

**课程设计报告**

**课程名称： 物联网数据存储与管理作业**

**专业班级： 物联网1501**

**学 号： U201514888**

**姓 名： 陈艺欣**

**指导教师： 施展**

**报告日期： 2018.4.28**

**计算机科学与技术学院**

目录

[1 障碍物传感器实验 3](#_Toc512553326)

[1.1 实验目的 3](#_Toc512553327)

[1.2 实验原理 3](#_Toc512553328)

[1.3 实验步骤 3](#_Toc512553329)

[1.4 实验中遇到的问题及分析 4](#_Toc512553330)

[1.5 代码注释 4](#_Toc512553331)

[1.6 实验结果 5](#_Toc512553332)

[1.7 应用 6](#_Toc512553333)

## 基础环境

### Git和Github

### 

图 1 Git版本

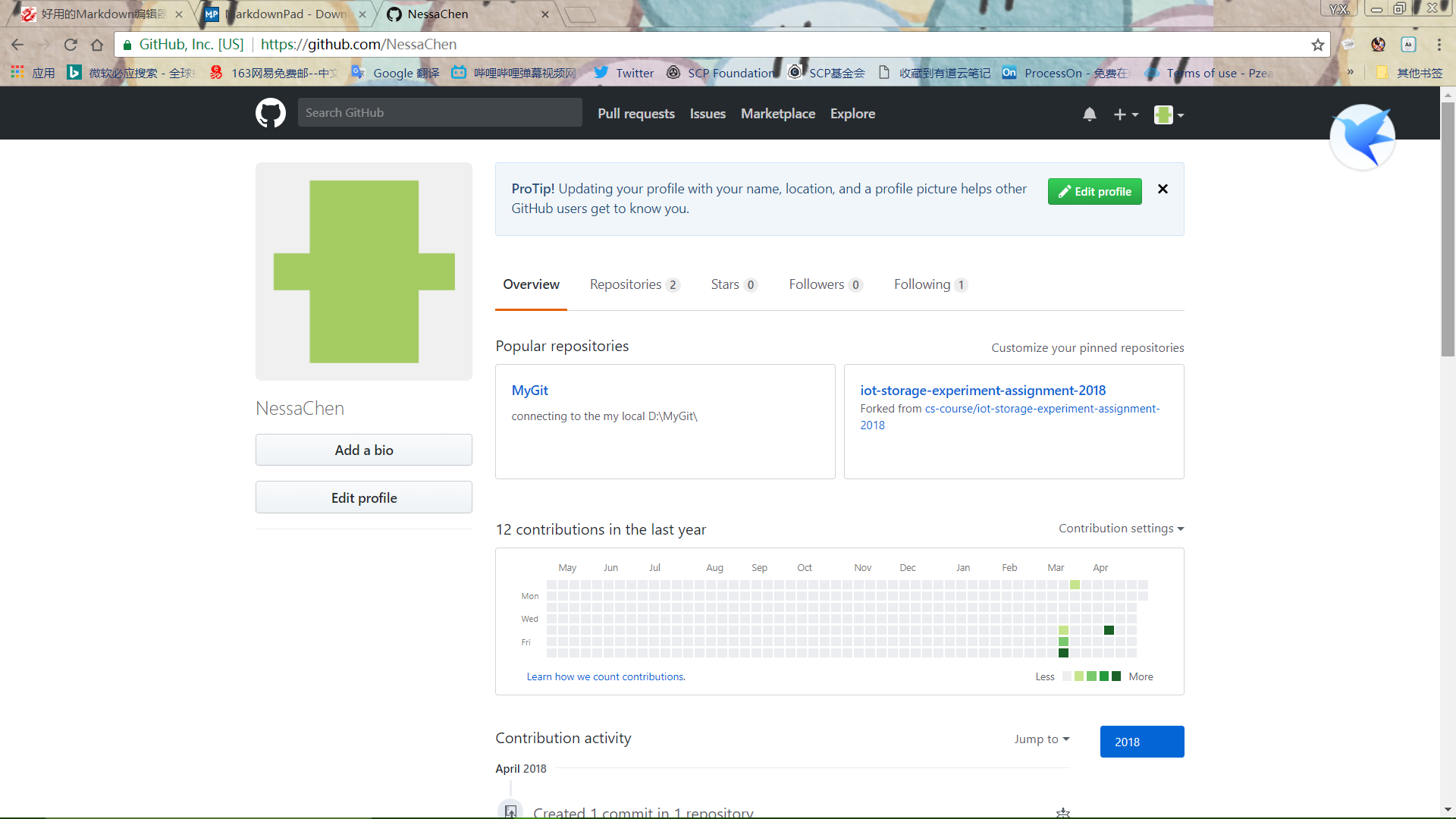


图 2 Github账号

### Python

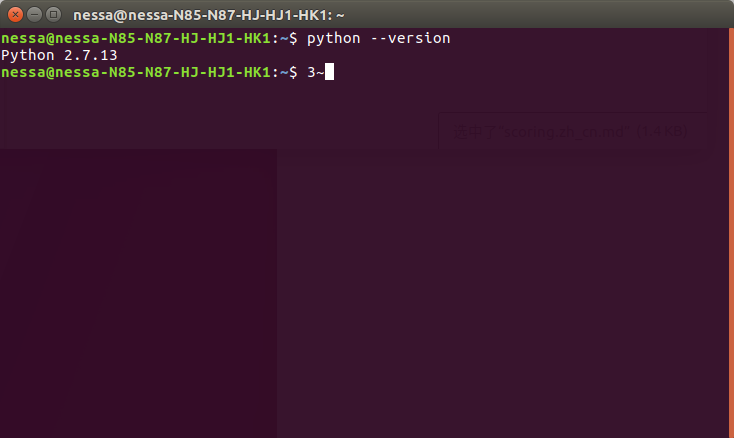


图 3 Python版本

### Java

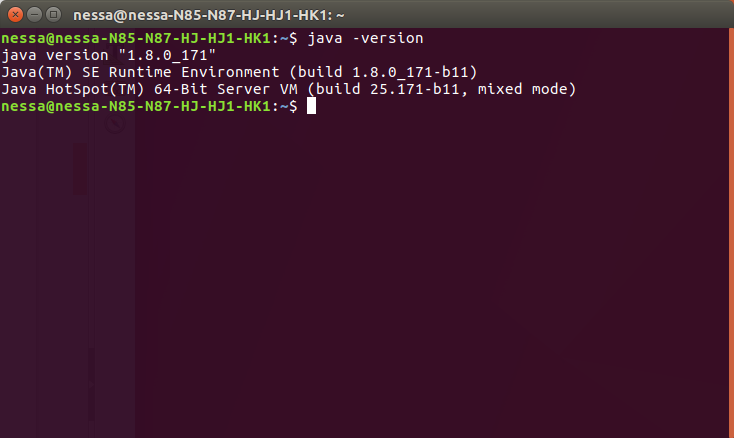


图 4 Java版本

### 实验内容

|  |  |  |
| --- | --- | --- |
| Server | Minio Server | s3proxy |
| Client | Minio Client | aws-shell |
| Benchmark | cosbench | cosbench |

