ALIBABA CLOUD

阿里云

云原生关系型数据库PolarDB O引擎

Polardb Oracle（合集）

文档版本：20220214

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，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。

3. 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。阿里云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，并在阿里云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阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

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 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.动态与公告 05
1.1. 新版本更新说明 05
1.2. 安全公告 31
1.2.1. Apache Log4j2漏洞对PolarDB无影响 31
2.性能白皮书 32
3.快速入门 41
4.数据迁移/同步 53
4.1. 数据迁移和同步方案概览 53
4.2. 从自建Oracle迁移至PolarDB-O集群（迁移结构） 53
4.3. 从自建Oracle迁移至PolarDB-O集群（迁移数据） 56
4.4. PolarDB O引擎集群间的单向同步 61
5.常见问题 67
6.服务等级协议 71

1.动态与公告

1.1. 新版本更新说明

本文介绍PolarDB O引擎的产品功能动态，分别为内核小版本、控制台、时空数据库和API的更新说明。

> **说明** 您可通过如下语句查看PolarDB O引擎的内核小版本的版本号：
>
> ```
> show polar_version;
> ```

2022年01月

内核小版本（V1.1.20）

类别	名称	描述	相关文档
新增	支持表的闪回删除	新增闪回删除功能，可以快速恢复已经删除的表，以及查看和清理回收站。	闪回删除
	OPT_PARAM Hint功能扩展	增加了OPT_PARAM Hint在SQL级别指定特定的参数值。	OPT_PARAM Hints
	PX多级分区表	支持多级分区的并行查询。	对分区表使用跨机并行查询
	DML use GLOBAL INDEX	支持使用GLOBAL INDEX SCAN进行DML操作。	DML支持GLOBAL INDEX SCAN
	支持GBK/GB18030编码	支持GBK/GB18030服务器编码格式。	无

时空数据库（V4.4）

类别	名称	描述	相关文档
新增	ST_LinearStretch函数	新增ST_LinearStretch函数，将影像进行线性拉伸，作为一个新的raster返回。	ST_LinearStretch
	ST_BuildPercentiles函数	新增ST_BuildPercentiles函数，统计影像的像素值百分比。	ST_BuildPercentiles
	ST_ComputeStatistics函数	新增ST_ComputeStatistics函数，一次性计算栅格数据的统计值，直方图以及百分比。	ST_ComputeStatistics
	ST_Percentile函数	新增ST_Percentile函数，获取影像的像素百分比。	ST_Percentile
	并行计算	ST_SummaryStats和ST_BuildHistogram函数支持并行计算能力。	无

类别	名称	描述	相关文档
优化	ST_ExportTo函数	ST_ExportTo函数在导出数据时，如果目标文件已存在，返回false取代报错消息。	无
	ST_ImportFrom函数	ST_ImportFrom函数在导入数据时出错，会返回具体的错误详情方便排查错误原因。	ST_ImportFrom
缺陷修复	修复栅格裁剪时由于四舍五入可能会有半个像素丢失的问题。		
	修复由于栅格数据带有 `NAN` 和 `INF` 等数据，在导入和统计值会出错的问题。		

2021年12月

内核小版本（V1.1.19）

类别	名称	描述	相关文档
新增	sql mapping	PolarDB O引擎支持SQL映射功能，将错误的SQL映射为正确的SQL，从而避免由于错误SQL引起的问题。	polar_sql_mapping
	PX支持并行Update	跨机并行查询支持并行update，包括带子查询的update。 通过设置polar_px_enable_update参数，开启或关闭该功能，默认值为false。 • 设置为true时，表示开启并行update功能。 • 设置为false时，表示关闭并行update功能。 通过设置polar_px_update_dop_num参数，控制并行update的并发数，默认为6，范围为1~128。 通过设置polar_px_enable_remove_update_redundant_motion参数，控制当读写worker数相同的情况时，是否移除多余的motion算子，默认为false。 • 设置为true时，表示开启移除多余motion算子的功能。 • 设置为false时，表示关闭移除多余motion算子的功能。	跨机并行查询
	PX支持并行Delete	跨机并行查询支持并行delete，包括带子查询和不带子查询的并行delete。 通过设置polar_px_enable_delete参数，开启或关闭该功能，默认值为false。 • 设置为true时，表示开启并行delete功能。 • 设置为false时，表示关闭并行delete功能。 通过设置polar_px_delete_dop_num参数，控制并行delete的并发数，默认为6，范围为1~128。 通过设置polar_px_enable_remove_delete_redundant_motion参数，控制当读写worker数相同的情况时，是否移除多余的motion算子，默认为false。 • 设置为true时，表示开启移除多余motion算子的功能。 • 设置为false时，表示关闭移除多余motion算子的功能。	

类别	名称	描述	相关文档
	PX支持partition wise join	跨机并行查询支持分区表的partition wise join高级特性。	对分区表使用跨机并行查询
	兼容Oracle视图	PolarDB O引擎兼容Oracle的all_tab_columns /all_tab_comments2个视图，以及all_objects视图的部分列。	ALL_TAB_COLUMNS、ALL_OBJECTS
	支持国密加密算法	PolarDB O引擎支持国密加密算法，增强数据库安全性。	无

优化	支持DBMS_SQL.TO_CURSOR_NUMBER /DBMS_SQL.TO_REFCURSOR	支持DBMS_SQL.TO_CURSOR存储过程。
		例如，
		`-- 从本地动态SQL转换为DBMS_SQL`
		`curid := dbms_sql.to_cursor_number(cursrc);`
			无

时空数据库（V4.3）

类别	名称	描述	相关文档
新增	ST_PixelAsPolygon函数	新增ST_PixelAsPolygon函数，将栅格像素转换为Polygon对象。	ST_PixelAsPolygon
	ST_PixelAsPolygons函数	新增ST_PixelAsPolygons函数，将栅格像素转换为Polygon对象集合。	ST_PixelAsPolygons
	ST_PixelAsPoint函数	新增ST_PixelAsPoint函数，将栅格像素转换为基于点方式表述的几何对象。	ST_PixelAsPoint
	ST_PixelAsCentroid函数	新增ST_PixelAsCentroid函数，将栅格像素转换为基于中心点方式表述的几何对象，	ST_PixelAsCentroid
	ST_PixelAsPoints函数	新增ST_PixelAsPoints函数，将栅格像素转换为点对象集合。	ST_PixelAsPoints
	ST_PixelAsCentroids函数	新增ST_PixelAsCentroids函数，将栅格像素转换为中心点对象集合。	ST_PixelAsCentroids
优化	几何模型	Geometry兼容PostGIS 3.1.4接口。	无
	ST_AsDatasetFile函数	ST_AsDatasetFile函数支持通过Grib2数据格式输出。	无
缺陷修复	修复栅格netcdf导入时，空间参考获取不正确的问题。		
	修复多外包框下equals结果不正确，导致可能外包框更新错误的问题。		

2021年10月

内核小版本（V1.1.17）

类别	名称	描述	相关文档
新增	分区表DML支持执行期修剪	分区表支持update、delete在执行器期间（runtime）执行分区剪枝。	分区剪枝
	聚合函数支持order by子句	聚合函数支持包含 `order by` 子句的简单查询。	聚合函数
	trigger支持指定schema	支持在triggername前允许指定schema。	触发器
	二维关联数组支持	新增支持二维关联数组。	关联数组
	PX支持并行写并行读的Insert....Select....加速	跨机并行查询在Insert...Select...单写多读的基础上，改进为并行写并行读，进一步提升并行Insert...Select...的性能。 可以通过设置polar_px_insert_dop_num参数，控制写并行度，默认值为6。 • 当polar_px_insert_dop_num设置为1时，执行单写多读计划。 • 当polar_px_insert_dop_num设置为大于1时，执行多写多读计划。	跨机并行查询
	PX支持Insert...Select from tableless的加速	跨机并行查询支持Insert...Select from tableless（无表查询）的加速，进一步提升并行Insert...Select...的性能。 可以通过设置polar_px_enable_insert_from_tableless参数，开启或关闭无表查询加速，默认值为0。 • polar_px_enable_insert_from_tableless=1，表示开启。 • polar_px_enable_insert_from_tableless=0，表示关闭。	
	PX 支持Insert...Select对有序性敏感的SQL强制有序	跨机并行查询可以通过设置参数 `polar_px_enable_insert_order_sensitive`，确保写入的结果强制有序，默认值为true。 • 将 `polar_px_enable_insert_order_sensitive` 设置为true，则SQL语句中含有 `order by` 或者是无表查询的情况，写入结果强制有序。 • 将 `polar_px_enable_insert_order_sensitive` 设置为false，则并行Insert不保证有序。	
	PX支持RowNum	跨机并行查询新增支持Oracle兼容的RowNum表达式。	
	PX阻塞DDL超时取消	当DDL操作与PX查询发生锁冲突而等待时，可以设置DDL等待锁的时间，如果等待超时，则强制中断PX查询，优先让DDL获得锁并完成执行。	
	Leading Hint支持	支持Leading Hint，Leading Hint是一个能指定超过一个表的多表Hint。	Leading Hint
	DELETE FROM subquery	支持DELETE FROM subquery。	无

类别	名称	描述	相关文档
	select unique的语法支持	新增select unique语法规则，支持了unique作为distinct的同义词。	无
	UPDATING IN SPL TRIGGER	在DML trigger中，支持updating('colname')条件表达式。	无
	merge命令中的update/delete支持where	支持了merge命令中update/delete出现where的场景。	无
	insert select 支持别名	支持了insert命令select别名的情况。	无
	copy支持多个字符的分隔	Copy功能新增支持多个字符的分隔。	无
优化	修复PX对polar_row_id的支持	跨机并行查询支持polar_row_id。	无
	standby并行回放	Standby节点通过多进程来加速回放，减少standby节点与master节点之间的延迟。	无

时空数据库（V4.1）

类别	名称	描述	相关文档
优化	ST_ImportFrom函数	增强ST_ImportFrom函数，增加OSS文件映射为内存文件功能，提升数据导入速度。	ST_ImportFrom、ganos.raster.memory_oss_file_max_size
	Gist/TrajGist索引	GisT索引和TrajGisT索引新增算子族trajgist_ops_multi，用于多外包框索引键值。	GisT索引、TrajGisT索引、ganos.trajectory.index_split_config
缺陷修复	新增矢量金字塔支持元数据信息查询。		
	修复轨迹对象执行相交判断时内存泄露问题。		

2021年9月

内核小版本（V1.1.16）

类别	名称	描述	相关文档
	outline支持手动query改写	在PolarDB查询改写不符合预期时，支持使用outline来实现SQL的查询改写。	调整执行计划

类别	名称	描述	相关文档
新增	PX支持AssertOp算子	PX支持执行器运行时约束检查。	跨机并行查询
	PX支持Oracle Decode表达式	PX新增支持Oracle Decode表达式。	
	PX支持动态分区裁剪	PX支持动态分区剪裁，提升分区表hash join时的性能。	
	Bind Message中支持Holdable Portal	支持SQL语句的跨事务访问， 需要在SQL中添加 `polar_holdable_hint` 的HINT， 例如， `SELECT /*+ polar_holdable_hint */;` 。	无
	Oracle风格的select for update	`select for update of` 全面兼容Oracle语法。	无
优化	PX支持Const表达式	跨机并行查询支持Const表达式。	无
	算子级内存限制	通过算子级内存限制，降低OOM风险，提升PolarDB稳定性。	无

时空数据库（V4.0）

类别	名称	描述	相关文档
优化	栅格表面计算函数	增强栅格表面计算函数返回Raster对象取代像素矩阵。	ST_Aspect、ST_Slope、ST_Hillshade
	ST_Split函数	增强ST_Split按照指定的规则对轨迹对象进行切分返回子轨迹数组。	ST_Split
缺陷修复	修复ST_SrEqual在某些条件下判断失败的问题。		

2021年8月

内核小版本（V1.1.15）

类别	名称	描述	相关文档
新增	Index Advisor记录索引前后的代价	Index_advisor_log表中增加两列，分别表示增加索引前后的代价，方便上层应用（例如，数据库自治服务DAS） 评估创建索引带来的收益。	无
	PX索引加速构建支持GLOBAL INDEX	PX新增支持了加速创建B-Tree索引的GLOBAL索引。	跨机并行查询、使用跨机并行查询加速索引创建、创建GLOBAL INDEX

2021年7月

内核小版本（V1.1.14）

类别	名称	描述	相关文档
新增	适配GCC9编译	GCC升级至GCC9.2.1，支持更多编译优化，提高性能。	无
	outline支持hint	outline支持hint功能来控制需要被outline抓取的执行计划。	调整执行计划
	跨机并行查询支持hash分区	支持hash分区的并行查询，从而平衡I/O，改善系统性能。	对分区表使用跨机并行查询
	GLOBAL INDEX支持对分区表进行DDL操作	支持在对分区表进行DDL操作时（如DROP、SPLIT、MERGE等），操作该分区表下的所有GLOBAL INDEX。	支持对分区表进行DDL操作
优化	ORA_HASH函数兼容性	ORA_HASH函数替换为DBMS_UTILITY.GET_HASH_VALUE，计算给定字符串的散列值。	无

时空数据库（V3.9）

类别	名称	描述	相关文档
新增	ST_RemoveDriftPoints函数	新增ST_RemoveDriftPoints函数，使用指定规则删除轨迹中的漂移点。	ST_removeDriftPoints
	ST_Split函数	新增ST_Split函数，使用指定的几何对象对轨迹进行切分。	ST_Split
	ST_ExportTo函数	新增ST_ExportTo函数，将轨迹导出到外部文件存储，数据库内仅保留元数据。	ST_ExportTo
	ST_IsExternal函数	新增ST_IsExternal函数，检查轨迹是否存储于外部文件。	ST_IsExternal
	ST_ImportFrom函数	新增ST_ImportFrom函数，将外部存储模式的轨迹重新转化为存储在数据库内的轨迹。	ST_importFrom
	ST_StorageLocation函数	新增ST_StorageLocation函数，返回轨迹存储在外部的位置。	ST_StorageLocation
	ST_AKID函数	新增ST_AKID函数，返回读取轨迹时，与OSS交互时所使用的AccessKeyID。	ST_AKID
	ST_SetAccessKey函数	新增ST_SetAccessKey 函数，设置读取轨迹时，与OSS交互时所使用的AccessKey（包括AccessKeyID和AccessKeySecret，类似用户名与密码）。	ST_SetAccessKey
	ST_SetAKID函数	新增ST_SetAKID函数，设置读取轨迹时，与OSS交互时所使用的AccessKeyID。	ST_SetAkId
	ST_SetAKSecret函数	新增ST_SetAKSecret函数，设置读取轨迹时，与OSS交互时所使用的AccessKeySecret。	ST_SetAkSecret

类别	名称	描述	相关文档
	ST_SetStorageLocation函数	新增ST_SetStorageLocation函数，设置读取轨迹时，外部文件的存储位置。	ST_SetStorageLocation
	ST_DeleteGtf函数	新增ST_DeleteGtf函数，删除指定文件夹下所有的轨迹导出的文件。	ST_DeleteGTF
缺陷修复	修复创建矢量金字塔使用字符串方式过滤失败问题。		
	修复QGIS连接时无法列出图层列表的问题。		

2021年6月

内核小版本（V1.1.13）

类别	名称	描述	相关文档
新增	polar_stat_sql/polar_stat_plan增加IO统计展示	• 新增可通过查询不同视图获取相关资源信息，进而方便定位性能问题。 • 新增PolarDB O引擎性能诊断，包含实例问题分析和SQL性能分析。 • polar_stat_sql/polar_stat_plan视图增加IO统计展示相关参数。	性能视图、性能诊断
	跨机并行查询	跨机并行查询功能可用于BRIN索引扫描，进一步提升查询性能。	跨机并行查询概述
		新增polar_bt_write_page_buffer_size参数，指定了索引构建过程中的写IO策略。	使用跨机并行查询加速索引创建
		支持从三种粒度指定跨机并行查询的范围： • 系统粒度：指定所有session所有查询是否开启跨机并行查询。 • 会话粒度：指定当前session是否开启跨机并行查询。 • 查询粒度：指定具体查询是否开启跨机并行查询。	使用跨机并行查询进行分析型查询
		支持对分区表使用跨机并行查询的功能，提升数据库性能。	对分区表使用跨机并行查询
优化	基于RTO的crash recover时间改进	支持配置polar_crash_recovery_rto参数，指定实例期望的RTO时间，从而保证在发生crash recovery时，数据库可以在期望的RTO时间内完成崩溃恢复重启，恢复可用状态。	无

时空数据库（V3.8）

类别	名称	描述	相关文档
新增	ST_AsDatasetFile函数	新增ST_AsDatasetFile函数，支持将指定范围的栅格对象以指定文件格式的二进制流进行返回。	ST_AsDatasetFile

新增类别	名称	描述	相关文档
	ST_RasterDrivers函数	新增ST_RasterDrivers函数，支持对所有栅格数据驱动的状态进行查询。	ST_RasterDrivers
缺陷修复	修复ST_Clip函数指定空间参考进行重投影操作失败的问题。		

