基于MySQL 8 做分布式数据库，有哪些坑？

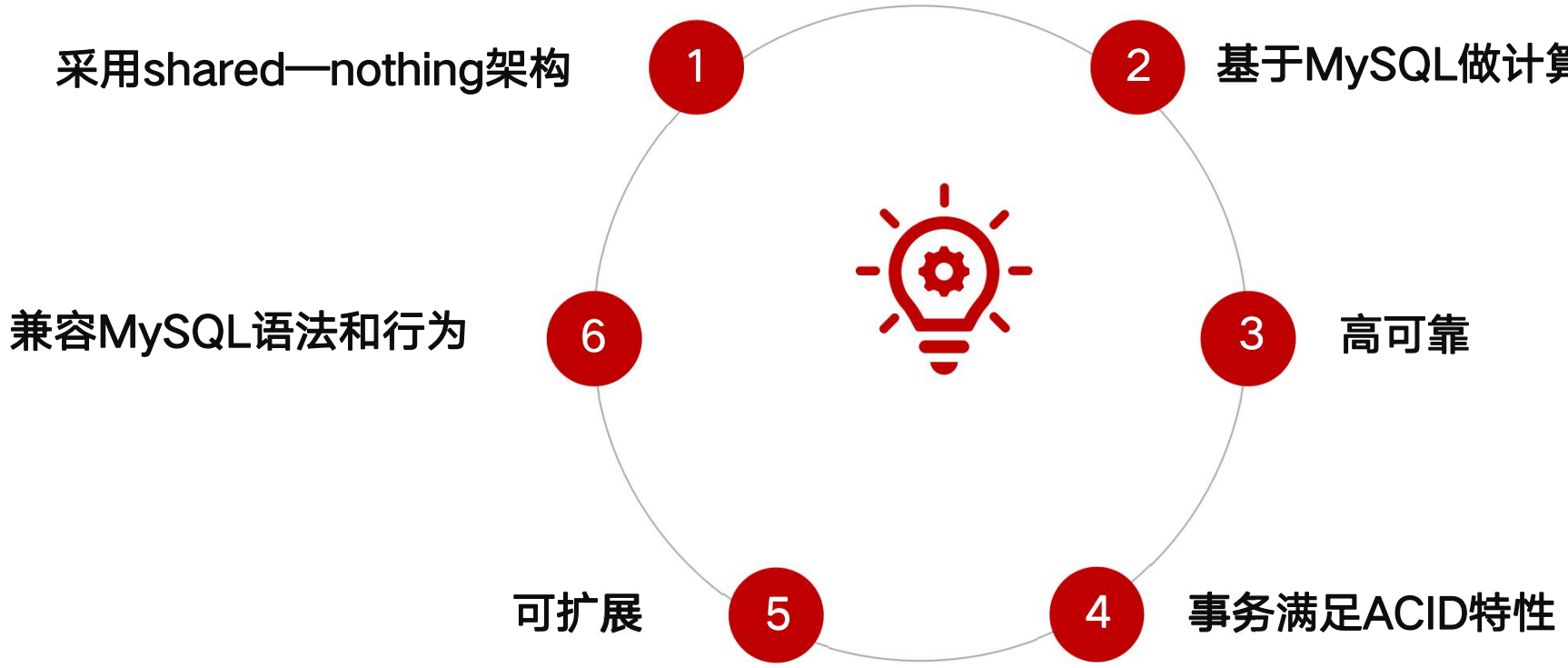
演讲人：万里数据库　王斌

01

什么是基于MySQL的分布式数据库？

GreatDB
万里数据库

基于MySQL的分布式数据库的限定条件

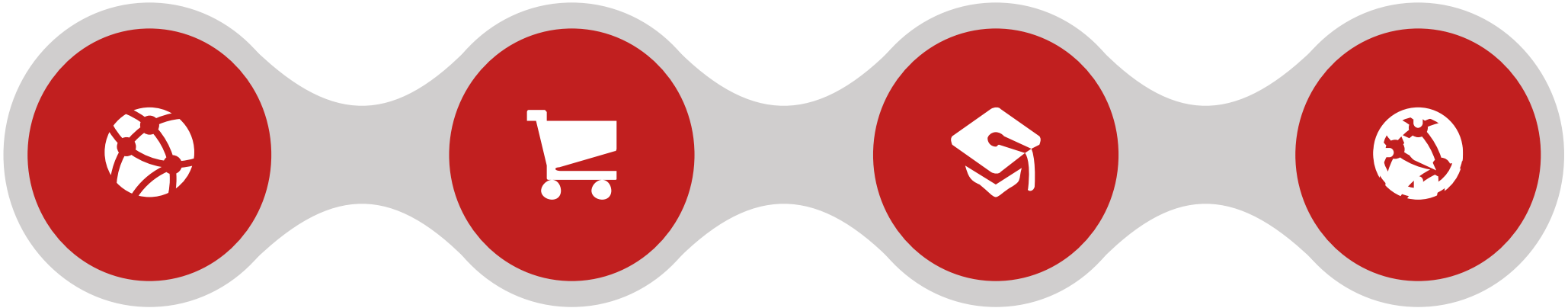


02

基于MySQL做分布式数据库，有什么优点？

GreatDB
万里数据库

基于MySQL做分布式数据库的优点



生态环境好

开发成本低

易使用，学习成本低

支持用户更多复杂特性

03

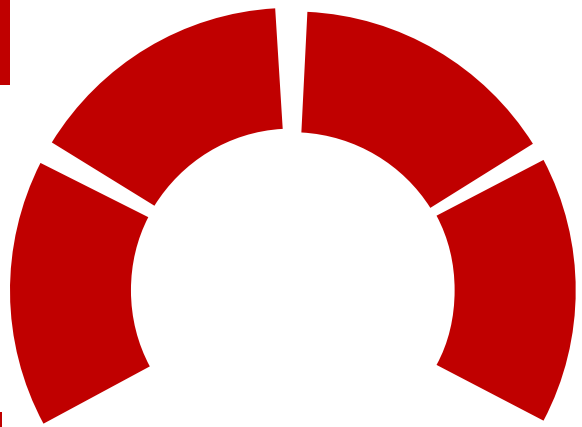
基于MySQL做分布式数据库，有哪些特点？

