# 概述

主要涉及故障检测、故障隔离、故障备份和故障恢复。

# 特征

首先高可用架构应该具备如下特征:

数据库对前端业务透明，业务不会因为数据库故障产生中断。

非主节点的数据应该和主节点的数据实时或者最终保持一致。

当业务因高可用机制发生数据库切换时，切换前后的数据库内容应当一致，不会因为数据缺失或者数据不一致而影响业务。

## 主从切换

高可用最主要的就是出现故障时，能够做到业务的无缝迁移，即主备切换。

当其中一台机器的服务宕机后，对于服务调用者来说，能够迅速的切换到其他可用服务，当服务升级为主服务，这种切换速度应当控制在秒级别（几秒钟）。

当宕机的服务恢复之后，自动变为从服务，主从服务角色切换。主从切换一定是要付出代价的，所以当主服务恢复之后，也就不再替换现有的主服务。

## 负载均衡

当服务的请求量比较高的时候，一台服务不能满足需求，这时候需要多台机器提供同样的服务，将所有请求分发到不同机器上。

高可用架构中应该具有丰富的负载均衡策略和易调节负载的方式。甚至可以自动化智能调节，例如由于机器性能的原因，响应时间可能不一样，这时候可以向性能差的机器少一点分发量，保证各个机器响应时间的均衡。

## 易横向拓展

当用户越来越多，已有服务不能承载更多的用户的时候，便需要对服务进行拓展，扩展的方式最好是不触动原有的服务，对于服务的调用者是透明的。

# 架构

## 选型

目前MySQL高可用方案有很多，几种典型的高可用架构选型有:

**主从或主主半同步复制：**通过依赖MySQL本身的复制，Master制作一个或多个热副本，在Master故障时，将服务切换到热副本从而达到高可用的效果。

**MHA+多节点集群：**基于MHA的集群方案，通常和其他第三方方案组合实现

**分布式协议：**基于分布式协议的高可用方案，常见的有Galera Cluster,PXC和MGR

**基于共享存储方案：**如SAN存储，这种方案可以实现网络中不同服务器的数据共享，共享存储能够为数据库服务器和存储解耦。

**基于磁盘复制方案：**如DRBD，DRDB是一个以linux内核模块方式实现的块级别同步复制技术。它通过网卡将主服务器的每个块复制到另外一个服务器块设备上，并在主设备提交块之前记录下来。类似共享存储解决方案。

## 架构

## 总结

MySQL高可用方案的建议，这些也是基于一些高可用的实践所做的总结。

1) 行业内多活的设计目标不是多写，需要先实现跨机房的高可用容灾和计划内的机房间数据切换

2) 引入consul的域名管理，解决VIP方案带来的一些潜在瓶颈（域名的业务属性，实现单机多实例，读写分离的域名配置）

3) 对于consul的整体定位不局限于集群环境，在单实例，集群，分布式中间件方向都可以采用，consul是作为一种通用的基础域名服务。

4) 同机房高可用方案的落地，需要和应用方对接程序端对域名的支持情况，在不同语言的客户端侧会有一些配置的差异。

5) 在已有高可用方案MHA基础上平滑过渡和改进，在后续新业务尝试引入MGR的方案

6) consul 业务API的开发，对数据库层面的业务可持续性访问（服务注销，服务发现）做一些补充和定制，保证consul服务的技术可控

7) 异机房高可用实现应用无缝切换，计划内切换，会有业务中断/延时，保证可控的基础上，应用端无须修改连接配置，需要测试DNS的域名转发等策略，计划外切换，需要做确认才可完成。

8) 对已有的分布式方案，可以采用MGR,中间件，consul的组合方案，实现读写分布式扩展。

# 方案

目前常用的高可用方案，自底层而上包括：

1. 数据库层面——主从复制，MySQL采用这种方式

金融领域很少采用，因为这种模式将主节点和从节点以及主从节点之间的网络环境紧紧地绑在一起，主节点的稳定性将不再由他们自己决定，而要同时看从节点和网络环境：一旦从节点或者网络环境抖动，主节点的性能就会受到直接影响。如果主节点和从节点之间是跨机房设置跨城市部署，发生这种概率的机会更大，影响也会更加显著。从某种程度上讲，和单节点模式相比，这种模式下主节点的稳定性不但没有增加，反而是降低了。

