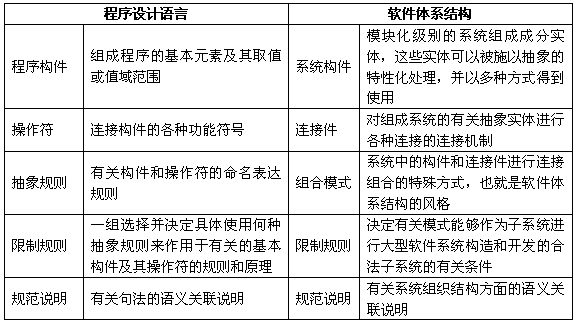
第2章 软件体系结构建模

2、体系结构描述语言与程序设计语言有什么区别？

ADL与其他的语言比较具有以下能力：

1. 构造能力：ADL能够使用较小的独立体系结构元素来建造大型软件系统；
2. 抽象能力：ADL使得软件体系结构中的构件和连接件描述可以只关注他们的抽象特性，而不管其具体的实现细节；
3. 重用能力：ADL使得组成软件系统的构件，连接件甚至是软件体系结构都成为软件系统开发和设计的可重用部件；
4. 组合能力：ADL使得其描述的每一系统元素都有其自己的布局结构，这种描述布局结构的特点使得ADL支持软件系统的动态变化组合；
5. 异构能力：ADl允许多个不同的体系结构描述关联存在；
6. 分析和推理能力：ADL允许对其描述的体系结构进行多种不同的性能和功能上的多种推理分析。ADL与需求语言的区别在于后者描述的是问题空间，而前者描述的是接空间。ADL与建模语言的区别在于后者对整体行为的关注要大于部分的关注，而ADl集中在构建的表示上。

**典型元素含义比较：**



第三章 软件体系结构风格

**1.软件体系结构风格产生背景**

（1）在集中式计算技术时代广泛使用的是大型机/小型机计算模型。它是通过一台物理上与宿主机相连接的非智能终端来实现宿主机上的应用程序。

（2）20世纪80年代以后，集中式结构逐渐被以PC机为主的微机网络所取代。个人计算机和工作站的采用，永远改变了协作计算模型，从而导致了分散的个人计算模型的产生。

**2．C/S和B/S架构的区别P67**

C/S: P56

C/S软件体系结构是基于资源不对等，且为实现共享而提出来的，是20世纪90年代成熟起来的技术，C/S体系结构定义了工作站如何与服务器相连，以实现数据和应用分布到多个处理机上。

C/S体系结构有三个主要组成部分：数据库服务器、客户应用程序和网络。

B/S: P66

浏览器/服务器（B/S）风格就是上述三层应用结构的一种实现方式，其具体结构为：浏览器/Web服务器/数据库服务器。

B/S体系结构主要是利用不断成熟的WWW浏览器技术，结合浏览器的多种脚本语言，用通用浏览器就实现了原来需要复杂的专用软件才能实现的强大功能，并节约了开发成本。从某种程度上来说，B/S结构是一种全新的软件体系结构。

