

OceanBase 数据库社区版 V4.1.0

[版本概要](#)

[特性更新](#)

[功能增强](#)

[MySQL 兼容](#)

[性能提升](#)

[稳定性增强](#)

[高可用增强](#)

[运维能力提升](#)

[产品形态](#)

[性能报告](#)

[SYSBENCH OLTP 负载测试](#)

[BMSQL TPCC 测试](#)

[TPCH 测试](#)

[TPC-DS 测试](#)

[兼容性变更](#)

[产品行为变更](#)

[视图/实体表/虚拟表](#)

[配置项变更](#)

[升级说明](#)

[周边配套](#)

[约束限制](#)

OceanBase 数据库 V4.0 社区版本是对分布式数据库系统架构设计的全面升级，OceanBase 数据库 V4.1 版本面向公有云&开源用户场景需求，在对功能健全优化的同时，不断对产品进行打磨和完善，在对 MySQL 8.0 兼容、性能、易用性、成本方面进行优化增强，目标支持项目规模化复制，是真正可全面商业化推广的版本。

版本概要

本版本的核心能力特性如下：

- 功能增强。
支持 SQL 级资源隔离，租户级 IOPS 隔离优化，支持 Latin1 字符集等。
- MySQL 兼容性增强。
MySQL 8.0 版本系统函数、变量和 SQL MODE 系统化完善，支持 GIS 数据类型。
- 性能提升。
 - TP 性能优化。
 - SYSBENCH 性能模型，综合读写性能（Read Write）1024 并发测试性能相比于 V4.0 版本提升约 40%。
 - TPCC 性能模型，tpmC (NewOrders) 1000 warehouse 测试性能相比于 V4.0 版本提升约 8%。
 - AP性能优化。
 - TPC-H 查询性能优化，100GB 数据量顺序执行 22 条 SQL，整体性能相比于 V4.0 版本提升约 17%。
 - TPC-DS 查询性能优化，100GB 数据量顺序执行 99 条 SQL，整体耗时约 161s。
 - 数据导入性能优化，在 c6a.12xlarge 规格（48vCPU，96GB 内存）机器上使用旁路导入 LINEITEM 表（74GB），单 ODS 节点导入速率可以达到 165MB/s。
 - 基于 OBProxy 的分布式事务路由优化，提升分布式事务性能。
 - 其它性能优化：Nest Loop Join 算子性能优化、Index Skip Scan 执行优化、Truncate Table 并行执行优化、编译反馈优化。
- 稳定性增强。
扩展支持 LOB 存储规格上限。
- 高可用增强。
支持基于归档日志的物理备库，支持更多场景达成 RTO 时间降低到 8s 以内。
- 运维能力提升。
 - 支持 4.0 系列版本在线升级。
 - 实时统计信息收集。
 - 日志系统与可读性优化整改。
 - 持久化与存储优化：自适应合并（Compaction）、小表空间优化等。
- 产品形态。
发布单机产品形态。

特性更新

功能增强

- 支持 SQL 级资源隔离。

在实际项目交付中我们发现，客户业务希望能够对资源隔离进行更细微的控制，比如某个用户执行不同 SQL 使用不同的资源规格进行隔离（同一个租户下不同 user 的资源隔离、user 执行的不同 SQL 使用不同的资源），通过这种细粒度的应用方式帮助业务分配和隔离资源（通常为 OLAP 和 OLTP 业务的资源），减少业务之间的互相影响。

OceanBase 数据库 V4.1 版本支持创建资源组，支持绑定用户与资源组，后续用户在执行的业务 SQL 时，OceanBase 数据库根据预定义的 SQL 规则匹配相应的资源组，通过使用 `DBMS_RESOURCE_MANAGER` 系统包帮助用户自定义业务资源隔离方案策略，当前该资源隔离方案仅支持 MySQL 模式下对 CPU 的资源隔离。

- 租户级 IOPS 隔离优化。

OceanBase 数据库 V4.0 版本开始支持租户级 IOPS 隔离，通过在租户间按照 Unit 划分和租户内按照 IO 类别（category）划分的二级方式，可以基本保证租户内的不同任务只使用分配好的资源而不会挤兑其他任务，但同时也存在其他一些问题，例如 1) 固定的 IO 类别不能支持用户根据任务作业定制，2) 同一租户下的不同用户无法再进行资源隔离。因此我们将 IO 资源与 CPU 资源一样纳入到 Resource Manager 框架，实现用户界面统一的资源隔离体系。

OceanBase 数据库 V4.1 版本通过 `DBMS_RESOURCE_MANAGER` 系统包来支持用户自定义 IO 资源隔离策略，通过在包函数 `CREATE_PLAN_DIRECTIVE` 设置 `MIN_IOPS`、`MAX_IOPS` 和 `WEIGHT_IOPS` 参数来设置隔离计划，IOPS 隔离方案同时支持 Oracle 租户与 MySQL 租户。

- 支持 Latin1 字符集。

随着 OceanBase 数据库社区用户和海外客户数目的增多，越来越多的项目提出了 Latin1 字符集的需求，希望可以在数据库层面进行支持从而避免项目迁移过程中复杂的字符集转换工作。因此 OceanBase 数据库在 V4.1 版本对 Latin1 字符集进行新增支持，同时在 MySQL 模式支持 `latin1_bin` 和 `latin1_swedish_ci` 字符序，在 Oracle 模式下支持 `latin1_bin` 字符序。

