

证书号第 1506277 号



发明专利证书

P1200070

发明名称：一种大型 XML 文件的网络传输方法

发明人：孙伟丰;李浩;王恒;程仁波;刘钰;王文军;郑程光;赖铮
罗正海;徐邵稀;胡奎;关健;赵伟

专利号：ZL 2012 1 0013186.5

专利申请日：2012 年 01 月 16 日

专利权人：北大方正集团有限公司;方正信息产业控股有限公司
上海方正数字出版技术有限公司

授权公告日：2014 年 10 月 29 日

本发明经过本局依照中华人民共和国专利法进行审查，决定授予专利权，颁发本证书并在专利登记簿上予以登记。专利权自授权公告之日起生效。

本专利的专利权期限为二十年，自申请日起算。专利权人应当依照专利法及其实施细则规定缴纳年费。本专利的年费应当在每年 01 月 16 日前缴纳。未按照规定缴纳年费的，专利权自应当缴纳年费期满之日起终止。

专利证书记载专利权登记时的法律状况。专利权的转移、质押、无效、终止、恢复和专利权人的姓名或名称、国籍、地址变更等事项记载在专利登记簿上。



局长
申长雨

申长雨





(12) 发明专利

(10) 授权公告号 CN 102571966 B

(45) 授权公告日 2014. 10. 29

(21) 申请号 201210013186. 5

(22) 申请日 2012. 01. 16

(73) 专利权人 北大方正集团有限公司

地址 100871 北京市海淀区成府路 298 号方正大厦 5 层

专利权人 方正信息产业控股有限公司
上海方正数字出版技术有限公司

(72) 发明人 孙伟丰 李浩 王恒 程仁波
刘钰 王文军 郑程光 赖铮
罗正海 徐邵稀 胡奎 关键
赵伟

(74) 专利代理机构 北京英赛嘉华知识产权代理有限公司
11204

代理人 王达佐

(51) Int. Cl.

H04L 29/08 (2006. 01)

(56) 对比文件

CN 1901549 A, 2007. 01. 24, 说明书第 2 页第 3 行 - 第 15 行, 第 5 页第 30 行 - 第 12 页第 1 行 .

CN 101364235 A, 2009. 02. 11, 全文 .

CN 101847998 A, 2010. 09. 29, 全文 .

US 2011/0222596 A1, 2011. 09. 15, 全文 .

CN 1635492 A, 2005. 07. 06, 说明书第 3 页第 2 行 - 第 10 页第 26 行 .

审查员 孙慧珠

权利要求书 2 页 说明书 8 页 附图 1 页

(54) 发明名称

一种大型 XML 文件的网络传输方法

(57) 摘要

本发明公开了一种大型 XML 文件的网络传输方法, 包括 :S1 读取 XML 文档。S2 对 XML 文档的所有节点标签进行生成索引值的预处理。S3 删除 XML 文档中的节点的冗余右标签。S4 根据当前网络负载状况, 动态调整 XML 文档分片文件的大小, 完成 XML 文档的压缩处理和分片传输, 进一步包括 :S41 设置分片的初始大小为 Initial_Size ;S42 取出若干个数据作为一个数据包进行压缩处理, 并进行网络传输, 其中该数据包的大小为 Initial_Size ;S43 根据所述数据包的传输时间, 判断当前网络负载状况, 并得出最大传输单元的值 ;S44 根据当前网络负载状况和最大传输单元的值以及数据的传输时间重新确定分片大小为 New_Size ;S45 重新取大小为 New_Size 的数据包进行压缩处理, 网络传输 ;S46 重复步骤 S43 至步骤 S45。



1. 一种大型 XML 文件的网络传输方法,其特征在于,包括以下步骤:

S1:读取 XML 文档;

S2:对 XML 文档的所有节点标签进行生成索引值的预处理,包括:

S21:寻找出 XML 文档中所有节点标签信息,并将所述节点标签信息保存至一张节点数据表中,并对所述节点数据表中的每一项赋予唯一编号作为该项在表中的索引值;

S22:在完成所述节点数据表的建立后,使用节点所对应的索引号对所述 XML 文档进行相应的处理,对于所述节点在文档中的出现的位置使用其对应的节点索引号进行代替;

S3:删除 XML 文档中的节点的冗余右标签;

S4:根据当前网络负载状况,动态调整 XML 文档分片文件的大小,完成 XML 文档的压缩处理和分片传输,包括:

S41:设置分片的初始大小为 Initial_Size;

S42:取出若干个数据作为一个数据包进行压缩处理,并进行网络传输;其中所述数据包的大小为 Initial_Size;

S43:根据所述数据包的传输时间,判断当前网络负载状况,并得到网络自动生成的最大传输单元的值;

S44:根据当前网络负载状况和最大传输单元的值以及数据的传输时间重新确定分片大小为 New_Size;

S45:重新取大小为 New_Size 的数据包进行压缩处理,并进行网络传输;

S46:重复步骤 S43 至步骤 S45,直至 XML 文档全部传输完成。

2. 根据权利要求 1 所述的方法,其特征在于,所述步骤 S43 中,根据所述数据包的传输时间,判断当前网络负载状况,具体包括:

A431:步骤 S42 中数据包传输后,计算得到所述数据包的数据传输时间 TT,其中,所述数据传输时间 TT 为所述数据包发送时间与响应时间之间的差值;

A432:设定所述数据传输时间 TT 的初始上限值 TT_MAX,用以描述数据在该网络中的传输所需的最大时间,即在网络异常拥堵的情况下,数据由发送端传输到接收端所需的最大传输时间;

A433:判断网络负载状况:

若所述数据传输时间 TT 在区间 $(0, TT_MAX]$ 内,则认为该网络负载较为正常;若所述数据传输时间 TT 在区间 $(TT_MAX, +\infty)$ 内则表明该段网络状况较差。

3. 根据权利要求 1 所述的方法,其特征在于,所述步骤 S44 进一步包括:

S441:在时间 T 内随机抽取 $T_1、T_2、\dots、T_{N-1}、T_N、\dots、T_m$ 个数据传输时间,这里 T 指的是采样时间;

S442:计算所述 m 个数据传输时间的统计算术平均值,即统计平均传输时间:

$$AT_m = m / (1/T_1 + 1/T_2 + \dots + 1/T_m) \quad (4.1)$$

S443:设定一个阈值,用以描述所述 AT_m 的最大值;

S444:比较 $T_{N-1}、T_N$ 两段时间内数据报文的统计平均传输时间 AT_N 和 AT_{N-1} 的差值;检查所述 AT_N 和所述 AT_{N-1} 的关系是否满足如公式 (4.2),如满足,则可以进一步按照公式 (4.3) 来增加发送分片的大小,否则执行 S445;

$$1 - ((AT_N - AT_{N-1}) / AT_{N-1}) \leq \text{阈值} \quad (4.2)$$

$$\text{New_Size} = \text{Initial_Size} * (1+a)^N \quad (4.3)$$

其中:Initial_Size为初始分片大小,New_Size为调整后的数据分片大小,a为数据分片的增幅大小,N为增幅指数,其初始值可以设置为1,每次执行所述步骤S444时候将N设置为 $N = N \pm 1$;

S445:将a设置为-a,N设置为 $N-1 (N \geq 1)$,并用公式(4.3)来重新调整数据分片的大小。

4. 根据权利要求1所述的方法,其特征在于,步骤S42中取出的数据同时包括该数据在整个文档中的节点数据表。

一种大型 XML 文件的网络传输方法

技术领域

[0001] 本发明涉及 XML 文件的传输,特别涉及一种大型 XML 文件的网络传输方法。

背景技术

[0002] XML(可扩展标记文件)是一种标记语言,其具有如下优点:(1)自描述性;(2)支持文档内容的验证;(3)允许不同领域的标记语言的存在;(4)支持高级搜集,故而,XML 格式的文件在现今的互联网应用中被大量使用。由于现今 Internet 网络带宽容量的限制,使得系统在进行大型 XML 文件传输时,对 XML 进行有效的压缩及传输机制的设计显得尤为重要。

[0003] 大型 XML 文件传输现有如下的两种方式:(1)文件结构感知;(2)非文件结构感知两种方式;而在文件结构感知方式中又分为:基于 schema 方式和非 schema 方式两种。

[0004] 考虑到 XML 文件是以文本方式进行存储,在非结构感知方式下现行系统多采用传统的文本压缩方式进行 XML 文件的压缩,例如:采用 bzip2, gzip 等方式。虽然该种方式在对文本文件压缩时具有较大的压缩率,但同时此种方式缺点也较为明显:(1)压缩需占用较多计算资源(CPU,内存,I/O等);(2)需要在发送端和接收端,同时具有相同的压缩/解压缩算法;(3)无法对压缩后的文件进行在线查询。

