DTCC

第十一届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2020

架构革新 高效可控

IT168.com ChinaUnix.net ITPUB

北京国际会议中心 | 2020/12/21-12/23

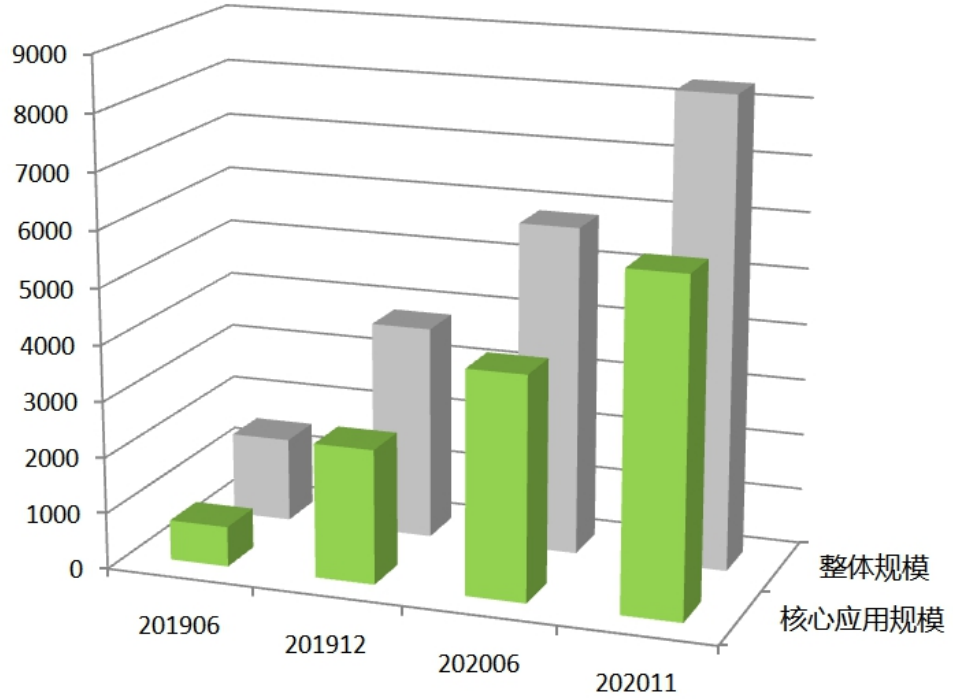
工商银行核心应用MySQL治理实践

中国工商银行软件开发中心 林镇熙

现状

快速增长

- 整体规模
- 核心应用规模



挑战及对策

处理能力

分布式体系建设：分布式服务、软负载、事务、消息、批量、缓存、数据库、对象存储、文件存储等九大运行支撑平台

成本控制

大规模云化部署，容器化比例超90%，提升资源使用效率约4-5倍

运维能力

一体化运维平台，覆盖部署、监控、高可用故障自动化切换

运行风险

核心应用大量接入，如何保障业务稳定？

- 提升研发质量？提前发现问题？
- 降低问题影响？提升诊断能力？

01. 现状与挑战

02. 治理思路与方案

03. 后续提升思路

治理思路与方案

事前预防

表结构审核
代码审核
健康检查

事中应急

监控查杀
应急恢复

事后诊断

信息采集

规范编制

规范编制

操作

- 每个表必须建立主键
- 禁止给库、表、字段单独设置字符集/排序规则
- 。。。

量化

- 扫描命中比（联机rows_examined:rows_sent<100:1）
- 事务大小（undo<10万）
- 。。。

避坑

- 大表truncate改为drop + create table (bug:68184)
- 禁用replace into (bug:73563)
- 。。。

规范编制

易理解

[规范示例]

- 条款1

条款1解读

- 条款2

条款2解读

可落地

- 表结构审核系统
- 代码审核系统
- 。。。

治理思路与方案

事前预防

表结构审核
代码审核
健康检查

事中应急

监控查杀
应急恢复

事后诊断

信息采集

规范编制

事前 - 表结构审核

规范控制

- 每个表必须建立主键
- 禁止单独设置字符集/排序规则
- 禁用TIMESTAMP数据类型
- 。。。

版本控制

- 通过系统生成建表脚本
- 通过系统生成表结构变更脚本

事前 - 代码审核

基于MyBatis开发模式，扫描mapper.xml

- SELECT/UPDATE/DELETE语句不带WHERE条件
- 查询条件最左以通配符'%'或'_'开始导致不能匹配索引
- 使用sysdate()函数导致不能匹配索引
- 使用replace into (bug:73563)
- 。。。

事前 - 健康检查

关注慢SQL

- ps.events_statements_summary_by_digest
- 关注：运行时间、扫描命中比例

SQL	时间	执行次数	执行时间（秒）	扫描记录数	返回记录数
SELECT * FROM T1 WHERE A=?	08:00	200	1,800	180,000,000	200
	09:00	300	2,700	270,000,000	300
	08:00~09:00	100	900	90,000,000	100