GreatDB
万里数据库

基于MySQL做分布式数据库，相比单机，有哪些特点？

01 时延大
需要更高并发才能追上单机

02 网络路径长



03 高可用要求高

04 网络IO超越存储IO
成为最主要的瓶颈

04

基于MySQL做分布式数据库，有哪些坑呢？

GreatDB
万里数据库

MySQL 8 有哪些坑？

Coarse-grained latch

锁调度机制不合理

默认字符集不同导致
的性能大坑

支持分布式事务的XA
代码存在大量bug

高可用机制不完善

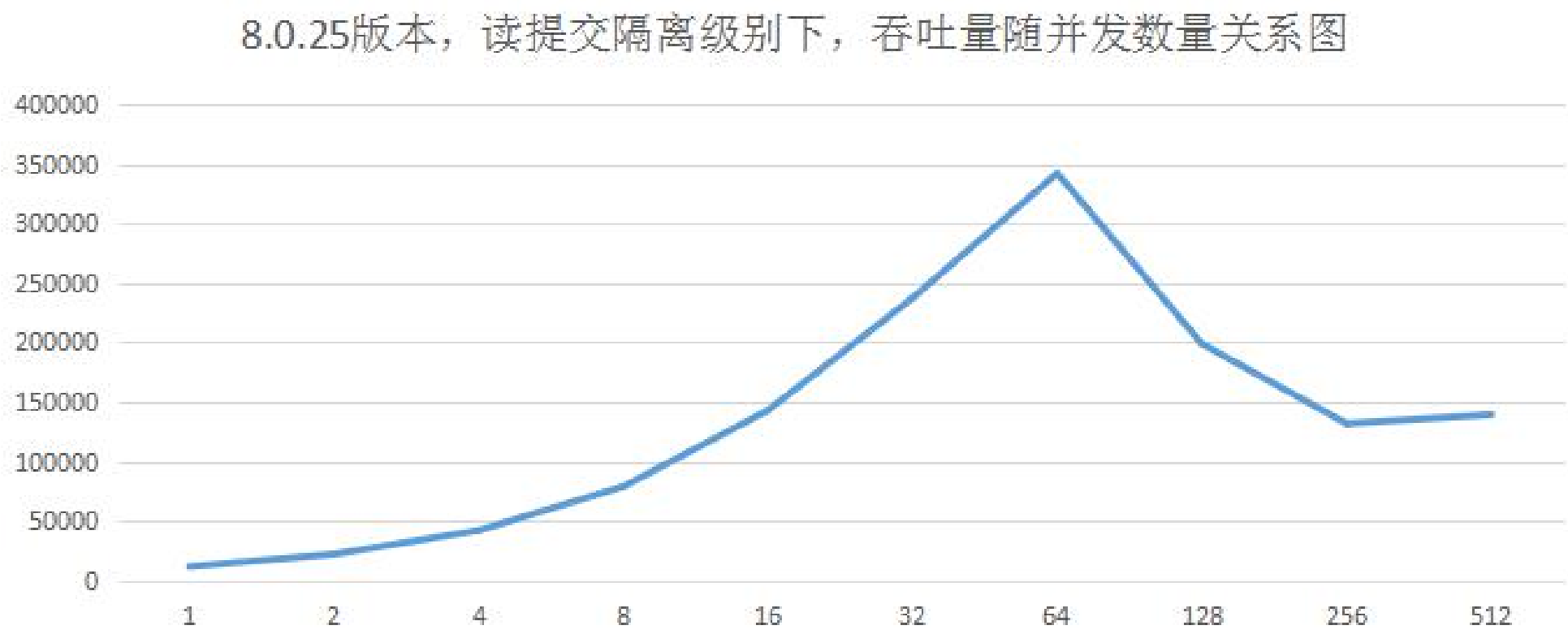
执行计划只考虑
单机

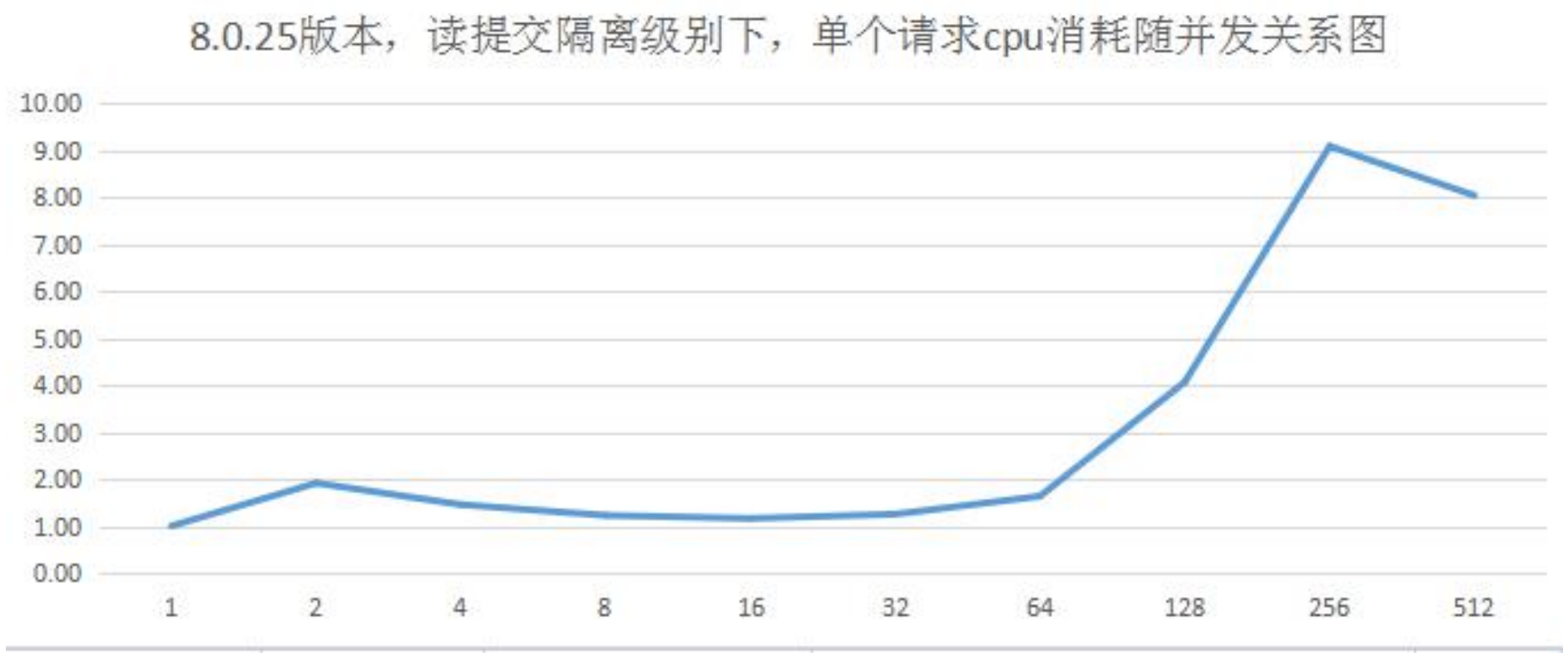
部分场景下
