Manual

环境配置

需要安装以下环境：

- Docker
- Java 8
- Maven
- mongosh
- MongoDB Database Tools
- Mongo Compass

数据准备：将 db-generation 解压缩放置到 ${PROJECT}/（指代项目目录）下，重命名为 data 目录

部署方法

脚本介绍

- /data-generate 目录
  - 该脚本会使用 /data 目录下的数据，并将生成的新数据一并放置在 /data 下
- /script 目录
  - setup.sh、setup.bat：初始化并启动容器，包括数据加载等，用于第一次启动
  - shutdown.sh、shutdown.bat：停止容器的运行，并删除容器
  - start.sh、start.bat：开始容器的运行（第一次启动请使用 setup.sh）
  - stop.sh、stop.bat：停止容器的运行

部署流程

数据生成

使用脚本：/data-generate/genTable_mongoDB10G.py

```
# 进入到相应目录，然后运行 genTable_mongoDB10G.py
python3 genTable_mongoDB10G.py
```

启动服务

首次启动：使用脚本 /script/setup.sh

```
# 进入到 /script 目录，然后运行 setup.sh
./setup.sh
```

setup.sh 脚本说明

```
#!/bin/bash
```

```
# hdfs
docker-compose -f ../hdfs/docker-compose.yml up -d
sleep 10
# 创建目录，将 /data/articles/ 上传至 hdfs
docker exec -it namenode hadoop fs -mkdir /articles/
docker exec -it namenode hadoop fs -put /data/articles/ /articles/


# redis
docker pull redis:7.0.0
docker run -itd --name redis -p 6379:6379 redis
docker network connect hdfs_hadoopnet redis --ip 172.21.0.6


# mongodb
# config servers
docker-compose -f ../mongodb/docker-compose-configsvr.yml up -d
mongosh --host localhost --port 40011 ../mongodb/setup-configsvr.js
# shard servers
docker-compose -f ../mongodb/docker-compose-shardsvr.yml up -d
mongosh --host localhost --port 40021 ../mongodb/setup-shardsvr-rs1.js
mongosh --host localhost --port 40031 ../mongodb/setup-shardsvr-rs2.js
# mongos
docker-compose -f ../mongodb/docker-compose-mongos.yml up -d
sleep 5
mongosh --host localhost --port 40002 ../mongodb/setup-mongos.js
docker network connect hdfs_hadoopnet mongos --ip 172.21.0.7

# load data
# 使用 mongoimport 直接导入生成的数据
mongoimport --host localhost --port 40002 -d ddbms -c user --file ../data/user.dat
mongoimport --host localhost --port 40002 -d ddbms -c article --file
../data/article.dat
mongoimport --host localhost --port 40002 -d ddbms -c read --file ../data/read.dat
# 使用 js 脚本生成 be-read 和 rank 表
mongosh --host localhost --port 40002 ../mongodb/generate-beread.js
mongosh --host localhost --port 40002 ../mongodb/generate-rank.js
```

环境查看说明

Mongo

首先在 Docker Desktop 中 进入 mongos 容器：

在 Terminal 中 运行如下命令可以查看导入的数据

```
# 进入 mongodb
mongo

# 查看 db
show dbs

# 使用 ddbms 并查询 user 表
use ddbms
db.user.find()
```

