

 hepangda 于 2018-08-10 16:43:56 发布 1800 收藏 2

分类专栏：网络编程

 网络编程 专栏收录该内容 0 订阅 3 篇文章 订阅

## FastCGI 协议分析

FastCGI（快速通用网关接口）是一种让交互程序与Web服务器通信的协议，可以算是CGI的增强版本。FastCGI减少了网页服务器与CGI程序之间的开销，使得服务器可以同时处理更多的网页请求。目前很多常见的网页服务器都已经支持了该协议。

### 通信格式

Web服务器与FastCGI程序之间通过流式套接字来通信，既包含Unix套接字（Unix Domain Socket），也可以使用最常见的TCP/IP协议套接字。FastCGI程序提供的内容与普通的CGI程序相似。由于与CGI程序具有环境变量、标准输入和标准输出等众多获取信息的渠道不同，FastCGI程序仅仅通过Socket获取信息。

### 消息头

FastCGI规定，任意一个FastCGI数据包必须以1个8字节的消息头开始：

```
1 struct FCGI_Header {
2     unsigned char version;
3     unsigned char type;
4     unsigned char requestIdB1;
5     unsigned char requestIdB0;
6     unsigned char contentLengthB1;
7     unsigned char contentLengthB0;
8     unsigned char paddingLength;
9     unsigned char reserved;
10 };
```

在此消息头中，第一个字节（version）表示FastCGI协议的版本，目前一般定义为1。第二个字节（type）表示数据包的类型，这些类型有：

```
1 enum FCGI_Type {
2     FCGI_BEGIN_REQUEST = 1, // (WEB->FastCGI) 表示一次请求的开始
3     FCGI_ABORT_REQUEST = 2, // (WEB->FastCGI) 表示终止一次请求
4     FCGI_END_REQUEST = 3, // (FastCGI->WEB) 请求已被处理完毕
5     FCGI_PARAMS = 4, // (WEB->FastCGI) 表示一个向CGI程序传递的环境变量
6     FCGI_STDIN = 5, // (WEB->FastCGI) 表示向CGI程序传递的标准输入
7     FCGI_STDOUT = 6, // (FastCGI->WEB) 表示CGI程序的标准输出
8     FCGI_STDERR = 7, // (FastCGI->WEB) 表示CGI程序的标准错误输出
9     FCGI_DATA = 8, // (WEB->FastCGI) 向CGI程序传递的额外数据
10    FCGI_GET_VALUES = 9, // (WEB->FastCGI) 向FastCGI程序询问一些环境变量
11    FCGI_GET_VALUES_RESULT = 10, // (FastCGI->WEB) 询问环境变量的结果
12    FCGI_UNKNOWN_TYPE = 11 // 未知类型，可能用作拓展
13 };
```

之后第三、四个字节（requestIdB1 requestIdB0）合起来表示本次请求的编号，其中requestIdB1是请求编号的高八位，requestIdB0是请求编号的低八位。这个字段的存在允许Web服务器在一次连接中向FastCGI服务器发送多个不同的请求，只要使用不同的请求编号即可。然后第五、六个字节（contentLengthB1 contentLengthB0）合起来表示消息头后仍有多个字节的数据（数据长度），contentLengthB1表示其高八位，contentLengthB0表示其低八位。数据长度的表示范围在0-65535之间，因而若数据超过65535字节，则必须将之分为多个数据包来传输。第七个字节则表示“填充长度”，为了使字节对齐，提高网络传输效率，可能需要在数据包后附加无用的数据来保证对齐，这些无用数据的长度就存储在paddingLength中，当然，填充长度不可超过255字节。第八个字节为保留字段，目前的FastCGI还不使用这一字段。

### 消息体

在发送完毕消息头之后，应当附加一个消息体。根据消息类型（type字段）的不同，发送的格式也略有区别。

#### FCGI\_BEGIN\_REQUEST

该类型的消息体格式是固定的一个八字节结构体，因此该类型的消息头中表示数据长度的字段值也应当是固定的。格式被定义为：

```
1 struct FCGI_BeginRequestBody {
2     unsigned char roleB1;
3     unsigned char roleB0;
4     unsigned char flags;
5     unsigned char reserved[5];
6 };
```

