

利用 FastCGI 应用程序 提高 Apache WebServer 性能^①

于中江, 车文刚, 郭剑毅, 马桂芳

(昆明理工大学信息工程与自动化学院, 云南昆明 650051)

摘要 通用网关接口(CGI)已在网络应用中广泛使用.随着 WebServer 负载的增加, CGI 应用程序已不能迅速的响应客户请求.为了弥补高负载下 CGI 应用程序性能瓶颈的不足, 本文采用在广泛使用 Apache WebServer 中加入 FastCGI 模块, 利用 FastCGI 应用程序来提高 Apache WebServer 的性能的方法.

关键词: FastCGI; Apache; WebServer

中图分类号: TP393.02 **文献标识码:** A **文章编号:** 1007-855X(2001)06-035-04

0 前言

若干年来,通用网关接口(CGI)已成为服务器端应用程序开发的事实标准^[1].通过 CGI 应用程序,可以在 Web 上向客户提供动态的、交互式的内容服务.但随着 CGI 应用程序的广泛使用,发现在大量的 CGI 负担下 WebServer 的性能表现不尽人意.分析 CGI 标准的工作流程发现 CGI 应用程序存在瓶颈问题: WebServer 对每个客户请求的 CGI 应用程序启动新的 CGI 进程,然后该进程完成它的任务并结束^[2].在高负载下,启动进程和初始化所需要的时间严重影响 WebServer 响应请求的速度.

为了弥补 CGI 标准的不足,笔者在广泛使用的 Apache WebServer 中加入 FastCGI 模块,把 CGI 应用程序转换为 FastCGI 应用程序,明显地提高了 WebServer 的性能.本文详细分析了 FastCGI 应用程序的工作流程和特点,并以具体的实例来说明如何把 CGI 应用程序转换为 FastCGI 应用程序.

1 FastCGI 模块

1.1 FastCGI 的工作流程

FastCGI 由 Open Market 公司创建,是正在发展的、非专有的、推荐的开放标准, Apache 当前用第三方模块支持 FastCGI^[1].

FastCGI 标准是在 CGI 标准的的基础上发展的,继承了 CGI 标准的大多数特性.与 CGI 标准不同的是:FastCGI 应用程序是常驻内存的,他们被重复用来服务于多个请求;FastCGI 标准支持分布式结构, FastCGI 应用程序能在任何支持 TCP/IP 的主机上使用.当采用多线程方式编写 FastCGI 应用程序时,能进一步的提高 WebServer 的性能. FastCGI 与 WebServer 之间通过 TCP/IP 连接来传送环境信息、标准输入、输出和错误信息.

FastCGI 的工作流程是: WebServer 创建 FastCGI 应用程序进程, FastCGI 应用程序对它自己进行初始化,并等待来自 WebServer 的连接;当客户请求到来时, WebServer 打开与 FastCGI 进程的连接,并通过

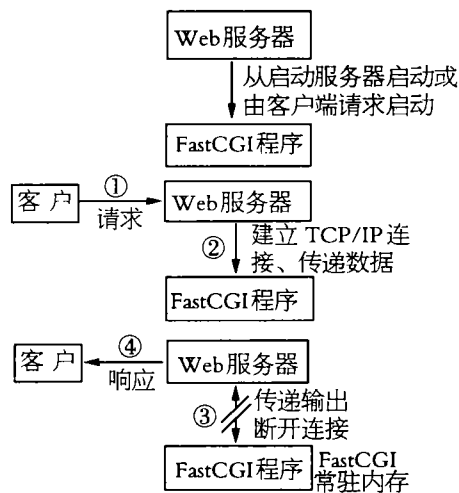


图1 FastCGI 工作流程

① 收稿日期:2001-03-01;

第一作者简介:于中江,男,1975年生,硕士.研究方向:计算机网络.

该连接发送 CGI 环境变量信息和标准输入给 FastCGI 进程;FastCGI 进程通过同样的连接,将标准输出和错误信息发送回服务器;当 FastCGI 进程关闭这个连接时,请求完成,然后 FastCGI 进程等待来自 Web 服务器的另外一次连接.如图 1 所示.

1.2 在 Apache 中加入 FastCGI 模块

Apache 是广泛使用的、高性能的 Web 服务器,在世界上有超过百分之五十以上的 WebServer 使用 Apache^[1]. Apache 服务器由 Apache 内核代码和编译到服务器中的模块组成,为了在 Apache 中加入 FastCGI 模块,需要下载 Apache 内核源代码和 FastCGI 模块源代码,分别在 <http://www.apache.org> 和 <http://www.fastcgi.com> 处获得相应的源代码.当前稳定的 Apache 内核版本为 Apache 1.3.12, FastCGI 模块的版本为 FastCGI 2.2.8.

