

---

上次修改时间：2001 年 6 月 6 日

# ECperf<sup>TM</sup>规范

---

I 版本 1.0, 最终版本

太阳微系统公司

发送评论 至 : [ecperf\\_feedback@eng.sun.com](mailto:ecperf_feedback@eng.sun.com)



---

ECperf 规范 (“规范”)

版本: 1.0

状态: 食品接触鑫

发行日期: 2001年6月6日

版权所有 2001 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303, U.S.A.保留所有权利。

#### 通知

本规范受版权保护, 其中描述的信息可能受一项或多项美国专利、外国专利或未决申请的保护。除以下许可另有规定外, 及其许可方 (如有) 事先书面授权, 不得以任何方式以任何形式复制本规范的任何部分。对本规范及其中描述的信息的任 Sun 网站法律条款中规定的出口控制和一般条款的约束。查看、下载或以其他方式复制规范, 即表示您同意您已阅读、理解此处规定的所有条款和条件。

Sun 特此向您提供全额支付、非排他性、不可转让、全球性的有限许可 (无再许可权), 根据 Sun 的知识产权, 这些知识产权对实践规范至关重要, 仅在内部实践规范或创建规范的洁净室实施, 该规范: (i) 包括当前规范没有子集或超集; (ii) 实现 Sun 定义的规范的所有接口和功能, 没有子集或超集; (iii) 包括您选择实现的任何可选组件 (在规范中定义) 的完整实现, 没有子集或超集; (iv) 实现此类可选组件的所有接口和功能, 不进行子集或超集; (v) 不会向“java.\*”或“javax.\*”软件包或子软件包 (或 Sun 定义的其他软件包) 添加任何其他软件包、类或接口; (vi) 在洁净室 Sun 提供的与最新发布规范版本相关的所有测试要求; (vii) 不源自任何 Sun 源代码或二进制代码材料; 以及 (viii) 不包含源代码或二进制代码材料。这

规范包含 Sun 的专有信息, 只能根据此处规定的许可条款使用。如果您未能遵守本许可的任何规定, 本许可终止或到期后, 您必须停止使用或销毁本规范。

#### 商标

本协议未授予 Sun 或 Sun 许可方的任何商标、服务标记或商号的任何权利、所有权或利益。Sun、Sun Microsystems、Sun JavaBeans、EJB、JDBC、Java Naming and Directory Interface、“Write One Run Anywhere”、Java ServerPages、JDK、JavaBeans、Sun Microsystems, Inc. 在美国和其他国家/地区的商标或注册商标。

#### 免责声明

规范按“原样”提供, 是实验性的, 可能包含无法或不会纠正的缺陷或缺陷

太阳。SUN 不作 任何明示或明示或保证的陈述或保证  
暗示, 包括但不限于适销性保证,  
适用于特定目的, 或不侵犯  
规格的帐篷适用于任何目的或任何  
PRACTICE 或实施此类内容不会侵犯任何内容  
第三方专利、版权、商业秘密或其他权利。本文档不代表在任何产品中发布或实施规范的  
任何部分的任何承诺。

SPecification 可能包括技术错误或排版-  
错误。定期将更改添加到信息中  
其中; 这些更改将合并到规范的新版本 (如果有)。SUN  
可能会对以下方面进行改进和/或更改  
规格中描述的产品和/或程序  
任何时间。对规范中此类更改的任何使用都将受适用版本的规范的当时许可的约束。

责任限制  
在法律未禁止的范围内, SUN或其许可方在任何情况下均不对任何损害负责, 包括但不限于  
损失  
收入、利润或数据, 或特殊、间接、后果性、信息信息  
牙科或惩罚性损害赔偿, 无论其原因如何, 也无论  
责任理论, 因任何提供、实践、修改或任何使用本细则而引起或与之相关的责任理论, 即  
使 SUN 和/或其许可方已被告知此类损害的可能性。

对于因您将本规范用于内部评估以外的任何目的而提出的任何索赔, 以及因向您提供的任  
何规范的更高版本或发行版与根据本许可提供给您的规范不兼容的任何索赔, 您将对 Sun  
及其许可方进行赔偿、使其免受损害并为其辩护。

#### 限制权利图例

美国政府: 如果本规范由美国政府或代表美国政府或美国政府主承包商或分包商 (任何级  
别) 获得, 则政府对软件和随附文档的权利应仅与本许可证中规定的权利一致; 这是雅高与  
48 C.F.R. 227.7202-4 (用于国防部 (DOD) 采购) 和 48 C.F.R. 2.101 和  
12.212 (用于非国防部采购) 的规定。

#### 报告

您可能希望报告您在评估规范时可能发现的任何歧义、不一致或不准确之处 (“反馈”)。  
如果您向 Sun 提供任何反馈, 您特此: (i)  
同意此类反馈是在非专有和非机密的基础上提供的, 并且 (ii) 授予 Sun  
永久、非排他性、全额支付、不可撤销的许可, 并有权通过多个级别的分许可证持有人进  
行再许可, 以合并、披露和使用反馈, 但不限于与规范和未来版本相关的任何目的,  
实现及其测试套件。

(LFI#90427/表格编号#011801)

---

版本 1.0, 最终版本, 2001 年 6 月 6 日



第1条 - 引言	7
1.1 引言	7
1.2 业务和应用环境	8
第2条 - 应用程序设计	13
2.1 客户域模型	13
2.2 制造领域模型	14
2.3 供应商领域模型	15
2.4 企业域名模型	16
2.5 数据库模型	17
2.6 企业数据库	17
2.7 订单数据库	20
2.8 制造数据库	21
2.9 供应商数据库	23
第 3 条 - 工作负载描述	26
3.1 术语定义	26
3.2 客户领域的业务	27
3.3 制造领域的EC交易	30
3.4 企业领域的EC交易	33
3.5 供应商域中的 EC交易	34
3.6 供应商模拟器	35
3.7 订单输入 应用程序	36
3.8 制造应用	38
第 4 条 - 扩展和运行规则	40
4.1 术语定义	40
4.2 商业产品要求	40
4.3 扩展 基准	41
4.4 数据库要求	44
4.5 Bean 部署需求	44

4.6 订单输入 驱动程序要求	45
4.7 制造驱动程序要求	47
4.8 ECperf指标的计算	47
4.9 测量间隔要求	48
4.10所需报告	48
4.11交易属性要求	50
4.12驱动程序规则	52
4.13供应商模拟器规则	52
4.14基准优化规则	52
第 5 条 - SUT 配置	54
5.1 被测系统（SUT）要求	54
5.2 配置示例	54
第6条 - 定价	57
6.1 性价比指标	57

版本 1.0，最终版本，2001 年 5 月 29 日

6.2 定价组件	6457
	64
6.3 定价规则	6558
	65
第 7 条 - 全面披露	6760
	67
7.1 术语的定义	6760
	67
7.2 一般充分披露要求	6760
	67
7.3 总和	68 玛丽声明
	6861
	68



7.4 第 4 条 伸缩和运行规则相关条目	6861
	68
7.5 第 5 条 SUT 和驱动程序相关项目	6962
	69
7.6 第 6 条 定价相关项目	7063
	70
第 8 条 - 结果的处理	7164
	71
8.1 结果提交	7164
	71
8.2 公布结果中的违规行为	7265
	72
8.3 撤回结果	7265
	72
8.4 合理使用 ECperf	72 结果
	7265
	72
附录 A - Bean 接口定义	7367
	73
附录 B 摘要说明样本	8685
	86
附录 C UML 图	8989
	89

第 1 条 - 引言

1.1 引言

ECperf 是一个 *Enterprise JavaBeans (EJB)™* 基准测试，旨在衡量 J2EE 服务器和容器的可伸缩性和性能。

从根本上说，Enterprise JavaBeans 是一个用于构建可扩展的分布式应用程序的基础架构，这些应用程序是使用面向 component 的对象事务中间件（OTM）实现的。随着需求的增加，Enterprise JavaBeans 应用程序需要自动扩展。因此，如果要真正帮助提高实际客户看到的性能，ECperf 工作负载应具有表 1 中所示的特征。

表 1.ECperf 工作负载的黄金法则

黄金法则	特征	描述	
展示企业组件	完全分布式	业务问题应该需要使用全球服务和数据，其确切位置是先验的。	
	冗余和可用服务	服务应该是冗余的，并且完全可用，运行时映射对应用程序开发人员来说是未知的。	
	中间件焦点	尽量强调中间层（无论是逻辑层还是物理层），而不是客户端层或数据库服务器层。	
	可扩展的业务领域	随着建模业务规模的增长，服务、数据、用户和地理位置分散的网站数量也应随之增长。	
	普遍性	只要存在兼容的容器，企业就应该能够部署在任何供应商的 H/W 和 O/S 上。	Bean
命令可信度	真实世界	性能工作负载应具有实际系统的特征。	
	复杂性	工作负载应捕获公司内部、外部和公司间业务流程。	
	开放	工作负载应该可以在所有流行的全功能 EJB 应用程序服务器上实现。	

中立            没有选择工作负载功能，以使任何 特定产品  
看起来好（或坏）。忠实于现实世界的业务需求至关重要。

表 1.ECperf 工作负载的黄金法则

黄金法则	描述
符合 辐射/集成开 程序员发应用 开发 工艺	工作负载应像由客户使用用于标准化部署的可重用组件一样进行开发。
范围开发技 能	该实现应假定财富 500 强开发人员了解业务逻辑，而不是系统编程。
模式规单纯 格 最佳行 业 实践	工作负载应易于理解，易于实施，并在合理的时间内运行，并具有出色的可重复性。
行业标准 基准	工作负载 应考虑 运行它的系统的价格。

GUI 和表示不是此工作负载的重点，也不是日常 DBMS 可伸缩性的各个方面（例如，数据库 I/O、并发性、内存管理等）。其他标准工作负载（如 TPC-C、TPC-D、TPC-W 等）充分强调了这些压力。ECperf 强调 EJB 容器处理内存管理、连接池、钝化/激活、高速缓存等复杂性的能力。

**注释：**虽然为了便于阅读，注释与正文分开，但注释是标准的一部分，并被强制执行。但是，作为附录B包含的摘要陈述仅作为示例提供，并且明确不是本标准的一部分。

**注意：**规范中具有此类描述的符号，但不属于当前标准的一部分。通过将措辞保留在文档中，可以更轻松地扩展未来版本中的工作负载。

## 1.2                      业务和应用环境

出于兴趣、范围和家庭性的原因，我们将 制造、供应链管理和订单/库存 作为业务问题的“故事情节”。这是一个多肉的、工业实力分散的问题。它是重量级的、任务关键型、全球性的 7x24，并且需要强大的、可扩展的基础架构。这是许多财富 500 强公司感兴趣的。最重要的是，它需要使用有趣的 Enterprise JavaBeans 服务，包括：

- 事务组件
  - 分布式事务
  - 消息传递和异步任务管理
  - 具有多站点服务器的多个公司服务提供商
  - 旧版应用程序的接口和使用
  - 安全传输
  - 基于角色的身份验证
- 
- 对象持久性

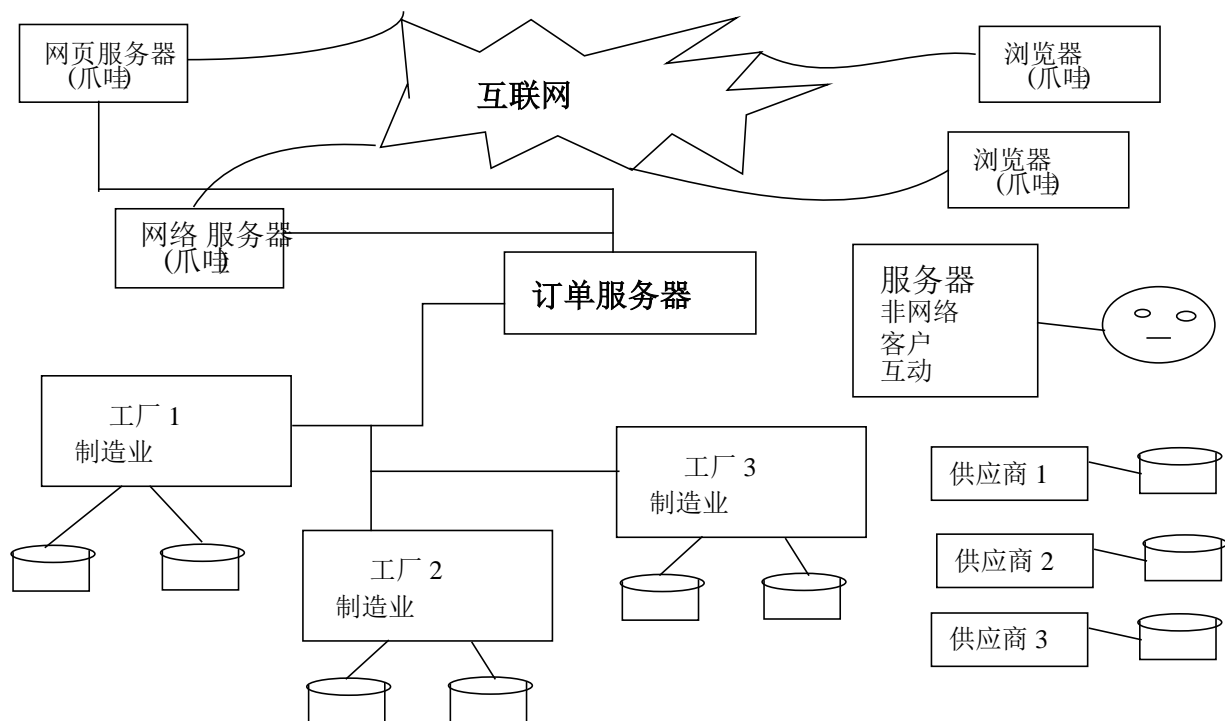


图 1:

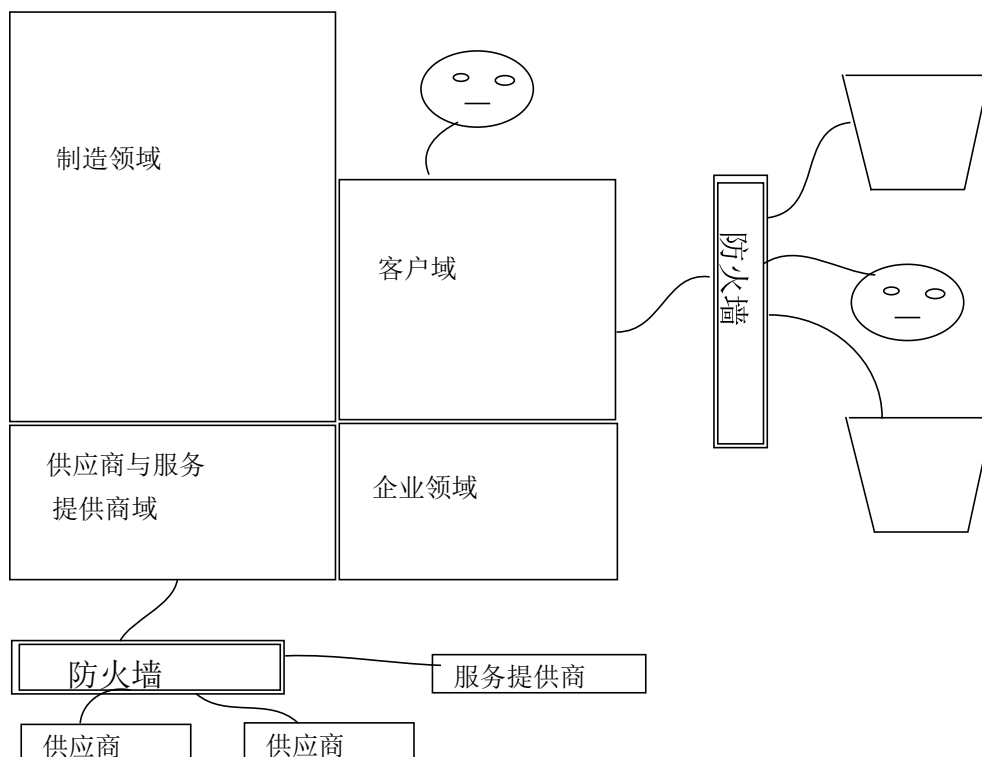
### EJB 全球 分布式业务

企业积极使用网络，允许客户直接指定产品配置、订购和状态检查。这些企业努力实现制造、库存、供应链管理和客户计费的完全自动化。图 1 描述了这种业务的本质。

他们使用“准时制”制造理念，寻求最高的效率和最少的库存。他们的客户和d  
业务流程同时运行，是事务性的、安全的、全球性的、24x7  
的，并且完全分布式。像这样的公司希望他们的应用程序开发人员专注于业务逻辑，而不  
是可扩展事务的复杂性和微妙之处，messaging，目录/命名，安全性，身份验证或服务到特  
定硬件的映射。这些企业更喜欢使用基于可扩展、可扩展、分布式组件的 RAD/IDE  
基础架构进行开发。

