2019
05
08-10
北京新云南皇冠假日酒店

数据风云十年变迁

DTCC 第十届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2019

IT168 ChinaUnix ITPUB

京东弹性数据库探索实践

郭伟
京东 技术架构部

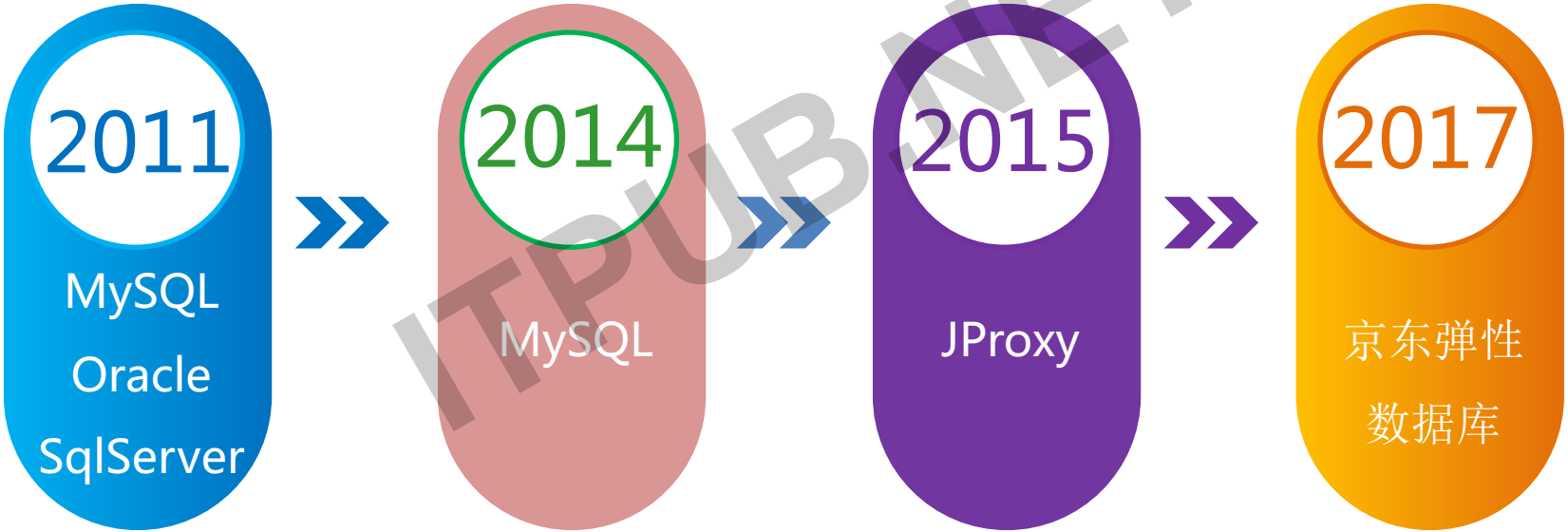
京东技术架构部

作为京东零售的技术基础设施，技术架构部专注于大规模分布式系统的设计开发与工程实施，重点建设数据中心集群管理和大规模计算资源调度、商品图片系统以及智能化、分布式存储与数据库系统、消息队列与服务框架、日志平台与监控系统等，持续推进京东零售整体系统架构的技术突破与创新。

