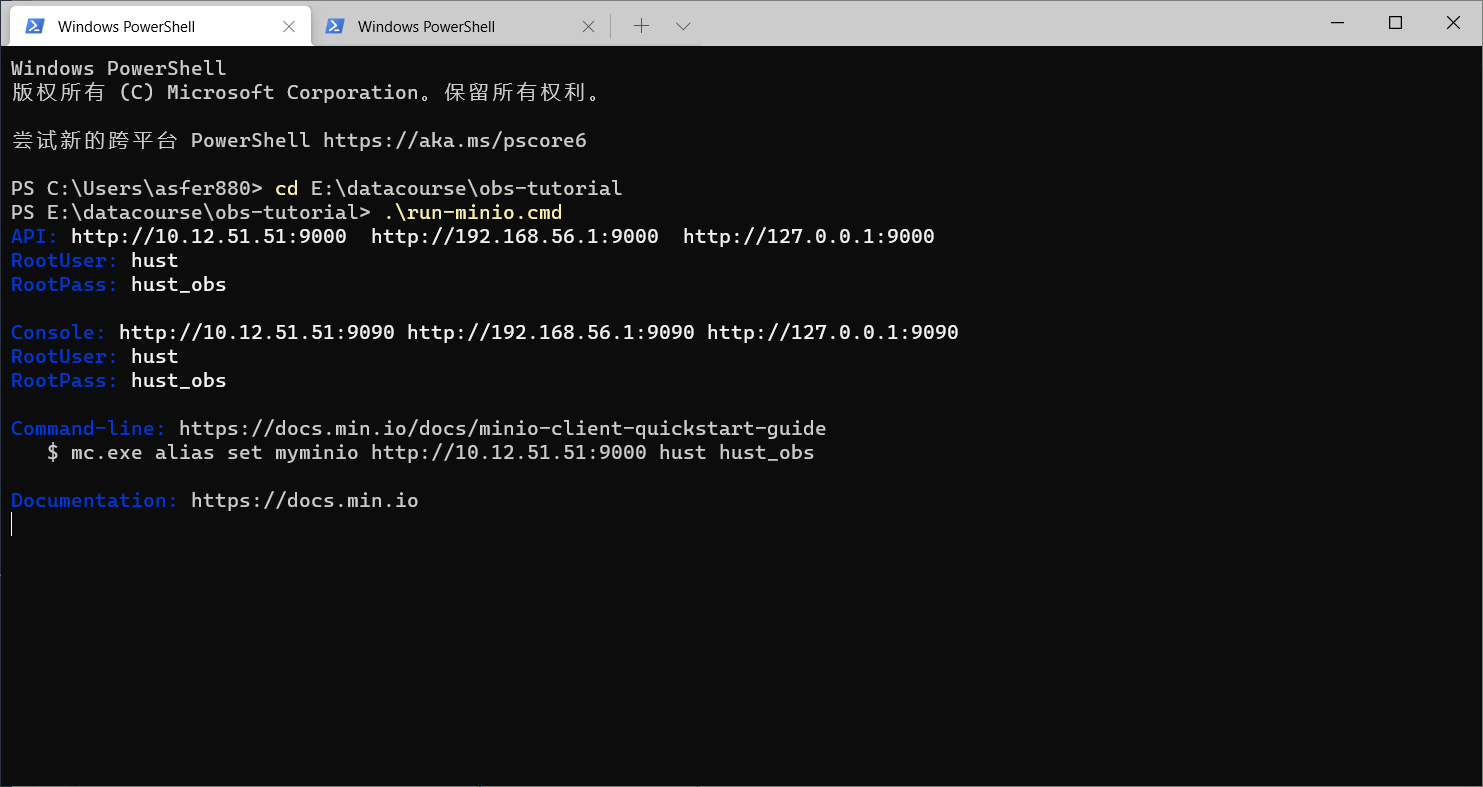
