



2019 级

《物联网数据存储与管理》课程

实 验 报 告

姓 名 焦嘉骏

学 号 U201915181

班 号 计算机 1908 班

日 期 2022.04.14

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.....	2
五、实验过程.....	2
六、实验总结.....	5
参考文献.....	5

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，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。

对象存储根本上改变了存储蓝图。它处理和解决了曾经被认为是棘手的存储问题：不间断可扩展性、弹性下降、限制数据持久性、无限技术更新和成本失控。

MinIO 是在 GNU Affero 通用公共许可证 v3.0 下发布的高性能对象存储。它是与 Amazon S3 云存储服务兼容的 API。使用 MinIO 为机器学习、分析和应用程序数据工作负载构建高性能基础架构。

三、实验环境

操作系统：windows11
Cpu i7-8565u
显卡：uhd620 mx250
客户端 mc
服务端 minio

四、实验内容

4.1 对象存储技术实践

实验步骤

1. 下载服务端 minio 并运行。
2. 下载客户端 mc 对 minio 服务器进行访问。
3. 载入 cosbench 进行测试并分析性能。

4.2对象存储性能分析

1. 观察 cosbench 的运行结果，分析吞吐率、延迟等性能参数。
2. 改变并发数量（works），再次测试分析现象。

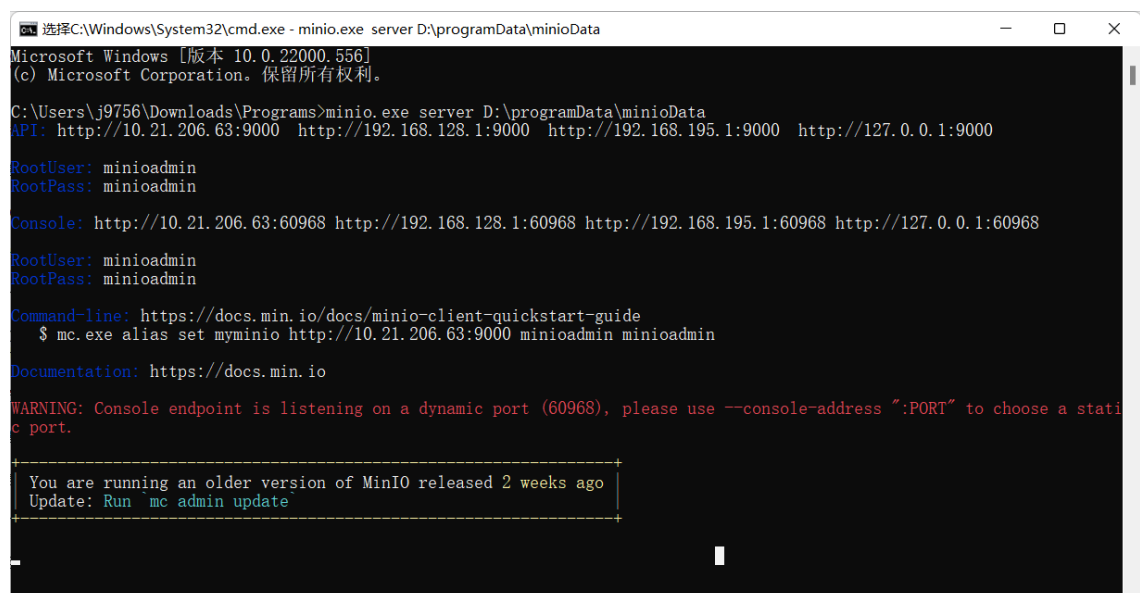
五、实验过程

5.1 对象存储技术实践

1. 下载 windows 版本 minio 客户端和服务端。
2. 运行 minio:

minio.exe server D:\programData\minioData

如图 1



```
选择C:\Windows\System32\cmd.exe - minio.exe server D:\programData\minioData
Microsoft Windows [版本 10.0.22000.556]
(c) Microsoft Corporation。保留所有权利。

C:\Users\j9756\Downloads\Programs>minio.exe server D:\programData\minioData
API: http://10.21.206.63:9000 http://192.168.128.1:9000 http://192.168.195.1:9000 http://127.0.0.1:9000

RootUser: minioadmin
RootPass: minioadmin

Console: http://10.21.206.63:60968 http://192.168.128.1:60968 http://192.168.195.1:60968 http://127.0.0.1:60968

RootUser: minioadmin
RootPass: minioadmin

Command-line: https://docs.min.io/docs/minio-client-quickstart-guide
$ mc.exe alias set myminio http://10.21.206.63:9000 minioadmin minioadmin

Documentation: https://docs.min.io

WARNING: Console endpoint is listening on a dynamic port (60968), please use --console-address ":PORT" to choose a static port.

You are running an older version of MinIO released 2 weeks ago
Update: Run mc admin update
```

