

面向大规模金融对账文件的近似比对模型及系统^①

尹祥龙¹, 王伟², 陈煜¹, 周继恩¹, 任明¹, 徐景良¹, 万鑫明¹

¹(中国银联股份有限公司, 上海 201201)

²(中国科学院软件研究所, 北京 100190)

摘要: 针对 TB 级的大规模金融对账文件的近似比对问题, 本文深入分析了金融对账文件的特点, 以提升比对速度作为研究目标, 提出了一种多层次的近似比对模型—UpCompare 模型. UpCompare 模型以多进程为扩展基础, 采用哈希索引建立映射表结合快速致胜策略为核心算法. 测试结果表明, 运用 UpCompare 模型, 我国银行卡清算系统的每日清算文件近似比对效率提升了 5 倍以上.

关键词: 海量文件; 金融对账文件; 近似比对; 哈希索引

Massive Financial Reconciliation File Approximate Comparison Model and System

YIN Xiang-Long¹, WANG Wei², CHEN Yu¹, ZHOU Ji-En¹, REN Ming¹, XU Jing-Liang¹, WAN Xin-Ming¹

¹(China UnionPay, Shanghai 201201, China)

²(Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Focus on TB Level massive financial reconciliations file approximate comparison problem, this paper researched a number of the financial reconciliations file features to enhance the mass data comparing speed in financial sector, and proposed a multi-level approximate comparison model - UpCompare Model. UpCompare Model is a kind of multi-process technology based on hash index table, using fast winning strategy as the core algorithm, effectively solving the massive financial reconciliations file approximate comparison problem. Experimental results show that, by using UpCompare Model, bank transaction settle system efficiency improved more than 5× in daily financial reconciliations file approximate comparison.

Key words: massive files; financial reconciliation file; approximate comparison; hash index

随着金融领域信息技术的发展, 金融对账文件的类型越来越多, 数据量越来越大. 以银行业为例, 大型商业银行的银行卡清算系统, 其每天需要处理 TB 级数据量的金融对账文件. 金融对账文件的特点有: 一是单个文件容量大, 可以达到 GB 级; 二是文件数量众多, 通常在数万、甚至数十万; 三是单个文件中以换行符分割, 每行对应一笔交易记录, 行与行之间无序、无关联性; 四是部分对账文件需要根据约定打包压缩.

商用文件比对软件可以提供基本的内容比对及可视化展现功能, 但无法满足大规模金融对账文件近似比对的效率要求. 运行在大型集群的廉价硬件设备上

的开源 Hadoop 框架, 提供了基于 MapReduce 编程模型实现大规模数据的并行处理^[1,2]. 但是, 在处理金融对账文件, 这些方法存在严重的通信开销; 并且, 由于文件数据量大、格式特殊, 常采用抽样选取部分文件, 排序后进行比对, 导致比对结论的可信度不高. 因此, 研究提升大规模金融对账文件的近似比对效率就格外重要且紧迫. 本文提出一种多层次的近似比对模型 UpCompare, 模型以多进程为扩展基础, 采用哈希索引^[3]建立映射表, 并结合快速致胜策略为核心算法, 将数据缓存在共享内存^[4]中进行比对. 测试结果表明, 运用该模型, 我国银行卡清算系统的每日清算

^① 收稿时间:2015-07-23;收到修改稿时间:2015-09-14

文件近似比对效率提升了 5 倍以上。

1 相关研究

现有的商用文件比对软件，如 Ultra Compare 和 Beyond Compare，主要用于有序文本文件的顺序比对、文件比对结果的直观展示、文件夹批量比对等。例如 Beyond Compare 可以遍历文件夹并批量比对所包含文件，还可以比较 Word、Excel 等 Office 文档内容。这些商用文件比对软件提供了基本的内容比对及可视化展现功能，但无法满足大规模金融对账文件的近似比对需求，主要存在以下问题：