- 发展历程
- 整体架构
- 集群调度
- Resharding
- 故障迁移

- 发展历程
- 整体架构
- 集群调度
- Resharding
- 故障迁移

发展历程

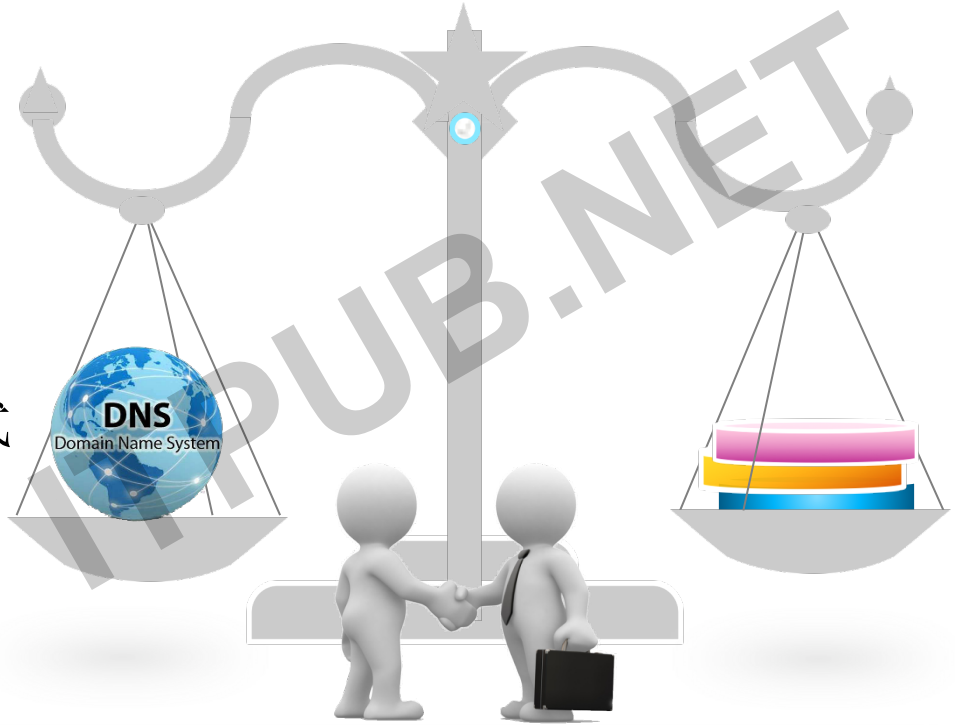


痛点

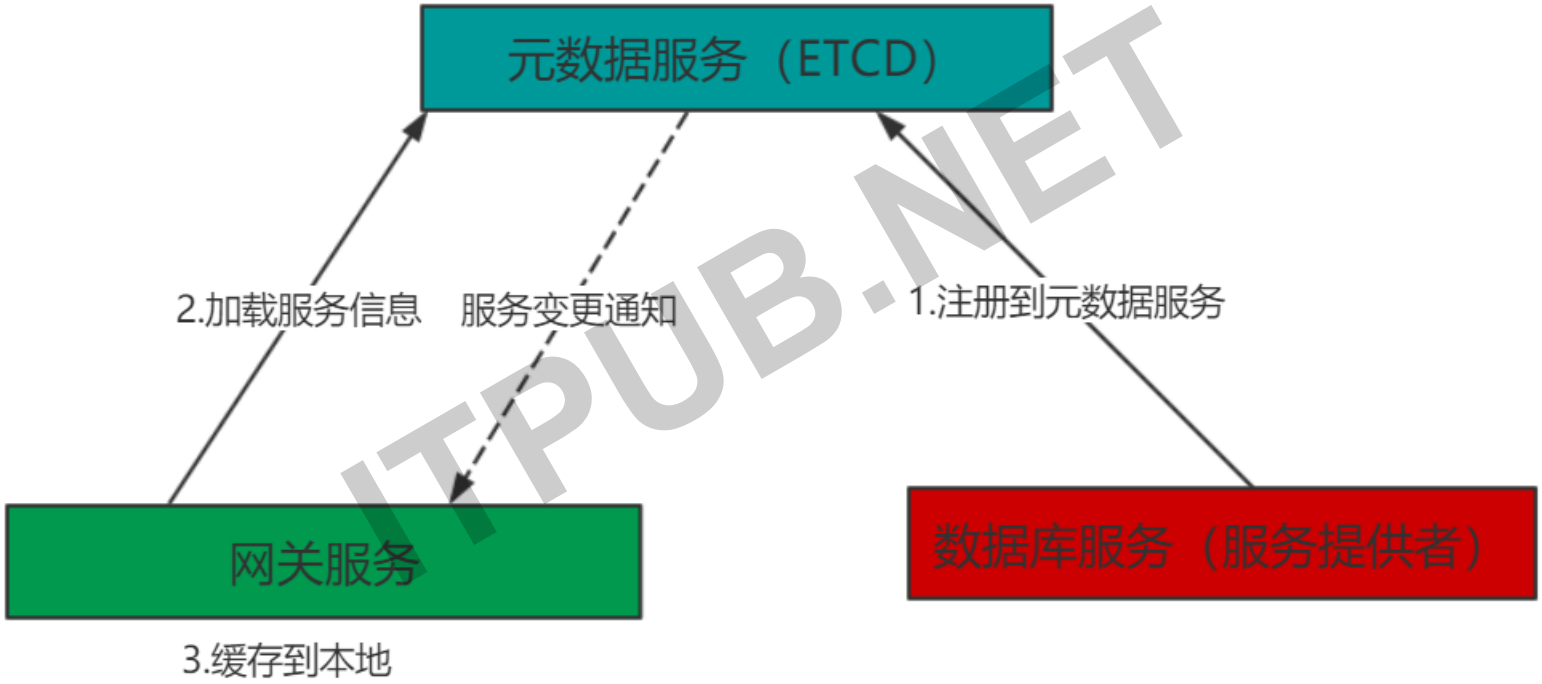
不是在迁数据
就是在迁数据的路上

痛点

基于DNS方式

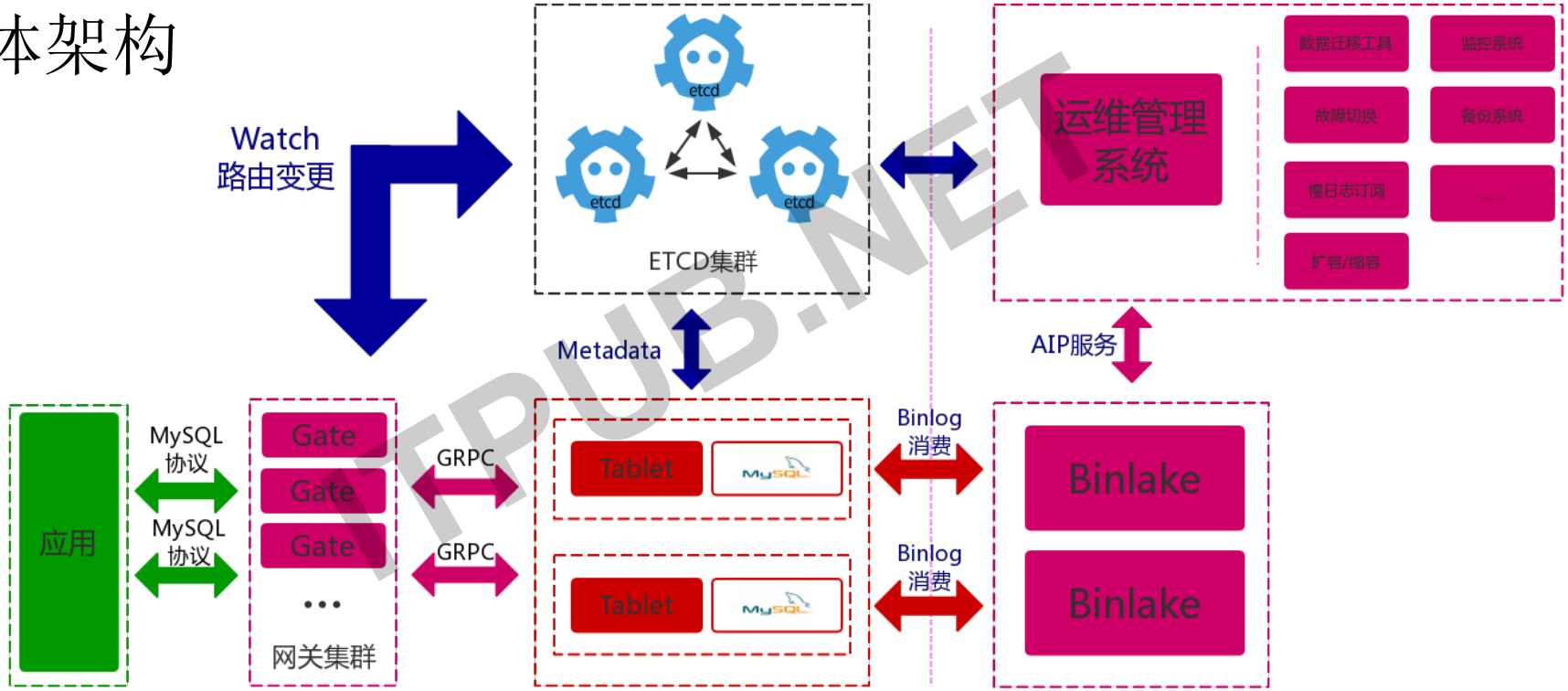


基于服务发现

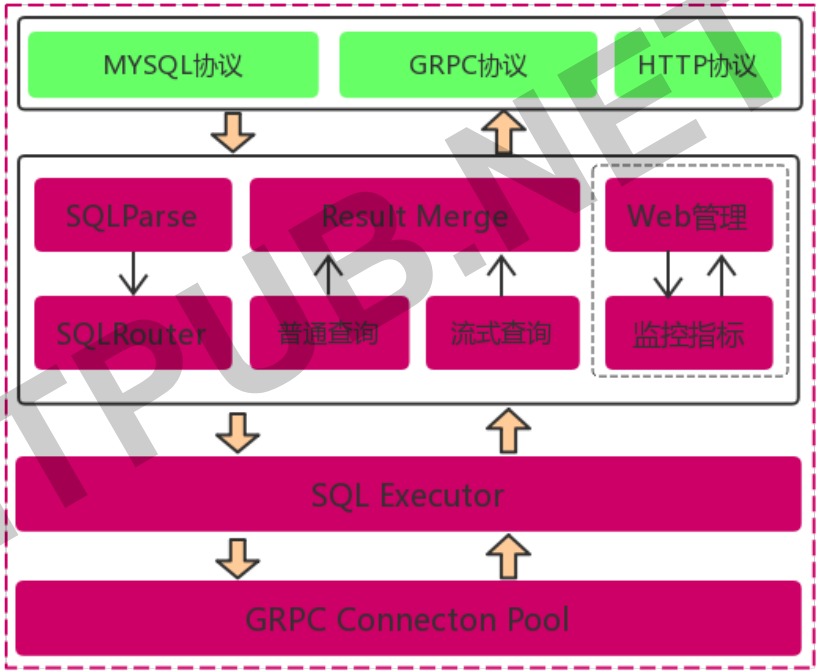

元数据服务（ETCD）
2.加载服务信息
服务变更通知
1.注册到元数据服务
网关服务
数据库服务（服务提供者）
3.缓存到本地

- 发展历程
- 整体架构
- 集群调度
- Resharding
- 故障迁移

整体架构

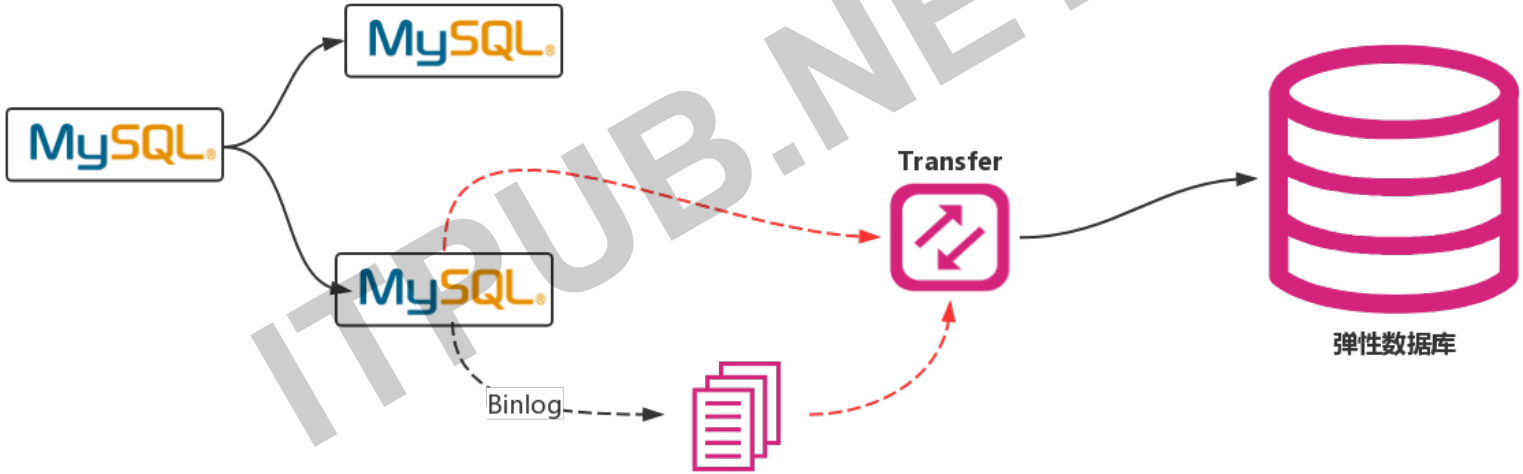


网关架构



Transfer架构

数据迁移工具

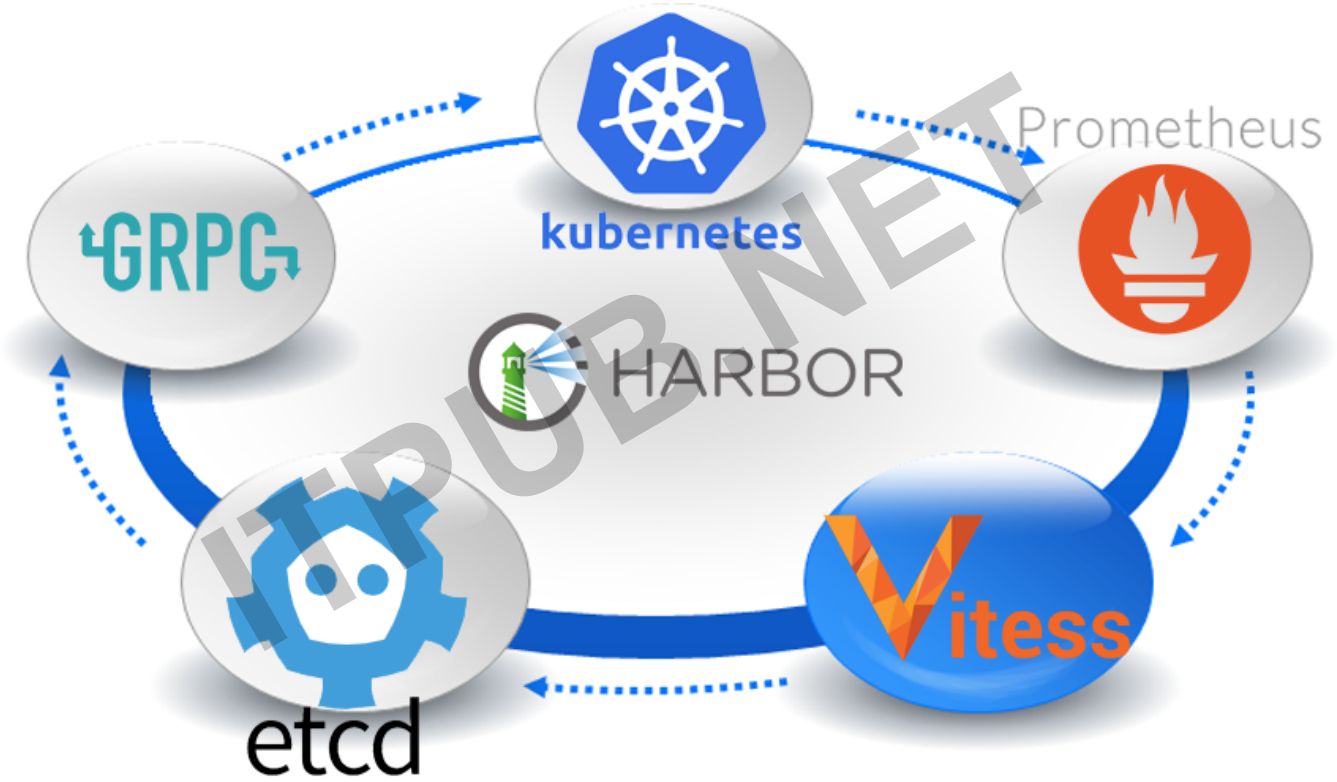


- 发展历程
- 整体架构
- 集群调度
- Resharding
- 故障迁移

Cloud Native定义

CNCF对于Cloud Native应用的定义，是使用开源工具栈来实现如下特征:

- 容器化（Containerized）：每个部分（应用程序，进程等）都封装在自己的容器中。这有助于重复性，透明度和资源隔离。
- 动态编排（Dynamically orchestrated）：动态的调度和管理容器以优化资源利用。
- 面向微服务（Microservices oriented）：应用程序基于微服务架构，显著提高架构演进的灵活性和可维护性。



调度策略

反亲和性AntiAffinity:

- 匹配有更多的逻辑组合，不只是字符的完全相等
- 调度分成软策略（soft）和硬策略（hard），在软策略的情况下，如果没有满足调度条件的节点，pod 会忽略这条规则，继续完成调度过程。在硬策略的情况下，如果没有满足条件的节点，就不断重试；
- 软策略：preferredDuringSchedulingIgnoredDuringExecution
