2019 级

《大数据存储与管理》课程

实 验 报 告

姓　名　高世文

学　号　U201915040

班　号　物联网 1901

日　期　2022.04.18

目　　录

一、实验目的

1. 熟悉对象存储技术，代表性系统及其特性；
2. 实践对象存储系统，部署实验环境，进行初步测试；
3. 基于对象存储系统，分析性能问题，架设应用实践。

二、实验背景

本实验为对象存储入门实验。

实验第一项是进行环境搭建，除基本环境外还包括服务端、客户端等配置；

实验第二项是性能观测，选择 COSbench、S3 bench 或 benchmark 进行观测。

观测指标：吞吐率 Throughput、延迟 Latency，以及环境参数：对象尺寸 object size、并发性、服务器数量等。

三、实验环境

操作系统	Win10
服务端	Minio
客户端	Minio Client
评测工具	S3 Bench

四、实验内容

实验内容总体包括：

服务端 Minio 的安装；

客户端 Minio 的安装和创建 bucket 尝试；

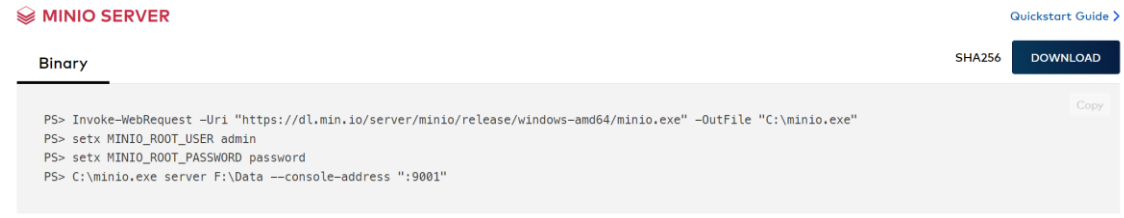
S3 Bench 的安装及运行；

观测 S3 Bench 运行数据，得出结论。

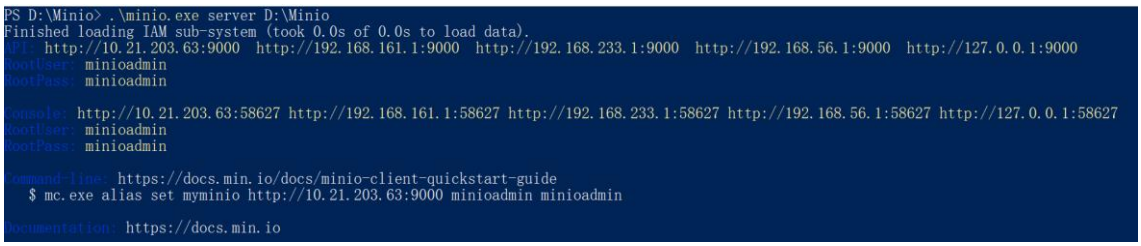
4.1 对象存储技术实践

（1）.Minio 的安装

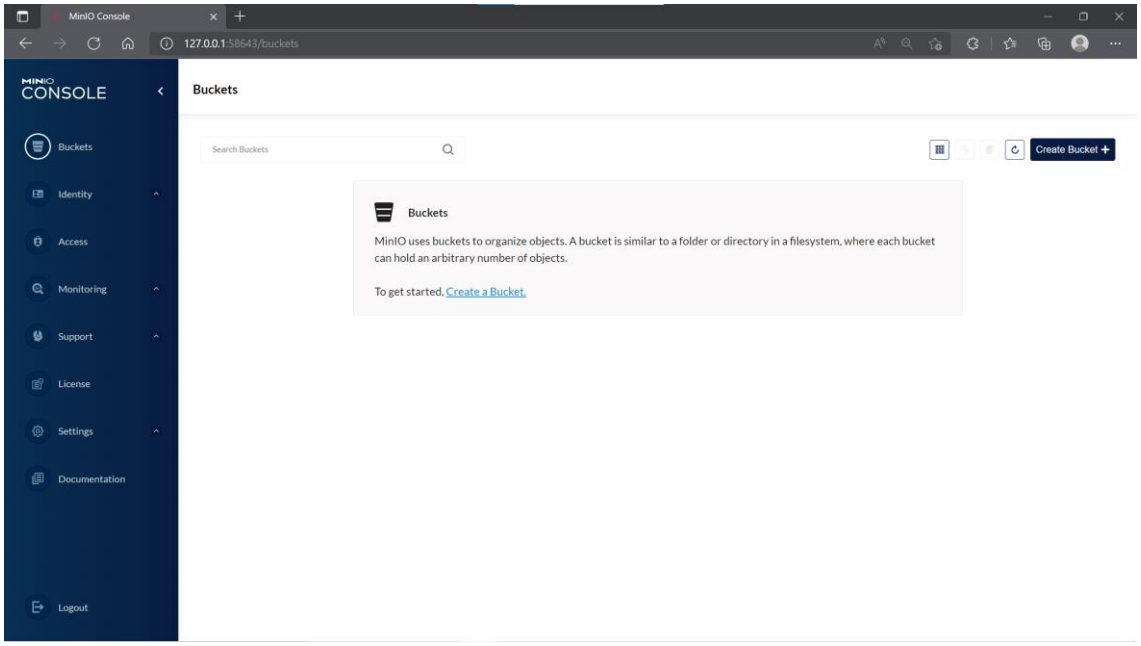
前往 [MinIO | Code and downloads to create high performance object storage](https://min.io/download?license=agpl&platform=windows) 下载，如下图：



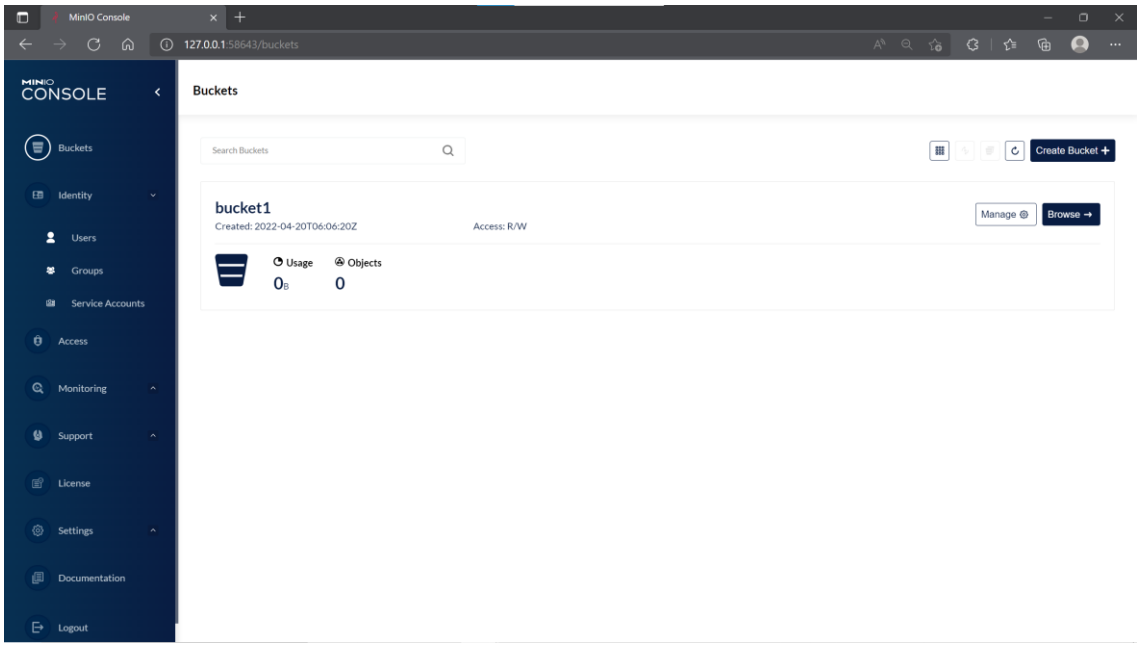
将 minio 放在路径 D:\Minio 下，Shell 窗口运行命令`.\minio.exe server D:\Minio` 如下图：