2021年5月

内核小版本（V1.1.12）

类别	名称	描述	相关文档
新增	跨机并行查询	支持跨机并行查询计算引擎： • 支持多个只读节点分布式地执行SQL查询，加速PolarDB O引擎的查询性能。 • 支持通过跨机并行查询加速构建索引，提升大表场景下索引创建效率。	跨机并行查询
	分区表支持全局索引	分区表支持全局索引；同时支持并发构建索引，即在创建索引的过程中不阻塞表上的DML操作。	创建GLOBAL INDEX
优化	性能监控	新增max_slot_wal_keep_size参数，限制使用复制槽（replication slot）的文件大小，防止WAL文件堆积。	无
	DBLink	默认开启Unsafe Collate下推和ALL/ANY下推，从而极大提高查询性能。可通过polar_enable_pushable_unsafe_collate_remote与polar_enable_pushable_all_any_remote参数进行关闭。	使用DBLink跨库查询数据
	Rownum	对Rownum进行了优化，兼容rownum和varchar2/text比较时，按照数值类型比较的行为。	无
	polar_resource_group插件	polar_resource_group插件新增total_mem_limit_remain_size、idle_mem_limit_rate、enable_terminate_active、policy_mem_release、enable_log等参数，避免内存占用过多而影响数据库进程。	无
	提高只读节点启动速度	提高只读节点的启动速度，提升性能。	无
	主备切换	通过并行回放，加速恢复主备切换后的数据库服务，通常情况下主库异常可以15s内恢复服务。	无

时空数据库（V3.7）

类别	名称	描述	相关文档
新增	时空并行查询	支持空间索引并行查询，提升查询效率。	开启时空并行查询
优化	栅格图像处理	ST_SubRaster函数支持栅格像元类型转换以及像元值拉伸。支持多波段遥感影像转三波段图像的AI识别场景。	ST_SubRaster

类别	名称	描述	相关文档
缺陷修复	修复栅格解析无效的直方图信息可能会导致数据库崩溃的问题。		
	修复进行投影变化操作后，数据库退出时可能会崩溃的问题。		
	修复栅格数据导入时可能会出现崩溃的问题。		

2021年4月

内核小版本（V1.1.11）

类别	名称	描述	相关文档
新增	idle_session_timeout参数	新增idle_session_timeout参数，用于控制空闲会话的保留时间，超出保留时间的会话将会被释放。 • 取值范围：0~2147483647 0（默认值）表示关闭功能。 • 单位 ○ `'ms'`：毫秒（默认单位） ○ `'s'`：秒 ○ `'min'`：分钟 ○ `'h'`：小时 ○ `'d'`：天	设置参数
	约束DDL	支持 `ALTER TABLE ... ENABLE/DISABLE CONSTRAINT`，用于启用或禁用约束（CONSTRAINT）。	ALTER TABLE ... ENABLE/DISABLE CONSTRAINT
	表空间DDL	支持表空间（TABLESPACE）相关的DDL操作，例如CREATE TABLESPACE。	无
	Numeric数据类型	兼容Oracle的Numeric数据类型。如若需要可以通过设置 `polar_enable_comp_numeric_format` 参数进行开启，该参数默认关闭。开启命令如下： `SET polar_enable_comp_numeric_format=on;`	无
缺陷修复	修复DBMS_JOB内置包跨库使用的问题。		
	修复CONNECT BY与ROWNUM函数连用的问题。		
	修复CONNECT BY找不到上层对象的问题。		

时空数据库（V3.6）

类别	名称	描述	相关文档
		新增ST_RPCGeoreference函数，用于获取栅格数据RPC（Rational Polynomial Coefficients）信息。	ST_RPCGeoreference

类别	名称	描述	相关文档
新增	栅格数据类型	新增ST_SetRPCGeoreference函数，用于设置栅格数据RPC信息。	ST_SetRPCGeoreference
		新增ST_RPCRectify函数，用于根据栅格影像的RPC参数对栅格进行校正操作，返回校正后的栅格对象。	ST_RPCRectify
	GisT索引	新增使用并行方式创建GisT索引。	并行创建空间索引
缺陷修复	修复使用pg_dump时，自定义的空间参考无法导出的问题。		

2021年3月

内核小版本（V1.1.10）

类别	名称	描述	相关文档
新增	分区表	分区表支持Before Row-level触发器。	Before Row-level触发器
	polar_outline插件	支持使用polar_outline调整执行计划，实现SQL语句优化。	调整执行计划
优化	分区表	INTERVAL RANGE分区支持子分区模板。	子分区模板
	性能优化	通过在共享内存中缓存表大小信息，优化了数据文件I/O读性能。	无
	兼容性	INTERVAL与FLOAT8数据类型支持如下操作符： • + • - • * • /	无
	兼容性	新增OPT_PARAM HINT，通过该HINT可以在SQL级别指定特定参数的值。	OPT_PARAM Hints

时空数据库（V3.5）

类别	名称	描述	相关文档
新增	大对象存储优化	新增Simple存储策略，优化大对象存储。	使用Simple存储策略
优化	栅格对象	使用栅格对象更新一个具有分块数据的栅格对象进行时，系统会自动删除原有栅格对象的分块数据。	无
	修复Trajectory扩展无法升级的问题。		
	修复某些情况下，栅格对象采用Average重采样时出错的问题。		

缺陷修复类别	名称	描述	相关文档
	修复轨迹对象中如果多个时间戳的轨迹点相同，查询结果可能不正确的问题。		
	修复Geos对象转换失败后，直接退出的问题，同时对转换失败的原因进行提示。		

内核小版本（V1.1.9）

类别	名称	描述	相关文档
新增	分区表	新增INTERVAL RANGE分区。INTERVAL RANGE分区是RANGE分区的扩展。当新插入的数据超过现有分区的范围时，该分区允许数据库自动创建新分区。	INTERVAL RANGE分区
		RANGE分区和INTERVAL RANGE分区可以相互转换。	转换RANGE和INTERVAL RANGE分区
		新增SUBPARITITION TEMPLATE子分区模板，用于快速定义子分区，简化子分区的规范。您只需在子分区模板中定义一次子分区描述符，然后将该子分区模板应用于表中的每个分区，从而快速批量地为表中的分区定义子分区描述符。	子分区模板
	AWR监控	自动负载信息库（Automatic Workload Repository，简称AWR）是一种性能收集和分析工具。通过AWR工具，可以从数据库的动态视图等统计信息记录中生成一份该时段的统计分析报告，分析数据库在某个时间段的性能。	无
优化	分区表	分区表在计划阶段进行修剪优化，提高查询性能。	无
	数据类型	PolarDB O引擎使用timestamp作为Date数据类型，精确到微秒；而Oracle的Date数据类型只精确到秒，为了保证PolarDB O引擎和Oracle的兼容性，将自动把Date数据类型转换为 timestamp(0)。	无
	WAL	采用流水线技术，优化WAL开销占比，提高整体系统的性能。	无
	TO_CHAR	在服务端自动识别 `TO_CHAR(text1, text2)` 的格式：如果text1为合法的日期类型，text2为合法的日期格式类型， 则自动将text1当成timestamp来处理。	无
缺陷修复	修复无用SIGHUP信号。		
	修复MERGE INTO分区表会导致崩溃的问题。		
	修复pg_icu_collate_names时会访问非法内存的问题。		
	修复ALTER QUEUE ADD CALL访问非法内存的问题。		
	修复POLICY系统表定义中rdname字段的属性设置为定长，但是传入新值时使用的确是变成字符串，导致访问非法内存的问题。		
	修复TDE功能中没有声明导致函数返回值地址被截断的问题。		

类别	名称	描述	相关文档
	修复roaringbitmap插件对输入异常参数时访问非法内存的问题。		
	修复当只读节点落后很多日志时，可能不会及时给主节点流复制反馈，导致流复制链接中断的问题。		

时空数据库（V3.4）

类别	名称	描述	相关文档
新增	大对象存储优化	为了加速时空数据索引构建、提高空间查询效率，优化时空对象的存储模式，支持使用时空大对象特征签名，新增polar_enable_storage_partial参数，支持时空对象行内与行外组合存储。	时空大对象特征签名
	栅格模块支持函数并行化执行	ST_ImportFrom函数支持栅格数据并行化导入。	ST_ImportFrom
		ST_BuildPyramid函数支持栅格数据并行创建金字塔。	ST_BuildPyramid
	Trajectory Empty	新增Trajectory Empty对象，支持某些场景下不符合要求而返回NULL对象。	无
缺陷修复	修复ST_AddZ函数在16BSI情况下可能结果不正确的问题。		
	修复Trajectory模块在某些情况下无法升级的问题。		
	修复Trajectory模块中部分函数无法利用索引的问题。		
	修复部分nd函数没有正确处理时间段不相交的场景，导致返回error而非false的问题。		

2021年1月

内核小版本（V1.1.8）

类别	名称	描述	相关文档
新增	DBMS_STATS插件	支持DBMS_STATS插件，用于收集数据库运行过程中的相关统计信息，提供现有统计信息的备份，以及将备份统计项进行还原。	DBMS_STATS
	写后读事务拆分	在READ COMMITTED级别下，支持写后读语句的事务拆分。可将事务开启到事务结束中所有的读请求发送到只读节点，从而保证事务中的读写一致性。	事务拆分
	OR表达式	原有OR表达式无法自动转换为UNION ALL，导致部分情况下查询效率很低，必须全表扫描。现支持将OR表达式自动转换为UNION ALL，实现逻辑优化，大幅提升查询性能。	无
	ECPG兼容性	• ECPGPlus兼容Pro*C部分语法。 • ECPGPlus兼容原生Oracle的连接方式。	无
	Oracle兼容性	`GROUP BY CONST/EMPTY` 子句兼容Oracle。	无

优化类别	名称	描述	相关文档
	PBP	默认打开主节点的持久化缓冲池（Persisted Buffer Pool）。	无
	同步Kafka分区表	PolarDB O引擎到kafka同步分区表配置优化。按父表名配置，同步后的表名也是父表名。	无
	用户	调整 `CREATE USER ... IDENTIFIED BY ...` 语法仅支持创建普通用户。	创建用户
缺陷修复	修复DBMS_JOB的无主键表在逻辑复制中删除任务报错的问题。		
	修复资源组（Resource Group）进程在内存资源超限时出现内存泄漏的问题。		
	修复Standby Promote时间较长的问题。		
	修复***_all_tables视图中temporary字段的临时表参数显示问题。		
	修复审计日志越界内存访问的问题。		
	修复PolarDB O引擎到PostgreSQL的DBLink查询结果截断的问题。		
	修复分区表子分区多分区键剪枝（prune）缺陷。		

时空数据库（V3.3）

类别	名称	描述	相关文档
新增	ST_JaccardSimilarity函数	新增ST_JaccardSimilarity函数，用于计算轨迹对象相似度。	ST_JaccardSimilarity
	ST_Transform函数	新增ST_Transform函数，用于将轨迹从一个空间参考系转换到另一个空间参考系。	ST_Transform
优化	矢量金字塔	创建矢量金字塔时，支持用户指定创建范围，解决在部分场景下无法自动计算空间范围的问题。	无
缺陷修复	修复某些环境下，Ganos数据目录设置不正确的问题。		

2020年12月

内核小版本（V1.1.7）

类别	名称	描述	相关文档

类别	名称	描述	相关文档
新增	分区表	新增Merge Partitions功能，用于合并两个分区表。	• ALTER TABLE... MERGE PARTITION • ALTER TABLE...MERGE SUBPARTITION
	兼容Oracle风格解码	支持Oracle的Base64解码风格，可以让BASE64_DECODE函数自动识别错误格式的编码串，然后将编码串中有效的前半部分进行解码。	polar_enable_base64_decode
	日期格式	支持插入或更新 `YYYYMMDDHH24MISS` 格式的日期。	无
	SYS.***_TAB_COLS视图	新增 `SYS.***_TAB_COLS` 视图，用于查询表和视图中所有列的信息。	• ALL_TAB_COLS • DBA_TAB_COLS • USER_TAB_COLS
	polar_utility插件	新增 `polar_utility` 插件。	polar_utility
	存储过程	存储过程允许返回holdable refcursor。	无
	DBMS_JOB包	新增DBMS_JOB包，您可以通过高权限账号进行调用或跨库调度。	DBMS_JOB
优化	UTL_ENCODE	新建的普通用户支持使用UTL_ENCODE插件。	UTL_ENCODE
	polar_agg插件	polar_agg插件的状态默认为开启。	无
	fdw插件	增强fdw插件的稳定性。	无
缺陷修复	修复存储过程中使用 `SELECT INTO` 查询空值报错的问题。		
	修复执行 `ALTER TABLE... EXCHANGE PARTITION` 函数后，进行索引扫描导致约束失效的问题。		

控制台

类别	名称	描述	相关文档

类别	名称	描述	相关文档
新增	DBLink	新增PolarDB O引擎到ECS自建PostgreSQL的DBLink，实现从PolarDB O引擎到PostgreSQL的跨库数据互通。结合PostgreSQL中的DBLink，可以实现两种不同数据库之间的双向数据互通。 路径：集群详情页的**配置与管理 > 数据库管理**。 	创建到PostgreSQL的DBLink

时空数据库（V3.2）

类别	名称	描述	相关文档
新增	ST_AsPNG	新增矢量金字塔返回图片格式（基于流形式）功能，用于矢量数据的快速图形化显示。	ST_AsPNG
	栅格数据	新增栅格数据类型JPEG2000压缩算法，支持16bit栅格数据压缩存储。	无
优化	st_dwithin	优化st_dwithin距离查询，提升查询性能。	无
缺陷修复	修复轨迹数据类型时间相交错误问题。		
	修复Ganos内存拓扑索引的缺陷。		

API

类别	名称	描述
优化	CreateDBLink	支持PolarDB O引擎到到ECS自建PostgreSQL的DBLink。

2020年11月

内核小版本（V1.1.6）

类别	名称	描述	相关文档
新增	兼容性	兼容PostgreSQL 11.9版本。	什么是
	主备切换	默认开启OnlinePromote功能，减少HA切换时间，提高集群可用性。	自动/手动主备切换
	DBLink	支持通过DBLink链接另一个PolarDB O引擎集群下的数据库。	DBLink概述
	修复部分场景下节点重启会出现卡顿的问题。		

类别	名称	描述	相关文档
缺陷修复	修复部分场景下内存溢出导致的集群崩溃的问题。		
	修复部分场景下只读节点崩溃的问题。		
	修复执行 `client_encoding=gbk` 命令时，UTF8编码格式转换为GBK时报错的问题。		
	修复NVL函数中第一个参数为空时报错的问题。		

控制台

类别	名称	描述	相关文档
新增	DBLink	支持创建PolarDB O引擎到PolarDB O引擎的DBLink，实现从前者到后者的跨库数据访问，从而保证不同PolarDB O引擎数据库之间的数据互通。 路径：集群详情页的**配置与管理 > 数据库管理**。 	DBLink概述

API

类别	名称	描述
新增	CreateDBLink	创建PolarDB O引擎的DBLink。
	DescribeDBLinks	查询PolarDB O引擎的DBLink信息。
	DeleteDBLink	删除PolarDB O引擎的DBLink。

2020年10月

内核小版本（V1.1.5）

类别	名称	描述	相关文档
新增	审计日志	支持审计日志中由预编译型SQL语句打印出的日志为形式可执行的SQL语句。	无
	HASH分区	新增指定数量创建HASH分区表语法。	CREATE TABLE ... PARTITION BY HASH
	分区表	新增指定分区或者子分区名称查询指定分区的数据。	查询指定分区
	HOST参数	支持HOST参数，用于获取客户端的HOST值。	无
	视图	优化polar_stat_activity视图，增加query_id字段。	无

优化类别	名称	描述	相关文档
	兼容性	内核小版本升级到11.8，修复多个稳定性缺陷。	无
缺陷修复	修复半同步无法超时的问题。		
	修复CONNECT BY语句与WITH语句联用失败的问题。		

时空数据库（V3.1）

类别	名称	描述	相关文档
新增	Trajectory数据类型	新增Gist索引支持索引轨迹类型，并提供六种不同维度的算子族以支持不同维度的分析需求。	无
		新增时空外包框类型BoxND，可用于时空范围表示以及存储轨迹。	无
		新增对应不同维度的相交(&&)、包含(@>)、被包含(<@)算子。	无
		新增ST_ndIntersects、ST_ndDWithin、ST_ndContains、ST_ndWithin四类轨迹处理函数。	• ST_{Z|T|2D|2DT|3D|3DT}Intersects • ST_{2D|2DT|3D|3DT}DWithin • ST_{T|2D|2DT|3D|3DT}Contains • ST_{T|2D|2DT|3D|3DT}Within
		对轨迹类型提供统计信息收集功能，以及根据统计信息预估扫描代价功能。	无
		提供新的索引方式TrajGist，提供更好的索引选择。	无
	ganos_update函数	新增ganos_update函数，用 `select ganos_update();` 命令可以升级所有的Ganos插件到最新版本。	无
优化	时空范围查询	优化时空范围查询，GiST索引二阶段查询优化，提升查询性能。	无
	矢量金字塔功能	支持任意SRID坐标的源数据，支持3857和4326两种瓦片输出。	无
		新增pixelSize参数设置，对点数据进行聚合，减少瓦片的数量。	无
	修复某些情况下更新Ganos Raster失败问题。		

类别	名称	描述	相关文档
缺陷修复	修复Ganos二进制文件更新到新版本可能出现崩溃的问题。		
	修复用默认参数构建矢量金字塔点数据后，顶级瓦片数据量过大的问题。		

2020年9月

内核小版本（V1.1.4）

类别	名称	描述	相关文档
新增	GROUP_ID函数	新增GROUP_ID()函数。	GROUP_ID
	分区表	支持在分区表上并发创建索引CREATE INDEX CONCURRENTLY。	无
	表索引改为不可见	支持将普通表的索引修改为不可见ALTER INDEX [NAME] INVISIBLE。	无
	ali_decoding插件	支持ali_decoding插件，用于数据库间增量同步DML语句。	无
	pg_cron插件	支持pg_cron插件的定时任务记录查询功能。	PG_CRON
	to_char()	支持to_char()传入字符串。	无
	数据类型	为实现ROWID的prid_类型与text等字符串类型进行比较或text等字符串类型与ROWID的prid_类型进行比较，新增以下操作符： • < • <= • > • >= • = • <>	无
	视图	新增进程状态信息视图。	无
		新增SQL信息视图耗时统计。	无
	审计日志	新增审计日志包含出错SQL。	无
缺陷修复	修复在特定场景下数据页预取可能会导致死锁的问题。		
	修复分区表修剪时JOIN查询不到正确分区的问题。		

