

基于会话外观模式的 EJB 性能调优

梁惠平¹, 吕述望²

(1. 中国科学院研究生院, 北京 100039;

2. 北京科技大学, 北京 100083)

摘要: 通过对影响 EJB 性能因素的分析, 给出了对 EJB 应用程序调优的策略和方法, 并详细说明了采用会话外观模式的 EJB 设计模式后对系统性能的影响。会话外观模式简化了 Web 容器和 EJB 容器之间的耦合, 减少了在网络边界上的调用过程, 同时实现了对安全权限和事务的集中管理。

关键词: EJB; 性能优化; 会话外观模式; 耦合

中图分类号: TP302.7

文献标识码: A

EJB performance optimization based on session appearance model

LIANG Hui Ping¹, LV Shu Wang²

(1. Graduate School of the Chinese Academy of Sciences, Beijing 100039, China;

(2. University of Science and Technology Beijing, Beijing 100083, China)

Abstract: The paper gives out the methods to optimize EJB applications by analyzing the items which affect EJB performance. And it mainly introduces the affect to system performance after adopting session appearance model. Session appearance method simplifies coupling between WEB container and EJB container, decreases process calls in net border, and implements centralized management to security right and business.

Key words: EJB; performance optimization; session appearance model; coupling

目前, 软件设计大多采用多层体系架构, 实现了应用层与用户层分离, 这种架构形式具有更强的通用性和兼容性, 提高了系统的性能和功能。

在多层体系架构的应用层中, 由于传统的 COM+ (Component Object Model) 组件技术的移植性和通用性比较有限, 一般都采用 J2EE 的关键技术——EJB 组件技术, 该技术已得到世界上大多数厂家的支持, 使系统的升级、开发过程中有更大的选择空间。

由于 EJB 组件技术作为 J2EE 系统技术的基础, 所以选择不同的 EJB 组件模型和设计模式, 在很大程度上可以影响 EJB 应用程序的整体性能, 通常整体性能是被很小的一部分的代码所决定。本文通过对影响 EJB 性能因素的分析, 给出了 EJB 应用程序调优的策略和方法。并详细说明了采用一种会话外观模式的 EJB 设计模式后对系统性能的影响。

1 影响 EJB 性能的因素

影响 EJB 应用程序性能的因素包括以下几点:

(1) 网络带宽和网络业务的数量: 如果网络带宽偏低而网络业务较高, 则应用服务器上的最优化的 EJB 应用程序也变得低效。

(2) 硬件和操作系统: 硬件和操作环境直接影响应用服务器的可伸缩性和性能, 从而影响 EJB 应用程序的性能。

(3) 应用服务器(和它们的配置)专有特性、可伸缩性的性能: 应用程序的运行也依赖于运行应用程序的应用服务器, 应用服务器的品牌、配置和设置都会影响应用服务器的性能和伸缩性, 从而影响 EJB 应用程序的性能和伸缩性。

(4) EJB 应用程序部署商为特定的应用服务器优化应用程序的知识和经验: 如果部署商没有为操作环境和应用程序服务器优化地设置、部署参数, 那么即使是最优化的 EJB 应用程序也会拙劣的执行。

(5) EJB 应用程序的设计和实现: 一般情况下, Bean 开发人员无法控制网络带宽、使用的硬件类型和应用服

技术与方法 Technique and Method

务器的品牌。但能够做的是:尽力地优化 EJB 以实现提高性能;分析装配参数的优化设置向装配商和部署商提供尽可能多的信息。

2 EJB 应用程序的设计决策

所谓 EJB 设计决策主要是如何选择 3 种 Bean: 实体 Bean、会话 Bean 和消息驱动 Bean; 如何选择 BMP 和 CMP; 如何选择本地接口和远程接口^[1,2,9]。

3 种 Bean 的选择: 如果事务中要编译和维护数据, 就要选择实体 Bean, 如果数据时间是数据的瞬像, 则可使用无状态的会话 Bean, 并从实体 Bean 中取得信息, 返回数值对象。会话 Bean 表示业务过程, 可以维护所有的业务逻辑, 包括数据验证。消息驱动 Bean 主要用于异步通信的过程中。

