Atitit 网关协议cgi wsgi fcgi fastcgi

目录

[1.1. CGI(common gateway unterface) 1](#_Toc13205)

[1.2. 2.1 WSGI: 1](#_Toc29836)

[1.3. 2.3 uwsgi: 2](#_Toc29189)

## CGI(common gateway unterface)

wsgi: Web服务器网关接口（Python Web Server Gateway Interface，缩写为WSGI）CGI(common gateway unterface)

它最初是在1993年由NCSA（美国国家超级电脑应用中心）为NCSA HTTPd Web服务器开发的。

CGI的第一个实现是采用Perl编写的。Perl被广泛用于编写CGI程序，但CGI本身是独立于任何语言的。事实上，CGI程序可以使用任何脚本语言甚至是编译型语言分别独立实现，比如Unix Shell、Python、Ruby、PHP、C/C++等。

服务器端 CGI 程序接收信息有三种途径：环境变量、命令行和标准输入。其中环境变量是指 CGI 定义一组环境变量，通过环境变量可传递数据。服务器收到来自浏览器的数据，调用 CGI 脚本，CGI 脚本将收到的数据转换成环境变量并从中取出所需要的内容。<form>标签的 METHOD 属性来决定具体使用哪一种方法。在“METHOD=GET”时，向 CGI 传递表单编码信息的是通过命令来进行的。表单编码信息大多数是通过环境变量 QUERY\_STRING 来传递的。若“METHOD=POST”，表单信息通过标准输入来读取。还有一种不使用表单就可以向 CGI 传送信息的方法，那就是把信息直接附在 URL 地址后面，信息和URL 之间用问号（?）来进行分隔。GET 方法是对数据的一个请求，被用于获得静态文档。GET 方法通过将发送请求信息附加在 URL 后面的参数。当 GET 方法被使用时，CGI 程序将会从环境变量 QUERY\_STRING获取数据。为了正确的响应客户端发来的请求，CGI 必须对 QUERY\_STRING 中的字符串进行分析。当用户需要从服务器获取数据，但服务器上的数据不得改变时，应该用 GET 方法；但是如果请求中的字符串超过了一定长度，通常是 1024 字节，那么这时，只能用 POST 方法。POST 方法：浏览器将通过填写表单将数据传给服务器时一般采用POST 方法。在发送的数据超过 1024 字节时必须采用 POST 方法。当 POST 方法被使用时，Web 服务器向CGI 程序的标准输入 STDIN 传送数据。环境变量 CONTENT\_LENGTH 存放着发送的数据长度。CGI 程序必须检查环境变量 REQUEST\_METHOD 以确定有没有采用了 POST 方法，并决定是否要读取标准输入STDIN [3]  。

## 环境变量列表

[编辑](https://baike.baidu.com/item/CGI/javascript:;)

SERVER\_NAME：运行CGI序为机器名或IP地址。

SERVER\_INTERFACE：WWW[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "https://baike.baidu.com/item/CGI/_blank)的类型，如：CERN型或NCSA型。

SERVER\_PROTOCOL：通信协议，应当是HTTP/1.0。

SERVER\_PORT：TCP端口，一般说来web端口是80。

HTTP\_ACCEPT：HTTP定义的浏览器能够接受的数据类型。

[HTTP\_REFERER](https://baike.baidu.com/item/HTTP_REFERER" \t "https://baike.baidu.com/item/CGI/_blank)：发送[表单](https://baike.baidu.com/item/%E8%A1%A8%E5%8D%95" \t "https://baike.baidu.com/item/CGI/_blank)的文件URL。（并非所有的浏览器都传送这一变量）

HTTP\_USER-AGENT：发送表单的[浏览](https://baike.baidu.com/item/%E6%B5%8F%E8%A7%88" \t "https://baike.baidu.com/item/CGI/_blank)的有关信息。

GETWAY\_INTERFACE：CGI程序的版本，在UNIX下为 CGI/1.1。

PATH\_TRANSLATED：PATH\_INFO中包含的实际路径名。

PATH\_INFO：浏览器用GET方式发送数据时的附加路径。

SCRIPT\_NAME：CGI程序的路径名。

QUERY\_STRING：[表单](https://baike.baidu.com/item/%E8%A1%A8%E5%8D%95" \t "https://baike.baidu.com/item/CGI/_blank)输入的数据，URL中问号后的内容。

REMOTE\_HOST：发送程序的[主机名](https://baike.baidu.com/item/%E4%B8%BB%E6%9C%BA%E5%90%8D" \t "https://baike.baidu.com/item/CGI/_blank)，不能确定该值。

REMOTE\_ADDR：发送程序的机器的IP地址。

REMOTE\_USER：发送程序的人名。

CONTENT\_TYPE：POST发送，一般为application/xwww-form-urlencoded。

CONTENT\_LENGTH：POST方法输入的数据的字节数。

## ****FastCGI****

FastCGI是一个可伸缩地、高速地在HTTP server和动态脚本语言间通信的接口。多数流行的HTTP server都支持FastCGI，包括Apache、Nginx和lighttpd等，同时，FastCGI也被许多脚本语言所支持，其中就有PHP。