- (1)大规模(文件数量在数万、甚至数十万)、大容量文件(文件大小可以达到 GB 级)的快速比对；
- (2)以行为单位的杂序文本文件比对；
- (3)对部分内容可容忍差异的兼容性比对(两个近似金融对账文件中，如果两笔交易的清算时间戳等非关键要素存在差异，可认为是正确匹配的交易)。

2 UpCompare模型

UpCompare 模型采用以文件为切片的多进程并行比对方式，提升大规模文件的批量比对效率。在此基础上，UpCompare 模型将单个金融对账文件的比对流程分为三个层次：文件属性比对层、文件内容快速比对层、文件内容精细比对层，结合快速致胜思想，通过流程控制调度上述三个层次完成单个文件的比对任务，即在执行两个相似文件的比对时，自顶而下，依次比

对，如果上一个层次发现差异即跳过后续层次的比对，从而节约了时间和系统的开销。如图 1 所示为 UpCompare 模型结构图。

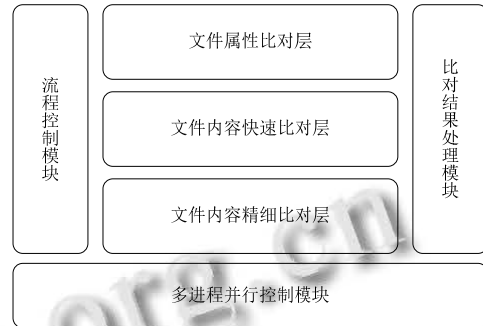


图 1 UpCompare 模型结构图

2.1 多进程并行控制模块

现有的商用文件比对软件，主要解决小数据量的文件比对问题，采用单进程的比对处理机制，难以应用于大规模的文件比对处理。UpCompare 模型采用了以文件为切片的多进程控制并行处理方式，由图 1 中的多进程并行控制模块具体实现。

多进程并行控制模块的功能是根据事先设定的进程数量启动多个进程，每个进程循环调用流程控制模块执行文件比对流程。该模块的最小任务切片单位是单个文件，任务分派方式采用预先均分结合时间片轮转动态调配机制，进程数根据运行系统资源情况推荐并由比对人员设定。多进程并行处理的原理图如图 2 所示，其关键处理步骤包括：

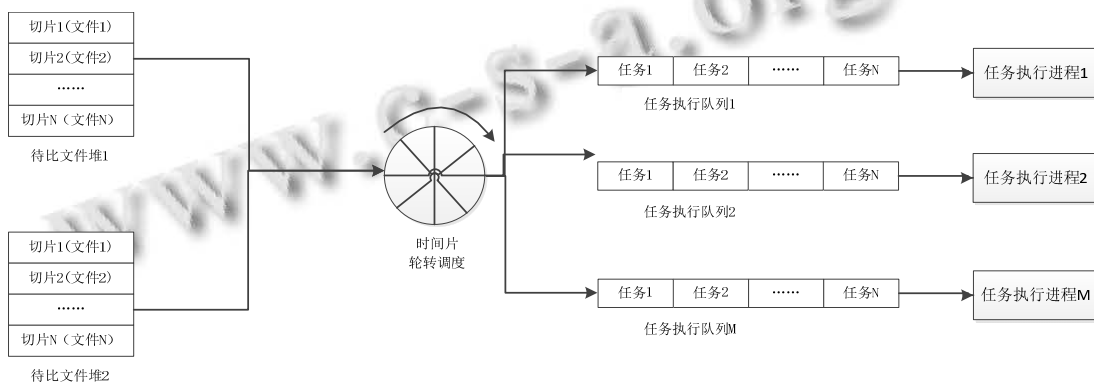


图 2 多进程并行处理原理图

(1) 设定进程数：计算对账文件容量大小，根据系统资源情况和经验数据估算合理的进程数量，并推荐给比对人员，由其最终决定进程数量；

(2) 分配任务：根据比对人员设定的进程数量，按照待比对文件容量大小给各进程初步均分文件比对任务，将任务列表插入到各进程的待比对文件队列中；

在任务分配过程中,如果在两个待比对文件堆中发现对应目录下无相同的文件夹或文件名,则直接调用比对结果处理模块输出到比对结果中,不再做为比对任务加入队列;

(3) 启动执行:启动各进程,调用比对流程控制模块,依次执行队列中的比对任务;

(4) 动态调配:多进程并行控制模块以时间片轮询各进程执行队列忙闲状态,并为比对文件任务动态调配执行进程.

