第1章

绪　论

本章主要阐述大数据技术的概念、三种典型的大数据平台(Hadoop、Spark 和 Storm)和发展趋势。

1.1　大数据技术概述

大数据(big data)是指无法在一定时间范围内用常规软件工具进行捕捉、管理和处理的数据集合,是需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力的海量、高增长率和多样化的信息资产。

1.1.1　大数据的特点

大数据的 5V 特点(IBM 公司提出):Volume(大量)、Velocity(高速)、Variety(多样)、Value(低价值密度)、Veracity(真实性)。

(1) Volume(大量):大数据中数据的采集、存储和计算的量都非常大。一般认为,只有起始计量单位达到 PB 的数据才可以被称为大数据(1PB=1024TB=1048576GB)。

(2) Velocity(高速):由于数据具有时效性,超出一定的时间就会失去其作用,需要尽可能实时地完成对海量数据处理。这是大数据区别于传统数据挖掘的显著特征。

(3) Variety(多样):包括结构化、半结构化和非结构化数据。随着互联网和物联网的发展,又扩展到网页、社交媒体、感知数据,涵盖音频、图片、视频、模拟信号等,真正诠释了数据的多样性,也对数据的处理能力提出了更高的要求。

(4) Value(低价值密度):随着互联网以及物联网的广泛应用,信息感知无处不在,数据量无比庞大,但价值密度较低。如何在海量数据中获得有价值的信息,是大数据时代最需要解决的问题。

(5) Veracity(真实性):大数据中的内容是与真实世界中的发生息息相关的,要保证数据的准确性和可信赖度。研究大数据就是从庞大的网络数据中提取出能够解释和预测现实事件的过程。

1.1.2　大数据与数据科学的关系

数据科学是研究探索赛博空间(Cyberspace)中数据界(datanature)奥秘的理论、方法和技术,研究的对象是数据界中的数据。数据科学的研究对象是 Cyberspace 的数据,是新的科学。数据科学主要有两个内涵:一个是研究数据本身,研究数据的各种类型、状态、属性

及变化形式和变化规律；另一个是为自然科学和社会科学研究提供的一种新的方法，称为科学研究的数据方法，其目的在于揭示自然界和人类行为现象和规律。

大数据和数据科学的关系可以用图 1-1 进行说明。数据科学是作为一个与大数据相关的新兴学科出现的，在大数据处理的理论研究方面，新型的概率和统计模型将是主要的研究工具。数据科学基础问题体系本身就是大数据领域的研究热点。同时，数据科学将带动多学科融合。

图 1-1　具有多学科交叉特点的数据科学

1.1.3　大数据的关键技术

Google 公司的三篇论文奠定了大数据技术与算法的基础。

2003 年，Google 公司发布 Google File System 论文，这是一个可扩展的分布式文件系统，用于大型的、分布式的、对大量数据进行访问的应用，运行于廉价的普通硬件上，提供容错功能。从根本上说：文件被分割成很多块，使用冗余的方式储存于商用机器集群上。

2004 年，Google 公司发布 MapReduce 论文，描述了大数据的分布式计算方式，主要思想是将任务分解后在多台处理能力较弱的计算节点中同时处理，然后将结果合并，从而完成大数据处理。

2006 年，Google 公司发布 Bigtable 论文，启发了无数的 NoSQL 数据库，如 Cassandra、HBase、MongoDB 等。

大数据技术，就是从各种类型的数据中快速获得有价值信息的技术。大数据领域已经涌现出了大量新的技术，它们成为大数据采集、存储、处理和呈现的有力武器。大数据处理关键技术一般包括：大数据采集、大数据预处理、大数据存储及管理、大数据分析及挖掘、大数据展现和应用(大数据检索、大数据可视化、大数据应用、大数据安全等)。

1. 大数据采集技术

数据采集是指通过 RFID 射频数据、传感器数据、社交网络交互数据及移动互联网数据等方式获得的各种类型的结构化、半结构化(或称为弱结构化)及非结构化的海量数据，是大数据知识服务模型的根本。重点要突破分布式高速高可靠数据爬取或采集、高速数据全映像等大数据收集技术；突破高速数据解析、转换与装载等大数据整合技术；设计质量评估

