



(12) 发明专利申请

(10) 申请公布号 CN 102457283 A

(43) 申请公布日 2012. 05. 16

(21) 申请号 201010528952. 2

(22) 申请日 2010. 10. 28

(71) 申请人 阿里巴巴集团控股有限公司

地址 英属开曼群岛大开曼岛资本大厦一座  
四层 847 号邮箱

(72) 发明人 杨含飞 马天笑

(74) 专利代理机构 北京同达信恒知识产权代理  
有限公司 11291

代理人 郭润湘

(51) Int. Cl.

H03M 7/30 (2006. 01)

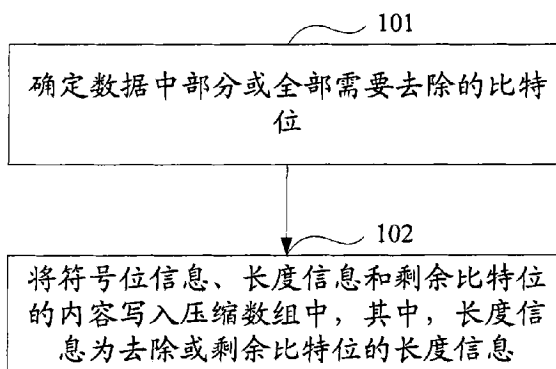
权利要求书 2 页 说明书 9 页 附图 3 页

(54) 发明名称

一种数据压缩、解压缩方法及设备

(57) 摘要

本申请公开了一种数据压缩、解压缩方法及设备, 主要内容包括: 在对数据进行压缩时, 去除数据中部分或全部的无效比特位, 只将剩余的比特位的内容、长度信息以及符号位写入压缩数组中进行存储以及传输。由于本申请方案无需存储数据的全部无效比特位, 因此, 能够减少存储数据所占用的空间, 使数据压缩效率较高, 减少存储、传输的数据量。



1. 一种数据压缩的方法,其特征在于,包括:

确定数据中部分或全部需要去除的比特位;

将所述数据的符号位信息、长度信息和剩余比特位的内容作为压缩后的内容,写入压缩数组中,其中,所述长度信息为去除或剩余的比特位的长度信息。

2. 如权利要求 1 所述的方法,其特征在于,在确定数据中部分或全部需要去除的比特位之前,所述方法还包括:

若确定所述数据是负数,则记录该数据是负数的信息,并将所述数据转换为正数。

3. 如权利要求 1 所述的方法,其特征在于,确定数据中部分或全部需要去除的比特位,具体包括:

按照数据中各比特位的顺序,依次判断每个包含 X 个比特位的单元内是否存在内容非 0 比特位;

在确定首个存在内容非 0 比特位的单元时,确定之前不包含非 0 比特位的单元内的比特位是需要去除的比特位;

所述 X 为大于 1 且不大于数据的总比特位长度一半的正整数。

4. 如权利要求 3 所述的方法,其特征在于,所述数据的总比特位长度为所述 X 的整数倍,所述长度信息为去除或剩余的比特位的单元的数量;

通过以下公式确定需要写入压缩数组中剩余比特位的单元的数量:

$$Y = (L/X) - Z$$

其中:Y 表示需要写入压缩数组中剩余比特位的单元的数量;L 为数据的总比特位长度;Z 表示首个存在内容非 0 比特位的单元之前,不包含非 0 比特位的单元的数量。

5. 如权利要求 1 至 4 中任一权利要求所述的方法,其特征在于,所述压缩数组中有第一指针和第二指针,所述第一指针指向压缩数组中排在首位的可用字节,所述第二指针指向该可用字节中排在首位的可用比特位,当压缩后的数据内容写入压缩数组后,移动所述第一指针和第二指针。

6. 如权利要求 5 所述的方法,其特征在于,在确定剩余比特位的长度信息之后,且在将压缩后的内容写入压缩数组之前,还包括:

根据所述第一指针和第二指针的指向确定压缩数组中当前可用的比特位长度 M,并判断 M 是否小于需要写入压缩数组中的内容占用的比特位长度 N;

若是,则扩充压缩数组的容量,使得扩充后压缩数组可用的比特位长度不小于 N,并执行将压缩后的内容写入压缩数组的操作;

若否,则执行将压缩后的内容写入压缩数组的操作。

7. 一种数据解压缩的方法,其特征在于,包括:

获得压缩数组;

根据数据总的比特位长度、压缩数组中所述数据的符号位信息、长度信息和剩余比特位的内容作为压缩后的内容,得到解压缩的数据,其中,所述长度信息为去除或剩余的比特位的长度信息。

8. 如权利要求 7 所述的方法,其特征在于,若在压缩数据时,所述压缩数组中比特位的长度信息 Y 是通过公式  $Y = (L/X) - Z$  确定的,则去除的比特位长度  $P = Z \times X$ ;

其中:L 为数据的总比特位长度;Z 表示首个存在内容非 0 比特位的单元之前,不包含

非 0 比特位的单元的数量 ;X 为大于 1 且不大于数据的总比特位长度一半的正整数。

9. 一种数据压缩设备,其特征在于,包括:

确定模块,用于确定数据中部分或全部需要去除的比特位;

