HBase的 - 非SQL数据库，业绩评价

1 多林Carstoiu， *2* 埃琳娜Lepadatu， *3* 米哈伊·加斯帕
1" *布加勒斯特的布加勒斯特"大学dorin.carstoiu@yahoo.com*
2,3" *布加勒斯特的布加勒斯特"大学lepadatu.elena@gmail.com，*

gaspar.mihai@yahoo.com
DOI：10.4156 / ijact.vol2。issue5.4

抽象

HBase的是BigTable的开源版本 - 由谷歌对大量结构化数据的管理开发的分布式存储系统。HBase的模拟最受BigTable的提供的功能的。像大多数非SQL数据库系统，HBase的是用Java编写。当前工作的目的是评估与SQL数据库进行比较HBase的实施的表演，当然，与BigTable的提供的表演。测试的目的是评估有关随机写入和行，顺序写入和顺序读出，他们如何受到影响，通过增加柱的家庭数量，并使用MapReduce函数的随机读取性能。

关键词： *HBase的，BigTable的，MapReduce的，Hadoop的，数据管理部，NameNode会*

1.简介

在一般的方法，数据库是存储在外部媒体数据的方法。通常，数据库存储在一个或多个文件，具有信息检索出它们的可能性。使用数据库的应用程序通常被设计的方式，通常是非常有效的大型信息管理的复杂环境中（数据处理）（也许密集型？）。这也是采用数据库存储大规模应用的主要目标。在数据库中组织数据是集中的数据库管理员监视下一个形式，具有以下优点：减少存储的数据的冗余，节省数据以避免出现不一致，数据共享，气势保持数据的完整性和存储标准化的安全限制的可能性。作为一个总体思路便于使用的是主要标准。

虽然有几种方法在数据库中组织数据，关系数据库是最有效的方法是使用一组数学理论来组织数据之一。一个典型的数据库模型是关系之一，根据其中数据被存储在表中。一个表是含有一组项目，具有定义的一组对应于相应的表中的列的属性的每个项目的数据结构。此外表中，数据库可以包括：存储过程，视图，用户和用户组，类型的数据对象，并且还作用[11]。

集中式数据库的主要目标是：统一控制，数据冗余，效率和最小化，最后但并非最不重要，完整性和数据安全。由于其提高可用性，可靠性和灵活性，而且由于在数据库技术和通信，进度简单的和/或分布式数据库系统现在已很普遍。

作为一个数据库逻辑整合，但在几台机器，可以通过网络基础设施进行通信的物理分布式分布式数据库可以被定义。

关系型数据库的主要优点是数据的可获得性和易用性在获取必要的信息。这是高度与所选择的SQL语言也是它的正确使用相关
当情况需要大量的数据集，关系数据库失去他们的权力。因此分布式关系数据库是由非SQL数据库的版本越来越多的取代。一个不同的观点维持一个事实，即建立一个分布式数据库所需的必要的基础设施是非常昂贵和可扩展性是很难实现的。在这种特定情况下完整地保留关系数据库的权性质是非常难以得到（如索引，完整性，外国和主键），因此不能令人满意。在另一方面他们在整个世界广为扩大，由于其在审讯容易。

许多现代应用程序包括数据库服务器，回答来自许多客户端访问多个Web服务器的请求。在这种情况下，人们可以经常发现演出是低于预期

那些。在这种情况下，许多考虑升级硬件，没有考虑到数据库服务器。如果我们把Amazon.com的它运行Oracle数据库，优化和扩展的例子，我们可以考虑的是，SQL技术已经达到可扩展性的最大点。

目前，有后台数据库系统和大量的在他们顶部的应用程序，众说纷纭。这些应用的诸多瓶颈是由于SQL组件，它在适合80年代的电脑，但不再以目前的架构非常复杂的方式进行非常简单的任务。主要的大公司开发的基于SQL的数据库管理系统很大程度上依赖于硬件，以确保所需的性能。溶液可以被分配在多个机器上的软件，在这种情况下，许可成本变得高昂。这就需要一种新的方法，其中在性能有很大的提高，需要微不足道的成本，并提供了良好的扩展性。

这是正常的考虑其他方法。这样一个很好的例子是谷歌的做法，使用BigTable的是半结构化的数据库，这使来自互联网缓存[3]大多数信息。这两种方法的比较，得出的结论是传统的SQL数据库系统，如Oracle，DB2和其他实现，不适合某一类的应用程序。类似的方法，以大表，从谷歌，围绕80年代引入了通过所谓的“分层文件系统”操作系统。

