

中间件异地多活架构演变

基础平台部-中间件

2017年8月28日



消息中间件-JMQ

分布式服务框架-JSF

分布式内存存储-JIMDB

单机房容量上限

用户体验

- 1、并不是所有用户访问的内容都能通过CDN加速
- 2、单机房故障导致服务中断

异地灾备，保障数据安全

延迟

跨地域访问，延迟增大，怎么避免多次的往返交互，相同数据避免多次传输

数据冲突

多地需要对同一条记录进行修改，相互复制就会出现覆盖冲突

一地写入，或者把数据可以拆分为多条记录，各负责一部分，业务存在拆分、读合并等业务改造

强一致

有一部分基础数据不允许出现延迟，只能一地写入和读取

带宽与成本

专线带宽成本高，长距离网络质量相对会差一些

为了应对机房宕机，其他机房需要准备更多的机器承担故障机房的流量

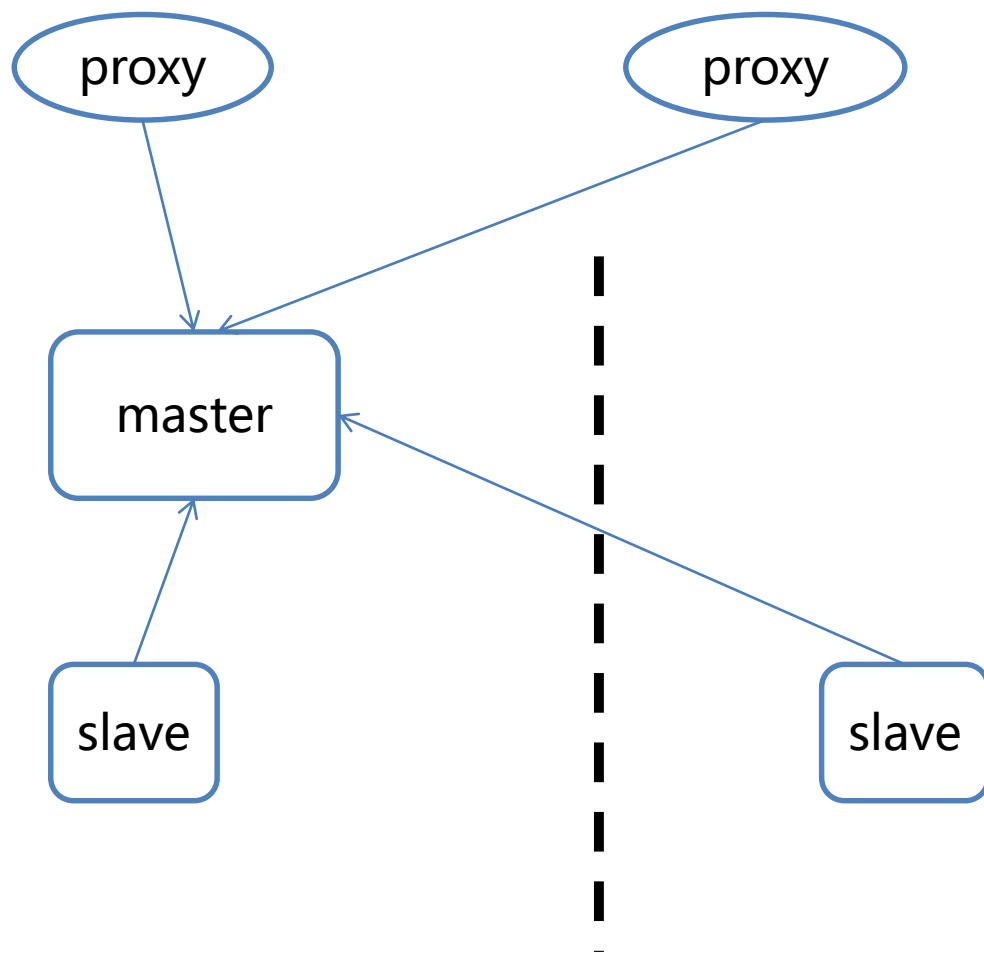
数据库中间件解决强一致数据跨地域复制与访问

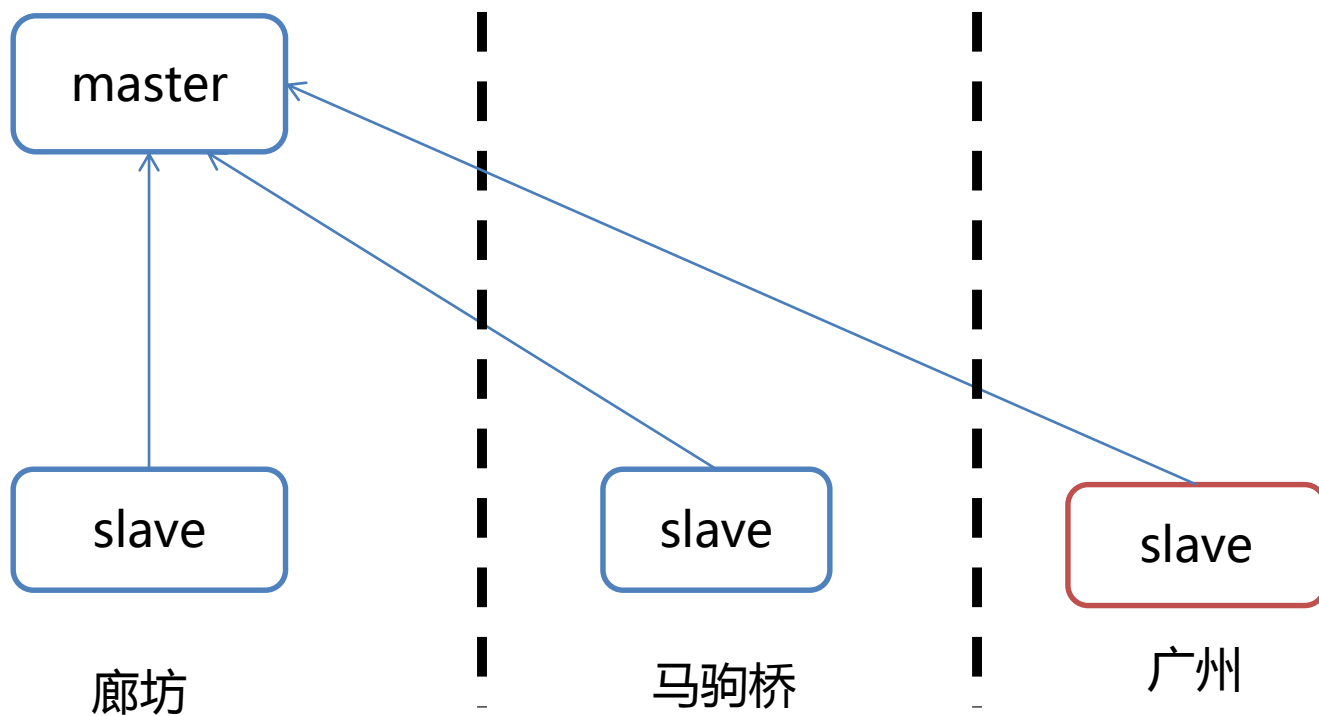
内存存储JIMDB进行跨地域复制

框架服务JSF注册信息全局同步，多地域注册与发现

消息中间件JMQ同机房生产与消费

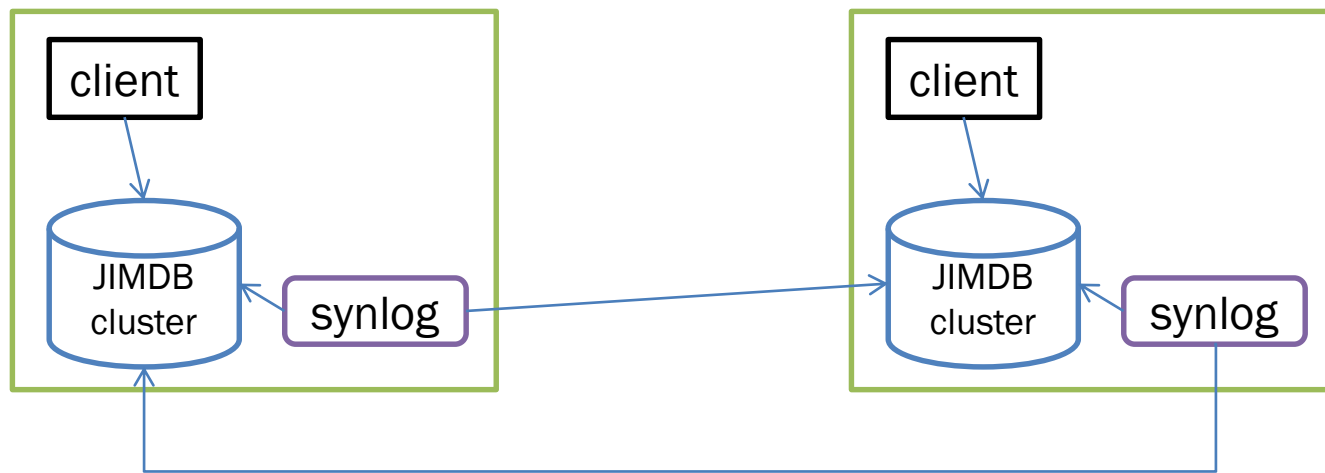
- 1、元数据需要增加机房标识
- 2、依赖数据库中间层，一地写入，多地复制和读取，当某一地故障，切换写入接入点就可以恢复管理功能
- 3、管理端不可用可能短暂影响新增服务能力，但是不能影响已有的服务能力





