火龙果•整理
uml.org.cn

第三章 大数据技术体系现状

根据大数据处理的生命周期，大数据的技术体系通常可以分为大数据采集与预处理，大数据存储与管理，大数据计算模式与系统，大数据分析与挖掘，大数据可视化计算以及大数据隐私与安全等几个方面。

3.1 大数据采集与预处理

3.1.1 问题与挑战

根据MapReduce产生数据的应用系统分类，大数据的采集主要有四种来源：管理信息系统、Web 信息系统、物理信息系统、科学实验系统。

1. **管理信息系统**是指企业、机关内部的信息系统，如事务处理系统、办公自动化系统，主要用于经营和管理，为特定用户的工作和业务提供支持。数据的产生既有终端用户的原始输入，也有系统的二次加工处理。系统的组织结构上是专用的，数据通常是结构化的。
2. **Web 信息系统**包括互联网上的各种信息系统，如社交网站、社会媒体、搜索引擎等，主要用于构造虚拟的信息空间，为广大用户提供信息服务和社交服务。系统的组织结构是开放式的，大部分数据是半结构化或无结构的。数据的产生者主要是在线用户。电子商务、电子政务是在 Web 上运行的管理信息系统。
3. **物理信息系统**是指关于各种物理对象和物理过程的信息系统，如实时监控、实时检测，主要用于生产调度、过程控制、现场指挥、环境保护等。系统的组织结构上是封闭的，数据由各种嵌入式传感设备产生的，可以是关于物理、化学、生物等性质和状态的基本测量值，也可以是关于行为和状态的音频、视频等多媒体数据。
4. **科学实验系统**，实际上也属于物理信息系统，但其实验环境是预先设定的，主要用于研究和学术，数据是有选择的、可控的，有时可能是人工模拟生成的仿真数据。

在物理信息系统中，对于一个具体的物理对象，可采用不同观测手段，对其

火龙果•整理
uml.org.cn

不同的属性（方面）进行测量，如测量一辆行驶汽车的尺寸、速度、路线、尾气、外观等，其观测结果为具有不同形式的数据，这些数据代表实体不同的模态，称为多模态(multi-modal)。对于一个实体的多模态原始数据，需要做融合处理(data fusion)。在融合处理中，需要减少误差，保证数据的完整性和正确性。在高级的嵌入式系统或数据采集系统中，通常具有数据质量控制和数据融合处理功能[2]。

从人-机-物三元世界观点看，管理信息系统和 Web 信息系统属于人与计算机的交互系统，物理信息系统属于物与计算机的交互系统。关于物理世界的原始数据，在人-机系统中，是通过人实现融合处理的；而在物-机系统中，需要通过计算机等装置做专门的处理。融合处理后的数据，被转换为规范的数据结构，输入并存储在专门的数据管理系统中，如文件或数据库，形成专门的数据集。

对于不同的数据集，可能存在不同的结构和模式，如文件、XML 树、关系表等，表现为数据的异构性（heterogeneity）。对多个异构的数据集，需要做进一步集成处理（data integration）或整合处理（data consolidation），将来自不同数据集的数据收集、整理、清洗，转换后，生成到一个新的数据集，为后续查询和分析处理提供统一的数据视图。

通常大数据描述了一个对象(物理的或逻辑的)或一个过程的全景式的和全周期的状态，因此，其来源必然是多源的，其形式是多模态的。数据的多源和多模态的不确定性和多样性，必然导致数据的质量存在差异，严重影响到数据的可用性。由于数据量的大规模性，即使错误数据的相对比例不大，而绝对的错误数据量也是非常可观的。据国际咨询机构调查，全球财富 1000 强企业中 25%以上的企业信息信息系统存在不正确的数据，美国企业信息系统中 1%-30%的数据存在各种错误，美国工业企业由于数据错误而引起的生产事故和决策错误，每年造成 6000 多亿美元的损失[3]。

数据的可用性取决于数据质量。数据质量的定义有很多说法。按照文献[4]的定义，数据质量包含 5 种特性：精确性、一致性、完整性、同一性和实效性。精确性指数据符合规定的精度，不超出误差范围；一致性指数据之间不能存在相互矛盾；完整性指数据的值不能为空；同一性指实体的标识是唯一的；时效性指数据的值反映了实际的状态。此外，考虑到人为因素，还可以要求第 6 个性质，真实性，即数据不能是人工伪造的。

火龙果•整理
uml.org.cn

3.1.2 主要进展

针对管理信息系统中异构数据库集成技术、Web 信息系统中的实体识别技术和 DeepWeb 集成技术、传感器网络数据融合技术已经有很多研究工作，取得了较大的进展，已经推出了多种数据清洗和质量控制工具[31]，例如，美国 SAS 公司的 Data Flux，美国 IBM 公司的 Data Stage，美国 Informatica 公司的 InformaticaPower Center。

但是，针对各种类型、各种应用的大数据的特点，如何保证一致性、精确性、完整性、统一性、时效性、真实性等六个性质，并且保证可行的处理效率，还缺乏全面系统的研究，许多新问题有待于发现和解决。

3.1.3 发展趋势

为了保证大数据的可用性，首先必须在数据的源头上把好质量关，做好从原始数据到高质量信息的预处理。具体的关键技术有：

（1） **数据源的选择和高质量原始数据的采集方法。**用于从可靠的高质量数据源里，获得高质量的原始数据。为了确保数据源的质量，需要建立数据源的质量评估理论模型，包括数据源的综合质量评估和高质量数据源的选择方法。然后，针对各种模态数据的特点，建立高质量多模态数据的获取方法，包括：有效的数据采集方法、多模态数据融合算法、数据的保质转换算法、数据精确性和一致性方面的错误校验和纠错、数据完整性方面的缺失值估计、数据的时效性检测、数据的真实性验证等。

（2） **多源数据的实体识别和解析方法。**用于识别和合并相同的实体，区分不同的实体。为了高质量的数据集成奠定基础，必须保证数据的实体同一性，解决来自多个数据源的多模态数据的实体识别问题。需要建立多源数据的实体关联模型和识别模型、多源多模态数据的实体自动识别方法、实体识别效果的评估模型等。

（3） **数据清洗和自动修复方法。**根据正确性条件和数据约束规则，清除不合理和错误的数据，对重要的信息进行修复，保证数据的完整性。需要建立数据正确性语义模型、关联模型和数据约束规则、数据错误模型和错误识别学习框架、针

[31] Gartner 研究报告, Magic Quadrant for Data Quality Tools, 2012 年 8 月，http://useready.com/wp-content/uploads/2013/07/Gartner-Data-Quality-2012.pdf

火龙果•整理
uml.org.cn

对不同错误类型的自动检测和修复算法、错误检测与修复结果的评估模型和评估方法等。

（4） **高质量的数据整合方法。**在数据采集和实体识别的基础上，进而实现数据到信息的高质量整合。需要建立多源多模态信息集成模型、异构数据智能转换模型、异构数据集成的智能模式抽取和模式匹配算法、自动的容错映射和转换模型及算法、整合信息的正确性验证方法、整合信息的可用性评估方法等。

（5） **数据演化的溯源管理。**用于对数据的演化过程进行跟踪和记录，以保证和控制数据的质量。需要建立世系模型及其追踪技术，主要包括时空、多粒度、多路径和不确定的海量信息演化的演化模型和演化描述方法、演化模式的正向性评估模型与方法、演化的可逆性判定与近似求解算法、分布式、多粒度、概率化的世系追踪技术等。

总之，大数据的采集和预处理是大数据的源头，在源头上把好质量关，对大数据的后续处理和分析至关重要。因此，对大数据的使用者、研究者、开发者以及上级主管部门，提出如下建议：

（1） 提高用户对大数据可用性的重要性的认识，切实开展大数据质量控制，确保大数据处理和分析结果的正确性；

（2） 针对大数据质量控制面临的挑战性问题，学术界应加强对大数据可用性评估和保证的关键技术的研究和开发；

（3） 大数据的质量控制具有广泛的需求和巨大的市场前景，工业界应注重大数据可用性的评估，加强数据质量保证软件的开发和推广。

（4） 建议政府有关部门尽快建立关于大数据可用性（数据质量）的标准，保证大数据的统一质量，有效保证大数据的利用价值。

3.2 大数据存储与管理

3.2.1 问题与挑战

大数据给存储系统带来了 3 个方面的挑战：1）存储规模大，通常达到 PB（1,000 TB）甚至 EB（1,000 PB）量级。2）存储管理复杂，需要兼顾结构化、非结构化和半结构化的数据。3）数据服务的种类和水平要求高，换言之，上层

