

GBase 8s 数据库服务器安装与优化配置指南

南大通用数据技术股份有限公司

General Data Technologies Co., Ltd.

版权所有© GBASE 2024

GBASE[®]

天津总公司：天津市高新区华苑产业园区工华道 2 号天百中心 3 层

电 话：022-58815678

传 真：022-58815679

北京分公司：北京市朝阳区安定路五号院 13 号北投投资大厦 A 座 606-608

电 话：010-88866866

传 真：010-88864556

<http://www.gbase.cn>

E-mail: info@gbase.cn

GBase 8s 数据库服务器安装与优化配置指南，南大通用数据技术股份有限公司
版权所有© GBASE 2024，保留所有权利。

版权声明

本档所涉及的软件著作权、版权和知识产权已依法进行了相关注册、登记，由南大通用数据技术股份有限公司合法拥有，受《中华人民共和国著作权法》、《计算机软件保护条例》、《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

免责声明

本档包含的南大通用公司的版权信息由南大通用公司合法拥有，受法律的保护，南大通用公司对本文档可能涉及到的非南大通用公司的信息不承担任何责任。在法律允许的范围内，您可以查阅，并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经南大通用公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将视为侵权，南大通用公司具有依法追究其责任的权利。

本档中包含的信息如有更新，恕不另行通知。您对本文档的任何问题，可直接向南大通用数据技术股份有限公司告知或查询。

未经本公司明确授予的任何权利均予保留。

通讯方式

南大通用数据技术股份有限公司

中国天津市高新区华苑产业园区工华道 2 号天百中心 3 层 (300384)

电话：400-013-9696

邮箱：info@gbase.cn

商标声明

GBase® 是南大通用数据技术股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由南大通用公司合法拥有，受法律保护。未经南大通用公司书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯南大通用公司商标权的，南大通用公司将依法追究其法律责任。

前言

概述

本文档描述了 GBase 8s 数据库通用的安装部署操作方法，以及对 GBase 8s 数据库通用使用方式的优化配置。

读者对象

本文档主要适用于以下技术人员：

- 数据库管理员
- 数据库运维人员
- 数据库初学者

初学者注意事项

需要具备一定的 Linux 操作经验，了解常用的命令行含义。

目 录

前言	III
概述	III
读者对象	III
初学者注意事项	III
1. GBASE 8S 数据库介绍	1
2. 数据库安装配置与准备	2
2.1. 兼容性	2
2.2. 操作系统内核参数	2
2.3. 底层存储配置	3
2.4. 数据库安装介质准备	3
3. 数据库安装部署	5
3.1. 安装环境检查	5
3.2. 操作系统配置	6
3.2.1. 设置主机名称及网络	6
3.2.2. 关闭域名反向解析功能	6
3.2.3. 关闭 selinux	6
3.2.4. 关闭或者配置防火墙	6
3.3. 自动化安装脚本安装	7
3.3.1. 执行自动化安装脚本安装	7
3.4. 数据库典型安装	12
3.4.1. 创建用户组及用户	12
3.4.2. 执行典型安装	13
3.5. 数据库自定义安装	19
3.5.1. 创建用户组及用户	20
3.5.2. 磁盘规划	20
3.5.2.1. 使用本地文件系统	20
3.5.2.2. 使用逻辑卷组管理	21
3.5.2.3. 使用 LUN 磁盘裸设备管理	24
3.5.3. 安装数据库软件	27
3.5.4. 配置、初始化数据库实例	27
3.5.5. 创建数据库内部用户	37
3.6. 数据库参数优化与调整	38
3.6.1. 配置参数优化	38
3.6.2. 关闭 sysadmin 库运行	39
3.6.3. 数据库空间自动扩展 (chunk 自动扩展)	39
3.7. 数据库加入自启动服务	40
4. 附录: GBASE 8S 数据库内存管理与配置	42
4.1. 数据库内存配置	43
4.1.1. 常驻段内存使用量计算	43
4.1.2. 缓冲池内存使用量计算	43
4.1.3. 虚拟段内存使用量计算	44
4.1.4. 消息段内存使用时计算	44

4.1.5.	数据库内存段计算	44
4.2.	数据库安装常见问题	45
4.2.1.	安装问题 1: The version file is not exists	45
4.2.2.	安装问题 2: 环境变量 PATH 中无 JVM.....	46
4.2.3.	安装问题 3: 安装目录不存在	46
4.2.4.	安装问题 4: 安装目录非空	46
4.2.5.	初始化问题 5: 无法分配内存	46
4.2.6.	初始化问题 6: 初始化内存失败, 无法打开主 chunk.....	47
4.2.7.	连接问题 7: 无法连接到数据库服务器 (908)	48
4.2.8.	连接问题 8: 连接参数	48
4.2.9.	连接问题 9: 工具连接报错: 内部错误	49
4.2.10.	安装问题 10: 无法执行二进制文件	50
4.2.11.	安装问题 11: 无法打开共享对象文件 libnsl.so.1.....	51
4.2.12.	安装问题 12: 一个或多个系统依赖包未安装	51

1. GBase 8s 数据库介绍

GBase 8s 是天津南大通用数据技术股份有限公司自主研发的、成熟稳定的企业级分布式事务型数据库，拥有自主知识产权。产品达到安全数据库四级标准（国际 B2），支持国密算法，支持 SQL92/99、ODBC、JDBC、ADO.NET、GCI(OCI/OCCI)、Python 接口等国际数据库规范和开发接口。支持分布式部署、集中式部署、共享存储高可用部署、支持两地三中心高可用部署。GBase 8s 分布式架构具备高容量、高并发、高性能等特性。为超大规模数据管理提供高性价比的通用计算平台。

GBase 8s 适用于 OLTP 应用场景，包括金融、电信行业的关键核心业务系统，安全、党政、国防等行业对信息安全性有较高要求的信息系统，以及大型企业的经营类、管理类信息系统，能够提供 7*24 小时不间断运行处理能力，在 80% 以上场景中可以替代国际主流数据库。

GBase 8s 原生支持通用平台，如 AMD64、ARM64、IBM POWER7/8、PR_RISC、IA64 等硬件平台；原生支持通用操作系统，如 Linux（包含常见分支：CentOS/RHEL/Ubuntu）、AIX（包含：IBM AIX 5/6/7、HP-UX、SunOS）等。

GBase 8s 已完成对全部国产平台的深度适配工作，支持鲲鹏、飞腾、龙芯、申威、兆芯、海光等系列的 CPU。已完成全部主流国产操作系统的适配，包括统信 UOS V20、麒麟（中标麒麟 V7、银河麒麟及麒麟 V10）、中标普华、凝思磐石、万里红、中科方德、浪潮 K-UX 等。

2. 数据库安装配置与准备

数据库安装部署前需要对先决条件进行准备配置，包括操作系统的安装、参数配置、网络配置、底层存储配置和数据库安装介质的准备。

2.1. 兼容性

GBase 8s 数据库需要具备良好的平台兼容能力，以满足不同环境的实际要求，下表为操作系统和服务器的兼容性建议：

➤ 操作系统

UOS 统信操作系统、麒麟、中科方德、普华、凝思、红旗、一铭、深之度

64 位 Unix 系列(AIX、HP-UX、Solaris)

64 位 Linux 系列 (CentOS 、 RedHat 、 SUSE、Asianux、Debian、Ubuntu)

➤ CPU 芯片

鲲鹏 920、龙芯 3A\3B、飞腾系列、申威系列、IBM POWER、HP-UX Itanium、Linux Intel、Solaris SPARC 以及国产浪潮等平台。

➤ 整机

支持主流国产整机：浪潮天梭 K1 小型机、曙光龙腾服务器、华为 KunLun、紫光华山 Superdome X。

2.2. 操作系统内核参数

以 RHEL7 或者 CentOS7 为例

确认如下内核参数的值，需要更改/etc/sysctl.conf 并使之生效

```
# shm 不小于以下值即可
kernel.shmmax = 18446744073692774399
kernel.shmall = 18446744073692774399
kernel.shmmni = 4096
kernel.sem = 32000 1024000000 32000 32000
vm.swappiness = 0
```

用户资源限制，/etc/security/limits.conf 或者/etc/security/limits.d/20-nproc.conf 中增加修改如下内容

*	soft	nproc	1048576
root	soft	nproc	unlimited

```
*          soft    nofile    1048576
root       soft    nofile    unlimited
```

如果用户 `nofile` 和 `nproc` 设置值不生效，需要修改 `/etc/systemd/system.conf` 配置文件。注：该步骤需要重启才生效。

```
# 去掉#号，并设置相应的值
DefaultLimitNOFILE=1048576
DefaultLimitNPROC=1048576
```

以上均为生产建议值，按实际生产环境调整。

修改 `/etc/systemd/logind.conf` 配置文件，关闭 `RemoveIPC` 功能

```
# 去掉#号
RemoveIPC=no
```

重启服务或重启操作系统

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

结果验证确认

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

2.3. 底层存储配置

底层存储推荐配置性能更优的固态硬盘，RAID 采用 RAID5 或者 RAID10 的模式。

磁盘 LUN 分为逻辑日志 LUN、数据归档 LUN、数据 LUN 三种磁盘类型。

具体空间大小可以根据实际要求来规划设计。

2.4. 数据库安装介质准备

根据安装环境准备申请下载数据库软件介质。

按以下内容选择合适的软件包

主版本号：8.7/8.8/8.9

子版本号：详细的发行版本 2.2.0、3.0.0_1、3.3.0_2、3.5.1_3

编译环境：编译该软件包的环境。安装环境应与此相同或者相近。

YHKylin4_FT 表示是飞腾平台（FT ARM64）、银河麒麟系统

RHEL6_x86_64 表示是 x86 架构（64bit）、RHEL6 系统

AIX5L 表示 IBM Power 架构、AIX5 系统

如：`GBase8sV8.8_3.0.0_1_93e040_RHEL6_x86_64.tar` 表示适用于 x86_64 架构，操作系统为 RHEL6（或者兼容）的软件安装包。

3. 数据库安装部署

数据库安装部署前需要满足兼容性的要求，安装部署一般要求提供多种的安装部署方式：图形化界面、命令行界面。每种安装模式需要支持向导式的安装界面，用户可以简单方便的根据向导完成安装部署。

注意事项：

数据库安装前需要规划安装用户和安全性要求，尽量避免弱口令化的模式。

某些国产操作系统中，限制了/data 目录的执行权限。因此执行数据库软件安装和数据库安装目录不能放置于/data 目录。

3.1. 安装环境检查

数据库软件安装与运行有一定的操作系统环境需求，特别是在某些国产化操作系统中。通常情况下，GBase 8s 数据库需要以下程序或者库文件支持。

- 需要的程序，一般位于/usr/bin 目录下，不同的操作系统，所在的目录不同

unzip

tar

java（可选，若使用系统 jre 环境需要）

- 需要的库文件，一般位于/usr/lib64 目录下，不同的操作系统，所在的目录不同，一般可通过的 find / -name 的方式查找。

libc.so.6

libm.so.6

libncurses.so.5

libtinfo.so.5

libnsl.so.1（可选，若使用系统 jre 环境需要）

libaio.so.1（可选，使用使用 KAIO，需要）

注：有些操作系统中，必须使用操作系统的 jre 环境，这时 libnsl.so.1 就是必须的。

如果缺少相应的，在有 apt 源或者 yum 源时，可直接安装。如，使用 yum 安装：

```
yum -y install unzip tar java-1.8.0-openjdk glibc-devel ncurses-libs libnsl libaio
```

3.2. 操作系统配置

3.2.1. 设置主机名称及网络

配置/etc/hosts，确认 hostname 输出的主机名与 IP 地址对应匹配

例如：主机名是 node01，对应的物理网卡使用的 IP 是 192.168.80.101

```
# /etc/hosts
192.168.80.101 node01
```

非常重要：如果您准备使用 典型安装-自动创建实例 的方式，必须完成此操作！

3.2.2. 关闭域名反向解析功能

在没有使用 DNS 的环境，可关闭域名反向解析功能，以加速网络连接

修改/etc/nsswitch.conf 配置文件，hosts 参数仅保留 files，注释掉其它值

```
# /etc/nsswitch.conf
hosts:      files # dns myhostname
```

3.2.3. 关闭 selinux

临时关闭 selinux，省去对数据库软件安装的影响

修改/etc/selinux/config 配置文件，修改 SELINUX 的值为 disabled，重启系统

```
# /etc/selinux/config
SELINUX=disabled
```

3.2.4. 关闭或者配置防火墙

操作系统自带的防火墙可能限制了数据库端口的访问，因此，需要关闭防火墙或者设置防火墙策略，开放相应的端口。

不同的操作系统，甚至相同操作系统的不同版本，使用的防火墙都不同，因此，关闭防火墙的方式各不相同，同样的，开放端口的方式也不相同。

例如，RHEL7 或 CentOS7 中关闭防火墙的操作命令为

```
# RHEL7 或者 CentOS7 或者使用相同内核的操作系统
systemctl stop firewalld
systemctl disable firewalld
```

开放端口的操作命令为

```
# RHEL7 或者 CentOS7 或者使用相同内核的操作系统
firewall-cmd --permanent --zone=public --add-port=9088/tcp
firewall-cmd --reload
```

3.3. 自动化安装脚本安装

【注意】3.3 自动化安装脚本安装（适用于快速熟悉与开发环境）、3.4 数据库典型安装（适用于快速开发）和 3.5 数据库自定义安装（适用于生产环境）三种方式中，选择一种方式安装即可。

