Hadoop文件系统性能分析

周轶男 王 宇
（江南计算技术研究所，江苏 无锡）

摘 要：随着网络技术的飞速发展，许多企业和团体通过数据存储服务提供商进行数据的存储、计算和交互。Hadoop是Apache基金会资助的开源项目，其文件系统HDFS具有的海量、高可扩展性、高可靠性、高性能等面向云计算领域应用的重要特征。文章通过在PC上安装HDFS，构建了分布式环境，通过实验证明了Hadoop文件系统在大数据量情况下的性能优势。

关键词：Hadoop；文件系统；云计算

Research on the Performance of Hadoop File System

Zhou Yinan Wang Yu
(JiangNan Institute of Computing Technology, Wuxi, Jiangsu)

Abstract: As the rapid development of Web technology, lots of companies and organizations deal with the storage, computation, and interaction of data through data storage service provider. As an open source project sponsored by Apache, Hadoop's file system which is named HDFS has many important features, such as great capacity, high extensibility, high reliability, and superior performance, which are oriented in cloud computing. This paper establishes a distributed environment through installing HDFS on PC. It is proved by experiment that, under the premise of massive data, Hadoop's file system has better performance.

Key words: Hadoop; file system; cloud computing

0 引言

Hadoop文件系统(HDFS)是一个运行在普通的硬件之上的分布式文件系统，它和现有的分布式文件系统有着很多的相似性，然而和其他分布式文件系统的区别也是很明显的，HDFS是高容错性的，可以部署在低成本的硬件之上，HDFS提供高吞吐量应对应用程序数据访问，它适合大数据集的应用程序，HDFS放开一些POSIX的需求去实现流式地访问文件数据，HDFS开始是为开源的apache项目nutch的基础结构而创建，HDFS是Hadoop项目的一部分，而Hadoop又是lucene的一部分。HDFS在设计的时候就考虑到平台的可移植性[1]。这种特性方便了HDFS作为大规模数据应用平台的推广。本文对Hadoop文件系统进行深入的研究，分析其在不同数据量下的性能。

1 分布式文件系统模型

分布式文件系统是分布式系统的关键技术之一，能够以文件的方式实现信息资源的共享。在云计算环境中，分布式文件系统承担着为用户提供文件服务的重任，它要保证用户在访问、保存在云中的文件时能够获得接近甚至在某些方面超出其在使用本地磁盘时的服务质量(包括性能、可靠性等)。

分布式文件系统通过网络为用户提供远程文件服务，它的设计目标是要使得用户感知不到其访问的是存储在远程服务器中的文件。因此，分布式文件系统的设计特别强调系统对用户的透明性。满足用户的透明性需求对于分布式文件系统设计非常关键，直接影响了用户对远程文件的访问体验。除此以外，还有其他一些设计需求，包括分布式文件系统需要具有高可用性，能够支持异构客户端的并发访问，能够提供文件数据的多个拷贝并保证文件数据的一致性和安全性等[2]。

针对这些需求，已经有数量众多的分布式文件系统被提出，它们在设计和实现上各具特点。为能够对这些文件系统进行分析和比较，提出了分布式文件系统的远程文件服务模型。该文件服务模型得到了学术界和产业界的广泛认同，主要由扁平文件(FlatFile)服务、目录服务和客户端模块3部分组成。其中，扁平文件服务实现对服务器磁盘上保存的文件内容的操作，负责创建、删除文件以及读写文件的内容和属性，为每个文件创建惟一的文件标识符供后续操作时引用；目录服务实现的是文件的文本名字与其对应标识符之间的映射，负责目录的创建、删除以及目录中文件的增删和查找，生成的目录也是以文件方式保存并由扁平文件服务负责管理；客户端模块则是运行在客户端上，负责封装对扁平文件服务和目录服务的访问，提供了从客户端本地文件系统的文件操作接口到远程服务器的相关功能调用的映射。扁平文件服务和目录服务向客户端模块提供高效、基于网络通信的调用接口，这些接口的功能能够组成完备的文件操作集合。客户端模块将这些操作接口进一步封装，再以与常规的本地文件操作接口接近的方式提供给应用程序，尽量减少远程文件存放对用户应用程序执行造成的影响，实现透明化。