模型，开发数据质量技术。

大数据采集一般分为大数据智能感知层和基础支撑层。大数据智能感知层主要包括数据传感体系、网络通信体系、传感适配体系、智能识别体系及软硬件资源接入系统，实现对结构化、半结构化、非结构化的海量数据的智能化识别、定位、跟踪、接入、传输、信号转换、监控、初步处理和管理等，必须着重攻克针对大数据源的智能识别、感知、适配、传输、接入等技术；基础支撑层提供大数据服务平台所需的虚拟服务器，结构化、半结构化及非结构化数据的数据库及物联网络资源等基础支撑环境，重点攻克分布式虚拟存储技术，大数据获取、存储、组织、分析和决策操作的可视化接口技术，大数据的网络传输与压缩技术，大数据隐私保护技术等。

2. 大数据预处理技术

大数据预处理技术主要完成对已接收数据的抽取、清洗等操作。

(1) 抽取：因获取的数据可能具有多种结构和类型，数据抽取过程可以帮助我们将这些复杂的数据转化为单一的或者便于处理的构型，以达到快速分析处理的目的。

(2) 清洗：对于大数据，并不全是有价值的，有些数据并不是我们所关心的内容，而另一些数据则是完全错误的干扰项，因此要对数据通过过滤“去噪”而提取出有效数据。

3. 大数据存储及管理技术

大数据存储与管理要用存储器把采集到的数据存储起来，建立相应的数据库，并进行管理和调用；重点解决复杂结构化、半结构化和非结构化大数据管理与处理技术；主要解决大数据的可存储、可表示、可处理、可靠性及有效传输等几个关键问题；开发可靠的分布式文件系统(DFS)、能效优化的存储、计算融入存储、大数据的去冗余及高效低成本的大数据存储技术；突破分布式非关系型大数据管理与处理技术、异构数据的数据融合技术、数据组织技术，研究大数据建模技术；突破大数据索引技术；突破大数据移动、备份、复制等技术。

开发新型数据库技术，数据库分为关系型数据库、非关系型数据库及数据库缓存系统。其中，非关系型数据库主要指的是 NoSQL 数据库，分为键值数据库、列存数据库、图存数据库以及文档数据库等类型。关系型数据库包含了传统关系数据库系统以及 NewSQL 数据库。

开发大数据安全技术。改进数据销毁、透明加解密、分布式访问控制、数据审计等技术；突破隐私保护和推理控制、数据真伪识别和取证、数据持有完整性验证等技术。

4. 大数据分析及挖掘技术

大数据分析技术包括改进已有数据挖掘和机器学习技术；开发数据网络挖掘、特异群组挖掘、图挖掘等新型数据挖掘技术；突破基于对象的数据连接、相似性连接等大数据融合技术；突破用户兴趣分析、网络行为分析、情感语义分析等面向领域的大数据挖掘技术。

数据挖掘是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中，提取隐含在其中的，人们事先不知道的，但又是潜在有用的信息和知识的过程。数据挖掘涉及的技术方法很多，有多种分类法。

(1) 根据数据挖掘任务可分为分类或预测模型发现、数据总结、聚类、关联规则发现、序列模式发现、依赖关系或依赖模型发现、异常和趋势发现等。

(2) 根据数据挖掘对象可分为关系数据库、面向对象数据库、空间数据库、时态数据库、文本数据源、多媒体数据库、异质数据库、遗产数据库，以及万维网(Web)。

(3) 根据数据挖掘方法,可粗分为机器学习方法、统计方法、神经网络方法和数据库方法。

从数据挖掘任务和数据挖掘方法的角度,大数据分析及挖掘技术着重突破以下几方面。

(1) 可视化分析。数据可视化无论对于普通用户还是数据分析专家,都是最基本的功能。数据图像化可以让数据自己说话,让用户直观地感受到结果。

(2) 数据挖掘算法。图像化是将机器语言翻译给人看,而数据挖掘就是机器的母语。分割、集群、孤立点分析还有各种各样五花八门的算法让我们精炼数据,挖掘价值。这些算法一定要能够应付大数据的量,同时还具有很高的处理速度。