第一、二字节合起来表示在本次请求中，Web服务器希望CGI程序担当的角色，其中roleB1表示其高八位，roleB0则表示其低八位。目前FastCGI协议中规定了三种角色，分别为：

```
1 enum FCGI_Role {
2     FCGI_RESPONDER = 1, // 响应器
3     FCGI_AUTHORIZER = 2, // 认证器
4     FCGI_FILTER = 3 // 过滤器
5 };
```

一般而言，Web服务器基本需要FastCGI做响应器（FCGI\_RESPONDER）角色。

第三个字节表示Web服务器希望FastCGI所做的一些行为，但目前只有控制FastCGI服务器处理完请求之后是否关闭Socket连接的功能，之后可能会做拓展。

```
1 enum FCGI_Flags {
2     FCGI_KEEP_CONN = 1;
3 };
```

当flags与FLAG5\_KEEP\_CONN按位与结果为1时，则表示FastCGI服务器处理完本次请求之后不关闭连接，否则则关闭该Socket连接。

消息体中的其余字节目前尚无作用，留作拓展。

#### FCGI\_END\_REQUEST

该类型的消息体格式与也是固定的一个八字节结构体，因此该类型的消息头中表示数据长度的字段值也应当是固定的。格式被定义为：

```
1 struct FCGI_EndRequestBody {
2     unsigned char appStatusB3;
3     unsigned char appStatusB2;
4     unsigned char appStatusB1;
5     unsigned char appStatusB0;
6     unsigned char protocolStatus;
7     unsigned char reserved[3];
8 };
```

前四字节合起来表示CGI程序的退出状态，与之前相同，此处是一个网络字节序，需要手动转换。第五字节表示FastCGI协议的状态码，可能的取值有：

```
1 enum FCGI_ProtocolStatus {
2     FCGI_REQUEST_COMPLETE = 0, // 请求正常完成
3     FCGI_CANT_MPX_CONN = 1, // FastCGI服务器不支持并发处理，请求已被拒绝
4     FCGI_OVERLOADED = 2, // FastCGI服务器耗尽了资源或达到限制，请求已被拒绝
5     FCGI_UNKNOWN_ROLE = 3 // FastCGI不支持指定的role，请求已被拒绝
6 };
```

后三字节目前尚无作用。

#### FCGI\_PARAMS

该类型的消息体格式为：

```
1 struct FCGI_ParamsBody {
2     unsigned char nameLengthB3;
3     unsigned char nameLengthB2;
4     unsigned char nameLengthB1;
5     unsigned char nameLengthB0;
6     unsigned char valueLengthB3;
7     unsigned char valueLengthB2;
8     unsigned char valueLengthB1;
9     unsigned char valueLengthB0;
10    unsigned char nameData[NAME_LENGTH]; // NAME_LENGTH与前四字节所表示数字相同
11    unsigned char valueData[VALUE_LENGTH]; // VALUE_LENGTH与第五到八字节所表示的数字相同
12 };
```

消息体的前四字节表示传递给FastCGI的环境参数名的长度，计算方法为（B3 & 0x7f）<< 24）+（B2 << 16）+（B1 << 8）+ B0，第五到八字节则表示环境值的长度，计算方法同上。在这八字节之后，紧跟环境参数名，再跟相应的环境值。

### 其他消息体

其他消息体后直接跟具体内容，没有规定的格式。若type为FCGI\_ABORT，那么不必跟消息体，在消息头中将contentLength设为0即可。另，type为FCGI\_DATA、FCGI\_GET\_VALUES和FCGI\_GET\_VALUES\_RESULT的情况本文不做介绍，可参考FastCGI Specification。

### 通信模式

下面我们模拟一次客户端发起的POST请求，简单说明Web服务器和FastCGI服务之间的通信过程：

```
1 POST /fastcgi.page HTTP/1.1
2 Host: a.host.name
3 Connection: keep-alive
4 Content-Type: application/x-www-form-urlencoded
```

目录

FastCGI 协议分析

通信格式

消息头

消息体

通信模式

CGI的常见参数

CGI的“标准输出”格式

分类专栏

|  |     |
|--|-----|
|  算法           | 5篇  |
|  ACM          | 7篇  |
|  Assembly     | 4篇  |
|  C++          | 12篇 |
|  C 语言         | 2篇  |
|  Linux        | 12篇 |
|  C            | 8篇  |
|  网络编程         | 3篇  |
|  golang       | 1篇  |
|  java         | 1篇  |
|  scala        | 1篇  |
|  闲谈         | 1篇  |
|  XiyouLinux | 1篇  |
|  系统编程       | 1篇  |

```
5 Content-Length: 11
6
7 a=b&c=d&e=f
```

