

Vastbase

管理员指南

【版权声明】

©2007-2020 北京海量数据技术股份有限公司 版权所有

本档著作权归 **北京海量数据技术股份有限公司**（简称“海量数据”）所有，未经海量数据事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本档内容。

北京海量数据技术股份有限公司保留所有的权利。

【商标声明】



及其它海量数据产品和服务相关的商标均为 **北京海量数据技术股份有限公司** 及其关联公司所有。

本档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本档意在向客户介绍海量数据与云图数据全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的产品、服务的种类、服务标准等应由您与海量数据之间的商业合同约定，除非双方另有约定，否则，海量数据对本档内容不做任何明示或模式的承诺或保证。

目录

1. 启停 Vastbase.....	1
2. 状态查询.....	3
3. 实例主备切换.....	5
4. 生成配置文件.....	7
5. 例行维护.....	9
5.1. 日维护检查项.....	9
5.2. 检查操作系统参数.....	11
5.2.1. 检查办法.....	11
5.2.2. 异常处理.....	12
5.3. 检查 Vastbase 健康状态.....	13
5.3.1. 检查办法.....	13
5.3.2. 异常处理.....	20
5.4. 检查数据库性能.....	24
5.4.1. 检查办法.....	24
5.4.2. 异常处理.....	25
5.5. 检查和清理日志.....	27
5.5.1. 检查操作系统日志.....	27
5.5.2. 检查 Vastbase 运行日志.....	27
5.5.3. 清理运行日志.....	28
5.6. 检查时间一致性.....	29
5.7. 检查应用连接数.....	30
5.8. 例行维护表.....	31
5.9. 例行重建索引.....	33
5.10. 数据安全维护建议.....	34
6. 备份与恢复.....	36
6.1. 概述.....	36
6.2. 物理备份恢复.....	37
6.2.1. vb_basebackup.....	37
6.3. 逻辑备份恢复.....	40
6.3.1. vb_dump.....	40
6.3.2. vb_dumpall.....	50
6.3.3. vb_restore.....	55
7. 高危操作一览表.....	63
8. 日志参考.....	64
8.1. 日志类型简介.....	64
8.2. 系统日志.....	65
8.3. 操作日志.....	65

8.4. 审计日志.....	66
8.5. WAL 日志.....	66
8.6. 性能日志.....	67

1. 启停 Vastbase

启动 Vastbase

步骤 1 以操作系统用户 omm 登录数据库主节点。

步骤 2 使用以下命令启动 Vastbase。

```
gs_om -t start
```

📖 说明

双机启动必须以双机模式启动，若中间过程以单机模式启动，则必须修复才能恢复双机关系，用 vb_ctl build 进行修复，vb_ctl 的使用方法请参见《Vastbase 工具参考》。

---结束

停止 Vastbase

步骤 1 以操作系统用户 omm 登录数据库主节点。

步骤 2 使用以下命令停止 Vastbase。

```
gs_om -t stop
```

📖 说明

启停节点及 AZ 的操作请参见《Vastbase 工具参考》中“服务端工具 > gs_om”章节。

---结束

示例

启动 Vastbase：

```
gs_om -t start
Starting cluster.
=====
Successfully started.
```

停止 Vastbase：

```
gs_om -t stop
Stopping cluster.
=====
Successfully stopped cluster.
=====
End stop cluster.
```

错误排查

如果启动 Vastbase 或者停止 Vastbase 服务失败，请根据日志文件中的日志信息排查错误，参见 8 日志参考。

如果是超时导致启动失败，可以执行如下命令，设置启动超时时间，默认超时时间为 300s。

```
gs_om -t start --time-out=300
```

2. 状态查询

背景信息

Vastbase 支持查看整个 Vastbase 的状态，通过查询结果确认 Vastbase 或者单个主机的运行状态是否正常。

前提条件

Vastbase 已经启动。

操作步骤

步骤 1 以操作系统用户 omm 登录数据库主节点。

步骤 2 使用如下命令查询 Vastbase 状态：

```
gs_om -t status --detail
```

Vastbase 状态显示结果的参数说明请参见表 1-1。

若要查询某主机上的实例状态，同时查看与该主机实例有主从关系的其他实例状态，请在命令中增加“-h”项。示例如下：

```
gs_om -t status -h plat2 --detail
```

其中，plat2 为待查询主机的名称。

若只需要查看当前主机的实例状态，使用如下命令即可：

```
gs_om -t status -h plat1
```

----结束

参数说明

表 1-1 状态说明

字段	字段含义	字段值
cluster_state	Vastbase 状态。显示整个 Vastbase 是否运行正常。	<ul style="list-style-type: none">Normal: 表示 Vastbase 可用，且数据有冗余备份。所有进程都在运行，主备关系正常。Unavailable: 表示 Vastbase 不可用。Degraded: 表示 Vastbase 可用，但存在故障的数据库节点、数据库主节点实例。
node	主机名称	表示该实例所在的主机名称。多 AZ 时会显示 AZ 编号。
node_ip	主机 IP	表示该实例所在的主机 IP。

字段	字段含义	字段值
instance	实例 ID	表示该实例的 ID。
state	实例状态	<ul style="list-style-type: none"> • Primary: 表示实例为主实例。 • Standby: 表示实例为备实例。 • Secondary: 表示实例为从备实例。 • Pending: 表示该实例在仲裁阶段。 • Unknown: 表示实例状态未知。 • Down: 表示实例处于宕机状态。

示例

查看 Vastbase 详细状态信息，含实例状态信息。

```

gs_om -t status --detail
[ Cluster State ]

cluster_state : Normal
redistributing : No
current_az    : AZ_ALL

[ Datanode State ]

node          node_ip      instance      state          | node
node_ip      instance      state
-----
1 pekpopgsci00235 10.244.62.204 6001 /opt/vastbase/cluster/data/dn1 P Primary Normal | 2
pekpopgsci00238 10.244.61.81 6002 /opt/vastbase/cluster/data/dn1 S Standby Normal

```


3. 实例主备切换

操作场景

Vastbase 在运行过程中，数据库管理员可能需要手工对数据库节点做主备切换。例如发现数据库节点主备 failover 后需要恢复原有的主备角色，或怀疑硬件故障需要手动进行主备切换。

📖 说明

主备切换为维护操作，确保 Vastbase 状态正常，所有业务结束后，再进行切换操作。

操作步骤

步骤 1 以操作系统用户 omm 登录数据库任意节点，执行如下命令，查看主备情况。

```
gs_om -t status --detail
```

步骤 2 以操作系统用户 omm 登录准备切换为主节点的备节点，执行如下命令。

```
vb_ctl switchover -D /home/omm/cluster/dn1/
```

/home/omm/cluster/dn1/为备数据库节点的数据目录。

须知

对于同一集群，上一次主备切换未完成，不能执行下一次切换。对于业务正在操作时，发起 switchover，可能主机的线程无法停止导致 switchover 显示超时，实际后台仍然在运行，等主机线程停止后，switchover 即可完成。比如在主机删除一个大的分区表时，可能无法响应 switchover 发起的信号。

步骤 3 switchover 成功后，执行如下命令记录当前主备机器信息。

```
gs_om -t refreshconf
```

---结束

示例

将数据库节点备实例切换为主实例。

1. 查询集群状态。

```
gs_om -t status --detail
[ Cluster State ]

cluster_state   : Normal
redistributing  : No
current_az      : AZ_ALL

[ Datanode State ]

node            node_ip      instance      state         | node
node_ip        instance    state
-----
-----
```

```
1 pekpopgsci00234 10.244.60.70 6001 /home/wuqw/cluster/dn1 P Primary Normal | 2 pekpopgsci00238
10.244.61.81 6002 /home/wuqw/cluster/dn1 S Standby Normal
```

2. 登录备节点，进行主备切换。

```
vb_ctl switchover -D /home/wuqw/cluster/dn1/
[2020-06-17 14:28:01.730][24438][][vb_ctl]: vb_ctl switchover ,datadir is -D "/home/wuqw/cluster/dn1"
[2020-06-17 14:28:01.730][24438][][vb_ctl]: switchover term (1)
[2020-06-17 14:28:01.768][24438][][vb_ctl]: waiting for server to switchover.....
[2020-06-17 14:28:11.175][24438][][vb_ctl]: done
[2020-06-17 14:28:11.175][24438][][vb_ctl]: switchover completed (/home/wuqw/cluster/dn1)
```

3. 保存集群主备机器信息。

```
gs_om -t refreshconf
Generating dynamic configuration file for all nodes.
Successfully generated dynamic configuration file.
```

错误排查

如果 switchover 过程中出现失败，请根据日志文件中的日志信息排查错误，参见 8 日志参考。

异常处理

异常判断标准如下：

- ❖ 业务压力下，主备实例切换时间长，这种情况不需要处理。
- ❖ 其他备机正在 build 的情况下，主机需要发送日志到备机后，才能降备，导致主备切换时间长。这种情况不需要处理，但应尽量避免 build 过程中进行主备切换。
- ❖ 切换过程中，因网络故障、磁盘满等原因造成主备实例连接断开，出现双主现象时，此时请参考如下步骤修复。

警告

出现双主状态后，请按如下步骤恢复成正常的主备状态。否则可能会造成数据丢失。

1. 执行以下命令查询数据库当前的实例状态。

```
gs_om -t status --detail
```

若查询结果显示两个实例的状态都为 Primary，这种状态为异常状态。

2. 确定降为备机的节点，在节点上执行如下命令关闭服务。

```
vb_ctl stop -D /home/omm/cluster/dn1/
```

3. 执行以下命令，以 standby 模式启动备节点。

```
vb_ctl start -D /home/omm/cluster/dn1/ -M standby
```

4. 保存集群主备机器信息。

```
gs_om -t refreshconf
```

5. 查看集群状态，确认实例状态恢复。

4. 生成配置文件

背景信息

在 Vastbase 使用过程中，如果静态配置文件无意损坏后，会影响 Vastbase 感知 Vastbase 拓扑结构和主备关系 gs_om 工具生成静态配置文件，替换已经损坏的配置文件，Vastbase 的正常运行。

前置条件

无。

操作步骤

步骤 1 以操作系统用户 omm 登录数据库主节点。

步骤 2 执行如下命令会在本服务器指定目录下生成配置文件。

```
gs_om -t generateconf -X /opt/software/Vastbase/clusterconfig.xml --distribute
```

/opt/software/Vastbase/clusterconfig.xml 为 Vastbase 安装时 XML 配置文件的存放目录。

📖 说明

- 执行命令后，日志信息中会有新文件的存放的目录。以一主两备环境为例，打开新文件存放目录，会出现 3 个以主机名命名的配置文件，需要用这 3 个文件分别替换对应主机的配置文件。
- 若不使用--distribute 参数，需执行**步骤 3**将静态配置文件分配到对应节点；若使用--distribute 参数，则会将生成的静态配置文件自动分配到对应节点，无需执行**步骤 3**。

步骤 3 （可选）分别替换 3 台主机的/opt/vastbase/app/bin 目录下损坏的静态配置文件。

这里以其中一台主机为例进行介绍。

```
mv /opt/huawei/wisequery/script/static_config_files/cluster_static_config_SIA1000056771  
/opt/vastbase/app/bin/cluster_static_config
```

----结束

示例

在 Vastbase 中的任意主机上执行如下命令，生成配置文件：

```
gs_om -t generateconf -X /opt/software/Vastbase/clusterconfig.xml --distribute  
Generating static configuration files for all nodes.  
Creating temp directory to store static configuration files.  
Successfully created the temp directory.  
Generating static configuration files.  
Successfully generated static configuration files.  
Static configuration files for all nodes are saved in  
/opt/huawei/Bigdata/mppdb/wisequery/script/static_config_files.  
Distributing static configuration files to all nodes.  
Successfully distributed static configuration files.
```

打开生成的配置文件目录，会看到新生成的 3 个文件：

```
cd /opt/huawei/Bigdata/mppdb/wisquery/script/static_config_files
ll
total 456
-rwxr-xr-x 1 omm dbgrp 155648 2016-07-13 15:51 cluster_static_config_plat1
-rwxr-xr-x 1 omm dbgrp 155648 2016-07-13 15:51 cluster_static_config_plat2
-rwxr-xr-x 1 omm dbgrp 155648 2016-07-13 15:51 cluster_static_config_plat3
```

5. 例行维护

[5.1 日维护检查项](#)

[5.2 检查操作系统参数](#)

[5.3 检查 Vastbase 健康状态](#)

[5.4 检查数据库性能](#)

[5.5 检查和清理日志](#)

[5.6 检查时间一致性](#)

[5.7 检查应用连接数](#)

[5.8 例行维护表](#)

[5.9 例行重建索引](#)

[5.10 数据安全维护建议](#)

为保证 Vastbase 数据库中的数据安全，避免丢失数据，非法访问数据等事故发生，请仔细阅读以下内容。

5.1. 日维护检查项

检查 Vastbase 状态

通过 Vastbase 提供的工具查询数据库和实例状态，确认数据库和实例都处于正常的运行状态，可以对外提供数据服务。

- ❖ 检查实例状态

```
gs_check -U omm -i CheckClusterState
```

- ❖ 检查参数

```
vastbase=# SHOW parameter_name;
```

- ❖ 修改参数

```
vb_guc reload -D /vastbase/data/dbnode -c "paraname=value"
```

检查锁信息

锁机制是数据库保证数据一致性的重要手段，检查相关信息可以检查数据库的事务和运行状况。

- ❖ 查询数据库中的锁信息。

```
vastbase=# SELECT * FROM pg_locks;
```

- ❖ 查询等待锁的线程状态信息。

```
vastbase=# SELECT * FROM pg_thread_wait_status WHERE wait_status = 'acquire lock';
```

- ❖ 查询等待锁的事件信息。

```
vastbase=# SELECT node_name, thread_name, tid, wait_status, query_id FROM  
pgxc_thread_wait_status WHERE wait_status = 'acquire lock';
```

- ❖ 结束系统进程

查找正在运行的系统进程，然后使用 kill 命令结束此进程。

```
ps ux  
kill -9 pid
```

统计事件数据

SQL 语句长时间运行会占用大量系统资源，用户可以通过查看事件发生的时间，占用内存大小来了解现在数据库运行状态。

- ❖ 查询事件的时间

查询事件的线程启动时间、事务启动时间、SQL 启动时间以及状态变更时间。

```
vastbase=# SELECT backend_start,xact_start,query_start,state_change FROM pg_stat_activity;
```

- ❖ 查询当前服务器的会话计数信息

```
vastbase=# SELECT count(*) FROM pg_stat_activity;
```

- ❖ 系统级统计信息

查询当前使用内存最多的会话信息。

```
vastbase=# SELECT * FROM pv_session_memory_detail() ORDER BY usedsize desc limit 10;
```

对象检查

表、索引、分区、约束等是数据库的核心存储对象，其核心信息和对象维护是 DBA 重要的日常工作。

- ❖ 查看表的详细信息

```
vastbase=# \d+ table_name
```

- ❖ 查询表统计信息

```
vastbase=# SELECT * FROM pg_statistic;
```

- ❖ 查看索引的详细信息

```
vastbase=# \d+ index_name
```

- ❖ 查询分区表信息

```
vastbase=# SELECT * FROM pg_partition;
```

- ❖ 收集统计信息

使用 ANALYZE 语句收集数据库相关的统计信息。

使用 VACUUM 语句可以回收空间并更新统计信息。

- ❖ 查询约束信息

```
vastbase=# SELECT * FROM pg_constraint;
```

SQL 报告检查

使用 EXPLAIN 语句查看执行计划。

备份

数据备份重于一切，日常应检查备份执行情况，并检查备份有效性，确保备份能够保障数据安全，备份安全加密也应兼顾。

- ❖ 导出指定用户:

```
vb_dump dbname -p port -f out.sql -U user_name -W password
```

- ❖ 导出 schema:

```
vb_dump dbname -p port -n schema_name -f out.sql
```

- ❖ 导出 table:

```
vb_dump dbname -p port -t table_name -f out.sql
```

基本信息检查

基本信息包括版本、组件、补丁集等信息，定期检查数据库信息并登记在案是数据库生命周期管理的重要内容之一。

- ❖ 版本信息

```
vastbase=# SELECT version();
```

- ❖ 容量检查

```
vastbase=# SELECT pg_table_size('table_name');  
vastbase=# SELECT pg_database_size('database_name');
```