比5.7性能还差

Latch粒度和锁调度机制决定了MySQL并发能力

GreatDB
万里数据库

MySQL并发能力

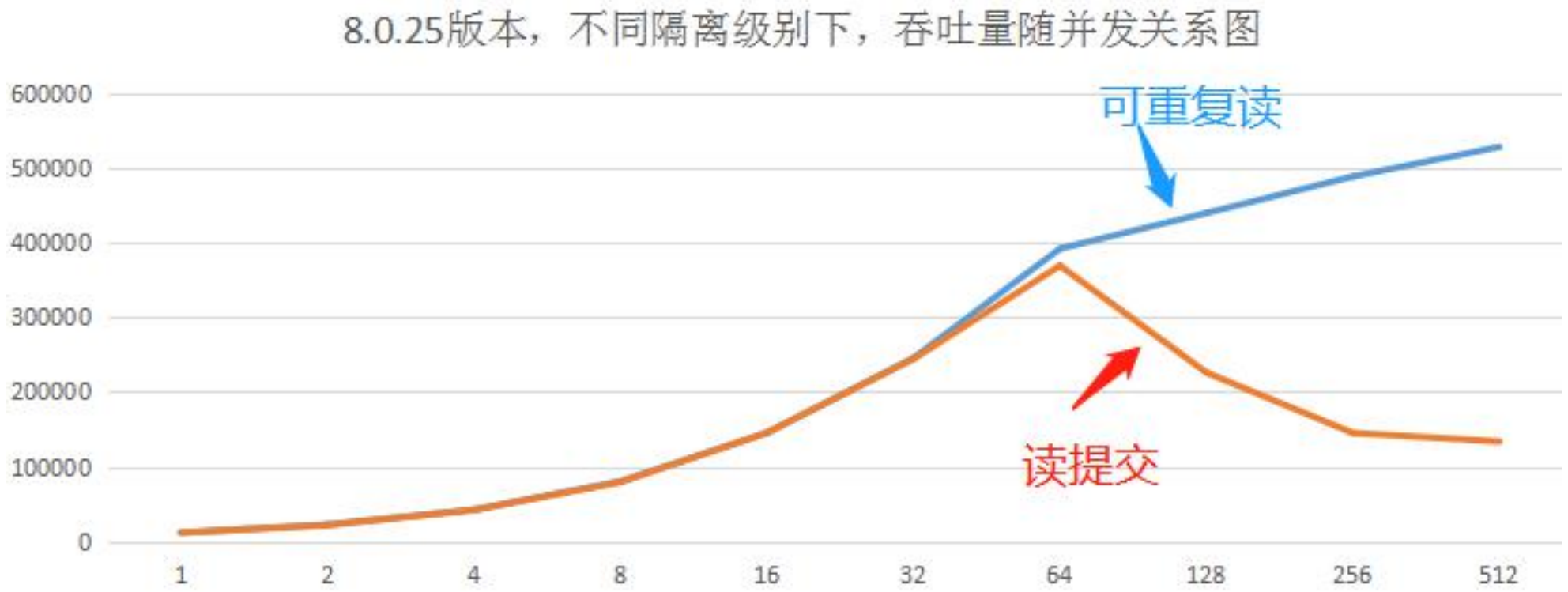

8.0.25版本，读提交隔离级别下，吞吐量随并发数量关系图
400000
350000
300000
250000
200000
150000
100000
50000
0
1
2
4
8
16
32
64
128
256
512


8.0.25版本，读提交隔离级别下，单个请求cpu消耗随并发关系图
10.00
9.00
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
1
2
4
8
16
32
64
128
256
512

