一种基于Hash－B－树索引的海量用电数据快速检索算法

针对智能用电双向交互平台中海量电力用户用电信息的分析处理、存储及快速检索需求，基于Hadoop平台建立了海量用电信息的并行处理及云存储体系，并针对海量用电信息的特点，提出了一种基于Hash－B－树索引的海量用电数据快速检索算法，介绍了Hash－B－树索引查找算法及实现方式，通过建立索引缩小查找范围，在保证不浪费存储空间的前提下提高检索速度。实验证明，该算法能够从海量数据中高效、快速地检索到所需的用电信息。

骆凯波 蒲 箭 胡 君 赵 莉/重庆市电力公司

骆凯波/工程师

关键词/Keywords
云存储·
索引·
Hash－B－树·
Map－Reduce·

0 引言

近年来，坚强智能电网作为电力行业电网未来发展方向，引起了社会的广泛关注。在社会对于能源需求越来越大，智能电网建设规模不断扩大的条件下，智能电网的覆盖范围随之广阔，智能设备数量巨大、种类繁多，每个设备包含数量众多的数据采集点，导致用电数据激增，呈现海量趋势。海量用电数据存储、分析处理及快速检索成为迫切需要解决的问题。

开源分布式计算平台Hadoop因面向海量数据能够提供高性能存储、分布式并行处理与查询服务，具有高容错性和强扩展性而备受业界青睐，并在海量数据存储及分布式并行处理中得到广泛应用[1-2]，作为其核心的HDFS提供了一种高吞吐量、高容错性及支持TB乃至PB量级超大规模数据云存储体系[3-4]。然而，在解决了海量用电数据的并行化处理及存储后，如何从海量用电数据中快速、精确地检索到用电信息成为又一个急需解决的问题。

当前，现有云数据处理检索优化技术还不够成熟[5-7]，对于海量数据的检索，主要有B/B＋/B－树检索、哈希检索和分布式哈希表等。针对支持海量数据存储的Hadoop平台，通过建立索引加快数据查询的速度成为数据查询的热点[8-9]，相关研究人员提出了可以加快数据查询的几种数据索引结构。文献[10]设计了直接分区索引，数据查询的时间复杂度是O(1)。然而，如果存储的数据量很大，建立的索引占用的存储空间随之增大，用空间换时间来提高查询速度反而得不偿失，因此该方法不适合海量用电数据的查询。文献[11]提出HDFS的B＋树索引结构，B＋树将这些数据块位置信息全部保存，利用该B＋树搜索一个关键字对应数据记录的文件块位置。在Map－Reduce任务运行过程中提供数据定位信息。然而，该索引结构在新节点加入或删除时，必须进行文件块位置信息的更新，这将造成索引重建的开销。

针对电力用户用电数据海量的特点，利用Map－Reduce计算模型[12]实现海量用电数据的并行分析处理，基于Hash索引和B－树索引的优势，构建Hash－B－树索引，克服了直接分区索引占用的存储空间大的缺陷，而且在新的节点加入或删除时，不必进行索引重建，大大提高了效率。本文算法基于Hadoop平台进行试验，实现了用电数据的快速检索。算法的提出能够为智能用电领域海量数据快速检索技术的研究提

中国知网 https://www.cnki.net

供很好的参考。

1 基于Hash－B－树索引的用电数据快速检索算法

1.1 Hash－B－树索引的构建

电力用户的用电信息，经过分析及处理后存储在HDFS中，其存储的物理结构如图1所示。用电用户每天的用电信息经过分析、处理后，按用户号由小到大的顺序存储在1个或多个64 MB数据块中，几天的数据也可能存放在一个64 MB数据块中，为了减少B－树在创建及检索时的时间开销，提高构建及检索速度，每个数据块划分为小的分区Sector，一个分区对应树节点中的一个关键字，按照用户号由小到大的顺序有序存储固定数目的用户信息（即一个用户段）。

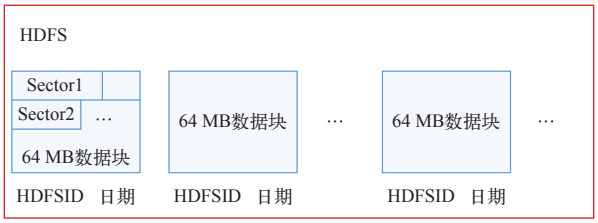

图1 海量用电数据存储的物理结构

针对电力用户海量用电数据的存储，为方便用户快速检索到某天的用电数据，建立索引结构以有效提高检索速度[13-14]。Metis在文献[15]中提出了Hash B＋树索引结构，该结构具有很高的读写性能，但是该索引结构在查询时需要查询到叶子节点才能查询到所需信息。本文提出适合电力用户用电信息查询需求的Hash－B－树索引结构，每次查询只要查询到所需信息所在的节点就截止，在减少平均检索长度的同时提高了检索效率。

