第 3 章

大数据处理技术

第3章

大数据处理技术

随着互联网的发展，最早面对大数据的是运维搜索引擎的公司，如Google、百度等，然后就是运维社交网络、电子商务的公司，如腾讯、阿里巴巴、Facebook、Twitter、亚马逊等。现在大数据的概念和价值正被大众广泛地认知和重视，大数据的价值利用正在改变众多的行业。

大数据一开始是通过大量网页、个人博客、社交媒体、社交网络、网上销售等产生的，后来随着传感器越来越便宜，移动设备、用户的地理信息、遥感、软件日志、相机、麦克风、射频识别（RFID）、无线传感器网络成为新的更加庞大的数据源，有人推算每隔三年全世界产生的数据量就会翻一倍。因为存储成本的下降，廉价和大量的信息开始被收集和利用起来。

3.1 什么是大数据4V特征

Gartner的分析师Doug Laney早在2001年就提出了著名的大数据“3V”定义，即Volume、Velocity、Variety。

Volume（容量）：巨大的体量，每天从各种来源产生的海量数据。

Velocity（速度）：数据流的速度是前所未有的。

Variety（多样性）：来源广泛，数据格式和种类繁多，有结构化、非结构化和半结构化数据。

随着人们对大数据的深入了解，4V 逐渐被广泛接受，也就是在原来的基础上再加上 Value（价值），这说明大数据具有极高的价值，但是价值密度比较低。

3.2 大数据的存储与分析

面对大数据，传统的计算机体系结构、数据库管理系统、数据分析软件在存储能力、处理能力和处理速度上都已经无法满足要求，因为数据处理工作可能需要在数十台甚至数千台服务器上进行，而且硬件设备需要不断扩充。这就要求在数据量大幅增加的情况下，系统具有良好的横向可扩展性，也就是只需要不断向网络添加廉价的服务器，就能扩充存储和计算的能力，不需要对现有系统做任何调整。

分布式系统在硬件上是一组分布式节点（服务器、组件），它们通过网络相互连接，并且通过分布式系统的协同软件协同所有的节点，并为用户提供简单的界面，使用户感觉所使用的系统是统一的，即用户无须了解复杂的硬件系统。

由于使用分布式系统，就需要面对网络故障和设备单点故障给数据存储和处理带来的问题。一个分布式系统是无法同时满足一致性（C）、可用性（A）、分区容忍性（P）的。

一致性（C）：在分布式系统中的所有数据副本都是一样的，而且在并发

操作的过程中始终是一致的。例如，银行应用程序，在分布的、并发的操作过程中始终显示正确的账户余额。

可用性（A）：当部分节点或网络出现故障后，分布式系统作为整体能否继续提供可靠的服务。

分区容忍性（P）：如果因为网络的问题，导致数据副本不一致，就会出现分区。分区容忍性就是在多大的程度上容忍数据副本不一致。

系统设计可能的选项有以下三种。

（1）选择 CA：如银行应用程序，不能出现分区（数据副本不一致），也就是分区容忍度为 0，因为数据分区可能导致账户余额出现错误，所以只能选择 CA。

（2）选择 AP：数据副本可能不一致。有些数据即使较长时间不一致也没有什么后果，如用户评论，在不同的节点上只更新局部的评论数据，当系统空闲时再根据需要对数据副本进行统一处理。

（3）选择 CP：不强调可用性。前面提到银行应用程序常常选择 CA，当数据副本不一致的时候应用就不能用了，但是能不能在部分数据副本不一致的时候，提供一定的可用性，如不是查余额，而是查历史信用，这时可以选择 CP，也就是允许分区，牺牲部分可用性。

分布式系统无法同时满足 CAP，策略必然是选择其中两个，降低第三个的期待。

3.3　Hadoop 概要

据 Hadoop 的联合创始人 Doug Cutting 和 Mike Cafarella 所述，Hadoop 的起源是 Google 的两篇文章：*The Google File System* 和 *MapReduce*：*Simplified Data*

Processing on Large Clusters；Hadoop 最初是为了开发搜索引擎 Nutch 而开发的，由于其在 Nutch 引擎中有着良好的应用，2006 年从 Nutch 中分离出来，成为一套完整而独立的软件；Doug Cutting 以他儿子的玩具大象的名字 Hadoop 为其命名，从 Nutch 中分离出来的初始代码包括用于 HDFS（Hadoop Distributed File System）的大约 5000 行代码和用于 MapReduce 的大约6000行代码。

