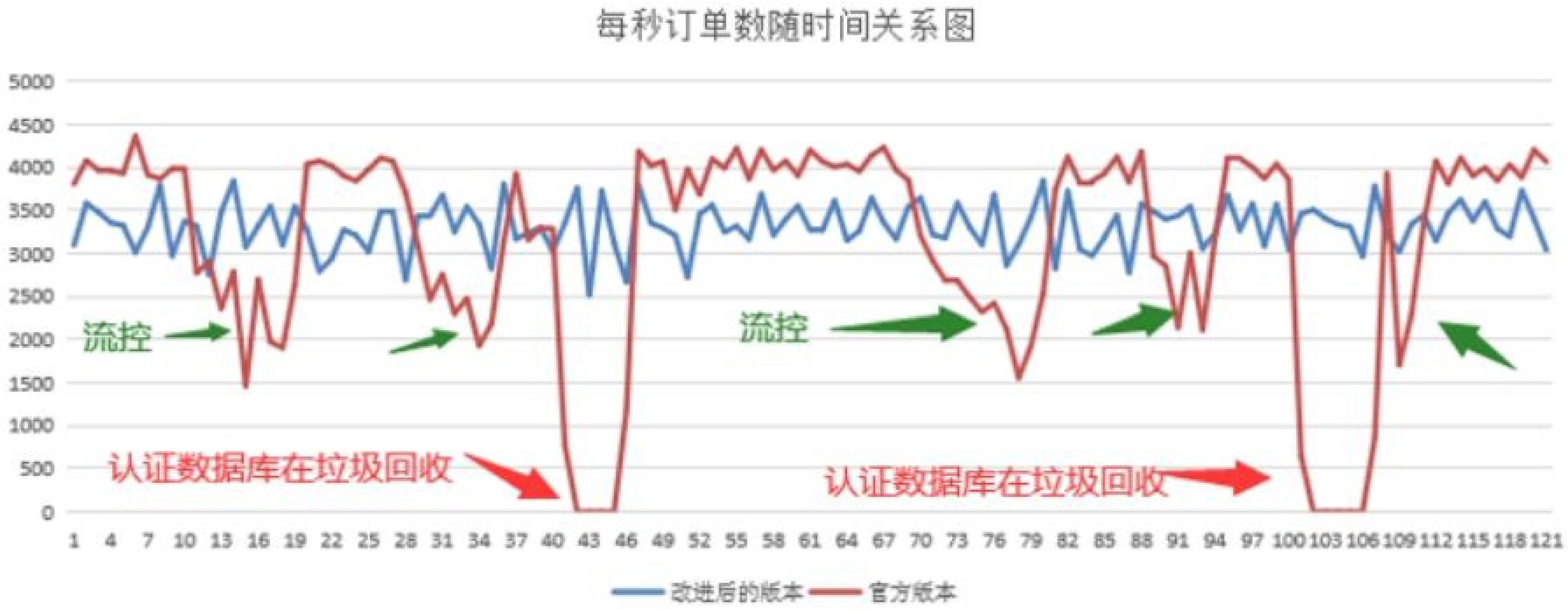
对MGR中的底层孟子算法进行了改进
正常情况下一个RTT完成事务

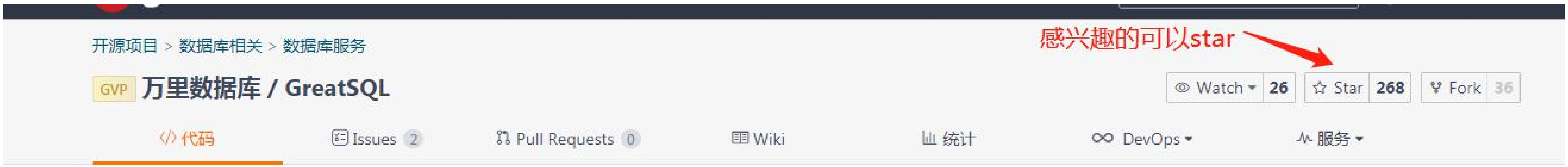
消除多处性能抖动

更加公平的协程调度算法

GreatSQL实现

快速单主模式+多主重构
（新算法+新数据结构）


每秒订单数随时间关系图
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73 76 79 82 85 88 91 94 97 100 103 106 109 112 115 118 121
流控
流控
认证数据库在垃圾回收
认证数据库在垃圾回收
改进后的版本
官方版本



