大数据的存储与处理

主编：王道平

资料整理
火龙果软件
荐

本章教学要点

本章主要介绍大数据的存储方式、数据仓库和大数据的处理框架等内容，需要掌握**分布式文件系统、数据仓库、Hadoop和Spark处理框架**等重要知识点，在此基础上，还要熟悉了解NoSQL数据库和云存储等知识点，对大数据的存储和处理有进一步的理解。

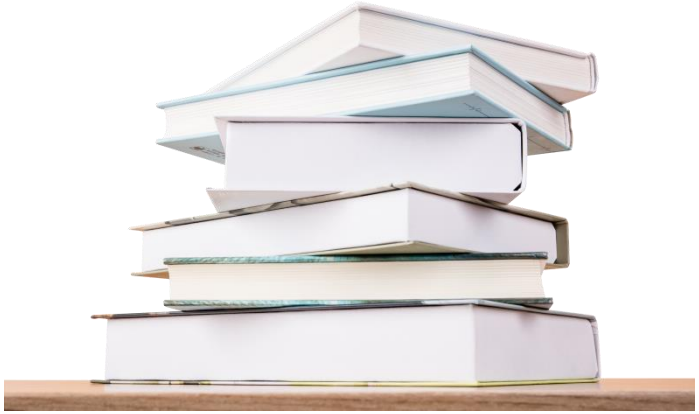
资料整理
火龙果软件
荐

目录

5.1 大数据的存储方式

5.2 数据仓库

5.3 大数据的处理框架

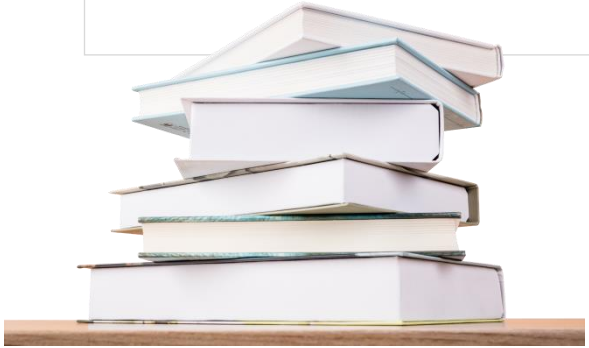


资料整理
火龙果软件
荐

01

大数据的存储方式

PART ONE



5.1 大数据的存储方式

Web技术的快速发展和移动互联网终端的广泛使用，使得数据性质发生根本性变化，这种特性使其与传统的企业数据区分开，**不再集中化、高度结构化和易于管理。**

数据规模庞大、节点分布、数据类型多样化等是大数据存储系统所面对的新问题，目前，大数据的存储方式主要有以下几种。

分布式文件系统

NoSQL数据库

云存储

资料整理
火龙果软件
荐

5.1.1 分布式文件系统

1. 文件系统的概念

文件系统（File System，FS）是计算机操作系统用于明确存储设备或分区上的文件的方法和数据结构，是操作系统在计算机存储设备上组织文件的方法，负责管理和存储文件信息，是计算机操作系统最重要的组成部分之一。

文件系统**由3个部分组成**：文件系统的接口，对对象操作和管理的软件集合，对象及属性。文件系统确定了文件命名的规则，例如文件名的长度、可以使用的合法字符、文件后缀的长度以及通过目录结构找到文件的指定路径的格式等等。

资料
火龙果软件

5.1.1 分布式文件系统

常见的文件系统见下表。

名称	适用场景
New Technology File System（NTFS）	微软公司的Windows系列操作系统
Extended File Allocation Table File System（ExtFAT）	用于支持U盘、闪存盘的存储
Hierarchical File System（HFS）	苹果公司的Mac OS系列操作系统
HFS的改进版本（HFS+）	
Apple File System（APFS）	
Second Extended File System（Ext2）	Linux系列操作系统
Third Extended File System（Ext3）	

资料整理
火龙果软件
荐

5.1.1 分布式文件系统

文件系统诞生之初是为**本地存储的数据服务的**，即本地文件系统模式，文件系统所管理的文件都存储在本地节点上，**本地节点存储能力有限**，无法进行大幅度扩展，同时它也只能管理本地节点所存储的文件，单机模式下所存储的数据没有备份，**存在单点故障风险**，而且**并发能力比较差**，难以满足大数据存储的需求，分布式文件系统应运而生。

随着互联网的快速发展，由各种计算机组成的网络在社会中无处不在，在互联网中，不存在集中式的控制中心，而是存在大量分离且互联的节点，**分布式的理念就是随着互联网的发展而诞生的，并且在计算机网络存储领域得到了快速发展和应用。**

资料整理
火龙果软件
荐

5.1.1 分布式文件系统

2. 分布式文件系统的概念

分布式文件系统（Distributed File System，DFS）是**基于分布式理念的文件系统**，将固定于某个地点的某个本地文件系统扩展到多个地点和多类型的文件系统。

特点：相对于本地文件系统而言，分布式文件系统所管理的存储资源并不是全部直接连接在本地节点上，而是通过计算机网络管理连接节点的存储资源，这些节点可能存在于不同的区域中，在空间上存在一定距离，众多的节点组成一个文件系统网络，通过网络进行节点间的通信和数据传输。

在使用分布式文件系统时，人们无需关心数据存储在哪个节点上或者是从哪个节点获取的，就像使用本地文件系统一样管理和存储文件系统中的数据。

资料整理
火龙果软件
荐

5.1.1 分布式文件系统

2. 分布式文件系统的概念

优势：传统的数据存储方式是集中式的存储，即使用能力强大的服务器集中统一处理计算机网络中大量的数据存储任务，但这种大型服务器设备的建造和维护成本极高，而分布式的数据存储方式可以将需要海量存储能力才能处理的问题拆分成许多小块，然后将这些小块分配给同一套系统中不同的存储节点进行处理，很好的克服了集中式存储的弊端，更加满足大数据时代的存储需求。

分布式文件系统将服务范围扩展到了整个网络，**改变了数据的存储和管理方式，拥有了本地文件系统所无法具备的数据备份、数据安全等优点**，为分布式存储的实现与发展奠定了坚实的基础。

资料整理
火龙果软件
荐

5.1.1 分布式文件系统

3. 典型的分布式文件系统

基于多种分布式文件系统的研究成果，人们对体系结构的认识不断深入，分布式文件系统在体系结构、系统规模、性能、可扩展性和可用性等方面经历了很大的变化，下面介绍几种典型的分布式文件系统。

- 网络文件系统
- 通用并行文件系统
- 谷歌文件系统
- Hadoop分布式文件系统

资料整理
火龙果软件
荐

5.1.1 分布式文件系统

（1）网络文件系统

网络文件系统（Network File System，NFS）是FreeBSD（一种类UNIX操作系统）支持的一种文件系统，**允许网络中的计算机之间通过TCP/IP协议网络共享资源。**

在NFS的应用中，本地NFS的客户端应用可以方便地读写位于远端NFS服务器上的文件，就像访问本地文件一样。NFS利用UNIX系统中的虚拟文件系统（Virtual File System，VFS）机制，通过规范的文件访问协议和远程过程调用客户机对文件系统的请求，并转发到服务器端进行处理；服务器端在VFS机制之上，通过本地文件系统完成文件的处理，实现了全局的分布式文件系统。

资料整理
火龙果软件
荐

5.1.1 分布式文件系统

（2）通用并行文件系统

通用并行文件系统（General Parallel File System，GPFS）是IBM公司的第一个共享文件系统。GPFS的磁盘数据结构可以**支持大容量的文件系统和大文件**，通过采用分片存储、较大的文件系统块（Block）和数据预读等方法获得较高的数据吞吐率；采用扩展哈希（Extensible Hashing）技术来支持含有大量文件和子目录的大目录，提高文件的查找和检索效率。

GPFS采用不同粒度的分布式锁来解决系统中并发访问和数据同步的问题。GPFS每个节点都有各自独立的日志，且单个节点失效时，系统中的其他节点可以代替失效节点检查文件系统日志，进行元数据恢复操作。GPFS**支持在线动态增减存储设备**，能够在线重新平衡系统中的数据，**能够有效支持高端应用连续作业的需要。**

资料整理
火龙果软件
荐

5.1.1 分布式文件系统

（3）谷歌文件系统

谷歌文件系统（Google File System，GFS）是谷歌**为了存储海量搜索数据而开发实现的分布式文件系统**，它由一个Master节点和大量的ChunkServer节点构成。

GFS中心是一个Master节点，根据文件索引找寻数据块，系统保证每个Master节点都有相应的复制品，以便于在其出现问题时进行切换。**GFS把文件分成64MB的数据块，减小了元数据的大小**，使Master节点能够方便地将元数据放置在内存中以提高访问效率。**在Chunk层，GFS将节点失效视为常态，因此将数据块复制到集群中不同的ChunkServer节点上，默认每个数据块保存3个副本。**

资料整理
火龙果软件
荐

5.1.1 分布式文件系统

（4）Hadoop分布式文件系统

Hadoop分布式文件系统（Hadoop Distributed File System，HDFS）是Hadoop框架中支持大数据存储的分布式文件系统，**是在GFS模型的基础上开发实现的。**

HDFS优化了大文件的流式读取方式，将一个大型文件分割成多个固定大小的数据块（通常为128MB），分散存储到集群的大量节点上，能够支持千万级的文件处理。HDFS中认为硬件故障是常态，将数据块在集群的不同服务器节点间进行复制，默认情况下会有3个相同的数据块备份，实现了高容错性。基于此，**HDFS可以选择部署在廉价的服务器上，降低了系统硬件成本。**同时因为开源Hadoop的缘故，HDFS得到了广泛发展和应用。

资料整理
火龙果软件
荐

5.1.2 NoSQL数据库

1.关系型数据库

逻辑上采用关系模型来组织数据的数据库，称之为关系型数据库（Relational Database，RDB）。关系型数据库中存在大量的关系数据表，使用结构化查询语言（SQL）作为数据库操作语言，也称为SQL数据库。

关系型数据库主要以Oracle、MySQL、Microsoft SQL Server为代表。

资料整理
火龙果软件
荐

5.1.2 NoSQL数据库

2.关系型数据库的优势和局限

关系型数据库作为当前主流的数据存储方式，其独特的优势包括：

（1）**良好的交互性**，任何掌握了SQL语言使用方法的用户均可以操作数据库，无需具有其他计算机语言基础；

（2）**标准化**，使用户能够跨平台、跨系统运行数据库系统，并对第三方附件和工具提供支持；

（3）**支持纵向扩展**，能够解决从快写为主导的传输到扫描密集型深入分析等问题；

（4）**采用正交形式呈现数据和存储数据**，支持JSON（JavaScript Object Notation）和其他结构化对象格式。

移动互联网时代的商业服务要求数据库能够实时地处理每秒千万次甚至亿次的读写请求，同时还要具有较高的数据可用性，**关系型数据库很难满足上述需求或者需要付出昂贵的代价来满足**，业界迫切需要一种支持横向扩展、成本低廉、较为灵活且高度可用的数据库，NoSQL数据库的诞生满足了大数据的存储需要。

资料整理
火龙果软件
荐

5.1.2 NoSQL数据库

3. NoSQL数据库的概念

NoSQL是“**Not Only SQL**”的英文简写，NoSQL数据库是不同于传统的关系型数据库的统称，**即非关系型（non-relational）的数据库的统称**，它**并不专指某一个产品或一种技术，而是代表一类产品及一系列的不同类型的数据存储与处理的技术。**

NoSQL数据库设计的**核心特征是面向于特定的问题**，采用不同存储方式的NoSQL数据库适用于不同的应用情景，其主要的存储方式将在本节后续进行详细介绍。

NoSQL数据库的诞生并不是为了取代关系型数据库，它们两者之间是一种互补的关系，会从对方身上学习优秀的特性，以应对需求日益复杂且日益严苛的应用程序。

资料整理
火龙果软件
荐

5.1.2 NoSQL数据库

3. NoSQL数据库的概念

与传统的关系型数据库相比，NoSQL数据库具有以下特点：

（1）**非关系型数据库，灵活的数据模型。**创建数据库之前，不需要事先定义数据模式、表的结构，例如在键值式存储与文档式存储中，允许应用在一个数据单元中存入任何结构的数据。

（2）**弹性的横向扩展性**，可以将多个服务器从逻辑上视为一个实体，运行在集群服务器上，随着负载的变化可将数据库调整分布到多个不同的主机上，性价比高，可支持的数据量大。同时，可以在系统运行的时候，动态增加或者删除节点，不需要停机维护，数据即可自动迁移。

（3）**与关系型数据库具有的ACID特性不同，NoSQL数据库保证的是BASE特性**：基本可用（Basically Available，BA），NoSQL数据库允许分布式系统中某些部分出现故障，系统的其余部分仍然可用，允许继续部分访问；柔性状态（Soft state，S），在NoSQL数据库的数据处理过程中，允许存在数据状态暂时不一致的情况；最终一致（Eventually Consistent，E），将柔性状态产生的短暂数据不一致情况，经过纠错处理最后转化为数据一致。

资料整理
火龙果软件
荐

5.1.2 NoSQL数据库

4. NoSQL数据库的存储方式

在NoSQL数据库中，最常用的存储方式有**键值式存储、文档式存储、列式存储、图形式存储**等，与关系型数据库中仅有一种存储方式不同，在一个NoSQL数据库可以存在多种存储方式。