非SQL数据库开始更认真地在2009年初的时候，他们提出，可以在不需要出现在关系数据库管理系统的关系功能的系统中使用分布式数据库的解决方案体现。需要对他们来说，如前面提到的，出现了为容易获得的可扩展性的解决方案。BigTable的开创了市场对于非SQL数据库，通过提供大量的可扩展性和批量访问大数据集信息高吞吐量在2004年初开始。它是建立在谷歌文件系统。BigTable的，具有以下的开源项目如卡桑德拉，Hypertable的，HBase的沿，手摇共享几个特点：密钥值存储，批次分析，在大量的机器分配和共享的数据。这些另一个重要特点是，为了获得可扩展性的理想水平，

2.相关工作

有几个原因证明的基于键值对[1]的存储解决方案的选择。他们之中有一些是：

- 许多RDBMS并不保证体面复制并收购了强大的RDBMS导致许可费用过高;
- 有必要存储大量半结构化数据的;
- 这是为了对付新的语言，如二郎神的借口;
- 数据存储和访问次数最多基于主密钥;
- 复杂的联接操作都没有处理数据的必要;
- 数据量非常大，所造成的复制错误情况的管理提出了这个问题变得非常难以处理;

例如Facebook使用草堆，因而存储大量数据的一个文件中具有独立的索引，需要元数据的1M用于数据的1G [2]。一些项目已经发展成为一个替代RDBMS，其中一些超过一个key-value存储。对于每个样品，有许多的主要特征定义：

- 实现语言 - Java进行Voldermort，卡珊德拉，HBase的，对于二郎林戈，楷，Scalaris，Dynomite，C（C ++）对于Hypertable的，ThruDB，MemcacheDB;
- 数据模型：多斑点，面向文档或BigTable的;
- 主要基于复制和分区容错性。

其中一些提供分布式基于与复制功能键 - 值对存储设施。一个重要问题是与数据供应来填充动态网页，特别是对于Web应用程序的等待时间。延迟取决于环境和缓存所需数据的存在。一般情况下，我们预计不超过10ms更多的数据是可用的，需要以其他方式成本分析，以提高性能。一些最流行的非SQL数据库实现的是：

1.谷歌的Bigtable [3]是设计成用于管理结构化数据的分布式存储系统
高度可扩展的。该系统已经证明了其效率从谷歌重要的应用。作为

例如，在这里我们可以提到谷歌分析，谷歌地球，谷歌财经。BigTable的提供了一个简单的数据模型的客户使用行，列和时间戳索引。从视图的BigTable的数据模型点是稀疏的，分布式的持久性，多维有序映射，其中在图中的每个值是一个字节的不可解释数组。作为非SQL数据库的典型特征列在集称为列族其通常包含相同类型的信息分组。时间戳被引入由于这样的事实，每个单元格可以包含相同的数据的多个版本。

2.非SQL数据库的另一个例子是卡桑德拉。该系统的开发开始于
Facebook的前几年。目前，该项目是开源的，并且仍然在Apache软件基金会。该系统是发电机的分布式架构和BigTable的提升列的家庭模式之间的混合。从视图卡桑德拉数据模型点它是一个多维图象通过其中每个应用程序创建其自己的密钥空间的关键索引。除了列族被引入不同列的新概念，代表列的列表。数据秩序井然在RandomWrite和SequentialWrite操作列。每一行内列由它们的名字排序为好。通过卡桑德拉提出的分区解决方案是类似于由BigTable的先驱提供的溶液和包括使用散列函数的。内部持久性依赖于本地文件系统上。

3.另一个例子是来自LinkedIn工程师推广应用。伏地魔[14]
提供上述相同的主要特征，同时也提供了一些新的问题。其中一些将是：系列化，为只读节点的支持，压缩。LinkedIn使用该系统作为其底层存储系统。

3. Hadoop框架

Hadoop的方法本非关系世界非常简单，提供了可靠的共享存储和分析系统[12]。存储由HDFS通过MapReduce的供给，和分析。还有其他部分的Hadoop，但这些功能是使用最频繁的，因此最重要的。

这种架构的主要优势是成本的降低。基础设施是围绕引力可以存储日期pentabytes并且还可以提供给它的共享访问网络的想法。即使几个问题可以出现围绕写作的想法，并从多个磁盘读取，一个简单的查询的检索时间非常改善，因此缺点是非常可以接受的。时服用考虑这种做法的问题是进入脑海的另一个方面是硬件故障：一旦你开始使用硬件的许多作品，其中一个可能失败的几率是非常高的。用于管理数据丢失的常见方法是在整个复制：数据的冗余拷贝被系统保存，以便在发生故障时，还有另一个副本可用。Hadoop的通过保持默认重复三次[12，13]的数据的每个块处理这个问题。第二个问题是，大多数分析任务需要能够将数据以某种方式结合起来; 从一个磁盘上读取数据可能需要与来自任何其他100个磁盘的数据进行组合。各种分布式系统允许数据从多个源进行组合，但正确地这样做，这是极具挑战性。

