



(12)发明专利

(10)授权公告号 CN 104573082 B

(45)授权公告日 2017.11.14

(21)申请号 201510042456.9

G06F 3/06(2006.01)

(22)申请日 2015.01.28

(56)对比文件

CN 101453688 A, 2009.06.10,

US 2013/0110915 A1, 2013.05.02,

杨婧 等.基于向量关系表的自动数据收集  
算法.《计算机工程与应用》.2007,第43卷(第15  
期), 176-179.

(65)同一申请的已公布的文献号  
申请公布号 CN 104573082 A

(43)申请公布日 2015.04.29

审查员 邱川

(73)专利权人 武汉大学

地址 430072 湖北省武汉市武昌区珞珈山  
武汉大学

(72)发明人 潘少明 徐正全 种衍文 李红  
李明 汤戈

(74)专利代理机构 武汉科皓知识产权代理事务  
所(特殊普通合伙) 42222

代理人 严彦

(51)Int.Cl.

G06F 17/30(2006.01)

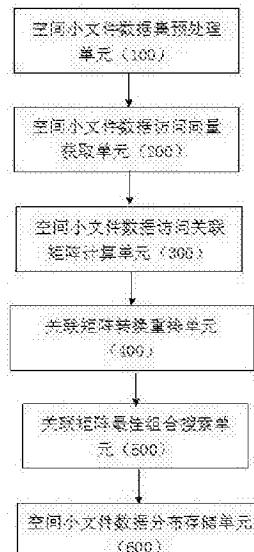
权利要求书4页 说明书12页 附图3页

(54)发明名称

基于访问日志信息的空间小文件数据分布  
存储方法及系统

(57)摘要

本发明提供基于访问日志信息的空间小文  
件数据分布存储方法及系统,包括将空间小文  
件数据集分成频繁访问的子集和非频繁访问的  
子集,提取频繁访问的空间小文件数据子集的访问  
序列,分段计算各频繁访问的空间小文件数据的  
关联度,并将各频繁访问的空间小文件数据相互  
之间的关联度数值组成关联矩阵;对关联矩阵中  
各元素数值进行大小转换后利用RCM排序算法重  
排后输出,对重排后的关联矩阵利用局部逼近搜  
索法寻找最佳组合,利用最佳组合对频繁访问的  
空间小文件数据进行分布存储,以及对非频繁访  
问的空间小文件数据根据空间位置相邻关系分  
开存储。本发明提高了空间小文件数据的并行访  
问性能。



1. 一种基于访问日志信息的空间小文件数据分布存储方法,其特征在于:对任一种空间小文件数据类型,执行包括以下步骤:

步骤1,将空间小文件数据集,按照访问频率不同分成频繁访问的子集和非频繁访问的子集;包括以下子步骤,

步骤1.1,获取各空间小文件数据访问热度,实现如下,

设空间小文件数据集为 $F = \{f_1, f_2, \dots, f_N\}$ ,包含空间小文件数据 $f_1, f_2, \dots, f_N$ ,其中N为空间小文件数据的总个数,第i个空间小文件数据标记为 $f_i, i=1, 2, \dots, N$ ;

设访问日志信息中记录依次访问了空间小文件数据 $f_{a_1}, f_{a_2}, \dots, f_{a_M}$ ,空间小文件数据的访问日志序列为 $R = \{f_{a_1}, f_{a_2}, \dots, f_{a_M}\}$ , $A = (a_1, a_2, \dots, a_M)$ 为空间小文件数据访问序列向量, $a_t \in [1, N]$ ,访问序号 $t=1, 2, \dots, M$ ,其中M为对F中所有空间小文件数据的访问总次数;

统计每个空间小文件数据 $f_i$ 在访问日志序列R中出现的次数 $\lambda_i$ ,以 $\lambda_i$ 为该空间小文件数据 $f_i$ 的访问热度;

步骤1.2,根据空间小文件数据访问热度提取被频繁访问的空间小文件数据,实现如下,

输入预设判别参数 $\lambda$ ,

若空间小文件数据集F中空间小文件数据 $f_i$ 的访问热度 $\lambda_i > \lambda$ ,则空间小文件数据 $f_i$ 为频繁访问的空间小文件数据,否则 $f_i$ 属于非频繁访问的空间小文件数据;

步骤1.3,根据步骤1.2所得频繁访问的空间小文件数据构成空间小文件数据集的子集,实现如下,

设所有频繁访问的空间小文件数据所构成子集为 $F_1 = \{f_{i_1}, f_{i_2}, \dots, f_{i_{N_1}}\}$ ,其中 $N_1$ 为频繁访问的空间小文件数据总个数,第 $i_1, j_1$ 个频繁访问的空间小文件数据分别标记为 $f_{i_1}$ 和 $f_{j_1}$ , $i_1, j_1 \in [1, N_1]$ ;

步骤2,从访问日志信息中提取频繁访问的空间小文件数据子集的访问序列,包括按照时间先后顺序形成访问序列 $R_1 = \{f_{a_{i_1}}, f_{a_{i_2}}, \dots, f_{a_{i_{N_1}}}\}$ , $A_1 = (a_{i_1}, a_{i_2}, \dots, a_{i_{N_1}})$ 为频繁访问空间小文件数据访问序列向量, $a_{i_t} \in [1, N_1]$ ,访问序号 $t_1 = (1_1, 2_1, \dots, M_1)$ ,其中 $M_1$ 为对 $F_1$ 中所有频繁访问空间小文件数据的访问总次数;

步骤3,利用频繁访问的空间小文件数据子集的访问序列分段计算各频繁访问的空间小文件数据的关联度,并将各频繁访问的空间小文件数据相互之间的关联度数值组成关联矩阵;包括以下子步骤,

步骤3.1,根据存储服务器数量m、频繁访问空间小文件数据子集长度 $N_1$ 计算频繁访问序列分段长度 $n = N_1/m$ ;

步骤3.2,根据访问序列分段长度对频繁访问序列进行分段,实现如下,

按照访问顺序,将频繁访问空间小文件数据访问序列向量 $A_1$ 以n个元素一组分割为若干子向量,表示为 $A_1 = (S_1, S_2, \dots, S_l)$ ,其中子向量 $S_k = (a_{k1}, a_{k2}, \dots, a_{kn})$ , $a_{kj} \in [1, N_1]$ , $1 \leq k \leq l$ , $1 \leq j \leq n$ ;将 $A_1$ 中所有子向量集合记为S,S = { $S_k : k \in [1, l]$ } ;

步骤3.3,计算频繁访问的空间小文件数据相互之间的关联度数值,实现如下,

定义函数 $R_{S_k}(i_1, j_1)$ ,

$$R_{S_k}(i_1, j_1) = \begin{cases} 1 & i_1 \in \tilde{S}_k, j_1 \in \tilde{S}_k, i_1 \neq j_1 \\ 0 & \text{其它} \end{cases} \quad 1 \leq i_1 \leq N_1, \quad 1 \leq j_1 \leq N_1, \quad 1 \leq k \leq l$$

其中  $\tilde{S}_k = \{a_{k1}, a_{k2}, \dots, a_{kn}\}$  为  $S_k$  中的所有元素组成的集合；函数  $R_{S_k}(i_1, j_1)$  表示在长度为  $n$  的访问周期内频繁访问的空间小文件数据  $f_{i_1}$  和  $f_{j_1}$  是否具有关联性；

定义函数  $R_S(i_1, j_1)$ ，

$$R_S(i_1, j_1) = \sum_{k=1}^l R_{S_k}(i_1, j_1) \quad 1 \leq i_1 \leq N_1, \quad 1 \leq j_1 \leq N_1$$

其中  $R_S(i_1, j_1)$  表示  $S$  对  $f_{i_1}$  和  $f_{j_1}$  的总关联度；

步骤3.4，将频繁访问的空间小文件数据相互之间的关联度数值组成关联矩阵  $R_S$ ，

$$R_S = (R_S(i_1, j_1))_{N_1 \times N_1} \quad 1 \leq i_1 \leq N_1, \quad 1 \leq j_1 \leq N_1$$

步骤4，对关联矩阵中各元素数值进行大小转换后利用RCM排序算法重排后输出；

步骤5，对重排后的关联矩阵利用局部逼近搜索法寻找最佳组合；

步骤5包括以下子步骤，

步骤5.1，初始化当前迭代次数  $d=1$ ；

步骤5.2，采用局部逼近搜索法寻找一个最佳组合，包括在当前的矩阵中寻找一个  $n \times n$  的块，使得该矩阵中  $n \times n$  块内对应的矩阵元素值最大，相应的  $n$  个文件构成一个最佳组合；第一次执行步骤5.2时，当前的矩阵为步骤4所得重排后的关联矩阵；后续执行步骤5.2时，当前的矩阵为前一次迭代所得的矩阵；

步骤5.3，在本次迭代执行步骤5.2搜索得到一个由  $n$  个文件组成的一个最佳组合后，将关联矩阵中对应  $n$  个文件的关联矩阵元素删除，得到  $(N_1-dn) \times (N_1-dn)$  的矩阵；

步骤5.4，判断是否  $d=m-1$ ，否则令  $d=d+1$ ，以本次迭代执行步骤5.3所得  $(N_1-dn) \times (N_1-dn)$  的矩阵为当前的矩阵，返回步骤5.2进行下一次迭代继续搜索下一个最佳组合，是则停止搜索，共得到  $m$  个最佳组合；

