

FastCGI 中文参考手册

(Perl 语言相关部分)

作者: hoowa.sun

想必所有人都了解 CGI 技术的通用性与效率有多糟糕。那么本文所描述的所有内容仅限于一种可以大幅度提高页面处理技术的东东，它的名字叫 FastCGI，他的速度效率最少要比 CGI 技术提高 5 倍以上 (要知道目前各种流行的页面处理技术 ASP、PHP、JSP/Servlet 还没有哪个能超过)。这并不是一种新技术，笔者在 2000 年第一次用过此技术。大家一定惊讶，既然有这种技术为什么我们还要编写这本参考手册，原因显而易见，虽然此技术已经产生自今，但中文资料还是少的可怜。好！废话到此为之止。

FastCGI 的技术原理

如果了解 FastCGI 的技术原理就要了解何为"短生存期应用程序",何为"长生存期应用程序"。

先从 CGI 技术开刀，以下是 CGI 技术的理论：每次当客户请求一个 CGI 的时候，Web 服务器就请求操作系统生成一个新的 CGI 进程。当 CGI 满足要求后，服务器就杀死这个进程。服务器对客户端的每个请求都要重复这样的过程。

而 FastCGI 技术的理论为：FastCGI 程序一旦产生后，他可以持续工作,足够满足客户的请求直到被明确的终止。如果你希望通过协同处理来提高程序的性能，你可以请求 Web 服务器运行多个 FastCGI 应用程序的副本。

CGI 就是所谓的短生存期应用程序，FastCGI 就是所谓的长生存期应用程序。

由于 FastCGI 程序并不需要不断的产生新进程，可以大大降低服务器的压力。并且产生较高的应用效率。

自今，较为流行的 Java 语言 Servlet 技术在设计上是以参考 FastCGI 的技术运行所设计。

FastCGI 的特点

1.打破传统页面处理技术

传统的页面处理技术，程序必须与 Web 服务器或 Application 服务器处于同一台服务器中。这种历史已经早 N 年被 FastCGI 技术所打破，FastCGI 技术的应用程序可以被安装在服务器群中的任何一台服务器，而通过 TCP/IP 协议与 Web 服务器通讯，这样做既适合开发大型分布式 Web 群，也适合高效数据库控制。

2.明确的请求模式

CGI 技术没有一个明确的角色，在 FastCGI 程序中，程序被赋予明确的角色（响应器角色、认证器角色、过滤器角色）。

3.合理的程序结构

起初，真的很讨厌 FastCGI 应用程序的结构要求。没关系，您经过一段时间编写后就会喜欢这种结构，只有这种完全规范的结构才能让您的程序更有效率。

FastCGI 技术支持语言与 Web 服务器

在本手册中仅介绍如何用 Perl 语言来实现 FastCGI 技术。实际，只要符合 FastCGI 技术规范，都可以使用本技术。FastCGI 技术目前支持语言有：C/C++、Java、Perl、Tcl、Python、SmallTalk、Ruby 等。

FastCGI 技术需要在 Web 服务器中安装支持部分才能运行，目前这部分已经完全支持：Apache、IIS、Pi3Web、Zeus 等。

1. Apache 服务器安装方法

Linux 系统 Apache:

首先需要准备好一些安装包，如果您已经安装过 Apache 请参考以下步骤重新编译。

Apache HTTP SERVER

在本文编写的时候，稳定版为 apache.1.3.27。我们这里所使用的是源代码版 http://apache.linuxforum.net/dist/httpd/apache_1.3.27.tar.gz 其他版或则这个地址有错误请到 www.apache.org 找相关版本。

Mod_FastCGI

当前是 2.3.0 版! http://www.fastcgi.com/dist/mod_fastcgi.tar.gz

如果需要下载其他版或地址有错误请到 www.fastcgi.com 的 Servers 部分找到相关版本。

好! 安装包准备完毕，我们可以开始安装过程了。

1. 解压 Apache 安装包 `tar zxvf apache_1.3.27.tar.gz`
2. 解压 `mod_fastcgi.tar.gz` 包
3. 将解压 `mod_fastcgi.tar.gz` 包后的目录复制到 `apache_1.3.27` 解压后目录的 `src/modules` 下并更名为 `fastcgi`
4. 使用 Apache 安装配置命令 `./configure --activate-module=src/modules/fastcgi/libfastcgi.a`
5. 执行 `make`
6. 执行 `make install`
7. 如过程没出现错误说明安装成功。请测试 `<Apache 安装后目录>/bin/httpd -l`
8. 如果列表中有 `mod_fastcgi.c` 表示成功!

