数据库电子书

数据库新技术风向标：Hadoop

Hadoop 是一个分布式系统架构，它可以用来应对海量数据的存储，而这样的数据量往往是以 PB 甚至 ZB 来计算的。Hadoop 的存储系统我们称作 Hadoop Distributed File System(HDFS)

- *数据库新技术：Hadoop*
- *企业如何选择数据库新技术：Hadoop、MapReduce*
- *Hadoop 的首要任务是标准化*
- *大数据新技术中 Hadoop 关注度最高*

TechTarget 中国

数据库新技术风向标：

Hadoop

Hadoop 是一个分布式系统架构，它可以用来应对海量数据的存储，而这样的数据量往往是以 PB 甚至 ZB 来计算的。——James Kobielus

未来商业智能系统：Hadoop 来当家

现在，当人们提到大数据的时候首先想起的技术往往是 Hadoop MapReduce，像 Hadoop 这样的分布式架构在 10 年之前的运用是非常少的——互联网发展刚刚起步，从地球诞生到 2003 年的全球数据量一共是 5EB，而我们现在每两天就能生成 5EB 的数据。传统的交易数据库在应对数据激增的挑战时已经显现出不足，企业越来越多地开始部署数据仓库、商业智能系统来进行数据分析等工作。随着 Hadoop MapReduce 在大数据方面起到了越来越重要的作用，那么我们今天就在这里了解一下什么是 Hadoop MapReduce，它们对如今的 IT 起着怎样的作用。

什么是 Hadoop？

Hadoop 是一个分布式系统架构，它可以用来应对海量数据的存储，而这样的数据量往往是以 PB 甚至 ZB 来计算的。Hadoop 的存储系统我们称作 Hadoop Distributed File System(HDFS)，它是由 Doug Cutting 创建的，其灵感来源于 Google 的一篇学术论文。Doug Cutting 是谁呢？著名 Apache 开

源项目 Lucene 和 Nutch 的作者。重要的是 Hadoop 也是开源的。

Hadoop 项目创始人 Doug Cutting

什么是 MapReduce？

拿新浪微博来举个例子，用户每分钟都会生成几万甚至几十万条信息，这个数据量是非常大的。新浪的数据中心有大量的服务器在生成数据，那么我们如何能够快速地访问这些数据?Hadoop 使用的就是 MapReduce，它的概念第一次出现也是在 Google 的论文中。MapReduce 遵循“分治法”，数据以 KeyValue 对来组织。它以并行的方式来处理一个计算节点中的数据，这些数据会分布在许多不同的系统当中。对数据进行整理分类之后进行处理。

Hadoop MapReduce 的影响

针对一个标准 PC 服务器，Hadoop 将连接到所有的服务器然后将数据分布

到这些节点当中。它将所有的节点视为一个大的文件系统，对数据进行存储和处理，因此它是一个 100%的分布式文件系统。如果数据量增加到之前系统无法承受的情况，我们还可以增加额外的节点，让整个系统的扩展性更好。Hadoop MapReduce 在成本方面同商业软硬系统相比具有一定优势，因为其开源的属性。随着 Hadoop 的逐渐普及，相信技术人员成本也会进一步降低，Hadoop 的价值也将凸显出来。此外，Hadoop 还是 NoSQL 数据库的主要部署架构之一。

目前，Hadoop 项目已经由 Yahoo 公司转移到了 Hortonworks，这是一家硅谷风投公司 Benchmark Capital 与前者合资组建的公司，他们将继续开发该技术。雅虎软件工程副总裁 Eric Baldeschwieler 将担任 Hortonworks 公司 CEO。而最近，Hadoop 的支持者之一社交网站 Facebook 也迁移了 30 PB 的 Hadoop 集群。除了开源社区的支持，Hadoop 也得到了商业软件供应商的青睐，据笔者了解，越来越多的传统数据库厂商也在他们的产品中逐渐增加 Hadoop 特性，其中包括了 Oracle、Teradata 等。以下厂商的数据仓库和 BI 产品已经添加了对 Hadoop 和 MapReduce 的支持：

- Greenplum
- Informatica
- Teradata（AsterData）
- Pentaho
- Talend

总之，如果 Hadoop MapReduce 以及 NoSQL 等技术得到广泛运用的话，传统 SQL 数据库系统不能解决的非结构化数据将不再成为问题。而大数据概念不断推广，Hadoop 与商业系统的搭配将成为一种必然的趋势，数据集成软件也将在数据挖掘等场景中扮演重要的角色。

（作者：孙瑞　来源：TT 中国）

大数据分析时代：Hadoop MapReduce

当 Yahoo 宣布成立新公司 Hortonworks 接手 Hadoop 服务之后，业内的目光再次集中到这家大型互联网公司，而这一次的关键字是“大数据”。

在波士顿举行的 Enzee Univers 2011 大会上，厂商、分析师以及咨询师认为目前大数据技术已经在企业软件中占据了一席之地。无论目前结构化数据还是非结构化数据，它们在深度与广度上都飞速地增长着，企业能否有效管理并挖掘利用这些数据将决定信息化建设的发展走势。

Hadoop MapReduce：企业数据仓库的替代品？

针对大数据领域，其实有很多技术提供商都参与了 Yahoo 的项目。Apache Hadoop 是一个开源项目，Yahoo 就是其中最大的贡献者;Google MapReduce 是 Hadoop 架构的一个主要在组件，开发出的软件可以用来分析大数据集，它在目前的火爆程度已经无需赘言;Cloudera 是 Hadoop 最早的技术支持、服务和软件提供商，它今后将直接与 Yahoo 的 Hortonworks 展开竞争。此外，EMC 还推出了付费的 Hadoop 产品并基于 MapR Technologies 公司的技术。

据 Yahoo 前任首席数据官 Usama Fayyad 的说法，在一些场景中 MapReduce 和 Hadoop 可以良好地协同，为大型计算任务提供网格支持，而并

不是所有。有些情况下，它们并不是必须品，但现在许多企业都在过度地追捧使用 MapReduce 和 Hadoop，这将造成不良的影响。其中一位参会人员 Shawn Rogers 也表示，目前 Hadoop 过分部署的问题已经逐渐浮出水面："新技术的出现就像玩具公司推出新产品一样，我们总会第一时间把它买回家，现在是时候反思一下 Hadoop 的弊端了。"

Forrester 机构的高级分析师 James Kobielus 表示，其实部署 Hadoop 并不是完全必要的。具有 shared-nothing 并行处理架构的企业数据仓库平台完全可以支持数据库内分析(in-database analytics)和高性能数据管理。Kobielus 在他即将发表的报告中，向早期 Hadoop 实施者提出了一份调查，询问他们针对 PB 级别数据仓库是否首先考虑试错法(tried-and-true approach)。

Kobielus 说："根据案例调查显示，许多企业都会利用 Teradata 或者 Oracle 的产品作为 EDW。但是他们也会在 Hadoop 上构建大数据项目，其中原因很多，比如通过使用 Apache Hadoop，他们能够避于支付大量的软件许可费用，还可以根据变更的需求更改原代码从而得到更高的灵活性，此外全球的 Hadoop 社区也不断涌现出惊艳的创新。"

Kobielus 也同意前两位的观点，并不是所有的认为都需要用到 Hadoop 和 MapReduce，尽管 Hadoop 将逐渐成为出类拔萃的分析平台，但它目前与企业数据仓库相比，在实时集成以及健壮的高可用性方面都存在这一定的缺陷。

尽管 Hadoop 存在一定缺陷是不争的事实，但还是有许多企业用户已经将 Hadoop 软件纳入了他们的数据管理系统工具中。Intuit 公司的数据仓库架构师 Arup Ray 向我们介绍，他的公司在进行即席分析(ad hoc analysis)时，已经将 Hadoop 当做 ETL 引擎了。此外，Intuit 还使用了 Netezza 的技术进行部分分析工作。

相反地，像 T-Mobile 这样的大型电信运营商还是拒绝使用 Hadoop 技术，它们的网络系统主管 Christine Twiford 说："针对是否使用 Hadoop 我们也进行了讨论，但最后我们还是选择使用 Netezza 产品，我认为它以及完全能满足我们的需求了。" 据介绍，T-Mobile 早在五年前就更换了 Oracle 应用，转而使用 Netezza 的产品，在数据加载速度上提高了 50%。

