



审查员 汪婷静

(22) 申请日 2020.09.17

(65) 同一申请的已公布的文献号

申请公布号 CN 112104658 A

(43) 申请公布日 2020.12.18

(73) 专利权人 山东方寸微电子科技有限公司

地址 250000 山东省济南市高新区舜华路

2000号舜泰广场9号北楼803-1室

专利权人 青岛方寸微电子科技有限公司

(72)发明人 张洪柳 杨洪杰 刘超 司浩

(74) 专利代理机构 济南圣达知识产权代理有限公司

公司 37221

专利代理师 张庆蹇

(51) Int.Cl.

H04L 69/04 (2022.01)

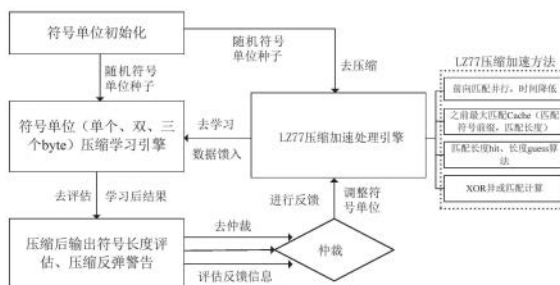
权利要求书2页 说明书7页 附图4页

(54) 发明名称

一种报文压缩方法及系统

(57) 摘要

本发明属于报文压缩领域,提供了一种报文压缩方法及系统。其中,报文压缩方法包括初始化待压缩报文的符号单位并分别传送至学习引擎和加速压缩引擎;利用加速压缩引擎对于待压缩报文进行压缩处理,并将压缩后的报文数据馈入至学习引擎进行学习,以确定符号单位是否最优;将学习引擎学习后的结果馈入到评估引擎,以确定是否具有最小的总长度;将符号单位评估信息馈入到仲裁器进行仲裁选择,由仲裁器决定压缩引擎的最优压缩符号单位。其能够平衡电路资源代价与时间代价,在提高压缩速度的同时,节省电路资源。



1. 一种报文压缩方法,其特征在于,包括:

初始化待压缩报文的符号单位并分别传送至学习引擎和加速压缩引擎;

利用加速压缩引擎对于待压缩报文进行压缩处理,并将压缩后的报文数据馈入至学习引擎进行学习,以确定符号单位是否最优;在加速压缩引擎中,采用多路S基指针并行前向匹配的方法加快待压缩报文的压缩;其中,S基指针是sub-string的起始位置;

将学习引擎学习后的结果馈入到评估引擎,以确定是否具有最小的总长度;

将符号单位评估信息馈入到仲裁器进行仲裁选择,由仲裁器决定压缩引擎的最优压缩符号单位。

2. 如权利要求1所述的报文压缩方法,其特征在于,在多路S基指针并行前向匹配的过程中,对所有S基指针得到的匹配子串长度进行比较,只留下最长的匹配长度。

3. 如权利要求2所述的报文压缩方法,其特征在于,如果有相同的最大匹配长度,该最大匹配长度距离固定的需要被压缩码流窗口的左边界偏移位置对应最远的S基指针位置。

4. 如权利要求1所述的报文压缩方法,其特征在于,在加速压缩引擎中,基于cache的最大匹配长度预测方法来加快前向匹配速度。

5. 如权利要求4所述的报文压缩方法,其特征在于,基于cache的最大匹配长度预测方法来加快前向匹配速度的过程为:

把最大匹配的子串的前缀作为键值K,进行哈希计算,得到hash(K)并作为cache的索引,cache索引对应的寻址后数据内容为最大匹配长度,不断进行填充,直到cache满;其中,cache包括时间cache和空间cache;

等到cache存有一定内容,进行cache的hit操作,如果hit,就得到当下正在进行的子串的预测最大匹配长度;

如果对于同样的hash(K)索引,匹配长度大于cache里的既存长度,则进行cache内容,即最大长度更新;如果匹配的位置offset超过预设值,则在时间cache里更新,否则,在空间cache里更新。

6. 如权利要求1所述的报文压缩方法,其特征在于,在仲裁器中,多路S基指针位置guess淘汰候选方法,以候选者进行异或匹配计算,输出多路S基指针并行搜索结果。

7. 一种报文压缩系统,其特征在于,包括学习引擎、加速压缩引擎、评估引擎和仲裁器;

学习引擎和加速压缩引擎均用于接收初始化的待压缩报文的符号单位;

加速压缩引擎,用于对于待压缩报文进行压缩处理,并将压缩后的报文数据馈入至学习引擎;在加速压缩引擎中,采用多路S基指针并行前向匹配的方法加快待压缩报文的压缩;其中,S基指针是sub-string的起始位置;

学习引擎,用于对接收的报文数据进行学习,以确定符号单位是否最优;还将学习后的结果馈入到评估引擎;

