OCEANBASE

OceanBase 数据库

版本发布记录

| 产品版本: V4.0.0

| 文档版本: 20221102

声明

蚂蚁集团版权所有©2020，并保留一切权利。

未经蚂蚁集团事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

商标声明

及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

免责声明

由于产品版本升级、调整或其他原因，本文档内容有可能变更。蚂蚁集团保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在蚂蚁集团授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击 设置> 网络> 设置网络类型 。
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面，单击 确定 。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

OceanBase 数据库社区版 V4.0.0

OceanBase 社区发布 V4.0.0 版本是对分布式数据库系统架构设计的全面升级，定位为 Beta 测试版本，社区会关注用户使用反馈，不断对产品进行打磨和完善，应用项目投产上线需要慎重审视和仔细评估。

OceanBase V4.0 版本在保证功能特性不丢失的前提下，重新审视了数据库与分布式系统两个领域最基础的设计，全新推出业内首个单机分布式一体化架构，重点构建 HTAP 和云化两个基础能力属性。与此同时，本版本也从架构上解决了 V3.2 版本的设计瓶颈，支持更多用户业务关注的多个核心能力，在内核功能、兼容性、稳定性、性能上取得突破。

版本概要

本版本的核心能力特性如下：

- 架构升级与受益

支持单机分布式一体化架构，包含自适应日志流、支持超大事务、RTO 时间降低到 8s 以内、NTP 服务依赖优化、支持分区数量能力上限等版本基础核心能力构建。

- 内核能力增强

Online DDL 能力增强，支持租户级备份，字符集扩展，支持数据编码，支持 IOPS 隔离，LOB 规格上限扩展，支持表锁和死锁检测等。

- 兼容性增强

支持 DDL 语句的外键约束，支持视图列信息展示，支持 DML 触发器，支持更多 SQL MODE 和函数等。扩展支持 SEQUENCE 对象，支持存储程序，支持 SQL 文本中的预处理，支持自增列做为分区键。

- 性能大幅提升

- SYSBENCH 性能优化，综合读写性能（Read Write）1024 并发测试性能相比于 V3.1 版本提升 1 倍。

- TPCB 查询性能优化，100GB 数据量顺序执行 22 条 SQL，整体性能相比于 V3.1 版本提升 5 倍。

- 运维能力提升

支持全链路追踪，支持 SESSION 状态的监控和诊断（ASH），标准化视图优化，支持 Schema History 回收功能，支持自动清空回收站功能等。

特性更新架构

升级与收益

- 自适应日志流介绍

OceanBase 早期版本的架构体系里以分区为基本单元进行操作，当系统内的分区数量达到一定程度之后，以分区为单元的操作的消耗也随之增大，逐渐形成了 OceanBase 的使用痛点：单节点支持的分区数量受到限制，单节点上涉及跨分区的数据修改也需要两阶段提交协议来保证事务的原子性等问题。

日志流是由 OceanBase 自动创建和管理的实体，它代表了一批数据的集合，包括若干分区和有序的 Redo 日志流。在新的系统架构下，一个 Unit 内的所有分区的事务修改日志可以都记录在一个日志流中，通过日志流把修改同步到其他 Zone 的对应 Unit 上。OceanBase 的每个租户每有一个 Unit，就会有一个对应的日志流。系统会把一个日志流和其所对应的分区关系固定下来，只有迁移发生时，才会改变这个对应关系。

- 支持超大大事务

基于新的自适应日志流架构，对事务引擎进行重新设计，解决了分布式数据库普遍的大事务场景使用痛点，比如事务大、参与者数量多、事务提交卡等问题。新事务引擎能稳定应对在线业务、批处理、订正数据等多种业务场景，使得用户在各自繁杂的业务场景下可以放心的使用数据库。

- RTO < 8s

基于全新的自动选主协议以及全面的探活机制，进一步将机器故障场景下系统恢复时间降低到 8s 以内，帮助业务系统更快恢复，最大程度减少业务影响，给业务带来持续可用的能力。

- NTP 服务优化

基于全新的自动选主协议，取消了对 NTP 时钟的依赖，打破原来早期版本对所有节点的时钟偏差控制在 100ms 以内的强需求。OceanBase V4.0 版本允许的时钟偏差可以达到 2s，同时支持动态修改时钟，不会对数据正确性和集群稳定运行带来影响。

