云计算中Hadoop技术研究与应用综述

夏靖波 韦泽鲲 付 凯 陈 珍
(空军工程大学信息与导航学院 西安 710077)

摘 要 Hadoop作为当今云计算与大数据时代背景下最热门的技术之一,其相关生态圈与Spark技术的结合一同影响着学术发展和商业模式。首先介绍了Hadoop的起源和优势,阐明相关技术原理,如MapReduce,HDFS,YARN,Spark等;然后着重分析了当前Hadoop学术研究成果,从MapReduce算法的改进与创新、HDFS技术的优化与创新、二次开发与其它技术相结合、应用领域创新与实践4个方面进行总结,并简述了国内外应用现状。而Hadoop与Spark结合是未来的趋势,最后展望了Hadoop未来研究的发展方向和亟需解决的问题。

关键词 云计算,大数据,Hadoop,Spark,MapReduce
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.11.002

Review of Research and Application on Hadoop in Cloud Computing

XIA Jing-bo WEI Ze-kun FU Kai CHEN Zhen
(Institute of Information and Navigation, Air Force Engineering University, Xi'an 710077, China)

Abstract Hadoop is one of the most popular technologies in the area of cloud computing and big data nowadays, the combination of its relevant software ecosystem with Spark technology influences the academic development and business model. This paper firstly introduced the origin and advantages of Hadoop, and clarified the relevant technical principles, such as MapReduce, HDFS, YARN, Spark and so on. Then we focused on the analysis of the current Hadoop academic research achievements, and summarized four aspects: the improvement and innovation of the MapReduce algorithm, optimization and innovation of technology of HDFS, secondary development and other combination, innovation and practice of application field. And then the developing situation of domestic and foreign application was described. Hadoop with the Spark is the trend of the future. This paper finally discussed the development direction of the future research and some crucial problems which should be solved pressingly.

Keywords Cloud computing, Big data, Hadoop, Spark, MapReduce

1 概述

随着2008年9月4日《自然》(Nature)杂志刊登了一个名为"Big Data"的专辑[1],大数据时代正式宣告到来,伴随着大数据的研究,云计算、虚拟化技术伴随而生。

大数据(big data)也称海量数据,通常来说,凡是数据量超过一定大小,导致常规软件无法在一个可接受的时间范围内完成对其进行抓取、管理和处理工作的数据即可称为大数据[2]。

与大数据同时产生的概念是云计算,云计算[3]本质上是一种服务提供模型,通过该模型可以随时、随地、按需地通过网络访问共享资源池的资源,这个资源池内容包括计算资源、网络资源、存储资源等。

大数据与云计算的关系可以概述为:云计算的核心是业务模型,本质是数据处理技术,大数据即是其资产,是云计算的升级方向。

云计算技术的虚拟化、可扩展、按需服务以及资源池灵活调度等特性颠覆了传统网路技术模式和商业模式,海量非结构化的数据分析处理急需一种高效并行的编程模型。目前国内外的解决方案具有多种模式,而由Apache软件基金会研发的Hadoop作为大数据分析处理的主流技术迅速崛起。

Hadoop是一个分布式系统基础架构,也是一个可开发与运行处理大规模数据的软件平台。21世纪初,谷歌公司用廉价PC集群搭建了大型的MPP搜索引擎系统,成功解决了巨大数据量的搜索问题。并于2003年、2004年和2006年在学术会议SOSP和OSDI上发表了有关GFS(Google文件系统)[4]、Map/Reduce(编程环境)[5]和BigTable(数据模型)[6]的论文,这3篇奠基性的论文促成了Hadoop[7]的诞生。2004年,Cutting D和Cafarella M J根据Google Lab论文实施,取名Hadoop。

Hadoop的出现解决了大数据并行计算、存储、管理[8,9]等关键问题,用户可以在不了解分布式底层细节的情况下开

到稿日期:2015-10-13 返修日期:2016-03-09 本文受陕西省自然科学基金项目(2012JZ8005)资助。
夏靖波(1963-),男,博士后,教授,博士生导师,主要研究方向为通信网络管理、云计算、虚拟化技术,E-mail:50217711k@sina.com;韦泽鲲(1992-),男,硕士生,主要研究方向为云计算、Hadoop、态势感知;付 凯(1987-),男,博士生,主要研究方向为大数据、态势感知、复杂网络;陈 珍(1991-),女,硕士生,主要研究方向为云计算、网络态势感知。

发分布式程序，即功能的透明性，开发者只需要实现 map，reduce 等接口，而不需要关注底层系统级的问题，便可充分利用集群的威力高速运算和存储。Hadoop 集群具有高可靠、高扩展、高效和高容错的特性。同时，其开源的特性使其飞速发展与进化，并被广泛用于在线旅游、移动数据、电子商务、能源开采、节能、基础架构管理、图像处理、诈骗检测、IT 安全、医疗保健等领域。在海量数据处理上 Hadoop 得到了广泛的认可，但其在实时性、流处理方面仍存在很大不足。

其优点总结如下：

(1)高可靠性。Hadoop 采用按位存储与处理数据的技术，经过实际应用检验其具有较高可靠性；

(2) 高扩展性。Hadoop 通过在计算机集簇间分配数据进而完成计算任务，通过软件配置，这些集簇能够非常容易扩展到数以千计的节点中；

(3)高效性。Hadoop 能够动态地在节点之间移动数据，同时保证各个节点的动态负载平衡；

(4)高容错性。Hadoop 能够自动地将失败的任务重新分配，自动保存数据的多个副本，运维成本较低。

本文的目的是尝试对当前 Hadoop 相关的技术进行介绍与整理分类，并对典型的技术进行简单的剖析，合理推测 Hadoop 未来的研究与发展方向。

2 Hadoop 自身技术发展与进化

Hadoop 是开源的分布式编程计算框架，其核心由 Hadoop Common，HDFS，MapReduce[10] 3 部分组成。其中 Hadoop Common 项目为 Hadoop 整体架构提供基支撑性功能，主要包括文件系统、远程过程调用协议和串行化库。HDFS (Hadoop Distributed File System)[11]是一项适合构建于廉价集群上的分布式文件系统，能够存储海量非结构化的数据，具有高可靠性、高吞吐量、低成本的特点。MapReduce 是一种编程模型和软件框架，核心思想是 Map 和 Reduce，即任务的分解与结果的汇总。Hadoop 的基本运行环境包含 MapReduce 和 HDFS 两类组件。

