CDC：DB到ES实时同步

李猛 lemon
2019/11/27, 技术架构师, Elastic深度用户,@物流速运

序言

概念定义

- CDC概念
  - Change Data Capture 变更数据捕捉
  - 几乎所有数据库都基于WAL机制
- Elasticsearch是近实时数据库，不是实时数据库
  - 内部基于Refresh机制
- Elasticsearch不是关系型数据库， 不具备关系数据库严格的 ACID 特性
  - Nosql是乐观锁模式
  - 关系数据库是悲观锁模式
- 任何两个数据库之间数据同步都有以下问题
  - 数据一致性
  - 数据实时性

01.需求背景

DB到ES实时同步需求背景

技术需求背景

DB局限性

- 复杂条件查询能力
- 关联查询效率低
- 不具备弹性扩展能力
- 索引创建/使用复杂度
- 超大数据量

技术需求背景

DB局限性

- 复杂条件查询能力
- 关联查询效率低
- 不具备弹性扩展能力
- 索引创建/使用复杂度
- 超大数据量

ES互补性

- 高效查询效率
- 弹性扩展能力
- 索引创建/使用方便
- 反范式关联能力

技术需求背景

DB局限性

- 复杂条件查询能力
- 关联查询效率低
- 不具备弹性扩展能力
- 索引创建/使用复杂度
- 超大数据量

ES互补性

- 高效查询效率
- 弹性扩展能力
- 索引创建/使用方便
- 反范式关联能力

DB ≠ ES

- DB：我很全能
- ES ：我很专注

业务需求背景

业务领域复杂度

- 单业务领域水平分库分表
- 多业务领域垂直分库分表

业务需求背景

业务领域复杂度

- 单业务领域水平分库分表
