**图表

描述已自动生成**

图片包含 游戏机, 盘子, 食物, 画

描述已自动生成

北京海量数据技术股份有限公司

202X年XX月

**【客户名称】**

**PG数据库巡检报告**

**修改历史**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **日期** | **版本号** | **作者** | | **修改说明** | | **更改请求号** |
| 2021/11/16 | 1.0 | 何俊庆 | 新建 | |  | |
|  |  |  | |  | |  |
|  |  |  | |  | |  |

注释：“变更请求号”为文档正式发布后需要变更时的编号。

**审阅**

|  |  |  |
| --- | --- | --- |
| **姓名** | **日期** | **备注** |
|  |  |  |
|  |  |  |
|  |  |  |

**分发**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **公司** | | **姓名** | | **日期** | | **备注** |
| 海量数据 | ApolloVB | | 2021/12/16 | |  | |
|  | |  | |  | |  |
|  | |  | |  | |  |

**目 录**

[1 概述及总结 4](#_Toc26696)

[1.1 概要 4](#_Toc27027)

[1.2 巡检范围 4](#_Toc14895)

[1.3 总结建议 4](#_Toc14476)

[2 VastbaseE100数据库巡检 5](#_Toc25554)

[2.1 操作系统分析结果 5](#_Toc18969)

[2.1.1 操作系统配置信息 5](#_Toc3435)

[2.1.2 selinux动态配置信息 5](#_Toc5557)

[2.1.3 Transparent Huge Pages 5](#_Toc10246)

[2.2 数据库运行结果 6](#_Toc18865)

[2.2.1 配置文件检查 6](#_Toc3144)

[2.2.2 数据库定制参数 7](#_Toc31086)

[2.2.3 数据库错误日志分析 8](#_Toc20525)

[2.2.4 数据库空间使用分析 8](#_Toc31374)

[2.2.5 数据库连接分析 9](#_Toc5119)

[2.2.6 数据库性能分析 10](#_Toc20638)

[2.2.7 数据库垃圾分析 11](#_Toc18397)

[2.2.8 数据库年龄分析 11](#_Toc17147)

[2.2.9 数据库归档与流复制 12](#_Toc13951)

[2.2.10 数据库风险分析 12](#_Toc22281)

[2.2.11 重置统计信息 14](#_Toc18207)

# 概述及总结

## 概要

数据库在整个业务系统中处于非常核心的地位，数据库的性能好坏将直接影响到整个应用系统的性能。但往往在应用系统实际运行过程中，由于系统数据量的增加、业务模块应用逻辑的修改、应用版本变更、用户量增长等各种原因都可能使数据库性能下降，从而导致应用系统性能下降，影响用户感知。因此需要定期对数据库进行全面的性能分析，分析性能变化趋势及产生这些变化的原因，以便做出优化调整，保持应用系统良好的性能。

## 巡检范围

|  |  |  |
| --- | --- | --- |
| **序号** | **数据库** | **数据库版本** |
| 1 | PG | 11 |

## 总结建议

|  |  |  |
| --- | --- | --- |
| **业务系统** | **分析项目** | **调整建议** |
| **XXX管理系统** | 操作系统配置文件、静态配置信息 |  |
| Transparent Huge Pages |  |
| 数据库配置文件检查 |  |
| 数据库性能分析 |  |
| 数据库垃圾分析 |  |

# Postgresql数据库巡检

## 操作系统分析结果

|  |  |  |
| --- | --- | --- |
| **栏目** | **状态** | **调整建议** |
| **操作系统配置文件、静态配置信息** |  |  |
| **selinux 动态配置信息** |  |  |
| **Transparent Huge Pages** |  |  |

* + 1. **操作系统配置信息**

----->>>---->>> 操作系统配置文件、静态配置信息:

VM-16-10-centos

----->>>---->>> 以太链路信息:   
1: lo: ,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000  
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
2: eth0: ,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000  
 link/ether 52:54:00:74:7d:2d brd ff:ff:ff:ff:ff:ff

----->>>---->>> IP地址信息:   
1: lo: ,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
 inet 127.0.0.1/8 scope host lo  
 valid\_lft forever preferred\_lft forever  
 inet6 ::1/128 scope host   
 valid\_lft forever preferred\_lft forever  
2: eth0: ,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc mq state UP group default qlen 1000  
 link/ether 52:54:00:74:7d:2d brd ff:ff:ff:ff:ff:ff  
 inet 10.0.16.10/22 brd 10.0.19.255 scope global eth0  
 valid\_lft forever preferred\_lft forever  
 inet6 fe80::5054:ff:fe74:7d2d/64 scope link   
 valid\_lft forever preferred\_lft forever

----->>>---->>> 路由信息:   
default via 10.0.16.1 dev eth0   
10.0.16.0/22 dev eth0 proto kernel scope link src 10.0.16.10   
169.254.0.0/16 dev eth0 scope link metric 1002

----->>>---->>> 操作系统内核:   
Linux VM-16-10-centos 3.10.0-1160.11.1.el7.x86\_64 #1 SMP Fri Dec 18 16:34:56 UTC 2020 x86\_64 x86\_64 x86\_64 GNU/Linux

----->>>---->>> (MB):   
 total used free shared buff/cache available  
Mem: 3789 723 292 1548 2773 921  
Swap: 0 0 0

----->>>---->>> CPU:   
Architecture: x86\_64  
CPU op-mode(s): 32-bit, 64-bit  
Byte Order: Little Endian  
CPU(s): 2  
On-line CPU(s) list: 0,1  
Thread(s) per core: 1  
Core(s) per socket: 2  
Socket(s): 1  
NUMA node(s): 1  
Vendor ID: AuthenticAMD  
CPU family: 23  
Model: 49  
Model name: AMD EPYC 7K62 48-Core Processor  
Stepping: 0  
CPU MHz: 2595.124  
BogoMIPS: 5190.24  
Hypervisor vendor: KVM  
Virtualization type: full  
L1d cache: 32K  
L1i cache: 32K  
L2 cache: 4096K  
L3 cache: 16384K  
NUMA node0 CPU(s): 0,1  
Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr\_opt pdpe1gb rdtscp lm art rep\_good nopl extd\_apicid eagerfpu pni pclmulqdq ssse3 fma cx16 sse4\_1 sse4\_2 x2apic movbe popcnt aes xsave avx f16c rdrand hypervisor lahf\_lm cmp\_legacy cr8\_legacy abm sse4a misalignsse 3dnowprefetch osvw topoext retpoline\_amd ibpb vmmcall fsgsbase bmi1 avx2 smep bmi2 rdseed adx smap clflushopt xsaveopt xsavec xgetbv1 arat

----->>>---->>> 块设备:   
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT  
sr0 11:0 1 146.8M 0 rom   
vda 253:0 0 80G 0 disk   
└─vda1 253:1 0 80G 0 part /

----->>>---->>> 拓扑:

----->>>---->>> 进程树:   
systemd,1 --switched-root --system --deserialize 22  
 ├─systemd-journal,379  
 ├─lvmetad,405 -f  
 ├─systemd-udevd,409  
 ├─auditd,601  
 │ └─{auditd},602  
 ├─dbus-daemon,659,dbus --system --address=systemd: --nofork --nopidfile --systemd-activation  
 ├─ntpd,672,ntp -u ntp:ntp -g  
 ├─acpid,673  
 ├─polkitd,676,polkitd --no-debug  
 │ ├─{polkitd},681  
 │ ├─{polkitd},682  
 │ ├─{polkitd},683  
 │ ├─{polkitd},684  
 │ ├─{polkitd},685  
 │ └─{polkitd},686  
 ├─lsmd,677,libstoragemgmt -d  
 ├─dhclient,1066 -q -lf /var/lib/dhclient/dhclient--eth0.lease -pf /var/run/dhclient-eth0.pid -H VM-16-10-centos eth0  
 ├─iscsid,1131 -f  
 ├─tuned,1135 -Es /usr/sbin/tuned -l -P  
 │ ├─{tuned},1636  
 │ ├─{tuned},1637  
 │ ├─{tuned},1665  
 │ └─{tuned},1669  
 ├─rshim,1139  
 │ └─{rshim},1150  
 ├─watchdog.sh,1162 /usr/local/sa/agent/watchdog.sh  
 │ └─sleep,30093 60  
 ├─secu-tcs-agent,1184  
 ├─rsyslogd,1404 -n  
 │ ├─{rsyslogd},1421  
 │ └─{rsyslogd},1423  
 ├─atd,1413 -f  
 ├─agetty,1517 --noclear tty1 linux  
 ├─agetty,1519 --keep-baud 115200,38400,9600 ttyS0 vt220  
 ├─master,1725 -w  
 │ ├─qmgr,1747,postfix -l -t unix -u  
 │ ├─cleanup,11730,postfix -z -t unix -u  
 │ ├─local,11731,postfix -t unix  
 │ ├─trivial-rewrite,23336,postfix -n rewrite -t unix -u  
 │ ├─pickup,23849,postfix -l -t unix -u  
 │ └─bounce,25892,postfix -z -t unix -u  
 ├─sshd,2111 -D  
 │ ├─sshd,14292  
 │ │ └─bash,14294  
 │ │ └─pg\_all.sh,30649 ./pg\_all.sh /home/postgres/local /home/postgres/data 15432  
 │ │ └─pstree,30685 -a -A -c -l -n -p -u -U -Z  
 │ └─sshd,14337  
 │ └─bash,14339  
 ├─crond,4804 -n  
 ├─sgagent,8827 -d  
 │ └─{sgagent},8833  
 ├─barad\_agent,8929  
 │ ├─barad\_agent,8935  
 │ └─barad\_agent,8936  
 │ ├─{barad\_agent},8954  
 │ ├─{barad\_agent},8960  
 │ └─{barad\_agent},30644  
 ├─systemd-logind,14757  
 ├─vastbase,15709,vastbase  
 │ ├─{vastbase},300  
 │ ├─{vastbase},2383  
 │ ├─{vastbase},7720  
 │ ├─{vastbase},13323  
 │ ├─{vastbase},15710  
 │ ├─{vastbase},15719  
 │ ├─{vastbase},15720  
 │ ├─{vastbase},15721  
 │ ├─{vastbase},15722  
 │ ├─{vastbase},15724  
 │ ├─{vastbase},15726  
 │ ├─{vastbase},15728  
 │ ├─{vastbase},15735  
 │ ├─{vastbase},15738  
 │ ├─{vastbase},15739  
 │ ├─{vastbase},15740  
 │ ├─{vastbase},15744  
 │ ├─{vastbase},15745  
 │ ├─{vastbase},15751  
 │ ├─{vastbase},15769  
 │ ├─{vastbase},15770  
 │ ├─{vastbase},15772  
 │ ├─{vastbase},15773  
 │ ├─{vastbase},15774  
 │ ├─{vastbase},15775  
 │ ├─{vastbase},15776  
 │ ├─{vastbase},15777  
 │ ├─{vastbase},15778  
 │ ├─{vastbase},15779  
 │ ├─{vastbase},15780  
 │ ├─{vastbase},15781  
 │ ├─{vastbase},15782  
 │ ├─{vastbase},15783  
 │ ├─{vastbase},15784  
 │ ├─{vastbase},15785  
 │ ├─{vastbase},15786  
 │ ├─{vastbase},15787  
 │ ├─{vastbase},15788  
 │ ├─{vastbase},23410  
 │ ├─{vastbase},24586  
 │ ├─{vastbase},24587  
 │ ├─{vastbase},24955  
 │ ├─{vastbase},28246  
 │ ├─{vastbase},29974  
 │ └─{vastbase},31711  
 └─postgres,16655,postgres  
 ├─postgres,16656  
 ├─postgres,16658  
 ├─postgres,16659  
 ├─postgres,16660  
 ├─postgres,16661  
 ├─postgres,16662  
 └─postgres,16663

----->>>---->>> /etc/sysctl.conf   
net.ipv4.ip\_forward = 0  
net.ipv4.conf.default.rp\_filter = 1  
net.ipv4.conf.default.accept\_source\_route = 0  
kernel.core\_uses\_pid = 1  
net.ipv4.tcp\_syncookies = 1  
kernel.msgmnb = 65536  
kernel.msgmax = 65536  
net.ipv4.conf.all.promote\_secondaries = 1  
net.ipv4.conf.default.promote\_secondaries = 1  
net.ipv6.neigh.default.gc\_thresh3 = 4096   
net.ipv4.neigh.default.gc\_thresh3 = 4096  
kernel.softlockup\_panic = 1  
kernel.sysrq = 1  
net.ipv6.conf.all.disable\_ipv6=0  
net.ipv6.conf.default.disable\_ipv6=0  
net.ipv6.conf.lo.disable\_ipv6=0  
kernel.numa\_balancing = 0  
net.ipv4.tcp\_max\_tw\_buckets = 10000  
net.ipv4.tcp\_tw\_reuse = 1  
net.ipv4.tcp\_tw\_recycle = 1  
net.ipv4.tcp\_keepalive\_time = 30  
net.ipv4.tcp\_keepalive\_intvl = 30  
net.ipv4.tcp\_retries2 = 12  
net.ipv4.ip\_local\_reserved\_ports = 15400-15407,20050-20057  
net.core.wmem\_max = 21299200  
net.core.rmem\_max = 21299200  
net.core.wmem\_default = 21299200  
net.core.rmem\_default = 21299200  
kernel.sem = 250 6400000 1000 25600  
net.ipv4.tcp\_rmem = 8192 250000 16777216  
net.ipv4.tcp\_wmem = 8192 250000 16777216  
vm.min\_free\_kbytes = 194009  
net.core.netdev\_max\_backlog = 65535  
net.ipv4.tcp\_max\_syn\_backlog = 65535  
net.core.somaxconn = 65535  
kernel.shmall = 1152921504606846720  
kernel.shmmax = 18446744073709551615

