

# CGI 调试系统的设计与实现

贺龙涛<sup>1</sup> 方滨兴<sup>2</sup> 云晓春<sup>1</sup>

<sup>1</sup>(哈尔滨工业大学计算机科学与技术系, 哈尔滨 150001)

<sup>2</sup>(国家计算机网络与信息安全管理中心, 北京 100031)

E-mail: hlt@hope.hit.edu.cn

**摘要** Web 应用开发的各种技术中, CGI 技术占有重要地位。文章简单介绍了 CGI 技术, 并分析了 CGI 程序的运行环境。在分析了各种传统的 CGI 调试技术的优缺点的基础上, 提出了一个通用的、操作系统平台无关、Web 服务器平台无关的 CGI 调试系统。

**关键词** CGI Web 服务器 浏览器 调试

**文章编号** 1002-8331-(2001)21-0034-03 **文献标识码** A **中图分类号** TP391

## Design and Implement of an Debugger for CGI

He Longtao<sup>1</sup> Fang Binxing<sup>2</sup> Yun Xiaochun<sup>1</sup>

<sup>1</sup>(Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

<sup>2</sup>(National Computer Network and Information System Security Administration Center, Beijing 100031)

**Abstract:** CGI plays an important role in Web application development. CGI technology is introduced briefly and the run time environment of CGI program is analyzed. After the advantage and disadvantage of many traditional CGI debugging technologies are analyzed, a common CGI debugging system is put forward, which is OS and Web Server independent.

**Keywords:** CGI, Web server, browser, Debug

### 1 引言

Web 技术的发展强烈地刺激了 Internet 的发展, 而 Web 技术的发展则取决于 Web 页面的用户友好性和提供的信息多样性。为了不断地提高 Web 页面的友好性和多样性, 主要有三方面的技术:

(1) Client 端技术。也就是在 Web 浏览器中增加新的功能, 不断满足用户对浏览器不断增加的需求。这方面已经做了大量的工作:

HTML<sup>[1]</sup>语言版本由 2.0、3.2、4.0、4.01 到现在的 XHTML1.0 版, 不断进行升级; XML(可扩展标记语言)大大地扩展了 Web 界面的描述、定制能力;

各种动态解释脚本如 JavaScript、Jscript、VBScript 的出现使得 Web 浏览器的功能进一步扩展, 界面更加友好;

Plug-ins 技术如 Active-X 技术, 以及 Java Applet 的出现, 使得 Web 的交互能力大大增强。

(2) Client-Server 通讯协议。主要是提高 Client 与 Server 间网络传输的效率和安全性。HTTP<sup>[2]</sup>协议由 0.9、1.0 到 1.1, 效率不断提高, 另外 Netscape 公司提出的 Web 加密传输协议 SSL<sup>[3]</sup>也由 1.0、2.0 到 3.0 版, 提供的越来越多的加密方式, 使得 Web 访问更加安全。

(3) Server 端技术。也就是在 Web 服务器中增加新的功能。传统的静态网页技术已经远不能满足需求, 已经开发出多种动态网页生成技术:

CGI(Common Gateway Interface), 即公用网关接口, 这是提出最早、最成熟、支持操作系统平台、服务器类型最多的动态网页生成技术;

服务器 API 技术, 主要包括 ISAPI 和 NSAPI 技术, 由于 API 技术与服务器紧密结合, 运行效率高, 但是适应面窄, 开发周期长;

HTML 脚本嵌入技术, 包括 ASP、PHP 和 JSP, 目标是让 Web 程序员快速的开发出动态的网页。

CGI 作为一种成熟的服务器端技术, 广泛地应用于各种 Web 服务环境, 提供了强大的动态网页技术, 以及与数据库访问能力。但是, 由于 CGI 程序调用方式及运行环境的特殊性, 使得的 CGI 程序难以使用传统的调试工具来进行调试, 成为 web 开发人员难以回避、却又难以解决的难题。

文章在分析 CGI 的调用方式及运行环境, 研究传统的 CGI 调试方法之后, 设计并实现了一个 CGI 程序调试系统, 该系统结合传统调试工具对 CGI 程序进行调试, 大大地加速了 CGI 程序开发过程。