压缩模块,用于将所述数据的符号位信息、长度信息和剩余比特位的内容作为压缩后的内容,写入压缩数组中,其中,所述长度信息为去除或剩余的比特位的长度信息。

10. 一种解压缩设备,其特征在于,包括:

获得模块,用于获得压缩数组;

解压缩模块,用于根据数据总的比特位长度、压缩数组中所述数据的符号位信息、长度信息和剩余比特位的内容作为压缩后的内容,得到解压缩的数据,其中,所述长度信息为去除或剩余的比特位的长度信息。

## 一种数据压缩、解压缩方法及设备

### 技术领域

[0001] 本申请涉及计算机技术领域,尤其涉及一种数据压缩、解压缩方法及设备。

### 背景技术

[0002] 在需要对大量数据进行处理系统中,往往需要对大批量的数据进行存储和传输,数据量越大,存储和传输这些数据所占用的存储空间、计算资源以及运行时间也就越多,对数据量大、并发量高的系统而言,存储和传输大批量数据可能会降低系统性能。

[0003] 为了减少存储和传输大批量数据对系统性能的影响,常用的方式是,通过压缩数据来减少存储、传输的数据量以提高数据库的读写性能以及数据的传输性能。目前常用的压缩数据的方式主要有去重压缩和差分压缩两种方式。

[0004] 去重压缩方式是指:针对各数据中出现的相同字符串,只保存一个副本且记录每个相同字符串出现的位置,删除数据中重复出现的字符串,从而实现对数据的压缩。去重压缩方式对于文本格式的数据而言效果较好,特别是针对文本中有大量重复字符串的情况,效果尤为明显。

[0005] 差分压缩方式是指:从一批数值型的数据中选择一个或者几个数据作为基准数据,然后计算其他数据与基准数据之间的差值,最后只存储基准数据和差值,从而实现数据的压缩。

[0006] 不论是去重压缩方式还是差分压缩方式,压缩后存储的数据都是文本格式的数据,而上述两种压缩方式对于非文本格式的数据压缩效果却不明显。下面以长整型数据为例,说明去重压缩方式和差分压缩方式对非文本格式的数据的压缩效果较差的问题。

[0007] 针对去重压缩方式对长整型数据的压缩:

[0008] 由于去重压缩方式是对文本结构的字符串进行操作,因此,需要将长整型数据转换为字符串格式。字符串中的每个字符占用 2 字节,而一个长整型数据占用 8 字节,因此,在将长整型数据转换为字符串形式后,由于如果转换后字符的数量超过 4,因此,长整型数据转换成字符串形式的数据后占用的空间反而大于原长整型数据占用的空间,甚至是原长整型数据占用空间的数倍,如原长整型数据占用的空间是 8 字节,而转换为字符串后占用的空间反而达到 16 字节。

[0009] 由于长整型数据转换为字符串后增加了占用的空间,因此,即使通过去重压缩方式对转换为字符串形式的数据进行压缩减少数据量,但从整体看来,数据量减少并不明显,甚至在长整型数据转换后的字符数量较多时,通过去重压缩后的数据占用的空间多于原长整型数据占用的空间。

[0010] 针对差分方式对长整型数据的压缩:

[0011] 差分方式的原理是用后一个数据减去前一个数据,只存储两个数据的差值,而对于长整型数据而言,两个长整型数据相减得到的还是长整型数据,仍需要占用 8 字节。若将通过差分方式压缩后的长整型数据再进行去重方式的压缩效果相对较好,但差分方式压缩只对有规律性的长整型数据有效果,对于随机性较高的长整型数据的压缩效果较差。对于

随机性较高的长整型数据进行差分方式压缩后再进行去重压缩的效果也不明显。

[0012] 综上所述,目前对非文本形式的数据(如长整型数据)的压缩方式效果较差,需要找到一种针对非文本形式的数据(如长整型数据)的高效压缩方式,以减少一批量数据在存储、传输时对系统性能的影响。

### 发明内容

[0013] 本申请目的在于:提供一种数据压缩、解压缩方法及设备,用以解决现有技术中存在对非文本形式的数据的压缩效果较差的问题。

[0014] 一种数据压缩的方法,包括:

[0015] 确定数据中部分或全部需要去除的比特位;

[0016] 将所述数据的符号位信息、长度信息和剩余比特位的内容作为压缩后的内容,写入压缩数组中,其中,所述长度信息为去除或剩余的比特位的长度信息。

[0017] 一种数据解压缩的方法,包括:

[0018] 获得压缩数组;

[0019] 根据数据总的比特位长度、压缩数组中所述数据的符号位信息、长度信息和剩余比特位的内容作为压缩后的内容,得到解压缩的数据,其中,所述长度信息为去除或剩余的比特位的长度信息。

[0020] 一种数据压缩设备,包括:

[0021] 确定模块,用于确定数据中部分或全部需要去除的比特位;

[0022] 压缩模块,用于将所述数据的符号位信息、长度信息和剩余比特位的内容作为压缩后的内容,写入压缩数组中,其中,所述长度信息为去除或剩余的比特位的长度信息。

[0023] 一种解压缩设备,包括:

[0024] 获得模块,用于获得压缩数组;

[0025] 解压缩模块,用于根据数据总的比特位长度、压缩数组中所述数据的符号位信息、长度信息和剩余比特位的内容作为压缩后的内容,得到解压缩的数据,其中,所述长度信息为去除或剩余的比特位的长度信息。