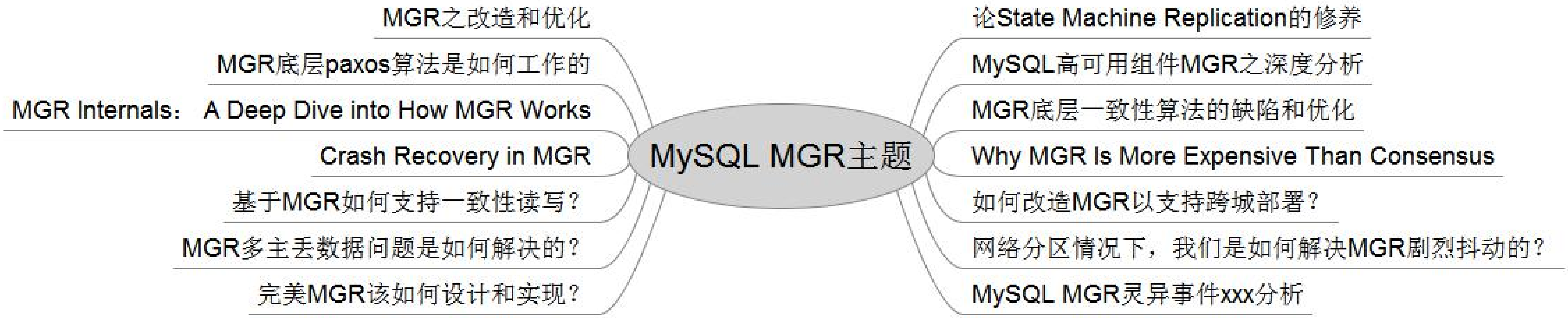
感兴趣的可以star

开源项目 > 数据库相关 > 数据库服务

GVP 万里数据库 / GreatSQL

Watch 26 Star 268 Fork 36

代码 Issues 2 Pull Requests 0 Wiki 统计 DevOps 服务



MySQL MGR主题

MGR之改造和优化

MGR底层paxos算法是如何工作的

MGR Internals： A Deep Dive into How MGR Works

Crash Recovery in MGR

基于MGR如何支持一致性读写？

MGR多主丢数据问题是如何解决的？

完美MGR该如何设计和实现？

论State Machine Replication的修养

MySQL高可用组件MGR之深度分析

MGR底层一致性算法的缺陷和优化

Why MGR Is More Expensive Than Consensus

如何改造MGR以支持跨城部署？

网络分区情况下，我们是如何解决MGR剧烈抖动的？

MySQL MGR灵异事件xxx分析

数据库产品供应商

北京万里开源软件有限公司

专注国产自主可控基础软件产品研发与服务

