

项目七

使用 MongoDB 实现数据库服务

任务四 监控 MongoDB



扫码看视频

【任务介绍】

监控是数据库管理的重要内容之一，通过监控可以实时了解数据库的运行状态，及时发现系统异常，有效地避免数据库故障，保障业务顺利可用。本任务通过云服务、实用工具和内置命令实现 MongoDB 的监控。

本任务在任务三的基础上进行（虚拟机使用的服务器是 Project-07-Task-02）。

【任务目标】

- （1）实现使用 `mongostat` 监控 MongoDB 的运行状态。
- （2）实现使用 `mongotop` 获取 MongoDB 的运行性能。
- （3）实现使用 MongoDB Web Console 监控 MongoDB。
- （4）实现使用 MongoDB Cloud Manager 监控 MongoDB。

【操作步骤】

步骤 1：通过 MongoDB Database Tools 监控 MongoDB。

1. 通过 RPM 包安装 MongoDB Database Tools

MongoDB Database Tools 用于管理和监控 MongoDB 的命令行实用程序的集合。

通过 `wget` 工具下载 MongoDB Database Tools 的 RPM 包，并进行安装。本任务的使用版本为 100.8.0。

操作命令：

```

1. # 通过 wget 下载 mongodb-database-tools-100.8.0
2. [root@Project-07-Task-02 ~]# wget https://fastdl.mongodb.org/tools/db/mongodb-database-tools-rhel80-x86_64-100.8.0.rpm
3. --2023-09-05 06:43:13-- https://fastdl.mongodb.org/tools/db/mongodb-database-tools-rhel80-x86_64-100.8.0.rpm
4. Resolving fastdl.mongodb.org (fastdl.mongodb.org)... 54.192.18.79, 54.192.18.31, 54.192.18.99, ...
5. Connecting to fastdl.mongodb.org (fastdl.mongodb.org)[54.192.18.79]:443... connected.
6. HTTP request sent, awaiting response... 200 OK
7. Length: 53745748 (51M) [binary/octet-stream]
8. Saving to: 'mongodb-database-tools-rhel80-x86_64-100.8.0.rpm'
9. mongodb-database-tools-rhel80-x86_64-100.8.0. 100%[=====
=>] 51.26M 8.98MB/s in 5.1s
10. 2023-09-05 06:43:30 (10.0 MB/s) - 'mongodb-database-tools-rhel80-x86_64-100.8.0.rpm' saved [53745748/53745748]
11. # 通过 RPM 命令进行安装
12. [root@Project-07-Task-02 ~]# rpm -ivh mongodb-database-tools-rhel80-x86_64-100.8.0.rpm
13. warning: mongodb-database-tools-rhel80-x86_64-100.8.0.rpm: Header V3 RSA/SHA256 Signature, key ID 90cfb1f5: NOKEY
14. Verifying... ##### [100%]
15. Preparing... ##### [100%]
16. Updating / installing...
17. 1:mongodb-database-tools-100.8.0-1 ##### [100%]

```

操作命令+配置文件+脚本程序+结束

MongoDB Database Tools 集合包含了多种类型的工具，具体见表 7-4-1。

表 7-4-1 MongoDB Database Tools 工具列表

名称	类型	描述
mongodump	Binary Import / Export	以二进制格式导出数据
mongorestore	Binary Import / Export	将 mongodump 导出的数据进行转储还原
bsondump	Binary Import / Export	将 BSON 格式转换为 JSON
mongoimport	Data Import / Export	导入数据，支持 JSON、CSV 或 TSV 格式
mongoexport	Data Import / Export	导出数据，支持 JSON、CSV 格式
mongostat	Diagnostic Tools	查看单实例或集群的状态
mongotop	Diagnostic Tools	查看实例的读写耗时
mongofiles	GridFS Tools	从其他成员来看，该成员不可访问