评估引擎,用于对学习后的结果进行评估,以确定是否具有最小的总长度;还将符号单位评估信息馈入到仲裁器;

仲裁器,其用于仲裁选择,决定压缩引擎的最优压缩符号单位。

8. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于,该程序被处理器执行时实现如权利要求1-6中任一项所述的报文压缩方法中的步骤。

9. 一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计

算机程序,其特征在于,所述处理器执行所述程序时实现如权利要求1-6中任一项所述的报文压缩方法中的步骤。

一种报文压缩方法及系统

技术领域

[0001] 本发明属于报文压缩领域,尤其涉及一种报文压缩方法及系统。

背景技术

[0002] 本部分的陈述仅仅是提供了与本发明相关的背景技术信息,不必然构成在先技术。

[0003] LZ77压缩作为一种经典的报文压缩处理方法,其将look-ahead window里面的substring,从search window里面去寻找匹配substring,并且匹配到最大长度匹配substring,如图1所示。发明人发现,传统的报文压缩处理方法借助于计算机软件实现,没有充分考虑压缩处理涉及到的数字逻辑硬件运算时间开销(cycles/单次匹配),也没有给出具体的硬件逻辑实现方法,不能达到电路资源代价与时间代价之间的平衡。

发明内容

[0004] 为了解决上述问题,本发明提供一种报文压缩方法及系统,其能够平衡电路资源代价与时间代价,在提高压缩速度的同时,节省电路资源。

[0005] 为了实现上述目的,本发明采用如下技术方案:

[0006] 本发明的第一个方面提供一种报文压缩方法。

[0007] 在一个或多个实施例中,一种报文压缩方法,包括:

[0008] 初始化待压缩报文的符号单位并分别传送至学习引擎和加速压缩引擎;

[0009] 利用加速压缩引擎对于待压缩报文进行压缩处理,并将压缩后的报文数据馈入至学习引擎进行学习,以确定符号单位是否最优;

[0010] 将学习引擎学习后的结果馈入到评估引擎,以确定是否具有最小的总长度;

[0011] 将符号单位评估信息馈入到仲裁器进行仲裁选择,由仲裁器决定压缩引擎的最优压缩符号单位。

[0012] 本发明的第二个方面提供一种报文压缩系统。

[0013] 在一个或多个实施例中,一种报文压缩系统,包括学习引擎、加速压缩引擎、评估引擎和仲裁器;

[0014] 学习引擎和加速压缩引擎均用于接收初始化的待压缩报文的符号单位;

[0015] 加速压缩引擎,用于对于待压缩报文进行压缩处理,并将压缩后的报文数据馈入至学习引擎;

[0016] 学习引擎,用于对接收的报文数据进行学习,以确定符号单位是否最优;还将学习后的结果馈入到评估引擎;

[0017] 评估引擎,用于对学习后的结果进行评估,以确定是否具有最小的总长度;还将符号单位评估信息馈入到仲裁器;

[0018] 仲裁器,其用于仲裁选择,决定压缩引擎的最优压缩符号单位。

[0019] 本发明的第三个方面提供一种计算机可读存储介质。

[0020] 一种计算机可读存储介质,其上存储有计算机程序,该程序被处理器执行时实现如上述所述的报文压缩方法中的步骤。

[0021] 本发明的第四个方面提供一种计算机设备。

[0022] 一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,所述处理器执行所述程序时实现如上述所述的报文压缩方法中的步骤。

[0023] 与现有技术相比,本发明的有益效果是:

[0024] 本实施例利用加速压缩引擎对于待压缩报文进行压缩处理,并将压缩后的报文数据馈入至学习引擎进行学习,以确定符号单位是否最优,实现了基于预先学习进行自适应性调整最佳匹配级别;将学习引擎学习后的结果馈入到评估引擎,以确定是否具有最小的总长度;将符号单位评估信息馈入到仲裁器进行仲裁选择,由仲裁器决定压缩引擎的最优压缩符号单位,提高了报文压缩匹配的灵活性以及压缩处理速度,平衡了电路资源代价与时间代价。

附图说明

[0025] 构成本发明的一部分的说明书附图用来提供对本发明的进一步理解,本发明的示意性实施例及其说明用于解释本发明,并不构成对本发明的不当限定。

[0026] 图1是传统LZ77压缩算法流程;

[0027] 图2是本发明实施例的报文压缩流程;

[0028] 图3是本发明实施例的并行的多路需要遍历的多路S基指针位置;

[0029] 图4是本发明实施例的基于固定的S基指针与固定的L指针以及前移临时S指针、临时L指针,进行前向搜寻、匹配最大长度的子串;

[0030] 图5是本发明实施例的基于cache的最大匹配长度预测;

