(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 105893358 A
(43)申请公布日 2016.08.24

(21)申请号 201410464824.4

(22)申请日 2014.09.12

(71)申请人 江苏国贸酝领智能科技股份有限公司

地址 215000 江苏省苏州市工业园区唯亭镇唯文路5号

(72)发明人 陈宏庆 顾永青

(51)Int.Cl.

G06F 17/30(2006.01)

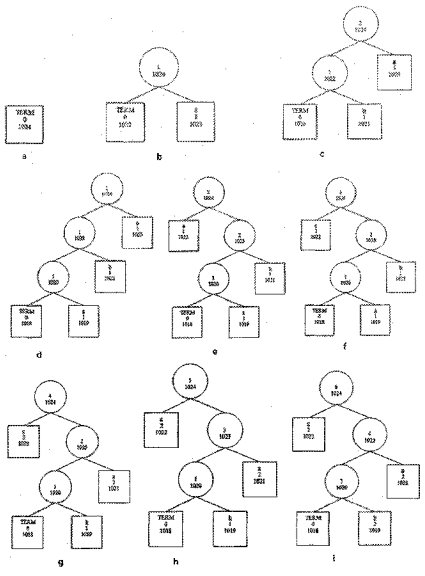
权利要求书1页 说明书5页 附图1页

(54)发明名称

一种文件的实时压缩方法

(57)摘要

本方法应用于实时文件传输压缩领域。要对一个通过网络传输的文件进行压缩，传统的做法是等文件传输完后保存到本地存储空间中，再对文件进行扫描，通过统计文件中字符出现的概率分布情况，制定合适的编码策略，然后第二次扫描文件，对文件的每个字符生成压缩编码，写入到压缩文件中，最后删除原文件。网络传输和扫描，占用了大量的时间，而且在压缩的过程中原始文件和压缩文件同时存在，会占用本地存储空间。本发明是为了创造一种针对网络传输的文件进行快速压缩的方法，当文件传输完毕后，压缩文件就已经生成，同时在压缩的过程中不会占用其他的存储空间。



CN 105893358 A

1. 本发明在文件传输过程中使用可变树对传输的文件数据进行编码，因此要求保护使用可变树的方式对网络传输中的文件进行编码。

2. 本发明在文件传输过程中压缩时采用双通道 TCP 连接控制，一个通道传输控制指令，一个通道传输文件数据，因此要求保护在文件传输过程中使用双通道 TCP 连接实现边传输边压缩的设计方法。

一种文件的实时压缩方法

一、技术领域

[0001] 本发明应用于网络数据传输文件压缩与解压缩领域，主要解决在网络传输文件数据过程中，边传输边压缩的问题。

二、背景技术

[0002] 要对一个通过网络传输的文件进行压缩，传统的做法是等文件传输完后，保存到本地存储空间中，再对文件进行扫描，通过统计文件中字符出现的概率分布情况，制定合适的编码策略，然后第二次扫描文件，对文件的每个字符应用压缩编码，写入到压缩文件中，最后删除原始文件。网络传输和两次扫描，占用了大量的时间，而且在压缩的过程中原始文件和压缩文件同时存在，如果文件较大，会占用大量存储空间。本发明主要是为了创造一种针对网络传输的文件进行快速压缩、压缩过程中不占用其他存储空间的方法。

三、发明内容

[0003] 一个发送端向一个接收端发送文件，接收端一边接收文件一边对文件进行压缩，等到文件发送完毕后，压缩文件也同时生成，压缩的过程中不会产生临时文件，不会占用其他的存储空间。

[0004] 1. 发送端和接收端通过两个 TCP 连接进行通信，一个连接负责发送指令，一个连接负责发送数据。

[0005] 2. 发送端通过指令发送连接发出文件发送请求，请求的消息格式如下：

[0006] type，fid，file-size，filename-length，filename。

[0007] type :8bit，无符号数，消息的类型，发送请求消息为 1。

[0008] fid :64bit，无符号数，被传输文件的唯一标识符，指令传输 TCP 连接成功后，fid 被初始化为一个 0 ～ 2^64-1 之间的随机数，每传输完一个文件该数值就增加一，到达 2^64-1 后回到 0，以此规律循环。