### 2 CGI 简介及工作环境分析

CGI<sup>[4]</sup>(公用网关接口)是 Web 服务器和运行在服务器端的外部程序之间的一个接口规范, 按照这个规范编写并运行的外部程序就叫做 CGI 程序, 其目的是实现 Web 服务器与客户端浏览器之间的动态交互。CGI 程序通过由 Web 服务器来调用

**基金项目:**国防重点预研项目

**作者简介:**贺龙涛, 男, 1974 年生, 博士研究生, 研究方向: 计算机网络与信息安全, 网络应用技术, 网络病毒检测与防范。方滨兴, 博士生导师, 研究方向: 计算机网络与信息安全, 计算机病毒并行计算, 体系结构。

实现和 Web 浏览器的交互,如图 1 所示,Web 服务器将 Web 浏览器发送来的信息,传送给 CGI 程序,由 CGI 程序进行处理,CGI 程序在处理完后将响应结果再回送给 Web 服务器,然后再由 Web 服务器发送到 Web 浏览器。如果用到了外部服务如数据库等,均由 CGI 程序去与外部服务进行交互。

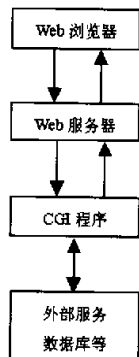


图 1 浏览器、服务器和 CGI 程序的交互

CGI 程序一般完成 Web 网页中表单(Form)数据的处理、数据库查询和实现与传统应用系统的集成等工作。CGI 程序可以用任何程序设计语言编写,如 Shell 脚本语言、Perl、C/C++ 语言等。

CGI 接口标准包括环境变量、标准输入、标准输出三部分。当 Web 浏览器向 Web 服务器发出一个包含有激活 CGI 程序的访问请求时,Web 服务器就为这个 CGI 程序设置运行上下文并执行它。CGI 程序通过环境变量和标准输入读取 Web 服务器提供的客户端输入数据,然后对这些数据进行处理。处理完成后,CGI 程序通过标准输出 Web 服务器将处理结果传送给浏览器。

## 2.1 环境变量

环境变量是文本串(名字/值对),它们定义了程序的执行环境,可以被操作系统或其他程序设置,也可以被其他程序访问。环境变量是 Web 服务器传递相关信息给 CGI 程序的主要手段。Web 服务器为 CGI 程序设置了一些环境变量,用来向 CGI 程序传递一些重要的参数。以下是 CGI 程序设计中常常要用到的一些环境变量:

REQUEST\_METHOD:指的是当 Web 服务器传递数据给 CGI 程序时所采用的方法,分为 GET 和 POST 两种方法。GET 方法仅通过环境变量 QUERY\_STRING 来传递数据给 CGI 程序,而 POST 方法通过标准输入传递数据给 CGI 程序,并使用环境变量 CONTENT\_LENGTH 来指定传递的数据长度。因此 POST 方法可较方便地传递较多的数据给 CGI 程序。

QUERY\_STRING:当使用 GET 方法时,Form 中的数据最后放在 QUERY\_STRING 中,传递给 CGI 程序。

CONTENT\_LENGTH:当使用 POST 方法时指定通过标准输入传递给 CGI 程序的数据字节数。

CONTENT\_TYPE:传递给 CGI 程序数据的 MIME 类型,通常为“application/x-www-form-urlencoded”,它是从 HTML Form 中以 POST 方法传递数据给 CGI 程序的数据编码类型,称为 URL 编码类型。

REMOTE\_ADDR、REMOTE\_HOST:访问该 CGI 程序的 Web 浏览器的机器的 IP 地址及域名。

SCRIPT\_NAME、SCRIPT\_FILENAME:该 CGI 程序的名称及绝对路径名。

## 2.2 标准输入

当 Web 服务器传递数据给 CGI 程序采用 POST 的方法时,CGI 程序像其他可执行程序一样,可通过标准输入(stdin)从 Web 服务器得到输入信息,如 Form 中的数据。

