

ALIBABA CLOUD

阿里云

PolarDB O引擎  
安装部署与升级指南

产品版本：2.2.2

文档版本：20220511

 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 <b>设置</b> > <b>网络</b> > <b>设置网络类型</b> 。
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	在 <b>结果确认</b> 页面，单击 <b>确定</b> 。
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>

# 目录

1.关于本手册	05
2.PolarDB集群组件介绍	06
3.安装部署	07
3.1. 硬件需求	07
3.2. 软件需求	07
3.3. 预配置操作系统	08
3.4. 安装与配置集群管理工具	09
3.5. 安装与部署数据库集群	12
3.6. 部署后检查	12
4.升级版本	14
5.附录	15
5.1. 数据库引擎参数说明	15
5.2. 数据库引擎参数模板	17
5.3. 规格参数	22

# 1.关于本手册

本手册对PolarDB O引擎的安装、部署和升级进行指导。

读者对象：

- 项目交付工程师。
- 运维工程师。

## 2.PolarDB集群组件介绍

本章节介绍了PolarDB O引擎集群主要包含的组件。

PolarDB O引擎集群主要包含以下几个组件：

- PolarDB Engine：数据库引擎。
- PolarDB ClusterManager Controller(CM)：高可用集群管理中心，是数据库集群的控制流入口。
- Node Driver(Agent)：DB节点控制agent，主要执行CM发起的管理操作。
- Proxy：提供集群内的主节点和所有只读节点组成的读写分离和负载均衡服务。
- Universe：负责监控采集和审计日志采集。

## 3. 安装部署

### 3.1. 硬件需求

安装PolarDB O引擎数据库操作系统应具备基本的硬件要求。

在实际业务中，硬件配置的规划需考虑数据规模及所期望的数据库响应速度。

- 数据节点：

需求项	配置说明
服务器架构	支持X86和ARM架构。
硬盘	<ul style="list-style-type: none"><li>◦ 系统盘（<code>/usr/local</code>所在磁盘）预留10GB空间用于安装应用程序包。</li><li>◦ 1个本地数据盘，容量至少256GB。</li></ul>
CPU	最少使用4核的CPU。
内存	至少使用16GB的内存。
网络要求	千兆以上以太网。

- 日志节点：

需求项	配置说明
服务器架构	支持X86和ARM架构。
硬盘	<ul style="list-style-type: none"><li>◦ 系统盘（<code>/usr/local</code>所在磁盘）预留10GB空间用于安装应用程序包。</li><li>◦ 1个本地数据盘，容量至少1TB。</li></ul>
CPU	最少使用1核的CPU。
内存	至少使用2GB的内存。
网络要求	千兆以上以太网。

### 3.2. 软件需求

安装PolarDB O引擎数据库操作系统应具备基本的软件要求。

需求类别	配置说明
操作系统	Linux CentOS 7.2或以上。建议字符集设置为LANG=en_US.UTF-8
文件系统	本地盘文件系统为Ext4



需求类别	配置说明
glibc版本	<p>glibc版本为2.15或以上。</p> <p>glibc版本检查方法。</p> <ol style="list-style-type: none"> <li>1. 执行以下命令，获取glibc版本号。 <pre>readelf -s /usr/local/polardb_o_current/bin/polar-postgres   grep @GLIBC   awk -F '[ @]' '{printf \$(NF-1) "\n";}'   sort   uniq</pre> </li> <li>2. 执行以下命令，获取glibc版本号列表。 <pre>strings /lib64/libc.so.6   grep "^GLIBC_2."</pre> </li> </ol> <p>glibc版本列表中包含步骤1获取到的glibc版本号时，表示glibc版本满足要求，否则表示glibc版本不满足要求。</p>

### 3.3. 预配置操作系统

本章节介绍对操作系统进行预配置的相关操作。

#### 预配置操作

1. 关闭透明大页（THP）。

执行以下命令，验证是否已关闭透明大页：

```
cat /sys/kernel/mm/transparent_hugepage/enabled
```

结果为 `always madvise[never]` 表示已关闭。否则执行以下命令，关闭透明大页：

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
```

再次验证是否已关闭透明大页。

2. 设置合理的大页内存（Hugepagesize）。

- 对于x86架构机器，执行以下命令，建议Hugepagesize=2048 kB。

```
grep Hugepage /proc/meminfo
```

预期结果为Hugepagesize: 2048 kB。

- 对于ARM架构（如鲲鹏 920 系列）机器，建议的Hugepagesize需要参考PAGESIZE来设置。

```
# 查看当前系统页大小（PAGESIZE）
getconf PAGESIZE
# 查看当前系统大页内存（Hugepagesize）
grep Hugepagesize /proc/meminfo
```

建议的PAGESIZE和Hugepagesize匹配关系如下表所示：

PAGESIZE	Hugepagesize
64K	524288KB (512MB)



PAGESIZE	Hugepagesize
16K	32768KB (32MB)
4K	2048KB (2MB)

如果当前Hugepagesize与建议值匹配，则无需修改；如不匹配，可参考以下步骤进行修改：