图 2 描述了与图 1  
相同的业务，但显示了逻辑域。术语“域”是指描述不同的业务运营领域的逻辑实体。

建模的四个ECperf域是：制造，供应商和服务提供商，客户和企业。在所建模大小的计算  
中，假定每个域都有单独的数据库和应用程序。最有可能的是，它们也是在单独的计算硬  
件上实现的。公司内部的领域之间以及与外部供应商和客户之间存在生产者-  
消费者关系。由于历史原因，每个域可能具有不同的实体 ID（即，客户域中使用的客户  
ID 可能与供应商域中用于同一客户的 ID  
不同）。正是出于这个原因，全球客户或供应商数据库很可能存在于公司领域，并附带应  
用程序。



---

**图 2: 全球多域业务**

分布式数据库和应用程序为 ECperf 工作负载奠定了基础。客户（包括分销商、政府和 OEM）通过多种方式联系企业，包括直接通过 Web 联系。个人对个人的联系，例如通过电话或通过销售人员，也映射到此 Web 访问方案，因为客户服务代表或销售人员自己使用 Web 界面。全球所有办事处和工厂都经常访问其他办事处或工厂的数据，并且有时必须将其数据与全球持有的数据进行比较/整理/验证。该公司还与完全独立的供应商公司互动。每个供应商都有自己独立的计算资源集。

企业业务是成功的。它通过多种不同的方式生长。这些包括简单的增长以及多元化和公司合并。第一种方法，增长，简单地意味着公司获得更多客户并销售更多产品。它的工厂越来越大，而且越来越多。企业需要各种供应商;它们也会成长。

公司还可以通过多元化或合并来实现增长。在这两种情况下，现在是一家企业集团的公司可能会承担全新的业务线。它建立了新的division，每个愿景都有其可能不同的客户和供应商。新部门的客户、制造工厂和供应商的联合称为**生产组（PG）**。图 3

显示了公司经历了一个增长阶段后的状态。随着PG数量的增加，现有PG的大小越来越大，最新的PG最小。单个公司域充当各个PG之间的协调点。这代表了在现实世界中发现的非线性缩放model。

**注意：** 1.0 版仅限于单个生产组。有关此 PG 中的详细缩放信息，请参阅第 4 条。

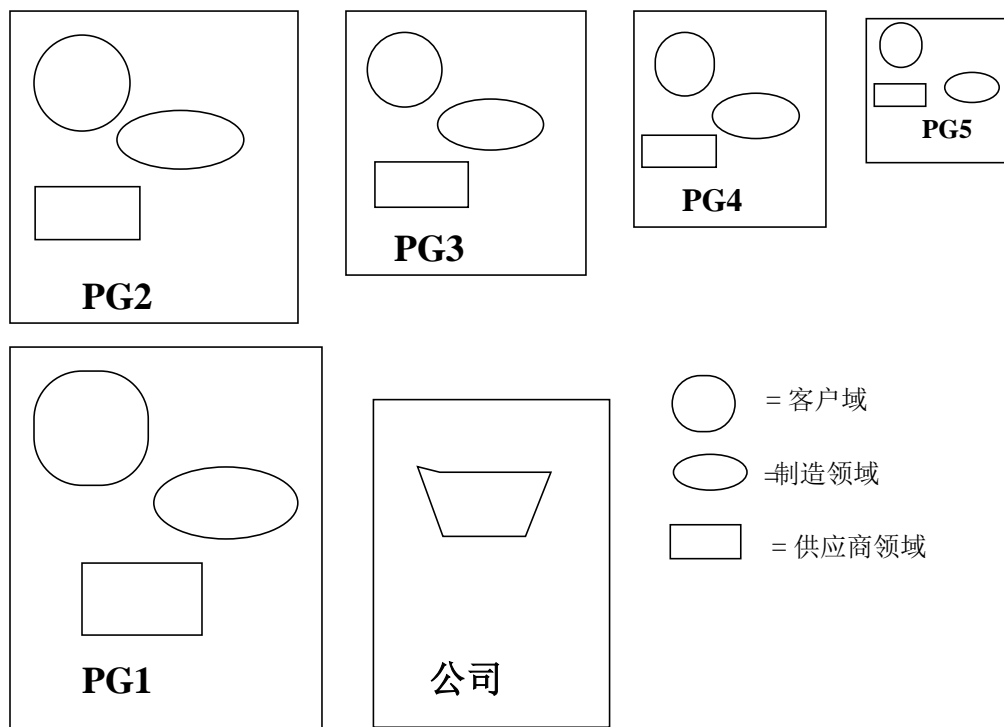


图 3:

### 扩展分布式全球业务

企业业务是全球性的。它的制造、组装或分销工厂分散在许多城市。每个位置都有多台计算机，每台计算机可能是应用程序服务器、数据库服务器、连接多路复用器或其他什么。

最终用户可能拥有台式机、桌面等。有许多不同的计算机为这项业务做许多不同类型的事情也就不足为奇了。有些计算机只提供单一服务（例如，包含单个 **vital** 数据库），而其他计算机则整合了许多服务。某些计算机可能配置为高可用性（H/A）配置，以便在节点发生故障时服务可以继续。另一方面，成本因素可能决定并非所有企业的计算机都可以进行故障转移。相反，业务可以配置映射到独立计算机的冗余服务，将其留给运行时环境来查找该业务服务的提供程序。

#### ECperf

业务由客户、工厂（制造、组装或分销）和供应商组成。该业务是复杂的，分布式的，并且使用许多以各种拓扑排列的独立计算机。它提出了大量有趣的方法来强调应用程序 **servers** 和中间件。



## 第 2 条 - 应用程序设计

### 2.1 客户域模型

客户域中的工作本质上是 OLTP。OrderEntry 应用程序在此域中运行，其功能包括添加新订单、更改现有订单以及检索特定订单或特定客户的所有订单的状态。

订单由个人客户和分销商下达。两者之间的区别在于订购的商品数量。大约 57% 的系统工作与分销商的订单（即包含大量项目）有关，43% 来自个人客户。订单来自现有客户和新客户。在任一情况下，都会通过向公司域发送请求 t 对客户执行信用检查。根据客户是分销商、回头客还是首次客户等，订单会应用各种折扣。

现有订单可能会更改。订购商品的数量可能会更改或订单可能会更改全部取消。已发货或已进入发货流程的订单无法取消。

客户或销售人员可以查看特定订单的状态。客户（通常是分销商）还可以检查其所有未完成订单的状态。

#### 2.1.1 Bean 定义

客户域的 UML 图显示在附录 C.2 中。

##### 2.1.1.1 订单

有关 OrderSes 接口的程序列表，请参阅附录 A。它使用各种实体 bean（OrderEnt、OrderLineEnt、ItemEnt）来实现其方法。OrderEntry 应用程序使用 OrderSes 接口的方法，从不直接访问实体 bean。

##### 2.1.1.2 订购

此实体 Bean 管理对订单表的访问。

##### 2.1.1.3 订单行

此实体 bean 管理对 OrderLine 表的访问，并从 OrderEnt Bean 中调用。应用程序不直接访问此 Bean，因为它是一个依赖对象。

**注释：**根据 EJB 1.1 规范，依赖对象最好作为单独的 Java 类（而不是企业 bean）实现。但是，由于 ECperf 的目的是强调 EJB 性能，因此决定将 OrderLine 实现为实体 Bean。

#### **2.1.1.4 条目**

此实体 Bean 管理对 Item 表的访问，并从 OrderEnt Bean 中调用。应用程序不直接访问此 Bean。

#### 2.1.1.5 订购客户

此无状态会话 Bean 负责在客户域中添加和验证客户。它使用实体 bean OrderCustomerEnt 来完成其任务。

#### 2.1.1.6 订购客户

此实体 Bean 管理

对订单数据库中的“客户”表的访问。它提供了添加和验证客户以及计算客户折扣的方法。

#### 2.1.1.7 购物车

此有状态会话 Bean

模拟客户在下订单之前将使用的购物车。客户端将商品添加到购物车后，将使用 OrderEnt bean 购买购物车的内容。

## 2.2 制造领域模型

该领域用于研究制造工厂中生产线的活动。有两种类型的生产线：*计划线*和*大订单线*。计划的行按计划运行并生成预定义数量的小部件。另一方面，只有当从分销商等客户那里收到大订单时，大订单才会运行。

当*工作订单*进入系统时，制造就开始了。每个工作订单适用于特定类型的小部件的特定数量。计划行工作订单通常是作为预测应用程序的结果创建的。大订单行工作订单是作为客户订单的结果生成的。创建工作订单时，将检索相应类型的小组件的物料清单

（BOM），并从库存中取出所需部件。当小组件在装配线上移动时，工作订单状态将更新以反映进度。工作订单完成后，将其标记为完成并更新库存。

随着零件库存的耗尽，需要找到供应商并发出采购订单。这是通过联系供应商域来完成的。

### 2.2.1 Bean 定义

制造领域的UML图显示在附录C.3至C.5中。

#### 2.2.1.1 工单

此状态较少会话 Bean 的每个实例管理一个工作订单，并使用实体 Bean

WorkOrderEnt，以及供应商域中的会话 Bean BuyerSes（请参阅条款 2.5）来完成其任务。

#### 2.2.1.2 大订单

这个无状态会话 Bean 处理大订单的创建和搜索。此 Bean 使用实体 Bean LargeOrderEnt 来完成其任务。

#### **2.2.1.3 接收**

此无状态会话 Bean 由供应商域中的 ReceiverSes Bean 创建（参见条款 2.3.1.2），负责将组件添加到库存中。此 Bean 使用实体 Bean ComponentEnt 来完成其任务。

#### **2.2.1.4 出发**

此实体 Bean 是需要访问制造数据库中的 compon ent 表的所有实体 Bean 的超类。它负责对零件表执行所有操作。

#### **2.2.1.5 汇编**

此实体 Bean 负责管理程序集。装配体由“零件”表中的一行表示，由多个零部件组成。

#### **2.2.1.6 工作订单**

此实体 Bean 管理对制造数据库中 WorkOrder 表的访问。

#### **2.2.1.7 大阶**

此实体 Bean 管理从客户域产生的定制订单（称为大订单）。

#### **2.2.1.8 组件**

此实体 Bean 管理对“零件”表中的组件和程序集的访问。

#### **2.2.1.9 库存**

此实体 Bean 管理对制造数据库中库存表的访问。它提供了添加和删除清单的方法。

#### **2.2.1.10 BOMnt**

此实体 Bean 管理对制造数据库中 BOM 表的访问。它提供了检索有关创建制造组件所需的部件的信息的方法。

### **2.3 供应商多玛模型**

该域负责与供应商的互动。供应商域根据需要订购的零件、需要它们的时间和供应商的报价来决定选择哪个供应商。公司向所选供应商发送采购订单。采购订单将包括要购买的各种零件的数量、必须交付到的站点以及必须交付的日期。从供应商处收到部件时，供应商域将结束向制造域发送消息以更新库存。

#### **2.3.1 供应商域中的豆类**

供应商域的 UML 图显示在附录 C.6 中。

### 2.3.1.1 买家

此有状态会话 Bean 根据它从制造域收到的请求发出采购订单，并使用实体 Bean 组件、供应商和 POEnt 来完成其任务。

### 2.3.1.2 接收器

此无状态会话 Bean 的每个实例都管理与一个采购订单相关的组件的交付。此 Bean 使用以下实体 Bean 来完成其任务：POEnt 和 ComponentEnt。

### 2.3.1.3 供应商

此实体 Bean 管理对供应商数据库中“供应商”表的访问。

### 2.3.1.4 供应商组件

此实体 Bean 管理对供应商数据库中的供应商组件表的访问。这个bean是从SupplierEnt中调用的。

### 2.3.1.5 POEnt

此实体 Bean 管理对供应商数据库中的采购订单表的访问。它还负责创建采购订单中的采购订单行。

### 2.3.1.6 极点

此实体 Bean 管理对供应商数据库中的采购订单行表的访问。这个bean是从POEnt内部调用的。

### 2.3.1.7 SComponentEnt

此实体 Bean 管理对供应商数据库中组件表的访问。

## 2.4 企业域名模型

此域管理客户、零件和供应商的全球列表。有关所有客户的信用信息（包括信用额度）仅保存在公司域的数据库中。这是为了提供最大的安全性和隐私性。

对于每个新订单，客户域都会请求向公司域进行信用价值检查。  
客户折扣也会在企业域中为每个新订单或每当订单更改时计算。折扣计算基于规则来确定客户属于哪个类别。

### 2.4.1 企业领域的 Bean

公司域 bean 的 UML Diagram 显示在附录 C.1 中。

#### 2.4.1.1 客户关系

此实体企业 Bean 负责跟踪公司数据库中的客户信息。有关客户界面的程序列表，请参阅附录 A。客户端应用程序从不直接调用此 Bean。

#### 2.4.1.2 折扣

此实体 Bean 管理有关客户折扣的信息。

#### 2.4.1.3 规则

此实体 Bean 管理有关客户折扣规则的信息。各种折扣类别的规则表示为 Java 表达式并保存在数据库中。

### 2.5 数据库模型

ECperf 数据库的组件被定义为由 4 个独立的数据库组成（有关如何组合这些数据库的信息，请参阅条款 4.3.2）。每个域中都存在一个数据库，由多个表组成。基准实现假定可通过 JDBC 访问的 SQL 数据库。这是因为企业更有可能首先将其应用程序迁移到 component 模型，将其数据留在遗留数据库中。

**注意：**此版本的规范只允许支持 JDBC 的 RDBMS。

### 2.6 企业数据库

公司数据库存在于企业域中，并具有客户、供应商和部件的主副本。其中一些信息将在其他数据库中重复。假设每晚（或每周）批处理程序将从公司数据库中提取数据，根据需要对其进行调整，并更新其他数据库中的相应表。

#### 2.6.1 客户表

客户表包含有关公司所有客户的信息。此表的一部分复制到其他域中。信用评级/限额、账户余额等机密信息仅保存在 公司 数据库中。其他域必须查询此表才能获取信息。

字段名称	字段定义	注释
C_ID	整数	客户 ID（主键）
C_FIRST	瓦尔查尔（16）	名字（如果 c_last 公司，则未使用）
C_LAST	瓦尔查尔（16）	姓氏
C_STREET1	瓦尔查尔（20）	街道地址，1 号线
C_STREET2	瓦尔查尔（20）	街道地址，2 号线
C_CITY 瓦尔	查尔（20）	市
C_STATE	字符（2）	状态
C_COUNTRY	瓦尔查尔（10）	国家
C_ZIP	瓦尔查尔（9）	邮政编码
C_PHONE	瓦尔查尔（16）	电话

C_CONTACT	瓦尔查尔（25）	联系人
-----------	----------	-----

字段名称	字段定义	评论
C_SINCE	日期	客户自
C_BALANCE	数字（9，2）	账户余额
C_CREDIT	字符（2）	信用
C_CREDIT_LIMIT	数字（9，2）	评级“BC”或“GC” 信用额度
C_YTD_PAYMENT	数字（9，2）	年初至今的付款

2.6.2 零件表

有一个全局零件表，用于标识公司的所有小部件。这些包括制造的小部件（即组件）和购买的组件。

字段名称	字段说明	评论
P_ID	字符（15）	编码部件号（主键）
P_NAME	瓦尔查尔（10）	部件名称
P_DESC	瓦尔查尔（100）	描述
P_REV	瓦尔查尔（6）	修订号
P_UNIT	瓦尔查尔（10）	本部分的计量单位
P_COST	数字（9，2）	物料成本
P_PRICE	数字（9，2）	商品价格
P_PLANNER	整数	计划员代码 -
P_TYPE	整数	计划零件的人员 产品代码
P_IND	整数	制造或采购零件
P_LOMARK	整数	库存低水位线
P_HIMARK	整数	库存的高水位线

属性P\_TYPE表示这是什么类型的产品（例如：桌面/服务器/组件）。如果P\_IND指定“制造”，则它在物料清单表中应具有相应的条目。只有制造零件是订单数据库中ITEM表的一部分。

2.6.3 供应商表

供应商表包含公司所有供应商的列表。

字段名称	字段说明	评论
------	------	----



SUPP_ID	整数	供应商 ID（主键）
SUPP_NAME	瓦尔查尔（16）	供应商名称
SUPP_STREET1	瓦尔查尔（20）	街道地址，1号线
SUPP_STREET2	瓦尔查尔（20）	街道地址，2号线
SUPP_CITY	瓦尔查尔（20）	城市
SUPP_STATE	字符（2）	州
字段名称	字段 描述	注释
SUPP_COUNTRY	瓦尔查尔（10）	国家
SUPP_ZIP	瓦尔查尔（9）	邮政编码
SUPP_PHONE	瓦尔查尔（16）	电话
SUPP_CONTACT	瓦尔查尔（25）	联系人

2.6.4 站点表

站点表包含公司各个站点的地址。其中包括制造站点（供应商将小部件运送到的位置）和仓库（公司运送小部件的位置）。

字段名称	字段说明	评论
SITE_ID	整数	主键
SITE_NAME	瓦尔查尔（16）	供应商名称
SITE_STREET1	瓦尔查尔（20）	街道地址，1号 线
SITE_STREET2	瓦尔查尔（20）	街道地址，2号 线
SITE_CITY	瓦尔查尔（20）	城市
SITE_STATE	字符（2）	州
SITE_COUNTR Y	瓦尔查尔（10）	国家
SITE_ZIP	瓦尔查尔（9）	邮编