## 2.3 标准输出

CGI 程序通过标准输出(stdout)将输出信息传送给 Web 服务器。传送给 Web 服务器的信息可以用各种格式,通常是以纯文本或者 HTML 文本的形式。Web 服务器对这些结果进行必要的检查,如果 CGI 程序产生的结果格式有问题,web 服务器会给出一些错误信息,必要的时候还会给出错误日志信息。如果程序运行结果格式正确,Web 服务器就会把结果发给客户,这中间 Web 服务器还要加上一些 HTTP 协议所需要的 HTTP 头信息。CGI 的输出必须是包含一个格式如 HTTP 的头、一个空行、以及正文信息,且只能包含 Content-type, Location, Status 三种头。

## 3 传统 CGI 调试方法

由于 CGI 运行时的大量信息是由 Web 服务器为其创建的环境变量中得到,这使得通常的 CGI 调试是繁琐而效率低下的。依据调试环境的不同,主要分为两大类:

### (1)结合 Web 服务器功能的调试。

由于在正常情况下,CGI 程序的运行环境是由 Web 服务器创建并提供的,因此直接在 Web 服务器环境下调试 CGI 程序就是很自然的事了。也就是,用 Web 浏览器来访问要调试的 CGI 程序,直接在浏览器上观察 CGI 程序的输出是否正确,在不正确的时候根据结果检查 CGI 程序,以便改正错误。但是从 Web 浏览器上得到的信息往往是不够的,当 CGI 程序非正常退出时,Web 服务器常常只给 Web 浏览器返回一个“Internal Server Error”错误信息,在这种情况下,可以去察看 Web 服务器的错误日志文件(在 Apache 下,错误日志文件名由配置文件 httpd.conf 中的 ErrorLog 指令指定),因为 Web 服务器将 CGI 程序由标准错误输出(stderr)写出的信息写到错误日志文件中,因此可以从错误日志文件中获取很多有利于判断错误位置的信息。具体调试方法有:静态调试,即简单地根据 Web 浏览器或者错误日志中提供的错误信息去检查 CGI 程序,发现错误。源码级调试,即在静态调试找不到错误问题所在位置时,调试者可以在 CGI 源程序的一些关键点或可疑点插入输出代码,将一些关心的信息显示输出到标准输出 stdout 或者标准错误输出 stderr 上,然后用 Web 浏览器访问 CGI 程序,再在 Web 浏览器上或者错误日志文件中观察输出的结果。

总的说来,结合 Web 服务器功能的调试技术由于没有利用动态调试工具,需要调试者具有丰富的编程语言经验,调试效率极低。

### (2)外壳型调试。

由于 CGI 程序要求的运行环境是已知的,可以编写一个外壳程序来完全模拟 Web 服务器为 CGI 程序创建运行环境,然后直接执行 CGI 程序或者调用动态调试工具来调试 CGI 程序<sup>[9]</sup>。外壳型调试需要研究 Web 服务器的行为,才能对 Web 服务器进行完全的模拟,CGI 程序的运行环境也需要调试者来准备,优点是能够对 CGI 程序进行动态调试,有利于尽快地发现

问题。

### (3) 专用工具型调试。

为了能够迅速开发和调试 CGI 程序,已经出现专用的 CGI 开发环境,例如 VDB<sup>®</sup>就是一个专用于调试用 Perl 语言书写的 CGI 脚本程序的工具。专用工具能够方便地模拟 CGI 所需的运行环境,动态地调试 CGI 程序,调试效率较高,但是,使用专用工具生成的 CGI 程序,一般平台依赖性较强,语言种类也有限。

## 4 CGI 调试系统的设计思想

为了克服专用工具型调试对平台的依赖性,以及对语言种类的限制,文章提出了一种可以广泛使用的 CGI 调试系统。

分析以上非专用工具型的调试方法发现,结合 Web 服务器功能的调试技术虽然调试起来很麻烦,但是它不需要专门为调试进行麻烦的准备,直接随着 CGI 程序在指定 Web 站点的发布而调试,操作简单,且调试与发布无缝连接;而外壳型调试技术对动态调试的支持很强,但是需要调试者自己另行编程准备调试所需的上下文,这样就可以考虑取长补短,将结合 Web 服务器功能的调试技术与外壳型调试技术结合在一起,使得既对调试无须进行复杂的准备,又可以进行动态调试。要将结合 Web 服务器功能的调试技术与外壳型调试技术结合在一起,主要需要解决两方面的问题:

(1)如何能够方便地为调试 CGI 程序准备运行环境。由于 Web 服务器在接收到对 CGI 程序的访问的时候,会为 CGI 程序准备相应的运行环境,这样,可以在 CGI 程序与 Web 服务器之间插入一个程序,它接受 Web 服务器的调用,获取相应的运行环境,以某种形式存储起来,并以获得的运行环境来调用指定的 CGI 程序,返回结果。这样 CGI 程序运行环境的创建就简化为使用 Web 浏览器来访问指定的 CGI 程序。

(2)如何能够方便地动态调试 CGI 程序。当在以上的 Web 浏览器上观察到错误的返回结果后,就可以依据存储的运行环境,为 CGI 程序对应的调试器动态的创建 CGI 程序运行环境并调用调试器来对 CGI 程序进行调试。这项工作类似于外壳型调试,但是该系统在调试时所需要的“外壳”,是直接由程序生成的,无需进行复杂的编程。

由于结合后的新技术直接利用各种语言本身的调试工具来进行调试,因此对语言种类和操作系统平台的依赖性很小。

## 5 CGI 调试系统的实现

由 CGI 调试系统的设计思想可知,要实现调试系统,需要解决以下两个问题:

(1)如何在要调试的 CGI 程序与 Web 服务器之间插入程序。可以有两种方案:

直接对 Web 服务器有关 CGI 程序的模块编程,例如在 Apache Web 服务器中就可以对 `cgi_module.c` 编程,进行运行环境的存储工作,这种方式的优点是用户透明,缺点是要依赖于 Web 服务器,必须对每一种 Web 服务器实现需要插入的功能,工作量很大,而且存在 Web 服务器程序本身就不是可扩充的,因此这种方法不容易开发出 Web 服务器无关的调试系统;

编写一个通用的 CGI 程序,利用它来存储 Web 服务器创建的 CGI 程序运行环境,并模拟 Web 服务器来调用需要调试的 CGI 程序。这种方法是几乎完全与平台无关的,由于这个 CGI 程序只使用了几乎所有操作系统都提供的系统调用或者

函数,所以基本上可以使用一套程序源代码在不同系统上编译就可以满足要求了,而且由于绝大多数 Web 服务器都支持 Perl 语言的 CGI 程序,所以可以使用 Perl 语言书写的 CGI 程序来满足要求。

这样,该系统采用第二种方案,在要调试的 CGI 程序与 Web 服务器之间插入一个 CGI 程序,这个 CGI 程序在传递 Web 服务器对 CGI 程序的调用的同时,将单独调试 CGI 程序所需的运行环境存储下来。

(2)如何能够为调试 CGI 程序所需的动态调试器准备环境。现代操作系统,一般都支持大量程序的批量执行,例如 Unix 或 Linux 下的 shell 脚本程序,Windows 系统下的 .bat 批处理程序,利用这些批量执行程序可以先设置好 CGI 程序所需的环境变量,然后调用调试 CGI 程序所需的调试器,进行调试。而且一般调试器在启动的时候都会读取并执行一个初始化命令文件,例如 GDB 的 `.gdbinit` 文件以及 Perl 的 `.perldb` 文件,这样,还可以针对指定的 CGI 程序所需调试器定制初始化命令文件。

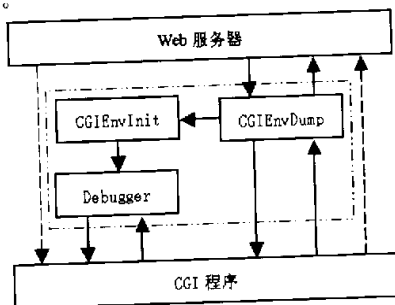


图 2 CGI 调试系统与 Web 服务器、CGI 程序的关系

综上所述,如图 2 所示,点划线内为 CGI 调试系统,两条指向 CGI 程序的实线箭头分别表示两种不同的调用模式:观察模式和调试模式。对 CGI 程序而言,如左边由 Web 服务器发出、指向 CGI 程序的虚线箭头所示,都可以视为是直接被 Web 服务器调用了;而 CGI 程序发出的两条实线箭头,则如右边由 CGI 程序发出、指向 Web 服务器的虚线箭头所示,可以视为是直接输出到 Web 服务器了。由图 2 可知,CGI 调试系统可以分为三部分:

(1)CGIEnvDump,这是一个 CGI 程序,接受 Web 服务器的调用,根据配置将所接收到的 CGI 运行环境(包括所有环境变量以及可能的输入)生成 CGIEnvInit,并模拟这个运行环境来调用真正需要执行的 CGI 程序并返回结果。这个 CGI 程序包括 C 语言和 Perl 两种版本,以利于系统的平台无关性。

(2)CGIEnvInit,由 CGIEnvDump 动态生成,是一个运行脚本,创建调试 CGI 程序所需的环境变量以及可能的输入,然后调用 debugger 来动态调试 CGI 程序。

(3)Debugger,是一个独立的调试器,根据需要,可以用来调试编译型语言如 C/C++ 的 GDB,也可以用来调试 Perl 脚本的 perl 程序(使用 `perl -d` 来进行调试)。

## 6 结束语

CGI 技术大大地促进了 Web 技术的发展,但是 CGI 程序的调试困难问题一直没有得到很好的解决。文章提出的通用

(下转 72 页)

$$=1+(d_p d_q - d_p)/d_q$$

(2)式:

$$d_{\neg p}=1-d_p \quad (\text{由} (*) \text{式})$$

$$=1-(d_p d_q)/d_q$$

$$=1-(d_p d_q)/(1-d_{\neg q}) \quad (\text{由} (*) \text{式})$$

(3)式:

$$d_{\neg p}=1-d_p \quad (\text{由} (*) \text{式})$$

$$=1-(d_q-(d_q-d_p))$$

$$=1-((1-d_{\neg q})-(d_q-d_p)) \quad (\text{由} (*) \text{式})$$

$$=d_{\neg p}+(d_q-d_p)$$

例3 设  $p_{0.6} \rightarrow q_{0.8}$ , 又有  $p_{0.7}, (\neg q)_{0.9}$ , 用正变换率推理规则进行推理。

解

$$(1) p_{0.6} \rightarrow q_{0.8} \quad [\text{前提}]$$

$$(2) p_{0.7} \quad [\text{前提}]$$

$$(3) q_{0.93} \quad [(1), (2), \text{规则}(1)-\text{①}; 0.93 \approx (0.8/0.6) \times 0.7]$$

$$(4) (\neg q)_{0.9} \quad [\text{前提}]$$

$$(5) (\neg p)_{0.925} \quad [(1), (4), \text{规则}(2)-\text{①};$$

$$0.925 = 1 + (0.6 \times 0.9 - 0.6) / 0.8]$$

可以看出,当真度变元中真度  $d'_p$  与  $d_p$  取值相同时,  $d'_q$  也就与  $d_q$  取值相同;  $d_{\neg q}$  与  $d_q$  取值相同时,  $d_{\neg p}$  也就与  $d_p$  取值相同。这时从形式上看,上面的第一组规则和第二组规则也就分别是通常的假言推理和拒取式了。进一步,当这两组规则中的真度变元都只取 0.1 两个值时,这两组规则也就退化为传统二值逻辑中的假言推理和拒取式了。所以,这两组基于真度变换的真度命题推理规则,也就是传统二值逻辑中的假言推理和拒取式那两个一般命题推理规则的推广。

正由于这组推理规则是传统二值逻辑推理规则一种推广,而且是把推理变成了计算,所以就不能苛求它们仍然满足传统二值逻辑推理中推理规则应满足的条件,即:对真度变元的任一组取值,只要推理规则前提(指 $\Rightarrow$ 号之前的部分)部分为真,则结论部分也真。

上述真度命题推理规则区别于传统二值逻辑中的命题推理规则的关键特征是,它把真度蕴涵式  $p \rightarrow q$  作为一种真度变换对待。那么,如果这种变换还是可逆的,则  $p$  和  $q$  就不分前后了,即只要知道其中一个,利用变换  $p \rightarrow q$  或其逆就可求得另一个。又由于  $(d_p d_q - d_p)/d_q$ :

$$d'_q = 1 - d'_p$$

这样,推理规则:

$$(p \rightarrow q) \wedge \neg q \Rightarrow \neg p$$

就可由推理规则:

$$(p \rightarrow q) \wedge q \Rightarrow p \quad (\text{用变换 } p \rightarrow q \text{ 逆表达式求 } d'_p)$$

