基于Hadoop的云计算基础架构分析

李 响

(葫芦岛第一职业中专计研中心，辽宁 葫芦岛 125001)

摘 要：Hadoop是一个可实现大规模分布式计算的开源软件平台，已经被广泛应用在云计算领域。从Hadoop分布式文件系统的整体架构入手，描述了其分布式数据存储、分布式任务分配、分布式并行计算和分布式数据库四方面的核心内容，并论述了HDFS的工作原理、文件操作流程及Map/Reduce工作原理和计算过程。目的是使开发人员能深入地理解Hadoop架构的工作原理与实现过程，为云计算背景下的应用程序开发提供参考。

关键词：Hadoop；云计算；分布式并行计算；HDFS；Map/Reduce

中图分类号：TP338.8 **文献标志码**：A **文章编号**：1006-8228(2011)11-04-02

Analysis of Cloud Computing Infrastructure Based on Hadoop

LI Xiang

(*Computer Research Center, the First Vocational School, Huludao, Liaoning 125001, China*)

Abstract: Hadoop is an open-source software platform which can achieve large-scale distributed computing, so it is widely used in cloud computing. Starting from the distributed file system architecture of Hadoop, we describe its core content in the four aspects of distributed data storage, distributed task assignment, distributed parallel computing and distributed database, and discuss HDFS working principle, file operation process, as well as Map/Reduce working principle and computation procedure. The aim is to make developers further understand the working principle and implementing process of Hadoop architecture, and provide reference for application development under cloud computing background.

Key words: Hadoop; cloud computing; distributed parallel computing; HDFS; Map/Reduce

0 引言

随着时代的发展，人们对数据的海量存储和超级计算能力提出了更高的要求，这在过去几十年里促进了硬件的发展，使芯片集成度符合摩尔定律呈指数增长，但是硬件的发展受到了物理极限的约束。另外，由于传统并行编程模型应用的局限性，客观上要求一种容易学习、使用、部署的新的并行编程框架。因此，产生了云计算。云计算概念由Google提出，是对分布式处理、并行处理和网格计算及分布式数据库的改进处理。Google在2006年推出Google的企业服务即为云计算服务的雏形[1]，用户只需要通过浏览器连接到Google，就可以进行相应的存储和计算处理。Google还提供了Google Docs、Google Desktop等作为个人网络用户的在线软件应用及云计算模式的初步体验。随着云计算理念和应用的推广，IBM、微软、Amazon等信息业巨头都已经参与到云计算研究和开发中，并且Hadoop架构也应允产生了，它对用户开源并迅速发展起来。

1 Hadoop平台介绍

Hadoop是Apache开源组织的一个分布式计算开源框架，它可以运行在大型集群的廉价硬件设备上，实现对集群的控制和管理。而且Hapoop为应用程序透明地提供了一组稳定可靠的接口，屏蔽了并行应用开发的细节，实现更加便捷地构建企业级的应用，并且能够实现海量数据的管理和分布式数据处理。

Hadoop最核心的设计就是分布式文件系统HDFS和Map/Reduce算法模型。分布式文件系统HDFS是专门为Map/Reduce作业所设计的文件系统。但HDFS并不是用来处理随机存取数据的，HDFS的设计中更多考虑到了数据批处理，而不是用户交互处理，比之数据访问的低延迟问题，更关键的在于数据访问的高吞吐量。因此，HDFS是一个给应用提供高吞吐量的分布式文件系统[2]，可能由成百上千的机器所构成，每个机器上存储着文件系统的部分数据。计算模型Map/Reduce是Hadoop的核心计算模型[3]，是用于在集群上分布式处理大数据集的软件架构。它将复杂的运行于大规模集群上的并行计算过程高度抽象到了两个函数，Map和Reduce，这是一个简单而又强大的模型。

Hadoop还包括对于结构化数据处理的HBase、数据仓库的基础设施Hive、并行计算的高层次数据流语言及执行框架Pig和分布式应用的高性能协调服务Zookeeper等子项目。

