数据库系统原理个人报告

——OceanBase 的高可用性及容灾方案

同济大学
TONGJI UNIVERSITY

院　　系　电子与信息工程学院

专　　业　计算机科学与技术

姓　　名　庄镇华

学　　号　1853790

题　　目　OceanBase 的高可用性及容灾方案

指导老师　李文根

目录

一、概述.....2
二、OceanBase 系统框架.....2
2.1 客户端.....2
2.2 RootServer.....2
2.3 UpdateServer.....2
2.4 ChunkServer.....2
2.5 MergeServer.....2
2.6 总结.....2
三、OceanBase 的高可用性及容灾方案.....3
3.1 为什么需要高可用性.....3
3.2 传统数据库产品的高可用性及容灾方案.....3
3.2.1 高端硬件产品.....3
3.2.2 数据库层面的“主从复制”技术.....3
3.2.3 存储层面的数据复制技术.....4
3.2.4 第三方数据复制产品.....4
3.2.5 传统数据库产品的问题.....4
3.3 OceanBase 的高可用性及容灾方案.....5
3.3.1 分布式多副本数据一致性技术.....5
3.3.2 OceanBase 常用的高可用性及容灾方案.....5
3.3.3 方案总结.....8
四、参考文献.....10

一、概述

在本次数据库前沿技术学习中，我们小组选择的题目是 OceanBase 数据库的分布式存储，我主要负责 OceanBase 的高可用性及容灾方案部分。

二、OceanBase 系统框架

2.1 客户端

用户使用 OceanBase 的方式和 MySQL 数据库完全相同，支持 JDBC、C 客户端访问，等等。基于 MySQL 数据库开发的应用程序、工具能够直接迁移到 OceanBase。

2.2 RootServer

管理集群中的所有服务器，子表(table)数据分布以及副本管理。RootServer 一般为一主一备，主备之间数据强同步。

2.3 UpdateServer

存储 OceanBase 系统的增量更新数据。UpdateServer 一般为一主一备，主备之间可以配置不同的同步模式。部署时，UpdateServer 进程和 RootServer 进程往往共用物理服务器。

2.4 ChunkServer

存储 OceanBase 系统的基线数据。基线数据一般存储两份或者三份，可以配置。

2.5 MergeServer

接受并解析用户的 SQL 请求，经过词法分析、语法分析、查询优化等一系列操作后转发给相应的 ChunkServer 或者 UpdateServer。如果请求的数据分布在多台 ChunkServer 上，MergeServer 还要对多台 ChunkServer 返回的结果进行合并。客户端和 MergeServer 之间采用原生的 MySQL 通信协议，MySQL 客户端可以直接访问 MergeServer。

2.6 总结

OceanBase 支持部署多个机房，每个机房部署一个包含 RootServer、MergeServer、ChunkServer 以及 UpdateServer 的完整 OceanBase 集群，每个集群由各自的 RootServer 负责数据划分、负载均衡、集群服务器管理等操作，集

群之间数据同步通过主集群的主 UpdateServer 往备集群同步增量更新操作日志实现。客户端配置了多个集群的 RootServer 地址列表，使用者可以设置每个集群的流量分配比例，客户端根据这个比例将读写操作发往不同的集群。

三、OceanBase 的高可用性及容灾方案

3.1 为什么需要高可用性

作为生产系统中最为关键的核心软件，数据库产品的高可用性一直是使用者极为关注的功能点。尤其是在金融这样一个特殊的领域里，无论是从监管的要求来看，还是从业务需求本身来看，都需要提供 24 x 7 持续不间断的服务，这就对金融行业中数据库产品的高可用性提出了很高的要求。不但需要应对个别硬件故障的情况，还必须能够应对机房整体故障和城市灾难等极端情况，保证数据库在各种意外情况下都能持续提供服务，即具备机房级容灾能力和城市级容灾能力。

3.2 传统数据库产品的高可用性及容灾方案

3.2.1 高端硬件产品

传统数据库产品最初都是单点架构，并不具备高可用设计，更多的是基于高端硬件产品满足“硬件可靠”的假设。

3.2.2 数据库层面的“主从复制”技术

随着时间的推移，传统数据库产品先后推出了采用“主从复制”架构的高可用方案，比如 Oracle 的 Data Guard 技术和 DB2 的 HADR 技术，其主要思路是：在原有的单数据库节点（主节点）之外再增加一个对等的数据库节点（从节点），通过数据库层面的复制技术（通常是日志复制）将主节点产生的数据实时复制到从节点；正常情况下从节点不提供对外服务，当主节点发生故障时，在从节点上执行“切主”动作将从节点变成主节点，继续提供服务。

