决胜大数据时代的技术全解手册

Broadview
www.broadview.com.cn

大数据技术全解

基础、设计、开发与实践

杨巨龙 编著

“三分虚拟化技术、七分分布式管理、十二分大数据”是企业在大数据时代下的信息化建设指导思想。本书遵循这一思想，重点阐述了企业如何利用开源的大数据相关产品来进行企业大数据系统的设计和开发。

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

内 容 简 介

“大数据”一词已成为当下最热门的词汇之一。人们热切期待大数据能给工作、学习和生活等方面带来前所未有的变化，而大数据的采集、存储、计算、分析挖掘、呈现和安全等技术正是实现这一变化的关键。本书的重点不在于对大数据技术的原理性介绍，而是从实战角度出发，系统地阐述如何基于 Hadoop 开源软件为客户设计、安装和开发一个大数据系统。本书还着重分享了具体实践中的一些案例，不仅对于从事大数据系统开发的 IT 技术人员具有珍贵的参考价值，而且对准备实施大数据系统项目的企业级客户也具有指导作用。

本书可以作为计算机软件专业的本科生和研究生的大数据技术教材，也可作为大数据技术开发培训、大数据系统咨询规划培训的教材，更是大数据领域内一线的需求分析人员、系统分析人员、高级开发人员和开发管理人员的必备参考书。

图书在版编目（CIP）数据

大数据技术全解：基础、设计、开发与实践 / 杨巨龙编著. —北京：电子工业出版社，2014.3
ISBN 978-7-121-22343-3

Ⅰ. ①大… Ⅱ. ①杨… Ⅲ. ①企业管理－数据管理 Ⅳ. ①F270.7

中国版本图书馆 CIP 数据核字（2014）第 006137 号

策划编辑：孙学瑛
责任编辑：李利健
印　　刷：三河市双峰印刷装订有限公司
装　　订：三河市双峰印刷装订有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：787×1092　1/16　印张：23.5　字数：613 千字
印　　次：2014 年 3 月第 1 次印刷
印　　数：4000 册　　定价：59.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。
质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。
服务热线：（010）88258888。

第 8 章

系统设计背景和目标

8.1 系统设计背景

这里讲的系统设计，指的是企业大数据系统设计。目前“大数据”的发展虽然还处于初级阶段，但大数据时代的序幕已经拉开。正所谓“竹外桃花三两枝，春江水暖鸭先知”，深处时代变革中的企业又一次面临大数据这一信息技术革命所带来的冲击，企业要么积极拥抱变化，提前做出变革；要么静观其变，择机而动。不管选择哪种方式，都是继互联网之后对企业的又一次智慧的考验。

当不知道现在在哪里，也不知道下一步到哪里去的时候，人们总是显得很绝望；当知道现在在哪里，但不知道下一步到哪里去的时候，人们总是显得很茫然；当知道现在在哪里，也提前定下了下一步到哪里去的目标，但所处的周边环境已经发生重大变化，面对该如何调整目标来适应环境的变化时，人们又总是会显得很无奈。

当前的企业信息化建设正处于这样一个周边环境已发生重大变化的时代中，如何调整下一步信息化建设目标以适应环境的变化，是摆在企业面前的一个急待解决的问题。21 世纪以来，信息化更加快速而深刻地改变着这个世界，大到全球经济社会发展的格局，小到企业的日常经营和每个百姓的工作生活，无不感受到信息化带来的冲击。智慧城市、智慧企业、物联网、下一代互联网、云计算、大数据等一个个新的理念、新的信息服务方式正在成为时下的热点。企业是顺应信息化时代的发展潮流，重新审视并及时调整商业模式，抓住这一信息化带来的重大机遇，还是驻足观望，或是不做改变依然自我，这确实是对企业智慧的考验。

企业信息化涉及企业的各个方面，是一项复杂的系统工程，一般要经历从初始到不断成熟的成长过程。美国管理信息系统专家诺兰总结企业信息化发展的规律，于 1980 年发表了著名的企业信息系统进化的阶段模型，即诺兰模型。诺兰认为，任何组织由手工信息系统向以计算机为基础的信息系统发展时，都存在着一条客观的发展道路和规律。

诺兰将企业信息系统的发展道路划分为 6 个阶段，即初始、扩展、控制、集成、数据管理和数据成熟。任何组织都必须从一个阶段发展到下一个阶段，不能跳跃式地发展。诺兰模型如图 8-1 所示。

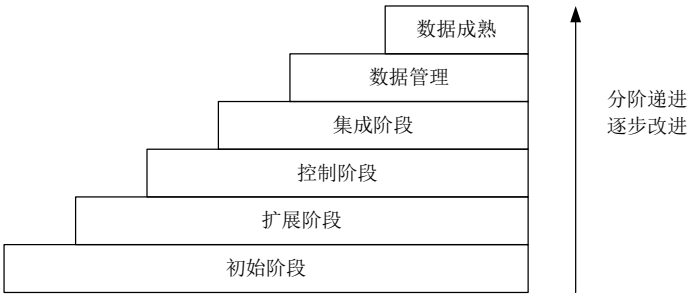

图 8-1　诺兰模型的 6 个阶段

对于数据管理阶段和数据成熟阶段，诺兰认为，在数据管理阶段，企业管理高层已经意识到了企业信息战略的重要性，并开始着手企业信息资源的统一规划；在数据成熟阶段，企业和数据是同步发展的，数据是企业的整体面貌的镜像，企业可以“以数据为镜”来做出发展决策。

尽管诺兰提出这一模型的时间是 20 世纪 80 年代，但在 30 多年后的今天，人们不难发现他预见的准确性。企业的信息化建设必然会走到以数据为中心的发展阶段，无论是否愿意，这条规律都是不可违背的。

航标灯：数据管理和数据成熟是企业信息化建设的高级阶段。

目前“大数据”的发展虽然还处于初级阶段，但大数据时代的序幕已经拉开，处于这个时代的个人和企业都必须顺应这一历史潮流，才不会被时代所遗弃。企业应当清晰地看到过去的信息化建设只能满足企业当下和过去面向局部的需要，而未来的信息化建设则要能够满足企业面向未来和全局的需要。

建设大数据系统是实现企业信息化跨越式发展的手段，也是企业顺应大数据时代做出转变的重要抓手。

8.2 系统设计目标

对照诺兰模型来分析国内企业信息化建设现状，不难发现大多数大中型企业经过几十年的信息化建设，处于第四和第五阶段之间，即系统集成和数据管理阶段之间。当然也有一批企业已率先进入了数据管理和数据成熟阶段，比如在电子信息领域、金融领域和电子商务领域中，就有不少先知先觉的企业率先采用大数据技术将企业的数据作为资产进行管理并加以应用了。

相对于这些先知先觉的企业，那些处于第四和第五阶段之间的企业也在积极探索和研究大数据技术，以便通过大数据技术的应用实现企业信息化建设的跨越式发展。

8.2.1 存在问题

这类处于第四和第五阶段之间的企业，大多数已经建设有面向生产管理的 ERP 系统、面向内部协同的 OA 系统、面向内部和外部交流的邮件系统和视频会议系统、面向线上经营的电子商务平台和面向员工知识传递的知识库系统等，并且对这些系统都进行了大集成。

大集成的特点包括两点：一是对外有一个统一的门户，用户通过单点登录使用各个系统的功能和访问各个系统的数据；二是后端数据进行了大集中，统一存储在一组存储服务器中，同时将具有共性的一些数据从各个应用系统中分离出来，实现应用系统间的共享，避免了系统的重复建设，如统一用户、统一权限、统一元数据等共性数据的提取。相对于过去的信息化建设，这已经前进了一大步，但还存在以下一些问题。

（1）数据虽然大集中了，但是只是数据存储位置的集中，而数据本身还是以应用系统为边界进行管理的。正是因为数据边界以应用系统来划分，如果需要对系统的数据间的关系进行统计分析，还需对应用系统进行进一步集成开发才能实现对数据的访问。

（2）大集成主要是定位在这些数据生产系统的集成，以及在集成后借助各应用系统自身对数据的统计分析功能将多个统计分析的数据放在门户界面上供用户自己做分析。也就是说，对于数据应用并得出有价值的分析结果这项工作，应用系统只是辅助，而不是主导，大量的

分析、加工工作还需要人工来处理。

（3）大集成的思想并没有站在“全业务、全系统、全信息”的角度去思考，还是一种粗放式的大集成思想。这就导致一些系统和数据并没有纳入到大集成的范围中，这就会造成数据缺失，比如企业员工个人电脑上的数据、互联网上用户对企业相关评价的数据等。而这些数据表面上对企业的生产不会产生直接影响，但有时会对企业产生致命的间接影响，比如核心员工的离职、互联网上对企业形象的负面宣传等。

航标灯：大集成、大集中为企业大数据建设奠定了基础。

8.2.2 设计目标

为了解决企业信息化大集成中还存在的问题，同时实现企业信息化建设由集成阶段向数据管理和成熟阶段的跨越式发展，依托大数据技术建设企业的大数据系统是一个最佳的选择。

大数据系统的设计是在正确的系统指导思想下所做出的，因此设计的目标要能体现这种思想。

1．指导思想

大数据系统的设计要遵循“全业务、全系统、全信息”和“数据是企业资产”这两个指导思想。

“全业务、全系统、全信息”中，全业务是指以企业为核心，将企业与政府主管部门、上下游厂商、内部组织和员工、企业产品的消费者和社会公众等的所有业务活动都纳入业务范畴，不是只包含企业内部人、财、物、产、供、销等这些主要业务活动，而是对这些业务活动进行进一步延伸。全系统是指所有围绕全业务建立的系统都是要关注的系统，这些系统包括企业内部系统和企业外部系统。上到决策系统，下到每个员工的个人终端系统都属于企业内部系统。企业外部系统既包括直接相关的系统，也包括间接相关的系统，如公共门户、社交网站等。全信息是指全系统所产生的数据信息，它可以是结构化、非结构和半结构化的。

“数据是企业资产”是指，在大数据时代，数据已经是继人、财、物之后的又一项重要资产。未来企业的竞争在于是否具有基于数据判断和基于数据决策的能力，而这一能力必须依赖于完整的、准确的、一致的数据。要想有这样的数据，企业的管理者就必须将企业在整个业务活动中的数据视为与企业的资金和物资同等重要。