2 Hadoop文件系统分析

2.1 Hadoop文件系统架构

HDFS采用master/slave架构。一个HDFS集群是有一个Namenode和一定数目的Datanode组成，如图1所示。Namenode是一个中心服务器，负责管理文件系统的namespace和客户端对文件的访问。Datanode在集群中一般是一个节点一个，负责管理节点上它们附带的存储。在内部，一个文件其实分成一个或多个block，这些block存储在Datanode集合里。Namenode执行文件系统的namespace操作，例如打开、关闭、重命名文件和目录，同时决定block到具体Datanode节点的映射。Datanode在Namenode的指挥下进行block的创建、删除和复制。Namenode和Datanode都是设计成可以跑在普通的、廉价的、运行Linux的机器上。HDFS采用java语言开发，因此可以部署在很大范围的机器上。一个典型的部署场景是一台机器跑一个单独的Namenode节点，集群中的其他机器各跑一个Datanode实例。这个架构并不排除一台机器上跑多个Datanode，不过这比较少见。

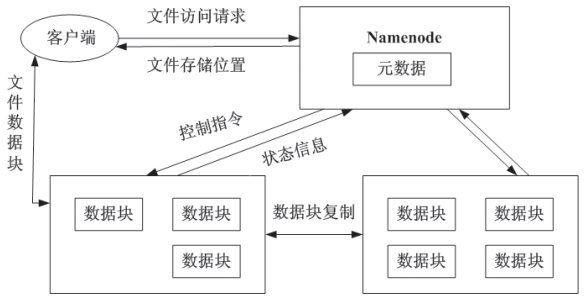

图1 HDFS体系架构

HDFS支持传统的层次型文件组织，与大多数其他文件系统类似，用户可以创建目录，并在其间创建、删除、移动和重命名文件。HDFS不支持userquotas和访问权限，也不支持链接(link)，不过当前的架构并不排除实现这些特性。Namenode维护文件系统的namespace，任何对文件系统namespace和文件属性的修改都将被Namenode记录下来。应用可以设置HDFS保存的文件的副本数目，文件副本的数目称为文件的replication因子，这个信息也是由Namenode保存。

2.2 数据复制

HDFS被设计成在一个大集群中可以跨机器地、可靠地存储海量的文件。它将每个文件存储成block序列，除了最后一个block，所有的block都是同样的大小。文件的所有block为了容错都会被复制。每个文件的block大小和replication因子都是可配置的。Replication因子可以在文件创建的时候配置，以后也可以改变。HDFS中的文件是write-one，并且严格要求在任何时候只有一个writer。Namenode全权管理block的复制，它周期性地从集群中的每个Datanode接收心跳包和一个Blockreport。心跳包的接收表示该Datanode节点正常工作，而Blockreport包括了该Datanode上所有的block组成的列表[3]。

副本的存放是HDFS可靠性和性能的关键。HDFS采用一种称为rack-aware的策略来改进数据的可靠性、有效性和网络带宽的利用。这个策略实现的短期目标是验证在生产环境下的表现，观察它的行为，构建测试和研究的基础，以便实现更先进的策略。庞大的HDFS实例一般运行在多个机架的计算机形成的集群上，不同机架间的两台机器的通讯需要通过交换机，显然通常情况下，同一个机架内的两个节点间的带宽会比不同机架间的两台机器的带宽大。

通过一个称为RackAwareness的过程，Namenode决定了每个Datanode所属的rackid。一个简单但没有优化的策略就是将副本存放在单独的机架上。这样可以防止整个机架（非副本存放）失效的情况，并且允许读数据的时候可以从多个机架读取。这个简单策略设置可以将副本分布在集群中，有利于组件失败情况下的负载均衡。但是，这个简单策略加大了写的代价，因为一个写操作需要传输block到多个机架。

