

DOI: 10.13203/j.whugis20130072

文章编号: 1671-8860(2014)10-1252-05

云环境下的时空数据小文件存储策略

熊 炼¹ 徐正全¹ 王 涛¹ 顾 鑫¹

¹ 武汉大学测绘遥感信息工程国家重点实验室, 湖北 武汉, 430079

摘要: 云环境下的网络应用中存在着海量的时空数据小文件, 针对 HDFS 处理小文件效率不高的问题, 提出了一种结合用户访问规律与数据自身属性的数据处理方案。该方案将用户访问转化为数据请求序列, 根据数据的时空属性提取并构建特征序列, 通过模板匹配找到用户在不同访问模式下的特征模板, 发现模板内文件之间的访问相关性, 合并相关文件。实验结果表明, 该方案有效地提高了系统对小文件的存储读取效率, 减小了网络应用的响应时间。

关键词: 云存储; 小文件; 时空数据; 特征提取; 模板匹配

中图法分类号: P208

文献标志码: A

随着云计算技术的飞速发展, 网络中的数据呈指数级增长。根据权威部门的统计, 世界上约 80% 的信息资源与地理位置有关^[1]。这些海量的时空数据的共同特点是文件小, 数量大, 均含空间、时间和属性这三个要素, 且与时间空间紧密相关。

HDFS 作为 GFS 的一个开源实现, 非常适合部署云存储平台^[2]。针对 HDFS 系统存储海量小文件数据的问题, 已有的解决方案都把焦点放在了分析文件之间的相关性或调整存储系统自身的结构上^[3-9], 能够很好地提高文件存储效率, 但对于网络应用中用户与网络的交互时间、文件的读取效率等并没有过多的关注。研究表明^[10-14], 通过分析用户的访问模式和访问规律, 制定合理的预取方案, 可以有效地减少用户与网络应用的交互时间, 提高读取效率。本文在用户访问行为特征和时空数据的时空特性的基础上, 将用户对网络应用的访问看作是一连串的数据请求序列, 按照数据的时空属性对序列进行参数化表示, 并利用特征提取构建特征序列, 通过模板匹配找到用户在不同访问模式下的特征模板, 发现隐藏在用户访问行为下的文件之间的访问相关性, 合并相关文件。

1 小文件存储策略

1.1 合并算法分析

时空特性是时空数据区别于其他数据的根本

性标志, 它具有以下特点: ① 多源性: 只要与时间和空间紧密相关的数据都可称为时空数据; ② 时空相关性: 数据具有空间局部性和时间上的自相关性。

云环境下的时空数据应用一般都是通过有限的预定义服务或服务组合来实现的, 用户对数据的访问也是通过预定义的服务接口进行。由于服务通常是根据用户选择的参数由服务接口函数对特定时空范围的数据内容进行分析、处理或挖掘, 以提供应用所需要的功能, 这就决定了其对数据文件的访问具有稳定的选择和组合规律。这种规律性为本文通过数据文件访问序列的时空属性分析来发现和提取其中所蕴含的数据访问模式提供了原理基础。合并算法流程如图 1 所示。

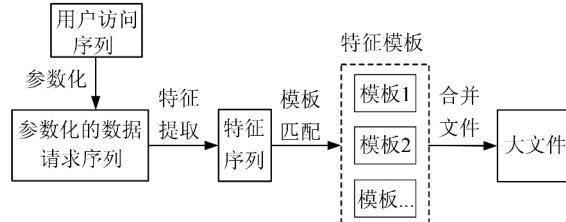


图 1 合并算法流程图

Fig. 1 Flow Chart of Merging Algorithm

1.2 合并算法实现

1.2.1 访问序列参数化与特征提取

假设系统中存储了 m 个时空数据小文件, 每个数据均含位置、时间和类型三种参数, 数据之间

收稿日期: 2013-04-23

项目来源: 国家 973 计划资助项目(2011CB302306); 国家自然科学基金资助项目(41271398)。

第一作者: 熊炼, 博士生。主要从事 Web 数据挖掘及分布式存储研究。E-mail: xionglian302@163.com

可通过参数进行区分。那么,时空数据集可表示为:

$$D = \{d(L_i, S_j, T_k), d \mid L_i \in L, S_j \in S, T_k \in T\} \quad (1)$$

其中, D 表示时空数据小文件集合; L 代表数据源所在的地理位置集; S 代表数据源类型集; T 代表数据产生的时标集。

用户访问时,会根据自身需求选择云平台提供的服务,所选择的服务会通过系统中预定义的服务接口函数对后台数据进行筛选请求和处理,并将最终结果返回给用户。如果云平台提供了 X 种服务,用户访问的第 i 个服务为:

$$ser(i) = pro_i(sel(D, \alpha_i, \beta_i, \gamma_i)), i \in X \quad (2)$$

其中, $pro_i()$ 为数据处理函数; $sel()$ 为数据选择函数; $\alpha_i \in L, \beta_i \in S, \gamma_i \in T$ 为用户选择参数。则一段时间内系统收到若干用户的访问可表示为一系列服务接口函数的组合:

$$f_{ser} = \bigcup_j ser(j), j \in X \quad (3)$$

涉及到的数据请求为:

$$\bigcup_j sel(j), j \in X \quad (4)$$

为了找到用户的访问规律,本文将 $\bigcup_j sel(j)$ 里面的第 n 个数据请求(数据位置为 L_p , 数据源类型为 S_i , 数据的时间标签为 T_u)参数化表示为:

$$f_n = f_n(L_p, S_i, T_u) \quad (5)$$

则 f_{ser} 里所有的数据请求可表示为:

$$f(L, S, T) = f_1, f_2, f_3, \dots, f_{n-1}, f_n \quad (6)$$

根据 Web 应用中用户访问的时空局部性以及数据自身所具有的时空特性,将请求数据的属性 L, S, T 变化与否这种二维逻辑状态作为特征进行提取,通过逐一对比数据请求序列 $f(L, S, T)$ 中任意两个连续的请求 f_n 和 f_{n-1} ,考察其属性 L, S, T 变化与否,并用以下特征状态表示:

$$\left\{ \begin{array}{l} A = \{f_n(L_p, S_i, T_u) \mid f_{n-1}(L_p, S_i, T_u)\} \\ B = \{f_n(L_p, S_i, T_v) \mid f_{n-1}(L_p, S_i, T_u)\} \\ C = \{f_n(L_o, S_i, T_u) \mid f_{n-1}(L_p, S_i, T_u)\} \\ D = \{f_n(L_p, S_j, T_v) \mid f_{n-1}(L_p, S_i, T_u)\} \\ E = \{f_n(L_o, S_j, T_u) \mid f_{n-1}(L_p, S_i, T_u)\} \\ F = \{f_n(L_o, S_i, T_v) \mid f_{n-1}(L_p, S_i, T_u)\} \\ G = \{f_n(L_o, S_j, T_v) \mid f_{n-1}(L_p, S_i, T_u)\} \end{array} \right. \quad (7)$$

其中,变量 $p, o \in L; i, j \in S; u, v \in T$ 。要注意的是,如果两个连续的请求数据其属性完全一样,则表明用户两次访问的是同一个数据。这对存储系统来说,并不会带来任何开销,因为第一次访问的文件已经存在缓存中,所以文中并不考虑这种状

态。

通过上面的特征提取,就可以在特征空间 $I = \{A, B, C, D, E, F, G\}$ 中,把具有多维时空属性的数据请求序列 $f(L, S, T)$ 转换成只有一维的特征序列 $f(I)$,如图 2 所示。



图 2 特征序列示意图

Fig. 2 Schematic Diagram of Feature Sequence

1.2.2 模板匹配与小文件合并

将原始数据请求序列 $f(L, S, T)$ 参数化表示后,通过特征提取并在特征空间中利用模板匹配技术对特征序列进行匹配,找到若干个最佳特征模板,为此给出以下定义。

定义 1 特征状态的数量:包含 n 个数据请求的序列 $f(L, S, T)$,其对应特征序列 $f(I)$ 有 $\varphi = n-1$ 个特征状态。

定义 2 特征模板的大小:特征模板包含 ω 个特征状态,其所对应的数据即为一个合并文件块的大小。

定义 3 特征序列之间的相似度:序列之间的相似度 r 定义为汉明距离,即对应位置特征状态相异的个数。距离越小,相似度越大。

根据以上定义,特征模板的创建过程如下:

1) 统计特征序列 $f(I)$ 中各特征状态的初始概率 P 。

2) 将特征序列 $f(I)$ 按顺序分成长为 ω 、连续但不重叠的 $\eta = [\varphi/\omega]$ 个特征块。

3) 统计并对比任意两两特征块,距离小于 r 的求公共子序列(长度设为 l_{cs}),当拥有同一个公共子序列的块数量 $k \geq [\eta/5]$ (经验值,设置太小会出现非常多的模板,影响文件的读取效率)时,将该公共子序列放入模板 Model 中,余下 $\omega - l_{cs}$ 项利用条件最大熵原理(条件即为 $f(I)$ 中各特征状态的初始概率 P)补充完整,由此生成一个完整的包含 ω 个特征状态的特征模板,并设置 $\eta = \eta - k$ 。

4) 剩余的特征块按照步骤 3) 循环生成模板 Mode2, Mode3, ...。

5) 对拥有公共子序列数量达不到 k 或相似度小于 r 的特征块,当作独立块进行处理,将 $f(I)$ 里所有的特征状态利用条件最大熵原理生成一个新的模板。

特征模板的数量是由特征序列 $f(I)$ 的长度 φ 和拥有距离小于 r 的特征块数量共同决定的,

通过上面模板创建算法循环迭代,直至只剩下独立块,再次利用条件最大熵生成最后一个模板。

将建立好的特征模板映射至原始的数据请求序列 $f(L, S, T)$, 通过分析属于同一模板内时空数量的特点发现用户的访问模式和规律, 如同属性长时间同一位置连续访问, 不同属性短时间邻近区域关联访问等, 最后经 HDFS Client 合并相关数据文件。合并后的小文件需要为其建立文件内部本地索引(Internal Index), 用来存储小文件的长度和偏移位置, 这样, HDFS Client 在处理数据请求时, 就可以直接通过文件内部索引进行定位, 从而方便快捷地获得所需的小文件数据。合并好的大文件应尽可能地放在连续的存储区间, 因为一些原本连续的小文件很可能部分放在了大文件的边缘位置, 当用户访问它时, HDFS Client 需要跨文件甚至跨节点读取。合并后的文件结构如图 3 所示。

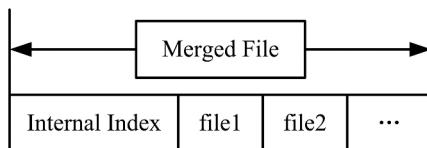


图 3 合并后文件结构图

Fig. 3 Structure Diagram of Merged File

2 实验评估

2.1 实验平台和环境

实验采用由 6 个节点组成的 HDFS 作为云存储平台, 块大小设为系统默认的 64 MB, 副本数量为 3 份。测试内容为文件的存储速度和读取速度, 其中多用户并发通过单 HDFS Client 多进程进行模拟, 每次实验重复 10 次结果取平均, 并与传统 HDFS 和 HAR 归档技术进行比较。

数据文件分布如图 4 所示, 包括多媒体数据、基础地理数据、传感数据和少量的遥感影像数据, 测试数据集总大小为 56.8 GB, 包含 1 829 054 个小文件, 范围从 3.22~2 780 kB, 平均大小为 36.75 kB, 其中 3.22~56 kB 占总体的 95.8%。

2.2 文件存储测试

文件的存储测试是一次性将总大小为 3.96 GB 的 100 000 个小文件数据通过 HDFS Client 写入系统中, 计算文件的平均写入速度, 实验结果如图 5 所示。

从图 5 可以看出, 首先, 不管对文件合并与否, HDFS 系统都是在有两个用户并发访问时写

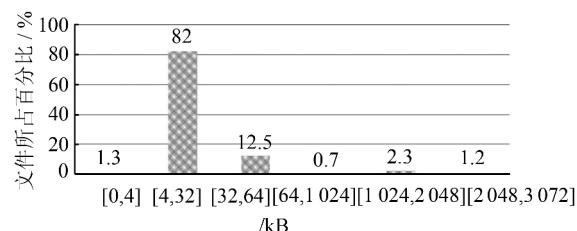


图 4 小文件分布

Fig. 4 Distribution of Small Files

入速率最大, 而且随着并发用户数的增加, 总的传输效率趋于稳定。其次, 由于小文件的合并, 三种算法均有较好的写入速度。但 HAR 合并机制是直接将多个小文件打包成一个文件存入至 HDFS 块中, 合并后的文件若要修改, 必须重新创建存档文件, 一定程度上影响了文件的存储速度。而本文算法考虑到用户的访问模式, 虽然在存储之前对小文件做了一个预处理, 但相对于大数据集来说, 对文件的整体写入速度影响不大, 依然有着非常好的存储效率。