航标灯：企业的大数据系统是以“全业务、全系统、全信息”和“数据是企业资产”作为核心指导思想的。

2．设计目标

大数据的系统设计目标包括总体设计目标、业务设计目标、作业设计目标和性能设计指标 4 个方面。

（1）总体设计目标

通过大数据系统的建设，实现企业基于应用系统为边界的数据位置大集中向无应用系统边界的数据内容大集中的转变。基于无应用系统边界的数据内容关联分析，不仅可以做出对企业过去历史轨迹的分析，还可以给出企业当前态势的描绘，更可以为企业做出未来发展趋势的预测，从而提升企业数据管理和企业数据应用的水平，使企业具备基于数据判断、基于

数据决策的能力。

航标灯：建立企业的大数据系统就是要使企业具备基于数据判断、基于数据决策的能力。

（2）业务设计目标

企业的业务大致可分为市场营销、危机管理、监测分析、生产管理、供应管理、财务管理、客户管理、宏观决策和内部管控等几个主要方面。大数据系统应能支撑这些业务在规划、计划、监测、检查、调整、统计、关联分析等方面的需要。

在监测分析方面，大数据系统应能对内部系统和外部系统的结构化、非结构、半结构数据内容进行采集并打破应用系统边界进行存储。在内部系统方面，应能细化到对员工终端的数据内容进行采集和存储。并且在采集和存储的基础上，提供跨应用系统的无数据类型约束的数据内容检索。

利用大数据的统计和机器学习等技术，可以为企业在市场分析、产品预测、企业规划、生产排程、总体态势等方面提供决策依据。利用大数据的聚类及特征项提取技术，可以找出客户流失、产品缺陷和仓储周转不畅等问题的根源，也可以给出客户和产品热点的分布状态。

航标灯：如何让企业的人、财、物、产、供、销等业务更加智慧，是企业的核心诉求。

（3）作业设计目标

作业目标主要是指系统的建设内容。为了支持总体目标和业务目标的实现，大数据系统建设内容应由能支持海量的、多类型的、低密度的、快速增长的数据的采集、存储、加工、利用和呈现等多个子系统构成，这些子系统将全面采用 Hadoop 的开源项目成果。子系统通过集成构成企业的大数据系统，为企业的大数据业务提供基础支撑环境。

大数据系统应采用“平台+应用”的设计思想。这些子系统集成在一起形成一个大数据平台，平台应能为大数据应用和开发提供强有力的支撑。大数据系统面向应用需提供 MapReduce 编程模型和多种脚本语言，以便开发人员借助这些模型和语言开发出基于大数据平台的业务应用程序，以支撑业务目标的实现。大数据系统还应提供各种接口，以实现现有系统对大数据系统的访问。

大数据系统应能高效地为各级各类管理者提供全面的统计和决策信息，并以表、图的方式进行可视化呈现。

（4）性能设计指标

大数据系统的性能设计指标主要包括系统性能、扩展性、可靠性、可用性、易用性和安全性等方面，如表 8-1 所示。

表 8-1　大数据系统的性能设计指标

指 标 类	指 标 项	指 标 要 求	指 标 说 明
系统性能	用户数	不受限制	大数据系统采用分布式系统，可以有效解决系统规模问题，对于用户数的快速增长，可以通过动态节点的扩充和数据块的复制来突破用户数的限制
	响应时间	增、删、改、查不受数据量大小的约束	大数据系统的数据增长对于整体运行效率影响不大，单机的运行效率始终保持一定水平。所以数据应用的时间自始至终会保持稳态
	存储容量	不受限制	通过数据节点的增加可以不断扩大数据的存储容量
扩展性		动态支持	大数据系统自身具有负载均衡的能力，随着应用规模的增

			大，可以通过增加集群节点，并快速复制来解决规模问题

续表

指 标 类	指 标 项	指 标 要 求	指 标 说 明
可靠性		高可靠	采用主从模式，从节点的失效可被感知并将任务分发到其他节点重新运行；主节点采用备份机制，失效后，可采用备份节点继续工作
安全性		高安全	数据块采用多副本技术，不会因为节点损坏而破坏数据文件的完整性。日志记录在分布式文件系统中，可依据日志进行失效节点的恢复
易用性		简单	MapReduce 分布式编程模式，使开发人员只需专注业务，对于代码的分发、数据分配和调用均由系统来完成。同时提供脚本语言，可用简单的语句实现对海量数据的操作
可用性		方便	对于应用用户而言，大数据的处理是透明的，并且处理结果以表格、图形的方式呈现

第 9 章

系统架构设计

9.1 逻辑架构设计

企业大数据系统是由多个子系统集成构成的系统，这些子系统将全面采用 Hadoop 的开源项目成果，比如非结构化存储采用 HDFS 分布式文件系统，结构化数据存储采用 HBase 分布式数据库系统。每个子系统都采用主从模式，即由一个主节点和多个从节点构成。大数据系统逻辑架构是一个多层次的体系架构，采用“分布式平台+分析式应用”的模式进行设计。

航标灯：分布式平台+分析式应用是企业大数据系统架构的关键。

9.1.1 系统逻辑架构

系统逻辑架构设计如图 9-1 所示。

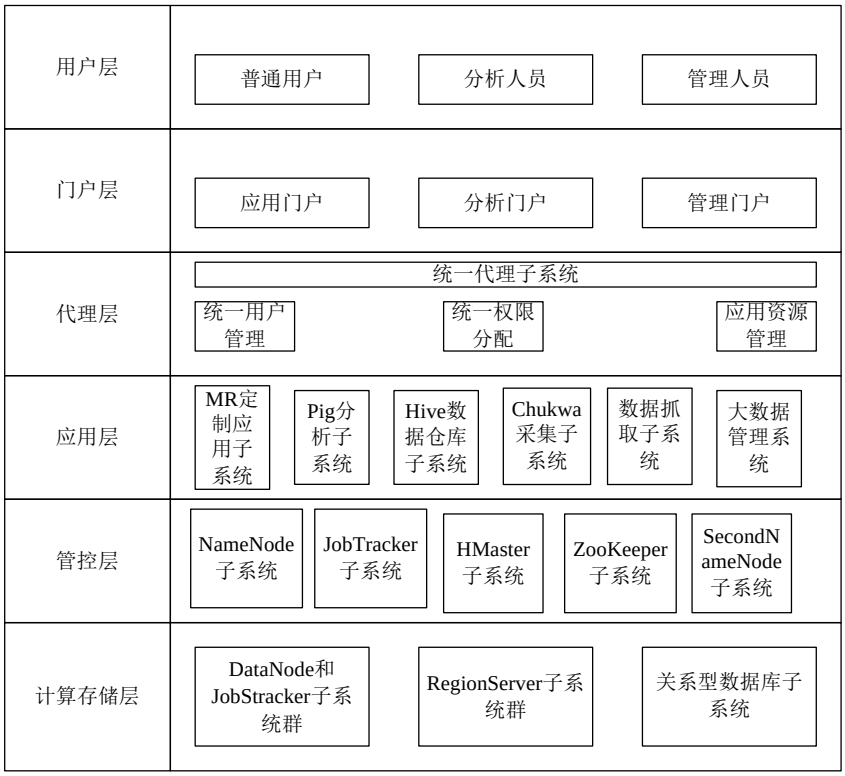

图 9-1　大数据系统逻辑架构

企业大数据系统的逻辑架构由用户层、门户层、代理层、应用层、管控层、计算存储层共 6 层组成。管控层和计算存储层构成了大数据的平台层，应用层的主要功能是支撑大数据

业务应用的开发和各子节点的后台管理。

（1）用户层

用户层是所有访问大数据系统的用户集合，根据用户职能的不同，大数据系统的用户可划分为 3 类：普通人员、分析人员和系统管理人员。

（2）门户层

门户层是用户访问大数据系统的统一入口，用户经应用层管理系统统一认证后，门户子系统按照用户身份的不同分别提供相应的个性化门户，用户只能查阅和操作系统授权的数据信息和应用功能。

（3）代理层

代理层是门户层和应用层各应用系统之间的统一代理，代理层由统一代理子系统构成。统一代理子系统由统一用户管理、统一应用资源管理、统一权限管理三个核心模块构成。代理层向门户层开放出 WebService 服务，供门户层的系统调用。当 WebService 服务接收到门户层发来的请求信息时，会向应用层的应用系统发出调用请求。

（4）应用层

应用层由数据应用类、数据采集类、后台管理类三类子系统构成。应用类子系统包括 MR 定制子系统、Pig 分析子系统和 Hive 数据仓库子系统，主要为普通用户和分析人员提供大数据分析和挖掘服务。数据采集类子系统包括 Chukwa 采集子系统和数据抓取子系统，主要是从企业现有系统、大数据应用系统和外部应用系统中采集数据并存储在大数据系统中。后台管理类子系统主要是对大数据系统的用户信息、节点设备信息和节点运行状态进行管理。

（5）管控层

管控层主要由 NameNode 子系统、JobTracker 子系统、HBase 的 Master 子系统、ZooKeeper 子系统和 SecondNameNode 子系统构成。

NameNode 子系统是 HDFS 系统的重要组成部分，主要提供目录和文件的元数据管理，是分布式文件读写的统一入口。SecondNameNode 子系统是其备份子系统。

JobTracker 子系统是分布式作业系统的重要组成部分，是 MapReduce 分布式程序的服务器端，主要是对作业的任务进行调度分配和运行状态的管理。

HBase 的 Master 子系统是 HBase 系统的主要组成部分，它主要负责引导初始安装，为 Region 域数据分别注册到 HBase 集群中的域服务器 HRegionServer，监控 HRegionServer 的运行，并在出现故障时进行恢复。

ZooKeeper 子系统主要是为 HMaster 和 HRegionServer 提供共享信息的统一协同服务。

（6）计算存储层

计算存储层由 HDFS 系统的 DataNode 和分布式作业系统的 TaskTracker 组成的子系统、HRegionServer 子系统和关系型数据库管理子系统构成。

HDFS 系统的 DataNode 和分布式作业系统的 TaskTracker 组成的子系统共同部署在一个机器节点上，为分布式文件提供数据块存储和数据块计算服务。