```
# 以设置 hugepagesize=2M 为例
# 编辑或创建 /etc/fstab 文件，确保有以下内容
none /dev/hugepages2M hugetlbfs pagesize=2M 0 0
```

- o pdbcli安装集群默认设置的大页内存上限为：本机物理内存总量的 65%。如果需要自定义这个值，请修改`config.yaml`文件中的`hugepage_mem_limit_gb`字段，该字段取值为一个正整数，单位为GB。

### 3. 设置内核调优参数。

```
# 禁用 debugfs
ls -l /sys/kernel/debug/x86
将所有名为 “*_enabled” 的文件内容设置为 0（如报错可忽略），例如：
echo 0 > /sys/kernel/debug/x86/ibrs_enabled
# 优化 mount 参数
ext4 类型建议使用：rw,nodiratime,noatime,data=writeback,nobarrier,delalloc
xfs 类型建议使用：nolargeio,logbsize=262144,noatime,nodiratime,swalloc
```

### 4. 将上述设置固化到开机启动项。

```
# 修改以下文件之前，强烈建议先进行备份
# 编辑 /etc/default/grub文件，增量修改 GRUB_CMDLINE_LINUX字段内容：
"noibrs noibpb nopti nospectre_v2 nospectre_v1 lltf=off nospec_store_bypass_disable no_stf_barrier mds=off tsx=on tsx_async_abort=off mitigations=off transparent_hugepage=never default_hugepagesz=2M hugepagesz=2M hugepages=5120"
# 其中 default_hugepagesz 和 hugepagesz 设为第2步中设置的建议值
# 先执行以下命令
test -d /sys/firmware/efi && echo EFI || echo BIOS
# 如果返回 EFI
grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg
# 如果返回 BIOS
grub2-mkconfig -o /boot/grub2/grub.cfg
# 如发生重启，可按 1、2、3 中的验证方式进行验证
```

## 3.4. 安装与配置集群管理工具

本章节介绍了如何安装与配置集群管理工具。

### 前提条件

获取并安装PolarDB O引擎管理工具pdbcli，具体请参见PolarDB O引擎统一运维工具pdbcli使用指南文档中的pdbcli工具安装章节。

### 集群管理工具安装

1. 下载`polarflex-${version}-${build-date}.tar.gz`安装包，并将其发送至客户端主机上。

**说明** 当前最新稳定版 `$(version)` 为2.2.2.3, `$(build-date)`为202204270144, 请根据实际情况进行修改。

2. 执行以下命令, 创建工作目录。

```
mkdir -p pdbcli-$(version)/
```

3. 执行以下命令, 解压安装文件。

```
tar -C polarflex-$(version)/ -xf polarflex-$(version)-$(build-date).tar.gz
```

4. 执行以下 `install.sh` 命令进入工作目录并开始安装, 该命令需要sudo权限。

```
cd pdbcli-$(version)/ && ./scripts/install.sh
```

5. 安装完成后, 执行以下命令, 确认安装的版本, 版本无误即表明安装正确。

```
pdbcli version
```

## 集群管理工具配置

`pdbcli`配置文件主要包括主机存储和配置参数, 可在执行命令时指定, 默认的`config.yaml`在`pdbcli-$(version)/`文件夹下。请结合实际部署环境的情况, 根据配置文件或以下样例中的说明修改参数值。

**说明** 除必填项外, 配置文件的默认值在当前本地盘模式下无须修改, 修改完成后, 可执行 `pdbcli validate` 验证配置是否有问题。关于`pdbcli validate`内容, 具体请参见PolarDB O引擎统一运维工具`pdbcli`使用指南文档中的验证配置文件`config.yaml`的内容章节。

`config.yaml`文件的内容如下所示, 其中默认情况是在三台hosts上部署PolarDB O引擎数据库集群。

```
all:
  ## 填写所有主机的信息, 包括主机名称、IP地址等。需确保各主机之间可以使用SSH通过root用户免密码登录。
  hosts:
    ## 主机名称按照 hostNN 来命名, 例如host01。
    host01:
      ansible_host: 10.XX.XX.1
    host02:
      ansible_host: 10.XX.XX.2
    host03:
      ansible_host: 10.XX.XX.3
  vars:
    ## 如下为具体的集群配置。
    ## 设置PolarDB O引擎数据库引擎的安装根目录, 默认为: /var/local/polaradb
    # polaradb_data_root_dir: /var/local/polaradb
    ## 设置PolarDB O引擎 CM (Cluster Manager) 的安装根目录, 默认为: /var/local/polaradb
    # polaradb_cm_root_dir: /var/local/polaradb_cluster_manager
    ## 【必填】设置cluster_id, 即集群ID。该设置会影响安装文件夹。
    ## 警告: 在数据库集群创建后, 请勿修改该ID。
    cluster_id: mycluster
    ## 【必填】设置external_storage_path, 即数据存储路径。
    ## 填写绝对路径, 该目录必须不存在或者为空。 数据库集群创建后, 请勿修改该外部存储路径。
    external_storage_path: /mnt/polaradb_cluster_mycluster
    ## 设置主库节点。如果未指定, 则默认指定all.children.db[0]作为主库。
    primary_db_host: host01
```

