《数据库》

openGauss数据库日志管理指导手册



华为技术有限公司

目录

[前 言 2](#_Toc51332081)

[简介 2](#_Toc51332082)

[内容描述 2](#_Toc51332083)

[前置条件 2](#_Toc51332084)

[实验环境说明 2](#_Toc51332085)

[实验概览 3](#_Toc51332086)

[1 日志管理 4](#_Toc51332087)

[1.1 实验介绍 4](#_Toc51332088)

[1.1.1 关于本实验 4](#_Toc51332089)

[1.1.2 实验目的 4](#_Toc51332090)

[1.1.3 系统日志 4](#_Toc51332091)

[1.1.4 操作日志 7](#_Toc51332092)

[1.1.5 审计日志 8](#_Toc51332093)

[1.1.6 WAL日志 12](#_Toc51332094)

[1.1.7 性能日志 13](#_Toc51332095)

[2 附录一：Linux操作系统相关命令 15](#_Toc51332096)

[3 附录二：openGauss数据库基本操作 25](#_Toc51332111)

# 前 言

## 简介

本指导书适用于学习openGauss数据库日志相关内容和管理方法，通过该指导书可以了解openGauss数据库中日志管理的内容，并对数据库进行日常管理维护、问题定位和数据库恢复的操作。

## 内容描述

本实验指导书主要内容为openGauss数据库中日志管理的内容，并对数据库进行日常管理维护、问题定位和数据库恢复的操作。

## 前置条件

由于本实验主要是在openEuler操作系统上进行openGauss数据库日志的管理，需要掌握Linux系统的基本操作和系统命令，详细请参见**附录一**。

## 实验环境说明

* 组网说明

本实验环境为openGauss数据库管理系统，安装在华为云openEuler弹性服务器ECS上。

* 设备介绍

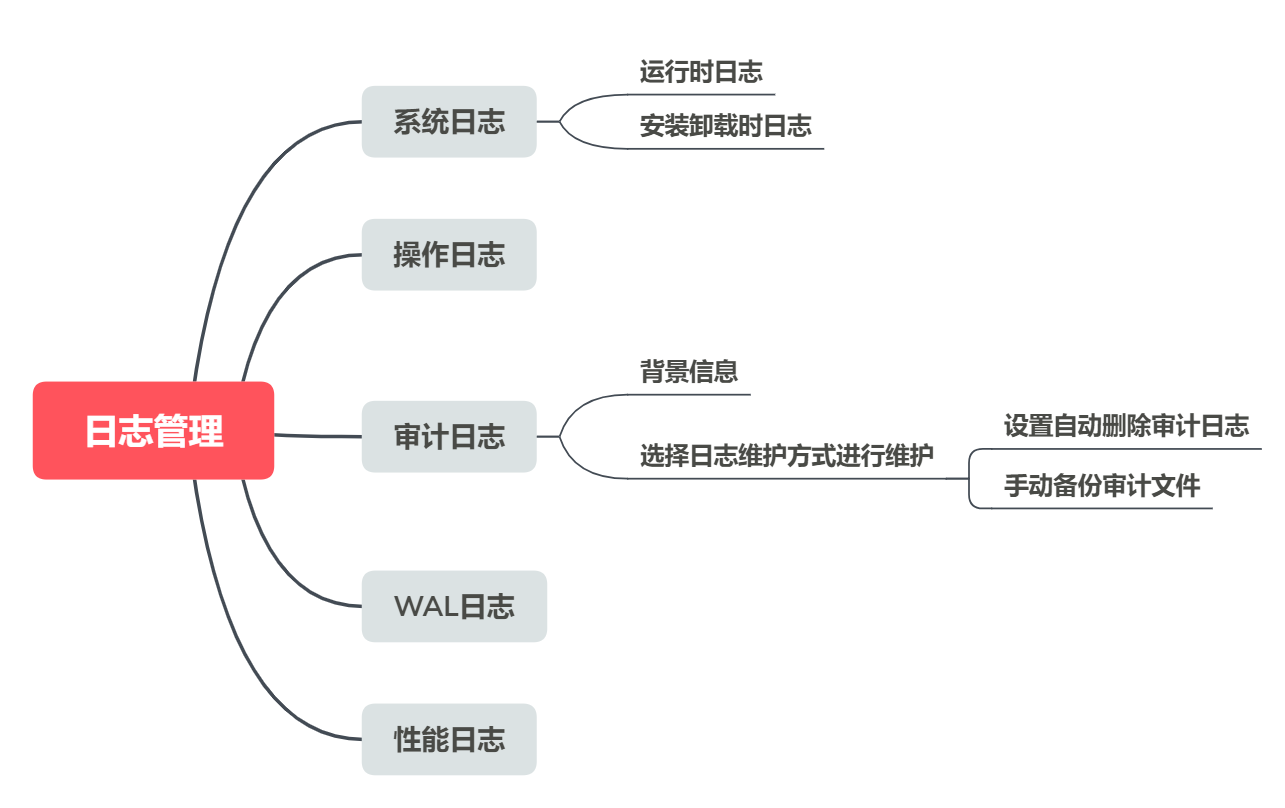
为了满足数据库原理与实践课程实验需要，建议每套实验环境软件版本采用以下配置：

设备名称、型号与版本的对应关系如下：

设备明细表

|  |  |  |
| --- | --- | --- |
| 设备名称 | 设备型号 | 软件版本 |
| 数据库 | openGauss | openGauss 1.1.0 |
| 操作系统 | openEuler | openEuler 20.3LTS |

