



(12) 发明专利

(10) 授权公告号 CN 108287666 B

(45) 授权公告日 2021.01.26

(21) 申请号 201810042737.8

(22) 申请日 2018.01.16

(65) 同一申请的已公布的文献号
申请公布号 CN 108287666 A

(43) 申请公布日 2018.07.17

(73) 专利权人 中国人民公安大学
地址 100045 北京市西城区木樨地南里1号

(72) 发明人 李欣 徐建鹏 孙海春 赵晓凡

(74) 专利代理机构 北京超凡志成知识产权代理
事务所(普通合伙) 11371
代理人 金相允

(51) Int.Cl.
G06F 3/06 (2006.01)

(56) 对比文件

CN 106484512 A, 2017.03.08

CN 107239336 A, 2017.10.10

US 2017344900 A1, 2017.11.30

CN 1450493 A, 2003.10.22

CN 106611277 A, 2017.05.03

CN 106161552 A, 2016.11.23

CN 104899100 A, 2015.09.09

肖博. 云存储调度关键技术研究与实践.《中国优秀硕士学位论文全文数据库》.2017,

审查员 刘展

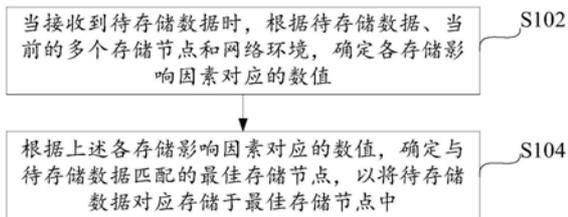
权利要求书3页 说明书12页 附图2页

(54) 发明名称

用于云存储环境的数据存储方法及装置

(57) 摘要

本发明提供了一种用于云存储环境的数据存储方法及装置,涉及云存储技术领域,该方法包括:当接收到待存储数据时,根据待存储数据、当前的多个存储节点和网络环境,确定各存储影响因素对应的数值;存储影响因素包括:待存储数据的数据量、待存储数据与存储节点之间的传输距离、存储节点的故障率、存储节点的剩余空间、存储节点的内存占用率、存储节点对应的网络带宽或存储节点对应的网络延迟;根据各存储影响因素对应的数值,确定与待存储数据匹配的最佳存储节点,以将待存储数据对应存储于最佳存储节点中。本发明能够基于多因素考虑而为待存储数据合理匹配存储节点,提供较好的数据存储方案。



1. 一种用于云存储环境的数据存储方法,其特征在于,包括:

当接收到待存储数据时,根据所述待存储数据、当前的多个存储节点和网络环境,确定各存储影响因素对应的数值;所述存储影响因素包括:待存储数据的数据量、待存储数据与存储节点之间的传输距离、存储节点的故障率、存储节点的剩余空间、存储节点的内存占用率、存储节点对应的网络带宽或存储节点对应的网络延迟;

通过三角模糊层次分析法确定各所述存储影响因素对应的权重;

根据各所述存储影响因素对应的数值和权重,基于遗传算法确定与所述待存储数据匹配的最佳存储节点,以将所述待存储数据对应存储于最佳存储节点中。

2. 根据权利要求1所述的方法,其特征在于,所述根据各所述存储影响因素对应的数值和权重,基于遗传算法确定与所述待存储数据匹配的最佳存储节点的步骤,包括:

对各所述存储影响因素对应的数值进行归一化处理;

将归一化处理后的各所述存储影响因素对应的数值、各所述存储影响因素对应的权重代入至预先建立的遗传目标函数,求解获得待存储数据与最佳存储节点的对应关系。

3. 根据权利要求2所述的方法,其特征在于,所述遗传目标函数的建立过程包括:

设置待存储数据的序列为 $X = \{x_1, x_2, \dots, x_m\}$,当前可用的存储节点的序列为 $Y = \{y_1, y_2, \dots, y_m\}$,建立存储方案矩阵 M_k 如下:

$$M_k = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_m \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_m \\ \cdots & \cdots & \cdots & \cdots \\ x_n y_1 & x_n y_2 & \cdots & x_n y_m \end{bmatrix}$$

且 M_k 需满足 $\sum_{j=1}^m x_i y_j = 1$;

其中, $M_k^{ij} = x_i y_j$,当 $x_i y_j = 1$ 时,表示 x_i 存储在 y_j 中;当 $x_i y_j = 0$ 时,表示 x_i 未存储在 y_j 中;

基于存储方案矩阵 M_k ,构造以下遗传目标函数 Z_k :

$$\begin{cases} Z_k = M_k \otimes P^T + \sum_{i=1}^n \sum_{j=1}^m x_i y_j \cdot D_j \cdot w_4 + Size \cdot w_5 \\ P = [P_1, P_2, \dots, P_m] \\ P_j = Space_j \cdot w_2 + Mem_j \cdot w_3 + NetB_j \cdot w_6 + NetD_j \cdot w_7 + Fail_i \cdot w_1 \end{cases}$$

且 Z_k 需满足 $\sum_{j=1}^m \sum_{i=1}^n x_i y_j \cdot Size_i \leq Capacity_j$;

其中,矩阵 P 为各所述存储节点对应的代价矩阵, D_j 为所述待存储数据所在的数据中心根节点与第 j 个存储节点之间的传输距离, $Size_i$ 为归一化处理后的第 i 个待存储数据的数据量, $Capacity_j$ 为归一化处理后的第 j 个存储节点的初始空间, $Space_j$ 为归一化处理后的第 j 个存储节点的剩余空间, Mem_j 为归一化处理后的第 j 个存储节点的内存使用率, $NetB_j$ 为归

一化处理后的第j个存储节点对应的网络带宽, $NetD_j$ 为归一化处理后的第j个存储节点对应的网络延迟, $Fail_j$ 为归一化处理后的第j个存储节点的故障率, w_1 为存储节点的故障率对应的权重, w_2 为存储节点的剩余空间对应的权重, w_3 为存储节点的内存使用率对应的权重, w_4 为待存储数据的传输距离对应的权重, w_5 为待存储数据的数据量对应的权重, w_6 为存储节点的网络带宽对应的权重, w_7 为存储节点的网络延迟对应的权重。

4. 根据权利要求3所述的方法, 其特征在于, 所述求解获得待存储数据与最佳存储节点的对应关系的步骤, 包括:

在求解所述遗传目标函数的过程中, 对 M_k 进行变异处理, 以实现所述遗传算法中的种群进化; 其中, 设定 M_k 为所述遗传算法中的染色体;

在所述种群进化的过程中, 采用多次迭代的方式计算得到最优存储方案矩阵 M^{AG} ; 其中, 每次迭代均选取适应度排名在预设范围内的优质染色体 M_N^G 作为在下次迭代过程中进行计算的 M_k , 并淘汰适应度排名在预设范围外的致死染色体 M_M^B ;

根据所述最优存储方案矩阵 M^{AG} , 确定待存储数据与最佳存储节点的对应关系。

5. 根据权利要求4所述的方法, 其特征在于, 所述变异处理的方式包括:

随机从存储方案矩阵 M_k 中挑选 M_k^{ij} ;

对于挑选得到的 M_k^{ij} , 在满足 $\sum_{j=1}^m x_i y_j = 1$ 的基础上, 如果 $M_k^{ij} = 1$, 调整为 $M_k^{ij} = 0$; 如果 $M_k^{ij} = 0$, 调整为 $M_k^{ij} = 1$ 。

6. 根据权利要求4所述的方法, 其特征在于, 所述根据所述最优存储方案矩阵 M^{AG} , 确定待存储数据与最佳存储节点的对应关系的步骤, 包括:

根据所述最优存储方案矩阵 M^{AG} 中各元素的值, 确定待存储数据的序列为 $X = \{x_1, x_2, \dots, x_m\}$ 与当前可用的存储节点的序列为 $Y = \{y_1, y_2, \dots, y_m\}$ 的存储匹配关系 $K = (k_1, k_2, \dots, k_m)$; 在存储匹配关系 K 中, m 为待存储数据的序列号, k_m 为序列号为 m 的待存储数据对应的最佳存储节点的序列号。

7. 根据权利要求3所述的方法, 其特征在于,

设定在待存储数据的序列 $X = \{x_1, x_2, \dots, x_m\}$ 中, $x_i = \{s_i, m_i, n_i\}$, s_i 为第 i 个待存储数据所需占用的存储空间, m_i 为第 i 个待存储数据所需占用的内存量, n_i 为第 i 个待存储数据所需占用的网络资源;

设定在存储节点的序列为 $Y = \{y_1, y_2, \dots, y_m\}$ 中, $y_i = \{S_i, M_i, N_i\}$, S_i 为第 i 个存储节点的存储空间, M_i 为第 i 个存储节点的内存资源, N_i 为第 i 个存储节点的网络资源;

假设第 i 个存储节点对应存储第 a 个待存储数据、第 b 个待存储数据和第 c 个待存储数据, 所述存储方案矩阵 M_k 需满足如下约束条件:

$$S_i \geq s_a + s_b + \dots + s_c$$

$$M_i \geq m_a + m_b + \dots + m_c$$

$$N_i \geq n_a + n_b + \dots + n_c$$

且求解得到的存储方案矩阵 M_k 满足 $T = \min \sum_{i=1}^m (t_i(s) + t_i(m) + t_i(n))$;

其中 $t_i(s)$ 为在数据存储过程中由待存储数据产生的耗时, $t_i(m)$ 在数据存储过程中由存储节点产生的耗时, $t_i(n)$ 为在数据存储过程中由网络环境产生的耗时。

8. 一种用于云存储环境的数据存储装置,其特征在于,包括:

因素确定模块,用于当接收到待存储数据时,根据所述待存储数据、当前的多个存储节点和网络环境,确定各存储影响因素对应的数值;所述存储影响因素包括:待存储数据的数据量、待存储数据与存储节点之间的传输距离、存储节点的故障率、存储节点的剩余空间、存储节点的内存占用率、存储节点对应的网络带宽或存储节点对应的网络延迟;

最佳节点确定模块,用于通过三角模糊层次分析法确定各所述存储影响因素对应的权重,根据各所述存储影响因素对应的数值和权重,基于遗传算法确定与所述待存储数据匹配的最佳存储节点,以将所述待存储数据对应存储于最佳存储节点中。

9. 一种计算机可读存储介质,所述计算机可读存储介质上存储有计算机程序,其特征在于,所述计算机程序被处理器运行时执行上述权利要求1至7任一项所述的方法的步骤。

用于云存储环境的数据存储方法及装置

技术领域

[0001] 本发明涉及云存储技术领域,尤其是涉及一种用于云存储环境的数据存储方法及装置。

背景技术

[0002] 云存储环境下的资源调度方案,主要是提供一种优选数据存储方案,也即能够将待存储数据合理的存储于存储节点中,以期得到最小的存储耗时。传统的云存储环境下的资源调度方法主要基于存储节点的剩余空间(如Openstack的Cinder块存储)来为数据选择合适的存储节点,也即为待存储数据实现合理的存储节点调度。然而,这种仅考虑存储节点的剩余空间来进行数据存储的方式,考虑因素单一,致使最终所得的数据存储方案不佳。

发明内容

[0003] 有鉴于此,本发明的目的在于提供一种用于云存储环境的数据存储方法及装置,能够基于多因素考虑而为待存储数据合理匹配存储节点,提供较好的数据存储方案。

[0004] 为了实现上述目的,本发明实施例采用的技术方案如下:

[0005] 第一方面,本发明实施例提供了一种用于云存储环境的数据存储方法,包括:

[0006] 当接收到待存储数据时,根据所述待存储数据、当前的多个存储节点和网络环境,确定各存储影响因素对应的数值;所述存储影响因素包括:待存储数据的数据量、待存储数据与存储节点之间的传输距离、存储节点的故障率、存储节点的剩余空间、存储节点的内存占用率、存储节点对应的网络带宽或存储节点对应的网络延迟;

[0007] 根据各所述存储影响因素对应的数值,确定与所述待存储数据匹配的最佳存储节点,以将所述待存储数据对应存储于最佳存储节点中。

[0008] 结合第一方面,本发明实施例提供了第一方面的第一种可能的实施方式,其中,所述根据各所述存储影响因素对应的数值,确定与所述待存储数据匹配的最佳存储节点的步骤,包括:

[0009] 通过三角模糊层次分析法确定各所述存储影响因素对应的权重;

[0010] 根据各所述存储影响因素对应的数值和权重,基于遗传算法确定与所述待存储数据匹配的最佳存储节点。

[0011] 结合第一方面的第一种可能的实施方式,本发明实施例提供了第一方面的第二种可能的实施方式,其中,所述根据各所述存储影响因素对应的数值和权重,基于遗传算法确定与所述待存储数据匹配的最佳存储节点的步骤,包括:

[0012] 对各所述存储影响因素对应的数值进行归一化处理;

[0013] 将归一化处理后的各所述存储影响因素对应的数值、各所述存储影响因素对应的权重代入至预先建立的遗传目标函数,求解获得待存储数据与最佳存储节点的对应关系。

[0014] 结合第一方面的第二种可能的实施方式,本发明实施例提供了第一方面的第三种可能的实施方式,其中,所述遗传目标函数的建立过程包括:

[0015] 设置待存储数据的序列为 $X = \{x_1, x_2, \dots, x_m\}$, 当前可用的存储节点的序列为 $Y = \{y_1, y_2, \dots, y_m\}$, 建立存储方案矩阵 M_k 如下:

$$[0016] \quad M_k = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_m \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_m \\ \cdots & \cdots & \cdots & \cdots \\ x_n y_1 & x_n y_2 & \cdots & x_n y_m \end{bmatrix}$$

[0017] 且 M_k 需满足 $\sum_{j=1}^m x_i y_j = 1$;