- 硬策略： requiredDuringSchedulingIgnoredDuringExecution

可选的操作符

- In：label 的值在某个列表中
- NotIn：label 的值不在某个列表中
- Exists：某个 label 存在
- DoesNotExist: 某个 label 不存在
- Gt：label 的值大于某个值（字符串比较）
- Lt：label 的值小于某个值（字符串比较）

```yaml
affinity:
    podAntiAffinity:
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 100
        podAffinityTerm:
          labelSelector:
            matchExpressions:
            - key: shard
              operator: In
              values:
              - {{.ShardLabel}}
            - key: keyspace
              operator: In
              values:
              - "{{.Keyspace}}"
          topologyKey: jdos.jd.com/rack
```

调度方案

方案1：  kubernetes + Vitess

方案2：  HashiCorp Nomad + Vitess

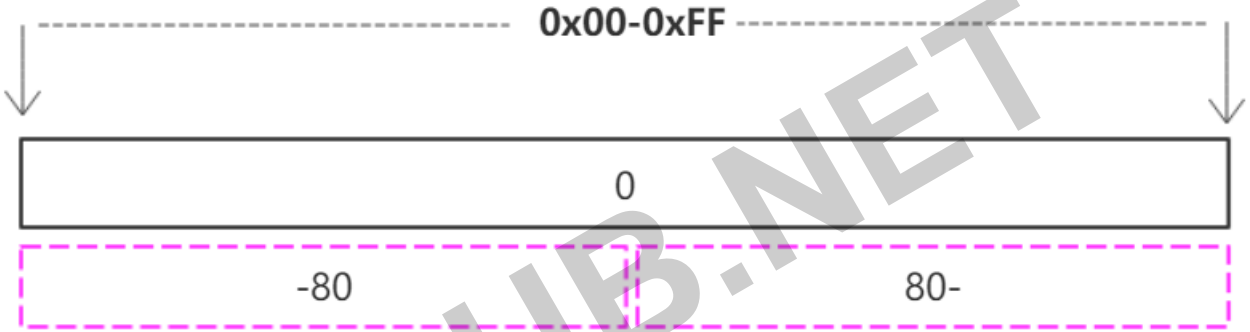
- 发展历程
- 整体架构
- 集群调度
- Resharding
- 故障迁移

Sharding/Resharding

通过添加（减少）从库或者拆分分片的方式以达到业务读写吞吐量均衡的目的；

场景	操作
提升读取的吞吐量	添加从库/拆分分片
提升写入的吞吐量	分片拆分
收回过渡分配的资源	合并分片/垂直扩缩/迁移缩容
增加地理多样性	增加新异地数据中心/添加异地从库
热数据	读取场景:添加从库/拆分热数据分片 写入场景:拆分热数据分片