[0026] 本申请有益效果如下:

[0027] 本申请实施例在对数据进行压缩时,去除数据中部分或全部无效的比特位,只将数据剩余的比特位的内容、长度信息以及符号位信息写入压缩数组中进行存储以及传输。由于本申请方案无需将全部的无效比特位压缩存储,因此,能够减少存储数据所占用的空间,使数据压缩效率较高,减少存储、传输的数据量。

### 附图说明

[0028] 图1为本申请实施例一数据压缩的方法示意图;

[0029] 图2为本申请实施例二数据压缩的方法示意图;

[0030] 图3为本申请实施例二压缩数组中存储的压缩后数据的格式;

[0031] 图4为本申请实施例三数据解压缩的方法示意图;

[0032] 图5为本申请实施例四数据压缩设备结构示意图;

[0033] 图6为本申请实施例五数据解压缩设备结构示意图。

## 具体实施方式

[0034] 由于非文本格式的数据（如长整型数据）中的字节以及各字节中的比特位并没有全部有效利用，也就是说，数据内容并不一定需要所有的比特位来表示，大多数情况下，只需要部分比特位就能够表示，为此，本申请从有效比特位的利用出发，提出了一种新的数据压缩方式，在存储非文本格式的数据（如长整型数据）时，通过去除部分或全部的无效比特位，将数据的符号位信息、数据的剩余的比特位的内容以及长度信息写入压缩数组中进行存储以及传输，实现数据的压缩。由于本申请方案无需存储数据的部分或全部的无效比特位，因此，能够减少存储数据所占用的空间，特别是针对无效比特位长度较长的情况下，压缩效果更加明显，有效地降低数据存储、传输时对系统性能的影响。

[0035] 本申请中用于压缩的数据是指：以计算机可读形式存储的非文本数据，如长整型（long）数据、整型（int）数据、无符号整型（unsigned）数据等多种类型的数据。后续的各实施例以长整型数据为例进行说明。

[0036] 本申请中涉及的压缩数组是在内存中开辟的用于存储压缩后数据的存储空间，在生成压缩数组时初始化该压缩数组的大小，并在需要时扩充压缩数组的大小，即增加用于存储压缩后数据的存储空间。

[0037] 本申请中涉及的比特位的长度是指比特位的数量。

[0038] 下面结合说明书附图对本申请实施例进行详细说明。

[0039] 实施例一

[0040] 如图 1 所示，为本申请实施例一中压缩数据的方法示意图，所述方法包括以下步骤：

[0041] 步骤 101：确定数据中部分或全部需要去除的比特位；

[0042] 步骤 102：将所述数据的符号位信息、长度信息和剩余比特位的内容作为压缩后的内容写入压缩数组中，其中，所述长度信息为去除或剩余比特位的长度信息，即去除掉的比特位的长度信息或者去除比特位后剩余的比特位的长度信息。

[0043] 通过本发明实施例一的方案，在压缩数据时，去除数据中部分或全部的无效数据，只对剩余的数据进行压缩处理，可以有效地减少存储的数据所占用的空间，提高数据压缩效率。

[0044] 实施例二

[0045] 在数据压缩过程中，需要对大量长整型数据进行压缩，本实施例可以按照串行方式依次对每一长整型数据执行本实施例各步骤的方案得到压缩后的长整型数据，也可以并行对多个长整型数据分别执行本实施例各步骤的方案。本实施例一针对一个长整型数据进行压缩为例进行说明。

[0046] 如图 2 所示，为本申请实施例二中压缩数据的方法示意图，所述方法包括以下步骤：

[0047] 步骤 201：确定数据中部分或全部需要去除的无效比特位；

[0048] 在压缩一个长整型数据之前，需要知道这个长整型数据的可压缩的无效比特位。一个长整型数据需要占用 8 字节，也就是 64 比特位，可以按照 64 比特位从高位至低位的顺序或者从低位至高位的顺序，确定 64 比特位中可以去除的无效比特位和剩余的比特位。

[0049] 若长整型数据是从低位起写入内容的,可按照高位至低位的顺序确定可以去除的比特位,通过去除高位的无效比特位而保留低位的剩余比特位的压缩方式,则由于剩余的低位的内容仍然是压缩前数据的内容,因此在解压缩时只需要补足高位的无效比特位,即能够正确解压缩。

[0050] 若按照低位至高位的顺序确定有效比特位,通过去除部分低位的内容而保留高位的内容的压缩方式,虽然压缩后剩余的高位比特位的内容不等于压缩前数据的内容,但是只要在解压缩时按照相同的顺序补足低位的内容,仍然能够正确解压缩。

[0051] 下面以从高位至低位的顺序为例,提供一种优选的确定需要去除的比特位的方式:

[0052] 第一步:从高位至低位依次判断数据中每个包含 X 个比特位的单元内是否存在内容非 0 比特位。

[0053] 其中,X 为大于 1 且不大于数据的总比特位长度一半的正整数,并且所述 X 能被数据的总比特位长度整除。由于本实施例中的数据是长整型数据,则 X 为大于 1 且不大于 32 的正整数且 X 能被 64 整除。