2.6.5 规则表

规则表 包含所有客户折扣规则，每个规则一行。架构中可以有多种类型的规则（由 R\_ID 标识），但只使用一种 - 即折扣。

字段名称	字段描述	注释
R_ID	瓦尔查尔（20）	规则类型（小学）
R_TEXT	长 varchar	规则文本，可以是多行

2.6.6折扣表

此表包含各种类别客户的折扣。当客户属于特定类别（基于规则表中的规则）时，将在折扣表中查找其折扣级别。

字段名称	字段描述	注释
D_ID	瓦尔查尔（64）	客户类别（主键）
D_PERCENT	整数	百分比折扣

2.7订单数据库

订单数据库存在于客户域中。它处理销售订单并运行 **订单输入** 应用程序。可以创建新订单、更改现有订单以及检索订单状态。

2.7.1 客户表

字段名称	字段定义	评论
C_ID	整数 瓦尔查尔（16）	
C_FIRST	瓦尔查尔（16） 瓦尔查尔（20）	客户 ID（主键）
C_LAST	瓦尔查尔（20） 瓦尔查尔（20）	名字（如果C_LAST是公司，则未使用） 姓氏
C_街道1	查尔（2） 瓦尔查尔（10）	收货街道地址，第 1
C_STREET2	瓦尔查尔（9） 瓦尔查尔（16）	行收货街道地址，第 2
C_CITY	瓦尔查尔（25） 日期	行收货城市
C_STATE		收货州收货国家/地区 收货区
C_COUNT RY	字段说明	邮政编码电话
C_ZIP		联系人客户自
C_PHONE	整数 整数 整数 数字（9，2）	
C_CONTA CT	数字（4，2）	
C_SINCE		

订单表

字段名称	评论
O_ID	订单 ID（主键）
O_C_ID	下此订单的客户
O_OL_CN T	订单行数
O_TOTAL	订单 总额
O_DISCOU NT	此订单的客户折扣
O_ENTRY_DATE	时间戳
O_SHIP_DATE	日期
2.7. 2 O_STATUS	订购发货日期
	完成订单的状态
	整数

2.7.3 订单行 表

字段名称	字段说明	评论
OL_ID		订单行 ID（主键）
OL_O_ID	整数 整数 字符（15） 整数 整数 日期	订单 ID（主键）
OL_I_ID		商品编号
OL_QTY		项目数量
OL_STATUS		状态：未完成，已发货
OL_SHIP_DATE	字段说明	
		订单行发货日期
项目表	字符（15） 数字（9，2） 字符（20） 字符（100） 浮	

字段名称	评论
I_ID	项 ID（主键）
I_PRICE	商品价格
I_NAME	名字
I_DESC	描述
I_DISCOUNT	此商品的折扣

2.7.4

2.8 制造数据库

制造数据库存在于制造域中，用于处理装配体、物料清单、库存和车间管理的制造。**客户**是订购物料的任何个人或组织。客户订购**小部件** - 这些是公司制造的产品（为简单起见，假设该公司不是经销商）。所有制造的小部件都是**装配体**，并且必须在 BOM 表中有一个条目。装配体由**零部件**组成。所有装配体和组合的列表构成了全局**零件**表。单个P\_ID唯一标识公司处理的每个组件。

2.8.1 零件表

此表派生自企业域中的“部件”表。

字段名称	字段说明	评论
P_ID	字符（15）	编码部件号（主键）
P_NAME	瓦尔查尔（10）	部件名称
P_DESC	瓦尔查尔（100）	描述
P_REV	瓦尔查尔（6）	修订号

P_PLANNE	整数			计划员代码 -
R				计划零件的人
				员
P_TYPE	整数			产品代码
字段名称	字段	描述	注释	
P_IND			整数	制造或采购零件
P_LOMARK				库存的整数低水位线
P_HIMARK				库存的整数高水位线

2.8.2 物料清单表

物料清单表包含公司生产的各种小部件的物料清单。每个小部件（装配体）由多个零部件组成。为简单起见，不处理子装配体（即装配体中的零部件不能依次是装配体）。

字段名称	字段说明	评论
B_COMP_ID	字符（15）	组件（主键）
B_ASSEMBLY_ID	字符（15）	它所属的程序集（主键）
B_LINE_NO	整数	物料清单中的行号（主键）
B_QTY	整数	数量/组装
B_ENG_CHANGE	瓦尔查尔（10）	工程变更参考
B_OPS	整数	操作 # - 使用流程中的哪个步骤
B_OPS_DESC	瓦尔查尔（100）	操作说明

此表中每个装配体将有多行，其中列出了其所有零部件。

2.8.3 工单 表

制造操作通过使用工作订单进行管理。工作订单指示需要制造的内容，并用于跟踪装配线的进度。

字段名称	字段说明	评论
WO_NUMBER	整数	工作订单编号（主键）
WO_O_ID	整数	销售订单
		ID（如果这是针对自定义
		订单）
WO_OL_ID	整数	销售订单中的订单行 ID
WO_STATUS	整数	现状

WO_ASSEMBL 字符（15）	正在制造的组件
Y_ID	
WO_ORIG_QT 整数	原装数量
Y	
WO_COMP_QT 整数	完成数量
Y	
WO_DUE_DAT 日期	订单到期日期
E	
WO_START_D 时间戳	工作开始的日期和时间
ATE	

系统创建工作订单并为其分配编号。工作订单可以是批处理装配体或单个销售订单项。如果是后者，则字段WO\_O\_ID和WO\_OL\_ID将标识此工作订单所针对的特定订单项。WO\_S TATUS monitor  
此工作订单的进度，并将包含“打开”、“已开始”、“已取消”、“已完成”、“已存档”等值。

2.8.4 大阶 表

此表是在客户域中接收的大型自定义订单的临时存储库。驱动程序将为此表中的每个条目创建工作订单，然后将其删除。

字段名称	字段说明	评论
LO_ID	整数	大订单编号
LO_O_ID	整数	销售订单编号
LO_OL_ID	整数	销售订单的订单项号
	字符（15）	订购的部件
LO_ASSEMBLY_I D		
LO_QTY	整数	订购数量
LO_DUE_DATE	日期	订单到期日期

2.8.5 库存表

库存表包含有关所有零件（已完成装配体和零部件）的数据。

字段名称	字段说明	评论
IN_P_ID	字符（15）	部件号（主键）
IN_QTY	整数	库存金额
IN_ORDERED	整数	订购数量
IN_LOCATION	字符（20）	仓库/箱子
IN_ACC_CODE	整数	财务代码 - 零件是否可用？

IN\_ACT\_DATE 日期

上次活动的日期

2.9 供应商数据库

供应商数据库存在于供应商域中，用于处理与供应商的交互。零件的采购订单从此域发给供应商，并接收转移到制造的库存。

2.9.1 供应商表

此表与公司数据库中的“供应商”表相同。参见条款 2.6.3

2.9.2 站点表

此表与公司数据库中的“站点”表相同。参见条款 2.6.4。

2.9.3 组件表

此表包含从外部供应商处购买的所有合作伙伴的信息。它的架构派生自公司数据库中的部件表。

字段名称	字段说明	评论
COMP_ID	字符（15）	编码部件号（主键）
COMP_NAME	瓦尔查尔（10）	部件名称
COMP_DESC	瓦尔查尔（100）	描述
COMP_UNIT	瓦尔查尔（10）	本部分的计量单位
COMP_COST	数字（9，2）	该部件当前最佳成本
QTY_ON_ORDE R	整数	跟踪未完成的采购订单
QTY_DEMAND ED	整数	制造商要求的数量
LEAD_TIME	整数	交货时间
CONTAINER_SI ZE	整数	订购金额

2.9.4 采购订单 表

公司向其供应商发出采购订单。每个采购订单将包含多行（每个部件一个）。

注释：POL\_P\_ID是部件号。  
实际上，它应该是供应商的零件号。为此，需要在零件编号和供应商之间进行映射（这是

2.9. 字段名称	字段说明	评论
5		主键
PO_NUMBER	整数	此采购订单的供应商
PO_SUPP_ID	整数	将此采购订单运送到的
PO_SITE_ID	整数	站点

采购订单 行表

字段名称	字段 说明	评论
POL_NUMBER	整数	采购 订单行
POL_PO_ID	整数	编号（主键）
POL_P_ID	字符（15）	这所属的采购订单（主键）
POL_QTY	整数	部件号
POL_BALANCE	数字（9， 2）	订购数量
	POL_DELDATE日期	未结余额
POL_MESSAGE	瓦尔查尔（100）	要求的交货日期

一个复杂的场景，因为每个零件需要有多个供应商）。供应商部件号通常是 BOM 的一部分，但为简单起见，始终使用单个部件号。

2.9.6 供应商组件 表

此表维护所采购的每个组件的供应商列表。每当供应商响应投标请求时，此表中的信息都会使用最新的定价和可用性信息进行标记。

字段名称	字段说明	评论
SC_P_ID	字符（15）	元件
SC_SUPP_ID	整数	此组件的供应商
SC_PRICE	数字（9， 2）	此组件的供应商价格
SC_DISCOU	浮	百分比折扣
NT		
SC_QTY	整数	折扣数量



SC\_DEL\_DA 整数  
TE

交货天数（从订购之日起）

## 第 3 条 - 工作负载描述

### 3.1 术语定义

3.1.1 **测试发起人** 一词是指发布基准测试结果的组织。这通常是主要的硬件或软件平台供应商。

3.1.2 **术语驱动程序** 或 **应用程序** 是指驱动基准测试的代码。驱动程序实现第 4 条中所述的运行规则，跟踪各种统计信息并报告最终指标。有关驱动程序的更多详细信息，请参阅条款 4.12。

3.1.3 术语 **EJB 容器**（或简称**容器**）是指控制 ECperf 工作负载的企业 Bean 生命周期的实体。有关更多详细信息，请参阅 EJB 1.1 规范。

3.1.4 术语 **ECtransaction** 是指对企业 Java Bean 的远程方法调用。

3.1.5 **术语业务交易** 是指由司机发起的工作单元，可能涉及 e 或多个 EC 交易。

3.1.6 术语 **数据库事务** 是本规范中描述的具有完整 ACID 属性的数据库工作单元，如第 4.11 条所述。数据库事务由容器或企业 e bean 作为事务的一部分启动。

3.1.7 术语 **[ x . y ]** 表示以 x 开始并以 y 结尾的值的封闭范围。

3.1.8 在 **[x.. y]** 表示随机独立选择并在 x 和 y 之间均匀分布（包括  $(x+y)/2$ ，精度位数与所示相同。例如，[0.01 . 100.00] 有 10,000 个唯一值，而 [1 ..100] 只有 100 个唯一值。

3.1.9 术语 **非均匀随机函数（NURand）** 在本规范中用于指用于生成客户 ID 和部件号的方法。此方法在指定的值范围内生成独立选择且不均匀分布的随机数 [x ..y]，并指定如下：

$$\text{NURand}(A, x, y) = ((\text{random}(0, A) | \text{random}(x, y)) \% (y - x + 1)) + x$$
 其中：

expr1 | expr2 代表 expr1 和 expr2 之间的按位逻辑 OR 运算  $\text{expr1 \% expr2}$

代表 expr1 模 expr2  $\text{random}(x, y)$  代表在 [x .. y] 内随机选择

A 是根据范围的大小 [x .. y]

NURand 在 TPC-W 基准测试规范中定义，该规范的版权归 TPC 所有。

3.1.10 术语 **测量间隔** 在本规范中用于指测试发起人报告性能指标的基准执行期间的稳定状态周期（详细要求见条款 4.9）。

3.1.11

术语**响应时间**在本规范中用于指从驱动程序发送的第一个字节请求业务事务到驱动程序收到最后一个字节以完成该业务事务所经过的时间（有关详细要求，请参阅条款 4.6.2 和 4.7.1）。

3.1.12 本规范中使用的**术语注入速率**是指来自客户域中 OrderEntry 应用程序的业务事务请求注入 SUT 的速率。

3.1.13 术语 **延迟时间**

在本规范中使用，是指从驱动程序接收到完成业务事务的最后一个字节到第一个字节发送给驱动程序请求下一个业务事务所经过的时间。延迟时间是响应时间和注入速率的函数。对于所需的注入速率，对于较大的响应时间，延迟时间会更小。

3.1.14

术语“**周期时间**”在本规范中用于指从驱动程序发送的第一个字节请求业务事务到驱动程序发送的第一个字节请求下一个业务事务所经过的时间。周期时间是响应时间和铺设时间的总和。

## 3.2 客户领域的业务

在客户域中运行的主要应用程序是订单输入。此工作负载不涉及 OrderEntry 应用程序中典型的用户交互。例如，搜索f或项目ID，获取项目描述，获取和验证客户信息等不处理。OrderEntry Driver 通过对 CartSes、 OrderSes 和 OrderCustomerSes 进行方法调用来实现此域中的业务事务。有关订单输入应用程序的描述，请参阅第 3.7 条。下面列出了 EC 交易：

### 3.2.1 新订单

此 EC事务将为具有一定数量的订单行的客户输入新订单。此 ECtransaction 由 OrderSes bean 的 newOrder

方法实现。它获取订购的所有商品的价格，同时考虑到商品折扣并计算总计。然后，它会计算客户对此订单的折扣，并调用公司域以检查客户是否有足够的信用额度来支付其购买费用。如果客户没有足够的信用额度，则会返回错误。否则，它将在订单表中输入一行，在订单行表中输入与

项目数一样多的行。如果订单是自定义订单，它将触发要在制造域中创建的大订单。

/\*\*

\* 新订单：为客户输入订单

\* @param customerId - 客户的ID

```

* @参数 数量
* @return int - 订单的ID
* @异常数据完整性异常 - 如果客户/项目不存在
* @异常不足信用异常 - 如果客户信用检查失败
* @exception CreateException - 如果创建订单 fails * @exception RemoteException - 如果存在系统故障 */
public int newOrder (int customerId, ItemQuantity[] 数量) 抛出 InenoughCreditException,
DataIntegrityException, RemoteException , CreateExcep tion;

```

### 3.2.2 变更单

此 EC交易将对现有的未交付订单进行更改。此 ECtransaction 由 OrderSes bean 的 changeOrder 方法实现。订单行将得到适当的修改。将更新数量字段并计算新订单总计。将重新评估此新订单总额的客户折扣和信用额度。如果数量为0，则删除订单行。如果传递了新的物料 ID，则会添加新的订单行。

```

/**
* 更改顺序：更改现有客户 或更改
* @参数订单ID - 正在更改的订单的ID
* @param数量 - itemId, 正在更改的订单行的数量对 * @exception 不足信用异常 -
如果客户信用检查失败 * @exception 远程异常 - 如果系统出现故障重新 */ 公共无效更改订单 (int orderId,
ItemQuantity[] 数量) 抛出不足信用异常、远程异常;

```

### 3.2.3 订单状态

此 EC事务获取特定订单及其所有订单行的状态。此 ECtransaction 由 OrderSes bean 的 getOrderStatus 方法实现。它找到客户ID和订单与订单 ID 对应的订单的发货日期。对于属于此订单的所有订单行，它将检索物料ID、订购物料数量和订单行发货日期。

```

/** * getOrderStatus: 检索订单的状态
* @param orderId - 订单的 ID * @return 订单 状态 对象
*
* @throws DataIntegrityExce ption * @exception RemoteException
- 如果存在系统故障 */ public OrderStatus getOrderStatus (int orderId)
抛出 DataIntegrityException, RemoteException;

```

### 3.2.4 客户状态

此 EC事务获取特定客户的所有输出订单的状态。此 ECtransaction 由 OrderSes bean 的 getCustStatus 方法实现。对于尚未为此客户交付的每个订单，它会检索订单

ID、订单行计数和订单发货日期。对于此订单中的所有订单行，它将检索物料 ID 和数量。

/\*\* \* 获取客户所有订单的状态

\*/

\* @param customerId int 客户 ID \* @return 客户状态 对象数组（每个订单一个）

\*/

\* @throws DataIntegrityException - 如果客户不存在 \* @exception RemoteException -

如果存在系统故障 \*/ public CustomerStatus[] getCustomerStatus (int customerId) 抛出

DataIntegrityException, RemoteException; }

### 3.2.5 取消订单

此 EC事务通过删除订单行及其关联的订单行来取消订单 。

/\*\*

\* 取消订单：取消现有客户 订单

\* @param orderId - 正在更改的订单的 ID \* @exception 远程异常 - 如果re 是系统故障 \*/

公共无效取消订单 (int orderId) 抛出远程异常;

### 3.2.6 添加客户

此 EC事务将新客户添加到客户域。它调用公司域中的 AddCustomer 事务，以将客户也添加到公司数据库。

/\*\* \* 此方法使用指定信息添加新客户

\*/

\* @param info - OrderCustomerEnt 的所有字段 \* @return id - 新客户的客户 ID \*/ public int

addCustomer (CustomerInfo info) 抛出 InvalidInfoException, DataIntegrityException, RemoteException;

