

2021 级

《数据中心技术》课程
实 验 报 告

姓 名 王楠

学 号 M202173782

班 号 硕 2108 班

日 期 2022.1.4

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
五、实验过程.....	1
六、实验总结.....	9
参考文献.....	10

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，架设实际应用，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。

就像文件一样，对象包含数据，但是和文件不同的是，对象在一个层结构中不会再有层级结构，每个对象都在一个被称作存储池的扁平地址空间的同一级别里，一个对象不会属于另一个对象的下一级。文件和对象都有与它们所包含的数据相关的元数据，但是对象是以扩展元数据为特征的。每个对象都被分配一个唯一的标识符，允许一个服务器或者最终用户来检索对象，而不必知道数据的物理地址，这种方法对于在云计算环境中自动化和简化数据存储有帮助。

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，一种是公有云存储，比如各种网盘，如常用的百度网盘；另外一种是有私有云存储，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（Cloud Object Storage）简称 COS，百度云（Baidu Object Storage）简称 BOS 等。

三、实验环境

操作系统	Windows 10 64 位
处理器	11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz 2.30 GHz
内存	16.0 GB
Go 版本	1.17.5
Server	Minio
Client	Minio Client
Test	S3 Bench

四、实验内容

1. 熟悉性能指标：吞吐率、带宽、延迟；
2. 分析不同负载下的指标、延迟的分布；
3. 观测尾延迟现象；
4. 尝试对冲请求方案。

五、实验过程

5.1 熟悉性能指标：吞吐率、带宽、延迟

通常是以 3 个指标来衡量网络的性能，分别是吞吐率、带宽、延时，它们表示的意义如下。

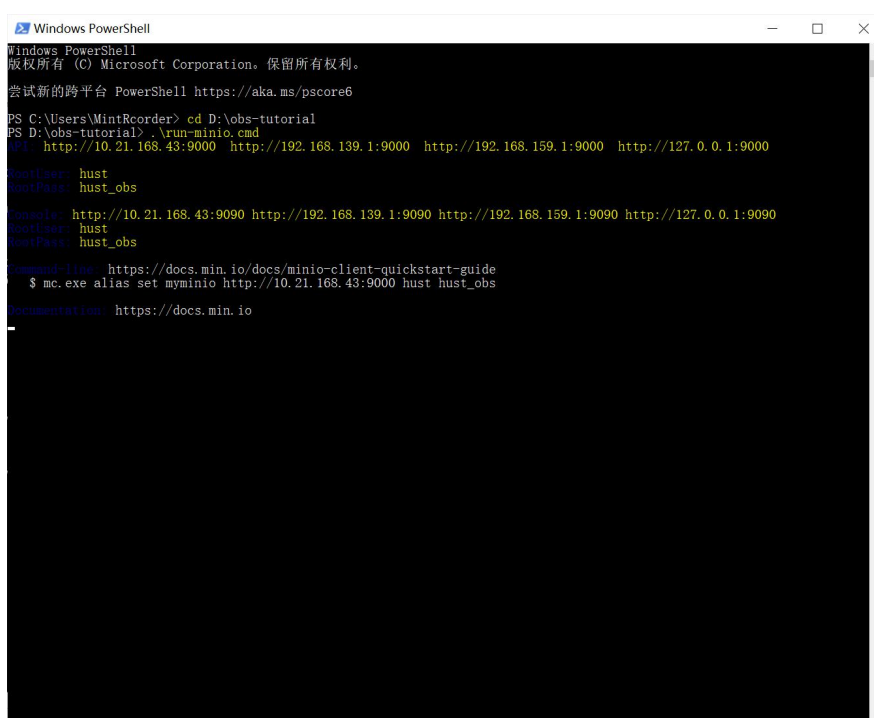
1.吞吐率，表示单位时间内成功传输的数据量，单位是 b/s（比特/秒）或者 B/s（字节/秒），吞吐受带宽限制，带宽越大，吞吐率的上限才可能越高。