## 实验概览



本实验概览图

# 日志管理

## 实验介绍

### 关于本实验

在实际的数据库管理工作中，小数据量的日志可以按照实验手册，使用cat查看。但需要注意的是，生产环境的日志量可能比较多，日志文件容量经常在GB级别左右，为了提升读取日志的有效性，一般建议使用more、tail、grep等命令查看日志，所以学会这些命令的基本使用方法也是必须的。另外，在生产环境节点较多（成千上万个生产节点）、日志类型较复杂的情况下，人工已无法满足实际生产需求，针对这种情况，一般使用脚本自动化分析或使用第三方软件对日志进行实时监控、分析、告警。

本实验主要描述openGauss数据库中日志管理的内容，并对数据库进行日常管理维护、问题定位和数据库恢复的操作。

### 实验目的

掌握openGauss数据库中日志管理的内容；

掌握对数据库进行日常管理维护、问题定位。

### 系统日志

openGauss运行时数据库节点以及openGauss安装部署时产生的日志统称为系统日志。如果openGauss在运行时发生故障，可以通过这些系统日志及时定位故障发生的原因，根据日志内容制定恢复openGauss的方法。

说明：

日志路径在安装openGauss时已由XML文件中gaussdbLogPath参数指定，如果未指定该参数值，则默认路径在/var/log/gaussdb。

数据文件路径在安装openGauss时已由XML文件中dataNode参数指定，如果未指定该参数值，则默认路径在/gaussdb/data/data\_dn。

#### 运行时日志

切换到omm用户，以操作系统用户omm登录数据库主节点。

**su - omm**

数据库节点的运行日志放在“$gaussdbLogPath/用户名/pg\_log /用户名/pg\_log”中，当前场景下用户名为”omm”，切换到pg\_log文件夹，并显示文件夹中的内容。

**cd /var/log/gaussdb/omm/pg\_log** ***##实际路径以$gaussdbLogPath为准***

**ls**

文件内容显示如下:

dn\_6001

数据库节点文件夹为”dn\_6001”,以自己实际数据库节点名先为准。

切换到“pg\_log”文件夹下的数据库节点文件夹中，查看日志文件。

**cd dn\_6001**

**ls**

日志文件列表如下：

postgresql-2020-07-21\_170715.log postgresql-2020-07-23\_145207.log

postgresql-2020-07-21\_174440.log postgresql-2020-07-23\_160709.log

postgresql-2020-07-22\_102526.log postgresql-2020-07-23\_162357.log

postgresql-2020-07-23\_140520.log postgresql-2020-07-24\_103226.log

postgresql-2020-07-23\_142946.log postgresql-2020-07-25\_000000.log

由于在运行过程中日志的大小可能会超过16 MB，所以在简单查询的过程中，可以先查看日志的大小（其中l是L的小写）。

**ll -h**

列表显示如下：

total 56M

-rw------- 1 omm dbgrp 56K Jul 21 17:38 postgresql-2020-07-21\_170715.log

-rw------- 1 omm dbgrp 3.5M Jul 21 20:51 postgresql-2020-07-21\_174440.log

-rw------- 1 omm dbgrp 12M Jul 22 21:10 postgresql-2020-07-22\_102526.log

-rw------- 1 omm dbgrp 46K Jul 23 14:06 postgresql-2020-07-23\_140520.log

-rw------- 1 omm dbgrp 83K Jul 23 14:32 postgresql-2020-07-23\_142946.log

-rw------- 1 omm dbgrp 1.1M Jul 23 15:45 postgresql-2020-07-23\_145207.log

-rw------- 1 omm dbgrp 373K Jul 23 16:23 postgresql-2020-07-23\_160709.log

-rw------- 1 omm dbgrp 5.3M Jul 23 21:03 postgresql-2020-07-23\_162357.log

-rw------- 1 omm dbgrp 15M Jul 24 23:59 postgresql-2020-07-24\_103226.log

-rw------- 1 omm dbgrp 19M Jul 25 16:52 postgresql-2020-07-25\_000000.log

可以查看内容较少的日志。

**cat postgresql-2020-07-23\_142946.log**

日志内容为：

2020-07-23 14:29:46.457 5f192e5a.1 [unknown] 281466519832976 [unknown] 0 dn\_6001 00000 0 [BACKEND] LOG: start create thread!

2020-07-23 14:29:46.457 5f192e5a.1 [unknown] 281466519832976 [unknown] 0 dn\_6001 00000 0 [BACKEND] LOG: create thread end!

2020-07-23 14:29:46.458 5f192e5a.10000 [unknown] 281466446432848 dn\_6001 0 dn\_6001 00000 0 [BACKEND] LOG: reaper backend started.

2020-07-23 14:29:46.458 5f192e5a.10000 [unknown] 281466463275600 dn\_6001 0 dn\_6001 00000 0 [BACKEND] LOG: [Alarm Module]alarm checker started.

