

## 知识储备

### 1. 业务系统概述

业务系统是指在一个专业方面，比如人事系统、财务系统、物资系统等，从某一个方面从上到下的组织架构及所有的业务工作。其中应用最多的是企业业务系统。企业业务系统是解决企业的外部问题，与解决企业内部问题的组织系统相辅相成。业务系统解决的问题是人如何将产品卖出，使得企业利润最大化。在业务系统中应用最多的有 ERP、CRM 等。

#### (1) ERP 系统

ERP(Enterprise Resource Planning)是企业资源计划的建成，由不同的功能模块组成的软件系统。建立在信息技术基础上，集信息技术与先进管理思想于一身，以系统化的管理思想，为企业员工及决策层提供决策手段的管理平台。它是从 MRP（物料需求计划）发展而来的新一代集成化管理信息系统，它扩展了 MRP 的功能，其核心思想是供应链管理。它跳出了传统企业边界，从供应链范围去优化企业的资源，优化了现代企业的运行模式，反映了市场对企业合理调配资源的要求。它对于改善企业业务流程、提高企业核心竞争力具有显著作用。

ERP 是一个信息高度集成的管理系统，不仅可以使企业内部的物流和资金流集成在一起，也可以将企业外部有关供应商以及其市场和客户的物流和资金流信息集成在一起。

目前国际上普遍被采用的 ERP 系统有 SAP，Baan，JDE，Oracle，QAD 等。而国内的 ERP 系统有用友、金蝶、浪潮、神州数码、新中大等。ERP 核心业务功能包括：财务会计；管理会计；销售管理；采购管理；客户关系管理；供应商管理；物流管理；生产计划；质量管理等。一个典型的 ERP 系统除了上述功能外。通常还包括项目管理、投资管理、资金管理辅助功能。

#### (2) CRM 系统

CRM 是指利用计算机网络实现的以“客户”为中心的系统。它通过服务、挖掘客户、维护客户与销售之间的关系，提高客户的满意度、提高服务质量，从而实现企业效益最大化。CRM 主要应用于以下几个方面：

- 客户档案有序管理

以往的客户资料都是通过纸质文档进行存储管理，有时候出去见完客户后，

得急匆匆的打开电脑进行录入客户资料，就怕忘记。而通过 CRM 系统，销售人员可以随时随地地录入客户资料、及时更新客户信息变动，确保信息的完整性。

- 合同管理

合同管理是维护企业双方的利益武器。所以利用 CRM 系统实时维护合同状态细节，及时更新合同的每一个细节及记录每一个合同管理时间、流程，让签约回款流程标准化，执行更加顺畅。

- 销售报表

以往的销售日报都是通过纸质进行上报，但是报表上无法显示销售工作的真实性。而利用 CRM 系统可以清晰的记录销售的每一个环节，每一个工作进度。因此，领导可以通过销售日报及时掌握最新动态，让销售工作的管理更加有序。

- 外勤签到

利用 CRM 系统可以通过 GPS 对员工外勤进行定位，结合现场拍照，还原外勤现场情况。可以很大程度减少销售人员虚假外勤签到的情况。同时，也可以根据销售人员外出拜访记录，了解客户的需求，把客户资源掌握在企业手里，以免造成销售人员一走，客户资源随之流失。

- 预测销售业绩

通过数据看板生成销售数据看板，可以预测销售业绩，直观的展示商机动态，领导可以随时查看商机跟进情况，预测销售结果能够指导团队管理。

## 2. 业务系统的行为数据产生及价值

在使用业务系统过程中，会产生不同的数据。比如用户操作过程中添加的数据，修改项目中具体的内容时产生的数据等，系统出现问题、操作失败等出现的数据。通常情况下，业务系统中产生的数据会有两种保存方式，数据库数据和日志数据。