搜索引擎先要用爬虫把全球的网页都抓下来，然后建立索引。由于网页的内容是不断改变的，所以对海量的数据建立索引、更新索引是一项计算量巨大的工作。同样，用户查询索引时，由于巨大的并发用户数，响应时间又限制在 1 秒左右，而且检索结果还有质量要求，难度也很大。Google 解决上述问题的技术手段分别是 GFS、MapReduce 和 BigTable。

2008 年 2 月，雅虎宣布将其搜索引擎产品部署到了一个 Hadoop 集群上，这个集群拥有 1 万个内核，在 209 秒内完成了 1TB 数据的排序。当然，对现在的大互联网公司而言，这个数据量的处理已经可以在数秒内完成。

Hadoop 是一个重要的 Apache 开源项目，它帮助用户使用简单的编程模型在集群的分布式环境中存储和处理大数据。

Hadoop 专注于解决大数据处理中的三个重要问题。

（1）大量、不断增长的数据存储。HDFS 提供了一种分布式的方式来存储大数据。通过分布式存储，当数据量增长时，计算机设备只需要水平扩展而不需要垂直扩展，即可以根据需要在运行时向 HDFS 集群添加新节点，而不是增加每个节点中的硬件。

（2）支持存储各种格式的数据。HDFS 支持存储各种格式的数据，无论是结构化的、半结构化的，还是非结构化的。

（3）高速处理数据。将处理逻辑发送到存储数据的节点，以便每个节点可以并行处理部分数据，然后将每个节点生成的所有中间输出合并在一起，并将最终结果送回用户，这部分由 MapReduce 和 Yarn 实现。

Hadoop 项目由以下模块组成。

- Hadoop Common：包含其他 Hadoop 模块所需的库和实用程序（只包含一些基础模块，没有具体原理，下面就不再详细介绍）。
- Hadoop 分布式文件系统（HDFS）：可以部署在廉价硬件上的分布式文件系统，可以实现以流的形式访问文件系统中的数据。
- Hadoop MapReduce：一种用于大规模数据处理的编程模型，它提供了一个框架，使得编程人员可以在不了解分布式并行系统的情况下，编写适合自己任务的数据处理任务。
- Hadoop Yarn：用于作业调度和集群资源管理的框架。

1. HDFS

HDFS 是在高度容错设计和低成本的硬件上部署的分布式文件系统，也就是说 HDFS 的可靠性不依赖昂贵的硬件。HDFS 可以较好地支持大文件，文件访问的方式是“一次写多次读”，而不是频繁写、随机修改文件内容。HDFS 的架构如图 3.1 所示。

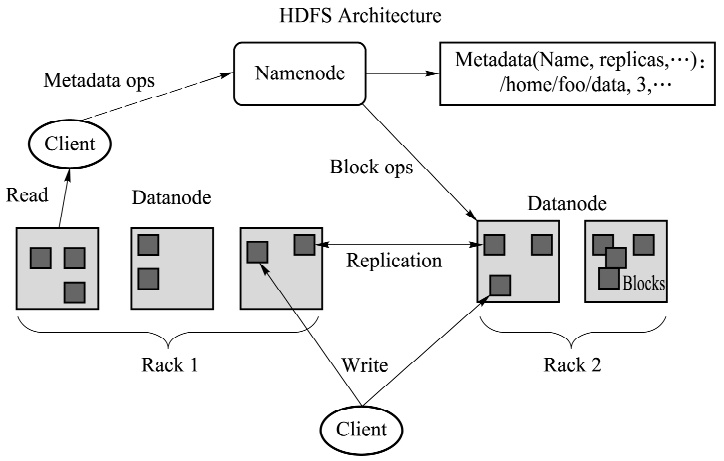

图 3.1　HDFS 的架构

HDFS 集群由一个 Namenode 组成，它是一个主服务器，管理文件系统元数据（文件名、目录结构、存放位置等）并管理客户机（Client）对文件的访问。此外，还有许多 Datanode，通常集群中的每个节点（这些节点上安装有 Linux 操作系统）有一个 Datanode，管理本节点的数据存储。文件被分割成一个或多个块（Blocks），这些块存储在一组 Datanode 中。文件名称和文件分割块之间的对应关系构成了名称空间，Namenode 执行文件系统命名空间操作，如打开、关闭和重命名存储在 HDFS 中的文件，并把这些操作映射到多个数据节点中（Datanode）。Datanode 为来自客户机（Client）的读写请求提供服务，还根据 Namenode 的指令执行块（Blocks）的创建、删除和复制操作。根据 Hadoop 的实现，数据节点的默认块大小从 64 MB 到 128 MB 不等，这可以通过数据节点的配置进行更改。