MySQL 兼容

- MySQL 8.0 兼容，系统函数、变量和 SQL MODE 系统化完善。

OceanBase 数据库 V4.0 及之前版本已经兼容了 MySQL 5.7 的大部分功能，但随着 OceanBase 数据库在公有云和社区上有了更多 MySQL 资深使用客户，其业务系统对 MySQL 8.0 存在特性依赖这个矛盾诉求愈加强烈。同时考虑到 MySQL 8.0 版本发布已有六七年时间，其版本特性也足够成熟。OceanBase 数据库从 V4.1 版本开始对 MySQL 8.0 能力进行系统化兼容支持，V4.1 版本主要包括如下：

- 系统函数

- 字符运算函数：`BIN_TO_UUID()`、`CHARACTER_LENGTH()`、`LOAD_FILE()`、`IS_UUID()`、`UUID_TO_BIN()`、`OCTET_LENGTH()`。
 - 时间日期函数：`ADDTIME()`、`DAYNAME()`。
 - 加解密函数：`DECODE()`、`DES_DECRYPT()`、`DES_ENCRYPT()`、`ENCODE()`、`ENCRYPT()`。
 - Perf 信息函数：`FORMAT_BYTES()`、`FORMAT_PICO_TIME()`。
 - 信息函数：`CURRENT_ROLE()`、`ICU_VERSION()`。

- 窗口函数：BIT_AND()、BIT_OR()、BIT_XOR()。
 - 其他函数：NAME_CONST()。
 - SQL Mode
 - 独立SQL Mode：REAL_AS_FLOAT、TIME_TRUNCATE_FRACTIONAL。
 - 组合 SQL Mode：ANSI、DB2、MAXDB、MSSQL、ORACLE、POSTGRESQL、TRADITIONAL。
 - Information Schema
 - 视图兼容性：视图字段没有变化，字段类型或宽度或约束有兼容性调整。如 VIEW_TABLE_USAGE、VIEWS、TABLES、STATISTICS、ENGINES、TABLE_CONSTRAINTS、REFERENTIAL_CONSTRAINTS、CHECK_CONSTRAINTS、PARAMETERS、ROUTINES、PARTITIONS 等。
 - 性能提升：基于本地表索引，普遍提升系统视图查询性能，如 STATISTICS 等。
 - Float(m, d)：完全兼容 MySQL 8.0 数据类型与精度。
 - 支持 GIS 数据类型。
- 空间信息系统（GIS）提供了对空间对象（点、线、面以及复杂的空间对象）的存储、计算、索引能力，在交通出行行业有着广泛的应用，可以帮助用户快速构建空间计算能力。OceanBase 数据库 V3.2.4 版本开始支持 GIS，OceanBase 数据库的 GIS 设计选择了基于四叉树空间划分的索引方案，支持兼容 MySQL 8.0 的 GIS 数据类型，支持 SRS（空间参考坐标系）元数据的管理及缓存，支持基于 QuadTree 的空间索引，支持单值类型（GEOMETRY/POINT/LINESTRING/POLYGON）和多值类型（MULTIPOINT/MULTILINESTRING/MULTIPOLYGON/GEOMETRYCOLLECTION），并实现 MySQL 8.0 常用的空间计算函数。

性能提升

- 数据导入性能优化。
- 数据导入性能是 OLAP 领域内的一个重要衡量指标，在数据分析/大数据的业务应用中，数据量大是其中一个关键特征，那么如何将业务数据（text/csv 格式）快速的装载到数据库内部，对外提供查询分析服务，是会影响 OLAP 领域数据库选择的技术能力和关键竞争力。
- 数据库系统对于数据导入的通常做法是从客户端接收到数据，SQL 层进行数据解析，容错处理，路由分发，事务封装最后再进程存储持久化，这么做在设计上存在如下几个瓶颈点：1) 一行一行数据进行，2) SQL 层过滤解析，3) CLOG 日志量大。因此 OLAP 系统数据库对此都有针对性优化，最常见的做法是通过旁路导入技术，将 SQL 的处理消耗代价屏蔽，使得数据可以直接持久化到磁盘上。OceanBase 数据库 V4.1 版本提供了旁路导入技术对数据加载执行路径进行精简，旁路导入跳过 SQL 层、事务模块、MemTable 等模块，直接将数据持久化到 SSTable，大大的提升了数据导入的速度。使用旁路导入能力需要为 load data 语句增加 HINT 标识，例如：`load data /*+ direct(true , 1024) parallel(64) */ infile '/data/1/hits.tsv' into table hits Fields Terminated By '\t';` 通过视图 `V$SESSION_LONGOPS` 可以观测导入进度，支持多种模式主键冲突处理和过程容错，当前仅支持 CSV 格式数据文件。使用旁路导入在

c6a.12xlarge 规格（48vCPU，96GB 内存）机器上对 LINEITEM 表（74GB）进行数据导入测试，单 OBPProxy 节点导入速率可以达到 165MB/s。