火龙果•整理
uml.org.cn

应用对存储系统的性能、可靠性等指标有不同的要求，而数据的大规模和高复杂度放大了达到这些指标的技术难度。这些挑战在存储领域并不是新问题，但在大数据背景下，解决这些问题的技术难度成倍提高，数据的量变终将引起存储技术的质变。

大数据环境下的存储与管理软件栈，需要对上层应用提供高效的数据访问接口，存取 PB 甚至 EB 量级的数据，并且能够在可接受的响应时间内完成数据的存取，同时保证数据的正确性和可用性；对底层设备，存储软件栈需要充分高效的管理存储资源，合理的利用设备的物理特性，以满足上层应用对存储性能和可靠性的要求。在大数据带来的新挑战下，要完成以上这些要求，需要更进一步的研究存储与管理软件技术[5]。

3.2.2 主要进展

根据为上层应用提供的访问接口和功能侧重不同，存储与管理软件主要包括文件系统和数据库；在大数据环境下，目前最适用的技术是分布式文件系统、分布式数据库以及以及访问接口和查询语言。

- **分布式文件系统**

分布式文件系统所管理的数据存储在分散的设备或节点上，存储资源通过网络连接。用分布式文件系统对大数据进行存储与管理，目前的研究，主要涉及到以下几个关键的技术：

（1） **高效元数据管理技术。**大数据应用下，元数据的规模也非常大，元数据的存取性能是整个分布式文件系统性能的关键。常见的元数据管理可以分为集中式和分布式元数据管理架构。集中式元数据管理架构采用单一的元数据服务器，优点是实现简单，但存在单点故障等问题。分布式元数据管理架构[6]则将元数据分散在多个结点上，从而解决了元数据服务器性能瓶颈问题，提高了可扩展性，但实现复杂，并引入了元数据一致性的问题。此外，还有一种无元数据服务器的分布式架构，使用在线算法组织数据，不需要专用的元数据服务器。但是该架构对数据一致性的保证很困难，实现复杂。文件目录遍历操作的效率低下，并且缺乏文件系统全局监控管理功能。

（2） **系统弹性扩展技术。**大数据环境下，数据规模和复杂度的增加往往非

火龙果•整理
uml.org.cn

常迅速，所以按需的扩展系统规模，是十分必要的。实现存储系统的高可扩展性首先要解决两个方面的重要问题，元数据的分配和数据的透明迁移。前者主要通过静态子树划分和动态子树划分技术实现，后者则侧重数据迁移算法的优化。此外，大数据存储系统规模庞大，结点失效率高，因此还需要实现一定程度上的自适应管理功能。系统必须能够根据数据量和计算的工作量估算所需要的结点个数，并动态地将数据在结点间迁移，以实现负载均衡；同时，结点失效时，数据必须可以通过副本等机制进行恢复，不能对上层应用产生影响。

（3） **存储层级内的优化技术。**构建存储系统时，需要基于成本和性能来考虑，因此存储系统通常采用多层不同性价比的存储器件组成存储层次结构。大数据的规模大，因此构建高效合理的存储层次结构，可以在保证系统性能的前提下，降低系统能耗和构建成本。利用数据访问局部性原理，可以从两个方面对存储层次结构进行优化。从提高性能的角度，可以通过分析应用特征，识别热点数据并对其进行缓存或与预取，通过高效的缓存预取算法和合理的缓存容量配比，以提高访问性能。从降低成本的角度，采用信息生命周期管理方法，将访问频率低的冷数据迁移到低速廉价存储设备上，可以在小幅牺牲系统整体性能的基础上，大幅降低系统的构建成本和能耗。

（4） **针对应用和负载的存储优化技术。**传统数据存储模型需要支持尽可能多的应用，因此需要具备较好的通用性。大数据具有大规模、高动态及快速处理等特性，通用的数据存储模型通常并不是最能提高应用性能的模型，而大数据存储系统对上层应用性能的关注远超过对通用性的追求。针对应用和负载来优化存储，就是将数据存储与应用耦合，放宽 POSIX 接口，简化或扩展分布式文件系统的功能，根据特定应用、特定负载、特定的计算模型对文件系统进行定制和深度优化，使应用达到最佳性能。这类优化技术在 Google[7]、Facebook[8]等互联网公司的内部存储系统上，管理超过 PB 级的大数据，能够达到非常高的性能。

（5） **针对存储器件特性的优化技术。**随着新型存储器件的发展和成熟，Flash、PCM 等逐渐开始在存储层级中占据一席之地，存储软件栈也随之开始逐渐发生变化。以 Flash 为例，起初各厂商通过闪存转换层 FTL 对新型存储器进行封装，以屏蔽存储器件的特性，适应存储软件栈的现有接口。但是随着 Flash 的普及，产生了许多针对应用对 FTL 进行的优化[9][10]，以及针对 Flash 特性进行

火龙果•整理
uml.org.cn

定制的文件系统，甚至有去掉FTL这层冗余直接操作Flash的存储解决方案[11]。传统的本地文件系统，包括分布式文件系统，是否能够与新型存储器件耦合，最大程度地利用这些存储器件新特性上的优势，需要存储软件开发者重新审视存储软件栈，去除存储软件栈的冗余，甚至需要修复一些不再合适的部分[12]。

➢ 分布式数据库

大数据时代企业对数据的管理、查询及分析的需求变化促生了一些新的技术的出现。需求的变化主要集中在数据规模的增长，吞吐量的上升，数据类型以及应用多样性的变化。数据规模和吞吐量的增长需求对传统的关系型数据库管理系统在并行处理，事务特性的保证，互联协议的实现，资源管理以及容错等各个方面带来了很多挑战。而数据类型以及应用的多样性带来了为了支持不同应用的数据管理系统。

● 事务性数据库

这类数据库主要包括 NoSQL 和 NewSQL。NoSQL（“Not Only SQL”或者“Not Relational”）系统往往通过放松对事务 ACID 语义的方法来增加系统的性能以及可扩展性（CAP 定理）。NoSQL[32]系统往往具有以下几个特征：

（1） 非关系数据模型，比如键值存储等

（2） 对简单操作比如键值查询的水平可扩展性，往往不支持 SQL 全集。

（3） 在多个节点中分割和复制数据的能力

（4） 弱并发一致性语义（比如最终一致性）

（5） 充分利用分布式索引和内存

根据管理数据的模式分类，NoSQL 系统可以分为三类：键值系统，文档存储系统以及图数据库。键值系统的代表性系统包括 BigTable，Dynamo, HBase，Gemfire, Redis,, Cassandra, 文档存储系统的代表包括 MongoDB 和 Couchbase，图数据库的代表是 Neo4j 等等。

NoSQL 系统通过对事务语义的放松达到系统的可扩展性，但是把一致性的维护交由用户来管理，这对很多对一致性要求不高的应用来说是足够的。但是如果应用需要保证一致性，这对开发人员来说就很困难了。NewSQL 是在这样的背景下诞生的。NewSQL 系统可以在提供类似 NoSQL 的可扩展性的同时保证事务

[32] NoSQL. http://en.wikipedia.org/wiki/NoSQL

火龙果•整理
uml.org.cn

ACID 属性，并且提供 SQL 用户接口。NewSQL 系统通常可以分为两类。

（1） 通用数据库：这类系统保持传统分布式数据库的功能，但是在设计分布式体系架构时充分考虑了大规模高吞吐系统的特性。这类系统的典型代表是 Spanner 和 NuoDB。

（2） 基于内存的数据库：这类系统基本上针对的是高吞吐短小事务，不再采用传统的关系型数据库设计。这类数据库的典型代表是 SQLFire 和 VoltDB。

- **分析型数据库**

分析型数据库在大数据时代也呈现了一种百家争鸣的局势。自从 MapReduce[13]被提出以及 Hadoop[33]的流行，出现了多家针对 Hadoop 的 SQL 分析引擎。代表性系统包括 Hive，HAWQ， Impala 和 Hadapt。

