【性能深度优化】MongoDB

前言

MongoDB是一个高性能可扩展基于文档的NoSQL数据库，高性能也需要在多个关键维度的配置，包括硬件、应用模式、模式设计、索引、磁盘I/O等。

存储引擎

WiredTiger是3.0以后的默认存储引擎，细粒度的并发控制和数据压缩提供了更高的性能和存储效率。3.0以前默认的MMAPv1也提高了性能。在MongoDB复制集中可以组合多钟存储引擎，各个实例实现不同的应用需求。

硬件说明

MongoDB初衷是采用水平扩展构建集群，而不是价格更高的硬件升级。采用复制保证高可用，自动分片使数据均匀分布在各节点服务器上；in-memory计算提供更高的计算性能，该引擎只有企业版有。

硬件配置要求

- 确保内存设置能满足性能需求：确保内存>索引容量+高频访问数据容量，通过storage.wiredTiger.engineConfig.cacheSizeGB或–wiredTigerCacheSizeGB进行设置。通过监控db.serverStatus()命令查看并评估。
- 写操作负载高的应用采用SSD：SATA接口的SSD提供强大的顺序写性能，能提高MongoDB数据和jounal日志文件的写入速度；同时最为强大的是随机读取性能，这点符合MongoDB的数据访问vi 模式。部署采用RAID-10，比RAID-5/RAID-6减少很多限制，并提供更高性能。
- 存储引擎配置数据压缩和I/O-intensive worloads:WiredTiger存储引擎支持本地压缩，能够更好的使用多核新处理器资源。

- 为每个MongoDB服务提供专用服务器：在单一服务器上安装多个MongoDB实例，系统会计算每个实例合适的缓存，并为每个实例分配默认的cache_size。通过storage.wiredTiger.engineConfig.cacheSizeGB参数调整。
- 使用多路查询路由：在不同服务器上使用多个mongos，最好将mongos部署在应用服务器上。
- 利用多核心：WiredTiger存储引擎是多线程且可以充分利用CPU多核心。MMAPv1存储引擎因为其并发模型，所以并不要求更多的CPU核心。
- 开启NTP时间同步服务
- 禁用NUMA：MongoDB运行在NUMA系统上会导致操作问题，禁用NUMA需要配置mermory interleave policy

编写脚本/etc/init.d/pro-startMongo

- disable NUMA

sysctl -w vm.zone_reclaim_mode=0
启动时必须指定numactl方式
numactl –interleave=all
禁用THP：Transparent Huge Pages（THP）是Linux内存管理策略会占用大量的内存

- 编写脚本/etc/init.d/pro-startMongoDB，加入了前面禁用NUMA部分

```
# disable transparent hugespages
case $1 in
  start)
    if [ -d /sys/kernel/mm/transparent_hugepage ]; then
```

```
    thp_path=/sys/kernel/mm/transparent_hugepage
  elif [ -d /sys/kernel/mm/redhat_transparent_hugepage ]; then
    thp_path=/sys/kernel/mm/redhat_transparent_hugepage
  else
    return 0
  fi

  echo 'never' > ${thp_path}/enabled
  echo 'never' > ${thp_path}/defrag

  re='^[0-1]+$'
  if [[ $(cat ${thp_path}/khugepaged/defrag) =~ $re ]]
  then
    # RHEL 7
    echo 0  > ${thp_path}/khugepaged/defrag
  else
    # RHEL 6
    echo 'no' > ${thp_path}/khugepaged/defrag
  fi

  unset re
  unset thp_path

      # disable NUMA
      sysctl -w vm.zone_reclaim_mode=0
  ;;
esac
```

更改脚本权限并加入自启动

sudo chmod 755 /etc/init.d/pro-startMongo

sudo chkconfig –add pro-startMongo

通过以下测试是否关闭了THP
cat /sys/kernel/mm/transparent_hugepage/enabled
cat /sys/kernel/mm/transparent_hugepage/defrag
如下显示表示成功
always madvise [never]