HRegionServer 子系统主要是为客户端提供分布式数据库系统的访问服务。

关系型数据库子系统主要是存储应用层后台管理系统和 Hive 数据仓库系统提供用户信息、机器设备信息、元数据、运行状态监测信息的存储服务。

9.1.2 系统运行逻辑

用户从门户登录到大数据系统以后，在用户交互界面上选择相应的系统功能，经应用系统对后台的分布式文件系统和分布式数据库系统进行访问，以满足用户的应用需求。系统的运行逻辑如图 9-2 所示。

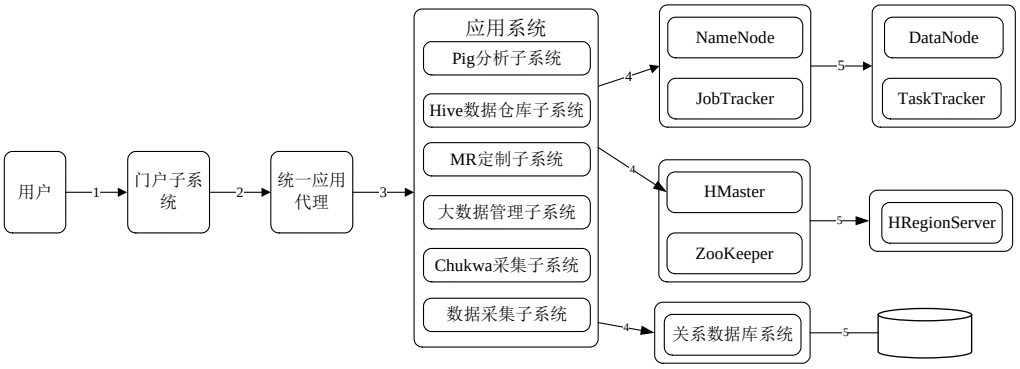

图 9-2　大数据系统运行逻辑图

图 9-2 中共有 3 条系统运行主线：第一条是对分布式文件读写和应用的运行主线，共有 5 步；第二条是对分布式数据库的读写和应用的运行主线，共有 5 步；第三条是对关系型数据进行读写和应用的运行主线，共有 4 步。其中三条主线的前 3 步都是一样的，只是在 4、5 步有所不同。下面对具体步骤内容进行说明。

（1）用户输入门户子系统的网址，进入门户子系统的登录主页面，输入用户名和密码。

（2）门户子系统将登录信息提交给统一应用代理，统一应用代理调用统一用户登录模块，检查用户的账户信息、密码，并根据用户角色返回个性化应用页面；用户选择相应的应用功能再次提交统一应用代理。

（3）根据统一应用代理应用类型的不同，调用不同的应用子系统。应用类型分为三类，分别是面向分布式文件的应用、面向分布式数据库的应用和面向传统关系型数据库的应用。面向分布式文件的应用会根据应用的细分类型调用 Pig 分析子系统、Hive 数据仓库子系统或 MR 定制子系统中的一种或几种向 HDFS 系统发起访问请求。面向分布式数据库的应用会调用 MR 定制子系统向 HBase 系统发起访问请求。面向关系型数据库的应用会调用大数据管理子系统、Hive 数据仓库向关系数据库系统发起访问请求。

（4）面向分布式文件系统的访问请求会由 NameNode 和 JobTracker 处理，向 DataNode 数据节点出指令。面向分布式数据库系统的访问请求会先到达 ZooKeeper，然后返回元数据表地址，以便应用系统定位 HRegionServer。面向关系型数据库系统的访问请求，会由关系型数据库系统直接处理，并将处理结果返回给应用系统。

（5）当 DataNode 和 TaskTrakcer 节点收到任务指令时，依据指令进行文件的处理。当 HRegionServer 收到应用发来的请求时，会对相应的 Region 进行处理。

航标灯：传统的应用只需调用分布式应用的接口，就可以变成基于大数据系统的应用。

9.2 功能架构设计

企业大数据系统由 6 层共 12 个系统构成。这 12 个系统按层次可分为：门户层 1 个系统，即门户系统；代理层 1 个系统，即统一应用代理系统；应用层 6 个系统，即 Pig 分析系统、Hive 数据仓库系统、MR 定制系统、大数据管理系统、Chukwa 采集系统和数据采集系统；管控层和计算存储层共有 4 个系统，即 HDFS 系统、HBase 系统、关系数据库系统、ZooKeeper 系统。

门户系统、统一应用代理系统、关系型数据库系统和数据采集系统在过去的信息化建设中都是现成的系统，可以直接部署应用，所以在功能架构设计中不再进行说明。MR 定制系统、HBase 系统、HDFS 系统在技术篇中已有说明，所以在功能架构设计中也不再说明。这里重点放在大数据管理系统、Pig 分析系统、Hive 数据仓库系统、Chukwa 采集系统和 ZooKeeper 系统这 5 个系统的功能架构设计上。

9.2.1 大数据管理系统的功能

大数据管理系统是管理员使用的一个系统。通过大数据管理系统可以实现对资源的管理，如集群命名、HBase 分布式数据库、HDFS 文件系统、MR 程序资源、节点资源、安装资源、用户状态等。通过大数据管理系统可以实现对关系的管理，如用户与分布式文件目录的关系、MR 程序资源与节点资源的关系、安装资源与节点资源的关系等。通过大数据管理系统可以实现对行为的管理，如 DataNode 的节点远程安装、节点的启动和停止和系统的运行监控。

航标灯：大数据管理系统是大数据系统的关键系统之一。

大数据管理系统的功能结构如图 9-3 所示。大数据管理系统由资源管理类、关系管理类、行为管理类共 13 个功能模块组成。

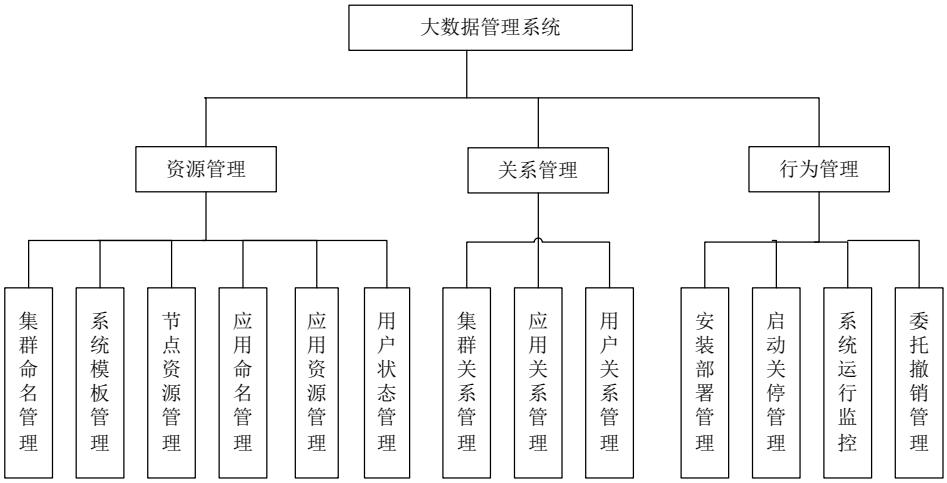

图 9-3 大数据管理系统的功能结构

航标灯：资源管理、关系管理、行为管理是任何一个信息系统都必须具备的三要素。

1．资源管理类的功能

资源管理类共有 6 个，分别是集群命名管理、系统模板管理、节点资源管理、应用命名

管理、应用资源管理、用户状态管理。

集群命名管理主要是创建集群、查询集群和维护集群。集群的类型包括 HDFS 集群、HBase 集群、ZooKeeper 集群和 JobTracker 集群，每一类集群都可以创建多个，每一个集群都需要有名称、类型、最多的节点数量和当前节点数量（由集群关系管理时生成）等信息。创建集群的工作本质上是对大数据系统做规划。当集群较多时，可以通过集群名和集群类型进行查询。维护集群是指对集群和集群下的节点进行删除，以及对集群状态进行查看。删除节点或集群时要先停止节点或集群（这些功能操作时是调用关系管理的相关功能实现的）。集群状态的查看是对集群整体和各节点情况的查看（查看功能是调用行为管理的相关功能实现的）。

系统模板管理主要是对模板进行上传、删除、下载等操作。系统模板包括 HDFS 集群、HBase 集群、ZooKeeper 集群和 JobTracker 集群 4 种。在一个节点上配置好一个集群模板后，将其打包成 tar，然后由系统模板的上传功能将其存储到管理系统中。

节点资源管理主要是对节点进行增加、删除和维护操作。当有一个新的节点加入到大数据系统后，管理员需要记录该节点的信息。记录信息的内容包括节点名、节点类型、节点 IP、节点配置等。节点类型包括 NameNode、DataNode、JobTracker、HMaster、Pig、MR 等。

应用命名管理主要是对应用类型进行创建、查询和维护操作。应用命名时的类型包括 MR 类型、Pig 类型、Hive 类型等。每一种类型的应用可以有多个。

应用资源管理主要是对应用程序进行上传、删除和下载等操作。应用资源包括 MR 程序、Pig 程序包、Hive 程序包等。每个应用资源需要给出名称、类型、版本号、作用等信息。

用户状态管理主要是对用户信息查询和用户状态查询的管理。用户信息是由于统一代理系统的统一用户管理所产生的信息。用户状态查询主要是对用户相关的文件目录、文件空间、文件类型、数据库表等关系信息的查看。

2．关系管理类的功能

关系管理类共有 3 个，分别是集群关系管理、应用关系管理和用户关系管理。

集群关系管理主要是用于建立节点、系统模板和集群间的关系。首先是建立集群与节点间的关系。集群和节点是一对多的关系，集群的类型将会决定节点类型的选择，如 HDFS 集群，其节点类型就只能是 NameNode、JobTracker、DataNode 和 TaskTracker。集群中节点的数量，将受集群预先定义的总数的限制。节点和集群可以建立关系，也可以解除关系，解除后的节点将放回资源池，成了自由节点。其次是建立节点和系统模板资源的关系。节点的类型将决定系统模板资源的类型，比如 NameNode 节点类型所对应的只能是 HDFS 系统资源模板。集群关系建立完成后，可以调用行为管理类的安装部署功能，将系统模板复制到节点机器中。

