

01.Nginx应用指南

- 01.Nginx应用指南
 - [Nginx基本简述](#)
 - [Nginx优秀特性](#)
 - [Nginx快速安装](#)
 - [Nginx安装目录](#)
 - [Nginx编译参数](#)
 - [Nginx常用模块](#)
 - [Nginx内置变量](#)
 - [Nginx编译安装](#)

徐亮伟, 江湖人称标杆徐。多年互联网运维工作经验, 曾负责过大规模集群架构自动化运维管理工作。擅长Web集群架构与自动化运维, 曾负责国内某大型电商运维工作。

个人博客"[徐亮伟架构师之路](#)"累计受益数万人。

笔者Q:552408925、572891887

架构师群:471443208

Nginx基本简述

Nginx是一个开源且高性能、可靠的HTTP中间件、代理服务。

开源: 直接获取源代码

高性能: 支持海量并发

常见的HTTP服务

- 1.HTTPD -> Apache基金会
- 2.IIS -> 微软
- 3.GWS -> Google
- 4.openresty
- 5.tengline -> 淘宝基于Nginx开发

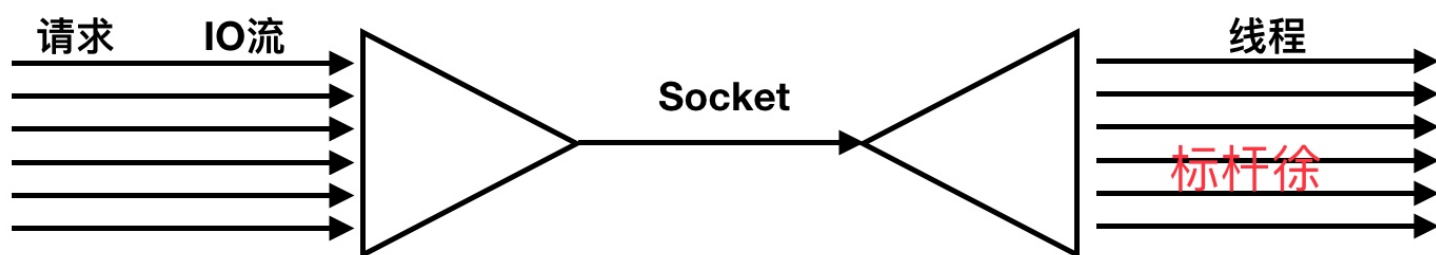
Nginx应用场景

静态处理
反向代理
负载均衡
资源缓存

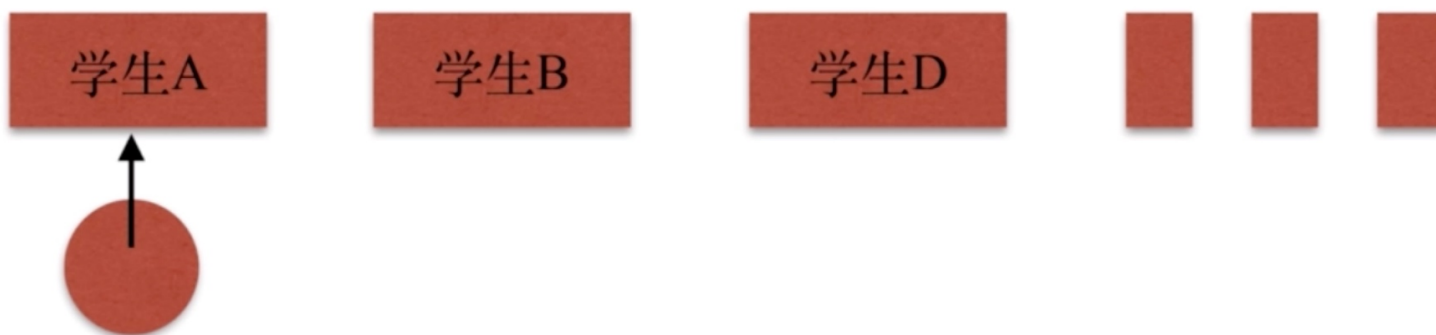
Nginx优秀特性

Nginx基于IO多路复用

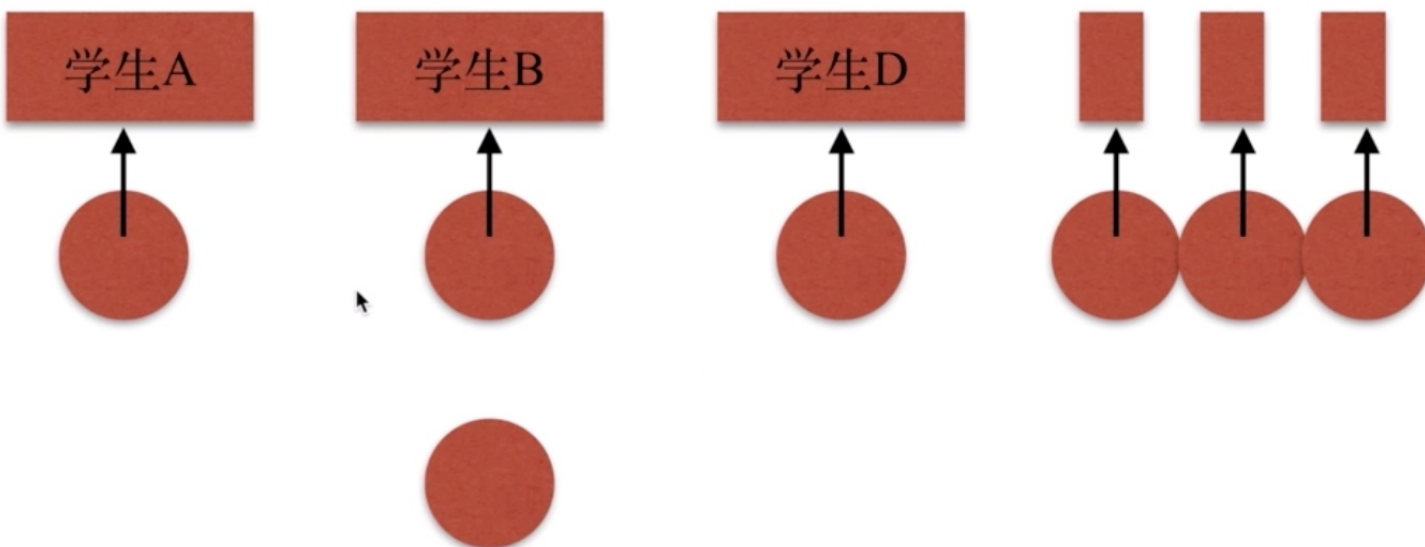
IO复用解决的是并发性的问题，Socket作为复用，



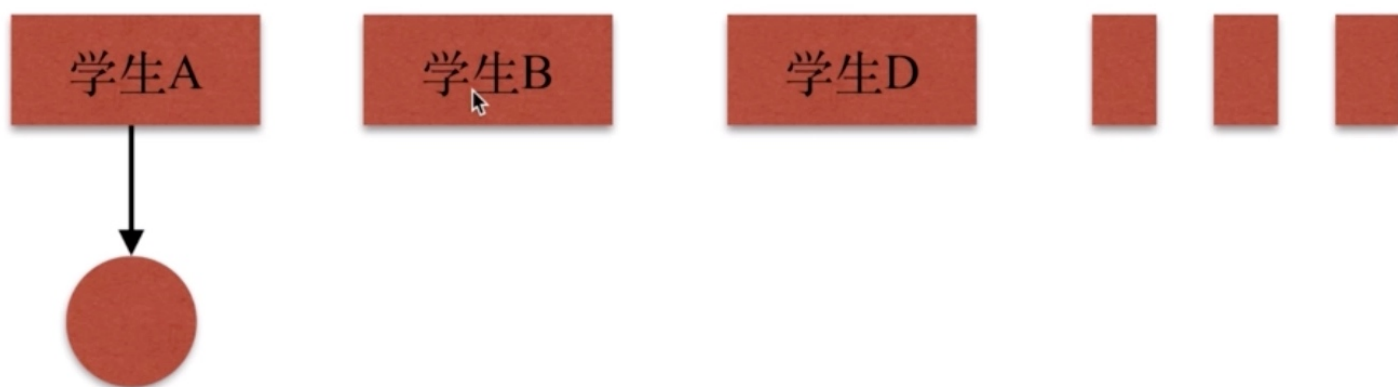
IO复用(串行,产生阻塞)



IO复用(多线程, 消耗大)