### 3.2.7 验证客户

此 EC事务通过检查客户的信息是否正确形成的字段来验证现有客户。此 EC事务的主要目的是模拟一些业务逻辑。

/\*\*

```
* 此方法检查是否存在具有 * 指定 ID 的客户。 */ public void  
validateCustomer (int id) 抛出 DataIntegrityException, RemoteException;
```

### 3.2.8 添加到购物车

此 EC事务将一个项目添加到购物车中，我们使用 CartSes bean 的添加方法。

```
/**  
* 方法 add - 将商品添加到购物车 * @param item_qty 它具有要添加的项目 ID 和数量。  
* @异常远程异常 如果出现系统故障。 */ public void add (ItemQuantity item_qty)  
抛出 RemoteException;
```

### 3.2.9 购物车

此 EC事务通过调用 CartSes bean 的购买方法来购买购物车的内容。此方法反过来在 OrderSes bean 上调用 newOrder。

```
/** * 方法购买 - 购买购物车的内容。这将调用 OrderSes bean 的 newOrder 方法 *。  
* * @return int - 创建的新订单的订单 ID。  
* @异常远程异常 如果出现系统故障。  
* @exception 创建异常 如果创建新订单失败。 * @exception 信用不足异常，如果新订单因信用不良而失败  
*/ public int buy () 抛出 RemoteException, CreateException, InenoughCreditEx-  
接受;
```

## 3.3 Manu 制造领域的 EC 交易

制造域负责创建和执行工作订单以及管理装配线。

### 3.3.1 调度工单

此

EC事务从模拟计划生产线的制造活动或为单个销售订单调用的制造应用程序（请参阅条款 3.8）调用。它在工作订单表中创建一行，标识在物料清单中构成此装配体的组件，并从库存中分配所需的零件。为单个销售订单（大订单）调用此

EC事务时，将在创建工作订单后删除大订单。

如果此事务需要任何库存不足的零件，则可能会导致调用供应商域。

```
/** * 从 Mfg 应用程序调用时计划工作订单的方法  
* @参数程序集 ID 程序集 ID  
* @param数量 原始数量 * @param到期日期 订单到期日期  
*  
* 创建的 @return 个工作订单的 ID * @exception 如果出现系统故障，则为远程异常 */ public  
整数调度工作订单 (String assemblyId, int qty, java.sql.日期到期日期) 抛出远程异常;
```

```
/** * 如果从客户域调用，则安排 work 订单的方法
* @参数销售ID 销售订单ID
* @param oLineId 订单行 ID
* @参数程序集 ID 程序集 ID
* @param qty 原始数量 * @param到期日 订单到期日期
*
* 创建 @return 的 workorder 的 id * 如果 存在系统故障@exception RemoteException*/ public Integer
scheduleWorkOrder ( int salesId, int oLineId, String assemblyId, int qty, java.sql.Date dueDate )
抛出 RemoteException;
```

3.3.2 更新工作顺序

当每个工作站完成其操作时，它将更新工作订单状态。使用附录 C.4 和 C.5 中的 UML 图描述的状态模式维护工作订单状态。

```
/** * 更新工作订单的状态
*
* @参数 wid 工作订单 编号
* @exception 远程异常， 如果存在 * 通信或系统故障 */ 公共无效更新工作订单（整数 wid）
抛出远程异常;
```

3.3.3 完成工作订单

此 EC事务 将工作订单标记为完成，并将小组件转移到库存。它获取此工作订单的程序集 ID 并更新此程序集的清单。它将已完成数量和工作订单状态更新为已完成。请注意，工作订单无限期保留。

```
/** * 完成工作订单：工作订单取消。
*
* * 将已完成的部分转移到库存。
```

ECperf™ Benchmark

Sun Microsystems Inc.

```
* @参数 wid 工作订单 编号
* @return布尔 值 false（如果无法完成）
* @exception 远程异常 如果存在 * 通信或系统故障 */ 公共布尔值 completeWorkOrder（整数 wid）
抛出远程异常;
```

3.3.4 创建大订单

此 EC交易是从客户域中启动的，前提是从未输入过大型自定义订单。这将导致基于销售订单创建工作订单。然后，工作订单将经历正常的制造流程，直到完成。

```
/** * 方法 createLargeOrder
```

---

```

*
*
* @param 订单销售订单的 ID
* @param oLineId 在销售订单中 * @param程序集 ID - 销售订单中的项目
* @参数 数量
* @参数 到期日期
* @返回大订单ID * @throws 如果系统出现故障，则出现远程异常
* */ public Integer createLargeOrder ( int orderId, int oLineId, String assemblyId, short qty,
java.sql.Date dueDate) 抛出 RemoteException;

```

### 3.3.5 查找大订单

此 EC事务由驱动程序调用，它检索当前存在于 LargeOrder 表中的所有大订单。请注意，大订单在计划后由计划工作订单删除。

```

/**
* 方法 查找大订单
* 检索所有当前存在的大订单 * @return LargeOrderInfo 元素的 Vector ， 每个元素包含：
* largeOrderId, orderId, orderLineId, assemblyId, quantity and dueDate
* * @throws 如果系统出现故障，则为远程异常
* */ public java.util.Vector findLargeOrders ( ) 抛出 RemoteException;

```

## 3.4 企业领域的欧洲经济交易体系

### 3.4.1 添加客户

添加新客户。此 EC事务 从客户域中调用，以动态添加新客户。

```

/** * 创建一个新客户。
* @param客户信息信息，包含：* 名字、姓氏、地址、联系人、客户自、余额和 *
年至今付款。
* / CustomerEnt create (CustomerInfo info) 抛出 java.rmi.RemoteException,
javax.ejb.CreateException;

```

### 3.4.2 有足够的信用

此 EC交易检查客户是否有信用余额来支付他的购买。这是在客户输入/更新订单时从客户域中的新订单和变更订单 EC事务调用的。

```

/** * 此方法从 * 中的客户 doma 中的 OrderBean 调用，以检查客户是否有足够的信任度。
* @param金额 - 所需的信用额度 * 如果存在足够的信用，则@return为真，否则为假 */ 公共 布尔值
具有足够的信用 (双倍金额) 抛出 远程异常;

```



### 3.4.3 获得百分比折扣

此 EC 事务根据折扣规则和客户历史记录计算客户对特定订单的折扣。这是从 客户域中的 NewOrder 和 ChangeOrder ECtransactions

调用的。规则引擎解析器执行的大部分工作，该解析器解析存储在 C\_RULE 表中的折扣规则并确定客户属于哪个类别。然后，它会查找 C\_DISCOUNT 表以获取此客户类别的百分比折扣。

```
/**
 * Method getPercentDiscount
 * @参数 金额 * @return 百分比折扣
 *
 * @throws dataIntegrityException * @throws
 * RemoteException
 * */ 公共双 倍获得百分比折扣（双倍金额）
```

ECperf™ Benchmark

Sun Microsystems Inc.

抛出 DataIntegrityException, RemoteException;

## 3.5 供应商域中的 EC 交易

供应商域负责供应完成工作订单所需的组件。组件订单从制造部门接收并发送给外部供应商。订单交货时，“制造”中的库存将更新。

### 3.5.1 添加组件

此 EC 事务

将检查是否有任何未完成的采购订单，并将该组件添加到要订购的组件列表中 d。

```
/** * 添加：将库存不足的组件添加到采购订单。
 * @param 组件 ID - 要添加到采购订单的组件的 ID。
 * @参数 数量必填 - 要购买的号码。
 * @异常 远程异常 - 如果出现系统故障。 */ public void add (String componentID, int qtyRequired) 抛出
 * RemoteException;
```

### 3.5.2 购买

将所有库存不足的组件添加到采购订单（AddComponent）后，ScheduleWorkOrder 将调用采购来购买组件。此 EC 交易

将找到所有可以在所需交货时间内交付所有组件的供应商，并选择提供最优惠价格的供应商并创建采购订单。它通过 http 连接将采购订单以 XML 格式发送到供应商模拟器。

```
/** * 采购：发出采购订单。
 * @exception ECperfException * @exception RemoteException - 如果出现系统故障。
```

\* @exception CreateException - 如果创建 Purchase 订单失败。 \* @exception  
FinderException \*/ public void purchase () 抛出 RemoteException, FinderException,  
CreateException, ECperfException;

### 3.5.3 交付采购订单

当此 EC事务收到来自  
指示采购订单已交付的供应商模拟器。它查找所有采购订单行并更新POL\_DELDATE字段。  
此交易将更新“制造”中的库存，并更新  
“组件”表中的QTY\_ON\_ORDER和QTY\_DEMANDED

/\*\* \* 交付采购订单 - 表示采购订单的一部分已送达。  
\* @param del - 包含有关 已交付的 POLine 的信息。  
\* @异常远程异常 - 如果出现系统故障。 \*/ public void deliverPO (DeliveryInfo del) 抛出  
RemoteException;

### 3.6 供应商模拟器

供应商模拟器是一个Java  
Servlet，可以在任何启用Java的Web服务器中运行。它不是SUT的一部分，因此它在外部的  
机器上运行。它通过符合 HTTP 1.1 协议的 HTTP 连接发送和接收 XML 文档来与 SUT  
通信。模拟器为供应商域提供了一种模拟向供应商发送和接收订单的过程的方法。

供应商模拟器接受来自供应商域中  
的采购订单，处理采购订单，然后在根据组件的提前期休眠一段时间后将采购订单中包含  
的订单行交付给 ReceiverSes Bean。供应商域和仿真器之间的这种交互如图 4 所示。

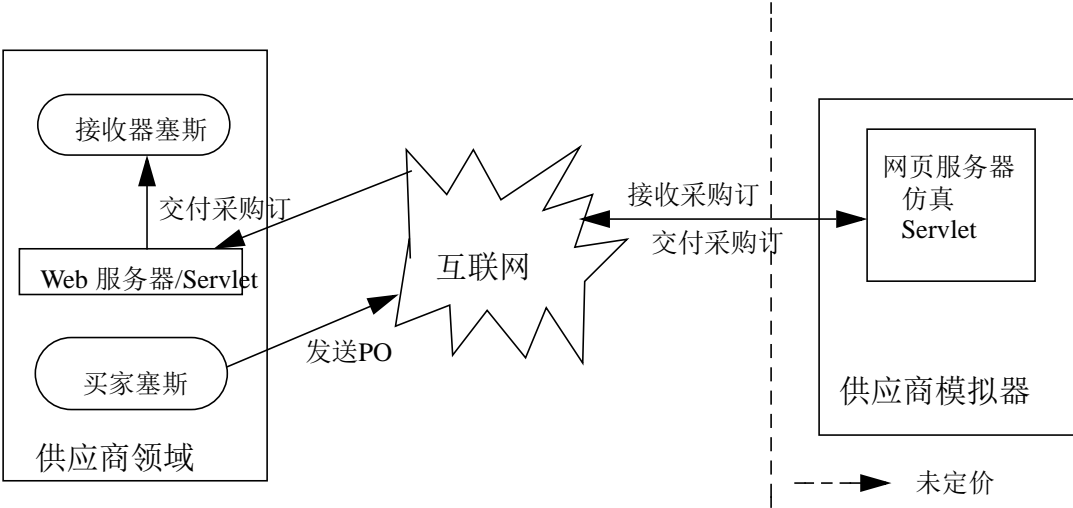


图 4: 供应商仿真器交互

3.6.1 采购订单

采购订单通过 HTTP连接以XML格式从供应商域（见条款3.5.2）接收。XML 文档中包含采购订单的详细信息：

po 采购订单 站点的编号 ID采购订单应交付到 numLines 的 站点的 ID  
采购订单中的 采购订单行数

对于采购订单中的每一行，必须提供以下内容：

行号	采购订单中的行号
编号	零件的 ID
数量	按订单数量
价格	该供应商的零件价格
交货时间	零件必须在最长交货时间内交付

3.6.2 零件交付

收到采购订单后，模拟器将分析 XML 并提取每个采购订单行。如果 XML 格式正确，则通过 HTTP 连接返回肯定响应，否则将返回否定响应。然后关闭 HTTP 连接。

然后，根据提前期按升序一次处理一个采购订单行。对于每一行，都会生成一个 XML 文档，其中包含：

poNumber	该行所属的采购订单的 ID
波林	采购订单中的行号
pId	正在交付的部件的 ID
数量	交付数量

然后，采购订单行的 XML 通过 HTTP 传输到供应商域中的 servlet，后者又调用 ReceiveSes Bean。请注意，供应商域中的 servlet 是必需的，因为供应商域和 E 模拟器之间的通信是通过 http 进行的。

3.7 订单输入 应用程序

订单输入应用程序负责输入新订单，更改现有订单和检索订单状态。

每个新订单都代表客户下达，其 ID 使用 NURand（A， 1， nCust）选择，其中 A 为 255，nCust 是客户池中的数字（请参阅条款 4.3.4）。如果所选客户不存在，则使用随机生成的数据为O\_Customer表中的各个字段添加新客户。如果选定的客户 ID 在初始数据库填充内，则会验证该客户 ID。

下一步是确定订单行。订货行的物料使用 NUR 和 (A, 1, NUM\_ITEMS) 进行选择, 其中 A 为 63, NUM\_ITEMS 是公司正在制造的物料数。订单中的订单行计数在 [1..5]90% 的情况下, 订单是由普通客户下达的, 该客户在 [10..20] 这个数字在所有订单行中平均分配。因此, 平均订单大小为 15 个小部件, 每个订单行的平均数量为 5.10% 的情况下, 假定订单由分销商下达 (称为大订单)。对于这些订单, 小部件的总数在 [100..200], 再次将它们平均分配给所有有序行, 其计数在 [1..5] 内随机选择。因此, 对于一个大订单, 平均订单大小为 150 个小部件, 它们在所有订单行中平均分布。

大订单的平均小部件数量是常规订单的 10 倍。让我们计算一下对于注射速率为 10 意味着什么。平均每秒将收到 5 个新订单 (基于 4.6.1 中的混合要求)。其中 4.5 个订单的平均大小为 15 个, 并且 0.5 来自平均规模为 150 的分销商。每秒订购的平均小部件为 因此, 常规订单为 67.5 (4.5 \* 15), 大订单为 75 (0.5 \* 150)。因此, 53% 的寡妇订购来自大订单。概括地说, 对于 Ir, 常规订单将有 6.75\*Ir 小部件, 大订单将有 7.5\*Ir 小部件, 总小部件订购率为 14.25\*Ir。

最后, 司机下订单。50% 的时间通过调用 OrderSes bean 的 newOrder 方法下订单。另外 50% 的时间, 通过首先将其添加到购物车来下订单。驱动程序通过使用 1000 辆推车的数组并选择要使用的购物车, 以便:

```
cartId = -ln(x) * 100 其中 x = 来自均匀分布的随机数, 使得 (0 < x <= 1) if (cartId > 1000)
cartId = 1000;
```

如果 cartId 不为空 (即购物车已存在), 驱动程序将删除购物车中的所有项目。然后它逐个添加所有订单行。添加所有物品后, 将购买购物车的内容。然后删除购物车 90% 的 time。10% 的情况下, 购物车保持不变, 以后可以重复使用。

驱动程序将所有成功的 newOrder 事务返回的 orderId 保存在 newOrderIds 数组中, 以供 changeOrder 和 getOrderStatus 事务使用。

对于每个更改 RDER 业务事务, 订单 ID 都是在 [1..NUM\_ORDERS] 加上 newOrderIds 数组, 其中 NUM\_ORDERS 是初始数据库中的订单数。驱动程序首先执行 getOrderStatus EC 事务, 以按要更改的顺序查找状态和订单行计数 n。如果它发现订单已发货或不再存在, 或者订单只有 1 个订单行, 它将在不同的订单行上重试交易次序。90% 的时间, 订单行将被 changed。剩下的 10% 的时间, 订单被删除。对于要更改的订单行, 在 [1..n]。对于这些订单行中的每一个, 数量交替递增或递减。对于要删除的 order, 驱动程序只在其中一个 newOrderId 上执行此操作, 即不会删除初始数据库填充中存在的订单。这将保留连续运行的表的基数。

对于 getOrderStatus business 事务, 在 [1..NUM\_ORDERS] 加上 newOrderIds 数组, 其中 NUM\_ORDERS 是初始数据库中的订单数。getOrderStatus EC transaction 使用选定的订单 ID 执行。

对于 getCustStatus 业务事务，在 [1.NUMCUSTOMERS]，其中 NUMCUSTOMERS 是初始数据库中的客户数以及执行的相应 EC 事务。

对于所有业务交易，驱动程序会跟踪平均响应时间、目标和实际周期时间、事务计数等，并在基准测试运行结束时生成包含最终指标和其他统计信息的报告。

3.8

制造应用

制造应用程序为每个 PG 运行两种类型的装配线：计划装配线和大订单装配线。为简单起见，假设制造不同小部件所需的工作是相同的。这些线路之间唯一真正的区别是它们的调度方式。