Hash表存储一年内每天到用户该天的用电信息所在数据块HDFSID的索引，由于Hash表长度为366（考虑闰年情况）已经确定，设置简单的哈希函数实现Hash表的快速查找，时间复杂度为O(1)。

B－树存储用户的用电信息所在的HDFSID及所在分区的索引。使用B－树索引结构能够通过用户号快速检索到用电数据所在的HDFSID及所在分区，其时间复杂度为O(logn)。

Hash－B－树索引结构如图2所示。

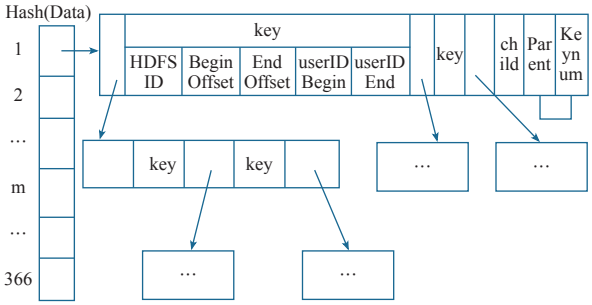

图2 Hash－B－树索引结构

Hash－B－树索引表伴随着用户海量用电数据的存储而建立。其建立过程如下。

1）对于某一天所有用户的用电信息，按照用户号由小到大的顺序排序后，选取固定数目的用户号作为一个区段存储在HDFS的某一数据块中，将当天的日期与对应B－树索引表的映射写入Hash表，将该区段的起始用户号userIDBegin、末尾用户号userIDEnd、所在数据块的HDFSID号、起始偏移BeginOffset及末偏移EndOffset作为一个关键字写入B－树索引表节点中，以此类推，将所有区段的用户信息都存储在HDFS的该64 MB数据块中，如果该数据块不能存下所有用户的信息，则将剩余用户信息存入另一数据块，直到该天的所有用户信息都存储完毕，并完成该日期所对应的Hash－B－树索引表的构建。

2）按照步骤1的操作方式将所有日期的用户信息存入HDFS中，同时建立完整的Hash－B－索引树。一块64 MB的数据块可能存储了所有用户一天的用电信息，也可能存储了所有用户几天的用电信息。

Hash－B－树的构建流程如图3所示。

1.2 用电数据快速检索流程

当用户提交一个查询请求时（包括用户号及日期），通过Hash表将查询的日期映射到该日期所对应的B－树中，根据用户提供的用户号查询B－树索引表，查询出该用户号所对应的用户信息所在数据块的HDFSID的起始偏移量及末偏移量。

中国知网 https://www.cnki.net

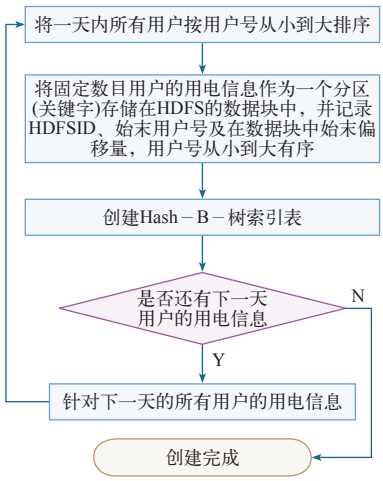

图3 电力用户海量用电数据快速检索流程

然后到所在的数据块中的指定位置范围内进行查询，用户号按照由小到大的顺序已排好序，利用二分查找算法实现用户某天的用电信息的检索。

电力用户海量用电数据的快速检索流程如图4所示。

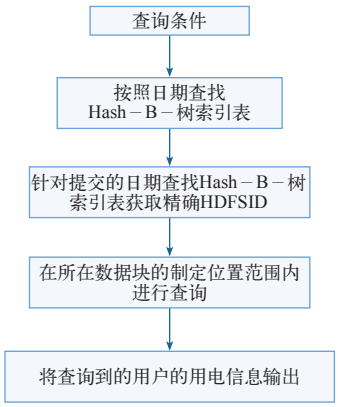

图4 电力用户海量用电数据快速检索流程

2 试验设计及结果分析

2.1 试验设计

试验环境：基于Apache Hadoop 1.0.4搭建了4个节点的Hadoop集群，1个主节点，3个从节点。HDFS副本数设为3，从节点最多可同时并行8个Map任务和8个Reduce任务，其他均为默认配置，各个节点的配置为：Intel Pentium 4 CPU 2.66 GHz双核，内存2 GB，硬盘250 GB。