[0018] 其中, $M_k^{ij} = x_i y_j$, 当 $x_i y_j = 1$ 时, 表示 x_i 存储在 y_j 中; 当 $x_i y_j = 0$ 时, 表示 x_i 未存储在 y_j 中;

[0019] 基于存储方案矩阵 M_k , 构造以下遗传目标函数 Z_k :

$$[0020] \quad \begin{cases} Z_k = M_k \otimes P^T + \sum_{i=1}^n \sum_{j=1}^m x_i y_j \cdot D_j \cdot w_4 + Size \cdot w_5 \\ P = [P_1, P_2, \dots, P_m] \\ P_j = Space_j \cdot w_2 + Mem_j \cdot w_3 + NetB_j \cdot w_6 + NetD_j \cdot w_7 + Fail_j \cdot w_1 \end{cases}$$

[0021] 且 Z_k 需满足 $\sum_{j=1}^m \sum_{i=1}^n x_i y_j \cdot Size_i \leq Capacity_j$;

[0022] 其中, 矩阵 P 为各所述存储节点对应的代价矩阵, D_j 为所述待存储数据所在的数据中心根节点与第 j 个存储节点之间的传输距离, $Size_i$ 为归一化处理后的第 i 个待存储数据的数据量, $Capacity_j$ 为归一化处理后的第 j 个存储节点的初始空间, $Space_j$ 为归一化处理后的第 j 个存储节点的剩余空间, Mem_j 为归一化处理后的第 j 个存储节点的内存使用率, $NetB_j$ 为归一化处理后的第 j 个存储节点对应的网络带宽, $NetD_j$ 为归一化处理后的第 j 个存储节点对应的网络延迟, $Fail_j$ 为归一化处理后的第 j 个存储节点的故障率, w_1 为存储节点的故障率对应的权重, w_2 为存储节点的剩余空间对应的权重, w_3 为存储节点的内存使用率对应的权重, w_4 为待存储数据的传输距离对应的权重, w_5 为待存储数据的数据量对应的权重, w_6 为存储节点的网络带宽对应的权重, w_7 为存储节点的网络延迟对应的权重。

[0023] 结合第一方面的第三种可能的实施方式, 本发明实施例提供了第一方面的第四种可能的实施方式, 其中, 所述求解获得待存储数据与最佳存储节点的对应关系的步骤, 包括:

[0024] 在求解所述遗传目标函数的过程中, 对 M_k 进行变异处理, 以实现所述遗传算法中的种群进化; 其中, 设定 M_k 为所述遗传算法中的染色体;

[0025] 在所述种群进化的过程中, 采用多次迭代的方式计算得到最优存储方案矩阵 M_N^{AG} ; 其中, 每次迭代均选取适应度排名在预设范围内的优质染色体 M_N^G 作为在下次迭代过程中

进行计算的 M_k ,并淘汰适应度排名在预设范围外的致死染色体 M_M^B ;

[0026] 根据所述最优存储方案矩阵 M^{AG} ,确定待存储数据与最佳存储节点的对应关系。

[0027] 结合第一方面的第四种可能的实施方式,本发明实施例提供了第一方面的第五种可能的实施方式,其中,所述变异处理的方式包括:

[0028] 随机从存储方案矩阵 M_k 中挑选 M_k^{ij} ;

[0029] 对于挑选得到的 M_k^{ij} ,在满足 $\sum_{j=1}^m x_i y_j = 1$ 的基础上,如果 $M_k^{ij} = 1$,调整为 $M_k^{ij} = 0$;如果 $M_k^{ij} = 0$,调整为 $M_k^{ij} = 1$ 。

[0030] 结合第一方面的第四种可能的实施方式,本发明实施例提供了第一方面的第六种可能的实施方式,其中,所述根据所述最优存储方案矩阵 M^{AG} ,确定待存储数据与最佳存储节点的对应关系的步骤,包括:

[0031] 根据所述最优存储方案矩阵 M^{AG} 中各元素的值,确定待存储数据的序列为 $X = \{x_1, x_2, \dots, x_m\}$ 与当前可用的存储节点的序列为 $Y = \{y_1, y_2, \dots, y_m\}$ 的存储匹配关系 $K = (k_1, k_2, \dots, k_m)$;在存储匹配关系 K 中, m 为待存储数据的序列号, k_m 为序列号为 m 的待存储数据对应的最佳存储节点的序列号。

[0032] 结合第一方面的第四种可能的实施方式,本发明实施例提供了第一方面的第七种可能的实施方式,其中,设定在待存储数据的序列 $X = \{x_1, x_2, \dots, x_m\}$ 中, $x_i = \{s_i, m_i, n_i\}$, s_i 为第 i 个待存储数据所需占用的存储空间, m_i 为第 i 个待存储数据所需占用的内存量, n_i 为第 i 个待存储数据所需占用的网络资源;

[0033] 设定在存储节点的序列为 $Y = \{y_1, y_2, \dots, y_m\}$ 中, $y_i = \{S_i, M_i, N_i\}$, S_i 为第 i 个存储节点的存储空间, M_i 为第 i 个存储节点的内存资源, N_i 为第 i 个存储节点的网络资源;

[0034] 假设第 i 个存储节点对应存储第 a 个待存储数据、第 b 个待存储数据和第 c 个待存储数据,所述存储方案矩阵 M_k 需满足如下约束条件:

[0035] $S_i \geq s_a + s_b + \dots + s_c$

[0036] $M_i \geq m_a + m_b + \dots + m_c$

[0037] $N_i \geq n_a + n_b + \dots + n_c$

[0038] 且求解得到的存储方案矩阵 M_k 满足 $T = \min \sum_{i=1}^m (t_i(s) + t_i(m) + t_i(n))$;

[0039] 其中 $t_i(s)$ 为在数据存储过程中由待存储数据产生的耗时, $t_i(m)$ 在数据存储过程中由存储节点产生的耗时, $t_i(n)$ 为在数据存储过程中由网络环境产生的耗时。

[0040] 第二方面,本发明实施例还提供一种用于云存储环境的数据存储装置,包括:

[0041] 因素确定模块,用于当接收到待存储数据时,根据所述待存储数据、当前的多个存储节点和网络环境,确定各存储影响因素对应的数值;所述存储影响因素包括:待存储数据的数据量、待存储数据与存储节点之间的传输距离、存储节点的故障率、存储节点的剩余空间、存储节点的内存占用率、存储节点对应的网络带宽或存储节点对应的网络延迟;

[0042] 最佳节点确定模块,用于根据各所述存储影响因素对应的数值,确定与所述待存储数据匹配的最佳存储节点,以将所述待存储数据对应存储于最佳存储节点中。

[0043] 第三方面,本发明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上存储有计算机程序,所述计算机程序被处理器运行时执行第一方面任一项所述的方法的步骤。

[0044] 本发明实施例提供了一种用于云存储环境的数据存储方法及装置,在接收到待存储数据时,能够根据待存储数据、当前的多个存储节点和网络环境确定各存储影响因素对应的数值,进而根据各存储影响因素对应的数值,确定与待存储数据匹配的最佳存储节点。这种基于多种存储影响因素的综合考虑而选取合适的存储节点的方式,考虑因素较为全面,能够基于全局进行考虑,以提供较好的数据存储方案。

