(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 111447044 A
(43)申请公布日 2020.07.24

(21)申请号 202010161945.7

(22)申请日 2020.03.10

(71)申请人 深圳市大数据研究院
地址 518172 广东省深圳市龙岗区龙翔大道2001号

(72)发明人 付希明 郭沅鑫 杨升浩

(74)专利代理机构 广州嘉权专利商标事务所有限公司 44205
代理人 洪铭福

(51)Int.Cl.
H04L 1/00(2006.01)
H04L 29/08(2006.01)
G06F 3/06(2006.01)

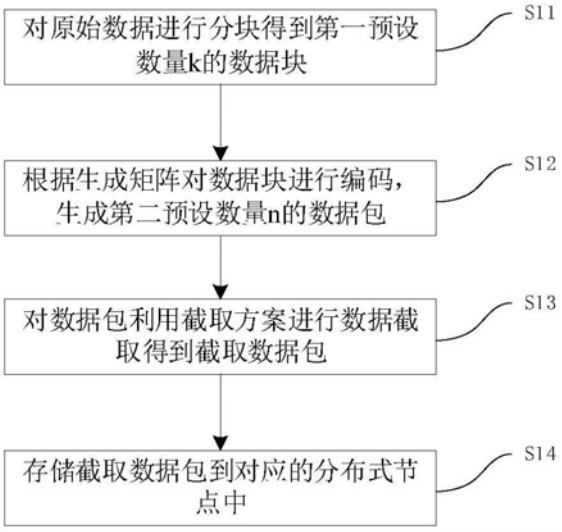
权利要求书2页 说明书7页 附图1页

(54)发明名称

分布式存储方法和传输译码方法

(57)摘要

本发明公开了分布式存储方法和传输译码方法。涉及分布式存储领域，其中，分布式存储方法通过对原始数据进行分块得到第一预设数量的数据块，然后根据生成矩阵对数据块进行编码，生成第二预设数量的数据包，对数据包利用截取方案进行数据截取得到截取数据包，存储截取数据包到对应的分布式节点中。根据本发明实施例的生成矩阵进行编码能够保证较高的编码效率，同时通过截取方案对数据包截取，存储时能够减少节点的存储冗余，从而保证整个分布式系统的存储冗余降低，节约了存储空间，提高了存储效率。



CN 111447044 A

1.一种分布式存储方法,其特征在于,包括:

对原始数据进行分块得到第一预设数量的数据块;

根据生成矩阵对所述数据块进行编码,生成第二预设数量的数据包;

对所述数据包利用截取方案进行数据截取得到截取数据包;

存储所述截取数据包到对应的分布式节点中。

2.根据权利要求1所述的一种分布式存储方法,其特征在于,所述对原始数据进行分块得到第一预设数量的数据块具体包括:所述数据块的数据长度相同。

3.根据权利要求1所述的一种分布式存储方法,其特征在于,所述生成矩阵表示为:

$$\Psi=\begin{pmatrix} z^{t_{1,1}} & z^{t_{1,2}} & \cdots & z^{t_{1,k}} \\ z^{t_{2,1}} & z^{t_{2,2}} & \cdots & z^{t_{2,k}} \\ \cdots & \cdots & \cdots & \cdots \\ z^{t_{n,1}} & z^{t_{n,2}} & \cdots & z^{t_{n,k}} \end{pmatrix}$$

所述数据包表示为:

$$\begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix}=\Psi\begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_k \end{pmatrix}$$

$$y_i=\sum_{j=1}^{k} z^{t_{i,j}}x_j$$

其中,$t_{i,j}$为非负整数,k表示第一预设数量,n表示第二预设数量,x_j表示数据块,y_i表示数据包,$z^{t_{i,j}}x_j$表示将x_j右移$t_{i,j}$个单位,前面补$t_{i,j}$个零。

4.根据权利要求3所述的一种分布式存储方法,其特征在于,所述截取方案表示为:

若满足条件:$n\geq 2k+1$,则有:

$$\hat{y}_i=\begin{cases} y_i[t_{i,k+1-i}+1:t_{i,k}+L],1\leq i\leq k-1, \\ y_i[1:t_{i,k}+L],k\leq i\leq n+1-k, \\ y_i[1:t_{i,n+1-i}+L],n+2-k\leq i\leq n. \end{cases}$$

若满足条件:$n<2k+1$,则有:

$$\hat{y}_i=\begin{cases} y_i[t_{i,k+1-i}+1:t_{i,k}+L],1\leq i\leq k-1, \\ y_i[t_{i,k+1-i}+1:t_{i,n+1-i}+L],n+1-k\leq i\leq k, \\ y_i[1:t_{i,n+1-i}+L],n+2-k\leq i\leq n. \end{cases}$$

其中,L表示所述数据块的数据长度,$\hat{y}_i$表示所述截取数据包。

5.一种分布式传输译码方法,其特征在于,包括:

连接并获取第一预设数量的节点数据,所述节点数据利用如权利要求1至4任一项所述的一种分布式存储方法存储在节点中;

对所述节点数据进行本地译码得到原始数据。

6.根据权利要求5所述的一种分布式传输译码方法，其特征在于，所述节点数据表示为：

$$\hat{x}_u = \begin{cases} \hat{y}_{i_u}[t_{i_u,u} - t_{i_u,k+1-i_u} + (1:L)], 1 \le i_u \le k-1, \\ \hat{y}_{i_u}[t_{i_u,u} - t_{i_u,1} + (1:L)], k \le i_u \le n. \end{cases}$$

其中，$\hat{x}_u$表示节点数据，$t_{i,j}$为非负整数，k表示第一预设数量，n表示第二预设数量，L表示所述数据块的数据长度，$\hat{y}_i$表示截取数据包。

7.根据权利要求6所述的一种分布式传输译码方法，其特征在于，所述本地译码的过程具体表示为：

步骤一：初始化译码向量$(l_1,...,l_k)$，所述节点数据表示为：$\hat{x}_u$且满足：$1 \leqslant u \leqslant k$；

步骤二：如果满足条件：$l_k < L$，则对于u从1到k，执行以下操作：

步骤三：如果满足条件$l_u < L$，且满足条件$u=1$或者$l_{u-1} > t_{i_u,u} - t_{i_u,u-1}$，则继续执行以下操作：

步骤四：将l_u更新为l_u+1；

步骤五：对于v从1到k，如果满足条件：$v \neq u$且$0 < l_u + t_{k+1-v,u} - t_{k+1-v,v} \leqslant L$，则执行：

$$\hat{x}_v[l_u + t_{k+1-v,u} - t_{k+1-v,v}] \leftarrow \hat{x}_v[l_u + t_{k+1-v,u} - t_{k+1-v,v}] \oplus \hat{x}_u[l_u]$$

重复执行步骤二到步骤五，直至得到每一个所述节点数据的原始数据。

8.一种共享秘密数据方法，在第二预设数量的设备中进行秘密数据共享，其特征在于：包括：

将所述秘密数据按照如权利要求1至4任一项所述的分布式存储方法进行分布式编码得到第二预设数量的数据包，每个所述设备中包括一个所述数据包；

任意选取第一预设数量的设备的所述数据包按照如权利要求5至7任一项所述的一种分布式传输译码方法进行本地译码得到所述秘密数据。

9.一种分布式数据处理设备，其特征在于，包括：

至少一个处理器；以及，与所述至少一个处理器通信连接的存储器；

其中，所述处理器通过调用所述存储器中存储的计算机程序，用于执行如权利要求1至4任一项所述的分布式存储方法或者如权利要求5至7任一项所述的一种分布式传输译码方法。

10.一种计算机可读存储介质，其特征在于，所述计算机可读存储介质存储有计算机可执行指令，所述计算机可执行指令用于使计算机执行如权利要求1至4任一项所述的分布式存储方法或者如权利要求5至7任一项所述的一种分布式传输译码方法。