键值式存储	文档式存储	列式存储	图形式存储

资料整理
火龙果软件
荐

5.1.2 NoSQL数据库

（1）键值式存储

顾名思义，键值式存储就是在数据库中**根据键值对数据**进行组织、索引和存储。键（Key）是数据的标识符，值（Value）是存放任意数据的容器，在同一个命名空间内，键的名字必须唯一，不能重复。采用键值式存储的数据库可称为键值数据库，键值式存储是NoSQL数据库中最简单、最常用的存储方式。常见的键值数据库有Redis、Voldemort和Oracle BDB等。

redis

资料整理
火龙果软件
荐

5.1.2 NoSQL数据库

（1）键值式存储

键值数据库的**优势**在于其**数据处理的速度非常快，具有极高的并发读写性能。缺点**在于**只能通过键来查询数据**，某些键值数据库不支持查询位于某个范围内的值，不支持类似于SQL式的标准查询语言。

不管值的存储格式如何，键值式存储**可根据数据的保存方式可以分为临时型方式、永久型方式和混合型方式3种。**

（1）临时型方式将所有的数据都保存在内存中，读写速度块，程序停止，数据就不存在了。

（2）永久型方式将数据保存在磁盘中，通过对磁盘进行读写来操作数据，性能受到一定影响。

（3）混合型方式融合了临时型方式与永久型方式，首先将数据保存在内存中，在满足特定条件的时候将数据写入到磁盘中，既确保了内存中数据的处理速度，又可以通过写入硬盘来保证数据的不丢失。

资料整理
火龙果软件
荐

5.1.2 NoSQL数据库

（2）文档式存储

采用文档式存储的NoSQL数据库称为文档数据库，文档数据库**以文档的形式来存储数据**，在存储数据前，并不需要像关系型数据库那样预先设置表的结构（Schema），就可以使用类似于关系表的数据结构。

文档数据库**主要面向的是实现海量数据存储的问题**，同时具有优异的查询性能，**并不要求具备高并发读写操作性能。**MongoDB、CouchDB是目前广泛流行的文档数据库。

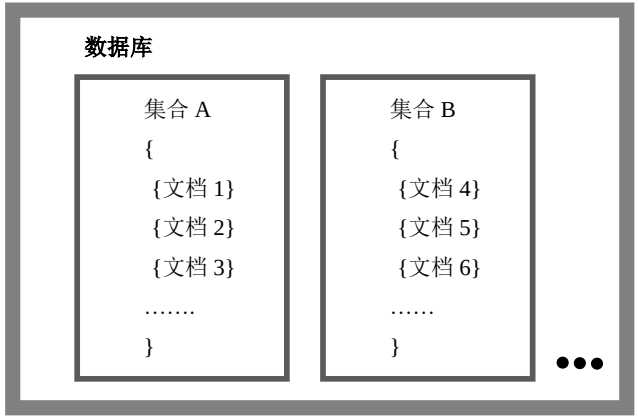
资料整理
火龙果软件
荐

5.1.2 NoSQL数据库

（2）文档式存储

文档、集合、数据库构成了文档数据库的三层逻辑结构，如图所示。

文档相当于关系数据库中的一条记录，但是文档并不需要提前设置数据模式即可存取；多个文档组成一个集合，集合相当于关系数据库的表；将多个集合在逻辑上组织在一起就是数据库。数据库中的每个记录都以文档形式存在，具备自我描述的功能，并且独立于任何其他文档。



资料整理
火龙果软件
荐

5.1.2 NoSQL数据库

（2）文档式存储

与关系型数据库相比，文档数据库具有很强的灵活性。

例如，在关系型数据库中，同一张关系表中所有的数据记录都必须具有完全相同的属性。而**在文档数据库的同一个集合中，文档的格式可以不完全相同**，如图所示，文档1和文档2可以属于文档数据库的同一个集合。

文档 1	文档 2
1. {	1. {
2. " keys":"word"	2. " keys":"word"
3. " type":"txt"	3. " type":"txt"
4. }	4. " value":[75,99,87.2]
	5. }

资料整理
火龙果软件
荐

5.1.2 NoSQL数据库

（3）列式存储

采用**列式存储的NoSQL数据库**，可以称为列族数据库，也称为宽列数据库，它可以实现按列存储数据，**便于存储结构化和半结构化数据以及进行数据压缩，**对于数据列的操作具有非常大的读写优势。目前广泛流行的列族数据库主要有Cassandra、HBase（Hadoop Database）等。

在列式数据库的逻辑结构中，**从顶层至底层依次是“列族（Column Family）”、“超列（Super Column）”和“列（Column）”**。列由键名、键值和时间戳组成，忽略时间戳，可当作一个键值对；超列相当于是字典（dict），它是一个包含其他列的特殊列，超列不能包含超列；一个列族中的所有数据都将保存在相同的文件中，它可以容纳多个超列或列。

资料整理
火龙果软件
荐

5.1.2 NoSQL数据库

（3）列式存储

样例数据

用户名	姓名	年龄（岁）	身高（cm）	爱好
用户 1	张三	22	170	羽毛球
用户 2	李四	45	165	（空）

关系型数据库按行存储

用户 1	张三	22	170	羽毛球
用户 2	李四	45	165	（空）

列族数据库按列存储

#	用户名
1	用户 1
2	用户 2

#	姓名
1	李四
2	张三

#	年龄
1	22
2	45

#	身高
1	用户 1
2	用户 2

#	爱好
1	羽毛球

资料整理
火龙果软件
荐

5.1.2 NoSQL数据库

（4）图形式存储

采用图形式存储的NoSQL数据库，称为图形数据库。它**运用图论的思想，将数据以图模型的方式进行组织存储**，在图模型中节点、边和特性是进行数据建模的3个重要组件。图模型中的节点代表实体，实体之间的关系则用节点之间的有向边来表示，节点和边的属性代表了实体和关系的特性。

在图形数据库中，对于数据的查询就是对于图形的遍历。因此，图形数据库有利于研究实体之间的关系，这是其他NoSQL数据库无法比拟的优势。常见的图形数据库包括Neo4J、InfoGrid等。

资料整理
火龙果软件
荐

5.1.2 NoSQL数据库

（4）图形式存储

图模型是一个被标记和标向的属性多重图。被标记的图每条边都有一个标签，它被用来作为那条边的类型。有向图允许边有一个固定的方向，从源点到目标节点，或从目标节点到源点。

属性图允许每个节点和边有一组可变的属性列表，其中的属性是关联某个名字的值，简化了图形结构。多重图允许两个节点之间存在多条边，这意味着两个节点可以由不同边连接多次，即使两条边有相同的尾、头和标记。

资料整理
火龙果软件
荐

5.1.3 云存储

1. 云存储的概念及组成

面对大数据的海量异构数据，传统存储技术面临建设成本高、运维复杂和可扩展性有限等问题，于是成本低廉、可扩展性高的云存储技术应运而生，并且得到了广泛的关注和应用。

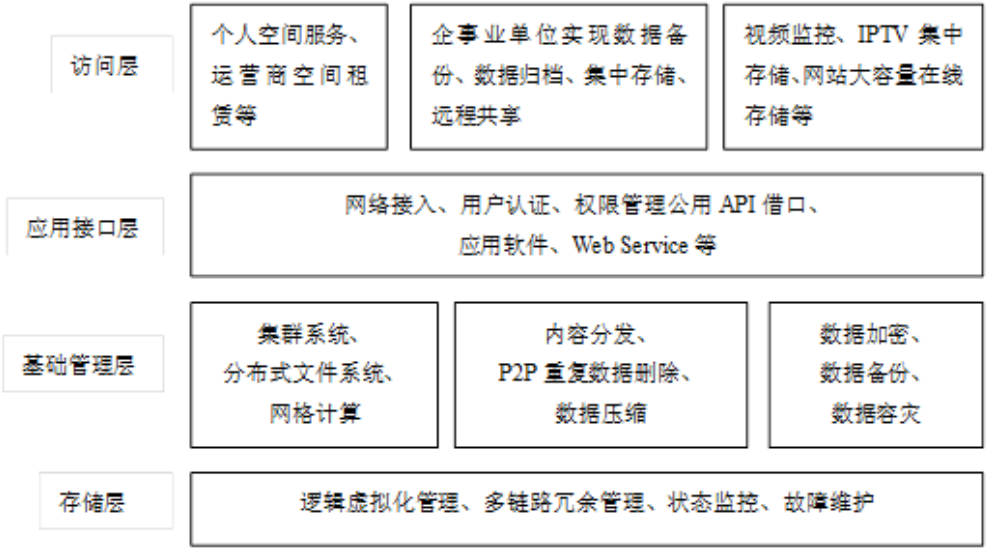
云存储（Cloud Storage）是在云计算概念上延伸和发展出来的一个新概念，属于新兴的网络存储技术，是指通过集群应用、网络技术或分布式文件系统等，**将网络中不同类型的存储设备通过应用软件集合起来协同工作，共同对外提供数据存储和业务访问功能的系统。**

资料整理
火龙果软件
荐

5.1.3 云存储

1. 云存储的概念及组成

云存储主要包含**存储层、基础管理层、应用接口层、访问层**，以存储设备为核心，通过应用软件来对外提供数据存储和业务访问服务。云存储的组成架构，如图所示。



资料整理
火龙果软件
荐

5.1.3 云存储

2. 云存储的应用及优劣势

云存储的应用主要包括3类：**基于公有云的存储、基于私有云的存储和基于混合云的存储。**

基于公有云的存储是第三方服务商为用户提供的网络存储服务，所有用户共享服务商的存储资源，但每个用户都是独立的，比如Dropbox、百度云盘。与前者不同，基于私有云的存储是建立在企业专属设施之上的，不与其他用户共享存储资源，最大限度地保证了数据的安全性。当前，越来越多的企业选择混合云的方式，既保证了数据的安全性，又满足了动态变化的存储需求。

资料整理
火龙果软件
荐

5.1.3 云存储

2. 云存储的应用及优劣势

云存储作为优秀的存储技术，其**主要优势包括**：

（1）设备层面：云存储的存储设备数量庞大，分布区域各异，多个设备之间协同合作，许多设备可以同时为一个人提供同一种服务，并且云存储都是平台服务，云存储的供应商会根据用户需求开发出多种平台，如IPTV应用平台、视频监控应用平台、数据备份应用平台等。只要有标准的公用应用接口，任何一个被授权的用户都可以通过一个简单的网址登录云存储系统，享受云存储服务。

（2）功能层面：云存储的容量分配不受物理硬盘的控制，可以按照客户的需求及时扩容，设备故障和设备升级都不会影响用户的正常访问。云存储技术针对数据重要性采取不同的复制策略，并且复制的文件存储在不同的服务器上，因此硬件损坏时，不管是硬盘还是服务器，服务始终不会终止。因为采用索引的架构，系统会自动将读写指令引导到其他存储节点，读写效能完全不受影响，管理人员只要更换硬件即可，数据也不会丢失，换上新的硬盘服务器后，系统会自动将文件复制回来，永远保持多备份的文件，从而避免数据丢失。在扩容时，只要安装好存储节点，接上网络，新增加的容量便会自动合并到存储中，并且数据会自动迁移到新存储的节点，不需要做多余的设定，大大缩减了维护人员的工作量。

资料整理
火龙果软件

5.1.3 云存储

2. 云存储的应用及优劣势

云存储作为优秀的存储技术，其**主要优势包括**：

（3）成本层面：传统存储模式下，一旦完成资金的一次性投入，系统无法在后续使用中动态调整。随着设备的更新换代，落后的硬件平台难以处置；随着业务需求的不断变化，软件需要不断地更新升级甚至重构来与之相适应，导致维护成本高昂，很容易发展到不可控的程度。

使用云存储服务可以免去企业在设备购买和技术人员聘用上的庞大开支，维护工作及系统的更新升级都由云存储服务提供商完成，而且公有云的租用费用和私有云的建设费用会随着云存储供应商竞争的日趋激烈而不断降低。

资料整理
火龙果软件
荐

5.1.3 云存储

2. 云存储的应用及优劣势

云存储的劣势主要体现在安全和访问速度两个方面：

（1）安全问题：云存储的好处在于任何一个被授权的用户都可以访问云存储系统，但这种便利性也是云存储的致命伤，因为每个用户的设备都可能成为被攻击的对象，造成数据泄露。同时，云存储借助大型存储设备将相同的数据分别存储在不同的地域，形成数据备份，帮助用户解决数据容灾问题。但从另一方面来说，用户只知道自身的数据存储在云存储中，并不知道数据的具体存储位置或是否在没有授予权限的情况下被他人访问。知识产权得不到相应的保护，数据所有权也得不到相应的保障。

