云计算异构环境下Hadoop性能分析

刘丹丹[1,2] 陈 俊[3] 梁 锋[4] 范小朋[1]

[1]（ 中国科学院深圳先进技术研究院先进计算与数字工程研究所 深圳 518055 ）
[2]（ 长春理工大学电子信息工程学院 长春 130022 ）
[3]（ 中国科学技术大学软件学院 合肥 230027 ）
[4]（ 南京大学软件学院 南京 210093 ）

摘 要 通过将虚拟化技术引入到传统的数据中心来实现计算资源的按需分配，云计算服务正获得日益广泛的应用，例如亚马逊所提供的弹性云计算服务EC2。另一方面，Hadoop作为MapReduce这一大规模数据的分布式并行计算模型的开源实现，在学术界和工业界都获得了越来越多的研究和应用。当前的一个研究热点问题就是如何将云平台这一异构化的底层基础设施，与Hadoop的上层计算模型有效结合起来，利用云平台所提供的弹性资源来充分发挥Hadoop高扩展性、高容错性、低硬件配置的优点。在这篇论文中，我们在异构云平台环境下进行了一系列的Hadoop性能测试和分析，并指出在这一环境下，由于虚拟机的高IO开销，导致Hadoop的性能相比传统的纯粹物理节点集群急剧降低。我们的工作可以作为研究云计算异构环境下如何提高Hadoop性能的一个重要基础。

关键词 Hadoop；云计算；异构性；系统性能

A Performance Analysis for Hadoop under Heterogeneous Cloud Computing Environments

LIU Dan-dan[1,2] CHEN Jun[3] LIANG Feng[4] FAN Xiao-peng[1]

[1](*Institute of Advanced Computing and Digital Engineering, Shenzhen Institutes of Advanced Technology, Chinese Academy of Science, Shenzhen* 518055, *China*)
[2](*School of Electronics and Information Engineering, Changchun University of Science and Technology, Changchun* 130022, *China*)
[3](*School of Software Engineering, University of Science and Technology of China, Hefei* 230027, *China*)
[4](*ISchool of Software Engineering, Nanjing University, Nanjing* 210093, *China*)

Abstract Cloud computing grows rapidly nowadays, which brings virtualization technology to traditional datacenters in order to implement service-on-demand of computing resources, such as Amazon's Elastic Cloud Computing (EC2) Services. Hadoop is an open-source implementation of Google's MapReduce, which is a distributed parallel computing model for large-scale dataset. Hadoop is gaining more and more focuses both in academy and industry. It is an open question that how to combine cloud computing infrastructures with Hadoop efficiently, i.e., making full use of the former's elastic resources and the latter's advantages of scalability, fault-tolerance and running on commodity hardware. In this paper, we carry out a series of experiments to evaluate and analyze the performance of Hadoop on our heterogeneous clouding computing testbed. We demonstrate that the performance of Hadoop is degraded under the scenario with high I/O overheads, compared with the traditional scenario where each node in a cluster is a physical machine. Our work can act as a basis for improving the performance of Hadoop under the cloud computing environments.

Keywords Hadoop; cloud computing; heterogeneity; system performance

---

基金项目：国家自然科学基金项目“无线移动网络环境中基于数据访问分布的合作缓存研究”（61202416）；国家科技重大专项“传感器网络关键技术研究（传输与组网）”(2009ZX03006-001)；广东省重大科技专项“中高速传感器网络关键技术”(2009A080207002)。

作者简介：刘丹丹，硕士研究生，研究方向为移动云计算、分布式计算、信号处理；陈俊，硕士，研究方向为分布式计算、虚拟化与云计算；梁锋，学士，研究方向为分布式计算、云计算；范小朋，博士，助理研究员，研究方向为云计算、移动计算、无线通信、软件工程，E-mail: ddl211@yeah.net。