[0005] 结构感知方式的压缩方式,其旨在利用对 XML 文档结构的感知来实现压缩的一种压缩方式,相对 XML 文档进行一般文本压缩处理的方式来说,该方法进一步提供了压缩率,但其依赖于其后端的压缩/解压算法和相应的 XML 文档的格式或 XML 文档的 Schema,而没有对于 XML 文档中的一些冗余信息进行相应的处理。

[0006] 基于对上述问题的讨论我们提出一种新的基于压缩和分批传输的大型 XML 文件网络传输方法,该方法具有较高的压缩率,较低的压缩/解压资源消耗,以及较快的传输速率等优点从而以较低的资源需求来完成海量数据的传输。

[0007] 在进行大型 XML 文件网络传输时,首要考虑的问题是网络带宽,其次是要考虑尽可能地减少所需要传输文档的大小,最后要考虑的是传输过程所造成文档误差的检查和修复能力。在此,我们主要关注前面两点,即如何在有限的网络带宽下,提高系统的吞吐量。

[0008] 现有对于大型 XML 文档进行网络传输时所采用的方法有上述提及的基于 XML 文档结构感知和非结构感知的两种主要方法,他们的重点均侧重于对于 XML 文档的压缩处理,而对于压缩后的文件的传输问题则没有过多的讨论。

[0009] 下面就主要讨论一下上述各个方案。首先,对于非结构感知,其主要侧重于对于 XML 文档进行相关的压缩,其所采用的方法是将 XML 文档作为文本文件进行压缩/解压处理,从而达到减少相应所需传输文件大小的目的。如在进行 XML 文档传输之前对相关文档使用文本压缩算法进行压缩,在接收端使用相应的解压算法对所接收的压缩文件进行解压并还原出源文件。该方案主要使用基于 LZ77 和 Huffman 编码,Burrows-Wheeler 转换和自适应统计数据压缩技术的 gzip, bzip2, PPM 等算法对 XML 文档进行相关压缩。

[0010] 结构感知的压缩算法主要利用了 XML 文档的高度结构化的特点,对 XML 文档进行

结构分析,将 XML 文档的结构和数据分开,将 XML 文档结构使用树型数据结构来表示,然后将各个节点的数据保存至该树的各个叶子节点上的同质的容器中。XMi11 作为独立模式的结构感知的压缩算法现在已经广泛的应用在大型 X M L 文档压缩中。例如对于如下的 X M L 文档 :

[0011]

```

<books>
  <book id="book1">
    <title> Discovery in deep ocean.</title>
    <price>12.5</price>
    <author>Brad.Pitt</author>
  </book>

```

[0012]

```

  <book id="book2">
    <title>anonymous</title>
    <price>21.4</price>
    <authro>annoy</author>
  </book>
</books>

```

[0013] XMi11 首先根据文档中的所有元素节点构造相应的元素容器,下表表示上述的 XML 文档所对应的元素容器及属性容器和数据容器的情况。

[0014] 元素表

[0015]

1	/books
2	/books/book
...	..
5	/books/book/ author

[0016] 属性表

[0017]

1	/books/book/ @id
---	---------------------

[0018] /books/book/title

[0019]

Discovery in deep ocean.
Anonymous

[0020] 对于基于 Schema 结构感知的压缩算法,算法主要根据系统中 XML 文档的 schema 对相应的 XML 文件进行编码和压缩。XML Schema 作为 XML 文档结构的描述,已经得到越来越广泛的应用。在 XML 所满足的 XML Schema 中包括所有 XML 中元素类型的定义。故而在数据处理之前,可以先分析 XML Schema,获取 XML 数据流中出现的结构事件序列,进而确定元素事件序列的编码,完成 XML 数据流的压缩。例如:XAUST XML 压缩算法将 DTD 的模式信息转换成一组确定性有限自动机 (DFA),每个 DFA 对应于 DTD 中的一个元素。然后将每个状态的转换都用一个元素标记,与此相关的转换行为是调用 DFA 模拟器,并将该行为用于标记该转换的元素。而后,XAUST 将同一元素的所有数据都归入单一容器,然后使用算 order-4 压缩器的单个模型以递增方式压缩该容器。使用 DTD 模式信息,XAUST 能够跟踪文档的结构,并能够准确预测预期符号。只要预测到符号是独特的,就不需要对其进行编码,因为解码器从 DTD 生成相同的模型,从而可以生成独特的预期符号。

