



# MySQL异构数据同步方案 赶集网CDC项目实践

蔡峰

[www.ganji.com](http://www.ganji.com)

DTCC2012



# 目录

- 关于异构数据同步
- Tungsten Replicator介绍
- 赶集网CDC项目实践
- CDC在赶集的应用案例



# 1 关于异构数据同步

# 传统网站面临的挑战

- 性能提升
- 多平台
  - web/wap/mobile
- 社交化（SNS）
- 离线/实时统计分析
- 数据挖掘
- 多机房

- MySQL

- ACID
- 查询功能强大
- 成熟的运维方案
- DBA



# 应用层解决...

- 方法

- Multi-Write
- Message Queue

- 优势

- 简单，灵活

- 缺点

- 业务代码复杂
- 开发工作量
- 效率下降
- 容易引入数据错误
- 数据一致性问题？
- 分布式事务？

适合Cache，简单业务



# 系统层面...

- Trigger+UDF
- Data Middleware
  - MySQL Proxy?
  - Vitess
  - Myfox-query-module
- MySQL Binlog
  - Flexviews, Amoeba, jBinlog

- 数据库级别实现的增量抽取解决方案
  - 监视表的数据变化。
  - 改善ETL的效率。
  - 为检索系统提供数据更新。
  - 实现第三方的数据同步。
  - 提供外部的数据触发操作。

ORACLE



IBM

MySQL 支持CDC吗?



## 2 Tungsten Replicator介绍





# Tungsten Replicator

- A high performance, open source, data replication engine for MySQL
- O'Reilly MySQL Application of the Year 2011
- 全部开源
  - 2011.04 发布版本2.0.3
  - 最新版本： 2.0.5



# 主要特性

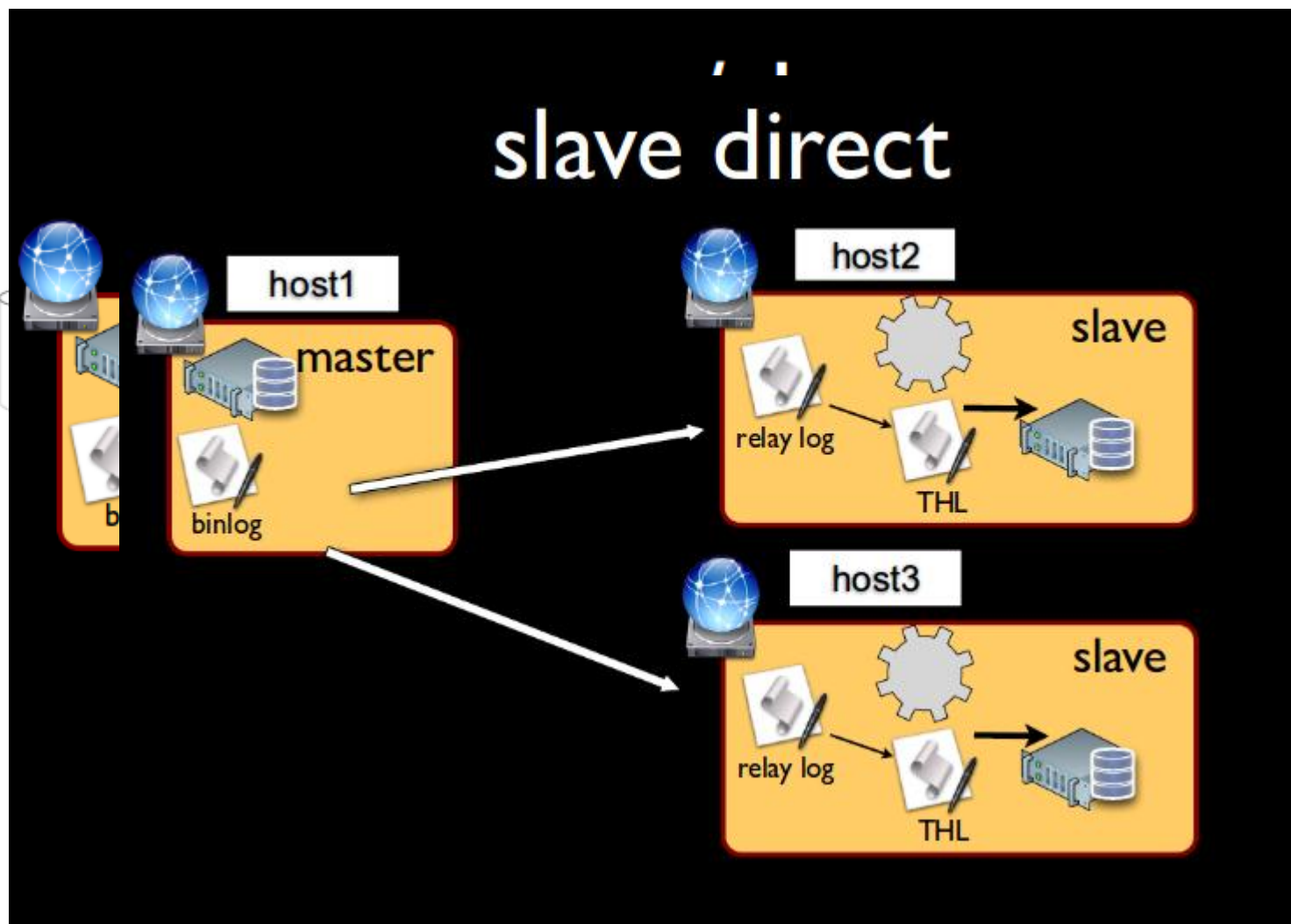
- Easy failover
- Multiple masters
- Multiple sources to a single slave
- Parallel replication
- Replicate to Oracle and PostgreSQL



