2019 级

《物联网数据存储与管理》课程

实 验 报 告

姓　名　李嘉骏

学　号　U201915017

班　号　物联网 1901 班

日　期　2023.5.29

目　　录

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

本次实验为对象存储入门实验，包括三个部分：服务器端的准备，客户端的准备以及测评工具的使用。

我使用的服务端是 Minio，是一个基于 Apache License v2.0 开源协议的对象存储服务。它兼容亚马逊 S3 云存储服务接口，非常适合于存储大容量非结构化的数据，例如图片、视频、日志文件、备份数据和容器/虚拟机镜像等，而一个对象文件可以是任意大小，从几 kb 到最大 5T 不等。

Minio 是一个非常轻量的服务,可以很简单的和其他应用的结合，类似 NodeJS, Redis 或者 MySQL。

使用的对象存储客户端也是 Minio 的客户端部分。

对于对象存储测评工具我选择了 COSbench，COSBench 是一个用于测试云对象存储系统的分布式基准测试工具，也允许用户为额外的存储系统创建适配器。

由两个主要组件组成

Driver（Load Generator）：

负责生成工作负载，向目标云对象存储发出操作；性能统计

Controller：

负责协调 drivers 集体执行工作，收集和汇总聚合来自 driver 实例的运行时状态或基准测试结果

三、实验环境

处理器	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.20 GHz
机带 RAM	24.0GB
操作系统	Windows10 操作系统
系统类型	64 位操作系统，基于 x64 的处理器
Java 版本	jre1.8.0_192
Python 版本	Python 3.9
服务器端	minio
客户端	minio
测评工具	COSBench

四、实验内容

4.1 对象存储技术实践

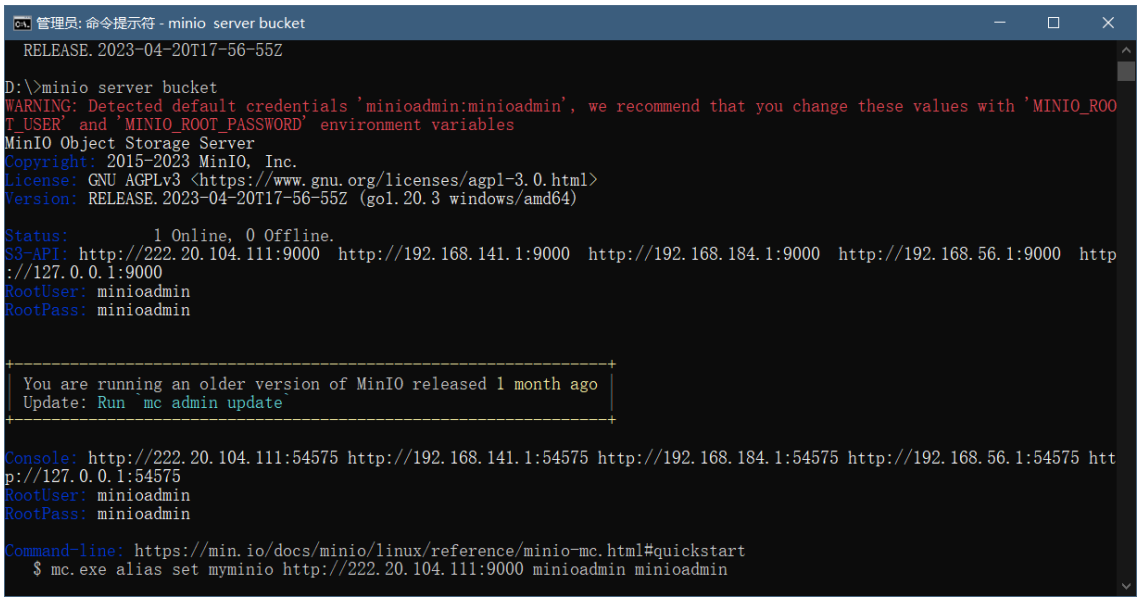
1. 在 Windows 环境下下载配置 minio 服务器；
2. 下载 mc 客户端，创建 bucket，上传文件；
3. 安装使用 cosbench 提交 xml 文件测试。