FastCGI是从CGI发展改进而来的。传统CGI接口方式的主要缺点是性能很差，因为每次HTTP服务器遇到动态程序时都需要重新启动脚本解析 器来执行解析，然后结果被返回给HTTP服务器。这在处理高并发访问时，几乎是不可用的。FastCGI像是一个常驻(long-live)型的CGI， 它可以一直执行着，只要激活后，不会每次都要花费时间去fork一次(这是CGI最为人诟病的fork-and-execute 模式)。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。它的速度效率最少要比CGI 技术提高 5 倍以上。它还支持分布式的运算, 即 FastCGI 程序可以在网站服务器以外的主机上执行并且接受来自其它网站服务器来的请求。

FastCGI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。众所周知，CGI解 释器的反复加载是CGI性能低下的主要原因，如果CGI解释器保持在内存中并接受FastCGI进程管理器调度，则可以提供良好的性能、伸缩性、 Fail-Over特性等等。FastCGI接口方式采用C/S结构，可以将HTTP服务器和脚本解析服务器分开，同时在脚本解析服务器上启动一个或者多 个脚本解析守护进程。当HTTP服务器每次遇到动态程序时，可以将其直接交付给FastCGI进程来执行，然后将得到的结果返回给浏览器。这种方式可以让 HTTP服务器专一地处理静态请求或者将动态脚本服务器的结果返回给客户端，这在很大程度上提高了整个应用系统的性能。

FastCGI的工作流程:

1. Web Server启动时载入FastCGI进程管理器（PHP-CGI或者PHP-FPM或者spawn-cgi)
2. FastCGI进程管理器自身初始化，启动多个CGI解释器进程(可见多个php-cgi)并等待来自Web Server的连接。
3. 当客户端请求到达Web Server时，FastCGI进程管理器选择并连接到一个CGI解释器。Web server将CGI环境变量和标准输入发送到FastCGI子进程php-cgi。
4. FastCGI子进程完成处理后将标准输出和错误信息从同一连接返回Web Server。当FastCGI子进程关闭连接时，请求便告处理完成。FastCGI子进程接着等待并处理来自FastCGI进程管理器(运行在Web Server中)的下一个连接。 在CGI模式中，php-cgi在此便退出。

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）。

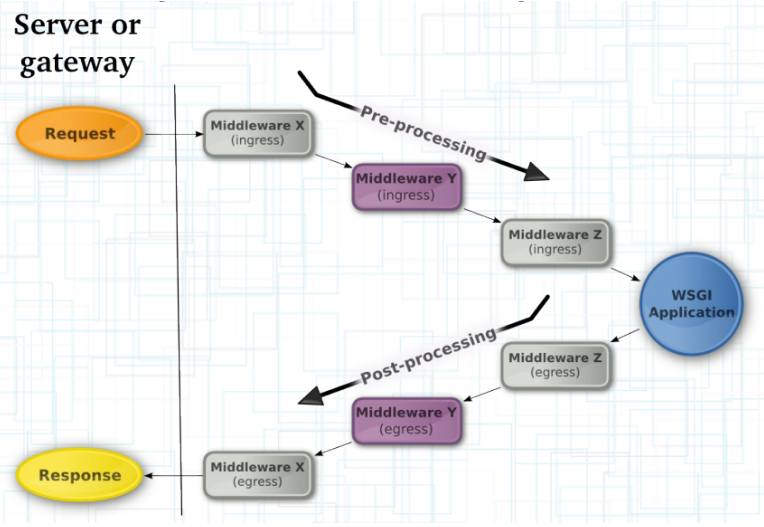
**2.uWSGI ,WSGI和uwsgi的区别**

## 2.1 WSGI:

WSGI，全称 Web Server Gateway Interface，或者 Python Web Server Gateway Interface ，是为 Python 语言定义的 Web 服务器和 Web 应用程序或框架之间的一种简单而通用的接口。也可以认为WSGI是一种通信协议。自从 WSGI 被开发出来以后，许多其它语言中也出现了类似接口。

WSGI 的官方定义是，the Python Web Server Gateway Interface。从名字就可以看出来，这东西是一个Gateway，也就是网关。网关的作用就是在协议之间进行转换。

WSGI 是作为 Web 服务器与 Web 应用程序或应用框架之间的一种低级别的接口，以提升可移植 Web 应用开发的共同点。WSGI 是基于现存的 CGI 标准而设计的。



2.3 uwsgi:

uwsgi是服务器和服务端应用程序的一种协议，规定了怎么把请求转发给应用程序和返回; uwsgi是一种线路协议而不是通信协议，在此常用于在uWSGI服务器与其他网络服务器的数据通信