

# Java 技术框架概述<sup>①</sup>

欧 锋, 邹 敏, 李晓桢

(江南计算技术研究所, 无锡 214083)

**摘 要:** 从体系结构的角度分析了目前流行的 java 技术框架, 详细阐述了 java 技术体系结构的各主要组成部分。对目前流行的 java 社区的各种技术进行了说明, 并对 java 技术的发展趋势进行了分析。

**关键词:** Java; JVM 技术框架; Java 社区; 发展趋势

## Survey on Java Technology Framework

OU Feng, ZOU Min, LI Xiao-Zhen

(Jiangnan Institute of Computing Technology, Wuxi 214083, China)

**Abstract:** This paper describes the popular Java technology framework from the perspective of architecture, elaborates on the major component of java technology architecture, introduces the popular Java communities with various techniques, and analyses the development trend of Java technology.

**Key words:** Java; JVM architecture of technology; Java community; trends of development

## 1 引言

Java 技术是随着 Internet 的普及而迅速推广起来的一种技术, 因其固有的网络移动、面向对象、平台无关、安全等诸多特征, 迅速成为企业级应用开发的首选技术。Java 虚拟机(JVM)在多个平台上实现了统一语言, Java 语言丰富的开放式类库大量使用设计模式, 成功地改变了很多程序员的编程思想和习惯。很多人了解 Java 语言, 知道 Java 虚拟机。但对于 Java 的技术体系结构是如何衔接和有效运作的, 却不是每个程序员都十分了解的。Java 的技术框架包括了 Java 语言、Java 虚拟机、Java API 和 Java Class 文件(字节码), 虽然 Java 一词被多数人认可为 Java 程序设计语言。可用图 1 来说明 Java 的技术框架各部分的关系。

Java 程序设计语言是编写 Java 程序的基础, Java 字节码和 class 文件是 Java 跨平台性的基础, java 文件通过 Java 编译器生成 class 文件。Java 类库是 Java 程序快速开发的根本, 也是 Java 社区最大的财富。Java 虚拟机是平台相关的, 在不同体系结构和不同的操作系统环境中, 都需要有与平台相关的 Java 虚拟机的实现。

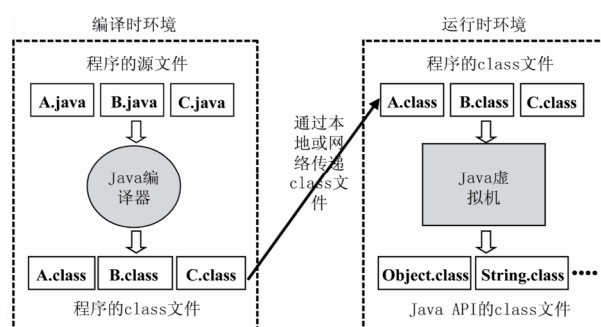


图 1 Java 技术框架

## 2 Java程序设计语言

Java 语言最初是由 James Gosling 为嵌入式消费电子应用设计的, 原名为“Oak”, 后经多年多人设计优化和多方支持帮助, 发展为通用目的的、并发的、基于 class 的面向对象程序设计语言。允许程序设计者一次编写 Java 程序, 可以在多个不同体系结构、不同操作系统、运行有 Java 虚拟机的平台上运行。