2. mongostat

mongostat 工具根据数据库的操作类型（例如插入、查询、更新、删除等）捕获并返回统计报告服务器上的负载分布。

命令详解: mongostat

【语法】

mongostat 【options】

【options】

--host=<hostname>[:port]:	数据库服务器主机地址, 可使用半角逗号隔开指定多个数据库服务器主端口号
--port=<port>:	数据库连接用户名
--username=<username>, -u=<username>:	数据库连接密码
--password=<password>, -p=<password>:	认证数据库
--authenticationDatabase=<dbname>:	使用连接字符串格式连接, 但不可与认证字段同时使用
--uri=<connectionString>:	格式化输出日期等字段
--humanReadable=<boolean>:	指定输出字段和字段名称, 可使用半角逗号隔开指定多个以 JSON 格式输出信息
-O=<field list>:	执行 mongostat 命令执行的间隔时间
--json:	
<sleeptime>:	

操作命令+配置文件+脚本程序+结束

操作命令:

1. # 使用 mongostat 监控单个 mongod 示例
2. [root@Project-07-Task-02 ~]# mongostat --username=admin --password=openeucler@mongodb#123 --authenticationDatabase=admin -o=insert,query,update,delete,net_in,net_out,conn,repl
3. insert query update delete net_in net_out conn repl
4. *0 *0 *0 *0 1.68k 66.5k 17 SLV
5. *0 *0 *0 *0 339b 62.8k 17 SLV
6. *0 *0 *0 *0 2.29k 64.9k 17 SLV

操作命令+配置文件+脚本程序+结束

mongostat 命令结果的字段说明如下。

- insert: 检测磁盘设备名称
- query: 每秒的执行查询操作数
- update: 每秒的执行更新操作数
- delete: 每秒的执行删除操作数
- getmore: 每秒获取更多(即游标批处理)操作数
- command: 每秒的执行操作数
- vsize: 进程使用的虚拟内存量(以 MB 为单位)
- res: 进程使用的驻留内存量(以 MB 为单位)
- locked: 全局写锁定中的时间百分比
- qr: 等待读数据的客户端数
- qw: 等待写数据的客户端数
- ar: 执行读数据的客户端数
- aw: 执行写数据的客户端数
- net_in: MongoDB 实例接收的网络流量(以字节为单位)



小贴士

- net_out: MongoDB 实例发送的网络流量（以字节为单位）
- conn: 数据库连接数
- set: 副本集的名称
- repl: 副本集成员状态

3. mongotop

mongotop 提供一种跟踪 MongoDB 实例 mongod 在读写数据上的时间消耗，提供每个集合级别的统计信息。默认情况下，mongotop 每秒返回一次值。

命令详解：mongotop

【语法】

mongotop 【options】

【options】

--host=<hostname>:<port>:	数据库服务器主机地址，可使用半角逗号隔开指定多个数据库服务器主端口号
--port=<port>:	数据库服务器主端口号
--username=<username>,-u=<username>:	数据库连接用户名
--password=<password>,-p=<password>:	数据库连接密码
--authenticationDatabase=<dbname>:	认证数据库
--uri=<connectionString>:	使用连接字符串格式连接，但不可与认证字段同时使用
--humanReadable=<boolean>:	格式化输出日期等字段
-O=<field list>:	指定输出字段和字段名称，可使用半角逗号隔开指定多个
--json:	以 JSON 格式输出信息
<sleeptime>:	执行 mongostat 命令执行的间隔时间

操作命令+配置文件+脚本程序+结束

操作命令：

1. # 使用 mongotop 监控单个 mongod 示例
2. [root@Project-07-Task-02 ~]# mongotop --username=admin --password=openeuler@mongodb#123 --authenticationDatabase=admin
3. 2023-09-05T09:06:56.410+0800 connected to: mongodb://localhost/
- 4.
5. ns total read write 2023-09-05T09:06:57+08:00
6. admin.system.keys 0ms 0ms 0ms
7. admin.system.users 0ms 0ms 0ms
8. admin.system.version 0ms 0ms 0ms
9. admin.test_collection 0ms 0ms 0ms
10. config.collections 0ms 0ms 0ms
11. config.external_validation_keys 0ms 0ms 0ms
12. config.image_collection 0ms 0ms 0ms
13. config.settings 0ms 0ms 0ms
14. config.system.sessions 0ms 0ms 0ms
15. config.tenantMigrationDonors 0ms 0ms 0ms