步骤6，利用步骤5所得最佳组合对频繁访问的空间小文件数据进行分布存储，以及对非频繁访问的空间小文件数据根据空间位置相邻关系分开存储。

2. 根据权利要求1所述基于访问日志信息的空间小文件数据分布存储方法，其特征在于：步骤4包括以下子步骤，

步骤4.1，获取关联矩阵中元素最大值，包括遍历关联矩阵所有元素值，并获取最大值  $R_{\max}$ ；

步骤4.2，对关联矩阵元素数值进行大小转换，包括遍历关联矩阵所有元素值，并执行操作  $R_S(i_1, j_1) = R_{\max} - R_S(i_1, j_1)$ ；

步骤4.3，利用标准RCM排序算法对关联矩阵进行重排。

3. 一种基于访问日志信息的空间小文件数据分布存储系统，其特征在于：包括以下单元，

空间小文件数据集预处理单元(100)，用于将任一种空间小文件数据类型的空间小文件数据集，按照访问频率不同分成频繁访问的子集和非频繁访问的子集；包括以下模块，

空间小文件数据访问频率统计模块(101)，用于获取各空间小文件数据访问热度，实现

如下，

设空间小文件数据集为  $F = \{f_1, f_2, \dots, f_N\}$ ，包含空间小文件数据  $f_1, f_2, \dots, f_N$ ，其中  $N$  为空间小文件数据的总个数，第  $i$  个空间小文件数据标记为  $f_i, i=1, 2, \dots, N$ ；

设访问日志信息中记录依次访问了空间小文件数据  $f_{a_1}, f_{a_2}, \dots, f_{a_M}$ ，空间小文件数据的访问日志序列为  $R = \{f_{a_1}, f_{a_2}, \dots, f_{a_M}\}$ ， $A = (a_1, a_2, \dots, a_M)$  为空间小文件数据访问序列向量， $a_t \in [1, N]$ ，访问序号  $t=1, 2, \dots, M$ ，其中  $M$  为对  $F$  中所有空间小文件数据的访问总次数；

统计每个空间小文件数据  $f_i$  在访问日志序列  $R$  中出现的次数  $\lambda_i$ ，以  $\lambda_i$  为该空间小文件数据  $f_i$  的访问热度；

频繁访问空间小文件数据集提取模块 (102)，用于根据空间小文件数据访问热度提取被频繁访问的空间小文件数据，实现如下，

输入预设判别参数  $\lambda$ ，

若空间小文件数据集  $F$  中空间小文件数据  $f_i$  的访问热度  $\lambda_i > \lambda$ ，则空间小文件数据  $f_i$  为频繁访问的空间小文件数据，否则  $f_i$  属于非频繁访问的空间小文件数据；

频繁访问空间小文件子集构建模块 (103)，用于根据频繁访问空间小文件数据集提取模块 (102) 所得频繁访问的空间小文件数据构成空间小文件数据集的子集，实现如下，

设所有频繁访问的空间小文件数据所构成子集为  $F_1 = \{f_{i_1}, f_{j_1}, \dots, f_{N_1}\}$ ，其中  $N_1$  为频繁访问的空间小文件数据总个数，第  $i_1, j_1$  个频繁访问的空间小文件数据分别标记为  $f_{i_1}$  和  $f_{j_1}$ ， $i_1, j_1 \in [1, N_1]$ ；

空间小文件数据访问向量获取单元 (200)，用于从访问日志信息中提取频繁访问的空间小文件数据子集的访问序列，包括按照时间先后顺序形成访问序列  $R_1 = \{f_{a_{t_1}}, f_{a_{t_2}}, \dots, f_{a_{M_1}}\}$ ， $A_1 = (a_{t_1}, a_{t_2}, \dots, a_{M_1})$  为频繁访问空间小文件数据访问序列向量， $a_{t_i} \in [1, N_1]$ ，访问序号  $t_1 = (1_1, 2_1, \dots, M_1)$ ，其中  $M_1$  为对  $F_1$  中所有频繁访问空间小文件数据的访问总次数；

空间小文件数据访问关联矩阵计算单元 (300)，用于利用频繁访问的空间小文件数据子集的访问序列分段计算各频繁访问的空间小文件数据的关联度，并将各频繁访问的空间小文件数据相互之间的关联度数值组成关联矩阵；包括以下模块，

频繁访问序列分段长度计算模块 (301)，用于根据存储服务器数量  $m$ 、频繁访问空间小文件数据子集长度  $N_1$  计算频繁访问序列分段长度  $n = N_1/m$ ；

频繁访问序列分段模块 (302)，用于根据访问序列分段长度对频繁访问序列进行分段，实现如下，

按照访问顺序，将频繁访问空间小文件数据访问序列向量  $A_1$  以  $n$  个元素一组分割为若干子向量，表示为  $A_1 = (S_1, S_2, \dots, S_l)$ ，其中子向量  $S_k = (a_{k1}, a_{k2}, \dots, a_{kn})$ ， $a_{kj} \in [1, N_1]$ ， $1 \leq k \leq l$ ， $1 \leq j \leq n$ ；将  $A_1$  中所有子向量集合记为  $S$ ， $S = \{S_k : k \in [1, l]\}$ ；

空间小文件数据关联度计算模块 (303)，用于计算频繁访问的空间小文件数据相互之间的关联度数值，实现如下，

定义函数  $R_{S_k}(i_1, j_1)$ ，

$$R_{S_k}(i_1, j_1) = \begin{cases} 1 & i_1 \in \tilde{S}_k, j_1 \in \tilde{S}_k, i_1 \neq j_1, 1 \leq i_1 \leq N_1, 1 \leq j_1 \leq N_1, 1 \leq k \leq l \\ 0 & \text{其它} \end{cases}$$

其中  $\tilde{S}_k = \{a_{k1}, a_{k2}, \dots, a_{kn}\}$  为  $S_k$  中的所有元素组成的集合; 函数  $R_{S_k}(i_1, j_1)$  表示在长度为 n 的访问周期内频繁访问的空间小文件数据  $f_{i_1}$  和  $f_{j_1}$  是否具有关联性;

定义函数  $R_S(i_1, j_1)$ ,

$$R_S(i_1, j_1) = \sum_{k=1}^l R_{S_k}(i_1, j_1) \quad 1 \leq i_1 \leq N_1, \quad 1 \leq j_1 \leq N_1$$

其中  $R_S(i_1, j_1)$  表示  $S$  对  $f_{i_1}$  和  $f_{j_1}$  的总关联度;

空间小文件数据关联矩阵生成模块(304), 用于将频繁访问的空间小文件数据相互之间的关联度数值组成关联矩阵  $R_S$ ,

$$R_S = (R_S(i_1, j_1))_{N_1 \times N_1} \quad 1 \leq i_1 \leq N_1, \quad 1 \leq j_1 \leq N_1$$

关联矩阵转换重排单元(400), 用于对关联矩阵中各元素数值进行大小转换后利用RCM排序算法重排后输出;

关联矩阵最佳组合搜索单元(500), 用于对重排后的关联矩阵利用局部逼近搜索法寻找最佳组合;

关联矩阵最佳组合搜索单元(500)包括以下模块,

初始化模块, 用于初始化当前迭代次数  $d=1$ ;

最佳组合搜索模块, 用于采用局部逼近搜索法寻找一个最佳组合, 包括在当前的矩阵中寻找一个  $n \times n$  的块, 使得该矩阵中  $n \times n$  块内对应的矩阵元素值最大, 相应的  $n$  个文件构成一个最佳组合; 最佳组合搜索模第一次工作时, 当前的矩阵为关联矩阵转换重排单元(400)所得重排后的关联矩阵; 最佳组合搜索模后续工作时, 当前的矩阵为前一次迭代所得的矩阵;

矩阵更新模块, 用于在最佳组合搜索模块进行本次迭代工作搜索得到一个由  $n$  个文件组成的一个最佳组合后, 将关联矩阵中对应  $n$  个文件的关联矩阵元素删除, 得到  $(N_1-dn) \times (N_1-dn)$  的矩阵;

判断输出模块, 用于判断是否  $d=m-1$ , 否则令  $d=d+1$ , 以矩阵更新模块本次迭代工作所得  $(N_1-dn) \times (N_1-dn)$  的矩阵为当前的矩阵, 命令最佳组合搜索模块进行下一次迭代工作继续搜索下一个最佳组合, 是则停止搜索, 共得到  $m$  个最佳组合;

空间小文件数据分布存储单元(600), 用于利用关联矩阵最佳组合搜索单元(500)所得最佳组合对频繁访问的空间小文件数据进行分布存储, 以及对非频繁访问的空间小文件数据根据空间位置相邻关系分开存储。

4. 根据权利要求3所述基于访问日志信息的空间小文件数据分布存储系统, 其特征在于: 关联矩阵转换重排单元(400)包括以下模块,

关联矩阵元素最大值获取模块(401), 用于获取关联矩阵中元素最大值, 包括遍历关联矩阵所有元素值, 并获取最大值  $R_{max}$ ;

关联矩阵元素值大小转换模块(402), 用于对关联矩阵元素数值进行大小转换, 包括遍历关联矩阵所有元素值, 并执行操作  $R_S(i_1, j_1) = R_{max} - R_S(i_1, j_1)$ ;