# Key Points

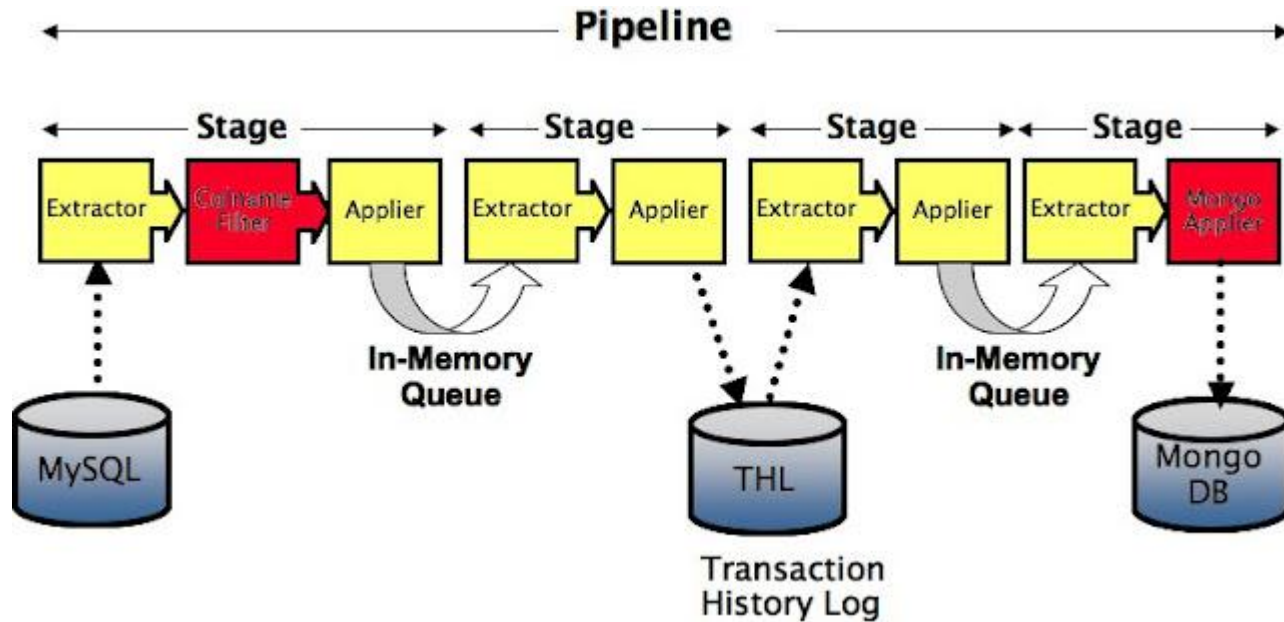
- Transaction History Logs (THL)
  - Global Transaction ID
  - Metadata
- Service Database

# TR 典型架构



## ■ MCQueueApplier

- Inspired by **MongoDBApplier**
- <https://github.com/tailorcai/tungsten-queue-applier>