- 优化分区能力上限

在自适应日志流的架构设计下，系统内部一个 Unit 内的所有分区共享一个资源组，大大降低了早期版本每个分区独立申请保存系统资源，提升系统资源的利用率，因此不再需要根据配置限制 OBServer 节点的分区上限个数，但分区上限仍受机器可用物理资源限制。

内核能力增强

- Online DDL 能力增强

在 OceanBase 早期的版本中，由于架构设计上的限制，对数据库 Online DDL 能力进行了有限支持，例如不支持主键修改操作给业务使用带来了诸多不便。得益于新版本一体化的架构设计，OceanBase 针对涉及到数据搬迁的 Online DDL 操作进行增强支持，主要包括：

- 支持添加主键、修改主键和删除主键
- 支持添加外键、删除外键
- 支持修改列名和列类型，包括字符、数值、日期等类型的相互转换
- 支持普通表转换为分区表

- 支持在线修改表或列的字符序

- 支持租户级备份

多租户是 OceanBase 的核心价值能力之一，在大多数客户系统中，用户都选择在同一个集群中创建了多个租户，每个租户代表一个业务单元，根据业务的不同种类和对客户的重要程度，需要有不同的备份频率和策略进行细粒度支持。在 OceanBase V4.0 版本中将集群级别粒度的备份能力细化拆分到租户级别粒度，支持按租户级别的备份，也支持将备份数据恢复到新租户。优化数据备份快照保留策略，减少备份期间的磁盘空间影响。同时拆分数据备份与日志备份存储目录，支持分别接入不同性能的备份介质。

有关租户级备份的更多详细信息，参见 [备份恢复](#)

- 扩展更多字符集

支持 UTF8、UTF16、GBK、GB18030 和 BINARY 字符集，新增支持 UTF8MB4_BIN、UTF16_BIN、GBK_BIN、GB18030_BIN、UTF8MB4_GENERAL_CI、UTF16_GENERAL_CI、GBK_CHINESE_CI和GB18030_CHINESE_CI 排序规则。

- 向量化引擎开源

OceanBase 在商业版 V3.2.3 全面实现了向量化引擎，以 Architecture aware 的设计改造了全部的算子和绝大部分常用的执行表达式，充分发掘现代 CPU 的 cache 特性以及优化指令，并应用于 TPC-H 的 benchmark 中。向量化带来了大量的算法优化可能，通过在向量化的框架下进行算法和数据结构优化，实测整体执行性能相比原先非向量化执行引擎性能提升普遍在 4-5 倍，很多算子和单场景可获得 10 倍以上的性能提升。在本次版本发布中，OceanBase 将其向量化引擎能力全部开源，帮助用户在 OLAP 场景下获取更好的性能。

- 数据编码开源

OceanBase 通过数据编码压缩技术实现了数据的高压缩比，是帮助用户减小存储成本重要技术手段。OceanBase 本次开源多种数据编码方法，包括字典编码、RLE 编码、常量编码、差值编码、前缀编码、列间编码等，并支持每一列自动选择最合适的数据编码。通过编码和压缩，使用相同的块大小（16KB）、以及相同的压缩算法（lz4），同样的数据存放在 OceanBase 中，要比在 MySQL 5.7 中平均节省一半的空间，同时没有损失任何查询性能。

有关数据编码的更多详细信息，参见 [数据压缩与编码](#)

- 支持 IOPS 隔离

OceanBase 早期的版本已经支持了租户级别的 CPU 和内存隔离，V4.0 版本开始支持租户间 IOPS 隔离，租户之间彼此感知不到对方对磁盘带宽的占用，避免租户间业务的 IO 资源争抢，实现完备的租户资源隔离能力。

用户通过 UNIT CONFIG 设置 UNIT 的资源规格，其中 MIN_IOPS、MAX_IOPS、IOPS_WEIGHT 是 IOPS 隔离相关资源，租户的可用资源与 UNIT 绑定。通知支持动态调整租户的 IOPS 规格，修改 UNIT CONFIG 中的 IOPS 相关设置即可实时生效。在租户内部，支持通过配置项 io_category_config 分配各类别 IO（业务请求、系统日志等）请求的百分比，进而细粒度控制 IO 资源分配与调度。

