第 5 章　大数据计算模式与处理系统

计算模式的出现有力推动了大数据技术和应用的发展，使其成为目前大数据处理最为成功、最广为接受使用的主流大数据计算模式。然而，现实世界中的大数据处理问题复杂多样，难以有一种单一的计算模式涵盖所有不同的大数据计算需求。

研究和实际应用中发现，由于 MapReduce 主要适合于进行大数据线下批处理，在面向低延迟和具有复杂数据关系和复杂计算的大数据问题时有很大的不适应性。因此，近几年来学术界和业界在不断研究并推出多种不同的大数据计算模式。

所谓大数据计算模式，即根据大数据的不同数据特征和计算特征，从多样性的大数据计算问题和需求中提炼并建立的各种高层抽象(Abstraction)或模型(Model)。

传统的并行计算方法主要从体系结构和编程语言的层面定义了一些较为底层的并行计算抽象和模型，但由于大数据处理问题具有很多高层的数据特征和计算特征，因此大数据处理需要更多地结合这些高层特征考虑更为高层的计算模式。

5.1　数据计算

面向大数据处理的数据查询、统计、分析、挖掘等需求，促生了大数据计算的不同计算模式，整体上我们把大数据计算分为离线批处理计算、实时交互计算和流计算三种。

5.1.1　离线批处理

随着云计算技术到广泛的应用的发展，基于开源的 Hadoop 分布式存储系统和 MapReduce 数据处理模式的分析系统也得到了广泛的应用。

Hadoop 通过数据分块及自恢复机制，能支持 PB 级的分布式的数据存储，以及基于 MapReduce 分布式处理模式对这些数据进行分析和处理。MapReduce 编程模型可以很容易地将多个通用批数据处理任务和操作在大规模集群上并行化，而且有自动化的故障转移功能。MapReduce 编程模型在 Hadoop 这样的开源软件带动下被广泛采用，应用到 Web 搜索、欺诈检测等各种各样的实际应用中。

Hadoop 是一个能够对大量数据进行分布式处理的软件框架，而且是以一种可靠、高效、可伸缩的方式进行处理，依靠横向扩展，通过不断增加廉价的商用服务器来提高计算和存储能力。用户可以轻松地在上面开发和运行处理海量数据的应用程序。以 Hadoop 平台为代表的大数据处理平台技术包括 MapReduce、HDFS、HBase、Hive、Zookeeper、Avro 和 Pig 等，已经形成了一个 Hadoop 生态圈，如图 5.1 所示。