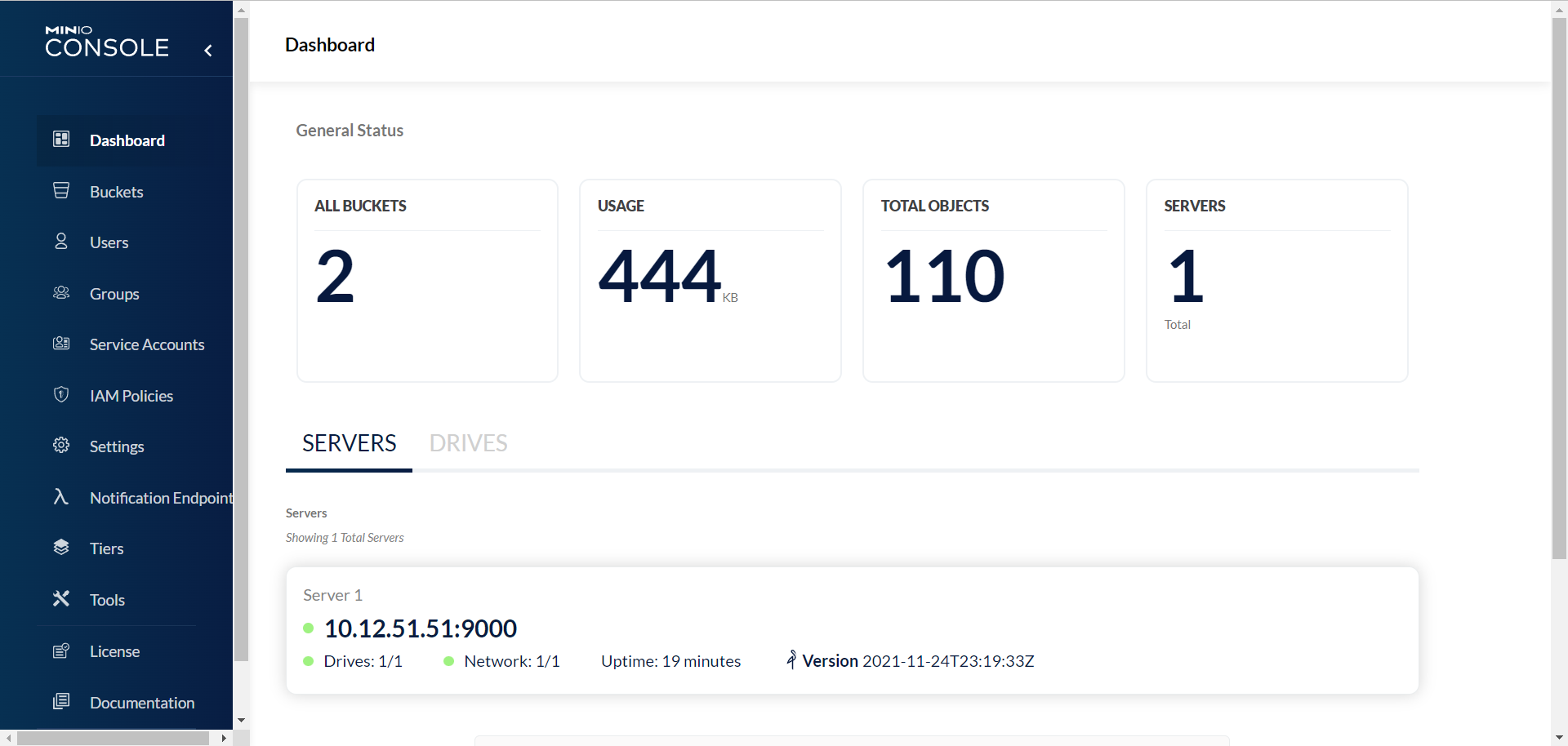
# 对象存储实验报告

