

WOTA

51CTO

World Of Tech 2017

全球架构与运维技术峰会

2017年4月14日-15日 北京富力万丽酒店

ARCHITECTURE



出品人及主持人：

**赖春波** 滴滴出行平台技术部  
高级技术总监

---

高可用架构

# 网易NDC高可用实践

主讲人：马进



**马进** 网易  
资深工程师

---

**分享主题：**  
网易数据传输服务NDC高可用实践

# NDC是什么？

- NDC，直译为网易数据运河，为网易各大产品提供结构化数据实时迁移，同步和订阅服务。功能包括异构数据库在线迁移，OLTP到OLAP的实时数据整合，数据库增量数据实时订阅等。是网易分布式数据库DDB和大数据解决方案Mammut的依赖组件



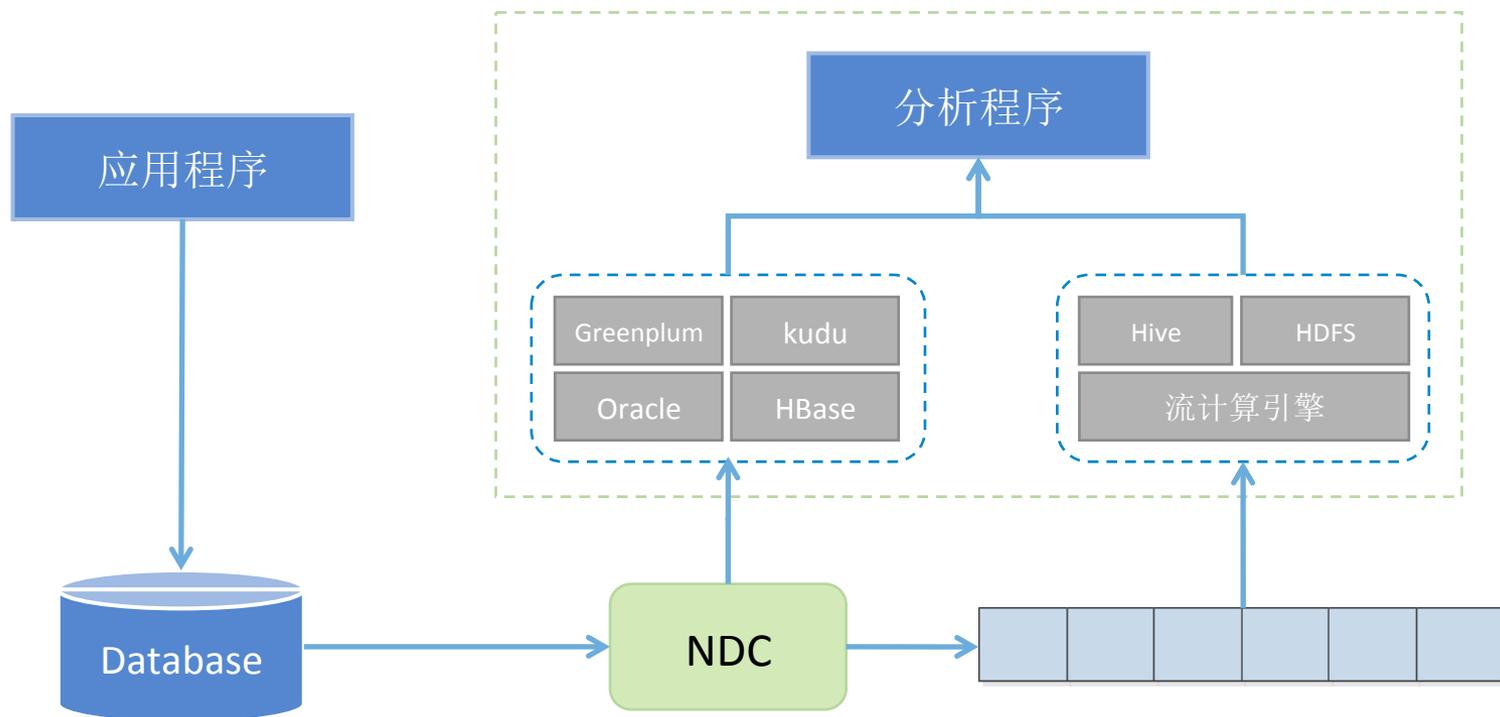
# 大纲

- 应用场景 why?
- 产品形态 what?
- 系统架构 how?
- 高可用实践

# NDC应用场景

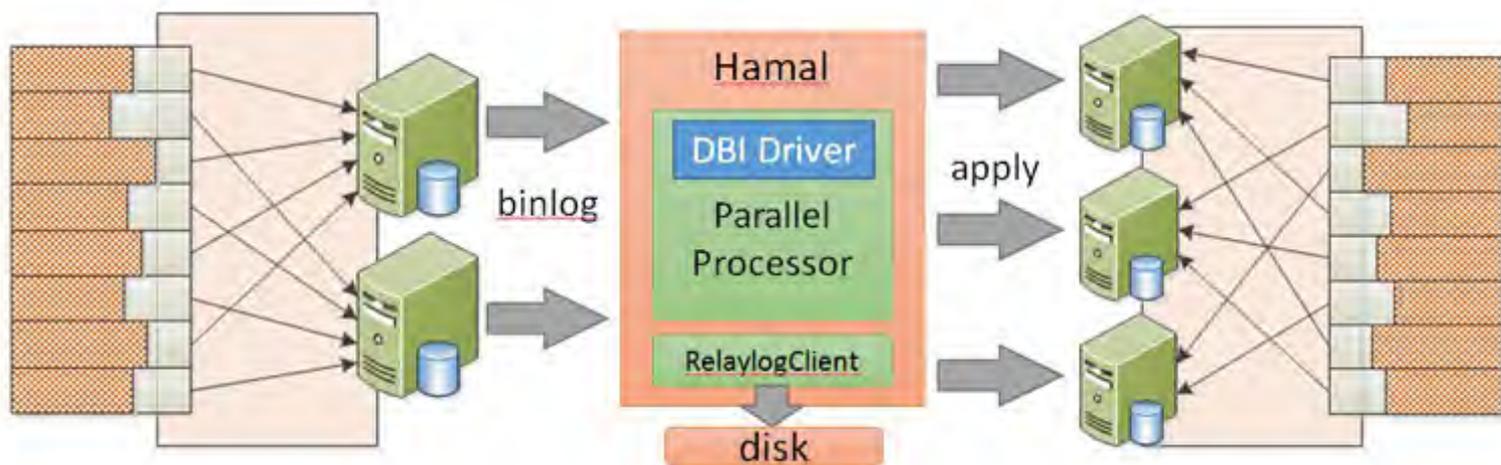
- OLAP数据同步

- OLTP实时同步到OLAP系统（代表：Kudu, HBase, Greenplum）
- OLTP同步到队列，定期merge到OLAP系统（代表：Hive, HDFS）



