

同济大学计算机系

数据库系统原理前沿调研报告



学 号	<u>2152813 2151769 2152513 2152809</u>
姓 名	<u>陆宇龙 吕博文 肖博文 曾崇然</u>
专 业	<u>计算机科学与技术</u>
授课老师	<u>关信红老师</u>

目录

1. 报告概述.....	3
1.1. 调研背景	3
1.2. 选题分析	3
1.2.1. 高并发技术手段分析.....	4
1.2.2. 高性能技术手段分析.....	4
1.2.3. 高可用技术手段分析.....	4
2. OceanBase 高并发处理技术分析.....	5
2.1. 并发性概念.....	5
2.2. OceanBase 并发控制手段	5
2.2.1. 串行执行.....	5
2.2.2. 可串行化的实现.....	6
2.2.3. 隔离级别实现性能的提高	6
2.2.4. OceanBase 使用的隔离级别.....	6
2.3. OceanBase 一致性问题解决	7
2.3.1. 使用多版本存储解决外部一致性问题	7
2.3.2. 基于互斥锁的多版本并发控制	8
2.3.3. OceanBase 事务调度具体分析	9
2.4. OceanBase 高并发性能优化措施.....	10
2.4.1. 负载均衡技术	10
2.4.2. 流量漏斗技术	10
2.4.3. CDN 技术	11
2.4.4. 池化技术.....	11
3. OceanBase 高性能保持技术分析.....	12
3.1. 分布式部署时实现高性能	12
3.1.1. 市面上数据库产品性能现状.....	12
3.1.2. OceanBase 在分布式部署时实现高性能的技术手段.....	12
3.2. 弹性伸缩和负载均衡实现高性能	14
3.2.1. 弹性伸缩.....	15
3.2.2. 负载均衡.....	15
3.2.3. 实际场景表现	15
4. 高可用保障技术.....	16
4.1. 容灾技术	17
4.1.1. 多副本容灾.....	17
4.1.2. 基于仲裁的容灾.....	17
4.1.3. 物理容灾.....	18
4.2. 故障探测和黑名单技术.....	18
4.3. 动态调节技术.....	19
4.3.1. 集群.....	19
4.3.2. 熔断.....	20
4.3.3. 降级与限流.....	20
5. 总结与展望	20
6. 参考文献.....	21

分布式数据库 OceanBase 高并发特性的相关技术研究

1. 报告概述

1.1. 调研背景

OceanBase 是由蚂蚁集团完全自主研发的国产原生分布式数据库。其创始于 2010 年，已连续 10 年平稳支撑“双 11”购物节等高并发访问场景。

OceanBase 数据库在 TPC-C 和 TPC-H 测试上刷新了世界纪录，其中 TPC-C 测试是针对联机事务处理数据库，通过模拟批发市场系统，用每分钟处理的新订单数衡量数据库性能的测试。TPC-H 测试是针对联机分析处理数据库，通过模拟决策支持系统的数据分析查询操作，用查询响应时间来衡量数据库性能的测试。

历经十余年的发展，OceanBase 数据库作为一款性能优秀的原生分布式数据库系统，凭借其优异的高并发性能，在国计民生许多重要行业得到了广泛应用。OceanBase 数据库被应用在政企、金融、保险、互联网企业等重要领域，为社会数字化转型提供了有力的支持。这些数据和案例充分表明，在高并发，大负载的工作环境下，OceanBase 数据库能够安全稳定运行，同时保持良好的性能。

1.2. 选题分析

如何在高并发环境下保持数据库系统的稳定性和查询性能，是 OceanBase 数据库在并发领域的关键核心技术。本次报告将围绕这一技术展开调研。

要实现一个数据库系统的高并发特性，关键在于优化三个核心技术领域，即高并发处理技术、高性能保持技术以及高可用保障技术。

1.2.1. 高并发技术手段分析

高并发处理对于数据库系统来说至关重要。面对极高的 QPS（每秒点击率）和高频率请求，如何有效地分配和利用软硬件资源是这项技术要解决的核心问题。为了实现这一目标，数据库系统可以采用多种技术手段，如负载均衡、流量漏斗、CDN 技术、以及池化技术等。这些技术手段可以组合使用，以实现在一定程度上提高数据库系统应对短时大流量访问的能力，从而确保系统在高并发环境下的稳定性和可靠性。

1.2.2. 高性能技术手段分析

在高并发环境下，数据库系统需要保持较快的访问速度和合理的响应时间，以确保用户体验和系统效率。为了实现这一目标，数据库系统可以采用多种技术手段，如缓存、异步处理、I/O 优化、以及数据库结构优化等。这些技术手段涵盖了计算机存储、网络传输、系统设计、软件算法等众多领域，为数据库系统的高性能提供了有力支持。然而，这一领域仍然存在巨大的研究空间，需要不断地探索和创新。