- 多业务领域垂直分库分表

业务查询需求

- 水平分库分表的聚合查询
- 多业务关联联合查询

业务需求背景

业务领域复杂度

- 单业务领域水平分库分表
- 多业务领域垂直分库分表

业务查询需求

- 水平分库分表的聚合查询
- 多业务关联联合查询

DB+ES

- 业务数据存储
- 业务数据查询

DB与ES结合问题

- DB解决了ACID事务
- ES解决了高效查询

elastic

DB与ES结合问题

DB+ES结合

- DB解决了ACID事务能力
- ES解决了高效查询

同步实时性

- 同步实时性要求
- 同步实时性能力

DB与ES结合问题

DB+ES结合

- DB解决了ACID事务能力
- ES解决了高效查询

同步实时性

- 同步实时性要求
- 同步实时性能力

数据一致性

- 如何保障一致性
- 如何修复数据

02.同步场景

表与索引映射关系

elastic

单数据表=单索引

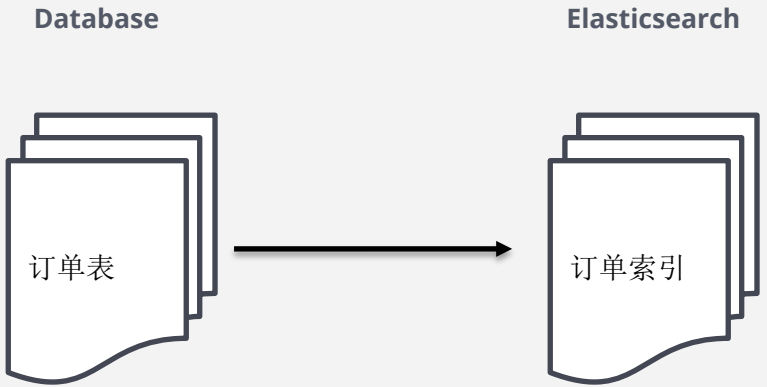
一对一映射关系

场景介绍：

ES作为DB的映射
DB为原始数据源
ES为查询引擎

用途说明

DB关联查询能力局限，水平分库分表
数据实时查询要求不高
DB索引能力局限
DB解决一致性，ES解决查询性能



单数据表=多索引

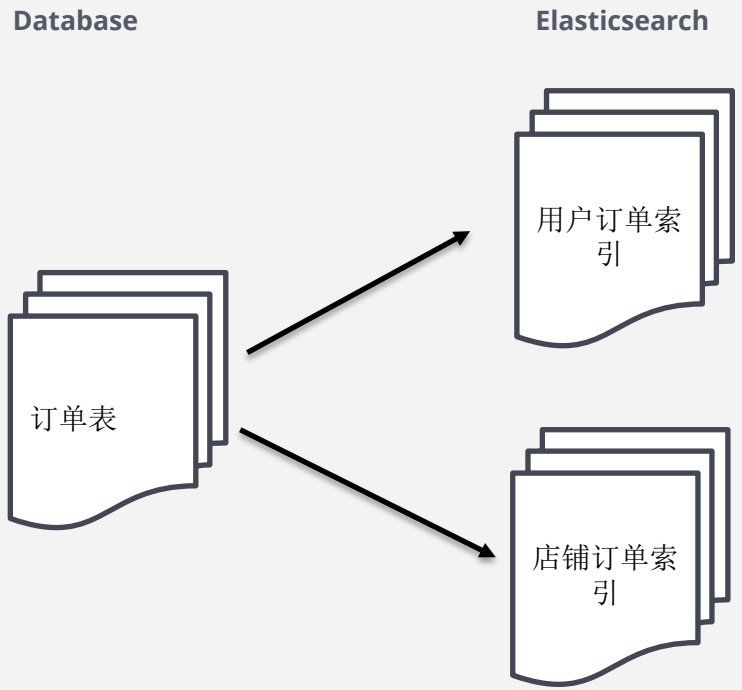
一对多映射关系

场景介绍

同一DB表成为多个索引的数据
DB表作为索引主体对象
DB表作为索引的子对象

用途说明

DB关联查询能力瓶颈
DB索引查询能力限制



多数据表=单索引

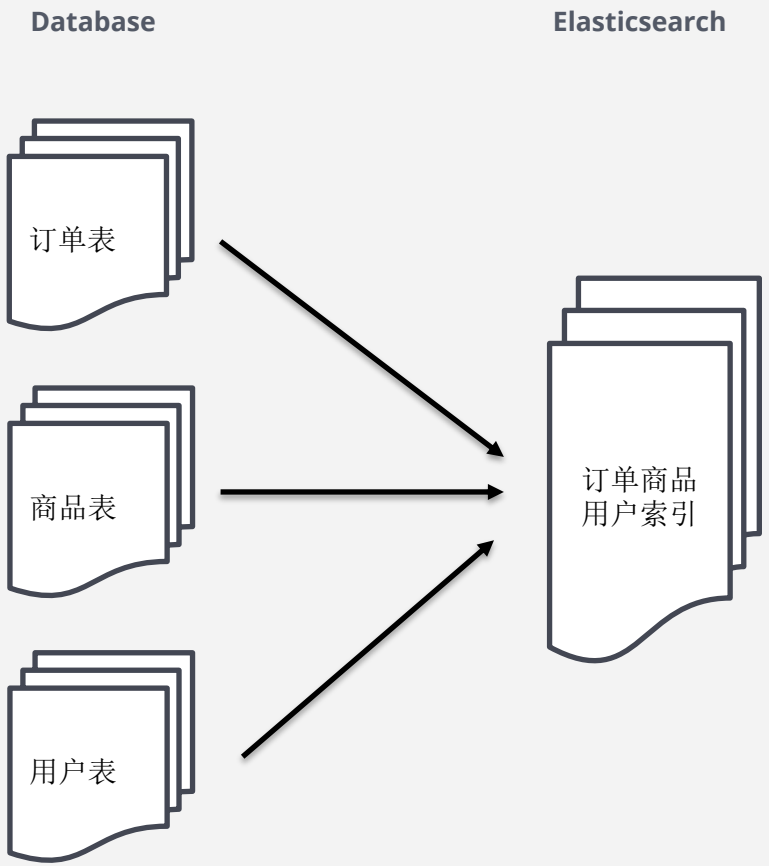
