

OSM Tile Server XPE NAS DS 2422+ SSD VMM

Введение

Оборудование

- **Материнская плата Supermicro MBD-H11SSL-I**
- **Процессор AMD EPYC™ 7551P OEM**
- **Raid5 массив 1tb SSD M.2 Samsung 970 EVO Plus**
- **256GB DDR4 3200MHz DIMM ECC Reg Micron**
- **OC Ubuntu Server 20.04 LTS**
- Основное ПО сервера **XPEology NAS DS2422+** с установленным 32 ядерным серверным процессором и 4 модулями оперативной памяти **DIMM ECC** по 64GB с настроенной Synology VMM и установленной на виртуальной машине OC Ubuntu Server 20.04 LTS. Так же установлено 3 диска SSD M.2 в Raid5 массив.
- -Установка и настройка Synology Virtual Machine Manager (VMM) Synology DS420+.
- -Установка Ubuntu 20.04 LTS Server на виртуальную машину (VMM) Synology DS420+.
- В целом, для среднего города понадобится 10-20gb памяти на жестком диске, 4gb ram и процессор не менее двух ядер. Если же нужна вся планета, необходимо предоставить более 1tb на ssd диске, 36gb ram и 8core процессор.
- Предполагается, что при выделенных под VMM ресурсах (AMD EPYC™ 7551P 16*2,00 GHz / 2*64gb ddr4 3200MHz DIMM ECC Reg / 3*1tb hdd raid5 1том) процессор будет загружаться при рендере на 40-50%, при учете того, что будет выделено 128gb виртуальной памяти и теоретическом задействовании всего 10% от этого объема.
- Все ниже написанное относиться к Ubuntu Linux 20.04 LTS (Focal Fossa) и запущенно в работу в августе 2024 года.

Подготовка оборудования

После физического монтажа всех комплектующих переходим к настройке

nvme ssd в качестве тома хранилища

[3*1000 ГБ SSD M.2 накопитель Samsung 970 EVO Plus \[MZ-V7S1T0BW\]](#) установлены в

Общая теория

- Tile Server OSM представляет собой набор программ и библиотек, которые работают вместе для создания тайлового (плиточного) сервера. Он состоит из 5 основных компонентов: mod_tile, renderd, mapnik, osm2pgsql и базы данных postgresql/postgis. Mod_tile — это модуль apache, который обслуживает кэшированные тайлы и решает,

какие тайлы нуждаются в повторном рендеринге — либо потому, что они еще не кэшированы, либо потому, что они устарели. Mapnik — это программная библиотека, которая осуществляет реальный рендеринг с использованием открытого рендеринга.

- Предполагается, что вы запускаете все от пользователя без регистрации root через «sudo». Имя пользователя без root, повторяемое по умолчанию, — «alisa». Если вы указываете иного пользователя, то вам необходимо будет заменить во всех командах на свое имя и добавить его в группу пользователей, которые могут использовать sudo для получения root-прав. Из вашей обычной учетной записи пользователя без регистрации root.

Установка необходимых библиотек

Итак, приступим:

```
sudo apt install libboost-all-dev git tar unzip wget bzip2 build-essential autoconf libtool libxml2-dev libgeos-dev libgeos++-dev libpq-dev libbz2-dev libproj-dev munin-node munin protobuf-c-compiler libfreetype6-dev libtiff5-dev libicu-dev libgdal-dev libcairo2-dev libcairomm-1.0-dev apache2 apache2-dev libagg-dev liblua5.2-dev ttf-unifont lua5.1 liblua5.1-0-dev
```

```
alisa@linux:~$ sudo apt install libboost-all-dev git tar unzip wget bzip2 build-essential autoconf libtool libxml2-dev libgeos-dev libgeos++-dev libpq-dev libbz2-dev libproj-dev munin-node munin protobuf-c-compiler libfreetype6-dev libtiff5-dev libicu-dev libgdal-dev libcairo2-dev libcairomm-1.0-dev apache2 apache2-dev libagg-dev liblua5.2-dev ttf-unifont lua5.1 liblua5.1-0-dev
[sudo] password for alisa:
```

Подтверждаем

```
1 upgraded, 509 newly installed, 0 to remove and 40 not upgraded.
Need to get 242 MB of archives.
After this operation, 1,246 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Процесс достаточно продолжительный и можно попить кофейку...

Установка postgresql/postgis

В Ubuntu есть предварительно упакованные версии как postgis, так и postgresql, поэтому их можно просто установить через диспетчер пакетов Ubuntu.

```
sudo apt install postgresql postgresql-contrib postgis postgresql-12-postgis-3 postgresql-12-postgis-3-scripts
```

```
alisa@linux:~$ sudo apt install postgresql postgresql-contrib postgis postgresql-12-postgis-3 postgresql-12-postgis-3-scripts
```

Здесь «postgresql» — это база данных, в которой мы собираемся хранить данные карты, а «postgis» добавляет к ней дополнительную графическую поддержку. Снова скажите «Yes» для установки.

```
Need to get 39.7 MB of archives.  
After this operation, 179 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

Создаем базу данных postgis. По умолчанию в различных программах предполагается, что база данных называется gis, и тут мы будем использовать то же соглашение, хотя это и не обязательно. Замените свое имя пользователя вместо alisa, где это используется ниже. Это должно быть имя пользователя, которое будет отображать карты с помощью Mapnik.

```
sudo -u postgres -i
```

```
alisa@linux:~$ sudo -u postgres -i  
postgres@linux:~$
```

```
createuser alisa
```

```
postgres@linux:~$ createuser alisa  
postgres@linux:~$
```

если ошибки нет, то продолжаем установку

Ошибка

```
alisa@map:~$ sudo -u postgres -i  
postgres@map:~$ createuser alisa  
createuser: error: could not connect to database template1: could not connect to  
server: No such file or directory  
Is the server running locally and accepting  
connections on Unix domain socket "/var/run/postgresql/.s.PGSQL.5432"?
```

перезапускаем postgresql

```
sudo service postgresql restart
```

```
alisa@map:~$ sudo service postgresql restart  
alisa@map:~$
```

проверяем

```
pg_isready
```

```
alisa@map:~$ pg_isready  
/var/run/postgresql:5432 - accepting connections  
alisa@map:~$
```

ответ «/var/run/postgresql:5432 - accepting connections»

Продолжаем установку

```
createdb -E UTF8 -O alisa gis
```

```
postgres@linux:~$ createdb -E UTF8 -O alisa gis  
postgres@linux:~$
```

Продолжая работать как пользователь «postgres», настройте PostGIS в б PostgreSQL (опять же, меняя alisa на свое имя пользователя ниже):

psql

```
postgres@linux:~$ psql
psql (12.11 (Ubuntu 12.11-0ubuntu0.20.04.1))
Type "help" for help.
```

```
postgres=# █
```

(ответ «postgres=#»)

\c gis

```
postgres=# \c gis
You are now connected to database "gis" as user "postgres".
gis=# █
```

(ответ: "You are now connected to database 'gis' as user 'postgres'.")

CREATE EXTENSION postgis;

```
gis=# CREATE EXTENSION postgis;
CREATE EXTENSION
gis=# █
```

(ответ CREATE EXTENSION)

CREATE EXTENSION hstore;

```
gis=# CREATE EXTENSION hstore;
CREATE EXTENSION
gis=# █
```

(ответ CREATE EXTENSION)

ALTER TABLE geometry_columns OWNER TO alisa;

```
gis=# ALTER TABLE geometry_columns OWNER TO alisa;
ALTER TABLE
gis=# █
```

(ответ ALTER TABLE)

ALTER TABLE spatial_ref_sys OWNER TO alisa;

```
gis=# ALTER TABLE spatial_ref_sys OWNER TO alisa;
ALTER TABLE
gis=# █
```

(ответ ALTER TABLE)

\q

```
gis=# \q
postgres@linux:~$ █
```

(выход из psql и приглашение к Linux)

exit

```
postgres@linux:~$ exit
logout
alisa@linux:~$
```

(это вернет нас к пользователю, которым мы были до того, как сделали «sudo -u postgres -i» выше)

Если вы еще не создали пользователя, создайте пользователя Unix и для этого пользователя, выбрав пароль при появлении запроса:

```
sudo useradd -m alisa
```

```
alisa@linux:~$ sudo useradd -m alisa
[sudo] password for alisa:
```

```
sudo passwd alisa
```

```
alisa@linux:~$ sudo useradd -m alisa
useradd: user 'alisa' already exists
alisa@linux:~$
```

Опять же, выше замените “alisa” на имя пользователя без регистрации root, которого вы выбрали.