MapReduce 编程模型是 Hadoop 的心脏，用于大规模数据集的并行运算。正是这种

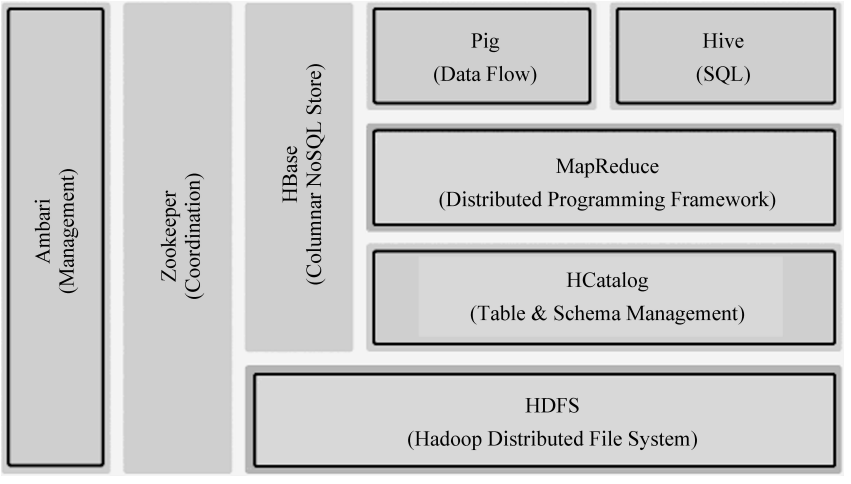

图 5.1 Hadoop 生态圈

编程模式，实现了跨越一个 Hadoop 集群中数百或数千台服务器的大规模扩展性。

分布式文件系统 HDFS 提供基于 Hadoop 处理平台的海量数据存储，其中的 NameNode 提供元数据服务，DataNode 用于存储文件系统的文件块。

HBase 是建立在 HDFS 之上，用于提供高可靠性、高性能、列存储、可伸缩、实时读写的数据库系统，可以存储非结构化和半结构化的松散数据。

Hive 是基于 Hadoop 的大型数据仓库，可以用来进行数据的提取、转化和加载(ETL)，存储、查询和分析存储在 Hadoop 中的大规模数据。

Pig 是基于 Hadoop 的大规模数据分析平台，可以把类 SQL 的数据分析请求转换为一系列经过优化处理的 MapReduce 运算，为复杂的海量数据并行计算提供了一个简单的操作和编程接口。

Zookeeper 是高效、可靠的协同工作系统，用于协调分布式应用上的各种服务，利用 Zookeeper 可以构建一个有效防止单点失效及处理负载均衡的协调服务。

Avro 作为二进制的高性能的通信中间件，提供了 Hadoop 平台间的数据序列化功能和 RPC 服务。

但 Hadoop 平台主要是面向离线批处理应用的，典型的是通过调度批量任务操作静态数据，计算过程相对缓慢，有的查询可能会花几小时甚至更长时间才能产生结果，对于实时性要求更高的应用和服务则显得力不从心。

MapReduce 是一种很好的集群并行编程模型，能够满足大部分应用的需求。虽然 MapReduce 是分布式/并行计算方面一个很好的抽象，但它并不一定适合解决计算领域的任何问题。例如，对于那些需要实时获取计算结果的应用，像基于流量的点击付费模式的广告投放、基于实时用户行为数据分析的社交推荐、基于网页检索和点击流量的反作弊统计等等。对于这些实时应用，MapReduce 并不能提供高效处理，因为处理这些应用逻辑需要执行多轮作业，或者需要将输入数据的粒度切分到很小。

5.1.2　实时交互计算

当今的实时计算一般都需要针对海量数据进行，除了要满足非实时计算的一些需求（如计算结果准确）以外，实时计算最重要的一个需求是能够实时响应计算结果，一般要求为秒级。实时计算一般可以分为以下两种应用场景：

（1）数据量巨大且不能提前计算出结果的，但要求对用户的响应时间是实时的。

主要用于特定场合下的数据分析处理. 当数据量庞大，同时发现无法穷举所有可能条件的查询组合，或者大量穷举出来的条件组合无用的时候，实时计算就可以发挥作用，将计算过程推迟到查询阶段进行，但需要为用户提供实时响应。这种情形下，也可以将一部分数据提前进行处理，再结合实时计算结果，以提高处理效率。

（2）数据源是实时的和不间断的，要求对用户的响应时间也是实时的。

数据源实时不间断的也称为流式数据。所谓流式数据，是指将数据看作是数据流的形式来处理。数据流是在时间分布和数量上无限的一系列数据记录的集合体；数据记录是数据流的最小组成单元。例如，在物联网领域传感器产生的数据可能是源源不断的，实时的数据计算和分析可以动态实时地对数据进行分析统计，对于系统的状态监控、调度管理具有重要的实际意义。

5.1.3　海量数据实时计算

海量数据的实时计算过程可以被划分为以下三个阶段：数据的产生与收集阶段、传输与分析处理阶段、存储和对外提供服务阶段，如图 5.2 所示。

图 5.2　实时计算过程

1. 数据实时采集

数据实时采集在功能上需要保证可以完整地收集到所有数据，为实时应用提供实时数据；响应时间上要保证实时性、低延迟；配置简单，部署容易；系统稳定可靠等。目前，互联网企业的海量数据采集工具包括 Facebook 开源的 Scribe、LinkedIn 开源的 Kafka、Cloudera 开源的 Flume、淘宝开源的 TimeTunnel、Hadoop 的 Chukwa 等，均可以满足每秒数百 MB 的日志数据采集和传输需求。