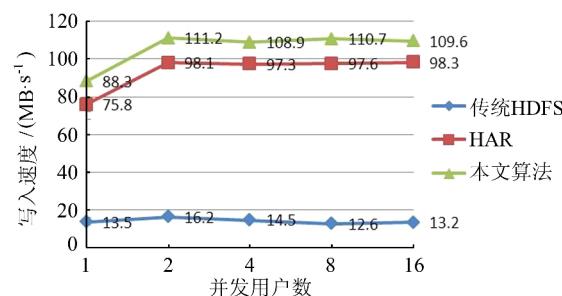


图 5 写入速度对比

Fig. 5 Comparison of Writing Time

2.3 文件读取测试

根据实验平台所提供的网络应用, 通过 HDFS Client 模拟用户的服务请求, 读取 3.96 GB 包含 100 000 个小文件的测试数据集, 统计各自的读取速度, 实验结果如图 6 所示。

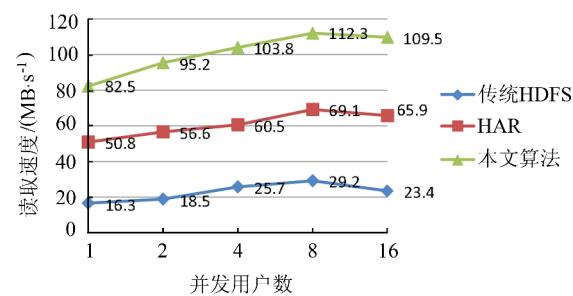


图 6 读取速度对比

Fig. 6 Comparison of Reading Time

从图 6 可以看出, 过多的小文件导致 HDFS 元数据检索复杂, 系统内部通信频繁, 大部分时间

都花在了系统开销上,增加了读取时间。HAR 归档技术虽然存储效率很高,但在文件的读取上并没有考虑到文件之间的相关性,降低了文件的读取命中率,导致 HDFS Client 与 NameNode 和 DataNode 频繁通信,极大地影响了数据的读取速度。而本文提到的合并算法是在分析用户访问规律和文件自身属性的基础上进行的,把相关联的文件尽可能地合并到了一起,有效地提高了系统的读取速率。

3 结语

本文针对在网络应用中 HDFS 处理海量时空小文件存储读取效率低这一问题,根据用户访问的时空局部性和数据自身的时空特性,提出了将用户的访问规律与文件自身属性相结合的小文件合并算法。该算法把用户访问看成是对数据文件的请求序列,根据序列中数据时空属性的变化情况提取并构建与之对应的特征序列,通过模板匹配找到用户在不同访问模式下的特征模板,最后分析属于同一模板内的数据特点,发现隐藏在用户访问行为下的文件之间的访问相关性,合并相关文件。实验结果表明,在存储同样大小和数量的小文件情况下,与传统的 HDFS 和 HAR 归档技术相比,本文所提出的方法有效地降低了 NameNode 的内存消耗,并明显提高了系统的存储读取效率。

下一步的工作将对已有的方法做出两点改进:①在数据请求序列的特征提取上,不仅仅局限于只考虑特征属性的变化与否这种逻辑状态,同时构建出多个特征序列来进行分析;②对特征空间里的特征序列使用动态模板匹配技术,在模板的命中率和通用性这两者间找到一个适中点,更为合理地合并相关文件。

