

С помощью системы нагрузочного тестирования провести имитацию “хабразффекта” на проект. Сделать выводы, насколько хорошо система справляется с наплывом пользователей и ботов.

- 1) Была установлена система нагрузочного тестирования Яндекс-Танк с помощью установщика библиотек python pip. Генератор нагрузки phantom установлен из пакетного менеджера (apt-get).

PIP-based installation

Other option is installing yandex-tank on your machine. We will describe the installation process for debian-based systems, but we think you can figure it out how to do it on your system of choice (some people run Yandex.Tank on their Macs for example). This installation process is slightly different from the one described in official docs because we need the latest version from github master branch for Overload.

These are the packages that are required to build different python libraries. Install them with `apt` :

```
sudo apt-get install python-pip build-essential python-dev libffi-dev gfortran libssl-dev
```

Update your pip:

```
sudo -H pip install --upgrade pip
```

Update/install your setuptools:

```
sudo -H pip install --upgrade setuptools
```

Install latest Yandex.Tank from master branch:

```
sudo -H pip install https://api.github.com/repos/yandex/yandex-tank/tarball/master
```

You'll probably need Phantom load generator, so install it from our ppa:

```
sudo add-apt-repository ppa:yandex-load/main && sudo apt-get update  
sudo apt-get install phantom phantom-ssl
```

По рекомендации описания яндекс-танк изменены настройки сети слейва для увеличения производительности системы нагрузочного тестирования.

```
ulimit -n 30000  
  
net.ipv4.tcp_max_tw_buckets = 65536  
net.ipv4.tcp_tw_recycle = 1  
net.ipv4.tcp_tw_reuse = 0  
net.ipv4.tcp_max_syn_backlog = 131072  
net.ipv4.tcp_syn_retries = 3  
net.ipv4.tcp_synack_retries = 3  
net.ipv4.tcp_retries1 = 3  
net.ipv4.tcp_retries2 = 8  
net.ipv4.tcp_rmem = 16384 174760 349520  
net.ipv4.tcp_wmem = 16384 131072 262144  
net.ipv4.tcp_mem = 262144 524288 1048576  
net.ipv4.tcp_max_orphans = 65536  
net.ipv4.tcp_fin_timeout = 10  
net.ipv4.tcp_low_latency = 1  
net.ipv4.tcp_syncookies = 0  
net.netfilter.nf_conntrack_max = 1048576
```

Впоследствии обнаружилась несовместимость версий пакета netort , пришлось дополнительно устанавливать дополнительную версию этого пакета.

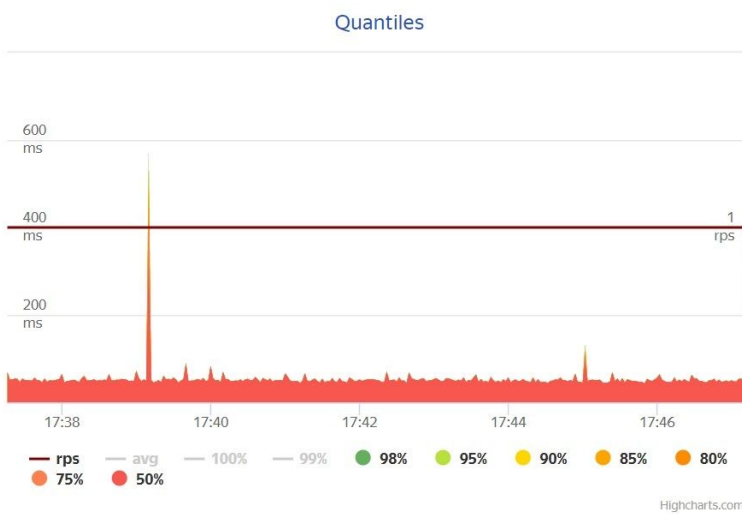
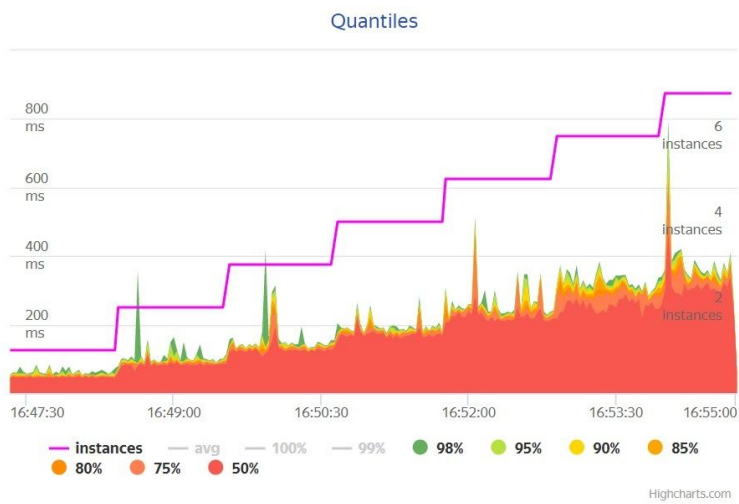
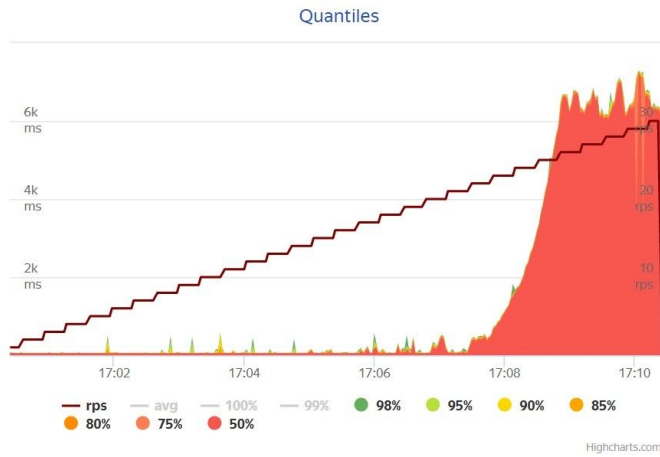
- 2) Произведено тестирование основного вебсайта с разными видами нагрузки - переменное количество запросов в секунду, переменное количество клиентов, малая постоянная нагрузка от одного клиента.