5.2. 检查操作系统参数

5.2.1. 检查办法

通过 Vastbase 提供的 `gs_checkos` 工具可以完成操作系统状态检查。

前提条件

- ❖ 当前的硬件和网络环境正常。
- ❖ 各主机间 root 互信状态正常。
- ❖ 只能使用 root 用户执行 `gs_checkos` 命令。

操作步骤

步骤 1 以 root 用户身份登录任意一台服务器。

步骤 2 执行如下命令对 Vastbase 节点服务器的 OS 参数进行检查。

```
gs_checkos -i A
```

检查节点服务器的 OS 参数的目的是为了保证 Vastbase 正常通过预安装，并且在安装成功后可以安全高效的运行。详细的检查项目请参见《Vastbase 工具参考》中“服务端工具 > `gs_checkos`”章节。

---结束

示例

执行 `gs_checkos` 前需要先使用 `gs_preinstall` 工具执行前置脚本，准备环境。以参数“A”为例。

```
gs_checkos -i A  
Checking items:
```

```

A1. [ OS version status ] : Normal
A2. [ Kernel version status ] : Normal
A3. [ Unicode status ] : Normal
A4. [ Time zone status ] : Normal
A5. [ Swap memory status ] : Normal
A6. [ System control parameters status ] : Normal
A7. [ File system configuration status ] : Normal
A8. [ Disk configuration status ] : Normal
A9. [ Pre-read block size status ] : Normal
A10.[ IO scheduler status ] : Normal
A11.[ Network card configuration status ] : Normal
A12.[ Time consistency status ] : Warning
A13.[ Firewall service status ] : Normal
A14.[ THP service status ] : Normal
Total numbers:14. Abnormal numbers:0. Warning number:1.

```

以参数"B"为例。

```

gs_checkos -i B
Setting items:
  B1. [ Set system control parameters ] : Normal
  B2. [ Set file system configuration value ] : Normal
  B3. [ Set pre-read block size value ] : Normal
  B4. [ Set IO scheduler value ] : Normal
  B5. [ Set network card configuration value ] : Normal
  B6. [ Set THP service ] : Normal
Total numbers:6. Abnormal numbers:0. Warning number:0.

```

5.2.2.异常处理

使用 `gs_checkos` 检查或设置状态为 Abnormal，可以使用如下命令查看详细的错误信息。

```
gs_checkos -i A --detail
```

其中，Abnormal 为必须处理项，影响 Vastbase 安装。Warning 可以不处理，不会影响 Vastbase 安装。

- ❖ 如果操作系统版本（A1）检查项检查结果为 Abnormal，需要将不属于混编范围的操作系统版本替换为混编范围内的操作系统版本。
- ❖ 如果内核版本（A2）检查项检查结果为 Warning，则表示 Vastbase 内平台的内核版本不一致。
- ❖ 如果 Unicode 状态（A3）检查项检查结果为 Abnormal，需要将各主机的字符集设置为相同的字符集，可以在 `/etc/profile` 文件中添加 `export LANG=XXX`（XXX 为 Unicode 编码）。

```
vim /etc/profile
```

- ❖ 如果时区状态（A4）检查项检查结果为 Abnormal，需要将各主机的时区设置为相同时区，可以将 `/usr/share/zoneinfo/` 目录下的时区文件拷贝为 `/etc/localtime` 文件。

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

- ❖ 如果交换内存状态（A5）检查项检查结果为 Abnormal，可能是因为 swap 空间大于 mem 空间，可减小 Swap 解决或者增大 Mem 空间解决。
- ❖ 如果系统控制参数（A6）检查项检查结果为 Abnormal，可以使用以下两种方法进行设置。
 - 可以使用如下命令进行设置。

```
gs_checkos -i B1
```


- 根据错误提示信息，在/etc/sysctl.conf 文件中进行设置。然后执行 sysctl -p 使其生效。

```
vim /etc/sysctl.conf
```

- ❖ 如果文件系统配置状态 (A7) 检查项检查结果为 Abnormal，可以使用如下命令进行设置。

```
gs_checkos -i B2
```

- ❖ 如果磁盘配置状态 (A8) 检查项检查结果为 Abnormal，需修改磁盘挂载格式为：

“rw,noatime,inode64,allocsize=16m”。

使用 linux 的 man mount 命令挂载 XFS 选项：

```
rw,noatime,inode64,allocsize=16m
```

也可以在/etc/fstab 文件中设定 XFS 选项。如下示例：

```
/dev/data /data xfs rw,noatime,inode64,allocsize=16m 0 0
```

- ❖ 如果预读块大小 (A9) 检查项检查结果为 Abnormal，可以使用如下命令进行设置。

```
gs_checkos -i B3
```

- ❖ 如果 IO 调度状态 (A10) 检查项检查结果为 Abnormal，可以使用如下命令进行设置。

```
gs_checkos -i B4
```

- ❖ 如果网卡配置状态 (A11) 检查项检查结果为 Warning，可以使用如下命令进行设置。

```
gs_checkos -i B5
```

- ❖ 如果时间一致性 (A12) 检查项检查结果为 Abnormal，需检查是否安装 ntp 服务，以及 ntp 服务是否启动；并与 ntp 时钟源同步。

- ❖ 如果防火墙状态 (A13) 检查项检查结果为 Abnormal，需关闭防火墙服务。使用如下命令进行设置。

- SuSE:

```
SuSEfirewall2 stop
```

- RedHat7:

```
systemctl disable firewalld
```

- RedHat6:

```
service iptables stop
```

- ❖ 如果 THP 服务 (A14) 检查项检查结果为 Abnormal，可以使用如下命令进行设置。

```
gs_checkos -i B6
```

5.3. 检查 Vastbase 健康状态

5.3.1. 检查办法

通过 Vastbase 提供的 gs_check 工具可以开展 Vastbase 健康状态检查。

注意事项

- ❖ 扩容新节点检查只能在 root 用户下执行，其他场景都必须在 omm 用户下执行。
- ❖ 必须指定 -i 或 -e 参数，-i 会检查指定的单项，-e 会检查对应场景配置中的多项。

- ❖ 如果-i 参数中不包含 root 类检查项或-e 场景配置列表中没有 root 类检查项,则不需要交互输入 root 权限的用户及其密码。
- ❖ 可使用--skip-root-items 跳过检查项中包含的 root 类检查,以免需要输入 root 权限用户及密码。
- ❖ 检查扩容新节点与现有节点之间的一致性,在现有节点执行 gs_check 命令指定--hosts 参数进行检查,其中 hosts 文件中需要写入新节点 ip。

操作步骤

方式 1:

步骤 1 以操作系统用户 omm 登录数据库主节点。

步骤 2 执行如下命令对 Vastbase 数据库状态进行检查。

```
gs_check -i CheckClusterState
```

其中, -i 指定检查项,注意区分大小写。格式: -i CheckClusterState、-i CheckCPU 或-i CheckClusterState, CheckCPU。

取值范围为所有支持的检查项名称,详细列表请参见《Vastbase 工具参考》中“服务端工具 > gs_checkos > Vastbase 状态检查表”,用户可以根据需求自己编写新检查项。

----结束

方式 2:

步骤 1 以操作系统用户 omm 登录数据库主节点。

步骤 2 执行如下命令对 Vastbase 数据库进行健康检查。

```
gs_check -e inspect
```

其中, -e 指定场景名,注意区分大小写。格式: -e inspect 或-e upgrade。

取值范围为所有支持的巡检场景名称,默认列表包括: inspect (例行巡检)、upgrade (升级前巡检)、expand (扩容前巡检)、binary_upgrade (就地升级前巡检)、health (健康检查巡检),用户可以根据需求自己编写场景。

----结束

方式 3:

步骤 1 以操作系统用户 omm 登录数据库主节点。

步骤 2 将巡检工具 gs_check 及相关目录 inspection 拷贝分发到所有扩容新节点。

步骤 3 将扩容新节点 ip 写到文件 ipListFile 中,以换行符进行分隔。

步骤 4 执行如下命令对扩容新节点进行扩容前检查。

```
gs_check -e expand_new_node --hosts ipListFile
```

-e 必须为 expand_new_node,即扩容前新节点检查。

----结束

Vastbase 巡检的主要作用是在 Vastbase 运行过程中，检查整个 Vastbase 状态是否正常，或者重大操作前（升级、扩容），确保 Vastbase 满足操作所需的环境条件和状态条件。详细的巡检项目和场景请参见《Vastbase 工具参考》中“服务端工具 > gs_checkos > Vastbase 状态检查表”。

示例

执行单项检查结果：

```
perfadm@lfgp000700749:/opt/huawei/perfadm/tool/script> gs_check -i CheckCPU
Parsing the check items config file successfully
Distribute the context file to remote hosts successfully
Start to health check for the cluster. Total Items:1 Nodes:3

Checking...          [=====] 1/1
Start to analysis the check result
CheckCPU.....OK
The item run on 3 nodes. success: 3

Analysis the check result successfully
Success. All check items run completed. Total:1 Success:1 Failed:0
For more information please refer to
/opt/huawei/wisquery/script/gspylib/inspection/output/CheckReport_201902193704661604.tar.gz
```

本地执行结果：

```
perfadm@lfgp000700749:/opt/huawei/perfadm/tool/script> gs_check -i CheckCPU -L

2017-12-29 17:09:29 [NAM] CheckCPU
2017-12-29 17:09:29 [STD] 检查主机 CPU 占用率，如果 idle 大于 30%并且 iowait 小于 30%.则检查项通过，否则检查项不通过
2017-12-29 17:09:29 [RST] OK

2017-12-29 17:09:29 [RAW]
Linux 4.4.21-69-default (lfgp000700749) 12/29/17 _x86_64_

17:09:24      CPU      %user      %nice      %system      %iowait      %steal      %idle
17:09:25      all      0.25      0.00      0.25      0.00      0.00      99.50
17:09:26      all      0.25      0.00      0.13      0.00      0.00      99.62
17:09:27      all      0.25      0.00      0.25      0.13      0.00      99.37
17:09:28      all      0.38      0.00      0.25      0.00      0.13      99.25
17:09:29      all      1.00      0.00      0.88      0.00      0.00      98.12
Average:      all      0.43      0.00      0.35      0.03      0.03      99.17
```

执行场景检查结果：

```
[perfadm@SIA1000131072 Check]$ gs_check -e inspect
Parsing the check items config file successfully
The below items require root privileges to execute: [CheckBlockdev CheckIOrequestqueue CheckIOConfigure
CheckCheckMultiQueue CheckFirewall CheckSshdService CheckSshdConfig CheckCronService
CheckNoCheckSum CheckSctpSeProcMemory CheckBootItems CheckFilehandle CheckNICModel CheckDropCache]
Please enter root privileges user[root]:root
Please enter password for user[root]:
Please enter password for user[root] on the node[10.244.57.240]:
Check root password connection successfully
Distribute the context file to remote hosts successfully
Start to health check for the cluster. Total Items:59 Nodes:2
```

```
Checking... [ ] 21/59
Checking... [=====] 59/59
Start to analysis the check result
CheckClusterState.....OK
The item run on 2 nodes. success: 2

CheckDBParams.....OK
The item run on 1 nodes. success: 1

CheckDebugSwitch.....OK
The item run on 2 nodes. success: 2

CheckDirPermissions.....OK
The item run on 2 nodes. success: 2

CheckReadOnlyMode.....OK
The item run on 1 nodes. success: 1

CheckEnvProfile.....OK
The item run on 2 nodes. success: 2 (consistent)
The success on all nodes value:
GAUSSHOME /usr1/vastbase/app
LD_LIBRARY_PATH /usr1/vastbase/app/lib
PATH /usr1/vastbase/app/bin

CheckBlockdev.....OK
The item run on 2 nodes. success: 2

CheckCurConnCount.....OK
The item run on 1 nodes. success: 1

CheckCursorNum.....OK
The item run on 1 nodes. success: 1

CheckPgxcgroup.....OK
The item run on 1 nodes. success: 1

CheckDiskFormat.....OK
The item run on 2 nodes. success: 2

CheckSpaceUsage.....OK
The item run on 2 nodes. success: 2

CheckInodeUsage.....OK
The item run on 2 nodes. success: 2

CheckSwapMemory.....OK
The item run on 2 nodes. success: 2

CheckLogicalBlock.....OK
The item run on 2 nodes. success: 2

CheckIOrequestqueue.....WARNING
The item run on 2 nodes. warning: 2
```

```
The warning[host240,host157] value:
On device (vdb) 'IO Request' RealValue '256' ExpectedValue '32768'
On device (vda) 'IO Request' RealValue '256' ExpectedValue '32768'

CheckMaxAsyIOrequests.....OK
The item run on 2 nodes. success: 2

CheckIOConfigure.....OK
The item run on 2 nodes. success: 2

CheckMTU.....OK
The item run on 2 nodes. success: 2 (consistent)
The success on all nodes value:
1500

CheckPing.....OK
The item run on 2 nodes. success: 2

CheckRXTX.....NG
The item run on 2 nodes. ng: 2
The ng[host240,host157] value:
NetWork[eth0]
RX: 256
TX: 256

CheckNetWorkDrop.....OK
The item run on 2 nodes. success: 2

CheckMultiQueue.....OK
The item run on 2 nodes. success: 2

CheckEncoding.....OK
The item run on 2 nodes. success: 2 (consistent)
The success on all nodes value:
LANG=en_US.UTF-8

CheckFirewall.....OK
The item run on 2 nodes. success: 2

CheckKernelVer.....OK
The item run on 2 nodes. success: 2 (consistent)
The success on all nodes value:
3.10.0-957.el7.x86_64

CheckMaxHandle.....OK
The item run on 2 nodes. success: 2

CheckNTPD.....OK
host240: NTPD service is running, 2020-06-02 17:00:28
host157: NTPD service is running, 2020-06-02 17:00:06

CheckOSVer.....OK
host240: The current OS is centos 7.6 64bit.
```

```
host157: The current OS is centos 7.6 64bit.

CheckSysParams.....WARNING
The item run on 2 nodes. warning: 2
The warning[host240,host157] value:
Warning reason: variable 'net.ipv4.tcp_retries1' RealValue '3' ExpectedValue '5'.
Warning reason: variable 'net.ipv4.tcp_syn_retries' RealValue '6' ExpectedValue '5'.
Warning reason: variable 'net.sctp.path_max_retrans' RealValue '5' ExpectedValue '10'.
Warning reason: variable 'net.sctp.max_init_retransmits' RealValue '8' ExpectedValue '10'.

CheckTHP.....OK
The item run on 2 nodes. success: 2

CheckTimeZone.....OK
The item run on 2 nodes. success: 2 (consistent)
The success on all nodes value:
+0800

CheckCPU.....OK
The item run on 2 nodes. success: 2

CheckSshdService.....OK
The item run on 2 nodes. success: 2

CheckSshdConfig.....WARNING
The item run on 2 nodes. warning: 2
The warning[host240,host157] value:

Warning reason: UseDNS parameter is not set; expected: no

CheckCronService.....OK
The item run on 2 nodes. success: 2

CheckStack.....OK
The item run on 2 nodes. success: 2 (consistent)
The success on all nodes value:
8192

CheckNoChecksum.....OK
The item run on 2 nodes. success: 2 (consistent)
The success on all nodes value:
Nochecksum value is N,Check items pass.

CheckSysPortRange.....OK
The item run on 2 nodes. success: 2

CheckMemInfo.....OK
The item run on 2 nodes. success: 2 (consistent)
The success on all nodes value:
totalMem: 31.260929107666016G

CheckHyperThread.....OK
The item run on 2 nodes. success: 2
```

```
CheckTableSpace.....OK
The item run on 1 nodes. success: 1

CheckSctpService.....OK
The item run on 2 nodes. success: 2

CheckSysadminUser.....OK
The item run on 1 nodes. success: 1

CheckGUCConsistent.....OK
All DN instance guc value is consistent.

CheckMaxProcMemory.....OK
The item run on 1 nodes. success: 1

CheckBootItems.....OK
The item run on 2 nodes. success: 2

CheckHashIndex.....OK
The item run on 1 nodes. success: 1

CheckPgxcRedistb.....OK
The item run on 1 nodes. success: 1

CheckNodeGroupName.....OK
The item run on 1 nodes. success: 1

CheckTDDDate.....OK
The item run on 1 nodes. success: 1

CheckDilateSysTab.....OK
The item run on 1 nodes. success: 1

CheckKeyProAdj.....OK
The item run on 2 nodes. success: 2

CheckProStartTime.....WARNING
host157:
STARTED COMMAND
Tue Jun 2 16:57:18 2020 /usr1/dmuser/dmserver/metricdbl/server/bin/vastbase --single_node -D
/usr1/dmuser/dmb1/data -p 22204
Mon Jun 1 16:15:15 2020 /usr1/vastbase/app/bin/vastbase -D /usr1/vastbase/data/dn1 -M standby

CheckFilehandle.....OK
The item run on 2 nodes. success: 2

CheckRouting.....OK
The item run on 2 nodes. success: 2

CheckNICModel.....OK
The item run on 2 nodes. success: 2 (consistent)
The success on all nodes value:
version: 1.0.0
```

```

model: Red Hat, Inc. Virtio network device

CheckDropCache.....WARNING
The item run on 2 nodes. warning: 2
The warning[host240,host157] value:
No DropCache process is running

CheckMpprcFile.....NG
The item run on 2 nodes. ng: 2
The ng[host240,host157] value:
There is no mpprc file

Analysis the check result successfully
Failed. All check items run completed. Total:59 Success:52 Warning:5 NG:2
For more information please refer to
/usr1/vastbase/tool/script/gspylib/inspection/output/CheckReport_inspect611.tar.gz