分布式存储方法和传输译码方法

技术领域

[0001] 本发明涉及分布式存储领域，尤其是涉及一种分布式存储方法和传输译码方法。

背景技术

[0002] 在分布式存储系统中，MDS(最大距离可分)码是一种有效的存储编码方案，主要解决数据恢复问题。在一个分布式系统中，网络中有n个节点，数据分割为k个块，编码得到n个数据包分别保存在n个节点上，如果数据可以通过任意k个节点恢复，则该编码为(n,k)MDS码，相比较传统的备份机制，MDS码的编解码效率大大提高，但是MDS编解码复杂度较高。2013年，Sung等人提出了基于XOR操作(即异或操作)的MDS编解码方案，该方案适用于任意规模的网络，其采用ZigZag译码，编解码复杂度大大降低，但是这种方案的传输带宽冗余较多。2014年，Fu等人提出了基于XOR操作的无冗余传输和解码方案，将带宽传输冗余降为零，消除了译码空间冗余，但是其编码存储冗余仍然较高。高冗余会增加传输成本，并且随着网络节点数量增加，其存储冗余会随之增加。

[0003] 因此需要提出一种能够减少每个节点存储冗余，同时具有高效的编解码效率和较小译码空间开销的分布式存储方法和传输译码方法。

发明内容

[0004] 本发明旨在至少解决现有技术中存在的技术问题之一。为此，本发明提出一种分布式存储方法，能够减少每个节点存储冗余，同时具有高效的编解码效率。

[0005] 第一方面，本发明的一个实施例提供了：一种分布式存储方法，包括：

[0006] 对原始数据进行分块得到第一预设数量的数据块；

[0007] 根据生成矩阵对所述数据块进行编码，生成第二预设数量的数据包；

[0008] 对所述数据包利用截取方案进行数据截取得到截取数据包；

[0009] 存储所述截取数据包到对应的分布式节点中。

[0010] 进一步地，所述对原始数据进行分块得到第一预设数量的数据块具体包括：所述数据块的数据长度相同。

[0011] 进一步地，所述生成矩阵表示为：

[0012] $$\Psi=\begin{pmatrix} z^{t_{1,1}} & z^{t_{1,2}} & \cdots & z^{t_{1,k}} \\ z^{t_{2,1}} & z^{t_{2,2}} & \cdots & z^{t_{2,k}} \\ \cdots & \cdots & \cdots & \cdots \\ z^{t_{n,1}} & z^{t_{n,2}} & \cdots & z^{t_{n,k}} \end{pmatrix}$$

[0013] 所述数据包表示为：

[0014] $$\begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix} = \Psi \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_k \end{pmatrix}$$

[0015] $$y_i = \sum_{j=1}^{k} z^{t_{i,j}} x_j$$

[0016] 其中，$t_{i,j}$为非负整数，k表示第一预设数量，n表示第二预设数量，x_j表示数据块，y_i表示数据包，$z^{t_{i,j}} x_j$表示将x_j右移$t_{i,j}$个单位，前面补$t_{i,j}$个零。

[0017] 进一步地，所述截取方案表示为：

[0018] 若满足条件：n≥2k+1，则有：

[0019] $$\hat{y}_i = \begin{cases} y_i[t_{i,k+1-i}+1 : t_{i,k}+L], 1 \le i \le k-1, \\ y_i[1 : t_{i,k}+L], k \le i \le n+1-k, \\ y_i[1 : t_{i,n+1-i}+L], n+2-k \le i \le n. \end{cases}$$