1.2.3. 高可用技术手段分析

高可用保障技术是确保数据库系统在高并发环境下稳定性和数据安全性的关键。当数据库系统负载过大，超过自身处理能力时，可能会出现阻塞、宕机、崩溃等严重问题，这将直接影响上层应用程序的运行。由于数据库系统存

储的数据往往涉及金融、政务等关键敏感信息，数据的安全性至关重要。因此，数据库系统需要设计多级安全保障机制，如访问控制、加密传输、数据备份与恢复等，以确保数据的安全性和完整性。同时，还需要建立完善的监控和报警机制，以便及时发现并处理潜在的安全威胁和故障问题。

综合运用这些关键技术，可以构建出具有高并发特性的数据库系统，为现代应用程序提供稳定、高效、安全的后端支持。下面报告将从上述三个方面对 OceanBase 数据库使用的技术展开具体分析。

2. OceanBase 高并发处理技术分析

2.1. 并发性概念

为了提高事务的处理能力，数据库允许事务并发地访问与修改同一个数据的行为称之为数据并发性，数据库并发性能的好坏通常由系统响应时间，吞吐量，并发事务处理数来衡量，提升高并发性能一直以来是数据库技术方面的主要研究方向与目标。

2.2. OceanBase 并发控制手段

2.2.1. 串行执行

最简单的并发控制就是串行（serial）执行，其中串行执行是指一个进程在另一个进程执行完(收到触发操作的回应)一个操作前不会触发下一个操作。但这明显不符合高并发的需求。

2.2.2. 可串行化的实现

串行执行不能满足高并发的需求，由此产生了可串行化的概念，即可以通过并行地执行事务内的多个操作，最终达到和串行执行相同的结果，两种常见的实现机制有两阶段锁和乐观锁机制。前者是通过排它地通过加锁限制其他的事务的冲突修改，并通过死锁检测机制回滚产生循环的事务保证无环；后者通过在提交时的检测阶段，回滚所有可能会导致异常的事务保证不会产生异常。

2.2.3. 隔离级别实现性能的提高

两阶段锁和乐观锁机制提供可串行化隔离级别会极大地影响性能，强制可串行化隔离往往伴随着大量的冲突等待和冲突失败，不符合商用数据库的性能要求，于是，数据库不得不放宽调度的验证，允许更多非可串行化的调度被执行，此时，多个事务并发执行的结果可能不再等价于任何一种串行执行的结果，此时数据库需要给用户做出保证：什么样的错误会发生，而什么样的错误不会发生，这些不同的保证就是数据库的隔离级别。

2.2.4. OceanBase 使用的隔离级别

其中，快照读和读已提交是比较常见的允许异常的并发控制，也是 OceanBase 数据库的选择，也就是说 OceanBase 数据库主要采用多版本并发控制模型下快照读、读已提交的两种隔离级别，并在分布式语义的隔离级别上保证外部的一致性。

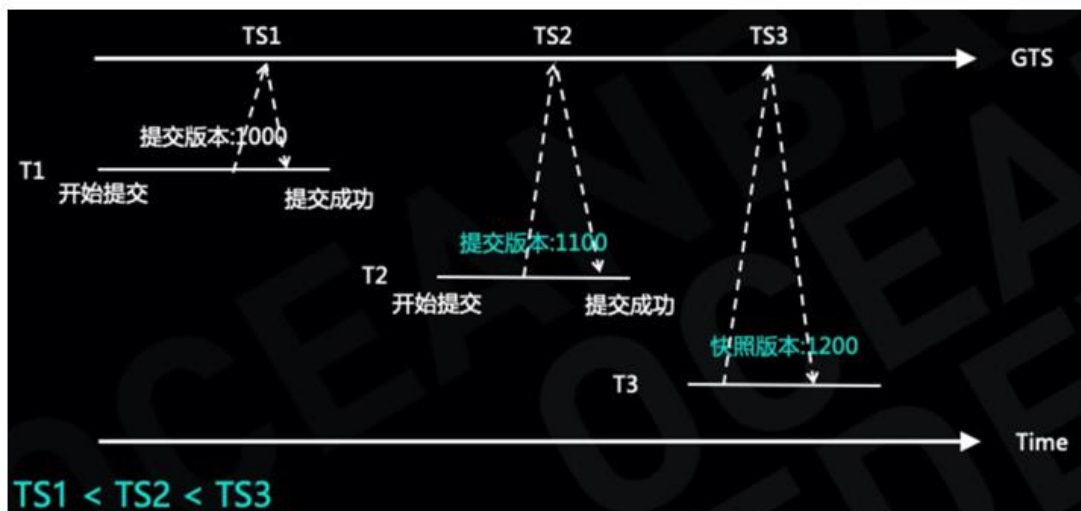
2.3. OceanBase 一致性问题解决

2.3.1. 使用多版本存储解决外部一致性问题

为了支持读写不互斥，解决外部一致性的问题，OceanBase 数据库从开始就选择了多版本作为存储，并让事务对于全局来说，维护两个版本号，读版本号 and 提交版本号，如图所示的本地最大读时间戳和最大提交时间戳。其次，在内存中会为每一次更新记录一个新的版本（可以做到读写不互斥）。