----->>>---->>> /etc/security/limits.conf   
\* soft nofile 100001  
\* hard nofile 100002  
root soft nofile 100001  
root hard nofile 100002  
\* soft nofile 100001  
\* hard nofile 100002  
root soft nofile 100001  
root hard nofile 100002  
\* soft memlock unlimited  
\* hard memlock unlimited  
root soft as unlimited  
omm soft as unlimited  
root hard as unlimited  
omm hard as unlimited  
root soft nproc unlimited  
omm soft nproc unlimited  
root hard nproc unlimited  
omm hard nproc unlimited

----->>>---->>> /etc/security/limits.d/\*.conf   
/etc/security/limits.d/20-nproc.conf :   
root soft nofile 1000000  
omm soft nofile 1000000  
root hard nofile 1000000  
omm hard nofile 1000000  
root soft nproc unlimited  
omm soft nproc unlimited  
root hard nproc unlimited  
omm hard nproc unlimited  
/etc/security/limits.d/90-nofile.conf :   
root soft nofile 1000000  
omm soft nofile 1000000  
root hard nofile 1000000  
omm hard nofile 1000000  
root soft nproc unlimited  
omm soft nproc unlimited  
root hard nproc unlimited  
omm hard nproc unlimited

----->>>---->>> /etc/sysconfig/iptables

----->>>---->>> /etc/fstab   
  
#  
# /etc/fstab  
# Created by anaconda on Thu Mar 7 06:38:37 2019  
#  
# Accessible filesystems, by reference, are maintained under '/dev/disk'  
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info  
#  
UUID=4b499d76-769a-40a0-93dc-4a31a59add28 / ext4 defaults 1 1

----->>>---->>> /etc/rc.local   
#!/bin/bash  
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES  
#  
# It is highly advisable to create own systemd services or udev rules  
# to run scripts during boot instead of using this file.  
#  
# In contrast to previous versions due to parallel execution during boot  
# this script will NOT be run after all other services.  
#  
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure  
# that this script will be executed during boot.  
  
touch /var/lock/subsys/local  
# secu-tcs-agent bootstart, install at Wed Oct 27 21:33:09 CST 2021  
/usr/local/sa/agent/secu-tcs-agent-mon-safe.sh > /dev/null 2>1  
/usr/local/qcloud/irq/net\_smp\_affinity.sh >/tmp/net\_affinity.log 2>1  
/usr/local/qcloud/cpuidle/cpuidle\_support.sh > /tmp/cpuidle\_support.log  
/usr/local/qcloud/rps/set\_rps.sh >/tmp/setRps.log 2>1  
/usr/local/qcloud/irq/virtio\_blk\_smp\_affinity.sh > /tmp/virtio\_blk\_affinity.log 2>1

----->>>---->>> /boot/grub/grub.conf

----->>>---->>> /var/spool/cron 用户cron配置   
/var/spool/cron/omm :   
  
/var/spool/cron/root :   
# secu-tcs-agent monitor, install at Wed Oct 27 21:33:09 CST 2021  
\* \* \* \* \* /usr/local/sa/agent/secu-tcs-agent-mon-safe.sh > /dev/null 2>1  
\*/5 \* \* \* \* flock -xn /tmp/stargate.lock -c '/usr/local/qcloud/stargate/admin/start.sh > /dev/null 2>1 '  
/var/spool/cron/vastbase :   
\*/1 \* \* \* \* sh /home/vastbase/moitor.sh >> /home/vastbase/moitor.log

----->>>---->>> chkconfig --list   
netconsole 0:off 1:off 2:off 3:off 4:off 5:off 6:off  
network 0:off 1:off 2:on 3:on 4:on 5:on 6:off  
opensmd 0:off 1:off 2:off 3:off 4:off 5:off 6:off

----->>>---->>> iptables -L -v -n -t filter 动态配置信息:   
Chain INPUT (policy ACCEPT 5872K packets, 386M bytes)  
 pkts bytes target prot opt in out source destination   
  
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)  
 pkts bytes target prot opt in out source destination   
  
Chain OUTPUT (policy ACCEPT 5769K packets, 358M bytes)  
 pkts bytes target prot opt in out source destination

----->>>---->>> iptables -L -v -n -t nat 动态配置信息:   
Chain PREROUTING (policy ACCEPT 5019K packets, 316M bytes)  
 pkts bytes target prot opt in out source destination   
  
Chain INPUT (policy ACCEPT 5019K packets, 316M bytes)  
 pkts bytes target prot opt in out source destination   
  
Chain OUTPUT (policy ACCEPT 148K packets, 9297K bytes)  
 pkts bytes target prot opt in out source destination   
  
Chain POSTROUTING (policy ACCEPT 148K packets, 9297K bytes)  
 pkts bytes target prot opt in out source destination

----->>>---->>> iptables -L -v -n -t mangle 动态配置信息:   
Chain PREROUTING (policy ACCEPT 5872K packets, 386M bytes)  
 pkts bytes target prot opt in out source destination   
  
Chain INPUT (policy ACCEPT 5872K packets, 386M bytes)  
 pkts bytes target prot opt in out source destination   
  
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)  
 pkts bytes target prot opt in out source destination   
  
Chain OUTPUT (policy ACCEPT 5769K packets, 358M bytes)  
 pkts bytes target prot opt in out source destination   
  
Chain POSTROUTING (policy ACCEPT 5769K packets, 358M bytes)  
 pkts bytes target prot opt in out source destination

----->>>---->>> iptables -L -v -n -t raw 动态配置信息:   
Chain PREROUTING (policy ACCEPT 5872K packets, 386M bytes)  
 pkts bytes target prot opt in out source destination   
  
Chain OUTPUT (policy ACCEPT 5769K packets, 358M bytes)  
 pkts bytes target prot opt in out source destination

----->>>---->>> mount 动态配置信息:   
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)  
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)  
devtmpfs on /dev type devtmpfs (rw,nosuid,size=1928476k,nr\_inodes=482119,mode=755)  
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)  
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)  
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)  
tmpfs on /run type tmpfs (rw,nosuid,nodev,mode=755)  
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)  
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,release\_agent=/usr/lib/systemd/systemd-cgroups-agent,name=systemd)  
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)  
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpuacct,cpu)  
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)  
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)  
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)  
cgroup on /sys/fs/cgroup/net\_cls,net\_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net\_prio,net\_cls)  
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)  
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)  
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)  
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)  
cgroup on /sys/fs/cgroup/perf\_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf\_event)  
configfs on /sys/kernel/config type configfs (rw,relatime)  
/dev/vda1 on / type ext4 (rw,relatime,data=ordered)  
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime)  
mqueue on /dev/mqueue type mqueue (rw,relatime)  
systemd-1 on /proc/sys/fs/binfmt\_misc type autofs (rw,relatime,fd=36,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe\_ino=10939)  
debugfs on /sys/kernel/debug type debugfs (rw,relatime)  
tmpfs on /run/user/0 type tmpfs (rw,nosuid,nodev,relatime,size=388020k,mode=700)  
binfmt\_misc on /proc/sys/fs/binfmt\_misc type binfmt\_misc (rw,relatime)

建议:

* + 1. **selinux动态配置信息**

----->>>---->>> selinux 动态配置信息:   
SELinux status: disabled

建议:

* + 1. **Transparent Huge Pages**

----->>>---->>> 建议禁用Transparent Huge Pages (THP):   
always madvise [never]  
[always] madvise never

----->>>---->>> 硬盘SMART信息(需要root):

----->>>---->>> /var/log/boot.log

----->>>---->>> /var/log/cron(需要root)   
Jan 6 09:41:01 VM-16-10-centos CROND[27099]: (root) CMD (/usr/local/sa/agent/secu-tcs-agent-mon-safe.sh > /dev/null 2>1)  
Jan 6 09:42:01 VM-16-10-centos CROND[27596]: (vastbase) CMD (sh /home/vastbase/moitor.sh >> /home/vastbase/moitor.log)  
Jan 6 09:42:01 VM-16-10-centos CROND[27597]: (root) CMD (/usr/local/sa/agent/secu-tcs-agent-mon-safe.sh > /dev/null 2>1)  
Jan 6 09:42:01 VM-16-10-centos CROND[27595]: (root) CMD (flock -xn /tmp/stargate.lock -c '/usr/local/qcloud/stargate/admin/start.sh > /dev/null 2>1 ')  
Jan 6 09:43:01 VM-16-10-centos CROND[28094]: (vastbase) CMD (sh /home/vastbase/moitor.sh >> /home/vastbase/moitor.log)  
Jan 6 09:43:01 VM-16-10-centos CROND[28093]: (root) CMD (flock -xn /tmp/stargate.lock -c '/usr/local/qcloud/stargate/admin/start.sh > /dev/null 2>1 ')  
Jan 6 09:43:01 VM-16-10-centos CROND[28095]: (root) CMD (/usr/local/sa/agent/secu-tcs-agent-mon-safe.sh > /dev/null 2>1)  
Jan 6 09:44:01 VM-16-10-centos CROND[28596]: (vastbase) CMD (sh /home/vastbase/moitor.sh >> /home/vastbase/moitor.log)  
Jan 6 09:44:01 VM-16-10-centos CROND[28598]: (root) CMD (/usr/local/sa/agent/secu-tcs-agent-mon-safe.sh > /dev/null 2>1)  
Jan 6 09:44:01 VM-16-10-centos CROND[28597]: (root) CMD (flock -xn /tmp/stargate.lock -c '/usr/local/qcloud/stargate/admin/start.sh > /dev/null 2>1 ')  
Jan 6 09:45:01 VM-16-10-centos CROND[29095]: (vastbase) CMD (sh /home/vastbase/moitor.sh >> /home/vastbase/moitor.log)  
Jan 6 09:45:01 VM-16-10-centos CROND[29098]: (root) CMD (flock -xn /tmp/stargate.lock -c '/usr/local/qcloud/stargate/admin/start.sh > /dev/null 2>1 ')  
Jan 6 09:45:01 VM-16-10-centos CROND[29096]: (root) CMD (/usr/local/sa/agent/secu-tcs-agent-mon-safe.sh > /dev/null 2>1)  
Jan 6 09:45:01 VM-16-10-centos CROND[29097]: (root) CMD (flock -xn /tmp/stargate.lock -c '/usr/local/qcloud/stargate/admin/start.sh > /dev/null 2>1 ')  
Jan 6 09:46:01 VM-16-10-centos CROND[29596]: (root) CMD (/usr/local/sa/agent/secu-tcs-agent-mon-safe.sh > /dev/null 2>1)  
Jan 6 09:46:01 VM-16-10-centos CROND[29598]: (root) CMD (flock -xn /tmp/stargate.lock -c '/usr/local/qcloud/stargate/admin/start.sh > /dev/null 2>1 ')  
Jan 6 09:46:01 VM-16-10-centos CROND[29597]: (vastbase) CMD (sh /home/vastbase/moitor.sh >> /home/vastbase/moitor.log)  
Jan 6 09:47:02 VM-16-10-centos CROND[30175]: (root) CMD (/usr/local/sa/agent/secu-tcs-agent-mon-safe.sh > /dev/null 2>1)  
Jan 6 09:47:02 VM-16-10-centos CROND[30176]: (root) CMD (flock -xn /tmp/stargate.lock -c '/usr/local/qcloud/stargate/admin/start.sh > /dev/null 2>1 ')  
Jan 6 09:47:02 VM-16-10-centos CROND[30177]: (vastbase) CMD (sh /home/vastbase/moitor.sh >> /home/vastbase/moitor.log)

建议:

## 数据库运行结果

|  |  |  |
| --- | --- | --- |
| **栏目** | **状态** | **调整建议** |
| **数据库信息** |  |  |
| **配置文件检查** |  |  |
| **用户或数据库级别定制参数** |  |  |
| **数据库错误日志** |  |  |
| **数据库空间使用分析** |  |  |
| **数据库连接分析** |  |  |
| **数据库性能分析** |  |  |
| **数据库垃圾分析** |  |  |
| **数据库年龄分析** |  |  |
| **数据库XLOG, 流复制状态分析** |  |  |
| **数据库安全或潜在风险分析** |  |  |
| **重置统计信息** |  |  |

* + 1. **数据库信息**

----->>>---->>> 数据库版本:   
 version   
---------------------------------------------------------------------------------------------------------  
 PostgreSQL 11.5 on x86\_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-44), 64-bit  
(1 row)

----->>>---->>> 用户已安装的插件版本:   
 current\_database | extname | extowner | extnamespace | extrelocatable | extversion | extconfig | extcondition   
------------------+--------------------+----------+--------------+----------------+------------+-----------+--------------  
 postgres | plpgsql | 10 | 11 | f | 1.0 | |   
 postgres | pg\_freespacemap | 10 | 2200 | t | 1.2 | |   
 postgres | pg\_stat\_statements | 10 | 2200 | t | 1.6 | |   
(3 rows)  
  
 current\_database | extname | extowner | extnamespace | extrelocatable | extversion | extconfig | extcondition   
------------------+-----------------+----------+--------------+----------------+------------+-----------+--------------  
 tpcc | plpgsql | 10 | 11 | f | 1.0 | |   
 tpcc | pg\_freespacemap | 10 | 2200 | t | 1.2 | |   
(2 rows)  
  
 current\_database | extname | extowner | extnamespace | extrelocatable | extversion | extconfig | extcondition   
------------------+-----------------+----------+--------------+----------------+------------+-----------+--------------  
 test | plpgsql | 10 | 11 | f | 1.0 | |   
 test | pg\_freespacemap | 10 | 2200 | t | 1.2 | |   
(2 rows)