- LOB 规格上限扩展

在 OceanBase 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，使得 LOB 存储上限扩展达到了 512MB，后续将持续优化到 TB 级别。

- 支持表锁和死锁检测

表锁允许业务以指定的方式锁定表或分区，避免业务并发操作造成数据破坏。在 OceanBase V4.0 版本中支持了 Online DDL 操作，因此必须配套表锁保护 DDL 与 DML 并发时的正确性问题。新增支持 LOCK TABLE 语法，支持 SHARE 和 EXCLUSIVE 两种锁定模式，同时支持对锁冲突的死锁检测。

MySQL 兼容性增强

- 支持 DDL 语句的外键约束

OceanBase 数据库默认开启外键约束检查，外键约束检查开关由租户变量 FOREIGN_KEY_CHECKS 来控制，要求约束的列的值取自于另外一个表的主键列。在早期的版本中，外键约束检查仅对 DML 操作有效，DDL 操作不受影响。OceanBase V4.0 版本中支持了 FOREIGN_KEY_CHECKS 系统变量对 DDL 部分的影响，其行为保持与 MySQL 兼容。

- 支持视图列信息展示

在 MySQL 数据库中，视图列信息和表列信息一样，被作为基础的元信息被存储在数据字典中，并通过 INFORMATION_SCHEMA.COLUMNS 显示给用户。OceanBase 数据库内部仅对表列信息进行了持久化存储，V4.0 版本通过采用动态解析视图定义的方法，避免了对视图复杂的依赖关系解析，实现在 INFORMATION_SCHEMA.COLUMNS 中展示视图列信息。

- SQL Mode 扩展支持

扩展支持 MySQL 默认开启的 SQL Mode，新增支持 NO_ZERO_DATE、ERROR_FOR_DIVISION_BY_ZERO 和 NO_AUTO_CREATE_USER 三个 SQL Mode，产品行为与 MySQL 兼容。针对 NO_ENGINE_SUBSTITUTION 仅支持语法兼容。

- 支持 SQL 文本中的预处理

支持 SQL 文本中的预处理语句（Prepared Statements），Prepared Statements 接口使用二进制协议相比交互式 SQL 接口具有更高的执行效率，使用方法大体如下：

- PREPARE 准备执行语句。 `PREPARE stmt1 FROM 'SELECT SQRT(POW(?,2) + POW(?,2))';`
- EXECUTE 执行准备好的语句。 `SET @a = 3; SET @b = 4; EXECUTE stmt1 USING @a, @b;;`
- DEALLOCATE PREPARE 释放一个准备好的语句。 `DEALLOCATE PREPARE stmt1;`

- 支持自增列做为分区键

例如：

```
create table t2(inv_id bigint not null auto_increment ,c1 bigint, primary key (inv_id)
) partition by hash(inv_id) partitions 8;
```

使用自增列作为分区键时需要额外注意，自增列的值全局唯一，但在分区内不保证始终增长，和原生 MySQL 行为不同。和其他分区方式相比，对这类分区表的插入操作性能会有一定的下降。

- 支持 SEQUENCE 对象

扩展支持 SEQUENCE 对象，满足业务系统对 SEQUENCE 对象的依赖诉求，降低客户在业务迁移过程中的适配复杂度。支持 CREATE/ALTER/DROP SEQUENCE 对象，支持获取 CURRVAL、NEXTVAL 和重置取值等对象操作，支持的对象取值范围从 INT64_MIN 到 INT64_MAX。

- 支持存储程序

支持兼容 MySQL 5.7 语法的存储程序（包含存储过程和存储函数），支持游标、流程控制语句、异常处理、存储程序相关的DDL操作和视图和状态查询。同时扩展支持多个系统包，例如 DBMS_STATS、DBMS_SESSION 等。