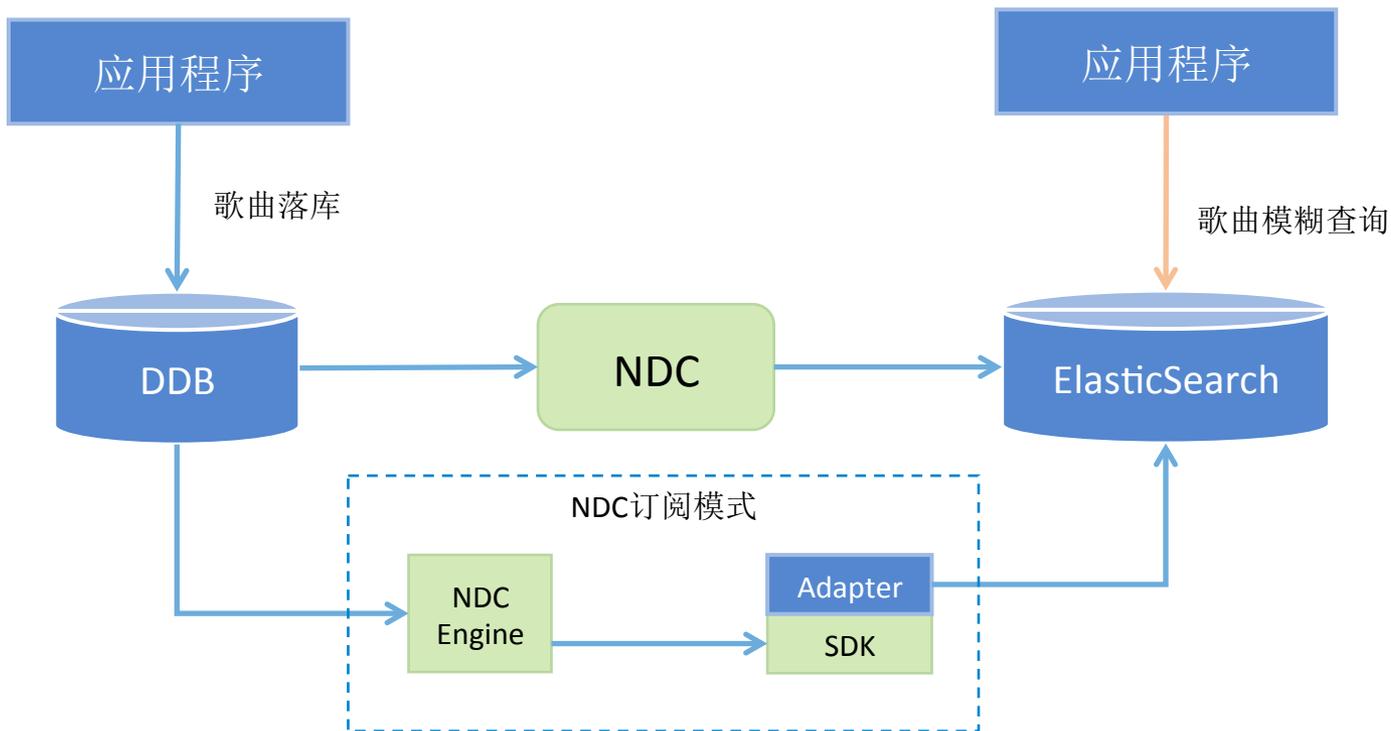
# NDC应用场景

- DDB在线数据迁移
  - 场景：在线扩容，机器迁移，更改均衡字段.....