2.1 MapReduce 模型

MapReduce[12]计算模式将一般的数据计算过程分为 Map 和 Reduce 两个阶段，将数据表述为键值对〈*key*，*value*〉的形式，通过多个高次函数的串接，将数据的计算转化为一些列函数的执行。处理过程如图 1 所示。

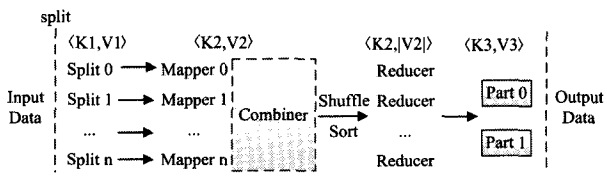

图 1 MapReduce 工作流程图

MapReduce 的过程可以用以下两个式子表述：

Map：〈*k*1，*v*1〉→*list*(〈*k*2，*v*2〉)

Reduce：〈*k*2，*list*(*v*2)〉→*list*(〈*k*3，*v*3〉)

用户首先读入数据，系统对数据进行分块，如图 1 中的 split 0 至 split n，这些数据块经处理后生成〈*k*，*v*〉键值对。由用户编写的 Map 函数读入一系列键值对〈*k*1，*v*1〉，经过分析处理产生一组中间的〈*k*2，*v*2〉键值对，再由 Partitioner 类将这些中间结果指定区分地写到输出文件中，随后 Combiner 类会对其进行合并，产生关于 *k*2 的键值对列表 *list*(*v*2)，这样 Map 部分的处理基本完成。

上一步的输出结果会作为输入进入 Reduce 函数中，经过组合(shuffle)、排序(sort)、聚集(reduce) 3 个阶段，对各个节点各个分区上的键值对进行分析处理，最后形成相对较小的键值对的集合 *list*(〈*k*3，*v*3〉)。

2.2 HDFS

HDFS[13,14]是运用于大规模廉价商用机子集群上的文件存储与传输系统，核心设计思想是“一次写入，多次读取”[15]访问方式，对文件进行分割后分别存放，将需要存储的大文件进行分割，形成 Block 数据块，从而完成大数据的存储。

针对 HDFS 并行读写的要求，HDFS 强化了处理的协同并发性，弱化了一致性的要求，增强了鲁棒性，同时加入写入锁的机制。当多个用户对多个节点进行读写操作时，不仅一个数据更改后与之关联的数据会发生更新，多用户在对同一文件进行写操作时也能正确写入。

一个 HDFS 集群包含一个 NameNode 与若干个 DataNode共同工作。NameNode 作为集群的主服务器，记录着集群总的工作状态以及 DataNode 节点的情况，是集群的中央指挥部。DataNode 则承担执行任务的角色，存储着 NameNode 指定存储的数据。

HDFS[16]读写交互数据时，以读为例，客户端先调用打开的方法向文件系统发出请求，文件系统通过远程协议，调用 NameNode 节点上的配置信息来获取在 DataNode 中文件的位置，返回地址到客户端，客户端依次读取多个 DataNode 上的数据文件，读取完毕后关闭文件系统数据读取的数据流，其流程如图 2 所示。

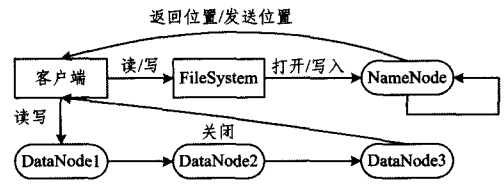

图 2 HDFS 读写流程图

2.3 YARN

YARN[8,17]是 Hadoop2.0 提出的新型资源管理系统，是一种适用于并行处理大规模数据的开源系统架构。它弥补了 Hadoop1.0 中存在的单点故障缺陷，设计思想是将 MapReduce 架构中 JobTracker 的两个主要功能——资源管理和任务调度/监控功能分离成为独立进程，即 ResourceManager 和 ApplicationMaster，更好地为并行编程模型提供支持。

YARN 主要由 ResourceManager，NodeManager，ApplicationMaster，Container 4 个组件构成。ResourceManager (RM)作为全局的资源管理器，负责整个集群系统资源管理和分配，核心是资源调度器 ResourceScheduler(RS)；NodeManager(NM)负责各个节点资源与任务的管理；ApplicationMaster(AM)作为应用程序的主控节点，负责监控和跟踪应用程序的执行状态；Container 则是每个节点负责封装动态资源(内存、硬盘、CPU、带宽等)的分配单元。YARN 中组件架构及流程如图 3 所示。

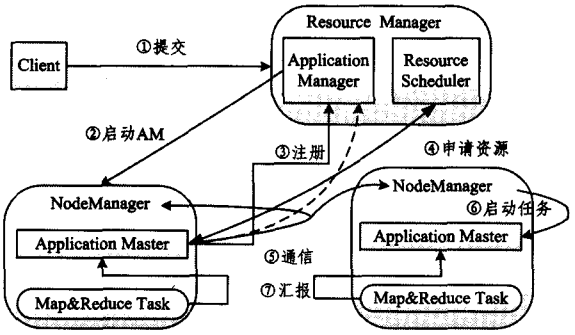

图 3 YARN 架构和工作流程

2.4 Hadoop 生态系统

从 2006 年 2 月 Apache 基金会的 Hadoop 项目成立，到 2008 年其成为 Apache 顶级项目，再到 2011 年 Hadoop1.0.0 版本发布，最后到 2012 年 Hadoop 2.0 的发布，Hadoop 的生态系统[8,10,15]不断趋于完善。除了 Hadoop Common，HDFS，Map-Reduce 之外，Hadoop 的相关技术如下：