- Web服务器收到、解析请求，和FastCGI服务器建立Socket连接；
- Web->FastCGI：发送 `FCGI_BEGIN_REQUEST` 消息；
- Web->FastCGI：发送 `FCGI_PARAMS` 消息，设置 `REQUEST_METHOD` 变量为 `POST` ；
- Web->FastCGI：发送 `FCGI_PARAMS` 消息，设置 `SCRIPT_NAME` 变量为 `/wwwroot/fastcgi.page`（此处根据CGI程序的不同和请求的不同，填写内容应当不同）；
- Web->FastCGI：发送 `FCGI_PARAMS` 消息，设置 `CONTENT_LENGTH` 变量为 `11` ；
- Web->FastCGI：发送 `FCGI_PARAMS` 消息，设置 `CONTENT_TYPE` 变量为 `application/x-www-form-urlencoded`（3-6步可以以任意顺序执行）；
- Web->FastCGI：发送 `FCGI_PARAMS` 消息，消息头的 `contentLength` 字段应当为0，且不包含消息体；
- Web->FastCGI：发送 `FCGI_STDIN` 消息，消息头的 `contentLength` 字段应当为 `11`（若本来的 `Content-Length` 比65535字节更大，应当分次发送，分次发送时应当注意次序），内容即为 `a=b&c=d&e=f` ；
- Web->FastCGI：发送 `FCGI_STDIN` 消息，消息头的 `contentLength` 字段应当为0，且不包含消息体；
- FastCGI->Web：（假设本次请求不产生错误），发送 `FCGI_STDOUT` 消息，若CGI程序给出的输出大于65535字节，将分次发送。
- FastCGI->Web：发送 `FCGI_STDOUT` 消息，本次消息的 `contentLength` 字段为 `0`，不具有消息体。
- FastCGI->Web：（假设此时请求已完成），按规定的格式发送 `FCGI_END_REQUEST` 消息。

总结一下，一次请求由一个 `FCGI_BEGIN_REQUEST` 消息开始，由一个 `FCGI_END_REQUEST` 消息结束。途中发送的消息，类似于 `FCGI_STDIN`、`FCGI_STDERR` 之类的消息，可以根据需要分次发送，在结束发送该类消息之后，发送一个同类型的 `contentLength` 字段为0的消息，表示该类型的消息已经结束。

需要注意的是，若不发送 `FCGI_PARAMS` 消息设置 `CONTENT_LENGTH` 参数，那么之后发送的 `FCGI_STDIN` 消息无效。若发送的 `FCGI_STDIN` 消息的 `contentLength` 字段之和与 `CONTENT_LENGTH` 的值不同，那么该请求也将失败。

## CGI的常见参数

在我们发送 `FCGI_PARAMS` 消息中，我们常设置以下参数：

| 参数名               | 意义   |
|-------------------|--|
| REQUEST_METHOD    | 信息传输方式（GET/POST/PUT等）                              |
| QUERY_STRING      | 查询字符串  |
| CONTENT_LENGTH    | 向CGI标准输入传递的信息长度（应当等于FCGI_STDIN消息contentLength字段之和） |
| CONTENT_TYPE      | 向CGI标准输入传递的信息类型                                    |
| PATH_INFO         | 路径信息   |
| PATH_TRANSLATED   | CGI程序的完成路径名  |
| SCRIPT_NAME       | 要执行的CGI程序的名字                                       |
| GATEWAY_INTERFACE | 服务器所实现的CGI版本                                       |
| SERVER_NAME       | 服务器的P或名字   |
| SERVER_PORT       | 主机的端口号   |
| SERVER_SOFTWARE   | 调用CGI程序的HTTP服务器的名称和版本号                             |
| REMOTE_ADDR       | 客户机的主机名  |
| REMOTE_HOST       | 客户机的IP地址   |
| ACCEPT            | HTTP请求中Accept字段                                    |
| ACCEPT_ENCODING   | HTTP请求中Accept-Encoding字段                           |
| ACCEPT_LANGUAGE   | HTTP请求中Accept-Language字段                           |
| AUTORIZATION      | HTTP请求中Authorization字段                             |
| IF_MODIFIED_SINCE | HTTP请求中If-Modified-Since字段                         |
| PRAGMA            | HTTP请求中Pragma字段                                    |
| REFERER           | HTTP请求中Referer字段                                   |
| USER_AGENT        | HTTP请求中User-Agent字段                                |