[0009] file-size :64bit，无符号数，文件大小。

[0010] filename-length :16bit，无符号数，文件名长度。

[0011] filename :长度不限制，字符串，文件名内容。

[0012] 3. 接收端进行请求的回复，正常的回复（准备接收）消息格式如下：

[0013] type，fid

[0014] type :8bit，无符号数，消息的类型，准备接收为 2。

[0015] fid :64bit，无符号数，要被接收的文件的唯一标识符。

[0016] 出错时的回复消息格式如下：

[0017] type，fid，code

[0018] type :8bit，无符号数，消息的类型，出错消息为 3。

[0019] fid :64bit，无符号数，要被接收的文件的唯一标识符。

[0020] code :8bit，无符号数，出错原因代码。具体如下表：

[0021]

code	含义

1	存储空间不足
2	正在传输文件
3	没有写入权限
4	其他原因

[0022] 4. 接收端在本地存储中创建压缩文件，文件名原文件名加上“.gmf”，文件开头的内容如下：

[0023] file-size，filename-length，filename

[0024] file-size：64bit，无符号数，文件大小。

[0025] filename-length：16bit，无符号数，文件名长度。

[0026] filename：不限制长度，字符串，文件名。

[0027] 5. 接收端初始化编码树，该树只有一个空叶节点，符号为 TERM，权值始终为 0，编号为 1024。

[0028] 6. 发送端通过数据发送连接开始发送数据，接收端进行数据接收。接收端每读进一个字符，检查该字符是否存在于编码树中：

[0029] 1) 如果不存在，则对该字符进行编码，从树的根节点开始一直到 TERM 字符，经过左孩子就编码为 0，经过右孩子编码为 1，直到到达该字符，最后加上该字符本身，产生的编码写入压缩文件中。然后生成一棵子树，用这棵子树代替原来的 TERM 结点，该子树的父节点符号为空，权值为 1(TERM 的权值 0 加上新的加入的符号节点权值 1)，编号为 TERM 原来的编号，其右分支节点为刚读入的字符，该结点符号为该字符，权值为 1，编号为现在父节点编号减去 1，左分支节点为一个新的空叶结点 TERM，编号为现在父节点编号减去 2。因为加入了新的节点，所以各个需要调整各个节点的权值，根据节点编号由小到大的顺序，在修改权值之前，将当前节点与块中具有相同权值的编号最大的节点进行交换（只交换字符和权值，不交换编号），并使后者的父节点成为新的当前节点，直到遇到根节点为止。

[0030] 2) 如果存在，则对该字符进行编码，从树的根节点开始一直到该字符，经过左孩子就编码为 0，经过右孩子编码为 1，直到到达该字符，产生的编码写入压缩文件中。然后调整各个节点的权值，根据节点编号由小到大的顺序，在修改该节点的权值之前，将当前节点与块中具有相同权值的编号最大的节点进行交换，并使后者的父节点成为新的当前节点，直到遇到根节点为止。

[0031] 7. 发送端发送完数据后，在指令 TCP 连接上发出文件发送完毕消息，消息格式如下：

[0032] type，fid

[0033] type：8bit，无符号数，消息的类型，发送完毕消息为 4。

[0034] fid：64bit，无符号数，已发送完毕的文件的唯一标识符。

[0035] 8. 接收端接收到文件发送完毕消息后，正常情况下返回成功接收消息，消息格式如下：

[0036] type，fid

[0037] type：8bit，无符号数，消息的类型，准备接收为 5。

[0038] fid：64bit，无符号数，确认已经被接收的文件的唯一标识符。

[0039] 出错时的回复消息格式如下：

[0040] type，fid，code

[0041] type :8bit,无符号数,消息的类型,出错消息为 6。

[0042] fid :64bit,无符号数,接收出错的文件的唯一标识符。

[0043] code :8bit,无符号数,出错原因代码。具体如下表 :