有状态的会话 Bean 和无状态的会话 Bean 的选择: 有状态的会话 Bean 需要更多的服务器资源, 它在方法调用之间维护数据。无状态的会话 Bean 需要更多的网络资源, 每次调用远程方法时, 需要的所有相关数据都要以参数的形式传入。

BMP 和 CMP 的选择: CMP 可以减少代码的编写量, 而且允许容器进行很多优化, 用容器实现持久性处理逻辑, 具有较强的移植性。而 BMP 中包含数据库访问代码, 可以更好地控制数据实现, 但额外的代码编写量大, 移植性差。

远程接口和本地接口的选择: 本地接口可以减少开销量, 允许本地组件进行本地通信, 在调用方法和参数中按引用对象传递, 本地接口的对象不需要支持 RMI-IIOP 协议, 传递速度快。远程接口的对象在调用中按数值传递, 提供了一个远程的、与位置无关的视图必须实现 RMI 接口, 造成较大的延迟。

3 调优 EJB 应用程序的策略

3.1 减少远程调用

和 EJB 的分布式结构有关的主要性能的关注点就是它所需要的网络通信量。当远程客户机对 EJB 进行调用时, 需要借助 RMI-IIOP 协议, 从而需要一定量的网络开销。其中包括客户机与容器之间通过网络传递数据的开销。减少网络开销的方法有两种: 一种是将会话 Bean 作为实体 Bean 的前端, 前端会话 Bean 通过对实体 Bean 的调用是本地方法调用, 从而可以避免大量网络通信的开销。另外这样可以表示逻辑与业务逻辑分离, 从而降低了 EJB 的复杂性。另一种是尽量使用本地接口, 使用引用传递参数避免了调用和解调用的开销, 从而减少了不必要的网络业务^[3,7,10]。

3.2 使用适当的 EJB 设计模式^[4,7,11]

设计模式可以认为是蓝图或模板解决方案, 这些解决方案可以容易地修改以解决在环境中重复发生的问题。设计模式的使用可以节省时间和精力。使用验证过的设计模式解决特定类型的问题。模式由以下元素组成:

(1) 模式名称: 简洁地表达了模式的本质。

(2) 背景: 问题发生的环境。

(3) 问题: 对待解决问题的细节描述。

(4) 用途: 有关形成模式的动机和基本原理的列表。

(5) 解决方案: 描述解决问题的方式, 用类图和顺序图解释它们的结构。

(6) 参与者及其职责: 模式中所有对象的列表。

EJB 设计模式分为: 数据访问对象模式、值对象模式、值对象装配器模式、值对象列表处理程序模式、服务定位器模式、会话外观模式、业务对象代表模式等 7 种模式。

可以根据不同的实际问题应用不同的模式。值对象模式通过减少对实体 Bean 的精细调用而降低潜在的网络通信量, 同时通过简单地访问 Bean 的字段值给客户端提供实体 Bean, 特别应用到远程客户端访问实体 Bean 的情况。

会话外观模式允许开发人员给客户端和表示层通过会话 Bean 提供大致的 EJB 事务逻辑访问。这种模式向客户提供了一个统一的访问层和接口, 负责定位、创建执行业务对象中的业务逻辑。使用会话 Bean 来封装和隐藏参与业务流程的业务对象之间交互的复杂性。

4 会话外观模式

采用 Grinder 测试工具测试开发的原型 EJB, Grinder 的默认行为是发布客户的 HTTP 请求, 然后模拟浏览器请求 Web 服务器, 通过建立 1 个轻便的分配器 Servlet, 将 Web 服务器上的请求给 EJB 容器。这样可以很好地模仿的现实中的情况, 如图 1 所示。

4.1 测试配置及测试脚本摘录

所有的测试在三台机器上运行: 一台运行 Grinder 及其线程, 另一台运行应用服务器, 第三台运行数据库服务器。测试在 1、5、10 个用户负载下运行, 通过对基准响应时间和总的事务率的测试结果, 表明使用会话外观模式的优越性。

