

基于 Map/Reduce 集群上的模式空间划分的序列模式挖掘

刘 骞, 陈 明

(中国石油大学 计算机科学与技术系, 北京 102249)

摘 要: 通过模式空间划分将基于 Map/Reduce 处理数据集与候选序列模式集的多对多的对应关系的问题转化为处理数据集与以频繁 1-序列为基的各子模式空间的多对多的对应关系问题, 大大缩小了中间结果键值对集合的规模, 避免了由于组合爆炸导致的单一 Map 节点的瓶颈问题. 通过三轮的 Map/Reduce 任务, 实现了模式空间和过滤规则的建立, 并在此基础上实现了各子模式空间上独立地进行序列模式的挖掘. 通过充分利用整个模式空间的全局特征及各子模式空间的个性特征, 设计了优化的非递归挖掘算法, 减少了前缀投影库构造次数及对构造的投影库的扫描次数, 从而提高了挖掘阶段的效率.

关键词: Map/Reduce; 模式空间划分; 序列模式挖掘; 云计算

中图分类号: TP39

文献标识码: A

文章编号: 1000-7180(2012)09-0149-03

Sequential Pattern Mining Based on Pattern Space Division in Map/Reduce Cluster

LIU Qian, CHEN Ming

(Department of Computer Science and Technology, China University of Petroleum, Beijing 102249)

Abstract: By means of pattern space division and based on Map/Reduce, the problem of processing the many-to-many corresponding relationship between the data set and the patterns set is converted to the problem of processing the many-to-many corresponding relationship between the data subsets and the pattern subspaces associated with the length-1 sequential patterns. Thus, the size of the intermediate key/value pairs set is reduced so dramatically that the problem of single Map node bottleneck which results from combinatorial explosion of candidate pattern space is avoided. Over three rounds of Map/Reduce tasks, the pattern space is constructed and divided, the filtering rules is set up and used, father more, the sequential pattern mining is realized in each pattern subspace independently. By making the best of both the universal trait of the whole pattern space and the individuality of pattern subspace, the optimized non-recursive algorithm is designed and implemented to improve the efficiency of mining phase by avoid unnecessary constructing of prefix projected databases and scanning of the constructed prefix projected databases.

Key words: Map/Reduce; pattern space division; sequential pattern mining; cloud computing

1 引言

Map/Reduce 取得巨大成功的原因在于它在对于大规模数据的分析型处理方面作了优化. 包括其存储子系统对读操作特别是顺序读操作进行的优化, 以及为使其能够可靠、高效、并行地加载数据而对写操作的优化^[1]. 在处理子系统上, 它一方面相对

于传统的数据分析工具增加了灵活性及通用性, 另一方面通过限制编程模型将复杂的分布并行计算的细节隐藏到运行时中从而大大降低了用户接口的复杂性, 使用户可以专注于对数据的处理本身^[2].

2 Map/Reduce 的特点

Map/Reduce 处理数据方式为把数据分析问题

转化为将规模为 M 的初始键值对集合处理成规模为 N 的结果键值对集合的问题. 在数据分析任务中 M 的规模往往非常大, Map/Reduce 通过运行许多 Map 任务来实现对 M 的分治. 每个 Map 任务中可能生成的中间结果键值对集合的规模仍为 $O(N)$. 必须保证 N 不至过大以保证 Map 任务生成的中间结果键值对数在合理范围从而保证对中间结果键值对进行排序以及 shuffle 完成后 Reduce 任务中对规模同样为 $O(N)$ 的结果键值对进行排序的开销不致过大.

由于 Map/Reduce 集群的计算子系统与存储子系统之间具有较高的数据传输效率, 瓶颈在于网络子系统, 故缩小 Map 阶段生成的中间结果文件体积对减少数据 shuffle 阶段的时间及整个任务的完成时间非常关键. 所以, 在 N 规模合理的前提下, 为了尽量缩短任务运行时间及保证任务能够正常完成, 应尽量缩小中间文件的体积^[2].

3 序列模式挖掘的特点

通过 Map/Reduce 实现序列模式挖掘的关键在于通过对问题合理的建模来控制 N 的规模. 针对序列模式挖掘的特点, 通过以下建模方式控制 N 的规模. (1) 构造模式空间, 并将其划分为多个不重不漏地覆盖该模式空间的子空间; (2) 通过编码将每个模式映射为特定子模式空间中的一个点, 从而将每个初始键值对中的许多模式映射到多个子模式空间中去; (3) 将每个子模式空间的基视为中间结果键值对的键, 而将各初始键值对子集中蕴含的模式在各子模式空间上的投影视为中间结果键值对的值. 图 1 示意了通过模式空间划分挖掘序列模式的方式.

