1. 编译器优化技术
2. 概述

编译器的目标是将程序代码翻译为本地机器码，输出代码优化质量的高低是决定编译器优秀与否的关键，所有的优化手段都是建立在代码的中间表示或者是机器码之上的，而不是Java源码。

1. 方法内联
2. 概述

方法内联是编译器最重要的优化手段，因为除了消除方法调用的成本之外，它更重要的意义是为其他的优化手段建立良好的基础。它的优化目的是将目标方法的代码原封不动地复制到发起调用地方法之中，避免发生真实的方法调用。

1. Java的方法内联

由于Java的默认实例方法是虚方法，需要在运行时进行方法接收者的多态选择，可能存在多于一个版本的方法接收者，只有是哦也能够invokespecial指令调用的私有方法，实例构造器，父类方法和使用invokestatic指令调用的静态方法才会在编译器进行解析。

对于虚方法，编译器静态地做内联很难确定应该使用哪个方法版本，为了解决虚方法和内联的矛盾，Java虚拟机引入了类型继承关系分析（CHA），用于确定在目前已经加载的类中，某个接口是否有多于一种的实现，某个类是否存在子类，某个子类是否覆盖了父类的某个虚方法等。这样，编译器在进行内联时会根据不同情况进行处理，如果时非虚方法，则直接内联；如果是虚方法，则向CHA查询该方法在当前状态下是否有多个方法版本，如果只有一个版本，就以当前的方法进行内联，这成为守护内联，不过由于Java是动态连接的，可能随时加载到新的类型改变CHA的结论，因此需要给这个内联留逃生门，当出现新的加载类时，必须抛弃已经编译的代码，退回到解释状态执行，或者重新编译。

如果CHA查询出来有多个方法版本可供选择，则使用内联缓存来缩减方法调用的开销，在这种状态下方法调用真的发生了，但是比起直接查询虚方法更快一些，内联缓存是一个建立在目标方法正常入口之前的缓存，记录下方法接收者的版本信息，并且在每次进行方法调用时比较接收者的版本。如果以后每次调用的方法版本都相同，则就是一种单态内联缓存，通过该缓存，比不内联的非虚方法调用仅多了依次类型判断开销；如果方法接收者不一致，则退化成超多态内联缓存，开销相当于查找虚方法表进行方法分派。

1. 逃逸分析
2. 原理

逃逸分析不是直接优化代码的手段，而是为其他优化措施提供依据的分析技术，基本原理就是分析对象动态作用域，当一个对象在方法里面被定义后，它可能被外部方法所引用，例如作为调用参数传递到其他方法中，这种成为方法逃逸；甚至可能被外部线程访问到，譬如赋值给可以在其他线程中访问的实例变量，这种称为线程逃逸。从不逃逸，方法逃逸到线程逃逸，称为对象从低到高的不同逃逸程度。根据不同的逃逸程度，会对这个对象实例采取不同程度的优化。

1. 栈上分配

Java堆中的对象对于各个线程都是共享和可见的，堆上的对象无论是标记可回收对象还是内存回收，都会消耗大量资源。如果确定一个对象不会逃逸出线程之外，让对象在栈上分配内存，内存会随着栈帧的出栈而销毁，降低垃圾收集系统的压力，栈上分配支持方法逃逸，不支持线程逃逸。

1. 标量替换

若一个数据无法再继续分解，如Java虚拟机中的原始数据类型可以称为标量；相对地，可以继续分解地数据称为聚合量，Java中的对象就是聚合量，如果将一个Java对象拆散，根据程序访问的情况，将其用到的成员变量恢复为原始类型来访问，这个过程就是标量替换。

加入逃逸分析能够证明一个对象不会被方法外部访问，并且这个对象可以拆散，则程序真正执行时可能不去创建这个对象，而是直接创建它的若干个被这个方法使用的成员变量来代替，这些成员变量可以分配再栈上，也可以为后续的优化手段创造条件。标量替换不允许对象逃逸出方法范围内，即方法逃逸。

1. 同步消除

如果逃逸分析确定一个变量不会逃逸出线程，无法被其他线程访问，则这个变量的读写不会有竞争，则同步措施可以安全地消除。

1. 公共子表达式消除

如果一个表达式已经被计算过了，并且值一直没有变化，则该表达式这次地出现称为公共子表达式，直接使用原来地值即可。如果这种优化限于程序基本块内，称为局部公共子表达式消除；如果优化的范围涵盖多个基本块，则称为全局公共子表达式消除。

1. 数组边界检查消除

Java语言中会对数组的上下界进行范围检查，否则会抛出一个运行时异常，对于程序员十分友好，避免了溢出攻击。但是对于虚拟机的执行子系统来说，每次数组元素的读写都带有一次隐含的条件判定操作，这是一种性能负担。

Java的各种安全检查，如数组越界，空指针访问，除数为0等都会有异常检查，这些检查会包含有隐式的开销。为了消除这些隐式的开销，如数组边界检查优化会尽可能地把运行期检查提前到编译器完成，或者隐式异常处理，如进程层面的异常处理器，只要数组不为空，就不会有额外的判空开销，但是一旦数组为空，则需要转到异常处理器中，速度比一次判空检查慢多了。型号，HotSpot虚拟机会根据运行期收集的性能监控信息自动选择合适的方案。