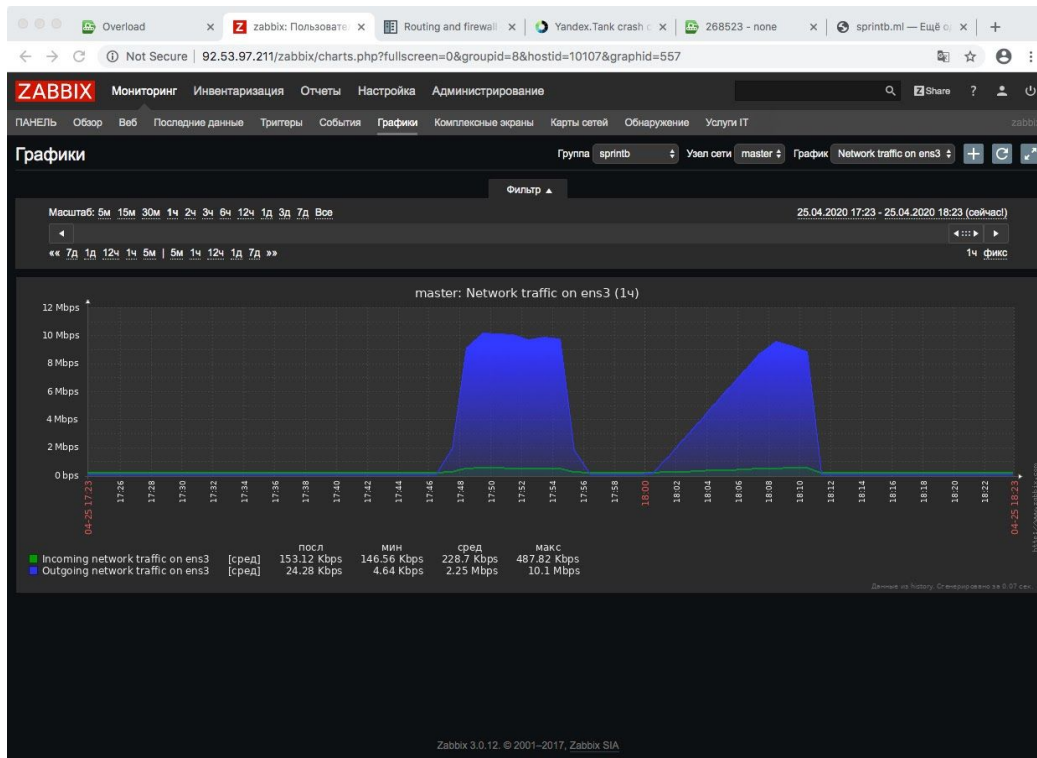
```
aggy@299652-mike199567:~$ cat loadmon.yaml
# simple test configuration with monitoring
overload:
  enabled: true
  package: yandex-tank.plugins.DataUploader
  token_file: "token.txt"
phantom:
  address: sprintb.ml:443 # [Target's address]:[target's port]
  ssl: true
  uris:
    - /
  load_profile:
    load_type: rps # schedule load by defining requests per second
    schedule: line(1, 30, 10m) # starting from 1rps growing linearly to 30rps during 10 minutes
console:
  enabled: true # enable console output
telegraf:
  enabled: false # let's disable telegraf monitoring for the first time
```

```
aggy@299652-mike199567:~$ cat loadinst.yaml
# simple test configuration with monitoring
overload:
  enabled: true
  package: yandex-tank.plugins.DataUploader
  token_file: "token.txt"
phantom:
  address: sprintb.ml:443 # [Target's address]:[target's port]
  ssl: true
  uris:
    - /
  load_profile:
    load_type: instances # schedule load by defining number of instances
    schedule: line(1, 10, 10m) # starting from 1 growing linearly to 10 during 10 minutes
    instances: 10
    loop: 10000
console:
  enabled: true # enable console output
telegraf:
  enabled: false # let's disable telegraf monitoring for the first time
```