- 基于 OBPProxy 的分布式事务路由优化，提升分布式事务性能。

在业务数据建模时通常要求其事务处理尽可能聚集在一个数据节点上，来避免分布式事务引发的跨节点访问性能下降问题。OceanBase 数据库早期版本仅能支持事务内请求在一个 OBPProxy 节点上完成协同处理，因此 OBPProxy 无法按照 SQL 本身访问数据的位置进行路由，这带来了分布式事务内 SQL 处理延迟增加，导致性能下降。客户业务在性能压测中，很多时候需要通过调整 SQL 顺序去优化性能，投入大量时间精力调优来降低分布式事务路由带来的负面性能影响。

OceanBase 数据库 V4.1 版本针对分布式事务路由进行优化设计，借助 OBPProxy 可将事务运行状态同步到 OBPProxy 节点上，不需要都发给协调者，每条 SQL 语句都可以计算路由，从而可以确保 SQL 请求可以在集群内任意 OBPProxy 节点上执行，从而消除不必要的处理延迟。通过 OBPProxy 的配置项 `enable_distributed_route` 可开启分布式事务路由优化，默认开启。当前不支持 XA 事务的路由优化。

- 编译反馈优化。

通过编译反馈优化技术，在 SYSBENCH 测试模型下验证，性能综合提升约 10-15%。

- 其它性能优化。
 - Nest Loop Join 算子性能优化，多级 Nest Loop Join 算子在特定 Group Rescan 场景下性能提升约 10 倍。
 - Index Skip Scan 执行优化，NDV 算子的计算量超过 50 万行数据时，性能可提升约 10 倍。
 - Truncate Table 并行执行优化，Truncate Table 语句执行效率提升约 10 倍。

稳定性增强

- 扩展支持 `DBMS_LOB` 规格上限。

在 OceanBase 数据库早期的版本中，LOB 数据存储大小限制在 48MB 以内，这对客户使用 LOB 带来了强约束限制。在本次架构升级中，通过存储层将 Lob 宏块的数据拆成多条 Lob Meta 进行存储，取数据的时候再将多条 Lob Meta 中的数据聚合成一个连续 Buffer 返回给 SQL 层处理，这样突破了数据存储大小的限制，通过普通 SQL 使用 LOB 规格扩展达到了 512MB，在 Oracle 模式下通过系统包 `DBMS_LOB` 使用 LOB 规格可扩展到 TB 级别。

高可用增强

- 支持基于归档日志的物理备库。

如何不断提升数据库的高可用能力是 OceanBase 数据库设计之初的目标之一，OceanBase 数据库通过分布式架构技术和多副本数据管理，可以满足大部分场景下业务系统高可用诉求。但是在银行、保险等对数据库高可用有极端追求的客户业务中，单集群的两地三中心在跨城网络通信不佳的情况下依然存在服务不可用风险（即地域级容灾），因此 OceanBase 数据库推出了物理备库的方案来解决跨城地域级容灾问题。

在 OceanBase 数据库早期的版本中，物理备库（V3.x 版本之前称为主备库）采用集群级日志传输的策略，主备集群需要相互感知日志同步状态且存在租户间日志耦合的问题。在 OceanBase 数据库 V4.1 版本中将集群级别粒度的物理备库细化到租户级别的物理备库，主租户与备租户相互不感知实现完成解耦，归档日志同步采用异步同步策略，通过三方介质 NFS/OSS 来完成，主备租户支持日志断点续传，Switchover 和 Failover 切换操作。在租户级的物理备库方案下，OceanBase 数据库的一个集群内即可以存在主租户也可以存在备租户，帮助业务应用获的更均衡的资源利用率。

- 支持更多场景达成 RTO 时间降低到 8s 以内。

OceanBase 数据库 V4.0 版本架构升级针对分布式故障检测、分区选举能力进行优化和设计，使能够在故障恢复时缩短 RTO 的时间 <8s，但是故障场景是复杂和耦合的，在 V4.1 版本中继续识别分析分布式数据库故障场景，并加以优化使得在大多数场景下可以达成 RTO<8s 这个系统设计目标，主要完善优化场景如下：

- 主 OBCServer 节点故障宕机，连同 OBProxy 对外一起恢复服务。
- 主 OBCServer 节点的日志盘故障或数据盘故障，主节点报错但不退出，集群分区 Leader 进行迁移和选主后对外提供服务。此时可用副本数减少，故障的节点仍需要人工干预恢复。

运维能力提升

- 支持 4.0 系列版本在线升级。

OceanBase 数据库 V4.0 版本是大的架构升级版本，其中存储数据、内部表、视图、配置项、部分功能行为均发生了不兼容变化，所以 V3.x 版本无法支持热升级到 V4.0 版本，建议通过 OMS 工具进行数据逻辑迁移进行升级。但是从 V4.1 版本开始，系统数据兼容和支持物理在线升级是 OceanBase 数据库核心关键能力目标，同时也是满足客户项目升级不中断诉求，降低升级复杂度和代价，助力 OceanBase 数据库 V4.1 版本在市场项目上应用和推广。