# queue-applier

- Tungsten

- RBR
  - Set binlog = "row"
- **ReplicationFilter**
- **ColNamesFilter**

- 编码

- JSON
- Schema/Table
- Insert/Update/Delete

存储: Queue

高性能

持久存储

Kestrel/Memcacheq

## Kestrel

- 子队列
- 持久化
- 可靠获取



# 类似项目

<http://www.tbdata.org/>

## DbSync

用于实时同步数据库数据到HDFS的产品，通过解析各类RDBMS的log文件来提取相应的数据库动作，进而达到数据库到HADOOP的数据同步，供相关部门提取增量数据，通过dbsync，能够了解并得到所有数据的任意变化轨迹。



## Databus

*A System for Timeline-Consistent Change Data Capture*

DTCC2012

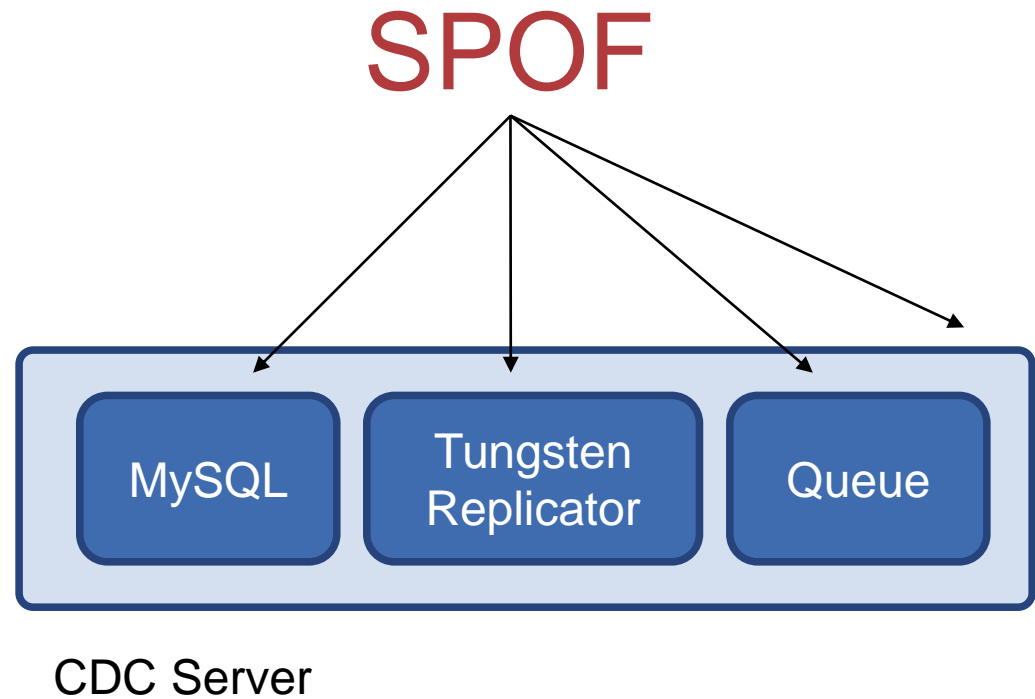


### 3 赶集网CDC项目实践



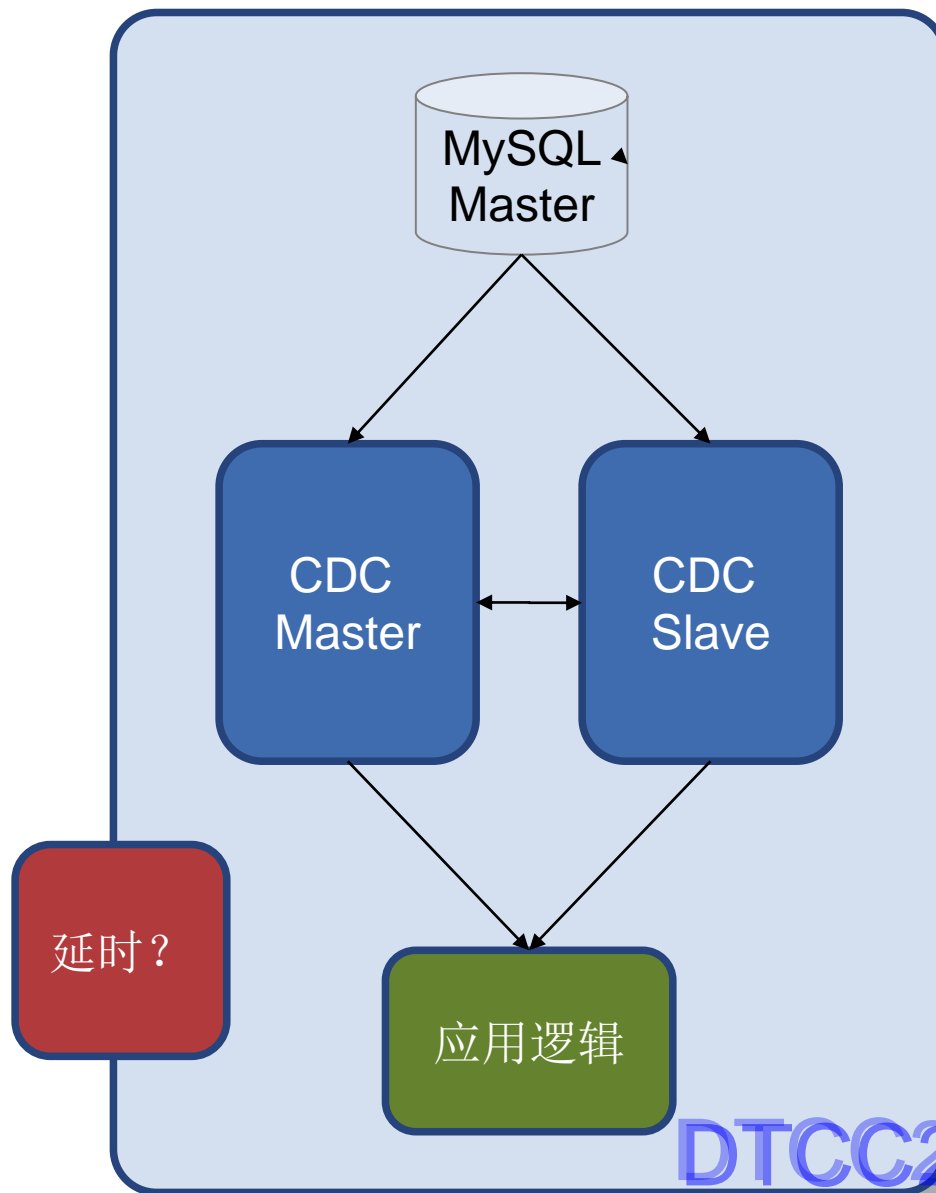
# HA?