操作命令+配置文件+脚本程序+结束



小贴士

mongotop 命令结果的字段说明如下。

- ns: 数据库命名空间
- db: 数据库的名称
- total: 在此命名空间上操作所消耗的总时长
- read: 在此名称空间上执行读操作所消耗的时间
- write: 在此名称空间上执行写操作所消耗的时间
- <timestamp>: 检测的有效时间

步骤 2: 使用 MongoDB Database Methods 监控 MongoDB。

Database Methods (数据库方法) 是 mongosh Methods (Mongo Shell 方法) 内置方法中的一类。数据库方法用于管理和监控 MongoDB 实例或集群的方法集。

(1) mongosh Methods 包含的方法类型说明如下。

- Collection: 集合方法集
- Cursor: 游标方法集
- Database: 数据库方法集
- Query Plan Cache: 查询计划缓存方法集
- Bulk Write Operation: 批量写操作方法集
- User Management: 用户管理方法集
- Role Management: 角色管理方法集
- Replication: 复制方法集
- Sharding: 分片方法集
- Constructors: 构造函数方法集
- Connection: 连接方法集
- Client-Side Field Level Encryption: 客户端字段级加密方法集
- Atlas Search Index Methods: 图谱搜索索引方法集

(2) Database Methods 包含的方法类型说明如下

- db.cloneDatabase(): 从指定主机上克隆数据库
- db.currentOp(): 显示当前正在进行的操作
- db.commandHelp(): 返回数据库命令的帮助信息
- db.createCollection(): 创建一个聚集集合 (table)
- db.cloneCollection(): 在 MongoDB 库实例之间复制集合数据
- db.copyDatabase(): 从指定的机器上复制指定的数据库数据到某个数据库
- db.dropDatabase(): 删除当前使用的数据库
- db.fsyncLock(): 刷新写入磁盘并锁定该数据库, 以防止写入操作, 并协助备份操作



小贴士

- db.fsycUnlock(): 允许继续进行写入锁住的数据库（解锁）
- db.getLogComponents(): 返回日志消息详细级别
- db.getLastErrorObj(): 查看完整的错误结果
- db.getMongo(): 查看当前数据库的连接机器地址
- db.getCollection(): 得到指定名称的聚集集合（table）
- db.getName(): 查看当前使用的数据库
- db.getPrevError(): 返回包含自上次错误复位所有的错误状态文件
- db.getCollectionNames(): 得到当前数据库的所有聚集集合
- db.getCollectionInfos(): 返回当前数据库中的所有集合信息
- db.getLastError(): 返回上一次错误，如果没有错误则为空
- db.hostInfo(): 返回当前数据库主机系统的相关信息
- db.killOp(): 终止指定的操作
- db.listCommands(): 显示公共数据库的命令列表
- db.logout(): 注销登录
- db.printCollectionStats(): 显示当前数据库所有聚集索引的状态
- db.resetError(): 重置数据库
- getPrevError()和 getPrevError: 返回的错误信息
- db.repairDatabase(): 修复当前数据库
- db.runCommand(): 运行一个数据库命令
- db.serverStatus(): 返回当前数据库状态的概要
- db.setProfilingLevel(): 修改当前数据库的分析级别
- db.stats(): 显示当前数据库状态
- db.shutdownServer(): 关闭当前数据库运行实例或安全停止有关操作进程
- db.setLogLevel(): 设置一个单独的日志信息级别
- db.version(): 查看当前数据库版本

1. db.serverStatus()

db.serverStatus()返回数据库状态的概述信息，包括连接信息、选举信息、流量控制、数据库锁信息、网络流量、操作延迟情况、操作统计、内存使用情况等信息。