OceanBase 数据库 V4.1 采用轮转升级的总体策略。

- a. 发起升级操作后，集群将进入混部状态，系统按 Zone 替换 OBCServer 的二进制程序，直至全部节点替换完毕后退出现部状态，此时高版本的 OBCServer 的二进制程序以兼容方式运行低版本的数据和行为。
- b. 在租户内执行升级命令，系统内部进行变量/系统表/数据校验和升级修正。集群内租户需逐一执行同样升级命令，最后完成集群整体升级。推荐使用 OCP 工具完成集群在线升级任务。

OceanBase 数据库 V4.1 支持的升级场景如下：

- 支持集群在线升级。
- 支持主备集群升级，主备集群需各自升级。建议先升级备集群后主集群，升级前建议先通过 Switchover 命令将集群内租户状态调整一致。
- 支持从 V4.0 版本开始的后续版本连续升级，包括功能版本和补丁版本。
- 升级成功的 OBCServer 节点不支持回退，因此建议升级前做好数据和二进制程序备份。
- 支持租户级版本升级。但不推荐同一节点不同租户的版本长期不一致，建议集群内所有租户协同升级到同一版本。

与此同时，升级过程中也存在如下约束限制：

- 禁止 DDL 操作，升级完成后自动开启。

- 禁止 Major Freeze 操作，升级完成后会自动开启。
- 禁止迁移复制和负载均衡，升级完成后会自动开启。
- 禁止物理备份恢复操作，升级完成后会自动开启。
- 禁止 Switchover/Failover 操作，升级完成后会自动开启。
- 禁止新建租户操作，升级完成后会自动开启。
- 日志系统与可读性优化整改。

OceanBase 数据库 V4.1 版本对日志输出进行了系统性分析和优化调整，主要包括：

- 日志级别定义：增加 DIAG 诊断日志级别，将无需用户感知的 WARN、ERROR 归到 DIAG 日志。对需要保留面向运维的预警/报错日志，进行日志级别分类调整，如下：

日志级别	面向对象	定义	原日志级别
ERROR	DBA/用户	OBServer 不能提供正常服务的异常，如磁盘满监听端口被占用等。也可能是产品后台的内部检查报错，例如 4377(dml defensive check error)，4103 (data checksum error) 等，需人工干预恢复。	
WARN	DBA/用户	出现非预期场景，OBServer 能提供服务，但行为可能不符合预期，例如写入限流告警。	
INFO	DBA/用户	系统状态变化信息。	INFO
EDIAG	研发	Error Diagnosis，协助故障排查的诊断信息，非预期的逻辑错误，如函数参数不符合预期等。	ERROR
WDIAG	研发	Warning Diagnosis，协助故障排查的诊断信息，预期内的错误，如函数返回失败。	WARN

TRACE	研发	任务级调试信息。	TRACE
DEBUG	研发	详细调试信息。	DEBUG

- 错误码关联：分类错误消息并完善对应的错误码。为用户的每一条告警、错误定义响应的错误码，用户可根据错误码检索文档寻找解决方法。
- 按错误码限流：针对 WDIAG 级别日志按错误码进行限流，被限流日志仅输出错误码及限流日志数。增加 `diag_syslog_per_error_limit` 配置项用于控制此限流功能，默认单个错误 200 条 DIAG 日志每秒。
- 日志文件自描述：在日志文件头部增加记录机器地址、机型信息 CPU 类型、OS 版本、OBServer 版本和 TimeZone 信息等。
- 视图标准化优化，对视图/内部表进行重新整改设计，统一标准化视图输出共 73 个，推荐用户通过视图获取 OceanBase 数据库元数据等信息，更多信息请参见 文档链接。
- 实时统计信息收集。

统计信息的准确度是优化器生产良好执行计划的基础，OceanBase 数据库早期版本已经支持多种方式生成统计信息：

- 通过系统包 `DBMS_STATS` 或 `ANALYZE` 命令手动触发收集。
- 通过定时任务自动收集统计信息。
- 在数据量变化超过一定阈值时自动收集统计信息。

在 OceanBase 数据库 V4.1 版本中我们新增支持 Online 统计信息收集能力，通过系统变量 `_optimizer_gather_stats_on_load` 或者 HINT 可以启用该能力，当业务支持 CTAS 或者 INSERT 语句时，OceanBase 数据库会实时增量更新统计信息，避免常规统计信息收集时全表扫描所带来的的系统开销。

- 持久化与存储优化。
 - 自适应合并（Compaction）：Compaction 会消耗大量的 IO 操作和 CPU 资源，因此针对不同的业务场景，需要采取不同的 Compaction 策略。在 LSM 存储体系下，Compaction 的核心目标是对这三类因子（写放大、读放大、空间放大）做最优平衡，自适应 Compaction 策略即是通过表的历史转储、查询行为进行统计分析，抽取当前 Tablet 特征，并依此判断该 Tablet 是否需要调度合并，在用户无感知的情况下完成合并。
 - 表空间优化：在小表多分区的场景下，宏块存储会有一定的存储放大现象，在分区数很多时容易出现磁盘空间不够的问题。通过将多个小 SSTable 存入一个物理宏块中进行合并存储，小表多分区场景下优化后宏块使用比例约 4%，宏块碎片比例约为 1.5%，大大提高了磁盘使用率。