- 典型错误
  - 服务崩溃
  - 网络通讯中断
- 同步延迟
- 操作阻塞

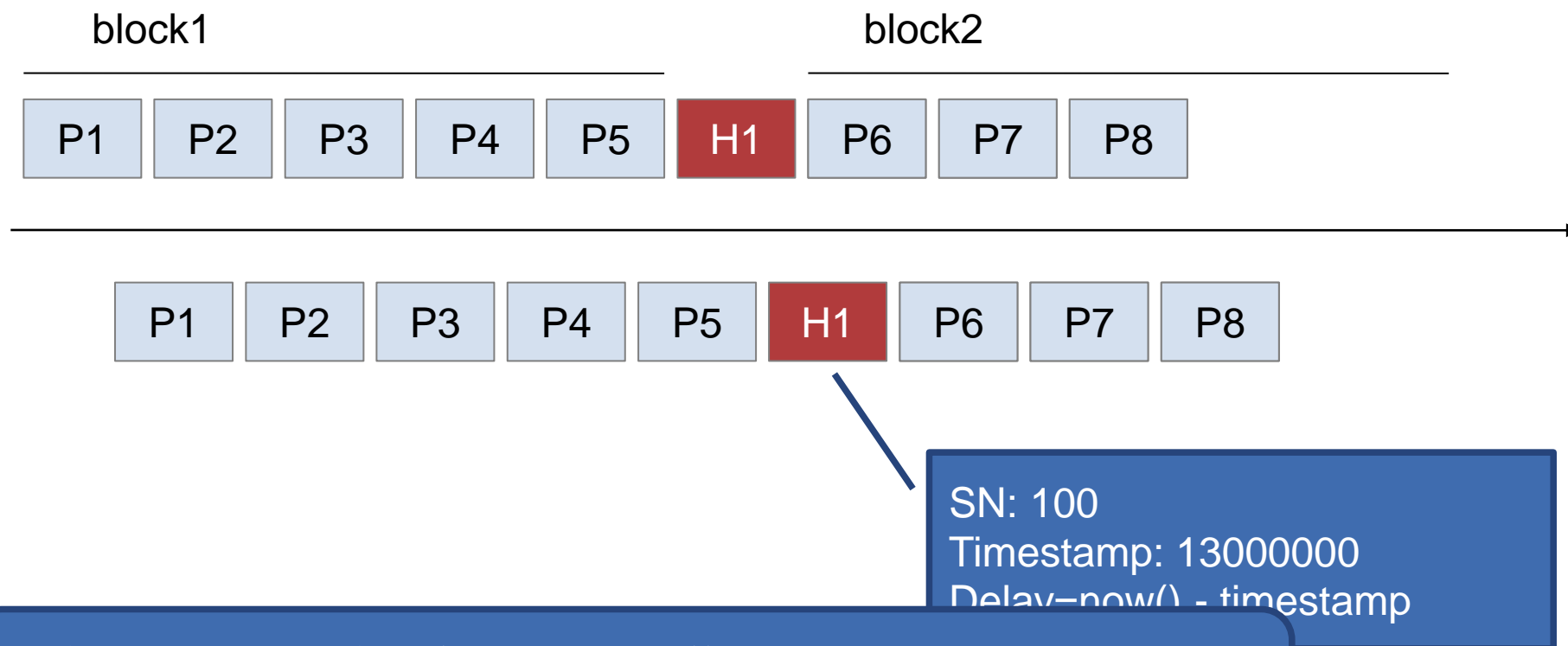


# Master/Slave

- 主从切换需求
  - 严格保持顺序
  - 数据重复可以
  - 数据丢失不可以
  - 切换对应用保持透明



# Master/Slave 2



定时更新heartbeat表，生成heartbeat数据

以Block为单位进行切换