前往浏览器打开 http://127.0.0.1:9000，输入 Usr 和 Pass，可以看到启动成功：



点击 Create a Bucket，尝试创建一个桶，可以看到创建成功：



（2）Minio Client 安装

前往 MinIO | Code and downloads to create high performance object storage 下载 mc.exe

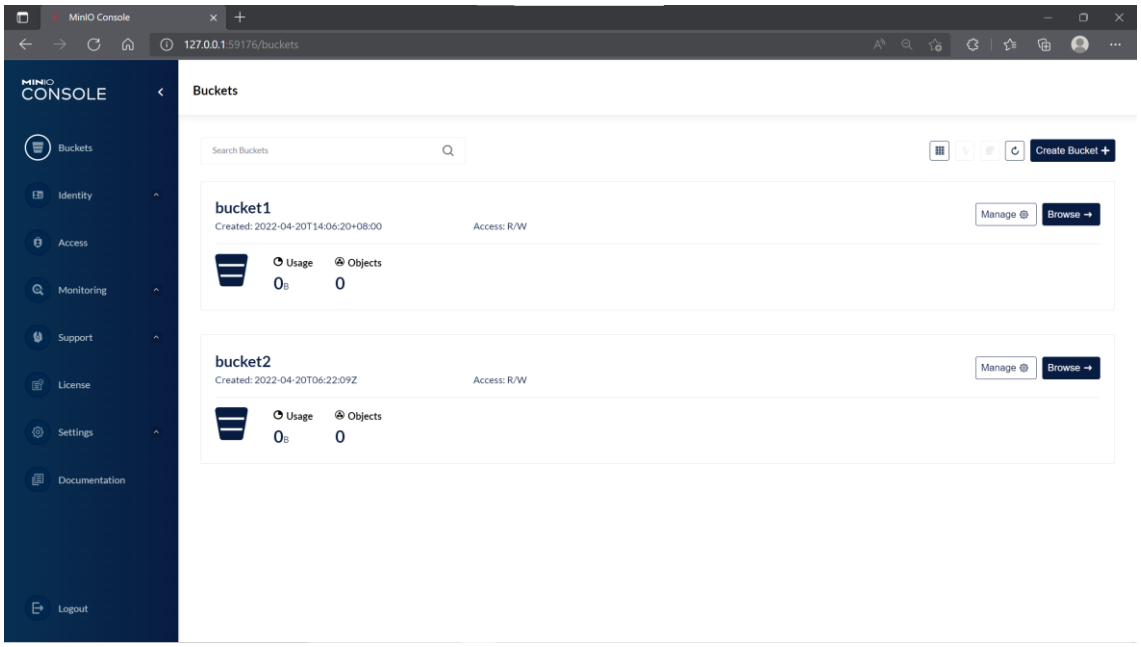
mc．exe 同样放在 Minio 文件夹下，尝试建立联系如下：

```
PS D:\Minio> .\mc.exe alias set minio/ http://127.0.0.1:9000 minioadmin minioadmin
Added `minio` successfully.
PS D:\Minio>
```

尝试创建新的 bucket：

```
PS D:\Minio> .\mc.exe mb minio/bucket2
Bucket created successfully `minio/bucket2`.
PS D:\Minio>
```

