



2013

# Oracle技术嘉年华

OTN China Tour 2013

数据库技术企业应用最佳实践

## MySQL压力测试经验

如何避免生产环境性能瓶颈

# 个人介绍

## □ 叶金荣

- Oracle ACE
- 搜狐畅游福州([www.17173.com](http://www.17173.com))
- 系统部经理
- [imysql@gmail.com](mailto:imysql@gmail.com)
- Weibo: @yejinrong



# 为什么要压力测试

---

- 采购新设备，评估新设备性能
- 开发新项目，评估数据库容量
- 新系统上线前，预估/模拟数据库负载
- 更换数据库版本，评估性能变化

# 关注指标

---

- CPU  
    %wait , %user , %sys
- 内存  
    只内存读写 , 不产生swap
- IO  
    IOPS、iowait、svctm、%util
- 数据库  
    TPS/TpmC、吞吐量 ( QPS )、响应时长、InnoDB指标

# 影响因素

---

- 硬件
  - CPU ( 省电模式、超线程、多核 )
  - 内存 ( 镜像模式、xen内核限制可用总内存大小 )
  - 阵列卡(BBU、CACHE、条带、读写策略、FW)
  - 硬盘 ( SSD/SAS )
- 系统
  - 内核参数 ( tcp相关 )
  - 文件系统
  - IO调度器

# 影响因素

---

- MySQL
  - TRANSACTION ISOLATION LEVEL
  - Buffer pool
  - Concurrency thread
  - Redo log
  - Binlog sync
  - innodb\_flush\_log\_at\_trx\_commit
  - ...

# 注意事项

---

- 只在本地加压
- 压测数据量小
- 压测时间过短
- 压测模式太少
- 压力负载过大或过小
- 每轮测试完毕要净化环境

# 测试环境

指标	测试环境A	测试环境B
机型	DELL PE R720 ( 2U PC Server )	DELL PE R710 ( 2U PC Server )
CPU	Xeon E5-2620(6核 , 12线程 , 2.0GHz, L3 15MB) * 2	Xeon E5620(4核 , 8线程 , 2.4GHz , L3 12MB) * 2
内存	32G(4G * 8)	32G(4G * 8)
阵列卡及设置	PERC H710 , 512MB , BBU(FW : 12.10.1-0001) , RAID 1+0 FORCE WB	PERC H700 , 512MB , BBU(FW : 12.10.1-0001) , RAID 1+0 FORCE WB
硬盘	15K RPM 300G SAS * 8	15K RPM 300G SAS * 6
网卡	Intel 1GbE	Broadcom 1GbE
操作系统	RHEL 6.4	RHEL 6.4
文件系统	xfs/ext4	xfs/ext4
MySQL版本	5.5.34	5.6.14



# 测试工具

---

- sysbench
  - Primarily for MySQL OLTP benchmarking , By MySQL AB
  - cpu、threads、mutex、memory、fileio、oltp
- tpcc-mysql
  - Primarily for MySQL OLTP benchmarking , By Percona
- tpch
  - Primarily for OLAP benchmarking
- tcpcopy
  - 模拟生产环境真实请求
- 其他
  - mysqlslap
  - sql-bench

# 测试工具 – sysbench

---

- 项目地址
  - <http://sysbench.sourceforge.net/>
- 分支版本
  - <https://code.launchpad.net/sysbench>
  - 特点：支持lua，增加选项 oltp-tables-count

# 测试工具 – sysbench

---

- 安装
  - `./configure --with-mysql-includes=path --with-mysql-libs=path && make && make install`
- 支持其他数据库
  - `with-pgsql`
  - `with-oracle`
- 运行
  - `sysbench --test=[mode] [other_options] prepare`
  - `sysbench --test=[mode] [other_options] run`
  - `sysbench --test=[mode] [other_options] cleanup`

# 测试工具 – sysbench

---

- 通用基准
  - 最大请求数：5,000,000
  - 并发线程数：8 ~ 512
- 基准 – OLTP
  - mode=complex
  - engine=innodb
  - oltp-table-size=100,000,000