在 Grinder 测试脚本中执行的请求是随着测试案例而变化的, 但本质上类似, 因为每个请求都是调用调度器 Servlet, 分析基于响应时间和总的事务率的结果^[5,6,8,10]。下面是基于会话外观测试模式的测试脚本摘录:

```
grinder.jvm.talk=base
#start/stop from Grinder Cart
grinder.receiveCartSignals=true
grinder.grinderAddress=192.168.0.1
grinder.grinderport=789
...
grinder.test1.parameter.url =http://sun2:5026/course/
Dispatcher
grinder.test1.parameter.header.Content =application/x-www-form-urlencoded
```

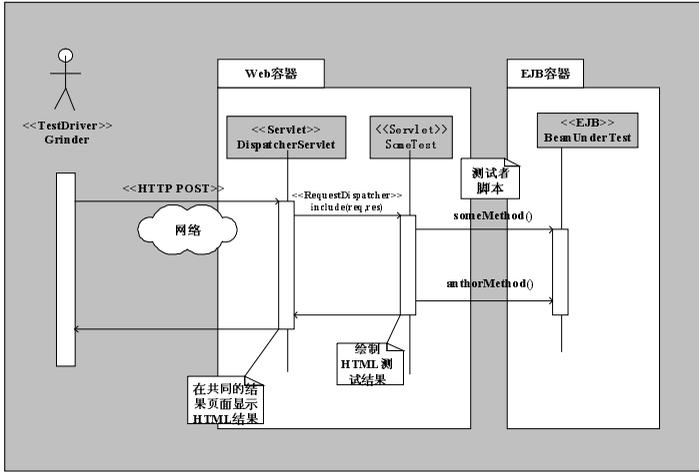


图1 Grinder 测试序列图

```
grinder.test.description=find a course title with "6a"
grinder.test.paramter.post=find-6a.dat
```

...

4.2 测试说明

定义容器管理的实体 Bean—CourseEJB，本文将对在 CourseEJB 前是否加入会话外观模式，说明会话外观模式的优势。因为 Web 组件对实体 Bean 进行多次远程调用时开销很大。会话外观模式使用无状态的会话 Bean 封装了 Web 组件需要执行的实体 Bean—CourseEJB 的业务逻辑^[2,4]。与图 2 所示的不使用会话外观模式 Facadeoff 序列图相比，图 3 所示会话外观模式使 Web 容器和 EJB

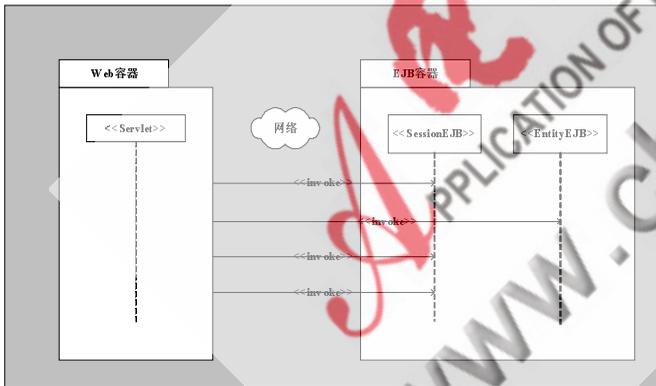


图2 不使用会话外观模式 FacadeOff 的序列图

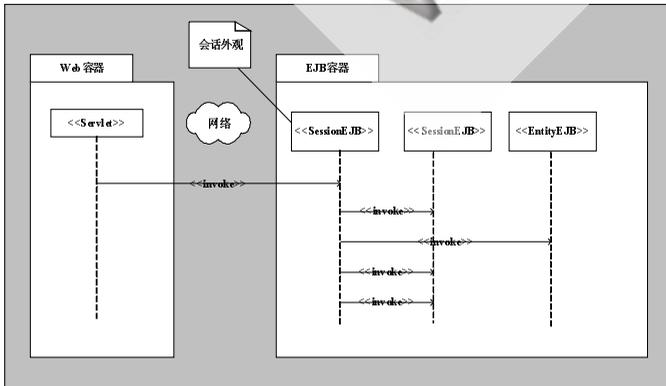


图3 使用会话外观模式 FacadeOn 的序列图

容器之间的耦合大大简化了，多个精细的远程实体存取器的调用变为较粗的远程调用。即所有实体方法的调用都封装在 1 个单一的会话 Bean 方法中：

```
public collection forCourseDescriptionLocal (String searchCourseID)
```