试验数据来源于2012年重庆市所有电力用户的用电信息，仅采用了电力用户经常查询的电量计算数据、线损计算数据。每天采集240万用户的用电信息，针对这些海量用电信息经并行化分析处理后，存储于HDFS中，并建立对应的Hash－B－树索引表，Hash表存储该日期所对应的B－树索引的地址，B－树存放每天采集的240万用户的信息所在数据块的索引，为了既保证B－树索引的层数尽可能少以减少B－树搜索的高度，又保证每个节点的关键字数不要过多，提高检索速度，在一个区段包含的用户数分别为100、500、1 000、2 000和3 000的情况下，设置B－树索引的层数与阶数如表1所示。

表1 B－树索引设置表

分区包含用户数	100	500	1 000	2 000	3 000
B－树层数	5	5	4	4	3
B－树阶数	8	6	7	6	10

为验证所提出的算法的可行性和有效性，进行如下试验。

1）试验1。分别采用MySQL数据库数据分析处理及存储、Hadoop平台下并行分析处理及HDFS存储方法对重庆市某天的所有电力用户的用电信息进行电量计算、线损计算并存储，统计其耗时。

2）试验2。选取电力用户的用电信息数据量分别为10万、40万、240万、480万、1 200万、2 400万和4 800万条，设置MySQL数据库存储下电力用户用电信息检索、HDFS存储下一级Hash索引方法检索、HDFS存储下二级Hash－B－树索引检索（1 000个用户号作为一个区段，即一个关键字）进行海量用电信息的检索试验。

3）试验3。为了检验二级Hash－B－树索引检索方法中分区包含用户号的数目对查询效率产生的影响，分别选取用户号个数为100、500、1 000、2 000和3 000作为一个区段，在数据量为2 400万、4 800万下进行试验，反复进行10次试验取平均值以验证算法的稳定性并考察其检索时间。

2.2 试验结果分析

针对试验1设计的试验，其对比试验结果如表2所示。

中国知网 https://www.cnki.net

表2 海量用电数据分析处理及存储时间比较表

处理阶段		MySQL数据库存储方式下耗时/s	Hadoop平台下并行处理及HDFS存储耗时/s
线损计算	损耗计算	782	368
	线损达标数计算	117	36
电量计算	计算计量点电量	239	133
	计算高压用户电量	166	79
	计算台区电量	144	74
	计算低压用户电量	176	98

从表2可以看出，Hadoop平台下对海量用电数据进行并行分析处理及存储的时间消耗，明显低于传统的MySQL数据库存储处理时间消耗，效率提高了近2倍，说明Hadoop平台下基于Map－Reduce模型的并行数据处理适合本文的海量用电信息的分析处理。

针对试验2设计的试验，其对比试验结果如表3所示。

表3 数据检索时间比较

数据量万条	MySQL存储下的检索/s	HDFS存储下Hash索引检索/s	HDFS存储下Hash－B－树索引检索/s
10	0.17	0.51	0.57
40	1.52	0.62	0.68
240	3.02	0.98	0.89
480	9.8	1.13	1.01
1 200	27	1.47	1.16
2 400	62	2.03	1.56
4 800	138	5.97	3.05

从表3中可以看出，当数据量较小（10万条）时，MySQL数据库存储下的检索比HDFS存储下Hash索引检索、HDFS存储下Hash－B－树索引检索表现出更好的检索性能，Hash索引与Hash－B－树索引的检索性能相当，但随着数据量逐渐增大，MySQL存储下的检索时间则成倍增长，当数据量为4 800万时，查询时间增长到了138 s。随着数据量的增长（从240万到2 400万），Hash－B－树索引的检索性能较Hash索引的检索性能逐渐提高，当数据量为2 400万、4 800万时，Hash－B－树索引的检索效率较Hash索引的检索效率分别提高了23.1%、48.9%，说明提出的Hash－B－树索引检索算法可行，并且在海量数据检索时，其高检索性能的优势更加明显。

针对试验3设计的试验，其实验结果如表4所示。

表4 不同分区下Hash－B－树索引检索试验比较

分区包含用户数	2 400万数据量检索时间/s	4 800万数据量检索时间/s
100	2.7	4.6
500	1.87	3.81
1 000	1.56	3.05
2 000	1.79	3.42
3 000	1.89	3.68

