



2020 级

《大数据存储与管理》课程
实 验 报 告

姓 名 刘颖杰

学 号 U201914992

班 号 CS1903 班

日 期 2023.05.29

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
4.1 对象存储技术实践	2
4.2 对象存储性能分析	2
五、实验过程	2
六、实验总结	3
参考文献	6

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

对象存储是一种通用术语，用于描述解决问题和处理离散单元的方法。它主要针对存储大容量非结构化数据，如图片、视频、日志文件、备份数据和容器/虚拟机镜像等。当前已有基于块和基于文件的存储系统，它们各自有一些优缺点。基于块的存储系统，磁盘块通过底层存储协议访问，读写速率快但不利于共享。基于文件的存储系统以文件为传输协议，读写速率慢但利于共享。对象存储可以发扬它俩各自的优点，既克服了块存储与文件存储各自的缺点，又提供高可靠、跨平台以及安全的数据共享的存储体系结构。在对象存储系统中，Minio 是一个开源的、基于 Apache License v2.0 开源协议的对象存储服务。

三、实验环境

硬件环境	版本
操作系统	ubuntu-20.04.1
虚拟机软件	VMware Workstation 14 Pro
CPU	AMD Ryzen 9 5900H
分配内存	2GB
软件环境	版本
客户端	minio
测试程序	S3bench
服务器 IP	Localhost:9000

四、实验内容

本次实验是对象存储技术相关实验，使用 minio、minio client、和 s3bench 等工具完成对数据的存储和分析。

4.1 对象存储技术实践

- 1.配置 Minio。向配置好的客户端发送文件。
- 2.配置 mc。在远端利用 mc 新建 bucket。
- 3.下载 s3bench 脚本，修改相关参数进行相应的性能测试。

4.2 对象存储性能分析

- 1.对 s3bench 脚本运行输出的结果进行相应的记录和处理。
- 2.改变不同参数观察不同的性能指标。
- 3.对相应的实验数据做出记录。

五、实验过程

5.1 对象存储技术实践

- 1.输入以下指令：wget <https://dl.min.io/server/minio/release/linux-amd64/minio> 来下载程序文件
 2. 使用 chmod +x minio 命令来给 minio 可执行权限
 3. 用以下命令设置账号密码以及监听端口
- MINIO_ROOT_USER=U201914992 MINIO_ROOT_PASSWORD=ba123 ./minio server ./data --console-address ":9099"。

```
Access key length should be at least 3, and secret key length at least 8 characters
root@ubuntu:/opt/minio# MINIO_ROOT_USER=U201915085 MINIO_ROOT_PASSWORD=wzywzy123 ./minio server ./data --console-address ":9099"
API: http://192.168.240.132:9000 http://127.0.0.1:9000
RootUser: U201915085
RootPass: wzywzy123

Console: http://192.168.240.132:9099 http://127.0.0.1:9099
RootUser: U201915085
RootPass: wzywzy123

Command-line: https://docs.min.io/docs/minio-client-quickstart-guide
$ mc alias set myminio http://192.168.240.132:9000 U201915085 wzywzy123
Documentation: https://docs.min.io
```

图 1

4. 下载 minio 客户端 mc，并赋予可执行权限
- wget https://dl.min.io/client/mc/release/linux-amd64/mc
- chmod +x mc
- 5.根据提示，设置 IP 等信息。
 6. 在浏览器访问服务器。在浏览器中输入 http://127.0.0.1:9000 可以访问服务器，

