第25卷第6期 华东交通大学学报 Vol. 25 No. 6
2008年12月 Journal of East China Jiaotong University Dec. ,2008

文章编号:1005-0523(2008)06-0047-04

基于LZW算法的校园网上网日志压缩的实现及应用

谢剑猛,凌仕勇

(华东交通大学 现代教育技术中心,江西 南昌 330013)

摘要:针对校园网网关日志数据量大,存储空间不足的问题,采用LZW无损压缩算法实现自动压缩上网日志,解决了日志的长期备份需求.并基于压缩文件,开发了网络监察工具,快速检索并分析所需要的日志信息.

关 键 词:LZW压缩;无损压缩;校园网管理;校园网日志

中图分类号:TP311 文献标识码:A

本校校园网的上网日志由思科6 509核心交换机通过端口镜像记录到日志服务器上,日均约有10 GB的存储量,日志服务器的磁盘容量为200 GB,只能存储20天的日志量,无法备份足够长的时间以供日后的监察(有关法规要求至少记录60天上网日志).为此,开发了基于LZW算法的日志压缩和解压缩程序,每天自动对生成的日志进行压缩.并在此压缩技术的基础上,开发了网络监察工具,快速检索和导出所需要的日志信息,并做出一些必要的统计.

1 LZW算法原理及实现

1.1 LZW算法特点

LZW(Lempel_Ziv_Welch)是Abraham Lempel、Jacob Ziv与Terry Welch创造的一种通用无损数据压缩算法[1].LZW算法的设计侧重于实现的速度,适用于原始数据串中有大量的子串多次重复出现,且重复的越多,压缩效果越好.校园网上网日志数据串正好存在大量重复子串,采用LZW算法既能达到压缩目的,对日志服务器的性能又不会产生较大影响.

1.2 编码算法

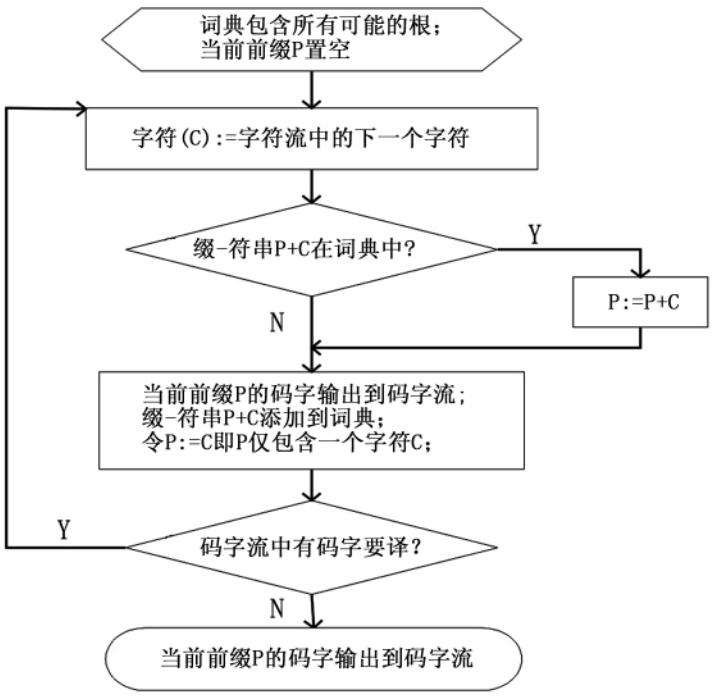

图1 LZW编码算法流程图

LZW编码是围绕称为词典的转换表来完成的.这张转换表用来存放称为前缀的字符序列,并且为每个表项分配一个码字,或者叫做序号.LZW编码器就是通过管理这个词典完成输入与输出之间的转换[2].LZW编码器使用了一种贪婪分析算法.LZW编

收稿日期:2008-10-25

作者简介:谢剑猛(1975-),男,江西崇仁人,讲师,研究方向为校园网络管理、计算机教学和教育技术.

码算法的具体执行流程如图 1 如示.

1.3 译码算法

LZW 译码算法中还用到另外两个术语:(1) 当前码字:指当前正在处理的码字,用 cW 表示,用 string. cW 表示当前缀_符串;(2) 先前码字:指先于当前码字的码字,用 pW 表示,用 string. pW 表示先前缀_符串. LZW 译码算法开始时,译码词典与编码词典相同,它包含所有可能的前缀根(roots). LZW 算法在译码过程中会记住先前码字(pW),从码字流中读当前码字(cW)之后输出当前缀_符串 string. cW,然后把用 string. cW 的第一个字符扩展的先前缀_符串 string. pW 添加到词典中[3]. LZW 译码算法具体执行流程如图 2 如示.

