文章编号:1671-4598(2017)03-0195-02 DOI:10.16526/j.cnki.11—4762/tp.2017.03.053 中图分类号:TP393 文献标识码:A

双重并行环境下最短路径的研究

孙玉强,李银银,顾玉宛
(常州大学 信息科学与工程学院,江苏 常州 213164)

摘要:并行问题和最短路径问题已成为一个热点研究课题,传统的最短路径算法已不能满足数据爆炸式增长的处理需求,尤其当网络规模很大时,所需的计算时间和存储空间也大大的增加;MapReduce模型的出现,带来了一种新的解决方法来解决最短路径;GPU具有强大的并行计算能力和存储带宽,与CPU相比具有明显的优势;通过研究MapReduce模型和GPU执行过程的分析,指出单独基于MapReduce模型的最短路径并行方法存在的问题,降低了系统的性能;论文的创新点是结合MapReduce和GPU形成双并行模型,并行预处理数据,针对最短路径中的数据传输和同步开销,增加数据动态处理器;最后实验从并行算法的性能评价指标平均加速比进行比较,结果表明,双重并行环境下的最短路径的计算,提高了加速比。

关键词:最短路径;并行计算;MapReduce;GPU;数据动态处理器

Research on Shortest Path in Dual Parallel Environment

Sun Yuqiang, Li Yinyin, Gu Yuwan
(School of Information Science & Engineering, Changzhou University, Changzhou 213164, China)

Abstract: Parallel problem and shortest path problem has become a hot research topic, traditional shortest path algorithm cannot meet the demand of the explosive growth of the data processing, especially when the network size is large, the computation time and storage space required is greatly increased. The emergence of MapReduce model, brings a new solution to solve the shortest path. GPU has powerful parallel computing capability and storage bandwidth, and CPU has obvious advantages. By studying MapReduce model and GPU implementation process analysis, pointed out the shortest path parallel method based on MapReduce model alone existing problems, and reduce the performance of the system. The innovation of this paper is combine MapReduce and GPU to form double parallel model, parallel preprocessing data, the data transfer and synchronization overhead for the shortest patht, increase data dynamic processor. Compared with the average speedup of performance evaluation index of parallel algorithm, the results show that the computation of the shortest path in double parallel environment improves the speedup.

Keywords: shortest path; parallel computing; MapReduce; GPU; data dynamic processor

0 引言

最短路径问题是数学界和计算机科学中的一个重要的研究课题,在数学建模的基础上,解决了城市规划、物流、交通管理、数字导航等领域的实际问题[1]。并行计算为最短路径提供了一种新的解决方案,尤其是在动态网络中最短路径的处理中。

MapReduce模型可以解决计算机的存储容量和数据的爆炸性增长的计算能力缺乏之间的矛盾,因为MapReduce本身具有封装数据分区、负载均衡、容错处理等细节,用户只需要将实际应用中的问题分为几个问题,它们是可并行操作的子问题[2],从而可有效降低解决问题的难度。

与CPU不同,GPU具有特殊的硬件结构,计算能力和处理能力是CPU十倍甚至几十倍,处理器带宽更高,功耗更低,因此,使用GPU加速MapReduce计算得到了广泛的关注。

1 研究背景

1.1 最短路径概述

给出最短路径的加权有向图 $G=(V, E, W)$,其中 V 是顶点集和边集 E,W 是权重集,每一个边的权重是非负实数[3];V 的顶点,称为源,计算最短路径从源到所有其他点的长度,其中长度是指边的权重的和;按照现实情况,边的权重可以通过时间、距离、成本、损失等来表示,并且是最小值[4]。

1.2 MapReduce模型

在该模型中,有两个过程,Map和Reduce。在Map阶段,将初始输入数据转化为键-值对<key, value>,然后将数据分布到集群中的所有计算节点进行并行处理,得到中间结果集[5];而Reduce阶段将对那些具有同一的中间结果进行处理,以获得终极的输出记录。

1.3 GPU

GPU设计用于高度密集型的并行计算,具有强大浮点计算能力,高带宽、隐藏的延迟和高性能的多处理器阵列的存储系统,主要用于大量线程的计算。

2 双重并行环境下最短路径的设计

2.1 双重并行计算的提出

GPU和MapReduce结合的原因如下:

(1) 目前,对GPU和MapReduce并行计算单独的研究一直无法满足需求,GPU具有更好的数据宽度和并行计算的能力,GPU比CPU闲置的时间更多,GPU使用得当,它可以减少CPU的占用时间,而且可以使得过度空闲的GPU可以被充分利用。

(2) MapReduce模型需要大量的CPU、存储器、高性能的宽带网络,MapReduce模型的Map和Reduce操作,有时往往需要频频的CPU计算[6],当有大量的并行计算任务的时候,甚至高达100%的占用率,有必要平衡,GPU参与系统的计算能力。

