海量数据下携程数据库架构的探索和实践

携程旅行网 - 孙天浩

个人简介

- 数据库技术专家
- SQLServer / MySQL / Python / Golang

- 数据库混布，无任何Sharding方案
- 程序日志存放在数据库
- 用复制分发做读写分离架构，跨BU，网状
- BI ETL取数，全量，5点前可以完成

早期现状

业务快速增长遇到的问题

2010年，业务快速发展……

- 单机存储容量限制
- 数据库本身处理能力有限
- QPS过高数据库性能下降
- DB和表过大影响正常运维，如备份，DDL

要求架构支撑10X业务发展！

数据库以及应用的解耦

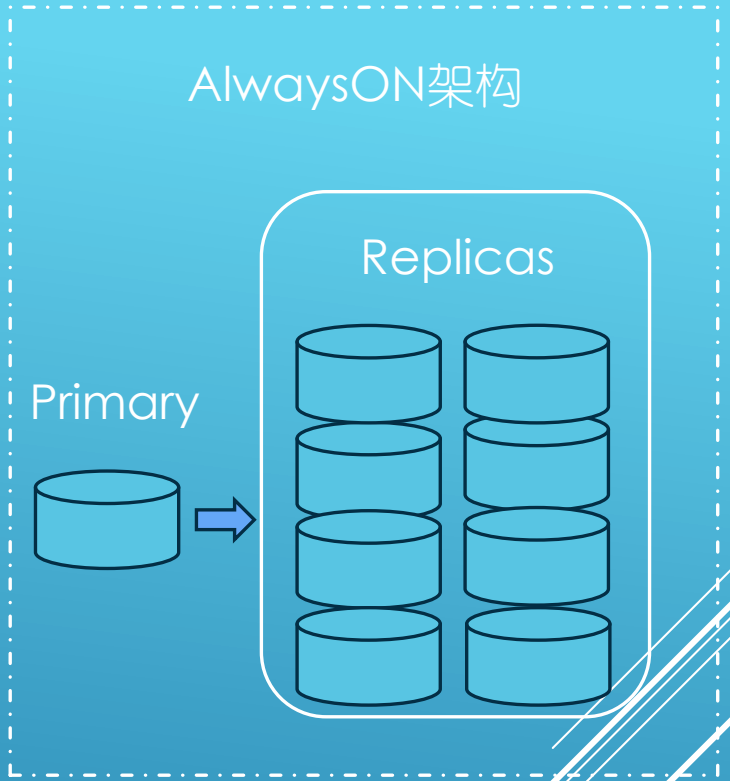
- 复杂业务拆分，降低系统耦合度
- 复制分发消除跨BU链路，改为调用接口
- 立项处理慢SQL，消除存储过程以及视图
- 数据库消除跨库Join，为拆分和数据库sharding做准备

