



2018 级

《物联网数据存储与管理》课程

实 验 报 告

姓 名 库尔夏提·亚森

学 号 U201714621

班 号 计算机 1804 班

日 期 2021.06.28

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.....	1
4.2 对象存储性能分析.....	1
五、实验过程.....	1
六、实验总结.....	8
参考文献.....	8

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

随着万维网的持续进化和物联网的持续发展，当今数据存储存在着巨大的挑战。一方面数据庞大，规模持续扩张；另一方面，数据的内容结构非常丰富。传统模式的存储技术效率低且管理复杂，而新产生的对象存储技术具有性能高、管理简单的特点。它具有存储“数据”和“属性”的特点，使它更可以满足未来的扩展操作。

Mock-S3，是用 Python 重写 fake-S3 实现的，沙盒环境中测试非常有用，无需实际调用 Amazon，其目标是最小化运行时依赖关系，并更像是一个开发工具来测试代码中的 S3 调用。

S3cmd 是一款免费的命令行工具和客户端，用于在 Amazon S3 和其他使用 S3 协议的云存储服务提供商中上传，检索和管理数据。

三、实验环境



设备名称	kurat-A320M-S2H >
内存	7.8 GiB
处理器	AMD® Ryzen 5 2600 six-core processor x 12
图形	NV168
磁盘容量	1.1 TB
系统名称	Ubuntu 20.04.2 LTS
操作系统类型	64 位
GNOME 版本	3.36.8
窗口系统	X11 窗口系统
软件更新	>

go 环境

```
kurxat@kurxat-A320M-S2H:~$ go version
go version go1.13.8 linux/amd64
```

四、实验内容

首先搭建实验环境，安装 go 语言环境。然后安装 minio 服务器端和客户端。最后运行 S3Bench 测试程序，分析各项指标。Go 环境安装版本如图 。

```
kurxat@kurxat-A320M-S2H:~$ go version
go version go1.13.8 linux/amd64
```

4.1 对象存储技术实践

1. 在本机上搭建一个 Minio 服务器端，将本机作为一个服务器。服务器存储空间即为本机的存储空间。
2. 在本机开启一个 Minio 客户端。可通过客户端对服务器内的文件进行操作。
3. 运行 S3Bench 来对其进行测评。

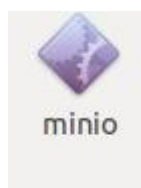
4.2 对象存储性能分析

对 Minio 服务器运行 S3Bench 进行测试。观察各项性能数据指标，修改测试代码，反复测试。

五、实验过程

5.1 minio 服务端的安装与运行

1. 下载 minio 得到可执行文件：



2. 运行 minio 服务端

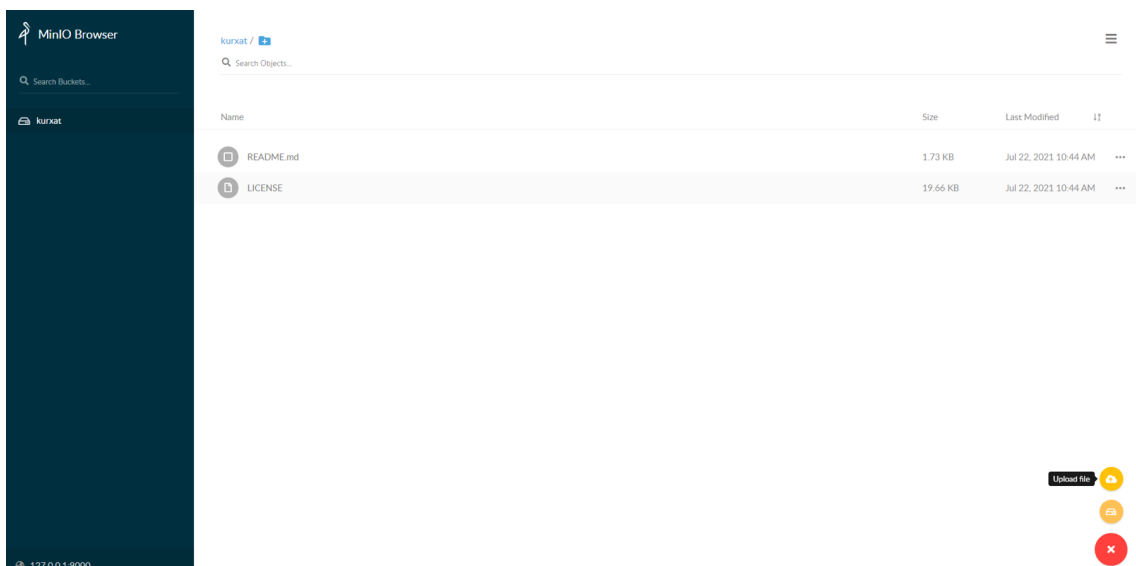
```
Endpoint: http://192.168.1.42:9000 http://127.0.0.1:9000
RootUser: minioadmin
RootPass: minioadmin

Browser Access:
http://192.168.1.42:9000 http://127.0.0.1:9000

Command-line Access: https://docs.min.io/docs/minio-client-quickstart-guide
$ mc.exe alias set myminio http://192.168.1.42:9000 minioadmin minioadmin

Object API (Amazon S3 compatible):
Go: https://docs.min.io/docs/golang-client-quickstart-guide
Java: https://docs.min.io/docs/java-client-quickstart-guide
Python: https://docs.min.io/docs/python-client-quickstart-guide
JavaScript: https://docs.min.io/docs/javascript-client-quickstart-guide
.NET: https://docs.min.io/docs/dotnet-client-quickstart-guide
Detected default credentials 'minioadmin:minioadmin', please change the credentials immediately by setting 'MINIO_ROOT_U
SER' and 'MINIO_ROOT_PASSWORD' environment values
```

3. 在浏览器输入 http: //127.0.0.1:900 进入页面



5.2minio 客户端的安装与测试

1. 官网下载 minio 客户端



2. 使用命令 ./mc--help 查看 mc 客户端可用参数

```

kurxat@kurxat-A320M-S2H:~$ ./mc --help
NAME:
  mc - MinIO Client for cloud storage and filesystems.

USAGE:
  mc [FLAGS] COMMAND [COMMAND FLAGS | -h] [ARGUMENTS...]

COMMANDS:
  ls      list buckets and objects
  mb      make a bucket
  rb      remove a bucket
  cat     display object contents
  head    display first 'n' lines of an object
  pipe    stream STDIN to an object
  share   generate URL for temporary access to an object
  cp      copy objects
  mirror  synchronize object(s) to a remote site
  find    search for objects
  sql     run sql queries on objects
  stat    show object metadata
  diff    list differences in object name, size, and date between two buckets
  rm      remove objects
  event   configure object notifications
  watch   listen for object notification events
  policy  manage anonymous access to buckets and objects
  admin   manage MinIO servers
  session resume interrupted operations
  config  configure MinIO client
  update  update mc to latest release
  version show version info

GLOBAL FLAGS:
  --config-dir value, -C value  path to configuration folder (default: "/home/aloha/.mc")
  --quiet, -q                  disable progress bar display
  --no-color                    disable color theme
  --json                        enable JSON formatted output

```

六、实验总结

通过这次的实验课程，让我学习到了对象存储技术的相关知识，并动手实践搭建了对象存储的服务端和客户端。

在这次实验的过程中，也遇到了一些问题，主要是在环境配置方面的，通过查阅网上的资料，最终还是得到了解决。虽然我的实验没有做完整但是在过程中学习到了很多，这些收获也会在今后的学习中对我有很大的帮助。

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.