第5章

支撑大数据的技术

导读案例

大数据企业的缩影——谷歌

谷歌创建于1998年9月，是美国的一家跨国科技企业，致力于互联网搜索、云计算、广告技术等领域，开发并提供大量基于互联网的产品与服务，主要利润来自AdWords等广告服务。

谷歌由在斯坦福大学攻读理工博士的拉里·佩奇和谢尔盖·布林共同创建。创始之初，谷歌官方的公司使命为“集全球范围的信息，使人人皆可访问并从中受益”。谷歌的总部称为Googleplex，位于美国加利福尼亚州圣克拉拉县的芒廷维尤。2011年4月，拉里·佩奇接替施密特担任首席执行官。2014年5月21日，市场研究公司明略行公布，谷歌取代苹果公司成为全球最具价值的商业品牌。2015年3月28日，谷歌和强生公司达成战略合作，联合开发能够做外科手术的机器人。2017年2月，Brand Finance发布2017年度全球500强品牌榜单，谷歌排名第一。

谷歌搜索引擎就是大数据的缩影，这是一个用来在互联网上搜索信息的简单快捷的工具，使用户能够访问一个包含超过80亿个网址的索引。谷歌坚持不懈地对其搜索功能进行革新，始终保持着自己在搜索领域的领先地位。据调查结果显示，仅一个月内，谷歌处理的搜索请求就高达122亿次。除此之外，谷歌不仅存储搜索结果中出现的网站链接，还存储人们的所有搜索行为，这就使谷歌能以惊人的洞察力掌握搜索行为的时间、内容以及它们是如何进行的。这些对数据的洞察力意味着谷歌可以优化其广告，使之从网络流量中获益，这是其他公司所不能企及的。另外，谷歌不仅可以追踪人的行为，还可以预测人们接下来会采取怎样的行动。换句话说，在你行动之前，谷歌就已经知道你在寻找什么了。这种对大量的人机数据进行捕捉、存储和分析，并根据这些数据做出预测的能力，就是人们所说的大数据。

谷歌的规模使其得以实施一系列大数据方法，而这些方法是大多数企业根本不曾具备的。而谷歌的优势之一是其拥有一支优秀的软件工程师队伍，这些工程师能为该公司提供前所未有的大数据技术。谷歌的另一个优势是它的基础设施，就谷歌搜索引擎本身的设计而言，数不胜数的服务器保证了谷歌搜索引擎之间的无缝连接。如果出现更多的处理或存

储信息需求，或某台服务器崩溃时，谷歌的工程师们只需添加服务器就能保证搜索引擎的正常运行。据估计，谷歌的服务器总数超过 100 万个。谷歌的机房如图 5-1 所示。

图 5-1　谷歌的机房

谷歌在设计软件时一直没有忘记自己所拥有的强大的基础设施。MapReduce 和 Google File System 就是两个典型的例子。《连线》杂志在 2012 年暑期的报道中称，这两种技术“重塑了谷歌建立搜索索引的方式”。许多公司现在都开始接受 Hadoop（MapReduce 和 Google File System 开发的一个开源衍生产品）开源代码，并开始利用 Hadoop，谷歌在多年前就已经开始大数据技术的应用了。事实上，当其他公司开始接受 Hadoop 开源代码时，谷歌已经将重点转移到其他新技术上了，这在同行中占据了绝对优势。这些新技术包括内容索引系统 Caffeine、映射关系系统 Pregel 以及量化数据查询系统 Dremel。

5.1　开源技术的商业支援

在大数据的演变中，开源软件起到了很大的作用。如今，Linux 已经成为主流操作系统，并与低成本的服务器硬件系统相结合。有了 Linux，企业就能在低成本硬件上使用开源操作系统，以低成本获得许多相同的功能。MySQL 开源数据库、Apache 开源网络服务器以及 PHP 开源脚本语言（最初为创建网站开发）搭配起来的实用性也推动了 Linux 的普及。随着越来越多的企业将 Linux 大规模地用于商业，它们期望获得企业级的商业支持和保障。在众多的供应商中，红帽子 Linux（Red Hat）脱颖而出，成为 Linux 商业支持及服务的市场领导者。甲骨文公司（Oracle）也并购了最初属于瑞典 MySQLAB 公司的开源 MySQL 关系数据库项目。

例如，Apache Hadoop 是一个开源分布式计算平台，通过 Hadoop 分布式文件系统（Hadoop Distributed File System，HDFS）存储大量数据，再通过名为 MapReduce 的编程模型将这些数据的操作分成小片段。Apache Hadoop 源自谷歌的原始创建技术，随后开发了一系列围绕 Hadoop 的开源技术。Apache Hive 提供数据仓库功能，包括数据抽取、转换、装载（ETL），即将数据从各种来源中抽取出来，再实行转换以满足操作需要（包括确保数据质量），然后装载到目标数据库。Apache HBase 则提供处于 Hadoop 顶部的海量结构化表的实时读写访问功能，它仿照了谷歌的 BigTable。同时，Apache Cassandra 通过复制数据

来提供容错数据存储功能。

在过去，这些功能通常只能从商业软件供应商处依靠专门的硬件获取。开源大数据技术正在使数据存储和处理能力——这些本来只有像谷歌或其他商用运营商之类的公司才具备的能力，在商用硬件上也得到了应用。这样就降低了使用大数据的先期投入，并且具备了使大数据接触到更多潜在用户的潜力。

开源软件在开始使用时是免费的，这使其对大多数人颇具吸引力，从而使一些商用运营商采用免费增值的商业模式参与到竞争当中。产品在个人使用或有限数据的前提下是免费的，但顾客需要在之后为部分或大量数据的使用付费。久而久之，采用开源技术的这些企业往往需要商业支援，一如当初使用 Linux 碰到的情形。像 Cloudera、HortonWorks 及 MapR 这样的公司在为 Hadoop 解决各种需要时，类似 DataStax 的公司也在为非关系数据库(cassandra)做着同样的事情，LucidWorks 之于 Apache Lucerne 也是如此(后者是一种开源搜索解决方案，用于索引并搜索大量网页或文件)。

5.2　大数据的技术架构

要容纳数据本身，IT 基础架构必须能够以经济的方式存储比以往更大量、类型更多的数据。此外，还必须能适应数据变化的速度。由于数量如此大的数据难以在当今的网络连接条件下快速移动，因此，大数据基础架构必须分布计算能力，以便能在接近用户的位置进行数据分析，减少跨越网络所引起的延迟。企业逐渐认识到必须在数据驻留的位置进行分析，分布这类计算能力，以便为分析工具提供实时响应将带来的挑战。考虑到数据速度和数据量，移动数据进行处理是不现实的，相反，计算和分析工具可能会移到数据附近。而且，云计算模式对大数据的成功至关重要。云模型在从大数据中提取商业价值的同时也能为企业提供一种灵活的选择，以实现大数据分析所需的效率、可扩展性、数据便携性和经济性。

仅仅存储和提供数据还不够，必须以新的方式合成、分析和关联数据，才能提供商业价值。部分大数据方法要求处理未经建模的数据，因此，可以对毫不相干的数据源进行不同类型数据的比较和模式匹配。这使得大数据分析能以新视角挖掘企业传统数据，并带来传统上未曾分析过的数据洞察力。