时空数据库（V3.0）

类别	名称	描述	相关文档

类别	名称	描述	相关文档
新增	支持具有SubSet的NetCDF数据类型数据	新增支持具有SubSet的NetCDF数据类型数据，可按照指定的图层名称导入。	无
	栅格数据自定义元数据以及时序信息	新增ST_MetaItems函数，获取所有的自定义元数据项目名称。	ST_MetaItems
		新增ST_MetaData函数， 获取自定义元数据项以及返回以JSON方式表达的元数据项。	ST_MetaData
		新增ST_SetMetaData函数，用于设置元数据项。	ST_SetMetaData
		新增ST_BeginDateTime函数，用于获取栅格数据的起始时间。	ST_BeginDateTime
		新增ST_EndDateTime函数，用户获取栅格数据的终止时间。	ST_EndDateTime
		新增ST_SetBeginDateTime函数，用于设置栅格数据的开始时间。	ST_SetBeginDateTime
		新增ST_SetEndDateTime函数，用于设置栅格数据的结束时间。	ST_SetEndDateTime
		新增ST_SetDateTime函数，用于设置栅格数据的开始和结束时间以及波段获取时间。	ST_SetDateTime
	支持栅格数据返回基于流形式的图片格式	新增ST_AsImage函数，用于获取基于流形式的图片格式。	ST_AsImage
		新增ST_AsPng，用于获取基于流形式的PNG图片格式。	ST_AsPNG
		新增ST_AsJPEG，用于获取基于流形式的JPEG图片格式。	ST_AsJPEG
	支持几何网格数据类型以及操作运算	新增GeomGrid数据类型。	GeomGrid SQL参考
		新增ST_AsText函数，用于将网格数据类型转换为文本表示方式。	ST_AsText
		新增ST_AsGeometry函数，用于将网格数据类型转换为几何数据类型。	ST_AsGeometry
		新增ST_AsBinary函数，用于将网格数据类型转换为二进制数据类型。	ST_AsBinary
		新增ST_AsBox函数， 用于将网格数据量类型转换为BOX数据类型。	ST_AsBox
		新增ST_AsGrid函数， 用于计算几何数据类型所对应的几何网格数据。	ST_AsGrid

类别	名称	描述	相关文档
		新增ST_GridFromText函数， 用于将基于文本表示网格转换为几何网格数据类型。	ST_GridFromText
		新增ST_GridFromBinary函数，用于将基于二进制的表示的网格转换为几何网格数据类型。	ST_GridFromBinary
		新增ST_Intersects函数，用于判断栅格数据类型与几何数据类型是否相交。	ST_Intersects
		新增ST_Contains函数，用于判断栅格数据与栅格数据，栅格数据与几何数据是否是包含关系。	ST_Contains
		新增ST_Within函数，用于判断网格数据与网格数据，网格数据与几何数据是否是被包含关系。	ST_Within
	矢量数据类型快速显示存储	新增ST_BuildPyramid函数，用于创建快显支撑数据表。	ST_BuildPyramid
		新增ST_DeletePyramid函数，用于删除快显支撑数据表。	ST_DeletePyramid
		新增ST_Tile函数，用于获取基于MVT格式的矢量数据。	ST_Tile
缺陷修复	修复在某些情况下创建金字塔会出现Out Of Memory的问题。		
	修复移动对象无法创建2000-01-01时间点的问题。		
	修复某些场景下移动对象使用ST_Intersection返回子轨迹错误的问题。		
	修复Ganos升级时会出现奔溃问题。		
	PostGIS兼容性升级到2.5.4。		

2020年8月

内核小版本（V1.1.3）

类别	名称	描述	相关文档
新增	视图	新增SQL执行统计信息视图。	无
		新增QPS统计信息视图。	无
		新增CGroup状态信息视图。	无
	高权限用户	为polar_superuser用户赋予dba_* views的SELECT权限。	无
缺陷修复	修复polar_stat_io_info视图显示错误。		
	修复AWR system wait信息缺失。		
	修复分区表修剪错误问题。		

类别	名称	描述	相关文档
	修复SLRU内存大小计算错误问题。		
	修复回放延迟导致RO重建问题。		

时空数据库（V2.9）

类别	名称	描述	相关文档
新增	COG文件格式支持	新增COG(Cloud Optimize Geotiff) 文件格式支持。支持读取COG文件格式中存储的金字塔信息。	无
	ST_AddZ函数	新增ST_AddZ函数，支持通过栅格数据的像素值为几何对象添加z值。	ST_AddZ
	栅格对象空间范围信息获取增强	新增ST_Extent函数，用于获得栅格对象的空间范围，以BOX形式返回。	ST_Extent
		新增ST_Envelope函数，用于获得栅格对象的空间范围，以几何对象形式返回。	ST_Envelope
		新增ST_ConvexHull函数，用于获得栅格对象的空间范围，以几何对象形式返回。	ST_ConvexHull
		新增ST_Height函数，用于获得栅格对象的像素高度。	ST_Height
		新增ST_Width函数，用于获得栅格对象的像素宽度。	ST_Width
缺陷修复	修复使用外部栅格数据会使用1*n分块导致性能局限性问题，允许用户通过存储选项自定义分块的大小。		
	修复ST_Values函数在查询某些方向的线对象时结果与坐标排序不一致的问题。		
	修复ST_BestPyramidLevel函数在某些情况下会返回负数的问题。		
	修复ST_BuildPyramid函数在某些情况下会重复创建金字塔的问题。		
	修复Truncate栅格表时未能清理对应的块表的问题。		
	修复ST_ExportTo函数对于CreateOption在某些情况下无效的问题。		
	修复ST_ClearChunks函数对于表名存在大小写时会出现错误的问题。		
	修复外部金字塔在某些情况下无法创建overview的问题。		
	修复具有外部金字塔的栅格对象无法创建内部金字塔的问题。		
	修复具有NaN数值的栅格数据在计算统计信息时会导致结果不正确的问题。		

2020年7月

内核小版本（V1.1.2）

类别	名称	描述	相关文档
新增	Pivot语法	Pivot语法支持使用table或subquery作为别名。	无
	ROWID伪列	支持ROWID伪列和ROWIDTOCHAR/CHARTOROWID函数。	无
	内置函数	支持LNNVL，DUMP，SYS_CONTEXT，USERENV，TZ_OFFSET和VSIZE函数。	无
	DBMS_JOB和DBMS_SCHEDULER	支持DBMS_JOB和DBMS_SCHEDULER，用于维护和管理作业队列。	无
	临时表	支持临时表文件写本地存储，大幅提升临时表操作的性能。	无
	触发事件	支持polar_superuser进行触发事件（Event Trigger）的逻辑订阅。	无
	自定义资源隔离	支持自定义资源隔离组发起请求（Request）。	无
优化	崩溃恢复流程	优化崩溃恢复流程，减少极端情况下的崩溃恢复耗时。	无
缺陷修复	修复polar_superuser无法使用索引建议器（index advisor）的问题。		

时空数据库（V2.8）

类别	名称	描述	相关文档
新增	栅格数据元数据访问接口增强	新增ST_XMin函数，用于获取栅格数据X方向最小值。	ST_XMin
		新增ST_YMin函数，用于获取栅格数据Y方向最小值。	ST_YMin
		新增ST_XMax函数，用于获取栅格数据X方向最大值。	ST_XMax
		新增ST_YMax函数，用于获取栅格数据Y方向最大值。	ST_YMax
		新增ST_ChunkHeight函数，用于获取栅格数据分块高度。	ST_ChunkHeight
		新增ST_ChunkWidth函数，用于获取栅格数据分块宽度。	ST_ChunkWidth
		新增ST_ChunkBands函数，用于获取栅格数据分块波段数量。	ST_ChunkBands
	ST_SrFromEsriWkt函数	新增ST_SrFromEsriWkt函数，用于支持Esri格式空间参考字符串转换为OGC格式空间参考字符串。	ST_SrFromEsriWkt
	栅格数据类型	新增栅格数据类型支持Zstd和Snappy压缩方式。	无
	点云数据类型	新增点云数据类型支持二进制拷贝功能。	无
	环境变量	新增支持PROJ_LIB和GDAL_DATA环境变量设置，同时部署相关数据。	无

类别	名称	描述	相关文档
缺陷修复	修复OSS路径非法导致数据库崩溃问题。		
	修复部分栅格数据导入SRID与定义不一致的问题。		

2020年6月

内核小版本（V1.1.1）

类别	名称	描述	相关文档
新增	nvl2函数	增加nvl2函数，根据第一个参数的值是否是NULL或空串，来返回第二或者第三个参数的值。	无
	polar_proxy_utils插件	新增polar_proxy_utils插件，用于管理与proxy相关的功能，主要支持只读UDF和只读表的配置，允许通过集群地址将只读UDF以及只读表的访问路由到只读节点。	无
	polar_resource_group插件	新增polar_resource_group插件，支持自定义资源隔离组，基于user、database、session粒度，通过cpu、memory维度进行资源隔离。	无
优化	可靠性和可用性	数据库计算节点和文件系统解耦，文件系统可独立运维，大幅提高数据库的可靠性和可用性。	无
	事务处理	使用单调递增版本号替代原有的活跃事务列表快照，大幅提升数据库事务处理性能。	无
	执行计划	执行计划优化，避免使用过旧的统计信息。	无
缺陷修复	修复了插件timescaledb在申请内存时出错时进程的异常问题。		
	修复了I/O监控功能中进程退出后没有汇总统计信息。		
	修复了lock_debug开启后，可能发生空指针异常问题。		
	修复了特定情况下导致pg_cron插件不可用的问题。		
	修复了社区已知的DSM死锁问题。		
	修复了用户连接数超限的问题。		

2020年5月

内核小版本（V1.1.0）

类别	名称	描述	相关文档
	TO_SINGLE_BYTE函数	新增TO_SINGLE_BYTE函数，用于将字符串中多字节字符转换为对应的单子节字符。	全角半角转换函数（TO_SINGLE_BYTE和TO_MULTI_BYTE）

类别	名称	描述	相关文档
新增	TO_MULTI_BYTE函数	新增TO_MULTI_BYTE函数，用于将字符串中单子节字符转换为对应的多字节字符。	全角半角转换函数（TO_SINGLE_BYTE和TO_MULTI_BYTE）
	REGEXP_LIKE函数	新增REGEXP_LIKE函数，和LIKE条件表达式类似，但REGEXP_LIKE可以指定POSIX兼容的正则表达式模式。	无
	RATIO_TO_REPORT函数	新增RATIO_TO_REPORT分析函数，用于计算一组值的和中某一个值的占比。	无
	DBMS_UTILITY.FORMAT_ERROR_STACK函数	新增DBMS_UTILITY.FORMAT_ERROR_STACK函数（宏），用于输出逐层的堆栈信息和错误信息。	无
	UTL_I18N.ESCAPE_REFERENCE函数和UTL_I18N.UNESCAPE_REFERENCE函数	新增UTL_I18N.ESCAPE_REFERENCE函数和UTL_I18N.UNESCAPE_REFERENCE函数，用于对HTML或XML字符的转义与反转义。	无
	INSERT ALL语法	新增INSERT ALL语法支持，您可以使用INSERT ALL语句向多张表插入数据。	无
	polar_concurrency_control插件	新增polar_concurrency_control插件，可以对事务执行、SQL查询、存储过程、DML等操作进行并发限制，您可以自定义大查询，并对大查询进行并发限制，优化高并发下的执行性能。	无
	pldebugger插件	新增pldebugger插件，用于存储过程的调试，您可以通过客户端（pgAdmin4 4.19 release及以上版本）调试本地存储过程。	使用pldebugger插件
	oss_fdw插件	新增oss_fdw插件，用于Aliyun OSS外部表支持，您可以通过OSS外部表将数据库数据写入到OSS，也可以通过OSS外部表将OSS数据加载到数据库中，OSS外部表支持并行和压缩，极大提高了导入和导出数据的性能，同时也可以使用这个功能来实现多类型存储介质的冷热数据存储。	使用oss_fdw读写外部数据文本文件
优化	ROUND函数	优化ROUND函数，支持Interval参数类型，返回天数（days）。	ROUND
	polar_stat_activity视图	优化polar_stat_activity视图，新增wait_info列和wait_time列，分别用于监控进程等待对象（pid或fd）的等待时长。	无
	索引	提供插入索引时的索引页预扩展功能，用于提升将数据插入带有索引的表的执行性能。	无

时空数据库（V2.7）

类别	名称	描述	相关文档
新增	MD5操作函数	新增ST_MD5Sum函数，用于获取栅格对象的MD5码值。	ST_MD5Sum
		新增ST_SetMD5Sum函数，用于设置栅格对象的MD5码值。	ST_SetMD5Sum
	空间栅格对象OSS认证方式	新增ST_AKId函数，用于获取以OSS方式存储的栅格对象的AcessKey ID	ST_AKId
		新增ST_SetAccessKey函数，用于设置以OSS方式存储的栅格对象的AcessKey ID和AccessKey Secret。	ST_SetAccessKey
		新增ST_SetAKId函数，用于设置以OSS方式存储的栅格对象的AcessKey ID。	ST_SetAKId
		新增ST_SetAKSecret函数，用于设置以OSS方式存储的栅格对象的AcessKey Secret。	ST_SetAKSecret
	空间栅格元数据操作函数	新增ST_ScaleX函数，用于获取栅格对象在空间参考系下X方向像素宽度。	ST_ScaleX
		新增ST_ScaleY函数，用于获取栅格对象在空间参考系下Y方向像素宽度。	ST_ScaleY
		新增ST_SetScale函数，用于设置栅格对象在空间参考系下像素宽度。	ST_ScaleY
		新增ST_SkewX函数，用于获取栅格对象在空间参考系下X方向旋转。	ST_SkewX
		新增ST_SkewY函数，用于获取栅格对象在空间参考系下Y方向旋转。	ST_SkewY
		新增ST_SetSkew函数，用于设置栅格对象在空间参考系下旋转。	ST_SetSkew
		新增ST_UpperLeftX函数，用于获取栅格对象在空间参考系下左上角点的X坐标。	ST_UpperLeftX
		新增ST_UpperLeftY函数，用于获取栅格对象在空间参考系下左上角点的Y坐标。	ST_UpperLeftY
		新增ST_SetUpperLeft函数，用于获取栅格对象在空间参考系下左上角点坐标。	ST_SetUpperLeft
		新增ST_PixelWidth函数，用于获取栅格对象在空间参考系下像素宽度。	ST_PixelWidth
		新增ST_PixelHeight函数，用于获取栅格对象在空间参考系下像素高度。	ST_PixelHeight
缺陷修复	修复由于聚集函数导致扩展升级失败的问题。		

1.2. 安全公告

1.2.1. Apache Log4j2漏洞对PolarDB无影响

近期，阿里云安全团队向Apache官方报告了Apache Log4j2远程代码执行漏洞（CVE-2021-44228）。云原生关系型数据库PolarDB技术团队确认该漏洞对云原生关系型数据库PolarDB无影响。
更多信息，请参见【漏洞预警】Apache Log4j2 远程代码执行漏洞（CVE-2021-44228）。

2.性能白皮书

PostgreSQL自带一款轻量级的压力测试工具pgbench。pgbench是一种在PostgreSQL（兼容Oracle）上运行基准测试的简单程序，它可以在并发的数据库会话中重复运行相同的SQL命令。本文介绍如何使用pgbench测试PolarDB O引擎集群的最大性能。

测试环境

- 所有测试均在华北1（青岛）地域完成，PolarDB集群和ECS实例在同一可用区。
- ECS的实例规格：ecs.g5.16xlarge（64核 256 GiB）。
- ECS存储规格：SSD本地盘200 GiB。
- 网络类型：专有网络（VPC），PolarDB集群和ECS实例在同一VPC。
- 操作系统：CentOS 7.6 x64。
- PolarDB集群节点数：1个主节点和1个只读节点。

> **说明** CentOS 6不支持PostgreSQL 11。

测试指标

- 只读QPS
  数据库只读时每秒执行的SQL数（仅包含SELECT）。
- 读写QPS
  数据库读写时每秒执行的SQL数（包含INSERT、SELECT、UPDATE）。

准备工作

- 安装测试工具
  执行如下命令在ECS实例中安装PostgreSQL 11。

```
yum install -y https://download.postgresql.org/pub/repos/yum/10/redhat/rhel-7.8-x86_64/pgdg-redhat-repo-42.0-11.noarch.rpm
yum install -y postgresql11
```

- 修改集群参数
  PolarDB集群需要修改的配置参数如下：

```
log_statement = 'none'
enable_hashjoin=off
enable_mergejoin=off
enable_bitmapscan=off
```

> **说明**
> - 目前仅log_statement参数支持在控制台修改。设置集群参数的方式，请参见设置集群参数。
> - 其他参数无法在控制台直接修改，您需要提交工单联系技术支持工程师申请修改。

修改配置参数后，重启集群让配置生效。

测试方法

1. 通过以下命令配置环境变量：

```
export PGHOST=<PolarDB集群地址的私网地址>
export PGPORT=<PolarDB集群地址的私网端口>
export PGDATABASE=postgres
export PGUSER=<PolarDB数据库用户名>
export PGPASSWORD=<PolarDB对应用户的密码>
```