2. MapReduce

Hadoop 的数据处理是通过 MapReduce 实现的，它是 Hadoop 提供的一个框架，通过这个框架让用户专注于业务的逻辑。MapReduce 需要根据任务的不同由用户自己编程定义（也有一些高层的工具，如 Hive 提供类似 SQL 的命令接口，用户编写完类似 SQL 的程序后，还是会自动翻译成 MapReduce 支持的程序）。下面通过一个单词计数的任务来了解 MapReduce 工作的过程，如图 3.2 所示。

（1）Splitting：首先将输入以固定大小分到三个节点。

（2）Mapping：然后，每个 Mapper 根据本节点中的单词的出现频率进行标记，得到多个“键－值对”，单词是键，频率是值。

（3）Shuffling：进行分区处理，同时进行排序和洗牌，以便将具有相同键的所有“键－值对”发送到相应的 Reducer。因此，每个 Reducer 都有一个唯一的键和一个与该键对应的值列表，如 Hadoop，（1，2，2）。

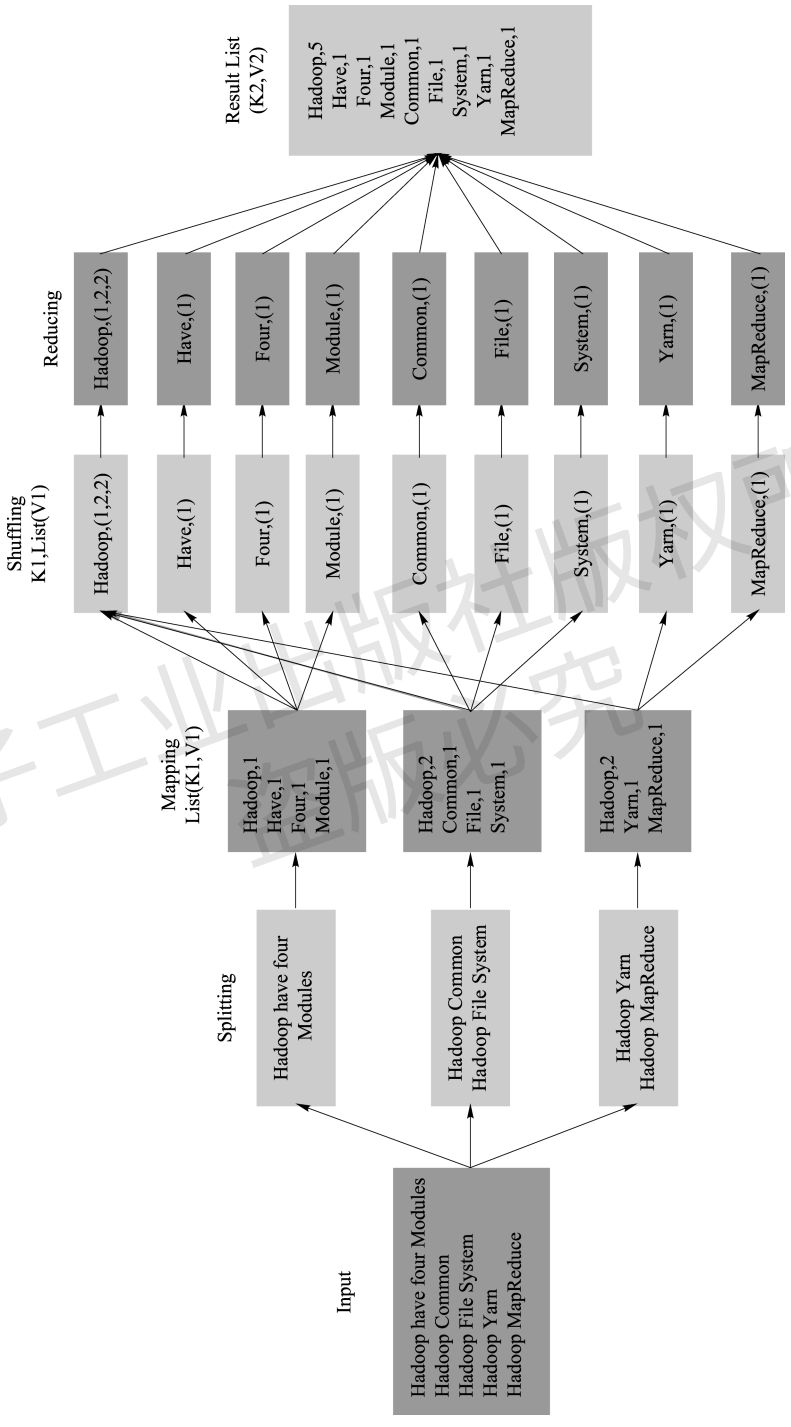

图 3.2　单词计数的任务的 MapReduce 工作过程

（4）Reducing：每个 Reducer 将接收到的“键 - 值列表”中出现的值进行计数。例如，接收到 Hadoop，（1，2，2），给出输出 Hadoop，5。