制造应用程序跟踪启动的工作订单数量、生成的小部件数量以及完成每个工作订单的平均时间。

3.8.1

规划线路

计划线连续并发运行，平均产生所有小部件的 47%。小部件的制造速度必须大致等于订购小部件的速度。这个想法是及时制造。工厂车间由总共  $p$  条计划线路组成（有关  $p$  的值，请参见条款 4.3.5），每条线路有 3 个站点。工作站标志着制造流程中的独特操作，例如多步骤制造、测试、包装等。

“制造”应用程序为每个计划行生成线程。每个计划行以每秒 0.2 个（或 5 秒内生成 1 个工作订单）的平均速率生成工作订单。每个工作订单平均将生成 11.25 个特定类型的小部件。使用 NURand (NUM\_ITEMS) 选择类型，其中  $A$  为 63，NUM\_ITEMS 是公司正在制造的物料数量。的情况下，工作订单的数量被选为 11，25% 的情况下，数量被选为 12。这会产生每个工作订单 11.25 个小部件。

第一个工作站将在生成工作订单 ID 后获取该 ID。然后，它将等待其指定的睡眠时间并更新工作订单状态。睡眠时间是 .333 秒的恒定量。工人继续移动到下一个工位，依此类推，直到到达最后一个工位（每个工位都知道它在链条中的位置）。因此，工单将在所有 3 个工位上花费 1 秒钟。在最后一站，工单被标记为 comtotal，物料被转移到库存中。测量从工作订单开始到结束的响应时间  $Tr$ ，并将延迟时间  $Td$  计算为：  
 $Td = 5 - Tr$

由于计划线需要在 5 秒内生成 1 个小部件，因此  $Tr$  可以高达 5 秒。如果  $Tr$  低于 5（即  $Td > 0$ ），则计划行将等待  $Td$  秒，然后循环回去创建下一个工作订单。

### 3.8.2 大订单行

大订单行处理的生产速度高于计划线可以生产特定小部件的速度。这些生产线上的活动总计来自收到的所有订单的 10%，占有所有制造小部件的 53%。

每次收到大订单时，客户域都会通知制造域。**LargeOrder** 行从制造应用程序开始，定期检查是否存在大型订单，并立即为每个订单安排新的工作订单。

与计划线一样，大型订单行有 3 个站点，每个站点花费的平均时间相同。  
传入订单所需的并发大订单行数与传入订单一样多。

---

## 第 4 条 - 扩展和运行规则

### 4.1 术语定义

4.1.1 术语“**部署单元**”是指部署特定域中的 Bean 的容器或容器集。

4.1.2 术语 **ECclient** 是指在  
Deployment Unit. EC客户端不一定映射到容器的 tcp 连接。

4.1.3 术语 **ECperf 引用 Bean** 是指为 ECperf 工作负载提供的企业 Bean 的实现。

4.1.4 术语 **ECperf 套件**是指为 ECperf 提供的完整套件。这包括 ECperf 引用  
Bean、驱动程序和装入程序。

4.1.5 术语 **BBops/min**是ECperf指标，表示在测量间隔内每分钟完成的平均成功**B**使用**OP**次数。  
BBop/min  
由在客户域中完成的业务事务总数组成，该总数与制造域中完成的工作订单总数（按分钟  
标准化）组成。

4.1.6 术语  $\$/BBops$ 在本规范中用于指SUT的总成本（见第7条）除以BBop的数量。

4.1.7 术语 **资源管理器**在本规范中用于管理数据库的软件产品，与数据库管理器相同。

### 4.2 商业产品要求

4.2.1 以下功能必须使用市售和支持的产品来实现：

- 操作系统
- Web 服务器和容器（用于供应商域和 供应商模拟器）
- EJB Server & Container
- 爪哇虚拟机
- 数据库服务器

4.2.2 SUT 必须在基准测试运行期间为所有应用程序组件提供满足 Java 2 平台企业版 1.2 版（J2EE）  
规范要求的运行时环境。SUT 必须满足 J2EE 兼容性要求，并且必须具有 品牌标识

Java 兼容，企业版。

**注释 1:** 当前版本的 ECperf 基准测试没有利用某些 J2EE 功能，例如：

- 喜欢
- 爪哇邮件
- JavaBeans 激活框架

**评论 2:** 新版本的 J2EE 兼容产品必须在可用日期之前通过 J2EE 兼容性测试套件（CTS）。有关可用性要求，请参阅条款 6.3.2。

4.2.3 用于运行基准测试的 ECperf 工具包版本必须与 ECperf 规范的版本（本文档）匹配。

## 4.3 扩展 基准

ECperf

基准测试的吞吐量由订单输入和制造应用程序的活动驱动。两种应用的吞吐量都与所选的注入速率直接相关。为了提高产量，需要提高注射速率。基准测试还要求  
在各个表中填充许多行。

**Comment:** 扩展要求的目的是保持 呈现给 SUT 的 EC事务负载、ECtransactions 访问的表的基数、注入速率和 生成负载的 EC客户端数量之间的比率。

### 4.3.1 伸缩要求

4.3.1.1 数据库缩放由阶次注入速率  $ir$  缩放作为阶跃函数完成，其方式是不需要在每次  $Ir$  的值更改时重新加载数据库。

4.3.1.2 站点和供应商表的 基数是固定的。

4.3.1.3 客户相关表（即客户、订单、订单行和工单）的基数将作为阶跃函数  $C$  增加，其定义为：

$C = 10$  的最小倍数  $\geq Ir$  对于 example，如果  $Ir = 72$ ，则  $C = 80$ 。

4.3.1.4

与物料相关的表，即零件、物料清单、库存和物料的基数将作为阶跃函数  $P$  缓慢增加，其定义为： $P = 100$  的最小倍数  $\geq Ir$  例如，如果  $Ir = 72$ ， $P = 100$ 。

### 4.3.2 数据库伸缩规则

以下缩放要求表示各个域中表的初始配置，其中  $C$  和  $P$  在 4.3.1.3 和 4.3.1.4 中定义。



表 2: 数据库伸缩规则

域	表名	基数 (以行为单位)	评论
企业			
	C_Site	1	
	C_Supplier	10	
	C_Customer	$75 * C$	
	C_Rule	1	
	C_Discount	6	
	C_Parts	$(11 * P)_1$	P 组件 + 10 * P 组件
订单			
	O_Customer	$75 * C$	NUM_CUSTOMERS
	O_Item	P	NUM_ITEMS
	O_Orders	$75 * C$	
	O_Orderline	$(225 * C)_1$	平均每笔订单 3 个
制造业			
	M_Parts	$(11 * P)_1$	
	M_BOM	$(10 * P)_1$	
	M_Workorder	P	
	M_Inventory	$(11 * P)_1$	
供应商			
	S_Site	1	
	S_Supplier	10	
	S_Component	$(10 * P)_1$	每个组件平均 10 个组件
	S_Supp_Component	$(100 * P)_1$	
	S_PurchaseOrder	$(.2 * P)_1$	2% 的组件
	S_PurchaseOrderLine	第 1 页	每个采购订单平均 5 个

1. 这些大小可能会根据实际生成的随机数而有所不同。

### 4.3.3 扩展数据库和容器

4.3.3.1 为了满足各种客户的要求，ECperf 基准测试可以在两个类别中运行，即**集中式**和**分布式**。结果发表在

集中式工作负载 无法与分布式工作负载上发布的结果进行比较。

4.3.3.2 在 工作负载的集中版本中，允许合并所有 4 个域。这意味着基准实现者可以选择运行单个部署单元，该部署单元访问包含所有数据库表的单个数据库。但是，基准实现者可以自由地将域分离到其部署单元中，并且仍然可以运行单个数据库。在 Cent 集合化工作负载中对 XA 两阶段提交没有要求。

4.3.3.3 ECPerf 的**分布式**版本 旨在对应用程序性能进行建模，其中 ECperf 建模的全球企业使用异构资源管理器跨业务域执行事务。在此模式 1 中，工作负载需要单独的部署单元和单独的数据库，以及 所有域中的单独 DBMS 实例。在跨多个域的 EC 事务中，需要符合 XA 标准的可恢复两阶段提交。此两阶段提交的配置需要以支持异构系统的方式完成。

**评论：** 即使实现可能对所有域使用相同的资源管理器，EJB 服务器/容器和资源管理器也无法利用同构资源管理器的知识来优化两阶段提交。

4.3.3.4 SUT由 一个或多个物理计算节点组成，其数量由实施者自由选择（包括SUT节点数== 1），但须符合第7条的原则要求。所需的数据库和 EJB 容器的总数可以根据需要映射到 SUT 计算节点，但须遵守上面列出的最小计数要求。但是，除了固有的消除 WAN/LAN 流量之外，实现不得特别利用数据库和容器的共置。

### 4.3.4 扩展 订单输入 应用程序

为了强调容器处理并发会话的能力，基准测试要求最小数量的 ECclient 等于  $5 * I_r$ ，其中  $I_r$  是所选的注入速率。这个数字在基准测试运行过程中不会改变，尽管这样做更能代表现实世界的电子商务。

对于每个新订单，每个 ECclient 都采用从 客户池，定义为  $nCust = 100 * C$ 。例如，如果  $I_r = 100$ ，则数据库最初填充了 7500 个客户（NUM\_CUSTOMERS），则将代表 10,000 个客户下新订单。

### 4.3.5 扩展制造应用程序

制造应用程序的缩放方式与订单输入应用程序类似。由于目标是即时制造，随着订单数量的增加，需要相应提高小部件的制造速度。这是通过将计划线的数量  $p$  按比例增加到  $I_r$  来实现的

$$p = 3 * I_r$$



## 4.4 数据库要求

### 4.4.1

所有表都必须具有数据库填充要求定义的适当缩放的行数（请参阅条款 4.3）。

### 4.4.2

除了 `ECperf` 工具包中的引用模式脚本中包含的对象之外，不能定义任何数据库对象，例如索引、视图等（请参阅 4.1.4）。可以修改数据类型，前提是它们在语义上等效于脚本中指定的标准类型。

### 4.4.3

在任何提交状态下，主键值在每个表中必须是唯一的。例如，对于水平分区的表，跨所有分区的行的主键值必须

### 4.4.4

必须使用 `ECperf` 加载程序填充数据库，该程序作为 `ECperf` 工具包的一部分提供。`load` 程序使用标准的 `SQL` 语句，并通过 `JDBC` 加载所有表，并且应该在所有表中保持不变。但是，允许出于移植目的进行修改。所做的所有此类修改必须在完整披露报告中披露。

## 4.5 Bean 部署要求

### 4.5.1

测试发起者必须运行 `ECperf` 引用 Bean。`ECperf Reference Beans` 有 `CMP` 和 `BMP` 版本。发起人可以选择部署 `CMP` 或两者的组合。另请参阅第 4.2 条了解容器要求。

### 4.5.2

允许对 `ECperf` 引用 Bean 进行的唯一更改是在我实现 Bean 管理的持久性（`BMP`）的类中。允许对 `BMP` 代码进行的唯一更改是用于类似于条款 4.4.4 的移植。所有代码修改都必须出现在完整披露报告中，并附有更改

### 4.5.3

必须使用随 `ECperf` 引用 Bean 提供的部署描述符，而无需进行任何修改。

**注释：**理想情况下，基准测试应该要求在所有基准测试实现中部署相同的 `ejb-jar` 文件。但是，`xmls` 中的某些环境是在生成时动态生成的。此外，由于产品的不成熟性，部署描述符内容被允许合并到单个文件中，拆分为多个文件只要它们的内容不变。

4.5.4 不允许提交 EJB 1.1 规范中指定的选项 A。假定数据库可由外部应用程序修改。

4.5.5 避免在事务结束时执行存储操作的特定于供应商的标志或参数 是

不允许。

4.6 订单输入 驱动程序要求

4.6.1 交易组合要求

订单输入驱动程序在客户域中重复执行业务事务。业务事务由驱动程序根据表 3 中所示的组合进行选择。由于我们要测试 EJB 容器的事务处理能力，因此上面的组合是最新的密集型的。在现实世界中，读者可能多于作家。

基准测试中实现的实际组合必须在每种交易类型目标组合的 5% 以内。例如， newOrder 交易可以在总组合的 47.5% 到 52.5% 之间变化。驱动程序检查并报告是否满足组合要求。

表 3.订单输入应用程序的混合要求

交易类型	百分比混合
新订单	50%
获取订单状态	20%
更改顺序	20%
获取客户状态	10%

4.6.2 响应时间要求

订单输入驱动程序测量并记录不同类型的业务事务的响应时间。仅包括测量间隔中成功完成的业务交易。每种类型的至少 90% 的业务事务的 Response Time 必须小于下面表 4 中指定的约束。每种事务类型的平均响应时间不得超过 90% 响应时间的 0.1 秒。此要求可确保所有用户都能看到合理的响应时间。例如， 如果 newOrder 事务的 90% 响应时间为 1 秒，则平均值不能大于 1.1 秒。驱动程序检查并报告是否满足响应时间要求。

表 4.订单输入应用程序的响应时间要求

交易类型	90% RT（以秒为单位）
新订单	2
获取订单状态	2

更改顺序	2	
获取客户状态	2	
<i>ECperf<sup>TM</sup> Benchmark</i>		<i>Sun Microsystems Inc.</i>

4.6.3 周期时间要求

对于每个业务事务， OrderEntry  
驱动程序从负指数分布中选择周期时间，该分布由以下公式计算，以便尽可能达到所选的最大平均注入速率。

$T_c = -\ln (x) / I_r$

哪里：

ln = 自然对数（基数 E）

x= 精度至少为 31 位的随机数，来自均匀分布，使得（0 < x <= 1）

I<sub>r</sub>= 平均注射速率

分布在平均值的 5 倍处截断。对于每个业务交易，驱动程序测量  
确定响应时间 Tr 并计算延迟时间 Td 为 Td = Tc - Tr。如果 Td >  
0，则驱动程序将在开始下一个事务处理之前休眠一段时间。如果选择的循环时间Tc小于Tr  
，则实际循环时间（Ta）大于所选的循环时间。平均实际周期时间允许偏离目标周期时间  
5%。驾驶员检查并报告是否满足周期时间要求。

4.6.4 其他要求

下表显示了 OrderEntry  
应用程序中各种数量允许的值范围。驾驶员将检查并报告是否满足这些要求。

表 5.杂项订单输入要求

数量	目标 价值	允许的最小值	允许的最大数量
小部件订购率/分钟	855 * 和	812.4 * 和	897.6 * 和
大订单 小部件订购率/分钟	450* 和	427.8 * 和	472.8 * 和
常规订单 小部件订购率/分钟	405 * 和	384.6 * 和	425.4 * 和
大订单百分比	10	9.5	10.5
购物车订单百分比	50	47.5	52.5
订单未通过信用检查的百分比	10	9	10
\$ 已删除的订单	10	9.5	10.5

4.6.5 客户域的性能指标

客户域的指标是事务/分钟，由测量间隔内成功完成的所有业务事务类型的总数除以测量间隔的长度（以分钟为单位）组成。

4.7

制造博士要求

4.7.1

响应时间要求

制造驱动因素测量并记录完成工作订单所花费的时间。仅包括测量间隔中成功完成的工作订单。至少 90% 的工作订单的响应时间小于 5 秒。平均响应时间不得大于 90% 响应时间的 0.1 秒。

4.7.2

其他要求

下表显示了 Manufa  
生产应用程序中各种数量允许的值范围。制造驱动程序将检查并报告运行是否满足这些要求。

表 6.其他制造要求

数量	目标值	允许的最小值	允许的最大允许值
大订单小部件速率/分钟	450* 和	405* 和 495* 和	
计划线路小部件速率/分钟	405* 和	364.8* 和 445.8*	和

4.7.3

制造领域的性能指标

制造域的指标是工作订单/分钟，无论是在计划生产线上还是在大订单生产线上生产。

4.8 ECperf 指标

的计算

4.8.1 ECperf 基准测试的两个主要指标是每分钟基准业务运营量（BBops/min）和定义为 \$/BBops 的价格绩效指标。

4.8.2 ECperf 基准 测试的总体指标是通过将客户域中的 OrderEntry  
应用程序和制造域中的制造应用程序的指标相加来计算的：

**BBops/min =（事务 + 工单）/分钟**

4.8.3 所有报告的  
BBops/min必须测量，而不是估计，并表示到小数点后两位，四舍五入到百分之一。

4.8.4 绩效指标必须报告为BBops/min@Std或BBops/min@Dist  
分别是分布式工作负载和分布式工作负载。有关这些工作负载的说明，请参阅条款 4.3.3.2 和条款 4.3.3.3。例如，如果测量结果为 123.45 BBops/min

---

集中式工作负载，必须报告为 123.45 BBops/min@Std。

## 4.9 测量间隔要求

订单和制造应用程序必须在基准测试运行开始时同时启动。

在测量间隔之前，必须有一个至少 10

分钟的爬坡期，在此时间结束时必须达到稳态吞吐量水平。在测量间隔结束时，必须保持稳态吞吐量水平至少 5 分钟，之后 run 可以终止。

### 4.9.1 稳态

报告的指标必须在测量间隔内计算，在此期间，吞吐量级别处于稳定状态，表示 SUT 的真正可持续性能。每个测量间隔至少为 30 分钟，并且必须代表 8 小时的运行。

**注释：**目的是将吞吐量的任何周期性波动或任何周期性活动（例如 JVM 垃圾收集、数据库检查点等）作为测量间隔的一部分包含在内。

### 4.9.2 重现性

为了证明稳态条件在测量间隔期间的可重复性，必须测量至少一个与报告的测量确定间隔持续时间相同的额外（非重叠）测量间隔，并且其 BBops/min 必须大于报告的 BBops/min。此重现性运行的指标要求在报告的 BBops/min 的 5% 以内。