Hive[34]是一个基于 MapReduce 的 SQL 引擎。基本原理是接受 SQL，解析 SQL，然后把 SQL 语句翻译成多个 MapReduce 的任务，通过 MapReduce 来实现基本的 SQL 操作。因为 Hive 基于 MapReduce，所以它把容错，执行以及资源管理的工作都交给了 MapReduce 框架，特点是简单与易于实现。但是它也有一些不可避免的缺陷，包括对标准 SQL 以及实时查询的支持，难于优化带来的查询性能低下，并且很难充分利用整个集群的资源，从而导致并发吞吐量较低。

HAWQ[35]（Hadoop with Query）是 Hadoop 领域与 SQL 兼容的大规模数据分析引擎。HAWQ 继承了 Hadoop 与 MPP 大规模数据库分析引擎的优点，实现了 HDFS 分布式存储与 MPP 执行引擎的结合。HAWQ 实现了 MPP 基于统计的优化器，支持数百万连接的网络互联协议，数据的多级划分与存储和高效的执行引擎。特点是与各种 BI 工具的兼容，实时查询的支持，以及与基于 MapReduce 系统相比的性能优势。

Impala 和 Hadapt 是另外两个基于 Hadoop 的 SQL 引擎。基本的出发点也是把 MPP 的技术引入 Hadoop。但目前还不是很成熟。

➢ 访问接口和查询语言

大数据系统的访问接口和查询语言取决于系统的存储模型。传统的 MPP 数据库都使用关系模型，其查询语言为标准的 SQL。而图数据库有自己的查询语

[33] Apache Hadoop. http://hadoop.apache.org/
[34] Apache Hive. http://hive.apache.org/
[35] EMC Pivotal Initiative. Pivotal HD: HAWQ. 2013. http://www.gopivotal.com/sites/default/files/Hawq_WP_042313_FINAL.pdf

火龙果•整理
uml.org.cn

言，可以实现子图匹配、路径查询等功能。Hadoop 本身使用的是 HDFS，MapReduce 编程接口可以作为其访问接口。构建在 Hadoop 之上的类数据库系统则提供各自存储模型所对应的查询语言和访问接口。例如，HBase 提供 API，用于对数据表进行 key-value 形式的查询和增删改。Hive 则提供称为 HiveQL 的查询语言，用于对关系表进行查询，HiveQL 同 SQL 非常相似，并附带一些 SQL 未提供的功能。为了方便对 hadoop 的使用，一系列的查询语言和附加访问接口被提出。Pig 是一种基于 MapReduce 的编程平台，它的访问语言 Pig Latin 是介于 SQL 和过程式程序设计语言之间的语言，结合了 SQL 申明式（declarative）语言的优势以及过程式程序设计的灵活性，被众多程序设计者所热衷。Sqoop 是一种用于在关系数据库和 hadoop 之间进行数据迁移的命令式语言。Mahout 则是构建在 hadoop 之上的机器学习引擎，也拥有自己的一套访问接口。

3.2.3 发展趋势

（1） 大数据索引和查询技术

索引和查询技术是数据处理系统的重要入口之一，近年来随着数据量（Volume）、数据处理速度（Velocity）和数据多样性（Variety）的快速发展，大数据相关的索引和查询技术做为大数据的主要入口之一也变得更为重要。传统的索引和查询技术虽然不能很好的解决大数据带来的挑战，然而其核心技术例如数据库和数据挖掘系统中使用的经典索引例如哈希索引、B 树索引、位图索引和 R 树索引，信息检索系统中的倒排索引等依然是大数据索引和查询系统的基石。

大数据带来的主要挑战是其庞大的数据量，单个节点不能或者无法有效地处理这种数量级的数据。此外数据增长速度非常快，这要求系统不但能处理已有的大数据，还要能快速的处理新数据。这些特征使得我们需要考虑很多在大数据环境中独有的因素来开发和选择大数据索引和查询技术。

分布式是处理大数据的一个基本思路，这同样适用于大数据索引和查询系统。分布式索引把全部索引数据水平切分后存储到多个节点上，这可以很好的解决两个问题：1）单个节点无法存储庞大的索引数据；2）单个节点构建索引的效率瓶颈。当业务增长，需要索引更多的数据或者更快的索引数据时，可以通过水平扩展增加更多的节点来解决。切分数据的方式有多种，常见的方法有随机方法、哈

火龙果•整理
uml.org.cn

希方法和区间方法。随机方法将所有数据随机分布到不同的节点，这种方法不支持更新操作。哈希方法根据某个列或者某些列（称为分布键）的哈希值将数据分布到不同的节点。区间方法将所有的数据按照不同区间分布到不同的节点。区间到节点的映射信息需要保存下来。不管使用什么样的切分方法，都需要注意数据分布的均匀性，避免大量数据分布到一个或者几个节点上，这样就失去了分布式计算的优势，因而对算法的选择和设计有一定要求。另外分布键的选择也很重要，好的分布键能将数据相对均匀的分布到不同的节点，从而达到负载均衡的目的。

由于索引数据是分布在不同的节点上，因而查询也是分布式的。所有节点或者部分节点的查询结果由主节点（主从架构）或者查询节点（点对点架构）进行汇总，然后得到最终结果。不同的分布式系统支持不同类型的查询语言和查询能力。分布式数据库系统支持 SQL 查询； NoSQL 产品类型和功能各异，有的仅支持主键查询，有的支持范围查询，有的还支持有限的 JOIN；全文检索系统的查询语法最为灵活，但通常不支持 JOIN 或者有限的支持 JOIN。

当一个节点故障时，将无法访问该节点上的数据。为了提高可用性，防止单点故障，通常使用镜像技术或者保存多个副本到不同的节点上。副本可以使用不同的分布策略，例如基于 Hadoop 的系统通常有 2 个副本，一个副本在同机架上的其他节点，另一个副本在其他机架的节点上。这样一方面可以有效利用数据局部性原理改进性能，另一方面可以最大化的保证数据的可用性。有些系统副本仅仅起数据备份的作用，这种类型的副本不能接受查询请求，主要目的是提高系统的可靠性。有的系统的副本还可以处理用户查询请求，从而实现负载均衡以最大化的利用系统资源。然而副本的引入也大大增加了系统的复杂性，因为分布式环境下任何一个节点可能在任何时刻出错：网络可能故障，磁盘可能故障，系统可能崩溃。多数系统采取保证数据高度一致性的策略：只有主副本接受写请求，然后通过文件块复制或者写管道将数据写入到其他副本。也有一些 NoSQL 系统采用最终一致性策略，这种策略中在某一个时刻数据在不同的副本上可能是不一致的，但是当没有对该数据的更新时，最终的访问将返回该数据的最新值。

当系统不能适应业务的需求时，需要对系统进行动态扩容，这通常需要进行数据的再分布，即根据新系统中节点的个数按照数据分布策略重新对数据进行分布。当数据量庞大时，扩容可能需要较多的时间。为了降低需要移动的数据量，

火龙果•整理
uml.org.cn

可以采取某些算法来实现，例如一致性哈希算法。

目前各大数据库厂商例如 Oracle、IBM、Greenplum 都已经有支持分布式索引和查询的产品，很多 NoSQL 数据库例如 Mongodb、HBase、Cassandra 也支持分布式索引和查询。还有很多面向全文检索的产品例如 Solr、ElasticSearch、Sphinx 均支持分布式全文索引和查询，且这些产品都是开源的。值得一提的是 Greenplum 的 GPText 将 Solr 的全文检索能力引入到了 Greenplum 数据库之中，使得它可以同时支持 SQL 和 Solr 的全文检索。

（2） 实时/流式大数据存储与处理

随着业务的增长，业界对大数据的速度（Velocity）维度越来越关注，过去需要几天或者几个小时才能回答的问题现在期望在几分钟、几秒甚至毫秒内得到解决。实时流数据存储和处理技术将会越来越多的被研究和开发。实时流式大数据的处理在很多方面和分布式系统在原理上有很多相似之处，然而也有其独特需求。

实时流数据处理系统包括流数据的实时存储和流数据的实时计算。流数据存储指的是快速高效的存储流式数据到数据库、数据仓库或者数据湖中；流数据的实时计算注重对流数据的快速高效处理、计算和分析。

（1） 数据流加载：实时流式大数据系统中，数据通常以流的方式进入系统。如何高效且可靠地将数据加载到大数据存储系统中成为流式大数据系统实现低延迟处理的基础。此外能够重新处理数据流中的数据也是一个很有价值的特性。