多对一映射关系

场景介绍：

多个DB表
一个索引
大宽表结构

用途说明

DB关联查询能力局限
ES解决关联查询问题
DB索引能力局限
单领域业务
通用查询能力



多数据表=多索引

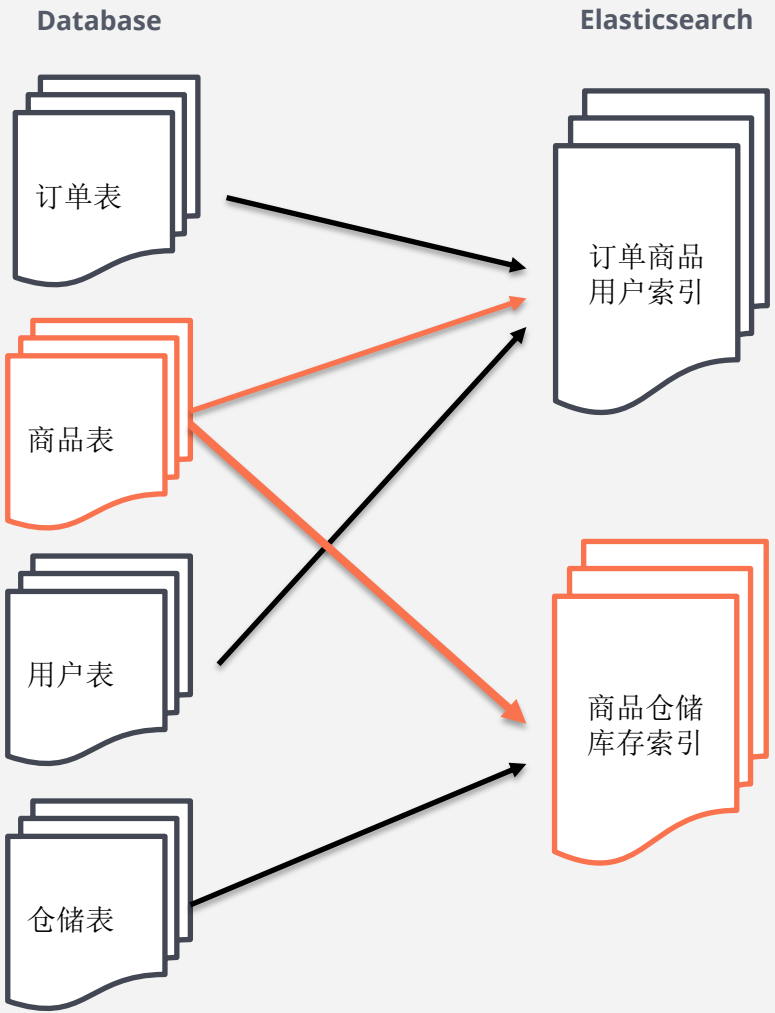
多对多映射关系

场景说明

多个DB表
多个索引

用途说明

DB关联查询能力局限
DB跨库查询能力局限
多个领域业务查询
通用查询能力



多源数据表=多索引

多源多表 多对多映射关系

场景说明

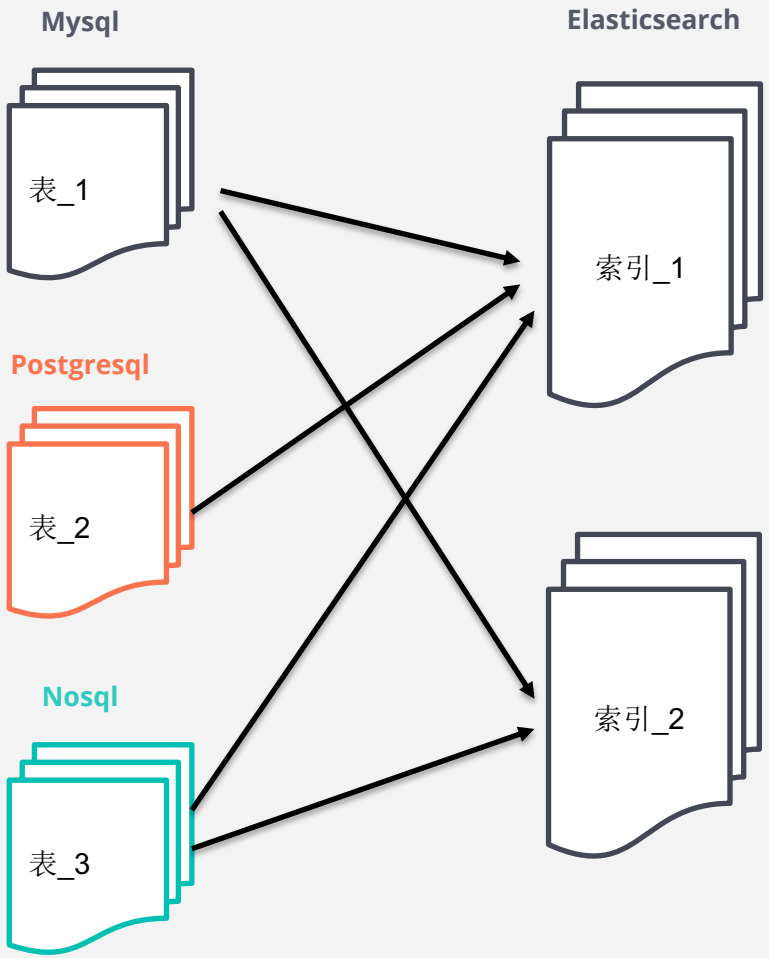
多种数据源表

关系数据库/非关系型数据库

多个索引映射

用途说明

多领域业务关联查询



03.技术方案

解决数据实时性与一致性

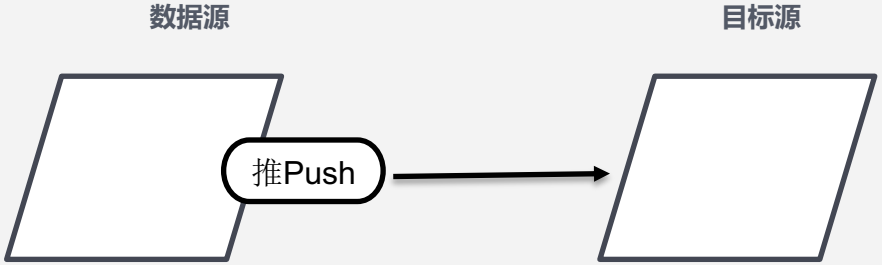
elastic

数据同步模式

同步理论

推Push

数据源主动推送到目标源



数据同步模式

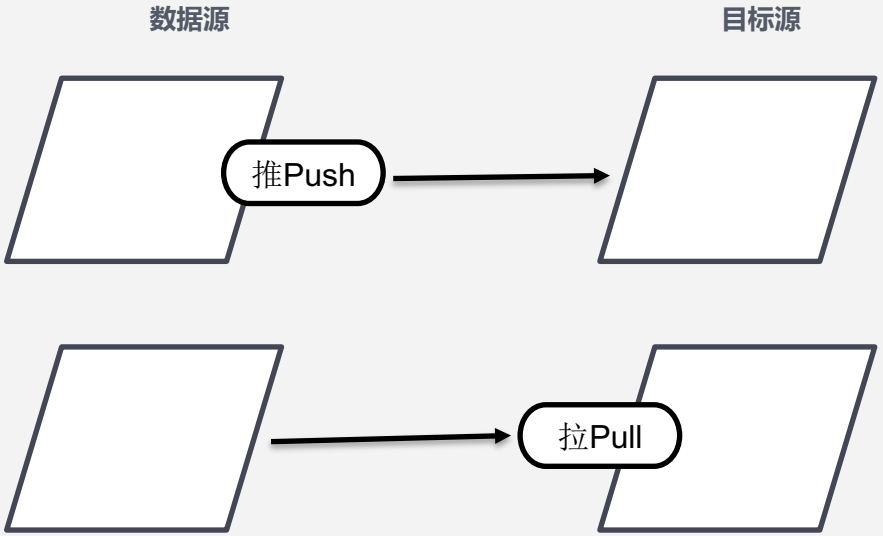
同步理论

推Push

数据源主动推送到目标源

拉Pull

目标源主动拉取数据源



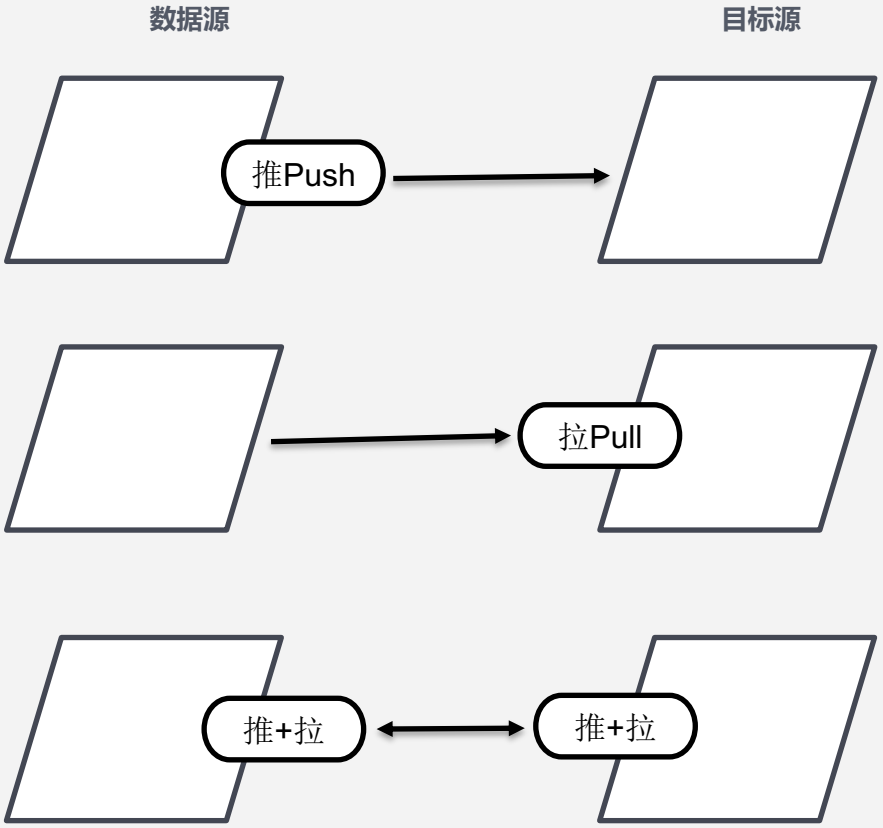
数据同步模式

同步理论

推Push
数据源主动推送到目标源

拉Pull
目标源主动拉取数据源

推+拉
数据源与目标源之间推拉结合



CDC技术方案

关键实现

db

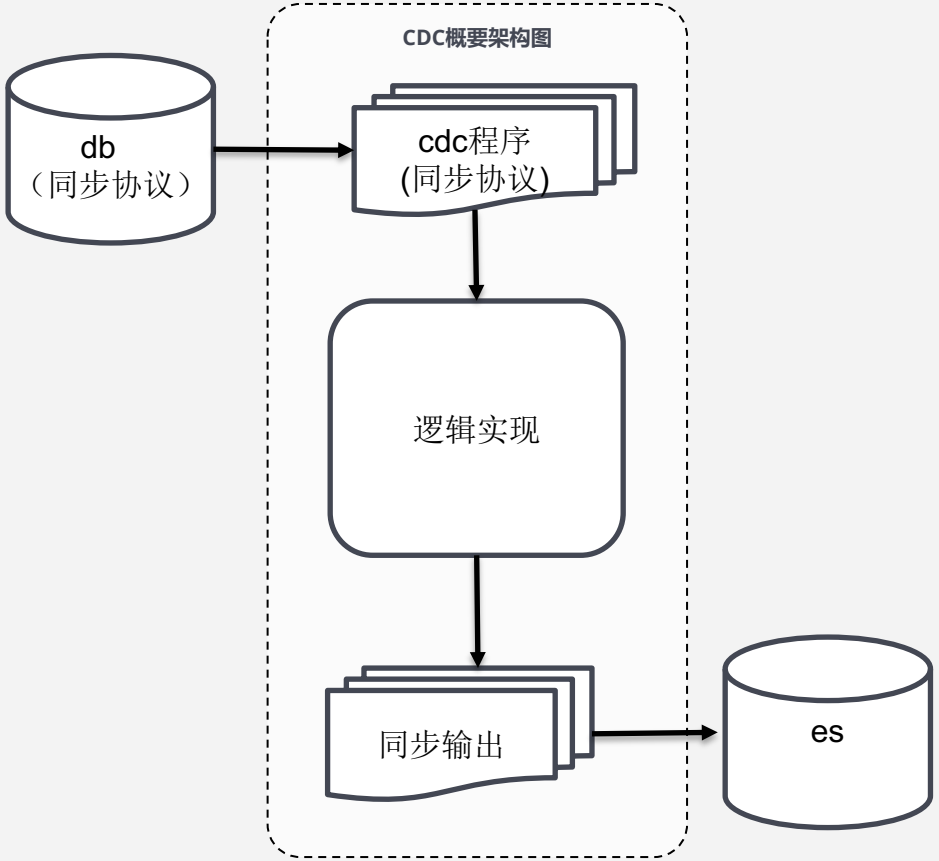
db启用cdc机制，记录变更到本地