----->>>---->>> 用户使用了多少种数据类型:   
 current\_database | typname | count   
------------------+-----------+-------  
 postgres | int8 | 13  
 postgres | int4 | 8  
 postgres | float8 | 7  
 postgres | cid | 6  
 postgres | xid | 6  
 postgres | oid | 5  
 postgres | tid | 3  
 postgres | text | 1  
 postgres | timestamp | 1  
 postgres | varchar | 1  
 postgres | bpchar | 1  
 postgres | bool | 1  
(12 rows)  
  
 current\_database | typname | count   
------------------+-----------+-------  
 tpcc | int4 | 66  
 tpcc | varchar | 23  
 tpcc | cid | 22  
 tpcc | xid | 22  
 tpcc | bpchar | 20  
 tpcc | tid | 11  
 tpcc | numeric | 11  
 tpcc | oid | 11  
 tpcc | timestamp | 4  
 tpcc | int8 | 2  
 tpcc | bool | 1  
(11 rows)  
  
 current\_database | typname | count   
------------------+-----------+-------  
 test | int8 | 1130  
 test | xid | 532  
 test | cid | 532  
 test | varchar | 445  
 test | tid | 266  
 test | oid | 266  
 test | text | 163  
 test | bool | 129  
 test | int2 | 124  
 test | bpchar | 79  
 test | numeric | 57  
 test | timestamp | 9  
 test | int4 | 4  
 test | float8 | 1  
 test | date | 1  
(15 rows)

----->>>---->>> 用户创建了多少对象:   
 current\_database | rolname | nspname | relkind | count   
------------------+----------+---------+---------+-------  
 postgres | postgres | public | r | 3  
 postgres | postgres | public | v | 1  
(2 rows)  
  
 current\_database | rolname | nspname | relkind | count   
------------------+---------+---------+---------+-------  
 tpcc | tpcc | public | i | 12  
 tpcc | tpcc | public | r | 10  
 tpcc | tpcc | public | S | 1  
(3 rows)  
  
 current\_database | rolname | nspname | relkind | count   
------------------+----------+---------+---------+-------  
 test | postgres | mail | i | 219  
 test | postgres | mail | r | 136  
 test | postgres | mail | S | 128  
 test | postgres | public | r | 2  
(4 rows)

----->>>---->>> 用户对象占用空间的柱状图:   
 current\_database | this\_buk\_no | rels\_in\_this\_buk | buk\_min | buk\_max   
------------------+-------------+------------------+---------+---------  
 postgres | 1 | 3 | 0 bytes | 0 bytes  
 postgres | 10 | 1 | 6672 kB | 6672 kB  
(2 rows)  
  
 current\_database | this\_buk\_no | rels\_in\_this\_buk | buk\_min | buk\_max   
------------------+-------------+------------------+------------+---------  
 tpcc | 1 | 19 | 8192 bytes | 25 MB  
 tpcc | 3 | 1 | 90 MB | 90 MB  
 tpcc | 6 | 1 | 177 MB | 177 MB  
 tpcc | 9 | 1 | 290 MB | 290 MB  
 tpcc | 10 | 1 | 340 MB | 340 MB  
(5 rows)  
  
 current\_database | this\_buk\_no | rels\_in\_this\_buk | buk\_min | buk\_max   
------------------+-------------+------------------+------------+------------  
 test | 1 | 136 | 0 bytes | 0 bytes  
 test | 6 | 346 | 8192 bytes | 8192 bytes  
 test | 10 | 3 | 16 kB | 16 kB  
(3 rows)

建议:

* + 1. **配置文件检查**

----->>>---->>> 获取pg\_hba.conf md5值:   
ea3f472e22e0a46dff01bca505fab659 /home/postgres/data/pg\_hba.conf  
建议:   
 主备md5值一致(判断主备配置文件是否内容一致的一种手段, 或者使用diff).

----->>>---->>> 获取pg\_hba.conf配置:   
local all all trust  
host all all 127.0.0.1/32 trust  
host all all ::1/128 trust  
local replication all trust  
host replication all 127.0.0.1/32 trust  
host replication all ::1/128 trust  
host all all 0.0.0.0/0 md5  
建议:   
 主备配置尽量保持一致, 注意trust和password认证方法的危害(password方法 验证时网络传输密码明文, 建议改为md5), 建议除了unix socket可以使用trust以外, 其他都使用md5或者LDAP认证方法.  
 建议先设置白名单(超级用户允许的来源IP, 可以访问的数据库), 再设置黑名单(不允许超级用户登陆, reject), 再设置白名单(普通应用), 参考pg\_hba.conf中的描述.

----->>>---->>> 获取postgresql.conf md5值:   
5bebbb84c4e61ce8503f4b37472c26e8 /home/postgres/data/postgresql.conf  
建议:   
 主备md5值一致(判断主备配置文件是否内容一致的一种手段, 或者使用diff).

----->>>---->>> 获取postgresql.conf配置:   
port = 15432   
max\_connections = 100   
shared\_buffers = 128MB   
dynamic\_shared\_memory\_type = posix   
max\_wal\_size = 1GB  
min\_wal\_size = 80MB  
log\_destination = 'stderr'   
logging\_collector = on   
log\_filename = 'postgresql-%Y-%m-%d.log'   
log\_min\_messages = warning   
log\_timezone = 'Asia/Shanghai'  
datestyle = 'iso, mdy'  
timezone = 'Asia/Shanghai'  
lc\_messages = 'en\_US.UTF-8'   
lc\_monetary = 'en\_US.UTF-8'   
lc\_numeric = 'en\_US.UTF-8'   
lc\_time = 'en\_US.UTF-8'   
default\_text\_search\_config = 'pg\_catalog.english'  
shared\_preload\_libraries='passwordcheck,auto\_explain,pg\_stat\_statements'  
auto\_explain.log\_min\_duration=100  
auto\_explain.log\_analyze=on  
auto\_explain.log\_timing=on  
auto\_explain.log\_buffers=on  
auto\_explain.log\_verbose=on  
auto\_explain.log\_triggers=off  
auto\_explain.log\_nested\_statements=on  
auto\_explain.log\_format='TEXT'  
auto\_explain.log\_level='LOG'  
auto\_explain.sample\_rate=1  
track\_io\_timing = on   
track\_activity\_query\_size = 2048   
pg\_stat\_statements.max = 10000   
pg\_stat\_statements.track = all   
pg\_stat\_statements.track\_utility = off   
pg\_stat\_statements.save = on   
log\_destination = 'stderr'  
log\_line\_prefix = '%t [%p]: [%l-1] user=%u,db=%d,client=%h '  
log\_checkpoints = on  
log\_connections = on  
log\_disconnections = on  
log\_lock\_waits = on  
log\_temp\_files = 0  
log\_autovacuum\_min\_duration = 0  
建议:   
 主备配置尽量保持一致, 配置合理的参数值.  
 建议修改的参数列表如下 ( 假设操作系统为128GB, 数据库独占操作系统, 数据库版本9.4.x, 其他版本可能略有不同, 未来再更新进来 ) :   
listen\_addresses = '0.0.0.0' # 监听所有IPV4地址  
port = 1921 # 监听非默认端口  
max\_connections = 4000 # 最大允许连接数  
superuser\_reserved\_connections = 20 # 为超级用户保留的连接  
unix\_socket\_directories = '.' # unix socket文件目录最好放在/home/postgres/data中, 确保安全  
unix\_socket\_permissions = 0700 # 确保权限安全  
tcp\_keepalives\_idle = 30 # 间歇性发送TCP心跳包, 防止连接被网络设备中断.  
tcp\_keepalives\_interval = 10  
tcp\_keepalives\_count = 10  
shared\_buffers = 16GB # 数据库自己管理的共享大小, 如果用大页, 建议设置为: - 100\*work\_mem - autovacuum\_max\_workers\*(autovacuum\_work\_mem or autovacuum\_work\_mem) - max\_connections\*1MB  
huge\_pages = try # 尽量使用大页, 需要操作系统支持, 配置vm.nr\_hugepages\*2MB大于shared\_buffers.  
maintenance\_work\_mem = 512MB # 可以加速创建索引, 回收垃圾(假设没有设置autovacuum\_work\_mem)  
autovacuum\_work\_mem = 512MB # 可以加速回收垃圾  
shared\_preload\_libraries = 'auth\_delay,passwordcheck,pg\_stat\_statements,auto\_explain' # 建议防止暴力破解, 密码复杂度检测, 开启pg\_stat\_statements, 开启auto\_explain, 参考 http://blog.163.com/digoal@126/blog/static/16387704020149852941586   
bgwriter\_delay = 10ms # bgwriter process间隔多久调用write接口(注意不是fsync)将shared buffer中的dirty page写到文件系统.  
bgwriter\_lru\_maxpages = 1000 # 一个周期最多写多少脏页  
max\_worker\_processes = 20 # 如果要使用worker process, 最多可以允许fork 多少个worker进程.  
wal\_level = logical # 如果将来打算使用logical复制, 最后先配置好, 不需要停机再改.  
synchronous\_commit = off # 如果磁盘的IOPS能力一般, 建议使用异步提交来提高性能, 但是数据库crash或操作系统crash时, 最多可能丢失2\*wal\_writer\_delay时间段产生的事务日志(在wal buffer中).   
wal\_sync\_method = open\_datasync # 使用pg\_test\_fsync测试wal所在磁盘的fsync接口, 使用性能好的.  
wal\_buffers = 16MB  
wal\_writer\_delay = 10ms  
checkpoint\_segments = 1024 # 等于shared\_buffers除以单个wal segment的大小.  
checkpoint\_timeout = 50min  
checkpoint\_completion\_target = 0.8  
archive\_mode = on # 最好先开启, 否则需要重启数据库来修改  
archive\_command = '/bin/date' # 最好先开启, 否则需要重启数据库来修改, 将来修改为正确的命令例如, test ! -f /home/postgres/archivedir/pg\_root/%f cp %p /home/postgres/archivedir/pg\_root/%f  
max\_wal\_senders = 32 # 最多允许多少个wal sender进程.  
wal\_keep\_segments = 2048 # 在pg\_xlog目录中保留的WAL文件数, 根据流复制业务的延迟情况和pg\_xlog目录大小来预估.  
max\_replication\_slots = 32 # 最多允许多少个复制插槽  
hot\_standby = on  
max\_standby\_archive\_delay = 300s # 如果备库要被用于只读, 有大的查询的情况下, 如果遇到conflicts, 可以考虑调整这个值来避免conflict造成cancel query.  
max\_standby\_streaming\_delay = 300s # 如果备库要被用于只读, 有大的查询的情况下, 如果遇到conflicts, 可以考虑调整这个值来避免conflict造成cancel query.  
wal\_receiver\_status\_interval = 1s  
hot\_standby\_feedback = off # 建议关闭, 如果备库出现long query，可能导致主库频繁的autovacuum(比如出现无法回收被需要的垃圾时)  
vacuum\_defer\_cleanup\_age = 0 # 建议设置为0，避免主库出现频繁的autovacuum无用功，也许新版本会改进。  
random\_page\_cost = 1.3 # 根据IO能力调整(企业级SSD为例 1.3是个经验值)  
effective\_cache\_size = 100GB # 调整为与内存一样大, 或者略小(减去shared\_buffer). 用来评估OS PAGE CACHE可以用到的内存大小.  
log\_destination = 'csvlog'  
logging\_collector = on  
log\_truncate\_on\_rotation = on  
log\_rotation\_size = 10MB  
log\_min\_duration\_statement = 1s  
log\_checkpoints = on  
log\_connections = on  
log\_disconnections = on  
log\_error\_verbosity = verbose # 在日志中输出代码位置  
log\_lock\_waits = on  
log\_statement = 'ddl'  
autovacuum = on  
log\_autovacuum\_min\_duration = 0  
autovacuum\_max\_workers = 10 # 根据实际频繁变更或删除记录的对象数决定  
autovacuum\_naptime = 30s # 快速唤醒, 防止膨胀  
autovacuum\_vacuum\_scale\_factor = 0.1 # 当垃圾超过比例时, 启动垃圾回收工作进程  
autovacuum\_analyze\_scale\_factor = 0.2   
autovacuum\_freeze\_max\_age = 1600000000  
autovacuum\_multixact\_freeze\_max\_age = 1600000000  
vacuum\_freeze\_table\_age = 1500000000  
vacuum\_multixact\_freeze\_table\_age = 1500000000  
auth\_delay.milliseconds = 5000 # 认证失败, 延迟多少毫秒反馈  
auto\_explain.log\_min\_duration = 5000 # 记录超过多少毫秒的SQL当时的执行计划  
auto\_explain.log\_analyze = true  
auto\_explain.log\_verbose = true  
auto\_explain.log\_buffers = true  
auto\_explain.log\_nested\_statements = true  
pg\_stat\_statements.track\_utility=off  
  