（5）最后，收集所有“键 - 值对”，并将其写入输出文件中。

3. Yarn

MapReduce 的执行过程需要 Yarn 进行资源的调度，Yarn 的架构如图 3.3 所示。

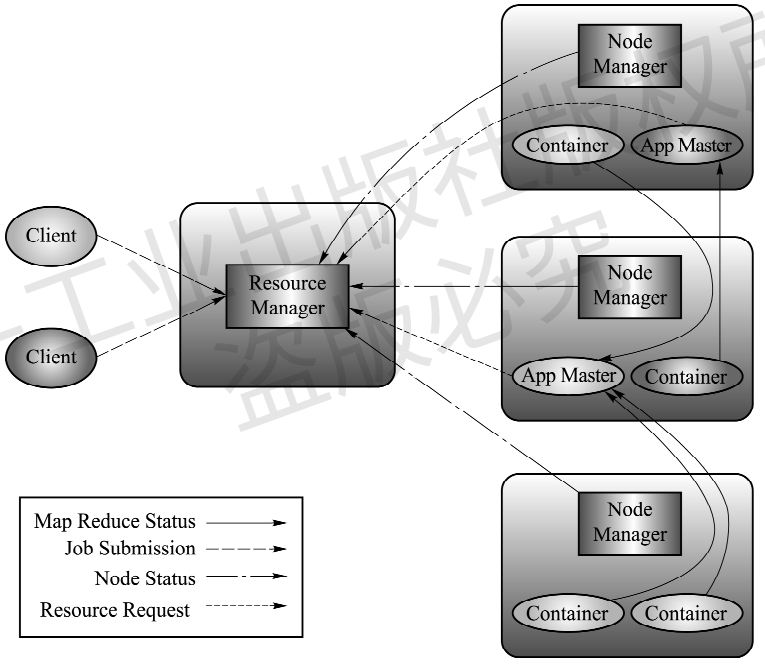

图 3.3　Yarn 的架构

（1）Resource Manager。

Resource Manager 是整个集群的资源管理器，运行在主服务器上，它管理资源并调度在 Yarn 上运行的应用程序。实际上，Resource Manager 有两个组

件：调度程序和应用程序管理器（Application Manager），调度程序负责将资源分配给各种正在运行的应用程序；应用程序管理器负责接收作业（Job）提交并协商执行应用程序的第一个容器（Container），还要不断和节点管理器（Node Manager）联系并跟踪各个节点是否故障。

（2）Node Manager。

Node Manager 存在于集群的每个节点上，负责管理容器并监控每个容器中资源（CPU、内存、硬盘、网络）的使用情况，还要跟踪节点的健康状况和进行日志管理，并与资源管理器保持持续的沟通。

3.3.1　Hadoop 的生态圈

围绕 Hadoop 核心系统，还有一系列的工具软件方便数据采集、数据载入、数据库管理、数据仓库管理、数据分析等方面的工作，从而构成了 Hadoop 的生态圈，如图 3.4 所示（因为软件非常多，这个图只包括了常用的软件，要注意很多软件有替代品）。

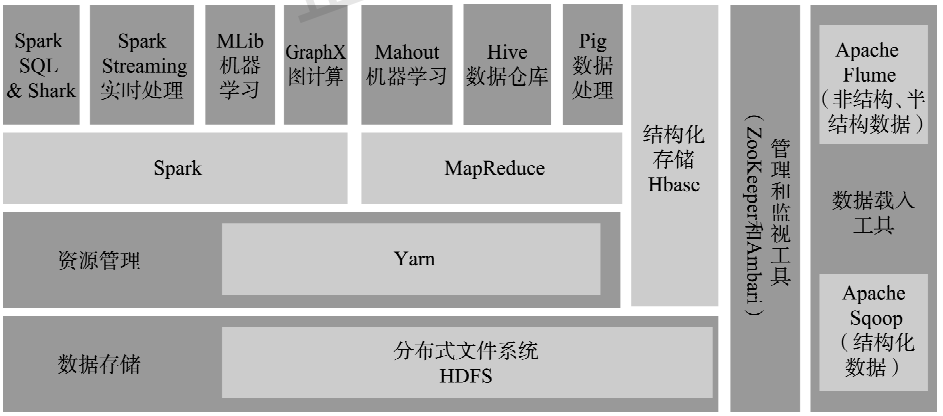

图 3.4　Hadoop 的生态圈

（1）Ambari：Ambari 旨在简化 Hadoop 管理，是帮助管理人员创建、管理

和监视 Hadoop 集群的软件。Ambari 提供了一个直观的、易于使用的 Web 界面，可以跨任意数量的主机安装 Hadoop 服务，包括管理 Hadoop 集群，处理集群的 Hadoop 服务配置，跨整个集群启动、停止和重新配置 Hadoop 服务，提供用于监视 Hadoop 集群的健康状况和状态的仪表板，在节点宕机、剩余磁盘空间不足时发出警报。Ambari 不但可以管理 Hadoop 核心，也能管理 Hadoop 生态的其他软件，如 Hive、Hbase、Sqoop、ZooKeeper 等。