使用自动化安装脚本安装，不需要创建用户组及用户，脚本将自动创建用户级及用户，自动安装软件，自动根据系统硬件资源优化数据库配置。默认使用的实例名称是 gbase01，端口使用 9088。

自动化安装脚本安装方式，只能使用本地文件系统，适用于普通开发者使用，或者数据库初学者练习使用，对于环境的要求不高。

3.3.1. 执行自动化安装脚本安装

以下安装示例，使用的硬件平台是 x86_64，操作系统的版本是 UOS20（1031），数据库版本是 GBase8sV8.8_3.0.0_1_93e040_RHEL6_x86_64.tar。

自动化安装脚本安装的特点：

- 1)、全自动化安装，默认无需干预；
- 2)、默认仅 6 个 1G 大小的字间；
- 3)、默认的字符集（zh_CN.utf8）；
- 4)、创建 testdb 测试库；

将数据库软件安装包和自动化安装脚本放置于相同目录下，执行 AutoInit_GBase8s.sh 开始安装。

```
root@netsky-PC:~# wget \
https://gbasedbt.com/dl/AutoInit_GBase8s/1.4.5/AutoInit_GBase8s_v1.4.5.tar

root@netsky-PC:~# ls -al *.tar
-rw-r--r-- 1 root root 30720 1月 19 13:23 AutoInit_GBase8s_v1.4.5.tar
-rw-r--r-- 1 root root 316221440 12月 23 2020 GBase8sV8.8_3.0.0_1_93e040_RHEL6_x86_64.tar
```

```
root@netsky-PC:~# tar -xvf AutoInit_GBase8s_v1.4.5.tar
AutoInit_GBase8s.sh
CheckEnv.sh
CleanAll.sh
HOW_TO_USE.txt
README.txt

root@netsky-PC:~# bash AutoInit_GBase8s.sh (执行自动安装)
[2022-01-19 13:28:06] The SYSDBA user is: gbasedbt
[2022-01-19 13:28:06] IPADDR: 0.0.0.0
[2022-01-19 13:28:06] Datadir: /data/gbase
[2022-01-19 13:28:06] Creating group [gbasedbt] and user [gbasedbt] with HOME
[/home/gbase].
[2022-01-19 13:28:06] Unzipping [GBase8sV8.8.3.0.0_1_93e040_RHEL6_x86_64.tar].
[2022-01-19 13:28:07] Check path INSTALL_DIR(/opt/gbase) security.
[2022-01-19 13:28:07] Execute software install, this will take a moment.
[2022-01-19 13:29:10] Building ~gbasedbt/.bash_profile .
[2022-01-19 13:29:10] Building /opt/gbase/etc/sqlhosts .
[2022-01-19 13:29:10] Building /opt/gbase/etc/onconfig.gbase01 .
[2022-01-19 13:29:10] Creating DATADIR: /data/gbase .
[2022-01-19 13:29:10] Change permission for directory: /data/gbase .
[2022-01-19 13:29:10] Change permission for directory: /data .
[2022-01-19 13:29:11] Start run database init: oninit -ivy
Reading configuration file '/opt/gbase/etc/onconfig.gbase01'...succeeded
Creating /GBASEDBTMP/.infxdirs...succeeded
Allocating and attaching to shared memory...succeeded
Creating resident pool 4310 kbytes...succeeded
Creating infos file "/opt/gbase/etc/infos.gbase01"...succeeded
Linking conf file "/opt/gbase/etc/.conf.gbase01"...succeeded
Initializing rhead structure...rhlock_t 16384 (512K)... rlock_t (2656K)... Writing
to infos file...succeeded
Initialization of Encryption...succeeded
Initializing ASF...succeeded
Initializing Dictionary Cache and SPL Routine Cache...succeeded
Bringing up ADM VP...succeeded
Creating VP classes...succeeded
Forking main_loop thread...succeeded
Initializing DR structures...succeeded
Forking 1 'soctcp' listener threads...succeeded
Starting tracing...succeeded
Initializing 8 flushers...succeeded
Initializing log/checkpoint information...succeeded
Initializing dbspaces...succeeded
```

```
Opening primary chunks...succeeded
Validating chunks...succeeded
Creating database partition...succeeded
Initialize Async Log Flusher...succeeded
Starting B-tree Scanner...succeeded
Init ReadAhead Daemon...succeeded
Init DB Util Daemon...succeeded
Initializing DBSPACETEMP list...succeeded
Init Auto Tuning Daemon...succeeded
Checking database partition index...succeeded
Initializing dataskip structure...succeeded
Checking for temporary tables to drop...succeeded
Updating Global Row Counter...succeeded
Forking onmode_mon thread...succeeded
Creating periodic thread...succeeded
Creating periodic thread...succeeded
Starting scheduling system...succeeded
Verbose output complete: mode = 5
OK
[2022-01-19 13:29:19] Creating system database.....
[2022-01-19 13:29:37] Creating dbspace plogdbs.
[2022-01-19 13:29:41] Creating dbspace llogdbs.
[2022-01-19 13:29:42] Creating dbspace tempdbs01
[2022-01-19 13:29:43] Creating smart blob space sbpace01
[2022-01-19 13:29:46] Creating dbspace datadbs01
[2022-01-19 13:29:49] Changing auto extend able on for chunk datadbs01
[2022-01-19 13:29:49] Creating default user for mapping user
[2022-01-19 13:29:50] Moving physical log to plogdbs.
[2022-01-19 13:29:57] Adding 10 logical log file in llogdbs.
[2022-01-19 13:30:00] Moving CURRENT logical log to new logical file.
[2022-01-19 13:30:13] Dropping logical log file which in rootdbs.
[2022-01-19 13:30:13] Create database testdb.
[2022-01-19 13:30:13] Creating file $INSTALL_DIR/etc/sysadmin/stop .
[2022-01-19 13:30:14] Optimizing database config.
[2022-01-19 13:30:14] Restart GBase 8s Database Server.
Reading configuration file '/opt/gbase/etc/onconfig.gbase01'...succeeded
Creating /GBASEDBTMP/.infxdirs...succeeded
Allocating and attaching to shared memory...succeeded
Creating resident pool 34422 kbytes...succeeded
Creating infos file "/opt/gbase/etc/.infos.gbase01"...succeeded
Linking conf file "/opt/gbase/etc/.conf.gbase01"...succeeded
Initializing rhead structure...rhlock_t 65536 (2048K)... rlock_t (26562K)... Writing
to infos file...succeeded
Initialization of Encryption...succeeded
```

```
Initializing ASF...succeeded
Initializing Dictionary Cache and SPL Routine Cache...succeeded
Bringing up ADM VP...succeeded
Creating VP classes...succeeded
Forking main_loop thread...succeeded
Initializing DR structures...succeeded
Forking 1 'soctcp' listener threads...succeeded
Starting tracing...succeeded
Initializing 32 flushers...succeeded
Initializing SDS Server network connections...succeeded
Initializing log/checkpoint information...succeeded
Initializing dbspaces...succeeded
Opening primary chunks...succeeded
Validating chunks...succeeded
Initialize Async Log Flusher...succeeded
Starting B-tree Scanner...succeeded
Init ReadAhead Daemon...succeeded
Init DB Util Daemon...succeeded
Initializing DBSPACETEMP list...succeeded
Init Auto Tuning Daemon...succeeded
Checking database partition index...succeeded
Initializing dataskip structure...succeeded
Checking for temporary tables to drop...succeeded
Updating Global Row Counter...succeeded
Forking onmode_mon thread...succeeded
Creating periodic thread...succeeded
Creating periodic thread...succeeded
Starting scheduling system...succeeded
Verbose output complete: mode = 5
[2022-01-19 13:30:32] Finish.
```

(以下安装完成信息)

```
--== GBase 8s Information for this install ==--
$GBASEDBTSERVER : gbase01
$GBASEDBTDIR    : /opt/gbase
USER HOME       : /home/gbase
DBSPACE DIR     : /data/gbase
IP ADDRESS      : 0.0.0.0
PORT NUMBER     : 9088
$DB_LOCALE     : zh_CN.utf8
$CLIENT_LOCALE : zh_CN.utf8
JDBC            URL                                     :
jdbc:gbasedbt-sqli://IPADDR:9088/testdb:GBASEDBTSERVER=gbase01;DB_LOCALE=zh_CN.u
tf8;CLIENT_LOCALE=zh_CN.utf8;IFX_LOCK_MODE_WAIT=10
```

```
JDBC USERNAME   : gbasedbt
JDBC PASSWORD   : GBase123
```

检查安装，切换到 gbasedbt 用户，自动加载环境变量

```
root@netsky-PC:~# su - gbasedbt

gbasedbt@netsky-PC:~$ more .bash_profile
export GBASEDBTDIR=/opt/gbase
export GBASEDBTSERVER=gbase01
export ONCONFIG=onconfig.$GBASEDBTSERVER
export PATH=$GBASEDBTDIR/bin:${PATH}
export GBASEDBTSQLHOSTS=/opt/gbase/etc/sqlhosts

export DB_LOCALE=zh_CN.utf8
export CLIENT_LOCALE=zh_CN.utf8
export GL_USEGLU=1
export DBDATE="Y4MD-"
export DBACCESS_SHOW_TIME=1
```

检查网络侦听

```
gbasedbt@netsky-PC:~$ onstat -g ntt

GBase Database Server Version 12.10.FC4G1AEE -- On-Line -- Up 00:00:59 -- 768264 Kbytes

global network information:
#netscb connects      read      write    q-free  q-limits  q-exceed alloc/max
  2/   2         0         0       0/   0  240/  10    0/   0    0/  -1

Individual thread network information (times):
      netscb thread name      sid    open    read    write address
      47eb9be0 soctcplst      3 13:30:27      0.0.0.0|9088|soctcp
      47eb7be0 soctcpoll      2 13:30:27
```

检查数据库空间

```
gbasedbt@netsky-PC:~$ onstat -d

GBase Database Server Version 12.10.FC4G1AEE -- On-Line -- Up 00:00:56 -- 768264 Kbytes

Dbspaces
address      number  flags    fchunk  nchunks  pgsize  flags  owner  name
47331028     1      0x70001  1       1        2048   N BA   gbasedbt rootdbs
4993ddc8     2      0x70001  2       1        2048   N BA   gbasedbt plogdbs
49f47028     3      0x60001  3       1        2048   N BA   gbasedbt llogdbs
49f47258     4      0x42001  4       1        2048   N TBA  gbasedbt tempdbs01
```



```

49f47488      5      0x68001      5      1      2048      N SBA      gbasedbt sbspace01
49f476b8      6      0x60001      6      1      2048      N BA      gbasedbt datadbs01
  6 active, 2047 maximum

Chunks
address  chunk/dbs  offset      size      free      bpages      flags pathname
47331258  1      1      0      512000  501679      PO-B-D /data/gbase/rootchk
49f48028  2      2      0      512000  11947      PO-B-D /data/gbase/plogchk
49f49028  3      3      0      512000  11947      PO-B-D /data/gbase/llogchk
49f4a028  4      4      0      512000  511947      PO-B-- /data/gbase/tempchk01
49f4b028  5      5      0      512000  477465  477465      POSB-D /data/gbase/sbspace01
          Metadata 34482  25659  34482
49f4c028  6      6      0      512000  510041      PO-BED /data/gbase/datachk01
  6 active, 32766 maximum

NOTE: The values in the "size" and "free" columns for DBspace chunks are
      displayed in terms of "pgsize" of the DBspace to which they belong.

Expanded chunk capacity mode: always
    
```

3.4. 数据库典型安装

【注意】3.3 自动化安装脚本安装（适用于快速熟悉与开发环境）、3.4 数据库典型安装（适用于快速开发）和 3.5 数据库自定义安装（适用于生产环境）三种方式中，选择一种方式安装即可。

数据库典型安装，需要手工创建用户组及用户，安装过程将自动创建实例并初始化，首次安装使用 ol_gbasedbt1210 的实例名称。

典型安装方式，只能使用本地文件系统，适用于普通开发者使用，或者数据库初学者练习使用，对于环境的要求不高。

3.4.1. 创建用户组及用户

使用 root 用户创建 gbasedbt 组

```
groupadd -g 1000 gbasedbt
```

创建 gbasedbt 用户

```
useradd -g gbasedbt -u 1000 -d /home/gbase -m -s /bin/bash gbasedbt
```

设置 gbasedbt 用户密码

```
passwd gbasedbt
```

注意：gbasedbt 用户是数据库系统管理员用户，使用操作系统验证密码访问，密码强度应当符合操作系统安全要求。

注意 2：某些操作系统中，首次设置的密码无效，需要首次通过 ssh 远程登陆并修改密码才能生效。

3.4.2. 执行典型安装

以下安装示例，使用的硬件平台是 x86_64，操作系统的版本是 UOS20（1031），数据库版本是 GBase8sV8.8_3.0.0_1_93e040_RHEL6_x86_64.tar。

典型安装-自动创建实例的特点：

- 1)、根据指定的用户数初始化空间大小；
- 2)、使用默认的字符集（不指定字符集，使用 en_US.819）；
- 3)、仅初始化系统库；

检查主机名

```
root@netsky-PC:~# grep -i $(hostname) /etc/hosts
127.0.1.1 netsky-PC
```