- 数据库数据

数据库保存的数据一般是用户新增的数据，以及导入的数据。除此之外对系统进行修改的最终结果数据也会保存在数据库中。数据库保存的数据是人们可以直观看到的信息。

- 日志数据

日志作为企业应用系统中一个重要的组成部分。它可以记录系统的几乎所有

行为,并且按照一定的格式表示。通过日志记录的信息可以为企业应用系统纠错,跟踪系统的运行,优化系统性能等。在安全管理中,日志系统通过记录被拒绝的访问,可以反映系统错误操作和恶意攻击的情况,它也是系统安全审计的主要方法之一。在高可靠的系统中,日志记录的信息能准确、及时反映系统的状况,从而可以保障系统运行的连续性。业务系统运行时,会把请求运行情况、异常情况记录在日志文件里。不同语言开发的业务系统有不同的日志数据生成格式,同时也与运行容器(IIS、Tomcat等)的配置有关。

通过分析业务系统产生的日志数据,可以分析业务系统的访问量、运行状态、异常情况、功能访问分布等情况;当系统发生问题时,可以快速定位系统异常问题的发生位置。

### (1) 日志概述

日志主要包括系统日志、安全日志和应用程序日志。系统开发人员和系统维护人员可以通过日志来了解服务器软硬件信息,检验配置过程中的错误及错误发生的原因。通过分析日志可以了解系统的负荷,性能安全性及时纠正系统的错误。

通常情况下,日志分散在不同的存储设备上,如果管理数百个服务器,依次查看每台机器的传统方法查阅日志,很麻烦效果也不高,目前很多公司都使用集中化管理日志,比如开源的 `syslog`,将所有的服务器上的日志收集汇总。

在对日志进行集中化管理过程中,可以将所有机器上的日志进行收集,完整的日志具有以下作用:

**信息查找:**通过检索日志信息, 可以找出系统中存在的 `bug`,并根据日志进行 `bug` 修改。

**服务诊断:**通过对日志信息进行统计分析,了解服务的负荷和服务运行状态,找出耗时请求并进行优化。

**数据分析:**如果是格式化的 `log`,可以做进一步的数据分析,统计、聚合出有意义的信息。

### (2) ELK 概述

ELK 主要用于收集集群日志,从而对日志进行进行有效的查询和检索。ELK 由 `ElasticSearch`、`Logstash` 和 `Kiabana` 三个开源工具组成。

`Elasticsearch` 是一个基于 `Lucene` 的开源分布式搜索服务器,拥有零配置、分

布式、索引自动分片、自动发现、索引副本即止、自动搜索负载等特点。它提供了一个分布式多用户能力的全文搜索引擎，基于 RESTful web 接口，使用 Java 语言开发，并作为 Apache 许可条款下的开放源码发布。主要用于云计算，能够达到实时搜索，稳定、可靠、快速、安装使用方便。

Logstash 主要功能是对日志进行收集、过滤并将其存储，方便以后搜索，自带一个 Web 界面，可以搜索和展示所有日志。

Kibana 是一款基于浏览器页面的 Elasticsearch 前端展示工具，可以为 Logstash 和 Elasticsearch 提供的日志分析友好的 Web 界面，主要用于汇总、分析和搜索重要数据日志。

在安装过程中，彼此之间存在一定的依赖关系，Kibana 依赖 Elasticsearch，Logstash 数据输出到 Elasticsearch，Elasticsearch 数据来源依赖 Logstash。按照依赖顺序安装相应的服务：ElasticSearch -> Logstash -> Kibana

### 3. 业务系统的开发语言与技术框架

不同的开发语言开发的业务系统，产生的日志数据格式是不一样的，在现实生活中，用于开发业务系统常用的开发语言有 .NET、Java 等。

#### (1) .NET

DotNET 是微软公司旗下的一种用于软件网络开发的新型技术。.NET Framework 是指 DotNET 的运行环境

.Net 平台是微软搭建的技术平台，技术人员在此平台上进行应用的搭建与开发。它提供了运行所必须的环境 .NET Framework 类库以及 CLR（公共语言运行时）。好比我们人类的生存必须基于有氧气的基础之下。程序也不例外，它的运行也需要很多所必需的环境。这就是 .Net 平台。

大多数 .NET 的日志框架有着一些共同的概念和特性。根据作用来分主要有以下组件或概念：记录器（Logger）、监视器/目标（Monitor/Target）、包装器（Wrapper）、过滤器（Filter）、布局（Layout）、严重性级别（Severity）。