基于上述考虑构建的适合大数据的 4 层堆栈式技术架构如图 5-2 所示。

图 5-2　4 层堆栈式大数据技术架构

(1) 基础层：整个大数据技术架构基础的最底层。要实现大数据规模的应用，企业需要一个高度自动化的、可横向扩展的存储和计算平台。这个基础设施需要从以前的存储孤岛发展为具有共享能力的高容量存储池。容量、性能和吞吐量必须可以线性扩展。

云模型鼓励访问数据并提供弹性资源池来应对大规模问题，解决了如何存储大量数据，以及如何积聚所需的计算资源来操作数据的问题。在云中，数据跨多个节点调配和分布，使得数据更接近需要它的用户，从而缩短响应时间和提高生产率。

(2) 管理层：要支持在多源数据上做深层次的分析，大数据技术架构中需要一个管理平台，使结构化和非结构化数据管理融为一体，具备实时传送和查询、计算功能。本层既包括数据的存储和管理，也涉及数据的计算。并行化和分布式是大数据管理平台所必须考虑的要素。

(3) 分析层：大数据应用需要大数据分析，分析层提供基于统计学的数据挖掘和机器学习算法，用于分析和解释数据集，帮助企业获得对数据价值深入的领悟。可扩展性强、使用灵活的大数据分析平台更可成为数据科学家的利器，起到事半功倍的效果。

(4) 应用层：大数据的价值体现在帮助企业进行决策和为终端用户提供服务的应用。不同的新型商业需求驱动了大数据的应用。另外，大数据应用为企业提供的竞争优势使得企业更加重视大数据的价值。新型大数据应用对大数据技术不断提出新的要求，大数据技术也因此在不断地发展变化中日趋成熟。

5.3 大数据处理平台

5.2 节学习了大数据的 4 层堆栈式技术架构，下面学习一些常用的大数据处理平台。本节具体讲述大数据处理平台 Hadoop、Storm 和 Spark 的特点及具体应用。

5.3.1 Hadoop

Hadoop 是以开源形式发布的一种对大规模数据进行分布式处理的技术。特别是处理大数据时代的非结构化数据时，Hadoop 在性能和成本方面都具有优势。Hadoop 采用 Java 语言开发，是对谷歌的 MapReduce、GFS(Google File System)和 Bigtable 等核心技术的开源实现，由 Apache 软件基金会支持，是以 Hadoop HDFS 和 MapReduce 为核心，以及一些支持 Hadoop 的其他子项目的通用工具组成的分布式大数据开发平台。它主要用于海量数据(PB 级数据是大数据的临界点)高效地存储、管理和分析，具有高可靠性和良好的扩展性，可以部署在大量成本低廉的硬件设备上，为分布式计算任务提供底层支持。Hadoop 标志如图 5-3 所示。下面从 4 个方面介绍 Hadoop。

图 5-3 Hadoop 标志

1. Hadoop 的特性

Hadoop 是一个能够对大量数据进行分布式处理的软件框架，并且是以一种可靠、高效、可伸缩的方式进行处理的。它具有以下几个方面的特性。

(1) 高可靠性。Hadoop 采用冗余数据存储方式，即使一个副本发生故障，其他副本也可以保证正常对外提供服务。

(2) 高效性。作为并行分布式计算平台，Hadoop 采用分布式存储和分布式处理两大核心技术，能够高效地处理 PB 级别的数据。

(3) 高可扩展性。Hadoop 的设计目标是可以高效、稳定地运行在廉价的计算机集群上，可以扩展到数以千计的计算机节点上。

(4) 高容错性。Hadoop 采用冗余数据存储方式，自动保存数据的多个副本，并且能够自动将失败的任务进行重新分配。

(5) 成本低。Hadoop 采用廉价的计算机集群，成本比较低，普通用户也很容易用自己的 PC 搭建 Hadoop 运行环境。

(6) 运行在 Linux 平台上。Hadoop 是基于 Java 开发的，可以较好地运行在 Linux 操作系统上。

(7) 支持多种编程语言。Hadoop 的应用程序也可以使用其他语言编写，如 C++。

目前，Hadoop 凭借其突出的优势，已经在各个领域得到了广泛的应用，而互联网领域是其应用的主要阵地。国内采用 Hadoop 的公司主要有百度、淘宝、网易、华为、中国移动等。

2. Hadoop 架构

Hadoop 是一个能够实现对大数据进行分布式处理的软件框架，由实现数据分析的 MapReduce 计算框架和底部实现数据存储的 HDFS 有机结合组成，它自动把应用程序分割成许多小的工作单元，并把这些单元放到集群中的相应节点上执行，而 HDFS 负责各个节点上数据的存储，实现高吞吐率的数据读写。Hadoop 的基础架构如图 5-4 所示。

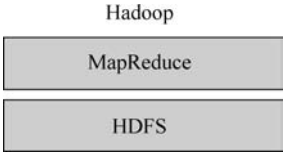

图 5-4　Hadoop 的基础架构

HDFS 是 Hadoop 的分布式文件存储系统，整个 Hadoop 的体系结构就是通过 HDFS 实现对分布式存储的底层支持，HDFS 是 Hadoop 体系中数据存储管理的基础。为了满足大数据的处理需求，HDFS 简化了文件的一致性模型，通过流式数据访问，提供高吞吐量应用程序数据访问功能，对超大文件的访问、集群中的节点极易发生故障造成节点失效等问题进行了优化，使其适合带有大型数据集的应用程序。HDFS 的系统架构如图 5-5 所示。

MapReduce 是一种处理大量半结构化数据集合的分布式计算机框架，是 Hadoop 的一个基础且核心组件。MapReduce 分为 Map 过程和 Reduce 过程，这两个过程将大任务进行细分处理再汇总结果。其中，Map 过程对数据集上的独立元素进行指定的操作，生成“键值对”形式的中间结果。Reduce 过程则对中间结果中相同“键”的所有“值”进行规约，以得到最终结果。MapReduce 这样的功能划分，使得其非常适合在分布式并行环境里进行数据处理。MapReduce 的基本原理如图 5-6 所示。

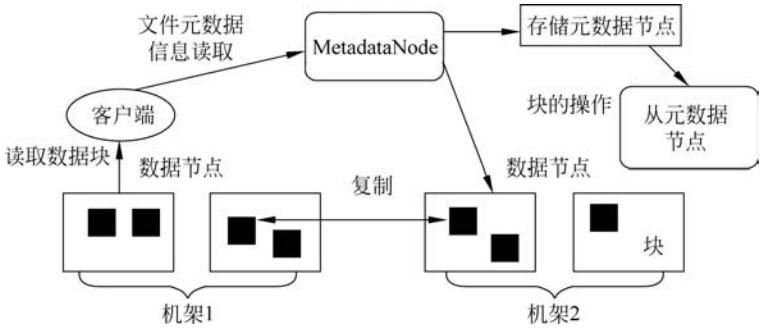

图 5-5 HDFS 的系统架构

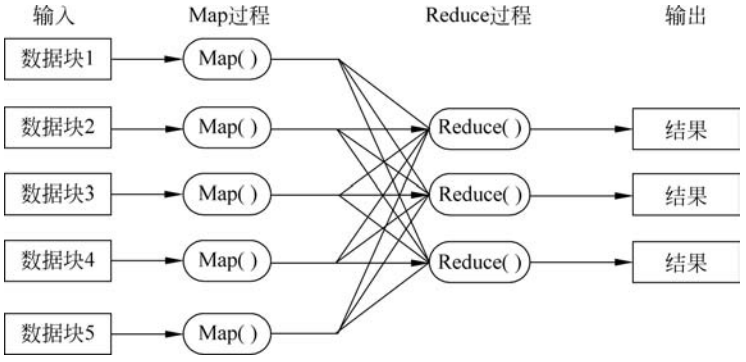

图 5-6 MapReduce 的基本原理

3. Hadoop 生态系统

经过多年的发展，Hadoop 生态系统不断完善和成熟，目前 Hadoop 已经发展成为包含很多项目的集合，形成了一个较为完善的生态系统。除了核心的 HDFS 和 MapReduce 以外，Hadoop 生态系统还包括 ZooKeeper、HBase、Hive、Pig、Mahout、Sqoop、Flume、Ambari 等功能组件，如图 5-7 所示。