该方法的实现等同于在 FacadeOff 中的实体 Bean 方法调用序列。

可以把会话模式理解为在更大的粒度上实现的业务流程，在应用中，底层的企业 Bean 虽可以实现多种功能模块，但这些企业 Bean 的功能模块单独存在对于客户是没有意义的，客户需要的是一次完整的“操作”过程，如果在这个工程中客户对每一种方法都要依次调用，从网络的负荷、系统安全性及事务的集中管理等角度上都是不合理的。因此可以将“操作”作为一个会话模式封装具体的业务过程，客户对象在需要业务功能时只需要调用一次远程接口，极大减少了在网络边界上的调用过程，同时实现了对安全权限和事务的集中管理。

4.3 测试结果比较

(1) 响应时间对比结果如图 4 所示。

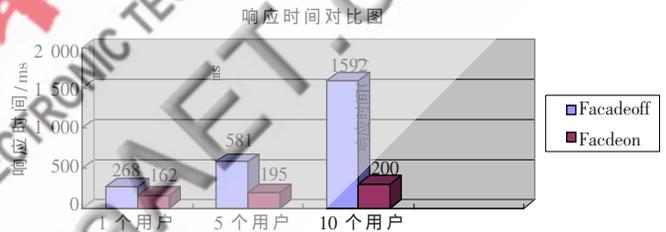


图4 响应时间对比

对于 1 个用户的负载，使用会话外观模式中使响应时间减少了 38%，而 10 个用户的负载使响应时间减少了 81%。

(2) 总的事务率对比结果如图 5 所示。

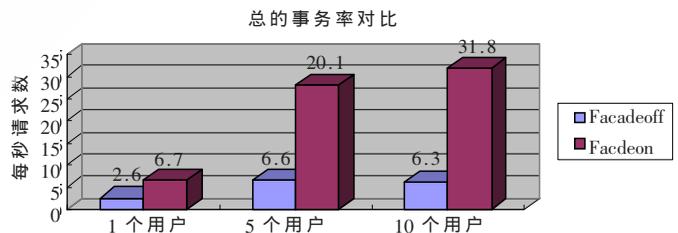


图5 总的事务率对比

这种影响是非常显著的，对于 1 个用户的吞吐量增加了 158%，10 个用户的吞吐量增加了 392%。

本文通过对影响 EJB 性能因素的分析，给出了对 EJB 应用程序调优的策略和方法。并详细说明了采用会话外观模式的 EJB 设计模式后对系统性能的影响。会话外观模式减少了在网络边界上的调用过程，同时实现了对安全权限和事务的集中管理，使 Web 容器和 EJB 容

技术与方法 Technique and Method

器之间的耦合大大简化了。随着应用负载的增加,从系统响应时间和总的事务率都凸现出会话外观模式的优势,在更大的粒度上实现了业务流程。

参考文献

- [1] 孙宏滨.基于 J2EE 三层结构分布系统的设计与实现[J].齐齐哈尔大学学报,2002,18(4):61.
- [2] 陈华军.J2EE 构建企业级应用解决方案[M].北京:人民邮电出版社,2002.
- [3] 班书昊.EJB 企业应用与开发实例[M].北京:北京科海电子出版社,2003.
- [4] 梁刚.面向企业信息系统的 J2EE[J].黑龙江通信技术,2002(1):38.
- [5] CAVANESS Chuck, Special edition using enterprise java beans[D].SUN 公司核心技术丛书 2002:22-27.
- [6] 袁国勇.EJB 组件开发模型及实现技术[J].应用技术,2002(12):22-23.
- [7] 高奇微.分布式中间层 EJB 的研究与开发[J].中国农业大学学报,2001,6(4):88.
- [8] 马廷淮.用 EJB 开发 J2EE 应用[J].计算机应用,2002,22(4):109-110.
- [9] 彭宏庆.基于 EJB 的多层分布式系统模型的研究与应用[J].航空计算技术,2002,32(4):106.
- [10] 高振东.基于 J2EE 系统的 EJB 技术研究与应用[J].航空计算技术,2003,33(1):71.
- [11] 段江.使用 EJB 组件技术开发多层应用[J].计算机辅助工程,2003(4):29-30.

(收稿日期:2009-05-11)