[0054] 其中,数据的总的比特位的长度以及比特位单元 X 的信息可以记录于压缩数组的附加信息中,解压时,首先获取该信息。

[0055] 本实施例中,X 的取值与本步骤执行时需要的运算次数以及可压缩的程度相关。若 X 取值较小,如 X 取值为 2,则 64 比特位的长整型数据中包括有 32 个包含 2 个比特位的单元,在判断存在非 0 比特位的单元时,最多需要运算 32 次,这将导致运算次数较多,但由于 X 取值较小,查找出的可去除的无效比特位较多,因此压缩效果较好。若 X 取值较大,如 X 取值为 8,64 比特位的长整型数据中有 8 个包含 8 个比特位的单元,在判断存在非 0 比特位的单元时,最多只需要运算 8 次,这样运算次数较少,但由于 X 取值较大,出现连续 8 个比特位内容为 0 的概率小于连续 2 个比特位内容为 0 的概率,因此查找出的可去除的无效比特位较少,压缩效果相对较差。通过仿真实验认为 X 取值为 4 时能够综合运算次数以及可压缩的程度的需求,仿真结果较好。

[0056] 若  $X = 4$ ,则以连续的 4 比特位为一个单元,从高位至低位依次查找是否存在包含非 0 比特位的单元。

[0057] 例如:某一长整型数据的 64 比特位为:0000 0000 0010.....0000,其中,左侧为高位,右侧为低位,在以 4 比特位为 1 个单元进行判断时,第一单元不存在非 0 的比特位,第二单元也不存在非 0 的比特位,第三单元存在非 0 的比特位。

[0058] 第二步:在确定出首个存在内容非 0 比特位的单元时,确定之前不包含非 0 比特位的单元内的比特位是需要去除的无效比特位。

[0059] 第三步:用数据的总比特位长度减去需要去除的无效比特位长度,得到剩余比特位的长度。

[0060] 具体地,可以通过下式计算确定剩余比特位长度:

$$[0061] \quad Y = (L/X) - Z \quad (1)$$

[0062] 其中:Y 表示需要写入压缩数组中剩余比特位的单元的数量;L 为数据的总比特位长度;Z 表示首个存在内容非 0 比特位的单元之前,不包含非 0 比特位的单元的数量,即需要去除的无效比特位所在的单元的数量。

[0063] 此时得到的 Y 值可以作为写入压缩数组中的长度信息,即写入压缩数组中剩余的比特位的单元的数量;也可以将 Z 值作为写入压缩数组中的长度信息,即去除的比特位的单元的数量。

[0064] 当然,本实施例也不限于将  $Y \times X$  得到的值作为写入压缩数组中的长度信息,即写入压缩数组中剩余比特位的数量;也可以将  $Z \times X$  得到的值作为写入压缩数组中的长度信息,即去除的比特位的数量。

[0065] 去除的比特位是无效比特位,剩余的比特位可以是有效的比特位,也可以包含有效比特位和无效比特位。例如:某一长整型数据的 64 比特位为:00000000 0010.....0000,若  $X = 4$ ,则去除的无效比特位是高 8 位,剩余的 56 比特位中,最高的 2 位也可是看作是无效比特位,后 54 是有效比特位;若  $X = 2$ ,则去除的无效比特位是高 10 位,剩余的 54 比特位是有效比特位。

[0066] 根据第一步所举实例,确定第一单元和第二单元中的 8 比特位是无效比特位,则后 56 位是有效比特位,有效比特位的长度为 56;从第三单元中的 0010 起,直至第 16 单元中的 0000,都是待存储的比特位内容。

[0067] 需要说明的是,本实施例中压缩的长整型数据可以是正数也可以是负数,若压缩长整型数据是正数,则执行步骤 201;若压缩的长整型数据是负数,则需要首先记录该数据是负数的信息,并将所述数据通过运算转换为正数后再执行步骤 201。

[0068] 步骤 202:判断压缩数组中当前可用的比特位长度 M 是否小于需要写入压缩数组中的内容需要占用的比特位长度 N,若是,则执行步骤 203;否则,执行步骤 205。

[0069] 所述 M、N 是正整数。

[0070] 如图 3 所示,为压缩数组中存储的压缩后的长整型数据格式,其中:

[0071] 第一位是符号位,用于存储数据是正数或负数的信息,如:在数据是正数时,第一位中存储的内容为 0,在数据是负数时,第一位中存储的内容为 1。

[0072] 第二至第五位是长度位,用于存储去除部分或全部无效比特位后剩余的比特位长度。这里的长度位可以直接存储比特位的数量,如去除 8 比特位,则长整型数据还剩余 56 比特位,则需要 6bit 来记录 56 这一数值。

[0073] 为了进一步减少数据压缩后的数据量,在本实施例中,也可以根据 X 的不同取值,在长度位中记录剩余比特位的单元的数量。如:若  $X = 4$ ,则长整型数据最多可划分为 16 个单元,即使每一单元内都存在非 0 比特位(即不存在需要去除的无效比特位),只需要 4bit 就足够存储 16 这一数值,在解码时,根据压缩数组中剩余比特位的单元的数量  $\times X =$  剩余的比特位长度,或者根据:数据总长度 - 压缩数组中去除比特位的单元的数量  $\times X =$  剩余的比特位长度。图 3 所示的结构就是以  $X = 4$  为例,预留出 4bit 用于存储剩余或去除的比特位单元的数量信息。若  $X = 8$ ,则长整型数据最多可划分为 8 组,因此,预留出 3bit 用于存储剩余比特位的单元的数量信息即可。