Resharding



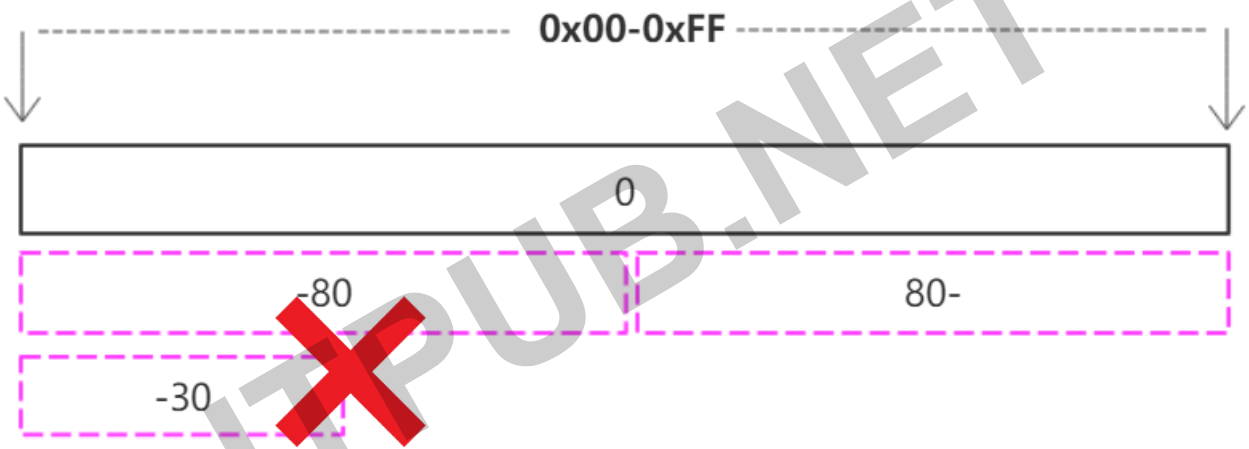
Shard
A shard is a division within a keyspace. A shard typically contains one MySQL master and many MySQL slaves.

Resharding

- Shard
- KeyRange

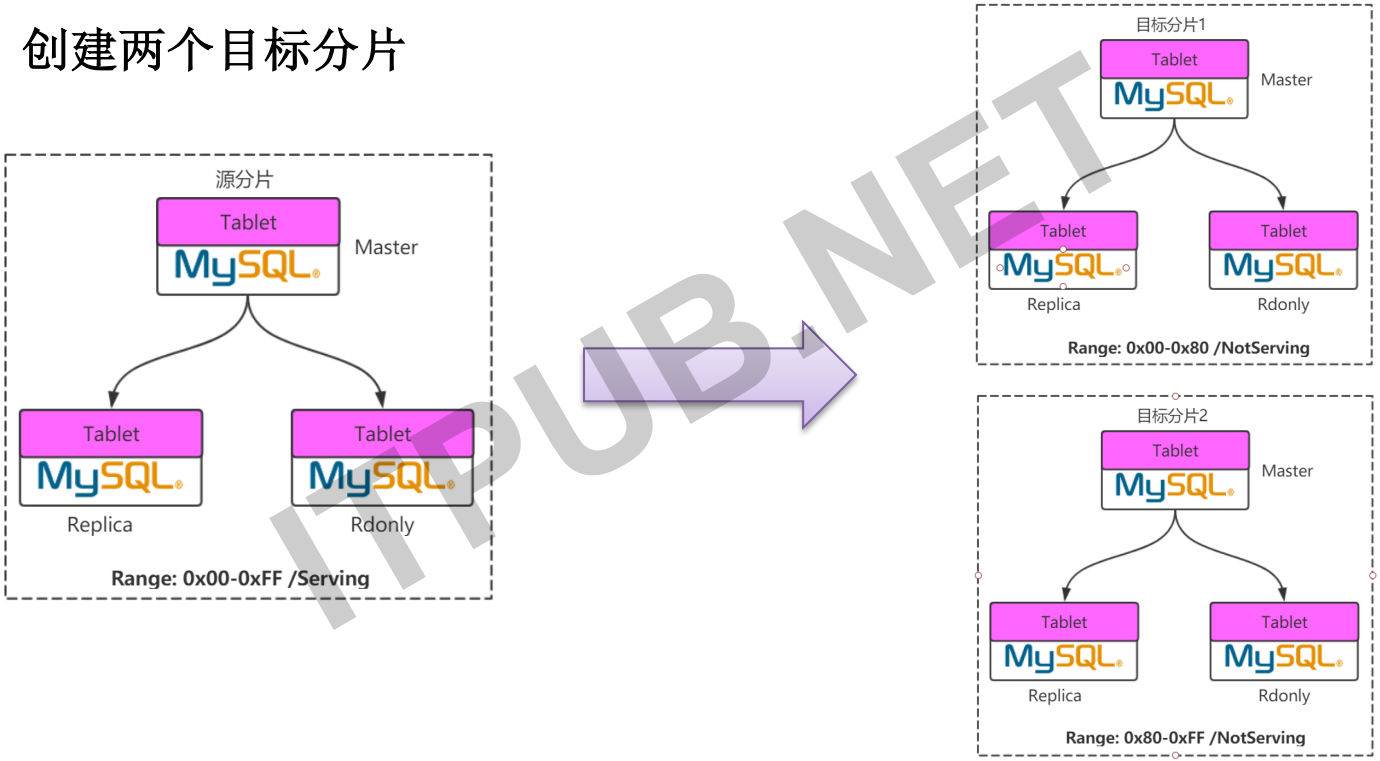
```
type Shard struct {
    MasterAlias     *TabletAlias
    KeyRange        *KeyRange
    IsMasterServing bool

    ...
}

// KeyRange describes a range of sharding keys, when range-based
// sharding is used.
type KeyRange struct {
    Start                []byte
    End                  []byte

    ...
}
```

