

**2019** 级

《大数据存储系统与管理》课程

**实 验 报 告**

**姓 名 赵英举**

**学 号 U201915093**

**班 号 物联网1901班**

**日 期 2022.04.20**

**计算机科学与技术学院**

1. **实验目的**

1. 熟悉对象存储技术及其特性；

2. 实践对象存储系统，部署实验环境，进行初步测试；

3. 基于搭建完成的服务器及客户端进行对象存储的性能分析。

1. **实验背景**

对象存储，也叫做基于对象的存储，是用来描述解决和处理离散单元的方法的通用术语，这些离散单元被称作为对象。对象存储根本上改变了存储蓝图。它处理和解决了曾经被认为是棘手的存储问题：不间断可扩展性、弹性下降、限制数据持久性、无限技术更新和成本失控。存储专家对其潜在的优势感到兴奋，尤其是他们的绝大多数数据都碰巧是被动的或者是冷数据。

1. **实验环境**

**计算机环境：**

Windows 11 家庭中文版

**环境配置：**

Java环境：在计算机中配置JDK1.7

对象存储服务端： Minio Client

对象存储客户端： Minio

对象存储性能测试工具：COSBench

1. **实验内容**

**1、实验一 环境配置**

熟悉基础环境：包括代码管理环境Gitee中仓库的建立和文件的上传、Java环境配置中的JDK版本安装。其中Java环境配置过程中，原来配置的jdk15版本过高致使COSBench安装过程中出现很多的报错，无法正常安装运行程序，最终安装了更低版本的jdk1.7完成了Java环境的配置。

**2、实验二 对象存储技术实践**

1）安装服务端和客户端

在Minio官网https://www.minio.io/downloads.html下载Minio的服务器和客户端，在Minio文件夹目录下进行安装：

在命令行中输入minio.exe server C:miniodata启动Minio服务器并在C盘下miniodata中创建文件夹，启动成功后会显示出服务器的端口信息和用户名、密码等，如图1.1所示。

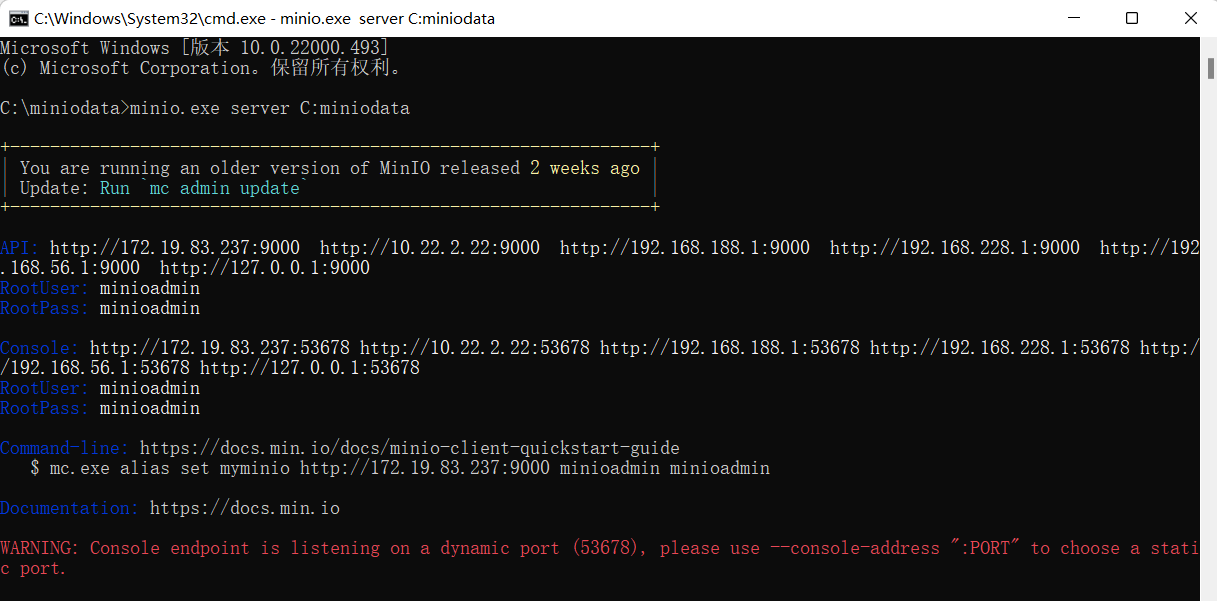


图1.1 启动Minio服务器

然后根据服务器启动中的提示信息输入命令行$ mc.exe alias set myminio http://10.10.93.140:9000 minioadmin minioadmin后启动客户端如图1.2所示。