登录界面如图 4-2。确认登录后，可以看到界面。

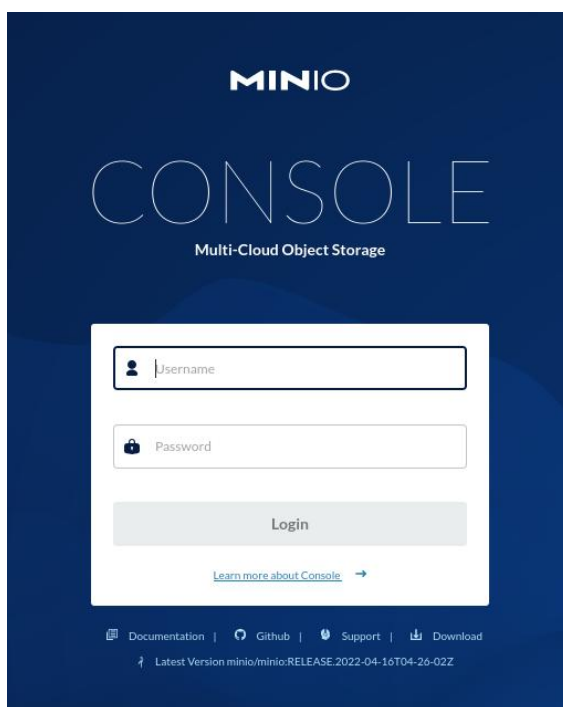


图 2

登录服务器

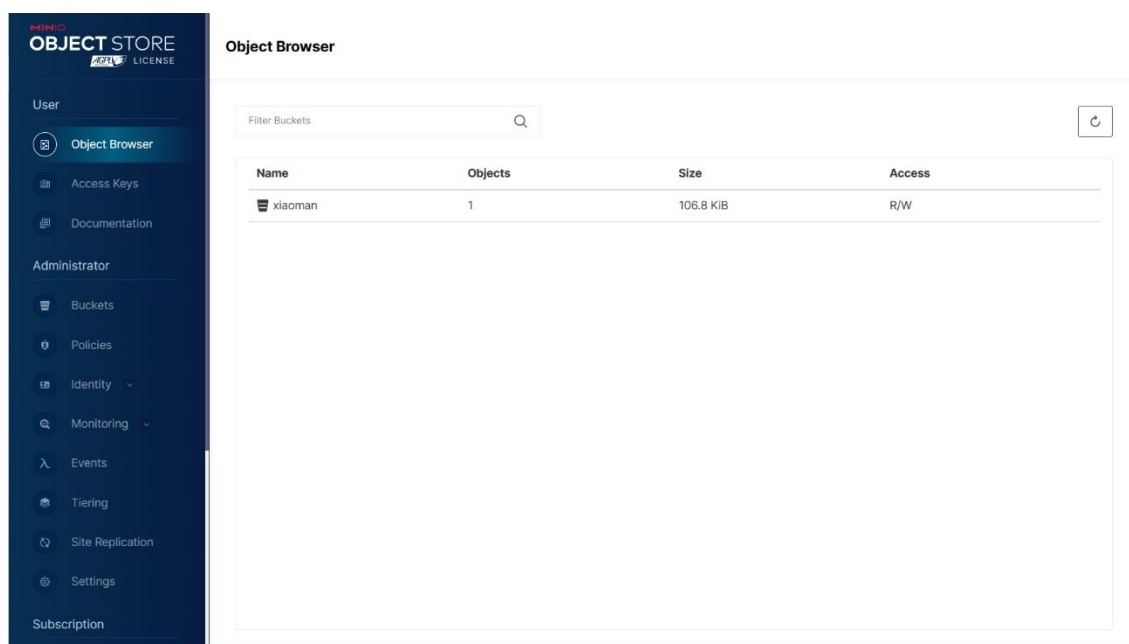


图 3

创建另外一个桶 `eeee`，并在本地创建文件 `qaz.txt`，然后将该文件上传到该桶当中。

5.2 对象存储性能分析

1. 下载安装 s3bench: `go get -u github.com/igneous-systems/s3bench`, 并修改相关参数, 使用用户名密码, IP 和 minio 服务器一致。