（2）Hbase：Hbase 类似 Google BigTable，可以管理非常大的表——数十亿行×数百万列，并且能随机地、实时地读、写和访问大数据，这是 HDFS 缺少的能力。

（3）Hive：Hive 是数据仓库软件，用于读、写和管理 Hadoop 文件系统中的大型数据集。它支持类似于 SQL 语言、帮助用户查询 Hadoop 中数据的查询语言——Hive SQL。Hive 编译器会把 Hive SQL 编译成 MapReduce 任务，从而降低了直接用 Java 编写 MapReduce 程序的难度。

（4）Mahout：Mahout 提供了线性代数和统计的 Java 库，在分布式环境下实现了协同过滤、聚类和分类的机器学习算法。它已经独立于 Hadoop 的 MapReduce 环境，支持 Spark 和 Flink。

（5）Pig：Pig 是一个分析大型数据集的平台，在此平台上开发人员可以使用 Pig Latin 语言来定义数据分析任务。Pig 的基础结构是编译器，该编译器把 Pig Latin 语言定义的数据分析任务转换成 MapReduce 程序。用 Pig 编程简单易学，而且具有良好的可扩展性，用户可以为自己的特殊目的创建自己的函数。Pig 降低了用户使用 MapReduce 的难度。

（6）Spark：Spark 是一个 Hadoop 数据计算引擎，和 MapReduce 类似。一个数据处理的流程会有很多个 MapReduce 过程，一个 MapReduce 过程会把 Reduce 的结果回写硬盘，下一个 MapReduce 过程再读出，接着再回写。这个

方法保证了数据不会因为故障损坏、丢失，但是却大大地降低了处理速度。Spark 将数据存放在内存之中，并且采用一种叫做弹性分布数据（Resilient Distributed Dataset，RDD）的技术将数据存储在集群的工作节点上的内存和磁盘中，以保证容错性。数据在内存中处理 Spark 的速度远远高于 MapReduce，缺点是需要更多的内存。

（7）ZooKeeper：ZooKeeper 是一个分布式的应用程序协调服务，是 Google 的类似产品 Chubby 的一个开源的实现，也是 Hbase 的重要组件。它为分布式应用提供一致性服务，提供的功能包括维护配置信息、域名服务、分布式同步、组服务等。

3.3.2　Hadoop 的局限性

Hadoop 的局限性有以下几点。

（1）Hadoop 不适合小数据的处理。Hadoop 分布式文件系统缺乏随机读取小文件的有效支持。

（2）Hadoop 是一个框架，不是解决方案。当面对一个数据分析的业务时，需要开发 MapReduce 代码和编写数据如何处理的代码，这样的开发任务，对很多业务型企业来说成本很高。

（3）用 MapReduce 的方式编程并不容易，集群环境调试程序也是一件困难的事。

（4）使用 Hive 和 Pig 可以用更高级的语言编写数据处理逻辑，但是最终都要映射到 MapReduce 程序，还是受到框架的限制。

（5）开源软件的获得成本很低，但是企业需要额外的人员进行开发和维护，总体使用成本并不低。

3.4 阿里云数加平台 MaxCompute 与 DataWorks 的功能

3.4.1 MaxCompute 的功能

大数据计算服务（MaxCompute，原名 ODPS）是一种快速、完全托管的 EB 级数据仓库解决方案。随着数据收集手段不断丰富，行业数据大量积累，数据规模已增长到了传统软件行业无法承载的海量数据（TB、PB、EB）级别。MaxCompute 致力于批量结构化数据的存储和计算，提供海量数据仓库的解决方案及分析建模服务。

由于单台服务器的处理能力有限，海量数据的分析需要分布式的计算模型。分布式的计算模型对数据分析人员要求较高且不易维护。数据分析人员不仅需要了解业务需求，同时还需要熟悉底层分布式计算模型。MaxCompute 为用户提供完善的数据导入方案及多种经典的分布式计算模型，用户可以不必关心分布式计算和维护细节，便可轻松完成大数据分析。MaxCompute 是阿里云提供的统一的大数据计算引擎，它和其他产品的关系密切，阿里云 MaxCompute 生态系统如图 3.5 所示。