MapReduce的 提供了一个编程模型，从磁盘读取和写入，把它变成了套键和值的组合抽象的问题。这是很容易比较经典的节目，由于它使用的功能和不同的编程语言如Java，Python和C的事实对目前的讨论很重要的一点是，有两个部分的处理，地图和减少功能，和仍然有类似于在实际混合发生的界面的第三部分。MapReduce的还提供可靠性，类似于HDFS。

现在，Hadoop是由Apache软件基金会的一个开源项目。的Hadoop如Zookeper，核心，猪一些其它部件提供进一步的功率在处理和研究大型数据集。

MapReduce程序通常并行工作，从而使人们能够访问足够多的机器为大规模的数据集[15]。MapReduce的工作通过将加工成两个阶段：图阶段和减少相位。每个相具有键 - 值对作为输入和输出，和类型的这些都可以由程序员选择。程序员还提供了两个功能：地图功能和减少功能，其中的每一个被写入一个优选的编程语言。

甲MapReduce工作是由客户端所需要的工作单元：它由所述输入数据，所述的MapReduce程序，和配置信息。map任务和reduce任务：Hadoop的通过将其划分为任务，其中有两种运行作业。MapReduce的还提供与作业跟踪器和多个任务服务器的。作业调度器通过协调所有作业系统上运行，并通过对任务的任务服务器运行建立一个时间表工作为主。任务服务器是运行任务和发送进度报告，作业服务器的奴隶。一个必须提到的一个重要方面是，如果一个任务失败，作业跟踪器可以重新安排它在不同的任务跟踪器[15]的事实。

输入到一个MapReduce工作通常被划分成固定大小的块称为输入分割。Hadoop的创建为每个分割，它运行由用户为在分割的每个记录中定义的函数或过程一个映射任务。

如果粗略地认为Hadoop的作为优化做到最好来运行的map / reduce数据所在的节点上的任务。这个概念通常被称为数据局部性，并且是在数据检索一个非常重要的方面。

地图任务写自己的输出到本地磁盘，而不是HDFS。地图输出为中间输出：它是由reduce任务处理，以产生最终的输出，而一旦工作完成地图输出可以扔掉。因此，将其存储在HDFS，与复制，将是矫枉过正。如果说之前的地图输出已被降低消耗的任务运行映射任务的节点出现故障，那么Hadoop的会自动重新运行另一个节点在地图上的任务来重新创建地图输出。

HDFS。 HDFS就派上用场了，当数据集跨出一台物理机的存储容量，我们需要几台机器来处理任务。该管理整个计算机网络的存储文件系统被称为分布式文件系统。由于他们是基于网络的，网络编程的所有并发症出现，从而使得分布式文件系统比普通的磁盘文件系统更为复杂。一个在这一领域面临的最大挑战是使容错的文件系统，以无数据丢失[12]一个节点故障。

块。 为HDFS默认测量单位是块的大小。这是数据的最小量，它可以读取或写入。HDFS具有块的概念，但它是一个更大的单元64 MB默认情况下。文件被分成块大小的块，其中存储作为独立的单元：因为他们在单个磁盘文件系统发生的事情发生在Hadoop中以同样的方式。具有基于块结构的组织有一个分布式文件系统颇有些优势。第一个好处是最明显的：一个文件可以比网络中的任何单个磁盘大。其次，使得块而不是文件的工作单元简化了存储子系统。简单的东西，所有的数据库开发者的愿望，尤其是在这种情况下节点故障是经常。与块存储子系统交易，

Namenodes和的Datanode。 甲HDFS群集具有两种类型的节点，在主 - 从配置操作：一个名称节点（主设备）和多个数据节点（从）。名称节点是一个与文件系统的命名空间电荷。它保持在树中的所有文件和目录的元数据。命名空间图像和更新日志是在一个更为持久的方式这些信息的表示。名称节点也知道在所有给定文件块的位置，但是，它并不存储块位置持续的数据节点，因为这些信息是从数据节点在系统启动时重建。对文件系统的访问是由用户通过与名称节点和数据节点[12]通信建立。

的Datanode是文件系统的工人。他们存储和检索块时，他们被告知（由客户或名称节点），他们报到定期与他们存储块的完整列表名称节点。

如果没有名称节点，不能使用的文件系统。事实上，如果运行名称节点的机器都将下降，所有在文件系统中的文件会丢失，因为就没有找出如何将文件从该数据节点块重建的方式。这就是为什么是使NameNode的弹性要失败的重要原因，和Hadoop想出了这个两个机制。第一个是备份组成文件系统的元数据的持久状态的文件。Hadoop的可配置使名称节点写入它的持久状态到多个文件系统。这些写入操作是同步的，通常原子。

HBASE。 HBase的是建立在HDFS之上的分布式面向列的数据库。当我们需要实时读/写随机访问大型数据集的数据的HBase进入现场。虽然我们

有他们没有考虑到非常大的规模和分布建立用户关系数据库的可能性[4,9]。

许多厂商试图找到一个解决这个问题，因此已经提出了不同的选择。他们提供复制和分区发展解决方案，数据库超出了单个节点的限制，但这些插件通常只是最后一种选择和复杂的安装和维护。