应用关系管理主要是用于建立节点、程序资源和应用间的关系。首先，建立应用与节点间的关系，应用和节点是一对多的关系。应用和节点没有类型约束关系，只有数量约束关系，比如 Pig 应用定义为 1 个节点，则节点与应用的关系数也只能为 1。其次，建立程序资源与节点的关系。由于节点预先和应用建立了关系，应用类型将决定应用程序的类型，比如 Pig 应用的节点只能与 Pig 应用程序资源建立关系。应用关系建立完成后，可以调用行为管理类的安装部署功能，将应用程序复制到节点机器中。

用户关系管理主要是用于建立、解除、调整和集群资源间的关系，包括用户与文件目录及文件、用户与文件空间、用户与数据库表、用户和程序资源之间的关系等。用户与文件目

录及文件建立关系，是指定用户可以访问该目录或者某个目标下的文件。用户与文件空间关系的建立，是对用户的文件空间存储空量的设置。用户与程序资源关系的建立，是指明用户可以调用哪些应用资源。没有建立关系的程序资源，用户是不能调用的。

3．行为管理类的功能

行为管理类共有 4 个，分别是安装部署管理、启动停止管理、委托撤销管理、运行监控管理。

安装部署管理是对集群关系管理和应用关系管理创建的关系生成安装部署任务列表，管理员针对安装部署任务列表，通过 SSH 方式将程序资源和系统模板资源复制到相应的节点机器中。

启动停止管理是对集群关系管理和应用关系管理创建的关系生成启动和停止的任务列表。任务有两种类型，一种是面向集群整体的，一种是针对集群中每个节点的，也就是说可以针对整体任务进行启动和停止操作，也可以针对单个任务进行启动和停止操作。

委托撤销管理是对 HDFS 集群上的节点进行添加和删除。添加一个新节点，只需将节点添加到共享文件 include 中和 slaves 中，然后用命令刷新集群的名称节点。删除一个节点只需要将撤销节点的地址加入到 exclude 文件中，停止对应节点的任务，然后刷新节点。

运行监控管理包括查看日志、监控、维护等功能。查看日志时要分别调用各集群的主页面进行查看。监控是对集群运行的全过程及其各节点上的运行状态进行监控。维护包括例行备份和系统升级操作。

9.2.2 ZooKeeper 系统的功能

ZooKeeper 通过一种和文件系统很像的层级命名空间来让分布式进程互相协同工作。这些命名空间由一系列数据寄存器组成，我们也叫这些数据寄存器为 znodes。这些 znodes 有点像是文件系统中的文件和文件夹，和文件系统不一样的是，文件系统的文件是存储在存储区上的，而 ZooKeeper 的数据是存储在内存上的。这就意味着 ZooKeeper 具有高吞吐和低延迟的特点。

ZooKeeper 实现了高性能、高可靠性和有序的访问。高性能保证了 ZooKeeper 能应用在大型的分布式系统上。高可靠性保证它不会由于单一节点的故障而造成任何问题。有序的访问能保证客户端可以实现较为复杂的同步操作。

航标灯：ZooKeeper 系统为多个系统间提供协调服务。

ZooKeeper 提供的功能包括：组服务、名字服务、配置维护、锁服务等，其功能结构图如图 9-4 所示。

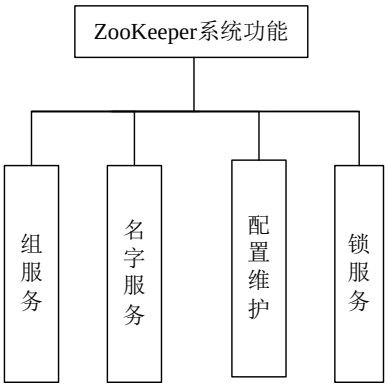

图 9-4　ZooKeeper 系统功能结构

组服务是一个主动的分布式数据结构服务。分布式数据结构按层次存储信息，这组信息是以整体为单位存储在多个服务器中的。组服务提供的服务包括创建组、加入组成员、列出组成员和删除组成员。

名字服务是指组成员中存储的是名字和数据。名字服务主要提供名字写入和读出两种服务。

配置维护是一种特殊的名字服务，主要用来存储 configration 的配置信息。其中，组成员名正好就是某个节点主机的名称，如 NameNode 的名称，而其存储的数据就是键值对信息。

锁同步是解决一组进程间的互斥问题，也就是在某一时刻只有一个进程可以持有锁。其解决办法是选择一个组成员作为锁，当 *N* 个进程要获取此锁时，可以在组成员创建多个有序列号的同级组成员，然后依据最小顺序号给进程返回锁信息，其他的则处理排队中。

9.2.3　Chukwa 采集系统的功能

Chukwa 是一个数据采集系统，它可以将各种类型的数据收集成适合 Hadoop 处理的文件保存在 HDFS 中，然后由 MapReduce 操作。Chukwa 本身也提供了很多内置的功能，可以帮助用户进行数据的收集和整理。其系统功能如图 9-5 所示。

代理是以进程的方式运行在集群上的每一个节点上的，负责采集原始数据，并将数据发送给收集器。代理中具体负责数据采集工作的功能模块是适配器，由它具体执行数据的采集。

收集器负责收集代理传递的数据，数据先存在内存中，当达到一定阈值后，写入 HDFS 分布式文件系统中。

MR 作业定时启动，对存储在 HDFS 上的文件进行分类、排序、去重及重并。

HICC 是 Chukwa 的管理中心，它将 MR 作业的处理结果以网页的形式呈现出来。

9.2.4　Pig 系统功能

Pig 是一个基于 Hadoop 的大规模数据分析平台，它提供的 SQL-LIKE 语言叫 Pig Latin，该语言的编译器会把类 SQL 的数据分析请求转换为一系列经过优化处理的 MapReduce 运算。Pig 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。其系统功能如图 9-6 所示。

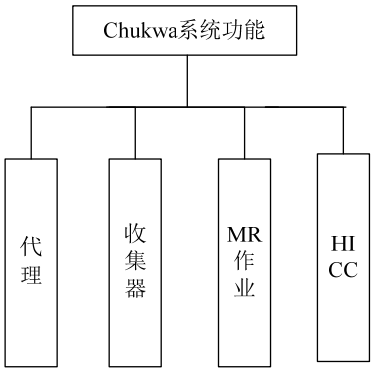

图 9-5　Chukwa 系统功能结构

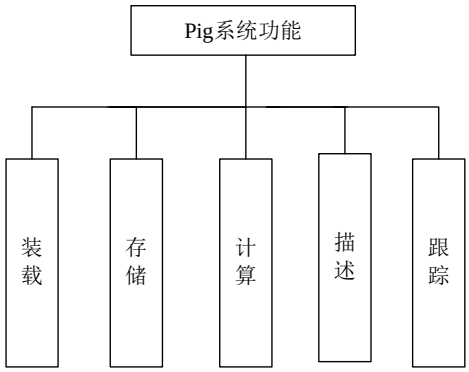

图 9-6　Pig 系统功能结构

- 装载（Load）是将一个有格式的文件加载到 Pig 系统中。
- 存储（Store）是将一个有格式的数据存储从 Pig 系统中存到一个目录中。
- Pig 中的计算包括求平均值、求最大、求最小、记录数计算、字段数计算、字段连接、字段比较等算法。在处理这些计算时可以采用循环和过涉两种方式处理。
- 描述包括结构关系描述（DESCRIBE）、命令的作用描述（EXPLAIN）和结果输出到屏幕（DUMP）三种描述结构、作用和数据结果的方式。
- 跟踪（ILLUSTRATE）是对 Pig 的某个计算的处理过程加以显示。

航标灯：假如觉得 MapReduce 编程太麻烦，可以采用 Pig 的脚本语言进行快速编程来处理大数据。

9.2.5　Hive 系统功能

Hive 是建立在 Hadoop 上的数据仓库基础构架。它提供了一系列的工具，可以用来进行数据提取转化加载（ETL），这是一种可以存储、查询和分析存储在 Hadoop 中的大规模数据的机制。Hive 定义了简单的类 SQL 查询语言，称为 HQL，它允许熟悉 SQL 的用户查询数据。同时，这种语言也允许熟悉 MapReduce 的开发者开发自定义的 mapper 和 reducer 来处理内建的 mapper 和 reducer 无法完成的复杂的分析工作。

Hive 没有专门的数据格式。Hive 可以很好地工作在 Thrift 上，控制分隔符，也允许用户指定数据格式。Hive 的系统功能如图 9-7 所示。

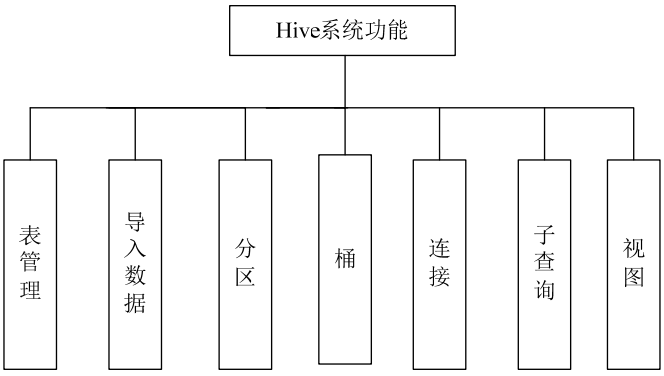

图 9-7　Hive 系统功能结构

- 表管理包括表的创建、修改和删除。
- 导入数据可以将数据从本地文件系统和 HDFS 文件系统导入到表中。
- 分区是对表的部分列进行拆分。
- 桶是指按行分开组织特定字段。用户可以把表或分区组织成桶。每个桶对应一个 Reduce 操作。
- 连接是将两个表中在共同数据项上相互匹配的那些行合并起来，连接方式分为内连接、左外连接、右外连接、全外连接和半连接 5 种。
- 子查询是指查询的来源可以是一个表的查询结果，Hive 对子查询的支持非常有限。

航标灯：Hive 可以让开发人员通过类 SQL 语句来轻松驾驭大数据。

9.3 数据架构设计

企业大数据系统中的数据架构设计必须遵循“全业务、全系统、全信息”的指导思想，在数据存储方式上需要将传统数据中心的以数据存储位置大集中转变为以企业主题为中心的数据内存存储大集中。也就是说，大数据的存储不是简单地把传统的应用系统所产生的数据整体搬迁到分布式数据环境下，而是要将其按主题和类型进行搬迁。