关联矩阵重排模块(403), 用于利用标准RCM排序算法对关联矩阵进行重排。

## 基于访问日志信息的空间小文件数据分布存储方法及系统

### 技术领域

[0001] 本发明属于空间小文件数据的分布存储技术领域,特别是涉及一种新的基于访问日志信息的空间小文件数据分布存储方法及系统。

### 背景技术

[0002] 海量空间信息的存储和快速访问一直是空间信息服务系统试图解决的重要问题,常用的空间信息服务系统如NASA地球观察系统每天采集的数据量达到了2TB,对这些数据的合理分布存储以便获得并行快速访问成为关键,其中一类重要的解决方案是通过对数据进行分布存储以实现对数据的并行访问来提高数据访问效率。

[0003] 目前比较典型的分布式文件存储系统主要包括如GFS (Google file system)、HDFS (Hadoop distributed file system) 以及Lustre等。但这些系统在存储性能上的改善主要体现在对大文件的存储处理上。如GFS,其存储策略主要是,将大文件分成固定长度的块(如64MB),然后将所有的块分别存储在不同的存储器上以提高数据的并行访问率(参考文献Ghemawat S,Gobioff H,Shun-Tak L.The Google file system.In:Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles (SOSP' 03).Bolton Landing,New York:IEEE,2003.1-15)。另一类典型的存储技术如RAID (Redundant Array of Independent Disks),也是将每一个大的数据文件分成几个数据块后分别存储在不同的磁盘以提高对该文件的并行访问。

[0004] 以上分布存储方法虽然对大文件数据有效,但针对小文件数据,由于无法继续进行分块,通过分块存储的方法适应性不足,目前通用的方法只是简单的将单个文件存储在单个存储服务器上,因而难以实现对多个小文件数据的并行访问,I/O效率不高。

[0005] 研究表明,目前大部分系统都存在大量的小文件数据,如美国国家能源研究科学计算中心的1300万个文件中有99%的文件小于64M,小于64K的文件更是占到了44%(参考文献Carns P,Lang S,Ross R,et al..Small-file access in parallel file systems [C].Parallel&Distributed Processing,2009.IPDPS 2009.IEEE International Symposium On.IEEE,2009:1-11)。

[0006] 事实上,基于金字塔模型的空间信息服务系统,如Google Earth、World Wind等同样是以小文件的形式存储空间数据。World Wind根据金字塔模型将地球分成不同分辨率的瓦片数据,每个瓦片数据保存为一个文件,每个瓦片数据的大小固定为 $512 \times 512$ 像素,每个瓦片文件大小不超过1MB(参考文献Boschetti L,Roy D P,Justice C O.Using NASA's World Wind virtual globe for interactive internet visualization of the global MODIS burned area product.Int J Remote Sens,2008,29 (11):3067-3072);Google Earth同样采用多分辨率模型存储空间数据,每个数据文件的大小也不超过64MB(参考文献Sample J T,Loup E.Tile-base geospatial information system:principle and practices.New York:Springer,2010.23-200)。

[0007] 总之,目前针对大文件数据的分布存储方法难以应用到小文件数据的存储,而针

对小文件数据的优化又集中在数据的访问优化(非存储优化,访问优化面向客户端,而存储优化面向服务端),如减少数据密集型应用程序的执行时间(参考文献J.Kim,A.Chandra, and J.B.Weissma.Using Data Accessibility for Resource Selection in Large-Scale Distributed Systems.IEEE Trans.Parallel Distributed Systems,vol.20,no.6,pp.788–801,June 2009),或降低小文件索引信息的开销(参考文献A.L.Chervenak, R.Schuler,M.Ripeanu,M.A.Amer,S.Bharathi,I.Foster,A.Iamnitchi, and C.Kesselman.The Globus Replica Location Service:Design and Experience.IEEE Trans.Parallel Distributed Systems,vol.20,no.9,pp.1260–1272,Sept.2009)等。但在分布式系统中,访问延迟时间的性能不仅与存取方法有关,而且与数据的分布存储模式有关。因此对小文件数据的优化问题尚未得到根本解决。

## 发明内容

[0008] 针对以上问题,本发明提供一种基于访问日志信息的空间小文件数据分布存储方法及系统,利用空间小文件数据的访问日志信息,分析各空间小文件数据之间的相互关系,并据此对空间小文件数据进行分布存储,以提高对空间小文件数据的并行访问率。

[0009] 本发明所述的一种基于访问日志信息的空间小文件数据分布存储方法及系统,所采用的技术方案是:

[0010] 一种基于访问日志信息的空间小文件数据分布存储方法,对任一种空间小文件数据类型,执行包括以下步骤:

[0011] 步骤1,将空间小文件数据集,按照访问频率不同分成频繁访问的子集和非频繁访问的子集;包括以下子步骤,

[0012] 步骤1.1,获取各空间小文件数据访问热度,实现如下,

[0013] 设空间小文件数据集为 $F = \{f_1, f_2, \dots, f_N\}$ ,包含空间小文件数据 $f_1, f_2, \dots, f_N$ ,其中N为空间小文件数据的总个数,第i个空间小文件数据标记为 $f_i, i=1, 2, \dots, N$ ;

[0014] 设访问日志信息中记录依次访问了空间小文件数据 $f_{a_1}, f_{a_2}, \dots, f_{a_M}$ ,空间小文件数据的访问日志序列为 $R = \{f_{a_1}, f_{a_2}, \dots, f_{a_M}\}$ , $A = (a_1, a_2, \dots, a_M)$ 为空间小文件数据访问序列向量, $a_t \in [1, N]$ ,访问序号 $t = 1, 2, \dots, M$ ,其中M为对F中所有空间小文件数据的访问总次数;

[0015] 统计每个空间小文件数据 $f_i$ 在访问日志序列R中出现的次数 $\lambda_i$ ,以 $\lambda_i$ 为该空间小文件数据 $f_i$ 的访问热度;

[0016] 步骤1.2,根据空间小文件数据访问热度提取被频繁访问的空间小文件数据,实现如下,

[0017] 输入预设判别参数 $\lambda$ ,

[0018] 若空间小文件数据集F中空间小文件数据 $f_i$ 的访问热度 $\lambda_i > \lambda$ ,则空间小文件数据 $f_i$ 为频繁访问的空间小文件数据,否则 $f_i$ 属于非频繁访问的空间小文件数据;

[0019] 步骤1.3,根据步骤1.2所得频繁访问的空间小文件数据构成空间小文件数据集的子集,实现如下,

[0020] 设所有频繁访问的空间小文件数据所构成子集为 $F_1 = \{f_{1_1}, f_{2_1}, \dots, f_{N_1}\}$ ,其中 $N_1$ 为频

繁访问的空间小文件数据总个数,第*i<sub>1</sub>*、*j<sub>1</sub>*个频繁访问的空间小文件数据分别标记为*f<sub>i<sub>1</sub></sub>*和*f<sub>j<sub>1</sub></sub>*,*i<sub>1</sub>*,*j<sub>1</sub>*∈[1,*N<sub>1</sub>*];

[0021] 步骤2,从访问日志信息中提取频繁访问的空间小文件数据子集的访问序列,包括按照时间先后顺序形成访问序列*R<sub>1</sub>*= $\{f_{a_{11}}, f_{a_{21}}, \dots, f_{a_{M_1}}\}$ ,*A<sub>1</sub>*=(*a<sub>11</sub>*,*a<sub>21</sub>*, $\cdots$ ,*a<sub>M\_1</sub>*)为频繁访问空间小文件数据访问序列向量,*a<sub>ij</sub>*∈[1,*N<sub>1</sub>*],访问序号*t<sub>1</sub>*=(*l<sub>1</sub>*,*2<sub>1</sub>*, $\cdots$ ,*M<sub>1</sub>*),其中*M<sub>1</sub>*为对F<sub>1</sub>中所有频繁访问空间小文件数据的访问总次数;

[0022] 步骤3,利用频繁访问的空间小文件数据子集的访问序列分段计算各频繁访问的空间小文件数据的关联度,并将各频繁访问的空间小文件数据相互之间的关联度数值组成关联矩阵;包括以下子步骤,

[0023] 步骤3.1,根据存储服务器数量m、频繁访问空间小文件数据子集长度N<sub>1</sub>计算频繁访问序列分段长度n=N<sub>1</sub>/m;

[0024] 步骤3.2,根据访问序列分段长度对频繁访问序列进行分段,实现如下,

[0025] 按照访问顺序,将频繁访问空间小文件数据访问序列向量A<sub>1</sub>以n个元素一组分割为若干子向量,表示为A<sub>1</sub>=(S<sub>1</sub>,S<sub>2</sub>, $\cdots$ ,S<sub>l</sub>),其中子向量S<sub>k</sub>=(a<sub>k1</sub>,a<sub>k2</sub>, $\cdots$ ,a<sub>kn</sub>),a<sub>kj</sub>∈[1,*N<sub>1</sub>*],1≤k≤l,1≤j≤n;将A<sub>1</sub>中所有子向量集合记为S,S={S<sub>k</sub>:k∈[1,l]};

