

# J2EE 分布式应用软件架构性能比较与选择

## Performance Comparison and Selection of Software Architectures for J2EE distributed Application

陈晓玲 吴少刚 周涛

(中国石油大学(华东)计算机与通信工程学院 山东东营 257061)

**摘要:**软件架构的选择是件复杂的事情,需要考虑多方面的因素。针对一个典型的 J2EE 分布式应用——网上购物系统,实现了两种逻辑架构的应用程序:Web 实现和 Web + EJB 实现。通过在单机和集群环境中部署应用进行测试,在性能方面比较了它们的优劣。最后,对软件架构的选择做了进一步的讨论。

**关键词:**逻辑架构 物理架构 Web 实现 Web + EJB 实现 用户访问模式 集群 配置结构

### 1 引言

随着计算机技术的飞速发展和 JAVA 技术应用的日益成熟,J2EE 平台已经成为设计开发企业应用最流行的技术。

J2EE 平台提供了一个多层的分布式应用程序模型,应用程序的逻辑根据其实现的不同功能被封装到组件中,这些组件根据其所属的层被安装到不同的机器中。除客户层外,一个典型的 J2EE 应用可以分为三层:Web 层(也称表示层)、商业逻辑层(即 EJB 层)和数据持久层。J2EE 应用服务器一般由 Web 容器和 EJB 容器组成。Web 容器管理诸如 Servlet、JSP 等 Web 组件的运行。这些组件向终端用户展现业务数据,这些数据可由业务逻辑层提供,也可以是 Web 组件直接访问数据持久层获得。EJB 容器负责管理 J2EE 应用程序中 Enterprise Bean 的执行。EJB 组件提供数据持久化机制和事务支持。

软件设计是一件复杂的事情,架构选择尤其重要。根据关注角度的不同,软件架构可分为逻辑架构和物理架构。逻辑架构指应用系统中组件之间的关系和逻辑层次,如一个 J2EE 分布式应用可分为 Web 层、业务逻辑层和数据持久层,分别包含了用户界面、商业逻辑元件、数据库及外部系统接口等各种组件,这便是逻辑架构。物理架构指应用组件在硬件上的部署和分布状况,如系统是运行在单机环境或集群中,Web 组件和 EJB 组件是运行在同一 JVM 中还是分布式部署。除此之外,架构选择还要考虑到其他非功能性特征,如可扩

展性、可靠性、强壮性、灵活性等。

### 2 分布式应用——网上购物系统

网上购物系统<sup>[1]</sup>是一个简单的电子商务应用系统,实现了用户注册、登录、书目浏览、商品订购、修改订单等基本功能。它有两种类型的交互操作,一种是只读操作,另一种是读写操作。只读操作包括主页、新产品列表、特定商品的详细信息等。读写操作包括用户注册、用户登陆、更新购物车状态、用户定单的查询显示等。

我们不使用任何形式的 entity bean。生产实践和相关研究<sup>[2]</sup>表明,entity bean 对系统性能影响很大,这导致了两个结果:Local EJB 标准和 Session facade 模式。但是即使使用 Session facade 模式结合 local entity bean,系统性能的提升仍然很有限。

我们实现了两种逻辑架构的网上购物系统。

#### 2.1 Web 实现

应用程序使用基于 Servlet \ JSP 等 Web 层组件的实现所有的业务逻辑和表示逻辑。用户的 session 信息使用 HttpSession 保存在 Web 层。Servlet 使用 JNDI 通过 JDBC 访问数据库。系统模型如图 1。

#### 2.2 Web + EJB 实现

应用程序基于 Servlet \ JSP + Session Beans 的实现。它包含 Web 层、EJB 层和数据持久层三层结构。使用 Session Beans 实现大部分的业务逻辑,Web 层 JSP \ Servlet 组件实现表示逻辑和小部分业务逻辑。用

户的 session 信息使用有状态 Session Bean 保存在商业逻辑层。采用有状态 Session Bean 的效益主要在于 (1) EJB 容器提供连接池和事务处理的支持; (2) 更符合面向对象设计, 可以节省某些 "Context Object" 这样的 Web 层设计模式和由此而来的每个 Web 请求重构 Context Object 的性能代价; (3) 支持 Web 层以外的外部客户。