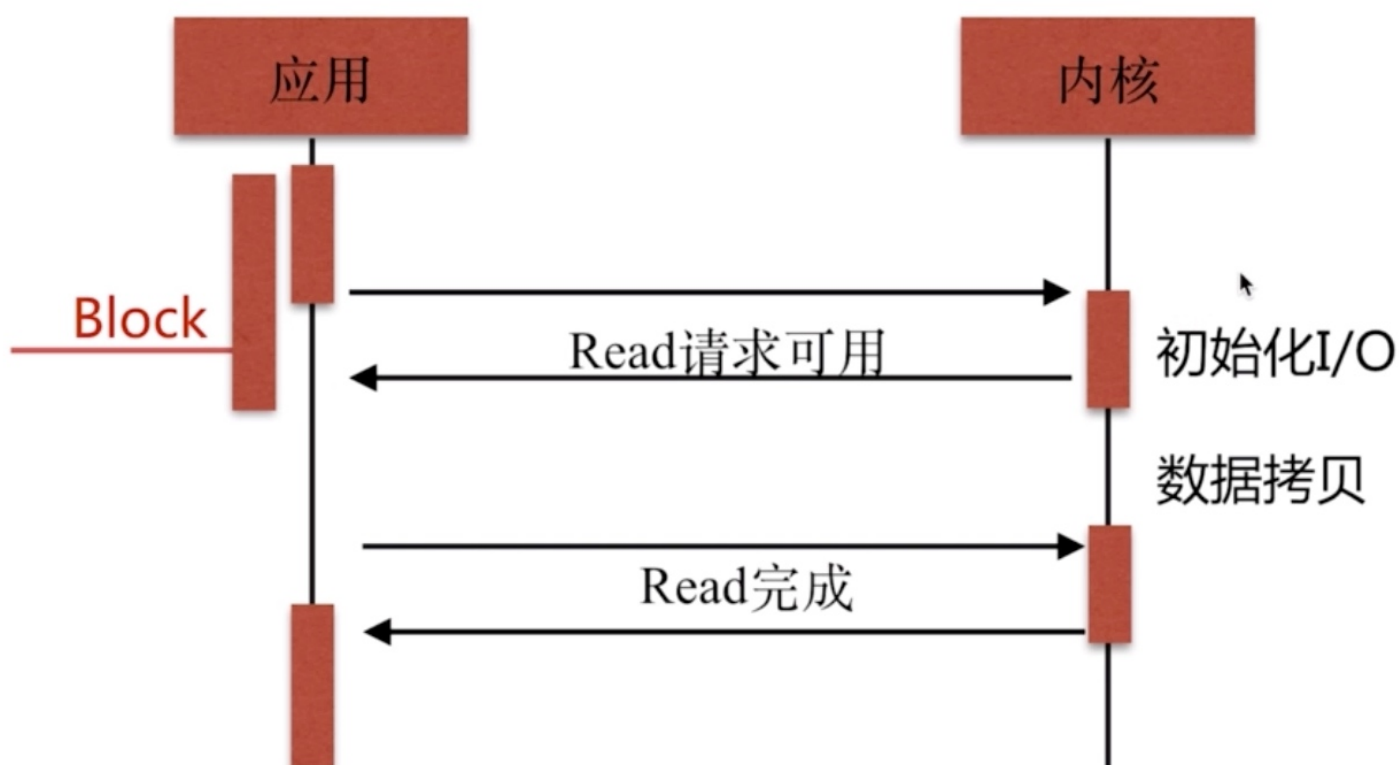
IO多路复用(主动上报)



多个描述符的I/O操作都能在一个线程内并发交替地顺序完成，这就叫I/O多路复用，这里的"复用"指的是复用同一个线程。

IO多路复用的实现方式有select、poll、Epoll

什么是select



select缺点

- 1.能够监视文件描述符的数量存在最大限制
- 2.线性遍历扫描效率低下

2.epoll模型

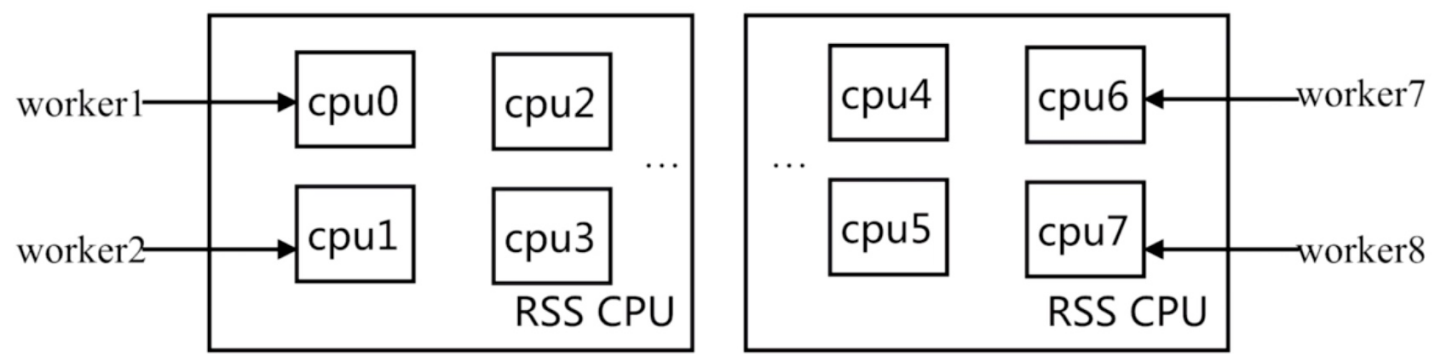
- 1.每当FD就绪，采用系统的回调函数之间将fd放入,效率更高。
- 2.最大连接无限制