Установка osm2pgsql

Далее мы установим osm2pgsql:

```
sudo apt install osm2pgsql
```

```
alisa@linux:~$ sudo apt install osm2pgsql
```

Ошибок обычно быть не должно

Mapnik

Далее мы установим Mapnik, используя версию по умолчанию в Ubuntu 20.04:

```
sudo apt install autoconf apache2-dev libtool libxml2-dev libbz2-dev
libgeos-dev libgeos++-dev libproj-dev gdal-bin libmapnik-dev mapnik-utils
python3-mapnik python3-psycopg2 python3-yaml
```

```
alisa@linux:~$ sudo apt install autoconf apache2-dev libtool libxml2-dev libbz2-
dev libgeos-dev libgeos++-dev libproj-dev gdal-bin libmapnik-dev mapnik-utils py
thon3-mapnik python3-psycopg2 python3-yaml
```

с подтверждением yes при установке

```
Do you want to continue? [Y/n]
```

Проверим правильность установки Mapnik:

```
python3
```

```
alisa@linux:~$ python3
Python 3.8.10 (default, Jun 22 2022, 20:18:18)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

```
import mapnik
```

```
>>> import mapnik
>>>
```

(Ответ: »> и без ошибок, то библиотека Mapnik была найдена Python.)

```
quit()
```

```
>>> quit()
alisa@linux:~$
```

ВЫХОД

Установка mod_tile и render

- Далее мы установим mod_tile и renderd. «mod_tile» — это модуль Apache, который обрабатывает запросы на тайлы; «render» — это демон, который фактически отображает тайлы, когда «mod_tile» их запрашивает. Мы будем использовать ветку «switch2osm»
- https://github.com/SomeoneElseOSM/mod_tile, которая сама является ответвлением
- https://github.com/openstreetmap/mod_tile, но модифицирована для поддержки Ubuntu 20.04 и с пара других изменений для работы на стандартном сервере Ubuntu, а не на одном из серверов рендеринга OSM.

Скомпилируем исходный код mod_tile

Создадим в домашнем каталоге папку src

```
mkdir ~/src
```

```
alisa@linux:~$ mkdir ~/src
alisa@linux:~$
```

и перейдем в эту папку

```
cd ~/src
```

```
alisa@linux:~$ cd ~/src
alisa@linux:~/src$
```

```
git clone -b switch2osm https://github.com/SomeoneElseOSM/mod_tile.git
```

```
alisa@linux:~/src$ git clone -b switch2osm https://github.com/SomeoneElseOSM/mod_tile.git
```

```
cd mod_tile
```

перейдя в каталог mod_tile

```
alisa@linux:~/src$ cd mod_tile
alisa@linux:~/src/mod_tile$
```

```
./autogen.sh
```

```
alisa@linux:~/src/mod_tile$ ./autogen.sh
```

(это должно закончиться "autoreconf: Leaving directory '.").

```
autoreconf: Leaving directory `.'
alisa@linux:~/src/mod_tile$
```

(в предыдущих установках мне приходилось проделывать это 2 раза для создания папки mod_tile в директории src)

```
/home/alisa/src/
```

Имя	Размер	Изменено	Права
 mod_tile		09.08.2022 0:39:01	rwXr-xr-
		09.08.2022 0:51:28	rwXrwxr-

```
./configure
```

```
alisa@linux:~/src/mod_tile$ ./configure
```

(это должно закончиться "config.status: executing libtool commands")

```
config.status: executing libtool commands
alisa@linux:~/src/mod_tile$
```

```
make
```

```
alisa@linux:~/src/mod_tile$ make
```

Обратите внимание, что здесь некоторые «тревожные» сообщения будут прокручиваться вверх по экрану. Однако это должно заканчиваться на «make[1]: Leaving directory '/home/alisa/src/mod_tile'»

```
a" "mod_tile.la" )
make[1]: Leaving directory '/home/alisa/src/mod_tile'
alisa@linux:~/src/mod_tile$
```

```
sudo make install
```

```
alisa@linux:~/src/mod_tile$ sudo make install
```

(это должно закончиться «make[1]: выход из каталога должно быть '/home/alisa/src/mod_tile'»)

```
make[1]: Leaving directory '/home/alisa/src/mod_tile'
alisa@linux:~/src/mod_tile$
```

```
sudo make install-mod_tile
```

```
alisa@linux:~/src/mod_tile$ sudo make install-mod_tile
```

(это должно закончиться «chmod 644 /usr/lib/apache2/modules/mod_tile.so»)

```
chmod 644 /usr/lib/apache2/modules/mod_tile.so
alisa@linux:~/src/mod_tile$
```

```
sudo ldconfig
```

```
alisa@linux:~/src/mod_tile$ sudo ldconfig
```

(без ответа)

```
alisa@linux:~/src/mod_tile$ sudo ldconfig
```

```
alisa@linux:~/src/mod_tile$
```

Конфигурация таблицы стилей

- Теперь, когда все необходимое программное обеспечение установлено, вам нужно загрузить и настроить таблицу стилей.
- Здесь мы будем использовать тот же стиль, что и на «стандартной» карте на веб-сайте openstreetmap.org.
- Переходим в домашний каталог.

```
cd ~/src
```

```
alisa@linux:~/src/mod_tile$ cd ~/src
```

```
alisa@linux:~/src$
```

```
git clone https://github.com/gravitystorm/openstreetmap-carto
```

```
alisa@linux:~/src$ git clone https://github.com/gravitystorm/openstreetmap-carto
```

клонировем репозиторий

```
Resolving deltas: 100% (12139/12139), done.
```

```
alisa@linux:~/src$
```

переходим в каталог openstreetmap-carto

```
cd openstreetmap-carto
```

```
alisa@linux:~/src$ cd openstreetmap-carto
```

установим подходящую версию компилятора «carto».

```
sudo apt install npm
```

```
alisa@linux:~/src/openstreetmap-carto$ sudo apt install npm
```

подтверждаем установку «у»

```
Do you want to continue? [Y/n] y
```

размер в 600mb, т.ч. перекур

```
sudo npm install -g carto
```

```
alisa@linux:~/src/openstreetmap-carto$ sudo npm install -g carto
```

```
carto -v
```

```
added 64 packages from 29 contributors in 139.725s
```

```
alisa@linux:~/src/openstreetmap-carto$ carto -v
```

Это должно ответить номером версии, которая должна быть не меньше: 1.2.0

```
1.2.0
```

```
alisa@linux:~/src/openstreetmap-carto$
```

преобразуем проект carto во что-то, понятное Mapnik:

```
carto project.mml > mapnik.xml
```

```
alisa@linux:~/src/openstreetmap-carto$ carto project.mml > mapnik.xml
```

Теперь у вас есть таблица стиля Mapnik XML в /home/alisa/src/openstreetmap-carto/mapnik.xml.

```
/home/alisa/src/openstreetmap-carto/
```

Имя	Размер	Изменено	Пра
mapnik.xml	2 679 KB	09.08.2022 1:57:24	rw-r
openstreetmap-carto.lua	13 KB	09.08.2022 1:33:23	rw-r

Загрузка данных

в домашнем каталоге создаем папку data

```
mkdir ~/data
```

```
alisa@linux:~/src/openstreetmap-carto$ mkdir ~/data
```

и переходим в эту папку

```
cd ~/data
```

```
alisa@linux:~/src/openstreetmap-carto$ cd ~/data
```

на странице загрузки <http://download.geofabrik.de/> выбираем нужную нам карту для и копируем с кнопки ее ссылку

```
wget http://download.geofabrik.de/russia-latest.osm.pbf
```

```
alisa@linux:~/data$ wget http://download.geofabrik.de/russia-latest.osm.pbf
```

- карта России размером 3,2Gb, процесс не быстрый, идем гулять пол часика с собачкой...
- [Слияние карт pbf](#)

```
Length: 3338100954 (3.1G) [application/octet-stream]  
Saving to: 'russia-latest.osm.pbf'
```

```
russia-latest.osm.p 0%[ ] 10.79M 1.94MB/s eta 30m 23s
```

У кого собачки нет, том может для примера загрузить карту Азербайджана размером 32Mb...

```
wget https://download.geofabrik.de/asia/azerbaijan-latest.osm.pbf
```

Следующая команда вставит загруженные ранее данные OpenStreetMap в базу данных. Этот шаг очень интенсивно использует дисковый ввод-вывод; импорт всей планеты может занять много часов, дней или недель в зависимости от аппаратного обеспечения.