2. 数据实时计算

传统的数据操作，首先将数据采集并存储在数据库管理系统（DBMS）中，然后通过 query 和 DBMS 进行交互，得到用户想要的答案。整个过程中，用户是主动的，而 DBMS 系统是被动的。但是，对于现在大量存在的实时数据，这类数据实时性强、数据量大、数据格式多种多样，传统的关系型数据库架构并不合适。新型的实时计算架构一般都是采用海量并行处理 MPP 的分布式架构，数据的存储及处理会分配到大规模的结点上进行，以

满足实时性要求，在数据的存储上，则采用大规模分布式文件系统，比如，Hadoop 的 HDFS 文件系统，或是新型的 NoSQL 分布式数据库。

3. 实时查询服务

实时查询服务的实现可以分为三种方式。

(1) 全内存：直接提供数据读取服务，定期 dump 到磁盘或数据库进行持久化。

(2) 半内存：使用 Redis、Memcache、MongoDB、BerkeleyDB 等数据库提供数据实时查询服务，由这些系统进行持久化操作。

(3) 全磁盘：使用 HBase 等以分布式文件系统(HDFS)为基础的 NoSQL 数据库，对于 Key-Value 引擎，关键是设计好 Key 的分布。

实时和交互式计算技术中，Google 的 Dremel 系统表现最为突出。Dremel 是 Google 的“交互式”数据分析系统，可以组建成规模上千的集群，处理 PB 级别的数据。作为 MapReduce 的发起人，Google 开发了 Dremel 系统将处理时间缩短到秒级，作为 MapReduce 的有力补充。

Dremel 作为 Google BigQuery 的 report 引擎，获得了很大的成功。与 MapReduce 一样，Dremel 也需要和数据运行在一起，将计算移动到数据上面。它需要 GFS 这样的文件系统作为存储层。Dremel 支持一个嵌套(nested)的数据模型，类似于 JSON。而传统的关系模型由于不可避免地有大量的 Join 操作，在处理如此大规模的数据的时候，往往是有心无力。Dremel 同时还使用列式存储，分析的时候，可以只扫描需要的那部分数据，以减少 CPU 和磁盘的访问量。同时列式存储是压缩友好的，使用压缩，可以减少存储量，发挥最大的效能。

5.1.4 流计算

在很多实时应用场景中，比如实时交易系统、实时诈骗分析、实时广告推送、实时监控、社交网络实时分析等，数据量大，实时性要求高，而且数据源是实时不间断的。新到的数据必须马上处理完，不然后续的数据就会堆积起来，永远也处理不完。反应时间经常要求在秒级以下，甚至是毫秒级，这就需要一个高度可扩展的流式计算解决方案。

流计算就是针对实时连续的数据类型而准备的。在流数据不断变化的运动过程中实时地进行分析，捕捉到可能对用户有用的信息，并把结果发送出去。在整个过程中，数据分析处理系统是主动的，用户处于被动接收的状态，如图 5.3 所示。

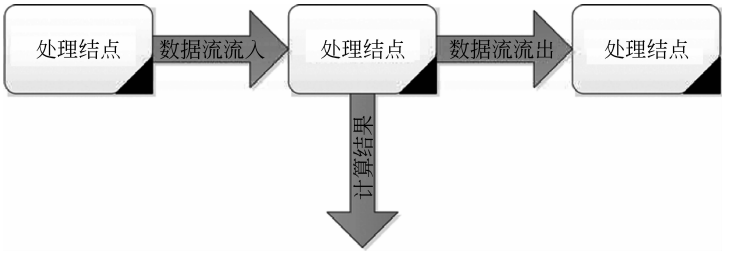

图 5.3 流计算过程

传统的流式计算系统，一般是基于事件机制，所处理的数据量也不大。新型的流处理技术，如 Yahoo 的 S4 主要解决的是高数据率和大数据量的流式处理。

S4 是一个通用的、分布式的、可扩展的、部分容错的、可插拔的平台。开发者可以很容易地在其上开发面向无界不间断流数据处理的应用。

5.2　聚类算法

聚类分析是一种重要的人类行为，早在孩提时代，一个人就通过不断改进下意识中的聚类模式来学会如何区分猫狗、动物植物。目前在许多领域都得到了广泛的研究和成功的应用，如用于模式识别、数据分析、图像处理、市场研究、客户分割、Web 文档分类等。