或者通过下图来观察，我们可以看出事务 T1 和事务 T2 在提交过程中分别向全局时间戳服务申请一个时间作为事务的提交版本号，事务 T3 也从全局时间戳服务申请一个时间戳做为事务的快照，显然全局时间戳服务可以保证 $TS1 < TS2 < TS3$ ，假如事务 T3 能够读到 T2 修改的数据，那么 T3 肯定也能够读到 T1 修改的数据，解决了外部一致性的问题。

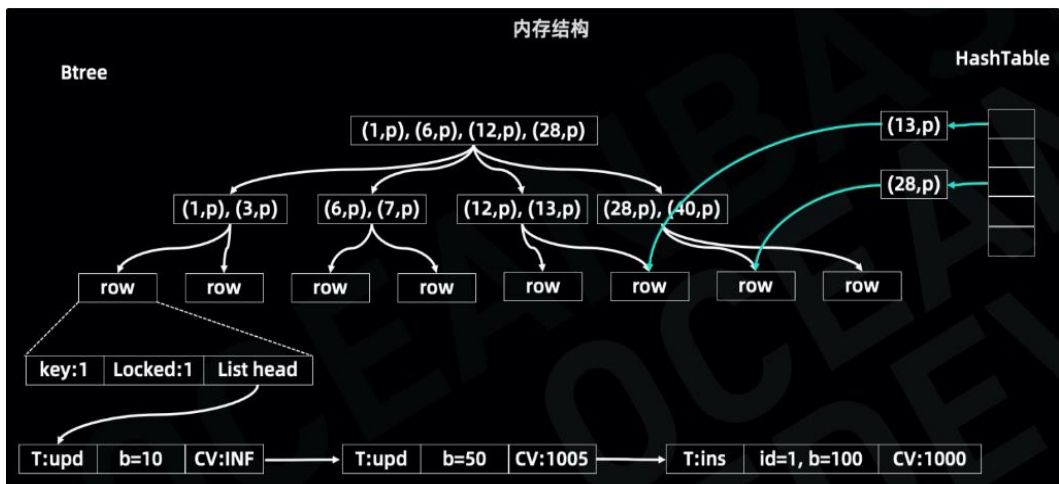


2.3.2. 基于互斥锁的多版本并发控制

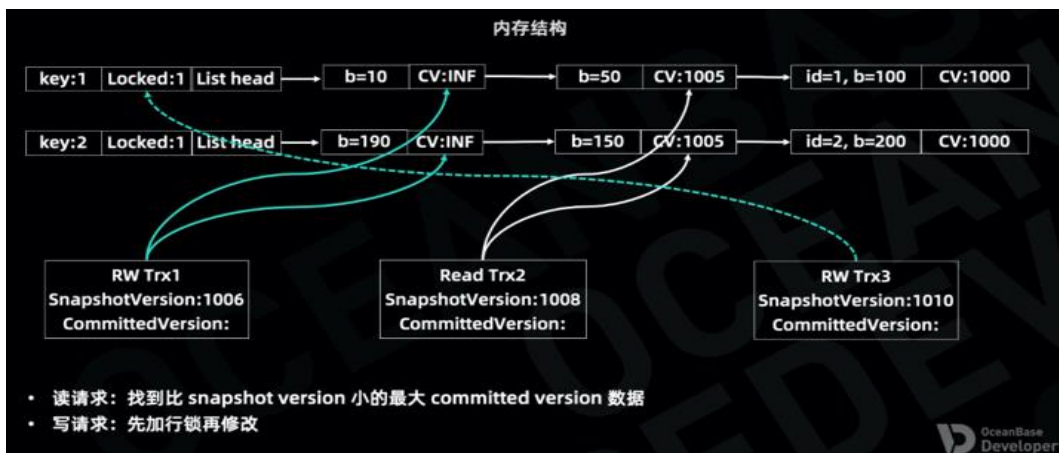
在多版本管理方面，OceanBase 采用基于互斥锁的多版本并发控制，其中 OceanBase 存储引擎采用的是 LSM tree 架构，将数据拆分成静态数据和动态数据，动态数据保存在 Memtable 中并定期 dump 到磁盘中，Memtable 采用 B+tree 以及 Hash 双索引结构对数据进行存储，其中 B+tree 用于范围查询，Hash 用于单行查询。