```
cd ~/data
```

```
osm2pgsql -d gis --create --slim -G --hstore --tag-transform-script
~/src/openstreetmap-carto/openstreetmap-carto.lua -C 4000 --number-processes
2 -S ~/src/openstreetmap-carto/openstreetmap-carto.style ~/data/russia-
latest.osm.pbf
```

```
alisa@linux:~/data$ cd ~/data
alisa@linux:~/data$ osm2pgsql -d gis --create --slim -G --hstore --tag-transfor
m-script ~/src/openstreetmap-carto/openstreetmap-carto.lua -C 4000 --number-proc
esses 2 -S ~/src/openstreetmap-carto/openstreetmap-carto.style ~/data/russia-lat
est.osm.pbf
```

Обратите внимание на параметр -c (-create) при импорте через osm2pgsql. С этим параметром удалится вся существующая информация с таблиц.

```
cd ~/data
osm2pgsql -d gis --append --slim -G --hstore --tag-transform-script
~/src/openstreetmap-carto/openstreetmap-carto.lua -C 4000 --number-processes
2 -S ~/src/openstreetmap-carto/openstreetmap-carto.style ~/data/russia-
latest.osm.pbf
```

Если вам необходимо сделать импорт ещё нескольких стран, то вместо -c указывайте параметр -a (-append).

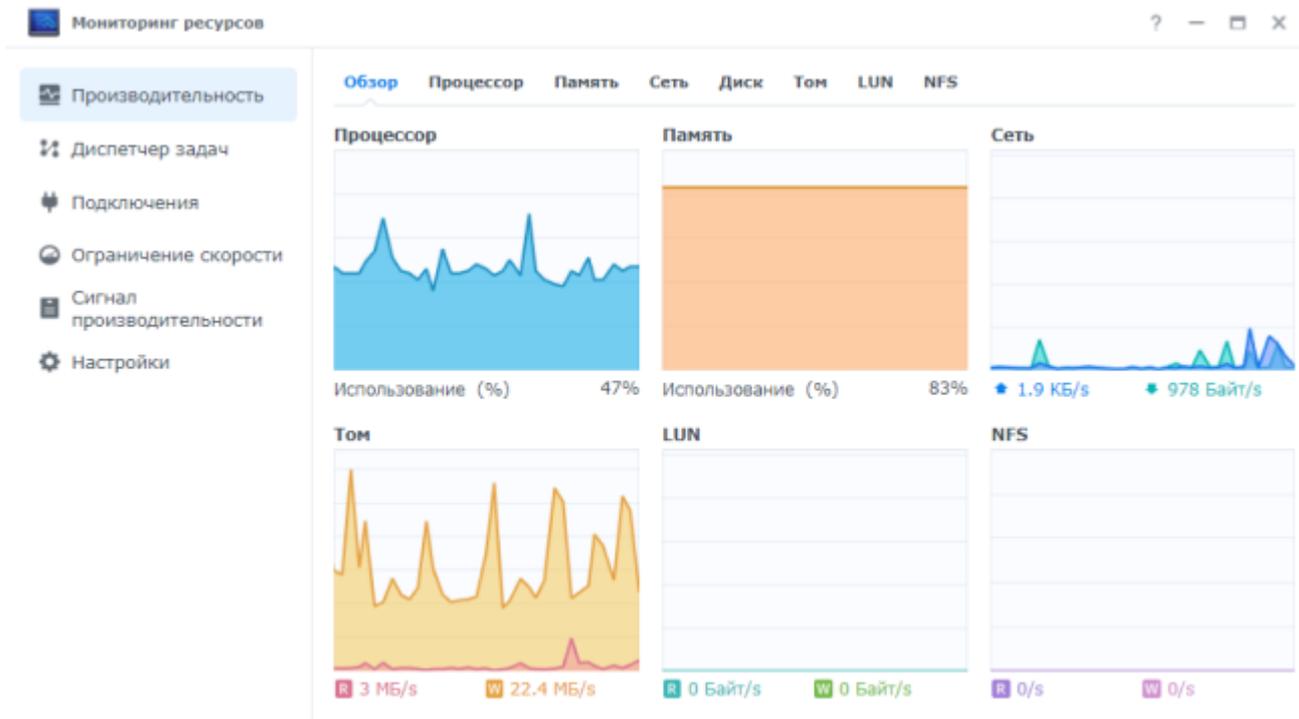
Командные опции:

- **-d gis**
 - База данных, с которой нужно работать (раньше gis была по умолчанию, теперь нужно указать).
- **-create**
 - Загружает данные в пустую базу данных, а не пытается добавить их в существующую (иначе -append).
- **-slim**
 - osm2pgsql может использовать разные макеты таблиц; «тонкие» таблицы работают для рендеринга.
- **-G**
 - Определяет, как обрабатываются мультиполигоны.
- **-hstore**
 - Позволяет тегам, для которых нет явных столбцов базы данных, использоваться для рендеринга.
- **-tag-transform-script**
 - Определяет сценарий lua, используемый для обработки тегов. Это простой способ обработки тегов OSM до того, как их обработает сам стиль, что значительно упрощает логику стиля.
- **-C 4000**
 - Выделено 4,0 Гб памяти osm2pgsql для процесса импорта. Если у вас меньше памяти, вы можете попробовать меньшее число, а если процесс импорта завершится из-за нехватки памяти, вам нужно будет попробовать меньшее число или меньший размер карты OSM.
- **-number-processes 2**
 - Используйте 2 ЦП. Если у вас есть больше доступных ядер, вы можете использовать больше.
- **-S**
 - Создайте столбцы базы данных в этом файле (на самом деле они не изменились по

сравнению с «openstreetmap-carto»)

- `~/src/openstreetmap-carto/openstreetmap-carto.style ~/data/russia-latest.osm.pbf`
- Место сохранения и файл данных для загрузки.

смотрим монитор ресурсов Synology NAS



- Команда загрузки должна завершится чем-то вроде «Osm2pgsql в целом занял 34583 секунд».

```
Osm2pgsql took 34583s overall
node cache: stored: 273576931(58.20%), storage efficiency: 52.18% (dense blocks:
4627, sparse nodes: 243191809), hit rate: 59.43%
alisa@linux:~/data$
```

- Пока идет загрузка карты размером 3,2gb (которая по факту заняла 07 часов 23 минуты), перейдем к разделу Шрифты, открыв второе окно подключения к серверу.

Создание индексов

- Начиная с версии v5.3.0 некоторые дополнительные индексы теперь необходимо применять вручную .
- переходим в папку openstreetmap-carto

```
cd ~/src/openstreetmap-carto/
```

```
alisa@linux:~/data$ cd ~/src/openstreetmap-carto/
```

- запускаем команду создание индексов

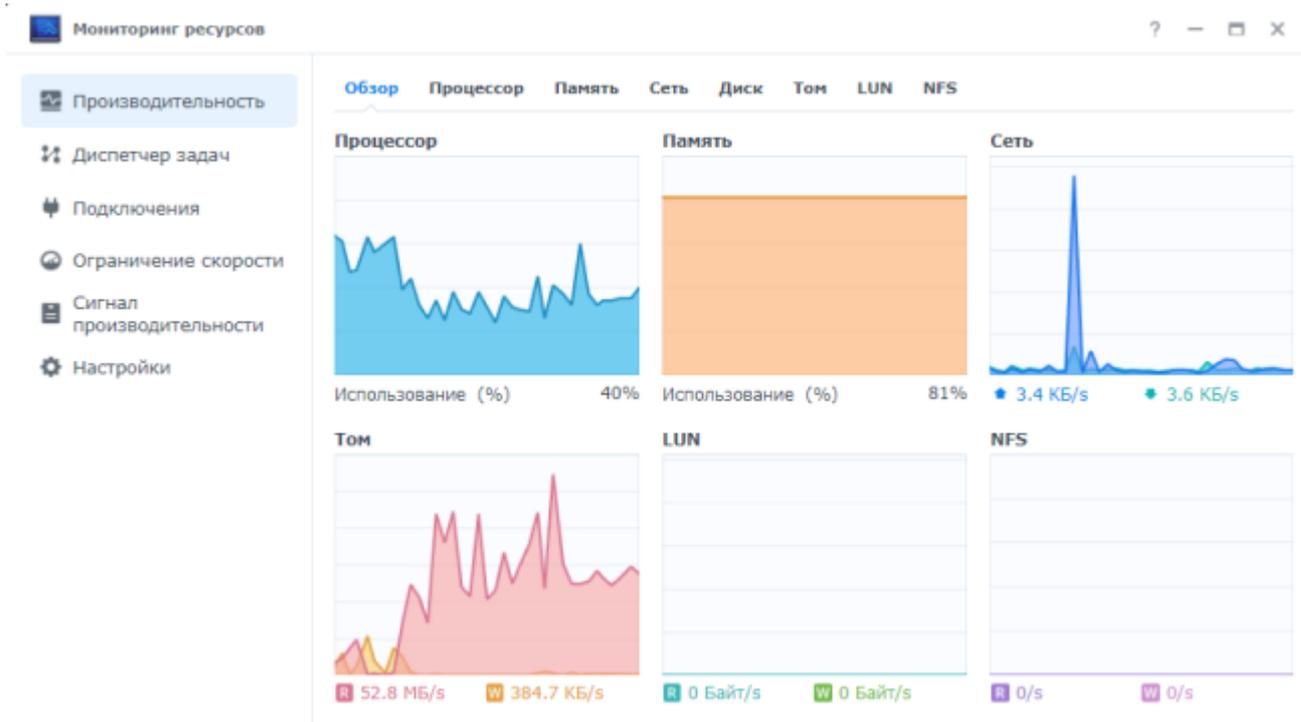
```
psql -d gis -f indexes.sql
```

```
alisa@linux:~/src/openstreetmap-carto$ psql -d gis -f indexes.sql
```

- (ответ: «CREATE INDEX» 15 раз.)