[0045] 本公开的其他特征和优点将在随后的说明书中阐述,或者,部分特征和优点可以从说明书推知或毫无疑义地确定,或者通过实施本公开的上述技术即可得知。

[0046] 为使本发明的上述目的、特征和优点能更明显易懂,下文特举较佳实施例,并配合所附附图,作详细说明如下。

附图说明

[0047] 为了更清楚地说明本发明具体实施方式或现有技术中的技术方案,下面将对具体实施方式或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图是本发明的一些实施方式,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。

[0048] 图1示出了本发明实施例所提供的一种用于云存储环境的数据存储方法的流程图;

[0049] 图2示出了本发明实施例所提供的一种云存储环境的存储影响因素的示意图;

[0050] 图3示出了本发明实施例所提供的另一种用于云存储环境的数据存储方法的示意图;

[0051] 图4示出了本发明实施例所提供的一种用于云存储环境的数据存储装置的结构示意图;

[0052] 图5示出了本发明实施例所提供的一种终端的结构示意图。

具体实施方式

[0053] 为使本发明实施例的目的、技术方案和优点更加清楚,下面将结合附图对本发明的技术方案进行清楚、完整地描述,显然,所描述的实施例是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0054] 云存储环境下的资源调度方案的存储最小耗时问题主要指对同一时刻到达数据中心的待存储数据,采用合适的策略将其存储到存储节点中,以期得到最小的存储耗时。其中,资源调度的过程又可理解为选取现有资源(存储节点)以存储数据的过程。在传统的云存储环境下的资源调度方法中,主要基于存储节点的剩余空间来为待存储数据匹配存储节点,资源调度决策的考虑因素太过单一,这将导致最终所得的数据存储方案只能陷入一个局部最优解,无法以全局的角度进行综合分析,也即无法得到全局最优解,因此现有的数据存储方法(也即,资源调度方法)所得结果并非最优,存储耗时仍旧较长。此外,由于云存储

环境下的资源调度也是一种复杂的离散组合最优化问题,在高复杂度的解空间中,使用现有仅基于存储节点的剩余空间的资源调度方法难以找到一个多项式时间算法来求其最优解,因此这种基于剩余空间进行资源调度的方法也较不合理。

[0055] 为改善上述问题,本发明实施例提供一种用于云存储环境的数据存储方法及装置,也可理解为本发明实施例提供了一种云存储环境下的数据存储方法及装置,该技术可应用于云存储场合中,以下对本发明实施例进行详细介绍。

[0056] 实施例一:

[0057] 参见图1所示的一种用于云存储环境的数据存储方法的流程图,该方法可以由终端执行,终端可以为计算机、上位机、服务器等智能终端,包括以下步骤:

[0058] 步骤S102,当接收到待存储数据时,根据待存储数据、当前的多个存储节点和网络环境,确定各存储影响因素对应的数值;存储影响因素包括:待存储数据的数据量、待存储数据与存储节点之间的传输距离、存储节点的故障率、存储节点的剩余空间、存储节点的内存占用率、存储节点对应的网络带宽或存储节点对应的网络延迟。其中,存储影响因素又可称为资源调度影响因素,数据量也即数据大小,剩余空间也即剩余存储空间。

[0059] 在本实施例中,主要考虑存储节点、待存储数据和网络环境三大指标,其中,参见图2所示的一种云存储环境的存储影响因素的示意图,示意出选取最佳存储节点需考虑存储节点、待存储数据和网络环境三大指标,且存储节点指标对应的存储影响因素为故障率、剩余空间和内存占用率,待存储数据指标对应的存储影响因素为数据大小和数据距离,网络环境指标对应的存储影响因素为网络带宽和网络延迟。

[0060] 在具体应用中,可以根据实际需要而灵活选取上述存储影响因素中的多个作为数据存储中的参考因素,在此不再赘述。

[0061] 步骤S104,根据上述各存储影响因素对应的数值,确定与待存储数据匹配的最佳存储节点,以将待存储数据对应存储于最佳存储节点中。

[0062] 在一种实施方式中,可以通过三角模糊层次分析法确定各存储影响因素对应的权重;根据各存储影响因素对应的数值和权重,基于遗传算法确定与待存储数据匹配的最佳存储节点。

[0063] 上述待存储数据可以为多个,也即多个待存储数据同时到达数据中心时,可以利用上述数据存储方法为多个待存储数据调度相应的最佳存储节点进行存储。

[0064] 本实施例的上述方法中,在接收到待存储数据时,能够根据待存储数据、当前的多个存储节点和网络环境确定各存储影响因素对应的数值,进而根据各存储影响因素对应的数值,确定与待存储数据匹配的最佳存储节点。这种基于多种存储影响因素的综合考虑而选取合适的存储节点的方式,考虑因素较为全面,能够基于全局进行考虑,得到全局最优解,存储耗时更少,得到的数据存储方案更加可靠,以便实现更好的负载平衡,使云资源存储系统能够稳定运行。

[0065] 三角模糊层次分析法是一种定性与定量相结合的系统分析方法,能够量化评价指标,为最优方案的选择提供了依据,对同一层次(等级)的要素进行两两比较,并根据评定尺度确定其相对重要程度,最后据此建立模糊判断矩阵,从而确定各要素的相对重要度。通过采用三角模糊层次分析法,可确定各存储影响因素对应的权重。在具体实施时,可采用“1-9”标度来衡量指标的重要性,通过对云存储环境的数据进行分析,对两两之间的存储影响

因素的比较可以都选取三组三角模糊数据,最终确定每个存储影响因素对应的权重,以便于构造遗传算法目标函数。其中,所有存储影响因素对应的权重的总和为1。

[0066] 在具体实施时,根据各存储影响因素对应的数值和权重,基于遗传算法确定与待存储数据匹配的最佳存储节点的步骤,包括:

[0067] (1) 对各存储影响因素对应的数值进行归一化处理。

[0068] 以待存储数据的数据量、待存储数据与存储节点之间的传输距离、存储节点的故障率、存储节点的剩余空间、存储节点的内存占用率、存储节点对应的网络带宽和存储节点对应的网络延迟共七个存储影响因素为例,令存储节点的剩余空间和存储节点对应的网络带宽为正指标(也即,数值越大越好),其余的存储影响因素均为反指标(也即,数值越小越好),通过对各数值进行归一化,可消除量纲效应。

[0069] (2) 将归一化处理后的各存储影响因素对应的数值、各存储影响因素对应的权重代入至预先建立的遗传目标函数,求解获得待存储数据与最佳存储节点的对应关系。

[0070] 在具体实施时,需要调度的资源主要为存储节点,具体需要考虑到存储节点的剩余空间、内存占用率和网络延迟。其中内存占用率与网络延迟会随着同时或连续存入的数据量的增加而上升,达到一定程度后会导致数据传输失败。除此之外,其它的存储影响因素可作为遗传目标函数的约束条件,以罚函数的形式体现在遗传目标函数中。

