



2021 级

《数据中心》课程 实 验 报 告

姓 名 舒欢

学 号 M202173810

班 号 硕 2109 班

日 期 2022.01.05

目录

- 一、实验目的.....1
- 二、实验背景.....1
- 三、实验环境.....1
- 四、实验内容.....1
 - 实验一：系统搭建.....1
 - Buckets 内容.....3
 - 实验二：性能观测.....3
 - 实验三：尾延迟.....4
- 实验五 实验总结.....6

一、实验目的

- 1.熟悉对象存储技术，代表性系统即其特性；
- 2.时间对象存储系统，步数实验环境，进行初步测试；
- 3.基于对象存储系统，假设实际应用，示范主要功能；

二、实验背景

对象存储，也叫做基于对象的存储，是用来描述解决和处理离散单元的方法的通用术语，这些离散单元被称作为对象。

对象存储是云主机时代重要的一款产品，对象存储就和以前流行的云主机一样，同样不用不担心硬件的问题，和云主机一样，对象存储也可以叫做云存储，云存储主要是有两种公有云存储，比如各种网盘，当然现在也就百度网盘了；另外一种私有云存储，不同的云厂商对云存储的叫法不同：比如：阿里云（Object Storage Service）简称 OSS，腾讯云（loud Object Storage）简称 COS 百度云（Baidu Object Storage）简称 BOS 链接等。

三、实验环境

- 1.网络环境：本实验的完全使用个人笔记本电脑，没有接入互联网（PC 联网，运行环境不需要互联网）
- 2.语言环境：python 3.6.7（anaconda3）
- 3.实验工具：服务端 Minio，客户端 Minio-Client，吞吐量基准测试工具 s3bench

四、实验内容

实验一：系统搭建

利用 cmd 启动 minio 程序

```
F:\Code\Git\obs-tutorial-master>run-minio.cmd

+-----+
| You are running an older version of MinIO released 1 month ago |
| Update: Run `mc admin update`                                |
+-----+

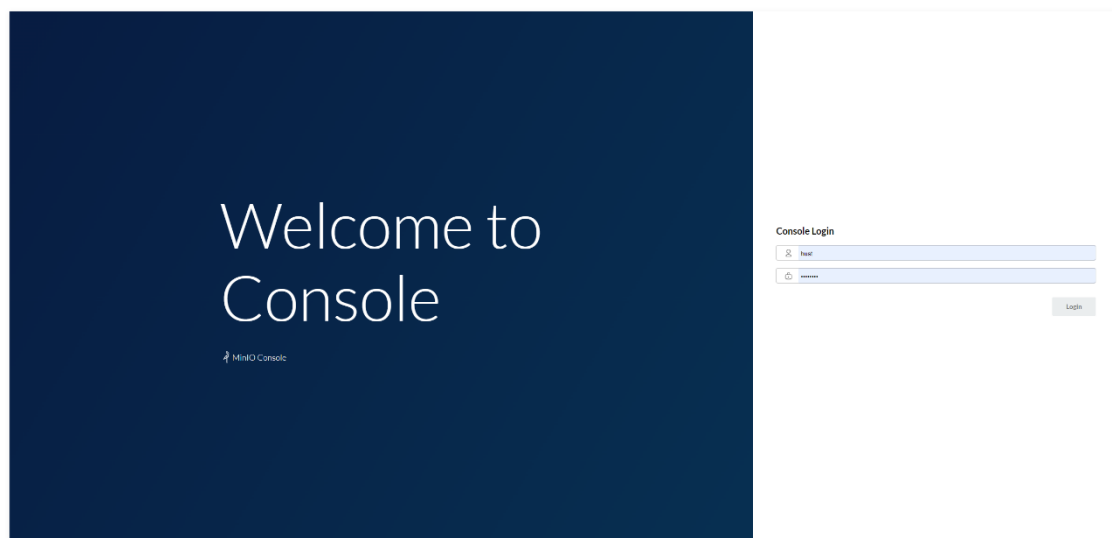
API: http://10.12.61.61:9000 http://127.0.0.1:9000
RootUser: hust
RootPass: hust_obs

Console: http://10.12.61.61:9090 http://127.0.0.1:9090
RootUser: hust
RootPass: hust_obs

Command-line: https://docs.min.io/docs/minio-client-quickstart-guide
$ mc.exe alias set myminio http://10.12.61.61:9000 hust hust_obs

Documentation: https://docs.min.io
```