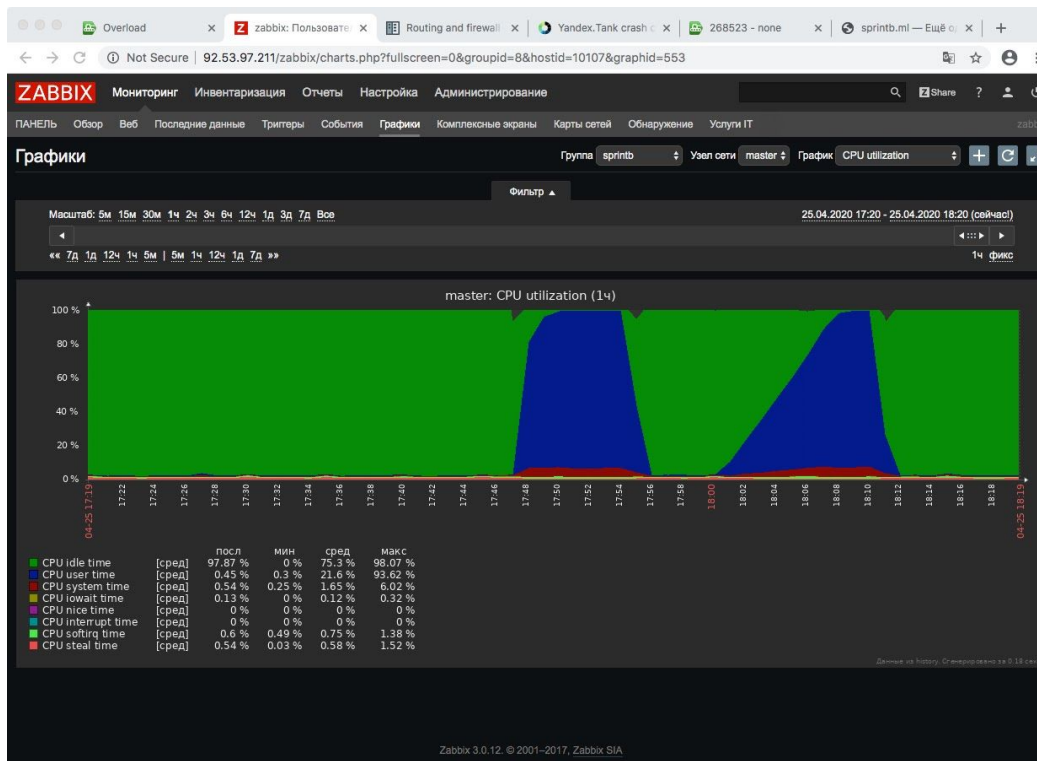
```
aggy@299652-mike199567:~$ cat loadconst.yaml
# simple test configuration
overload:
  enabled: true
  package: yandex-tank.plugins.DataUploader
  token_file: "token.txt"
phantom:
  address: sprintb.ml:443 # [Target's address]:[target's port]
  ssl: true
  uris:
    - /
  load_profile:
    load_type: rps # schedule load by defining requests per second
    schedule: const(1, 10m) # constant load 1 rps during 10 minutes
console:
  enabled: true # enable console output
telegraf:
  enabled: false # let's disable telegraf monitoring for the first time
```

3) Анализ нагрузки производился с помощью системы Яндекс Оверлоад и системы мониторинга zabbix