改造前架构图

- 1、网络故障，发生全量同步的次数会增多
- 2、跨机房写
- 3、控制管理跨地域访问影响性能
- 4、如果要跨地域添加多个副本，同一份数据多次传输



改造后架构图

循环复制：KEY增加机房标识

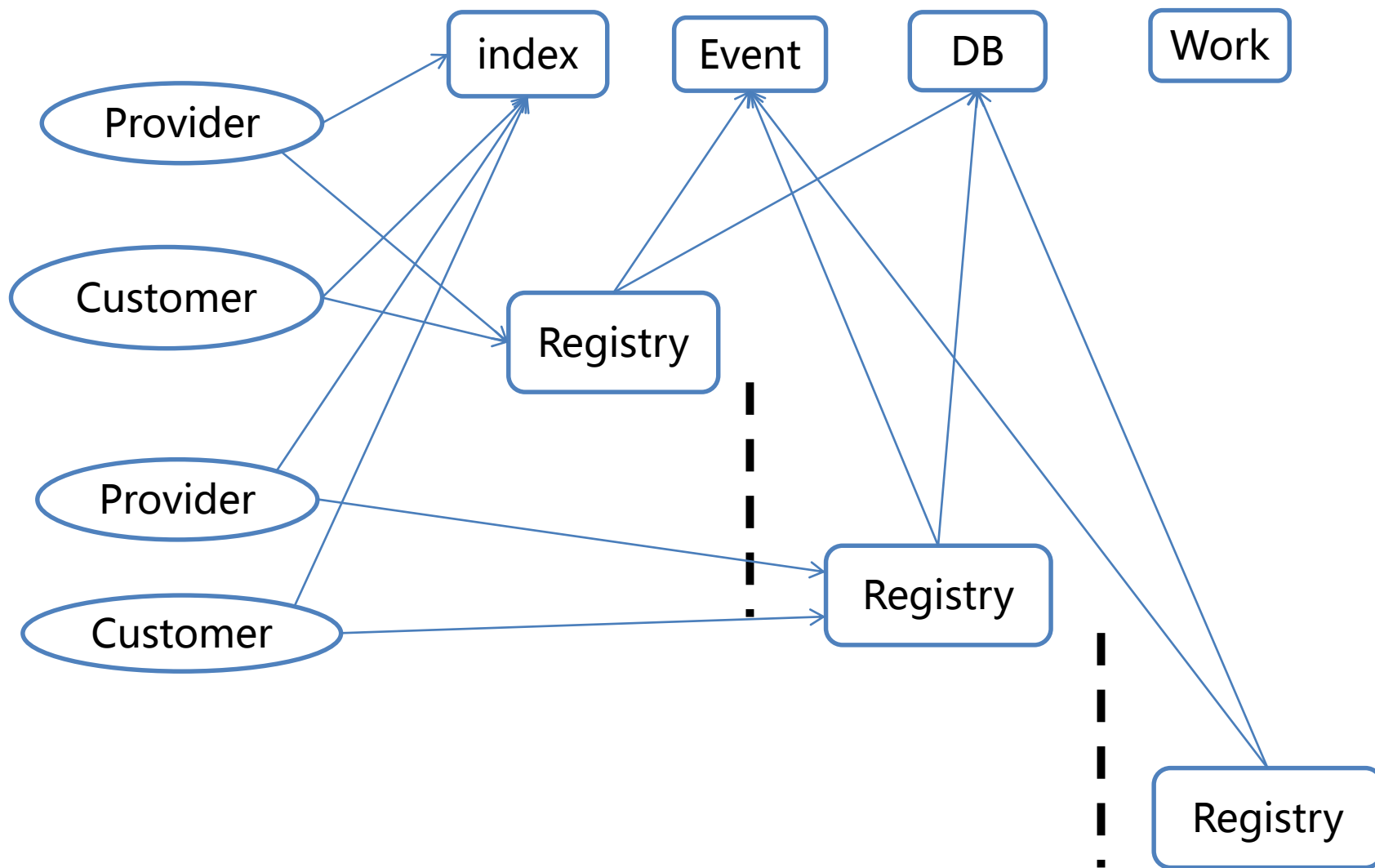
写入冲突：业务避免同一时间两地写入同一个KEY

优点：

- 多地可写

- 复制模块可以落地提供更强的复制积压堆积