2020-07-23 14:29:46.459 [MOT] <TID:21137/-----> <SID:-----/-----> [INFO] <JitExec> Using native LLVM version 7.0.0

2020-07-23 14:29:46.459 [MOT] <TID:21137/-----> <SID:-----/-----> [INFO] <Configuration> Configuring total memory for relative memory values to: 2048 MB

2020-07-23 14:29:46.459 [MOT] <TID:21137/-----> <SID:-----/-----> [INFO] <System> Startup: Loading configuration from /gaussdb/data/db1/mot.conf

2020-07-23 14:29:46.459 [MOT] <TID:21137/-----> <SID:-----/-----> [INFO] <Configuration> Loaded max\_mot\_global\_memory: 80% from total = 1638 MB

2020-07-23 14:29:46.459 [MOT] <TID:21137/-----> <SID:-----/-----> [INFO] <Configuration> Loaded max\_mot\_local\_memory: 15% from total = 307 MB

2020-07-23 14:29:46.459 [MOT] <TID:21137/-----> <SID:-----/-----> [INFO] <Configuration> Loading max\_connections from envelope into MOTEngine: 5000

2020-07-23 14:29:46.459 [MOT] <TID:21137/-----> <SID:-----/-----> [INFO] <Configuration> Configuring asynchronous redo-log handler due to synchronous\_commit=off

2020-07-23 14:29:46.460 [MOT] <TID:21137/-----> <SID:-----/-----> [WARNING] <Configuration> Adjusting MOT memory limits: global = 102 MB, local = 26 MB, session large store = 0 MB, total = 128 MB

2020-07-23 14:29:46.460 [MOT] <TID:21137/-----> <SID:-----/-----> [INFO] <Memory> Global Memory Limit is configured to: 0 MB --> 102 MB

2020-07-23 14:29:46.460 [MOT] <TID:21137/-----> <SID:-----/-----> [INFO] <Memory> Local Memory Limit is configured to: 0 MB --> 26 MB

2020-07-23 14:29:46.460 [MOT] <TID:21137/-----> <SID:-----/-----> [INFO] <Memory> Session Memory Limit is configured to: 0 KB --> 0 KB (maximum 5000 sessions)

2020-07-23 14:29:46.460 [MOT] <TID:21137/-----> <SID:-----/-----> [INFO] <Memory> Configured automatic chunk allocation policy to 'LOCAL' on single node machine

#### 安装卸载时日志

切换到omm用户，以操作系统用户omm登录数据库主节点。

**su - omm**

openGauss安装卸载时产生的日志放在“$gaussdbLogPath/用户名/om”目录下，当前场景下用户名为”omm”，切换到om文件夹，并显示文件夹中的内容。

**cd /var/log/gaussdb/omm/om *##实际路径以$gaussdbLogPath为准***

**ll -h**

文件夹中内容显示如下：

total 160K

-rw------- 1 omm dbgrp 9.1K Jul 21 17:07 **gs\_install-2020-07-21\_170455.log**

-rw------- 1 omm dbgrp 68K Jul 24 10:32 gs\_local-2020-07-21\_160847.log

-rw------- 1 omm dbgrp 35K Jul 24 10:32 gs\_om-2020-07-21\_174438.log

-rw------- 1 omm dbgrp 32K Jul 21 16:36 gs\_preinstall-2020-07-21\_160843.log

查看“gs\_install-2020-07-21\_170455.log”日志，了解openGauss安装情况。

**cat gs\_install-2020-07-21\_170455.log**

日志内容如下：

[2020-07-21 17:04:55.967536][27138][gs\_install][DEBUG]:The /opt/huawei/wisequery/omm\_mppdb/install\_step/install\_step.dat does not exits.

[2020-07-21 17:04:55.967641][27138][gs\_install][DEBUG]:gs\_install execution takes 7 steps in total

[2020-07-21 17:04:55.967709][27138][gs\_install][LOG][Step1]:Parsing the configuration file.

[2020-07-21 17:04:55.970059][27138][gs\_install][DEBUG]:Instance information of cluster:

ClusterName=dbCluster,AppPath=/opt/gaussdb/app,LogPath=/var/log/gaussdb,ClusterType=single-inst

HostName=db1,backIps=['192.168.0.58']

InstanceId=10001,MirrorId=-3,Host=db1,Port=0,DataDir=cm\_agent,XlogDir=,SsdDir=,InstanceType=-1,Role=5,ListenIps=['192.168.0.58'],HaIps=[]

InstanceId=6001,MirrorId=1,Host=db1,Port=26000,DataDir=/gaussdb/data/db1,XlogDir=,SsdDir=,InstanceType=0,Role=4,ListenIps=['192.168.0.58'],HaIps=['192.168.0.58'],azName=AZ1.

[2020-07-21 17:04:56.022578][27138][gs\_install][DEBUG][Step1]:Successfully parsed the configuration file.

[2020-07-21 17:04:56.067808][27138][gs\_install][LOG][Step2]:Check preinstall on every node.

……

alarm=/opt/huawei/snas/bin/snas\_cm\_cmd --time\_out=300

[2020-07-21 17:07:16.476528][27138][gs\_install][LOG]:Successfully started cluster.

[2020-07-21 17:07:16.476605][27138][gs\_install][DEBUG][Step7]:Successfully started the cluster.

[2020-07-21 17:07:16.487388][27138][gs\_install][LOG]:Successfully installed application.

[2020-07-21 17:07:16.487469][27138][gs\_install][LOG]:end deploy..

### 操作日志

操作日志是指数据库管理员使用工具操作数据库时以及工具被openGauss调用时产生的日志。如果openGauss发生故障，可以通过这些日志信息跟踪用户对数据库进行了哪些操作，重现故障场景。

#### 操作步骤

切换到omm用户，以操作系统用户omm登录数据库主节点。

**su - omm**

默认在“$GAUSSLOG/bin”目录下，其中$GAUSSLOG默认为“/var/log/gaussdb/用户名”， 当前场景下用户名为”omm”。

**cd /var/log/gaussdb/omm/bin**

**ls**

显示如下：

gs\_ctl gs\_guc gs\_initdb gs\_obs

以gs\_guc操作为例，进入gs\_guc文件夹，并查看文件的属性。

**cd gs\_guc**

**ll -h**

文件列表显示如下：

total 20K

-rw------- 1 omm dbgrp 17K Jul 23 17:20 gs\_guc-2020-07-21\_170713-current.log

查看操作日志文件

**cat gs\_guc-2020-07-21\_170713-current.log**

日志显示如下：

[2020-07-21 17:07:13]

expected instance path: [/gaussdb/data/db1/postgresql.conf]

gs\_guc set: ssl=on: [/gaussdb/data/db1/postgresql.conf]

gs\_guc set: ssl\_cert\_file='server.crt': [/gaussdb/data/db1/postgresql.conf]

gs\_guc set: ssl\_key\_file='server.key': [/gaussdb/data/db1/postgresql.conf]

gs\_guc set: ssl\_ca\_file='cacert.pem': [/gaussdb/data/db1/postgresql.conf]

Total instances: 1. Failed instances: 0.