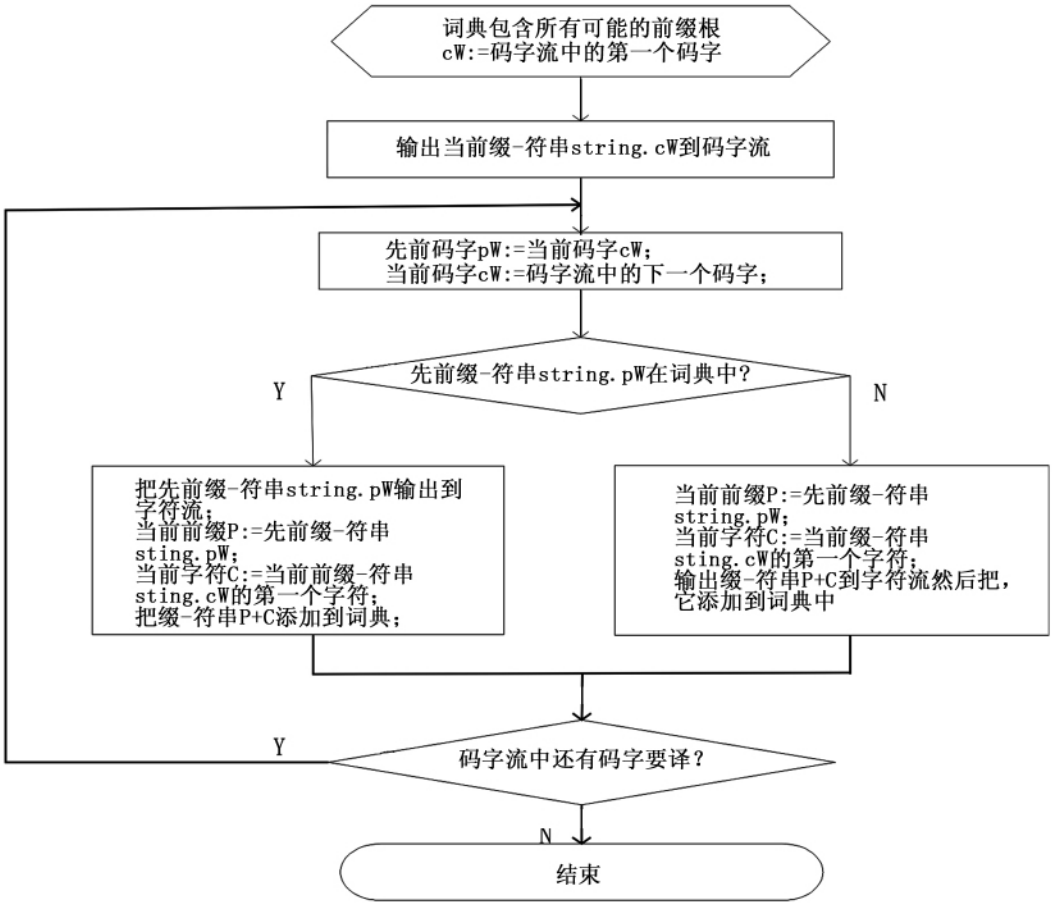

图 2 LZW 译码算法流程图

2 上网日志压缩的实现

2.1 上网日志的生成

日志服务器监控端:配置 IP: X. X. X. X,设置 UDP 端口号为 1514.

