南昌大学实验报告

姓名：**陈华豪**

学号：**6130116238**

邮箱地址：**6130116238@email.ncu.edu.cn**

专业班级：**网络工程161班**

实验日期：**2019.5.8**

课程名称：**云计算**

实验项目名称

Lab 5 Introduction to Cloud Computing-- Load Balancing

实验目的

- **Understanding the concept of load balancing**
- **Monitor the utilization status of each VM and each host**
- **Moving VMs from hot spot to cold ones**
- **Complete this experiment using at least two computers, the more the better**

实验基础

- Go to dockerhub and pull one image call mongo-express
    - https://hub.docker.com/_/mongo-express

```
docker pull mongo-express
```

```
root@vps-sfo181022:~# docker pull mongo-express
Using default tag: latest
latest: Pulling from library/mongo-express
bdf0201b3a05: Pull complete
28476b858903: Pull complete
b957d43c01b3: Pull complete
c406119d7ac9: Pull complete
cb7ab89ce172: Pull complete
61a9281e92d9: Pull complete
3b55ce185e82: Pull complete
Digest: sha256:8df289fc334f394f831700bba3db4509a60a21b60824f59206c3dd31ed718d30
Status: Downloaded newer image for mongo-express:latest
root@vps-sfo181022:~#
```

- Test this mongodb web service with simple SQL statement
    - Hints, you can generate your own DB file on random data

创建mongo-express容器需要先创建一个mongo容器
所以拉取mongo镜像

```
docker pull mongo
```

```
root@vps-sfo181022:~# docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
Digest: sha256:02c6031b363fb9a43f6633eb9db405db59c9dfdd0ce726baa4fab973939952a4
Status: Image is up to date for mongo:latest
root@vps-sfo181022:~#
```

创建一个mongo容器

```
docker run -p 27017:27017 -v /tmp/db:/data/db -d
docker exec -it hopeful_wing bash
mongo
```

```
root@vps-sfo181022:~# docker run -p 27017:27017 -v /tmp/db:/data/db -d mongo
0548f9ac7f7e9163b57f56583c5e2917fce3d1e39af8dce9dde2079720195acb
root@vps-sfo181022:~# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
     STATUS              PORTS                     NAMES
0548f9ac7f7e        mongo               "docker-entrypoint.s…"   47 seconds ago
     Up 46 seconds       0.0.0.0:27017->27017/tcp  hopeful_wing
846ef5174954        busybox             "/bin/sh -c 'i=0; wh…"   10 days ago
     Up 10 days                                    looper
root@vps-sfo181022:~# docker exec -it hopeful_wing bash
root@0548f9ac7f7e:/# mongo
MongoDB shell version v4.0.9
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("e8e81595-7974-4ec6-8cb3-0e2422a61f5b")
}
MongoDB server version: 4.0.9
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        http://docs.mongodb.org/
```