[0026] 步骤3.3,计算频繁访问的空间小文件数据相互之间的关联度数值,实现如下,

[0027] 定义函数R<sub>S<sub>k</sub></sub>(*i<sub>1</sub>*,*j<sub>1</sub>*),

$$[0028] R_{S_k}(i_1, j_1) = \begin{cases} 1 & i_1 \in \tilde{S}_k, j_1 \in \tilde{S}_k, i_1 \neq j_1 \\ 0 & \text{其它} \end{cases} \quad 1 \leq i_1 \leq N_1, \quad 1 \leq j_1 \leq N_1, \quad 1 \leq k \leq l$$

[0029] 其中 $\tilde{S}_k=\{a_{k1}, a_{k2}, \dots, a_{kn}\}$ 为S<sub>k</sub>中的所有元素组成的集合;函数R<sub>S<sub>k</sub></sub>(*i<sub>1</sub>*,*j<sub>1</sub>*)表示在长度为n的访问周期内频繁访问的空间小文件数据*f<sub>i<sub>1</sub></sub>*和*f<sub>j<sub>1</sub></sub>*是否具有关联性;

[0030] 定义函数R<sub>S</sub>(*i<sub>1</sub>*,*j<sub>1</sub>*),

$$[0031] R_S(i_1, j_1) = \sum_{k=1}^l R_{S_k}(i_1, j_1) \quad 1 \leq i_1 \leq N_1, \quad 1 \leq j_1 \leq N_1$$

[0032] 其中R<sub>S</sub>(*i<sub>1</sub>*,*j<sub>1</sub>*)表示S对*f<sub>i<sub>1</sub></sub>*和*f<sub>j<sub>1</sub></sub>*的总关联度;

[0033] 步骤3.4,将频繁访问的空间小文件数据相互之间的关联度数值组成关联矩阵R<sub>S</sub>,

$$[0034] R_S = (R_S(i_1, j_1))_{N_1 \times N_1} \quad 1 \leq i_1 \leq N_1, \quad 1 \leq j_1 \leq N_1$$

[0035] 步骤4,对关联矩阵中各元素数值进行大小转换后利用RCM排序算法重排后输出;

[0036] 步骤5,对重排后的关联矩阵利用局部逼近搜索法依次循环寻找m个最佳组合后输出,方法如下,

[0037] 步骤6,利用步骤5所得最佳组合对频繁访问的空间小文件数据进行分布存储,以及对非频繁访问的空间小文件数据根据空间位置相邻关系分开存储。

[0038] 而且,步骤4包括以下子步骤,

[0039] 步骤4.1,获取关联矩阵中元素最大值,包括遍历关联矩阵所有元素值,并获取最

大值 $R_{\max}$ ；

[0040] 步骤4.2,对关联矩阵元素数值进行大小转换,包括遍历关联矩阵所有元素值,并执行操作 $R_S(i_1, j_1) = R_{\max} - R_S(i_1, j_1)$ ;

[0041] 步骤4.3,利用标准RCM排序算法对关联矩阵进行重排。

[0042] 而且,步骤5包括以下子步骤,

[0043] 步骤5.1,初始化当前迭代次数 $d=1$ ;

[0044] 步骤5.2,采用局部逼近搜索法寻找一个最佳组合,包括在当前的矩阵中寻找一个 $n \times n$ 的块,使得该矩阵中 $n \times n$ 块内对应的矩阵元素值最大,相应的n个文件构成一个最佳组合;第一次执行步骤5.2时,当前的矩阵为步骤4所得重排后的关联矩阵;后续执行步骤5.2时,当前的矩阵为前一次迭代所得的矩阵;

[0045] 步骤5.3,在本次迭代执行步骤5.2搜索得到一个由n个文件组成的一个最佳组合后,将关联矩阵中对应n个文件的关联矩阵元素删除,得到 $(N_1-d_n) \times (N_1-d_n)$ 的矩阵;

[0046] 步骤5.4,判断是否 $d=m-1$ ,否则令 $d=d+1$ ,以本次迭代执行步骤5.3所得 $(N_1-d_n) \times (N_1-d_n)$ 的矩阵为当前的矩阵,返回步骤5.2进行下一次迭代继续搜索下一个最近组合,是则停止搜索,共得到m个最佳组合。

[0047] 本发明还相应提供一种基于访问日志信息的空间小文件数据分布存储系统,包括以下单元,

[0048] 空间小文件数据集预处理单元(100),用于将任一种空间小文件数据类型的空间小文件数据集,按照访问频率不同分成频繁访问的子集和非频繁访问的子集;包括以下模块,空间小文件数据访问频率统计模块(101),用于获取各空间小文件数据访问热度,实现如下,

[0049] 设空间小文件数据集为 $F = \{f_1, f_2, \dots, f_N\}$ ,包含空间小文件数据 $f_1, f_2, \dots, f_N$ ,其中N为空间小文件数据的总个数,第i个空间小文件数据标记为 $f_i, i=1, 2, \dots, N$ ;

[0050] 设访问日志信息中记录依次访问了空间小文件数据 $f_{a_1}, f_{a_2}, \dots, f_{a_M}$ ,空间小文件数据的访问日志序列为 $R = \{f_{a_1}, f_{a_2}, \dots, f_{a_M}\}$ , $A = (a_1, a_2, \dots, a_M)$ 为空间小文件数据访问序列向量, $a_t \in [1, N]$ ,访问序号 $t=1, 2, \dots, M$ ,其中M为对F中所有空间小文件数据的访问总次数;

[0051] 统计每个空间小文件数据 $f_i$ 在访问日志序列R中出现的次数 $\lambda_i$ ,以 $\lambda_i$ 为该空间小文件数据 $f_i$ 的访问热度;

[0052] 频繁访问空间小文件数据集提取模块(102),用于根据空间小文件数据访问热度提取被频繁访问的空间小文件数据,实现如下,

[0053] 输入预设判别参数 $\lambda$ ,

[0054] 若空间小文件数据集F中空间小文件数据 $f_i$ 的访问热度 $\lambda_i > \lambda$ ,则空间小文件数据 $f_i$ 为频繁访问的空间小文件数据,否则 $f_i$ 属于非频繁访问的空间小文件数据;

[0055] 频繁访问空间小文件子集构建模块(103),用于根据频繁访问空间小文件数据集提取模块(102)所得频繁访问的空间小文件数据构成空间小文件数据集的子集,实现如下,

[0056] 设所有频繁访问的空间小文件数据所构成子集为 $F_1 = \{f_{i_1}, f_{i_2}, \dots, f_{i_{N_1}}\}$ ,其中 $N_1$ 为频繁访问的空间小文件数据总个数,第 $i_1, j_1$ 个频繁访问的空间小文件数据分别标记为 $f_{i_1}$ 和

$f_{j_1}$ ,  $i_1, j_1 \in [1, N_1]$ ;

[0057] 空间小文件数据访问向量获取单元(200),用于从访问日志信息中提取频繁访问的空间小文件数据子集的访问序列,包括按照时间先后顺序形成访问序列 $R_t = \{f_{a_0}, f_{a_1}, \dots, f_{a_{M_1}}\}$ ,  
 $A_1 = (a_{1_1}, a_{2_1}, \dots, a_{M_1})$ 为频繁访问空间小文件数据访问序列向量, $a_{t_1} \in [1, N_1]$ ,访问序号 $t_1 = (1_1, 2_1, \dots, M_1)$ ,其中 $M_1$ 为对F<sub>1</sub>中所有频繁访问空间小文件数据的访问总次数;

[0058] 空间小文件数据访问关联矩阵计算单元(300),用于利用频繁访问的空间小文件数据子集的访问序列分段计算各频繁访问的空间小文件数据的关联度,并将各频繁访问的空间小文件数据相互之间的关联度数值组成关联矩阵;包括以下模块,频繁访问序列分段长度计算模块(301),用于根据存储服务器数量m、频繁访问空间小文件数据子集长度N<sub>1</sub>计算频繁访问序列分段长度n=N<sub>1</sub>/m;

[0059] 存储服务器数量参数m由外部输入。

[0060] 频繁访问序列分段模块(302),用于根据访问序列分段长度对频繁访问序列进行分段,实现如下,

[0061] 按照访问顺序,将频繁访问空间小文件数据访问序列向量A<sub>1</sub>以n个元素一组分割为若干子向量,表示为A<sub>1</sub>= (S<sub>1</sub>, S<sub>2</sub>, ..., S<sub>l</sub>),其中子向量S<sub>k</sub>=(a<sub>k1</sub>, a<sub>k2</sub>, ..., a<sub>kn</sub>), a<sub>kj</sub> ∈ [1, N<sub>1</sub>], 1 ≤ k ≤ l, 1 ≤ j ≤ n;将A<sub>1</sub>中所有子向量集合记为S,S= {S<sub>k</sub>:k ∈ [1, l]}

[0062] 空间小文件数据关联度计算模块(303),用于计算频繁访问的空间小文件数据相互之间的关联度数值,实现如下,

[0063] 定义函数 $R_{S_k}(i_1, j_1)$ ,

$$[0064] R_{S_k}(i_1, j_1) = \begin{cases} 1 & i_1 \in \tilde{S}_k, j_1 \in \tilde{S}_k, i_1 \neq j_1, 1 \leq i_1 \leq N_1, 1 \leq j_1 \leq N_1, 1 \leq k \leq l \\ 0 & \text{其它} \end{cases}$$