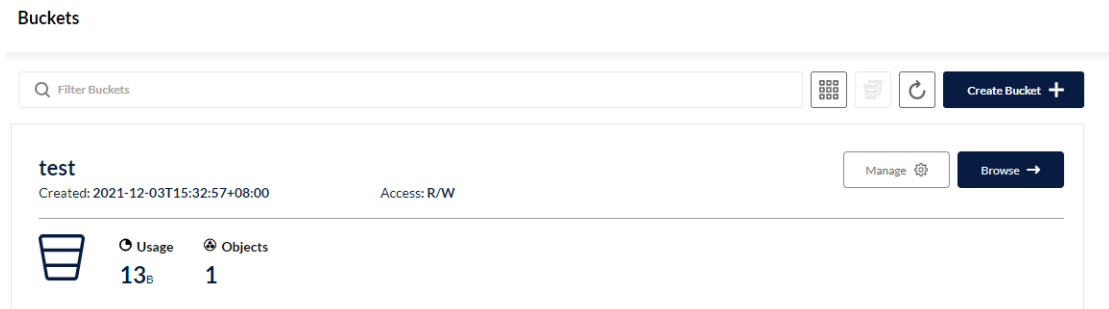
浏览器登录



服务端界面



Buckets 内容



实验二：性能观测

使用 s3bench 观测 minio 服务端系统的吞吐量、时延等，结果如下图所示

```
1366 x 749 owp\System32\cmd.exe - run-s3bench.cmd
Microsoft Windows [版本 10.0.18363.1556]
(c) 2019 Microsoft Corporation. 保留所有权利。

F:\Code\Git\obs-tutorial-master>run-s3bench.cmd

F:\Code\Git\obs-tutorial-master>s3bench.exe -accessKey=hust -accessSecret=hust_obs -bucket=test -endpoint=http://10.12.57.17:9000 -numClients=8 -
numSamples=256 -objectNamePrefix=loadgen -objectSize=1024
Test parameters
endpoint(s): [http://10.12.57.17:9000]
bucket: test
objectNamePrefix: loadgen
objectSize: 0.0010 MB
numClients: 8
numSamples: 256
verbose: %ld(bool=false)

Generating in-memory sample data... Done (1.9943ms)

Running Write test...
Running Read test...