[0074] 从第六位起是用于存储剩余比特位内容的比特位,根据实际需要写入的内容的大小预留相应的空间。

[0075] 在图 3 所示的压缩数组的格式中,符号位、长度位以及存储剩余比特位中内容的比特位之间的先后顺序不限定,但存储剩余比特位中内容的比特位中按照数据压缩前各比特位的顺序填写。



[0076] 由于压缩数组的容量是初始化时定义的,如在存储空间内开辟 20 字节作为初始化时压缩数组的容量,在向压缩数组中写入压缩后的数据时将不断地占用压缩数组的空间,因此,如果压缩处理的长整型数据较多,则可能出现压缩数组剩余空间不足以写入压缩后数据的情况,对此,在向压缩数组中写入压缩数据之前,本优选步骤 202 需要判断压缩数组中空余的比特位是否足以保证压缩数据的完整写入。

[0077] 从图 3 所示的压缩数据格式可以看出,一个长整型数据压缩后需要写入的内容包括:符号位、剩余比特位的长度和剩余比特位的内容,因此,压缩数组中当前可用的比特位长度  $M$  不得小于压缩后数据需要占用的比特位长度  $N$ 。

[0078] 本实施例中引入第一指针和第二指针,所述第一指针指向压缩数组中排在首位的可用字节,所述第二指针指向该可用字节中排在首位的可用比特位,每次向压缩数组中写入数据时,从第一指针和第二指针指向的位置起,依次向压缩数组中写入数据,并在写入数据过程中移动所述第一指针和第二指针。

[0079] 在确定压缩数组中当前可用的比特位长度  $M$  时,确定当前第一指针和第二指针指向的位置,并结合压缩数组在存储空间内结束的位置,计算出压缩数组中当前可用的比特位长度  $M$ 。

[0080] 步骤 203:扩充压缩数组的容量,使得扩充后压缩数组可用的比特位长度不小于  $N$ 。

[0081] 由于压缩数组扩容前可用的比特位为  $M$ ,因此,扩容的比特位不得小于  $N-M$ 。

[0082] 比如:在初始化时,在存储空间内为压缩数组开辟 10 字节的空间,在存储了若干个压缩后的长整型数据之后,根据第一指针和第二指针指向的位置确定压缩数组中还有  $M = 10$  个可用的比特位,而当前长整型数据压缩后需要占用  $N = 20$  个比特位,因此,至少需要扩充 10 比特位,由于扩充的时候是在存储空间内从压缩数组结束位置开始,将新增加的字节追加到原来的压缩数组中,因此,至少需要扩充 2 个字节。

[0083] 步骤 204:将压缩后的内容写入扩容后的压缩数组,并结束对所述数据的压缩操作。

[0084] 若在步骤 203 中扩充 2 字节,则压缩数组中可用的比特位从 10 增加至 26,足够写入压缩后的 20 比特位的内容,则根据当前第一指针和第二指针指向的位置依次写入压缩内容,在内容写入完成后,第一指针指向最后一个字节,第二指针指向最后一个字节的第 3 比特位。

[0085] 步骤 205:将压缩后的内容写入压缩数组中,结束对所述数据的压缩操作。

[0086] 通过上述步骤 201 ~ 步骤 205 的方案,针对非文本格式的数据,通过去除数据中部分或全部的比特位,只存储数据中剩余比特位的内容、长度信息以及符号信息,可以有效地压缩存储的数据量,降低系统存储、传输大数据量的负担。另外,以包含  $X = 4$  个比特位的单元为单位来确定可以去除的无效比特位,可以在节约运算次数的同时尽可能地提高压缩效率。进一步地,在存储剩余或去除比特位的长度信息时,只存储该剩余或去除比特位的单元的数量,这样可以更加提高压缩效率。针对长整型数据对本申请方案进行验证,发现数据的压缩率平均可达 50%,最佳可达 70%左右,压缩效率高。

[0087] 实施例三

[0088] 本申请实施例三是一种解压缩的方法,如图 4 所示,所述方法包括以下步骤:

[0089] 步骤 301 :接收压缩数组和压缩数组的附加信息 ;

[0090] 在本步骤中,压缩数组中的压缩数据可以按照实施例一的方式压缩得到的,也可以是通过其他方式压缩得到的。

[0091] 在本实施例中假设压缩数据是按照实施例一的方式压缩的,则压缩数组按照图 3 所示的结构存储压缩数据。压缩数组的附加信息可以包括数据的总的比特位的长度以及比特位单元 X 的信息。

[0092] 步骤 302 :从压缩数组的压缩数据中获得数据的符号位信息、长度信息和剩余比特位的内容,其中,所述长度信息为去除比特位的长度信息或者去除比特位后剩余的比特位的长度信息 ;从压缩数组的附加信息中获得数据的总的比特位的长度以及比特位单元 X 的信息 ;

