PostgreSQL一些运维优化

2018.06

杭州乘数科技
www.cstech.ltd

唐成（网名osdba）

《PostgreSQL修炼之道：从小工到专家》的作者，PostgreSQL中国用户会常务委员，中国开源软件推进联盟PostgreSQL分会特聘专家。从业近20年，拥有十几年数据库、操作系统、存储领域的工作经验，历任过阿里巴巴高级数据库专家，从事过阿里巴巴Greenplum、PostgreSQL、 MySQL数据库的架构设计和运维。

既熟悉数据库的运维工作，是最早的Oracle 9i的OCP，又懂开发，精通Java和python，擅长使用Java和python写数据迁移任务。

目录

优化准则

两种思路

第一种：“The fastest way to do something is don‘t do it”，把一些无用的步骤或用处不大的步骤去掉就是一种优化

第二种：就是做同样的一件事情，能够更快地做。更快地做，有多种方法，最简单的方法就是换硬件，让数据库跑在更快的硬件上。但换硬件，一般都是最后的选择，最有效的方法是优化算法，如让SQL走到更优的执行计划上。

衡量指标

响应时间：衡量数据库系统与用户交互式多久能够发出响应。

吞吐量：衡量在单位时间里可以完成的数据库任务。

优化准则

优化目标

性能目标：如CPU利用率或IOPS需要降到多少。每秒处理的SQL数或QPS需要提高到多少。

响应时间：需要从多少毫秒降低到多少。

吞吐量：每秒处理的SQL数或QPS需要提高到多少。

设计优先

一个已运行的数据库系统中，如果前期设计不合理，性能不高，后期在优化时，会很痛苦的，有可能永远无法达到高性能，因此，在新建一套数据库系统前，最先做的事应是设计优化。良好的设计能最大限度地发挥系统的性能

基准测试工具

基准测试工具

内存测试工具

- memtest86+
- stream

IO测试工具

- fio
- orion：Oracle 的I/O测试工具

数据库测试工具

- pgbench
- sysbench: https://github.com/osdba/sysbench_bin

硬件优化

- CPU及内存
  - 超线程
  - 几路服务器？
  - 128G或更大？
- Raid卡
  - 带写cache
  - Raid10 VS Raid 5
- 硬盘
  - SATA硬盘
  - SAS硬盘
  - SATA接口的SSD
  - PCIE接口的SSD
  - NVME

目录

文件系统及IO

Ext3及Ext4

日志模式：data=writeback,ordered, journal

建议使用data=ordered

XFS

推荐使用

Barriers

Barriers请求之前的所有在队列中的请求必须在Barries请求开始前被结束，并持久化到非易失性介质中。

Barriers请求之后的I/O需要等到其写入完成后才能得到执行

ext3/ext4 mount -o barriers=0 /dev/sdb1 /data/pgdata

XFS mount -o nobarrier /dev/sdb1 /data/pgdata

文件系统及IO

noatime

/dev/sdd1 / xfs noatime,errors=remount-ro 0 1

IO scheduler

如果不是SSD的话，还是使用CFQ，否则建议使用DEADLINE。

deadline

echo deadline > /sys/block/sda/queue/scheduler

永久设置 编辑grub文件修改块设备调度策略

vi /boot/grub.conf

elevator=deadline

关闭透明大页、numa

加上前面的默认IO调度，如下

vim /boot/grub.conf

elevator=deadline numa=off transparent_hugepage=never

操作系统参数优化

vi /etc/sysctl.conf

#此参数限制并发未完成的异步请求数目，应该设置避免I/O子系统故障

fs.aio-max-nr = 1048576

#该参数决定了系统中所允许的文件句柄最大数目，文件句柄设置代表linux系统中可以打开的文件的数量

fs.file-max = 76724600

kernel.sem = 20 13000 20 650

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。

从左至右 依次是 SEMMSL SEMMNS SEMOPM SEMMNI

SEMMSL 单个信号集中容纳最大信号数量

SEMMNS 整个系统中所有信号的最大数量

SEMOPM 在单个信号集中可操作的最大信号数量

SEMMNI 信号集的最大值：进程数/16

操作系统参数优化

kernel.shmall = 67108864
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

安全配置

配置OS防火墙

selinux

如果没有这方面的需求，建议禁用
# vim /etc/sysconfig/selinux
SELINUX=disabled
SELINUXTYPE=targeted

