**第10章 晚期（编译期）优化**

在部分商用虚拟机（Sun HotSpot）中，Java程序（\*.class）是通过解释器（Interpreter）进行解释执行的。

当虚拟机发现某个方法或代码块的运行特别频繁，就会把这些代码认定为“热点代码”（Hot Spot Code）。为了提高热点代码的执行效率，在运行时，虚拟机将会把这些代码编译成与本地平台相关的机器码，并进行各种层次的优化。完成这个任务的编译器称为即时编译器（JIT编译器）。Java虚拟机并无规定JIT编译器必须存在。

一、HotSpot虚拟机内的即时编译器

（1）解释器与编译器

大部分主流的商用虚拟机（HotSpot、J9）都同时包含解释器与编译器。

解释器与编译器两者各有优势：当程序需要迅速启动和执行时，解释器可以首先发挥作用，省去编译的时间，立即执行在程序运行后，随着时间推移，编译器逐渐发挥作用，把越来越多的代码编译成本地代码之后，可以获得更高的执行效率。

当程序运行环境中内存资源限制较大，可以使用解释执行节约内存，反之可以使用编译执行来提升效率。

因此，在虚拟机执行架构中，解释器与编译器经常配合工作。

HotSpot虚拟机中内置两个即时编译器，分别称为Client Compiler和Server Compiler，或者简称为C1编译器和C2编译器。目前主流的HotSpot虚拟机中，默认采用解释器与其中一个编译器直接配合的方式工作，称为“混合模式”。“混合模式”中，程序选择哪个编译器，取决于虚拟机运行的模式，HotSpot会根据自身版本与宿主机器的硬件性能自动选择运行模式。用户也可以使用参数强制使用“解释模式”或“编译模式”来执行程序。