[0093] 根据实施例一、实施例二中压缩数据时采用的方式不同,则在本步骤中解压缩的方式也有所不同。若压缩数组中的长度信息是去除比特位后剩余的比特位的长度信息,则以数据的总的比特位的长度为 64,即为长整型数据为例,以 X 个比特位为单元划分原数据的各比特位后,按照公式 (1) 计算确定的比特位长度信息 Y 是剩余的比特位的单元的数量,则去除的比特位长度  $P = [(L/X) - Y] \times X = Z \times X$ 。如 :X = 4,写入压缩数组中的比特位的单元的数量为 3,则写入压缩数组中的比特位长度 = 12,去除的比特位为 52。

[0094] 若压缩数组中的长度信息是去除的比特位的长度信息,仍以数据的总的比特位的长度为 64 为例,在压缩数据时以 X 个比特位为单元划分原数据的各比特位,假设写入压缩数组中的长度信息为 3,表示去除的比特位的单元的数量为 3,则在 X = 4 时,去除的比特位为 12。

[0095] 步骤 303 :根据压缩数据中数据的符号位信息、长度信息和剩余比特位的内容,以及数据的总的比特位的长度和比特位单元 X 的信息,得到解压缩的数据。

[0096] 在解压缩时,需要补充压缩时去除的无效比特位。根据压缩时按照由高位至低位的顺序去除无效比特位,或是由低位至高位的顺序去除无效比特位的顺序不同,补充的内容为 0 的比特位的位置不同。

[0097] 若在压缩数据时,按照比特位由高至低的顺序依次去除无效比特位,则在本步骤中,在所述数据写入压缩数组中的比特位内容的高位补充 P 个内容为 0 的比特位 ;若在压缩数据时,按照比特位由低至高的顺序依次去除无效比特位,则在本步骤中,在所述数据写入压缩数组中的比特位内容的低位补充 P 个内容为 0 的比特位。其中, P 为压缩时去除的比特位的数量。

[0098] 在补充比特位后,得到初步解压缩的数据,进而在根据符号位的内容,得到最终解压缩后的数据。若符号位的内容表示该数据是正数,则初步解压缩的数据与最终解压缩后的数据相同 ;若符号位的内容表示该数据是负数,则将初步解压缩的数据转换为负数后得到最终解压缩后的数据。

[0099] 实施例四

[0100] 与实施例二对应地,本申请实施例三还提供一种数据压缩设备,如图 5 所示,所述设备包括确定模块 11 和压缩模块 12,其中 :

[0101] 确定模块 11 用于确定数据中部分或全部需要去除的比特位 ;

[0102] 压缩模块 12 用于将所述数据的符号位信息、长度信息和剩余比特位的内容作为

压缩后的内容,写入压缩数组中,其中所述长度信息为去除比特位的长度信息或者去除比特位后剩余的比特位的长度信息。

[0103] 所述确定模块 11 具体用于按照数据中各比特位的顺序,依次判断每个包含比特位的单元 X 个比特位的单元单元内是否存在内容非 0 比特位,在确定首个存在内容非 0 比特位的单元时,确定之前不包含非 0 比特位的单元内的比特位是需要去除的无效比特位,所述 X 为不大于 1 且不大于数据的总比特位长度一半的正整数。

[0104] 写入压缩数组中的长度信息可以是该剩余比特位所在单元的数量,或是去除的比特位所在单元的数量。

[0105] 若长度信息是剩余比特位所在单元的数量,则确定模块 11 具体用于通过以下公式确定需要写入压缩数组中剩余比特位的单元的数量:

$$[0106] \quad Y = (L/X) - Z$$

[0107] 其中:Y 表示需要写入压缩数组中剩余比特位的单元的数量;L 为数据的总比特位长度;Z 表示首个存在内容非 0 比特位的单元单元之前,不包含非 0 比特位的单元单元的数量。

[0108] 所述设备还包括转换模块 13,用于在确定所述数据是负数时,记录该数据是负数的信息,并将所述数据转换为正数。

[0109] 所述设备还包括扩容模块 14,用于根据第一指针和第二指针的指向确定压缩数组中当前可用的比特位长度 M,并在判断 M 小于需要写入压缩数组中的内容需要占用的比特位长度 N 时,扩充压缩数组的容量,使得扩充后压缩数组可用的比特位长度不小于 N。

[0110] 所述第一指针指向压缩数组中排在首位的可用字节,所述第二指针指向该可用字节中排在首位的可用比特位,当压缩后的内容写入压缩数组后,移动所述第一指针和第二指针。

[0111] 本实施例四中的数据压缩设备还可以包括能够执行实施例一和实施例二各步骤的逻辑部件。

[0112] 实施例五

[0113] 与实施例三对应地,本申请实施例四还提供一种数据解压缩设备,如图 6 所示,所述设备包括获得模块 21 和解压缩模块 22,其中:

[0114] 获得模块 21,用于获得压缩数组;

[0115] 解压缩模块 22,用于根据数据总的比特位长度、压缩数组中所述数据的符号位信息、长度信息和剩余比特位的内容作为压缩后的内容,得到解压缩的数据,其中,所述长度信息为去除或剩余的比特位的长度信息。

[0116] 所述解压缩模块 22 可以用于根据数据总的比特位长度和长度信息确定需要补充的无效比特位的数量。

[0117] 在长度信息是剩余的比特位的长度时,将数据总的比特位长度减去剩余的比特位的长度得到需要补充的无效比特位的数量;特殊地,在长度信息是剩余的比特位的单元的数量,且每个单元包含 X 个比特位时,剩余的比特位的单元的数量  $\times X =$  剩余的比特位的长度,再将数据总的比特位长度减去剩余的比特位的长度得到需要补充的无效比特位的数量。