尽管如此，关于 Hadoop 和 MapReduce 的讨论还是连绵不绝，而像 T-Mobile 这样的公司也不止一家。在 TechTarget 最新的一份 IT 调查报告中显示，只有 1%的用户表示他们的数据仓库架构中使用了 Hadoop 技术，13%的用户表示在 2012 年有使用 Hadoop 的意向。结果与 Gartner 的报告相吻合。

目前技术产品推广的力度很大，而且大数据分析软件的竞争也空前的激烈。最新的 IBM Netezza Capacity Appliance 在上周的会议中正式问世，它具有在几分钟之内分析 10 PB 数据的能力，它也是 Netezza 被 IBM 收购之后推出的第一款设备。虽然 IBM 官方并没有明确指出，但我们都知道新设备是瞄准了大数据

领域。

开源能否引领新浪潮？

Forrester 机构的 Kobielus 指出，像 Hadoop 和 R 语言这样的开源工具已经成功开启了大数据分析之门。而 Rogers 则认为开源虽然在这一方面起到了非常积极的作用，但是它们是以一种不太成熟的方式进入市场的。就拿 Hadoop、Pentaho 和 Jaspersoft 举例，与传统的私有产品相比，开源技术的发展步伐更慢一些。虽然它是探索前沿的一个非常好的方式，但是开源的精神能否跟上主流的需求这是一个问题。

（作者：Nicole Laskowski 译者：孙瑞 来源：TT 中国）

Hadoop 的首要任务是标准化

虽然流行度逐渐升温，但是根据 Forrester 研究机构的高级数据管理分析师 James Kobielus 的说法，开源技术 Hadoop 在应对大数据分析时还存在这一定的障碍。其中包括了如何存储上百 TB 的数据以及 Hadoop 互操作性标准的缺失。

在 TechTarget 网站最近的一次采访中，Kobielus 向我们介绍了大数据存储的问题，以及为何标准化对于 Hadoop 普及来说是一件好事。

最近对于 Hadoop 技术和大数据分析的谈论非常多，Hadoop 受到了越来越多的认可，但是为什么并不是所有人都用它呢？

Kobielus：在部署一个大数据分析项目时，不管你用的是 Hadoop 集群还是传统的数据仓库，我们知道要应对的是几百 TB 的存储压力，这部分成本是十分昂贵的。所以大数据领域里，真正的成本因素是存储，要花多少钱购买存储设备？你能承受多大的存储？有多少数据可以存放在磁带中?最重要的是存储部分，而不是你选用了哪种技术。

在您的研究中，Hadoop 使用者中有多少企业的数据量已经达到 PB 级别了？

Kobielus：现实中，大多数 Hadoop 集群是达不到 PB 数据级别的，而且是差的很远，他们更多的是管理几百 TB 的数据。但是在我调查的客户中，很多

人表示数据增长到 PB 级别时，存储问题是最让人头疼的。这也就是为什么我们并没有看到很多扩展到 PB 级别的传统数据仓库，原因很简单，就是成本问题。

那么除了存储的成本问题之外，Hadoop 和大数据分析还有哪些挑战？

Kobielus：整个 Hadoop 生态系统还处在起步阶段，同传统的数据仓库技术相比还不成熟。目前主流的企业数据仓库厂商还有许多没有添加 Hadoop 的特性，即使是有，也是没有完全地集成到他们的核心数据仓库工具中。这是 Hadoop 不成熟的一个具体表现。

此外，Hadoop 社区并不标准，我的意思是它的标准化同其他开源社区存在一样的问题。许多用户或者公司登录同一个社区，然后自己构建软件并开放源代码。这些功能的确是被许多人用到，但是它缺乏一个统一的正式的标准，或者是批准过程。现在，Hadoop 或者开源社区中有许多人会说标准化是一条错误的路线。我也理解他们要表达的意思，但是事实就是在没有标准化的情况下，随之而来的就是风险，而大部分公司是无法承受这部分风险的。

为什么说没有标准化就是存在着潜在的风险呢？

Kobielus：事实上，Hadoop 集群目前还没有一个普遍的参考架构，而一个参考架构则可以为可插拔存储层提供一个明确的接口，同样为跨多平台的 MapReduce 互操作性提供一个标准的界面。这个架构和 SOA 社区在过去十多年开发的那些参考架构(SOAP、WSDL 和 UDDI 等)相类似，最终目的都是为了加

强互操作性。对于 Hadoop 来说，我们还没有互操作性和认证的测试，这对于许多领域来说都是致命的，比如你的公司是一家大型企业，你们在不同的部门中使用了 Hadoop 集群，而它们想要结成一个共同体。而现在还没有这样的标准，也没有实时数据控制与访问的技术说明。这样的技术对于许多大型企业在接受上会存在困难。

Hadoop 早期的使用者该如何应对互操作性问题？

Kobielus：如果你想要在分布式 Hadoop 中做真正的实时数据分析话，那么你需要去编写大量的代码来进行功能定制，然而许多时候还会出现 bug 或者根本无法工作。在这里有很大的风险，我认为业界目前最重要的应该是为互操作性和认证测试创建一个普遍的参考架构，并希望具体出炉一些正式的标准，比如 HDFS 版本等相关标准。

（作者：Mark Brunelli 译者：孙瑞 来源：TT 中国）

新技术中 Hadoop 关注度最高

随着企业中存储的数据量向 TB 甚至 PB 级别进发，越来越多的组织开始关注“大数据”这一话题。然而，存储这些数据已经是极具挑战性，更不要说去利用大数据进行分析了。如何将结构化与非结构化混杂的原始数据转换为可用的数据，并利用到实时商业智能系统中，可以帮助企业做出更加明智的决策。

传统的关系型数据仓库和 BI 系统有着一系列明确的实践以及成熟的产品，它们能够将高度结构化的数据充分利用起来，并允许业务用户将数据进行切分以便创建成报表供管理者进行分析与决策，它基本上回答了“正在发生什么”以及“为何会发生”这两个问题。举个例子：“在指定区域里，这一段时间内都售出了哪些商品?”

将分析运用到大数据的概念颠覆了我们所熟知的情况，为企业打开了一扇通往信息的大门，能够提出更多平时考虑不到的问题，最终可以帮助企业制定提升竞争优势的战略。

Forrester 研究机构的分析师 Brian Hopkins 表示：“在过去的 20-25 年里，企业用于决策的信息最多只占到 5%。在竞争如此激烈的商业环境下，企业不得不继续探索那剩余 95%的数据，能够更好利用这些数据的企业必将更具竞争力。

● 传统关系型数据库管理系统正努力跟上

与大多数 RDBMS 管理的结构化交易数据不同，“大数据”系统中充满了来自不同数据源的数据，包括社交媒体中的数据流、日常 Web 日志活动、全球定位系统的方位数据以及各种传感器生成的数据。

权威研究机构 Gartner 估计，目前全世界的信息量正在以每年 59%的速度增长，而速度将逐年递增，然而处理数据的能力却远远跟不上数据增长的速度。除了数据种类之外，数据变化的速度也是需要关注的内容，企业如不能快速地处理数据，那这些信息有可能将变得没有任何意义。

极限数据管理的概念让传统的数据仓库以及 BI 系统遭受了前所未有的挑战，在应对更新速度极快的海量数据时，传统关系型数据库显得力不从心。而同新技术相比，RDBMS 无论是在成本方面还是性能方面都不存在明显的优势。

来自 Ventana 研究机构的分析师总监 David Menninger 指出：“在数据分析方面，目前已经出现了思维模式的变迁，我们现在应该考虑如何利用细粒度数据得出更加准确或者说新型的分析结果。”

● 企业思维模式的转变

思维模式转变的催化剂是大量新技术的诞生，它们能够处理大数据分析所带来的 3 个 V 的挑战。新技术运动的核心就是大规模并行处理(MPP)数据库技术：在处理大型的数据集时，它能够将数据库工作负载自动地分配到多个商业服务器

当中，然后并行地运行这些任务，从而获得更高的性能提升。

在 MPP 技术基础之上诞生了新的列式数据库，与传统的行式数据库相反，它将数据以列的方式进行存储，大大节省了存储空间，数据库查询速度也相应提升。针对大型分析 BI 工作负载设计的数据库内分析也是值得我们深入研究的一项新技术，除了提升数据查询速度之外，在成本方面也有着很大的竞争优势。当然，软硬件集成的平台产品也是受到了越来越多的关注，从存储到数据仓库再到服务器、网络集成在一起，并进行必要的调优，集成设备可以满足大数据应用管理的需求。