Success to perform gs\_guc!

……

[2020-07-23 17:20:20]

Begin to perform gs\_guc for all datanodes.

[2020-07-23 17:20:20]

[2020-07-23 17:20:20]

expected instance path: [/gaussdb/data/db1/pg\_hba.conf]

Notice: the above configuration uses cluster internal communication, your configuration may affect the cluster internal communication.

gs\_guc sethba: host all dim 192.168.0.96/24 sha256: [/gaussdb/data/db1/pg\_hba.conf]

Total instances: 1. Failed instances: 0.

Success to perform gs\_guc!

Total instances: 1. Failed instances: 0.

Success to perform gs\_guc!

Total instances: 1. Failed instances: 0.

Success to perform gs\_guc!

### 审计日志

审计功能开启时会不断产生大量的审计日志，占用磁盘空间。用户可以根据磁盘空间的大小设置审计日志维护策略。

#### 前提条件

用户必须拥有审计权限。

omm用户连接数据库。

以操作系统用户omm登录数据库主节点。

**su - omm**

启动openGauss数据库服务

**gs\_om -t start**

使用如下命令连接数据库。

**gsql -d postgres -p 26000 -r**

postgres为需要连接的数据库名称，26000为数据库主节点的端口号。

#### 背景信息

与审计日志相关的配置参数及其含义请参见下表。

与审计日志相关的配置参数说明

|  |  |  |
| --- | --- | --- |
| 配置项 | 含义 | 默认值 |
| [audit\_directory](#d0e163837) | 审计文件的存储目录。 | /var/log/gaussdb/用户名/pg\_audit |
| [audit\_resource\_policy](#d0e163935) | 审计日志的保存策略。 | on（表示使用空间配置策略） |
| [audit\_space\_limit](#d0e163989) | 审计文件占用的磁盘空间总量。 | 1 GB |
| [audit\_file\_remain\_time](#d0e163966) | 审计日志文件的最小保存时间。 | 90天 |
| [audit\_file\_remain\_threshold](#d0e164009) | 审计目录下审计文件的最大数量。 | 1048576 |

说明：如果使用gs\_om工具部署openGauss，则审计日志路径为 “/var/log/gaussdb/用户名/pg\_audit”。

* + - * 审计日志删除命令为数据库提供的sql函数pg\_delete\_audit，其原型为：

**pg\_delete\_audit(timestamp** *startime***,timestamp** *endtime***)**

* + - * 其中参数startime和endtime分别表示审计记录的开始时间和结束时间。

目前常用的记录审计内容的方式有两种：记录到数据库的表中、记录到OS文件中。这两种方式的优缺点比较如下表所示。

两种方式的优缺点比较

|  |  |  |
| --- | --- | --- |
| 方式 | 优点 | 缺点 |
| 记录到表中 | 不需要用户维护审计日志。 | 由于表是数据库的对象，如果一个数据库用户具有一定的权限，就能够访问到审计表。如果该用户非法操作审计表，审计记录的准确性难以得到保证。 |
| 记录到OS文件中 | 比较安全，即使一个帐户可以访问数据库，但不一定有访问OS这个文件的权限。 | 需要用户维护审计日志。 |

从数据库安全角度出发，openGauss采用记录到OS文件的方式来保存审计结果，保证了审计结果的可靠性。

#### 选择日志维护方式进行维护。

##### 设置自动删除审计日志

审计文件占用的磁盘空间或者审计文件的个数超过指定的最大值时，系统将删除最早的审计文件，并记录审计文件删除信息到审计日志中。

注：审计文件占用的磁盘空间大小默认值为1024MB，用户可以根据磁盘空间大小重新设置参数。

配置审计文件占用磁盘空间的大小（audit\_space\_limit），查看已配置的参数。

postgres=# **SHOW audit\_space\_limit;**

audit\_space\_limit

-------------------

1GB

(1 row)

如果显示结果不为1 GB（1024 MB），执行“\q”命令退出数据库。

postgres=# **\q**

建议执行如下命令设置成默认值1024 MB。

**gs\_guc reload -N all -I all -c "audit\_space\_limit=1024MB"**

显示结果为：

"audit\_space\_limit=1024MB";

Begin to perform gs\_guc for all datanodes.

Total instances: 1. Failed instances: 0.

Success to perform gs\_guc!

配置审计文件个数的最大值（audit\_file\_remain\_threshold）。连接数据库，查看已配置的参数。

连接数据库：

**gsql -d postgres -p 26000 -r**

查看审计文件个数的参数：

postgres=# **SHOW audit\_file\_remain\_threshold;**

audit\_file\_remain\_threshold

-----------------------------

1048576

(1 row)

如果显示结果不为1048576，执行“\q”命令退出数据库。

postgres=# **\q**

建议执行如下命令设置成默认值1048576。

**gs\_guc reload -N all -I all -c "audit\_file\_remain\_threshold=1048576"**

显示为：

Begin to perform gs\_guc for all datanodes.