图 1

3. 访问管理网站 127.0.0.1:9000,登陆后如图 2:

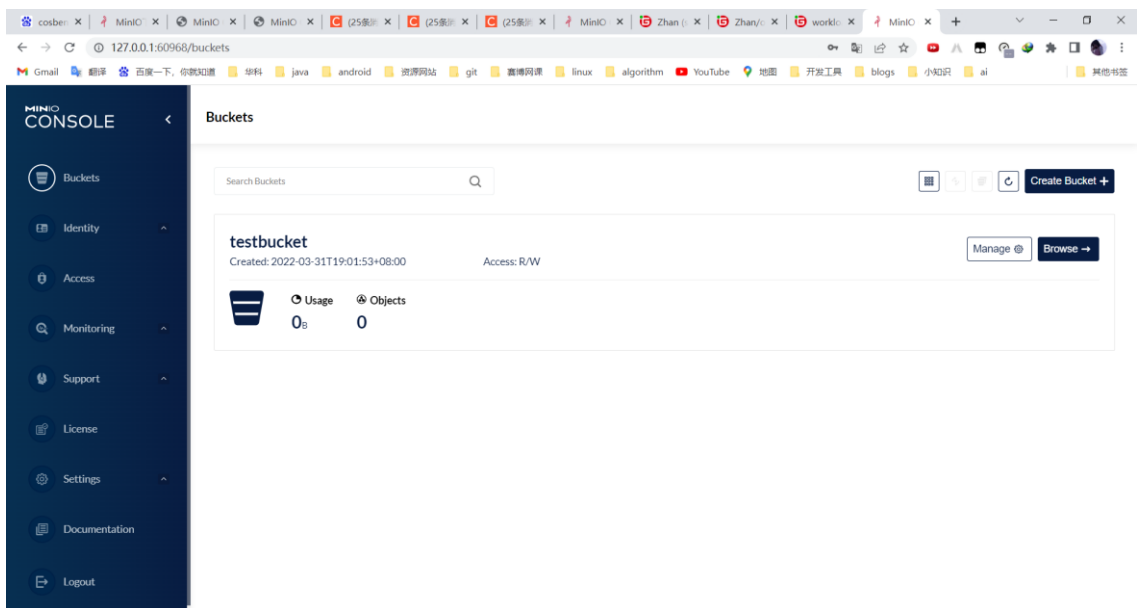


图 2

4. 创建 bucket 并测试

5.2 对象存储性能分析

1. 使用测试工具 cosbench，运行脚本启动程序 and 控制器，如图 3：

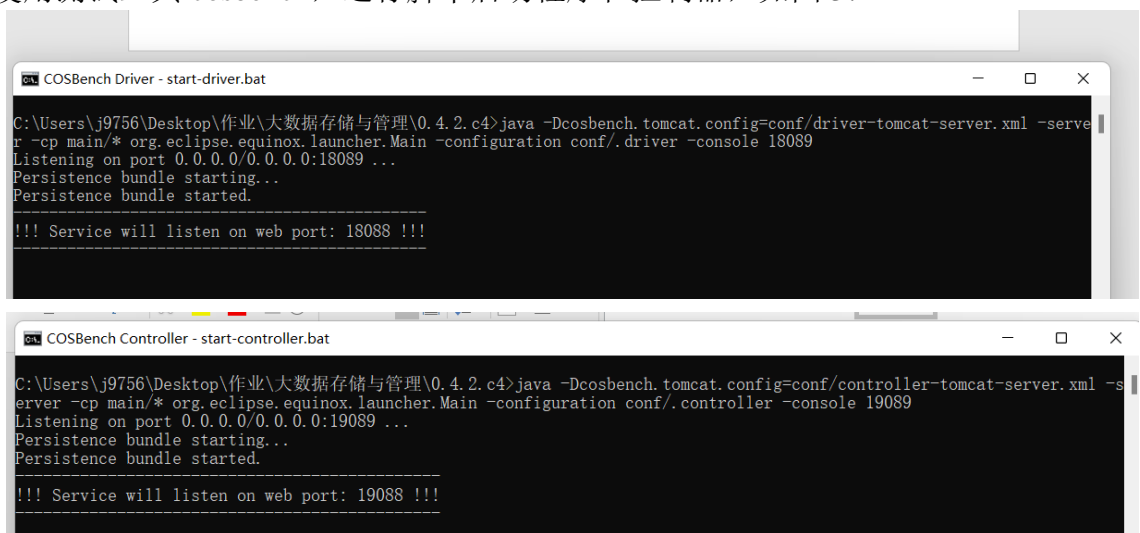


图 3

2. 浏览器访问（<http://127.0.0.1:19088/controller/index.html>）进入 cosbench 的测试，然后提交标准测试程序，对服务器进行模拟测试，部分测试图如图 4,测试结果如图 5：

