

華中科技大學

实验报告

课程名称： 数据中心技术

专业班级： 计算机硕 2108 班

学 号： M202173793

姓 名： 余志伟

指导教师： 施展

报告日期： 2022 年 1 月 6 日

计算机科学与技术学院

实验一

对象存储服务器选择 MinIO，选择 MinIO 的 Python SDK 与服务器交互，测试对象存储的基本功能。

在 MinIO 官网下载 minio.exe，再运行老师提供的 minio.cmd 脚本，即可启动 MinIO 的对象存储服务器。

```
API: http://10.12.57.44:9000 http://192.168.110.1:9000 http://192.168.12.1:9000 http://127.0.0.1:9000
RootUser: hust
RootPass: hust_obs

Console: http://10.12.57.44:9090 http://192.168.110.1:9090 http://192.168.12.1:9090 http://127.0.0.1:9090
RootUser: hust
RootPass: hust_obs

Command-line: https://docs.min.io/docs/minio-client-quickstart-guide
$ mc.exe alias set myminio http://10.12.57.44:9000 hust hust_obs

Documentation: https://docs.min.io
```

在浏览器中打开 <http://127.0.0.1:9000>，使用 hust 和 hust_obs 登录到 MinIO 的用户界面。











python 可用`pip3 install minio`安装 MinIO 的 SDK，就可连接到服务器，代码如下：

```
1. client = Minio(
2.     "127.0.0.1:9000",
3.     access_key="hust",
4.     secret_key="hust_obs",
5.     secure=False
6. )
```

create: 在对象存储服务器中新建一个 bucket，并向该 bucket 中添加一段文本和一张图片。

```
1. # create
2. if not client.bucket_exists('yzw'):
3.     client.make_bucket('yzw')
4. # upload objects
5. client.fput_object('yzw', '念奴娇·赤壁怀古.txt', '念奴娇·赤壁怀古.txt')
6. client.fput_object('yzw', '向日葵.jpg', '向日葵.jpg')
```

执行代码，对象上传到服务器。

Select	Name	Last Modified	Size	Options
<input type="checkbox"/>	 念奴娇·赤壁怀古.txt	Thu Jan 06 2022 14:18:50 GMT+0800	432 B	   
<input type="checkbox"/>	 向日葵.jpg	Thu Jan 06 2022 14:18:50 GMT+0800	56 KB	   

read: 从对象服务器上下载对象。

```
1. # read
2. client.fget_object('yzw', '念奴娇·赤壁怀古.txt', 'dl.txt')
3. client.fget_object('yzw', '向日葵.jpg', 'dl.jpg')
```

update: 将修改后的对象，上传到对象服务器相同的 bucket，覆盖原来的对象，完成对象的更新。

```
1. # update
2. f = open('念奴娇·赤壁怀古.txt', 'a', encoding='utf-8')
3. f.write('\n 滚滚长江东逝水 \n')
4. f.close()
5. client.fput_object('yzw', '念奴娇·赤壁怀古.txt', '念奴娇·赤壁怀古.txt')
```

delete: 删除完 bucket 中的对象后再删除 bucket。

```
1. # delete
2. client.remove_object('yzw', '念奴娇·赤壁怀古.txt')
3. client.remove_object('yzw', '向日葵.jpg')
4. client.remove_bucket('yzw')
```

实验二

使用 S3 Bench 对 MinIO 对象存储服务器的性能进行测试。

首先在 MinIO 中新建名为 loadgen 的 bucket，再调节脚本中的 S3 Bench 的命令参数进行测试。

保持 numClients=8，numSamples=1024 不变，objectSize 逐渐增加 1024 2048 4096 8192 32768，结果如下图：

<pre> Results Summary for Write Operation(s) Total Transferred: 1.000 MB Total Throughput: 0.15 MB/s Total Duration: 6.578 s Number of Errors: 0 Write times Max: 0.110 s Write times 99th %ile: 0.097 s Write times 90th %ile: 0.077 s Write times 75th %ile: 0.063 s Write times 50th %ile: 0.048 s Write times 25th %ile: 0.043 s Write times Min: 0.004 s Results Summary for Read Operation(s) Total Transferred: 1.000 MB Total Throughput: 0.21 MB/s Total Duration: 0.109 s Number of Errors: 0 Read times Max: 0.003 s Read times 99th %ile: 0.002 s Read times 90th %ile: 0.001 s Read times 75th %ile: 0.001 s Read times 50th %ile: 0.001 s Read times 25th %ile: 0.001 s Read times Min: 0.001 s Cleaning up 1024 objects... Deleting a batch of 1000 objects in range Deleting a batch of 24 objects in range Successfully deleted 1024/1024 objects </pre>	<pre> Results Summary for Write Operation(s) Total Transferred: 2.000 MB Total Throughput: 0.33 MB/s Total Duration: 6.140 s Number of Errors: 0 Write times Max: 0.108 s Write times 99th %ile: 0.092 s Write times 90th %ile: 0.075 s Write times 75th %ile: 0.061 s Write times 50th %ile: 0.047 s Write times 25th %ile: 0.032 s Write times Min: 0.004 s Results Summary for Read Operation(s) Total Transferred: 2.000 MB Total Throughput: 0.111 s Total Duration: 0.111 s Number of Errors: 0 Read times Max: 0.003 s Read times 99th %ile: 0.002 s Read times 90th %ile: 0.001 s Read times 75th %ile: 0.001 s Read times 50th %ile: 0.001 s Read times 25th %ile: 0.001 s Read times Min: 0.000 s Cleaning up 1024 objects... Deleting a batch of 1000 objects in range Deleting a batch of 24 objects in range Successfully deleted 1024/1024 objects </pre>	<pre> Results Summary for Write Operation(s) Total Transferred: 4.000 MB Total Throughput: 0.61 MB/s Total Duration: 6.505 s Number of Errors: 0 Write times Max: 0.107 s Write times 99th %ile: 0.092 s Write times 90th %ile: 0.075 s Write times 75th %ile: 0.061 s Write times 50th %ile: 0.047 s Write times 25th %ile: 0.045 s Write times Min: 0.003 s Results Summary for Read Operation(s) Total Transferred: 4.000 MB Total Throughput: 0.107 s Total Duration: 0.107 s Number of Errors: 0 Read times Max: 0.003 s Read times 99th %ile: 0.002 s Read times 90th %ile: 0.001 s Read times 75th %ile: 0.001 s Read times 50th %ile: 0.001 s Read times 25th %ile: 0.001 s Read times Min: 0.000 s Cleaning up 1024 objects... Deleting a batch of 1000 objects in range Deleting a batch of 24 objects in range Successfully deleted 1024/1024 objects </pre>	<pre> Results Summary for Write Operation(s) Total Transferred: 8.000 MB Total Throughput: 1.28 MB/s Total Duration: 6.263 s Number of Errors: 0 Write times Max: 0.122 s Write times 99th %ile: 0.093 s Write times 90th %ile: 0.077 s Write times 75th %ile: 0.062 s Write times 50th %ile: 0.047 s Write times 25th %ile: 0.033 s Write times Min: 0.003 s Results Summary for Read Operation(s) Total Transferred: 8.000 MB Total Throughput: 0.104 s Total Duration: 0.104 s Number of Errors: 0 Read times Max: 0.003 s Read times 99th %ile: 0.002 s Read times 90th %ile: 0.001 s Read times 75th %ile: 0.001 s Read times 50th %ile: 0.001 s Read times 25th %ile: 0.001 s Read times Min: 0.001 s Cleaning up 1024 objects... Deleting a batch of 1000 objects in range Deleting a batch of 24 objects in range Successfully deleted 1024/1024 objects </pre>	<pre> Results Summary for Write Operation(s) Total Transferred: 32.000 MB Total Throughput: 4.85 MB/s Total Duration: 6.603 s Number of Errors: 0 Write times Max: 0.108 s Write times 99th %ile: 0.092 s Write times 90th %ile: 0.076 s Write times 75th %ile: 0.061 s Write times 50th %ile: 0.047 s Write times 25th %ile: 0.045 s Write times Min: 0.004 s Results Summary for Read Operation(s) Total Transferred: 32.000 MB Total Throughput: 0.109 s Total Duration: 0.109 s Number of Errors: 0 Read times Max: 0.003 s Read times 99th %ile: 0.002 s Read times 90th %ile: 0.001 s Read times 75th %ile: 0.001 s Read times 50th %ile: 0.001 s Read times 25th %ile: 0.001 s Read times Min: 0.001 s Cleaning up 1024 objects... Deleting a batch of 1000 objects in range Deleting a batch of 24 objects in range Successfully deleted 1024/1024 objects </pre>
---	---	---	---	---

从以上数据看出，objectSize 对系统读写性能的影响不大，可能是都比较小的缘故。

写数据的耗费远大于读耗费。

随着块大小的增长，读写尾延迟均越来越明显。

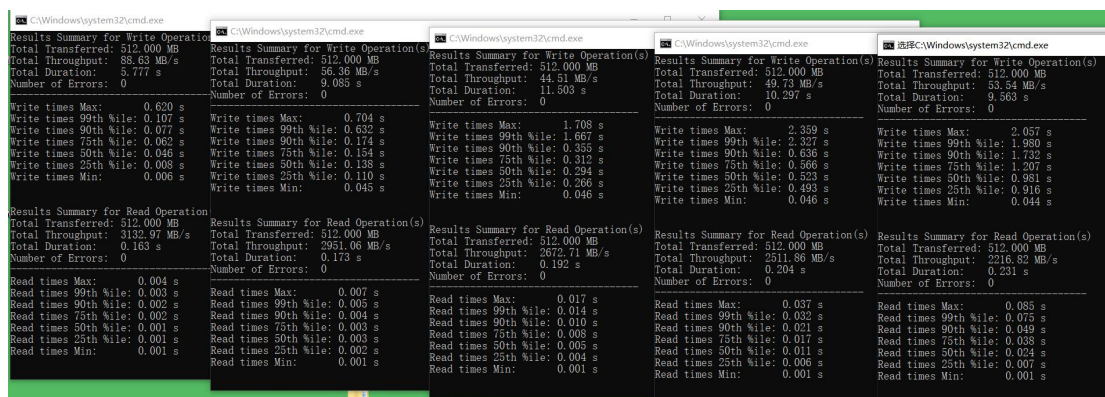
传输速度出现自适应增长，但到一定程度时，速度无法再自适应。

保持 numClients=8，objectSize=524288 (512KB) 样本数量 numSamples 逐渐增加 1024 2048 4096 8192 32768，结果如下图：

<pre> Results Summary for Write Operation(s) Total Transferred: 512.000 MB Total Throughput: 69.32 MB/s Total Duration: 7.386 s Number of Errors: 0 Write times Max: 1.860 s Write times 99th %ile: 0.584 s Write times 90th %ile: 0.073 s Write times 75th %ile: 0.063 s Write times 50th %ile: 0.046 s Write times 25th %ile: 0.014 s Write times Min: 0.006 s Results Summary for Read Operation(s) Total Transferred: 512.000 MB Total Throughput: 3230.99 MB/s Total Duration: 0.158 s Number of Errors: 0 Read times Max: 0.003 s Read times 99th %ile: 0.003 s Read times 90th %ile: 0.002 s Read times 75th %ile: 0.002 s Read times 50th %ile: 0.001 s Read times 25th %ile: 0.001 s Read times Min: 0.000 s </pre>	<pre> Results Summary for Write Operation(s) Total Transferred: 1024.000 MB Total Throughput: 66.50 MB/s Total Duration: 15.399 s Number of Errors: 0 Write times Max: 2.413 s Write times 99th %ile: 0.128 s Write times 90th %ile: 0.091 s Write times 75th %ile: 0.071 s Write times 50th %ile: 0.044 s Write times 25th %ile: 0.016 s Write times Min: 0.006 s Results Summary for Read Operation(s) Total Transferred: 1024.000 MB Total Throughput: 3144.00 MB/s Total Duration: 0.326 s Number of Errors: 0 Read times Max: 0.004 s Read times 99th %ile: 0.003 s Read times 90th %ile: 0.002 s Read times 75th %ile: 0.002 s Read times 50th %ile: 0.001 s Read times 25th %ile: 0.001 s Read times Min: 0.000 s </pre>	<pre> Results Summary for Write Operation(s) Total Transferred: 2048.000 MB Total Throughput: 63.35 MB/s Total Duration: 32.327 s Number of Errors: 0 Write times Max: 1.628 s Write times 99th %ile: 0.632 s Write times 90th %ile: 0.092 s Write times 75th %ile: 0.077 s Write times 50th %ile: 0.047 s Write times 25th %ile: 0.016 s Write times Min: 0.006 s Results Summary for Read Operation(s) Total Transferred: 2048.000 MB Total Throughput: 3096.53 MB/s Total Duration: 0.661 s Number of Errors: 0 Read times Max: 0.004 s Read times 99th %ile: 0.003 s Read times 90th %ile: 0.002 s Read times 75th %ile: 0.002 s Read times 50th %ile: 0.001 s Read times 25th %ile: 0.001 s Read times Min: 0.000 s </pre>	<pre> Results Summary for Write Operation(s) Total Transferred: 4096.000 MB Total Throughput: 62.86 MB/s Total Duration: 65.162 s Number of Errors: 0 Write times Max: 2.091 s Write times 99th %ile: 1.090 s Write times 90th %ile: 0.092 s Write times 75th %ile: 0.076 s Write times 50th %ile: 0.046 s Write times 25th %ile: 0.015 s Write times Min: 0.006 s Results Summary for Read Operation(s) Total Transferred: 4096.000 MB Total Throughput: 2387.46 MB/s Total Duration: 1.583 s Number of Errors: 0 Read times Max: 0.173 s Read times 99th %ile: 0.003 s Read times 90th %ile: 0.002 s Read times 75th %ile: 0.002 s Read times 50th %ile: 0.001 s Read times 25th %ile: 0.001 s Read times Min: 0.000 s </pre>	<pre> Results Summary for Write Operation(s) Total Transferred: 16384.000 MB Total Throughput: 63.12 MB/s Total Duration: 259.556 s Number of Errors: 0 Write times Max: 3.024 s Write times 99th %ile: 0.897 s Write times 90th %ile: 0.092 s Write times 75th %ile: 0.076 s Write times 50th %ile: 0.044 s Write times 25th %ile: 0.015 s Write times Min: 0.006 s Results Summary for Read Operation(s) Total Transferred: 16384.000 MB Total Throughput: 984.70 MB/s Total Duration: 16.639 s Number of Errors: 0 Read times Max: 0.305 s Read times 99th %ile: 0.006 s Read times 90th %ile: 0.005 s Read times 75th %ile: 0.004 s Read times 50th %ile: 0.004 s Read times 25th %ile: 0.004 s Read times Min: 0.002 s </pre>
---	--	--	--	---

随着 numSamples 的增长，总耗时也线性增长，传输速度无法自适应增长，读写尾延迟也越来越明显。

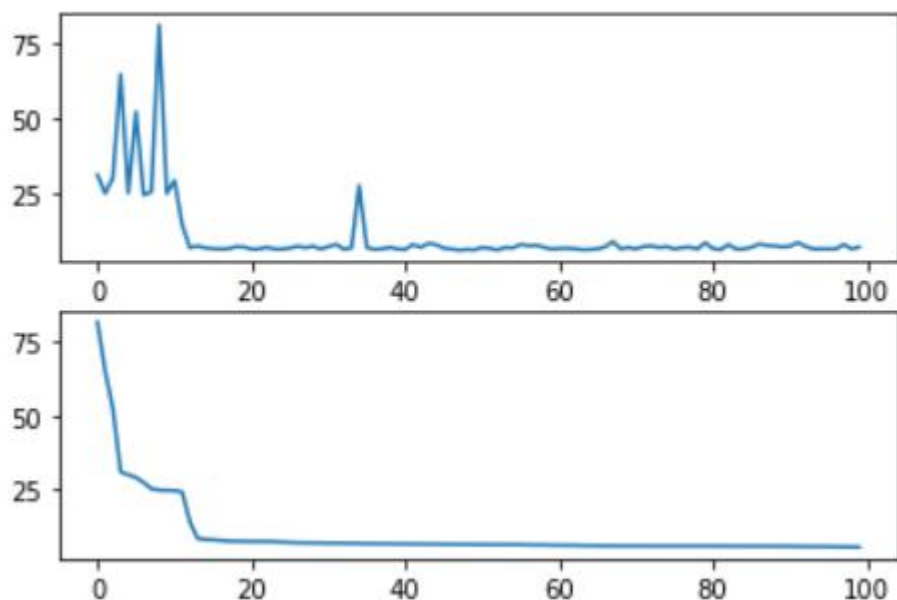
保持 numSamples= 1024，objectSize=524288 (512KB)，numClients 客户端数量逐渐增长 8 16 32 64 128，实验结果如下：

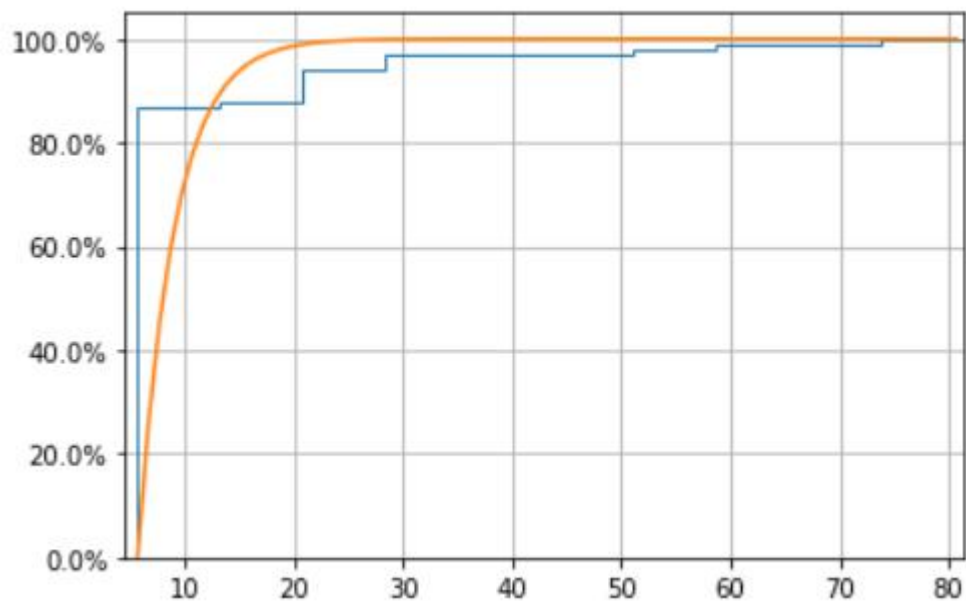


随着并发量的越来越高，总耗时越来越大，传输速度越来越低。
整体读写延迟越来越大，读写尾延迟越来越明显。

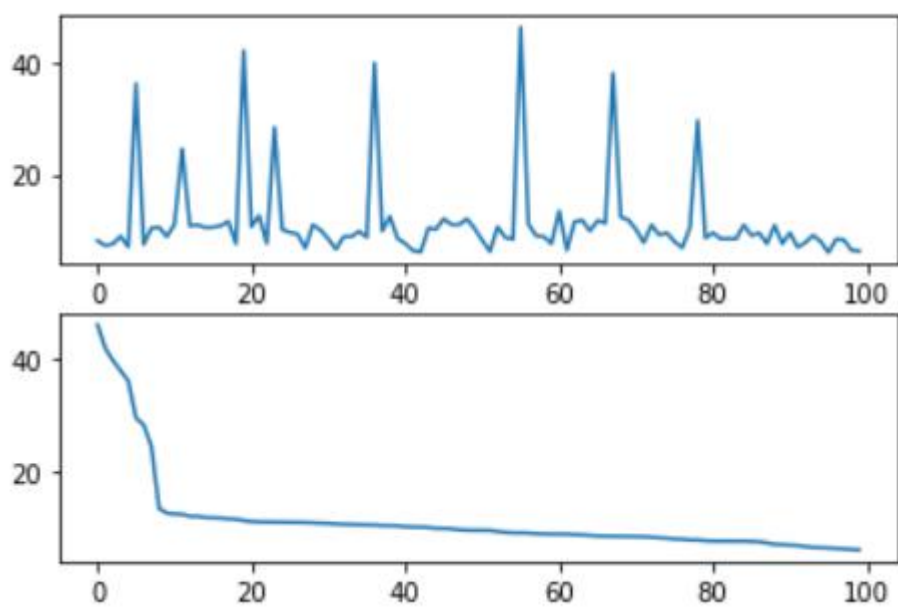
实验三

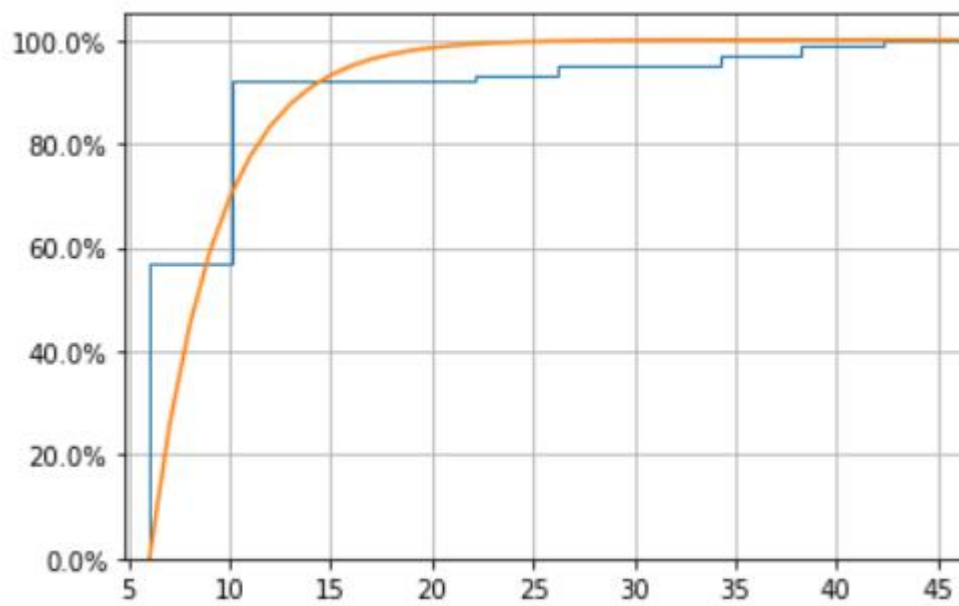
首先，不设置任何策略，运行老师给定的代码观察尾延迟现象。
绝大部分请求都可以在 10ms 之内完成，少数请求的延迟很高，甚至到达 75ms 以上，尾延迟现象明显。





使用关联请求的方式优化尾延迟问题。设定 30ms 为阈值，超过 30ms 未完成的请求，则重新发送一次，实验结果如下图：





尾延迟现象得到显著改善，全部请求都能够在 50ms 内完成，99% 的请求都能够在 40ms 内完成，但尾延迟现象仍然存在。