 建议的操作系统配置(根据实际情况修改) :   
vi /etc/sysctl.conf  
# add by digoal.zhou  
fs.aio-max-nr = 1048576  
fs.file-max = 76724600  
kernel.core\_pattern= /data01/corefiles/core\_%e\_%u\_%t\_%s.%p   
# /data01/corefiles事先建好，权限777  
kernel.sem = 4096 2147483647 2147483646 512000   
# 信号量, ipcs -l 或 -u 查看，每16个进程一组，每组信号量需要17个信号量。  
kernel.shmall = 107374182   
# 所有共享内存段相加大小限制(建议内存的80%)  
kernel.shmmax = 274877906944   
# 最大单个共享内存段大小(建议为内存一半), >9.2的版本已大幅降低共享内存的使用  
kernel.shmmni = 819200   
# 一共能生成多少共享内存段，每个PG数据库集群至少2个共享内存段  
net.core.netdev\_max\_backlog = 10000  
net.core.rmem\_default = 262144   
# The default setting of the socket receive buffer in bytes.  
net.core.rmem\_max = 4194304   
# The maximum receive socket buffer size in bytes  
net.core.wmem\_default = 262144   
# The default setting (in bytes) of the socket send buffer.  
net.core.wmem\_max = 4194304   
# The maximum send socket buffer size in bytes.  
net.core.somaxconn = 4096  
net.ipv4.tcp\_max\_syn\_backlog = 4096  
net.ipv4.tcp\_keepalive\_intvl = 20  
net.ipv4.tcp\_keepalive\_probes = 3  
net.ipv4.tcp\_keepalive\_time = 60  
net.ipv4.tcp\_mem = 8388608 12582912 16777216  
net.ipv4.tcp\_fin\_timeout = 5  
net.ipv4.tcp\_synack\_retries = 2  
net.ipv4.tcp\_syncookies = 1   
# 开启SYN Cookies。当出现SYN等待队列溢出时，启用cookie来处理，可防范少量的SYN攻击  
net.ipv4.tcp\_timestamps = 1   
# 减少time\_wait  
net.ipv4.tcp\_tw\_recycle = 0   
# 如果=1则开启TCP连接中TIME-WAIT套接字的快速回收，但是NAT环境可能导致连接失败，建议服务端关闭它  
net.ipv4.tcp\_tw\_reuse = 1   
# 开启重用。允许将TIME-WAIT套接字重新用于新的TCP连接  
net.ipv4.tcp\_max\_tw\_buckets = 262144  
net.ipv4.tcp\_rmem = 8192 87380 16777216  
net.ipv4.tcp\_wmem = 8192 65536 16777216  
net.nf\_conntrack\_max = 1200000  
net.netfilter.nf\_conntrack\_max = 1200000  
vm.dirty\_background\_bytes = 409600000   
# 系统脏页到达这个值，系统后台刷脏页调度进程 pdflush（或其他） 自动将(dirty\_expire\_centisecs/100）秒前的脏页刷到磁盘  
vm.dirty\_expire\_centisecs = 3000   
# 比这个值老的脏页，将被刷到磁盘。3000表示30秒。  
vm.dirty\_ratio = 95   
# 如果系统进程刷脏页太慢，使得系统脏页超过内存 95 % 时，则用户进程如果有写磁盘的操作（如fsync, fdatasync等调用），则需要主动把系统脏页刷出。  
# 有效防止用户进程刷脏页，在单机多实例，并且使用CGROUP限制单实例IOPS的情况下非常有效。   
vm.dirty\_writeback\_centisecs = 100   
# pdflush（或其他）后台刷脏页进程的唤醒间隔， 100表示1秒。  
vm.extra\_free\_kbytes = 4096000  
vm.min\_free\_kbytes = 2097152  
vm.mmap\_min\_addr = 65536  
vm.overcommit\_memory = 0   
# 在分配内存时，允许少量over malloc, 如果设置为 1, 则认为总是有足够的内存，内存较少的测试环境可以使用 1 .   
vm.overcommit\_ratio = 90   
# 当overcommit\_memory = 2 时，用于参与计算允许指派的内存大小。  
vm.swappiness = 0   
# 关闭交换分区  
vm.zone\_reclaim\_mode = 0   
# 禁用 numa, 或者在vmlinux中禁止.   
net.ipv4.ip\_local\_port\_range = 40000 65535   
# 本地自动分配的TCP, UDP端口号范围  
# vm.nr\_hugepages = 102352   
# 建议shared buffer设置超过64GB时 使用大页，页大小 /proc/meminfo Hugepagesize  
  
vi /etc/security/limits.conf  
\* soft nofile 1024000  
\* hard nofile 1024000  
\* soft nproc unlimited  
\* hard nproc unlimited  
\* soft core unlimited  
\* hard core unlimited  
\* soft memlock unlimited  
\* hard memlock unlimited

----->>>---->>> 获取recovery.done md5值:   
建议:   
 主备md5值一致(判断主备配置文件是否内容一致的一种手段, 或者使用diff).  
----->>>---->>> 获取recovery.done配置:   
建议:   
 在primary\_conninfo中不要配置密码, 容易泄露. 建议为流复制用户创建replication角色的用户, 并且配置pg\_hba.conf只允许需要的来源IP连接.

* + 1. **数据库定制参数**

----->>>---->>> 用户或数据库级别定制参数:   
 setdatabase | setrole | setconfig   
-------------+---------+-----------  
(0 rows)  
  
建议:   
 定制参数需要关注, 优先级高于数据库的启动参数和配置文件中的参数, 特别是排错时需要关注.

* + 1. **数据库错误日志分析**

----->>>---->>> 获取错误日志信息:   
2022-01-03 23:43:39 CST [31458]: [3-1] user=vbadmin,db=vastbase,client=[local] FATAL: role "vbadmin" does not exist  
2022-01-03 23:43:39 CST [31456]: [3-1] user=vbadmin,db=vastbase,client=[local] FATAL: role "vbadmin" does not exist  
2022-01-03 23:43:39 CST [31454]: [3-1] user=vbadmin,db=vastbase,client=[local] FATAL: role "vbadmin" does not exist  
2022-01-01 12:50:50 CST [24604]: [7-1] user=postgres,db=postgres,client=[local] ERROR: function pg\_current\_xlog\_location() does not exist at character 8  
2022-01-01 12:49:05 CST [24604]: [5-1] user=postgres,db=postgres,client=[local] ERROR: syntax error at or near ")" at character 571  
2022-01-01 12:41:51 CST [24604]: [3-1] user=postgres,db=postgres,client=[local] ERROR: relation "t" does not exist at character 53  
建议:   
 参考 http://www.postgresql.org/docs/current/static/errcodes-appendix.html .

----->>>---->>> 获取连接请求情况:   
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:54 CST [22747]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:54 CST [22745]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:54 CST [22743]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:54 CST [22741]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22739]: [2-1] user=postgres,db=test,client=127.0.0.1 LOG: connection authorized: user=postgres database=test  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22737]: [2-1] user=postgres,db=tpcc,client=127.0.0.1 LOG: connection authorized: user=postgres database=tpcc  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22735]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22733]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22731]: [2-1] user=postgres,db=test,client=127.0.0.1 LOG: connection authorized: user=postgres database=test  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22728]: [2-1] user=postgres,db=tpcc,client=127.0.0.1 LOG: connection authorized: user=postgres database=tpcc  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22726]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22724]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22722]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22720]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22718]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22716]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22714]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22712]: [2-1] user=postgres,db=test,client=127.0.0.1 LOG: connection authorized: user=postgres database=test  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22710]: [2-1] user=postgres,db=tpcc,client=127.0.0.1 LOG: connection authorized: user=postgres database=tpcc  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22703]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22701]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22699]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22697]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22663]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22651]: [2-1] user=postgres,db=test,client=127.0.0.1 LOG: connection authorized: user=postgres database=test  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22649]: [2-1] user=postgres,db=tpcc,client=127.0.0.1 LOG: connection authorized: user=postgres database=tpcc  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22647]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22645]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22643]: [2-1] user=postgres,db=test,client=127.0.0.1 LOG: connection authorized: user=postgres database=test  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22641]: [2-1] user=postgres,db=tpcc,client=127.0.0.1 LOG: connection authorized: user=postgres database=tpcc  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22639]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:53 CST [22637]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:52 CST [22633]: [2-1] user=postgres,db=test,client=127.0.0.1 LOG: connection authorized: user=postgres database=test  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:52 CST [22631]: [2-1] user=postgres,db=tpcc,client=127.0.0.1 LOG: connection authorized: user=postgres database=tpcc  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:52 CST [22629]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:52 CST [22627]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:52 CST [22625]: [2-1] user=postgres,db=test,client=127.0.0.1 LOG: connection authorized: user=postgres database=test  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:52 CST [22623]: [2-1] user=postgres,db=tpcc,client=127.0.0.1 LOG: connection authorized: user=postgres database=tpcc  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:52 CST [22621]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:51 CST [22619]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:51 CST [22617]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:51 CST [22569]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 21:04:51 CST [22564]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 20:39:50 CST [15717]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 20:39:50 CST [15712]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 20:39:33 CST [15590]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 20:39:33 CST [15585]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 20:36:53 CST [14803]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-03.log:2022-01-03 20:36:53 CST [14798]: [2-1] user=postgres,db=postgres,client=127.0.0.1 LOG: connection authorized: user=postgres database=postgres  
/home/postgres/data/log/postgresql-2022-01-01.log:2022-01-01 12:23:49 CST [24604]: [2-1] user=postgres,db=postgres,client=[local] LOG: connection authorized: user=postgres database=postgres  
建议:   
 连接请求非常多时, 请考虑应用层使用连接池, 或者使用pgbouncer连接池.

----->>>---->>> 获取认证失败情况:   
建议:   
 认证失败次数很多时, 可能是有用户在暴力破解, 建议使用auth\_delay插件防止暴力破解.

* + 1. **数据库慢SQL分析**

----->>>---->>> 慢查询统计:   
建议:   
 输出格式(条数,日期,用户,数据库,QUERY,耗时ms).   
 慢查询反映执行时间超过log\_min\_duration\_statement的SQL, 可以根据实际情况分析数据库或SQL语句是否有优化空间.

----->>>---->>> 慢查询分布头10条的执行时间, ms:

----->>>---->>> 慢查询分布尾10条的执行时间, ms:

* + 1. **数据库空间使用分析**

----->>>---->>> 输出文件系统剩余空间:   
Filesystem 1M-blocks Used Available Use% Mounted on  
devtmpfs 1884 0 1884 0% /dev  
tmpfs 1895 1 1895 1% /dev/shm  
tmpfs 1895 1 1895 1% /run  
tmpfs 1895 0 1895 0% /sys/fs/cgroup  
/dev/vda1 80506 32343 44789 42% /  
tmpfs 379 0 379 0% /run/user/0  
建议:   
 注意预留足够的空间给数据库.

----->>>---->>> 输出表空间对应目录:   
/home/postgres/data  
total 8  
drwx------ 2 postgres postgres 4096 Oct 31 12:02 .  
drwx------ 20 postgres postgres 4096 Jan 6 00:00 ..  
建议:   
 注意表空间如果不是软链接, 注意是否刻意所为, 正常情况下应该是软链接.

----->>>---->>> 输出表空间使用情况:   
 spcname | pg\_tablespace\_location | pg\_size\_pretty   
------------+------------------------+----------------  
 pg\_default | | 1082 MB  
 pg\_global | | 574 kB  
(2 rows)  
  
建议:   
 注意检查表空间所在文件系统的剩余空间, (默认表空间在/home/postgres/data/base目录下), IOPS分配是否均匀, OS的sysstat包可以观察IO使用率.

----->>>---->>> 输出数据库使用情况:   
 datname | pg\_size\_pretty   
-----------+----------------  
 tpcc | 1040 MB  
 postgres | 14 MB  
 test | 13 MB  
 template1 | 7733 kB  
 template0 | 7593 kB  
(5 rows)  
  
建议:   
 注意检查数据库的大小, 是否需要清理历史数据.

