

Отчёт по лабораторной работе №6
курс «Тестирование программного обеспечения Aquarius»

Выполнил студент группы ИП-211:
Вайберт Андреас Андреевич
Преподаватель курса:
Бочкарёв Борис Вячеславович

Цель работы

Научиться разрабатывать сценарии нагрузочного тестирования с использованием Locust, тестируя API OpenBMC и открытое публичное API. Оценить производительность системы, проанализировать задержки и возможные точки отказа.

Анализ результатов:

LOCUST

Hosthttps://localhost:2443

StatusRUNNING

Users110

RPS8.1

Failures0%

EDIT

STOP

RESET

STATISTICSCHARTSFAILURESEXCEPTIONSCURRENT RATIODOWNLOAD DATA! LOGSLOCUST CLOUD

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
POST	Create Session	110	0	24000	27000	28000	21969.7	2824	27896	299	0	0
GET	Get System Info	1230	0	13000	15000	27000	13541.19	9624	29133	2639	8.1	0
	Aggregated	1340	0	13000	24000	28000	14233.08	2824	29133	2446.91	8.1	0

LOCUST

Hosthttps://localhost:2443

StatusRUNNING

Users50

RPS6.8

Failures0%


EDIT

STOP

RESET

STATISTICSCHARTSFAILURESEXCEPTIONSCURRENT RATIODOWNLOAD DATA! LOGSLOCUST CLOUD

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
POST	Create Session	50	0	8100	12000	12000	8206.19	1334	12389	299	0	0
GET	Get System Info	278	0	5100	6900	13000	4983.65	621	13798	2639	6.8	0
	Aggregated	328	0	5400	11000	13000	5474.89	621	13798	2282.29	6.8	0

 **LOCUST**

Host

https://localhost:2443

Status

RUNNING

Users

25

RPS

5.9

Failures

0%

EDIT

STOP

RESET

STATISTICS

CHARTS

FAILURES

EXCEPTIONS


CURRENT RATIO

DOWNLOAD DATA

! LOGS

LOCUST CLOUD

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
POST	Create Session	25	0	1700	7400	8600	2405.49	557	8588	299	0	0
GET	Get System Info	189	0	940	3400	5600	1240.01	141	5837	2639	5.9	0
	Aggregated	214	0	980	3900	5800	1376.16	141	8588	2365.64	5.9	0

 **LOCUST**

Host

https://localhost:2443

Status

RUNNING

Users

10

RPS

3.1

Failures

0%

EDIT

STOP

RESET

STATISTICS

CHARTS

FAILURES

EXCEPTIONS


CURRENT RATIO

DOWNLOAD DATA

! LOGS

LOCUST CLOUD

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
POST	Create Session	10	0	530	670	670	533.59	462	670	299	0	0
GET	Get System Info	209	0	210	630	730	266.62	138	747	2639	3.1	0
	Aggregated	219	0	210	630	730	278.81	138	747	2532.15	3.1	0