说明 如何查看PolarDB O引擎集群的连接地址，请参见查看连接地址和端口。

2. 根据目标库大小初始化测试数据，具体命令如下：
   - 初始化数据10亿：

     ```
     /usr/pgsql-11/bin/pgbench -i -s 10000
     ```

   - 初始化数据5亿：

     ```
     /usr/pgsql-11/bin/pgbench -i -s 5000
     ```

   - 初始化数据1亿：

     ```
     /usr/pgsql-11/bin/pgbench -i -s 1000
     ```

3. 创建只读和读写的测试脚本。
   - 创建只读脚本*ro.sql*:
     a. 在命令行执行 `vim ro.sql` 命令。
     b. 按I键进入编辑页面。
     c. 在编辑页面输入如下内容：

     ```
     \set aid random_gaussian(1, :range, 10.0)
     SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
     ```

     d. 按ESC键退出编辑，输入 `:wq` 进行保存并退出。
   - 创建读写脚本*rw.sql*:
     a. 在命令行执行 `vim rw.sql` 命令。
     b. 按I键进入编辑页面。
     c. 在编辑页面输入如下内容：

     ```
     \set aid random_gaussian(1, :range, 10.0)
     \set bid random(1, 1 * :scale)
     \set tid random(1, 10 * :scale)
     \set delta random(-5000, 5000)
     BEGIN;
     UPDATE pgbench_accounts SET abalance = abalance + :delta WHERE aid = :aid;
     SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
     UPDATE pgbench_tellers SET tbalance = tbalance + :delta WHERE tid = :tid;
     UPDATE pgbench_branches SET bbalance = bbalance + :delta WHERE bid = :bid;
     INSERT INTO pgbench_history (tid, bid, aid, delta, mtime) VALUES (:tid, :bid, :ai
     d, :delta, CURRENT_TIMESTAMP);
     END;
     ```

     d. 按ESC键退出编辑，输入 `:wq` 进行保存并退出。

4. 在命令行执行如下命令测试：
   - 只读测试：

```
88C 710 GB(polar.o.x8.12xlarge)
总数据量10亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 256 -j 128 -T 120 -D
scale=10000 -D range=100000000
总数据量10亿，热数据5亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 256 -j 128 -T 120 -D
scale=10000 -D range=500000000
总数据量10亿，热数据10亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 256 -j 128 -T 120 -D
scale=10000 -D range=1000000000
64C 512 GB(polar.o.x8.8xlarge)
总数据量10亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 256 -j 128 -T 120 -D
scale=10000 -D range=100000000
总数据量10亿，热数据5亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 256 -j 128 -T 120 -D
scale=10000 -D range=500000000
总数据量10亿，热数据10亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 256 -j 128 -T 120 -D
scale=10000 -D range=1000000000
32C 256 GB(polar.o.x8.4xlarge)
总数据量10亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 128 -j 128 -T 120 -D
scale=10000 -D range=100000000
总数据量10亿，热数据5亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 128 -j 128 -T 120 -D
scale=10000 -D range=500000000
总数据量10亿，热数据10亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 128 -j 128 -T 120 -D
scale=10000 -D range=1000000000
16cC 128 GB(polar.o.x8.2xlarge)
总数据量10亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D sc
ale=10000 -D range=100000000
总数据量10亿，热数据5亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D sc
ale=10000 -D range=500000000
总数据量10亿，热数据10亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D sc
ale=10000 -D range=1000000000
8C 64 GB(polar.o.x8.xlarge)
总数据量10亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 64 -j 32 -T 120 -D sc
ale=10000 -D range=100000000
总数据量10亿，热数据5亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 64 -j 32 -T 120 -D sc
ale=10000 -D range=500000000
总数据量10亿，热数据10亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 64 -j 32 -T 120 -D sc
ale=10000 -D range=1000000000
8C 32 GB(polar.o.x4.xlarge)
```

```
总数据量10亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 64 -j 32 -T 120 -D sc
ale=10000 -D range=100000000
总数据量10亿，热数据5亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 64 -j 32 -T 120 -D sc
ale=10000 -D range=500000000
总数据量10亿，热数据10亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 64 -j 32 -T 120 -D sc
ale=10000 -D range=1000000000
4C 16 GB(polar.o.x4.large)
总数据量5亿，热数据5000万
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 32 -j 32 -T 120 -D sc
ale=5000 -D range=50000000
总数据量5亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 32 -j 32 -T 120 -D sc
ale=5000 -D range=100000000
2C 4 GB(polar.o.x4.medium)
总数据量1亿，热数据5000万
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 16 -j 32 -T 120 -D sc
ale=1000 -D range=50000000
总数据量1亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 16 -j 32 -T 120 -D sc
ale=1000 -D range=100000000
```

- 读写测试：

```
88C 710 GB(polar.o.x8.12xlarge)
总数据量10亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 256 -j 256 -T 120 -D
scale=10000 -D range=100000000
总数据量10亿，热数据5亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 256 -j 256 -T 120 -D
scale=10000 -D range=500000000
总数据量10亿，热数据10亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 256 -j 256 -T 120 -D
scale=10000 -D range=1000000000
64C 512 GB(polar.o.x8.8xlarge)
总数据量10亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 256 -j 256 -T 120 -D
scale=10000 -D range=100000000
总数据量10亿，热数据5亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 256 -j 256 -T 120 -D
scale=10000 -D range=500000000
总数据量10亿，热数据10亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 256 -j 256 -T 120 -D
scale=10000 -D range=1000000000
32C 256 GB(polar.o.x8.4xlarge)
总数据量10亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 128 -j 128 -T 120 -D
scale=10000 -D range=100000000
总数据量10亿，热数据5亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 128 -j 128 -T 120 -D
scale=10000 -D range=500000000
总数据量10亿，热数据10亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 128 -j 128 -T 120 -D
```

```
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 128 -j 128 -T 120 -D
scale=10000 -D range=1000000000
16C 128 GB(polar.o.x8.2xlarge)
总数据量10亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D sc
ale=10000 -D range=100000000
总数据量10亿，热数据5亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D sc
ale=10000 -D range=500000000
总数据量10亿，热数据10亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D sc
ale=10000 -D range=1000000000
8C 64 GB(polar.o.x8.xlarge)
总数据量10亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D sc
ale=10000 -D range=100000000
总数据量10亿，热数据5亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D sc
ale=10000 -D range=500000000
总数据量10亿，热数据10亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D sc
ale=10000 -D range=1000000000
8C 32 GB(polar.o.x4.xlarge)
总数据量10亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D sc
ale=10000 -D range=100000000
总数据量10亿，热数据5亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D sc
ale=10000 -D range=500000000
总数据量10亿，热数据10亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D sc
ale=10000 -D range=1000000000
4C 16 GB(polar.o.x4.large)
总数据量5亿，热数据5000万
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 16 -j 16 -T 120 -D sc
ale=5000 -D range=50000000
总数据量5亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 16 -j 16 -T 120 -D sc
ale=5000 -D range=100000000
2C 8 GB(polar.o.x4.medium)
总数据量1亿，热数据5000万
/usr/pgsql-11/bin/pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 8 -j 8 -T 120 -D scal
e=1000 -D range=50000000
总数据量1亿，热数据1亿
/usr/pgsql-11/bin/pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 8 -j 8 -T 120 -D scal
e=1000 -D range=100000000
```

说明

- scale：该值乘以10万表示测试数据量。
- range：表示活跃数据量。
- -c：表示测试连接数，测试连接数不代表该规格的最大连接数，最大连接数请参考计算节点规格。

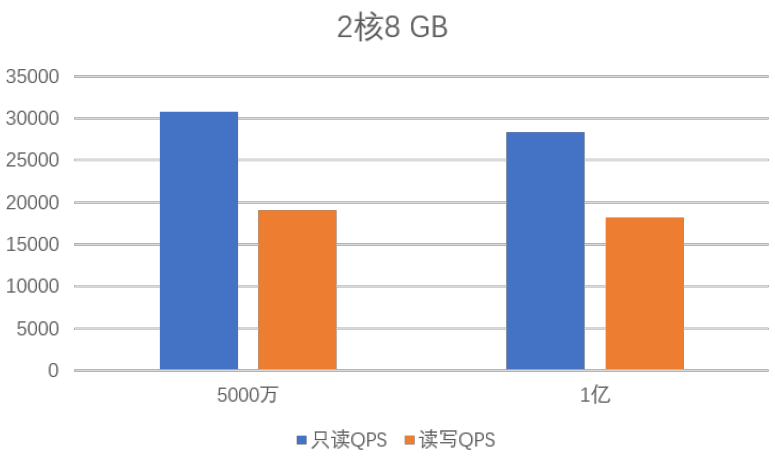
测试结果

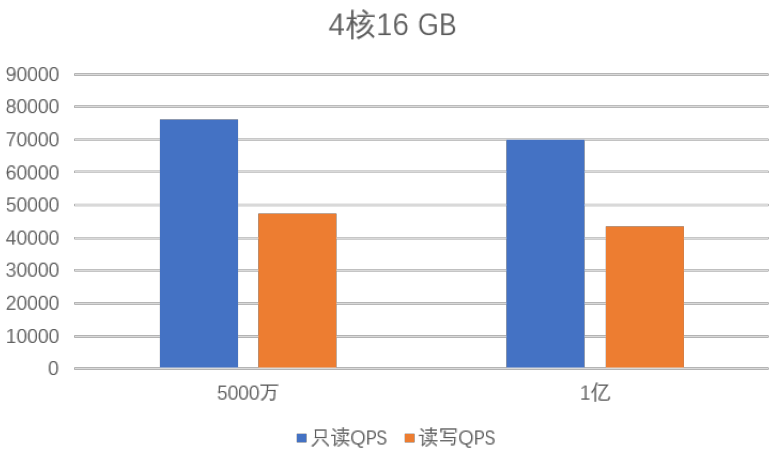
规格	测试数据量	热（活跃）数据量	只读QPS	读写QPS
polar.o.x8.12xlarge 88核 710 GB	10亿	1亿	630650.40	263746.37
		5亿	601262.24	247352.76
		10亿	589405.79	218788.92
polar.o.x8.8xlarge 64核 512 GB	10亿	1亿	592064.65	227352.23
		5亿	554777.11	213343.13
		10亿	508404.07	203331.35
polar.o.x8.4xlarge 32核 256 GB	10亿	1亿	522182.92	213212.34
		5亿	509372.85	203322.34
		10亿	489349.72	198531.37
polar.o.x8.2xlarge 16核 128 GB	10亿	1亿	263893.72	152863.27
		5亿	250221.62	129238.26
		10亿	239832.62	106212.82
polar.o.x8.xlarge 8核 64 GB	10亿	1亿	142836.56	69725.29
		5亿	133931.69	55927.65
		10亿	124151.02	50786.21
polar.o.x4.xlarge 8核 32 GB	10亿	1亿	126748.07	59738.33
		5亿	113432.32	48372.25
		10亿	104232.84	49763.64
polar.o.x4.large 4核 16 GB	5亿	5000万	76238.89	47388.27
		1亿	69892.83	43638.85

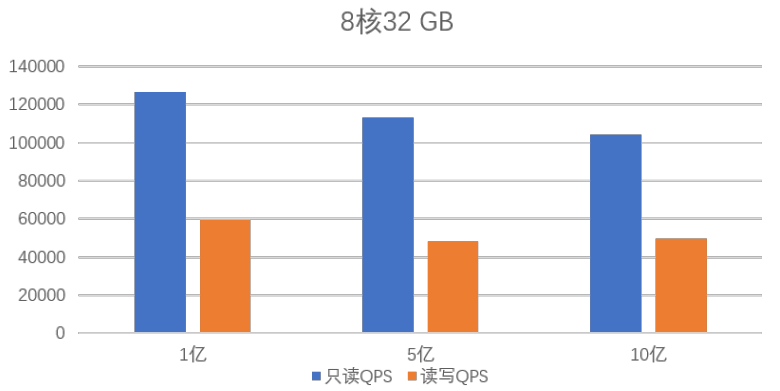
规格	测试数据量	热（活跃）数据量	只读QPS	读写QPS
polar.o.x4.medium 2核 8 GB	1亿	5000万	28320.70	18152.63
		1亿	30792.19	19111.15

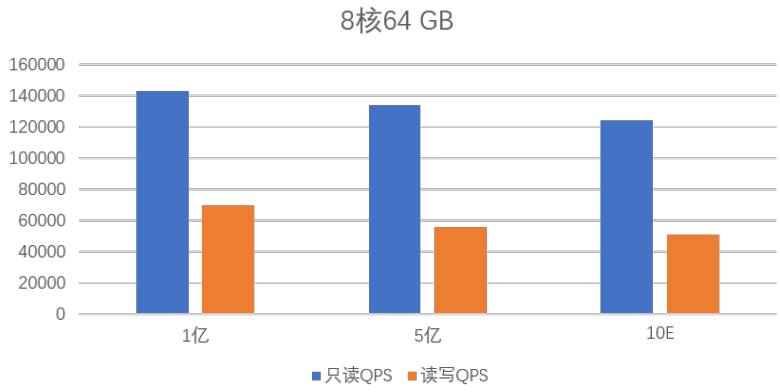
说明

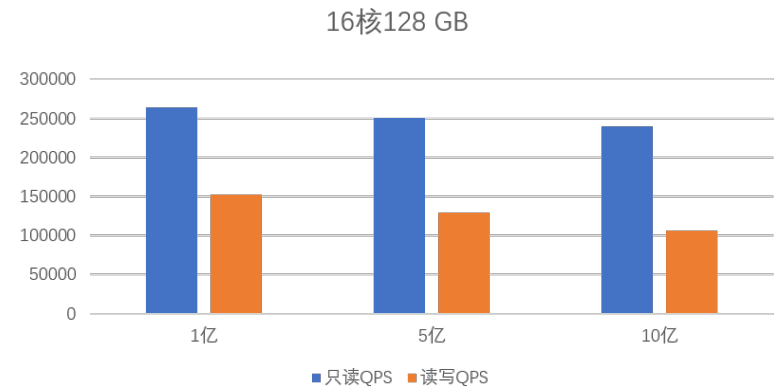
- 规格：PolarDB O引擎的规格代码。
- 测试数据量：本轮测试数据的记录条数。
- 热（活跃）数据量：本轮测试的查询、更新SQL的记录条数。
- 只读QPS：只读测试的结果，表示每秒请求数。
- 读写QPS：读写测试的结果，表示每秒请求数。

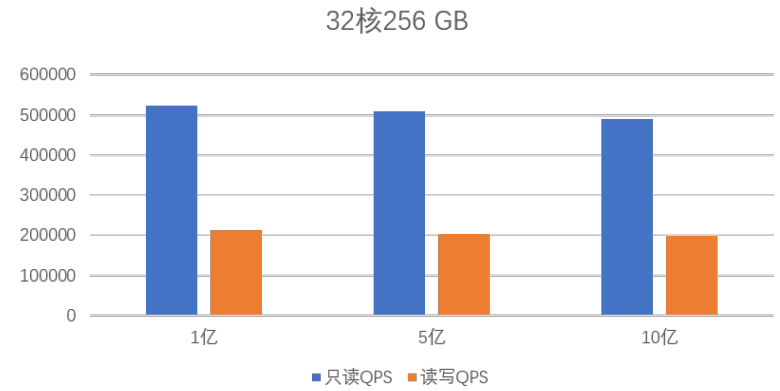


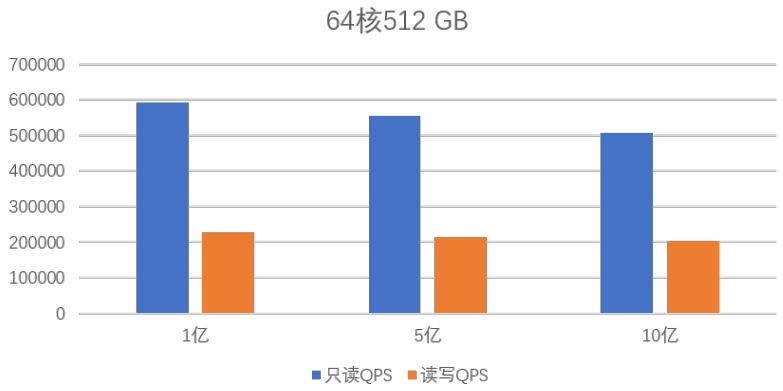


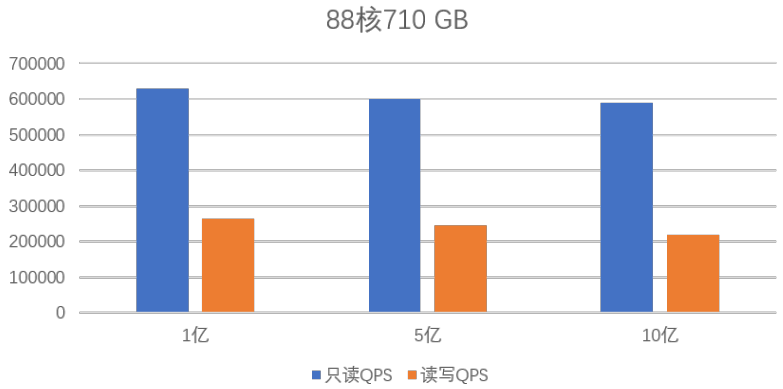

8核32 GB
140000
120000
100000
80000
60000
40000
20000
0
1亿
5亿
10亿
只读QPS
读写QPS


8核64 GB
160000
140000
120000
100000
80000
60000
40000
20000
0
1亿
5亿
10E
只读QPS
读写QPS