[0021] 对于非结构感知的文档压缩由于其所采用的压缩算法自身的缺陷使得该种方案在实际的应用中也存在着诸多的缺点。例如对于上述的 gzip 压缩算法,LZ77 算法使用对匹配数据的引用来取代数据部分。匹配数据已经通过编码器和解码器传递。这使得算法在应用的过程中首先在系统中传送相应的匹配数据,在大型 XML 文档中可能会有大量的匹配数据,从而进一步加剧了网络的负载。而 Huffman 编码使用一种特定方法选择每个符号的表示,使用当时短位字符串中的最常见的字符,而不使用不常见的源符号。而对于 bzip2 所使用的 Burrows-Wheeler 转换对于源文档中经常出现的字符串具有较高的压缩率,但对于连续情况较为松散的字符串其压缩率不甚理想,总体性能上较低。对于 PPM 算法其是基于上下文建模和预测的一种技术,算法在进行建模和预测时必然会消耗大量的计算资源,虽然其 PPM 具有简单,压缩率高等特点,但其也是一种相对昂贵的压缩算法。虽然从理论上讲依赖模式的压缩方法可以实现比非依赖模式的方式具有较高的压缩率,但其也存在着相应的缺点:首先,XML 文档的 schema 文件并不是总可以获得,一旦无法获得 XML 文档的 schema,那么算法就无法获得较高的压缩率。

[0022] 最大传输单元 (Maximum Transmission Unit, MTU) 是指一种通信协议的某一层上面所能通过的最大数据包大小 (以字节为单位)。最大传输单元这个参数通常与通信接口有关 (网络接口卡、串口等)。

[0023] 从上述分析可以看出,现有方案均将其重点放在发送和接收端对相应 XML 文档的压缩处理上,而忽略其对于压缩文档的网络传输问题。而此问题对于大型文档在网络中传输至关重要,由于在大文件传输的过程中会出现对网络资源使用的竞争问题。同时由于存在 MTU 的限制,在网络协议层会将大数据报文进行分割传输,同时网络负载的状况也会影响到网络的吞吐量,因为如果网络负载严重,网络会出现拥塞,数据包重传,虽然协议会对这些情况进行处理,但是如果在应用层减少该种情况发生的概率也会从另一个方面提高网络的吞吐量从而减少网络的拥塞。

[0024] 因此,针对上述现有方案所存在的缺点,需要提出一个基于压缩技术的大型 XML 文件传输方法。

发明内容

[0025] 针对上述方案所存在的不足,我们提出一种大型 XML 文件的网络传输方法,且在对 XML 文档进行压缩的基础上,在进行网络传输时,根据当前网络的负载状况对 XML 文档进行分片处理,分片文件的大小会随着网络负载状况的变化进行动态调整,从而保证网络负载在合理的范围内。

[0026] 本发明的技术方案如下:

[0027] 一种大型 XML 文件的网络传输方法,包括以下步骤:

[0028] S1:读取 XML 文档;

[0029] S2:对 XML 文档的所有节点标签进行生成索引值的预处理;

[0030] S3:删除 XML 文档中的节点的冗余右标签;

[0031] S4:根据当前网络负载状况,动态调整 XML 文档分片文件的大小,完成 XML 文档的压缩处理和分片传输。

[0032] 较佳地,所述步骤 S2 进一步包括:

[0033] S21:寻找出 XML 文档中所有节点标签信息,并将所述节点标签信息保存至一张节点数据表中,并对所述节点数据表中的每一项赋予唯一编号作为该项在表中的索引值;

[0034] S22:在完成所述节点数据表的建立后,使用节点所对应的索引号对所述 XML 文档进行相应的处理,对于所述节点在文档中的出现的位置使用其对应的节点索引号进行代替。

[0035] 较佳地,所述步骤 S4 进一步包括:

[0036] S41:设置分片的初始大小为 Initial_Size;

[0037] S42:取出若干个数据作为一个数据包进行压缩处理,并进行网络传输;其中所述数据包的大小为 Initial_Size;