吴迪 M202173669

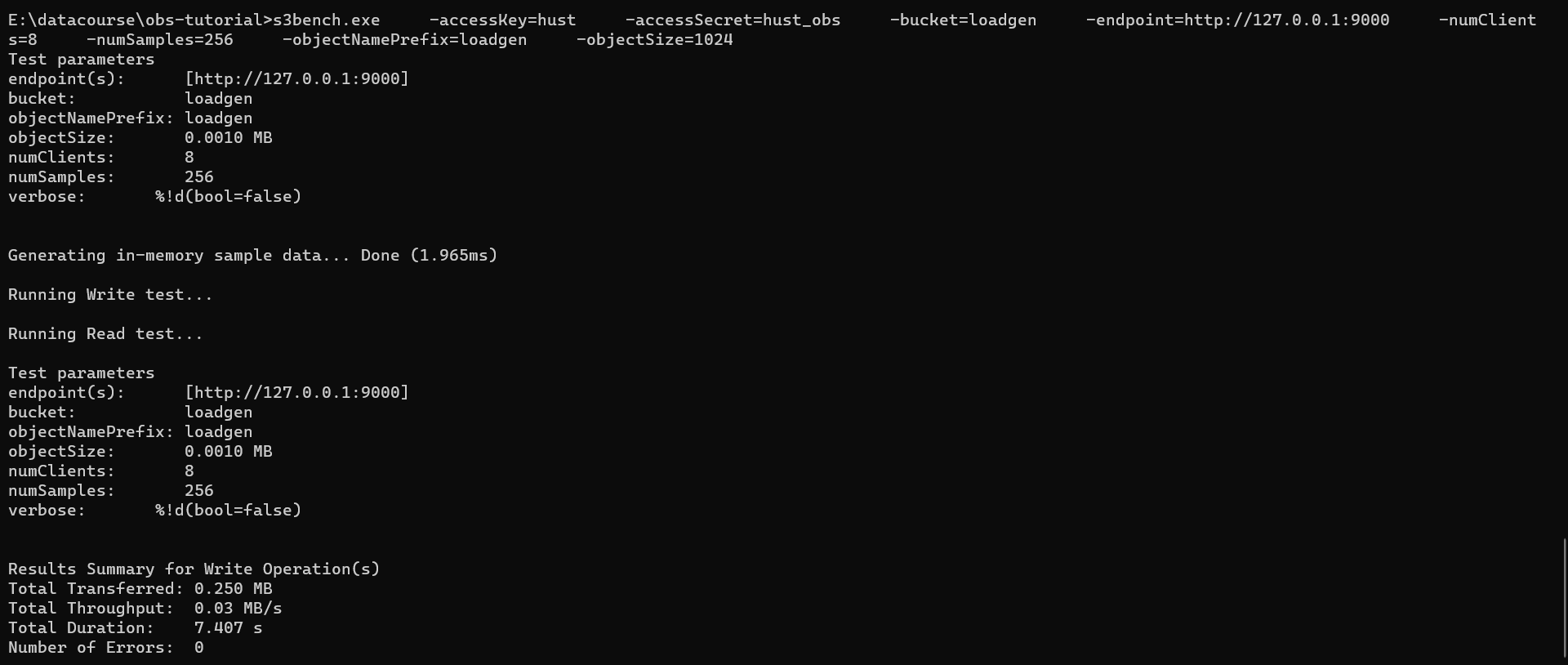
## 性能指标

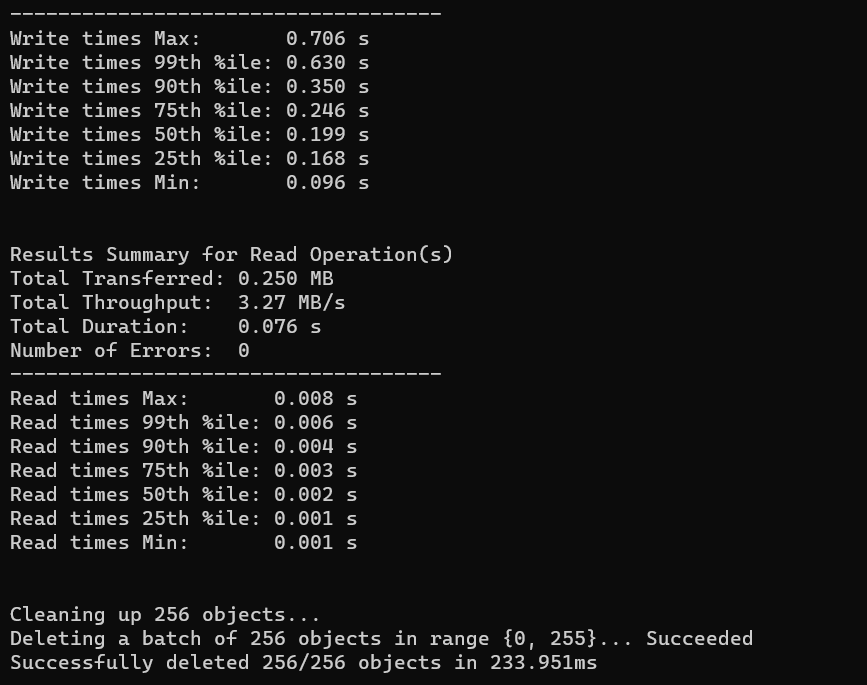
启动minio





通过s3bench向minio发送请求

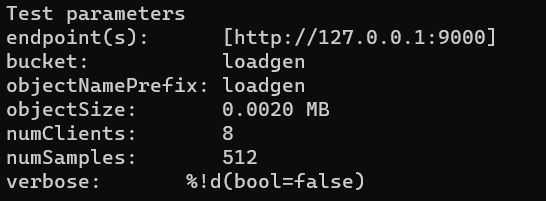