Resharding



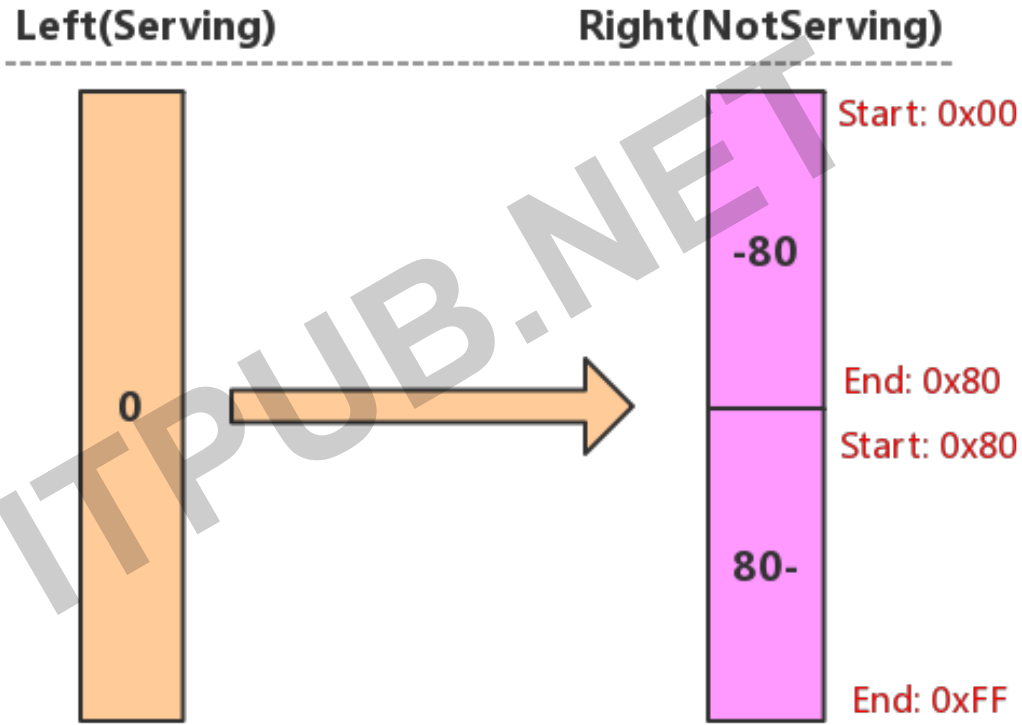
Resharding

Setp1：创建两个目标分片



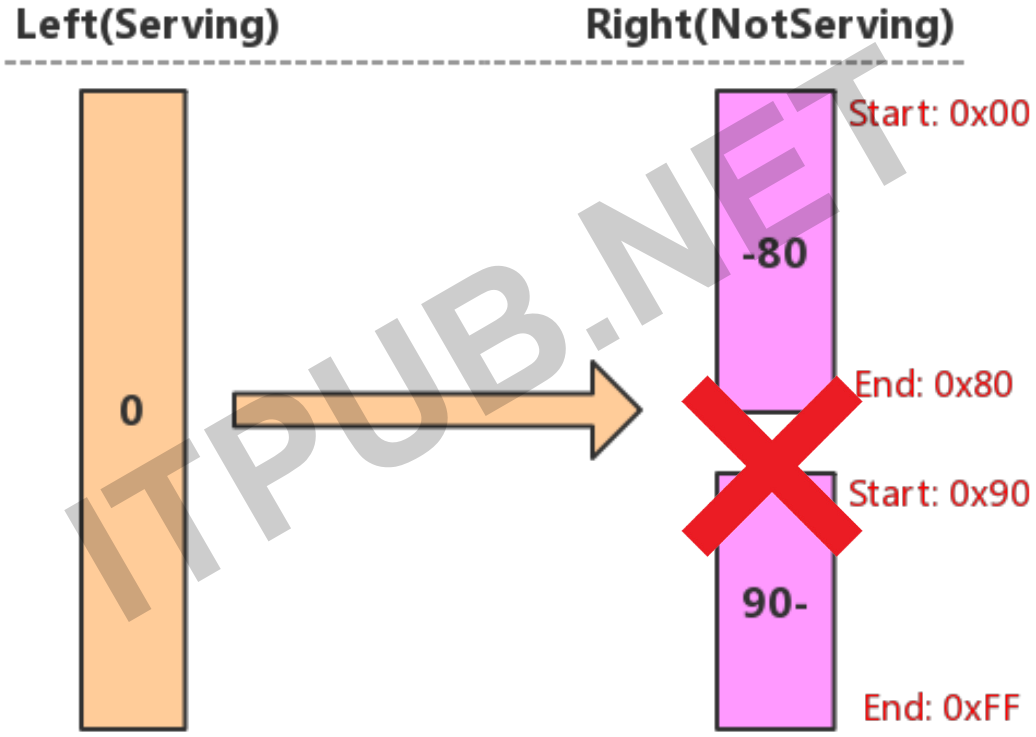
Resharding

初始化源分片、
目标分片信息



Resharding

目标分片无法覆盖所有Range



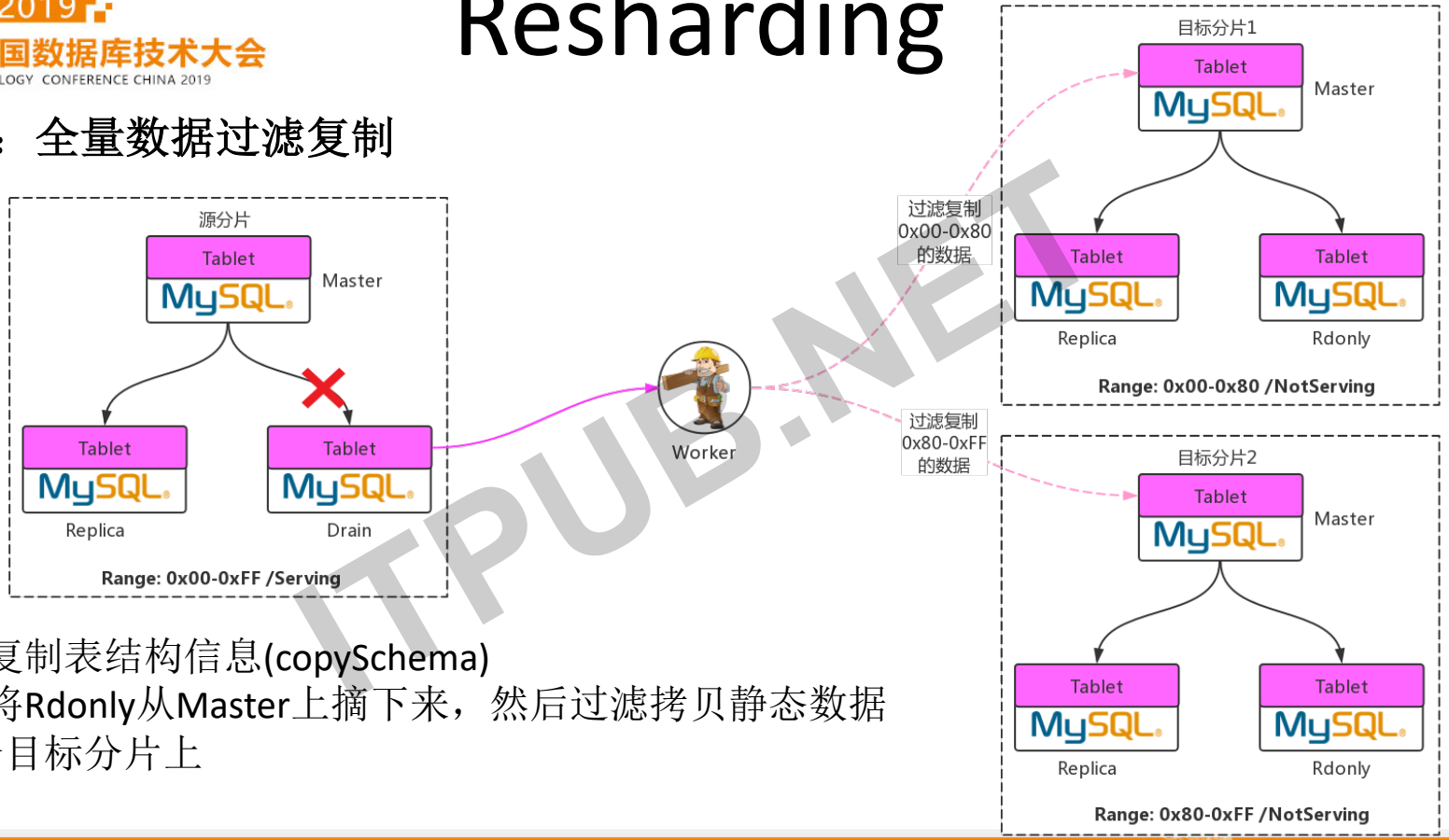
Resharding

合理性检查-SanityCheck：

- Destination Shard的master已经存在
- Destination Shard的Tablet只有master、replica、rdonly三种类型
- Destination Shard的Tablet都为未上线(NotServing)状态
- Source Shard为线上(Serving)状态