（2） 复杂事件处理（CEP）：数据流中的数据源是多种多样的，数据的格式也是多种多样，而数据的转换、过滤和处理逻辑更是千变万化，因而需要强大而又灵活的复杂事件处理引擎来适应各种场景下的需求。

（3） 高可用性：数据通过复杂处理引擎和流计算框架时，通常会经过很多步骤和节点，而其中任何一步都有出错的可能，为了保证数据的可靠性和精准投递，系统需要具有容错和去重能力。

（4） 流量控制和缓存：整个流系统可能有若干个模块，每个模块的处理能力和吞吐量差别很大，为了实现总体高效的数据处理，系统需要对流量进行控制和动态节点增加和删除的能力。当数据流入大于流出的速度时，还需要有一定的缓存能力，如果内存不足以缓存快速流入的数据时，需要能够持久化到存储层。

目前市场上已经出现了多种大数据实时处理技术，它们各有不同的侧重点，

火龙果•整理
uml.org.cn

例如数据传输技术有 Flume, Scribe, Kafka, Sqoop 等，计算框架有 Storm，S4，Spark 等。基于 Hadoop 的 SQL 处理引擎有 Impala， HAWQ 等。另外还有一些产品在大数据流计算框架之上提供分析即服务，例如 Cetas。大数据的实时存储与处理还有很多需要研究和解决的问题。

3.3 大数据计算模式与系统

3.3.1 问题与挑战

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能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）。例如，MapReduce 是一个并行计算抽象[13]，Bekerley 大学著名的 Spark 系统中的“分布内存抽象 RDD（RDD, a distributed memory abstraction[14]）”，CMU 著名的图计算系统 GraphLab 中的“图并行抽象”（Graph Parallel Abstraction[15]）等。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。

为了能更清晰地理解不同的大数据计算模式，首先需要梳理出大数据处理中主要的数据特征和计算特征维度，在此基础上进一步梳理目前出现的各种重要和典型的大数据计算模式。大数据处理包括以下典型的特征和维度：

（1） **数据结构特征**：根据数据结构特征大数据可分为结构化/半结构化数据处理与非结构化数据处理。

火龙果•整理
uml.org.cn

（2） **数据获取处理方式**：按照数据获取方式，大数据可分为批处理与流式计算（streaming）方式。

（3） **数据处理类型**：从数据处理类型来看，大数据处理可分为传统的查询分析计算和复杂的数据挖掘分析计算。

（4） **实时性或响应性能**：从数据计算响应性能角度看，大数据处理可分为实时/准实时与非实时计算，或者是联机（online）计算与线下（offline）计算。流式计算通常属于实时计算，查询分析类计算通常也要求具有高响应性能，而批处理和复杂数据挖掘计算通常属于非实时或线下计算。

（5） **迭代计算**：现实的数据处理中有很多计算问题需要大量的迭代计算（如一些机器学习算法），为此需要提供具有高效的迭代计算能力的计算模式。

（6） **数据关联性**：MapReduce 适用于处理数据关系较为简单的计算任务，但社会网络等具有复杂数据关系的计算任务则需要研究和使用图数据计算模式。

（7） **并行计算体系结构特征**：由于需要支持大规模数据的存储计算，大数据处理通常需要使用基于集群的分布式存储与并行计算体系结构和硬件平台。此外，为了克服传统的 MapReduce 框架在计算性能上的缺陷，人们从体系结构层面上提出了内存计算模式。

3.3.2 主要进展

根据大数据处理多样性的需求和以上不同的特征维度，目前出现了多种典型和重要的大数据计算模式。与这些计算模式相适应，出现了很多对应的大数据计算系统和工具[16]。由于单纯描述计算模式比较抽象和空洞，因此，在描述不同计算模式时，将同时给出相应的典型计算系统和工具，这将有助于对计算模式的理解以及对技术发展现状的把握，并进一步有利于在实际大数据处理应用中对合适的计算技术和系统工具的选择使用。

表3-1： 典型大数据计算模式与系统

典型大数据计算模式	典型系统
大数据查询分析计算	HBase，Hive，Cassandra，Impala，Shark，Hana 等
批处理计算	Hadoop MapReduce，Spark 等
流式计算	Scribe，Flume，Storm，S4，Spark Steaming 等
迭代计算	HaLoop，iMapReduce，Twister，Spark 等

火龙果•整理
uml.org.cn

图计算	Pregel，Giraph，Trinity，PowerGraph，GraphX 等
内存计算	Dremel，Hana，Spark 等

（1）大数据查询分析计算模式与典型系统

由于行业数据规模的增长已大大超过了传统的关系数据库的承载和处理能力，因此，目前需要尽快研究并提供面向大数据存储管理和查询分析的新的技术方法和系统，尤其要解决在数据体量极大时如何能够提供实时或准实时的数据查询分析能力，满足企业日常的经营管理需求[17]。然而，大数据的查询分析处理具有很大的技术挑战，在数量规模较大时，即使采用分布式数据存储管理和并行化计算方法，仍然难以达到关系数据库处理中小规模数据时那样的秒级响应性能。

大数据查询分析计算的典型系统包括 Hadoop[36]下的 HBase 和 Hive，Facebook 开发的 Cassandra，Google 公司的 Dremel[18]，Cloudera 公司的实时查询引擎 Impala；此外为了实现更高性能的数据查询分析，还出现了不少基于内存的分布式数据存储管理和查询系统，如 UC Berkeley AMPLab 的基于内存计算引擎 Spark[14]的数据仓库 Shark[19]，SAP 公司的 Hana 等。

（2）批处理计算模式与典型系统

最适合于完成大数据批处理的计算模式是 MapReduce[13]。MapReduce 是一个单输入、两阶段（Map 和 Reduce）的数据处理过程。首先，MapReduce 对具有简单数据关系、易于划分的大规模数据采用“分而治之”的并行处理思想；然后将大量重复的数据记录处理过程总结成 Map 和 Reduce 两个抽象的操作；最后 MapReduce 提供了一个统一的并行计算框架，把并行计算所涉及到的诸多系统层细节都交给计算框架去完成，以此大大简化了程序员进行并行化程序设计的负担。

MapReduce 的简单易用性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并行计算模式。在开源社区的努力下，开源的 Hadoop 系统目前已发展成为较为成熟的大数据处理平台，并已发展成一个包括众多数据处理工具和环境的完整的生态系统。目前几乎国内外的各个著名 IT 企业都在使用 Hadoop 平台进行企业内大数据的计算处理。Spark[14]也是一个批处理系统，其性能方面比 Hadoop MapReduce 有很大的提升，但是其易用性方面目前仍不如 Hadoop MapReduce。

[36] Apache Hadoop [EB/OL],http://hadoop.apache.org/ 2013,08,26

火龙果•整理
uml.org.cn

（3）流式计算模式与典型系统

流式计算是一种高实时性的计算模式，需要对一定时间窗口内应用系统产生的新数据完成实时的计算处理，避免造成数据堆积和丢失。很多行业的大数据应用，如电信、电力、道路监控等行业应用、以及互联网行业的访问日志处理，都同时具有高流量的流式数据和大量积累的历史数据，因而在提供批处理数据模式的同时，系统还需要能具备高实时性的流式计算能力。流式计算的一个特点是数据运动、运算不动，不同的运算节点常常绑定在不同的服务器上。

Facebook 的 Scribe 和 Apache 的 Flume 都提供了机制来构建日志数据处理流图。而更为通用的流式计算系统是 Twitter 公司的 Storm[37]、Yahoo 公司的 S4[20]、以及 UCBerkeley AMPLab 的 Spark Steaming[21]。

（4）迭代计算模式与典型系统

为了克服 Hadoop MapReduce 难以支持迭代计算的缺陷，业界和学术界对 Hadoop MapReduce 进行了不少改进研究。HaLoop[22] 把迭代控制放到 MapReduce 作业执行的框架内部，并通过循环敏感的调度器保证前次迭代的 Reduce 输出和本次迭代的 Map 输入数据在同一台物理机上，以减少迭代间的数据传输开销；iMapReduce[23]在这个基础上保持 Map 和 Reduce 任务的持久性，规避启动和调度开销；而 Twister[24]在前两者的基础上进一步引入了可缓存的 Map 和 Reduce 对象，利用内存计算和 pub/sub 网络进行跨节点数据传输。