cdc程序

cdc程序订阅db变更记录

逻辑实现

逻辑程序将db变更数据映射到es

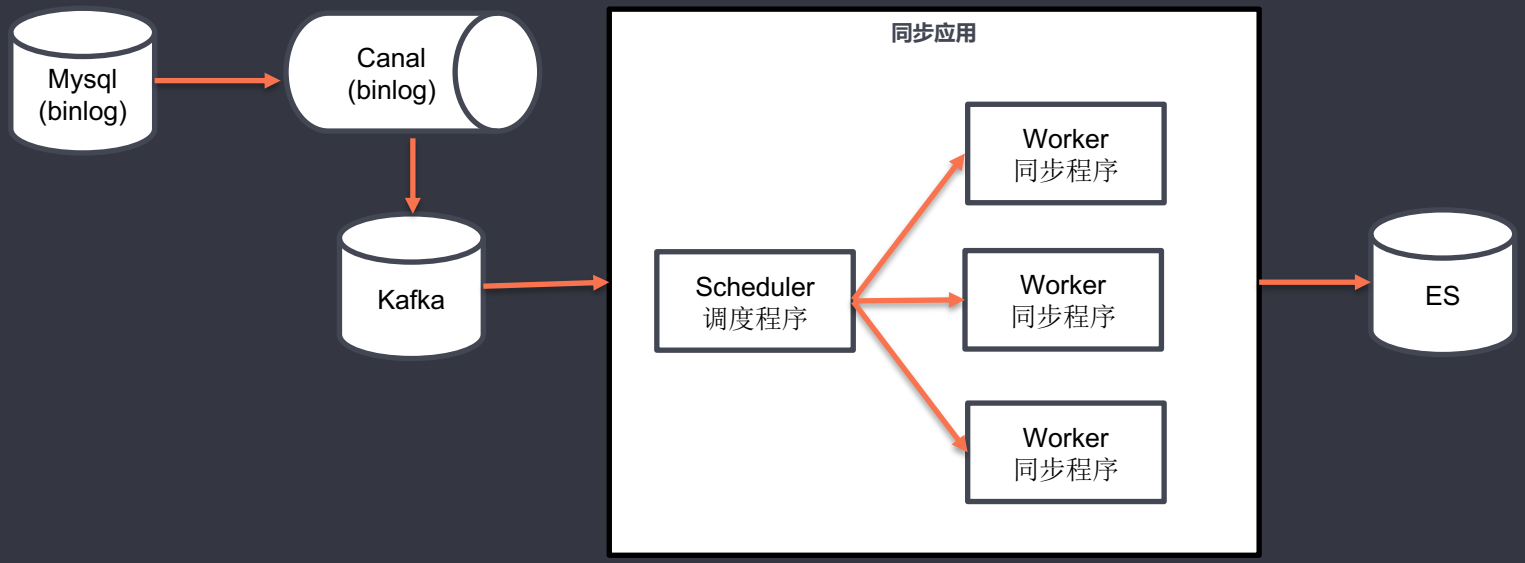


案例：Mysql 同步到ES

基于mysql-binlog实时同步

案例：Mysql同步到ES

技术架构



案例：Mysql同步到ES

Mysql-binlog：主从同步

Master主库：

捕捉变更记录

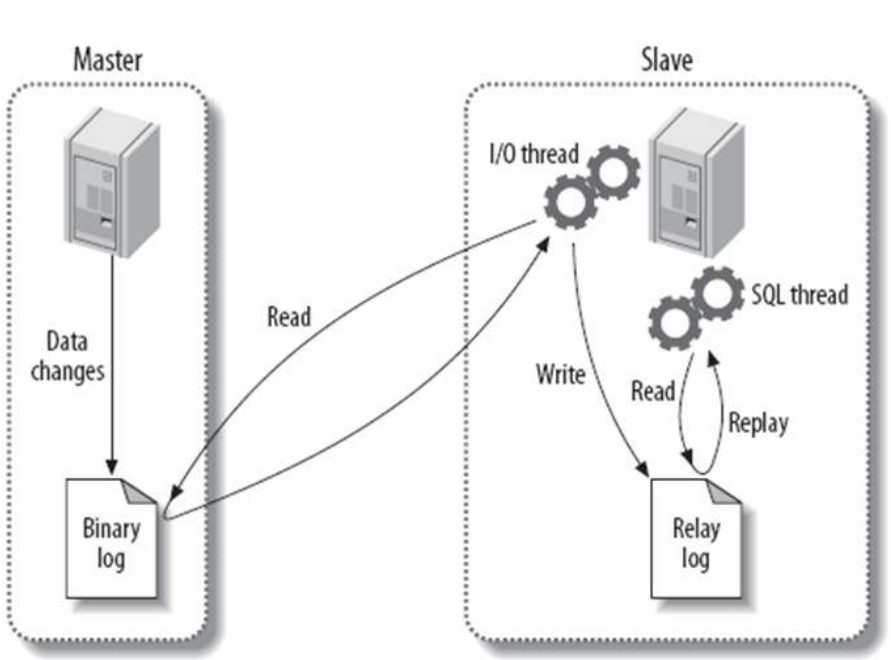
顺序保存到本地binlog文件

Slave从库：

同步主库变更记录binlog文件

回放主库binlog

更新从库数据



案例：Mysql同步到ES

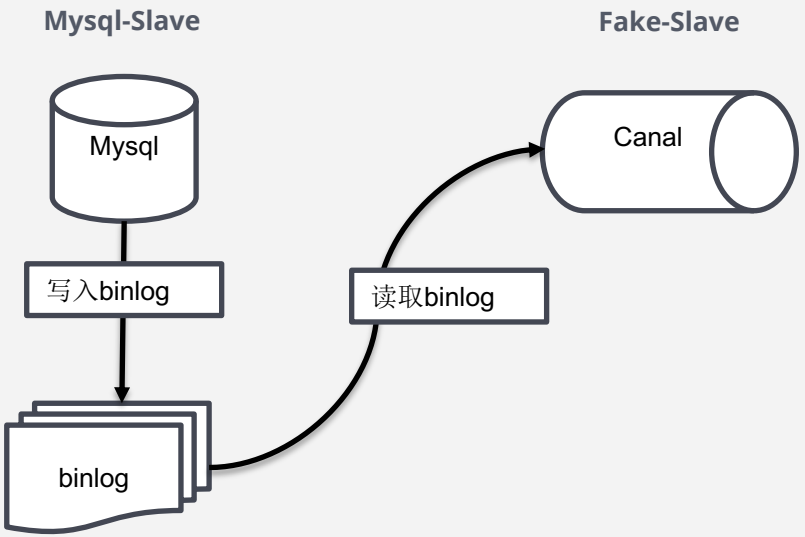
Canal：原理介绍

Binlog启用

选择Mysql从机
启用binlog机制
变更数据记录binlog文件

Canal读取数据

伪装Mysql从机
订阅Mysql同步
拉取变更数据
回放变更数据
解析变更数据
输出变更数据
保证数据顺序



案例：Mysql同步到ES

Canal解析样本：库名、表名、操作

```
{
    "data": [
        {
            "id": "1",
            "product_name": "Name_456"
        }
    ],
    "database": "product_db",
    "old": [
        {
            "id": "1",
            "product_name": "Name_123"
        }
    ],
    "table": "prodcut_table",
    "type": "UPDATE"
}
```

案例：Mysql同步到ES

Canal解析样本：变更数据

```
{
    "data": [
        {
            "id": "1",
            "product_name": "Name_456"
        }
    ],
    "database": "product_db",
    "old": [
        {
            "id": "1",
            "product_name": "Name_123"
        }
    ],
    "table": "prodcut_table",
    "type": "UPDATE"
}
```

案例：Mysql同步到ES

Canal解析样本：历史数据

```
{
    "data": [
        {
            "id": "1",
            "product_name": "Name_456"
        }
    ],
    "database": "product_db",
    "old": [
        {
            "id": "1",
            "product_name": "Name_123"
        }
    ],
    "table": "prodcut_table",
    "type": "UPDATE"
}
```