Total instances: 1. Failed instances: 0.

Success to perform gs\_guc!

##### 手动备份审计文件

当审计文件占用的磁盘空间或者审计文件的个数超过配置文件指定的值时，系统将会自动删除较早的审计文件，因此建议用户周期性地对比较重要的审计日志进行保存。

连接数据库，使用show命令获得审计文件所在目录（audit\_directory）。

连接数据库：

**gsql -d postgres -p 26000 -r**

获得审计文件所在目录：

postgres=# **SHOW audit\_directory;**

audit\_directory

---------------------------------------

/var/log/gaussdb/omm/pg\_audit/dn\_6001

(1 row)

退出数据库

postgres=# **\q**

将审计目录整个拷贝出来进行保存。

**cp -r /var/log/gaussdb/omm/pg\_audit/dn\_6001 /var/log/gaussdb/omm/pg\_audit/dn\_6001\_bak**

查看备份是否成功：

**cd /var/log/gaussdb/omm/pg\_audit/**

**ll -h**

total 8.0K

drwx------ 2 omm dbgrp 4.0K Sep 14 16:52 dn\_6001

drwx------ 2 omm dbgrp 4.0K Sep 15 11:43 dn\_6001\_bak

### WAL日志

预写式日志WAL（Write Ahead Log，也称为Xlog）是指如果要修改数据文件，必须是在这些修改操作已经记录到日志文件之后才能进行修改，即在描述这些变化的日志记录刷新到永久存储器之后。在系统崩溃时，可以使用WAL日志对openGauss进行恢复操作。

#### 操作步骤

以操作系统用户omm登录数据库主节点。

**su - omm**

以一个数据库节点为例，默认在“/gaussdb/data/data\_dn/pg\_xlog”目录下。其中“/gaussdb/data/data\_dn”代表openGauss节点的数据目录，当前情况下为“/gaussdb/data/db1/”。

**cd /gaussdb/data/**

**ls**

db1

切换到db1文件夹。

**cd db1**

**ls**

文件夹中内容如下：

base pg\_ctl.lock pg\_notify postgresql.conf

cacert.pem pg\_errorinfo pg\_replslot postgresql.conf.bak

gaussdb.state pg\_hba.conf pg\_serial postgresql.conf.lock

global pg\_hba.conf.bak pg\_snapshots postmaster.opts

gswlm\_userinfo.cfg pg\_hba.conf.lock pg\_stat\_tmp postmaster.pid

mot.conf pg\_ident.conf pg\_tblspc server.crt

pg\_clog pg\_llog pg\_twophase server.key

pg\_copydir pg\_location PG\_VERSION server.key.cipher

pg\_csnlog pg\_multixact **pg\_xlog**  server.key.rand

切换到pg\_xlog文件夹，查看WAL日志文件。

**cd pg\_xlog**

**ls**

日志文件列表如下：

00001000000010000004E 000000010000000100000066 00000001000000010000007E

00000001000000010000004F 000000010000000100000067 00000001000000010000007F

000000010000000100000050 000000010000000100000068

……

000000010000000100000063 00000001000000010000007B archive\_status

000000010000000100000064 00000001000000010000007C

000000010000000100000065 00000001000000010000007D

### 性能日志

性能日志主要关注外部资源的访问性能问题。性能日志指的是数据库系统在运行时检测物理资源的运行状态的日志，在对外部资源进行访问时的性能检测，包括磁盘、OBS等外部资源的访问检测信息。在出现性能问题时，可以借助性能日志及时的定位问题发生的原因，能极大地提升问题解决效率。

#### 操作步骤

以操作系统用户omm登录数据库主节点。

**su - omm**

数据库节点的性能日志目录在“$GAUSSLOG/gs\_profile”中各自对应的目录下, 其中$GAUSSLOG默认为“/var/log/gaussdb/用户名”， 当前场景下用户名为”omm”。切换到文件夹，查看文件列表的信息。

**cd /var/log/gaussdb/omm/gs\_profile/**

**ls**

dn\_6001

切换到dn\_6001文件夹。

**cd dn\_6001**

**ls –h**

性能日志内容显示如下:

total 4.0K

-rw------- 1 omm dbgrp 0 Jul 21 17:07 postgresql-2020-07-21\_170715.prf

-rw------- 1 omm dbgrp 0 Jul 21 17:44 postgresql-2020-07-21\_174440.prf

-rw------- 1 omm dbgrp 0 Jul 22 10:25 postgresql-2020-07-22\_102526.prf

-rw------- 1 omm dbgrp 0 Jul 23 14:05 postgresql-2020-07-23\_140520.prf

-rw------- 1 omm dbgrp 0 Jul 23 14:29 postgresql-2020-07-23\_142946.prf

-rw------- 1 omm dbgrp 0 Jul 23 14:52 postgresql-2020-07-23\_145207.prf

-rw------- 1 omm dbgrp 0 Jul 23 16:07 postgresql-2020-07-23\_160709.prf

-rw------- 1 omm dbgrp 0 Jul 23 16:23 postgresql-2020-07-23\_162357.prf

-rw------- 1 omm dbgrp 48 Jul 25 00:00 postgresql-2020-07-24\_103226.prf

-rw------- 1 omm dbgrp 0 Jul 25 00:00 postgresql-2020-07-25\_000000.prf

说明：

性能日志主要监控三种资源访问：磁盘、OBS、Hadoop。openGauss不支持OBS、Hadoop，所以只有磁盘访问的监控信息。

磁盘监控的访问信息主要在磁盘文件IO读写的时候进行统计。比如拷贝文件时的读文件IO，正常SQL执行时访问OS表文件的pread系统调用。

性能日志进行收集的配置：logging\_collector 是否进行日志收集；plog\_merge\_age 多久进行一次性能日志汇聚，单位毫秒。logging\_collector 参数为on，且plog\_merge\_age大于0，且是主机正常运行中，恢复过程不进行性能收集。