(3) 预测性分析。预测性分析可以让分析师根据图像化分析和数据挖掘的结果做出一些前瞻性判断。

(4) 语义引擎。语义引擎需要设计到有足够的人工智能以足以从数据中主动地提取信息。语言处理技术包括机器翻译、情感分析、舆情分析、智能输入、问答系统等。

(5) 数据质量和数据管理。数据质量与数据管理是管理的最佳实践,通过标准化流程和机器对数据进行处理可以确保获得一个预设质量的分析结果。

5. 大数据展现与应用技术

大数据技术能够将隐藏于海量数据中的信息和知识挖掘出来,为人类的社会经济活动提供依据,从而提高各个领域的运行效率,大幅提高整个社会经济的集约化程度。

在我国,大数据将重点应用于以下三大领域:商业智能、政府决策、公共服务。例如,商业智能技术,政府决策技术,电信数据信息处理与挖掘技术,电网数据信息处理与挖掘技术,气象信息分析技术,环境监测技术,警务云应用系统(道路监控、视频监控、网络监控、智能交通、反电信诈骗、指挥调度等公安信息系统),大规模基因序列分析比对技术,Web 信息挖掘技术,多媒体数据并行化处理技术,影视制作渲染技术,其他各种行业的云计算和海量数据处理应用技术等。

1.1.4 大数据的计算模式

基于大数据的分布式计算模式主要有 4 种,见表 1-1。

表 1-1 典型的大数据计算模式

大数据计算模式	解决问题	代表产品
批处理计算	针对大规模数据的批量处理	Hadoop/MapReduce、Spark 等
流计算	针对流数据的实时计算	Storm、S4、Flume、Streams、Puma、DStream、Super Mario、银河流数据处理平台等
图计算	针对大规模图结构数据的处理	Pregel、GraphX、Giraph、PowerGraph、Hama、GoldenOrb 等
查询分析计算	大规模数据的存储管理和查询分析	Dremel、Hive、Cassandra、Impala 等

1.2 基于 Hadoop 系统的大数据平台

Hadoop 是一个由 Apache 基金会所开发的分布式系统基础架构。用户可以在不了解分布式底层细节的情况下开发分布式程序,充分利用集群的威力进行高速运算和存储。

Hadoop 是一个能够让用户轻松架构和使用的分布式计算平台。用户可以轻松地在 Hadoop 上开发和运行处理海量数据的应用程序。

Hadoop 的官网是 http://hadoop.apache.org/，主界面如图 1-2 所示。截止到 2020 年 8 月，Hadoop 的最新版本是 3.3.0。