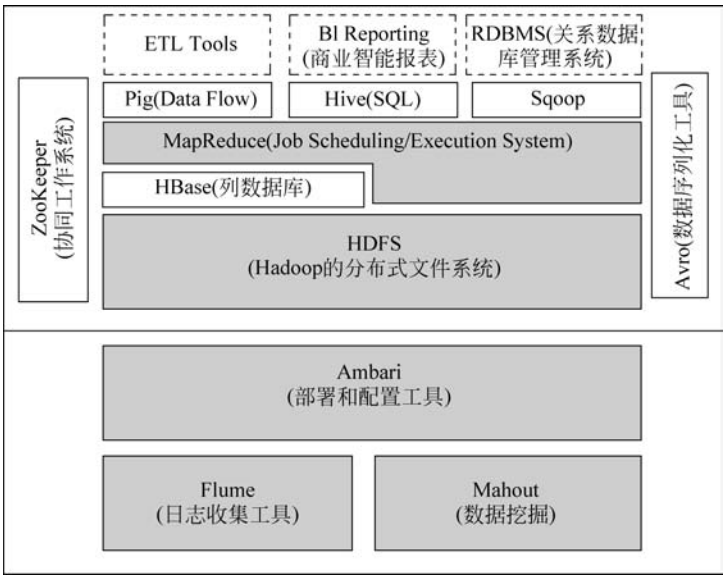

图 5-7 Hadoop 的生态系统

此生态系统提供了互补性服务或在核心层上提供了更高层的服务，使 Hadoop 的应用更加方便快捷。下面对 Hadoop 的生态系统进行简单介绍。

(1) Hive(基于 Hadoop 的数据仓库)。Hive 分布式数据仓库擅长数据展示，通常用于离线分析。Hive 管理存储在 HDFS 中的数据，提供了一种类似 SQL 的查询语言(HQL)查询数据。

(2) HBase(分布式列存储数据库)。HBase 是一个针对结构化数据的可伸缩、高可靠、高性能、分布式和面向列的动态模式数据库。和传统关系数据库不同，HBase 采用了谷歌 BigTable 的数据模型：增强的稀疏排序映射表(Key/Value)。其中，键由行关键字、列关键字和时间戳构成。HBase 可以对大规模数据进行随机、实时读写访问，同时，HBase 中保存的数据支持使用 MapReduce 来处理，它将数据存储和并行计算完美地结合在一起。

(3) ZooKeeper(分布式协作服务)。ZooKeeper 是协同工作系统，用于构建分布式应用，解决分布式环境下的数据管理问题，如统一命名、状态同步、集群管理、配置同步等。

(4) Sqoop(数据同步工具)。Sqoop 是 SQL-to-Hadoop 的缩写，是完成 HDFS 和关系数据库中的数据相互转移的工具。

(5) Pig(基于 Hadoop 的数据流系统)。Pig 提供相应的数据流语言和运行环境，实现数据转换(使用管道)和实验性研究(如快速原型)。适用于数据准备阶段，Pig 运行在由 Hadoop 基本架构构建的集群上。Hive 和 Pig 都建立在 Hadoop 基本架构之上，可以用来从数据库中提取信息，交给 Hadoop 处理。

(6) Mahout(数据挖掘算法库)。Mahout 的主要目标是实现一些可扩展的机器学习领域经典算法，旨在帮助开发人员更加方便、快捷地创建智能应用程序。Mahout 现在已经包含了聚类、分类、推荐引擎(协同过滤)和频繁集挖掘等广泛使用的数据挖掘方法。除了算法，Mahout 还包含数据的输入输出工具、与其他存储系统(如数据库、MongoDB 或 Cassandra)集成等数据挖掘支持架构。

(7) Flume(日志收集工具)。Flume 是 Cloudera 提供的一个高可用、高可靠、分布式的海量日志收集工具，即 Flume 支持在日志系统中定制各类数据发送方，用于收集数据；同时，Flume 提供对数据进行简单处理，并写到各种数据接收方(可定制)的能力。

(8) Avro(数据序列化工具)。Avro 是一种新的数据序列化(Serialization)格式和传输工具，设计用于支持大批量数据交换的应用。它的主要特点有：支持二进制序列化方式，可以便捷、快速地处理大量数据；动态语言友好，Avro 提供的机制使动态语言可以方便地处理数据。

(9) BI Reporting(Business Intelligence Reporting，商业智能报表)。BI Reporting 能提供综合报告、数据分析和数据集成等功能。

(10) RDBMS(关系数据库管理系统)。RDBMS 中的数据存储在被称为表(Table)的数据库中。表是相关记录的集合，由行和列组成，是一种二维关系表。

(11) ETL Tools。ETL Tools 是构建数据仓库的重要环节，由一系列数据仓库采集工具构成。

(12) Ambari。Ambari 旨在将监控和管理等核心功能加入 Hadoop。Ambari 可帮助系统管理员部署和配置 Hadoop、升级集群，并可提供监控服务。

4. Hadoop 的应用场景

Hadoop 比较擅长的是数据密集的并行计算，主要内容是对不同的数据做相同的事情，最后再整合，即 MapReduce 的映射规约。只要是数据量大、对实时性要求不高、数据块大的应用都适合使用 Hadoop 处理。具体应用场景有系统日志分析、用户习惯分析等。

5.3.2 Storm

Storm 是一个开源的、实时的计算平台，最初由工程师 Nathan Marz 编写，后来被 Twitter 收购并贡献给 Apache 软件基金会，目前已升级为 Apache 顶级项目。作为一个基于拓扑的流数据实时计算系统，Storm 简化了传统方法对无边界流式数据的处理过程，被广泛应用于实时分析、在线机器学习、持续计算、分布式远程调用等领域。Storm 的标志如图 5-8 所示。下面从 5 个方面介绍 Storm。

图 5-8 Storm 的标志

1. Storm 的特性

Storm 作为一个开源的分布式实时计算系统，可以简单、可靠地处理大量的数据流。和 Hadoop 中的 MapReduce 降低了并行批处理复杂性类似，Storm 降低了进行实时处理的复杂性，以便于程序开发人员迅速开发出实时处理数据的程序。Storm 支持水平扩展，具有高容错性，保证每个消息都会得到处理，且处理速度很快（在一个小集群中，每个节点每秒可以处理数以百万计的消息）。Storm 的部署和运维都很便捷，而且更为重要的是，可以使用任意编程语言来开发基于 Storm 的应用，这使得 Storm 成为当前大数据环境下非常流行的流数据实时计算系统。以下是 Storm 的特性。