代替,待求出  $p_{d_p}$  后,再由:

$$d'_{\neg p} = 1 - d'_p$$

求得  $\neg p_{d_{\neg p}}$ 。

既然如此,那么,上面第二组推理规则就可以由下面的推理规则来代替。

(2')采用逆变换的否定后件式真度假言推理

$$\text{①} (p \rightarrow q) \wedge q \Rightarrow p, d'_p = (d_p/d_q) d'_q \quad (\text{基于正变换率})$$

$$\text{②} (p \rightarrow q) \wedge q \Rightarrow p, d'_p = (d_p d_q)/d'_q \quad (\text{基于反变换率})$$

$$\text{③} (p \rightarrow q) \wedge q \Rightarrow p, d'_p = d'_q + (d_p - d_q) \quad (\text{基于变换差})$$

可以看出,这里的变换  $d'_p = (d_p/d_q) d'_q$  正是变换  $d'_q = (d_q/d_p) d'_p$  的逆变换,其余类同。

例4 仍用上例中的  $p_{0.6} \rightarrow q_{0.8}$  和  $\neg q_{0.9}$ , 用(2')中的规则①进行推理。

$$\text{解} (1) p_{0.6} \rightarrow q_{0.8} \quad [\text{前提}]$$

$$(2) \neg q_{0.9} \quad [\text{前提}]$$

$$(3) q_{0.1} \quad [\text{由}(2), (*) \text{式}]$$

$$(4) p_{0.075} \quad [\text{由}(1), (3) \text{和}(2')-\text{①}]$$

$$(5) \neg p_{0.925} \quad [\text{由} (*) \text{式}]$$

可以看出,这与上例所得的结果完全一样。

最后需说明的是,当真度蕴涵式的前件  $p$  是由真度联结词  $\wedge, \vee, \neg$  联结的复合命题时,应先根据相关联结词的定义求出前件  $p$  的总真度,然后在运用上述推理规则进行推理。

## 5 结束语

上面用变换的观点,把蕴涵式  $p \rightarrow q$  看作一种真度变换,从而得到了一套真度命题推理规则,这套规则既构成了一种命题近似推理的新方法,也是传统二值逻辑推理的一种推广。进一步,如果把该文中对蕴涵式的理解和相关的推理思想及方法推广到谓词逻辑,则就可得到一套谓词逻辑近似推理规则,从而得到一种新的模糊推理方法。事实上,在文献[2]中已做到了这一点。所以,作者认为:把蕴涵式  $p \rightarrow q$  作为一种真度变换思想以及所得到的一系列推理规则(包括基于命题逻辑和谓词逻辑)就是模糊推理的理论基础。(收稿日期:2001年6月)

## 参考文献

1. 王国俊. 非经典数理逻辑与近似推理[M]. 科学出版社, 2000
2. 康师友. 程度论. 一种基于程度的信息处理技术[M]. 陕西科学技术出版社, 2000

(上接 36 页)

的、与操作系统平台无关的、与 Web 服务器平台无关的 CGI 调试系统,能够很好地解决这个问题。(收稿日期:2001年7月)

## 参考文献

1. T. Berners-Lee, D. Connolly. Hypertext Markup Language (HTML)- 2.0 [S]. RFC 1866, 1995
2. R. Fielding, J. Gettys, J. Mogul et al. Hypertext Transfer Protocol-HTTP/

1.1 [S]. RFC 2616, 1999

3. Alan O. Freier, Philip Karlton, Paul C. Kocher. The SSL Protocol Version 3.0. draft-freier-ssl-version3-01.txt, 1996
4. Ken A. L. Coar, D. R. T. Robinson. The WWW Common Gateway Interface Version 1.1. http://cgi-spec.golux.com/draft-coar-cgi-v11-03.txt, 1999
5. http://www.big.com/scat/cgiidebug.html
6. http://www.qub.rinet.ru/group/our/vdb.htm