```
alisa@linux:~/src/openstreetmap-carto$ psql -d gis -f indexes.sql
CREATE INDEX
alisa@linux:~/src/openstreetmap-carto$
```

- заняло примерно 29 минут с ожидаемой нагрузкой



Загрузка Shape-файла

- Shape-файл — векторный формат для хранения объектов, описываемых геометрией и сопутствующими атрибутами.
- Несмотря на то, что большая часть данных для создания карты, берется из данных OpenStreetMap, загружается не все, некоторые шейп-файлы для таких вещей, как границы стран с малым масштабированием, по-прежнему необходимы. Чтобы загрузить и проиндексировать их:
- переходим в папку openstreetmap-carto

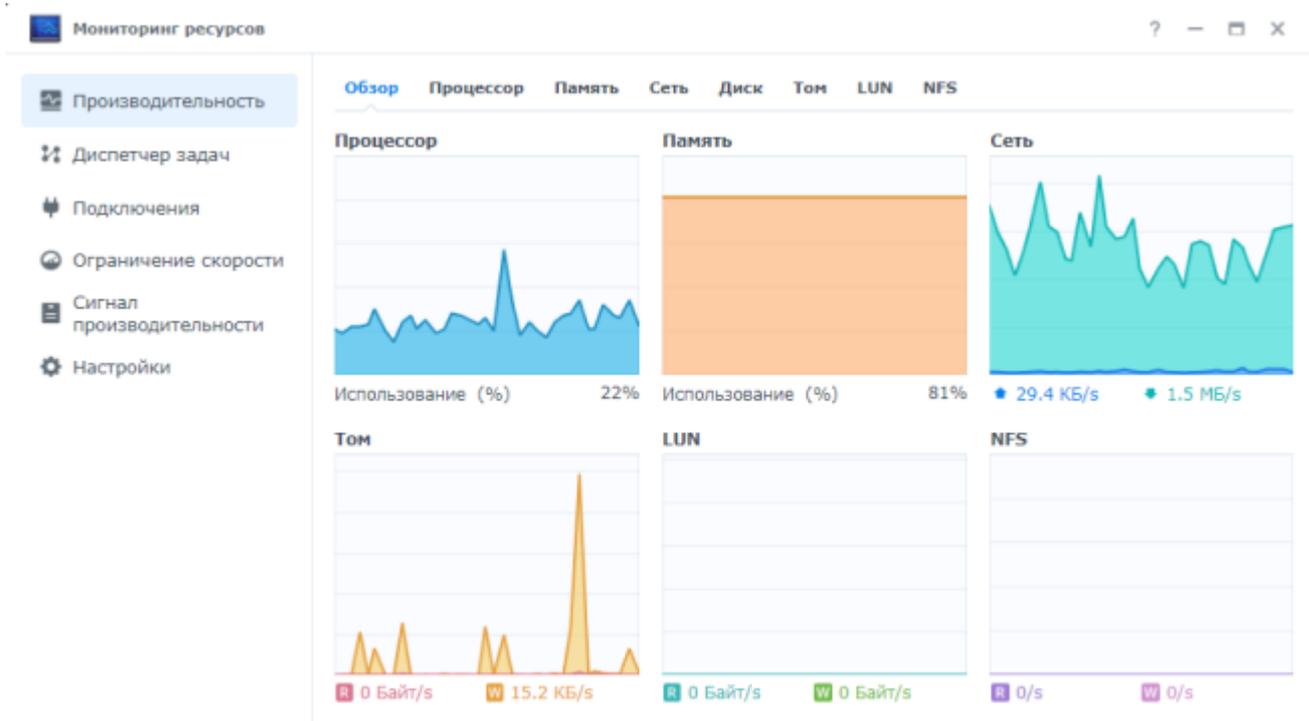
```
cd ~/src/openstreetmap-carto/
```

```
alisa@linux:~/src/openstreetmap-carto$ cd ~/src/openstreetmap-carto/
```

```
scripts/get-external-data.py
```

```
alisa@linux:~/src/openstreetmap-carto$ scripts/get-external-data.py
```

- Этот процесс включает в себя загрузку и требует некоторого времени — во время работы на экране мало что освещает. На самом деле он загружает таблица водных полигонов и прочее и заполняет каталог «data» ниже «openstreetmap-carto».
- это занято примерно 47 минут при прогнозируемой загрузке системы



Шрифты

переходим в каталог openstreetmap-carto в папке src

```
cd ~/src/openstreetmap-carto/
```

```
alisa@linux:~$ cd ~/src/openstreetmap-carto/
```

и собственно загружаем шрифты

```
scripts/get-fonts.sh
```

```
alisa@linux:~$ cd ~/src/openstreetmap-carto/  
alisa@linux:~/src/openstreetmap-carto$ scripts/get-fonts.sh
```

процесс занял 6 минут

						Dload	Upload	Total	Spent	Left	Speed
0	0	0	0	0	0	0	0	0	--:--:--	--:--:--	0
100	29.1M	100	29.1M	0	0	606k	0	0:00:49	0:00:49	--:--:--	756k

```
alisa@linux:~/src/openstreetmap-carto$
```

Настройка веб-сервера

визуализация

Отрываем с помощью текстового редактора vi файл конфигурации для «render» — «/usr/local/etc/renderd.conf»

```
sudo vi /usr/local/etc/renderd.conf
```

- нажимаем **i** и редактируем строку **XML=/home/renderaccount/src/openstreetmap-carto/mapnik.xml** меняя **renderaccount** на имя своего пользователя (в нашем случае это - **alisa**)
- нажимаем последовательно **esc** , **shift + :** , **w** , **q** , **!** для сохранения и выхода из редактора

```
[renderd]
num_threads=4
tile_dir=/var/lib/mod_tile
stats_file=/var/run/renderd/renderd.stats

[mapnik]
plugins_dir=/usr/lib/mapnik/3.0/input
font_dir=/usr/share/fonts/truetype
font_dir_recurse=1

[ajt]
URI=/hot/
TILEDIR=/var/lib/mod_tile
XML=/home/alisa/src/openstreetmap-carto/mapnik.xml
HOST=localhost
TILESIZE=256
MAXZOOM=20

~
~
~
~
:wq!
```

- нажимаем **enter**

Настройка Apache

создаем папку mod_tile

```
sudo mkdir /var/lib/mod_tile
```

```
alisa@linux:~/src/openstreetmap-carto$ sudo mkdir /var/lib/mod_tile
```

задаем права пользователя для этой папки

```
sudo chown alisa /var/lib/mod_tile
```

```
alisa@linux:~/src/openstreetmap-carto$ sudo mkdir /var/lib/mod_tile
alisa@linux:~/src/openstreetmap-carto$ sudo chown alisa /var/lib/mod_tile
alisa@linux:~/src/openstreetmap-carto$
```

создаем папку renderd

```
sudo mkdir /var/run/renderd
```

```
alisa@linux:~/src/openstreetmap-carto$ sudo mkdir /var/run/renderd
```

задаем права пользователя для этой папки

```
sudo chown alisa /var/run/renderd
```

```
alisa@linux:~/src/openstreetmap-carto$ sudo mkdir /var/run/renderd
alisa@linux:~/src/openstreetmap-carto$ sudo chown alisa /var/run/renderd
alisa@linux:~/src/openstreetmap-carto$
```

сообщаем Apache о «mod_tile» редактируя файл mod_tile.conf добавляя следующую строку
LoadModule tile_module /usr/lib/apache2/modules/mod_tile.so

```
sudo vi /etc/apache2/conf-available/mod_tile.conf
```

```
alisa@linux:~/src/openstreetmap-carto$ sudo vi /etc/apache2/conf-available/mod_t
ile.conf
```

- нажимаем **i** и вставляем строку **LoadModule tile_module /usr/lib/apache2/modules/mod_tile.so** в тело файла (в нашем случае это пустой файл)
- нажимаем последовательно **esc**, **shift + :**, **w**, **q**, **!** для сохранения и выхода из редактора

```
LoadModule tile_module /usr/lib/apache2/modules/mod_tile.so
~
~
~
~
:wq!
```

- нажимаем **enter**

выполняем команду

```
sudo a2enconf mod_tile
```

```
alisa@linux:~/src/openstreetmap-carto$ sudo a2enconf mod_tile
```

- сообщение о необходимости перезапуска Apache «systemctl reload apache2», чтобы активировать новую конфигурацию; мы пока не будем этого делать.