由于“主从复制”模式中缺少第三方仲裁者的角色，当主从节点之间的心跳信号异常时，从节点无法靠自己判断到底是主点故障了，还是主从之间网络故障了。此时，如果从节点认为是主节点故障而将自己自动切换为主节点，就极容易导致“双主”、“脑裂（brain split）”的局面，对用户来说这是绝对无法接受的结果。所以，数据库“主从复制”技术从来不会提供“从节点自动切换为主节点”的功能，一定要由人来确认主节点确实故障了，并手动发起从节点的切主动作，这就大大增加了系统恢复的时间（RTO）。

1. 底层硬件层面

在主机层面用HACMP技术以应对主机故障，或者在存储层面采取复制技术（比如FlashCopy）未提交数据冗余等，会使灾难切换方案变得很复杂，并且会有相对较长的故障恢复时间（RTO），所以通常不是数据库用户的首选。

1. 备份软件

支持异种数据库之间相互复制数据的产品，比如IBM CDC和Oracle Golden Gate（OGG）。这些产品的特点是比较灵活，可以支持异种数据库之间的数据复制，也可以指定只复制数据库中的部分对象（比如只复制指定几张数据表的数据）。但这些产品的缺点也很明显：首先相对于数据库主从复制来说，时延较大，通常会达到秒级以上，而且往往做不到数据库层面100%的完全复制。因此，这种方式通常作为不同数据库产品之间做数据“准”实时同步的手段，而不会作为数据库产品实现高可用及容灾的手段。

1. 集群层面

分布式多副本数据一致性技术，通常是基于Paxos协议或者Raft协议来实现。这种技术会将数据保存在多份副本上，每一次对数据的修改操作都会强同步到多数副本上，在保证了数据冗余的同时，不再像“主从复制”技术那样依赖某个数据节点的稳定性，从而消除了传统主从复制技术下从节点给主节点带来的危险。同时，在主节点故障的情况下，其余节点会自动选举出新的主节点以实现高可用（个别从节点故障则完全不影响服务），整个过程非常快速且完全无需人为干预。因此这种技术不仅能保证RPO=0，而且大大减小了RTO，相比传统主从复制技术，可以提供更强大的高可用能力。

此外，为了抵御机房级灾难和城市级灾难，可以将多份副本分散部署在多个机房里甚至多个城市里，以避免机房级灾难或者城市级灾难损失多数派副本。这样就具备了机房级和城市级容灾的能力，进一步加强了高可用的能力。

## 腾讯

## 阿里

<https://yq.aliyun.com/articles/637425?utm_content=m_1000015788>



# 应用

## 组建高可用

采用去中心化设计，整个集群无单点。

### 计算节点高可用

计算节点是无状态节点，一般要求为每个业务至少配置2个计算节点。

为降低计算节点异常对业务的影响，可以采用的措施：

1. 监控节点健康状况；
2. 在某些节点发生故障时，实现故障的接管；
3. 异常事务清理

### 数据节点高可用

为了提升数据服务的可靠性和可用性，安全组Group一般由多个数据节点组成。其中一个数据节点为主节点，提供读写服务，其他节点为备节点，提供读服务，在一个安全组中可以设置多个副本。

通过数据节点主备间通过数据库的复制技术来进行主备机之间的数据同步，主机至多个备机之间为星型复制模式，即主机直接向多个备机进行数据同步，具体方式目前支持半同步和异步两种。

为了权衡高可用和高性能，一般有以下几种策略：

**最大性能策略**

底层的数据节点提供最大的写性能，副本间采用异步赋值，即一旦日志数据写到主节点，事务即可提交。

**最大保护策略**

日志数据必须同时写出到主用数据节点和至少一个备用数据节点，事务才被提交。

**最大可用策略**

### 全局事务管理节点高可用

全局事务管理需要支持双机热备部署，主备双活，实现故障时秒级切换。

### 管理节点高可用

管理节点采用双机冷备的模式，可以实现秒级切换。

## 容灾

### 衡量指标

在容灾恢复方面，目前业界公认有三个目标值。

1. 恢复时间：企业能够忍受多长时间没有IT，处于停业状态；
2. 网络多长时间能够恢复；
3. 业务层面的恢复。

整个恢复过程中，最关键的衡量指标有两个：一个是RTO，一个是RPO。

所谓RTO，Recovery Time Object，它是指灾难发生后，从IT系统当前导致业务停顿开始，到IT系统恢复至可以支持各部门运作、恢复运营之时，此两点之间的时间段称为RTO。

所谓RPO，Recovery Point Object，是指从系统和应用数据而言，要实现能够恢复至可以支持各部门业务运作，系统及生产数据应恢复到怎么样的更新程度。这种更新程度可以是上一周的备份数据，也可以是上一次交易的实时数据。

### 同城双活

### 异地容灾