# 概述

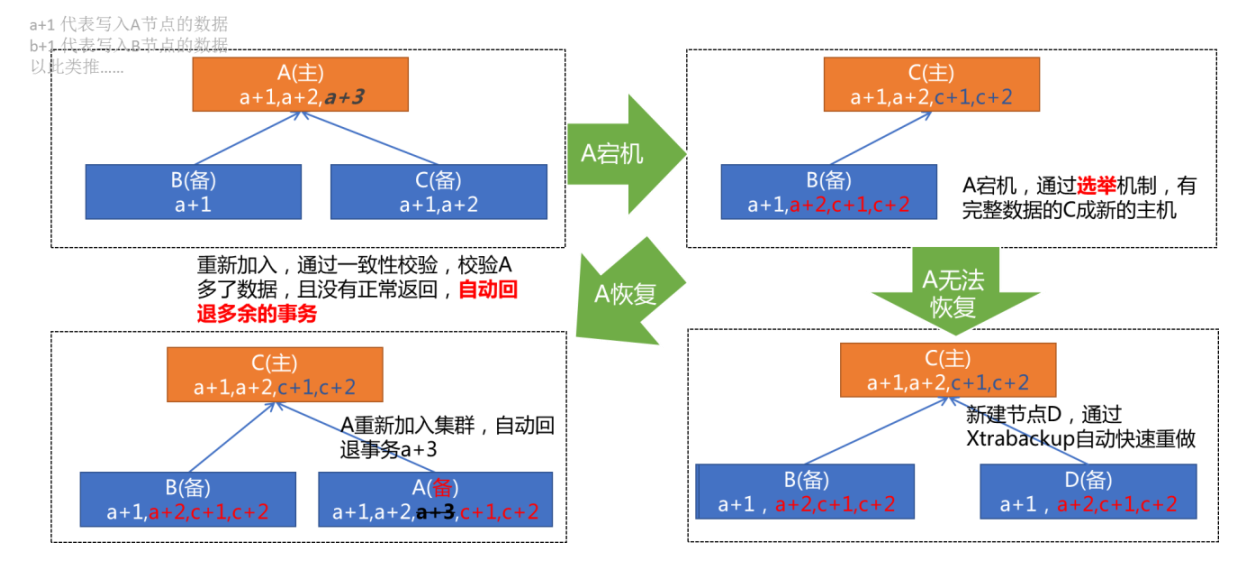
# 原理

# 分布式数据库实践

## TDSQL

在生产系统中，通常都需要用高可用方案来保证系统不间断运行；数据库作为系统数据存储和服务的核心能力，其可用要求高于计算服务资源。目前，TDSQL高可用方案通常是让多个数据库服务协同工作，当一台数据库故障，余下的立即顶替上去工作，这样就可以做到不中断服务或只中断很短时间，该方案简称主从高可用，也可以叫做主备高可用。在普通的主从高可用基础上， TDSQL支持：  
 ⚫ 支持故障自动转移，集群自动成员控制，故障节点自动从集群中移除；如果是实例级的主从切换，换后VIP（虚拟 IP）不变；基于强同步复制策略下，主从切换将保证主从数据完全一致，可满足金融级数据一致性要求。  
 ⚫ 支持故障自动恢复， 承载分片的物理节点故障，调度系统自动尝试恢复节点，如果原节点无法恢复，将在30分钟内自动申请新资源，并通过备份重建（ Rebuild）节点，并将节点自动加入集群，已确保实例长期来保持完整的高可用架构。  
 ⚫ 每个节点都包含完整的数据副本，可以根据DBA需求切换；  
 ⚫ 支持免切设置，即可以设置在某一特殊时期，不处理故障转移。  
 ⚫ 仅需x86设备，且无需共享存储设备即可支持；  
 ⚫ 支持跨可用区部署，实例的主机和从机可分处于不同机房（无论是否同城），数据之间通过专线网络进行实时的数据复制。本地为主机，远程为从机，首先访问本地的节点，若本地实例发生故障或访问不可达，则访问远程从机。 若配合腾讯VPC网络环境下，可支持同城双活架构，即业务系统可以直接在两个中心读写数据库。跨可用区部署特性为TDSQL提供了多可用区容灾的能力， 避免了单IDC部署的运营风险。

TDSQL的每一个分片都支持基于强同步的高可用方案，主数据库故障时将自动选举出最优备机立即顶替工作，切换过程对用户透明，且不改变访问IP。并且对数据库和底层物理设备提供7X24小时持续监控。发生故障时，TDSQL将自动重启数据库及相关进程；如果节点崩溃无法恢复，将通过备份文件自动重建节点（如下图）：



## OceanBase

## TiDB

## GoldenDB

### 故障判断

**怎么判断发生了故障？**

1. DBAgent主动上报DB的状态为失败，这个时候就认定故障，不需要额外进行判断
2. CM的DBAgent断链超过一定次数（默认是10s，3次），这种心跳检测只能说明CM和DBAgent断链了，还需要其他的一些判断：
3. 查看该主机下面的备机是SQL/IO线程是否异常，如果异常，这个时候不需要切换
4. 计算当前集群中与CM断链的总数，如果超过某个值（隐藏配置项），则认定是网络故障。