这些新技术对 RDBMS 的统治地位造成了极大的挑战，除以上所介绍的技术之外，另一个大家都比较熟悉的技术在大数据分析领域也扮演了重要的角色——Hadoop。扎根于开源社区，Hadoop 已经是目前大数据平台中应用率最高的技术，特别是针对诸如文本、社交媒体订阅以及视频等非结构化数据。除分布式文件系统之外，伴随 Hadoop 一同出现的还有进行大数据集处理 MapReduce 架构、可扩展的 NoSQL 数据库 Cassandra 以及 Hive 数据仓库等。

● 大数据应用 Hadoop 关注度最高

根据最新名为《Hadoop 与信息管理》的 Ventana 基准测试报告显示，有半数以上的受访者都在使用或者在评估 Hadoop 技术来作为大数据平台的标准。这部分用户希望能够更加深入地对数据进行分析，特别是大数据分析。然而，抛去对新技术近乎于偏执的狂热不谈，我们应该看到 Hadoop 技术的成熟度还处在相

对较低的阶段，而且相关的开源工具集也较少。

尽管新技术能够为企业解决许多难题，但是也给现有的数据仓库从业者带来了挑战。目前大数据分析真正的问题已经不是数据量，而是如何为你的企业带来更多的分析能力。来自 Gartner 的高级分析师 Yvonne Genovese 表示：“数据量的问题已经存在多年，但是这并不是极限数据真正关注的。企业 IT 人员应该更多地关注如何将数据转化为有用的信息，以及如何将有用信息转化为更好的决策。”

（作者：Beth Stackpole 译者：孙瑞 来源：TT 中国）

企业如何选择数据库新技术

最近，像 NoSQL 数据库、Hadoop 和 MapReduce 等新兴数据库技术正逐渐地成为使用传统关系型数据库的企业，特别是那些依赖高度数据集成计算需求的企业，进行系统选型的主流替代品。

但是现在几乎所有相关厂商都将他们生产的，使用这些技术的软件描述成可以“改变游戏规则”的软件。在这样的情况下，企业将不得不面对一个非常艰难的时期，即如何从其中找到适合自己企业的新数据库技术，如果这个技术确实存在的话。

为了能够帮助那些企业做出正确的选型决策，TechTarget 采访了 Ventana 咨询公司的 VP 兼研究室主任，数据管理技术专家 David Menninger 先生。Menniger 在采访中向我们展示了分析型数据库、Hadoop 和 MapReduce 的定义以及优缺点，他还特别提到了为什么 NoSQL 数据库更应被称为“不仅仅只有 SQL(Not-Only-SQL)”的数据库的原因。以下是我们的采访内容：

● 分析型数据库

为什么这一段时间里，我们总是能听到关于分析型数据库的相关消息?

分析型数据库之所以有这么大的吸引力，是因为它是以 SQL 作为基础的，换句话说，我们现在因为使用 Oralce、IBM、Teradata 或是其他数据库而积累的所有

的 SQL 的知识在这里是可以触类旁通的。而现在我们所听到的分析型数据库的范畴，应该包含列式结构数据库和 MPP(Massively Parallel Processing，大规模并行处理)技术，事实上大多数分析型数据库同时具备 MPP 和一部分列式数据库技术、以及在该领域里的厂商包括 EMC-Greenplum、Aster Data、Vertica InfoBright、Paraccel、IBM-Netezza 以及其他一些新兴的小厂商。

分析型数据库都有哪些优点呢?

最主要的就是成本优势。分析型数据库厂商们找到了能让数据库处理能力大幅提升的方法——在 SQL 为基础的环境下使数据库具备了处理更大容量数据和并发进程的能力。通常情况下，我认为他们最大的优势在于能够将上述的改进实现在低价位商用硬件上，除了 Netezza 之外，Netezza 的特殊之处在于它只能基于 IBM 的低价位商用硬件上，而这往往作为 IBM 商业解决方案的套件或打包方案的一部分罢了。所以，我才说分析型数据库最吸引人的地方就是使用它能够在并不需要增加昂贵的硬件环境下解决大容量数据的问题，从而从根本上降低企业的成本。每个人都需要成绩证明自己，企业的 IT 部门证明自己价值的方式就是通过使用分析型数据库向老板们证明不需要付出昂贵的代价他们一样能够解决问题。这才是最有吸引力的。

什么是分析型数据库的弱项呢?

弱项是分析型数据库并不能像 DB2 或 Oracle 那样完全地支持 SQL，比如因为 Oracle 自身就带有 PL/SQL，所以其他的产品没必要购买对 SQL 的支持。尽管 Netezza 中使用了一些企业数据库技术提供对 PL/SQL 的兼容支持，但仅仅是有而并不是 100%的支持。所以，你可以想象一下，如果你从你熟悉的 Oracle 环境下迁移到分析型数据库平台上，你所面对的最大的问题就是此前你所写的 SQL 语句在新环境下能不能执行的问题。

对于分析型数据库你还有什么要提醒大家的吗?

像这种第三方技术类的产品，其生态系统(指配套的硬件、软件、技术和服务等)并不像传统产品那么的健全，尽管它们也都是以 SQL 为基础的，甚至很多的工具可通用和移植。但是，我还想强调的就是它们的周边并不健全。

● Hadoop MapReduce

什么是 Hadoop?

从功能的角度上来看，Hadoop 提供的是一种能够在多进程或联机环境下进行大规模数据的存储和分析的解决方案。事实上它有两个组件：一个是分布式文件系统，它可以取出一组数据然后将它们分发给不同的机器并且提供冗余处理。你可以想象，在这个系统里，对每一个数据在 3 个不同的节点上进行了 3 次复制。因此，一旦有任何一个数据所在节点出现问题，还有两个其他节点的相同的数据可以使用。

这就是所谓的“HDFS”，Hadoop 分布式文件系统。对于 Hadoop 来说，它本身还有总共 11 个区块，但这些区块里最重要的两个就是 HDFS 和 MapReduce。HDFS 解决的是分析数据的问题，而 MapReduce 解决的是如果你将数据分发给多台机器，那么其中的一部分数据经分析后会合并至一台机器进行集中处理的问题。

Hadoop 和 MapReduce 的主要优势是什么?

它们确实有值得夸赞之处，但最大的优势在于它们是开源的，这就意味着它们对你来说是免费的，当然不能做到百分百的免费，因为毕竟你多多少少的都要根据需求购买一些付费的服务和支持。但它们却为你提供了低成本解决问题的方案。它们本身是没有数据库使用 License 限制的，于是它们很轻松地能够在 10 台、50 台或者上百台的机器上并发处理大规模数据。你只需制定一个相对简单的映射和简化的规则，它们将负责分配这些任务给每一台机器并确保所有的任务都能成功完成，如果有任何一台机器故障，它们将重新分配该机器上的任务给其他正常的机器。所以，Hadoop 在成本控制方面的潜在优势甚至超过它在分析数据库以及对于分析型数据库的可扩展性方面的优势。

Hadoop 和 MapReduce 最大的缺点是什么?

现在我们实际上处在一系列更为严重的风险和下滑趋势中，因为 Hadoop 以及 MapReduce 所处的环境并非 SQL 环境，这对于已经熟悉 SQL 环境和技能的你来

说，无疑是个不容忽视的难题。但是你也可以借助其他不同的技能。例如，你可以尝试使用一段大串的非 SQL 语句去实现一个 MapReduce 任务。这就使你不得不放弃你所积累和具备的 SQL 数据库经验和技能。正因为它运行在非 SQL 环境中，所以你可用到的工具远比你想象中的要少。

Hadoop 支持分等级的 SQL。Hadoop 11 个组件中有一个实际上就是限制 SQL 支持程度的。厂商们之所以提到对 SQL 和扩展性的限制，是因为可以售卖他们的相关支持工具。但是围绕着 Hadoop 的整个周边产业链远比分析型数据库的小很多。当然了，如果你确实有很大规模的数据需要处理，那这就另当别论了。

● NoSQL

对于企业来说什么时候才是选择 NoSQL 数据库的最好时机?