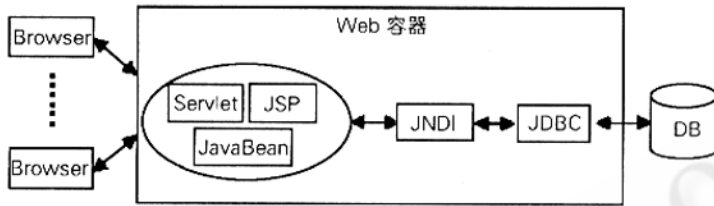


图 1 Web 层实现的逻辑架构

### 3 实验环境

#### 3.1 测试计划

应用程序的系统性能除程序本身外, 还依赖于组件分布和用户负载。用户所能感知的系统性能在很大程度上由用户的操作行为所决定。不同类型的用户可能会执行不同的操作, 因此, 当一个用户请求被执行时可能会调用不同的组件服务。我们引入用户访问模式<sup>[3]</sup>, 它反映了一种典型的用户行为。

我们将性能测试的用户分为两种访问模式: 浏览者和购买者。浏览者模式的用户仅仅浏览网站信息, 或是搜索自己感兴趣的商品或是没有特定目的的查看信息, 浏览者既不用注册登录, 也不会购买商品。浏览者的访问由一组先后相互关联的请求组成, 如下所示:

- 进入书店主页;
- 显示并查看特定种类的商品列表;
- 显示并查看特定子分类的商品列表选择一本书获得该书的详细资料;
- 显示并查看特定商品的详细信息, 如产品描述、价格、库存数、产地、商品编号等;
- 按照关键字搜索商品, 显示产品列表。

购买者模式的用户知道自己所要购买的商品, 登录后才能进行相关的购物操作, 测试请求如下:

- 进入书店主页;
- 用户登录书店;
- 显示并查看特定种类的商品列表;

- 显示并查看特定子分类的商品列表;
- 选择需要购买的商品将其放入购物车中;
- 用户结账;
- 用户确认订单;
- 用户登出系统。

#### 3.2 硬件

- Intel (R) Pentium (R) 4 CPU 2.93GHz x1
- 应用服务器内存 768MB

#### 3.3 软件

- 操作系统 WindowsXP SP2
- JVM Sun JDK 1.5.0 Update 6
- 客户模拟器 JMeter
- 负载均衡器 Apache + mod\_jk
- 应用服务器 JBoss3.2.6
- 数据库 MySQL5.0.18

### 4 实验结果与性能比较

#### 4.1 单机环境

单机环境中模拟浏览者用户, 结果如图 2。可以看出, Web 实现 (图中 Servlet \ JSP 所表示的曲线) 在 160 个用户时吞吐量达到峰值 165 rps。相比而言, Web + EJB 实现 (图中 S + SB 所表示的曲线) 约在 120 个用户时获得最大吞吐量 124 rps, 二者在性能上存在明显差距。

模拟购买者用户结果如图 3, Web 层实现 (图中 Servlet \ JSP 所表示的曲线) 大约在 140 个用户时达到最大吞吐量 142 rps, Web + EJB 实现 (图中 S + SB 所表示的曲线) 在 110 个用户时大致达到吞吐量峰值 104 rps, 性能上二者存在明显差距。

这两种模式下, Web 实现与 Web + EJB 实现存在明显性能差距的原因, 主要是因为 Web 组件和 EJB 组件之间以及 EJB 组件之间使用 RMI 通信。为了避免由于 RMI 所带来的通信消耗, 我们使用 Local Session Beans (图中 S + LSB 所表示的曲线) 替换 Web + EJB 实现中相应的 Session Beans, 系统性能有较大提升, 但和 Web 实现相比仍有一定差距。需要注意的是, Local EJB 组件不能被集群。