- 每个地域程序独立部署，独立控制，故障恢复能力强



改造前服务发布流程：

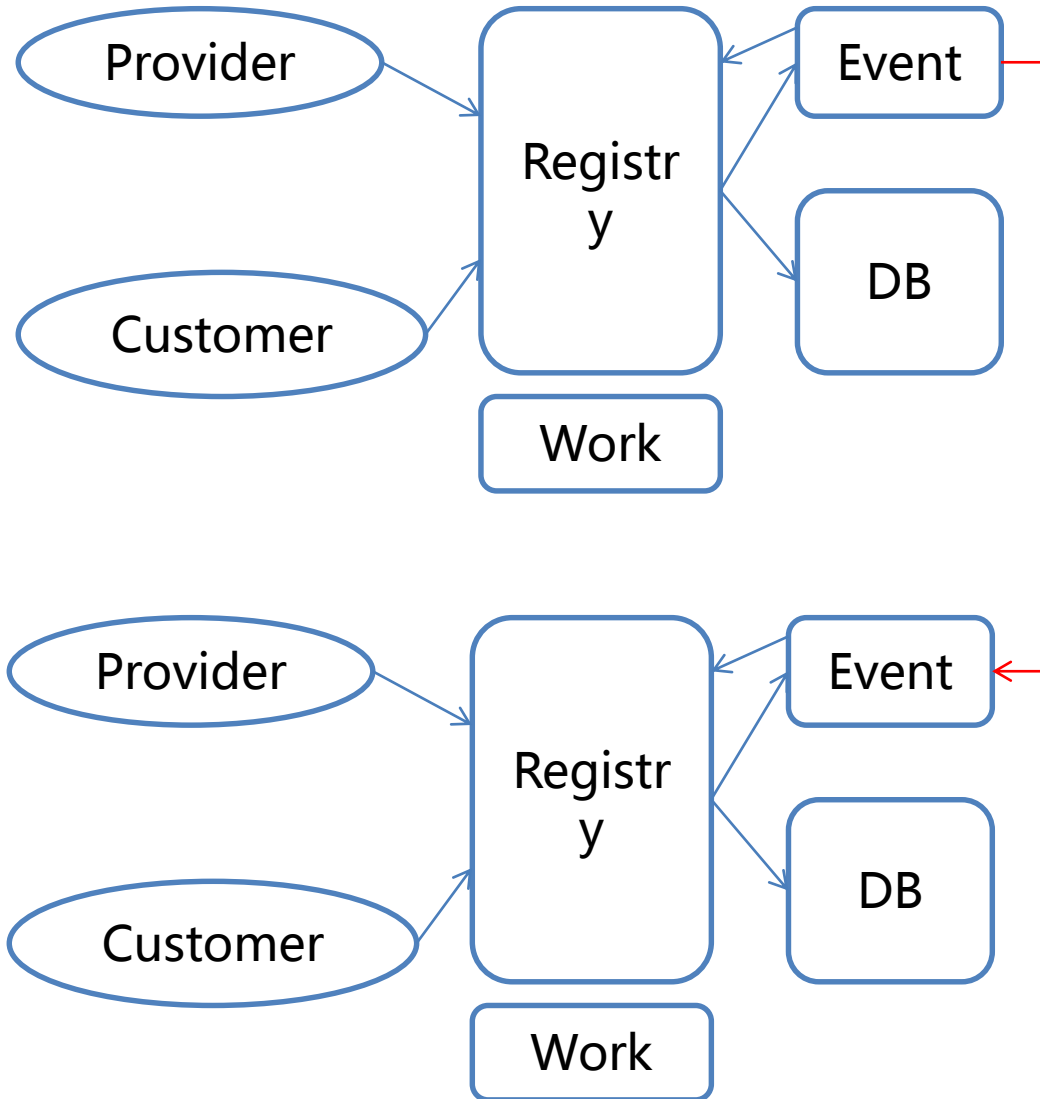
- 1、服务端发布时，服务端连接注册中心，注册中心把数据落地到一个统一的数据存储层；
- 2、当前注册中心发布上线事件，事件广播到各个注册中心的其他实例上
- 3、事件驱动集群的其他实例从数据层加载数据，服务端接口完成发布