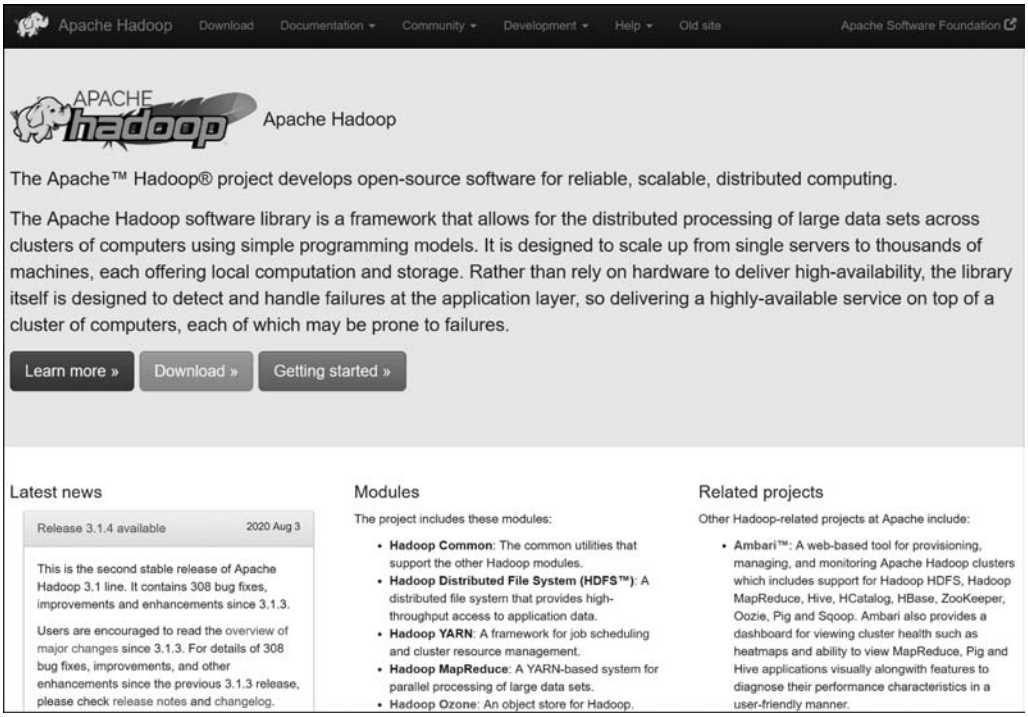

图 1-2　Hadoop 的官网主页信息

1.2.1　Hadoop 的特点

Hadoop 具有以下五个优点。

- 高可靠性：Hadoop 按位存储和处理数据的能力值得人们信赖。
- 高扩展性：Hadoop 是在可用的计算机集簇间分配数据并完成计算任务的，这些集簇可以方便地扩展到数以千计的节点中。
- 高效性：Hadoop 能够在节点之间动态地移动数据，并保证各个节点的动态平衡，因此处理速度非常快。
- 高容错性：Hadoop 能够自动保存数据的多个副本，并且能够自动将失败的任务重新分配。
- 低成本：与一体机、商用数据仓库以及 QlikView、Yonghong Z-Suite 等数据集市相比，Hadoop 是开源的，项目的软件成本因此会大幅降低。

Hadoop 被公认为行业大数据标准开源软件，在分布式环境下提供了海量数据的处理能力。几乎所有主流厂商都围绕 Hadoop 提供开发工具、开源软件、商业化工具和技术服务，如谷歌、雅虎、微软、思科、淘宝等都支持 Hadoop。

1.2.2 Hadoop 的生态系统

Hadoop2. x 相比较于 Hadoop1. x 来说，HDFS 的架构与 MapReduce 的都有较大的变化，且速度上和可用性上都有了很大的提高。Hadoop2. x 有两个重要的变更。

（1）HDFS 的 NameNodes 可以以集群的方式部署，增强了 NameNodes 的水平扩展能力和可用性。

（2）MapReduce 在 Hadoop2. x 中称为 MR2 或 YARN，将 JobTracker 中的资源管理及任务全生命周期管理（包括定时触发及监控），拆分成两个独立的组件，用于管理全部资源的 ResourceManager 以及管理每个应用的 ApplicationMaster。

ResourceManager 用于管理向应用程序分配计算资源，每个 ApplicationMaster 用于管理应用程序、调度以及协调。一个应用程序可以是经典的 MapReduce 架构中的一个单独的任务，也可以是这些任务的一个 DAG（有向无环图）任务。ResourceManager 及每台计算机上的 NodeManager 服务，用于管理那台计算机的用户进程，形成计算架构。每个应用程序的 ApplicationMaster 实际上是一个框架具体库，并负责从 ResourceManager 中协调资源及与 NodeManager（s）协作执行并监控任务。

Hadoop2. x 的生态系统如图 1-3 所示。

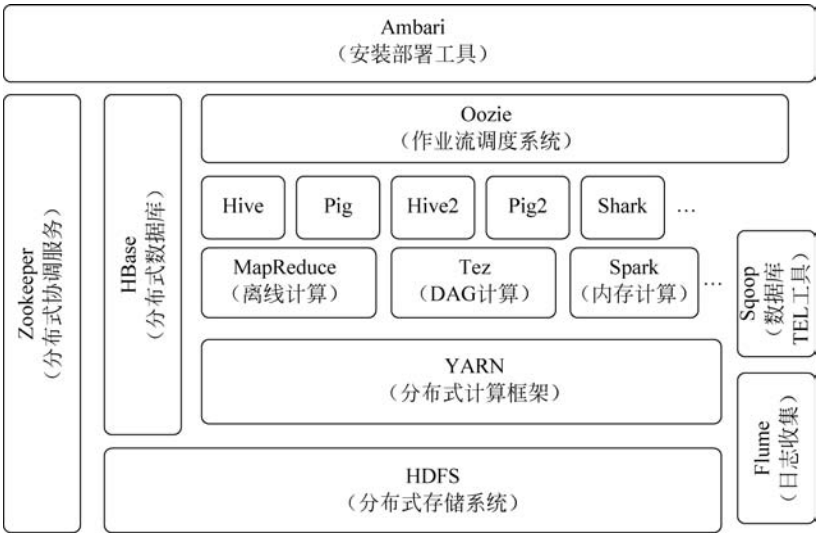

图 1-3　Hadoop2. x 的生态系统组成

1. HDFS（Hadoop 分布式文件系统）

HDFS 是 Hadoop 体系中数据存储管理的基础，是 GFS 克隆版。它是一个高度容错的系统，能检测和应对硬件故障，用于在低成本的通用硬件上运行。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，适合带有大型数据集的应用程序。它提供了一次写入、多次读取的机制，数据以块的形式，同时分布在集群的不同物理机器上。