通过表4可以看出，1 000个用户号作为一个分区，数据量为2 400万时，检索时间为1.56 s，数据量为4 800万时，检索时间为3.05 s，检索效率最高。100个用户号作为一个分区时，表现出较差的查询性能，数据量为2 400时，检索时间为2.7 s，数据量为4 800万时，检索时间为4.6 s。究其原因，当100个用户号作为一个区段时，使得B－索引树变得很大，检索时间都花费在了检索B－树索引上。当分区包含的用户数由1 000上涨到3 000时，检索性能开始小幅下降，究其原因是设置的分区包含的用户数不合理，使得在查询B－索引树和查询分区内用户信息时造成了时间上的浪费。以上数据说明，通过设置合理的分区所包含的用户号数，算法能达到很高的检索性能，说明算法可行，反复10次实验，时间消耗波动不大，说明提出的Hash－B－树索引检索算法具有较好的稳定性。

3 结束语

针对智能用电双向交互平台中电力用户海量用电信息的分析处理、存储及查询问题，进行了基于Hash－B－树索引的海量用电数据快速检索，主要工作如下。

1） 根据海量用电信息的特点及电力用户的用电信息存储需求，设计了基于Hadoop并行化处理及云存储平台的海量用电数据的组织方式、存放策略。

2）构建了Hash－B－树索引表，提出了基于Hash－B－树索引的海量数据快速检索算法。

中国知网 https://www.cnki.net

上述工作通过Hash检索，实现了用户查询日期到该日期所对应的B－树索引表的映射，通过B－树索引实现了用户用电信息所在数据块所在分区的定位，使得查询范围逐步缩小，在保证不浪费存储空间的前提下提高了检索速度。对比试验表明，基于Hash－B－树索引的海量数据快速检索算法是可行的、高效的。

参考文献

[1] 王章．海量历史数据库系统研究 [D]．武汉：华中科技大学，2012．

[2] 范帆．Hadoop中基于优先级的调度算法研究 [D]．上海：复旦大学，2012．

[3] 沈晴霓，杨雅辉，禹熹，等．一种面向多租户云存储平台的访问控制策略 [J]．小型微型计算机系统，2011，32 (11)：2223-2229．

[4] Keedong Yoo．Automatic document archiving for cloud storage using text mining-based topic identification technique [C]．2012 International Conference on Information and Computer Applications，2012．

[5] 史英杰，孟小峰．云数据管理系统中查询技术研究综述 [J]．计算机学报，2013，36 (2)：209-225．

[6] Pavlo A，Paulson E，Rasin A，et al．A comparison of approaches to large-scale data analysis [C]．Proceedings of the 2010 International Conference on Management of Data (SIGMOD2009)，Rhode Island，2009：165-178．

[7] Stonebraker M J，Abadi D，DeWitt D J，et al．MapReduce and parallel DBMS s：friends or foes [J]．Communications of the ACM，2010，53 (1)：64-71．

[8] 汤羽，王英杰，范爱华，等．基于HDFS开源架构与多级索引表的海量数据检索mDHT算法 [J]．计算机科学，2013，40 (2)：195-199．

[9] 长孙妮妮，张毅坤，华灯鑫，等．一种基于B＋树的混合索引结构 [J]．计算机工程，2012，38 (14)：35-40．

[10] An Mingyuan，Wang Yang，Wang Weiping，et al．Integrating DBMSs as a read-only exection layer into hadoop [C]．The 11th Internation Conference on Parallel and Distributed Computing，Applications and Technologies，IEEE，2010：17-26．

[11] Meng-Ju Hsieh，Chao-Rui Chang，Li-Yung Ho，et al．A scalable database management system for cloud computing [J]．Transaction on IEEE International Conference on Parallel Processing，2011：315-324．

[12] 李震，杜中军．云计算环境下的改进Map-Reduce模型 [J]．计算机工程，2012，38 (11)：27-29．

[13] An Mingyuan，Wang Yang，Wang Weiping．Using index in the mapreduce framework [C]．The 12th International Asia-Pacific Web Conference，2010：52-58．

[14] 魏东平，宗德君，孙华国．基于DTD的XML索引查询技术 [J]．计算机工程，2009，35 (18)：51-56．

[15] Kaashoek F，Morris R，Mao Y．Optimizing mapreduce for multicore architectures [R]．MIT Computer Science and Artificial Intelligence Laboratory：Technical Report MIT-CSAIL-TR-2010-020，2010．

（收稿日期：2015－09－09）EA

欢迎订阅 欢迎投稿 欢迎刊登广告

中国知网 https://www.cnki.net