
数据技术嘉年华
// Data Technology Carnival
开源 · 融合 · 数智化 — 引领数据技术发展 释放数据要素价值
中国DBA联盟
All China DBA Union
墨天轮

数据技术嘉年华

GreatSQL社区
做中国广受欢迎的开源数据库

演讲人：娄帅

中国DBA联盟 All China DBA Union

墨天轮

目录

CONTENTS

GreatSQL简介

GreatSQL是什么?

由**万里数据库**主导的**开源MySQL分支**

为什么要做GreatSQL?

- MySQL is The world's most popular open source database
- But

Steinar H. Gunderson

Sun, 05 Dec 2021 - Leaving MySQL

Today was my last day at Oracle, and thus also in the MySQL team.

When a decision comes to switch workplaces, there's always the question of "why", but that question always has multiple answers, and perhaps the simplest one is that I found another opportunity, and and as a whole, it was obvious it was time to move on when that arrived.

But it doesn't really explain why I did go looking for that somewhere else in the first place. The reasons for that are again complex, and it's not possible to reduce to a single thing. But nevertheless, let me point out something that I've been saying both internally and externally for the last five years (although never on a stage—which explains why I've been staying away from stages talking about MySQL): *MySQL is a pretty poor database, and you should strongly consider using Postgres instead.*[1]

stackoverflow About Products For Teams Search...

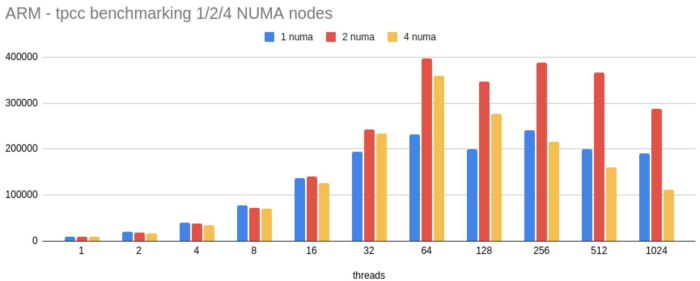
Home

PUBLIC

Questions

How do I create a sequence in MySQL?

Asked 8 years ago Modified 2 months ago Viewed 201k times



Oracle ✓
@Oracle

Replying to @FedorovMykhailo and @SAP

On behalf of Oracle's 150,000 employees around the world and in support of both the elected government o Ukraine and for the people of Ukraine, Oracle Corporation has already suspended all operations in the Russian Federation.

为什么是万里数据库?

01 中国第一个MySQL认证的金牌合作伙伴

02 中国第一个MySQL研发中心

03 中国第一个MySQL教育中心

04 中国第一个MySQL商业案例

01 修复NDB Cluster bug > 100

02 修复Replication bug > 300

03 完成NDB Cluster 新功能 (WorkLog) > 10

04 完成Replication 新功能 (WorkLog) > 20

05 完成MySQL 5.1 中文参考手册翻译

ID#	Date	Updated	Type	Status	Sev	Version	OS	CPU	Summary
108190	2022-08-18 13:02	2022-09-05 2:52	Connector / ODBC	Analyzing (52 days)	S1	8.0.30	Any	x86	ODBC SQLGetData return wrong value
107969	2022-07-26 3:48	2022-07-26 5:29	MySQL Server: Optimizer	Verified (93 days)	S2	8.0.29	Any	Any	wrong result when search binary columns
107635	2022-06-22 14:47	2022-07-20 15:43	MySQL Server: Group Replication	Closed (104 days)	S1	8.0.*	Any	Any	event scheduler cause error on group replication
107559	2022-06-14 8:45	2022-06-14 12:23	MySQL Server: Optimizer	Not a Bug (135 days)	S3		Any	Any	Why Switch_ref_item_slice in TemptableAggregateIterator::Init
104629	2021-08-16 3:08	2021-08-16 7:09	MySQL Server: Optimizer	Verified (437 days)	S1	8.0.25,5.7.35, 8.0.26	Any	Any	wrong result when outer join prune partition tables with is null predicate
103040	2021-03-18 14:50	2021-03-19 9:46	MySQL Server: Group Replication	Verified (587 days)	S3	8.0.*	Any	Any	minor fix for DEBUG message in XCOM
100800	2020-09-10 12:51	2021-12-02 11:20	MySQL Server: Optimizer	Closed (469 days)	S2	8.0.21, 8.0.11	Any	Any	wrong result when select int column with range
100783	2020-09-09 12:29	2020-09-10 4:50	MySQL Server: Optimizer	Can't repeat (777 days)	S2	8.0.19	Any	Any	wrong result with hash join
99647	2020-05-20 13:11	2020-05-22 12:02	MySQL Server: DML	Not a Bug (888 days)	S5	8.0.*	Any	Any	call file->position when necessary in sql_delete.cc
99628	2020-05-19 9:16	2020-05-28 14:59	MySQL Server: Replication	Verified (889 days)	S2	8.0.*	Any	Any	semi sync master not handle ack packet correctly when recv packet timeout
99094	2020-03-27 4:52	2020-03-27 10:38	MySQL Server: Information schema	Verified (944 days)	S1	8.0, 8.0.19	Any	Any	coredump when install information schema plugin