[0031] 图6是本发明实施例的多路S基指针位置guess淘汰候选方法。

具体实施方式

[0032] 下面结合附图与实施例对本发明作进一步说明。

[0033] 应该指出,以下详细说明都是例示性的,旨在对本发明提供进一步的说明。除非另有指明,本文使用的所有技术和科学术语具有与本发明所属技术领域的普通技术人员通常理解的含义。

[0034] 需要注意的是,这里所使用的术语仅是为了描述具体实施方式,而非意图限制根据本发明的示例性实施方式。如在这里所使用的,除非上下文另外明确指出,否则单数形式也意图包括复数形式,此外,还应当理解的是,当在本说明书中使用术语“包含”和/或“包括”时,其指明存在特征、步骤、操作、器件、组件和/或它们的组合。

[0035] 实施例一

[0036] 参照图2,本实施例的报文压缩方法,包括:

[0037] 步骤1:初始化待压缩报文的符号单位并分别传送至学习引擎和加速压缩引擎。

[0038] 其中,符号单位可以是单个byte、双byte或者三个byte。

[0039] 步骤2:利用加速压缩引擎对于待压缩报文进行压缩处理,并将压缩后的报文数据馈入至学习引擎进行学习,以确定符号单位是否最优。

[0040] 在具体实施中,在加速压缩引擎中,采用多路S基指针前向匹配并行的方法加快待压缩报文的压缩;其中,S基指针是sub-string的起始位置。

[0041] sub-string匹配窗的移动多路并行,实现了硬件资源消耗与时钟周期消耗的平衡。

[0042] 具体地,如图3所示,当L指针位置确定后(即look-ahead窗的左边界),需要前向、后向移动S指针,这里后向移动S指针可以并行执行,不存在数据依赖性、存在的结构依赖性可以充分利用多路并行匹配单元解决;

[0043] 如图3所示,L指针暂时固定,以充分移动S指针,S基指针是sub-string的起始位置,需要从L指针的后面一个符号单位的位置遍历到左边界,而左边界可能是search窗口的左边界,也可能是报文的左边界;

[0044] 在图3中,L指针是固定的,只有所有S指针都搜寻完毕,L指针根据搜寻的最大匹配长度,前移,前移的offset是最匹配长度+1;

[0045] S指针的左边界可能是search窗口的左边界,也可能是报文的左边界,需要进行左边界决定;

[0046] S指针可以沿着下降指针,并行去搜索,匹配计算。

[0047] 如图4所示,从每个S基指针向前搜寻最大的匹配sub-string,需要前移临时S指针与临时L指针,以得到最大长度的匹配子串;

[0048] 多路S基指针的前向搜寻,可以并行,即多路S基指针同时操作,或者分成组,一组一组的利用硬件匹配逻辑处理资源,进行前移临时S指针,当然,同步的移动临时L指针,以得到所有的或者一组一组的匹配长度,记录下每个S基指针的匹配长度;

[0049] 对所有S基指针得到的匹配子串长度进行比较,只留下最长的匹配长度,并且如果有相同的最大匹配长度,该最大匹配长度距离固定的L指针偏移位置对应最远的那个S基指针位置。

[0050] 此处需要说明的是,保证临时L指针不超越右边界,右边界可能是search窗的右边界或者报文的右边界。

[0051] 在图4中,sub-string(max)最大匹配子串指的是:基于固定的一个S基指针,前移S临时指针,L指针也固定,前移临时L指针,搜寻得到最大的匹配长度;

[0052] 对于其余S基指针,类似操作,并且并行操作处理,即不同的S基指针可以并行同时前进搜寻,以提高搜寻速度,提高压缩速率。

[0053] 本实施例基于预先学习进行自适应性调整最佳匹配级别,即符号单位可调。

[0054] 为了提高前移匹配的速度,本实施例在加速压缩引擎中,基于cache的最大匹配长度预测方法来加快前向匹配速度。

[0055] 如图5所示,基于cache的最大匹配长度预测方法来加快前向匹配速度的过程为:

[0056] 首先,cache包括2个子cache,一个是时间cache,另外一个为空间cache,LZ77加速压缩引擎处理时候,把最大匹配的子串的前缀,前缀可以是一个或2个符号单位,作为键值K,进行哈希计算,得到hash(K),其作为cache的索引,cache索引对应的寻址后数据内容为最大匹配长度,不断进行填充,直到cache满;

[0057] 等到cache存有一定内容,进行cache的hit操作,如果hit,就得到当下正在进行的子串的预测最大匹配长度,如果对于同样的hash(K)索引,匹配长度大于cache里的既存长

度,则进行cache内容,既最大长度更新,如果匹配的位置offset超过预设阈值,则在时间cache里更新,否则,就在空间cache里更新。

[0058] 如果cache发生miss,则进行cache的替换操作,把很久没有hit的cache索引以及对应最大匹配长度替换掉,同理,根据最大匹配时候的位置offset,选择长时程时间相关cache或近时程空间相关cache。