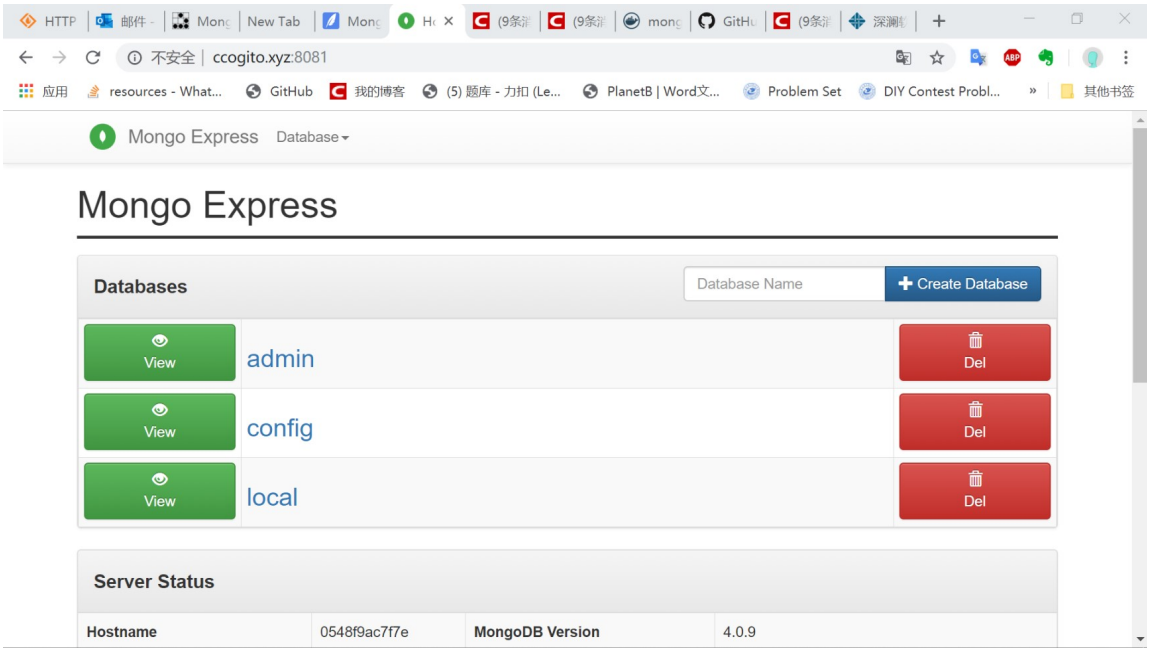
创建mongo-express容器

```
docker run -it --rm -p 8081:8081 /
--link hopeful_wing:mongo mongo-express
```

```
root@vps-sfo181022:~# docker run -it --rm -p 8081:8081 --link hopeful_wing:mong
 mongo-express
Waiting for mongo:27017...
Welcome to mongo-express
------------------------


Mongo Express server listening at http://0.0.0.0:8081
Server is open to allow connections from anyone (0.0.0.0)
basicAuth credentials are "admin:pass", it is recommended you change this in yo
r config.js!
Database connected
Admin Database connected
```

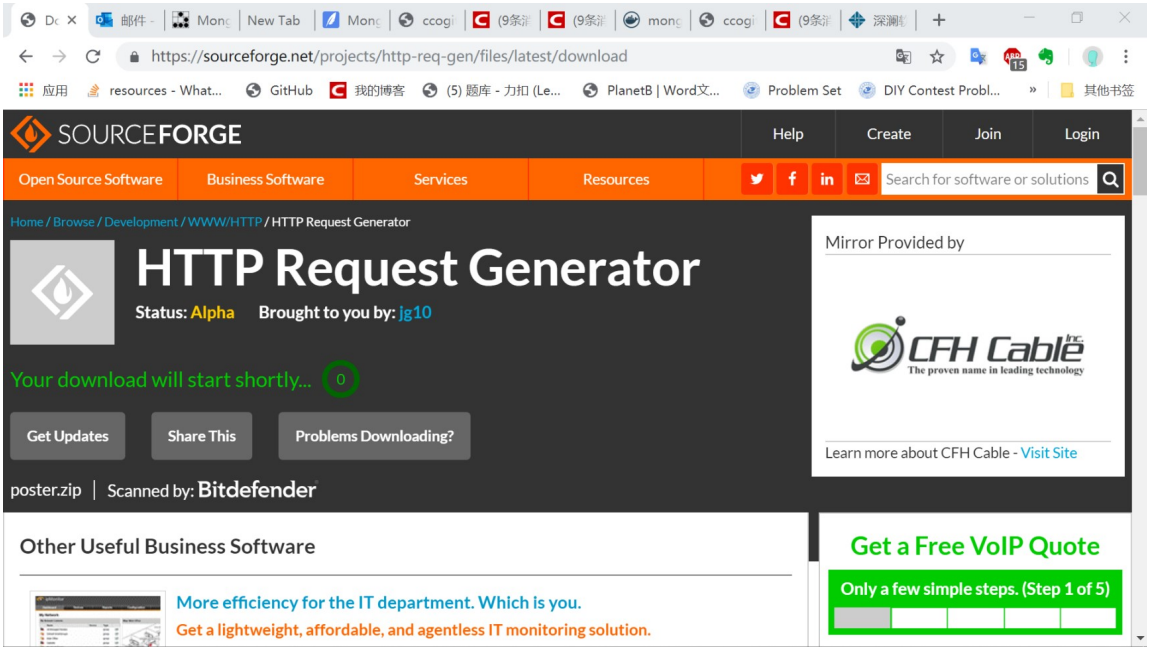
```
Mongo Express server listening at http://0.0.0.0:8081
Server is open to allow connections from anyone (0.0.0.0)
basicAuth credentials are "admin:pass", it is recommended you change this in yo
r config.js!
Database connected
Admin Database connected

GET / 200 66.437 ms - 8954
GET /public/css/bootstrap.min.css 200 5.701 ms - 121200
GET /public/css/bootstrap-theme.min.css 200 2.608 ms - 23409
GET /public/css/style.css 200 1.042 ms - 1883
GET /public/vendor-47ad38d08af7dbfa26be.min.js 200 1.476 ms - 126314
GET /public/img/mongo-express-logo.png 200 1.364 ms - 17847
GET /public/img/gears.gif 200 1.712 ms - 50281
GET /public/index-512f467a07c538127931.min.js 200 1.141 ms - 1042
GET /public/fonts/glyphicons-halflings-regular.woff2 200 1.744 ms - 18028
That database name is invalid.
POST / 302 25.079 ms - 23
```