GreatSQL特性

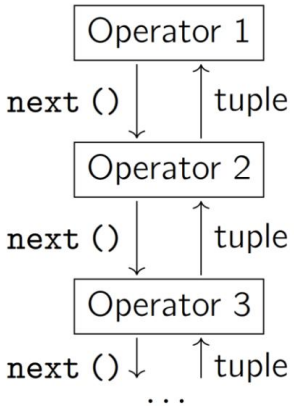
产品目标

1. 性能
2. 稳定性
3. 易用性

MySQL Executor

Iterator Model

1. pull-based: next()
2. tuple-at-one-time



WL#11785: Volcano iterator design

Affects: Server-8.0 — *Status*: Complete

Description | Requirements | High Level Architecture

Make a new API for iterating over records that is powerful enough to replace all existing record iter
API to replace READ_RECORD and the READ_RECORD-like interface in QEP_TAB (unifying the two).

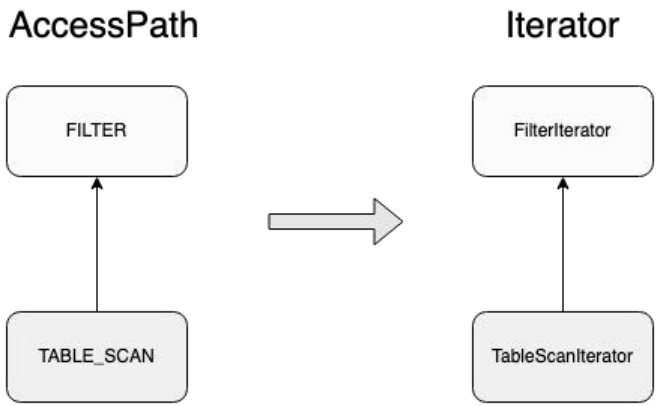
```
commit 2b6cd8d6bee1978f9e461be190dad48cd1c0255f
Author: Steinar H. Gunderson <steinar.gunderson@oracle.com>
Date:   Wed Apr 11 13:14:43 2018 +0200

    WL #11785: Volcano iterator design [patch 1/13, remove print_error]

    Remove the print_error parameter from READ_RECORD and treat it as always
    true; it's only ever set to false in sql_help.cc, and there doesn't appear
    to be a valid reason why these help functions should suppress errors.

    Change-Id: I051d99e144e2716e3c4abd5f10d95eb90a991da1
```

MySQL Executor




```
// ExecuteIteratorQuery 逻辑
m_root_iterator->Init();
for (;;) {
    int error = m_root_iterator->Read();
    query_result->send_data(thd, *fields)
}

// Executor执行堆栈
| > Query_expression::ExecuteIteratorQuery
| | > FilterIterator::Read
| | | > TableScanIterator::Read

int FilterIterator::Read() {
  for (;;) {
    int err = m_source->Read(); // 递归调用数据源
    if (err != 0) return err;
    bool matched = m_condition->val_int();
  }
}
```

MySQL局限性?

1. one-thread-per-connection: 单线程模型
2. Iterator局限性：tuple-at-one-time(数据传递没有解耦)

InnoDB Parallel Read

- 8.0.14InnoDB支持主索引并行读取: Parallel_reader
- 并行读是**计划并行**的基础

WL#11720: InnoDB: Parallel read of index

Affects: Server-8.0 — *Status*: Complete

Description | Requirements | High Level Architecture | Low Level Design

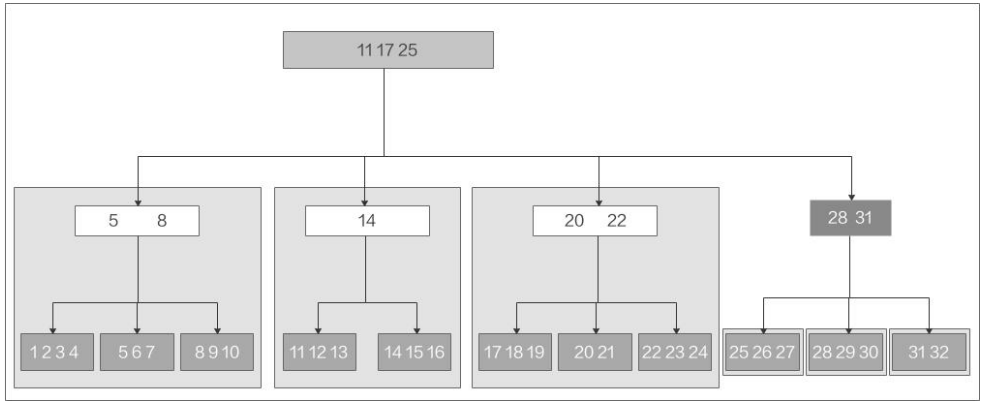
The latching protocol for Key_reader is:

1. Lock the dict_index_t::lock in SX mode. 2. Create the key range cursors for all the sub trees. 3. Release

The subtrees to scan in parallel are selected using the following algorithm:

表的切分：

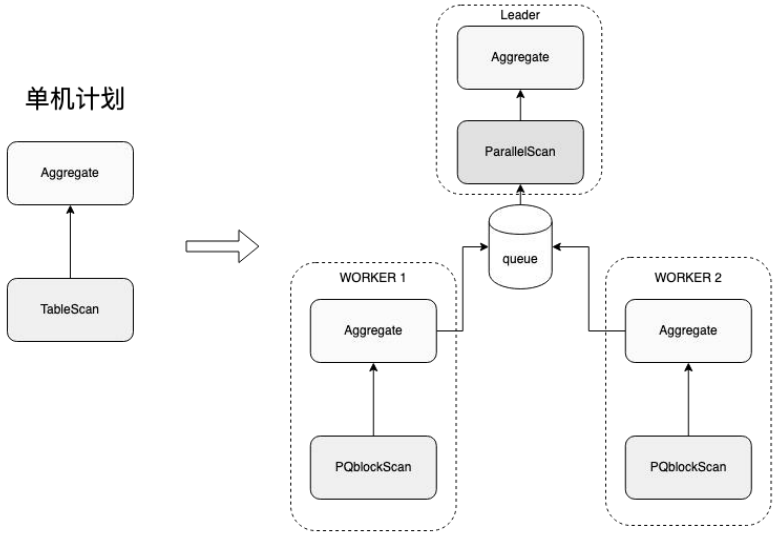
- 将扫描的数据拆分为多份。
- 负载不均衡时，可以进行第二次拆分。



Parallel Query Executor

SELECT AVG(a) from t1;

- **单线程计划 => 多线程并行计划**
  - Leader & Worker Threads
    - worker线程：并发抽取数据，执行计划
    - Leader线程：汇聚worker线程的计算结果
  - 各个线程拥有各自的计划
    - TableScan=> ParallelScan/PQBblockScan
    - 数据传递通过leader的上的队列

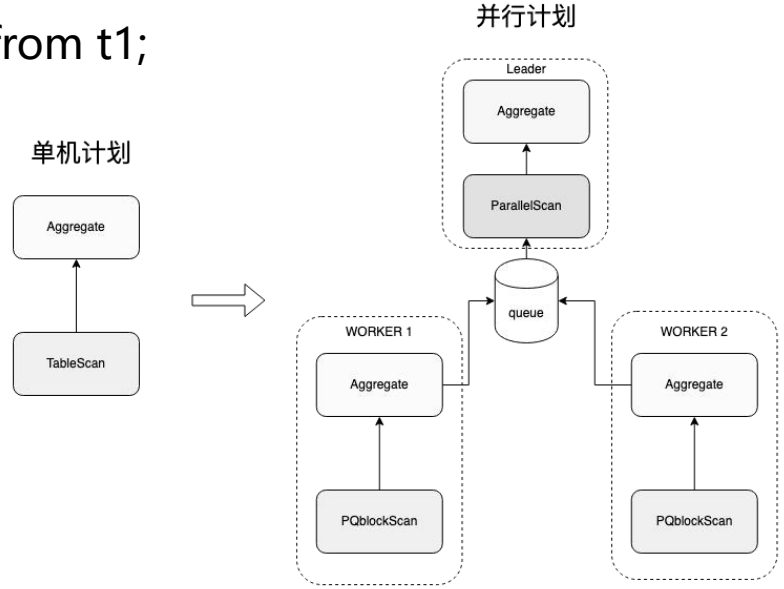



```
-> Aggregate: avg(`avg(a)`)
  -> Parallel scan on <temporary>
      -> Aggregate:
          -> PQblock scan on t1 using PRIMARY  (cost=4346.95 rows=43067)
```

Parallel Query Executor

SELECT AVG(a) from t1;

- **聚集函数AVG需要特殊处理(rebuild sum funcs)**
  - WORKER Thread
    - AGG节点传出的数据为分片数据的SUM和COUNT的值
  - LEADER Thread
    - AGG对SUM和COUNT进行分别累加