主要原因是MySQL trx_sys所用的latch粒度太粗，

影响了读提交的并发扩展性

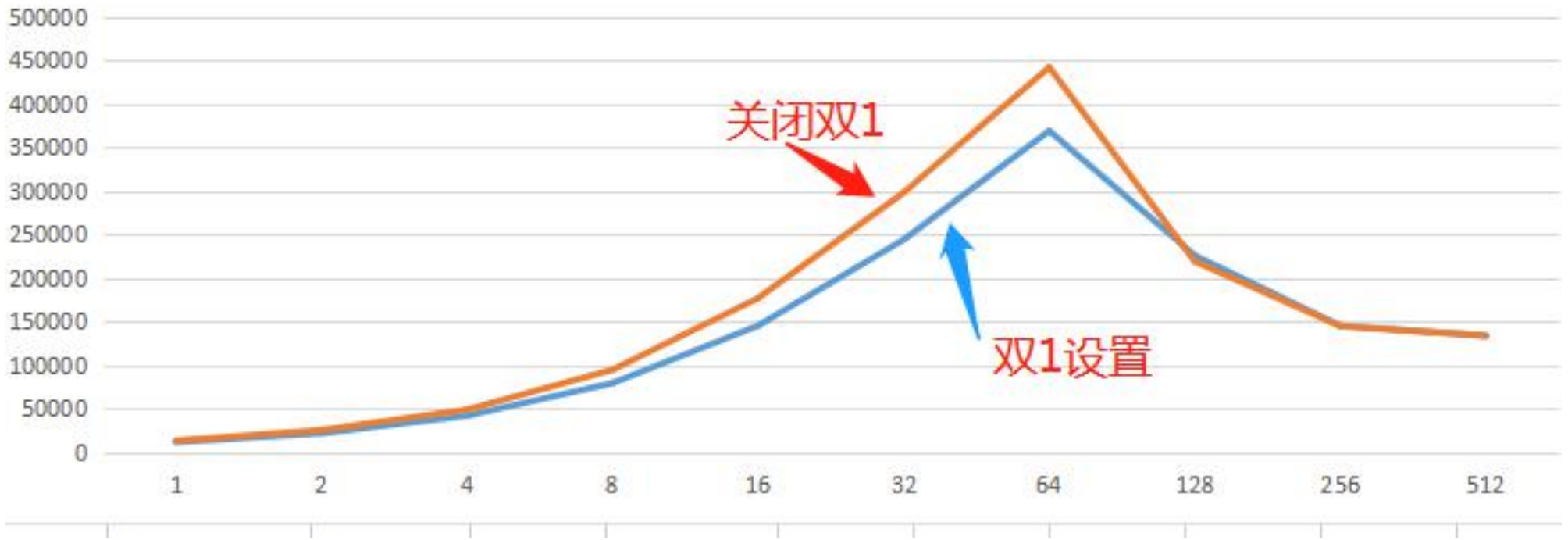
官方MySQL读提交 vs 官方MySQL可重复读


8.0.25版本，不同隔离级别下，吞吐量随并发关系图
600000
500000
400000
300000
200000
100000
0
1
2
4
8
16
32
64
128
256
512
可重复读
读提交