B+tree 的叶子节点保存了行数据的元信息，元信息里面有很多字段，这里只介绍 3 个字段，主键信息、锁信息以及链表指针：

- (1)锁信息表示是否有事务持有行锁，事务在修改数据之前需要先加行锁；
- (2)链表信息指向多个版本的数据，每个版本只保存增量信息，比如某一次修改只修改一个字段，增量信息只会记录该字段的变化的情况；



2.3.3. OceanBase 事务调度具体分析



OceanBase 分布式数据库系统，对事务进行调度，确保并发事务不会出现一致性问题。假设有 3 个并发事务，分别为读写事务 T1，只读事务 T2，读写事务 T3，其中事务 T1 还没有提交，持有行的行锁。

事务 T3 在对行 1 进行修改之前需要先持有行 1 的行锁，由于行 1 的行锁被事务 T1 持有，因此 T3 需要等待，直到事务 T1 提交成功并且将行锁解开。事务 T2 是一个只读事务，事务的快照版本号是 1008，在读取之前首先通过索引结构找到行的元数据，然后根据快照版本号找到比快照版本号小的最大提交版本号的数据。从下图可以看出事务 T2 能够读到提交版本号是 1005 的数据。以上就是 OceanBase 内部的读写并发控制机制，通过行上的互斥锁来

解决写写冲突的问题，通过多版本机制保证写不阻塞读，读不阻塞写。

OceanBase 的多版本并发控制实现非常的简单，快照版本就是一个时间戳，通过比较时间戳的大小就可以确定事务的可见性，不需要维护活跃事务。在有些分布式数据库系统里，维护了全局事务管理器，这个全局事务管理器本质上就是用来确定事务的快照的，当并发的事务比较多的时候，全局事务管理器容易成为集群的瓶颈，OceanBase 不需要维护全局事务管理器。还有一点就是 OceanBase 行的元数据上保存了锁信息，不需要额外的锁管理器。

2.4. OceanBase 高并发性能优化措施

在 OceanBase 数据库实现高并发性能的基础原理之上，下面部分介绍 OceanBase 为保证高并发性能所采取的一些优化措施：

2.4.1. 负载均衡技术

OceanBase 采用负载均衡技术来确保集群中的各个节点负载均衡，防止某个节点成为瓶颈影响整体性能。负载均衡技术通过动态监测各节点的负载情况，将请求合理地分配到不同的节点上，使得每个节点的负载相对均衡。这可以通过实时监测节点的资源利用率、处理能力等指标来实现。负载均衡技术有效地提高了整个系统的并发处理能力，确保每个节点都能够充分发挥其性能优势，提高系统的稳定性和可靠性。

2.4.2. 流量漏斗技术

为了应对高并发的流量冲击，OceanBase 引入了流量漏斗技术。流量漏斗技术主要通过对请求进行限流和调整流量的速率，以防止系统因过高的并发访

问而崩溃或性能下降。通过设置合理的阈值和限制规则，系统可以在高负载时拒绝一部分请求，确保系统在可承受范围内运行。这有助于防止系统过载，保障关键服务的稳定性，并在一定程度上优化系统性能。

2.4.3. CDN 技术

内容分发网络（CDN）技术被广泛应用于提高数据传输效率和减轻服务器压力。OceanBase 采用 CDN 技术来加速静态资源的传输，例如图片、CSS 和 JavaScript 等。通过将这些静态资源缓存到离用户更近的 CDN 节点上，可以有效减少数据传输的延迟，提高用户访问速度。这对于提高系统的并发性能和用户体验至关重要，特别是在大规模分布式系统中。

2.4.4. 池化技术

池化技术是一种资源管理的方法，通过对数据库连接、线程等资源进行池化，可以有效减少资源的创建和销毁开销，提高资源的重复利用率。在高并发场景下，频繁地创建和销毁资源会导致系统性能下降，而通过池化技术，可以事先创建一定数量的资源池，在需要时从池中获取资源，使用完毕后再放回池中，减少资源的频繁创建和销毁，提高系统的响应速度和资源利用率。