检查依赖，确认必须的程序和包都已经存在

```
root@netsky-PC:~# find /usr/bin -name 'unzip' -o -name 'tar' -o -name 'java'
/usr/bin/tar
/usr/bin/unzip

root@netsky-PC:~# find /usr/lib -name 'libc.so.6' -o -name 'libm.so.6' -o \
-name 'libncurses.so.5' -o -name 'libtinfo.so.5' -o -name 'libnsl.so.1' \
-o -name 'libaio.so.1'
/usr/lib/x86_64-linux-gnu/libc.so.6
/usr/lib/x86_64-linux-gnu/libm.so.6
/usr/lib/x86_64-linux-gnu/libncurses.so.5
/usr/lib/x86_64-linux-gnu/libnsl.so.1
/usr/lib/x86_64-linux-gnu/libtinfo.so.5
/usr/lib/x86_64-linux-gnu/libaio.so.1
```

创建用户组和用户，并设置密码

```
root@netsky-PC:~# groupadd -g 2000 gbasedbt

root@netsky-PC:~# useradd -g 2000 -u 2000 -d /home/gbase -m -s /bin/bash gbasedbt
```

```
root@netsky-PC:~# passwd gbasedbt
新的 密码: GBase123 (设置密码)
无效的密码: 太简单或太有规律
重新输入新的 密码: GBase123 (确认密码)
passwd: 已成功更新密码
```

解压缩软件包，执行 `ids_install` 开始安装

```
root@netsky-PC:~/server# ls -al ../GBase8sV8.8_3.0.0_1_93e040_RHEL6_x86_64.tar
-rw-r--r-- 1 root root 316221440 12月 23 2020 ../GBase8sV8.8_3.0.0_1_93e040_RHEL6_x86_64.tar

root@netsky-PC:~/server# tar -xvf ../GBase8sV8.8_3.0.0_1_93e040_RHEL6_x86_64.tar
doc/
doc/ids_machine_notes_12.10.txt
doc/ids_unix_relnotes_12.10.html
ids_install
ids.properties
onsecurity
.gbase.properties

root@netsky-PC:~/server# ./ids_install (开始安装)
Preparing to install...
Extracting the JRE from the installer archive...
Unpacking the JRE...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...

...(省略)...

1. Release Notes
The Release Notes can be found in
/root/server/doc/ids_unix_relnotes_12.10.html.
2. Launch Information Center
Access the GBase Information Center at http://www.gbase.cn.
To Begin Installation, respond to each prompt to proceed to the next step in
the installation.
If you want to change something on a previous step, type 'back'.
You may cancel this installation at any time by typing 'quit'.

PRESS <ENTER> TO CONTINUE: 【回车】

...(省略)...

PRESS <ENTER> TO CONTINUE: 【回车】 (查看协议)
```

...(省略)...

DO YOU ACCEPT THE TERMS OF THIS LICENSE AGREEMENT? (Y/N): **Y** (接受协议)

=====
Installation Location

Choose location for software installation.

Default Install Folder: /opt/GBASE/gbase

ENTER AN ABSOLUTE PATH, OR PRESS <ENTER> TO ACCEPT THE DEFAULT

: **/opt/gbase** (指定目录)

(注意: 指定的目录的上一级目录需要存在, 并所有用户可以访问)

INSTALL FOLDER IS: /opt/gbase

IS THIS CORRECT? (Y/N): **Y** (确认目录)

...(省略)...

Custom: Install the database server with specific features and software that you need.

Optionally install a configured database server instance.

Minimum disk space required: 75 MB (without a server instance)

->1- **Typical installation**

2- Custom installation

3- Extract the product files (-DLEGACY option)

ENTER THE NUMBER FOR YOUR CHOICE, OR PRESS <ENTER> TO ACCEPT THE DEFAULT:: **【回车】**

(默认: 典型安装)

...(省略)...

Create a server instance?

->1- **Yes - create an instance**

2- No - do not create an instance

ENTER THE NUMBER FOR YOUR CHOICE, OR PRESS <ENTER> TO ACCEPT THE DEFAULT:: **【回车】**

(默认: 创建实例)

...(省略)...

Select the number of expected database users.

->1- 1 - 100

2- 101 - 500

3- 501 - 1000

4- 1000+

ENTER THE NUMBER FOR YOUR CHOICE, OR PRESS <ENTER> TO ACCEPT THE DEFAULT:: **【回车】**
(按实际选择用户数)

...(省略)...

InstallAnywhere is now ready to install GBase Software Bundle onto your system at the following location:

/opt/gbase

PRESS <ENTER> TO INSTALL: **【回车】** (开始安装)

...(省略)...

Server Initialization

The server will now be initialized. Initialization might take quite a while, depending on the performance of your computer.

PRESS <ENTER> TO CONTINUE: **【回车】** (初始化实例)

Using the new instance

A database server instance was created. If you chose to initialize the instance, it is ready to use.

You can open a command prompt to an initialized instance by running one of the following commands at /opt/gbase, where ol_gbasedbt1210 is the path or filename of the instance.

```
Windows:
ol_gbasedbt1210.cmd

UNIX csh:
source ol_gbasedbt1210.csh

UNIX ksh or bourne:
./ol_gbasedbt1210.ksh

If initialization fails, check the online.log file for messages. The instance's
root chunk must be owned by the correct user and group, and it must have
readable and writable (rw) permission bits set only for owner and group.

PRESS <ENTER> TO CONTINUE: 【回车】 (若失败时的注意事项)

-----
Installation Complete
-----

Congratulations! GBase Software Bundle installation is complete.

Product install status:
GBase: Successful
GBase Connect: Successful

GBase 8s V8.8

For more information about using GBase products, see the GBase Information
Center at http://www.gbase.cn.

PRESS <ENTER> TO EXIT THE INSTALLER: 【回车】 (完成安装)
```

检查安装，切换到 gbasedbt 用户，加载环境变量

```
root@netsky-PC:~/server# su - gbasedbt

gbasedbt@netsky-PC:~$ cd /opt/gbase/

gbasedbt@netsky-PC:/opt/gbase$ more ol_gbasedbt1210.ksh
GBASEBTDIR=/opt/gbase
GBASEDBTSERVER=ol_gbasedbt1210
ONCONFIG=onconfig.ol_gbasedbt1210
GBASEDBTSQLHOSTS=/opt/gbase/etc/sqlhosts.ol_gbasedbt1210
```

```

GL_USEGLU=1
PATH=${GBASEBTDIR}/bin:${GBASEBTDIR}/extend/krakatoa/jre/bin:${PATH}
export GBASEBTDIR GBASEBTSERVER ONCONFIG GBASEBTSQLHOSTS GL_USEGLU PATH

gbasedbt@netsky-PC:/opt/gbase$ . ol_gbasedbt1210.ksh

```

检查网络侦听

```

gbasedbt@netsky-PC:/opt/gbase$ onstat -g ntt

GBase Database Server Version 12.10.FC4G1AEE -- On-Line -- Up 00:01:40 -- 208516 Kbytes

global network information:
#netscb connects      read      write    q-free  q-limits  q-exceed alloc/max
4/ 5      15      4321      4320    2/ 2    205/ 10    0/ 0    2/ 2

Individual thread network information (times):
netscb thread name    sid      open      read      write address
47a444c8 soctcplst      5 11:54:07      127.0.0.1|lo_gbasedbt1210|soctcp
47892ca0 soctcplst      4 11:54:07      netsky-PC|dr_gbasedbt1210|soctcp
476e3ad8 soctcplst      3 11:54:07 11:54:38      netsky-PC|ol_gbasedbt1210|soctcp
45f39c68 soctcpoll      2 11:54:07

```

检查相应的端口

```

gbasedbt@netsky-PC:/opt/gbase$ grep 'ol_gbasedbt1210' /etc/services
ol_gbasedbt1210      9088/tcp
ol_gbasedbt1210_json 27017/tcp      #JSON listener for ol_gbasedbt1210

```

检查数据库空间

```

gbasedbt@netsky-PC:/opt/gbase$ onstat -d

GBase Database Server Version 12.10.FC4G1AEE -- On-Line -- Up 00:01:28 -- 208516 Kbytes

Dbspaces
address      number  flags    fchunk  nchunks  pgsize  flags  owner  name
454b0028    1      0x70001  1       1        2048    N BA   gbasedbt rootdbs
47d757f0    2      0x1070001  2       1        2048    N PBA  gbasedbt plog
49102dd0    3      0x60001  3       1        2048    N BA   gbasedbt llog
47c7cdd0    4      0x60001  4       1        2048    N BA   gbasedbt datadbs1
47c80dd8    5      0x60001  5       1        2048    N BA   gbasedbt datadbs2
4906dda8    6      0x60001  6       1        2048    N BA   gbasedbt datadbs3
49655d78    7      0x60001  7       1        8192    N BA   gbasedbt data8dbs1
49860d78    8      0x60001  8       1        8192    N BA   gbasedbt data8dbs2
48d06d78    9      0x60001  9       1        8192    N BA   gbasedbt data8dbs3
48d20d58    10     0x42001  10      1        8192    N TBA  gbasedbt tmpdbspace
49942570    11     0x68001  11      1        2048    N SBA  gbasedbt sbpace1

```

```

47c26b38      12      0x4a001    12      1      2048      N UBA      gbasedbt tmpspace
12 active, 2047 maximum

Chunks
address  chunk/dbs offset size  free  bpages flags pathname
454b0258 1    1    0    78848 68251    PO-B-- /opt/gbase/storage/rootdbs
49408028 2    2    0    32768 0        PO-BE- /opt/gbase/storage/ol_gbasedbt1210_plog_p_1
48d20028 3    3    0    32768 15       PO-BE- /opt/gbase/storage/ol_gbasedbt1210_lllog_p_1
48fae028 4    4    0    32768 32715    PO-BE- /opt/gbase/storage/ol_gbasedbt1210_datadbs1_p_1
48e2f028 5    5    0    32768 32715    PO-BE- /opt/gbase/storage/ol_gbasedbt1210_datadbs2_p_1
48df0028 6    6    0    32768 32715    PO-BE- /opt/gbase/storage/ol_gbasedbt1210_datadbs3_p_1
48d5f028 7    7    0    8192  8139    PO-BE- /opt/gbase/storage/ol_gbasedbt1210_data8dbs1_p_1
48c4a028 8    8    0    8192  8139    PO-BE- /opt/gbase/storage/ol_gbasedbt1210_data8dbs2_p_1
48e89028 9    9    0    8192  8139    PO-BE- /opt/gbase/storage/ol_gbasedbt1210_data8dbs3_p_1
4985d028 10   10   0    8192  8139    PO-BE- /opt/gbase/storage/ol_gbasedbt1210_tmpdbspace_p_1
49834028 11   11   0    32768 23802 23802 POSB-- /opt/gbase/storage/ol_gbasedbt1210_sbpace1_p_1
          Metadata 1745  8466  1745
49506028 12   12   0    32768 23802 23802 POSB-- /opt/gbase/storage/ol_gbasedbt1210_tmpspace_p_1
          Metadata 1745  8466  1745
12 active, 32766 maximum

NOTE: The values in the "size" and "free" columns for DBspace chunks are
      displayed in terms of "pgsize" of the DBspace to which they belong.

Expanded chunk capacity mode: always

```

3.5. 数据库自定义安装

【注意】3.3 自动化安装脚本安装（适用于快速熟悉与开发环境）、3.4 数据库典型安装（适用于快速开发）和 3.5 数据库自定义安装（适用于生产环境）三种方式中，选择一种方式安装即可。

数据库自定义，需要手工创建用户组及用户，只进行软件安装，数据库空间创建、实例配置均需要手工完成。

自定义安装安装方式，可使用本地文件系统、逻辑卷组或者磁盘裸设备等各种可用设备。适用于生产环境配置，或者数据库管理员熟悉安装配置。对操作者的要求要高。

3.5.1. 创建用户组及用户

使用 root 用户创建 gbasedbt 组

```
groupadd -g 1000 gbasedbt
```

创建 gbasedbt 用户

```
useradd -g gbasedbt -u 1000 -d /home/gbase -m -s /bin/bash gbasedbt
```

设置 gbasedbt 用户密码

```
passwd gbasedbt
```

注意 1: gbasedbt 用户是数据库系统管理员用户，使用操作系统验证密码访问，密码强度应当符合操作系统安全要求。

注意 2: 某些操作系统中，首次设置的密码无效，需要首次通过 ssh 远程登陆并修改密码才能生效。

3.5.2. 磁盘规划

【注意】3.5.2.1 使用本地文件系统、3.5.2.2 使用逻辑卷组管理和 3.5.2.3 使用 LUN 磁盘裸设备管理三种方式，按实际环境选择一种方式安装即可。

GBase 8s 数据库可使用本地文件系统、逻辑卷组或者 LUN 磁盘裸设备等各种可用设备。各种方式各有优劣，以下按本地文件系统、逻辑卷组和 LUN 磁盘裸设备三种方式进行配置，按实际可选择其中一个方式。

示例安装使用以下的空间规划：

数据库空间名称	路径	大小	说明
rootdbs	/data/gbase/rootchk	1GB	根数据库空间
plogdbs	/data/gbase/plogchk	1GB	物理日志
llogdbs	/data/gbase/llogchk	1GB	逻辑日志
sbspace01	/data/gbase/sbspace1	1GB	智能大对象空间，视需要指定大小
tempdbs01	/data/gbase/tempchk01	1GB	临时数据库空间，16KB 页大小
datadbs01	/data/gbase/datachk01	1GB	数据使用的数据库空间，16KB 页大小

3.5.2.1. 使用本地文件系统

使用本地文件系统的方式，需要确保目录所在的文件系统上有足够的空间，gbasedbt 用户对/data/gbase 目录有访问权限，目录下的数据库使用的文件的属主和