[0020] 若满足条件：n＜2k+1，则有：

[0021] $$\hat{y}_i = \begin{cases} y_i[t_{i,k+1-i}+1 : t_{i,k}+L], 1 \le i \le k-1, \\ y_i[t_{i,k+1-i}+1 : t_{i,n+1-i}+L], n+1-k \le i \le k, \\ y_i[1 : t_{i,n+1-i}+L], n+2-k \le i \le n, \end{cases}$$

[0022] 其中，L表示所述数据块的数据长度，$\hat{y}_i$表示所述截取数据包。

[0023] 本发明实施例至少具有如下有益效果：存储时能够减少节点的存储冗余。

[0024] 第二方面，本发明的一个实施例提供了：一种分布式传输译码方法，包括：

[0025] 连接并获取第一预设数量的节点数据，所述节点数据利用如第一方面任一项所述的一种分布式存储方法存储在节点中；

[0026] 对所述节点数据进行本地译码得到原始数据。

[0027] 进一步地，所述节点数据表示为：

[0028] $$\hat{x}_u = \begin{cases} \hat{y}_{i_u}[t_{i_u,u} - t_{i_u,k+1-i_u} + (1:L)], 1 \le i_u \le k-1, \\ \hat{y}_{i_u}[t_{i_u,u} - t_{i_u,1} + (1:L)], k \le i_u \le n. \end{cases}$$

[0029] 其中，$\hat{x}_u$表示节点数据，$t_{i,j}$为非负整数，k表示第一预设数量，n表示第二预设数量，L表示所述数据块的数据长度，$\hat{y}_i$表示所述截取数据包。

[0030] 进一步地，所述本地译码过程具体表示为：

[0031] 步骤一：初始化译码向量$(l_1,\ldots,l_k)$，所述节点数据表示为：$\hat{x}_u$且满足：1≤u≤k；

[0032] 步骤二：如果满足条件：l_k＜L，则对于u从1到k，执行以下操作：

[0033] 步骤三：如果满足条件l_u＜L，且满足条件u＝1或者$l_{u-1} > t_{i_u,u} - t_{i_u,u-1}$，则继续执行以下操作：

[0034] 步骤四：将l_u更新为l_u+1；

[0035] 步骤五：对于v从1到k，如果满足条件：v≠u且0＜$l_u+t_{k+1-v,u}-t_{k+1-v,v}$≤L，则执行：

[0036] $\hat{x}_v[l_u + t_{k+1-v,u} - t_{k+1-v,v}] \leftarrow \hat{x}_v[l_u + t_{k+1-v,u} - t_{k+1-v,v}] \oplus \hat{x}_u[l_u]$

[0037] 重复执行步骤二到步骤五，直至得到每一个所述节点数据的原始数据。

[0038] 第三方面，本发明的一个实施例提供了：一种共享秘密数据方法，在第二预设数量的设备中进行秘密数据共享，其特征在于：包括：

[0039] 将所述秘密数据按照如第一方面任一项所述的分布式存储方法进行分布式编码得到第二预设数量的数据包，每个所述设备中包括一个所述数据包；

[0040] 任意选取第一预设数量的设备的所述数据包按照如第二方面任一项所述的一种分布式传输译码方法进行本地译码得到所述秘密数据。

[0041] 第四方面，本发明的一个实施例提供了：一种分布式数据处理设备，包括：

[0042] 至少一个处理器，以及与所述至少一个处理器通信连接的存储器；

[0043] 其中，所述处理器通过调用所述存储器中存储的计算机程序，用于执行如第一方面任一项所述的方法或第二方面任一项所述的方法。

[0044] 第五方面，本发明的一个实施例提供了：一种计算机可读存储介质，所述计算机可读存储介质存储有计算机可执行指令，所述计算机可执行指令用于使计算机执行如第一方面任一项所述的方法或第二方面任一项所述的方法。