1 引　言

传统的基于物理节点搭建而成的数据中心正在经历着一场变革—诞生于上世纪60年代并被最初应用到大型机上的虚拟化技术，被越来越广泛地部署到基于X86架构的、由通用廉价硬件组成的PC服务器上。不同于传统的物理机器，在虚拟机中所有的上层应用和中间层操作系统并不能直接与底层硬件交互，而需要经由hypervisor或VMM层才能接触到底层硬件。虚拟化技术提供了众多特有的优势，例如：（1）资源复用——在单一的硬件资源集合上同时实现多个虚拟机容器，且容器之间相互性能隔离；（2）迁移技术——虚拟机容器可以从所在的节点上无缝迁移到另一个节点上，并保证在整个迁移过程中容器中的应用程序仍然可以正常工作[1, 2]。在部署了虚拟化技术后，整个数据中心可以看成是一个巨大的资源池，用户可以按需定制和动态调整所需要的资源并且以虚拟机的形式呈现，而不是传统的以物理机节点这一固定的粗粒度来占用资源。

在此基础上，当一个数据中心引入虚拟化技术并接入互联网后，它就可以向外部用户提供资源租赁服务，即公共云服务。在2006年，世界上最大的在线商店亚马逊公司推出了一款弹性云计算服务（Elastic Cloud Computing）的平台，以虚拟机的形式向用户出租其数据中心的计算资源。用户可以按需选择不同的资源配置，按使用时长支付租金。用户通过因特网将个人数据或应用上传到亚马逊的云平台，并得到处理后的结果。

除了公有云平台外，基于对数据的隐私性、可靠性方面的考虑，云服务的另一种形式是将数据和应用保存在企业或机构的数据中心内部，而不对外界提供服务。这种服务模式称之为私有云服务。这一领域内的代表软件包括Eucalyptus、OpenStack、AbiCloud等。出于兼容性和拓展性方面的考虑，Eucalyptus等软件在实现了私有云平台之外，还提供了与EC2等公共云平台相兼容的接口，使得用户可以根据具体应用充分利用这两种平台。

云计算服务的兴起和广泛应用，必然会造成后台数据中心中数据规模量的急剧膨胀。如何高效可靠地存储和处理海量规模的数据，成为了云服务进一步发展所面临的关键问题。另一方面，MapReduce[3]作为一种处理海量规模数据集的分布式并行计算模型，因其高扩展性、高容错性、低硬件配置的优点，其开源实现版本Hadoop在学术界和工业界都获得了越来越多的研究和应用。因此，如何将MapReduce加入到云计算平台中，充分发挥其处理数据密集型应用的优势，是当前的一个研究热点。但是，由于虚拟化技术带来的高IO开销，云环境下的Hadoop相比传统的纯粹物理节点环境，会存在一定的性能差距。

在本文中，我们试图通过定量分析的方法对Hadoop在虚拟机和物理机的混合的环境下数据的读写性能进行分析，因此我们提供了一个包含了物理机节点PM和虚拟机节点VM的异构云平台，测试了集群中这两类节点在磁盘读写速度、Hadoop benchmarks等方面的性能，根据实验结果定量的分析这两类节点在Hadoop集群中的性能差异，并指出我们接下来将要进行的工作。

论文后续内容是如下组织的：第二部分介绍MapReduce的背景知识和相关工作，第三部分介绍云平台的设计和实现，第四部分介绍实验过程和结果并作分析，第五部分给出我们的结论并指出接下来的工作。

2 背　景

在这一章，首先我们会简单介绍MapReduce框架及其开源实现版本Hadoop，然后调研目前已有在云平台下Hadoop的实现方案和研究工作。

2.1 MapReduce及其开源实现Hadoop

MapReduce是由Google所提出的、用于数据密集型计算的分布式并行计算模型，在该框架下包含了两种函数：

Map函数：HDFS文件系统中的每个文件块对应一个map函数实例化的对象来处理，输出结果为一系列的中间键值对的集合，即：

$$map(key1, value1) \rightarrow list(key2, value2) \quad (1)$$

Reduce函数：有着相同键的每个中间键值对的集合由reduce函数的一个实例化对象合并后进行处理，汇总结果写入到HDFS文件系统，即：

$$reduce(key2, list(value2)) \rightarrow list(value2) \quad (2)$$

对MapReduce的使用者来说，无需由自己来设计和实现应用程序在集群中的分布和并行计算；只需要编写简单的Map和Reduce函数，MapReduce框架就会自动实现计算过程的并行化，并保持计算过程中数据的容错性、一致性等要求。MapReduce的计算过程如图1所示。