聚类就是按照某个特定标准（如距离准则）把一个数据集分割成不同的类或簇，使得同一个簇内的数据对象的相似性尽可能大，同时不在同一个簇中的数据对象的差异性也尽可能地大。即聚类后同一类的数据尽可能聚集到一起，不同数据尽量分离。

聚类技术正在蓬勃发展，对此有贡献的研究领域包括数据挖掘、统计学、机器学习、空间数据库技术、生物学以及市场营销等。各种聚类方法也被不断提出和改进，而不同的方法适合于不同类型的数据，因此对各种聚类方法、聚类效果的比较成为值得研究的课题。

5.2.1　聚类算法的分类

目前，有大量的聚类算法。而对于具体应用，聚类算法的选择取决于数据的类型、聚类的目的。如果聚类分析被用作描述或探查的工具，可以对同样的数据尝试多种算法，以发现数据可能揭示的结果。

主要的聚类算法可以划分为如下几类：划分方法、层次方法、基于密度的方法、基于网格的方法以及基于模型的方法。

每一类中都存在着得到广泛应用的算法，例如，划分方法中的 k-mean 聚类算法、层次方法中的凝聚型层次聚类算法、基于模型方法中的神经网络聚类算法等。

目前，聚类问题的研究不仅仅局限于上述的硬聚类，即每一个数据只能被归为一类，模糊聚类也是聚类分析中研究较为广泛的一个分支。模糊聚类通过隶属函数来确定每个数据隶属于各个簇的程度，而不是将一个数据对象硬性归类到某一簇中。目前已有很多关于模糊聚类的算法被提出，如著名的 FCM 算法等。

5.2.2　数据分类与聚类

聚类的算法有很多，现在已知的算法主要有四种类型：划分聚类、层次聚类、基于密度的聚类、基于表格的聚类。

1. 划分聚类

对于给定的数据集，划分聚类需要知道要划分簇的数目 k（$k \leqslant n$，n 是数据集中项的数目）。划分聚类将数据分为 k 组，每组至少有一项。大多数划分聚类都是基于距离的。一般情况下给出了聚类数目 k，首先会产生一个初始的划分，然后用迭代的方法通过更改数

据项所属的簇来提高划分的质量。一个好的划分的标准是同一个簇内的数据项彼此相似，相反地，不同簇的项有较大的区别。

实现全局最优划分往往很难在复杂度忍受的范围内做到。然而，大多数应用都选取了一些启发式方法。比如像选取贪心策略的 k-means 和 k-medoids 算法，都极大地提高了划分质量，并达到了一个局部最优解。这些启发式聚类算法在中小型数据集中挖掘类似球形簇表现非常好。

2. 层次聚类

层次聚类就是通过对数据集按照某种方法进行层次分解，直到满足某种条件为止。层次聚类根据划分的方法分为凝聚和分割两种。凝聚的方法也叫做自底向上方法。它每次迭代将最相近两个项(或者组)合并形成一个新的组，直至最终形成一个组或者达到其他停止的条件。

分割的方法也叫自顶向下，与凝聚的方法相反。开始的时候讲所有数据看成一个组，每一次迭代一个簇就被划分成两个小一点儿的簇。直到最终每个项都是一个簇或者达到了某个停止条件。层次聚类可以是基于距离、基于密度、基于连接的。层次聚类有一个缺点：一旦一个凝聚或分割形成了，这个操作就永远不能再更改了。这样的好处就是计算复杂度相对较低。

3. 基于密度的聚类

很多聚类算法都是根据距离计算的。这样很容易发现球形的簇，很难发现其他形状的簇。基于密度的算法认为，在整个样本空间点中，各目标类簇是由一群的稠密样本点组成的，而这些稠密样本点被低密度区域(噪声)分割，而算法的目的就是要过滤低密度区域，发现稠密样本点。这类算法往往重视数据项的密集程度，因此这些算法都是基于连接的。虽然是基于连接的，但是也强调了连接过程中数据项周围的密度。这样就能发现各种任意形状的聚类簇。