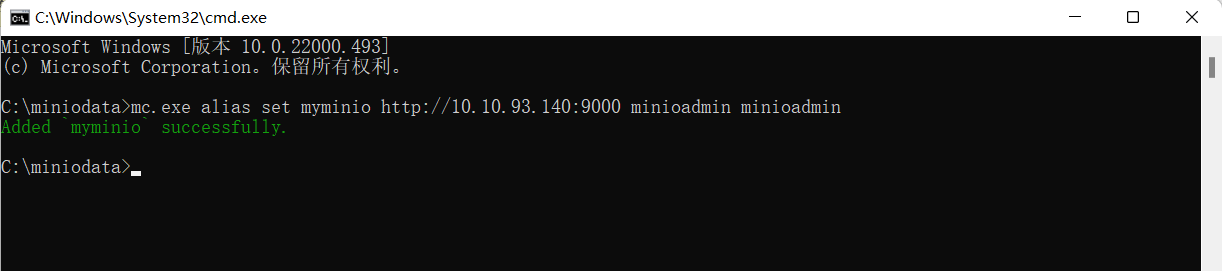


图1.2 启动Minio客户端

用户名和密码使用默认的，然后在浏览器中输入127.0.0.1:9000即可进行登录界面的显示，如图1.3所示。

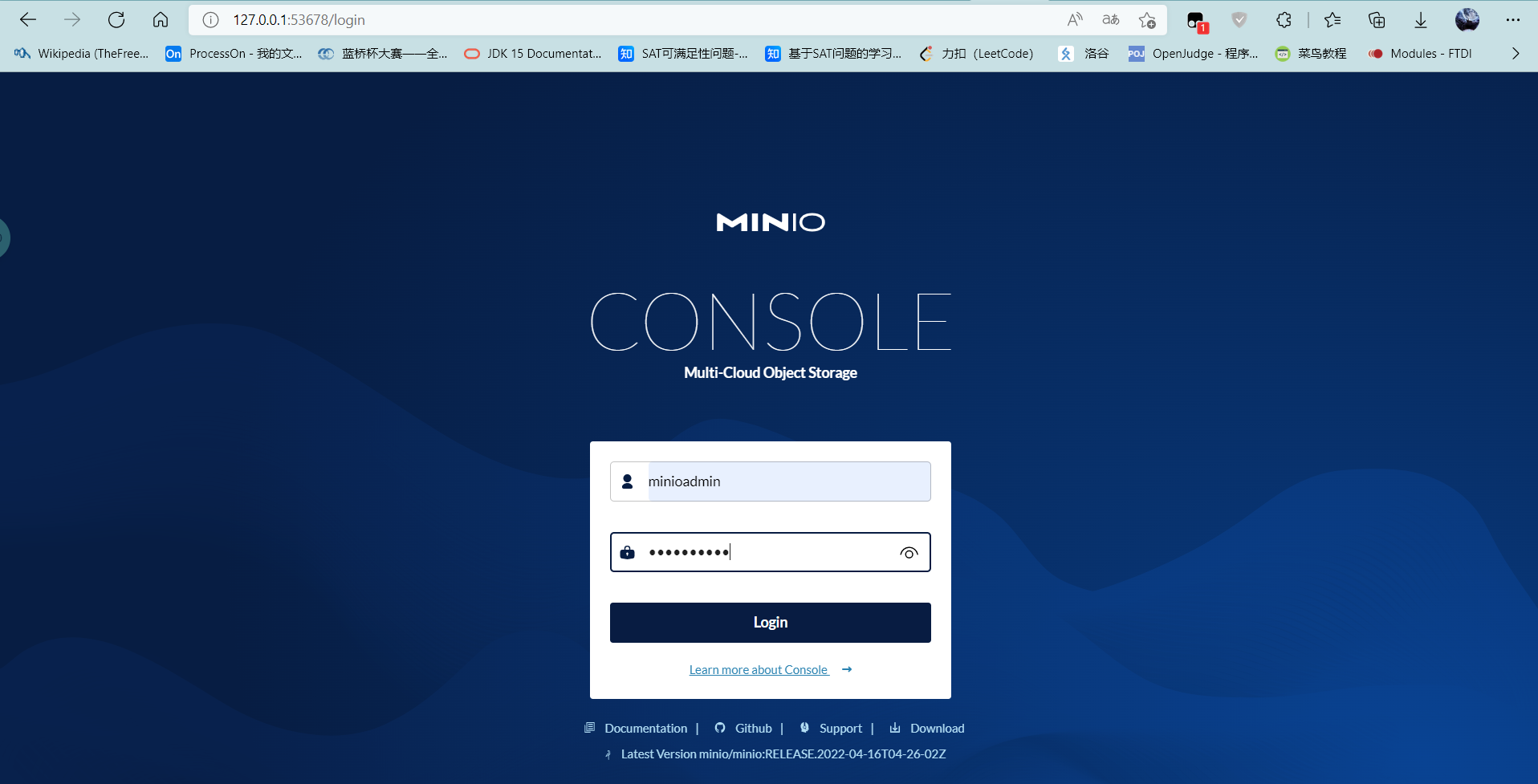


图1.3 登录Minio

输入默认的用户名和密码进入后界面如图1.4所示。

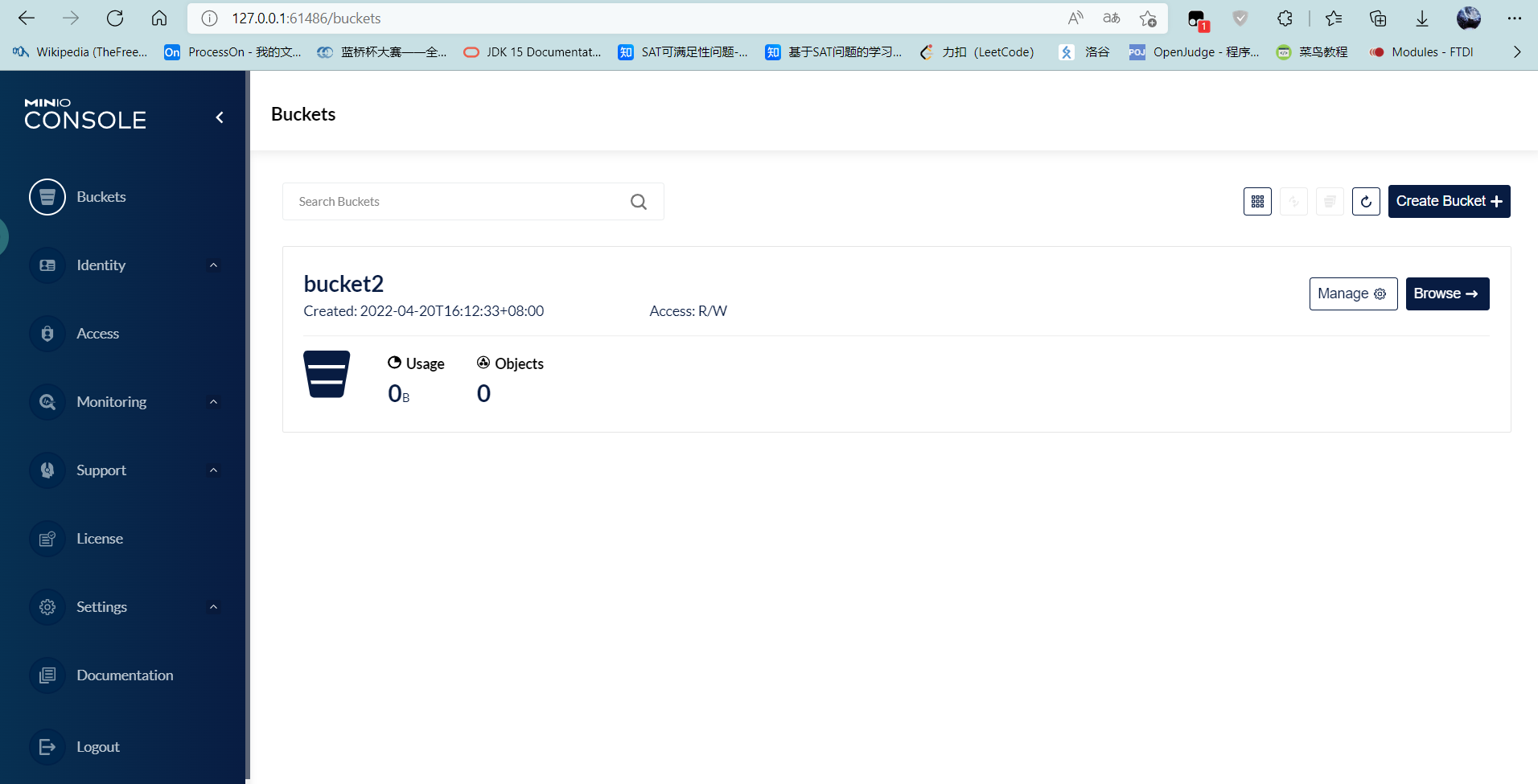
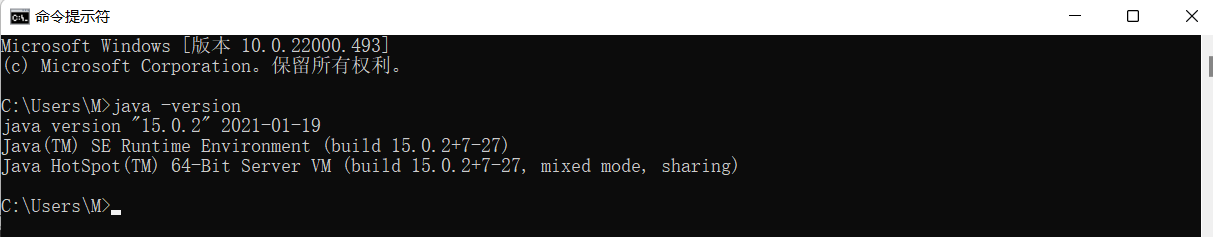


图1.4 进入Minio网页

可以在其中点击Create Buckets创建新的bucket并进行文件的上传和下载。

2）安装COSBench

首先下载的是release文件夹下的0.4.2版本的cosbench，但是Java版本是15.0.2如图1.5所示，然后启动过程中点击start-all.bat出现了很多的报错，由于在网上搜集资料显示是Java 的版本太高出现的映射警告，可以忽略，所以我选择直接打开COSBench，结果显示127.0.1拒绝访问，然后我搜集了很多解决127.0.1拒绝访问的办法，甚至改变system32中关于网络的配置文件，仍然不能解决问题；然后还下载了更低版本的0.4.2.C4的COSBench文件，结果还是不能正确启动。于是我认真查看COSBench启动过程中出现的报错日志，发现很多错误和之前Java学习中出现的报错很像，于是我下载了java1.7发现安装完成后，COSBench能够正常启动，如图1.6所示。