读写分离由复制分发改造为AlwaysON

- SQL Server AlwaysON，同步快延迟小，维护简单
- 理想的读写分离架构方案，支持跨IDC

BI ETL取数调整为增量

- 通过添加时间戳
- 取数源从快照调整为可读副本



虽然有些效果，但是随着数据量持续增长，系统瓶颈还是存在☹

短期方案 数据库垂直拆分

业务梳理

- 分析现有系统架构以及主要业务流程

划分类别

- 分析各数据表进行分类

垂直切分

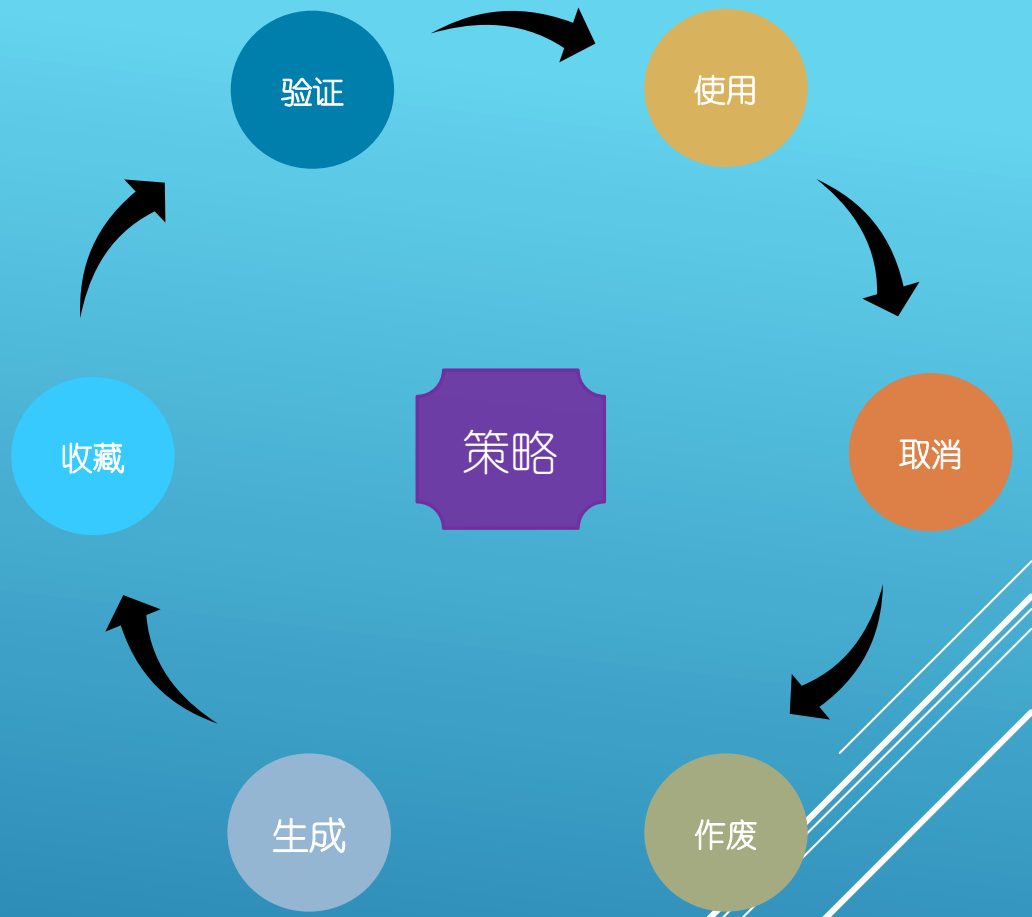
容量预估

- 分析数据库表结构以及数据量
- 分析增长趋势

水平切分

系统设计

- 数据库设计
- 数据异构设计
- 系统架构设计



消费券架构演变案例

SHARDING 拆分方式

Original Table

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
1	TAEKO	OHNUKI	BLUE
2	O.V.	WRIGHT	PURPLE
3	SELDA	JOHN	RED
4	JIM	JAMES	GREEN

Vertical Partitions

VP1

CUSTOMER ID	FIRST NAME	LAST NAME
1	TAEKO	OHNUKI
2	O.V.	WRIGHT
3	SELDA	JOHN
4	JIM	JAMES

VP2

CUSTOMER ID	FAVORITE COLOR
1	BLUE
2	PURPLE
3	RED
4	GREEN

Horizontal Partitions

HP1

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
1	TAEKO	OHNUKI	BLUE
2	O.V.	WRIGHT	PURPLE

HP2

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
3	SELDA	JOHN	RED
4	JIM	JAMES	GREEN