(1)HBsae(Hadoop Database)，是一个分布式的、面向列数据的开源数据库，适用于大规模非结构化数据的存储。

(2)HCatalog，是用于管理 Hadoop 系统产生的数据的表存储管理系统。

(3)Hive，是一个数据仓库工具，作用是将结构化的数据文件映射为数据库表，并且提供强的类 SQL 查询功能。

(4)Pig，是用于大数据分析的工具，特点为支持并行化处理，包括一个数据分析语言以及其运行环境。Pig 提供类 SQL 语言(Pig Latin)，通过 MapReduce 来处理大规模非结构化数据。

(5)Sqoop，在 Hadoop 系统与传统的数据库间进行数据交换的工具。

(6)Avro，一个基于二进制数据传输的中间件，能够将数据序列化，适合于本地或远程大批量数据交互。

(7)Chukwa，一个分布式数据收集和分析系统，用于监控大型分布式系统。

(8)Zookeeper，一个分布式应用程序协调服务器，对 Hadoop 集群的运维进行管理。

Hadoop 生态圈中的各个项目在云计算中承担着不同作用，以保证底层海量数据能够最大限度地为顶层应用发挥效能，具体架构如图 4 所示。

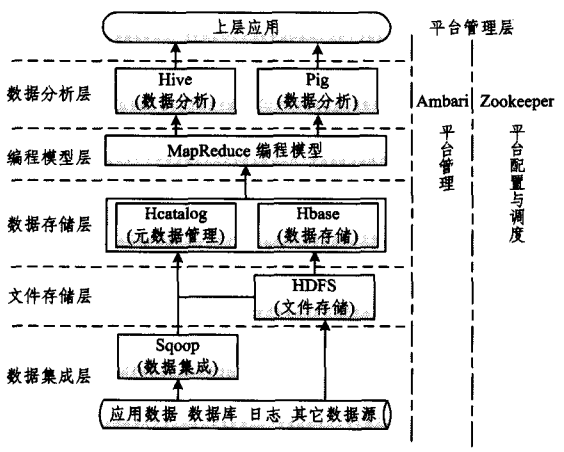

图 4 Hadoop 生态系统在云计算中的架构图

2.5 Spark

Spark[18]内存计算框架于 2009 年诞生于美国加州大学伯克利分校的 AMPLab 实验室，该框架设计开发的目标为 one stack to rule them all，即在一套软件堆栈内完成各种大数据分析任务。严格来说，Spark 并不是 Hadoop 生态圈的组成部分，而是一种与 Hadoop 相似的开源集群计算环境，但实际上公认是对 Hadoop 的补充。其主要特点为提供了一个集群的分布式内存抽象 RDD(Resilient Distributed Dataset)。RDD[18,19]是一个不可变的带分区记录集合，是 Spark 的编程模型。该模型提供两类 RDD 上的操作、转换和动作。转换用来定义一个新的 RDD，包括 map，flatMap，filter，union，sample，join，groupByKey，cogroup，ReduceByKey，cros，sortByKey，mapValues 等；动作是返回一个结果，包括 collect，reduce，count，save，lookupKey。

Spark 在执行任务时，整个流程在逻辑上形成有向无环执行图(DAG)。Spark 根据弹性分布式数据集 RDD 之间的不同的依赖关系切分成不同阶段(Stage)，每个阶段都有一系列函数执行的流水线，这是与 MapReduce 不同的地方。图 5 中 A 至 F 分别代表不同的 RDD，RDD 内的方框则代表分区。当数据从 HDFS 输入进 Spark 后，会形成 RDD A 和 RDD C，进入各自的阶段，如 RDD C 经过 Map 函数转化为 RDD D，而 RDD B 与 RDD E 执行 join 与 Shuffle 等操作转化为 F。最后 RDD F 通过 saveAsSequenceFile 输出并保存到 HDFS 中。

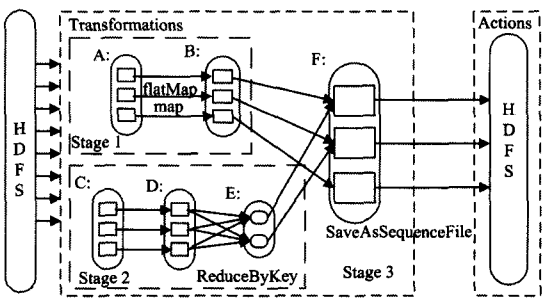

图 5 Spark 核心运行机制图

Spark 立足于内存计算，增强了多迭代批量处理能力，提高了大数据环境下处理数据的实时性，并保证了高容错性与高伸缩性。Spark 与 Hadoop 相比，在批量计算、迭代计算、类 SQL 查询等性能上得到了极大的提升，执行效率提升巨大。同时，该框架能够与 Hadoop 及其生态圈完美兼容，并拓展兼顾了数据仓库、流处理、图计算等多种计算范式。Spark 与 Hadoop 结合应用成为了大数据处理的主流方式。