4. 基于网格的聚类

这类算法将数据项的空间划分成有限数目的网格。所有的聚类操作都是在网格上进行的。这样最大的好处是计算速度相当快。因为计算过程跟数据项的数目没有关系，只与每一维网格的数目和维数有关系。对于大数据的数据挖掘问题，网格的方法效率往往会很不错。然而网格只是一种思想，这种思想往往要和其他的算法相结合才能解决好实际问题，比如聚类。

5.3 数据集成

近几十年来，科学技术的迅猛发展和信息化的推进，使得人类社会所积累的数据量已经超过了过去 5000 年的总和，数据的采集、存储、处理和传播的数量也与日俱增。企业实现数据共享，可以使更多的人更充分地使用已有的数据资源，减少资料收集、数据采集等重复劳动和相应费用。

但是，在实施数据共享的过程当中，由于不同用户提供的数据可能来自不同的途径，

其数据内容、数据格式和数据质量千差万别，有时甚至会遇到数据格式不能转换或数据转换格式后丢失信息等棘手问题，严重阻碍了数据在各部门和各软件系统中的流动与共享。因此，如何对数据进行有效的集成管理已成为增强企业商业竞争力的必然选择。

由于现代企业的飞速发展和企业逐渐从一个孤立结点发展成为不断与网络交换信息和进行商务事务的实体，企业数据交换也从企业内部走向了企业之间；同时，数据的不确定性和频繁变动，以及这些集成系统在实现技术和物理数据上的紧耦合关系，导致一旦应用发生变化或物理数据变动，整个体系将不得不随之修改。因此，我们进行数据集成将面临如何适应现代社会发展的复杂需求、有效扩展应用领域、分离实现技术和应用需求、充分描述各种数据源格式以及发布和进行数据交换等问题。

5.3.1　数据集成概述

1. 数据集成模型分类

数据集成是把不同来源、格式、特点、性质的数据在逻辑上或物理上有机地集中，从而为企业提供全面的数据共享。在企业数据集成领域，已经有了很多成熟的框架可以利用。目前通常采用联邦式、基于中间件模型和数据仓库等方法来构造集成的系统，这些技术在不同的着重点和应用上解决数据共享和为企业提供决策支持。在这里将对这几种数据集成模型做一个基本的分析。

1）联邦数据库系统(FDBS)

由半自治数据库系统构成，相互之间分享数据，联盟各数据源之间相互提供访问接口，同时联盟数据库系统可以是集中数据库系统或分布式数据库系统及其他类型数据库，松耦合而不提供统一的接口，但可以通过统一的语言访问数据源，其中的核心是必须解决所有数据源语义上的问题。

2）中间件模式

是目前比较流行的数据集成方法，它通过在中间层提供一个统一的数据逻辑视图来隐藏底层的数据细节，使得用户可以把集成数据源看为一个统一的整体。这种模型下的关键问题是如何构造这个逻辑视图并使得不同数据源之间能映射到这个中间层。

通过统一的全局数据模型来访问异构的数据库、遗留系统、Web 资源等。中间件位于异构数据源系统(数据层)和应用程序(应用层)之间，向下协调各数据源系统，向上为访问集成数据的应用提供统一数据模式和数据访问的通用接口。各数据源的应用仍然完成它们的任务，中间件系统则主要集中为异构数据源提供一个高层次检索服务。

3）数据仓库

数据仓库是在企业管理和决策中面向主题的、集成的、与时间相关的和不可修改的数据集合。其中，数据被归类为广义的、功能上独立的、没有重叠的主题。这几种方法在一定程度上解决了应用之间的数据共享和互通的问题，但也存在以下的异同：联邦数据库系统主要面向多个数据库系统的集成，其中数据源有可能要映射到每一个数据模式，当集成的系统很大时，对实际开发将带来巨大的困难。