Canal：关键设置

高可用

- 单实例性能
- 集群服务

Canal：关键设置

高可用

- 单实例性能
- 集群服务

表映射

- 逻辑表
- 分库分表
- 表主键设置

Canal：关键设置

高可用

- 单实例性能
- 集群服务

表映射

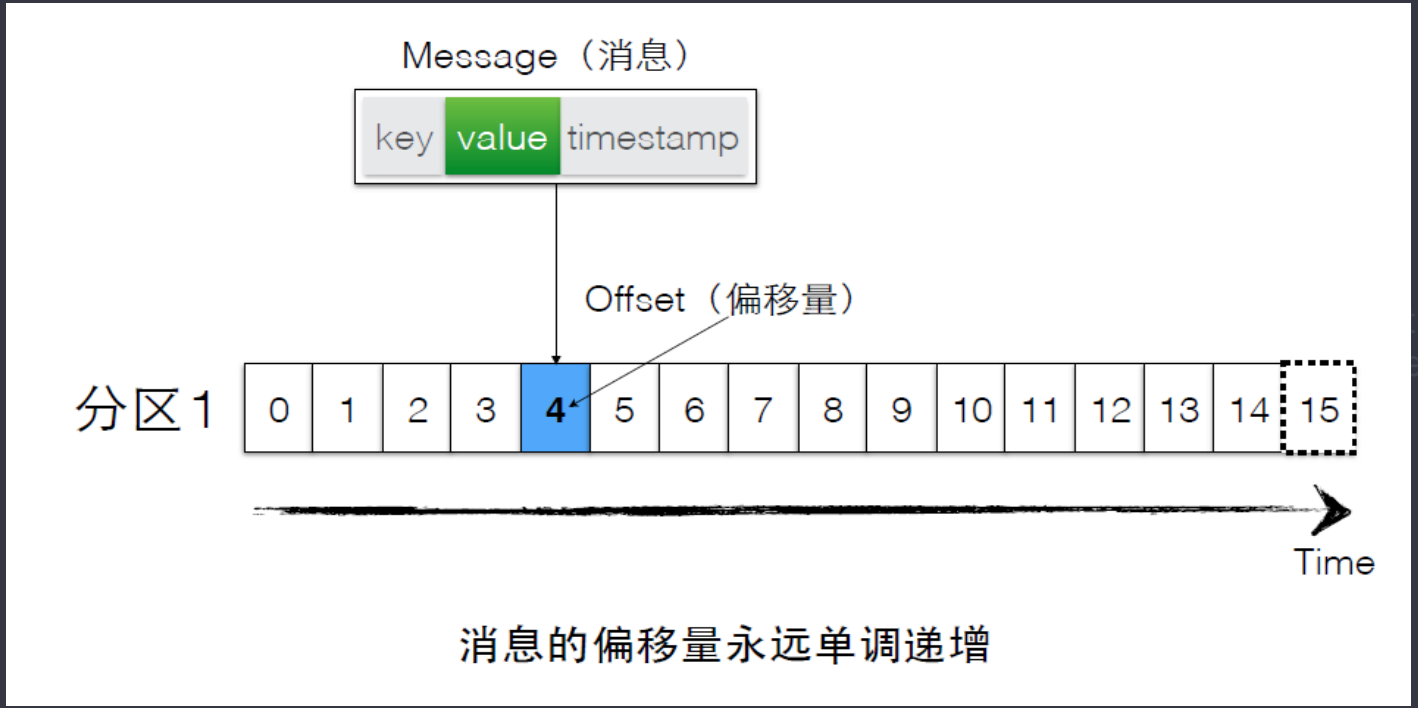
- 逻辑表
- 分库分表
- 表主键设置

Topic

- 分区数量
- 分区键

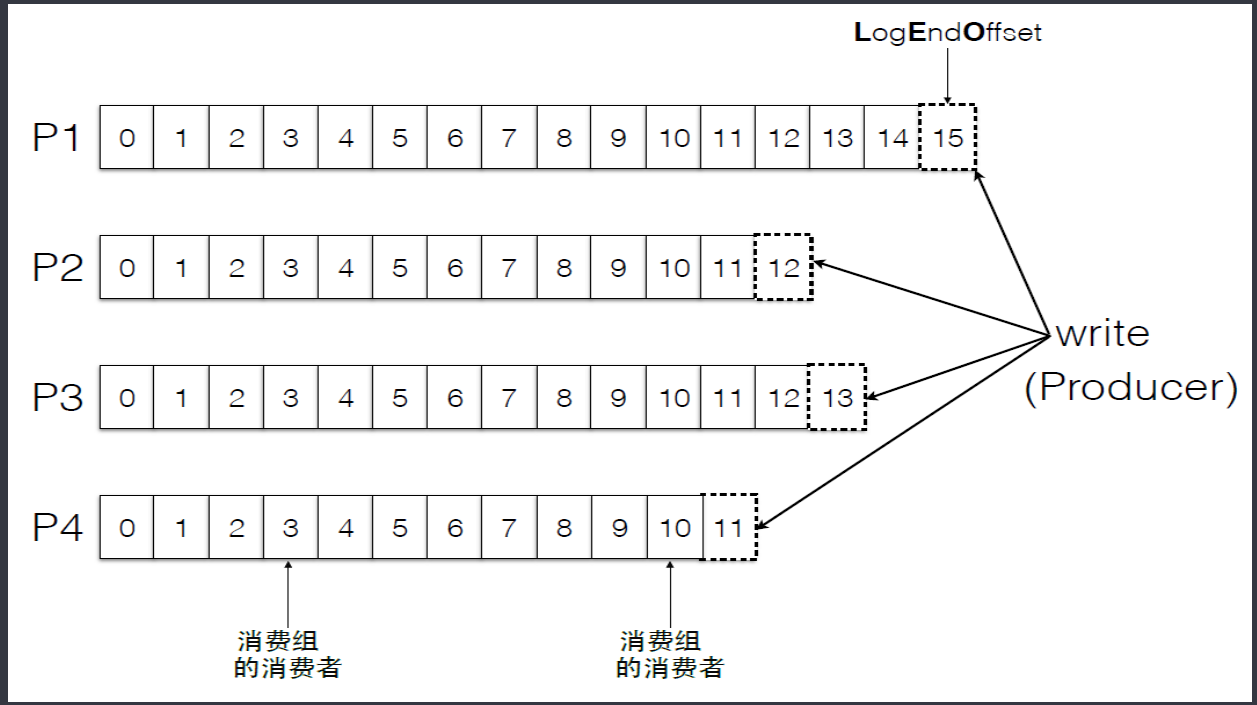
案例：Mysql同步到ES

Kafka：存储机制



案例：Mysql同步到ES

Kafka：消费机制



elastic

案例：Mysql同步到ES

同步程序：同步任务调度

同步调度配置

DB到ES数据映射

Kafka到ES的映射

案例：Mysql同步到ES

同步程序：同步任务调度

同步调度配置

DB到ES数据映射

Kafka到ES的映射

同步调度分配

调度算法分配

调度操作控制

调度指标监控

案例：Mysql同步到ES

同步程序：同步任务执行

Kafka模块

拉取同步数据

提交消费位置

案例：Mysql同步到ES

同步程序：同步任务执行

Kafka模块

拉取同步数据

提交消费位置

Mapper模块

DB表与ES索引映射

表字段与索引字段映射

案例：Mysql同步到ES

同步程序：同步任务执行

Kafka模块
拉取同步数据
提交消费位置

Mapper模块
DB表与ES索引映射
表字段与索引字段映射

Elastic模块
Bulk局部更新
设置doc_as_upsert：true

案例：Mysql同步到ES

同步程序：同步任务执行

Kafka模块
拉取同步数据
提交消费位置

Mapper模块
DB表与ES索引映射
表字段与索引字段映射

Elastic模块
Bulk局部更新
设置doc_as_upsert：true

Schedule模块
执行状态控制
执行状态指标

案例：Mysql同步到ES

数据同步全过程

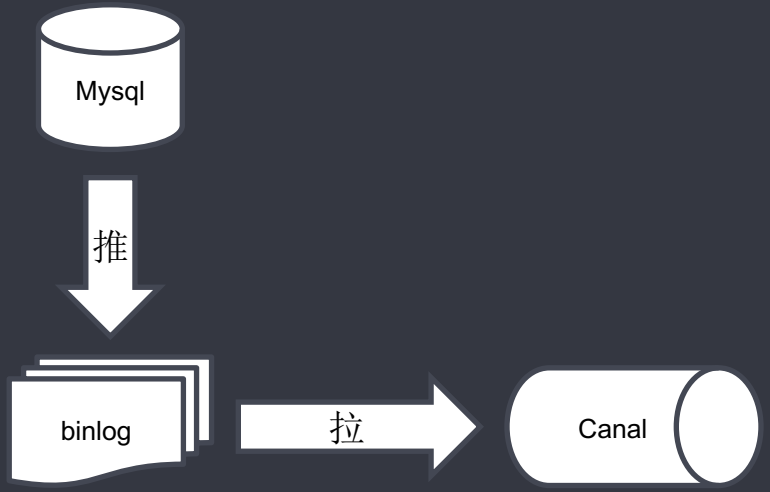
elastic

案例：Mysql同步到ES

数据同步全过程

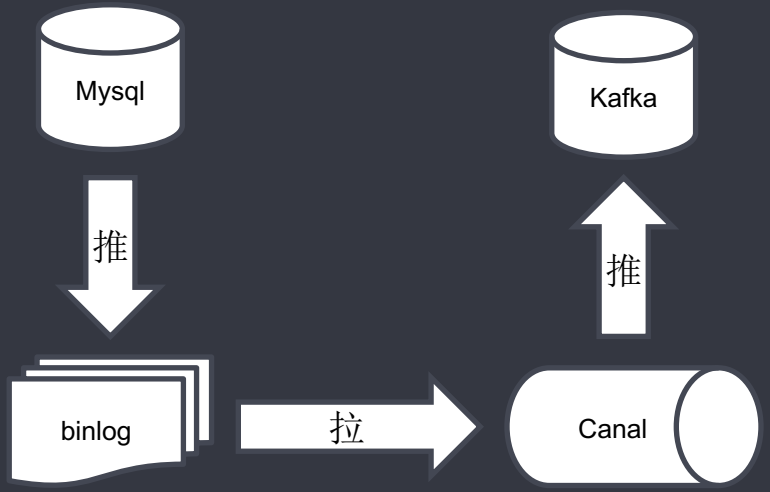
案例：Mysql同步到ES

数据同步全过程



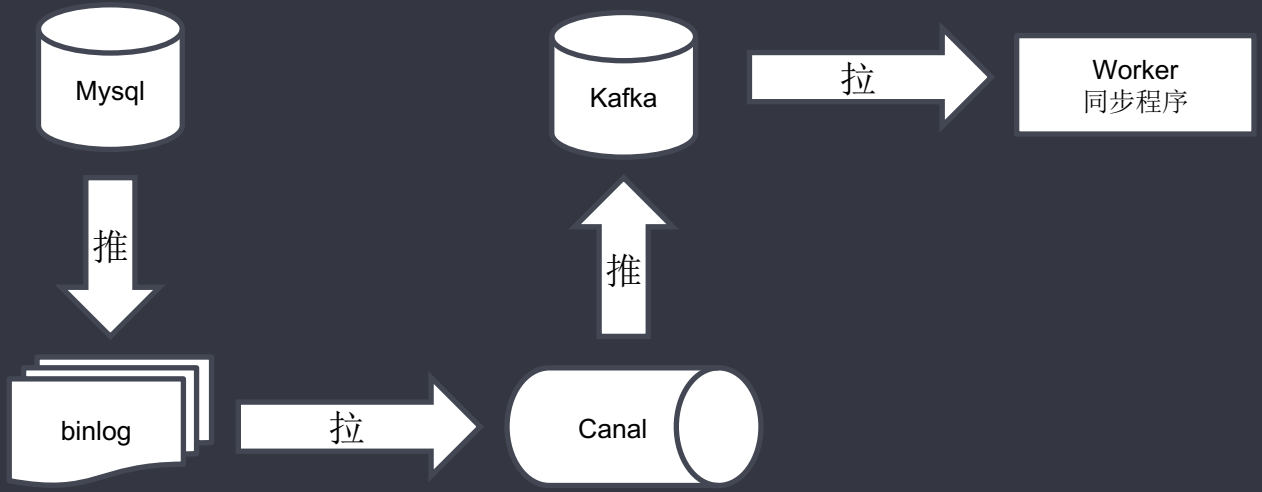
案例：Mysql同步到ES

数据同步全过程



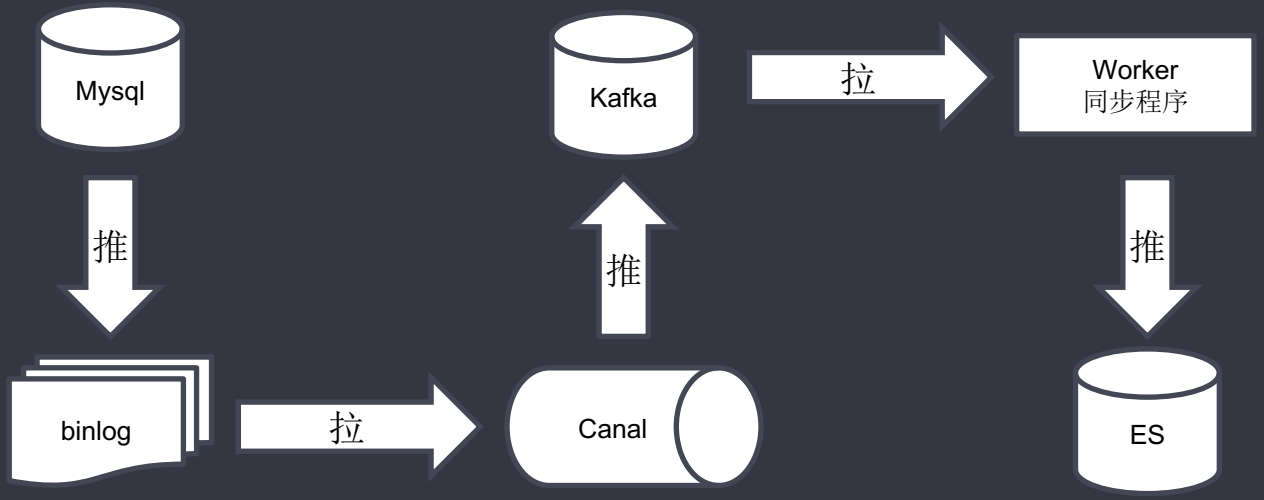