Master处理完成一个Block后，通知Slave处理该Block

切换时，Slave重放当前Block，确保数据不丢失

# Zookeeper



## 配置信息:

- Master CDC IP:PORT
- Slave CDC IP:PORT

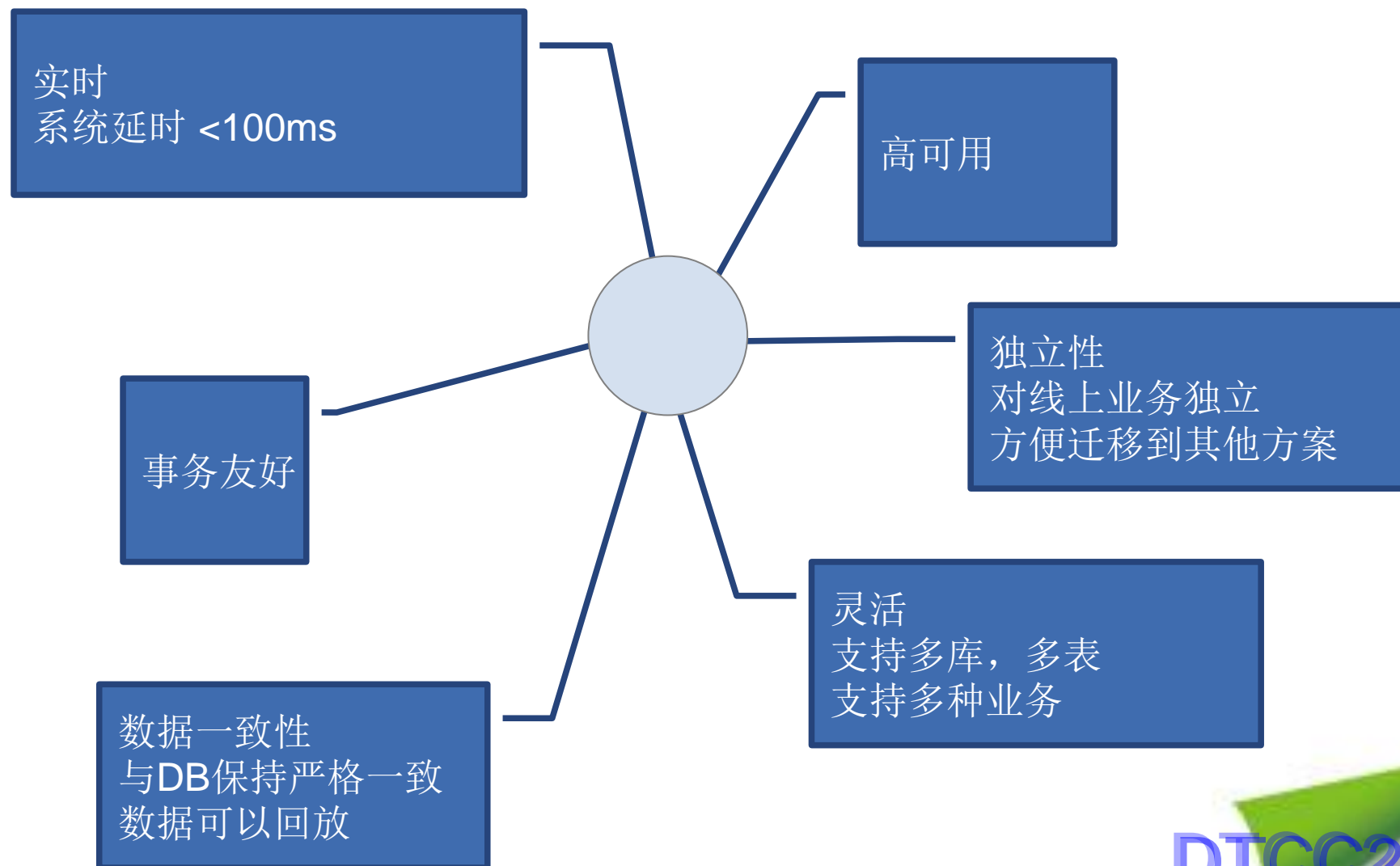
## 进程信息

- 0 – master HOST:PID
- 1 – slave HOST:PID
- 2 – standby HOST:PID