他们还危及RDBMS功能集。当我们谈论RDBMS的比例加入，复杂的查询，触发器，视图和外键约束望而却步成为运行昂贵。HBase的来满足从相反方向上的缩放问题。它是从底层向上构建，以通过增加节点只是规模。第一次使用HBase的是在网表，抓取网页和他们的属性，通过网页URL键的表。

使用地图降低数据存储到标记表的应用程序。表是由行和列组成。表细胞有不同的版本。默认情况下，这个版本只是在一个细胞中插入任何类型的信息的时间HBase的分配时间戳。考虑到保留了信息的类型，这是一个字节数组。

表行键也是字节数组，所以理论上任何可以作为从字符串到多头甚至序列化数据结构的二进制表示的行密钥。表行由行键，表的主键排序。所有表访问是经由表的主键[9]。

即使我们不谈论非SQL使用索引数据库HBase的使用列族为关系索引的响应。因此，行的列被分成列族。所有列的家庭成员有一个共同的前缀，因此，例如，列博客：图像和博客：作者是博客列家族的成员，而天气：标识属于天气家庭。

表会自动HBase的水平分割成多个区域。每个区域由一个表中的行的子集。的区域是由它的第一行，包括，行和最后一行，排他的，加上随机产生的区域识别符定义。正如HDFS和MapReduce与作业和任务跟踪器沿着由在主设备操作配置中的节点（namenodes）-slave（数据节点）的，MapReduceHBase HBase的特征还在于由负责一个或多个区域的簇的HBase的主节点服务器奴隶。HBase的主负责引导初始安装，用于向注册regionservers分配区域，以及用于回收RegionServer的失败[4]。主节点是不是真的装，主要是因为它的功能不会存储数据组成。

该regionservers携带零个或多个区域和客户端的读/写请求领域。RegionServer的从节点保留在HBase的的conf / regionservers文件，你会列出Hadoop的CONF数据节点和任务服务器/从机文件10]。启动和停止脚本是像那些在Hadoop中使用的远程命令机制同样基于SSH的运行。CONF / HBase的-site.xml中和CONF / hbase-env.sh文件用于保持群集的站点配置中，具有相同的格式及其等同起来HDFS的。

HBase的，也有一个名为-ROOT-和.META一些特殊的目录表。在其中保持当前列表中的状态，最近的历史和位置的所有地区。该-ROOT-表保存.META名单。台地区。该.META。表保存所有用户空间区域[10]的列表。

4.系统架构

用于测试的硬件平台由两台机器通过路由器连接在一起的：台式机和笔记本电脑。台式机的特点是：AMD闪龙2600+ 1.6 GHz时，768 MB DDR内存，160GB硬盘，操作系统为Windows XP SP2 32位。笔记本的特点是：英特尔Core 2 Duo T5550 1.83 GHz的，2 GB的DDR内存，160 GB硬盘驱动器，安装了Windows 64位操作系统。

有安装Hadoop的平台和HBase的两种方法：在Windows上使用Linux环境下的软件仿真，或直接在Linux操作系统上。选择此测试平台的选择是安装使用虚拟机，一个在每两台机器的Linux操作系统。对于这两个虚拟机，我们使用VMware播放。

到所述第一虚拟机（在网络的第一节点 - 笔记本），其表示群集的主已经通过VMware播放下列分配：1044 MB RAM，单个处理器，14 GB硬盘驱动器，一个网络适配器设置为桥接模式，以便该虚拟机被认为是

被连接到物理网络上。到第二虚拟机（桌面）已经被分配：524 MB的RAM，单个处理器，9 GB硬盘驱动器，网络适配器。

上所述两个虚拟机中安装了Ubuntu 9.10（KarmicKoala）32位。安装Linux后的Sun Java 1.6.0_20使用apt-get的工具集成到Ubuntu和专用的用户创建运行Hadoop和HBase的[5]，安装SSH服务器和SSH密钥创建，使两台机器都安装能不会被提示输入密码每次，然后Hadoop的平台安装开始彼此之间进行连接。

从Hadoop的平台网站的Hadoop的最新版本下载并设置为（的Sun Java路径上的更新，使用配置文件hadoop-env.sh）。下一步是配置使用三个XML文件的Hadoop应用：核心-site.xml中，HDFS-site.xml中和mapred-site.xml中。

在核site.xml文件默认的Hadoop文件系统的名称设置。在HDFS-site.xml文件的文件系统的位置和在HDFS使用的复制因子被设置。

在mapred-site.xml文件保持主机和MapReduce的作业服务器在运行的端口。之后，这些文件已被相应的修改，在HDFS文件系统格式化（这个应该只在第一次设置Hadoop集群来实现）。所有这些设置都在群集的两个节点上进行。