比较图 2 和图 3 我们还不难发现, 无论何种实现方式, 当用户访问模式不同时服务器的压力也不同。浏览者模式的用户由于很少涉及状态和访问数据持久层, 因此与之相对应的应用服务器的压力要小于购买

者模式的用户,系统吞吐量的峰值比购买者高约 16%,且吞吐量达到稳定峰值时的并发用户数也多于购买者模式。

实验中,集群<sup>[5]</sup>包含 1 台负载均衡器和 2 台应用服务器,并对服务器进行了优化。测试时并发用户数设定为 50(初步测试后确定并发用户数为 50,此时基本上保证吞吐量最大化而服务器又不会过载,测试错误率低于 1%)。

实验结果如表 1,此时 Web 实现和 Web + EJB 实现性能大致相当。单就性能而言,使用有状态 Session Bean 并不会比使用 HttpSession 更好,这和文献<sup>[6]</sup>中的结果基本一致。

表 1 集群环境购买者模式性能比较

应用程序实现方式	平均响应时间 (ms)	吞吐量 (rps)
Web 实现	637	43.9
Web + EJB 实现	669	40.5

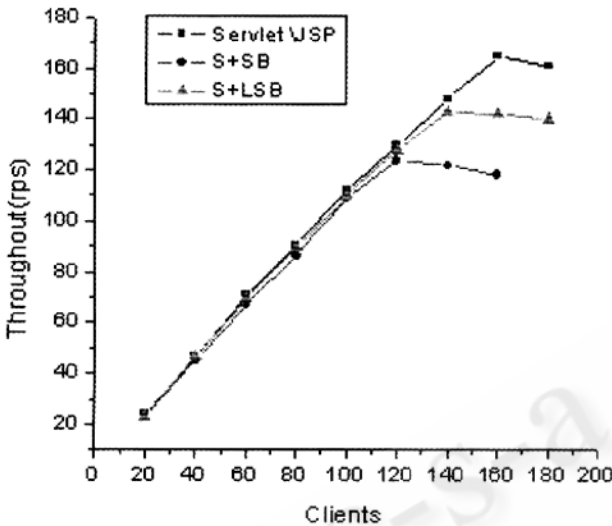


图 2 性能比较——浏览者模式吞吐量

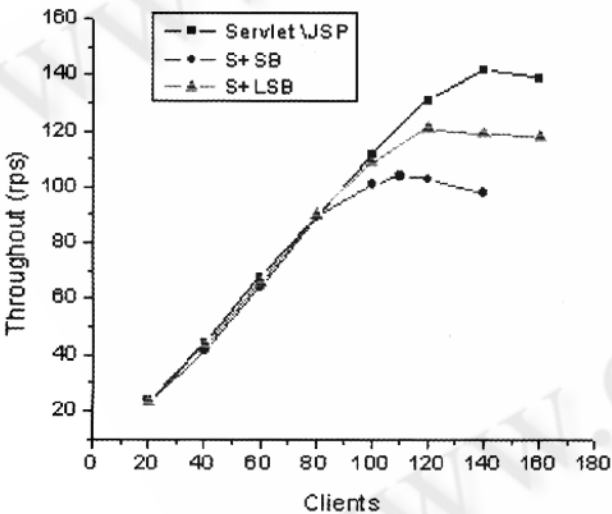


图 3 性能比较——购买者模式吞吐量

#### 4.2 集群环境

在 J2EE 框架中, JSP、Servlet、JDBC、EJB、JNDI 等对象能够被集群。J2EE 集群<sup>[4]</sup>主要包括 Web 层集群和 EJB 集群,其实质是负载均衡和故障转移技术的实现。实现故障转移需要进行状态复制,为此造成的系统资源消耗和性能退化不可忽略。

浏览者只浏览相关信息没有购买行为,不需要进行状态复制,所以集群环境性能测试只模拟购买者。

#### 5 架构选择的进一步讨论

根据关注角度的不同,软件架构可分为逻辑架构和物理架构。

从逻辑架构看,通常使用 Servlet \JSP 等 Web 层组件的 Web 系统应用在系统的逻辑层次划分上不如 Web + EJB 应用清晰明确,业务逻辑和表示层逻辑容易纠缠在一起。这种架构比较适合中小规模的企业应用,对大规模的系统应用则不太适合。值得一提的是,目前比较流行的 struts + hibernate 或 struts + spring + hibernate 设计也都属于 Web 系统应用。