关闭不必要的OS服务
chkconfig --list|grep on
关闭不必要的,例如
chkconfig iscsi off

OS资源限制

vim /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

关注一下/etc/security/limits.d目录中的文件内容，会覆盖limits.conf的配置。

目录

pg参数

```
$ vi postgresql.conf
以PostgreSQL 9.6, 512G内存主机为例
listen_addresses = '0.0.0.0'
port = 6532
max_connections = 1000
unix_socket_directories = '.'
tcp_keepalives_idle = 60
tcp_keepalives_interval = 10
tcp_keepalives_count = 10
shared_buffers = 128GB                  # 1/4 主机内存
maintenance_work_mem = 2GB       # min( 2G, (1/4 主机内存)/autovacuum_max_workers )
dynamic_shared_memory_type = posix
vacuum_cost_delay = 0
bgwriter_delay = 10ms
bgwriter_lru_maxpages = 1000
bgwriter_lru_multiplier = 10.0
bgwriter_flush_after = 0                # IO很好的机器，不需要考虑平滑调度
```

pg参数

max_worker_processes = 128
max_parallel_workers_per_gather = 0　　# 如果需要使用并行查询，设置为大于1，不建议超过 主机cores-2
old_snapshot_threshold = -1
backend_flush_after = 0 # IO很好的机器，不需要考虑平滑调度, 否则建议128~256kB
wal_level = replica
synchronous_commit = off #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). on表示表示事务提交需等待 WAL 刷到磁盘后才返回成功信息,设置成 off 时，事务提交不需等待 WAL 刷到磁盘就返回成功信息 on模式比off模式TPS约36%
full_page_writes = on # 支持原子写超过BLOCK_SIZE的块设备，在对齐后可以关闭。或者支持cow的文件系统可以关闭。
wal_compression = on #开启压缩
wal_buffers = 1GB　　# min(2047MB, shared_buffers/32) = 512MB
wal_writer_delay = 10ms
al_writer_flush_after = 0 # IO很好的机器，不需要考虑平滑调度, 否则建议128~256kB
checkpoint_timeout = 30min # 不建议频繁做检查点，否则XLOG会产生很多的FULL PAGE WRITE(when full_page_writes=on)。

pg参数

max_wal_size = 256GB　　# 建议是SHARED BUFFER的2倍
min_wal_size = 64GB　　# max_wal_size/4
checkpoint_completion_target = 0.05　# 硬盘好的情况下，可以让检查点快速结束，恢复时也可以快速达到一致状态。否则建议0.5~0.9
checkpoint_flush_after = 0　　# IO很好的机器，不需要考虑平滑调度, 否则建议128~256kB
archive_mode = on
archive_command = '/bin/date'　　#　后期再修改如 'archive.sh %p %f'
wal_keep_segments =　1000　#如果过小,那么在没有archive_log、而master很繁忙产生大量数据的前提下,wal log会丢失，复制用到的连接会被断掉的
max_wal_senders = 8
random_page_cost = 1.3　# IO很好的机器，不需要考虑离散和顺序扫描的成本差异
parallel_tuple_cost = 0
parallel_setup_cost = 0
min_parallel_relation_size = 0
effective_cache_size = 300GB　# 是postgresql能够使用的最大缓存 扣掉会话连接RSS，shared buffer,autovacuum worker,剩下的都是OS可用的CACHE。
force_parallel_mode = off
log_destination = 'csvlog'

pg参数

logging_collector = on
log_truncate_on_rotation = on
log_checkpoints = on
log_connections = on
log_disconnections = on
log_error_verbosity = verbose
log_timezone = 'PRC'
vacuum_defer_cleanup_age = 0
hot_standby_feedback = off　# 建议关闭，以免备库长事务导致 主库无法回收垃圾而膨胀。
max_standby_archive_delay = 300s
max_standby_streaming_delay = 300s
hot_standby = on　#在恢复是可以查询 左右做备份或者切换时不用该这个参数了
autovacuum = on
log_autovacuum_min_duration = 0
autovacuum_max_workers = 16　# CPU核多，并且IO好的情况下，可多点，但是注意16*autovacuum mem，会消耗较多内存，所以内存也要有基础。
autovacuum_naptime = 45s　　# 建议不要太高频率，否则会因为vacuum产生较多的XLOG。

pg参数

autovacuum_vacuum_scale_factor = 0.1
autovacuum_analyze_scale_factor = 0.1
autovacuum_freeze_max_age = 1600000000
autovacuum_multixact_freeze_max_age = 1600000000
vacuum_freeze_table_age = 1500000000
vacuum_multixact_freeze_table_age = 1500000000
datestyle = 'iso, mdy'
timezone = 'PRC'
lc_messages = 'C'
lc_monetary = 'C'
lc_numeric = 'C'
lc_time = 'C'
default_text_search_config = 'pg_catalog.english'
shared_preload_libraries='pg_stat_statements'

pg参数

配置pg_hba.conf

避免不必要的访问，开放允许的访问，建议务必使用密码访问。

$ vi pg_hba.conf

host replication pguser x.x.x.x/32 md5 # 指定ip流复制

host all postgres 0.0.0.0/0 reject # 拒绝超级用户从网络登录
host all all x.x.x.0/24 md5 # 其他用户登陆 从一个网段 或者指定ip

需要访问线上的数据库可以安装phpPgAdmin
http://phppgadmin.sourceforge.net/doku.php

Q&A