Java 语言具有简单性、面向对象、分布式、健壮性、安全性、体系结构中立、可移植性、解释型、高

<sup>①</sup> 收稿时间:2011-11-30;收到修改稿时间:2012-03-06

性能、多线程、动态性等多种特性。

Java 是 C++ 语言的“纯净”版本，没有头文件、指针运算(指针语法)、结构、联合、操作符重载、虚基类等等。Java 剔除了 C++ 中许多很少使用、难以理解、易混淆的特性。另外，Java 的基本解释器以及类支持大约仅为 40KB，加上基础的标准类库和对线程的支持(一个自含的微内核)大约只需增加 175KB。Java 是一个纯的面向对象语言，所有功能和数据都包含在类中，通过类和访问修饰符体现封装性，通过类扩展父类的特征体现继承性，通过接口实现体现多态性。Java 有一个扩展的例程库，用于处理 TCP/IP、HTTP、FTP 等网络访问，Java 程序能够通过 URL 打开和访问网络上的对象，就像处理本地文件系统一样。J2EE 是 Java 社区支持大规模分布式应用开发的独立的软件体系结构。Java 的设计目标从多方面保证了用 Java 编写的程序的可靠性，包括早期的问题检测和后期的运行时检测。Java 与 C++ 最大的不同在于 Java 采用的指针模型可以消除重写内存和损坏数据的可能性。Java 适用于网络和分布式的环境，在安全方面投入了很大的精力，使用 Java 可以构建防病毒和防篡改的系统。如禁止运行时堆栈溢出，禁止在自己的处理空间之外破坏内存，未经授权禁止读写文件等。Java 通过将 Java 程序编译成与体系结构无关的中间目标文件(字节码文件)并通过虚拟机运行中间目标文件实现体系结构中中立的特点。精心设计的字节码不仅可以很容易地在任何机器上被解释执行，还可以迅速地翻成本地机器的代码。通过虚拟机运行还有一些其它优点，如虚拟机可以检测指令序列的行为，以增强其安全性。Java 规范中没有“依赖具体实现”的地方，基本数据类型的大小以及有关算法都做了明确的说明。Java 解释器可以在任何移植了解释器的机器上执行 Java 字节码。字节码还可以在运行时通过即时编译器(JIT 编译器)被快速翻译成运行该应用程序的特定 CPU 的机器码。多线程可以带来更好的交互响应和实时行为。Java 与 C 或 C++ 相比更加具有动态性。它能够适应不断发展的环境。库中可以自由地添加新方法和实例变量，而对客户端却没有任何影响。在 Java 中找出运行时刻类型信息十分直观。当需要将某些代码添加到正在运行的程序中时，动态性将是一个非常重要的特性。如从 Internet 上下载代码，然后在浏览器上运行和必须在运行时分析对象的系统，比如 Java GUI 构建器、智能调试器、可插入组件以及对象数据库。

### 3 Java虚拟机(JVM)

Java 虚拟机可以指一个抽象规范，也可以指一个针对 JVM 规范的实现，还可以指启动运行的一个 Java 虚拟机实例。抽象规范是 JVM 实现者共同遵守的守则，具体实现可以来源于不同的提供商，可以是不同的体系结构，还可以是不同的操作系统平台。图 2 显示了 Java 虚拟机的内部体系结构。

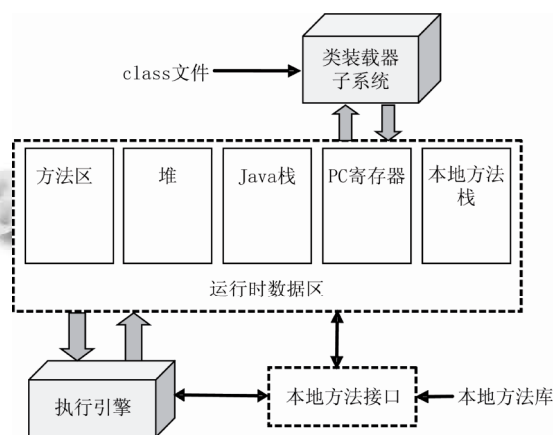


图 2 Java 虚拟机的内部体系结构

Java 虚拟机启动运行后，由类装载器子系统读取 java class 文件，并在内存中构建内存布局，运行时数据区主要包括方法区、堆、Java 栈、PC 寄存器和本地方法栈。方法区存放类数据，主要包括类型信息，动态创建的对象存放在堆中，当一个线程被创建时，会给它分配一个 Java 栈和 PC 寄存器，当执行 Java 方法时，PC 寄存器指向下一条要执行的字节码指令，Java 栈存放该线程中 Java 方法调用的状态——包括它的局部变量，被调用时传进来的参数，它的返回值，以及运算的中间结果等等。本地方法调用的状态，则依赖于具体的实现存放在本地方法栈中。