## 4.10 所需报告

### 4.10.1

在测量间隔期间开始和完成的客户和制造域中所有业务交易的响应时间的频率分布必须以图形格式报告。在每个图形中，x 轴表示响应时间，范围必须从 0 t 到所需的第 90 个百分位响应时间（N）的五倍。此 0 到 5N 范围必须分为 100 个等长区间。一个额外的间隔将包括从 5N 到无穷大的响应时间范围。必须报告所有 101 个间隔。y-axis 表示给定响应时间范围内每种类型的业务事务的频率，粒度至少为 10 个间隔。下面显示了此类图表的示例。



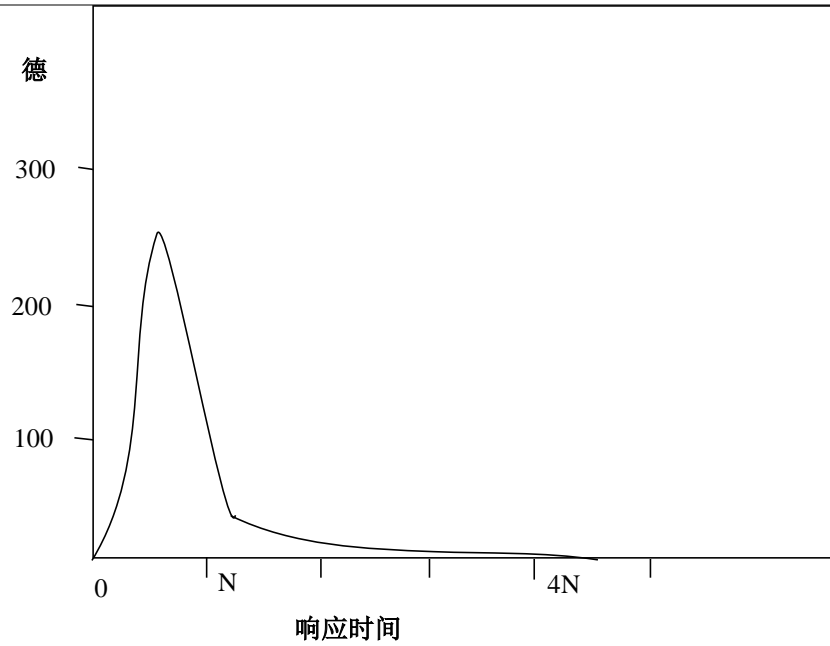


图 5: 示例响应时间图

4.10.2 必须为整个测试运行的制造应用程序报告  
工作订单吞吐量与闪烁时间（即挂钟时间）的关系图。x 轴表示从跑步开始经过的时间。y 轴表示业务事务中的吞吐量。必须至少使用 60 个不同的间隔，最大间隔大小为 30 秒。还必须报告测量间隔的打开和关闭。显示一个图表示例

下面。  
.

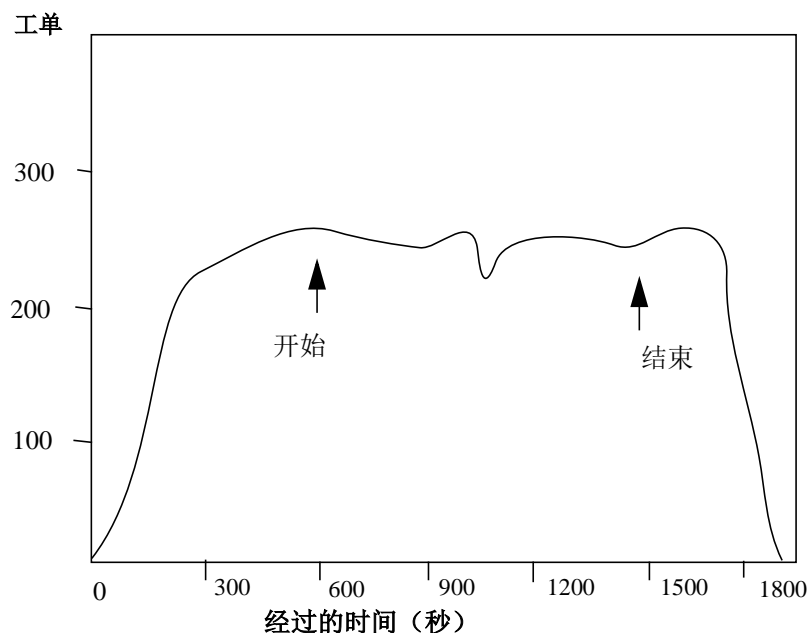


图 6: 示例吞吐量图

## 4.11 交易 属性要求

在此基准测试运行期间，事务处理系统的原子性、隔离性和持久性属性必须由被测系统支持。

### 4.11.1 原子性要求

被测系统必须保证数据库事务是原子的;系统将对数据执行所有单独的操作，或者确保不会对数据产生任何影响。

### 4.11.2 集中式工作负载的原子性测试

#### 4.11.2.1 在不公正性试验1

- 通过在O\_CUSTOMER表中查找c\_credit字段等于“BC”的客户，选择信用不良的客户。
- 修改 OrderEnt Bean 代码，以便在生成订单 ID 后立即打印它。
- 使用 Web 客户端应用程序输入此客户的新订单，该应用程序作为 ECperf 工具包的一部分分发。请注意由 Bean 代码打印的订单 ID。事务应失败，生成 InsufficientCreditException。
- 检索步骤 c 中记录的订单 ID 的状态。该顺序不应存在。
- 在数据库表O\_orderline查询ol\_o\_id与 步骤 c 中打印的订单 ID 匹配的行。不应返回任何行。

#### 4.11.2.2 原子城测试 2

- a. 通过在O\_CUSTOMER表中查找c\_credit字段等于“GC”的客户，选择信用良好的客户。
- b. 使用 Web 客户端应用程序为此客户输入新订单。事务应成功。记下返回的订单 ID。
- c. 检索上面注明的订单 ID 的状态。订单以及在步骤 b 中输入的订单行。应该显示。

#### 4.11.3 分布式工作负载的原子性测试

4.11.3.1 对集中式工作负载执行上述原子性测试 1 和 2。

#### 4.11.3.2 原子性试验 3

- a. 检测客户域中 OrderEnt Bean 中的 ejbCreate 代码，以显示新输入订单的订单 ID 和订单行 ID。对制造领域中的 LargeOrderEnt bean 执行相同的操作。更改 LargeOrderEnt bean 以将以下内容添加到 ejbStore 方法中：entityContext.setRollBackOnly ( )；
- b. 为任何客户输入一个新订单，确保它是一个大订单。请注意显示的订单 ID、订单行 ID 和大订单 ID 的值。
- c. 事务应回滚。验证O\_orders、O\_orderline和M\_largeorder表中具有上述步骤中检索到 id 的行是否不存在。

#### 4.11.4 隔离要求

各种隔离级别在 Java.sql.Connection 的 J2SE 文档中进行了描述。EJB 1.1 规范保留了在部署时基于每个资源指定隔离。这意味着 Bean 的所有业务方法共享相同的隔离级别。由于事务中通常涉及多个 bean，并且为了不惩罚只读事务，因此要求所有事务都具有 READ\_COMMITTED 或更高的隔离级别；即不允许脏读。

#### 4.11.5 耐久性要求

事务必须从 SUT 上的任何单一目标点持久。特别是，分布式阶段提交事务必须是持久的。没有针对持久性规定测试，因为这通常是数据库属性。

### 4.12 驱动程序规则

- 4.12.1 艾弗博士作为ECperf套件的一部分提供。发起人需要使用此驱动程序来运行 ECperf 基准测试。
- 4.12.2 驱动程序使用 EJB 容器支持的任何标准协议（如 RMI/JRMP、RMI/IIOP 等）与 SUT 通信。

---

4.12.3 驱动程序必须驻留在不属于 SUT 的系统上。

**注释：**此子句的目的是通过网络完成驱动程序和 SUT 之间的通信。

4.12.4 对于集中式工作负载，驱动程序系统必须使用单个 URL  
与容器建立通信，对于分布式工作负载，必须使用 4 个 URL（每个域一个）。

4.12.5 驱动程序系统上的 驱动程序调用的 EJB  
对象存根仅限于数据封送功能、负载平衡和故障转移功能。不允许基于 ECperf  
和/或基准配置的特定知识做出预配置决策。

4.12.6 驱动程序系统不得执行通常由SUT执行的任何处理，如 定义  
在第 5.1 条中。这包括但不限于：

- 执行部分或全部 ECperf Bean
- 缓存数据库或容器特定数据
- 向SUT传达有关即将发生的交易的信息

## 4.13 供应商仿真器规则

4.13.1 供应商仿真器作为 ECperf 工具包的一部分提供，可以部署在任何支持 Servlet 2.1 的 Web  
服务器上。

4.13.2 供应商仿真器必须驻留在不属于 SUT  
的系统上。但是，供应商仿真器可能会在其中一个驱动程序系统上重新设置。

**注释：**此条款的目的是通过网络完成供应商仿真器和 SUT 之间的通信。

## 4.14 基准优化规则

不允许特定于基准的优化。对 SUT  
上使用的配置或产品的任何优化都必须提高比此基准定义更大的工作负载类的性能，并且  
必须得到提供程序的支持和推荐。优化必须具有供应商的支持，并且应适合在与基准测试  
所代表的环境相当的环境中进行生产使用。禁止利用基准测试特定功能的优化。不适当的  
优化示例包括但不限于利用所使用的特定 SQL  
代码、各种字段或表的大小或基准测试中部署的 Bean 数量。

## 第 5 条 - SUT 配置

### 5.1 被测系统（SUT）要求

#### 5.1.1

SUT包括所有正在测试的组件。这包括网络连接、应用程序服务器/容器、数据库服务器等。

SUT包括：

- 支持工作负载和数据库所需的主机系统（包括硬件和软件）。
- 作为 SUT 一部分的主机与 SUT 的所有网络接口之间的所有网络组件（硬件和软件）。
- 在 SUT 中提供负载平衡的组件。

**注释 1：** 暂定要求形成物理 TCP/IP 连接所需的任何组件（通常称为 NIC、网络接口卡）成为 SUT 的一部分。

**注释 2：** 由 Driver 和 SUT 之间的一个或多个交换机组成的基本配置不被视为 SUT 的一部分。但是，如果使用任何软件/硬件来影响基本 IP 路由和交换之外的流量，则将其视为 SUT 的一部分。

5.1.1.1 SUT 必须有足够的在线磁盘存储，以支持任何扩展的系统文件和以报告的 BBops/min 执行 ECperf 事务组合 8 小时所产生的持久数据库填充。

5.1.2 SUT 服务来自驱动程序的远程方法调用，并返回由 ECperf 引用 Bean 生成的结果，这可能涉及从 RDBMS 检索信息。必须仅使用 JDBC 从 ECperf 引用 Bean（或代表 Bean 的容器）访问数据库。

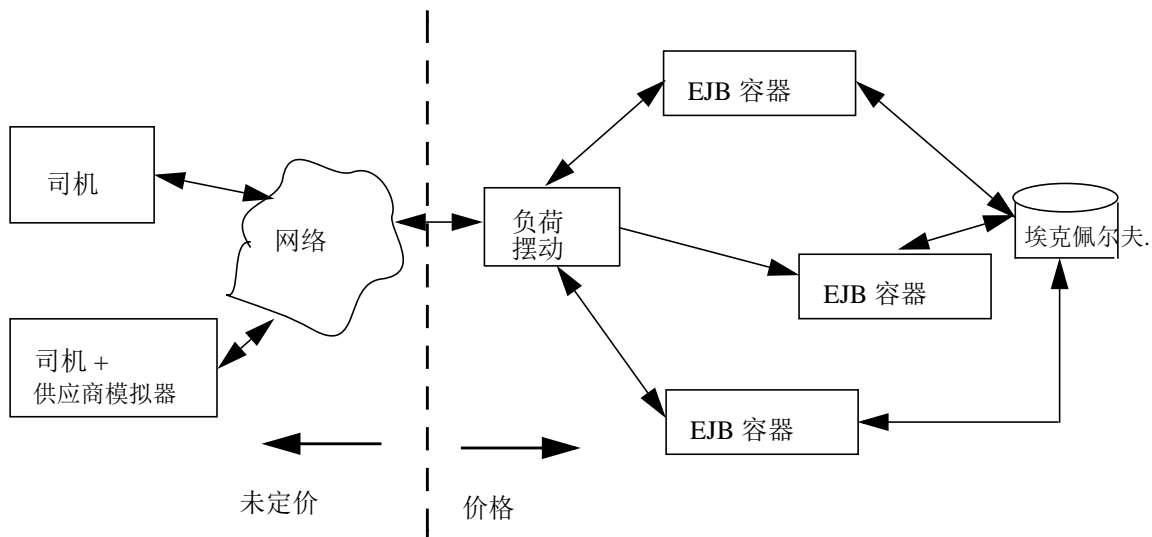
5.1.3 SUT 不得执行超出正在使用的服务器（EJB 容器、数据库服务器等）通常执行的操作之外的任何高速缓存操作。

**注释：** 目的是允许 EJB 容器和数据库服务器高速缓存正常工作，但不允许实现占用基准测试的有限性质，并高速缓存通常从服务器检索的信息。

---

## 5.2 配置示例

5.2.1 图 7 显示了集中式工作负载的驱动程序和 SUT 组件的示例布局。在此示例中，SUT 由 3 个容器组成，用于处理负载。  
容器之间的多路复用/负载平衡由商业产品完成。所有域都合并到一个数据库中。



---

图 7：集中式工作负载的示例配置

5.2.2 图 8 显示了分布式工作的驱动程序和 SUT 组件的示例布局-  
负荷。在此布局中，每个域都有一个容器，用于管理自己的数据-

基础

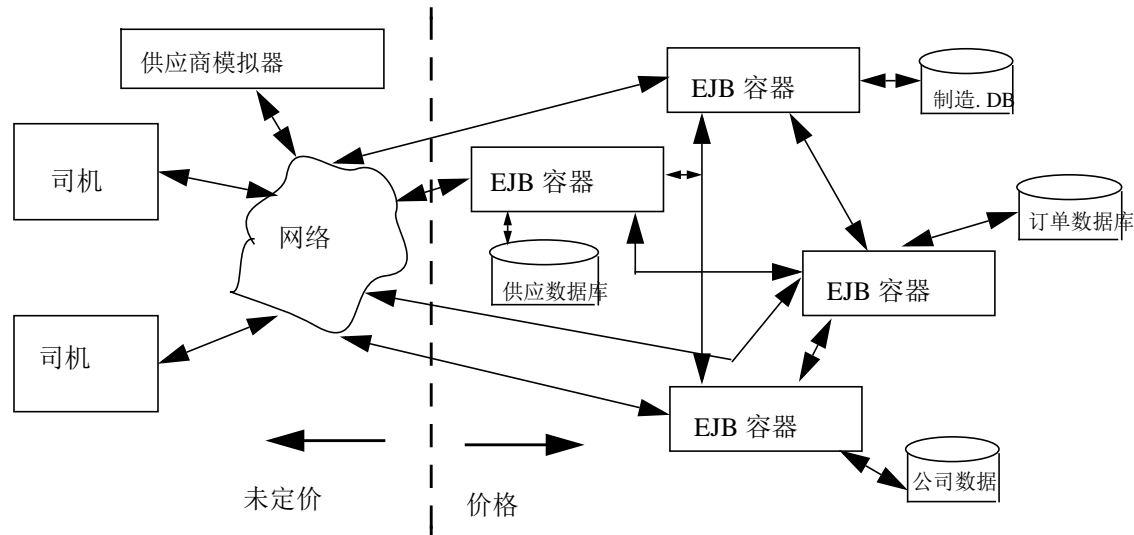


图 8:

分布式工作负载的配置示例

### 6.1 性价比指标

#### 6.1.1

除了第4.8条中定义的性能指标外，ECperf还包括一个价格/性能指标，定义为价格/BBops/min，即SUT的总价格除以报告的BBops/min。

6.1.2 价格/性能指标必须向上舍入到货币单位的最低有效数字。例如，如果总价格为 5,734,417 美元，报告的吞吐量为 105.12 BBops/分钟，则价格/性能为 54,551 美元/BBops。

