Тюнинг NGINX

Олег Черний
www.RIA.ua
Киев, Украина
28-29 марта 2009 CODECAMP

Компилируем, исключаем все лишнее

Компиляция nginx

- Модули, которые, возможно, вам не пригодятся: **mail, mail_ssl_module, http_perl_module, http_flv_module, http_dav_module**

- Модули, которые желательно включить при компиляции: **http_gzip_static_module, http_stub_status_module**

Вот как выглядит часть моего srpm, которая отвечает за конфигурацию nginx (%nginx_datadir, ... переменные spec-файла):

```
./configure \
--prefix=%nginx_datadir \
--conf-path=%nginx_etc/nginx.conf \
--sbin-path=%{_sbindir}/%{name} \
--error-log-path=%nginx_log/nginx.error.log \
--http-log-path=%nginx_log/nginx.log \
--http-client-body-temp-path=%nginx_spool/tmp/client \
--http-proxy-temp-path=%nginx_spool/tmp/proxy \
--http-fastcgi-temp-path=%nginx_spool/tmp/fastcgi \
--pid-path=%_var/run/nginx.pid \
--user=%nginx_user \
--group=%nginx_group \
--with-cc-opt="-I %_includedir/pcre/" \
--with-http_ssl_module \
--with-http_realip_module \
--with-http_addition_module \
--with-http_sub_module \
--with-http_dav_module \
--with-http_flv_module \
--with-http_gzip_static_module \
--with-http_stub_status_module \
--with-http_perl_module
```

Конфиг nginx - просто и понятно

Пример конфигурации

- Nginx писал админ для админов. Этот факт положительно отразился на синтаксисе конфигов, а также на простоте настройки.

```
user nginx;

# Число рабочих процессов, рекомендуется ставить по количеству ядер
worker_processes 8;

# Уменьшает число системных вызовов gettimeofday()
timer_resolution 100ms;

# Изменяет ограничение на число используемых файлов RLIMIT_NOFILE для рабочего процесса.
worker_rlimit_nofile 8192;

# Задаем приоритет рабочих процессов от -20 до 20 (отрицательное - более высокий приоритет).
worker_priority -5;

error_log /var/log/nginx/error.log;

pid /var/run/nginx.pid;

events {
  worker_connections 2048;
}
```

Основная секция:

```
http {
 include /etc/nginx/mime.types;
 default_type application/octet-stream;


 log_format main '$remote_addr - $remote_user [$time_local] $request '
 '"$status" $body_bytes_sent "$http_referer" '
 '"$http_user_agent" "$http_x_forwarded_for"';

 access_log /var/log/nginx/access.log main;

 # Включить sendfile(). Использование sendfile()
 # экономит системные вызовы, уменьшает число копирований данных,
 # позволяет использовать меньше физической памяти.
 sendfile on;
 keepalive_timeout 65;

 gzip on;
 gzip_min_length 1100;
 gzip_buffers 64 8k;
 gzip_comp_level 3;
 gzip_http_version 1.1;
 gzip_proxied any;
 gzip_types text/plain application/xml application/x-javascript text/css;

 # Load config files from the /etc/nginx/conf.vs directory
 include /etc/nginx/conf.vs/*.conf;
}
```

Пример простейшей конфигурации виртуального сервера:

```
server {
 listen 80;
 server_name _;

 location / {
 gzip_static on;
 root /var/nginx/html;
 index index.html index.htm;
 }

 error_page 404 /404.html;
 location = /404.html {
 root /var/nginx/html;
 }

 error_page 500 502 503 504 /50x.html;
 location = /50x.html {
 root /var/nginx/html;
 }
}
```

Синтаксис понятен в большинстве случаев даже без документации.

Условная классификация статического контента

Типы статического контента

"Легкий" контент

- html, css, js, xml, rss, txt.
- Хорошо поддается сжатию
- Требует мало места для хранения

Приминение nginx в любом случае даст заметный прирост производительности.