Test parameters
endpoint(s): [http://10.12.57.17:9000]
bucket: test
objectNamePrefix: loadgen
objectSize: 0.0010 MB
numClients: 8
numSamples: 256
verbose: %ld(bool=false)

Results Summary for Write Operation(s)
Total Transferred: 0.250 MB
Total Throughput: 0.03 MB/s
Total Duration: 8.186 s
Number of Errors: 0
-----
Write times Max: 0.493 s
Write times 99th %ile: 0.453 s
Write times 90th %ile: 0.340 s
Write times 75th %ile: 0.280 s
Write times 50th %ile: 0.245 s
Write times 25th %ile: 0.212 s
Write times Min: 0.084 s

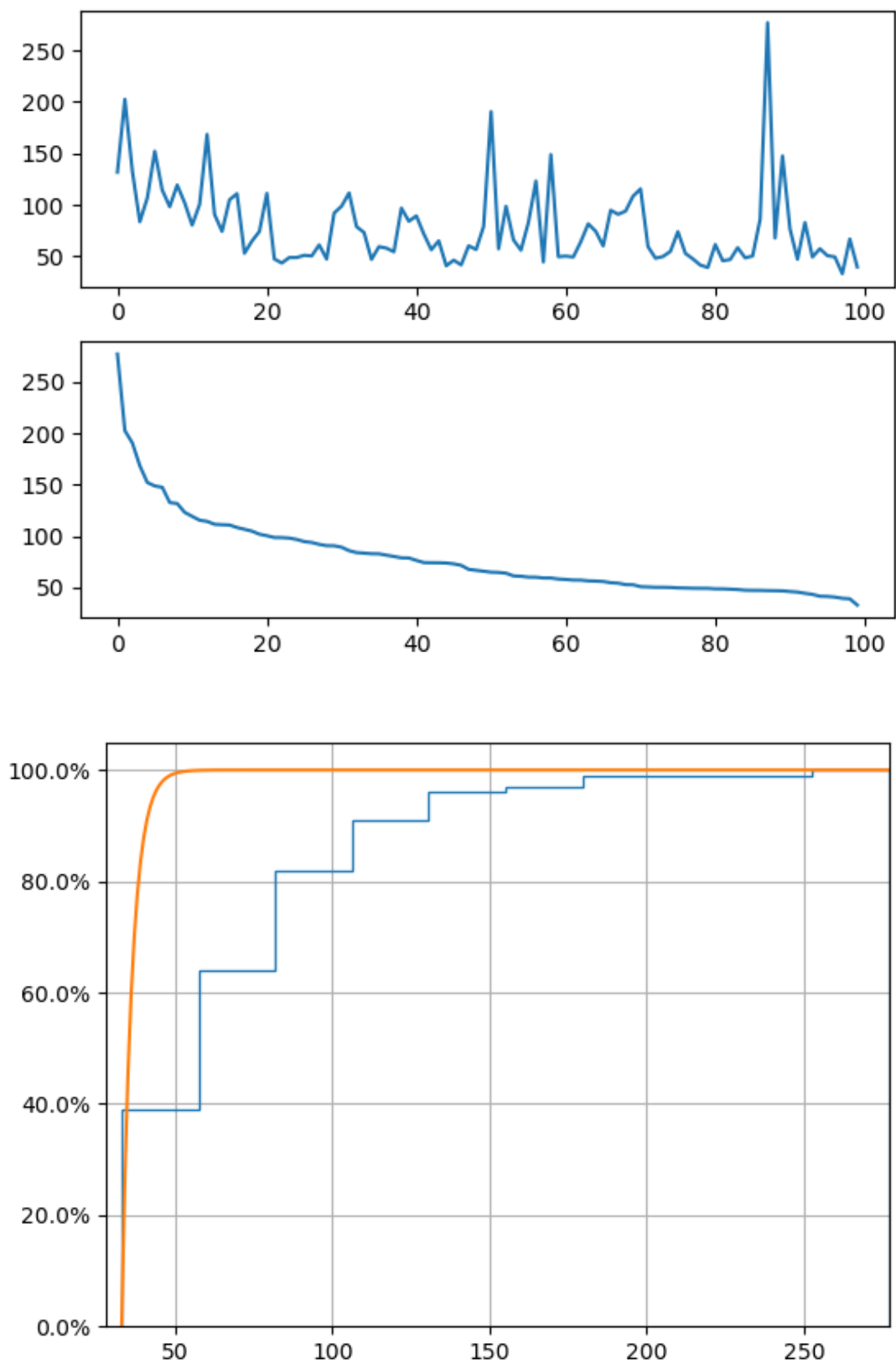
Results Summary for Read Operation(s)
Total Transferred: 0.250 MB
Total Throughput: 1.45 MB/s
Total Duration: 0.173 s
Number of Errors: 0
-----
Read times Max: 0.024 s
Read times 99th %ile: 0.022 s