在大多数情况下，replication因子是3，HDFS的存放策略是将一个副本存放在本地机架上的节点，一个副本放在同一机架上的另一个节点，最后一个副本放在不同机架上的一个节点。机架的错误远远比节点的错误少，这个策略不会影响到数据的可靠性和有效性。三分之一的副本在一个节点上，三分之二在一个机架上，其他保存在剩下的机架中，这一策略改进了写的性能。

为了降低整体的带宽消耗和读延时，HDFS会尽量让reader读最近的副本。如果在reader的同一个机架上有一个副本，那么就读该副本。如果一个HDFS集群跨越多个数据中心，那么reader也将首先尝试读本地数据中心的副本。

Namenode启动后会进入一个称为SafeMode的特殊状态，处在这个状态的Namenode是不会进行数据块的复制的。Namenode从所有的Datanode接收心跳包和Blockreport。Blockreport包括了某个Datanode所有的数据块列表。每个block都有指定的最小数目的副本。当Namenode检测确认某个Datanode的数据块副本的最小数目，那么该Datanode就会被认为是安全的；如果一定百分比（这个参数可配置）的数据块检测确认是安全的，那么Namenode将退出SafeMode状态，接下来它会确定还有哪些数据块的副本没有达到指定数目，并将这些block复制到其他Datanode。

2.3 数据完整性

从某个Datanode获取的数据块有可能是损坏的，这个损坏可能是由于Datanode的存储设备错误、网络错误或者软件bug造成的。HDFS客户端软件实现了HDFS文件内容的校验和。当某个客户端创建一个新的HDFS文件，会计算这个文件每个block的校验和，并作为一个单独的隐藏文件将这些校验和保存在同一个HDFSnamespace下。当客户端检索文件内容，它会确认从Datanode获取的数据跟相应的校验和文件中的校验和是否匹配，如果不匹配，客户端可以选择从其他Datanode获取该block的副本[4]。

3 实验与结果分析

3.1 设置节点和用户

准备5台机器，配置均为Core2处理器，2G内存，操作系统为Red Hat Linux。其中，一台作为Namenode，命名为master，两台作为dataNode，命名为slave01、slave02、slave03、slave04。在5台机器上都设置Hadoop用户。

设置Hadoop用户从master到slaves不需要密码。初始化namenode节点登录到namenode上，cd/data/hadoop/install/bin，然后格式化Image文件的存储空间：

./hadoopnamenode-format，保证Hadoop文件系统各个节点上配置文件的一致性。

3.2 结论分析

由于HDFS采用分布式并行处理的方式，在Namenode的硬件资源消耗没有达到瓶颈的情况下，HDFS的访问性能明显高于单机操作的性能。

在数据量比较小时，由于Namenode和Datanode之间的控制和任务调度占用了部分资源，所以HDFS的性能体现不明显。在文件数量比较多时，网络通信带来的损耗可以忽略不计，HDFS的性能优势可以充分地体现出来。