[0038] S43:根据数据包的传输时间,判断当前网络负载状况,并得到网络自动生成的最大传输单元的值;

[0039] S44:根据当前网络负载状况和最大传输单元的值以及数据的传输时间重新确定分片大小为 New_Size;

[0040] S45:重新取大小为 New_Size 的数据包进行压缩处理,并进行网络传输;

[0041] S46:重复步骤 S43 至步骤 S45,直至 XML 文档全部传输完成。

[0042] 较佳地,所述步骤 S43 中,根据所述数据包的传输时间,判断当前网络负载状况,具体包括:

[0043] A431:步骤 S42 中数据包传输后,计算得到所述数据包的数据传输时间 TT,其中,所述数据传输时间 TT 为所述数据包发送时间与响应时间之间的差值;

[0044] A432:设定所述数据传输时间 TT 的初始上限值 TT_MAX,用以描述数据在该网络中的传输所需的最大时间,即在网络异常拥堵的情况下,数据由发送端传输到接收端所需的最大传输时间;

[0045] A433:判断网络负载状况:

[0046] 若所述数据传输时间 TT 在区间 (0, TT_MAX] 内,则认为该网络负载较为正常;若所述数据传输时间 TT 在区间 (TT_MAX, +∞) 内则表明该段网络状况较差。

[0047] 较佳地,所述步骤 S44 进一步包括:

[0048] S441:在时间 T 内随机抽取 T_1 、 T_2 、 \dots 、 T_{N-1} 、 T_N 、 \dots 、 T_m 个数据传输时间,这里的 T 指的是采样时间;

[0049] S442:计算所述 m 个数据传输时间的统计算术平均值,即统计平均传输时间:

$$[0050] \quad AT_m = m / (1/T_1 + 1/T_2 + \dots + 1/T_m) \quad (4.1)$$

[0051] S443:设定一个阈值,用以描述所述 AT_m 的最大值;

[0052] S444:比较 T_{N-1} 、 T_N 两段时间内数据报文的统计平均传输时间 AT_N 和 AT_{N-1} 的差值;检查所述 AT_N 和所述 AT_{N-1} 的关系是否满足如公式 (4.2),如满足,则可以进一步按照公式 (4.3) 来增加发送分片的大小,否则执行 S445;

$$[0053] \quad 1 - ((AT_N - AT_{N-1}) / AT_{N-1}) \leq \text{阈值} \quad (4.2)$$

$$[0054] \quad \text{New_Size} = \text{Initial_Size} * (1+a)^N \quad (4.3)$$

[0055] 其中:Initial_Size 为初始分片大小,New_Size 为调整后的数据分片大小,a 为数据分片的增幅大小,N 为增幅指数,其初始值可以设置为 1,每次执行所述算法时候将 N 设置为 $N = N \pm 1$;

[0056] S445:将 a 设置为 -a,N 设置为 $N-1$ ($N \geq 1$),并用公式 (4.3) 来重新调整数据分片的大小。

[0057] 较佳地,步骤 S42 中取出的数据同时包括该数据在整个文档中的节点数据表。

[0058] 与现有技术相比,本发明的有益效果如下:

[0059] 第一,本发明实现了根据网络状况对 XML 文档实行实时的动态分片,保证最大程度利用网络带宽,减少传输次数,提高传输效率。

[0060] 第二,本发明实现了对 XML 文档极大程度的压缩,而将 XML 文档压缩后再进行网络传输可以减少传输时间,提高传输效率。

[0061] 第三,本发明可提供在线查询功能;且因为本发明中每个压缩分片中系统会提供该分片中数据所对应的节点数据表,用来标示该数据在整个 XML 文档中所处于的节点关系,故即使在没有将文档传输完整的情况下也可以借助节点数据表为用户提供查询功能。

附图说明

[0062] 图 1 为本发明具体实施例一种大型 XML 文件的网络传输方法的流程图;

[0063] 图 2 为本发明具体实施例步骤 S4 的流程图。

具体实施方式

[0064] 下方结合附图和具体实施例对本发明做进一步的描述。

[0065] 如图 1,一种大型 XML 文件的网络传输方法,包括以下步骤:

[0066] S1:读取 XML 文档;

[0067] S2:对 XML 文档的所有节点标签进行生成索引值的预处理;

[0068] S3:删除 XML 文档中的节点的冗余右标签;