（1）完整性。Storm 采用了 Acker 机制，保证数据不丢失；同时采用事务机制，保证数据的精确性。

（2）容错性。由于 Storm 的守护进程（Nimbus、Supervisor）都是无状态和快速恢复的，因此用户可以根据情况进行重启。当工作进程（Worker）失败或机器发生故障时，Storm 可自动分配新的 Worker 替换原来的 Worker，而且不会产生额外的影响。

（3）易用性。Storm 只需少量的安装及配置工作便可以进行部署和启动，并且进行开发时非常迅速，用户也易上手。

（4）免费和开源。Storm 是开源项目，同时，EPL 协议也是一个相对自由的开源协议，用户有权对自己的 Storm 应用开源或封闭。

（5）支持多种语言。Storm 使用 Clojure 语言开发，接口基本上都是由 Java 提供的，但 Storm 可以使用多种编程语言。并且 Storm 为多种编程语言实现了该协议的适配器，包括

Ruby、Python、PHP、Perl 等。

2. Storm 架构

Storm 采用的是主从架构模式(Master/Slave),主节点为 Nimbus,从节点为 Supervisor,其体系结构如图 5-9 所示。

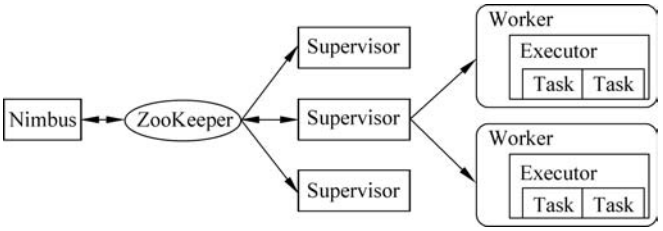

图 5-9　Storm 的体系结构

主节点 Nimbus 负责在集群分发任务(Topology)的代码以及监控等。在接收一个任务后,从节点 Supervisor 会启动一个或多个 Worker 来处理任务。所以实际上,任务最终都是分配到了 Worker 上。

3. Storm 的核心组件

Storm 的核心组件包括 Topology、Nimbus、Supervisor、Worker、Executor、Task、Spout、Bolt、Tuple、Stream、Stream 分组等,具体组件介绍如表 5-1 所示。

表 5-1　Storm 的核心组件介绍

组　　件	概　　念
Topology	一个实时计算应用程序,逻辑上被封装在 Topology 对象中,类似于 Hadoop 中的作业。不同的是,Topology 会一直运行到该进程结束
Nimbus	负责资源分配和任务调度,类似于 Hadoop 中的 JobTracker
Supervisor	负责接收 Nimbus 分配的任务,启动和停止管理的 Worker 进程,类似于 Hadoop 中的 TaskTracker
Worker	具体的逻辑处理组件
Executor	Storm 0.8 之后,Executor 是 Worker 进程中的具体物理进程,同一个 Spout/Bolt 的 Task 可能会共享一个物理进程,一个 Executor 中只能运行隶属于同一个 Spout/Bolt 的 Task
Task	每一个 Spout/Bolt 具体要做的工作内容,同时也是各个节点之间进行分组的单位
Spout	在 Topology 中产生数据源的组件。通常 Spout 获取数据源的数据,再调用 NextTuple()函数,发送数据供 Bolt 消费
Bolt	在 Topology 中接收 Spout 的数据,再执行处理的组件。Bolt 可以执行过滤、函数操作、合并、写数据库等操作。Bolt 接收到消息后调用 Execute()函数,用户可以在其中执行相应的操作
Tuple	消息传递的基本单元
Stream	源源不断传递的 Tuple 组成了 Stream,也就是数据流
Stream 分组	消息的分组方法。Storm 中提供若干实用的分组方式,包括 Shuffle、Fields、All、Global、None、Direct 和 Local or Shuffle 等

4. Storm 数据流

Storm 处理的数据被称为数据流(Stream)。数据流在 Storm 内各组件之间的传输形式是一系列元组(Tuple)序列,其传输过程如图 5-10 所示。每个 Tuple 内可以包含不同类型的数据,如 Int、String 等类型,但不同 Tuple 间对应位置上数据的类型必须一致,这是因为 Tuple 中数据的类型由各组件在处理前事先明确定义的。

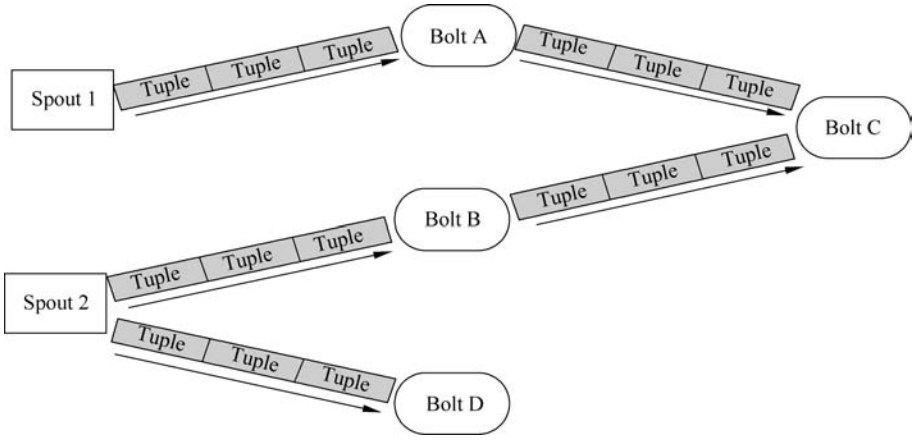

图 5-10 Storm 数据流的传输过程

Storm 集群中每个节点每秒可以处理成百上千个 Tuple,数据流在各个组件成分间类似于水流一样源源不断地从前一个组件流向后一个组件,而 Tuple 类似于承载数据流的管道。

5. Storm 的应用场景

Storm 的应用场景主要表现在信息流处理、连续计算和分布式远程程序调用等方面。

(1) 信息流处理(Stream Processing)。

Storm 可用来对数据进行实时处理并且及时更新数据库内容,同时兼具容错性和可扩展性。

(2) 连续计算(Continuous Computation)。

Storm 可进行连续查询并把结果即时反馈给客户端,例如在电子商务网站上实时搜索到商品信息等。

(3) 分布式远程程序调用(Distributed RPC)。

Storm 可用来并行处理密集查询,其拓扑结构是一个等待调用信息的分布函数,当它收到一条调用信息后,会对查询进行计算,并返回查询结果。

5.3.3 Spark

Spark 是由加州大学伯克利分校 AMP 实验室在 2009 年开发的,是一个开源的基于内存的大数据计算框架,保留了 Hadoop MapReduce 高容错和高伸缩的特性,不同的是 Spark 将中间结果保存在内存中,从而不再需要读写 HDFS,因此 Spark 能更好地适用于数据挖掘与机器学习等需要迭代的 MapReduce 模式的算法。另外,Spark 可以将 Hadoop 集群中的应用在内存中的运行速度提升约 100 倍,在磁盘上的运行速度提升约 10 倍。Spark 具有快