2.带宽，表示链路的最大传输速率，单位是 b/s（比特/秒），带宽越大，其传输能力就越强。

3.延时，表示请求数据包发送后，收到对端响应，所需要的时间延迟。不同的场景有着不同的含义，比如可以表示建立 TCP 连接所需的时间延迟，或一个数据包往返所需的时间延迟。

5.2 分析不同负载下的指标、延迟的分布

安装 Go 语言环境，进入 minio 官网，下载 minio.exe 文件到 D:\obs-tutorial 目录下，在 Windows PowerShell 中先使用 cd 命令进入目录，执行.\run-minio.cmd 命令配置 minio 服务端，取得服务端地址。

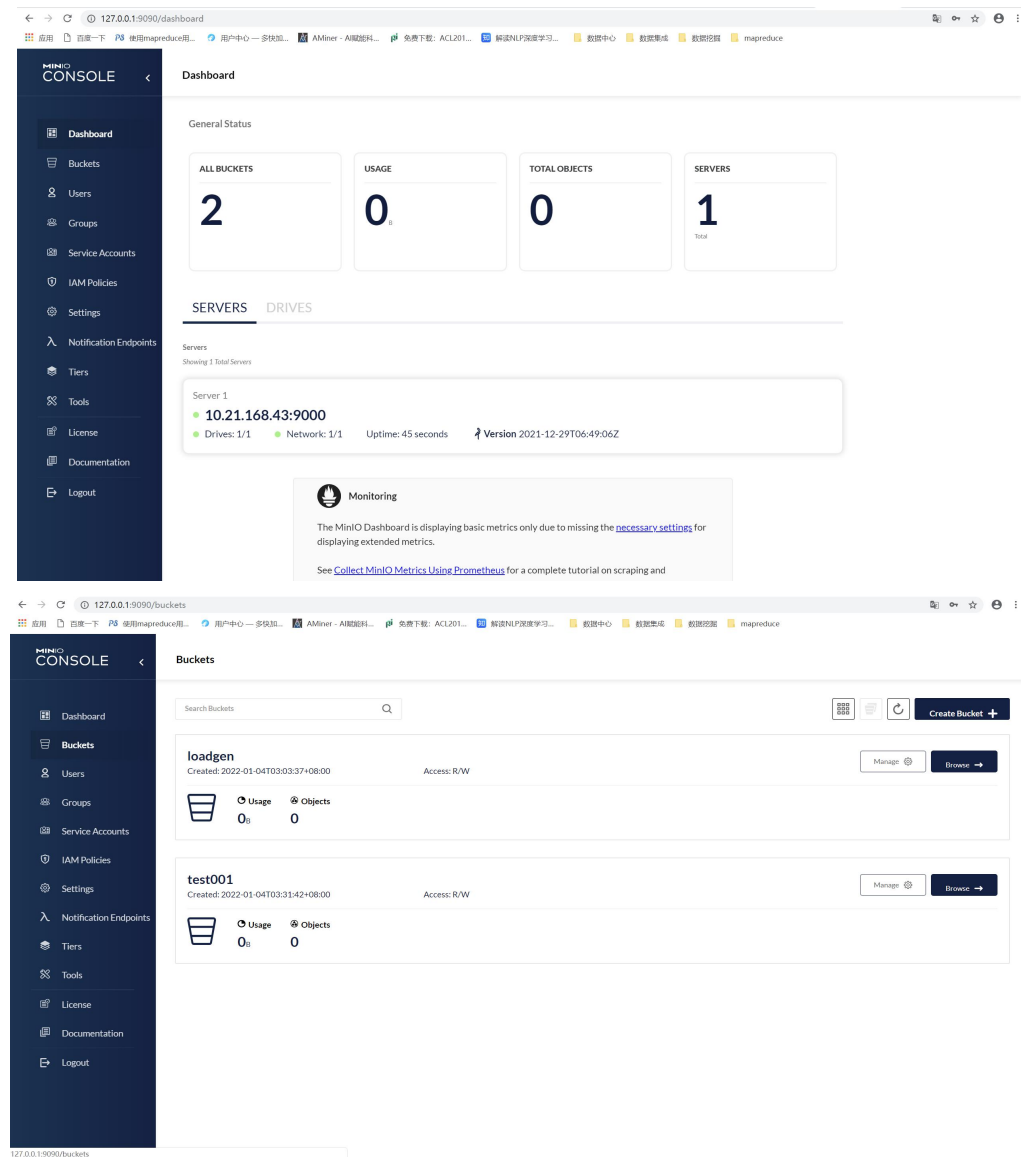
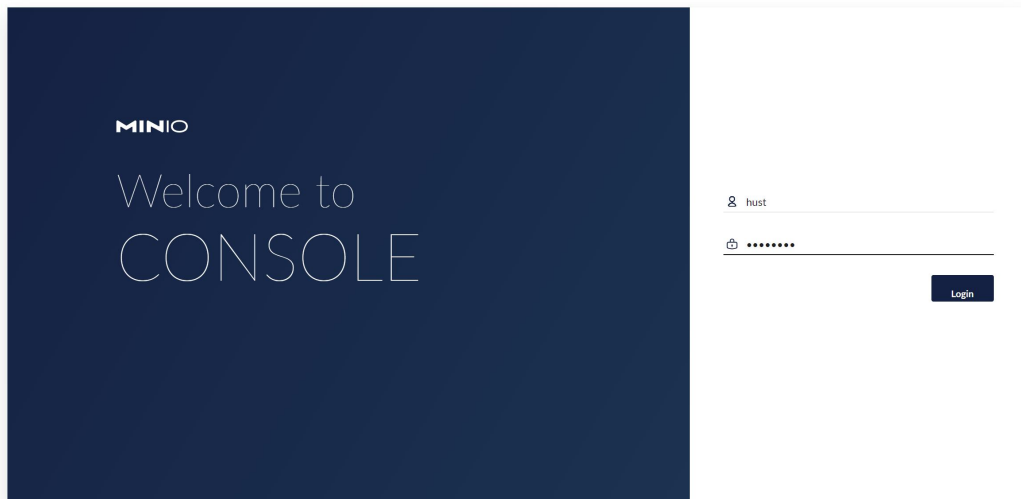


```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\MintRecorder> cd D:\obs-tutorial
PS D:\obs-tutorial> .\run-minio.cmd
ip: http://10.21.168.43:9000 http://192.168.139.1:9000 http://192.168.159.1:9000 http://127.0.0.1:9000
rootUser: hust
rootPass: hust_obs
Console: http://10.21.168.43:9090 http://192.168.139.1:9090 http://192.168.159.1:9090 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://10.21.168.43:9000 hust hust_obs
Documentation: https://docs.min.io
```