## Minio

### Server

从Minion官网下载，以/data文件夹作为服务器位置，通过./minio命令运行。

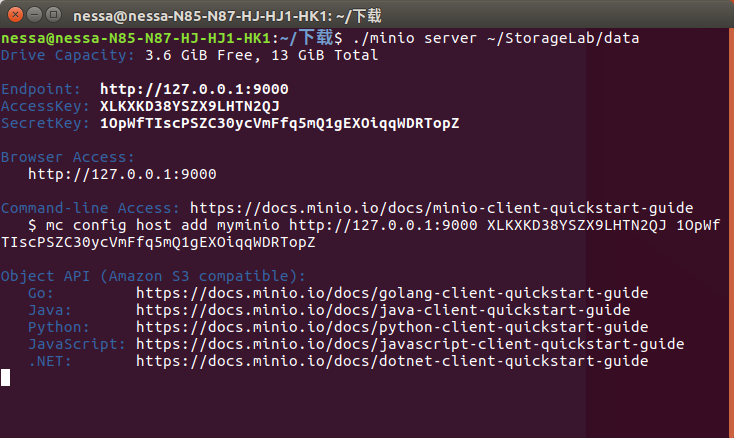


图 5 Minio运行终端输出

运行成功后点击终端显示的地址，打开Minio Browser

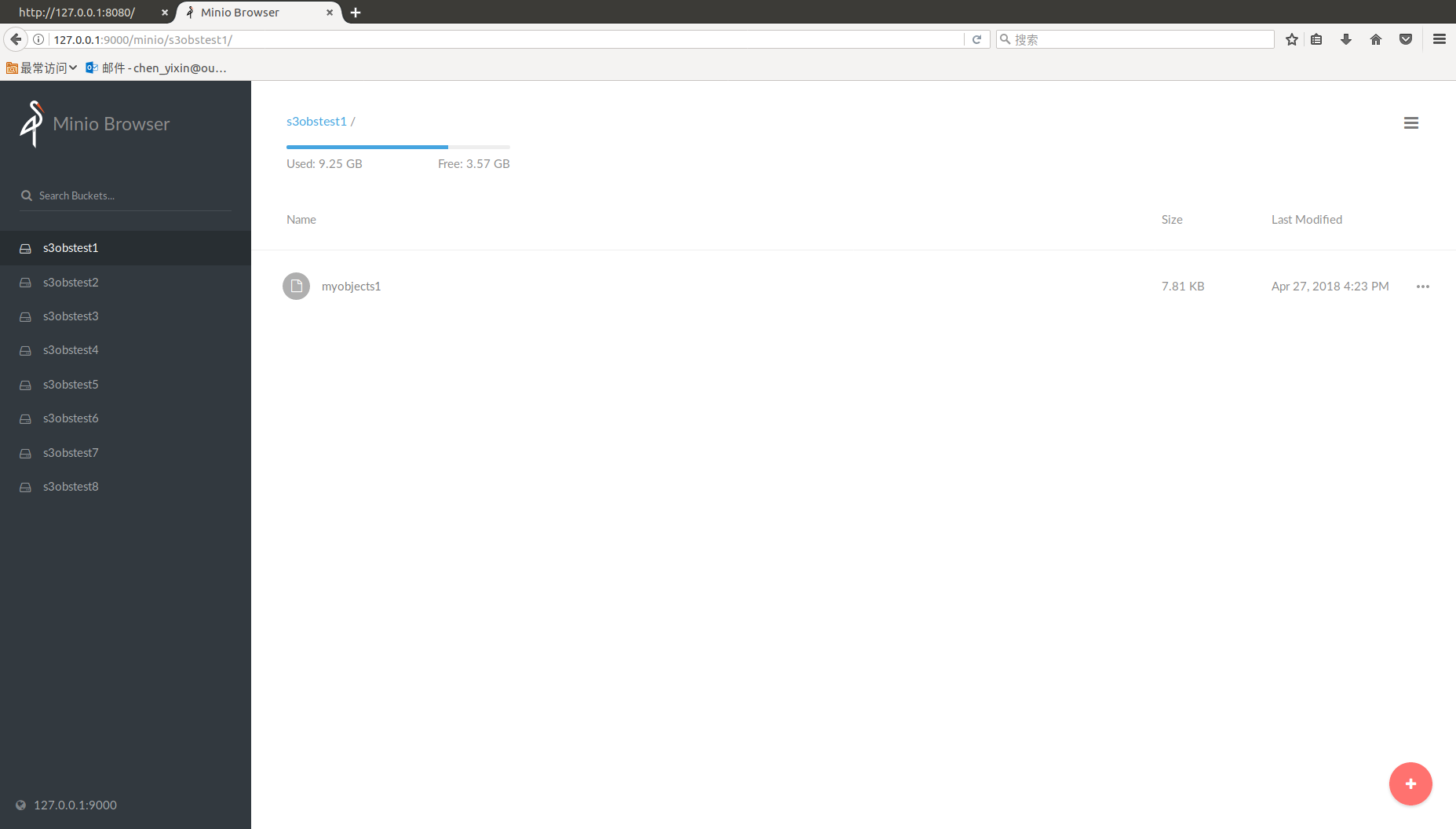


图 6 Minio Browser

### Client

从官网下载minio client，通过./mc命令运行。先添加minio host，然后通过ls命令访问host。

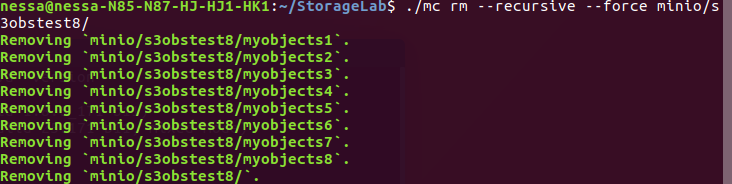
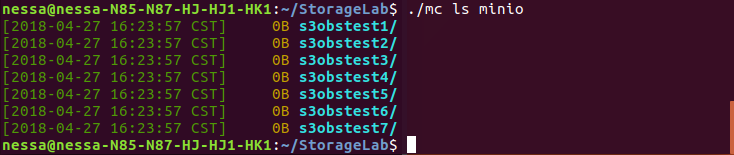