属组均为 `gbasedbt`，权限为属主和属组可读和可写（`0660`）

使用本地文件系统方式，支持单机、本地高可用集群（**HAC**）和远程容灾高可用集群（**RHAC**）。

```
root@netsky-PC:~# mkdir -p /data/gbase (创建目录)

root@netsky-PC:~# chmod 755 /data

root@netsky-PC:~# chmod 755 /data/gbase

root@netsky-PC:~# chown gbasedbt:gbasedbt /data/gbase

root@netsky-PC:~# su - gbasedbt

gbasedbt@netsky-PC:~$ cd /data/gbase

gbasedbt@netsky-PC:/data/gbase$ touch rootchk plogchk llogchk tempchk01 sbspace01
datachk01 (创建文件)

gbasedbt@netsky-PC:/data/gbase$ chmod 0660 * (修改权限)

gbasedbt@netsky-PC:/data/gbase$ ls -al
总用量 0
drwxr-xr-x 2 gbasedbt gbasedbt 101 1月 19 15:15 .
drwxr-xr-x 7 root      root      65 1月 19 15:12 ..
-rw-rw---- 1 gbasedbt gbasedbt  0 1月 19 15:15 datachk01
-rw-rw---- 1 gbasedbt gbasedbt  0 1月 19 15:15 llogchk
-rw-rw---- 1 gbasedbt gbasedbt  0 1月 19 15:15 plogchk
-rw-rw---- 1 gbasedbt gbasedbt  0 1月 19 15:15 rootchk
-rw-rw---- 1 gbasedbt gbasedbt  0 1月 19 15:15 sbspace01
-rw-rw---- 1 gbasedbt gbasedbt  0 1月 19 15:15 tempchk01
```

3.5.2.2. 使用逻辑卷组管理

使用逻辑卷级管理（**LVM**）的方式，`gbasedbt` 用户对 `/data/gbase` 目录有访问权限，数据库使用的逻辑卷组的属主和属组均为 `gbasedbt`，权限为属主和属组可读和可写（`0660`）

使用逻辑卷组管理的方式，支持单机、本地高可用集群（**HAC**）和远程容灾高可用集群（**RHAC**）。

逻辑卷组管理的一般步骤为：创建 `pv`，创建卷组 `vg`，创建逻辑卷 `lv`，修改属

主和权限，创建链接文件。

示例使用磁盘/dev/sdb 创建逻辑卷，对于 LVM 的参数优化，需要参考操作系统的优化。

```
root@netsky-PC:~# pvcreate /dev/sdb (创建物理卷)
Physical volume "/dev/sdb" successfully created.

root@netsky-PC:~# vgcreate vgdata /dev/sdb (创建卷组)
Volume group "vgdata" successfully created

root@netsky-PC:~# lvcreate -n lvrootchk -L 1050M vgdata (创建逻辑卷)
Rounding up size to full physical extent <1.03 GiB
Logical volume "lvrootchk" created.

root@netsky-PC:~# lvcreate -n lvplogchk -L 1050M vgdata (创建逻辑卷)
Rounding up size to full physical extent <1.03 GiB
Logical volume "lvplogchk" created.

root@netsky-PC:~# lvcreate -n lvllogchk -L 1050M vgdata (创建逻辑卷)
Rounding up size to full physical extent <1.03 GiB
Logical volume "lvllogchk" created.

root@netsky-PC:~# lvcreate -n lvtempchk01 -L 1050M vgdata (创建逻辑卷)
Rounding up size to full physical extent <1.03 GiB
Logical volume "lvtempchk01" created.

root@netsky-PC:~# lvcreate -n lvsbpace01 -L 1050M vgdata (创建逻辑卷)
Rounding up size to full physical extent <1.03 GiB
Logical volume "lvsbpace01" created.

root@netsky-PC:~# lvcreate -n lvdatachk01 -L 1050M vgdata (创建逻辑卷)
Rounding up size to full physical extent <1.03 GiB
Logical volume "lvdatachk01" created.

root@netsky-PC:~# cd /dev/vgdata

root@netsky-PC:/dev/vgdata# ls -al
总用量 0
drwxr-xr-x  2 root root  160 1月  19 16:16 .
drwxr-xr-x 18 root root 3480 1月  19 16:16 ..
lrwxrwxrwx  1 root root   7 1月  19 16:16 lvdatachk01 -> ../dm-5
lrwxrwxrwx  1 root root   7 1月  19 16:15 lvllogchk -> ../dm-2
lrwxrwxrwx  1 root root   7 1月  19 16:15 lvplogchk -> ../dm-1
lrwxrwxrwx  1 root root   7 1月  19 16:15 lvrootchk -> ../dm-0
```

```
lrwxrwxrwx 1 root root 7 1月 19 16:15 lvsubspace01 -> ../dm-4
lrwxrwxrwx 1 root root 7 1月 19 16:15 lvtempchk01 -> ../dm-3

root@netsky-PC:/dev/vgdata# chown gbasedbt:gbasedbt lv* (修改属主、组)

root@netsky-PC:/dev/vgdata# chmod 0660 lv* (修改权限)

root@netsky-PC:/dev/vgdata# ls -al ../dm-*
brw-rw---- 1 gbasedbt gbasedbt 253, 0 1月 19 16:15 ../dm-0
brw-rw---- 1 gbasedbt gbasedbt 253, 1 1月 19 16:15 ../dm-1
brw-rw---- 1 gbasedbt gbasedbt 253, 2 1月 19 16:15 ../dm-2
brw-rw---- 1 gbasedbt gbasedbt 253, 3 1月 19 16:15 ../dm-3
brw-rw---- 1 gbasedbt gbasedbt 253, 4 1月 19 16:15 ../dm-4
brw-rw---- 1 gbasedbt gbasedbt 253, 5 1月 19 16:16 ../dm-5

root@netsky-PC:/dev/vgdata# mkdir -p /data/gbase

root@netsky-PC:/dev/vgdata# chmod 755 /data

root@netsky-PC:/dev/vgdata# chmod 755 /data/gbase

root@netsky-PC:/dev/vgdata# chown gbasedbt:gbasedbt /data/gbase

root@netsky-PC:/dev/vgdata# su - gbasedbt

gbasedbt@netsky-PC:~$ cd /data/gbase (以下创建链接)
gbasedbt@netsky-PC:/data/gbase$ ln -s /dev/vgdata/lvrootchk rootchk
gbasedbt@netsky-PC:/data/gbase$ ln -s /dev/vgdata/lvplogchk plogchk
gbasedbt@netsky-PC:/data/gbase$ ln -s /dev/vgdata/lvllogchk llogchk
gbasedbt@netsky-PC:/data/gbase$ ln -s /dev/vgdata/lvtempchk01 tempchk01
gbasedbt@netsky-PC:/data/gbase$ ln -s /dev/vgdata/lvsubspace01 subspace01
gbasedbt@netsky-PC:/data/gbase$ ln -s /dev/vgdata/lvdatachk01 datachk01

gbasedbt@netsky-PC:/data/gbase$ ls -al
总用量 0
drwxr-xr-x 2 gbasedbt gbasedbt 102 1月 19 16:18 .
drwxr-xr-x 7 root root 65 1月 19 15:12 ..
lrwxrwxrwx 1 gbasedbt gbasedbt 23 1月 19 16:18 datachk01 -> /dev/vgdata/lvdatachk01
lrwxrwxrwx 1 gbasedbt gbasedbt 21 1月 19 16:17 llogchk -> /dev/vgdata/lvllogchk
lrwxrwxrwx 1 gbasedbt gbasedbt 21 1月 19 16:17 plogchk -> /dev/vgdata/lvplogchk
lrwxrwxrwx 1 gbasedbt gbasedbt 21 1月 19 16:17 rootchk -> /dev/vgdata/lvrootchk
lrwxrwxrwx 1 gbasedbt gbasedbt 23 1月 19 16:17 subspace01 -> /dev/vgdata/lvsubspace01
lrwxrwxrwx 1 gbasedbt gbasedbt 23 1月 19 16:17 tempchk01 -> /dev/vgdata/lvtempchk01
```

3.5.2.3. 使用 LUN 磁盘裸设备管理

使用 LUN 磁盘裸设备管理的方式, gbasedbt 用户对 /data/gbase 目录有访问权限, 数据库使用的 LUN 设备或者 LUN 设备分区的属主和属组均为 gbasedbt, 权限为属主和属组可读和可写 (0660)

使用 LUN 磁盘裸设备管理的方式, 支持单机、本地高可用集群 (HAC) 和远程容灾高可用集群 (RHAC)。如果 LUN 是共享磁盘设备, 支持共享存储高可用集群 (SSC)。

LUN 磁盘裸设备管理的一般步骤为: LUN 分区 (如果直接使用 LUN, 就不需要分区), 创建裸设备, 修改属主和权限, 创建链接文件。

示例使用 LUN 磁盘 /dev/sdb, 大小为 20G。对于小于 2TB 的 LUN 磁盘, 可以使用 fdisk 分区工具; 对于大于 2TB 的 LUN 磁盘, 只能使用 parted 分区工具。示例使用 parted 进行分区管理。

使用分区管理时, 磁盘分区应当一次性分区完成, 当设备被使用时, 将无法完成分区操作, 只能等使用设备的程序关闭后才可分区。

```
root@netsky-PC:~# parted /dev/sdb (划分区)
GNU Parted 3.2
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mklabel gpt
(parted) mkpart primary 1MB 1050MB
(parted) mkpart pprimary 1050MB 2100MB
(parted) mkpart pprimary 2100MB 3150MB
(parted) mkpart pprimary 3150MB 4200MB
(parted) mkpart pprimary 4200MB 5250MB
(parted) mkpart pprimary 5250MB 6300MB
(parted) print
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdb: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name      Flags
  1      1049kB  1050MB  1049MB             primary
  2      1050MB  2100MB  1051MB             pprimary
  3      2100MB  3150MB  1050MB             pprimary
```

4	3150MB	4200MB	1050MB	primary
5	4200MB	5250MB	1051MB	primary
6	5250MB	6300MB	1050MB	primary

示例使用的是直接的/dev/sdb，实际环境中应当要求使用多路径方式。

查看划好的分区

```
root@netsky-PC:~# cd /dev

root@netsky-PC:/dev# ls -al sdb*
brw-rw---- 1 root disk 8, 16 1月 19 17:15 sdb
brw-rw---- 1 root disk 8, 17 1月 19 17:15 sdb1
brw-rw---- 1 root disk 8, 18 1月 19 17:15 sdb2
brw-rw---- 1 root disk 8, 19 1月 19 17:15 sdb3
brw-rw---- 1 root disk 8, 20 1月 19 17:15 sdb4
brw-rw---- 1 root disk 8, 21 1月 19 17:15 sdb5
brw-rw---- 1 root disk 8, 22 1月 19 17:15 sdb6
```

编辑 60-raw-gbase.rules，增加 raw 化配置

```
root@netsky-PC:/dev# cd /usr/lib/udev/rules.d

root@netsky-PC:/usr/lib/udev/rules.d# vi 60-raw-gbase.rules      (编辑配置文件)

root@netsky-PC:/usr/lib/udev/rules.d# more 60-raw-gbase.rules
ACTION=="add", KERNEL=="sdb1", RUN+="/usr/sbin/raw /dev/raw/raw1 %N"
ACTION=="add", KERNEL=="sdb2", RUN+="/usr/sbin/raw /dev/raw/raw2 %N"
ACTION=="add", KERNEL=="sdb3", RUN+="/usr/sbin/raw /dev/raw/raw3 %N"
ACTION=="add", KERNEL=="sdb4", RUN+="/usr/sbin/raw /dev/raw/raw4 %N"
ACTION=="add", KERNEL=="sdb5", RUN+="/usr/sbin/raw /dev/raw/raw5 %N"
ACTION=="add", KERNEL=="sdb6", RUN+="/usr/sbin/raw /dev/raw/raw6 %N"
KERNEL=="raw1", OWNER="gbasedbt" GROUP="gbasedbt", MODE="0660"
KERNEL=="raw2", OWNER="gbasedbt" GROUP="gbasedbt", MODE="0660"
KERNEL=="raw3", OWNER="gbasedbt" GROUP="gbasedbt", MODE="0660"
KERNEL=="raw4", OWNER="gbasedbt" GROUP="gbasedbt", MODE="0660"
KERNEL=="raw5", OWNER="gbasedbt" GROUP="gbasedbt", MODE="0660"
KERNEL=="raw6", OWNER="gbasedbt" GROUP="gbasedbt", MODE="0660"
```

注：raw 这个命令，在不同的系统中位置不同，一般位置/usr/bin 或者/usr/sbin 目录下。手工执行 raw，如果报错如下：

```
root@netsky-PC:~# raw
raw: Cannot open master raw device '/dev/raw/rawctl': 没有那个文件或目录
```

则需要通过 modprobe raw 加载到模块中

```
root@netsky-PC:~# modprobe raw      (加载 raw)
```

重新加载配置文件

```
root@netsky-PC:~# udevadm control --reload-rules
```

重新加载磁盘，检查 raw 设备

```
root@netsky-PC:/usr/lib/udev/rules.d# partprobe /dev/sdb
```