思科 6509 三层交换机网络出口网关端口:配置日志的监控开关,时间标签及路由指向.

```
logging on
logging timestamp
logging host inside X. X. X. X 17/1514
```

这样,就将通过网关的所有访问信息记录到日志服务器监控端了,记录的格式如下所示:

<166> Jul 04 2007 13:07:33: %FWSM_6_302013: Built outbound TCP connection 225685875 for faddr 76. 9. 4. 211/80 gaddr 202. 101. 209. 67/2638 laddr 172. 16. 15. 243/2638 \par

<166> Jul 04 2007 13:07:33: %FWSM_6_302014: Teardown TCP connection 225685728 faddr 69. 9. 165. 95/80 gaddr 202. 101. 209. 67/2552 laddr 172. 16. 15. 243/2552 duration 0:00:05 bytes 18

(FIN Timeout) \par

从记录的格式可以看出,每次访问记录以 < 166 > 作为标示,这是 Cisco 的内部标识码; 接下来是以美国方式表达的日期表示,依次为月,日,年,时间; % FWSM_6_302013 为记录标示号,依次递增; 接下来的一段英文描述了连接的类型(TCP,UDP,…) ,连接号(225685875) ,访问的远端地址和端口(faddr 76. 9. 4. 211 /80) ,出口地址和端口(gaddr 202. 101. 209. 67 /2638) ,本地内部地址和端口(172. 16. 15. 243 /2638) ; 如果连接成功则还有持续时间和访问字节数量(duration 0: 00: 05 bytes 18) ; 最后 \par 为结束码. 由此可以看出,此记录格式的文本内容有很大的重复字符如 < 166 > ,TCP connection,faddr,gaddr,laddr,bytes,\par,以及基于阿拉伯数字的 IP,时间,数量,所以用 LZW 压缩能产生很大的压缩比.

2. 2　压缩文件存储格式

对于一个达 10 GB 数据量的文件来说,无法一次性读取整个文件,必须进行分块读取. 为此,程序中分配了 1 MB 的存储空间,每次读取 1 MB 的内容进行压缩,因此,带来的一个问题就是需要一个记录压缩后位置的配置文件. 配置文件可以针对每个被压缩的文件产生一个文件,或者将配置文件置于压缩后的文件头或尾. 考虑到解压缩的方便,采用了将配置文件置于压缩后的文件尾的方法.

压缩文件数据 = 压缩数据段 1 + 压缩数据段 2 + … + 压缩数据段 n + 配置节;

配置节 = 到压缩数据段 1 的总长度 + 空格 + 到压缩数据段 2 的总长度… + 空格 + 到压缩数据段 n 的总长度 + 配置节长度;

配置节长度 = (15 + 1) * n + 15(注 1: 每个段长为 15 个字符 + 1 个空格; 注 2: 压缩数据段 n 指的是原始数据 1 MB 的数据量被压缩后的数据) .

2. 3　压缩部分的代码实现

```
int CCompress::Compress( char*  inputfilename,char*  outputfilename)
{
    ULONGLONG nFilelen = infile. GetLength( ) ; //infile 为 CFile 类,nFilelen 为文件大小
    while( nTotalReadlen < = nFilelen) //nTotalReadlen = 读取的总大小
    {
        if( nFilelen - nTotalReadlen > = inbuflen)  nReadlen = inbuflen;
        else nReadlen = nFilelen - nTotalReadlen; //nReadlen = 需要读的大小
        if( nReadlen = = 0)  break; //读取完毕
        nReadlen = infile. Read( inbuf,nReadlen) ;
        if( nReadlen = = 0)
        {
            ……//失败处理
        }
        nTotalReadlen + = nReadlen;
        nWritelen = outbuflen;
        int ret = compress( outbuf,&nWritelen,inbuf,nReadlen) ; //调用压缩模块
        if( ret!  = 0)
        {
            ……//失败处理
        }
        outfile. Write( outbuf,nWritelen) ; //将压缩后的数据写到压缩文件中
        CzipDlg*  dlg = ( CzipDlg* ) theApp. m _ pMainWnd;
        dlg - > m _ progress. SetPos( ( double) nTotalReadlen* 100. 0 /nFilelen) ; //主界面的进度条显示
        nTotalWriteLen + = nWritelen;
        //配置节的处理
```

```
        tmpcfgbuf. Format( "%015u" ,nTotalWriteLen) ;
        strcat( cfgbuf,tmpcfgbuf. GetBuffer( 0) ) ;
        strcat( cfgbuf," ") ;
        cfglen + = tmpcfgbuf. GetLength( ) + 1;
    }
    tmpcfgbuf. Format( "%015u" ,cfglen) ;
    strcat( cfgbuf,tmpcfgbuf. GetBuffer( 0) ) ;
    outfile. Write( cfgbuf,cfglen + 15) ; //写入配置节到压缩文件中
    outfile. Flush( ) ;
    …//存储空间的清理
    return 0;
}
```