- 记录器：日志的对象，可以同时连接一个或多个不同的监视器，记录各种不同的消息和信息。

- 监视器：用于存储和显示日志消息的媒介，有多种存在形式。如一般的

文本文件、数据库、网络、控制台、邮箱等。

- 包装器：用于制定日志记录的方式。如：同步/异步记录，（出错时）回滚记录。

- 过滤器：用于根据严重性级别来过滤和忽略某类消息，只记录特定级别的日志消息。

- 布局：用于格式化输出。定义输出项和输出格式。

- 严重性级别：是对消息的分类，严重性级别表示消息的严重程度。也是过滤器的过滤依据。如：调试（DEBUG）、消息（INFO）、警告（WARN）、错误（ERROR）、严重错误（FATAL）严重性依次增强。

现在，已经有很多成熟的.NET 日志工具，免费的、收费的、开源的都有。一些主流的日志工具和框架有：NLog、log4net、Enterprise Library、ObjectGuy Framework 等。

## （2）Java

Java 是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 面向对象程序设计语言和 Java 平台的总称。由 James Gosling 和同事们共同研发，并在 1995 年正式推出。

Java 分为三个体系：

JavaSE（J2SE）（Java2 Platform Standard Edition，java 平台标准版）

JavaEE(J2EE)(Java 2 Platform,Enterprise Edition，java 平台企业版)

JavaME(J2ME)(Java 2 Platform Micro Edition，java 平台微型版)。

常用的日志框架：Log4j、Slf4j、Logback。

在 JDK 1.3 及以前，Java 打日志依赖 `System.out.println()`、`System.err.println()` 或者 `e.printStackTrace()`，Debug 日志被写到 `STDOUT` 流，错误日志被写到 `STDERR` 流。这样打日志有一个非常大的缺陷，即无法定制化，且日志粒度不够细。2001 年发布了 Log4j，后来成为 Apache 基金会的顶级项目。Log4j 在设计上非常优秀，对后续的 Java Log 框架有长久而深远的影响，它定义的 `Logger`、`Appender`、`Level` 等概念如今已经被广泛使用。Log4j 的短板在于性能，在 Logback 和 Log4j2 出来之后，Log4j 的使用也减少了。

Slf4j 也是现在主流的日志门面框架，使用 Slf4j 可以很灵活的使用占位符

进行参数占位，简化代码，拥有更好的可读性。

**Logback** 是 **Slf4j** 的原生实现框架，同样也是出自 **Log4j** 一个人之手，但拥有比 **log4j** 更多的优点、特性和更强大的性能，现在基本都用来代替 **log4j** 成为主流。

## 4. J2EE 框架下的业务系统开发模式

### (1) Java 介绍

Java 语言是一种面向对象的高级语言，目前全球已有上亿系统是使用 Java 进行开发。Java 是一门面向对象编程语言，不仅拥有 C++ 语言的各种优点，还摒弃了 C++ 中难以理解的多继承、指针等知识。因此 Java 语言具有功能强大和简单易用两个明显的特征。

Java 语言具有跨平台性、简单、面向对象、安全性等特点。

- 跨平台性

Java 语言跨平台性是指软件可以不受计算机硬件和操作系统的约束而在任意计算机环境下正常运行，同时 Java 中自带虚拟机，能够很好的实现跨平台性。

- 简单性

Java 语言的简单性表现在 Java 是一种简单的面向对象程序设计语言。Java 语言省略了 C++ 语言中所有的难以理解、容易混淆的特性，例如头文件、指针、结构、单元、运算符重载、虚拟基础类等。它更加严谨、简洁。

- 面向对象

Java 语言是一种面向对象语言，也集成了面向对象的诸多好处，比如代码扩展和代码复用等。

- 安全性

Java 语言编译时要进行 Java 语言和语义检查，保证每个变量对应一个相应的值，编译后生成 Java 类。运行时 Java 类需要类加载器载入，并经由字节码校验器校验之后才可以运行。Java 类在网络上使用时，对它的权限进行了设置，保证了被访问用户的安全性。