[0059] 步骤3:将学习引擎学习后的结果馈入到评估引擎,以确定是否具有最小的总长度。

[0060] 其中,学习引擎学习后的结果包括输出的参数,比如对于不同符号单位的最大匹配的位置偏移offset,针对不同符号单位的最大匹配长度,未压缩的符号单位等。

[0061] 步骤4:将符号单位评估信息馈入到仲裁器进行仲裁选择,由仲裁器决定压缩引擎的最优压缩符号单位。

[0062] 其中,符号单位评估信息包括符号长度评估以及可能的压缩反弹警告。

[0063] 为了提高压缩速度,本实施例在加速压缩引擎中,多路S基指针位置guess淘汰候选方法,以候选者进行异或匹配计算,输出多路S基指针并行搜索结果。

[0064] 如图6所示,当cache的hit操作成功,得到最大匹配长度L。加速压缩引擎对报文数据进行压缩,多路S基指针并行匹配,得到多路位置,多路位置经淘汰,得到候选位置,将匹配长度变为最大匹配长度的2倍,判断子串后缀是否匹配,若是则匹配长度为2L,若不匹配,则匹配长度为 $2/3 * L$;最后将候选者进行XOR异或匹配计算,结束多路S基指并行搜索并输出搜索结果。

[0065] 本实施例利用加速压缩引擎对于待压缩报文进行压缩处理,并将压缩后的报文数据馈入至学习引擎进行学习,以确定符号单位是否最优,实现了基于预先学习进行自适应性调整最佳匹配级别;将学习引擎学习后的结果馈入到评估引擎,以确定是否具有最小的总长度;将符号单位评估信息馈入到仲裁器进行仲裁选择,由仲裁器决定压缩引擎的最优压缩符号单位,提高了报文压缩匹配的灵活性以及压缩处理速度,平衡了电路资源代价与时间代价。

[0066] 实施例二

[0067] 本实施例提供了一种报文压缩系统,包括学习引擎、加速压缩引擎、评估引擎和仲裁器。

[0068] 具体地,学习引擎和加速压缩引擎均用于接收初始化的待压缩报文的符号单位。

[0069] 其中,符号单位可以是单个byte、双byte或者三个byte。

[0070] 加速压缩引擎,用于对于待压缩报文进行压缩处理,并将压缩后的报文数据馈入至学习引擎。

[0071] 在具体实施中,在加速压缩引擎中,采用多路S基指前向匹配针并行的方法加快待压缩报文的压缩;其中,S基指针是sub-string的起始位置。

[0072] sub-string匹配窗的移动多路并行,实现了硬件资源消耗与时钟周期消耗的平衡。

[0073] 具体地,如图3所示,当L指针位置确定后(即look-ahead窗的左边界),需要前向、后向移动S指针,这里后向移动S指针可以并行执行,不存在数据依赖性、存在的结构依赖性可以充分利用多路并行匹配单元解决;

[0074] 如图3所示,L指针暂时固定,以充分移动S指针,S基指针是sub-string的起始位置,需要从L指针的后面一个符号单位的位置遍历到左边界,而左边界可能是search窗口的左边界,也可能是报文的左边界;

[0075] 在图3中,L指针是固定的,只有所有S指针都搜寻完毕,L指针根据搜寻的最大匹配长度,前移,前移的offset是最匹配长度+1;

[0076] S指针的左边界可能是search窗口的左边界,也可能是报文的左边界,需要进行左边界决定;

[0077] S指针可以沿着下降指针,并行去搜索,匹配计算。

[0078] 如图4所示,从每个S基指针向前搜寻最大的匹配sub-string,需要前移临时S指针与临时L指针,以得到最大长度的匹配子串;

[0079] 多路S基指针的前向搜寻,可以并行,即多路S基指针同时操作,或者分成组,一组一组的利用硬件匹配逻辑处理资源,进行前移临时S指针,当然,同步的移动临时L指针,以得到所有的或者一组一组的匹配长度,记录下每个S基指针的匹配长度;

[0080] 对所有S基指针得到的匹配子串长度进行比较,只留下最长的匹配长度,并且如果有相同的最大匹配长度,该最大匹配长度距离固定的L指针偏移位置对应最远的那个S基指针位置。

[0081] 此处需要说明的是,保证临时L指针不超越右边界,右边界可能是search窗的右边界或者报文的右边界。

[0082] 在图4中,sub-string(max)最大匹配子串指的是:基于固定的一个S基指针,前移S临时指针,L指针也固定,前移临时L指针,搜寻得到最大的匹配长度;

[0083] 对于其余S基指针,类似操作,并且并行操作处理,即不同的S基指针可以并行同时前进搜寻,以提高搜寻速度,提高压缩速率。

[0084] 本实施例基于预先学习进行自适应性调整最佳匹配级别,即符号单位可调。

[0085] 为了提高前移匹配的速度,本实施例在加速压缩引擎中,基于cache的最大匹配长度预测方法来加快前向匹配速度。

[0086] 如图5所示,基于cache的最大匹配长度预测方法来加快前向匹配速度的过程为:

[0087] 首先,cache包括2个子cache,一个是时间cache,另外一个为空间cache,LZ77加速压缩引擎处理时候,把最大匹配的子串的前缀,前缀可以是一个或2个符号单位,作为键值K,进行哈希计算,得到hash(K),其作为cache的索引,cache索引对应的寻址后数据内容为最大匹配长度,不断进行填充,直到cache满;

[0088] 等到cache存有一定内容,进行cache的hit操作,如果hit,就得到当下正在进行的子串的预测最大匹配长度,如果对于同样的hash(K)索引,匹配长度大于cache里的既存长度,则进行cache内容,既最大长度更新,如果匹配的位置offset超过预设阈值,则在时间cache里更新,否则,就在空间cache里更新。

[0089] 如果cache发生miss,则进行cache的替换操作,把很久没有hit的cache索引以及对应最大匹配长度替换掉,同理,根据最大匹配时候的位置offset,选择长时程时间相关cache或近时程空间相关cache。

[0090] 学习引擎,用于对接收的报文数据进行学习,以确定符号单位是否最优;还将学习后的结果馈入到评估引擎。

[0091] 其中,学习引擎学习后的结果包括输出的参数,比如对于不同符号单位的最大匹配的位置偏移offset,针对不同符号单位的最大匹配长度,未压缩的符号单位等。

[0092] 评估引擎,用于对学习后的结果进行评估,以确定是否具有最小的总长度;还将符号单位评估信息馈入到仲裁器。

[0093] 仲裁器,其用于仲裁选择,决定压缩引擎的最优压缩符号单位。

[0094] 为了提高压缩速度,本实施例在仲裁器中,多路S基指针位置guess淘汰候选方法,以候选者进行异或匹配计算,输出多路S基指针并行搜索结果。

[0095] 如图6所示,当cache的hit操作成功,得到最大匹配长度L。加速压缩引擎对报文数据进行压缩,多路S基指针并行匹配,得到多路位置,多路位置经淘汰,得到候选位置,将匹配长度变为最大匹配长度的2倍,判断子串后缀是否匹配,若是则匹配长度为2L,若不匹配,则匹配长度为 $2/3 * L$;最后将候选者进行XOR异或匹配计算,结束多路S基指并行搜索并输出搜索结果。

[0096] 其中,符号单位评估信息包括符号长度评估以及可能的压缩反弹警告。

[0097] 本实施例利用加速压缩引擎对于待压缩报文进行压缩处理,并将压缩后的报文数据馈入至学习引擎进行学习,以确定符号单位是否最优,实现了基于预先学习进行自适应性调整最佳匹配级别;将学习引擎学习后的结果馈入到评估引擎,以确定是否具有最小的总长度;将符号单位评估信息馈入到仲裁器进行仲裁选择,由仲裁器决定压缩引擎的最优压缩符号单位,提高了报文压缩匹配的灵活性以及压缩处理速度,平衡了电路资源代价与时间代价。

[0098] 实施例三

[0099] 本实施例提供了一种计算机可读存储介质,其上存储有计算机程序,该程序被处理器执行时实现如上述实施例一所述的报文压缩方法中的步骤。

[0100] 本实施例利用加速压缩引擎对于待压缩报文进行压缩处理,并将压缩后的报文数据馈入至学习引擎进行学习,以确定符号单位是否最优,实现了基于预先学习进行自适应性调整最佳匹配级别;将学习引擎学习后的结果馈入到评估引擎,以确定是否具有最小的总长度;将符号单位评估信息馈入到仲裁器进行仲裁选择,由仲裁器决定压缩引擎的最优压缩符号单位,提高了报文压缩匹配的灵活性以及压缩处理速度,平衡了电路资源代价与时间代价。

[0101] 实施例四

[0102] 本实施例提供了一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,所述处理器执行所述程序时实现如上述实施例一所述的报文压缩方法中的步骤。

[0103] 本实施例利用加速压缩引擎对于待压缩报文进行压缩处理,并将压缩后的报文数据馈入至学习引擎进行学习,以确定符号单位是否最优,实现了基于预先学习进行自适应性调整最佳匹配级别;将学习引擎学习后的结果馈入到评估引擎,以确定是否具有最小的总长度;将符号单位评估信息馈入到仲裁器进行仲裁选择,由仲裁器决定压缩引擎的最优压缩符号单位,提高了报文压缩匹配的灵活性以及压缩处理速度,平衡了电路资源代价与时间代价。

[0104] 本领域内的技术人员应明白,本发明的实施例可提供为方法、系统、或计算机程序

产品。因此,本发明可采用硬件实施例、软件实施例、或结合软件和硬件方面的实施例的形式。而且,本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器和光学存储器等)上实施的计算机程序产品的形式。