通过trx_sys的latch拆分+ Read view优化来解决RC扩展性问题

MySQL灵异现象

8.0.25版本，读提交隔离级别下，吞吐量随并发关系图



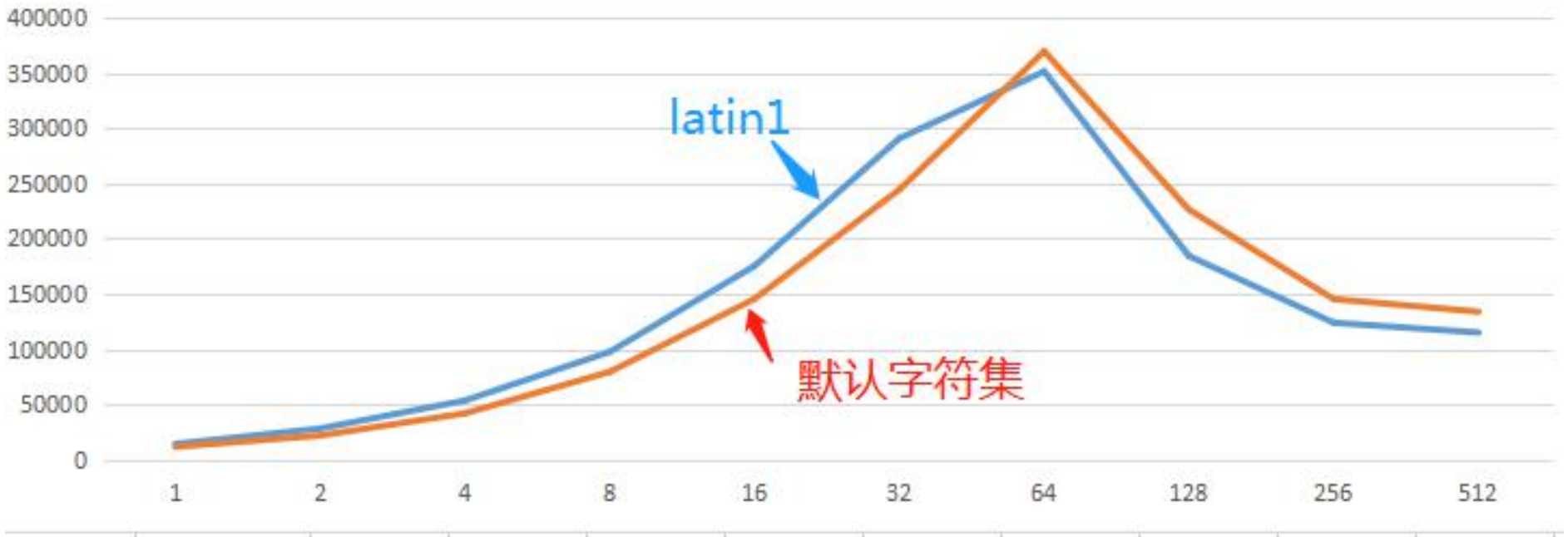
针对读提交，高并发下MySQL设置双1读写综合性能往往更好

锁调度机制不合理 + 锁竞争激烈 + Group commit高并发效率高

性能优化在复杂软件面前，往往是追求一种平衡

不同字符集带来的性能差异

8.0.25版本，latin1字符集 pk 默认字符集，读提交，吞吐量随并发关系图



MySQL XA机制是实现分布式事务的基础

GreatDB
万里数据库

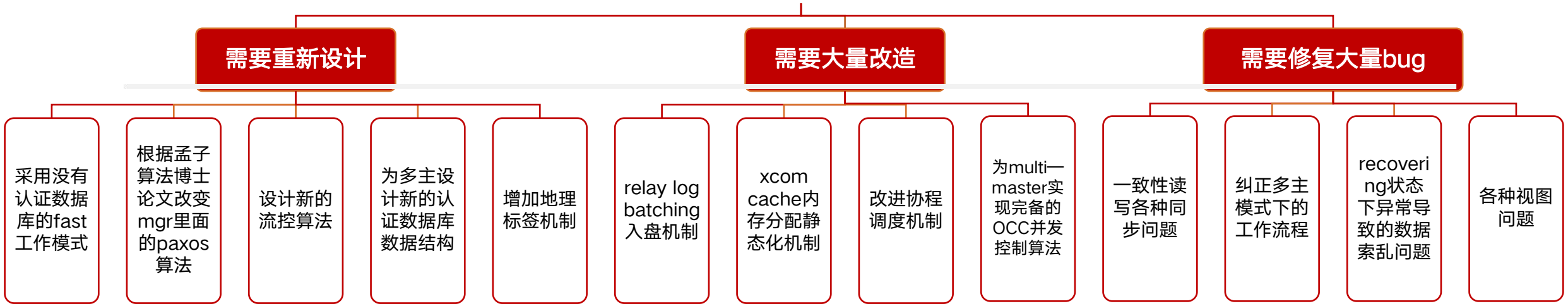
MySQL XA有哪些坑？

- 在刷xa prepare binlog之前没有写入xid到undo
- 切主后 hang死问题
- 节点被杀，且一阶段提交出现了error，没有回滚整个事务
- 涉及到 MGR xa replay的并行回放问题
- XA prepare bug related to crash recover
- 克隆（clone）后相关的重复XID问题
- XA一阶段提交崩溃导致丢数据问题
- crash safe 相关问题
  - xa commit
  - xa prepare

MySQL Group Replication高可用机制不完善

GreatDB
万里数据库

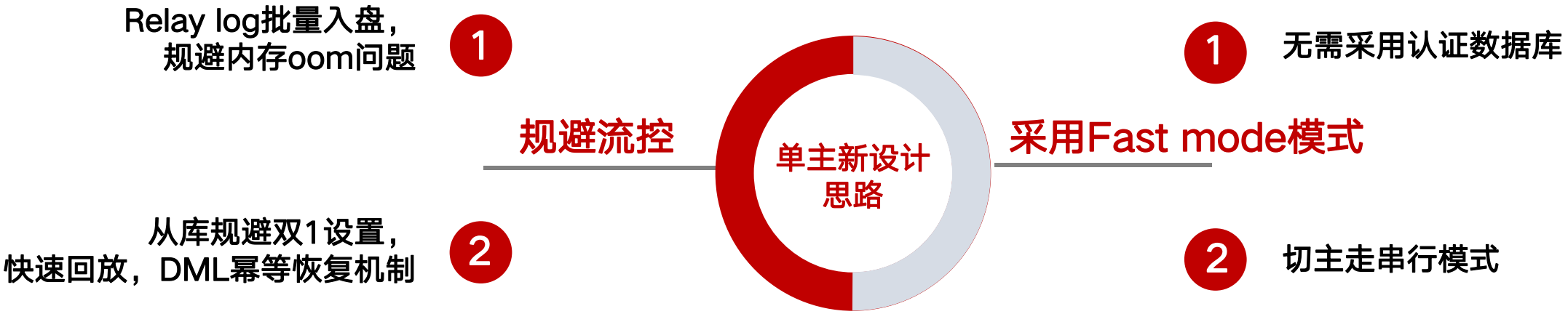
MGR完善



MGR改造样例


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

单主新设计思路


Relay log批量入盘，
规避内存oom问题
1
规避流控
从库规避双1设置，
快速回放，DML幂等恢复机制
2
单主新设计
思路
1
无需采用认证数据库
采用Fast mode模式
2
切主走串行模式