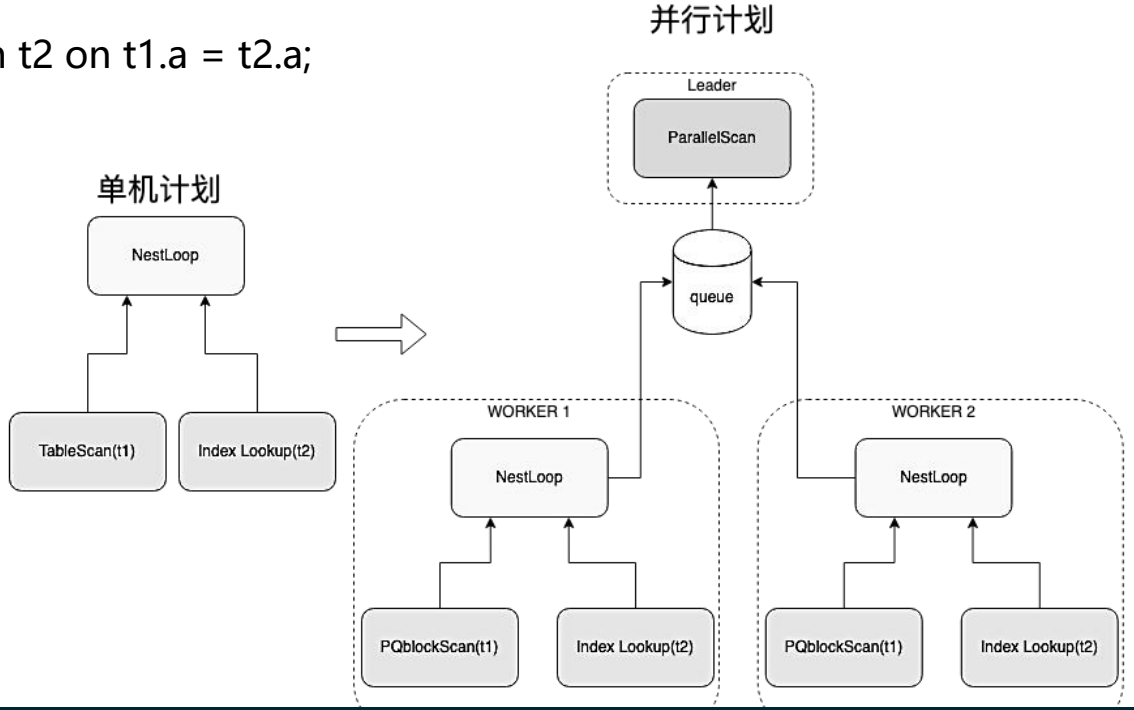



```
-> Aggregate: avg(`avg(a)`)
  -> Parallel scan on <temporary>
      -> Aggregate:
          -> PQblock scan on t1 using PRIMARY  (cost=4346.95 rows=43067)
```

Parallel Query Executor

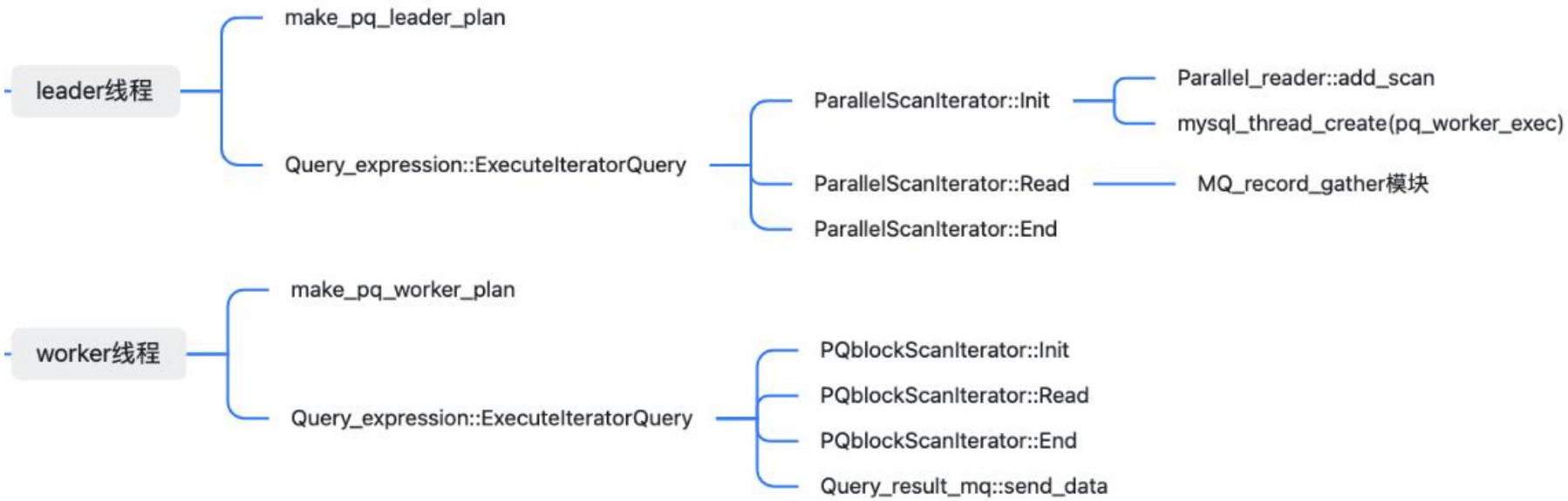
SELECT * from t1 join t2 on t1.a = t2.a;