## CGI的“标准输出”格式

CGI的标准输出的基本格式为：

```
1 NAME1: VALUE1\r\n
2 NAME2: VALUE2\r\n
3 ...
4 NAMEn: VALUEn\r\n
5 \r\n
6 (Content)
```

可以看到，CG的标准输出类似于一个HTTP Response，除过没有HTTP Response第一行的内容外，其余均十分相似。CGI的标准输出中有一个特别的部分，名为：`Status`，值为相应的HTTP状态码。CGI程序通过这些内容向Web服务器传递头部信息和期望的状态码。例如下面这个标准输出：

```
1 Status: 404
2 Content-Type: text/html
3
4 <html>
5   <body>
6     <p>404 Not Found</p>
7   </body>
8 </html>
```

则表示希望Web服务器将之状态码设为404，将 `Content-Type` 字段设为 `text/html`。相应内容则为由 `\r\n\r\n` 标识的空白行的内容。

PHP7系列：FastCGI协议详解（八）OooO\_mumux的博文 18

URL请求的过程 CGI FastCGI

Fastcgi 协议解析及 get&post 使用实例weixin\_30535913的博文 482

前言： 基于：csdn1 多神的描述 其实看上面两位大佬的博文就已经oak了，写的目地主要是自己总结学习一下。 基础： 1.基础的 WebServer应该支持客...

评论 1 您还未登录，请先 登录 后发表或查看评论

FastCGI规范\_jdr64的专栏1-17

本规范具有有限的有效性(Validity) 从应用的角度规范FastCGI应用和支持FastCGI的Web服务器之间的接口。 Web服务器的很多特性涉及FastCGI, 单从来说... 1-16

Fastcgi协议定义解释与说明\_小蒙之家1-16

各个值的详细定义参见http://www.fastcgi.com/devkit/doc/cgi-spec.html#S8 以下做一些概要说明 记录(Records, 可以顺序发送或者接受多个记录)的格式... 1-16

【PHP源码分析】FastCGI协议浅析weixin\_34409822的博文 80

【PHP源码分析】FastCGI协议浅析 它是建立在CGI/1.1基础之上的，把CGI/1.1里面的要传递的数据通过FastCGI协议定义的顺序和格式进行... 75

fastcgi协议 管窥星路 75

记录的通用结构： typedef struct { unsigned char version; unsigned char type; unsigned char requestIdB1; unsigned char requestIdB0; unsigned char c... 2-12

Fastcgi、CGI 是什么及区别\_yadong\_word的博文2-12

Fastcgi、CGI 是什么及区别 1. CGI是干嘛的? CGI是为了保证web server传递过来的数据是标准格式的,方便CGI程序的编写者。 2. web server(比如nginx... 2-20

4-14、FastCGI协议理论讲解\_wade1010的专栏2-20

FastCGI是一种协议,它是建立在CGI/1.1基础之上的,在CGI/1.1里面的要传递的数据通过FastCGI协议定义的顺序和格式进行传递。为了更好地理解PHP+FM... 6011

FastCGI介绍lida的博文 6011

本文主要介绍 FastCGI 的相关知识，点击此处查看。 1. 概述 FastCGI 实际上是增加了一些扩展功能的 CGI，是 CGI 的改进，描述了... 17+

FastCGI协议详解及代码实现zhang197093的博文 17+

FastCGI协议是在CGI协议的基础上发展出来的，如果想了解CGI协议，可以看我另一篇文章：动态web技术(二) --- CGI，FastCGI程序本身监听某个socket... 2-24

Fastcgi知识整理\_purpen2-24

当前是2.3.0版本http://www.fastcgi.com/dstmod\_fastcgi.tar.gz 如果需要下载其他版本或地址有错误请到www.fastcgi.com的Servers部分找到相关版本。 好安... 2-18

fastcgi协议分析与实例\_ACM Dream2-18

对于fastcgi这种数据收发协议来说,它所发送的每次请求或是回复(我之后的叙述中叫它们为消息)都有一个可提取的公共部分就是FCGI\_Header(请求头)及... 545

Fastcgi我重获无天的清新的博文 545

fastcgi 相比于cgi(fork+exec的方式，fastcgi一般由一个进程fork一个对应的cgi进程，这个cgi进程一直阻塞等待cgi的请求，这样缺点是浪费内存，优点是... 3323