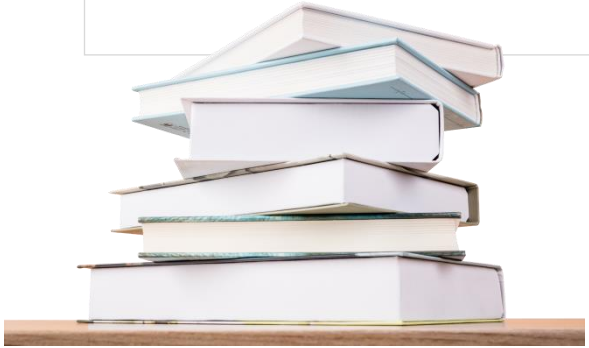
（2）访问速度问题：访问速度慢是当前云存储短时间内无法突破的一个瓶颈，也是用户最希望解决的问题。目前云存储还不能处理交易十分频繁的文件，要求网络连接速度快的数据库不是云存储的存储对象，Tier1、Tier2或以块（Block）存储为基础的数据存储也超出了云存储的存储能力，目前云存储的访问速度主要是受到云存储运营商拥有设备的性能和网络带宽的限制。

资料整理
火龙果软件
荐

02

数据仓库

PART TWO



5.2.1 数据仓库的概念与组成

1. 数据仓库的概念

20世纪80年代初，美国著名数据仓库专家W.H.Inmon在其《建立数据仓库》一书中首次提出了数据仓库的概念：**数据仓库是面向主题的、集成的、具有时变性以及非易失的数据集合，用以支持经营管理中的决策制定过程。**

数据仓库是一种联机分析处理（On-Line Analytical Processing，OLAP）数据库，它通过ETL（Extract-Transform-Load）工具将事务数据库中的数据进行抽取、转换和加载，使其能够更加满足分析的要求，通过分析平台根据用户的需求提供不同类型的数据集合，用于数据的深度理解与分析。简而言之，从数据仓库挖掘出对决策有用的信息与知识，是建立数据仓库的根本目的。

资料整理
火龙果软件
荐

5.2.1 数据仓库的概念与组成

1. 数据仓库的概念

由W.H.Inmon给出的数据仓库的定义可知，数据仓库具有以下4个特征：

面向主题的

数据仓库的数据是面向主题的，这是数据仓库的核心特征，是建立数据仓库的基本原则。传统数据库围绕着系统的功能应用进行组织和设计，目的是保障系统的功能能够运行实现，即完成具体的事务。数据仓库的目的是支持决策制定，决策决定了数据仓库的主题，决定了数据仓库围绕什么组织数据，与主题无关的数据不会被导入到数据仓库中。数据在进入数据仓库之后，就完成了从面向应用到面向主题的转变。

集成的

在构建数据仓库的过程中，**外部数据源并不唯一，数据格式也并不完全相同，将这些有差异的数据进行抽取、转换和加载，最终形成一个统一的整体**。要注意的是，这个过程绝不是将数据进行单纯的复制，然后迁移到数据仓库中，而且要将这些数据统一到数据仓库的数据模式上来，还要监视源数据发生的变化，并对数据仓库进行更新，应该说数据仓库是对源数据的增值和统一。数据集成是数据仓库技术中非常关键且非常复杂的内容。

5.2.1 数据仓库的概念与组成

1. 数据仓库的概念

由W.H.Inmon给出的数据仓库的定义可知，数据仓库具有以下4个特征：

时变性

数据仓库并不是一成不变的，它具有时变性，但是这种随着时间发生的变化与事务数据库中的数据变化并不完全相同，**并不是简单的新增、更新、删除或查询。数据被装载到数据仓库后，同时会产生一个对应的时间戳，**随着时间的推移，在实际业务中，这个数据字段可能早已发生变化，但是在数据仓库中，某个时间戳下对应的数据并不会发生改变，新的数据进入会有对应的新的时间戳。**时变性保证了数据仓库能够具有记录数据项的历史变化的功能，为后续分析和决策提供了依据。**

非易失的

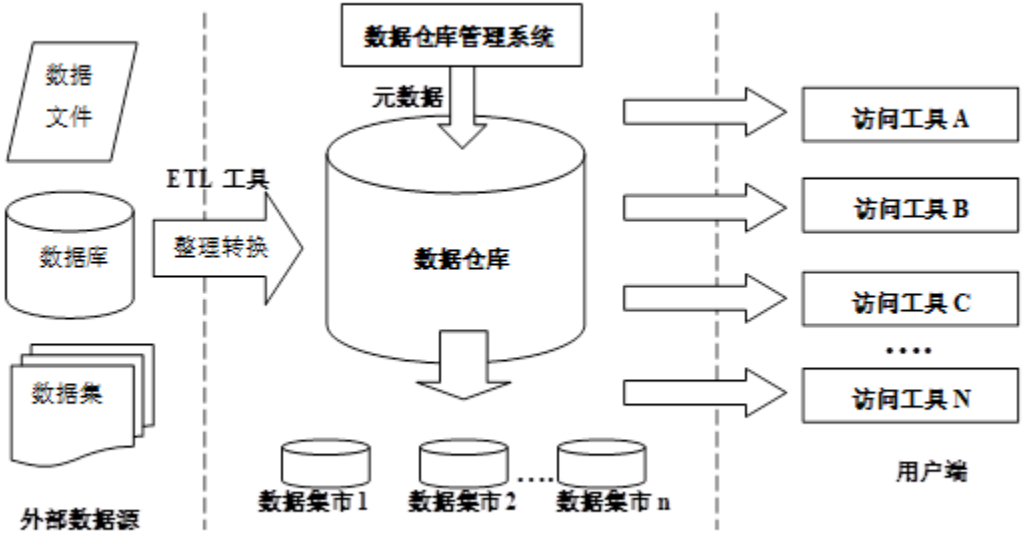
数据仓库内的数据主要是为企业决策分析提供支持，帮助企业更好地利用数据，挖掘数据中的价值，提高企业决策的效率和有效性。如果数据仓库内的数据总是在发生变化，将无法保证数据的有效性和准确性，增加企业决策工作的负担，降低决策的价值。**非易失特征保证了在数据仓库中，对于数据的操作主要是查询，不能轻易地进行更新和删除操作，维护了数据的稳定性。**非易失作为数据仓库的重要特征，为数据仓库在企业决策分析中发挥作用奠定了坚实的基础。

火龙果软件

5.2.1 数据仓库的概念与组成

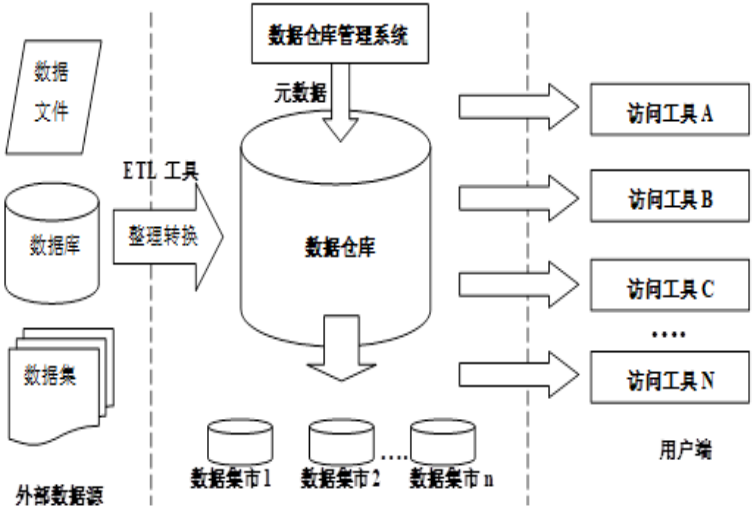
2. 数据仓库的组成

数据仓库系统由**数据仓库、ETL工具、元数据、访问工具、数据集市和数据仓库管理系统**6个部分组成，如图所示。



资料整理
火龙果软件
荐

5.2.1 数据仓库的概念与组成



2. 数据仓库的组成

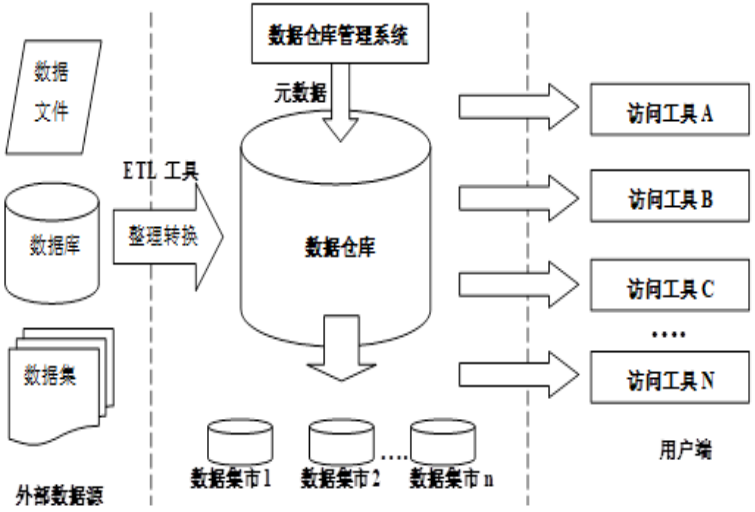
（1）**数据仓库**，是整个数据仓库系统的核心，是数据仓库系统存储数据的区域。

（2）**ETL工具**，它能够将数据从各种源数据库中抽取出来，按照数据仓库预定的主题对数据进行必要的转换和整理，最后装载到数据仓库中。

（3）**元数据**，是描述数据仓库中数据的数据，是数据仓库运行和维护的中心，它全面刻画了数据的内容、结构、获取方法和访问方法等。数据仓库服务器需要利用元数据来存储和更新数据仓库的数据，用户需要通过元数据来了解和访问数据。

资料整理
火龙果软件
荐

5.2.1 数据仓库的概念与组成



2. 数据仓库的组成

（4）访问工具，为用户访问数据仓库提供工具支撑，是数据仓库系统实现决策支持作用的关键部分。

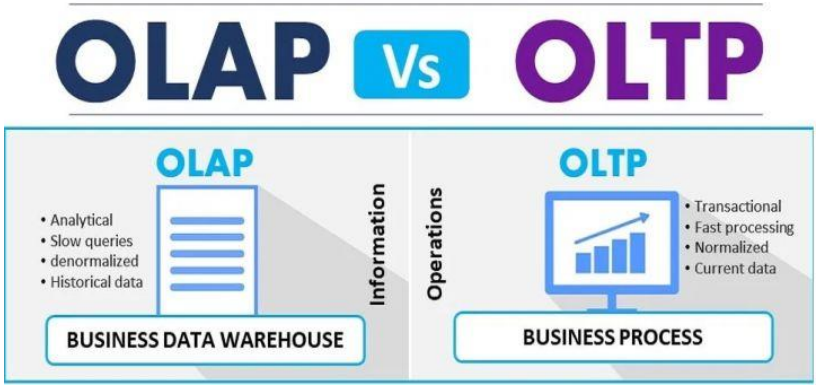
（5）数据集市（Data Mart）一般是为了特定的应用目的或应用范围，面向企业的某个部门（或主题），在逻辑或物理上划分出来的数据仓库的一个子集，可以理解为部门级数据仓库。

（6）数据仓库管理系统（Data Warehouse Management System，DWMS），是整个数据仓库系统的运转管理引擎。

资料整理
火龙果软件
荐

5.2.2 数据仓库与操作数据库的区别

操作数据库是面向联机事务处理（On-Line Transaction Processing，OLTP）的数据库，用于支持日常事务系统的运转，是当前应用非常广泛的数据库。例如，超市的交易系统数据库、医院的门诊数据库等都是典型的操作数据库，其中存储的主要是日常事务数据。下面分别从**系统层面、数据组织层面、数据操作层面**介绍一下两类数据库的特点。



资料整理
火龙果软件
荐

5.2.2 数据仓库与操作数据库的区别

1. 系统层面

数据仓库创建的目的是为了满足企业管理活动、制定决策的需要，它是面向特定主题而设计创建、组织数据的，管理层和决策相关人士是数据仓库的需求者和使用者。

操作数据库的存在是为了满足企业事务性活动数据存取的需要，支持日常事务操作，一线的业务人员、数据库管理员（DBA）以及数据库相关人士是主要的使用者。

2. 数据组织层面

数据仓库的数据来源是多元化的，数据源有多个、多种类型，存储的数据主要是历史数据以及经过转换处理的派生数据，数据仓库内数据的组织模型大多为星形或雪花形，呈现为多维数据立方体，数据综合性程度高。

操作数据库的数据来源比较单一，主要是事务系统中产生的描述事务的细节性数据，数据状态实时刷新，保持为当前时间点的最新状态，操作数据库主要基于实体-联系（E-R）模型组织数据，例如关系表。

资料整理
火龙果软件

5.2.2 数据仓库与操作数据库的区别

3. 数据操作层面

当数据装载到数据仓库内后，对于存储数据的操作主要是复杂的查询，并不会进行增删改操作，保证了数据的稳定性。此外，对数据仓库内的数据进行调用分析时，为了保证决策的有效性，涉及的数据量将会很大，系统响应的时间相对较长。

作为事务系统的存储系统，操作数据库会接收到用户端频繁地存取操作，并做出快速响应，用户的操作涵盖了增加、删除、更改、查询多个方面，但是每一个用户发送的请求涉及到的数据量相对较小，主要是与该用户有关的数据。