配置 Apache 的 httpd.conf

1. 设置 FastCGI 文件的处理类型，请在 `httpd.conf` 包含 `Addhandler` 部分添加一句 `AddHandler fastcgi-script .fcgi .fpl`
这样，Apache 就知道 `.fcgi` 与 `.fpl` 处理为 `fastcgi` 文件
2. 还要为您的网站设置 `ScriptAlias` 解析，这个解析很容易，与普通 CGI 设置方式相同 `ScriptAlias /fcgi-bin/ /usr/local/apache/fcgi-bin/`
我这里这样设置后 `fcgi` 程序将安装在 `/usr/local/apache/fcgi-bin/` 下

测试安装 `test.fcgi`

```
#!/usr/bin/perl
```

```
use FCGI;
my $count = 0;
my $request = FCGI::Request();

while($request->Accept() >= 0) {
    print "Content-type: text/html\n\n" ;
    print $count++;
}
}
```

Win32 系统 Apache:

准备安装包

1. 下载 Apache for win32 的版本，我们就下稳定版 1.3.27
http://nagoya.apache.org/mirror/httpd/binaries/win32/apache_1.3.27-win32-x86-no_src.msi
2. 下载 ActivePerl
<http://hoowa.tab.net.cn/soft/ActivePerl-5.8.0.802-MSWin32-x86.msi>

配置 Apache 以支持 http 服务

1. 安装 ActivePerl,最好安装到 C 盘，因为 Apache 默认是在 C 下
2. 安装 Apache,装好后修改其目录下的 conf 文件夹下的 httpd.conf 文件

寻找到 **ServerName**。这里定义你的域名，如果前面有 #，记得删除它。

寻找到 **ServerAdmin**。这里输入你的 E-Mail 地址。如果你仅仅是单机使用，改不改没什么关系

在 **ServerName** 下面不远有个 **Options** 和 **AllowOverride**，将他们后面的参数去掉改为 **All**,请注意大小写

配置 Apache 以支持 CGI

1. 假设 Perl 安装在 C:\Perl 目录,找到 **ScriptAlias /cgi-bin/ "C:/Apache/cgi-bin"**（假设您将 apache 装在 c 盘下），那么 **C:/Apache/cgi-bin** 就是您存放 **cgi** 的目录
2. 在 **ScriptAlias** 这句下面,Options 和 **AllowOverride**，将他们后面的参数去掉改为 **All**
3. 寻找到 **AddHandler cgi-script .cgi**。删除前面的 #，在后面加上 **.pl**

注意：当您浏览 **cgi** 程序出现 500 错误的时候可以看一下您的程序，是不是以：**#!C:\perl\bin\perl** 开有。

配置 Apache 以支持 Fastcgi

1. 下载 **mod_fastcgi-AP1.dll**
http://www.fastcgi.com/dist/mod_fastcgi-AP1.dll

2. 将.dll 文件复制到../modules 下，并修改 httpd.conf 文件，加入这两句

```
LoadModule fastcgi_module modules/mod_fastcgi-2.4.0-AP1.dll
AddHandler fastcgi-script .fcgi .fpl
```

3. 保存后就可以运行 fcgi 程序了.

注意：如果没有安装 FCGI package,就在命令提示符下输入

```
ppm
```

```
search FCGI
```

```
INSTALL FCGI
```

2. IIS 服务器安装方法

3. 安装 FCGI.pm 模块

在 Linux 系统中：

登陆以下地址 <http://www.cpan.org/modules/by-module/FCGI/>

下载最新版的 FCGI（在本文编写的时候为）<http://www.cpan.org/modules/by-module/FCGI/FCGI-0.67.tar.gz>

解压 tar zxvf FCGI-0.67.tar.gz

编译 perl Makefile.PL

make

make install

完成！

在 Windows 系统中，以及使用 ActivePerl：

进入 MS-DOS 模式或 cmd 环境

输入 ppm 命令

在 ppm 提示符下输入 install FCGI

自动安装……

完成

FastCGI 的程序结构

一个 FastCGI 的应用程序主要包含两部分：初始化部分、回应循环部分。

```
#初始化编码
```

```
#开始回应循环
```

```
    #回应内容
```

```
#回应结束循环
```

```
#!/usr/bin/perl
```

```
#以下内容都是初始化部分
use FCGI;
my $count = 0;
my $request = FCGI::Request();
```

```
#以下内容都是回应部分
while($request->Accept() >= 0) {
    print "Content-type: text/html\n\n" ;
    print $count++;
}
}
```

当应用程序被初始化后，初始编码仅仅只运行一次。初始编码就像开启数据库和编译后的应用程序一样常常是十分耗时的操作。

整个回应部分循环始复，直到客户要求终止。回应循环部分从调用FCGI_Accept开始。FCGI_Accept程序执行后会挂启程序(程序循环部分虽在内存中，但是并不运行)，除非客户对FastCGI应用程序发出请求。客户的请求一旦到达程序，FCGI_Accept会运行回应部分的内容一次，然后再次中止程序，等待客户下一个请求的到来。整个回应循环部分只会在系统管理员或者Web服务器杀死了FastCGI应用程序后才能消除。

FastCGI 应用程序举例

1. 一个典型的FastCGI应用程序，这个程序将在初始化部分初始变量\$count为0，每次请求运行都将累加一。

```
#!/usr/bin/perl
use FCGI;
use strict;

my $count = 0;
my $request = FCGI::Request;

while($request->Accept >= 0){
    $count++;
    print "Content-type: text/html\n\n" ;
    print qq~
        <HTML>
        <HEAD>
        <TITLE>FastCGI</TITLE>
        </HEAD>
        <BODY>请求已发生次数: $count
        </BODY>
        </HTML>~;
}
}
```

