

转载请注明链接: https://blog.csdn.net/feather_wch/article/details/79800614

总结Java虚拟机、Dalvik虚拟机、ART虚拟机的相关要点。

Android虚拟机基本知识(JVM\DVM\ART\JIT)

1、JVM是什么?

1. JVM 是 Java Virtual Machine
2. JVM 本质是 软件 , 是 计算机硬件的一层软件抽象 , 屏蔽了 底层实现的差异
3. Java 编译的程序会生成 JVM字节码 , 字节码需要通过 JVM 被翻译成 机器指令
4. JVM 本质作用是将平台无关的 .class 内的字节码翻译成平台相关的 机器码 -实现跨平台

2、Dalvik虚拟机(DVM)是什么

1. 是 android 中使用的虚拟机。所有的 Android程序 都运行在Android系统进程中, 每个 进程 都对应一个 Dalvik虚拟机实例
2. DVM和JVM 都提供了 对象生命周期管理、 堆栈管理、 线程管理、 安全 和 异常管理 以及 垃圾回收 等重要功能

3、DVM和JVM的区别

1. JVM运行的是 JAVA字节码 , 通过解释 class中内容 来运行程序。
2. DVK运行的是 Dalvik字节码 , 均由 Java字节码 转换而来, 并都被打包到 DEX(Dalvik Executable) 执行文件, DVM 通过解释 DEX文件 来执行这些字节码。
3. JVM 生成的class文件, 每个class文件都会有一个常量池, 会导致信息冗余。体积过大。
4. DVM 生成的 Dex 文件共享 一个常量池 能减少体积。
5. JAVA虚拟机 基于栈结构 , 程序在运行时虚拟机需要频繁的从栈上读取写入数据, 这个过程需要更多的指令分派与内存访问次数, 会耗费很多CPU时间。
6. Dalvik虚拟机 基于寄存器架构 , 数据的访问通过寄存器间直接传递, 这样的访问方式比基于栈方式要快很多。

4、JIT(Just in time)和AOT(Ahead of time)是什么?

1. Just In Time , 即时编译技术
2. Ahead Of Time , 预编译技术
3. JVM 的做法是将 字节码 逐条解释翻译, 比C/C++编译的可执行二进制字节码程序慢。通过 JIT 会将一些方法编译成对引得汇编代码, 存储起来, 以后调用的时候不需要解释可以直接运行。(Dalvik虚拟机在2.2中增加了JIT)
4. AOT 就是指C/C++这类语言, 编译器在编译时直接将程序源码编译成目标机器码, 运行时直接运行机器码。

5、DVM和ART(Android Runtime)虚拟机的区别与联系

1. Dalvik虚拟机 执行的是 dex字节码 , ART虚拟机 执行的是 本地机器码

2. DVM 原理：依靠 JIT编译器 去解释执行，运行时 动态地 将执行频率很高的 dex字节码 翻译成 本地机器码 然后在执行，但是将dex字节码翻成本地机器码是发生在应用程序的运行过程中，并且应用程序每一次重新运行的时候，都要重新做这个翻译工作。
3. ART 原理：应用程序仍然是一个包含 dex字节码 的apk文件。在 安装应用的时候， dex中的字节码将被编译成 本地机器码，之后每次打开应用，执行的都是本地机器码。移除了运行时的解释执行，效率更高，启动更快。（安卓在4.4中发布了ART运行时）

6、ART的优点

1. 系统性能显著提升
2. 应用启动更快、运行更快、体验更流畅、触感反馈更及时
3. 续航能力提升
4. 支持更低的硬件

7、ART的缺点

1. 更大的存储空间占用，可能增加10%-20%
2. 更长的应用安装时间

学习和参考资料

1. [JAVA虚拟机、Dalvik虚拟机和ART虚拟机简要对比](#)