1.CGI定义 CGI被称为“公共网关接口”(Common Gateway Interface)，HTTP服务器与你的或其它机器上的程序进行“交谈”的一种工具，其程序须运行在网络... 229

Fastcgi原理restasmrph的专栏 229

1.关于cgi 通用网关接口（CommonGatewayInterface:CGI）描述了客户端和服务程序之间的传输数据的一种标准，可以让一个客户端，从网页浏览器向... 372

麻省理工-协议文档 go的转发器实现 Go实现FastCGI Proxy Client 系列（四） Go实现FastCGI Proxy Client 系列（三） Go实现FastCGI Proxy Client 系列... 211

FastCGI介绍(伪接口)sun007700的专栏 211

https://www.cnblogs.com/wuhuqin/archive/2013/06/06/4982008.html CGI（Common Gateway Interface）公共网关接口，是HTTP服务器与其他程序通信... 把票自己。 1610

fastcgi协议分析与实例1.序言 最近正在用c++实现一个简单web服务器。在实现之前大体理清了一下一个web服务器主要做什么事情。当时单纯的认为只需要将浏览器发来的http请求解... 116

FastCGI模块weixin\_34086598的博文 116

FastCGI模块（FastCGI）摘要 这个模块允许nginx同FastCGI协同工作，并且控制哪些参数将被安全传递。例： location / { fastcgi\_pass localhost:9000; ... 619

FastCGI的认识与使用bigfaceCatzzzz的博文 619

hepangda 前9年 暂无认证

41 20万+ 134万+ 7万+ 4级

1146 54 45 31 89

积分 粉丝 获赞 评论 收藏

私信 关注

搜博文

热门文章

[ArchLinux] 安装及KDE桌面环境安装配置 18481

[C++] C++ 17简单上手 (二) —— string\_View 14782

[ArchLinux] 搜狗拼音输入法的安装 10253

[Assembly] 程序的机器级表示 (三)：控制令、算术和逻辑指令 3670

[Asio] Using Asio with C++11 2981

最新评论

[Assembly] 程序的机器级表示 (三)：... pioneer.：谢谢您的解答，已经搞清楚 1

[Assembly] 程序的机器级表示 (三)：... hepangda：类比到C语言的话，假设有ent'p... 不加括号类似于直接用p的概念，加括... 1

[Assembly] 程序的机器级表示 (三)：... pioneer.：你好问一下，寄存器%rsp加上括号()变成(%rsp)代表什么意思？还有哪... 1

[这很难吗？] C/C++ 求Fibonacci数列的... TonyBringwater：计算器打表哈哈哈哈哈哈 1

[IPC] 共享内存——分析和使用 fuji\_... 疫情 我说怎么感觉文章质量如此之... 原来是撞到了hepangda大神的博文 1

您愿意向朋友推荐“博客详情页”吗？

01

02

03

04

05

06

07

08

09

10

11

12

CGI 1.1 什么是 CGI 通用网关接口(Common Gateway Interface, CGI)描述了客户端和服务服务器之间传输数据的一种标准。可以让一个客户端,从

FastCGI 协议详解

为什么需要 FastCGI 因为客户端不能理解 CGI, web 服务器充当了客户端的代理, 和后端通信; web 服务器通过 CGI 协议与后端的程序进行交互, 完成

PHP 源码中 FastCGI 模块浅析 商业广告

目录: 1. 初识 FastCGI 协议 1.1 快速理解 2. 基础认识 2.1 FastCGI 配置 2.2 FastCGI 配置预处理 3. 构造 FastCGI 请求 3.1 FastCGI 请求构造 3.2 3.3

FastCGI 使用介绍 [更新发布](#)

FastCGI 介绍 FastCGI : 快速通用网关接口 ( Fast Common Gateway Interface / FastCGI ) 是一种让交互程序与 Web 服务器通信的协议。 FastCGI 息...

Super 博客的推荐

layui 的推荐

alpha 熊的推荐

©2022 CSDN 皮肤主题: 大白 设计师: CSDN 官方商务 返回首页

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kfu@csdn.net 在线客服 工作时间 8:30-22:00

hepangda 关注

点赞 2 评论 1 收藏 1 分享 1 专栏目录






 2
 
 1
  2
 

[专栏目录](#)