目前，一个具有快速和灵活的迭代计算能力的典型系统是 UC Berkeley AMPLab 的 Spark，其采用了基于分布式内存的弹性数据集模型实现快速的迭代计算。

（5）图计算模式与典型系统

社交网络、Web 链接关系图等都包含大量具有复杂关系的图数据，这些图数据规模常达到数十亿的顶点和上万亿的边数。这样大的数据规模和非常复杂的数据关系，给图数据的存储管理和计算分析带来了很大的技术难题。用 MapReduce 计算模式处理这种具有复杂数据关系的图数据通常不能适应，为此，需要引入图计算模式。

大规模图数据处理首先要解决数据的存储管理问题，通常大规模图数据也需

[37] Storm[EB/OL], http://storm-project.net/ 2013.07.21

火龙果•整理
uml.org.cn

要使用分布式存储方式。但是，由于图数据的数据关系很强，分布存储就带来了一个重要的图划分问题（Graph Partitioning）。在有效的图划分策略下，大规模图数据得以分布存储在不同节点上，并在每个节点上对本地子图进行并行化处理。与任务并行和数据并行的概念类似，由于图数据并行处理的特殊性，人们提出了一个新的“图并行[4]”（Graph Parallel）的概念。目前已经出现了很多分布式图计算系统，其中较为典型的系统包括 Google 公司的 Pregel[25]，Facebook 对 Pregel 的开源实现 Giraph，微软的 Trinity[26]，Berkeley AMPLab 的 GraphX[27]，以及 CMU 的 GraphLab 以及由其衍生出来的目前性能最快的图数据处理系统 PowerGraph[15]。

（6）内存计算模式与典型系统

Hadoop MapReduce 为大数据处理提供了一个很好的平台。然而，由于 MapReduce 设计之初是为大数据线下批处理而设计的，随着很多需要高响应性能的大数据查询分析计算问题的出现，MapReduce 其在计算性能上往往难以满足要求。随着内存价格的不断下降以及服务器可配置的内存容量的不断提高，用内存计算完成高速的大数据处理已经成为大数据计算的一个重要发展趋势。Spark 则是分布内存计算的一个典型的系统，SAP 公司的 Hana 则是一个全内存式的分布式数据库系统。

3.3.3 发展趋势

近几年来，随着大数据处理和应用需求急剧增长，同时也由于大数据处理的多样性和复杂性，针对以上的典型的大数据计算模式，学术界和业界不断研究推出新的或改进已有的计算模式和系统工具平台，目前主要有以下三方面的重要发展趋势和方向。

（1）主流的 Hadoop 平台改进后将与其他计算模式和平台共存

由于 MapReduce 当初的设计目标主要是针对具有简单数据关系的大数据线下批处理，使得它在系统构架和计算性能上存在不少不足之处，难以适用于那些具有复杂数据关系和复杂计算模式（如迭代计算、图计算等）的大数据处理计算任务。但尽管如此，由于 Hadoop 生态系统已发展成为目前最主流的大数据处理

火龙果•整理
uml.org.cn

平台、并得到广泛的使用。考虑到兼容性，目前业界和学术界并不会完全抛弃 Hadoop 平台，而是试图不断改进和发展现有的平台，增加其对各种不同大数据处理问题的适用性。Hadoop 社区正努力扩展现有的计算模式框架和平台，以便能解决现有版本在计算性能、计算模式、系统构架和处理能力上的诸多不足，这正是目前 Hadoop 2.0 新版本“YARN”的努力目标。目前不断有新的计算模式和计算系统出现，预计今后相当长一段时间内，Hadoop 平台将与各种新的计算模式和系统共存，并相互融合，形成新一代的大数据处理系统和平台。

（2）混合计算模式将成为满足多样性大数据处理和应用需求的有效手段

现实世界大数据应用复杂多样，可能会同时包含不同特征的数据和计算，在这种情况下单一的计算模式多半难以满足整个应用的需求，因此需要考虑不同计算模式的混搭使用。

混合计算模式可体现在两个层面上：一是传统并行计算所关注的体系结构与低层并行程序设计语言层面计算模式的混合，例如，在体结构层，可根据大数据应用问题的需要搭建混合式的系统构架，如 MapReduce 集群+GPU-CUDA 的混合，或者 MapReduce 集群+基于 MIC（Intel Xeon Phi 众核协处理系统）的 OpenMP/MPI 的混合模型。

混合模式的另一个层面是大数据处理高层计算模式的混合。比如，一个大数据应用可能同时需要提供流式计算模式以便接受和处理大量流式数据，提供基于 SQL 或 NoSQL 的数据查询分析能力以便进行日常的数据查询分析，提供线下批处理和迭代计算已完成基于机器学习的深度数据挖掘分析；一些大数据计算任务可能还涉及到复杂图计算或者间接转化为图计算问题。因此，很多大数据处理问题将需要混合使用多种计算模式。此外，为了提高计算性能，各种计算模式还可以与内存计算模式混合，实现高实时性的大数据查询和计算分析。

混合计算模式之集大成者当属 UCBerkeley AMPLab 的 Spark 系统，其涵盖了几乎所有典型的大数据计算模式，包括迭代计算、批处理计算、内存计算、流式计算（Spark Streaming）、数据查询分析计算（Shark）、以及图计算（GraphX）。Spark 提供了一个强大的内存计算引擎，实现了优异的计算性能，同时还保持与 Hadoop 平台的兼容性。因此，随着系统的不断稳定和成熟，Spark 有望成为与 Hadoop 共存的新一代大数据处理系统和平台。

火龙果•整理
uml.org.cn

（3）内存计算将成为高实时性大数据处理的重要技术手段和发展方向

Hadoop 在处理大数据时计算性能不高、难以满足实时性或高响应性计算任务的要求，为此，人们一直努力改进 Hadoop 的计算性能。但是，在现有 Hadoop 平台面向大数据线下处理的基本构架和工作机制下，性能的改进和提升空间非常有限，难以逾越计算性能低下的障碍；而随着大数据的规模不断扩大，这个问题将越来越为突出。为此，目前已经逐步形成一个基本共识，即随着内存成本的不断降低，内存计算将成为最终跨越大数据计算性能障碍、实现高实时高响应计算的一个最有效技术手段。因此，目前越来越多的研究者和开发者在关注基于内存计算的大数据处理技术，不断推出各种基于内存计算的计算模式和系统。

内存计算是一种在体系结构层面上的解决方法，因此，它可以与各种不同的计算模式相结合，从基本的数据查询分析计算，到批处理和流式计算，再到迭代计算和图计算，都可以基于内存计算加以实现，因此我们可以看到各种大数据计算模式下都有基于内存计算实现的系统，比较典型的系统包括 SAP 的 Hana 内存数据库，微软的图数据计算系统 Trinity，UC Berkeley AMPLab 的 Spark 等。

由于优异的计算性能，内存计算将成为今后高实时性大数据处理的重要技术手段和发展方向。

3.4 大数据分析与挖掘

3.4.1 问题与挑战

大数据时代，不同领域不同格式的数据从生活的各个领域涌现出来。大数据往往含有噪声，具有动态异构性，是相互关联和不可信的。尽管含有噪声，大数据往往比小样本数据更有价值。这是因为从频繁模式和相关性分析得到的一般统计量通常会克服个体的波动，会发现更多可靠的隐藏的模式和知识。另一方面，互相连接的大数据形成大型异构信息网。通过信息网，冗余的信息可用于弥补数据缺失所带来的损失，可用于交叉核对数据的不一致性，进一步验证数据间的可信关系，并发现数据中隐藏的关系和模型。数据挖掘需要集成的、经过清洗的、可信的、可高效访问的数据，需要描述性查询和挖掘界面，需要可扩展的挖掘算法以及大数据计算环境。与此同时，数据挖掘本身也可以用来提高数据质量和可

火龙果•整理
uml.org.cn

信度，帮助理解数据的语义，提供智能的查询功能。只有能够鲁棒地进行大数据分析，大数据的价值才能发挥出来。另一方面，从大数据得出的知识有助于纠正错误，并消除歧义[38]。

大数据环境下的分析和挖掘方法与传统的小样本统计分析有着根本的不同，具有如下挑战：

（1） **数据量的膨胀**。随着数据生成的自动化以及数据生成速度的加快，数据分析需要处理的数据量急剧膨胀。一种处理大数据的方法是使用采样技术，通过采样，可以把数据规模变小，以便利用现有的技术手段进行数据管理和分析。然而在某些应用领域，采样将导致信息的丢失，比如 DNA 分析等。在明细数据上进行分析，意味着需要分析的数据量将急剧膨胀和增长[28]。如何对 TB 级的大数据进行分析是一大挑战；