产品形态

- 支持单机产品形态。
随着 OceanBase 数据库 V4.0 版本单机分布式一体化架构升级逐步完成，我们也顺利推出 OceanBase 数据库单机产品，支持通过 OCP/OBD 来部署安装单机版本，来帮助 ISV、小客户、社区用户等快速上手体验 OceanBase 数据库 V4.1 版本所带来的新产品能力，同时我们支持单机版本

在线转化升级为分布式版本，在数据管理规模无法满足项目要求或业务需要更高的高可用能力时，可以轻松实现“角色”转换。

性能报告

测试环境规格：

资源	规格
CPU 平台架构	x86_64
ECS 类型	ecs.g7.8xlarge
计算资源	32 核
内存资源	128GB
磁盘资源	500G ESSD 云盘
操作系统	CentOS Linux release 7.9.2009 (Core)

测试版本：

产品	版本信息
OceanBase 数据库 (V4.1)	OBServer (OceanBase_CE 4.1.0.0)

REVISION: 100000172023031416-6d4641c7bcd0462f0bf3faed011803c087ea1705

BUILD_TIME: Mar 14 2023 16:53:58|

|OceanBase 数据库代理 ODP (V4.1) |OBProxy (OceanBase 4.1.0.0 1)

REVISION: 5617-local-e6798c479feaab9f9a60b89f87e4df5e284250b6

BUILD_TIME: Mar 11 2023 21:42:11|

|OceanBase 数据库 (V4.0) |OBServer (OceanBase_CE 4.0.0.0)

REVISION: 103000022023011215-05bbad0279302d7274e1b5ab79323a2c915c1981

BUILD_TIME: Jan 12 2023 15:28:27|

REVISION: 1-local-9d5c90e562c61d9fcf8894993187aa20239db47e

BUILD_TIME: Oct 28 2022 23:01:19|

SYSBENCH OLTP 负载测试

测试方案：

1. 通过 OBD 部署 OceanBase 集群，OBProxy 和 Sysbench 分别安装在两台机器上，作为客户端的压力机器，避免资源抢占。其中 OBProxy 单独部署在 64c128g 的机器上（机型 ecs.c7.16xlarge）。
2. OceanBase 集群规模为 1: 1: 1，部署成功后先新建跑 Sysbench 测试的租户及用户（sys 租户是管理集群的内置系统租户，请勿直接使用 sys 租户进行测试），设置租户的 Primary Zone 为 RANDOM，RANDOM 表示新建表分区的 leader 随机到这 3 台机器。
3. 启动 Sysbench 客户端，进行 point_select、read_write、read_only 和 write_only 测试。
4. 每轮测试 `--time` 设置为 60s，线程数取值可以为 32/64/128/256/512/1024。
5. 测试数据量为 30 张非分区表，100 万行数据，即 `--tables=30`，`--table_size` 设置为 1000000。

租户规格：

- MAX_CPU = 26
- MEMORY_SIZE = 70g

测试结果：

Point Select 性能

Threads	V4.0 QPS	V4.0 95% Latency (ms)	V4.1 QPS	V4.1 95% Latency (ms)
32	139594.87	0.26	135146.05	0.27
64	257088.19	0.29	251219.37	0.30
128	453625.18	0.34	431564.13	0.36
256	746710.21	0.50	686271.21	0.55
512	964910.06	0.81	920502.51	0.92

1024	988444.97	1.64	972018.58	1.76
------	-----------	------	-----------	------

Read Only 性能

Threads	V4.0 QPS	V4.0 95% Latency (ms)	V4.1 QPS	V4.1 95% Latency (ms)
32	100505.96	6.43	117325.66	4.65
64	189356.32	6.55	214733.98	5.18
128	326115.21	7.43	370499.19	6.09
256	479483.89	10.46	572924.33	8.13
512	584193.84	19.29	705032.91	13.95
1024	561898.43	70.55	755182.87	28.16

Write Only 性能

Threads	V4.0 QPS	V4.0 95% Latency (ms)	V4.1 QPS	V4.1 95% Latency (ms)
32	38301.16	6.43	42329.95	5.37
64	71365.18	6.91	74408.79	6.09
128	117430.28	8.13	126026.90	7.30
256	182082.02	10.84	197493.43	9.56
512	252286.87	16.71	288284.95	13.95
1024	290982.27	31.37	354554.10	25.74

Read Write 性能

Threads	V4.0 QPS	V4.0 95% Latency (ms)	V4.1 QPS	V4.1 95% Latency (ms)
32	63498.13	12.30	74857.82	9.39
64	120362.58	12.52	138795.77	10.27

128	197802.61	15.55	234340.23	12.08
256	303103.47	20.37	365657.91	16.12
512	383006.31	33.12	479109.44	25.74
1024	389587.53	104.84	561333.10	48.34

BMSQL TPCC 测试

测试方案：

1. 通过 OBD 部署 OceanBase 集群，OBProxy 和 TPC-C 单独部署在一台机器上，防止客户端的压力不足成为性能瓶颈。
2. 3 节点的 OceanBase 集群部署规模为 1: 1: 1，部署成功后先新建跑 TPC-C 测试的租户及用户（sys 租户是管理集群的内置系统租户，请勿直接使用 sys 租户进行测试），设置租户的 Primary Zone 为 RANDOM。

