DOI:10.3969/j.issn.1009-6868.2016.02.003
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.TN.20160225.1736.004.html

大数据分析平台——从扩展性优先到性能优先

Big Data Analytic Platforms: Changing the Priority from Scalability to Performance

郑纬民/ZHENG Weimin
陈文光/CHEN Wenguang
(清华大学 计算机科学与技术系,北京 100084)
(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

中图分类号:TP393 文献标志码:A 文章编号:1009-6868 (2016) 02-0011-003

摘要: 认为现有以MapReduce/Spark等为代表的大数据处理平台在解决大数据问题的挑战问题方面过多考虑了容错性,忽视了性能。大数据分析系统的一个重要的发展方向就是兼顾性能和容错性,而图计算系统在数据模型上较好地考虑了性能和容错能力的平衡,是未来的重要发展方向。

关键词: 大数据;分布与并行处理;并行编程;容错;可扩展性

Abstract: Existing big data analytic platforms, such as MapReduce and Spark, focus on scalability and fault tolerance at the expense of performance. We discuss the connections between performance and fault tolerance and show they are not mutually exclusive. Distributed graph processing systems are promising because they make a better tradeoff between performance and fault tolerance with mutable data models.

Key words: big data; distributed and parallel processing; parallel programming; fault tolerance; scalability

随着信息化技术的发展,人类可以产生、收集、存储越来越多的数据,并利用这些数据进行决策,从而出现了大数据的概念。大数据的定义很多,比较流行的定义是Gartner公司提出的简称为3V的属性,即数据量大(Volume),到达速度快(Velocity)和数据种类多(Variety)。大数据分析利用数据驱动的方法,在科学发现、产品设计、生产与营销、社会发展等领域具有应用前景。

由于大数据的3V属性,需要在多台机器上进行分布与并行处理才能满足性能要求,因此传统的关系型数据库和数据挖掘软件很难直接应用在大数据的处理分析中。传统的超级计算技术,虽然具有很强的数据访问和计算能力,但其使用的MPI编程模型编程较为困难,对容错和自动负载平衡的支持也有缺陷,主要运行在高成本的高性能计算机系统上,对于主要在数据中心运行的大数据分析不是非常适合。

为了解决大数据的分析处理所面临的编程困难,负载不平衡和容错困难的问题,业界发展出了一系列技术,包括分布式文件系统、数据并行编程语言和框架以及领域编程模式来应对这些挑战。以MapReduce[1]和Spark[2]为代表的大数据分析平台,是目前较为流行的大数据处理生态环境,得到了产业界的广泛使用。

但是在文章中,我们通过分析认为:MapReduce和Spark系统将容错能力作为设计的优先原则,而在系统的处理性能上做了过多的让步,使得所需的处理资源过多,处理时间很长,这样反而增加了系统出现故障的几率。通过进一步分析性能与容错能力的关系,我们提出了一种性能优先兼顾扩展性的大数据分析系统构建思路,并以一个高性能图计算系统为例,介绍了如何用这种思路构建大数据分析系统。

1 以MapReduce/Spark为代表的大数据分析平台

现有的大数据分析平台主要基于开源的Hadoop系统,该系统使用Hadoop分布式文件系统(HDFS),通过多个备份的方法保证大量数据的可靠存储和读取性能,其上的Hive[3]系统支持数据查询,Hadoop MapReduce则支持大数据分析程序的开发。

与传统的并行编程方法MPI[4]相比,MapReduce是近年来并行编程领域的重要进展。尽管Map和Reduce

收稿时间:2016-01-14
网络出版时间:2016-02-25
基金项目:国家重点基础研究发展("973")计划(2014CB340402);国家自然科学基金(61525202)

在函数语言中早已被提出，但将其应用于大规模分布并行处理应归功于Jeff Dean和Ghemewat Sanjay。在MapReduce并行编程模型中，用户仅需要编写串行的Map函数体和Reduce函数体，MapReduce框架就可以完成并行的计算，并实现了自动容错和负载均衡。这对于数据中心中采用的异构服务器、低成本服务器集群是非常重要的。MapReduce开始仅能在使用通用中央处理器（CPU）的分布式系统上运行，但后来被移植到图形处理器（GPU）和多种加速器上。

MapReduce需要将中间结果保存到磁盘中，从而大大影响了性能，美国加州伯克利大学提出的Spark系统可以看做是基于内存的MapReduce模型，通过将中间结果保存在内存中，大大提高了数据分析程序的性能，类似思路的系统还包括HaLoop[5]和Twister[6]等。

Spark和MapReduce在大数据领域取得了巨大的成功，已经成为事实上的大数据处理标准。它们与分布式文件系统HDFS、查询系统Hive都集成在Hadoop系统中，为大数据的存储、查询和处理提供了相对完整的解决方案。这一系统也具有完整的开源社区支持和商业公司支持，HortonWorks和Cloudera提供Hadoop的发行版和服务，DataBricks为Spark提供发行版和服务。IBM于2016年宣布将投入10亿美元开发Spark。