Java 程序运行过程，如图 9-8 所示，程序员编写的源码经编译器编译转化为字节码，字节码被加载到 JVM，由 JVM 解释成机器码在计算机上运行。

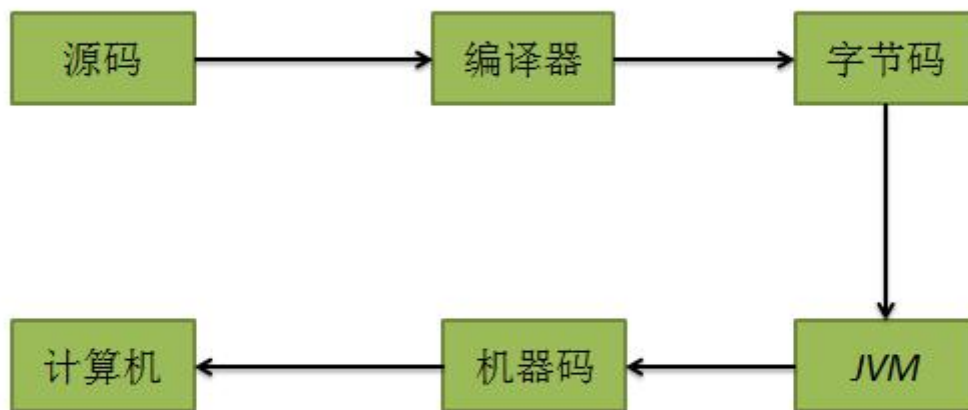


图 9-8 Java 程序运行过程

## (2) Java 用途

Java 是一种常用的计算机编程软件，它是每位程序员必学的技能之一。学习 Java 主要有以下用途：

- 可以用来编写网站

现在很多大型网站都用 jsp 写的，JSP 全名为 Java ServerPages。它是一种动态网页技术，比如我们熟悉的淘宝、天猫、163 邮箱，一些政府网站都是采用 JSP 编写的。

- 可以用来做软件

一般编程语言都可以做软件的，Java 也不例外，例如 Eclipse，MyEclipse 等知名 Java 开发工具。有关开发软件组件，可以了解 JavaSwing 编程或者 awt 相关知识。

- 可以做 Android

Android 是一种基于 Linux 的自由及开放源代码的操作系统，其源代码是 Java。所以市场上见到的手机系统例如 MIUI，阿里云，乐蛙等，都是修改源代码再发行的。Java 做安卓不单单是指系统，还有 APP，对于更多的开发人员来说，他们更多的时间是花在开发 APP 上面。

- 可以做游戏

在以前诺基亚还很流行的时候，你玩的手机游戏有 90%以上都是 Java 开发的。当然现在已经很少人再去开发了，都转到安卓上去了。电脑上也有 java 开发的游戏，经典的是：《我的世界》(mine craft，简称 MC)。



现在很流行的 B/S (浏览器/服务器) 架构管理程序很多就是用 **JAVA** 开发的。  
Java 根据用途的不同, 分为 3 个版本。

Java SE: java 的标准版, 是其他版本的基础, 主要用于开发桌面应用程序。

Java EE: java 的企业版, 主要用于开发企业级分布式网络程序。

Java ME: 主要用于嵌入式系统开发。

### (3) JavaEE 简介

Java EE (Java Enterprise Edition) 是建立在 Java 平台上的企业级应用的解决方案。JavaEE 基于 Java SE (Java Standard Edition) 平台, 提供了一组用于开发和运行的可移植的、健壮的、可伸缩的、可靠的和安全的服务器应用程序的 API (Application Programming Interface)。

JavaEE 是 J2EE 的升级版本, 它具有 J2SE 平台的所有功能, 同时对 EJB、Servlet、JSP、XML 等技术也有较好的支持。JavaEE 设计的目标是为企业级应用提供一种体系结构, 以使企业级应用的开发、部署以及管理等问题更加简单。另外, Java EE 是一个标准, 各平台开发商则依据该标准开发出不同的应用服务器。Sun 公司推出 JavaEE 框架的目的是为了弥补传统 C/S 架构模式应用的不足, 来适应日益增长的 B/S 架构应用的开发。

Java EE 为企业应用开发提供了大量的有价值的服务。包括开发具有扩展性、安全性及易维护性的分布式应用, 需要的大多数系统级功能服务。在 JavaEE 应用中, 通常采用表现层 (UI)、业务逻辑层 (BLL)、数据访问层 (DAL) 的三层架构模型。图 9-9 是一个典型的 Java EE 企业应用的三层结构模型。