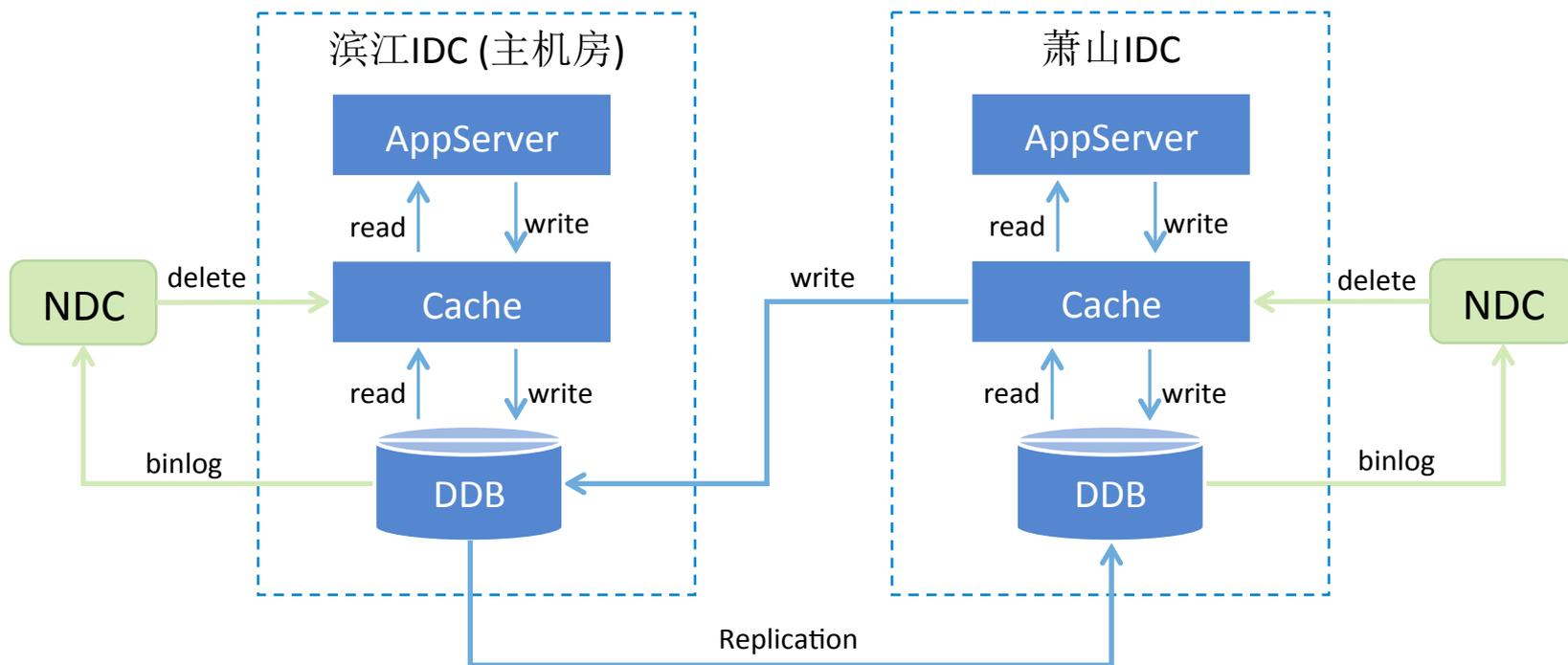
# NDC应用场景

- 数据库第三方索引支持
  - 全文索引，地图索引等



# NDC应用场景

- 多机房缓存淘汰
  - 原则：同步先于淘汰



# 小结

- 应用方视角

- 数据迁移：异构数据库在线迁移，在线扩缩容
- 数据同步：跨机房，跨域，跨国的实时数据同步
- 数据订阅：数据驱动业务，业务间异步解耦

- 大数据视角

- 数据整合：OLTP到OLAP的数据整合
- 数据集成：*“making all the data an organization has available in all its services and systems”*

- 核心需求

- 获取数据库实时变更的能力
- 数据发布的能力

# NDC产品形态

- 平台化

- 平台化的WEB管理工具
- 平台化的资源管理和调度
- 平台化报警监控

- 插件化

- 不同数据源extractor插件化
- 不同数据源applier插件化
- 账号系统插件化

# NDC产品形态

Q 请输入JobID							
<input type="button" value="启动"/> <input type="button" value="暂停"/> <input type="button" value="停止"/> <input type="button" value="释放"/>							
<input type="checkbox"/>	任务名称	任务状态	创建时间	最近上报时间	执行地址	全量迁移	增量延迟
<input type="checkbox"/>	yixin_archive_db53	Initfailed	2016/12/27 17:23:14	2016/12/27 17:23:14		<div style="width: 0%;"></div> 0%	0 ms
<input type="checkbox"/>	yixin_archive_db51	Initfailed	2016/12/27 17:23:14	2016/12/27 17:23:14		<div style="width: 0%;"></div> 0%	0 ms
<input type="checkbox"/>	yixin_archive_db49	Initfailed	2016/12/27 17:23:14	2016/12/27 17:23:14		<div style="width: 0%;"></div> 0%	0 ms
<input type="checkbox"/>	yidun_migrate_rds-1	Suspend	2016/12/17 11:06:12	2017/01/02 22:45:28	10.171.160.52:7100	<div style="width: 100%;"></div> 100%	588 ms
<input type="checkbox"/>	yidun_migrate_rds-3	Suspend	2016/12/17 11:06:12	2017/01/02 22:45:28	10.171.160.52:7100	<div style="width: 100%;"></div> 100%	897 ms
<input type="checkbox"/>	yidun_migrate_rds-2	Suspend	2016/12/17 11:06:12	2017/01/02 22:45:28	10.171.160.52:7100	<div style="width: 100%;"></div> 100%	606 ms
<input checked="" type="checkbox"/>	nim_rds_4_online	Alive	2016/12/09 11:04:38	2017/01/02 22:45:28	10.171.160.52:7100	<div style="width: 100%;"></div> 100%	194 ms
<input type="checkbox"/>	nim_rds_3_online	Alive	2016/12/09 11:00:10	2017/01/02 22:45:28	10.171.160.51:7100	<div style="width: 100%;"></div> 100%	165 ms
<input type="checkbox"/>	nim_rds_2_online	Alive	2016/12/09 10:20:07	2017/01/02 22:45:28	10.171.160.52:7100	<div style="width: 100%;"></div> 100%	0 ms
<input type="checkbox"/>	nim_rds_1_online	Alive	2016/12/09 09:53:08	2017/01/02 22:45:28	10.171.160.51:7100	<div style="width: 100%;"></div> 100%	0 ms