3 Hadoop 发展与应用

3.1 Hadoop 学术研究与发展现状

通过对 Hadoop Summit，Spark Summit 等峰会最新动态的研究，以及大量文献的查阅，笔者发现现有的研究成果主要集中在以下 4 个主题。

3.1.1 MapReduce 算法的改进与创新

随着基于 MapReduce 编程模型的应用程序越来越多，MapReduce(Hadoop1.0 版本)性能的缺陷不断暴露出来。其主要不足有以下 3 点：

(1)MapReduce 是一种离线计算框架，实时性较差；

(2)MapReduce 不支持显示的迭代计算；

(3)MapReduce 主要针对松耦合型数据，在处理难以分割的紧耦合数据时效率较低。因此，优化 MapReduce 算法、提高系统性能就变得极为重要。

Yahoo 和美国加利福尼亚大学联合开发了一个改进型

MapReduce 模型——Map-Reduce-Merge[20]。在 Reduce 过程后增加了 Merge 过程，对松耦合型异构数据集的处理结果进行合并，改善了处理相关异构数据集上的缺陷。

伊利诺伊大学 Verma A[21]等人为改进 MapReduce 模型过程中键值对排序合并的不足，提出了 Barrierless MapReduce 并行编程模型。该模型修改了 Reduce 函数，使其能直接处理中间键值对，省去了排序与合并的步骤，但缺点是增加了用户编程负担。之后，Verma A 等人为了最小化集群作业所消耗的执行时间，提出了自动寻找最优调度方案的启发式算法 BalancedPool[22]。

为解决迭代型 MapReduce 改进模型效率不高的问题，挪威特罗姆瑟大学改进了 Map-Reduce-Merge 模型，其提出了 Oivos[23]模型，其能够自动管理多次执行 MapReduce 或 Map-Reduce-Merge 过程，提高迭代效率。

Bu 等人[24,25]提出 HaLoop 模型对迭代策略进行了改进，在 JobTracker 中，将每个 Job 单一的 Map-Reduce 对改为若干个 Map-Reduce 对，复用 Job，中间结果不用输出到硬盘，并增加了 Loop Control 机制对迭代循环进行控制，减少了 I/O，加快了处理速度，提高了效率。

Venkatesh[26]引入内存机制重构 Hadoop 内核，将内存缓存机制和缓存感知任务调度加入其中，提高了运行能力，其甚至优于 Spark 模型。

在大规模异构环境下，个别节点的性能会影响整个集群的性能。Chen 等人提出了 Samr（Self-Adaptive MapReduce）[27]自适应的任务调度策略，动态发现执行最慢的节点并备份其任务。华盛顿大学 Kwon 等人针对数据分布不均问题，提出了 SkewTune[28]动态负载均衡策略，其能够拆分较慢节点中的任务，并分配给快节点执行。国防科技大学 Li 等人[29]提出了 Shuffle 过程中 Hash 函数在不同 Key 数据量分布得不均衡是任务分布不均的问题，改进了任务调度策略。Ahmad 等人[30]针对异构集群下的 MapReduce 性能低的问题，提出了 Tarazu 模型，对异构集群下的 Map 与 Reduce 以及中间过程进行优化。Cherniak 等人[31]提出了多种优化并发 MapReduce 的方法，根据不同数据的查询进行负载优化，为并发 Hive 和 MaoReduce 任务分配适当的资源。

加州 Berkeley 大学的 Spark 为当下最为认可的大数据框架，其核心技术为弹性分布式数据集（RDD），该架构包括迭代计算、批处理计算、内存计算、流式计算（Spark Streaming），数据查询分析计算（Shark）以及图计算（GraphX）[32]，是 MapReduce模型的代替模型，前面一节已经分析过，这里不再赘述，但 Spark 的意义和研究价值不言而喻。

3.1.2 HDFS 技术的优化与创新

HDFS 是为方便海量大文件的存储和读写而设计产生的，因此在处理小文件时性能急剧下降。针对这点不足，Hadoop 自身提供了归档文件、序列文件、合并文件 3 种技术，但都因编程成本较高而未被广泛采用。

付松龄等人[33]针对读取随机均匀分布的小文件数据，设计并实现了扁平式数据存储的轻量级文件系统 FlatLFS，通过优化数据节点中数据块的存取来优化整体性能。但因为舍弃了传统层次式文件管理系统，因此其欠缺灵活性，只能适用于数据服务器的数据块管理。

张春明等人[34]提出了一种基于小文件合并的 HIFM（Hierarchy Index File Merging）算法，通过小文件之间的相关性分析以及结构感知，合并生成大文件，并生成分层索引。

熊安萍等人[35]针对 HDFS 海量小文件存储时元数据服务器上节点内存开销过大、合并文件中小文件访问效率不高的问题，提出一种改进的基于混合索引的小文件存储策略。应用分类器分类标记小文件，并在元数据服务器上建立H-B+树索引，在存储节点根据小文件大小建立不同的块内索引，达到提高小文件访问效率的目标。

HDFS 的负载均衡也是研究的重点之一，通过编码的方式能够进行 HDFS 负载优化。朱媛媛等人[36]提出了一种基于 GE 码的 HDFS 存储优化策略，其能够通过引入较少冗余效验码来提高可靠性和控制容错度。宋宝燕等人[37]基于范德蒙码提出了一种 HDFS 分散式副本动态存储策略——VanHDFS，通过动态效验码将容错度控制在理想范围内，并结合伽罗华有限域理论对其编译码操作进行优化，提高了 200%的数据存储可靠性，并节约了 30%的存储开销。

对于集群的物理条件与作业要求，卢美莲等[38]提出一种基于 CMM 的多阶段多目标 HDFS 负载均衡决策模型，该模型能以集群剩余 CPU、内存、磁盘作为先决条件，构建有向无环图，确立最优负载均衡。

席屏等人[39]设计并实现了多层一致性哈希的 HDFS 副本放置策略。该模型先经过一致性哈希算法对机架进行感知，获取数据副本对应的机架位置，并获得数据节点位置。该策略在数据存储、上传速率方面有较大提高，提高了自适应性。