(重新加载磁盘)

```
root@netsky-PC:/usr/lib/udev/rules.d# raw -qa
```

```
/dev/raw/raw1: bound to major 8, minor 17
```

```
/dev/raw/raw2: bound to major 8, minor 18
```

```
/dev/raw/raw3: bound to major 8, minor 19
```

```
/dev/raw/raw4: bound to major 8, minor 20
```

```
/dev/raw/raw5: bound to major 8, minor 21
```

```
/dev/raw/raw6: bound to major 8, minor 22
```

```
root@netsky-PC:/usr/lib/udev/rules.d# ls -al /dev/raw/raw*
```

```
crw-rw---- 1 gbasedbt gbasedbt 162, 1 1月 21 11:10 /dev/raw/raw1
```

```
crw-rw---- 1 gbasedbt gbasedbt 162, 2 1月 21 11:10 /dev/raw/raw2
```

```
crw-rw---- 1 gbasedbt gbasedbt 162, 3 1月 21 11:10 /dev/raw/raw3
```

```
crw-rw---- 1 gbasedbt gbasedbt 162, 4 1月 21 11:10 /dev/raw/raw4
```

```
crw-rw---- 1 gbasedbt gbasedbt 162, 5 1月 21 11:10 /dev/raw/raw5
```

```
crw-rw---- 1 gbasedbt gbasedbt 162, 6 1月 21 11:10 /dev/raw/raw6
```

```
crw-rw---- 1 root disk 162, 0 1月 21 10:59 /dev/raw/rawctl
```

创建链接文件

```
root@netsky-PC:~# mkdir /data/gbase
```

```
root@netsky-PC:~# chmod 755 /data
```

```
root@netsky-PC:~# chmod 755 /data/gbase
```

```
root@netsky-PC:~# chown gbasedbt:gbasedbt /data/gbase
```

```
root@netsky-PC:~# su - gbasedbt
```

```
gbasedbt@netsky-PC:~$ cd /data/gbase
```

(以下创建链接)

```
gbasedbt@netsky-PC:/data/gbase$ ln -s /dev/raw/raw1 rootchk
```

```
gbasedbt@netsky-PC:/data/gbase$ ln -s /dev/raw/raw2 plogchk
```

```
gbasedbt@netsky-PC:/data/gbase$ ln -s /dev/raw/raw3 llogchk
```

```
gbasedbt@netsky-PC:/data/gbase$ ln -s /dev/raw/raw4 tempchk01
```

```
gbasedbt@netsky-PC:/data/gbase$ ln -s /dev/raw/raw5 sbspace01
```

```
gbasedbt@netsky-PC:/data/gbase$ ln -s /dev/raw/raw6 datachk01
```

```
gbasedbt@netsky-PC:/data/gbase$ ls -l
```

```
总用量 0
```

```
lrwxrwxrwx 1 gbasedbt gbasedbt 13 1月 24 09:41 datachk01 -> /dev/raw/raw6
```

```
lrwxrwxrwx 1 gbasedbt gbasedbt 13 1月 24 09:40 llogchk -> /dev/raw/raw3
```

```
lrwxrwxrwx 1 gbasedbt gbasedbt 13 1月 24 09:40 plogchk -> /dev/raw/raw2
lrwxrwxrwx 1 gbasedbt gbasedbt 13 1月 24 09:40 rootchk -> /dev/raw/raw1
lrwxrwxrwx 1 gbasedbt gbasedbt 13 1月 24 09:41 sbspace01 -> /dev/raw/raw5
lrwxrwxrwx 1 gbasedbt gbasedbt 13 1月 24 09:41 tempchk01 -> /dev/raw/raw4
```

3.5.3. 安装数据库软件

解压数据库软件，执行静默安装，仅安装软件文件到指定目录下

```
root@netsky-PC:~/server# tar -xvf ../GBase8sV8.8_3.0.0_1_93e040_RHEL6_x86_64.tar
doc/
doc/ids_machine_notes_12.10.txt
doc/ids_unix_relnotes_12.10.html
ids_install
ids.properties
onsecurity
.gbase.properties

root@netsky-PC:~/server# ./ids_install -i silent \
-DLICENSE_ACCEPTED=TRUE -DUSER_INSTALL_DIR=/opt/gbase
```

(执行静默安装)

3.5.4. 配置、初始化数据库实例

初始化数据库实例的步骤大致为：配置用户环境变量，数据库配置文件，初始化数据库实例，调整数据库使用的空间。

1)、配置用户环境变量配置文件.bash_profile

编辑或者创建 gbasedbt 用户目录下的.bash_profile 配置文件，内容如下：

```
root@netsky-PC:~/server# su - gbasedbt

gbasedbt@netsky-PC:~$ vi .bash_profile

gbasedbt@netsky-PC:~$ more .bash_profile
export GBASEDBTDIR=/opt/gbase (安装目录)
export GBASEDBTSERVER=gbase01 (实例名称)
export ONCONFIG=onconfig.$GBASEDBTSERVER (配置文件名称)
export PATH=$GBASEDBTDIR/bin:${PATH} (加入 PATH)
export GBASEDBTSQLHOSTS=/opt/gbase/etc/sqlhosts (SQLHOSTS 路径)

export DB_LOCALE=zh_CN.utf8 (数据库编码)
export CLIENT_LOCALE=zh_CN.utf8 (客户端编码)
```



```
export GL_USEGLU=1 (启用 GLU)
export DBDATE="Y4MD-" (指定 DBDATE)
export DBACCESS_SHOW_TIME=1
```

2)、配置 ONCONFIG 配置文件 onconfig.gbase01

复制\$GBASEDBTDIR/etc/onconfig.std 为\$ONCONFIG 配置文件，修改主要参数内容如下：

```
ROOTPATH /data/gbase/rootchk # ROOTDBS 路径
ROOTSIZE 1024000 # ROOTDBS 大小
PHYSBUFF 1024 # 物理日志缓存
LOGBUFF 1024 # 逻辑日志缓存
DBSPACETEMP tempdbs01 # 默认的临时数据库空间
SBSPACENAME sbspace01 # 默认的智能大对象空间
DBSERVERNAME gbase01 # 实例名称
NETTYPE soctcp, 1, 100, NET # 网络连接
MULTIPROCESSOR 1 # 启用多 CPU
VPCLASS cpu, num=2, noage # CPU VP 数量
CLEANERS 32 # 页清理数量
LOCKS 2000000 # 初始锁的数量
DEF_TABLE_LOCKMODE row # 默认锁的类型
SHMVIRTSIZE 512000 # 虚拟内存大小
LTAPEDEV /dev/null # 逻辑日志备份目录
STACKSIZE 2048 # 堆栈大小
ALLOW_NEWLINE 1 # 允许字符型字段换行
DS_TOTAL_MEMORY 1024000 # 决策性操作使用的内存
DS_NONPDQ_QUERY_MEM 256000 # 非决策性操作使用的内存
TEMPTAB_NOLOG 1 # 指定临时表不使用日志
DUMPSHMEM 0 # 不 DUMP 内存
USERMAPPING ADMIN # 使用内部用户
BUFFERPOOL size=16k, buffers=50000, lrus=8, lru_min_dirty=50, lru_max_dirty=60
```

3)、配置 GBASEDBTSQLHSOTS 配置文件 sqlhosts

编辑或者创建 sqlhosts 配置文件

```
gbasedbt@netsky-PC:~$ cd $GBASEDBTDIR/etc/

gbasedbt@netsky-PC:/opt/gbase/etc$ vi sqlhosts

gbasedbt@netsky-PC:/opt/gbase/etc$ more sqlhosts
# 实例名称 网络协议 IP 地址 端口号
gbase01 onsoctcp 0.0.0.0 9088
```

4)、配置内部用户使用的 allowed.surrogates（位于/etc/gbasedbt 目录下）

创建/etc/gbasedbt 目录，创建 allowed.surrogates 配置文件，内容如下：

```
root@netsky-PC:~# mkdir -p /etc/gbasedbt

root@netsky-PC:~# chmod 755 /etc/gbasedbt

root@netsky-PC:~# echo "USERS:daemon" > /etc/gbasedbt/allowed.surrogates

root@netsky-PC:~# chmod 644 /etc/gbasedbt/allowed.surrogates

root@netsky-PC:~# more /etc/gbasedbt/allowed.surrogates
# 指定使用 daemon 用户为代理用户
USERS:daemon
```

5)、初始化数据库

通过 `oninit -ivy` 初始化数据库实例。

警告：-i 参数仅限第一次初始化数据库实例时使用，后期启动时应使用 `oninit`。

```
gbasedbt@netsky-PC:~$ oninit -ivy (初始化实例)
Reading configuration file '/opt/gbase/etc/onconfig.gbase01'...succeeded
Creating /GBASEDBTMP/.infxdirs...succeeded
Allocating and attaching to shared memory...succeeded
Creating resident pool 279630 kbytes...succeeded
Creating infos file "/opt/gbase/etc/.infos.gbase01"...succeeded
Linking conf file "/opt/gbase/etc/.conf.gbase01"...succeeded
Initializing rhead structure...rhlock_t 262144 (8192K)... rlock_t (265625K)...
Writing to infos file...succeeded
Initialization of Encryption...succeeded
Initializing ASF...succeeded
Initializing Dictionary Cache and SPL Routine Cache...succeeded
Bringing up ADM VP...succeeded
Creating VP classes...succeeded
Forking main_loop thread...succeeded
Initializing DR structures...succeeded
Forking 1 'soctcp' listener threads...succeeded
Starting tracing...succeeded
Initializing 32 flushers...succeeded
Initializing log/checkpoint information...succeeded
Initializing dbspaces...succeeded
Opening primary chunks...succeeded
Validating chunks...succeeded
Creating database partition...succeeded
Initialize Async Log Flusher...succeeded
Starting B-tree Scanner...succeeded
Init ReadAhead Daemon...succeeded
Init DB Util Daemon...succeeded
Initializing DBSPACETEMP list...succeeded
```

```

Init Auto Tuning Daemon...succeeded
Checking database partition index...succeeded
Initializing dataskip structure...succeeded
Checking for temporary tables to drop...succeeded
Updating Global Row Counter...succeeded
Forking onmode_mon thread...succeeded
Creating periodic thread...succeeded
Creating periodic thread...succeeded
Starting scheduling system...succeeded
Verbose output complete: mode = 5

```

检查数据库侦听，确认已经启动

```

gbasedbt@netsky-PC:~$ onstat -g ntt

GBase Database Server Version 12.10.FC4G1AEE -- On-Line -- Up 00:06:47 -- 1716640 Kbytes

global network information:
#netscb connects      read      write    q-free  q-limits  q-exceed alloc/max
  2/   3       11      4262    4261    2/   2  170/  10    0/   0    2/   2

Individual thread network information (times):
      netscb thread name      sid    open    read    write address
      586e1bb0 soctcplst      3 11:08:08 11:11:12      0.0.0.0|9088|soctcp
      57100ab0 soctcppoll      2 11:08:09

```

查询数据库运行日志 `online.log`，4 个 `sys` 开头的系统库均已经创建完成，表示实例初始化完成。

```

gbasedbt@netsky-PC:~$ tail -n 1000 /opt/gbase/tmp/online.log | grep 'sys'
11:08:09 Building 'sysmaster' database ...
11:08:15 'sysmaster' database built successfully.
11:08:16 'sysutils' database built successfully.
11:08:16 'sysuser' database built successfully.
11:08:22 Building 'sysadmin' database ...
11:08:22 'sysadmin' database built successfully.

```

6)、增加数据库空间

刚初始化的实例，仅有一个根数据库空间 `rootdbs`。实际业务环境需要有独立的物理日志、逻辑日志、临时数据库空间、智能大对象空间和业务数据库空间。

```

gbasedbt@netsky-PC:~$ onstat -d

GBase Database Server Version 12.10.FC4G1AEE -- On-Line -- Up 00:05:38 -- 1716640 Kbytes

Dbspaces
address  number  flags      fchunk  nchunks  pgsz   flags  owner  name

```

```

5657e028 1 0x40001 1 1 2048 N BA gbasedbt rootdbs
1 active, 2047 maximum

Chunks
address chunk/dbs offset size free bpages flags pathname
5657e258 1 1 0 512000 446665 P0-B-- /data/gbase/rootchk
1 active, 32766 maximum

NOTE: The values in the "size" and "free" columns for DBspace chunks are
displayed in terms of "pgsize" of the DBspace to which they belong.

Expanded chunk capacity mode: always

```

通过 onspaces 创建相应的空间

```

# 物理日志使用的数据库空间
gbasedbt@netsky-PC:~$ onspaces -c -d plogdbs -p /data/gbase/plogchk -o 0 -s 1024000
Verifying physical disk space, please wait ...
Space successfully added.

** WARNING ** A level 0 archive of Root DBSpace will need to be done.

# 逻辑日志使用的数据库空间
gbasedbt@netsky-PC:~$ onspaces -c -d llogdbs -p /data/gbase/llogchk -o 0 -s 1024000
Verifying physical disk space, please wait ...
Space successfully added.

** WARNING ** A level 0 archive of Root DBSpace will need to be done.

# 临时数据库空间
gbasedbt@netsky-PC:~$ onspaces -c -d tempdbs01 -t -k 16 -p /data/gbase/tempchk01 -o
0 -s 1024000
Verifying physical disk space, please wait ...
Space successfully added.

# 智能大对象空间
gbasedbt@netsky-PC:~$ onspaces -c -S sbospace01 -p /data/gbase/sbospace01 -o 0 -s
1024000
Verifying physical disk space, please wait ...
Space successfully added.

** WARNING ** A level 0 archive of Root DBSpace will need to be done.

# 业务使用的数据库空间
gbasedbt@netsky-PC:~$ onspaces -c -d datadbs01 -k 16 -p /data/gbase/datachk01 -o 0

