

**《现代数据中心》**

**实验报告**

**课程名称： 现代数据中心**

**专业班级： 硕2110**

**学 号： M202173860**

**姓 名： 许海涛**

**指导教师： 施展、童薇**

**报告日期： 2022年1月5日**

**计算机科学与技术学院**

目录

[1.实验一：系统搭建 3](#_Toc92301031)

[2.实验二：性能观测 3](#_Toc92301032)

[3.实验三：尾延迟挑战 9](#_Toc92301033)

# 1.实验一：系统搭建

代码管理和仓库：Git tutorial [https://github.com/cs-course/git-tutorial](https://gitee.com/link?target=https%3A%2F%2Fgithub.com%2Fcs-course%2Fgit-tutorial)

系统环境：python 3.8

服务端：Minio

客户端：Minio Client

在完成系统搭建后，我们先运行run-minio.cmd文件，设置minio\_root\_user=hust，minio\_root\_password=hust\_obs，完成服务端与客户端的链接，面向对象存储服务开始运行。在浏览器窗口输入127.0.0.1:9000，然后输入 之前设置好的use 和passport，可以看到如图1-1所示界面，表示成功完成系统搭建任务，开始运行面向对象存储服务。

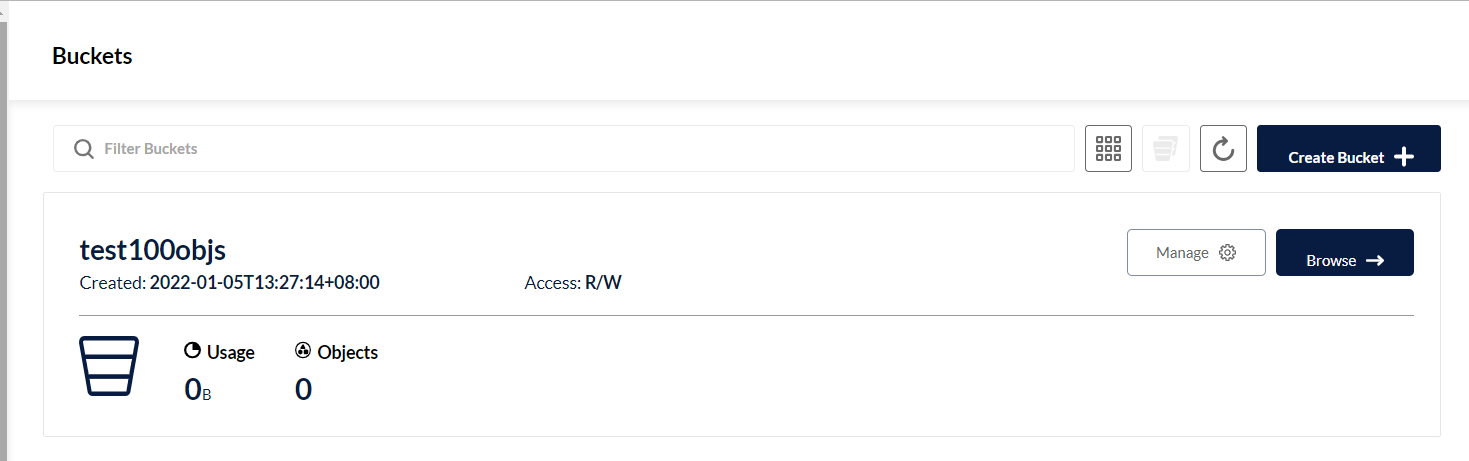


图1-1. 面向对象存储服务

# 2.实验二：性能观测

在性能观测实验部分选用的测评工具是S3 Bench。下面是该实验的一些相关概念。

吞吐率：原指一个业务系统在单位时间内提供的产量（或服务量）。在计算机或数据通信系统，指的是单位时间内通过某通信信道（a communication channel）或某个节点成功交付数据的平均速率，通常以每秒比特数（bps， bits per second ）为单位。吞吐受带宽限制，带宽越大，吞吐率的上限才可能越高。