对于提高 HDFS 读写的效率，Lu Kun[40]针对 HDFS 碎片化存储造成多源数据分析提取的低效问题，提出了改进型的 HDFS+模型，通过并发写入将不同节点的文件合成单一文件，提高了并行分析的效率。Nusrat Islam 等人[41]提出了一种高性能缓存策略 MEM-HDFS，其能将传输的数据块缓存在内存，极大地提高输入输出效率，读取执行时间降低约 40%。

3.1.3 二次开发与其它技术相结合

Twister[42]系统为美国印第安纳大学开发的 MapReduce 改进系统，是一种轻量级的、高效的、迭代式的计算框架。Twister 利用流处理技术，将需处理的数据全部驻留在内存之中，并采用了专用的消息分发机制，通过广播和分散型通信进行数据传输。其任务调度池（Task Pool）机制有效地避免了任务重复迭代创建，非常适合迭代式应用，但其缺点是没有 DFS 的支持。

传统关系型数据库 RDBMS 在大数据的冲击下，不断被 Hadoop 的 HDFS 所取代，对于 RDBMS 性能上的优势，HadoopDB[43]试图融合 MapReduce 和 RDBMS 技术。该系统分为两层，上层使用 Hadoop 进行任务的分解调度，下层用 RDBMS 进行数据的查询处理，但其性能仍落后于传统数据库。Yale 大学 Abouzeid 等人[44]试图通过列存储、持续装载等技术改进 HadoopDB 的性能。Gruska N 将关系数据库与MapReduce进行了结合，结果在数据加载、连接、查找、聚合等操作中均表现出较好的性能。

Qin Xiongpai[45]阐述了将 RDBMS 和 MapReduce 融合成一个统一的大数据分析系统的思想，并在 Hadoop 中对数据进行了 OLAP 分析和查询实验。

Phoenix[46]是美国斯坦福大学设计的基于多核/多线程平台上处理密集型作业任务的 MapReduce 计算框架。该系统实现了内存缓冲区的快速共享,避免了因数据拷贝而产生的开销。但是在数据量极大的情况下,Phoenix 系统存在崩溃的风险。

清华大学的 Tao Xu 等人[47]针对大数据结构环境,开发了一个交叉平台交互轮询的系统——Banian,该系统体系架构主要分为 3 层:存储层、调度执行层、应用查询层,而 HDFS 作为分布式存储海量数据的存储层。该系统对 PB 级数据查询的能力比 Hadoop 生态圈中的 Hive 性能优异得多。

Storm[48]是一个开源分布式的、容错的实时计算系统,其专注于实时流数据的处理。成都大学的靳永超等人[49]提出一种基于 Storm 和 Hadoop 的新型大数据解决方案,将 Storm 的实时流处理和 Hadoop 的批处理进行融合集成,提高了集群的处理性能和扩展性。当然,在商业领域这两者已结合投入应用。

3.1.4 应用领域创新与实践

SQL 与 Hadoop 的融合一直以来被视为 Hadoop 系统未来的杀手级应用。Rubao Lee 开发了 Ysmart 系统[50],该系统能高效率完成 SQL 到 MapReduce 的数据转换。

而 IBM 公司推出了 BigSQL,就是这样的一个大规模并行处理 SQL 的引擎,可将 SQL 访问数据库的模式直接转移到 Hadoop 集群上,可直接部署在物理的 HDFS 集群上。BigSQL 实现了原生方式的 Hadoop 数据访问,使得能像访问传统数据仓库那样去访问迁移到 Hadoop 平台的数据。

Zhao Huan 等[51]提出的云计算下中文旅游信息搜寻平台基于 Hadoop,Solr 和 Nutch 3 款强大的开源软件。该平台用 Nutch 去爬取网页上的中文信息并将它导入 Solr 中,通过相关类函数分类形成文字片段,再通过 Hadoop 分布式集群分析处理为用户提供旅游信息的搜寻。

同样是中文语义聚类算法,Chen Hong[52]将 IKAnalyzer(IK)与 ICTCLAS(IC)中文语义挖掘算法应用于 Hadoop 云平台下,并取得了很好的效果。

Hadoop 应用领域的创新主要集中于商业领域,且更加注重处理的效率和运维的成本。

3.2 商业的应用与发展

目前 Hadoop 较为流行的商业发行版本分别是 Cloudera CDH,Hortonworks 和 MapR[53] 3 家公司对 Apache 的 Hadoop 进行打包、改进发行的版本,他们对 Hadoop 做了相应的改变,确保所有生态圈的软件一起工作,并提供技术支持。

国外的 Google,Yahoo,IBM[54]等都是 Hadoop 的最大的支持者和应用者。IBM 的蓝云 Bluemix 是构建云框架的基础设施,其 IBM InfoSphere BigInsights 构建于 Apache Hadoop 之上,可提供大规模的静态数据进行分析;InfoSphere Streams 则可采用内存计算方式分析实时数据。A9.com 是 Amazon 基于 Hadoop 构建的商品搜索索引,提供了强大的搜索功能。Facebook 使用 Hadoop 存储内部日志与多维数据,基于 Hive 等进行日志分析和数据挖掘。Twitter 用 Hadoop 的 HDFS 存储微博数据、日志文件和中间数据,其采用 Cloudera's CDH2 系统来存储压缩数据。