2. 设置测试用的并行客户端数量为 8 个, 数据对象大小为 32KB, 对象数量为 256 进行实验

```
Test parameters
endpoint(s):      [http://127.0.0.1:9000]
bucket:           test1
objectNamePrefix: test1
objectSize:       0.0010 MB
numClients:       8
numSamples:       256
verbose:          %!d(bool=false)

Generating in-memory sample data... Done (3.9908ms)

Running Write test...

Running Read test...

Test parameters
endpoint(s):      [http://127.0.0.1:9000]
bucket:           test1
objectNamePrefix: test1
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:  0.10 MB/s
Total Duration:    2.426 s
Number of Errors:  0
-----
Write times Max:      0.172 s
Write times 99th %ile: 0.166 s
Write times 90th %ile: 0.102 s
Write times 75th %ile: 0.093 s
Write times 50th %ile: 0.077 s
Write times 25th %ile: 0.055 s
Write times Min:      0.017 s
Results Summary for Read Operation(s)
Total Transferred: 0.250 MB
Total Throughput:  1.02 MB/s
Total Duration:    0.244 s
Number of Errors:  0
-----
Read times Max:      0.023 s
Read times 99th %ile: 0.021 s
Read times 90th %ile: 0.015 s
Read times 75th %ile: 0.009 s
Read times 50th %ile: 0.006 s
Read times 25th %ile: 0.004 s
Read times Min:      0.002 s

Cleaning up 256 objects...
Deleting a batch of 256 objects in range {0, 255}... Succeeded
Successfully deleted 256/256 objects in 1.3288742s
```

图 4

发现读取速率远大于写入速率。我们将 `objectSize`、`numClients` 和 `numSamples` 分别修改，得到以下表格：

Object size	Num clients	Num samples	写速度 MB/s	写时间 MB/s	读速度 MB/s	读时间 s
0.001MB	10	100	0.91	0.108s	1.59	0.062
20MB	10	100	128.40	0.152s	233.52	0.084
200MB	10	100	392.80	50.916s	659.84	30.310
20MB	20	100	471.09	2.245s	613.13	3.262
20MB	30	100	453.42	4.411s	561.41	3.562
20MB	10	200	420.58	9.511s	736.45	5.431
20MB	10	300	453.00	13.245s	697.78	8.599

在实验过程中，可以看到：

1. 写入读写未曾出错
2. 读取速率大于写入速率
3. 随着 `object size` 增大，读写速率也会增大
4. 随着 `client` 增大，速率先减后增。
5. 随着 `sample` 增大，逐渐增大。

六、实验总结

通过本次面向对象存储的入门实验，我对对象存储技术有了一定的了解，明白了在已有基于块和基于文件的存储系统的情况下，仍然需要面向对象存储系统的原因。在实验中，我在 Linux 环境下使用 minio 作为服务端，并且使用 s3-bench 进行了测试。尽管测试数据有所限制，但是总体而言速度和成功率都表现十分可观。另外，我也意识到通过简单的 minio 搭建便可以达成类似网盘的效果，这也是一个不小的惊喜。整个实验过程中，最大的难点在于配置环境。由于我的系统版本和大多数使用者不一致，导致 go 语言的安装包怎么都弄不好。后来是通过查阅资料，修改了配置之后才得以解决该问题。类似的，由于 s3-bench 的权限问题，我开始忘记分配权限，出现了一大堆奇怪的错误。之后在 root 权限下调用脚本，结果因为之前失败的进程没有杀死造成失败，最终才成功解决问题。这次实验让

我深刻地体会到了 Linux 系统下权限的重要性。总体来看，我从中受益匪浅，希望今后还能有更多体验类似的实验。

参考文献

- [1] WANG C, ZHANG L, SONG S, et al. Performance Evaluation of Object Storage Services: A Benchmarking Study[J]. IEEE Access, 2021, 9: 51224-51237.
- [2] SMITH D K. Object Storage Explained: Definition, Challenges and Opportunities[M]. Archway Publishing, 2018.
- [3] KARUNAKARAN M, RAJAVEL K S, RENUGADEVI M R. Comparative Study of Cloud Object Storage Systems[J]. Journal of Network and Computer Applications, 2017, 91: 1-17.
- [4] ZHANG X, ZHANG Y, ZHOU F, et al. Measurement and Analysis of Object Storage Workloads on Ceph[J]. Future Generation Computer Systems, 2016, 54: 173-186.
- [5] WU Y, WANG Y, CHEN H, et al. An Experimental Study of Cloud Object Storage Workloads[C]//Proceedings of the 2017 International Conference on Computing, Networking and Communications (ICNC). Santa Clara, CA, USA: IEEE, 2017: 373-377.