通过浏览器登陆 127.0.0.1:9000 查看效果，用设定的用户名和密码登录，新建一个名为“loadgen”的 Bucket，完成系统搭建步骤。



使用评测工具 S3 Bench，通过命令 `go get -u github.com/igneous-systems/`

s3bench 安装 S3 Bench，将预编译 Windows 执行程序下载放置在目录中。

```
D:\obs-tutorial>pause
请按任意键继续. . .
PS D:\obs-tutorial> ls

目录: D:\obs-tutorial

Mode                LastWriteTime         Length Name
----                -
d-----          2022/1/4           1:58      certs
d-----          2022/1/4          17:34      root
              2021/12/30          10:37          73 .gitignore
              2021/12/30          10:37         396 config-osm.cmd
              2021/12/30          10:37         755 config-osm.sh
              2021/12/30          10:37        4241 known-issues.md
              2021/12/30          10:37        7416 latency-collect.ipynb
              2021/12/30          10:37       33578 latency-plot.ipynb
      -a-----          2022/1/4          17:34        1918 latency.csv
              2021/12/30          10:37       20133 LICENSE
      -a-----          2022/1/4           1:06    120404200 minio.exe
              2021/12/30          10:37       17022 obs-simulation.ipynb
              2021/12/30          10:37       10384 README.md
              2021/12/30          10:37       10308 README.zh.cn.md
              2021/12/30          10:37         635 run-minio.cmd
              2021/12/30          10:37         631 run-minio.sh
              2021/12/30          10:37        150 run-mock-s3.cmd
              2021/12/30          10:37        136 run-mock-s3.sh
      -a-----          2022/1/4          18:41        723 run-s3bench.cmd
              2021/12/30          10:37       1263 run-s3bench.sh
              2021/12/30          10:37       1417 run-s3proxy.sh
              2021/12/30          10:37        412 run-zfile.cmd
      -a-----          2022/1/4           2:21    11873792 s3bench.exe
              2021/12/30          10:37        4302 workload-example.xml

PS D:\obs-tutorial>
```