取得源代码后,在相应的目录(如用户的主目录中)下展开各自的源代码,编译连接后重启 WebServer 即可.以 RedHat 7.0 平台下为例:

```
# tar xzvf apache_1.2.12.tar.gz
# tar xzvf mod_fastcgi.tar.gz
# mv mod_fastcgi2.2.8 apache_1.3.12src/modules/fastcgi
# cd apache_1.3.12
# ./configure --prefix=/usr/local/apache --activate-module=src/modules/fastcgi/libfastcgi.a
# make;make install;
# /usr/local/apache/bin/apachectl restart
```

图 2 在 Apache 中加入 FastCGI

1.3 将 CGI 应用程序转化为 FastCGI 应用程序

在 Web 中,应用的比较多的是外部认证程序,笔者就以外部认证程序为例来说明如何将 CGI 应用程序转换为 FastCGI 应用程序.

需求使用外部数据库服务器(MySQL)来存储认证信息:用户名、口令,利用 FastCGI 应用程序来保持与数据库服务器的持续连接并完成客户端对用户名和密码的认证访问请求查询.运行环境为:操作系统 RedHat 7.0; WebServer 为 Apache 1.3.12; FastCGI 模块采用 mod_fastcgi.c 模块; FastCGI 开发工具集为 devkit-2.2.0; FastCGI 采用 ANSI C 编程语言、编译器采用 GNU 的 gcc.

```
#include "fcgi_stdio.h" /* 开发工具集提供的头文件 */ /* CGI 应用程序中无此行代码 */
*/
#include <stdlib.h> /* C 语言标准库头文件 */
#include "mysql.h" /* MySQL 数据库头文件 */
MYSQL pMysql; /* MySQL 数据库句柄,全局变量 */
void Initialize()
{
    /* 建立与 MySQL 数据库的连接;初始化变量 */
    pMysql = mysql_connect(Mysql, Host, Use, Pwd);
}
void PerformRequest()
{
    /* 认证程序主体:完成一次客户请求(获取环境变量、标准输入,完成验证,将结果返回给 Web-Server) 为提高性能,在这里均采用多线程方式编写 */
}
void VariableCollect() {
```

```
mysql_close(pMysql);    /* 关闭与数据库的连接, 释放初始化时分配的内存空间 */
}
int main() {
    Initialize();        /* 初始化函数 */
    /* FastCGI 开发工具集提供的接口函数. FCGI-Accept 函数的功能是当客户端发送一次请求时,
    该函数返回值大于零(如客户端没有请求、该函数阻塞)、创建一个类似与 CGI 的环境变量表和
    标准输入供应用程序主体使用.
    while (FCGI_Accept() >= 0)
    {
        /* CGI 应用程序中无此行代码 2 */
        PerformRequest(); /* 应用程序主体函数 */
        /* FastCGI 开发工具集提供的接口函数. FCGI_Finish 函数结束一次来自 WebServer 的请
        求, 释放此次 FCGI_Accept 函数中分配的内存. */
        FCGI_Finish();    /* CGI 应用程序中无此行代码 3 */
    }
    /* CGI 应用程序中无此行代码 4 */
    VariableCollect();    /* 变量收集函数 */
    return 0;
}
```

经过编译连接, 获得可执行程序, 存放在相应的 WebServer 目录中, 在客户浏览器中调用一次, 该程序就会常驻内存来响应客户端的请求. 从程序的注释中我们可以看出, CGI 应用程序中只须加入四行代码, 就可转换为 FastCGI 应用程序. 在每天至少有 50000 次认证请求的情况下, 利用 CGI 应用程序时, 按 50000 次计, 就需要派生(fork)进程^[3]、初始化变量、连接断开数据库等重复 50 000 次, 同时也增加了服务器的进程数, 从而使新派生的进程获得 CPU 时间片所等待的时间会很长, 严重影响了 WebServer 的性能和响应速度; 而利用 FastCGI 应用程序时, 只须做一次派生(fork)进程和连接断开数据库的操作, 减少了服务器的进程数, 提高了系统性能^[4].

在将 CGI 应用程序转换为 FastCGI 应用程序时也要注意几个问题: 在 CGI 应用程序中有时不做内存管理、不关闭打开文件等操作, 当 CGI 应用程序执行后退出时交由操作系统恢复内存、关闭文件. 但在 FastCGI 应用程序中, 只要 WebServer 不撤消这样的 FastCGI 应用程序, 它们就一直驻留在内存中, 随着时间的推移不断漏损内存, 并最终导致系统资源匮乏; 如果在循环函数中含有可能干扰主体代码第二次运行的代码时, 要添加重新初始化某些变量、数组等的代码; 在 FastCGI 应用程序中, 常在单个可执行的应用程序中完成多个小的 CGI 应用程序的功能, 以减少需要管理的进程数, 充分利用共享缓存信息.