```

## 系统参数设置
## 开启或关闭firewalld防火墙服务，默认为false，即关闭状态。如果开启，则还需要手动打开服务对应端口
。
firewalld_enabled: false
## 数据库参数设置
## 设置监听端口，默认值为1521。
# polardb_port: 1521
## 设置最大连接数，默认为2048。
# polardb_max_connections: 2048
## PolarDB replication账户设置
## 警告：数据库集群创建后，请勿修改该账户设置。
## 设置PolarDB replication用户名，默认为user_rep。
# polardb_rep_username: user_rep
## 设置PolarDB replication密码，默认为pgsql。
# polardb_rep_password: pgsql
## CM参数设置
## CM服务 HTTP 监听端口，默认为5500。
# cm_service_port: 5500
## CM服务 HTTPS 监听端口，默认为5501。
# cm_tls_service_port: 5501
## CM consensus服务监听端口，默认为5502。
# cm_consensus_port: 5502
## CM是否开启TLS，默认不开启，即false。当前版本暂不支持开启。
# cm_tls_enabled: false
## proxy参数设置
## PolarDB Proxy工作并发数，默认为2。
# polardb_proxy_concurrency: 2
## PolarDB Proxy服务端口，默认为12366。
# polardb_proxy_port: 12366
## PolarDB Proxy管理服务端口，默认为12367。
# polardb_proxy_admin_port: 12367
## RW_TYPE为1时有效。主节点是否参与读请求的负载均衡。默认为true。如果设置为false，则读请求不发往主库。
# polardb_proxy_master_accept_ready: true
## 是否开启事务拆分。默认为true。如果设置为false，事务所有请求路由到主库；如果设置为true，则事务中写之前的读请求可以路由到只读库。写之后的读还是路由到主库。
# polardb_proxy_trx_split: true
## 是否开启会话一致性。默认为true。如果设置为false，不保证会话内读写一致性；如果设置为true，保证会话一致性。效果为同一个连接内，读请求一定能读到这个连接之前写入的数据。
# polardb_proxy_casual_reads: true
## 是否开启读写分离。设置为false时，所有请求路由到主库
# polardb_proxy_rw_split: true
## agent参数设置
## Node Driver服务端口，默认为12355。
# ue_node_driver_service_port: 12355
## 为主机列表中的主机设置不同的角色。
children:
## db: 数据库集群主机的分组
## cm: Cluster Manager集群管理主机的分组
## proxy: Proxy集群主机的分组
db:
  hosts:
    ## db host可包含以下选项：
    # 【必填】 polardb_polar_hostid, 保证主机索引唯一

```

```

# external_storage_path:
# polardb_proxy_aux_db_readonly: enable readonly for aux db instance
# polardb_dma_node_type: logger or learner
host01:
  polardb_polar_hostid: 1
host02:
  polardb_polar_hostid: 2
# host03:
  polardb_polar_hostid: 3
vars:
  ## polardb_custom_params为用户自定义参数。格式为: '<key> = <value>'。等号前后必须各有一个空格。value若为字符串类型,则必须使用单引号。
  ## 例如: polar_datadir = '/1739656-1/data'
  ## 如果无需配置自定义参数,可留空 (polardb_custom_params: []), 或者使用注释符号将其屏蔽。
  # polardb_custom_params: []
  # - archive_mode = off
  # - archive_command = ''
  ## CM节点分组。将要安装CM服务的节点列在此处。当前仅支持配置为1个或3个节点作为CM节点。
  cm:
    ## 若此处CM节点分组包含三个主机,则会配置为三节点高可用模式。
    hosts:
      host01:
      host02:
      host03:
  var:
    ## proxy节点分组。将要安装proxy服务的节点列在此处。
  proxy:
    ## proxy节点分组中,建议包含至少两个节点以保障高可用及负载均衡。如果不需要proxy组件,hosts设置为空即可。
    hosts:
      host01:
      host02:
  var:

```

## 3.5. 安装与部署数据库集群

本章节将介绍如何通过pdbcli工具安装和部署数据库集群。

### PolarDB O引擎数据库及各组件安装和卸载

执行以下命令,通过pdbcli工具一键安装或卸载所有组件的软件。