执行 `.run-s3bench.cmd` 命令开始 s3bench 基准测试。

```

Windows PowerShell
请按任意键继续. . .
PS D:\obs-tutorial> .\run-s3bench.cmd

D:\obs-tutorial>s3bench.exe -accessKey=hust -accessSecret=hust_obs -bucket=loadgen -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: loadgen
objectNamePrefix: loadgen
objectSize: 0.0010 MB
numClients: 8
numSamples: 256
verbose: %!d(bool=false)

Generating in-memory sample data... Done (0s)

Running Write test...

Running Read test...

Test parameters
endpoint(s): [http://127.0.0.1:9000]
bucket: loadgen
objectNamePrefix: loadgen
objectSize: 0.0010 MB
numClients: 8
numSamples: 256
verbose: %!d(bool=false)

Results Summary for Write Operation(s)
Total Transferred: 0.250 MB
Total Throughput: 0.56 MB/s
Total Duration: 0.448 s
Number of Errors: 0
-----
Write times Max: 0.057 s
Write times 99th %ile: 0.057 s
Write times 90th %ile: 0.031 s
Write times 75th %ile: 0.020 s
Write times 50th %ile: 0.008 s
Write times 25th %ile: 0.006 s
Write times Min: 0.004 s

Results Summary for Read Operation(s)
Total Transferred: 0.250 MB
Total Throughput: 6.67 MB/s
Total Duration: 0.037 s
Number of Errors: 0
-----
Read times Max: 0.006 s
Read times 99th %ile: 0.004 s
Read times 90th %ile: 0.002 s
Read times 75th %ile: 0.001 s
Read times 50th %ile: 0.001 s
Read times 25th %ile: 0.001 s
Read times Min: 0.000 s

Cleaning up 256 objects...
Deleting a batch of 256 objects in range [0, 255]... Succeeded
Successfully deleted 256/256 objects in 188.2125ms

D:\obs-tutorial>pause
请按任意键继续.

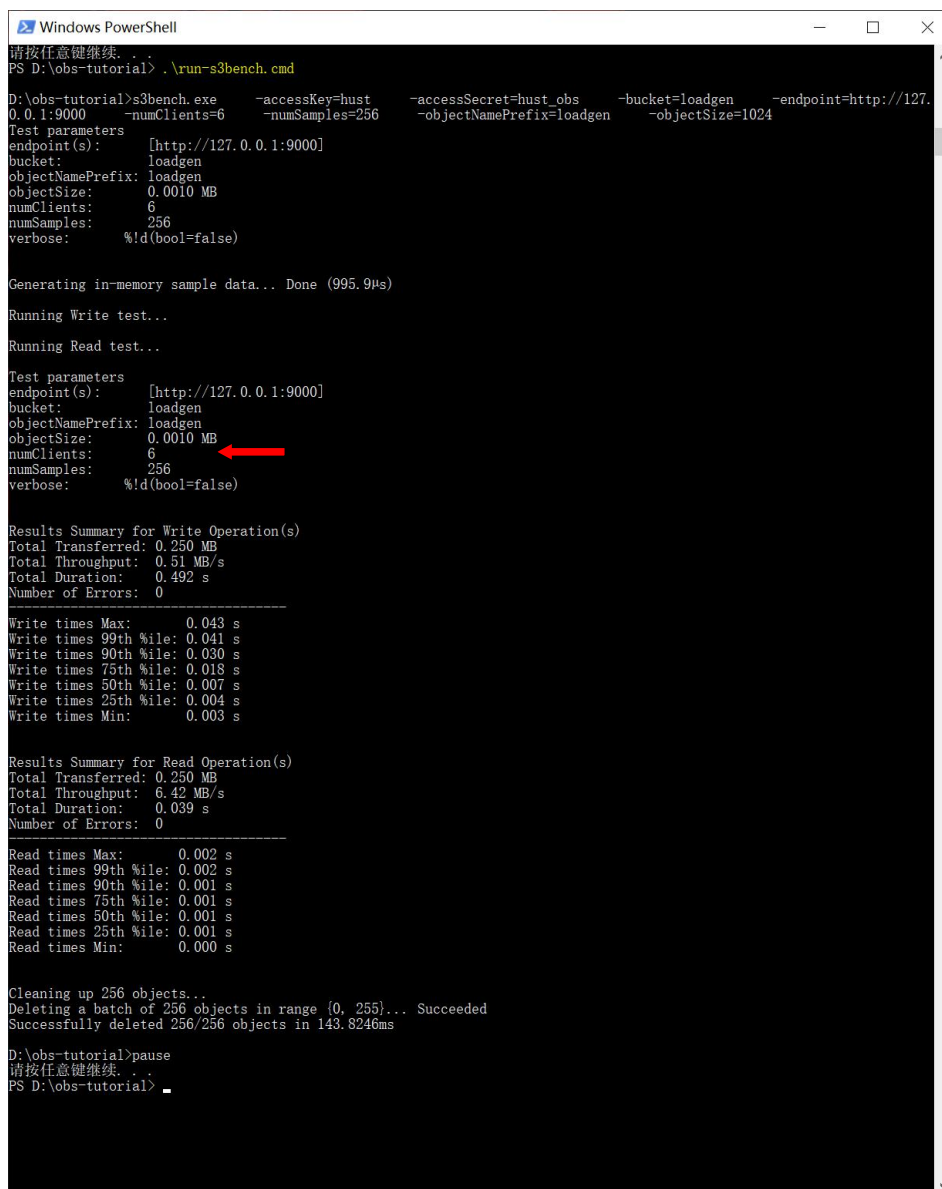
```