（2） **数据深度分析需求的增长**。为了从数据中发现知识并加以利用进而指导人们的决策，必须对大数据进行深入的分析，而不是仅仅生成简单的报表。这些复杂的分析必须依赖于复杂的分析模型，很难用 SQL 来进行表达，统称为深度分析。人们不仅需要通过数据了解现在发生了什么，更需要利用数据对将要发生什么进行预测，以便在行动上做出一些主动的准备。比如通过预测客户的流失预先采取行动，对客户进行挽留。这里，典型的 OLAP 数据分析操作(对数据进行聚集、汇总、切片和旋转等)已经不够用，还需要路径分析、时间序列分析、图分析、What-if 分析以及由于硬件/软件限制而未曾尝试过的复杂统计分析模型等；

（3） **自动化、可视化分析需求的出现**。因为数据规模很大，要对大数据进行有效分析，分析过程需要按照完全自动化的方式进行。这就要求计算机能够理解数据在结构上的差异，明白数据所要表达的语义，然后“机械”地进行分析。对大数据分析来说，设计一个好的适于分析的数据表示模式是非常重要的。此外，大数据也使下一代可实时应答的交互式数据分析成为可能。将来，系统应该能够根据网站的内容自动构造查询，自动提供热门推荐，自动分析数据的价值并决定是否需要保存。目前，在保证交互式响应的同时如何进行 TB 级的复杂查询处理已成为一个重要的研究课题。

[38] Challenges and Opportunities with Big Data，A community white paper developed by leading researchers across the United States, www.cra.org/ccc/files/docs/init/bigdatawhitepaper.pdf

火龙果•整理
uml.org.cn

3.4.2 主要进展

针对上面提到的挑战，研究者提出了一些试验性的解决方法和途径，其中的许多方法具有一定的实际应用价值。例如，针对传统分析软件扩展性差以及 Hadoop 分析功能薄弱的特点，IBM 公司的研究人员致力于对 R 和 Hadoop 进行集成[29]。R 是开源的统计分析软件，通过 R 和 Hadoop 的深度集成，把计算推向数据并且并行处理，使 Hadoop 获得了强大的深度分析能力。另有研究者实现了 Weka（类似于 R 的开源的机器学习和数据挖掘工具软件）和 MapReduce 的集成[30]。标准版 Weka 工具只能在单机上运行，并且不能超越 1GB 内存的限制。经过算法的并行化，在 MapReduce 集群上，Weka 不仅突破了原有的可处理数据量的限制，轻松地对超过 100GB 的数据进行分析，同时利用并行计算提高了性能。经过改造的 Weka，赋予 MapReduce 技术深度分析的能力。另有开发者发起了 Apache Mahout 项目的研究，该项目是基于 Hadoop 平台的大规模数据集上的机器学习和数据挖掘开源程序库，为应用开发者提供了丰富的数据分析功能。

针对频繁模式挖掘、分类和聚类等传统的数据挖掘任务，研究人员也提出了相应的大数据解决方案。如，Iris Miliaraki 等人提出了一种可扩展的在 MapReduce 框架下进行频繁序列模式挖掘的算法[31]，Alina Ene 等人用 MapReduce 实现了大规模数据下的 K-center 和 k-median 聚类方法[32]，Kai-Wei Chang 等人提出了针对线性分类模型的大数据分类方法[33]。U Kang 等人使用“Belief Propagation 算法（简称 BP）”处理大规模图数据发掘异常模式[34]。

另有一些研究针对大规模图数据进行分析。Jayanta Mondal 等人[35]提出了一个基于内存的分布式数据管理系统来管理大规模动态变化的图以支持低延迟的查询处理方法，提出了一种混合的复制（replication）策略来检测结点读写的频率从而动态的决定哪些数据需要复制（replication）。Shengqi Yang 等人[36]对基于集群上的大规模图数据管理和局部图的访问特征（广度优先查询和随机游走等）进行研究，为了在图查询处理中减少机器间通讯，提出来分布式图数据环境，同时提出了两级别划分管理架构。Jiewen Huang 等人提出了一个多节点的可扩展 RDF 数据管理系统，比目前系统的效率高出 3 个数量级。

火龙果•整理
uml.org.cn

3.4.3 发展趋势

（1） **更加复杂、更大规模的分析和挖掘。**在大数据新型计算模式上实现更加复杂和更大规模的分析和挖掘是大数据未来发展的必然趋势。例如，需要进行更细粒度的仿真、时间序列分析、大规模图分析和大规模社会计算等等。另一方面，在大数据上进行复杂的分析和挖掘，需要灵活的开发、调试、管理等工具的支持。

（2） **大数据的实时分析和挖掘。**面对大数据，分析和挖掘的效率成为此类大数据应用的巨大挑战。尽管可以利用大规模集群并行计算，以 MapReduce 为代表的并行计算模型并不适合高性能的处理结构化数据的复杂查询分析。在数十 TB 以上的数据规模上，分析和发掘的实时性受到了严峻的挑战，是目前尚未彻底解决的问题。而查询和分析的实时处理能力，对于人们及时获得决策信息，做出有效反应是非常关键的前提。

（3） **大数据分析和挖掘的基准测试。**各种大数据分析和挖掘系统各有所长，其在不同类型分析挖掘下，会表现出非常不同的性能差异。目前迫切需要通过基准测试，了解各种大数据分析和挖掘系统的优缺点，以明确能够有效支持大数据实时分析和挖掘的关键技术，从而有针对的进行深入研究。

3.5 大数据可视化分析

3.5.1 问题与挑战

在大数据时代，数据的数量和复杂度的提高带来了对数据探索，分析，理解和呈现的巨大挑战。除了直接的统计或者数据挖掘的方式，可视化通过交互式视觉表现的方式来帮助人们探索和解释复杂的数据。一个典型的可视化流程是首先将数据通过软件程序系统转化为用户可以观察分析的图像。利用人类视觉系统高通量的特性，用户通过视觉系统，结合自己的背景知识，对可视化结果图像进行认知，从而理解和分析数据的内涵与特征。同时，用户还可以交互地改变可视化程序系统的设置，改变输出的可视化图像，获得对数据的不同侧面的理解。因此，可视化是一个交互与循环往复的过程。

火龙果•整理
uml.org.cn

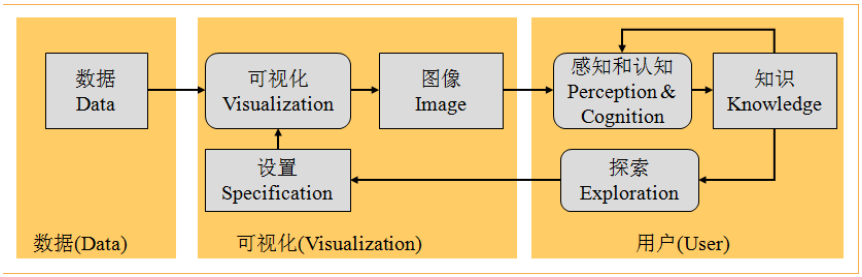

图3-1： 可视化流程（源自[37]）

可视化能够迅速和有效地简化与提炼数据流，帮助用户交互筛选大量的数据，可视化所提供的洞察力有助于使用者更快更好地从复杂数据中得到新的发现，这使得可视化成为数据科学中不可或缺的重要部分

人类对于数据对象通过作图的方式帮助理解分析古已有之。例如古人的地图和星图，早期物理学家对实验结果的绘图。现代意义上的可视化源自于计算机技术的发展，首先是对于科学数据的可视化，其后扩展到更广泛的信息可视化。二十一世纪开始后，随着反恐等需求，对于海量、复杂数据的分析进一步催生了可视分析，通过可视界面，结合人机交互和背景自动数据分析挖掘，对海量复杂数据开展分析。

