SOFTSTONE 软通动力 | 数字转型新动力

天鹤数据库

产品文档

目录

1 关于天鹤数据库

天鹤数据库（iSoftStone Enterprise Database，简称 ISSEDB）是软通动力基于 openGauss 内核自主研发、安全可控的数据库，支持 intel、鲲鹏等多种处理器，具备高性能，高可用，高兼容、自动化运维等特点，可为各行业的客户提供高效、稳定、专业的数据库服务。

1.1 产品特性

高性能

通过列存储、向量化执行引擎、融合引擎等关键技术，实现百亿数据量查询秒级响应。

高可用

同城跨 AZ（Available Zone）容灾，数据不丢失，分钟级恢复。

高安全性

支持访问控制、加密认证、数据库审计、动态数据脱敏等安全特性，提供全方位端到端的数据安全保护。

高可靠

闪回和回收站

通常在表数据被错误的 UPDATE、DELETE、TRUNCATE 和 DROP 时数据难以恢复，即便恢复也仅能通过 PITR（Point-in-time recovery，基于时间点恢复）的方式恢复到错误操作前的时刻。这种修复方式会导致整个数据库不可用，并且一些用户不希望“撤销”的表操作也同样会被“撤销”。ISSEDB 对 TIMECAPSULE 以及 RECYCLEBIN 的支持，使用户可以通过指定时间戳进行闪回查询，或者指定时间戳对表进行闪回，获取到错误 DELETE、UPDATE 前的历史数据。通过从 RECYCLEBIN 中闪回 TRUNCATE、以及 DROP 的表对象，用户可以将表数据恢复到错误操作前，大大提高了用户数据的可靠性。

可维护性好

支持 WDR 诊断、慢 SQL 诊断、Session 诊断等多种维护手段，准确快速定位问题。具备 AI4DB 能力，能够通过 AI 算法实现数据库自调优、自监控、自诊断等。

1.2 基本功能

标准 SQL 支持：

支持标准的 SQL92/SQL99/SQL2003/SQL2011 规范，支持 GBK、GB18030、UTF-8、SQL ASCII 以及 Latin-1 字符集，支持 SQL 标准函数与分析函数，支持存储过程。

应用程序接口：

支持标准 JDBC 4.0 的特性、ODBC 3.5 特性。

数据库存储管理功能：

支持表空间，可以把不同表规划到不同的存储位置。

提供主备双机：

事务支持 ACID 特性、单节点故障恢复、双机数据同步，双机故障切换等。

安全管理：

支持 SSL 安全网络连接、用户权限管理、密码管理、安全审计等功能，保证数据库在管理层、应用层、系统层和网络层的安全性。

1.3 产品优势

（1）**高可用**：场景化容灾部署；平均交易响应时间<60 ms 同城 RPO=0,RTO<60s; 异地 RPO<10s,RTO<10min

（2）**极致性能**：支持高吞吐强一致性事务能力，日均业务量 10w+单节点：处理能力达 150 万 tpmC；分布式强一致：32 节点 1500 万 tpmC

（3）**高扩展**：容量和性能按需水平扩展性能容量按需水平扩展到 1000+节点，线性比达 0.85

（4）**数据安全**：纯软全密态数据库技术实现数据从传输、计算到存储的全程加密，解决数据库云隐私泄露及第三方信任问题

1.4 应用场景

大并发、大数据量、以联机事务处理为主的交易型应用，如政务、金融、电商、020、 电信 CRM/计费等，服务能力支持高扩展、弹性扩缩，应用可按需选择不同的部署规模。

2 技术白皮书

2.1 产品介绍

2.1.1 产品定位

是一款支持 SQL2003 标准语法，支持主备部署的高可用关系型数据库。

- 多种存储模式支持复合业务场景，新引入提供原地更新存储引擎。
- NUMA 化数据结构支持高性能。
- Paxos 一致性日志复制协议，主备模式，CRC 校验支持高可用。
- 支持全密态计算、账本数据库等安全特性，提供全方位端到端的数据安全保护。
- 通过 Table Access Method 接口层支持多存储引擎。

2.1.2 应用场景

（1）交易型应用

大并发、大数据量、以联机事务处理为主的交易型应用，如电商、金融、O2O、电信 CRM/计费等，可按需选择不同的主备部署模式。

（2）物联网数据

物联网场景如工业监控、远程控制、智慧城市及其延展领域、智能家居和车联网等。物联网场景的特点是传感监控设备的种类和数量多、数据采样频率高、数据存储为追加模型、对数据的操作和分析并重。

2.1.3 技术特点

分布式执行框架

业务应用下发 SQL 给 Coordinator，SQL 可以包含对数据的增（insert）、删（delete/drop）、改（update）、查（select）。Coordinator 利用数据库的优化

器生成执行计划，每个 DN 会按照执行计划的要求去处理数据。

因为数据是通过一致性 Hash 技术均匀分布在每个节点，因此 DN 在处理数据的过程中，可能需要从其他 DN 获取数据，ISSEDB 提供了三种 stream 流（广播流、聚合流和重分布流）来降低数据在 DN 节点间的流动。

DN 将结果集返回给 Coordinate 进行汇总。Coordinator 将汇总后的结果返回给业务应用。

GTM-Lite 技术

GTM-Lite 技术可以在保证事务全局强一致的同时，提供高性能的事务处理能力，避免了单 GTM 的性能瓶颈。这里的高性能事务管理指的是无锁、多版本、高并发事务技术。而且分布式的 GTM-Lite 方案提供全局事务快照和提交号管理，实现强一致性，且无中心节点性能瓶颈。

基于 NUMA-Aware 实现高性能事务处理

跨 AZ/Region 容灾技术带来高可用

集群内 HA，数据不丢失，业务秒级中断；同城跨 AZ 容灾，数据不丢失，分钟级恢复，以及两地三中心部署。支持多中心统一查询及全局一致读，整体资源利用率高；灵活的高可用方案：通过配置多副本，可以实现 DC，AZ，Region 级高可用容灾策略;负载分担及故障无缝切换；支持平滑在线扩容。

Scale-out 在线横向扩展带来高扩展

集群 HA，多层级冗余实现系统无单点故障

ISSEDB 通过硬件冗余、实例冗余、数据冗余，实现整个系统无单点故障，高可用。

2.1.4 软件架构

逻辑架构

OM

运维管理模块（Operation Manager）。提供数据库日常运维、配置管理的管理接口、工具。

CM

数据库管理模块（Cluster Manager）。管理和监控数据库系统中各个功能单元和物理资源的运行情况，确保整个系统的稳定运行。

客户端驱动

客户端驱动（Client Driver）。负责接收来自应用的访问请求，并向应用返回执行结果。客户端驱动负责与 ISSEDB 实例通信，发送应用的 SQL 命令，接收 ISSEDB 实例的执行结果。

ISSEDB 主备

ISSEDB 主备（Datanode）。负责存储业务数据、执行数据查询任务以及向客户端返回执行结果。ISSEDB 实例包含主、备两种类型，支持一主多备。建议将主、备 ISSEDB 实例分散部署在不同的物理节点中。

Storage

服务器的本地存储资源，持久化存储数据。

2.2 部署方案

2.2.1 常用概念

单机

单机指的是只有一个数据库实例。

双机

双机指的是系统中存在主备数据库实例，主实例支持读写，备实例支持只读。

一主多备

一主多备指的是在系统存在一个主机，多个备机。ISSEDB 最多支持 8 个备机。

冷热备份

冷备份：是指备份就是一个简单的备份集，不可以提供服务。

热备份：是指备份实例可以对外提供服务 。

2.2.2 部署形态汇总

单机和双机两种部署形态方案介绍请见表 1。

表 1 ISSEDB 部署形态汇总表

部署形态	技术方案	高可用	基础设置要求	业务场景	场景特点	技术规格
单机	单机	无高可用能力	单机房	物理机	·对系统的可靠性和可用性无任何要求 ·主要用于体验试用以及调测场景	系统 RTO 和 RPO 不可控 ·无实例级容灾能力，一旦出现实例故障，系统不可用 ·一旦实例级数据丢失，则数据永久丢失，无法恢复
主备	主机+备机	抵御实例级故障	单机房	单机房	·节点间无网络延迟 ·要求承受实例级故障 ·适用于对系统可靠性要求不高的场景	· RPO=0 ·实例故障 RTO<10s ·无 AZ 级容灾能力 ·推荐主备最大可用模式
一主多备	主机+多个备机 Quorum/Paxo 5	抵御实例级故障	单机房	物理机	·节点间无网络延迟 ·要求承受实例级故障	· RPO=0 ·节点间无网络延迟 ·要求承受实例级故障 ·实例故障 RTO<10s ·无 AZ 级容灾能力 ·推荐主备同步模式 ·最少 2 个副本，最多 4 个副本

软硬件规格说明

ISSEDB 支持的 CPU 和 OS 见下表。

表 2 ISSEDB 软硬件规格

交付模式	CPU	OS
开源线下	x86_64	centos7.6、openEuler 20.3、openEuler 22.3、麒麟 V10
	鲲鹏	openEuler 20.3、openEuler 22.3

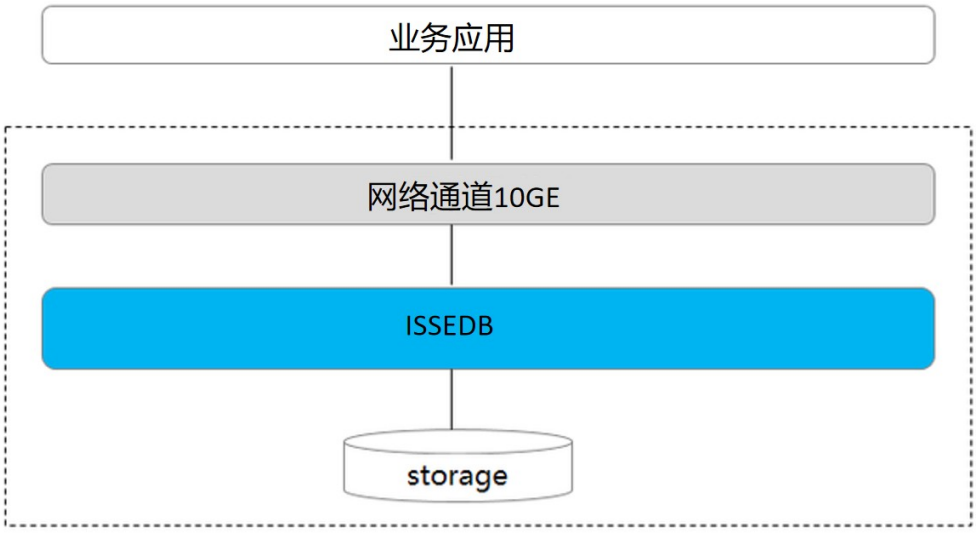
2.2.3 部署方案介绍

整体部署方案可以分为三类：单机部署、一主一备部署、一主多备部署。

单机部署

单机部署形态是一种非常特殊的部署形态，这种形态对于可靠性、可用性均无任何保证。由于只有一个数据副本，一旦发生数据损坏、丢失，只能通过物理备份恢复数据。这种部署形态，一般用于数据库体验用户，以及测试环境做语法功能调测等场景。不建议用于商业现网运行。

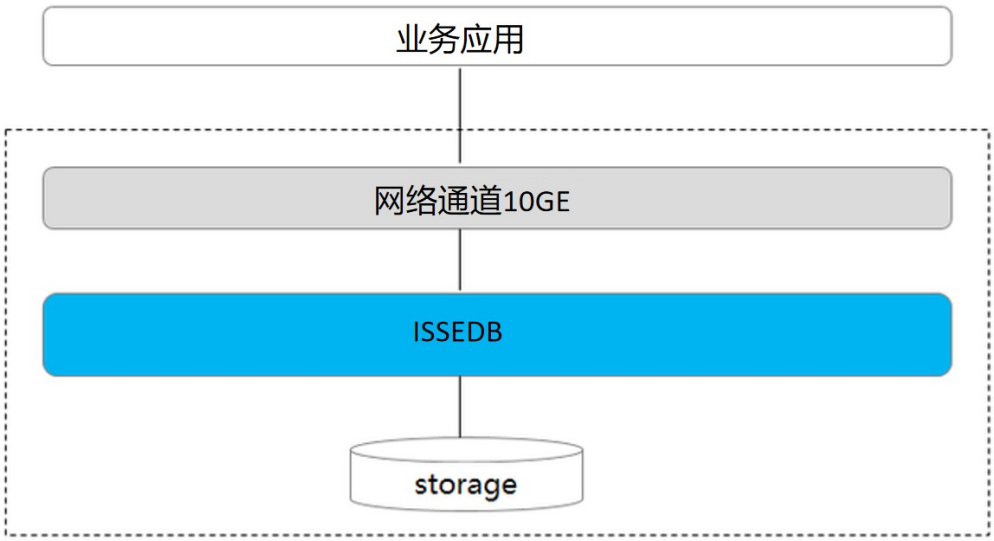
图 1 单机部署形态图



主备部署

主备模式相当于两个数据副本，主机和备机各一个数据副本，备机接收日志、执行日志回放。

图 2 单机部署形态图



一主多备部署

多副本的部署形态，提供了抵御实例级故障的能力，适用于不要求机房级别容灾，但是需要抵御个别硬件故障的应用场景。

一般多副本部署时使用 1 主 2 备模式，总共 3 个副本，3 个副本的可靠性为 99.99%，可以满足大多数应用的可靠性要求。

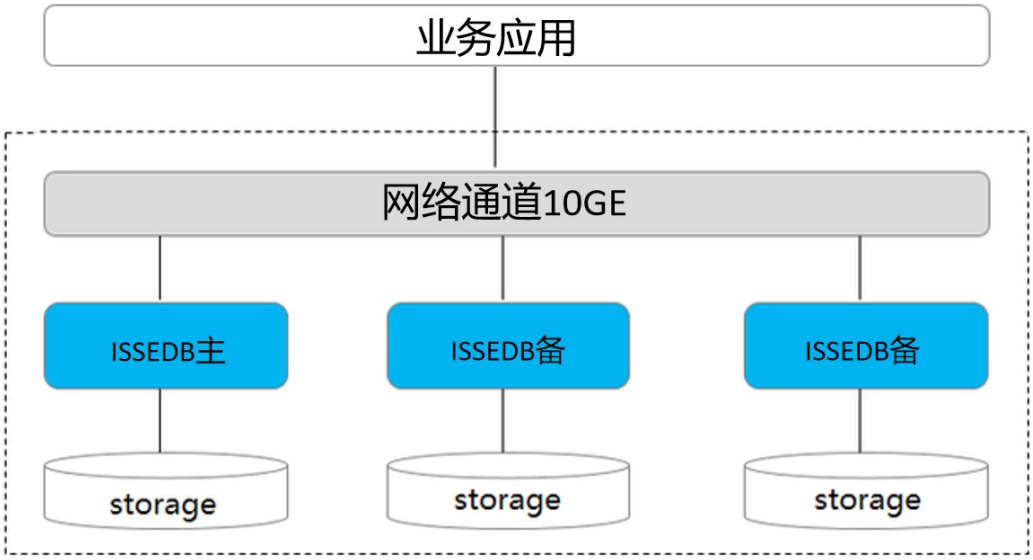
主备间 Quorum 复制，至少同步到一台备机，保证最大性能。

主备任意一个节点故障，不影响业务的进行。

数据有三份，任何一个节点故障，系统仍然有双份数据确保继续运行。任何一个备份都可以升主。

主备实例之间不可部署在同一台物理机上。

图 3 一主多备部署形态图



2.3 数据库核心技术

2.3.1 环境检查

2.3.1.1 操作系统与平台要求

本文列举了安装部署 ISSEDB 的操作系统与平台要求，同时指导用户如何查看当前的系统环境信息。

用户可以选择先查看当前系统环境信息，再结合表 1 操作系统与处理器组合情况列出的信息判断当前环境是否满足安装部署 ISSEDB 的操作系统与平台要求，并选择合适的数据库软件安装程序，为后续安装部署做准备。

操作系统环境和平台信息

ISSEDB 针对不同操作系统和 CPU 架构的组合提供了支持，支持情况如表 1 所示：

表 1 操作系统与 CPU 架构组合情况

操作系统	版本	处理器
openEuler	·openEuler 20.03 ·openEuler 20.03 SP2	·X86 ·鲲鹏 920 ·飞腾 s2500 ·海光 C86
	openEuler 20.03 SP3	·X86 ·鲲鹏 920
	openEuler 22.03	·X86 ·鲲鹏 920 ·海光 C86
	openEuler 22.03 SP1	·X86 ·鲲鹏 920
CentOS	CentOS 6.x	·X86
	CentOS 7.x	·X86 ·鲲鹏 920
	CentOS 8.x	·X86 ·鲲鹏 920 ·海光 C86
RedHat	RedHat 6.x	·X86
	·RedHat 7.x ·RedHat 8.x	·X86 ·鲲鹏 920
	kylin V7 desktop	·鲲鹏 920
	·kylin 4.0.2 SP2 ·kylin 4.0.2 SP4	·X86
	·kylin V10 ·kylin V10 国防版	·鲲鹏 920
	kylin V10 桌面版	·X86 ·鲲鹏 920 ·海光 C86
	kylin V10 国防版 2204-Build03-update1	·鲲鹏 920 ·飞腾 s2500

查看环境信息

查看方法

一般可使用如下命令查看环境信息：

```
lscpu
cat /etc/os-release
```

若为麒麟操作系统，可使用如下命令查看其小版本号（SP1、SP2）：

```
nkvers
```

用户需根据操作系统、操作系统版本、CPU 架构、CPU 型号选择对应的数据库软件安装程序。

示例

以下系统环境为 x86 平台，使用 Intel 芯片，操作系统版本为 RedHat7.8。

执行如下命令查询环境信息：

```
lscpu
cat /etc/os-release
```

返回结果分别为：

```
Architecture:          x86_64
……
Model name:            Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz
……

NAME="Red Hat Enterprise Linux Server"
VERSION="7.8 (Maipo)"
……
```

则可选择操作系统版本为“rhel_7（即 RedHat 7）”，CPU 为“x86_64”的安装包

2.3.1.2 软件环境要求

依赖软件

安装 ISSEDB 数据库的基础依赖包如下：

zlib-devel、libaio、libuuid、readline-devel、krb5-

libs、libicu、libxslt、tcl、perl、openldap、pam、openssl-

devel、libxml2、bzip2、expect

说明：麒麟环境需要预装的依赖包括：

readline、python2、libicu、cracklib、libxslt、tcl、perl、openldap、pam、system ed-libs、bzip2、gettext、libaio、ncurses-libs。且麒麟 V10 环境下需安装 libatomic，否则可能导致初始化失败

Ubuntu 环境需要预装的依赖包括：libreadline5 、zlib1g、libxml2、libaio1、libncurses5、gettext。使用 apt install <name>命令来执行安装依赖。

安装示例语句如下：

```
yum install -y zlib-devel libaio libuuid readline-devel krb5-libs libicu libxslt tcl perl openldap pam openssl-devel libxml2 bzip2 expect gcc
```