速、易用、通用、兼容性好 4 个特点，实现了高效的 DAG(Directed Acyclic Graph，有向无环图)执行引擎，支持通过内存计算高效处理数据流。可以使用 Java、Scala、Python、R 等语言轻松构建并行应用程序以及通过 Python、Scala 的交互式 Shell 在 Spark 集群中验证解决思路是否正确。Spark 的标志如图 5-11 所示。下面从 4 个方面介绍 Spark。

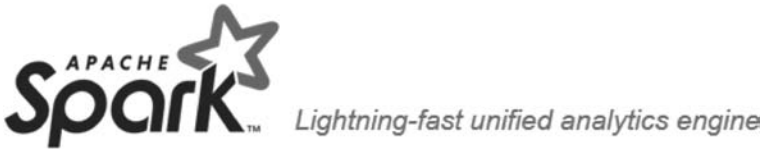

图 5-11　Spark 的标志

1. Spark 的特性

Spark 是一种基于内存的、分布式的、大数据处理框架，在 Hadoop 的强势之下，Spark 凭借快速、简洁易用、通用性以及支持多种运行模式 4 大特征，冲破固有思路，成为很多企业的标准大数据分析框架。作为现在主流的运算框架，Spark 有很多优点。

(1) 快速。

Spark 函数运行时，绝大多数函数是可以在内存中迭代的，只有少部分函数需要落地到磁盘。因此，Spark 与 MapReduce 相比，在计算性能上有显著的提升。

(2) 易用。

Spark 不仅计算性能突出，在易用性方面也是其他同类产品难以比拟的。一方面，Spark 提供了支持多种语言的 API，如 Scala、Java、Python、R 等，使用户开发 Spark 程序十分方便。另一方面，Spark 是基于 Scala 语言开发的，由于 Scala 是一种面向对象的、函数式的静态编程语言，因此其强大的类型推断、模式匹配、隐式转换等一系列功能，结合丰富的描述能力，使得 Spark 应用程序代码非常简洁。Spark 的易用性还体现在其针对数据处理提供了丰富的操作。

(3) 通用性。

Spark 没有出现时，进行计算需要安装 MapReduce，批处理需要安装 Hive，实时分析需要安装 Storm，机器学习需要安装 Mahout 或者 Mllib，查询需要安装 Hbase。Spark 出现后，计算时用 Spark Core，进行 SQL 操作时可以使用 Spark SQL，实时分析时就用 Spark Streaming。相对于第一代的大数据生态系统 Hadoop 中的 MapReduce，Spark 无论是在性能还是在方案的统一性方面，都有着极大的优势。

(4) 支持多种运行模式。

Spark 支持多种运行模式：本地(Local)运行模式、独立运行模式等。Spark 集群的底层资源可以借助于外部的框架进行管理，目前 Spark 对 Mesos 和 Yarn 提供了相对稳定的支持。在实际运行环境中，中小规模的 Spark 集群通常可满足一般企业绝大多数的业务需求，而在搭建此类集群时推荐采用 Standalone 模式(不采用外部的资源管理框架)。该模式使得 Spark 集群更加轻量级。

2. Spark 的体系架构

Spark 的体系架构包括 Spark Core 以及在 Spark Core 基础上建立的应用框架 Spark

SQL、Spark Streaming、MLlib、GraphX、Structured Streaming、SparkR。Spark Core 是 Spark 中最重要的部分，相当于 MapReduce，SparkCore 和 MapReduce 完成的都是离线数据分析。Core 库主要包括 Spark 的主要入口点（即编写 Spark 程序用到的第一个类）、整个应用的上下文（SparkContext）、弹性分布式数据集（RDD）、调度器（Scheduler）、对无规则的数据进行重组排序（Shuffle）和序列化器（Seralizer）等。Spark SQL 提供通过 Hive 查询语言（HiveQL）与 Spark 进行交互的 API，将 Spark SQL 查询转换为 Spark 操作，并且每个数据库表都被当作一个 RDD。Spark Streaming 对实时数据流进行处理和控制，允许程序能够像普通 RDD 一样处理实时数据。MLlib 是 Spark 提供的机器学习算法库。GraphX 提供了控制图、并行图操作和计算的算法和工具。Spark 的体系架构如图 5-12 所示。

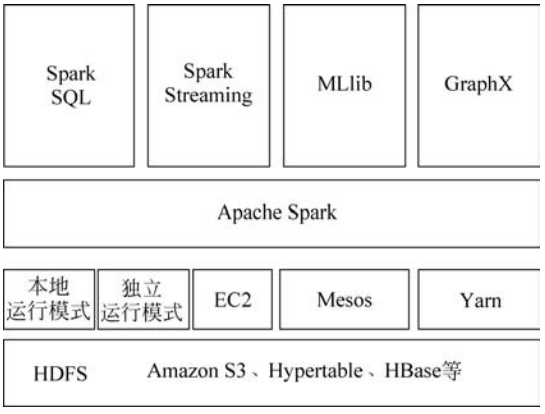

图 5-12　Spark 的体系架构

3. Spark 的扩展功能

目前 Spark 的生态系统以 Spark Core 为核心，然后在此基础上建立了处理结构化数据的 Spark SQL、对实时数据流进行处理的 Spark Streaming、用于图计算的 GraphX、机器学习算法库 MLlib 4 个子框架，如图 5-13 所示。整个生态系统实现了在一套软件栈内完成各种大数据分析任务的目标。下面分别对这 4 个子框架进行简单介绍。

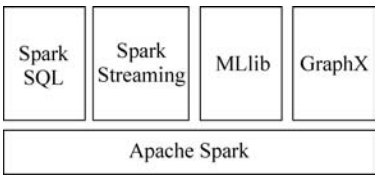

图 5-13　Spark 的生态系统

1）Spark SQL

Spark SQL 的主要功能是分析处理结构化数据，可以随时查看数据结构和正在执行的运算信息。Spark SQL 是 Spark 用来处理结构化数据的一个模块，不同于 Spark RDD 的基本 API，Spark SQL 接口提供了更多关于数据结构和正在执行的计算结构的信息，并利用这些信息去更好地进行优化。

2）Spark Streaming

Spark Streaming 用于处理流式计算，是 Spark 核心 API 的一个扩展。它支持可伸缩、高吞吐量、可容错的处理实时数据流，能够和 Spark 的其他模块无缝集成。Spark Streaming 支持从多种数据源获取数据，如 Kafka、Flume、Kinesis 和 HDFS 等，获取数据后可以通过 Map()、Reduce()、Join()和 Window()等高级函数对数据进行处理。最后还可以将处理结果推送到文件系统、数据库等。

3) MLlib

MLlib(Machine Learning lib)是 Spark 的机器学习算法的实现库,同时包括相关的测试和数据生成器,专为在集群上并行运行而设计,旨在使机器学习变得可扩展和更容易实现。MLlib 目前支持常见的机器学习算法,还包括了底层的优化原语和高层的管道 API。

