

Web服务器工作原理详解（基础篇）

原创

置顶

青城山小和尚

于 2018-08-14 17:12:00 发布


阅读量7.2w

收藏 663

点赞数 166

分类专栏: Linux-C

文章标签: Web服务器 HTTP CGI 动态网页 HTTP服务器

Linux-C 专栏收录该内容

24 订阅 9 篇文章

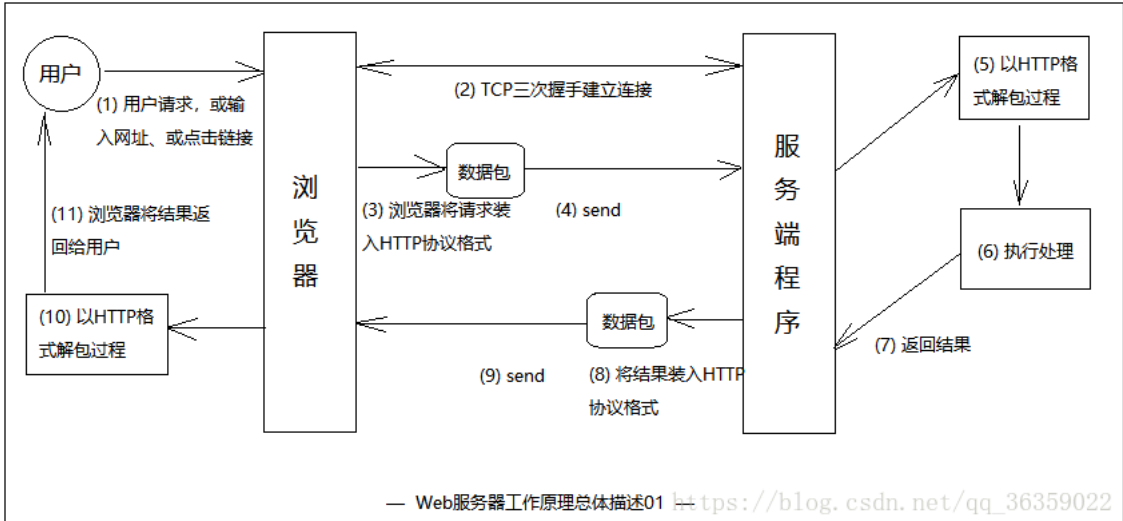
概述：Web服务器概念较为广泛，我们最常说的Web服务器指的是网站服务器，它是建立在Internet之上并且驻留在某种计算机上的程序。Web服务器Web客户端(如浏览器)提供文档或其他服务，只要是遵循HTTP协议而设计的网络应用程序都可以是Web客户端。

Web服务器和HTTP服务器可以说是同一个东西，当然非得细分的话，HTTP服务器是建立在HTTP协议之上的提供文档浏览的服务器，更多的是提供件。而Web服务器涵盖了HTTP服务器(这一点可以自行百度百科)，Web服务器不仅能够存储信息，还能在用户通过Web浏览器提供的信息的基础上运序。

Web服务器 约等于 HTTP服务器 + 其他服务

目前所熟知的Web服务器有很多，其最主流的是 Apache, Nginx, IIS
各大Web服务器的实现细节都不同，是为了某种情形而设计开发的。但是它们的基础工作原理是相同的，这也是本次基础篇所讲解的内容。