结果如下表示成功：

```
  Verifying      : libselinux-python-2.5-14.1.el7.x86_64                                                    37/43
  Verifying      : e2fsprogs-libs-1.42.9-13.el7.x86_64                                                      38/43
  Verifying      : krb5-libs-1.15.1-34.el7.x86_64                                                           39/43
  Verifying      : libselinux-utils-2.5-14.1.el7.x86_64                                                     40/43
  Verifying      : libcom_err-1.42.9-13.el7.x86_64                                                          41/43
  Verifying      : libselinux-2.5-14.1.el7.x86_64                                                           42/43
  Verifying      : e2fsprogs-1.42.9-13.el7.x86_64                                                           43/43

Installed:
  libicu.x86_64 0:50.2-4.el7_7                                  openssl-devel.x86_64 1:1.0.2k-25.el7_9

Dependency Installed:
  keyutils-libs-devel.x86_64 0:1.5.8-3.el7  krb5-devel.x86_64 0:1.15.1-51.el7_9      libcom_err-devel.x86_64 0:1.42.9-19.el7
  libkadm5.x86_64 0:1.15.1-51.el7_9         libselinux-devel.x86_64 0:2.5-15.el7     libsepol-devel.x86_64 0:2.5-10.el7
  libverto-devel.x86_64 0:0.2.5-4.el7

Updated:
  libuuid.x86_64 0:2.23.2-65.el7_9.1        libxml2.x86_64 0:2.9.1-6.el7_9.6           libxslt.x86_64 0:1.1.28-6.el7

Dependency Updated:
  e2fsprogs.x86_64 0:1.42.9-19.el7          e2fsprogs-libs.x86_64 0:1.42.9-19.el7      krb5-libs.x86_64 0:1.15.1-51.el7_9
  libblkid.x86_64 0:2.23.2-65.el7_9.1       libcom_err.x86_64 0:1.42.9-19.el7          libmount.x86_64 0:2.23.2-65.el7_9.1
  libselinux.x86_64 0:2.5-15.el7            libselinux-python.x86_64 0:2.5-15.el7      libselinux-utils.x86_64 0:2.5-15.el7
  libsmartcols.x86_64 0:2.23.2-65.el7_9.1   libss.x86_64 0:1.42.9-19.el7               openssl.x86_64 1:1.0.2k-25.el7_9
  openssl-libs.x86_64 1:1.0.2k-25.el7_9     util-linux.x86_64 0:2.23.2-65.el7_9.1

Complete!
```

Python

python 需要通过 –enable-shared 方式编译

操作系统	Python 版本
openEuler	Python 3.7.X
CentOS	Python 3.6.X
麒麟	Python 3.7.X

Asianux	Python 3.6.X

编译 python

编译前，先将编译的依赖库装好

yum install -y gcc gcc-c++ make cmake build-essential zlib-devel libffi-devel openssl-devel sqlite-devel

wget https://repo.huaweicloud.com/python/3.6.4/Python-3.6.4.tgz

从镜像站下载到源码之后，就可以一路安装了。

./configure --prefix=/usr/bin/python3.6.4 -enable-shared

make

make install

然后再设置一下环境变量，包括库搜索路径和命令搜索路径，然后就可以运行了。

export LD_LIBRARY_PATH=/usr/bin/python3.6.4/lib:$LD_LIBRARY_PATH

export PATH=/usr/bin/python3.6.4/bin:$PATH

2.3.1.3 硬件环境要求

本文主要介绍 ISSEDB 的硬件环境要求，列出了最低及建议的硬件配置，更多信息参见详细硬件环境要求。

最低及建议硬件配置

开发与测试环境

ISSEDB 数据库服务器在开发与测试环境中应具备的硬件配置要求如下：

名称	CPU	内存	本地存储	网络
ISSEDB	2 核+	4GB+	SAS,100G	千兆

生产环境

ISSEDB 数据库服务器在生产环境中应具备的最低及建议硬件配置要求如下：

注：在实际产品中，硬件配置的规划需要考虑用户的数据规模以及期望的数据

项目	最低配置	建议配置
内存	功能调试建议内存最低 8GB 以上。 搭建一主两备集群，服务器内存配置推荐为 8G 及以上，否则会出现 shared_memory 不够的情况。	性能测试和商业部署时，单实例部署建议 128GB 以上。 集群搭建一主两备时，服务器内存配置推荐为 8G 及以上。
CPU	功能调试时，CPU 配置最小为 1×8 核 2.0GHz。	性能测试和商业部署时，单实例部署建议 1×16 核 2.0GHz 以上。 支持超线程和非超线程两种 CPU 模式。ISSEDB 各节点的 CPU 模式要求一致。
硬盘	用于安装 ISSEDB 的硬盘需最少满足如下要求： 至少 1GB 用于安装 ISSEDB 的应用程序包。 每个主机需大约 300MB 用于元数据存储。 预留 70%以上的磁盘剩余空间用于数据存储。 预留 30G 磁盘用于存放 WAL 数据文件。 搭建双机单活共享存储需准备两个 lun 裸存储设备。其中作为仲裁存储设备的大小最少为 64M。	·建议系统盘配置为 Raid1，数据盘配置为 Raid5，且规划 4 组 Raid5 数据盘用于安装 ISSEDB。注意参考 Raid 配置。 ·搭建双机单活共享存储的两个 lun 裸存储设备，建议大小分别为 100G 和 1G。
网络	单机安装最低为千兆网络。 集群安装最低为万兆网络。	单机安装时建议千兆以上网络。 集群安装时建议万兆以上网络。

库响应速度，请根据实际情况进行规划。

服务器

单机部署时要求 1 台物理机或虚拟机。

集群部署时要求服务器与主备数量相关，如部署一主两备环境要求 3 台物理机或虚拟机。

CPU

ISSEDB 支持 CPU 的超线程和非超线程两种模式。但 ISSEDB 各节点的设置需保持一致。

ISSEDB 支持 x86、Arm 架构，支持 Intel、鲲鹏、飞腾、海光等芯片

内存

复杂的查询对内存的需求量比较高，在高并发场景下，可能出现内存不足的情况。该情况下建议用户使用大内存的机器，或使用负载管理限制系统的并发量。集群搭建一主两备时，服务器内存配置推荐为 8G 及以上，否则会出现 shared_memory 不够的情况。

硬盘

ISSEDB 支持使用 SSD 盘作为数据库的主存储设备，支持 SAS 接口和 NVME 协议的 SSD 盘，以 RAID 的方式部署使用。建议系统盘配置为 Raid1，数据盘配置为 Raid5。

Raid 配置 Disk Cache Policy 一项需要设置为 Disabled，否则机器异常掉电后有数据丢失的风险。

搭建双机单活共享存储需准备两个 lun 裸存储设备，设备路径规划将其分别作为共享存储设备和仲裁存储设备。

共享存储设备的大小取决于数据量，建议大小为 100G 以上。

仲裁存储设备的大小建议为 1G，至少为 64M。

网络

单机安装时要求千兆以上网络。

集群安装时要求万兆以上网络。

2.3.1.4 系统和环境变量配置

本小节主要指导用户检查和修改安装 ISSEDB 数据库要求的系统和环境配置。

为了保证 ISSEDB 数据库的正常使用，需要对操作系统和安装环境进行配置，包括如下内容：

防火墙配置

本文介绍如何对目标主机进行防火墙配置。

为了保证 ISSEDB 的正常使用，需要将节点间的访问端口打通才可以保证读写请求、数据心跳等信息的正常传输。

在普遍场景中，数据库的业务服务和数据库节点的网络联通都是在安全域内完成数据交互。

如果没有特殊安全的要求，建议将目标节点的防火墙进行关闭。参见方案一。

否则建议按照端口使用规则，将端口信息配置到防火墙服务的白名单中，参见方案二。

检测及关闭目标部署机器的防火墙

方案一：关闭防火墙

步骤 1 以 root 用户登录操作系统。

步骤 2 检查防火墙状态（以 CentOS Linux release 7 为例）。

```
sudo firewall-cmd --state
sudo systemctl status firewalld.service
```

步骤 3 关闭防火墙服务。

```
sudo systemctl stop firewalld.service
```

步骤 4 关闭防火墙自动启动服务。

```
sudo systemctl disable firewalld.service
```

步骤 5 检查防火墙状态。

```
sudo firewall-cmd --state
sudo systemctl status firewalld.service
```

方案二：开放数据库端口

步骤 1 新增端口至防火墙白名单。

```
firewall-cmd --zone=public --permanent --add-port=5432/tcp
```

步骤 2 重新加载防火墙白名单。

```
firewall-cmd --reload
```

步骤 3 查看端口列表。

```
firewall-cmd --list-port
```

下列各 IP 端口均为各服务的默认端口，用户应根据实际的规划做对应替换。

服务名称	端口号	说明
ISSEDB	5432	数据库服务端口
	5433（数据库服务端口+1）	数据库内部工具使用端口
ntp	123	ntp 默认端口
HAS	55434	集群本地监听端口
	55435	集群心跳端口
	55436	集群服务端口
	8008	集群通讯端口
DCS	2379	DCS 间通讯端口
	2380	DCS 间通讯端口

以数据库服务端口 5432 为例，开放端口的命令如下：

```
firewall-cmd --zone=public --permanent --add-port=5432/tcp
firewall-cmd -reload
firewall-cmd -list-ports
```

时间同步配置

本文介绍如何对数据库节点配置时间同步。

为了保证 ISSEDB 的正常使用，需要同步各节点的时间，若时间不同步有可能会阻碍数据库后续使用，例如出现日志中时间节点不准确等问题。

检查时间配置

使用如下命令可查看当前时间是否符合规范：

```
ll /etc/localtime
date
hwclock --show
```

设置时间

规范时间配置，使单机或高可用集群中数据库主备节点保持时间同步，可选择以下方法：

方法一：将数据库各节点的时区设置为相同时区

步骤 1 以 root 用户登录操作系统。

步骤 2 将/usr/share/zoneinfo/目录下的时区文件拷贝为/etc/localtime 文件，从而设置时区和时间。

```
cp /usr/share/zoneinfo/$主时区/$次时区 /etc/localtime
```

方法二：手工设定时间

步骤 1 以 root 用户登录操作系统。

步骤 2 执行 date -s 命令设置时区和时间。

以设置当前时间为 2020-08-03 14:15:00 为例。

```
date -s "2020-08-03 14:15:00"
```

步骤 3 将系统时间写入硬件时间。

```
hwclock –w
```

方法三：使用时间服务器同步时间

对于高可用集群生产环境，应规范时间配置，使数据库主备节点保持时间同步。选择主备节点所在局域网中一个合适的节点作为时间服务器（若条件有限，也可使用数据库主节点），使用 ntpd 或 chronyd 服务配置时间同步。操作步骤以使用 ntpd 服务为例。

步骤 1 安装 ntpd 服务。

```
yum install -y ntpd
>CentOS 或 RedHat 等系统环境下，chronyd 服务与 ntpd 服务冲突，因此
chronyd、ntpd 两种时间同步方式只能二选一。
>
>当使用 ntpd 服务时，需要禁用 chronyd 服务。
>
>```
systemctl disable chronyd && systemctl stop chronyd
```
```

步骤 2 编辑配置文件。

如果已经存在 ntp 服务器，假如为 192.168.100.100，可以配置所有节点指向该 ntp 服务器时间。编辑高可用集群中每个节点的/etc/ntp.conf 文件：

本操作目的是让 NTP Server 和其自身保持同步。

若在/etc/ntp.conf 文件中定义的 server 都不可用时，将使用 local 时间作为 ntp 服务提供给 ntp 客户端。
```

```
server 192.168.100.100 prefer
driftfile /var/lib/ntp/drift
broadcastdelay 0.008
```

如果没有外部的时间服务器，可以选择集群中的某个节点作为时间服务器。选择一个节点为 ntp 服务器（以主节点 192.168.100.1 为例）在选择的节点上配置/etc/ntp.conf 文件：

```
server 127.127.1.0
fudge 127.127.1.0 stratum 10
driftfile /var/lib/ntp/drift
broadcastdelay 0.008
```

编辑高可用集群中其他节点的/etc/ntp.conf 文件：

```
server 192.168.100.1 prefer
driftfile /var/lib/ntp/drift
broadcastdelay 0.008
```

若在 ntp 服务启动时修改 ntpd.conf 配置文件，需要重新启动 ntp 服务使其生效。

步骤 3 步骤 2 的两种方式按需选择，然后启动各节点 ntpd 服务：

```
systemctl start ntpd
systemctl enable ntpd
systemctl status ntpd
```

步骤 4 等待 5 分钟使时间服务器开始提供服务，5 分钟后检查各节点时间是否同步。

```
date
```

步骤 5 确定时间同步后，在各节点将系统时间写入硬件时间。

```
hwclock –w
```

系统内核参数配置

本文将介绍系统内核参数，并指导用户如何设置系统内核参数。

系统内核参数介绍

fs.aio-max-nr

该参数表示可以拥有的异步 IO 请求数目。

推荐值：1048576。

在 Linux 6 和 7 上使用 aio 时需要设置 fs.aio-max-nr 来适应异步 IO。

设置语句：

```
sysctl fs.aio-max-nr='1048576'
```

file-max

该参数表示每次登录会话可以打开的文件数。

在 Linux 6 和 7 上，file-max 的值与内核资源使用参数 max_files_per_process 对应。

设置语句：

```
sysctl fs.file-max='76724600'
```

kernel.sem

该参数包含 4 个参数（排序为 SEMMSL、SEMMNS、SEMOPM、SEMMNI）。

SEMMSL：每个信号量 set 中信号量最大个数，最小取值 250。

SEMMNS：Linux 系统中信号量最大个数，至少取值 32000，等于 SEMMSL*SEMMNI。

SEMOPM：SEMOP 系统调用允许的信号量最大个数，至少取值 100；或者等于 SEMMSL。

SEMMNI：Linux 系统信号量 set 最大个数，最少 128。

ipcs -l 或 ipcs -u 查看信号量。

每 16 个进程一组，每组信号量需要 17 个信号量。可根据需要修改 kernel.sem 值。

SEMMSL 代表信号量，SEMMNI 代表组。

设置语句：

```
sysctl kernel.sem='4096 2097152000 4096 512000'
```

kernel.shmall

该参数用于控制共享内存页数，等于系统内存（建议设置为 80%）/ PAGE_SIZE，该参数设置太小有可能导致数据库启动报错。

单位：byte

示例：物理内存为 128GB，PAGE 大小为 4096，则 kernel.shmall=128*1024*1024*1024*0.8\/4096=26843545。

使用 getconf PAGE_SIZE 命令查看 page_size 大小。

内存大小需换算成 bytes。

设置语句：

```
sysctl kernel.shmall='26843545'
```

kernel.shmmax

该参数表示最大单个共享内存段大小（建议为大于 shared_buffer 值）， 等于系统内存*0.5。

单位：bytes

示例：物理内存为 128GB，则：kernel.shmmax=128*1024*1024*1024*0.5=68719476736。

内存大小需换算成 bytes。

设置语句：

```
sysctl kernel.shmmax='68719476736'
```

kernel.shmmni

该参数系统范围内共享内存段的最大数量。

默认值：4096。

每个 ISSEDB 数据库集群至少 2 个共享内存段。

设置语句：

```
sysctl kernel.shmmni='819200'
```

net.core.netdev_max_backlog

该参数表示允许送到队列的数据包的最大数目。

iptables 防火墙链表相关。

设置语句：

```
sysctl net.core.netdev_max_backlog='10000'
```

net.core.rmem_default

该参数表示预留用于接收缓冲的内存默认值。

单位：bytes。

设置语句：

```
sysctl net.core.rmem_default='262144'
```

net.core.rmem_max

该参数表示预留用于接收缓冲的内存最大值。

单位：bytes。

设置语句：

```
sysctl net.core.rmem_max='4194304'
```

net.core.wmem_default

该参数表示预留用于发送缓冲的内存默认值。

单位：bytes。

设置语句：

```
sysctl net.core.wmem_default='262144'
```

net.core.wmem_max

该参数表示预留用于发送缓冲的内存最大值。

单位：bytes。

设置语句：

```
sysctl net.core.wmem_max='4194304'
```

net.core.somaxconn

该参数表示 socket 监听的 backlog（队列）上限。

默认值：128

设置语句：

```
sysctl net.core.somaxconn='4096'
```

net.ipv4.tcp_fin_timeout

该参数表示 FIN_WAIT_2 状态的超时时长。

单位：秒。

加快僵死进程回收速度。

设置语句：

```
sysctl net.ipv4.tcp_fin_timeout='5'
```

vm.dirty_background_bytes

该参数表示触发回刷的脏页数据量，超过该参数后脏页刷到磁盘。

单位：btyes。

设置语句：

```
sysctl vm.dirty_background_bytes='409600000'
```

vm.dirty_expire_centisecs

该参数表示脏数据的过期时间，超过该时间系统会将脏数据回写到磁盘上。

单位：百分之一秒。

比 vm.dirty_expire_centisecs 值旧的脏页，将被刷到磁盘。

以下命令中 3000 代表 30 秒。

设置语句：

```
sysctl vm.dirty_expire_centisecs='3000'
```

vm.dirty_ratio

该参数表示脏数据百分比，超过这个百分比，新的 IO 请求将会被阻挡，直到脏数据被写进磁盘。

设置语句：

```
sysctl vm.dirty_ratio='80'
```

vm.dirty_writeback_centisecs

该参数表示多长时间，后台刷脏页进程会被唤醒一次，检查是否有缓存需要清理。

单位：百分之一秒。

参数设置为 100 代表 1 秒，如下命令中 50 代表 0.5 秒。

设置语句：

```
sysctl vm.dirty_writeback_centisecs='50'
```

vm.overcommit_memory

该参数表示内存分配策略，可选值：0、1、2

0：表示内核将检查是否有足够的可用内存供应用进程使用，如果有足够的可用内存，内存申请允许；否则，内存申请失败，并把错误返回给应用进程。

1：表示内核允许分配所有的物理内存，而不管当前的内存状态如何。

2：表示内核允许分配超过所有物理内存和交换空间总和的内存。

在分配内存时 vm.overcommit_memory 设置为 0，vm.overcommit_ratio 参数可以不设置。

设置语句：

```
sysctl vm.overcommit_memory='0'
```

vm.swappiness

该参数表示激活交换之前可用内存的百分比。数值越低，使用的交换越少，并且物理内存中保留的内存页越多。

默认值：60

设置为 0 时表示关闭交换分区，数据库服务器不建议使用 swap，0 值仅在极限测试时使用。

设置语句：

```
sysctl vm.swappiness='60'
```

net.ipv4.ip_local_port_range

该参数可以设置本地动态端口分配范围，防止占用监听端口。

设置语句：

```
sysctl net.ipv4.ip_local_port_range='40000 65535'
```

fs.nr_open

该参数表示单个进程可分配的最大文件数。对于有很多对象（表、视图、索引、序列和物化视图等）的 PostgreSQL 数据库，建议设置为 2000 万。

单位：个

建议设置为 2000 万。

设置语句：

```
sysctl fs.nr_open='20000000'
```

设置内核参数

设置内核参数前需参考内核参数介绍，结合实际情况调节参数大小，否则会影响数据库的安装部署。

其中关键配置项含义如下，建议用户在配置时着重关注：

kernel.shmall：该参数用于控制共享内存页数，等于系统内存（建议设置为 80%，单位：byte）/PAGE_SIZE（getconf PAGE_SIZE 获取），该参数设置太小有可能导致数据库启动报错。

kernel.shmmax：该参数表示最大单个共享内存段大小（建议为大于 shared_buffer 值），等于系统内存*0.5，单位：byte。

kernel.shmmni：该参数系统范围内共享内存段的最大数量，默认值：4096。

vm.dirty_background_bytes：该参数表示触发回刷的脏页数据量，超过该参数，脏页刷到磁盘，单位：btye。

方式一：逐一设置内核参数

步骤 1 以 root 用户登录操作系统。

步骤 2 设置内核参数（逐一设置），参数及其取值参见内核参数介绍。

```
sysctl key=values
```

key 表示参数名

values 表示设置的参数值，参数值应用"包起来。

步骤 3 重载配置，使其在不关机的情况下生效。

```
sysctl -p
```

方式二：批量设置内核参数

步骤 1 以 root 用户登录操作系统。

步骤 2 编译内核参数配置文件/etc/sysctl.conf，将内核信息写入文件末尾。

当系统物理内存 > 8G 时，需重新计算内核参数设置示例中 kernel.shmall 和 kernel.shmmax 的值。

以下内核参数示例仅供参考，如需特殊化设置需结合参数介绍和实际情况进行设置。

```
fs.aio-max-nr=1048576
fs.file-max= 76724600
kernel.sem = 4096 2097152000 4096 512000
kernel.shmall = 26843545         # pages,  80% MEM or higher
kernel.shmmax = 68719476736     # bytes,  80% MEM or higher
kernel.shmmni = 819200
net.core.netdev_max_backlog = 10000
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 4194304
net.core.somaxconn = 4096
net.ipv4.tcp_fin_timeout = 5
vm.dirty_background_bytes = 409600000
vm.dirty_expire_centisecs = 3000
vm.dirty_ratio = 80
vm.dirty_writeback_centisecs = 50
vm.overcommit_memory = 0
vm.swappiness = 0
net.ipv4.ip_local_port_range = 40000 65535
fs.nr_open = 20480000
```

