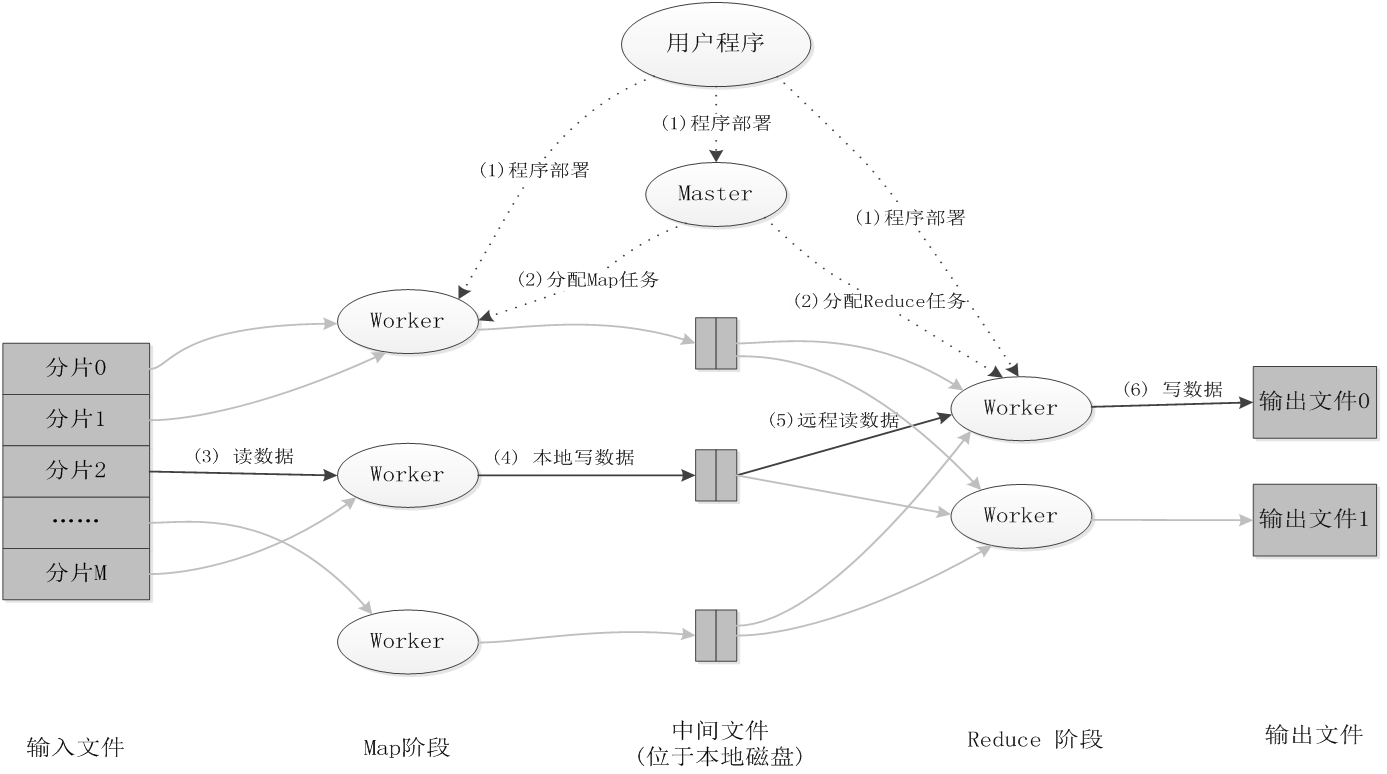
**第十二章**

**MapReduce工作流程（p18）**

（1）Resource Manager给提交的客户端MapReduce配置相应的计算资源（以Container形式）并启动多个工作线程Worker，一部分Worker将承担map任务（这部分Worker可称为Mapper），另一部分Worker将承担reduce任务（这部分Worker可称为Reducer）；

（2）对应每一个split，Application Master会生成一个map任务，然后分派一个Mapper去执行该map任务；Application Master也会生成一定数目的reduce任务分派给Reducer去执行。reduce任务个数取决于集群中可用的reduce任务槽（slot）的数目, 通常设置得比slot数目小一些，这样可以预留一些资源处理可能发生的错误；

