简介

GraalVM是一个高性能的JDK发行版，旨在加速用Java和其他JVM语言编写的应用程序的执行，同时还为JavaScript，Python和许多其他流行语言提供运行时。

运行时模式[#](https://www.graalvm.org/22.1/docs/introduction/#runtime-modes)

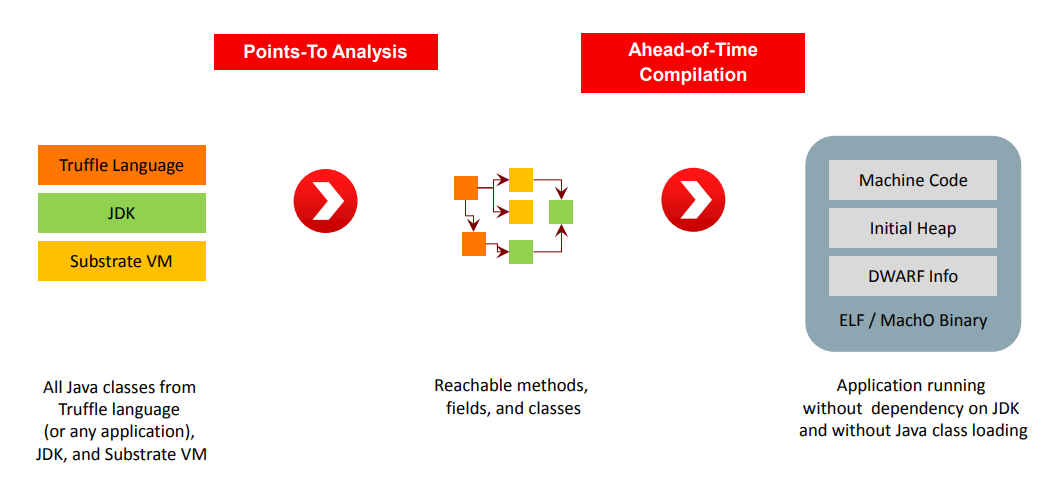
JVM 运行时

即传统的jvm运行模式。在运行时，应用程序在 JVM 上正常加载和执行。JVM 将 Java 或任何其他 JVM 本机语言的字节码传递给编译器，编译器将其编译为机器代码并执行。

native image

native image是一种提前将 Java 代码编译为独立可执行文件**的技术**。此可执行文件包括应用程序类、其依赖项中的类、运行时库类以及来自 JDK 的静态链接的本机代码。它不在 Java VM 上运行，但包括必要的组件，如内存管理、线程调度等。与传统JVM 相比，生成的程序具有更快的启动时间和更低的运行时内存开销。

与传统的Java虚拟机不同，Native Image是封闭式的静态分析和编译，不支持class的动态加载，程序运行所需要的多有依赖项均在静态分析阶段完成。此外GraalVM Native Image运行在一个名为SubstrateVM的轻量级的虚拟机之上。虽说是轻量级虚拟机，SubstrateVM却拥有运行Java程序所必需的所有组件，包括deoptimizer、gc及thread scheduling等。

Native Image的运行机制如图：  


编译器先通过静态分析找到Java程序用到的所有类，方法和字段以及一个非常小的SVM运行时，然后通过AOT编译，生成一个可执行文件

Truffle

[Truffle](https://www.graalvm.org/22.1/reference-manual/java-on-truffle/)是Java虚拟机规范的实现，使用[Truffle语言实现框架](https://www.graalvm.org/22.1/graalvm-as-a-platform/language-implementation-framework/)构建，它是一个完整的Java VM，包括所有核心组件，实现与Java运行时环境库相同的API，并重用GraalVM中的所有JAR和本机库。相当于一个用Native image实现的jvm。

构建Native image