Лабораторная работа 6

Цель лабораторной работы: создать высоконагруженный счётчик посещения ссылок.

**Содержание**

* 1 Установка
  + 1.1 окружение для сборки
  + 1.2 Redis
  + 1.3 NginX + redis2\_module
* 2 Настройка
* 3 Управление NginX'ом
* 4 Тестирование
* 5 Задание
* 6 Расширенное задание

Установка

Окружение для сборки

sudo apt-get install --yes libreadline-dev libncurses5-dev libpcre3-dev libssl-dev perl make build-essential

Redis

sudo apt-get install --yes redis-server redis-tools

echo 'ULIMIT=65536' >> /etc/default/redis-server

service redis-server restart

[Ссылка](http://redis.io/topics/quickstart) на quick start

NginX + redis2\_module

sudo -i

mkdir -p /opt/{nginx-build,nginx,redis2}/

cd /opt/redis2

wget <https://github.com/openresty/redis2-nginx-module/archive/refs/tags/v0.15.tar.gz>

tar xf v0.15.tar.gz

cd -

cd /opt/nginx-build/

wget ' [https://nginx.org/download/nginx-1.18.0.tar.gz'](http://nginx.org/download/nginx-1.9.15.tar.gz')

tar -xzvf nginx-1.18.0.tar.gz

cd nginx-1.18.0/

apt-get install libpcre3 libpcre3-dev zlib1g zlib1g-dev libssl-

./configure --prefix=/opt/nginx --add-module=/opt/redis2/redis2-nginx-module-0.15/

make -j2

make install

Настройка

sudo -i

cd /opt/nginx/

touch nginx.conf

nano nginx.conf

**Вставляете текст:**

events {

worker\_connections 4096;

}

http {

server {

listen 8000;

location / {

# $uri variable holds part of url after "address:port"

# Redis Server is available via 6379 port on localhost

## DO YOUR WORK HERE ##

# docs: <https://github.com/openresty/redis2-nginx-module>

}

}

}

Управление NginX'ом

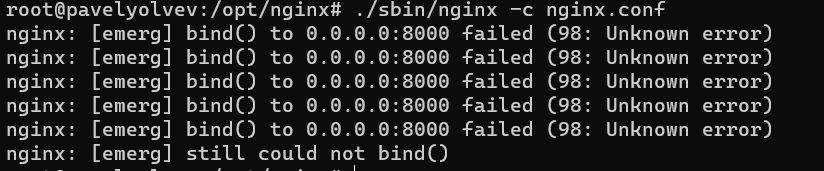
Запуск:

cd /opt/nginx/

ulimit -n 65536

./sbin/nginx -c nginx.conf

Если при запуске появляется



Значит сервер уже запущен.

Для перезапуска выполните: pkill nginx # для остановки и

./sbin/nginx -c nginx.conf # для запуска сервера

**Важно: Перезапуск нужен для применения изменений файла nginx.conf**

Остановка:

pkill nginx # останавливает все nginx-процессы в системе

Тестирование

Apache Benchmark

Установка:

sudo apt-get install -y apache2-utils

Запуск:

ulimit -n 65536

ab -n 10000 -kc 500 <http://server:8000/url1>

Задание

1. Воспроизвести аналогичную систему.
2. Настроить её так, чтобы в Redis записывалось каждое посещение страницы (ключ - url страницы, например "/my/awesome/page.html")

Пример nginx.conf:

events {

worker\_connections 4096;

}

http {

server {

listen 8000;

location / {

# $uri variable holds part of url after "address:port"

# Redis Server is available via 6379 port on localhost

## DO YOUR WORK HERE ##

# docs: https://github.com/openresty/redis2-nginx-module

}

location /MyAwesomePage {

redis2\_pass 127.0.0.1:6379;

redis2\_query incr mycounter; #mycounter – имя ключа redis для записи посещений (может быть любым, главное чтобы совпадало с настройкой ключа в redis)

}

}

}

На страницу /MyAwesomePage будет выведен результат запроса redis2\_query incr mycounter;

**Важно: для работы счетчика необходимо создать базу на redis куда будут записываться посещения:**

(в терминале):

Redis-cli

Select 0

Set mycounter 0

1. Разобраться и понимать в архитектуру и работе стенда.
2. Провести тестирование с помощью Apache Benchmark ("ab")
3. Убедиться, что все показы записались в базу (например с помощью "redis-cli").

Redis-cli

get <имя ключа>

Расширенное задание

**Cамостоятельно** доработать данный стенд:

* используя Python/Flask
* подключить код к NginX через "uwsgi"
* который будет (речь идёт про код)
  + выводить список всех ключей Redis (отвечать по адресу: "/admin/all\_keys")
  + выводить значение конкретного счётчика (пример: "/admin/key2value/some/url"; в данному случае должно быть отображено кол-во посещений адреса "/some/url")
  + удалять конкретный счётчик из базы (пример: "/admin/key2flush/some/url2/")