- 支持 DML 触发器

支持在表上创建触发器，兼容 MySQL 5.6 语法。当在该表上 DML 操作满足条件时、触发用户自定义行为。

- 支持更多函数

- 支持函数 ADDTIME()，将指定的时间间隔添加到给定的日期和时间。例如：

```
SELECT ADDTIME('2007-12-31 23:59:59.999999', '1 1:1:1.000002');
```
- 支持函数 DAYNAME()，返回给定日期的工作日名称。例如：

```
SELECT DAYNAME('2018-01-8');
```
- 支持聚合函数 BIT_AND()/BIT_OR()/BIT_XOR()，返回表达式的按位与/或/异或的运算结果。
- 新增函数 UUID_SHORT()，返回 64-bit 无符号整数。例如：

```
SELECT UUID_SHORT();
```

运维能力提升

- 支持全链路追踪

OceanBase 作为一款久经沙场的分布式数据库，其内部数据访问的链路已经非常复杂，当线上出现超时等问题的时候，往往无法有效快速定位问题出现的第一现场，需要依靠有经验的运维人员对追个环节进行排查，验证影响到运维效率和故障影响速度。OceanBase V4.0

版本设计了一套全链路追踪的机制，能够提升全链路问题定位的效率，贯穿从业务 APP > 客户端驱动(JDBC, OCI) > 代理 (OBProxy) > 数据库节点 (OBServer) 到全部流程，用户通过 PL/SQL 或 OBClient 接口在应用程序 APP 中设置相关标识信息 (MODULE/ACTION/CLIENT_INFO/CLIENT_IDENTIFIER) 到正在使用的链接中，运维人员可以通过 PL/SQL 程序包，控制相关应用程序设置的标识信息维度是否打开全链路跟踪诊断以及设置诊断输出策略；诊断日志以 OpenTracing 数据模型输出到 OBProxy 和 OBServer 日志文件中进行保存，通过对诊断日志解析，即可达到追踪每个 SQL 每个事务在全链路中执行耗时等相关诊断信息。

有关全链路追踪的详细介绍，参见 [全链路追踪](#)

- 支持 SESSION 状态的监控和诊断 (ASH)

OceanBase 数据库早期版本已经支持获取当前正在执行的 SQL 的状态信息，包括等待事件等信息，但只能通过 `__all_virtual_session_wait` 查询到 session 最近一次等待事件。OceanBase V4.0 版本支持更全面的 session 与等待事件之间的关系图 (ASH)，不仅包含当前执行的 SQL 的状态，还包含 SESSION、USER、SQL、WaitEvent 等多个维度状态历史信息。ASH 能够以 1s 为周期采样系统中所有 Active Session 状态，采用过程中全程不加锁不影响业务 SQL 的正常执行。OCP 通过对这些状态采样信息进行综合汇总分析，帮助用户了解过去一段时间里系统的负载以及等待状态，及时发现并预警问题。

- 支持实时执行计划监控

OceanBase 在 SQL 执行计划的诊断中引入了 SQL Plan Monitor, SQL Plan Monitor 提供了关于逻辑执行计划、物理执行计划、算子吐行数、算子开始/结束时间点等信息，SQL Plan Monitor 信息只能在 SQL 执行完毕后获得，可以通过 `GV$SQL_PLAN_MONITOR` 租户级视图获取相关信息。OceanBase V4.0 版本支持实时 SQL Plan Monitor，可以实时的查看当前租户各 SQL 执行计划状态，在客户实际业务场景或者是诊断分析场景，如果存在 SQL 卡住的问题，此时通过实时 SQL Plan Monitor 也可以查询各个执行线程执行算子的执行状态。

- 支持 Schema History 回收功能

解决 Schema History 只增不删导致 Schema History 过多，影响 OBServer 启动慢的问题。通过隐藏配置项 `_schema_history_recycle_interval` 控制 Schema History 回收周期，该配置项缺省值是 0，表示关闭 schema history 回收功能。

- 支持自动清空回收站功能

OceanBase 数据库回收站提供以租户为单位，当磁盘空闲空间不足时，按照 FIFO 策略，自动清理回收站空间的功能。增加配置项 `recyclebin_object_expire_time` 用于指定回收站中对象的过期时间。