通过工具gs\_log导出文件进行查看。

本实验结束。

# 附录一：Linux操作系统相关命令

Linux中的命令格式为：command [options] [arguments] 中括号表示可选的，即有些命令不需要选项也不需要参数，但有的命令在运行时需要多个选项或参数。

* options(选项)：选项是调整命令执行行为的开关，选项的不同决定了命令的显示结果不同。
* agruments(参数)：参数是指命令的作用对象。

## vi/vim

文本编辑器，若文件存在则是编辑，若不存在则是创建并编辑文本。

命令语法：

vim [参数]

参数说明：可编辑的文件名。

命令示例：

* 编辑名为clusterconfig的xml文本：

vim clusterconfig.xml

注：

vim编辑器有以下三种模式：

* 正常模式：其它模式下按Esc或Ctrl+[进入，左下角显示文件名或为空。
* 插入模式：正常模式下按i键进入，左下角显示--INSERT--。
* 可视模式：正常模式下按v键进入，左下角显示--VISUAL--。

退出命令（正常模式下）：

* :wq 保存并退出。
* :q! 强制退出并忽略所有更改。
* :e! 放弃所有修改，并打开原有文件。

## cd

显示当前目录的名称，或切换当前的目录（打开指定目录）。

命令语法：

cd [参数]

参数说明：

* 无参数：切换用户当前目录。
* . ：表示当前目录；
* .. ：表示上一级目录；
* ~ ：表示home目录；
* / ：表示根目录。

命令示例：

* 切换到usr目录下的bin目录中：

cd /usr/bin

* 切换到用户home目录：

cd

* 切换到当前目录(cd后面接一个.)：

cd .

* 切换到当前目录上一级目录(cd后面接两个.)：

cd ..

* 切换到用户home目录：

cd ~

* 切换到根目录下：

cd /

注：切换目录需要理解绝对路径和相对路径这两个概念。

* 绝对路径：在Linux中，绝对路径是从/（即根目录）开始的，例如 /opt/software、/etc/profile, 如果目录以 / 就是绝对目录。
* 相对路径：是以 . 或 .. 开始的目录。 . 表示用户当前操作所在的位置，而 .. 表示上级目录。例如 ./gs\_om 表示当前目录下的文件或者目录。

## mv

文件或目录改名(move (rename) files)或将文件或目录移入其它位置,经常用来备份文件或者目录。

命令语法：

mv [选项] 参数1 参数2

常用选项：

* -b：若需覆盖文件，则覆盖前先行备份。

参数说明：

* 参数1：源文件或目录。
* 参数2：目标文件或目录。

命令示例：

* 将文件python重命名为python.bak：

mv python python.bak

* 将/physical/backup目录下的所有文件和目录移到/data/dbn1目录下：

mv /physical/backup/\* /data/dbn1

## curl

在Linux中curl是一个利用URL规则在命令行下工作的文件传输工具。支持文件的上传和下载，是综合传输工具。

命令语法：

curl [选项] [URL]

常用选项：

* -A/--user-agent <string>：设置用户代理发送给服务器；
* -C/--continue-at <offset>：断点续转；
* -D/--dump-header <file>：把header信息写入到该文件中；
* -e/--referer：来源网址；
* -o/--output：把输出写到该文件中；
* -O/--remote-name：把输出写到该文件中，保留远程文件的文件名；
* -s/--silent：静默模式。不输出任何东西；
* -T/--upload-file <file>：上传文件；
* -u/--user <user[:password]>：设置服务器的用户和密码；
* -x/--proxy <host[:port]>：在给定的端口上使用HTTP代理；
* -#/--progress-bar：进度条显示当前的传送状态。

参数说明：

* URL：指定的文件传输URL地址。

命令示例：

* 将url(<https://mirrors.huaweicloud.com/repository/conf/CentOS-7-anon.repo)的内容保存到/etc/yum.repos.d/CentOS-Base.repo>文件中。

curl -o /etc/yum.repos.d/CentOS-Base.repo <https://mirrors.huaweicloud.com/repository/conf/CentOS-7-anon.repo>

* 如果在传输过程中掉线，可以使用-C的方式进行续传。

curl -C -O <https://mirrors.huaweicloud.com/repository/conf/CentOS-7-anon.repo>

## yum

Shell 前端软件包管理器。基于 RPM 包管理，能够从指定的服务器自动下载 RPM 包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软体包，无须繁琐地一次次下载和安装。

命令语法：

yum [options] [command] [package ...]