[0065] 其中 $\tilde{S}_k = \{a_{k1}, a_{k2}, \dots, a_{kn}\}$ 为S<sub>k</sub>中的所有元素组成的集合;函数 $R_{S_k}(i_1, j_1)$ 表示在长度为n的访问周期内频繁访问的空间小文件数据 $f_{i_1}$ 和 $f_{j_1}$ 是否具有关联性;

[0066] 定义函数R<sub>S</sub>(i<sub>1</sub>, j<sub>1</sub>),

$$[0067] R_S(i_1, j_1) = \sum_{k=1}^l R_{S_k}(i_1, j_1) \quad 1 \leq i_1 \leq N_1, 1 \leq j_1 \leq N_1$$

[0068] 其中R<sub>S</sub>(i<sub>1</sub>, j<sub>1</sub>)表示S对 $f_{i_1}$ 和 $f_{j_1}$ 的总关联度;

[0069] 空间小文件数据关联矩阵生成模块(304),用于将频繁访问的空间小文件数据相互之间的关联度数值组成关联矩阵R<sub>S</sub>,

$$[0070] R_S = (R_S(i_1, j_1))_{N_1 \times N_1} \quad 1 \leq i_1 \leq N_1, 1 \leq j_1 \leq N_1$$

[0071] 关联矩阵转换重排单元(400),用于对关联矩阵中各元素数值进行大小转换后利用RCM排序算法重排后输出;

[0072] 关联矩阵最佳组合搜索单元(500),用于对重排后的关联矩阵利用局部逼近搜索法寻找最佳组合;

[0073] 空间小文件数据分布存储单元(600),用于利用关联矩阵最佳组合搜索单元(500)

所得最佳组合对频繁访问的空间小文件数据进行分布存储,以及对非频繁访问的空间小文件数据根据空间位置相邻关系分开存储。

[0074] 而且,关联矩阵转换重排单元(400)包括以下模块,

[0075] 关联矩阵元素最大值获取模块(401),用于获取关联矩阵中元素最大值,包括遍历关联矩阵所有元素值,并获取最大值R<sub>max</sub>;

[0076] 关联矩阵元素值大小转换模块(402),用于对关联矩阵元素数值进行大小转换,包括遍历关联矩阵所有元素值,并执行操作R<sub>S</sub>(i<sub>1</sub>, j<sub>1</sub>) = R<sub>max</sub> - R<sub>S</sub>(i<sub>1</sub>, j<sub>1</sub>) ;

[0077] 关联矩阵重排模块(403),用于利用标准RCM排序算法对关联矩阵进行重排。

[0078] 而且,关联矩阵最佳组合搜索单元(500)包括以下模块,

[0079] 初始化模块,用于初始化当前迭代次数d=1;

[0080] 最佳组合搜索模块,用于采用局部逼近搜索法寻找一个最佳组合,包括在当前的矩阵中寻找一个n×n的块,使得该矩阵中n×n块内对应的矩阵元素值最大,相应的n个文件构成一个最佳组合;最佳组合搜索模第一次工作时,当前的矩阵为关联矩阵转换重排单元(400)所得重排后的关联矩阵;最佳组合搜索模后续工作时,当前的矩阵为前一次迭代所得的矩阵;

[0081] 矩阵更新模块,用于在最佳组合搜索模块进行本次迭代工作搜索得到一个由n个文件组成的一个最佳组合后,将关联矩阵中对应n个文件的关联矩阵元素删除,得到(N<sub>1</sub>-dn) × (N<sub>1</sub>-dn) 的矩阵;

[0082] 判断输出模块,用于判断是否d=m-1,否则令d=d+1,以矩阵更新模块本次迭代工作所得(N<sub>1</sub>-dn) × (N<sub>1</sub>-dn) 的矩阵为当前的矩阵,命令最佳组合搜索模块进行下一次迭代工作继续搜索下一个最近组合,是则停止搜索,共得到m个最佳组合。

[0083] 本发明具有的有益效果是:空间小文件数据由于数量巨大,但用户访问行为存在聚集性,大部分请求集中在少部分空间小文件数据,为此本发明通过对空间小文件数据进行访问热度分类后,对频繁访问的空间小文件数据利用访问日志信息计算其相互之间的关联度,并在组成关联矩阵后通过局部逼近搜索法寻找最佳分布存储组合方案,并对不同热度的空间小文件数据采用不同的方案分布存储,在有限的计算资源消耗下,实现海量空间小文件数据的最优分布存储,达到提高其并行访问性能,改善空间信息系统的服务能力的目的。因此,本发明能够降低服务器内部空间小文件数据访问时的并发率,从而最终获得服务器间高的空间小文件数据并行访问率,提高了空间小文件数据服务性能,且减少计算数据量,效率较高,具有较好的工程实践性,可应用于大规模分布式环境下空间小文件数据的分布式存储技术领域。

## 附图说明

[0084] 图1是本发明实施例中系统结构示意图。

[0085] 图2是本发明实施例中空间小文件数据集预处理单元100结构示意图。

[0086] 图3是本发明实施例中空间小文件数据访问关联矩阵计算单元300结构示意图。

[0087] 图4是本发明实施例中关联矩阵转换重排单元400结构示意图。

[0088] 图5是本发明实施例中方法流程图。

## 具体实施方式

[0089] 分布式环境下,对空间小文件数据的访问难以通过对数据的分块分布存储实现对其的并行访问,因此需要分析各个空间小文件数据之间的相互关系,以实现在对空间小文件数据进行访问时,尽可能的使所请求的空间小文件数据存储在不同的存储服务器中,以最大可能的实现对空间小文件数据的并行获取,从而提高空间信息服务系统的性能。

[0090] 由于空间小文件数据数量巨大,大规模的空间小文件数据的存储组合优化计算复杂度高,搜索时间开销大,为此需要对空间小文件数据进行热度分类,并根据不同热度分别采用不同的方法获取最佳存储组合方案。

[0091] 以下对本发明技术方案的具体实施提供详细建议说明。

[0092] 本发明所述空间小文件数据,包含空间数据类型及空间坐标位置,每个空间小文件数据较小,不适于将其继续分成多份并分别存储在不同的服务器上以提高其并行访问效率。所述访问日志信息是由对应的空间信息服务系统按照时间顺序记录的,各个客户端应用访问空间小文件数据的日志信息,包括所访问的空间小文件数据类型和坐标。所述的访问日志信息由空间信息服务系统在运行过程中记录,形式包括但不限于文件、数据库。

[0093] 所述空间小文件数据包含不同类型,包括但不限于SRTM30 (the 30m of global Shuttle Radar Topography Mission terrain data files)、SRTM90。

[0094] 所述的一种基于访问日志信息的空间小文件数据分布存储方法及系统,针对每种类型的空间小文件数据分别处理,所述方法和系统对不同类型的空间小文件数据处理过程相同。

[0095] 如图5所示,本发明的方法所采用的技术方案是:一种基于访问日志信息的空间小文件数据分布存储方法及系统,对任一种空间小文件数据类型,执行包括以下步骤:

[0096] (1) 频繁访问空间小文件数据子集提取:将空间小文件数据集,按照访问频率不同分成频繁访问子集和非频繁访问子集;包括以下子步骤,

[0097] ①获取各空间小文件数据访问热度。

[0098] 设空间小文件数据集为 $F = \{f_1, f_2, \dots, f_N\}$ ,包含空间小文件数据 $f_1, f_2, \dots, f_N$ ,其中N为空间小文件数据的总个数,第i个空间小文件数据标记为 $f_i, i=1, 2, \dots, N$ 。

[0099] 设访问日志信息中记录依次访问了空间小文件数据 $f_{a_1}, f_{a_2}, \dots, f_{a_M}$ ,空间小文件数据的访问日志序列为 $R = \{f_{a_1}, f_{a_2}, \dots, f_{a_M}\}$ ,对应的称 $A = (a_1, a_2, \dots, a_M)$ 为空间小文件数据访问序列向量, $a_t \in [1, N]$ (访问序号 $t = 1, 2, \dots, M$ ),其中M为对F中所有空间小文件数据的访问总次数。

[0100] 统计每个 $f_i$  ( $f_i \in F$ ) 在访问日志序列R中出现的次数 $\lambda_i$ ,则 $\lambda_i$ 为该空间小文件数据 $f_i$ 的访问热度。

[0101] ②根据空间小文件数据访问热度提取被频繁访问的空间小文件数据。

[0102] 输入频繁访问空间小文件数据的预设判别参数 $\lambda$ ,

[0103] 若空间小文件数据集F中空间小文件数据 $f_i$ 的访问热度 $\lambda_i > \lambda$ ,则空间小文件数据 $f_i$ 为频繁访问的空间小文件数据,否则, $f_i$ 属于非频繁访问的空间小文件数据。

[0104] ③根据②所获得的频繁访问空间小文件数据构成空间小文件数据集F的子集

[0105] 若设所有频繁访问的空间小文件数据构成的子集为  $F_1 = \{f_{1_1}, f_{2_1}, \dots, f_{N_1}\}$ ，其中  $N_1$  为频繁访问的空间小文件数据总个数，第  $i_1, j_1$  个频繁访问的空间小文件数据分别标记为  $f_{i_1}$  和  $f_{j_1}$ ， $i_1, j_1 \in [1, N_1]$ 。

