

**实验报告**

**课程名称：数据中心技术**

**专业班级： 硕2110班**

**学 号： M202173879**

**姓 名： 邓天聪**

**授课教师： 施展**

**报告日期： 2022.1.6**

**计算机科学与技术学院**

目录

[实验一：系统搭建 3](#_Toc92405072)

[实验二：性能观测 5](#_Toc92405073)

[**2.1 numClients和objectSize不变** 5](#_Toc92405074)

[**2.2 numClients和numSample不变** 8](#_Toc92405075)

[**2.3 objectSize和numSample不变** 10](#_Toc92405076)

[实验三：尾延迟挑战 12](#_Toc92405077)

**实验一：系统搭建**

近年来，随着深度学习技术的发展和深度卷积神经网络（convolution neural network，CNN）的出现，特别是VGGNET、GOOGLENET等卷积网络模型的提出，在图像分类识别方面取得了重大突破。此外，由于深度学习在处理端到端问题时具有很强的学习能力，因此它可以处理更复杂的任务，如目标检测、视频智能监控、人体姿态估计等。

系统：windows

服务端：minio

客户端：osm

1、下载minio，在bin目录下创建data文件夹作为数据存储的文件夹，以管理员身份打开cmd，运行run-minio.cmd程序，搭建MinIO服务器。服务器成功启动后会在cmd输出默认的用户名和密码，为hust、hust\_obs。

文本

描述已自动生成

电脑截图

描述已自动生成

2、osm.exe在浏览器中访问cmd输出的网址127.0.0.1:9000，进入minio服务器端的图形管理界面。

图形用户界面, 应用程序, PowerPoint

描述已自动生成

在buckets面板上创建一个名为“loadgen”的bucket桶，用于性能测试。

图形用户界面, 文本, 应用程序

描述已自动生成

**实验二：性能观测**

测评工具S3 Bench

**2.1 numClients和objectSize不变**

当numSample为256时

文本, 信件

描述已自动生成

文本

描述已自动生成

由运行结果所示，共256个文件。其中，写操作总吞吐率为0.34MB/s。

写操作总时间为0.738s，其中最大写时长为0.077s，最小写时长为0.007s，其中90%的写操作在0.041s时完成，99%的写操作在0.062s时完成。

而读操作总吞吐量为10.17MB/s，读操作总时间为0.025s，其中最大读时长为0.002s，最小读时长为0.000s，其中90%的读操作在0.001s时完成，99%的读操作在0.002s时完成。

numSample改为512时

文本, 信件

描述已自动生成

文本

描述已自动生成

numSample为1024时

文本, 信件

描述已自动生成

文本

描述已自动生成

numSample为2084时

文本, 信件

描述已自动生成

文本

描述已自动生成

**2.2 numClients和numSample不变**

当objectSize为2048时

文本, 信件

描述已自动生成

文本

描述已自动生成

当objectSize改为4096时

文本, 信件

描述已自动生成

文本

描述已自动生成

当objectSize改为8192时

文本, 信件

描述已自动生成

文本

描述已自动生成

写的总吞吐率降低，总用时减少；读的吞吐率增加

**2.3 objectSize和numSample不变**

当numClients改为16时候

文本, 信件

描述已自动生成

文本

描述已自动生成

当numClients改为32时候

文本, 信件

描述已自动生成

文本

描述已自动生成

当numClients改为64时候

文本, 信件

描述已自动生成

文本

描述已自动生成

读的总吞吐率减少，总用时增加

**实验三：尾延迟挑战**

一百个请求的延迟分布情况如下图所示。在排序后，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。

图形用户界面, 图表, 直方图

描述已自动生成

画出百分比延迟的图，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。我们可以看到多达百分之九十的请求在十五毫秒左右完成，对于百分之九十五的请求在二十毫秒左右完成。

图表

描述已自动生成

从图中可以看出存在一部分的写请求的开销远远超过其他写请求，就是尾延迟现象。

设置30ms作为界限，如果请求的时间超过30ms，那么就认定为失败，再发送一个相同的请求，实验结果如下：

图形用户界面, 图表, 直方图

描述已自动生成

在本环境中，此方法下99%的请求都可以在30ms内发送出去

图表

中度可信度描述已自动生成

可以看到在本实验的环境中，延迟整体偏低，但尾延迟的现象仍然存在。