```

pdbcli install/uninstall cluster --config=config.yaml

```

### PolarDB O引擎数据库集群部署和删除

执行以下命令,通过pdbcli运维工具一键部署或删除数据库集群。

```

pdbcli create/delete cluster --config=config.yaml

```

## 3.6. 部署后检查

安装和部署数据库后，需要进行安装后检查。

## 集群状态检查

执行以下命令，通过pdbcli工具检查集群状态。

```
pdbcli status --config=config.yaml
```

## 数据库检查

集群部署成功，并且在数据库启动之后，您可以通过PolarDB O引擎内核二进制包中自带的psql客户端进行连通测试，执行以下命令。

```
psql -h$PGDATA -p$PGPORT -c 'select 1;'
```

如果返回1，则证明数据库安装成功，并且启动成功。

## 版本检查

您可以通过PolarDB O引擎内核二进制包中自带的psql客户端，执行以下命令，查询当前使用的数据库版本。

```
psql -h$PGDATA -p$PGPORT -c"select version()"
```

您还可执行以下命令查询PolarDB O引擎内核小版本。

```
show polar_version;
```

 说明 例如，当前版本为 PostgreSQL 11.2 (POLARDB Database Compatible with Oracle 11.2.9)。

## 进程检查

PolarDB O引擎数据库启动成功之后，会在数据库目录下生成一个postmaster.pid，其中文件第一行内容是PolarDB O引擎守护进程pid，您可以通过 `ps -ef | grep ***`（其中\*\*\*为守护进程pid）命令，得到所有的PolarDB O引擎进程，其中有如下几类常见辅助进程。

- `/usr/local/polardb_o_current/bin/polar-postgres -D data` 为PolarDB O引擎的守护进程。
- `postgres: logger` 为PolarDB O引擎的打印日志进程。
- `postgres: checkpointer` 为PolarDB O引擎的周期性检查点进程。
- `postgres: background writer` 为PolarDB O引擎的周期性刷脏进程。
- `postgres: walwriter` 为PolarDB O引擎的定期WAL日志刷盘进程。
- `postgres: autovacuum launcher` 为PolarDB O引擎的自动清理调度进程。
- `postgres: stats collector` 为PolarDB O引擎的统计信息收集进程。
- `postgres: consensus` 为PolarDB O引擎DMA集群的强一致同步进程。

## 4. 升级版本

本章节介绍了如何通过pdbcli工具升级所有组件的软件。

您可以通过pdbcli工具升级所有组件的软件，进入pdbcli工具安装目录后，具体的操作步骤如下所示：

1. 将新版本安装包存储在 `assets/roles/common/files/` 目录。
2. 修改 `assets/roles/common/defaults/main.yml` 文件，修改相应的组件包名。

```
rpms:
  cm:
    aarch64:
    x86_64:
    - polardb-clustermanager-1.0.0-20210412174523.x86_64
  db:
    aarch64:
    - PolarDB-0-0200-2.0.1-20210402171601.aliyos7.aarch64
    x86_64:
    - PolarDB-0-0200-2.0.1-20210409112107.aliyos7.x86_64
  proxy:
    aarch64:
    x86_64:
    - t-polardb-proxy-2.3.12-20200927165203.x86_64
  ue:
    aarch64:
    x86_64:
    - t-polardb-universe-1.1.23-20210317090148.aliyos7.x86_64
    - t-polardb-universe-plugin-registry-0.0.1-20210317090604.aliyos7.x86_64
```

3. 执行以下 `pdbcli upgrade` 命令，按需进行组件升级。