- 只有一张表可以进行parallel read。
- TableScan的表可以转换为Parallel Scan




```
-> Parallel scan on <temporary>
  -> Nested loop inner join  (cost=1.15 rows=2)
      -> PQblock scan on t2  (cost=0.45 rows=2)
      -> Single-row index lookup on t1 using PRIMARY (a=t2.a)  (cost=0.30 rows=1)
```

Parallel Query Executor



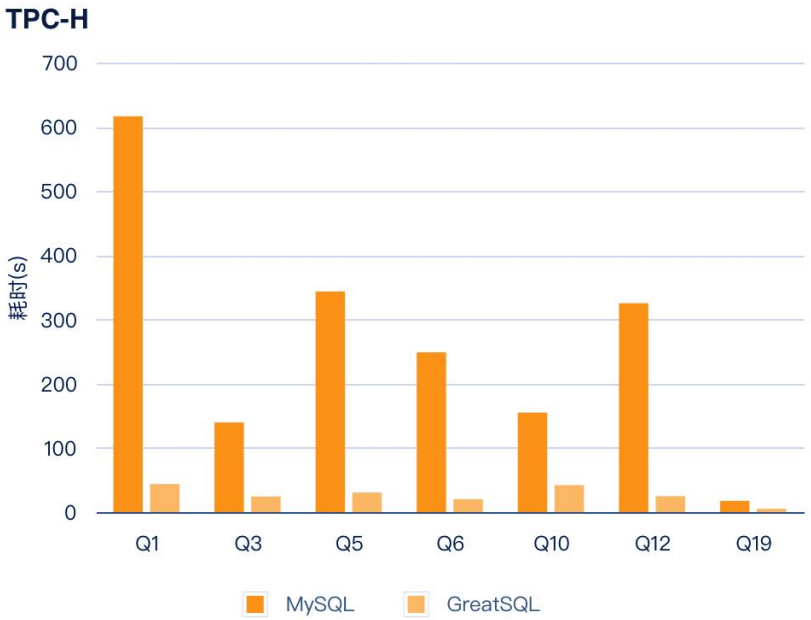
leader线程

- make_pq_leader_plan根据原始的执行计划生成leader自己的执行计划。
- init过程对并行表进行切分，worker线程循环从队列里面拿一个分片去执行。
- read过程，汇总数据，做必要的aggregate操作。

worker线程

- 调用make_pq_worker_plan生成自己的执行计划。
- 原来执行计划中可以替换的一个迭代器替换为并行的迭代器PQblockScanIterator。
- read函数与innodb存储引擎的接口交互，从innodb中获取数据，然后发送到消息队列供leader线程取用。

并行的利与弊



优势：

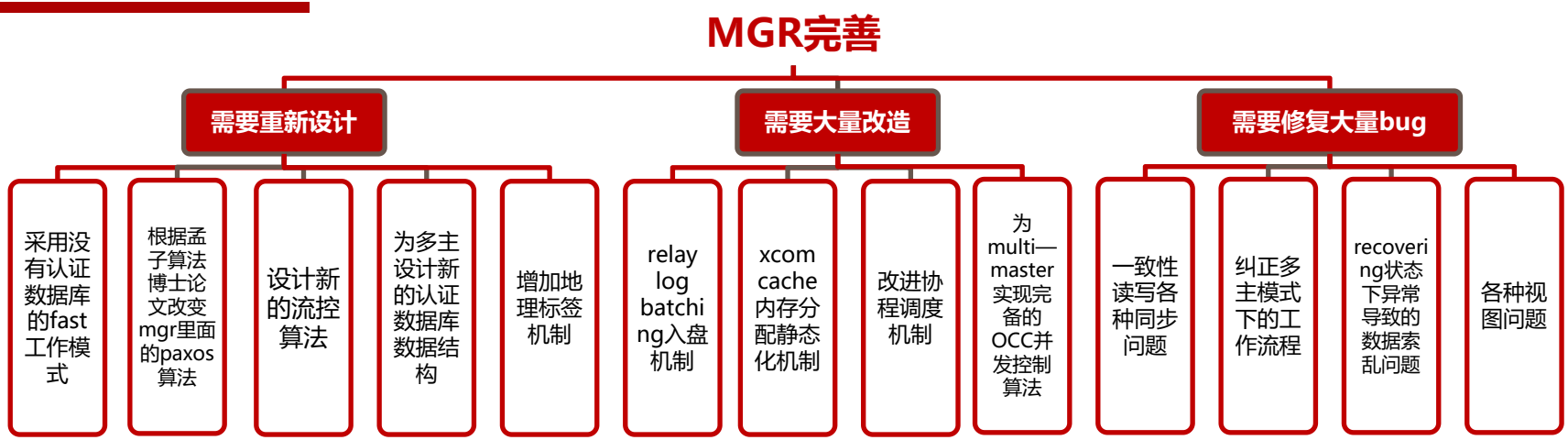
1. 充分利用多核优势，加快语句执行效率
2. Parallel Read保证了数据一致性