数据仓库技术在另外一个层面上表达数据之间的共享，它主要是为了针对企业某个应用领域提出的一种数据集成方法，也就是我们在上面所提到的面向主题并为企业提供

数据挖掘和决策支持的系统。

2. 数据高速缓存器是关键

对数据集成体系结构来说，关键是拥有一个包含有目标计划、源-目标映射、数据获得、分级抽取、错误恢复和安全性转换的数据高速缓存器。此外，数据高速缓存器包含有预先定制的数据抽取工作，这些工作自动位于一个企业的后端及数据仓库之中。

一个高速缓存器作为企业和电子商务数据的一个单一集成点，最大限度地减少了对直接访问后端系统和进行复杂实时集成的需求。这个高速缓存器从后端系统中卸载众多不必要的数据请求，因此使电子商务公司可以增加更多的用户，同时让后端系统从事其指定的工作。

数据集成软件与企业应用集成厂商和程序集成商进行联合，而不是取代它们。的确，由于数据集成软件越来越普遍地被用来作为 B2B 集成的一个工具，它会引人注目地改造 B2B 集成商一起合作的方式以及企业向 Internet 迁移的方式。

3. 数据集成对于企业信息系统的作用

数据集成的出现使企业能够将后端的 ERP 信息迁移到 Internet 上。数据集成产品在一个公司的 Internet 计算机与 SAP、Oracle 和 PeopleSoft 等公司的后端系统之间提供“高速缓存”或数据分级。

数据集成提供了在一个企业主计算机上存储的后端信息的一个镜像。当一个 Internet 客户需要检查一项订单的状态时，这项查询就被转移到数据集成软件。因此，并非总需要访问该企业的主计算机。数据集成软件拥有足够的智能，知道什么时候与主计算机保持同步以便使数据不断更新。为电子商务应用集成 ERP 数据是通过数据分级和直接访问 ERP 数据这两者的结合来完成的，它包括使用一个数据服务器和一些数据高速缓存器。数据集成软件以智能方式将直接实时的和分批的数据存取方法混合起来，以便从一个 ERP 系统中抽取数据。

数据从一个或多个源前进到一个或多个目标表以及信息类型（如 XML），数据移动的步骤包括确定应该从中抽取数据的源、数据应当进行的转换以及向什么地方发送数据。用户通过一个图形用户接口来指定数据映射和转换。

由用户定义的程序控制每一块数据的移动并确定这种移动之间的内部相关性。例如，如果一个目标表依靠其他目标表的值，则使用一些程序来指定一个数据服务器应当按什么次序来管理这些目标表中的单个数据移动。数据移动可以被设计来以批量方式或实时方式运行，并由管理员来创建和管理，以控制 ERP、电子商务、客户关系管理、供应链管理以及通信应用之间的数据移动。

数据移动使用分布式查询优化、多线程、存储器内数据转换和并行流水线操作来提供很高的数据通过量和可伸缩性。例如，要管理抽取程序并从 SAP 软件中来执行批量数据抽取，可使用优化的 ABAP 代码（SAP 的专有编程语言），不需要开发和维护定制的 ABAP 代码。

数据集成是企业进一步发展面临的问题。通过数据模型建模和相关应用技术在企业信息集成应用上做了一定的分析。在有效应用模型设计思想开发应用的同时，应重点把

握以下几点。

(1) 模型的时效性：包括开发期模型和运行期模型，而运行期模型则显示了模型驱动的核心思想。

(2) 模型的进化性：它揭示了模型是否可以根据应用的变化而自我进行改变。

(3) 模型的层级性：随着系统的复杂性增加，模型可以由多层级构成。

4. 传统数据集成方法的不足

传统数据集成方法存在不足之处。它们不能解决当今 IT 环境的复杂性，也不能覆盖 IT 必须执行的一系列方案的处理。

对于连接数百(或数千)个应用程序的不同单点解决方案，它们仅仅分裂运营数据并将其锁定在部门应用程序中，例如 ERP 和 CRM。以应用程序为中心的数据集成方法没有考虑所有企业数据。例如，它们不能处理计划数据，这些计划数据通常保存在 Excel 电子数据表中，而未保存在部门数据库应用程序中。它们也不能解决驻留在企业外部的有关 BPO 或 SaaS 供应商的数据或与贸易合作伙伴共享的数据。