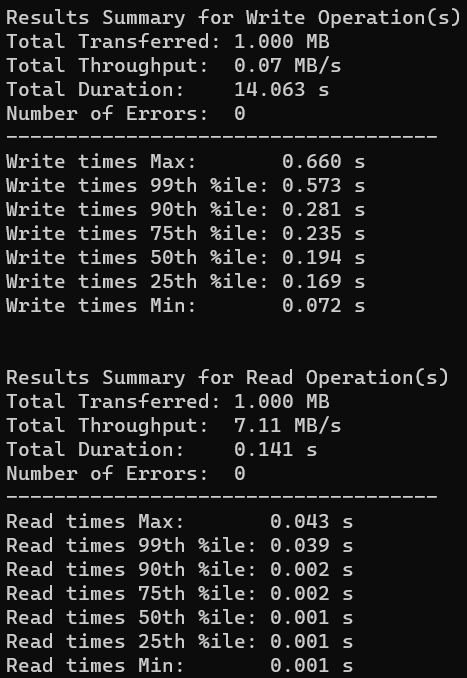


## 不同负载下指标、延迟的分布

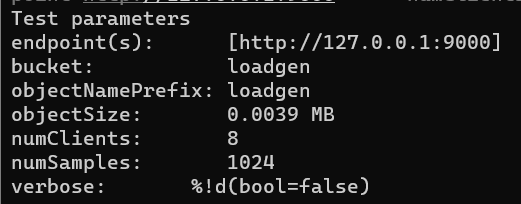
更改负载，修改对象大小：1024，2048，4096以及对象数量256，512，1024，通过s3bench向minio发送请求，观测延迟分布