带宽：在数字设备中，带宽指单位时间能通过链路的数据量。通常以bps来表示，即每秒可传输之位数。带宽越大，其传输能力就越强。

延时：表示请求数据包发送后，收到对端响应，所需要的时间延迟。不同的场景有着不同的含义，比如可以表示建立 TCP 连接所需的时间延迟，或一个数据包往返所需的时间延迟。

正式测试如下所示：

对于numclients测试case如下所示：

  （1）-numClients=8 ，-numSamples=1024，-objectSize=1024，时候，测试数据如图2-1所示：

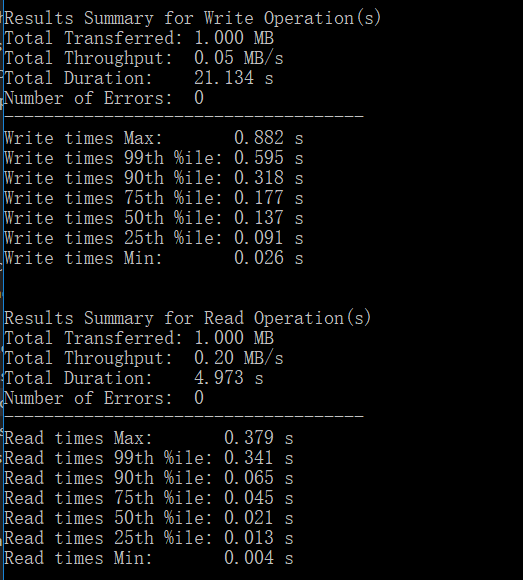


图2-1 numclient测试数据1

（2）-numClients=16，-numSamples=1024，-objectSize=1024，时候，测试数据如图2-2所示：

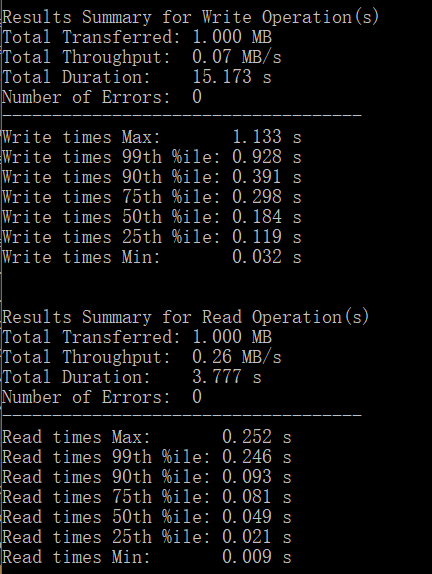


图2-2 numclient测试数据2

（3）-numClients=32，-numSamples=1024，-objectSize=1024，时候，测试数据如图2-3所示：

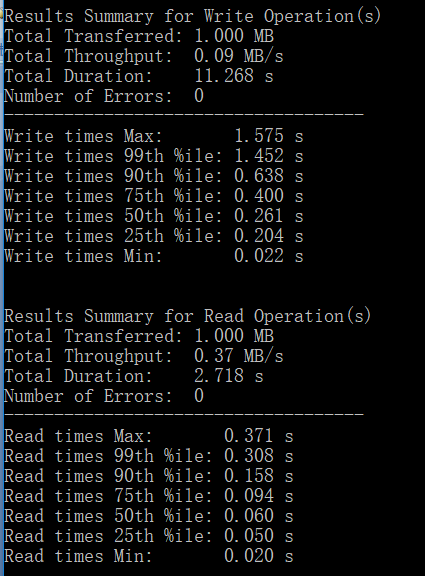


图2-3 numclient测试数据3

我们可以看到，随着numclient数目的增加，总的带宽throughput在变大，延时duration都在减少。

对于objectSize测试case如下所示：

  （1）-numClients=8 ，-numSamples=256，-objectSize=1024，时候，测试数据如下：

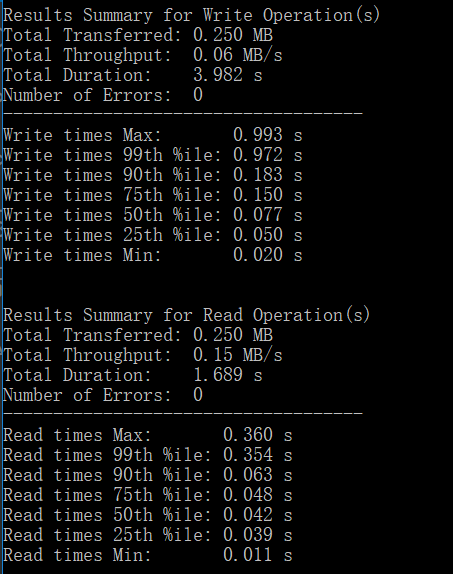


