[**第一章 安装OpenResty(Nginx+Lua)开发环境**](http://jinnianshilongnian.iteye.com/blog/2186270)

**博客分类：**

* [跟我学Nginx+Lua开发](http://jinnianshilongnian.iteye.com/category/333854)

[nginx](http://www.iteye.com/blogs/tag/nginx)[lua](http://www.iteye.com/blogs/tag/lua)[ngx\_lua](http://www.iteye.com/blogs/tag/ngx_lua)[openresty](http://www.iteye.com/blogs/tag/openresty)

首先我们选择使用[OpenResty](http://openresty.org/)，其是由Nginx核心加很多第三方模块组成，其最大的亮点是默认集成了Lua开发环境，使得Nginx可以作为一个Web Server使用。借助于Nginx的事件驱动模型和非阻塞IO，可以实现高性能的Web应用程序。而且OpenResty提供了大量组件如Mysql、Redis、Memcached等等，使在Nginx上开发Web应用更方便更简单。目前在京东如实时价格、秒杀、动态服务、单品页、列表页等都在使用Nginx+Lua架构，其他公司如淘宝、去哪儿网等。

**安装环境**

安装步骤可以参考<http://openresty.org/#Installation>。

1、创建目录/usr/servers，以后我们把所有软件安装在此目录

**Java代码  [收藏代码](javascript:void())**

1. mkdir -p /usr/servers
2. cd /usr/servers/

2、安装依赖（我的环境是ubuntu，可以使用如下命令安装，其他的可以参考openresty安装步骤）

**Java代码  [收藏代码](javascript:void())**

1. apt-get install libreadline-dev libncurses5-dev libpcre3-dev libssl-dev perl

3、下载ngx\_openresty-1.7.7.2.tar.gz并解压

**Java代码  [收藏代码](javascript:void())**

1. wget http://openresty.org/download/ngx\_openresty-1.7.7.2.tar.gz
2. tar -xzvf ngx\_openresty-1.7.7.2.tar.gz

ngx\_openresty-1.7.7.2/bundle目录里存放着nginx核心和很多第三方模块，比如有我们需要的Lua和LuaJIT。

3、安装LuaJIT

**Java代码  [收藏代码](javascript:void())**

1. cd bundle/LuaJIT-2.1-20150120/
2. make clean && make && make install
3. ln -sf luajit-2.1.0-alpha /usr/local/bin/luajit

4、下载ngx\_cache\_purge模块，该模块用于清理nginx缓存

**Java代码  [收藏代码](javascript:void())**

1. cd /usr/servers/ngx\_openresty-1.7.7.2/bundle
2. wget https://github.com/FRiCKLE/ngx\_cache\_purge/archive/2.3.tar.gz
3. tar -xvf 2.3.tar.gz

5、下载nginx\_upstream\_check\_module模块，该模块用于ustream健康检查

**Java代码  [收藏代码](javascript:void())**

1. cd /usr/servers/ngx\_openresty-1.7.7.2/bundle
2. wget https://github.com/yaoweibin/nginx\_upstream\_check\_module/archive/v0.3.0.tar.gz
3. tar -xvf v0.3.0.tar.gz

6、安装ngx\_openresty

**Java代码  [收藏代码](javascript:void())**

1. cd /usr/servers/ngx\_openresty-1.7.7.2
2. ./configure --prefix=/usr/servers --with-http\_realip\_module  --with-pcre  --with-luajit --add-module=./bundle/ngx\_cache\_purge-2.3/ --add-module=./bundle/nginx\_upstream\_check\_module-0.3.0/ -j2
3. make && make install

--with\*\*\*                安装一些内置/集成的模块

--with-http\_realip\_module  取用户真实ip模块

-with-pcre               Perl兼容的达式模块

--with-luajit              集成luajit模块

--add-module            添加自定义的第三方模块，如此次的ngx\_che\_purge

8、到/usr/servers目录下

**Java代码  [收藏代码](javascript:void())**

1. cd /usr/servers/
2. ll

会发现多出来了如下目录，说明安装成功

**/usr/servers/luajit**：luajit环境，luajit类似于java的jit，即即时编译，lua是一种解释语言，通过luajit可以即时编译lua代码到机器代码，得到很好的性能；

**/usr/servers/lualib**：要使用的lua库，里边提供了一些默认的lua库，如redis，json库等，也可以把一些自己开发的或第三方的放在这；

/usr/servers/nginx ：安装的nginx；

通过/usr/servers/nginx/sbin/nginx  -V 查看nginx版本和安装的模块

7、启动nginx

/usr/servers/nginx/sbin/nginx

接下来该配置nginx+lua开发环境了

**配置环境**

配置及Nginx HttpLuaModule文档在可以查看<http://wiki.nginx.org/HttpLuaModule>。

1、编辑nginx.conf配置文件

**Java代码  [收藏代码](javascript:void())**

1. vim /usr/servers/nginx/conf/nginx.conf

2、在http部分添加如下配置

**Java代码  [收藏代码](javascript:void())**

1. #lua模块路径，多个之间”;”分隔，其中”;;”表示默认搜索路径，默认到/usr/servers/nginx下找
2. lua\_package\_path "/usr/servers/lualib/?.lua;;";  #lua 模块
3. lua\_package\_cpath "/usr/servers/lualib/?.so;;";  #c模块