性能报告

测试环境规格：

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
OBServer (V4.0.0_CE)	observer (OceanBase_CE 4.0.0.0) REVISION: 100000152022102115-06570c6f755c1fabe3ca9a25bef54af77e7c8c9c BUILD_TIME: Oct 21 2022 17:22:38
OBProxy (V4.0.0)	obproxy (OceanBase 4.0.0 2) REVISION: 1-local-b1ef5a6b5b196916ae26bde a814765e5165a801f BUILD_TIME: Oct 22 2022 11:06:04

SYSBENCH OLTP 负载测试

测试方案：

1. 通过 OBD 部署 OceanBase 集群，ODP 和 Sysbench 单独部署在一台机器上，作为客户端的压力机器。
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	V4.0.0_CE QPS	V4.0.0_CE		V3.1.0_CE QPS	V3.1.0_CE	
		95%	Latency (ms)		95%	Latency (ms)
32	180360.71	0.21		139124.75	0.28	
64	312254.88	0.26		232522.69	0.36	
128	473423.04	0.41		322185.31	0.63	
256	571193.03	0.89		362650.60	1.58	
512	604975.51	2.97		387072.57	3.96	
1024	614351.07	4.33		401404.45	7.84	

Read Only 性能

Threads	V4.0.0_CE QPS	V4.0.0_CE		V3.1.0_CE QPS	V3.1.0_CE	
		95%	Latency (ms)		95%	Latency (ms)
32	124009.84	5.47		105416.23	6.28	
64	222034.29	5.88		138571.41	9.22	
128	355395.57	7.04		205698.25	13.22	
256	453947.58	12.30		253858.84	23.10	
512	524999.55	20.74		286324.43	42.61	
1024	510261.30	70.55		279067.64	123.28	

Write Only 性能

Threads	V4.0.0_CE QPS	V4.0.0_CE		V3.1.0_CE QPS	V3.1.0_CE	
		95% Latency (ms)	95% Latency (ms)		95% Latency (ms)	95% Latency (ms)
32	37798.74	6.43		35737.28	7.04	
64	72534.26	6.67		57660.74	9.56	
128	125263.17	7.70		81073.80	14.73	
256	188289.15	10.84		98663.59	25.74	
512	239281.86	18.95		111863.72	44.98	
1024	285313.68	34.95		119307.33	89.16	

Read Write 性能

Threads	V4.0.0_CE QPS	V4.0.0_CE		V3.1.0_CE QPS	V3.1.0_CE	
		95% Latency (ms)	95% Latency (ms)		95% Latency (ms)	95% Latency (ms)
32	68305.00	11.65		64412.17	12.08	
64	123581.29	12.98		89716.44	17.95	
128	203527.24	16.71		106215.37	32.53	
256	276437.90	25.74		114100.12	66.84	
512	319334.46	48.34		157859.02	99.33	
1024	314807.75	147.61		146540.16	225.98	

BMSQL TPCC测试

测试方案:

1. 通过 OBD 部署 OceanBase 集群，ODP 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：

指标	V4.0.0_CE	V3.1.0_CE
tpmC (NewOrders)	307021.0	295855.92
tpmTOTAL	682517.67	657398.9

TPCH 测试

测试方案：

- 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 ODP，测试时直连任意一台机器即可。
- 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。
- 测试数据量：100GB。

租户规格：

- MAX_CPU = 26
- MEMORY_SIZE = 70g

测试结果（单位均为秒）：

Query	V4.0.0_CE	V3.1.0_CE	提升%
Q1	2.34	14.04	500.00%
Q2	0.14	1.12	700.00%
Q3	0.72	13.57	1784.72%
Q4	0.56	2.51	348.21%
Q5	2.25	12.31	447.11%
Q6	0.23	7.33	3086.96%
Q7	1.52	10.38	582.89%
Q8	0.70	11.42	1531.43%
Q9	5.22	30.99	493.68%
Q10	1.24	6.84	451.61%
Q11	0.23	1.22	430.43%
Q12	1.62	8.64	433.33%
Q13	2.41	7.59	214.94%
Q14	0.36	1.51	319.44%
Q15	0.79	3.01	281.01%
Q16	0.66	2.66	303.03%
Q17	0.63	8.60	1265.08%
Q18	0.93	7.88	747.31%
Q19	0.78	9.36	1100.00%
Q20	1.17	10.95	835.90%
Q21	2.42	12.27	407.02%
Q22	1.24	4.05	226.61%
Total	28.16	188.25	568.50%