在 NoSQL 的体系里，性能才是所有问题的关键。也许最初，NoSQL 确实被定义为无 SQL 的，但事实上它却是不仅仅只有 SQL。它具备一定的 SQL 能力。NoSQL 实际上是为处理非常高性能需求状况的，比如像支持实时的网游或者实时证券交易行情系统等等，这些系统需求的共同之处就是需要你存储关键值匹配而非一些结构化的文件。而这恰恰就是 NoSQL 所做的一切。NoSQL 是开源的，事实上到目前为止我还没听说有商业性的 NoSQL 产品。我理解这是属于开发文化或技术圈氛围等领域的事情，如果你对 NoSQL 之路感兴趣，你尽可加入其中，

会有不断的更多的开源产品出现的。

（作者：Mark Brunelli 译者：武扬　来源：TT 中国）

深入理解 Hadoop 与数据仓库的概念

那些想要弄清楚“大数据”概念的组织需要做出一个选择，是要采用传统的数据仓库概念和现有的数据仓库架构，还是不熟越来越流行的开源 Hadoop 分布式处理平台，或者使用这二者的结合。

那些想要从简单的 BI 报表转向深度数据挖掘与预测分析的企业，第三种选项看上去是最靠谱的。TechTarget 网站最近采访了 Forrester 机构的高级数据管理分析师 James Kobielus，他向我们分析了企业如何从快速变化的海量数据中获取有价值的洞察力。在本文中，您将了解到如何将现有数据仓库架构的功能发挥到最大，Hadoop 的优势与劣势，以及大数据时代中每一个数据仓库厂商的发展等。

我看到了对大数据几个不同的定义，请问 Forrester 是如何理解时下这一流行概念的？

James Kobielus：大数据事实上是引用极限可扩展分析的概念，“极限可扩展分析”这个词在我看来是人们所说大数据的核心。在某种程度上，是可以用三个 V 来概括的：Volume，数据量，可以使 TB 可以是 PB 甚至更大;Velocity，数据流动速度，实时的获取、转换、查询与访问数据;Variety，数据的种类，包括各种结构化数据、非结构化数据以及半结构化数据。在分析方面，它是指所有能够挖掘并获取意义的数据集。

企业对数据仓库概念应如何理解，才能够搞清大数据的意义？

Kobielus：我认为数据仓库能够通过三种方式来帮助企业处理好数据问题：第一、在一个企业数据仓库中，你按照主题领域来划分组织你的数据，而这些主题领域往往是比较稳定的，很长一段时间内都不会有任何改变，比如数据仓库架构中的 OLAP cube，无论是物理上实现还是逻辑上的划分。换句话说，你的客户数据在一个分区里，财务数据在另一个，HR 数据在第三个，以此类推。这样做的好处就是有利于你根据数据的关联性来匹配下游的应用和用户。这就是数据仓库数据库管理的核心所在，也是通过数据仓库来处理大数据的最重要的方式。

那么第二种方式是什么？

Kobielus：第二种方式是数据库内分析的概念以及利用数据仓库执行数据剖析、数据清洗以及数据挖掘或者回归分析。换句话说，就是做全套的数据挖掘，但是是在数据仓库内部执行。这能够帮助你处理好数据，因为你使用数据挖掘或者回归分析来从根本上了解数据集模式。然后使用数据库内挖掘(in-database data mining)来填充下游的分析数据集市，数据挖掘和统计模型专业人士可以利用它将复杂的模式实现可视化。举例来说，他们使用那些模式来辨别潜在的大客户，这样可以有限将他们设定为销售的目标。使用数据库内分析以及像 MapReduce 这样的技术，可以在一个高并发高扩展的数据库架构内将数据挖掘自动化。

数据库内分析目前的应用状况如何?是不是每个企业都会用到它？

Kobielus：虽然不是所有人都会用到数据库内分析技术，但是我们可以看到越来越多的企业已经对它产生了浓厚的兴趣。如果你的数据挖掘规模很大，数据库内分析已经被视为是最佳实践。众所周知，目前大量实际生产中的数据仓库都是面向操作型商业智能的，它们更多的是在生产报表、执行即席查询(ad hoc query)等，很少进行数据挖掘。但随着数据量的增长，数据挖掘的必要性也就凸现出来，而数据库内分析的价值也将体现。利用这一技术的目标就是加速并扩展你的数据挖掘项目，同时根据一组通用的参考数据使所有的挖掘在数据仓库中保持一致。

第三种最佳实践是什么？

Kobielus：第三就是将数据仓库作为数据治理的核心，主数据可以合理地在数据仓库中进行维护。当你的数据仓库作为数据治理与数据清洗的核心时，它能够帮助你搞清楚所有的信息。在整个企业架构中，也许会有成百上千个应用在向数据仓库中添加数据。数据就像洪水一般实时地流动，数据仓库就是其中的枢纽，确保大数据集可靠恰当地用在下游的消费当中。

在大数据蔓延的今天，传统的数据仓库厂商都为客户做了哪些努力？

Kobielus：Teradata、Oracle-Exadata、IBM-Netezza、HP-Vertica 等等都在做大数据。绝大一部分数据仓库厂商能够利用网格或者云架构将他们的产品扩展

到 PB 级别，而且也有绝大一部分能够完成数据库内分析，即在大规模并行数据仓库网格或者云环境中实现。他们还可以在企业数据仓库之内来支持数据转化和数据清洗功能。

从现在大多数的媒体报道来看，处理大数据挑战，Hadoop 似乎是最好的办法，您怎么认为？

Kobielus：如果你想要处理好大数据，你需要企业数据仓库和 Hadoop 的组合来完成。我不同意人们把 Hadoop 看作是处理大数据问题唯一的救命稻草。其实现在的企业数据仓库基本上已经能够做到 Hadoop 可以实现的任何功能。Hadoop 同传统的企业数据仓库系统相比，优势就是开源，它是免费的，但是需要提醒企业用户不要忽视开源 Hadoop 的许多无形维护费用。可以说 Hadoop 是未来五到十年内下一代企业数据仓库发展的最大动力。

（作者：Mark Brunelli 译者：孙瑞 来源：TT 中国）

Java 开发 2.0：

用 Hadoop MapReduce 进行大数据分析

美国政府产生大量数据，只有一部分是普通民众所感兴趣的。各种政府机构免费发布关于 US 经济健康状况和更改社会人口统计资料的数据。U.S. Geological Survey (USGS)发布国内外地震数据。

世界各地每天都有很多个小型地震发生。其中大多数发生在地壳深处，没有人能感觉到，尽管如此，但是监听站仍然会进行记录。USGS 以 CSV（或逗号分隔值）文件的格式发布每周地震数据。

每周文件平均不是很大 — 只有大约 100KB 左右。但是，它可以作为学习 Hadoop 的基础。记住，Hadoop 有能力处理更 大的数据集。

跟踪震动

我近期从 USGS 网站下载的 CSV 文件有大约 920 多行。如 清单 1 所示：

清单 1.一个 USGS 地震数据文件的行数统计

```
$> wc -l eqs7day-M1.txt

 920 eqs7day-M1.txt
```

CVS 文件内容如清单 2 所示（这是前两行）：

清单 2. CVS 文件的前两行

```
$> head -n 2 eqs7day-M1.txt

Src,Eqid,Version,Datetime,Lat,Lon,Magnitude,Depth,NST,Region

ci,14896484,2,"Sunday, December 12, 2010 23:23:20 UTC",33.3040,-116.4130,1.0,11.70,22,

 "Southern California"
```

这就是我称之为信息丰富 的文件，尤其是当您想到它总共有 920 行记录时。然而我只想知道在该文件报告的这一周内每一天有多少次地震发生。我想知道在这 7 天内哪个区域是地震频发区。

我第一个想到的就是使用简单的 grep 命令来搜索每天的地震数。看看这个文件，我发现数据记录是从 12 月 12 开始的。因此我对该字符串执行了一次 grep-c，其结果如清单 3 所示：

清单 3.12 月 12 有多少次地震发生？

```
$> grep -c 'December 12' eqs7day-M1.txt

98
```

安装 Hadoop 如果您之前没有安装 Hadoop，那么现在就装。第一步，下载最新版二进制文件，解压，然后在您的路径上设置 Hadoop 的 bin 目录。完成这些您就可以直接执行 hadoop 命令了。使用 Hadoop 要求您执行它的 hadoop 命令，而不是像您所见到的那样调用 java 命令。您可以向 hadoop 命令传选项，诸如在哪里可以找到您的 Java 二进制文件（例如，表示您的 map 和 reduce 实现）。在我的示例中，我创建了一个 jar 文件，告诉 Hadoop 我想在我的 jar 文件内运行哪个任务。我也向 Hadoop 类路径添加了一些运行我的应用程序所需的附加二进制文件。