4.2 对象存储性能分析

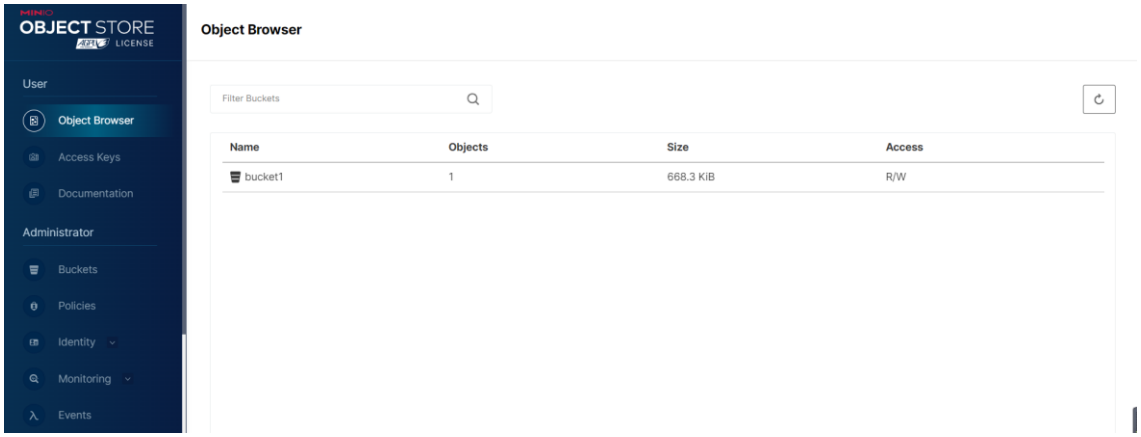
1. 测试读写性能的优劣。
2. 测试 work 值对各项指标的影响，包含吞吐率，带宽等。
3. 测试不同块大小对各项指标的影响。

五、实验过程

5.1 配置 minio server



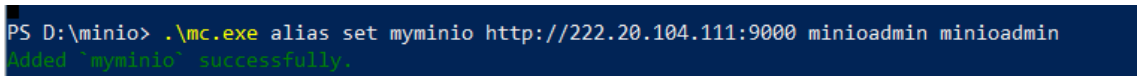
根据显示的内容可以在浏览器中打开 http://220.20.104.111:9000，这是服务器的可视化管理界面，输入 AccessKey 和 Sercretkey（两个都是 minioadmin）登录。



此时可以创建桶，并且存入文件。

5.2 配置 minio client

下载 minio 的 mc 客户端，通过命令行添加服务器 myminio，可以用 ls 指令列出存在的服务端。

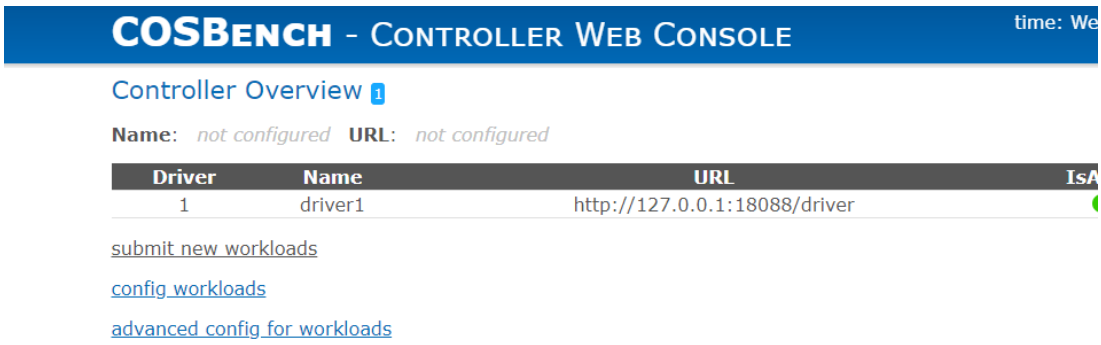


```
PS D:\minio> mc config host ls
gcs
  URL       : https://storage.googleapis.com
  AccessKey : YOUR-ACCESS-KEY-HERE
  SecretKey : YOUR-SECRET-KEY-HERE
  API       : S3v2
  Path      : dns

local
  URL       : http://localhost:9000
  AccessKey :
  SecretKey :
  API       :
  Path      : auto

myminio
  URL       : http://222.20.104.111:9000
  AccessKey : minioadmin
  SecretKey : minioadmin
  API       : s3v4
  Path      : auto
```

可以看到我们的 myminio 已经显示出来了。

5.3 使用 COSBench 测试

打开文件夹中的 start-all.bat 可以直接同时打开 controller 和 driver。
然后打开 web.bat 即可进入 cosbench 的测试页面



点击 submit new workloads 就可以上传测试文件进行测试。
结果如下

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	2877 ms	2876.75 ms	2.79 op/s	22.3 KB/S	100%
op2: prepare - write	8 ops	128 KB	2933.88 ms	2933.62 ms	2.73 op/s	43.67 KB/S	100%
op3: prepare - write	8 ops	256 KB	2921.12 ms	2921 ms	2.75 op/s	88.06 KB/S	100%
op4: prepare - write	8 ops	512 KB	1764.75 ms	1759.62 ms	5.36 op/s	342.88 KB/S	100%
op5: prepare - write	8 ops	1.02 MB	2959.75 ms	2958.38 ms	2.69 op/s	344.51 KB/S	100%
op6: prepare - write	8 ops	2.05 MB	2449.12 ms	2443.88 ms	3.37 op/s	863.44 KB/S	100%
op7: prepare - write	8 ops	4.1 MB	2966.62 ms	2961.62 ms	2.7 op/s	1.38 MB/S	100%
op8: prepare - write	8 ops	8 MB	2857.38 ms	2728.25 ms	2.81 op/s	2.81 MB/S	100%
op1: read	1.98 kops	15.83 MB	6.92 ms	6.79 ms	66.3 op/s	530.39 KB/S	97.83%
op2: write	473 ops	3.78 MB	474.64 ms	474.63 ms	15.85 op/s	126.79 KB/S	100%
op1: read	2.62 kops	41.94 MB	5.65 ms	5.47 ms	88.4 op/s	1.41 MB/S	97.54%
op2: write	637 ops	10.19 MB	348.37 ms	348.25 ms	21.5 op/s	343.99 KB/S	100%
op1: read	2.23 kops	71.33 MB	5.5 ms	5.17 ms	74.56 op/s	2.39 MB/S	95.62%
op2: write	608 ops	19.46 MB	175.37 ms	175.01 ms	20.34 op/s	650.83 KB/S	100%
op1: read	2.3 kops	147.07 MB	5.98 ms	5.27 ms	77 op/s	4.93 MB/S	92.66%
op2: write	610 ops	39.04 MB	170.78 ms	169.67 ms	20.44 op/s	1.31 MB/S	100%
op1: read	940 ops	120.32 MB	9.46 ms	7.75 ms	31.33 op/s	4.01 MB/S	100%
op2: write	229 ops	29.31 MB	91.86 ms	89.16 ms	7.63 op/s	977.07 KB/S	100%

将 work-example 中的块大小保持 128k 不变，workers 的数量从 1 依次翻倍到 128，保存并重新提交，结果如下

op1: read	1.28 kops	163.2 MB	9.67 ms	7.94 ms	42.56 op/s	5.45 MB/S	100%
op2: write	307 ops	39.3 MB	57.07 ms	53.16 ms	10.25 op/s	1.31 MB/S	100%
op1: read	2.14 kops	273.41 MB	9 ms	7.41 ms	71.26 op/s	9.12 MB/S	100%
op2: write	505 ops	64.64 MB	80.31 ms	76.32 ms	16.85 op/s	2.16 MB/S	100%
op1: read	2.89 kops	370.18 MB	9.31 ms	7.73 ms	96.67 op/s	12.37 MB/S	100%
op2: write	705 ops	90.24 MB	131.25 ms	127.66 ms	23.56 op/s	3.02 MB/S	100%
op1: read	3.08 kops	394.75 MB	10.13 ms	8.66 ms	103.66 op/s	13.27 MB/S	100%
op2: write	761 ops	97.41 MB	271.4 ms	267.78 ms	25.58 op/s	3.27 MB/S	100%
op1: read	3.11 kops	397.57 MB	12.84 ms	11.24 ms	104.64 op/s	13.39 MB/S	100%
op2: write	735 ops	94.08 MB	591.15 ms	587.53 ms	24.79 op/s	3.17 MB/S	100%
op1: read	3.48 kops	445.31 MB	23.1 ms	21.54 ms	121.36 op/s	15.53 MB/S	100%
op2: write	808 ops	103.42 MB	1038 ms	1034.39 ms	28.21 op/s	3.61 MB/S	100%
op1: read	3.72 kops	476.29 MB	16.24 ms	14.77 ms	141.52 op/s	18.11 MB/S	100%
op2: write	828 ops	105.98 MB	1972.86 ms	1969.66 ms	31.5 op/s	4.03 MB/S	100%
op1: read	4.25 kops	543.62 MB	37.11 ms	35.56 ms	165.42 op/s	21.17 MB/S	100%
op2: write	890 ops	113.92 MB	3619.2 ms	3616.02 ms	34.19 op/s	4.38 MB/S	100%