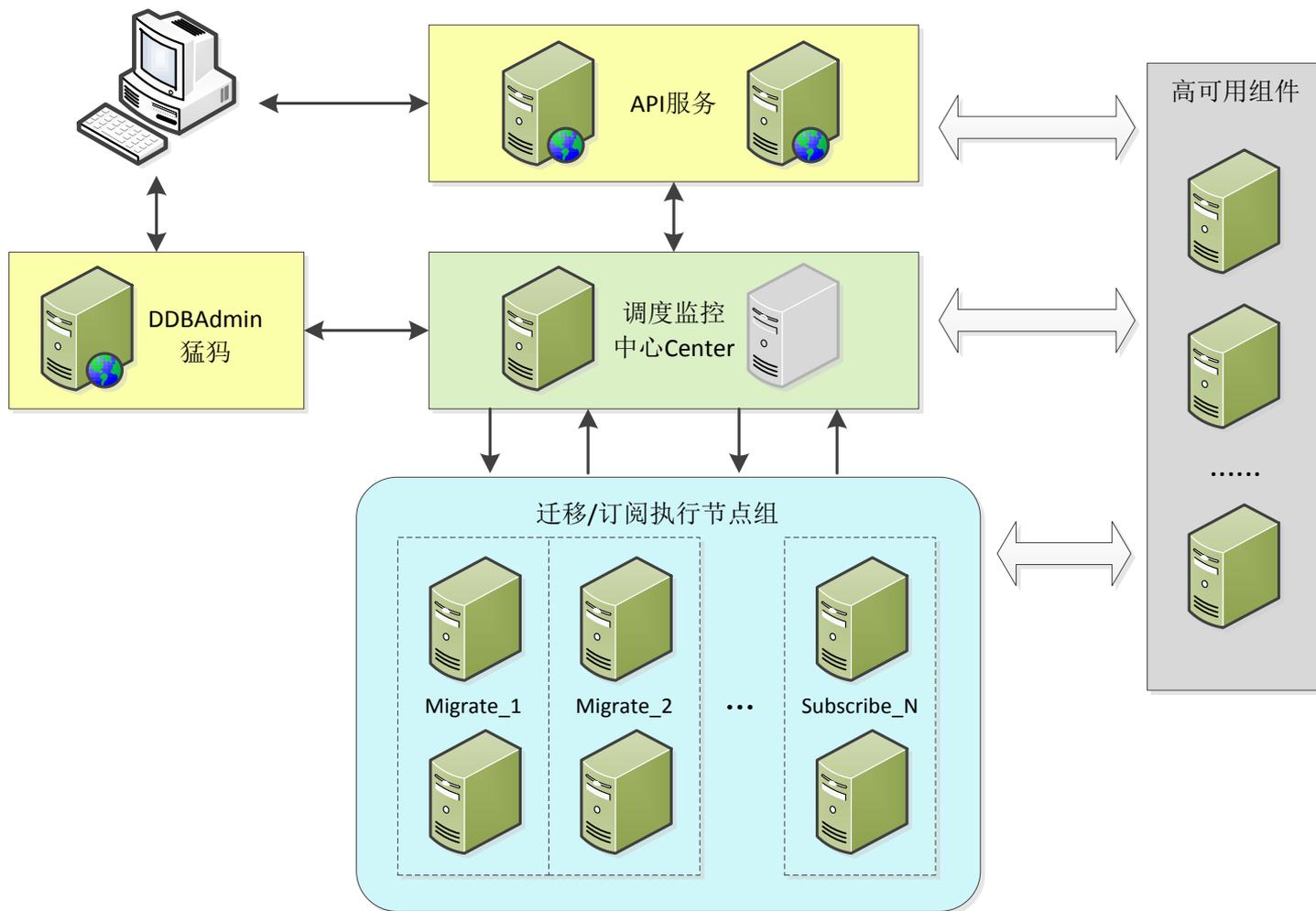
  

全量迁移		增量迁移		预检查	
源端最新位置	mysql-bin.001737:388808496	待迁移日志大小(字节)	3,521	迁移速度(字节/秒)	351,446
最新推进位置	mysql-bin.001737:388804975	任务ID	16	任务状态	Alive
最后迁移位置	mysql-bin.001737:388804500	开始时间	2016/12/16 16:44:21	进度(%)	100
延迟时间(ms)	194 ms				
预计时间(ms)	10 ms				
已缓存日志大小(字节)	389,681,441				