收稿日期:2016-10-14; 修回日期:2016-11-17。

作者简介:孙玉强(1956-),男,博士,教授,主要从事并行计算方向的研究。

2.2 MapReduce 和 GPU 并行模型

图 1 显示了基于 GPU 的 MapReduce 并行模型图。CPU 首先读取硬盘或存储的数据文件，并将其分块，然后调用 GPU 的计算，在 GPU 中进行 Map 操作的每个数据块，并开始启动 Map 任务[5]；Map 任务将产生多个中间键－值对，然后 GPU 开始排序操作，中间键－值对按关键字排序；随后 CPU 把已排序的中间键－值重新分块，每一块递给 GPU 计算部分中的 Reduce 操作，然后开始 Reduce 任务。最后，得到 Reduce 任务的多个输出记录，并开始合并操作，得到最终的输出值给 CPU 调用[7]。

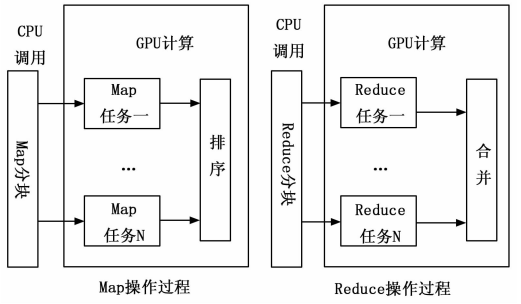

图 1 基于 GPU 的 MapReduce 并行模型图

2.3 MapReduce 和 GPU 双重并行最短路径实现

前期处理过程：

预处理是初级阶段，负责初始数据和节点之间的数据的初始分派。用户需要初始化系统，并根据应用程序的具体特点，设置合适的环境变量和不同的流程，如设置工作节点、键－值对的数据类型、GPU 的线程块儿的大小、完成所需的工作环节（map、reduce）、输出位置和方式[5]。之后初始的数据将被系统分成相同的大小，适于在 GPU 计算的数据块，这些块被分布到集群中的所有节点。

图 2 所示是基于 Mapreduce 的最短路径流程图。其中有两个子阶段降低了系统的性能。一个是中间结果写到磁盘的 map 阶段，主要目的是提高系统的可靠性，但是以牺牲系统性能为代价，因为对象的建立、销毁、垃圾回收等，会花很多时间。

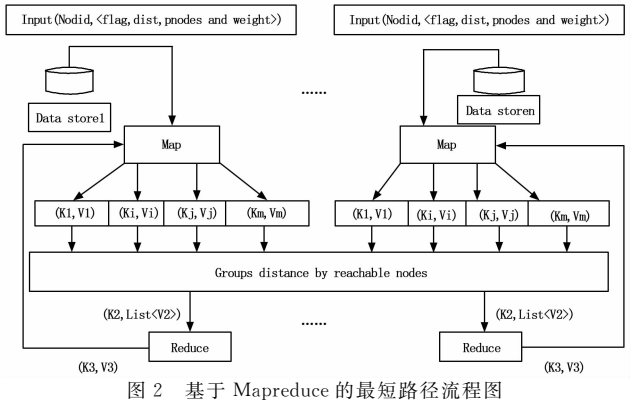

图 2 基于 Mapreduce 的最短路径流程图

另一个阶段是将 map 的结果通过网络传送给 Reduce 节点的过程，混合排序操作，中间结果传输期间的网络和同步成本是降低系统性能的一个重要因素。

针对上述问题进行改进，图 3 显示了 MapReduce 和 GPU 的双并行处理机制，GPU 嵌入 Hadoop，结合 GPU 和多核心 CPU 资源。在实现时，或因为 CPU 和 GPU 的数据传输和同步开销太大，不能发挥 GPU 的优点。为此添加数据动态处理器，有了动态数据处理器，能够实时监测计算节点 GPU 内存，数据的动态分解与组合，确保 GPU 计算最适合的数据，GPU 速度优化。