[0105] 本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0106] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0107] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0108] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,所述的程序可存储于一计算机可读取存储介质中,该程序在执行时,可包括如上述各方法的实施例的流程。其中,所述的存储介质可为磁碟、光盘、只读存储记忆体(Read-Only Memory,ROM)或随机存储记忆体(Random AccessMemory,RAM)等。

[0109] 以上所述仅为本发明的优选实施例而已,并不用于限制本发明,对于本领域的技术人员来说,本发明可以有各种更改和变化。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

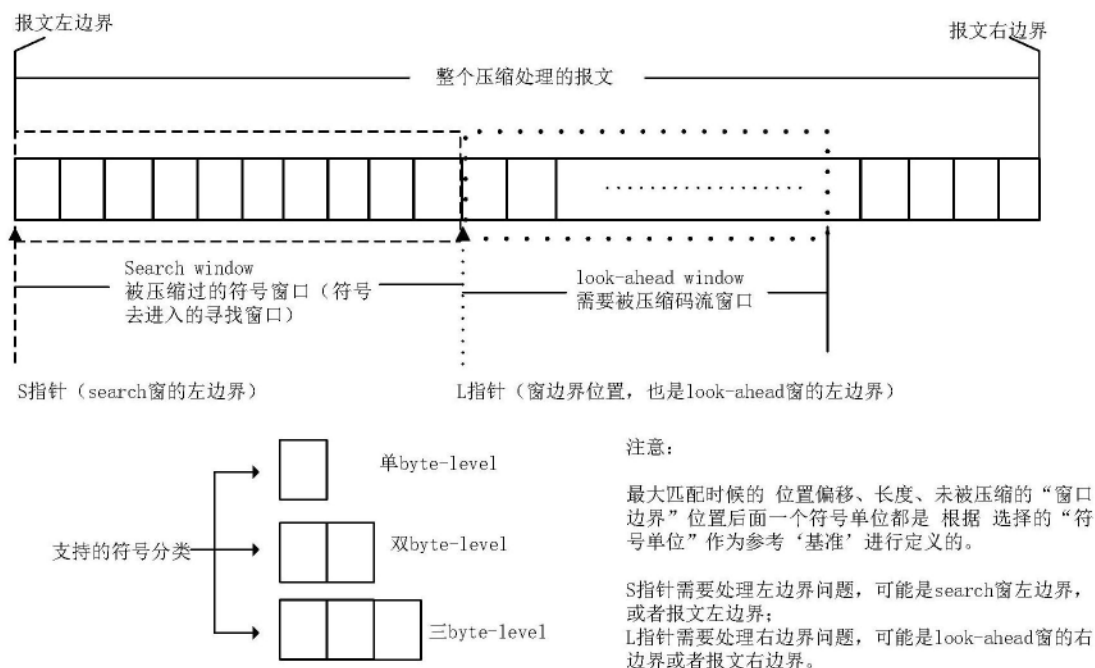