[0044]

code	含义
1	存储空间不足
2	正在传输文件
3	接收的文件数据不全
4	其他原因

[0045] 下面着重对压缩编码的生成方法和文件解压进行说明,例如原文件名为 a.txt,二进制文件的字节内容是 abccab,压缩步骤如下 :

[0046] 1. 接收端初始化编码树,该树只有一个空叶节点,符号为 TERM,权值始 0,编号为 1024,说明书附图 1 中的 a 是执行该步骤后的编码树。

[0047] 2. 读取第一个字节为 a,因为树中没有字符 a,所以 a 的编码是从树根节点定位到 TERM 节点的编码加上字符 a 本身,因为只有一个 TERM 节点,所以 a 的编码就是 a,写入压缩编码 a。然后生成一棵子树,用这棵子树代替原来的 TERM 结点,该子树的父节点符号为空,编号为 TERM 原来的编号 1024,其右分支节点为字符为 a,权值为 1,编号为现在父节点编号减去 1 即 1023,左分支节点为一个新的空叶结点 TERM,编号为现在父节点编号减去 2 即 1022。虽然根节点的权值进行了增加,但是该节点编号是最大的,所以不做任何交换动作,最后修改节点权值为 1(TERM 的权值 0 加上节点 a 的权值 1),说明书附图 1 中的 b 是执行该步骤后的编码树。

[0048] 3. 读取第二个字节 b,树中没有字符 b,从根节点到 TERM 节点的路径编码加上 b 字符本身就是字符 b 的编码,写入压缩编码 0b。使用包含新 TERM 节点和字符 b 替代旧的 TERM 节点,并根节点的权值加 1。说明书附图 1 中的 c 是执行该步骤后的编码树。

[0049] 4. 读取第三个字节 c,树中没有字符 c,写入 c 的编码为 00c。使用包含新 TERM 节点和字符 b 替代旧的 TERM 节点,说明书附图 1 中的 d 是执行该步骤后的编码树,此时需要对编号为 1022 的节点的权值进行加 1 操作,但是此时与编号为 1022 的节点权值相同的块有 1023,1021,1019,其中 1023 最大,所以需要将 1022 与 1023 进行交换,对权值进行加 1 处理,然后将 1023 的父节点作为当前节点,因为是根节点,所以直接将根节点的权值加 1 即可,说明书附图 1 中的 e 是执行 该步骤后的编码树。

[0050] 5. 读取第 4 个字节 c,树中有字符 c,写入 c 的编码为 101,将要对 1019 节点的权值加 1,此时与 1019 权值相同的节点有 1021 和 1022,最大编号是 1022,所以需要交换 1022 和 1019,交换后对编号为 1022 的节点权值进行加 1 处理,然后将 1022 的父节点作为当前节点,因为是根节点,所以直接将根节点的权值加 1 即可,说明书附图 1 中的 f 是执行该步骤后的编码树。

[0051] 6. 读取第 5 个字节 a,树中有字符 a,写入 a 的编码为 101,将要对 1019 节点的权值加 1,此时与 1019 权值相同的节点有 1021,而且 1021 大于 1019,所以需要交换 1021 和 1019,交换后对编号为 1021 的节点权值进行加 1 处理,说明书附图 1 中的 g 是执行该步骤后的编码树,然后需要调整 1021 的父节点 1023 的权值,因为与 1023 权值 2 相同的块有 1022,但是 1022 小于 1023,所以不进行交换,将 1023 的权值加 1 变成 3,下面需要调整 1024 的权

值，因为是根节点，所以直接将权值加 1，变成 5，说明书附图 1 中的 h 是执行该步骤后的编码树。

[0052] 7. 读取第 6 个字节 b，树中有字符 b，写入 b 的编码为 101，将要对 1019 的节点权值加 1，此时没有与 1019 权值 1 相同的块，不需要进行交换，直接将 1019 节点的权值加 1 变成 2，然后将要对 1020 节点的权值加 1，此时没有与 1020 节点权值 1 相同的块，不需要进行交换，直接将 1020 节点的权值加 1 变成 2，然后将对 1023 节点的权值加 1，此时没有与 1023 权值一样的块，不需要进行交换，直接将 1023 节点的权值加 1 变成 4，然后将要对 1024 节点的权值加 1，因为是根节点，所以直接加 1 变成 6。说明书附图 1 中的 i 是执行该步骤后的编码树。