16核128 GB
300000
250000
200000
150000
100000
50000
0
1亿
5亿
10亿
只读QPS
读写QPS


32核256 GB
600000
500000
400000
300000
200000
100000
0
1亿
5亿
10亿
只读QPS
读写QPS


64核512 GB
700000
600000
500000
400000
300000
200000
100000
0
1亿
5亿
10亿
只读QPS
读写QPS


88核710 GB
700000
600000
500000
400000
300000
200000
100000
0
1亿
5亿
10亿
只读QPS
读写QPS

3.快速入门

快速入门旨在介绍如何创建PolarDB O引擎集群、进行基本设置以及连接数据库集群，使您能够了解从购买PolarDB O引擎到开始使用的流程。

使用流程

通常，从购买PolarDB O引擎（创建新集群）到可以开始使用，您需要完成如下操作。

1. 步骤一：创建集群
2. 步骤二：设置集群白名单
3. 步骤三：创建数据库账号
4. 步骤四：连接数据库集群
5. 步骤五：创建示例数据库
6. 步骤六：查询数据

步骤一：创建PolarDB O引擎集群

1. 登录PolarDB控制台。
2. 单击页面左上角**创建新集群**。
3. 选择**包年包月**或**按量付费**。

> **说明**
>
> - **包年包月**：在创建集群时支付计算节点（一个主节点和一个只读节点）的费用，而存储空间会根据实际数据量按小时计费，并从账户中按小时扣除。如果您要长期使用该集群，**包年包月**方式更为经济，而且购买时长越长，折扣越多。
> - **按量付费**：无需预先支付费用，计算节点和存储空间（根据实际数据量）均按小时计费，并从账户中按小时扣除。如果您只需短期使用该集群，可以选择**按量付费**，用完即可释放，节省费用。

4. 设置如下参数。

参数	说明
地域	集群所在的地理位置。购买后无法更换地域。 **说明** 请确保PolarDB与需要连接的ECS创建于同一个地域，否则它们无法通过内网互通，只能通过外网互通，无法发挥最佳性能。

参数	说明
创建方式	创建PolarDB集群的方式。 ○ **创建主集群**：创建一个全新的PolarDB集群。 ○ **从回收站恢复**：您可以通过从回收站中恢复已删除集群的备份来创建新集群。 ■ **原版本**：已删除集群的版本。 ■ **已删除集群**：已删除的集群名称。 ■ **历史备份**：选择想要恢复的备份。 **说明** 其他选项用于创建其它引擎的数据库。
主可用区	集群的主可用区。 ○ 可用区是地域中的一个独立物理区域，不同可用区之间没有实质性区别。 ○ 您可以选择将PolarDB与ECS创建在同一可用区或不同的可用区。 ○ 您只需要选择主可用区，系统会自动选择备可用区。
网络类型	固定为**VPC**专有网络，无需选择。
VPC网络 **VPC交换机**	请确保PolarDB与需要连接的ECS创建于同一个VPC，否则它们无法通过内网互通，无法发挥最佳性能。 ○ 如果您已创建符合您网络规划的VPC，直接选择该VPC。例如，如果您已创建ECS，且该ECS所在的VPC符合您的规划，那么选择该VPC。 ○ 如果您未创建符合您网络规划的VPC，您可以使用默认VPC和交换机： ■ 默认VPC： ■ 在您选择的地域中是唯一的。 ■ 网段掩码是16位，如172.31.0.0/16，最多可提供65536个私网IP地址。 ■ 不占用阿里云为您分配的VPC配额。 ■ 默认交换机： ■ 在您选择的可用区中是唯一的。 ■ 网段掩码是20位，如172.16.0.0/20，最多可提供4096个私网IP地址。 ■ 不占用VPC中可创建交换机的配额。 ○ 如果以上默认VPC和交换机无法满足您的要求，您可以自行创建VPC和交换机，详情请参见创建和管理专有网络。

参数	说明
兼容性	○ **MySQL 8.0**（与MySQL 8.0完全兼容），原生支持并行查询，特定场景下性能提升十倍，详情请参见并行查询（Parallel Query）。 ○ **MySQL 5.7**（与MySQL 5.7完全兼容）。 ○ **MySQL 5.6**（与MySQL 5.6完全兼容）。 ○ **PostgreSQL 11**（与PostgreSQL 11 完全兼容）。 ○ **兼容Oracle语法**（高度兼容Oracle语法）。 **说明** 当前华北1（青岛）、美国（弗吉尼亚）、英国（伦敦）和澳大利亚（悉尼）四个地域暂不支持**PostgreSQL 11**和**兼容Oracle语法**。
系列	默认为**集群版（2-16个节点）【推荐】**。
节点规格	按需选择。所有PolarDB节点均为独享型，性能稳定可靠。 更多关于计算节点规格的详情，请参见规格与定价。
节点个数	如果源集群系列为**集群版（2-16个节点）【推荐】**， 系统将默认创建规格相同的两个节点（一主一只读），无需选择。 **说明** 如果主节点故障，系统会自动将只读节点切换为新的主节点，并重新生成一个只读节点，关于只读节点的更多信息，请参见产品架构。
存储费用	无需选择。系统会根据实际数据使用量按小时计费，详情请参见规格与定价。 **说明** 创建集群时无需选择存储容量，存储容量随数据量的增减而自动弹性伸缩。
开启TDE	选择是否开启TDE加密。启用TDE加密后，PolarDB将对集群数据文件进行加密，对于业务访问透明，会有5%~10%的性能损失。 **说明** TDE功能开启后不可关闭。
集群名称	输入集群名称，集群名称需满足如下要求： ○ 不能以 `http://` 或 `https://` 开头。 ○ 长度为2~256个字符。 如果留空，系统将为自动生成一个集群名称，创建集群后还可以修改。
资源组	从已创建资源组中选择一个目标资源组。 **说明** 资源组是在单个云账号下将一组相关资源进行统一管理的容器，一个资源只能归属于一个资源组，详情请参见RAM资源分组与授权。

5. 设置**集群数量**后，单击**立即购买**。

> **说明** 最多可以一次性创建50个集群，适用于游戏批量开服等业务场景。

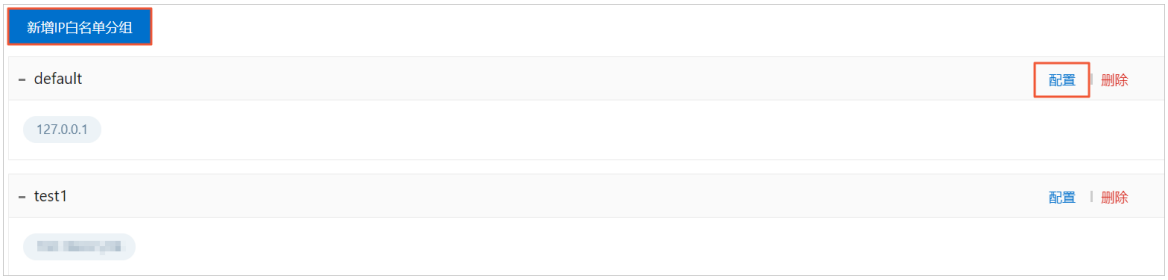
6. 在**确认订单**页面确认订单信息，阅读并选中服务协议，单击**立即开通**即可。

   开通成功后，需要10~15分钟创建集群，之后您就可以在**集群列表**中看到新创建的集群。

> **说明**
> - 当集群中的节点状态为**创建中**时，整个集群可能仍未创建完成，此时集群不可用。只有当集群状态为**运行中**时，集群才可以正常使用。
> - 请确认已选中正确的地域，否则无法看到您创建的集群。
> - 当您的数据量较大时，推荐您购买PolarDB存储包，相比按小时付费，预付费购买存储包有折扣，购买的容量越大，折扣力度就越大，详情请参见。

步骤二：设置集群白名单

1. 登录PolarDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 找到目标集群，单击集群ID。
4. 在左侧导航栏，单击**配置与管理 > 集群白名单**。
5. 在**集群白名单**页面，您可以**新增IP白名单分组**或**配置**已有白名单。



   - 新增白名单分组
     a. 单击**新增IP白名单分组**。

b. 在**新增IP白名单分组**对话框，输入分组名称和允许访问的IP白名单地址。

> **说明** IP白名单分组名称需满足如下要求：
> - 由小写字母、数字、下划线（_）组成。
> - 由字母开头、字母或数字结尾。
> - 长度为2~120个字符。

- 配置白名单

  a. 单击目标IP白名单分组名称右侧的**配置**。

b. 在**配置白名单**对话框，输入允许访问的IP白名单地址。

> **说明**
>
> - 每个集群都默认包含一个 `default` 的白名单分组，且只包含IP地址 `127.0.0.1`，表示任何IP地址均无法访问该数据库集群。
> - 若将IP白名单设置为 `%` 或者 `0.0.0.0/0`，表示允许任何IP地址访问数据库集群。该设置将极大降低数据库的安全性，如非必要请勿使用。

6. 单击**确定**即可。

> **说明** 目前支持创建50个IP白名单，所有IP白名单累积支持添加1000个IP地址或地址段。

步骤三：创建数据库账号

1. 登录PolarDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 找到目标集群，单击集群ID。
4. 在左侧导航栏中，单击**配置与管理 > 账号管理**。
5. 单击**创建账号**。
6. 设置以下参数：

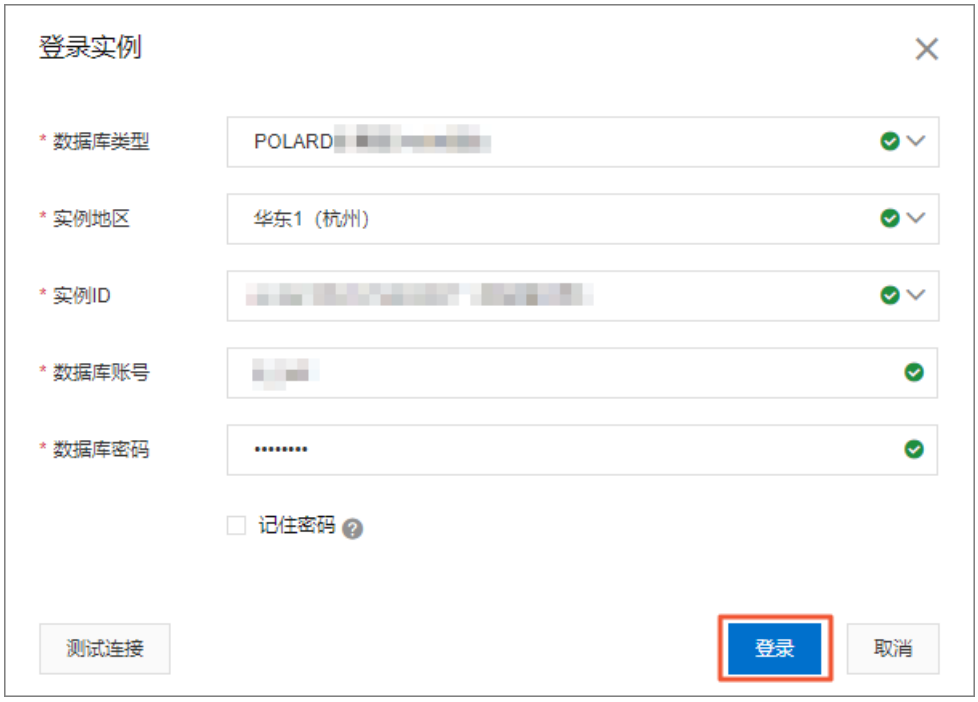
参数	说明
账号名	填写账号名称。要求如下： ○ 以小写字母开头，以字母或数字结尾。 ○ 由小写字母、数字或下划线组成。 ○ 长度为2~16个字符。 ○ 不能使用某些预留的用户名，如root、admin。
账号类型	○ 创建高权限账号，选择**高权限账号**。 ○ 创建普通账号，选择**普通账号**。
密码	设置账号的密码。要求如下： ○ 由大写字母、小写字母、数字或特殊字符组成，至少包含其中三类。 ○ 长度为8~32个字符。 ○ 特殊字符为： !@#$%^&*()_+
确认密码	再次输入密码。
备注	备注该账号的相关信息，便于后续账号管理。要求如下： ○ 不能以http://或https://开头。 ○ 必须以大小写字母或中文开头。 ○ 可以包含大小写字母、中文、数字、下划线（_）或连字符（-）。 ○ 长度为2~256个字符。

7. 单击**确定**。

步骤四：连接数据库集群

通过客户端连接PolarDB O引擎，本文以DMS为例，操作步骤如下：

1. 登录PolarDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 找到目标集群，单击集群ID。
4. 单击**基本信息**页面右上角的**登录数据库**。
5. 在弹出的对话框中，输入PolarDB集群中创建的**数据库账号**和**数据库密码**。



6. 单击**登录**。

> **说明** 如果您是首次使用DMS连接PolarDB集群，系统会提示您授权白名单，单击**确认**后即可完成授权。

7. 登录后刷新DMS页面，在左侧导航栏中，单击**已登录实例**。
8. 找到目标数据库，双击目标数据库即可切换到目标数据库。

更多连接方式请参见连接数据库集群。

步骤五：创建示例数据库

- 以下示例将为您演示如何创建表 `dept` 、表 `emp` 和表 `jobhist` 。

```
CREATE TABLE dept (
    deptno          NUMBER(2) NOT NULL CONSTRAINT dept_pk PRIMARY KEY,
    dname           VARCHAR2(14) CONSTRAINT dept_dname_uq UNIQUE,
    loc             VARCHAR2(13)
);
```

```
CREATE TABLE emp (
    empno           NUMBER(4) NOT NULL CONSTRAINT emp_pk PRIMARY KEY,
    ename           VARCHAR2(10),
    job             VARCHAR2(9),
    mgr             NUMBER(4),
    hiredate        DATE,
    sal             NUMBER(7,2) CONSTRAINT emp_sal_ck CHECK (sal > 0),
    comm            NUMBER(7,2),
    deptno          NUMBER(2) CONSTRAINT emp_ref_dept_fk
                        REFERENCES dept(deptno)
);
```

```
CREATE TABLE jobhist (
    empno           NUMBER(4) NOT NULL,
    startdate       DATE NOT NULL,
    enddate         DATE,
    job             VARCHAR2(9),
    sal             NUMBER(7,2),
    comm            NUMBER(7,2),
    deptno          NUMBER(2),
    chgdesc         VARCHAR2(80),
    CONSTRAINT jobhist_pk PRIMARY KEY (empno, startdate),
    CONSTRAINT jobhist_ref_emp_fk FOREIGN KEY (empno)
        REFERENCES emp(empno) ON DELETE CASCADE,
    CONSTRAINT jobhist_ref_dept_fk FOREIGN KEY (deptno)
        REFERENCES dept (deptno) ON DELETE SET NULL,
    CONSTRAINT jobhist_date_chk CHECK (startdate <= enddate)
);
```

- 以下示例将为您演示如何在表中插入数据。

```
INSERT INTO dept VALUES (10,'ACCOUNTING','NEW YORK');
INSERT INTO dept VALUES (20,'RESEARCH','DALLAS');
INSERT INTO dept VALUES (30,'SALES','CHICAGO');
INSERT INTO dept VALUES (40,'OPERATIONS','BOSTON');
```

```
INSERT INTO emp VALUES (7369,'SMITH','CLERK',7902,'17-DEC-80',800,NULL,20);
INSERT INTO emp VALUES (7499,'ALLEN','SALESMAN',7698,'20-FEB-81',1600,300,30);
INSERT INTO emp VALUES (7521,'WARD','SALESMAN',7698,'22-FEB-81',1250,500,30);
INSERT INTO emp VALUES (7566,'JONES','MANAGER',7839,'02-APR-81',2975,NULL,20);
INSERT INTO emp VALUES (7654,'MARTIN','SALESMAN',7698,'28-SEP-81',1250,1400,30);
INSERT INTO emp VALUES (7698,'BLAKE','MANAGER',7839,'01-MAY-81',2850,NULL,30);
INSERT INTO emp VALUES (7782,'CLARK','MANAGER',7839,'09-JUN-81',2450,NULL,10);
INSERT INTO emp VALUES (7788,'SCOTT','ANALYST',7566,'19-APR-87',3000,NULL,20);
INSERT INTO emp VALUES (7839,'KING','PRESIDENT',NULL,'17-NOV-81',5000,NULL,10);
INSERT INTO emp VALUES (7844,'TURNER','SALESMAN',7698,'08-SEP-81',1500,0,30);
INSERT INTO emp VALUES (7876,'ADAMS','CLERK',7788,'23-MAY-87',1100,NULL,20);
INSERT INTO emp VALUES (7900,'JAMES','CLERK',7698,'03-DEC-81',950,NULL,30);
INSERT INTO emp VALUES (7902,'FORD','ANALYST',7566,'03-DEC-81',3000,NULL,20);
INSERT INTO emp VALUES (7934,'MILLER','CLERK',7782,'23-JAN-82',1300,NULL,10);
```

```
INSERT INTO jobhist VALUES (7369,'17-DEC-80',NULL,'CLERK',800,NULL,20,'New Hire');
INSERT INTO jobhist VALUES (7499,'20-FEB-81',NULL,'SALESMAN',1600,300,30,'New Hire');
INSERT INTO jobhist VALUES (7521,'22-FEB-81',NULL,'SALESMAN',1250,500,30,'New Hire');
INSERT INTO jobhist VALUES (7566,'02-APR-81',NULL,'MANAGER',2975,NULL,20,'New Hire');
INSERT INTO jobhist VALUES (7654,'28-SEP-81',NULL,'SALESMAN',1250,1400,30,'New Hire');
INSERT INTO jobhist VALUES (7698,'01-MAY-81',NULL,'MANAGER',2850,NULL,30,'New Hire');
INSERT INTO jobhist VALUES (7782,'09-JUN-81',NULL,'MANAGER',2450,NULL,10,'New Hire');
INSERT INTO jobhist VALUES (7788,'19-APR-87','12-APR-88','CLERK',1000,NULL,20,'New Hire')
;
INSERT INTO jobhist VALUES (7788,'13-APR-88','04-MAY-89','CLERK',1040,NULL,20,'Raise');
INSERT INTO jobhist VALUES (7788,'05-MAY-90',NULL,'ANALYST',3000,NULL,20,'Promoted to Ana
lyst');
INSERT INTO jobhist VALUES (7839,'17-NOV-81',NULL,'PRESIDENT',5000,NULL,10,'New Hire');
INSERT INTO jobhist VALUES (7844,'08-SEP-81',NULL,'SALESMAN',1500,0,30,'New Hire');
INSERT INTO jobhist VALUES (7876,'23-MAY-87',NULL,'CLERK',1100,NULL,20,'New Hire');
INSERT INTO jobhist VALUES (7900,'03-DEC-81','14-JAN-83','CLERK',950,NULL,10,'New Hire');
INSERT INTO jobhist VALUES (7900,'15-JAN-83',NULL,'CLERK',950,NULL,30,'Changed to Dept 30
');
INSERT INTO jobhist VALUES (7902,'03-DEC-81',NULL,'ANALYST',3000,NULL,20,'New Hire');
INSERT INTO jobhist VALUES (7934,'23-JAN-82',NULL,'CLERK',1300,NULL,10,'New Hire');
```