改造前服务不存活检测：

- 1、检测任务选举出一个实例轮询接口的心跳信息，把没有心跳的接口进行下线，
- 2、下线信息记录到数据存储层，发布下线事件
- 3、服务接口从集群的其他实例中下线

跨地域后面临的问题：

- 1、统一的数据存储中心，跨地域多机房加载，延迟高，同一份数据多次专线调用
- 2、基础服务依赖单个存储中心的高可用，网络中断可能导致服务一段时间内不可用
- 3、事件通知中心可能存在跨地域调用



改造后发布流程：

- 1、调用端和服务端只访问本地的注册中心
- 2、探活任务只负责本中心的接口存活检测
- 3、注册中心数据先在本本地落地，再通过注册中心传递服务数据到其他中心的注册中心
- 4、服务端和调用端通过分组标识区分跨机房调用还是同机房调用

目标：

服务接口的发布、下线，被调用端发现等都通过本中心完成，不需要跨数据中心，当任何一个数据中心故障或者网络中断，都不会影响其他数据中心的发布和服务发现。

管理端依赖的用户信息，一些元数据管理等通过数据库中间层一地写入，多地读取，当发生故障时短暂的不可用并不会影响关键路径上的服务，通过数据库中间层切换写入地点就可以马上恢复。

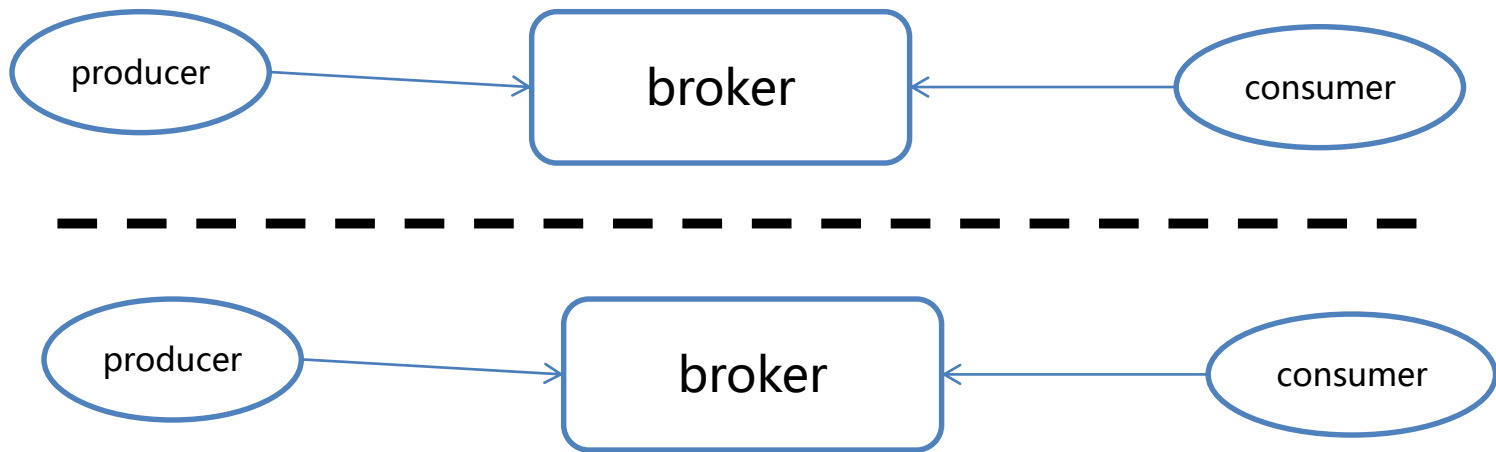
消息数据是流动的，不需要被多次访问，大部分场景下数据跨机房灾备的意义不大。

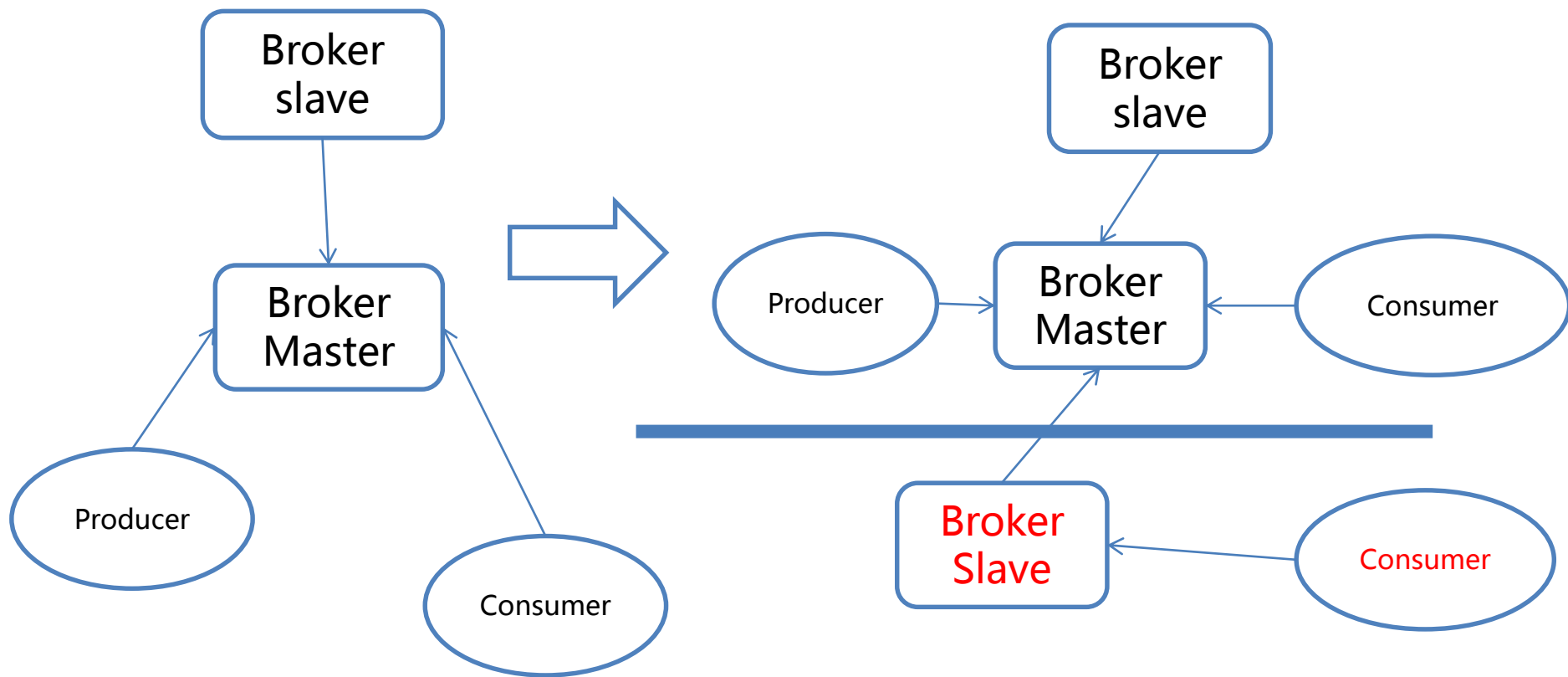