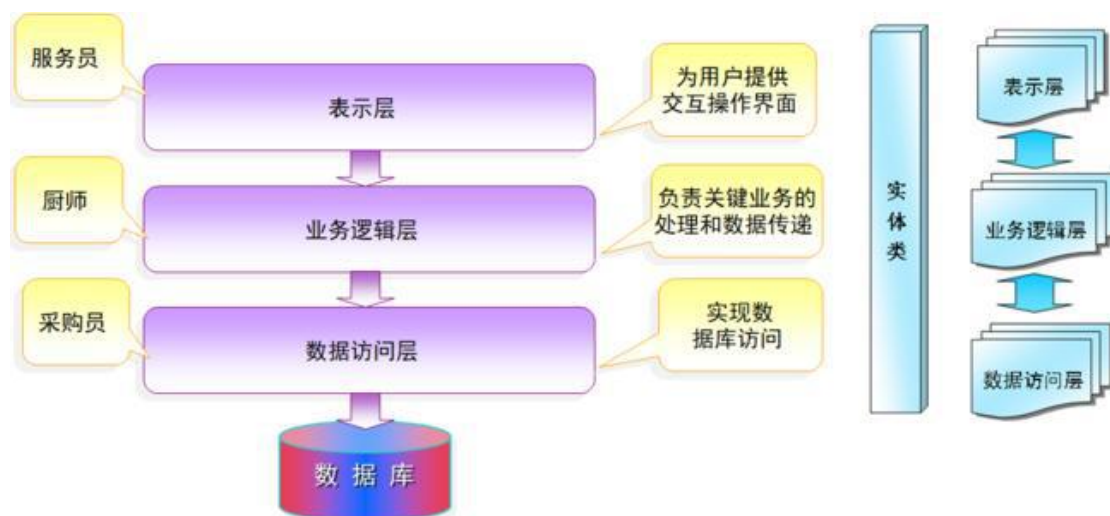


图 9-9 Java EE 企业应用的三层结构模型



表示层：处于 Java EE 应用中的最上层，直接面向用户的一种交互形式，通常的形式是页面，主要用于显示数据和接收数据。同时，还负责检查用户输入的数据的正确性和有效性，展现样式，传递各种友好提示信息。

业务逻辑层：作为一个中间层，在数据的交换过程中起到关键的作用，对数据业务逻辑进行处理。

数据访问层：其作用是负责数据的持久化，持久化载体可以是关系数据库系统、文件、LDAP 服务器等，直接操作数据库。对数据库进行新增、修改、删除、查找等。

#### (4) JavaEE 下的两大框架对比

JavaEE 下有两个应用比较多的框架，分别是 SSH (spring+Struts2+hibernate) 和 SSM (Spring+SpringMVC+MyBatis)。

在实际开发过程中，两个框架具有以下共性：

- 两个框架都是 IoC 容器+MVC 框架+ORM 框架。

IoC 容器是把创建和查找依赖对象的控制权交给容器而不是自己实例化对象；

MVC 框架采用 MVC 分层，模型层处理数据逻辑，通常是模型对象在数据库存取数据，视图层处理数据显示，控制器层处理用户交互，通常从视图读取数据，控制用户输入，并向模型发送数据；

ORM 框架即对象-关系映射模型，在数据库表和持久化对象间进行映射，这样在操作数据库时只需要对对象操作。其中 SSH 框架使用的是 Hibernate，SSM 框架使用的是 MyBatis。MyBatis 的 sql 语句是手动编写的，可以进行更为细致的 SQL 优化，可以减少查询字段，具有高度灵活，可优化，易维护的特点。但需要维护 SQL 和结果映射，工作量大。Hibernate 面向对象查询，以对象化的思维操作数据库，hql 与具体的数据库无关，移植性更好。

- 在 IoC 容器方面，SSH 和 SSM 框架都都 Spring 容器。

Spring 是轻量级的 IoC 和 AOP 容器。IoC 容器是 Spring 的核心，负责创建对象，管理对象，装配对象，配置对象，并且管理这些对象的整个生命周期。管理对象动态向某个对象提供其他对象，通过依赖注入来实现，Spring 有三种注入方式：接口注入、Set 注入和构造注入。Spring AOP 即面向切面编程，可以用在