查看数据分片

```
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("6383754ed5180dab2a303515")
  }
  shards:
        {  "_id" : "shardsvr_rs1",  "host" : "shardsvr_rs1/shardsvr1:27018,shardsvr2:27018",  "state" : 1,  "tags" : [ "DBMS1", "DBMS1-2" ] }
        {  "_id" : "shardsvr_rs2",  "host" : "shardsvr_rs2/shardsvr3:27018,shardsvr4:27018",  "state" : 1,  "tags" : [ "DBMS2", "DBMS1-2" ] }
  active mongoses:
        "4.4.14" : 1
  autosplit:
        Currently enabled: yes
  balancer:
        Currently enabled:  yes
        Currently running:  no
        Failed balancer rounds in last 5 attempts:  0
        Migration Results for the last 24 hours:
                No recent migrations
  databases:
        {  "_id" : "config",  "primary" : "config",  "partitioned" : true }
        {  "_id" : "ddbms",  "primary" : "shardsvr_rs1",  "partitioned" : true,  "version" : {  "uuid" : UUID("51ad99c4-def4-41d0-a11d-8e6eb6ed9e76"),  "lastMod" : 1 } }
                ddbms.article
                        shard key: { "category" : 1, "aid" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                shardsvr_rs1    3
                                shardsvr_rs2    2
                        { "category" : { "$minKey" : 1 }, "aid" : { "$minKey" : 1 } } -->> { "category" : "science", "aid" : { "$minKey" : 1 } } on : shardsvr_rs1 Timestamp(1, 0)
                        { "category" : "science", "aid" : { "$minKey" : 1 } } -->> { "category" : "science", "aid" : { "$maxKey" : 1 } } on : shardsvr_rs1 Timestamp(1, 1)
                        { "category" : "science", "aid" : { "$maxKey" : 1 } } -->> { "category" : "technology", "aid" : { "$minKey" : 1 } } on : shardsvr_rs2 Timestamp(1, 2)
                        { "category" : "technology", "aid" : { "$minKey" : 1 } } -->> { "category" : "technology", "aid" : { "$maxKey" : 1 } } on : shardsvr_rs2 Timestamp(1, 3)
                        { "category" : "technology", "aid" : { "$maxKey" : 1 } } -->> { "category" : { "$maxKey" : 1 }, "aid" : { "$maxKey" : 1 } } on : shardsvr_rs1 Timestamp(1, 4)
                         tag: DBMS1-2  { "category" : "science", "aid" : { "$minKey" : 1 } } -->> { "category" : "science", "aid" : { "$maxKey" : 1 } }
                         tag: DBMS2  { "category" : "technology", "aid" : { "$minKey" : 1 } } -->> { "category" : "technology", "aid" : { "$maxKey" : 1 } }
                ddbms.beread
                        shard key: { "category" : 1, "_id" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                shardsvr_rs1    3
                                shardsvr_rs2    2
                        { "category" : { "$minKey" : 1 }, "_id" : { "$minKey" : 1 } } -->> { "category" : "science", "_id" : { "$minKey" : 1 } } on : shardsvr_rs1 Timestamp(1, 0)
                        { "category" : "science", "_id" : { "$minKey" : 1 } } -->> { "category" : "science", "_id" : { "$maxKey" : 1 } } on : shardsvr_rs1 Timestamp(1, 1)
                        { "category" : "science", "_id" : { "$maxKey" : 1 } } -->> { "category" : "technology", "_id" : { "$minKey" : 1 } } on : shardsvr_rs2 Timestamp(1, 2)
                        { "category" : "technology", "_id" : { "$minKey" : 1 } } -->> { "category" : "technology", "_id" : { "$maxKey" : 1 } } on : shardsvr_rs2 Timestamp(1, 3)
                        { "category" : "technology", "_id" : { "$maxKey" : 1 } } -->> { "category" : { "$maxKey" : 1 }, "_id" : { "$maxKey" : 1 } } on : shardsvr_rs1 Timestamp(1, 4)
                         tag: DBMS1-2  { "category" : "science", "_id" : { "$minKey" : 1 } } -->> { "category" : "science", "_id" : { "$maxKey" : 1 } }
                         tag: DBMS2  { "category" : "technology", "_id" : { "$minKey" : 1 } } -->> { "category" : "technology", "_id" : { "$maxKey" : 1 } }
                ddbms.rank
                        shard key: { "temporal_granularity" : 1, "_id" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                shardsvr_rs1    3
                                shardsvr_rs2    4
                        { "temporal_granularity" : { "$minKey" : 1 }, "_id" : { "$minKey" : 1 } } -->> { "temporal_granularity" : "daily", "_id" : { "$minKey" : 1 } } on : shardsvr_rs1 Timestamp(1, 0)
                        { "temporal_granularity" : "daily", "_id" : { "$minKey" : 1 } } -->> { "temporal_granularity" : "daily", "_id" : { "$maxKey" : 1 } } on : shardsvr_rs1 Timestamp(1, 1)
                        { "temporal_granularity" : "daily", "_id" : { "$maxKey" : 1 } } -->> { "temporal_granularity" : "monthly", "_id" : { "$minKey" : 1 } } on : shardsvr_rs2 Timestamp(1, 2)
                        { "temporal_granularity" : "monthly", "_id" : { "$minKey" : 1 } } -->> { "temporal_granularity" : "monthly", "_id" : { "$maxKey" : 1 } } on : shardsvr_rs2 Timestamp(1, 3)
```

Hadoop

访问 localhost:9870 可以看到数据成功导入