SHARDING优势以及存在的问题

Benefits

- 通过扩展解决容量瓶颈
- 分散负载，提高吞吐
- 加快查询响应时间
- 易于维护，备份恢复时间，表结构变更时间等

Drawbacks

- 加大架构复杂度
- 分片的扩容/缩容困难
- 全局唯一分片key问题
- 数据记录的维度问题
- 跨库事务问题
- 同组数据跨库问题

数据记录的维度问题

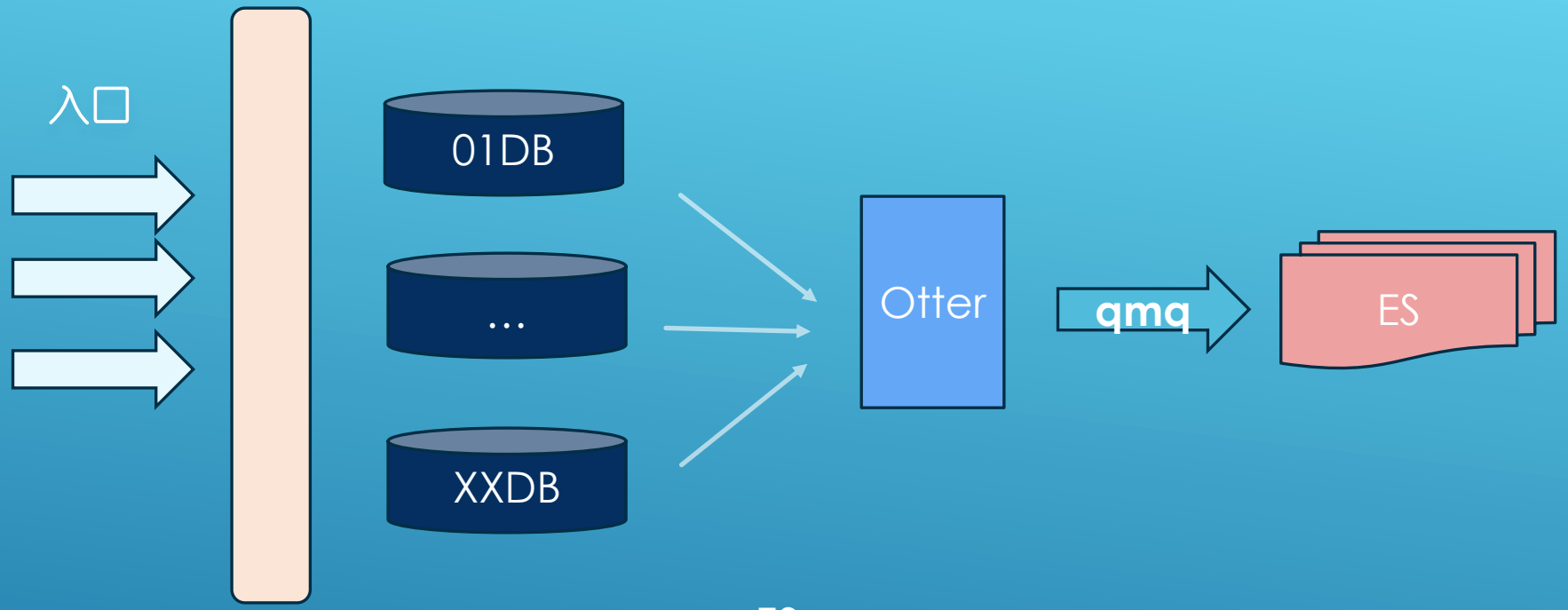
- 数据异构

例:客户从携程上订了机票，酒店，景点门票等订单后，需要查询刚刚下的订单？

思路：各BU产品订单，会放在以订单号为分片键的shard库内，直接查的话需要从各BU的订单库内以用户ID查找其下的订单，这样性能肯定不符合要求。

做法：携程通过OrderIndex项目实时将各BU的订单信息同步OI库内，以用户ID作为分片键，最终用户看到“我的携程”内的数据实际是基础部门以用户ID查询后的数据。

SHARDING 数据异构



ES:

面向文档型数据库，可以作为全文搜索引擎
数据可各维度查询

SHARDING 数据异构

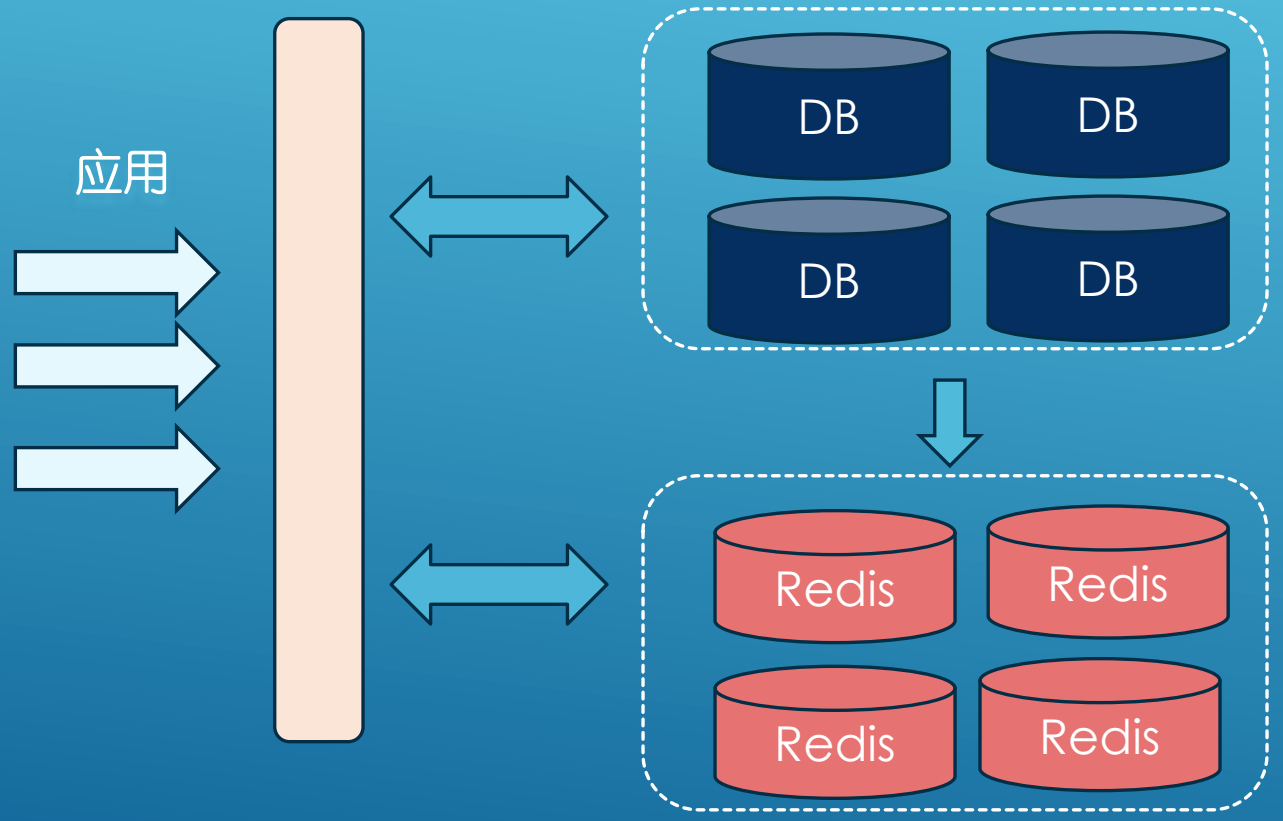
SHARDING 数据跨库问题

跨库事务问题

- 从架构设计上尽量避免
  - 同shard片内所有表需以相同维度的分片键；
  - 或表之间无耦合关系；
- 特殊情况下通过消息队列来做补偿机制，尽最大可能保证事务最终一致性；