需要考虑机房级别的故障后，能够继续提供服务能够

TOPIC需要跨数据中心分配

重要消息可以跨机房备份

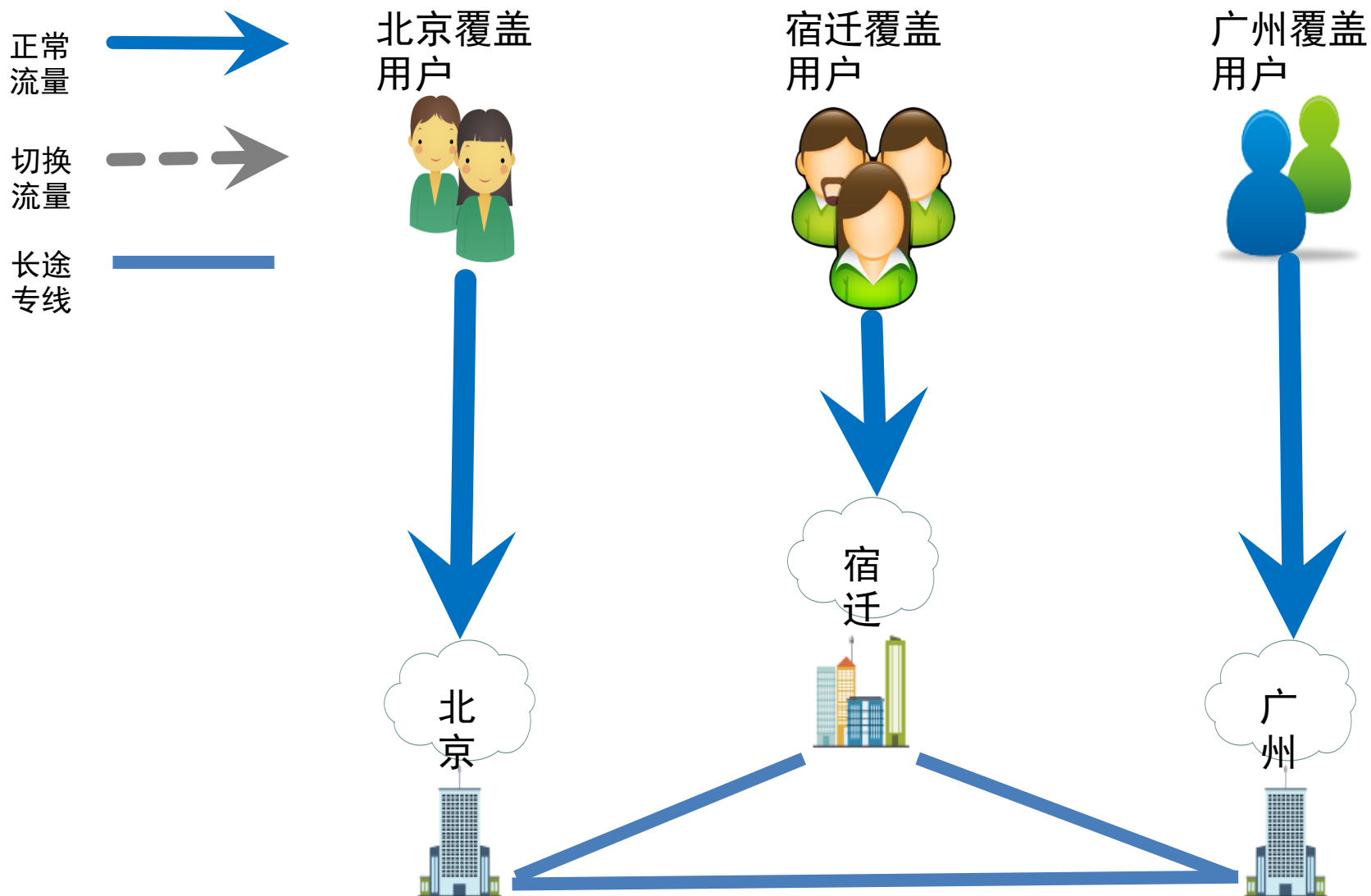
消费与发送业务客户端与BROKER通信不跨机房，复制可以跨机房





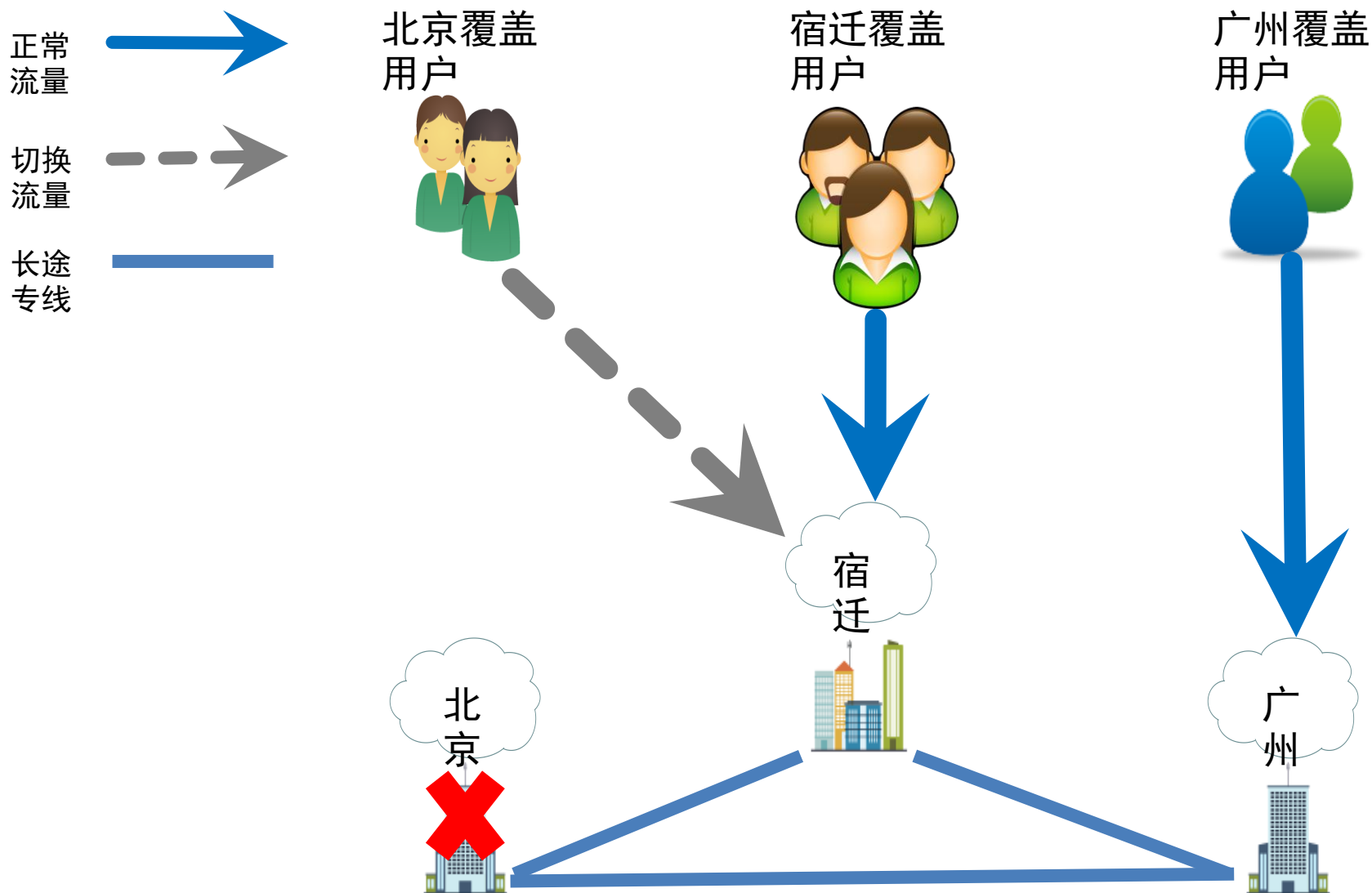
故障切换

JD.COM 京东



故障切换

JD.COM 京东



Q&A

THANKS