OceanBase 通过池化技术有效地管理数据库连接、线程等资源，为高并发场景提供了稳定的支持。

通过以上优化技术的综合应用，OceanBase 数据库能够更好地满足高并发性能的需求，保障系统的稳定性、可靠性和用户体验。这些优化措施共同构建

了一个高效的数据库系统，使其在处理大规模并发请求时能够保持较低的响应时间、高吞吐量，并确保数据一致性和可靠性。

3. OceanBase 高性能保持技术分析

3.1. 分布式部署时实现高性能

3.1.1. 市面上数据库产品性能现状

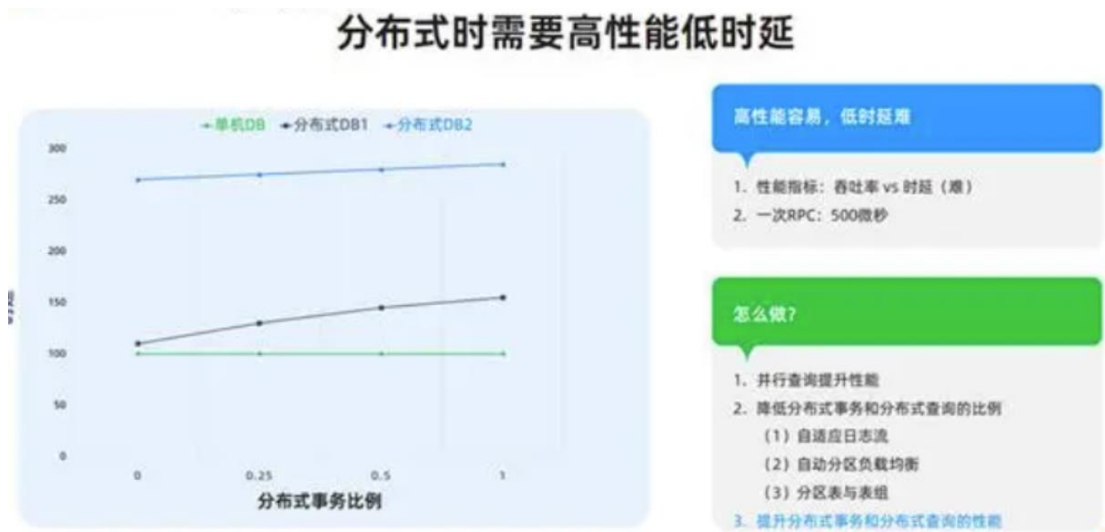
随着大数据、5G 和 AI 的发展，目前市场上的许多数据库产品已经十分成熟。在数据量不是特别大的情况下，性能较好的单机数据库足以应付很多场景。

但当数据库数量变大以后，一些产品的缺点就暴露出来：尽管一些产品采用了可扩展、灵活伸缩的分布式架构，却不支持 SQL；还有些虽然支持一部分 SQL 功能，也支持一些分布式的能力，但是却牺牲掉了部分单机性能，因此用户在面临数据库选型时就会陷入困扰——没有一款数据库融合了集中式和分布式的双重技术优势。

3.1.2. OceanBase 在分布式部署时实现高性能的技术手段

基于这样的市场需求，OceanBase 研发团队采用了单机分布式一体化架构，在保证数据库可扩展、高可用的分布式特性的前提下还不会牺牲集中数据库的优势。在此架构下，分布式部署时要实现和确保高性能、低时延的特性，需要进行特殊设计。随着分布式事务比例增加，单机数据并没有什么明显变化，但随着分布式事务增加，分布式数据库具有有通信等开销，性能与单机数