2 Hadoop模型整体分析

2.1 Hadoop的分布式数据存储

2.1.1 HDFS的工作原理

HDFS采用Master/Slave结构，其工作原理如图1所示，即

收稿日期：2011-8-01

作者简介：李响(1979-)，男，辽宁省宽甸县人，讲师，计算机专业教师在读计算机工程硕士，研究方向：计算机信息处理。

由一个管理结点(NameNode)和多个数据节点(DataNode)组成,每个结点均是一台普通的计算机。NameNode是关键模块,进行文件系统元数据的管理和控制服务,对外提供创建、打开、删除和重命名文件或目录的功能。但其底层实现上是把文件切割成Block,然后这些Block分散地存储于不同的DataNode上,DataNode提供数据块存储和查询服务,并且负责处理数据的读写请求。每个Block还可以复制数份,存储于不同的DataNode上,达到容错容灾的目的。NameNode通过维护一些数据结构,记录了每一个文件被切割成了多少个Block,这些Block可以从哪些DataNode中获得,各个DataNode的状态等重要信息。

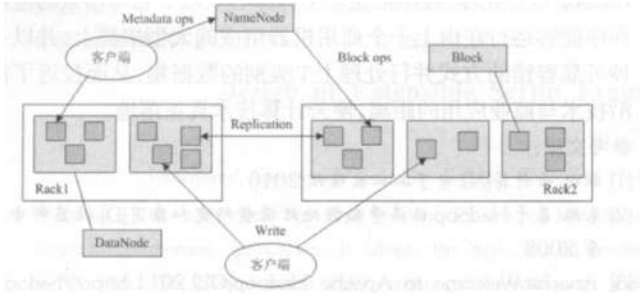


图1 HDFS的工作原理图

2.1.2 HDFS的保障可靠性措施

错误检测和快速、自动地恢复是HDFS最核心的架构目标。为了达到数据的稳定性,它把数据存储到多个存储节点上,这样就保证只要有一个数据副本存在,数据使用者就可以安全地使用这些数据。并且,HDFS还建立了数据节点心跳检测、数据完整性检测、安全模式、块报告及空间回收机制等来确保可靠性。

2.1.3 HDFS的文件操作流程

HDFS的"一次写入多次读取"的文件访问模型,即一个文件经过创建、写入和关闭之后就不需要改变,这使高吞吐量的数据访问成为可能。HDFS的具体文件操作流程如下。

⑴ HDFS写文件流程

①客户端把数据缓存到本地临时文件夹。

②临时文件夹数据超过64M,客户端联系NameNode,NameNode分配DataNode,DataNode依照客户端的位置被排列成一个有着最近物理距离和最小的序列。

③与序列的第一个数据服务器建立Socket连接,发送请求,然后等待回应,依次下传,客户端得到回包,流水线建立成功。

④正式发送数据,以4K为大小传送。

⑵ HDFS读文件流程

①客户端联系NameNode,得到所有数据块信息,以及数据块对应的所有数据服务器的位置信息。

②尝试从某个数据块对应的一组数据服务器中选出一个,进行连接。

③数据被一个包一个包地发送回客户端,等到整个数据块的数据都被读取完了,就会断开此链接,尝试连接下一个数据块对应的数据服务器,整个流程,依次如此反复,直到所有数据都读取完为止。

2.2 Hadoop的分布式任务分配

2.2.1 Hadoop任务分配工作原理

Hadoop任务分配工作原理如图2所示。Hadoop中有一个作为主控的JobTracker,用于调度和管理其他的TaskTracker,JobTracker可以运行于集群中任一台计算机上。TaskTracker负责执行任务,必须运行于DataNode上,即DataNode既是数据存储结点,也是计算结点。JobTracker将Map任务和Reduce任务分发给空闲的TaskTracker,让这些任务并行运行,并负责监控任务的运行情况。如果某一个TaskTracker出故障了,JobTracker会将其负责的任务转交给另一个空闲的TaskTracker重新运行。