# 测试工具 – sysbench

---

- 测试用例

```
sysbench --test=oltp --mysql-host=host \  
--mysql-user=user --mysql-password=passwd \  
--oltp-test-mode=complex --mysql-table-engine=innodb \  
--oltp-table-size=100000000 --mysql-db=db \  
--oltp-table-name=test_tbl --num-threads=128 \  
--max-requests=5000000 run  
  
--oltp-test-mode 测试模式：complex/simple/nontrx
```

# 测试工具 – sysbench

- 测试结果

```
OLTP test statistics:
  queries performed:
    read:          70000
    write:         25000
    other:         10000
    total:        105000
  transactions:    5000   (2001.82 per sec.)
  deadlocks:       0      (0.00 per sec.)
  read/write requests: 95000 (38034.59 per sec.)
  other operations: 10000  (4003.64 per sec.)

Test execution summary:
  total time:      2.4977s
  total number of events: 5000
  total time taken by event execution: 79.6005
  per-request statistics:
    min:          0.0051s
    avg:          0.0159s
    max:          0.0795s
  approx. 95 percentile: 0.0250s

Threads fairness:
  events (avg/stddev): 156.2500/3.72
  execution time (avg/stddev): 2.4875/0.00
```

# 测试工具 – sysbench

---

- 不足
  - 只能模拟简单OLTP
  - 测试表列数少
  - 测试表数据类型少
  - 标准版本只能支持一个测试表

# 测试工具 – tpcc mysql

---

- 安装
  - 下载 bzr branch lp:~percona-dev/perconatools/tpcc-mysql
  - 直接make即可
  - create\_table.sql - 创建数据表
  - add\_fkey\_idx.sql – 创建索引及外键
- 初始化加载数据
  - tpcc\_load db\_host db\_name db\_user db\_passwd db\_warehouse\_num
  - 例如：tpcc\_load localhost tpcc1000 user passwd 1000
- 运行OLTP测试
  - ./tpcc\_start -h localhost -d tpcc1000 -u root -p 'xx' -w 1000 -c 32 -r 120 -l 3600 -f ./tpcc\_mysql\_20120314



# 测试工具 – tpcc mysql

---

- 基准 - OLTP
  - warehouse = 1000
  - max connection = 8 ~ 512
  - warm up = 120(s)
  - run time/duration = 3600(s)

# 测试工具 – tpcc mysql

---

- 测试用例

```
./tpcc_start -h host -d db -u user -p passwd \  
-w 500 -c 128 -r 120 -l 3600
```

-w warehouse/仓库数

-c concurrent threads/并发线程

-r warmup/数据预热时长

-l during time/持续加压时长

# 测试工具 – tpcc mysql

- 测试结果

```
<Constraint Check> (all must be [OK])
[transaction percentage]
    Payment: 43.48% (>=43.0%) [OK]
    Order-Status: 4.35% (>= 4.0%) [OK]
    Delivery: 4.35% (>= 4.0%) [OK]
    Stock-Level: 4.35% (>= 4.0%) [OK]
[response time (at least 90% passed)]
    New-Order: 100.00% [OK]
    Payment: 100.00% [OK]
    Order-Status: 100.00% [OK]
    Delivery: 100.00% [OK]
    Stock-Level: 100.00% [OK]

<Tpms>
    34035.332 Tpms
```

# 测试工具 – tpcc mysql

---

- 不足
  - 测试表模式设计未化
  - 存在外键
  - 部分索引不合理

# 测试工具 – tpch

---

- 安装

- 下载 [http://www.tpc.org/tpch/spec/tpch\\_2\\_16\\_0.zip](http://www.tpc.org/tpch/spec/tpch_2_16_0.zip)
- cp makefile.suite makefile
- 编辑 makefile 文件

```
CC = gcc
```

```
DATABASE = MYSQL
```

```
MACHINE = LINUX
```

```
WORKLOAD = TPCH
```

- 编辑tpcd.h , 增加宏定义

```
#ifdef MYSQL
```

```
#define GEN_QUERY_PLAN ""
```

```
#define START_TRAN "START TRANSACTION"
```

```
#define END_TRAN "COMMIT"
```

```
#define SET_OUTPUT ""
```

```
#define SET_ROWCOUNT "limit %d;\n"
```

```
#define SET_DBASE "use %s;\n"
```

```
#endif
```