在主从节点的部署方式上，除了本地单机房部署外，往往也支持同城灾备部署和异地灾备部署，因此也就具备了机房级容灾和城市级容灾的能力。很多新兴的数据库产品（如 MySQL）也是采用“主从复制”模式来实现高可用及容灾特性。

3.2.3 存储层面的数据复制技术

除了数据库层面的主从复制技术之外，还有一些在底层硬件上实现的高可用方案，比如在主机层面用 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）来提供数据冗余等。这些技术虽然也可以用来实现高可用和容灾能力，但和应用的整合度不高，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。

3.2.4 第三方数据复制产品

近些年还出现了一些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偏大，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。

3.2.5 传统数据库产品的问题

- 通常情况下无法做到 RPO = 0，即主节点发生故障或者灾难时有交易数据的损失。

(所谓 RPO，Recovery Point Objective，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样的更新程度)

- RTO 相对较大，通常以十分钟甚至小时为计算单位，会给业务带来比较大的损失。

(所谓 RTO，Recovery Time Objective，它是指灾难发生后，从 IT 系统当机导致业务停顿之时开始，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)

造成这一情况的根本原因，是“主从复制”模式下从节点不具备自动切主的能力。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，还是主从之间的网络故障了。此时，如果从节点认为是主节点故障而将自己自动切换成主节点，就极容易导致“双主”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，并手工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）。

3.3 OceanBase 的高可用性及容灾方案

3.3.1 分布式多副本数据一致性技术

与传统数据库的高可用技术不同，分布式多副本数据一致性技术通常是基于 Paxos 协议或者 Raft 协议来实现的。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派副本上，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风险。

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工干预。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”技术来说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中，以避免机房级灾难或者城市级灾难损毁多数派副本。这样就具备了机房级容灾和城市级容灾的能力，进一步加强了高可用的能力。

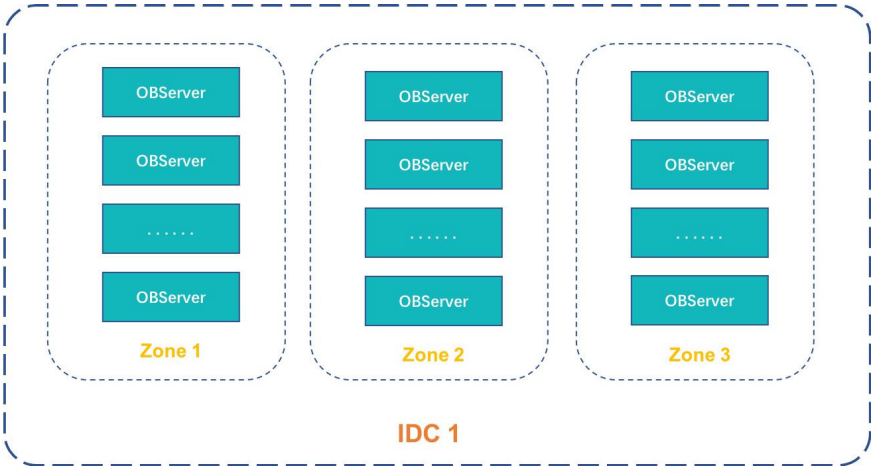
3.3.2 OceanBase 常用的高可用性及容灾方案

OceanBase 数据库从诞生之初，就利用 Paxos 协议在底层实现了多副本数据一致性，具有“RPO = 0、低 RTO（通常在 30s 以下）、故障时自动切换”的优势。而经过多年实际应用场景的历练后，尤其是像支付宝、淘宝、网商银行这种高并发、高访问量、24 x 7 持续交易场景的磨练，OceanBase 数据库已经摸索出一套完整的、经过实践检验的高可用及容灾方案。

3.3.2.1 单机房 3 副本

这是最简单的一种方案，在一个机房内找足够多的机器，把它们划分成 3 个 Zone，每个 Zone 里一份副本。(一个 Zone 是一个或者一些 OBServer 组成的

逻辑集合，一个 Zone 里的所有 OBServer 都在一个机房内。从分布式多副本数据一致性协议的角度来看，可以认为一个 Zone 就是 OceanBase 数据库集群的一个“副本”，如果是三副本架构那就会有三个 Zone)