MaxCompute 构建于阿里云计算平台之上，并且为用户使用该引擎提供了如下功能。

1. 数据存储

（1）支持大规模计算存储，适用于 TB 级以上规模的存储及计算需求，最大可达 EB 级别。

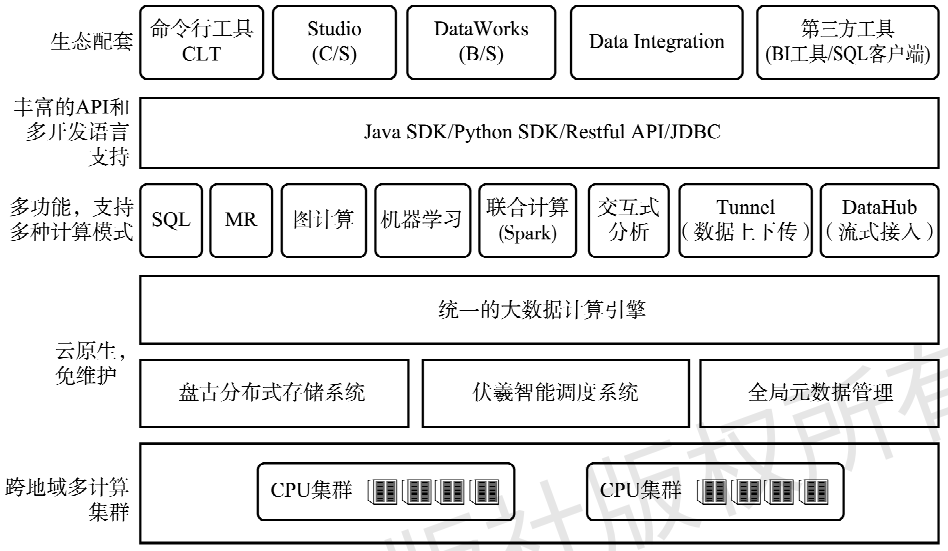

图 3.5　阿里云 MaxCompute 生态系统

（2）数据分布式存储，多副本冗余，数据存储对外仅开放表的操作接口，不提供文件系统访问接口。

（3）自研数据存储结构，表数据列式存储，默认高度压缩，后续将提供兼容 ORC 的 Ali-ORC 存储格式。

（4）支持外表，将存储在 OSS（文件形式存储）、OTS（表格形式存储）的数据映射为二维表。

（5）支持 Partition、Bucket 的分区、分桶存储。

（6）更底层不是 HDFS，而是阿里云自研的盘古分布式存储系统，但可借助 HDFS 理解对应的表之中文件的体系结构、任务并发机制。

（7）使用时，存储与计算解耦，不需要仅仅为了存储增加不必要的计算资源。

2. 多种计算模型

需要说明的是，传统数据仓库场景下，实践中有大部分的数据分析需求可以通过 SQL+UDF 来完成。但随着企业对数据价值的重视，以及更多不同的角色开始使用数据，企业也会要求有更丰富的计算功能来满足不同场景、不同用户的需求。MaxCompute 不仅仅提供 SQL 数据分析语言，它还在统一的数据存储和权限体系之上，支持多种计算模型。

3. MaxCompute SQL

MaxCompute 100% 支持数据库评测基准 TPC－DS，同时语法高度兼容 Hive，有 Hive 背景的开发者可以直接上手，特别是在大数据规模下性能强大。

（1）完全自主开发的 Compiler，语言功能开发更灵活且迭代快，语法语义检查更加灵活高效。

（2）基于代价的优化器，更智能、更强大、更适合复杂的查询。

（3）基于 LLVM 的代码生成，让执行过程更高效。

（4）支持复杂数据类型（array，map，struct）。

（5）支持 Java、Python 语言的 UDF/UDAF/UDTF。

（6）语法：values、cte、semijoin、from 倒装、subquery operations、set operations（union /intersect /minus）、select transform 、user defined type、grouping set（cube/rollup/grouping set）、脚本运行模式、参数化视图。

（7）支持外表（通过 StorageHandler 读取外部数据源，支持非结构化数据的读取）。

4. MapReduce

MaxCompute 支持 MapReduce 编程接口（提供优化增强的 MaxCompute Ma-

pReduce，也提供高度兼容 Hadoop 的 MapReduce 版本），不暴露文件系统，输入输出都是通过 MaxCompute 客户端工具、DataWorks 提交作业的。

5. MaxCompute Graph

（1）MaxCompute Graph 是一套面向迭代的图计算处理框架。图计算作业使用图进行建模，图由点（Vertex）和边（Edge）组成，点和边包含权值（Value）。

