DocumentDB vs Mongo 测试 report

Owner	Yannan
Created time	@November 30, 2022 11:48 AM
Reviewed	Not started

Install MongoDB on Amazon EC2

```
Sample数据
{
  "_id": {
    "$oid": "6384c60b0adc6e57cede87c6"
  },
  "journeyId": 1,
  "upmid": "upmid_0",
  "mobile": "180000000000",
  "email": "180000000000@nike.com",
  "nuid": "nuid_0",
  "smsConsent": false,
  "emailConsent": false,
  "nextStepId": 1,
  "traits": [
    {
      "tag": "sexual",
      "value": "male"
    },
    {
      "tag": "city",
      "value": "ShangHai"
    },
    {
      "tag": "device",
      "value": "IOS"
    }
  ]
}

{
  "_id": {
    "$oid": "6384c60b0adc6e57cede87c8"
  },
  "journeyId": 1,
  "upmid": "upmid_2",
  "mobile": "180000000002",
  "email": "180000000002@nike.com",
  "nuid": "nuid_2",
  "smsConsent": false,
  "emailConsent": false,
  "nextStepId": 1,
  "traits": [
    {
      "tag": "sexual",
      "value": "male"
    },
    {
      "tag": "city",
      "value": "Beijing"
    },
    {
      "tag": "device",
      "value": "IOS"
    }
  ]
}
```

更新语句

```
db.audience_2.updateMany({"$and":[
{"traits.tag": "sexual", "traits.value": "male"},
{"traits.tag": "city", "traits.value": "ShangHai"},
{"traits.tag": "device", "traits.value": "Android"}]},
{$set:{"nextStepId":"step32"}});
```

Mongo Client

```
// SSH到mongo client Using VSCode
edit this file: `/Users/heyannan/.ssh/config`

```conf
Host MongoClient
  Hostname ec2-43-192-28-184.cn-northwest-1.compute.amazonaws.com.cn
  IdentityFile /Users/heyannan/Documents/key/yannan-ec2.pem
  User ec2-user
```


// Create virtual environment
python3 -m venv env

// Use the environment
source env/bin/activate

// Install dependencies
pip install pymongo

// (Optional) Save dependencies to requirements.txt
pip freeze > requirements.txt

// (Optional) Install from requirements.txt
pip install -r requirements.txt

// import mock data
python data.py

// connect to mongo primary node
mongo --host ec2-69-230-248-223.cn-northwest-1.compute.amazonaws.com.cn:27017

// Show all dbs to check the data ingestion progress
show dbs

admin   0.000GB
config   0.000GB
db2   0.487GB
local   0.437GB
test   0.000GB
test-connection 0.000GB

// Use db2
use db2
switched to db db2

// Get current db name
db.getCollectionNames()

// 测试mongo connection before insert the real data
>>> import pymongo
>>> client= pymongo.MongoClient("mongodb://ec2-69-230-248-223.cn-northwest-1.compute.amazonaws.com.cn:27017")
>>> db = client['test-connection']
>>> collection = db['test-collection']
>>> post = {"author": "Mike"}
>>> posts = db.posts
>>> posts.insert_one(post)
<pymongo.results.InsertOneResult object at 0x7f8c82bf0910>
```

```
>>> db.posts.find()
<pymongo.cursor.Cursor object at 0x7f8c8218b250>
>>> db.posts.find_one()
{'_id': ObjectId('6388216fe5e3338a65de338e'), 'author': 'Mike'}
```

DocumentDB设置

```
wget https://s3.cn-north-1.amazonaws.com.cn/rds-downloads/rds-combined-ca-cn-bundle.pem
mongo --ssl --sslAllowInvalidCertificates --host yannantest.cluster-csu8g7al8jjy.docdb.cn-northwest-1.amazonaws.com.cn:27017 --sslCAFile rd

python data-documentdb.py

myclient = pymongo.MongoClient(
    "mongodb://heyannan:nannan740740@yannantest.csu8g7al8jjy.docdb.cn-northwest-1.amazonaws.com.cn:27017/?tls=true&tlsCAFile=rds-combined-c
)
```