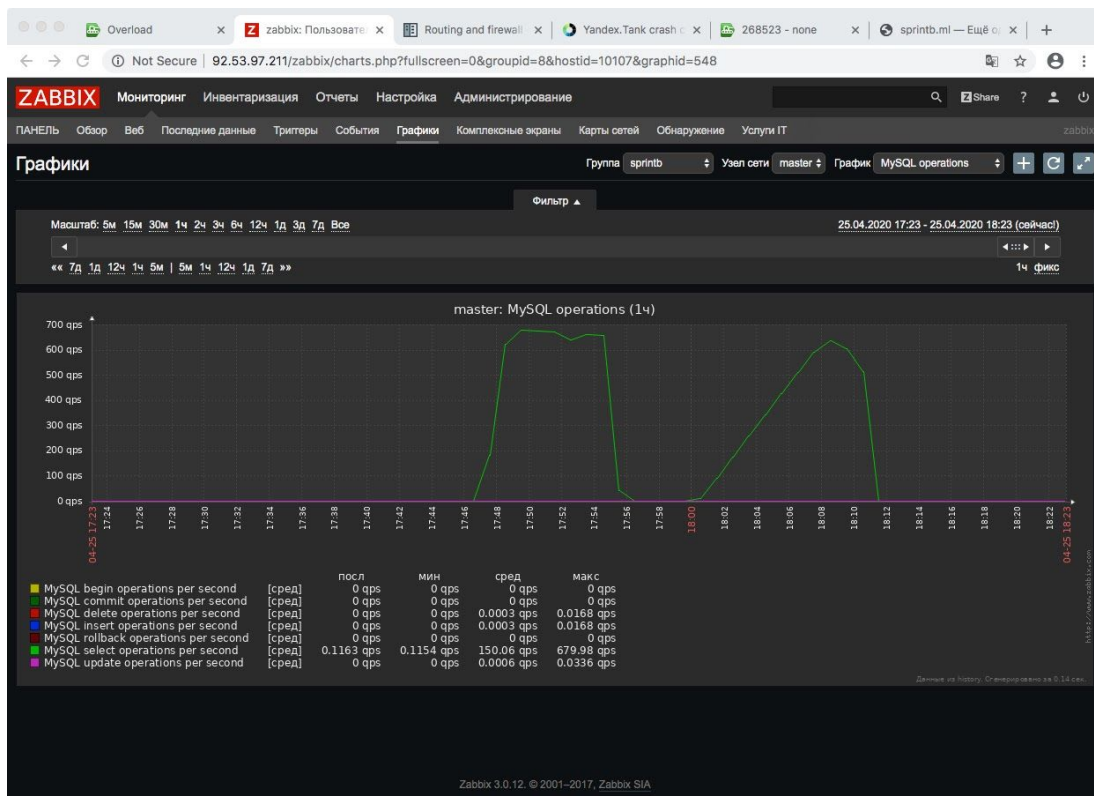
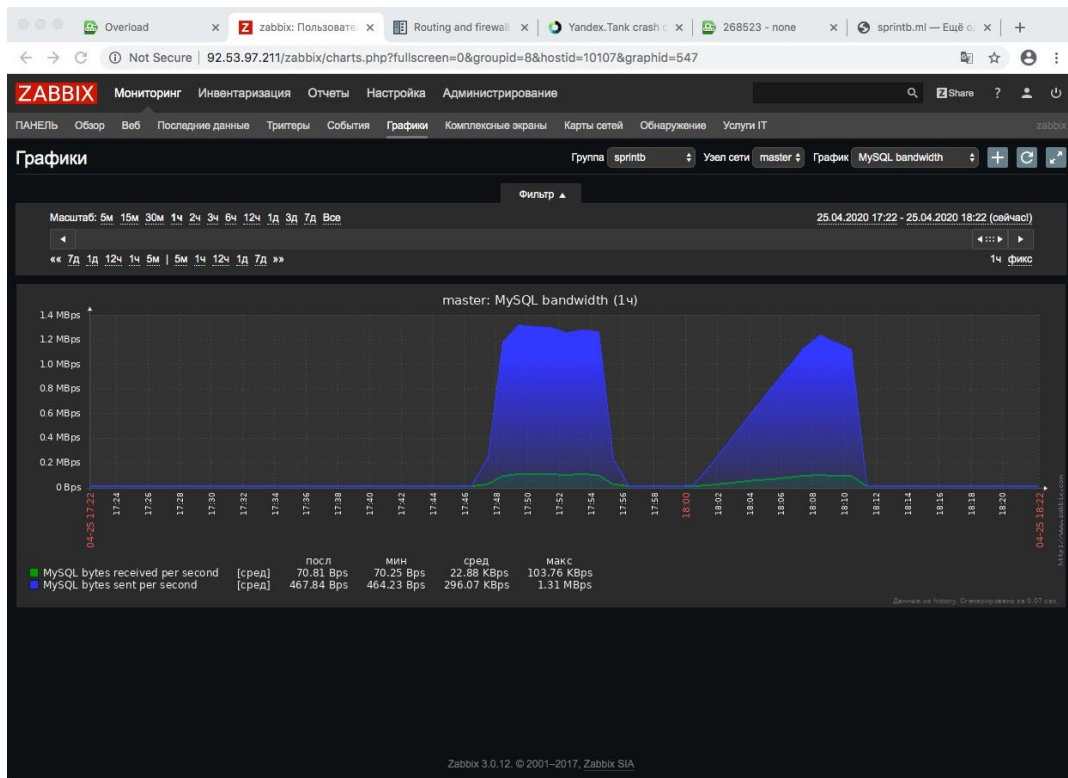