（2）通过迭代对图进行编辑、演化，最终求解出结果。

（3）典型应用有 PageRank、单源最短距离算法、K－均值聚类算法等。

（4）使用 MaxCompute Graph 提供的接口 Java SDK 编写图计算程序并通过 MaxCompute 客户端工具的 jar 命令提交任务。

6. PyODPS

PyODPS 是 MaxCompute 的 Python SDK，同时也提供 DataFrame 框架和类似 pandas 的语法，能利用 MaxCompute 强大的处理能力来处理超大规模数据。

（1）PyODPS 提供对 ODPS 对象（如表 、资源 、函数等）的访问。

（2）支持通过 run_sql/execute_sql 的方式来提交 SQL。

（3）支持通过 open_writer 和 open_reader 或原生 Tunnel API 的方式来上传、下载数据。

（4）PyODPS 提供 DataFrame API 和类似 pandas 的接口，能充分利用 MaxCompute 的计算能力进行 DataFrame 的计算。

（5）PyODPS DataFrame 提供了很多 pandas－like 的接口，扩展了它的语法，如增加了 MapReduce API 以适应大数据环境。

（6）利用 map 、apply 、map_reduce 等方便在客户端写函数、调用函

数的方法，用户可在这些函数里调用第三方库，如 pandas、scipy、scikit-learn、nltk。

7. Spark

MaxCompute 提供了 Spark on MaxCompute 的解决方案，是 MaxCompute 提供的兼容开源的 Spark 计算服务，让它在统一的计算资源和数据集权限体系之上提供 Spark 计算框架，并支持用户以熟悉的开发使用方式提交运行 Spark 作业。

（1）支持原生多版本 Spark 作业：Spark1. x/Spark2. x/Spark3. x 作业都可运行。

（2）开源系统的使用体验：Spark-submit 提交方式（暂不支持 Spark-shell/Spark-sql 的交互式）提供原生的 Spark WebUI 供用户查看。

（3）通过访问 OSS、OTS、Database 等外部数据源，实现更复杂的 ETL 处理，支持对 OSS 非结构化数据的读写。

（4）使用 Spark 面向 MaxCompute 内外部数据开展机器学习，扩展应用场景。

8. 交互式查询和分析支持

MaxCompute 产品的交互式查询和分析支持功能的特性如下。

（1）兼容 PostgreSQL：兼容 PostgreSQL 协议的 JDBC/ODBC 接口，所有支持 PostgreSQL 数据库的工具或应用使用默认驱动都可以轻松地连接到 MaxCompute 项目；支持主流 BI 及 SQL 客户端工具的连接访问，如 Tableau、帆软 BI、Navicat、SQL Workbench/J 等。

（2）显著提升的查询性能：提升了一定数据规模下的查询性能，查询结

果秒级可见，支持 BI 分析、Ad-hoc、在线服务等场景。

9. 机器学习

MaxCompute 内置上百种机器学习算法。目前，MaxCompute 的机器学习能力由 PAI 产品统一提供服务，同时 PAI 提供了深度学习框架、Notebook 开发环境、GPU 计算资源及模型在线部署的弹性预测服务。PAI 产品与 MaxCompute 在项目和数据方面无缝集成。

MaxCompute 已经在阿里巴巴集团内部大规模应用，如大型互联网企业的数据仓库和 BI 分析、网站的日志分析、电子商务网站的交易分析、用户特征和兴趣挖掘等。

3.4.2　DataWorks 的功能

DataWorks（数据工场，原大数据开发套件）是阿里云重要的 PaaS 平台产品，为用户提供数据集成、数据开发、数据地图、数据质量和数据服务等全方位的产品服务和一站式开发管理的界面，帮助企业专注于数据价值的挖掘和探索。

用户可以使用 DataWorks 对数据进行传输、转换和集成等操作，从不同的数据存储引入数据，并进行转化和开发，最后将处理好的数据同步至其他数据系统。DataWorks 提供九个核心功能模块，以数据为基础，以全链路加工为核心，提供数据汇聚、研发、治理、服务等多种功能，既能满足平台用户的数据需求，又能为上层应用提供各种行业解决方案。DataWorks 整体功能架构如图 3.6 所示。