### 6.2 定价组件

#### 6.2.1 如何

6.2.1.1 必须包括SUT的全部价格（见第5.1条），包括1年期的所有硬件，软件和支持。

6.2.1.2 所有定价的硬件组件必须是新的，未经翻新或以前拥有。

6.2.1.3 ECperf 的用户数针对每个域，并定义为等于测试配置中的 EC客户机数。上述用户数量的任何使用定价都应基于提供定价组件的公司的定价策略。

#### 6.2.2 附加组件

6.2.2.1 如果SUT的操作、管理或维护明确需要，则操作员控制台和备用设备等其他组件也必须定价。

6.2.2.2 如果在安装或更新期间需要从特定设备加载软件，则必须为该设备定价。

#### 6.2.3 支持

##### 6.2.3.1

硬件和软件支持必须定价为每周7天，每天24小时，无论是现场服务，还是通过中央支持设施提供的标准服务。

##### 6.2.3.2

如果中央支持设施已定价，则在测量运行期间，必须在SUT上安装并运行连接到中央支架所需的所有硬件和软件并使其正常工作。

##### 6.2.3.3

硬件支持请求的响应时间不得超过 4 小时，其组件



必须更换 SUT 才能返回到测试的配置。

#### 6.2.3.4 如果客户在4小时内确定

要更换的部件出现故障，则允许以备件代替硬件支持要求。必须为指定零件的额外 10%，至少 2 个，定价。在 7 天内提供现场更换的备件支持服务也必须为 1 年支持期定价。

#### 6.2.3.5 软件的移植请求必须在 4 小时内包括问题确认。软件支持必须在 1 年内至少包括 1 个软件更新分发，包括软件及其文档。

## 6.3 定价规则

### 6.3.1 定价规则 的目的是

按客户支付的全价对测试的系统进行定价。不允许假设该客户进行的其他购买。这是一次性的独立购买。

### 6.3.2

使用的所有硬件和软件必须由客户订购。对于任何尚未普遍发布的产品，完整披露报告必须包括承诺的一般交付日期（见条款7.3.1）。该日期不得超过完整披露提交日期后的 3 个月。所有使用的产品必须是道具的最终版本，而不是原型。一般发布的产品性能必须在发布结果的2%以内。

**评论：**目的是测试客户将使用的产品，而不是原型。可以使用产品的测试版，前提是 G 能源可用性（GA）在 3 个月内。

### 6.3.3 使用的所有定价必须是标价。不允许任何形式的折扣。

### 6.3.4 只要此套餐是标准产品，就可以 使用套餐定价。

### 6.3.5 所有定价均应以当前单位为单位，该单位为该国家/地区的客户将用于支付系统并反映当地零售定价。价格应向上舍入到所用货币单位的最低有效数字。例如，所有美国定价都应四舍五入到最接近的美元。

### 6.3.6 必须披露所有定价来源和价格生效日期。

### 6.3.7

由第三方（即非测试主办方）提供的所有项目必须明确说明。每个第三方供应商的物品和价格必须单独列出。

### 6.3.8

完整披露报告中显示的定价必须反映客户在明细账单上看到的详细程度。有关定价披露示例，请参阅附录 B。

#### 6.3.9

结果可以在另一个国家重新发布，只有蓖麻毒素g发生变化，允许替换原始完整披露中的本地成分，提供替代产品

以相同的产品描述或规格出售。

**评论：其目的是通过**允许出于不影响性能的国家特定原因（例如电压、产品编号、工业/安全、键盘差异等）替换设备来鼓励当地国家/地区定价。

## 第 7 条 - 全面披露

### 7.1 术语定义

7.1.1 “完全披露” 一词是指报告基准结果时必须提供的信息。

7.1.2 术语“全面披露报告” 是指PDF或HTML表格的可打印报告，该报告是完全披露的一部分。

7.1.3 术语“完全披露存档”是指作为完全披露一部分的文件的软拷贝存档。

7.1.4 术语摘要声明

是指在报告基准时必须提供的基准摘要信息。摘要声明也是完整披露报告的一部分。

### 7.2 一般全面披露要求

7.2.1 为了使结果被认为符合ECperf基准规范，需要进行全面披露。

**第1条：**本披露的目的是让任何人都可以在给定适当的文档和软件的情况下复制该基准的结果。

**评论2：**在以下各节中，如果没有具体提及必须披露的地点，则必须在完整披露报告中进行披露。存档或 Summay 声明中的披露被明确指出。

7.2.2

完整披露报告中各节的顺序和标题必须与ECperf标准规范（即本文件）中各节的顺序和标题一致。目的是让读者尽可能轻松地比较和对比不同 Full Dis闭包报告中的材料。

7.2.3 摘要声明必须在全面披露报告的开头附近包含（见第7.3条）。

7.2.4 必须提供一份声明，说明基准保荐人和其他参与公司。

7.2.5 必须提供机器人h测量和定价配置的图表，并附有描述

差异。这包括但不限于：

- 处理器的数量和类型。
- 分配的内存大小，以及测试特有的内存的任何特定映射/分区。
- 磁盘机（和控制器，如果适用）的数量和类型。
- 测试中物理使用的 LAN（例如以太网）连接数，包括路由器等。

- 软件组件（例如，EJB 服务器/容器、DBMS、客户机进程、软件负载平衡器等）的类型和运行时执行位置。

## 7.3 总结声明

7.3.1 摘要语句是 ECperf 基准测试和运行结果的高级视图。摘要说明的一个例子见附录B。摘要报表必须以与所报告的基准相同的格式包含此示例中包含的所有信息。

7.3.2 驱动程序摘要报告必须作为摘要声明的一部分出现。其中包括 ECperf.summary、Orders.summary 和 Mfg.summary 文件。

7.3.3 驱动程序为运行验证生成的 Audit.report 文件必须显示为摘要语句的一部分。

## 7.4 条款 4 伸缩和运行规则相关项

7.4.1 必须标识 所有使用的市售

软件产品。必须为所有客户可调参数和选项提供设置，这些参数和选项已从实际产品中的默认值更改，包括但不限于：

- 操作系统选项。
- J2EE 服务器选项。
- 用于供应商域和模拟器的 Web 容器选项。
- EJB 容器选项。
- 数据库选项。

**评论1：**通过提供所有参数和选项的完整列表可以满足此要求。

7.4.2 对于 J2EE 兼容产品的新版本，应指明其预计通过 J2EE 兼容性测试套件（CTS）的日期。

7.4.3 必须披露用于加载数据库的订单注入率。

7.4.4 完整披露存档必须包括 所有表定义语句和用于设置数据库的所有其他语句。

7.4.5 如果修改了 ECperf 工具包 中的加载程序（见条款 4.4.4），则所有此类修改

*ECperf™ Benchmark*

*Sun Microsystems Inc.*

---

必须披露，修改后的程序必须包含在完全披露档案中。

## 7.4.6

用于为数据库设备创建任何逻辑卷的所有脚本/程序必须作为完整披露存档的一部分包含在内。必须明确描述表和日志在所有媒体上的分布。

7.4.7 必须公开 EJB 容器使用的持久性类型，无论是 CMP、BMP 还是混合模式。如果使用混合模式，则必须枚举使用 CMP 和 BMP 部署的 Bean 列表。

7.4.8 如果修改了 ECperf 参考 Bean（见条款 4.1.3），则描述修改的声明必须出现在完整披露报告中，并且修改后的代码必须包含在完整披露存档中。

7.4.9 所有使用 d 的部署描述符都必须包含在完整披露存档中。

7.4.10 必须

披露重现性运行的 BBops/min（见条款 4.9.2）。重现性运行的整个输出目录必须包含在名为 RepeatRun 的指导中的完全披露存档中。

7.4.11 必须绘制所有事务响应时间的频率分布图（见条款 4.10.1）。

7.4.12 必须报告工作订单吞吐量与运行时间的关系图（见条款 4.10.2）。

7.4.13 用于运行 ACID 测试的脚本 s/程序及其输出必须包含在完整披露存档中。

## 7.5 第 5 条 SUT 和驱动程序相关项目

7.5.1 如果有任何软件/硬件用于影响超出基本 IP 路由的网络流量和切换，必须披露其他软件/硬件和设置。参见条款 5.1.1。

7.5.2 驱动程序的输入参数必须通过包含配置/运行属性文件和 bin/driver.sh 脚本来公开，该脚本用于在完全披露存档中运行基准测试。如果启动器包被修改，其源必须包含在完全披露存档中。

7.5.3 必须披露测试/定价配置中使用的网络带宽。

7.5.4 必须披露驱动程序用于与 SUT 通信的协议（例如 RMI / IIOP）。

7.5.5 如果驱动程序系统执行条款 4.12.5 中定义的任何负载平衡功能，则必须披露这些功能的详细信息。

7.5.6 使用的客户端系统的数量和类型，以及处理器的数量和类型，内存必须公开 ORY 和网络配置。

## 7.6 第 6 条 定价相关项目

### 7.6.1

必须报告定价系统中使用的硬件和软件的详细清单。每个单独或可提取的物料必须具有供应商部件号、描述和发行/修订级别，以及正式发布状态或承诺交付日期。如果使用封装定价，则必须披露封装的供应商部件号和唯一标识封装每个组件的说明。还必须报告价格来源和价格生效日期。

7.6.2 必须报告整个配置的总价，包括：硬件、软件和维护费用。建议对 Sepa 费率组件进行定价。

### 7.6.3

必须报告价格计算中使用的产品的正式供货日期（供货日期）。当定价系统包含具有不同可用性日期的产品时，定价系统的报告可用性日期必须是所有组件承诺可用的日期。

7.6.4 对于任何使用定价，保荐人必须披露：

- 组件定价时使用的级别。
- 允许此类定价的公司政策声明。

**注释：**使用定价可能包括但不限于操作系统、EJB 服务器和数据库服务器软件。

### 7.6.5

系统定价应包括以下组件的小计：使用的硬件、服务器软件和网络组件。标准定价表的示例如附录 B 所示。

7.6.6 系统定价必须包括使用非赞助公司品牌的行项目指示。系统定价还必须包括第三方定价的行项目指示。请参阅附录 B 中的示例。

## 第 8 条 - 结果的处理

该条款描述了 结果发布、质疑和撤回的流程。

### 8.1 结果提交

当测试发起人决定发布 ECperf 基准测试结果时，他必须生成第 7 条中所述的完整披露，并通过电子邮件将其提交给 ECperf 审查委员会 [ecperf\\_review@eng.sun.com](mailto:ecperf_review@eng.sun.com)。ECperf 审查委员会至少有 2 周的时间来审查结果，并将在 2 周后的定期会议上投票决定是否接受结果。

在检讨期间，Review 委员会成员会严格保密所有检讨资料。在成员公司内部，此信息应仅与提供对结果进行彻底审查所需的人员共享。

在审查结束时，如果没有与给定提交相关的未决起诉，则结果将获得批准，并在 ECperf 结果网站上发布完整披露。一旦发布，披露中包含的信息将被视为公开。

如果某一提交在其审查周期结束时存在未决问题，而这些问题尚未也无法解决，以使大多数委员会成员满意，则委员会投票认为该提交不符合要求，不应发布。保荐人被告知其结果中的问题，包括规范中未满足的所有相关条款。

如果对提交的内容进行了细微的更正，例如不影响性能的调整选项或产品名称的文档，则更正后的结果将保留在与原始提交相同的发布时间表上。如果进行了重大更正，例如新运行、不同的性能结果或不同的系统配置，则更正后的结果将作为新提交开始审查。

为了确保提交者包含所有适当的信息，而不会不公平地损害其技术，所有审查结果的成员必须遵循以下准则：

- 在原始提交完成其审查周期之前，其他成员不得使用完整披露中出现的新的和独特的优化技术。

提交者必须接受，试图超越目前正在审查的结果的竞争对手被认为是常见的做法，不被禁止。参赛者还可以随时使用新闻稿、产品文档、以前发布的 ECperf 结果等中的公开信息。

为了确保对产品已用于产生基准结果的供应商的公平性，以下规则适用：

- 在结果审查期间，如果成员认为该产品对基准的性能至关重要，则成员可以反对使用其硬件或软件产品发布结果。如果提出此类异议，结果将保留六（6）周，除非提前撤回反对意见。在预扣期内，反对者可以就同一配置提交更好的结果。在这种情况下，将审查反对者的结果

首先，如果被接受，原始提交将被拒绝。如果反对者不选择发布，那么如果发现原始提交符合要求，将被接受。

## 8.2 已发布结果中的违规行为

如果ECperf专家组在遵守规范方面出现重大问题，也可以对先前发布的结果进行重新审查。专家组保留在发现已发布结果违反规范时将其标记为不合规的权利。

已建立以下过程来处理在已发布的结果中发现违规的情况：

- 任何人都可以通过指定规范中被违反的条款以及违规的发生方式来对已发布的结果提出质疑。质疑必须发送给ECperf专家组。
- 专家组将评估挑战，并听取挑战者和基准赞助商的论点。然后，它将决定结果是否违规以及违规的性质。
- **轻微**违规通常涉及文档错误，发起人可以通过提交新的全面披露来纠正。修订后的披露应仅包含对发现错误的修复，并将替换当前发布的结果。
- **重大**违规将导致从结果网站上发布的摘要中删除指标，并将该列替换为“NC”（不合规）。完整披露将从网站上删除，并替换为对结果被发现不合规原因的解释。

## 8.3 撤销结果

保荐人可以随时选择通过向ECperf专家组发送通知来撤回已发布的ed结果。结果将从ECperf结果网站的摘要页面中删除，并移至“撤回的结果”类别。撤回的结果不能公开引用或用于发布结果的其他供应商进行比较。

## 8.4 合理使用 ECperf 结果

结果发布的目的是允许供应商在新闻稿、营销材料等中宣传其产品的性能。但是，供应商在使用ECperf结果时需要尽职尽责，并应遵守以下规则：

- **ECperf**  
结果应始终使用性能和性价比指标以及运行它的类别（集中式或分布式）进行引用。应包含指向 ECperf 官方结果网站上的结果的指针。
- **术语 ECperf** 应仅用于指 ECperf  
结果网站上发布的结果，这些结果遵循本条款中所述的规则。具体来说，在实验室中获得的外推或结果不能称为ECperf结果。
- 不允许与“已撤回”或“不合规”结果进行比较。



## 附录 A Bean 接口定义

本附录描述了各种企业 Bean 及其接口。请注意，此处的接口定义可能不是最新的。在任何时候，ECperf 工具包都包含必须使用的源代码的最终版本。

### A.1 订单 接口

```
/* * 版权所有 (c) 1998-2000 Sun Microsystems, Inc.保留所有权利。
* 本软件是 Sun Microsystems, Inc. 的专有信息。
* 使用受许可条款的约束。

* * $Id: 订单.java, v 1.5 2000/10/05 19: 34: 18 香蒂 Exp $

* */ package com.sun.ecperf.orders.orderses.ejb;

import javax.ejb.*;

import java.rmi.*;

金额 com.sun.ecperf.common.*; amount
com.sun.ecperf.orders.helper.*;

/**
* 这是OrderSessionBean的接口，它是一个包装器
* 对于订单和订单行实体 bean。会话 Bean 还 * 实现了 getCustStatus
方法来检索属于特定客户的所有订单 */ 公共接口 OrderSes 扩展了 EJBObject {

/**
* newOrder: 为客户输入订单
* @param customerId - 客户的ID
* @参数 数量
* @return int - 订单的ID
* @抛出数据 完整性异常
```

```

* @exception 不足信用异常 - 如果客户信用检查失败 * @exception 创建异常 - 如果创建订单失败
* @exception 远程异常 - 如果存在系统故障 */ public int newOrder (int customerId, ItemQuantity[]
数量) 抛出 InsufficientCreditException, DataIntegrityException, RemoteException,
CreateException;

/**
* 更改订单: 更改现有客户订单 * @param 订单 ID - 正在更改的订单的 ID
* @参数 数量
* @exception 不足信用异常 - 如果客户信用检查失败 * @exception 远程异常 -
如果存在系统故障 */ 公共无效更改订单 (int orderId, ItemQuantity[] 数量)
抛出不足信用异常, 远程异常;

/**
* 取消订单: 取消现有客户订单
* @param orderId - 正在更改的订单的 ID * @exception 远程异常 - 如果出现系统故障 */
公共无效取消订单 (int orderId) 抛出远程异常;

/**
* getOrderStatus: 检索订单的状态
* @参数 订单 ID - 订单的 ID * @return 订单状态对象
*
* @throws DataIntegrityException * @exception RemoteException - 如果存在系统故障 */ public
OrderStatus getOrderStatus (int orderId) 抛出 DataIntegrityException, RemoteException;

/** * 获取客户所有订单的状态
*
* @param customerId int customer id
* @return 客户状态对象数组 (每个订单一个) *
* @throws DataIntegrityException * @exception RemoteException - 如果存在系统故障 */ public
CustomerStatus[] getCustomerStatus (int customerId) 抛出 DataIntegrityException,
RemoteException; }

```

## A.2 项目数量.java

```

/* * 版权所有 (c) 1998-2000 Sun Microsystems, Inc.保留所有权利。

```

```
* 本软件是 Sun Microsystems, Inc. 的专有信息。
* 使用受许可条款的约束。

* * $Id: 项目数量.java, v 1.2 2000/10/04 20: 52: 52 拉梅什 Exp $
* */ package com.sun.ecperf.orders.helper;

/** * 此类的对象用于表示订单行项目中的项目、数量对 *。 */ public class
ItemQuantity 实现 java.io.Serializable {

    /** * 构造函数 项数量
    *
    *
    * @param itemId    * @param itemQuantity
    * */ public ItemQuantity (String itemId, int itemQuantity) { this.itemId = itemId;
    this.itemQuantity = itemQuantity; }

    公共字符串 项 ID;          public int
itemQuantity; }
```