----->>>---->>> TOP 10 size对象:   
 current\_database | nspname | relname | relkind | pg\_size\_pretty | seq\_scan | seq\_tup\_read | idx\_scan | idx\_tup\_fetch | n\_tup\_ins | n\_tup\_upd | n\_tup\_del | n\_tup\_hot\_upd | n\_live\_tup | n\_dead\_tup   
------------------+------------+----------------+---------+----------------+----------+--------------+----------+---------------+-----------+-----------+-----------+---------------+------------+------------  
 postgres | public | operation | r | 6672 kB | 0 | 0 | | | 0 | 0 | 0 | 0 | 0 | 0  
 postgres | pg\_catalog | pg\_proc | r | 608 kB | 0 | 0 | 81 | 132 | 0 | 0 | 0 | 0 | 0 | 0  
 postgres | pg\_catalog | pg\_depend | r | 448 kB | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  
 postgres | pg\_catalog | pg\_attribute | r | 392 kB | 1 | 2645 | 1318 | 3255 | 0 | 0 | 0 | 0 | 0 | 0  
 postgres | pg\_catalog | pg\_description | r | 280 kB | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  
 postgres | pg\_catalog | pg\_collation | r | 232 kB | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  
 postgres | pg\_catalog | pg\_statistic | r | 152 kB | 0 | 0 | 35 | 27 | 0 | 0 | 0 | 0 | 0 | 0  
 postgres | pg\_catalog | pg\_operator | r | 120 kB | 0 | 0 | 52 | 69 | 0 | 0 | 0 | 0 | 0 | 0  
 postgres | pg\_catalog | pg\_rewrite | r | 96 kB | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0  
 postgres | pg\_catalog | pg\_class | r | 96 kB | 2420 | 639812 | 1330 | 1330 | 0 | 0 | 0 | 0 | 0 | 0  
(10 rows)  
  
 current\_database | nspname | relname | relkind | pg\_size\_pretty | seq\_scan | seq\_tup\_read | idx\_scan | idx\_tup\_fetch | n\_tup\_ins | n\_tup\_upd | n\_tup\_del | n\_tup\_hot\_upd | n\_live\_tup | n\_dead\_tup   
------------------+------------+------------------+---------+----------------+----------+--------------+----------+---------------+-----------+-----------+-----------+---------------+------------+------------  
 tpcc | public | bmsql\_stock | r | 340 MB | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  
 tpcc | public | bmsql\_order\_line | r | 290 MB | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  
 tpcc | public | bmsql\_customer | r | 177 MB | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  
 tpcc | public | bmsql\_history | r | 25 MB | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  
 tpcc | public | bmsql\_oorder | r | 19 MB | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  
 tpcc | public | bmsql\_item | r | 10 MB | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  
 tpcc | public | bmsql\_new\_order | r | 3896 kB | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  
 tpcc | pg\_catalog | pg\_proc | r | 608 kB | 0 | 0 | 46 | 93 | 0 | 0 | 0 | 0 | 0 | 0  
 tpcc | pg\_catalog | pg\_depend | r | 456 kB | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  
 tpcc | pg\_catalog | pg\_attribute | r | 416 kB | 1 | 2796 | 1277 | 3237 | 0 | 0 | 0 | 0 | 0 | 0  
(10 rows)  
  
 current\_database | nspname | relname | relkind | pg\_size\_pretty | seq\_scan | seq\_tup\_read | idx\_scan | idx\_tup\_fetch | n\_tup\_ins | n\_tup\_upd | n\_tup\_del | n\_tup\_hot\_upd | n\_live\_tup | n\_dead\_tup   
------------------+------------+----------------+---------+----------------+----------+--------------+----------+---------------+-----------+-----------+-----------+---------------+------------+------------  
 test | pg\_catalog | pg\_attribute | r | 1112 kB | 1 | 7111 | 1736 | 5230 | 0 | 0 | 0 | 0 | 0 | 0  
 test | pg\_catalog | pg\_proc | r | 608 kB | 0 | 0 | 46 | 93 | 0 | 0 | 0 | 0 | 0 | 0  
 test | pg\_catalog | pg\_depend | r | 568 kB | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  
 test | pg\_catalog | pg\_attrdef | r | 288 kB | 0 | 0 | 132 | 696 | 0 | 0 | 0 | 0 | 0 | 0  
 test | pg\_catalog | pg\_description | r | 280 kB | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  
 test | pg\_catalog | pg\_collation | r | 232 kB | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  
 test | pg\_catalog | pg\_class | r | 184 kB | 2416 | 1811067 | 2222 | 2222 | 0 | 0 | 0 | 0 | 0 | 0  
 test | pg\_catalog | pg\_statistic | r | 184 kB | 0 | 0 | 25 | 20 | 0 | 0 | 0 | 0 | 0 | 0  
 test | pg\_catalog | pg\_type | r | 160 kB | 1 | 855 | 49 | 47 | 0 | 0 | 0 | 0 | 0 | 0  
 test | pg\_catalog | pg\_operator | r | 120 kB | 0 | 0 | 42 | 59 | 0 | 0 | 0 | 0 | 0 | 0  
(10 rows)  
  
建议:   
 经验值: 单表超过8GB, 并且这个表需要频繁更新 或 删除+插入的话, 建议对表根据业务逻辑进行合理拆分后获得更好的性能, 以及便于对膨胀索引进行维护; 如果是只读的表, 建议适当结合SQL语句进行优化.

* + 1. **数据库连接分析**

----->>>---->>> 当前活跃度:   
 now | state | count   
-------------------------------+--------+-------  
 2022-01-06 09:47:57.390583+08 | active | 1  
 2022-01-06 09:47:57.390583+08 | | 5  
(2 rows)

建议:

如果active状态很多, 说明数据库比较繁忙. 如果idle in transaction很多, 说明业务逻辑设计可能有问题. 如果idle很多, 可能使用了连接池, 并且可能没有自动回收连接到连接池的最小连接数.

建议:

给超级用户和普通用户设置足够的连接, 以免不能登录数据库.

----->>>---->>> 用户连接数限制:   
 rolname | rolconnlimit | connects   
----------+--------------+----------  
 postgres | -1 | 2  
(1 row)

建议:

给用户设置足够的连接数, alter role ... CONNECTION LIMIT .

----->>>---->>> 数据库连接限制:   
 datname | datconnlimit | connects   
----------+--------------+----------  
 postgres | -1 | 1  
(1 row)

建议:

给数据库设置足够的连接数, alter database ... CONNECTION LIMIT .

* + 1. **数据库性能分析**

----->>>---->>> TOP 5 SQL : total\_cpu\_time   
-[ RECORD 1 ]-------+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  
rolname | postgres  
datname | test  
per\_call\_time | 19.71056  
userid | 10  
dbid | 16620  
queryid | 2247481896277058284  
query | select current\_database(),buk this\_buk\_no,cnt rels\_in\_this\_buk,pg\_size\_pretty(min) buk\_min,pg\_size\_pretty(max) buk\_max from( select row\_number() over (partition by buk order by tsize),tsize,buk,min(tsize) over (partition by buk),max(tsize) over (partition by buk),count(\*) over (partition by buk) cnt from ( select pg\_relation\_size(a.oid) tsize, width\_bucket(pg\_relation\_size(a.oid),tmin-$1,tmax+$2,$3) buk from (select min(pg\_relation\_size(a.oid)) tmin,max(pg\_relation\_size(a.oid)) tmax from pg\_class a,pg\_namespace c where a.relnamespace=c.oid and nspname !~ $4 and nspname>$5) t, pg\_class a,pg\_namespace c where a.relnamespace=c.oid and nspname !~ $6 and nspname>$7 ) t)t where row\_number=$8  
calls | 1  
total\_time | 19.71056  
min\_time | 19.71056  
max\_time | 19.71056  
mean\_time | 19.71056  
stddev\_time | 0  
rows | 3  
shared\_blks\_hit | 5665  
shared\_blks\_read | 0  
shared\_blks\_dirtied | 0  
shared\_blks\_written | 0  
local\_blks\_hit | 0  
local\_blks\_read | 0  
local\_blks\_dirtied | 0  
local\_blks\_written | 0  
temp\_blks\_read | 0  
temp\_blks\_written | 0  
blk\_read\_time | 0  
blk\_write\_time | 0  
-[ RECORD 2 ]-------+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  
rolname | postgres  
datname | test  
per\_call\_time | 18.625932  
userid | 10  
dbid | 16620  
queryid | -211143377306445070  
query | select current\_database(),b.nspname,c.relname,c.relkind,pg\_size\_pretty(pg\_relation\_size(c.oid)),a.seq\_scan,a.seq\_tup\_read,a.idx\_scan,a.idx\_tup\_fetch,a.n\_tup\_ins,a.n\_tup\_upd,a.n\_tup\_del,a.n\_tup\_hot\_upd,a.n\_live\_tup,a.n\_dead\_tup from pg\_stat\_all\_tables a, pg\_class c,pg\_namespace b where c.relnamespace=b.oid and c.relkind=$1 and a.relid=c.oid order by pg\_relation\_size(c.oid) desc limit $2  
calls | 1  
total\_time | 18.625932  
min\_time | 18.625932  
max\_time | 18.625932  
mean\_time | 18.625932  
stddev\_time | 0  
rows | 10  
shared\_blks\_hit | 2182  
shared\_blks\_read | 0  
shared\_blks\_dirtied | 0  
shared\_blks\_written | 0  
local\_blks\_hit | 0  
local\_blks\_read | 0  
local\_blks\_dirtied | 0  
local\_blks\_written | 0  
temp\_blks\_read | 0  
temp\_blks\_written | 0  
blk\_read\_time | 0  
blk\_write\_time | 0  
-[ RECORD 3 ]-------+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  
rolname | postgres  
datname | tpcc  
per\_call\_time | 12.731471  
userid | 10  
dbid | 16384  
queryid | -211143377306445070  
query | select current\_database(),b.nspname,c.relname,c.relkind,pg\_size\_pretty(pg\_relation\_size(c.oid)),a.seq\_scan,a.seq\_tup\_read,a.idx\_scan,a.idx\_tup\_fetch,a.n\_tup\_ins,a.n\_tup\_upd,a.n\_tup\_del,a.n\_tup\_hot\_upd,a.n\_live\_tup,a.n\_dead\_tup from pg\_stat\_all\_tables a, pg\_class c,pg\_namespace b where c.relnamespace=b.oid and c.relkind=$1 and a.relid=c.oid order by pg\_relation\_size(c.oid) desc limit $2  
calls | 1  
total\_time | 12.731471  
min\_time | 12.731471  
max\_time | 12.731471  
mean\_time | 12.731471  
stddev\_time | 0  
rows | 10  
shared\_blks\_hit | 489  
shared\_blks\_read | 0  
shared\_blks\_dirtied | 0  
shared\_blks\_written | 0  
local\_blks\_hit | 0  
local\_blks\_read | 0  
local\_blks\_dirtied | 0  
local\_blks\_written | 0  
temp\_blks\_read | 0  
temp\_blks\_written | 0  
blk\_read\_time | 0  
blk\_write\_time | 0  
-[ RECORD 4 ]-------+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  
rolname | postgres  
datname | postgres  
per\_call\_time | 12.207891  
userid | 10  
dbid | 13287  
queryid | -211143377306445070  
query | select current\_database(),b.nspname,c.relname,c.relkind,pg\_size\_pretty(pg\_relation\_size(c.oid)),a.seq\_scan,a.seq\_tup\_read,a.idx\_scan,a.idx\_tup\_fetch,a.n\_tup\_ins,a.n\_tup\_upd,a.n\_tup\_del,a.n\_tup\_hot\_upd,a.n\_live\_tup,a.n\_dead\_tup from pg\_stat\_all\_tables a, pg\_class c,pg\_namespace b where c.relnamespace=b.oid and c.relkind=$1 and a.relid=c.oid order by pg\_relation\_size(c.oid) desc limit $2  
calls | 1  
total\_time | 12.207891  
min\_time | 12.207891  
max\_time | 12.207891  
mean\_time | 12.207891  
stddev\_time | 0  
rows | 10  
shared\_blks\_hit | 390  
shared\_blks\_read | 0  
shared\_blks\_dirtied | 0  
shared\_blks\_written | 0  
local\_blks\_hit | 0  
local\_blks\_read | 0  
local\_blks\_dirtied | 0  
local\_blks\_written | 0  
temp\_blks\_read | 0  
temp\_blks\_written | 0  
blk\_read\_time | 0  
blk\_write\_time | 0  
-[ RECORD 5 ]-------+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  
rolname | postgres  
datname | postgres  
per\_call\_time | 7.061715  
userid | 10  
dbid | 13287  
queryid | -1256725786399044094  
query | select spcname,pg\_tablespace\_location(oid),pg\_size\_pretty(pg\_tablespace\_size(oid)) from pg\_tablespace order by pg\_tablespace\_size(oid) desc  
calls | 1  
total\_time | 7.061715  
min\_time | 7.061715  
max\_time | 7.061715  
mean\_time | 7.061715  
stddev\_time | 0  
rows | 2  
shared\_blks\_hit | 4  
shared\_blks\_read | 0  
shared\_blks\_dirtied | 0  
shared\_blks\_written | 0  
local\_blks\_hit | 0  
local\_blks\_read | 0  
local\_blks\_dirtied | 0  
local\_blks\_written | 0  
temp\_blks\_read | 0  
temp\_blks\_written | 0  
blk\_read\_time | 0  
blk\_write\_time | 0

建议:

检查SQL是否有优化空间, 配合auto\_explain插件在csvlog中观察LONG SQL的执行计划是否正确.

----->>>---->>> 索引数超过4并且SIZE大于10MB的表:   
 current\_database | nspname | relname | pg\_size\_pretty | idx\_cnt   
------------------+---------+---------+----------------+---------  
(0 rows)  
  
 current\_database | nspname | relname | pg\_size\_pretty | idx\_cnt   
------------------+---------+---------+----------------+---------  
(0 rows)  
  
 current\_database | nspname | relname | pg\_size\_pretty | idx\_cnt   
------------------+---------+---------+----------------+---------  
(0 rows)

----->>>---->>> 上次巡检以来未使用或使用较少的索引:   
 current\_database | schemaname | relname | indexrelname | idx\_scan | idx\_tup\_read | idx\_tup\_fetch | pg\_size\_pretty   
------------------+------------+---------+--------------+----------+--------------+---------------+----------------  
(0 rows)  
  
 current\_database | schemaname | relname | indexrelname | idx\_scan | idx\_tup\_read | idx\_tup\_fetch | pg\_size\_pretty   
------------------+------------+----------------+---------------------+----------+--------------+---------------+----------------  
 tpcc | public | bmsql\_customer | bmsql\_customer\_idx1 | 0 | 0 | 0 | 16 MB  
 tpcc | public | bmsql\_oorder | bmsql\_oorder\_idx1 | 0 | 0 | 0 | 10064 kB  
(2 rows)  
  
 current\_database | schemaname | relname | indexrelname | idx\_scan | idx\_tup\_read | idx\_tup\_fetch | pg\_size\_pretty   
------------------+------------+---------+--------------+----------+--------------+---------------+----------------  
(0 rows)

建议:

索引数量太多, 影响表的增删改性能, 建议检查是否有不需要的索引.