```

```

-s 1024000
Verifying physical disk space, please wait ...
Space successfully added.

** WARNING ** A level 0 archive of Root DBSpace will need to be done.

# 创建空间后
gbasedbt@netsky-PC:~$ onstat -d

GBase Database Server Version 12.10.FC4G1AEE -- On-Line -- Up 00:24:19 -- 1716640 Kbytes

Dbspaces
address  number  flags    fchunk  nchunks  pgsz   flags  owner  name
5657e028 1       0x60001 1        1        2048   N BA   gbasedbt rootdbs
5a737da8 2       0x60001 2        1        2048   N BA   gbasedbt plogdbs
5a723d78 3       0x60001 3        1        2048   N BA   gbasedbt llogdbs
5a744520 4       0x42001 4        1        16384  N TBA  gbasedbt tempdbs01
5a99ad78 5       0x68001 5        1        2048   N SBA  gbasedbt sbspace01
597eeaf8 6       0x60001 6        1        16384  N BA   gbasedbt datadbs01
 6 active, 2047 maximum

Chunks
address  chunk/dbs  offset  size  free  bpages  flags  pathname
5657e258 1    1    0    512000  446663  PO-B-- /data/gbase/rootchk
5a3cd028 2    2    0    512000  511947  PO-B-- /data/gbase/plogchk
5a732028 3    3    0    512000  511947  PO-B-- /data/gbase/llogchk
5a409028 4    4    0    64000   63947  PO-B-- /data/gbase/tempchk01
5a447028 5    5    0    512000  477465  477465  POSB-- /data/gbase/sbspace01
                                     Metadata 34482  25659  34482
5a824028 6    6    0    64000   63947  PO-B-- /data/gbase/datachk01
 6 active, 32766 maximum

NOTE: The values in the "size" and "free" columns for DBspace chunks are
      displayed in terms of "pgsize" of the DBspace to which they belong.

Expanded chunk capacity mode: always

```

将物理日志从根数据库空间移到物理日志专用的数据库空间

```

# 移动物理日志，并改变大小
gbasedbt@netsky-PC:~$ onparams -p -s 1000000 -d plogdbs -y
Log operation started. To monitor progress, use the onstat -l command.
** WARNING ** Because the physical log has been modified, a level 0 archive
must be taken of the following spaces before an incremental archive will be
permitted for them: rootdbs plogdbs

```

```
(see Database Server Administrator's manual)

# 查看物理日志位置及大小
gbasedbt@netsky-PC:~$ onstat -l

GBase Database Server Version 12.10.FC4G1AEE -- On-Line -- Up 00:31:03 -- 1716640 Kbytes

Physical Logging
Buffer bufused  bufsize  numpages  numwrits  pages/io
P-2  5          512      534       16        33.38
      phybegin      physize  phypos    phyused   %used
      2:53          500000  0         5         0.00
```

将逻辑日志文件从根数据库空间移到逻辑日志专用的数据库空间，默认初始化时有 6 个逻辑日志文件位于 rootdbs 上，逻辑日志只有已经备份或者新加未使用的情况下才能删除，同时整个实例中不能少于 3 个逻辑日志。按照这样的规则：在逻辑日志专用的数据库空间上创建新的逻辑日志文件，然后将当前的逻辑日志移动到逻辑日志专用的数据库空间上，删除在 rootdbs 上的逻辑日志文件，以达到移动逻辑日志的目的。

```
# 当前的逻辑日志位于 rootdbs 上
gbasedbt@netsky-PC:~$ onstat -l

GBase Database Server Version 12.10.FC4G1AEE -- On-Line -- Up 00:31:03 -- 1716640 Kbytes

Physical Logging
Buffer bufused  bufsize  numpages  numwrits  pages/io
P-2  5          512      534       16        33.38
      phybegin      physize  phypos    phyused   %used
      2:53          500000  0         5         0.00

Logical Logging
Buffer bufused  bufsize  numrecs   numpages  numwrits  recs/pages  pages/io
L-1  0          512      166624    14241     3765      11.7        3.8
      Subsystem    numrecs   Log Space used
      OLDRSAM      166297    23906880
      SBLOB        5         252
      HA          15        660
      DDL         307       106836

address      number  flags    uniqid   begin          size  used  %used
566f0f88    1      U-B----  1        1:25263        5000  5000  100.00
5917a570    2      U-B----  2        1:30263        5000  5000  100.00
5917a5d8    3      U---C-L  3        1:35263        5000  4241  84.82
5917a640    4      A-----  0        1:40263        5000   0    0.00
```

```

5917a6a8      5      A----- 0      1:45263      5000      0      0.00
5917a710      6      A----- 0      1:50263      5000      0      0.00
  6 active, 6 total

```

在 llogdbs 上增加 10 个逻辑日志, 每个大小是 100MB

```
gbasedbt@netsky-PC:~$ for i in {1..10}
```

```
> do
```

```
> onparams -a -d llogdbs -s 100000
```

```
> done
```

```
Log operation started. To monitor progress, use the onstat -l command.
```

```
Logical log successfully added.
```

```
Log operation started. To monitor progress, use the onstat -l command.
```

```
Logical log successfully added.
```

```
Log operation started. To monitor progress, use the onstat -l command.
```

```
Logical log successfully added.
```

```
Log operation started. To monitor progress, use the onstat -l command.
```

```
Logical log successfully added.
```

```
Log operation started. To monitor progress, use the onstat -l command.
```

```
Logical log successfully added.
```

```
Log operation started. To monitor progress, use the onstat -l command.
```

```
Logical log successfully added.
```

```
Log operation started. To monitor progress, use the onstat -l command.
```

```
Logical log successfully added.
```

```
Log operation started. To monitor progress, use the onstat -l command.
```

```
Logical log successfully added.
```

```
Log operation started. To monitor progress, use the onstat -l command.
```

```
Logical log successfully added.
```

```
Log operation started. To monitor progress, use the onstat -l command.
```

```
Logical log successfully added.
```

显示新增加的逻辑日志, 位于 logdbs 上

```
gbasedbt@netsky-PC:~$ onstat -l
```

```
GBase Database Server Version 12.10.FC4G1AEE -- On-Line -- Up 00:36:32 -- 1716640 Kbytes
```

Physical Logging

```
Buffer bufused  bufsize  numpages  numwrits  pages/io
```

```
P-2  0      512      579      27      21.44
```

```
  phybegin      physize  phypos  phyused  %used
```

```
  2:53      500000  45      0      0.00
```

Logical Logging

```
Buffer bufused  bufsize  numrecs  numpages  numwrits  recs/pages  pages/io
```

```
L-3  0      512      166697  14273     3797      11.7      3.8
```

```

Subsystem    numrecs    Log Space used
OLDRSAM      166359    23911564
SBLOB        5          252
HA           26        1144
DDL          307       106836

address      number    flags    uniqid    begin          size    used    %used
566f0f88    1        U-B----  1        1:25263       5000   5000   100.00
5917a570    2        U-B----  2        1:30263       5000   5000   100.00
5917a5d8    3        U--C-L  3        1:35263       5000   4273   85.46
5917a640    4        A-----  0        1:40263       5000    0     0.00
5917a6a8    5        A-----  0        1:45263       5000    0     0.00
5917a710    6        A-----  0        1:50263       5000    0     0.00
58cf95b0    7        A-----  0        3:53          50000   0     0.00
58cf9680    8        A-----  0        3:50053      50000   0     0.00
58cf9618    9        A-----  0        3:100053    50000   0     0.00
58cf96e8   10       A-----  0        3:150053    50000   0     0.00
58cf9758   11       A-----  0        3:200053    50000   0     0.00
58cf97d0   12       A-----  0        3:250053    50000   0     0.00
58cf9850   13       A-----  0        3:300053    50000   0     0.00
58cf98d8   14       A-----  0        3:350053    50000   0     0.00
58cf9968   15       A-----  0        3:400053    50000   0     0.00
58cf9a00   16       A-----  0        3:450053    50000   0     0.00
16 active, 16 total

# 切换当前逻辑日志到 uniqid 7 上逻辑日志文件上
gbasedbt@netsky-PC:~$ onmode -l && onmode -l && onmode -l && onmode -l && onmode -c

# 查看当前逻辑是否位于 uniqid 7 上
gbasedbt@netsky-PC:~$ onstat -l

GBase Database Server Version 12.10.FC4G1AEE -- On-Line -- Up 00:42:56 -- 1716640 Kbytes

Physical Logging
Buffer bufused  bufsize  numpages  numwrits  pages/io
P-2  0        512      579       27        21.44
    phybegin      physize  phypos    phyused   %used
    2:53          500000  66       9         0.00

Logical Logging
Buffer bufused  bufsize  numrecs  numpages  numwrits  recs/pages  pages/io
L-3  0        512     166762   14285    3809      11.7       3.8
    Subsystem    numrecs  Log Space used
    OLDRSAM      166423  23918348
    
```



```

SLOB      5      252
HA        27     1188
DDL       307    106836

address    number  flags   uniqid  begin                size   used   %used
566f0f88   1      U-B---- 1      1:25263              5000  5000  100.00
5917a570   2      U-B---- 2      1:30263              5000  5000  100.00
5917a5d8   3      U-B---- 3      1:35263              5000  4274  85.48
5917a640   4      U-B---- 4      1:40263              5000   1    0.02
5917a6a8   5      U-B---- 5      1:45263              5000   7    0.14
5917a710   6      U-B---- 6      1:50263              5000   1    0.02
58cf95b0   7      U--C-L 7      3:53                 50000  8    0.02
58cf9680   8      A----- 0      3:50053              50000  0    0.00
58cf9618   9      A----- 0      3:100053             50000  0    0.00
58cf96e8  10     A----- 0      3:150053             50000  0    0.00
58cf9758  11     A----- 0      3:200053             50000  0    0.00
58cf97d0  12     A----- 0      3:250053             50000  0    0.00
58cf9850  13     A----- 0      3:300053             50000  0    0.00
58cf98d8  14     A----- 0      3:350053             50000  0    0.00
58cf9968  15     A----- 0      3:400053             50000  0    0.00
58cf9a00  16     A----- 0      3:450053             50000  0    0.00

16 active, 16 total

# 删除 uniqid 1 至 6 的逻辑日志文件
gbasedbt@netsky-PC:~$ for i in {1..6}
> do
>   onparams -d -l $i -y
> done
Logical log 1 successfully dropped.
Logical log 2 successfully dropped.
Logical log 3 successfully dropped.
Logical log 4 successfully dropped.
Logical log 5 successfully dropped.
Logical log 6 successfully dropped.

# 移动逻辑日志文件完成后
gbasedbt@netsky-PC:~$ onstat -l

GBase Database Server Version 12.10.FC4G1AEE -- On-Line -- Up 00:46:48 -- 1716640 Kbytes

Physical Logging
Buffer bufused  bufsize  numpages  numwrits  pages/io
P-2  0      512      625      35      17.86
      phybegin      physize  phypos    phyused   %used

```

```

2:53          500000    91      0      0.00

Logical Logging
Buffer bufused  bufsize  numrecs   numpages  numwrits  recs/pages  pages/io
L-3  0         512     166826   14309    3833      11.7       3.7
  Subsystem    numrecs   Log Space used
  OLDRSAM      166481   23923400
  SBLOB        5        252
  HA           33       1452
  DDL          307     106836

address        number  flags  uniqid  begin                size  used  %used
58cf95b0       7      U---C-L 7      3:53                 50000  26   0.05
58cf9680       8      A----- 0      3:50053              50000  0    0.00
58cf9618       9      A----- 0      3:100053             50000  0    0.00
58cf96e8      10     A----- 0      3:150053             50000  0    0.00
58cf9758      11     A----- 0      3:200053             50000  0    0.00
58cf97d0      12     A----- 0      3:250053             50000  0    0.00
58cf9850      13     A----- 0      3:300053             50000  0    0.00
58cf98d8      14     A----- 0      3:350053             50000  0    0.00
58cf9968      15     A----- 0      3:400053             50000  0    0.00
58cf9a00      16     A----- 0      3:450053             50000  0    0.00
10 active, 10 total
    
```

3.5.5. 创建数据库内部用户

创建数据库内部用户，需要创建内部用户使用的映射目录，可创建一个默认用户（default user），使其包含默认的配置信息。

创建映射目录

```

gbasedbt@netsky-PC:~$ mkdir -p $HOME/users
gbasedbt@netsky-PC:~$ chmod 777 $HOME/users
    
```

创建默认用户，创建一个数据库用户 dbtuser

```

gbasedbt@netsky-PC:~$ dbaccess sysuser -
> CREATE DEFAULT USER WITH PROPERTIES USER daemon HOME '/home/gbase/users';
> CREATE USER dbtuser WITH PASSWORD 'GBase123';
    
```

注：使用 **Ctrl + c** 或者 **Ctrl + ** 退出 dbaccess 交互模式

测试内部用户登陆，交互工输入密码

```

gbasedbt@netsky-PC:~$ dbaccess - -
    
```

```
> CONNECT TO 'testdb@gbase01' USER 'dbtuser';
```