[0118] 在长度信息是去除的比特位的长度时,直接根据长度信息补充相应的数量的无效

比特位；特殊地，在长度信息是去除的比特位的单元的数量，且每个单元包含 X 个比特位时，确定去除的无效比特位 = 去除的比特位的单元的数量  $\times X$ 。

[0119] 解压缩设备中的解压缩模块 22 与压缩设备中的压缩模块 12 的功能是相对应的。

[0120] 若压缩设备中的压缩模块 12 按照比特位由高至低的顺序压缩数据，则解压缩模块 22 在写入压缩数组中的比特位内容的高位补充 P 个内容为 0 的比特位；若压缩设备中的压缩模块 12 按照比特位由低至高的顺序压缩数据，则解压缩模块 22 在所述数据写入压缩数组中的比特位内容的低位补充 P 个内容为 0 的比特位。所述 P 为去除的比特位的数量。

[0121] 本实施例五中的数据解压缩设备还可以包括能够执行实施例三各步骤的逻辑部件。

[0122] 本实施例五和实施例五中的压缩设备和解压缩设备可以协同工作，由压缩设备对数据进行压缩并存储，当压缩数据传输至其他设备或需要解压缩时，由解压缩设备按照对应的解压缩方式得到原数据。

[0123] 本领域内的技术人员应明白，本申请的实施例可提供为方法、系统、或计算机程序产品。因此，本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本申请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、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

[0124] 本申请是参照根据本申请实施例的方法、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。

[0125] 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。

[0126] 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。

[0127] 尽管已描述了本申请的优选实施例，但本领域内的技术人员一旦得知了基本创造性概念，则可对这些实施例做出另外的变更和修改。所以，所附权利要求意欲解释为包括优选实施例以及落入本申请范围的所有变更和修改。

[0128] 显然，本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的精神和范围。这样，倘若本申请的这些修改和变型属于本申请权利要求及其等同技术的范围之内，则本申请也意图包含这些改动和变型在内。