MySQL执行计划只考虑单机

分布式数据库，往往网络是主要瓶颈

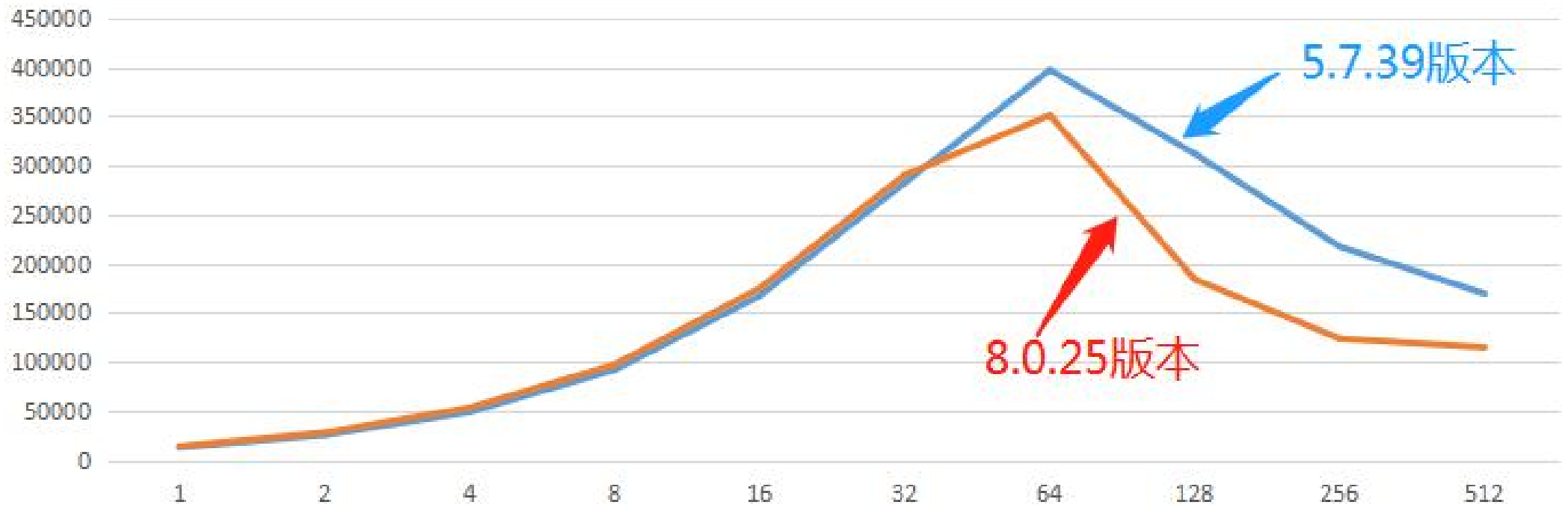
GreatDB
万里数据库

MySQL NDB都无法利用网络来优化执行计划

MySQL 8对5.7的性能改造目前并不总是很成功

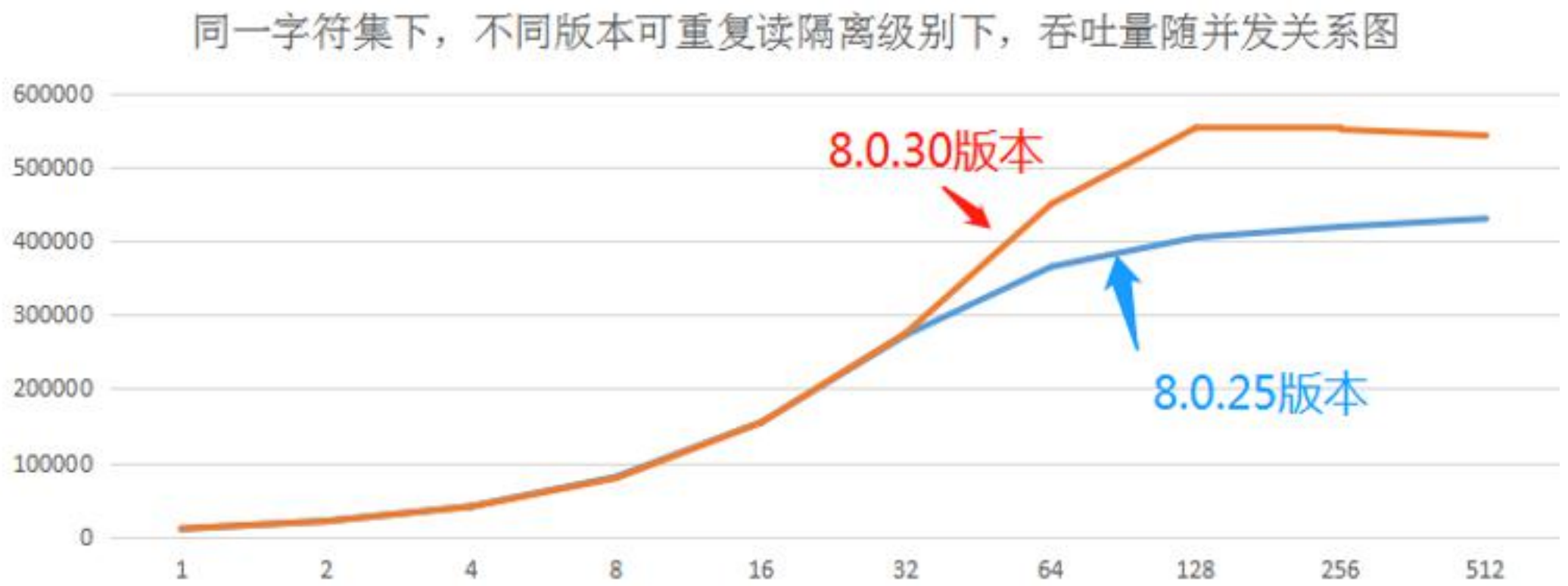
同一字符集下，不同版本可重复读隔离级别下，吞吐量随并发关系图

同一字符集下，不同版本读提交隔离级别下，吞吐量随并发关系图



根源在于MySQL 8.0.25的trx_sys所用的latch粒度太粗

最新版本在trx_sys latch进行了的一些简易拆分


同一字符集下，不同版本可重复读隔离级别下，吞吐量随并发关系图
600000
500000
400000
300000
200000
100000
0
1
2
4
8
16
32
64
128
256
512
8.0.30版本
8.0.25版本

correctness issues. One of the issues addressed caused an XA transaction to be described incorrectly as "recovered", which occurred when a client session disconnected from an XA transaction after `XA PREPARE`. (Bug #31870582)

- **InnoDB:** TempTable debug assertion code for an `Indexed_cells` member function (`cell_from_mysql_buf_index_read()`) did not account for non-nullable columns with zero length. (Bug #31091089)

- **InnoDB:** Using the InnoDB memcached plugin, attempting to retrieve multiple values in a single get command returned an incorrect value. (Bug #29675958, Bug #95032)

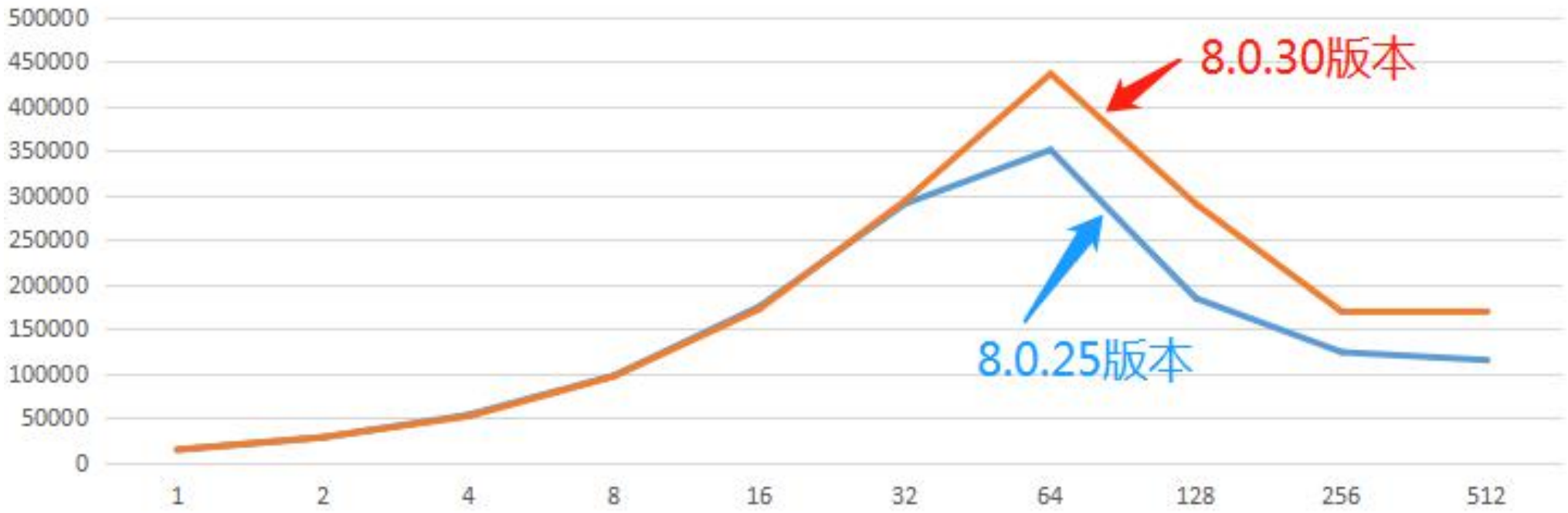
- **InnoDB:** The `trx_sys_t::serialisation_mutex` was introduced to reduce contention on the on the `trx_sys_t::mutex`. The new mutex protects the `trx_sys_t::serialisation_list` when a transaction number is assigned, which was previously protected by the `trx_sys_t::mutex`.