[0045] 本发明的有益效果是：

[0046] 本发明的分布式存储方法通过对原始数据进行分块得到第一预设数量的数据块，然后根据生成矩阵对数据块进行编码，生成第二预设数量的数据包，对数据包利用截取方案进行数据截取得到截取数据包，存储截取数据包到对应的分布式节点中。根据本发明实施例的生成矩阵进行编码能够保证较高的编码效率，同时通过截取方案对数据包截取，存储时能够减少节点的存储冗余，从而保证整个分布式系统的存储冗余降低，节约了存储空间，提高了存储效率。

[0047] 应当理解的是，以上的一般描述和后文的细节描述仅是示例性和解释性的，并不能限制本公开。

附图说明

[0048] 此处的附图被并入说明书中并构成本说明书的一部分，示出了符合本公开的实施例，并与说明书一起用于解释本公开的原理。显而易见地，下面描述中的附图仅仅是本公开的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。在附图中：

[0049] 图1是本发明实施例中分布式存储方法的一具体实施例流程示意图；

[0050] 图2是本发明实施例中分布式传输译码方法的一具体实施例流程示意图。

具体实施方式

[0051] 为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对照附图说明本发明的具体实施方式。显而易见地，下面描述中的附图仅仅是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图，并获得其他的实施方式。

[0052] 除非另有定义，本文所使用的所有的技术和科学术语与属于本发明的技术领域的

技术人员通常理解的含义相同。本文中在本发明的说明书中所使用的术语只是为了描述具体的实施例的目的，不是旨在于限制本发明。

[0053] 附图中所示的流程图仅是示例性说明，不是必须包括所有的内容和操作/步骤，也不是必须按所描述的顺序执行。例如，有的操作/步骤还可以分解，而有的操作/步骤可以合并或部分合并，因此实际执行的顺序有可能根据实际情况改变。

[0054] 实施例一：

[0055] 本发明实施例一提供一种分布式存储方法。现有的在Sung和Fu的编码方案中，每个节点存储的信息均存在冗余，例如n个节点中，节点i的存储冗余为i(k-1)，其中最大冗余可以达到n(k-1)。随着网络中节点数量增加，存储冗余随之增加。而本实施例的分布式存储方法能够减少每个节点的存储冗余，同时具有高效的编码效率。

[0056] 图1为本发明实施例提供的一种分布式存储方法的流程示意图，如图1所示，该方法包括以下步骤：

[0057] S11：对原始数据进行分块得到第一预设数量k的数据块，数据块的数据长度相同。

[0058] S12：根据生成矩阵对数据块进行编码，生成第二预设数量n的数据包。

[0059] S13：对数据包利用截取方案进行数据截取得到截取数据包；

[0060] S14：存储截取数据包到对应的分布式节点中。

[0061] 下面详细描述上述步骤。

[0062] 本实施例在步骤S11中，假设原始数据的数据长度为kL比特，将其进行均匀大小分块得到k个长度为L比特的数据块，记为：$x_1,\ldots,x_k$，L表示该数据块的数据长度。

[0063] 步骤S12中，生成矩阵$\Psi=(z^{t_{i,j}})$为一个n×k的矩阵，其中，$t_{i,j}$为非负整数，生成矩阵Ψ可以表示为：

[0064] $$\Psi=\begin{pmatrix} z^{t_{1,1}} & z^{t_{1,2}} & \cdots & z^{t_{1,k}} \\ z^{t_{2,1}} & z^{t_{2,2}} & \cdots & z^{t_{2,k}} \\ \cdots & \cdots & \cdots & \cdots \\ z^{t_{n,1}} & z^{t_{n,2}} & \cdots & z^{t_{n,k}} \end{pmatrix} \quad (1)$$

[0065] 其中，生成矩阵Ψ满足递增性质，表示为：$\forall i_2>i_1, j_2>j_1,\ t_{i_2,j_2}-t_{i_2,j_1}>t_{i_1,j_2}-t_{i_1,j_1}\geq 0$，其中“=”只有在满足条件：$i_1=1$时成立。