DataWorks 主要有以下几个方面的功能。

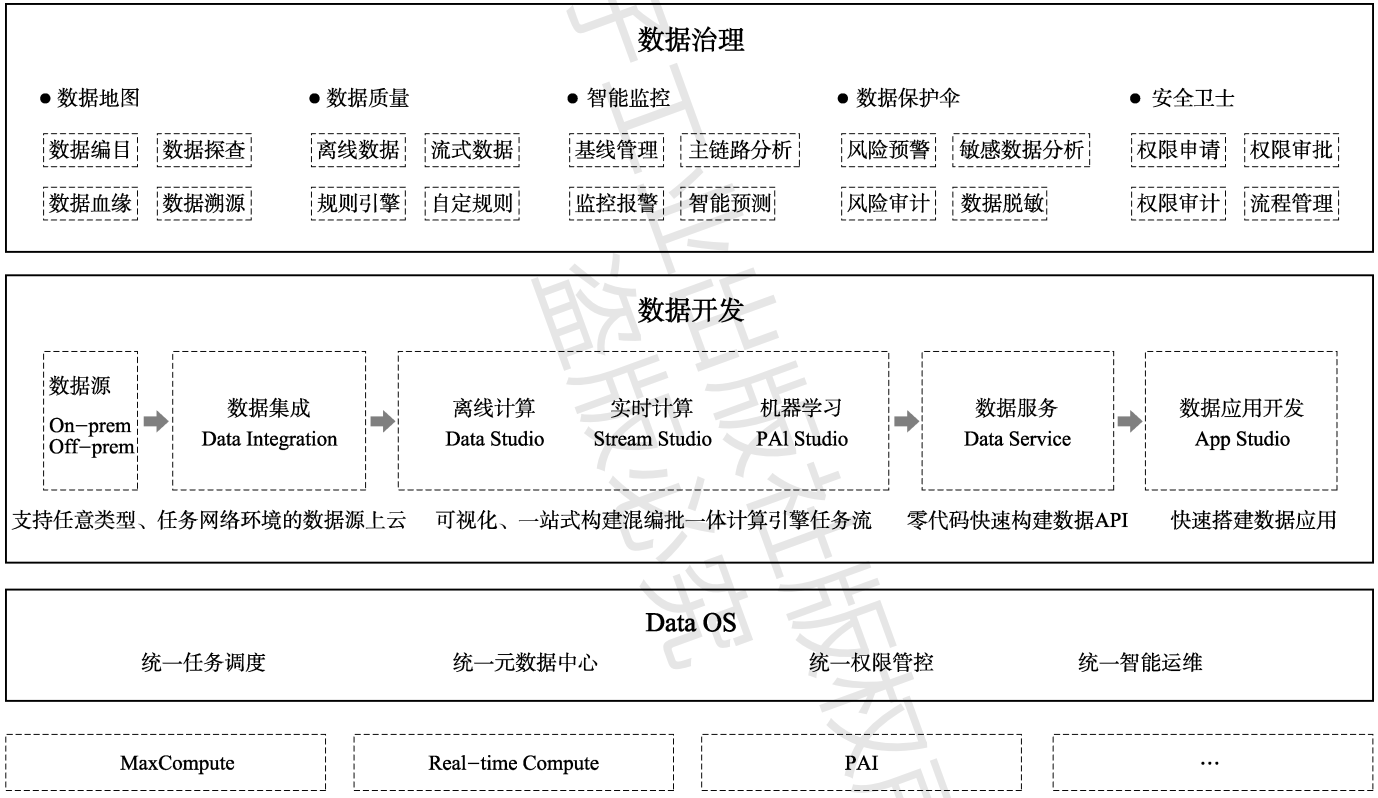

图 3.6　DataWorks 整体功能架构

1. 全面托管的调度

（1）DataWorks 提供强大的调度功能。

①支持根据时间、依赖关系进行任务触发的机制。

②支持每日千万级别的任务，根据 DAG 关系准确、准时地运行。

③支持分钟、小时、天、周、月多种调度周期配置。

（2）完全托管的服务，无须关心调度的服务器资源问题。

（3）提供隔离功能，确保不同租户之间的任务不会相互影响。

2. 支持多种节点类型

DataWorks 支持离线同步、shell、ODPS SQL、ODPS MR 等多种节点类型，通过节点之间的相互依赖对复杂的数据进行分析处理。

（1）数据转化：依托 MaxCompute 强大的能力，保证了大数据的分析处理性能。

（2）数据同步：依托 DataWorks 中数据集成的强力支撑，支持超过 20 种数据源，为用户提供稳定、高效的数据传输功能。

3. 可视化开发

DataWorks 提供可视化的代码开发和工作流设计器页面，无须搭配任何开发工具，通过简单拖拽式开发，即可完成复杂的数据分析任务。只要有浏览器和网络，即可随时随地进行开发工作。

4. 监控报警

运维中心提供可视化的任务监控管理工具，支持以 DAG 图的形式展示任

务运行时的全局情况，可以方便地配置各类报警方式，任务发生错误时可及时通知相关人员，保证业务正常运行。