来自同一个站点，HBase的最新版本已被下载。像Hadoop的，有HBase的命名hbase-env.sh其中的Sun Java 6路径被标记和IPv6必须禁止的配置文件。

在HBase的-site.xml文件四个属性被修改：HBase的，HBase的根文件夹运行饲养员的模式（全分布式）和位置和名称。所有这些设置都在群集的两个节点上进行。

5，关于绩效评价

5.1。通过行数从表测试

表中的HBase的存储记录由行ID升序排列。这个测试的目的是评估时采取行顺序和随机读取和写入数据的性能。使用10.000，100.000，300.000，500.000，700.000 HBase的性能进行了评估，

1.000.000行。对于每个上述情况下面的步骤随访：

步骤1.称为“测试”的表，用一个单一的列族和单个列创建，然后插入含有随机生成1.000字节值（随机插入由行ID）行。

步骤2.将步骤1中创建的表被删除。另一个表称为“测试”与单个列家庭和创建一列，然后插入含有随机生成1.000字节值（顺序插入由行ID）行。

步骤3.将来自步骤2使用表，多个读数等于从表中的记录的数量的作了顺序（顺序读取）。

步骤4.将来自步骤2使用表，多个读数等于从表中的记录的数量的随机制成（随机读取）。

步骤5.将步骤2中，在记录的更新顺序制成（顺序更新）使用该表。

步骤6.将步骤2使用表，上的记录进行更新，随机（随机更新）制成。

Ť能够1。 ŤĚ 重由N个 ü RO的MBER WS

行数	10000	100000	300000	500000	700000百万	
顺序插入/第二	1112	1136	1119	1650	1560	1443
随机插入/第二	1045	1075	1098	1461	1194	832
连续读数/秒	2534	2173	1875年	1418	1075	1154
随机读数/秒	3519	2127	1875年	1466	1069	1054
连续更新/秒	1154	1123	1110	1091	967	824
随机更新/秒	1107	1190	1127	1014	745	666

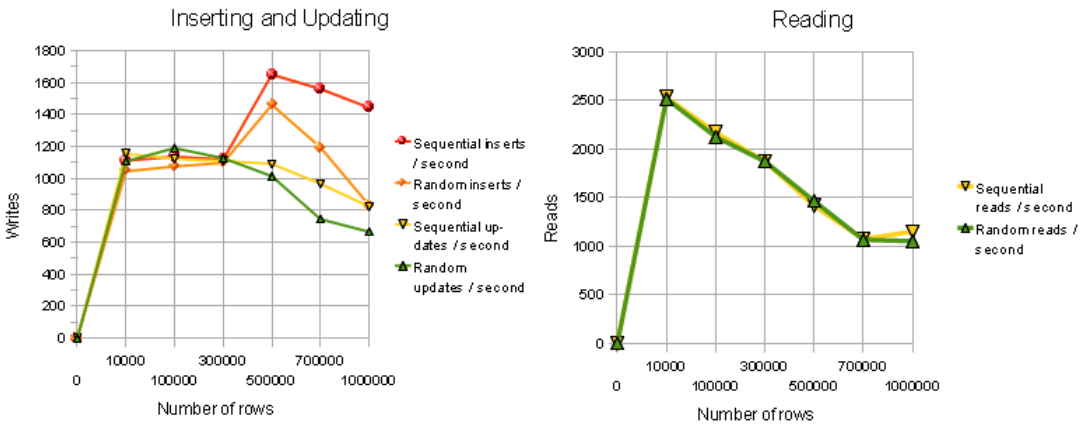

图1。 插入，更新和阅读记录

从图1中应当注意，存在在随机插入，这是容易预测微不足道的性能损失。重要的是，两种模式之间的差别是非常小的（顺序和随机插入）。

写入速度（在插入）均为最高为一些500000行，速度与所述行数的增加稍有下降。

关于表中现有的行的更新，以证明比插入快多达300,000行的表工作，随着越来越多的行的更新记录的平均速度下降。

图1示出之间顺序和随机读取的是因为操作的HBase方式非常小的差异。

最高读取速度已经报道了较少的记录表中，随着越来越多的记录下降。然而，似乎有许多行的表保持近似恒定的速度。

与上具有大量的记录不同于由BigTable的获得这些表执行这六个主要业务谷歌的BigTable，HBase的性能相比，HBase的是远低于Bigtable的，谁宣布采用约4000读数单tablet服务器连续读数的平均车速每秒[6]。一个这种差异的可能理由的可能是，在本次测试使用的硬件资源是低于由谷歌使用。