一、Web服务器工作 原理图 解



首先我们暂时不考虑HTTP协议的各种请求方式，我们先跟着**（Web服务器工作原理总体描述01）这张图，将一次Web服务的工作流程过一遍，我们浏览器作为客户端

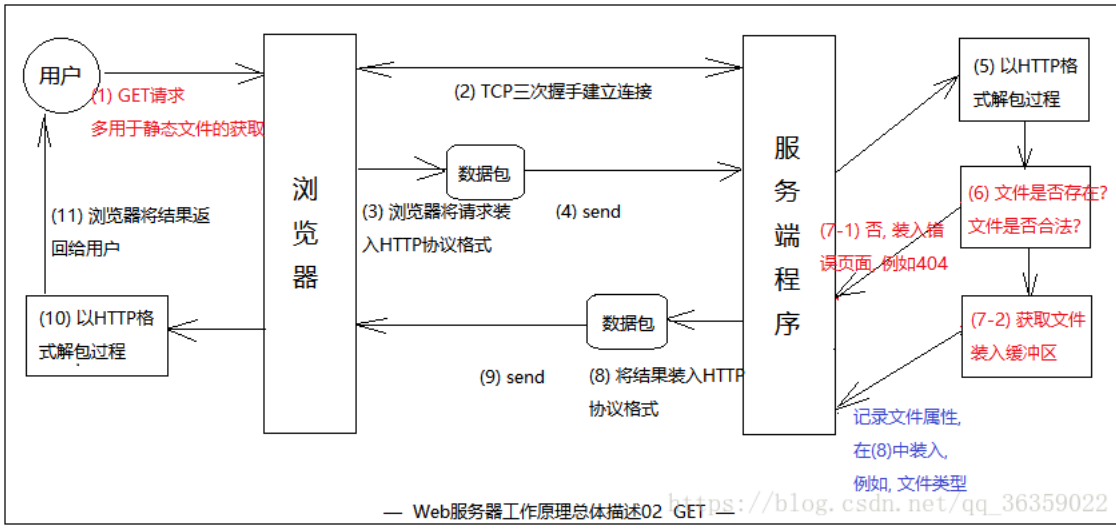
- (1) 用户做出了一个操作，可以是填写网址敲回车，可以是点击链接，可以是点击按键等，接着浏览器获取了该事件。
- (2) 浏览器与对端服务程序建立TCP连接。
- (3) 浏览器将用户的事件按照HTTP协议格式**打包成一个数据包，其实质就是在待发送缓冲区中的一段有着HTTP协议格式的字节流。
- (4) 浏览器确认对端可写，并将该数据包推入Internet，该包经过网络最终递交到对端服务程序。
- (5) 服务端程序拿到该数据包后，同样以HTTP协议格式解包，然后解析客户端的意图。
- (6) 得知客户端意图后，进行分类处理，或是提供某种文件、或是处理数据。
- (7) 将结果装入缓冲区，或是HTML文件、或是一张图片等。
- (8) 按照HTTP协议格式将(7)中的数据打包
- (9) 服务器确认对端可写，并将该数据包推入Internet，该包经过网络最终递交到客户端。
- (10) 浏览器拿到包后，以HTTP协议格式解包，然后解析数据，假设是HTML文件。
- (11) 浏览器将HTML文件展示在页面

以上为Web服务器工作基本原理。其实不难发现，这仅仅只是一个简单的网络通信。我们应该深信，作为一个服务器，其根本的工作无非有三个

- 1. 接收数据 2. 发送数据 3. 数据处理
- 而Web服务器的本质就是 接收数据 → HTTP解析 → 逻辑处理 → HTTP封包 → 发送数据
- 高级的服务器无非就是将这三个部分更加细致的设计了。

二、Web服务器之提供静态文件工作原理图解

Web服务器最主要的功能是提供静态的文件。日常的上网浏览大多是网页浏览，少数时候才会有些数据的提交操作。因此，我们结合上一张图示来GET请求下的Web服务器工作原理。