图 7 minio client操作

### cosbench

通过Docker运行cosbench，出现如下输出，表示cosbench运行成功。

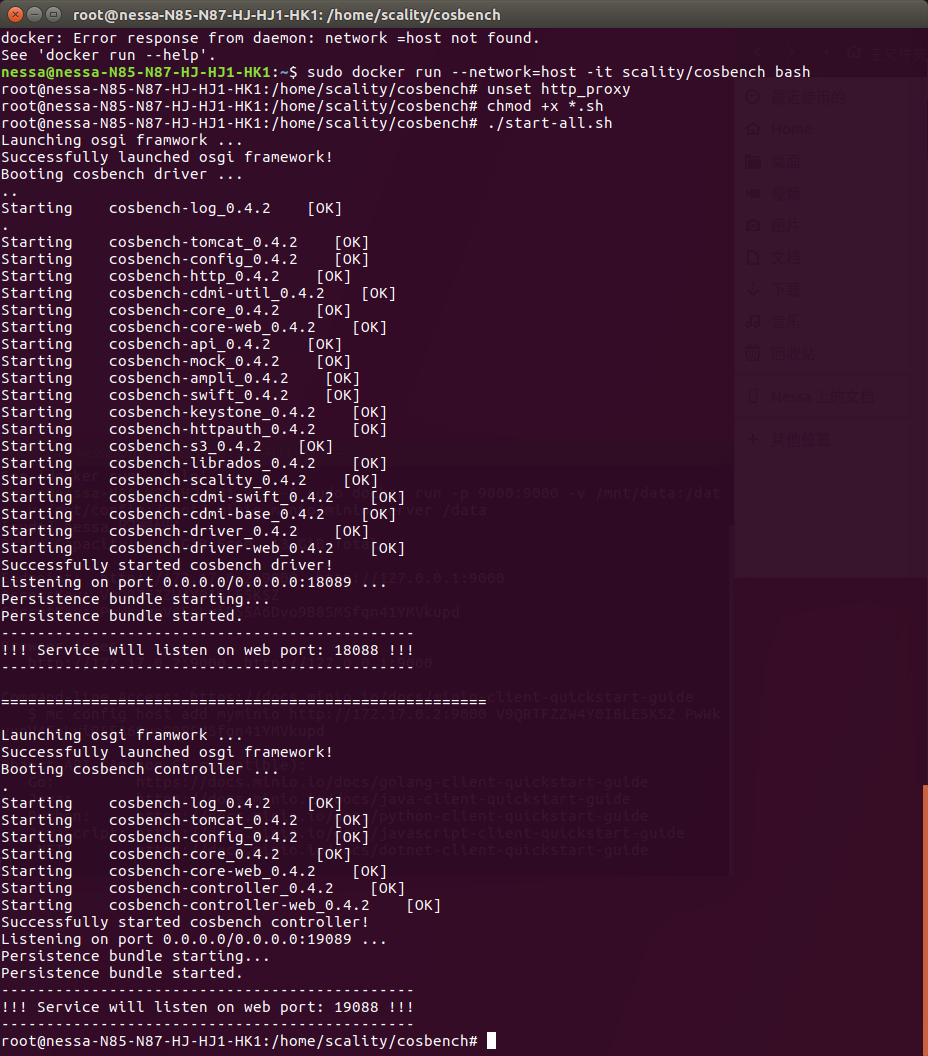


图 8 cosbench终端输出

打开<http://127.0.0.1:19088/controller/index.html>。

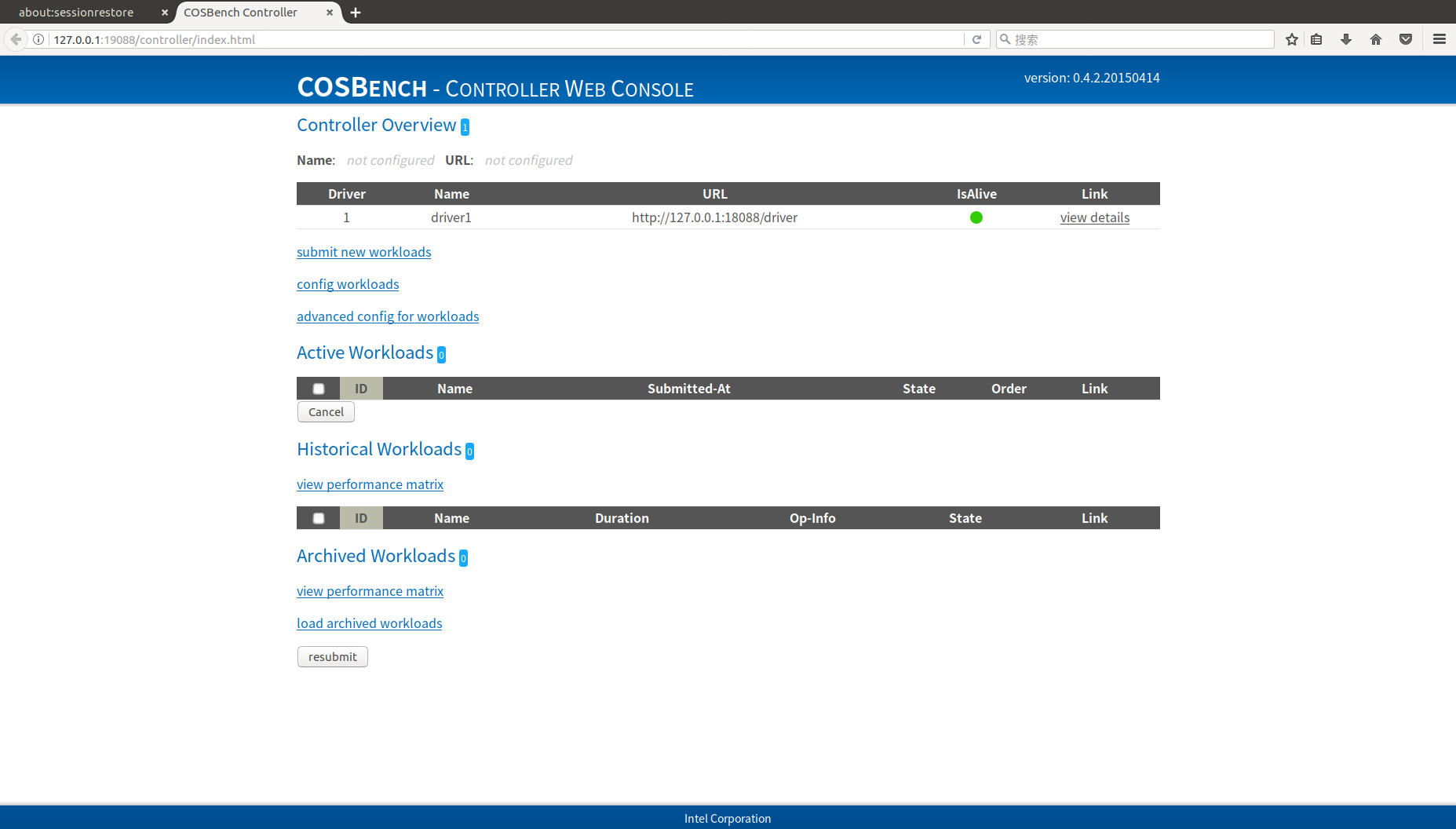


图 9 Cosbench Controller Web Console

编辑workloads文件。

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  **<workload** name="sample" description="sample benchmark"**>**  **<storage** type="s3" config="accesskey=V9QRTFZZW4Y0I8LESKSZ;secretkey=PwWkyupV4EoLviR55A6Dvo9B8SMSfqn41YMVkupd;proxyhost=9000;endpoint=http://127.0.0.1:9000" **/>**  **<workflow>**  **<workstage** name="init"**>**  **<work** type="init" workers="1" config="cprefix=s3test;containers=r(1,8)" **/>**  **</workstage>**  **<workstage** name="prepare"**>**  **<work** type="prepare" workers="8" config="cprefix=s3test;containers=r(1,1);objects=r(1,8);sizes=c(8)KB" **/>**  **<work** type="prepare" workers="8" config="cprefix=s3test;containers=r(2,2);objects=r(1,8);sizes=c(16)KB" **/>**  **<work** type="prepare" workers="8" config="cprefix=s3test;containers=r(3,3);objects=r(1,8);sizes=c(32)KB" **/>**  **<work** type="prepare" workers="8" config="cprefix=s3test;containers=r(4,4);objects=r(1,8);sizes=c(64)KB" **/>**  **<work** type="prepare" workers="8" config="cprefix=s3test;containers=r(5,5);objects=r(1,8);sizes=c(128)KB" **/>**  **<work** type="prepare" workers="8" config="cprefix=s3test;containers=r(6,6);objects=r(1,8);sizes=c(256)KB" **/>**  **<work** type="prepare" workers="8" config="cprefix=s3test;containers=r(7,7);objects=r(1,8);sizes=c(512)KB" **/>**  **<work** type="prepare" workers="8" config="cprefix=s3test;containers=r(8,8);objects=r(1,8);sizes=c(1)MB" **/>**  **</workstage>**  **<workstage** name="8kb"**>**  **<work** name="8kb" workers="8" runtime="30"**>**  **<operation** type="read" ratio="80" config="cprefix=s3test;containers=c(1);objects=u(1,8)" **/>**  **<operation** type="write" ratio="20" config="cprefix=s3test;containers=c(1);objects=u(9,16);sizes=c(8)KB" **/>**  **</work>**  **</workstage>**  **<workstage** name="16kb"**>**  **<work** name="16kb" workers="8" runtime="30"**>**  **<operation** type="read" ratio="80" config="cprefix=s3test;containers=c(2);objects=u(1,8)" **/>**  **<operation** type="write" ratio="20" config="cprefix=s3test;containers=c(1);objects=u(9,16);sizes=c(16)KB" **/>**  **</work>**  **</workstage>**  **<workstage** name="32kb"**>**  **<work** name="32kb" workers="4" runtime="30"**>**  **<operation** type="read" ratio="80" config="cprefix=s3test;containers=c(3);objects=u(1,8)" **/>**  **<operation** type="write" ratio="20" config="cprefix=s3test;containers=c(1);objects=u(9,16);sizes=c(32)KB" **/>**  **</work>**  **</workstage>**  **<workstage** name="64kb"**>**  **<work** name="64kb" workers="4" runtime="30"**>**  **<operation** type="read" ratio="80" config="cprefix=s3test;containers=c(4);objects=u(1,8)" **/>**  **<operation** type="write" ratio="20" config="cprefix=s3test;containers=c(1);objects=u(9,16);sizes=c(64)KB" **/>**  **</work>**  **</workstage>**  **<workstage** name="128kb"**>**  **<work** name="128kb" workers="1" runtime="30"**>**  **<operation** type="read" ratio="80" config="cprefix=s3test;containers=c(5);objects=u(1,8)" **/>**  **<operation** type="write" ratio="20" config="cprefix=s3test;containers=c(1);objects=u(9,16);sizes=c(128)KB" **/>**  **</work>**  **</workstage>**  **<workstage** name="256kb"**>**  **<work** name="256kb" workers="1" runtime="30"**>**  **<operation** type="read" ratio="80" config="cprefix=s3test;containers=c(6);objects=u(1,8)" **/>**  **<operation** type="write" ratio="20" config="cprefix=s3test;containers=c(1);objects=u(9,16);sizes=c(256)KB" **/>**  **</work>**  **</workstage>**  **<workstage** name="512kb"**>**  **<work** name="512kb" workers="1" runtime="30"**>**  **<operation** type="read" ratio="80" config="cprefix=s3test;containers=c(7);objects=u(1,8)" **/>**  **<operation** type="write" ratio="20" config="cprefix=s3test;containers=c(1);objects=u(9,16);sizes=c(512)KB" **/>**  **</work>**  **</workstage>**  **<workstage** name="1mb"**>**  **<work** name="1mb" workers="1" runtime="30"**>**  **<operation** type="read" ratio="80" config="cprefix=s3test;containers=c(8);objects=u(1,8)" **/>**  **<operation** type="write" ratio="20" config="cprefix=s3test;containers=c(1);objects=u(9,16);sizes=c(1)MB" **/>**  **</work>**  **</workstage>**  **<workstage** name="cleanup"**>**  **<work** type="cleanup" workers="1" config="cprefix=s3test;containers=r(1,8);objects=r(1,16)" **/>**  **</workstage>**  **<workstage** name="dispose"**>**  **<work** type="dispose" workers="1" config="cprefix=s3test;containers=r(1,8)" **/>**  **</workstage>**  **</workflow>**  **</workload>** |