  Thanks to Zhai Weixiang for the contribution. (Bug #27933068, Bug #90643)

明显改善了可重复读的高并发能力

- **Partitioning:** When a table was partitioned by `TIMESTAMP` and a timestamp literal with a time zone offset was used in the `WHERE` clause of a `SELECT` statement, it was possible for a partition to be omitted from the result set.

同一字符集下，不同版本读提交隔离级别下，吞吐量随并发关系图



sql/8.0/en/news-8-0-26.html

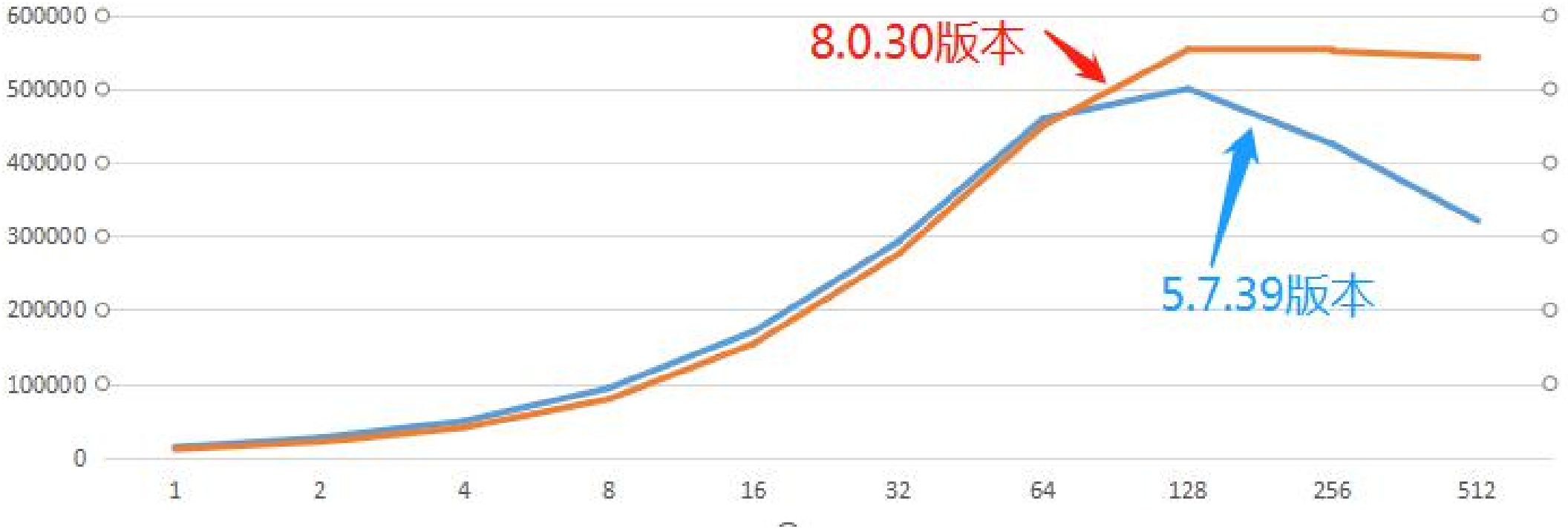
was not optimal for low-concurrency workloads. (Bug #32880577)

- **InnoDB:** A string value setting for the `innodb_redo_log_encrypt` variable was not handled properly. (Bug #32851525)
- **InnoDB:** Read-write transaction set (`trx_sys->rw_trx_set`) shards, each with a dedicated mutex, were introduced to alleviate transaction system mutex (`trx_sys->mutex`) contention caused by transaction set insertions and removals. Related enhancements include moving transaction set modifiers to less critical locations, eliminating heap allocation inside of the `TrxUndoRsegs` constructor, converting transaction state (`trx->state`) and transaction start time (`trx->start_time`) fields to `std::atomic` fields, and new assertion code to validate threads that operate on transactions. (Bug #32832196)
- **InnoDB:** Record buffer logic for the `InnoDB` memcached `GET` command was revised. (Bug #32828352)
- **InnoDB:** The `ut_list` base member in the `InnoDB` sources now locates list nodes using the element portion of the list type rather than storing a member pointer in the base node of a list at runtime, which waisted resources. The patch also includes other `ut_list` related code improvements. (Bug #32820458)