```
升级任务执行组件: pdbcli upgrade agent
升级监控采集组件: pdbcli upgrade monitor
升级集群管理组件: pdbcli upgrade cm
升级数据库组件:   pdbcli upgrade db
升级集群代理组件: pdbcli upgrade proxy
```

具体可参见PolarDB O引擎统一运维工具pdbcli使用指南文档中的升级数据库引擎和升级其他数据库组件章节。



## 5.附录

### 5.1. 数据库引擎参数说明

本章节介绍了数据库引擎的参数说明。

名称	生效方式	取值范围	备注
polar_bulk_extend_size	reload	[0-1073741823]	一次性批量扩展的大小。
polar_clog_slot_size	reload	[128-8192]	事务槽的数量，在clog锁争抢的情况下可以增大，提高性能。
polar_comp_dynatune	reload	[0-100]	自动参数计算与适配功能，数据库启动时，将自动计算一些资源相关参数的值(memory, buffer、effective_cache_size等等)，覆盖掉用户在postgresql.conf参数配置中设置的值，作为新的启动参数。
polar_comp_dynatune_profile	reload	[oltp reporting mixed]	自动参数计算与适配功能，数据库启动时，将自动计算一些相关参数的值(work_mem、成本因子、自动垃圾回收相关参数、等等)，覆盖掉用户在postgresql.conf参数配置中设置的值，作为新的启动参数。
polar_comp_enable_pruning	set/reload	[on off]	设置是否启用分区表裁剪优化。
polar_comp_redwood_date	set/reload	[on off]	当DATE在命令中显示为列的数据类型时，如果配置参数polar_comp_redwood_date设置为true，则在表定义存储在数据库中时，它将转换为TIMESTAMP。因此，时间部分也将与日期一起存储在列中。这与Oracle的DATE数据类型是一致的。如果polar_comp_redwood_date设置为false，则CREATE TABLE或ALTER TABLE命令中列的数据类型将保留为本地PostgreSQL DATE数据类型，并如此存储在数据库中。PostgreSQL DATE数据类型只在列中存储日期，而不存储时间部分。无论polar_comp_redwood_date的设置如何，当DATE在任何其他上下文中显示为数据类型（例如SPL声明部分中变量的数据类型、SPL存储过程或SPL函数中形参的数据类型或SPL函数的返回类型）时，它始终在内部转换为TIMESTAMP，因此，如果存在时可以处理时间部分。
polar_comp_redwood_greatest_least	set/reload	[on off]	当参数中有一个NULL时，Oracle会就返回NULL。而PostgreSQL会忽略NULL，除非所有参数都是NULL才返回NULL。



名称	生效方式	取值范围	备注
polar_comp_redwood_raw_names	set/reload	[on off]	Oracle, 默认对象名使用大写存储。PostgreSQL默认使用小写存储。同时如果使用了非默认行为的对象名(例如Oracle中混入小写, PG对象名中混入大写), 那么在Oracle中显示是会使用双引号对这些对象名。PG不会使用双引号显示这些对象名。
polar_comp_redwood_strings	set/reload	[on off]	连接字符串时, 如果有NULL的字符串, Oracle当成empty字符处理。而PG会返回NULL(即strict模式, 连接符对应的函数为strict模式, 那么只要有任一参数为NULL, 直接返回NULL)。
polar_csn_enable	reload	[on off]	设置是否启动CSN性能优化。
polar_enable_connectby_multiexpr	set/reload	[on off]	设置是否启用connect by多表达式。
polar_enable_physical_repl_non_super_wal_snd	reload	[on off]	设置是否允许普通用户是否物理复制。
polar_enable_syslog_pipe_buffer	reload	[on off]	设置是否启动审计日志优化。
polar_max_log_files	reload	[-1-2147483647]	设置保留的错误日志文件数量。
polar_max_non_super_wal_snd	reload	[-1-262143]	设置最大普通用户流复制连接数。
polar_num_active_global_temp_table	reload	[0-1000000]	大于0启用全局临时表特性。
polar_spl_savepoint_enable	reload	[on off]	设置是否启用savepoint特性。
polar_supported_extensions	仅显示	仅显示	设置可支持的插件列表。
polar_enable_varchar2_length_with_byte	set/reload	[on off]	设置启用varchar2(n char)特性。
polar_enable_alter_trigger	reload	[on off]	设置启用alter trigger enable/disable特性。
polar_pipelined_table_function_enable	set/reload	[on off]	设置是否启用pipelined table function特性。
polar_enable_rowid	set/reload	[on off]	设置是否启用rowid优化特性。
polar_use_statistical_repagas	reload	[on off]	设置是否启用统计信息缓存特性, 对于prepare语句场景有明显加速效果; 副作用统计信息不准。

名称	生效方式	取值范围	备注
polar_force_unlogged_to_logged_table	set/reload	[on off]	设置是否默认将unlogged table转为loggedtable。
polar_ratio_to_report_enable	set/reload	[on off]	设置是否启用ratio_to_report函数。
polar_enable_listagg	reload	[on off]	设置是否启用listagg特性。
polar_comp_stmt_level_tx	set/reload	[on off]	设置是否和Oracle事务行为保持一致。打开后，如果有中间事务失败，应用层如果调用commit，已经成功的语句仍然可以提交。这个参数会影响性能。
polar_auditlog_max_query_length	reload	[512, 1073741823]	设置审计日志截断长度的阈值，超过这个阈值的审计日志会被截断。
polar_stat_stale_cost	set/reload	[on off]	即使统计信息错误，也能设置是否选择正确的索引。

## 5.2. 数据库引擎参数模板

本章节介绍了数据引擎的参数模板。

以下参数模版对应2核8G实例，结合规格参数进行修改可以得到合适的数据库参数：

```
# 关键配置，按照实际修改
port=1521
## 磁盘相关
polar_vfs.localfs_test_mode=on
polar_enable_shared_storage_mode=on
polar_datadir='$POLARDATA'
## 其他关键配置
archive_mode=off #是否打开归档模式
polar_enable_persisted_buffer_pool=off
#规格参数
polar_max_non_super_conns=400
shared_buffers=6GB
effective_cache_size=6GB
maintenance_work_mem=500MB
wal_buffers=16MB
max_wal_size=4GB
min_wal_size=3GB
max_connections=3300
polar_xlog_record_buffers=16
polar_parallel_new_bgwriter_threshold_lag=500MB
work_mem=4MB
autovacuum_max_workers=5
autovacuum_work_mem=200MB
max_parallel_workers=2
max_parallel_workers_per_gather=2
max_parallel_maintenance_workers=2
```