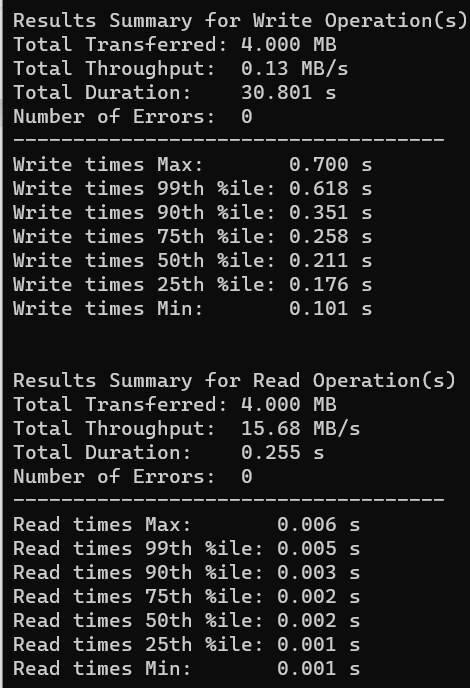
数量：512，大小2048KB





数量：1024，大小4096KB



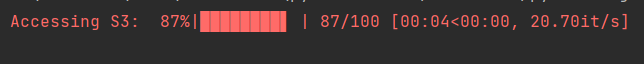


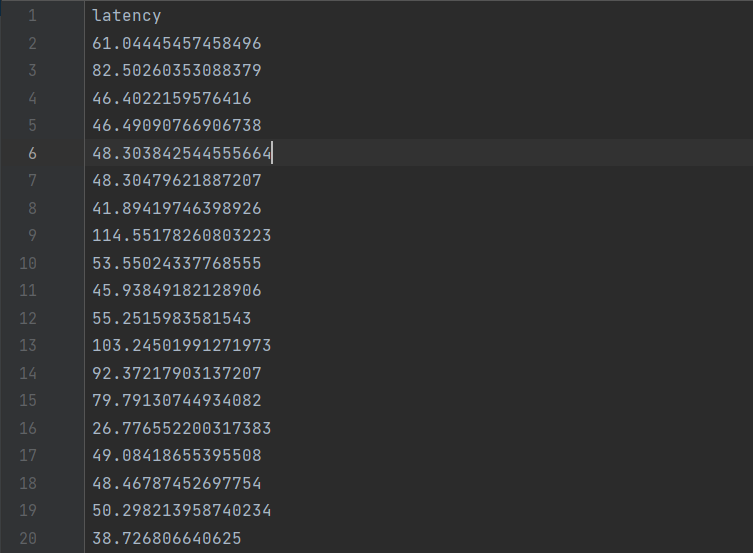
可以发现共同的特点，有部分的写请求用时远远超过前面的写请求

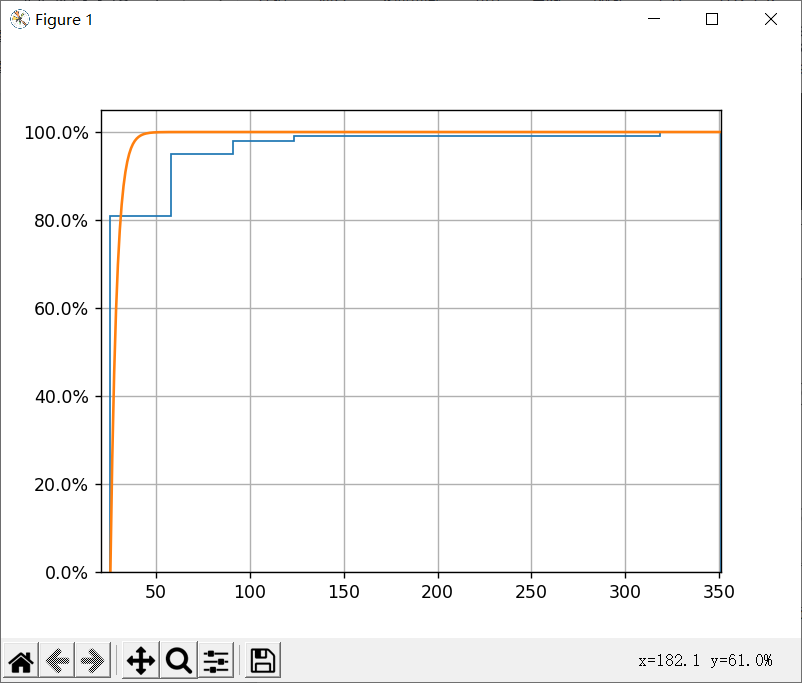
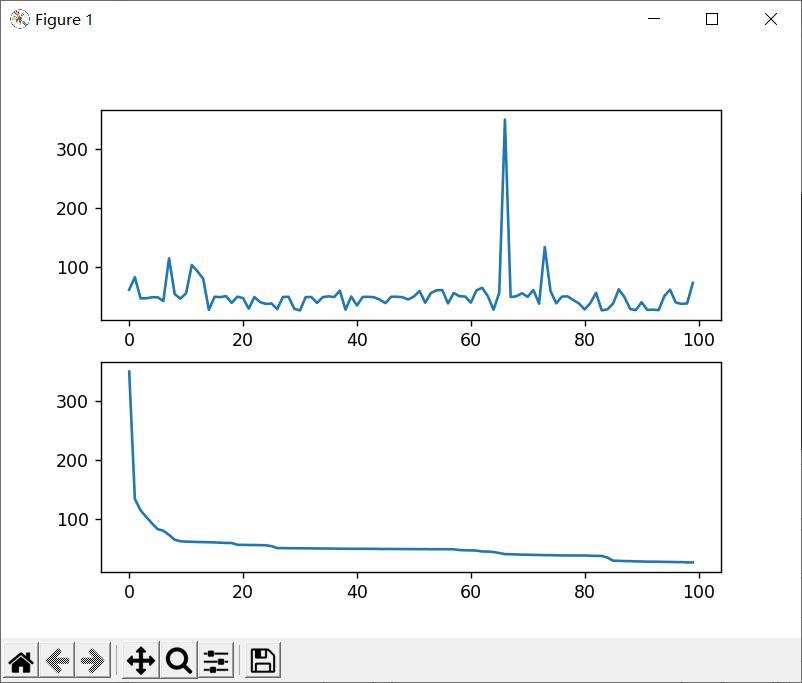
## 观测尾延迟