第七章面向服务体系架构SOA

SOA三要素 6.4.8

The coordination layer（协调层）

Information Entities（信息实体）

The relationship between BPM and SOA（BPM和SOA之间关系）

SOA是关于构建可在设计时未知的上下文中重用的软件组件

BPM（Business Process Management业务流程管理）是关于能够精确建模并可能更改使用企业组件的上下文

将BPM和SOA结合起来

基础是强大的理论的支持

一个大的部分仍然缺少：“状态”（恕我直言）

从业务流程到业务流程的定义

一旦基础到位我们应该看到特定领域语言（DSL）的出现，更接近用户的方式很多（不是程序员）

想想关于业务流程

第11章 Hadoop生态系统

1. Hadoop总体架构

**系统架构：**

部署在低成本的Intel/Linux硬件平台上

由多台装有Intel x86处理器的服务器或PC机组成

通过高速局域网构成一个计算集群

各个节点上运行Linux操作系统

**三大主要模式：**

单机模式（standalone mode）

虚拟分布模式（pseudo-distributed mode）

完全分布模式（completely distributed Mode）

**Hadoop软件架构：**

组成：

基于HDFS/HBase的数据存储系统

基于YARN/Zookeeper的管理调度系统

支持不同计算模式的处理引擎

**数据存储系统**

组成：

分布式文件系统HDFS（Hadoop Distributed File System）

分布式非关系型数据库Hbase

数据仓库及数据分析工具Hive和Pig

用于数据采集、转移和汇总的工具Sqoop和Flume。

HDFS文件系统构成了Hadoop数据存储体系的基础

**管理调度系统：**

Zookeeper：提供分布式协调服务管理

Oozie：负责作业调度

Ambari：提供集群配置、管理和监控功能

Chukwa：大型集群监控系统

YARN：集群资源调度管理系统

1. HDFS文件系统

分布式文件系统：

结构：

物理存储资源和对象分散存储在通过网络相连的远程节点上

主控服务器（也称元数据服务器）：负责管理命名空间和文件目录，

远程数据服务器（也称存储服务器）节点：存储实际文件数据

特点：

透明性

高可用性

支持并发访问

可扩展性

安全性

分布式文件系统：

物理存储资源和对象分散存储在通过网络相连的远程节点上

主控服务器（也称元数据服务器）：负责管理命名空间和文件目录，

远程数据服务器（也称存储服务器）节点：存储实际文件数据

1. 分布式存储架构

本节以Hbase为例讲述分布式存储架构

Hbase集群部署

物理部署：Hadoop集群

软件部署：四大组件

Master

Region Server

Zookeeper

Client

4. Hadoop资源管理与作业调度

实现方案：三大组件

Zookeeper提供分布式协同服务

Oozie 提供作业调度和工作流执行

YARN 提供集群资源管理服务

第12章 MapReduce计算模型

**MapReduce的工作流程：**

（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）；（图见PPT所示）

（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；

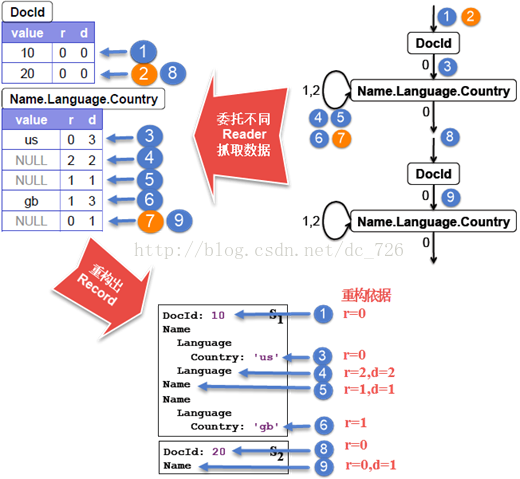
（3）Mapper读取分派给它的输入数据split，并生成相应的键值表；

（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13章 交互式处理



1.      FSM委托Reader1读取DocId第一行，通过r=0重建记录。

2.      检查DocId第二行，发现r=0，则**Reader1停在当前“游标”位置**。FSM将状态变化到Name.Language.Country。

3.      FSM委托Reader2读取Name.Language.Country第一行，通过r=0重建记录。

4.      FSM委托Reader2读取Name.Language.Country第二行。**通过r=2(说明Language字段重复，即Language有多个)重建记录。**

5.      FSM委托Reader2读取Name.Language.Country第三行。**通过r=1和d=1(说明只有Name字段不是NULL)重建记录。**

6.      略过第四行。

7.      检查到第五行，发现r=0，Reader2停在当前位置。FSM再次发生状态变化，继续重建文档2的记录。

8.      FSM委托**Reader1继续读取DocId第二行(之前Reader1就停在这里了)**。

9.      到这里应该已经很清楚了，最后过程就略说了：DocId中没有数据了，FSM状态变化，Reader2继续读取Country的最后一行数据，重建出记录。

**Drill特点：**

**学习成本低**

**低延迟的SQL查询**

**动态查询自描述数据文件（json，text，Parquet），MPR-DB/Hbase表，不需要元数据定义的hive元数据。ANSI SQL**

**嵌套数据支持**

**与ApacheHive一体化（Hive表和视图的查询，支持所有的Hive文件格式和HiveUDFS）**

**BI/SQL工具集成使用标准的JDBC驱动程序**

**访问多个数据源**

**用户自定义UDF**

**高性能（设计上高吞吐量和低延迟，不使用通用的执行引擎，柱形矢量引擎）**

第14章 流计算