```
max_sync_workers_per_subscription=2
polar_logindex_mem_size=0
polar_xlog_queue_buffers=0
polar_logindex_bloom_blocks=8
polar_xlog_page_buffers=16
#其他配置
huge_pages=on #是否使用大页内存
full_page_writes=on #是否启用全页写
polar_enable_lazy_checkpoint=off
dbms_alert.max_alerts=100
autovacuum_vacuum_scale_factor=0.02
enable_parallel_hash=on
log_temp_files=100000
vacuum_freeze_table_age=200000000
max_worker_processes=256
wal_compression=off
auth_delay.milliseconds=3000
authentication_timeout=60
log_lock_waits=on
polar_comp_stmt_level_tx=off
superuser_reserved_connections=100
track_functions=pl
synchronous_standby_names=''
enable_parallel_append=on
enable_hashjoin=on
pg_stat_statements.track_utility=off
bgwriter_flush_after=128
log_checkpoints=on
auto_explain.sample_rate=1
datestyle='redwood,show_time'
polar_ring_buffer_vacuum=128
autovacuum_analyze_threshold=50
dynamic_shared_memory_type=posix
checkpoint_completion_target=0.9
jit=off
log_connections=off
effective_io_concurrency=0
polar_comp_early_lock_release=off
quote_all_identifiers=off
polar_max_non_super_wal_snd=32
log_statement_stats=off
archive_timeout=0
polar_global_temp_table_debug=off
deadlock_timeout=1000
shared_preload_libraries='polar_vfs,polar_worker,dbms_pipe,edb_gen,pg_stat_statements,auth_
delay,a
uto_explain,pg_cron,plugin_debugger,polar_monitor_preload'
statement_timeout=0
force_parallel_mode=off
log_rotation_age=0
synchronize_seqscans=on
auto_explain.log_verbose=off
checkpoint_warning=30
log_truncate_on_rotation=on
```

```
cursor_tuple_fraction=0.1
autovacuum_naptime=30
ssl=off
log_parser_stats=off
parallel_tuple_cost=0.1
polar_parallel_new_bgwriter_threshold_time=10
min_parallel_index_scan_size=64
polar_parallel_bgwriter_check_interval=10
cpu_index_tuple_cost=0.005
log_filename='postgresql-%Y-%m-%d_%H%M%S.log'
polar_comp_redwood_greatest_least=on
max_standby_streaming_delay=30000
cpu_operator_cost=0.0025
default_with_oids=on
commit_delay=0
polar_comp_dynatune=0
enable_bitmapscan=on
from_collapse_limit=8
max_logical_replication_workers=32
vacuum_cost_page_hit=1
backend_flush_after=0
polar_bulk_extend_size=512
max_stack_depth=6144
gin_pending_list_limit=4096
debug_print_parse=off
max_pred_locks_per_page=2
enable_material=on
polar_comp_redwood_date=on
polar_wal_snd_reserved_for_superuser=32
pg_stat_statements.max=5000
wal_keep_segments=0
wal_receiver_status_interval=10
wal_log_hints=off
max_wal_senders=64
extra_float_digits=0
enable_hashagg=on
log_error_verbosity=default
polar_temp_relation_file_in_shared_storage=off
logging_collector=on
enable_mergejoin=on
polar_comp_dynatune_profile=oltp
log_replication_commands=off
polar_enable_show_polar_comp_version=on
wal_level=logical
vacuum_multixact_freeze_table_age=200000000
auto_explain.log_analyze=off
vacuum_multixact_freeze_min_age=5000000
wal_retrieve_retry_interval=5000
polar_auditlog_max_query_length=4096
ssl_cert_file='server.crt'
unix_socket_permissions=0700
vacuum_defer_cleanup_age=0
autovacuum=on
enable_nestloop=on
```