Resharding

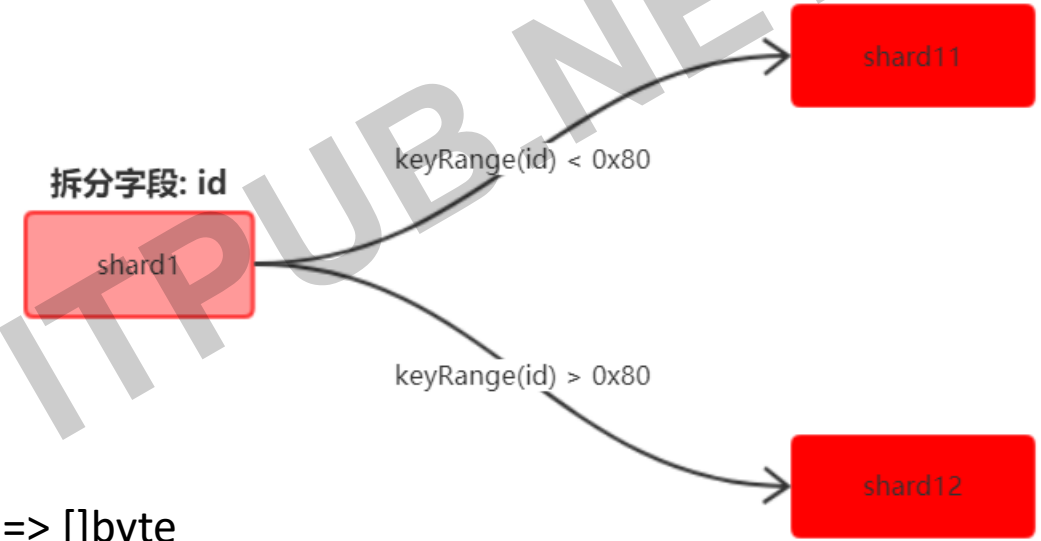
Setp2：全量数据过滤复制



（1）复制表结构信息(copySchema)
（2）将Rdonly从Master上摘下来，然后过滤拷贝静态数据到两个目标分片上

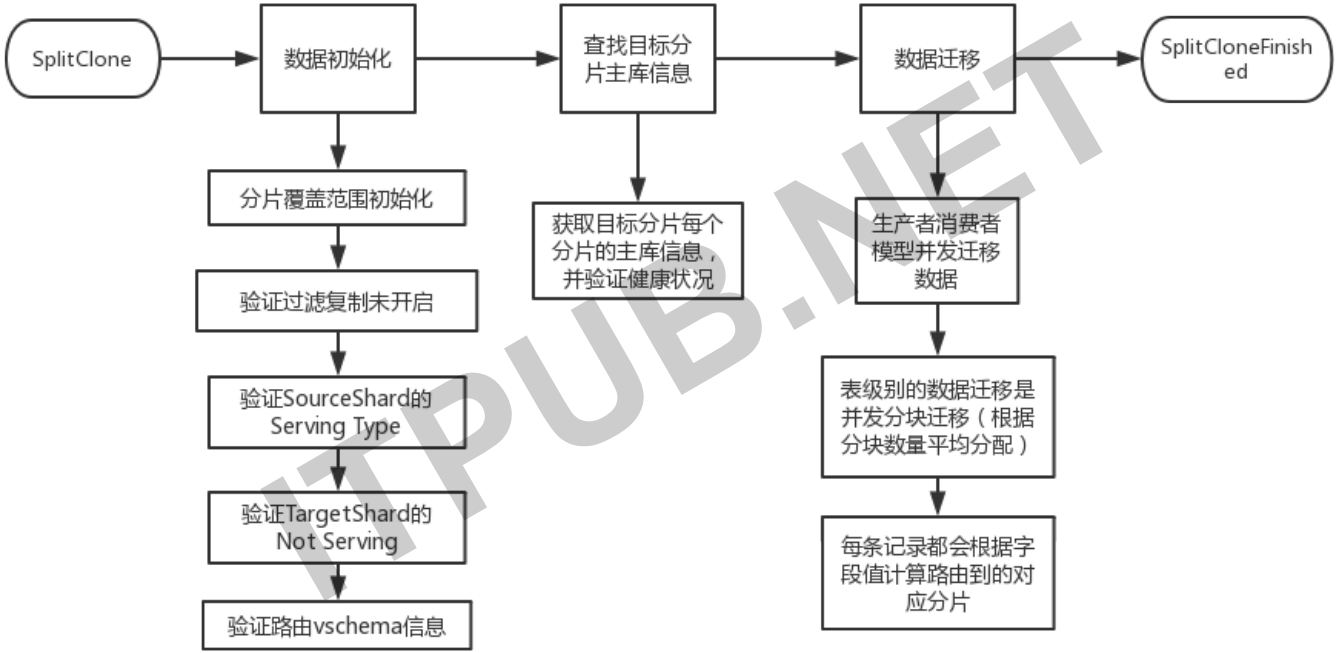
Resharding

过滤复制：
在Destination Master上开启一个BinlogPlayer，BinlogPlayer去Source Shard 的replicaTablet读取binlog，根据binlog和Sharding Key决定是否执行binlog



keyRange(id) = Hash(id) => []byte

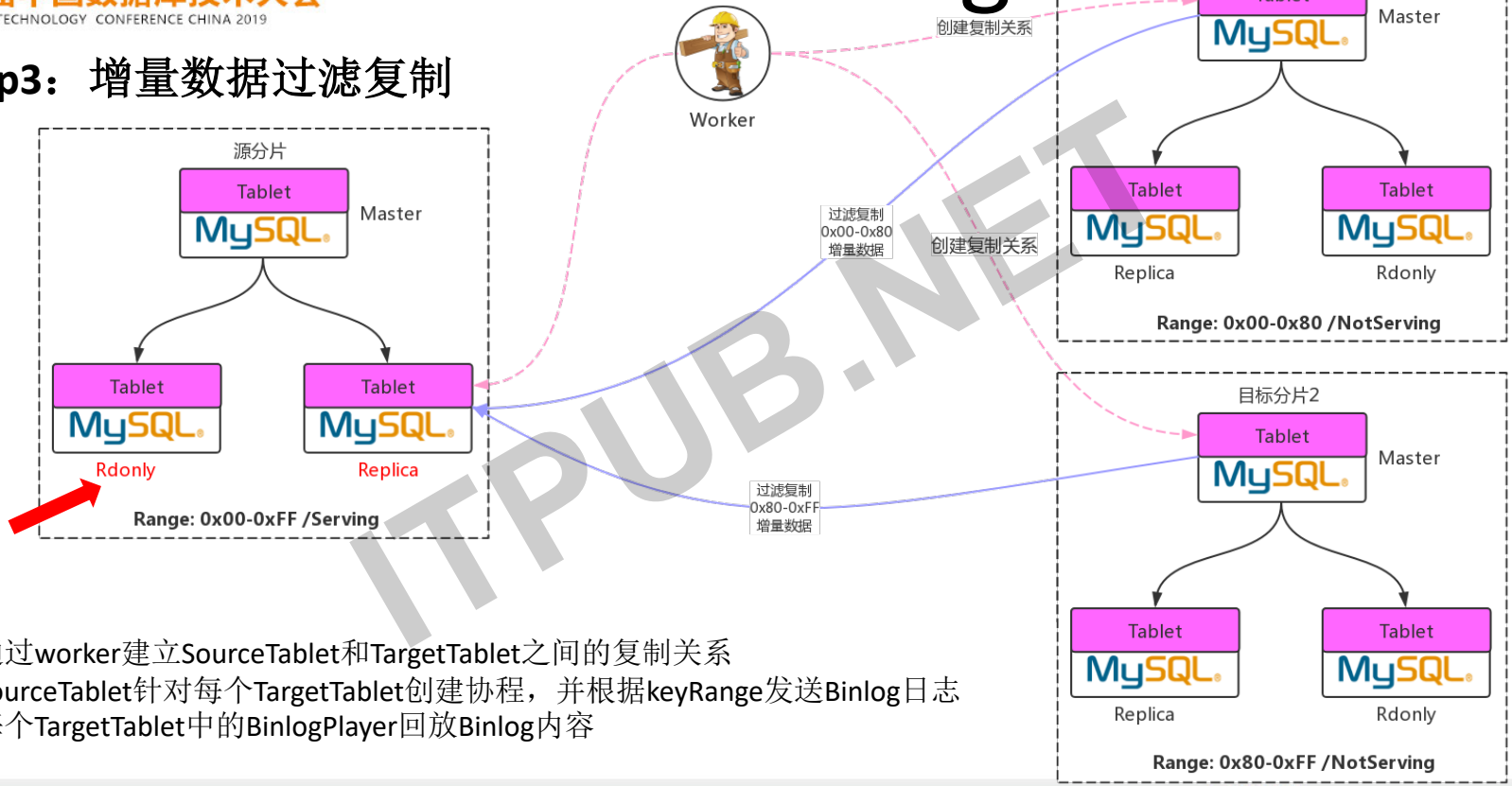
Resharding



SELECT id, x1, x2, x3, x4 FROM t1 WHRER id >= ‘start’ AND id < ‘end’ ORDER BY id;