观察执行结果，Total transferred 所有请求的响应数据长度总和为 0.25MB，有

共计 256 个大小相同的对象。在写操作中：总吞吐率为 0.56MB/s，总耗时 0.448s。最长耗时 0.057s，最短耗时 0.004s。99%的在 0.057s 内完成，90%的在 0.031s 内完成。读操作中：总吞吐率为 6.67MB/s，总耗时 0.037s。最长耗时 0.006s，最短耗时 0.000s。容易看到 75%的操作可以在 0.001s 内完成，而统计 90%的操作却需要两倍即 0.002s 以内的时间，统计 99%的操作则需更进一步在 90%的时间上再翻倍，即符合长尾分布。

测试不同负载，通过更改脚本参数 numclient 和 objectsize，分别测试 numclient 为 6、8、10 以及 objectsize 为 512、1024、2048 时的情况。

Numclient 为 6。



```
Windows PowerShell
请任意键继续...
PS D:\obs-tutorial> .\run-s3bench.cmd

D:\obs-tutorial>s3bench.exe -accessKey=hust -accessSecret=hust_obs -bucket=loadgen -endpoint=http://127.0.0.1:9000
0.0.1:9000 -numClients=6 -numSamples=256 -objectNamePrefix=loadgen -objectSize=1024

Test parameters
endpoint(s): [http://127.0.0.1:9000]
bucket: loadgen
objectNamePrefix: loadgen
objectSize: 0.0010 MB
numClients: 6
numSamples: 256
verbose: %!d(bool=false)

Generating in-memory sample data... Done (995.9µs)

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

Test parameters
endpoint(s): [http://127.0.0.1:9000]
bucket: loadgen
objectNamePrefix: loadgen
objectSize: 0.0010 MB
numClients: 6
numSamples: 256
verbose: %!d(bool=false)

Results Summary for Write Operation(s)
Total Transferred: 0.250 MB
Total Throughput: 0.51 MB/s
Total Duration: 0.492 s
Number of Errors: 0

Write times Max: 0.043 s
Write times 99th %ile: 0.041 s
Write times 90th %ile: 0.030 s
Write times 75th %ile: 0.018 s
Write times 50th %ile: 0.007 s
Write times 25th %ile: 0.004 s
Write times Min: 0.003 s

Results Summary for Read Operation(s)
Total Transferred: 0.250 MB
Total Throughput: 6.42 MB/s
Total Duration: 0.039 s
Number of Errors: 0

Read times Max: 0.002 s
Read times 99th %ile: 0.002 s
Read times 90th %ile: 0.001 s
Read times 75th %ile: 0.001 s
Read times 50th %ile: 0.001 s
Read times 25th %ile: 0.001 s
Read times Min: 0.000 s

Cleaning up 256 objects...
Deleting a batch of 256 objects in range {0, 255}... Succeeded
Successfully deleted 256/256 objects in 143.8246ms

D:\obs-tutorial>pause
请任意键继续...
PS D:\obs-tutorial>
```

Numclient 为 10。

```
PS D:\obs-tutorial> .\run-s3bench.cmd

D:\obs-tutorial>s3bench.exe -accessKey=hust -accessSecret=hust_obs -bucket=loadgen -endpoint=http://127.0.0.1:9000 -numClients=10 -numSamples=256 -objectNamePrefix=loadgen -objectSize=1024

Test parameters
endpoint(s): [http://127.0.0.1:9000]
bucket: loadgen
objectNamePrefix: loadgen
objectSize: 0.0010 MB
numClients: 10
numSamples: 256
verbose: %!d(bool=false)

Generating in-memory sample data... Done (1.9996ms)

Running Write test...

Running Read test...

Test parameters
endpoint(s): [http://127.0.0.1:9000]
bucket: loadgen
objectNamePrefix: loadgen
objectSize: 0.0010 MB
numClients: 10
numSamples: 256
verbose: %!d(bool=false)

Results Summary for Write Operation(s)
Total Transferred: 0.250 MB
Total Throughput: 0.19 MB/s
Total Duration: 1.299 s
Number of Errors: 0
-----
Write times Max: 0.108 s
Write times 99th %ile: 0.108 s
Write times 90th %ile: 0.078 s
Write times 75th %ile: 0.062 s
Write times 50th %ile: 0.047 s
Write times 25th %ile: 0.036 s
Write times Min: 0.004 s

Results Summary for Read Operation(s)
Total Transferred: 0.250 MB
Total Throughput: 9.27 MB/s
Total Duration: 0.027 s
Number of Errors: 0
-----
Read times Max: 0.003 s
Read times 99th %ile: 0.002 s
Read times 90th %ile: 0.002 s
Read times 75th %ile: 0.001 s
Read times 50th %ile: 0.001 s
Read times 25th %ile: 0.001 s
Read times Min: 0.000 s

Cleaning up 256 objects...
Deleting a batch of 256 objects in range {0, 255}... Succeeded
Successfully deleted 256/256 objects in 143.0406ms

D:\obs-tutorial>pause
请按任意键继续. . .
```