5.2。经柱家庭的数量测试

一个值得讨论的问题是可以在HBase的表使用列族的数量。在该试验中，我们研究了两个速度读，写，从具有多个列族更新表的行和我们尝试识别可以在HBase的表[8]可以使用列族的最大数量。千柱的家庭 - 这个评估是由相对少数列的家庭数量较多的制造。

用10，50，100，200，500，700和1000列的家庭表HBase的性能进行了评价。对于每个被做了以下步骤的上述情况：

步骤1.所谓“测试”的表格与多个列的家庭，一列和单行创建。对于每一列中（顺序插入）随机生成该单列1.000字节值插入。

步骤2.使用在步骤1中创建的表，一些顺序5000个读数制成。步骤3.使用在步骤1中创建的表，一些5000随机读取作了。步骤4.使用在步骤1中创建的表，从表中的行的顺序更新作了。图2显示的主要问题之一是时间列家庭添加到表中。建立一个家庭的列所需的时间大大增加。

Ť能2。 Ť通过esting ñ棕土Ø˚F科拉姆ñ家庭

列数 家庭	10	50	100	200	500	700	1000
建立时间为一列 家庭（MS）	12	24	45	56	133	100	拒绝连接
顺序插入（列 家庭/秒）	181	41	77	92	51	32	拒绝连接
连续读（列 家庭/秒）	800	23	142	39	6	Eclipse的崩溃	拒绝连接
随机读取（列 家庭/秒）	800	23	140	40	6	Eclipse的崩溃	拒绝连接
连续更新（列 家庭/秒	136	50	76	97	56	Eclipse的崩溃	拒绝连接

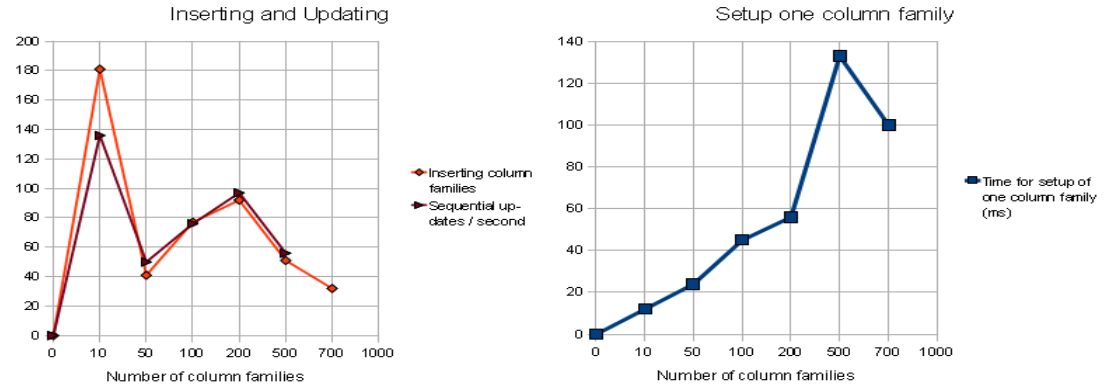

图2。 插入和更新记录

的记录写入速度（插入）并没有从这些升级的不同，但在数超过200名族越大，速度明显下降。在700倍的家庭，可以做在建表和插入记录的唯一的表。任何进一步的运行契机Eclipse的突然关闭。

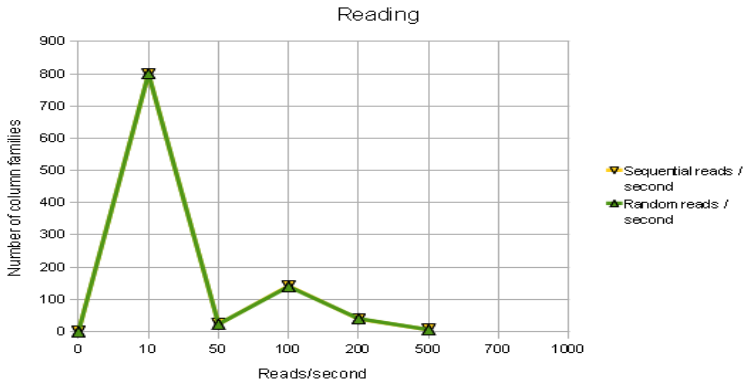

图3。 阅读记录

从表中读取记录是一样慢顺序和随机。当从10列家庭表中读取数据，操作快捷，但由于列家庭数量的增加，速度减小越来越多。

HBase的是建立在谷歌的BigTable规格。谷歌专家认为，列族可以在有限的数目使用，不超过几百个。这是事实，数量可高达500，但性能大大降低。在超过500个族，表

可以建立的，但不能使用[7,8]。在试图建立一个表，列1000户，得到的答复是一快一："连接被拒绝"。

6.改进

6.1。使用HBase的表作为源MapReduce的软件测试

在这个测试中，我们分析工作的HBase的MapReduce与算法的方式。大致。那会基于非常大的日志文件报告真实情况的模拟审理。测试台结构是相同的，作为第一个测试中，为了延长迄今所作的比较。