[0053] 8. 最终 abccab 的压缩编码是 :a0b00c101101101，文件名为 a. txt. gmf。原来存储这个文件需要使用 48bit，通过压缩后只需要 36bit。

[0054] 对压缩文件进行解压缩，期望生成的原文件二进制内容为 :abccab，解压文件的步骤如下 :

[0055] 1. 读取压缩文件的原始文件大小为 6 个字节，原始文件名为 a. txt，通过原始文件名创建原始空文件。

[0056] 2. 初始化编码树，该树只有一个空叶节点，符号为 TERM，权值始 0，编号为 1024，说明书附图 1 中的 a 是执行该步骤后的编码树。

[0057] 3. 读取压缩编码中的第一个字节，为字符 a，写入原始文件，然后将字符 a 加入到树中，方法与压缩时的相同，说明书附图 1 中的 b 是执行该步骤后的编码树。

[0058] 4。读取压缩编码中的一位 0，到达 TERM 节点，然后从压缩编码中读取一个字节，为字符 b，将字符 b 写入原始文件，将字符 b 加入到树中，说明书附图 1 中的 c 是执行该步骤后的编码树。

[0059] 5. 读取压缩编码中的一位 0，到达 1022，再读取压缩编码中的一位 0，到达 TERM，然后从压缩编码中读取一个字节，为字符 c，将字符 c 写入原始文件，将字符 c 加入到树中，说明书附图 1 中的 e 是执行该步骤后的编码树。

[0060] 6. 读取压缩编码中的位 101，最终到达 1019 节点，该节点字符为 c，将 c 写入原始文件，将 1019 节点的权值加 1，说明书附图 1 中的 f 是执行该步骤后的编码树。

[0061] 7. 读取压缩编码中的位 101，最终到达 1019 节点，该节点字符为 a，将 a 写入原始文件，将 1019 节点的权值加 1，说明书附图 1 中的 h 是执行该步骤后的编码树。

[0062] 8. 读取压缩编码中的位 101，最终到达 1019 节点，该节点字符为 b，将 b 写入原始文件，将 1019 节点的权值加 1，说明书附图 1 中的 i 是执行该步骤后的编码树。

[0063] 9. 最终完成解压缩操作，原文件二进制内容为 abccab，与预期的一致。

图 1 是文件压缩和解压缩过程中出现的各种编码树变换图。

图 1 中 a 是初始的只包含一个空节点的编码树 ;

图 1 中 b 是加入字符 a 节点后的编码树 ;

图 1 中 c 是加入字符 b 节点后的编码树 ;

图 1 中 d 是加入字符 c 节点后的编码树 ;

图 1 中 e 是加入字符 c 节点后权值进位后的编码树 ;

图 1 中 f 是再次加入字符 c 节点后的编码树 ;