操作命令：

1. # 使用 mongotop 监控单个 mongod 示例
2. [root@Project-07-Task-02 ~]# mongosh
3. Current Mongosh Log ID: 64f6829fdd88c4c0bd042bf7
4. Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.9.1
5. Using MongoDB: 6.0.8
6. Using Mongosh: 1.9.1

```
7. For mongosh info see: https://docs.mongodb.com/mongodb-shell/
8. db-cluster-mongodb [direct: primary] test> use admin;
9. switched to db admin
10. db-cluster-mongodb [direct: primary] admin> db.auth("admin","openeuler@mongodb#123");
11. { ok: 1 }
12. db-cluster-mongodb [direct: primary] admin> db.serverStatus()
13. {
14.   host: 'Project-07-Task-02',
15.   version: '6.0.8',
16.   process: 'mongod',
17.   pid: Long("3433"),
18.   uptime: 19610,
19.   uptimeMillis: Long("19609809"),
20.   uptimeEstimate: Long("19609"),
21.   localTime: ISODate("2023-09-05T01:39:08.831Z"),
22.   activeIndexBuilds: {
23.     total: 0,
24.     phases: {
25.       scanCollection: 0,
26.       drainSideWritesTable: 0,
27.       drainSideWritesTablePreCommit: 0,
28.       waitForCommitQuorum: 0,
29.       drainSideWritesTableOnCommit: 0,
30.       processConstraintsViolatonTableOnCommit: 0,
31.       commit: 0
32.     }
33.   },
34.   asserts: {
35.     regular: 0,
36.     warning: 0,
37.     msg: 0,
38.     user: 579,
39.     tripwire: 0,
40.     rollovers: 0
41.   },
42.   batchedDeletes: { batches: 0, docs: 0, stagedSizeBytes: 0, timeMillis: Long("0") },
43.   catalogStats: {
44.     collections: 2,
45.     capped: 0,
46.     clustered: 0,
47.     timeseries: 0,
48.     views: 0,
49.     internalCollections: 21,
50.     internalViews: 1
51.   },
52.   connections: {
```

```

53.     current: 16,
54.     available: 51184,
55.     totalCreated: 149,
56.     active: 6,
57.     threaded: 16,
58.     exhaustIsMaster: 0,
59.     exhaustHello: 3,
60.     awaitingTopologyChanges: 4
61.   },
62. # 为了排版方便，以下输出信息中省略了部分信息
63.   ok: 1,
64.   '$clusterTime': {
65.     clusterTime: Timestamp({ t: 1693877940, i: 1 }),
66.     signature: {
67.       hash: Binary(Buffer.from("1a4c66a4b10060e0ed1fb67ab4e85fedf66ef474", "hex"), 0),
68.       keyId: Long("727501738782333381")
69.     }
70.   },
71.   operationTime: Timestamp({ t: 1693877940, i: 1 })
72. }

```

操作命令+配置文件+脚本程序+结束

2. db.stats()

db.stats()可用于查看单个数据库使用状态的统计信息。

操作命令：

```

1. # 使用 db.stats()查看信息
2. [root@Project-07-Task-02 ~]# mongosh
3. Current Mongosh Log ID: 65103882ec88e4d630ef2985
4. Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=
2000&appName=mongosh+1.9.1
5. Using MongoDB:      6.0.8
6. Using Mongosh:      1.9.1
7.
8. For mongosh info see: https://docs.mongodb.com/mongosh-shell/
9.
10. db-cluster-mongodb [direct: secondary] test> use admin;
11. switched to db admin
12. db-cluster-mongodb [direct: secondary] admin> db.auth("admin","openeuler@mongodb#123");
13. { ok: 1 }
14. db-cluster-mongodb [direct: secondary] admin> db.stats()
15. {
16.   db: 'admin',
17.   collections: 4,
18.   views: 0,
19.   objects: 9,

```

```

20.   avgObjSize: 293.77777777777777,
21.   dataSize: 2644,
22.   storageSize: 135168,
23.   indexes: 5,
24.   indexSize: 163840,
25.   totalSize: 299008,
26.   scaleFactor: 1,
27.   fsUsedSize: 4188913664,
28.   fsTotalSize: 17782554624,
29.   ok: 1,
30.   '$clusterTime': {
31.     clusterTime: Timestamp({ t: 1695559310, i: 1 }),
32.     signature: {
33.       hash: Binary(Buffer.from("173339425d89ae415445eb884e928129228e783e", "hex"), 0),
34.       keyId: Long("727501738782333381")
35.     }
36.   },
37.   operationTime: Timestamp({ t: 1695559310, i: 1 })
38. }
39. db-cluster-mongodb [direct: secondary] admin>