Read times 90th %ile: 0.008 s
Read times 75th %ile: 0.006 s
Read times 50th %ile: 0.004 s
Read times 25th %ile: 0.003 s
Read times Min: 0.001 s

Cleaning up 256 objects...
Deleting a batch of 256 objects in range [0, 255]... Succeeded
Successfully deleted 256/256 objects in 683.7562ms

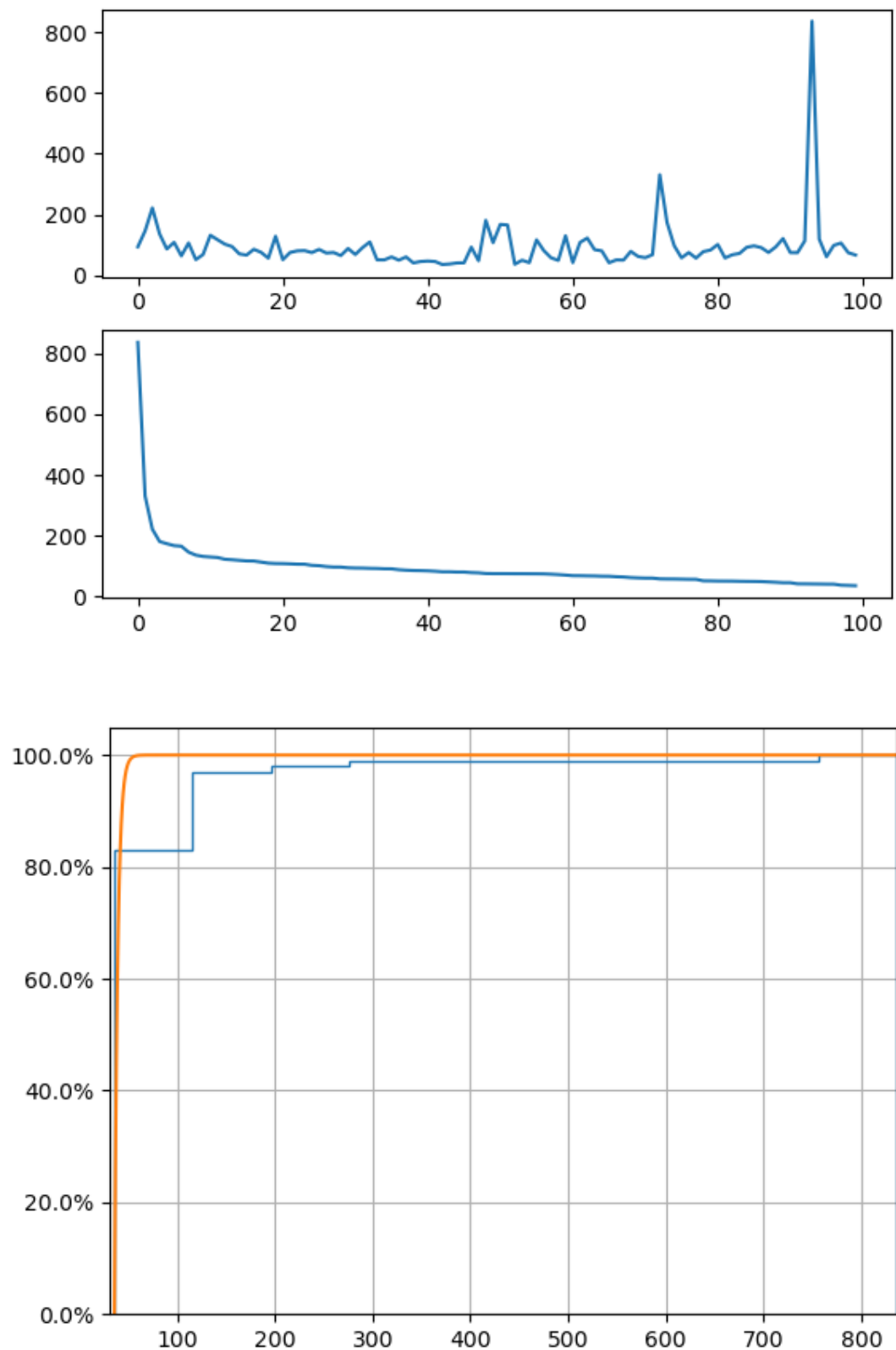
F:\Code\Git\obs-tutorial-master>pause
请按任意键继续. . .
```

实验三：尾延迟

通过运行 obs-tutorial 中的 python 脚本，得到如下结果



由上图可知，当数据传输达到 93%左右是，出现剧烈的抖动，此时对其进行打流，从而减少尾延迟的影响，下图是尝试进行对冲请求优化后的结果：



实验五 实验总结

本次实验是我首次接触对象存储的概念，一开始的时候，比较慌张，因为之前没有接触过，对于实验所用的工具流程都比较陌生，同时，对于对象存储的理论知识也是在上课的时候才开始接触。

所以本次实验完成的比较基础，都是按照老师提供的脚本运行、配置，然后观察最后的结果。总体来说，没有遇到太大的困难，一些环境的问题，也比较容易就解决了。同时整个实验流程做下来，让我对对象存储方面的知识有了一定的了解，让我了解到了一些自己研究领域外的其他领域的知识，开阔了自己的视野，拓宽了自己的知识面，非常感谢老师设置这次实验，让我收获良多。