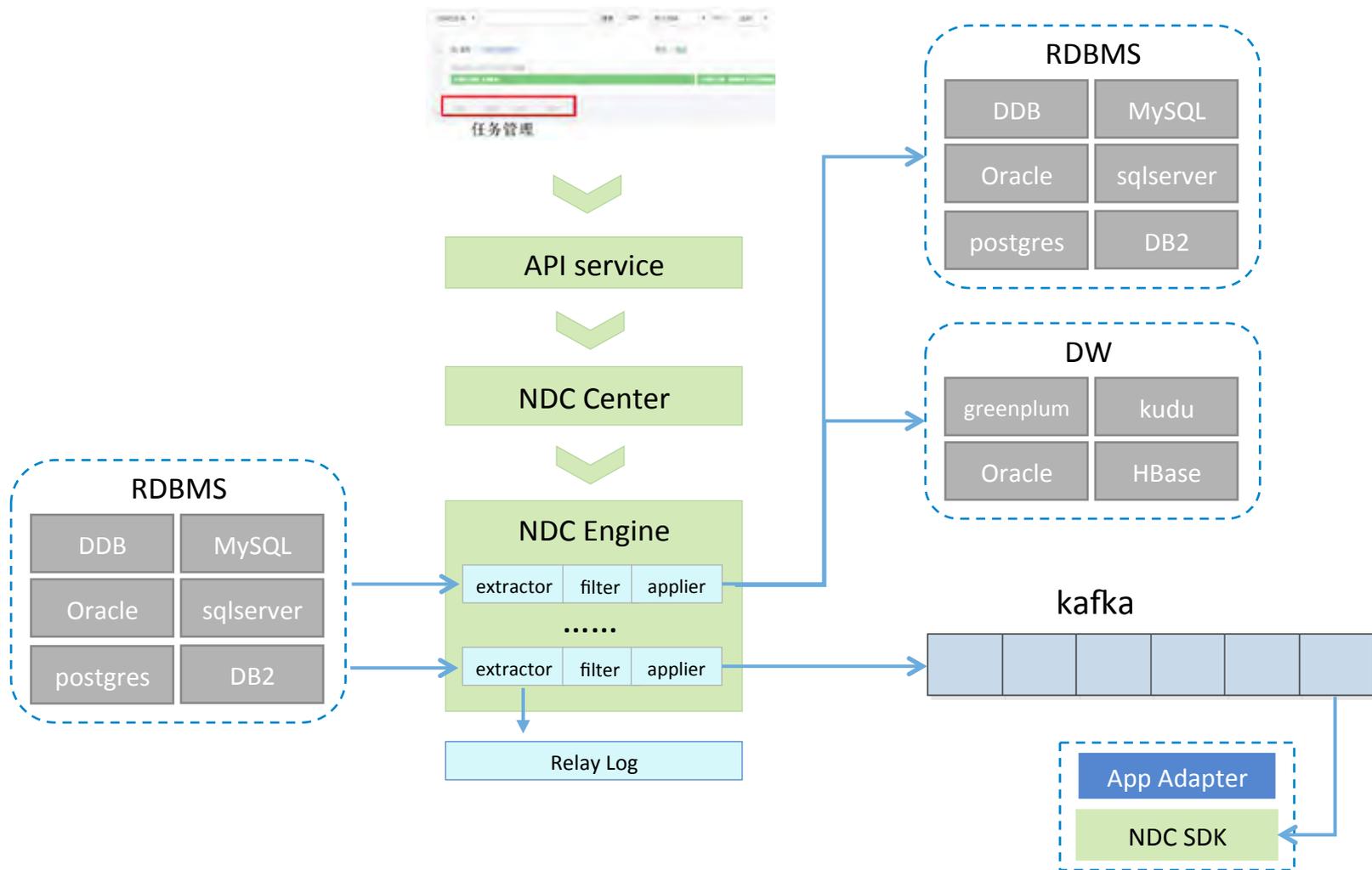
# NDC猛犸定位



# 系统架构

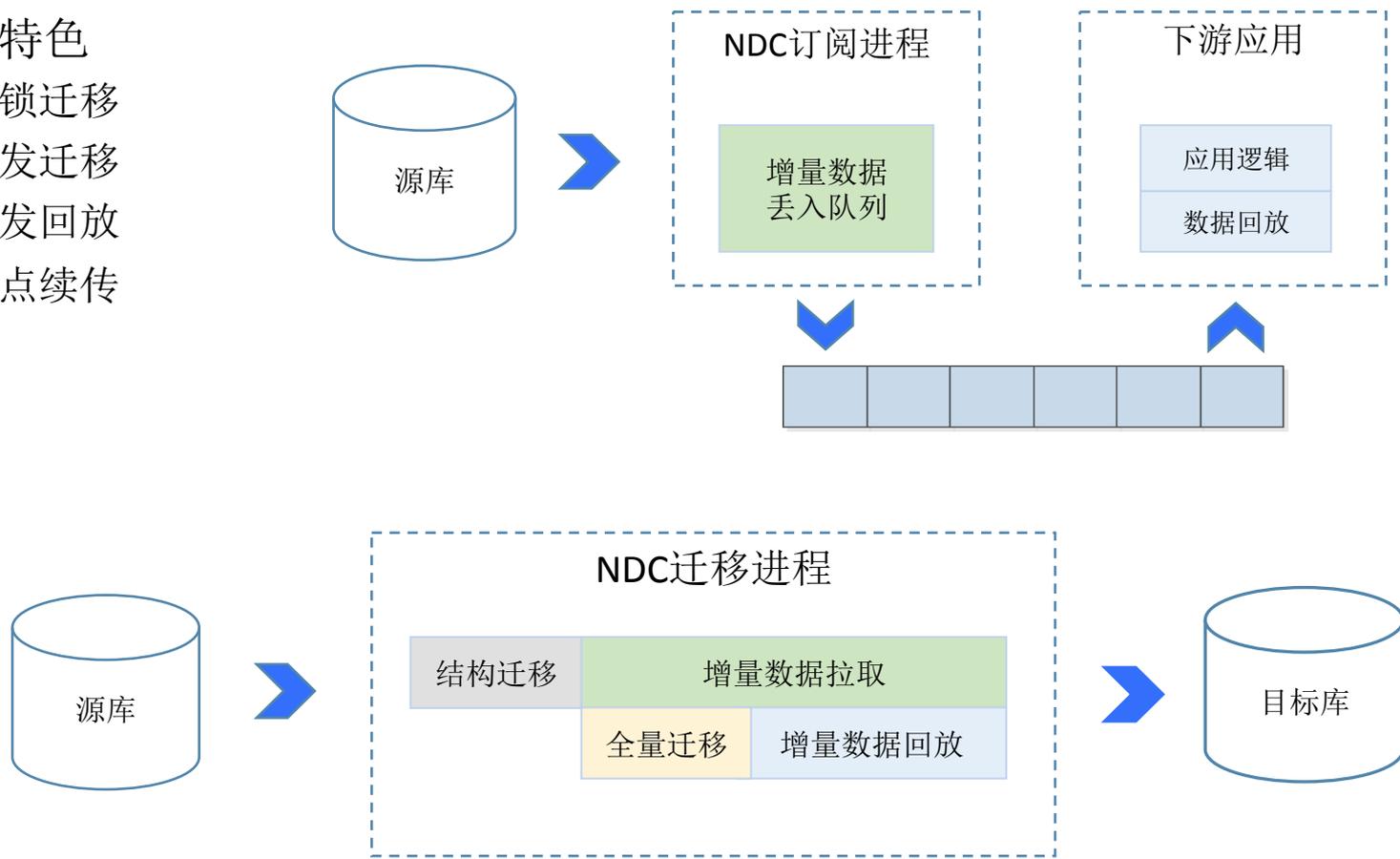


# 系统架构

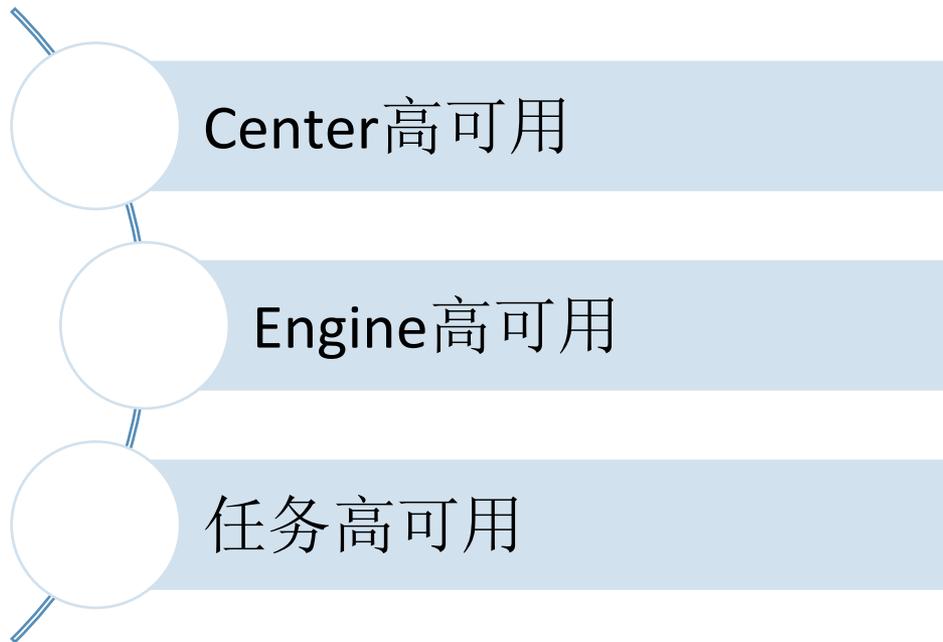


# NDC原理

- 服务特色
  - 无锁迁移
  - 并发迁移
  - 并发回放
  - 断点续传

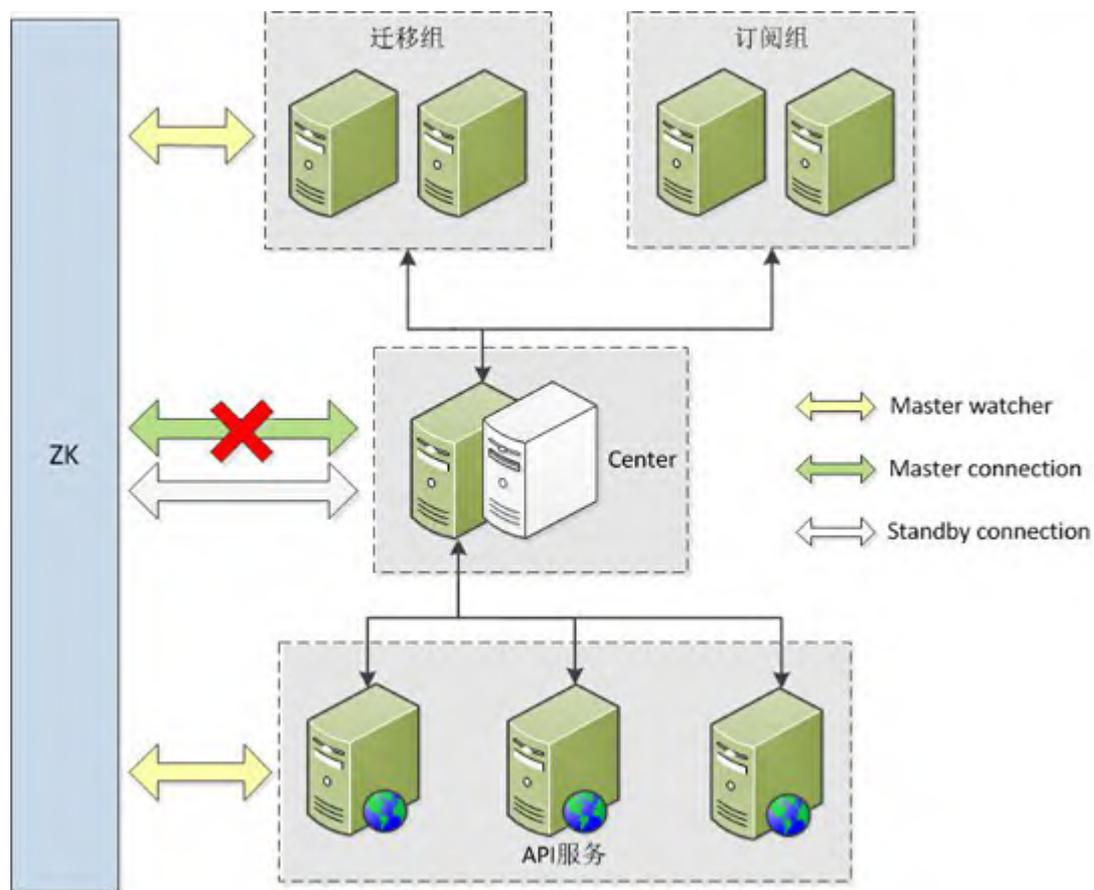


# 高可用实践



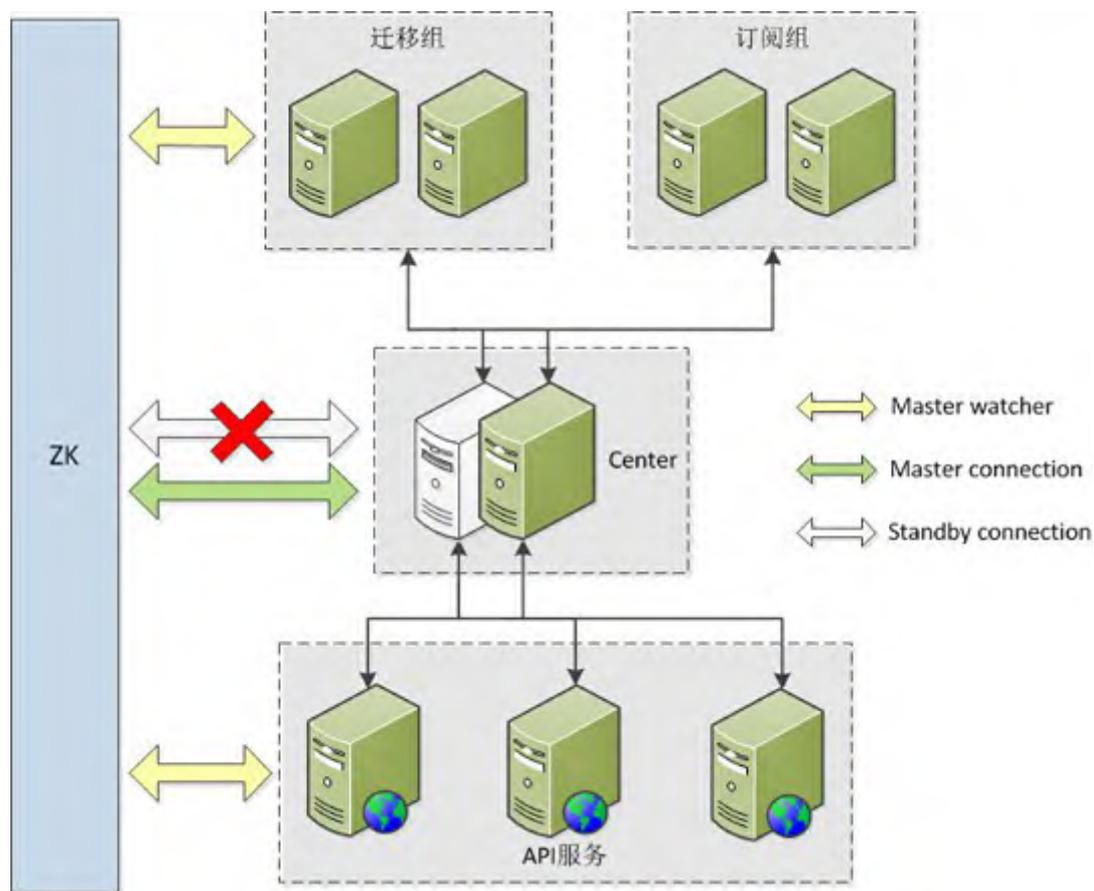
# Center高可用

- 解决方案
  - Zookeeper + Cruator
  - LeaderSelector
  - PathCache
- 类似方案
  - Keepalived
- 存在问题
  - Brain split



# Center高可用

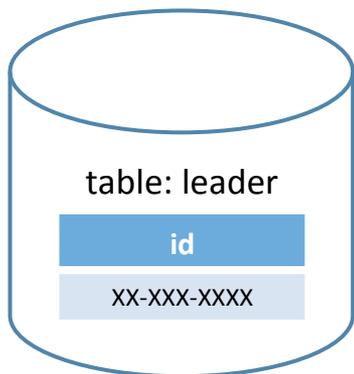
- 解决方案
  - Zookeeper + Cruator
  - LeaderSelector
  - PathCache
- 类似方案
  - Keepalived
- 存在问题
  - Brain split



# 脑裂解决方案

- 基本前提
  - Center状态存在系统库
  - 系统库是高可用的
- 基本方案
  - Lock leader为每个操作上锁
  - Switch leader时先lock leader