资料整理
火龙果软件
荐

5.2.2 数据仓库与操作数据库的区别

数据仓库与操作数据库的区别

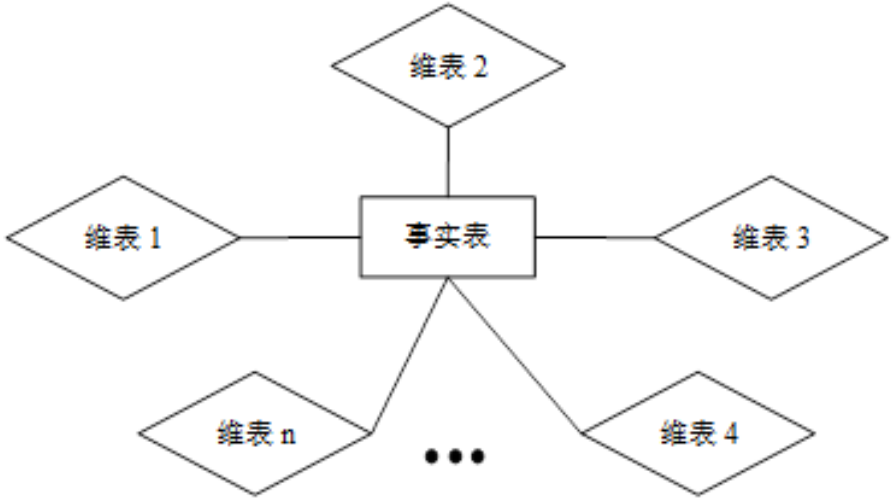
	数据仓库	操作数据库
系统目的	支持企业分析、决策	支持企业事务操作
系统设计	面向主题	面向事务
使用人群	管理层、决策专家	用户、业务人员、数据库管理员（DBA）、数据库专家
数据来源	多元化	单一
数据内容	历史数据，派生数据	当前数据
数据模型	多维模型，星形或雪花形	基于实体-联系（E-R）模型
数据特点	综合数据，稳定	细节数据，动态
存取操作	复杂查询	增加、删除、更改、简单查询
存取频率	相对较低	非常高
响应时间	相对较慢	非常快
数据操作规模	非常大	相对较少
衡量指标	信息输出有效性	性能、可用性、可靠性

资料整理
火龙果软件
荐

5.2.3 数据仓库的数据模型

1. 星形模式

星形模式是**最简单**的多维数据模式，由两类表（事实表和维表）通过星形方式连接而成，**事实表是星形模式的核心，围绕事实表的是维表**，如图所示。



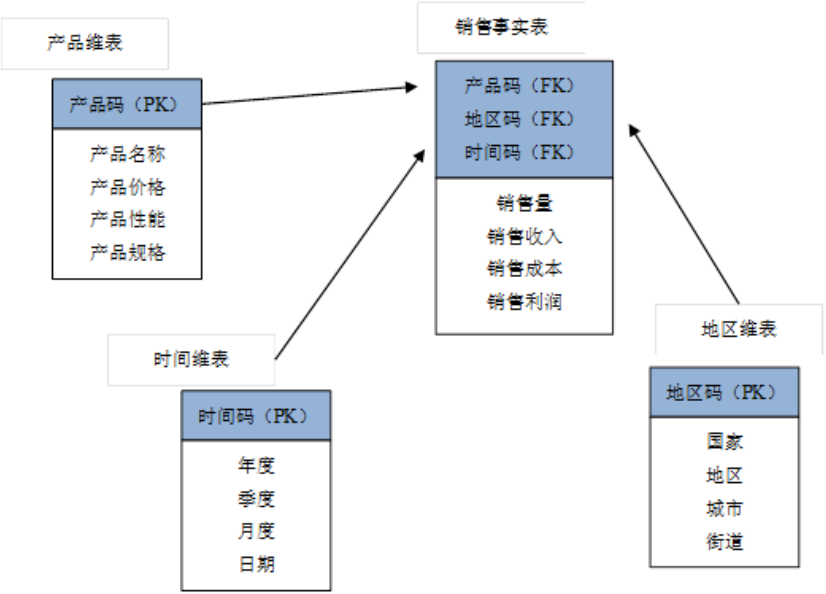
资料整理
火龙果软件
荐

5.2.3 数据仓库的数据模型

1. 星形模式

事实表主要包含了业务数据信息，用于分析性查询，数据取值一般是度量值，数据量非常大。**维表包含了事实表中所有维度的详细信息**，一般是作为查询的约束条件，数据取值一般为描述性信息。

维表与事实表的关系是一对多或一对一，维表中的主关键字在事实表中作为外关键字存在，所有维表的主关键字组合起来作为事实表的主关键字。星形模式的维表只与事实表发生关联，维表与维表之间没有直接联系。右图展示了某销售公司数据仓库中的星形模式。



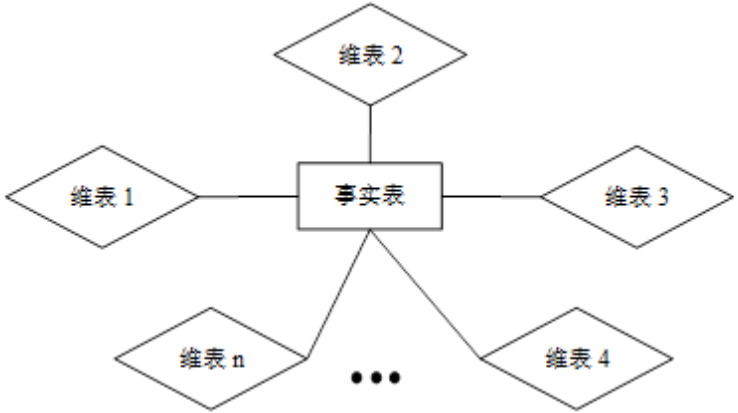
资料整理
火龙果软件
荐

5.2.3 数据仓库的数据模型

1. 星形模式

与关系模型相比，星形模式将主要数据都在庞大的事实表中，所以只要扫描事实表就可以进行查询，减少了表连接（Join）操作，查询访问效率较高。**在星形模式中，维度表被故意地非规范化了，保存了该维度的所有层次信息，减少了查询时数据关联的次数，提高了查询效率，但是维表之间的数据共用性较差。**

	字段名称	数据类型	说明
🔑	ID	自动编号	
	username	文本	员工用户名
	checkname	文本	考核名称
	checkdate	文本	考核日期
	checkcommment	备注	考核评语
	checksort	文本	考核类型
	checktype	文本	考核方式
	checkresult	文本	考核结果
	remark	备注	备注说明
	recname	文本	记录人
	recdate	日期/时间	记录时间
	updatename	文本	更改人
	updatedate	日期/时间	更改时间

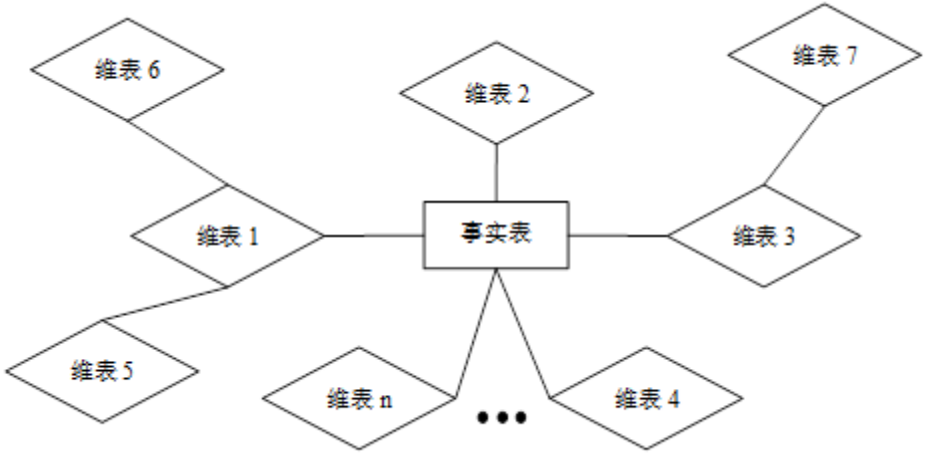


资料整理
火龙果软件
荐

5.2.3 数据仓库的数据模型

2. 雪花模式

在进行数据仓库的数据建模时，有时因为维的层次比较复杂，**在星形模式下，用一个维表来描述可能会形成过多的冗余数据**，因此可以用多个层级的维表来描述事实表中的一个维度，这种模式称为雪花形模式，如图所示。**本质上，雪花形模式是对星形模式的扩展。**



资料整理
火龙果软件
荐

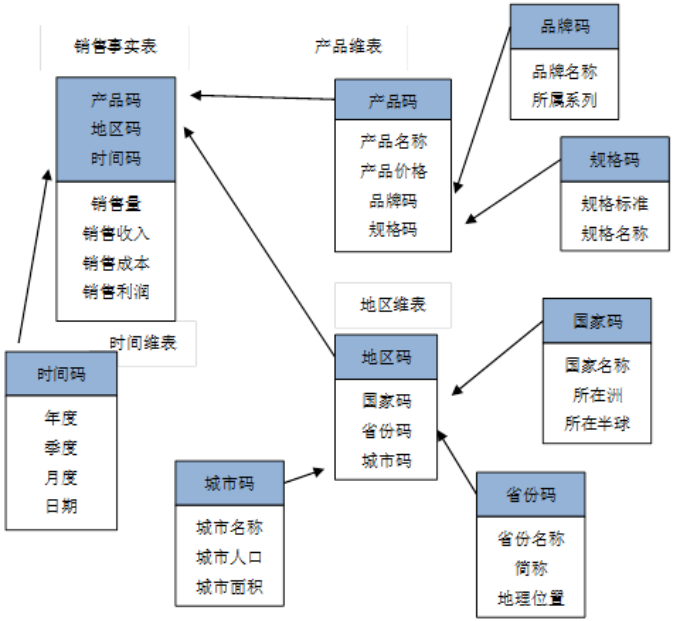
5.2.3 数据仓库的数据模型

2. 雪花模式

在雪花形模式中，**有些维表并不直接与事实表关联，而是通过其他维表与事实表关联**，特别是派生维和实体属性对应的维，这样就减少了事实表中的一条记录。

因此，当维度较多特别是派生维和实体属性维较多时，适合使用雪花形模式。

右图展示了某销售公司数据仓库的雪花形模式。



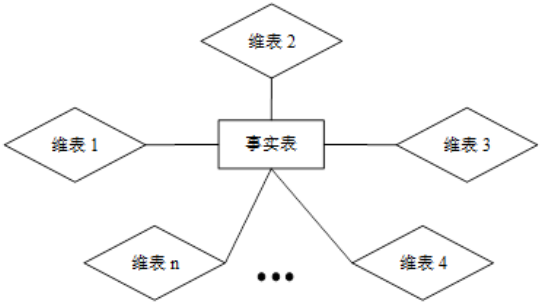
资料整理
火龙果软件
荐

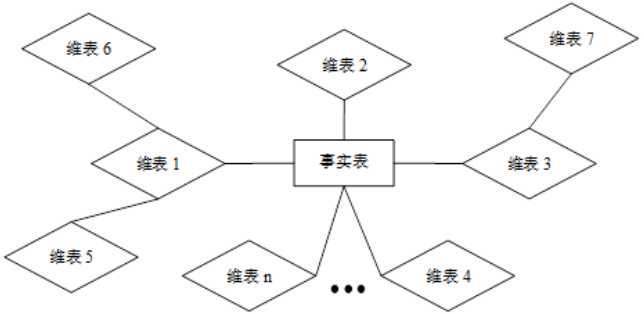
5.2.3 数据仓库的数据模型

2. 雪花模式

在星形模式中，维表并不是规范化的。雪花形模式将星形模式的维表进行规范化，才能将原有的维表扩展出下一层级的维表，用不同维表之间的关联实现维的层次。

虽然雪花形模式的维表规范化实现了维表重用，简化了维护工作。**但是，查询时使用雪花形模式要比星形模式进行更多的关联操作，降低了查询效率。**



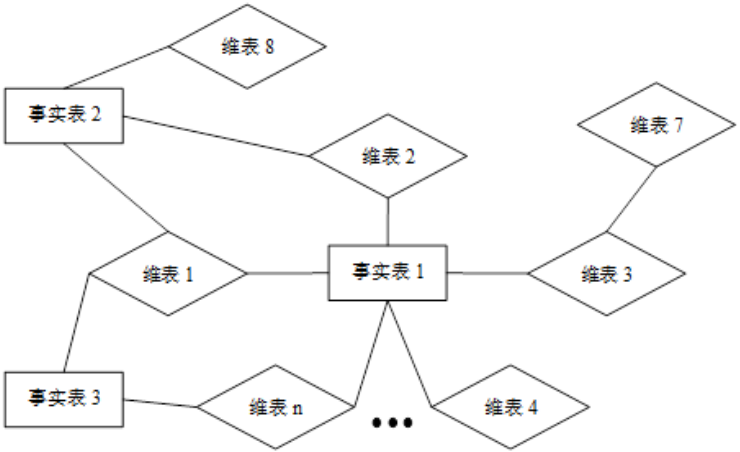


资料整理
火龙果软件
荐

5.2.3 数据仓库的数据模型

3. 事实星座模式

在目前的实际应用中，数据仓库通常由多个主题构成，并不会只有一个事实表，**一般包含多个事实表，维表也是多层级的、公共的，各个事实表之间可能会共享某些维表**，这种模式可以看做是星形模式的汇集，称为事实星座模式。事实星座模式的结构如图所示。