步骤 3 信息写入完成后保存退出。

```
:wq
```

步骤 4 重载配置，使其在不关机的情况下生效。

```
sysctl -p
```

SELINUX 配置

步骤 1 查看是否开启 SELINUX，如果是未开启则是 diabled，已开启则是 enforcing。

```
getenforce
```

步骤 2 临时关闭 SELINUX。

```
setenforce 0
```

步骤 3 通过修改配置文件永久关闭 SELINUX。

1、编辑配置文件。

```
vi /etc/selinux/config
```

2、将 SELINUX=enforcing 修改为 SELINUX=disabled。

3、重启系统。

```
reboot
```

远程登录配置

ISSEDB 安装时需要 root 帐户远程登录访问权限，可以通过如下步骤设置使用 root 用户远程登录。

步骤 1 以 root 用户登录操作系统。

步骤 2 修改 PermitRootLogin 配置，允许用户远程登录。

1、打开 sshd_config 文件

```
vi /etc/ssh/sshd_config
```

2、修改权限配置，可以使用以下两种方式实现：

将 PermitRootLogin no 注释。

```
#PermitRootLogin no
```

将“PermitRootLogin”改为“yes”。

```
PermitRootLogin yes
```

3、执行:wq 保存并退出编辑页面

步骤 3 重启 ssh 使命令生效。

```
service sshd restart
```

IPC 参数配置

首次安装时需要配置 IPC 参数。

背景信息

当 RemoveIPC=yes 时，操作系统会在用户退出时，删除该用户的 IPC 资源（共享内存段和信号量），从而使得 ISSEDB 服务器使用的 IPC 资源被清理，可能引发数据库宕机，所以需要设置 RemoveIPC 参数为 no。

操作步骤

步骤 1 以 root 用户登录操作系统。

步骤 2 进入 /etc/systemd/logind.conf 文件，查看是否已经设置了 RemoveIPC=no，如果没有则执行步骤 3，否则跳过。

```
vi /etc/systemd/logind.conf
```

步骤 3 在配置文件末尾新增配置项 RemoveIPC=no。

步骤 4 进入/usr/lib/systemd/system/systemd-logind.service 文件，查看是否已经设置了 RemoveIPC=no，如果没有设置则执行步骤 5，否则跳过。

```
vi /usr/lib/systemd/system/systemd-logind.service
```

步骤 5 修改或添加配置项 RemoveIPC=no。

步骤 6 重新加载配置参数。

```
systemctl daemon-reload
systemctl restart systemd-logind
```

步骤 7 检查修改是否生效。

```
loginctl show-session | grep RemoveIPC
systemctl show systemd-logind | grep RemoveIPC
```

设置网卡 MTU 值（可选）

将各数据库节点和交换机的网卡 MTU 值（最大传输单元）设置为相同大小（MTU 值≥1500），推荐值：8192（MTU 值可根据需要自行修改）。需要连同交换机一起修改（修改方法请咨询交换机厂商）。

步骤 1 以 root 用户登录操作系统。

步骤 2 执行如下命令，设置网卡 MTU 值。

网卡编号可通过 ip a 命令查看。

方法一：

```
#ifconfig 网卡编号 mtu 值
ifconfig eth0 mtu 8192
```

方法二：

```
cat /sys/class/net/网卡编号/mtu
echo "8192" > /sys/class/net/网卡编号/mtu
```

添加排序规则（可选）

2.3.2 单机安装

准备工作

软件版本

IP 信息及端口配置

项目名称	描述说明	备注
主机名	isoftnode2	主备主机名
IP 地址	192.168.146.162	IP
端口号	26000	DBnode 监听端口

用户名规划

项目名称	名称	所属类型	规划建议
用户名	omm	操作系统	参照官网
组名	dbgrp	操作系统	参照官网

软件目录规划

目录名称	对应名称	目录作用
/opt/software/issedb	software	安装软件存放目录
/opt/issedb/install/app	ISSEDBAppPath	数据库安装目录
/opt/issedb/log	ISSEDBLogPath	日志目录
/opt/issedb/install/data/db1	dataNode1	主备节点数据存放目录
/opt/issedb/tmp	tmpMppdbPath	临时文件目录
/opt/issedb/issedbtools/om	ISSEDBToolPath	数据库工具目录
/opt/issedb/corefile	corePath	数据库 core 文件目录

设置各节点主机名称

步骤 1 分别在各节点执行：

```
Vi /etc/hostname
hostnamectl set-hostname isoftnode2 #在数据库主节点 192.168.146.162 下执行
```

步骤 2 设置各节点/etc/hosts

```
vi /etc/hosts
192.168.146.162 isoftnode2
```

创建数据库安装用户和目录

步骤 1 以 root 用户登录操作系统。

步骤 2 创建数据库安装用户（可自定义），设定初始密码（需要重复输入 2 次且完全一致）。

```
useradd -m omm
passwd omm@2023
```

自定义的操作系统用户名可用于安装数据库，文件属组和属主需要进行相应替换。

步骤 3 创建数据库 ISSEDB 目录。

```
mkdir -p /opt/software/issedb
chmod 755 -R /opt/software
```

修改资源限制

步骤 1 （各节点）以 root 用户登录操作系统。

步骤 2 执行　vi /etc/security/limits.conf ，在文件末尾添加如下内容，保存退出。

```
issedb soft nproc unlimited
issedb hard nproc unlimited
issedb soft stack unlimited
issedb hard stack unlimited
issedb soft core unlimited
issedb hard core unlimited
issedb soft memlock unlimited
issedb hard memlock unlimited
issedb soft nofile 10240000
issedb hard nofile 10240000
```

安装数据库

前提条件

系统与环境配置

根据此部分内容对数据库主备节点操作系统环境进行配置，以下操作注意需使用 root 用户执行。

防火墙配置

关闭防火墙，或开启端口。

SELINUX 配置

临时或永久关闭 SELINUX。

时间同步配置

启动集群前必须先进行一次时间同步。

远程登录配置

设置 root 允许远程登录。

系统内核参数配置

根据系统环境实际情况配置系统内核参数。

IPC 参数设置

关闭 RemoveIPC。

预装依赖

安装基础依赖包。

下载及解压安装包

```
-- 下载软件包
cd /opt/software/issedb
wget    https://issedb-5.obs.cn-north-4.myhuaweicloud.com/issedb/centos-x86_64/issedb-5.0.0-CentOS-x86_64-ALL.tar.gz
# 解压软件包，解压 issedb-5.0.0-CentOS-64bit-all.tar.gz 后需保留 .sha256 文件，否则执行预检查时会报[GAUSS-50201] : The /opt/software/issedb/issedb-2.0.0-RedHat-64bit.sha256 does not exist.
```

```
tar -zxvf issedb-5.0.0-CentOS-64bit-all.tar.gz
tar -zxvf issedb-5.0.0-CentOS-64bit-om.tar.gz
```

创建 XML 配置文件

配置 XML 文件（主节点）

黄色字体需要根据需求替换

```
-- 节点一 root 用户
-- 在/opt/software/issedb 目录下创建 cluster_config.xml 配置文件
-- 执行操作如下
cat > cluster_config.xml<<EOF
<?xml version="1.0" encoding="UTF-8"?>
<ROOT>
    <!-- ISSEDB 整体信息 -->
    <CLUSTER>
        <PARAM name="clusterName" value="dbCluster" />
        <PARAM name="nodeNames" value="isoftnode2" />
        <PARAM name="backIp1s" value="192.168.146.162"/>
        <PARAM name="ISSEDBAppPath" value="/opt/issedb1/app" />
        <PARAM name="ISSEDBLogPath" value="/opt/issedb1/log/issedb" />
        <PARAM name="ISSEDBToolPath" value="/opt/issedb1/tool" />
        <PARAM name="corePath" value="/opt/issedb1/corefile"/>
        <PARAM name="clusterType" value="single-inst"/>
    </CLUSTER>
    <!-- 每台服务器上的节点部署信息 -->
    <DEVICELIST>
        <!-- node1 上的节点部署信息 -->
        <DEVICE sn="1000001">
            <PARAM name="name" value="isoftnode2"/>
            <PARAM name="azName" value="AZ1"/>
            <PARAM name="azPriority" value="1"/>
            <!-- 如果服务器只有一个网卡可用，将 backIP1 和 sshIP1 配置成同一个
IP -->
            <PARAM name="backIp1" value="192.168.146.162"/>
            <PARAM name="sshIp1" value="192.168.146.162"/>

            <!--dbnode-->
            <PARAM name="dataNum" value="1"/>
            <PARAM name="dataPortBase" value="26000"/>
            <PARAM name="dataNode1" value="/opt/issedb1/data/db1"/>
        </DEVICE>
    </DEVICELIST>
</ROOT>
EOF
```

参数简介

实例类型	参数	说明
整体信息	name	主机名称

实例类型	参数	说明
azName	指定 azName（Available Zone Name），字符串（不能含有特殊字符），例如 AZ1、AZ2、AZ3。	
azPriority	指定 azPriority 的优先级。	
backIp1	主机在后端存储网络中的 IP 地址（内网 IP）。所有 ISSEDB 主机使用后端存储网络通讯。	
sshIp1	设置 SSH 可信通道 IP 地址（外网 IP）。若无外网，则可以不设置该选项或者同 backIp1 设置相同 IP。	

初始化安装环境

```
# root 用户 节点一操作

-- 设置 lib 库
export         LD_LIBRARY_PATH=/opt/software/issedb/script/ISpylib/clib:
$LD_LIBRARY_PATH

-- 执行预执行
python3 /opt/software/issedb/script/gs_preinstall -U omm -G dbgrp -X
/opt/software/issedb/cluster_config.xml
# 预执行过程中，从日志来看是拷贝了 software 目录下文件到其他节点，将节点一的
scripts 下 ssh-agent.sh 拷贝到其他节点/root/.ssh/目录下，执行完并将其删除，创
建了 omm 用户，每个节点/etc/profile 下设置环境变量
# 预执行详细执行过程可查看 /opt/issedb/log/omm/om 目录下

# 执行过程可查看/opt/issedb1/log/issedb/omm/om 目录下 gs_preinstall 日志文
件 gs_preinstall_xxx.log 文件

-- 预检查执行结果如下
Parsing the configuration file.
Successfully parsed the configuration file.
Installing the tools on the local node.
Successfully installed the tools on the local node.
Are you sure you want to create trust for root (yes/no)?yes  -- 输入 yes
Please enter password for root
Password:   输入 root 口令
Successfully created SSH trust for the root permission user.
Setting host ip env
Successfully set host ip env.
Distributing package.
Begin to distribute package to tool path.
Successfully distribute package to tool path.
```

```
Begin to distribute package to package path.
Successfully distribute package to package path.
Successfully distributed package.
Are you sure you want to create the user[omm] and create trust for it
(yes/no)? yes -- 输入 yes
Please enter password for cluster user.
Password:  -- 设置 omm 口令
Please enter password for cluster user again.
Password:  -- 再次输入 omm 口令
Generate cluster user password files successfully.

Successfully created [omm] user on all nodes.
Preparing SSH service.
Successfully prepared SSH service.
Installing the tools in the cluster.
Successfully installed the tools in the cluster.
Checking hostname mapping.
Successfully checked hostname mapping.
Creating SSH trust for [omm] user.
Please enter password for current user[omm].
Password:   -- 输入 omm 口令
Checking network information.
All nodes in the network are Normal.
Successfully checked network information.
Creating SSH trust.
Creating the local key file.
Successfully created the local key files.
Appending local ID to authorized_keys.
Successfully appended local ID to authorized_keys.
Updating the known_hosts file.
Successfully updated the known_hosts file.
Appending authorized_key on the remote node.
Successfully appended authorized_key on all remote node.
Checking common authentication file content.
Successfully checked common authentication content.
Distributing SSH trust file to all node.
Distributing trust keys file to all node successfully.
Successfully distributed SSH trust file to all node.
Verifying SSH trust on all hosts.
Successfully verified SSH trust on all hosts.
Successfully created SSH trust.
Successfully created SSH trust for [omm] user.
Checking OS software.
Successfully check os software.
Checking OS version.
Successfully checked OS version.
Creating cluster's path.
Successfully created cluster's path.
Set and check OS parameter.
Setting OS parameters.
Successfully set OS parameters.
Warning: Installation environment contains some warning messages.
Please get more details by "/opt/software/issedb/script/gs_checkos -i A
-h node1,node2,issedb-db3 --detail".
Set and check OS parameter completed.
```

```
Preparing CRON service.
Successfully prepared CRON service.
Setting user environmental variables.
Successfully set user environmental variables.
Setting the dynamic link library.
Successfully set the dynamic link library.
Setting Core file
Successfully set core path.
Setting pssh path
Successfully set pssh path.
Setting Cgroup.
Successfully set Cgroup.
Set ARM Optimization.
No need to set ARM Optimization.
Fixing server package owner.
Setting finish flag.
Successfully set finish flag.
Preinstallation succeeded.

-- 查看预安装结果信息，并根据预检查调整参数
[root@op_master script]# /opt/software/issedb/script/gs_checkos -i A -h
op_master,op_slave --detail
Checking items:
   A1. [ OS version status ]                                    : Normal
        [op_slave]
        centos_7.6.1810_64bit
        [op_master]
        centos_7.6.1810_64bit
   A2. [ Kernel version status ]                                : Normal
          The names about all kernel versions are same. The value is
"3.10.0-957.el7.x86_64".
   A3. [ Unicode status ]                                       : Normal
      The values of all unicode are same. The value is "LANG=en_US.UTF-
8".
   A4. [ Time zone status ]                                     : Normal
       The informations about all timezones are same. The value is
"+0800"
    A5. [ Swap memory status ]                                    :
Warning
        [op_slave]
SwapMemory 4160745472 TotalMemory 4123009024
        [op_master]
SwapMemory 4160745472 TotalMemory 4123017216
   A6. [ System control parameters status ]                     : Normal
       All values about system control  parameters are correct.
   A7. [ File system configuration status ]                     : Normal
       Both soft nofile and hard nofile are correct.
   A8. [ Disk configuration status ]                            : Normal
        The value about XFS mount parameters is correct.
   A9. [ Pre-read block size status ]                           : Normal
       The value about Logical block size is correct.
   A11.[ Network card configuration status ]                   : Warning
        [op_slave]
BondMode Null
```

```
              Warning  reason:  network  'ens33'  'mtu'  RealValue  '1500'
ExpectedValue '8192'
        [op_master]
BondMode Null
              Warning  reason:  network  'ens33'  'mtu'  RealValue  '1500'
ExpectedValue '8192'
    A12.[ Time consistency status ]                                    :
Warning
        [op_master]
         The NTPD not detected on machine and local time is "2023-07-25
15:23:44".
        [op_slave]
         The NTPD not detected on machine and local time is "2023-07-25
15:23:44".
   A13.[ Firewall service status ]                                 : Normal
       The firewall service is stopped.
   A14.[ THP service status ]                                      : Normal
       The THP service is stopped.
Total numbers:13. Abnormal numbers:0. Warning numbers:3.

=========================================================================
======
-- 根据执行 detail 命令结果对预检查告警进行调整
-- 预检查过程中遇到的问题可参照 后文 附录 部分
```

执行数据库安装

修改属主

```
-- 务必确保已在节点一执行预检查
-- root 用户 节点一操作
chmod -R 755 /opt/software/issedb/script/
chown -R omm:dbgrp /opt/software/issedb/script/
```

执行安装

```
-- 节点一切换到 omm 用户执行
su - omm
-- 执行如下命令
gs_install    -X    /opt/software/issedb/cluster_config.xml    --ISinit-
parameter="--encoding=UTF8"\
> --dn-guc="max_connections=2000" \
> --dn-guc="max_process_memory=2GB" \
> --dn-guc="shared_buffers=128MB" \
> --dn-guc="bulk_write_ring_size=128MB" \
> --dn-guc="cstore_buffers=16MB"


#    执 行 过 程 可 查 看 /opt/issedb1/log/issedb/omm/om 及 各 节
点/opt/issedb1/log/issedb/omm/pg_log/ 下日志文件
#
[omm@op_master            script]$            ./gs_install            -X
/opt/software/issedb/cluster_config.xml \
> --ISinit-parameter="--encoding=UTF8" \
> --dn-guc="max_connections=2000" \
> --dn-guc="max_process_memory=2GB" \
```

```
> --dn-guc="shared_buffers=128MB" \
> --dn-guc="bulk_write_ring_size=128MB" \
> --dn-guc="cstore_buffers=16MB"
Parsing the configuration file.
Check preinstall on every node.
Successfully checked preinstall on every node.
Creating the backup directory.
Successfully created the backup directory.
begin deploy..
Installing the cluster.
begin prepare Install Cluster..
Checking the installation environment on all nodes.
begin install Cluster..
Installing applications on all nodes.
Successfully installed APP.
begin init Instance..
encrypt cipher and rand files for database.
Please enter password for database:
Please repeat for database:
[GAUSS-50306] : The password of database is incorrect.The two passwords
are different, please enter password again.
Please enter password for database:
Please repeat for database:
begin to create CA cert files
The         sslcert         will         be         generated         in
/opt/issedb/install/app/share/sslcert/om
Create CA files for cm beginning.
Create            CA            files            on            directory
[/opt/issedb/install/app_a07d57c3/share/sslcert/cm].     file     list:
['cacert.pem', 'server.key', 'server.crt', 'client.key', 'client.crt',
'server.key.cipher',       'server.key.rand',       'client.key.cipher',
'client.key.rand']
Non-dss_ssl_enable, no need to create CA for DSS
Cluster installation is completed.
Configuring.
Deleting instances from all nodes.
Successfully deleted instances from all nodes.
Checking node configuration on all nodes.
Initializing instances on all nodes.
Updating instance configuration on all nodes.
Check consistence of memCheck and coresCheck on database nodes.
Successful check consistence of memCheck and coresCheck on all nodes.
Configuring pg_hba on all nodes.
Configuration is completed.
Successfully installed application.
end deploy.
```

卸载数据库

卸载 gsdb 的过程包含卸载 gsdb 和对 gsdb 服务器的环境清理。

前提条件：　关闭数据库。

卸载 gsdb

步骤 1 以数据库安装用户 omm 登录数据库节点。

```
su - omm
```

步骤 2 进入安装程序所在目录。

```
cd /opt/software/issedb/script
```

步骤 3 执行卸载命令。

```
./gs_uninstall
```

卸载完成后，已安装的 gsdb 程序（即数据库安装目录$ISSEDBHOME 对应目录）和 gsdb 相关环境变量将被删除，仅保留数据库实例。

2. 3. 3 集群部署

准备工作

安装要求

本文以部署 ISSEDB 一主一备集群为例，部署环境要求如下：

环境	说明
服务器数量	2 台物理机或虚拟机
操作系统	操作系统与平台要求
软件环境要求	软件环境要求
硬件环境要求	硬件环境要求

安装流程

软件版本

IP 信息及端口配置

项目名称	描述说明	备注
主机名	Node1、Node2	主备主机名
IP 地址	192.168.146.130、192.168.146.131	主备主机名及 IP