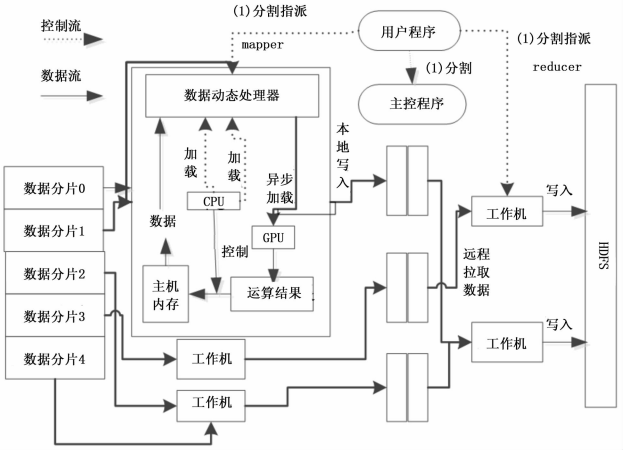

图 3 MapReduce 和 GPU 双重并行处理机制

GPU 完成大规模数据的并行计算任务，其他 GPU 代替不了 CPU 执行的非计算任务仍由 CPU 完成，如对硬盘的访问，访问物理地址，读取和写入文件。

当用户程序调用 Map 函数时，读取本地文件（key－value 形式），产生输出结果后，数据分片函数将输出记录分割成几个不同的块，具体过程如下：

首先，任务通过 MapReduce 机制分发，将数据传输到工作节点，实现第一级多节点并行，然后基于节点的配置每个节点使用 GPU 等并行方式实现第二级并行，工作节点基于任务的大小和当前节点的 GPU 性能将数据发送到数据动态处理器。动态数据处理器分割或组合数据以创建满足 GPU 功能（确保高速、全负载 GPU 操作）的数据块，数据被发送到 GPU 处理[8]。最后，GPU 计算结果被传输到主机内存进行后续处理，这种多级的处理机制可显示 GPU 的计算能力。

与 Map 阶段不同，Reduce 阶段具有同一的中间键－值对记录在一起，最后，按照用户的需要，合并或存储结果，并且被系统最终清理。

3 实验结果

前面介绍了 MapReduce 和 GPU 双并行模型及处理机制和最短路径的实现，下面进行实验测试。实验使用两台普通服务器、一台 D－Link 百兆交换机组成的集群上，在运行调试之前，需要执行以下步骤：首先通过 CUDA 开发工具编译 map 函数，将 C 语言文件编译为 DLL 库文件，然后通过使用 java 语言调用本地库文件的 JNI 方法来编译系统框架，配置环境基本上是一样的。测试结果如表 1 所示。实验数据显示了从 10 000个结点到 10^5 个结点，再到 10^6 及 10^7 个结点求解最短路径所用的时间。

表 1 运行时间对比表

结点	MapReduce 并行计算	普通并行计算	GPU 和 MapReduce 双重并行计算	平均加速比
10^4	20.542s	26.456s	13.328s	1.41
10^5	65.598s	80.652s	39.592s	1.62
10^6	124.657s	150.223s	71.242s	1.75
10^7	303.412s	376.815s	180.485s	1.87

（下转第 230 页）

本并没有随着任务量的减小而明显减小。这是因为，当 ST 增加时，满足截止时间约束边缘的任务较多。由于这些任务所允许的执行时间较紧迫，所以调度器被迫安排一些高价格的能力较强的资源来执行它，所以总体成本不会明显降低。

表 5 给出了 2 种调度方案所分配的资源类型和数量。正如上述分析，ST 越大，调度器所调用的高级资源越多。总体来说，提出 ILP 模型所调用的普通和高级资源数量都要小于 SLAA 方案，所以具有较小的执行成本。

表 5 资源配置

调度场景		SLAA	ILP
启动时间	ST=0	28 * r3. large	24 * r3. large
	ST=10	27 * r3. large	23 * r3. large
	ST=20	28 * r3. large+ 1 * r3. xlarge	21 * r3. large+ 1 * r3. xlarge
	ST=30	22 * r3. large+ 3 * r3. xlarge	17 * r3. large+ 2 * r3. xlarge
	ST=40	18 * r3. large+2 * r3. xlarge+2 * r3. 2xlarge	16 * r3. large+ 3 * r3. xlarge
	ST=50	14 * r3. large+5 * r3. xlarge+1 * r3. 4xlarge	11 * r3. large+3 * r3. xlarge+1 * r3. 2xlarge
	ST=60	21 * r3. large+4 * r3. xlarge+ 2 * r3. 2xlarge+1 * r3. 4xlarge	16 * r3. large+2 * r3. 2 xlarge+1 * r3. 4xlarge

4 结束语

为了提高云平台对大数据分析应用的执行效率，提出了一种 BDAaaS 架构，通过接纳控制器筛选出可执行的 BDAA 任务，并建立相应的 SLA。然后，通过 ILP 资源调度模型在满足 SLA 保证下为 BDAA 分配资源，以此最小化任务执行成本。在不同提交时间的任务申请下进行调度仿真，结果证明了提出方法能够有效降低执行成本，具有有效性和可行性。

参考文献：

[1] 李晓飞. 基于云计算技术的大数据处理系统的研究 [J]. 长春工程学院学报（自然科学版），2014，15（1）：116－118.

[2] 徐 聪. 大数据应用在云计算平台的优化部署与调度策略研究 [D]. 北京：清华大学，2015.

[3] Arun J，Hazaruthin M M，Karthik M. Analytics as a service delivery model for the cloud [A]. IEEE International Conference on Engineering and Technology [C]. IEEE，2015：1－5.