2. YARN（分布式资源管理器）

YARN 是一种分层的集群框架，分层结构的本质是 ResourceManager，这个实体控制

整个集群并管理应用程序向基础计算资源的分配。ResourceManager 将各个资源部分(计算、内存、带宽等)精心安排给基础 NodeManager(YARN 的每个节点代理)。

3. MapReduce(分布式计算框架)

Hadoop MapReduce 是 Google MapReduce 克隆版,用于进行大数据量的计算。它屏蔽了分布式计算框架细节,将计算抽象成 Map 和 Reduce 两部分,其中 Map 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 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 非常适合在大量计算机组成的分布式并行环境里进行数据处理。

4. HBase(分布式列存数据库)

HBase 是一个建立在 HDFS 之上、面向列的,针对结构化数据的可伸缩、高可靠、高性能、分布式的动态模式数据库。HBase 采用了 BigTable 的数据模型:增强的稀疏排序映射表(Key/Value),其中的键由行关键字、列关键字和时间戳构成。HBase 提供了对大规模数据的随机、实时读写访问。HBase 中保存的数据可以使用 MapReduce 来处理,它将数据存储和并行计算完美地结合在一起。

5. Zookeeper(分布式协作服务)

Zookeeper 用于解决分布式环境下的数据管理问题:统一命名、状态同步、集群管理、配置同步等。Hadoop 的许多组件依赖于 Zookeeper,它运行在计算机集群上面,用于管理 Hadoop 操作。

6. Hive(数据仓库)

Hive 定义了一种类似 SQL 的查询语言(HQL),将 SQL 转化为 MapReduce 任务在 Hadoop 上执行。通常用于离线分析。

7. Oozie

Oozie 是一个开源的工作流和协作服务引擎,基于 Apache Hadoop 的数据处理任务。Oozie 是可扩展的、可伸缩的面向数据的服务,运行在 Hadoop 平台上。Oozie 包括一个离线的 Hadoop 处理的工作流解决方案,以及一个查询处理 API。

8. Ambari

Ambari 是一种基于 Web 的工具,支持 Apache Hadoop 集群的供应、管理和监控。Ambari 已支持大多数 Hadoop 组件,包括 HDFS、MapReduce、Hive、Pig、HBase、Zookeeper、Sqoop 和 Hcatalog 等。Ambari 使用 Ganglia 收集度量指标,用 Nagios 支持系统报警,当需要引起管理员的关注时(如节点停机或磁盘剩余空间不足等问题),系统将向其发送邮件。

Hadoop 主要应用于数据量大的离线场景。其特征如下。

- 数据量大。一般真正线上用 Hadoop 的,机器集群规模大多为上百台到几千台。
- 离线。MapReduce 框架下,很难处理实时计算,作业都以日志分析这样的线下作业为主。
- 数据块大。由于 HDFS 设计的特点,Hadoop 适合处理文件块大的文件。

例如,百度每天都会有用户对侧边栏广告进行点击。这些点击都会被记入日志。然后在离线场景下,将大量的日志使用 Hadoop 进行处理,分析用户习惯等信息。

1.3 基于 Spark 系统的大数据平台

Spark 是专为大规模数据处理而设计的快速通用的计算引擎，是 UC Berkeley AMP lab（加州大学伯克利分校的 AMP 实验室）所开源的类 Hadoop MapReduce 的通用并行框架。Spark 拥有 Hadoop MapReduce 所具有的优点；但不同于 MapReduce 的是，Job 中间输出结果可以保存在内存中，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 的算法。

Spark 的官网是 http://spark.apache.org/，主界面如图 1-4 所示。截止到 2020 年 8 月，Spark 的最新版本是 3.0.0，对应最新的 Hadoop 版本是 3.2。支持的编程语言有 Java、Scala、Python、R 和 SQL。