这种方案具备一定程度的高可用能力，可抵御个别硬件故障，比如在个别服务器宕机、个别硬盘损坏等情况下，数据库集群还能持续提供服务。此外，这种方案具有部署方便，成本低的特点，只要有一个机房，机房内有足够多的联网机器，就可以部署了。

但这种方案也有一个非常明显的劣势：不具备容灾能力。如果发生机房级灾难或者城市级灾难，首先会导致交易停止，而极端情况下（比如机房所有机器损毁）甚至会导致数据丢失。

综合来看，这种方案虽然部署方便，也具备高可用特性，但其容灾的能力却是最低的，对于具有容灾要求的系统来说显然是不适合的。如果用户的硬件条件有限，只能提供一个机房，并且用户对系统的容灾能力没有要求，那么这种方案是一个非常合适的选择。

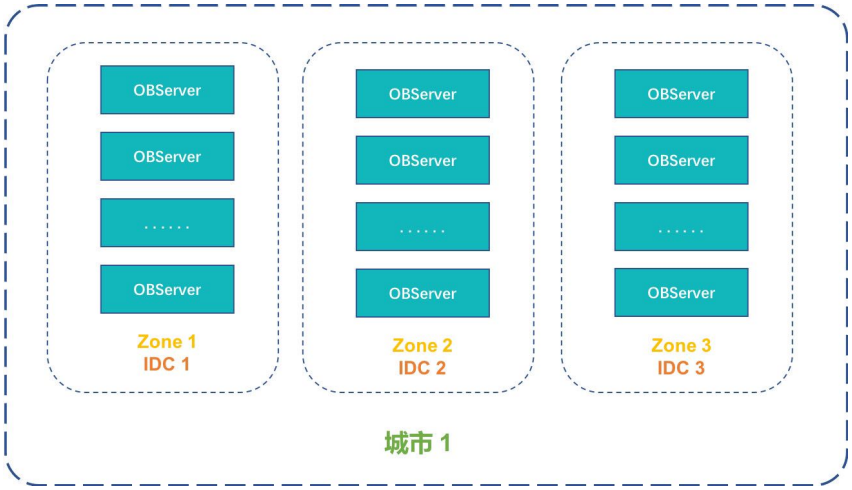
3.3.2.2 同城 3 机房 3 副本

同样是个城市内部署 3 副本，这种方案相对于“单机房 3 副本”来说就更进了一步：在同一城市内找 3 个不同的机房，每个机房内部署 1 个 Zone（1 份副本），形成一个跨机房部署的数据库集群。

由于分布式多副本数据一致性协议要将每一个事务的数据强同步到多数派副本中，这种部署模式下必然导致频繁的跨机房同步操作。为了保证数据库的写性能，对机房之间的网络质量有比较高的要求，通常要求任何两个机房之间的网络时延不超过 2 毫秒。

相对于“单机房 3 副本”来说，这种方案的优势是很明显的：除了可以抵

御个别硬件故障，还可以抵御机房级灾难：任何一个机房遇到灾难，都不会导致交易停止或者数据丢失。



不过，3 机房对用户的基础设施来说提出了一定的挑战，也增加了用户的部署成本。如果考虑到上面说的任意 2 个机房之间都要做到“网络低延时”，那成本会进一步增加。因此，在考虑这种部署方案时，要确保用户能提供符合要求的基础设施。最后，这种方案仍然不具备城市级容灾的能力，如果发生城市级灾难，还是会导致交易停止，甚至有数据丢失的风险。

综合来看，如果能够提供同城 3 机房的硬件设施，并且没有城市级容灾的要求，那么推荐使用这种方案，可以在城市内提供非常好的高可用及容灾能力。事实上，OceanBase 数据库的一些外部企业级客户就是采用了这种部署方式。