Resharding

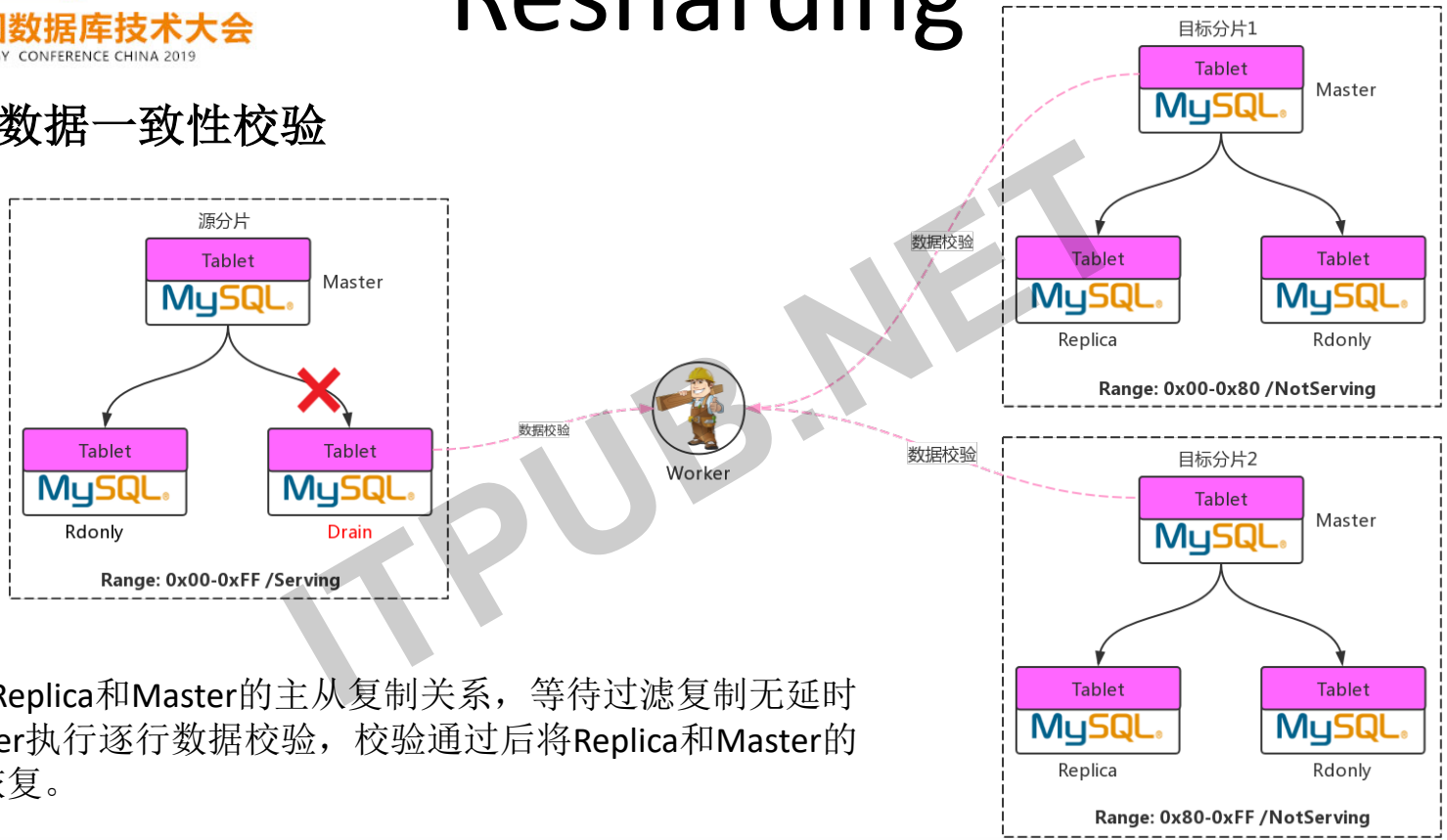
Setp3：增量数据过滤复制



（1）通过worker建立SourceTablet和TargetTablet之间的复制关系
（2）SourceTablet针对每个TargetTablet创建协程，并根据keyRange发送Binlog日志
（3）每个TargetTablet中的BinlogPlayer回放Binlog内容

Resharding

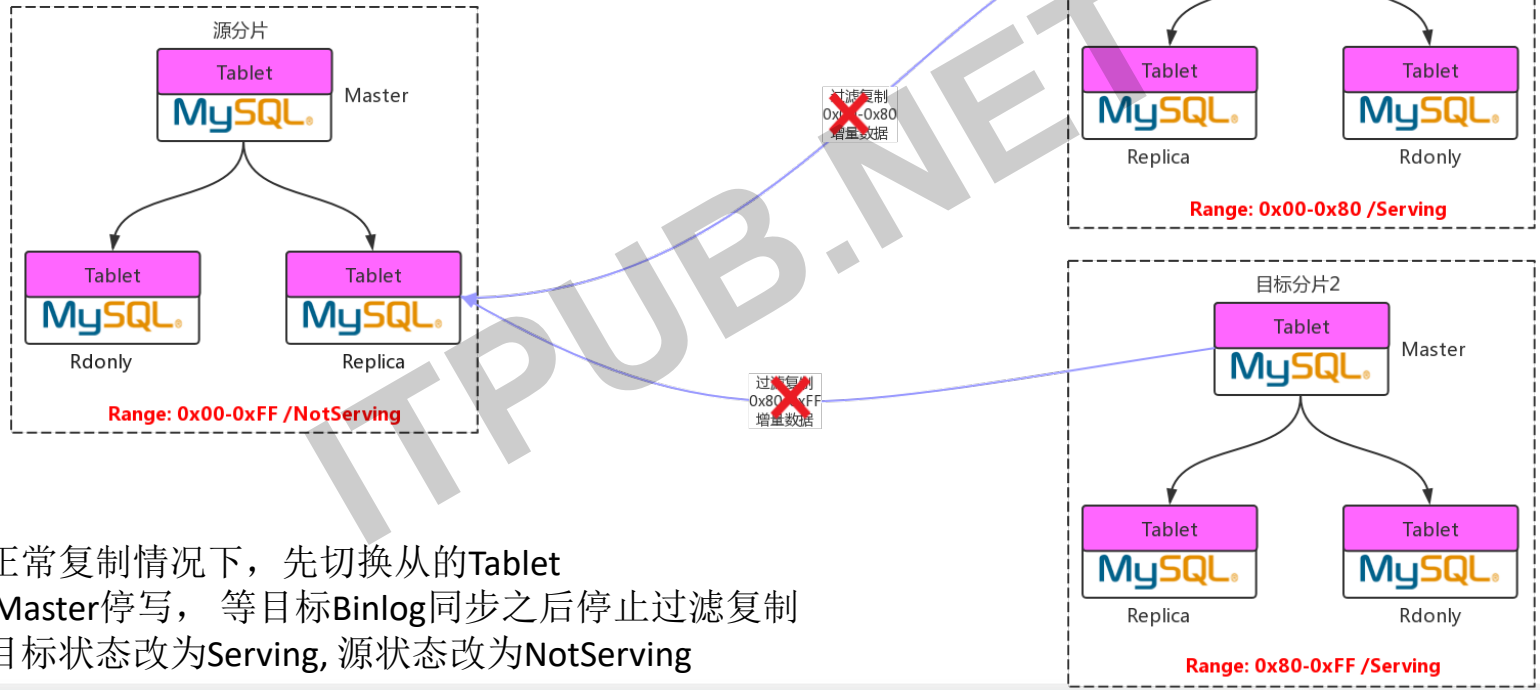
Step4：数据一致性校验



（1）移除Replica和Master的主从复制关系，等待过滤复制无延时
（2）Worker执行逐行数据校验，校验通过后将Replica和Master的复制关系恢复。

Resharding

Step5：切换ServerdType到目标分片



（1）在正常复制情况下，先切换从的Tablet
（2） 源Master停写， 等目标Binlog同步之后停止过滤复制
（3）将目标状态改为Serving, 源状态改为NotServing

Resharding

迁移过程中首先需要迁移rdonly和replica模式的Tablet，最后迁移master。迁移rdonly和replica只是通过修改路由信息将读操作转移到Destination Shard相应的rdonly和replica上，而不会停止过滤复制。

为什么不先迁移master呢？

如果首先迁移了master，replica和rdonly还是原来的Shard提供服务，这时Source Shard数据已经不更新了，相当于replica和rdonly在使用一份历史快照数据提供服务。