浏览器查看创建成功：



演示完毕后即可删除 bucket2

（3）S3 Bench 安装运行

前往文件分享 (weiyun.com)下载 s3bench.exe

前往 run-s3bench.cmd · Zhan/obs-tutorial - Gitee.com 下载运行脚本

run-s3bench.cmd 和 s3bench 置于同一级目录下，编辑 run-s3bench.cmd，修改用户名、密码、桶为自己创建的值：

```
@rem -accessKey       Access Key
@rem -accessSecret    Secret Key
@rem -bucket=loadgen   Bucket for holding all test objects.
@rem -endpoint=http://127.0.0.1:9000 Endpoint URL of object storage service being tested.
@rem -numClients=8    Simulate 8 clients running concurrently.
@rem -numSamples=256   Test with 256 objects.
@rem -objectNamePrefix=loadgen Name prefix of test objects.
@rem -objectSize=1024        Size of test objects.
@rem -verbose         Print latency for every request.

s3bench.exe ^
   -accessKey=minioadmin ^
   -accessSecret=minioadmin ^
   -bucket=bucket1 ^
   -endpoint=http://127.0.0.1:9000 ^
   -numClients=8 ^
   -numSamples=256 ^
   -objectNamePrefix=loadgen ^
   -objectSize=1024
pause
```

保存退出，双击 run-s3bench.cmd 运行：

```
D:\obs-tutorial-master>s3bench.exe    -accessKey=minioadmin    -accessSecret=minioadmin    -bucket=bucket1    -endpoint=http://127.0.0.1:9000    -numClients=8    -numSamples=
256    -objectNamePrefix=loadgen    -objectSize=1024
Test parameters
endpoint(s):      [http://127.0.0.1:9000]
bucket:           bucket1
objectNamePrefix: loadgen
objectSize:       0.0010 MB
numClients:       8
numSamples:       256
verbose:      %!d(bool=false)
```

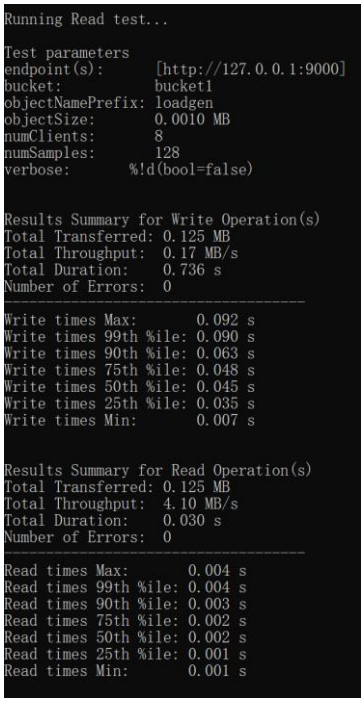
运行结果：

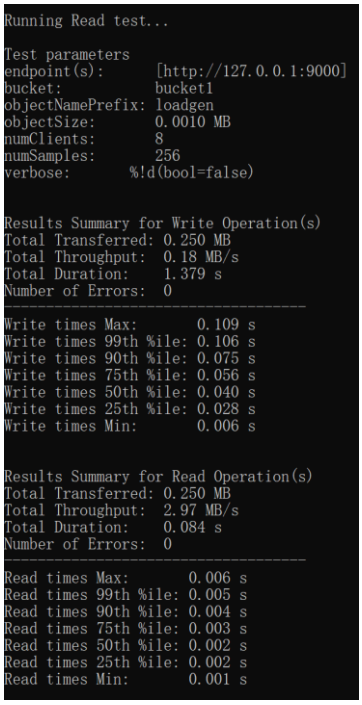
```
Results Summary for Write Operation(s)
Total Transferred: 0.250 MB
Total Throughput:  0.17 MB/s
Total Duration:    1.448 s
Number of Errors:  0
------------------------------------

Write times Max:       0.114 s
Write times 99th %ile: 0.112 s
Write times 90th %ile: 0.072 s
Write times 75th %ile: 0.060 s
Write times 50th %ile: 0.044 s
Write times 25th %ile: 0.031 s
Write times Min:       0.005 s
```