现在，我知道在 12 月 12 日有 98 条记录，也就是说有 98 次地震。我只能沿着这条记录向下，对 12 月 10 日的记录执行一次 grep，接着是 11 号，等等。这听起来有点乏味。更糟糕的是，我还需要知道在该文件中的是哪几天。我确实不关心这些，甚至有时候我可能无法获取该信息。事实上，我只想知道在七天这样一个时间段内任何一天的地震次数，使用 Hadoop 我就可以很容易的获取这一信息。

Hadoop 只需要几条信息就可以回答我的第一个和第二个问题：即，要处理哪条输入以及如何处理 map 和 reduce。我也必须提供了一个可以将每件事都联系起来的作业。在我开始处理这些代码之前，我需要花点时间确定我的 CSV 数据整齐有序。

使用 opencsv 进行数据解析

除了地震 CSV 文件的第一行之外，第一行是文件头，每一行都是一系列逗号分隔数据值。我只对数据的 3 个部分感兴趣：日期、地点和震级。为了获取这些资料，我将使用一个很棒的开源库 opencsv，它将会帮助我分析 CSV 文件。

作为一个测试优先的工具，我首先编写一个快捷 JUnit 测试，确认我可以从 CSV 文件的一个样例行获取的我所需要的信息，如清单 4 所示：

清单 4. 解析一个 CSV 行

```
public class CSVProcessingTest {

 private final String LINE = "ci,14897012,2,\"Monday, December 13, 2010 " +

          "14:10:32 UTC\",33.0290,-115." +

          "5388,1.9,15.70,41,\"Southern California\"";

 @Test

 public void testReadingOneLine() throws Exception {

  String[] lines = new CSVParser().parseLine(LINE);

  assertEquals("should be Monday, December 13, 2010 14:10:32 UTC",
```

```
        "Monday, December 13, 2010 14:10:32 UTC", lines[3]);
      assertEquals("should be Southern California",
        "Southern California", lines[9]);
      assertEquals("should be 1.9", "1.9", lines[6]);
    }
}
```

正如您在清单 4 中所看到的，opencsv 处理逗号分隔值非常容易。该解析器仅返回一组 String，所以有可能获取位置信息（别忘了，在 Java 语言中数组和集合的访问是从零开始的）。

转换日期格式

当使用 MapReduce 进行处理时，map 函数的任务是选择一些要处理的值，以及一些键。这就是说，map 主要处理和返回两个元素：一个键和一个值。回到我之前的需求，我首先想知道每天会发生多少次地震。因此，当我在分析地震文件时，我将发布两个值：键是日期，值是一个计数器。reduce 函数将对计数器（只是一些值为 1 的整数）进行总计。因此，提供给我的是在目标地震文件中某一个日期出现的次数。

由于我只对 24 小时时段内的信息感兴趣，我得剔除每个文件中的日期的时间部分。在 清单 5 中，我编写了一个快速测试，验证如何将一个传入文件中的特定日期信息转换成一个更一般的 24 小时日期：

清单 5.日期格式转换

```
@Test
public void testParsingDate() throws Exception {
 String datest = "Monday, December 13, 2010 14:10:32 UTC";
 SimpleDateFormat formatter = new SimpleDateFormat("EEEEE, MMMMM dd, yyyy HH:mm:ss Z");
 Date dt = formatter.parse(datest);
 formatter.applyPattern("dd-MM-yyyy");
 String dtstr = formatter.format(dt);
 assertEquals("should be 13-12-2010", "13-12-2010", dtstr);
}
```

在清单 5 中，我使用了 SimpleDateFormat Java 对象，将 CSV 文件中格式为 Monday, December 13, 2010 14:10:32 UTC 的日期 String 转换成了更一般的 13-12-2010。

Hadoop 的 map 和 reduce

现在我已经找到了处理 CSV 文件以及其日期格式的解决方法。我要开始在 Hadoop 中实施我的 map 和 reduce 函数了。这个过程需要理解 Java 泛型，因为 Hadoop 选择使用显式类型，为了安全起见。

当我使用 Hadoop 定义一个映射实现时，我只扩展 Hadoop 的 Mapper 类。然后我可以使用泛型来为传出键和值指定显式类。类型子句也指定了传入键和值，这对于读取文件分别是字节数和文本行数。

EarthQuakesPerDateMapper 类扩展了 Hadoop 的 Mapper 对象。它显式地将其输出键指定为一个 Text 对象，将其值指定为一个 IntWritable，这是一个 Hadoop 特定类，实质上是一个整数。还要注意，class 子句的前两个类型是 LongWritable 和 Text，分别是字节数和文本行数。

由于类定义中的类型子句，我将传入 map 方法的参数类型设置为在 context.write 子句内带有该方法的输出。如果我想指定其他内容，将会出现一个编译器问题，或 Hadoop 将输出一个错误消息，描述类型不匹配的消息。

清单 6.一个映射实现

```
public class EarthQuakesPerDateMapper extends
Mapper<LongWritable,
 Text, Text, IntWritable> {
@Override
protected void map(LongWritable key, Text value, Context context)
throws IOException,
 InterruptedException {
if (key.get() > 0) {
 try {
  CSVParser parser = new CSVParser();
  String[] lines = parser.parseLine(value.toString());
  SimpleDateFormat formatter =
   new SimpleDateFormat("EEEEE, MMMMM dd, yyyy HH:mm:ss Z");
  Date dt = formatter.parse(lines[3]);
```

```
        formatter.applyPattern("dd-MM-yyyy");
        String dtstr = formatter.format(dt);
        context.write(new Text(dtstr), new IntWritable(1));
      } catch (ParseException e) {}
    }
  }
}
```

清单 6 中的 map 实现比较简单：本质上是，Hadoop 为在输入文件中找到的每一行文本调用这个类。为了避免除了 CSV 头部，首先检查是否字节数（key 对象）为零。然后执行清单 4 和 5 中的步骤：捕获传入日期，进行转换，然后设置为传出键。我也提供了一个数：1。就是说，我为每个日期编写一个计数器，当 reduce 实现被调用时，获取一个键和一系列值。在本例中，键是日期及其值，如 清单 7 所示：

清单 7.一个 map 输出和 reduce 输入的逻辑视图

```
"13-12-2010":[1,1,1,1,1,1,1,1]
"14-12-2010":[1,1,1,1,1,1]
```

"15-12-2010":[1,1,1,1,1,1,1,1,1]

注意，context.write(new Text(dtstr), new IntWritable(1))（在清单 6 中）构建了如 清单 7 所示的逻辑集合。正如您所了解的，context 是一个保存各种信息的 Hadoop 数据结构。context 被传递到 reduce 实现，reduce 获取这些值为 1 的值然后总和起来。因此，一个 reduce 实现逻辑上创建如 清单 8 所示的数据结构：

清单 8.一个 reduce 输出视图

"13-12-2010":8

"14-12-2010":6

"15-12-2010":9

我的 reduce 实现如 清单 9 所示。与 Hadoop 的 Mapper 一样，Reducer 被参数化了：前两个参数是传入的键类型（Text）和值类型（IntWritable），后两个参数是输出类型：键和值，这在本例中是相同的。

清单 9.reduce 实现

public class EarthQuakesPerDateReducer extends Reducer<Text, IntWritable, Text,

IntWritable> {

```
@Override
protected void reduce(Text key, Iterable<IntWritable> values, Context
context)
throws IOException, InterruptedException {
int count = 0;
for (IntWritable value : values) {
 count++;
}
context.write(key, new IntWritable(count));
}
}
```

我的 reduce 实现非常简单。正如我在清单 7 中所指出的，传入的是实际上是一个值的集合，在本例中是 1 的集合，我所做的就是将它们加起来，然后写出一个新键值对表示日期和次数。我的 reduce 代码可以挑出您在清单 8 中所见到的这几行。逻辑流程看起来像这样：

"13-12-2010":[1,1,1,1,1,1,1,1] -> "13-12-2010":8

当然，这个清单的抽象形式是 map -> reduce。

定义一个 Hadoop Job

现在我已经对我的 map 和 reduce 实现进行了编码，接下来所要做的是将所有这一切链接到一个 Hadoop Job。定义一个 Job 比较简单：您需要提供输入和输出、map 和 reduce 实现（如清单 6 和清单 9 所示）以及输出类型。在本例中我的输出类型和 reduce 实现所用的是同一个类型。