```
alisa@linux:~/src/openstreetmap-carto$ sudo a2enconf mod_tile
Enabling conf mod_tile.
To activate the new configuration, you need to run:
systemctl reload apache2
alisa@linux:~/src/openstreetmap-carto$
```

- сообщаем Apache о «renderd»
- открываем в редакторе файл 000-default.conf

```
sudo vi /etc/apache2/sites-available/000-default.conf
```

```
alisa@linux:~/src/openstreetmap-carto$ sudo vi /etc/apache2/sites-available/000-
default.conf
```

- вставляем следующие строки между строками «[ServerAdmin](#)» и «[DocumentRoot](#)»

```
LoadTileConfigFile /usr/local/etc/renderd.conf
ModTileRenderdSocketName /var/run/renderd/renderd.sock
# Timeout before giving up for a tile to be rendered
ModTileRequestTimeout 0
# Timeout before giving up for a tile to be rendered that is otherwise
missing
ModTileMissingRequestTimeout 30
```

- нажимаем **i** и вставляем строки в тело файла между строками «[ServerAdmin](#)» и «[DocumentRoot](#)»
- нажимаем последовательно **esc** , **shift + :** , **w** , **q** , **!** для сохранения и выхода из редактора

```
ServerAdmin webmaster@localhost
LoadTileConfigFile /usr/local/etc/renderd.conf
ModTileRenderdSocketName /var/run/renderd/renderd.sock
# Timeout before giving up for a tile to be rendered
ModTileRequestTimeout 0
# Timeout before giving up for a tile to be rendered that is otherwise missing
ModTileMissingRequestTimeout 30
    DocumentRoot /var/www/html

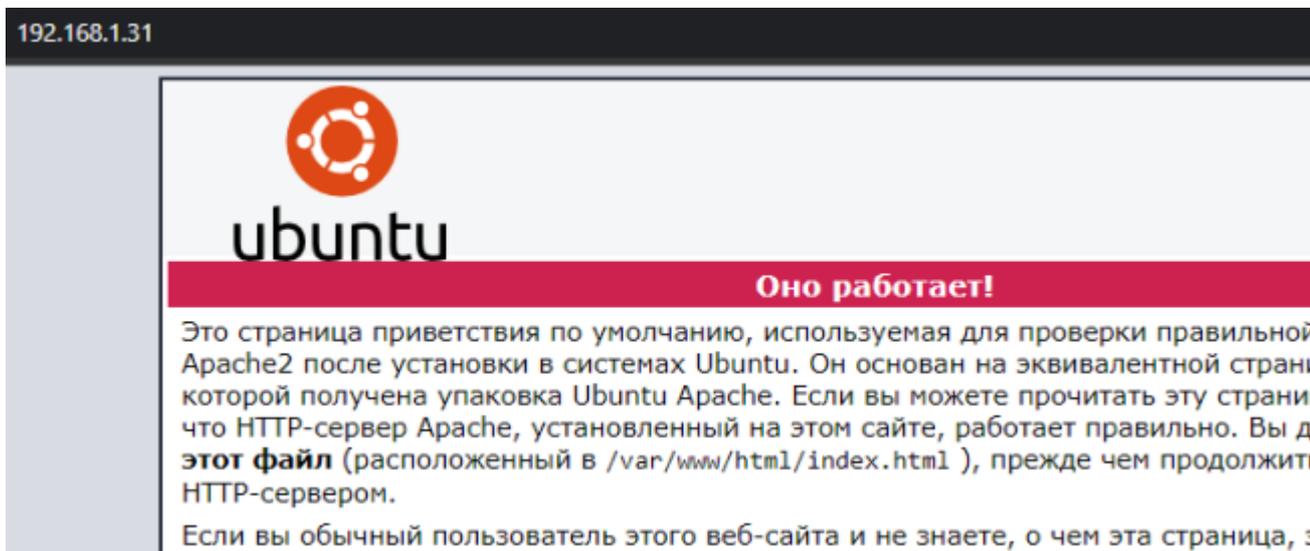
    # Available loglevels: trace8, ..., tracel, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
:wq!
```

- нажимаем **enter**
- дважды перезагрузите apache:

```
sudo service apache2 reload
sudo service apache2 reload
```

```
alisa@linux:~/src/openstreetmap-carto$ sudo service apache2 reload
Job for apache2.service failed.
See "systemctl status apache2.service" and "journalctl -xe" for details.
alisa@linux:~/src/openstreetmap-carto$ sudo service apache2 reload
Job for apache2.service failed.
See "systemctl status apache2.service" and "journalctl -xe" for details.
alisa@linux:~/src/openstreetmap-carto$
```

- проверим наш Apache набрав в браузере адрес нашего виртуального сервера (в нашем случае: <http://192.168.1.31/>)



- **Дальше не продолжаем если не дождалась загрузки данных в базу и Создание индексов**

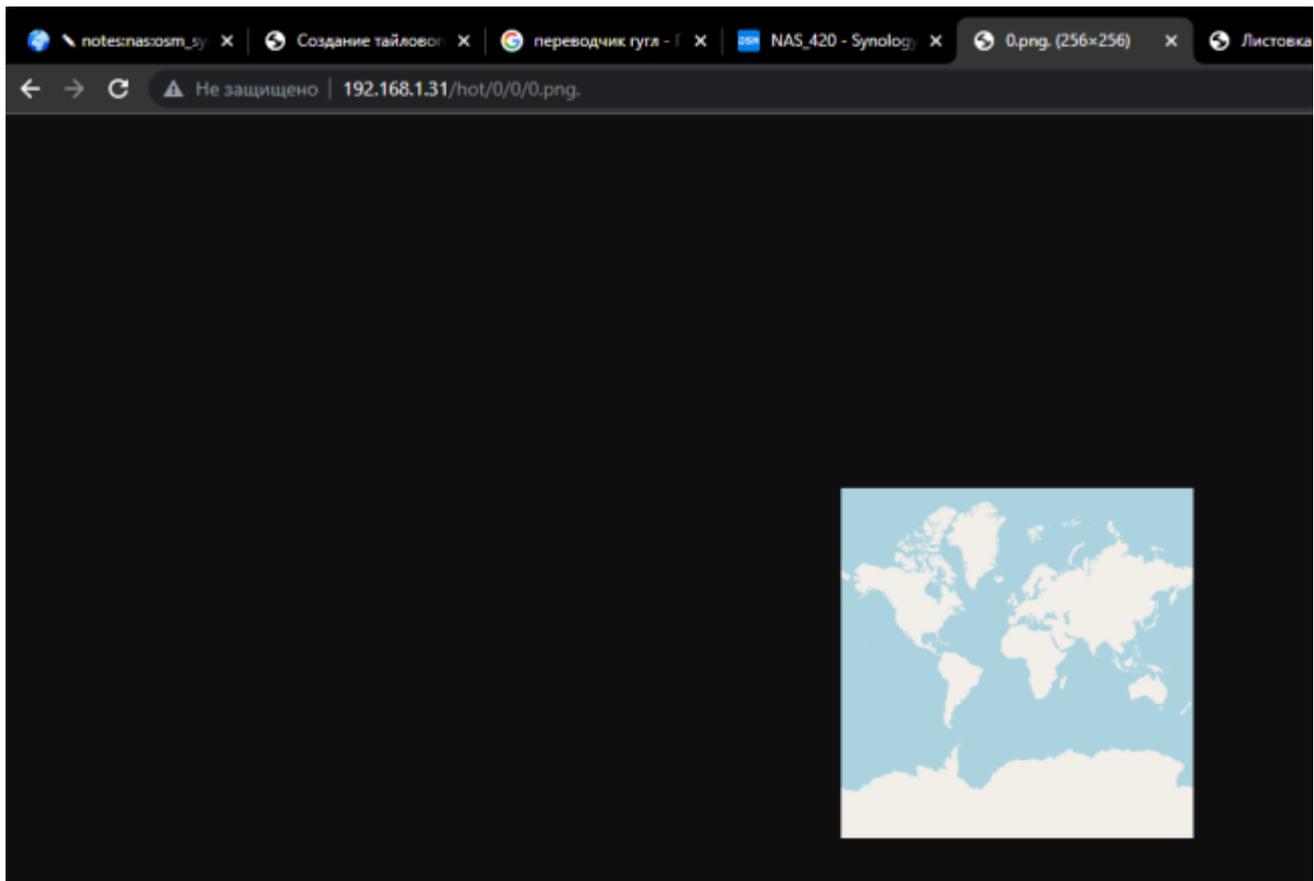
Запуск рендера в первый раз

запустим renderd, чтобы попытаться отрендерить некоторые плитки. Сначала мы запустим его на переднем плане, чтобы видеть любые ошибки по мере их возникновения:

```
sudo -u 'alisa' renderd -f -c /usr/local/etc/renderd.conf
```

```
alisa@linux:~/src/openstreetmap-carto$ sudo -u 'alisa' renderd -f -c /usr/local/
etc/renderd.conf
```

- Откройте веб-браузер по адресу: <http://yourserveripaddress/hot/0/0/0.png> (в нашем случае <http://192.168.1.31/hot/0/0/0.png>)



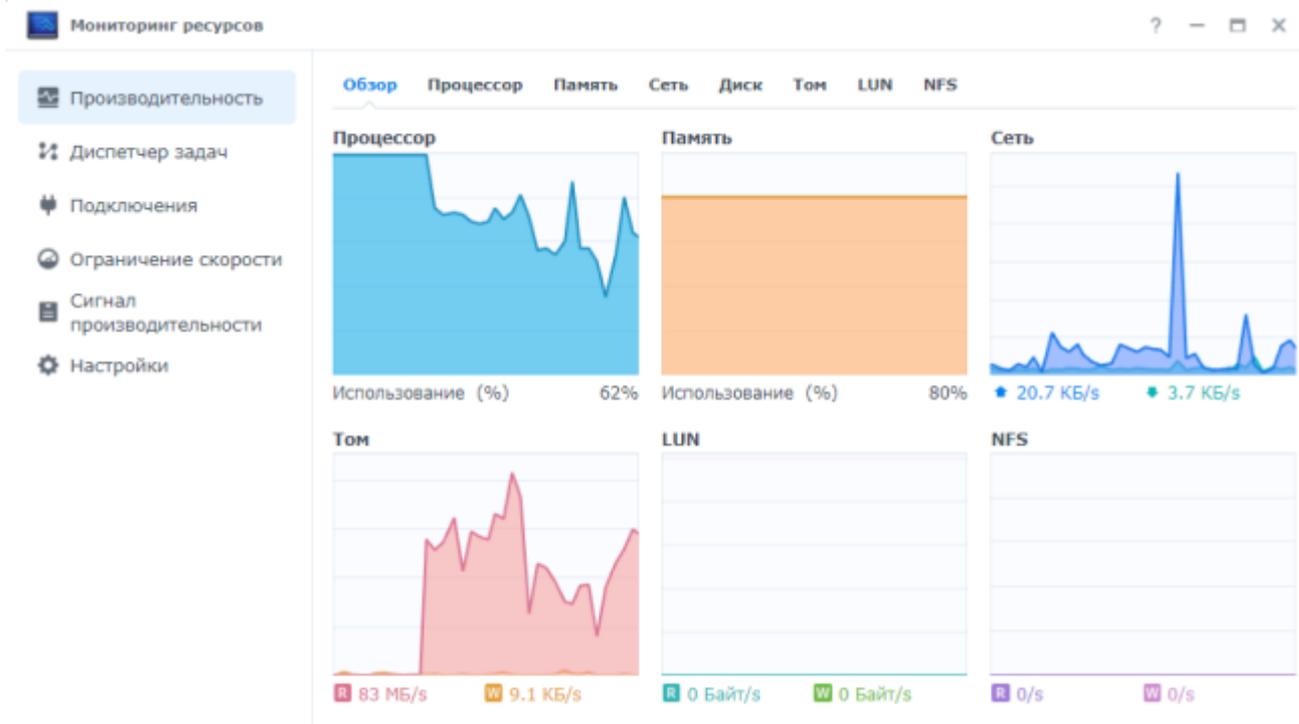
- Если все это работает, нажмите **control-c**, чтобы завершить процесс рендеринга переднего плана.
- **при возникновении необходимости остановить renderd**

```
systemctl stop renderd
```

- иногда рендер идет с командой, т.е. без `sudo -u 'alisa'`

```
renderd -f -c /usr/local/etc/renderd.conf
```

- процессор был недолго в шоке
- на Synology DS720+ такого шока нет и все происходит быстрее в 3-4 раза



- Не знаю почему, но при наличии папки `renderd` и прав на неё, система ругается на её отсутствие (проверено при 3х установках на Synology Nas. Повторно выполним эти команды...
- создаем папку `renderd`

```
sudo mkdir /var/run/renderd
```

```
alisa@linux:~/src/openstreetmap-carto$ sudo mkdir /var/run/renderd
```

задаем права пользователя для этой папки

```
sudo chown alisa /var/run/renderd
```

```
alisa@linux:~/src/openstreetmap-carto$ sudo mkdir /var/run/renderd
alisa@linux:~/src/openstreetmap-carto$ sudo chown alisa /var/run/renderd
alisa@linux:~/src/openstreetmap-carto$
```

Запуск рендера в фоновом режиме

Далее мы настроим «рендеринг» для работы в фоновом режиме. Отредактируем файл «`~/src/mod_tile/debian/renderd.init`», чтобы «`RUNASUSER`» был установлен для учетной записи без имени `root`, который мы использовали ранее, например «`alisa`», и копируем его в системный каталог. откроем файл в редакторе

```
vi ~/src/mod_tile/debian/renderd.init
```

- нажимаем **i** и меняем в строке `RUNASUSER` **renderaccount** на ваше имя пользователя (в нашем случае это `alisa`)
- нажимаем последовательно **esc**, **shift + :**, **w**, **q**, **!** для сохранения и выхода из редактора

```
# Short-Description: Mapnik rendering daemon
# Description:      Mapnik rendering daemon.
### END INIT INFO

# Do NOT "set -e"

# PATH should only include /usr/* if it runs after the mountnfs.sh script
PATH=/sbin:/usr/sbin:/bin:/usr/bin
DESC="Mapnik rendering daemon"
NAME=renderd
DAEMON=/usr/local/bin/$NAME
DAEMON_ARGS="-c /usr/local/etc/renderd.conf"
PIDSOCKDIR=/var/run/$NAME
PIDFILE=$PIDSOCKDIR/$NAME.pid
SCRIPTNAME=/etc/init.d/$NAME
RUNASUSER=alisa

# Exit if the package is not installed
[ -x "$DAEMON" ] || exit 0

# Read configuration variable file if it is present
[ -r /etc/default/$NAME ] && . /etc/default/$NAME

:wq!
```

- нажимаем **enter**
- скопируем это в системный каталог

```
sudo cp ~/src/mod_tile/debian/renderd.init /etc/init.d/renderd
```

```
alisa@linux:~$ sudo cp ~/src/mod_tile/debian/renderd.init /etc/init.d/renderd
```

```
sudo chmod u+x /etc/init.d/renderd
```

```
alisa@linux:~$ sudo chmod u+x /etc/init.d/renderd
```

```
sudo cp ~/src/mod_tile/debian/renderd.service /lib/systemd/system/
```

```
alisa@linux:~$ sudo cp ~/src/mod_tile/debian/renderd.service /lib/systemd/system/
```

- Файл «render.service» является служебным файлом «systemd». Используемая здесь версия просто вызывает команды инициализации в старом стиле. Чтобы проверить, что команда запуска работает:

```
sudo /etc/init.d/renderd start
```

```
alisa@linux:~$ sudo /etc/init.d/renderd start
```

- (ответить «Starting renderd (через systemctl): renderd.service».)