3、为了方便开发我们在/usr/servers/nginx/conf目录下创建一个lua.conf

**Java代码  [收藏代码](javascript:void())**

1. #lua.conf
2. server {
3. listen       80;
4. server\_name  \_;
5. }

4、在nginx.conf中的http部分添加include lua.conf包含此文件片段

**Java代码  [收藏代码](javascript:void())**

1. include lua.conf;

5、测试是否正常

**Java代码  [收藏代码](javascript:void())**

1. /usr/servers/nginx/sbin/nginx  -t

如果显示如下内容说明配置成功

nginx: the configuration file /usr/servers/nginx/conf/nginx.conf syntax is ok

nginx: configuration file /usr/servers/nginx/conf/nginx.conf test is successful

**HelloWorld**

1、在lua.conf中server部分添加如下配置

**Java代码  [收藏代码](javascript:void())**

1. location /lua {
2. default\_type 'text/html';
3. content\_by\_lua 'ngx.say("hello world")';
4. }

2、测试配置是否正确

**Java代码  [收藏代码](javascript:void())**

1. /usr/servers/nginx/sbin/nginx  -t

3、重启nginx

**Java代码  [收藏代码](javascript:void())**

1. /usr/servers/nginx/sbin/nginx  -s reload

 4、访问如<http://192.168.1.6/lua>（自己的机器根据实际情况换ip），可以看到如下内容

hello world

5、lua代码文件

我们把lua代码放在nginx配置中会随着lua的代码的增加导致配置文件太长不好维护，因此我们应该把lua代码移到外部文件中存储。

**Java代码  [收藏代码](javascript:void())**

1. vim /usr/servers/nginx/conf/lua/test.lua

**Java代码  [收藏代码](javascript:void())**

1. #添加如下内容
2. ngx.say("hello world");

然后lua.conf修改为

**Java代码  [收藏代码](javascript:void())**

1. location /lua {
2. default\_type 'text/html';
3. content\_by\_lua\_file conf/lua/test.lua; #相对于nginx安装目录
4. }

此处conf/lua/test.lua也可以使用绝对路径/usr/servers/nginx/conf/lua/test.lua。

6、lua\_code\_cache

默认情况下lua\_code\_cache  是开启的，即缓存lua代码，即每次lua代码变更必须reload nginx才生效，如果在开发阶段可以通过lua\_code\_cache  off;关闭缓存，这样调试时每次修改lua代码不需要reload nginx；但是正式环境一定记得开启缓存。

**Java代码  [收藏代码](javascript:void())**

1. location /lua {
2. default\_type 'text/html';
3. lua\_code\_cache off;
4. content\_by\_lua\_file conf/lua/test.lua;
5. }

开启后reload nginx会看到如下报警

nginx: [alert] lua\_code\_cache is off; this will hurt performance in /usr/servers/nginx/conf/lua.conf:8

7、错误日志

如果运行过程中出现错误，请不要忘记查看错误日志。

**Java代码  [收藏代码](javascript:void())**

1. tail -f /usr/servers/nginx/logs/error.log

到此我们的基本环境搭建完毕。

**nginx+lua项目构建**

以后我们的nginx lua开发文件会越来越多，我们应该把其项目化，已方便开发。项目目录结构如下所示：

example

    example.conf     ---该项目的nginx 配置文件

    lua              ---我们自己的lua代码

      test.lua

    lualib            ---lua依赖库/第三方依赖

      \*.lua

      \*.so

其中我们把lualib也放到项目中的好处就是以后部署的时候可以一起部署，防止有的服务器忘记复制依赖而造成缺少依赖的情况。

我们将项目放到到/usr/example目录下。

/usr/servers/nginx/conf/nginx.conf配置文件如下(此处我们最小化了配置文件)

**Java代码  [收藏代码](javascript:void())**

1. #user  nobody;
2. worker\_processes  2;
3. error\_log  logs/error.log;
4. events {
5. worker\_connections  1024;
6. }
7. http {
8. include       mime.types;
9. default\_type  text/html;
11. #lua模块路径，其中”;;”表示默认搜索路径，默认到/usr/servers/nginx下找
12. lua\_package\_path "/usr/example/lualib/?.lua;;";  #lua 模块
13. lua\_package\_cpath "/usr/example/lualib/?.so;;";  #c模块
14. include /usr/example/example.conf;
15. }

通过绝对路径包含我们的lua依赖库和nginx项目配置文件。

/usr/example/example.conf配置文件如下

**Java代码  [收藏代码](javascript:void())**

1. server {
2. listen       80;
3. server\_name  \_;
5. location /lua {
6. default\_type 'text/html';
7. lua\_code\_cache off;
8. content\_by\_lua\_file /usr/example/lua/test.lua;
9. }
10. }

lua文件我们使用绝对路径/usr/example/lua/test.lua。

到此我们就可以把example扔svn上了。