案例：Mysql同步到ES

数据同步全过程



案例：Mysql同步到ES

数据同步全过程



elastic

其它数据库同步

关系型数据库/非关系型数据库

1. Postgresql：logical decoding
2. Sqlserver：Change Data Capture/Change Tracking
3. Oracle：Redo log /Oracle GoldenGate
4. Mongodb：Replicate sets
5. Elasticsearch：Translog

其它数据库同步

关系型数据库/非关系型数据库

1. Postgresql：logical decoding
2. Sqlserver：Change Data Capture/Change Tracking
3. Oracle：Redo log /Oracle GoldenGate
4. Mongodb：Replicate sets
5. Elasticsearch：Translog

elastic

其它数据库同步

关系型数据库/非关系型数据库

1. Postgresql：logical decoding
2. Sqlserver：Change Data Capture/Change Tracking
3. Oracle：Redo log /Oracle GoldenGate
4. Mongodb：Replicate sets
5. Elasticsearch：Translog

elastic

其它数据库同步

关系型数据库/非关系型数据库

1. Postgresql：logical decoding
2. Sqlserver：Change Data Capture/Change Tracking
3. Oracle：Redo log /Oracle GoldenGate
4. Mongodb：Replicate sets
5. Elasticsearch：Translog

elastic

其它数据库同步

关系型数据库/非关系型数据库

1. Postgresql：logical decoding
2. Sqlserver：Change Data Capture/Change Tracking
3. Oracle：Redo log /Oracle GoldenGate
4. Mongodb：Replicate sets
5. Elasticsearch：Translog

elastic

04.总结与展望

注意事项/后续计划

注意事项

- 瞬间批量更新DB数据
- CDC性能瓶颈
- 同步程序性能瓶颈