[0071] 设定在数据存储过程中考虑待存储数据的数据量、待存储数据与存储节点之间的传输距离、存储节点的故障率、存储节点的剩余空间、存储节点的内存占用率、存储节点对应的网络带宽和存储节点对应的网络延迟共七个存储影响因素,遗传目标函数的建立过程包括:

[0072] 设置待存储数据的序列为 $X = \{x_1, x_2, \dots, x_m\}$,当前可用的存储节点的序列为 $Y = \{y_1, y_2, \dots, y_m\}$,设置 $n \leq m$,也即设置待存储数据的个数少于存储节点的个数,建立存储方案矩阵 M_k 如下:

$$[0073] \quad M_k = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_m \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_m \\ \cdots & \cdots & \cdots & \cdots \\ x_n y_1 & x_n y_2 & \cdots & x_n y_m \end{bmatrix}$$

[0074] 且 M_k 需满足 $\sum_{j=1}^m x_i y_j = 1$;

[0075] 其中, $M_k^{ij} = x_i y_j$,当 $x_i y_j = 1$ 时,表示 x_i 存储在 y_j 中;当 $x_i y_j = 0$ 时,表示 x_i 未存储在 y_j 中;

[0076] 基于存储方案矩阵 M_k ,构造以下遗传目标函数 Z_k :

$$\begin{cases}
 Z_k = M_k \otimes P^T + \sum_{i=1}^n \sum_{j=1}^m x_i y_j \cdot D_j \cdot w_4 + \text{Size} \cdot w_5 \\
 P = [P_1, P_2, \dots, P_m] \\
 P_j = \text{Space}_j \cdot w_2 + \text{Mem}_j \cdot w_3 + \text{NetB}_j \cdot w_6 + \text{NetD}_j \cdot w_7 + \text{Fail}_j \cdot w_1
 \end{cases}$$

$$[0078] \quad \text{且 } Z_k \text{ 需满足 } \sum_{j=1}^m \sum_{i=1}^n x_i y_j \cdot \text{Size}_i \leq \text{Capacity}_j ;$$

[0079] 其中,矩阵P为各存储节点对应的代价矩阵, D_j 为待存储数据所在的数据中心根节点与第j个存储节点之间的传输距离(可以为路由跳数), Size_i 为归一化处理后的第i个待存储数据的数据量, Capacity_j 为归一化处理后的第j个存储节点的初始空间, Space_j 为归一化处理后的第j个存储节点的剩余空间, Mem_j 为归一化处理后的第j个存储节点的内存使用率, NetB_j 为归一化处理后的第j个存储节点对应的网络带宽, NetD_j 为归一化处理后的第j个存储节点对应的网络延迟, Fail_j 为归一化处理后的第j个存储节点的故障率, w_1 为存储节点的故障率对应的权重, w_2 为存储节点的剩余空间对应的权重, w_3 为存储节点的内存使用率对应的权重, w_4 为待存储数据的传输距离对应的权重, w_5 为待存储数据的数据量对应的权重, w_6 为存储节点的网络带宽对应的权重, w_7 为存储节点的网络延迟对应的权重。其中, w_1 至 w_7 的七个权重的总和为1。为了使遗传目标函数的求解更为简洁,假设所有存储节点具有相同的I/O频率,考虑到数据量在整体上对存储节点的选取影响偏低,因此可统一设置各待存储数据的数据量为常数Size。

[0080] 设定在待存储数据的序列 $X = \{x_1, x_2, \dots, x_m\}$ 中, $x_i = \{s_i, m_i, n_i\}$, s_i 为第i个待存储数据所需占用的存储空间, m_i 为第i个待存储数据所需占用的内存量, n_i 为第i个待存储数据所需占用的网络资源;

[0081] 设定在存储节点的序列为 $Y = \{y_1, y_2, \dots, y_m\}$ 中, $y_i = \{S_i, M_i, N_i\}$, S_i 为第i个存储节点的存储空间, M_i 为第i个存储节点的内存资源, N_i 为第i个存储节点的网络资源;

[0082] 假设第i个存储节点对应存储第a个待存储数据、第b个待存储数据和第c个待存储数据,存储方案矩阵 M_k 需满足如下约束条件:

$$[0083] \quad S_i \geq s_a + s_b + \dots + s_c$$

$$[0084] \quad M_i \geq m_a + m_b + \dots + m_c$$

$$[0085] \quad N_i \geq n_a + n_b + \dots + n_c$$

$$[0086] \quad \text{且求解得到的存储方案矩阵 } M_k \text{ 满足 } T = \min \sum_{i=1}^m (t_i(s) + t_i(m) + t_i(n));$$

[0087] 其中 $t_i(s)$ 为在数据存储过程中由待存储数据产生的耗时, $t_i(m)$ 在数据存储过程中由存储节点产生的耗时, $t_i(n)$ 为在数据存储过程中由网络环境产生的耗时。

[0088] 在求解获得待存储数据与最佳存储节点的对应关系的步骤中,具体可参照如下方式执行:

[0089] (1) 在求解遗传目标函数的过程中,对 M_k 进行变异处理,以实现遗传算法中的种群

进化;其中,设定 M_k 为遗传算法中的染色体。

[0090] 其中,变异处理的方式包括:随机从存储方案矩阵 M_k 中挑选 M_k^{ij} ;对于挑选得到的 M_k^{ij} ,在满足 $\sum_{j=1}^m x_i y_j = 1$ 的基础上,如果 $M_k^{ij} = 1$,调整为 $M_k^{ij} = 0$;如果 $M_k^{ij} = 0$,调整为 $M_k^{ij} = 1$ 。

[0091] 在具体实现时,可以给出如下定义:设存在矩阵 T 为 $n \times m$ 阶初等矩阵,则称种群中个体交换后的结果为 $M'_k = M_k \cdot T$;将染色体 M_k 中若干行中的“1”调整为“0”,并在该行随机选择一个“0”调整为“1”,称之为变异。在确保 $\sum_{j=1}^m x_i y_j = 1$ 成立的基础上,即可完成种群的进化。

[0092] (2) 在种群进化的过程中,采用多次迭代的方式计算得到最优存储方案矩阵 M^{AG} ;其中,每次迭代均选取适应度排名在预设范围内的优质染色体 M_N^G 作为在下次迭代过程中进行计算的 M_k ,并淘汰适应度排名在预设范围外的致死染色体 M_M^B 。

[0093] 在具体实现时,对于进化过程中所产生的致死染色体,本实施例提供了一种基于学习的致死染色体自我改善算法,首先有如下定义:

[0094] 种群中适应度排名前10%的个体染色体为优质染色体 M_N^G ;种群中适应度排名

后10%及不符合 $\sum_{j=1}^m \sum_{i=1}^n x_i y_j \cdot Size_i \leq Capacity_j$ 的染色体称为致死染色体 M_M^B 。具体的,该