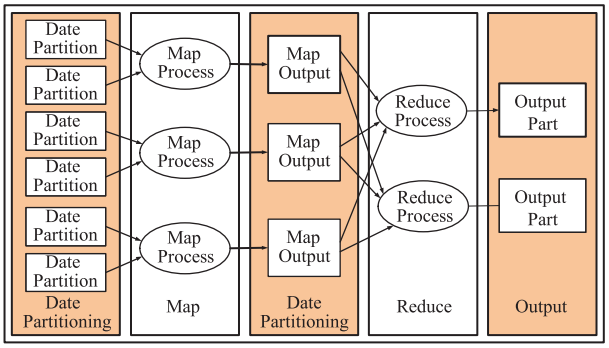

图1 MapReduce数据处理流程图

Hadoop是MapReduce框架的一个开源实现，由Java语言编写，目前是Apache软件基金会下属的一级项目。Hadoop包含三个子项目：Common、HDFS和MapReduce。其中，Common是一系列的工具集，包括文件系统、RPC、序列化库函数等；HDFS是一种分布式的文件系统，以数据流的方法读写大规模数据集，数据集按指定的大小被分割成众多的文件块，且每个文件块具有多个副本，分布在集群的多个节点中；Hadoop MapReduce是对MapReduce框架的具体实现，处理大规模数据集的MapReduce作业会被分割成众多的Map和Reduce任务，分布在整个集群并行执行。

2.2 异构云计算环境

随着云计算概念的兴起及其应用的日益广泛，用户可以通过互联网从远程的云端租赁一定数量的虚拟机节点用于执行自己的计算任务，使用完成后再返还给云端，从而省去了一次性购入大量设备的成本以及后续的管理和维护成本。用户可以指定租赁的虚拟机数目，并且弹性定制每个虚拟机节点所具备的硬件资源。例如，亚马逊EC2服务提供了small instance、medium instance、high-CPU instance等多种虚拟机类型，每一种都有不同的CPU、内存等资源的配额。另一方面，企业也可以把已有的传统的数据中心与流行的云计算平台结合，协同完成作业。在实践中，我们发现传统数据中心中的物理机节点相比于云平台中的虚拟机节点，以及不同资源配额的虚拟机节点之间，必然存在着结构和性能上的差异性，即异构性。在虚拟化和云计算带来的异构性环境中，分析并提高各种应用的性能，具有很强的研究意义和现实意义。目前，在网格计算、高性能计算等领域，已有大量基于虚拟机展开的研究工作[4-6]。在计算模型方面，MapReduce领域也已有了相关的研究，例如虚拟机和云环境中的Hadoop性能测试[7-9]、异构环境下的MapReduce性能提高[10]、调整虚拟机环境下的MapReduce系统框架以优化性能[11]等工作。

3 实验异构平台设计

在这一章节中，首先我们会介绍我们搭建的小型云平台的基本情况，包括硬件配置和操作系统、虚拟化软件，以及如何通过将虚拟机引入到传统的由物理节点构成的集群中来实现异构性。接下来，我们会介绍所选用的性能测试指标benchmarks，在真实的异构环境中，我们不仅使用了linux下的测试命令，也使用了Hadoop环境下的一系列用于性能测试的作业。

3.1 云平台

我们的实验平台由四台PC组成，包括一台1U尺寸的Dell R410抽屉式服务器，以及三台塔式组装机。服务器端装备有一颗四核2.13G主频的Intel Xeon E5506 CPU，8G内存以及三块300G的SAS硬盘；每台组装机装备有一颗四核2.8G主频的Intel i5 760 CPU，8G内存，一块80G的Intel SSD硬盘和四块2T容量的Seagate 5900转SATA硬盘。四台机器之间由一台TP-LINK 24口千兆交换机连接构成一个小型的以太网络。