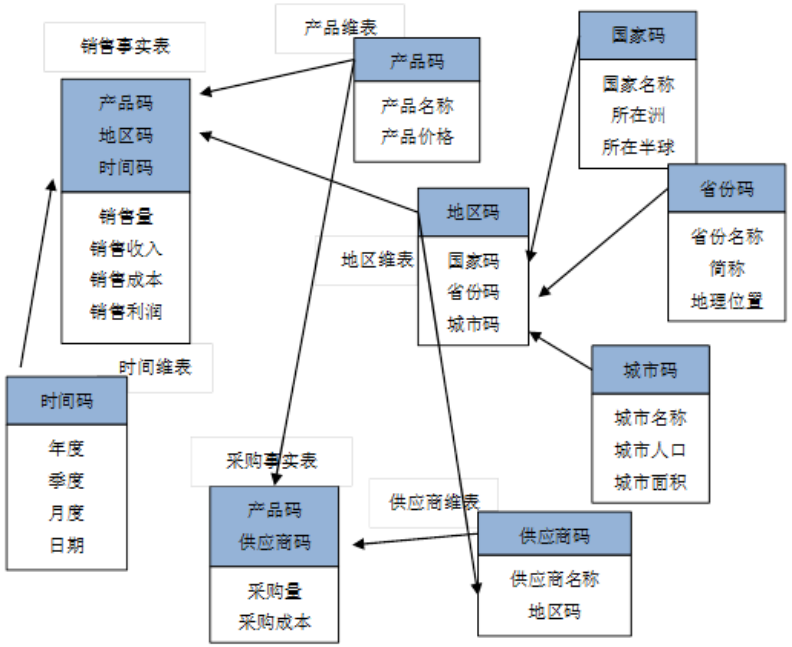
资料整理
火龙果软件
荐

5.2.3 数据仓库的数据模型

3. 事实星座模式

事实星座模式是数据仓库最常使用的逻辑模型，尤其是在企业级数据仓库中，事实星座模式占比远超于星形模式和雪花形模式。

右图展示了某销售公司数据仓库的事实星座模式。

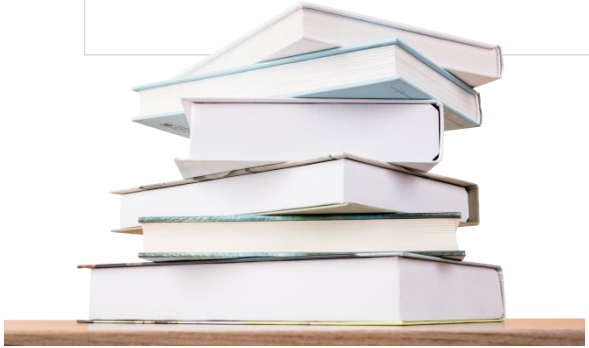


资料整理
火龙果软件
荐

03

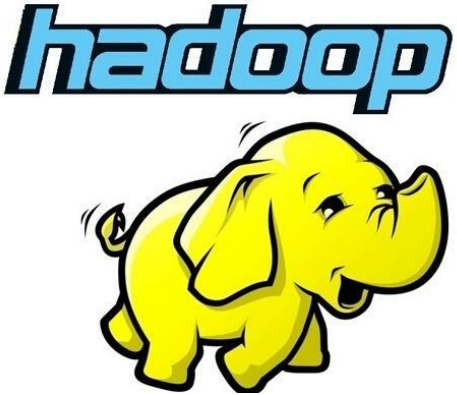
大数据的处理框架

PART THREE



5.3.1 Hadoop

由美国阿帕奇（Apache）软件基金会所开发的Hadoop是**一个开源的分布式系统基础架构**，任何用户均可以通过这个框架来开发和运行能够处理大数据的应用程序。Hadoop包含两大核心部分：**HDFS和MapReduce**，分别实现了大数据集的数据分区和并行计算。**Hadoop集群具有良好的横向扩展性，其存储和计算能力的随着集群主机数量的增加而不断扩展**，拥有数百个主机的集群可以快速处理以PB为单位的数据集。

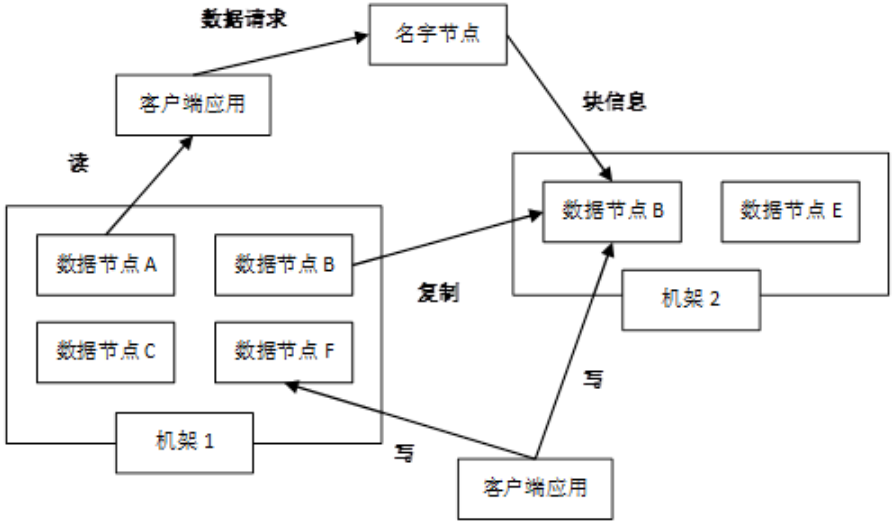


资料整理
火龙果软件
荐

5.3.1 Hadoop

1. HDFS

HDFS是Hadoop中实现大数据存储功能的分布式文件系统，采用主从架构，由一个名字节点（Name Node）和大量数据节点（Data Node）组成，如图所示。



资料整理
火龙果软件
荐

5.3.1 Hadoop

（1）名字节点

名字节点是HDFS中的核心节点，确定了文件系统内部唯一的命名空间，负责管理文件系统内部的名字空间（Name Space）、元数据以及控制客户端对文件的访问。**名字节点不存储文件数据本身，存储的是文件数据块到具体数据节点的映射关系。**名字节点负责执行文件系统名字空间的相关操作，如打开、关闭和重命名文件或目录等。

名字节点将文件系统元数据的变动持续记录到Edit事务日志中，并将其存储在本地文件系统，同时，定期将整个文件系统命名空间（包括数据块的映射表和文件系统的配置等信息）的快照存储于本地文件系统FsImage文件中，当名字节点重启时，Edit事务日志将与FsImage文件合并，进行名字节点的灾后恢复。

资料整理
火龙果软件
荐

5.3.1 Hadoop

（2）数据节点

HDFS中文件被分割成固定大小的数据块，**由名字节点负责将这些数据块分配到相应的数据节点上，数据节点负责处理文件系统客户端的读写请**求，在名字节点的统一调度下进行数据块的创建、删除和复制，数据节点上存储的是文件数据本身。

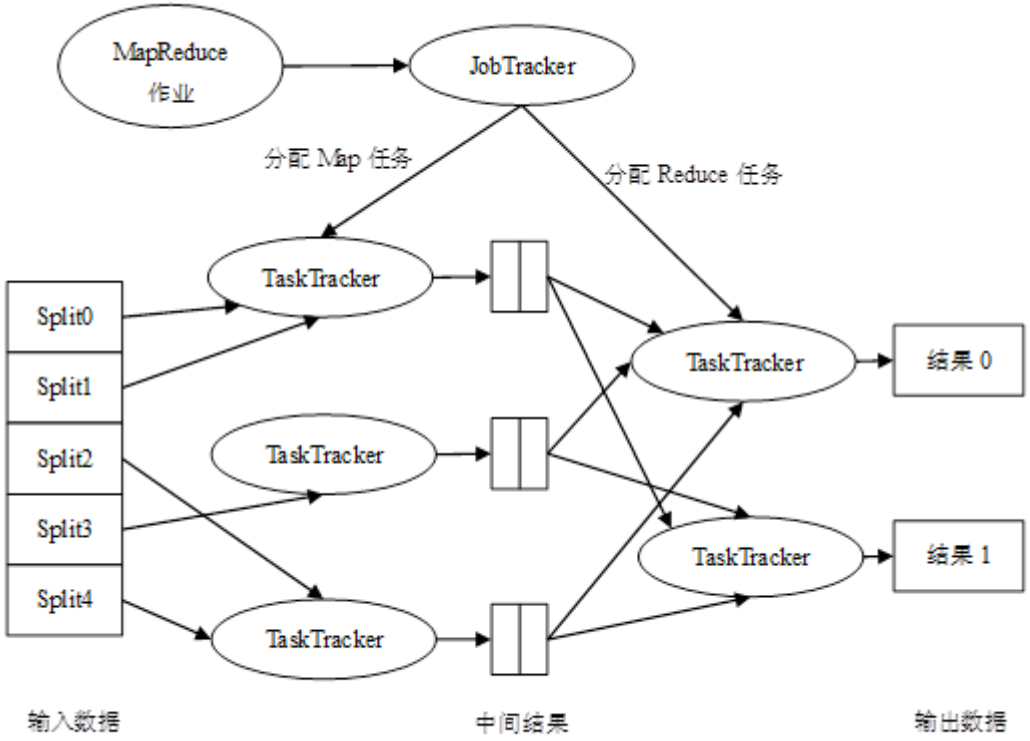
数据节点定期向名字节点发送心跳（heartbeat）消息，每条信息包含一个数据块报告，名称节点可以根据这个报告验证块映射和其他文件系统元数据的情况。如果数据节点不能发送心跳消息，名称节点将采取修复措施，重新复制在该节点上丢失的数据块。**为了尽量减小系统全局的带宽消耗读延迟，HDFS会尽量返回给读操作一个离它最近的数据块副本。**假如在读节点的同一个机架上就有，则直接读取该数据块副本，如果集群跨越多个数据中心，那么本地数据中心的数据块副本优先于远程数据中心的数据块副本。

资料整理
火龙果软件工作室

5.3.1 Hadoop

2. MapReduce

Hadoop中的MapReduce是基于谷歌的MapReduce模型的开源实现，是一种并行计算架构，**其中存在JobTracker和TaskTracker两类作业单元**，如图所示。



资料整理
火龙果软件
荐

5.3.1 Hadoop

JobTracker主要对MapReduce作业的执行进行监督和管理，TaskTracker负责MapReduce作业中Map（映射）任务和Reduce（归约）任务的具体实现。

处理一个MapReduce作业主要包括以下几个过程：

（1）Data Splitting（数据分割）：当把一个MapReduce作业提交给Hadoop集群时，**相关的输入数据首先被划分为多个片段**，然后JobTracker挑选空闲的TaskTracker，**并将切分的数据片段发送到TaskTracker上；**

（2）Mapping（映射）：TaskTracker根据应用的需求执行Map任务，将不同的数据片段映射成相应的键值对（Key-Value），存储到哈希结构<k1，v1>中；

资料整理
火龙果软件
荐

5.3.1 Hadoop

JobTracker主要对MapReduce作业的执行进行监督和管理，TaskTracker负责MapReduce作业中Map（映射）任务和Reduce（归约）任务的具体实现。

处理一个MapReduce作业主要包括以下几个过程：

（3）Shuffling（洗牌）：为了提高计算效率、同时避免Reducing阶段发生数据相关性，**要对由Map任务产生的中间结果进行再次划分和合并处理**，保证具有相关性的数据发送到执行Reduce任务的同一个TaskTracker上；

（4）Reducing（归约）：由JobTracker挑选空闲的TaskTracker并行地执行Reduce任务，分析所接受到的键值对，如果是键相同的配对，那就将它们的值进行合并，从而获得与每个键值相对应的数据集合，作为运算结果。

资料整理
火龙果软件
荐

5.3.1 Hadoop

3. Hadoop的优势

基于HDFS和MapReduce的强大性能，开源的Hadoop自诞生起就受到广大学者和开发人员的喜爱，其优势主要体现在：

高可靠性和容错性

HDFS对于大数据集**进行分区存储、备份**以及MapReduce的并行计算架构均提高了系统可靠性和容错性。比如，Hadoop中的数据都有多处备份，**如果一个地方的数据发生丢失或损坏，它能够自动地从其他的副本进行复原**。类似的，失败的计算任务也可以分配到新的资源节点上，进行自动重试。

资料整理
火龙果软件
荐

5.3.1 Hadoop

3. Hadoop的优势

基于HDFS和MapReduce的强大性能，开源的Hadoop自诞生起就受到广大学者和开发人员的喜爱，其优势主要体现在：

高扩展性

扩展分为纵向和横向，纵向是增加单机的资源，总是会达到瓶颈，而横向则是增加集群中的机器数量，获得近似线性增加的性能，不容易达到瓶颈。**Hadoop搭建的集群系统中的节点资源，采用的就是横向方式，可以方便地进行扩充，**无论是存储性能还是计算性能均可随节点数的增加而保持近似于线性的提高。

资料整理
火龙果软件
荐

5.3.1 Hadoop

3. Hadoop的优势

基于HDFS和MapReduce的强大性能，开源的Hadoop自诞生起就受到广大学者和开发人员的喜爱，其优势主要体现在：

低成本

基于高可靠性和容错性，Hadoop搭建的集群系统**可选用价格便宜、易于扩展的低端商用服务器**，甚至是个人电脑都可以称为资源节点，不必选用价格昂贵、不易扩展的高端服务器。