算法首先为每个致死染色体从优质染色体集合中选择一个作为学习模板,若两者在相同基因位置的值都为1,则将其保留,然后根据学习率随机从学习模板中选择若干值为1的基因,将其替换到致死染色体的对应位置中,则该行原为1的基因值替换为0,最后,将所有优质染色体的基因进行累加,并将累加值等于优质染色体个数的基因进行保留且置为1,得到该种群的一个综合优质染色体 M^{AG} ,并将 M^{AG} 中值为1的基因位置所对应的致死染色体基因值置为1,相应地该行原为1的基因替换为0。该算法在保证种群基因多样性的基础上提高了基因质量。

[0095] (3) 根据最优存储方案矩阵 M^{AG} ,确定待存储数据与最佳存储节点的对应关系。具体实施时,根据最优存储方案矩阵 M^{AG} 中各元素的值,确定待存储数据的序列为 $X = \{x_1, x_2, \dots, x_m\}$ 与当前可用的存储节点的序列为 $Y = \{y_1, y_2, \dots, y_m\}$ 的存储匹配关系 $K = (k_1, k_2, \dots, k_m)$;在存储匹配关系 K 中, m 为待存储数据的序列号, k_m 为序列号为 m 的待存储数据对应的最佳存储节点的序列号。

[0096] 在前述基础上,可参见图3所示的另一种用于云存储环境的数据存储方法的示意图,为便于理解,在此对图3进行解释说明:

[0097] 1,根据当前的存储节点对调度影响因素(也即前述的存储影响因素)进行分析,确定调度影响因素。在一种实施方式中,可以确定调度影响因素包括:待存储数据的数据量、待存储数据与存储节点之间的传输距离、存储节点的故障率、存储节点的剩余空间、存储节点的内存占用率、存储节点对应的网络带宽或存储节点对应的网络延迟。

[0098] 2,根据分析得到的调度影响因素和用户数据序列(也即,前述的待存储数据的序

列),进行三角模糊层次分析,得到层次分析结果,也即确定各调度影响因素对应的权重。

[0099] 3,根据层次分析结果以及用户数据序列,构造改进的遗传算法。其中,改进的遗传算法指的是基于本实施例提出的遗传目标函数进行求解。

[0100] 4,根据用户数据序列和构造的改进遗传算法,得到优选存储方案。

[0101] 5,根据优选存储方案为待存储数据分配存储节点,也即实现存储节点的调度。

[0102] 图3中各步骤的具体实现方式可参照前述内容,在图3中的各数字与上述步骤相对应,表示执行次序,在此不再赘述。

[0103] 综上所述,本实施例提供的用于云存储环境的数据存储方法,能够基于全局进行考虑,得到全局最优解,存储耗时更少,得到的数据存储方案更加可靠,以便实现更好的负载均衡,使云资源存储系统能够稳定运行。

[0104] 实施例二:

[0105] 对应于前述实施例提供的数据存储方法,本实施例提供了一种数据存储装置,参见图4所示的一种用于云存储环境的数据存储装置的结构示意图,包括:

[0106] 因素确定模块402,用于当接收到待存储数据时,根据待存储数据、当前的多个存储节点和网络环境,确定各存储影响因素对应的数值;存储影响因素包括:待存储数据的数据量、待存储数据与存储节点之间的传输距离、存储节点的故障率、存储节点的剩余空间、存储节点的内存占用率、存储节点对应的网络带宽或存储节点对应的网络延迟;

[0107] 最佳节点确定模块404,用于根据各存储影响因素对应的数值,确定与待存储数据匹配的最佳存储节点,以将待存储数据对应存储于最佳存储节点中。

[0108] 本发明实施例提供的上述用于云存储环境的数据存储装置,在接收到待存储数据时,能够根据待存储数据、当前的多个存储节点和网络环境确定各存储影响因素对应的数值,进而根据各存储影响因素对应的数值,确定与待存储数据匹配的最佳存储节点。这种基于多种存储影响因素的综合考虑而选取合适的存储节点的方式,考虑因素较为全面,能够提供较好的数据存储方案。

[0109] 在具体实施时,上述因素确定模块302包括:

[0110] 权重确定单元,用于通过三角模糊层次分析法确定各存储影响因素对应的权重;

[0111] 最佳节点确定单元,根据各存储影响因素对应的数值和权重,基于遗传算法确定与待存储数据匹配的最佳存储节点。

[0112] 最佳节点确定单元进一步用于:对各存储影响因素对应的数值进行归一化处理;将归一化处理后的各存储影响因素对应的数值、各存储影响因素对应的权重代入至预先建立的遗传目标函数,求解获得待存储数据与最佳存储节点的对应关系。

[0113] 遗传目标函数的建立过程包括:

[0114] 设置待存储数据的序列为 $X = \{x_1, x_2, \dots, x_m\}$,当前可用的存储节点的序列为 $Y = \{y_1, y_2, \dots, y_m\}$,建立存储方案矩阵 M_k 如下:

$$[0115] \quad M_k = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_m \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_m \\ \cdots & \cdots & \cdots & \cdots \\ x_n y_1 & x_n y_2 & \cdots & x_n y_m \end{bmatrix}$$

[0116] 且 M_k 需满足 $\sum_{j=1}^m x_i y_j = 1$;

[0117] 其中, $M_k^{ij} = x_i y_j$, 当 $x_i y_j = 1$ 时, 表示 x_i 存储在 y_j 中; 当 $x_i y_j = 0$ 时, 表示 x_i 未存储在 y_j 中;

[0118] 基于存储方案矩阵 M_k , 构造以下遗传目标函数 Z_k :

$$[0119] \left\{ \begin{array}{l} Z_k = M_k \otimes P^T + \sum_{i=1}^n \sum_{j=1}^m x_i y_j \cdot D_j \cdot w_4 + Size \cdot w_5 \\ P = [P_1, P_2, \dots, P_m] \\ P_j = Space_j \cdot w_2 + Mem_j \cdot w_3 + NetB_j \cdot w_6 + NetD_j \cdot w_7 + Fail_j \cdot w_1 \end{array} \right.$$

[0120] 且 Z_k 需满足 $\sum_{j=1}^m \sum_{i=1}^n x_i y_j \cdot Size_i \leq Capacity_j$;

[0121] 其中, 矩阵 P 为各存储节点对应的代价矩阵, D_j 为待存储数据所在的数据中心根节点与第 j 个存储节点之间的传输距离, $Size_i$ 为归一化处理后的第 i 个待存储数据的数据量, $Capacity_j$ 为归一化处理后的第 j 个存储节点的初始空间, $Space_j$ 为归一化处理后的第 j 个存储节点的剩余空间, Mem_j 为归一化处理后的第 j 个存储节点的内存使用率, $NetB_j$ 为归一化处理后的第 j 个存储节点对应的网络带宽, $NetD_j$ 为归一化处理后的第 j 个存储节点对应的网络延迟, $Fail_j$ 为归一化处理后的第 j 个存储节点的故障率, w_1 为存储节点的故障率对应的权重, w_2 为存储节点的剩余空间对应的权重, w_3 为存储节点的内存使用率对应的权重, w_4 为待存储数据的传输距离对应的权重, w_5 为待存储数据的数据量对应的权重, w_6 为存储节点的网络带宽对应的权重, w_7 为存储节点的网络延迟对应的权重。