航标灯：大数据系统建设是对企业现有数据资产的一次盘点。

9.3.1 数据总体架构

一级主题可以按照主体、物质、能量、信息、对象进行划分。主体可以分为企业、员工、供应商、上级政府、行业协会等二级主题；对象可分为客户、公众等二级主题；物质可划分为产品、部件、零件等二级主题；能量可分为规划、计划、生产、采购、供应、销售、统计、监控、日志等二级主题；信息可分为单证、文件、报表、参考等二级主题。总之，主题是将系统要素进行拆分后按作用归类的命名。

主题划分完后，按照主题在分布文件中创建目录，同时在分布式数据库中按照主题创建数据库表。由于企业大数据系统还有一些关系型数据库系统，如 Hive、Chukwa、大数据管理系统等，所以还需为这些系统的数据存储和访问创建相应的库表。

大数据系统的数据来源可以是企业现有的信息系统所产生的数据、企业各用户终端产生的数据、外部合作单位系统产生的数据，也可以是互联网各类网站产生的与企业相关的数据。

基于上述分析，企业大数据系统的数据总体架构是一个两层次、多结构的架构。数据总体架构如图 9-8 所示。

图 9-8　大数据系统数据总体架构

大数据系统的数据总体架构由大数据层和数据源层两层构成，数据类型可以分为结构化、非结构化、半结构化三种。

（1）大数据层：由分布式文件数据、分布式数据库数据、关系型数据库三个部分构成，它们是并列和协同的关系。分布式文件和数据库数据是按主题来划分，并进行目录和表定义的。关系型数据库主要是用于支撑大数据应用系统的用户数据和元数据的存储和应用。

（2）数据源层：大数据层中的数据很大一部分是依赖于数据源层的，由大数据的采集系统将数据采集加工后存储在数据源层的存储设备中。数据源层由企业现有的信息系统所产生的数据、企业各用户终端产生的数据、外部合作单位系统产生的数据、互联网各类网站产生的与企业相关的数据等构成。

航标灯：任何非结构化的都是表面现象，其本质还是结构化的。

9.3.2　分布式文件数据结构

分布式文件的数据结构由目录和文件构成。一级目录是企业的所有二级主题的汇总。二级目录由一级主题名和序号构成，比如一级目录中的 employee，即员工数据的存放目录，二级目录为 emplee-001001，其中 001001 是员工在企业中的编号。三级目录可按文件类型或作用类型进行定义，如相片、音像、文件等。三级目录以下由目录使用者具体定义。图 9-9 所示是一个目录结构示例，读者可按此方式进行企业自身的目录结构定义。

如图 9-9 所示，企业的非结构数据（也可以是结构数据）按照主题进行存储。其中企业所有员工的终端数据都存放在员工这个目录，对于一个企业而言，员工的终端里

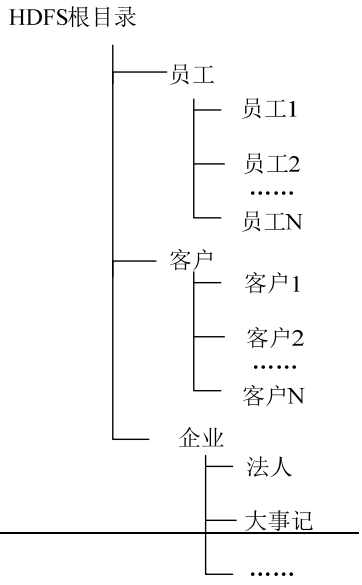

图 9-9　分布式文件数据目录结构示例

存储了大量能反映出员工与企业真实关系的数据。

航标灯：归类存放数据是根本之道。

9.3.3 分布式数据库数据结构

分布式数据库与分布式文件不同。在分布式文件存储时，主要是对目录按主题进行规划后，就可以将原有的文件原封不动地存入到 HDFS 系统中了。而分布式数据库需要对原有的库表进行改造才能存入，当然改动并不大，只是将那些外键是其他某张表的主键的表作为一个主题下的一组表进行集中存放。在技术篇已经对其有描述，读者可参看技术篇中的库表设计一节。表 9-1 是一个企业的分布式数据库的数据结构示例。

表 9-1 企业分布式数据库数据结构示例

表	行 键	列 族	说 明
员工	员工编号	个人信息 家庭信息 奖惩信息 升降职信息 ……	以员工为主题，将与员工主题相关的多个基本表作为分布式员工表的列族，如个人信息、家庭信息等。行键为员工编号
企业	企业代码	基本信息 法人信息 投资人信息 ……	以企业为主题，将与企业主题相关的多个基本表作为分布式企业表的列族，如基本信息、法人信息等。行键为企业代码

续表

表	行 键	列 族	说 明
产品	产品编号	基础信息 产品功能 安装说明 使用说明 ……	以产品为主题，将与产品主题相关的多个基本表作为分布式产品表的列族，如基础信息、产品功能等。行键为产品编码

9.3.4 关系型数据库数据构成

企业大数据系统中的一些应用系统，其结构化数据量非常小，没有必要将其存在分布式数据库表中，所以将其存在关系型数据库中。这些应用系统包括统一代理系统、大数据管理系统、Hive 数据仓库系统和 Chukwa 采集系统。这些应用系统存储的数据都是以关系表的形式存在的，这些关系表的构成如图 9-10 所示。

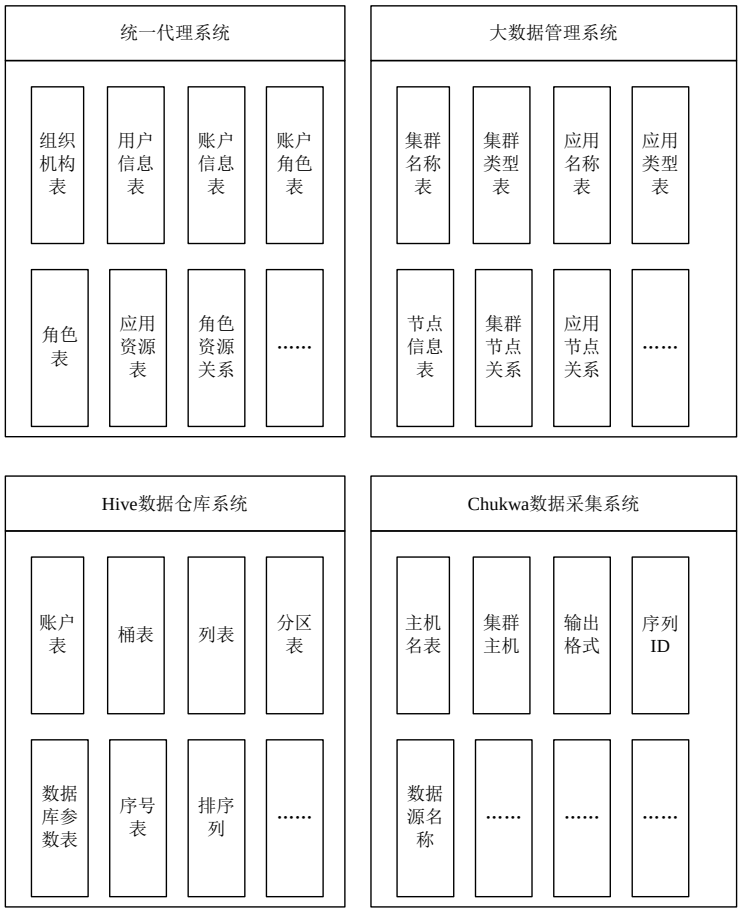

图 9-10　关系数据库数据构成

大数据关系型数据库数据由 4 部分构成，这 4 部分之间是并列关系，分别存在于不同的逻辑库中，每个应用系统由多张关系表构成。

第 10 章 运行架构设计

运行架构的设计着重考虑的是企业大数据系统运行期的质量属性，比如性能、可伸缩性和持续可用性。大规模用户并发和海量数据处理是企业大数据系统在运行架构设计时重点要解决的问题。

航标灯：大规模用户并发和海量数据处理是企业大数据系统在运行架构设计时重点要解决的问题。

大规模用户并发是采用集群的方式来解决的，集群技术在传统管理信息系统中的应用已经非常成熟。海量数据处理是采用分布式方式来解决的，Hadoop 是一个成熟的、被广泛应用的分布式系统，而本书介绍的企业大数据系统正是建立在 Hadoop 系统上。企业大数据系统的运行架构正是采用这两种方式来构建的，如图 10-1 所示。

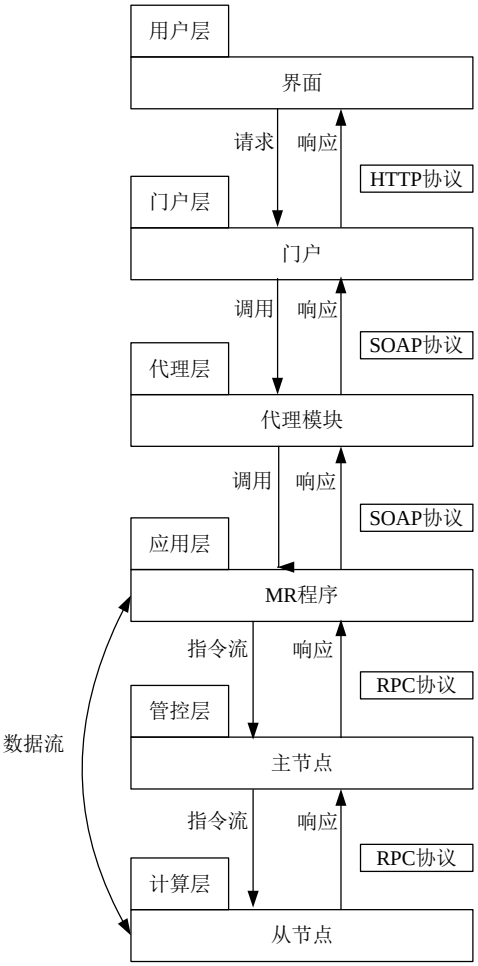

图 10-1　企业大数据系统运行架构

门户层、代理层、应用层采用集群方式来解决大规模、高并发的用户访问。门户层和代理层之间、代理层和应用层之间均采用 SOAP 协议进行调用请求和信息返回。

管控层由多个主节点构成，这些主节点都是单进程的，它们的负载较轻，这是因为应用层与分布式系统之间的交互是采用指令流和数据流分离的方式来处理的。应用层和计算层的多个节点直接进行数据的交互。应用层、管控层、计算层之间均采用 RPC 协议进行交互。