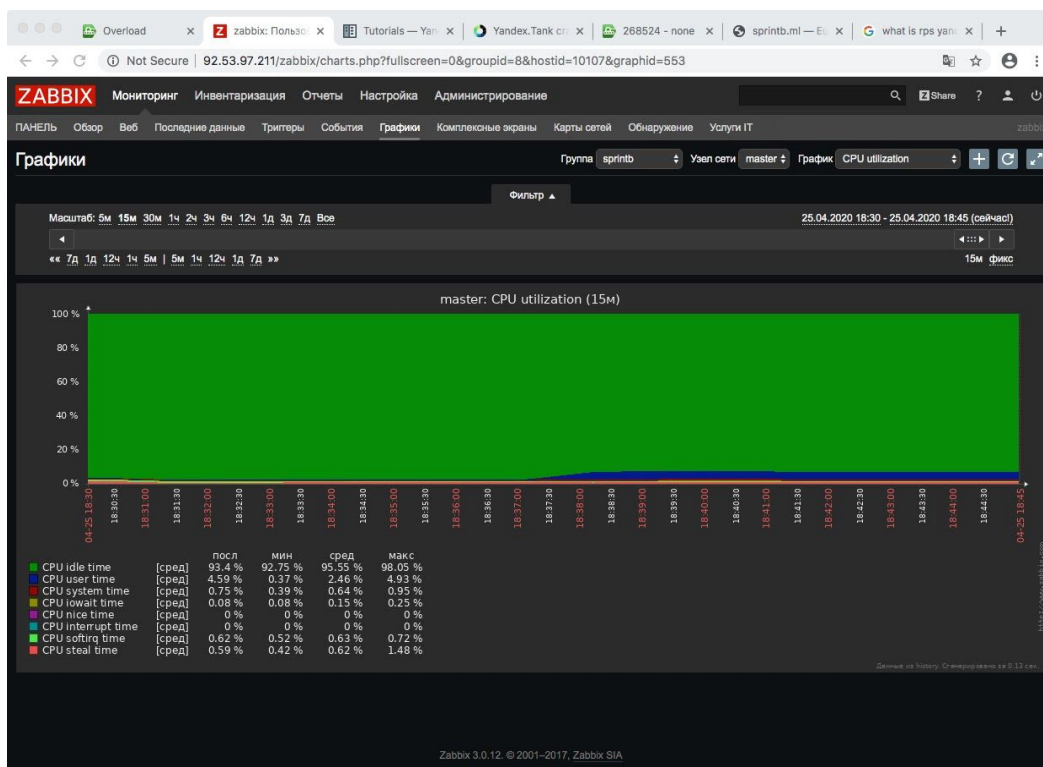
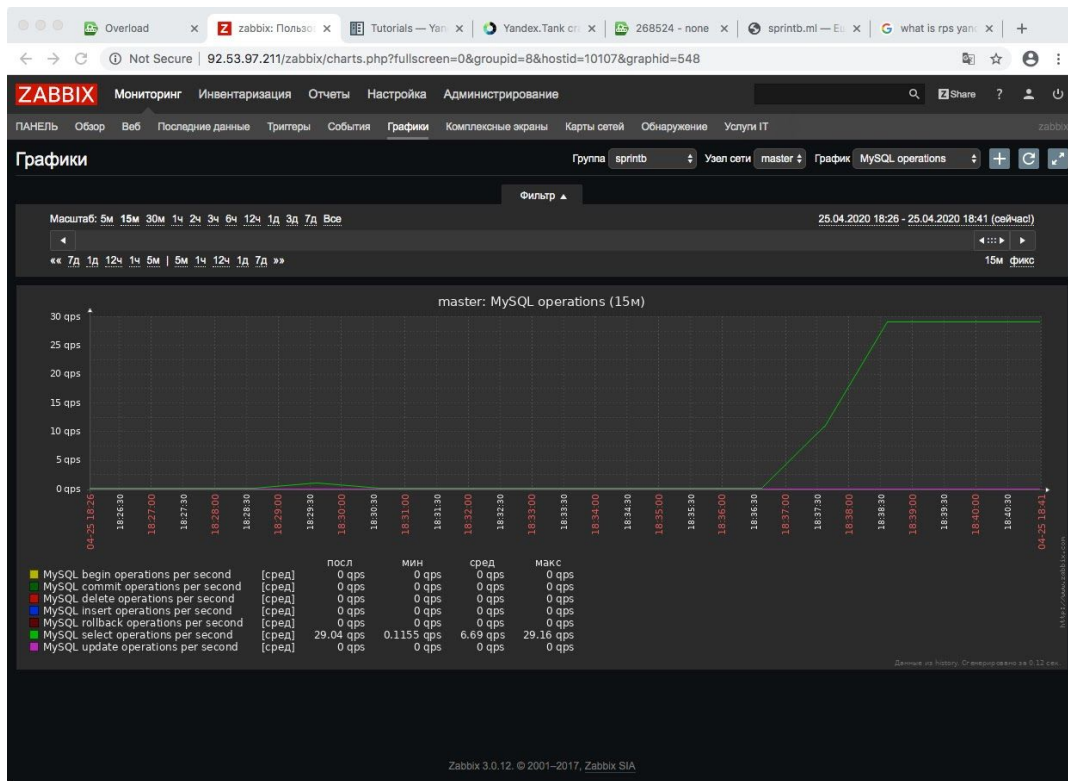
4) Анализ данных мониторинга заббивка выявил нехватку ресурсов CPU при относительно небольших нагрузках (23 rps)