2. FastCGI程序为长生存期应用程序，如在设计中有缺陷会产生内存溢出问题，对服务器造成安全隐患。

且这种溢出是不可预见性。本程序初始化了一个控制变量，一旦用户请求次数达到预先设置的数量程序结束，下一次请求将重新初始化。

```
#!/usr/bin/perl
use FCGI;
use strict;

my $count = 0;
my $request = FCGI::Request;
my $session_life = 1000;

while($request->Accept >=0){
    $count++;
    print "Content-type: text/html\n\n" ;
    print qq~
        <HTML>
        <HEAD>
        <TITLE>FastCGI</TITLE>
        </HEAD>
        <BODY>请求已发生次数: $count
        </BODY>
        </HTML>~;

    if ($count>$session_life){
        end;
    }
}
```

书写 FastCGI 的规范要求

FastCGI 应用程序对编写者提出了一种严格要求的挑战。

1. 在程序头必须加载 strict 模块。格式为 use strict;
2. 任何变量的使用之前一定不要忘记声明变量 my \$variable;
3. 不要将一些初始化的内容放在回应部分，这样不仅无法提高效率还会降低效率。例如对数据库的连接部分。
4. 同样，也不要吧什么东西都放在初始化部分。
5. 在程序调试后发现有 BUG 或修改，请先直接修改程序。然后杀死系统中的程序进程，要不你的修改永远不被执行。如果杀不好，就重启 Apache 吧。

FCGI.pm 模块的使用

模块名称: Fast CGI Module

使用格式:

```
use FCGI;
```

```
$request = FCGI::Request();
```

```
while($request->Accept() >=0) {  
    #CGI Process..  
}
```

FCGI 支持的使用方法：

FCGI::Request

```
$request = FCGI::Request(  
    $input_fh,  
    $output_fh,  
    $error_fh,  
    fileno($socket),  
);
```

\$input_fh,\$output_fh,\$error_fh 是设置 input/output/error 的句柄，默认情况下分别是 STDIN、STDOUT、STDERR

\$socket 的用处在于分离 FastCGI 应用程序与 Web 服务器的情况下才需要使用。它允许程序通过设置的套接字与 Web 服务器通讯。

FCGI::OpenSocket(path, backlog)

在程序中创建一个套接字，在建立独立的 FastCGI 应用程序中用到。

FCGI::CloseSocket(socket)

关闭打开的 FCGI::OpenSocket。

\$request->Accept

当有新的请求产生，以上句子返回 0,其他情况下返回-1。

\$request->Attach

重新设置文件句柄连接到相关服务器上。

\$request->Detach

临时分离文件句柄。

\$request->Finish

结束当前请求，这本来由 Accept 来自动完成，但有写程序中在返回给客户请求后可能需要做一些其他工作，比如操作数据库。

\$request->Flush

清洗当前连接中的所有输入输出。

\$env = \$req->GetEnvironment()

返回来环境变量。

(\$in, \$out, \$err) = \$req->GetHandles()

返回标准文件句柄。

\$isfcgi = \$req->IsFastCGI()

返回当前程序运行的状态是否 FastCGI。

FastCGI 的效率与应用范围

效率这个词不仅仅一词出现过。我也不仅一次讲解过效率的由来。

现在所有的解释性语言(Perl、Java、Python)的运行效率，都要比编译形语言(ANSI C)慢的多(一到几十倍)。但奇怪的是，大家似乎看到的是 Jsp/Servlet 不错的效率。这是因为在网络世界中，对效率的影响不在语言的本身，而在以下几个方面：1、技术模型 2、系统 I/O 带宽 3、程序设计。

FastCGI 是一种半成熟(这样说是因为 C 语言成熟用了十多年，自今 Java 语言与 C++语言都是不断在完善中)的技术，它经过多年的应用实践已经获得广泛的支持。

FastCGI 技术适合各种程序设计上的要求，我们建议使用在以下要求中：

1. 想使用 J2EE 以外的一种高效率 OpenSource 技术。
2. 分布式数据库控制系统
3. 高访问量页面访问系统
4. 性能评测
5. 混毕业设计

FastCGI 的资源

FastCGI 技术官方网站：<http://www.fastcgi.com>

FastCGI 中文讨论组：<http://www.ilcatperl.org>

中国 Perl 组织：<http://www.perlchina.org>

FastCGI 技术案例：

<http://www.hotmail.com>

<http://www.yahoo.com>

<http://www.donews.com>

声明

本文作者：hoowa.sun

感谢：royce(部分内容是他编写的)、bloves(找到很多 Bug)

获得方法：<http://www.ilcatperl.org>

任何网站(非赢利性)转载无需经过作者同意，请保留任何版权声明。

未经作者同意禁止任何杂志转载。