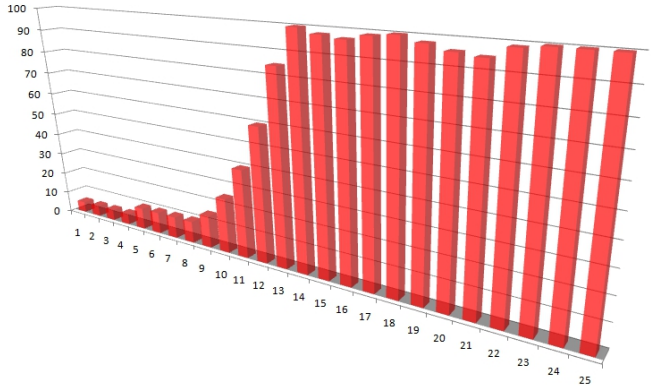
- 案例小结：08:00~09:00期间，平均执行时间=900/100=9秒，扫描命中比例=90,000,000/100=900000:1，存在明显的效率问题

治理思路与方案

事中 - 监控查杀

慢SQL的危害

- 扫描数据吃CPU资源，导致CPU飚高



- innodb_thread_concurrency用尽，导致交易堵塞

```
mysql> show engine innodb status\G
...
--------------
ROW OPERATIONS
--------------
16 queries inside InnoDB, 60 queries in queue
```

事中 - 监控查杀

慢SQL自动查杀

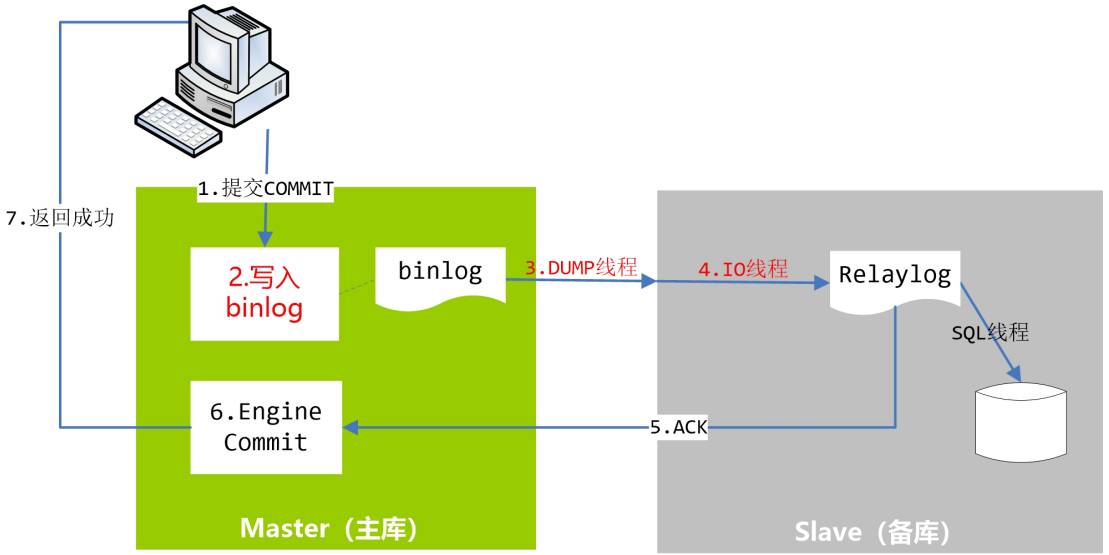
- 联机超过阈值（例如：10秒）自动执行kill
- 监控：ps.threads (show processlist)
- 联机、批量用户分离，针对用户差异性处理

```
+-------+-----------+-------------------+--------+---------+------+--------+---------------------------------+
| Id    | User      | Host              | db     | Command | Time | State  | Info                            |
+-------+-----------+-------------------+--------+---------+------+--------+---------------------------------+
| 20338 | appuser   | 192.168.1.2:39067 | dbname | Query   |  12  | update | Update t1 set a=3               |
| 20339 | appuser   | 127.0.0.1:36455   | dbname | Sleep   | 641  |        | NULL                            |
| 20340 | batchuser | 192.168.1.3:36456 | testdb | Query   |   1  | update | Insert into t2 select * from t3 |
+-------+-----------+-------------------+--------+---------+------+--------+---------------------------------+
```

事中 - 监控查杀

大事务的危害

- binlog写入、传输、回放缓慢
- 交易写入堵塞
- 主库故障，切还是不切?
- 。。。



事中 - 监控查杀

大事务自动查杀

- 监控：show engine innodb status
- 超过阈值（例如：10万）自动执行kill

```
---TRANSACTION 3236, ACTIVE 84 sec inserting, thread declared inside InnoDB 4882
mysql tables in use 1, locked 1
1 lock struct(s), heap size 1136, 0 row lock(s), undo log entries 3245768
MySQL thread id 13535, OS thread handle 139673278408448, query id 40827 localhost root executing
load data infile '/data/linzx/tmp/20190329_big_trx/test.txt' into table tt.t_big_tx_load (a)
```

事中 - 应急恢复

数据被误删除/更新，如何恢复?

[常规方案]

- 恢复PXB存量备份 + 增量binlog（单线程）

[改良方案1]

- 恢复存量备份 + 伪装slave回放增量binlog（多线程）

[改良方案2]

- DML：基于binlog的反向回放
- DDL：基于文件系统的恢复

治理思路与方案

事前预防

表结构审核
代码审核
健康检查

事中应急

监控查杀
应急恢复

事后诊断

信息采集

规范编制

事后 - 问题诊断

数据采集

[常规数据]

- CPU
- 内存
- IO、
- 网络
- QPS
- 连接数
- 。。。

[高密度采集]

- ps.threads
- show engine innodb status
- 。。。

[低密度采集]

- ps.events_statements_summary_by_digest
- ps.events_statements_history
- 。。。

后续提升思路

定位

预判

自愈

THANKS