劣势：

1. 性能对比列存数据库，仍有差距
2. SQL支持范围有局限
3. AP TP相互影响
4. 对原生代码侵入性很强，merge痛苦

HTAP应该是什么样呢？Heatwave?

稳定性提升——MGR



更快：快速探测异常情况；流控机制更精准；

更高：高并发下，吞吐持续且稳定；

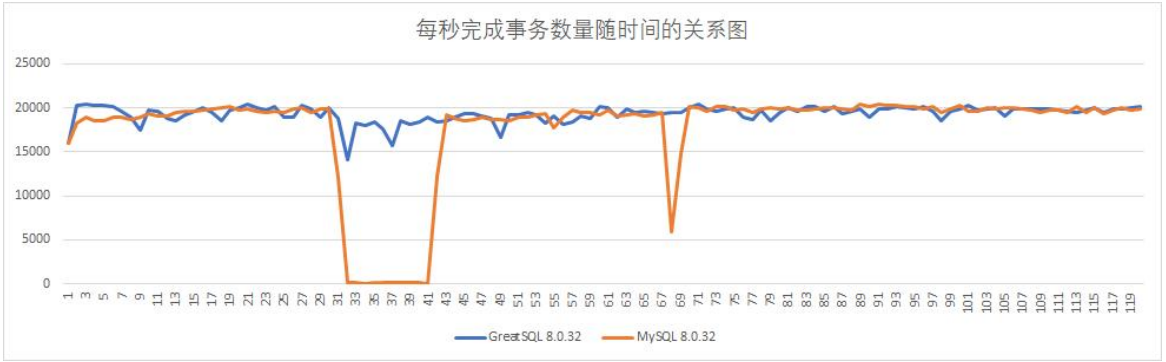
更强：更强的鲁棒性、更多的功能性。

稳定性提升——MGR

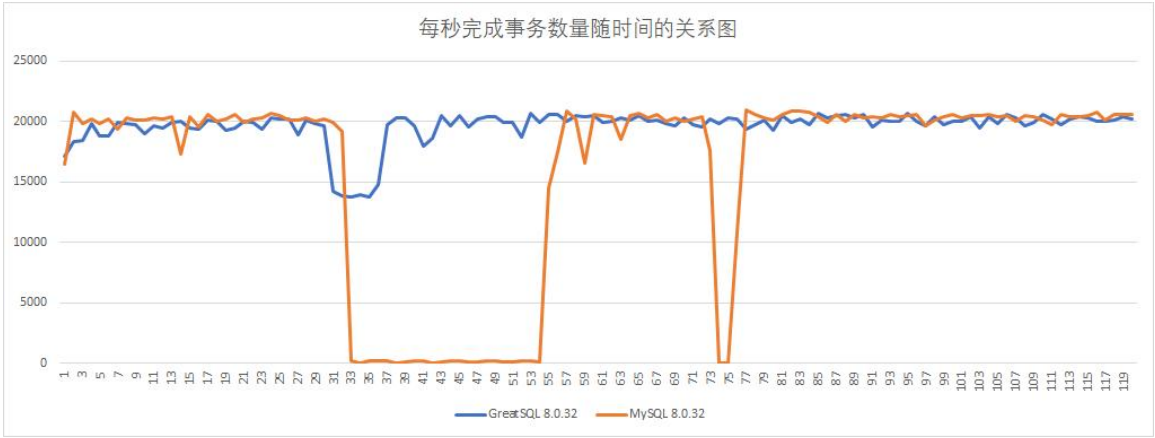

每秒订单数随时间关系图
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73 76 79 82 85 88 91 94 97 100 103 106 109 112 115 118 121
流控
流控
认证数据库在垃圾回收
认证数据库在垃圾回收
改进后的版本
官方版本