可以看出，随着 worker 增加，吞吐量、时延和带宽都在增加，但是增速的变化很小，说明 worker 数量还没有达到上限，性能也随着 worker 的数量增加而提升，如果达到了饱和状态，那么服务器就会出现性能降低的情况。

op1: read	2.96 kops	23.7 MB	7.81 ms	7.68 ms	98.91 op/s	791.3 KB/S	100%
op2: write	743 ops	5.94 MB	129.75 ms	129.71 ms	24.8 op/s	198.43 KB/S	100%
op1: read	2.83 kops	45.22 MB	8.27 ms	8.1 ms	94.41 op/s	1.51 MB/S	100%
op2: write	723 ops	11.57 MB	132.7 ms	132.32 ms	24.15 op/s	386.45 KB/S	100%
op1: read	2.92 kops	93.44 MB	7.68 ms	7.48 ms	97.64 op/s	3.12 MB/S	100%
op2: write	732 ops	23.42 MB	132.35 ms	131.3 ms	24.48 op/s	783.33 KB/S	100%
op1: read	2.99 kops	191.42 MB	8.42 ms	7.78 ms	99.92 op/s	6.39 MB/S	100%
op2: write	729 ops	46.66 MB	129.32 ms	127.12 ms	24.35 op/s	1.56 MB/S	100%
op1: read	2.67 kops	341.63 MB	9.99 ms	8.26 ms	89.23 op/s	11.42 MB/S	100%
op2: write	655 ops	83.84 MB	141.61 ms	137.88 ms	21.9 op/s	2.8 MB/S	100%
op1: read	2.22 kops	567.04 MB	10.68 ms	7.47 ms	74.48 op/s	19.07 MB/S	100%
op2: write	544 ops	139.26 MB	174.91 ms	167.47 ms	18.29 op/s	4.68 MB/S	100%
op1: read	1.87 kops	958.98 MB	13.97 ms	7.67 ms	62.5 op/s	32 MB/S	100%
op2: write	449 ops	229.89 MB	208.35 ms	194.16 ms	14.98 op/s	7.67 MB/S	100%
op1: read	1.13 kops	1.13 GB	21.59 ms	9.09 ms	37.96 op/s	37.96 MB/S	100%
op2: write	298 ops	298 MB	317.79 ms	285.89 ms	10 op/s	10 MB/S	100%

如果 worker 的数量不变，块大小从 8kb 依次翻倍到 1Mb，运行结果如下

可以看出，随着块增大，读写的时延有增大趋势，并且增加速度越来越多。带宽一直增加，吞吐量一开始稳定在一个值左右，当超出处理能力后快速下降。

可以得出，当块较小时，服务器处理的能力和带宽相对充足，因此吞吐率基本不变，时延增加不多，而带宽值几乎也是翻倍地增加，单当块增加到一定程度，服务器能力就开始饱和甚至不足，吞吐率就严重下降。

六、实验总结

在本次实验中，我搭建了简单的分布式对象存储系统 minio，并通过工具对 性能进行了测试分析。Minio 小巧方便，容易上手。

在比较了各种测试工具之后，因为我不太想搭建虚拟机，而大多数测试工具是必须在 Linux 系统中使用，最终选择了 COSbench 这个工具。最开始下载的 0.4.2 版本一直在报错，经过上网查询之后，改用了 0.4.2-c4 版本进行实验，最终也是测试成功。

通过自己搭建对象存储系统，我对大数据存储管理的原理和概念的理解更加深刻了，另一方面，这门课补充了在大学期间缺失的 Git 技能，我受益匪浅。

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）