3.6. 数据库参数优化与调整

3.6.1. 配置参数优化

数据库性能优化需要根据业务使用情况进行调整，通过尽可能少的磁盘访问获得所需要的数据。

数据库配置文件 ONCONFIG 包含了常用、默认的参数，根据实际硬件资源及业务实际访问情况，优化与调整部分参数。以下是常用的需要调整的参数。

参数名称	修改后的参数值	参数说明
PHYSBUFF	2048	物理日志缓冲区大小，建议
LOGBUFF	2048	逻辑日志缓冲区大小，建议
DBSPACETEMP	tempdbs01	临时数据库空间名称，可多个
SBSPACE	sbspace01	智能大对象默认空间，仅一个
SYSSBSPACE	sbspace01	智能大对象空间系统管理空间
NETTYPE	soctcp, 2, 200, NET	网络连接池配置
NUMFDSERVERS	4	网络线程池连接交换服务数量
MULTIPROCESSOR	1	开启使用多 CPU
VPCLASS	cpu, num=80, aff=(40-79, 120-159), noage	指定 CPU VP 的数量，建议 CPU 数-1 若是 NUMA 架构，最多使用 2 个 NODE
AUTO_TUNE	0	关闭系统部分参数的自动调整
AUTO_REPREPARE	1	自动重新预处理
CLEANERS	128	页清理线程的数量
LOCKS	100000000	指定初始锁的数量 根据硬件和业务需求调整
DEF_TABLE_LOCKMODE	row	指定默认的表锁模式
SHMVIRTSIZE	40960000	指定虚拟内存段大小，单位是 KB 根据硬件调整
SHMADD	10240000	指定虚拟内存段扩展大小 根据硬件调整
SHMTOTAL	0	指定数据库使用的最大内存量，可指定至操作系统内存的 80-90%
STACKSIZE	2048	指定堆栈大小
ALLOW_NEWLINE	1	允许字符类字段内换行
USELASTCOMMITTED	NONE	指定使用最后提交读
DS_MAX_QUERIES	5	同时支持的决策并行查询数量
DS_TOTAL_MEMORY	40960000	决策并行查询使用的内存大小

		根据硬件和业务需求调整
DS_NONPDQ_QUERY_MEM	256000	非决策并行查询使用的内存大小，最大为 DS_TOTAL_MEMORY 的 1/4
TEMPTAB_NOLOG	1	指定临时表使用无日志方式
DUMPSHMEM	0	指定断言失败时，不 DUMP 内存
BUFFERPOOL	size=2k, buffers=1000000, lrus=128, lru_min_dirty=50, lru_max_dirty=60	指定 2K 页面使用的缓冲池 根据硬件和业务需求调整
BUFFERPOOL	size=16k, buffers=5000000, lrus=128, lru_min_dirty=50, lru_max_dirty=60	指定 16K 页面使用的缓冲池 根据硬件和业务需求调整

3.6.2. 关闭 sysadmin 库运行

如非必要，可以关闭 sysadmin 库运行。

使用 gbasedbt 用户登陆，在 \$GBASEDBTDIR/etc/sysadmin 目录下创建一个普通文件 stop，数据库重启后，将不会再运行 sysadmin 中指定的任务。

```

gbasedbt@netsky-PC:~$ cd $GBASEDBTDIR/etc/sysadmin

gbasedbt@netsky-PC:/opt/gbase/etc/sysadmin$ touch stop

gbasedbt@netsky-PC:/opt/gbase/etc/sysadmin$ onmode -ky

gbasedbt@netsky-PC:/opt/gbase/etc/sysadmin$ oninit -vy

gbasedbt@netsky-PC:/opt/gbase/etc/sysadmin$ onstat -g sql

GBase Database Server Version 12.10.FC4G1AEE -- On-Line -- Up 00:00:15 -- 1716640 Kbytes

```

重启数据库后，sysadmin 库将不再运行。

3.6.3. 数据库空间自动扩展（chunk 自动扩展）

使用文件系统时，如果有空间自动扩展需求，需开启 chunk 自动扩展功能。

使用 gbasedbt 用户登陆，在 sysadmin 库中执行管理函数 task 或者 admin，设置相应的 chunk 为自动扩展。

```

gbasedbt@netsky-PC:~$ dbaccess sysadmin -
> execute function task('modify chunk extendable on',6);

```

其中：数字 6 为对应的 chunk 编号。

3.7. 数据库加入自启动服务

数据库随操作系统启动而启动功能，需要使用操作系统的 systemd 功能。

警告：随操作系统启动仅限于单机环境使用。

以 RHEL7/CentOS7 为例：

在/usr/lib/systemd/system/目录下创建服务配置文件 gbasedbtd.service，内容如下：

```
[Unit]
Description=GBase 8s Database Server
Documentation=file:/opt/gbase/release/en_us/0333
Wants=network-online.target
After=network.target network-online.target

[Service]
Type=oneshot
User=gbasedbt
Group=gbasedbt
RemainAfterExit=yes
EnvironmentFile=/etc/sysconfig/gbasedbtd_service
ExecStart=/opt/gbase/bin/oninit
ExecStop=/opt/gbase/bin/onmode -ky
#Restart=on-abort

[Install]
WantedBy=multi-user.target
```

在/etc/sysconfig 目录下创建环境文件 gbasedbtd_service，内容应与.bash_profile 内容一致，参考如下：

```
# gbasedbtd.service environmental variables
GBASEBTDIR=/opt/gbase
GBASEBTDSEVER=gbase01
ONCONFIG=onconfig.gbase01
PATH=$GBASEBTDIR/bin:${PATH}
GBASEBTDSQLHOSTS=/opt/gbase/etc/sqlhosts
DB_LOCALE=zh_CN.utf8
CLIENT_LOCALE=zh_CN.utf8
GL_USEGLU=1
DBDATE="Y4MD"
```

启动 gbasedbtd 服务（数据库未启动时）

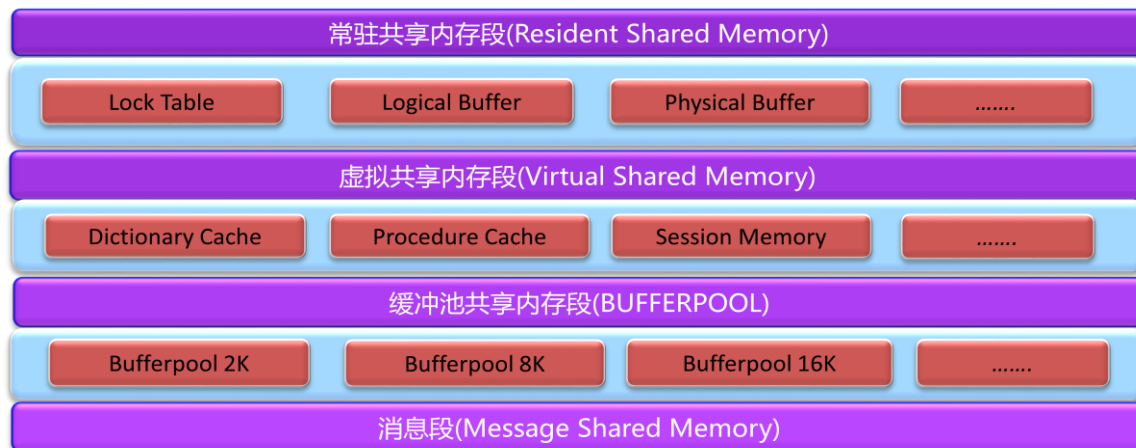
```
systemctl start gbasedbtd
```

设置 gbasedbtd 服务随系统启动

```
systemctl enable gbasedbtd
```

4. 附录：GBase 8s 数据库内存管理与配置

GBase 8s 数据库共享内存由常驻段、缓冲池、虚拟段和消息段四部分组成。如下图所示：



常驻段 (resident segment)

共享内存常驻段包括以下部分：

共享内存头、逻辑日志缓冲区、物理日志缓冲区、高可用数据复制缓冲区、锁表

缓冲池 (bufferpool)

缓冲池是共享内存段中的最大组成部分，用于缓存磁盘数据，降低 I/O 开销，提高数据库运行效率。缓冲池的大小由数据库配置参数 BUFFERPOOL 指定。

虚拟段 (virtual segment)

数据库所有的会话都有一个或多个内存池。当数据库服务器需要内存时，它会先查看指定的池。如果池中可用的内存不足以满足请求，那么数据库服务器将从系统池添加内存。如果数据库服务器不能在系统池中找到足够的内存，那么它会动态地给虚拟部分分配更多的段。

共享内存的虚拟部分存储以下数据：

内部表、大缓冲区、会话数据、线程数据（堆栈和堆）、数据分布高速缓存、字典高速缓存、SPL 例程高速缓存、SQL 字典高速缓存、排序池、全局池。

消息段 (message segment)

共享内存消息段用于使用 IPC 共享内存连接的应用与数据库之间的通信。

共享内存的通信部分的大小约等于 12 千字节乘以共享内存通信所需的期望连接数 (nettype ipcshm)。

4.1. 数据库内存配置

数据库使用的内存总量是常驻段、虚拟段、消息段和缓冲池四部分内存分配的总和。

4.1.1. 常驻段内存使用量计算

常驻段包括锁表、逻辑日志缓冲区、物理日志缓冲区、共享内存头等信息，常驻段的内存使用量可以使用以下方法计算：

初始化锁内存量 = ONCONFIG 配置文件中的 LOCKS * 136

逻辑日志缓冲区大小 = ONCONFIG 配置文件中的 LOGBUFF * 1024 * 3

物理日志缓冲区大小 = ONCONFIG 配置文件中的 PHYSBUFF * 1024 * 2

常驻段内存大小 = (初始化锁内存量 + 逻辑日志缓冲区大小 + 物理日志缓冲区大小 + 1,200,000) * 1.02 字节

注：计算均不是精确计算，原因是常驻段的内存结构类型比较多，并不能完全举例。

4.1.2. 缓冲池内存使用量计算

缓冲池比较简单，即 ONCONFIG 配置参数中指定的 BUFFERPOOL 的总大小。如：ONCONFIG 配置参数中配置了两个 BUFFERPOOL，需要将两者相加。

```
BUFFERPOOL size=2k, buffers=200000, lrus=8, lru_min_dirty=50, lru_max_dirty=60  
BUFFERPOOL size=16k, buffers=200000, lrus=32, lru_min_dirty=50, lru_max_dirty=60
```

每个缓冲池大小 = size * buffers * 1024

缓冲池总大小 = 2K 缓冲池大小 + 16K 缓冲池大小

注：计算均不是精确计算，原因是维护缓冲池的结构同样需要内存，其内存用量与操作系统相关。

4.1.3. 虚拟段内存使用量计算

初始化的虚拟段的内存大小为 ONCONFIG 配置参数中指定的 SHMVIRT SIZE, 当数据库内存不足时, 会向操作系统申请新的内存段, 大小为 ONCONFIG 配置参数中指定的 SHMADD。

若申请的内存段次数多的时候, 实际分配的内存段大小可能为 SHMADD 的倍数大小。若申请内存时, 数据库总内存使用量达到 ONCONFIG 中的 SHMTOTAL 限制, 申请内存操作将会失败。

初始虚拟内存段大小 = ONCONFIG 配置文件中的 SHMVIRT SIZE * 1024

4.1.4. 消息段内存使用时计算

消息段内存仅处理共享内存通信连接。其内存使用量估算为:

消息段内存大小 = ONCONFIG 配置参数 NETTYPE 中 ipcshm 的配置连接数 * 12 * 1024

注: 消息段现在基本上不再配置。

4.1.5. 数据库内存段计算

数据库 ONCONFIG 配置参数涉及的内存的列表如下:

参数名称	参数值
LOGBUFF	1024
PHYSBUFF	1024
LOCKS	2000000
NETTYPE	soctcp,2,200,CPU
SHMVIRT SIZE	4096000
SHMADD	1024000
BUFFERPOOL	size=2K,buffers=200000,lrus=8,lru_min_dirty=50,lru_max_dirty=60
BUFFERPOOL	size=16K,buffers=200000,lrus=32,lru_min_dirty=50,lru_max_dirty=60

初始化锁内存量 = 2,000,000 * 136 = 272,000,000

逻辑日志缓冲区大小 = 1024 * 1024 * 3 = 3145728

物理日志缓冲区大小 = 1024 * 1024 * 2 = 2097152

常驻段(R) 内存大小 = (272,000,000 + 3145728 + 2097152 + 1,200,000) * 1.02 = 284011737.6

缓冲池(B) 2K = 2 * 200,000 * 1024 = 409,600,000

缓冲池(B) 16K = 16 * 200,000 * 1024 = 3,276,800,000

虚拟段(V) 内存大小 = 4096000 * 1024 = 4,194,304,000

数据库使用的总内存大小为：284011737.6 + 409,600,000 + 3,276,800,000 + 4,194,304,000 = 8164715737.6，约与 onstat -g seg 输出的 size 值 8256380928 相匹配。

```
[gbasedbt@PC ~]$ onstat -g seg
GBase Database Server Version 12.10.FC4G1TL -- On-Line -- Up 6 days 22:32:02 -- 8062872 Kbytes

Segment Summary:
id      key      addr      size      ovhd      class  blkused  blkfree
2260992 52564801 44000000 290177024 3833912   R      70844    0
2293761 52564802 554bc000 4194304000 49153848  V      98038    925962
2326530 52564803 14f4bc000 452349952 1         B      110437   0
2359299 52564804 16a421000 3319549952 1         B      810437   0
Total:  -      -      -      8256380928 -         -      1089756  925962

(* segment locked in memory)
No reserve memory is allocated
```

总内存使用量

各内存段内存使用量 内存段类型

注：onstat -g seg 的输出中，size 表示内存大小，class 标识中

R：常驻段

V：虚拟段

B：缓冲池（每个页大小的缓冲池对应一个）

M：消息段

4.2. 数据库安装常见问题

4.2.1. 安装问题 1: The version file is not exists

描述：执行 ids_install 安装时，报“The version file is not exist.”