```
Starting renderd (via systemctl): renderd.service.
alisa@linux:~$
```

- Чтобы это запускалось автоматически каждый раз:

```
sudo systemctl enable renderd
```

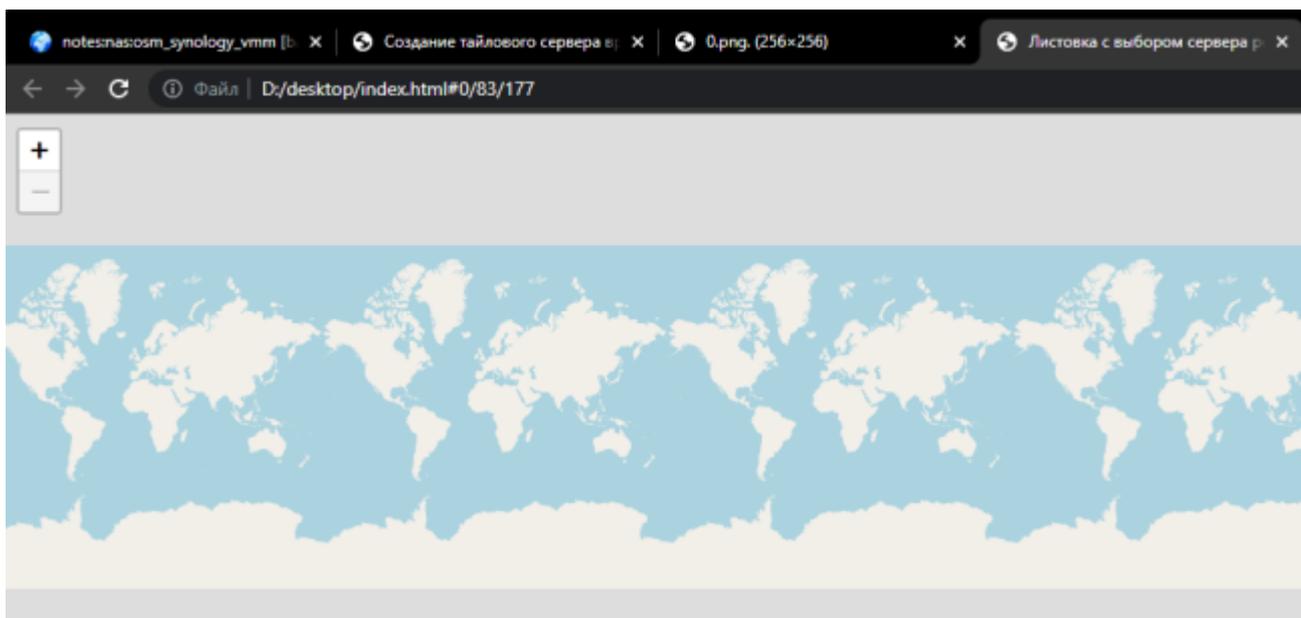
```
alisa@linux:~$ sudo systemctl enable renderd
```

Просмотр тайлов (плиток)

для просмотра плитки, используйте html-файл «sample_leaflet.html» в папке «extra» mod_tile или загрузите zip архив с этим файлом внизу страницы. Просто откройте его в веб-браузере на машине, где вы установили тайловый сервер. Если это невозможно, потому что вы устанавливаете на сервер без локального веб-браузера, вы можете отредактировать его, заменив «127.0.0.1» IP-адресом сервера и скопировав его ниже «/var/www/html». Мы запускаем его по сети с указанием IP-адреса виртуального сервера NAS Synology (192.168.1.31)

Пример загрузки карты

- Начало работы и загрузка 0 level zoom карты России

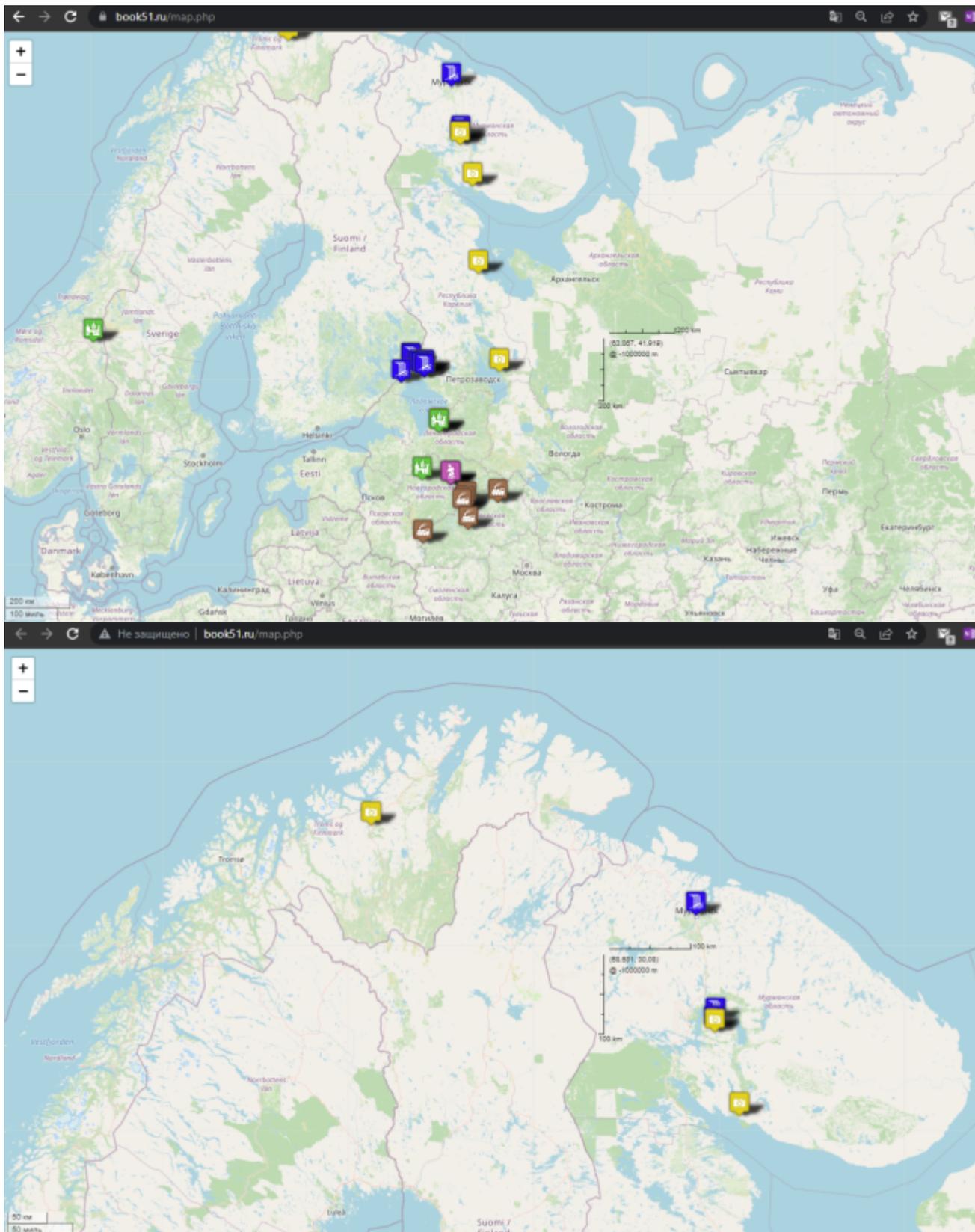


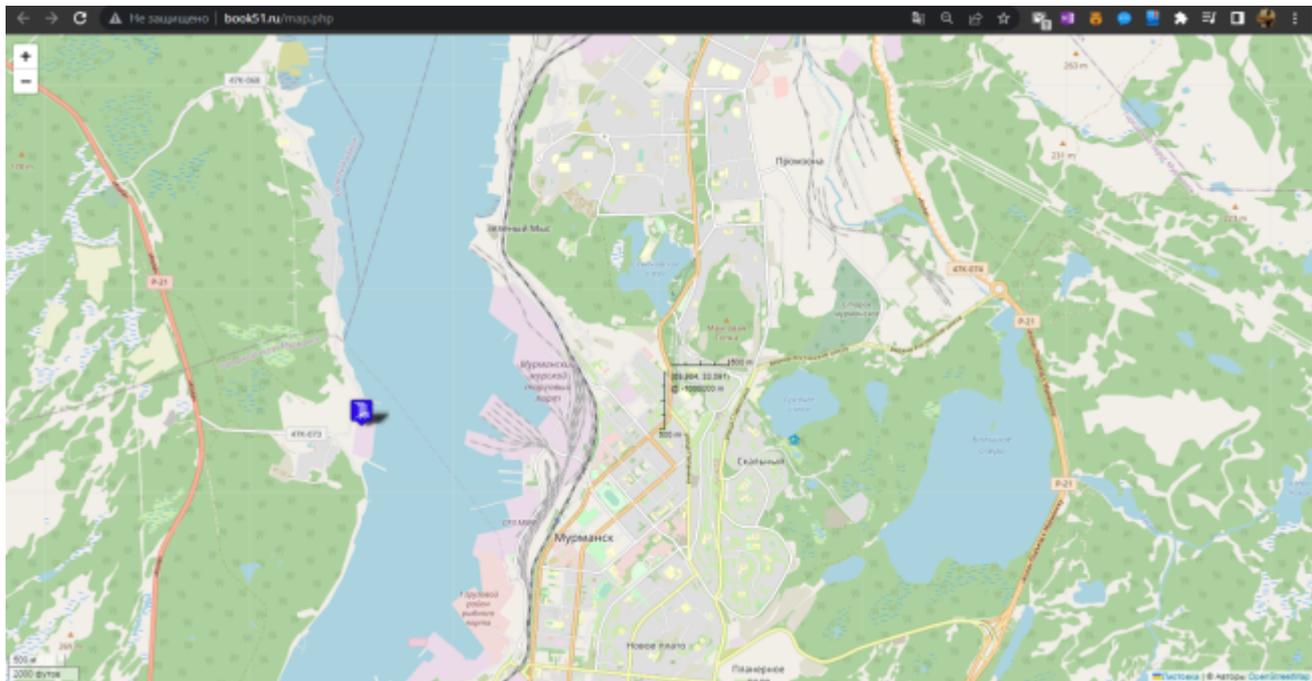
- маленькое видео, как за 1 час происходит визуальный рендеринг 9 level zoom карты России на 3,2gb на устройстве Synology Nas DS420+. Стоит перематывать и смотреть загрузку процессора, так как память неизменна в 80%. Уровни с 1 по 4 пишутся в кэш за 15-20 минут... 8 уровень около 30 минут... Ну а 9 на видео.. немногим более часа визуальный рендеринг карты России за 1 час на 9 уровне зума. Напомню что установлена дополнительная планка памяти на 8gb.

[zoom_9.mp4](#)

- в принципе, считаю проделанную работу, не плохим результатом! И OSM Tale Server имеет право существовать на устройстве Synology NAS DS 420+ для личного пользования или небольшой группы пользователей личного сервера OSM.
- Напомню, что данный Ubuntu Tile Server установлен на виртуальную машину Synology VMM на устройстве DS420+, где работает все, что было установлено ранее. Попробую

переключить мой личный сайт <https://book51.ru/> с картой и слоями на вновь созданный виртуальный сервер.





- Загрузка Synology DS420+ при работающем OSM Tile Server на Synology VMM (8gb VMM не дает установить система, поэтому стоит 6gb)

The screenshot displays the Synology DSM interface. A 'Мониторинг ресурсов' (Resource Monitoring) window is open, showing the following data:

Resource	Usage (%)	Current Rate
Процессор (CPU)	95%	245.1 MB/s