数据库有差距。OceanBase 采取了两种方法来解决以上问题。



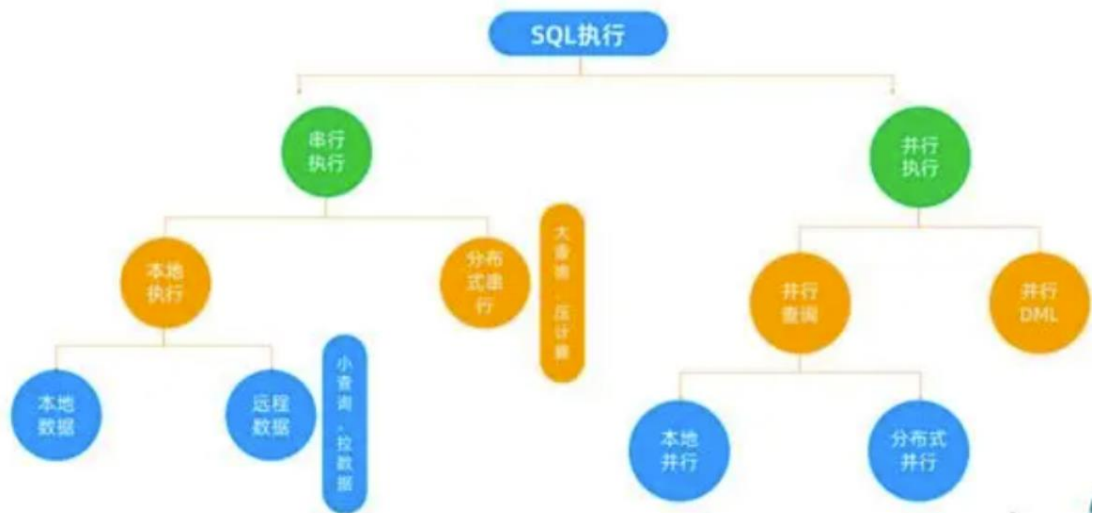
3.1.2.1. 合理确定分布式事务和分布式查询的比例

对于很多在线型的交易，全局性的事务，分布式事务的比例往往并不多。在典型的 TPC-C 场景下，TPC-C 里跨 warehouse 的交易查询只有 10%，即并不是所有东西都要付出分布式开销。对此，OceanBase 团队开发了自适应日志流。

OceanBase 做了很多自适应的策略，去自动化的根据表的设计去做自动的负载均衡，自动的做数据排布减少分布式事务和分布式查询的比例，如果这些自动化的措施都达不到这个目标，用户同样可以使用 OceanBase 团队提供的分区表和表组的功能自己去控制、降低分布式事务。

3.1.2.2. 并行查询提高性能

如果没有办法优化业务来减少分布式 SQL，对于分布式 SQL 这部分，OceanBase 会自适应执行引擎，分为串行执行和并行执行。



对于并行执行，OceanBase 的执行引擎可以在单机内进行并行处理，这相对于一些开源的单机数据库具有优势。早期的 MySQL 版本无法在单机内进行并行处理，而 OceanBase 可以利用多机资源，如果一台机器的 CPU 不足够，可以利用整个集群的 CPU 资源来实现并行执行。

对于串行执行，OceanBase 明确区分了大数据量访问和小数据量访问。对于小数据量访问，可以将数据拉到本机上执行，减少网络传输延迟。而对于大数据量访问，可以将整个 SQL 的执行计划下推到具体的机器上进行数据过滤。这个过程是优化器根据自适应性选择的。

通过这种自适应执行的方式，OceanBase 可以根据数据量的大小和网络延迟等因素灵活选择并行执行或串行执行，并利用多机资源进行并行处理，以最大程度地提高分布式 SQL 的执行效率和性能，进而提高 Oceanbase 的性能，实现数据库的高性能。

3.2. 弹性伸缩和负载均衡实现高性能

OceanBase 数据库运用了弹性伸缩和负载均衡的基本概念。弹性伸缩是指

根据系统需求动态调整计算资源，以便在负载变化时保持高可用性。负载均衡则是将大量计算请求分散到多个计算节点上，以实现更高的系统性能。

3.2.1. 弹性伸缩

在 OceanBase 中，弹性伸缩和负载均衡的实现依赖于其独特的分布式架构。OceanBase 采用多副本复制技术，通过将数据分散存储在多个节点上，实现了高度的可扩展性和容错性。当系统负载增加时，OceanBase 可以自动增加节点数量，从而提升整体性能。而当负载降低时，系统则会自动缩减节点数量，以降低成本。这种自适应的弹性伸缩功能使得 OceanBase 能够在不同负载情况下保持出色的性能表现。

3.2.2. 负载均衡

另一方面，OceanBase 还实现了精细的负载均衡机制。通过智能地分配和调度请求，OceanBase 能够将大量请求均匀地分配给各个计算节点，确保每个节点都处于合理的负载状态。这种负载均衡策略不仅可以提高系统性能，还能够有效避免节点过载和资源浪费。

3.2.3. 实际场景表现

在实际应用场景中，OceanBase 的弹性伸缩和负载均衡功能表现得尤为突出。例如，在一个电商网站的高峰期，OceanBase 可以通过自动增加节点数量，轻松应对瞬时的高并发流量。而在深夜或平峰期，系统则会自动缩减节点数量，实现成本控制。这种智能的资源管理方式使得 OceanBase 能够在保证高性能的同时，降低运营成本。

OceanBase 的负载均衡机制还使其能够应对复杂多变的应用场景。无论是读操作还是写操作，OceanBase 都能通过智能调度算法将请求分配到合适的计算节点上，确保整个系统的性能和稳定性。这使得 OceanBase 在处理大规模数据存储、实时数据分析等复杂应用时表现出色。

OceanBase 的弹性伸缩和负载均衡原理是其分布式数据库系统的核心优势之一。通过自适应的资源管理和精细的请求调度策略，OceanBase 能够为用户提供出色的性能、高可用性和灵活性。未来，随着数据规模的不断扩大和业务需求的不断复杂化，OceanBase 弹性伸缩和负载均衡原理的技术还将继续优化，从而提供更高效稳定的数据库的服务。