其他流程基本不变，着重在于红色与蓝色部分。

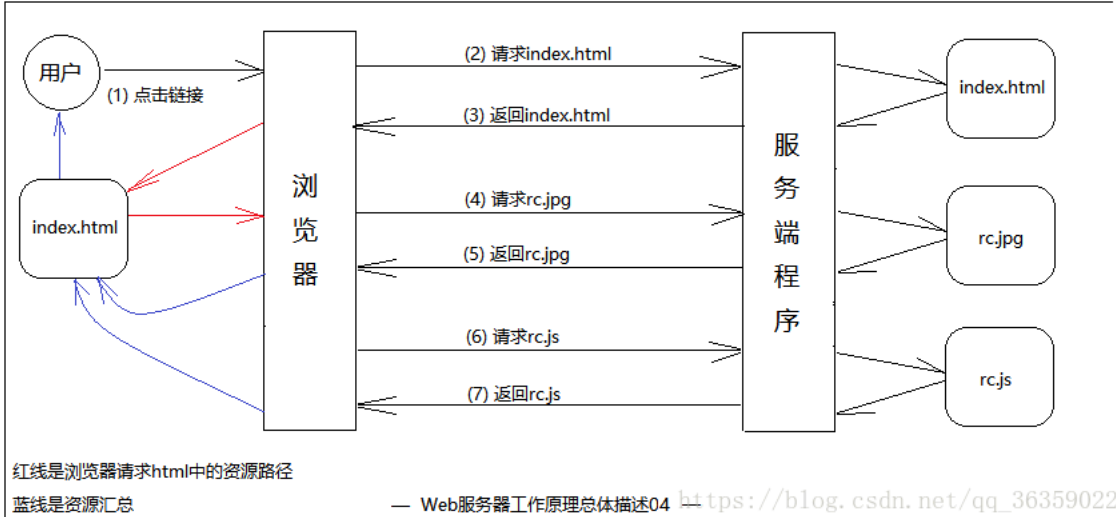
(1) 当用户点击一个网页链接或浏览器加载一些资源(css.jpg ...)时产生。

(6) 服务程序解包后，确定其为GET请求，并且是对该服务器上的某一资源的请求。首先服务程序会去确认该路径是否存在，再确定该路径的文件是否可取。

(7-1) 如果请求的路径有误，或者该资源不能被用户获取，则返回错误提示页面。很多服务器的错误页面只有404，更专业的应该是将错误分类并返回代码页面。

(7-2) 如果该路径合法且文件可以被获取，那么服务程序将根据该文件类型进行不同的装载过程，记录其类型作为(8)中HTTP协议中对应的返回类型，头。

假设以点击一个页面链接为例，浏览器首先将HTML文件请求过来，再以同样的流程对HTML文件中包含的资源文件路径进行依次请求。



三、Web服务器之数据提交工作原理图解

仅仅只是网页的浏览并不能满足所有人的需求，客户端与服务器应当是有数据交互的。

即使单方面的资源请求任然是网络的主力军。

我们应该清楚的知道，数据提交对于用户来说有什么作用。

(1) 资源上传 (2) 登陆验证 (3) API接口调用 (4) 远程指令等

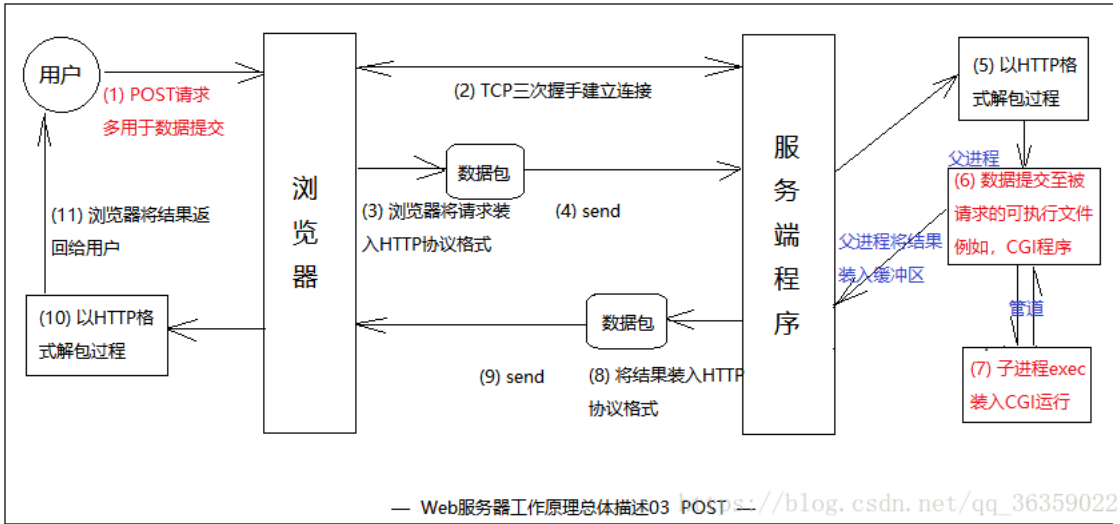
数据提交使得用户的操作性有了质的飞跃，它使得HTTP短连接获取静态文件的方式提升到了动态交互的层次上。该性质也催化出各式各样的编程语言例如PHP，JavaWeb。