解析：操作系统缺少必要的 unzip 工具，需要安装 unzip。CentOS 系统中可通过 yum install unzip 安装。

如果安装目录下的隐藏文件 .gbase.properties 缺失，也会出现此问题。重新解压缩安装包后，再次安装。

4.2.2. 安装问题 2: 环境变量 PATH 中无 JVM

描述: 执行 `ids_install` 安装时, 报“No Java virtual machine could be found from your PATH environment variable. You must install a VM prior to running this program.”

解析: 提示信息为无 Java 虚拟机, 按照提示安装 JRE/JDK。支持的 JRE/JDK 版本为 1.6 及以上, 如 CentOS 自带的 `java-1.8.0-openjdk`。

4.2.3. 安装问题 3: 安装目录不存在

描述: 执行 `ids_install` 安装时, 报“The parent directory of the user install path is not exists, please choose another one.”

解析: 提示信息为指定的目录上级目录不存在。默认的安装目录为: `/opt/GBASE/gbase`, 如果 `/opt/GBASE` 目录不存在, 则会报这个错误。需要在安装前创建 `/opt/GBASE` 目录。

4.2.4. 安装问题 4: 安装目录非空

描述: 执行 `ids_install` 安装时, 报“User install directory is not empty, please choose another one.”

解析: 提示信息为指定的目录非空。安装数据库需要在空目录或者不存在的目录下 (上级目录需存在, 系统会自动创建目录), 手工指定其它非空目录或者不存在目录 (上级目录需存在, 系统会自动创建目录)。

4.2.5. 初始化问题 5: 无法分配内存

描述: 执行数据库初始化或者启动数据库时, 报

```
Allocating and attaching to shared memory... FAILED
WARNING: server initialization failed or timed out.
Check the message log, online.log, for errors.
```

解析: 出现问题, 根据提示要求查看 `online.log`, 使用 `onstat -m` 命令可查看最近 20 行 `online.log` 日志信息。如果日志里包含类似以下信息

```
19:03:42 shmget: EEXIST: key 52564801: shared memory already exists
19:03:42 mt_shm_init: can't create resident segment.
```

表示数据库的内存段已经被占用，可能的情况是：

- 1) 当前实例已经启动，即通过 `onstat -命令` 检查状态，如果已经是 `On-Line` 状态，即

```
GBase Database Server Version 12.10.FC4G1AEE -- On-Line -- Up 00:07:02 -- 208516
Kbytes
```

表示数据库不需要再初始化或者重新启动，保持现有状态即可。

- 2) 当前实例未启动，即通过 `onstat -命令` 检查状态，如果已经是未启动状态，即

```
shared memory not initialized for GBASEDBTSERVER 'ol_gbasedbt1210'
```

则表示之前的数据库关闭或者异常时，内存未能及时释放，执行 `onclean -ky` 命令强制清理内存，然后通过 `ipcs -m` 命令查看 `key` 中与报错信息中一致的 `0x52564801` 是否存在，如果不存在，可以重新启动数据库实例；如果存在，使用 `root` 用户执行 `ipcrm -M 0x52564801` 清理共享内存，同时清理 `0x525648XX` 开头各个内存段，完成后可以重新启动数据库实例。

4.2.6. 初始化问题 6：初始化内存失败，无法打开主 chunk

描述： 执行数据库初始化或者启动数据库时，报

```
Opening primary chunks...FAILED
oninit: Fatal error in shared memory initialization
WARNING: server initialization failed or timed out.
Check the message log, online.log, for errors.
```

解析： 出现问题，根据提示要求查看 `online.log`，使用 `onstat -m` 命令可查看最近 20 行 `online.log` 日志信息。如果日志里包含类似以下信息：

- 1) “19:07:08 The chunk '/opt/gbase/storage/rootdbs' must have READ/WRITE permissions for owner and group (660).”

表示数据库使用的 `chunk` 对应的文件或者文件系统权限不对，按提示要求修改为属主及属组均为 `gbase`，权限为 `660`（即：`chown gbase:gbase /opt/gbase/storage/rootdbs, chmod 660 /opt/gbase/storage/rootdbs`），完成后重新启动数据库。

- 2) “19:10:51 The chunk '/opt/gbase/storage/rootdbs' must have READ/WRITE permissions for owner (600).”

数据库使用的 `chunk` 对应的文件或者文件系统权限应为属主及属组均为 `gbase`，

权限为 660，这里提示需要 600 权限。说明 oninit 可执行文件的属主及权限不对，或者是非 root 用户安装的数据库。我们要求使用 root 用户来安装数据库。故在 \$GBASEDBTDIR 目录下，使用 root 用户执行 RUNasroot.installserver 脚本以完成对所有已安装的文件权限修正。同时修改 chunk 文件或者文件系统的属主及属组均为 gbasbedbt，权限为 660（即：`chown gbasbedbt:gbasedbt /opt/gbase/storage/rootdbs , chmod 660 /opt/gbase/storage/rootdbs`），完成后重新初始化实例或者重启实例。

4.2.7. 连接问题 7：无法连接到数据库服务器（908）

描述：连接数据库时，报“908: Attempt to connect to database server (ol_gbasedbt1210) failed.”

解析：根据提示信息，表示当前无法连接到指定的数据库。对于此问题，需要从最底层往上排查。

首先，检查数据库的侦听是否正常，命令 `onstat -g ntt` 的输出结果中，`thread name` 为 `soctcplst` 对应的 `address` 中应包括 主机名或 IP 地址|服务名称或者端口号|`soctcp`（如 `localhost.localdomain|ol_gbasedbt1210|soctcp`），该信息来源地 `sqlhosts` 配置文件（建议在 `sqlhosts` 配置文件中指定 IP 地址及端口，而不是使用主机名或者服务名称）。如果不存在符合条件的 `soctcplst`，则需要修改 `sqlhosts` 为正确配置，并重启数据库。如果存在 `soctcplst`，但 IP 是本地 loop 地址（127.0.0.1），也需修改为实际物理 IP，并重启数据库。

如果侦听正常，则继续检查操作系统防火墙设置，请关闭防火墙或者放通数据库使用的端口。

4.2.8. 连接问题 8：连接参数

描述：GBase DataStudio 连接参数配置

JDBC URL: jdbc:gbasedbt-sqli://bd.gbasedbt.com:9088/sysmaster:INFORMIXS

主机: bd.gbasedbt.com 端口: 9088

实例名: ol_gbasedbt1210

数据库/模式: sysmaster

用户名: gbasedbt

密码: ●●●●●●●●

CM连接

CM文件: 打开

解析:

主机 -> 填写能访问到数据库服务器的 IP 地址或者域名(能通过 ssh 客户端访问)

端口 -> sqlhosts 配置文件中的端口号(通过 onstat -g ntt 可看到)

实例名 -> \$GBASEDBTSERVER 配置参数

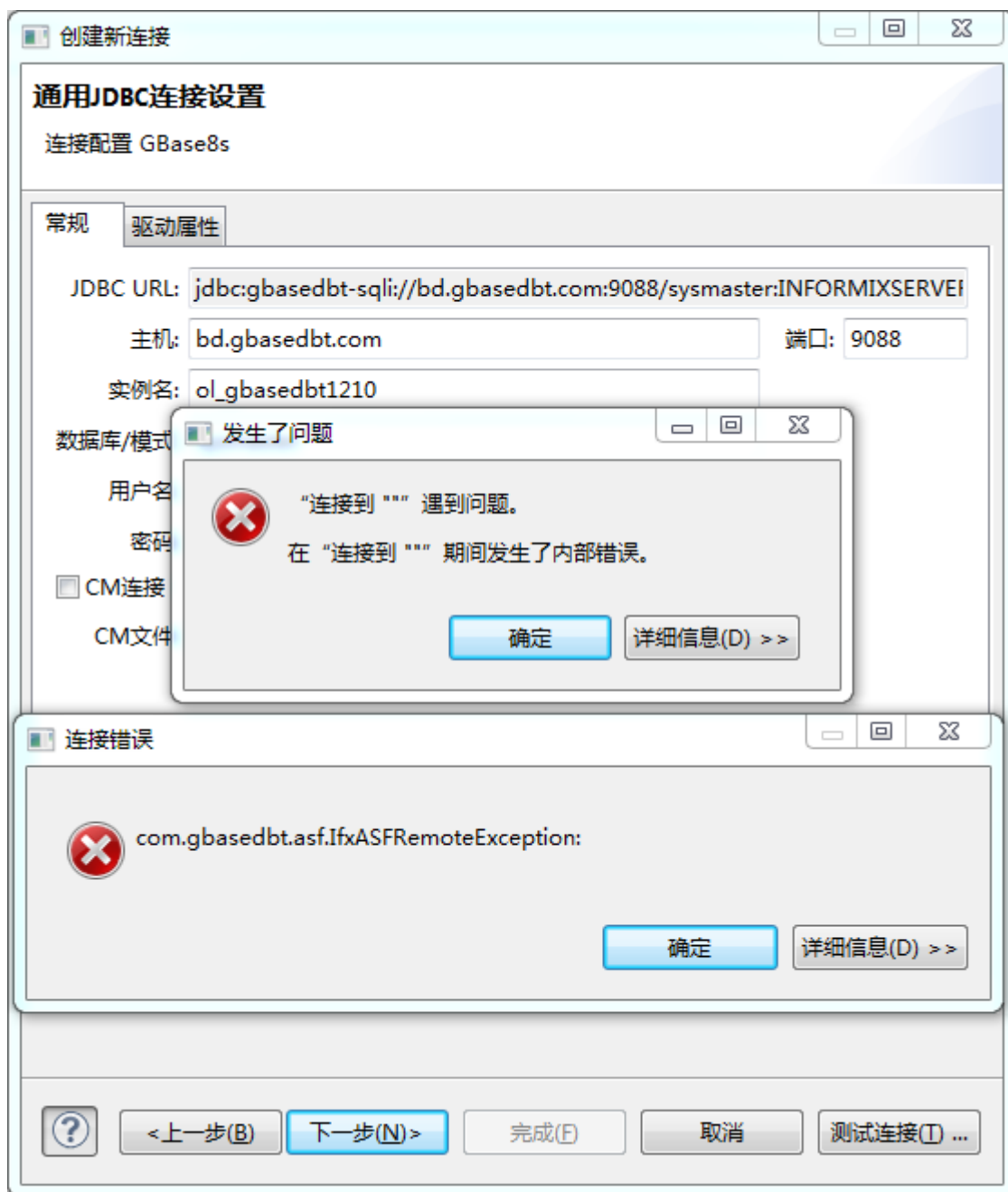
数据库/模式 -> create database 语句创建的库。可使用系统库 sysmaster

用户名 -> 操作系统用户, 默认的系统用户是 gbasedbt

密码 -> 用户名在操作系统上的密码(注意密码强度要求)

4.2.9. 连接问题 9: 工具连接报错: 内部错误

描述: GBase DataStudio 连接数据库报连接错误, 如下图:



解析：该问题是指定的用户密码不符合密码强度要求（GBase 8s v8 2.0.1a2_2 版本有密码强度要求，需要密码不少于 8 位且需包含大小写字母数字），在数据库服务器上执行 `onstat -m` 能看到“09:17:03 listener-thread: err = -961: oserr = 0: errstr = : The password is too simple.”信息。

修改用户密码以符合密码强度要求（最长可能需要 15 分钟生效）。

4.2.10. 安装问题 10：无法执行二进制文件

描述：执行 `ids_install` 安装时，报

```
./ids_install: line 3319: /tmp/install.dir.15072/Linux/resource/jre/jre/bin/java:
cannot execute binary file
./ids_install: line 3319: /tmp/install.dir.15072/Linux/resource/jre/jre/bin/java:
Success
```

解析：提示信息是数据库安装包里自带的 java 命令是不能执行的二进制文件，原因是：当前的平台与数据库安装包使用的平台不同（如 x86 平台上使用了其它平台（飞腾、龙芯、Power 等）的安装包）。X86 平台对应的文件名后缀是 x86_64.tar，飞腾平台对应的文件名后缀是 FT.tar，龙芯平台对应的文件名后缀是 Loongson3B.tar。

4.2.11. 安装问题 11：无法打开共享对象文件 libnsl.so.1

描述：安装时报

```
libjvm.so
preloadLibrary(/tmp/install.dir.33990/Linux/resource/jre/jre/lib/amd64/libjava.so): libnsl.so.1: 无法打开共享对象文件：没有那个文件或目录
```

解析：安装 libnsl

```
yum install libnsl -y
```

注：如果无法通过 yum/apt 源安装，可在/usr/lib64 目录（不同系统该目录可能不同），做 libnsl.so.1 的链接至 libnsl.so.2。

4.2.12. 安装问题 12：一个或多个系统依赖包未安装

描述：安装时报

```
One or more prerequisite system libraries are not installed on your computer. Install
libncurses.so.5 and then restart the installation program.
```

解析：

1) 安装 ncurses-devel

```
yum install ncurses-devel
```

2) 在/usr/lib64 目录下创建链接文件，执行

```
cd /usr/lib64
ln -s libncurses.so.6.1 libncurses.so.5
ln -s libtinfo.so.6.1 libtinfo.so.5
```

3) 在完成 2)后仍然报错

则需要加载 LD_LIBRARY_PATH 环境变量

```
export LD_LIBRARY_PATH=/usr/lib64:$LD_LIBRARY_PATH
```