排除上述两种情况，就认定主机fail了，此时需要执行故障切换。

### 切换条件检查

进入切换的一些前期检查，判断当前情况下是否可以进行切换操作？

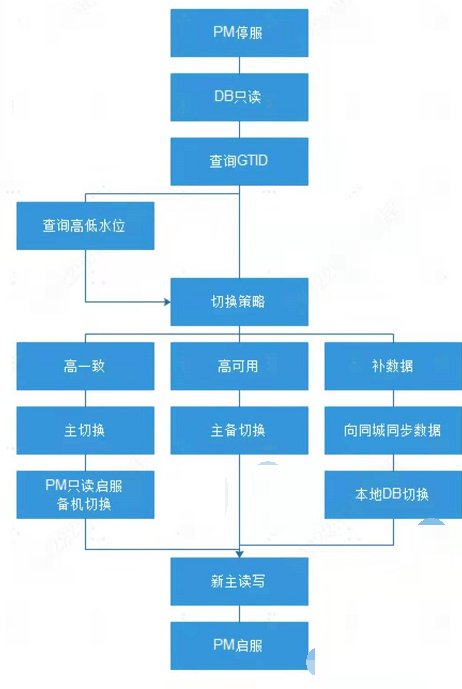
1. 当前CM的状态，是否已经启服，如果未启服，则不会执行切换
2. 根据切换策略（手动同城还是自动切换），初步判断是否存在可用的DB来供选择切换，如果没有可用的DB（RDB中存储相关备机信息），则同样不会执行切换

### 状态机

故障切换包含很多流程，每个流程会对应一个状态，目前包含的状态有：

1. PM停服
2. 设定DB只读
3. 查询DB的GTID
4. 查询水位信息
5. 切换策略判断（高可用、高一致性、补数据）
6. 连接到同城DB进行补数据
7. 设定PM只读启服
8. 主备切换
9. 新主设定读写
10. PM启服

状态转换图如下：



各个状态机的含义：

1. PM停服

只要涉及主的切换都会首先需要到PM申请停服，当前停服也不是PM的操作，是通过PM透传给各个proxy，让proxy执行停服，proxy停服后，再接收到外部的业务就直接拒绝了，停服是group级别的，某个group的停服是不影响其他group的业务操作的，在申请停服的时候会将在哪个group哪个主机停服这些详细信息发送给proxy，停服需要所有的proxy都停止成功，如果有一个停服失败，都表示停服任务失败。

1. DB只读

设定DB的read\_only配置项为ON，这个是设定该group下的所有机器，在成功情况下，所有DB（除了故障主机）本身的值就应该是只读的，这个地方重新设置一下是为了更加保险一些，设定备机只读的目的主要是防止：

1. 某个proxy在与PM断链后，仍然往旧主上发送数据（脑裂）
2. 用户或者其他程序直连到DB上
3. 查询GTID

这个流程很简单，就是向该city下面的所有备机发送查询GTID的请求，后期通过GTID来选择新主。

注：这个与TDSQL的同步机制不同，TDSQL要求所有备机全部同步成功才认为成功，切换的时候任何一个备机都是可以的（可以设置备机的优先级，OceanBase则是通过paxos智能选主），因为肯定是数据一致的。我们只要求有一个备机返回成功即可，在切换备机的时候并不一定任何一个备机都是与主机数据一致的，所以需要查询GTID最大的那个备机。这种方式可以明显提高写操作的效率，但是会稍微降低一些切换的效率。

1. 查询高低水位

在一致性检查时，需要从MDS查询高低水位信息，来判断是否有一致性副本存在，返回的有三个值，高水位（也称为正常水位），低于高水位，低于低水位，在低于低水位的情况就认定不存在一致性副本，切换失败，在高水位和低于高水位说明可能存在有一致性副本，不一定肯定存在，所以还需要结合多节点故障来判断，简单来说，在高水位的情况下，如果存在同城有DB故障，则不能切换，低于高水位就是如果存在本地或同城有DB故障，就不能切换。

1. 切换策略

根据配置的策略切换，以及是否可切同城选择相应的切换策略，目前有三个策略：

1. 高一致性
2. 高可用性
3. 补数据
4. 高一致性

从名称可以看出，优先保证数据的一致性才能进行切换，在主机故障了，宁可不发生切换，也不能在可能丢失数据的情况下执行切换，这种高一致性的保证主要是通过查询高低水位和多节点故障来判断是否有一致性副本存在，如果存在一致性副本，再根据切换策略选择GTID最大的DB切换为主，在备机切主成功后，此时会向PM发送只读启服的操作，同时让原来连接在旧主上面的备机切换到新主上面。

1. 高可用性

和高一致性区别就是，在高可用场景下，通过允许丢失数据来保证可以立即恢复到可用状态，处理逻辑也比较简单，就是选择一个GTID最大的DB为主就可以了。

1. 补数据

这个和高一致性，高可用性的选择策略不同的就是，在查询所有备机的GTID时发现GTID值最大的在同城，这个就需要看该集群是否允许自动切换到同城，如果允许切换到同城，再继续判断是高一致性还是高可用性，如果不允许切换到同城，那就要走补数据策略，具体流程就是除了GTID值最大的DB，其他所有的备机，都先连接到该DB上同步数据，当其他备机都同步完成后，在从本地选择一台DB作为新主来进行切换。

1. DB读写

当选择一台备机作为新主，切换成功后，设定该新主的read\_only为OFF，保证新主是可读写的权限。

1. PM启服

切换完成后，通知PM启服，同时告诉PM新主的IP和Port，让其在新主上启服。