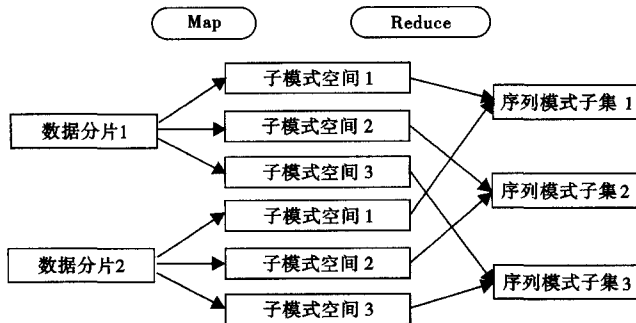


图 1 序列模式信息转移汇聚过程图

4 Map/Reduce 上的序列模式挖掘

序列数据存储于集群的分布式文件系统中, 序列中的各个项已进行整数编码. 通过三轮 Map/Re-

duce 任务分别实现 1-序列集的提取、2-序列集的提取以及模式空间的构建和模式的挖掘. 任务中的输入数据、中间结果数据、输出数据都以 Google data buffers protocol 格式^[3]存储及处理.

4.1 提取 1-序列集, 生成编码方案

通过一轮 Map/Reduce 任务提取 1-序列集, 并按其相应支持度计数降序排序. Map 任务处理各数据分片时, 若以各个项为中间结果键值对的键, 对应支持度计数为值, 将导致 Map 函数序列的 emit, 而且不利于压缩中间结果数据^[4]. 故采用以项及相应支持度计数为元素的线性表作为中间结果键值对的值, “1”作为中间结果键值对的键. 对 Map 任务对应的数据分片中的项的计数基于将项的整数编码作为索引的数组结构实现. 使得每个 Map 任务生成以“1”为键, 按项的整数编码为索引的线性表作为值的唯一的键值对. 按该方式组织中间结果键值对时, 中间文件的大小与 Map 任务的个数成正比, 通过使每个 Map 任务的数据分片尽量大便可缩小生成的中间文件的总的大小. 同时该方式可避免了对中间结果键值对的排序开销^[4]. Reduce 任务只有一个, 最终生成一个以“1”为键, 按序列项的支持度计数降序排列的线性表作为值的唯一键值对. 由于中间结果键值对的键唯一, Reduce 任务对各 Map 任务的键值对的合并无需等所有 Map 任务完成即可开始. 当 Reduce 任务合并了所有 Map 任务的键值对后, 得到一个线性表, 根据用户提供的支持度阈值 support threshold, 对该线性表中支持度计数不小于 support 的各项根据支持度计数降序排序, 生成一个线性表. 该线性表中项的顺序作为对序列、模式的编码方案及构造并划分模式空间的依据, 称作 1-序列列表, 记作 List.

4.2 提取 2-序列集, 生成过滤规则

通过第二轮 Map/Reduce 任务提取 2-序列集, 并将其以 $n \times 2n$ 二维矩阵的方式进行存储. 同样, Map 任务的数据分片应尽量大, 并使得每个 Map 任务只生成一个以“1”为键, 二维矩阵作为值的中间结果键值对. 二维矩阵的行列皆为上一轮 Map/Reduce 中生成的线性表中的项, 各元素为行项、列项构成的候选二项的支持度计数. Reduce 任务也只有一个, 最终生成一个以“1”为键, 表示各候选 2-序列是否为 2-序列的以 0、1 为元素的 $n \times 2n$ 的二维矩阵. 其中第 i 列及第 $n+i$ 列分别表示第 List 中的第 i 项是否与各行号对应项作为不同元素及同一元素中项以后缀形式构成 2-序列. 该二维

矩阵作为模式空间划分阶段进一步压缩中间结果键值对及在子空间上挖掘序列模式阶段扫描投影库时是否对当前项进行计数以及记录当前项位置的依据. 该二维矩阵作为过滤规则, 称作 2-序列矩阵, 记作 Matrix.

4.3 构造并划分模式空间, 进行序列模式挖掘

4.3.1 Map 阶段: 构造及划分模式空间

将前面生成的 List 及 Matrix 作为参数传给各 Map 任务. 根据 List 对序列、模式进行编码并在此基础上构造、划分序列模式空间. 将序列中的序列模式信息映射到各个子空间中去, 并以 Matrix 作为过滤规则对候选模式进行剪枝, 提高划分模式空间的效率, 降低生成的中间文件的空间占用.

Map 任务由以下三个步骤完成: (1) 通过将相应输入分片中的各序列根据 List 剔除不在 List 中的项并将各元素中的项按 List 中项的顺序编码为一个新的序列 CSList. (2) 将 subCSList_i 记为 CS-List 的以第 i 项 t 为前缀且删除不能与 t 构成 2-序列的项后得到的子序列. 通过一次扫描 CSList 提取出各 subCSList_i 并以 t 作为索引将 subCSList_i 添加到 List 的第 t 项对应的线性表中. 当内存不足时将各线性表的内容作为一个分片附加到相应中间结果文件^[4]. (3) 当 Map 任务对应数据分片处理完成后, 通过一次 emit 多个以 1-序列为键, 各键对应的文件作为值构成中间结果键值对. 同样, 通过使每个中间文件对应一个键值对, 可避免中间及结果键值对的排序开销.