```

操作命令+配置文件+脚本程序+结束

db.stats()方法结果的字段说明如下。

- db: 数据库的名称
- collections: 数据库中的集合数
- views: 数据库中的视图数
- objects: 数据库中对象（即 document）数
- avgObjSize: 每个文档的平均大小（以字节为单位）
- dataSize: 数据库中保存的未压缩数据的总大小
- storageSize: 分配给数据库中用于文档存储的所有集合的磁盘空间总和
- freeStorageSize: 分配给数据库中用于文档存储的所有集合的可用空间的总和
- indexes: 数据库中所有集合的索引总数
- indexSize: 数据库中所有集合的索引总数
- indexFreeStorageSize: 分配给数据库中所有索引的磁盘空间总和，包括可用索引空间
- totalSize: storageSize 与 indexSize 的总和
- totalFreeStorageSize: freeStorageSize 与 indexFreeStorageSize 的总和
- fsUsedSize: MongoDB 所在的文件系统上使用的所有磁盘空间的总大小



小贴士

3. db.collection.collStats()

该方法返回指定集合的存储统计信息。

操作命令：

```

1. # 使用 db.collection.collStats()查看信息
2. [root@Project-07-Task-02 ~]# mongosh
3. Current Mongosh Log ID: 65103d4963d275001c750663
4. Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2
   000&appName=mongosh+1.9.1
5. Using MongoDB:      6.0.8
6. Using Mongosh:      1.9.1
7.
8. For mongosh info see: https://docs.mongodb.com/mongosh-shell/
9.
10. db-cluster-mongodb [direct: primary] test> use admin;
11. switched to db admin
12. db-cluster-mongodb [direct: primary] admin> db.auth("admin","openeuler@mongodb#123");
13. { ok: 1 }
14. db-cluster-mongodb [direct: primary] admin> db.test_collection.stats()
15. {
16.   ok: 1,
17.   capped: false,
18.   wiredTiger: {
19.     metadata: { formatVersion: 1 },
20.     creationString: 'access_pattern_hint=none,allocation_size=4KB,app_metadata=(formatVersion=1),assert=
   (commit_timestamp=none,durable_timestamp=none,read_timestamp=none,write_timestamp=off),block_allocat
   ion=best,block_compressor=snappy,cache_resident=false,checksum=on,colgroups=,collator=,columns=,diction
   ary=0,encryption=(keyid=,name=),exclusive=false,extractor=(format=btree,huffman_key=,huffman_value=,ign
   ore_in_memory_cache_size=false,immutable=false,import=(compare_timestamp=oldest_timestamp,enabled=fal
   se,file_metadata=,metadata_file=,repair=false),internal_item_max=0,internal_key_max=0,internal_key_trunca
   te=true,internal_page_max=4KB,key_format=q,key_gap=10,leaf_item_max=0,leaf_key_max=0,leaf_page_ma
   x=32KB,leaf_value_max=64MB,log=(enabled=false),lsm=(auto_throttle=true,bloom=true,bloom_bit_count=1
   6,bloom_config=,bloom_hash_count=8,bloom_oldest=false,chunk_count_limit=0,chunk_max=5GB,chunk_siz
   e=10MB,merge_custom=(prefix=,start_generation=0,suffix=),merge_max=15,merge_min=0),memory_page_i
   mage_max=0,memory_page_max=10m,os_cache_dirty_max=0,os_cache_max=0,prefix_compression=false,pr
   efix_compression_min=4,readonly=false,source=,split_deepen_min_child=0,split_deepen_per_child=0,split_pc
   t=90,tiered_object=false,tiered_storage=(auth_token=,bucket=,bucket_prefix=,cache_directory=,local_retention
   =300,name=,object_target_size=0),type=file,value_format=u,verbose=[],write_timestamp_usage=none',
21.   type: 'file',
22.   uri: 'statistics:table:collection-0--6398009657706681439',
23. # 为了排版方便，以下输出信息中省略了部分信息
24. },
25. sharded: false,
26. size: 22,
27. count: 1,

```

```
28.   numOrphanDocs: 0,
29.   storageSize: 32768,
30.   totalIndexSize: 32768,
31.   totalSize: 65536,
32.   indexSizes: { _id_: 32768 },
33.   avgObjSize: 22,
34.   ns: 'admin.test_collection',
35.   nindexes: 1,
36.   scaleFactor: 1
37. }
```