2.2 流程控制模块

流程控制模块负责调度文件属性比对层、文件内容快速比对层、文件内容精细比对层、比对结果处理等模块完成单个文件的比对任务. UpCompare 模型采用快速致胜比对策略,其处理文件比对任务的核心思想首先执行开销最小的比对层次任务,比如发现文件名或文件大小不一致即退出,不再进入后续比对步骤.如图3所示为文件比对任务的处理流程图.

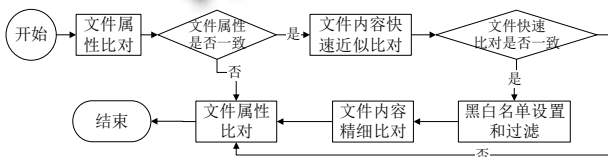


图3 文件比对任务处理流程图

(1) 文件属性比对层:采用快速制胜的策略,依次比较两个文件待比对文件的类型、大小等属性,如果发现某一项属性不一致,则立即返回比对结果,不再进入后续比对层处理;特别地,对于文件类型为压缩文件类型,需要先解压缩再比对.

(2) 文件内容快速比对层:假定待比对任务的两个文件分别为文件 A 和文件 B,该层次的比对方法首先遍历文件 A 的所有行,对文件 A 的行数据关键域建立索引数组,采用高效率的行压缩技术以及低耦合的散列技术在内存中建立文件 A 的索引表;然后,遍历文件 B 的各行,比较 B 中的各行数据的哈希值是否在文件 A 的索引表中存在,如果不存在,则表示该行是两个文件的不同处,将该行的相关信息输出到比对结果的记录日志记录.如果文件 A 与文件 B 相比是一致的,则使用相同的方法将文件 B 与文件 A 进行比对.

(3) 文件内容精细比对层:通过设置文件白名单、黑名单以及文件行的待比较关键域和比对策略,选择近似比对的程度,合理地过滤掉符合金融对账文件差

异的部分,实现比对过程的精细控制.

2.3 结果处理模块

结果处理模块负责将比对差异结果进行个性化输出,同时,支持比对过程的日志记录显示.

(1) 个性化输出:通过获取上面文件比对过程中输出的结果,根据个性化设置展示给比对人员,增强了结果展示的易读性、多样性.通常情况下,比对人员根据原比对文件目录的结构,筛选出不一致行的数据组织成不一致文件目录;同时,按照文件标准规范,拆分出文件和报表比对输出的所有不同处记录到比对报告中,并将不一致的文件名使用醒目的颜色标记.

(2) 日志记录显示:日志的功能是所有操作的过程信息输出到日志中,方便比对人员分析查找比对过程中的详细操作记录.通过汇总比对的文件列表详单,统计出比对耗时、差异率、差异的行数和字段数,并以图形的方式显示.

3 UpCompare模型在我国银行卡清算系统中的应用

UpCompare 模型已在我国银行卡清算系统中得到应用,针对系统特点,本模型在技术实现过程中有以下两个关键点.

3.1 文件快速比对层算法的实现

对文本内容信息与关键字的匹配已有多种匹配算法,如单模式匹配算法中的 BM 算法和 KMP 算法,多模式匹配算法中的 AC 算法、Wu-Manber 算法等^[5].本文针对金融对账文件独有的特点,提出的 UpCompare 模型文件快速对比层算法,可以很好的实现大规模金融对账文件的比对.