2 大数据分析平台性能的重要性

尽管以Spark/MapReduce为代表的大数据分析平台已经得到了广泛应用，然而，其性能方面的问题也日益暴露出来。一些研究表明：对一些大数据分析问题来说，使用Spark在几十台机器上的性能甚至不如在某些优化过的程序在单机上的性能，例如对Twitter数据集来说，Spark在128个处理器核上需要857 s，而优化良好的单线程程序完成同样的处理功能仅需要300 s的时间[7]，即在中小规模数据集上Spark的性能功耗比比单线程程序要差2个数量级，甚至在绝对处理时间上也比单线程程序要慢。

Spark/MapReduce的性能问题，根源在于其设计理念上陷入了一个误区：即以容错能力为优先的设计目标，忽视了处理性能。例如，MapReduce和Spark都采用只读数据集的概念，这一方面大大方便了系统进行容错，但也使得系统在处理相当一部分应用时，性能会受到严重影响。例如，对于广泛使用的广度优先图搜索问题，需要记录哪些结点被访问过，这个数据集如果是只读的，就只能在每次遍历迭代时生成新的数据集，这会大大增加所需的内存复制操作和内存容量需求，使得性能大大下降。

而实际上处理性能的提高，对提高系统的容错能力也是有正面意义的。一个数据分析任务的总执行时间，可以按如式(1)估算（为描述方便，公式中略有简化）：

总执行时间 = 无故障执行时间①+无故障时容错机制开销②+故障发生概率*无故障执行时间*单次故障恢复时间③ (1)

Spark的设计主要对②进行优化，即通过只读数据集简化无故障容错机制的开销，却大大增加了①的无故障执行时间，而③实际是与①正相关的，即相同机器数，执行时间越长，出故障的概率越大，所需故障恢复时间也就越长。

从上面的分析可以看出：Spark的设计理念，即使对容错本身来说，也很难说是合理的，因为如果性能损失太大，无故障执行时间增加太多，会使得在②减少的开销被③抵消甚至超越[8]。

因此，我们认为：大数据分析系统的一个重要的发展方向就是兼顾性能和容错性。我们需要进一步在编程模型和框架上开展研究，在保持自动负载平衡和一定容错能力的基础上，提供优化的系统性能。

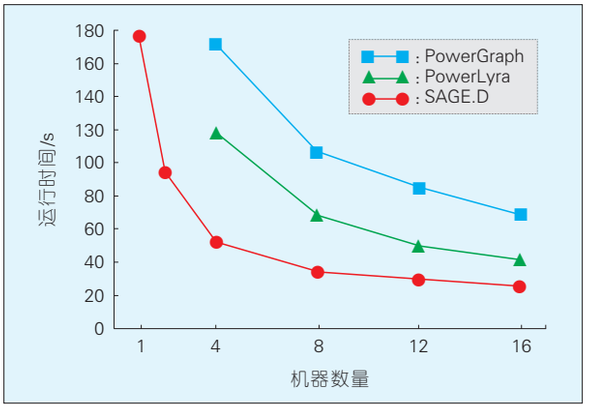
以Pregel[9]和GraphLab[10]等的图计算编程框架是这一类工作的代表，这些编程模型主要提供了基于图结点（vertex）的编程抽象，并沿着图的边进行通信，与Map-Reduce相比，这类图编程框架在处理图数据（如社交网络、航运网络和生物网络等）时比Map-Reduce/Spark的表达更加自然，所获得的性能也要好得多。这方面的工作引起了全球研究者和工业界的广泛关注，这些工作针对图计算中的负载不均衡、随机访问多、同步和异步等问题提出了解决方案。PowerGraph[11]和PowerLyra[12]系统是在GraphLab上改进后的图计算系统，其性能比GraphLab又有显著提高。GridGraph[13]提出了利用二维混洗的数据结构对图计算进行优化，可以有效减少图计算中的随机内存访问，提高处理性能。基于GirdGraph的分布式图计算系统SAGE.D其性能比PowerLyra进一步又提高了1倍左右。如图1所示：SAGE.D可以在16台机器上以30 s的时间内完成Twitter数据集的20次PageRank迭代，性能比Spark提高了接近30倍。

我们可以看到：在某些分析任务上，基于图计算系统的性能比基于Spark的分析系统快1～2个数量级。这意味着基于图计算系统在执行期间内发生错误的机会仅为Spark的1/10以下，从而不仅在执行性能方面，在容错能力方面也优于Spark。

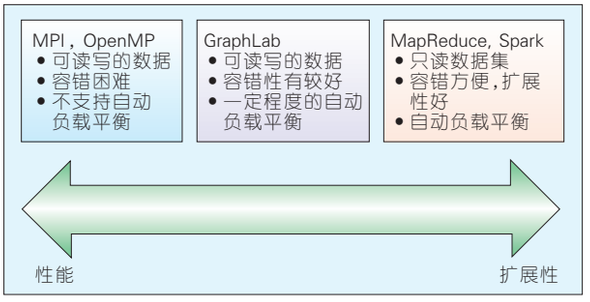
3 大数据问题展望

未来的大数据问题会呈现两种趋势：

（1）具有较小上限的大数据问题。以社交网络的分析问题为例，目前Facebook有约10亿活跃用户，用户之间的关注关系大约有1 000亿个，大约需要几个TB的内存容量。社交网络的结点是用户，地球上只有几十亿人口，社交网络的分析问题其上限