参 考 文 献

- [1] Liu Yuefeng. Synthetic Dissertation of Geographic Information Service[J]. *Geomatics World*, 2004, 2 (6):26-29(刘岳峰. 地理信息服务概述[J]. 地理信息世界,2004,2(6):26-29)
- [2] Yu Si, Gui Xiaolin, Huang Ruwei, et al. Improving the Storage Efficiency of Small Files in Cloud Storage [J]. *Journal of Xi'an Jiaotong University*, 2011, 45(6):59-63(余思,桂小林,黄汝维,等. 一种提高云存储中小文件存储效率的方案[J]. 西安交通大学学报,2011, 45(6):59-63)
- [3] Hadoop Archive. Hadoop Archives Guide [EB/OL]. http://hadoop.apache.org/common/docs/current/adoop_archives.html, 2011
- [4] Sequence File. Sequence File Wiki [EB/OL]. <http://wiki.apache.org/hadoop/SequenceFiles>, 2011
- [5] Venner J. Pro Hadoop[M]. Berkeley: Apress 2009
- [6] Jiang L, Li B, Song M L. The Optimization of HDFS Based on Small Files[C]. The 3rd IEEE International Conference on Data of Conference, Beijing, China, 2010
- [7] Dong B, Zheng Q H, Tian F. An Optimized Approach for Storing and Accessing Small Files on Cloud Storage[J]. *Journal of Network and Computer Applications*, 2012, 35(6):1 847-1 862
- [8] Liu X H, Han J Z, Zhong Y Q. Implementing Web-GIS on Hadoop: A Case Study of Improving Small File I/O Performance on HDFS[C]. IEEE International Conference on Data of Conference, New Orleans, LA, United States, 2009
- [9] Dong B, Qiu J, Zheng Q H. A Novel Approach to Improving the Efficiency of Storing and Accessing Small Files on Hadoop: A Case Study by PowerPoint Files[C]. IEEE 7th International Conference on Services Computing, SCC2010, Miami, FL, United States, 2010
- [10] Arun K, Mark S, Zhang L. Analysis and Characterization of Large-scale Web Server Access Patterns and Performance[J]. *World Wide Web*, 1999, 2(1/2): 85-100
- [11] Ban Z J, Gu Z M, Jin Y. An Online PPM Prediction Model for Web Prefetching[C]. The 9th Annual ACM International Workshop on Web Information and Data Management, Lisboa, Portugal, 2007
- [12] Wu Lihui. Research on Customized Web Crawling [D]. Beijing: Institute of Computing Technology, Chinese Academy of Science, 2005(吴丽辉. 个性化的 Web 信息采集技术研究[D]. 北京:中国科学院计算技术研究所,2005)
- [13] Xu Baowen, Zhang Weifeng. Applying Data Mining to Web Pre-Fetching[J]. *Chinese Journal of Computers*, 2001, 24(4):430-436(徐宝文,张卫丰. 数据挖掘技术在 Web 预取中的应用研究[J]. 计算机学报,2001,24(4):430-436)
- [14] Zhang X D. A Popularity-based Prediction Model for Web Prefetching[J]. *Computer*, 2003, 36(3):63-70

On the Store Strategy of Small Spatio-Temporal Data Files in Cloud Environment

XIONG Lian¹ XU Zhengquan¹ WANG Tao¹ GU Xin¹

1 State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China

Abstract: Internet applications create massive small spatio-temporal data files in cloud environments. Therefore a method aiming to raise the processing efficiency of small files in HDFS; a data scheme combining user access and data features, is proposed. This scheme regards a user access stream as file request sequence, and constructs a characteristic sequence by spatio-temporal attribute extraction. A characteristic template of different user access patterns is formed when analyzing the characteristic sequence by template matching. Then merger-related files are analyzed. Experimental results show that our scheme improves the storage efficiency for small files, and also decreases network application response times.

Key words: cloud storage; small file; spatio-temporal data; feature extraction; template matching

First author: XIONG Lian, PhD candidate, specializes in Web data mining and distributed storage. E-mail: xionglian302@163.com

Foundation support: The National 973 Program of China, No. 2011CB302306; the National Natural Science Foundation of China, No. 41271398.

(上接第 1240 页)

Intelligent Monitoring for Dense Crowd Using a Fisheye Camera

HU Xuemin¹ ZHENG Hong² GUO Lin¹

1 School of Physics and Electronics, Hubei University, Wuhan 430062, China

² School of Electronic Information, Wuhan University, Wuhan 430072, China

Abstract: Crowd monitoring in dense crowd scene has become an important and difficult topic in the field of automatic surveillance system. A dense crowd monitoring approach based on perspective weight model for fisheye images is proposed in this paper. Experimental results show that the proposed approach is effective and feasible for dense crowd monitoring in indoor scenes.

Key words: dense crowd; perspective weight model; density estimation; crowd counting

First author: HU Xuemin, PhD, specializes in image processing, machine vision, intelligent computing and intelligent system. E-mail: huxuemin2003@163.com

Foundation support: Research Fund of Hubei University, No. 030-098343; Science and Technology Research Program for the Young Talents of Hubei Education Committee of China, No. Q20131004.