测试环境

数据量大小5000w items - 25.337GB

MongoDB：宁夏region，3个AZ（cn-northwest-1a，cn-northwest-1b，cn-northwest-1c），每个AZ两个MongoDB node（r5.xlarge）

DocumentDB： 单node（r5.xlarge）

MongoDB 测试1

writeConcern - Mongo **4.0版本 default是1**

```
db.col_nike_test01.updateMany(
...   {
...     $and: [
...       { "traits.tag": "sexual", "traits.value": "male" },
...       { "traits.tag": "city", "traits.value": "ShangHai" },
...       { "traits.tag": "device", "traits.value": "Android" },
...     ],
...   },
...   { $set: { nextStepId: "step32" } }
... );
;
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
rs-test:PRIMARY> print("执行耗时：", new Date().getTime() - time);
执行耗时： 167941
```

modifiedCount=0的原因是5000w的数据里没有”ShangHai“只有”Shanghai“。

在MongoDB集群上只跑了这一个测试，因为mongo server 因为一些问题被中断了

DocumentDB 测试1

```
db.col_nike_test01.updateMany(
...   {
...     $and: [
...       { "traits.tag": "sexual", "traits.value": "male" },
...       { "traits.tag": "city", "traits.value": "ShangHai" },
...       { "traits.tag": "device", "traits.value": "Android" },
...     ],
...   },
...   { $set: { nextStepId: "step32" } }
... );
;
```

```
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
执行耗时：241539
```

可以看出，跑同样的query，documentDB（writeConcern: 4）花了4分钟，mongoDB集群（writeConcern: 1）花了接近3分钟。

DocumentDB 测试2

```
db.col_nike_test01.updateMany(
  {
    $and: [
      { "traits.tag": "sexual", "traits.value": "male" },
      { "traits.tag": "city", "traits.value": "Shanghai" },
      { "traits.tag": "device", "traits.value": "Android" },
    ],
  },
  { $set: { nextStepId: "step32" } }
);

{
        "acknowledged" : true,
        "matchedCount" : 2084905,
        "modifiedCount" : 2084905
}
rs0:PRIMARY> print("执行耗时：", new Date().getTime() - time);
执行耗时： 889584
```

DocumentDB在5000w数据里更新208w数据花了近15分钟。

Nike自己的测试

测试环境：

MongoDB：Nike本机单节点

DocumentDB：db.r6g.16xlarge 单节点？

测试1

```
var time = new Date().getTime();
db.audience_2.find({"$and":[
{"traits.tag": "sexual", "traits.value": "female"},
{"traits.tag": "city", "traits.value": "Guangzhou"},
{"traits.tag": "device", "traits.value": "IOS"}]}).limit(1);
print("执行耗时：", new Date().getTime() - time);

'执行耗时：'
MongoDB: 3ms
DocumentDB: 19604ms
```

测试2

```
documentDB在没有index
db.audience_3.ensureIndex({"traits.tag": 1, "traits.value": 1})的情况下
做更新5000W中的310W大概花了438367ms

***Update***
var time = new Date().getTime();
db.audience_3.updateMany({"$and":[
{"traits.tag": "sexual", "traits.value": "male"},
{"traits.tag": "city", "traits.value": "Beijing"},
{"traits.tag": "device", "traits.value": "Android"}]},
{$set:{"nextStepId":"step41"}});
print("执行耗时：", new Date().getTime() - time);
'执行耗时：' // ms
438367
```

```
然后再查询大概144ms
***Find***
var time = new Date().getTime();
db.audience_3.find({"$and":[
{"traits.tag": "sexual", "traits.value": "male"},
{"traits.tag": "city", "traits.value": "Beijing"},
{"traits.tag": "device", "traits.value": "Android"}]});
print("执行耗时：", new Date().getTime() - time);
'执行耗时：'
144
```