航标灯：企业大数据系统包括 HTTP、SOAP、RPC 三种协议。

10.1　物理架构设计

企业大数据系统的各层次系统最终要部署到主机节点中，这些节点通过网络连接为一个整体，为企业的大数据应用提供物理支撑。企业大数据系统共由 6 个逻辑层次构成，多个逻辑层可以映射到一个物理节点上，也可以映射到多个物理节点上。

映射时需要考虑三个方面的问题：一是是否容易识别，即通过物理节点的 IP 地址就能知道这个节点的作用域，通过多个物理节点的 IP 地址能知道这些节点是否是一个集群的；二是是否足够集约，对于负载轻的软件系统，如果每一个软件系统单独部署在一个物理节点，会造成物理节点的浪费；三是是否能够同构，对于物理节点最好能够统一配置，不仅便于统一管理，而且还可以实现重用，只需一次配置，多个物理节点同构复制，就可以实现动态扩展。

Google 公司和 Facebook 公司早期都是“草根”企业，它们都不得不选择使用大量的廉价商用硬件来搭建自己的分布式系统。基于廉价商用硬件搭建的分布式系统在运行效率、可靠性、可扩展性方面都被证明能够经得起大规模、高并发、海量数据应用的检验。

进行企业大数据系统的物理架构设计时要采用这种方式，即将所有的系统都部署在一个由同配置的、多台廉价商用硬件组成的物理机集群中，同时兼顾集约性，将负载轻的软件系统集中部署在有限的物理节点上。

10.1 节将从网络拓扑和软/硬件选型两个方面来进行企业大数据系统的物理架构设计的描述。

10.1.1　网络拓扑

网络设计是企业大数据系统的物理架构设计的重要组成部分之一。它主要是通过网络域的划分来体现出企业大数据系统的不同层次及不同应用，以便网络管理人员对网络上的多个物理节进行分域管理。其网络拓扑图如图 10-2 所示。

从图 10-2 可以看出，企业大数据系统共分为用户域、应用系统域、安全管理域、分布式文件域、分布式数据库域、分布式协同域和关系数据库域 7 个域。大数据是企业最重要的资产，对企业至关重要，故用一个交换机将分布式文件、分布式数据库、关系型数据库三个域与其他的域隔开。

航标灯：在物理架构设计上要坚持分区分域的原则。

10.1.2　软/硬件选型

企业大数据系统在软/硬件选型方面将遵循两个原则：一是硬件上将遵循“同品牌、同型

号、同配置”的原则，不因应用系统或分布式系统的不同而不同；二是软件上将遵循“同操作系统、同 JVM 虚拟机、同系统账户体系”的原则。

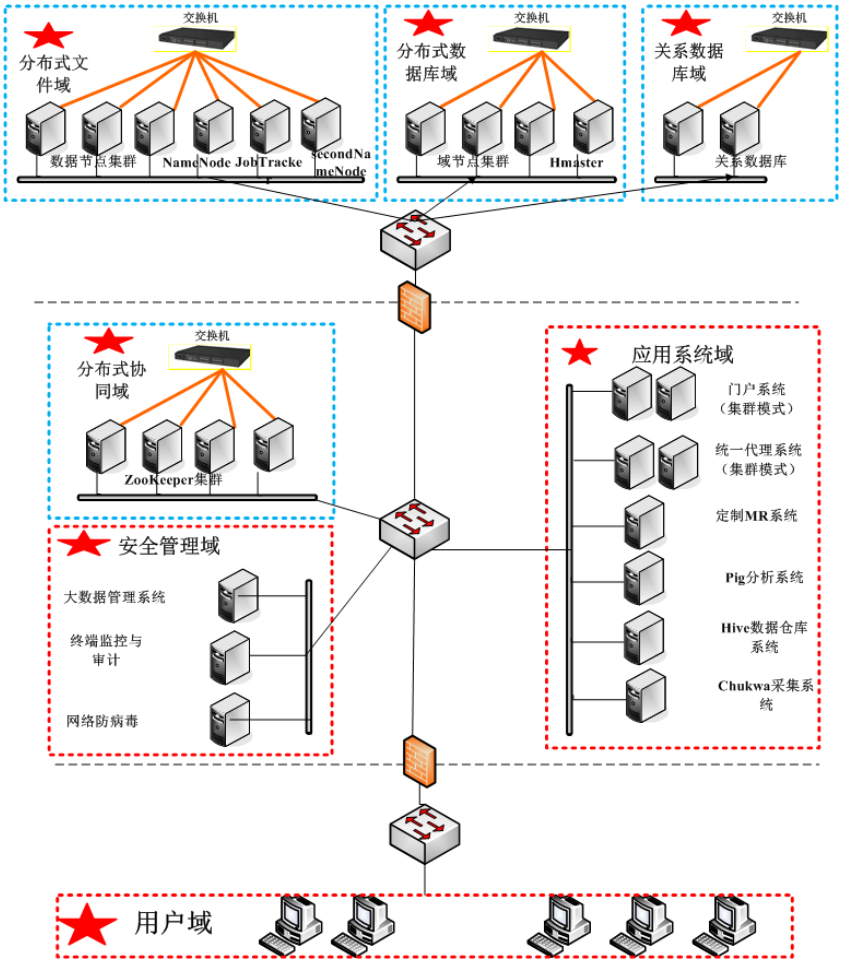

图 10-2　企业大数据系统网络拓扑图

推荐硬件选型如表 10-1 所示。

表 10-1　推荐硬件选型列表

品　　牌	设 备 名	配　　置	备　　注
用户自定	节点机	CPU：2 个 4 核，主频为 2.4GHz 以上 内存：16GB 硬盘：500GB 网卡：千兆	大数据系统的所有节点机器的配置基本相同

每个节点的基础软件选型如表 10-2 所示。

表 10-2　节点基础软件选型列表

操 作 系 统	JDK	SSH	目　　录
Linux	Sun 的 JDK1.6 及以上	都需配置	统一定义安装目录

分布式系统的选型见本书的安装篇部分。

10.2 集成架构设计

企业大数据系统本身由多个系统集成而成，每个系统都提供了多种协议和接口，以便企业大数据系统的内部系统间集成（如 Hive 系统调用 HDFS 系统的接口进行集成）和外部系统与大数据系统的集成（如 ERP 系统调用 Thrift 接口与 HDFS 进行集成）。

企业大数据系统的集成可以分为总体集成和专项集成。总体集成是指企业大数据系统的各组成系统间的集成，通过总体集成可以构成高效、可靠、安全运行的企业大数据系统。当企业大数据系统之外的某个应用系统（如 ERP 系统）或大数据系统之内的某个应用系统（如 MR 系统），只想与 HDFS 系统、HBase 系统、Pig 系统等进行集成，那么可通过调用这些系统开放的接口来实现，这种集成方式就是专项集成。

航标灯：企业大数据系统的集成可以分为总体集成和专项集成。

从系统的集成层次来看，又可以将企业大数据系统的集成分为界面集成、功能集成、数据集成等多种集成方式。集成层次图如图 10-3 所示。

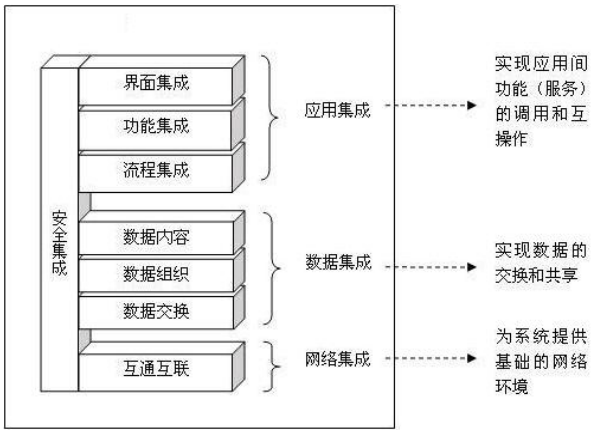

图 10-3　企业大数据系统集成层次图

图 10-3 给出的是一个完整的多层次的集成方式，而企业大数据系统在集成方式上主要包括界面集成、功能集成和数据集成三种。

10.2.1 总体集成设计

总体集成是指企业大数据系统的各组成系统间的集成，通过总体集成可以构成高效、可靠、安全运行的企业大数据系统。总体集成工作主要包括门户界面集成、应用功能集成和主从节点集成。

航标灯：总体集成工作主要包括门户界面集成、应用功能集成和主从节点集成。

1．门户界面集成

门户界面集成的方法是应用系统门户采用嵌入页面集成技术，通过统一代理系统定位到应用系统中具体页面的 URL，然后由应用系统门户向应用系统发起页面请求，由应用系统返

回具体页面的内容。其集成后的运行时序关系如图 10-4 所示。

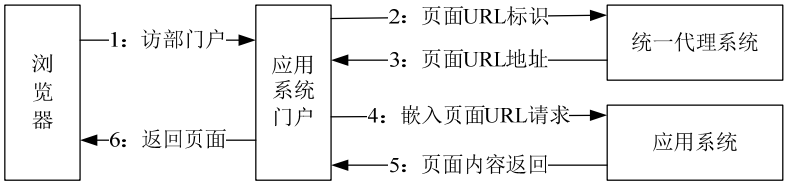

图 10-4　门户界面集成运行时序关系

在门户页面集成技术中各部分的作用如下。

- 浏览器

用户访问应用系统门户的客户端工具。

- 应用系统门户

1．是应用系统基于界面交互统一的出/入口；

2．可以采用 IFramePortlet 整合技术来集成应用系统的界面。

- 统一代理系统

1．具有单点登录集成功能；

2．定义所有应用系统界面 URL 的内外标识；

3．为应用系统门户提供应用系统的具体 URL 地址。

- 应用系统

是界面、业务逻辑、数据访问的一个运行容器，是真正为用户提供应用功能的系统。

2．应用功能集成

应用功能集成的方法是以统一代理系统为核心，各应用系统的功能以 WebService 方式注册在统一代理系统中。统一代理系统既可以作为外部系统与应用系统的中介，为外部系统提供功能服务，同时也可以为内部系统间功能的相互调用提供服务。应用功能集成的模型如图 10-5 所示。