# 测试工具 – tpch

---

- 初始化
  - 初始化测试表数据： `./dbgen -s 100`
  - 生成测试数据： `mysql -f tpch < dss.ddl`
  - 默认的初始化模式无主键、无索引
  - `LOAD DATA INFILE`导入数据
  - 注意`max_binlog_cache_size`限制，需要切分文件导入
  - 执行修改主键/外键/额外索引脚本
  - 数据表名全部改成小写，适应TPC-H测试SQL脚本
  - 运行`qgen`生成测试SQL
  - 修改部分SQL语句
  - 拆分完成测试SQL脚本成23个测试SQL
- 运行
  - 执行23个测试脚本，记录运行时长

# 测试工具 – tpch

---

- 基准 - tpch
  - warehouse = 1000
  - 单进程
  - tpch侧重OLAP模型，而MySQL并不适合OLAP，因此warehouse设定较小

# 测试工具 – tpch

---

- 测试用例

生成测试数据：`./dbgen -s 1000`

执行OLAP查询：`time mysql -f tpch < ./queries/tpch_${NN}.sql`

`-s warehouse/仓库数`



# 测试工具 – tpch

---

- 不足
  - 初始化比较麻烦
  - 部分索引不合理
  - MySQL本身不擅长做OLAP，测试模式有局限

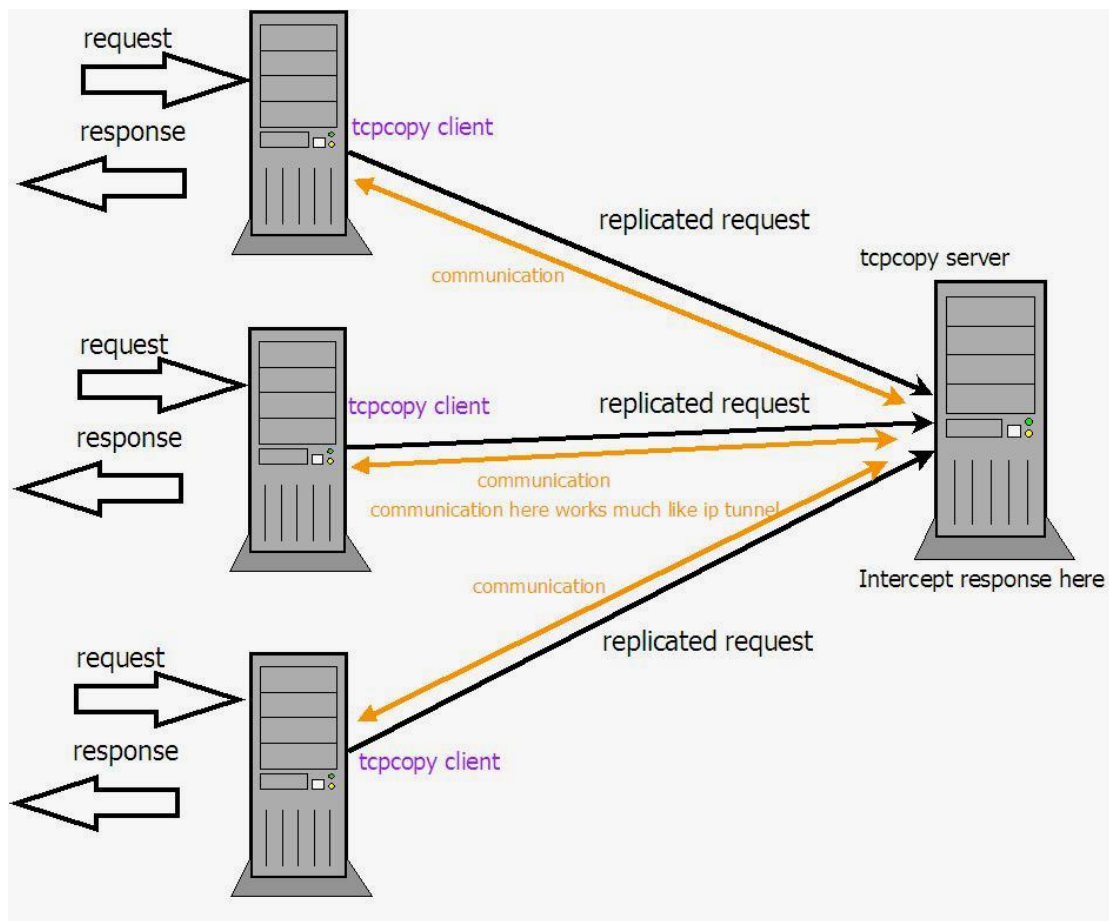
# 测试工具 – tcpcopy

---

- 安装
  - 项目地址：<https://github.com/wangbin579/tcpcopy>
  - 下载，执行 `sh autogen.sh` 后直接编译安装即可
  - 按照文档配置server和client端，即可启动压测

# 测试工具 - tcpcopy

- 测试用例



# 测试工具 – tcpcopy

---

- 不足

- 测试MySQL时，测试机需要开启 skip-grant-tables，否则无法正常进行，因为MySQL需要进行认证；在线服务器无需调整
- 测试过程中不能执行flush privileges，否则上述选项会失效
- 每次启用tcpcopy时，都需要重启mysqld，不能在线直接应用，否则不能转发包
- 不支持prepare语义
- 可能需要调整部分内核tcp参数，避免出现queue dropped
- 无法100%保证生产环境与测试环境数据一致性