```
log_statement=all
polar_bgwriter_max_batch_size=5000
commit_siblings=5
fsync=on
log_hostname=off
max_pred_locks_per_relation=-2
tcp_keepalives_count=10
idle_in_transaction_session_timeout=3600000
max_files_per_process=1000
log_planner_stats=off
default_with_rowids=on
debug_print_plan=off
log_disconnections=off
polar_vfs.logic_ins_id=1350018684
row_security=on
auto_explain.log_format=text
enable_gathermerge=on
log_duration=off
unix_socket_directories=.
log_file_mode=0600
polar_use_statistical_relpages=on
wal_sender_timeout=60000
password_encryption=md5
polar_redo_hashtable_size=131072
polar_enable_connectby_multiexpr=on
polar_check_checkpoint_legal_interval=10
vacuum_cleanup_index_scale_factor=0.1
enable_partition_pruning=on
hot_standby=on
polar_comp_redwood_raw_names=on
polar_stat_stale_cost=0.5
default_text_search_config='pg_catalog.simple'
log_line_prefix='\\1\\n\\t%p\\t%r\\t%u\\t%d\\t%t\\t%e\\t%T\\t%S\\t%U\\t%E\\t\\t'
polar_vfs.localfs_test_mode=on
bgwriter_lru_maxpages=1000
archive_command=''
checkpoint_timeout=180
polar_reserved_polar_super_conns=8
auto_explain.log_nested_statements=off
polar_comp_redwood_strings=on
autovacuum_analyze_scale_factor=0.05
vacuum_cost_delay=0
default_transaction_deferrable=off
tcp_keepalives_idle=60
log_executor_stats=off
auto_explain.log_buffers=off
log_directory=log
listen_addresses='*'
min_parallel_table_scan_size=1024
max_replication_slots=64
db_dialect=redwood
temp_buffers=1024
constraint_exclusion=partition
autovacuum_vacuum_cost_limit=10000
temp_buffers=1024
```

```
track_activity_query_size=1024
polar_vfs.libmm_num_partition=32
max_pred_locks_per_transaction=64
polar_repl_slots_reserved_for_superuser=32
vacuum_freeze_min_age=50000000
polar_read_ahead_xlog_num=200
polar_enable_physical_repl_non_super_wal_snd=off
default_statistics_target=100
pg_stat_statements.save=off
track_activities=on
wal_receiver_timeout=30000
synchronous_commit=on
polar_parallel_bgwriter_delay=10
auto_explain.log_timing=on
old_snapshot_threshold=-1
polar_clog_slot_size=512
polar_bgwriter_sleep_lsn_lag=100
polar_parallel_bgwriter_workers=5
polar_csn_enable=on
log_timezone=UTC
debug_print_rewritten=off
enable_tidscan=on
data_sync_retry=off
track_io_timing=on
auto_explain.log_min_duration=-1
lock_timeout=0
tcp_keepalives_interval=10
trace_sort=off
log_min_duration_statement=5000
max_prepared_transactions=800
enable_partitionwise_aggregate=on
autovacuum_vacuum_threshold=50
parallel_setup_cost=1000
default_transaction_read_only=off
autovacuum_multixact_freeze_max_age=140000000
log_rotation_size=131072
track_commit_timestamp=off
timezone='UTC'
vacuum_cost_page_dirty=20
polar_spl_savepoint_enable=on
polar_num_active_global_temp_table=1000
vacuum_cost_page_miss=10
enable_seqscan=on
log_destination='polar_multi_dest'
wal_writer_delay=10
polar_enable_fullpage_snapshot=off
enable_indexonlyscan=on
seq_page_cost=1
polar_buffer_copy_min_modified_count=5
cpu_tuple_cost=0.01
random_page_cost=1.1
wal_writer_flush_after=64
max_standby_archive_delay=30000
bgwriter_lru_multiplier=10
track_counts=on
```

```
track_counts=on
trace_notify=off
pg_stat_statements.track=all
polar_max_log_files=20
gin_fuzzy_search_limit=0
checkpoint_flush_after=256
wal_sync_method=fdatasync
polar_enable_default_polar_comp_guc=on
polar_enable_syslog_pipe_buffer=off
vacuum_cost_limit=10000
polar_logical_repl_workers_reserved_for_superuser=16
hot_standby_feedback=on
enable_partitionwise_join=on
join_collapse_limit=8
temp_file_limit=10000000
polar_bgwriter_batch_size_flushlist=100
max_locks_per_transaction=64
ssl_key_file='server.key'
polar_enable_varchar2_length_with_byte=off
debug_pretty_print=on
polar_copy_buffers=16384
dbms_pipe.total_message_buffer=30
auto_explain.log_triggers=off
optimizer_mode=choose
update_process_title=on
polar_max_super_conns=100
polar_comp_enable_pruning=on
log_autovacuum_min_duration=0
autovacuum_vacuum_cost_delay=0
bgwriter_delay=10
enable_sort=on
autovacuum_freeze_max_age=1200000000
polar_vfs.libmm_size=67108864
enable_indexscan=on
polar_enable_full_page_write_in_backup=on
polar_enable_lazy_checkpoint_in_backup=off
polar_enable_checkpoint_in_backup=on
polar_enable_switch_wal_in_backup=on
```

## 5.3. 规格参数

本章节介绍了数据库的规格参数。

以下参数pdbcli工具根据机器规格自动调整。

### 2c8g