[0066] 进一步地，如果满足条件：$t_{i,j}=i(j-1)$，生成矩阵Ψ为范德蒙矩阵。

[0067] 利用生成矩阵Ψ对k个数据块进行编码得到n个数据包，表示为$y_1,\ldots,y_n$，编码过程表示为：

[0068] $$\begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix}=\Psi\begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_k \end{pmatrix} \quad (2)$$

[0069] 即每个编码之后的数据包都能表示为：

[0070] $$y_i=\sum_{j=1}^{k} z^{t_{i,j}}x_j \quad (3)$$

[0071] 其中，x_j表示数据块，y_i表示数据包，$z^{t_{i,j}}x_j$表示将x_j右移$t_{i,j}$个单位，前面补$t_{i,j}$个零。

[0072] 为了便于下面描述，对于上述数据包y_i，$1\leqslant i\leqslant n$，用$y_i[l]$表示数据包y_i的第l位的元素，用$y_i[l_1:l_2]$表示数据包y_i的第l_1位到第l_2位的元素，用$y_i[l+(l_1:l_2)]$表示$y_i[l+l_1:l+l_2]$。

[0073] 传统的编码方案中，将编码后的数据直接存储，从而会产生较大的存储冗余，本实施例在存储数据之前对n个数据包进行截取操作，通过截取方案保证任取k个截取后的截取数据包都是可以恢复待存储的原始数据。

[0074] 根据系统参数n和L，本实施例步骤S13中的截取方案表示为：

[0075] 若满足条件：$n\geqslant 2k+1$，则有：

[0076] $$\hat{y}_i=\begin{cases} y_i[t_{i,k+1-i}+1:t_{i,k}+L],1\leq i\leq k-1,\\ y_i[1:t_{i,k}+L],k\leq i\leq n+1-k,\\ y_i[1:t_{i,n+1-i}+L],n+2-k\leq i\leq n. \end{cases} \tag{4}$$

[0077] 若满足条件：$n<2k+1$，则有：

[0078] $$\hat{y}_i=\begin{cases} y_i[t_{i,k+1-i}+1:t_{i,k}+L],1\leq i\leq k-1,\\ y_i[t_{i,k+1-i}+1:t_{i,n+1-i}+L],n+1-k\leq i\leq k,\\ y_i[1:t_{i,n+1-i}+L],n+2-k\leq i\leq n. \end{cases} \tag{5}$$

[0079] 其中，L表示数据块的数据长度，$\hat{y}_i$表示截取数据包。

[0080] 与传统的编码方案相比，本实施例存储方案中每个节点至少可以减少$\frac{1}{2}nk(k-1)$的存储冗余，从而降低系统的存储冗余，同时编码过程只有XOR操作(异或操作)和shift操作(移位操作)，相关的译码不需要额外的空间开销，可以实现无冗余传输和本地译码。

[0081] 本实施例根据生成矩阵对原始数据进行编码能够保证较高的编码效率，同时通过截取方案对数据包截取，存储时能够减少节点的存储冗余，从而保证整个分布式系统的存储冗余降低，节约了存储空间，提高了存储效率。

[0082] 可以理解的是，本实施例除了可以用在分布式存储系统中之外，还可以应用在区块链或者秘密共享等应用领域，在此不做领域限定。

[0083] 例如：在云存储应用中，由于数据分块并存储在分布式节点中，当前的主要方案有备份机制和RS编码(如谷歌文件系统)，但是备份机制存储效率低下，RS码的编解码复杂度较高，因此可以采用本实施例的方案提高存储效率和降低编解码复杂度。

[0084] 又比如：在以存储证明为背景的区块链应用系统中，其中的数据存储仍然以备份和RS编码为主，也可以采用该发明方案来节省服务器的存储和存储证明的复杂度。

[0085] 实施例二：

[0086] 本实施例提供一种分布式传输译码方法，用于对实施例一中分布式系统节点中存储的数据进行译码，如图2所示，为本实施例的分布式传输译码方法流程示意图，包括：