在Cosbench Controller Web Console中，点击submit上传上述workload文件。

这里我的Cosbench出了问题，所有操作全部失败报错“fail to perform operation”，如图。

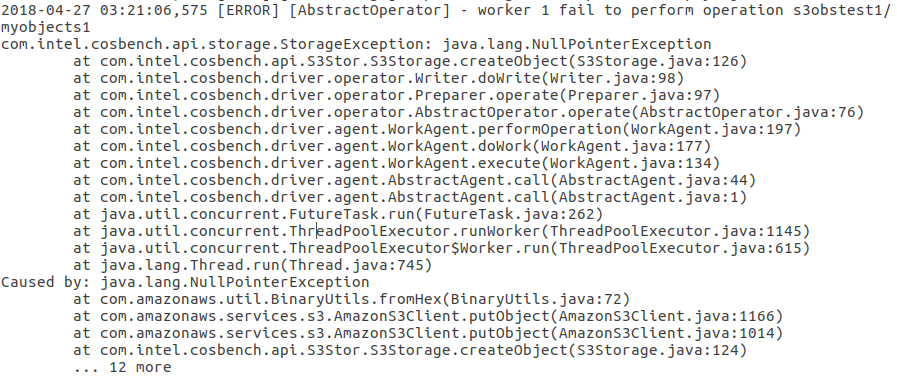


图 10 Cosbench报错信息

但是object文件和bucket都被成功创建了。

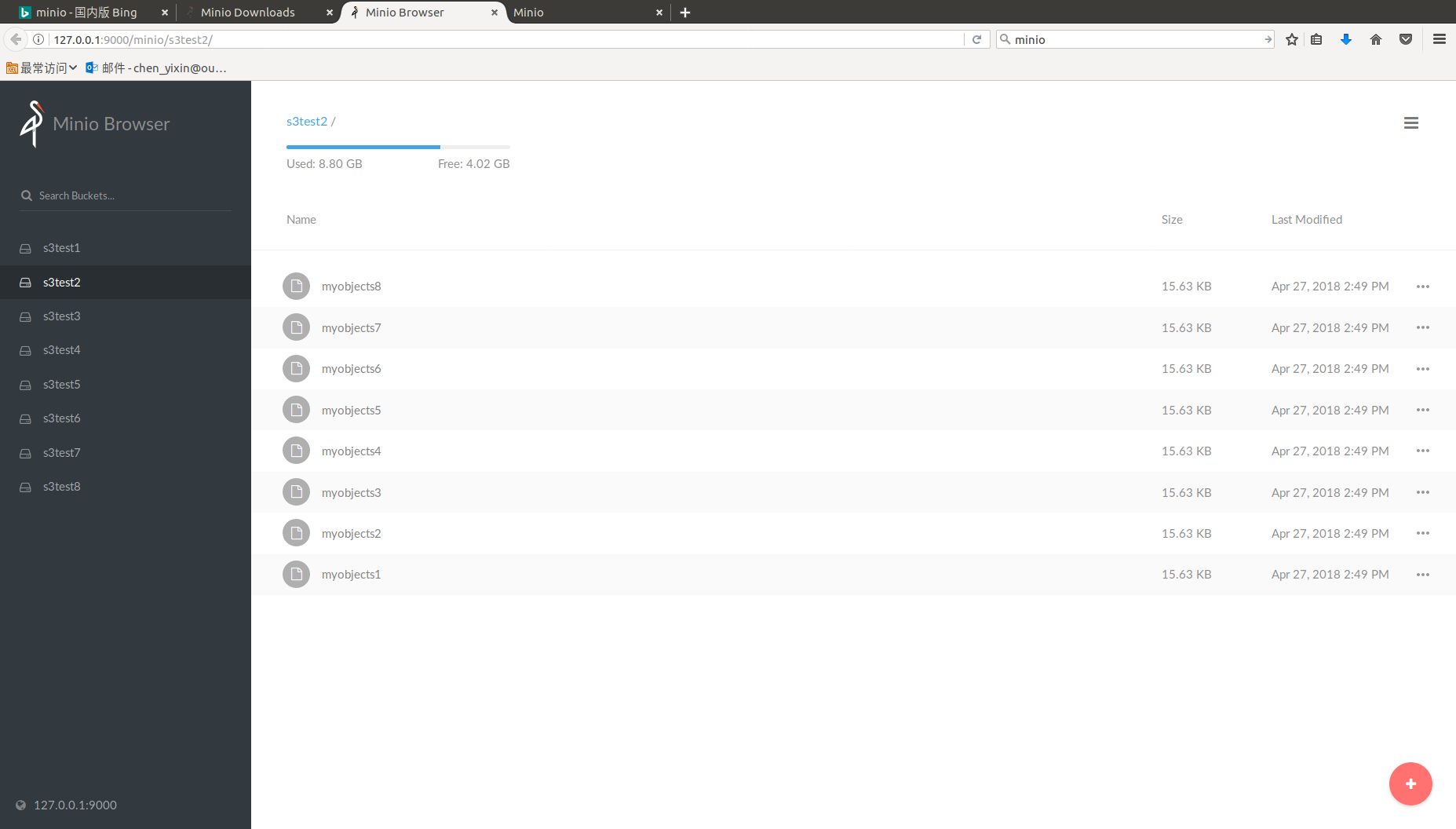


图 11 跑Cosbench时Minio Browser的显示

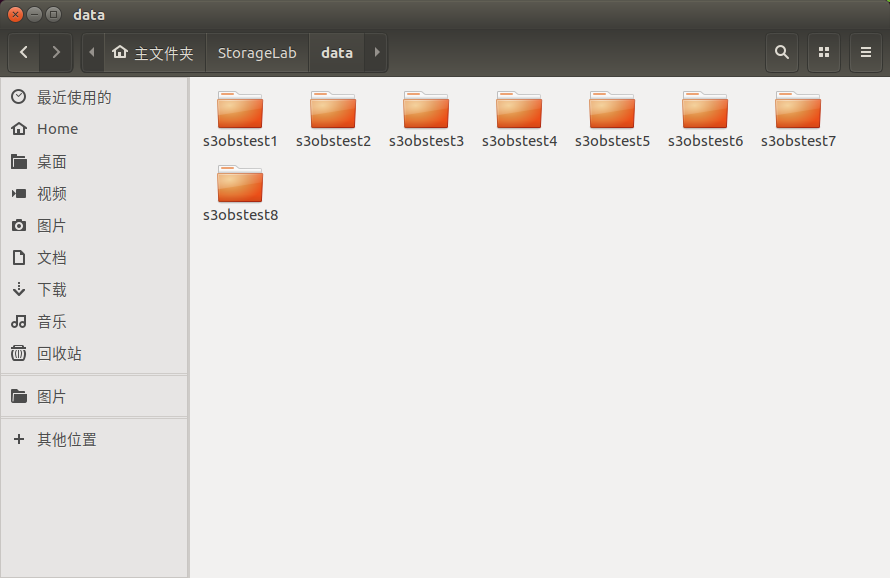


图 12 跑Cosbench时本地文件的显示

查阅资料无果（最后也没弄出来真的实在没办法了）。在老师的建议下将server改成s3proxy。

## s3proxy

### Server

在网上下载realease版s3proxy，编辑配置文件s3proxy.conf，通过指令./s3proxy –properties s3proxy.conf运行。