3.3.2.3 3 地 3 机房 5 副本

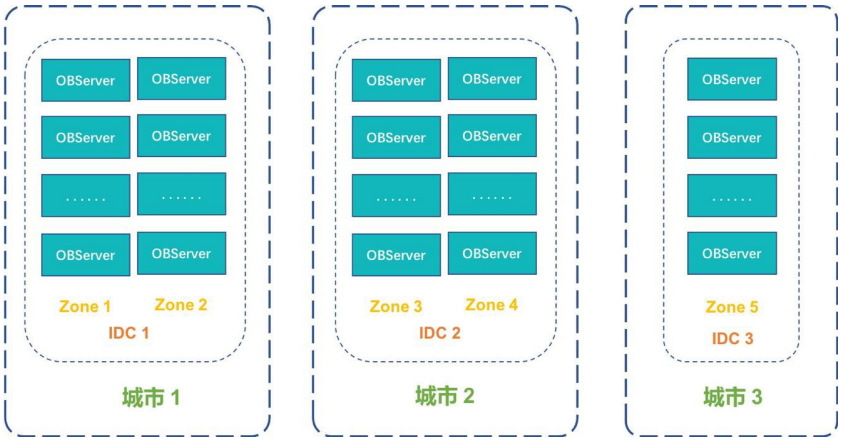
首先需要有 3 个城市，每个城市各有 1 个机房，利用这 3 个机房组成一个跨机房部署的 OB 集群。其次，这种方案对不同机房间的网络质量有一定要求，通常来说需要满足下面的条件：

1）有 2 个城市的距离相对较近（比如杭州和上海），它们之间的网络时延低于 10 毫秒（ms）。这 2 个城市的机房中各部署 2 个 Zone（副本）。

2）第 3 个城市和前两个城市的距离相对较远（比如深圳），它和前 2 个城市之间的网络时延应保证在 30 毫秒（ms）内。这个城市的机房内部署 1 个 Zone（副本）。

在这种部署模式中，距离较近的 2 个城市有 2 个 IDC，合计 4 份副本，构成了 Paxos 协议要求的多数派，因此日常写交易中的强同步操作会集中在这 2 个城市中，不会涉及到距离较远的第 3 个城市，有效避免了远距离 RPC 对交易

性能带来的影响。如果 2 个距离较近的城市中有任何一个 Zone 发生故障，剩下的 3 个 Zone 依旧构成多数派，还是不会涉及到距离较远的第 3 个城市，性能不会受到影响。如果这 2 个城市中有 1 个城市发生了机房级灾难或者城市级灾难，剩下的 1 个城市和距离较远的第 3 个城市合在一起还有 3 个 Zone，依旧构成多数派，还是可以继续提供服务，但此时远距离 RPC 操作将不可避免，性能也会因此而受到影响。因此，这种方案可以全方位抵御个别硬件故障、机房级灾难和城市级灾难，提供最高级别的高可用性，使数据安全性得到了最大程度的保障。



不过，这种方案在实际部署中面临着一些问题和挑战。首先，需要在 3 个城市内各有一个机房，3 个城市之间要满足“2 近 1 远”，而且相互之间的网络时延也要满足一定条件，这对用户的基础设施条件提出了非常大的挑战，即使对高端企业级用户来说，也很难满足这个条件，最多只具备 2 地 3 机房的条件。另外，5 副本相对于 3 副本来说增加了 2 个副本，进一步提高了硬件成本，也加大了这个方案的难度。

如果实际应用对 SLA 提出了最高要求，需要抵御机房级灾难和城市级灾难，并且希望做到“RPO = 0、低 RTO、故障时自动切换”，那么此方案将是不二之选。事实上，网商银行的数据库集群部署就是采用这种架构，支付宝中的部分核心数据也是采用了这种架构，它们为业务提供了最佳的数据安全性保障。

3.3.3 方案总结

方案名称	方案特点	基础设施要求	适用场景
单机房 3 副本	RPO = 0,RTO 低，故障自动切换。可抵御个别硬件故障，无法抵制机房灾难或者城市灾难。	单机房	对机房级容灾能力和城市级容灾能力没有要求
同城 3 机房 3 副本	RPO = 0,RTO 低,故障自动切换。可抵御个别硬件故障和机房级灾难，无法抵御城市级灾难。	同城 3 机房。同城机房间网络时延低。	需要机房级容灾能力，但对城市级容灾能力没有要求。
3 地 3 机房 5 副本	RPO = 0,RTO 低,故障自动切换。可抵御个别硬件故障、机房级灾难和城市级灾难。	3 地 3 机房。其中两个城市距离较近，网络延时低。	需要机房级容灾能力和城市级容灾能力。

四、参考文献

[1] 杨传辉. 大规模分布式储存系统——原理解析与构架实战. 机械工业出版社.

[2] (美)braham Silberschatz, Henry F.Korth, S.Sudarsha. Database System Concepts,6E.机械工业出版社.

[3] 世界领先！一文详解 OceanBase 的高可用及容灾方案

https://mp.weixin.qq.com/s?__biz=MzU0ODgOOTIyNw==&mid=2247485915&idx=1&sn=68d57a3bc0cb8ae543e0c85aeb28e83e&source=41#wechat_redirect