国内应用和研究 Hadoop 的企业也不断增加,主要包括淘宝、百度、腾讯、网易等互联网公司以及中国移动、华为、农业银行等传统企业[55]。阿里巴巴是国内最先使用 Hadoop 的公司之一,淘宝使用 MapReuce Java Jobs,Streaming Jobs,Hive Jobs 构建其独特的云梯系统,连接底层数据仓库,为数据魔方、淘数据、推荐引擎、量子统计等应用提供支撑。当然,淘宝已换成 Spark 架构以支持更为迅速的实时性处理分析。百度也是 Hadoop 的最大使用者之一,截至 2012 年底,其拥有集群规模达 10 个,单机群超过 2800 台机器节点,每天处理数据量高达 9000TB。百度对 Hadoop 进行了深度改造,开发了 HCE(HadoopC++)系统以及自己的日志分析平台、数据仓库系统等,为公司各个团队提供分析计算和存储服务。腾讯的分布式数据仓库 TDW(Tencent distributed Data Warehouse)基于 Hadoop 和 Hive 而构建,改变了传统数据仓库无法线性扩展、可控性差的缺陷,目前已改为基于 Spark 框架而构建。华为构建了 FusionInsight[56]大数据平台,通过实时数据处理引擎,以事件驱动(Event-driven)模式解决高速事件流的实时计算问题。中国移动在通信领域广泛使用 Hadoop,基于 MapReduce 对其数据处理的分布式计算模式进行改造,利用了 HDFS 来实现分布式存储,开发了数据挖掘工具集 BC-PDM 和 HugeTable 数据仓库等系统。

结束语 本文从大数据、云计算起源和概念引入方面阐明了 Hadoop 框架的基本原理与发展,对 MapReduce,HDFS,Yarn,Spark 相关技术与运行机制进行了简明的原理剖析,简单介绍了 Hadoop 生态系统以及在大数据处理环境下的基本架构。着重分析了 Hadoop 学术研究与发展现状,将其分为 MapReduce 算法的改进与创新、HDFS 技术的优化与创新、二次开发与其它技术相结合、应用领域创新与实践,分别对其进行了阐述与分析,最后从商业角度简述了国内外的应用。不足之处在于因篇幅有限,未涉及 Hadoop 安全领域,但云计算和 Hadoop 的安全问题之重要性不言而喻。

尽管在学术界改进 Hadoop 的算法如此多,但因为研究学习成本和性能局限性,大多数未被商业广泛采纳与应用。唯独 Apache 基金会的研究成果如 MapReduceV2,Spark,Hive,Yarn,Hbase 等得到了普遍的应用与认可。基于内存计算的 Spark 框架已经被广泛采纳与应用,正逐渐替代 Hadoop,但 Spark 依旧存在着很大的研究空间。我们认为今后研究工作可能侧重于以下 3 个方面:

(1)Hadoop 生态系统与 Spark,Storm 更加完美地融合,改进 Hadoop 实时性、流处理、图处理迭代式计算上的不足,同时需解决大规模异构条件下的自适应匹配和效率以及公平问题。

(2)Hadoop 与 Spark 生态系统数据仓库的性能的提升,与传统相关型数据库更完美地兼容,使 SQL 查询等功能在 Hadoop 上高效运行。

(3)Hadoop 以及 Spark 的安全问题,集群与集群间、节点与节点之间通信认证、身份识别、访问权限、数据隔离与备份等安全问题以及其相关开销优化仍为研究的重点。

综上所述,MapReduce 和 Spark 相关技术必将和越来越多的学科领域相结合,改变人们的思维方式和商业模式。由于 Hadoop 和 Spark 等无法单一解决所有大数据问题,毫无疑问,大数据处理的深度和广度必将孕育出更为成熟强大、更具有普适性的一套综合性的大数据分析处理系统。

参 考 文 献

[1] Big data: science in the petabyte era [OL]. [2015-9-10]. http://www.nature.com/nature/journal/v455/n7209/edsumm/e080904-01.html

[2] Mayer-Schönberger V, Cukier K. BIG DATA [M]. Hodder Export, 2013

[3] Wang L Z, Laszewski G, Younge A, et al. Cloud computing: a perspective[J]. ACM SIGCOMM Computer Communication Review, 2009, 39(1): 50-55

[4] Ghemawat S, Gobioff H, Leung S T. The google file system [C]// Proc of the 19th ACM Symposium on Operating System Principles. 2003: 29-43

[5] Dean J, Ghemawat S. MapRecuce: Simplified data processing on large clusters[C]// Proc of the 6th Symposium on Operating System Design and Implementation. 2004: 137-150

[6] Chang F, Dean J, Ghemawat S, et al. A distributed storage system for structured data[C]// Proc of the 7th USENIX Symp. on Operating Systems Design and Implementation. 2006: 205-218

[7] Apache. Hadoop [OL]. [2015-9-10]. http://Hadoop.apahce.org/index.html

[8] White T. Hadoop: The Definitive Guide[M]. 周敏奇, 王晓玲, 金澈清, 钱卫宁, 译. Hadoop: 权威指南. 清华大学出版社, 2014

[9] Srinath Perera Thilina Gunarathne. Hadoop MapRecuce Cookbook[M]. 北京: 人民邮电出版社, 2015

[10] 刘军. Hadoop 大数据处理[M]. 北京: 人民邮电出版社. 2013

[11] Apache. HDFS Architecture Guide[OL]. [2015-9-10]. http://hadoop.apahce.org/docs/1.6.1/hdfs_design.html

[12] Lam C. Hadoop in action[M]. USA: Manning Publications, 2012

[13] Yan C R, Li T, Huang Y F, et al. Hmfs: efficient support of small files processing over HDFS[J]. Algorithms Archit Parallel Process Lect Notes Comput Sci., 2014, 86(31): 54-67

[14] Liu X, Yu Q, Liao J. FastDFS: a high performance distributed file system. [J]. ICIC Express Lett Part B Appl Int J Res Surv, 2014, 5(6): 1741-1746