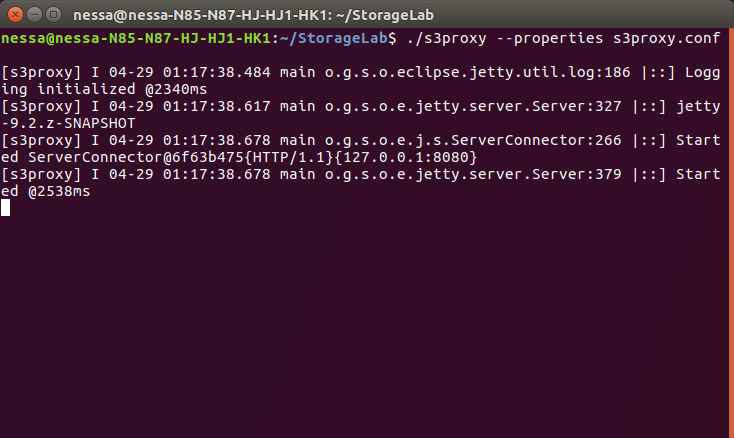


图 13 s3proxy运行终端显示

运行成功后访问<http://127.0.0.1:8080>。

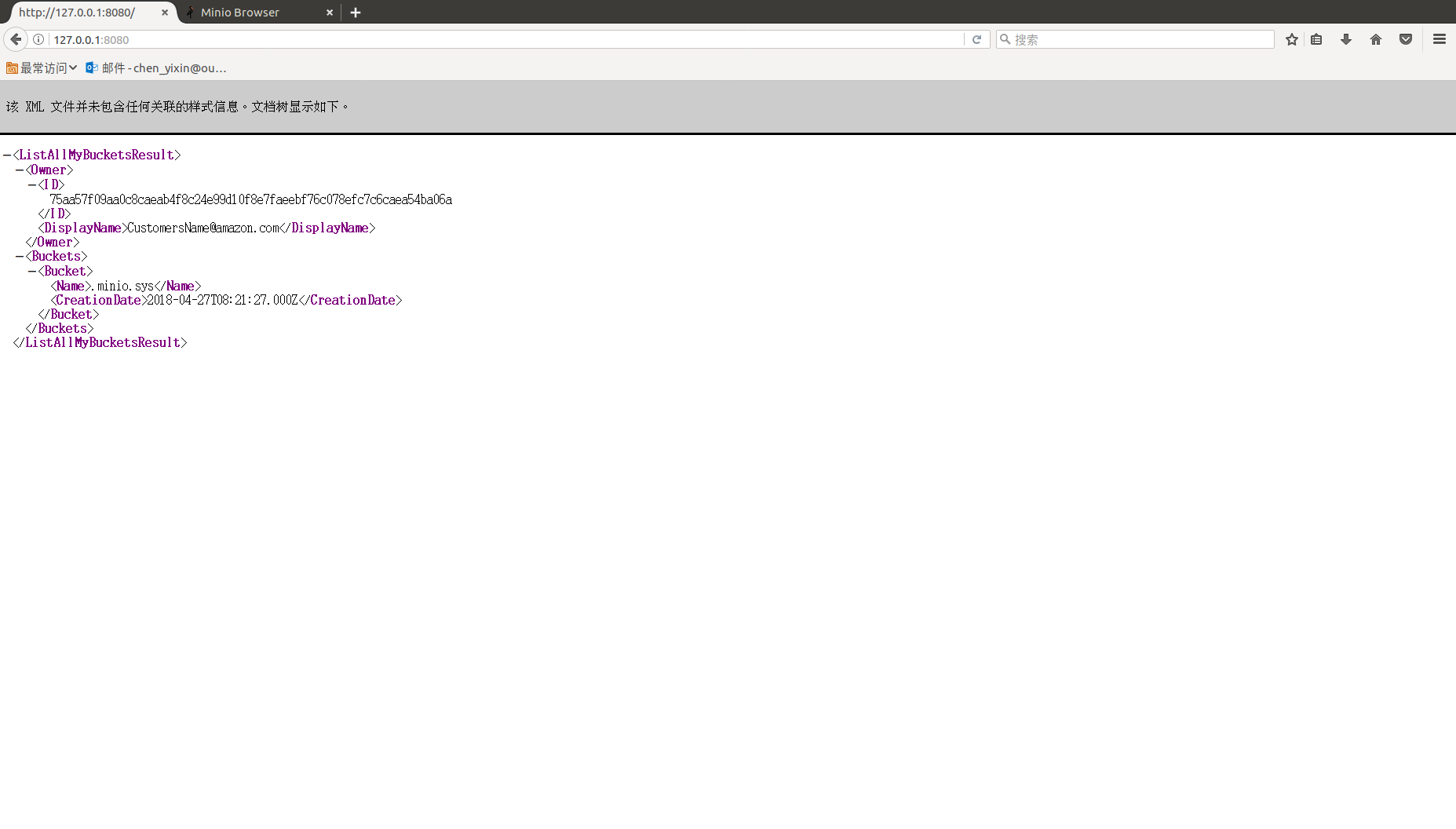


图 14 s3proxy网页

### Client（aws-shell）

通过pip install aws-shell安装aws-shell。之后运行aws configure命令配置链接，并通过aws configure set default.s3.signature\_version s3v4配置s3链接类型。

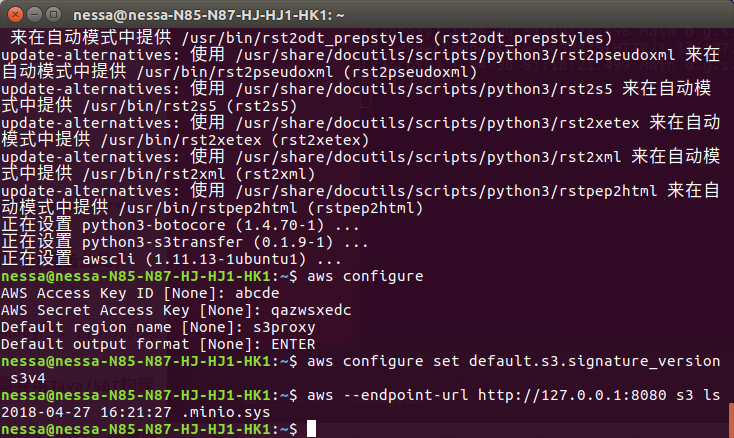


图 15 aws configure

接下来通过aws –endpoint-url <http://127.0.0.1:8080> s3 ls访问server。

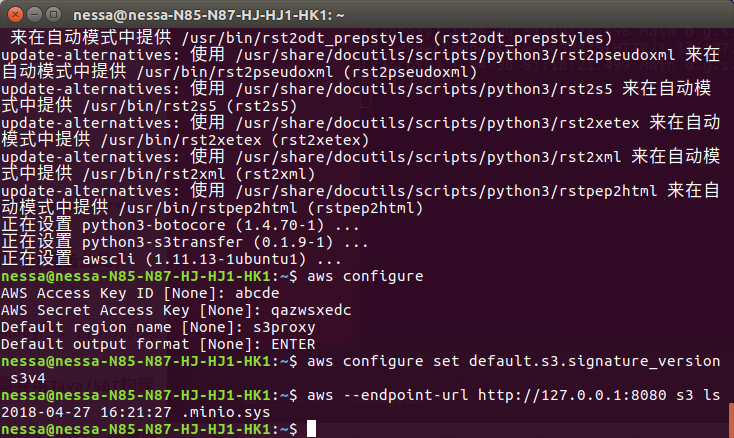


图 16 访问s3proxy server

### cosbench

修改workload文件，将以下部分

|  |
| --- |
| **<storage** type="s3" config="accesskey=V9QRTFZZW4Y0I8LESKSZ;secretkey=PwWkyupV4EoLviR55A6Dvo9B8SMSfqn41YMVkupd;proxyhost=9000;endpoint=http://127.0.0.1:9000" **/>** |

修改为

|  |
| --- |
| **<storage** type="s3" config="accesskey=abcde;secretkey=qazwsxedc;endpoint=http://127.0.0.1:8080" **/>** |