高效性

由于采用多个资源并行处理，**使得Hadoop不再受限于单机操作（特别是较慢的磁盘I/O读写），**可以快速地完成大规模任务。加上其所具有的高扩展性，**随着硬件资源的增加，性能将会得到进一步的提升。**另外，HDFS和MapReduce均采用数据就近处理的策略，减少系统中的数据通信开销。

透明性

Hadoop提供了一种抽象机制，对应用开发人员隐藏了系统层细节，应用开发人员可以在不了解底层细节的情况下，开发和运行分布式程序。比如，在MapReduce中，开发人员仅需描述需要计算什么，致力于其应用本身计算问题的算法设计，而具体如何计算则交由系统处理。

火龙果软件

5.3.1 Hadoop

4. YARN

随着不断发展和应用程度的加深，人们也逐渐意识到第一代Hadoop中存在的一些问题。

比如，HDFS的名字节点和MapReduce的JobTracker均**存在单点故障风险**，不利于系统稳定。另外，将JobTracker作为Hadoop集群的资源管理器，除了MapReduce作业以外，Hadoop集群不能够再运行其他作业，容易造成集群资源的浪费。

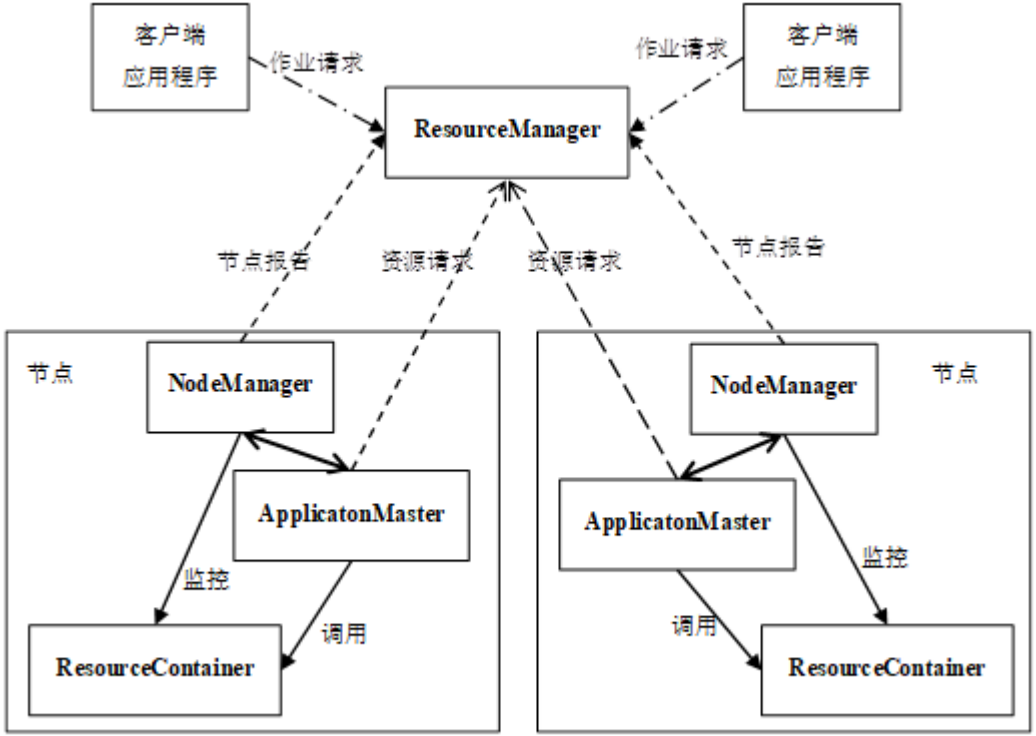
面对第一代Hadoop存在的问题，大量研究人员开始开发新的Hadoop架构，目前**已推出基于YARN（Yet Another Resource Negotiator）的Hadoop 2.x系列版本**，使得Hadoop架构更加完善。

资料整理
火龙果软件
荐

5.3.1 Hadoop

4. YARN

YARN是Hadoop中全新的分布式资源管理器，也称为通用应用程序调度器，**为所有建立在YARN之上的应用程序提供了统一的资源管理和调度服务**，其架构如图所示。



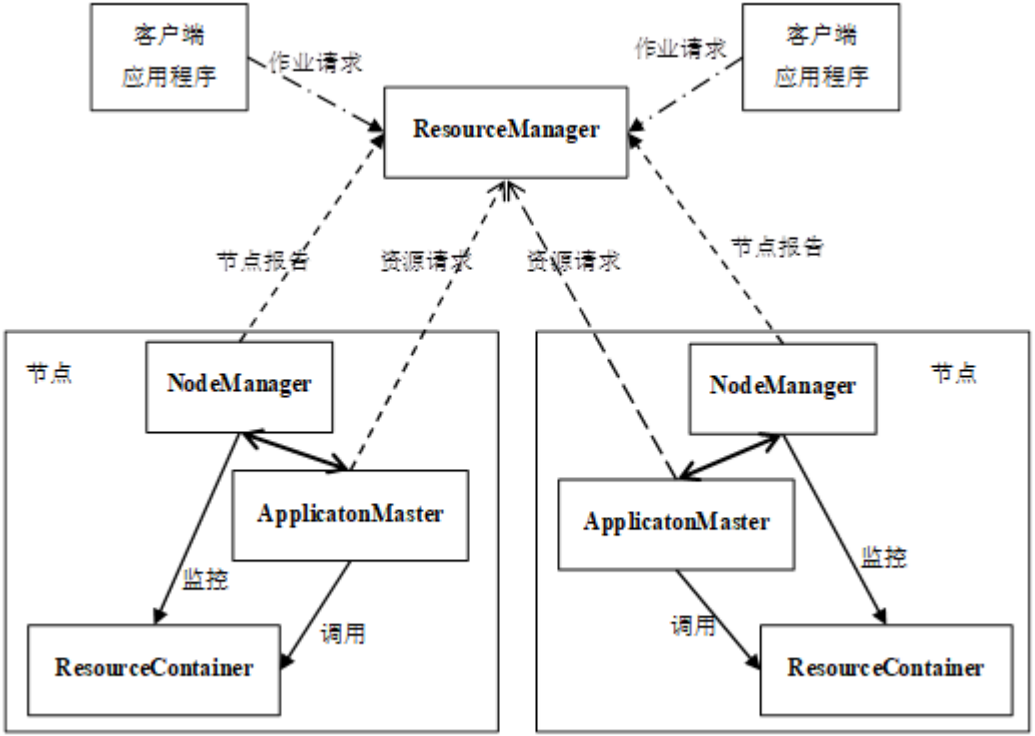
资料整理
火龙果软件
荐

5.3.1 Hadoop

4. YARN

为从根本上解决原有MapReduce架构的性能瓶颈，**YARN将原有JobTracker的主要功能（资源管理和作业调度/监控）进行了分解**，创建了一个全局的Resource Manager和若干个管理应用程序的Application Master。

这里所说的应用程序既可以是传统意义上的MapReduce作业，也可以是基于有向无环图（Directed Acyclic Graph，DAG）的作业。



资料整理
火龙果软件
荐

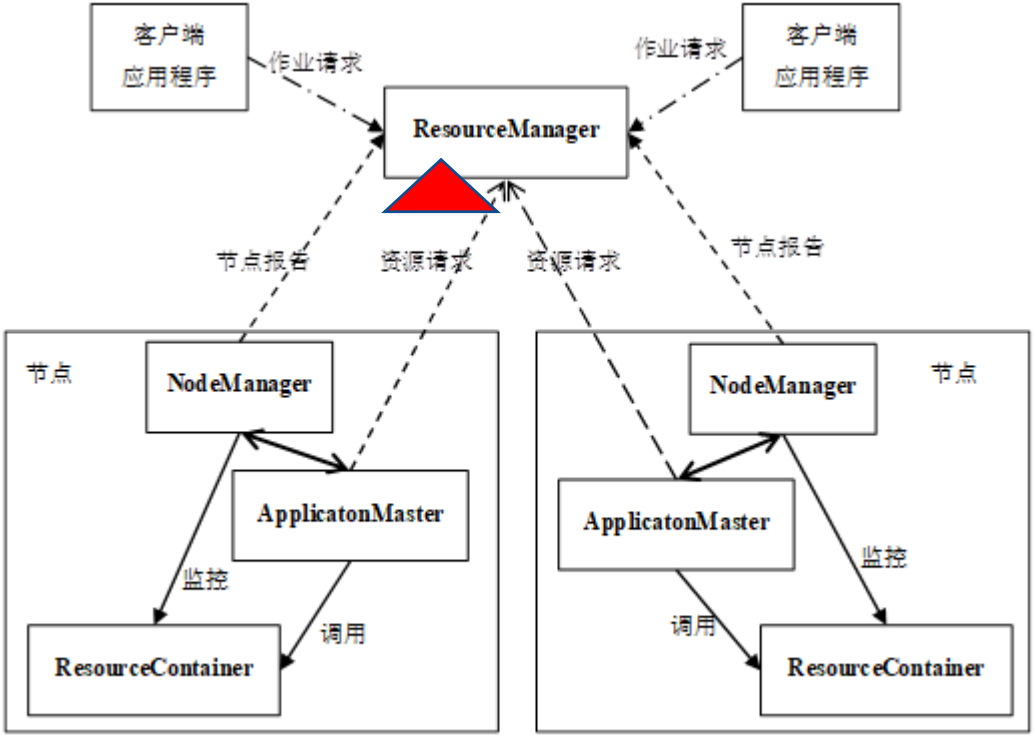
5.3.1 Hadoop

4. YARN

YARN主要包含以下**4个**组件：

（1）**Resource Manager是一个全局的资源管理器**，负责整个系统的资源管理和分配。它主要由两个组件构成：**调度器（Scheduler）和应用程序管理器（Application Manager）**。

调度器负责分配应用程序运行所需的资源；应用程序负责处理客户端提交的作业以及协调Resource Container来给Application Master使用。



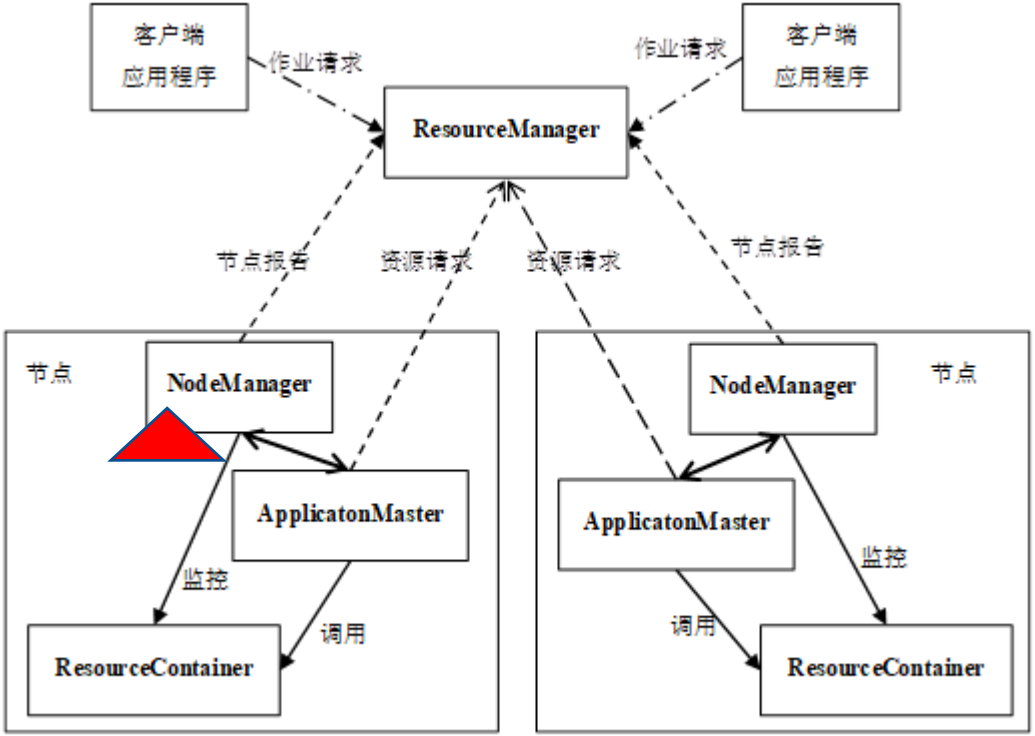
资料整理
火龙果软件
荐

5.3.1 Hadoop

4. YARN

YARN主要包含以下**4个**组件：

（2）**Node Manager**是Resource Manager在集群中所有运行节点上的代理，**主要负责启动Resource Container以及周期性地监视该Resource Container运行占用的资源情况**，若是超过了原有声明的资源量，则会终止相关进程的运行。