◀图1
在Twitter数据集上图计算系统完成20轮PageRank算法迭代的时间



◀图2
不同并行编程模型在设计理念和运行时支撑方面的差异

就是将全部人口数作为网络结点。

随着摩尔定律的持续作用，我们今天已经可以很容易地买到内容容量为TB量级的服务器，今后可望达到几十甚至数百TB。不断增长的硬件能力与较小上限的大数据问题相遇的结果，就是把今天的大数据问题变为明天的小数据问题，把今天需要数十、数百服务器解决的问题变为今后只需要几台甚至单台服务器就可以解决的问题。

针对这类应用，显然性能优化的大数据分析处理平台能够获得更好的性价比。

（2）具有较大上限的大数据问题。高性能计算中的很多问题规模具有非常大的上限，例如气候模拟，需要将空间分成网格、时间分片，显然空间上和时间上的进一步细分都会导致计算量和存储量的大幅度增加，人类已有的计算能力还远远无法满足高精度气候模拟的要求。针对这类应用，性能优化的大数据分析处理平台能够通过减少运行时间，提高系统的处理效率和处理规模。图2展示了不同并行编程模型在设计理念和运行时支撑方面的差异。

综上所述，现有以Spark为代表的大数据处理平台在解决大数据问题的挑战问题方面过多考虑了容错性，忽视了性能。我们认为图计算系统在数据模型上较好地考虑了性能和容错能力的平衡，是未来的重要发展方向。

参考文献

[1] DEAN, JEFFREY, SANJAY G. MapReduce: Simplified Data Processing on Large Clusters [J]. Communications of the ACM, 2008, 51 (1): 107–113. DOI: 10.1145/1327452.1327492
[2] ZAHARIA M, CHOWDHURY M, DAS T, et al. Resilient Distributed Datasets: A Fault–Tolerant Abstraction for In–Memory Cluster Computing[C]// Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. USA: USENIX Association, 2012:15–28
[3] THUSOO A, SARMA S J, JAIN N, et al. Hive: A Warehousing Solution over a Map–Reduce Framework [J]. Proceedings of the VLDB Endowment, 2009, 2(2): 1626–1629. DOI: 10.14778/1687553.1687609
[4] GROPP W, LUSK E, DOSS N, et al. "A High–Performance, Portable Implementation of the MPI Message Passing Interface Standard [J]. Parallel Computing, 1996, 22(6): 789–828. DOI: 10.1016/0167–8191(96)00024–5
[5] BU Y, HOWE B, BALAZINSKA M, et al. HaLoop: Efficient Iterative Data Processing on Large Clusters [J]. Proceedings of the VLDB Endowment, 2010, 3(1): 285–296. DOI: 10.14778/1920841.1920881
[6] EKANAYAKE, JALIYA. Twister: A Runtime for Iterative Mapreduce [C]//Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing. USA: ACM, 2010: 810–818
[7] FRANK M, MICHAEL I, MURRAY D G. Scalability! But at what COST [C]//5th Workshop on Hot Topics in Operating Systems (HotOS XV). USA: USENIX Association, 2015
[8] KWAK, HAEWOON. What is Twitter, A Social Network or A News Media? [C]/ Proceedings of the 19th International Conference on World Wide Web. USA: ACM, 2010: 591–600
[9] MALEWICZ, GRZEGORZ. Pregel: A System for Large–Scale Graph[C]// Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. USA: ACM, 2010: 135–146
[10] LOW, YU C. Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud [J].Proceedings of the VLDB Endowment, 2012, 5(8): 716–727
[11] GONZALEZ, Joseph E. PowerGraph: Distributed Graph–Parallel Computation on Natural Graphs [J]. OSDI, 2012, 12(1): 23–27
[12] CHEN R. Powerlyra: Differentiated Graph Computation and Partitioning on Skewed Graphs[C]//Proceedings of the Tenth European Conference on Computer Systems. USA: ACM, 2015: 1–15
[13] ZHU X, HAN W, CHEN W. GridGraph: Large–Scale Graph Processing on a Single Machine Using 2–Level Hierarchical Partitioning[C]//Proceedings of the Usenix Annual Technical. USA: ASM, 2015: 375–386

作者简介

郑纬民，清华大学计算机科学与技术系教授、博士生导师；长期从事计算机系统结构、大规模数据存储、高性能计算等领域的科研教学工作；主持并完成了“973”、“863”、自然科学基金等科研项目36项，负责或参与工程项目11项；获国家科技进步一等奖1次，国家科技进步奖二等奖2次，国家技术发明奖二等奖1次；发表论文500余篇，著作10部。

陈文光，清华大学计算机科学与技术系教授，现为中国机算机学会杰出会员、副秘书长、杰出讲者，青年科技论坛（YOCSEF）荣誉委员，ACM中国理事会副主席等；获国家科技进步二等奖1次，部级科技一等奖2次；发表文章50余篇。