## 同步状态信息:

- Master SN
- Master Delay
- Slave SN
- Slave Delay

# 优点



- 监控和报警（zabbix...）

- 主从同步状态
- 延迟(MySQL,CDC)

- DBA？

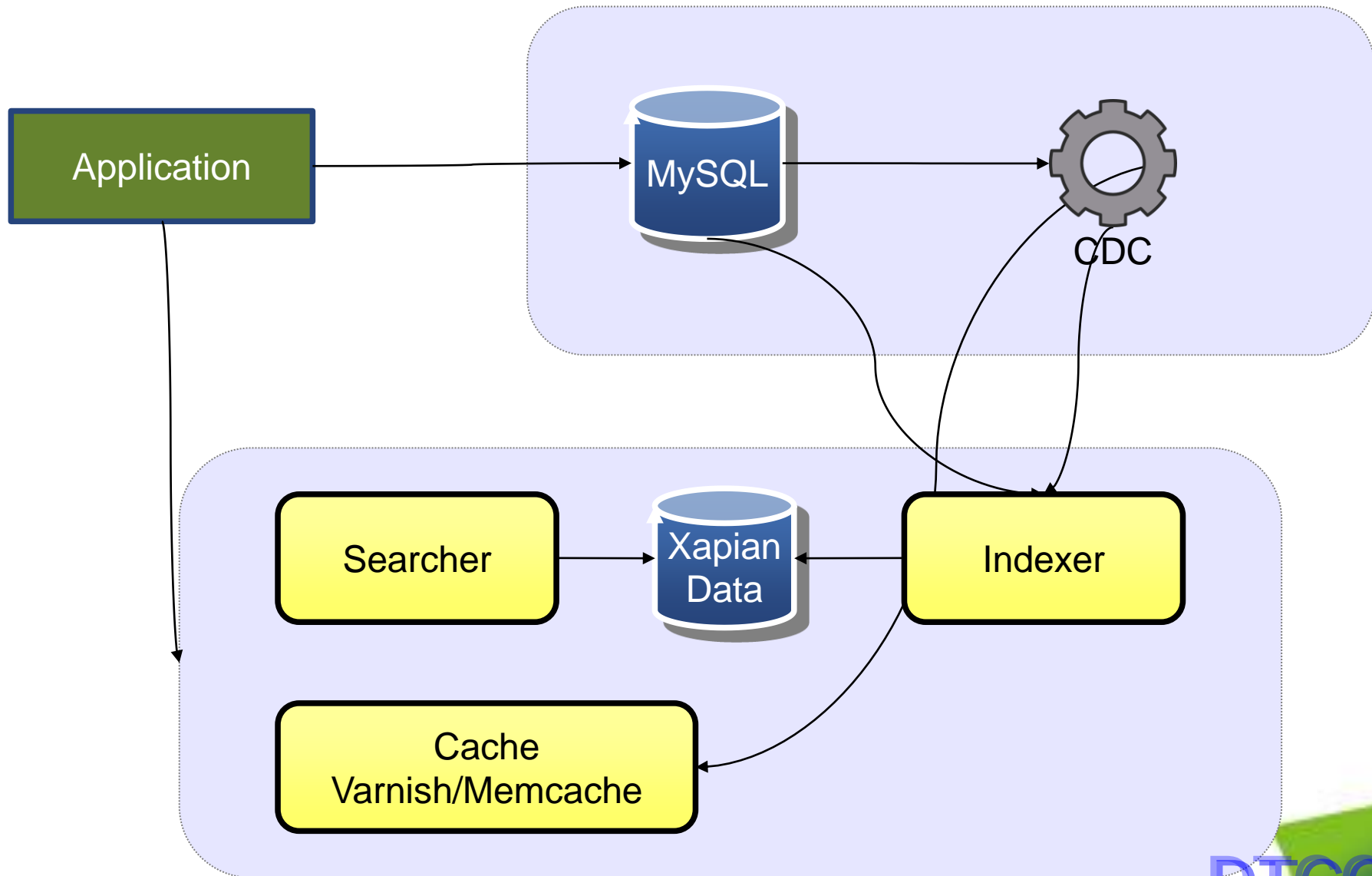
- 批量更新一定加Limit
- 数据分片<10W
- 插入sleep，缓解延迟影响
- 不规范的SQL语句监控？