[0069] S4:根据当前网络负载状况,动态调整 XML 文档分片文件的大小,完成 XML 文档的压缩处理和分片传输。

[0070] 其中,步骤 S2 进一步包括:

[0071] S21 :寻找出 XML 文档中所有节点标签信息,例如 :节点的名称、节点的属性值等信息,并将这些信息保存至一张节点数据表中,并对表中的每一项赋予唯一编号作为该项在表中的索引值 ;

[0072] S22 :在完成节点数据表的建立后,使用节点所对应的索引号对 XML 文档进行相应的处理,对于该节点在文档中的出现的位置使用其对应的节点索引号进行代替。通过此项处理后,可以预期大大的减少文档中的冗余信息。例如 :对于出现的一对标签 <Name>Brad. Pitt</Name>,可将其简化成 <1>Brad. Pitt</1>。

[0073] 在完成上述步骤 S1-S2 后,可以发现现有的文档已被大大的简化,在此基础上我们对简化的 XML 进行进一步的处理 :删除所有节点的右标签,因为我们可以发现在删除该标签后,其 XML 文档并没有减少相关结构的信息,节点的右标签属于冗余信息,其标识了某节点的结束位置,而该位置我们可以从下一节点的开始标签获得其上一节点的右标签。

[0074] 虽然经过上述两步处理后 XML 文档大小会得到相应的减少,但对于大型 XML 文档其大小仍然是一个非常大的数值,如果将该 XML 文件压缩成为一个单独的压缩文件传输,会造出以下几个问题 :首先在传输大文件时候,会给网络带来非常大的负载压力 ;(2) 对于单个文件来说,如果在传输过程中发生异常,导致接收端接收的文件为错误文件,使得接收端无法正确的解压,使得相应的发送端的压缩工作和发送工作变为无用功,消耗了发送端的计算资源和公共的网络带宽资源 ;(3) 无法进行有效的在线查询功能。由于单个压缩文件巨大使得在线查询需要消耗大量的计算资源。为解决上述三个问题,我们在对处理后的大型 XML 文档进行分片压缩和传输。为了尽可能的提高网络传输能力和 TCP/IP 协议的利用率,在系统发送时会自动调整分片大小。我们知道网络的 MTU 大小决定了是否对传输数据进行分片处理,如果过大的数据会导致在传输过程中进行较多的数据分割,从而造成一定的资源消耗 (因为每个分片会增加 40 字节的消耗,20 字节 TCP 报文头 +20 字节 IP 报文头),故而在对 XML 文档进行分片压缩时,每个压缩分片的大小会根据当前网络负载状况和网络中 MTU 的情况进行动态调整以保证网络。同时,为了提供在线查询功能,在每个压缩分片中系统会提供该分片中数据所对应的节点数据表,用来标示该数据在整个 XML 文档中所处于的节点关系,这样在进行在线查询是可提供准确的查询。再者,对应分片压缩传输,如在传输过程中某一分片初始出现异常,只需将该分片重传即可,而无需重传整个压缩文件。

[0075] 故,步骤 S4 是非常关键的步骤。

[0076] 如图 2,步骤 S4 进一步包括 :

[0077] S41 :设置分片的初始大小为 Initial_Size ;

[0078] S42 :取出若干个数据作为一个数据包进行压缩处理,并进行网络传输 ;其中该数据包的大小为 Initial_Size ;步骤 S42 中取出的数据同时包括该数据在整个文档中的节点数据表。

[0079] S43 :根据所述数据包的传输时间,判断当前网络负载状况,并得到网络自动生成的最大传输单元 MTU 的值 ;

[0080] S44 :根据当前网络负载状况和最大传输单元的值以及数据的传输时间重新确定分片大小为 New_Size ;

[0081] S45 :重新取大小为 New_Size 的数据包进行压缩处理,并进行网络传输 ;

[0082] S46 :重复步骤 S43 至步骤 S45,直至 XML 文档全部传输完成。

[0083] 步骤 S43 中,根据所述数据包的传输时间,判断当前网络负载状况,具体包括:

[0084] A431 :步骤 S42 中数据包传输后,计算得到该数据包的数据传输时间 TT,其中,数据传输时间 TT 为该数据包发送时间与响应时间之间的差值;

[0085] A432 :设定数据传输时间 TT 的初始上限值 TT_MAX,用以描述数据在该网络中的传输所需的最大时间,即在网络异常拥堵的情况下,数据由发送端传输到接收端所需的最大传输时间;