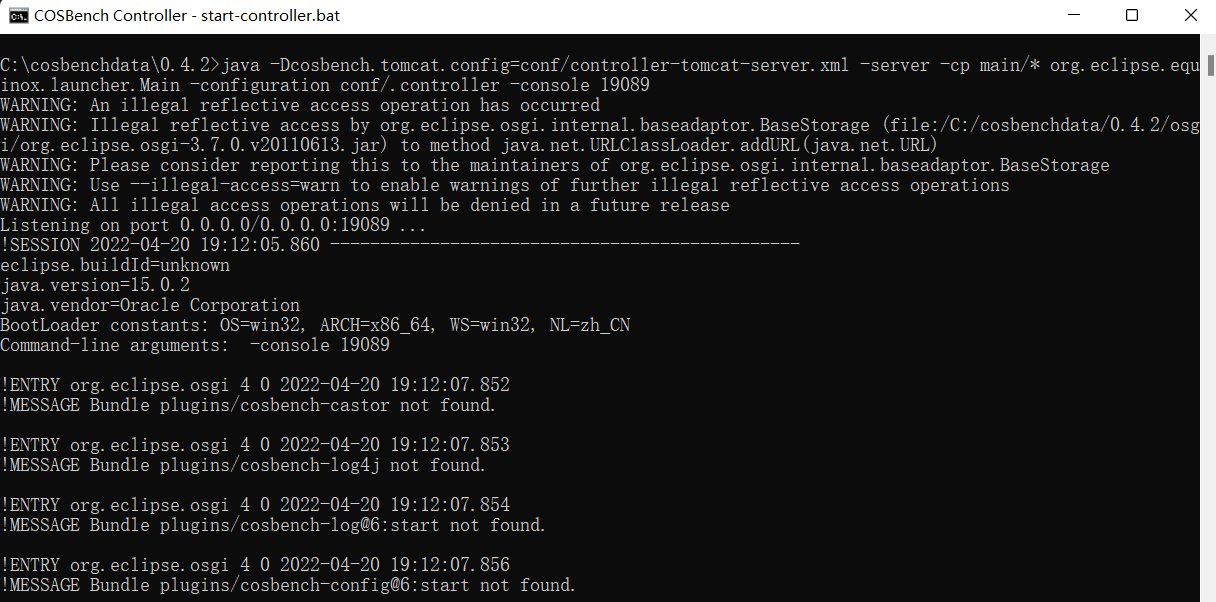
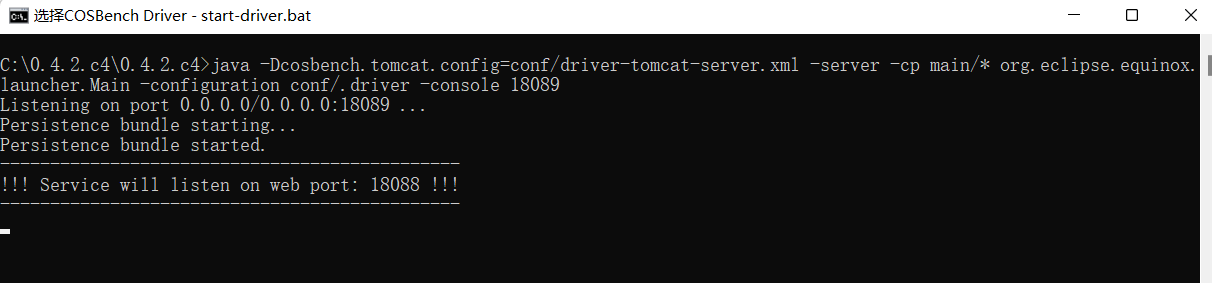


图1.5 java版本太高报错



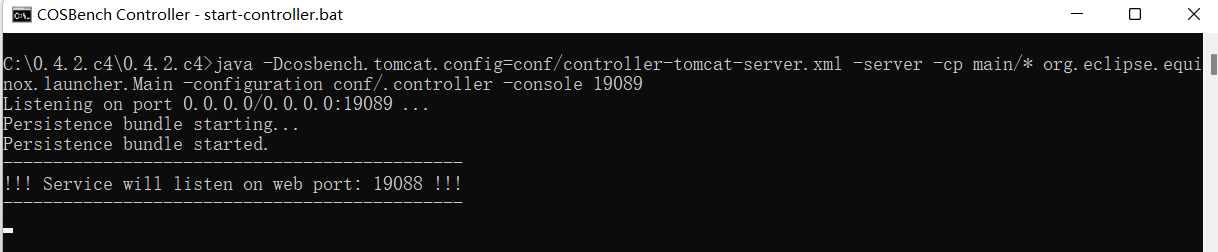


图1.6 成功进入COSBench

然后在浏览器中输入http://127.0.0.1:19088/controller/进入COSBench的controller,点击submit new workloads，如图1.7所示。