- 数据反复覆盖变化

注意事项

DB刷数据问题

- 瞬间批量更新DB数据
- CDC性能瓶颈
- 同步程序性能瓶颈
- 数据反复覆盖变化

DB多表关联深度

- 关联深度影响索引性能
- 反向关联影响同步性能

注意事项

DB刷数据问题

- 瞬间批量更新DB数据
- CDC性能瓶颈
- 同步程序性能瓶颈
- 数据反复覆盖变化

DB多表关联深度

- 关联深度影响索引性能
- 反向关联影响同步性能

ES高级类型限制

- Array对象类型
- Nested对象类型
- Join类型
- Shape类型
- 高级类型转换

问题遗留

- DB与ES数据自动比对校验
- 经济高效的方案探讨

elastic

问题遗留

数据校验

- DB与ES数据自动比对校验
- 经济高效的方案探讨

数据修复

- DB与ES数据一致性自动修复
- 经济高效方案探讨

问题遗留

数据校验

- DB与ES数据自动比对校验
- 经济高效的方案探讨

数据修复

- DB与ES数据一致性自动修复
- 经济高效方案探讨

技术演进

- 引入Flink技术平台

总结与延伸

DB与ES的关系

1. ES在大多数应用场景可以完全替代DB
2. DB适合场景 ？强ACID
3. ES适合场景？高效查询
4. DB与ES混合，DB解决ACID问题，ES解决高效查询
5. 数据实时交换平台需求，满足多种DB数据任意交换

总结与延伸

DB与ES的关系

1. ES在大多数应用场景可以完全替代DB
2. DB适合场景 ？强ACID
3. ES适合场景？高效查询
4. DB与ES混合，DB解决ACID问题，ES解决高效查询
5. 数据实时交换平台需求，满足多种DB数据任意交换

总结与延伸

DB与ES的关系

1. ES在大多数应用场景可以完全替代DB
2. DB适合场景 ？强ACID
3. ES适合场景？高效查询
4. DB与ES混合，DB解决ACID问题，ES解决高效查询
5. 数据实时交换平台需求，满足多种DB数据任意交换

总结与延伸

DB与ES的关系

1. ES在大多数应用场景可以完全替代DB
2. DB适合场景 ？强ACID
3. ES适合场景？高效查询
4. DB与ES混合，DB解决ACID问题，ES解决高效查询
5. 数据实时交换平台需求，满足多种DB数据任意交换

总结与延伸

DB与ES的关系

1. ES在大多数应用场景可以完全替代DB
2. DB适合场景 ？强ACID
3. ES适合场景？高效查询
4. DB与ES混合，DB解决ACID问题，ES解决高效查询
5. 数据实时交换平台需求，满足多种DB数据任意交换

谢谢！

elastic