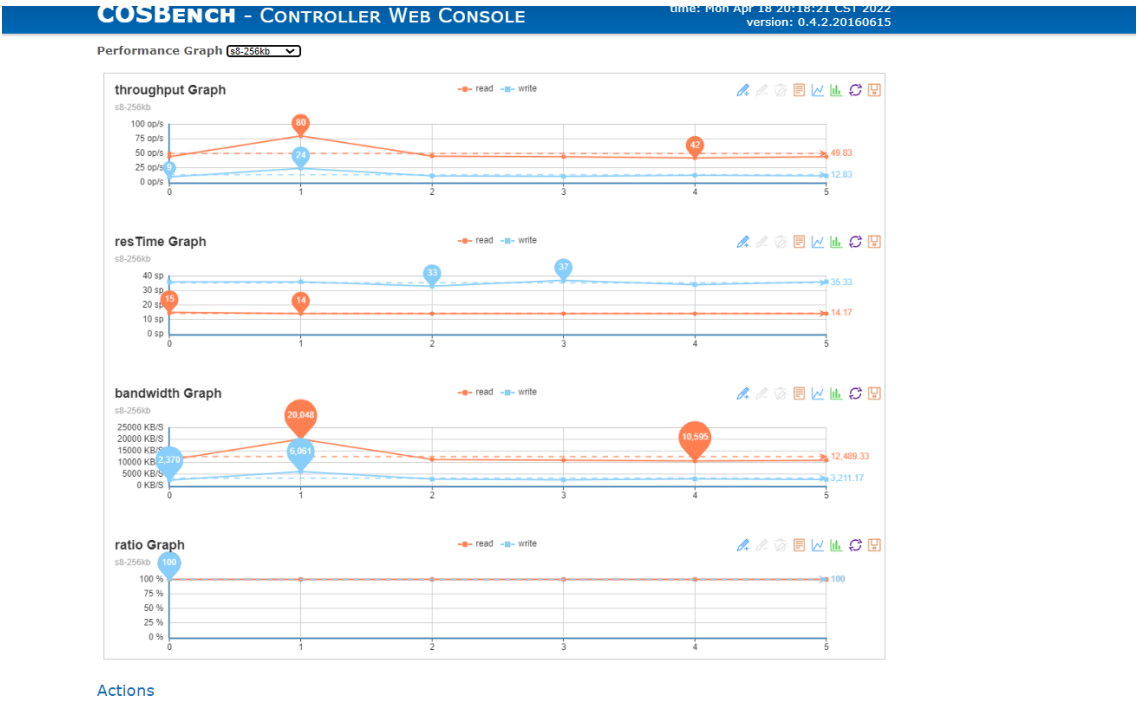


图 4

[show performance details](#)

Stages

Current Stage	Stages completed	Stages remaining	Start Time	End Time	Time Remaining	
ID	Name	Works	Workers	Op-Info	State	Link
w5-s1-init	init	1 wks	1 wkrs	init	completed	view details
w5-s2-prepare	prepare	8 wks	64 wkrs	prepare	completed	view details
w5-s3-8kb	8kb	1 wks	8 wkrs	read, write	completed	view details
w5-s4-16kb	16kb	1 wks	8 wkrs	read, write	completed	view details
w5-s5-32kb	32kb	1 wks	4 wkrs	read, write	completed	view details
w5-s6-64kb	64kb	1 wks	4 wkrs	read, write	completed	view details
w5-s7-128kb	128kb	1 wks	1 wkrs	read, write	completed	view details
w5-s8-256kb	256kb	1 wks	1 wkrs	read, write	completed	view details
w5-s9-512kb	512kb	1 wks	1 wkrs	read, write	completed	view details
w5-s10-1mb	1mb	1 wks	1 wkrs	read, write	completed	view details
w5-s11-cleanup	cleanup	1 wks	1 wkrs	cleanup	completed	view details
w5-s12-dispose	dispose	1 wks	1 wkrs	dispose	completed	view details

There are 12 stages in this workload.