项目名称	描述说明	备注
端口号	5000、26000	cm 监听端口、DBnode 监听端口

用户名规划

项目名称	名称	所属类型	规划建议
用户名	omm	操作系统	参照官网
组名	dbgrp	操作系统	参照官网

软件目录规划

目录名称	对应名称	目录作用
/opt/software/issedb	software	安装软件存放目录
/opt/issedb/app	ISSEDBAppPath	数据库安装目录
/opt/issedb/log/issedb	ISSEDBLogPath	日志目录
/opt/issedb/data/d1/	dataNode1	主备节点数据存放目录
/opt/issedb/tmp	tmpMppdbPath	临时文件目录
/opt/issedb/tool	ISSEDBToolPath	数据库工具目录
/opt/issedb/corefile	corePath	数据库 core 文件目录
/opt/issedb/cm	cmDir	CM 数据目录

设置各节点主机名称

步骤 1 分别在各节点执行：

```
Vi /etc/hostname
hostnamectl set-hostname node1   #在数据库主节点 192.168.146.130 下执行
```

```
hostnamectl set-hostname node2  #在数据库备节点 192.168.146.131 下执行
```

步骤 2 设置各节点/etc/hosts

```
vi /etc/hosts
192.168.146.130 node1
192.168.146.131 node2
```

创建数据库安装用户和目录

步骤 1 以 root 用户登录操作系统。

步骤 2 创建数据库安装用户（可自定义），设定初始密码（需要重复输入 2 次且完全一致）。

```
useradd -m omm
passwd omm@2023
```

自定义的操作系统用户名可用于安装数据库，文件属组和属主需要进行相应替换。

步骤 3（主节点）创建数据库软件目录（可自定义）

```
mkdir -p /opt/software/issedb/
chmod 755 -R /opt/software
```

修改资源限制

步骤 1 (各节点)以 root 用户登录操作系统。

步骤 2 执行　vi /etc/security/limits.conf，在文件末尾添加如下内容，保存退出。

```
issedb soft nproc unlimited
issedb hard nproc unlimited
issedb soft stack unlimited
issedb hard stack unlimited
issedb soft core unlimited
issedb hard core unlimited
issedb soft memlock unlimited
issedb hard memlock unlimited
issedb soft nofile 10240000
issedb hard nofile 10240000
```

部署数据库

数据库节点

本次部署数据库节点列表如下：

节点名称	IP	默认端口
数据库主节点	192.168.146.157	5432,5433,26001,26002,26003
数据库备节点	192.168.146.158	5432,5433,26001,26002,26003

前提条件

系统与环境配置

根据此部分内容对数据库主备节点操作系统环境进行配置，以下操作注意需使用 root 用户执行。

防火墙配置

关闭防火墙，或开启端口。

SELINUX 配置

临时或永久关闭 SELINUX。

时间同步配置

启动集群前必须先进行一次时间同步。

远程登录配置

设置 root 允许远程登录。

系统内核参数配置

根据系统环境实际情况配置系统内核参数。

IPC 参数设置

关闭 RemoveIPC。

预装依赖

安装基础依赖包。

下载及解压安装包

```
-- 下载软件包
cd /opt/software/issedb
wget    https://issedb-5.obs.cn-north-4.myhuaweicloud.com/issedb/centos-x86_64/issedb-5.0.0-CentOS-x86_64-ALL.tar.gz
# 解压软件包,解压 issedb-5.0.0-CentOS-64bit-all.tar.gz 后需保留 .sha256 文件,否则执行预检查时会报 [GAUSS-50201] : The /opt/software/issedb/issedb-2.0.0-RedHat-64bit.sha256 does not exist.

tar -zxvf issedb-5.0.0-CentOS-64bit-all.tar.gz
tar -zxvf issedb-5.0.0-CentOS-64bit-om.tar.gz
tar -zxvf issedb-5.0.0-CentOS-64bit-cm.tar.gz
```

创建 XML 配置文件

配置 XML 文件（主节点）

黄色字体需要根据需求替换

```
-- 节点一 root 用户
-- 在/opt/software/issedb 目录下创建 cluster_config.xml 配置文件
-- 执行操作如下
cat > cluster_config.xml<<EOF
<?xml version="1.0" encoding="UTF-8"?>
<ROOT>
    <!-- ISSEDB 整体信息 -->
    <CLUSTER>
    <!-- 数据库名称 -->
        <PARAM name="clusterName" value="ISSEDB" />
    <!-- 数据库节点名称(hostname) -->
        <PARAM name="nodeNames" value="op_master,op_slave" />
    <!-- 节点 IP，与 nodeNames 一一对应 -->
                                                <PARAM      name="backIp1s"
value="192.168.146.157,192.168.146.158"/>
    <!-- 数据库安装目录-->
        <PARAM name="ISSEDBAppPath" value="/opt/issedb/install/app" />
    <!-- 日志目录-->
        <PARAM name="ISSEDBLogPath" value="/opt/issedb/log" />
    <!-- 临时文件目录-->
        <PARAM name="tmpMppdbPath" value="/opt/issedb/tmp"/>
    <!--数据库工具目录-->
                                          <PARAM     name="ISSEDBToolPath"
value="/opt/issedb/gausstools/om" />
    <!--数据库 core 文件目录-->
        <PARAM name="corePath" value="/opt/issedb/corefile"/>
    </CLUSTER>
    <!-- 每台服务器上的节点部署信息 -->
    <DEVICELIST>
        <!-- op_master 上的节点部署信息 -->
        <DEVICE sn="1000001">
        <!-- op_master 的 hostname -->
            <PARAM name="name" value="op_master"/>
        <!-- op_master 所在的 AZ 及 AZ 优先级 -->
            <PARAM name="azName" value="AZ1"/>
            <PARAM name="azPriority" value="1"/>
         <!-- 如果服务器只有一个网卡可用，将 backIP1 和 sshIP1 配置成同一个 IP
-->
            <PARAM name="backIp1" value="192.168.146.157"/>
            <PARAM name="sshIp1" value="192.168.146.157"/>

        <!--CM-->
     <!--CM 数据目录-->
            <PARAM name="cmDir" value="/opt/issedb/install/data/cm" />
            <PARAM name="cmsNum" value="1" />
      <!--CM 监听端口-->
            <PARAM name="cmServerPortBase" value="5000" />
            <PARAM name="cmServerlevel" value="1" />
```

```
      <!--CM 所有实例所在节点名及监听 ip-->
                                          <PARAM   name="cmServerListenIp1"
value="192.168.146.157,192.168.146.158" />
                                           <PARAM   name="cmServerRelation"
value="op_master,op_slave" />

      <!--dbnode-->
            <PARAM name="dataNum" value="1"/>
      <!--DBnode 端口号-->
            <PARAM name="dataPortBase" value="26000"/>
      <!--DBnode 主节点上数据目录，及备机数据目录-->
                                                  <PARAM    name="dataNode1"
value="/opt/issedb/install/data/db1,op_slave,/opt/issedb/install/data/
db1"/>
      <!--DBnode 节点上设定同步模式的节点数-->
            <PARAM name="dataop_master_syncNum" value="0"/>
        </DEVICE>

          <!-- op_slave 上的节点部署信息，其中"name"的值配置为主机名称
(hostname) -->
        <DEVICE sn="1000002">
            <PARAM name="name" value="op_slave"/>
            <PARAM name="azName" value="AZ1"/>
            <PARAM name="azPriority" value="1"/>
            <!-- 如果服务器只有一个网卡可用，将 backIP1 和 sshIP1 配置成同一个
IP -->
            <PARAM name="backIp1" value="192.168.146.158"/>
            <PARAM name="sshIp1" value="192.168.146.158"/>
            <PARAM name="cmDir" value="/opt/issedb/install/data/cm" />
        </DEVICE>
    </DEVICELIST>
</ROOT>
EOF
```

参数简介

实例类型	参数
整体信息	name
azName	指定 azName（Available Zone Name），字符串（不能含有特殊字符），例如 AZ1、AZ2、AZ3。
azPriority	指定 azPriority 的优先级。
backIp1	主机在后端存储网络中的 IP 地址（内网 IP）。所有 ISSEDB 主机使用后端存储网络通讯。
sshIp1	设置 SSH 可信通道 IP 地址（外网 IP）。若无外网，则可以不

实例类型	参数
	设置该选项或者同 backIp1 设置相同 IP。

初始化安装环境

```
# root 用户 节点一操作

-- 设置 lib 库
export         LD_LIBRARY_PATH=/opt/software/issedb/script/ISpylib/clib:
$LD_LIBRARY_PATH

-- 执行预执行
python3 /opt/software/issedb/script/gs_preinstall -U omm -G dbgrp -X
/opt/software/issedb/cluster_config.xml
# 预执行过程中，从日志来看是拷贝了 software 目录下文件到其他节点，将节点一的
scripts 下 ssh-agent.sh 拷贝到其他节点/root/.ssh/目录下，执行完并将其删除，创
建了 omm 用户，每个节点/etc/profile 下设置环境变量
# 预执行详细执行过程可查看 /opt/issedb/log/omm/om 目录下

# 执行过程可查看 /opt/issedb/log/omm/om 目录下 gs_preinstall 日志文件
gs_preinstall_xxx.log 文件

-- 预检查执行结果如下
Parsing the configuration file.
Successfully parsed the configuration file.
Installing the tools on the local node.
Successfully installed the tools on the local node.
Are you sure you want to create trust for root (yes/no)?yes  -- 输入 yes
Please enter password for root
Password:   输入 root 口令
Successfully created SSH trust for the root permission user.
Setting host ip env
Successfully set host ip env.
Distributing package.
Begin to distribute package to tool path.
Successfully distribute package to tool path.
Begin to distribute package to package path.
Successfully distribute package to package path.
Successfully distributed package.
Are you sure you want to create the user[omm] and create trust for it
(yes/no)? yes -- 输入 yes
Please enter password for cluster user.
Password:  -- 设置 omm 口令
Please enter password for cluster user again.
Password:  -- 再次输入 omm 口令
Generate cluster user password files successfully.

Successfully created [omm] user on all nodes.
Preparing SSH service.
Successfully prepared SSH service.
Installing the tools in the cluster.
```

```
Successfully installed the tools in the cluster.
Checking hostname mapping.
Successfully checked hostname mapping.
Creating SSH trust for [omm] user.
Please enter password for current user[omm].
Password:   -- 输入 omm 口令
Checking network information.
All nodes in the network are Normal.
Successfully checked network information.
Creating SSH trust.
Creating the local key file.
Successfully created the local key files.
Appending local ID to authorized_keys.
Successfully appended local ID to authorized_keys.
Updating the known_hosts file.
Successfully updated the known_hosts file.
Appending authorized_key on the remote node.
Successfully appended authorized_key on all remote node.
Checking common authentication file content.
Successfully checked common authentication content.
Distributing SSH trust file to all node.
Distributing trust keys file to all node successfully.
Successfully distributed SSH trust file to all node.
Verifying SSH trust on all hosts.
Successfully verified SSH trust on all hosts.
Successfully created SSH trust.
Successfully created SSH trust for [omm] user.
Checking OS software.
Successfully check os software.
Checking OS version.
Successfully checked OS version.
Creating cluster's path.
Successfully created cluster's path.
Set and check OS parameter.
Setting OS parameters.
Successfully set OS parameters.
Warning: Installation environment contains some warning messages.
Please get more details by "/opt/software/issedb/script/gs_checkos -i A
-h node1,node2,issedb-db3 --detail".
Set and check OS parameter completed.
Preparing CRON service.
Successfully prepared CRON service.
Setting user environmental variables.
Successfully set user environmental variables.
Setting the dynamic link library.
Successfully set the dynamic link library.
Setting Core file
Successfully set core path.
Setting pssh path
Successfully set pssh path.
Setting Cgroup.
Successfully set Cgroup.
Set ARM Optimization.
No need to set ARM Optimization.
Fixing server package owner.
```

```
Setting finish flag.
Successfully set finish flag.
Preinstallation succeeded.

-- 查看预安装结果信息，并根据预检查调整参数
[root@op_master script]# /opt/software/issedb/script/gs_checkos -i A -h
op_master,op_slave --detail
Checking items:
   A1. [ OS version status ]                                    : Normal
        [op_slave]
        centos_7.6.1810_64bit
        [op_master]
        centos_7.6.1810_64bit
   A2. [ Kernel version status ]                                : Normal
         The names about all kernel versions are same. The value is
"3.10.0-957.el7.x86_64".
   A3. [ Unicode status ]                                       : Normal
      The values of all unicode are same. The value is "LANG=en_US.UTF-
8".
   A4. [ Time zone status ]                                     : Normal
        The informations about all timezones are same. The value is
"+0800"
    A5. [ Swap memory status ]                                  :
Warning
        [op_slave]
SwapMemory 4160745472 TotalMemory 4123009024
        [op_master]
SwapMemory 4160745472 TotalMemory 4123017216
   A6. [ System control parameters status ]                     : Normal
       All values about system control  parameters are correct.
   A7. [ File system configuration status ]                     : Normal
       Both soft nofile and hard nofile are correct.
   A8. [ Disk configuration status ]                            : Normal
        The value about XFS mount parameters is correct.
   A9. [ Pre-read block size status ]                           : Normal
       The value about Logical block size is correct.
   A11.[ Network card configuration status ]                   : Warning
        [op_slave]
BondMode Null
             Warning reason: network 'ens33' 'mtu' RealValue '1500'
ExpectedValue '8192'
        [op_master]
BondMode Null
             Warning reason: network 'ens33' 'mtu' RealValue '1500'
ExpectedValue '8192'


    A12.[ Time consistency status ]                             :
Warning
        [op_master]
         The NTPD not detected on machine and local time is "2023-07-25
15:23:44".
        [op_slave]
         The NTPD not detected on machine and local time is "2023-07-25
15:23:44".
```

```
   A13.[ Firewall service status ]                                 : Normal
       The firewall service is stopped.
   A14.[ THP service status ]                                      : Normal
       The THP service is stopped.
        The THP service is stopped.
Total numbers:13. Abnormal numbers:0. Warning numbers:3.

=========================================================================
======
-- 根据执行 detail 命令结果对预检查告警进行调整
-- 预检查过程中遇到的问题可参照 后文 附录 部分
```

执行集群安装

修改属主

```
-- 务必确保已在节点一执行预检查
-- root 用户 节点一操作
chmod -R 755 /opt/software/issedb/script/
chown -R omm:dbgrp /opt/software/issedb/script/
```

执行安装

```
-- 节点一切换到 omm 用户执行
su - omm
-- 执行如下命令
gs_install    -X    /opt/software/issedb/cluster_config.xml    --ISinit-
parameter="--encoding=UTF8"\
> --dn-guc="max_connections=2000" \
> --dn-guc="max_process_memory=2GB" \
> --dn-guc="shared_buffers=128MB" \
> --dn-guc="bulk_write_ring_size=128MB" \
> --dn-guc="cstore_buffers=16MB"


# 执行过程可查看/opt/issedb/log/omm/om 及各节点/opt/issedb/log/omm/pg_log/
下日志文件
#
[omm@op_master            script]$            ./gs_install            -X
/opt/software/issedb/cluster_config.xml \
> --ISinit-parameter="--encoding=UTF8" \
> --dn-guc="max_connections=2000" \
> --dn-guc="max_process_memory=2GB" \
> --dn-guc="shared_buffers=128MB" \
> --dn-guc="bulk_write_ring_size=128MB" \
> --dn-guc="cstore_buffers=16MB"
Parsing the configuration file.
Check preinstall on every node.
Successfully checked preinstall on every node.
Creating the backup directory.
Successfully created the backup directory.
begin deploy..
Installing the cluster.
begin prepare Install Cluster..
Checking the installation environment on all nodes.
begin install Cluster..
```

```
Installing applications on all nodes.
Successfully installed APP.
begin init Instance..
encrypt cipher and rand files for database.
Please enter password for database:
Please repeat for database:
[GAUSS-50306] : The password of database is incorrect.The two passwords
are different, please enter password again.
Please enter password for database:
Please repeat for database:
begin to create CA cert files
The         sslcert         will         be         generated         in
/opt/issedb/install/app/share/sslcert/om
Create CA files for cm beginning.
Create            CA            files            on            directory
[/opt/issedb/install/app_a07d57c3/share/sslcert/cm].     file     list:
['cacert.pem', 'server.key', 'server.crt', 'client.key', 'client.crt',
'server.key.cipher',       'server.key.rand',       'client.key.cipher',
'client.key.rand']
Non-dss_ssl_enable, no need to create CA for DSS
Cluster installation is completed.
Configuring.
Deleting instances from all nodes.
Successfully deleted instances from all nodes.
Checking node configuration on all nodes.
Initializing instances on all nodes.
Updating instance configuration on all nodes.
Check consistence of memCheck and coresCheck on database nodes.
Successful check consistence of memCheck and coresCheck on all nodes.
Configuring pg_hba on all nodes.
Configuration is completed.
Starting cluster.
=========================================================================
Successfully started primary instance. Wait for standby instance.
=========================================================================
.
Successfully started cluster.
=========================================================================
cluster_state      : Normal
redistributing     : No
node_count         : 2
Datanode State
    primary           : 1
    standby           : 1
    secondary         : 0
    cascade_standby   : 0
    building          : 0
    abnormal          : 0
    down              : 0

Successfully installed application.
end deploy.
```

3 开发者指南

3.1 各类函数

3.1.1 数学函数

abs(x)

描述：绝对值。

返回值类型：和输入相同。

cbrt(dp)

描述：立方根。

返回值类型：double precision

ceil(x)

描述：不小于参数的最小的整数。

返回值类型：整数。

degrees(dp)

描述：把弧度转为角度。

返回值类型：double precision

exp(x)

描述：自然指数。

返回值类型：dp or numeric，不考虑隐式类型转换的情况下与输入相同。

floor(x)

描述：不大于参数的最大整数。

返回值类型：与输入相同。

ln(x)

描述：自然对数。

返回值类型：dp or numeric，不考虑隐式类型转换的情况下与输入相同。

示例：

log(x)

描述：以 10 为底的对数。

返回值类型：与输入相同。

log(b numeric, x numeric)

描述：以 b 为底的对数。

返回值类型：numeric

mod(x,y)

描述：x/y 的余数（模）。如果 x 是 0，则返回 0。

返回值类型：与参数类型相同。

pi()

描述：“π”常量。

返回值类型：double precision

power(a double precision, b double precision)

描述：a 的 b 次幂。

返回值类型：double precision

radians(dp)

描述：把角度转为弧度。

返回值类型：double precision

random()

描述：0.0 到 1.0 之间的随机数。

返回值类型：double precision

round(x)

描述：离输入参数最近的整数。

返回值类型：与输入相同。

round(v numeric, s int)

描述：保留小数点后 s 位，s 后一位进行四舍五入。

返回值类型：numeric

sign(x)

描述：输出此参数的符号。

返回值类型：-1 表示负数，0 表示 0，1 表示正数。

sqrt(x)

描述：平方根。

返回值类型：dp or numeric，不考虑隐式类型转换的情况下与输入相同。

trunc(x)

描述：截断（取整数部分）。

返回值类型：与输入相同。

trunc(v numeric, s int)

描述：截断为 s 位小数。

返回值类型：numeric

3.1.2 三角函数列表

acos(x)