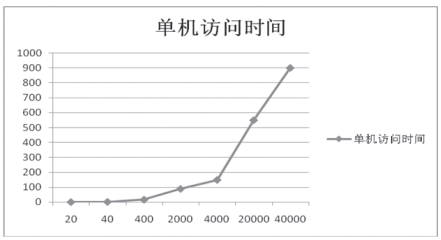

图2 单机访问时间

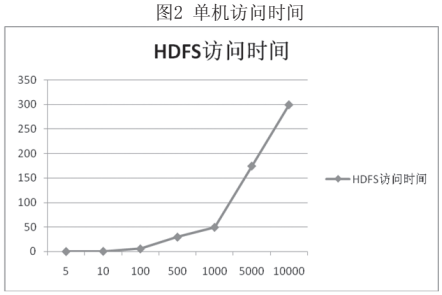

图3 HDFS访问时间

（下接11页）

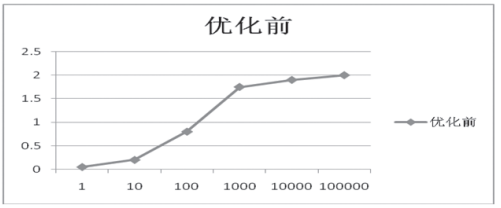

图1 GPU对AES算法的加速效果

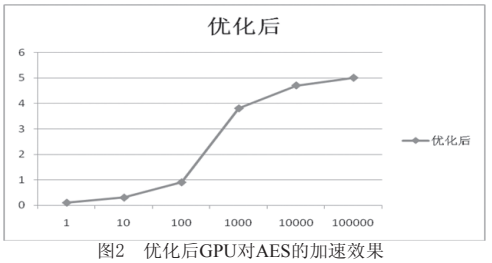

图2 优化后GPU对AES的加速效果

5 结论

本文介绍了在GPU上实现AES加密算法的方法。首先介绍了AES算法，然后对CUDA中的GPU结构和CUDA编程模型进行了深入的研究。最后在GPU和CPU平台上对设计进行了实验对比，取得了理想的加速效果。其实在大多数应用情况下，目前计算机显卡配置的GPU 运算潜能并没有完全释放出来，本文介绍的加密方法是GPU 通用计算具体应用的一个体现。虽然目前以CUDA 为代表的GPU仍然存在精度不高，程序编写限制较多的缺点，但随着并行流处理概念的进一步发展，GPU通用计算技术将在各个领域发挥更大的作用。

参考文献：

[1] 徐少平, 文喜, 肖建, 等. 一种基于Cg语言在图形处理器GPU上实现加密的方法[J]. 计算机应用与软件, 2008, 25(4)： 260-262.
[2] 吴亚联, 段斌. AES密码计算构建的设计及应用[J]. 计算机工程, 2005, 31(21):181-186.
[3] 曹华平, 罗守山, 温巧燕, 等. AES算法轮密钥与种子密钥之间的关系研究[J]. 北京邮电大学学报， 2002, 25(4): 47-50.
[4] Yeom Y J, Cho Y K,Yung M T. High-speed implementations of block cipher ARIA using graphics processing units[C] //International Conference on Multimedia and Ubiquitous Engineering(IEEE), 2008:271-275.
[5] NVIDIA. NVIDIA CUDA Programming Guide ver. 2.1.2008[EB/OL]. http://wenku.baidu.com/view/0ea27035eefdc8d376ee3282.html.
[6] Belenko A. Faster Password Recovery with Modern GPUs. TROOPERS 08,2008[EB/OL]. http://www.elcomsoft.com/presentations/faster_password_recovery_with_modern_GPUs.pdf

作者简介：
商 凯，江南计算技术研究所
电话：13665133105
电子信箱：mengzhimin915@sohu.com
通信地址：江苏无锡33信箱342号（214083）

（上接16页）

4 结论

通过对分布式文件系统和Hadoop文件系统在模型和体系架构方面的分析，可以发现，HDFS在设计和实现时都充分考虑了海量数据在存储和管理等方面的需求，并最终获得了系统的高可扩展性、高可靠性和高性能，满足了云计算领域相关用户的需求。最后通过实验证明了HDFS在大数据量、可并行化数据处理上的性能优势。

参考文献：

[1] Hadoop community. Hadoop distributed file system，2010[DB/OL]. http://hadoop.apache.org/hdfs
[2] George C, Jean D, Tim K. Distributed systems：Concepts and design(3rd Edition)[M]. Addison-Wesley Publishers Limited，2000.
[3] Wang F, Qiu J, Yang J, et al. Hadoop high availability through metadata replication[C]. Proceeding of the First International Workshop on Cloud Data Management，Hong Kong，China，November 2009.
[4] Gluster community. Gluster file system[DB/OL]. http://www.gluster.org，2010

作者简介：
周轶男，江南计算技术研究所
电话：13665133105
电子信箱：mengzhimin915@sohu.com
通信地址：江苏无锡33信箱342号（214083）