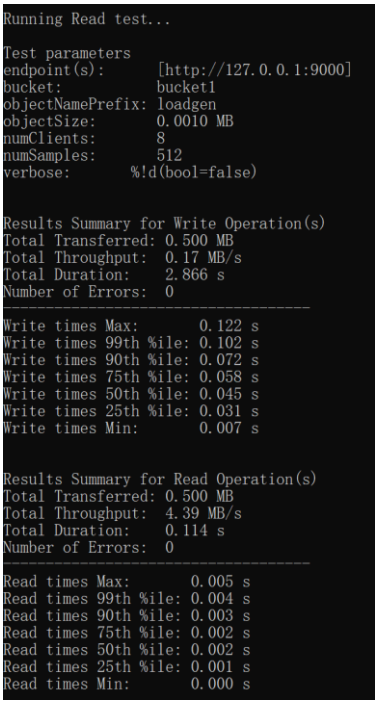
```
Results Summary for Read Operation(s)
Total Transferred: 0.250 MB
Total Throughput:  4.77 MB/s
Total Duration:    0.052 s
Number of Errors:  0
------------------------------------

Read times Max:       0.004 s
Read times 99th %ile: 0.003 s
Read times 90th %ile: 0.002 s
Read times 75th %ile: 0.002 s
Read times 50th %ile: 0.002 s
Read times 25th %ile: 0.001 s
Read times Min:       0.001 s
```

4.2 对象存储性能分析

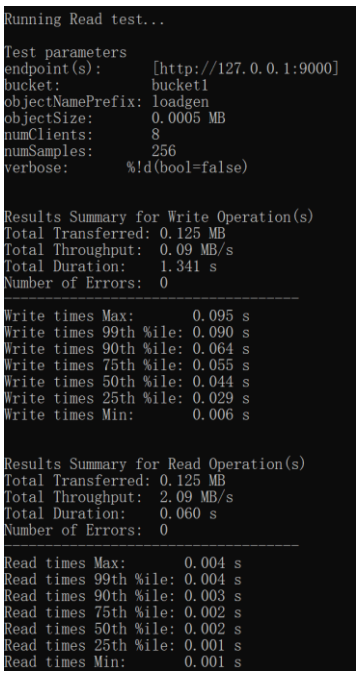
固定 objectSize 0.001MB，测试 numSamples 分别为 128、256、512 时的 结果，如下所示：

