

# 【入门篇】Nginx + FastCGI 程序 (C/C++) 搭建高性能web service的Demo及部署发布

 [blog.csdn.net/allenlinrui/article/details/19419721](http://blog.csdn.net/allenlinrui/article/details/19419721)

≡ 分类：

由于最近工作的需要，本人学习了一下利用高性能web server - Nginx，来发布C/C++编写的fastCGI程序，详细细节如下。

## 1.介绍

Nginx - 高性能web server，这个不用多说了，大家都知道。

FastCGI程序 - 常驻型CGI程序，它是语言无关的、可伸缩架构的CGI开放扩展，其主要行为是将CGI解释器进程保持在内存中并因此获得较高的性能。

Nginx要调用FastCGI程序，需要用到FastCGI进程管理程序（因为nginx不能直接执行外部的cgi程序，我们可使用lighttpd中的spawn-fastcgi来让nginx可支持外部cgi运行。也有其他方法安装nginx-fcgi来让nginx支持cgi，这里是使用spawn-fastcgi的方法），来达到调用FastCGI程序的目的。Nginx本身没有集成类似的模块，而Apache具备该功能模块，所以不需要额外安装FastCGI进程管理程序。

## 2.工作原理

Nginx不支持对外部程序的直接调用或者解析，所有的外部程序（包括PHP）必须通过FastCGI接口来调用。FastCGI接口在Linux下是socket（这个socket可以是文件socket，也可以是ip socket）。为了调用CGI程序，还需要一个FastCGI的wrapper（wrapper可以理解为用于启动另一个程序的程序），这个wrapper绑定在某个固定socket上，如端口或者文件socket。

当Nginx将CGI请求发送给这个socket的时候，通过FastCGI接口，wrapper接收到请求，然后派生出一个新的线程，这个线程调用解释器或者外部程序处理脚本并读取返回数据；接着，wrapper再将返回的数据通过FastCGI接口，沿着固定的socket传递给Nginx；最后，Nginx将返回的数据发送给客户端。这就是Nginx+FastCGI的整个运作过程，如图1所示。

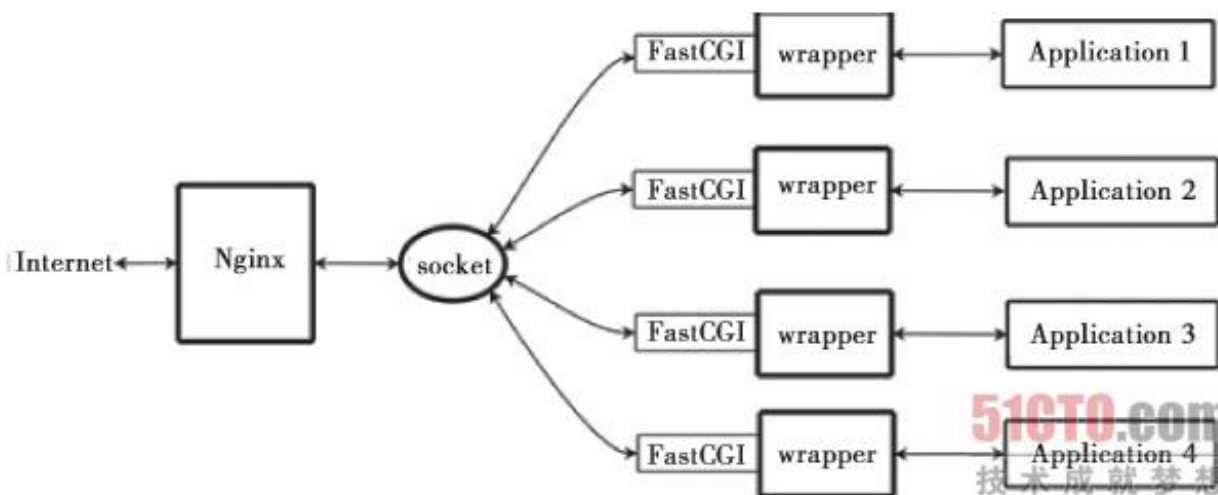


图1 Nginx+FastCGI运行过程

FastCGI接口方式在脚本解析服务器（CGI应用程序服务器）上启动一个或者多个守护进程对动态脚本进行解析，这些进程就是FastCGI进程管理器，或者称为FastCGI引擎。spawn-fcgi与PHP-FPM都是FastCGI进程管理器（支持PHP和C/C++）。