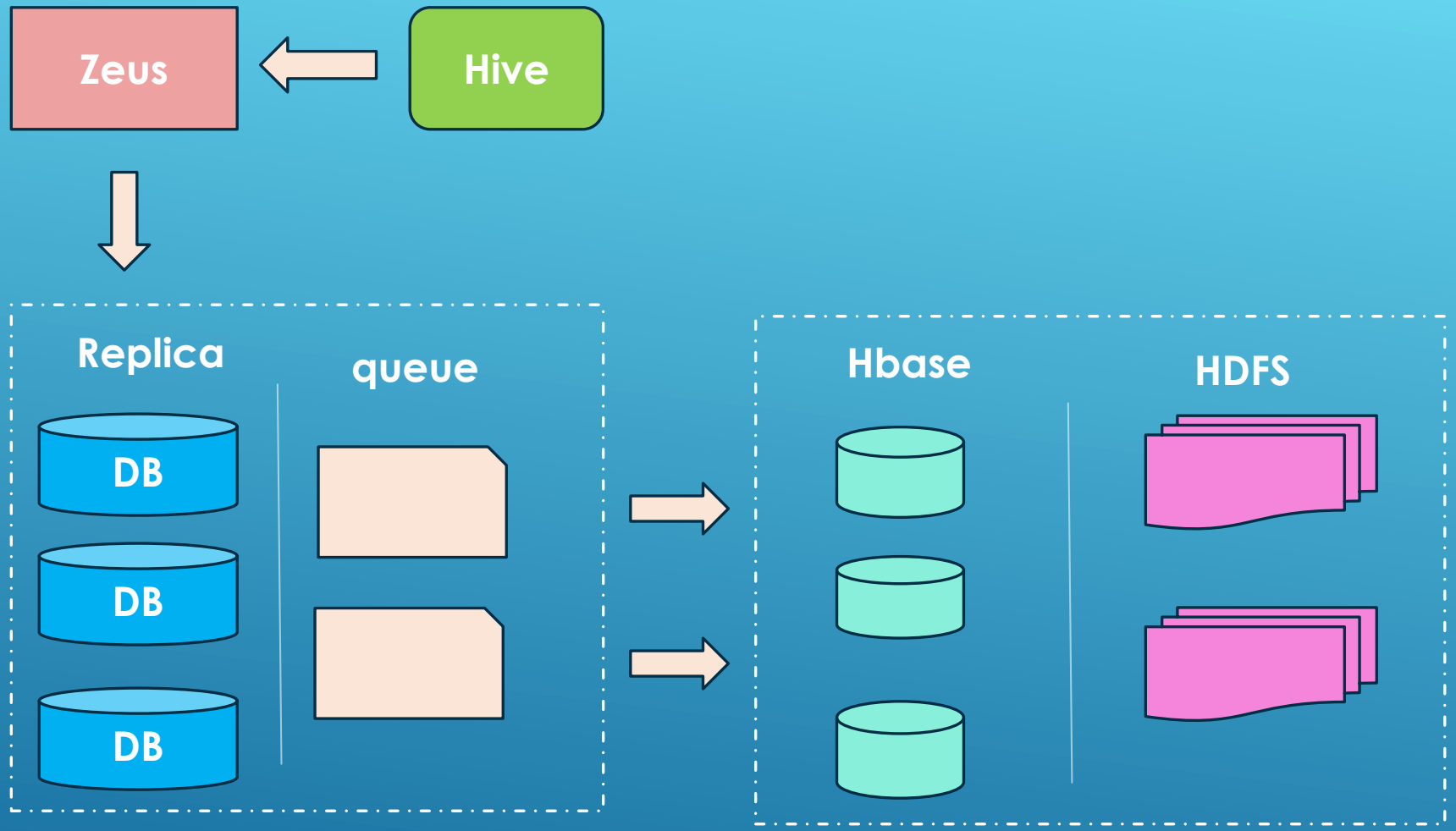
同组数据跨库问题

- Memcache
- Redis
  - 公司开发了Credis/Xpipe，支持分片扩展以及IDC间数据同步



数据缓存

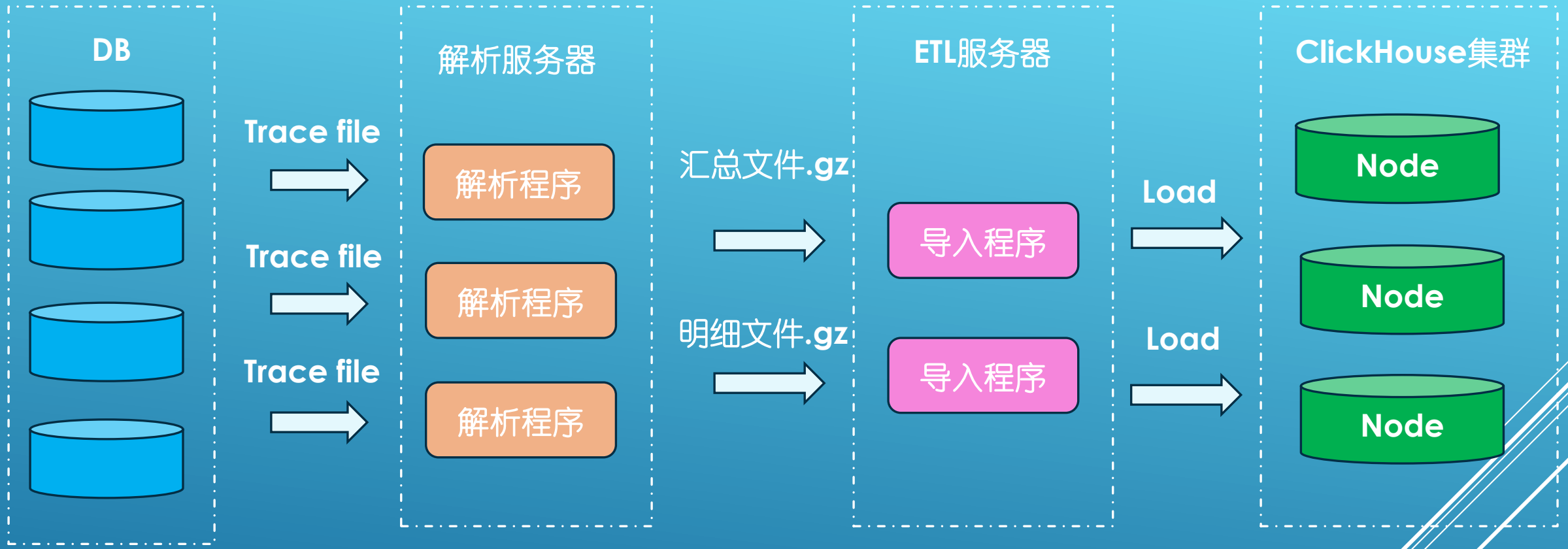
分布式数据库解决方案



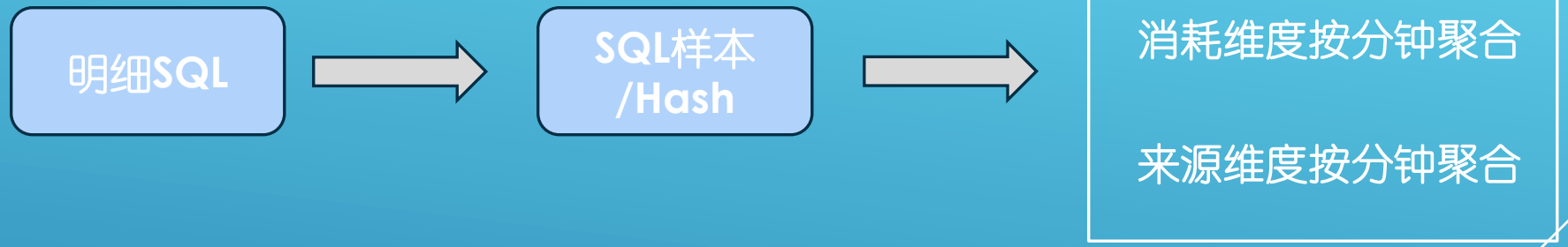
NoSQL - HBase:

- ► 解决了扩展问题
- ► 缺少ACID，SQL 支持
- ► Region Compaction

数据离线计算



TRACE数据流式处理



- 日均大小从2T 增长到40T
- 数据存放从SQLSERVER 到 GreenPlum 再到 ClickHouse的演进
- 整体延迟在2分钟以内

TRACE数据汇总分析


XEvent Watcher
排名
聚合
明细
来源
应用
比较
信息
哈希
平均值
DBA
解析服务器压力图
语句开始时间：
2019-03-28 12:41:0
2019-03-28 12:51:0
服务器：
xxx
数据库：
xxx
查询
下载
SQL(Sample)
begintime
abortcounts
execcounts
Duration(ms)
reads
writes
cpu
rowccounts
type
link
2019-03-28 12:4 0 26980 315733 (20.23 %) 30953972 (16.69 2 (0.00 %) 296789 (25.86 % 1302502 null
2019-03-28 12:4 0 819 91241 (5.85 %) 50324122 (27.14 1 (0.00 %) 87389 (7.61 %) 4140 null


语句开始时间：
2019-03-28 01:41:0
2019-03-28 12:51:0
服务器：
xxx
数据库：
xxx
查询
性能指标每分钟均值
Values
17.5k
15k
12.5k
10k
7.5k
5k
2.5k
0k
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
CPU均值（毫秒）
读取均值
写入均值
返回行数均值
执行次数

TRACE聚合展示

- NewSQL
  - 解决扩展问题
  - 支持ACID
  - 支持SQL
  - Paxos / raft 数据强一致性保证
  - OLTP / OLAT 均有适用场景

谢谢！