描述：反余弦。

返回值类型：double precision

asin(x)

描述：反正弦。

返回值类型：double precision

atan(x)

描述：反正切。

返回值类型：double precision

atan2(y, x)

描述：y/x 的反正切。

返回值类型：double precision

cos(x)

描述：余弦。

返回值类型：double precision

cot(x)

描述：余切。

返回值类型：double precision

sin(x)

描述：正弦。

返回值类型：double precision

tan(x)

描述：正切。

返回值类型：double precision

3.1.3 字符串函数和操作符

string || string

描述：连接字符串。

返回值类型：text

bit_length(string)

描述：字符串的位数。

返回值类型：int

convert(string bytea, src_encoding name, dest_encoding name)

描述：以 dest_encoding 指定的目标编码方式转化字符串 bytea。src_encoding 指定源编码方式，在该编码下，string 必须是合法的。

返回值类型：bytea

lower(string)

描述：把字符串转化为小写。

返回值类型：varchar

octet_length(string)

描述：字符串中的字节数。

返回值类型：int

overlay(string placing string FROM int [for int])

描述：替换子字符串。FROM int 表示从第一个 string 的第几个字符开始替换，for int 表示替换第一个 string 的字符数目。

返回值类型：text

position(substring in string)

描述：指定子字符串的位置。字符串区分大小写。

返回值类型：int，字符串不存在时返回 0。

substring(string [from int] [for int])

描述：截取子字符串，from int 表示从第几个字符开始截取，for int 表示截取几个字节。

返回值类型：text

substring(string from pattern)

描述：截取匹配 POSIX 正则表达式的子字符串。如果没有匹配它返回空值，否则返回文本中匹配模式的那部分。

返回值类型：text

trim([leading |trailing |both] [characters] from string)

描述：从字符串 string 的开头、结尾或两边删除只包含 characters 中字符（缺省是一个空白）的最长的字符串。

返回值类型：varchar

upper(string)

描述：把字符串转化为大写。

返回值类型：varchar

ascii(string)

描述：参数 string 的第一个字符的 ASCII 码。

返回值类型：integer

(1 row)

btrim(string text [, characters text])

描述：从 string 开头和结尾删除只包含 characters 中字符（缺省是空白）的最长字符串。

返回值类型：text

chr(integer)

描述：给出 ASCII 码的字符。

返回值类型：varchar

convert(string bytea, src_encoding name, dest_encoding name)

描述：以 dest_encoding 指定的目标编码方式转化字符串 bytea。src_encoding 指定源编码方式，在该编码下，string 必须是合法的。

返回值类型：bytea

initcap(string)

描述：将字符串中的每个单词的首字母转化为大写，其他字母转化为小写。

返回值类型：text

length(string)

描述：获取参数 string 中字符的数目。

返回值类型：integer

lpad(string text, length int [, fill text])

描述：通过填充字符 fill（缺省时为空白），把 string 填充为 length 长度。如果 string 已经比 length 长则将其尾部截断。

返回值类型：text

ltrim(string [, characters])

描述：从字符串 string 的开头删除只包含 characters 中字符（缺省是一个空白）的最长的字符串。

返回值类型：varchar

md5(string)

描述：将 string 使用 MD5 加密，并以 16 进制数作为返回值。

说明： MD5 加密算法安全性低，存在安全风险，不建议使用。

返回值类型：text

repeat(string text, number int)

描述：将 string 重复 number 次。

返回值类型：text。

replace(string text, from text, to text)

描述：把字符串 string 里出现的所有子字符串 from 的内容替换成子字符串 to 的内容。

返回值类型：text

rpad(string text, length int [, fill text])

描述：使用填充字符 fill（缺省时为空白），把 string 填充到 length 长度。如果 string 已经比 length 长则将其从尾部截断。

返回值类型：text

rtrim(string text [, characters text])

描述：从字符串 string 的结尾删除只包含 characters 中字符（缺省是个空白）的最长的字符串。

返回值类型：text

split_part(string text, delimiter text, field int)

描述：根据 delimiter 分隔 string 返回生成的第 field 个子字符串（从出现第一个 delimiter 的 text 为基础）。

返回值类型：text

strpos(string, substring)

描述：指定的子字符串的位置。和 position(substring in string)一样，不过参数顺序相反。

返回值类型：int

to_hex(number int or bigint)

描述：把 number 转换成十六进制表现形式。

返回值类型：text

translate(string text, from text, to text)

描述：把在 string 中包含的任何匹配 from 中字符的字符转化为对应的在 to 中的字符。如果 from 比 to 长，删掉在 from 中出现的额外的字符。

返回值类型：text

3.1.4 类型转换相关函数

to_char(timestamp, text)

描述：将时间戳类型的值转换为指定格式的字符串。

返回值类型：text

to_char(interval, text)

描述：将时间间隔类型的值转换为指定格式的字符串。

返回值类型：text

to_char(int, text)

描述：将整数类型的值转换为指定格式的字符串。

返回值类型：text

to_char(double precision/real, text)

描述：将浮点类型的值转换为指定格式的字符串。

返回值类型：text

to_char(numeric, text)

描述：将数字类型的值转换为指定格式的字符串。

返回值类型：text

to_date(text, text)

描述：将字符串类型的值转换为指定格式的日期。

返回值类型：timestamp without time zone

to_number(text, text)

描述：将字符串类型的值转换为指定格式的数字。

返回值类型：numeric

to_timestamp(text, text)

描述：将字符串类型的值转换为指定格式的时间戳。

返回值类型：timestamp

to_timestamp(double precision)

描述：把 UNIX 纪元转换成时间戳。

返回值类型：timestamp with time zone

3.2 数据库对象介绍

3.2.1Database 和 Schema 设计

ISSEDB 中可以使用 Database 和 Schema 实现业务的隔离，区别在于 Database 的隔离更加彻底，各个 Database 之间共享资源极少，可实现连接隔离、权限隔离等，Database 之间无法直接互访。Schema 隔离的方式共用资源较多，可以通过 grant 与 revoke 语法便捷地控制不同用户对各 Schema 及其下属对象的权限。

从便捷性和资源共享效率上考虑，推荐使用 Schema 进行业务隔离。

建议系统管理员创建 Schema 和 Database，再赋予相关用户对应的权限。

Database 设计建议

【规则】在实际业务中，根据需要创建新的 Database，不建议直接使用数据库

实例默认的 postgres 数据库。

【建议】一个数据库实例内，用户自定义的 Database 数量建议不超过 3 个。

【建议】为了适应全球化的需求，使数据库编码能够存储与表示绝大多数的字符，建议创建 Database 的时候使用 UTF-8 编码。

【关注】创建 Database 时，需要重点关注字符集编码（ENCODING）和兼容性（DBCOMPATIBILITY）两个配置项。ISSEDB 支持 A、B、C 和 PG 四种兼容模式，分别表示兼容 O 语法、MY 语法、TD 语法和 POSTGRES 语法，不同兼容模式下的语法行为存在一定差异，默认为 A 兼容模式。

【关注】Database 的 owner 默认拥有该 Database 下所有对象的所有权限，包括删除权限。删除权限影响较大，请谨慎使用。

Schema 设计建议

【关注】如果该用户不具有 sysadmin 权限或者不是该 Schema 的 owner，要访问 Schema 下的对象，需要同时给用户赋予 Schema 的 usage 权限和对象的相应权限。

【关注】如果要在 Schema 下创建对象，需要授予操作用户该 Schema 的 create 权限。

【关注】Schema 的 owner 默认拥有该 Schema 下对象的所有权限，包括删除权限。删除权限影响较大，请谨慎使用。

3.2.2 表设计

ISSEDB 的数据分布在各个 DN 上。总体上讲，良好的表设计需要遵循以下原则：

【关注】将表数据均匀分布在各个 DN 上。数据均匀分布，可以防止数据在部分 DN 上集中分布，从而导致因存储倾斜造成数据库实例有效容量下降。通过选择合适的分布列，可以避免数据倾斜。

【关注】将表的扫描压力均匀分散在各个 DN 上。避免扫描压力集中在部分 DN 上，而导致性能瓶颈。例如，在事实表上使用等值过滤条件时，将会导致扫描压力不均匀。

【关注】减少需要扫描的数据量。通过分区表的剪枝机制可以大幅减少数据的扫描量。

【关注】尽量减少随机 I/O。通过聚簇/局部聚簇可以实现热数据的连续存储，将随机 I/O 转换为连续 I/O，从而减少扫描的 I/O 代价。

【关注】尽量避免数据 shuffle。shuffle，是指在物理上，数据从一个节点，传输到另一个节点。shuffle 占用了大量宝贵的网络资源，减小不必要的数据 shuffle，可以减少网络压力，使数据的处理本地化，提高数据库实例的性能和可支持的并发度。通过对关联条件和分组条件的仔细设计，能够尽可能地减少不必要的数据 shuffle。

选择存储方案

【建议】表的存储类型是表定义设计的第一步，客户业务类型是决定表的存储类型的主要因素，表存储类型的选择依据请参考表 1。

表 1 表的存储类型及场景

存储类型	适用场景
行存	点查询（返回记录少，基于索引的简单查询）。 增、删、改操作较多的场景。
列存	统计分析类查询（关联、分组操作较多的场景）。 即席查询（查询条件不确定，行存表扫描难以使用索引）。

选择分布方案

【建议】表的分布方式的选择一般遵循以下原则：

表 2 表的分布方式及使用场景

分布方式	描述	适用场景
Hash	表数据通过 Hash 方式散列到数据库实例中的所有 DN 上。	数据量较大的事实表。
Replication	数据库实例中每一个 DN 都有一份全量表数据。	维度表、数据量较小的事实表。
Range	表数据对指定列按照范围进行映射，分布到对应 DN。	用户需要自定义分布规则的场景。
List	表数据对指定列按照具体值进行映射，分布到对应 DN。	用户需要自定义分布规则的场景。

选择分区方案

当表中的数据量很大时，应当对表进行分区，一般需要遵循以下原则：

【建议】使用具有明显区间性的字段进行分区，比如日期、区域等字段上建立分区。

【建议】分区名称应当体现分区的数据特征。例如，关键字+区间特征。

【建议】将分区上边界的分区值定义为 MAXVALUE，以防止可能出现的数据溢出。

典型的分区表定义如下：

```
CREATE TABLE staffS_p1
(
   staff_ID       NUMBER(6) not null,
   FIRST_NAME     VARCHAR2(20),
   LAST_NAME      VARCHAR2(25),
   EMAIL          VARCHAR2(25),
   PHONE_NUMBER   VARCHAR2(20),
   HIRE_DATE      DATE,
   employment_ID  VARCHAR2(10),
   SALARY         NUMBER(8,2),
   COMMISSION_PCT NUMBER(4,2),
   MANAGER_ID     NUMBER(6),
   section_ID     NUMBER(4)
)
PARTITION BY RANGE (HIRE_DATE)
(
    PARTITION HIRE_19950501 VALUES LESS THAN ('1995-05-01 00:00:00'),
    PARTITION HIRE_19950502 VALUES LESS THAN ('1995-05-02 00:00:00'),
    PARTITION HIRE_maxvalue VALUES LESS THAN (MAXVALUE)
);
```

选择分布键

Hash 表的分布键选取至关重要，如果分布键选择不当，可能会导致数据倾斜，从而导致查询时，I/O 负载集中在部分 DN 上，影响整体查询性能。因此，在确定 Hash 表的分布策略之后，需要对表数据进行倾斜性检查，以确保数据的均匀分布。分布键的选择一般需要遵循以下原则：

【建议】选作分布键的字段取值应该比较离散，以便数据能在各个 DN 上均匀分布。当单个字段无法满足离散条件时，可以考虑使用多个字段一起作为分布键。一般情况下，可以考虑选择表的主键作为分布键。例如，在人员信息表中选择证件号码作为分布键。

【建议】在满足第一条原则的情况下，尽量不要选取在查询中存在常量过滤条件的字段作为分布键。例如，在表 dwcjk 相关的查询中，字段 zqdh 存在常量过滤条件“zqdh='000001'”，那么就应当尽量不选择 zqdh 字段做为分布键。

【建议】在满足前两条原则的情况，尽量选择查询中的关联条件为分布键。当关联条件作为分布键时，join 任务的相关数据都分布在 DN 本地，将极大减少 DN 之间的数据流动代价。

3.2.3 规划存储模型

ISSEDB 支持行列混合存储。行、列存储模型各有优劣，建议根据实际情况选择。通常 ISSEDB 用于 TP 场景的数据库，默认使用行存储，仅对执行复杂查询且数据量大的 AP 场景时，才使用列存储。

行存储是指将表按行存储到硬盘分区上，列存储是指将表按列存储到硬盘分区上。默认情况下，创建的表为行存储。

行、列存储有如下优缺点：

存储模型	优点	缺点
行存	数据被保存在一起。INSERT/UPDATE 容易。	选择（Selection）时即使只涉及某几列，所有数据也都会被读取。
列存	查询时只有涉及到的列会被读取。 投影（Projection）很高效。 任何列都能作为索引。	选择完成时，被选择的列要重新组装。 INSERT/UPDATE 比较麻烦。

一般情况下，如果表的字段比较多（大宽表），查询中涉及到的列不多的情况下，适合列存储。如果表的字段个数比较少，查询大部分字段，那么选择行存储比较好。

存储类型	适用场景
行存	点查询（返回记录少，基于索引的简单查询）。

存储类型	适用场景
	增、删、改操作较多的场景。
列存	统计分析类查询（关联、分组操作较多的场景）。 即席查询（查询条件不确定，行存表扫描难以使用索引）。

3.2.4 字段设计

选择数据类型

在字段设计时，基于查询效率的考虑，一般遵循以下原则：

【建议】尽量使用高效数据类型。

选择数值类型时，在满足业务精度的情况下，选择数据类型的优先级从高到低依次为整数、浮点数、NUMERIC。

【建议】当多个表存在逻辑关系时，表示同一含义的字段应该使用相同的数据类型。

【建议】对于字符串数据，建议使用变长字符串数据类型，并指定最大长度。请务必确保指定的最大长度大于需要存储的最大字符数，避免超出最大长度时出现字符截断现象。除非明确知道数据类型为固定长度字符串，否则，不建议使用 CHAR(n)、BPCHAR(n)、NCHAR(n)、CHARACTER(n)。

3.2.5 约束设计

DEFAULT 和 NULL 约束

【建议】如果能够从业务层面补全字段值，那么，就不建议使用 DEFAULT 约束，避免数据加载时产生不符合预期的结果。

【建议】给明确不存在 NULL 值的字段加上 NOT NULL 约束，优化器会在特定场景下对其进行自动优化。

【建议】给可以显式命名的约束显式命名。除了 NOT NULL 和 DEFAULT 约束外，其他约束都可以显式命名。

局部聚簇

Partial Cluster Key（局部聚簇，简称 PCK）是列存表的一种局部聚簇技术，在 ISSEDB 中，使用 PCK 可以通过 min/max 稀疏索引实现事实表快速过滤扫描。PCK 的选取遵循以下原则：

【关注】一张表上只能建立一个 PCK，一个 PCK 可以包含多列，但是一般不建议超过 2 列。

【建议】在查询中的简单表达式过滤条件上创建 PCK。这种过滤条件一般形如 col op const，其中 col 为列名，op 为操作符 =、>、>=、<=、<，const 为常量值。

【建议】在满足上面条件的前提下，选择 distinct 值比较多的列上建 PCK。

唯一约束

【关注】行存表、列存表均支持唯一约束。

【建议】从命名上明确标识唯一约束，例如，命名为“UNI+构成字段”。

主键约束

【关注】行存表、列存表均支持主键约束。

【建议】从命名上明确标识主键约束，例如，将主键约束命名为 “PK+字段名”。

外键约束

【关注】行存表支持外键约束，列存表不支持外键约束。

【建议】从命名上明确标识外键约束，例如，将外键约束命名为 “FK+字段名”。

检查约束

【关注】行存表支持检查约束，而列存表不支持。

【建议】从命名上明确标识检查约束，例如，将检查约束命名为 “CK+字段名”。

3.2.6 视图和关联表设计

视图设计

【建议】除非视图之间存在强依赖关系，否则不建议视图嵌套。

【建议】视图定义中尽量避免排序操作。

关联表设计

【建议】表之间的关联字段应该尽量少。

【建议】关联字段的数据类型应该保持一致。

【建议】关联字段在命名上，应该可以明显体现出关联关系。例如，采用同样名称来命名。

3.2.7SQL 编写

DDL

【建议】在 ISSEDB 中，建议 DDL（建表、comments 等）操作统一执行，在批处理作业中尽量避免 DDL 操作。避免大量并发事务对性能的影响。

【建议】在非日志表（unlogged table）使用完后，立即执行数据清理（truncate）操作。因为在异常场景下，ISSEDB 不保证非日志表（unlogged table）数据的安全性。

【建议】临时表和非日志表的存储方式建议和基表相同。当基表为行存（列存）表时，临时表和非日志表也推荐创建为行存（列存）表，可以避免行列混合关联带来的高计算代价。

【建议】索引字段的总长度不超过 50 字节。否则，索引大小会膨胀比较严重，带来较大的存储开销，同时索引性能也会下降。

【建议】不要使用 DROP…CASCADE 方式删除对象，除非已经明确对象间的依赖关系，以免误删。

数据加载和卸载

【建议】在 insert 语句中显式给出插入的字段列表。例如：

INSERT INTO task(name,id,comment) VALUES ('task1','100','第 100 个任务');

【建议】在批量数据入库之后，或者数据增量达到一定阈值后，建议对表进行 analyze 操作，防止统计信息不准确而导致的执行计划劣化。

【建议】如果要清理表中的所有数据，建议使用 truncate table 方式，不要使用 delete table 方式。delete table 方式删除性能差，且不会释放那些已经删除了的数据占用的磁盘空间。

类型转换

【建议】在需要数据类型转换（不同数据类型进行比较或转换）时，使用强制类型转换，以防隐式类型转换结果与预期不符。

【建议】在查询中，对常量要显式指定数据类型，不要试图依赖任何隐式的数据类型转换。

【关注】若 sql_compatibility 参数设置为 A，在导入数据时，空字符串会自动转化为 NULL。如果需要保留空字符串需要 sql_compatibility 参数设置为 C。

查询操作

【建议】除 ETL 程序外，应该尽量避免向客户端返回大量结果集的操作。如果结果集过大，应考虑业务设计是否合理。

【建议】使用事务方式执行 DDL 和 DML 操作。例如，truncate table、update table、delete table、drop table 等操作，一旦执行提交就无法恢复。对于这类操作，建议使用事务进行封装，必要时可以进行回滚。

【建议】在查询编写时，建议明确列出查询涉及的所有字段，不建议使用“SELECT *”这种写法。一方面基于性能考虑，尽量减少查询输出列；另一方面避免增删字段对前端业务兼容性的影响。

【建议】在访问表对象时带上 schema 前缀，可以避免因 schema 切换导致访问到非预期的表。

【建议】超过 3 张表或视图进行关联（特别是 FULL JOIN）时，执行代价难以估算。建议使用 WITH TABLE AS 语句创建中间临时表的方式增加 SQL 语句的可读性。

【建议】尽量避免使用笛卡尔积和 FULL JOIN。这些操作会造成结果集的急剧膨胀，同时其执行性能也很低。

【关注】NULL 值的比较只能使用 IS NULL 或者 IS NOT NULL 的方式判断，其他任何形式的逻辑判断都返回 NULL。例如：NULL<>NULL、NULL=NULL 和 NULL<>1 返回结果都是 NULL，而不是期望的布尔值。

【关注】需要统计表中所有记录数时，不要使用 count(col) 来替代 count(*)。count(*)会统计 NULL 值（真实行数），而 count(col)不会统计。