连接至8081端口：

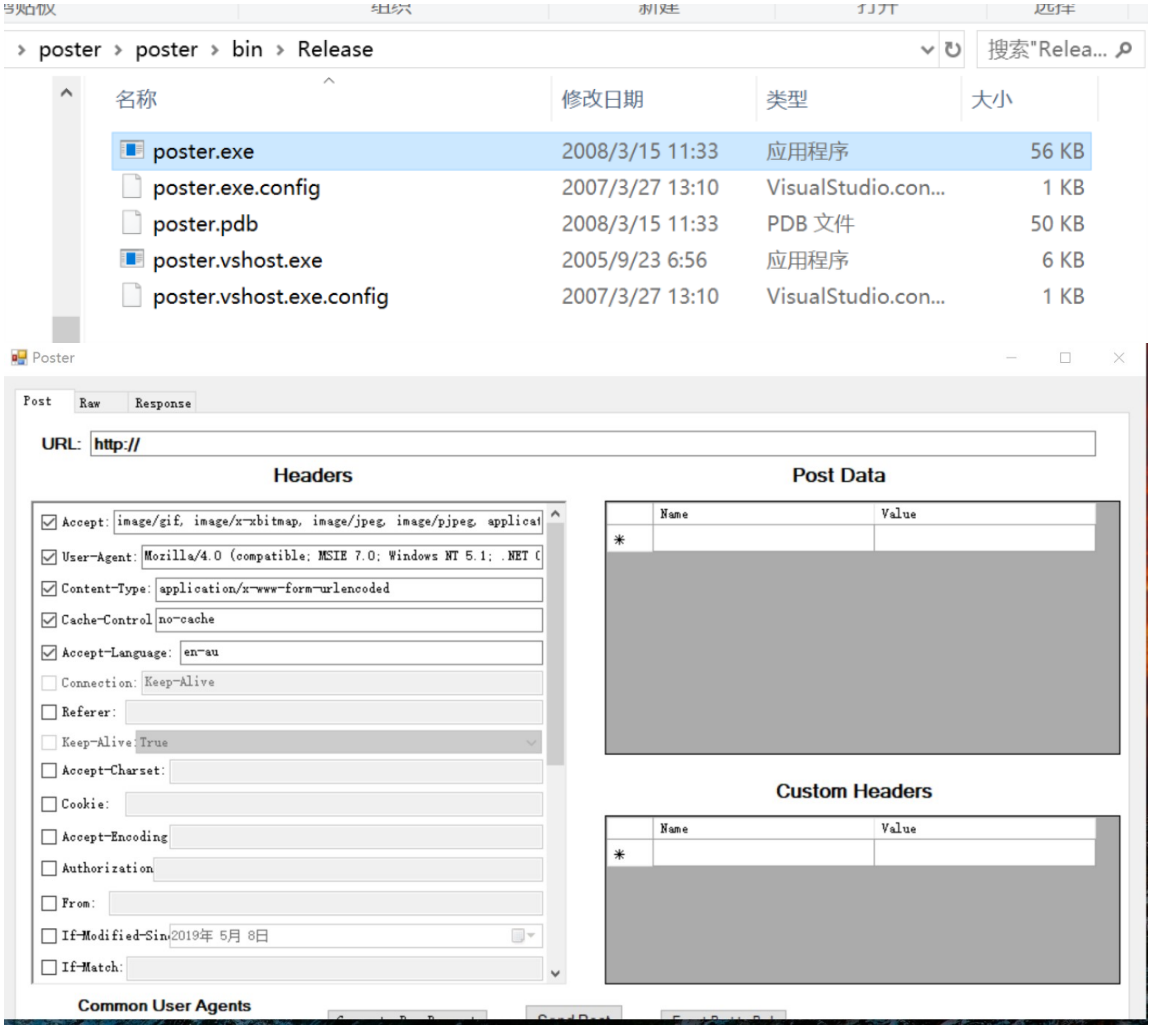


- Write a HTTP request generator that wraps SQL statement
    - https://sourceforge.net/projects/http-req-gen/
    - https://github.com/Kong/httpsnippet