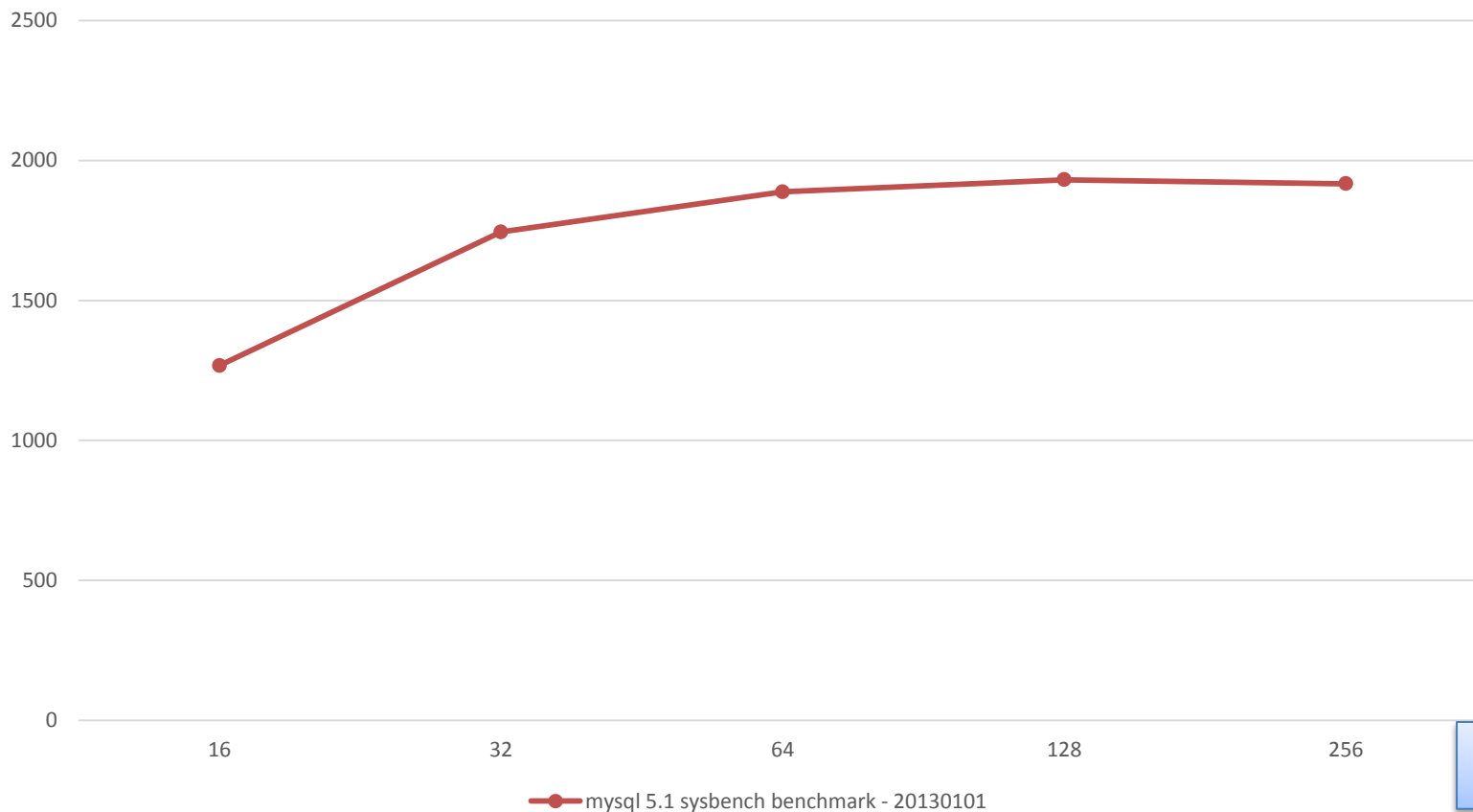
# 测试报告 – sysbench

线程数	第一轮 (tps)	第二轮 (tps)	第三轮 (tps)	均值 (tps)
16	1266.867	1267.683	1267.275	1267.275
32	1744.833	1744.350	1744.592	1744.592
64	1891.250	1884.600	1887.925	1887.925
128	1938.483	1923.983	1931.233	1931.233
256	1920.350	1913.017	1916.684	1916.684