用于创建可小程序的可能的要求："鉴于代表谁是注册在该网站用户访问日志，以保存每个网页（从网站）的表，计算出有多少次每个用户访问该网站的页面"。该表是由行ID和列家族包括单个列。假设每个用户的访问被保存：

- "ID" + USER_ID +" _" +时间戳的行ID形成;
- 所访问的网页的URL地址。

使用500.000 10.000表，100.000，300.000，HBase的+ MapReduce的性能进行了评估，
700.000和1.000.000行。对于每一个的以下步骤上面的例为过程：第1步插入升序称为"测试"的表中仅具有一个列族称为"coloana1" 和含1.000字节\随机产生的数据串一个column.Records创建为了通过排ID（顺序地）。

每个记录的行ID的结构示于表3中。

表3。 行ID结构

	字段1	现场2	现场3	现场4
内容	"ID"	数（与每个用户相关 介于1和2000）	"_"	唯一的编号（保持行标识的唯一性）
类型	串	诠释	串	诠释

第2步：它实现的MapReduce算法剧本开始，它需要所有的数据从表"测试"，记录通过记录，使用表的全扫描和通过当前记录的行ID的功能地图（ ）。功能地图（ ）删除行ID字符"_"（区域3），并且在表中确保一个唯一行ID的最终数量（字段4）并转发到减少（ ）函数对等（用户ID，1） - 加1以价值来标记的用户的"访问"。减少（ ）函数接收并收集一对值，以使得，当完成时，插入新的从在"结果"表应用所述算法产生的数据。

该脚本给下表列出的结果。如图所示，执行时间与增加行数增加了，却不料多少并没有增加。对于从一百万条记录的表处理数据时，总时间约14分钟。一个地图步骤的平均执行时间为约94毫秒和减少步骤的平均性能为大约32毫秒。

表4。 总EXE C ution恬 ES

记录数	10000	100000	300000	500000	700000	百万
MapReduce的总执行时间 程序（秒）	11	39	115	314	379	881

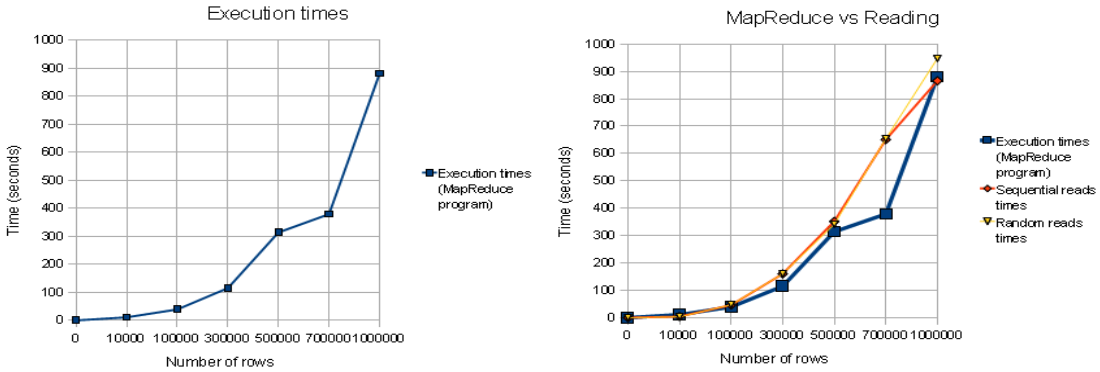

图4。 使用的MapReduce

与进行（从数据库读取数据）第一测试的结果从该测试（MapReduce的程序执行）获得的结果进行比较，应该注意的是，虽然在MapReduce的程序中比在HBase的性能测试程序执行多个操作，从读出的数据与多条记录（即运行相同的数据提取操作加上它们的处理和插入在HBase的的“结果”表）的表，MapReduce的程序执行时间越小。

这个结果是预料之中的合理的，因为：

- 在MapReduce的记录不通过一个脚本（顺序或随机）从数据库中提取，但他们通过HBase的提供的仪器“表扫描”中提取。由于这样的事实，这个仪器是HBase的集成，它经过优化，更快的提取记录;

- 的功能图（ ）和降低（ ）的步骤被执行速度极快，因此执行时间越小。

七，结论

在使用Oracle Server和PL / SQL红帽平台进行我们的测试表明，在SQL数据库足够，而不是频繁少数或数据询问他们是可行的，但不是在与一个Hadoop架构中注册的结果表现安装在可比的Centos 4.8的二进制图像。

HBASE与正朝着面向列的其他非SQL数据库一起往往与关系数据库进行比较。即使他们在执行过程中有很大的不同，他们有相同的目标：提出一个解决方案，数据存储和探询的信息大数据集。可能出现的可能出现的问题提出的事实，两者之间的比较，即使他们很大的不同是值得做的事情。