Lock leader:  
select id from leader for update



## Center operation

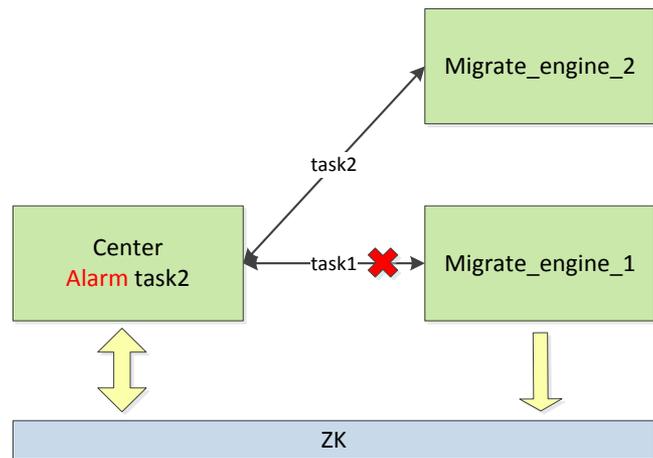
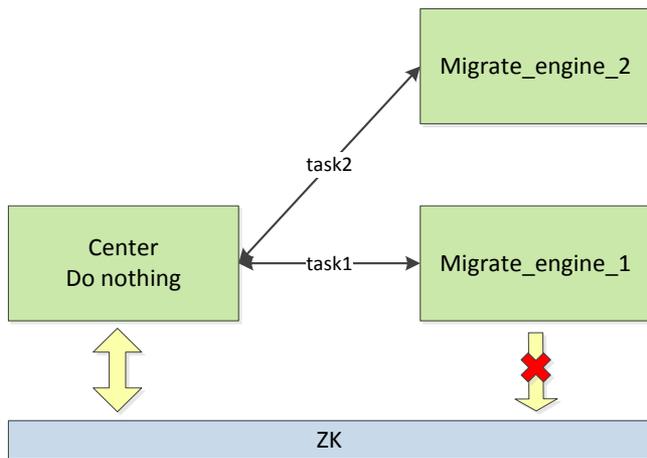
```
lock leader;  
if (leader id matched)  
    Operate;  
else  
    return error;  
commit/rollback;
```