稳定性提升——MGR

加入节点



杀节点



稳定性提升——MGR

地理标签

- 解决多IDC数据同步的问题
- 事务提交时，每个IDC中至少一个节点确认事务
- 每个IDC中至少一个节点有最新事务
- group_replication_zone_id
  - 0 ~ 8

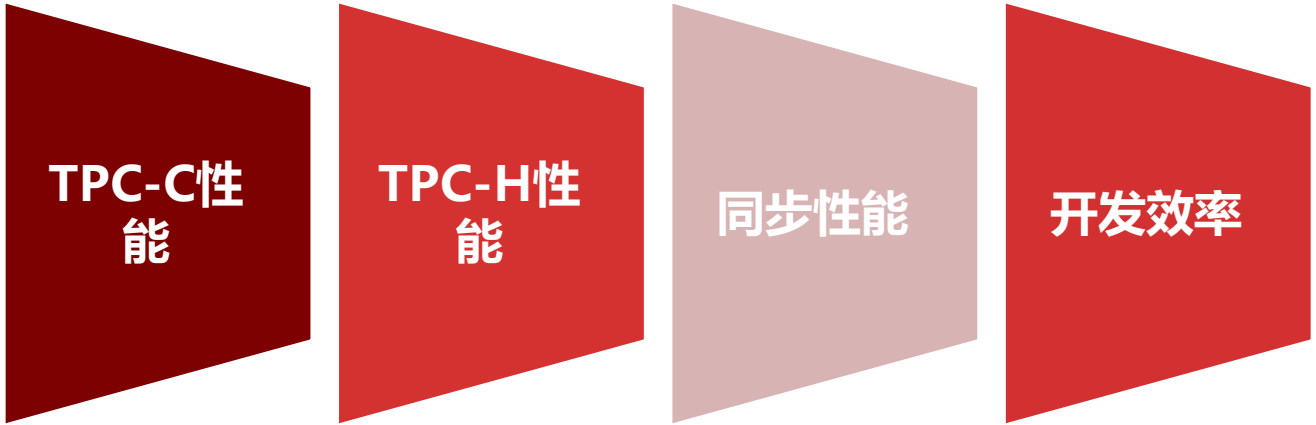
快速单主模式

- 不使用原来的事务认证模式，只需在本地认证
- 降低内存消耗，提升高并发时的MGR性能
- 特别适合单主模式且跨IDC部署场景
- group_replication_single_primary_fast_mode
  - 0
  - 1

仲裁节点

- 不存储用户数据
- 没有binlog，也不需要回放relay log
- 只参与MGR状态投票/仲裁
- 系统负载非常低，可以在一个服务器上部署多实例
- group_replication_arbitrator