在软件选择上，我们选用Xen 3.0版本的完全虚拟化方案作为我们的虚拟化软件，在三台组装机的其中一台上搭建出一个虚拟机VM，并给这台虚拟机配置4个VCPU，7G内存和500G硬盘空间，从而使VM具有与其他物理节点相近的底层计算和IO资源。所有的物理节点和VM运行的都是2.6.18内核的Centos 5.5 64bits操作系统，Hadoop版本选用0.20.2，HDFS中文件块大小为64MB，块的副本数为3。Dell服务器作为Hadoop架构中的master节点，两台物理组装机和虚拟机VM作为slaves节点。

3.2 实验设计

由于虚拟化带来的高IO开销，因此首先我们需要测试Hadoop集群中物理机节点和虚拟机节点各自的磁盘读写性能，以便后面实现性能对照。我们使用linux下的dd和hdparm命令分别测试各节点的写性能和读性能。对dd命令，我们设置每次写入的数据大小为32M（在HDFS文件系统中，文件块的大小一般设置为32M或64M），并连续写入一百次，从而总的写入数据量大小为3.2G。这里需要注意的是，dd命令需要附加参数conv=fdatasync，以保证最终的数据被写入到磁盘而不是写入内存缓存即返回。对每个节点，我们连续测试6次写速度。

为了测试磁盘真实的读性能，我们采用hdparm

-t命令（dd命令测得的是从内存缓存读数据的速度，不是真实的磁盘读速度）。对每个节点连续测试多次，以最终得到一系列稳定的结果。

此外在Hadoop平台上，我们用采用Hadoop集群下的TestDFSIO和Sort两个典型的benchmark来测试整个平台的性能。对于TestDFSIO benchmark，它通过MapReduce中单个作业的方法来测试集群HDFS文件系统中多个文件并发读或写的性能；每个文件的读写以一个单独的map任务实现，并由一个reduce函数完成数据的汇总工作。在实验中，我们设置每次读写文件的大小都为1000MB，测试并发文件数即map数分别为3、6、8时的集群HDFS IO性能。对于sort benchmark，它将HDFS文件系统中指定的二进制数据集通过Map和Reduce函数进行排序，并将排序后的数据集写入到HDFS文件系统；sort代表了Hadoop中的一种典型应用，即输入和输出数据集中的记录内容和大小不变，但每条记录之间的位置经过洗牌过程后按一定的顺序重新排列。在实验中，我们先通过Hadoop下的randomWriter作业分别生成27G和2G两种大小的随机二进制文件，作为sort的输入。首先我们对27G数据集执行一次sort benchmark，记录物理机节点PMs和虚拟机节点VMs在map和reduce阶段各自的处理时间；接着，对2G的输入文件，我们连续执行三次sort benchmark得到统计上的分布。