Objectsize 为 512。

```
PS D:\obs-tutorial> .\run-s3bench.cmd

D:\obs-tutorial>s3bench.exe -accessKey=hust -accessSecret=hust_obs -bucket=loadgen -endpoint=http://127.0.0.1:9000 -numClients=8 -numSamples=256 -objectNamePrefix=loadgen -objectSize=512

Test parameters
endpoint(s): [http://127.0.0.1:9000]
bucket: loadgen
objectNamePrefix: loadgen
objectSize: 0.0005 MB
numClients: 8
numSamples: 256
verbose: %!d(bool=false)

Generating in-memory sample data... Done (999.4µs)

Running Write test...

Running Read test...

Test parameters
endpoint(s): [http://127.0.0.1:9000]
bucket: loadgen
objectNamePrefix: loadgen
objectSize: 0.0005 MB
numClients: 8
numSamples: 256
verbose: %!d(bool=false)

Results Summary for Write Operation(s)
Total Transferred: 0.125 MB
Total Throughput: 0.12 MB/s
Total Duration: 1.053 s
Number of Errors: 0

-----
Write times Max: 0.081 s
Write times 99th %ile: 0.069 s
Write times 90th %ile: 0.056 s
Write times 75th %ile: 0.042 s
Write times 50th %ile: 0.034 s
Write times 25th %ile: 0.016 s
Write times Min: 0.008 s

Results Summary for Read Operation(s)
Total Transferred: 0.125 MB
Total Throughput: 4.41 MB/s
Total Duration: 0.028 s
Number of Errors: 0

-----
Read times Max: 0.002 s
Read times 99th %ile: 0.002 s
Read times 90th %ile: 0.001 s
Read times 75th %ile: 0.001 s
Read times 50th %ile: 0.001 s
Read times 25th %ile: 0.001 s
Read times Min: 0.000 s

Cleaning up 256 objects...
Deleting a batch of 256 objects in range {0, 255}... Succeeded
Successfully deleted 256/256 objects in 143.1689ms

D:\obs-tutorial>pause
请按任意键继续. . .
```

Objectsize 为 2048。

```
PS D:\obs-tutorial> .\run-s3bench.cmd
D:\obs-tutorial>s3bench.exe -accessKey=hust -accessSecret=hust_obs -bucket=loadgen -endpoint=http://127.0.1:9000
0.0.1:9000 -numClients=8 -numSamples=256 -objectNamePrefix=loadgen -objectSize=2048
Test parameters
endpoint(s): [http://127.0.0.1:9000]
bucket: loadgen
objectNamePrefix: loadgen
objectSize: 0.0020 MB
numClients: 8
numSamples: 256
verbose: %!d(bool=false)

Generating in-memory sample data... Done (994.8µs)

Running Write test...

Running Read test...

Test parameters
endpoint(s): [http://127.0.0.1:9000]
bucket: loadgen
objectNamePrefix: loadgen
objectSize: 0.0020 MB
numClients: 8
numSamples: 256
verbose: %!d(bool=false)

Results Summary for Write Operation(s)
Total Transferred: 0.500 MB
Total Throughput: 0.61 MB/s
Total Duration: 0.821 s
Number of Errors: 0
-----
Write times Max: 0.075 s
Write times 99th %ile: 0.069 s
Write times 90th %ile: 0.048 s
Write times 75th %ile: 0.039 s
Write times 50th %ile: 0.024 s
Write times 25th %ile: 0.008 s
Write times Min: 0.004 s

Results Summary for Read Operation(s)
Total Transferred: 0.500 MB
Total Throughput: 13.53 MB/s
Total Duration: 0.037 s
Number of Errors: 0
-----
Read times Max: 0.003 s
Read times 99th %ile: 0.003 s
Read times 90th %ile: 0.002 s
Read times 75th %ile: 0.001 s
Read times 50th %ile: 0.001 s
Read times 25th %ile: 0.001 s
Read times Min: 0.000 s

Cleaning up 256 objects...
Deleting a batch of 256 objects in range {0, 255}... Succeeded
Successfully deleted 256/256 objects in 153.3618ms

D:\obs-tutorial>pause
请按任意键继续. . .
```

5.3 观测尾延迟现象

安装 anaconda3、python3.7、vs code，打开目录对 latency-collect 和 latency-plot 的代码进行修改以适用于自己搭建的 minio 服务端，通过执行 latency-collect 获取尾延迟分布数据，接着执行 latency-plot 画出延迟分布图像和排队论模型预测。