清单 10. 一个将 map 和 redece 绑在一起的 Job

```
public class EarthQuakesPerDayJob {

 public static void main(String[] args) throws Throwable {

  Job job = new Job();

  job.setJarByClass(EarthQuakesPerDayJob.class);

  FileInputFormat.addInputPath(job, new Path(args[0]));

  FileOutputFormat.setOutputPath(job, new Path(args[1]));

  job.setMapperClass(EarthQuakesPerDateMapper.class);
```

```
        job.setReducerClass(EarthQuakesPerDateReducer.class);

        job.setOutputKeyClass(Text.class);

        job.setOutputValueClass(IntWritable.class);

        System.exit(job.waitForCompletion(true) ? 0 : 1);

      }

    }
```

在清单 10 中，我使用一个 main 方法将所有这一切绑在一起，该方法有两个参数：地震 CSV 文件的目录，以及生成报告的输出目录（Hadoop 更喜欢创建该目录）。

为了执行这个小框架，我需要将这些类打包。我还需要告知 Hadoop 在哪里可以找到 opencsv 二进制文件。然后可以通过命令行执行 Hadoop，如 清单 11 所示：

清单 11.执行 Hadoop

```
$> export HADOOP_CLASSPATH=lib/opencsv-2.2.jar

$> hadoop jar target/quake.jar com.b50.hadoop.quake.EarthQuakesPerDayJob
```

~/temp/mreduce/in/ ~/temp/mreduce/out

运行这些代码，Hadoop 开始运行时您将可以看到一堆文本在屏幕上一闪而过。我所用的 CSV 文件相比专门用于处理这种情况的 Hadoop，那真是小巫见大巫！hadoop 应该可以在几秒钟内完成，具体取决于您的处理功能。

完成这些后，您可以使用任何编辑器查看输出文件内容。还可以选择直接使用 hadoop 命令。正如 清单 12 所示：

清单 12.读取 Hadoop 输出

```
$> hadoop dfs -cat part-r-00000

05-12-2010      43

06-12-2010      143

07-12-2010      112

08-12-2010      136

09-12-2010      178

10-12-2010      114

11-12-2010      114
```

12-12-2010　　79

如果您像我一样，在清单 12 中首先会注意到的就是每天地震数—12 月 9 日就有 178 次地震。希望您也会注意到 Hadoop 实现了我所想要的：整齐地列出我的研究范围内每天的地震次数。

编写另一个 Mapper

接下来，我想找到地震发生在哪里，以及如何快速计算出在我的研究范围内记录地震次数最多的是哪个区域。当然，您已经猜到了，Hadoop 可以轻松地做到。在这个案例中，键不再是日期而是区域。因此，我编写了一个新的 Mapper 类。

清单 13.一个新的 map 实现

```
public class EarthQuakeLocationMapper extends Mapper<LongWritable,
Text, Text,
 IntWritable> {
@Override
protected void map(LongWritable key, Text value, Context context)
throws IOException,
 InterruptedException {
```

```
if (key.get() > 0) {
String[] lines = new CSVParser().parseLine(value.toString());
context.write(new Text(lines[9]), new IntWritable(1));
}
}
}
```

和之前获取日期然后进行转换相比，在清单 13 中我所作的是获取位置，这是 CSV 阵列中的最后一个条目。

相比一个庞大的位置和数字列表，我将结果限制在那些 7 天内出现 10 次的区域。

清单 14.哪里的地震较多？

```
public class EarthQuakeLocationReducer extends
Reducer<Text, IntWritable, Text, IntWritable> { @Override protected
void
reduce(Text key, Iterable<IntWritable> values, Context context) throws
```

IOException, InterruptedException { int count = 0; for (IntWritable value :

values) { count++; } if (count >= 10) { context.write(key, new

IntWritable(count)); } } }

清单 14 中的代码和清单 9 中的代码非常类似；然而，在本例中，我限制了输出大于或等于 10。接下来，我将 map 和 reduce，以及其他 Job 实现绑在一起，进行打包，然后和平常一样执行 Hadoop 获取我的新答案。

使用 hadoop dfs 目录显示我所请求的新值：

清单 15.地震区域分布

$> hadoop dfs -cat part-r-00000

Andreanof Islands, Aleutian Islands, Alaska 24

Arkansas 40

Baja California, Mexico 101

Central Alaska 74

Central California 68

Greater Los Angeles area, California 16

Island of Hawaii, Hawaii 16

Kenai Peninsula, Alaska 11

Nevada 15

Northern California 114

San Francisco Bay area, California 21

Southern Alaska 97

Southern California 115

Utah 19

western Montana 11

从清单 15 还可以得到什么？首先，北美洲西海岸，从墨西哥到阿拉斯加是地震高发区。其次，阿肯色州明显位于断带层上，这是我没有意识到的。最后，如果您居住在北部或者是南加州（很多软件开发人员都居住于此），您周围的地方每隔 13 分钟会震动一次。

结束语

使用 Hadoop 分析数据轻松且高效，对于它对数据分析所提供的支持，我只是了解皮毛而已。Hadoop 的设计旨在以一种分布式方式运行，处理运行 map 和 reduce 的各个节点之间的协调性。作为示例，本文中我只在一个 JVM 上运行 Hadoop，该 JVM 仅有一个无足轻重的文件。

Hadoop 本身是一个功能强大的工具，围绕它还有一个完整的、不断扩展的生态系统，可以提供子项目至基于云计算的 Hadoop 服务。Hadoop 生态系统演示了项目背后丰富的社区活动。来自社区的许多工具证实了大数据分析作为一个全球业务活动的可行性。有了 Hadoop，分布式数据挖掘和分析对所有软件创新者和企业家都是可用的，包括但不限于 Google 和 Yahoo!这类大企业。

（作者：Andrew Glover 来源：developerWorks 中国）

解读下一代 Apache Hadoop MapReduce

随着 Hadoop 的流行，其局限性也在一定程度体现，各大公司也在 hadoop 上做了很多修改，下面是雅虎对 Hadoop 下一代的重构计划。

回顾

海量数据业务中，使用数量少规模大的集群比使用数量多规模小集群的成本低。规模大的集群能处理大数据集，同时也能支持更多的任务和用户。

Apache Hadoop MapReduce 框架大约能够支持 4000 台机器。下一代的 Apache Hadoop MapReduce 框架会纳入一个通用的资源调度器，用户可以自定义每一个应用程序的执行。相比早期，故障时间在大规模高可靠性的集群中代价更高，更大规模的集群上保证安全性和多重用户才能支持大规模的用户。新的架构要加强它的创新性，灵活性和硬件使用。

背景

当前 Hadoop MapReduce 框架的实现表明了它的使用年限。

从目前集群的规模和其工作负荷的变化趋势来看，MapReduce 的 JobTracker 需要大规模的调整来修复它在可扩展性，内存消耗，线程模型，可靠性和性能上的缺陷。在过去的 5 年中，我们做了一些 bug 的修复，但是最近这些修复的成本越来

越高，这表明对框架做出改变的难度越来越大。框架具有缺陷和对框架的补救是可以理解的，也是比较久远的，甚至可以追溯到 2007 年，这是我们修复 MapReduce’ s jira 的文档：

https://issues.apache.org/jira/browse/MAPREDUCE-278.

从操作的角度来看，现在的 Hadoop MapReduce 框架在有任何重要的或者不重要的变化(例如 bug 修复，性能提升和特性化)时，都会强制进行系统级别的升级更新。更糟的是，它不管用户的喜好，强制让集群的每一个用户同时更新;这些更新会让用户为了验证他们之前的应用程序是不是适用新的 Hadoop 版本而浪费大量时间。

需求

考虑如何改进 Hadoop MapReduce 框架时，注意那些高优先级的需求是非常重要的。下一代的 MapReduce 框架最迫切的需求有以下几个：

- 可靠性
- 可用性
- 可扩展性-超过 10000 台机器的集群和 200000 个 core
- 向后兼容(向前兼容)-确保在下一代的框架上用户的 MapReduce 应用不做改变依然能够运行。

- 变化-能让用户控制 Hadoop 软件栈的升级
- 可预测的潜在因素- 一个用户关心的主要方面
- 集群的使用

第二等级的需求是：

- 对 MapReduce 支持不同的编程模型
- 支持短期的服务

从上面给出的要求，需要对 Hadoop 下层处理数据的公共组件进行重新考虑。事实上，hadoop 社区里已经有一些舆论说现在 MapReduce 框架的架构不能满足上述目标，必须要进行重构，看我们 2008 年 2 月在 jira 上制定的目标。

https://issues.apache.org/jira/browse/MAPREDUCE-279.