[0086] A433 :判断网络负载状况:

[0087] 若所述数据传输时间 TT 在区间 $(0, TT_MAX]$ 内,则认为该网络负载较为正常;若所述数据传输时间 TT 在区间 $(TT_MAX, +\infty)$ 内则表明该段网络状况较差。

[0088] 其中,所述步骤 S44 进一步包括:

[0089] S441 :在时间 T 内随机抽取 $T_1, T_2, \dots, T_{N-1}, T_N, \dots, T_m$ 个数据传输时间,这里的 T 指的是采样时间;例如在 60Mins 内,我们收集 m 个数据传输时间,就是上面的 $T_1, T_2, \dots, T_{N-1}, T_N, \dots, T_m$ 。

[0090] S442 :计算该 m 个数据传输时间的统计算术平均值,即统计平均传输时间:

$$[0091] \quad AT_m = m / (1/T_1 + 1/T_2 + \dots + 1/T_m) \quad (4.1)$$

[0092] S443 :设定一个阈值,用以描述所述 AT_m 的最大值;这里的阈值一般是按照经验值来选值。例如可以按照网络空闲时间的 3-5 倍来设置。

[0093] S444 :比较 T_N, T_{N-1} 两段时间内数据报文的统计平均传输时间 AT_N 和 AT_{N-1} 的差值;

[0094] 检查 AT_N 和 AT_{N-1} 的关系是否满足如公式 (4.2),如满足,则可以进一步按照公式 (4.3) 来增加发送分片的大小,否则执行 S445;

$$[0095] \quad 1 - ((AT_N - AT_{N-1}) / AT_{N-1}) \leq \text{阈值} \quad (4.2)$$

$$[0096] \quad \text{New_Size} = \text{Initial_Size} * (1+a)^N \quad (4.3)$$

[0097] 其中:Initial_Size 为初始分片大小,初始为 1K Bytes。New_Size 为调整后的数据分片大小, a 为数据分片的增幅大小, N 为增幅指数,其初始值可以设置为 1,每次执行该算法时将 N 设置为 $N = N \pm 1$ 。

[0098] a 的值也是由经验来设置大小,比如 $a = 0.3$ 或者 0.8 。比如说我们取 $a = 0.3$ 则当 $N = 1$ 的时候,新的分片大小则为: $\text{Initial_Size} * (1+0.3)^1$, N 为增幅指数, N 可以按照每次加 / 减一的方式来,这样可以在很快的时间内增加或者降低一个初始状态。

[0099] S445 :将 a 设置为 -a, N 设置为 $N-1 (N \geq 1)$, 并公式 (4.3) 来重新调整数据分片的大小。

[0100] 同时,为了提供在线查询功能,在每个压缩分片中系统会提供该分片中数据所对应的节点数据表,用来标示该数据在整个 XML 文档中所处于的节点关系,这样在进行在线查询是可提供准确的查询。

[0101] 例如,虽然有些系统提供分片传输,但是由于在分片中不提供该分片的关系信息,使得我们在数据传输过程中无法查询该分片中的信息,如:存在分片信息 English<1>100<2>200 这样的分片,我们无法在线查询,因为我们无法知道分片中的 English, 1, 2 以及 100, 200 所代表的含义,基于上述原因我们在分片中加入该分片的节点关系信息,这样就使得系统支持在线查询。例如,这样的分片 0:name, 1:price, 2:quantity

|<0>English<1>100<2>200 就可以实现在线查询的支持。再者,对应分片压缩传输,如在传输过程中某一分片初始出现异常,只需将该分片重传即可,而无需重传整个压缩文件。

[0102] 本发明优选实施例只是用于帮助阐述本发明。优选实施例并没有详尽叙述所有的细节,也不限制该发明仅为所述的具体实施方式。显然,根据本说明书的内容,可作很多的修改和变化。本说明书选取并具体描述这些实施例,是为了更好地解释本发明的原理和实际应用,从而使所属技术领域技术人员能很好地利用本发明。本发明仅受权利要求书及其全部范围和等效物的限制。