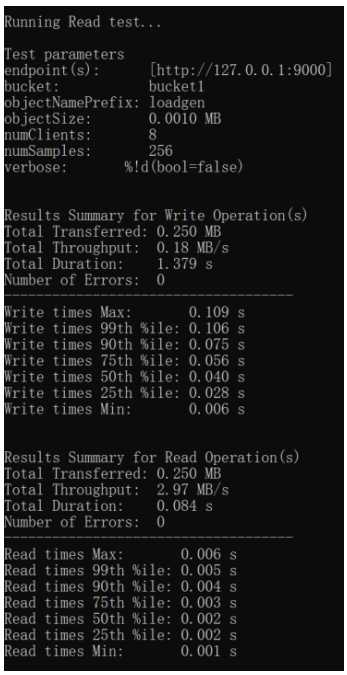


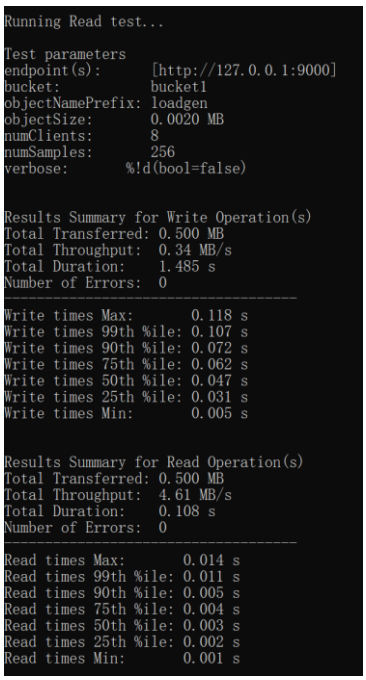




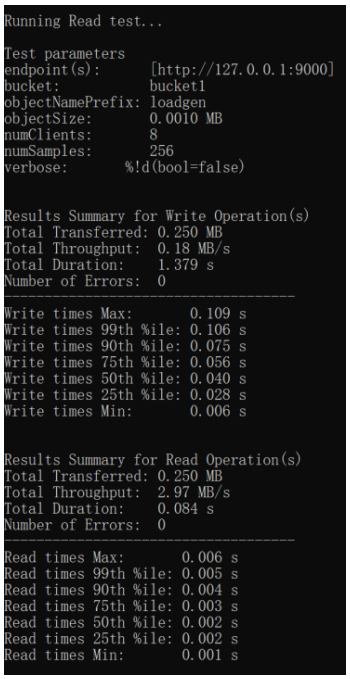
固定 numSamples 为 256，测试 objectSize 分别为 0.0005MB、0.001MB、0.002MB 时的结果，如下图所示：

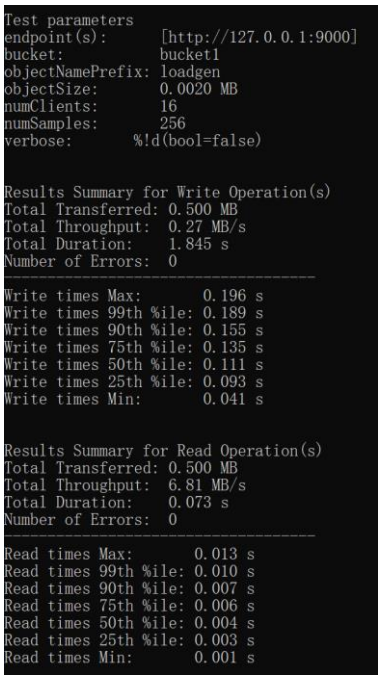


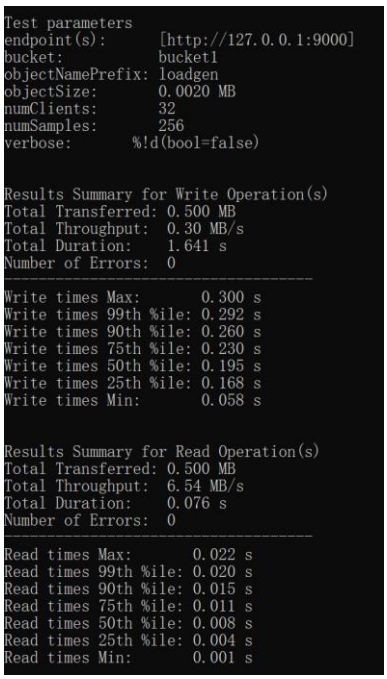




固定 numSamples 为 256， objectSize 为 0.001MB，测试并发数分别为 8、16、32 的结果，如下图所示：







五、实验总结

根据上述数据，观察可得以下结论。

1.块大小对读写效率的影响：
块越大，读写的时延越大；
这种趋势在尺寸较小时现象并不明显，但当块增大到一定程度时时延会快速增加；

2.并发数对性能的影响：
并发数越多，吞吐量、延迟和带宽都呈增加趋势；
其增加有一个饱和值，当趋于饱和时，服务器性能达到上限，出现堵塞。

参考文献

[1] ZHENG Q, CHEN H, WANG Y 等. COSBench: A Benchmark Tool for Cloud Object Storage Services[C]//2012 IEEE Fifth International Conference on Cloud Computing. 2012: 998－999.
[2] ARNOLD J. OpenStack Swift[M]. O'Reilly Media, 2014.
[3] WEIL S A, BRANDT S A, MILLER E L 等. Ceph: A Scalable, High-performance Distributed File System[C]//Proceedings of the 7th Symposium on Operating Systems Design and Implementation. Berkeley, CA, USA: USENIX Association, 2006: 307－320.
[4] Dean J, Barroso L A. Association for Computing Machinery, 2013. The Tail at Scale[J]. Commun. ACM, 2013, 56(2): 74－80.

[5] Delimitrou C, Kozyrakis C. Association for Computing Machinery, 2018. Amdahl’s Law for Tail Latency[J]. Commun. ACM, 2018, 61(8): 65–72.
（可以根据实际需要更新调整）