Java 栈由许多栈帧(stack frame)或帧(frame)组成，一个栈帧包含一个 Java 方法的调用状态。当线程调用一个 Java 方法时，虚拟机压入一个新的栈帧到该线程的 Java 栈中；当该方法返回时，这个栈帧被从该线程的 Java 栈中弹出并抛弃。Java 虚拟机没有寄存器，其指令集使用 Java 栈来存储中间数据。这样设计使 JVM 的指令集尽量紧凑，同时也便于 Java 虚拟机在那些只有少量通用寄存器的平台上实现。另外，Java 虚拟机这种基于栈的体系结构，也有助于运行时某些虚拟机实现的动态编译器和即时编译器的代码优化。各线程

的 Java 栈内存区是私有的,任何线程都不能访问另一个线程的 PC 寄存器和 Java 栈。

#### 4 字节码及Java Class文件

Java class 文件是对 Java 程序二进制文件格式的精确定义。每一个 Java class 文件都对一个 Java 类或者 Java 接口作出了全面描述。一个 Java class 文件只能有一个类或者接口。无论 Java class 文件在何种系统上产生,无论 Java 虚拟机在何种系统上运行,对 Java class 文件的精确定义使得所有 Java 虚拟机都能够正确地读取和解释所有 Java class 文件。

class 文件的版本也是随着 JDK 的版本演进而演进的,查看 Java class 文件的版本号,可以看 Java class 文件的第 5~8 个字节。其中 5、6 字节为次版本号,7、8 字节为主版本号。比如由版本号为 1.6.0\_10 的 javac 生成的 class 文件,次版本号为 0,主版本号为 50。Java class 文件的结构如图 3 所示。

```
ClassFile {
    u4 magic;
    u2 minor_version;
    u2 major_version;
    u2 constant_pool_count;
    cp_info constant_pool[constant_pool_count-1];
    u2 access_flags;
    u2 this_class;
    u2 super_class;
    u2 interfaces_count;
    u2 interfaces[interfaces_count];
    u2 fields_count;
    field_info fields[fields_count];
    u2 methods_count;
    method_info methods[methods_count];
    u2 attributes_count;
    attribute_info attributes[attributes_count];
}
```

图 3 Java class 文件结构

Java 字节码文件结构是一个复杂的数据结构,Java class 文件是 8 位字节的二进制流。数据项按顺序存储在 class 文件中,相邻的项之间没有任何间隔,这样可以使 class 文件紧凑。占据多个字节空间的按照高位在前的顺序分为几个连续的字节存放。和 Java 类可以包含多个不同的字段、方法、方法参数和局部变量一样,Java class 文件也能够包含许多不同大小的项。

在 class 文件中,可变长度项的大小和长度位于其实际数据之前。这个特性使得 class 文件流可以从头到尾被顺序解析,首先读出项的大小,然后读出项的数据。

#### 5 Java类库

Java 类库是 Java 程序设计人员最喜欢最离不开的工具,Java 类库是一个庞大的库结构。java.lang.Object 类是 Java 类库层次结构中的根,所有的类、数组等都继承 Object 类的方法。随着技术的发展,Java 类库也是不断扩展的。目前从 JDK1.6.0 的 API 文档来分析,共有 200 余个包。按照相关功能划分,Java 类库大致可以分为以下几类:

- 1)基础、工具类
- 2)管理控制类
- 3)图形相关类
- 4)网络处理相关类
- 5)文件 IO 相关类
- 6)安全相关类
- 7)Web Services 相关类
- 8)数据库操作相关类
- 9)CORBA 相关类
- 10)W3C 相关类
- 11)Applet 相关类
- 12)组件框架相关类

#### 6 Java社区与Java技术的发展趋势

随着网络的普及和 Java 技术的日益成熟,采用 Java 技术来开发应用程序和信息系统越来越多,造就了庞大的 Java 开发队伍。为了能够更快速的开发出高质量的应用系统,在 Java 技术领域,涌现出很多很好的 Java 技术框架,如 Struts、Spring、Hibernate 等。这些 Java 技术框架的出现,又进一步推动了 Java 技术的普及和发展。

随着网络的普及,网络计算、普适计算、服务计算、云计算、多核、众核等新兴计算技术的发展,对 Java 技术的发展也产生了积极的推动作用。在 Java 技术的框架中,Java 语言、Java 虚拟机、Java 库、Java 字节码文件等也会针对新技术的发展而作出相应的变革。Java 语言是否要适应多核和众核的并行编程模式;Java 虚拟机是否要充分利用多核的资源,加速 Java 程序的执行,优化垃圾回收算法,支持事务存储等;Java