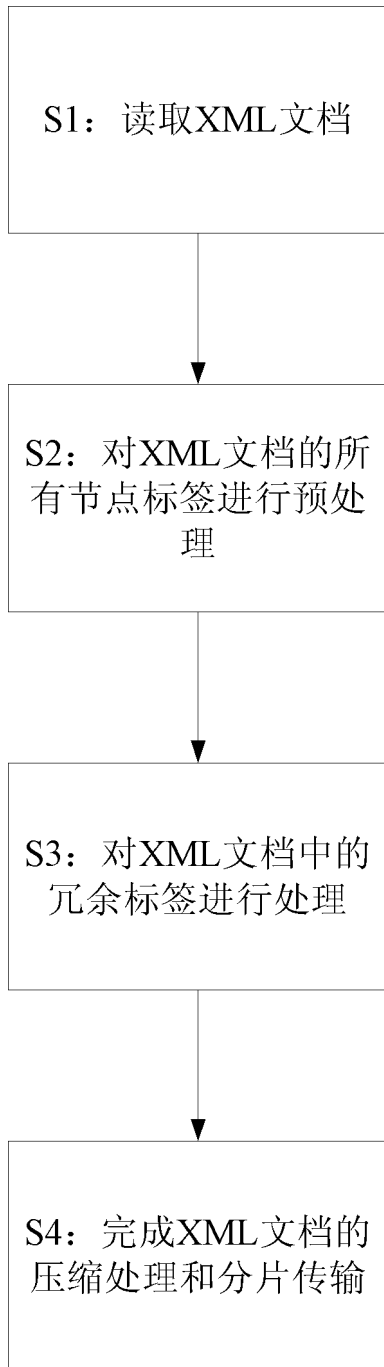


图 1

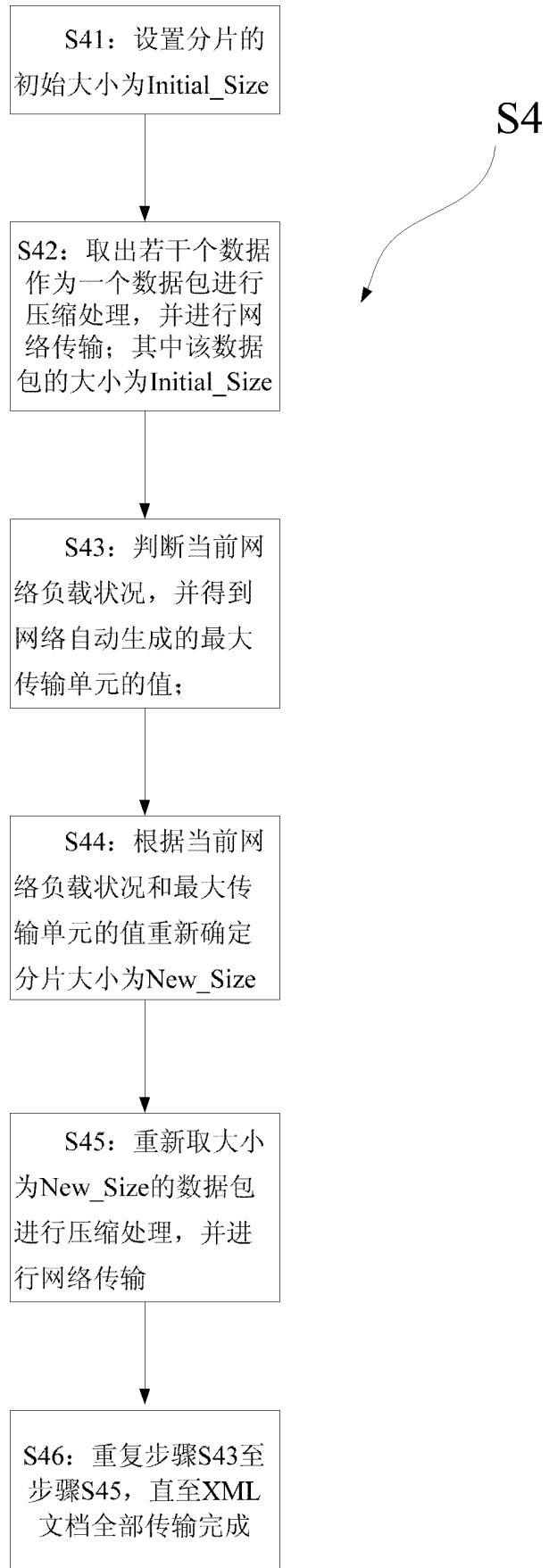


图 2