3.5.2 主要进展

在可视化的发展中，首先面对大规模数据挑战的是在科学可视化方向。高通量仪器设备、模拟计算以及互联网应用等都在快速产生着庞大的数据，对 TB 乃至 PB 量级数据的分析和可视化成为现实的挑战。大规模数据的可视化和绘制主要是基于并行算法设计的技术，合理利用有限的计算资源，高效地处理和分析特定数据集的特性。很多情况下，大规模数据可视化的技术通常会结合多分辨率表示等方法，以获得足够的互动性能。在科学大规模数据的并行可视化工作中，主要涉及数据流线化(Data Streaming)，任务并行化(Task Parallelism)，管道并行化(Pipeline Parallelism)和数据并行化(Data Parallelism)四种基本技术[38]：。

数据流线化将大数据分为相互独立的子块后依次处理。在数据规模远大于计算资源时是主要的一类可视化手段。它能够处理任意大规模的数据，同时也可能提供更有效的缓存使用效率，并减少内存交换。但通常这类方法需要较长的处理时间，难以提供对数据的交互挖掘。离核渲染是数据流线化的一种重要形式。在

火龙果•整理
uml.org.cn

另外一些情况下，数据则是以流的形式实时逐步获得，必须要有能够适应数据涌现形式的可视化方法。

任务并行化是把多个独立的任务模块平行处理。这类方法要求将一个算法分解为多个独立的子任务，并需要相应的多重计算资源。其并行程度主要受限于算法的可分解粒度以及计算资源中节点的数目。管道并行化则是同时处理各自面向不同数据子块的多个独立的任务模块。以上任务并行化和管道并行化两类方法，如何达到负载的平衡是实现高效分析的关键难点。

数据并行化是将数据分块后进行平行处理，通常称为单程序多数据流(SPMD)模式。这类方法能达到高度的平行化，并且在计算节点增加的时候可以达到较好的可扩展性。对于非常大规模的并行可视化，节点之间的通讯往往是制约因素，提供合理的通讯模式是高效结果的关键，而提高数据的本地性也可以大大提高效率。以上这些技术往往在实践中相互结合，从而构建一个更高效的解决方法。

在信息可视化和可视分析方面，相对对大规模数据的处理出现的相应要晚得多。很多技术，例如多维数据可视化中的平行坐标，多尺度分析，散点图矩阵，层次数据可视化中的树图，图可视化中的多种布局算法，文本可视化的一些基本方法，并不是都有很好的可扩展性。在面对大数据挑战的可视化中，需要做出相应的调整。

传统对网络数据的可视化可以通过图的形式实现，这是将网络中的每个节点简化为图中的节点，网络中的联系可视化为图中的边，这样网络数据的可视化可以通过经典的节点－边的形式表现。这类可视化方法的难点主要在于图的排布算法。有效的图布局应该能够直观地揭示节点之间的联系，类似的、相互联系紧密的节点会聚集在一起。但是现在大规模的网络数据的节点可能高达数百万，其边可能高达数亿，这样的网络数据难以使用传统的图可视化方法可视化。

高维信息可以通过维度压缩、平行坐标等手段实现可视化。但是在数据达到一定规模以后，这样的方法并不能很好扩展。一些可能的方案包括提供一些子空间的选择，用户可以根据分析需要，在高维度空间选择适合问题解决的子空间，从而缩小数据规模。

图形硬件对于大规模数据可视化具有重要意义。最新的超级计算机大量地应用 GPU 作为计算单元。如何更好发掘最新的图形硬件潜力，提供更加灵活的大

火龙果•整理
uml.org.cn

数据可视化和绘制的解决方法是具有重大意义的课题。

3.5.3 发展趋势

面对大数据，结合国际学者的各种观点[39]，相应的大数据可视化与分析也面临着各种挑战：

（1） **原位分析（In Situ Analysis）**。传统的可视化方式是先将数据存储于磁盘、然后根据可视化的需要进行读取分析。这一种处理方式对于超过一定量级的数据来说并不适合。最初是对应与 exascale 规模的超级计算机计算获得的大量科学数据产生的挑战，I/O 几乎成为无法克服的困难。科学家提出了原位可视分析的概念，在数据仍在内存中时就会做尽可能多的分析。数据在进行了一定的可视化（同时也是数据规模的简化），能极大地减少 I/O 的开销，只有极少数的视觉投影后的次生数据需要转移到显示平台。这个方法可以实现数据使用与磁盘读取比例的最大化，从而最大限度地克服 I/O 的瓶颈限制。然而，它也带来了一系列设计与实现上的挑战，包括交互分析、算法、内存、I/O、工作流和线程的相关问题。原位分析要求可视化方案和计算紧密结合，这样很多传统的可视化方法需要进行修改或者筛选才可以用于这样的可视化模式。由于可视化的一部分处理在计算核点上进行，那样就会对可以进行的处理方案有所限制。

（2） **大数据可视化中的人机交互**。在可视化和可视分析中用户界面与交互设计扮演着越来越重要的角色。用户必须通过合理的交互方式，才可以有效地探索发现数据中的隐含信息，进行可视推理，通过意义构建，获得新的认知。然而尽管数据规模和机器的计算能力都在持续快速地增长，千百年来人的认知能力却是始终不变的。以人为中心的用户界面与交互设计面临的挑战是复杂和多层次的，并且在不同领域都有交叠。机器自动处理系统对于一些需要人类参与判断的分析过程往往表现不佳。其他的挑战则源于人的认知能力，现有技术不足以让人的认知能力发挥到极限。我们需要提供更好的人机交互界面和设计，方便使用者，特别是专家用户能够最大程度发挥其背景知识，在数据的分析中扮演更加积极的角色。从更广泛的一个意义上说，可视化可以建立一个可视的交互界面，提供人和数据的对话。

（3） **协同与众包可视分析**。在大数据时代，个人或者少数几个分析用户可

火龙果•整理
uml.org.cn

能无法面对数据规模和复杂度带来的挑战。大数据分析中往往设计多种不同来源甚至领域的数据。利用众人的智慧，通过众包等模式进行有效的复杂可视化成为一种必然的选择。在众包可视化工作中，如何设计合理高效的可视化平台，承载相应的复杂高难度的可视化系统工作；如何设计交互的中间模式，支持多用户的协调工作；如何反映多用户的差别，都是可以研究的课题。和协同的可视分析方式比较，协同可视化趋于少数的几个领域专家交互合作开展对数据的可视分析，众包可视化则更趋向不特定多数的使用者，规模也更大。如何开展有效的众包和协同可视化，是非常重要的研究课题。

（4） 可扩展性与多级层次问题。在大规模数据可视分析的可扩展性问题上，建立多级层次是主流的解决办法。这种方法可以通过建立不同大小的层面，提供用户在不用解析度下的数据浏览分析能力。但是当数据量增大时，层级的深度与复杂性也随之增大。在继承关系复杂且深度大的层次关系中巡游与搜索最优解是可扩展性分析的主要挑战。

（5） 不确定性分析和敏感性分析。不确定性的量化问题可以追溯到由实验测量产生数据的时代。如今，如何量化不确定性已经成为许多领域的重要问题。了解数据中不确定性的来源对于决策和风险分析十分重要。随着数据规模增大，直接处理整个数据集的能力也受到了极大的限制。许多数据分析任务中引入数据的不确定性。不确定性的量化及可视化对未来的大数据可视分析工具而言极端重要，我们必须发展可应对不完整数据的分析方法，许多现有算法必须重新设计，进而考虑数据的分布情况。一些新兴的可视化技术会提供一个不确定性的直观视图，来帮助用户了解风险，从而帮助用户选择正确的参数，减少产生误导性结果的可能。从这个方面来看，不确定性的量化与可视化将成为绝大多数可视分析任务的核心部分。另一方面，对于可视化而言，用户的交互或者新的参数的输入，都会导致不同可视化结果的出现。在大数据的情况下，向用户提供背景知识，告知预期的操作可能引发的可视化结果的变化程度，或者用户当前所在参数空间的周边状况，这一些都属于对可视分析结果的敏感性分析，对于高效的可视化交互是极端重要的。

（6） 可视化与自动数据计算挖掘的结合。可视化提供了用户对数据的直观分析，用户可以通过交互界面对数据进行分析了解。同时，我们要注意到很多的

火龙果•整理
uml.org.cn

数据分析是批量的。我们如何能够将一些比较确定的分析任务利用机器自动完成，同时引导用户来进行更具有挑战性的可视分析工作，是可视分析发展中的核心课题。

（7） **面向领域和大众的可视化工具库**。提供相应的工具库可以大大提高不同领域分析数据的能力。

大数据时代涌现并推动了很多可视化商业化的机会。Tabealu 的成功上市反映了市场对可视化工具的需求。类似 IBM Manyeyes 这样在线可视化工具的流行，则表明在一定程度上满足了广大普通用户对可视化方法的需求。国际的几个大公司也在开展相应的研究，企图把可视化引入其不同的数据分析和展示的产品中。各种可能相关的商品也将会不断出现，对可视化服务的商业需求将是未来的一个最大方向。