4. 高可用保障技术

数据库系统在教育架构中承担了数据存储和查询的功能，其对企业数据安全和保障业务连续性至关重要。高可用是分布式数据库系统架构设计中首要考虑的因素，也是保证数据库高并发性能的重要环节。高可用诉求包括服务高可用和数据高可靠。OceanBase 数据库具备多样化的服务高可用技术，其中最重要的技术是容灾技术。容灾技术一方面保证了数据库的数据安全。另一方面，在遇到小型事故时，OceanBase 数据库可以自动隔离、自动处理这些事故，尽可能使系统在最短时间内恢复正常，保证系统在遭受小型干扰事故时仍然能够有足够的软硬件资源，以保证高并发性能。

4.1. 容灾技术

数据库的容灾是指为了应对数据库可能遭遇的灾害（如自然灾害、人为错误或系统故障等），而采取的一系列预防和恢复措施。这些措施包括数据备份、数据恢复、系统冗余等。容灾的目的是确保数据库在遭遇灾害时能够快速恢复，并尽量减少数据丢失或服务中断的影响。容灾技术等高可用保障技术是保证数据库在部分设备异常环境下保持高并发性能的重要一环，是数据库在实际运行环境下性能的另一种体现方式。对于 OceanBase 数据库，其采取了包括集群内的多副本容灾和基于仲裁的容灾以及集群间的物理备库容灾。

4.1.1. 多副本容灾

OceanBase 数据库基于 Paxos 协议实现了多副本容灾方案，为用户提供少数派故障时的高可用能力。Paxos 协议是一种解决分布式系统中多个节点之间就某个值达成一致的通信协议。多副本容灾技术面向单个集群，事务日志持久化并在多个副本之间同步日志数据，基于 Paxos 协议保证日志数据在多数派副本持久化成功，同时通过成员变更提供容灾能力。多副本容灾面向集群内少数派节点异常的场景，具备极好的故障恢复速度及数据零丢失能力。

4.1.2. 基于仲裁的容灾

基于仲裁的容灾方案，是 OceanBase 数据库以 Paxos 多副本容灾方案为基础创新性地提供的高可用解决方案。在业务层面上，仲裁方案保证数据在多数派副本或全部副本上强同步，并在半数全功能型副本故障的情况下，自动进行故障降级，保证数据不丢失的同时业务持续可用。基于仲裁的容灾同时解决

了传统数据库最大保护或最大可用方案中，业务服务连续性和数据完整性不可兼得的问题，可以避免出现数据不一致风险。

4.1.3. 物理容灾

物理备库容灾是 OceanBase 数据库高可用解决方案的重要组成部分。物理备库容灾技术面向多个集群，多集群间传输事务日志，建立基于日志的物理热备服务。物理备库采用独立的主备库架构，主备关系存在于租户级别，主备之间通过网络直连或第三方日志服务建立传输渠道，只传输日志。不同于集中式架构，独立主备库架构下，各个集群是相互独立的，用户可以更加灵活地管理群。

4.2. 故障探测和黑名单技术

OceanBase 数据库主要将数据库故障分为两类，主要包括机器和进程故障(机器宕机、网络故障、进程 core 等机器硬件问题或进程问题)和业务逻辑问题(OceanBase 数据库自身逻辑问题导致服务不可用)。当前 OceanBase 的策略是在处理 OBServer 节点机器和进程故障的基础上，根据一些已知现象去处理一些业务逻辑问题，如 OBServer 节点返回的特定错误码（如机器内存不足）、OBServer 节点长时间无响应等。在分布式系统中，故障探测的结果有三种：成功、失败和超时。其中，超时是最难处理的情况。比如一个 SQL 执行了 100s 仍然未返回结果，此时很难确定执行这个 SQL 的 OBServer 节点是否正常，有可能仅仅是 OBServer 节点执行速度慢，但也有可能是 OBServer 节点出现了问题。如果慢 SQL 出现的频率过高，也会成为数据库系统高并发性能的制约因

素。对于业务逻辑引发故障，与数据库的实现紧密相关，很难去确定问题边界并解决所有问题。要进一步提升 OceanBase 数据库的性能，这是一个值得探索的前沿方向。

黑名单机制指的是系统探测出 OBCServer 节点状态后，先修改黑名单，然后根据黑名单进行节点拉黑或节点洗白操作。根据不同的探测机制，数据库系统实现了状态黑名单、探测黑名单和活不可用黑名单三种不同的黑名单，根据黑名单信息的来源进行区分。当某个 OBCServer 宕机或因其他原因停止提供服务时，黑名单机制能够迅速将其加入黑名单，从而避免其他应用或请求再次访问该节点。这可以防止因无效或故障的访问请求导致的系统瓶颈或性能下降，从而提高系统的并发性能和整体稳定性。