----->>>---->>> 数据库统计信息, 回滚比例, 命中比例, 数据块读写时间, 死锁, 复制冲突:   
 datname | rollback\_ratio | hit\_ratio | blk\_read\_time | blk\_write\_time | conflicts | deadlocks   
-----------+----------------+-----------+---------------+----------------+-----------+-----------  
 postgres | 0.00 % | 100.00 % | 0 | 0 | 0 | 0  
 tpcc | 0.00 % | 100.00 % | 0 | 0 | 0 | 0  
 template1 | 0.00 % | 99.98 % | 134.28 | 0 | 0 | 0  
 template0 | 0.00 % | 0.00 % | 0 | 0 | 0 | 0  
 test | 0.00 % | 100.00 % | 0 | 0 | 0 | 0  
(5 rows)

建议:

回滚比例大说明业务逻辑可能有问题, 命中率小说明shared\_buffer要加大, 数据块读写时间长说明块设备的IO性能要提升, 死锁次数多说明业务逻辑有问题, 复制冲突次数多说明备库可能在跑LONG SQL.

----->>>---->>> 检查点, bgwriter 统计信息:   
-[ RECORD 1 ]---------+------------------------------  
checkpoints\_timed | 121  
checkpoints\_req | 0  
checkpoint\_write\_time | 0  
checkpoint\_sync\_time | 0  
buffers\_checkpoint | 0  
buffers\_clean | 0  
maxwritten\_clean | 0  
buffers\_backend | 0  
buffers\_backend\_fsync | 0  
buffers\_alloc | 0  
stats\_reset | 2022-01-05 23:43:32.065371+08

建议:

checkpoint\_write\_time多说明检查点持续时间长, 检查点过程中产生了较多的脏页.

checkpoint\_sync\_time代表检查点开始时的shared buffer中的脏页被同步到磁盘的时间, 如果时间过长, 并且数据库在检查点时性能较差, 考虑一下提升块设备的IOPS能力.

buffers\_backend\_fsync太多说明需要加大shared buffer 或者 减小bgwriter\_delay参数.

* + 1. **数据库垃圾分析**

----->>>---->>> 表引膨胀检查:   
-[ RECORD 1 ]----+----------------------------------  
db | postgres  
schemaname | pg\_catalog  
tablename | pg\_class  
tups | 342  
pages | 12  
otta | 10  
tbloat | 1.2  
wastedpages | 2  
wastedbytes | 16384  
wastedsize | 16384 bytes  
iname | pg\_class\_tblspc\_relfilenode\_index  
itups | 342  
ipages | 5  
iotta | 8  
ibloat | 0.6  
wastedipages | 0  
wastedibytes | 0  
wastedisize | 0 bytes  
totalwastedbytes | 16384  
-[ RECORD 2 ]----+----------------------------------  
db | postgres  
schemaname | pg\_catalog  
tablename | pg\_class  
tups | 342  
pages | 12  
otta | 10  
tbloat | 1.2  
wastedpages | 2  
wastedbytes | 16384  
wastedsize | 16384 bytes  
iname | pg\_class\_relname\_nsp\_index  
itups | 342  
ipages | 6  
iotta | 8  
ibloat | 0.8  
wastedipages | 0  
wastedibytes | 0  
wastedisize | 0 bytes  
totalwastedbytes | 16384  
-[ RECORD 3 ]----+----------------------------------  
db | postgres  
schemaname | pg\_catalog  
tablename | pg\_class  
tups | 342  
pages | 12  
otta | 10  
tbloat | 1.2  
wastedpages | 2  
wastedbytes | 16384  
wastedsize | 16384 bytes  
iname | pg\_class\_oid\_index  
itups | 342  
ipages | 4  
iotta | 8  
ibloat | 0.5  
wastedipages | 0  
wastedibytes | 0  
wastedisize | 0 bytes  
totalwastedbytes | 16384  
-[ RECORD 4 ]----+----------------------------------  
db | postgres  
schemaname | pg\_catalog  
tablename | pg\_operator  
tups | 788  
pages | 15  
otta | 14  
tbloat | 1.1  
wastedpages | 1  
wastedbytes | 8192  
wastedsize | 8192 bytes  
iname | pg\_operator\_oid\_index  
itups | 788  
ipages | 5  
iotta | 12  
ibloat | 0.4  
wastedipages | 0  
wastedibytes | 0  
wastedisize | 0 bytes  
totalwastedbytes | 8192  
-[ RECORD 5 ]----+----------------------------------  
db | postgres  
schemaname | pg\_catalog  
tablename | pg\_operator  
tups | 788  
pages | 15  
otta | 14  
tbloat | 1.1  
wastedpages | 1  
wastedbytes | 8192  
wastedsize | 8192 bytes  
iname | pg\_operator\_oprname\_l\_r\_n\_index  
itups | 788  
ipages | 6  
iotta | 12  
ibloat | 0.5  
wastedipages | 0  
wastedibytes | 0  
wastedisize | 0 bytes  
totalwastedbytes | 8192  
  
-[ RECORD 1 ]----+----------------------  
db | tpcc  
schemaname | public  
tablename | bmsql\_customer  
tups | 300000  
pages | 22685  
otta | 21586  
tbloat | 1.1  
wastedpages | 1099  
wastedbytes | 9003008  
wastedsize | 9003008 bytes  
iname | bmsql\_customer\_pkey  
itups | 300000  
ipages | 1158  
iotta | 20852  
ibloat | 0.1  
wastedipages | 0  
wastedibytes | 0  
wastedisize | 0 bytes  
totalwastedbytes | 9003008  
-[ RECORD 2 ]----+----------------------  
db | tpcc  
schemaname | public  
tablename | bmsql\_customer  
tups | 300000  
pages | 22685  
otta | 21586  
tbloat | 1.1  
wastedpages | 1099  
wastedbytes | 9003008  
wastedsize | 9003008 bytes  
iname | bmsql\_customer\_idx1  
itups | 300000  
ipages | 2045  
iotta | 20852  
ibloat | 0.1  
wastedipages | 0  
wastedibytes | 0  
wastedisize | 0 bytes  
totalwastedbytes | 9003008  
-[ RECORD 3 ]----+----------------------  
db | tpcc  
schemaname | public  
tablename | bmsql\_stock  
tups | 1000006  
pages | 43465  
otta | 42585  
tbloat | 1.0  
wastedpages | 880  
wastedbytes | 7208960  
wastedsize | 7208960 bytes  
iname | bmsql\_stock\_pkey  
itups | 1000000  
ipages | 2745  
iotta | 39770  
ibloat | 0.1  
wastedipages | 0  
wastedibytes | 0  
wastedisize | 0 bytes  
totalwastedbytes | 7208960  
-[ RECORD 4 ]----+----------------------  
db | tpcc  
schemaname | public  
tablename | bmsql\_order\_line  
tups | 3001535  
pages | 37058  
otta | 36413  
tbloat | 1.0  
wastedpages | 645  
wastedbytes | 5283840  
wastedsize | 5283840 bytes  
iname | bmsql\_order\_line\_pkey  
itups | 3001566  
ipages | 11560  
iotta | 27409  
ibloat | 0.4  
wastedipages | 0  
wastedibytes | 0  
wastedisize | 0 bytes  
totalwastedbytes | 5283840  
-[ RECORD 5 ]----+----------------------  
db | tpcc  
schemaname | public  
tablename | bmsql\_history  
tups | 300000  
pages | 3239  
otta | 3084  
tbloat | 1.1  
wastedpages | 155  
wastedbytes | 1269760  
wastedsize | 1269760 bytes  
iname | bmsql\_history\_pkey  
itups | 300000  
ipages | 825  
iotta | 2460  
ibloat | 0.3  
wastedipages | 0  
wastedibytes | 0  
wastedisize | 0 bytes  
totalwastedbytes | 1269760  
  
-[ RECORD 1 ]----+-------------------------------  
db | test  
schemaname | pg\_catalog  
tablename | pg\_attrdef  
tups | 696  
pages | 36  
otta | 34  
tbloat | 1.1  
wastedpages | 2  
wastedbytes | 16384  
wastedsize | 16384 bytes  
iname | pg\_attrdef\_oid\_index  
itups | 696  
ipages | 4  
iotta | 33  
ibloat | 0.1  
wastedipages | 0  
wastedibytes | 0  
wastedisize | 0 bytes  
totalwastedbytes | 16384  
-[ RECORD 2 ]----+-------------------------------  
db | test  
schemaname | pg\_catalog  
tablename | pg\_attrdef  
tups | 696  
pages | 36  
otta | 34  
tbloat | 1.1  
wastedpages | 2  
wastedbytes | 16384  
wastedsize | 16384 bytes  
iname | pg\_attrdef\_adrelid\_adnum\_index  
itups | 696  
ipages | 4  
iotta | 33  
ibloat | 0.1  
wastedipages | 0  
wastedibytes | 0  
wastedisize | 0 bytes  
totalwastedbytes | 16384  
-[ RECORD 3 ]----+-------------------------------  
db | test  
schemaname | pg\_catalog  
tablename | pg\_amproc  
tups | 458  
pages | 4  
otta | 3  
tbloat | 1.3  
wastedpages | 1  
wastedbytes | 8192  
wastedsize | 8192 bytes  
iname | pg\_amproc\_fam\_proc\_index  
itups | 458  
ipages | 4  
iotta | 2  
ibloat | 2.0  
wastedipages | 2  
wastedibytes | 16384  
wastedisize | 16384 bytes  
totalwastedbytes | 24576  
-[ RECORD 4 ]----+-------------------------------  
db | test  
schemaname | pg\_catalog  
tablename | pg\_amop  
tups | 722  
pages | 7  
otta | 6  
tbloat | 1.2  
wastedpages | 1  
wastedbytes | 8192  
wastedsize | 8192 bytes  
iname | pg\_amop\_opr\_fam\_index  
itups | 722  
ipages | 5  
iotta | 4  
ibloat | 1.3  
wastedipages | 1  
wastedibytes | 8192  
wastedisize | 8192 bytes  
totalwastedbytes | 16384  
-[ RECORD 5 ]----+-------------------------------  
db | test  
schemaname | pg\_catalog  
tablename | pg\_amop  
tups | 722  
pages | 7  
otta | 6  
tbloat | 1.2  
wastedpages | 1  
wastedbytes | 8192  
wastedsize | 8192 bytes  
iname | pg\_amop\_fam\_strat\_index  
itups | 722  
ipages | 5  
iotta | 4  
ibloat | 1.3  
wastedipages | 1  
wastedibytes | 8192  
wastedisize | 8192 bytes  
totalwastedbytes | 16384  
  
建议:   
 根据浪费的字节数, 设置合适的autovacuum\_vacuum\_scale\_factor, 大表如果频繁的有更新或删除和插入操作, 建议设置较小的autovacuum\_vacuum\_scale\_factor来降低浪费空间.   
 同时还需要打开autovacuum, 根据服务器的内存大小, CPU核数, 设置足够大的autovacuum\_work\_mem 或 autovacuum\_max\_workers 或 maintenance\_work\_mem, 以及足够小的 autovacuum\_naptime .   
 同时还需要分析是否对大数据库使用了逻辑备份pg\_dump, 系统中是否经常有长SQL, 长事务. 这些都有可能导致膨胀.   
 使用pg\_reorg或者vacuum full可以回收膨胀的空间.   
 参考: http://blog.163.com/digoal@126/blog/static/1638770402015329115636287/   
 otta评估出的表实际需要页数, iotta评估出的索引实际需要页数;   
 bs数据库的块大小;   
 tbloat表膨胀倍数, ibloat索引膨胀倍数, wastedpages表浪费了多少个数据块, wastedipages索引浪费了多少个数据块;   
 wastedbytes表浪费了多少字节, wastedibytes索引浪费了多少字节;

----->>>---->>> 索引膨胀检查:   
-[ RECORD 1 ]----+--------------------------  
db | postgres  
schemaname | pg\_catalog  
tablename | pg\_depend  
tups | 7471  
pages | 56  
otta | 55  
tbloat | 1.0  
wastedpages | 1  
wastedbytes | 8192  
wastedsize | 8192 bytes  
iname | pg\_depend\_reference\_index  
itups | 7471  
ipages | 47  
iotta | 34  
ibloat | 1.4  
wastedipages | 13  
wastedibytes | 106496  
wastedisize | 106496 bytes  
totalwastedbytes | 114688  
-[ RECORD 2 ]----+--------------------------  
db | postgres  
schemaname | pg\_catalog  
tablename | pg\_depend  
tups | 7471  
pages | 56  
otta | 55  
tbloat | 1.0  
wastedpages | 1  
wastedbytes | 8192  
wastedsize | 8192 bytes  
iname | pg\_depend\_depender\_index  
itups | 7471  
ipages | 42  
iotta | 34  
ibloat | 1.2  
wastedipages | 8  
wastedibytes | 65536  
wastedisize | 65536 bytes  
totalwastedbytes | 73728  
-[ RECORD 3 ]----+--------------------------  
db | postgres  
schemaname | pg\_catalog  
tablename | pg\_amproc  
tups | 458  
pages | 4  
otta | 3  
tbloat | 1.3  
wastedpages | 1  
wastedbytes | 8192  
wastedsize | 8192 bytes  
iname | pg\_amproc\_fam\_proc\_index  
itups | 458  
ipages | 4  
iotta | 2  
ibloat | 2.0  
wastedipages | 2  
wastedibytes | 16384  
wastedisize | 16384 bytes  
totalwastedbytes | 24576  
-[ RECORD 4 ]----+--------------------------  
db | postgres  
schemaname | pg\_catalog  
tablename | pg\_amproc  
tups | 458  
pages | 4  
otta | 3  
tbloat | 1.3  
wastedpages | 1  
wastedbytes | 8192  
wastedsize | 8192 bytes  
iname | pg\_amproc\_oid\_index  
itups | 458  
ipages | 4  
iotta | 2  
ibloat | 2.0  
wastedipages | 2  
wastedibytes | 16384  
wastedisize | 16384 bytes  
totalwastedbytes | 24576  
-[ RECORD 5 ]----+--------------------------  
db | postgres  
schemaname | pg\_catalog  
tablename | pg\_ts\_config\_map  
tups | 304  
pages | 2  
otta | 2  
tbloat | 0.0  
wastedpages | 0  
wastedbytes | 0  
wastedsize | 0 bytes  
iname | pg\_ts\_config\_map\_index  
itups | 304  
ipages | 4  
iotta | 2  
ibloat | 2.0  
wastedipages | 2  
wastedibytes | 16384  
wastedisize | 16384 bytes  
totalwastedbytes | 16384  
  