```

5.3.2.异常处理

如果发现检查结果异常，可以根据以下内容进行修复。

表 1-2 检查 Vastbase 运行状态

检查项	异常状态	处理方法
CheckClusterState (检查 Vastbase 状态)	Vastbase 未启动或 Vastbase 实例未启动	使用以下命令启动 Vastbase 及实例。 <code>gs_om -t start</code>
	Vastbase 状态异常或 Vastbase 实例异常	检查各主机、实例状态，根据状态信息进行排查。 <code>gs_check -i CheckClusterState</code>
CheckDBParams (检查数据库参数)	数据库参数错误	通过 vb_guc 工具修改数据库参数为指定值。
CheckDebugSwitch (检查调试日志)	日志级别不正确	使用 vb_guc 工具将 log_min_messages 改为指定内容。
CheckDirPermissions (检查目录权限)	路径权限错误	修改对应目录权限为指定数值 (750/700) 。 <code>chmod 750 DIR</code>
CheckReadonlyMode (检查只读模式)	只读模式被打开	确认数据库节点所在磁盘使用率未超阈值(默认 60%)且未在执行其他运维操作。 <code>gs_check -i CheckDataDiskUsage</code> <code>ps ux</code>

检查项	异常状态	处理方法
		使用 vb_guc 工具关闭 Vastbase 只读模式 <pre>vb_guc reload -N all -I all -c 'default_transaction_read_only = off' vb_guc reload -N all -I all -c 'default_transaction_read_only = off'</pre>
CheckEnvProfile (检查环境变量)	环境变量不一致	重新执行前置更新环境变量信息。
CheckBlockdev (检查磁盘预读块)	磁盘预读块大小不为 16384	使用 gs_checkos 设置预读块大小为 16384KB，并写入自启动文件。 <pre>gs_checkos -i B3</pre>
CheckCursorNum (检查游标数)	检查游标数失败	检查数据库能否正常连接，Vastbase 状态是否正常。
CheckPgxcgroup (检查重分布状态)	有未完成重分布的 pgxc_group 表	继续完成扩容或缩容的数据重分布操作。 <pre>gs_expand、gs_shrink</pre>
CheckDiskFormat (检查磁盘配置)	各节点磁盘配置不一致	将各节点的磁盘规格改为相同。
CheckSpaceUsage (检查磁盘空间使用率)	磁盘可用空间不足	清理或扩展对应目录所在的磁盘。
CheckInodeUsage (检查磁盘索引使用率)	磁盘可用索引不足	清理或扩展对应目录所在的磁盘。
CheckSwapMemory (检查交换内存)	交换内存大于物理内存	将交换内存调小或关闭。
CheckLogicalBlock (检查磁盘逻辑块)	磁盘逻辑块大小不为 512	使用 gs_checkos 修改磁盘逻辑块大小为 512KB，并写入开机自启动文件。 <pre>gs_checkos -i B4</pre>
CheckIOrequestqueue (检查 IO 请求)	IO 请求值不为 32768	使用 gs_checkos 设置 IO 请求值为 32768，并写入开机自启动文件。 <pre>gs_checkos -i B4</pre>
CheckCurConnCount (检查当前连接数) 111	当前连接数超过最大连接数的 90%	断开未使用的数据库主节点连接。
CheckMaxAsyIOrequests	最大异步请求值小	使用 gs_checkos 设置最大异步请求值为 104857600 和

检查项	异常状态	处理方法
(检查最大异步请求)	于 104857600 或当前节点数据库实例数) *1048576	当前节点数据库实例数*1048576 中的最大值。 <code>gs_checkos -i B4</code>
CheckMTU (检查 MTU 值)	MTU 值不一致	设置各节点的 MTU 一致为 1500 或 8192。 <code>ifconfig eth* MTU 1500</code>
CheckIOConfigure (检查 IO 配置)	IO 配置不是 deadline	使用 gs_checkos 设置 IO 配置为 deadline, 并写入开机自启动文件。 <code>gs_checkos -i B4</code>
CheckRXTX (检查 RXTX 值)	网卡 RX/TX 值不是 4096	使用 checkos 设置 Vastbase 使用的物理网卡 RX/TX 值为 4096 <code>gs_checkos -i B5</code>
CheckPing (检查网络通畅)	存在 VastbaseIP 无法 ping 通	检查异常 ip 间网络设置和状态、防火墙状态。
CheckNetWorkDrop (检查网络丢包率)	网络通信丢包率高于 1%	检查对应 IP 间网络负载、状态。
CheckMultiQueue (检查网卡多队列)	未开启网卡多队列并未将网卡中断绑定到不同 CPU 核心	开启网卡多队列并将网卡队列中断绑定到不同的 CPU 核心。
CheckEncoding (检查编码格式)	各节点编码格式不一致	在/etc/profile 中写入一致的编码信息。 <code>echo "export LANG=XXX" >> /etc/profile</code>
CheckFirewall (检查防火墙)	防火墙未关闭	关闭防火墙服务。 redHat(CentOS)7.x: <code>systemctl status firewalld.service</code> redHat(CentOS)6.x: <code>service iptables down</code> SuSE: <code>SuSEfirewall2 down</code>
CheckKernelVer (检查内核版本)	内核版本不一致	使用 gs_replace 替换不一致的节点。 <code>gs_replace</code>
CheckMaxHandle (检查最大文件句柄数)	最大文件句柄数小于 1000000	设置 91-nofile.conf/90-nofile.conf 最大文件句柄数软硬限制为 1000000。 <code>gs_checkos -i B2</code>
CheckNTPD (检查时间同)	NTPD 服务未开启	开启 NTPD 服务并设置时钟一致。

检查项	异常状态	处理方法
步服务)	或时间误差超过一分钟	
CheckOSVer (检查操作系统版本)	操作系统不被支持或不在同一混搭列表	使用 gs_replace 将异常节点替换为受支持系统或同一混搭列表内的系统的节点。 <code>gs_replace</code>
CheckSysParams (检查操作系统参数)	操作系统参数设置不满足要求	使用 gs_checkos 进行参数设置或手动设置。 <code>gs_checkos -i B1</code> <code>vim /etc/sysctl.conf</code>
CheckTHP (检查 THP 服务)	THP 服务未开启	使用 gs_checkos 设置 THP 服务 <code>gs_checkos -i B6</code>
CheckTimeZone (检查时区)	时区不一致	设置各节点为同一时区 <code>cp /usr/share/zoneinfo/\$主时区/\$次时区 /etc/localtime</code>
CheckCPU (检查 CPU)	CPU 占用过高或 IO 等待过高	进行 CPU 配置升级或磁盘性能升级
CheckSshdService (检查 SSHD 服务)	未开启 SSHD 服务	启动 SSHD 服务并写入开机自启动文件 <code>server sshd start</code> <code>echo "server sshd start" >> initFile</code>
CheckSshdConfig (检查 SSHD 配置)	SSHD 服务配置错误	设置 SSHD 服务, <code>PasswordAuthentication=no;</code> <code>MaxStartups=1000;</code> <code>UseDNS=yes;</code> <code>ClientAliveInterval=10800/ClientAliveInterval=0</code> 并重启服务: <code>server sshd start</code>
CheckCrondService (检查 Crond 服务)	Crond 服务未启动	安装 Crond 服务并启用
CheckStack (检查堆栈大小)	堆栈大小小于 3072	使用 gs_checkos 设置为 3072 并重启堆栈值过小进程。 <code>gs_checkos -i B2</code>
CheckNoCheckSum (检查 NoCheckSum 参数)	NoCheckSum 设置错误或不一致	设置各节点的 NoCheckSum 值一致(存在 redHat6.4/6.5 且为 bond0 时全部设为 Y, 否则全部设为 N) <code>echo Y > /sys/module/sctp/parameters/no_checksums</code>
CheckSysPortRange (检查系统 ip 端口)	系统 ip 端口不在预	设置系统 ip 端口范围参数到 26000-65535 之中; 设置

检查项	异常状态	处理方法
查系统端口设置)	期范围内或 Vastbase 端口在 系统 ip 端口内	Vastbase 端口在系统 ip 端口范围外 <code>vim /etc/sysctl.conf</code>
CheckMemInfo (检查内存信息)	各节点内存大小不一致	使用相同规格的物理内存
CheckHyperThread (检查超线程)	未开启 CPU 超线程	开启 CPU 超线程
CheckTableSpace (检查表空间)	表空间路径和 Vastbase 路径存 在嵌套或表空间路 径相互存在嵌套	将表空间数据迁移到路径合法的表空间中
CheckSctpService (检查 Sctp 服务)	未开启 Sctp 服务	部署并开启 Sctp 服务 <code>modprobe sctp</code>

5.4. 检查数据库性能

5.4.1. 检查办法

通过 Vastbase 提供的性能统计工具 `gs_checkperf` 可以对硬件性能进行检查。

前提条件

- ❖ Vastbase 运行状态正常。
- ❖ 运行在数据库之上的业务运行正常。

操作步骤

- 步骤 1 以操作系统用户 `omm` 登录数据库主节点。
- 步骤 2 执行如下命令对 Vastbase 数据库进行性能检查。

```
gs_checkperf
```

---结束

具体的性能统计项目请参见《Vastbase 工具参考》中“服务端工具 > `gs_checkperf` > 性能检查项”。

示例

以简要格式在屏幕上显示性能统计结果。

```
gs_checkperf -i pmk -U omm
Cluster statistics information:
  Host CPU busy time ratio      : 1.43      %
  MPPDB CPU time % in busy time : 1.88      %
  Shared Buffer Hit ratio       : 99.96      %
  In-memory sort ratio         : 100.00      %
  Physical Reads                : 4
  Physical Writes               : 25
  DB size                       : 70      MB
  Total Physical writes         : 25
  Active SQL count              : 2
  Session count                 : 3
```

5.4.2. 异常处理

使用 gs_checkperf 工具检查 Vastbase 性能状态后，如果发现检查结果发现异常，可以根据以下内容进行修复。

表 1-3 检查 Vastbase 级别性能状态

异常状态	处理方法
主机 CPU 占有率高	<ol style="list-style-type: none">1、更换和增加高性能的 CPU。2、使用 top 命令查看系统哪些进程的 CPU 占有率高，然后使用 kill 命令关闭没有使用的进程。 <pre>top</pre>
Vastbase CPU 占有率高	<ol style="list-style-type: none">1、更换和增加高性能的 CPU。2、使用 top 命令查看数据库哪些进程的 CPU 占有率高，然后使用 kill 命令关闭没有使用的进程。 <pre>top</pre> <ol style="list-style-type: none">3、使用 gs_expand 工具扩容，增加新的主机均衡 CPU 占有率。
共享内存命中率低	<ol style="list-style-type: none">1、扩大内存。2、使用如下命令查看操作系统配置文件/etc/sysctl.conf，调大共享内存 kernel.shmmax 值。 <pre>vim /etc/sysctl.conf</pre>
内存中排序比率低	扩大内存。
I/O、磁盘使用率高	<ol style="list-style-type: none">1、更换高性能的磁盘。

异常状态	处理方法
	2、调整数据布局，尽量将 I/O 请求较合理的分配到所有物理磁盘中。 3、全库进行 VACUUM FULL 操作。 <pre>vacuum full;</pre> 4、进行磁盘整理。 5、降低并发数。
事务统计	查询 pg_stat_activity 系统表，将不必要的连接断开。（登录数据库后查询： <pre>vastbase=# \d+ pg_stat_activity;</pre>

表 1-4 检查节点级别性能状态

异常状态	处理方法
CPU 占有率高	1、更换和增加高性能的 CPU。 2、使用 top 命令查看系统哪些进程的 CPU 占有率高，然后使用 kill 命令关闭没有使用的进程。 <pre>top</pre>
内存使用率过高情况	扩大或清理内存。
I/O 使用率过高情况	1、更换高性能的磁盘。 2、进行磁盘清理。 3、尽可能用内存的读写代替直接磁盘 I/O，使频繁访问的文件或数据放入内存中进行操作处理。

表 1-5 会话/进程级别性能状态

异常状态	处理方法
CPU、内存、I/O 使用率过高情况	查看哪个进程占用 CPU/内存高或 I/O 使用率高，若是无用的进程，则 kill 掉，否则排查具体原因。例如 SQL 执行占用内存大，查看是否 SQL 语句需要优化。

表 1-6 SSD 性能状态

异常状态	处理方法
SSD 读写性能故障	使用以下命令查看 SSD 是否有故障，排查具体故障原因。 <pre>gs_checkperf -i SSD -U omm</pre>

5.5. 检查和清理日志

日志是检查系统运行及故障定位的关键手段。建议按月度例行查看操作系统日志及数据库的运行日志。同时，随着时间的推移，日志的增加会占用较多的磁盘空间。建议按月度清理数据库的运行日志。

5.5.1. 检查操作系统日志

建议按月检查操作系统日志，排除操作系统运行异常隐患。

操作步骤

执行如下命令查看操作系统日志文件。

```
vim /var/log/messages
```

关注其中近一个月出现的 kernel、error、fatal 等字样，根据系统报警信息进行处理。

5.5.2. 检查 Vastbase 运行日志

数据库运行时，某些操作在执行过程中可能会出现错误，数据库依然能够运行。但是此时数据库中的数据可能已经发生不一致的情况。建议按月检查 Vastbase 运行日志，及时发现隐患。