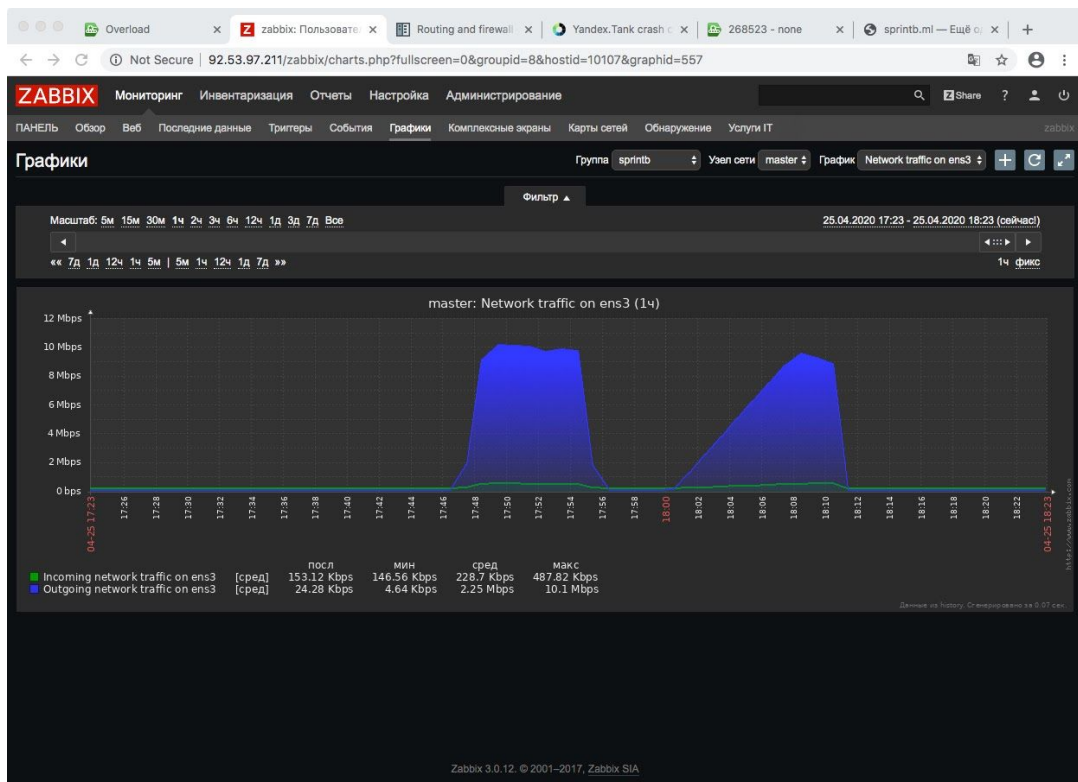
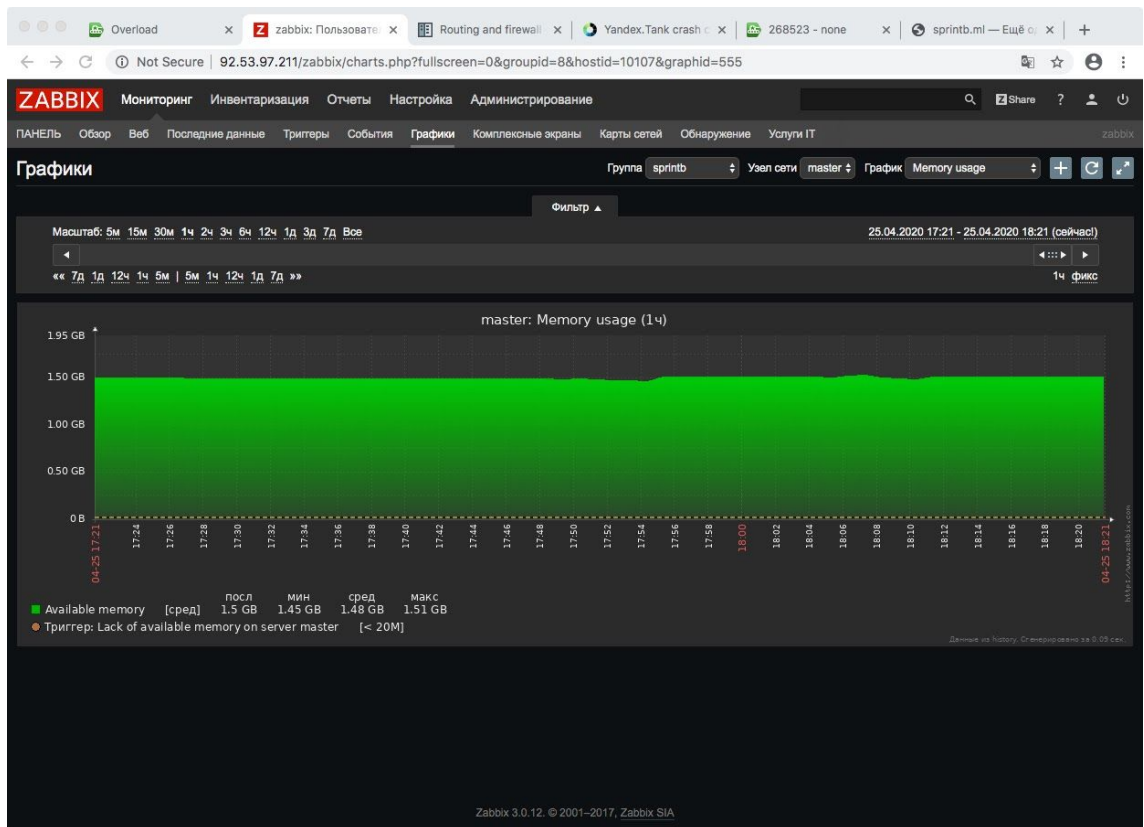
- 5) Дальнейшее исследование показало, что CPU грузится из-за большого количества запросов в базу данных при каждом запросе основной страницы сайта.