（3）Mapper读取分派给它的输入数据split，并生成相应的键值表；

（4）Mapper执行计算处理任务，将中间结果输出保存在本地缓存；

（5）Application Master调度reducer读取mapper的中间输出文件，执行reduce任务；

（6）Reducer将最后结果写入输出文件保存到HDFS。

第十章

P3

**数据采集层**（系统日志、网络爬虫、无线传感器网络、物联网、以及各种数据源）

**数据清洗、抽取与建模**（将各种类型的结构化、非结构化、异构数据转化为标准存储格式数据，并定义数据属性及值域）

**数据存储架构**（集中式或分布式文件系统、关系型数据库或分布式数据库、行存储数据结构或列存储数据结构，键值对结构，哈希表（Hash Table）检索等）

P4

针对不同类型数据的计算模型

（如针对非结构化数据的MapReduce批处理模型、针对动态数据流的流计算（Stream Computing）模型、针对结构化数据的大规模并发处理（MPP）模型、基于物理大内存的高性能内存计算（In-memory Computing）模型）

针对应用需求的各类数据分析算法

（回归分析、聚合算法、关联规则算法、决策树算法、贝叶斯分析、机器学习算法等）

提供数据计算处理各种开发工具包和运行支持环境的计算平台

（Hadoop[5], Spark[6], Storm[7]）

**P7**

**大数据数据特点:**

**多种数据源**

（企业数据、商务数据、个人社交数据、政府统计数据、互联网数据、物联网数据、系统日志数据、基因测序数据、大气物理监测数据、地球卫星观测数据。。。。。。等等）

**异构型数据**

（文字、图片、音频、视频）

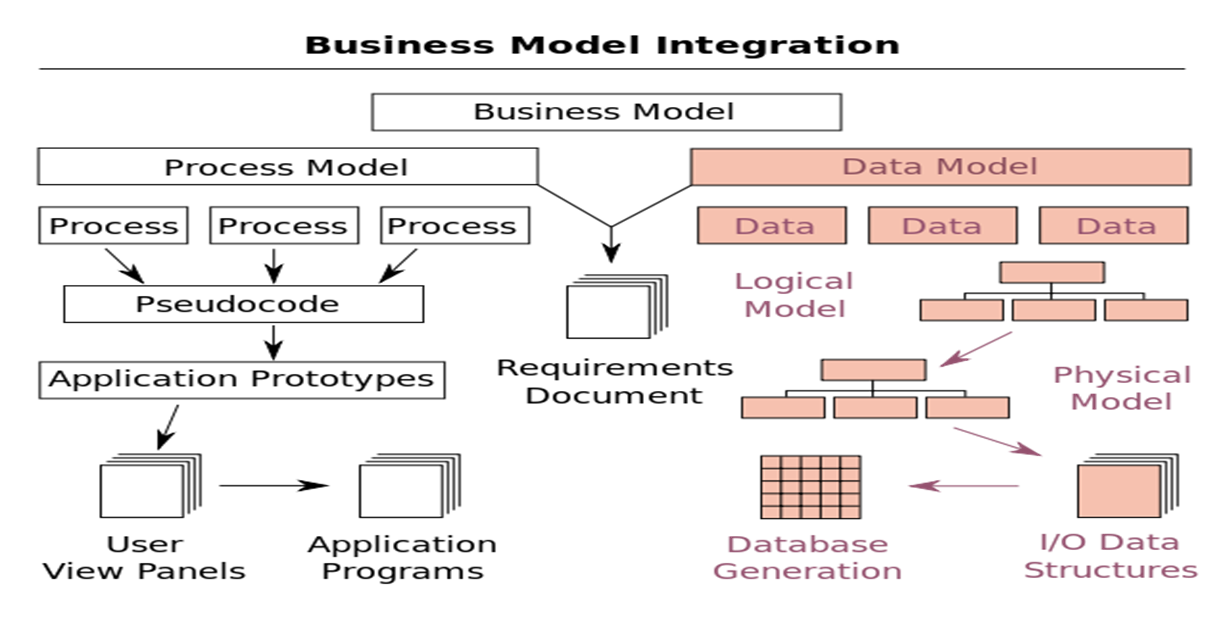
**非结构型数据**

（医学影像资料、银行凭证扫描件、碎片化的通信记录、截屏图等）

原始数据很难直接存入数据库

经常遇到的问题是原始数据的格式不能被数据处理平台识别和处理，很多情况下原始数据还存在记录缺失、值域缺损、数据质量参差不齐等问题。

**P11**



在进行数据集成和数据库实现时，要注意三个层次数据模型描述和定义的一致性。该图描述了商业数据的建模和集成过程，请注意右侧的数据建模流程（带色图标）与左侧的业务模型步骤的对应关系。

**P19**

**NoSQL数据库特点：**

1） **不需要预定义数据格式**：不需要预先定义严格的数据表结构，数据每条记录都可能有不同的属性和格式，当插入数据时，并不需要预先定义它们的格式；

2） **无共享架构**：NoSQL数据库往往将数据集划分后存储在各个本地服务器上，从本地磁盘读取数据的性能往往好于通过网络传输读取数据的性能，从而提高了系统读写速度；