GreatSQL展望

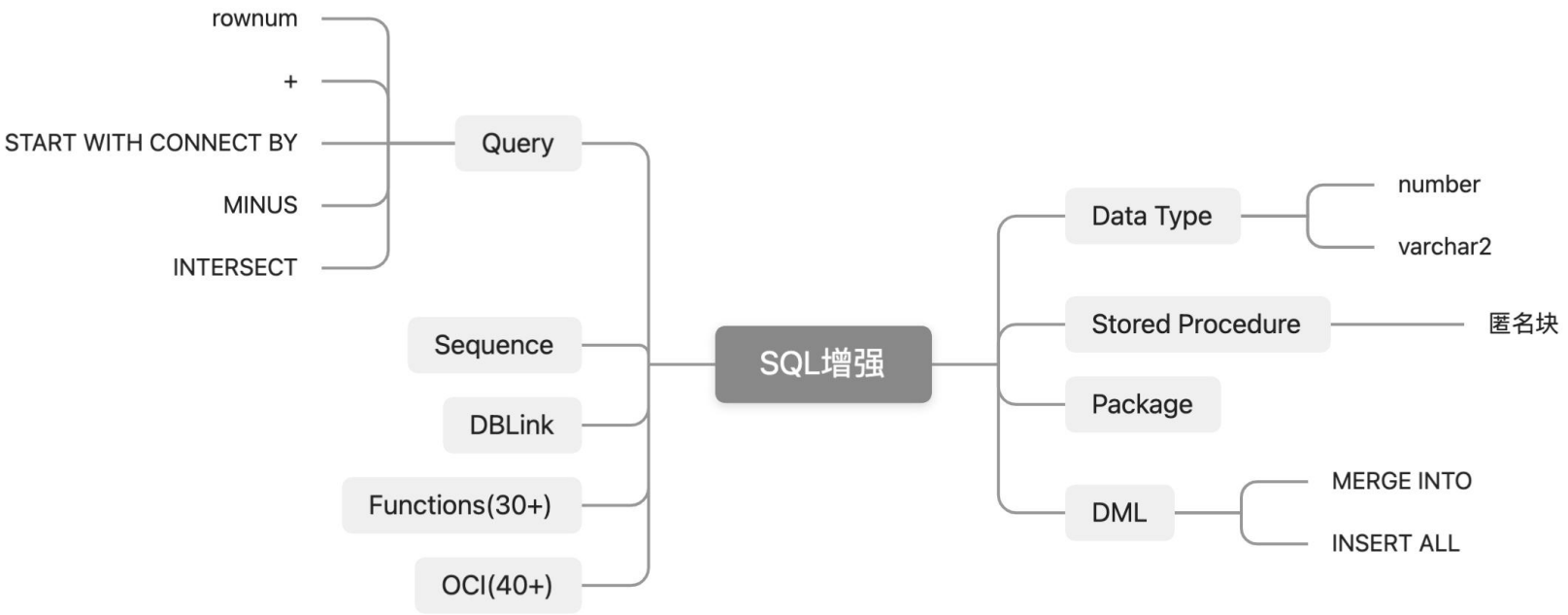


GreatSQL展望

- **性能提升**
  - AP性能提升
  - TP性能提升
  - 物理备份工具
  - 并行导入功能

- **稳定性提升**
  - 双机高可用方案
  - 三副本高可用方案

GreatSQL展望

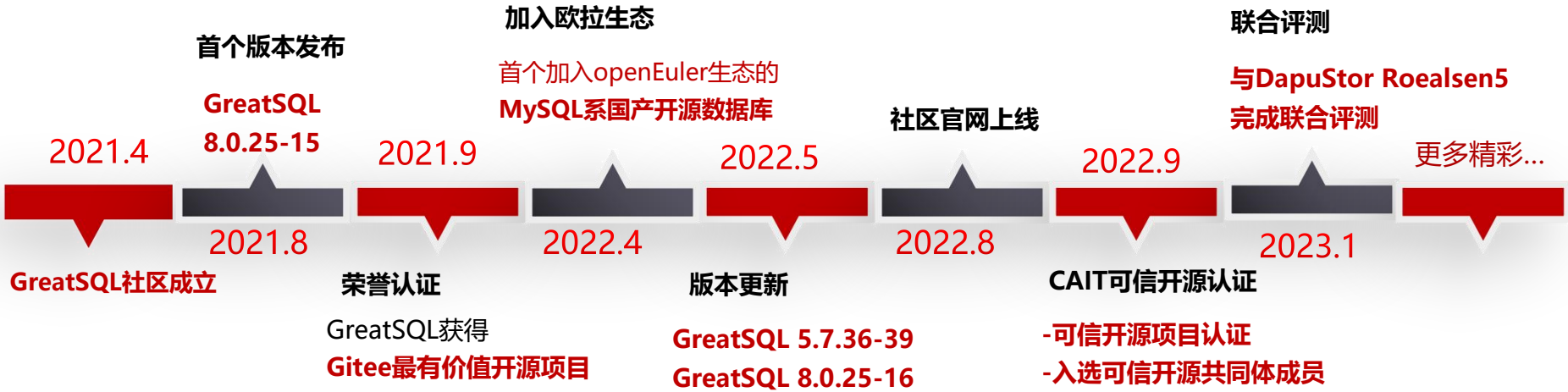

rownum
+
START WITH CONNECT BY
MINUS
INTERSECT
Query
Sequence
DBLink
Functions(30+)
OCI(40+)
SQL增强
Data Type
number
varchar2
Stored Procedure
匿名块
Package
DML
MERGE INTO
INSERT ALL

GreatSQL社区

社区简介

分类	介绍
关于GreatSQL社区	GreatSQL社区成立于2021年，是一个MySQL开源数据库社区，由**万里数据库发起**，致力于通过开放的社区合作，构建国内自主MySQL版本及开源数据库技术，推动中国开源数据库及应用生态繁荣发展。
愿景	**成为中国广受欢迎的开源数据库社区**
关于GreatSQL开源数据库	GreatSQL开源数据库**是适用于金融级应用的国内自主MySQL版本**，专注于提升MGR可靠性及性能，支持InnoDB并行查询等特性，可以作为MySQL或Percona Server的可选替换，用于线上生产环境，且完全免费并兼容MySQL或Percona Server。

GreatSQL社区发展历程



GreatSQL社区现状

01 **国内活跃的MySQL开源社区**

- 活跃的社区微信群、QQ群 & 微信公众号
- 活跃参与者超2000人
- 社区用户包括：芸擎、福富、...

02 **获得中国信通院 CAIT可信开源认证**

- 可信开源项目认证
- 入选可信开源共同体成员

03 **加入openEuler生态**

- 首个加入openEuler生态的MySQL系国产开源数据库，openEuler22.09版本正式合入GreatSQL

GreatSQL，更流畅

谢谢观看

THANKS FOR WATCHING

中国DBA联盟
All China DBA Union

墨天轮