步骤六：查询数据

您可以使用SELECT语句来查询表，以下示例将查询 `emp` 表中所有的列。

```
SELECT * FROM emp;
 empno | ename  |    job    | mgr  |      hiredate      |   sal   |  comm   | deptno
-------+--------+-----------+------+--------------------+---------+---------+--------
  7369 | SMITH  | CLERK     | 7902 | 17-DEC-80 00:00:00 |  800.00 |         |     20
  7499 | ALLEN  | SALESMAN  | 7698 | 20-FEB-81 00:00:00 | 1600.00 |  300.00 |     30
  7521 | WARD   | SALESMAN  | 7698 | 22-FEB-81 00:00:00 | 1250.00 |  500.00 |     30
  7566 | JONES  | MANAGER   | 7839 | 02-APR-81 00:00:00 | 2975.00 |         |     20
  7654 | MARTIN | SALESMAN  | 7698 | 28-SEP-81 00:00:00 | 1250.00 | 1400.00 |     30
  7698 | BLAKE  | MANAGER   | 7839 | 01-MAY-81 00:00:00 | 2850.00 |         |     30
  7782 | CLARK  | MANAGER   | 7839 | 09-JUN-81 00:00:00 | 2450.00 |         |     10
  7788 | SCOTT  | ANALYST   | 7566 | 19-APR-87 00:00:00 | 3000.00 |         |     20
  7839 | KING   | PRESIDENT |      | 17-NOV-81 00:00:00 | 5000.00 |         |     10
  7844 | TURNER | SALESMAN  | 7698 | 08-SEP-81 00:00:00 | 1500.00 |    0.00 |     30
  7876 | ADAMS  | CLERK     | 7788 | 23-MAY-87 00:00:00 | 1100.00 |         |     20
  7900 | JAMES  | CLERK     | 7698 | 03-DEC-81 00:00:00 |  950.00 |         |     30
  7902 | FORD   | ANALYST   | 7566 | 03-DEC-81 00:00:00 | 3000.00 |         |     20
  7934 | MILLER | CLERK     | 7782 | 23-JAN-82 00:00:00 | 1300.00 |         |     10
(14 rows)
```

如果您需要查询指定的列，可以执行如下命令：

```
SELECT ename, sal, sal * 24 AS yearly_salary, deptno FROM emp;
 ename  |   sal    | yearly_salary | deptno
--------+---------+---------------+--------
 SMITH  |  800.00 |      19200.00 |     20
 ALLEN  | 1600.00 |      38400.00 |     30
 WARD   | 1250.00 |      30000.00 |     30
 JONES  | 2975.00 |      71400.00 |     20
 MARTIN | 1250.00 |      30000.00 |     30
 BLAKE  | 2850.00 |      68400.00 |     30
 CLARK  | 2450.00 |      58800.00 |     10
 SCOTT  | 3000.00 |      72000.00 |     20
 KING   | 5000.00 |     120000.00 |     10
 TURNER | 1500.00 |      36000.00 |     30
 ADAMS  | 1100.00 |      26400.00 |     20
 JAMES  |  950.00 |      22800.00 |     30
 FORD   | 3000.00 |      72000.00 |     20
 MILLER | 1300.00 |      31200.00 |     10
(14 rows)
```

如果您要查询部门20中工资超过1000的员工，可以执行如下命令：

```
SELECT ename, sal, deptno FROM emp WHERE deptno = 20 AND sal > 1000;
 ename |   sal    | deptno
-------+---------+--------
 JONES | 2975.00 |     20
 SCOTT | 3000.00 |     20
 ADAMS | 1100.00 |     20
 FORD  | 3000.00 |     20
(4 rows)
```

请求按排序顺序返回查询结果，例如按姓名顺序返回查询结果，可以执行如下命令：

```
SELECT ename, sal, deptno FROM emp ORDER BY ename;
 ename  |   sal    | deptno
--------+---------+--------
 ADAMS  | 1100.00 |     20
 ALLEN  | 1600.00 |     30
 BLAKE  | 2850.00 |     30
 CLARK  | 2450.00 |     10
 FORD   | 3000.00 |     20
 JAMES  |  950.00 |     30
 JONES  | 2975.00 |     20
 KING   | 5000.00 |     10
 MARTIN | 1250.00 |     30
 MILLER | 1300.00 |     10
 SCOTT  | 3000.00 |     20
 SMITH  |  800.00 |     20
 TURNER | 1500.00 |     30
 WARD   | 1250.00 |     30
(14 rows)
```

如果您需要查询某一列中不同的值，例如查询 `job` 列中有哪些职业，可以使用如下命令：

```
SELECT DISTINCT job FROM emp;
    job
-----------
 ANALYST
 CLERK
 MANAGER
 PRESIDENT
 SALESMAN
(5 rows)
```

4.数据迁移/同步

4.1. 数据迁移和同步方案概览

云数据库PolarDB提供了多种数据迁移方案，可满足不同上云、迁云的业务需求，使您可以在不影响业务的情况下平滑将数据库迁移至云数据库PolarDB。

数据迁移

使用场景	文档链接
从自建数据库迁移至PolarDB	• 从自建Oracle迁移至PolarDB-O集群（迁移结构） • 从自建Oracle迁移至PolarDB-O集群（迁移数据）

数据同步

使用场景	文档链接
PolarDB兼容Oracle数据间同步	PolarDB O引擎集群间的单向同步

4.2. 从自建Oracle迁移至PolarDB-O集群（迁移结构）

本文介绍如何使用ADAM将自建Oracle数据库的结构迁移至PolarDB-O集群，然后使用DTS进行数据迁移。

前提条件

- 自建Oracle数据库的版本为9i、10g、11g、12c、18c或19c版本。
- 自建Oracle数据库已开启ARCHIVELOG（归档模式），设置合理的归档日志保持周期且归档日志能够被访问，详情请参见ARCHIVELOG。
- 自建Oracle数据库已开启Supplemental Logging，且已开启supplemental_log_data_pk，supplemental_log_data_ui，详情请参见Supplemental Logging。
- 自建Oracle数据库中的待迁移表需具备主键或非空唯一索引。
- 如果您的自建Oracle版本为12c及以上，待迁移表的名称长度需不超过30个字节。
- 如PolarDB-O集群地区为北京、杭州、张家口、深圳、上海以外的地域，PolarDB-O集群需申请公网连接地址，申请方式，请参见申请连接地址。
- PolarDB-O集群（简称PolarDB集群）的 `polar_comp_redwood_raw_names` 参数已设置为*off*，详情请参见设置集群参数。

步骤一：采集数据库结构信息

ADAM目前提供下述两种采集方式供您选择。

采集方式	适用场景	文档链接

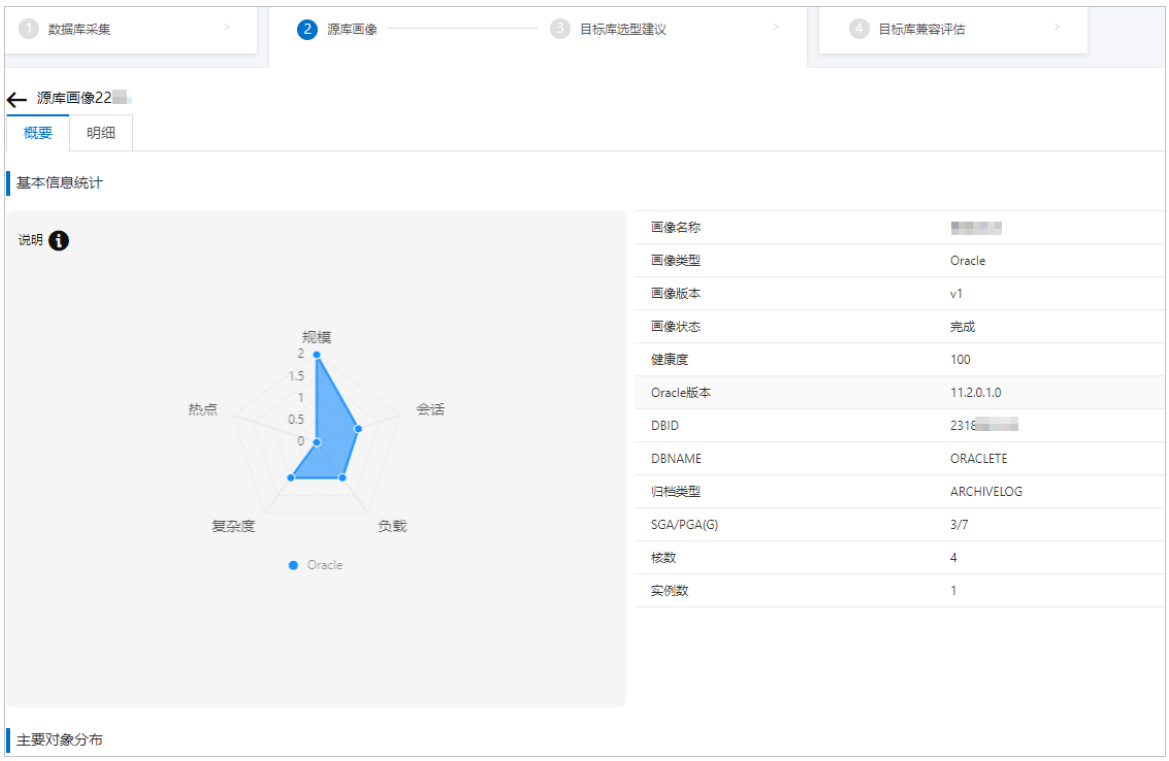
采集方式	适用场景	文档链接
在线数据库采集	• 源库具有公网IP，可通过外网访问。 • 源库不具有公网IP，可通过数据库网关DG连接。	在线采集
部署采集器采集	源库无法连通公网。	下载采集器并采集数据

步骤二：分析数据库画像

1. 登录ADAM控制台，单击左侧导航栏的**数据库迁云 > 数据库评估**。
2. 在**数据库采集**页签的底部，单击**下一步查看源库画像**。
3. 单击**新建画像**，设置画像的基本信息并单击**创建**。

配置	说明
画像名	为画像配置具有业务意义的名称便于后续识别。
类型	固定为Oracle。
报告语言类型	根据需求选择为**中文**或**英文**。
数据文件	i. 单击**上传**按钮。 ii. 选择在步骤一：采集数据库结构信息中采集结果所输出的压缩包文件。

4. 返回画像列表，画像分析完成后，您可以单击**操作**列的**详情**，从数据库的规模、会话、风险、热点、复杂度、负载等维度分析评估数据库的具体信息。



> **说明**
> - 更多关于数据库画像的分析解读，请参见数据库画像分析。
> - 分析完成后，您还可以单击页面下方的**下一步查看目标库选型建议**，了解迁移到各种目标库的兼容情况分析，详情请参见目标库选型建议。

步骤三：目标库兼容性评估

1. 在ADAM控制台，单击左侧导航栏的**数据库迁云 > 数据库评估**。
2. 单击**目标库兼容评估**页签。
3. 单击**新建项目**，设置项目的基本信息并单击**创建**。

配置	说明
项目名	为项目配置具有业务意义的名称便于后续识别。
项目类型	选择为**Oracle_To_POLARDB O**。
目标库版本	选择为**POLARDB O**。
报告语言	根据需求选择为**中文**或**英文**。
源库画像	选择在步骤二中创建的源库画像。
LOB数据迁移到OSS	是否需要将LOB（Large Object）数据迁移到对象存储服务OSS（Object Storage Service）中，根据需求选择为**是**或**否**。
选择评估Schema	根据业务需求，选择待评估的Schema信息。 i. 在**未选择**区域框中，勾选待评估的Schema。 ii. 单击 > 将其添加至**已选择**区域框。

该项目将创建成功并进入**处理中**状态。

4. 等待项目评估完成，单击**操作**列的**详情**，全面评估目标库的兼容性、规格与迁移风险，了解ADAM提供相应的解决方案。

> **说明** 更多关于评估结果的分析解读，请参见数据库评估。您也可以使用数据库专家服务，由阿里云数据库专家为您提供应用架构和数据库架构治理、评估选型、制定数据库平滑迁云迁移方案。

步骤四：数据库改造迁移

改造类型	准备工作	具体步骤

改造类型	准备工作	具体步骤
线上改造迁移	• 确认账号权限： ○ 如果您当前使用的是主账号请忽略此步骤。 ○ 如果是主账号下属的RAM子账号： ■ 使用主账号登录并进入数据库改造迁移页面。 ■ 授权子账号，子账号将自动继承授权信息。 • 将ADAM服务器的IP白名单添加至PolarDB-O集群中。IP白名单，请参见配置白名单，添加方式，请参见设置集群白名单 ○ 如ADAM与目标数据库在同一地域，需添加VPC网络白名单。 ○ 如ADAM与目标数据库不在同一地域，需添加公网白名单。	详情请参见线上改造迁移。

警告 线上改造仅执行结构迁移，在迁移过程中可评估相关的提示信息，然后执行结构订正以获得更好的兼容性。

后续步骤

从自建Oracle迁移至PolarDB O引擎集群（迁移数据）

4.3. 从自建Oracle迁移至PolarDB-O集群（迁移数据）

本文介绍使用DTS迁移自建Oracle的数据至PolarDB-O集群。

前提条件

已使用ADAM完成结构迁移，详情请参见从自建Oracle迁移至PolarDB O引擎集群（迁移结构）中的步骤一至步骤四。

费用说明

仅在增量数据迁移正常运行期间计费（包含增量数据迁移暂停期间），详情请参见产品定价。

注意事项

DTS在执行全量数据迁移时将占用源库和目标库一定的读写资源，可能会导致数据库服务器负载上升。因此您需要在执行数据迁移前评估数据迁移对源库和目标库性能的影响，同时建议您在业务低峰期执行数据迁移。

准备工作

登录待迁移的Oracle数据库，创建用于采集数据的账号并授权。

说明 如您已创建包含下述权限的账号，可跳过本步骤。

数据库	结构迁移	全量迁移	增量数据迁移
自建Oracle数据库	schema的owner权限	schema的owner权限	DBA

数据库	结构迁移	全量迁移	增量数据迁移
PolarDB集群	schema的owner权限	schema的owner权限	schema的owner权限

- 自建Oracle数据库请参见CREATE USER和GRANT。
- PolarDB集群请参见创建账号。

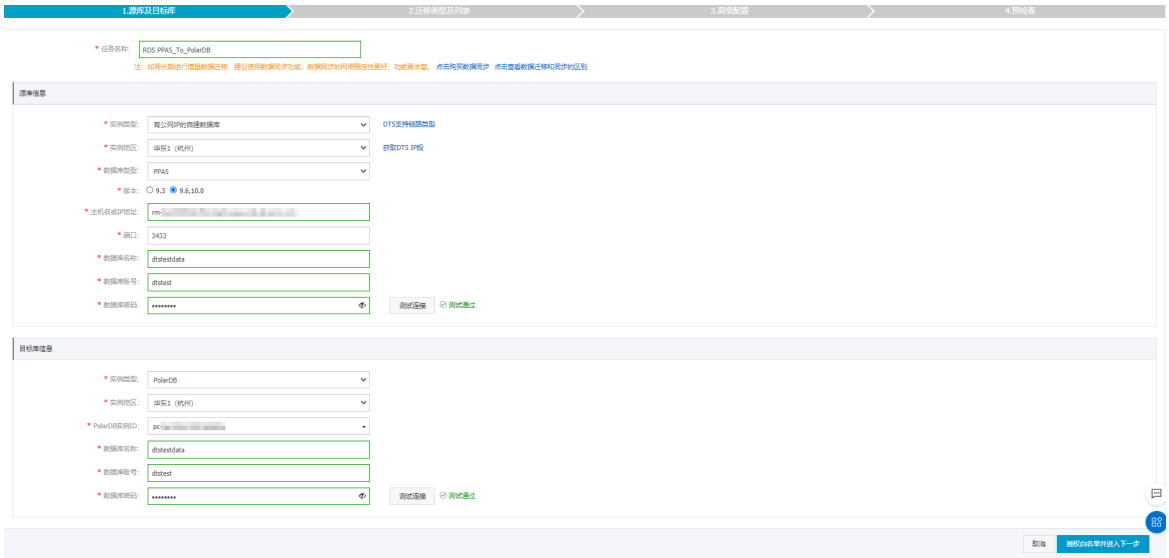
> **注意** 如需执行增量数据迁移，且不允许授予DBA权限，您可以参照以下内容为数据库账号授予更精细化的权限。