图2-4 objectSize测试数据1

（2）-numClients=8，-numSamples=256，-objectSize=2048，时候，测试数据如下：

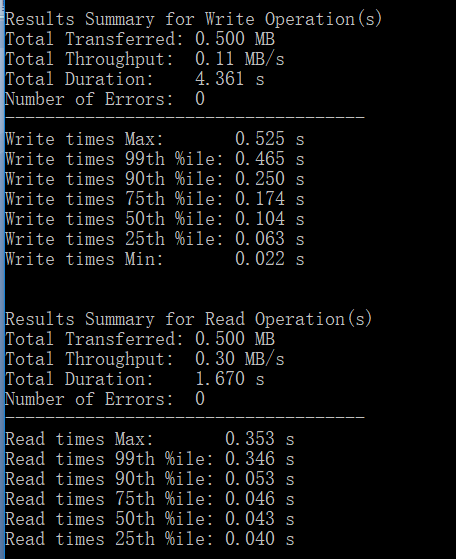


图2-5 objectSize测试数据2

（3）-numClients=8，-numSamples=256，-objectSize=4096，时候，测试数据如下：

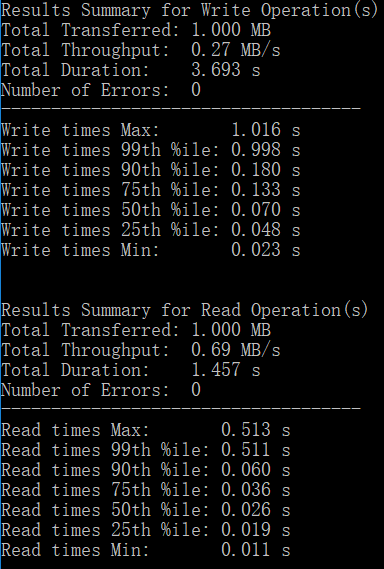


图2-6 objectSize测试数据3

我们可以看到，随着objectsize数目的增加，总的带宽throughput在增加，延时duration在减少。

对于numSample测试case如下所示：

（1）-numClients=8 ，-numSamples=256，-objectSize=1024，时候，测试数据如下：

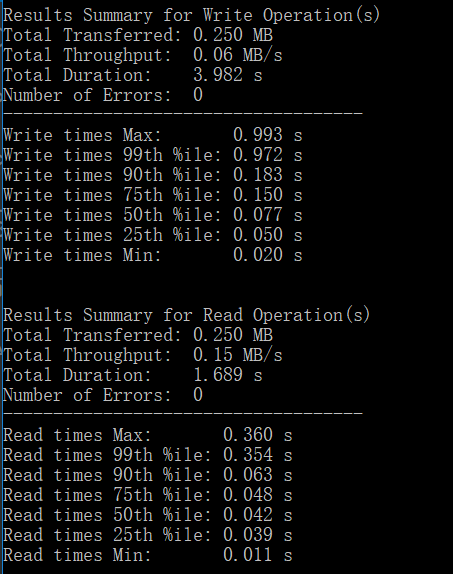


图2-7 numSample测试数据1

（2）-numClients=8，-numSamples=512，-objectSize=1024，时候，测试数据如下：

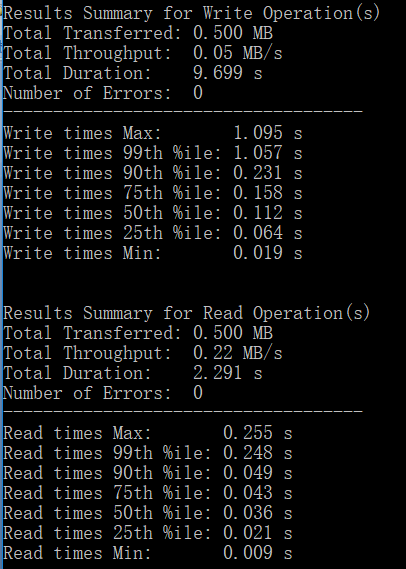


图2-8 numSample测试数据2

（3）-numClients=8，-numSamples=1024，-objectSize=1024，时候，测试数据如下：

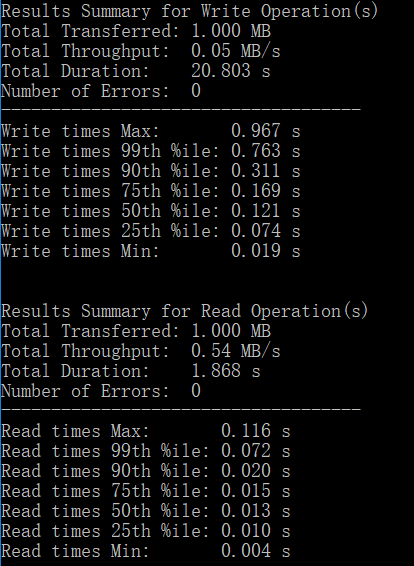


图2-9 numSample测试数据3