4　实验结果

在这一节中，我们用一系列的benchmarks来衡量搭建的云平台的性能，并根据测试结果分析虚拟机所带来的异构性对整个集群性能的影响。

4.1　磁盘读写性能

Hadoop对数据密集型应用的处理性能，很大程度上取决于底层各个节点的数据读写速度。因此，在测试Hadoop集群性能之前，我们先测试各个节点的读写性能。

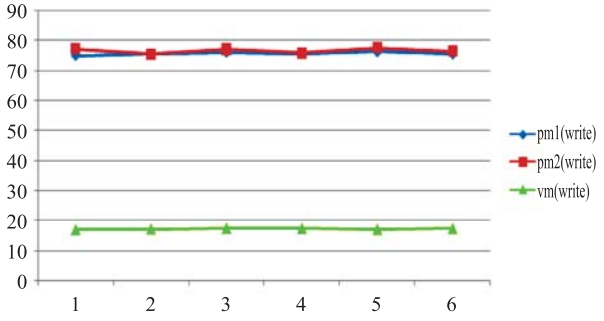

图2　物理机节点PMs与虚拟机节点VM的磁盘写速度（单位：MB/s）

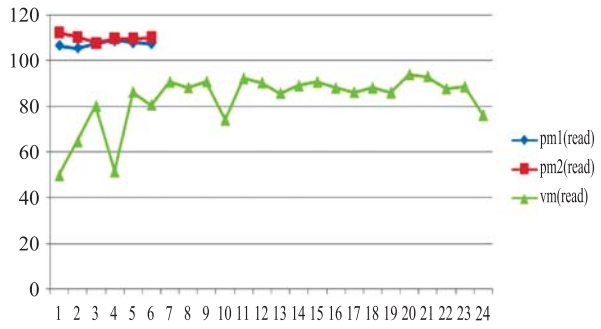

图3　物理机节点PMs与虚拟机节点VM的磁盘读速度（单位：MB/s）

在写速度方面，由图2可以看出，虚拟机节点VM的性能要远差于物理机节点PM1和PM2，只有前者速度的23%左右。而在读速度方面，由图3可以看出，VM在经过一段时间的热身后，读速度稳定在80MB/s左右，约为PM1和PM2读速度的75%。在Xen的架构中，用户域DominU的所有IO请求都必须经由Domin0才能与底层的实际IO设备交互，由此产生的响应延迟使得虚拟机的IO性能要差于物理机；并且根据我们的实验结果，写性能的开销要远远大于读性能的开销。写性能上PM和VM的巨大差距，会导致在Hadoop应用中，数据越被频繁地写入硬盘，VM节点的执行时间相比PM节点的执行时间差距越大。

4.2　HDFS文件系统吞吐量

吞吐量是衡量整个集群的总体IO性能的关键指

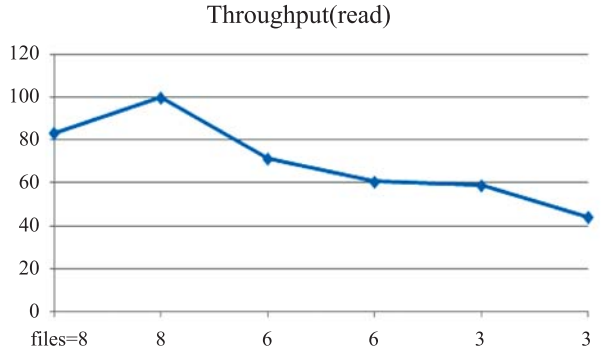

图4　Hadoop集群的写吞吐量（单位：MB/s）

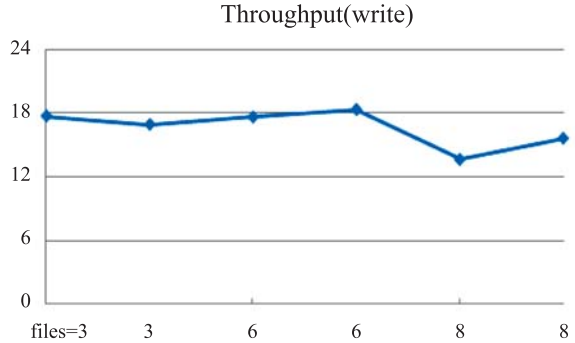

图5　Hadoop集群的读吞吐量（单位：MB/s）

标。MapReduce大规模数据量的应用场景，要求其底层文件系统必须能够达到良好的吞吐量，才能发挥其并行计算的优势。

在实际实验中，由于每个节点初始分配的写数据量是一样的，且在整个TestDFS写数据的作业执行中没有任务被成功地备份执行，所以总数据量的完成时间取决于集群中写速度最慢的节点的完成时间。根据上一节的磁盘读写性能测试，我们知道虚拟机节点VM的写性能最差，约为17MB/s。所以整个集群的吞吐量也应该取决于VM节点的写速度。由图4可以看出，测得的集群写吞吐量在13~18MB/s之间，从而验证了我们的想法。

