

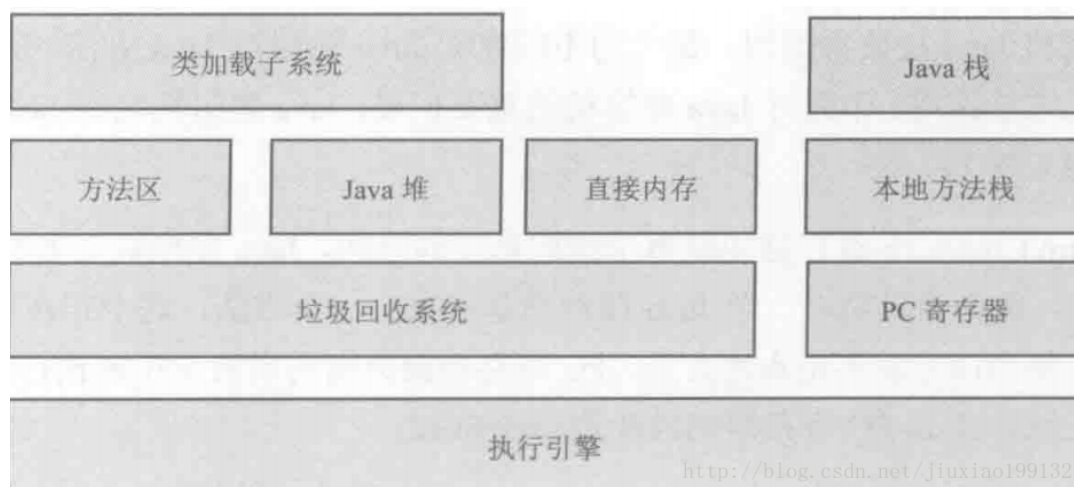
前言

Visual, VMware属于系统系统虚拟机。运行与系统操作系统上的一款软件，提供仿真硬件环境，可以运行一些其他操作系统的软件平台。

Java虚拟机是一个虚拟机，属于程序虚拟机。我们编写的程序运行与Java虚拟机上，再由Java虚拟机与操作系统进行交互从而运行我们编写的程序。

Java虚拟机有多个版本（hotspot, Openjdk, J9等），这里所述基于hotspot，因为这个版本是使用最多，最主流的一个版本

jvm结构



如上图所示，Java虚拟机由9个部分组成：

- **类加载子系统**：负责从文件系统或者网络中加载Class信息，加载的类信息存放于一块称为方法区的内存空间。除了类信息外，方法区中可能还会存放运行时常量池信息，包括字符串字面量和数字量（这部分常量信息是Class文件中常量池部分的内存映射）
- **Java堆**在虚拟机启动的时候建立，它是Java程序最主要的内存工作区

域。几乎所有的Java对象实例都放于Java堆中。堆空间是所有线程共享的，这是一块与Java应用密切相关的内存区间

- 直接内存是在Java堆外的、直接向系统申请的内存区间。通常，访问直接内存的速度会优于Java堆。因此出于性能考虑，读写频繁的场所可能会考虑使用直接内存。由于直接内存存在Java堆外，因此它的大小不会受限于Xmx指定的最大堆大小，但是系统内存是有限的，Java堆和直接内存的总和依然受限于操作系统能给出的最大内存。Java的NIO库允许Java程序使用直接内存
- 垃圾回收系统是对Java堆，方法区，直接内存进行回收，释放内存的一个系统，它是自动执行的，主要的清理对象就是Java堆
- 每一个Java虚拟机线程都有一个私有的Java栈。一个线程的Java栈在线程创建的时候被创建。Java保存着帧信息，Java栈中保存着局部变量、方法参数，同时和Java方法的调用、返回密切相关
- 本地方法栈和Java栈非常类似，最大的不同在于Java栈用于Java方法的调用，而本地方法栈用于本地方法调用
- PC寄存器也是每个线程私有的空间，Java虚拟机会为每一个Java线程创建PC寄存器。在任意时刻，一个Java线程总是在执行一个方法，这个正在被执行的方法称为当前方法。如果当前方法不是本地方法，PC寄存器就会指向当前正在被执行的指令。如果当前方法是本地方法，那么PC寄存的值就是undefined
- 执行引擎是Java虚拟机最核心组件之一，它负责执行虚拟机的字节码。现代虚拟机为了提高执行效率。会使用即时编译技术方法编译成机器码后再执行
- 类型信息和类的静态变量都存储在方法区中，常量池本身是方法区中的一个数据结构。常量池中存储了如字符串、final变量值、类名和方法名常量