[4] Wu L，Garg S K，Buyya R. Service Level Agreement（SLA）Based SaaS Cloud Management System [A]. IEEE，International Conference on Parallel and Distributed Systems [C]. IEEE，2015：440－447.

[5] Alrokayan M，Vahid Dastjerdi A，Buyya R. SLA－Aware Provisioning and Scheduling of Cloud Resources for Big Data Analytics [A]. IEEE International Conference on Cloud Computing in Emerging Markets [C]. IEEE，2014：1－8.

[6] 王德文，刘晓萌. 基于改进粒子群算法的云计算平台资源调度 [J]. 计算机应用研究，2015，32（11）：3230－3234.

[7] 刘 曦，张潇璐，张学杰. 异构云系统中基于智能优化算法的多维资源公平分配 [J]. 计算机应用，2016，36（8）：2128－2133.

[8] 周芸韬. 基于 MQAAR 的移动自组织网络路由方案 [J]. 湘潭大学自然科学学报，2016，38（3）：69－73.

[9] 林清滢，陆锡聪，徐 林. 云计算中面向 SLA 的作业分层优先级调度策略 [J]. 计算机科学，2014，41（1）：316－317.

[10] Garg S K，Toosi A N，Gopalaiyengar S K，et al. SLA－based virtual machine management for heterogeneous workloads in a cloud datacenter [J]. Journal of Network & Computer Applications，2014，45（4）：108－120.

[11] Manzini R，Accorsi R，Cennerazzo T，et al. The scheduling of maintenance. A resource－constraints mixed integer linear programming model [J]. Computers & Industrial Engineering，2015，8（7）：561－568.

[12] 谢丽霞，严焱心. 云计算环境下的服务调度和资源调度研究 [J]. 计算机应用研究，2015，35（2）：528－531.

[13] Zhu L，Li Q，He L. Study on Cloud Computing Resource Scheduling Strategy Based on the Ant Colony Optimization Algorithm [J]. International Journal of Computer Science Issues，2012，9（5）：131－138.

[14] 张希翔，李陶深. 云计算下适应用户任务动态变更的调度算法 [J]. 华中科技大学学报自然科学版，2012，40（1）：165－169.

[15] Genez T A L，Bittencourt L F，Madeira E R M. Workflow scheduling for SaaS / PaaS cloud providers considering two SLA levels [J]. Network Operations & Management Symposium IEEE，2012，104（5）：906－912.

[16] Poola D，Ramamohanarao K，Buyya R. Enhancing Reliability of Workflow Execution Using Task Replication and Spot Instances [J]. Acm Transactions on Autonomous & Adaptive Systems，2016，10（4）：1－21.

（上接第 196 页）

相比上述实验结果，在计算最短路径时，结点相同的情况下，MapReduce 和 GPU 双重并行条件下最短路径的计算比 MapReduce 计算或普通的并行计算速度更快，所用的时间明显减少，GPU 加速的 MapReduce 模型，充分发挥优势，在结点增加时，平均加速比也有所增加，即双重并行条件下的最短路径的计算，提高了大规模数据并行处理的优势。

4 结论

本文利用 GPU 来加速 MapReduce 构成双并行模型，说明将 GPU 和 MapReduce 二者相结合的原因，将 GPU 应用到 MapReduce 过程中，实现多层次并行，研究了双重并行环境下最短路径的实现，加入了预处理和数据动态处理器。最后通过实验进行验证，结果表明，双重并行环境下最短路径的计算具有双重并行的效果。

参考文献：

[1] 张凌洁，赵英. 基于 GPU 的并行 APSP 问题的研究 [J]. 电子设计工程，2012，20（17）：15－18.

[2] 钮 亮，张宝友. MapReduce 求解物流配送单源最短路径研究 [J]. 电子技术应用，2014，40（3）：123－125.

[3] 杨 玲，李仁发，唐 卓. 基于 MapReduce 的单源最短路径算法研究 [J]. 微计算机信息，2011（12）：97－99.

[4] 王晓东. 算法设计与分析 [M]. 北京：清华大学出版社，2003.

[5] 郭钇汝. 基于 GPU 集群系统的 MapReduce 编程模型研究 [D]. 济南：山东大学，2014.

[6] 曾青华，袁家斌. 基于 MapReduce 和 GPU 双重并行计算的云计算模型 [J]. 计算机与数字工程，2013，41（3）：333－336.

[7] 瞿李峰. 基于 GPGPU 的 MapReduce 高性能并行计算模型研究与应用 [D]. 桂林：桂林理工大学，2009.

[8] 张 凯，秦 勃，刘其成. 基于 GPU－Hadoop 的并行计算框架研究与实现 [J]. 计算机应用研究，2014，31（8）：2548－2550.