租户规格：

- MAX_CPU = 26
- MEMORY_SIZE = 70g

软件版本：

mysql-connector-java-5.1.47

Benchmark SQL V5.0

测试配置：

- warehouse = 1000
- loadWorder=40
- terminals=800
- runMins=5
- newOrderWeight=45
- paymentWeight=43
- orderStatusWeight=4
- deliveryWeight=4

- stockLevelWeight=4

测试结果：

	V4.0	V4.1
tpmC (NewOrders)	307021.0	332559.26
tpmTOTAL	682517.67	738374.78

TPCH 测试

测试方案：

1. 使用 OBD 部署 OceanBase 数据库集群。TPC-H 客户端需要部署在一台机器上，作为客户端的压力机器。您无需部署 OBProxy，测试时直连任意一台机器即可。
2. 3 节点的 OceanBase 集群部署规模为 1: 1: 1，部署成功后先新建跑 TPCH 测试的租户及用户（sys 租户是管理集群的内置系统租户，请勿直接使用 sys 租户进行测试），设置租户的 Primary Zone 为 RANDOM。先执行数据装载，然后多次顺序执行 22 个 SQL 后取均值。
3. 测试数据量：100GB。

租户规格：

- MAX_CPU = 26
- MEMORY_SIZE = 70g

测试结果：

Query	V4.0 (s)	V4.1 (s)
Q1	2.34	2.06
Q2	0.14	0.22
Q3	0.72	1.50
Q4	0.56	0.46
Q5	2.25	0.95
Q6	0.23	0.13
Q7	1.52	1.48

Q8	0.70	0.61
Q9	5.22	2.95
Q10	1.24	1.02
Q11	0.23	0.18
Q12	1.62	1.13
Q13	2.41	1.95
Q14	0.36	0.28
Q15	0.79	0.88
Q16	0.66	0.62
Q17	0.63	0.54
Q18	0.93	0.83
Q19	0.78	0.61
Q20	1.17	1.09
Q21	2.42	2.63
Q22	1.24	1.09
Total	28.16	23.21

TPC-DS 测试

测试方案：

1. 使用 OBD 部署 OceanBase 数据库集群。客户端需要部署在一台机器上，作为客户端的压力机器。您无需部署 OBProxy，测试时直连任意一台机器即可。
2. 3 节点的 OceanBase 集群部署规模为 1: 1: 1，部署成功后先新建跑 TPCDS 测试的租户及用户（sys 租户是管理集群的内置系统租户，请勿直接使用 sys 租户进行测试），设置租户的 Primary Zone 为 RANDOM。先执行数据装载，然后多次顺序执行 99 个 SQL 后取均值。
3. 测试数据量：100GB。

租户信息：

- MAX_CPU = 26

- MEMORY_SIZE = 70g

测试结果：

Query	V4.1 (s)
Q1	0.63
Q2	3.46
Q3	0.16
Q4	9.43
Q5	1.63
Q6	0.37
Q7	0.28
Q8	0.39
Q9	1.89
Q10	0.50
Q11	5.79
Q12	0.28
Q13	2.04
Q14	11.38
Q15	0.23
Q16	0.92
Q17	0.52
Q18	0.32
Q19	0.17
Q20	0.19
Q21	0.29
Q22	1.11
Q23	14.20

Q24	1.56
Q25	0.62
Q26	0.21
Q27	0.35
Q28	7.44
Q29	0.64
Q30	0.27
Q31	0.62
Q32	0.11
Q33	0.77
Q34	0.37
Q35	0.87
Q36	0.31
Q37	0.42
Q38	1.49
Q39	2.31
Q40	0.16
Q41	0.18
Q42	0.11
Q43	0.64
Q44	0.45
Q45	0.39
Q46	0.81
Q47	1.16
Q48	0.61
Q49	0.86
Q50	0.81

Q51	2.82
Q52	0.11
Q53	0.24
Q54	1.05
Q55	0.11
Q56	1.01
Q57	0.79
Q58	0.60
Q59	13.64
Q60	1.13
Q61	0.30
Q62	0.40
Q63	0.23
Q64	1.84
Q65	2.70
Q66	0.46
Q67	16.12
Q68	0.67
Q69	0.44
Q70	1.18
Q71	0.54
Q72	1.47
Q73	0.31
Q74	6.36
Q75	2.26
Q76	0.49
Q77	0.49

Q78	3.64
Q79	0.65
Q80	1.78
Q81	0.30
Q82	0.67
Q83	0.69
Q84	0.19
Q85	0.84
Q86	0.36
Q87	1.59
Q88	0.65
Q89	0.30
Q90	0.19
Q91	0.13
Q92	0.10
Q93	0.75
Q94	0.62
Q95	6.37
Q96	0.38
Q97	0.93
Q98	0.30
Q99	0.70
Total Time	160.83