常用选项：

* -h：查看帮助；
* -y：当安装过程提示选择全部为 "yes"；
* -q：不显示安装的过程。

参数说明：

* command：要进行的操作。
* package：安装的包名。

命令示例：

* 列出所有可更新的软件清单命令：

yum check-update

* 更新所有软件命令：

yum update

* 列出所有可安裝的软件清单命令：

yum list

* 安装指定的软件：

yum install -y libaio-devel flex bison ncurses-devel glibc.devel patch lsb\_release wget python3

## wget

wget是Linux下下载文件的最常用命令。wget支持HTTP,HTTPS和FTP协议,支持自动下载,即可以在用户退出系统后在后台执行,直到下载结束。

命令语法:

wget [选项] [URL]

常用选项:

* -c：接着下载没下载完的文件；
* -b：启动后转入后台执行；
* -P：指定下载目录；
* -O：变更下载文件名；
* --ftp-user --ftp-password：使用FTP用户认证下载。

参数说明:

* 指定的文件下载URL地址。

命令示例：

* 下载openGauss数据库安装文件到当前文件夹：

wget <https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.0.0/x86/openGauss-1.0.0-CentOS-64bit.tar.gz>

* 使用wget断点续传：

wget –c <https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.0.0/x86/openGauss-1.0.0-CentOS-64bit.tar.gz>

## ln

为某一个文件在另外一个位置建立一个同步的链接（软硬链接，不带选项为硬链接）。

当需要在不同的目录，用到相同的文件时，就不需要在每一个需要要的目录下都放一个必须相同的文件，我们只要在某个固定的目录，放上该文件，然后在 其它的目录下用ln命令链接（link）它就可以，不必重复的占用磁盘空间。

命令语法：

ln [选项] 参数1 参数2

常用选项：

* -b --删除，覆盖以前建立的链接；
* -d --允许超级用户制作目录的硬链接；
* -s --软链接(符号链接)。

参数说明：

* 参数1：源文件或目录。
* 参数2：被链接的文件或目录。

命令示例：

* 为python3文件创建软链接/usr/bin/python，如果python3丢失，/usr/bin/python将失效：

ln -s python3 /usr/bin/python

* 为python3创建硬链接/usr/bin/python，python3与/usr/bin/python的各项属性相同：

ln python3 /usr/bin/python

## mkdir

创建指定的名称的目录，要求创建目录的用户在当前目录中具有写权限，并且指定的目录名不能是当前目录中已有的目录。

命令语法：

mkdir [选项] [参数]

常用选项：

* -p --可以是一个路径名称。此时若路径中的某些目录尚不存在,加上此选项后,系统将自动建立好那些尚不存在的目录,即一次可以建立多个目录（递归）；
* -v --每次创建新目录都显示信息；
* -m --设定权限<模式> (类似chmod)，而不是rwxrwxrwx减umask。

参数说明：

* 需要创建的目录。

命令示例：

* 创建一个空目录：

mkdir test

* 递归创建多个目录：

mkdir -p /opt/software/openGauss

* 创建权限为777的目录(目录的权限为rwxrwxrwx)：

mkdir –m 777 test

## chmod命令

更改文件权限。

命令语法：

chmod [选项] <mode> <file...>

常用选项：

* -R, --以递归的方式对目前目录下的所有文件与子目录进行相同的权限变更。

参数说明：

* mode：权限设定字串，详细格式如下 ：

[ugoa...][[+-=][rwxX]...][,...]，

其中，[ugoa...]：u 表示该档案的拥有者，g 表示与该档案的拥有者属于同一个群体(group)者，o 表示其他以外的人，a 表示所有（包含上面三者）；[+-=]：+ 表示增加权限，- 表示取消权限，= 表示唯一设定权限；[rwxX]：r 表示可读取，w 表示可写入，x 表示可执行，X表示只有当该档案是个子目录或者该档案已经被设定过为可执行。

* file：文件列表（单个或者多个文件、文件夹）。

命令示例：

* 设置所有用户可读取文件 cluterconfig.xml：

chmod ugo+r cluterconfig.xml

或

chmod a+r cluterconfig.xml

* 设置当前目录下的所有档案与子目录皆设为任何人可读写：

chmod -R a+rw \*

数字权限使用格式：

* 这种使用方式中，规定数字4、2和1表示读、写、执行权限，即r=4,w=2,x=1。
* 例：rwx = 7（4+2+1）；rw = 6（4+2）；r-x = 5 （4+0+1）；r-- = 4（4+0+0）；--x = 1（0+0+1）；

每个文件都可以针对三个粒度，设置不同的rwx(读写执行)权限。即我们可以用用三个8进制数字分别表示 拥有者 、群组 、其它组( u、 g 、o)的权限详情，并用chmod直接加三个8进制数字的方式直接改变文件权限。语法格式为 ：

chmod <abc> file...

其中，a,b,c各为一个数字，分别代表User、Group、及Other的权限，相当于简化版的chmod u=权限,g=权限,o=权限 file...，而此处的权限将用8进制的数字来表示User、Group、及Other的读、写、执行权限。

命令示例：

* 赋予cluterconfig.xml文件可读可写可执行权限（所有权限）：

chmod 777 cluterconfig.xml

* 赋予/opt/software/openGauss目录下所有文件及其子目录 用户所有权限组可读可执行权限，其他用户可读可执行权限：

chmod R 755 /opt/software/openGauss

## chown

利用 chown 将指定文件的拥有者改为指定的用户或组，用户可以是用户名或者用户ID；组可以是组名或者组ID；文件是以空格分开的要改变权限的文件列表，支持通配符。只有系统管理者(root)才有这样的权限。使用权限 : **root**。

命令语法：

chown [选项] user[:group] file...