类似的，集群的读吞吐量也取决于集群中最后一个执行完写任务的节点的读速度。根据上一节的磁盘读写性能测试结果，物理机节点PMs和虚拟机节点VM的读速度平均约为110MB/s和80MB/s。由图5可以看出，集群的读吞吐量均值约为70MB/s之间，与磁盘读性能结果基本吻合。需要注意的是，在测试读吞吐量时，数据的波动范围很大，测得的最大值为100MB/s，最小值仅为60MB/s。因为TestDFS读数据的作业中每个节点执行任务所需的时间相对写数据任务要快很多，因此慢速节点上的任务更容易由快速节点成功地备份执行；当成功备份执行的任务数越多时，测得的吞吐量就越大，反之测得值越小，上限为物理机节点PMs的写速度，下限为虚拟机节点PM的读速度。另外，并发任务数越多，备份执行成功这一事件的概率也越大，所以在任务数为8时测得的吞吐量最大，任务数为3时测得的吞吐量最小。

4.3 Sort benchmark

在本实验中，我们对随机产生的数据集进行排序并记录花费的时间，这也是衡量高性能计算、大规模分布式系统的一个重要的测试指标。我们首先在集群中运行一次27G数据量的sort benchmark，观察到物理机节点PMs和虚拟机节点VM在reduce阶段存在巨大的执行时间差异，如表1所示。

表1 27G数据集下sort benchmark中reduce任务各阶段执行时间分布

Task Id	Node	Shuffle phase	Sort phase	Reducer phase	Total time
reducer_00	b5	25mins, 20s	3mins, 37s	46min, 29s	1hrs, 15mins, 26s
reducer_01	Test	25mins, 50s	17mins, 1s	46min, 43s	1hrs,29mins, 34s
reducer_02	b6	24mins, 47s	4mins, 1s	44min, 44s	1hrs, 15mins, 24s
reducer_03	b5	24mins, 22s	3mins, 25s	45min, 45s	1hrs, 13mins, 32s
reducer_04	b6	24mins, 47s	4mins, 7s	46min, 27s	1hrs, 15mins, 21s

可以看出，PMs和VMs在shuffle和reducer阶段执行时间没有明显的差异；但在sort阶段，VM花费的时间约为PMs的4.2~4.8倍，且sort阶段占据了整个reduce执行时间的20%。传统的Hadoop集群中作业各个阶段的执行时间如图6所示。

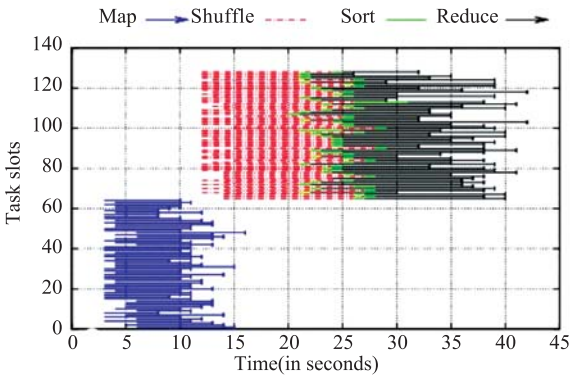

图6 传统Hadoop集群作业各阶段执行时间分布图

根据我们的实验，集群中物理机节点PMs上reduce执行时间在shuffle、sort和reducer三个阶段的分布比例约为7:1:14，sort阶段所占的比例很小（4%~5%），与图6的分布图基本吻合。而对虚拟机节点VM，三阶段的分布比例约为3:2:6，sort阶段占据了18%的时间，与图6的分布图差异明显。这样的原因是，在sort阶段，输入数据是混合存储在内存和磁盘区域。当数据量小时，可以被完全存储在内存缓存区中，从而整个sort排序过程是在内存中进行的（称之为内排序）而不会产生磁盘IO；当数据量大到内存缓存区无法完全存储时，多余的数据集会被存储到磁盘中进行外部排序（即以分治的方式将数据逐段地从磁盘读入内存缓存区，输出结果再写入到磁盘），从而产生了多次的磁盘往返IO。由于虚拟机节点VM写速度远低于物理机节点PMs，从而VM在sort阶段所耗费的时间要远大于PMs。