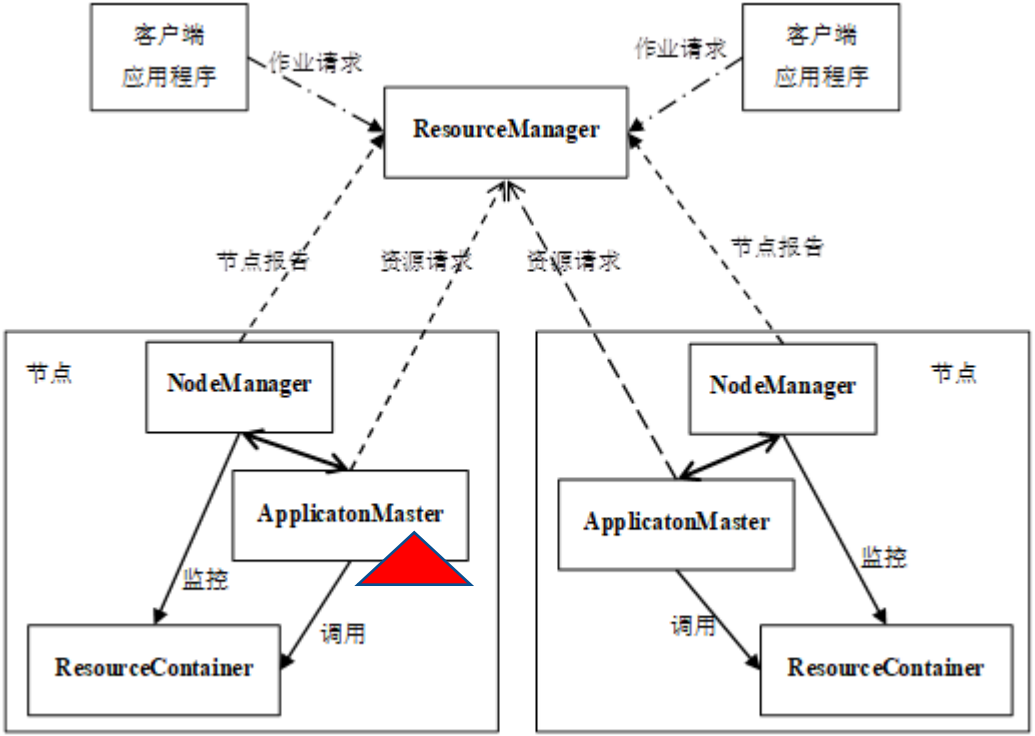
资料整理
火龙果软件
荐

5.3.1 Hadoop

4. YARN

YARN主要包含以下**4个**组件：

（3）**ApplicatonMaster是表示针对每个应用程序的应用管理器**，与Node Manager协同工作管理和监控部署在YARN集群上各种应用程序的运行实例。



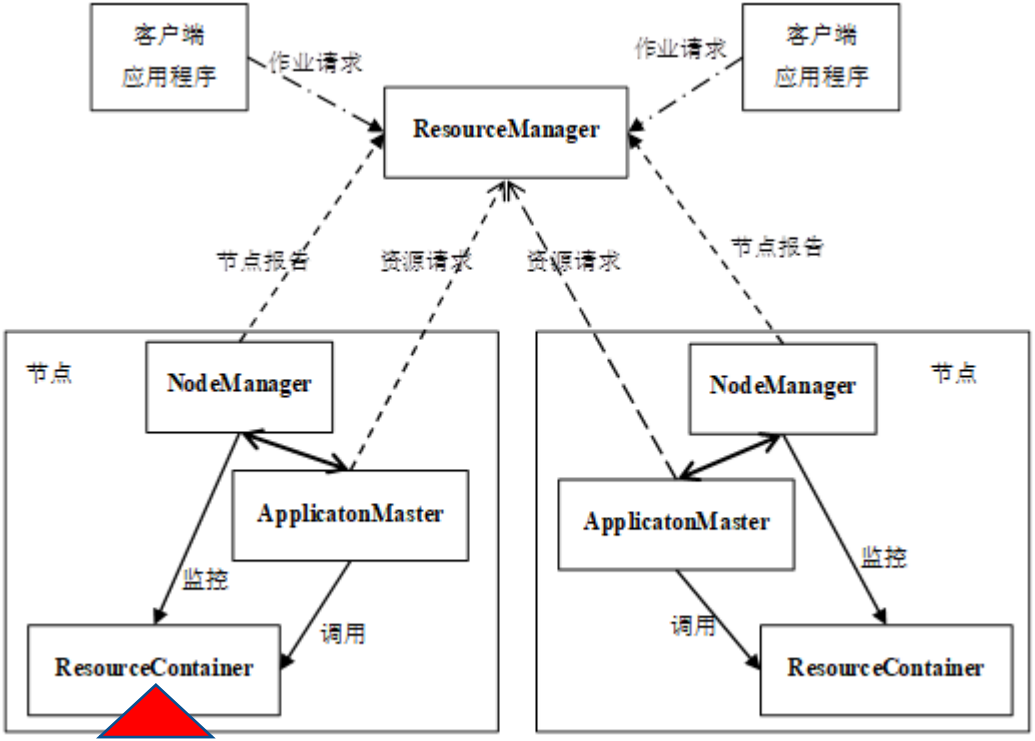
资料整理
火龙果软件
荐

5.3.1 Hadoop

4. YARN

YARN主要包含以下**4个**组件：

（4）**ResourceContainer是用来管理YARN集群中各种资源的容器**，包括内存、CPU、磁盘和网络等资源，每个应用程序需要不同类型的资源，因此需要不同的容器，与现有的MapReduce固定类型的资源使用模型有显著区别。



资料整理
火龙果软件
荐

5.3.1 Hadoop

4. YARN

此外，基于YARN的Hadoop 2.x系列版本还为HDFS引入了两个重要的新功能：**联盟（Federation）和高可用性（High Available，AH）**。

联盟允许在多个名字节点主机之间共享元数据，有助于HDFS增强扩展性，并且还提供数据隔离功能，允许不同的应用程序或开发团队运行自己的名字节点，而不必担心影响同一集群上的其他名字节点。

高可用性消除了原有架构中存在的单点故障风险，提供了主备名字节点自动切换的能力。

资料整理
火龙果软件
荐

5.3.2 Spark

随着大数据的发展，人们对大数据的处理要求越来越高，原有的批处理架构MapReduce适合离线计算，无法满足对实时性要求较高的作业，如实时推荐、用户行为分析等。因此，美国加州大学伯克利分校的AMP实验室用Scala语言开发出了开源分布式轻量级通用计算框架Spark，可以有效地满足大数据处理的需求。

资料整理
火龙果软件
荐

5.3.2 Spark

1. Spark的特征

Spark框架采用了基于内存的设计思想，可以将作业处理过程中的中间结果保存在内存中，不再需要读写底层的HDFS，在性能上比MapReduce快100倍左右，但其对内存的要求非常高，集群中一个节点通常需要配置24GB的内存。**因此，有时把MapReduce称为批处理计算框架，把Spark称为实时计算框架、内存计算框架或流式计算框架。**

Spark易于使用，支持多语言（Scala、Java、R、Python等）编写程序，降低了开发者工作难度。Spark自带了80多个算子，同时允许在命令解释程序（Shell）中进行交互式计算，利用Spark可以像编写单机程序一样编写分布式程序，并搭建大数据内存计算平台，实现海量数据的实时处理。

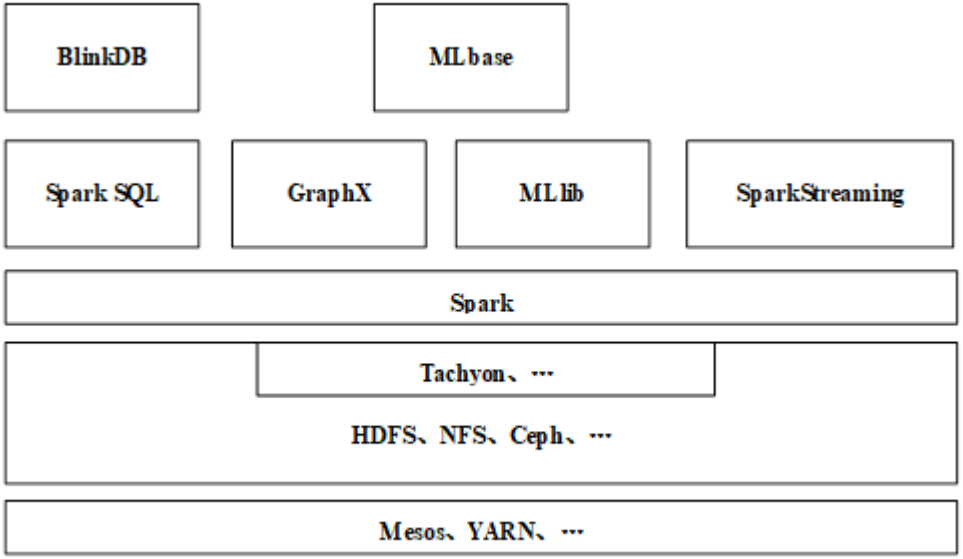
Spark充分利用和集成了Hadoop等其他第三方组件，**既可以采用独立部署的方式运行，还可以运行在YARN等集群管理系统之上，还可以运行在任何Hadoop数据源上**，如Hive、HBase、HDFS等，用户可以轻易迁移已有的持久化层数据。

资料整理
火龙果软件
荐

5.3.2 Spark

2. Spark生态系统

目前，Spark已经发展成为包含众多子项目的大数据计算平台，**为用户提供大数据处理的一站式解决方案，**以Spark为的生态系统称为伯克利数据分析栈（Berkeley Data Analysis Stack，BDAS），**打造了全栈（Full Stack）多计算范式的高效数据流水线**，帮助用户可以在同一个工作流中任意搭配使用这些计算范式。

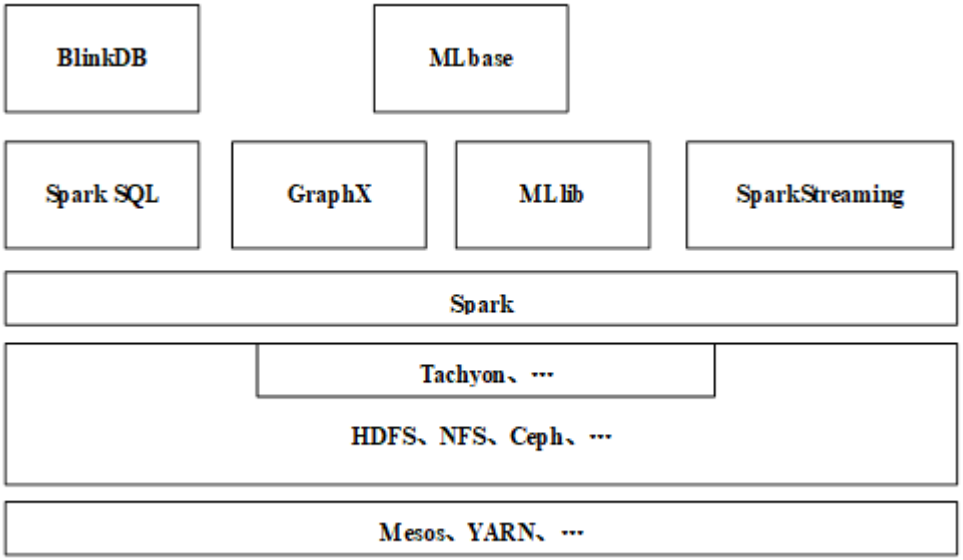


资料整理
火龙果软件
荐

5.3.2 Spark

2. Spark生态系统

除了简单的Map及Reduce功能之外，Spark生态系统还包括支持SQL查询与分析的查询引擎Spark SQL、具有机器学习功能的系统MLbase及底层的分布式机器学习库MLlib、并行图计算框架GraphX、流计算框架Spark Streaming、采样近似计算的查询引擎BlinkDB、内存分布式文件系统Tachyon、资源管理框架Mesos等子项目。这些子项目在Spark上层提供了更高层、更丰富的计算范式，如图所示。

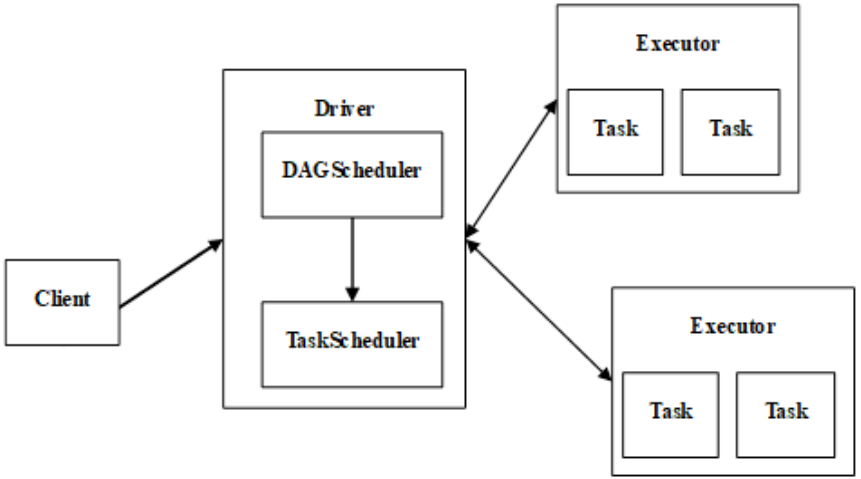


资料整理
火龙果软件
荐

5.3.2 Spark

3. Spark的组成

与Hadoop类似，**Spark也采用了主从架构的模式**，主进程节点负责控制整个集群、调配集群资源，从进程节点负责具体作业的执行、完成计算任务，Spark的基本架构如图所示。