正如我们上面所HBASE指出的是面向使用列的分布式数据库系统。HBase的是的Hadoop的延续，提供读具有基于HDFS架构的数据存储/写随机存取。一个非常大的行数（十亿大小），大列数（百万规模），水平分区的能力和容易复制的一个大的能力：它是从零开始以下几个重要的原则建系统中的节点的数目。

该表结构反映提供的数据序列化，存储和撤出了很好的支持物理组织。真正的问题属于谁必须以适当的方式进行存储空间管理应用程序的用户。

在严格的意义上关系数据库通过观察12条规则科德定义。一般的关系数据库有一个使用具有ACID属性和强大的SQL引擎背后的行和列的固定结构。重音落在强一致性，参照完整性，从语言SQL的物理和复杂的查询抽象。然后，我们可以轻松地创建二级指标，把内部和外部连接复杂，使用功能，如求和，计数，排序，分组和跟踪多个表，行和列的数据。

对于中小型应用，关系数据库和MySQL类型PostgeSql提供简单，灵活，成熟的使用，事情有时是不可替代的。但是，如果我们想扩展

到更大的数据量，我们发现，关系数据库不再是可取的，因为它们在性能和分布显著下降。

缩放信息几乎涉及所有破了Codd的关系型数据库的规则。因此，关系数据库的实力不在于在处理大型数据集。

虽然我们生活在世界上的解决方案似乎是Hadoop的，指的是信息时代具有非常大量的数据，它的未来将在最终用户和软件可以与它进行交互的方式明确。否则，关系数据库的能力和他们的广泛来自于易用性，由于结构化查询语言。Hadoop's当量为SQL中，询问程序的XQuery，能够处理从XML文件中提取的数据。这样我们就可以使用面向对象的程序使用XQuery一起用于构建基于XML的应用程序。Hadoop的另一个优点是，它并没有存储在内存中的任何空记录的事实。

在使用Hadoop和HBase的平台的优势，我们可以发现，该数据是并行处理的，所以执行时间decreases.The数据被复制，这样总有空间每台机器上的备份和就业问题传递到HDFS。

另一个好处是，一个或多个列的家庭可以添加在任何时间删除。为了添加或删除列族，该表必须先被禁用，因此，直到它被重新激活仍无法使用。

缺点可能是HBase的不支持表之间的联接。这不是一个主要的缺点，因为所有信息都必须在一个单一的表放在一起，并可以更容易地访问。通过这种方式，它消除了对连接的需要。

提取HBase的数据被证明是相当快速的，不论表项的个数。此外，提供的HBase进行表扫描，其中HBase的+ MapReduce的测试已被证明是由于比表扫描的选项内的HBase实现读取顺序/随机数据更有效的可能性。HBase的的另一个好处是动物园管理员的使用其目的是释放的各种任务的主节点作为检查群集服务器，客户端应用程序的可用性，用于发送答复表根。

8.参考

[1] R.琼斯，防RDBMS：分布式键值存储的列表，http://www.metabrew.com/article/anti-rdbms-a-list-of-distributed-key-value-stores/。[2] J.索贝尔，大海捞针：照片千亿高效的存储，

http://perspectives.mvdirona.com/2008/06/30/FacebookNeedleInAHaystackEfficientStorageOfBill ionsOfPhotos.aspx。[3] F.昌，J.迪安，S。等，Bigtable的：分布式存储系统对于结构化数据，OSDI

2006年。

[4] R.罗森，HBase的提交者， HBase的，http://docs.thinkfree.com/docs/view.php?dsn=858186

[5] HBase的，www.apache.org/hadoop/HBase/HBaseArchitecture [6] A. Khetrapal，V.内甚，HBase的和Hypertable的大规模分布式存储，系统：为开源BigTable的实现方式中，性能评价，http://www.ankurkhetrapal.com/downloads/HypertableHBaseEval2.pdf [7] A.饶，S.藏，HBase的-0.20.0性能评价，http://cloudepr.blogspot.com/2009_08_01_archive.html [8] K.达纳，Hadoop的HBase的性能评价，http://www.cs.duke.edu/~kcd/hadoop/kcdhadoop-report.pdf [9] J.格拉茨，J D Crzans，HBase的推移实时，在Hadoop的峰会2009年HBase的介绍[10] HBase的-0.20.2文档，http://hadoop.apache.org/hbase/docs/r0.20.2。[11] K. Loney，“数据库10g - 完整的参考”，编辑了Mc GRAW希尔，2004。[12] T.白，Hadoop的 - 权威指南，埃德奥赖利，2009年6月[13] J.文纳，Hadoop的专业版，版Apress出版，2009。[14] J.克雷普斯，项目伏地魔，LinkedIn 2008. [15] J.院长S.格玛沃特，MapReduce的：简单数据处理大型集群，谷歌公司，

2004年。