2 结 论

通过在 Apache WebServer 中加入 FastCGI 模块, 并把 CGI 应用程序转换为 FastCGI 应用程序, 利用 FastCGI 应用程序常驻内存、多线程和分布式结构等特点明显地提高了 Apache WebServer 的性能和响应速度. 笔者在实际的应用开发中大量采用 FastCGI 标准开发应用程序, 取得了很好的效果.

参考文献:

- [1] [美] Mohammed J. Kabir 著, 刘洪勋, 王效东, 陈晓燕等译. Apache 服务器实用大全[M]. 中国水利水电出版社, 1999. 1, 147, 189.
- [2] [美] Jeffrey Dwight, Michal Erwin, Robert Niles 著, 金帆工作室译. CGI 开发使用手册[M]. 北京: 机械工业出版社, 西蒙与舒斯特国际出版公司, 1998. 6, 13.
- [3] [美] W. Richard Stevens 著, 尤晋元等译. UNIX 环境高级编程[M]. 北京: 机械工业出版社, 2000. 142.
- [4] 测试数据来源 <http://www.yymail.com>.

Using FastCGI Applications to Improve the Performance of Apache WebServer

YU Zhong-jiang, CHE Wen-gang, GUO Jian-yi, MA Gui-fang

(The Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650051, China)

Abstract CGI has been widely used in network application. With the increasing of the Apache WebServer's loads, CGI applications can't respond to clients' requirements rapidly. In order to settle the performance bottleneck of CGI applications in high loads, this paper provides a method: adding FastCGI module to widely-used Apache WebServer, to improve the performance of Apache WebServer by FastCGI applications.

Key words: FastCGI; Apache; WebServer

个旧锡矿的矿床成因与找矿新发现

个旧锡矿是驰名中外的特大型锡—多金属矿床,锡储量和半个世纪来的锡产量均占世界的10%以上。解放后,云南地质勘查部门在个旧矿区做过大量的地质勘查工作。许多中外著名的地质学家也在个旧锡矿做过不少研究,获得了丰富的成果,为个旧锡工业的发展作出了突出的贡献。

现在个旧锡矿的主要问题是:随着多年的开采,储量消耗大,矿石资源紧张,急需开展找矿研究。个旧锡矿找矿要取得大的突破,必须有新的找矿思路。

关于个旧锡矿的矿床成因,过去绝大多数人持“花岗岩岩浆期后气化热液成矿”的观点。在云锡公司和云南有色地质局308队的大力支持和帮助下,由昆明理工大学国土资源工程学院秦德先、薛传东、田毓龙、刘春学、洪托、谈树成、刘星、罗祖虞、孟清等人组成的项目组,近年参与云南省省院省校科技合作项目“个旧锡矿深部和外围成矿预测及矿山增储研究”的科研,有不少新的发现,获得了新的成矿观点。大量野外研究,结合部分室内资料分析表明,个旧锡矿区锡多金属矿床的成矿作用经历了印支中期基性火山—沉积成矿、喷流热水沉积成矿和燕山晚期花岗岩改造(叠加)成矿。在空间上,矿床为“两楼一梯”的结构模式,即“下楼”为层状火山—沉积铜(锡)矿体,“上楼”为层状喷流热水沉积锡、铜(铅、银)矿体,“一梯”为花岗岩改造(叠加)成矿,形成了上下贯通的不规则状和脉状锡、铜、铅、银矿体。矿源来自基性火山岩、基底地层,花岗岩浆活动主要提供改造成矿的动力和热力条件,也可能提供了部分矿源。由于多期次、多来源和多种成矿作用叠加成矿,从而形成了现今的特大型矿床。

伴随着矿床成因模式认识的变化,找矿思路、找矿标志和预测准则将有新的改变,找矿空间和找矿远景将会有扩大。这些问题,目前正在进行深入研究。

另外,在上述新认识的基础上,我们对原冶金部西南地质勘探公司物探一分队(1980~1986年)在个旧东区北部266km²范围内按(50~100)m×10m×20m网度所采取的38702件原生晕样品分析数据(每件样品分析Sn、Cu、Pb、Zn、Mn、Bi、As、Cd、Ag、Hg等10个元素),加上本次我们系统采取的220件构造地球化学样品的分析结果,应用数学地质的方法在计算机上进行了重新处理,共圈定各元素不同级别的异常37个。再根据成矿地质条件、异常强度组合特点及前期探矿工程控制情况,确定了8块找矿靶区。通过对其中一个靶区的工程验证,已初见成效。

(昆明理工大学国土资源工程学院 秦德先 供稿)