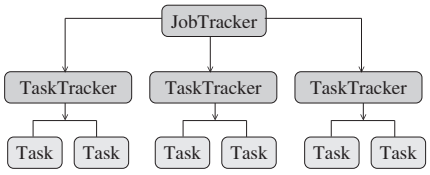


图2 Hadoop任务分配工作原理图

2.2.2 Hadoop的任务分配过程

①JobClient向JobTracker请求一个新的作业ID,并计算作业的输入划分。

②JobTracker接收作业后,为每个划分创建一个Map任务。

③TaskTracker定期发送心跳信号告诉JobTracker其是否存活。如果存活,JobTracker会为它分配一个任务,并使用心跳方法的返回值将任务传给TaskTracker。

④TaskTracker接受到任务后,开始执行任务(Task)。

2.3 Hadoop的分布式并行计算

2.3.1 Map/Reduce的工作原理

Map/Reduce的工作原理如图3所示。当一个计算作业向Map/Reduce框架提交时,Map/Reduce会首先把计算作业拆分成若干个Map任务[4],然后分配到不同的节点上去执行,每一个Map任务处理输入数据中的一部分,当Map任务完成后,它会生成一些中间文件,这些中间文件将会作为Reduce任务的输入数据。Reduce任务的主要目标就是把前面若干个Map的输出汇总到一起并输出。

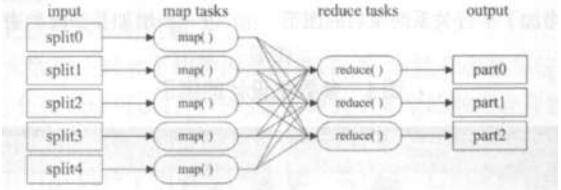


图3 Map/Reduce的工作原理图

2.3.2 Map/Reduce的计算过程

Map/Reduce计算模型的核心是Map和Reduce两个函数,这两个函数由用户负责实现,功能是按一定的映射规则将输入的<key,value>对转换成另一个或一批<key,value>对输出。Map过程通过在输入列表中的每一项执行函数,生成一系列的输出列表。Reduce过程再将Map的输出列表作为输入列表,随后生成一个聚集值,作为最后的输出,其中所有相同键值的列表被输入到同一个Reduce任务中。

另外，在Map前还会对输入的数据有Split(分割)的过程，保证任务并行效率，在Map之后还会有Shuffle(混合)的过程，用于提高Reduce的效率以及减小数据传输的压力。为了减少数据在网络上的传输，降低对网络带宽的需求，从而保证分布式计算的高效性。HDFS还为应用提供了将它们自己移动到数据附近的接口。

2.4 Hadoop的分布式数据库

2.4.1 HBase的工作原理

HBase的服务器体系结构也是遵从了简单的主从服务器架构，由HBaseMaster主服务器和Hregion服务器群构成，实现对大表的结构化数据的存储。对用户来说，每个表是一堆数据的集合，靠主键来区分。物理上，一张表被拆分成多块，每一块称为一个Hregion。使用用表名+开始/结束主键来区分一个Hregion。一个Hregion会保存一个表里面某段连续的数据，从开始主键到结束主键，一张完整的表格是保存在多个Hregion上面的，HBase通过管理整个区域某部分的节点来管理整个表。

HBase本质上是一个稀疏的，长期存储的(存在硬盘上)，多维度的，排序的映射表[5]。这张表的索引是行关键字、列关键字和时间戳。每个值是一个不解释的字符数组，数据都是字符串，没类型。用户在表格中存储数据，每一行都有一个可排序的主键和任意多的列。由于是稀疏存储的，所以同一张表里面的每一行数据都可以有截然不同的列。

2.4.2 Hive的工作过程

Hive在Hadoop的架构体系中承担了一个SQL解析的过程，查询语言会被Hive编译器编译成Map/Reduce任务，在整个任务执行中，HiveSQL任务经历了“语法解析→生成执行Task树→生成执行计划→分发任务→Map/Reduce执行任务计划”的这样一个过程。任务由执行引擎调度，具体执行在底层的Hadoop集群。