方法针对单个金融对账文件进行建模,抽象比对对象.考虑如下情况:比对对象文件 A 和文件 B,它们以行记录分割、行之间没有顺序、串值都唯一.例如,文件 A 是由数据 1、2、3、5 组成,文件 B 是由数据 1 组成,建立的模型如图 4 所示.在此基础上,单个文件比对流程如图 5 所示.

首先,读取文件 A,使用哈希(Hash)函数对 A 的每一行内容进行编码,得到数组:

$$A[i]=\{\text{索引值 } Inx_i^A, \text{冲突数 } C_i^A, \text{行号链表 } H_i^A\},$$

其中,索引值 Inx_i^A 是第 i 行内容的哈希索引值,是数组 $A[i]$ 的唯一主键,冲突数 C_i^A 默认值为 1,行号链表 H_i^A 为一个存放行号的链表,默认存放第 i 行的行号;当文

件 A 中存在内容相同的行时, 计算出的哈希索引值会重复冲突, 不能满足唯一主键的要求, 为了避免冲突, 数组不新增数组项, 而是在发生冲突的数组项中记录该索引值冲突行的数量, 以及在对应的行号链表中记录该冲突的所有行号。

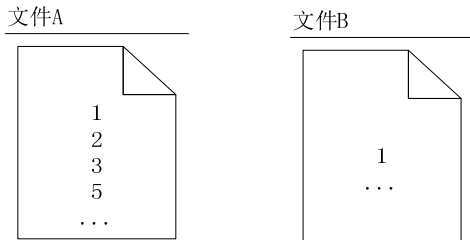


图 4 文件比对模型图

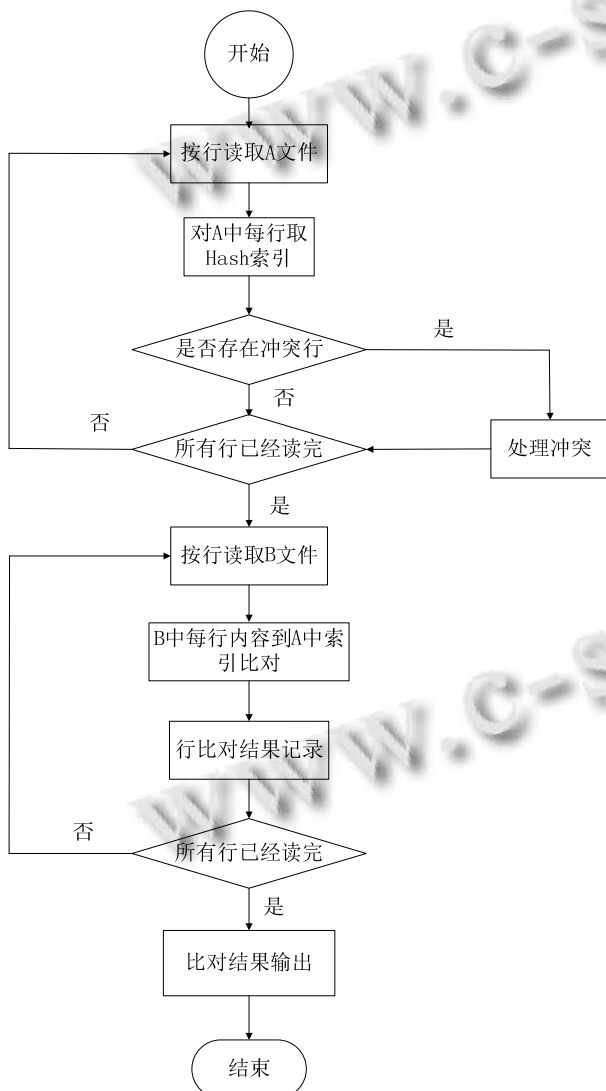


图 5 单个文件比对处理流程图

接着, 逐行读取文件 B, 使用哈希函数对 B 的 j

行内容进行编码, 得到哈希索引值 Inx_j^B , 然后在数组 $A[i]$ 中查找是否有值相等的索引值, 如果没有, 则说明文件 B 中这一行在 A 中不存在; 否则, 把 B 文件行 j 的内容依次与行号链表 H_i^A 对应的每一行内容比对, 如果发现相同内容的行, 假设 Inx_i^A 与 Inx_j^B 相等, 则将 A 文件的冲突数 C_i^A 减一并从行号链表 H_i^A 中移除文件 A 中的该行号, 如果冲突数 C_i^A 被减为 0, 则从数组中删除该索引值对应的数组项; 如果依次比对没发现相同内容的行, 说明文件 B 中这一行在 A 中不存在。