[15] Dong Xin-hua, Li Rui-xuan, Zhou Wan-wan. Performance Optimization and Feature Enhancements of Hadoop System [J]. Journal of Computer Research and Development, 2013, 50(S2): 1-15(in Chinese)
董新华, 李瑞轩, 周湾湾, 等. Hadoop 系统性能优化与功能增强综述[J]. 计算机研究与发展, 2013, 50(S2): 1-15

[16] 王晓华. MapReduce2.0 源码分析与编程实战[M]. 北京: 人民邮电出版社, 2014

[17] Apache. Hadoop NextGen MapReduce(YARN) [OL]. [2015-9-10]. http://hadoop.apache.org /docs/current2/ hadoop-yarn/hadoop-yarn-site/ YARN.html

[18] Apache. Spark[OL]. [2015-9-20]. http://spark.apache.org

[19] 高彦杰. Spark 大数据处理: 技术、应用与性能优化[M]. 北京: 机械工业出版社, 2014

[20] Yang H, Dasdan A, Hsiao R L, et al. Map-reduce-merge: Simplified relational data processing on large clusters[C]// Proc of the 2007 ACM SIUMOD Int Conf on Management of Data (SIUMOD'07). New York: ACM, 2007: 1029-1040

[21] Verma A, Zea N, Cho B, et al. Breaking the MapReduce stage barrier [C]// Proc of 2010 IEEE Int Conf on Cluster Computing (CLUSTER'10). Piscataway, NJ: IEEE, 2010: 235-244

[22] Verma A, Cherkasova L, Campbell R H. Two sides of a coin: Optimizing the schedule of MapReduce jobs to minimize their makespan and improve cluster performance [C]// Proc of the 20th IEEE Int Symp on Modeling. Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS'12). Piscataway, NJ: IEEE, 2012: 11-18

[23] Valvag S V, Johansen D. Ovios: simple and efficient distributed data processing[C]// Pro of the 10th IEEE Int Conf on High Performance Computing and Communications (HPCC'08). Piscataway, NJ: IEEE, 2008: 113-122

[24] Bu Y, Howe B, Balazinska M, et al. HaLoop: Efficient iterative data processing on large clusters[J]. Proc of the VLDB Endowment, 2010, 3(1/2): 285-296

[25] Bu Y, Howe B, Balazinska M, et al. The HaLoop approach to large-scale iterative data analysis[J]. VLDB Journal, 2012, 21(2): 169-190

[26] Nandakumar V. Transparent in-memory cache for Hadoop-MapReduce[D]. Master of Applied Science Graduate Department of Electrical and Computer, 2014

[27] Chen Q, Zhang D, Cuo M, et al. Samr: A self-adaptive MapReduce scheduling algorithm in heterogeneous environment[C]// Proc of the 10th IEEE Int Conf on Computer and Information Technology (CIT'IO). Piscataway, NJ: IEEE. 2010: 2736-2743

[28] Kwon Y C, Balazinska M, Howe B, et al. Skewtune Mitigating skew in mapreduce applications[C]// Proc of the 2012 ACM SIUMOD Int Conf on Management of Data (SIUMOD'12). New York: ACM, 2012: 25-36

[29] Li D, Chen Y, Hai R H. Skew-aware task scheduling in clouds [C]// Proc of the 6th IEEE Int Symp on Service Oriented System Engineering (SOSE'13). Piscataway, NJ: IEEE, 2013: 341-346

[30] Ahmad F, Chakradhar S T, Raghunathan A, et al. Tarazu: Optimizing MapReduce on heterogeneous clusters[C]// Proc of the 17th Int Conf on Architectural Support for Programming Languages and Operating Systems (ASPLOS'12). NewYork: ACM, 2012: 61-74

[31] Cherniak A, Zaidi H, Zadorozhny V. Optimization strategies for A/B testing on HADOOP[J]. Proceedings of the VLDB Endowment, 2013, 6(11): 973-984

[32] Hu Jun, Hu Xian-De, Chen Jia-xing. Big Data Hybrid Computing Mode Based on Spark[J]. Computer Systems & Applications, 2015(24): 214-218(in Chinese)
胡俊, 胡贤德, 程家兴. 基于 Spark 的大数据混合计算模型[J]. 计算机系统应用, 2015(24): 214-218

[33] Fu Song-ling, Liao Xiang-ke, Huang Chen-lin, et al. FlatLFS: a lightweight file system for optimizing the performance of accessing massive small files[J]. Journal of National University of Defense Technology, 2013, 35(2): 120-126(in Chinese)
付松龄, 廖湘科, 黄辰林, 等. FlatLFS: 一种面向海量小文件处理优化的轻量级文件系统[J]. 国防科技大学学报, 2013, 35(2): 120-126

[34] Zhang Chun-ming, Rui Jian-wu, He Ting-ting. An approach for storing and accessing small files on Hadoop[J]. Computer Applications and Software, 2012(11): 95-100(in Chinese)
张春明, 芮建武, 何婷婷. 一种 Hadoop 小文件存储和读取的方法[J]. 计算机应用与软件, 2012(11): 95-100

(下转第 48 页)

[23] Balaban M, Maraee A, Sturm A, et al. A pattern-based approach for improving model quality [J]. Software System Model, 2015, 14(4): 1527-1555

[24] Ye Y, Jiang Z B, Diao X D, et al. Extended event-condition-action rules and fuzzy Petri nets based exception handling for workflow management [J]. Expert Systems with Applications, 2011, 38(9): 10847-10861

[25] Dines Bjørner. Software Engineering 2 Specification of Systems and Language[M]. Springer, 2006: 316-322

[26] Steinberg D, Budinsky F, Paternostro M, et al. EMF: Eclipse Modeling Framework(2nd Edition) [M]. Addison-Wesley Professional, 2008: 104-124

[27] ATLAS. ATL recognized as a standard solution for model transformation in Eclipse [EB/OL]. [2007-01-15]. http://www.eclipse.org/gmt