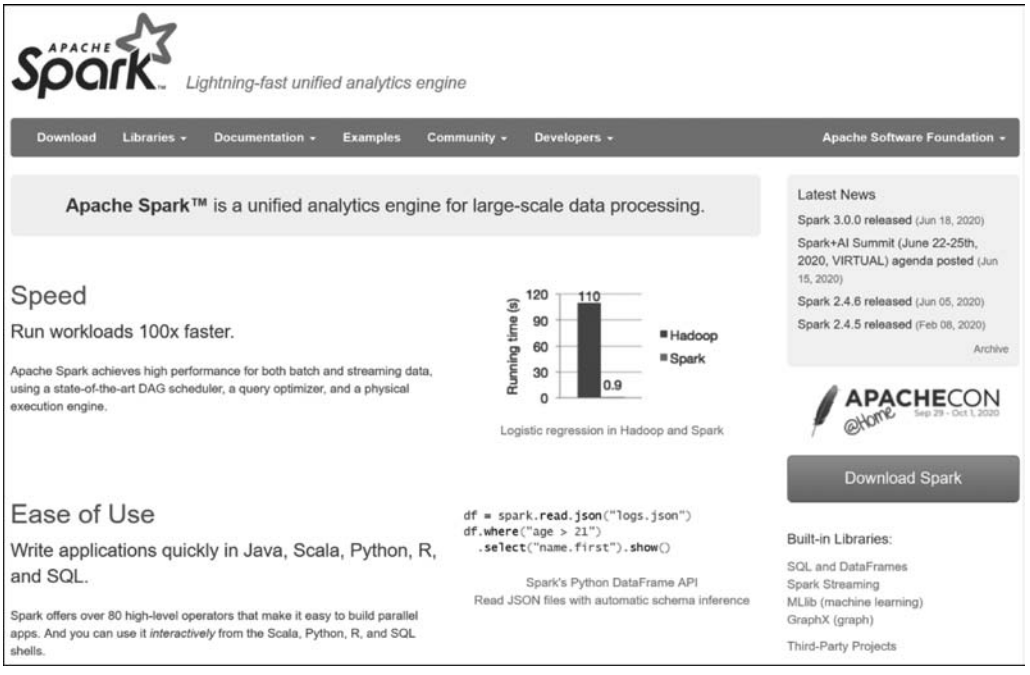

图 1-4　Spark 的主页信息

测试表明，在内存计算下，Spark 比 Hadoop 快 100 倍。Spark 提供了 80 多个高级运算符，具有良好的易用性。同时，Spark 提供了大量的库，包括 Spark Core、Spark SQL、Spark Streaming、MLlib、GraphX。开发者可以在同一个应用程序中无缝组合使用这些库。

1.3.1 Spark 的生态系统

Spark 的生态系统结构如图 1-5 所示。

Spark 能够以独立集群模式运行，也能运行于 EC2、Hadoop YARN 或 Apache Mesos 之上。Spark 能够访问的设计源包括 HDFS、Cassandra、HBase、Hive、Tachyon 以及其他 Hadoop 数据源。

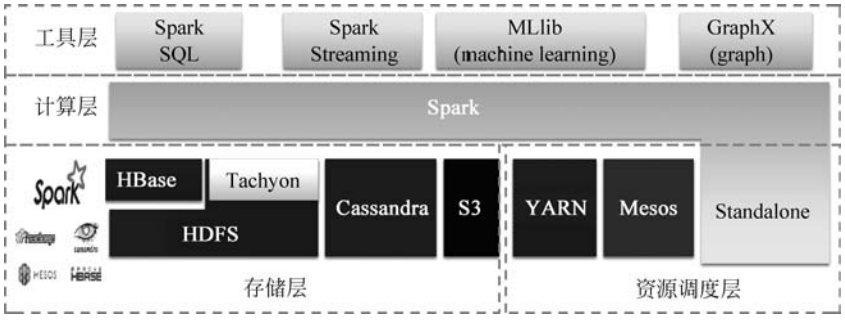

图 1-5　Spark 的生态系统结构

Spark 的主要应用场景见表 1-2。

表 1-2　Spark 生态系统组件的应用场景

应用场景	时间跨度	其他框架	Spark 生态系统中的组件
复杂的批量数据处理	小时级	MapReduce、Hive	Spark
基于历史数据的交互式查询	分钟级、秒级	Impala、Dremel、Drill	Spark SQL
基于实时数据流的数据处理	毫秒级、秒级	Storm、S4	Spark Streaming
基于历史数据的数据挖掘	—	Mahout	MLlib
图结构数据的处理	—	Pregel、Hama	GraphX