需授予的精细化权限（单击展开）

授权命令示例（单击展开）

操作步骤

1. 登录数据传输控制台，单击左侧导航栏的**数据迁移**。
2. 在**迁移任务列表**页面顶部，选择迁移的目标集群所属地域。
3. 单击页面右上角的**创建迁移任务**。
4. 配置迁移任务的源库和目标库连接信息。



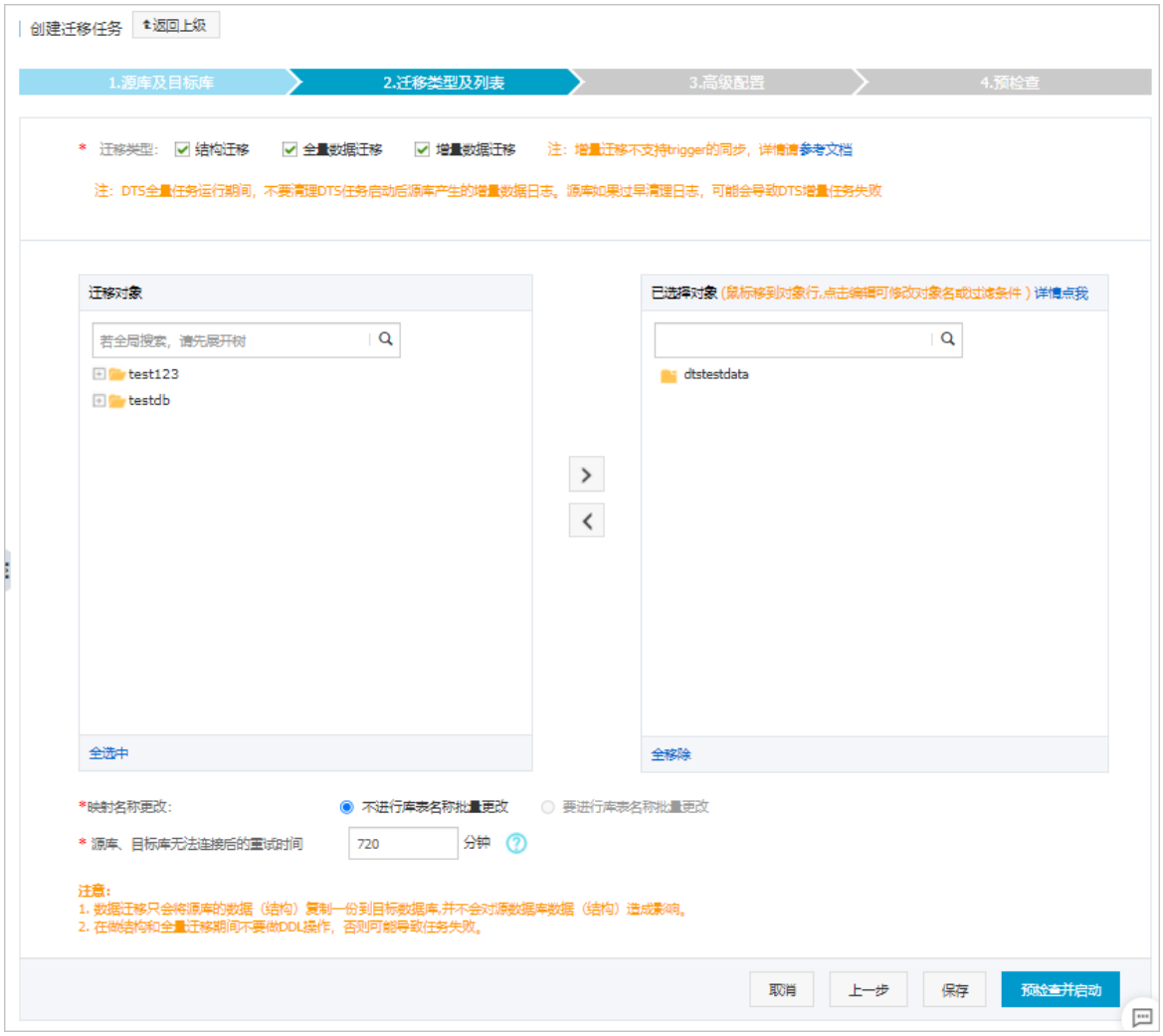
类别	配置	说明
无	任务名称	DTS会自动生成一个任务名称，建议配置具有业务意义的名称（无唯一性要求），便于后续识别。
	实例类型	根据源库的部署位置进行选择，本文以**有公网IP的自建数据库**为例介绍配置流程。 **说明** 当自建数据库为其他实例类型时，您还需要执行相应的准备工作，详情请参见准备工作概览。

类别	配置	说明
源库信息	实例地区	当实例类型选择为**有公网IP的自建数据库**时，**实例地区**无需设置。 **说明** 如果您的自建Oracle数据库具有白名单安全设置，您需要在**实例地区**配置项后，单击**获取DTS IP段**来获取到DTS服务器的IP地址段，并将获取到的IP地址段加入自建Oracle数据库的白名单安全设置中。
	数据库类型	选择**Oracle**。
	主机名或IP地址	填入自建Oracle数据库的访问地址。
	端口	填入自建Oracle数据库的服务端口，默认为**1521**。 **说明** 本案例中，该服务端口需开放至公网。
	实例类型	○ **非RAC实例**：选择该项后，您需要填写**SID**信息。 ○ **RAC实例**：选择该项后，您需要填写**ServiceName**信息。
	数据库账号	填入自建Oracle的数据库账号，权限要求请参见准备工作。
	数据库密码	填入该数据库账号对应的密码。 **说明** 源库信息填写完毕后，您可以单击**数据库密码**后的**测试连接**来验证填入的源库信息是否正确。源库信息填写正确则提示**测试通过**；如果提示**测试失败**，单击**测试失败**后的**诊断**，根据提示调整填写的源库信息。
目标库信息	实例类型	选择**PolarDB**。
	实例地区	选择目标PolarDB集群所属地域。
	PolarDB实例ID	选择目标PolarDB集群ID。
	数据库名称	填入待迁入数据的数据库名称。
	数据库账号	填入目标PolarDB集群的数据库账号，权限要求请参见准备工作。
	数据库密码	填入该数据库账号对应的密码。 **说明** 目标库信息填写完毕后，您可以单击**数据库密码**后的**测试连接**来验证填入的目标库信息是否正确。目标库信息填写正确则提示**测试通过**；如果提示**测试失败**，单击**测试失败**后的**诊断**，根据提示调整填写的目标库信息。

5. 配置完成后，单击页面右下角的**授权白名单并进入下一步**。

> **警告**
>
> - 如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见DTS服务器的IP地址段；如果源或目标数据库是IDC自建数据库或其他云数据库，则需要您手动添加对应地区DTS服务的IP地址，以允许来自DTS服务器的访问。
> - 上述场景中，DTS自动添加或您手动添加DTS服务的公网IP地址段可能会存在安全风险，一旦使用本产品代表您已理解和确认其中可能存在的安全风险，并且需要您做好基本的安全防护，包括但不限于加强账号密码强度防范、限制各网段开放的端口号、内部各API使用鉴权方式通信、定期检查并限制不需要的网段，或者使用通过内网（专线/VPN网关/智能网关）的方式接入。
> - DTS任务完成或释放后，建议您手动检测并删除DTS相关的服务器IP地址段。

6. 选择迁移类型和迁移对象。



配置	说明

配置	说明
迁移类型	由于已经在从自建Oracle迁移至PolarDB O引擎集群（迁移结构）中完成了结构迁移，此处无需勾选结构迁移。 ○ 如果只需要全量迁移，仅勾选**全量数据迁移**。 ○ 如果需要不停机迁移，同时勾选**全量数据迁移**和**增量数据迁移**。 说明 ○ 如果未勾选**增量数据迁移**，为保障数据一致性，数据迁移期间请勿在自建Oracle数据库中写入新的数据。 ○ 增量数据迁移阶段支持同步的SQL操作为：INSERT、UPDATE、DELETE。
迁移对象	在**迁移对象**框中选择待迁移的对象（对象的结构须已通过ADAM迁移至PolarDB集群），然后单击 › 图标将其移动至**已选择对象**框。 警告 由于已使用ADAM完成了结构迁移，请勿执行对象名映射操作，否则将导致迁移失败。

7. 完成预检查并启动迁移任务。
   i. 单击页面右下角的**预检查并启动**。

   说明
   - 在迁移任务正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动迁移任务。
   - 如果预检查失败，单击具体检查项后的 ⓘ，查看失败详情。
     - 您可以根据提示修复后重新进行预检查。
     - 如无需修复告警检测项，您也可以选择**确认屏蔽**、**忽略告警项并重新进行预检查**，跳过告警检测项重新进行预检查。

   ii. 预检查通过后，单击**下一步**。
   iii. 在弹出的**购买配置确认**对话框，选择**链路规格**并勾选**数据传输（按量付费）服务条款**。
   iv. 单击**购买并启动**，迁移任务正式开始。
8. 迁移完成后，结束数据迁移任务。

   警告 为尽可能地减少数据迁移对业务的影响，建议按照推荐的方法切换业务并建立回退方案（将PolarDB集群的增量数据实时迁移回源Oracle数据库），详情请参见业务切换流程。如果无需切换业务，可结束迁移任务。

   ○ 全量数据迁移

请勿手动结束迁移任务，否则可能导致数据不完整。您只需等待迁移任务完成即可，迁移任务会自动结束。

- 增量数据迁移
  迁移任务不会自动结束，您需要手动结束迁移任务。
  a. 观察迁移任务的进度变更为**增量迁移**，并显示为**无延迟**状态时，将源库停写几分钟，此时**增量迁移**的状态可能会显示延迟的时间。
  b. 等待迁移任务的**增量迁移**再次进入**无延迟**状态后，手动结束迁移任务。

4.4. PolarDB O引擎集群间的单向同步

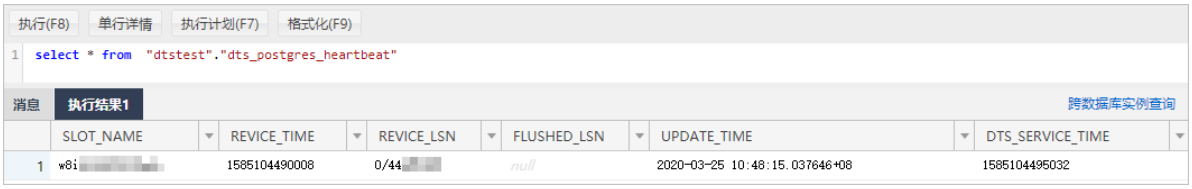
云原生关系型数据库PolarDB O引擎是阿里巴巴自主研发的下一代云原生关系型数据库，高度兼容Oracle语法。通过数据传输服务DTS（Data Transmission Service），您可以实现PolarDB O引擎集群间的单向数据同步。

前提条件

- 源PolarDB O引擎集群需为最新版本，升级方式详情请见版本管理。
- 源PolarDB O引擎集群中，待同步的表需具备主键或非空唯一索引。
- 源PolarDB O引擎集群中，wal_level参数的值需设置为*logical*，即在预写式日志WAL（Write-ahead logging）中增加支持逻辑编码所需的信息。详情请参见设置集群参数。

注意事项

- DTS在执行全量数据初始化时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据同步前评估源库和目标库的性能，同时建议您在业务低峰期执行数据同步（例如源库和目标库的CPU负载在30%以下）。
- 一个数据同步作业只能同步一个数据库，如有多个数据库需要同步，您需要为每个数据库配置数据同步作业。
- 为保障同步延迟时间展示的准确性，DTS会在源库中新增一个表，表名为 `dts_postgres_heartbeat` ，结构及内容如下图所示。



- 在数据同步过程中，如果同步对象的选择粒度为Schema，在待同步的Schema中创建了新的表或使用RENAME命令重建了待同步的表，您需要在对该表写入数据前执行 `ALTER TABLE schema.table REPLICA IDENTITY FULL;` 命令。

> **说明** 将上述命令中的 `schema` 和 `table` 替换成真实的Schema名和表名。

支持同步的SQL操作

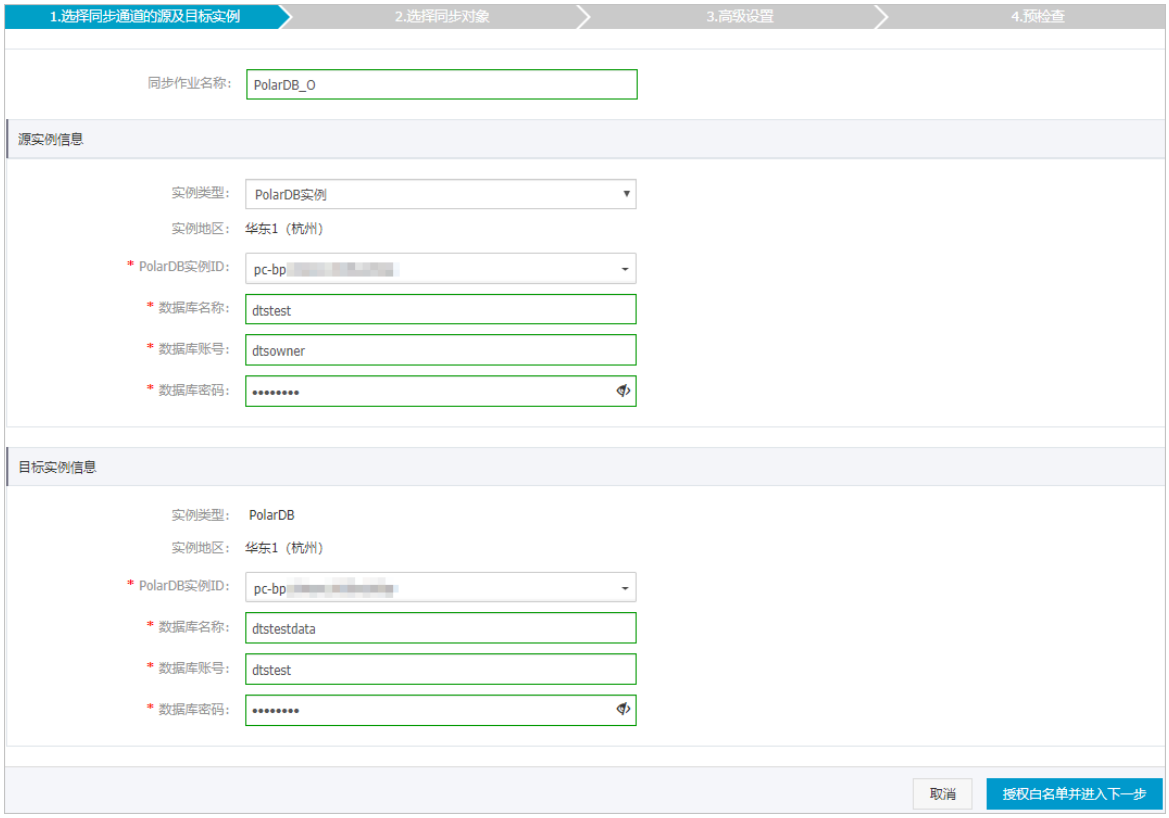
INSERT、UPDATE、DELETE

操作步骤

1. 购买数据同步作业，详情请参见购买流程。

> **说明** 购买时，源实例和目标实例均选择为POLARDB，并选择同步拓扑为**单向同步**。

2. 登录数据传输控制台。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。
5. 定位至已购买的数据同步实例，单击**配置同步链路**。
6. 配置同步通道的源实例及目标实例信息。