常用选项:

* -c : 显示更改的部分的信息；
* -f : 忽略错误信息；
* -R : 处理指定目录以及其子目录下的所有文件。

参数说明

* user : 新的文件拥有者的使用者 ID。
* group : 新的文件拥有者的使用者组(group)。
* flie：文件。

命令示例：

* 将文件 file1.txt 的拥有者设为omm，群体的使用者dbgrp:

chown omm:dbgrp /opt/software/openGauss/clusterconfig.xml

* 将目前目录下的所有文件与子目录的拥有者皆设为omm，群体的使用者dbgrp:

chown -R omm:dbgrp \*

## ls

列出文件和目录的内容。

命令语法：

ls [选项] [参数]

常用选项：

* -l --以长格式显示，列出文件的详细信息，如创建者，创建时间，文件的读写权限列表等等；
* -a --列出文件下所有的文件，包括以"."和".."开头的隐藏文件 （Linux下文件隐藏文件是以 .开头的，如果存在 .. 代表存在着父目录）；
* -d --列出目录本身而非目录内的文件，通常要与-l一起使用；
* -R --同时列出所有子目录层，与-l相似，只是不显示出文件的所有者，相当于编程中的“递归”实现；
* -t --按照时间进行文件的排序，Time（时间）；
* -s --在每个文件的后面打印出文件的大小，size（大小）；
* -S --以文件的大小进行排序。

参数说明：

* 目录或文件。

命令示例：

* 以长格式列出当前目录中的文件及目录：

ls -l

## cp

复制文件或者目录。

命令语法：

cp [选项] 参数1 参数2

常用选项：

* -f --如果目标文件无法打开则将其移除并重试(当 -n 选项存在时则不需再选此项)；
* -n --不要覆盖已存在的文件(使前面的 -i 选项失效)；
* -I --覆盖前询问(使前面的 -n 选项失效)；
* -p --保持指定的属性(默认：模式,所有权,时间戳)，如果可能保持附加属性：环境、链接、xattr 等；
* -R,-r --复制目录及目录内的所有项目。

参数说明：

* 参数1：源文件。
* 参数2：目标文件。

命令示例：

* 将home目录中的abc文件复制到opt目录下：

cp /home/abc /opt

注：目标文件存在时，会询问是否覆盖。这是因为cp是cp -i的别名。目标文件存在时，即使加了-f标志，也还会询问是否覆盖。

## rm

删除一个目录中的一个或多个文件或目录，它也可以将某个目录及其下的所有文件及子目录均删除。对于链接文件，只是删除了链接，原有文件均保持不变。

rm是一个危险的命令，使用的时候要特别当心，否则整个系统就会毁在这个命令（比如在/（根目录）下执行rm \* rf）。所以，我们在执行rm之前最好先确认一下在哪个目录，到底要删除什么东西，操作时保持高度清醒的头脑。

命令语法：

rm [选项] 文件

常用选项：

* -f --忽略不存在的文件，从不给出提示；
* -r --指示rm将参数中列出的全部目录和子目录均递归地删除。

参数说明：

* 需要删除的文件或目录。

命令示例：

* 删除文件：

rm qwe

注：输入rm qwe命令后，系统会询问是否删除，输入y后就会删除文件，不想删除文件则输入n。

* 强制删除某个文件：

rm-rf clusterconfig.log

## cat

连接文件并在标准输出上输出。这个命令常用来显示文件内容，或者将几个文件连接起来显示，或者从标准输入读取内容并显示，它常与重定向符号配合使用。

命令语法：

cat [选项] [参数]

常用选项：

* -E --在每行结束显示$；
* -n –由1开始对给所有输出行编号；
* -b 或 --number-nonblank：和 -n 相似，只不过对于空白行不编号；
* -v --使用 ^ 和 M- 符号，除了 LFD 和 TAB 之外。

参数说明：

* 可操作的文件名。

命令示例:

* 显示testfile文件的内容：

cat textfile

* 把 textfile1 和 textfile2 的文档内容加上行号（空白行不加）之后将内容追加到 textfile3 文档里：

cat -b textfile1 textfile2 >> textfile3

* 向/etc/profile中追加内容（输入EOF表示结束追加）：

cat >>/etc/profile<<EOF

>export LD\_LIBRARY\_PATH=$packagePath/script/gspylib/clib:$LD\_LIBRARY\_PATH

>EOF

注：EOF是end of file的缩写，表示"文字流"（stream）的结尾。"文字流"可以是文件（file），也可以是标准输入（stdin）。在Linux系统之中，EOF是当系统读取到文件结尾，所返回的一个信号值（也就是-1）。

# 附录二：openGauss数据库基本操作

## 查看数据库对象

* 查看帮助信息：

postgres=# \?

* 切换数据库：

*postgres*=# \c dbname

* 列举数据库：

使用\l元命令查看数据库系统的数据库列表。

postgres=# \l

使用如下命令通过系统表pg\_database查询数据库列表。

*postgres*=# SELECT datname FROM pg\_database;

* 列举表：

postgres=# \dt

* 列举所有表、视图和索引：

postgres=# \d+

使用gsql的\d+命令查询表的属性。

*postgres*=# \d+ tablename

* 查看表结构：

*postgres*=# \d tablename

* 列举schema：

postgres=# \dn

* 查看索引：

postgres=# \di

* 查询表空间：

使用gsql程序的元命令查询表空间。

postgres=# \db

检查pg\_tablespace系统表。如下命令可查到系统和用户定义的全部表空间。

*postgres*=# SELECT spcname FROM pg\_tablespace;

* 查看数据库用户列表：

*postgres*=# SELECT \* FROM pg\_user;

* 要查看用户属性：

*postgres*=# SELECT \* FROM pg\_authid;

* 查看所有角色：

*postgres*=# SELECT \* FROM PG\_ROLES;

## 其他操作

* 查看openGauss支持的所有SQL语句。

postgres=#\h

* 切换数据库：

*postgres*=# \c dbname

* 切换用户：

*postgres*=# \c – username

* 退出数据库：

postgres=# \q