【关注】在执行 count(col)时，将“值为 NULL”的记录行计数为 0。在执行 sum(col)时，当所有记录都为 NULL 时，最终将返回 NULL；当不全为 NULL 时，“值为 NULL”的记录行将被计数为 0。

【关注】count(多个字段)时，多个字段名必须用圆括号括起来。例如，count((col1,col2,col3))。注意：通过多字段统计行数时，即使所选字段都为 NULL，该行也被计数，效果与 count(*)一致。

【关注】count(distinct col)用来计算该列不重复的非 NULL 的数量，NULL 将不被计数。

【关注】count(distinct (col1,col2,…))用来统计多列的唯一值数量，当所有统计字段都为 NULL 时，也会被计数，同时这些记录被认为是相同的。

【建议】使用连接操作符“||”替换 concat 函数进行字符串连接。因为 concat 函

数生成的执行计划不能下推，导致查询性能严重劣化。
【建议】使用下面时间相关的宏替换 now 函数来获取当前时间。因为 now 函数生成的执行计划无法下推，导致查询性能严重劣化。

3.3 相关概念

数据库

数据库用于管理各类数据对象，与其他数据库隔离。创建数据对象时可以指定对应的表空间，如果不指定相应的表空间，相关的对象会默认保存在 PG_DEFAULT 空间中。数据库管理的对象可分布在多个表空间上。

表空间

在 ISSEDB 中，表空间是一个目录，可以存在多个，里面存储的是它所包含的数据库的各种物理文件。由于表空间是一个目录，仅是起到了物理隔离的作用，其管理功能依赖于文件系统。

模式

ISSEDB 的模式是对数据库做一个逻辑分割。所有的数据库对象都建立在模式下面。ISSEDB 的模式和用户是弱绑定的，所谓的弱绑定是指虽然创建用户的同时会自动创建一个同名模式，但用户也可以单独创建模式，并且为用户指定其他的模式。

用户和角色

ISSEDB 使用用户和角色来控制对数据库的访问。根据角色自身的设置不同，一个角色可以看做是一个数据库用户，或者一组数据库用户。在 ISSEDB 中角色和用户之间的区别只在于角色默认是没有 LOGIN 权限的。在 ISSEDB 中一个用户唯一对应一个角色，不过可以使用角色叠加来更灵活地进行管理。

事务管理

在事务管理上，ISSEDB 采取了 MVCC（多版本并发控制）结合两阶段锁的方式，其特点是读写之间不阻塞。ISSEDB 没有将历史版本数据统一存放，而是和当前元组的版本放在了一起。ISSEDB 没有回滚段的概念，但是为了定期清除历史版

本数据引入了一个 VACUUM 线程。一般情况下用户不用关注它，除非要做性能调优。此外，ISSEDB 是自动提交事务。

4 管理员指南

4.1 集群管理

查看集群状态

[omm@isoftnode1 script]$ gs_om -t status --detail

[Cluster State]

cluster_state : Normal

redistributing : No

current_az : AZ_ALL

[Datanode State]

node node_ip port instance state

---

1 isoftnode1 192.168.146.161 5432 6001 /opt/issedb1/data/d1 P Primary Normal

2 isoftnode4 192.168.146.164 5432 6002 /opt/issedb1/data/d4 S Standby Normal

停止集群

[omm@isoftnode1 script]$ gs_om -t stop;

Stopping cluster.

==========

Successfully stopped cluster.

==========

End stop cluster.

启动集群

[omm@isoftnode1 script]$ gs_om -t start;

Starting cluster.

===

[SUCCESS] isoftnode1

2023-07-28 14:56:33.308 64c366a1.1 [unknown] 140593643417856 [unknown] 0 dn_6001_6002 01000 0 [BACKEND] WARNING: could not create any HA TCP/IP sockets

2023-07-28 14:56:33.309 64c366a1.1 [unknown] 140593643417856 [unknown] 0 dn_6001_6002 01000 0 [BACKEND] WARNING: Failed to initialize the memory protect for g_instance.attr.attr_storage.cstore_buffers (16 Mbytes) or shared memory (1188 Mbytes) is larger.

[SUCCESS] isoftnode4

2023-07-28 14:56:34.716 64c366a2.1 [unknown] 140365746151680 [unknown] 0 dn_6001_6002 01000 0 [BACKEND] WARNING: could not create any HA TCP/IP sockets

2023-07-28 14:56:34.717 64c366a2.1 [unknown] 140365746151680 [unknown] 0 dn_6001_6002 01000 0 [BACKEND] WARNING: Failed to initialize the memory protect for g_instance.attr.attr_storage.cstore_buffers (16 Mbytes) or shared memory (1188 Mbytes) is larger.

===

Successfully started.

启停节点

[omm@isoftnode4 d4]$ gs_om -t stop -D /opt/issedb1/data/d4

Stopping cluster.

===

[GAUSS-53606]: Can not stop the database, the cmd is source /home/omm/.bashrc; python3 '/opt/issedb1/tool/script/local/StopInstance.py' -U omm -R /opt/issedb1/app -t 300 -D /opt/issedb1/data/d4 -m fast, Error:

[FAILURE] isoftnode1:

[GAUSS-53609]: Can not stop the database, Error:

[GAUSS-53610]: The input dataDir(/opt/issedb1/data/d4) may be incorrect..

[SUCCESS] isoftnode4:

.[omm@isoftnode4 d4]$ gs_om -t status --detail

[Cluster State]

cluster_state : Degraded

redistributing : No

current_az : AZ_ALL

[Datanode State]

node node_ip port instance state

---

1 isoftnode1 192.168.146.161 5432 6001 /opt/issedb1/data/d1 P Primary Normal

2 isoftnode4 192.168.146.164 5432 6002 /opt/issedb1/data/d4 S Down Manually stopped

[omm@isoftnode4 d4]$ gs_om -t start -D /opt/issedb1/data/d4

Starting cluster.

==========

[SUCCESS] isoftnode4

gs_om 作为服务端的工具，输出的日志更简洁一些，而 gs_ctl 作为数据库内部工具，会显示数据库内部的实际的信息，所以内容更多，也更底层一些。

主备切换

主备切换需要在备机操作

查看当前集群主备状态

```
[omm@isoftnode1 script]$ gs_om -t status --detail
[   Cluster State   ]

cluster_state   : Normal
redistributing  : No
current_az      : AZ_ALL

[  Datanode State   ]

      node     node_ip        port     instance        state
-----------------------------------------------------------------------------------------
1   isoftnode1  192.168.146.161  5432          6001  /opt/issedb1/data/d1     P  Primary
Normal
2   isoftnode4  192.168.146.164  5432          6002  /opt/issedb1/data/d4     S  Standby
Normal
```

查看备机状态是否可以切换

```
[omm@isoftnode4 d4]$ gs_ctl query -D /opt/issedb1/data/d4/
[2023-07-28      15:15:35.668][19156][][gs_ctl]:      gs_ctl      query      ,datadir      is
/opt/issedb1/data/d4
  HA state:
            local_role                : Standby
            static_connections          : 1
            db_state                  : Normal
            detail_information          : Normal

  Senders info:
No information
  Receiver info:
            receiver_pid              : 18494
            local_role                : Standby
```

peer_role : Primary

peer_state : Normal

state : Normal

sender_sent_location : 0/4001BC8

sender_write_location : 0/4001BC8

sender_flush_location : 0/4001BC8

sender_replay_location : 0/4001BC8

receiver_received_location : 0/4001BC8

receiver_write_location : 0/4001BC8

receiver_flush_location : 0/4001BC8

receiver_replay_location : 0/4001B28

sync_percent : 100%

channel : 192.168.146.164:47386<--192.168.146.161:5433

备机正常可以切换

主备切换

[omm@isoftnode4 d4]$ gs_ctl switchover -D /opt/issedb1/data/d4

[2023-07-28 15:16:55.469][19198][][gs_ctl]: gs_ctl switchover ,datadir is /opt/issedb1/data/d4

[2023-07-28 15:16:55.469][19198][][gs_ctl]: switchover term (1)

[2023-07-28 15:16:55.474][19198][][gs_ctl]: waiting for server to switchover.....

[2023-07-28 15:17:00.592][19198][][gs_ctl]: done

[2023-07-28 15:17:00.592][19198][][gs_ctl]: switchover completed (/opt/issedb1/data/d4)

再次查看集群状态

[omm@isoftnode1 data]$ gs_om -t status --detail

[Cluster State]

cluster_state : Degraded

redistributing : No

current_az : AZ_ALL

```
[  Datanode State   ]

      node     node_ip     port     instance       state
-----------------------------------------------------------------------------------------
1   isoftnode1 192.168.146.161 5432             6001 /opt/issedb1/data/d1    P Primary
Normal
2   isoftnode4 192.168.146.164 5432              6002 /opt/issedb1/data/d4    S Down
Manually stopped
[omm@isoftnode1 data]$
[omm@isoftnode1 data]$
[omm@isoftnode1 data]$ gs_om -t status --detail
[   Cluster State   ]

cluster_state   : Normal
redistributing  : No
current_az      : AZ_ALL

[  Datanode State   ]

      node     node_ip       port     instance        state
-----------------------------------------------------------------------------------------
1   isoftnode1  192.168.146.161  5432          6001  /opt/issedb1/data/d1     P  Standby
Normal
2   isoftnode4  192.168.146.164  5432          6002  /opt/issedb1/data/d4     S  Primary
Normal
```

切换成功

4.2 备份恢复

准备工作(测试数据，备份目录)

创建逻辑备份的存储目录

[omm@isoftnode2 ~]$ mkdir /opt/backup

[omm@isoftnode2 ~]$ cd /opt/backup/

--创建备份恢复用户，需要具有 super 或者 sysadmin 权限

issedb=# create user test IDENTIFIED BY 'huawei@1234' sysadmin ;

CREATE ROLE

--创建恢复测试数据库 testdb

issedb=#CREATE TABLESPACE test_tbs RELATIVE LOCATION 'tablespace/test_tbs1';

CREATE DATABASE testdb WITH TABLESPACE = test_tbs;

CREATE TABLESPACE

issedb=# CREATE DATABASE testdb WITH TABLESPACE = test_tbs;

CREATE DATABASE

--在 testdb 数据库里，创建测试表 test1、test2：

testdb=# CREATE TABLE test1(col int);

CREATE TABLE

testdb=# CREATE TABLE test2(col int);

CREATE TABLE

--查看数据

testdb=# \d

List of relations

 Schema | Name | Type | Owner | Storage

--------+-------+-------+-------+---------------------------------

 public | test1 | table | omm | {orientation=row,compression=no}

 public | test2 | table | omm | {orientation=row,compression=no}

(2 rows)

gs_dump 逻辑备份和恢复（sql 格式）

--使用 test 用户，备份数据库 testdb：

[omm@isoftnode2 backup]$ gs_dump -U test -W huawei@1234 -f

/opt/backup/backup.sql -p 26000 testdb -F p

gs_dump[port='26000'][testdb][2023-07-27 19:53:31]: The total objects number is 432.

gs_dump[port='26000'][testdb][2023-07-27 19:53:31]: [100.00%] 432 objects have been dumped.

gs_dump[port='26000'][testdb][2023-07-27 19:53:31]: dump database testdb successfully

gs_dump[port='26000'][testdb][2023-07-27 19:53:31]: total time: 1563 ms

逻辑恢复：

--使用用户 test，执行用 gs_dump 生成的 sql 脚本，将数据恢复到 template1 数据库中：

[omm@isoftnode2 backup]$ gsql -d template1 -U test -W huawei@1234 -p 26000 -f /opt/backup/backup.sql

SET

SET

SET

SET

SET

SET

SET

CREATE SCHEMA

ALTER SCHEMA

SET

SET

SET

CREATE TABLE

ALTER TABLE

CREATE TABLE

ALTER TABLE

REVOKE

REVOKE

GRANT

GRANT

total time: 14 ms

验证逻辑备份恢复

```
[omm@isoftnode2 script]$ issql -d testdb -p 26000 -r
issql ((isoftDB 5.0.0 build 4ec82cbe) compiled at 2023-07-27 09:58:20 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

testdb=# \d
                    List of relations
 Schema | Name  | Type  | Owner |             Storage
--------+-------+-------+-------+---------------------------------
 public | test1 | table | omm   | {orientation=row,compression=no}
 public | test2 | table | omm   | {orientation=row,compression=no}
(2 rows)

testdb=# \q
[omm@isoftnode2 script]$ issql -d template1 -p 26000 -r
issql ((isoftDB 5.0.0 build 4ec82cbe) compiled at 2023-07-27 09:58:20 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

template1=# \d
                    List of relations
 Schema | Name  | Type  | Owner |             Storage
--------+-------+-------+-------+---------------------------------
 public | test1 | table | omm   | {orientation=row,compression=no}
 public | test2 | table | omm   | {orientation=row,compression=no}
(2 rows)
```

gs_backup 备份恢复集群信息

备份集群二进制和参数文件

[omm@isoftnode2 ~]$ gs_backup -t backup --backup-dir=/opt/backup --all

Parsing configuration files.

Successfully parsed the configuration file.

Performing remote backup.

Remote backup succeeded.

Successfully backed up cluster files.

[omm@isoftnode2 backup]$ ll

total 454084

-rw-------. 1 omm dbgrp   1355 Jul 27 19:53 backup.sql

-rw-------. 1 omm dbgrp 464916480 Jul 28 09:16 binary.tar

-rw-------. 1 omm dbgrp   61440 Jul 28 09:16 parameter.tar

恢复参数文件

[root@isoftnode2 db1]# ll /opt/issedb1/data/db1/pg_hba.conf /opt/issedb1/data/db1/postgresql.conf
-rw-------. 1 omm dbgrp 4587 Jul 27 16:23 /opt/issedb1/data/db1/pg_hba.conf
-rw-------. 1 omm dbgrp 36085 Jul 27 16:23 /opt/issedb1/data/db1/postgresql.conf
[root@isoftnode2 db1]# mv /opt/issedb1/data/db1/pg_hba.conf /opt/issedb1/data/db1/pg_hba.conf_bak
[root@isoftnode2 db1]# mv /opt/issedb1/data/db1/postgresql.conf /opt/issedb1/data/db1/postgresql.conf_bak
[root@isoftnode2 db1]# ll /opt/issedb1/data/db1/pg_hba.conf /opt/issedb1/data/db1/postgresql.conf
ls: cannot access /opt/issedb1/data/db1/pg_hba.conf: No such file or directory
ls: cannot access /opt/issedb1/data/db1/postgresql.conf: No such file or directory
[omm@isoftnode2 backup]$ gs_backup -t restore --backup-dir /opt/backup/ -h isoftnode2 --parameter -l /opt/backup/backup.log
Parsing configuration files.
Successfully parsed the configuration file.
Performing remote restoration.
Successfully restored cluster files.

[root@isoftnode2 db1]# ll /opt/issedb1/data/db1/pg_hba.conf /opt/issedb1/data/db1/postgresql.conf
-rw-------. 1 omm dbgrp 4587 Jul 27 16:23 /opt/issedb1/data/db1/pg_hba.conf
-rw-------. 1 omm dbgrp 36085 Jul 27 16:23 /opt/issedb1/data/db1/postgresql.conf

gs_basebackup 物理备份

[omm@isoftnode2 back]$ gs_basebackup -D /opt/back -p 26000
INFO: The starting position of the xlog copy of the full build is: 0/7000028. The slot minimum LSN is: 0/0. The disaster slot minimum LSN is: 0/0. The logical slot

minimum LSN is: 0/0.

[2023-07-28 09:43:59]:begin build tablespace list

[2023-07-28 09:43:59]:finish build tablespace list

[2023-07-28 09:43:59]:begin get xlog by xloistream

[2023-07-28 09:43:59]: check identify system success

[2023-07-28 09:43:59]: send START_REPLICATION 0/7000000 success

[2023-07-28 09:43:59]: keepalive message is received

[2023-07-28 09:43:59]: keepalive message is received

[2023-07-28 09:44:04]:gs_basebackup: base backup successfully

```
[omm@isoftnode2 back]$ ll
total 5008
-rw-------. 1 omm dbgrp     206 Jul 28 09:43 backup_label
drwx------. 5 omm dbgrp      41 Jul 28 09:44 base
-rw-------. 1 omm dbgrp    4399 Jul 28 09:44 cacert.pem
drwx------. 2 omm dbgrp    4096 Jul 28 09:44 global
-rw-------. 1 omm dbgrp 4915200 Jul 28 09:44 gswlm_userinfo.cfg
-rw-------. 1 omm dbgrp   21012 Jul 28 09:44 mot.conf
drwx------. 2 omm dbgrp      26 Jul 28 09:43 pg_clog
drwx------. 2 omm dbgrp      26 Jul 28 09:43 pg_csnlog
-rw-------. 1 omm dbgrp       0 Jul 28 09:44 pg_ctl.lock
drwx------. 2 omm dbgrp       6 Jul 28 09:44 pg_errorinfo
-rw-------. 1 omm dbgrp    4587 Jul 28 09:44 pg_hba.conf
-rw-------. 1 omm dbgrp    4587 Jul 28 09:44 pg_hba.conf_bak
-rw-------. 1 omm dbgrp    4587 Jul 28 09:44 pg_hba.conf.bak
-rw-------. 1 omm dbgrp    1024 Jul 28 09:44 pg_hba.conf.lock
-rw-------. 1 omm dbgrp    1636 Jul 28 09:44 pg_ident.conf
drwx------. 4 omm dbgrp      39 Jul 28 09:44 pg_llog
drwx------. 3 omm dbgrp      24 Jul 28 09:43 pg_location
drwx------. 2 omm dbgrp      35 Jul 28 09:44 pg_logical
drwx------. 4 omm dbgrp      36 Jul 28 09:43 pg_multixact
drwx------. 2 omm dbgrp      26 Jul 28 09:43 pg_notify
drwx------. 2 omm dbgrp       6 Jul 28 09:44 pg_replslot
drwx------. 2 omm dbgrp       6 Jul 28 09:43 pg_serial
drwx------. 2 omm dbgrp       6 Jul 28 09:43 pg_snapshots
drwx------. 2 omm dbgrp      25 Jul 28 09:44 pg_stat_tmp
drwx------. 2 omm dbgrp      58 Jul 28 09:44 pg_tblspc
drwx------. 2 omm dbgrp       6 Jul 28 09:43 pg_twophase
-rw-------. 1 omm dbgrp       4 Jul 28 09:44 PG_VERSION
drwx------. 3 omm dbgrp      60 Jul 28 09:44 pg_xlog
-rw-------. 1 omm dbgrp   36085 Jul 28 09:44 postgresql.conf
-rw-------. 1 omm dbgrp   36085 Jul 28 09:44 postgresql.conf_bak
-rw-------. 1 omm dbgrp   36085 Jul 28 09:44 postgresql.conf.guc.bak
-rw-------. 1 omm dbgrp    1024 Jul 28 09:44 postgresql.conf.lock
-rw-------. 1 omm dbgrp       0 Jul 28 09:44 postmaster.pid.lock
-rw-------. 1 omm dbgrp    4402 Jul 28 09:44 server.crt
-rw-------. 1 omm dbgrp    1766 Jul 28 09:44 server.key
-rw-------. 1 omm dbgrp      56 Jul 28 09:44 server.key.cipher
-rw-------. 1 omm dbgrp      24 Jul 28 09:44 server.key.rand
drwx------. 5 omm dbgrp      67 Jul 28 09:44 undo
[omm@isoftnode2 back]$
```

gs_probackup 增量备份和恢复

1、初始化备份目录

初始化备份目录。执行如下命令在指定的目录下创建 backups/和 wal/子目录，分别用于存放备份文件和 WAL 文件,例如指定目录为/opt/probackup/