4.3. 动态调节技术

OceanBase 数据库高可用特性具体包括集群、熔断、降级、限流。这些高可用特性都是为了确保 OceanBase 数据库在面临故障或高并发访问时，能够提供稳定可靠的服务。在实际应用中，根据具体需求和场景，可以选择不同的高可用方案，以确保数据的可靠性和业务的连续性，保证 SQL 查询请求的执行效率。

4.3.1. 集群

当主节点出现故障时，备节点可以自动检测到这一情况，并快速切换为主节点，继续提供服务。这种自动切换机制可以确保服务的连续性和稳定性，减少因节点故障导致的业务中断时间。同时，OceanBase 数据库还支持跨机房部

署，可以在不同地理位置的节点之间进行数据同步，进一步提高数据的可靠性和可用性。

4.3.2. 熔断

熔断机制是一种在分布式系统中的故障隔离策略，它的目的是防止系统因部分组件的故障而连锁反应，导致整个系统的瘫痪。当某个节点或组件出现异常情况时，如 CPU 使用率过高、内存溢出、网络延迟增大等，系统会迅速捕捉到这些异常信号。熔断机制会迅速地将其从系统中隔离，以避免故障的扩散和影响范围的扩大，引发数据库整体并发性能下降

4.3.3. 降级与限流

当系统面临巨大的压力，如高并发请求、数据量剧增或某些非核心功能异常时，为了确保核心业务的正常运行，可以手动触发降级机制。降级机制的目的是通过关闭部分功能或降低服务性能，来减轻系统的压力，从而确保系统的稳定性和可用性。当并发访问量超出其承载能力时，限流机制可以确保系统不会因过载而出现服务中断或性能下降。通过限制并发访问量，可以确保系统始终运行在稳定的状态下，保障核心业务逻辑的性能。

5. 总结与展望

OceanBase 数据库之所以拥有强大的高并发能力，关键在于其在分布式数据库基本原理的基础上，通过软硬件调度技术，合理地将高并发事务分配到适合的硬件设备上进行处理，并在软件层面上通过优秀的算法将高并发事务进行

任务分解和调度。在保持业务逻辑的正确性的前提下，OceanBase 数据库充分利用了并行计算等技术手段，通过软件算法组织计算机系统计算资源。以上技术是 OceanBase 数据库能够取得获得优异高并发性能的根本保证。

OceanBase 数据库强大的高并发性能，与其原生的分布式架构密切相关。分布式架构数据库能够依托更多的分布式的硬件资源，对大规模访问流量进行逐级分流，由此获得高并发和高性能。但是相较于单机数据库，分布式架构系统对于系统结构管理，数据同步与通信，负载分配与平衡提出了非常高的要求。如何实现并优化上述分布式数据库技术，也是当今数据库探索的前沿领域。

OceanBase 作为商用数据库系统，如何降低其在真实应用场景下的使用成本，同时使上述前沿技术充分发挥效能，体现在高并发性能的提升上，是一项及其复杂的系统工程。将高并发处理技术、高性能保持技术以及高可用保障技术进行技术组合，同时获取数据库性能和使用成本两者的平衡，需要长期的应用实践探索。

6. 参考文献

[1] OceanBase 数据库. 并发控制概述 [EB/OL].(2023-07-18)[2023-12-26]. <https://www.oceanbase.com/docs/enterprise-oceanbase-database-cn-10000000000356648>

[2] OceanBase 数据库. OceanBase 高并发场景技术解读[EB/OL]. (2023-10-25)[2023-12-26]. <https://open.oceanbase.com/blog/7499503392>

[3] 科技观察说. OceanBase 如何在分布式部署时，实现高性能低时延？ [EB/OL]. (2023-04-23)[2023-12-26]. <https://baijiahao.baidu.com/s?id=1763932725147352375&wfr=spider&for=pc>

[4] 百度开发者中心. 弹性伸缩：OceanBase 的高性能秘诀[EB/OL].(2023-07-17)[2023-12-26]. <https://developer.baidu.com/article/detail?tid=434558>

[5] 51CTO 博客. OceanBase 架构实现高性能 OceanBase 架构介绍[EB/OL]. (2023-10-18)[2023-12-26]. https://blog.51cto.com/u_16213596/7917490

小组分工：

整体规划，总结整合：2152813-陆宇龙

高并发技术实现部分调研：2151769-吕博文

高性能技术实现部分调研：2152513-肖博文

高可用技术实现部分调研：2152809-曾崇然