4) GraphX

GraphX 是 Spark 中用于图形并行计算的组件,通过引入核心抽象 Resilient Distributed Property Graph(一种点和边都带属性的有向多重图)扩展了 Spark RDD 这种抽象的数据结构。Property Graph 有 Table 和 Graph 两种视图,但只有一份物理存储,物理存储由 VertexRDD 和 EdgeRDD 这两个 RDD 组成。这两种视图都有自己独有的操作符,从而使操作更加灵活,提高了执行效率。

当一个图的规模非常大时,就需要使用分布式图计算框架。与其他分布式图计算框架相比,GraphX 最大的贡献是在 Spark 中提供了一站式数据解决方案,可以方便且高效地完成图计算的一整套流水作业。GraphX 采用分布式框架的目的是将对巨型图的各种操作包装成简单的接口,从而在分布式存储、并行计算等复杂问题上对上层透明,从而使得开发者可以更加聚焦在图计算相关的模型设计和使用上,而不用关心底层的分布式细节,极大地满足了对分布式图处理的需求。

4. Spark 的应用场景

Spark 是专为大规模数据处理而设计的快速通用的计算引擎,现已形成一个高速发展、应用广泛的生态系统,其主要应用场景如下。

(1) 多次操作特定数据集的应用场合。Spark 是基于内存的迭代计算框架,适用于需要多次操作特定数据集的应用场合。需要反复操作的次数越多,所需读取的数据量越大,受益越大;数据量小但是计算密集度较大的场合,受益就相对较小。

(2) 粗粒度更新状态的应用。由于 RDD 的特性,Spark 不适用那种异步细粒度更新状态的应用,例如 Web 服务的存储或者增量的 Web 爬虫和索引,就是对于那种增量修改的应用模型不适合。

(3) 数据量不是特别大,但是适合实时统计分析的需求应用。

目前,大数据技术在互联网公司主要应用在广告、报表、推荐系统等业务上。在广告业务方面需要大数据技术做应用分析、效果分析、定向优化等;在推荐系统方面则需要大数据技术优化相关排名、个性化推荐以及热点分析等。这些应用场景的普遍特点是计算量大、效率要求高,Spark 恰恰可以满足这些要求。

5.3.4　Hadoop、Spark 与 Storm 的比较

处理框架是大数据系统一个最基本的组件,负责对系统中的数据进行计算。按照处理类型的不同,可分为批处理框架、流处理框架和混合框架。

Hadoop 是一种专用于批处理的处理框架。Hadoop 重新实现了相关算法和组件堆栈,让大规模批处理技术变得更易用。Hadoop 包含多个组件,通过配合使用可处理批数据。

Hadoop 的处理功能来自 MapReduce 引擎。MapReduce 的处理技术符合使用键值对的 Map、Shuffle、Reduce 算法要求。由于这种方法严重依赖持久存储,每个任务需要多次执行读取和写入操作,因此速度相对较慢。但磁盘空间通常是服务器上最丰富的资源,

这意味着 MapReduce 可以处理非常海量的数据集，同时也意味着相比其他类似技术，Hadoop 的 MapReduce 通常可以在廉价硬件上运行，因为该技术并不需要将一切都存储在内存中。而且 MapReduce 具备极高的缩放潜力，实际生产中曾经出现过包含数万个节点的应用。

Storm 是一种侧重于极低延迟的流处理框架，也是实时处理领域的最佳解决方案。该技术可处理非常大量的数据，通常比其他解决方案更低的延迟提供结果。如果处理速度直接影响用户体验，如需要将处理结果直接提供给访客打开的网站页面，此时 Storm 是一个很好的选择。Storm 与 Trident 配合使得用户可以用微批代替纯粹的流处理。

在操作性方面，Storm 可与 Hadoop 的 YARN 资源管理器进行集成，因此可以很方便地融入现有 Hadoop 部署。除了支持大部分处理框架，Storm 还可支持多种语言，为用户的拓扑定义提供了更多选择。

Spark 是一种包含流处理能力的下一代批处理框架。与 Hadoop 的 MapReduce 引擎基于各种相同原则开发相比，Spark 主要侧重于通过完善的内存计算和处理优化机制加快批处理工作负载的运行速度。Spark 可作为独立集群部署（需要相应存储层的配合），或可与 Hadoop 集成并取代 MapReduce 引擎。

Spark 在内存中处理数据的方式已大幅改善，而且，Spark 在处理与磁盘有关的任务时速度也有很大提升，因为通过提前对整个任务集进行分析可以实现更完善的整体式优化。为此 Spark 可创建代表所需执行的全部操作、需要操作的数据，以及操作和数据之间关系的有向无环图（Directed Acyclic Graph，DAG）。

Spark 相对于 Hadoop 或 MapReduce 的主要优势是速度。在内存计算策略和先进的 DAG 调度等机制的帮助下，Spark 可以用更快的速度处理相同的数据集。Spark 的另一个重要优势在于多样性。该产品可作为独立集群部署，或与现有 Hadoop 集群集成。该产品可运行批处理和流处理，运行一个集群即可处理不同类型的任务。除了引擎自身的能力外，围绕 Spark 还建立了包含各种库的生态系统，可为机器学习、交互式查询等任务提供更好的支持。相比于其他框架，Spark 任务的代码也更易于编写，因此可大幅提高生产力。

Hadoop、Storm 与 Spark 这三种框架，每个框架都有自己的最佳应用场景。所以，在不同的应用场景下，应该选择不同的框架。Spark 是多样化工作负载处理任务的最佳选择。Spark 批处理能力以更高内存占用为代价提供了无与伦比的速度优势。对于重视吞吐率而非延迟的工作负载，则比较适合使用 Spark Streaming 作为流处理解决方案。Hadoop 及其 MapReduce 处理引擎提供了一套久经考验的批处理模型，最适合处理对时间要求不高的非常大规模数据集，通过非常低成本的组件即可搭建完整功能的 Hadoop 集群，使得这一廉价且高效的处理技术可以灵活应用在很多案例中。与其他框架和引擎的兼容与集成能力使得 Hadoop 可以成为使用不同技术的多种工作负载处理平台的底层基础。对于延迟需求很高的纯粹的流处理工作负载，Storm 可能是最适合的技术。该技术可以保证每条消息都被处理，可配合多种编程语言使用。由于 Storm 无法进行批处理，因此如果需要这些能力可能还需要使用其他软件。如果对严格的一次处理保证有比较高的要求，此时可考虑使用 Storm 与 Trident 配合处理。不过在这种情况下，其他流处理框架也许更适合。

5.4　云计算

目前，云计算已经成为推动社会生产力变革的新生力量。从技术上看，大数据与云计算的关系就像一枚硬币的正反面一样密不可分。大数据的主要任务是对海量数据进行分析，它需要通过云计算的分布式处理、分布式数据库、云存储和虚拟化技术来实现；云计算为大数据提供了弹性可拓展的基础设施，也是产生大数据的平台之一。

5.4.1　云计算的概念与特点

2006 年，谷歌公司 CEO 埃里克·施密特(Eric Schmidt)在搜索引擎大会上首次提出“云计算”的概念及体系架构，并快速得到了业界认可。2008 年，“云计算”概念全面进入中国。2009 年，中国首届云计算大会召开，此后云计算技术和产品迅速发展起来。