-[ RECORD 1 ]----+--------------------------  
db | tpcc  
schemaname | public  
tablename | bmsql\_new\_order  
tups | 90000  
pages | 487  
otta | 485  
tbloat | 1.0  
wastedpages | 2  
wastedbytes | 16384  
wastedsize | 16384 bytes  
iname | bmsql\_new\_order\_pkey  
itups | 90000  
ipages | 349  
iotta | 265  
ibloat | 1.3  
wastedipages | 84  
wastedibytes | 688128  
wastedisize | 688128 bytes  
totalwastedbytes | 704512  
-[ RECORD 2 ]----+--------------------------  
db | tpcc  
schemaname | pg\_catalog  
tablename | pg\_depend  
tups | 7640  
pages | 57  
otta | 57  
tbloat | 0.0  
wastedpages | 0  
wastedbytes | 0  
wastedsize | 0 bytes  
iname | pg\_depend\_reference\_index  
itups | 7471  
ipages | 47  
iotta | 34  
ibloat | 1.4  
wastedipages | 13  
wastedibytes | 106496  
wastedisize | 106496 bytes  
totalwastedbytes | 106496  
-[ RECORD 3 ]----+--------------------------  
db | tpcc  
schemaname | pg\_catalog  
tablename | pg\_depend  
tups | 7640  
pages | 57  
otta | 57  
tbloat | 0.0  
wastedpages | 0  
wastedbytes | 0  
wastedsize | 0 bytes  
iname | pg\_depend\_depender\_index  
itups | 7471  
ipages | 42  
iotta | 34  
ibloat | 1.2  
wastedipages | 8  
wastedibytes | 65536  
wastedisize | 65536 bytes  
totalwastedbytes | 65536  
-[ RECORD 4 ]----+--------------------------  
db | tpcc  
schemaname | pg\_catalog  
tablename | pg\_amproc  
tups | 458  
pages | 4  
otta | 3  
tbloat | 1.3  
wastedpages | 1  
wastedbytes | 8192  
wastedsize | 8192 bytes  
iname | pg\_amproc\_oid\_index  
itups | 458  
ipages | 4  
iotta | 2  
ibloat | 2.0  
wastedipages | 2  
wastedibytes | 16384  
wastedisize | 16384 bytes  
totalwastedbytes | 24576  
-[ RECORD 5 ]----+--------------------------  
db | tpcc  
schemaname | pg\_catalog  
tablename | pg\_amproc  
tups | 458  
pages | 4  
otta | 3  
tbloat | 1.3  
wastedpages | 1  
wastedbytes | 8192  
wastedsize | 8192 bytes  
iname | pg\_amproc\_fam\_proc\_index  
itups | 458  
ipages | 4  
iotta | 2  
ibloat | 2.0  
wastedipages | 2  
wastedibytes | 16384  
wastedisize | 16384 bytes  
totalwastedbytes | 24576  
  
-[ RECORD 1 ]----+--------------------------  
db | test  
schemaname | pg\_catalog  
tablename | pg\_depend  
tups | 9631  
pages | 71  
otta | 71  
tbloat | 0.0  
wastedpages | 0  
wastedbytes | 0  
wastedsize | 0 bytes  
iname | pg\_depend\_reference\_index  
itups | 9631  
ipages | 59  
iotta | 44  
ibloat | 1.3  
wastedipages | 15  
wastedibytes | 122880  
wastedisize | 122880 bytes  
totalwastedbytes | 122880  
-[ RECORD 2 ]----+--------------------------  
db | test  
schemaname | pg\_catalog  
tablename | pg\_depend  
tups | 9631  
pages | 71  
otta | 71  
tbloat | 0.0  
wastedpages | 0  
wastedbytes | 0  
wastedsize | 0 bytes  
iname | pg\_depend\_depender\_index  
itups | 9631  
ipages | 57  
iotta | 44  
ibloat | 1.3  
wastedipages | 13  
wastedibytes | 106496  
wastedisize | 106496 bytes  
totalwastedbytes | 106496  
-[ RECORD 3 ]----+--------------------------  
db | test  
schemaname | pg\_catalog  
tablename | pg\_amproc  
tups | 458  
pages | 4  
otta | 3  
tbloat | 1.3  
wastedpages | 1  
wastedbytes | 8192  
wastedsize | 8192 bytes  
iname | pg\_amproc\_fam\_proc\_index  
itups | 458  
ipages | 4  
iotta | 2  
ibloat | 2.0  
wastedipages | 2  
wastedibytes | 16384  
wastedisize | 16384 bytes  
totalwastedbytes | 24576  
-[ RECORD 4 ]----+--------------------------  
db | test  
schemaname | pg\_catalog  
tablename | pg\_amproc  
tups | 458  
pages | 4  
otta | 3  
tbloat | 1.3  
wastedpages | 1  
wastedbytes | 8192  
wastedsize | 8192 bytes  
iname | pg\_amproc\_oid\_index  
itups | 458  
ipages | 4  
iotta | 2  
ibloat | 2.0  
wastedipages | 2  
wastedibytes | 16384  
wastedisize | 16384 bytes  
totalwastedbytes | 24576  
-[ RECORD 5 ]----+--------------------------  
db | test  
schemaname | pg\_catalog  
tablename | pg\_ts\_config\_map  
tups | 304  
pages | 2  
otta | 2  
tbloat | 0.0  
wastedpages | 0  
wastedbytes | 0  
wastedsize | 0 bytes  
iname | pg\_ts\_config\_map\_index  
itups | 304  
ipages | 4  
iotta | 2  
ibloat | 2.0  
wastedipages | 2  
wastedibytes | 16384  
wastedisize | 16384 bytes  
totalwastedbytes | 16384

建议:

如果索引膨胀太大, 会影响性能, 建议重建索引, create index CONCURRENTLY ... .

----->>>---->>> 垃圾数据:   
 current\_database | schemaname | relname | n\_dead\_tup   
------------------+------------+---------+------------  
(0 rows)  
  
 current\_database | schemaname | relname | n\_dead\_tup   
------------------+------------+---------+------------  
(0 rows)  
  
 current\_database | schemaname | relname | n\_dead\_tup   
------------------+------------+---------+------------  
(0 rows)  
  
建议:   
 通常垃圾过多, 可能是因为无法回收垃圾, 或者回收垃圾的进程繁忙或没有及时唤醒, 或者没有开启autovacuum, 或在短时间内产生了大量的垃圾 .   
 可以等待autovacuum进行处理, 或者手工执行vacuum table .

* + 1. **数据库年龄分析**

----->>>---->>> 数据库年龄:   
 datname | age | age\_remain   
-----------+-----+------------  
 postgres | 969 | 2147482679  
 tpcc | 969 | 2147482679  
 template1 | 969 | 2147482679  
 template0 | 969 | 2147482679  
 test | 969 | 2147482679  
(5 rows)

建议:

数据库的年龄正常情况下应该小于vacuum\_freeze\_table\_age, 如果剩余年龄小于5亿, 建议人为干预, 将LONG SQL或事务杀掉后, 执行vacuum freeze .

----->>>---->>> 表年龄:   
 current\_database | rolname | nspname | relkind | relname | age | age\_remain   
------------------+----------+------------+---------+---------------+-----+------------  
 postgres | postgres | pg\_toast | t | pg\_toast\_2609 | 969 | 2147482679  
 postgres | postgres | pg\_toast | t | pg\_toast\_2606 | 969 | 2147482679  
 postgres | postgres | pg\_catalog | r | pg\_statistic | 969 | 2147482679  
 postgres | postgres | pg\_toast | t | pg\_toast\_2604 | 969 | 2147482679  
 postgres | postgres | pg\_toast | t | pg\_toast\_1255 | 969 | 2147482679  
(5 rows)  
  
 current\_database | rolname | nspname | relkind | relname | age | age\_remain   
------------------+----------+------------+---------+---------------+-----+------------  
 tpcc | postgres | pg\_toast | t | pg\_toast\_2609 | 969 | 2147482679  
 tpcc | postgres | pg\_toast | t | pg\_toast\_2606 | 969 | 2147482679  
 tpcc | postgres | pg\_toast | t | pg\_toast\_2604 | 969 | 2147482679  
 tpcc | postgres | pg\_catalog | r | pg\_statistic | 969 | 2147482679  
 tpcc | postgres | pg\_toast | t | pg\_toast\_1255 | 969 | 2147482679  
(5 rows)  
  
 current\_database | rolname | nspname | relkind | relname | age | age\_remain   
------------------+----------+------------+---------+---------------+-----+------------  
 test | postgres | pg\_toast | t | pg\_toast\_2609 | 969 | 2147482679  
 test | postgres | pg\_toast | t | pg\_toast\_2606 | 969 | 2147482679  
 test | postgres | pg\_toast | t | pg\_toast\_2604 | 969 | 2147482679  
 test | postgres | pg\_catalog | r | pg\_statistic | 969 | 2147482679  
 test | postgres | pg\_toast | t | pg\_toast\_1255 | 969 | 2147482679  
(5 rows)

建议:

表的年龄正常情况下应该小于vacuum\_freeze\_table\_age, 如果剩余年龄小于5亿, 建议人为干预, 将LONG SQL或事务杀掉后, 执行vacuum freeze .

----->>>---->>> 长事务, 2PC:   
(0 rows)  
  
(0 rows)

建议:

长事务过程中产生的垃圾, 无法回收, 建议不要在数据库中运行LONG SQL, 或者错开DML高峰时间去运行LONG SQL. 2PC事务一定要记得尽快结束掉, 否则可能会导致数据库膨胀.

参考: http://blog.163.com/digoal@126/blog/static/1638770402015329115636287/

* + 1. **数据库归档与流复制**

----->>>---->>> 是否开启归档, 自动垃圾回收:   
 name | setting   
-----------------+------------  
 archive\_command | (disabled)  
 archive\_mode | off  
 autovacuum | on  
(3 rows)

建议:

建议开启自动垃圾回收, 开启归档.

----->>>---->>> 归档统计信息:   
 now\_xlog | archived\_count | last\_archived\_wal | last\_archived\_time | failed\_count | last\_failed\_wal | last\_failed\_time | stats\_reset   
--------------------------+----------------+-------------------+--------------------+--------------+-----------------+------------------+-------------------------------  
 000000010000000000000051 | 0 | | | 0 | | | 2022-01-05 23:43:32.068857+08  
(1 row)  
  
建议:   
 如果当前的XLOG文件和最后一个归档失败的XLOG文件之间相差很多个文件, 建议尽快排查归档失败的原因, 以便修复, 否则pg\_xlog目录可能会撑爆.

----->>>---->>> 流复制统计信息:   
(0 rows)

建议:

关注流复制的延迟, 如果延迟非常大, 建议排查网络带宽, 以及本地读xlog的性能, 远程写xlog的性能.

----->>>---->>> 流复制插槽:   
 pg\_wal\_lsn\_diff | slot\_name | plugin | slot\_type | datoid | database | temporary | active | active\_pid | xmin | catalog\_xmin | restart\_lsn | confirmed\_flush\_lsn   
-----------------+-----------+--------+-----------+--------+----------+-----------+--------+------------+------+--------------+-------------+---------------------  
(0 rows)  
  
建议:   
 如果restart\_lsn和当前XLOG相差非常大的字节数, 需要排查slot的订阅者是否能正常接收XLOG, 或者订阅者是否正常. 长时间不将slot的数据取走, pg\_xlog目录可能会撑爆.