下一代的 MapReduce

重构根本的思想是将 JobTracker 两个主要的功能分离成单独的组件，这两个功能是资源管理和任务调度/监控。新的资源管理器全局管理所有应用程序计算资源的分配，每一个应用的 ApplicationMaster 负责相应的调度和协调。一个应用程序无非是一个单独的传统的 MapReduce 任务或者是一个 DAG(有向无环图)任务。ResourceManager 和每一台机器的节点管理服务器能够管理用户在那台机器上的进程并能对计算进行组织。事实上，每一个应用的 ApplicationMaster 是一个详细

的框架库，它结合从 ResourceManager 获得的资源和 NodeManager 协同工作来运行和监控任务。

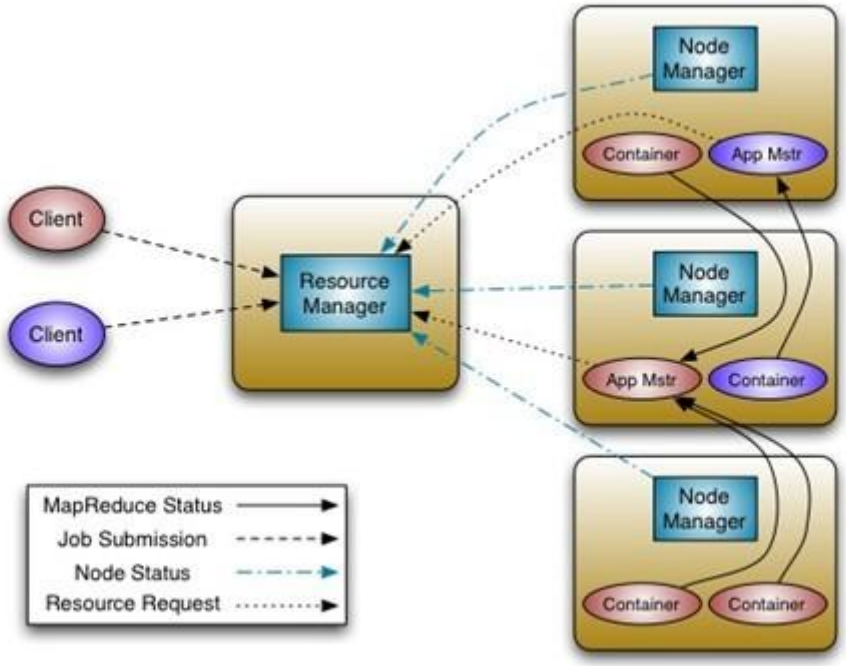
ResourceManager 支持分层级的应用队列，这些队列享有集群一定比例的资源。从某种意义上讲它就是一个纯粹的调度器，它在执行过程中不对应用进行监控和状态跟踪。同样，它也不能重启因应用失败或者硬件错误而运行失败的任务。

资源管理器是基于应用程序对资源的需求进行调度的;每一个应用程序需要不同类型的资源因此就需要不同的容器。资源包括：内存，CPU，磁盘，网络等等。可以看出，这同现在 Hadoop Mapreduce 固定类型的资源使用模型有显著区别，它给集群的使用带来负面的影响。资源管理器提供一个调度策略的插件，它负责将集群资源分配给多个队列和应用程序。调度插件可以基于现有的能力调度和公平调度模型。

节点管理器是每一台机器框架的代理，是执行应用程序的容器，监控应用程序的资源使用情况(CPU，内存，硬盘，网络)并且向调度器汇报。

每一个应用的 ApplicationMaster 的职责有：向调度器索要适当的资源容器，运行任务，跟踪应用程序的状态和监控它们的进程，处理任务的失败原因。

架构



和当前 Hadoop Mapreduce 框架实现的逐一对比和改进

可扩展性

将集群资源的管理，应用程序的生命周期，和对各个组件的管理相分离，这使得架构更好更优雅。Hadoop MapReduceJobTracker 花费相当一部分时间和效率来管理应用的生命周期，这是导致软件遭难的主要原因，将它转移到每一个具体的应用程序上有显著的意义。

可扩展性和目前硬件发展趋势的关系特别重要，现在的 Hadoop MapReduce 已经能被部署到了 4000 台机器的集群上。但是，2009 年的 4000 台最好的机器(8 核，16G RAM，4TB 硬盘)只是 2011 年 4000 台机器(16 核，48G RAM，24TB

硬盘)能力的一半。因此，运营成本的提升迫使我们使用超过 6000 台机器来组建更大的集群。

可用性

l-资源管理器使用 Apache 的 ZooKeeper 来处理失败情况，当资源管理器出错时，根据 ZooKeeper 里存储的集群状态可以很快的从备份机恢复。资源管理器会重启队列里和正在运行的所有应用。

lApplicationMaster-下一代的 MapReduce 支持对 ApplicationMaster 设置检查点的能力。MapReduce ApplicationMaster 能够从失败的状态中恢复，因为它先前已经将自己的状态存到 HDFS 里。

线兼容性

下一代的 MapReduce 使用线兼容模型使不用版本的服务器和客户端能互相通信。在未来的版本，这个特性能使集群轮替式升级--一个操作上的优势。

创新和灵活性

改进架构的一个主要的好处是让 MapReduce 有效的变成了一个系统级的资源库。计算框架(资源管理器和节点管理器)完全是非商业化的，并且 MapReduce 的特性也是免费的。

这个特性允许端用户在同一集群上同时使用不用版本的 MapReduce。这也允许不同的应用使用不同版本的 MapReduce ApplicationMaster 和运行环境。这样为应用的 bug 修复提供了很大的灵活性，增强的功能和新特性也不需要对整个集群升级。它同时也允许端用户按照自己的计划升级他们的 MapReduce 应用的版本，可以显著的增强集群的操作性。

运行用户自定义的 MapReduce 任务而不影响软件的稳定性能够促进创新。它还可以带来其他方面的特性，例如能够将用户版本的 MapReduce 程序引入在线 Hadoop 模型而又不影响其他用户。

集群的使用

下一代的 Mapreduce 的资源管理器使用一个通用概念给每个应用调度和分配资源。

集群的每一个机器概念上由很多资源组成，例如内存，CPU，I/O，带宽等等。基于应用程序定义的资源请求类型，每一个机器都是可被取代的，也可以当作容器分配给应用程序。同一个机器上可以支持多个用户，此机器上的一个容器是一组进程的集合，它独立于其他容器。

因此，它可以废弃现在的固定类型 map 的表示方法，并能减少 Mapreduce 程序的资源使用(slot)。固定类型的资源在不同时间段对集群的使用有显著的负面影响，它使 map 或者是 reduce 的资源匮乏。

除 MapReduce 以外还支持其他的编程模型

下一代的 MapReduce 提供一个完全免费的计算框架来支持 Mapreduce 和其他的编程模型。

框架允许用户通过实现一个自定义的 ApplicationMaster 来支持任何特定的框架，此 ApplicationMaster 能够从 ResourceManager 请求资源，并且这些资源满足一些常用的特性，例如隔离性，容量保证等等。

总之，它要支持多种编程模型，例如 Mapreduce，MPI，Master-Worker 和迭代模型，在同一个 Hadoop 集群上能对不用的应用程序运用合适的框架。这对应用(例如 K-Means，Page-Rank)来说非常重要，用户自定义的应用也许会超过 MapReduce 应用一个数量级。