[show error statistics details](#)

图 5

3.minio 性能分析，运行结果如图 6，可以看出 minio 写入成功率为 100%，而读取成功率相对较低：

Final Result

General Report

Op-Type	Op-Count	Byte-Count	Avg-ResTime	Avg-ProcTime	Throughput	Bandwidth	Succ-Ratio
op1: init -write	0 ops	0 B	N/A	N/A	0 op/s	0 B/S	N/A
op1: prepare -write	8 ops	64 KB	155.88 ms	155.75 ms	75.48 op/s	603.81 KB/S	100%
op2: prepare -write	8 ops	128 KB	213.88 ms	213.5 ms	40.86 op/s	653.74 KB/S	100%
op3: prepare -write	8 ops	256 KB	286 ms	285.88 ms	28.89 op/s	924.35 KB/S	100%
op4: prepare -write	8 ops	512 KB	317.38 ms	317.25 ms	26.01 op/s	1.66 MB/S	100%
op5: prepare -write	8 ops	1.02 MB	293.38 ms	292.88 ms	27.73 op/s	3.55 MB/S	100%
op6: prepare -write	8 ops	2.05 MB	120.5 ms	118.25 ms	79.05 op/s	20.24 MB/S	100%
op7: prepare -write	8 ops	4.1 MB	329.88 ms	327 ms	24.51 op/s	12.55 MB/S	100%
op8: prepare -write	8 ops	8 MB	343.75 ms	338.25 ms	23.65 op/s	23.65 MB/S	100%
op1: read	5.62 kops	44.99 MB	14.51 ms	14.44 ms	187.51 op/s	1.5 MB/S	62.54%
op2: write	2.35 kops	18.82 MB	46.08 ms	46.02 ms	78.45 op/s	627.62 KB/S	100%
op1: read	5.48 kops	87.63 MB	14.51 ms	14.39 ms	182.62 op/s	2.92 MB/S	59.36%
op2: write	2.3 kops	36.88 MB	46.01 ms	45.91 ms	76.85 op/s	1.23 MB/S	100%
op1: read	1.86 kops	59.36 MB	14.38 ms	14.17 ms	61.96 op/s	1.98 MB/S	34.05%
op2: write	1.35 kops	43.1 MB	30.31 ms	30.03 ms	45 op/s	1.44 MB/S	100%
op1: read	1.27 kops	81.15 MB	14.28 ms	13.96 ms	42.79 op/s	2.74 MB/S	26.03%
op2: write	1.24 kops	79.68 MB	38.74 ms	38.43 ms	42.02 op/s	2.69 MB/S	100%
op1: read	1.35 kops	173.06 MB	14.32 ms	13.62 ms	45.08 op/s	5.77 MB/S	100%
op2: write	359 ops	45.95 MB	29.48 ms	28.22 ms	11.97 op/s	1.53 MB/S	100%
op1: read	1.32 kops	338.18 MB	14.26 ms	13.03 ms	44.22 op/s	11.32 MB/S	100%
op2: write	330 ops	84.48 MB	33.35 ms	31.26 ms	11.05 op/s	2.83 MB/S	100%
op1: read	1.28 kops	655.36 MB	14.87 ms	12.54 ms	42.67 op/s	21.85 MB/S	100%
op2: write	298 ops	152.58 MB	36.71 ms	32.97 ms	9.93 op/s	5.09 MB/S	100%
op1: read	1.13 kops	1.13 GB	15.62 ms	11.52 ms	37.81 op/s	37.81 MB/S	100%
op2: write	286 ops	286 MB	42.84 ms	37 ms	9.54 op/s	9.54 MB/S	100%
op1: cleanup -delete	128 ops	0 B	16.41 ms	16.41 ms	60.84 op/s	0 B/S	100%
op1: dispose -delete	0 ops	0 B	N/A	N/A	0 op/s	0 B/S	N/A

图 6

由运行结果可知并发数量越多，读性能逐渐下降，同时吞吐率下降，但读写速度会加快。

六、实验总结

1. 首先，在 gitee 上首次实现了人生中国的第一个 pull request，体会到了和大佬合作的感觉，为以后进企业工作或者做团队大型项目打下基础。

2. 了解了什么是大数据存储，以及个人用户使用的小型对象存储管理系统，对企业的大型系统也有了初步的一些认识。

3. 通过测试，实现了 minio 性能的考察，获益匪浅。

参考文献

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

- [4] Dean J, Barroso L A. Association for Computing Machinery, 2013. The Tail at Scale[J]. Commun. ACM, 2013, 56(2): 74 – 80.
- [5] Delimitrou C, Kozyrakis C. Association for Computing Machinery, 2018. Amdahl’s Law for Tail Latency[J]. Commun. ACM, 2018, 61(8): 65 – 72.
- (可以根据实际需要更新调整)