

**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](#_Toc509412095)

[二、实验背景 1](#_Toc509412096)

[三、实验环境 1](#_Toc509412097)

[四、实验内容 1](#_Toc509412098)

[4.1 对象存储技术实践 1](#_Toc509412099)

[4.2 对象存储性能分析 1](#_Toc509412100)

[五、实验过程 1](#_Toc509412101)

[六、实验总结 1](#_Toc509412102)

[参考文献 1](#_Toc509412103)

# 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。

# 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

文本

描述已自动生成MINIO\_ROOT\_USER=U201914992 MINIO\_ROOT\_PASSWORD=ba123 ./minio server ./data --console-address ":9099"。

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。

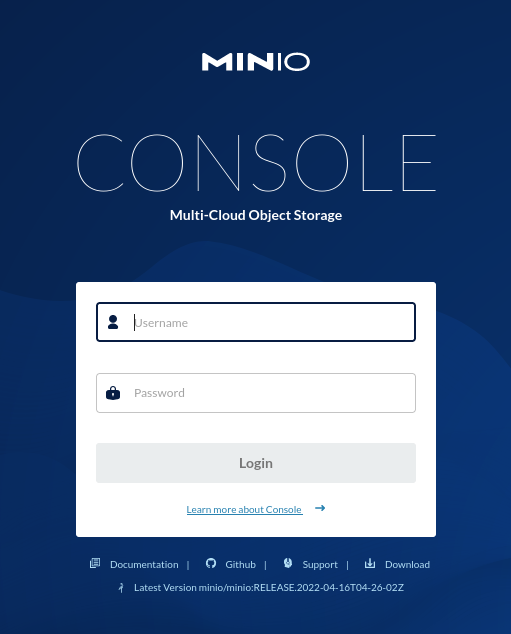
登录服务器

图 2

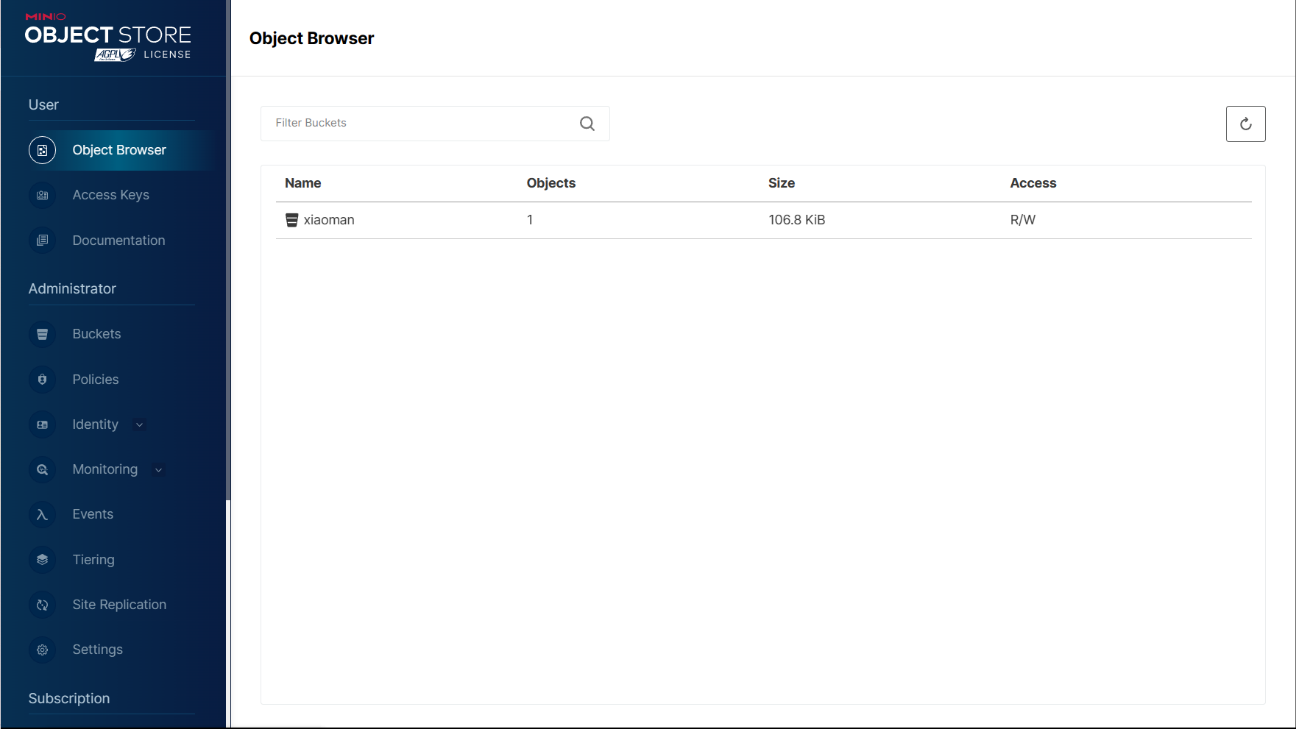


图 3

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，使用户名密码，IP和minio服务器一致。

文本

描述已自动生成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

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，我从中获益匪浅，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.