轻量级

- 1.功能模块少
- 2.代码模块化

CPU亲和(affinity)

将CPU核心和Nginx工作进程绑定方式，把每个worker进程固定在一个cpu上执行，减少切换cpu的 `cache miss` ，获得更好的性能。



sendfile

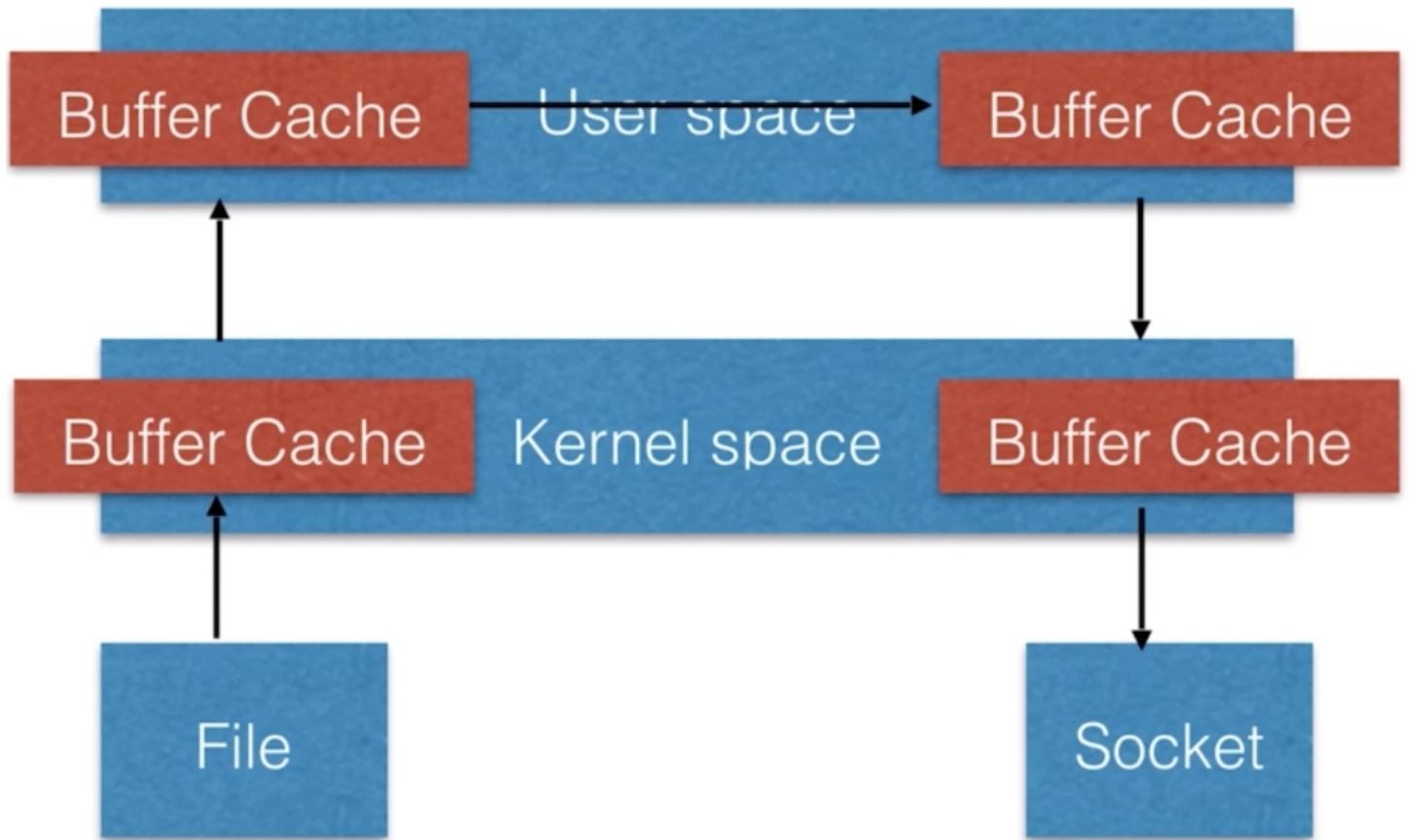
传统文件传输, 在实现上其实是比较复杂的, 其具体流程细节如下:

- 1.调用read函数，文件数据被复制到内核缓冲
- 2.read函数返回，文件数据从内核缓冲区复制到用户缓冲区
- 3.write函数调用，将文件数据从用户缓冲区复制到内核与socket相关的缓冲区。
- 4.数据从socket缓冲区复制到相关协议引擎。

传统文件传输数据实际上是经过了四次复制操作:

硬盘—>内核buf—>用户buf—>socket缓冲区(内核)—>协议引擎

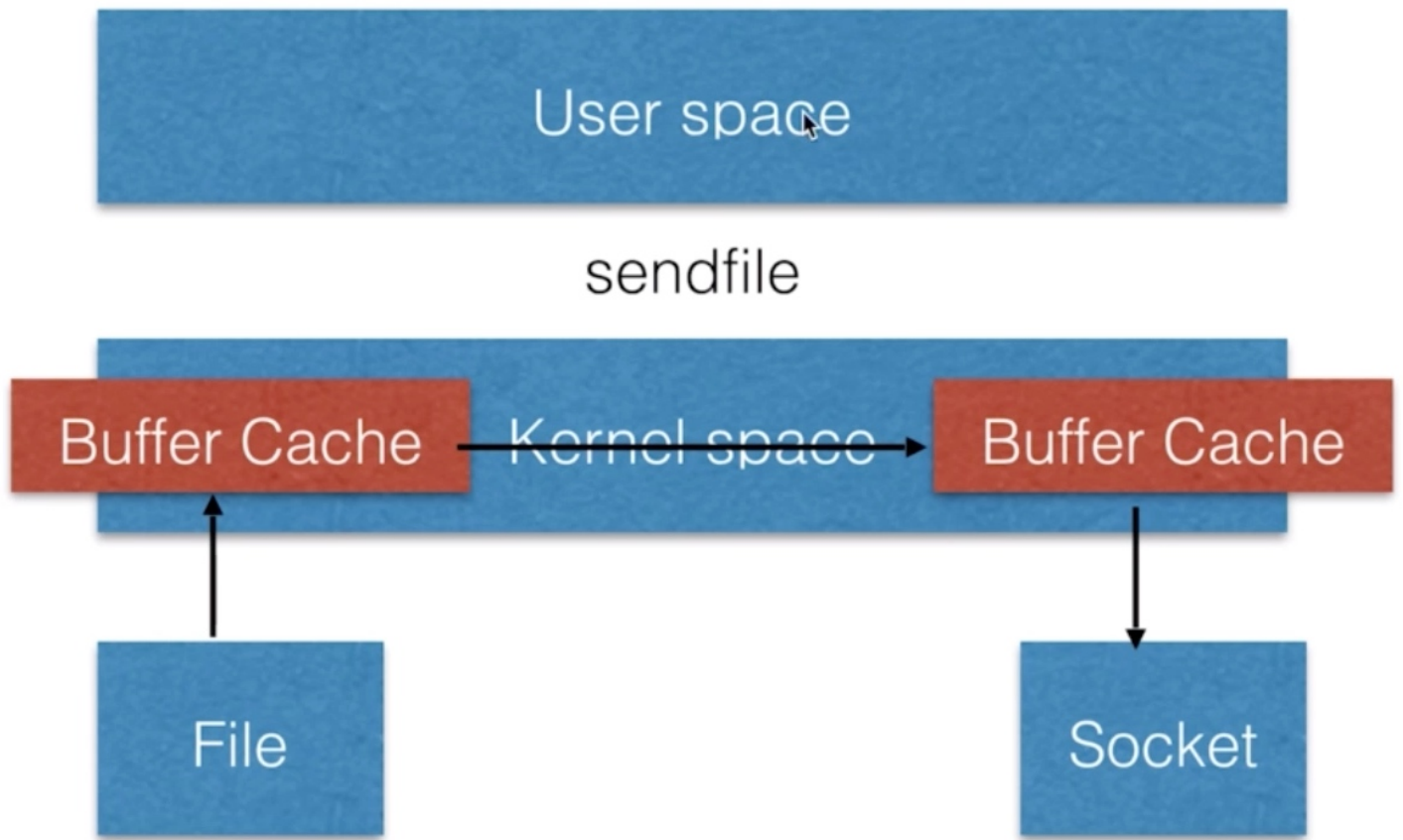
也就是说传统的文件传输需要经过多次上下文的切换才能完成拷贝或读取, 效率不高。



sendfile文件传输是在内核中操作完成的, 函数直接在两个文件描述符之间传递数据, 从而避免了内核缓冲区数据和用户缓冲区数据之间的拷贝, 操作效率很高, 被称之为零拷贝。