运行obs-tutorial中的python脚本

得到结果如下：





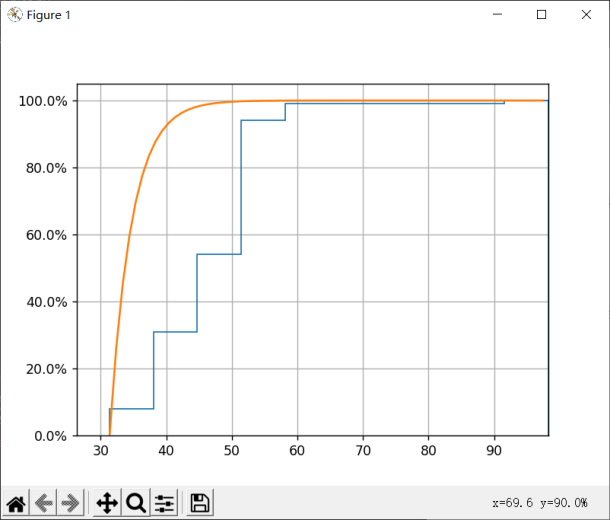
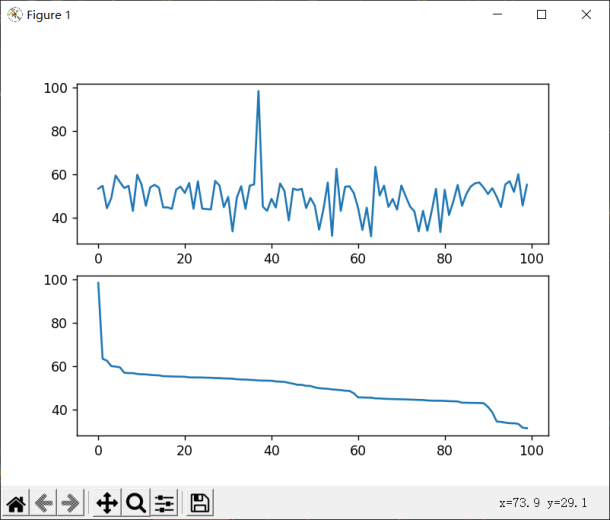


从图中可以看出存在一部分的写请求的开销远远超过其他写请求，就是尾延迟现象。

## 尝试对冲请求

从上图的运行结果来看，大约90%的数据都可以在60ms内完成

因此设置60ms作为界限，如果请求的时间超过60ms，那么就认定为失败，再发送一个相同的请求，实验结果如下：



在本环境中，此方法下99%的请求都可以在60ms内发送出去，比之前的结果好了很多。