并通过Cosbench Controller Web Console上传测试文件。Cosbench成功运行。

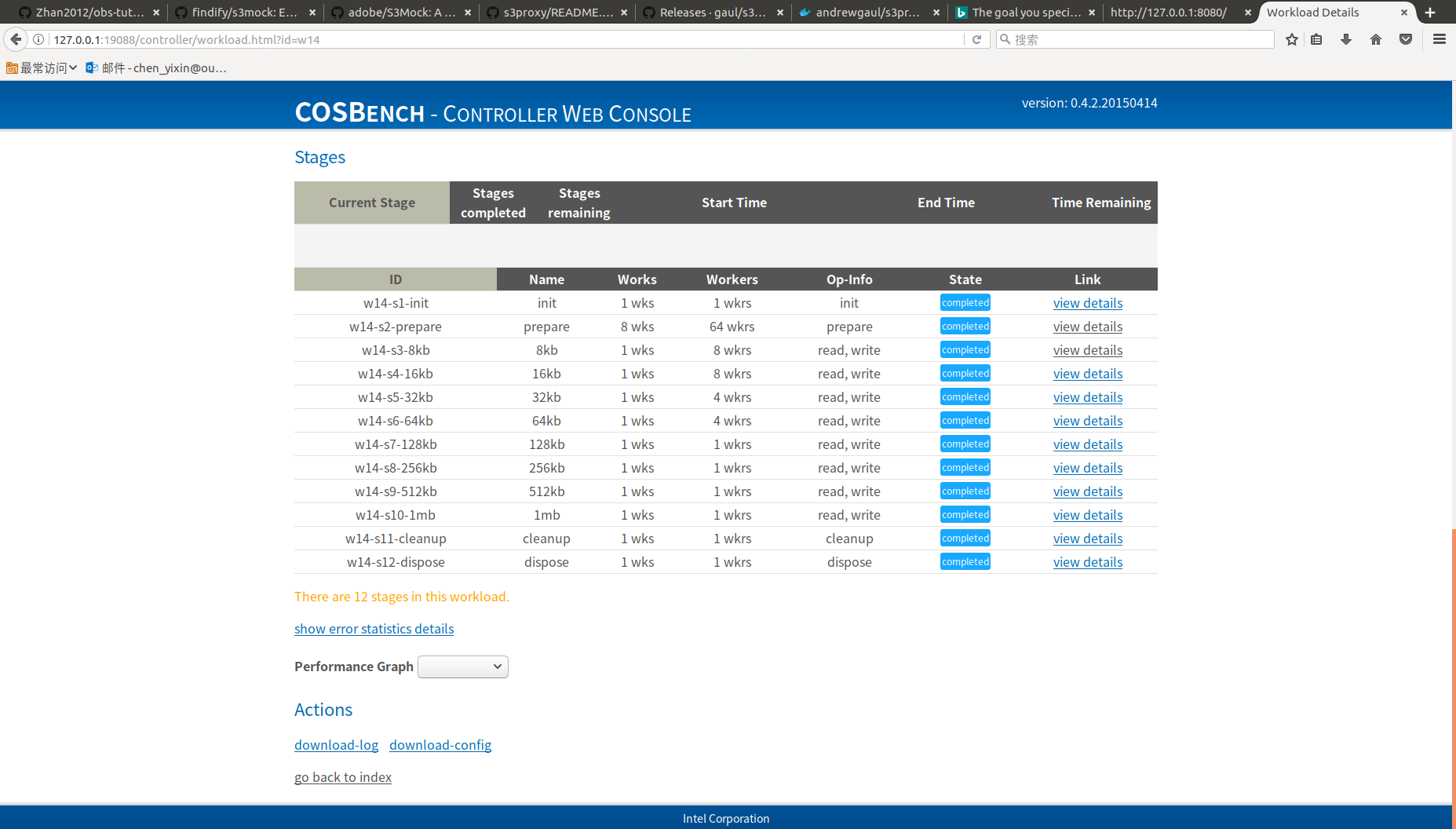


图 17 cosbench成功运行

点击Performance Graph的下拉框，可以查看统计图。