Типы статического контента

“Тяжелый" контент

- фото, видео, аудио-файлы
- Узким местом выступает, в первую очередь, дисковая система, размер оперативной памяти, пропускная способность канала.
- Задача раздачи такого типа контента делится на две: хранение контента и, собственно, раздача контента.

Все будет упираться в скорость работы дискового накопителя/RAID-массива

Проблемы быстродействия

Что делать, когда сайт начинает тормозить?

- Не паниковать
- Попробуйте увеличить количество worker_processes
- Поможет установка worker_priority в -5 и меньше (до -20)
- Временно отключить логи access_log off
- Анализ узких мест системы, например, с помощью **top, iostat, df -h, iptraf**
- Добавьте оперативной памяти или усовершенствуйте дисковую систему

Приемы оптимизации

"Легкий" контент

- Создаем виртуальный диск (tmpfs) и там размещаем “легкую” статику. В /etc/fstab добавляем
```
none /var/www/img_virtual tmpfs size=1g,mode=1777 0 0
```
- Сжимаем контент gzip-ом. Запускаем в нашей виртуальной папке:
```
for i in `find ./* -type f -name '*.js'`; do echo $i; gzip -c -9 $i > $i.gz; done;
for i in `find ./* -type f -name '*.css'`; do echo $i; gzip -c -9 $i > $i.gz; done;
```
в конфиг добавляем строчку “gzip_static on”, также можно включить online упаковку для динамических файлов “gzip on”
- Устанавливаем заголовки для проксирования контента директивой “expires 1y”
- Задаем кеширование дескрипторов файлов директивой “open_file_cache”

"Тяжелый" контент

- Настройка системы таким образом, чтоб свести к минимуму использование swap. Иногда приходится отключать sendfile (sendfile off)
- Если позволяет оперативная память, создайте виртуальный диск, на который поместите самые "запрашиваемые" файлы. Теперь мы можем применить директиву try_files (если файл не будет найден, на виртуальном диске будет обращение к backend):

```
location / {
    root /var/www/;
    try_files /img_virtual/hot/$uri  @storage;
}
location @storage {
    proxy_pass http://backend;
    proxy_set_header  Host        $host;
}
```

"Тяжелый" контент

- Если стороннюю программу по формированию кеша писать нет возможности, используйте директиву proxy_store:

```
location @storage {
    proxy_pass http://backend;
    proxy_set_header  Host       $host;

    proxy_store    on;
    proxy_store_access   user:rw  group:rw  all:r;
    proxy_temp_path      /var/www/img_virtual/hot/;
    root /var/www/img_virtual/hot/;
}
```

со временем кеш надо чистить, например так:

```
cd /var/www/img_virtual/hot/
find ./ -type f -amin +60 -delete
```

"Тяжелый" контент

- Если storage большой, занимает терабайты -- оперативой вопрос не решить, можно на фронтенд-е собрать RAID.

  Берем побольше винтов SAS, полноценный RAID-котроллер (fakeraid,hostraid это головная боль!). Монтируем туда swap, spool и кеш.

  Для нечасто меняющегося контента и нечастой перезаписи кеша можно применять SSD-винчестеры. Это работает быстро, у таких винчестеров нет такой характеристики как seek-to-seek, малый расход энергии (например для Intel X25-M 0,15Вт), хорошая скорость отдачи (до 250 MB/s).

"Тяжелый" контент

- Кеширование проксированных запросов.

  Модуль появился в nginx 0.7.44.

  "Сходить" на диск во многих случаях значительно дешевле, чем обращение в сеть. Главная задача такого кеширования - свести к необходимому минимуму сетевые опрерации и использовать "интелектуальное управление" дисковым кешем.

  Пример использования:

```
proxy_cache_path /data/nginx/cache levels=1:2 keys_zone=one:10m;

мена файлов в кэше будут такого вида:
/data/nginx/cache/c/29/b7f54b2df7773722d382f4809d6529c
```

Спасибо за внимание !

- Время для вопросов
- Полезные ссылки:
  http://sysoev.ru/
  http://habrahabr.ru/blogs/nginx/
  http://webo.in/articles/all/09-nginx-configuration/
  http://linux.ria.ua/