## Switch leader

```
lock leader;  
update leader id;  
Commit;  
load meta data;  
Initialize server  
.....
```

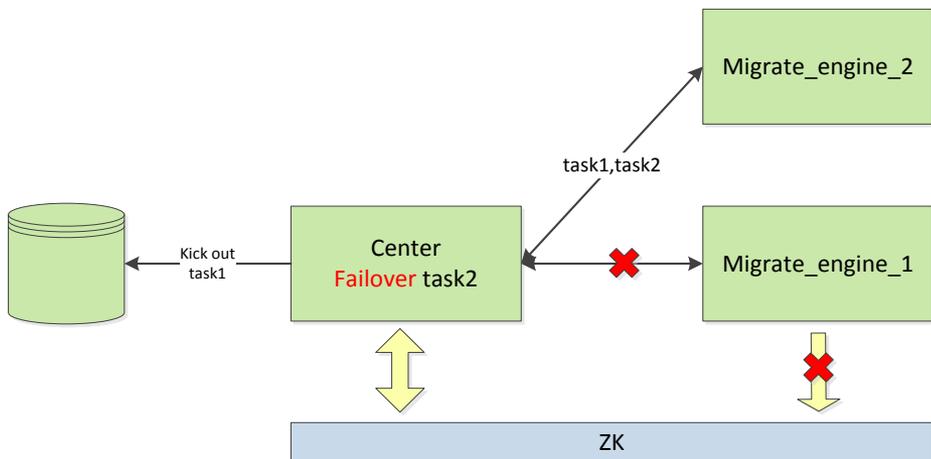


# Engine高可用



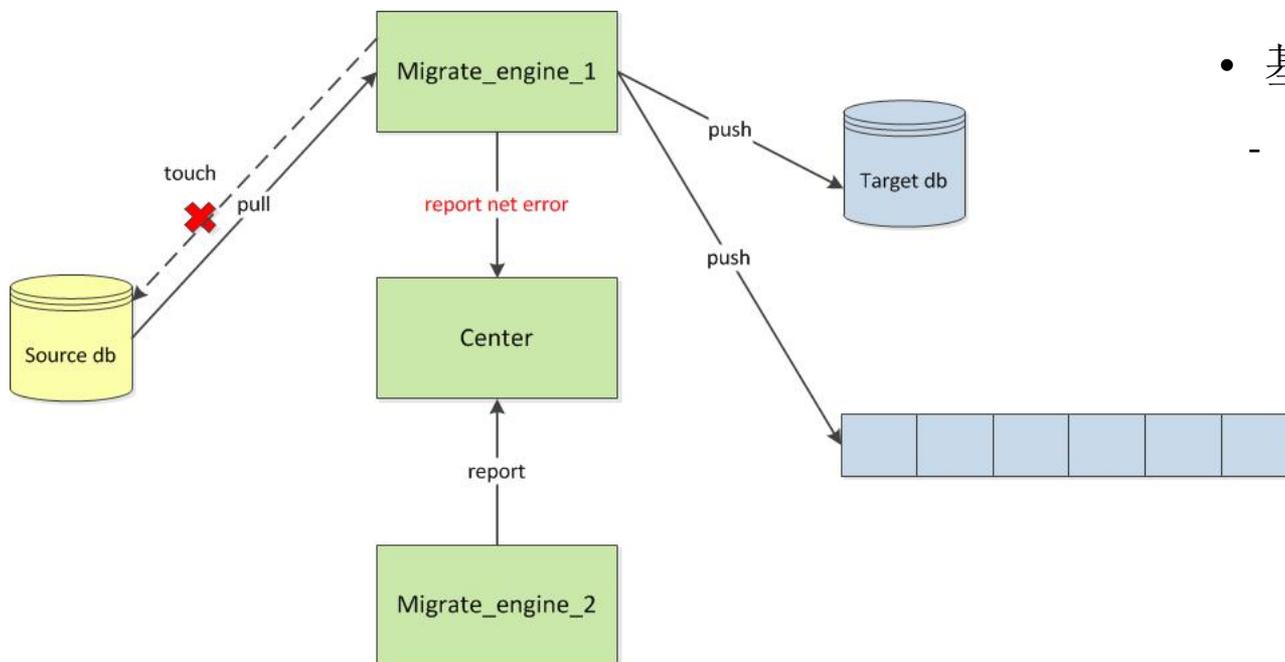
## • 要点

- 租约与ZK双重验证
- kick out先于fail over



# 任务高可用

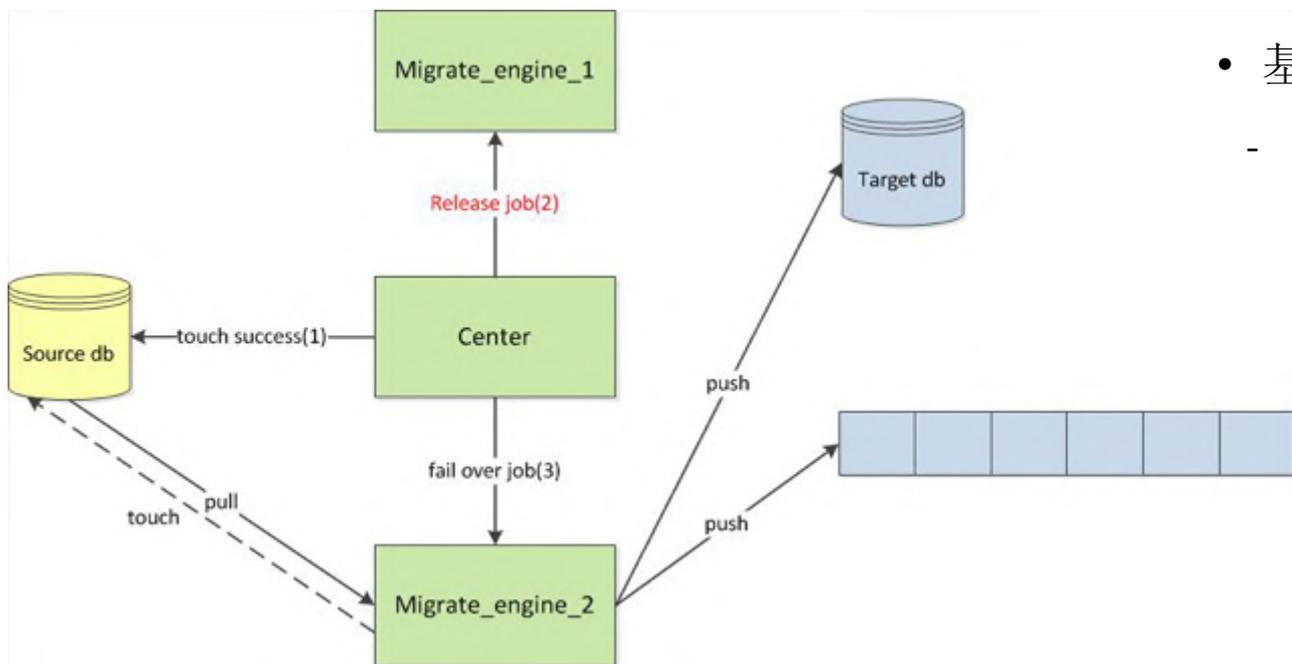
- 要点
  - 执行进程定期探活源端
  - 探活失败后先本地重试N次再上报



- 基本前提
  - 断点续传

# 任务高可用

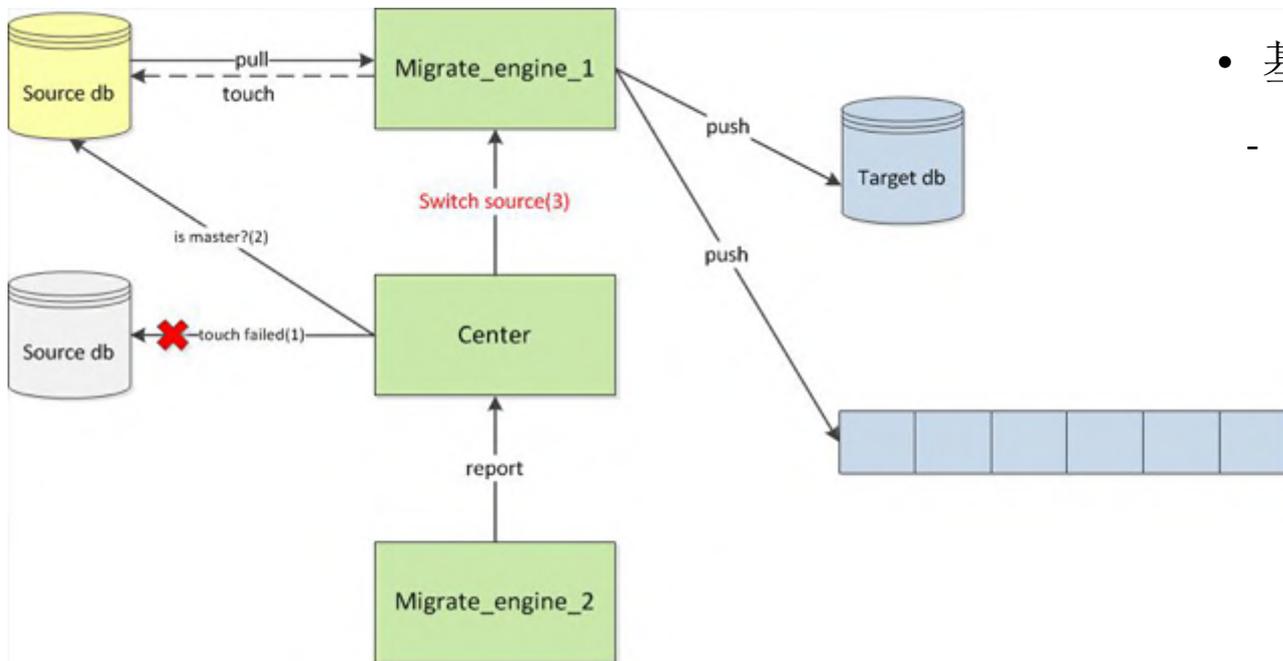
- 要点
  - Center再次探活源端，成功
  - Job fail over到Engine2



- 基本前提
  - 断点续传

# 源端漂移

- 要点
  - Center再次探活源端，成功
  - Job fail over到Engine2



- 基本前提
  - 断点续传

# 小结

- 设计原则
  - 监控先于高可用
  - 高可用分层，不过度设计
  - 高可用插件化，保持系统精简
  - 多重验证，避免误切
  
- 源端漂移问题
  - 如何保证数据不丢？
  - GTID, 基于触发器

Thank you !