Память (Memory)	81%	0 Байт/s
Сеть (Network)	-	25.2 КБ/s
Тон (iSCSI)	-	0 Байт/s
LUN	-	0 Байт/s
NFS	-	0/s

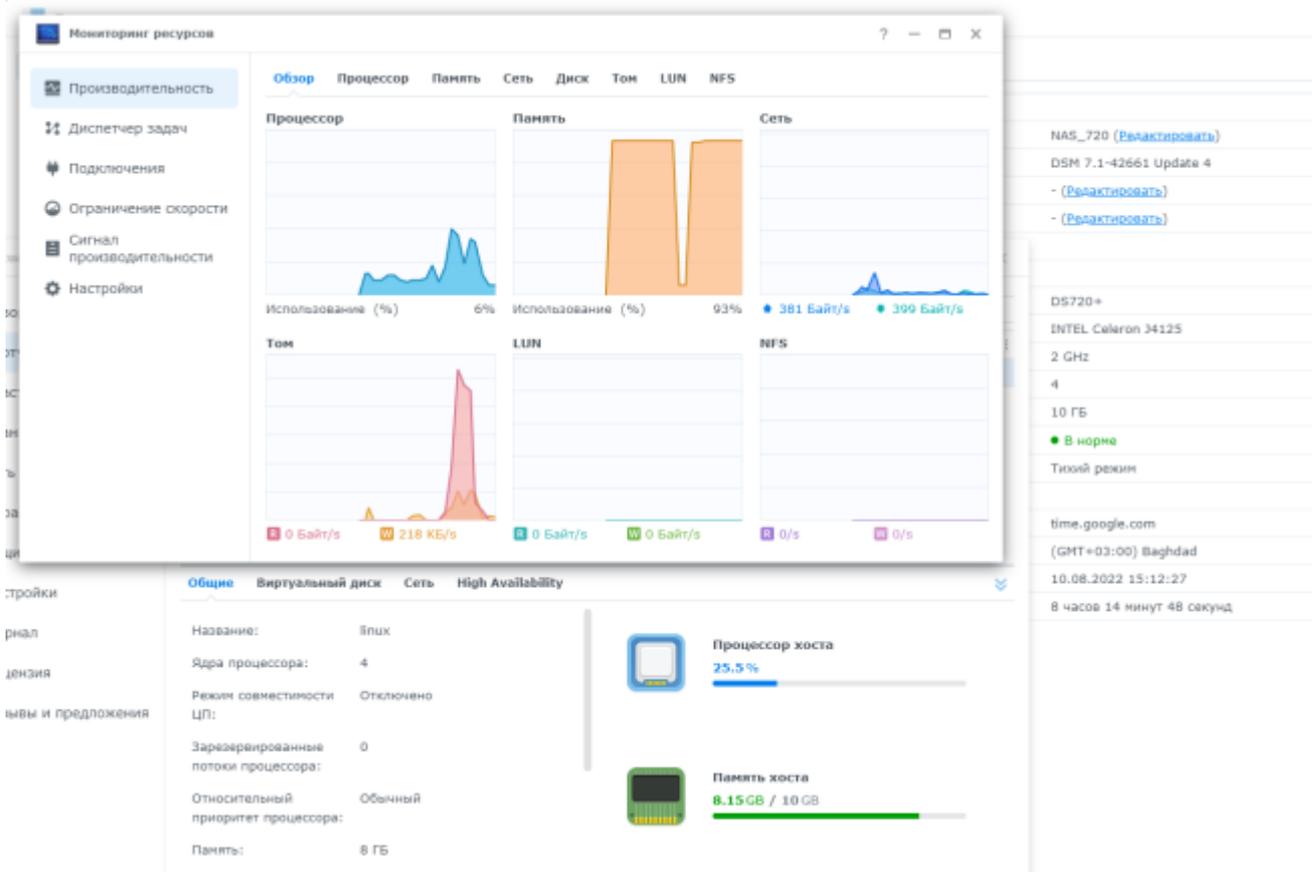
Below the monitoring window, the system information for the 'Ubuntu Server' virtual machine is visible:

- Название: Ubuntu Server
- Ядра процессора: 2
- Режим совместимости ЦП: Отключено
- Зарезервированные потоки процессора: 0
- Относительный приоритет процессора: Обычный
- Память: 6 GB

On the right side, the hardware specifications for the Synology DS420+ are listed:

- Model: NAS_420
- OSM: DSM 7.1-42661 Update 3
- Processor: INTEL Celeron J4025
- Frequency: 2 GHz
- Cache: 2
- Memory: 10 GB
- Power: 6 Вт (в норме)
- Power Mode: Тихий режим
- USB 3.0 Hub: VIA Labs, Inc.
- Network: NS105X - Norelsys
- Storage: Elements Portable (WDBUZ0) - Western Digital Tec
- USB 2.0 Hub: VIA Labs, Inc.
- Storage: SCC-3400 Series - Samsung Electronics Co., Ltd
- Power: BS650E - Cyber Power System, Inc.

- Для сравнения аналогичная работа OSM Tile Server на Synology VMM на Synology DS720+ и загрузка процессора



Файлы для загрузки

- Ubuntu Server 20.04.2 LTS
 - html-файл для просмотра плиток
 - Mod_tile: модуль веб-сервера Apache
- Leaflet — библиотека для интерактивных карт.
- OSM Carto: стиль стандартного слоя листов OSM
 - water-polygons-split-3857.zip
 - ne_110m_admin_0_boundary_lines_land.zip
 - antarctica-icesheet-outlines-3857.zip
 - antarctica-icesheet-polygons-3857.zip
 - simplified-water-polygons-split-3857.zip

Карты OSM

- azerbaijan-latest.osm.pbf
- russia-latest.osm.pbf
- planet-240506.osm.pbf

Ссылки и литература

[Создание тайлового сервера вручную \(20.04 LTS\)](#)

[Ubuntu 20.04 LTS: пользовательская документация](#)
[OSM2PGSQL: импортер данных OSM в базу данных PostgreSQL/PostGIS](#)
[PostgreSQL: база данных с открытым исходным кодом](#)
[Mapnik: отрисовка основного Slippy Map слоя для OSM](#)
[OSM Carto: стиль стандартного слоя листов OSM](#)
[Mod_tile: передача кэшированных тайлов](#)
[Leaflet — библиотека JS для интерактивных карт.](#)
[Synology Open Street Map Server](#)

From:

<http://vladpolskiy.ru/> - **book51.ru**

Permanent link:

http://vladpolskiy.ru/doku.php?id=software:nas:osm_synology_vmm_ds_2422&rev=1722844567

Last update: **2024/08/05 10:56**