[0106] 同样可设非频繁访问的空间小文件数据集为  $F_2 = \{f_{1_2}, f_{2_2}, \dots, f_{N_2}\}$ ，其中  $N_2$  为非频繁访问的空间小文件数据总个数。其中  $N_1 + N_2 = N$ 。

[0107] (2) 频繁访问空间小文件数据子集访问序列提取：从访问日志信息中提取频繁访问的空间小文件数据子集的访问序列；

[0108] 访问日志信息记录了空间数据的坐标，不同的坐标代表了不同的数据。因此可从访问日志信息按照访问时间先后顺序提取所访问的空间小文件数据的坐标信息。具体实施时，具体信息提取方式可依据访问日志信息的记录格式确定。坐标信息为空间小文件数据的空间经纬度坐标。

[0109] 根据频繁访问空间小文件数据子集提取访问序列子集，实现如下，

[0110] 对访问日志信息中空间小文件数据按照访问时间先后顺序，取其中频繁访问的空间小文件数据，形成频繁访问的空间小文件数据子集的访问序列  $R_1 = \{f_{a_{1_1}}, f_{a_{2_1}}, \dots, f_{a_{M_1}}\}$ ，对应的称  $A_1 = (a_{1_1}, a_{2_1}, \dots, a_{M_1})$  为频繁访问空间小文件数据访问序列向量， $a_{t_1} \in [1, N_1]$  (访问序号  $t_1 = (1_1, 2_1, \dots, M_1)$ )，其中  $M_1$  为对  $F_1$  中所有频繁访问空间小文件数据的访问总次数。

[0111] (3) 关联度计算与关联矩阵获取：利用频繁访问的空间小文件数据子集的访问序列分段计算各频繁访问的空间小文件数据的关联度，并将各频繁访问的空间小文件数据相互之间的关联度数值组成关联矩阵；包括以下子步骤，

[0112] ①根据存储服务器数量、频繁访问空间小文件数据子集长度  $N_1$  计算频繁访问序列分段长度  $n$ 。

[0113] 存储服务器数量  $m$  可由外部输入，例如由系统配置文件输入。

[0114] 通过公式  $n = N_1 / m$  计算得到频繁访问序列分段长度  $n$ 。

[0115] ②根据访问序列分段长度对频繁访问序列进行分段。

[0116] 按照频繁访问空间小文件数据的访问顺序，将频繁访问空间小文件数据访问序列向量  $A_1$  以  $n$  个元素一组分割为若干子向量，表示为： $A_1 = (S_1, S_2, \dots, S_l)$ ，其中子向量  $S_k = (a_{k1}, a_{k2}, \dots, a_{kn})$ ， $a_{kj} \in [1, N_1]$ ， $1 \leq k \leq l$ ， $1 \leq j \leq n$  为  $A_1$  中的长度为  $n$  的子向量。将  $A_1$  中所有长度为  $n$  的访问向量集合记为  $S$ ，即  $A_1$  中所有子向量的集合  $S = \{S_k : k \in [1, l]\}$ 。

[0117] ③计算频繁访问的空间小文件数据相互之间的关联度数值

[0118] 首先计算各分段内空间小文件数据相互关联程度，对  $\forall S_k \in S$ ，定义函数：

[0119]

$$R_{S_k}(i_1, j_1) = \begin{cases} 1 & i_1 \in \tilde{S}_k, j_1 \in \tilde{S}_k, i_1 \neq j_1, 1 \leq i_1 \leq N_1, 1 \leq j_1 \leq N_1, 1 \leq k \leq l \\ 0 & \text{其它} \end{cases} \quad (1)$$

[0120] 其中  $\tilde{S}_k = \{a_{k1}, a_{k2}, \dots, a_{kn}\}$  为  $S_k$  中的所有元素组成的集合。 $R_{S_k}(i_1, j_1)$  函数的意义在于，在长度为  $n$  的访问周期内频繁访问的空间小文件数据  $f_{i_1}$  和  $f_{j_1}$  是否具有关联性。

[0121] 在此基础上,定义函数:

[0122]

$$R_s(i_1, j_1) = \sum_{k=1}^l R_{S_k}(i_1, j_1) \quad 1 \leq i_1 \leq N_1, \quad 1 \leq j_1 \leq N_1 \quad (2)$$

[0123] 则  $R_s(i_1, j_1)$  表示 S 对  $f_{i_1}$  和  $f_{j_1}$  的总关联度。

[0124] ④将频繁访问的空间小文件数据相互之间的关联度数值组成关联矩阵。

[0125] 将所有  $N_1$  个频繁访问的空间小文件数据相互之间的关联度用矩阵表示,即可得到如下的关联矩阵  $R_s$ 。

[0126]

$$R_s = (R_s(i_1, j_1))_{N_1 \times N_1} \quad 1 \leq i_1 \leq N_1, \quad 1 \leq j_1 \leq N_1 \quad (3)$$

[0127] (4) 关联矩阵转换和重排输出:对关联矩阵中各元素数值进行大小转换后利用RCM排序算法重排后输出;包括以下子步骤,

[0128] ①获取关联矩阵中元素最大值。

[0129] 遍历关联矩阵所有元素值,并获取最大值  $R_{\max}$ 。

[0130] ②对关联矩阵元素数值进行大小转换。

[0131] 遍历关联矩阵所有元素值,并执行操作  $R_s(i_1, j_1) = R_{\max} - R_s(i_1, j_1)$ ,对关联矩阵元素值大小进行转换。

[0132] ③利用标准RCM排序算法对关联矩阵进行重排。

[0133] 采用标准RCM排序算法对关联矩阵进行重排,目标是将关联矩阵中非零元素集中在对角线附近。将重排后的新矩阵记为 P。标准RCM排序算法为现有技术,具体实施时可参考文献 Gibbs N E, Poole W G, Stockmeyer P K. An algorithm for reducing the bandwidth and profile of a sparse matrix. SIAM Journal on Numerical Analysis, 1976, 13 (2) : 236–250。

[0134] (5) 最优存储分布组合搜索输出。

[0135] 对(4)所得重排后的关联矩阵利用局部逼近搜索法寻找最佳组合以便获得对该子集空间小文件数据的最高并行访问率。局部逼近搜索法为现有技术,具体实施时可参考文献 XIA Kai, Wen-zhan. Adaptive Genetic Algorithm Based on Local Search Mechanism Quickly Solving TSP. Journal of Zhejiang Institute of Science and Technology, 2014, 31 (3)。

[0136] 根据(4)所得重排后的关联矩阵,迭代使用局部逼近搜索法,每执行局部逼近搜索一次得到一个包含 n 个文件的最佳组合,最后可得到 m 个组合,以便后续分别存储在 m 个存储服务器上。每个组合由 n 个文件组成,n 个文件相互之间的关联度值在矩阵中对应了一个  $n \times n$  的块;具体实现如下:

[0137] ①初始化当前迭代次数  $d=1$ ;

[0138] ②采用局部逼近搜索法寻找一个最佳组合,包括在当前的矩阵中寻找一个  $n \times n$  的块,使得该矩阵中  $n \times n$  块内对应的矩阵元素值最大,相应的 n 个文件构成一个最佳组合;

[0139] 第一次执行②时,当前的矩阵为(4)所得重排后的关联矩阵,矩阵大小为  $N_1 \times N_1$ ;后续执行②时,当前的矩阵为前一次迭代所得  $(N_1 - (d-1)n) \times (N_1 - (d-1)n)$  的矩阵;

[0140] ③在本次迭代执行②搜索得到一个由n个文件组成最佳组合后,将关联矩阵中对应n个文件的关联矩阵元素删除,得到 $(N_1-d_n) \times (N_1-d_n)$ 的矩阵,缩小继续搜索的关联矩阵大小,可以节省搜索时间;

[0141] ④判断是否 $d=m-1$ ,否则令 $d=d+1$ ,以本次迭代执行③所得 $(N_1-d_n) \times (N_1-d_n)$ 的矩阵(在 $d=d+1$ 后即 $(N_1-(d-1)n) \times (N_1-(d-1)n)$ )为基础作为当前的矩阵,返回②进行下一次迭代继续搜索下一个最近组合,是则停止搜索,当前矩阵为 $n \times n$ ,即可直接得到最后一个由n个文件组成最佳组合,和依次循环搜索 $m-1$ 次得到的最佳组合一起,共得到m个最佳组合。

[0142] (6) 空间小文件数据分布存储:利用(5)最终得到的最佳组合对频繁访问的空间小文件数据进行分布存储,以及对非频繁访问的空间小文件数据根据其空间位置相邻关系分开存储。

[0143] 实施例按照获取的最佳组合对频繁访问的空间小文件数据进行分布存储以获取该空间小文件数据的最高并行访问率。

[0144] 通过步骤(5)获得的最佳分布存储组合的空间小文件数据,具有相互之间关联度低的特点(即经过矩阵元素值大小转换后在重排后的关联矩阵中对应元素值大),则可将一个最佳组合中的所有空间小文件数据存储在一个服务器中,以此获得相互之间低的并发访问要求(即实现了不同服务器之间高的并行访问率)。