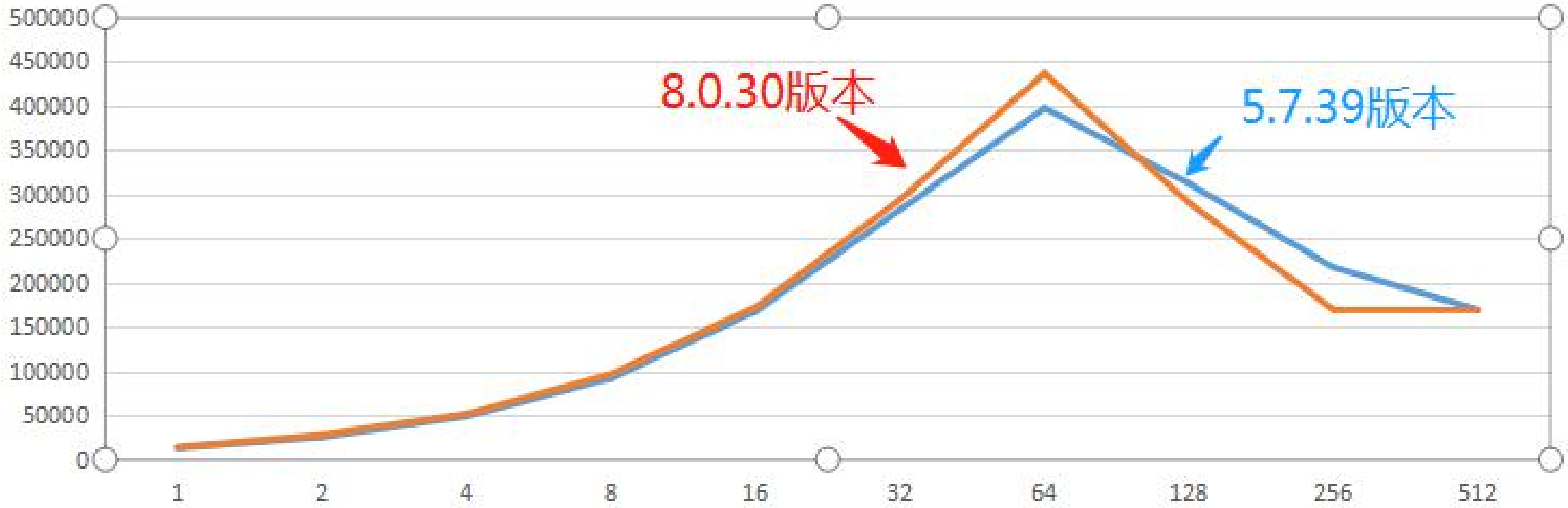
小幅提升了读提交的高并发能力

MySQL 5.7.39 vs MySQL 8.0.30

同一字符集下，不同版本可重复读隔离级别下，吞吐量随并发关系图

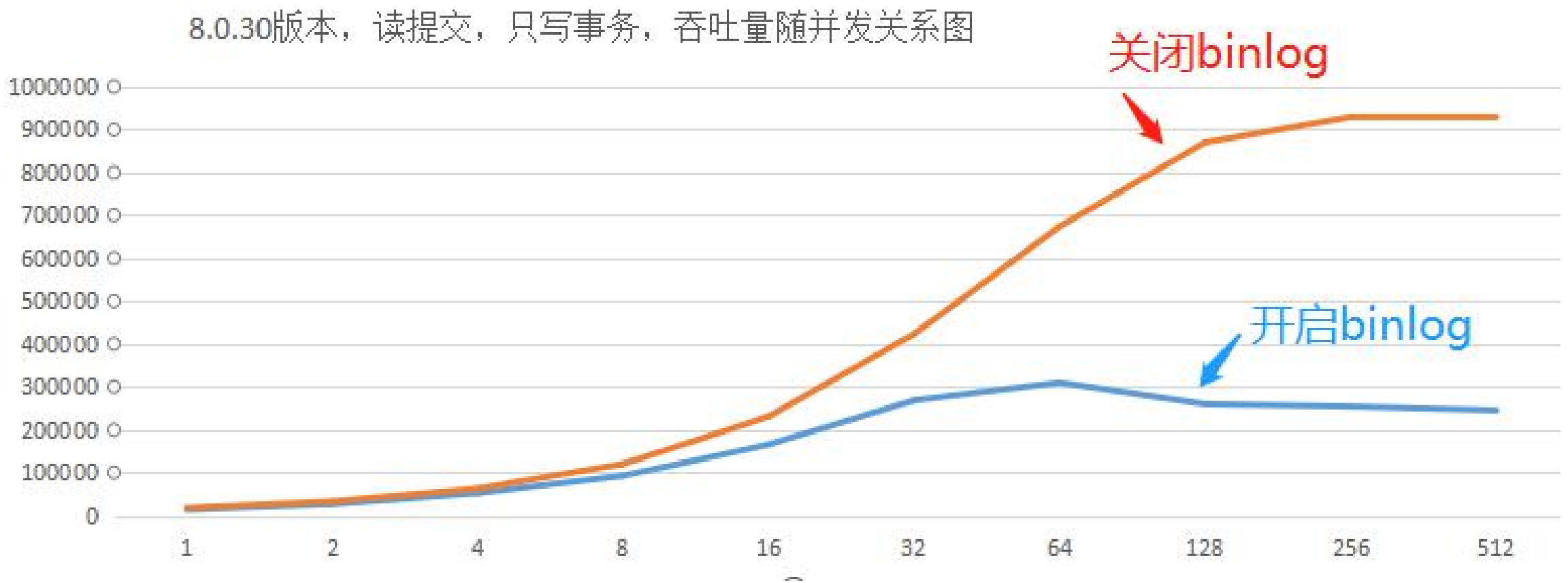


同一字符集下，不同版本读提交隔离级别下，吞吐量随并发关系图



trx_sys所利用的latch仍然可以进一步拆分，但**难度越来越大**

调度机制在MySQL 8多个地方仍然有巨大的优化空间


8.0.30版本，读提交，只写事务，吞吐量随并发关系图
关闭binlog
开启binlog
1000000
900000
800000
700000
600000
500000
400000
300000
200000
100000
0
1
2
4
8
16
32
64
128
256
512

困难是暂时的，前途是光明的

GreatDB
万里数据库

开源项目 > 数据库相关 > 数据库服务

GVP 万里数据库 / GreatSQL

Watch 49 | Star 410 | Fork 64

代码 | Issues 6 | Pull Requests 1 | Wiki | 统计 | 流水线 | 服务

全部 | 搜索 Issue | 看板 | 里程碑 | + 新建 Issue

全部 **开启的** 5 | 进行中 1 | 已完成 6 | 已拒绝 0 | 创建者 | 负责人 | 标签 | 项目 | 里程碑 | 优先级 | 排序