[omm@isoftnode2 opt]$ gs_probackup init -B /opt/probackup

INFO: Backup catalog '/opt/probackup' successfully inited

[omm@isoftnode2 probackup]$ ll

total 0

drwx------. 2 omm dbgrp 6 Jul 28 10:37 backups

drwx------. 2 omm dbgrp 6 Jul 28 10:37 wal

2、新增备份实例

添加一个新的备份实例。gs_probackup 可以在同一个备份目录下存放多个数据库实例的备份。例如数据目录为/opt/issedb1/data/db1。

[omm@isoftnode2 probackup]$ gs_probackup add-instance -B /opt/probackup/ -D /opt/issedb1/data/db1/ --instance instance_local

INFO: Instance 'instance_local' successfully inited

[omm@isoftnode2 backups]$ ll

total 0

drwx------. 3 omm dbgrp 45 Jul 28 11:44 instance_local

3、全量备份

创建指定实例的备份。在进行增量备份之前，必须至少创建一次全量备份。

[omm@isoftnode2 instance_local]$ gs_probackup backup -B /opt/probackup/ --instance instance_local -d testdb -b FULL

INFO: Backup start, gs_probackup version: 2.4.2, instance: instance_local, backup ID: RYHLGK, backup mode: FULL, wal mode: STREAM, remote: false, compress-algorithm: none, compress-level: 1

LOG: Backup destination is initialized

ERROR: could not connect to database testdb: connect to server failed: No such file or directory

WARNING: Backup RYHLGK is running, setting its status to ERROR

出现以上报错是因为没有在生成的 pg_probackup.conf 配置文件中添加数据库连

接信息

执行下面语句在配置文件中添加数据库连接信息

[omm@isoftnode2 instance_local]$ gs_probackup set-config -B /opt/probackup/ --instance=instance_local -d testdb -p 26000

[omm@isoftnode2 instance_local]$ cat pg_probackup.conf

Backup instance information

pgdata = /opt/issedb1/data/db1

system-identifier = 3423299692713438

Connection parameters

pgdatabase = testdb

pgport = 26000

[omm@isoftnode2 instance_local]$ gs_probackup backup -B /opt/probackup/ --instance instance_local -d testdb -b FULL

INFO: Backup start, gs_probackup version: 2.4.2, instance: instance_local, backup ID: RYHMEW, backup mode: FULL, wal mode: STREAM, remote: false, compress-algorithm: none, compress-level: 1

LOG: Backup destination is initialized

LOG: This issedb instance was initialized with data block checksums. Data block corruption will be detected

LOG: Database backup start

LOG: started streaming WAL at 0/9000000 (timeline 1)

[2023-07-28 11:44:57]: check identify system success

[2023-07-28 11:44:57]: send START_REPLICATION 0/9000000 success

[2023-07-28 11:44:57]: keepalive message is received

INFO: PGDATA size: 620MB

INFO: Start transferring data files

[2023-07-28 11:44:58]: keepalive message is received

LOG: Creating page header map "/opt/probackup/backups/instance_local/RYHMEW/page_header_map"

[2023-07-28 11:45:03]: keepalive message is received

[2023-07-28 11:45:08]: keepalive message is received

[2023-07-28 11:45:13]: keepalive message is received

[2023-07-28 11:45:18]: keepalive message is received

[2023-07-28 11:45:23]: keepalive message is received

[2023-07-28 11:45:28]: keepalive message is received

[2023-07-28 11:45:33]: keepalive message is received

[2023-07-28 11:45:38]: keepalive message is received

[2023-07-28 11:45:44]: keepalive message is received

[2023-07-28 11:45:49]: keepalive message is received

[2023-07-28 11:45:55]: keepalive message is received

[2023-07-28 11:46:01]: keepalive message is received

[2023-07-28 11:46:06]: keepalive message is received

[2023-07-28 11:46:11]: keepalive message is received

[2023-07-28 11:46:17]: keepalive message is received

INFO: Data files are transferred, time elapsed: 1m:20s

INFO: wait for pg_stop_backup()

INFO: pg_stop backup() successfully executed

LOG: stop_lsn: 0/9000308

LOG: Looking for LSN 0/9000308 in segment: 000000010000000000000009

INFO: Wait for LSN 0/9000308 in streamed WAL segment /opt/probackup/backups/instance_local/RYHMEW/database/pg_xlog/000000010000000000000009

LOG: Found WAL segment: /opt/probackup/backups/instance_local/RYHMEW/database/pg_xlog/000000010000000000000009

LOG: Thread [0]: Opening WAL segment "/opt/probackup/backups/instance_local/RYHMEW/database/pg_xlog/000000010000000000000009"

LOG: Found LSN: 0/9000308

LOG: finished streaming WAL at 0/A000000 (timeline 1)

LOG: Getting the Recovery Time from WAL

LOG: Thread [0]: Opening WAL segment "/opt/probackup/backups/instance_local/RYHMEW/database/pg_xlog/000000010000000000000009"

INFO: Syncing backup files to disk

INFO: Backup files are synced, time elapsed: 0

INFO: Validating backup RYHMEW

INFO: Backup RYHMEW data files are valid

INFO: Backup RYHMEW resident size: 636MB

INFO: Backup RYHMEW completed

备份成功

查看备份状态信息

[omm@isoftnode2 instance_local]$ gs_probackup show -B /opt/probackup/ --instance instance_local

==========

Instance Version ID Recovery Time Mode WAL Mode TLI Time Data WAL Zratio Start LSN Stop LSN Type Status

==========

instance_local 9.2 RYHNH7 2023-07-28 12:07:58+08 FULL STREAM 1/0 6s 620MB 16MB 1.00 0/D000028 0/D0001E8 FILE OK

4、对数据库做 DML 操作

对数据库进行 DML 操作后做增量备份

[omm@isoftnode2 ~]$ gsql -d testdb -p 26000 -r

gsql ((issedb 5.0.0 build 4ec82cbe) compiled at 2023-07-27 09:58:20 commit 0 last mr)

Non-SSL connection (SSL connection is recommended when requiring high-security)

Type "help" for help.

```
testdb=# \d
                        List of relations
 Schema | Name  | Type  | Owner |          Storage
--------+-------+-------+-------+---------------------------------
 public | test1 | table | omm   | {orientation=row,compression=no}
 public | test2 | table | omm   | {orientation=row,compression=no}
(2 rows)

testdb=# \d test1
       Table "public.test1"
 Column |  Type   | Modifiers
--------+---------+-----------
 col    | integer |

testdb=# insert into test1 values(111);
INSERT 0 1
```

5、增量备份

```
[omm@isoftnode2 instance_local]$ gs_probackup  backup -B /opt/probackup/ --instance instance_local -d testdb -b PTRACK
INFO: Backup start, gs_probackup version: 2.4.2, instance: instance_local, backup ID: RYHN0T, backup mode: PTRACK, wal mode: STREAM, remote: false, compress-algorithm: none, compress-level: 1
LOG: Backup destination is initialized
LOG: This issedb instance was initialized with data block checksums. Data block corruption will be detected
LOG: Database backup start
LOG: Latest valid FULL backup: RYHMEW
INFO: Parent backup: RYHMEW
LOG: started streaming WAL at 0/B000000 (timeline 1)
```

[2023-07-28 11:58:06]: check identify system success

[2023-07-28 11:58:06]: send START_REPLICATION 0/B000000 success

[2023-07-28 11:58:06]: keepalive message is received

INFO: PGDATA size: 620MB

LOG: Current tli: 1

LOG: Parent start_lsn: 0/9000028

LOG: start_lsn: 0/B000028

INFO: Extracting pagemap of changed blocks

INFO: change bitmap start lsn location is 0/9000028

INFO: change bitmap end lsn location is 00000000/0B000028

INFO: Pagemap successfully extracted, time elapsed: 0 sec

[2023-07-28 11:58:06]: keepalive message is received

INFO: Start transferring data files

[2023-07-28 11:58:11]: keepalive message is received

[2023-07-28 11:58:16]: keepalive message is received

INFO: Data files are transferred, time elapsed: 13s

INFO: wait for pg_stop_backup()

INFO: pg_stop backup() successfully executed

LOG: stop_lsn: 0/B0001E8

LOG: Looking for LSN 0/B0001E8 in segment: 000000010000000000000000B

INFO: Wait for LSN 0/B0001E8 in streamed WAL segment /opt/probackup/backups/instance_local/RYHN0T/database/pg_xlog/000000010000000000000000B

[2023-07-28 11:58:24]: keepalive message is received

LOG: Found WAL segment: /opt/probackup/backups/instance_local/RYHN0T/database/pg_xlog/000000010000000000000000B

LOG: Thread [0]: Opening WAL segment "/opt/probackup/backups/instance_local/RYHN0T/database/pg_xlog/000000010000000000000000B"

LOG: Found LSN: 0/B0001E8

LOG: finished streaming WAL at 0/C000000 (timeline 1)

LOG: Getting the Recovery Time from WAL

LOG: Thread [0]: Opening WAL segment "/opt/probackup/backups/instance_local/RYHN0T/database/pg_xlog/000000010000000000000000B"

INFO: Syncing backup files to disk

INFO: Backup files are synced, time elapsed: 0

INFO: Validating backup RYHN0T

INFO: Backup RYHN0T data files are valid

INFO: Backup RYHN0T resident size: 273MB

INFO: Backup RYHN0T completed

查看备份状态

[omm@isoftnode2 instance_local]$ gs_probackup show -B /opt/probackup/ --instance instance_local

```
================================================================================================================================
 Instance        Version  ID      Recovery Time           Mode    WAL Mode  TLI  Time   Data   WAL  Zratio  Start LSN  Stop LSN   Type  Status
================================================================================================================================
 instance_local  9.2      RYHNI7  2023-07-28 12:08:32+08  PTRACK  STREAM    1/1  6s     257MB  16MB   1.00  0/F000028  0/F0001E8  FILE  OK
 instance_local  9.2      RYHNH7  2023-07-28 12:07:58+08  FULL    STREAM    1/0  6s     620MB  16MB   1.00  0/D000028  0/D0001E8  FILE  OK
```

6、将增量备份和全量备份合并。

将指定的增量备份与其父完全备份之间的所有增量备份合并到父完全备份。父完全备份将接收所有合并的数据，而已合并的增量备份将作为冗余被删除。

[omm@isoftnode2 instance_local]$ gs_probackup merge -B /opt/probackup/ --instance instance_local -i RYHNI7

INFO: Merge started

INFO: Merging backup RYHNI7 with parent chain

INFO: Validate parent chain for backup RYHNI7

INFO: Validating backup RYHNH7

INFO: Backup RYHNH7 data files are valid

INFO: Validating backup RYHNI7

INFO: Backup RYHNI7 data files are valid

LOG: Restore directories and symlinks... in /opt/probackup/backups/instance_local/RYHNH7/database

INFO: Start merging backup files

LOG: Creating page header map "/opt/probackup/backups/instance_local/RYHNH7/page_header_map_tmp"

INFO: Backup files are successfully merged, time elapsed: 8s

INFO: Delete: RYHNI7 2023-07-28 12:08:32+08

LOG: Rename /opt/probackup/backups/instance_local/RYHNH7 to /opt/probackup/backups/instance_local/RYHNI7

INFO: Rename merged full backup RYHNH7 to RYHNI7

INFO: Validating backup RYHNI7

INFO: Backup RYHNI7 data files are valid

INFO: Merge of backup RYHNI7 completed

[omm@isoftnode2 instance_local]$ ll

total 4

-rw-------. 1 omm dbgrp 159 Jul 28 11:44 pg_probackup.conf

drwx------. 3 omm dbgrp 97 Jul 28 12:10 RYHNI7

7、删除 testdb 库,并恢复

[omm@isoftnode2 ~]$ gsql -d postgres -p 26000 -r

gsql ((issedb 5.0.0 build 4ec82cbe) compiled at 2023-07-27 09:58:20 commit 0 last mr)

Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

issedb=# drop database testdb;
DROP DATABASE
[omm@isoftnode2 ~]$ gsql -d testdb -p 26000 -r
gsql: FATAL: database "testdb" does not exist
使用 restore 子命令前，应先停止 ISSEDB 进程，并清空 db1 目录
[omm@isoftnode2 ~]$ gs_ctl stop -D /opt/issedb1/data/db1
[2023-07-28 12:18:57.916][79294][][gs_ctl]: gs_ctl stopped ,datadir is /opt/issedb1/data/db1
waiting for server to shut down..... done
server stopped
[root@isoftnode2 ~]# cd /opt/issedb1/data
[root@isoftnode2 data]# mv db1 db1_bak

执行 restore 恢复到误删前的数据状态
[omm@isoftnode2 instance_local]$ gs_probackup restore -B /opt/probackup/ --instance instance_local -D /opt/issedb1/data/db1 -i RYHNI7
LOG: Restore begin.
LOG: check tablespace directories of backup RYHNI7
LOG: check external directories of backup RYHNI7
INFO: Validating backup RYHNI7
INFO: Backup RYHNI7 data files are valid
LOG: Thread [1]: Opening WAL segment "/opt/probackup/backups/instance_local/RYHNI7/database/pg_xlog/000000010000000000000000F"
INFO: Backup RYHNI7 WAL segments are valid
INFO: Backup RYHNI7 is valid.
INFO: Restoring the database from backup at 2023-07-28 12:08:31+08
LOG: Restore directories and symlinks... in /opt/issedb1/data/db1

INFO: Start restoring backup files. PGDATA size: 636MB

LOG: Start thread 1

INFO: Backup files are restored. Transfered bytes: 1000MB, time elapsed: 28s

INFO: Restore incremental ratio (less is better): 157% (1000MB/636MB)

INFO: Syncing restored files to disk

INFO: Restored backup files are synced, time elapsed: 1s

INFO: Restore of backup RYHNI7 completed.

启动数据库并测试数据库数据是否恢复

[omm@isoftnode2 ~]$ gs_ctl start -D /opt/issedb1/data/db1

[2023-07-28 12:22:08.206][79316][][gs_ctl]: gs_ctl started,datadir is /opt/issedb1/data/db1

[2023-07-28 12:22:12.404][79316][][gs_ctl]: waiting for server to start...

.0 LOG: [Alarm Module]can not read GAUSS_WARNING_TYPE env.

0 LOG: [Alarm Module]Host Name: isoftnode2

0 LOG: [Alarm Module]Host IP: isoftnode2. Copy hostname directly in case of taking 10s to use 'gethostbyname' when /etc/hosts does not contain <HOST IP>

0 LOG: [Alarm Module]Cluster Name: dbCluster

0 LOG: [Alarm Module]Invalid data in AlarmItem file! Read alarm English name failed! line: 57

0 WARNING: failed to open feature control file, please check whether it exists: FileName=ISSEDB.version, Errno=2, Errmessage=No such file or directory.

0 WARNING: failed to parse feature control file: issedb.version.

0 WARNING: Failed to load the product control file, so issedb cannot distinguish product version.

0 LOG: bbox_dump_path is set to /opt/issedb1/corefile/

2023-07-28 12:22:13.187 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 DB010 0 [REDO] LOG: Recovery parallelism, cpu count = 2, max = 4, actual = 2
2023-07-28 12:22:13.187 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 DB010 0 [REDO] LOG: ConfigRecoveryParallelism, true_max_recovery_parallelism:4, max_recovery_parallelism:4
2023-07-28 12:22:13.399 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 00000 0 [BACKEND] LOG: [Alarm Module]can not read GAUSS_WARNING_TYPE env.

2023-07-28 12:22:13.399 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 00000 0 [BACKEND] LOG: [Alarm Module]Host Name: isoftnode2

2023-07-28 12:22:13.399 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 00000 0 [BACKEND] LOG: [Alarm Module]Host IP: isoftnode2. Copy hostname directly in case of taking 10s to use 'gethostbyname' when /etc/hosts does not contain <HOST IP>

2023-07-28 12:22:13.399 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 00000 0 [BACKEND] LOG: [Alarm Module]Cluster Name: dbCluster

2023-07-28 12:22:13.399 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 00000 0 [BACKEND] LOG: [Alarm Module]Invalid data in AlarmItem file! Read alarm English name failed! line: 57

.2023-07-28 12:22:13.416 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 00000 0 [BACKEND] LOG: loaded library "security_plugin"
2023-07-28 12:22:13.418 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 01000 0 [BACKEND] WARNING: could not create any HA TCP/IP sockets
2023-07-28 12:22:13.418 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 01000 0 [BACKEND] WARNING: could not create any HA TCP/IP sockets

2023-07-28 12:22:13.421 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 00000 0 [BACKEND] LOG: InitNuma numaNodeNum: 1 numa_distribute_mode: none inheritThreadPool: 0.

2023-07-28 12:22:13.421 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 01000 0 [BACKEND] WARNING: Failed to initialize the memory protect for g_instance.attr.attr_storage.cstore_buffers (16 Mbytes) or shared memory (1188 Mbytes) is larger.

2023-07-28 12:22:14.278 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 00000 0 [CACHE] LOG: set data cache size(12582912)

2023-07-28 12:22:17.650 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 00000 0 [SEGMENT_PAGE] LOG: Segment-page constants: DF_MAP_SIZE: 8156, DF_MAP_BIT_CNT: 65248, DF_MAP_GROUP_EXTENTS: 4175872, IPBLOCK_SIZE: 8168, EXTENTS_PER_IPBLOCK: 1021, IPBLOCK_GROUP_SIZE: 4090, BMT_HEADER_LEVEL0_TOTAL_PAGES: 8323072, BktMapEntryNumberPerBlock: 2038, BktMapBlockNumber: 25, BktBitMaxMapCnt: 512

2023-07-28 12:22:18.419 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 00000 0 [BACKEND] LOG: issedb: fsync file "/opt/issedb1/data/db1/issedb.state.temp" success

2023-07-28 12:22:18.419 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 00000 0 [BACKEND] LOG: create issedb state file success: db state(STARTING_STATE), server mode(Normal), connection index(1)

2023-07-28 12:22:18.443 64c34274.1 [unknown] 139730238255360 [unknown] 0 dn_6001 00000 0 [BACKEND] LOG: max_safe_fds = 976, usable_fds = 1000, already_open = 14

bbox_dump_path is set to /opt/issedb1/corefile/

.