手动编码数据集成方法也不起作用。手动编码费时费力，并且还容易犯错。由于 IT 机构力求管理更多的数据和更多的数据格式，手动编码通常导致更复杂——而不是更简单。它会增加维护成本并使 IT 效率下降。

在数据质量方面的表现如何？传统数据集成方法无法保证所有数据(客户数据、物料与资产数据以及财务数据)保持完整、一致、准确和最新，而无论数据驻留于何处。

如果继续采用传统方法进行数据集成，即按部门、按应用程序或按数据库，在“孤岛”中进行数据集成，那么有可能需要花费更多时间和金钱来管理复杂情况并“保持业务持续运转”，而不是集中精力来处理新的业务规则。

5. 新的数据集成方法的特点

IT 机构需要采用可靠的新方法进行数据集成，这些新方法可以完成如下工作：

- 集成企业内的所有内部预置数据孤岛，包括非结构化数据。
- 集成云计算应用程序和系统中的外部数据。
- 与贸易合作伙伴之间以企业对企业的形式无缝交换数据。
- 确保所有数据的质量。
- 经济高效地管理应用程序生命周期。

数据集成平台是一整套全面的技术，包括访问、发现、清洗、集成并为扩张的企业提供数据。数据集成平台支持各种数据集成项目，例如，数据仓库、数据迁移、测试数据管理、数据存档、数据整合、主数据管理、数据同步、B2B Data Exchange。

6. 理想的数据集成平台

数据集成平台必须解决企业间数据碎片的问题，以更快地做出数据驱动型业务决策和更有效地进行业务运作。它必须作为企业技术基础提供服务，提供容易掌控的方法来集成数据。

要满足这些需求，数据集成平台必须具备四个特性：全面、统一、开放和经济。

1）全面

理想的数据集成平台必须具备全面的功能集，使 IT 机构可以根据要求随时随地为企业提供可以信赖的数据。借助一整套可随意支配的数据集成功能，IT 机构的生产效率可以获得数十倍的提升。

2）支持完整的数据集成生命周期

数据集成平台必须支持数据集成生命周期中的所有五个关键步骤：访问、发现、清洗、集成和交付（见图 5.4）。

图 5.4　数据集成生命周期

第 1 步：访问。

大多数机构的数据存储在数千个位置，不只限于企业内部，还存放在防火墙外的业务合作伙伴或 SaaS 供应商的“云”中。无论何种来源或结构，所有数据都必须可以接受访问。必须从隐秘的大型主机系统、关系数据库、应用程序、XML、消息甚至从电子数据表之类的文档中提取数据。

第 2 步：发现。

数据源——特别是记录不详尽或来源未知——必须探查才能了解其内容和结构。需要推断数据中隐含的模式和规则。必须标记潜在的数据质量问题。

第 3 步：清洗。

必须清洗数据以确保其质量、准确性和完整性。必须解决错误或疏漏问题。必须强制执行数据标准，并且对值进行验证。必须删除重复的数据条目。

第 4 步：集成。

要跨越多个系统保持一致的数据视图，必须集成并转换数据，以便协调不同系统在定义各种数据元素并使之结构化的方式上存在的差异。例如，对于“客户盈利”，营销系统和财务系统可能具有完全不同的业务定义和数据格式，这些差异必须得到解决。

第 5 步：交付。

必须以适当的格式、在适当的时间将适当的数据交付给所有需要数据的应用程序和用户。交付数据的范围涵盖从支持实时业务运营的单个数据元素或记录到用于趋势分析和企业报告的数百万个记录。必须确保数据的高可用性和交付安全性。

此外，数据集成平台还必须支持如下各部分工作：

（1）审计、管理和监控。

数据管理员和 IT 管理员需要协作进行审计、管理和监控数据。不断地对关键指标