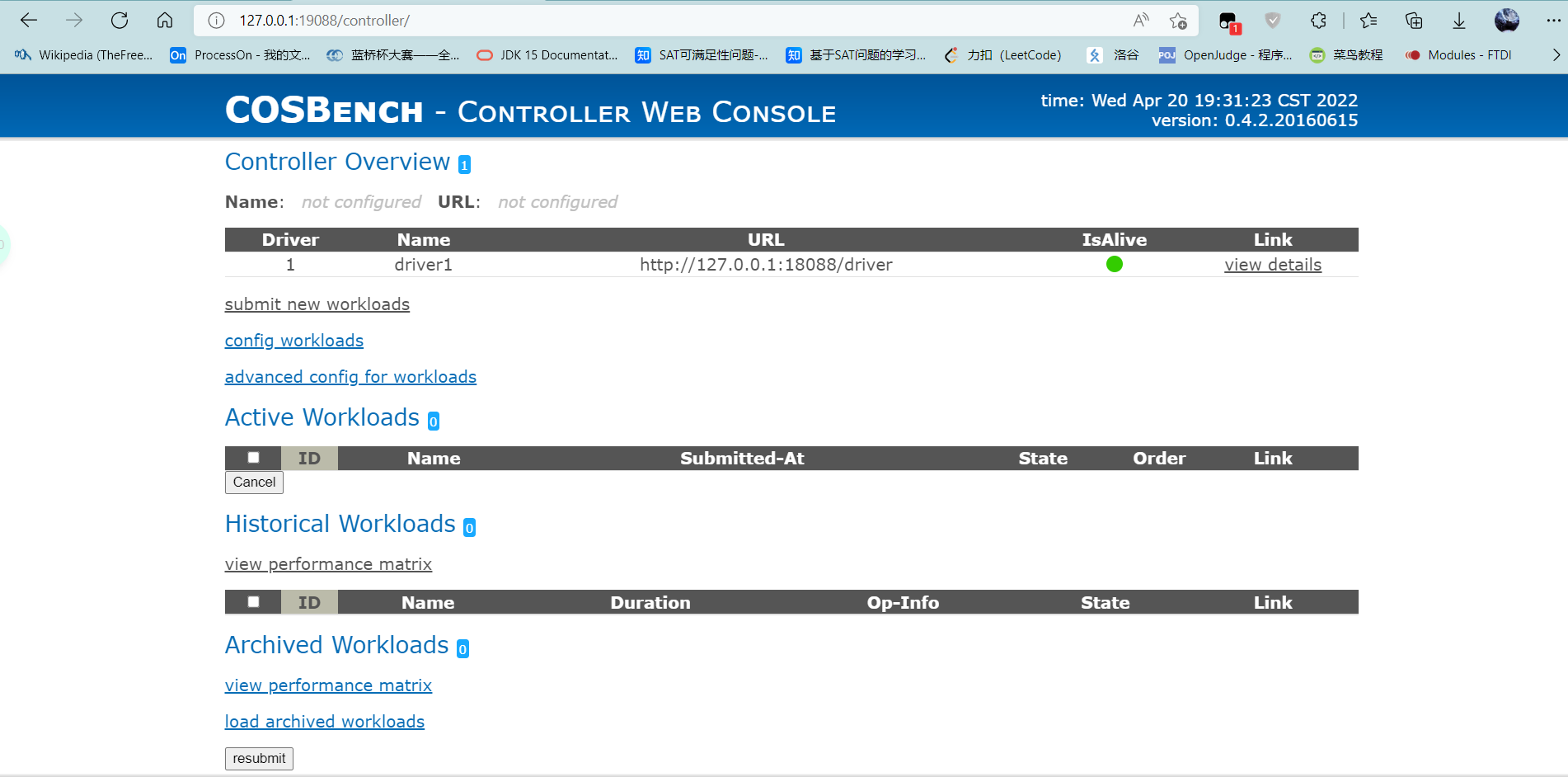


图1.7 COSBench的submit new workloads

点击选择文件，选择obs-tutorial-master中的workload-example.xml文件，选择submit最终点击view details查看细节，实验结果如图1.8、1.9所示。