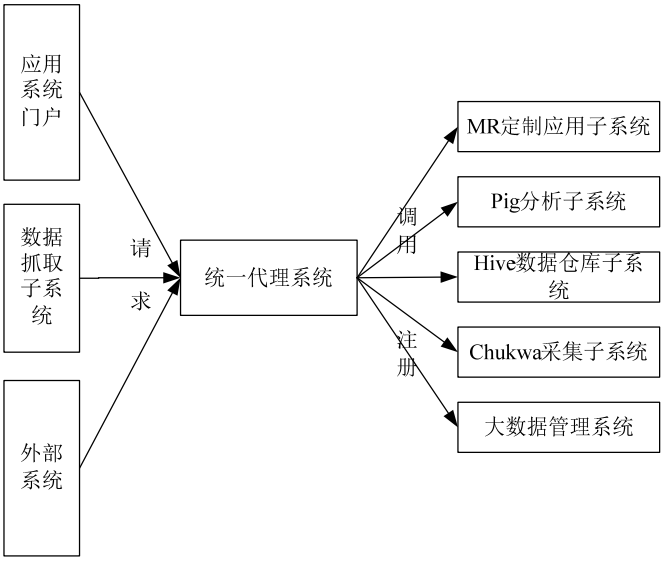

图 10-5　应用功能集成模型

应用系统将 WebService 的服务注册到统一应用代理系统，由统一代理应用系统将其转化成统一的对外 WebService（类似 ESB，即企业服务总线）。应用系统门户等内外部系统通过调用统一的对外 WebService 来向统一代理系统发出服务请求。

内部应用系统也是通过调用统一的对外 WebService 来实现内部系统间的调用的，如 Pig 分析系统调用 MR 定制应用系统的服务时，不是直接调用 MR 的服务，而是经统一代理系统来调用。

3．主从节点集成

无论是 HDFS 系统，还是 HBase 系统，都采用主从模式组合在一起形成一个完整的系统。这些分布式系统节点上所安装的系统都是一样的，配置文件也一样，只有节点的 IP 地址不同。正是通过在配置文件中设定主从节点的 IP 地址，启动命令时可以按照配置文件上的主从节点 IP 的设定，启动节点上相应的进程，使其成为集群中的主节点或者从节点。而从节点也根据配置文件中设定的主节点 IP 地址，将从节点的运行情况发送给相应的主节点。不像门户界面集成和应用功能集成，需要针对每一个功能点在配置文件中加以注册说明，主从节点采用同一个配置文件，其集成模型如图 10-6 所示。

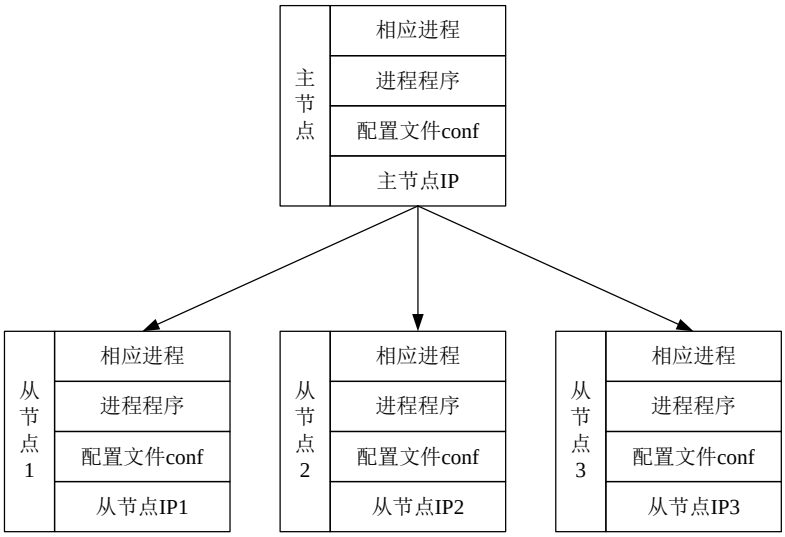

图 10-6　主从节点集成模型

从图 10-6 可以看出，只要节点 IP 与配置文件中的主节点 IP 一致，则启动进程程序中属于主节点的进程，从而该节点成为主节点。从原理上看，这种通过配置文件进行集成的方式属于功能层面的集成。

10.2.2　专项集成设计

专项集成是指企业大数据系统中的各个系统提供专门的接口，满足内部系统间和外部系统之间的集成。专项集成设计主要包括 HDFS 系统、HBase 系统、ZooKeeper 系统、ChuKwa 系统、Pig 系统和 Hive 系统的对外接口设计。

（1）HDFS 的接口设计

HDFS 分布式文件系统主要是为用户提供分布式文件数据的读写服务。HDFS 提供了多种接口使外部系统能够实现分布式文件数据的读写。其接口设计模型如图 10-7 所示。

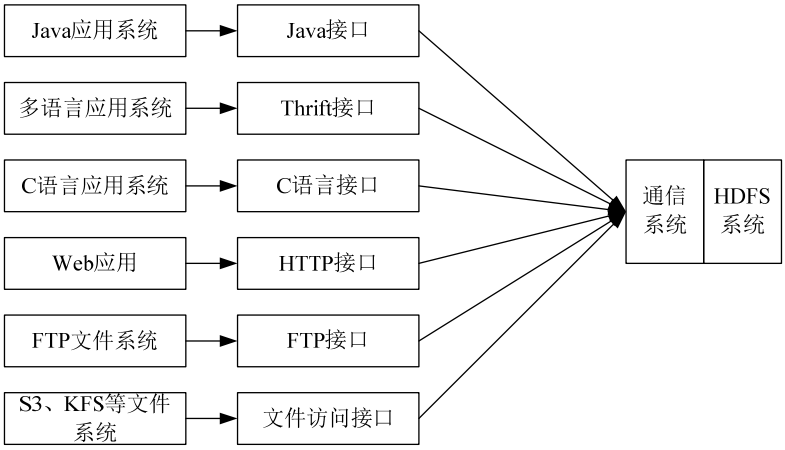

图 10-7　HDFS 分布式文件系统接口设计模型

HDFS 分布式文件系统提供的接口包括 Java 接口、C 语言接口、Thrift 接口、HTTP 接口和多个文件系统的访问接口。另外，HDFS 还提供了多个命令行程序，用户可以使用这些命令来管理和操作分布式文件系统。

（2）HBase 的接口设计

HBase 分布式数据库系统主要是为用户提供分布式数据库数据的读、写、计算服务，其接口设计模型如图 10-8 所示。

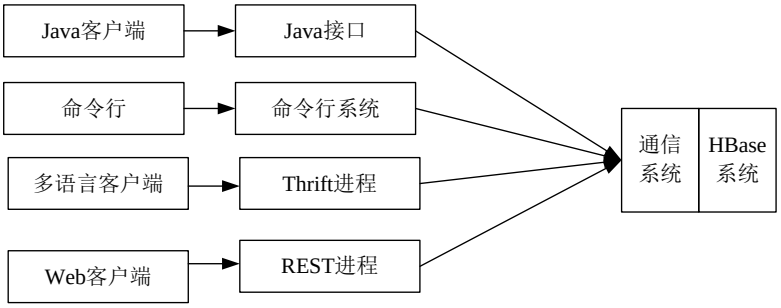

图 10-8　HBase 分布数据库系统接口设计模型

HBase 分布式数据系统主要为用户提供了 Java 接口、命令行系统、多语言客户端、Web 客户端四种接口方式来实现对数据库系统进行访问。

（3）ZooKeeper 接口设计

ZooKeeper 系统为分布式系统提供组服务、名字服务和锁服务等。其中，HBase 系统与 ZooKeeper 系统是紧耦合的，其接口设计模型如图 10-9 所示。

图 10-9　ZooKeeper 系统接口设计模型

ZooKeeper 系统提供了 Java 接口，供应用程序调用 ZooKeeper 系统提供的服务。

（4）Chukwa 的接口设计

Chukwa 是一个数据采集系统，它可以将各类型的数据收集成适合 Hadoop 处理的文件保存在 HDFS 中，然后由 MapReduce 操作。其接口模型如图 10-10 所示。

Chukwa 系统提供了一个代理端，将代理端部署在要采集的数据系统中，然后将采集的数据经收集器写入 HDFS 系统中。Chukwa 系统为用户提供了程序接口、命令行和网页三种方式来操作和访问 Chukwa 文件。

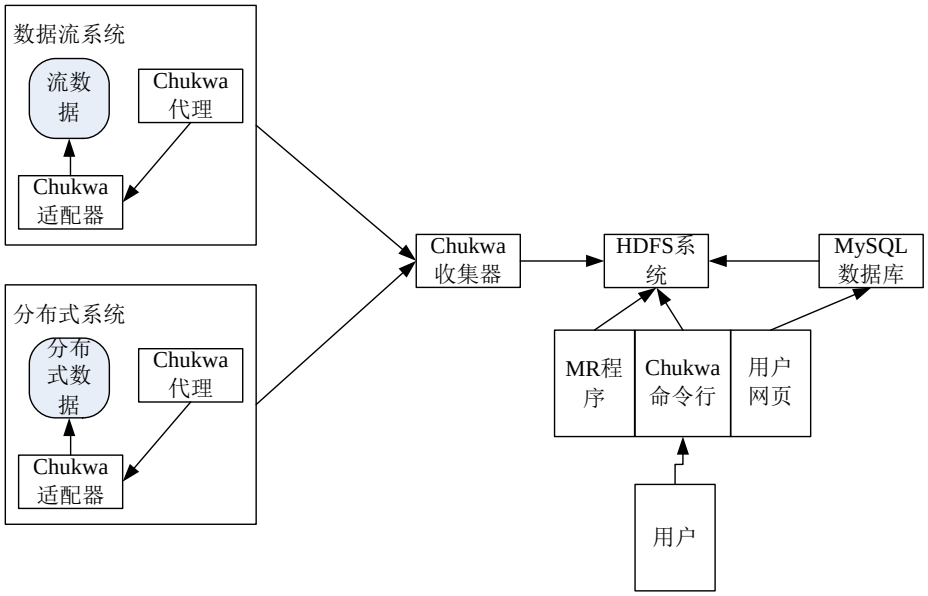

图 10-10　Chukwa 系统接口设计模型

（5）Pig 的接口设计

Pig 是一个基于 Hadoop 的大规模数据分析平台，它可以把类 SQL 的数据分析请求转换为一系列经过优化处理的 MapReduce 运算。其接口设计如图 10-11 所示。