- 1.系统调用sendfile函数通过 DMA 把硬盘数据拷贝到 kernel buffer,
- 2.数据被 kernel 直接拷贝到另外一个与 socket 相关的 kernel buffer。
- 3.DMA把数据从kernel buffer直接拷贝给协议栈。

这里没有用户空间和内核空间之间的切换, 在内核中直接完成了从一个buffer到另一个buffer的拷贝。



Nginx快速安装

Mainline version 开发版

Stable version 稳定版

Legacy version 历史版本

基础环境准备:

```
// 确认系统网络
[root@Nginx ~]# ping baidu.com

// 关闭firewalld
[root@Nginx ~]# systemctl stop firewalld
[root@Nginx ~]# systemctl disable firewalld

// 临时关闭selinux
[root@Nginx ~]# setenforce 0

// 初始化基本目录
[root@Nginx ~]# mkdir /soft/{code,logs,package,backup} -p

// 基本安装包
[root@Nginx ~]# yum install -y gcc gcc-c++ autoconf \
```

```
pcrc pcre-devel make automake wget httpd-tools vim tree
```

```
//配置Nginx官方Yum源
[root@Nginx ~]# vim /etc/yum.repos.d/nginx.repo
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/7/$basearch/
gpgcheck=0
enabled=1
```

```
//安装Nginx
[root@Nginx ~]# yum install nginx -y
```

```
//查看Nginx当前版本
[root@Nginx ~]# nginx -v
nginx version: nginx/1.12.2
```

Nginx安装目录

为了让大家更清晰的了解 Nginx 软件的全貌，有必要介绍下 Nginx 安装后整体的目录结构及文件功能。

```
[root@Nginx ~]# rpm -ql nginx
```

如下表格对 Nginx 安装目录做详细概述

路径	类型	作用
/etc/nginx /etc/nginx/nginx.conf /etc/nginx/conf.d /etc/nginx/conf.d/default.conf	配置文件	Nginx主配置文件
/etc/nginx/fastcgi_params /etc/nginx/scgi_params /etc/nginx/uwsgi_params	配置文件	Cgi、Fastcgi、Uwsgi配置文件
/etc/nginx/win-utf /etc/nginx/koi-utf /etc/nginx/koi-win	配置文件	Nginx编码转换映射文件
/etc/nginx/mime.types	配置文件	http协议的Content-Type与扩展名

/usr/lib/systemd/system/nginx.service	配置文件	配置系统守护进程管理器
/etc/logrotate.d/nginx	配置文件	Nginx日志轮询,日志切割
/usr/sbin/nginx /usr/sbin/nginx-debug	命令	Nginx终端管理命令
/etc/nginx/modules /usr/lib64/nginx /usr/lib64/nginx/modules	目录	Nginx模块目录
/usr/share/nginx /usr/share/nginx/html /usr/share/nginx/html/50x.html /usr/share/nginx/html/index.html	目录	Nginx默认站点目录
/usr/share/doc/nginx-1.12.2 /usr/share/man/man8/nginx.8.gz	目录	Nginx的帮助手册
/var/cache/nginx	目录	Nginx的缓存目录
/var/log/nginx	目录	Nginx的日志目录

Nginx编译参数

查看Nginx编译参数

```
[root@Nginx ~]# nginx -V
```

下表展示了Nginx编译参数选项以及作用

编译选项	作用
--prefix=/etc/nginx --sbin-path=/usr/sbin/nginx --modules-path=/usr/lib64/nginx/modules --conf-path=/etc/nginx/nginx.conf --error-log-path=/var/log/nginx/error.log --http-log-path=/var/log/nginx/access.log --pid-path=/var/run/nginx.pid --lock-path=/var/run/nginx.lock	程序安装目录和路径

<code>--http-client-body-temp-path=/var/cache/nginx/client_temp</code> <code>--http-proxy-temp-path=/var/cache/nginx/proxy_temp</code> <code>--http-fastcgi-temp-path=/var/cache/nginx/fastcgi_temp</code> <code>--http-uwsgi-temp-path=/var/cache/nginx/uwsgi_temp</code> <code>--http-scgi-temp-path=/var/cache/nginx/scgi_temp</code>	临时缓存文件
<code>--user=nginx</code> <code>--group=nginx</code>	设定Nginx进程启动用户和组(安全)
<code>--with-cc-opt</code>	设置额外的参数将被添加到CFLAGS变量
<code>--with-ld-opt</code>	设置附加的参数, 链接系统库