* + 1. **数据库风险分析**

----->>>---->>> 密码泄露检查:   
 检查 ~/.psql\_history :   
create user tpcc superuser password '1qaz!QAZ';  
create user grp\_nd with password '1qaz!QAZ';  
create user lis with password '1qaz!QAZ';  
create user test with password '1222222';  
 检查 \*.csv :   
VM-16-10-centos\_15432/12.1\_psql\_history.txt:create user tpcc superuser password '1qaz!QAZ';  
VM-16-10-centos\_15432/12.1\_psql\_history.txt:create user grp\_nd with password '1qaz!QAZ';  
VM-16-10-centos\_15432/12.1\_psql\_history.txt:create user lis with password '1qaz!QAZ';  
VM-16-10-centos\_15432/12.1\_psql\_history.txt:create user test with password '1222222';  
pg\_all.sh:cat $PGDATA/${pg\_log\_dir}/postgresql-${this\_month}\*.log | grep -E "^[0-9]" | grep -i -r -E "role|group|user" |grep -i "password"|grep -i -E "create|alter" >>$dir\_txt/12.1\_psql\_history.txt  
 检查 /home/postgres/data/recovery.\* :   
 检查 pg\_stat\_statements :   
 query   
-------  
(0 rows)  
  
 检查 pg\_authid :   
 rolname | rolsuper | rolinherit | rolcreaterole | rolcreatedb | rolcanlogin | rolreplication | rolbypassrls | rolconnlimit | rolpassword | rolvaliduntil   
---------+----------+------------+---------------+-------------+-------------+----------------+--------------+--------------+-------------+---------------  
(0 rows)  
  
 检查 pg\_user\_mappings, pg\_views :   
 current\_database | umid | srvid | srvname | umuser | usename | umoptions   
------------------+------+-------+---------+--------+---------+-----------  
(0 rows)  
  
 current\_database | schemaname | viewname | viewowner | definition   
------------------+------------+----------+-----------+------------  
(0 rows)  
  
 current\_database | umid | srvid | srvname | umuser | usename | umoptions   
------------------+------+-------+---------+--------+---------+-----------  
(0 rows)  
  
 current\_database | schemaname | viewname | viewowner | definition   
------------------+------------+----------+-----------+------------  
(0 rows)  
  
 current\_database | umid | srvid | srvname | umuser | usename | umoptions   
------------------+------+-------+---------+--------+---------+-----------  
(0 rows)  
  
 current\_database | schemaname | viewname | viewowner | definition   
------------------+------------+----------+-----------+------------  
(0 rows)

建议:

如果以上输出显示密码已泄露, 尽快修改, 并通过参数避免密码又被记录到以上文件中(psql -n) (set log\_statement='none'; set log\_min\_duration\_statement=-1; set log\_duration=off; set pg\_stat\_statements.track\_utility=off;) .

明文密码不安全, 建议使用create|alter role ... encrypted password.

----->>>---->>> 用户密码到期时间:   
 rolname | rolvaliduntil   
---------------------------+------------------------  
 lis | 2019-04-10 16:58:00+08  
 pg\_monitor |   
 pg\_read\_all\_settings |   
 pg\_read\_all\_stats |   
 pg\_stat\_scan\_tables |   
 pg\_read\_server\_files |   
 postgres |   
 pg\_execute\_server\_program |   
 pg\_signal\_backend |   
 tpcc |   
 grp\_nd |   
 pg\_write\_server\_files |   
(12 rows)

建议:

到期后, 用户将无法登陆, 记得修改密码, 同时将密码到期时间延长到某个时间或无限时间，暂不需要修改。

----->>>---->>> 普通用户对象上的规则安全检查:   
 current\_database | schemaname | tablename | rulename | definition   
------------------+------------+-----------+----------+------------  
(0 rows)  
  
 current\_database | schemaname | tablename | rulename | definition   
------------------+------------+-----------+----------+------------  
(0 rows)  
  
 current\_database | schemaname | tablename | rulename | definition   
------------------+------------+-----------+----------+------------  
(0 rows)

建议:

防止普通用户在规则中设陷阱, 注意有危险的security invoker的函数调用, 超级用户可能因为规则触发后误调用这些危险函数(以invoker角色).

----->>>---->>> 普通用户自定义函数安全检查:   
 current\_database | rolname | nspname | proname   
------------------+---------+---------+---------  
(0 rows)  
  
 current\_database | rolname | nspname | proname   
------------------+---------+---------+---------  
(0 rows)  
  
 current\_database | rolname | nspname | proname   
------------------+---------+---------+---------  
(0 rows)

建议:

防止普通用户在函数中设陷阱, 注意有危险的security invoker的函数调用, 超级用户可能因为触发器触发后误调用这些危险函数(以invoker角色).

----->>>---->>> unlogged table 和 哈希索引:   
 current\_database | rolname | nspname | relname   
------------------+---------+---------+---------  
(0 rows)  
  
 current\_database | pg\_get\_indexdef   
------------------+-----------------  
(0 rows)  
  
 current\_database | rolname | nspname | relname   
------------------+---------+---------+---------  
(0 rows)  
  
 current\_database | pg\_get\_indexdef   
------------------+-----------------  
(0 rows)  
  
 current\_database | rolname | nspname | relname   
------------------+---------+---------+---------  
(0 rows)  
  
 current\_database | pg\_get\_indexdef   
------------------+-----------------  
(0 rows)

建议

unlogged table和hash index不记录XLOG, 无法使用流复制或者log shipping的方式复制到standby节点, 如果在standby节点执行某些SQL, 可能导致报错或查不到数据

在数据库CRASH后无法修复unlogged table和hash index, 不建议使用.

----->>>---->>> 触发器, 事件触发器:   
 current\_database | relname | tgname | proname | tgenabled   
------------------+----------------------+----------------+-----------------+-----------  
 postgres | test\_trigger\_src\_tbl | insert\_trigger | tri\_insert\_func | O  
(1 row)  
  
 current\_database | rolname | proname | evtname | evtevent | evtenabled | evttags   
------------------+---------+---------+---------+----------+------------+---------  
(0 rows)  
  
 current\_database | relname | tgname | proname | tgenabled   
------------------+------------------+------------------------------+----------------------+-----------  
 tpcc | bmsql\_warehouse | RI\_ConstraintTrigger\_a\_16486 | RI\_FKey\_noaction\_upd | O  
 tpcc | bmsql\_warehouse | RI\_ConstraintTrigger\_a\_16485 | RI\_FKey\_noaction\_del | O  
 tpcc | bmsql\_warehouse | RI\_ConstraintTrigger\_a\_16445 | RI\_FKey\_noaction\_upd | O  
 tpcc | bmsql\_warehouse | RI\_ConstraintTrigger\_a\_16444 | RI\_FKey\_noaction\_del | O  
 tpcc | bmsql\_district | RI\_ConstraintTrigger\_a\_16461 | RI\_FKey\_noaction\_upd | O  
 tpcc | bmsql\_district | RI\_ConstraintTrigger\_a\_16460 | RI\_FKey\_noaction\_del | O  
 tpcc | bmsql\_district | RI\_ConstraintTrigger\_a\_16450 | RI\_FKey\_noaction\_upd | O  
 tpcc | bmsql\_district | RI\_ConstraintTrigger\_a\_16449 | RI\_FKey\_noaction\_del | O  
 tpcc | bmsql\_district | RI\_ConstraintTrigger\_c\_16447 | RI\_FKey\_check\_upd | O  
 tpcc | bmsql\_district | RI\_ConstraintTrigger\_c\_16446 | RI\_FKey\_check\_ins | O  
 tpcc | bmsql\_customer | RI\_ConstraintTrigger\_a\_16471 | RI\_FKey\_noaction\_upd | O  
 tpcc | bmsql\_customer | RI\_ConstraintTrigger\_a\_16470 | RI\_FKey\_noaction\_del | O  
 tpcc | bmsql\_customer | RI\_ConstraintTrigger\_a\_16456 | RI\_FKey\_noaction\_upd | O  
 tpcc | bmsql\_customer | RI\_ConstraintTrigger\_a\_16455 | RI\_FKey\_noaction\_del | O  
 tpcc | bmsql\_customer | RI\_ConstraintTrigger\_c\_16452 | RI\_FKey\_check\_upd | O  
 tpcc | bmsql\_customer | RI\_ConstraintTrigger\_c\_16451 | RI\_FKey\_check\_ins | O  
 tpcc | bmsql\_history | RI\_ConstraintTrigger\_c\_16463 | RI\_FKey\_check\_upd | O  
 tpcc | bmsql\_history | RI\_ConstraintTrigger\_c\_16462 | RI\_FKey\_check\_ins | O  
 tpcc | bmsql\_history | RI\_ConstraintTrigger\_c\_16458 | RI\_FKey\_check\_upd | O  
 tpcc | bmsql\_history | RI\_ConstraintTrigger\_c\_16457 | RI\_FKey\_check\_ins | O  
 tpcc | bmsql\_new\_order | RI\_ConstraintTrigger\_c\_16468 | RI\_FKey\_check\_upd | O  
 tpcc | bmsql\_new\_order | RI\_ConstraintTrigger\_c\_16467 | RI\_FKey\_check\_ins | O  
 tpcc | bmsql\_order\_line | RI\_ConstraintTrigger\_c\_16483 | RI\_FKey\_check\_upd | O  
 tpcc | bmsql\_order\_line | RI\_ConstraintTrigger\_c\_16482 | RI\_FKey\_check\_ins | O  
 tpcc | bmsql\_order\_line | RI\_ConstraintTrigger\_c\_16478 | RI\_FKey\_check\_upd | O  
 tpcc | bmsql\_order\_line | RI\_ConstraintTrigger\_c\_16477 | RI\_FKey\_check\_ins | O  
 tpcc | bmsql\_stock | RI\_ConstraintTrigger\_c\_16493 | RI\_FKey\_check\_upd | O  
 tpcc | bmsql\_stock | RI\_ConstraintTrigger\_c\_16492 | RI\_FKey\_check\_ins | O  
 tpcc | bmsql\_stock | RI\_ConstraintTrigger\_c\_16488 | RI\_FKey\_check\_upd | O  
 tpcc | bmsql\_stock | RI\_ConstraintTrigger\_c\_16487 | RI\_FKey\_check\_ins | O  
 tpcc | bmsql\_stock | RI\_ConstraintTrigger\_a\_16481 | RI\_FKey\_noaction\_upd | O  
 tpcc | bmsql\_stock | RI\_ConstraintTrigger\_a\_16480 | RI\_FKey\_noaction\_del | O  
 tpcc | bmsql\_item | RI\_ConstraintTrigger\_a\_16491 | RI\_FKey\_noaction\_upd | O  
 tpcc | bmsql\_item | RI\_ConstraintTrigger\_a\_16490 | RI\_FKey\_noaction\_del | O  
 tpcc | bmsql\_oorder | RI\_ConstraintTrigger\_a\_16476 | RI\_FKey\_noaction\_upd | O  
 tpcc | bmsql\_oorder | RI\_ConstraintTrigger\_a\_16475 | RI\_FKey\_noaction\_del | O  
 tpcc | bmsql\_oorder | RI\_ConstraintTrigger\_c\_16473 | RI\_FKey\_check\_upd | O  
 tpcc | bmsql\_oorder | RI\_ConstraintTrigger\_c\_16472 | RI\_FKey\_check\_ins | O  
 tpcc | bmsql\_oorder | RI\_ConstraintTrigger\_a\_16466 | RI\_FKey\_noaction\_upd | O  
 tpcc | bmsql\_oorder | RI\_ConstraintTrigger\_a\_16465 | RI\_FKey\_noaction\_del | O  
(40 rows)  
  
 current\_database | rolname | proname | evtname | evtevent | evtenabled | evttags   
------------------+---------+---------+---------+----------+------------+---------  
(0 rows)  
  
 current\_database | relname | tgname | proname | tgenabled   
------------------+---------+--------+---------+-----------  
(0 rows)  
  
 current\_database | rolname | proname | evtname | evtevent | evtenabled | evttags   
------------------+---------+---------+---------+----------+------------+---------  
(0 rows)

建议

请管理员注意触发器和事件触发器的必要性.

----->>>---->>> 检查是否使用了a-z 0-9 \_ 以外的字母作为对象名:   
 current\_database | relname | relkind   
------------------+---------+---------  
(0 rows)  
  
 current\_database | typname   
------------------+---------  
(0 rows)  
  
 current\_database | proname   
------------------+---------  
(0 rows)  
  
 current\_database | nspname | relname | attname   
------------------+---------+---------+---------  
(0 rows)  
  
 current\_database | relname | relkind   
------------------+---------+---------  
(0 rows)  
  
 current\_database | typname   
------------------+---------  
(0 rows)  
  
 current\_database | proname   
------------------+---------  
(0 rows)  
  
 current\_database | nspname | relname | attname   
------------------+---------+---------+---------  
(0 rows)  
  
 current\_database | relname | relkind   
------------------+---------+---------  
(0 rows)  
  
 current\_database | typname   
------------------+---------  
(0 rows)  
  
 current\_database | proname   
------------------+---------  
(0 rows)  
  
 current\_database | nspname | relname | attname   
------------------+---------+---------+---------  
(0 rows)

建议：

建议任何identify都只使用 a-z, 0-9, \_ (例如表名, 列名, 视图名, 函数名, 类型名, 数据库名, schema名, 物化视图名等等).

----->>>---->>> 锁等待:   
(0 rows)

建议:

锁等待状态, 反映业务逻辑的问题或者SQL性能有问题, 建议深入排查持锁的SQL.

----->>>---->>> 继承关系检查:   
 inhrelid | inhparent | inhseqno   
----------+-----------+----------  
(0 rows)  
  
 inhrelid | inhparent | inhseqno   
----------+-----------+----------  
(0 rows)  
  
 inhrelid | inhparent | inhseqno   
----------+-----------+----------  
(0 rows)

建议:

如果使用继承来实现分区表, 注意分区表的触发器中逻辑是否正常, 对于时间模式的分区表是否需要及时加分区, 修改触发器函数 . "

建议继承表的权限统一, 如果权限不一致, 可能导致某些用户查询时权限不足.

* + 1. **重置统计信息**

----->>>---->>> 重置统计信息:   
 pg\_stat\_reset   
---------------  
   
(1 row)  
  
 pg\_stat\_reset   
---------------  
   
(1 row)  
  
 pg\_stat\_reset   
---------------  
   
(1 row)  
  
 pg\_stat\_reset\_shared   
----------------------  
   
(1 row)  
  
 pg\_stat\_reset\_shared   
----------------------  
   
(1 row)  
  
----->>>---->>> 重置pg\_stat\_statements统计信息:   
pg\_stat\_statements\_reset  
  
(1 row)

建议: