# 数据中心实验报告

姓名: 黄志颖 学号 M202173706

## 实验一环境搭建

1 使用 minio, 在 minio.exe 目录下使用命令 E:\sjzx\minio.exe server E:\Data --console-address ":9001"

```
Microsoft Windows [版本 10.0.19043.1415]
(c) Microsoft Corporation。保留所有权利。

E:\sjzx>E:\sjzx\minio.exe server E:\Data --console-address ":9001"

You are running an older version of MinIO released 1 month ago Update: Run mc admin update

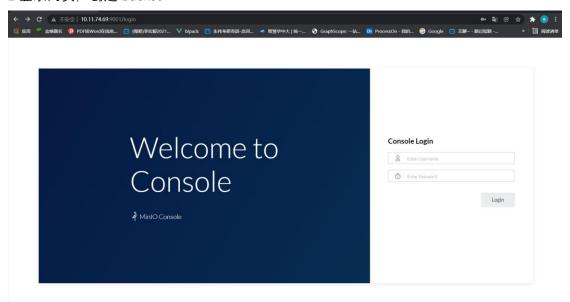
API: http://10.11.74.69:9000 http://127.0.0.1:9000 RootUser: admin RootPass: password

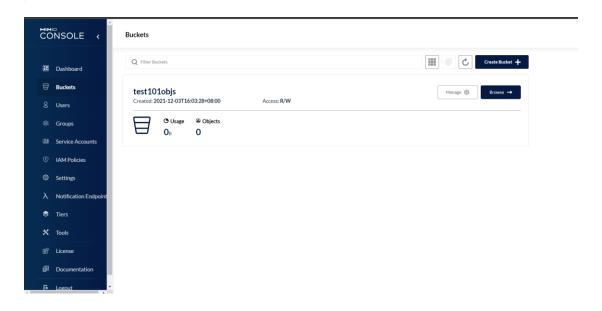
Console: http://10.11.74.69:9001 http://127.0.0.1:9001 RootUser: admin RootPass: password

Command-line: https://docs.min.io/docs/minio-client-quickstart-guide $ mc. exe alias set myminio http://10.11.74.69:9000 admin password

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

### 2 登录网页,创建 bucket





## 实验二: 性能观测

使用 s3bench 观测性能,通过调整环境参数对象尺寸、并发性、服务器数量。查看指标,包含:吞吐率 Throughput、延迟 latency,

1 初始配置 -numClients=10 -numSamples=100 -objectSize=1024

```
Test parameters
endpoint(s): [http://127.0.0.1:9000]
bucket: test101objs
objectNamePrefix: test101objs
objectSize:
                            0.0010 MB
numClients:
                            10
numSamples:
                            100
                       %!d(bool=false)
verbose:
Generating in-memory sample data... Done (2.0004ms)
Running Write test...
Running Read test...
Test parameters
endpoint(s):
                            [http://127.0.0.1:9000]
test101objs
bucket:
objectNamePrefix: test101objs
objectSize:
                            0.0010 MB
nuπClients:
                            10
numSamples:
                            100
                       %!d(bool=false)
verbose:
Results Summary for Write Operation(s)
Total Transferred: 0.098 MB
Total Throughput: 0.18 MB/s
Total Duration:
                             0.528 s
Number of Errors: 0
Write times Max:
                                   0.105 s
Write times Max: 0.105 s
Write times 99th Wile: 0.105 s
Write times 90th Wile: 0.079 s
Write times 75th Wile: 0.064 s
Write times 50th Wile: 0.060 s
Write times 25th Wile: 0.031 s
Write times Min:
                                   0.004 s
Results Summary for Read Operation(s)
Total Transferred: 0.098 MB
Total Throughput: 7.64 MB/s
Total Duration: 0.013 s
Number of Errors: 0
                                  0.002 s
Read times Max:
Read times Max: 0.002 s
Read times 99th Wile: 0.002 s
Read times 90th Wile: 0.002 s
Read times 75th Wile: 0.002 s
Read times 50th Wile: 0.001 s
Read times 25th Wile: 0.001 s
Read times Min:
                                  0.001 s
Cleaning up 100 objects...
Deleting a batch of 100 objects in range {0, 99}... Succeeded
Successfully deleted 100/100 objects in 63.0457ms
```

```
Test parameters
endpoint(s):
                          [http://127.0.0.1:9000]
bucket: test101objs
objectNamePrefix: test101objs
objectSize:
                          0.0010 MB
numClients:
                          20
                          100
numSamples:
verbose:
                     %!d(boo1=false)
Generating in-memory sample data... Done (1.0001ms)
Running Write test...
Running Read test...
Test parameters
endpoint(s):
                          [http://127.0.0.1:9000]
bucket: test101objs
objectNamePrefix: test101objs
objectSize: 0.0010 MB
numClients:
                          20
                          100
numSamp1es:
verbose:
                     %!d(boo1=fa1se)
Results Summary for Write Operation(s)
Total Transferred: 0.098 MB
Total Throughput: 0.21 MB/s
Total Duration: 0.476 s
Number of Errors: 0
Write times Max:
                                 0.142 s
Write times Max: 0.142 s
Write times 99th Wile: 0.142 s
Write times 90th Wile: 0.110 s
Write times 75th Wile: 0.108 s
Write times 50th Wile: 0.094 s
Write times 25th Wile: 0.081 s
Vrite times Min∶
                                 0.032 s
Results Summary for Read Operation(s)
Total Transferred: 0.098 MB
Total Throughput: 9.41 MB/s
Total Duration: 0.010 s
Number of Errors: 0
Read times Max:
                                0.004 s
Read times 99th %ile: 0.004 s
Read times 90th %ile: 0.003 s
Read times 30th Mile: 0.003 s
Read times 75th Mile: 0.002 s
Read times 50th Mile: 0.002 s
Read times 25th Mile: 0.002 s
Read times Min:
                                0.001 s
Cleaning up 100 objects...
Deleting a batch of 100 objects in range {0, 99}... Succeeded
Successfully deleted 100/100 objects in 63.234ms
```

#### 写期间延迟变大吞吐量年达

读期间可以看到指标吞吐率由 7.64MB/s 编程 9.41MB/s,最大时延由 0.002s 增长到 0.004s 总时间减少

```
Test parameters
endpoint(s): [http://127.0.0.1:9000]
bucket: test101objs
objectNamePrefix: test101objs
objectSize:
nuπClients:
                                0.0010 MB
numSamples:
                           %!d(bool=false)
verbose:
Generating in-memory sample data... Done (1.0001ms)
Running Write test...
Running Read test...
Test parameters
endpoint(s): [http://127.0.0.1:9000]
bucket: test101objs
objectNamePrefix: test101objs
objectSize:
                                0.0010 MB
numClients:
                                10
numSamples:
verbose:
                           %!d(boo1=false)
Results Summary for Write Operation(s)
Total Transferred: 0.195 MB
Total Throughput: 0.19 MB/s
Total Duration: 1.029 s
Number of Errors: 0
Write times Max:
                                         0.107 s
Write times Max: 0.107 s
Write times 99th Wile: 0.106 s
Write times 90th Wile: 0.078 s
Write times 75th Wile: 0.063 s
Write times 50th Wile: 0.048 s
Write times 25th Wile: 0.031 s
Write times Min: 0.004 s
Results Summary for Read Operation(s)
Total Transferred: 0.195 MB
Total Throughput: 8.29 MB/s
Total Duration: 0.024 s
Total Duration:
Number of Errors:
Read times Max:
                                        0.004 s
Read times 99th %ile: 0.004 s
Read times 90th %ile: 0.002 s
Read times 75th %ile: 0.001 s
Read times 50th %ile: 0.001 s
Read times 25th Wile: 0.001 s
Read times Min: 0.001 s
Cleaning up 200 objects...
Deleting a batch of 200 objects in range {0, 199}... Succeeded
Successfully deleted 200/200 objects in 129.1128ms
```

#### 写期间持续时间变长

读期间可以看到指标吞吐率由 7.64MB/s 变成 8.29MB/s, 最大时延由 0.002s 增长到 0.004s 但 75%都能在 0.001s 内跑完, 总时间显著增加。

```
E:\sjzx>s3bench.exe -accessKey=admin -accessSecret=password -endp
Test parameters
                            [http://127.0.0.1:9000]
endpoint(s):
                            test101objs
bucket:
objectNamePrefix: test101objs
objectSize: 0.0020 MB
numClients:
numSamples:
                            100
                       %!d(boo1=false)
verbose:
Generating in-memory sample data... Done (1.0006ms)
Running Write test...
Running Read test...
Test parameters
endpoint(s):
                            [http://127.0.0.1:9000]
test101objs
bucket: test101objs
objectNamePrefix: test101objs
objectSize:
                            0.0020 MB
numClients:
numSamples:
                            10
                            100
verbose:
                       %!d(bool=false)
Results Summary for Write Operation(s)
Total Transferred: 0.195 MB
Total Throughput: 0.44 MB/s
Total Duration: 0.441 s
Number of Errors:
Write times Max:
                                   0.110 s
Write times Max. 0.110 s
Write times 99th %ile: 0.110 s
Write times 90th %ile: 0.067 s
Write times 75th %ile: 0.062 s
Write times 50th %ile: 0.030 s
                                   0.004 s
Write times Min:
Results Summary for Read Operation(s)
Total Transferred: 0.195 MB
Total Throughput: 17.08 MB/s
Total Duration: 0.011 s
Number of Errors: 0
Read times Max:
                                 0.002 s
Read times Max. 0.002 s
Read times 99th Wile: 0.002 s
Read times 90th Wile: 0.002 s
Read times 75th Wile: 0.001 s
Read times 50th Wile: 0.001 s
Read times 25th %ile: 0.001 s
Read times Min:
Cleaning up 100 objects...
Deleting a batch of 100 objects in range {0, 99}... Succeeded
Successfully deleted 100/100 objects in 64.2714ms
```

#### 写期间吞吐量变大

读期间可以看到指标吞吐率由 7.64MB/s 变成 17.08MB/s,最大时延不变,总时间不变

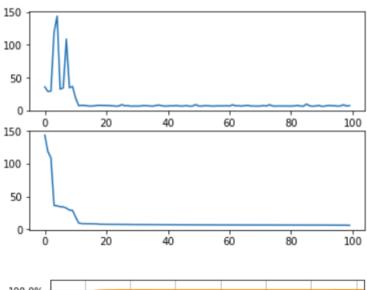
## 5总结

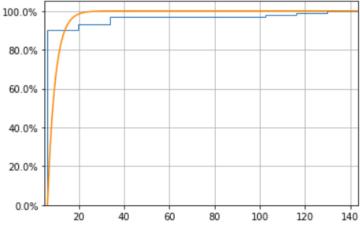
numClients	numSamples	Objectsize	吞吐率	总延迟	吞吐率	总延迟
			(写) MB/s	(写) s	(读) MB/s	(读) s
10	100	1024	0.18	0.528	7.64	0.013
20	100	1024	0.21	0.476	9.41	0.010
10	200	1024	0.19	1.029	8.29	0.024
10	100	2048	0.44	0.441	17.08	0.011

分析得出,大致为,访问的线程越多,对象大小越大,尾延迟现象越明显。

# 实验三: 尾延迟

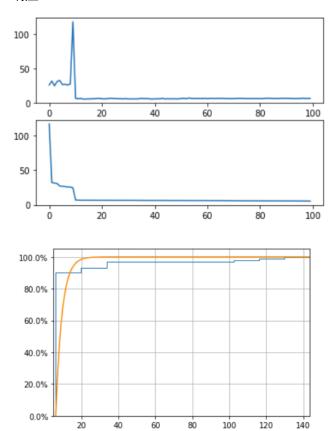
1尾延迟观测,原始尾延迟,队列长度为100



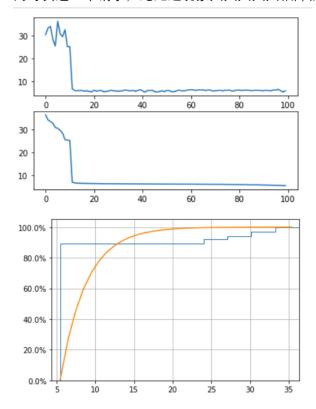


### 2 关联请求缓解尾延迟现象

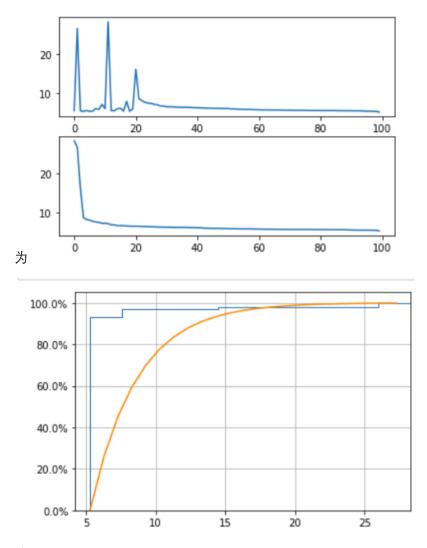
同时发送两个请求,响应时撤销后续请求尾延迟现象大幅降低,大部分请求都能在短时间内响应



同时发送3个请求,尾延迟现象大大大大大幅降低,大部分请求都能在短时间内响应



3 对冲请求, 在关联两个请求时,在此种情况下依旧有 5%的请求延迟过高,对这部分请求进行对冲结果



效果显著