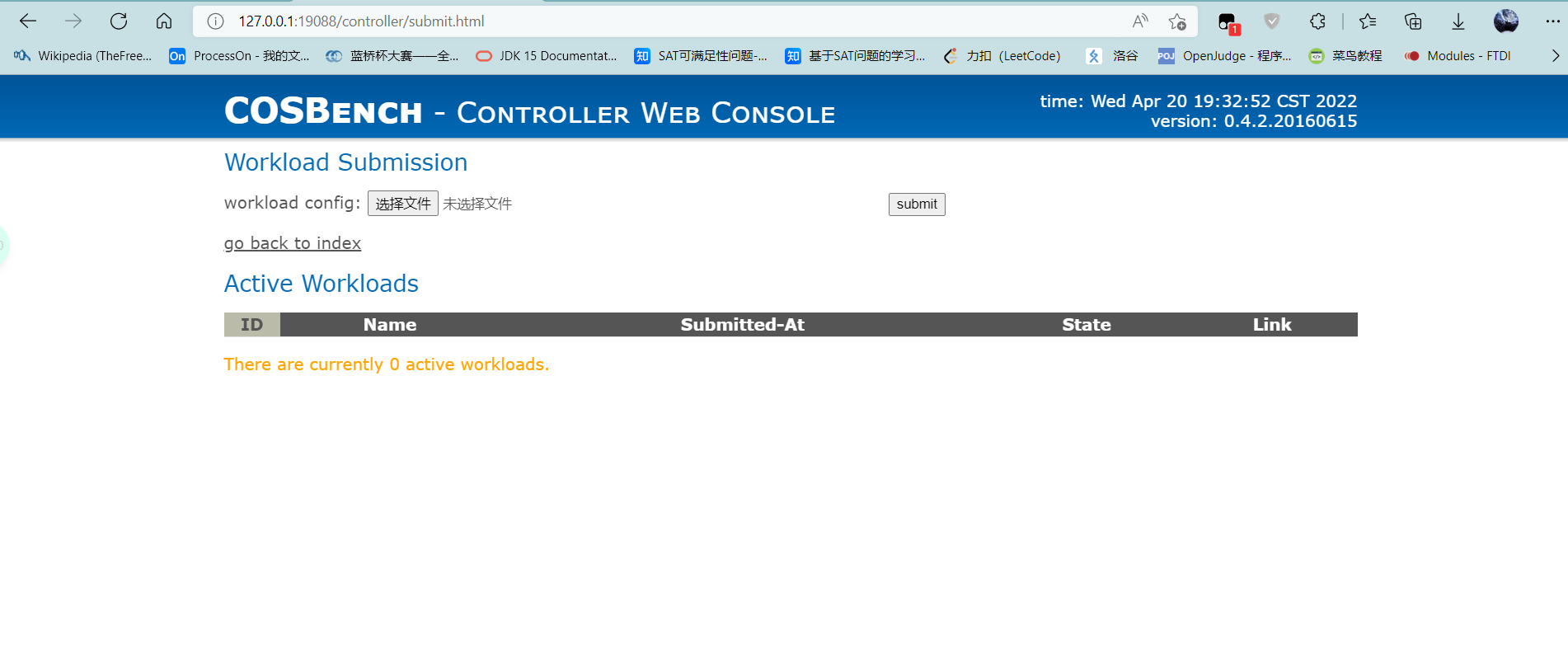


图1.8 submit文件

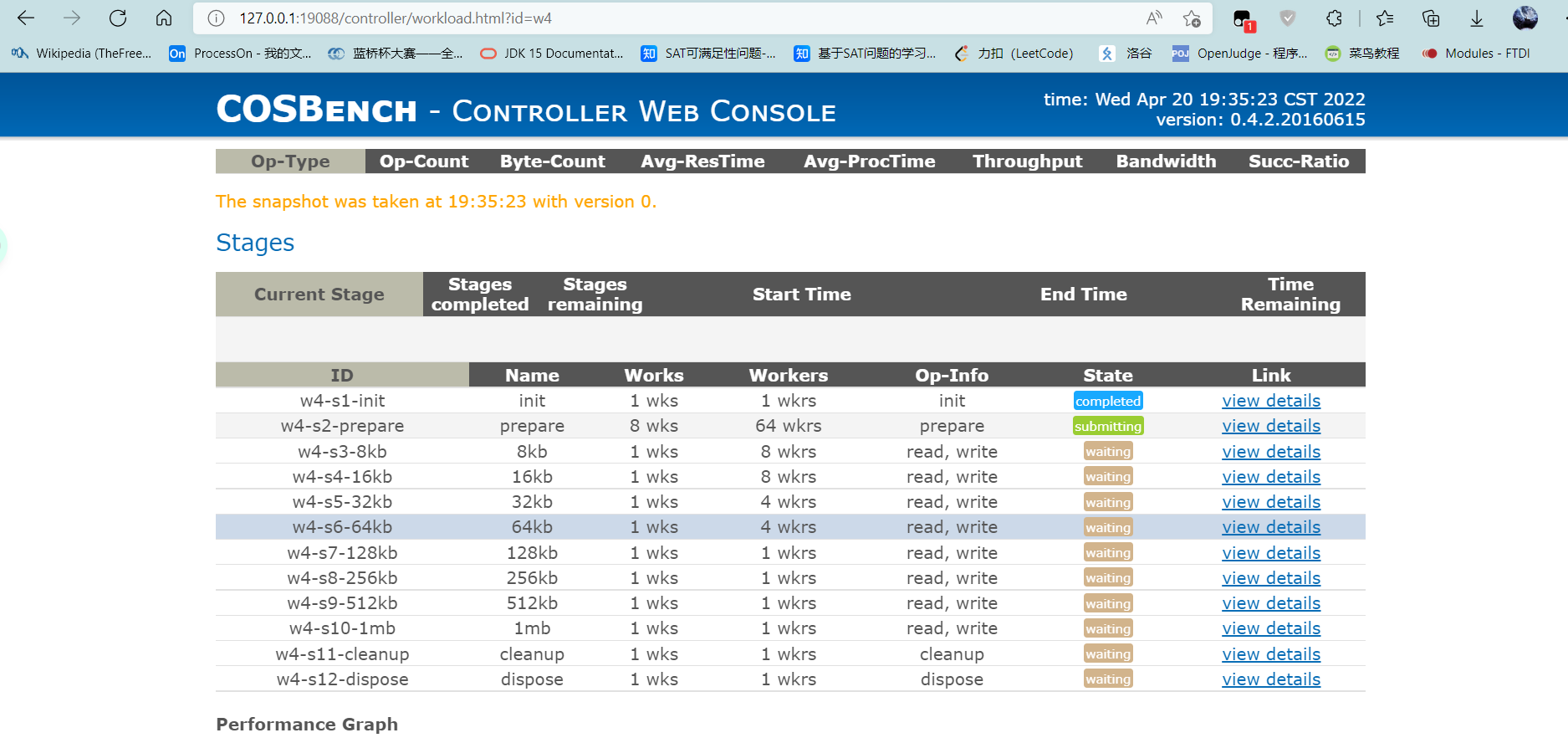


图1.9 view detail

1. **结果分析**

监视所有的文件传输过程，最终结果如图1.10、1.11、1.12所示。

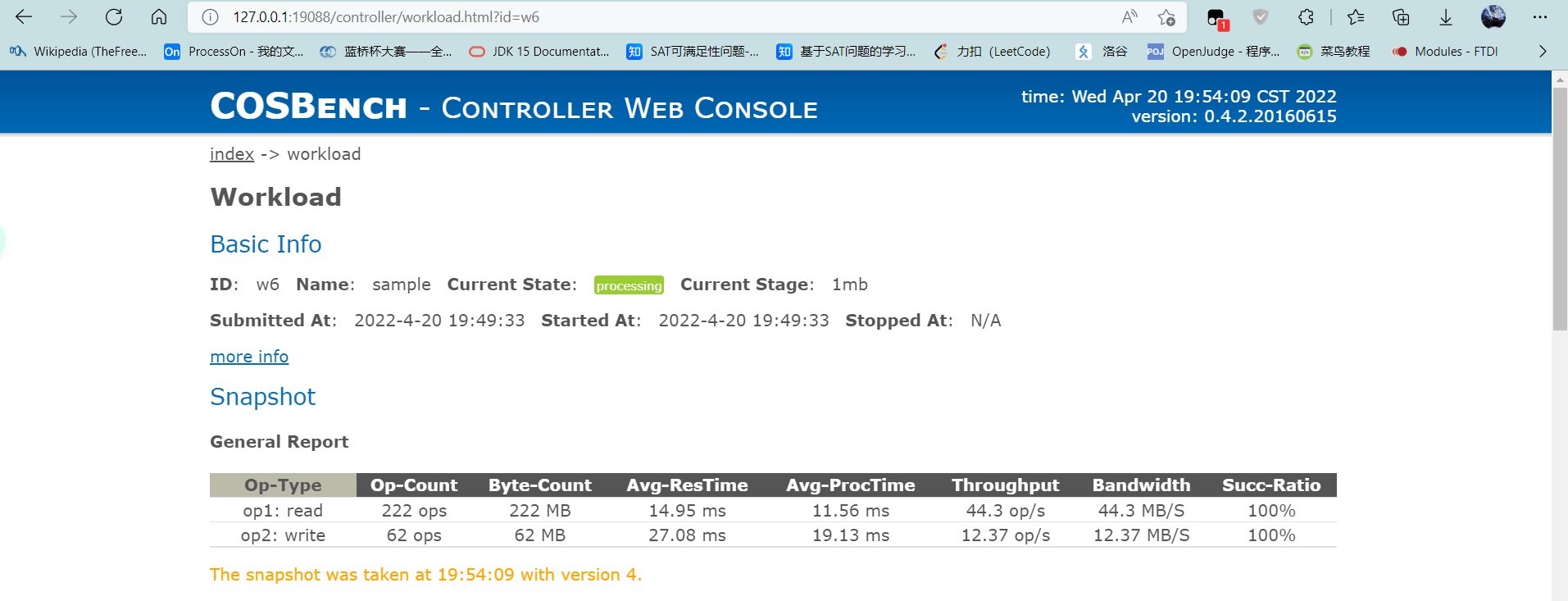
****

图1.10 监视结果图1

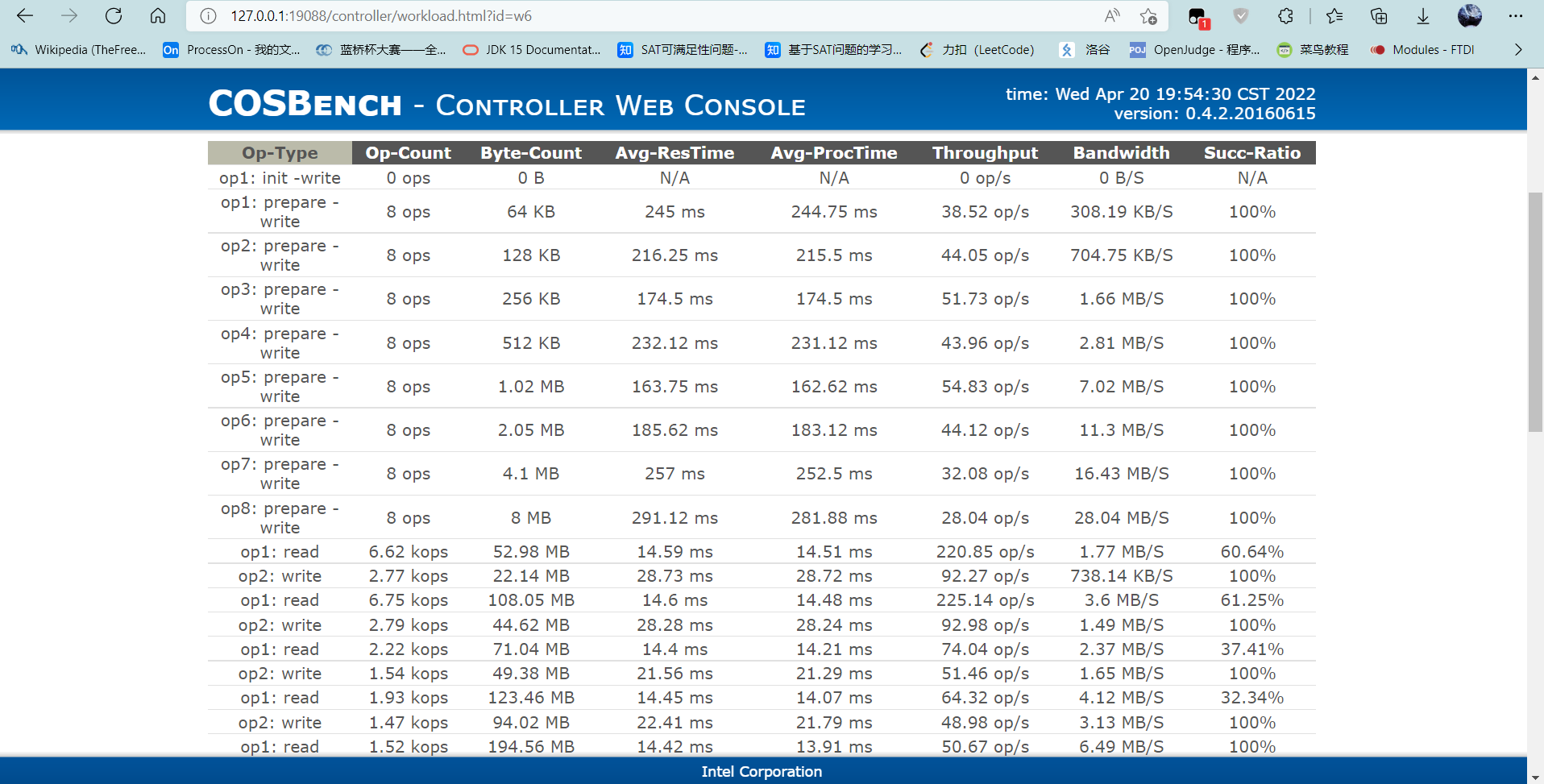