- 设置ulimit：默认的ulimit设置太小，以下是推荐设置

-f (file size): unlimited
-t (cpu time): unlimited
-v (virtual memory): unlimited
-n (open files): 64000
-m (memory size): unlimited
-u (processes/threads): 64000
我是新建/etc/security/limits.d/99-mongodb-nproc.conf文件，加入如下配置：
mongod soft fsize unlimited
mongod hard fsize unlimited
mongod soft cpu unlimited
mongod hard cpu unlimited
mongod soft as unlimited
mongod hard as unlimited
mongod soft nofile 64000
mongod hard nofile 64000
mongod soft nproc 64000
mongod hard nproc 64000

应用、模式设计

- 模式设计：关联模式和嵌套模式，见下图，根据应用设计最合理的模式。

- 文档的key命名尽量短，比如ParkingId可以改为pid。超过16MB，大文档尽量使用GridFS
- 在集群环境下，避免scatter-gather查询。其实这依赖于sharding的方式和sharding key的选择，要尽量满足大部分的业务需求。
- 读写分离设计：在读延迟允许范围内，进行读写分离是个不错的选择，可以大大降低主节点压力。
- 类似关系型数据库，一些优化建议：只查询和更新需要的字段，减少带宽；避免使用否定条件查询；合理使用覆盖索引并考虑最左前缀匹配原则；

硬盘I/O

- 预读设置：使用blockdev –setra 命令设置预读，单位是512B（扇区大小），不少于32kb，也不要设置太大，浪费I/O，设置大的预读有助于顺序读，具体的配置根据业务需求设定。
- 使用EXT4或者XFS文件系统。不要使用EXT3哦。
- 禁止Access Time设置。操作系统会维护文件最后的访问时间metadata，对于数据库意味着每次文件系统每访问一个页就会提交一个写操作，这将降低整个数据库的性能。

编辑/etc/fstab文件，指定noatime和nodiratime
/dev/sdb /data ext4 defaults,noatime,nodiratime 1 2

- 禁用Huges Pages虚拟内存页，MongoDB使用普通虚拟内存页性能更好，这个前面提到过。
- 使用RAID10，性能高于RAID-5或RAID-6
- swap设置：设置充足的swap空间防止OOM程序杀掉*mongod进程，对于Linux下的MMAPv1存储引擎，不会使用

swap空间，可以少量设置swap空间；Windows下的MMAPv1存储引擎需要设置swap空间；WiredTiger存储引擎，需要设置较大的swap空间。使用WiredTiger测试过程中，发现对于大查询，没有足够的swap空间甚至会报错。在/etc/sysctl.conf设置vm.swappiness为0，尽量使用内存。swap分区尽可能大些，以下是一些建议。测试过程中是128G内存，我设置交换分区内存大小也是128G。

- 日志和数据文件分开存储，日志存储在普通SAS盘上，数据存储在SSD上。数据库的读一般都是随机读，SSD在随机读性能上表现非常好；日志文件主要是顺序写，速度本身较高，而SSD在顺序读写上的表现非常糟糕，而且比普通SATA性能还要低。Mongodb的日志有journal和syslog，journal日志只能放在dbPath下面的journal目录中，而系统日志可以指定日志路径。journal可以挂载个SATA硬盘设备。这样做到日志和数据分离。
- 多路径设置：设置storage.wiredTiger.engineConfig.directoryForIndexes使索引和数据分离，设置该参数为true后，会在dbPath下面建立一个index目录，在index目录下挂载一个硬盘设备，使索引和数据做到分离。设置directoryPerDB每个数据库不同目录，对于每个目录最好挂载到不同的硬盘设备上。
- 压缩：WiredTiger提供snappy和zlib两种数据压缩方式。snappy提供压缩比低但是性能损耗低，zlib压缩比高但性能损耗较高，通过storage.wiredTiger.collectionConfig.blockCompressor参数设置