4.3.2 Reduce 阶段: 各子空间上的序列模式挖掘

算法思想为通过一个队列 Queue 以广度优先方式实现对序列哈希树的逻辑上的遍历. 即并不真正构造物理上的完整的哈希树结构, 而是把根据 Matrix 推断可能继续增长的当前长度的所有前缀序列的相关信息加入 Queue 队尾. 通过非递归方式并及时释放处理完成的序列占用的空间来提高效率减小开销. 算法伪码如下:

```
while(Queue.length() > 0){
```

1. Queue 的队首元素出队, 并将对应的序列标记为 sequence, 投影数据库标记为 projectedDatabase;

2. 对序列 sequence 中各项对应的 Matrix 的各行进行逻辑‘与’操作后与 List 进行逻辑‘与’操作, 根据结果构造 nodeList, nodeList 的每个元素结构为 {bool tag; int support; File projectedDb}, 结果中的非零元素对应 nodeList 的元素 tag 标记为 true;

3. 若 nodeList 中所有元素的 tag 皆为 false, 则 continue; // 无需扫描 projectedDatabase;

4. 扫描 projectedDatabase, 根据各数据序列中将 nodeList 的 tag 为 true 的元素的 support 增加并将各项所在的数据序列的编号及在序列中的位置加入 projectedDb; /* 对 Matrix 支持继续增长的序列计数并同时构造投影库, 使得对任何投影库的扫描最多为一次 */

5. 将 nodeList 中 support 不小于 min_support 的元素对应的项作为后缀附加到 sequence 构成 sequence' 与相应 projectedDb 构成 Queue 的元素加入 Queue 队尾;

6. 将 nodeList 中 support 小于 min_support 的元素删除, 释放存储空间.

```
}
```

至此, 完成了各 1-序列对应的子模式空间上的序列模式的挖掘. 由于各子空间的独立性, 各子空间上挖掘出的模式不重不漏的覆盖了整个模式空间. 故各 Reduce 任务输出的并集即为整个模式空间上的序列模式.

5 结束语

本文通过采取模式空间划分的策略, 在 Map/Reduce 集群上实现了对像序列模式挖掘这样的复杂的数据分析任务的分治, 解决了由于模式集的组合爆炸导致的单一节点瓶颈问题. 通过建立过滤规则, 充分利用了 2-序列矩阵数据结构的高效性. 通过并行度较高的 Map 阶段开销的较小增加, 大大缩小了中间文件的规模及 shuffle 阶段的时间消耗. 以过滤规则为指导, 避免了不必要的投影数据库扫描, 优化的非递归算法减小了系统开销, 显著地提高子模式空间进行序列模式挖掘的效率.

参考文献:

- [1] Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung. The Google file system[C]// Proceedings of the 19th Symposium on Operating Systems Principles. Lake George, New York: IEEE, 2003: 29-43.
- [2] Dean Jeffrey, Ghemawat Sanjay. Map/reduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.
- [3] Google Inc. Protocol Buffers: Google's data interchange format[EB/OL]. (2010) [2012-02-05]. <http://code.google.com/p/protobuf/> Accessed 26.01.10
- [4] McCreddie R. Map/Reduce indexing strategies: Studying scalability and efficiency[J]. Information processing and Management (2011), 2010.

(下转第 156 页)

参考文献:

- [1] Cafarella, M. J.; Halevy, A.; Wang, D. Z.; Wu, E.; and Zhang, Y. 2008. WebTables: Exploring the Power of Tables on the Web[C] // Proceedings of VLDB-08. Auckland, New Zealand: ACM, 2008: 538-549.
- [2] Crestan E, Pantel P. Web-scale Knowledge extraction from semi-structured tables[C] // Proceedings of WWW-2010. Raleigh, USA: ACM, 2010: 1081-1082.
- [3] 廖涛, 刘宗田, 孙荣. 表格定位技术的研究与实现[J]. 计算机科学, 2009, 36(9): 227-230.
- [4] Liu Bing, Web Data Mining. 俞勇, 薛贵荣, 韩定一, 译[M]. 北京: 清华大学出版社, 2009.
- [5] Chen H, Tsai S, Tsai J. Mining tables from large-scale HTML texts[C] // Proceedings of COLING-00. Saarbrücken, Germany: A CL, 2000: 166-172.
- [6] Robert G, Wilks Y. Information extraction: Beyond document retrieval[J]. Journal of Documentation, 1998, 54(1): 70-105.
- [7] Gatterbauer W, Bohunsky P, Herzog M, et al. Towards domain-independent information extraction from web tables[C] // Proceedings of WWW-07. Banff, Canada: ACM, 2007: 71-80.
- [8] 鲜学丰, 方魏, 赵朋朋, 等. 一种 Deep Web 数据源质量评估模型[J]. 微电子学与计算机, 2008, 25(10): 47-50.

作者简介:

李 贵 男, (1964—), 博士, 教授. 研究方向为现代软件工程理论与方法、Web 信息挖掘、集成与应用等职能信息处理与集成技术、Web 服务计算与 Web 智能算法.

(上接第 151 页)

作者简介:



刘 翥 男, (1985—), 博士生. 研究方向为分布式计算、计算智能、软件工程.



陈 明 男, (1949—), 教授, 博士生导师. 研究方向为分布式计算、计算智能、软件工程.