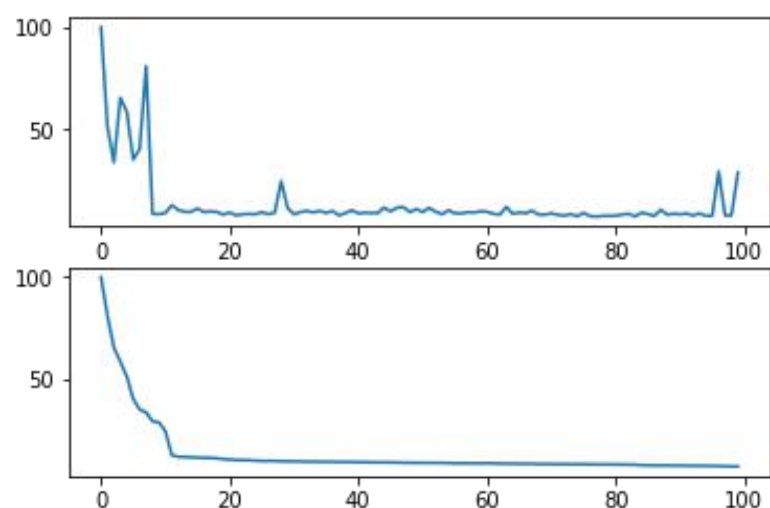


图 1

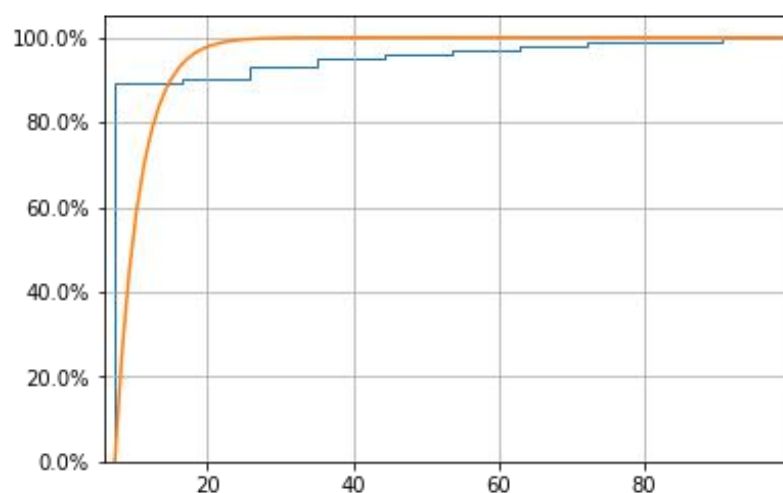


图 2

从图 1 上面的图像可以看出，在实验过程中，一开始延迟很大且波动剧烈，中间基本稳定，但也存在偶尔延迟突然小幅度增大的情况。下面的图像，可以看出，延迟高的请求只有前面部分，大多数情况下延迟较低。

第二张图蓝色线绘制的百分位延迟分布，黄色线是排队论模型拟合实测数据得到的情况。蓝色线呈指数累计概率分布，多达 85% 的请求可以在 10ms 以内完成，至于 95% 的请求需 30 多毫秒以内，对于不同实验，某个百分位的出现可能更小也可能更大，发起对冲请求是否合理需根据分布判断，与平均延迟时间、最大最小值没有直接关系。越复杂的系统部件越多，难保证每个部件无误工作，哪怕出现细小瑕疵，都会出现如黄色线的陡峭尾延迟问题。

5.4 实验中遇到的问题

没有根据自己搭建的 minio 服务端修改 latency-collect 的代码，导致报错。执行 `.\run-minio.cmd` 命令取得服务端地址。

六、实验总结

通过这次实验，学习了如何搭建 minio 服务器，并学会了如何通过网页客户端

进行文件上传和下载，并通过 **S3 Bench** 来测试其读写效率。学习了一些 **GitHub** 的知识，为之后的工作和学习打下了一些基础。

参考文献

- [1] ARNOLD J. OpenStack Swift[M]. O' Reilly Media, 2014.
- [2] 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.
- [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.