3） **弹性可扩展**：可以在系统运行的时候动态增加或者删除结点而不需要停机维护，数据可以自动迁移；

4） **数据分区：**相对于将数据存放于同一个节点，NoSQL数据库则是将数据进行分区，将记录分散在多个节点上面。并且通常分区的同时还要做复制，这样既提高了并行性能，又能保证没有单点失效的问题；

5) **异步复制**：NoSQL的复制往往是基于日志的异步复制, 这样数据就可以尽快地写入一个节点，而不会被网络传输引起迟延, 缺点是并不总是能保证一致性。NoSQL数据库提供的是基于BASE原则[20]的最终一致性。

**P20**

**NoSQL数据库性能要求——CAP理论**

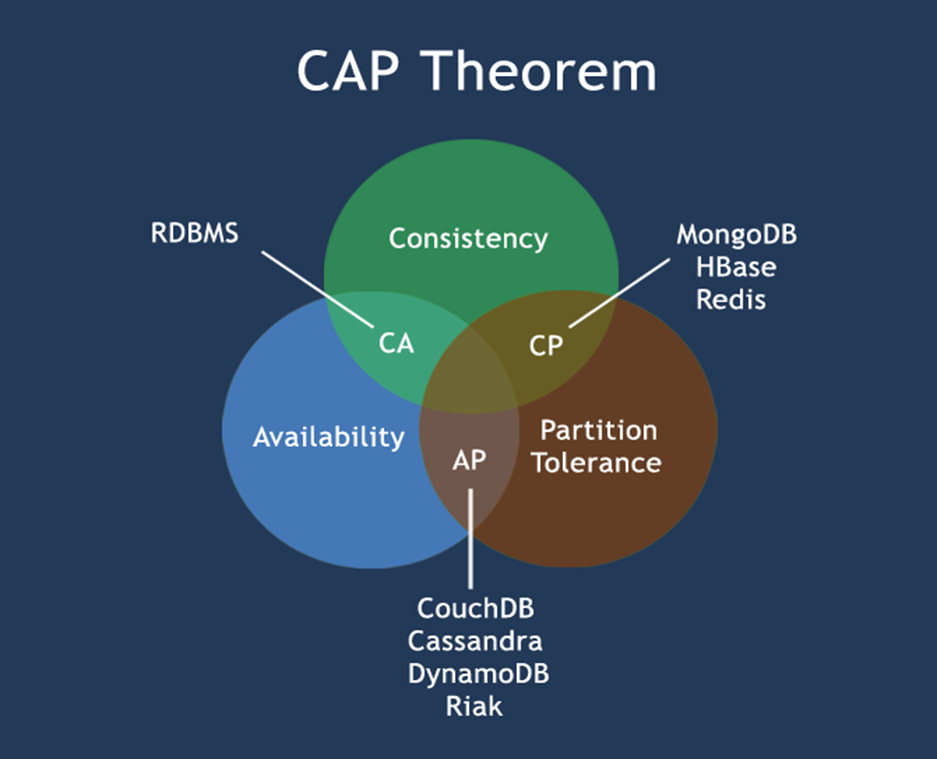
**C：一致性（Consistency）**即在分布式环境中多个存储节点的数据在同一时间具有相同的数据值，所有数据备份的更新是同步的；

**A：可用性（Availability）**能快速读取数据，在合理的时间内返回操作结果，并保证每个请求不管成功或者失败都有响应；

**P：分区容忍性（Partition-tolerance）**指数据分区的容错性，即系统中的某个分区无法与其它节点通信时不影响系统其余部分的正常运行，或者是系统部分数据的错误或丢失不影响系统的整体运行。

**P21**

**CAP理论**



关系型数据库（RDBMS）更强调数据一致性和可用性、而基于健值紧密关联的数据表也不适合拆分，因此关系型数据库是典型的基于CA法则（Consistency & Availability）的设计。

超大规模数据需要对数据进行切割划分，以支持并行计算处理，因此NoSQL数据库首先会选择分区容忍性（Partition-tolerance）法则。

在此基础上，有一类NoSQL数据库倾向于支持数据一致性（事实上只能保证最终一致性），称之为CP型数据库，代表有MongoDB [23]，HBase [24]，Redis [25]等。CP型数据库比较重视数据同步，但在节点间网络连接发生问题时，等待时延较长；

另一类则选择强调数据可用性、放弃了运算过程中的数据一致性，被称为AP型数据库，如CouchDB[26]，Cassandra[27]，DynamoDB[28]等。AP型数据库强调计算处理效率，但需要面对数据不同步的问题。