3 结束语

由于云计算体系结构的复杂性，要将该技术落实到企业应用是一项非常复杂的系统工程。由以上分析可见，Hadoop是一个使用简易的可用于云计算的软件框架，用户可以全身心地进行应用程序的开发，其他的并行编程中的种种复杂问题，如分布式存储，工作调度，负载平衡，容错处理，网络通信等，均由Hadoop负责处理，程序员完全不用操心。基于它写出来的应用程序能够运行在由上千个商用机器组成的大型集群上，并以一种可靠容错的方式并行处理上T级别的数据集，从而拉近了前沿技术与商业应用的距离，使云计算技术真正落地。

参考文献：

[1] 刘鹏.云计算[M].电子工业出版社,2010.

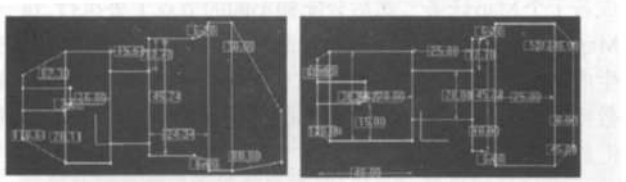
[2] 朱珠.基于Hadoop的海量数据处理模型研究和应用[D].北京邮电大学,2008.

[3] Apache.Welcome to Apache Hadoop[OL].2011.http://hadoop.apache.org/.

[4] 谢桂兰,罗省贤.基于Hadoop MapReduce模型的应用研究[J].微型机与应用,2010.8:4~7

[5] [美]怀特,周傲英,曹大聘,译Hadoop权威指南〔中文版〕[M].清华大学出版社,2010.

(上接第3页)



(c) 增加了平行关系约束后的图形 (d) 进一步增加并修改约束后的图形

图4 参数化设计的例子

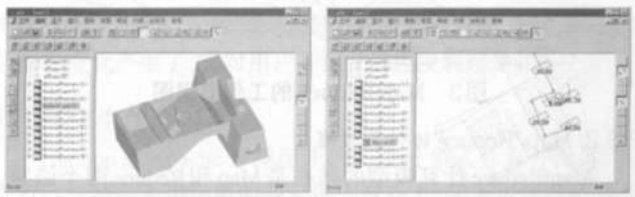


图5 基于约束求解的特征编辑

5 结束语

本文在分析了现有参数化设计方法的基础上，提出了一种基于子图的拟序列化设计和约束求解方法，该方法具有下列特点：(1)基于子图的方法使设计者既可用通常的点、线、圆弧等的基本图元，又可用高层的子图设计，利用子图编辑可加快设计过程；(2)结构约束优于尺寸约束的拟序列化设计方法，可实现变量化设计，提高设计效率；(3)基于标识的约束模型，使该方法可统一二、三维约束求解。该约束求解技术统一了二、三维约束，既可用于二维草图设计，也可用于三维特征编辑。

参考文献：

[1] Change-Xue Feng , Andrew Kusiak. Constrain-based design of parts. Computer Aided Design,1995.27(5):343~352

[2] Willian Bouma, Loannis Fudos, Christoph Hoffmann. Geometric constrain Solver. Computer Aided Design,1995.27(6):487~501

[3] 高小山,蒋鲲. 几何约束求解研究综述[J]. 计算机辅助设计与图形学学报,2004.4.

[4] 夏鸿建,王波兴,陈立平. 三维几何约束求解的变分算法[J]. 计算机辅助设计与图形学学报,2006.12.

[5] 石志良,陈涛,黄学良,陈立平. 几何约束求解可构造模式研究[J].工程图学学报,2008.2.

[6] 高曙明,彭群生.一种基于几何推理的参数化设计方法.计算机学报,1994.11.

[7] 萬建新,彭群生,董金祥,沈创.基于约束的形状自动求解新算法. 计算机学报,1995.18(2)