Nginx常用模块

Nginx模块分为 Nginx官方模块以及Nginx第三方模块

Nginx编译选项	模块作用
<code>ngx_http_core_module</code>	包含一些核心的http参数配置，对应Nginx的配置区块部分
<code>ngx_http_access_module</code>	访问控制模块，用来控制网站用户对Nginx的访问
<code>ngx_http_gzip_module</code>	压缩模块，对Nginx返回的数据压缩，属于性能优化模块
<code>ngx_http_fastcgi_module</code>	fastci模块，和动态应用相关的模块，例如PHP
<code>ngx_http_proxy_module</code>	proxy代理模块
<code>ngx_http_upstream_module</code>	负载均衡模块，可以实现网站的负载均衡功能及节点的健康检查。
<code>ngx_http_rewrite_module</code>	URL地址重写模块
<code>ngx_http_limit_conn_module</code>	限制用户并发连接数及请求数模块
<code>ngx_http_limit_req_module</code>	限制Nginx request processing rate根据定义的key

ngx_http_log_module	访问日志模块，以指定的格式记录Nginx客户访问日志等信息
ngx_http_auth_basic_module	Web认证模块，设置Web用户通过账号密码访问Nginx
nginx_http_ssl_module	ssl模块，用于加密的http连接，如https

Nginx内置变量

http核心模块的内置变量

- http请求变量
- Nginx内置变量
- 自定义变量

```
$uri: 当前请求的uri, 不带参数
$request_uri: 请求的uri, 带完整参数
$host: http请求报文中host首部, 如果没有则以处理此请求的虚拟主机的主机名代替
$hostname: nginx服务运行在主机的主机名
$remote_addr: 客户端IP
$remote_port: 客户端端口
$remote_user: 使用用户认证时客户端用户输入的用户名
$request_filename: 用户请求中的URI经过本地root或alias转换后映射的本地文件路径
$request_method: 请求方法, GET POST PUT
$server_addr: 服务器地址
$server_name: 服务器名称
$server_port: 服务器端口
$server_protocol: 服务器向客户端发送响应时的协议, 如http/1.1 http/1.0
$scheme: 在请求中使用scheme, 如http://xxx.com中的http
$http_HEADER: 匹配请求报文中指定的HEADER
$http_host: 匹配请求报文中的host首部
$document_root: 当前请求映射到的root配置
```

Nginx编译安装

openresty编译安装

```
//1. 基础环境准备
mkdir /soft/package/src -p
mkdir /soft/package/src/nginx_3rd -p
useradd -s /sbin/nologin -M www
yum -y install wget make gcc gcc-c++ zlib-devel openssl \
```

```
openssl-devel pcre-devel kernel keyutils patch perl zlib \  
tcl readline-devel glibc libxslt-devel gd-devel GeoIP-devel \  
libevent libevent-devel
```

//2. 安装依赖插件

```
cd /soft/package/src
```

#openresty安装包

```
wget https://openresty.org/download/openresty-1.11.2.2.tar.gz  
tar -xf openresty-1.11.2.2.tar.gz
```

#libdrizzle模块

```
wget http://openresty.org/download/drizzle7-2011.07.21.tar.gz  
tar xzvf drizzle7-2011.07.21.tar.gz  
cd drizzle7-2011.07.21/  
./configure --without-server  
make libdrizzle-1.0  
make install-libdrizzle-1.0
```

#openssl依赖包安装

```
wget https://www.openssl.org/source/openssl-1.0.2d.tar.gz  
tar -xf openssl-1.0.2d.tar.gz
```