****

图1.11 监视结果图2

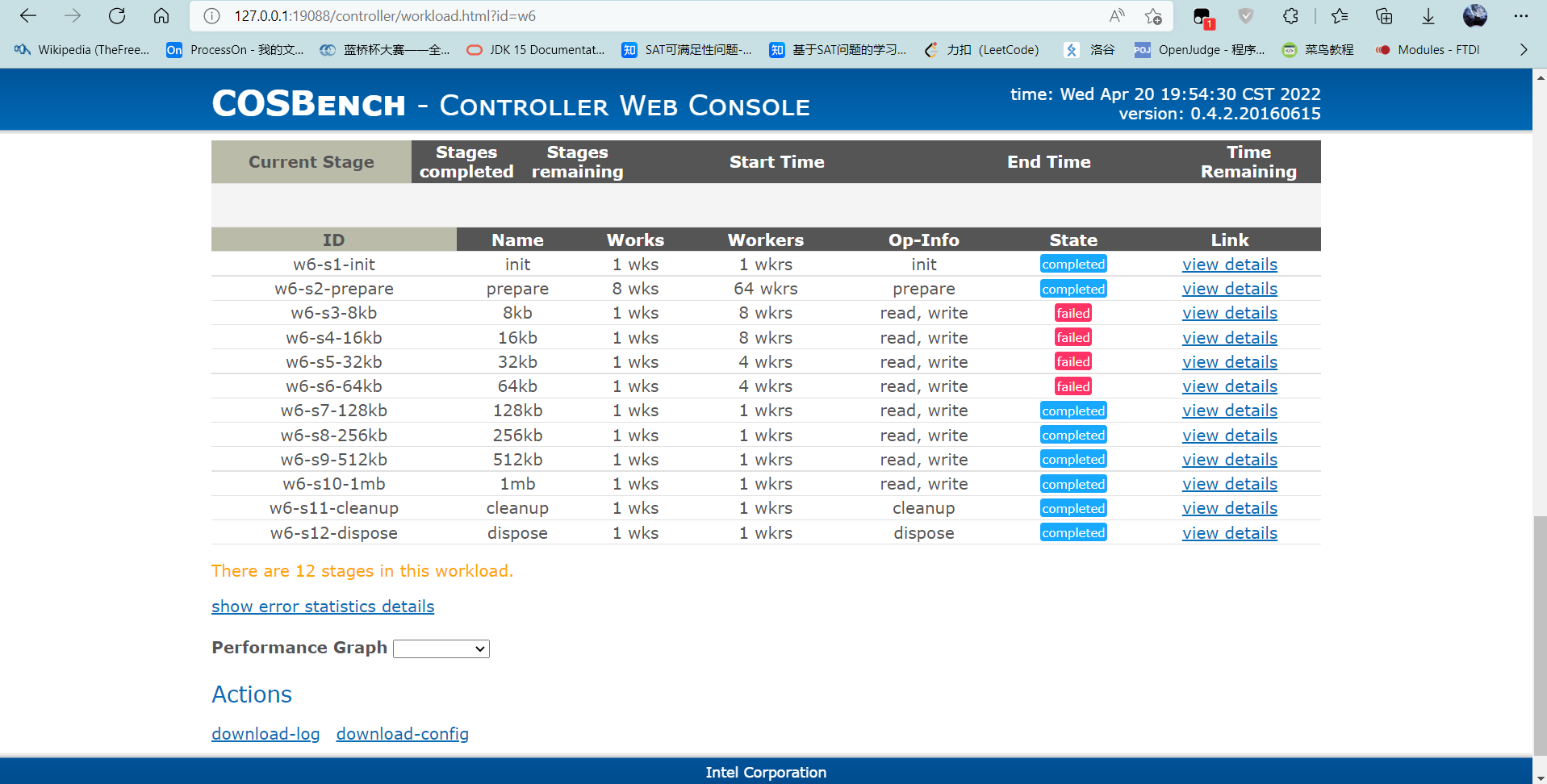
****

图1.12 监视结果图3

发现在采用Minio作为服务端时，读和写的过程中有的时候成功率可以为100%，有的时候会失败，不为100%的情况存在于size较小的情况下，随着size的增大读写成功率均为100%。

**实验中注意的小问题：**

在测试对象存储性能前将workload-example.xml用记事本打开，修改其中的用户名及密码。

1. **实验总结**

本次实验是学习如何搭建对象存储以及在搭建好的环境下进行对象存储性能的测试实验。首先对我来说是加强了代码管理工具gitee的使用熟练度，其次通过搭建Minio客户端和服务器进行文件的上传和下载并在此基础上通过COSbench测试读写效率，也算是加深了对象存储的学习。并且在系统环境搭建过程中遇到了很多的问题，大多数是由于自己的粗心导致的，吃一堑长一智以后会更加小心；另外在解决问题的过程中我也增长了自身解决问题的能力。

总的来说本次实验是比较成功的，在实验过程中也学到了很多的东西，对大数据的存储也有了更加宽广的见识。