总结

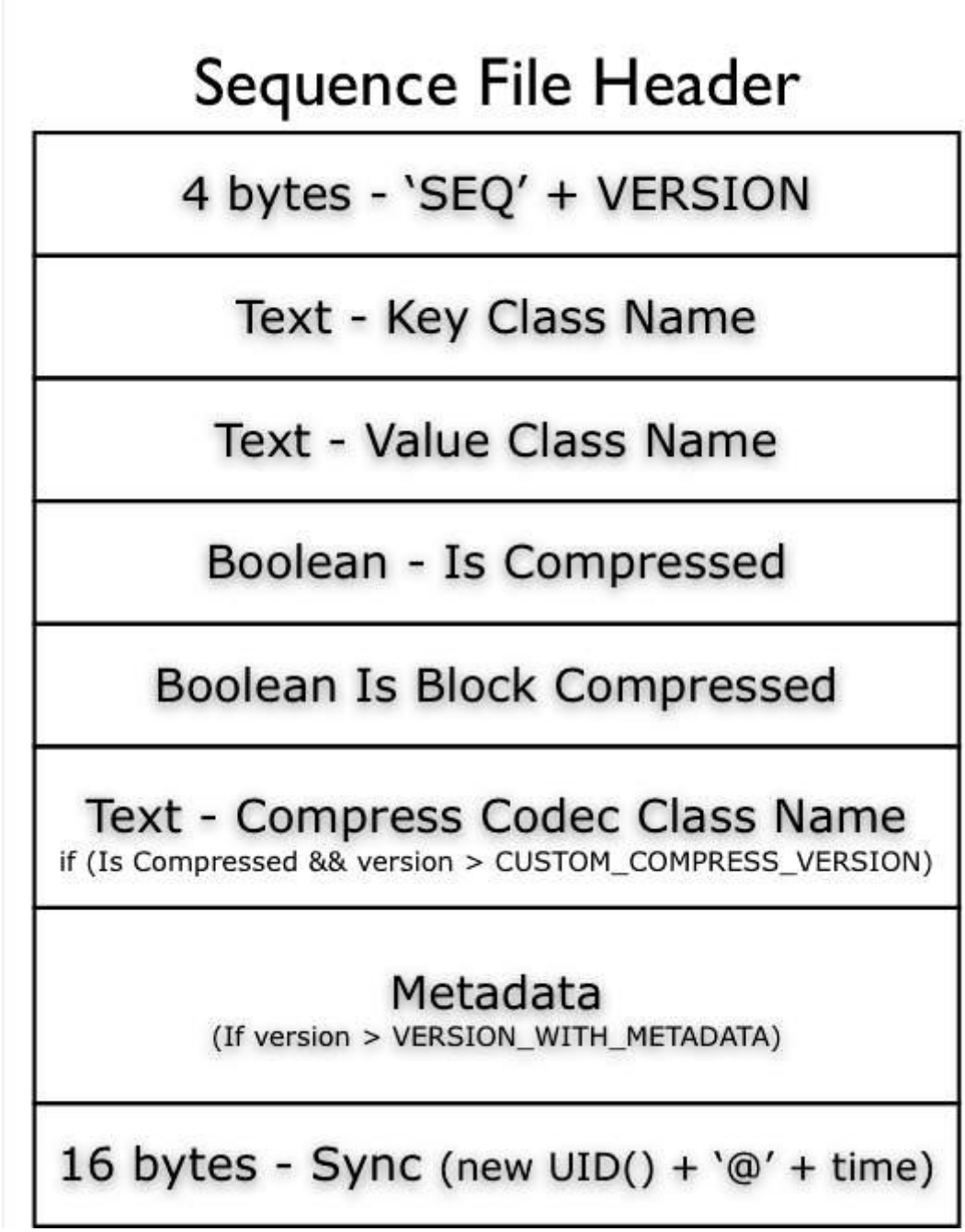
Apache Hadoop，特别是 Hadoop MapReduce 对于处理海量数据集是一个十分成功的开源项目。我们的目的是重构 Hadoop MapReduce 来解决框架现有的问题，提高框架的可用性，加强集群的利用率，同时提供对多个编程模型的支持，

也推进未来的变革。我们会协同 Apache Hadoop 社区来完成这个任务，同时将 Apache Hadoop 推进到下一个大数据量的时代。

（作者：Arun C Murthy 来源：Yahoo!）

Hadoop 数据类型与文件结构剖析

1.Hadoop’ s SequenceFile



SequenceFile 是 Hadoop 的一个重要数据文件类型，它提供 key-value 的

存储，但与传统 key-value 存储(比如 hash 表，btree)不同的是，它是 appendonly 的，于是你不能对已存在的 key 进行写操作。每一个 key-value 记录如下图，不仅保存了 key，value 值，也保存了他们的长度。

SequenceFile 有三种压缩态：

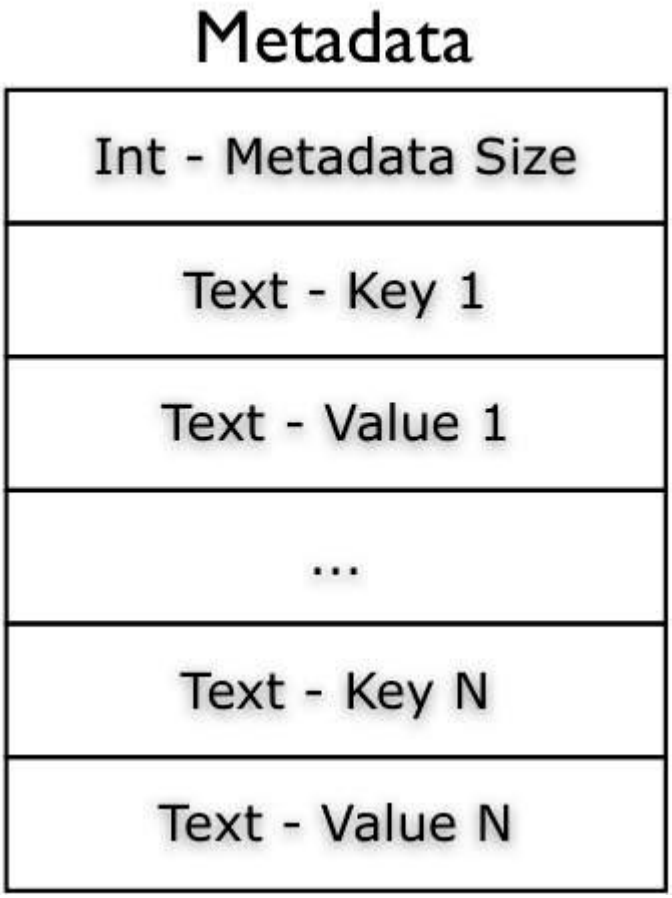
Uncompressed – 未进行压缩的状态

Record Compressed - 对每一条记录的 value 值进行了压缩(文件头中包含上使用哪种压缩算法的信息)

Block-Compressed – 当数据量达到一定大小后，将停止写入进行整体压缩，整体压缩的方法是把所有的 keylength,key,vlength,value 分别合在一起进行整体压缩

文件的压缩态标识在文件开头的 header 数据中。

在 header 数据之后是一个 Metadata 数据，他是简单的属性/值对，标识文件的一些其他信息。Metadata 在文件创建时就写好了，所以也是不能更改的。



2.MapFile, SetFile, ArrayFile 及 BloomMapFile

SequenceFile 是 Hadoop 的一个基础数据文件格式，后续讲的 MapFile, SetFile, ArrayFile 及 BloomMapFile 都是基于它来实现的。

MapFile – 一个 key-value 对应的查找数据结构，由数据文件/data 和索引文件 /index 组成，数据文件中包含所有需要存储的 key-value 对，按 key 的顺

序排列。索引文件包含一部分 key 值，用以指向数据文件的关键位置。

SetFile – 基于 MapFile 实现的，他只有 key，value 为不可变的数据。

ArrayFile – 也是基于 MapFile 实现，他就像我们使用的数组一样，key 值为序列化的数字。

BloomMapFile – 他在 MapFile 的基础上增加了一个 /bloom 文件，包含的是二进制的过滤表，在每一次写操作完成时，会更新这个过滤表。

（作者：Jon Zuanich 译者：nosqlfan 来源：Cloudera）

我们的编辑团队

您若有何意见与建议，欢迎与我们的编辑联系。

诚挚感谢以下人员热情参与 TechTarget 中国《数据库电子书》的内容编辑工作！

诚邀更多的数据库专业人士加入我们的内容建设团队！

沈宏
TechTarget中国特邀技术编辑。具有丰富的软件开发及测试经验，多年以来一直致力于数据库优化、性能测试等方向的研究与探索。

孙瑞
TechTarget 中国数据库网站编辑，三年网络媒体从业经验。负责“TT 数据库”网站的内容建设，熟悉数据库以及商业智能等企业信息化领域，拥有计算机学士学位。