前提条件

- ❖ 收集日志的主机网络通畅且未宕机，数据库安装用户互信正常。
- ❖ 日志收集工具依赖操作系统工具如 gstack，如果未安装该工具，则提示错误后，跳过该收集项。

操作步骤

步骤 1 以操作系统用户 omm 登录数据库主节点。

步骤 2 执行如下命令收集数据库日志。

```
gs_collector --begin-time="20160616 01:01" --end-time="20160616 23:59"
```

20160616 01:01 为日志的开始时间，20160616 23:59 为日志的结束时间。

步骤 3 根据[步骤 2](#)的界面输出提示，进入相应的日志收集目录，解压收集的日志，并检查数据库日志。

以下以日志收集路径“/opt/vastbase/tmp/vastbasea_mppdb/collector_20160726_105158.tar.gz”为例进行操作。

```
tar -xvzf /opt/vastbase/tmp/vastbasea_mppdb/collector_20160726_105158.tar.gz  
cd /opt/vastbase/tmp/vastbasea_mppdb/collector_20160726_105158
```

---结束

示例

- ❖ 以--begin-time 与--end-time 为参数执行 gs_collector 命令。

```
gs_collector --begin-time="20160616 01:01" --end-time="20160616 23:59"
```

当显示类似如下信息表示日志已经归档。

```
Successfully collected files  
All results are stored in /tmp/vastbasea_mppdb/collector_20160616_175615.tar.gz.
```

- ❖ 以--begin-time, --end-time 与-h 为参数执行 gs_collector 命令。

```
gs_collector --begin-time="20160616 01:01" --end-time="20160616 23:59" -h plat2
```

当显示类似如下信息表示日志已经归档。

```
Successfully collected files  
All results are stored in /tmp/vastbasea_mppdb/collector_20160616_190225.tar.gz.
```

- ❖ 以--begin-time, --end-time 与-f 为参数执行 gs_collector 命令。

```
gs_collector --begin-time="20160616 01:01" --end-time="20160616 23:59" -f  
/opt/software/vastbase/output
```

当显示类似如下信息表示日志已经归档。

```
Successfully collected files  
All results are stored in /opt/software/vastbase/output/collector_20160616_190511.tar.gz.
```

- ❖ 以--begin-time, --end-time 与--keyword 为参数执行 gs_collector 命令。

```
gs_collector --begin-time="20160616 01:01" --end-time="20160616 23:59" --keyword="os"
```

当显示类似如下信息表示日志已经归档。

```
Successfully collected files.  
All results are stored in /tmp/vastbasea_mppdb/collector_20160616_190836.tar.gz.
```

- ❖ 以--begin-time, --end-time 与-o 为参数执行 gs_collector 命令。

```
gs_collector --begin-time="20160616 01:01" --end-time="20160616 23:59" -o  
/opt/software/vastbase/output
```

当显示类似如下信息表示日志已经归档。

```
Successfully collected files.  
All results are stored in /opt/software/vastbase/output/collector_20160726_113711.tar.gz.
```

- ❖ 以--begin-time, --end-time 与-l 为参数 (文件名必须以.log 为后缀) 执行 gs_collector 命令。

```
gs_collector --begin-time="20160616 01:01" --end-time="20160616 23:59" -l  
/opt/software/vastbase/logfile.log
```

当显示类似如下信息表示日志已经归档。

```
Successfully collected files.  
All results are stored in /opt/software/vastbase/output/collector_20160726_113711.tar.gz.
```

5.5.3. 清理运行日志

数据库运行过程中会产生大量运行日志，占用大量的磁盘空间，建议清理过期日志文件，只保留一个月的日志。

操作步骤

步骤 1 以操作系统用户 omm 登录数据库主节点。

步骤 2 清理日志。

1. 将超过 1 个月的日志备份到其他磁盘。
2. 进入日志存放目录。

```
cd $GAUSSLOG
```

3. 进入相应的子目录，使用如下方式删除 1 个月之前产生的日志。

```
rm 日志名称
```

日志文件的命名格式为 “postgresql-年-月-日_HHMMSS”。

---结束

5.6. 检查时间一致性

数据库事务一致性通过逻辑时钟保证，与操作系统时间无关，但是系统时间不一致会导致诸多潜在问题，主要是后台运维和监控功能异常，因此在月度检查时建议检查各个节点的时间一致性。

操作步骤

步骤 1 以操作系统用户 omm 登录数据库主节点。

步骤 2 创建记录 Vastbase 各节点的配置文件（mpphosts 文件目录用户可随意指定，建议放在/tmp 下）。

```
vim /tmp/mpphosts
```

增加各节点的主机名称。

```
plat1  
plat2  
plat3
```

步骤 3 保存配置文件。

```
:wq!
```

步骤 4 执行如下命令，输出各节点上的时间到“/tmp/sys_ctl-os1.log”文件中。

```
for ihost in `cat /tmp/mpphosts`; do ssh -n -q $ihost "hostname;date"; done >  
/tmp/sys_ctl-os1.log
```

步骤 5 根据输出确认各个节点的时间一致性，节点之间时间差异不能超过 30 秒。

```
cat /tmp/sys_ctl-os1.log  
plat1  
Thu Feb 9 16:46:38 CST 2017  
plat2  
Thu Feb 9 16:46:49 CST 2017  
plat3  
Thu Feb 9 16:46:14 CST 2017
```

---结束

5.7. 检查应用连接数

如果应用程序与数据库的连接数超过最大值，则新的连接无法建立。建议每天检查连接数，及时释放空闲的连接或者增加最大连接数。

操作步骤

步骤 1 以操作系统用户 omm 登录数据库主节点。

步骤 2 使用如下命令连接数据库。

```
vsql -d postgres -p 8000
```

postgres 为需要连接的数据库名称，8000 为数据库主节点的端口号。

连接成功后，系统显示类似如下信息：

```
vsql ((Vastbase 1.0 build 290d125f) compiled at 2020-05-08 02:59:43 commit 2143 last mr 131
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.
```

```
vastbase=#
```

步骤 3 执行如下 SQL 语句查看连接数。

```
vastbase=# SELECT count(*) FROM (SELECT pg_stat_get_backend_idset() AS backendid) AS s;
```

显示类似如下的信息，其中 2 表示当前有两个应用连接到数据库。

```
count
-----
      2
(1 row)
```

步骤 4 查看现有最大连接数。

```
vastbase=# SHOW max_connections;
```

显示信息如下，其中 200 为现在的最大连接数。

```
max_connections
-----
              200
(1 row)
```

---结束

异常处理

如果显示的连接数接近数据库的最大连接数 max_connections，则需要考虑清理现有连接数或者增加新的连接数。

步骤 1 执行如下 SQL 语句，查看 state 字段等于 idle，且 state_change 字段长时间没有更新过的连接信息。

```
vastbase=# SELECT * FROM pg_stat_activity where state='idle' order by state_change;
```

显示类似如下的信息：

```
datid | datname | pid | usesysid | username | application_name | client_addr
| client_hostname | client_port | backend_start | xact_start | quer
y_start | state_change | waiting | enqueue | state | resource_pool
| query
```


- ❖ VACUUM 对每个表维护了一个可视化映射来跟踪包含对别的活动事务可见的数组的页。一个普通的索引扫描首先通过可视化映射来获取对应的数组,来检查是否对当前事务可见。若无法获取,再通过堆数组抓取的方式来检查。因此更新表的可视化映射,可加速唯一索引扫描。
- ❖ VACUUM 可避免执行的事务数超过数据库阈值时,事务 ID 重叠造成的原有数据丢失。
- ❖ ANALYZE 可收集与数据库中表内容相关的统计信息。统计结果存储在系统表 PG_STATISTIC 中。查询优化器会使用这些统计数据,生成最有效的执行计划。

操作步骤

步骤 1 使用 VACUUM 或 VACUUM FULL 命令,进行磁盘空间回收。

- ❖ VACUUM:

对表执行 VACUUM 操作

```
vastbase=# VACUUM customer;
VACUUM
```

可以与数据库操作命令并行运行。(执行期间,可正常使用的语句: SELECT、INSERT、UPDATE 和 DELETE。不可正常使用的语句: ALTER TABLE)。

对表分区执行 VACUUM 操作

```
vastbase=# VACUUM customer_par PARTITION ( P1 );
VACUUM
```

- ❖ VACUUM FULL:

```
vastbase=# VACUUM FULL customer;
VACUUM
```

需要向正在执行的表增加排他锁,且需要停止其他所有数据库操作。

步骤 2 使用 ANALYZE 语句更新统计信息。

```
vastbase=# ANALYZE customer;
ANALYZE
```

使用 ANALYZE VERBOSE 语句更新统计信息,并输出表的相关信息。

```
vastbase=# ANALYZE VERBOSE customer;
ANALYZE
```

也可以同时执行 VACUUM ANALYZE 命令进行查询优化。

```
vastbase=# VACUUM ANALYZE customer;
VACUUM
```

📖 说明

VACUUM 和 ANALYZE 会导致 I/O 流量的大幅增加,这可能会影响其他活动会话的性能。因此,建议通过“vacuum_cost_delay”参数设置《开发者指南》中“GUC 参数说明 > 资源消耗 > 基于开销的清理延迟”。

步骤 3 删除表

```
vastbase=# DROP TABLE customer;
vastbase=# DROP TABLE customer_par;
vastbase=# DROP TABLE part;
```

当结果显示为如下信息,则表示删除成功。

```
DROP TABLE
```

---结束

维护建议

- ❖ 定期对部分大表做 VACUUM FULL，在性能下降后为全库做 VACUUM FULL，目前暂定每月做一次 VACUUM FULL。
- ❖ 定期对系统表做 VACUUM FULL，主要是 PG_ATTRIBUTE。
- ❖ 启用系统自动清理进程 (AUTOVACUUM) 自动执行 VACUUM 和 ANALYZE，回收被标识为删除状态的记录空间，并更新表的统计数据。

5.9. 例行重建索引

背景信息

数据库经过多次删除操作后，索引页面上的索引键将被删除，造成索引膨胀。例行重建索引，可有效的提高查询效率。

数据库支持的索引类型为 B-tree 索引，例行重建索引可有效的提高查询效率。

- ❖ 如果数据发生大量删除后，索引页面上的索引键将被删除，导致索引页面数量的减少，造成索引膨胀。重建索引可回收浪费的空间。
- ❖ 新建的索引中逻辑结构相邻的页面，通常在物理结构中也是相邻的，所以一个新建的索引比更新了多次的索引访问速度要快。

重建索引

重建索引有以下两种方式：

- ❖ 先删除索引 (DROP INDEX)，再创建索引 (CREATE INDEX)。

在删除索引过程中，会在父表上增加一个短暂的排他锁，阻止相关读写操作。在创建索引过程中，会锁住写操作但是不会锁住读操作，此时读操作只能使用顺序扫描。

- ❖ 使用 REINDEX 语句重建索引。
 - 使用 REINDEX TABLE 语句重建索引，会在重建过程中增加排他锁，阻止相关读写操作。
 - 使用 REINDEX INTERNAL TABLE 语句重建 desc 表 (包括列存表的 cudesc 表) 的索引，会在重建过程中增加排他锁，阻止相关读写操作。

操作步骤

假定在导入表“areaS”上的“area_id”字段上存在普通索引“areaS_idx”。重建索引有以下两种方式：

- ❖ 先删除索引 (DROP INDEX)，再创建索引 (CREATE INDEX)
 - a. 删除索引。

```
vastbase=# DROP INDEX areaS_idx;  
DROP INDEX
```

b. 创建索引。

```
vastbase=# CREATE INDEX areaS_idx ON areaS (area_id);  
CREATE INDEX
```

❖ 使用 REINDEX 重建索引。

– 使用 REINDEX TABLE 语句重建索引。

```
vastbase=# REINDEX TABLE areaS;  
REINDEX
```

– 使用 REINDEX INTERNAL TABLE 重建 desc 表（包括列存表的 cudesc 表）的索引。

```
vastbase=# REINDEX INTERNAL TABLE areaS;  
REINDEX
```

📖 说明

在重建索引前，用户可以通过临时增大 maintenance_work_mem 和 psort_work_mem 的取值来加快索引的重建。

5.10. 数据安全维护建议

为保证 Vastbase 数据库中的数据安全，避免丢失数据，非法访问数据等事故发生，请仔细阅读以下内容。

避免数据被丢失

建议用户规划周期性的物理备份，且对备份文件进行可靠的保存。在系统发生严重错误的情况下，可以利用备份文件，将系统恢复到备份前的状态。

避免数据被非法访问

- ❖ 建议对数据库用户进行权限分级管理。数据库管理员根据业务需要，建立用户并赋予权限，保证各用户对数据库的合理访问。
- ❖ 对于 Vastbase 的服务端和客户端（或基于客户端库开发的应用程序），最好也部署在可信任的内网中。如果服务端和客户端一定要部署在非信任的网络中，需要在服务启动前，打开 SSL 加密，保证数据在非信任网络上的传输安全。需要注意的是，打开 SSL 加密会降低数据库的性能。

避免系统日志泄露个人数据

- ❖ 将调试日志发给他人进行分析前，请删除个人数据。

📖 说明

因为日志级别 (log_min_messages) 设置为 DEBUGx (x 为 DEBUG 级别，取值范围为 1~5) 时，调试日志中记录的信息可能包含用户的个人数据。

- ❖ 将系统日志发给其他人进行分析前，请删除个人数据。因为在默认配置下，当 SQL 语句执行错误时，日志中会记录出错的 SQL 语句，而这些 SQL 语句中可能包含用户个人数据。

❖ 将 `log_min_error_statement` 参数的值设置为 `PANIC` 可以避免在系统日志中记录出错的 SQL 语句，但是关闭此功能将导致发生错误时难以定位。

6. 备份与恢复

[6.1 概述](#)

[6.2 物理备份恢复](#)

[6.3 逻辑备份恢复](#)

6.1. 概述

数据备份是保护数据安全的重要手段之一，为了更好的保护数据安全，Vastbase 数据库支持两种备份恢复类型、多种备份恢复方案，备份和恢复过程中提供数据的可靠性保障机制。

备份与恢复类型可分为逻辑备份与恢复、物理备份与恢复。

- ❖ 逻辑备份与恢复：通过逻辑导出对数据进行备份，逻辑备份只能基于备份时刻进行数据转储，所以恢复时也只能恢复到备份时保存的数据。对于故障点和备份点之间的数据，逻辑备份无能为力，逻辑备份适合备份那些很少变化的数据，当这些数据因误操作被损坏时，可以通过逻辑备份进行快速恢复。如果通过逻辑备份进行全库恢复，通常需要重建数据库，导入备份数据来完成，对于可用性要求很高的数据库，这种恢复时间太长，通常不被采用。由于逻辑备份具有平台无关性，所以更为常见的是，逻辑备份被作为一个数据迁移及移动的主要手段。
- ❖ 物理备份与恢复：通过物理文件拷贝的方式对数据库进行备份，以磁盘块为基本单位将数据从主机复制到备机。通过备份的数据文件及归档日志等文件，数据库可以进行完全恢复。物理备份速度快，一般被用作对数据进行备份和恢复，用于全量备份的场景。通过合理规划，可以低成本进行备份与恢复。