[0087] S21：连接并获取第一预设数量k的节点数据，其中节点数据利用如实施例一任一项所述的一种分布式存储方法存储在节点中。

[0088] 即在传输阶段，用户可以随机选取k个节点取其节点的存储数据，节点记为$i_1,...,i_k$，满足条件：$n \geq i_1 > ... > i_k \geq 1$，将节点中的节点数据包$\hat{x}_u$传输给用户。

[0089] S22：对节点数据进行本地译码得到原始数据。

[0090] 本实施例中，采用本地译码，能够消除额外的译码空间消耗。

[0091] 具体的译码过程如下所示。

[0092] 首先，将节点数据表示为：

[0093] $$\hat{x}_u = \begin{cases} \hat{y}_{i_u}[t_{i_u,u} - t_{i_u,k+1-i_u} + (1:L)], 1 \leq i_u \leq k-1, \\ \hat{y}_{i_u}[t_{i_u,u} - t_{i_u,1} + (1:L)], k \leq i_u \leq n. \end{cases} \quad (6)$$

[0094] 其中，$\hat{x}_u$表示节点数据，$t_{i,j}$为非负整数，k表示第一预设数量，n表示第二预设数量，L表示数据块的数据长度，$\hat{y}_i$表示截取数据包。

[0095] 用户得到节点数据$\hat{x}_u$，($1 \leq u \leq k$)后，对进行$\hat{x}_u$本地译码，可以得到原始数据$x_1,...,x_k$，本实施例中本地译码过程表示为：

[0096] 步骤一：初始化译码向量$(l_1,...,l_k)$，节点数据表示为：$\hat{x}_u$且满足：$1 \leq u \leq k$；

[0097] 步骤二：如果满足条件：$l_k < L$，则对于u从1到k，执行以下操作，否则本地译码过程结束。

[0098] 步骤三：如果满足条件：$l_u < L$且$u=1$，或者，$l_{u-1} > t_{i_u,u} - t_{i_u,u-1}$，则继续执行以下操作，否则回到步骤二。

[0099] 步骤四：将l_u更新为l_u+1；

[0100] 步骤五：对于v从1到k，如果满足条件：$v \neq u$且$0 < l_u + t_{k+1-v,u} - t_{k+1-v,v} \leq L$，则执行：

[0101] $$\hat{x}_v[l_u + t_{k+1-v,u} - t_{k+1-v,v}] \leftarrow \hat{x}_v[l_u + t_{k+1-v,u} - t_{k+1-v,v}] \oplus \hat{x}_u[l_u] \quad (7)$$

[0102] 重复执行步骤二到步骤五，直至得到每一个节点数据的原始数据。

[0103] 本实施例的译码方法的复杂度为$k(k-1)L$，即译码一个单位长度的数据，需要k-1次的XOR操作，该方式的译码复杂度低于Sung的方案，与Fu的方案一样，但是本实施例的本地译码，能够消除额外的译码空间开销。

[0104] 实施例三：

[0105] 本实施例针对实施例一和实施例二，提出一种具体的应用场景，提供一种共享秘密数据方法，在第二预设数量的设备中进行秘密数据共享，包括：

[0106] 将秘密数据按照如实施例一任一项的分布式存储方法进行分布式编码得到第二预设数量的数据包，每个设备中包括一个数据包。

[0107] 任意选取第一预设数量的设备的数据包按照如实施例二任一项的一种分布式传输译码方法进行本地译码得到秘密数据，其中秘密数据还包括例如权限、秘钥等。

[0108] 即在有n个用户参与的秘密共享方案中，将秘密数据分成k个数据块并用如实施例一所述的方法进行编码得到n个数据包，每个用户持有1个数据包，则任意的k个用户可以通过实施例二所述的译码方式进行本地译码而共享秘密，从而实现了基于XOR操作的秘密共享方案。

