解决J2EE系统应用性能问题常用方法 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/253/2021_2022__E8_A7_A3_E5_86_B3J2EE_c104_253540.htm 性能问题的最明显表现是网页的响应时间变慢。 在J2EE系统中，经常体现有下面更为基本的症状： 应用服务器资源的使用情况 JVM堆的使用情况 系统资源的使用情况 数据库资源的使用情况 网络活动 这些现象表明J2EE应用依赖很多外部资源，并且是运行在一个层次化的执行模式的环境中： 由于Java虚拟机和应用服务器掩盖了操作系统和硬件的特性，所以在设计软件系统时，架构工程师更应该深刻理解整个操作环境。 在设计软件系统时，架构工程师应把性能和可扩展性放在首位，然后开始寻找容易解决的问题，反应时间缓慢通常的原因是访问数据库效率低和过多地调用远程对象和方法。接下来，架构工程师可继续寻找不明显的原因，例如算法的累积影响和不必要的开销。 现在市场上的各个J2EE应用服务器有很多配置项目。这里只简单介绍一些常见的性能优化配置项目。 很多应用服务器都有一些与J2EE规范有关的操作系统配置项目或非标准的特性，这可以提高系统性能。应该化时间来理解这些性能配置。 Java虚拟机堆和垃圾回收设置 任何Java应用的性能调整基础都涉及到堆的大小和垃圾回收设置。（这里主要讨论Sun HotSpor JVM）. 堆可分为三代，年轻的（新的），年老的和持久的。Hotspot JVM的内存基本配置包括最大堆大小，初始堆大小和年轻一代堆的大小。当配置最大堆大小时可参考下面一些指导： 最大大小应小于物理内存，避免虚存的页面调度。 需要减去其他进程使用的内存 在负载测试时进行优化 注意不要

将最大堆大小设置得过大。堆越大，内存中保存的对象越多。内存中对象越多，回收过程时间越长。 配置初试堆大小的一般性策略包括： 将初始大小设置为最大堆大小 将初始大小设置为最大堆大小的1/4到1/2 对于年轻一代堆大小，Sun 推荐是设置为最大堆大小的1/3。 也可以选择不同的垃圾回收算法。首先是增量垃圾回收。该算法的意思是减少单个对象回收停顿时间，这样的结果是整体回收性能的下降。该算法将相互引用的对象分组，然后尝试按组回收。尝试回收的部分越小，回收处理的时间往往会越少。 1.4.1版的HotSpot JVM增加了两个垃圾回收算法：并行算法和并发算法。 在年轻一代堆中实现了并行算法。在多处理器的机器上，这种回收算法使用了多线程来提高性能。虽然这个算法会暂停所有的应用线程，但是由于利用了多个CPU使得回收时间非常快。在年轻一代堆中，该算法显着地减少了回收带来的停顿。 在年老一代堆中实现了并发算法。在应用中最大限度地执行并发。回收过程分为4个阶段，覆盖了可回收对象的标记和清除操作。前两个过程会暂停应用线程，后两阶段可与应用并发执行。并发垃圾回收算法的"最大限度并发"特点可以使JVM利用更大的堆和多个CPU。因此应关注由于采用缺省的mark-compact(标记-压缩)和stop-the-world（停顿所有处理）等垃圾回收算法所带来的延迟和吞吐量问题。 J2EE应用服务器是多线程的应用。应用服务器的线程是一种资源池，处理请求和和应用服务器的内部功能等任务共享这些资源。 很多应用服务器允许为特定的任务或应用配置不同大小的线程池。通常需要增加这些线程池的大小以满足应用负载的需要。 架构工程师应该避免将线程池大小设置过大，这是因为会增加上下文交换的

次数，从而降低应用的性能。线程池的大小通常应该能最大利用机器上的CPU，同时又不能使CPU过载。 EJB配置项目 在应用服务器中，很多不同类型的EJB是以资源池的方式实现的。通常这些池大小和初始Bean的数量会明显影响应用的性能。 架构工程师应该避免将这些池大小设置的过大，这样会导致不必要地消耗JVM和操作系统内存。另外，将初始Bean数量设置过高会使得应用服务器的启动时间长的难以接受。 在应用服务器中，缓存很多不同类型的EJB。缓存大小和超时设置通常也会对应用性能带来显着影响。 架构工程师应该避免将缓寸大小设置过大，这同样会不必要地消耗大量JVM和操作系统内存。此外，应避免设置过长的超时--例如当EJB不用时，仍被缓存---，这也会导致不必要地消耗大量内存。 数据库配置项目 J2EE规范要求应用服务器厂商必须提供数据库连接资源池功能。通常增加数据库连接池的大小会提高性能。架构工程师应该考虑不同类型的SQL操作（例如事务型和批处理型）应使用不同的连接池。如果一个消息Bean执行批处理操作，那么应该为此另创建一个连接池，而不要与事务型操作使用同一个连接池。 很多J2EE应用服务器提供了Prepared Statement 的缓存功能。创建Prepared Statement是很耗费资源的。在事务型的J2EE应用中通常执行很多同样的SQL语句，只是参数不同而已。所以在应用中应发挥数据库配置项目的作用，尽量使用Prepared Statement。 