到目前为止，云计算并没有一个统一的定义，业界对云计算的定义达 100 多种，下面列出两个经典的云计算的概念。

(1) 维基百科：云计算是一种动态扩展的计算模式，通过网络将虚拟化的资源作为服务提供给用户。

(2) 美国国家标准与技术研究院(National Institute of Standards and Technology，NIST)：云计算是一种无处不在的、便捷的通过互联网访问的一个可定制的 IT 资源(IT 资源包括网络、服务器、存储、应用软件和服务)共享池，是一种按使用量付费的模式。它能够通过最少量的管理或与服务供应商的互动实现计算资源的迅速供给和释放。这也是现阶段广为接受的云计算的定义。

简而言之，云计算是一种通过互联网以服务的方式提供动态可伸缩的虚拟化资源的计算模式。云计算的资源是分布式架构，通过虚拟化技术实现动态易扩展。终端用户不需要了解“云”中基础设施的细节，不必具有相应的专业知识，也无须直接进行控制，只关注自己真正需要什么样的资源以及如何通过网络来得到相应的服务。与传统计算机系统相比，云计算具有如下特点。

(1) 超大规模。目前，谷歌云计算已经拥有服务器达 100 多万台，Amazon、IBM、Yahoo 等公司也都分别拥有数十万台服务器。众多服务器给用户提供了强悍的计算能力。

(2) 虚拟化。云计算支持用户在云端覆盖的范围内随时随地、使用各种各样的终端获取云服务。用户所请求的资源来自“云”，而不是固定的物理实体，用户也无须了解应用运行的具体位置。

(3) 高可靠性。云计算使用数据多副本容错、计算节点同构可互换等措施，从而有效地保障了云计算的可靠性。

(4) 通用性。在“云”的支撑下，人们可以构造出各种各样的应用，同一个“云”也可以同时支撑不同的应用同时运行。

(5) 高可扩展性。云的规模可以弹性伸缩，满足应用和用户规模增长的需要。

(6) 按需服务。“云”是一个庞大的资源池，用户可根据自己的需要自行决定购买哪些服务。

(7) 极其廉价。不采用复杂而昂贵的节点进行构建，成本低廉。其自动化集中式管理

使大量企业无须负担日益高昂的数据中心管理成本。再加上“云”的强通用性使资源的利用率大幅度提升。

5.4.2 云计算的主要部署模式

2011 年,在美国《联邦云计算战略报告》中定义了 4 种云:公有云、私有云、混合云和社区云,国内将社区云改成了联合云。社区云应用比较窄,它是面向社团组织内用户的云计算服务平台。所以根据实际情况,云计算按部署模式主要可分为公有云、私有云和混合云,如图 5-14 所示。

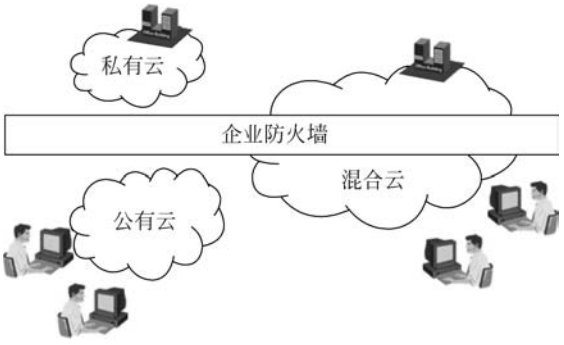

图 5-14 云计算的主要部署模式

1. 公有云

公有云提供面向社会大众、公共群体的云计算服务。公有云用户以付费的方式,根据业务需要弹性使用 IT 分配的资源,用户不需要自己构建硬件、软件等基础设施和后期维护,可以在任何地方、任何时间、多种方式,以互联网的形式访问获取资源。公有云如同日常生活中按需购买使用的水、电一样,人们可以方便、快捷地享受服务。

目前,比较流行的公有云平台有国外的亚马逊云平台 AWS(Amazon Web Services)、GAE(Google App Engine)等,国内的阿里云、SAE(Sina App Engine)、BAE(Baidu App Engine)等。亚马逊的 AWS 提供了大量基于云的全球性产品,包括计算、存储、数据库、分析、联网、移动产品、开发人员工具、管理工具、物联网、安全性和企业级应用程序。这些服务及应用程序可帮助企业或组织快速发展自己的业务、降低 IT 成本,使来自中国乃至全球的众多客户从中获益。公有云有很多优点,但最大的缺点是难以保证数据的私密性。

2. 私有云

私有云提供面向应用行业/组织内的云计算服务。私有云一般由一个组织来使用,同时由这个组织来运营,如政府机关、移动通信、学校等内部使用的云平台。私有云可较好地解决数据私密性问题,对移动通信、公安等数据私密性要求特别高的企业或机构,建设私有云将是一个必然的选择。

使用私有云提供的云计算服务需要一定的权限,一般只提供给企业内部员工使用。其主要目的是合理地组织企业已有的软硬件资源,提供更加可靠、弹性的服务供企业内部使用。比较流行的私有云平台有 VMware vCloud Suite 和微软公司的 Microsoft System Center 2016。

3. 混合云

混合云是把公有云和私有云进行整合，取二者的优点，是近年来云计算的主要模式和发展方向。私有云主要面向企业用户，出于安全考虑，企业更愿意将数据存放在私有云中，但是同时又希望可以获得公有云的计算资源。在这种情况下，混合云越来越多地被采用。它对公有云和私有云进行融合和匹配，以获得更佳的效果。这种个性化的解决方案，达到了既省钱又安全的目的，给企业带来真正意义上的云计算服务。混合云是未来云发展的方向。混合云既能利用企业在 IT 基础设施的巨大投入，又能解决公有云带来的数据安全等问题，是避免企业变成信息孤岛的最佳解决方案。混合云强调基础设施是由两种或多种云组成的，但对外呈现的是一个完整的整体。企业正常运营时，把重要数据保存在自己的私有云中（如财务数据），把不重要的信息或需要对公众开放的信息放到公有云中。

组建混合云的利器是 OpenStack，它可以把各种云计算平台资源进行异构整合，构建企业级混合云，使企业可以根据自己的需求灵活自定义各种云计算服务。在搭建企业云计算平台时，使用 OpenStack 架构是最理想的解决方案，虽然入门门槛较高，但是随着项目规模的扩大，企业终将从中受益，因为不必支付云平台中软件的购买费用。

混合云计算的典型案例是 12306 火车票购票网站。12306 购票网站最初是私有云计算，消费者平时用 12306 购票没有问题，但是一到节假日（如春节）有大量购票需求时，消费者在购票时就会出现页面响应慢或者页面报错的情况，甚至还会出现无法付款的情况，用户体验特别差。为了解决上述问题，12306 火车购票网站与阿里云签订战略合作，由阿里云提供计算能力以满足业务高峰期查票检索服务，而支付业务等关键业务在 12306 自己的私有云环境之中运行。两者组合成一个新的混合云，对外呈现的还是一个完整的系统。

5.4.3　云计算的主要服务模式

云计算的主要服务模式有 3 类：基础即服务（Infrastructure as a Service，IaaS）、平台即服务（Platform as a Service，PaaS）和软件即服务（Software as a Service，SaaS），如图 5-15 所示。IaaS 侧重于硬件资源服务，注重计算资源的共享，消费者通过互联网可以从完善的计算机基础设施中获得服务；PaaS 侧重于平台服务，以服务平台或者开发环境提供服务；SaaS 侧重于软件服务，通过网络提供软件程序服务。