兼容性变更

产品行为变更

功能	变更版本	变更点说明
浮点类型	4.1	V4.0: 不再支持 float(m, d) 格式定义, 仅支持 float(m) 和 float(m, 0) 的格式定义。 V4.1: 完全兼容 MySQL 8.0 数据类型与精度
备份文件名后缀	4.1	新增后缀 <code>.obbak</code> 标识 OceanBase 数据库的数据备份的文件。 新增后缀 <code>.obarc</code> 标识 OceanBase 数据库的日志归档的文件
UNUSABLE索引	4.1	删除分区后产生的 UNUSABLE 索引, 通过备份恢复操作不会再恢复出来。
MAJOR FREEZE	4.1	新增支持指定一个租户和特定 Tablet 或特定表, 发起合并任务。

视图/实体表/虚拟表

视图/实体表/虚拟表	用途及不兼容点说明	状态变化	变更版本
<code>__all_virtual_plan_table</code>	保存当前 session explain 的计划。	新增	4.1
<code>__all_virtual_sql_plan</code>	保存用户执行过的逻辑计划。	新增	4.1

<code>__all_virtual_h a_diagnose</code>	在诊断表中额外加入对于 gc 当前状态的诊断信息（ <code>gc_diagnose_info</code> 、 <code>restore_handler_role</code> 、 <code>restore_handler_proposal_id</code> 、 <code>restore_context_info</code> 、 <code>restore_err_context_info</code> ）。	变更	4.1
<code>__all_virtual_m alloc_sample_info</code>	查询各模块内存分配的堆栈信息。	新增	4.1
<code>__all_ddl_task_ status</code>	增加 <code>execution_id</code> 列来标记 DDL 的每次单副本构建任务。	变更	4.1
<code>__all_virtual_s pan_info</code>	存放诊断信息。	新增	4.1
<code>__all_virtual_s how_trace</code>	展示处理后的诊断信息。	新增	4.1
<code>__all_virtual_i o_scheduler</code> 展示 IOPS 与带宽的统计信息。	新增	4.1	
<code>__all_virtual_i o_status</code>	展示不同租户、不同类别的请求 IO 排队情况。	新增	4.1
<code>V\$OB_SQL_AUDIT</code>	新增加 <code>PARTITION_HIT</code> 字段，用于判断是否会发生分区路由，是否是远程执行。	变更	4.1

<code>DBA_RSRC_PLAN_DIRECTIVES</code>	展示 resource manager 中的 IO 资源配置情况	新增	4.1
<code>V\$OB_TABLET_STATS</code> 自适应 Compaction 的 Tablet 级别的采集信息，有助于问题排查和定位。	新增	4.1	
<code>DBA_OB_TENANTS</code>	扩展普通支持普通租户可查看，增加展示例如 LOCALITY、同步位点、租户类型等信息 (<code>TENANT_ROLE</code> 、 <code>SWITCHOVER_STATUS</code> 、 <code>SWITCHOVER_EPOCH</code> 、 <code>SYNC_SCN</code> 、 <code>REPLAYABLE_SCN</code> 、 <code>READABLE_SCN</code> 、 <code>RECOVERY_UNTIL_SCN</code> 、 <code>LOG_MODE</code> 、 <code>ARBITRATION_SERVICE_STATUS</code>)。	变更	4.1
<code>V\$OB_LOG_STAT</code>	新增列 <code>arbitration_member</code> 、 <code>degraded_list</code> ，表示仲裁成员和降级副本列表。	变更	4.1
<code>V\$OB_TRANSACTION_SCHEDULERS</code>	展示事务的 scheduler 信息，有助于问题排查和定位。	新增	4.1

<code>DBA_OB_ARBITRATION_SERVICE</code>	展示当前集群的仲裁服务配置信息。	新增	4.1
<code>DBA_OB_LS_ARB_REPLICA_TASKS</code>	展示当前租户及对应用户租户正在运行中的仲裁副本任务。	新增	4.1
<code>DBA_OB_LS_ARB_REPLICA_TASK_HISTORY</code>	展示当前租户及对应用户租户执行过的仲裁副本任务历史。	新增	4.1
<code>V\$OB_ARCHIVE_DEST_STATUS</code>	查询租户各个归档目的端状态。	新增	4.1
<code>DBA_OB_LS_LOG_ARCHIVE_PROGRESS</code>	展示日志流级别归档进度。	新增	4.1
<code>DBA_OB_LS_LOG_RESTORE_STAT</code>	展示日志流级别恢复汇总信息。	新增	4.1
<code>DBA_OB_CLUSTER_EVENT_HISTORY</code>	仅展示集群升级相关事件，仅系统租户可访问。	新增	4.1
<code>DBA_OB_TENANTS</code>	展示所有租户的基本信息。	变更	4.1

配置项变更

配置项	用途及不兼容点说明	状态变化	变更版本
<code>ob_proxy_readonly_transaction_routing_policy</code>	只读语句路由策略。	废弃	4.1
<code>ob_plan_table_memory_limit</code>	控制优化器保存逻辑计划使用的内存上限。	默认值 32MB，取值范围：1 – 1024。	4.1、
<code>tenant_task_queue_size</code>	租户请求队列长度。	默认值 65536 调整为 8192。	4.1
<code>ob_enable_trace_log</code>	是否启用 SQL Trace 功能。	废弃	4.1