最后, 输出文件 A 与文件 B 的比对结果, 包含相同的行号列表和不相同的行号列表。

3.2 索引冲突的处理

由于 Hash 索引存储技术是一种压缩存储技术, 会遇到 Hash 索引值冲突的情况, 如何高效的处理索引冲突关系到整个比对算法的效率。UpCompare 使用常用的 Hash 算法, 可以有效地对相似度较大的金融对账流水文件进行 Hash 散列, 使其均匀分布在 Hash 表中, 并且对冲突节点使用链表方式存储, 在算法执行过程中可以有效的增加和删除冲突节点。

为了简单说明, 假设 Hash 函数采用 $f(x) = A[i] \% 4$, 那么可以得到以下比对过程, 输入文件 A、B 内容如下:

$$A = \{1, 2, 3, 5\} \quad (\text{其中“,”号表示换行符})$$

$$B = \{1, 2\}$$

比对过程中, 文件 A 的索引数组演进变化情况如下

$$A = \begin{bmatrix} 1 & 2 & 1, 4 \\ 2 & 1 & 2 \\ 3 & 1 & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 4 \\ 2 & 1 & 2 \\ 3 & 1 & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 4 \\ 3 & 1 & 3 \end{bmatrix}$$

比对结果: 文件 B 中的内容是文件 A 的子集, 并且 A 中第 1、2 行内容在 B 中存在, A 中第 3、4 行内容在 B 中不存在。

3.3 测试数据

针对近 5 万个、数据量共计达到 5TB 的金融对账文件, 使用 UpCompare 模型进行比对。测试环境采用 IBM P570 的 6 个逻辑服务器, 每个配置为 8 核 CPU、16GB 内存, 待比对文件部署在 NFS 并通过千兆网络访问。测试结果如表 1 所示, 与传统比对方法相比, UpCompare 的比对效率提升了 5 倍以上。

与原有比对技术相比: 1) UpCompare 模型利用快速致胜思想, 避免了文件近似比对过程中容易引起的额外时间、空间开销问题, 提高了比对效率; 2) 将数据

缓存在共享内存中进行比对,避免了排序、压缩、解压、拷贝和移动等文件 I/O 操作; 3)减少了各个环节的人工干预,提高了大规模金融对账文件近似比对的自动化程度.

表 1 效率对比结果

模型名称	时间(s)	网络 I/O(MB)
传统比对方法	169586	22956244
UpCompare 模型	28433	9348526

4 结语

本文对大规模金融对账文件的快速相似比对问题进行了研究,提出了一种采用哈希索引建立映射表结合快速致胜策略的 UpCompare 模型.测试数据表明 UpCompare 模型能大幅提高金融对账文件的比对效率.

参考文献

- 1 Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, 2008, 51(1): 189-195.
- 2 Lee KH, Lee YJ, Choi H, Chung YD, Moon B. Parallel data processing with MapReduce: A survey. *ACM SIGMOD Record*, 2011, 40(4): 11-20.
- 3 科曼. 潘金贵, 译. 算法导论. 机械工业出版社, 2006.
- 4 史蒂文斯, 拉弋. Unix 环境高级编程. 尤晋元, 张亚英, 戚正伟, 译. 北京: 人民邮电出版社, 2006.
- 5 何文华. 基于海量数据的多模式匹配算法研究. *计算机应用与软件*, 2012, 29(4): 275-277, 296.