从物理架构看,J2EE 集群可分别在 Web 层、EJB 层以及数据持久层进行集群,Web 应用的架构没有 Web + EJB 应用的架构灵活。而且,和有状态 Session Bean 相比,HttpSession 在集群环境中存在一些限制,如存储在 HttpSession 中的所有对象必须是序列化的 (serializable),这样会限制应用程序的设计和结构。

J2EE 技术,特别是 EJB,是为分布式计算而诞生的。减弱事务的功能性、可复用的 remote 组件使多层应用变得流行起来,但我们不必把什么都实现分布式。应当注意的是,一般情况下,Web + EJB 应用中使用的 Web 容器和 EJB 容器分布在同一个应用服务器中(即同一 JVM 中),被称为配置结构,同一台应用服务器中的 EJB 容器如果失效,WEB 容器也必定失效。在配置结构中,无论是单机还是集群,EJB 的故障转移只有在

客户继续存在的情况下才能发生,作为客户的 WEB 已经无法正常工作,故障转移一个 EJB 呼叫是没有意义的。加之性能因素等,显然在配置结构中,Web + EJB 应用使用 Local EJB 取代相应的 EJB 更加合理。

虽然配置结构应用比较广泛,但在某些情况下,分布式架构(Web 容器和 EJB 容器分离)和 Web + EJB 应用仍然不可替代,如(1)除 Web 容器外,EJB 还要为 J2SE、J2ME 之类“瘦客户”提供服务;(2)EJB 组件和 Web 组件处于不同的安全级别,在物理实现上需要分离;(3)如果系统 Web 层和 EJB 层极度不均衡,使用分布式架构更合理。

此外,由于状态处理会消耗较多的系统资源,像 eBay 这样需要处理大量交易的系统采用完全“无状态”的架构<sup>[7]</sup>对具有类似功能的应用系统很值得借鉴。

## 6 结束语

软件设计中,架构的选择是件复杂的事情,既要考虑软件开发的难度和软件当前适用规模,同时还要考虑软件的可维护性、性能以及部署环境等。以一个网上购物系统为例,实现了功能相同两种逻辑结构的应用程序:Web 实现和 Web + EJB 实现。通过在单机和集群环境部署应用进行测试,在性能方面比较了它们的优劣,并对软件架构的选择做了进一步的讨论。

### 参考文献

1 耿祥义、张跃平, JSP 实用教程[M], 北京:清华大学

出版社, 2003.

- 2 E. Cecchet, J. Marguerite, and W. Zwaenepoel. Performance and scalability of EJB applications[C]. In Proceedings of the 17th ACM Conference on Object - Oriented Programming, Systems, Languages and Applications (OOPSLA), November 2002.
- 3 D. Llambiri, A. Totok, and V. Karamcheti. Efficiently Distributing Component - Based Applications Across Wide - Area Environments[C]. Proceedings of the International Conference on Distributed Computing Systems (ICDCS2003), Providence, RI, May 2003.
- 4 Wang Yu. Uncover the Hood of J2EE Clustering[EB/OL]. <http://www.theserverside.com/articles/article.tss?l=J2EEClustering>, 2005.
- 5 Sacha Labourey and Bill Burke. JBoss AS Cluster - Ing (Edition 7) [EB/OL]. <http://docs.jboss.org/jbossas/clustering/JBossClustering7.pdf>, 2004 - 05 - 21.
- 6 Peter Zadrozny. HTTP Session Object vs Stateful EJB [EB/OL]. <http://dev2dev.bea.com/pub/a/2002/10/zadrozny.html>, 2002 - 10 - 23.
- 7 Ceperez. Nuggets of Wisdom from eBay's Architecture [EB/OL]. <http://www.manageability.org/blog/stuff/about-ebays-architecture>, 2004 - 06 - 21.