为了验证这一推测，我们又执行了2G数据量的sort benchmark，并重复执行三次，得到的统计结果如表2所示。

表2 2G数据集下sort benchmark中reduce任务各阶段执行时间分布

Task Id	Node	Shuffle phase	Sort phase	Reducer phase	Total time
reducer_00	b5	25mins, 20s	3mins, 37s	46min, 29s	1hrs, 15mins, 26s
reducer_01	Test	25mins, 50s	17mins, 1s	46min, 43s	1hrs,29mins, 34s
reducer_02	b6	24mins, 47s	4mins, 1s	44min, 44s	1hrs, 15mins, 24s
reducer_03	b5	24mins, 22s	3mins, 25s	45min, 45s	1hrs, 13mins, 32s
reducer_04	b6	24mins, 47s	4mins, 7s	46min, 27s	1hrs, 15mins, 21s

在sort阶段，由于输入数据量小，每个节点平均只有700M的工作负载，整个过程完全在memory中进行内排序而不需要在硬盘中外排序，从而不产生磁盘IO，故PMs和VM执行时间几乎没有差别（都是显示0秒处理时间）。

5 结束语

随着云计算服务的日益流行和MapReduce在处理数据密集型应用领域的广泛使用，如何在云平台上实现MapReduce架构来处理云平台中的大规模数据已成为当前的一个研究热点。由于通常情况下云平台都是基于虚拟化技术来封装底层资源，而虚拟机在IO上固有的高开销和低性能，使得MapReduce在云平台上的实现，相比于传统的基于物理机节点搭建的集群，存在严重的性能下降。在这本文中，我们实现了一个由物理机和虚拟机节点混合组成的异构云平台，并在这一平台上进行了一系列的基于Hadoop的性能测试，包括各节点各自的读写性能、HDFS文件系统的吞吐量，以及基于MapReduce的sort benchmark；并分析了由于虚拟机的引入，导致在性能测试中出现的多种瓶颈。我们的工作可以作为研究云计算环境与Hadoop性能的一个重要基础，为基于Hadoop平台研究者提供数据处理方面的参考。

参 考 文 献

[1] Clark C, Fraser K, Hand S, et al. Live migration of virtual machines [C] // Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation, Boston, USA. New York: ACM Press, 2005: 273-286.

[2] Zhao M, Renato J. Figueiredo. Experimental study of virtual machine migration in support of reservation of cluster resources [C] // Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing, Reno, USA. New York: ACM Press, 2007: 1-8.

[3] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters [M]. Communications of the ACM-50th Anniversary Issue: 1958-2008, 2008, 51(1): 107-113.

[4] Figueiredo R J, Dinda P A, Fortes J A B. A case for grid computing on virtual machines [C] // Proceedings of the 23rd International Conference on Distributed Computing Systems, Gainesville, FL, USA. New York: IEEE Press, 2003: 550-559.

[5] Mergen M F, Uhlig V, Krieger O, at el. Virtualization for high-performance computing [J]. The ACM Special Interest Group on Operating Systems Operating Systems Review, 2006, 40(2): 8-11.

[6] Huang W, Liu J X, Abali B, et al. A case for high performance computing with virtual machines [C] // Proceedings of the 20th Annual International Conference on Supercomputing, Cairns, Queensland, Australia. New York: ACM Press, 2006: 125-134.

[7] Ibrahim S, Hai Jin, Lu Lu, at el. Evaluating mapreduce on virtual machines: the hadoop case [M]. CloudCom 2009, Lecture Notes in Computer Science 5931, 2009: 519-528.

[8] Horacio G-V, Kontagora M. Performance evaluation of mapreduce using full virtualisation on a departmental cloud [J]. International Journal of Applied Mathematics and Computer Science, 2011, 21(2): 275-284.

[9] Shafer J. I/O virtualization bottlenecks in cloud computing today [C] // Proceedings of the 2nd Conference on I/O Virtualization, Pittsburgh, USA, 2010: 5-5.

[10] Zaharia M, Konwinski A, Joseph A D, at el. Improving mapreduce performance in heterogeneous environments [C] // Proceedings of the 8th Usenix Conference on Operating Systems Design and Implementation, San Diego, USA. New York: ACM Press, 2008: 29-42.

[11] Ibrahim S, Jin H, Cheng B, at el. Cloudlet: towards mapreduce implementation on virtual machines [C] // Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing, Garching, Germany. New York: ACM Press, 65-66.