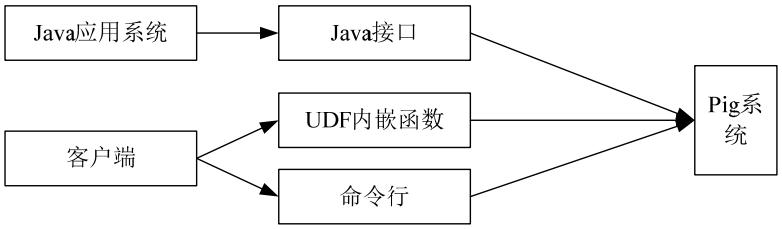

图 10-11　Pig 系统接口设计模型

Pig 系统提供了 Java 接口、UDF 内嵌函数和命令行三种方式，供用户对分布式文件上的

数据进行分析。

（6）Hive 的接口设计

Hive 是建立在 Hadoop 上的数据仓库基础构架。它提供了一系列的工具，可以用来进行数据提取转化加载（ETL），这是一种可以存储、查询和分析存储在 Hadoop 中的大规模数据的机制。其接口模型如图 10-12 所示。

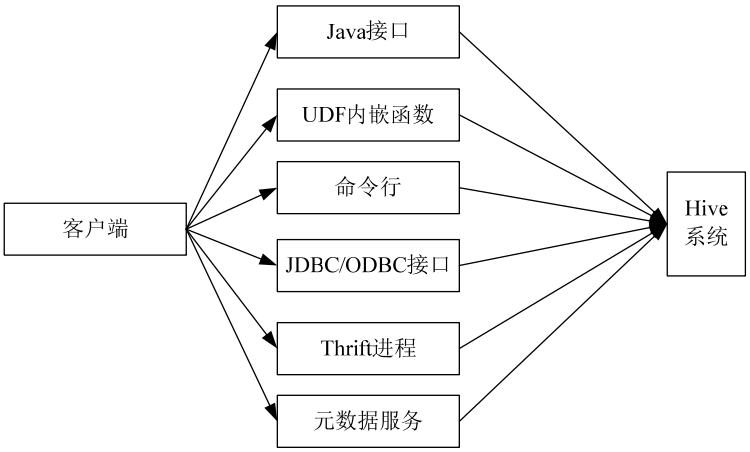

图 10-12　Hive 系统接口设计模型

Hive 系统提供了 Java 接口、UDF 内嵌函数、命令行和 Thrift 进程等多种接口方式，能够方便用户对分布式文件上的数据进行挖掘。

10.3　安全架构设计

由于企业大数据系统的数据资源和计算资源广泛地分布在多个节点上，所以用户的身份、权限等安全，数据资源的存储、传输、访问等安全，以及计算资源的访问、监控、调整、恢复等安全，都是企业大数据系统在进行安全架构设计时需要重点考虑的问题。

企业大数据系统的安全架构设计模型图如图 10-13 所示。

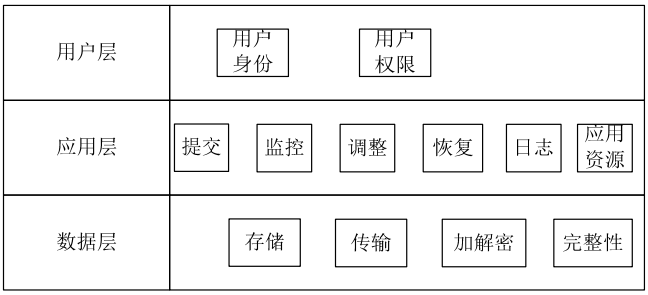

图 10-13　企业大数据系统安全架构设计模型图

从图 10-13 中可以看出，企业大数据系统的安全架构是由针对三层的安全设计构成，这三层分别是用户层、应用层和数据层。针对每一层的关键行为加入安全要素的设计，以确保大数据系统的整体安全。

航标灯：针对每一层的关键行为加入安全要素的设计，以确保大数据系统的整体安全。

10.3.1 用户层安全

用户层的安全主要是指用户身份安全和用户权限安全，主要由统一代理系统来负责。当用户在登录时和登录后访问应用资源、数据资源时，统一代理系统将对用户身份进行认证，对用户权限进行检查。

统一代理系统应当能支持多种用户身份认证支撑模式，比如常规的口令认证和特殊的证书认证、USB Key 认证等。对于用户身份认证，传统的应用系统已经相当成熟，建设者可以直接将原有的系统集成到大数据系统中，实现对用户的身份认证。

用户权限也可以直接将原有的用户权限系统集成到大数据系统中，实现对用户权限的管理，但需要对资源目录管理进行改造。分布式文件的权限管理粒度到文件级，所以在资源目录中对用户的文件授权也只能到文件级。分布式数据的权限管理粒度只能到行级和列族级，而不像传统应用系统可以到字段表，所以在资源目录中对用户的数据授权与以往不同。

10.3.2 应用层安全

应用层安全主要是在于是否能保证应用安全可靠地运行。应用层安全关注的行为包括分布式任务提交、进度和状态监管、运行任务的调整、任务的恢复运行、日志记录和资源权限检查。

HDFS 分布式文件系统和 HBase 分布式文件系统都提供了相应的机制，以确保应用任务的安全运行。HDFS 系统通过 JobTracker 来进行 MapReduce 任务的分配、调度和调整，以确保 MapReduce 任务的安全可靠运行。HBase 系统的 HMaster 主节点和 HRegionServer 为了解决数据库中数据“脏读”和“脏写”的问题，会采用 ZooKeeper 的锁服务。

HDFS 和 HBase 都提供了面向不同类型用户的日志，以帮助这些用户对应用运行进行审计。

大数据管理系统也负责应用的部分安全，如对用户可访问的目录和数据库表的权限。各应用系统会调用大数据管理系统，以确定用户是否有权访问这些资源。

10.3.3 数据层安全

数据层安全重点放在数据是否会丢失、传输过程是否安全、敏感数据是否有加密、数据的完整性是否被破坏这四个方面。

对于分布式文件系统 HDFS 而言，每一个文件的数据块都采用了多副本机制，并将这些副本都存放在不同的节点上。当某个节点的副本失效时，HDFS 还会在一个新的节点上复制一个副本，以确保副本数量与设定要求始终一致。在文件的完整性上，HDFS 对每一个块都采用 CRC-32 的校验方式来确保数据的完整性。

对于分布式数据库 HBase，它提供了相应的分布式数据库安全机制来确保数据不丢失。当 HRegionServer 节点失效时，HMaster 可以通过 HLog 日志恢复失效前的节点数据，从而复原一个新的节点，确保分布式数据不丢失。在数据库数据的完整性方面，因为 HBase 的数据库文件存储是建立在 HDFS 基础上的，所以它相应地继承了 HDFS 的数据完整性的保障机制，能够确保数据库数据的完整性。

为了保障在网络上数据传输的安全，利用数据加密（Encryption）技术，可确保数据在网络传输过程中不会被截取或窃听。SSL（Secure Socket Layer）是为网络通信提供安全及数据

完整性保障的一种安全协议，它已被广泛地应用于 Web 浏览器与服务器之间的身份认证和加密数据传输方面。HDFS 提供有相应的 HTTPS 方式的文件读写接口，确保了数据传输过程的安全。

Hadoop 没有专门针对数据的加解密设置，用户只能在 MapReduce 程序和 HBase 应用程序中做专门的定制开发。

10.4 开发架构设计

企业大数据系统开发架构设计的关注点在于软件开发环境中软件模块的实际组织方式，具体涉及源程序文件、配置文件、源程序包、编译后的目标文件、第三方库文件等。由于企业大数据系统建立在Hadoop全套开源的分布式系统基础之上，而Hadoop全套系统是采用Java开发的，所以企业大数据系统也全套采用 Java 开发。

航标灯：建议企业大数据系统全面采用 Java 语言进行开发。

开发架构将依据逻辑架构的层次来设计，因为开发架构正是对逻辑架构的层次和内部构成进行代码实现，并利用代码来组织出符合逻辑架构的系统。企业大数据系统的开发架构图如图 10-14 所示。

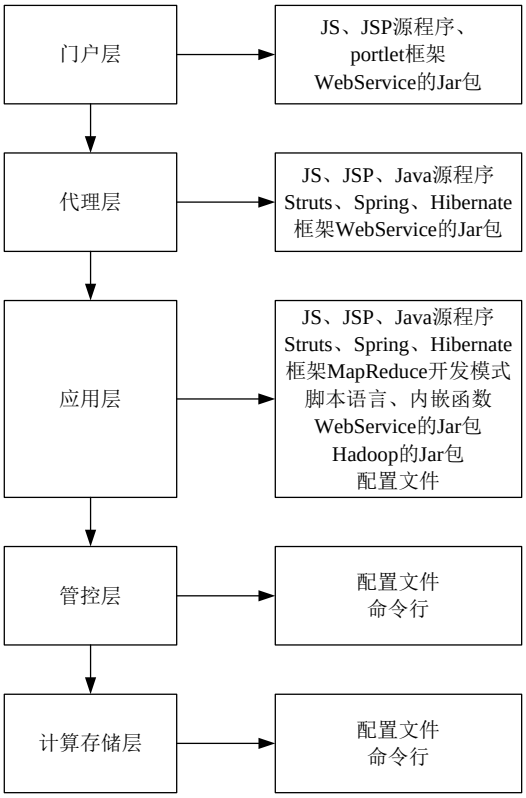

图 10-14　企业大数据系统开发架构图

从图 10-14 可以看出，门户层和代理层都是典型的采用 J2EE 技术标准和 Web 开发方式的应用系统。分布式应用的开发主要集中在应用层，其中最核心的就是按照 MapReduce 开发

模式进行分布式应用程序开发，其次是内嵌函数和脚本语言的开发。管控层和计算存储层的开发主要就是配置文件的开发，以及利用命令行对这两个层的系统进行模式设置和一些检查。

在 IDE 的开发环境中，企业大数据系统创建的工程项目主要分为两类，一类是 Web 工程，另一类是 Java 工程。Web 工程主要是开发门户层、代理层和应用层中的大数据管理系统等。Java 工程可以分成两种，一种是 MapReduce 工程，一种是内嵌函数和脚本编制工程。Java 工程可以按应用系统来划分，比如 MR 定制工程、Pig 工程、Hive 工程和 Chuckwa 工程等。