资料整理
火龙果软件
荐

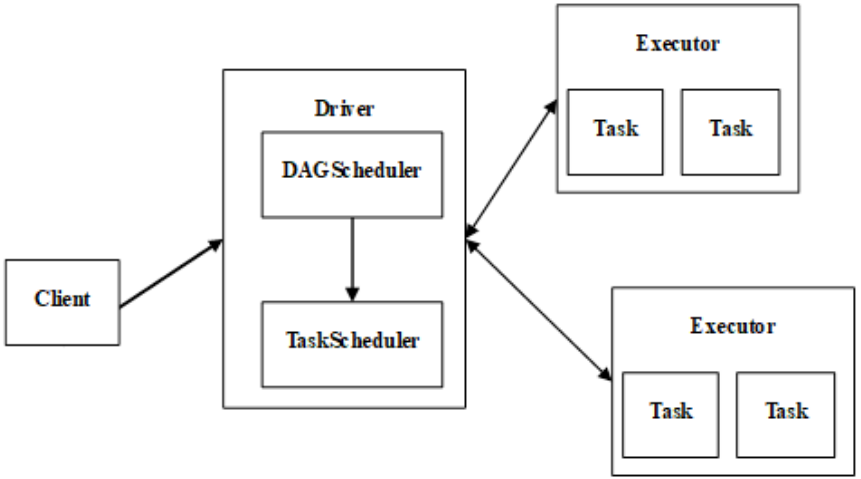
5.3.2 Spark

3. Spark的组成

Spark的核心组件主要包括：

（1）Client：客户端进程，负责提交Spark作业；

（2）SparkContext：作业的上下文，控制作业的生命周期，与Spark作业一一对应；



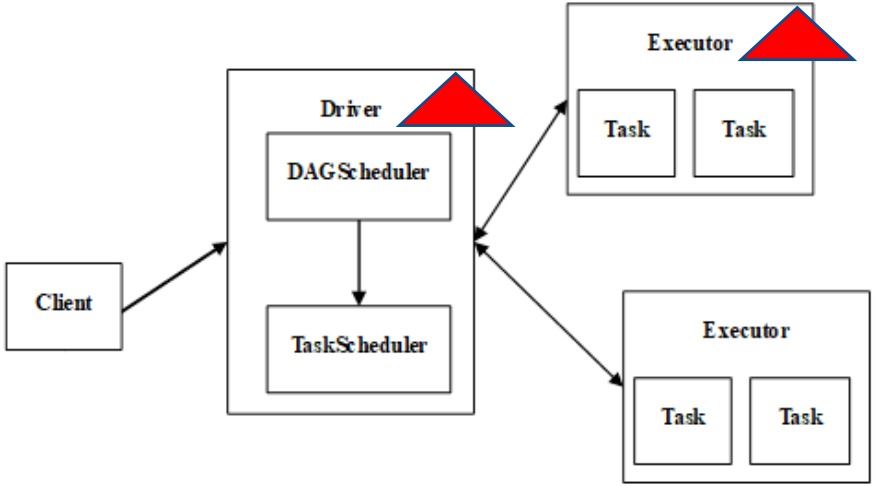
资料整理
火龙果软件
荐

5.3.2 Spark

3. Spark的组成

（3）Driver：主要负责Spark作业的解析，并创建SparkContext；**调用内部DAGScheduler将作业转化为基于阶段的DAG**，进而通过TaskScheduler分发Task给Executor执行；

（4）Executor：负责执行Driver分发的Task，集群中一个从进程节点可以启动多个Executor，**每个Executor可以执行多个Task；**



资料整理
火龙果软件
荐

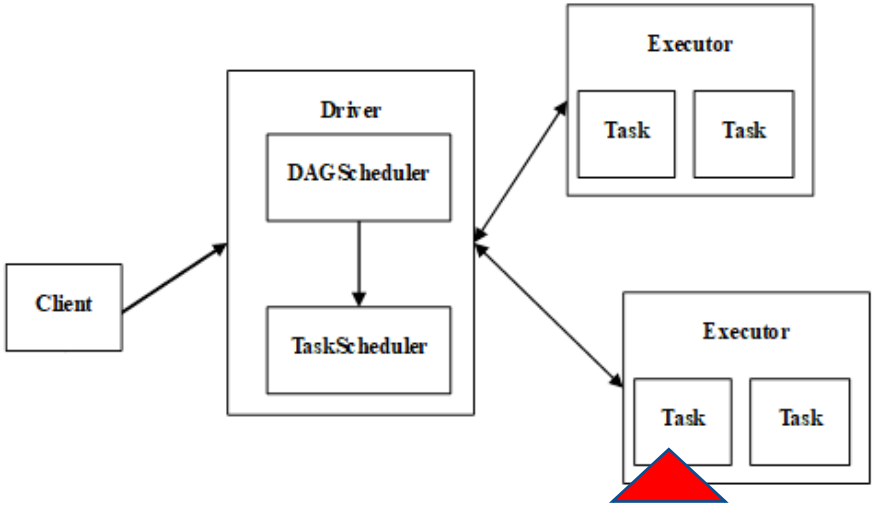
5.3.2 Spark

3. Spark的组成

（5）DAG：有向无环图，Spark实现了DAG的计算模型，将一个计算任务按照计算规则分解为若干子任务，这些子任务之间根据逻辑关系构建成有向无环图；

（6）RDD：弹性分布式数据集（Resilient Distributed Dataset），可以理解为一种只读的分布式多分区的数组，**Spark计算操作都是基于RDD进行的，是Spark的基本计算单元，一组RDD可形成执行的RDD Graph**；

（7）Task：Spark运行的基本单位，负责处理RDD的计算逻辑。



火龙果软件

5.3.2 Spark

3. Spark的组成

一个完整的Spark作业流程主要包含以下几个步骤：

（1）Client提交作业请求；

（2）根据作业请求，主进程节点调用从进程节点启动Driver组件，Driver组件向主进程节点申请资源，然后将作业转化为RDD Graph；

（3）Driver内的DAGScheduler将RDD Graph转化为基于阶段的DAG提交给TaskScheduler；

（4）TaskScheduler根据DAG调度Executor执行相应的Task，在执行的过程中，其他组件协同工作，确保整个作业顺利执行。

资料整理
火龙果软件
荐

5.3.2 Spark

4. 弹性分布式数据集RDD

RDD是Spark的核心数据结构，从字面上理解有些困难，它是一种分布式多分区只读的数组，**Spark计算操作都是基于RDD进行的。RDD具有几个特性：只读、多分区和分布式。**生成RDD的方式有很多种，可以将HDFS块文件转换成RDD，也可以由一个或多个RDD转换成新的RDD，失效的RDD会自动进行重构。

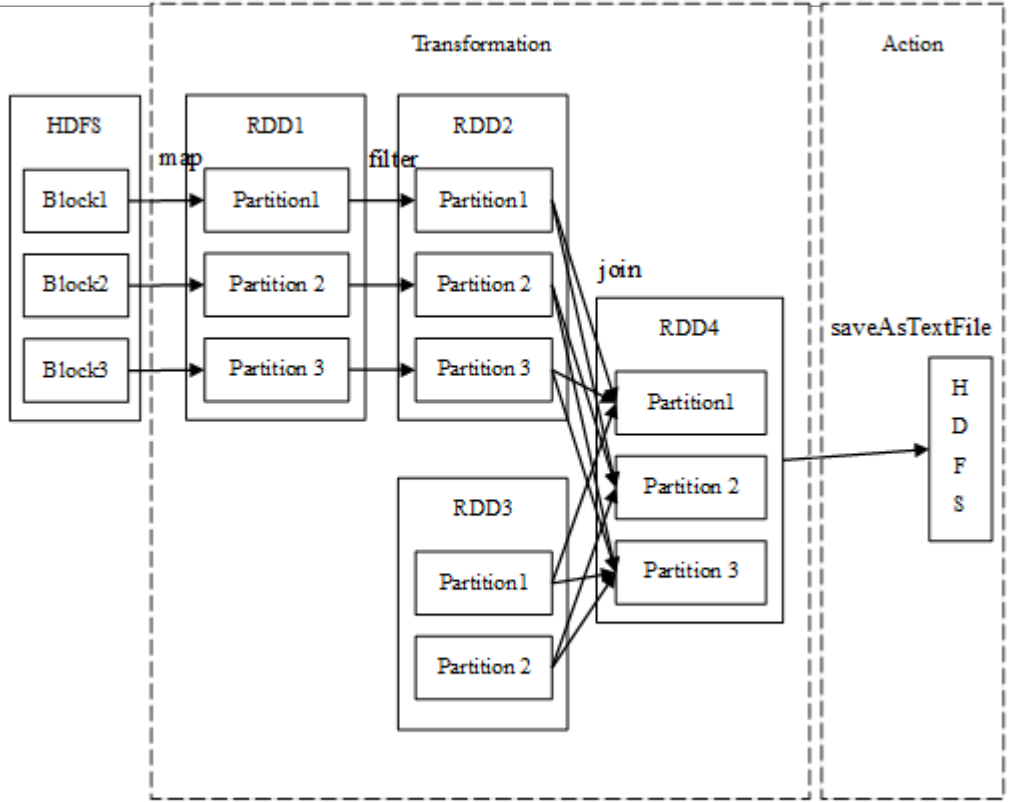
在Spark中，**RDD提供Transformation和Action两种算子。**Transformation算子内容非常丰富，采用延迟执行的方式，在逻辑上定义了RDD的依赖关系和计算逻辑，但并不会真正触发执行动作，只有等到Action算子才会触发真正执行操作。Action算子常用于最终结果的输出。

资料整理
火龙果软件
荐

5.3.2 Spark

4. 弹性分布式数据集RDD

如图所示，在Spark应用中，整个执行流程在逻辑上会形成DAG模型。Action算子触发之后，将所有累积的算子形成一个DAG，然后由调度器调度该图上的任务进行运算。Spark的调度方式与MapReduce有所不同。Spark根据RDD之间不同的依赖关系切分形成不同的阶段，一个阶段包含一系列函数执行流水线。

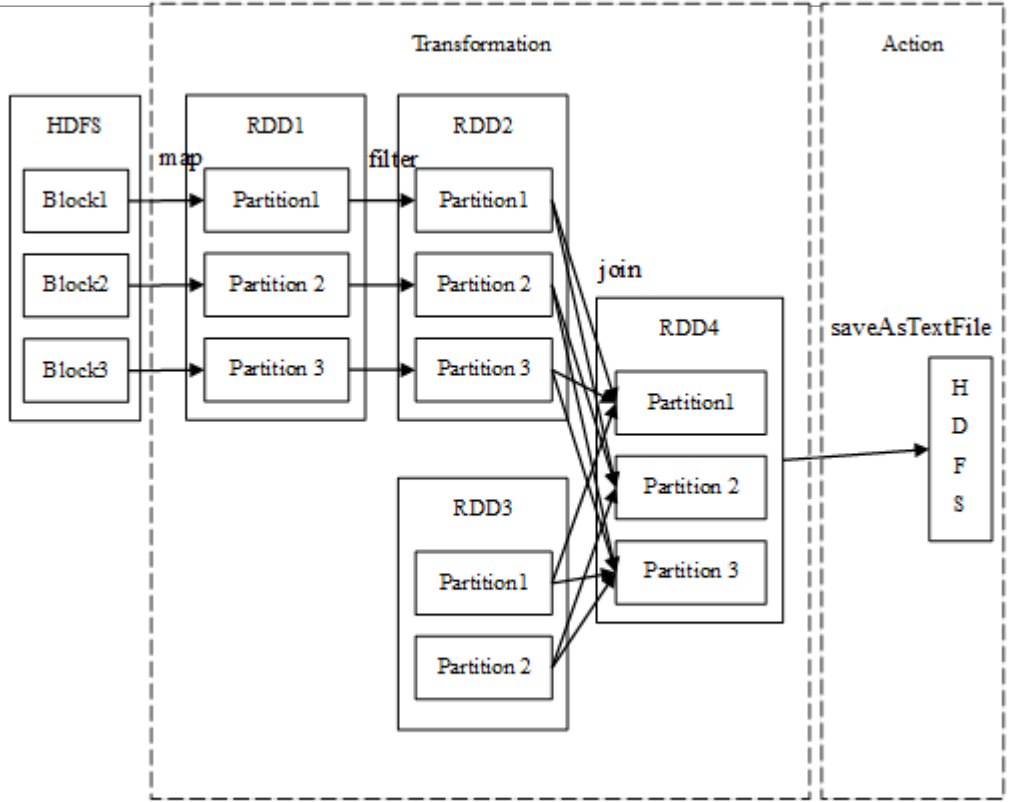


资料整理
火龙果软件
荐

5.3.2 Spark

4. 弹性分布式数据集RDD

图中共有4个不同的RDD，RDD内的方框代表分区。数据从HDFS输入Spark，生成Spark RDD，经过map、filter、join等多次Transformation算子操作，最终调用saveAsTextFileAction算子将结果集输出到HDFS，并以文件形式保存。



资料整理
火龙果软件
荐

感谢观看

资料整理
火龙果软件
荐