- 6) При этом загрузка на одного пользователя при одном qps (при чтении одной страницы в секунду) достаточно мала, но уже при десятках одновременных десятках запросов сервер не выдерживает нагрузки.



7) Существенной загрузки памяти и сетевых ресурсов не выявлено.

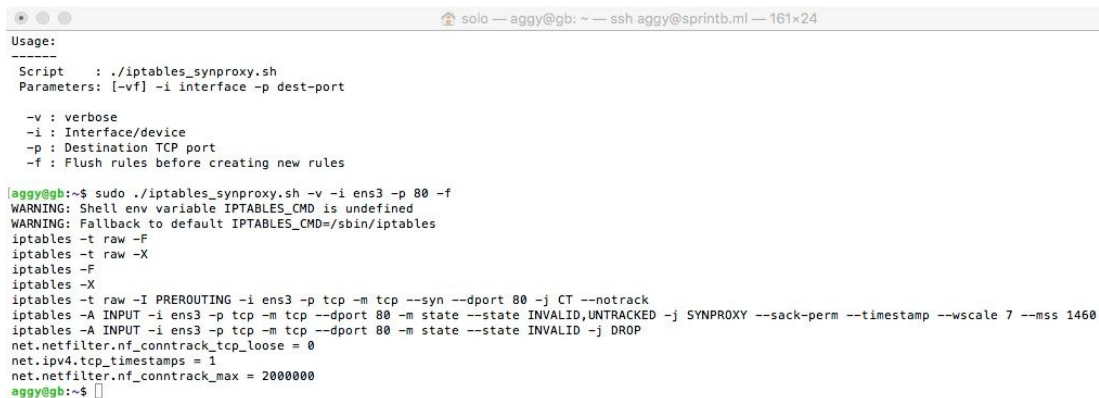


- 8) Таким образом, в качестве вывода можно сказать, что сайт на движке wordpress требует оптимизации обращений к базам данных или радикального увеличения ресурсов CPU (распараллеливания процессорных мощностей и/или серверов баз данных)

* Установка и настройка защиты от TCP SYN flood атаки.

Одним из распространенных вариантов ddos атаки является SYN flood атака, при этом одним из вариантов защиты от подобной атаки является настройка специальных правил сервиса iptables , установка модуля synproxy для iptables и настройка сетевых параметров (netfilter).

1. В ходе работы для начала текущие правила iptables были сохранены утилитой iptables-save
2. Для настройки новых правил использовался скрипт https://github.com/netoptimizer/network-testing/blob/master/iptables/iptables_synproxy.sh