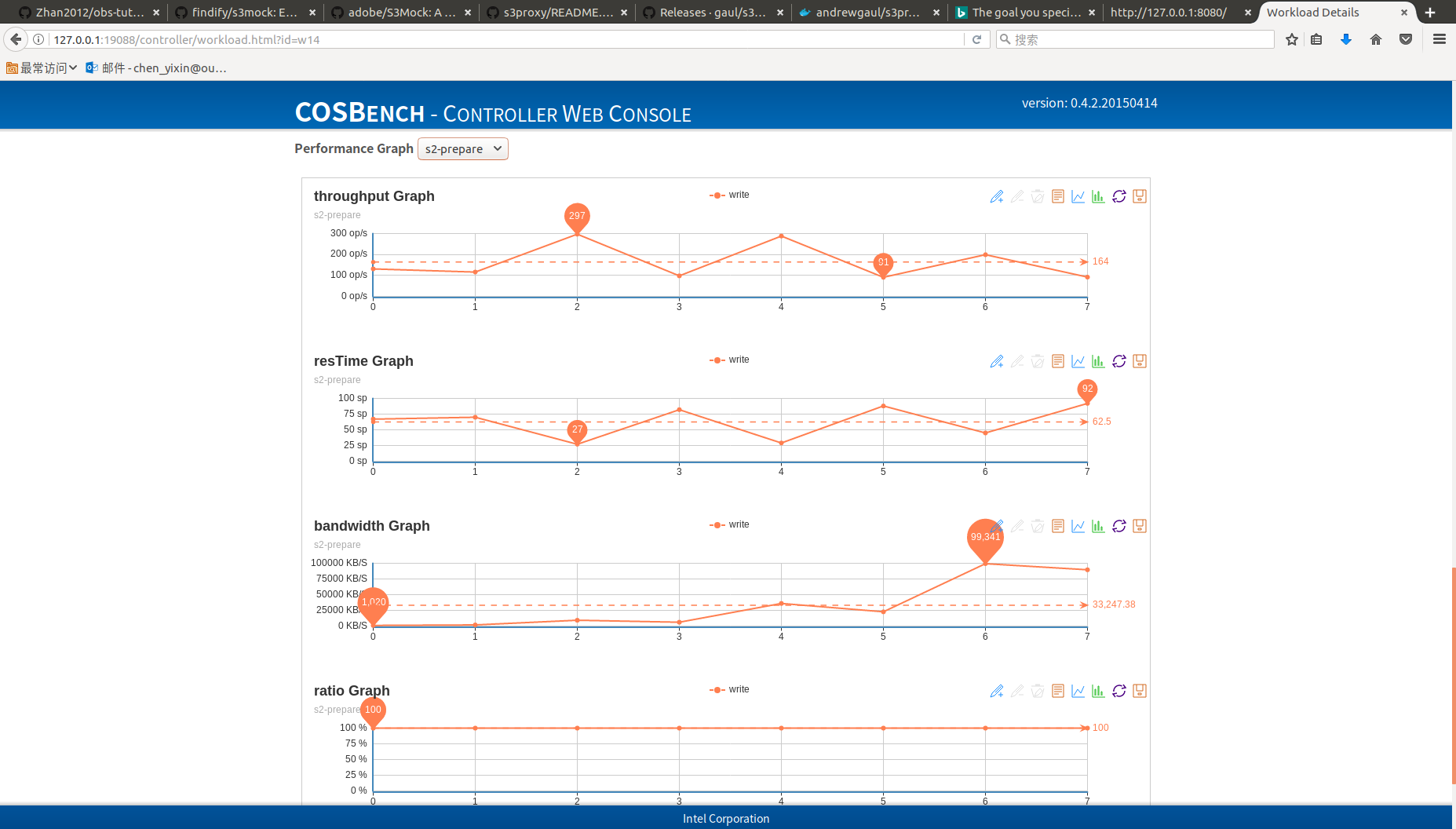


图 18 prepare阶段

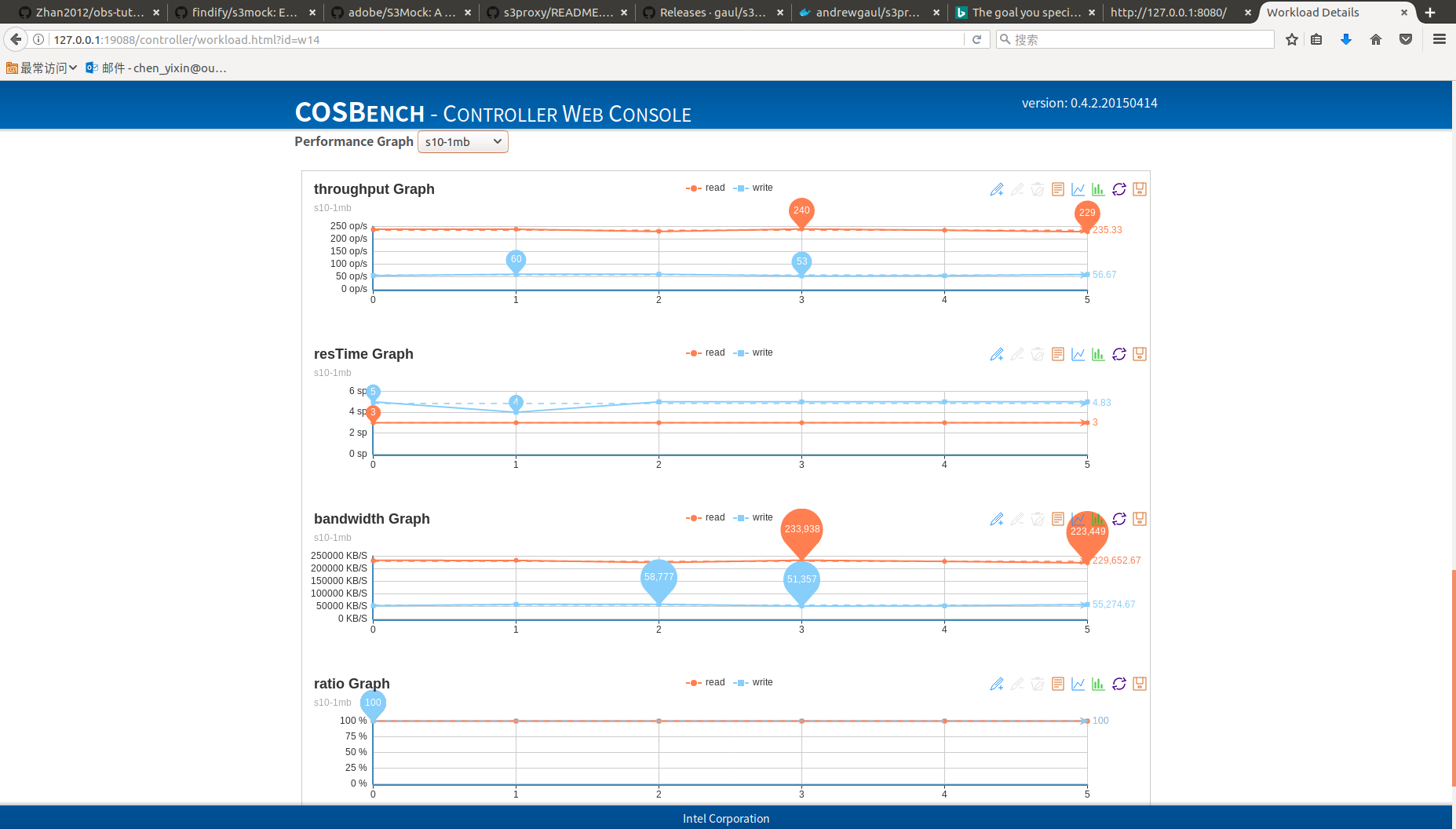


图 19 测试文件大小为1MB时

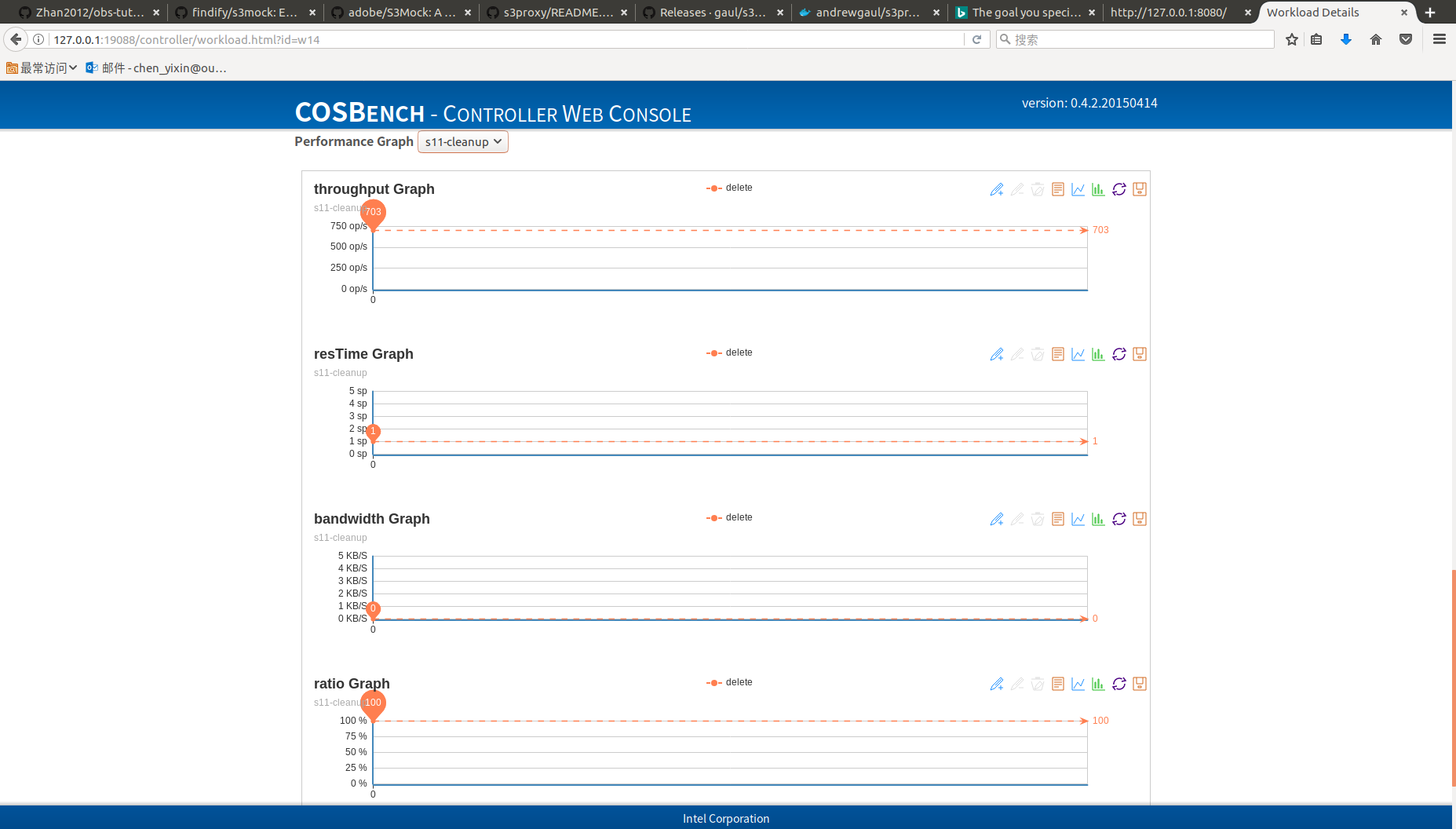


图 20 清空所写入文件时