介绍到这里，大家应该都对该模式有了一定的了解，下面开始进行实战！

### 3.环境部署

#### 3.1.Nginx的安装、部署与配置

nginx下载目录 <http://nginx.org/en/download.html>

这我们使用的是nginx-1.5.10

##### [安装]

下载以后解压并安装（请记得看README）

**./configure**（注意了类似checking for \*\*\* ... not found项，可能是依赖包没有，则需要安装依赖包）

缺少**pcre**，则需要额外安装 <http://www.pcre.org/>（或者采用apt-get或yum的安装方式）

缺少**zlib**，则需要额外安装 <http://www.zlib.net/>（或者采用apt-get或yum的安装方式）

缺少**OpenSSL**，则需要额外安装 <http://www.openssl.org>（或者采用apt-get或yum的安装方式）

如果需要配置安装额外的功能模块，可以参考这里 <http://wiki.codemongers.com/NginxChsInstall>

**make**

**make install**（默认安装到/usr/local/nginx）

##### [配置和管理]

###### 1) 执行选项

-c </path/to/config> 为 Nginx 指定一个配置文件，来代替缺省的。不输入则使用默认的配置文

-t 不运行，而仅仅测试配置文件。nginx 将检查配置文件的语法的正确性，并尝试打开配置文件中

-v 显示 nginx 的版本。

-V 显示 nginx 的版本，编译器版本和配置参数。

## 2) 检查配置文件

**sudo ./nginx -t**

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

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

## 3) 启动 - 默认和特殊

**/usr/local/nginx/sbin/nginx** (默认启动方式)

**/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf** (指定配置文件启动)

## 4) 查看nginx进程号 (master是主进程)

**ps -ef | grep nginx**

## 5) 重新加载配置文件

**sudo kill -HUP [nginx主进程号]**

通过系统的信号控制 Nginx

可以使用信号系统来控制主进程。默认，nginx 将其主进程的 pid 写入到 /usr/local/nginx/logs/nginx.pid 文件中。通过传递参数 给 ./configure 或使用 pid 指令，来改变该文件的位置。

主进程可以处理以下的信号：

命令 说明 备注

TERM, INT 快速关闭

QUIT 从容关闭

HUP 重载配置 用新的配置开始新的工作进程 从容关闭旧的工作进程

USR1 重新打开日志文件

USR2 平滑升级可执行程序

WINCH 从容关闭工作进程

## 6) 默认目录结构

主目录：**/usr/local/nginx/**

配置目录：**/usr/local/nginx/conf/**

root目录：**/usr/local/nginx/html/**

可执行文件路径：**/usr/local/nginx/sbin/**

## 3.2.spawn\_fastcgi的安装、部署与配置

spawn\_fastcgi <https://github.com/lighttpd/spawn-fcgi>

这里使用的是1.6.3的版本 <https://github.com/lighttpd/spawn-fcgi/releases/tag/v1.6.3>

下载以后解压并安装（请记得看README）

如果没有**configure**，请先执行**./autogen.sh**，生成configure

```
./configure
```

```
make
```

编译好以后，将可执行文件移动到Nginx的sbin目录下

```
cp ./src/spawn-fcgi /usr/local/nginx/sbin/ （cp到nginx的安装目录下）
```

### 3.3.fastcgi库的安装（库绝对不是必须的，觉得技术好的大牛可以自己写）

库地址 <http://www.fastcgi.com/dist/fcgi.tar.gz>

下载以后，解压并安装（默认安装）

```
./configure
```

```
make
```

```
make install
```

## 4.Demo和web发布

### 4.1.Demo程序

**[CGI程序]**

**[cpp]** [view plain copy](#)

```
1. #include <fcgi_stdio.h>
2. #include <stdlib.h>
3. int main() {
4.     int count = 0;
5.     while (FCGI_Accept() >= 0) {
6.         printf("Content-type: text/html\r\n"
7.             "\r\n"
8.             ""
9.             "FastCGI Hello!"
10.            "Request number %d running on host%s "
11.            "Process ID: %d\n", ++count, getenv("SERVER_NAME"), getpid());
12.     }
13.     return 0;
```

14. }

### [编译]

```
g++ demo.cc -o demo -lfcgi
```

直接运行可执行文件，看看能否正常运行。如果出现缺少库libfcgi.so.0，则自己需要手动把/usr/local/lib/libfcgi.so.0库建立一个链接到usr/lib/目录下：**ln -s /usr/local/libfcgi.so.0 /usr/lib/**（或者把so的库路径添加到/etc/ld.so.conf，并执行ldconfig更新一下）

## 4.2.Web发布

1) 将CGI可执行程序移动到nginx的安装目录下 /usr/local/nginx/cgiabin （文件夹不存在则自己创建）

2) 启动spawn-fcgi管理进程，并绑定server IP和端口（不要跟nginx的监听端口重合）

```
/usr/local/nginx/sbin/spawn-fcgi -a 127.0.0.1 -p 8088 -f /usr/local/nginx/cgiabin/demo
```

查看一下9002端口是否已成功：**netstat -na | grep 8088**

3) 更改**nginx.conf**配置文件，让nginx转发请求

在http节点的子节点-"server节"点中下添加配置

```
location ~ \.cgi$ {  
    fastcgi_pass 127.0.0.1:8088;  
    fastcgi_index index.cgi;  
    fastcgi_param SCRIPT_FILENAME fcgi$fastcgi_script_name;  
  
    include fastcgi_params;  
}
```

4) 重启nginx或者重新加载配置文件

重新加载配置文件

```
sudo kill -HUP [pid]
```

或者

重启nginx

```
killall nginx
```

**./nginx**

5) 打开浏览器访问一下吧

**`http://localhost/demo.cgi`**

搞定收工，心里又小小的激动了一把！

**allenrlin**

**2014/2/18**

---

参考文献与资料

[1]Nginx+FastCGI运行原理 <http://book.51cto.com/art/201202/314840.htm>

[2]Nginx下配置FastCGI <http://www.cppblog.com/woaidongmao/archive/2011/06/21/149090.html>

[3]nginx + fastcgi+c/c++搭建高性能Web框架 <http://blog.csdn.net/marising/article/details/3932938>

[4]什么是CGI、FastCGI、PHP-CGI、PHP-FPM、Spawn-FCGI <http://www.mike.org.cn/articles/what-is-cgi-fastcgi-php-fpm-spawn-fcgi/>