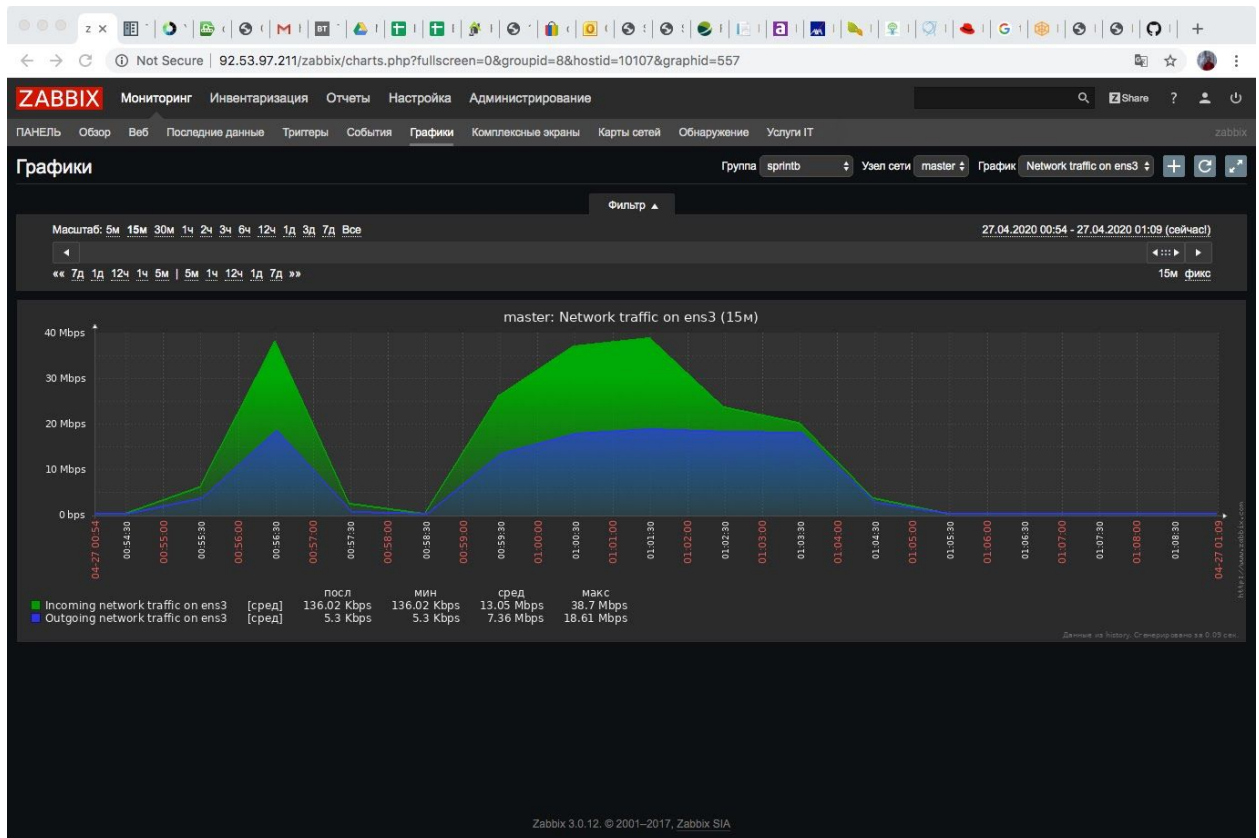


```
solo — aggy@gb: ~ — ssh aggy@sprintb.ml — 161x24
Usage:
Script      : ./iptables_synproxy.sh
Parameters: [-vf] -i interface -p dest-port

-v : verbose
-i : Interface/device
-p : Destination TCP port
-f : Flush rules before creating new rules

aggy@gb:~$ sudo ./iptables_synproxy.sh -v -i ens3 -p 80 -f
WARNING: Shell env variable IPTABLES_CMD is undefined
WARNING: Fallback to default IPTABLES_CMD=/sbin/iptables
iptables -t raw -F
iptables -t raw -X
iptables -F
iptables -X
iptables -t raw -I PREROUTING -i ens3 -p tcp -m tcp --syn --dport 80 -j CT --notrack
iptables -A INPUT -i ens3 -p tcp -m tcp --dport 80 -m state --state INVALID,UNTRACKED -j SYNPROXY --sack-perm --timestamp --wscale 7 --mss 1460
iptables -A INPUT -i ens3 -p tcp -m tcp --dport 80 -m state --state INVALID -j DROP
net.netfilter.nf_conntrack_tcp_loose = 0
net.ipv4.tcp_timestamps = 1
net.netfilter.nf_conntrack_max = 2000000
aggy@gb:~$
```

3. Для проверки устойчивости использовался пакет Hping3 , который был установлен на slave сервере.
4. Каких-либо проблем/замедлений в работе основного сайта обнаружено не было.
5. Zabbix показал существенное увеличение входного трафика , загрузка CPU не увеличивалась.



6. При повторении теста с исходными правилами iptables не удалось заблокировать работу сайта. Видимо какой-то из шлюзов уже ограничивает трафик (от одного до другого сервера 5 прыжков). Возможно имеет смысл повторить тест с возвратом исходных настроек netfilter.

В процессе сделали бэкап iptables

```
iptables-save > /home/alexandr/iptables.rules
```

Для восстановления

```
iptables-restore < /home/alexandr/iptables.rules
```

Настройка sysctl.conf

```
net.ipv4.tcp_syncookies=1
```

```
net.ipv4.tcp_timestamps = 1
```

```
net.netfilter.nf_conntrack_tcp_loose=0
```

```
net.netfilter.nf_conntrack_max=2000000
```

```
sysctl -a
```

```
iptables -t raw -I PREROUTING -i 80 -p tcp -m tcp --syn --dport 80 -j CT --notrack
```

```
iptables -A INPUT -i 80 -p tcp -m tcp --dport 80 -m state --state INVALID,UNTRACKED -j SYNPROXY  
--sack-perm --timestamp --wscale 7 --mss 1460
```

```
iptables -A INPUT -m state --state INVALID -j DROP
```

Для просмотра статистики:

```
iptables -t raw -vnL
```

```
cat /proc/net/stat/synproxy
```