<code>ob_enable_show_trace</code>	是否启用 Show Trace 功能。	默认值 0，关闭 Show Trace 功能。	4.1
<code>_data_storage_io_timeout</code>	配置数据盘最大 IO 超时时间。	默认值 120s 调整为 10s，取值下线 5s 调整为 1s。	4.1
<code>_enable_pkt_nio</code>	控制新的 RPC 网络框架 pkt-nio 的启用和关闭。	默认值 True。	4.1
<code>rpc_memory_limit_percentage</code>	租户下的 RPC 可以使用的最大内存。	默认值 0，表示不对 RPC 内存进行限制。	4.1
<code>_private_buffer_size</code>	配置日志数据达到 buffer size 时触发写日志。	默认值 256KB 调整为 16KB。	4.1
<code>_mvcc_gc_using_min_txn_snapshot</code>	是否用户使用活跃事务快照来保留多版本数据。	默认值 True。	4.1
<code>_ob_enable_dynamic_worker</code>	控制自适应线程功能的开关。	默认值 True。	4.1
<code>data_storage_warning_tolerance_time</code>	容忍数据盘 IO 失败的时间，超出则认为磁盘损坏。	默认值 5s，取值范围：1 – 300。	4.1
<code>log_storage_warning_tolerance_time</code>	容忍日志盘 IO 失败的时间，超出则认为磁盘损坏。	默认值 5s，取值范围：1 – 300。	4.1
<code>_min_malloc_sample_interval</code>	控制内存分配的最小采样间隔。	默认值 16，取值范围：1 – 10000。	4.1
<code>_max_malloc_sample_interval</code>	控制内存分配的最大采样间隔。	默认值 256，取值范围：1 – 10000。	4.1

<code>ob_max_read_stale_time</code>	Session 级别控制弱读延迟阈值, 当前 Session 上 weak 读请求, 读到的数据一定在设定的延迟阈值内, 不同 Session 可以设置为不同的延迟阈值。	默认值 5s。	4.1
<code>log_archive_currency</code>	控制日志归档总的工作线程数量, 支持动态增加、减少线程数量。	生效范围由集群级变更为租户级。	4.1
<code>log_restore_currency</code>	控制物理恢复/备库恢复日志的工作线程数量, 支持动态增加、减少线程数量。	生效范围由集群级变更为租户级。	4.1
<code>_ob_enable_direct_load</code>	是否开启旁路导入功能。	默认值 True。	4.1
<code>_enable_transaction_internal_routing</code>	事务内执行的 DML 语句是否可路由至任意 OBServer。	默认值 True。	4.1
<code>log_restore_source</code>	日志恢复路径。	默认值“”。	4.1
<code>ob_startup_mode</code>	observer 的启动模式。	默认值为 <code>normal</code> , 指定为 <code>arbitration</code> 时表示以仲裁模式启动。	4.1
<code>_ob_plan_cache_auto_flush_interval</code>	控制定时自动 flush plan cache 的时间间隔。	默认值 0s, 代表不进行自动 flush。	4.1
<code>enable_user_defined_rewrite_rules</code>	控制用户自定义规则是否开启的开关。	默认值 False。	4.1

升级说明

- 不支持从 V3.x 版本在线升级到 V4.x 版本。
- 支持通过 OMS 将 V3.x 版本数据使用逻辑迁移升级到 V4.1 版本。
- 支持从 V4.0 版本在线升级到 V4.1 版本。

周边配套

OceanBase 数据库 V4.1.0 推荐使用的平台工具版本如下。

组件	配套版本
OceanBase Database Proxy (ODP)	V4.1.0
OceanBase Connector/J	V2.4.3
OceanBase Connecotr/ODBC	V2.0.7
OceanBase Client (OBClient)	V2.2.2
OceanBase Deployer (OBD)	V2.0.0
OceanBase Agent (OBAgent)	V1.3.0
ob-operator	V1.2.0
MySQL JDBC	V5.1.47

约束限制

- OceanBase 数据库 V4.1.0 暂时不支持副本扩容后的数据负载均衡，扩容前已经存在的数据无法均衡到新添加的节点上，计划在 V4.2 版本完善负载均衡能力。
- OceanBase 数据库 V4.1.0 暂时不支持复制表，后续 V4.2 版本将补齐该能力。
- OceanBase 数据库 V4.1.0 支持从主集群发起数据备份，后续版本将支持从备集群发起数据备份。
- OceanBase 数据库 V4.1.0 节点在生产环境最低要求 运行规格为 4C16GB，租户内存资源大于 2GB，CPU 资源大于 1C。
- OceanBase 数据库 V4.1.0 当前仅支持 F 副本，其它类型副本如 R 副本将在后续版本补齐。
- OceanBase 数据库 V4.1.0 当前仅支持调大租户的 Unit 个数（`UNIT_NUM`），后续版本将支持调小租户的 Unit 个数。
- OceanBase 数据库 V4.1.0 支持从 V4.0 版本在线升级到 V4.1 版本，OBD 暂未适配完成，后续版本会补齐该能力。
- OceanBase 数据库 V4.1.0 版本 `DROP DATABASE` 操作一次删除数据库内的表的个数超过 200

个时存在删除失败的风险，建议业务先 `DROP TABLE` 后再执行 `DROP DATABASE` 操作，后续 V4.2 版本将修复该问题。