Resharding

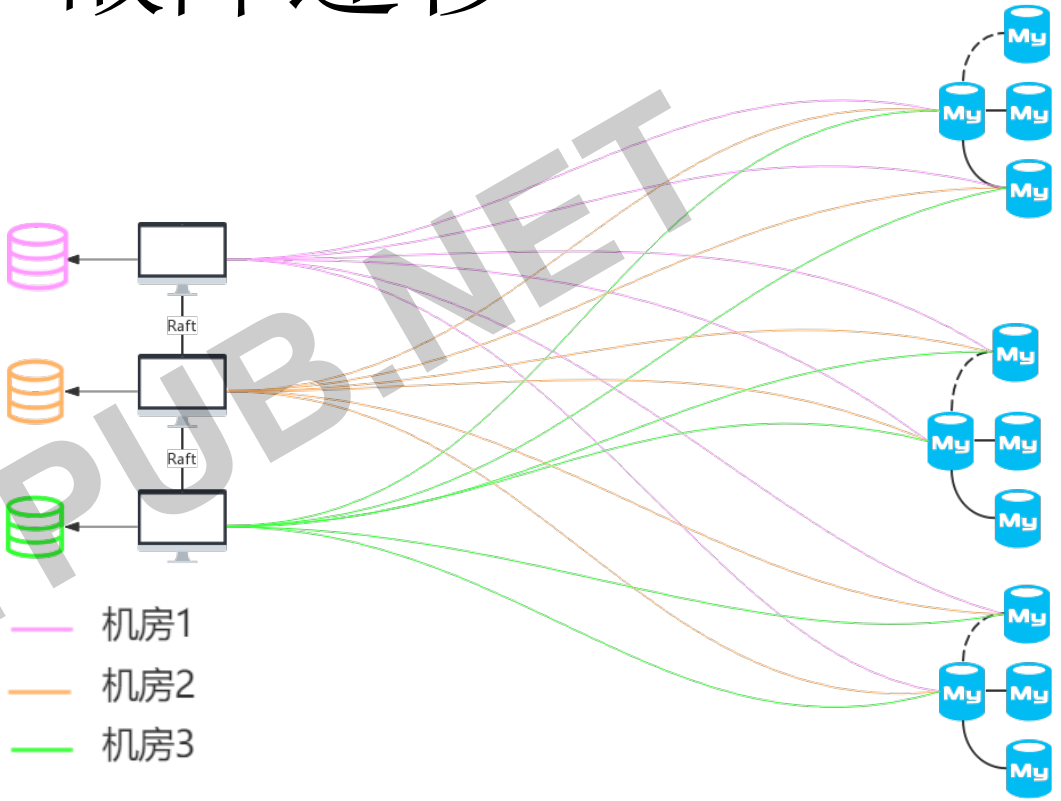
迁移master的主要步骤：

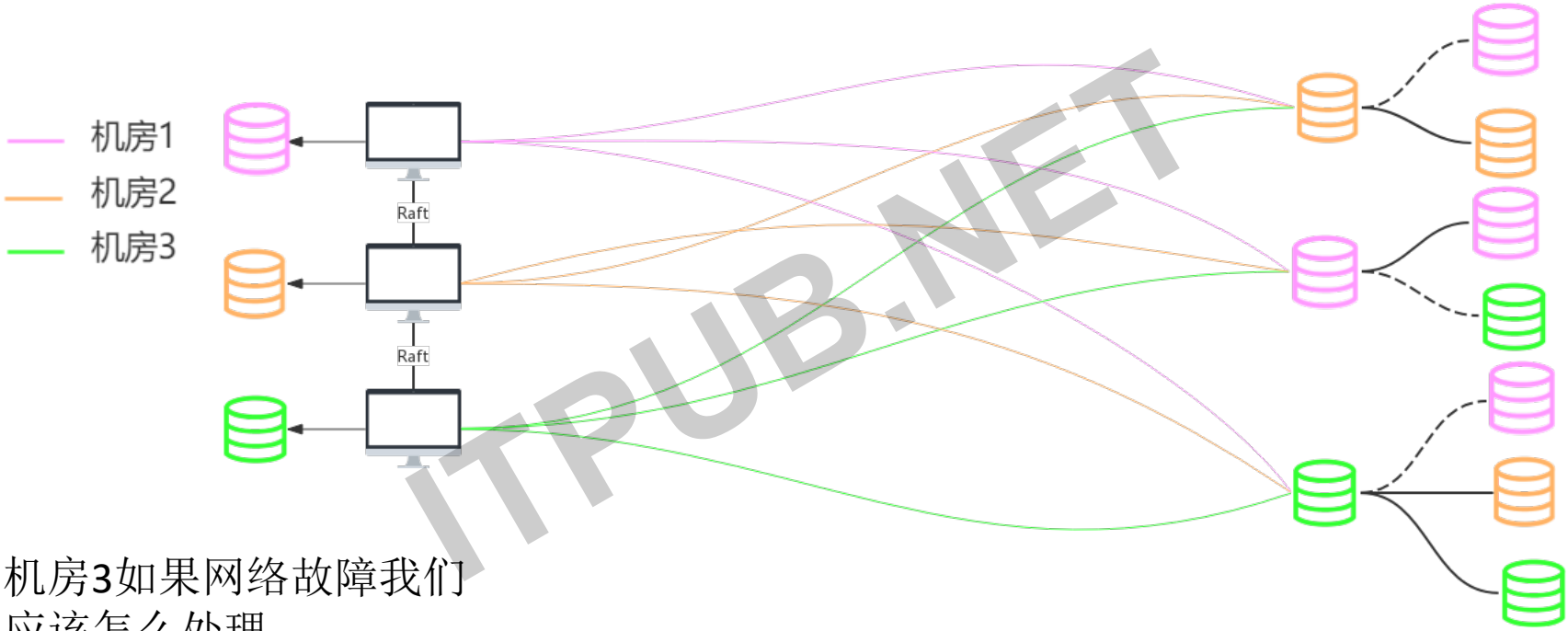
- 停止master的读写服务并得到master的binlog位置pos
- 等待所有的Destination Shard追binlog到 pos位置，保证无复制延迟
- 停止所有Destination Shard的过滤复制
- 修改DestinationShard的servedType为服务状态、更新路由信息并广播

- 发展历程
- 整体架构
- 集群调度
- Resharding
- 故障迁移

故障迁移

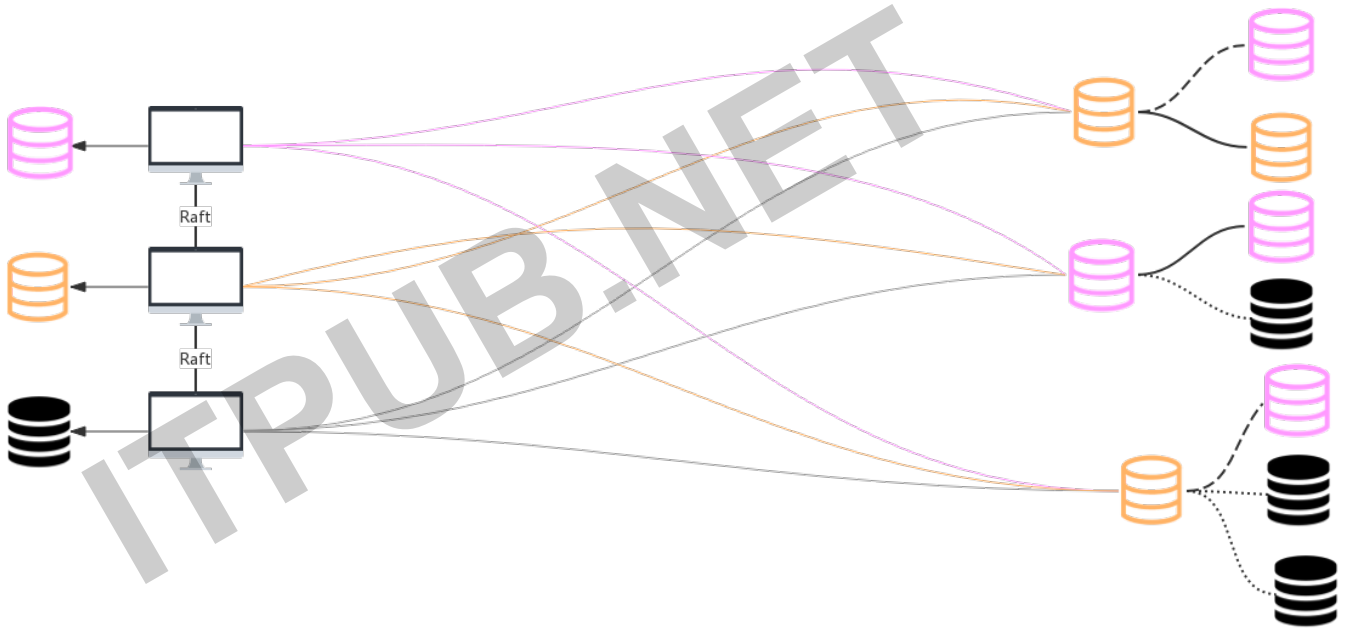
- 一主两（三）从
- 一个（两）半同步
- 一个异步





机房3如果网络故障我们应该怎么处理

- 两副本
- 网络隔离



最终形态