操作命令+配置文件+脚本程序+结束

db.collection.collStats()方法的结果的字段如下。

db.collection.collStats()方法结果的字段说明如下。

- ns: 集合命名空间
- size: 内存中所有记录未压缩的存储总量
- timeseries: 集合时序信息
- count: 集合中的对象或文档数
- avgObjSize: 集合中对象或文档的平均存储量
- numOrphanDocs: 集合中独立文档数
- storageSize: 分配给此集合的存储总量
- freeStorageSize: 不适用于内存中的存储量
- nindexes: 集合上的索引数
- indexDetails: 索引的相关信息
- indexBuilds: 构建的索引
- totalIndexSize: 索引总量
- totalSize: storageSize 与 totalIndexSize 的总和
- indexSizes: 集合上每个索引的键和大小
- scaleFactor: 命令使用的比例值
- capped: 集合上限, 如果集合有上限, 则此字段将为 true
- max: 集合上限中可能存在的文档最大数
- maxSize: 集合上限中可能存在文档的最大存储总量
- wiredTiger: 存储引擎
- inMemory: 正在使用的内存量。



步骤 3: 使用 MongoDB Compass 监控 MongoDB 服务器。

单击“Performance”选项卡切换到监控指标选项, 在监控指标选项中可查看操作、数据读写、网络、内存、慢查询等实时情况, 如图 7-4-1 所示。

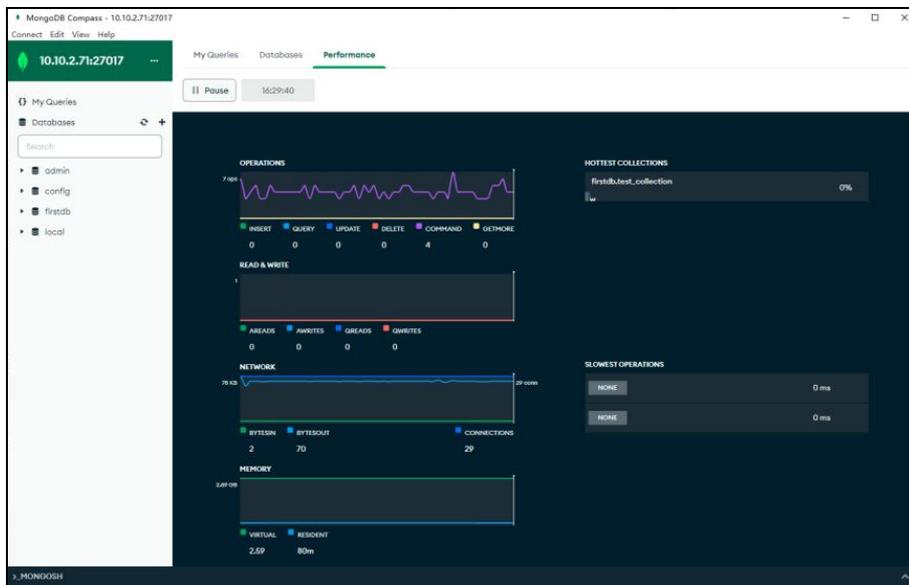


图 7-4-1 查看性能

MongoDB Compass 监控指标说明，见表 7-4-2。

表 7-4-2 MongoDB Compass 监控指标说明

图表/表格	描述
Operations	显示数据库处理的各类操作数量，与 mongostat 相关
Read & Write	显示活动读取数、排队读取数、活动数写入、排队写入
Network	显示数据库操作的网络性能
Memory	显示内存统计信息
Hottest Collections	显示报告的活动最多的集合。集合对应于图表中的给定时刻；即当在图表上移动时，相应的集合在表格中突出显示/显示
Slowest Operations	显示最慢的操作。集合对应于图表中的给定时刻；即当在图表上移动时，相应的集合在表格中突出显示/显示

任务五 使用 Percona Monitoring and Management 监控 MongoDB

【任务介绍】

Percona Monitoring and Management（简称“PMM”）是一款开源数据库监控和管理工具软件，支持 MongoDB、MySQL、MariaDB、PostgreSQL 等常用数据库



扫码看视频