



《数据中心技术》课程

实 验 报 告

姓 名 尹宇枫

学 号 M202173853

院 系 计算机科学与技术

日 期 2022 年 1 月 7 日

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

随着互联网的不断扩张和云计算技术的进一步推广，海量的数据在个人、企业、研究机构等源源不断地产生，如何有效、快速、可靠地存取这些日益增长的海量数据成了关键的问题。

传统的存储解决方案面临越来越多的困难，比如数据量的指数级增长对不断扩容的存储空间提出要求、实时分析海量的数据对存储计算能力提出要求等。因此，需要能处理海量数据、高性能、易扩展、可伸缩、高可用的新型存储方案。对象存储系统(Object-Based Storage System)是综合了 NAS 和 SAN 的优点，同时具有 SAN 的高速直接访问和 NAS 的数据共享等优势，提供了高可靠性、跨平台性以及安全的数据共享的存储体系结构。

三. 实验环境

实验环境	信息
硬件环境	处理器: Intel® Core™ i5-7300HQ CPU @ 2.50GHz × 4
	内存: 4GB
软件环境	OS: Ubuntu 21.04
	Virtualization: VMware
	Python: Python 3.9.5
	GO: go version go1.11.5 linux/amd64
	服务器端: mock-s3
	客户端: s3cmd
	测试工具: s3bench

四. 实验内容

4.1 对象存储技术实践

1. 服务器端采用 Minio、客户端采用 s3cmd。对 Minio 下对象存储等概念及实现有了一定初步的认识，并通过 s3cmd 实现了 create、read、update、and delete

- (缩写为 CRUD) 等基本操作;
2. 借助 S3 Bench, 对对象存储的相关性能进行了观测;

4.2 对象存储性能分析

1. 对吞吐率 Throughput、延迟 Latency、对象尺寸 object size 等指标进行测试;
2. 尾延迟相关思考。

五. 实验过程

1. 安装并配置 Java、Python、Go
2. 安装 mock-s3
3. 安装 s3cmd

```
HTTP Proxy server name:

New settings:
  Access Key: hust
  Secret Key: hust_obs
  Default Region: US
  S3 Endpoint: 127.0.0.1:9000
  DNS-style bucket+hostname:port template for accessing a bucket: 9000
  Encryption password: hust_obs
  Path to GPG program: /usr/bin/gpg
  Use HTTPS protocol: False
  HTTP Proxy server name:
  HTTP Proxy server port: 0

Test access with supplied credentials? [Y/n] Y
Please wait, attempting to list all buckets...
WARNING: Could not refresh role
ERROR: Test failed: [Errno 111] Connection refused

Retry configuration? [Y/n] n
```

4. 测试服务端与客户端

```
s3cmd mb test0bucket #创建新bucket
s3cmd put ~/demo.txt s3://test0_bucket #上传测试
```

5. 安装 s3bench 进行性能观测
测试结果:

```

Test parameters
endpoint(s):      [http://127.0.0.1:9000]
bucket:           loadgen
objectNamePrefix: loadgen
objectSize:       0.0312 MB
numClients:       8
numSamples:       256
verbose:          %!d(bool=false)

Results Summary for Write Operation(s)
Total Transferred: 8.000 MB
Total Throughput:  6.98 MB/s
Total Duration:    1.147 s
Number of Errors:  0
-----
Write times Max:      1.029 s
Write times 99th %ile: 0.151 s
Write times 90th %ile: 0.042 s
Write times 75th %ile: 0.018 s
Write times 50th %ile: 0.012 s
Write times 25th %ile: 0.011 s
Write times Min:      0.003 s

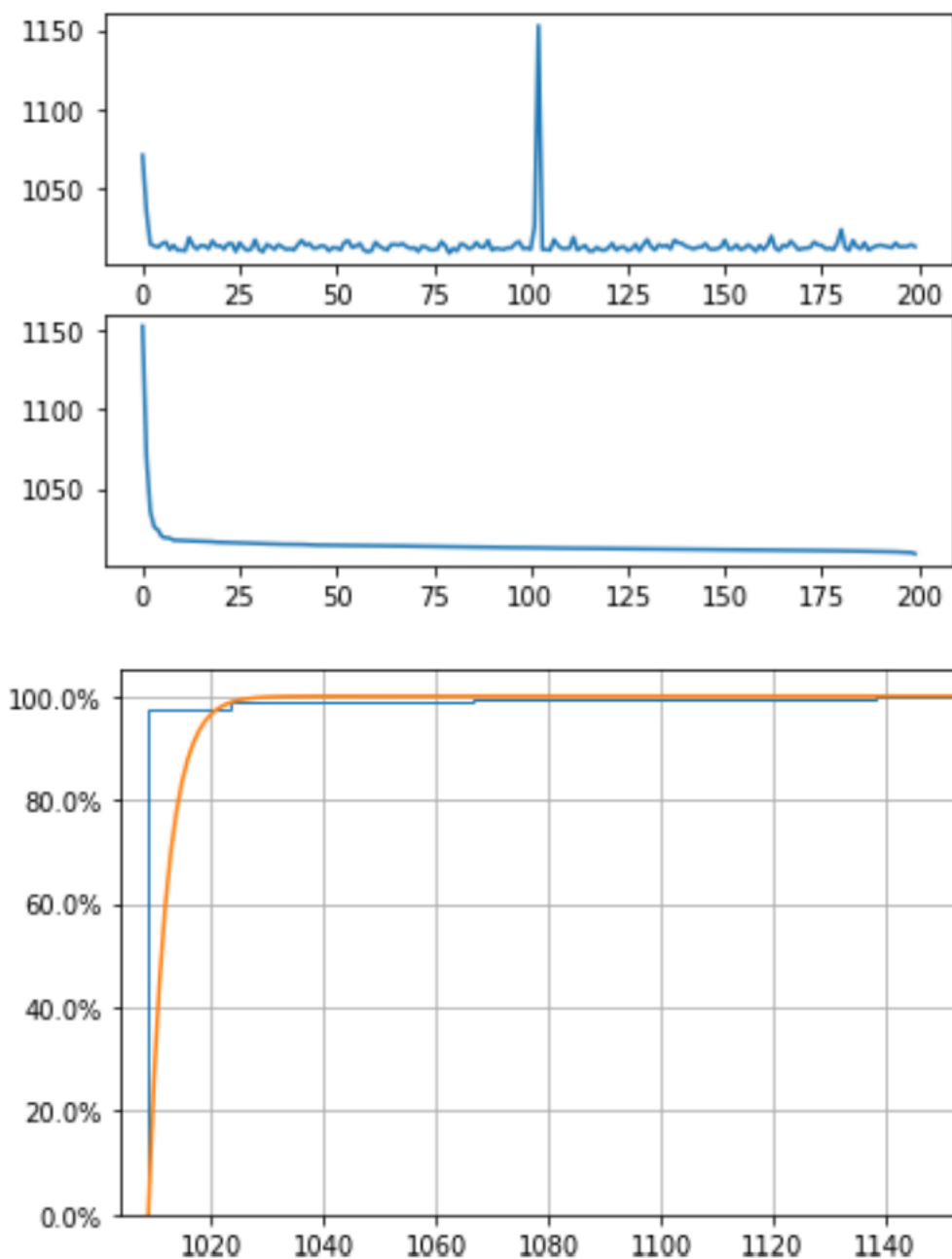
Results Summary for Read Operation(s)
Total Transferred: 8.000 MB
Total Throughput:  7.54 MB/s
Total Duration:    1.060 s
Number of Errors:  0
-----
Read times Max:       1.033 s
Read times 99th %ile: 0.108 s
Read times 90th %ile: 0.042 s
Read times 75th %ile: 0.018 s
Read times 50th %ile: 0.011 s
Read times 25th %ile: 0.010 s
Read times Min:       0.007 s

Cleaning up 256 objects...
Deleting a batch of 256 objects in range {0, 255}... Succeeded
Successfully deleted 256/256 objects in 60.027437ms

```

6. 尾延迟挑战

请求延迟分布情况与排队论模型拟合实测数据如下：



六. 实验总结

本次实验使我对对象存储有了初步的认识。实验的环境配置部分让我熟悉Linux上的一些操作。后面的对象存储性能分析、尾延迟处理由浅入深逐渐将我引入对象存储这个技术。这个实验让我对存储领域的一些知识有了基本的概念，开拓了在计算机上的视野。总体来说，十分有收获。