本地库是否要适应多核和众核的环境; Java 类库已经在 Web Service 编程方面作出了积极的努力, 推动了服务计算技术的发展; Java 字节码文件是否增加新的内容以适应新技术的发展呢; Java 在对象数据库开发方面如何提供支持; 由于 Java 技术框架的特点, 在高性能计算方面, Java 一直没有好的表现。然而 Java 在 HPC 方面的努力一直没有停止过。

## 7 小结

Java 技术是目前流行的、最广泛采用的软件开发技术, 但很多人只是从 Java 程序设计语言的角度理解 Java 技术, 缺乏对 Java 技术框架的全面认识。本文针对 Java 的整体技术框架进行了概述, 阐述了 Java 程序

(上接第 196 页)

## 4 结语

在本文中, 改进的全局 K 均值算法解决了聚类分析时的问题, 不仅可以利用最小目标函数来解决初始中心问题, 还可以避免孤立点的问题。马氏距离有效地消除错误数据的样本数据, 从而减少了计算量。因为动态模糊神经网络是不容易获得的梯度信息, 网络培训时间太长, 结构复杂等缺点, 减法聚类用于简化网络结构, 并把改进 PSO 算法用于优化网络参数, 避免经验的盲目性和随意性。仿真结果表明, 它优于传统的动态递归模糊神经网络, 它需要更少的隐藏节点和参数, 具有较高的预测精度和实用性。

### 参考文献

- 1 薛尧予, 王建林, 于涛, 赵利强. 基于改进 PSO 算法的过程模型参数估计. 仪器仪表学报, 2010, 31(1): 178-182.
- 2 Shivakumar RL. Implementation of an Innovative Bio Inspired GA and PSO Algorithm for Controller design considering Steam GT Dynamics. Journal of Computer Science Issues, January 2010, 24(2): 132-140.

设计语言、Java 虚拟机、Java 库、Java 字节码文件之间的关系, 最后在新兴技术不断涌现的今天, 对 Java 技术的发展趋势进行了探讨。

### 参考文献

- 1 Java 虚拟机规范. [http://java.sun.com/docs/books/jvms/second\\_edition/html/VMSpecTOC.doc.html](http://java.sun.com/docs/books/jvms/second_edition/html/VMSpecTOC.doc.html)
- 2 Gosling J, Joy B, Steele G, Bracha G. Java™ Language Specification. The 3rd Edition. Prentice Hall, Jun 14, 2005.
- 3 Horstmann CS, Cornell G. 叶乃文, 等译. JAVA2 核心技术卷 I: 基础知识(原书第 7 版). 北京: 机械工业出版社, 2006.
- 4 Horstmann CS, Cornell G. 陈昊鹏, 等译. JAVA2 核心技术卷 II: 高级特性(原书第 7 版). 北京: 机械工业出版社, 2006.

- 3 武俊峰, 艾岭. 一种基于改进算法的模糊模型辨识. 哈尔滨理工大学学报, 2010, 20(5): 165-170.
- 4 穆朝絮, 张瑞民, 孙长银. 基于粒子群优化的非线性系统最小二乘支持向量机预测控制方法. 控制理论与应用, 2010, 27(2): 164-168.
- 5 王博, 孙玉坤. 基于数据场聚类的模糊神经网络在发酵过程中的应用. 仪器仪表学报, 2009, 30(5): 944-948.
- 6 Xue YY, Wang JL, Yu T, Zhao LQ. Parameter estimation of fermentation process model based on an improved PSO algorithm. Chinese Journal of Scientific Instrument, 2010, 31(1): 178-182.
- 7 田雨波. 混合神经网络技术. 2nd ed. 北京: 科学出版社, 2009. 148-160.
- 8 吕奕清, 林锦贤. 基于 MPI 的并行 PSO 混合 K 均值聚类算法. 计算机应用, 2011, 32(1): 112-118.
- 9 李国勇. 智能控制及其 MTLAB 的实现. 3rd ed. 北京: 电子工业出版社, 2007. 156-188.