## 4 CDC在赶集的应用案例

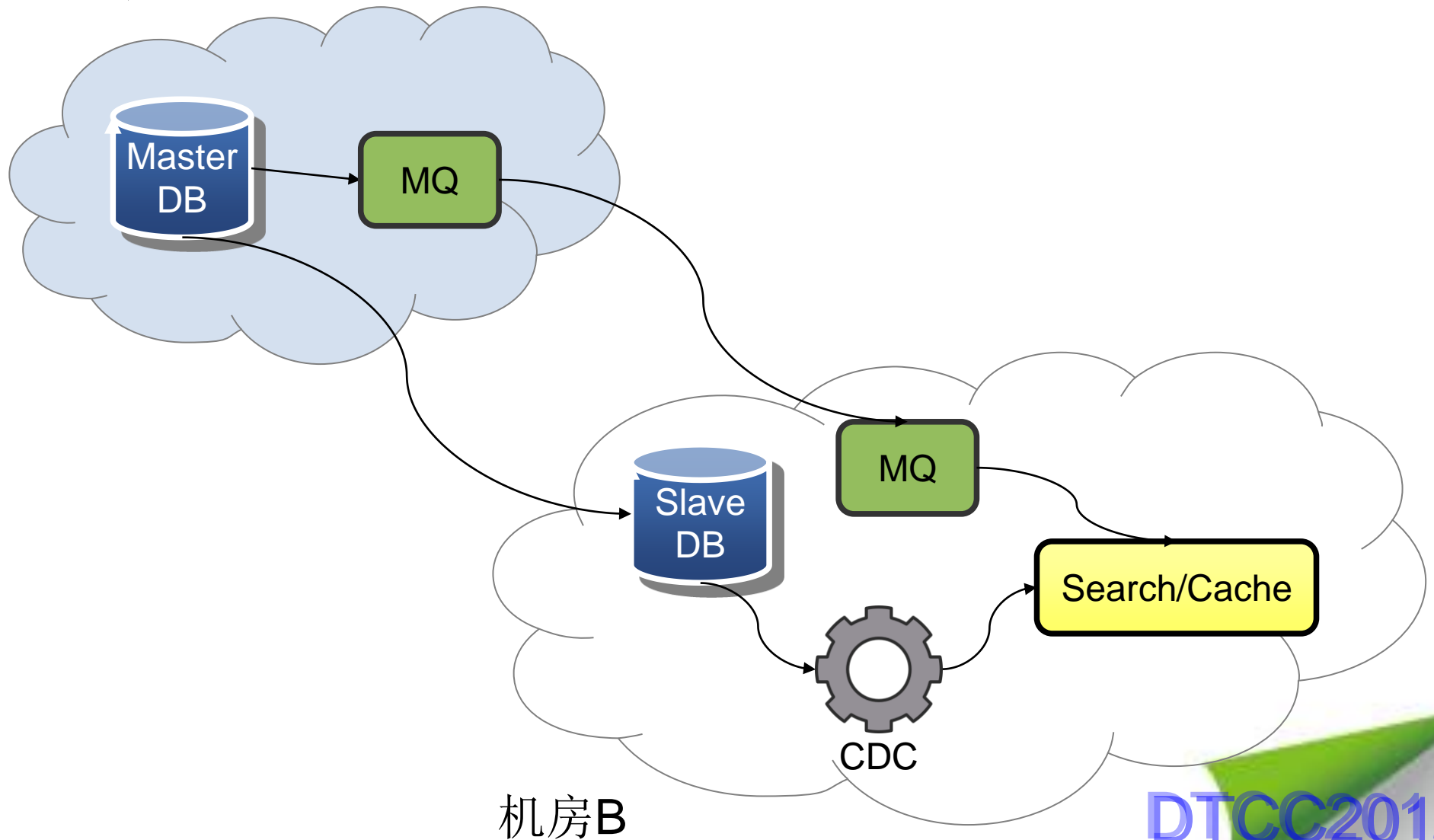
# Indexing & Caching

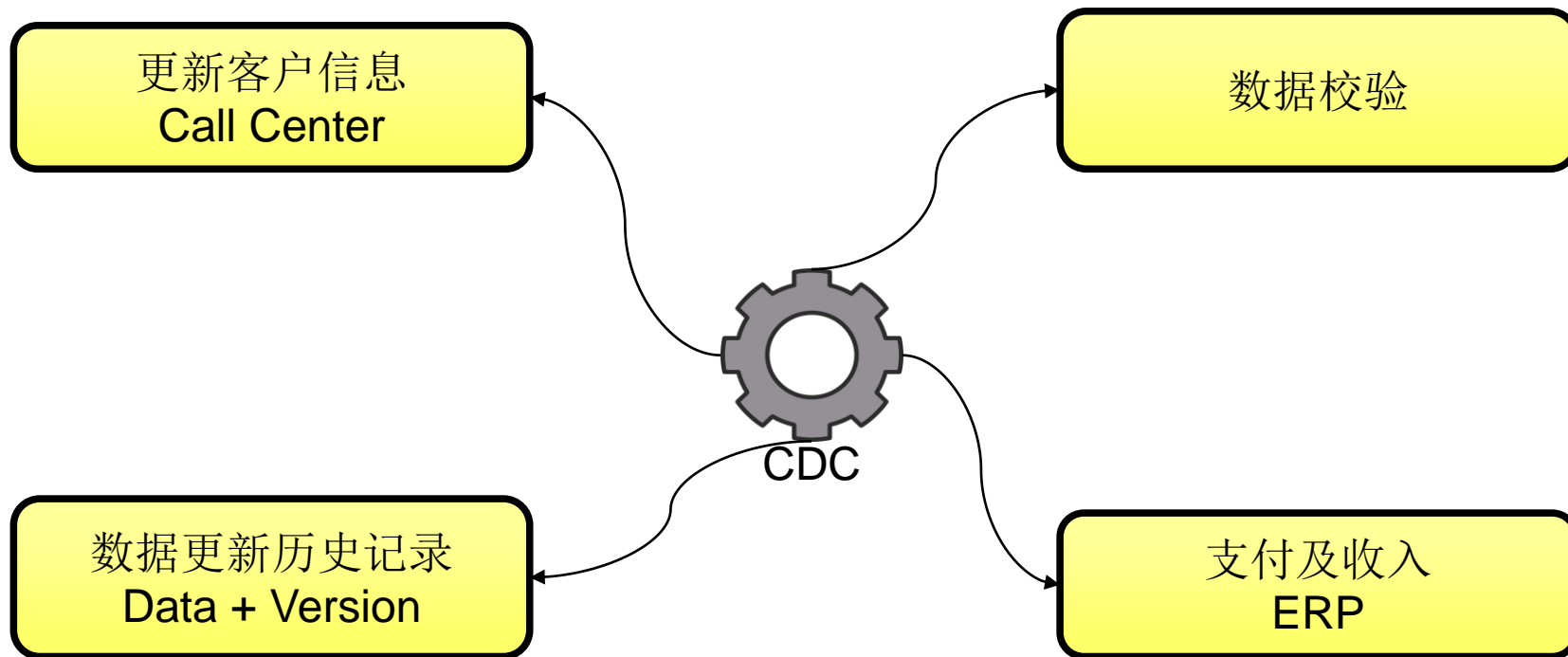




# 多机房同步

主机房A





产品线: >5  
库表: >800  
数据量: 500W+/天  
维护: 2人 (DBA+RD)

- 应用开发
  - 历史数据回放/全量数据抽取
  - 更友好的**API**接口及订阅功能
  - 多种数据一致性保证
  
- 系统
  - Upgrade to Tungsten 2.0.5
  - JSON to BSON
  - 主从自动切换



谢谢！



DTCC2012