操作系统产品供应商

GreatDB 万里数据库

基本情况

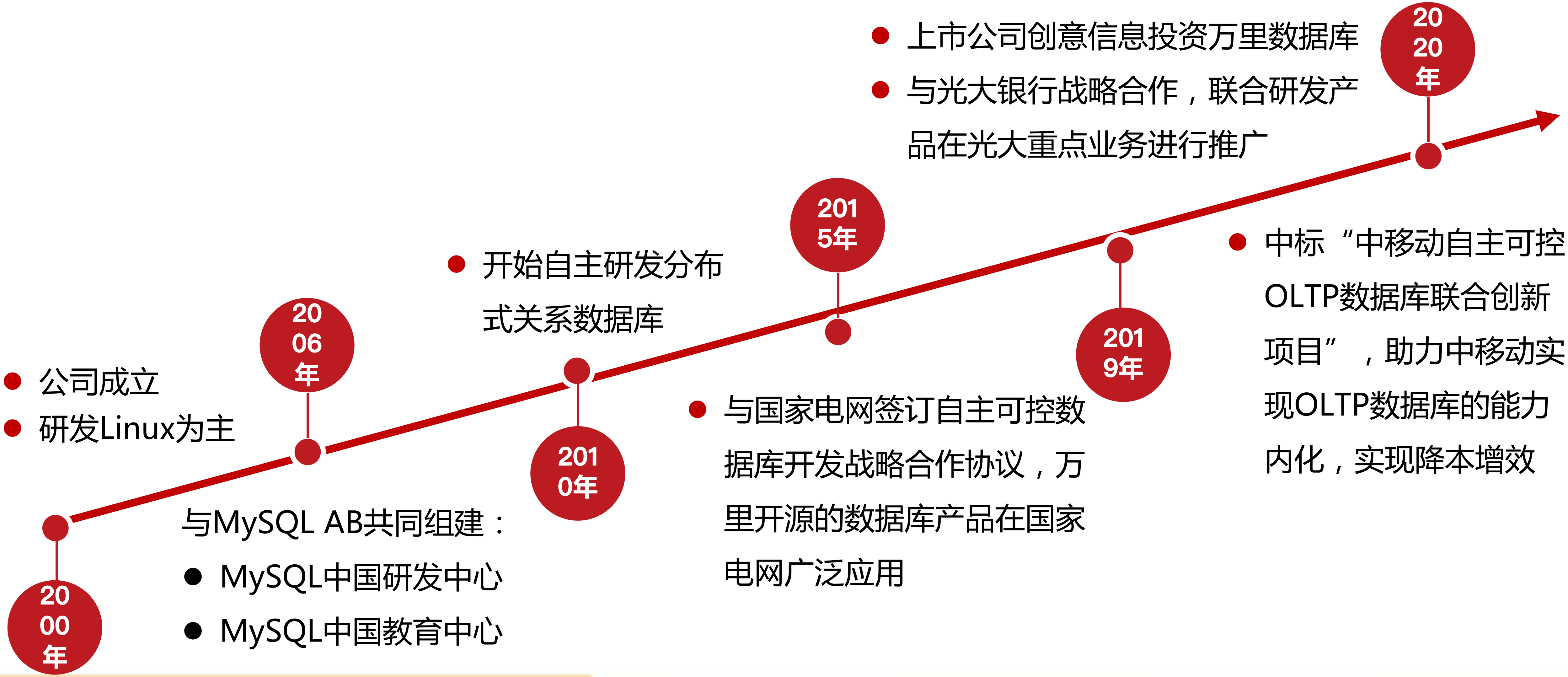
公司成立于2000年10月，创意信息旗下企业，全资控股拓林思软件，20多年基础软件技术沉淀，100%内资背景。

核心竞争力

核心研发人员来自MySQL研发中心，熟练掌握数据库源代码；分布式数据库成功应用在金融、电信和能源等行业核心应用系统。

服务团队

经验丰富的售后实施团队，提供开发支持、系统优化和驻场支持等多种服务内容，给客户提供原厂7*24小时技术服务。



2000年

- 公司成立
- 研发Linux为主

2006年

与MySQL AB共同组建：

- MySQL中国研发中心
- MySQL中国教育中心

2010年

- 开始自主研发分布式关系数据库

2015年

- 与国家电网签订自主可控数据库开发战略合作协议，万里开源的数据库产品在国家电网广泛应用

2019年

- 中标“中移动自主可控OLTP数据库联合创新项目”，助力中移动实现OLTP数据库的能力内化，实现降本增效

2020年

- 上市公司创意信息投资万里数据库
- 与光大银行战略合作，联合研发产品在光大重点业务进行推广

GreatDB
万里数据库

寻找发光的你

THANKS