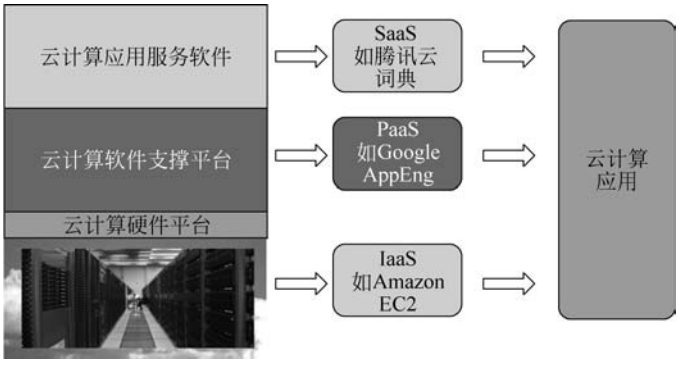

图 5-15　云计算的 3 类服务模式

(1) IaaS。

该层指云计算服务商提供虚拟的硬件资源,用户通过网络租赁即可搭建自己的应用系统。IaaS 属于底层,向用户提供可快速部署、按需分配、按需付费的高安全与高可靠的计算能力,并向用户提供存储能力的租用服务,还可为应用提供开放的云服务接口。用户可以根据业务需求,灵活租用相应的云基础资源。

(2) PaaS。

该层将云计算应用程序开发和部署的平台作为一种服务提供给客户。该服务包括应用设计、应用开发、应用测试和应用托管等。开发者只需要上传代码和数据就可以使用云服务,而不需要关心底层的具体实现方式和管理模式。

(3) SaaS。

该层通过部署硬件基础设施对外提供服务。用户可以根据各自的需求购买虚拟或实体的计算、存储、网络等资源。用户可以在购买的空间内部署和运行包括操作系统和应用程序在内的软件,而不需管理或控制任何云计算基础设施(事实上也不能管理或控制),但用户可以选择操作系统、存储空间并部署自己的应用,也可以控制有限的网络组件(如防火墙、负载均衡器等)。

无论是 IaaS、PaaS 还是 SaaS,其核心理念都是为用户提供按需服务。总体上来说,云计算通过互联网将超大规模的计算与存储资源整合起来,再以可信任服务的形式按需提供给用户。

5.4.4 云计算的主要技术

从技术角度看,云计算包含云设备和云服务两个部分。云设备包含用于弹性计算的服务器设备,用于数据保存的存储设备,用于数据通信的网络设备和其他用于云安全、互联网应用、负载均衡等硬件设备。云服务包含用于物理资源虚拟化调度管理的弹性计算服务、大数据处理服务、网络应用服务、数据安全服务、应用程序接口服务、自动化运维和管理服务等。总体来说,云计算的主要技术有以下 3 种。

(1) 虚拟化技术。

虚拟化指计算单元不在真实的单元上而在虚拟的单元上运行,是一种优化资源和简化管理的计算方案。虚拟化技术适合在云计算平台中应用,虚拟化的核心解决了云计算等对硬件的依赖,提供统一的虚拟化界面;通过虚拟化技术,人们可以在一台服务器上运行多台虚拟机,从而实现了对服务器的优化和整合。虚拟化技术使用动态资源伸缩的手段,降低了云计算基础设施的使用成本,并提高负载部署的灵活性。

(2) 中间件技术。

支持应用软件的开发、运行、部署和管理的支撑软件被称为中间件。中间件是运行在两个层次之间的一种组件,是在操作系统和应用软件之间的软件层次。中间件可以屏蔽硬件和操作系统之间的兼容问题,并具有管理分布式系统中的节点间的通信、节点资源和协调工作等功能。通过中间件技术,人们可将不同平台的计算节点组成一个功能强大的分布式计算系统。而云环境下的中间件技术,其主要功能是对云服务资源进行管理,包含用户管理、任务管理、安全管理等,为云计算的部署、运行、开发和应用提供高效支撑。

（3）云存储技术。

在云计算中，云存储技术通常和虚拟化技术相互结合起来，通过对数据资源的虚拟化，提高访问效率。目前，数据存储技术 HDFS 和谷歌公司的 GFS 具有高吞吐率、分布式和高速传输等优点，因此，采用云存储技术，可满足云计算为大量用户提供云服务的需求。

5.4.5　云计算与大数据的关系

大数据复杂的需求对技术实现和底层计算资源都提出了更高的要求，而云计算所具备的弹性伸缩、动态调配、资源虚拟化、支持多租户、支持按量计费或按需使用及绿色节能等基本要素，正好契合了新型大数据处理技术的需求，也正在成为解决大数据问题的未来计算技术发展的重要方向。大数据与云计算的关系如图 5-16 所示。

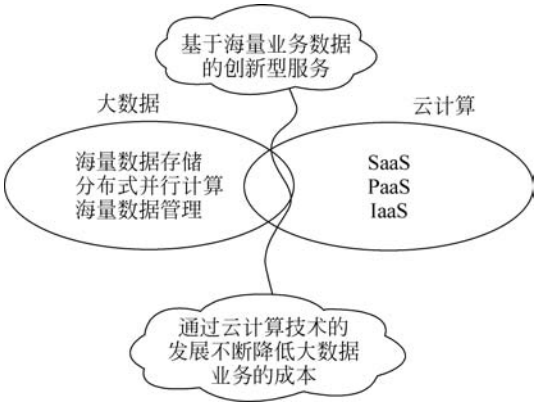

图 5-16　大数据与云计算的关系

云计算的实质是服务，是一种新兴的商业计算模式。“云”概念的提出是因为它的规模很大，可以根据业务动态伸缩。云计算是提供给这种商业模式的具体实现，是互联网产业发展到一定阶段的必然产物。云计算与大数据是一对相辅相成的概念，它们描述了面向数据时代信息技术的两个方面，云计算侧重于提供资源和应用的网络化交付方法，大数据侧重于应对数据量巨大所带来的技术挑战。

云计算的核心是业务模式，其本质是数据处理技术。数据是资产，云计算为数据资产提供了存储、访问的场所和计算能力，即云计算更偏重海量数据的存储和计算，以及提供云计算服务，运行云应用。但是云计算缺乏盘活数据资产的能力，挖掘价值性信息和进行预测性分析，为国家治理、企业决策乃至个人生活服务，这是大数据的核心议题。云计算是基础设施架构，大数据是思想方法，大数据技术将帮助人们从大体量、高度复杂的数据中分析、挖掘信息，从而发现价值和预测趋势。

知识巩固与技能训练

一、名词解释

1. Hadoop　2. Storm　3. Spark　4. IaaS　5. PaaS　6. SaaS

二、思考题

1. 简述大数据的四层堆栈式技术架构。
2. 简述 HDFS 存储数据的优点。
3. 简述 Hadoop 生态系统包括哪些组成部分，以及各部分对应的功能。
4. 根据自己的理解，说一说云计算有什么特点。
5. 简述云计算的 3 种主要服务模式。
6. 简述云计算与大数据之间的关系。

三、网络搜索和浏览

结合查阅的相关文献资料，列举 Hadoop、Spark 与 Storm 的适用场景。