[0145] 根据空间小文件数据的坐标信息,实施例对非频繁访问空间小文件数据根据其空间位置相关性进行分开存储。

[0146] 针对步骤(1)的 $F_2$ ,按照位置相邻,则存入不同服务器的原则,将非频繁访问的空间小文件数据存储在服务器中。

[0147] 根据空间数据访问特征,空间数据访问具有空间访问路劲的连续性,因此,相邻的空间数据具有较高的概率被同时被访问,因此,存储在不同的服务器可以减少并发,提高并行率。

[0148] 具体实施时,所述的频繁访问空间小文件数据的判别参数、关联矩阵RCM排序算法参数、存储服务器数量可由外部输入或由本领域技术人员预先设定。

[0149] 参见图1,本发明还相应提供一种基于访问日志信息的空间小文件数据分布存储系统,包括以下单元,

[0150] 空间小文件数据集预处理单元(100),用于将任一种空间小文件数据类型的空间小文件数据集,按照访问频率不同分成频繁访问的子集和非频繁访问的子集;参见图2,包括以下模块,空间小文件数据访问频率统计模块(101),用于获取各空间小文件数据访问热度,实现如下,

[0151] 设空间小文件数据集为 $F = \{f_1, f_2, \dots, f_N\}$ ,包含空间小文件数据 $f_1, f_2, \dots, f_N$ ,其中N为空间小文件数据的总个数,第i个空间小文件数据标记为 $f_i, i=1, 2, \dots, N$ ;

[0152] 设访问日志信息中记录依次访问了空间小文件数据 $f_{a_1}, f_{a_2}, \dots, f_{a_M}$ ,空间小文件数据的访问日志序列为 $R = \{f_{a_1}, f_{a_2}, \dots, f_{a_M}\}$ , $A = (a_1, a_2, \dots, a_M)$ 为空间小文件数据访问序列向量, $a_t \in [1, N]$ ,访问序号 $t=1, 2, \dots, M$ ,其中M为对F中所有空间小文件数据的访问总次数;

[0153] 统计每个空间小文件数据 $f_i$ 在访问日志序列R中出现的次数 $\lambda_i$ ,以 $\lambda_i$ 为该空间小文

件数据 $f_i$ 的访问热度；

[0154] 频繁访问空间小文件数据集提取模块(102)，用于根据空间小文件数据访问热度提取被频繁访问的空间小文件数据，实现如下，

[0155] 输入预设判别参数 $\lambda$ ，

[0156] 若空间小文件数据集F中空间小文件数据 $f_i$ 的访问热度 $\lambda_i > \lambda$ ，则空间小文件数据 $f_i$ 为频繁访问的空间小文件数据，否则 $f_i$ 属于非频繁访问的空间小文件数据；

[0157] 频繁访问空间小文件子集构建模块(103)，用于根据频繁访问空间小文件数据集提取模块(102)所得频繁访问的空间小文件数据构成空间小文件数据集的子集，实现如下，

[0158] 设所有频繁访问的空间小文件数据所构成子集为 $F_1 = \{f_{i_1}, f_{j_1}, \dots, f_{N_1}\}$ ，其中 $N_1$ 为频繁访问的空间小文件数据总个数，第 $i_1, j_1$ 个频繁访问的空间小文件数据分别标记为 $f_{i_1}$ 和 $f_{j_1}$ ， $i_1, j_1 \in [1, N_1]$ ；

[0159] 空间小文件数据访问向量获取单元(200)，用于从访问日志信息中提取频繁访问的空间小文件数据子集的访问序列，包括按照时间先后顺序形成访问序列 $R_1 = \{f_{a_{t_1}}, f_{a_{t_2}}, \dots, f_{a_{t_{M_1}}}\}$ ， $A_1 = (a_{1_1}, a_{2_1}, \dots, a_{M_1})$ 为频繁访问空间小文件数据访问序列向量， $a_{t_i} \in [1, N_1]$ ，访问序号 $t_1 = (1_1, 2_1, \dots, M_1)$ ，其中 $M_1$ 为对 $F_1$ 中所有频繁访问空间小文件数据的访问总次数；

[0160] 空间小文件数据访问关联矩阵计算单元(300)，用于利用频繁访问的空间小文件数据子集的访问序列分段计算各频繁访问的空间小文件数据的关联度，并将各频繁访问的空间小文件数据相互之间的关联度数值组成关联矩阵；参见图3，包括以下模块，频繁访问序列分段长度计算模块(301)，用于根据存储服务器数量m、频繁访问空间小文件数据子集长度 $N_1$ 计算频繁访问序列分段长度 $n = N_1/m$ ；

[0161] 频繁访问序列分段模块(302)，用于根据访问序列分段长度对频繁访问序列进行分段，实现如下，

[0162] 按照访问顺序，将频繁访问空间小文件数据访问序列向量 $A_1$ 以n个元素一组分割为若干子向量，表示为 $A_1 = (S_1, S_2, \dots, S_l)$ ，其中子向量 $S_k = (a_{k1}, a_{k2}, \dots, a_{kn})$ ， $a_{kj} \in [1, N_1]$ ， $1 \leq k \leq l$ ， $1 \leq j \leq n$ ；将 $A_1$ 中所有子向量集合记为 $S$ ， $S = \{S_k : k \in [1, l]\}$ ；

[0163] 空间小文件数据关联度计算模块(303)，用于计算频繁访问的空间小文件数据相互之间的关联度数值，实现如下，

[0164] 定义函数 $R_{S_k}(i_1, j_1)$ ，

$$[0165] R_{S_k}(i_1, j_1) = \begin{cases} 1 & i_1 \in \tilde{S}_k, j_1 \in \tilde{S}_k, i_1 \neq j_1 \\ 0 & \text{其它} \end{cases} \quad 1 \leq i_1 \leq N_1, \quad 1 \leq j_1 \leq N_1, \quad 1 \leq k \leq l$$

[0166] 其中 $\tilde{S}_k = \{a_{k1}, a_{k2}, \dots, a_{kn}\}$ 为 $S_k$ 中的所有元素组成的集合；函数 $R_{S_k}(i_1, j_1)$ 表示在长度为n的访问周期内频繁访问的空间小文件数据 $f_{i_1}$ 和 $f_{j_1}$ 是否具有关联性；

[0167] 定义函数 $R_S(i_1, j_1)$ ，

$$[0168] R_S(i_1, j_1) = \sum_{k=1}^l R_{S_k}(i_1, j_1) \quad 1 \leq i_1 \leq N_1, \quad 1 \leq j_1 \leq N_1$$

- [0169] 其中 $R_s(i_1, j_1)$ 表示S对 $f_{i_1}$ 和 $f_{j_1}$ 的总关联度；
- [0170] 空间小文件数据关联矩阵生成模块(304)，用于将频繁访问的空间小文件数据相互之间的关联度数值组成关联矩阵 $R_s$ ，
- [0171]  $R_s = (R_s(i_1, j_1))_{N_1 \times N_1} \quad 1 \leq i_1 \leq N_1, \quad 1 \leq j_1 \leq N_1$
- [0172] 关联矩阵转换重排单元(400)，用于对关联矩阵中各元素数值进行大小转换后利用RCM排序算法重排后输出；
- [0173] 关联矩阵最佳组合搜索单元(500)，用于对重排后的关联矩阵利用局部逼近搜索法寻找最佳组合；
- [0174] 空间小文件数据分布存储单元(600)，用于利用关联矩阵最佳组合搜索单元(500)所得最佳组合对频繁访问的空间小文件数据进行分布存储，以及对非频繁访问的空间小文件数据根据空间位置相邻关系分开存储。
- [0175] 参见图4，关联矩阵转换重排单元(400)进一步包括以下模块，
- [0176] 关联矩阵元素最大值获取模块(401)，用于获取关联矩阵中元素最大值，包括遍历关联矩阵所有元素值，并获取最大值 $R_{max}$ ；
- [0177] 关联矩阵元素值大小转换模块(402)，用于对关联矩阵元素数值进行大小转换，包括遍历关联矩阵所有元素值，并执行操作 $R_s(i_1, j_1) = R_{max} - R_s(i_1, j_1)$ ；
- [0178] 关联矩阵重排模块(403)，用于利用标准RCM排序算法对关联矩阵进行重排。
- [0179] 关联矩阵最佳组合搜索单元(500)包括以下模块，
- [0180] 初始化模块，用于初始化当前迭代次数 $d=1$ ；
- [0181] 最佳组合搜索模块，用于采用局部逼近搜索法寻找一个最佳组合，包括在当前的矩阵中寻找一个 $n \times n$ 的块，使得该矩阵中 $n \times n$ 块内对应的矩阵元素值最大，相应的 $n$ 个文件构成一个最佳组合；最佳组合搜索模第一次工作时，当前的矩阵为关联矩阵转换重排单元(400)所得重排后的关联矩阵；最佳组合搜索模后续工作时，当前的矩阵为前一次迭代所得的矩阵；
- [0182] 矩阵更新模块，用于在最佳组合搜索模块进行本次迭代工作搜索得到一个由 $n$ 个文件组成的最佳组合后，将关联矩阵中对应 $n$ 个文件的关联矩阵元素删除，得到 $(N_1-dn) \times (N_1-dn)$ 的矩阵；
- [0183] 判断输出模块，用于判断是否 $d=m-1$ ，否则令 $d=d+1$ ，以矩阵更新模块本次迭代工作所得 $(N_1-dn) \times (N_1-dn)$ 的矩阵为当前的矩阵，命令最佳组合搜索模块进行下一次迭代工作继续搜索下一个最近组合，是则停止搜索，共得到 $m$ 个最佳组合。
- [0184] 各模块具体实现可与方法具体步骤一致，本发明不予赘述。
- [0185] 本文中所描述的具体实施例仅仅是对本发明精神作举例说明。本发明所属技术领域的技术人员可以对所描述的具体实施例做各种各样的修改或补充或采用类似的方式替代，但并不会偏离本发明的精神或者超越所附权利要求书所定义的范围。