1.3.2　Spark 与 Hadoop 的比较

1. 框架比较

在 Hadoop 中，MapRedcue 通过 shuffle 将 Map 和 Reduce 两个阶段连接起来。套用 MapReduce 模型解决问题时，须将问题分解为若干个有依赖关系的子问题，每个子问题对应一个 MapReduce 作业，最终所有这些作业形成一个 DAG。

Spark 是通用的 DAG 框架，可以将多个有依赖关系的作业转换为一个大的 DAG。其核心思想是将 Map 和 Reduce 两个操作进一步拆分为多个元操作，这些元操作可以灵活组合，产生新的操作，并经过一些控制程序组装后形成一个大的 DAG 作业。

2. 中间结果存储方式

在 DAG 中，由于有多个 MapReduce 作业组成，每个作业都会从 HDFS 上读取一次数据和写一次数据(默认写三份)，即使这些 MapReduce 作业产生的数据是中间数据也需要写 HDFS，如图 1-6 所示。

Hadoop 的这种表达作业依赖关系的方式比较低效，会浪费大量不必要的磁盘和网络 IO，根本原因是作业之间产生的数据不是直接流动的，而是借助 HDFS 作为共享数据存储系统。

而在 Spark 中，使用内存(内存不够使用本地磁盘)替代了使用 HDFS 存储中间结果，如图 1-7 所示。对于迭代运算效率更高。

3. 操作模型

Hadoop 只提供了 Map 和 Reduce 两种操作。所有的作业都得转换成 Map 和 Reduce 的操作。

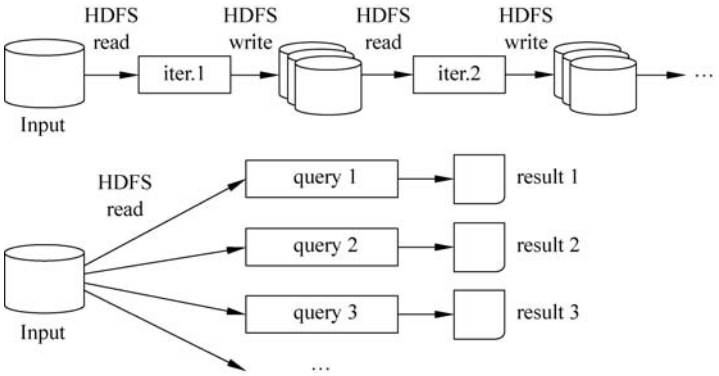

图 1-6　Hadoop 对中间结果的处理过程

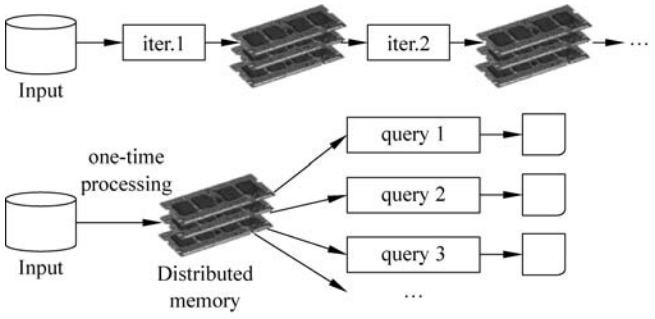

图 1-7　Spark 对中间结果的处理过程

而 Spark 提供很多种的数据集操作类型，如 Transformations 包括 map、filter、flatMap、sample、groupByKey、reduceByKey、union、join、cogroup、mapValues、sort、partionBy 等多种操作类型，actions 操作包括 count、collect、reduce、lookup、save 等多种。这些多种多样的数据集操作类型，给开发上层应用的用户提供了方便。

4. 编程模型

Hadoop 就是唯一的 Data Shuffle 一种模式。

Spark 用户可以命名、物化、控制中间结果的存储、分区等，编程方式更灵活。

5. 缓存

Hadoop 无法缓存数据集。

Spark 的 60％内存用来缓存 RDD，对于缓存后的 RDD 进行操作，节省 IO 接口，效率高。

6. 应用场景

Hadoop 用于离线大规模分析处理。

对于 Hadoop 适用的场景，Spark 基本上都适合（在只有 Map 操作或者只有一次 Reduce 操作的场景下，Spark 比 Hadoop 的优势不明显）。对于迭代计算，Spark 比 Hadoop 有更大的优势。

7. 其他

Hadoop 对迭代计算效率低。Spark 使用 Scala 语言，更简洁、高效；Spark 对机器学习算法、图计算能力有很好的支持。