以下为 Vastbase 支持的两类数据备份恢复方案，备份方案也决定了当异常发生时该如何恢复。

表 1-7 两种备份恢复类型对比

备份类型	应用场景	支持的介质	优缺点
逻辑备份与恢复	适合于数据量小的场景。 目前用于表备份恢复，可以备份恢复单表和多表。	<ul style="list-style-type: none">• 磁盘• SSD	可按用户需要进行指定对象的备份和恢复，灵活度高。 当数据量大时，备份效率低。
物理备份与恢复	适用于数据量大的场景，主要用于全量数据备份恢复，也可对整个数据库中的 WAL 归档日志和运行日志进行备份恢复。		数据量大时，备份效率高。

当需要进行备份恢复操作时，主要从以下四个方面考虑数据备份方案。

- 备份对业务的影响在可接受范围。
- 数据库恢复效率。

为尽量减小数据库故障的影响，要使恢复时间减到最少，从而使恢复的效率达到最高。

- 数据可恢复程度。

当数据库失效后，要尽量减少数据损失。

- 数据库恢复成本。

在现网选择备份策略时参考的因素比较多，如备份对象、数据大小、网络配置等，表 6-2 列出了可用的备份策略和每个备份策略的适用场景。

表 1-8 备份策略典型场景

备份策略	关键性能因素	典型数据量	性能规格
集群备份	<ul style="list-style-type: none"> • 数据大小 • 网络配置 	数据：PB 级 对象：约 100 万个	备份： <ul style="list-style-type: none"> • 每个主机 80 Mbit/s (NBU/EISOO+磁盘) • 约 90%磁盘 I/O 速率 (SSD/HDD)
表备份	<ul style="list-style-type: none"> • 表所在模式 • 网络配置 (NBU) 	数据：10 TB 级	备份：基于查询性能速度+I/O 速度 说明 多表备份时，备份耗时计算方式： 总时间 = 表数量 × 起步时间 + 数据总量 / 数据备份速度 其中： <ul style="list-style-type: none"> • 磁盘起步时间为 5s 左右，NBU 起步时间比 DISK 长（取决于 NBU 部署方案）。 • 数据备份速度为单节点 50MB/s 左右（基于 1GB 大小的表，物理机备份到本地磁盘得出此速率）。 表越小，备份性能更低。

6.2. 物理备份恢复

6.2.1. vb_basebackup

背景信息

Vastbase 部署成功后，在数据库运行的过程中，会遇到各种问题及异常状态。Vastbase 提供了 vb_basebackup 工具做基础备份。。

vb_basebackup 工具由数据库安装用户执行。

📖 说明

vb_basebackup 仅支持全量备份，不支持增量。

前提条件

- ❖ 可以正常连接 Vastbase 数据库。
- ❖ 在进行还原时，需要保证各节点备份目录中存在备份文件，若备份文件丢失，则需要从其他节点进行拷贝。

语法

- ❖ 显示帮助信息

```
vb_basebackup -? | --help
```

- ❖ 显示版本号信息

```
vb_basebackup -V | --version
```

参数说明

vb_basebackup 参数可以分为如下几类：

- ❖ -D directory
备份文件输出的目录，必选项。
- ❖ -x
相当于-X 的 fetch 方法
- ❖ -X
在备份中包含所需的事务日志文件（WAL 文件）。

支持的事务日志收集方式为 fetch 或者 stream

- ❖ 常用参数：
 - -c, --checkpoint=fast|spread
设置检查点模式为 fast 或者 spread
 - -l, --label=LABEL
为备份设置标签
 - -P, --progress
启用进展报告
 - -v, --verbose
启用冗长模式
 - -V, --version
打印版本后退出

- ?, --help

显示 vb_basebackup 命令行参数

❖ 连接参数

- h, --host=HOSTNAME

指定正在运行服务器的主机名或者 Unix 域套接字的路径。

- p, --port=PORT

指定数据库服务器的端口号。

可以通过 port 参数修改默认端口号。

- U, --username=USERNAME

指定连接数据库的用户。

- s, --status-interval=INTERVAL

发送到服务器的状态包的时间(以秒为单位)

- w, --no-password

不出现输入密码提示。

- W, --password

当使用-U 参数连接本地数据库或者连接远端数据库时, 可通过指定该选项出现输入密码提示。

示例

```
vb_basebackup -D /home/test/trunk/install/data/backup -h 127.0.0.1 -p 21233
INFO: The starting position of the xlog copy of the full build is: 0/1B800000. The slot minimum LSN
is: 0/1B800000.
```

操作步骤

当数据库发生问题需要从备份进行恢复时, 步骤如下:

📖 说明

vb_basebackup 工具只做基础备份, 要结合使用 pg_xlog/目录下日志备份将数据库恢复到某一时间点的状态。

步骤 1 停止 Vastbase 数据库, 具体操作请参见《管理员指南》。

步骤 2 将数据库数据目录和所有表空间复制到一个临时位置, 以备日后需要时使用。

步骤 3 删除数据库数据目录下、正在使用的任何表空间的根目录下所有的现有文件和子目录。

步骤 4 使用数据库系统用户权限从备份中还原数据库文件。

步骤 5 删除 pg_xlog/中存在的所有文件。

步骤 6 如[步骤 2](#)中备份的 pg_xlog/目录下存在未归档的日志文件, 请手动复制到数据库的 pg_xlog/目录下。

- 步骤 7** 在数据库数据目录中创建恢复命令文件 `recovery.conf`。文件包含 `restore_command`、`recovery_target_name` 等信息，请根据实际情况配置文件内容。
 - 步骤 8** 重启服务器。服务器将进入恢复模式，并将读取所需的已归档 WAL 文件。恢复过程完成后，服务器会将 `recovery.conf` 重命名为 `recovery.done`，然后可以开始正常的数据库操作。
 - 步骤 9** 检查数据库的内容，确保数据库已恢复到所需状态。
- 结束

6.3. 逻辑备份恢复

6.3.1. vb_dump

背景信息

`vb_dump` 是 Vastbase 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）。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。

`vb_dump` 工具由操作系统用户 `omm` 执行。

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

`vb_dump` 工具支持导出完整一致的数据。例如，T1 时刻启动 `vb_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。

`vb_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。

- ❖ 纯文本格式的 SQL 脚本文件：包含将数据库恢复为其保存时的状态所需的 SQL 语句。通过 `vsql` 运行该 SQL 脚本文件，可以恢复数据库。即使在其他主机和其他数据库产品上，只要对 SQL 脚本文件稍作修改，也可以用来重建数据库。
- ❖ 归档格式文件：包含将数据库恢复为其保存时的状态所需的数据，可以是 `tar` 格式、目录归档格式或自定义归档格式，详见表 6-3。该导出结果必须与 6.3.3 `vb_restore` 配合使用来恢复数据库，`vb_restore` 工具在导入时，系统允许用户选择需要导入的内容，甚至可以在导入之前对等待导入的内容进行排序。

主要功能

`vb_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 1-9 所示。

表 1-9 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 <code>vsql</code> 命令行终端程序执行，用于重新创	小型数据库，一般推荐纯文本格式。	使用 <code>vsql</code> 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

格式名称	-F 的参数值	说明	建议	对应导入工具
		建数据库对象并加载表数据。		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 6.3.3 vb_restore 可以选择要从自定义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。	-	

📖 说明

可以使用 vb_dump 程序将文件压缩为纯文本或自定义归档导出文件，减少导出文件的大小。生成纯文本导出文件时，默认不压缩。生成自定义归档导出文件时，默认进行中等级别的压缩。vb_dump 程序无法压缩已归档导出文件。通过压缩方式导出纯文本格式文件，vsq1 无法成功导入数据对象。

注意事项

禁止修改导出的文件和内容，否则可能无法恢复成功。

为了保证数据一致性和完整性，vb_dump 会对需要转储的表设置共享锁。如果表在别的事务中设置了共享锁，vb_dump 会等待锁释放后锁定表。如果无法在指定时间内锁定某个表，转储会失败。用户可以通过指定--lock-wait-timeout 选项，自定义等待锁超时时间。

语法

```
vb_dump [OPTION]... [DBNAME]
```

📖 说明

“dbname”前面不需要加短或长选项。“dbname”指定要连接的数据库。

例如：

不需要-d，直接指定“dbname”。

```
vb_dump -p port_number postgres -f dump1.sql
```

或者

```
export PGDATABASE=postgres
vb_dump -p port_number -f dump1.sql
```

环境变量： PGDATABASE

参数说明

通用参数：

❖ -f, --file=FILENAME

将输出发送至指定文件或目录。如果省略该参数，则使用标准输出。如果输出格式为(-F c/-F d/-F t)时，必须指定-f 参数。如果-f 的参数值含有目录，要求目录对当前用户具有读写权限，并且不能指定已有目录。

❖ -F, --format=c|d|t|p

选择输出格式。格式如下：

- p|plain: 输出一个文本 SQL 脚本文件（默认）。
- c|custom: 输出一个自定义格式的归档，并且以目录形式输出，作为 vb_restore 输入信息。该格式是最灵活的输出格式，因为能手动选择，而且能在恢复过程中将归档项重新排序。该格式默认状态下会被压缩。
- d|directory: 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。
- t|tar: 输出一个 tar 格式的归档形式，作为 vb_restore 输入信息。tar 格式与目录格式兼容；tar 格式归档形式在提取过程中会生成一个有效的目录格式归档形式。但是，tar 格式不支持压缩且对于单独表有 8GB 的大小限制。此外，表数据项的相应排序在恢复过程中不能更改。

输出一个 tar 格式的归档形式，也可以作为 vsql 输入信息。

❖ -v, --verbose

指定 verbose 模式。该选项将导致 vb_dump 向转储文件输出详细的对象注解和启动/停止次数，向标准错误流输出处理信息。

❖ -V, --version

打印 vb_dump 版本，然后退出。

❖ -Z, --compress=0-9

指定使用的压缩比级别。

取值范围：0~9

- 0 表示无压缩。
- 1 表示压缩比最小，处理速度最快。
- 9 表示压缩比最大，处理速度最慢。

针对自定义归档格式，该选项指定单个表数据片段的压缩，默认方式是以中等级别进行压缩。p、tar 归档格式目前不支持压缩。

❖ --lock-wait-timeout=TIMEOUT

请勿在转储刚开始时一直等待以获取共享表锁。如果无法在指定时间内锁定某个表，就选择失败。可以以任何符合 SET statement_timeout 的格式指定超时时间。

❖ `-?, --help`

显示 `vb_dump` 命令行参数帮助，然后退出。

转储参数：

❖ `-a, --data-only`

只输出数据，不输出模式(数据定义)。转储表数据、大对象和序列值。

❖ `-b, --blobs`

该参数为扩展预留接口，不建议使用。

❖ `-c, --clean`

在将创建数据库对象的指令输出到备份文件之前，先将清理（删除）数据库对象的指令输出到备份文件中。（如果目标数据库中没有任何对象，`vb_restore` 工具可能会输出一些提示性的错误信息）

该选项只对文本格式有意义。针对归档格式，可以在调用 `vb_restore` 时指定选项。

❖ `-C, --create`

备份文件以创建数据库和连接到创建的数据库的命令开始。（如果命令脚本是这种方式执行，无所谓在运行脚本之前连接的是哪个数据库。）

该选项只对文本格式有意义。针对归档格式，可以在调用 `vb_restore` 时指定选项。

❖ `-E, --encoding=ENCODING`

以指定的字符集编码创建转储。默认情况下，以数据库编码创建转储。（得到相同结果的另一个办法是将环境变量“PGCLIENTENCODING”设置为所需的转储编码。）

❖ `-n, --schema=SCHEMA`

只转储与模式名称匹配的模式，此选项包括模式本身和所有它包含的对象。如果该选项没有指定，所有在目标数据库中的非系统模式将会被转储。写入多个 `-n` 选项来选择多个模式。此外，根据 `vsql` 的 `\d` 命令所使用的相同规则，模式参数可被理解成一个 pattern，所以多个模式也可以通过在该 pattern 中写入通配符来选择。使用通配符时，注意给 pattern 打引号，防止 shell 扩展通配符。

📖 说明

- 当 `-n` 已指定时，`vb_dump` 不会转储已选模式所附着的任何其他数据库对象。因此，无法保证某个指定模式的转储结果能够自行成功地储存到一个空数据库中。
- 当 `-n` 指定时，非模式对象不会被转储。

转储支持多个模式的转储。多次输入 `-n schemaname` 转储多个模式。

例如：

```
vb_dump -h host_name -p port_number postgres -f backup/bkp_sch12.sql -n sch1 -n sch2
```

在上面这个例子中，`sch1` 和 `sch2` 会被转储。

❖ **-N, --exclude-schema=SCHEMA**

不转储任何与模式 `pattern` 匹配的模式。Pattern 将参照针对 `-n` 的相同规则来理解。可以通过输入多次 `-N`，不转储与任何 `pattern` 匹配的模式。

当同时输入 `-n` 和 `-N` 时，会转储与至少一个 `-n` 选项匹配、与 `-N` 选项不匹配的模式。如果有 `-N` 没有 `-n`，则不转储常规转储中与 `-N` 匹配的模式。

转储过程支持排除多个模式。

在转储过程中，输入 `-N exclude schema name` 排除多个模式。

例如：

```
vb_dump -h host_name -p port_number postgres -f backup/bkp_sch12.sql -N sch1 -N sch2
```

在上面这个例子中，`sch1` 和 `sch2` 在转储过程中会被排除。

❖ **-o, --oids**

转储每个表的对象标识符 (OIDs)，作为表的一部分数据。该选项用于应用以某种方式参照了 OID 列的情况。如果不是以上这种情况，请勿使用该选项。

❖ **-O, --no-owner**

不输出设置对象的归属这样的命令，以匹配原始数据库。默认情况下，`vb_dump` 会发出 `ALTER OWNER` 或 `SET SESSION AUTHORIZATION` 语句设置所创建的数据库对象的归属。如果脚本正在运行，该语句不会执行成功，除非是由系统管理员触发（或是拥有脚本中所有对象的同一个用户）。通过指定 `-O`，编写一个任何用户都能存储的脚本，且该脚本会授予该用户拥有所有对象的权限。

该选项只对文本格式有意义。针对归档格式，可以在调用 `vb_restore` 时指定选项。

❖ **-s, --schema-only**

只转储对象定义（模式），而非数据。

❖ **-S, --sysadmin=NAME**

该参数为扩展预留接口，不建议使用。

❖ **-t, --table=TABLE**

指定转储的表（或视图、或序列、或外表）对象列表，可以使用多个 `-t` 选项来选择多个表，也可以使用通配符指定多个表对象。

当使用通配符指定多个表对象时，注意给 `pattern` 打引号，防止 shell 扩展通配符。

当使用 `-t` 时，`-n` 和 `-N` 没有任何效应，这是因为由 `-t` 选择的表的转储不受那些选项的影响。

📖 说明

- `-t` 参数选项个数必须小于等于 100。
- 如果 `-t` 参数选项个数大于 100，建议使用参数 `--include-table-file` 来替换。
- 当 `-t` 已指定时，`vb_dump` 不会转储已选表所附着的任何其他数据库对象。因此，无法保证某个指定表的转储结果能够自行成功地储存到一个空数据库中。

- -t tablename 只转储在默认搜索路径中可见的表。-t *.tablename 转储数据库下所有模式下的 tablename 表。
- t schema.table 转储特定模式中的表。
- -t tablename 不会导出表上的触发器信息。

例如：

```
vb_dump -h host_name -p port_number postgres -f backup/bkp_sh12.sql -t schema1.table1 -t schema2.table2
```

在上面这个例子中，schema1.table1 和 schema2.table2 会被转储。

❖ --include-table-file=FILENAME

指定需要 dump 的表文件。

❖ -T, --exclude-table=TABLE

不转储的表（或视图、或序列、或外表）对象列表，可以使用多个-t 选项来选择多个表，也可以使用通配符指定多个表对象。

当同时输入-t 和-T 时，会转储在-t 列表中，而不在-T 列表中的表对象。

例如：

```
vb_dump -h host_name -p port_number postgres -f backup/bkp_sh12.sql -T table1 -T table2
```