日志和事务管理等方面。

### （5）微服务架构

微服务架构是围绕着微服务思想构建的一系列结果的简称，是一种在传统软件应用架构的基础上，按照功能拆分为更加细粒度的服务，拆分出的每个服务都是一个独立的应用。

微服务架构的技术选型如表 9-1 所示。

表 9-1 微服务架构的技术选型

服务	技术
微服务实例的开发	SpringBoot
服务的注册与发现	Spring Cloud Eureka
负载均衡	Spring Cloud Ribbon
服务容错	Spring Cloud Hystrix
API 网关	Spring Cloud Zuul
分布式配置中心	Spring Cloud Config
调试	Swagger
部署	Docker

#### 1) Spring Boot

Spring Boot 集成了原有 Spring 框架的优秀基因，省去了 Spring 繁琐的配置，使用使用 Spring Boot 很容易创建一个独立运行（运行 jar,内嵌 Servlet 容器）、准生产级别的基于 Spring 框架的项目。Spring Boot 具有以下作用：

- 可以独立运行的 Spring 项目：可以以 jar 的形式独立运行，通过 `java -jar xx.jar` 即可运行
- 内嵌 servlet 容器：可以选择内嵌 tomcat，jetty 等
- 提供 servlet 简化 maven 配置：一个 maven 项目，使用了 `jar spring-boot-starter-web` 时，就会自动加载 Spring-boot 的依赖包
- 自动配置 spring：Spring boot 会根据在类路径中的 jar 包、类，为 jar 包中的类自动封装配置 bean
- 准生产的应用健康：提供基于 `http\ssh\telnet` 运行时的项目进行监控
- 无代码生成和 xml 配置：主要通过条件注解来实现

## 2) springCloud

Spring Cloud 为开发人员提供了快速构建分布式系统中一些常见模式的工具（例如配置管理，服务发现，断路器，智能路由，微代理，控制总线）。分布式系统的协调导致了样板模式，使用 Spring Cloud 开发人员可以快速地支持实现这些模式的服务和应用程序。他们将在任何分布式环境中运行良好，包括开发人员自己的笔记本电脑，裸机数据中心，以及 Cloud Foundry 等托管平台。Spring Cloud 流程如图 9-10 所示。

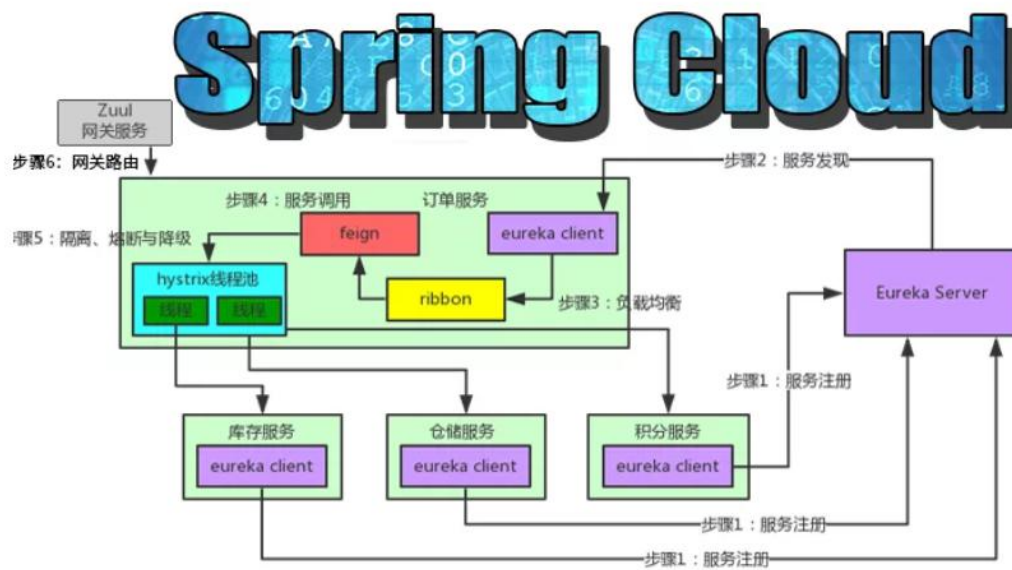


图 9-10 Spring Cloud 流程