 **LOCUST**

Host

https://localhost:2443

Status

RUNNING

Users

5

RPS

1.4

Failures

0%

EDIT

STOP

RESET

STATISTICS

CHARTS

FAILURES

EXCEPTIONS

CURRENT RATIO

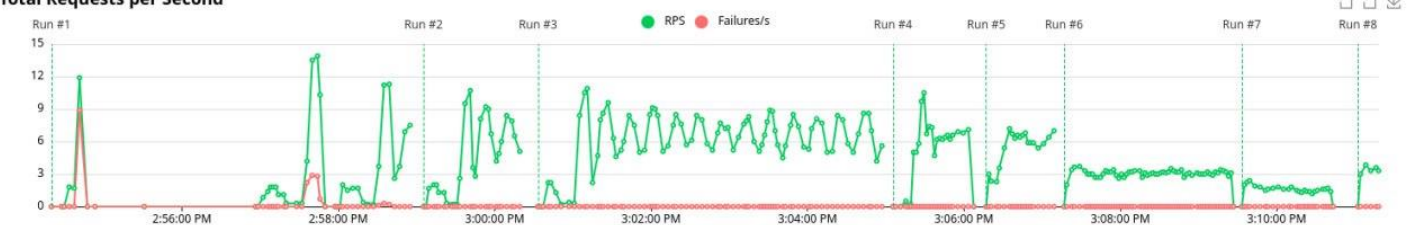
DOWNLOAD DATA

! LOGS

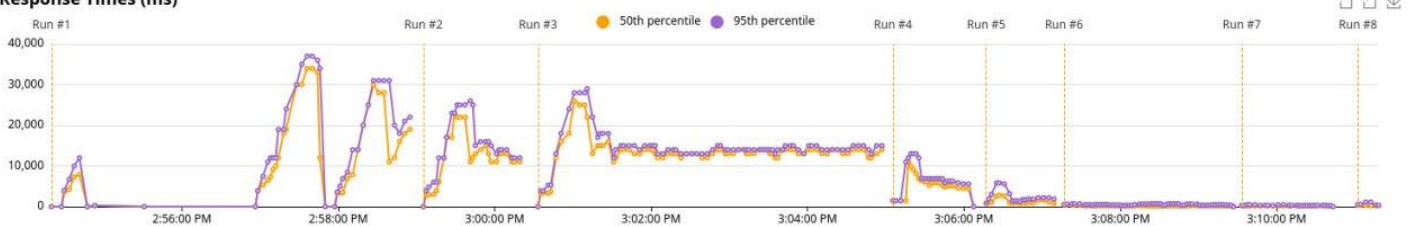
LOCUST CLOUD

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
POST	Create Session	5	0	280	400	400	300.1	271	397	299	0	0
GET	Get System Info	90	0	180	370	460	210.38	137	457	2639	1.4	0
	Aggregated	95	0	180	380	460	215.1	137	457	2515.84	1.4	0

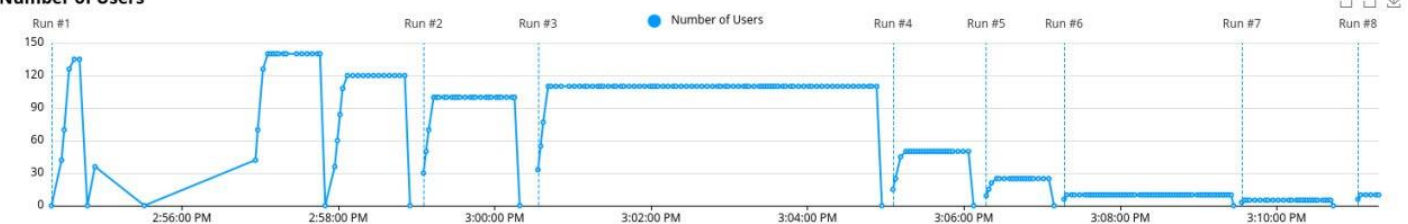
Total Requests per Second



Response Times (ms)



Number of Users



Кол-во пользователей	Среднее время отклика (мс)
5	215
10	278
25	1376
50	5474
110	14233

Для OpenVms время отклика оставалось стабильным до при 10-20 пользователей. При 25 и более пользователей задержка сильно заметна и можно уже считать, что система перегружена. При более чем 110 пользователей запросы начинали не выполняться из-за сбоев при создании сессии.

locustfile.py:

```

from locust import HttpUser, task, between, events

import logging

logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)

BMC_URL = "https://localhost:2443"

```

```

AUTH_URL = f"{BMC_URL}/redfish/v1/SessionService/Sessions"
CREDENTIALS = {"UserName": "root", "Password": "OpenBmc"}

@events.test_start.add_listener
def on_test_start(environment, **kwargs):
    logger.info("Test started")

@events.test_stop.add_listener
def on_test_stop(environment, **kwargs):
    logger.info("Test stopped")

class OpenBMCUser(HttpUser):
    host = BMC_URL
    wait_time = between(1, 5)

    def on_start(self):
        try:
            response = self.client.post(
                "/redfish/v1/SessionService/Sessions",
                json=CREDENTIALS,
                headers={"Content-Type": "application/json"},
                verify=False,
                name="Create Session"
            )
            if response.status_code not in (200, 201):
                logger.error(f"Session creation failed: {response.status_code}, {response.text}")
            return
            self.token = response.headers.get("X-Auth-Token")
            if not self.token:
                logger.error("No X-Auth-Token found in response headers")
            else:
                logger.info("Session token created successfully")
        except Exception as e:
            logger.error(f"Error creating session: {e}")

    @task
    def get_system_info(self):
        if not hasattr(self, "token") or not self.token:
            logger.warning("Skipping task: No valid session token")
            return
        self.client.get(
            "/redfish/v1/Systems/system",
            headers={"X-Auth-Token": self.token},
            verify=False,
            name="Get System Info"
        )

class PublicAPIUser(HttpUser):
    host = "Public APIs"

```

```
wait_time = between(1, 5)
```

```
@task
```

```
def get_posts(self):
```

```
    self.client.get(
```

```
        "https://jsonplaceholder.typicode.com/posts",
```

```
        name="Get Posts"
```

```
    )
```

```
@task
```

```
def get_weather(self):
```

```
    self.client.get(
```

```
        "https://wttr.in/Novosibirsk?format=j1",
```

```
        name="Get Weather"
```

```
    )
```