在上面这个例子中，table1 和 table2 在转储过程中会被排除。

❖ --exclude-table-file=FILENAME

指定不需要 dump 的表文件。

📖 说明

同--include-table-file，其内容格式如下：

```
schema1.table1
schema2.table2
.....
```

❖ -x, --no-privileges|--no-acl

防止转储访问权限（授权/撤销命令）。

❖ --binary-upgrade

该参数为扩展预留接口，不建议使用。

❖ --binary-upgrade-usermap="USER1=USER2"

该参数为扩展预留接口，不建议使用。

❖ --column-inserts|--attribute-inserts

以 INSERT 命令带列名 (INSERT INTO 表 (列、...) 值...) 方式导出数据。这会导致恢复缓慢。但是由于该选项会针对每行生成一个独立分开的命令，所以在重新加载某行时出现的错误只会导致那行丢失，而非整个表内容。

❖ --disable-dollar-quoting

该选项将禁止在函数体前使用美元符号\$，并强制使用 SQL 标准字符串语法对其进行引用。

- ❖ `--disable-triggers`
该参数为扩展预留接口，不建议使用。
- ❖ `--exclude-table-data=TABLE`
指定不转储任何匹配表 `pattern` 的表这方面的数据。依照针对 `t` 的相同规则理解该 `pattern`。
可多次输入 `--exclude-table-data` 来排除匹配任何 `pattern` 的表。当用户需要特定表的定义但不需要其中的数据时，这个选项很有帮助。
排除数据库中所有表的数据，参见 [--schema-only](#)。
- ❖ `--inserts`
发出 `INSERT` 命令（而非 `COPY` 命令）时转储数据。这会导致恢复缓慢。
但是由于该选项会针对每行生成一个独立分开的命令，所以在重新加载某行时出现的错误只会导致那行丢失，而非整个表内容。注意如果重排列顺序，可能会导致恢复整个失败。列顺序改变时，`--column-inserts` 选项不受影响，虽然会更慢。
- ❖ `--no-security-labels`
该参数为扩展预留接口，不建议使用。
- ❖ `--no-tablespaces`
不输出选择表空间的命令。使用该选项，无论默认表空间是哪个，在恢复过程中所有对象都会被创建。
该选项只对文本格式有意义。针对归档格式，可以在调用 `vb_restore` 时指定选项。
- ❖ `--no-unlogged-table-data`
该参数为扩展预留接口，不建议使用。
- ❖ `--non-lock-table`
该参数为扩展预留接口，不建议使用。
- ❖ `--include-alter-table`
转储表删除列。该选项会记录列的删除。
- ❖ `--quote-all-identifiers`
强制对所有标识符加引号。为了向后续版本迁移，且其中可能涉及引入额外关键词，在转储相应数据库时该选项会有帮助。
- ❖ `--section=SECTION`
指定已转储的名称区段（`pre-data`、`data`、和 `post-data`）。
- ❖ `--serializable-deferrable`
转储过程中使用可串行化事务，以确保所使用的快照与之后的数据库状态一致；要实现该操作需要在无异常状况的事务流中等待某个点，因为这样才能保证转储成功，避免引起其他事务出现 `serialization_failure` 要重新再做。

但是该选项对于灾难恢复没有益处。对于在原始数据库进行升级的时候，加载一个数据库的拷贝作为报告或其他只读加载共享的转储是有帮助的。没有这个选项，转储会反映一个与任何事务最终提交的序列化执行不一致的状态。

如果当 `vb_dump` 启动时，读写事务仍处于非活动状态，即便使用该选项也不会对其产生影响。如果读写事务处于活动状态，转储的开始时间可能会延迟一段不确定的时间。

❖ `--use-set-session-authorization`

输出符合 SQL 标准的 `SET SESSION AUTHORIZATION` 命令而不是 `ALTER OWNER` 命令来确定对象所有权。这样令转储更加符合标准，但是如果转储文件中的对象的历史有些问题，那么可能不能正确恢复。并且，使用 `SET SESSION AUTHORIZATION` 的转储需要数据库系统管理员的权限才能转储成功，而 `ALTER OWNER` 需要的权限则低得多。

❖ `--with-encryption=AES128`

指定转储数据需用 AES128 进行加密。

❖ `--with-key=KEY`

AES128 密钥长度必须是 16 字节。

❖ `--include-depend-objs`

备份结果包含依赖于指定对象的对象信息。该参数需要同 `-t/--include-table-file` 参数关联使用才会生效。

❖ `--exclude-self`

备份结果不包含指定对象自身的信息。该参数需要同 `-t/--include-table-file` 参数关联使用才会生效。

❖ `--dont-overwrite-file`

文本、tar、以及自定义格式情况下会重写现有文件。这对目录格式不适用。

例如：

设想这样一种情景，即当前目录下 `backup.sql` 已存在。如果在输入命令中输入 `-f backup.sql` 选项时，当前目录恰好也生成 `backup.sql`，文件就会被重写。

如果备份文件已存在，且输入 `--dont-overwrite-file` 选项，则会报告附带 ‘转储文件已经存在’ 信息的错误。

```
vb_dump -p port_number postgres -f backup.sql -F plain --dont-overwrite-file
```

📖 说明

- `-s/--schema-only` 和 `-a/--data-only` 不能同时使用。
- `-c/--clean` 和 `-a/--data-only` 不能同时使用。
- `--inserts/--column-inserts` 和 `-o/--oids` 不能同时使用，因为 `INSERT` 命令不能设置 OIDS。
- `--role` 和 `--rolepassword` 必须一起使用。
- `--binary-upgrade-usermap` 和 `--binary-upgrade` 必须一起使用。
- `--include-depend-objs/--exclude-self` 需要同 `-t/--include-table-file` 参数关联使用才会生效
- `--exclude-self` 必须同 `--include-depend-objs` 一起使用。

连接参数:

❖ -h, --host=HOSTNAME

指定主机名称。如果数值以斜杠开头, 则被用作到 Unix 域套接字的路径。缺省从 PGHOST 环境变量中获取 (如果已设置), 否则, 尝试一个 Unix 域套接字连接。

该参数只针对 Vastbase 外, 对 Vastbase 内本机只能用 127.0.0.1。

例如: 主机名

环境变量: PGHOST

❖ -p, --port=PORT

指定主机端口。在开启线程池情况下, 建议使用 pooler port, 即主机端口+1。

环境变量: PGPORT

❖ -U, --username=NAME

指定所连接主机的用户名。

不指定连接主机的用户名时, 用户默认系统管理员。

环境变量: PGUSER

❖ -w, --no-password

不出现输入密码提示。如果主机要求密码认证并且密码没有通过其它形式给出, 则连接尝试将会失败。该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

❖ -W, --password=PASSWORD

指定用户连接的密码。如果主机的认证策略是 trust, 则不会对系统管理员进行密码验证, 即无需输入 -W 选项; 如果没有 -W 选项, 并且不是系统管理员, “Dump Restore 工具” 会提示用户输入密码。

❖ --role=ROLENAME

指定创建转储使用的角色名。选择该选项, 会使 vb_dump 连接数据库后, 发起一个 SET ROLE 角色名命令。当所授权用户 (由 -U 指定) 没有 vb_dump 要求的权限时, 该选项会起到作用, 即切换到具备相应权限的角色。某些安装操作规定不允许直接以超系统管理员身份登录, 而使用该选项能够在不违反该规定的情况下完成转储。

❖ --rolepassword=ROLEPASSWORD

指定角色名的密码。

说明

如果 Vastbase 有任何本地数据要添加到 template1 数据库, 请谨慎将 vb_dump 的输出恢复到一个真正的空数据库中, 否则可能会因为被添加对象的定义被复制, 出现错误。要创建一个无本地添加的空数据库, 需从 template0 而非 template1 复制, 例如:

```
CREATE DATABASE foo WITH TEMPLATE template0;
```

tar 归档形式的文件大小不得超过 8GB (tar 文件格式的固有限制)。tar 文档整体大小和任何其他输出格式没有限制，操作系统可能对此有要求。

由 vb_dump 生成的转储文件不包含优化程序用来做执行计划决定的统计数据。因此，最好从某转储文件恢复之后运行 ANALYZE 以确保最佳效果。转储文件不包含任何 ALTER DATABASE...SET 命令，这些设置由 vb_dumpall 转储，还有数据库用户和其他完成安装设置。

示例

使用 vb_dump 转储数据库为 SQL 文本文件或其它格式的操作，如下所示。

示例中“Bigdata@123”表示数据库用户密码；“backup/MPPDB_backup.sql”表示导出的文件，其中 backup 表示相对于当前目录的相对目录；“37300”表示数据库服务器端口；“postgres”表示要访问的数据库名。

📖 说明

导出操作时，请确保该目录存在并且当前的操作系统用户对其具有读写权限。

示例 1: 执行 vb_dump，导出 postgres 数据库全量信息，导出的 MPPDB_backup.sql 文件格式为纯文本格式。

```
vb_dump -U omm -W Bigdata@123 -f backup/MPPDB_backup.sql -p 37300 postgres -F p
vb_dump[port='37300'][postgres][2018-06-27 09:49:17]: The total objects number is 356.
vb_dump[port='37300'][postgres][2018-06-27 09:49:17]: [100.00%] 356 objects have been dumped.
vb_dump[port='37300'][postgres][2018-06-27 09:49:17]: dump database postgres successfully
vb_dump[port='37300'][postgres][2018-06-27 09:49:17]: total time: 1274 ms
```

使用 vsql 程序从纯文本导出文件中导入数据。

示例 2: 执行 vb_dump，导出 postgres 数据库全量信息，导出的 MPPDB_backup.tar 文件格式为 tar 格式。

```
vb_dump -U omm -W Bigdata@123 -f backup/MPPDB_backup.tar -p 37300 postgres -F t
vb_dump[port='37300'][postgres][2018-06-27 10:02:24]: The total objects number is 1369.
vb_dump[port='37300'][postgres][2018-06-27 10:02:53]: [100.00%] 1369 objects have been dumped.
vb_dump[port='37300'][postgres][2018-06-27 10:02:53]: dump database postgres successfully
vb_dump[port='37300'][postgres][2018-06-27 10:02:53]: total time: 50086 ms
```

示例 3: 执行 vb_dump，导出 postgres 数据库全量信息，导出的 MPPDB_backup.dmp 文件格式为自定义归档格式。

```
vb_dump -U omm -W Bigdata@123 -f backup/MPPDB_backup.dmp -p 37300 postgres -F c
vb_dump[port='37300'][postgres][2018-06-27 10:05:40]: The total objects number is 1369.
vb_dump[port='37300'][postgres][2018-06-27 10:06:03]: [100.00%] 1369 objects have been dumped.
vb_dump[port='37300'][postgres][2018-06-27 10:06:03]: dump database postgres successfully
vb_dump[port='37300'][postgres][2018-06-27 10:06:03]: total time: 36620 ms
```

示例 4: 执行 vb_dump，导出 postgres 数据库全量信息，导出的 MPPDB_backup 文件格式为目录格式。

```
vb_dump -U omm -W Bigdata@123 -f backup/MPPDB_backup -p 37300 postgres -F d
vb_dump[port='37300'][postgres][2018-06-27 10:16:04]: The total objects number is 1369.
vb_dump[port='37300'][postgres][2018-06-27 10:16:23]: [100.00%] 1369 objects have been dumped.
vb_dump[port='37300'][postgres][2018-06-27 10:16:23]: dump database postgres successfully
vb_dump[port='37300'][postgres][2018-06-27 10:16:23]: total time: 33977 ms
```

示例 5: 执行 vb_dump, 导出 postgres 数据库信息, 但不导出/home/MPPDB_temp.sql 中指定的表信息。导出的 MPPDB_backup.sql 文件格式为纯文本格式。

```
vb_dump -U omm -W Bigdata@123 -p 37300 postgres --exclude-table-file=/home/MPPDB_temp.sql -f backup/MPPDB_backup.sql
vb_dump[port='37300'][postgres][2018-06-27 10:37:01]: The total objects number is 1367.
vb_dump[port='37300'][postgres][2018-06-27 10:37:22]: [100.00%] 1367 objects have been dumped.
vb_dump[port='37300'][postgres][2018-06-27 10:37:22]: dump database postgres successfully
vb_dump[port='37300'][postgres][2018-06-27 10:37:22]: total time: 37017 ms
```

示例 6: 执行 vb_dump, 仅导出依赖于指定表 testtable 的视图信息。然后创建新的 testtable 表, 再恢复依赖其上的视图。

- ❖ 备份仅依赖于 testtable 的视图。

```
vb_dump -s -p 37300 postgres -t PUBLIC.testtable --include-depend-objs --exclude-self -f backup/MPPDB_backup.sql -F p
vb_dump[port='37300'][postgres][2018-06-15 14:12:54]: The total objects number is 331.
vb_dump[port='37300'][postgres][2018-06-15 14:12:54]: [100.00%] 331 objects have been dumped.
vb_dump[port='37300'][postgres][2018-06-15 14:12:54]: dump database postgres successfully
vb_dump[port='37300'][postgres][2018-06-15 14:12:54]: total time: 327 ms
```

- ❖ 修改 testtable 名称。

```
vsq -p 37300 postgres -r -c "ALTER TABLE PUBLIC.testtable RENAME TO testtable_bak;"
```

- ❖ 创建新的 testtable 表。

```
CREATE TABLE PUBLIC.testtable(a int, b int, c int);
```

- ❖ 还原依赖于 testtable 的视图。

```
vsq -p 37300 postgres -r -f backup/MPPDB_backup.sql
```

相关命令

6.3.2 vb_dumpall, 6.3.3 vb_restore

6.3.2.vb_dumpall

背景信息

vb_dumpall 是 Vastbase 用于导出所有数据库相关信息工具, 它可以导出 Vastbase 数据库的所有数据, 包括默认数据库 postgres 的数据、自定义数据库的数据、以及 Vastbase 所有数据库公共的全局对象。

vb_dumpall 工具由操作系统用户 omm 执行。

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

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

vb_dumpall 在导出 Vastbase 所有数据库时分为两部分:

- ❖ vb_dumpall 自身对所有数据库公共的全局对象进行导出, 包括有关数据库用户和组, 表空间以及属性 (例如, 适用于数据库整体的访问权限) 信息。
- ❖ vb_dumpall 通过调用 vb_dump 来完成 Vastbase 中各数据库的 SQL 脚本文件导出, 该脚本文件包含将数据库恢复为其保存时的状态所需要的全部 SQL 语句。

以上两部分导出的结果为纯文本格式的 SQL 脚本文件，使用 vsql 运行该脚本文件可以恢复 Vastbase 数据库。

注意事项

- ❖ 禁止修改导出的文件和内容，否则可能无法恢复成功。
- ❖ 为了保证数据一致性和完整性，vb_dumpall 会对需要转储的表设置共享锁。如果某张表在别的事务中设置了共享锁，vb_dumpall 会等待此表的锁释放后锁定此表。如果无法在指定时间内锁定某张表，转储会失败。用户可以通过指定--lock-wait-timeout 选项，自定义等待锁超时时间。
- ❖ 由于 vb_dumpall 读取所有数据库中的表，因此必须以 Vastbase 管理员身份进行连接，才能导出完整文件。在使用 vsql 执行脚本文件导入时，同样需要管理员权限，以便添加用户和组，以及创建数据库。