2.4 上网日志的快速监察

以前对网络信息的监察需在 linux 模拟环境 cygwin 下利用 linux 的查找命令进行查找,查找一个信息需非常长的时间(有时要一个小时以上) . 考虑到计算机花费的时间主要在 I/O 处理上,如果编制搜索软件在 10 GB 以上的文件中搜索某个信息并统计也需花费很长的时间. 现在利用压缩后的文件,直接在不到 800 MB 的文件中查找,速度可加快 10 倍,特别是对于多个关键字信息的查找更能大大节省时间. 例如,搜索 IP 为 172. 16. 29. 198 在 2007 年 12 月 11 日的访问情况,搜索时间约 7 分钟,解压后得到上网日志如表 1:

表 1 日志查询结果

Time	faddr	gaddr	laddr	duration
Dec 11 2007 14: 15: 17	218. 3. 53. 180/80	218. 65. 102. 178/30209	172. 16. 29. 198/2	0: 00: 04 bytes 9968 (TCP FINs)
Dec 11 2007 14: 20: 28	221. 207. 249. 210/8080	218. 65. 102. 178/49323	172. 16. 29. 198/2	0: 00: 10 bytes 6232 (TCP Reset_I)
Dec 11 2007 14: 21: 36	58. 51. 99. 78/80	218. 65. 102. 178/52251	172. 16. 29. 198/2909	0: 00: 24 bytes 11303 (TCP Reset_I)

3 结论

基于 LZW 算法实现的上网日志的压缩和监察程序,自开发投入使用以来,节约了大量的存储空间,省去了日志服务器磁盘投入的费用,提高了日志监察效率. 校园网每天上网日志的存储量从原来的 10 G 到现在的不到 800 M,日志服务器可存储 8 个月的日志. 压缩一个 10 GB 大小的日志文件约需 22 分钟,解压则约需 12 分钟,压缩比约为 12 ~ 15: 1.

参考文献:

[1] Welch T A. A technique for high_performance data compression [J] . IEEE Computer,1984,17(6) : 8 - 18.

[2] Nelson M,贾起东译. 数据压缩技术原理与范例 [M] . 北京科学出版社,1995.

[3] 王 平. LZW 无损压缩算法的实现与研究 [J] . 计算机工程,2002,28(7) : 98 - 99.

(下转第 62 页)

参考文献:

[1] 王 腾,姚丹霖. Online Judge 系统的设计开发 [J]. 计算机应用与软件. 2006,22(12):129 - 130,137.

[2] 黄国平. 用 AJAX 技术改进在线考试系统 [J]. 南通职业大学学报. 2006,16(3):113 - 116.

[3] Elliotte Rusty Harold. Processing XML with Java [M]. Beijing: Publishing House of Electronics Industry,2003.

[4] 张金涛. 基于 Linux 的 Apache + JSP + Oracle [M]. 北京: 清华大学出版社,2002.

[5] 谢小乐. J2EE 经典实例祥解 [M]. 北京: 人民邮电出版社,2003.

[6] 孙卫琴. Tomcat 与 Java Web 开发技术详解 [M]. 北京: 电子工业出版社,2006.

A New Scoring Method of Online Judging System

TAN Bin_Wen, WANG Gen_sheng, ZHOU Juan

(School of Information Engineering, East China Jiaotong University, Nanchang 330013)

Abstract: The technology of machine scoring in the multiple_choice and fills_up topic is already mature, but there are many defects in the programming topic. At first, the existing deficiencies in the machine scoring are analyzed in detail, a solution is proposed, and then the judging theory is exceptionally analyzed. Finally, the JSP technology and redirect technology are used. With its feasibility, this solution has a good prospect.

Key words: online judging system; machine scoring; JSP; redirect

(责任编辑: 周尚超)

(上接第 50 页)

Application and Realization of Campus Network Logs Compression Based on LZW Algorithm

XIE Jian_meng, LING Shi_yong

(Center of Modern Education and Technology, East China Jiaotong University, Nanchang 330013, China)

Abstract: Aiming at the problem of large amount log data and inadequate storage space in campus network gateway, the paper develops a LZW lossless compression algorithm to automatically compress Campus network log, which resolves the need of long_term backup. Based on the compressed files, it develops a network monitoring tool, which can quickly retrieve and analyze the required information.

Key words: LZW compression, lossless compression, campus network management, campus network logs

(责任编辑: 王建华)