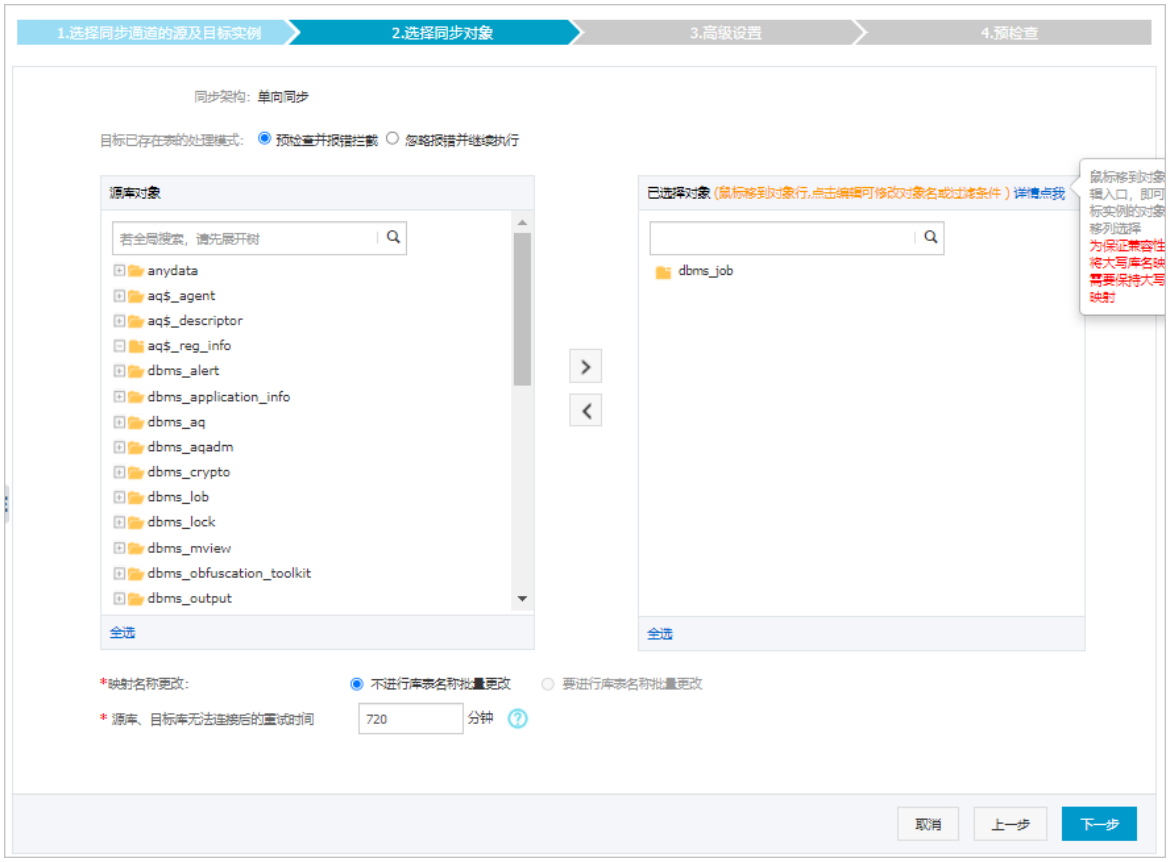
类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，建议配置具有业务意义的名称（无唯一性要求），便于后续识别。
	实例类型	固定为**PolarDB实例**，不可变更。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	PolarDB实例ID	选择源PolarDB集群ID。

源实例信息类别	配置	说明
	数据库名称	填入待同步的数据库名称。
	数据库账号	填入源PolarDB集群的高权限账号。数据库账号创建方法请参见创建数据库账号。
	数据库密码	填入数据库账号对应的密码。
目标实例信息	实例类型	固定为**PolarDB**，不可变更。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	PolarDB实例ID	选择目标PolarDB集群ID。
	数据库名称	填入同步的目标数据库名称。
	数据库账号	填入目标PolarDB集群的数据库账号，该账号需具备**数据库Owner**权限。 **注意** **数据库Owner**在创建数据库时已指定。
	数据库密码	填入数据库账号对应的密码。

7. 单击页面右下角的**授权白名单并进入下一步**。

> **说明** 此步骤会将DTS服务器的IP地址自动添加到源和目标PolarDB集群的白名单中，用于保障DTS服务器能够正常连接源和目标集群。

8. 配置目标已存在表的处理模式和同步对象。



配置项目	配置说明
目标已存在表的处理模式	○ **预检查并报错拦截**：检查目标数据库中是否有同名的表。如果目标数据库中没有同名的表，则通过该检查项目；如果目标数据库中有同名的表，则在预检查阶段提示错误，数据同步作业不会被启动。 **说明** 如果目标库中同名的表不方便删除或重命名，您可以设置同步对象在目标实例中的名称来避免表名冲突。 ○ **忽略报错并继续执行**：跳过目标数据库中是否有同名表的检查项。 **警告** 选择为**忽略报错并继续执行**，可能导致数据不一致，给业务带来风险，例如： ■ 表结构一致的情况下，如果在目标库遇到与源库主键的值相同的记录，在初始化阶段会保留目标库中的该条记录；在增量同步阶段则会覆盖目标库的该条记录。 ■ 表结构不一致的情况下，可能会导致无法初始化数据、只能同步部分列的数据或同步失败。

配置项目	配置说明
选择同步对象	在**源库对象**框中单击待同步的对象，然后单击 > 图标将其移动至**已选择对象**框。 同步对象的选择粒度为库、表。 **说明** ○ 如果选择整个库作为同步对象，那么该库中所有对象的结构变更操作都会同步至目标库。 ○ 默认情况下，同步对象的名称保持不变。如果您需要改变同步对象在目标集群中的名称，请使用对象名映射功能，详情请参见设置同步对象在目标实例中的名称。
映射名称更改	如需更改同步对象在目标实例中的名称，请使用对象名映射功能，详情请参见库表列映射。
源、目标库无法连接重试时间	当源、目标库无法连接时，DTS默认重试720分钟（即12小时），您也可以自定义重试时间。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。 **说明** 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

9. 上述配置完成后，单击页面右下角的**下一步**。
10. 配置同步初始化。

初始化类型	说明
结构初始化	DTS将同步对象的结构定义同步到目标PolarDB集群。目前支持的对象包括：表、视图、同义词、触发器、存储过程、存储函数、包、自定义类型。 **注意** 当同步对象包含触发器时可能导致数据不一致，解决方案请参见源库存在触发器时如何配置同步作业。

初始化类型	说明
全量数据初始化	DTS会将源库中同步对象的存量数据，全部同步至目标PolarDB集群。 **注意** 在同步启动运行后，请勿对同步对象执行DDL操作，否则可能导致同步失败。

11. 上述配置完成后，单击页面右下角的**预检查并启动**。

> **说明**
> - 在同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动同步作业。
> - 如果预检查失败，单击具体检查项后的 ⓘ，查看失败详情。
> - 您可以根据提示修复后重新进行预检查。
> - 如无需修复告警检测项，您也可以选择**确认屏蔽**、**忽略告警项并重新进行预检查**，跳过告警检测项重新进行预检查。

12. 在**预检查**对话框中显示**预检查通过**后，关闭**预检查**对话框，同步作业将正式开始。
13. 等待同步作业的链路初始化完成，直至处于**同步中**状态。

    您可以在**数据同步**页面，查看数据同步作业的状态。

5.常见问题

本文介绍PolarDB O引擎的常见问题和解答。

基本问题

- Q：什么是PolarDB?
  A：PolarDB是一个关系型数据库云服务，目前已在全球十多个地域（Region）的数据中心部署，向用户提供开箱即用的在线数据库服务。PolarDB目前支持3种独立的引擎，分别可以100%兼容MySQL、100%兼容PostgreSQL、高度兼容Oracle语法，存储容量最高可达100 TB。详情请参见什么是。
- Q：为什么云原生关系型数据库PolarDB优于传统数据库？
  A：相较于传统数据库，云原生关系型数据库PolarDB支持上百TB级别海量数据存储，提供高可用和高可靠保障、快速弹性升降级、无锁备份等功能，详情请参见产品优势。
- Q：PolarDB是什么时候发布？什么时候开始商用？
  A：2017年9月发布公测，2018年3月开始商用。
- Q：集群和节点分别指的是什么？
  A：PolarDB集群版采用多节点集群的架构，集群中有一个主节点和多个只读节点。单个PolarDB集群支持跨可用区，但不能跨地域，面向集群进行管理和计费。详情请参见常见术语。
- Q：支持哪些编程语言？
  A：PolarDB支持Java、Python、PHP、Golang、C、C++、.NET、Node.js等编程语言。
- Q：PolarDB是分布式数据库吗？
  A：是的，PolarDB是基于Parallel Raft一致性协议的分布式存储集群，计算引擎是由1~16个分布在不同服务器上的计算节点构成，存储容量最高可达100 TB，最高支持88核710 GB内存，可在线动态扩容存储和计算资源，扩容时不会影响业务的正常运行。
- Q：购买PolarDB后，如果需要分库分表是否还需要购买PolarDB-X数据库中间件？
  A：是的。
- Q：PolarDB是否支持表的分区？
  A：支持。
- Q：PolarDB是否已经自动包含了分区机制？
  A：PolarDB在存储层做了分区，对用户透明，无感知。
- Q：对比原生MySQL，PolarDB单表最多支持存储多少数据量？
  A：PolarDB不限制单表大小，但单表大小受磁盘空间大小限制，详情请参见使用限制。

费用

- Q：PolarDB的费用都包含哪些？
  A：包含存储空间、计算节点、备份（附赠免费额度）、SQL洞察（可选），详情请参见规格与定价。
- Q：收费的存储空间都包含哪些内容？
  A：包含数据库表文件、索引文件、undo日志文件、Redo日志文件、slowlog文件及少量的系统文件，详情请参见规格与定价。
- Q：PolarDB的存储包怎么用？
  A：购买的包年包月或按量付费的集群，均可使用存储包抵扣存储费用。例如您有3个存储容量均为40 GB的集群（即总容量为120 GB），这3个集群可以共享一个100 GB的存储包，多出的20 GB则按量计费，详情请参见购买存储包。

集群访问（读写分离）

- Q：如何实现PolarDB的读写分离？
  A：只需在应用程序中使用集群地址，即可根据配置的读写模式实现读写分离，详情请参见配置数据库代理。

- Q：一个PolarDB集群内最多可以支持多少个只读节点？
  A：PolarDB采用分布式集群架构，一个集群包含一个主节点和最多15个只读节点（至少一个，用于保障高可用）。
- Q：多个只读节点间负载不均衡的原因是什么？
  A：只读节点间负载不均衡的原因有只读节点连接数较少、自定义集群地址分配时未包括某个只读节点等。
- Q：造成主节点负载高或低的原因是什么？
  A：造成主节点（主库）负载高的原因有直连主地址、主库接受读请求、存在大量的事务请求、主从复制延迟高导致请求被路由到主库、只读节点异常导致读请求被路由到主库等。
  而主节点负载较低的原因可能是主库开启了不接受读选项。
- Q：怎么降低主节点的负载？
  A： 您可以使用如下几种方式降低主节点负载：
  - 使用集群地址来连接PolarDB集群，详情请参见配置数据库代理。
  - 如果由于事务较多导致主节点压力大，您可以通过控制台打开事务拆分功能，把事务中的部分查询路由到只读节点，详情请参见事务拆分。
  - 如果由于复制延迟导致请求被路由到主库，您可以考虑降低一致性等级（如使用最终一致性），详情请参见一致性级别。
  - 主库接受读请求，可能也会导致主库负载高，您可以通过控制台开启主库不接受读功能，减少读请求被路由到主库。
- Q：为什么读不到刚插入的数据？
  A：该问题可能是由于一致性级别的配置导致的，PolarDB的集群地址支持如下几种一致性级别：
  - 最终一致性：不论是同一会话（连接）或不同会话，最终一致性都不保证读能够马上读到刚插入的数据。
  - 会话一致性：一定能够读到同一会话插入之后的数据。

  > **说明** 一致性等级越高，性能越差，对主库的压力越大，请谨慎选择。对于大多数应用场景会话一致性能够保证业务正常工作，对于少数有强一致性的需求的语句，可以通过Hint `/* FORCE_MASTER */` 来实现，详情请参见一致性级别。

- Q：如何强制SQL到主节点执行？
  使用集群地址时，在SQL语句前加上 `/* FORCE_MASTER */` 或 `/* FORCE_SLAVE */` ，即可强制指定这条SQL的路由方向，详情请参见一致性级别选择最佳实践。
  - `/* FORCE_MASTER */` 强制请求被路由到主库。该用法可以用于解决少数一致性要求较高的读请求的场景。
  - `/* FORCE_SLAVE */` 强制请求被路由到从库。该用法可以用于解决少数PolarDB代理由于保证正确性，要求特殊语法被路由到从库的场景（比如存储过程的调用，multistatement的使用等语句默认是会被路由到从库）。

  > **说明**
  > - 若您需要通过MySQL官方命令行执行上述Hint语句，请在命令行中加上-c参数，否则该Hint会被MySQL官方命令行过滤导致Hint失效，具体请参见MySQL官方命令行。
  > - Hint的路由优先级最高，不受一致性级别和事务拆分的约束，使用前请进行评估。
  > - Hint语句里不要有改变环境变量的语句，例如 `/*FORCE_SLAVE*/ set names utf8;` 等，这类语句可能导致查询结果非预期。

- Q：是否可以给不同的业务分配不同的地址？不同地址间是否可以达到隔离的效果？
  A：您可以创建多个自定义地址给不同的业务使用，若底层节点不同则自定义地址间可同时具备隔离的效果，不会互相影响。关于如何创建自定义地址，详情请参见创建自定义集群地址。
- Q：如果有多个只读节点，如何为其中某个只读节点单独创建单节点地址？
  A：仅当集群地址读写模式为只读且集群内拥有三个及以上节点时，才支持创建单节点地址，详细操作步骤请参见创建自定义集群地址。

  警告 创建单节点地址后，当此节点故障时，该地址可能会出现最多1小时不可用的情况，请勿用于生产环境。

- Q：一个集群内最多允许创建多少个单节点地址？
  A：如果您的集群内有3个节点，则只允许为其中1个只读节点创建单节点地址；若集群内有4个节点，则允许为其中2个只读节点创建各自的单节点地址，以此类推。
- Q：只用了主地址，但是发现只读节点也有负载，是否主地址也支持读写分离？
  A：主地址不支持读写分离，始终只连接到主节点。只读节点有少量QPS是正常现象，与主地址无关。

管理与维护

- Q：主节点（主）与只读节点（备）是否存在复制延迟？
  A：是，它们之间存在毫秒级延迟。
- Q：什么情况下会导致复制延迟增大？
  A：出现如下情况时会导致复制延迟增大：
  - 主节点写入负载高，产生了过多的Redo日志，导致只读节点来不及应用。
  - 只读节点负载过高，抢占了过多原本属于应用Redo日志的资源。
  - I/O出现瓶颈，导致读写Redo日志过慢。
- Q：存在复制延迟的情况下，如何保证查询的一致性？
  A：您可以使用集群地址并为其选择合适的一致性级别。目前一致性从高到低分别为会话一致性和最终一致性，详情请参见配置数据库代理。
- Q：单节点故障的情况下是否可以保证RPO为0？
  A：可以。
- Q：升级规格配置（比如从2核8 GB升级到4核16 GB）后端是怎么实现的？对业务有什么影响？
  A：PolarDB的代理（Proxy）和数据库节点（Node）均需要升级到最新的配置，采用多个节点滚动升级的方式尽量减少对业务的影响。目前每次升级大概需要10~15分钟，对业务的影响时间不超过30秒，期间可能会产生1~3次连接闪断，详情请参见变更配置。
- Q：添加节点要多久？是否会影响业务？
  A：每增加一个节点需要5分钟，对业务无影响。关于如何添加节点，详情请参见增加只读节点。

  说明 新增只读节点之后新建的读写分离连接会转发请求到该只读节点。新增只读节点之前建立的读写分离连接不会转发请求到新增的只读节点，需要断开该连接并重新建立连接，例如，重启应用。

- Q：升级到最新修订版本需要多久？是否会影响业务？
  A：PolarDB采用多节点滚动升级的方式尽量减少对业务的影响。版本升级一般不超过30分钟，升级过程中会重启数据库代理Proxy或内核引擎DB，可能会导致数据库连接闪断。请您尽量在业务低峰期执行升级操作，并且确保您的应用有自动重连机制。详情请参见版本管理。
- Q：如何进行故障自动切换？

A：PolarDB采用双活（Active-Active）的高可用集群架构，可读写的主节点和只读节点之间自动进行故障切换（Failover），系统自动选举新的主节点。PolarDB每个节点都有一个故障切换（Failover）优先级，决定了故障切换时被选举为主节点的概率高低。当多个节点的优先级相同时，则有相同的概率被选举为主节点，详情请参见自动/手动主备切换。

备份与恢复

- Q：PolarDB采用什么备份方式？
  A：PolarDB采用快照（Snapshot）的方式进行备份，详情请参见备份数据。
- Q：数据库恢复的速度如何？
  A：目前，基于备份集（快照）进行恢复（克隆）的速度是40分钟/TB。如果是恢复到任意时间点，则需要包含应用Redo日志的时间，这部分的恢复速度大概是20~70秒/GB，整个恢复时间是这两部分之和。

性能和容量

- Q：表个数上限是多少？表个数到多少时有可能会引起性能下降？
  A：表个数的上限受文件数量限制，详情请参见使用限制。
- Q：表分区能够提高PolarDB的查询性能吗？
  A：通常来说，如果查询SQL能够落在某个分区内，是可以提升性能的。
- Q：PolarDB是否支持创建1万个数据库？数据库个数上限是多少？
  A：PolarDB支持创建1万个数据库。数据库个数上限受文件数量限制，详情请参见使用限制。
- Q：IOPS是怎么限制和隔离的？是否会出现多个PolarDB集群节点的I/O争抢？
  A：PolarDB集群的每个节点根据规格大小设置IOPS，每个节点之间IOPS独立隔离，互不影响。
- Q：只读节点的性能变慢是否会影响主节点？
  A：只读节点的负载过高、复制延迟增高时，可能会少量增加主节点的内存消耗。
- Q：打开SQL洞察（全量SQL日志审计），对性能有什么影响？
  A：无影响。
- Q：PolarDB使用了什么高速网络协议？
  A：PolarDB的数据库计算节点和存储节点之间，以及存储数据多副本之间，都使用了双25 Gbps RDMA技术，提供低延迟、高吞吐的强劲I/O性能。
- Q：PolarDB外网连接的带宽上限是多少？
  A：PolarDB外网连接的带宽上限为10 Gbit/s。
- Q：重启节点需要的时间很长怎么办？
  A：当您的集群中存在的文件数量越多，节点重启需要的时间会越长。此时您可以通过修改innodb_fast_startup参数值为*ON*来加速重启，关于如何修改参数，请参见设置集群参数。

> **说明** 仅8.0和5.6版本支持配置该参数。

6.服务等级协议

请参见云原生关系型数据库PolarDB O引擎服务等级协议。