语法

```
vb_dumpall [OPTION]...
```

参数说明

通用参数：

- ❖ -f, --filename=FILENAME
将输出发送至指定文件。如果这里省略，则使用标准输出。
- ❖ -v, --verbose
指定 verbose 模式。该选项将导致 vb_dumpall 向转储文件输出详细的对象注解和启动/停止次数，向标准错误流输出处理信息。
- ❖ -V, --version
打印 vb_dumpall 版本，然后退出。
- ❖ --lock-wait-timeout=TIMEOUT
请勿在转储刚开始时一直等待以获取共享表锁。如果无法在指定时间内锁定某个表，就选择失败。可以以任何符合 SET statement_timeout 的格式指定超时时间。
- ❖ -?, --help
显示 vb_dumpall 命令行参数帮助，然后退出。

转储参数：

- ❖ -a, --data-only
只转储数据，不转储模式（数据定义）。
- ❖ -c, --clean
在重新创建数据库之前，执行 SQL 语句清理（删除）这些数据库。针对角色和表空间的转储命令已添加。
- ❖ -g, --globals-only

只转储全局对象（角色和表空间），无数据库。

❖ -o, --oids

转储每个表的对象标识符（OIDs），作为表的一部分数据。该选项用于应用以某种方式参照了 OID 列的情况。如果不是以上这种情况，请勿使用该选项。

❖ -O, --no-owner

不输出设置对象的归属这样的命令，以匹配原始数据库。默认情况下，vb_dumpall 会发出 ALTER OWNER 或 SET SESSION AUTHORIZATION 语句设置所创建的模式元素的所属。如果脚本正在运行，该语句不会执行成功，除非是由系统管理员触发（或是拥有脚本中所有对象的同一个用户）。通过指定-O，编写一个任何用户都能存储的脚本，且该脚本会授予该用户拥有所有对象的权限。

❖ -r, --roles-only

只转储角色，不转储数据库或表空间。

❖ -s, --schema-only

只转储对象定义（模式），而非数据。

❖ -S, --sysadmin=NAME

在转储过程中使用的系统管理员名称。

❖ -t, --tablespaces-only

只转储表空间，不转储数据库或角色。

❖ -x, --no-privileges

防止转储访问权限（授权/撤销命令）。

❖ --column-inserts|--attribute-inserts

以 INSERT 命令带列名（INSERT INTO 表（列、...）值...）方式导出数据。这会导致恢复缓慢。但是由于该选项会针对每行生成一个独立分开的命令，所以在重新加载某行时出现的错误只会导致那行丢失，而非整个表内容。

❖ --disable-dollar-quoting

该选项将禁止在函数体前使用美元符号\$，并强制使用 SQL 标准字符串语法对其进行引用。

❖ --disable-triggers

该参数为扩展预留接口，不建议使用。

❖ --inserts

发出 INSERT 命令（而非 COPY 命令）时转储数据。这会导致恢复缓慢。注意如果重排列顺序，可能会导致恢复整个失败。--column-inserts 选项更加安全，虽然可能更慢些。

❖ --no-security-labels

该参数为扩展预留接口，不建议使用。

❖ --no-tablespaces

请勿输出创建表空间的命令，也请勿针对对象选择表空间。使用该选项，无论默认表空间是哪个，在恢复过程中所有对象都会被创建。

❖ `--no-unlogged-table-data`

该参数为扩展预留接口，不建议使用。

❖ `--quote-all-identifiers`

强制对所有标识符加引号。为了向后续版本迁移，且其中可能涉及引入额外关键词，在转储相应数据库时该选项会有帮助。

❖ `--dont-overwrite-file`

不重写当前文件。

❖ `--use-set-session-authorization`

输出符合 SQL 标准的 SET SESSION AUTHORIZATION 命令而不是 ALTER OWNER 命令来确定对象所有权。这样令转储更加符合标准，但是如果转储文件中的对象的历史有些问题，那么可能不能正确恢复。并且，使用 SET SESSION AUTHORIZATION 的转储需要数据库系统管理员的权限才能转储成功，而 ALTER OWNER 需要的权限则低得多。

❖ `--with-encryption=AES128`

指定转储数据需用 AES128 进行加密。

❖ `--with-key=KEY`

AES128 密钥长度必须是 16 字节。

❖ `--include-templatedb`

转储过程中包含模板库。

❖ `--binary-upgrade`

该参数为扩展预留接口，不建议使用。

❖ `--binary-upgrade-usermap="USER1=USER2"`

该参数为扩展预留接口，不建议使用。

❖ `--tablespaces-postfix`

该参数为扩展预留接口，不建议使用。

❖ `--parallel-jobs`

指定备份进程并发数，取值范围为 1~1000。

📖 说明

- `-g/--globals-only` 和 `-r/--roles-only` 不能同时使用。
- `-g/--globals-only` 和 `-t/--tablespaces-only` 不能同时使用。
- `-r/--roles-only` 和 `-t/--tablespaces-only` 不能同时使用。
- `-s/--schema-only` 和 `-a/--data-only` 不能同时使用。
- `-r/--roles-only` 和 `-a/--data-only` 不能同时使用。
- `-t/--tablespaces-only` 和 `-a/--data-only` 不能同时使用。
- `-g/--globals-only` 和 `-a/--data-only` 不能同时使用。

- --tablespaces-postfix 和--binary-upgrade 必须一起使用。
- --binary-upgrade-usermap 和--binary-upgrade 必须一起使用。
- --parallel-jobs 和-f/--file 必须一起使用。

连接参数：

❖ -h, --host

指定主机的名称。如果取值是以斜线开头,它将用作 Unix 域套接字的目录。默认值取自 PGHOST 环境变量；如果没有设置,将启动某个 Unix 域套接字建立连接。

该参数只针对 Vastbase 外,对 Vastbase 内本机只能用 127.0.0.1。

环境变量：PGHOST

❖ -l, --database

指定所连接的转储全局对象的数据库名称,并去寻找还有其他哪些数据库需要被转储。如果没有指定,会使用 postgres 数据库,如果 postgres 数据库不存在,会使用 template1。

❖ -p, --port

指定服务器所监听的 TCP 端口或本地 Unix 域套接字后缀,以确保连接。默认值设置为 PGPORT 环境变量。

在开启线程池情况下,建议使用 pooler port,即监听端口+1。

环境变量：PGPORT

❖ -U, --username

所连接的用户名。

环境变量：PGUSER

❖ -w, --no-password

不出现输入密码提示。如果服务器要求密码认证并且密码没有通过其它形式给出,则连接尝试将会失败。该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

❖ -W, --password

指定用户连接的密码。如果主机的认证策略是 trust,则不会对系统管理员进行密码验证,即无需输入-W 选项；如果没有-W 选项,并且不是系统管理员,“Dump Restore 工具”会提示用户输入密码。

❖ --role

指定创建转储使用的角色名。选择该选项,会使 vb_dumpall 连接数据库后,发起一个 SET ROLE 角色名命令。当所授权用户(由-U 指定)没有 vb_dumpall 要求的权限时,该选项会起到作用,即切换到具备相应权限的角色。某些安装操作规定不允许直接以系统管理员身份登录,而使用该选项能够在不违反该规定的情况下完成转储。

❖ --rolepassword

指定具体角色用户的角色密码。

说明

由于 vb_dumpall 内部调用 vb_dump，所以一些诊断信息参见 6.3.1 vb_dump。

一旦恢复，最好在每个数据库上运行 ANALYZE，优化程序提供有用的统计数据。

vb_dumpall 恢复前需要所有必要的表空间目录才能退出；否则，对于处在非默认位置的数据库，数据库创建会失败。

示例

使用 vb_dumpall 一次导出 Vastbase 的所有数据库。

📖 说明

vb_dumpall 仅支持纯文本格式导出。所以只能使用 vsql 恢复 vb_dumpall 导出的转储内容。

```
vb_dumpall -f backup/bkp2.sql -p 37300
vb_dump[port='37300'][dbname='postgres'][2018-06-27 09:55:09]: The total objects number is 2371.
vb_dump[port='37300'][dbname='postgres'][2018-06-27 09:55:35]: [100.00%] 2371 objects have been
dumped.
vb_dump[port='37300'][dbname='postgres'][2018-06-27 09:55:46]: dump database dbname='postgres'
successfully
vb_dump[port='37300'][dbname='postgres'][2018-06-27 09:55:46]: total time: 55567 ms
vb_dumpall[port='37300'][2018-06-27 09:55:46]: dumpall operation successful
vb_dumpall[port='37300'][2018-06-27 09:55:46]: total time: 56088 ms
```

相关命令

6.3.1 vb_dump, 6.3.3 vb_restore

6.3.3. vb_restore

背景信息

vb_restore 是 Vastbase 提供的针对 vb_dump 导出数据的导入工具。通过此工具可由 vb_dump 生成的导出文件进行导入。

vb_restore 工具由操作系统用户 omm 执行。

主要功能包含：

- ❖ 导入到数据库

如果连接参数中指定了数据库，则数据将被导入到指定的数据库中。其中，并行导入必须指定连接的密码。

- ❖ 导入到脚本文件

如果未指定导入数据库，则创建包含重建数据库所必须的 SQL 语句脚本并写入到文件或者标准输出。等效于直接使用 vb_dump 导出为纯文本格式。

命令格式

```
vb_restore [OPTION]... FILE
```

说明

- FILE 没有短选项或长选项。用来指定归档文件所处的位置。
- 作为前提条件，需输入 dbname 或-I 选项。不允许用户同时输入 dbname 和-I 选项。
- vb_restore 默认是以追加的方式进行数据导入。为避免多次导入造成数据异常，在进行导入时，建议使用"-c" 参数，在重新创建数据库对象前，清理（删除）已存在于将要还原的数据库中的数据库对象。

参数说明

通用参数：

- ❖ -d, --dbname=NAME

连接数据库 dbname 并直接导入到该数据库中。

- ❖ -f, --file=FILENAME

指定生成脚本的输出文件，或使用-I 时列表的输出文件。

默认是标准输出。

说明

-f 不能同-d 一起使用。

- ❖ -F, --format=c|d|t

指定归档格式。由于 vb_restore 会自动决定格式，因此不需要指定格式。

取值范围：

- c/custom：该归档形式为 4.21-vb_dump 的自定义格式。
- d/directory：该归档形式是一个目录归档形式。
- t/tar：该归档形式是一个 tar 归档形式。

- ❖ -l, --list

列出归档形式内容。这一操作的输出可用作-L 选项的输入。注意如果像-n 或-t 的过滤选项与-l 使用，过滤选项将会限制列举的项目（即归档形式内容）。

- ❖ -v, --verbose

指定 verbose 模式。

- ❖ -V, --version

打印 vb_restore 版本，然后退出。

- ❖ -?, --help

显示 vb_restore 命令行参数帮助，然后退出。

导入参数：

- ❖ -a, --data-only

只导入数据，不导入模式（数据定义）。vb_restore 的导入是以追加方式进行的。

- ❖ -c, --clean

在重新创建数据库对象前，清理（删除）已存在于将要还原的数据库中的数据库对象

❖ **-C, --create**

导入到数据库之前请创建数据库。（选择该选项后，以-d 打头的数据库将被用作发布首个 CREATE DATABASE 命令。所有数据将被导入到出现在归档文件的数据库中。）

❖ **-e, --exit-on-error**

当发送 SQL 语句到数据库时如果出现错误，请退出。默认状态下会继续，且在导入后会显示一系列错误信息。

❖ **-I, --index=NAME**

只导入已列举的 index 的定义。允许导入多个 index。如果多次输入-I index 导入多个 index。

例如：

```
vb_restore -h host_name -p port_number -d postgres -I Index1 -I Index2 backup/MPPDB_backup.tar
```

在上面这个例子中，Index1 和 Index2 会被导入。

❖ **-j, --jobs=NUM**

运行 vb_restore 最耗时的部分（如加载数据、创建 index、或创建约束）使用并发任务。该选项能大幅缩短导入时间，即将一个大型数据库导入到某一多处理器的服务器上。

每个任务可能是一个进程或一个线程，这由操作系统决定；每个任务与服务器进行单独连接。

该选项的最优值取决于服务器的硬件设置、客户端、以及网络。还包括这些因素，如 CPU 核数量、硬盘设置。建议是从增加服务器上的 CPU 核数量入手，更大的值（服务器上 CPU 核数量）在很多情况下也能导致数据文件更快的被导入。当然，过高的值会由于超负荷反而导致性能降低。

该选项只支持自定义归档格式。输入文件必须是常规文件（不能是像 pipe 的文件）。如果是通过脚本文件，而非直接连接数据库服务器，该选项可忽略。而且，多任务不能与--single-transaction 选项一起使用。

❖ **-L, --use-list=FILENAME**

只导入列举在 list-file 中的那些归档形式元素，导入顺序以它们在文件中的顺序为准。注意如果像-n 或-t 的过滤选项与-L 使用，它们将会进一步限制导入的项目。

一般情况下，list-file 是通过编辑前面提到的某个-I 参数的输出创建的。文件行的位置可更改或直接删除行，也可使用分号 (;) 在行的开始注出。见下文的举例。

❖ **-n, --schema=NAME**

只导入已列举的模式中的对象。

该选项可与-t 选项一起用以导入某个指定的表。

多次输入-n schemaname 可以导入多个模式。

例如：

```
vb_restore -h host_name -p port_number -d postgres -n sch1 -n sch2 backup/MPPDB_backup.tar
```

在上面这个例子中，sch1 和 sch2 会被导入。

❖ -O, --no-owner

不输出设置对象的归属这样的命令，以匹配原始数据库。默认情况下，vb_restore 会发出 ALTER OWNER 或 SET SESSION AUTHORIZATION 语句设置所创建的模式元素的所属。除非是由系统管理员（或是拥有脚本中所有对象的同一个用户）进行数据库首次连接的操作，否则语句会失败。使用-O 选项，任何用户名都可用于首次连接，且该用户拥有所有已创建的对象。

❖ -P, --function=NAME(args)

只导入已列举的函数。请按照函数所在转储文件中的目录，准确拼写函数名称和参数。

当-P 单独使用时，表示导入文件中所有'function-name(args)'函数；当-P 同-n 一起使用时，表示导入指定模式下的'function-name(args)'函数；多次输入-P，而仅指定一次-n，表示所有导入的函数默认都是位于-n 模式下的。

可以多次输入-n schema-name -P 'function-name(args)'同时导入多个指定模式下的函数。

例如：

```
./vb_restore -h host_name -p port_number -d postgres -n test1 -P 'Func1(integer)' -n test2 -P 'Func2(integer)' backup/MPPDB_backup.tar
```

在上面这个例子中，test1 模式下的函数 Func1(i integer)和 test2 模式下的函数 Func2(j integer)会被一起导入。

❖ -s, --schema-only

只导入模式（数据定义），不导入数据（表内容）。当前的序列值也不会导入。

❖ -S, --sysadmin=NAME

该参数为扩展预留接口，不建议使用。

❖ -t, --table=NAME

只导入已列举的表定义、数据或定义和数据。该选项与-n 选项同时使用时，用来指定某个模式下的表对象。-n 参数不输入时，默认为 PUBLIC 模式。多次输入-n <schemaname> -t <tablename> 可以导入指定模式下的多个表。

例如：

导入 PUBLIC 模式下的 table1

```
vb_restore -h host_name -p port_number -d postgres -t table1 backup/MPPDB_backup.tar
```

导入 test1 模式下的 test1 和 test2 模式下 test2

```
vb_restore -h host_name -p port_number -d postgres -n test1 -t test1 -n test2 -t test2 backup/MPPDB_backup.tar
```

导入 PUBLIC 模式下的 table1 和 test1 模式下 test1

```
vb_restore -h host_name -p port_number -d postgres -n PUBLIC -t table1 -n test1 -t table1 backup/MPPDB_backup.tar
```