我们可以看到，随着numSample数目的增加，总的带宽throughput在增加，延时duration在减少。

可以看到，当加大负载也就是增多numClient和objectSize的时候，尾大不掉的情况一直存在。

# 3.实验三：尾延迟挑战

在尾延迟挑战实验部分，先对于实验环境进行一个初始化的操作，完成密钥准备和会话建立以及服务连接的工作。然后需要准备负载, 可以按照几种不同请求到达率（Inter-Arrival Time，IAT）设置，在初始化本地数据文件以后，发起请求并计算系统的停留时间，按照请求到达率来限制并执行跟踪请求，其中，分辨按照100ms内不超过两个请求，四个请求，八个请求进行设置。完成到达率设置之后，按照预设的IAT发起请求，并采集延迟，将延迟保存在CSV文件中，最后运行latency-plot.ipynb文件进行画图。

可以看到，其一百个请求的延迟分布情况如下图3-1所示。我们可以看到，虽然平均延迟满足用户需求，但是总是有例外，在平均延迟之上，我们看到的平均延迟是被平均之后的指标。在排序后，我们可以看到一个很明显的长尾分布图，大部分情况下延迟比较低，但是少部分情况会有较高延迟。

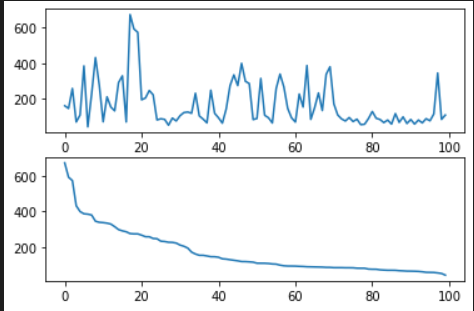


图3-1 一百个请求的延迟分布情况

此外，我们还画出了百分比延迟的图，如图3-2所示，并与绘制的排队论的模型进行拟合，可以看到阶梯状的蓝色的线，本质是一个指数的累计概率分布，表示对应的百分位的出现可能更小或者更大，我们发布对冲请求的合理时机与分布有关，而与平均延迟，延迟的最大值和最小值无关。

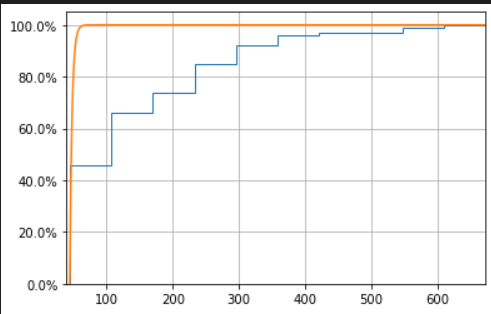


图3-2 百分比延迟图