**P22**

**NoSQL数据库事物要求——BASE法则**

**Basically Available 基本可用性**: 意指分布式系统的一部分发生问题变得不可用时，其他部分仍然可以正常使用，也就是允许出现分区失败的情形；

**Soft-state软状态:** 意指数据状态在执行中可以有一段时间不同步，具有一定的滞后性。与此对应的是Hard-state 硬状态，指整个过程中都必须数据同步，保证数据一致性；

**Eventual consistency最终一致性:** 指在事务执行过程中，容许后续的读取操作暂时读不到最新数据，但在事务结束时，所有数据都要更新，达到最终一致性。

**P38**

**MapReduce**

**计算流程**

Split（数据划分）

Map（映射）

Collect&Sort（聚合排序，也称Shuffle）

Reduce（简化）

Store（数据存储）

特点

分治策略

缺点

硬盘数据读取频繁，处理时效性较差

具体过程：程序从分布式文件系统读入大数据集并切分为Split (分片)、 Map函数处理Split并输出中间结果、Shuffle把中间结果分区排序整理后发送给Reduce函数、Reduce完成具体计算任务并将最终结果写入文件系统。

分治策略：将一个大数据集分割为多个小尺度子集（split），然后让计算程序靠近每个子集，同时并行完成计算处理。

**P40**

流计算模型

是一种处理实时动态数据的计算模型

以流计算处理平台Storm [7]为例

其数据采集系统将实时数据（消息队列MetaQ或Socket导入数据、前端业务数据、Log监控数据等）通过平台的数据接入层导入Storm平台

Storm实时处理系统则承担数据实时计算分析任务；计算结果则导入数据落地层（Hadoop的HDFS存储系统、MySQL数据库或Lustre文件系统），提供对用户的实时查询服务。

另外，Storm还有一个元数据管理器统一协调前端业务数据写入，定义实时数据类型及描述格式，并指导数据落地层如何处理结果数据。

应当注意，Storm实时处理系统并不是将所有的实时数据都导入落地层，大部分无用的实时数据在完成计算后即丢弃。

**P41**

**Hadoop**

是目前最为广泛应用的开源大数据计算平台，它提供了一套完整的开放式计算架构、技术标准和开发工具，可以运行在通用标准的廉价服务器集群上，在学术界和工业界都拥有最多的用户。

**Cloudera**

是一个基于Hadoop平台的大数据商业计算产品。Cloudera目前提供一个免费的 CDH (Cloudera Distribution Hadoop) 版本，但不包括其他Cloudera开发的工具和功能库。Cloudera商业版包括Cloudera Manager（提供管理、监控、诊断、集成功能）、海量日志采集系统Flume，支持定制数据发送方（console（控制台）、RPC（Thrift-RPC）、text（纯文本）、tail（UNIX tail）、syslog（系统日志，支持TCP和UDP两种模式），exec（命令执行）等，并提供数据简单处理。Cloudera还提供对存储在Hadoop/HDFS/HBase平台数据的SQL查询工具Impala[57]，以及web管理器Hue。

**Spark**

是由美国加州大学伯克利分校AMP Lab提供的一个基于内存计算模型的开源大数据并行处理框架，它可以搭建在Hadoop平台上，利用HDFS文件系统存储数据，但在文件系统之上构建了一个弹性分布式数据架构 (RDD, Resilient Distributed Dataset)，用于支撑高效率的分布式内存计算。

**Storm**

是一个分布式实时处理系统，最早由BackType公司开发，后属于Twitter所有，现在成为Apache开源项目。与MapReduce模型将大规模数据首先导入数据库、然后再进行计算处理的方式不同，Storm采用连续计算（continuous computation）模型，对输入数据流(stream)做在线连续处理，计算结果也以数据流的形式动态输出给用户，这就避免了MapReduce离线批处理模式带来的严重时延问题。

**MapReduce计算引擎**

提供大数据的划分、节点分配、作业调度及计算结果融汇等功能，直接支持上层应用的开发。

**Google的交互式计算引擎**

采用Dremel [48], PowerDrill [54]技术，提供了对大规模数据集的快速计算分析

**图并行计算引擎**

提供对网路图数据（社交网络、电信网络、脑功能连接网络这一类数据常常可用权重有向图来表征）的高效计算处理（Google搜索引擎处理的数据量中有20%是用图计算引擎来处理），这方面的技术包括Google的Pregel [46]，开源技术的Hama [47]，GraphLab [58]等。

**S4（Simple Scalable Streaming System）**

是Yahoo！提供的一个流计算引擎，最初目标是提高cost-per-click广告点击率问题，通过实时数据计算预测用户对广告的可能的点击行为