[2023-07-28 12:22:21.110][79316][][gs_ctl]: done

[2023-07-28 12:22:21.110][79316][][gs_ctl]: server started (/opt/issedb1/data/db1)

[omm@isoftnode2 ~]$

[omm@isoftnode2 ~]$ gsql -d testdb -p 26000 -r

gsql ((issedb 5.0.0 build 4ec82cbe) compiled at 2023-07-27 09:58:20 commit 0 last

mr)

Non-SSL connection (SSL connection is recommended when requiring high-security)

Type "help" for help.

testdb=# select * from test1;

col

111

(1 row)

恢复成功

4.3 数据导入与导出

copy 导入导出

表数据导出到 txt 文件

testdb=# copy test to '/opt/data/test' with (delimiter '|',format 'TEXT');

COPY 2

[omm@isoftnode2 data]$ cat test

1|100

2|20

testdb=# copy (select * from test where id=1) to '/opt/data/test_id' with (delimiter '|',format 'TEXT');

COPY 1

[omm@isoftnode2 data]$ cat test_id

1|100

导入

testdb=# select * from test3;

id | value

----+-------

```
(0 rows)
testdb=# copy test3 from '/opt/data/test' with (delimiter '|',format 'TEXT');
COPY 2
testdb=# select * from test3;
 id | value
----+-------
  1 |   100
  2 |    20
(2 rows)
```

5 工具命令参考

5.1 gsql

gsql 是 ISSEDB 提供在命令行下运行的数据库连接工具，可以通过此工具连接服务器并对其进行操作和维护，除了具备操作数据库的基本功能，gsql 还提供了若干高级特性，便于用户使用。

5.1.1 基本功能

连接数据库： 详细操作请参见《快速入门》中“访问 ISSEDB > 使用 gsql 访问 ISSEDB”章节。

执行 SQL 语句： 支持交互式地键入并执行 SQL 语句，也可以执行一个文件中指定的 SQL 语句。

执行元命令： 元命令可以帮助管理员查看数据库对象的信息、查询缓存区信息、格式化 SQL 输出结果，以及连接到新的数据库等。元命令的详细说明请参见元命令参考。

5.1.2 高级特性

gsql 的高级特性如表 1 所示。

表 1 gsql 高级特性

特性名称	描述
变量	gsql 提供类似于 Linux 的 shell 命令的变量特性，可以使用 gsql 的原命令\set 设置一个变量，格式如下： \set varname value 要删除一个变量请使用如下方式： \unset varname 说明：

特性名称	描述
	变量只是简单的名称/值对，这里的值可以是任意长度。 变量名称必须由字母（包括非拉丁字母）、数字和下划线组成，且对大小写敏感。 如果使用\set varname 的格式（不带第二个参数），则只是设置这个变量而没有给变量赋值。 可以使用不带参数的\set 来显示所有变量的值。 变量的示例和详细说明请参见**变量**。
SQL 代换	利用 gsql 的变量特性，可以将常用的 SQL 语句设置为变量，以简化操作。 SQL 代换的示例和详细说明请参见 **SQL 代换**。
自定义提示符	gsql 使用的提示符支持用户自定义。可以通过修改 gsql 预留的三个变量 PROMPT1、PROMPT2、PROMPT3 来改变提示符。 这三个变量的值可以用户自定义，也可以使用 gsql 预定义的值。详细请参见**提示符**。
命令自动补齐	根据 ISSEDB 语法规则，gsql 支持使用 Tab 键进行命令的自动补齐，当编译时指定了选项--with-readline，且客户端连接时指定“-r”参数，此功能被打开。例如，crea 后键入 Tab，gsql 会将其补齐为 create。 说明： 支持数据库 SQL 关键字如 select、create、table 等。 支持表名、视图名等自定义标识符的补齐。 元命令选项'S'、'+'不支持自动补齐。 对表进行补齐时，只有前缀是“pg_”才会补齐系统表。 不支持建表时字段类型的补齐。 select 后不支持任何补齐。 不支持常量与宏的自动补齐。 select * from a,b... 不支持第二个开始表的自动补齐, insert into t1 (col1, col2, ...) 不支持第二个列的自动补齐。 不支持 create tablespace 语句 with 以及 with 后参数的自动补齐。

特性名称	描述
	创建索引不支持 local、global 的自动补齐，修改索引不支持 rebuild 自动补齐。 set 语句仅支持自动补全 USER 和 SUPERUSER 级别的参数。 不支持 if exists 的自动补齐。 不支持表名.列名的自动补齐，如 alter sequence <name> owned by tableName.colName，owned by。 不支持自定义操作符自动补齐。使用复制粘贴这种方式输入命令，如果粘贴的命令里面有 TAB 键有可能会使输入命令的格式错乱，无法正常执行。 "\t\n@$><=;|&{() "这些特殊字符在 sql 语句中具有固定含义。如果自定义表名中包含这些特殊字符，那么输入的 sql 语句从这些字符开始不支持自动补齐。
客户端操作历史记录	gsql 支持客户端操作历史记录，当客户端连接时指定“-r”参数，此功能被打开。可以通过\set 设置记录历史的条数，例如，\set HISTSIZE 50，将记录历史的条数设置为 50，\set HISTSIZE 0，不记录历史。 说明： 客户端操作历史记录条数默认设置为 32 条，最多支持记录 500 条。当客户端交互式输入包含中文字符时，只支持 UTF-8 的编码环境。 出于安全考虑，将包含 PASSWORD、IDENTIFIED 敏感词的记录识别为敏感信息，不会记录到历史信息中，即不能通过上下翻回显。

5.1.3 使用指导

前提条件

连接数据库时使用的用户需要具备访问数据库的权限。

背景信息

使用 gsql 命令可以连接远程数据库服务。连接远程数据库服务时，需要在服务器上设置允许远程连接，详细操作请参见《开发者指南》中“数据库使用 > 连接数据库 > 使用 gsql 连接 > 远程连接数据库”章节。

操作步骤

（1）使用 gsql 连接到 ISSEDB 服务器。

gsql 工具使用-d 参数指定目标数据库名、-U 参数指定数据库用户名、-h 参数指定主机名、-p 参数指定端口号信息。

（2）执行 SQL 语句。

（3）执行 gsql 元命令。

注意事项

1、gsql 元命令的格式是反斜杠后面紧跟一个动词，然后是任意参数。参数命令动词和其他参数以任意个空白字符间隔。

2、要在参数里面包含空白，必须用单引号把它引起来。要在这样的参数里包含单引号，可以在前面加一个反斜杠。任何包含在单引号里的内容都会被进一步进行类似 C 语言的替换：\n（新行）、\t（制表符）、\b（退格）、\r（回车）、\f（换页）、\digits（八进制表示的字符）、\xdigits（十六进制表示的字符）。

3、“‘包围的内容被当做一个命令行传入 shell。该命令的输出（删除了结尾的新行）被当做参数值。

4、如果不带引号的参数以冒号（:）开头，它会被当做一个 gsql 变量，并且该变量的值最终会成为真正的参数值。

5、有些命令以一个 SQL 标识的名称（比如一个表）为参数。这些参数遵循 SQL 语法关于双引号的规则：不带双引号的标识强制转换成小写，而双引号保护字母不进行大小写转换，并且允许在标识符中使用空白。在双引号中，成对的双引号在结果名称中分析成一个双引号。比如，FOO“BAR”BAZ 解析成 fooBARbaz；而“A weird”“name”解析成“A weird”name。

6、对参数的分析在遇到另一个不带引号的反斜杠时停止。这里会认为是一个新的元命令的开始。特殊的双反斜杠序列（\\）标识参数的结尾并将继续分析后面的 SQL 语句（如果存在）。这样 SQL 和 gsql 命令可以自由的在一行里面混合。但是在任何情况下，一条元命令的参数不能延续超过行尾。

5.1.4 常见问题处理

连接性能问题

开启了 log_hostname，但是配置了错误的 DNS 导致的连接性能问题。

在连接上数据库，通过“show log_hostname”语句，检查数据库中是否开启了

log_hostname 参数。

如果开启了相关参数，那么数据库内核会通过 DNS 反查客户端所在机器的主机名。这时如果数据库主节点配置了不正确的/不可达的 DNS 服务器，那么会导致数据库建立连接过程较慢。此参数的更多信息，详见《数据库参考》中“GUC 参数说明 > 错误报告和日志 > 记录日志的内容”章节中关于“log_hostname”的描述。

数据库内核执行初始化语句较慢导致的性能问题。

此种情况定位较难，可以尝试使用 Linux 的跟踪命令：strace。

```
strace gsql -U MyUserName -W MyPassWord -d postgres -h 127.0.0.1 -p 23508 -r -c '\q'
```

此时便会在屏幕上打印出数据库的连接过程。比如较长时间停留在下面的操作上：

```
sendto(3, "Q\0\0\0\25SELECT VERSION()\0", 22, MSG_NOSIGNAL, NULL, 0) = 22
poll([{fd=3, events=POLLIN|POLLERR}], 1, -1) = 1 ([{fd=3, revents=POLLIN}])
```

此时便可以确定是数据库执行“SELECT VERSION()”语句较慢。

在连接上数据库后，便可以通过执行“explain performance select version()”语句来确定初始化语句执行较慢的原因。更多信息，详见《性能调优指南》中“SQL 调优指南 > SQL 执行计划介绍”章节。

另外还有一种场景不太常见：由于数据库主节点所在机器的磁盘满或故障，此时所查询等受影响，无法进行用户认证，导致连接过程挂起，表现为假死。解决此问题清理数据库主节点的数据盘空间便可。

TCP 连接创建较慢问题。

此问题可以参考上面的初始化语句较慢排查的做法，通过 strace 跟踪，如果长时间停留在：

```
connect(3,{sa_family=AF_FILE,path="/home/test/tmp/issedb_llt1/.s.Pgsql.61052"}, 110) = 0 或者 connect(3, {sa_family=AF_INET, sin_port=htons(61052), sin_addr=inet_addr("127.0.0.1")}, 16) = -1 EINPROGRESS (Operation now in progress)
```

那么说明客户端与数据库端建立物理连接过慢，此时应当检查网络是否存在不稳定、网络吞吐量太大的问题。

创建连接故障

（1）gsql: could not connect to server: No route to host

此问题一般是指定了不可达的地址或者端口导致的。请检查-h 参数与-p 参数是否添加正确。

（2）gsql: FATAL: Invalid username/password,login denied.

此问题一般是输入了错误的用户名和密码导致的，请联系数据库管理员，确认用户名和密码的正确性。

（3）gsql: FATAL: Forbid remote connection with trust method!

数据库由于安全问题，禁止远程登录时使用 trust 模式。这时需要修改 pg_hba.conf 里的连接认证信息。具体的设置信息请参见：《数据库管理指南》中“管理数据库安全 > 客户端接入认证 > 配置文件参考”章节。

说明：请不要修改 pg_hba.conf 中 ISSEDB 主机的相关设置，否则可能导致数据库功能故障。建议业务应用部署在 ISSEDB 之外，而非 ISSEDB 内部。

（4）连接数据库，添加“-h 127.0.0.1”可以连接，去掉后无法连接问题。

通过执行 SQL 语句“show unix_socket_directory”检查数据库主节点使用的 Unix 套接字目录，是否与 shell 中的环境变量$PGHOST 一致。

如果检查结果不一致，那么修改 PGHOST 环境变量到 GUC 参数 unix_socket_directory 指向的目录便可。

关于 unix_socket_directory 的更多信息，详见《数据库参考》中“GUC 参数说明 > 连接和认证 > 连接设置”章节中的说明。

（5）The “libpq.so” loaded mismatch the version of gsql, please check it.

此问题是由于环境中使用的 libpq.so 的版本与 gsql 的版本不匹配导致的，请通过“ldd gsql”命令确认当前加载的 libpq.so 的版本，并通过修改 LD_LIBRARY_PATH 环境变量来加载正确的 libpq.so。

(6)gsql: symbol lookup error: xxx/gsql: undefined symbol: libpqVersionString

此问题是由于环境中使用的 libpq.so 的版本与 gsql 的版本不匹配导致的（也有可能是环境中存在 PostgreSQL 的 libpq.so），请通过“ldd gsql”命令确认当前加载的 libpq.so 的版本，并通过修改 LD_LIBRARY_PATH 环境变量来加载正确的 libpq.so。

（7）gsql: connect to server failed: Connection timed out

Is the server running on host “xx.xxx.xxx.xxx” and accepting TCP/IP connections on

port xxxx?

此问题是由于网络连接故障造成。请检查客户端与数据库服务器间的网络连接。如果发现从客户端无法 PING 到数据库服务器端，则说明网络连接出现故障。请联系网络管理人员排查解决。

ping -c 4 10.10.10.1

PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.From 10.10.10.1: icmp_seq=2 Destination Host UnreachableFrom 10.10.10.1 icmp_seq=2 Destination Host UnreachableFrom 10.10.10.1 icmp_seq=3 Destination Host UnreachableFrom 10.10.10.1 icmp_seq=4 Destination Host Unreachable--- 10.10.10.1 ping statistics ---4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 2999ms

（8）gsql: FATAL: permission denied for database “postgres”

DETAIL: User does not have CONNECT privilege.

此问题是由于用户不具备访问该数据库的权限，可以使用如下方法解决

1 使用管理员用户 dbadmin 连接数据库。

gsql -d postgres -U dbadmin -p 15400

2 赋予该用户访问数据库的权限。

GRANT CONNECT ON DATABASE postgres TO user1;

5.2 gs_dump

5.2.1 背景信息

gs_dump 是 ISSEDB 用于导出数据库相关信息的工具，用户可以自定义导出一个数据库或其中的对象（模式、表、视图等），回收站对象除外。支持导出的数据库可以是默认数据库 postgres，也可以是自定义数据库。

gs_dump 工具由安装 ISSEDB 数据库的操作系统用户执行。

gs_dump 工具在进行数据导出时，其他用户可以访问 ISSEDB 数据库（读或写）。

gs_dump 工具支持导出完整一致的数据。例如，T1 时刻启动 gs_dump 导出 A 数据库，那么导出数据结果将会是 T1 时刻 A 数据库的数据状态，T1 时刻之后对 A 数据库的修改不会被导出。

gs_dump 工具在进行数据导出时生成的列不会被转储。

gs_dump 支持导出兼容 v1 版本数据库的文本格式文件。

gs_dump 支持将数据库信息导出至纯文本格式的 SQL 脚本文件或其他归档文件中。

gs_dump 可以创建四种不同的导出文件格式，通过“-F”或者“–format=”选项指定，具体如表 1 所示。

表 1 导出文件格式

格式名称	-F 的参数值	说明	建议	对应导入工具
纯文本格式	p	纯文本脚本文件包含 SQL 语句和命令。命令可以由 gsql 命令行终端程序执行，用于重新创建数据库对象并加载表数据。	小型数据库，一般推荐纯文本格式。	使用 gsql 工具恢复数据库对象前，可根据需要使用文本编辑器编辑纯文本导出文件。
自定义归档格式	c	一种二进制文件。支持从导出文件中恢复所有或所选数据库对象。	中型或大型数据库，推荐自定义归档格式。	使用 gs_restore 可以选择要从自定义归档/目录归档/tar 归档导出文件中导入相应的数据库对象。
目录归档格式	d	该格式会创建一个目录，该目录包含两类文件，一类是目录文件，另一类是每个表和 blob 对象对应的数据文件。	-	
tar 归档格式	t	tar 归档文件支持从导出文件中恢复所有或所选数据库对象。tar 归档格式不支持压缩且对于单独表大小应小于 8GB。	-	

5.2.2 语法

gs_dump [OPTION]... [DBNAME]

gs_dump -p port_number postgres -f dump1.sql

或者

export PGDATABASE=postgres

 gs_dump -p port_number -f dump1.sql

环境变量： PGDATABASE

5.3 gs_om

5.3.1 背景信息

ISSEDB 提供了 gs_om 工具帮助对 ISSEDB 进行维护，包括启动 ISSEDB、停止 ISSEDB、查询 ISSEDB 状态、查询静态配置、生成静态配置文件、查询 ISSEDB 状态详细信息、生成动态配置文件、SSL 证书替换、显示帮助信息和显示版本号信息等功能。

5.3.2 前提条件

需以操作系统用户 omm 执行 gs_om 命令。

5.3.3 语法

启动 ISSEDB

gs_om -t start [-h HOSTNAME] [-D dataDir] [--time-out=SECS] [--security-mode=MODE] [--cluster-number=None] [-l LOGFILE]

停止 ISSEDB

gs_om -t stop [-h HOSTNAME] [-D dataDir] [--time-out=SECS] [-m MODE] [-l LOGFILE]

重启 ISSEDB

gs_om -t restart [-h HOSTNAME] [-D dataDir] [--time-out=SECS] [--security-mode=MODE] [-l LOGFILE] [-m MODE]

查询 ISSEDB 状态

gs_om -t status [-h HOSTNAME] [-o OUTPUT] [--detail] [--all] [-l LOGFILE]

生成静态配置文件

gs_om -t generateconf -X XMLFILE [--distribute] [-l LOGFILE]

gs_om -t generateconf --old-values=old --new-values=new [--distribute] [-l LOGFILE]

生成动态配置文件，备机 failover 或 switchover 成主机后，需要执行此操作

gs_om -t refreshconf

查看静态配置

gs_om -t view [-o OUTPUT]

查询 ISSEDB 状态详细信息

gs_om -t query [-o OUTPUT]

SSL 证书替换

gs_om -t cert --cert-file=CERTFILE [-l LOGFILE]

gs_om -t cert --rollback

开启、关闭数据库内 kerberos 认证

gs_om -t kerberos -m [install|uninstall] -U USER [-l LOGFILE] [--krb-client|--krb-server]

显示帮助信息

gs_om -? | --help

显示版本号信息

gs_om -V | --version

参数说明

5.4 gs_install

5.4.1 背景信息

数据库的部署是一个复杂的过程。ISSEDB 提供了 gs_install 工具来帮助完成 ISSEDB 的安装和部署。

ISSEDB 安装部署，要求用户指定配置文件，配置文件中会指定程序安装路径、实例数据目录、主备关系、实例数、各实例的业务 IP 端口等信息。

前提条件

已成功执行前置脚本 gs_preinstall。

用户需确保各个节点上的 locale 保持一致。

需要使用前置时设置的 ISSEDB 用户进行安装操作。

语法

5.4.2 安装 ISSEDB

gs_install -X XMLFILE [--gsinit-parameter="PARAMETER" [...]] [--dn-guc="PARAMETER" [...]] [--alarm-component=ALARMCOMPONENT] [--time-out=SECS] [-l LOGFILE]

说明： 安装时若不指定字符集，默认字符集为 SQL_ASCII，为简化和统一区域 loacle 默认设置为 C，若想指定其他字符集和区域，请在安装时使用参数 –gsinit-parameter="–locale=LOCALE"来指定，LOCALE 为新数据库设置缺省的区域。

显示帮助信息

gs_install -? | --help

显示版本号信息

gs_install -V | --version

5.5 gs_preinstall

5.5.1 背景信息

ISSEDB 提供了 gs_preinstall 工具来帮助完成 ISSEDB 的环境配置，以保证 ISSEDB 安装的正常进行。

5.5.2 语法

准备 ISSEDB 环境

gs_preinstall -U USER -G GROUP -X XMLFILE [-L] [--skip-os-set] [--env-var="ENVVAR" [...]] [--sep-env-file=MPPRCFILE] [--skip-hostname-set] [-l LOGFILE] [--non-interactive]

显示帮助信息

gs_preinstall -? | --help

显示版本号信息

gs_preinstall -V | --version

表 1 ISSEDB 默认创建的环境变量

环境变量名称	说明
MPPDB_ENV_SEPARATE_PATH	ISSEDB 环境变量分离文件路径
GPHOME	ISSEDB 工具目录
PATH	ISSEDB 工具脚本目录
LD_LIBRARY_PATH	ISSEDB 引用第三方动态库路径
PYTHONPATH	python 软件路径
GAUSS_WARNING_TYPE	告警类型
GAUSSHOME	ISSEDB 安装路径
GAUSS_VERSION	ISSEDB 版本号
PGHOST	ISSEDB 用户的临时目录路径
GS_CLUSTER_NAME	ISSEDB 名称
GAUSSLOG	ISSEDB 日志路径
GAUSS_ENV	ISSEDB 环境变量标识

设置“/etc/syslog-ng/syslog-ng.conf”文件，在文件中添加如下内容：

```
template t_issedb {template("$DATE $SOURCEIP $MSGONLY\n");template_escape(no);};
source s_issedb{ udp(); };
filter f_issedb  { level(err,  crit) and match('issedb'); };
destination d_issedb { file("/var/log/syslog_MPPDB", template(t_issedb)); };
log { source(s_issedb); filter(f_issedb); destination(d_issedb); };
```

设置“/etc/sysconfig/syslog”文件，在文件中添加如下内容：

```
SYSLOGD_OPTIONS="-r -m 0"
KLOGD_OPTIONS="-x"
```

说明： 该配置需要在 ISSEDB 每台机器上都要修改。