#pcre依赖

```
wget -O pcre-8.37.tar.gz \  
http://sourceforge.net/projects/pcre/files/pcre/8.37/pcre-8.37.tar.gz/download  
tar -xf pcre-8.37.tar.gz
```

```
wget ftp://91.193.69.2/distributive/FreeBSD/ports/local-distfiles/osa/nginx-accessk  
ey-2.0.3.tar.gz  
tar xf nginx-accesskey-2.0.3.tar.gz  
mv nginx-accesskey-2.0.3 ngx_3rd/nginx-accesskey-master
```

```
cd /soft/package/src/nginx_3rd  
wget -O ngx_http_consistent_hash.zip \  
https://codeload.github.com/replay/nginx_http_consistent_hash/zip/master  
unzip ngx_http_consistent_hash.zip
```

```
wget -O nginx-static-etags.zip \  
https://codeload.github.com/mikewest/nginx-static-etags/zip/master  
unzip nginx-static-etags.zip
```

```
wget -O nginx-http-footer-filter.zip \  
https://codeload.github.com/alibaba/nginx-http-footer-filter/zip/master  
unzip nginx-http-footer-filter.zip
```

```
wget -O form-input-nginx-module.zip \  
https://codeload.github.com/calio/form-input-nginx-module/zip/master
```

```
unzip form-input-nginx-module.zip
```

```
wget -O ngx_http_accounting_module.zip \  
https://codeload.github.com/Lax/ngx_http_accounting_module/zip/master  
unzip ngx_http_accounting_module.zip
```

```
wget -O ngx_log_if.zip \  
https://codeload.github.com/cfsego/ngx_log_if/zip/master  
unzip ngx_log_if.zip
```

```
wget -O nginx-limit-upstream.zip \  
https://codeload.github.com/cfsego/nginx-limit-upstream/zip/master  
unzip nginx-limit-upstream.zip
```

```
wget -O limit_upload_rate.zip \  
https://codeload.github.com/cfsego/limit_upload_rate/zip/master  
unzip limit_upload_rate.zip
```

```
wget -O ngx_devel_kit.zip \  
https://codeload.github.com/simpl/ngx_devel_kit/zip/master  
unzip ngx_devel_kit.zip
```

```
wget -O nginx-module-vts.zip \  
https://codeload.github.com/vozlt/nginx-module-vts/zip/master  
unzip nginx-module-vts.zip
```

```
wget -O nginx_tcp_proxy_module.zip \  
https://codeload.github.com/yaoweibin/nginx_tcp_proxy_module/zip/master  
unzip nginx_tcp_proxy_module.zip
```

```
wget -O nginx-module-sts.zip \  
https://codeload.github.com/vozlt/nginx-module-sts/zip/master  
unzip nginx-module-sts.zip
```

```
wget -O nginx-http-concat.zip \  
https://codeload.github.com/alibaba/nginx-http-concat/zip/master  
unzip nginx-http-concat.zip
```

```
wget -O nginx-url.zip \  
https://codeload.github.com/vozlt/nginx-module-url/zip/master  
unzip nginx-url.zip
```

```
wget -O nginx-access-plus.zip \  
https://codeload.github.com/nginx-clojure/nginx-access-plus/zip/master  
unzip nginx-access-plus.zip
```

//3. 编译openresty

```
cd /soft/package/src/nginx_openresty-1.11.2.2
./configure \
--user=www \
--group=www \
--prefix=/soft/openresty-1.11 \
--pid-path=/soft/openresty-1.11/nginx/pid \
--error-log-path=/soft/log/nginx/error.log \
--http-log-path=/soft/log/nginx/access.log \
--http-proxy-temp-path=/soft/openresty-1.11/nginx/proxy_temp \
--http-fastcgi-temp-path=/soft/openresty-1.11/nginx/fastcgi_temp \
--http-client-body-temp-path=/soft/openresty-1.11/nginx/client_body_temp \
--with-openssl=/soft/package/src/openssl-1.0.2d \
--with-pcre=/soft/package/src/pcre-8.37 \
--with-libdrizzle=/usr/local \
--with-threads \
--with-file-aio \
--with-http_sub_module \
--with-http_dav_module \
--with-http_flv_module \
--with-http_ssl_module \
--with-http_xslt_module \
--with-http_iconv_module \
--with-http_geoip_module \
--with-http_realip_module \
--with-http_gunzip_module \
--with-http_drizzle_module \
--with-http_addition_module \
--with-http_gzip_static_module \
--with-http_secure_link_module \
--with-http_stub_status_module \
--with-http_auth_request_module \
--with-http_random_index_module \
--with-http_image_filter_module \
--without-http_uwsgi_module \
--without-http_scgi_module \
--with-mail \
--with-mail_ssl_module \
--add-module=/soft/package/src/nginx_3rd/nginx-accesskey-master/ \
--add-module=/soft/package/src/nginx_3rd/nginx-http-consistent-hash-master/ \
--add-module=/soft/package/src/nginx_3rd/nginx-static-etags-master/ \
--add-module=/soft/package/src/nginx_3rd/nginx-http-footer-filter-master/ \
--add-module=/soft/package/src/nginx_3rd/nginx-http-accounting-module-master/ \
--add-module=/soft/package/src/nginx_3rd/nginx-log-if-master/ \
--add-module=/soft/package/src/nginx_3rd/nginx-http-concat-master/ \
--add-module=/soft/package/src/nginx_3rd/nginx-module-vts-master/ \
--add-module=/soft/package/src/nginx_3rd/nginx-module-url-master/ \
--add-module=/soft/package/src/nginx_3rd/nginx-access-plus-master/src/c/
gmake && gmake install && ln -s /soft/openresty-1.11 /soft/openresty
```