```
polar_max_non_super_conns='400'  
shared_buffers=6GB  
effective_cache_size=6GB  
maintenance_work_mem=500MB  
wal_buffers=16MB  
max_wal_size=4GB  
min_wal_size=3GB  
max_connections='3300'  
polar_parallel_new_bgwriter_threshold_lag=500MB  
work_mem=4MB  
autovacuum_max_workers=5  
autovacuum_work_mem=200MB  
max_parallel_workers=2  
max_parallel_workers_per_gather=2  
max_parallel_maintenance_workers=2  
max_sync_workers_per_subscription=2
```

## 4c16g

```
polar_max_non_super_conns='800'  
shared_buffers=12GB  
effective_cache_size=12GB  
maintenance_work_mem=1GB  
wal_buffers=16MB  
max_wal_size=8GB  
min_wal_size=4GB  
max_connections='3300'  
polar_parallel_new_bgwriter_threshold_lag=500MB  
work_mem=4MB  
autovacuum_max_workers=5  
autovacuum_work_mem=400MB  
max_parallel_workers=2  
max_parallel_workers_per_gather=2  
max_parallel_maintenance_workers=2  
max_sync_workers_per_subscription=2
```

## 8c32g

```
polar_max_non_super_conns='1600'  
shared_buffers=24GB  
effective_cache_size=24GB  
maintenance_work_mem=2GB  
wal_buffers=16MB  
max_wal_size=16GB  
min_wal_size=8GB  
max_connections='3300'  
polar_xlog_record_buffers=16  
polar_parallel_new_bgwriter_threshold_lag=750MB  
work_mem=8MB  
autovacuum_max_workers=5  
autovacuum_work_mem=800MB  
max_parallel_workers=4  
max_parallel_workers_per_gather=4  
max_parallel_maintenance_workers=4  
max_sync_workers_per_subscription=4
```

## 8c64g

```
polar_max_non_super_conns='3200'  
shared_buffers=48GB  
effective_cache_size=48GB  
maintenance_work_mem=4GB  
wal_buffers=16MB  
max_wal_size=32GB  
min_wal_size=16GB  
max_connections='3300'  
polar_xlog_record_buffers=16  
polar_parallel_new_bgwriter_threshold_lag=1500MB  
work_mem=16MB  
autovacuum_max_workers=5  
autovacuum_work_mem=1600MB  
max_parallel_workers=4  
max_parallel_workers_per_gather=4  
max_parallel_maintenance_workers=4  
max_sync_workers_per_subscription=4  
polar_logindex_mem_size=0  
polar_xlog_queue_buffers=0  
polar_logindex_bloom_blocks=8  
polar_xlog_page_buffers=16
```

## 16c128g

```
polar_max_non_super_conns='6400'  
shared_buffers=96GB  
effective_cache_size=96GB  
maintenance_work_mem=4GB  
wal_buffers=16MB  
max_wal_size=64GB  
min_wal_size=32GB  
max_connections='36100'  
polar_xlog_record_buffers=16  
polar_parallel_new_bgwriter_threshold_lag=3GB  
work_mem=32MB  
autovacuum_max_workers=8  
autovacuum_work_mem=2GB  
max_parallel_workers=12  
max_parallel_workers_per_gather=12  
max_parallel_maintenance_workers=8  
max_sync_workers_per_subscription=12  
polar_logindex_mem_size=0  
polar_xlog_queue_buffers=0  
polar_logindex_bloom_blocks=8  
polar_xlog_page_buffers=16
```

### 32c256g

```
polar_max_non_super_conns='12800'  
shared_buffers=192GB  
effective_cache_size=192GB  
maintenance_work_mem=4GB  
wal_buffers=16MB  
max_wal_size=128GB  
min_wal_size=64GB  
max_connections='36100'  
polar_xlog_record_buffers=16  
polar_parallel_new_bgwriter_threshold_lag=6GB  
work_mem=64MB  
autovacuum_max_workers=8  
autovacuum_work_mem=4GB  
max_parallel_workers=28  
max_parallel_workers_per_gather=24  
max_parallel_maintenance_workers=16  
max_sync_workers_per_subscription=28  
polar_logindex_mem_size=0  
polar_xlog_queue_buffers=0  
polar_logindex_bloom_blocks=8  
polar_xlog_page_buffers=16
```

### 88c710g

```
polar_max_non_super_conns='36000'  
shared_buffers=465GB  
effective_cache_size=465GB  
maintenance_work_mem=4GB  
wal_buffers=16MB  
max_wal_size=128GB  
min_wal_size=64GB  
max_connections='36100'  
polar_xlog_record_buffers=16  
polar_parallel_new_bgwriter_threshold_lag=5GB  
work_mem=64MB  
autovacuum_max_workers=8  
autovacuum_work_mem=8GB  
max_parallel_workers=32  
max_parallel_workers_per_gather=24  
max_parallel_maintenance_workers=16  
max_sync_workers_per_subscription=32
```