兼容性变更产品行为变更

功能

变更点说明

备份能力	不再支持集群级备份，支持租户级备份 不支持二次备份功能 数据校验能力、归档日志压缩和手工清理功能将在后续版本支持
时钟源	租户时钟源不再支持 LTS 模式
主备库	不再支持集群级主备库，后续版本将支持租户级主备库
Queuing 表	不再支持
加密投票型副本	不再支持
Partition Group	不再支持
Table Group	不再支持 binding=true 的 Table Group
合并	不再支持集群级合并，支持租户级合并 不再支持轮转合并，租户内所有 Zone 一起合并
RESOURCE UNIT	参数设置变更，删除参数存储空间 (MAX_DISK_SIZE) 和 Session 个数 (MAX_SESSION_NUM)，增加参数日志盘空间 (LOG_DISK_SIZE)。删除参数 MAX_MEMORY / MIN_MEMORY，使用 MEMORY_SIZE 参数设置内存资源。 必选参数为 MAX_CPU 和 MEMORY_SIZE，可选参数 MIN_CPU 默认值等于 MAX_CPU，LOG_DISK_SIZE 默认值为 MEMORY_SIZE * 3，MAX_IOPS、MIN_IOPS、IOPS_WEIGHT 默认值是根据租户 CPU 个数按比例分配。
Meta 租户	新增 Meta 租户，是 OceanBase 内部自管理的租户，每创建一个用户租户会创建一个对应的 Meta 租户，其生命期与用户租户保持一致。Meta 租户用于存储和管理用户租户的集群私有数据，这部分数据不需要跨库物理同步以及物理备份恢复，例如：配置项、位置信息、副本信息、日志流状态、备份恢复相关信息、合并信息等。
租户的 UNIT 资源配置	不再支持按 Zone 个性化配置租户的 Unit 个数，只能按照 Unit Group 维度整体调整，扩缩容时要求租户所有 Zone 的 Unit 个数相同
Primary Zone 和 Locality 配置	不再支持表级、DB 级、tablegroup 级配置 primary zone 和 locality，仅支持租户级别配置 primary zone 和 locality
MySQL 模式下，语法 <code>/*! xxx */;</code>	版本命令全部执行，不支持的命令不进行报错处理
浮点类型	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

视图/实体表/虚拟表

- OceanBase V4.0 新增诸多自有视图，主要分为两大类：数据字典视图和动态性能视图。数据字典视图用于展示内部表这些实体表中存储的数据或数据的组合，用以替代之前直接对内部表的查询。动态性能视图用于展示系统运行过程中很多内存状态信息，用以替代之前直接对虚拟表的查询。举例如下：
 - DBA_OB_SERVERS 展示集群的服务器节点信息，当系统管理员使用 `ADD SERVER` 和 `DELETE SERVER` 运维命令增删机器后，这里查询的机器数量就会发生变化。该视图同时展示 RootService 是否在机器上（WITH_ROOTSERVER）、心跳是否正常（STATUS）、程序的版本号（BUILD_VERSION）等信息。
 - GV\$OB_UNITS 展示每台服务器上具体有哪些租户的哪些 UNIT 的动态性能视图。该视图用于查询各个服务器上最新的 UNIT 信息，包含了 Meta 租户的 UNIT 和隐藏系统租户的 UNIT，Meta 租户 UNIT 是每个系统内部构造的。
- OceanBase V4.0 数据库对实体表和虚拟表进行了重新梳理和设计。举例如下：
 - 实体表 `__all_tenant_partition_meta_table` 记录 PG 内 partition 汇报的元信息，V4.0 不再支持 PG 故将该表删除。
 - 实体表 `__all_table_v2` 记录 table 多版本 schema 信息，V4.0 将表名称变更为 `__all_table`。
 - 虚拟表 `__all_virtual_core_meta_table` 展示系统租户 `__all_core_table` 表副本的位置信息，V4.0 去掉的关键字段 `table_id`、`partition_id`、`partition_cnt`，增加关键字段 `ls_id`。