# 测试报告 – sysbench

TPS

mysql 5.1 sysbench benchmark - 20130101



threads

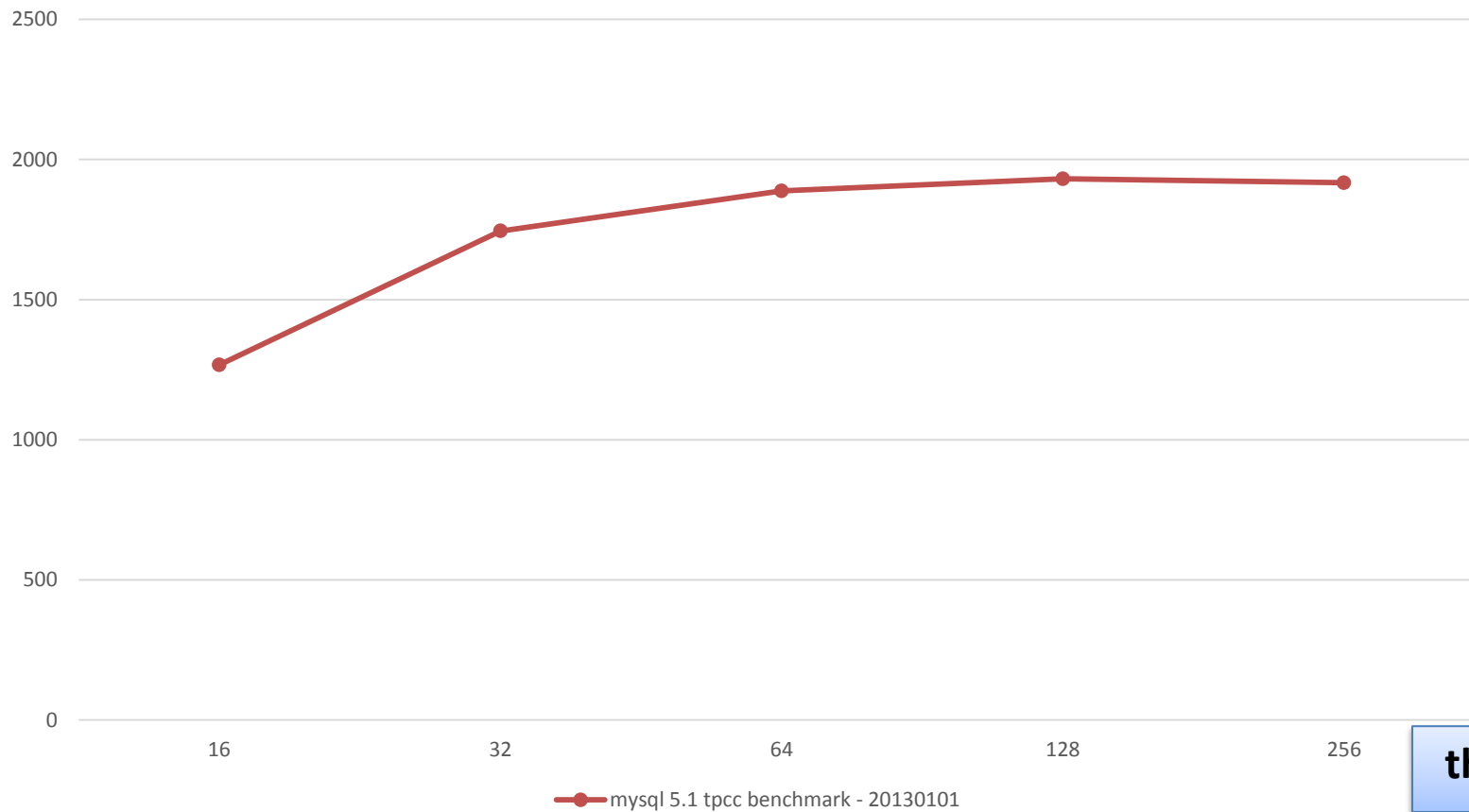
# 测试报告 – tpcc mysql

线程数	第一轮 (Tpms)	第二轮 (Tpms)	第三轮 (Tpms)	均值 (Tpms)
16	1266.867	1267.683	1267.275	1267.275
32	1744.833	1744.350	1744.592	1744.592
64	1891.250	1884.600	1887.925	1887.925
128	1938.483	1923.983	1931.233	1931.233
256	1920.350	1913.017	1916.684	1916.684