[0122] 最佳节点确定单元进一步用于:

[0123] 在求解遗传目标函数的过程中, 对 M_k 进行变异处理, 以实现遗传算法中的种群进化; 其中, 设定 M_k 为遗传算法中的染色体; 变异处理的方式包括: 随机从存储方案矩阵 M_k 中挑选 M_k^{ij} ; 对于挑选得到的 M_k^{ij} , 在满足 $\sum_{j=1}^m x_i y_j = 1$ 的基础上, 如果 $M_k^{ij} = 1$, 调整为 $M_k^{ij} = 0$; 如

果 $M_k^{ij} = 0$, 调整为 $M_k^{ij} = 1$ 。

[0124] 在种群进化的过程中, 采用多次迭代的方式计算得到最优存储方案矩阵 M^{AG} ; 其中, 每次迭代均选取适应度排名在预设范围内的优质染色体 M_N^G 作为在下次迭代过程中进行计算的 M_k , 并淘汰适应度排名在预设范围外的致死染色体 M_M^B ;

[0125] 根据最优存储方案矩阵 M^{AG} , 确定待存储数据与最佳存储节点的对应关系。

[0126] 最佳节点确定单元进一步用于: 根据最优存储方案矩阵 M^{AG} 中各元素的值, 确定待存储数据的序列为 $X = \{x_1, x_2, \dots, x_m\}$ 与当前可用的存储节点的序列为 $Y = \{y_1, y_2, \dots, y_m\}$ 的存储匹配关系 $K = (k_1, k_2, \dots, k_m)$; 在存储匹配关系 K 中, m 为待存储数据的序列号, k_m 为序列号为 m 的待存储数据对应的最佳存储节点的序列号。

[0127] 设定在待存储数据的序列 $X = \{x_1, x_2, \dots, x_m\}$ 中, $x_i = \{s_i, m_i, n_i\}$, s_i 为第 i 个待存储数据所需占用的存储空间, m_i 为第 i 个待存储数据所需占用的内存量, n_i 为第 i 个待存储数据所需占用的网络资源;

[0128] 设定在存储节点的序列为 $Y = \{y_1, y_2, \dots, y_m\}$ 中, $y_i = \{S_i, M_i, N_i\}$, S_i 为第 i 个存储节点的存储空间, M_i 为第 i 个存储节点的内存资源, N_i 为第 i 个存储节点的网络资源;

[0129] 假设第 i 个存储节点对应存储第 a 个待存储数据、第 b 个待存储数据和第 c 个待存储数据, 存储方案矩阵 M_k 需满足如下约束条件:

$$[0130] \quad S_i \geq s_a + s_b + \dots + s_c$$

$$[0131] \quad M_i \geq m_a + m_b + \dots + m_c$$

$$[0132] \quad N_i \geq n_a + n_b + \dots + n_c$$

[0133] 且求解得到的存储方案矩阵 M_k 满足 $T = \min \sum_{i=1}^m (t_i(s) + t_i(m) + t_i(n))$;

[0134] 其中 $t_i(s)$ 为在数据存储过程中由待存储数据产生的耗时, $t_i(m)$ 在数据存储过程中由存储节点产生的耗时, $t_i(n)$ 为在数据存储过程中由网络环境产生的耗时。

[0135] 本实施例所提供的装置, 其实现原理及产生的技术效果和前述实施例相同, 为简要描述, 装置实施例部分未提及之处, 可参考前述方法实施例中相应内容。

[0136] 实施例三:

[0137] 对应于前述实施例, 本实施例提供了一种终端, 该终端包括存储器以及处理器, 存储器用于存储支持处理器执行实施例一提供的用于云存储环境的数据存储方法的程序, 处理器被配置为用于执行存储器中存储的程序。具体的, 该终端可以为计算机、上位机、服务器等。

[0138] 进一步, 本实施例还提供了一种计算机存储介质, 用于储存为实施例一提供的用于云存储环境的数据存储方法所用的计算机软件指令。

[0139] 图5为本发明实施例提供的一种终端的结构示意图, 包括: 处理器50, 存储器51, 总线52和通信接口53, 处理器50、通信接口53和存储器51通过总线52连接; 处理器50用于执行存储器51中存储的可执行模块, 例如计算机程序。

[0140] 其中, 存储器51可能包含高速随机存取存储器(RAM, Random Access Memory), 也可能还包括非不稳定的存储器(non-volatile memory), 例如至少一个磁盘存储器。通过至少一个通信接口53(可以是有线或者无线)实现该系统网元与至少一个其他网元之间的通信连接, 可以使用互联网, 广域网, 本地网, 城域网等。

[0141] 总线52可以是ISA总线、PCI总线或EISA总线等。总线可以分为地址总线、数据总线、控制总线等。为便于表示, 图5中仅用一个双向箭头表示, 但并不表示仅有一根总线或一种类型的总线。

[0142] 其中, 存储器51用于存储程序, 处理器50在接收到执行指令后, 执行程序501, 前述本发明实施例任一实施例揭示的流过程定义的装置所执行的方法可以应用于处理器50中, 或者由处理器50实现。

[0143] 处理器50可能是一种集成电路芯片, 具有信号的处理能力。在实现过程中, 上述方法的各步骤可以通过处理器50中的硬件的集成逻辑电路或者软件形式的指令完成。上述的处理器50可以是通用处理器, 包括中央处理器(Central Processing Unit, 简称CPU)、网络

处理器 (Network Processor, 简称NP) 等;还可以是数字信号处理器 (Digital Signal Processing, 简称DSP)、专用集成电路 (Application Specific Integrated Circuit, 简称ASIC)、现成可编程门阵列 (Field-Programmable Gate Array, 简称FPGA) 或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。可以实现或者执行本发明实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合本发明实施例所公开的方法的步骤可以直接体现为硬件译码处理器执行完成,或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器,闪存、只读存储器,可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于存储器51,处理器50读取存储器51中的信息,结合其硬件完成上述方法的步骤。

[0144] 本发明实施例所提供的用于云存储环境的数据存储方法及装置的计算机程序产品,包括存储了程序代码的计算机可读存储介质,所述程序代码包括的指令可用于执行前面方法实施例中所述的方法,具体实现可参见方法实施例,在此不再赘述。

[0145] 所述功能如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器 (ROM, Read-Only Memory)、随机存取存储器 (RAM, Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0146] 最后应说明的是:以上所述实施例,仅为本发明的具体实施方式,用以说明本发明的技术方案,而非对其限制,本发明的保护范围并不局限于此,尽管参照前述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:任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,其依然可以对前述实施例所记载的技术方案进行修改或可轻易想到变化,或者对其中部分技术特征进行等同替换;而这些修改、变化或者替换,并不使相应技术方案的本质脱离本发明实施例技术方案的精神和范围,都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以所述权利要求的保护范围为准。