如果你留意目前主流的那些大型服务器，你会发现再高级再牛逼的东西实际是也是最基础的东西建造的。那么我们还可以顺便学习一下最古老的动态



青城山小和尚

关注



其他流程基本不变，着重在于红色与蓝色部分。

- (1) 用户提交数据，假设用户点击一个按钮提交填好的信息。在(3)中将以POST格式写入，并填入提交至服务端的可执行程序的路径。
 - (6) 服务端将参数与该CGI绑定，复制进程，用管道传递参数和接收结果
 - (7) 子进程执行CGI，接收(6)父进程传来的参数，运算完成返回结果。
- 最后父进程将结果装入静态模板文件，放入缓冲区

四、动态技术

我们得明白，Web服务器是以短连接为主，并且获取的数据到达浏览器的那一刻一定是静态的不变的。那么所谓动态实际是指两种情况

1. 服务端产生：

- (1) 用户POST提交数据到某个程序，程序根据该数据作为参数运行，得出结果并装入静态的模板页面中，返回该静态页面。但对于用户来说，同做了一个操作后数据不一样了。好了，这就是动态页面。(CGI原理)
- (2) PHP的原理是，用户GET请求一个php后缀的文件，服务器先执行该php后缀文件中的PHP代码，将结果填入代码的位置，再返回。当然也可参与运算再返回。

2. 客户端产生：

- (1) 用户GET请求一个JavaScript文件，服务端不做任何运算返回该静态文件。浏览器收到该JS文件，在本地执行并更新页面。
- (2) 用户POST提交数据到服务端，服务端根据该提交的数据指令返回静态文件，浏览器收到后执行并更新。

文章知识点与官方知识档案匹配，可进一步学习相关知识

网络技能树 首页 概览 46205 人正在系统学习中

小型WEB服务器MyWebServer_gr.bkill.com.zip

【MyWebServer：一款小型WEB服务器详解】在互联网技术领域，Web服务器是不可或缺的一部分，它们负责接收HTTP请求并返回HTML页面或相关资源。本篇文章将

决战Nginx：系统卷 - 高性能Web服务器详解与运维.part2

《决战Nginx系统卷：高性能Web服务器详解与运维》第一部分首先讲述了Nginx服务器的功能、模块管理和进程管理，然后讲述Nginx如何处理请求，在这个基础之上再讨

7 条评论



caoxh_1 热评 写的很好，感谢分享

Web_服务原理_web服务运行的原理

默认工作在 TCP 80 端口 HTTPS(HTTP over SSL) 基于SSL 的 HTTP 协议.使用 SSL 协议来保护 HTTP 传输,使 HTTP 协议更加安全 浏览器(Browser): 可以接收并解析 H

Web 程序运行原理简介

VI

Web 程序运行原理简介 用 Go 语言实现的一个最简单的 Web 服务器 对 Go 语言不了解的可以跳过代码 package main import ("fmt" "net/http") func hello (w http.Respon:

WEB工作原理

weixin_43627016的

一、web的工作原理： 1、用户在浏览器中输入要访问的web站点地址或在已打开的站点点击超链接。 2、由dns进行域名解析，找到服务器的IP地址，向该地址指向的wel

物联网系统运维——移动电商服务器单点部署，web服务器部署，Nginx Web服务介绍,Nginx性能,部署，架构，及实验： ... 最新发布 WZY22502701的

●模块化设计良好的扩展性，可以通过模块方式进行功能扩展。主控进程和worker是同步实现的，一个worker出现问题，会立刻启动另一个worker。一万个长连接(keep-a

Web服务器工作原理详解
服务器



青城山小和尚

关注