# 测试报告 – tpcc mysql

TpmC

mysql 5.1 tpcc benchmark - 20130101



threads



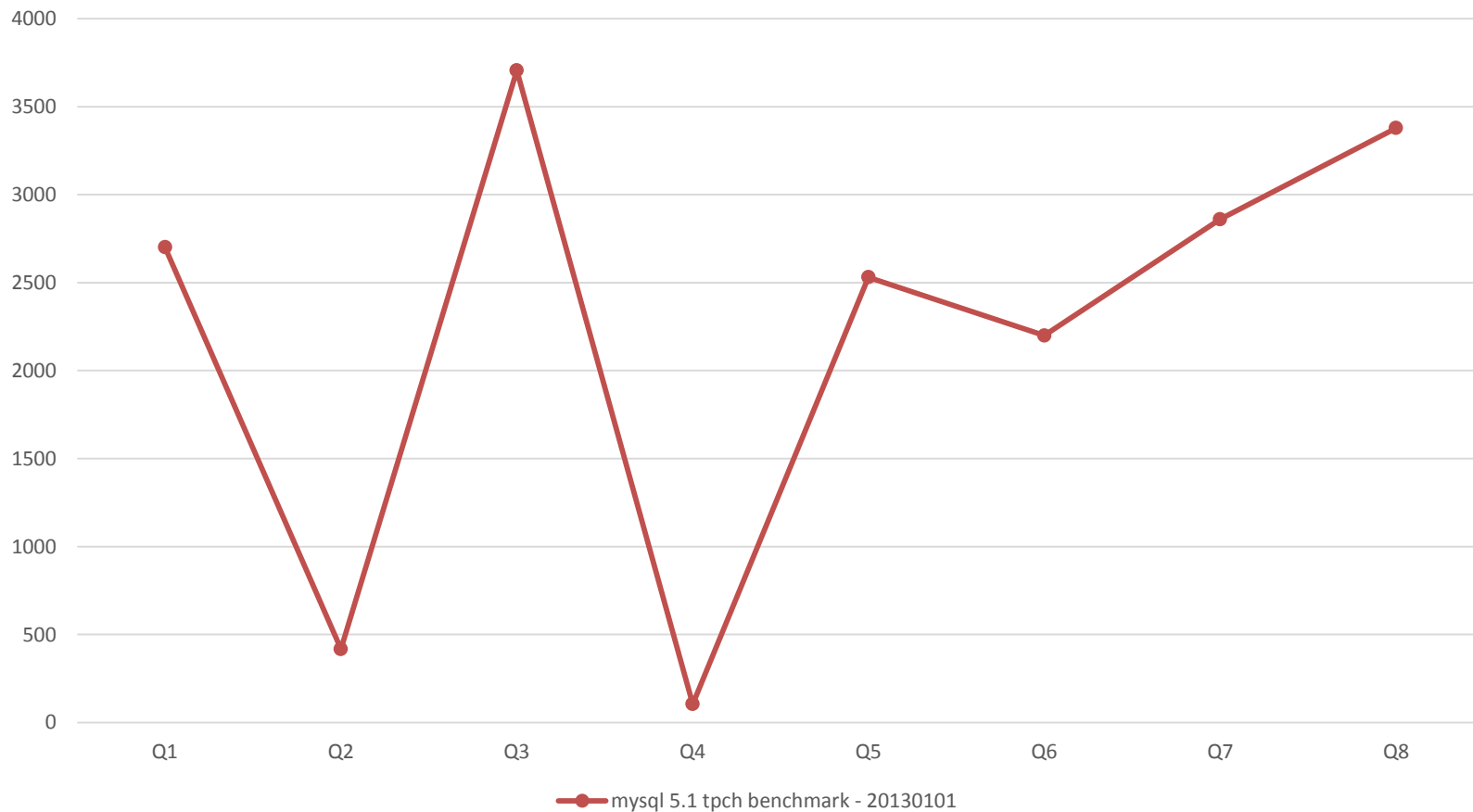
# 测试报告 – tpch

查询	第一轮	第二轮	第三轮	均值 (s)
Q1	2701.215	2697.523	2704.187	2700.975
Q2	416.314	418.774	417.474	417.521
Q3	3667.766	3755.572	3692.900	3705.413
Q4	105.804	105.493	102.411	104.569
Q5	2525.357	2536.164	2530.909	2530.810
Q6	2202.317	2194.628	2201.490	2199.478
Q7	2829.224	2860.434	2890.436	2860.031
Q8	3335.373	3383.118	3417.130	3378.540

# 测试报告 – tpch

耗时

mysql 5.1 tpch benchmark - 20130101



# 可靠性测试

---

- 模拟意外事件
  - 断电(硬件冷重启)
  - RESET(硬件热重启)
  - 阵列卡掉线
  - 磁盘掉线
  - REBOOT(系统重启)
  - 正常关闭服务(kill -TERM)
  - 异常关闭服务(kill -9)
  - 磁盘空间满
  - 删除文件
  - 破坏性修改已打开文件
  - ...

# 可靠性测试

---

- 长期极限高压
  - 持续数小时、数天、数周运行高负载计算、IO任务
  - 考验服务器在高压下的性能波动情况
  - 考验硬件设备在高压下的稳定性表现
- 模拟恶劣环境
  - 供电不稳
  - 通风冷却不好
  - 湿气大、灰尘多

# 参考

---

- <http://baike.baidu.com/view/2776305.htm>
- <https://code.google.com/p/tcpcopy>
- <http://imysql.com/2012/12/21/tpch-for-mysql-manual.html>
- <http://imysql.com/2012/08/04/tpcc-for-mysql-manual.html>
- <http://imysql.com/node/312>
- [http://imysql.com/2009/04/24/using\\_mysqlslap\\_for\\_load\\_testing.html](http://imysql.com/2009/04/24/using_mysqlslap_for_load_testing.html)
- [http://imysql.com/2008\\_07\\_25\\_innodb\\_vs\\_pbxt](http://imysql.com/2008_07_25_innodb_vs_pbxt)
- <http://imysql.com/2011/10/22/using-tcpcopy-for-request-replication-or-testing.html>

# 附件

---

- 整合sysbench测试脚本 [\[下载\]](#)
- 整合tpcc-mysql测试脚本 [\[下载\]](#)
- 整合tpch测试脚本 [\[下载\]](#)
- 汇总下载 [\[下载\]](#)
- 测试结果表格模板 [\[下载\]](#)



# Thanks