[28] Jouault F, Allilaire F, Bezivin J, et al. ATL: a model transformation tool [J]. Science of Computer Programming, 2008, 72(1/2): 31-39

[29] W3C. Web Service Choreography Interface(WSCI)(Version1.0) [EB/OL]. www.w3.org/TR/wsci

(上接第 11 页)

[35] Xiong An-ping, Huang Rong, Zou Yang. A kind of HDFS small files storage strategy based on hybrid index [J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2014, 27(1): 97-102(in Chinese)
熊安萍, 黄容, 邹样. 一种基于混合索引的 HDFS 小文件存储策略[J]. 重庆邮电大学学报(自然科学版), 2014, 27(1): 97-102

[36] Zhu Yuan-yuan, Wang Xiao-jing. HDFS optimization program based on GE coding [J]. Journal of Computer Applications, 2013, 33(3): 730-733(in Chinese)
朱媛媛, 王晓京. 基于 GE 码的 HDFS 优化方案[J]. 计算机应用, 2013, 33(3): 730-733

[37] Song Bao-yan, Wang Jun-Lu, Wang Yan. Optimized Storage Strategy Research of HDFS Based on Vandermonde Code[J]. Chinese Journal of Computers, 2015, 9(38): 1826-1836(in Chinese)
宋宝燕, 王俊陆, 王妍. 基于范德蒙码的 HDFA 优化存储策略研究[J]. 计算机学报, 2015, 9(38): 1826-1836

[38] Lu Mei-lian, Zhu Liang-liang. Load Balancing Strategy Based on CMM Model in HDFS[J]. Journal of Beijing University of Posts and Telecommunications, 2014, 10(37): 20-25(in Chinese)
卢美莲, 朱亮亮. 基于 CMM 模型的 HDFS 负载均衡策略[J]. 北京邮电大学学报, 2014, 10(37): 20-25

[39] Xi Ping, Xue Feng. Replica Placement Strategy Based on Multilayer Consistent Hashing in HDFS[J]. Computer Systems & Applications, 2015, 24(2): 127-133(in Chinese)
席屏, 薛峰. 多层一致性哈希的 HDFS 副本放置策略[J]. 计算机系统应用, 2015, 24(2): 127-133

[40] Kun L, Dai Dong, Sun Ming-ming. HDFS+: Concurrent Writes Improvements for HDFS[C]// Proc of IEEE International Conference on Big Data. 2013: 182-183

[41] Islam N S, Lu X, Wasi-ur-Rahman M, et al. In-Memory I/O and Replication for HDFS with Memcached: Early Experiences[C]// Proc of IEEE International Conference on Big Data. 2014: 213-218

[42] Ekanayake J, Li H, Zhang B, et al. Twister: A runtime for iterative mapreduce [C]// Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing. ACM. 2010: 810-818

[43] Abouzeid A, Bajda-Pawlikowski K, Abadi D J, et al. HadoopDB: An architectural hybrid of MapReduce and DBMS technologies for analytical workloads [J]. PVLDB, 2009, 2(1): 922-933

[44] Abouzeid A, Bajda-Pawlikowski K, Adadi D J, et al. HadoopDB in action: Building real world applications [C]// Elamagarmid AK, Agrawal D, eds. Proc of th SIGMOD. Indiana: ACM Press, 2010: 1111-1114

[45] Qin X, Wang H, Li F, et al. Beyond Simple Integration of RDBMS and MapReduce—Paving the Way toward a Unified System for Big Data Analytics: Vision and Progress[C]// 2012 Second International Conference on Cloud and Green Computing (CGC). IEEE, 2012: 716-725

[46] Talbot J, Yoo R M, Kozyrakis C. Phoenix: modular MapReduce for shared-memory systems[C]// Proceedings of the Second International Workshop on MapReduce and its Applications. ACM. 2011: 9-16

[47] Xu T, Wang D S, Liu G D. Banian: A Cross-Platform Interactive Query System for Structure Big Data[J]. Tsinghua Science and Technology, 2015, 7(11): 62-71

[48] Anderson Q. Storm real-time processing cookbook [M]. Birmingham: Packt Publishing, 2015

[49] Jin Yong-chao, Wu Huai-gu. Research on the Big Data Process Framework Based on Storm and Hadoop[J]. Modern Computer. 2015(3): 1419-1423(in Chinese)
靳永超, 吴怀谷. 基于 Storm 和 Hadoop 的大数据处理架构的研究[J]. 现代计算机, 2015(3): 1419-1423

[50] Lee R, Luo T, Huai Y, et al. Ysmart: Yet Another Sql-to-mapreduce Translator[C]// 2011 International Conference on Distributed Computing Systems(ICDCS). IEEE, 2011: 25-36

[51] Zhao Huan, Chen Xi. Chinese Tourism Information Search Platform based on Cloud Computing [C]// International Industrial Informatics and Computer Engineering Conference (IIICEC 2015). Beijing, 2015: 1236-1240

[52] Chen Hong. Research on Chinese segmentation algorithm based on Hadoop cloud platform [C]// Information Technology and Mechatronics Engineering Conference(ITOEC 2015). 2015: 134-138

[53] Hadoop 各商业发行版之比较[OL]. [2016-1-10]. http://www.cnblogs.com/iceTing/p/3392362.html

[54] 翟周伟. Hadoop 核心技术[M]. 北京: 机械工业出版社, 2015: 4

[55] Hadoop 目前在国内外的现状介绍[OL]. [2015-9-10]. http://www.thebigdata.cn/Hadoop/14142.html?utm_source=tuicool

[56] FusionInsight 大数据平台[OL]. [2016-1-10]. http://carrier.huawei.com/cn/products/IT/cloud-computing/fusioninsight