图 1 中 g 是再次加入字符 a 节点后的编码树；

图 1 中 h 是再次加入字符 a 节点后权值进位后的编码树；

图 1 中 i 是再次加入字符 b 节点后的编码树。

四、具体实施方式

[0064] 1. 配置 TCP/IP 网络，并且网络上两端能互通。

[0065] 2. 在网络的一端启动发送端，在网络的另一端启动接收端。

[0066] 3. 发送端发送文件，接收端能产生压缩文件。

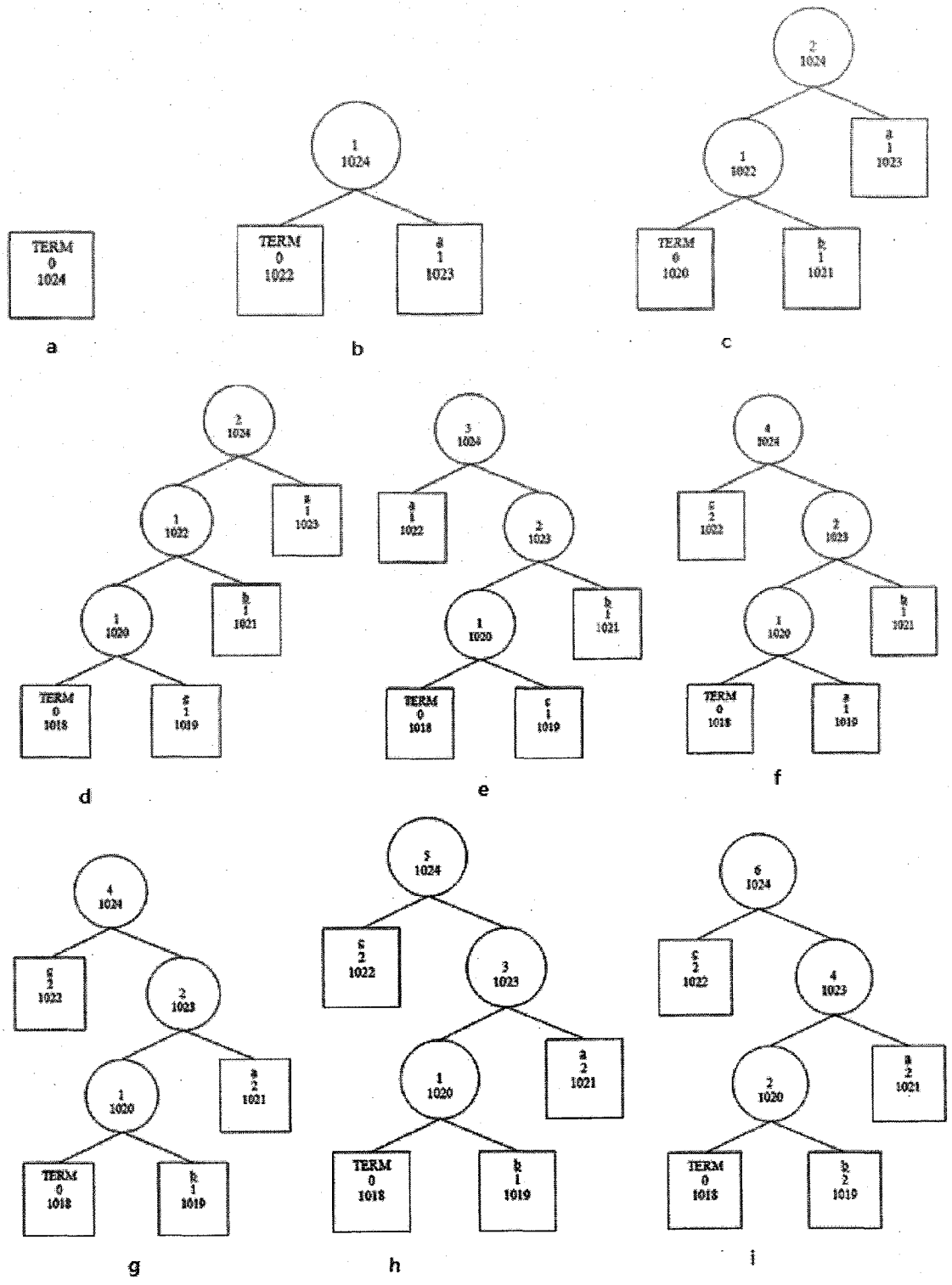

TERM
0
1024
a
1
1024
TERM
0
1022
a
1
1023
b
2
1024
1
1022
a
1
1023
TERM
0
1020
b
1
1021
c
2
1024
1
1022
a
1
1023
1
1020
b
1
1021
TERM
0
1018
c
1
1019
d
3
1024
a
1
1022
2
1023
1
1020
b
1
1021
TERM
0
1018
c
1
1019
e
4
1024
c
2
1022
2
1023
1
1020
b
1
1021
TERM
0
1018
a
1
1019
f
4
1024
c
2
1022
2
1023
1
1020
a
2
1021
TERM
0
1018
b
1
1019
g
5
1024
c
2
1022
3
1023
1
1020
a
2
1021
TERM
0
1018
b
1
1019
h
6
1024
c
2
1022
4
1023
2
1020
a
2
1021
TERM
0
1018
b
2
1019
i

图 1