有关视图的详细信息，参见 [系统视图](#)。

配置项变更

由于架构升级很多配置项已不再起作用，V4.0 对于无效的配置项进行了删减和变更。比如用于触发 Major freeze 的 `minor_freeze_times` 已经删除。还调整了一些配置项的默认值，因为从底层解决了大事务能力的支持，事务不再受冻结操作影响，所以 `freeze_trigger_percentage` 默认值调整为 20%。

配置项	用途及不兼容点说明	状态变化
<code>minor_warm_up_duration_time</code>	数据预热相关	废弃
<code>minor_deferred_gc_time</code>	转储完成延迟回收 SSTable 的时间	废弃
<code>_minor_deferred_gc_level</code>	转储完成延迟回收 SSTable 的级别	废弃

max_kept_major_version_number	最多保存 major 版本数量	废弃
_enable_sparse_row	是否开启稀疏行	废弃
minor_freeze_times	两次 major 之间的 minor 次数	废弃
clog_usage_limit_size	日志盘使用总空间，改由租户 unit config 设置	废弃
enable_separate_sys_clog	系统表、用户表日志流分离存储	废弃
clog_max_unconfirmed_log_count	滑动窗口长度	废弃
_ob_clog_disk_buffer_cnt	batch buffer 的数目	废弃
clog_cache_priority	clog cache 优先级	废弃
index_clog_cache_priority	ilog cache 优先级	废弃
_clog_aggregation_buffer_amount	聚合提交所使用的 buffer 数目	废弃
_flush_clog_aggregation_buffer_timeout	聚合提交 freeze 间隔	废弃
_enable_clog_rpc_aggregation	是否对所有日志 RPC 开启聚合	废弃
system_cpu_quota	500 租户 CPU 规格	废弃
flush_log_at_trx_commit	对标 MySQL 同名配置项	废弃
election_cpu_quota	election 租户 CPU 规格	废弃
enable_election_group	是否开启选举虚拟组	废弃
enable_log_archive	是否开启日志归档	废弃
clog_disk_usage_limit_percentage	日志盘使用空间占总磁盘空间的上限	废弃
clog_disk_utilization_threshold	日志盘正常运行期间使用空间占总磁盘空间的百分比	废弃
minor_compact_trigger	储触分层转储的阈值	语义变化，集群级配置项变更为租户级配置项
major_compact_trigger	冻结后触发版本合并的阈值	语义变化，集群级配置项变更为租户级配置项
freeze_trigger_percentage	转储或合并转储触发阈值	默认值由 70% 调整为 20%
writing_throttling_trigger_percentage	memtable 写入限流触发阈值	默认值由 100% 调整为 60%

writing_throttling_maximum_duration	memtable 写入限流时长	默认值由 1h 调整为 2h
ob_trx_timeout	事务超时时长	默认值 100s 调整为 1d
ob_trx_idle_timeout	空闲事务超时时长	默认值由 120s 调整为 1d

升级说明

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

周边配套

OceanBase 数据库 V4.0.0 推荐使用的配套工具版本如下：

组件	配套版本
OCP 社区版	V4.0.0
ODC 社区版	V4.0.0
OceanBase Proxy (OBProxy)	V4.0.0
OceanBase JDBC	V2.4.0
MySQL JDBC	V5.1.47
OceanBase ODBC	V2.0.6
OceanBase Client (OBClient)	V2.2.0
OceanBase Deployer (OBD)	V1.6.0
OceanBase Agent (OBAgent)	V1.2.0
ob-operator	V1.1.0

约束限制

- OceanBase V4.0.0 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 V4.1 版本补齐负载均衡能力。
- OceanBase V4.0.0 暂时不支持主备库模式部署的产品形态，计划在 V4.1 版本补齐主备库产品形态。
- OceanBase V4.0.0 暂时不支持复制表，后续版本将补齐该能力。
- OceanBase V4.0.0 暂时不支持 INT 类型到 BIGINT 类型的变更，后续版本将补齐该能力。
- OceanBase V4.0.0 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 V4.0.0 节点在生产环境最低要求运行规格为 4C16GB，租户 CPU 资源建议 2C 及以上，租户内存资源建议 4GB 及以上。
- OceanBase V4.0.0 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 V4.0.0 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。