[0109] 另外，本发明还提供分布式数据处理设备，包括：

[0110] 至少一个处理器，以及与所述至少一个处理器通信连接的存储器；

[0111] 其中，所述处理器通过调用所述存储器中存储的计算机程序，用于执行如实施例一所述的方法。计算机程序即程序代码，当程序代码在分布式数据处理上运行时，程序代码用于使设备执行本说明书上述如实施例一任一项所述的分布式存储方法或者如实施例二任一项所述的一种分布式传输译码方法中的步骤。

[0112] 另外，本发明还提供一种计算机可读存储介质，计算机可读存储介质存储有计算机可执行指令，其中计算机可执行指令用于使计算机执行如实施例一所述的方法。

[0113] 不失一般性，所述计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括以用于存储诸如计算机可读指令、数据结构、程序模块或其他数据等信息的任何方法或技术实现的易失性和非易失性、可移动和不可移动介质。计算机存储介质包括RAM、ROM、EPROM、EEPROM、闪存或其他固态存储其技术，CD-ROM、DVD或其他光学存储、磁带盒、磁带、磁盘存储或其他磁性存储设备。当然，本领域技术人员可知所述计算机存储介质不局限于上述几种。

[0114] 本发明实施例中根据生成矩阵进行编码能够保证较高的编码效率，同时通过截取方案对数据包截取，存储时能够减少节点的存储冗余，从而保证整个分布式系统的存储冗余降低，节约了存储空间，提高了存储效率，同时，本发明实施例的译码方案采用本地译码，能够消除额外的译码空间开销。

[0115] 以上各实施例仅用以说明本发明的技术方案，而非对其限制，尽管参照前述各实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分或者全部技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本发明各实施例技术方案的范围，其均应涵盖在本发明的权利要求和说明书的范围当中。

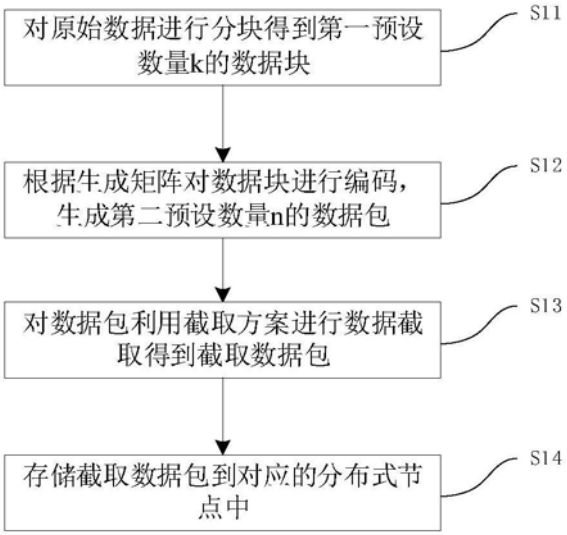

对原始数据进行分块得到第一预设数量k的数据块
S11
根据生成矩阵对数据块进行编码，生成第二预设数量n的数据包
S12
对数据包利用截取方案进行数据截取得到截取数据包
S13
存储截取数据包到对应的分布式节点中
S14

图1

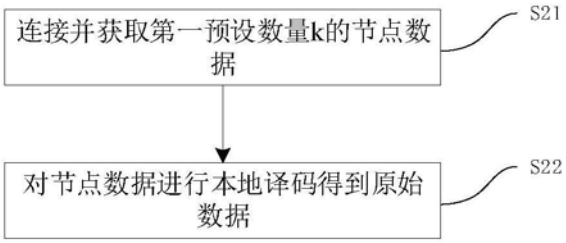

连接并获取第一预设数量k的节点数据
S21
对节点数据进行本地译码得到原始数据
S22

图2