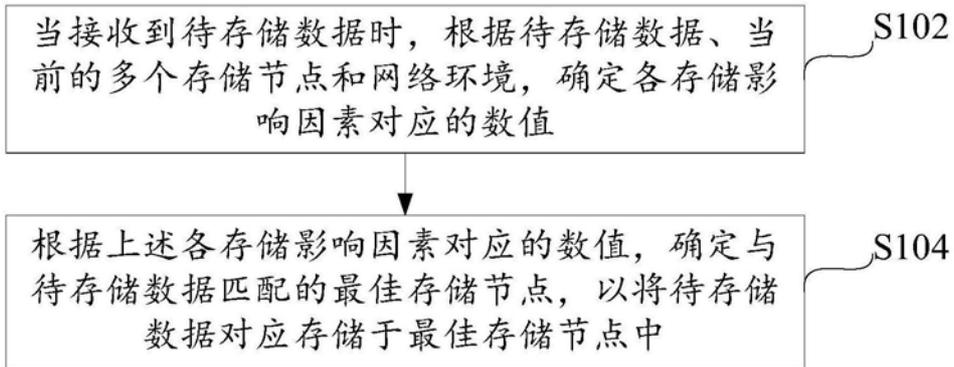


图1

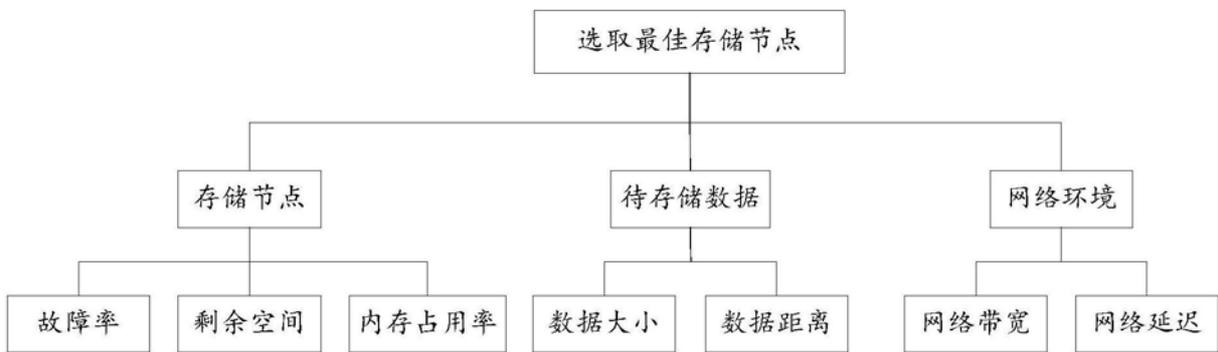


图2

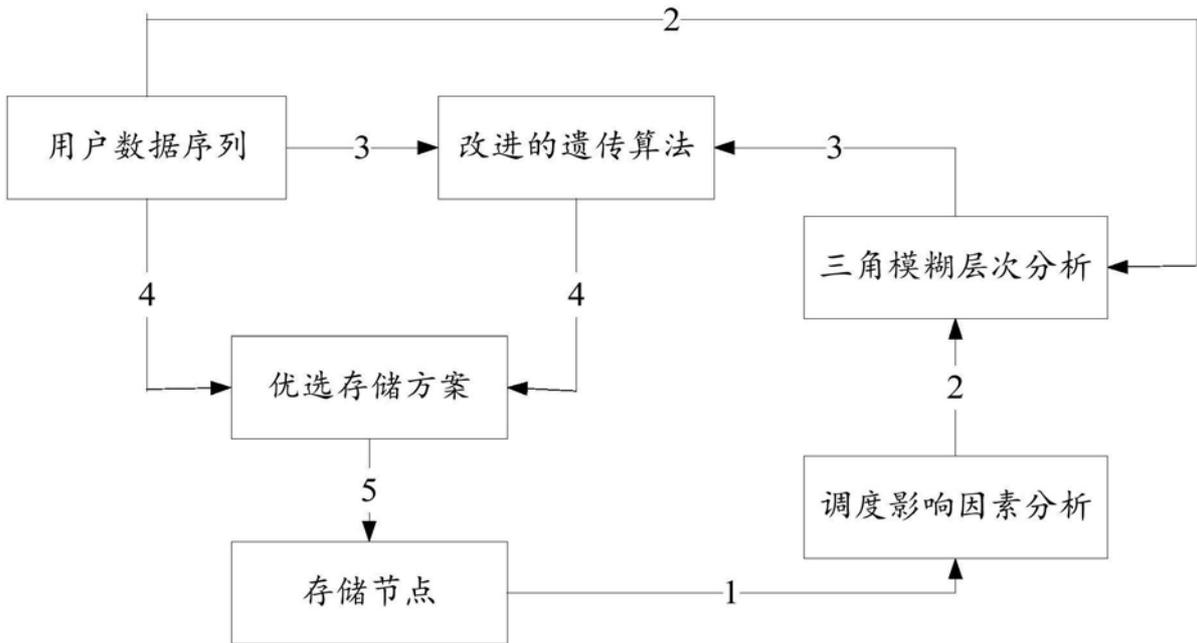


图3

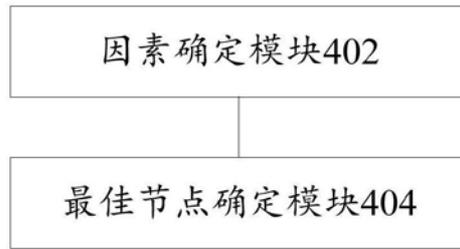


图4

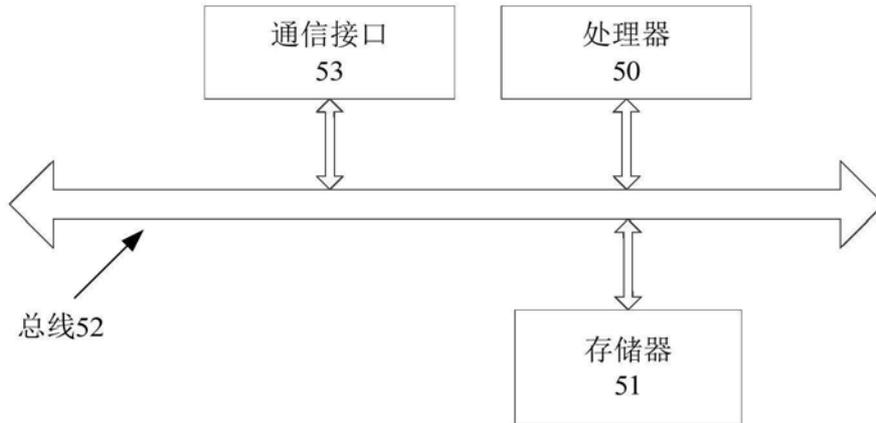


图5