Java性能调优工具“ JRMC ”的介绍 PDF转换可能丢失图片或格式，建议阅读原文
https://www.100test.com/kao_ti2020/278/2021_2022_Java_E6_80_A7_E8_83_BD_c97_278613.htm 【编者按】 Java虚拟机(JVM)及垃圾收集器(GC)负责管理大多数的内存任务，但是Java应用系统中还是有可能出现内存泄漏。事实上，OOM之类的现象在大型项目中也是一个常见的问题。避免内存泄漏的第一步是要弄清楚它是如何发生的，然后对症下药。 【正文】 Java虚拟机(JVM)及垃圾收集器(GC)负责管理大多数的内存任务，但是Java应用系统中还是有可能出现内存泄漏。事实上，OOM之类的现象在大型项目中也是一个常见的问题。避免内存泄漏的第一步是要弄清楚它是如何发生的，然后对症下药。 那究竟是什么导致了 Java 程序中的内存泄漏呢？难道 Java 虚拟机的垃圾收集器不应该管理未使用的内存吗？是的，它会进行管理，但是垃圾收集的对象只能是不再被引用的对象。但是，某些不再需要的对象，却在系统的某个地方仍在引用它，这样就不能对这些对象进行垃圾收集，在日志中的大量String对象的生成以及编写Java代码时的一些常见的内存泄漏陷阱等等都会造成内存泄漏，但是要在开发阶段完成找出造成泄漏的代码是非常困难的。 在大型企业系统中，Java代码中的内存泄漏是常见而且难于解决的问题。这些泄漏问题通常是在最不愿意它发生的正式生产环境中发现的，而且它也很难于在开发与测试环境中得到重现。这是为什么呢？生产环境中的系统需要处理更大量的数据，而且有可能在运行很长时间后才会发现 Java堆在缓慢地增长。最终，导致系统内存耗尽。 因此本文介绍一种新工具BEA JRockit

Mission Control，用来诊断泄漏并指出根本原因。该工具的开销非常小，因此可以使用它来寻找生产环境中的系统的内存泄漏。 简介 BEA JRockit Mission Control（以下简称为JRMC）于2005年12月面世，并从JRockit R26.0.0版本开始捆绑了这个工具套件，目前最新的版本是2.0.1。它是一组以极低的开销来监控、管理和分析生产环境中的应用程序的工具。它包括三个独立的应用程序：内存泄漏监测器（Memory Leak Detector）、JVM运行时分析器（Runtime Analyzer）和管理控制台（Management Console）。 JRockit Management Console JRockit Management Console是一个基于JMX的控制台，用于监控和管理多个JRockit实例，提供至关重要的状态数据和控制JRockit JVM的运行时特性的方法。它捕获并显示关于垃圾收集器（GC）暂停、内存、堆使用和CPU负载的实时数据，以及部署在JVM内部MBean服务器上注册的所有JMX MBean所公开的信息。JVM管理包括对CPU相似性、垃圾收集策略和内存池大小的动态控制，还包括一个开销低的方法分析器和一个异常计数器。 JRockit Runtime Analyzer JRockit Runtime Analyzer（JRA）是一个JVM分析器，是一个随需应变的“ 动态记录器 ”Java应用程序，它记录了Java应用程序和JVM在一段预定的时间内的详细记录。然后通过JRA应用程序对记录下来的文件进行离线分析。所记录的数据包括对方法的调用跟踪、错误的同步、锁定的分析，还有垃圾收集统计信息，优化决策以及对象统计信息和其他重要的应用程序/JVM行为。它的目的是让JRockit开发人员能够找到良好的方法来基于现实应用程序优化JVM，对于帮助客户在生产和开发环境中解决问题十分有用。 JRA由两个部分组成：JVM中的记录引擎

和可以用于分析结果记录的GUI应用程序。记录引擎使用的信息源有几种，包括JRockit Hot Spot Detector（优化引擎也使用它来决定应该优化哪些方法）、操作系统、JRockit Memory System（最出名的就是垃圾收集器）和JRockit锁定分析器（如果支持的话）。 JRockit Memory Leak Detector 虽然Java的自动内存管理机制把开发人员从显式地分配和释放所使用内存的重担下解放出来，但如果程序继续引用不再有用的对象时，内存泄漏还是有可能发生。JRockit Memory Leak Detector工具用来发现和查找内存泄漏原因。趋势分析器为用户提供了一个趋势分析，可以发现非常缓慢的泄漏，显示详细的堆统计信息（包括指向泄漏对象和分配位置的引用类型和实例），可以说明应用程序中每个类使用堆空间的情况，显示某一类型的实例使用了多少空间、它们占用了堆的哪一部分、存在多少个实例以及每秒钟堆空间使用的增加速度（以字节为单位），并快速找出泄漏原因。使用先进的图形化表现技术，以便更容易定位和理解有时比较复杂的信息。 JRockit Memory Leak Detector还提供快速找出泄漏原因的手段。可以在趋势分析表中选择一个怀疑类型，所有具有指向选中类型的实例的类型都可以显示在一个图中。图形节点可以随意展开，用户可以回溯到导致引用的最终原因。类的实例可以被显示，指向一个选中实例的所有实例都可以在一张实例图中显示出来。可以跟踪某个类的所有分配情况。 