须知

-t 不支持 schema_name.table_name 的输入格式。

- ❖ -T, --trigger=NAME
该参数为扩展预留接口。
- ❖ -x, --no-privileges/--no-acl
防止导入访问权限 (grant/revoke 命令)。
- ❖ -1, --single-transaction
执行导入作为一个单独事务 (即把命令包围在 BEGIN/COMMIT 中)。
该选项确保要么所有命令成功完成, 要么没有改变应用。该选项意为--exit-on-error。
- ❖ --disable-triggers
该参数为扩展预留接口, 不建议使用。
- ❖ --no-data-for-failed-tables
默认状态下, 即使创建表的命令失败 (如表已经存在), 表数据仍会被导入。使用该选项, 像这种表的数据会被跳过。如果目标数据库已包含想要的表内容, 这种行为会有帮助。
该选项只有在直接导入到某数据库中时有效, 不针对生成 SQL 脚本文件输出。
- ❖ --no-security-labels
该参数为扩展预留接口, 不建议使用。
- ❖ --no-tablespaces
不输出选择表空间的命令。使用该选项, 无论默认表空间是哪个, 在导入过程中所有对象都会被创建。
- ❖ --section=SECTION
导入已列举的区段 (如 pre-data、data、或 post-data)。
- ❖ --use-set-session-authorization
该选项用来进行文本格式的备份。
输出 SET SESSION AUTHORIZATION 命令, 而非 ALTER OWNER 命令, 用以决定对象归属。
该选项使转储更加兼容标准, 但通过参考转储中对象的记录, 导入过程可能会有问题。使用 SET SESSION AUTHORIZATION 的转储要求必须是系统管理员, 同时在导入前还需参考"SET SESSION AUTHORIZATION", 手工对导出文件的密码进行修改验证, 只有这样才能进行正确的导入操作, 相比之下, ALTER OWNER 对权限要求较低。
- ❖ --with-key=KEY
AES128 密钥长度必须是 16 字节。

📖 说明

如果转储被加密, 则必须在 vb_restore 命令中输入--with-key <keyname>选项。如果未输入, 用户会收到错误信息。应该输入转储时所输入的相同的 key。

转储格式为 c 或 t 时，转储内容已被处理，因此转储格式为 c 或 t 是输入不受加密限制。

须知

- 如果安装过程中有任何本地数据要添加到 template1 数据库，请谨慎将 vb_restore 的输出载入到一个真正的空数据库中；否则可能会因为被添加对象的定义被复制，而出现错误。要创建一个无本地添加的空数据库，需从 template0 而非 template1 复制，例如：

```
CREATE DATABASE foo WITH TEMPLATE template0;
```

- vb_restore 不能选择性地导入大对象；例如只能导入那些指定表的对象。如果某个归档形式包含大对象，那所有大对象都会被导入，或一个都不会被导入，如果它们通过 -L、-t 或其他选项被排除。

说明

1. -d/--dbname 和 -f/--file 不能同时使用；
2. -s/--schema-only 和 -a/--data-only 不能同时使用；
3. -c/--clean 和 -a/--data-only 不能同时使用；
4. 使用 --single-transaction 时，-j/--jobs 必须为单任务；
5. --role 和 --rolepassword 必须一起使用。

连接参数：

- ❖ -h, --host=HOSTNAME

指定的主机名称。如果取值是以斜线开头，他将用作 Unix 域套接字的目录。默认值取自 PGHOST 环境变量；如果没有设置，将启动某个 Unix 域套接字建立连接。

该参数只针对 Vastbase 外，对 Vastbase 内本机只能用 127.0.0.1。

- ❖ -p, --port=PORT

指定服务器所监听的 TCP 端口或本地 Unix 域套接字后缀，以确保连接。默认值设置为 PGPORT 环境变量。

在开启线程池情况下，建议使用 pooler port，即监听端口+1。

- ❖ -U, --username=NAME

所连接的用户名。

- ❖ -w, --no-password

不出现输入密码提示。如果服务器要求密码认证并且密码没有通过其它形式给出，则连接尝试将会失败。该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

- ❖ -W, --password=PASSWORD

指定用户连接的密码。如果主机的认证策略是 trust，则不会对系统管理员进行密码验证，即无需输入 -W 参数；如果没有 -W 参数，并且不是系统管理员，“vb_restore”会提示用户输入密码。

- ❖ --role=ROLENAME

指定导入操作使用的角色名。选择该参数，会使 vb_restore 连接数据库后，发起一个 SET ROLE 角色名命令。当所授权用户（由 -U 指定）没有 vb_restore 要求的权限时，该参数会起到作用，

即切换到具备相应权限的角色。某些安装操作规定不允许直接以初始用户身份登录，而使用该参数能够在不违反该规定的情况下完成导入。

- ❖ `--rolepassword=ROLEPASSWORD`
指定具体角色用户的角色密码。

示例

特例：执行 `vsq1` 程序，使用如下选项导入由 `vb_dump/vb_dumpall` 生成导出文件夹（纯文本格式）的 `MPPDB_backup.sql` 文件到 `postgres` 数据库。

```
vsq1 -d postgres -p 8000 -W Bigdata@123 -f /home/omm/test/MPPDB_backup.sql
SET
SET
SET
SET
SET
SET
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
CREATE INDEX
CREATE INDEX
CREATE INDEX
SET
CREATE INDEX
REVOKE
REVOKE
GRANT
GRANT
total time: 30476 ms
```

`vb_restore` 用来导入由 `vb_dump` 生成的导出文件。

示例 1：执行 `vb_restore`，将导出的 `MPPDB_backup.dmp` 文件（自定义归档格式）导入到 `postgres` 数据库。

```
vb_restore -W Bigdata@123 backup/MPPDB_backup.dmp -p 8000 -d postgres
vb_restore: restore operation successful
vb_restore: total time: 13053 ms
```

示例 2：执行 `vb_restore`，将导出的 `MPPDB_backup.tar` 文件（tar 格式）导入到 `postgres` 数据库。

```
vb_restore backup/MPPDB_backup.tar -p 8000 -d postgres
vb_restore[2017-07-21 19:16:26]: restore operation successful
vb_restore[2017-07-21 19:16:26]: total time: 21203 ms
```

示例 3：执行 `vb_restore`，将导出的 `MPPDB_backup` 文件（目录格式）导入到 `postgres` 数据库。

```
vb_restore backup/MPPDB_backup -p 8000 -d postgres
vb_restore[2017-07-21 19:16:26]: restore operation successful
vb_restore[2017-07-21 19:16:26]: total time: 21003 ms
```

示例 4：执行 `vb_restore`，使用自定义归档格式的 `MPPDB_backup.dmp` 文件来进行如下导入操作。导入 `PUBLIC` 模式下所有对象的定义和数据。在导入时会先删除已经存在的对象，如果原对象存在跨模式的依赖则需手工强制干预。

```
vb_restore backup/MPPDB_backup.dmp -p 8000 -d postgres -e -c -n PUBLIC
vb_restore: [archiver (db)] Error while PROCESSING TOC:
vb_restore: [archiver (db)] Error from TOC entry 313; 1259 337399 TABLE table1 vastbasea
vb_restore: [archiver (db)] could not execute query: ERROR: cannot drop table table1 because other
objects depend on it
DETAIL: view t1.v1 depends on table table1
HINT: Use DROP ... CASCADE to drop the dependent objects too.
Command was: DROP TABLE public.table1;
```

手工删除依赖，导入完成后再重新创建。

```
vb_restore backup/MPPDB_backup.dmp -p 8000 -d postgres -e -c -n PUBLIC
vb_restore[2017-07-21 19:16:26]: restore operation successful
vb_restore[2017-07-21 19:16:26]: total time: 2203 ms
```

示例 5: 执行 vb_restore, 使用自定义归档格式的 MPPDB_backup.dmp 文件来进行如下导入操作。
只导入 PUBLIC 模式下表 table1 的定义。

```
vb_restore backup/MPPDB_backup.dmp -p 8000 -d postgres -e -c -s -n PUBLIC -t table1
vb_restore[2017-07-21 19:16:26]: restore operation successful
vb_restore[2017-07-21 19:16:26]: total time: 21000 ms
```

示例 6: 执行 vb_restore, 使用自定义归档格式的 MPPDB_backup.dmp 文件来进行如下导入操作。
只导入 PUBLIC 模式下表 table1 的数据。

```
vb_restore backup/MPPDB_backup.dmp -p 8000 -d postgres -e -a -n PUBLIC -t table1
vb_restore[2017-07-21 19:16:26]: restore operation successful
vb_restore[2017-07-21 19:16:26]: total time: 20203 ms
```

相关命令

6.3.1 vb_dump, 6.3.2 vb_dumpall

7. 高危操作一览表

各项操作请严格遵守指导书操作，同时如下高危操作禁止随意执行。

表 7-1 中描述在产品的操作与维护阶段，进行日常操作时应注意的严禁操作。

表 1-10 禁用操作

操作名称	操作风险
严禁修改数据目录下文件名，权限，内容不能修改，不能删除内容。	导致数据库节点实例出现严重错误，并且无法修复。
严禁删除数据库系统表或系统表数据。	删除系统表将导致无法正常进行业务操作。

表 7-2 中描述在产品的操作与维护阶段，进行日常操作时应注意的高危操作。

表 1-11 高危操作

操作分类	操作名称	操作风险	风险等级	规避措施	重大操作观察项目
数据库	不能直接在配置文件中手动修改端口号。	导致数据库启动不了或者连接不上。	▲▲▲▲▲	尽量使用工具修改，不要手动操作	无。
	不能随意修改 pg_hba.conf 配置文件中的内容。	导致客户端连接不上。	▲▲▲▲▲	严格根据产品手册操作。	无。
	不能手动修改 pg_xlog 的内容。	导致数据库无法启动，数据不一致。	▲▲▲▲▲	尽量使用工具修改，不要手动操作。	无。
作业	使用 kill -9 终止作业进程。	导致作业占用的系统资源无法释放。	▲▲▲	尽量登录数据库使用 pg_terminate_backend, pg_cancel_backend 操作终止作业，或使用 Ctrl+C 终止作业进程。	观察资源使用情况。

8. 日志参考

[8.1 日志类型简介](#)

[8.2 系统日志](#)

[8.3 操作日志](#)

[8.4 审计日志](#)

[8.5 WAL 日志](#)

[8.6 性能日志](#)

8.1. 日志类型简介

在数据库运行过程中，会出现大量日志，既有保证数据库安全可靠的 WAL 日志（预写式日志，也称为 Xlog），也有用于数据库日常维护的运行和操作日志等。在数据库发生故障时，可以参考这些日志进行问题定位和数据库恢复的操作。

日志类型

日志类型的详细说明请参见下表。

表 1-12 日志类型

类型	说明
系统日志	数据库系统进程运行时产生的日志，记录系统进程的异常信息。
操作日志	通过客户端工具（例如 vb_guc）操作数据库时产生的日志。
Trace 日志	打开数据库的调试开关后，会记录大量的 Trace 日志。这些日志可以用来分析数据库的异常信息。
黑匣子日志	数据库系统崩溃的时候，通过故障现场堆、栈信息可以分析出故障发生时的进程上下文，方便故障定位。黑匣子具有在系统崩溃时，dump 出进程和线程的堆、栈、寄存器信息的功能。
审计日志	开启数据库审计功能后，将数据库用户的某些操作记录在日志中，这些日志称为审计日志。
WAL 日志	又称为 REDO 日志，在数据库异常损坏时，可以利用 WAL 日志进行恢复。由于 WAL 日志的重要性，所以需要经常备份这些日志。
性能日志	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

类型	说明
	测，包括磁盘、OBS、HadoopVastbase 等外部资源的访问检测信息。

8.2. 系统日志

Vastbase 运行时数据库节点以及 Vastbase 安装部署时产生的日志统称为系统日志。如果 Vastbase 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 Vastbase 的方法。

日志文件存储路径

数据库节点的运行日志放在“/var/log/vastbase/用户名/pg_log”中各自对应的目录下。
OMVastbase 安装卸载时产生的日志放在“/var/log/vastbase/用户名/om”目录下。

日志文件命名格式

数据库节点运行日志的命名规则：
postgresql-创建时间.log

默认情况下，每日 0 点或者日志文件大于 16MB 或者数据库实例（数据库节点）重新启动后，会生成新的日志文件。

日志内容说明

❖ 数据库节点每一行日志内容的默认格式：

日期+时间+时区+用户名称+数据库名称+会话 ID+日志级别+日志内容

8.3. 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 Vastbase 调用时产生的日志。如果 Vastbase 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。

日志文件存储路径

默认在“\$GAUSSLOG/bin”目录下，如果环境变量\$GAUSSLOG 不存在或者变量值为空，则工具日志信息不会记录到对应的工具日志文件中，日志信息只会打印到屏幕上。

其中\$GAUSSLOG 默认为“/var/log/vastbase/用户名”。

📖 说明

如果使用 om 脚本部署时，则日志路径为 “/var/log/vastbase/用户名”。

日志文件命名格式

日志文件命名格式为：

- ❖ 工具名-日志创建时间.log
- ❖ 工具名-日志创建时间-current.log

其中，“工具名-日志创建时间.log”是历史日志文件，“工具名-日志创建时间-current.log”是当前日志文件。

如果日志大小超过 16MB，在下次调用该工具时，会重命名当前日志文件为历史日志文件，并以当前时间生成新的当前日志文件。

例如将“vb_guc-2015-01-16_183728-current.log”重命名为“vb_guc-2015-01-16_183728.log”，然后重新生成“vb_guc-2015-01-17_142216-current.log”。

维护建议

建议定时对过期的日志文件进行转储，以避免大量日志占用太多的磁盘空间和避免重要日志丢失。

8.4. 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。

关于如何设置审计日志维护策略请参见《开发者指南》中“管理数据库安全 > 设置数据库审计 > 维护审计日志”章节。

8.5. 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 Vastbase 进行恢复操作。

日志文件存储路径

以一个数据库节点为例，默认在“/vastbase/data/data_dn/pg_xlog”目录下。

其中“/vastbase/data/data_dn”代表 Vastbase 节点的数据目录。

日志文件命名格式

日志文件以段文件的形式存储的，每个段为 16MB，并分割成若干页，每页 8KB。对 WAL 日志的命名说明如下：一个段文件的名称由 24 个十六进制组成，分为三个部分，每个部分由 8 个十六进制字符组成。第一部分表示时间线，第二部分表示日志文件标号，第三部分表示日志文件的段标号。时间线由 1 开始，日志文件标号和日志文件的段标号由 0 开始。

例如，系统中的第一个事务日志文件是 000000010000000000000000。

📖 说明

这些数字一般情况下是顺序增长使用的（要把所有可用数字都用光也需要非常长的时间），但也存在循环使用的情况。

日志内容说明

WAL 日志的内容取决于记录事务的类型，在系统崩溃时可以利用 WAL 日志进行恢复。

默认配置下，Vastbase 每次启动时会先读取 WAL 日志进行恢复。

维护建议

WAL 日志对数据库异常恢复有重要的作用，建议定期对 WAL 日志进行备份。

8.6. 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、HadoopVastbase 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。

日志文件存储路径

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。

日志文件命名格式

数据库节点的性能日志的命名规则：

postgresql-创建时间.prf

默认情况下，每日 0 点或者日志文件大于 20MB 或者数据库实例（数据库节点）重新启动后，会生成新的日志文件。

日志内容说明

数据库节点每一行日志内容的默认格式：

主机名称+日期+时间+实例名称+线程号+日志内容



电话: 010-82838118

地址: 北京市海淀区学院路 30 号科大天工大厦 B 座 6 层

官网: www.vastdata.com.cn