图1

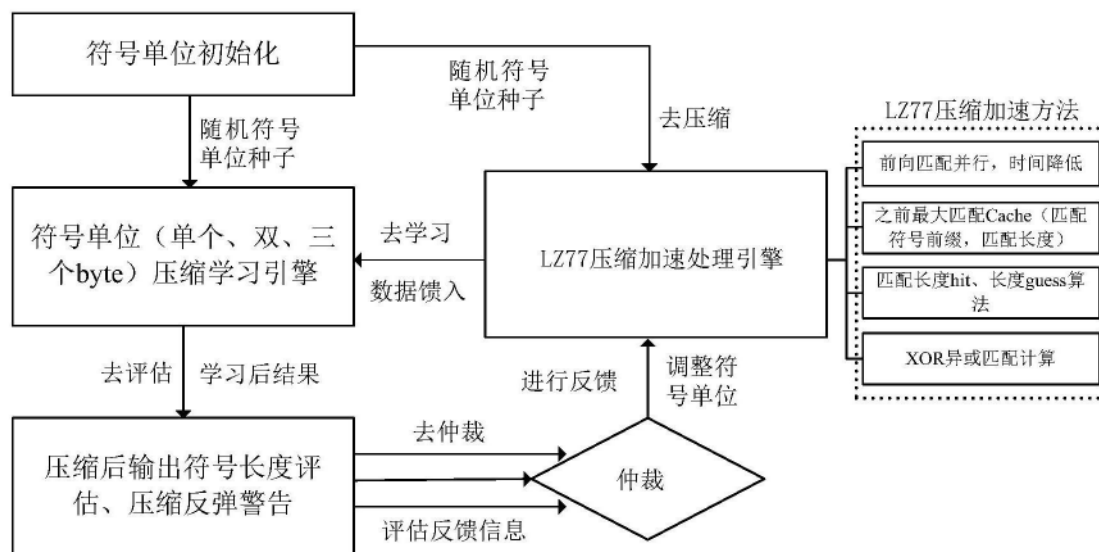


图2

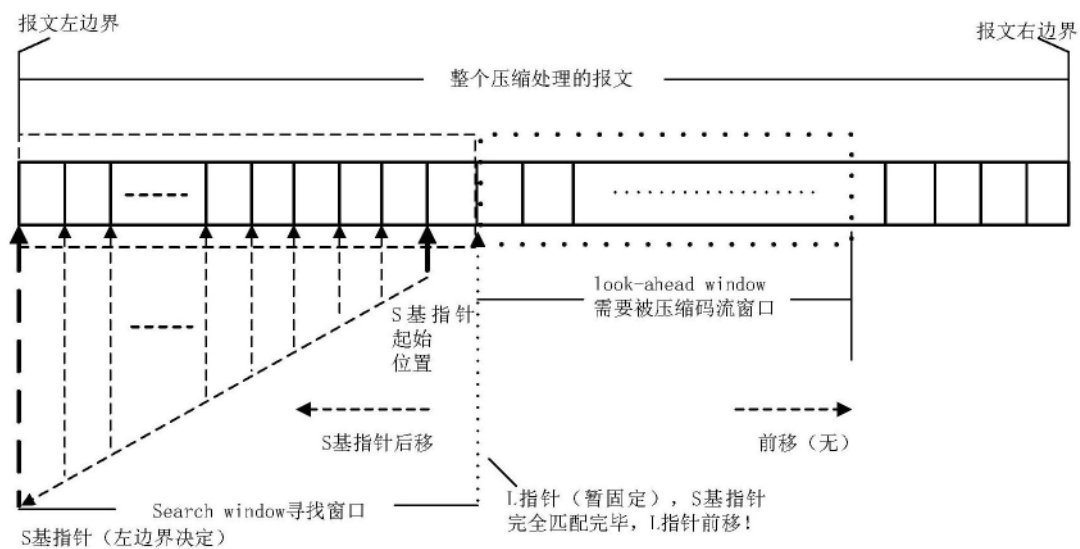


图3

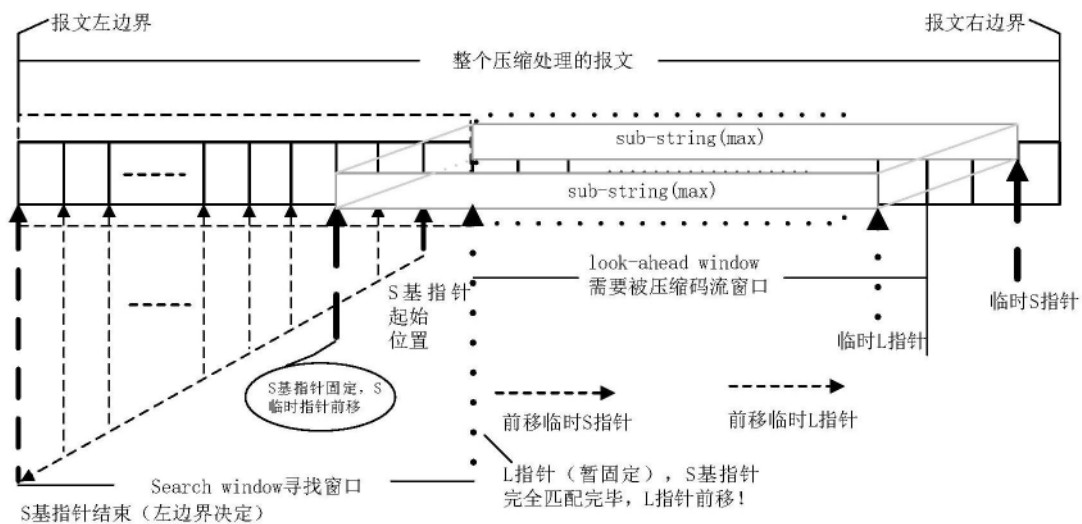


图4

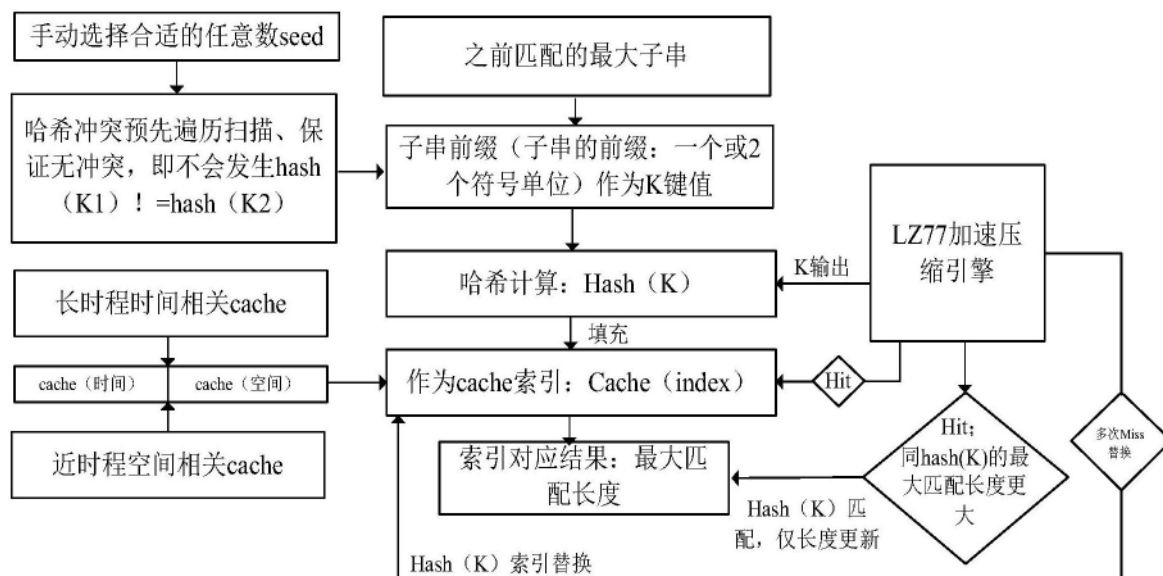


图5

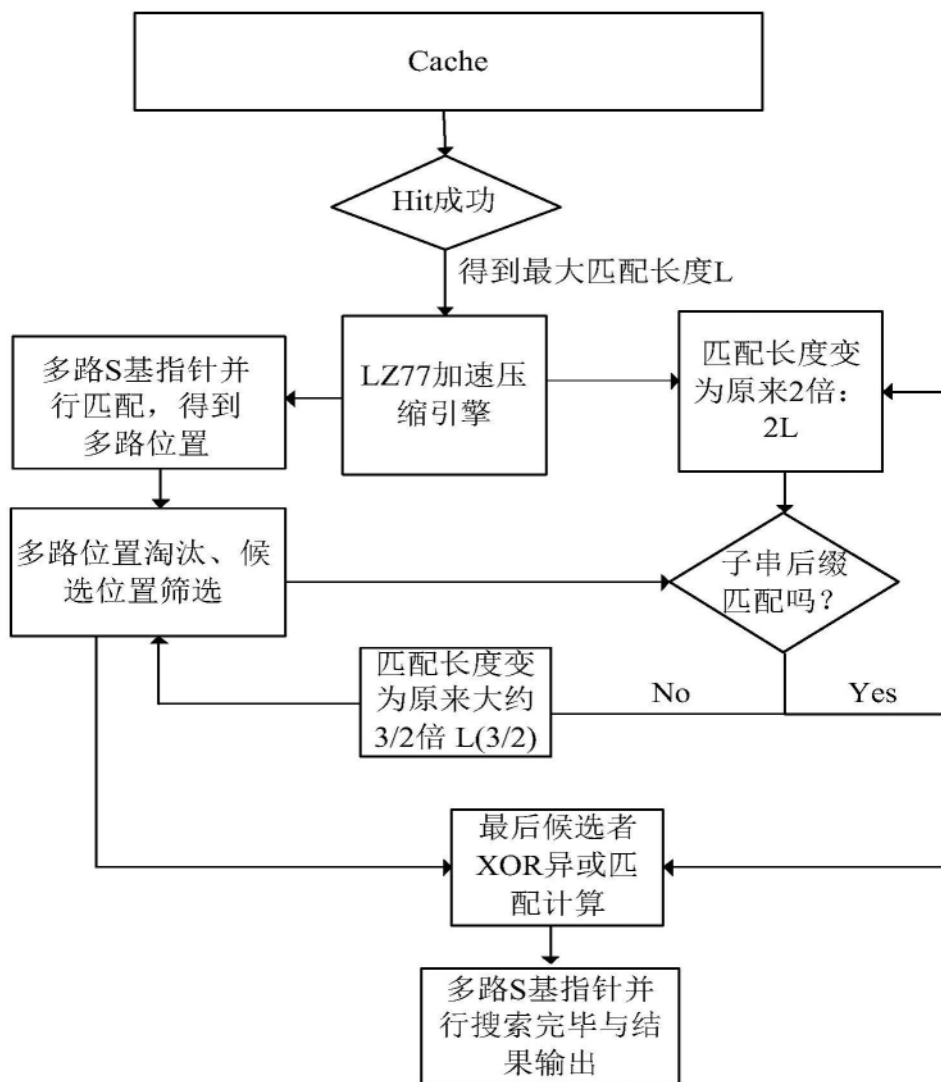


图6