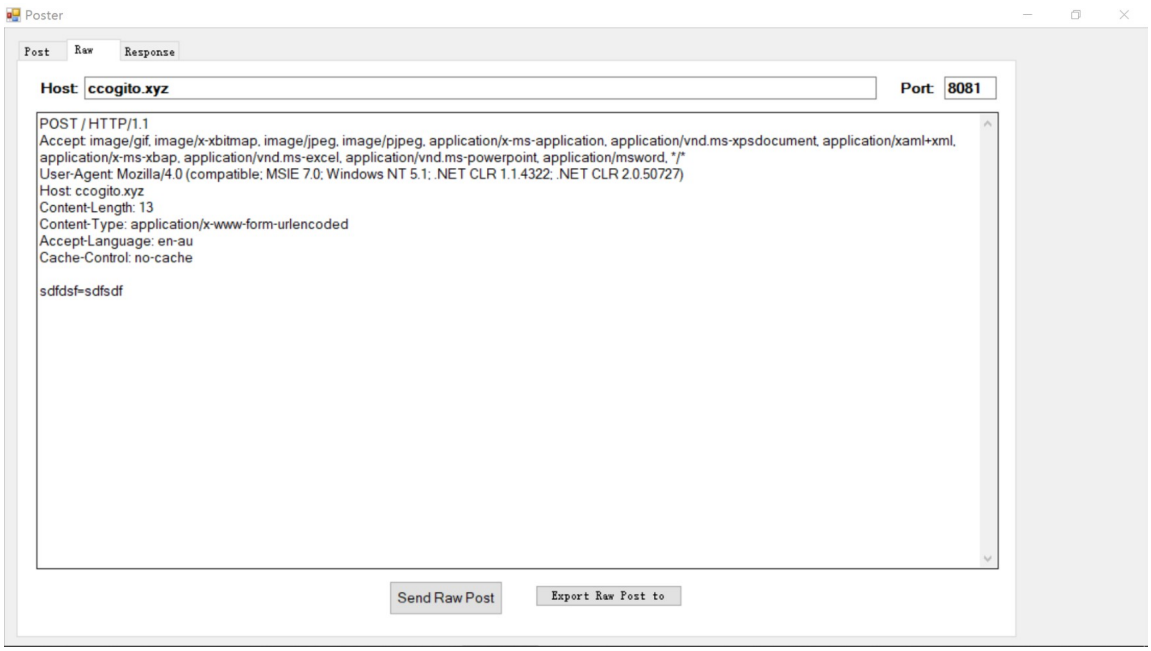
下载http-req-gen



打开poster.exe



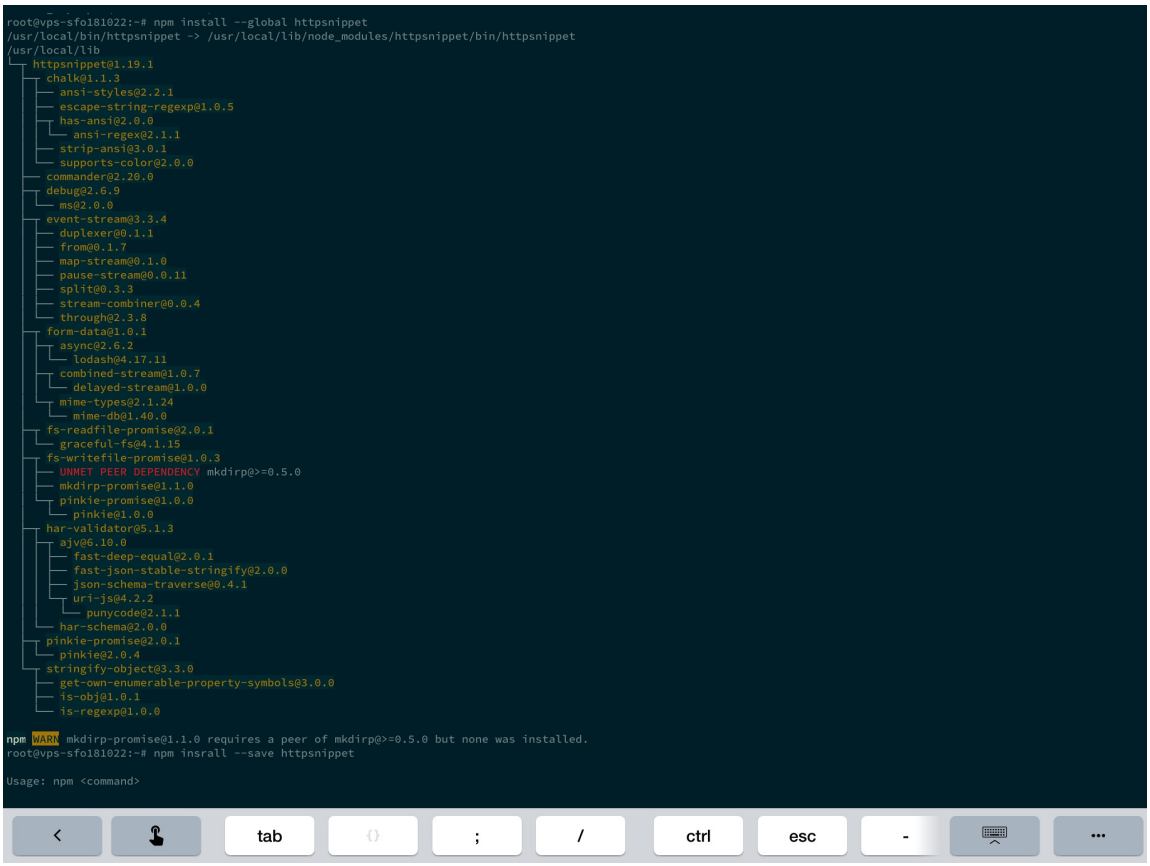
发送请求至8081端口：



得到response：

安装httpsnippet

```
# to use in cli
npm install --global httpsnippet
# to use as a module
npm install --save httpsnippet
```



实验步骤

- First, connect your request generator to the MongoDB service, basically shooting different SQL statement at a self-defined rate

- Second, make multiple duplicates of the MongoDB service, and randomly distribute the SQL statement to all service hosts
- Third, based on your homework 2-3, use your hypercall to detect the runtime system status, including CPU utilization, memory utilization, etc.
- Set a threshold for such status and name it as the hotspot when it is over
- E.g., when CPU util > 80%, this is a hot spot, you need to either move some hosts to other machines, or distribute less workloads to this spot

实验数据或结果

在上述项目中

实验思考

参考资料