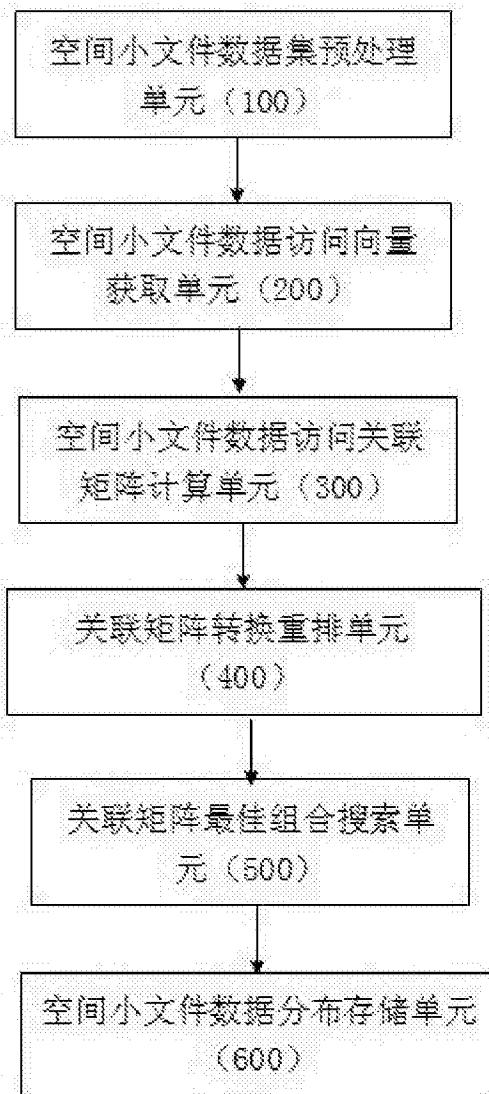


图 1

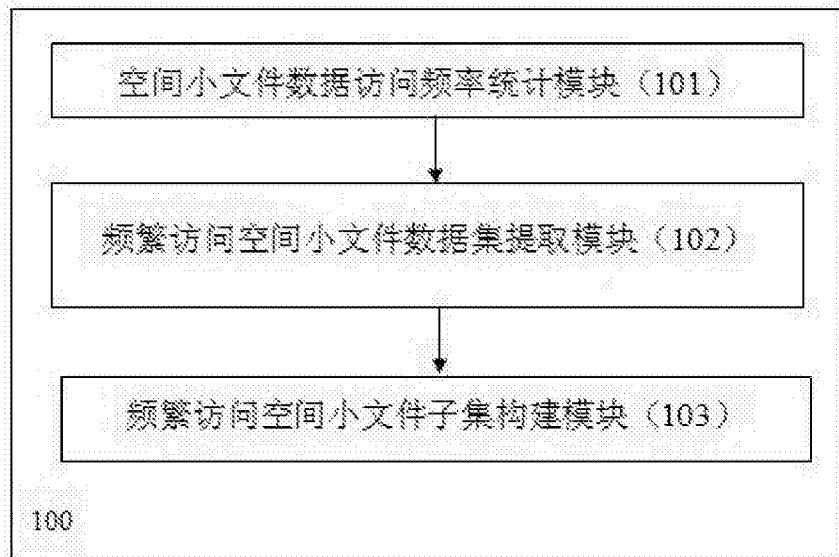


图 2

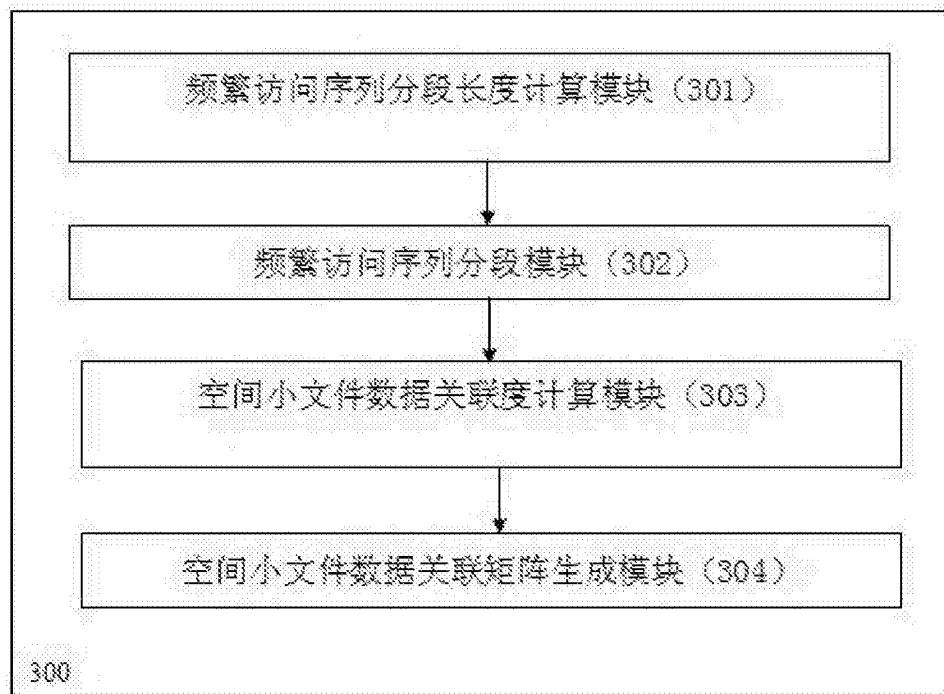


图 3

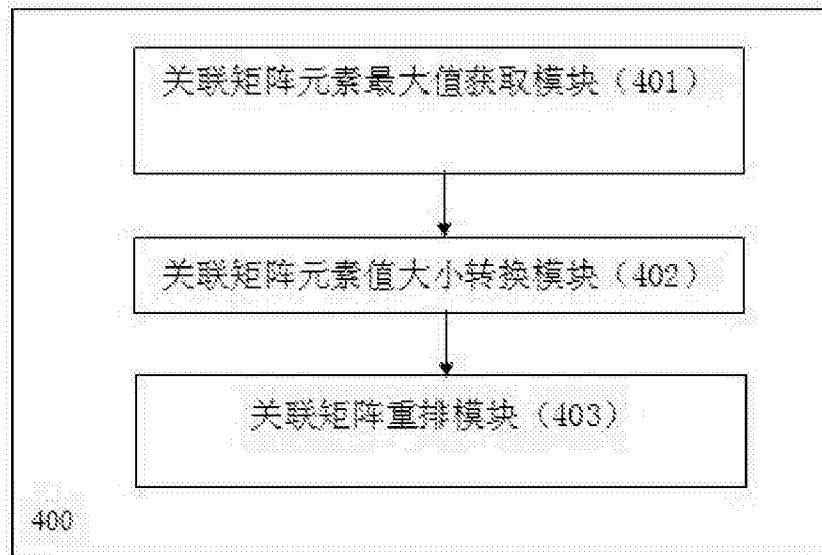


图 4

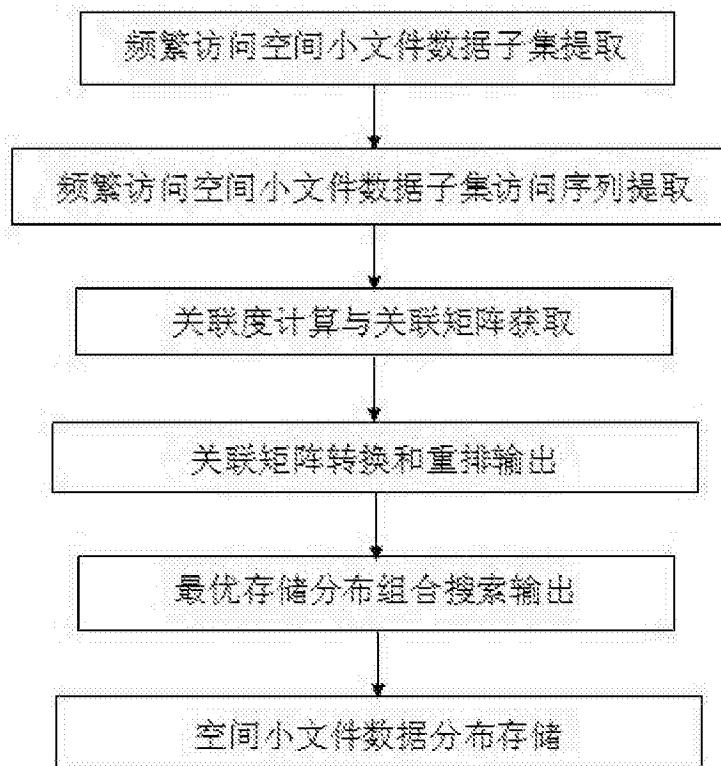


图 5