3.6 大数据隐私与安全

3.6.1 问题与挑战

隐私[40]是当事人不愿意被他人知道或他人不便知道的敏感信息，它与公共利益、群体利益无关，具有隐藏特性。安全[41]是指不受威胁，没有危险、危害、损失。信息安全[42]是指采取技术和管理的安全保护手段，保护软硬件与数据不因偶然的或恶意的原因而遭到破坏、更改、显露。

在大数据时代，传统的隐私数据内涵与外延有了巨大突破与延伸，隐私数据保护不力所造成的恐慌已不能由个人或团体承受，隐私数据保护技术面临更多的挑战。大数据时代下的隐私数据保护与安全体系除涉及技术、管理外，还涉及法律、人伦、生物、道德、商业利益、生活方式等；不只是团体或区域，还涉及到国家安全与国际秩序[43，44]。隐私数据泄露影响的波及面很可能会突破个人、团体或区域的限制，发展到全球性影响。

从本质上来说，大数据的安全与隐私问题就是我们要能够在大数据时代兼顾安全与自由，个性化服务与商业利益，国家安全与个人隐私的基础上，从数据中挖掘其潜在的巨大商业价值和学术价值，并使其研究成果真正的服务于社会。

在大数据时代，随着我们对大数据的进一步认识和研究，呈现出的安全隐私

火龙果•整理
uml.org.cn

挑战有以下几个方面[44]：

（1） **大数据时代的安全与传统安全相比，变得更加复杂**：一方面，大量的数据汇集，包括大量的企业运营数据、客户信息、个人的隐私和各种行为的细节记录。这些数据的集中存储增加了数据泄露风险，而这些数据不被滥用，也成为人身安全的一部分。另一方面，大数据对数据完整性、可用性和秘密性带来挑战，在防止数据丢失、被盗取和被破坏上存在一定的技术难度，传统的安全工具不再像以前那么有用。

（2） **使用数据过程中的安全问题**：用数据挖掘和数据分析获取商业价值[43]的时候，黑客也可以利用大数据分析向企业发起攻击。黑客可能会打限度地收集有用信息，如社交网络、邮件、微博、电子商务、电话和家庭住址等，使得数据安全异常严重。

（3） **对大数据分析较高的企业和团体，面临更多的安全挑战**：对于电子商务、金融、天气预报的分析预测、复杂网络计算和广域网感知等领域，恶意性攻击会造成更会严重的后果。

（4） **基于位置的隐私数据暴露严重**[46]：个体用户的移动设备的广泛使用，如手机，移动 GPS 设备等，以及一些网站获取用户位置信息等可以很容易得到用户的移动轨迹。而根据研究发现，用户的移动模式和用户身份识别之间有着强烈的对应关系，使得用户的隐私很容易暴露。同时，用户的位置信息保护比用户的身份信息保护更具有挑战性，因为我们在获取数据时要保证较高的精度。

（5） **缺乏相关的法律法规保证**：目前为止，还没有严格的法律法规来保证用户的数据隐私安全[47]。特别是一些涉及用户敏感数据的一些记录，而这些数据也容易被一些非法和不道德组织或个体使用，对用户和社会造成严重的影响和损失，例如，频繁发生的互联网公司数据库泄露事件，特别是 2013 年曝光的美国国家安全局“棱镜计划”监听项目。

（6） **大数据的共享问题**：共享问题的主要本质是数据的加密性和数据的有效性之间的矛盾。从社会应用角度考虑，我们会尽可能的提高数据的获取技术，以保证数据的有效性，而从保护用户隐私的角度考虑，我们有必要对数据进行相关操作以降低获取数据的敏感性，从而造成了两者之间的矛盾，两者之间如何进行最佳折中确实非常困难。

火龙果•整理
uml.org.cn

（7） **真实数据的动态性变化**：真实性的大数据随着时间呈现出动态变化性，使得我们对于大数据的分析计算提出了一些新的方法和技术[48,49]，因而在处理时将面对更为复杂的形式，加大了大数据安全隐私保护的困难。

（8） **多元数据的融合挑战**：大数据来自于生活，学术，商业等各个方面，而数据之间的彼此相关性，使得数据的安全隐私保护更为复杂，如何在多元数据融合的大趋势下保证用户的隐私不被泄露是一项重大挑战。

3.6.2 主要进展

数据的安全与隐私问题近年来一直是国内外学者关注的重大研究课题，并且针对不同的应用和数据类型都有相关的研究成果，总的来说，目前所拥有的方法有：

（1） **文件访问控制技术**：通过文件访问控制来限制呈现对数据的操作，在一定程度解决数据安全问题。

（2） **基础设备加密**：其本质是对大数据的存储设备进行安全防护，但不能解决大数据安全的本质问题。

（3） **匿名化保护技术**：匿名化技术适用于各类数据和众多应用，并且算法通用性高，能保证发布数据的真实性，实现简单。匿名化过程不可逆，如决策分类器的构建，聚类等应用，如 k-匿名模型，m-invariance 等。但匿名化技术对隐私保护效果并不明显，使得隐私泄露可能性很大，

（4） **加密保护技术**：加密保护技术能够保证数据的真实性，可逆性和无损性，对隐私保护程度很高，主要应用与分布式下的数据挖掘和操作，如 SMC 模型，分布式关联规则挖掘算法[50]，差分隐私等。但是该技术的计算开销很大，对大数据的支持不大适用。

（5） **基于数据失真的技术**：该技术可应用与关联规则的挖掘和隐藏等，如随机干扰，随机化，阻塞，凝聚等。数据失真技术的实现比较简单，但会造成数据的偏差，可能造成数据价值的丧失，

（6） **基于可逆的置换算法**：可逆的置换算法可以保证数据的真实性，并且效率比较高，常用于数据中心的大规模系统隐私保护，如位置变换，映射变化等。但该技术对于安全隐私保护力度仍然不够充分。

火龙果•整理
uml.org.cn

3.6.3 发展趋势

随着大数据的不断发展和研究，其巨大价值在被不断挖掘的过程中，数据的安全和隐私发展呈现出新的发展趋势和挑战：

（1） NoSQL 有待进一步完善：迎合了大数据的时代，适合非结构化数据的存储和分析，有灵活、可扩展性强、降低复杂性等特点，但是在安全保护上有待进一步提高。

（2） 针对于 APT 的攻击：在大数据时代，我们在利用数据来获取价值，APT 的攻击隐藏在数据内部，很难被我们发现，所以专门针对于 APT 攻击的研究是非常很总要的。

（3） 大数据的迅速发展和数据量的急剧增加及急速的动态变化，使得我们在对数据的操作时所面临的的安全问题更加严重。

（4） 数据的多元化与彼此的关联性进一步发展，深度挖掘技术，分析方法，算法模型的进一步优化和提高，使得对单一数据的安全隐私保护方法变得极其脆弱，需要针对多元数据融合提出新的安全隐私保护技术。

（5） 针对目前的大数据计算，主要采取的是分布式计算方法。而采用分布式计算的时候必然面临着数据传输，信息交互等过程，如何在这个过程中保护数据价值不泄露，信息不丢失，保护所有站点的安全与分布式系统的隐私是大数据发展面对的重大挑战。

（6） 目前，社交网络成为现代生活不可或缺的部分，一般来说，社交网络都会获取个体用户的位置信息（如 facebook，新浪微博等），以及此网络的迅速动态变化和实时交互等性质，使得我们对网络的安全加密与数据保护更为困难，而作为目前迅速发展起来的社交网络，我们需要进一步加强此方面的安全隐私保护。

（7） “三权分立”的模式应成为一种趋势，即数据的采集过程保护，存储管理保护，以及数据的分析使用过程的安全保护需要由不同的管理决策者来执行，这样可以在一定程度上保护大数据的安全隐私。

最后，大数据的保护需要学术界、商业界以及政府部门的共同参与，需要形成有效的安全机制和国家法律法规来约束和保护大数据的安全隐私，从而保证大数据时代的健全、安全发展。