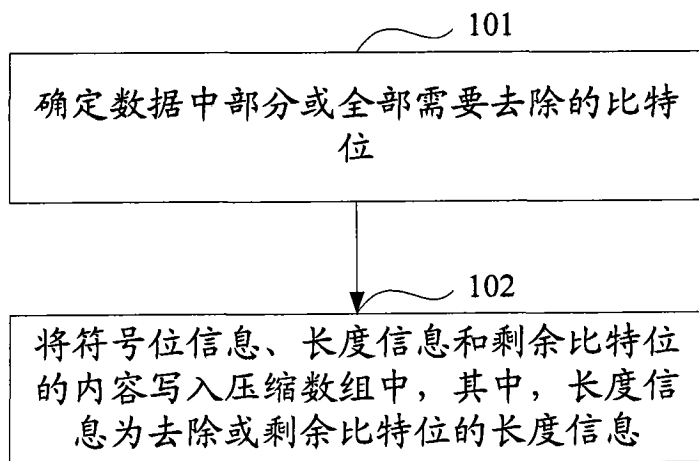


图 1

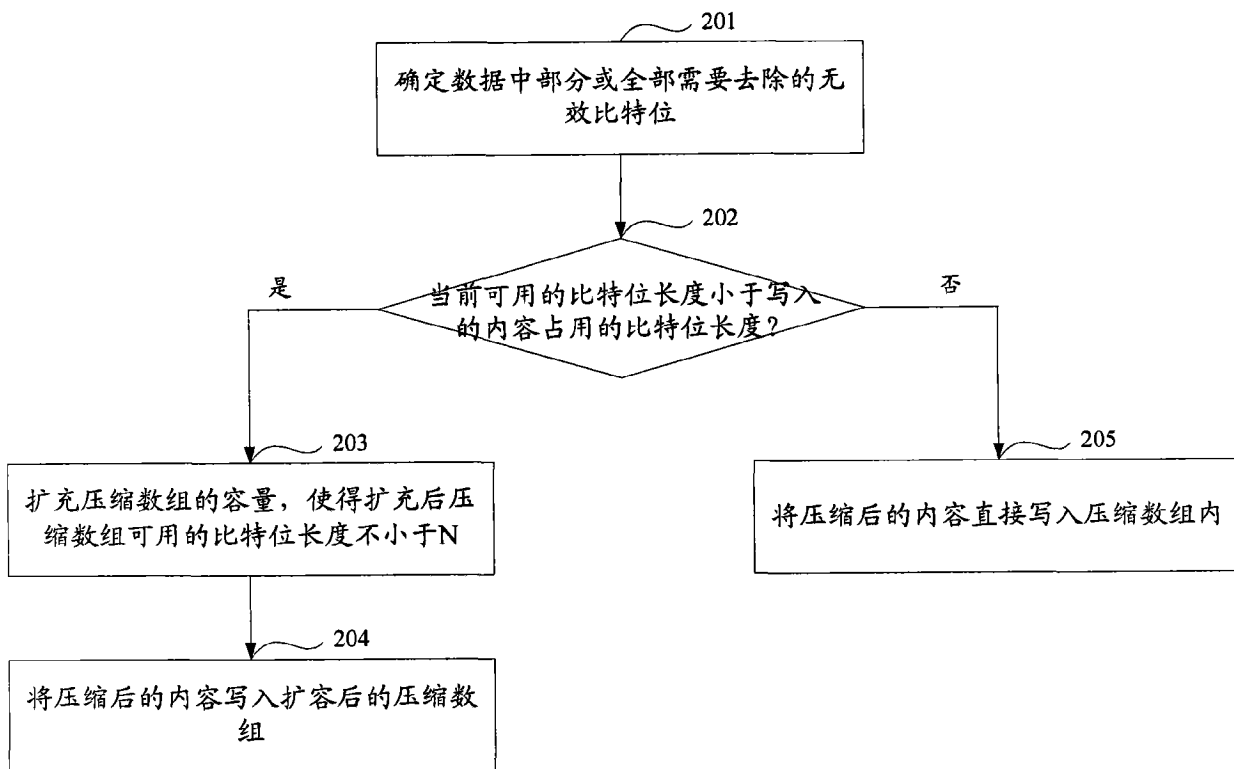


图 2

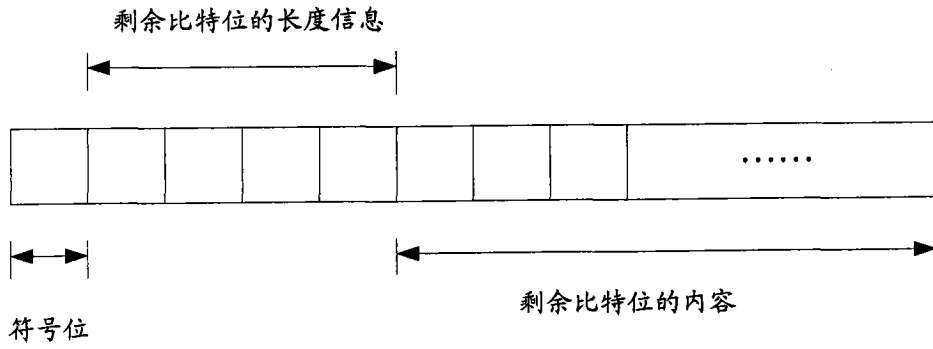


图 3

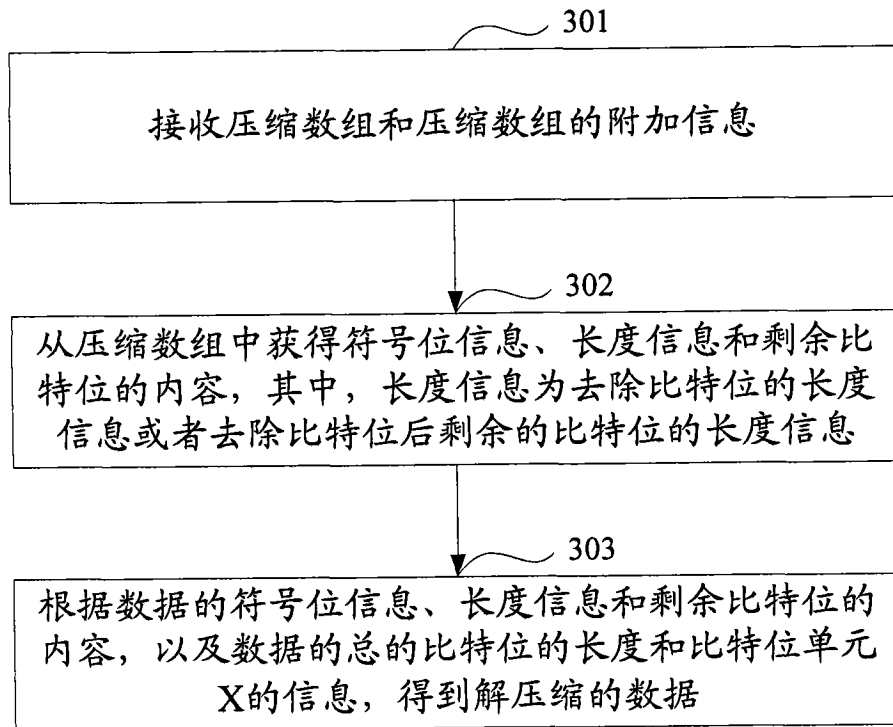


图 4

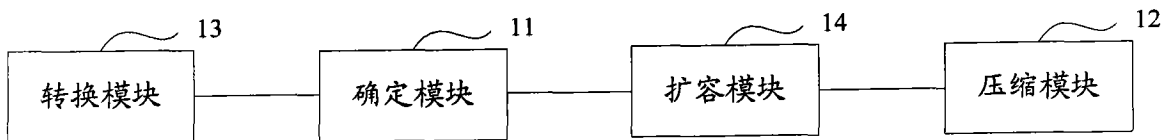


图 5

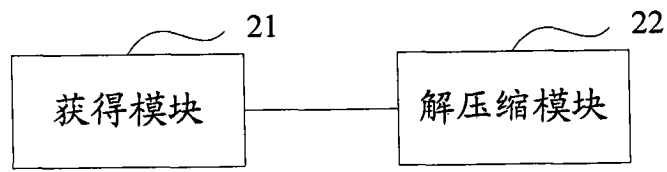


图 6