### A.3 客户信息.java

```
/** * 版权所有 (c) 1998-2000 Sun Microsystems, Inc.保留所有权利。
* 本软件是 Sun Microsystems, Inc. 的专有信息。
* 使用受许可条款的约束。
* * $Id: 客户信息.java, v 1.3 2000/10/04 19: 24: 10 ramesh Exp $
* */ package com.sun.ecperf.common;

导入 java.io.*;

/** * 此类是传递给 OrderCustomer bean 的 addCustomer * 方法的 CustomerInfo 对象
* * @author Shanti Subramanyam */ public class CustomerInfo 实现 Serializable {

    /**
```

```

* Constructor CustomerInfo
*
*
* @参数 第一
* @参数 最后
* @参数 地址
* @参数 联系人
* @参数 信用
* @参数 信用限制
* @参数 余额 * @param 年初至今付款
* */ 公共客户信息（字符串在前，字符串在后，地址地址，
                                字符串联系人，字符串信用，双信用限制，
                                双倍余额，双倍年初至今付款）{

this.firstName = first;    this.lastName = last;    this.address = 地址;    这个.联系 = 联系;    this.since
= new java.sql.Date (new java.util.Date () . 获取时间 ());    这个信用=信用;
this.creditLimit = creditLimit;    这个.余额 = 余额;    这.年初至今付款 = 年初至今付款; }

public Integer customerId; 公共字符串名字;
公共字符串姓氏; 公共地址地址;
公共字符串联系人; 公共字符串信用;
公共双信用额度; 公共java.sql.日期以来;
公共双倍余额; 公开双倍年初至今支付;
}

```

#### A.4 客户状态.java

```

/* * 版权所有 (c) 1998-2000 Sun Microsystems, Inc.保留所有权利。
* 本软件是 Sun Microsystems, Inc. 的专有信息。
* 使用受 许可条款的约束。

* * $Id: 客户状态.java, v 1.2 2000/10/04 20: 52: 51 ramesh Exp $
* */ package com.sun.ecperf.orders.helper;

import java.io.Serializable;

```



```
* @param info * @return id - 新创建客户的客户 ID */ public int addCustomer (CustomerInfo info)
    抛出 InvalidInfoException, DataIntegrityException, RemoteException;

/** * 方法 getPercentDiscount
 *
 *
 * @参数 客户ID
 * @参数 总计
 *
 * @返回
 *
 * @throws dataIntegrityException * @throws RemoteException
 * */ public double getPercentDiscount (int customerId, double total) 抛出 DataIntegrityException,
    RemoteException; }
```

## A.6 工作接口

```
/*
 * * 版权所有 (c) 1999-2000 Sun Microsystems, Inc.保留所有权利。
 * * 本软件是 Sun Microsystems, Inc. 的专有信息。
 * * 使用受许可条款的约束。 * @author 阿贾伊·米塔尔
 * * akmits@eng.sun.com 04/03/2000 */ package com.sun.ecperf.mfg.workorderses.ejb;

import javax.ejb.EJBObject;

import java.rmi.RemoteException;

/** * 此接口是 WorkOrderSes * 会话 Bean 的远程接口。此豆子是无状态的。
 *
 * @author Ajay Mittal (英语: a uthor Ajay Mittal)
 *
 * */ 公共接口 WorkOrderSes 扩展了 EJBObject { // 如果从制造应用程序 (驱动程序) 调用
```

```
/** * 计划工作订单的方法。
* @参数程序集 ID 程序集 ID
* @param qty 原始数量 * @param到期日 订单到期日期
*
* @return * 如果出现系统故障，则@exception RemoteException */ public Integer
scheduleWorkOrder (String assemblyId, int qty, java.sql.Date到期日期) 抛出远程异常;
如果从订单域调用

/** * 计划工作订单的方法。
* @参数销售ID 销售订单ID
* @param oLineId 订单行 ID
* @参数程序集 ID 程序集 ID
* @参数数量 原始数量 * @参数到期日期 订单到期日期
*
* @return * 如果存在系统故障，则@exception RemoteException */ public Integer
scheduleWorkOrder ( int salesId, int oLineId, String assemblyId, int qty, java.sql.Date
dueDate) 抛出 RemoteException;

/** * 完成工作订单：工作订单取消。
* * 将已完成的部分转移到库存。 * @param wid 工作订单 ID
* @return布尔 值 false (如果无法完成)
* @exception 远程异常 如果存在 * 通信或系统故障 */ 公共布尔值 completeWorkOrder (整数 wid)
抛出远程异常;

/** * cancelWorkOrder：工作订单取消。
*
* 将已完成的部分转移到库存。中止剩余工作订单 * @param wid 工作订单 ID
* @return布尔 值 false (如果无法完成)
* @exception 远程异常 如果存在 * 通信或系统故障 */ 公共布尔值取消工作订单 (整数 wid)
抛出远程异常;

/** * 在工作订单中完成数量
*
* @参数 wid 工作订单 编号
```

```

* @return int 状态
* @exception 远程异常 如果有 * 通信或系统故障 */ public int getWorkOrderCompleteQty (整数
wid) 抛出 RemoteException;

/** * 获取工作订单的状态
*
* @参数 wid          工作订单 编号
* @return int 状态
* @exception 远程异常 如果有 * 通信或系统故障 */ public int getWorkOrderStatus (整数 wid)
抛出远程异常; /** * 更新工作订单的状态
*
* @参数 wid          工作订单 编号
* @except          如果 存在 * 通信或系统故障 */ public void updateWorkOrder (Integer wid) 抛出
RemoteException; }

```

## A.7 大订单 接口

```

/*
* * 版权所有 (c) 1999-2000 Sun Microsystems, Inc.保留所有权利。
* * 本软件是 Sun Microsystems, Inc. 的专有信息。
* 使用受许可条款的约束。*/ package com.sun.ecperf.mfg.largeorderses.ejb;

import javax.ejb.EJBObject;

import java.rmi.RemoteException;

/** * 此接口是大订单 * 会话 Bean 的远程接口。此豆子是无状态的。 * @author
Agnes Jacob */ 公共接口 LargeOrderSes 扩展了 EJBObject {

/** * 方法 createLargeOrder
*
*

```



```
* @参数 订单ID
* @param oLineId
* @参数 程序集ID
* @参数数量 * @param到期日期
*
* @返回
* * @throws 远程异常
* */ public Integer createLargeOrder ( int orderId, int oLineId, String assemblyId, short qty,
    java.sql.Date dueDate) 抛出 RemoteException;

/** * 方法 findLargeOrders
*
*
* @返回
* * @throws 远程异常
* */ public java.util.Vector findLargeOrders () 抛出 RemoteException; }
```

## A.8 大订单信息.java

```
/** * 版权所有 (c) 1998-2000 Sun Microsystems, Inc.保留所有权利。
* 本软件是 Sun Microsystems, Inc. 的专有信息。
* 使用受许可条款的约束。
* * $Id: LargeOrderInfo.java, v 1.2 2000/10/04 20: 08: 52 ramesh Exp $
* */ package com.sun.ecperf.mfg.helper; import java.io.*;

/** * 类 大订单信息
*
*
* @作者
* @version %I%, %G% */ public class LargeOrderInfo 实现 Serializable {

    公共整数 ID; public int salesOrderId;
    public int orderLineNumber; 公共 字符串
    程序集 ID; 公共空头数量; public java.sql.date dueDate;
```

```

/**
 * 方法 重复
 *
 *
 * @返回
 * */ public LargeOrder Info duplicate ( ) { LargeOrderInfo loi = new LargeOrderInfo ( ) ;

        loi.id = this.id;          loi.salesOrderId      = this.salesOrderId;
loi.orderLineNumber = this.orderLineNumber;      loi.assemblyId      =
this.assemblyId;      loi.qty          = this.qty;      loi.dueDate      =
this.dueDate;

        返回 意向书;
    }
}

```

## A.9 买家 界面

```

/* * 版权所有 (c) 1998-2000 Sun Microsystems, Inc.保留所有权利。
 * 本软件是Sun Microsystems, Inc.的专有信息。
 * 使用受许可条款的约束。
 * * $Id:
 * */ package com.sun.ecperf.supplier.buyerses.ejb;

import javax.ejb.*;

导入 java.rmi.*;

金额 com.sun.ecperf.common.*;

/**
 * 这是 BuyerSessionBean 的接口。它是从制造域到供应商域的 * 接口。
 * Mfg 域反复调用 add 方法来添加组件

```

```

* 库存低。添加完所有组件后，制造域调用采购以发出采购订单。
* * 研究 方法getPOStatus，看看我们如何在Mfg中使用它。
* * @author 达米安·盖伊
* */ 公共接口 BuyerSes 扩展了 EJBOBJECT {

    /** * 添加：将发明中含量较低的组件添加到PO。 * @param component_id -
    要添加到采购订单的组件的 ID。

    * @param qty_required - 要购买的号码。
    * @异常远程异常 - 如果出现系统故障。 */ public void add (String component_id, int
    qty_required) 抛出 RemoteException;

    /** * 采购：发出采购订单。

    * @抛出 ECperfException * @exception 远程异常 - 如果出现系统故障。
    * @exception 创建异常 - 如果创建采购订单失败。 * @exception FinderException */ public void
    purchase () 抛出 RemoteException, FinderException, CreateException, ECperfException;

    /** * getPOStatus：检索未完成 PO 的状态信息。

    * @param poID - 采购订单的 ID。
    * @返回 POStatus - 对象 包含 PO 状态信息。
    * @异常远程异常 - 如果出现系统故障。

    */

    public POStatus getPOStatus (int poID) // 抛出 RemoteException;
}

```

## A.10 R 接收 接口

```

/*
* * 版权所有 (c) 1999-2000 Sun Microsystems, Inc.保留所有权利。
* * 本软件是 Sun Microsystems, Inc. 的专有信息。
* 使用受许可条款的约束。 * @author 阿贾伊·米塔尔
*
* akmits@eng.sun.com 04/03/2000 */ package
    com.sun.ecperf.mfg.receiveSes.ejb;

```

```
import javax.ejb.EJBObject;

import java.rmi.RemoteException;

/** * 此接口是 ReceiveSes * 会话 Bean 的远程接口 。此豆子是无状态的。
 * * @author 阿贾伊·米塔尔
 *
 * */ 公共接口 ReceiveSes 扩展了 EJBObject {

    /** * 将组件添加到库存的方法。
    * @param compoID 唯一组件 ID
    * @param numComponents 要添加 的组件数 *
    如果出现系统故障，则 @exception RemoteException */ public void
    addInventory (String compoID, int numComponents) 抛出
    RemoteException; }
```

## A.11 接收器 接口

```
/** * 版权所有 (c) 1998-2000 Sun Microsystems, Inc.保留所有权利。
 * 本软件是 Sun Microsystems, Inc. 的专有信息。
 * 使用受许可条款的约束。

 * $Id: 接收器.java, v 1.2 2000/10/04 22: 27: 48 ramesh Exp $
 *
 */ package com.sun.ecperf.supplier.receiveverses.ejb;

import 语句 import javax.ejb.*;

导入 java.rmi.*;

金额 com.sun.ecperf.supplier.helper.*;

/** * 无状态会话 bean 的 远程接口 * 接收器。
```

```

* * @author Damian Guy */ 公共接口 ReceiverSes 扩展了 EJBObject {

/** * 交付采购订单 - 表示采购订单的一部分已送达。

* @param del - 包含有关已交付的 POLine 的信息。 * @exception 远程异常 -
如果出现系统故障。 */ public void deliverPO (DeliveryInfo del) 抛出 RemoteException; }

```

## A.12 交货信息.java

```

/* * 版权所有 (c) 1998-2000 Sun Microsystems, Inc.保留所有权利。
* 本软件是 Sun Microsystems, Inc. 的专有信息。
* 使用受许可条款的约束。
* */ package com.sun.ecperf.supplier.helper; import java.io.Serializable;

/** * 此类是传递给 Receiver Bean 的 deliveredPO * 方法的 DeliveryInfo 对象。
* * @author Damian Guy */ 公共类 DeliveryInfo 实现 Serializable {

    public int poId; 公共国际
    line_number; 公共字符串 part_id;
    公共数量;

    /** * 构造函数 交付信息
    *
    *
    * @param poId
    * @参数 line_number
    * @参数 part_id
    * @参数 数量
    * */ public DeliveryInfo (int poId, int line_number, String part_id, int qty) {

        this.poId = poId; this.line_number =
        line_number; this.part_id = part_id; this.qty
        = qty; }
    }

```

---

## 附录 B 摘要陈述样本

---

本附录显示了一个摘要语句的示例。摘要声明必须作为全面披露的一部分，并出现在全面披露报告的开头（见第8.3条）。

摘要语句的第一页必须显示基准测试结果和高级配置信息。必须列出基准测试中使用的所有商业组件。

摘要报表的第二页是定价电子表格，必须遵守第 7 条中的要求。下面显示了定价电子表格示例。

简要说明的最后几页必须包括由

正在报告的基准运行的 ECperf 驱动程序。它们是 ECperf.summary、Orders.summary 和 Mfg.summary。

版本 1.0, 最终版本, 2001 年 5 月 29 日

版本 1.0, 最终版本, 2001 年 5 月 29 日

**XYZ 公司: BigServer 1234**  
**ABC Inc. : 超快 应用服务器 1.0**  
**ODS 软件 : 高级 DBMS 3.5**

**指标: n.nn BBops/min@Std\$xxx /BBops/min@Std**  
**可用日期: 日星期一 yyyy**  
**Bean 部署模式: BMP/CMP/混合配置:**

**显示 SUT 中**所有系统的图表

系统	软件	处理器	内存磁盘
大服务器 1234	超快 应用服务器 1.0 JDK 1.3 新操作系统 2.0	4x奔腾III 700兆赫 2MB 电子缓存	4国标 9国标 内部
小型服务器 2	高级数据库管理系统 3.5 新操作系统 2.0	2x奔腾III 450兆赫	1GB 1 快速 数组 4x9国标

1MB  
电子缓存

版本 1.0，最终版本，2001 年 5 月 29 日

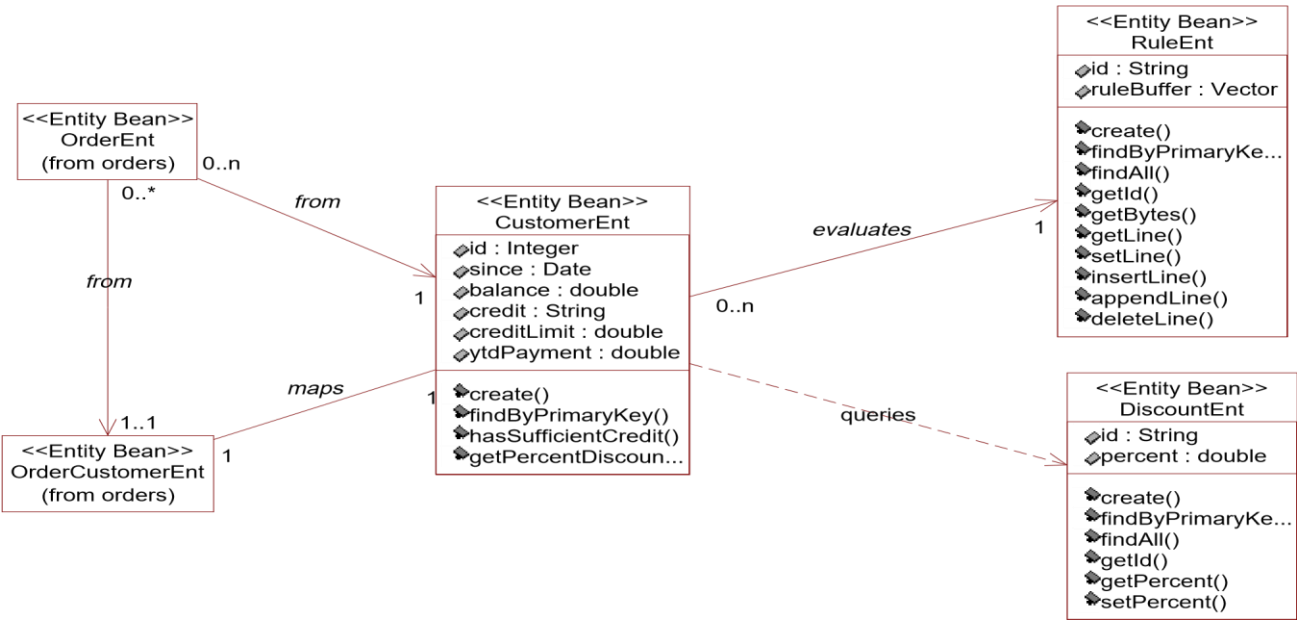
XYZ 公司		大服务器 1234			ECperf	版本	1.0
					标准工作负载		
描述	部件号	第三方 品牌定价	单价	数量	扩展 价格	维护 价格	
硬件							
1234 与 4x奔腾 III 700Mhz 2MB 电子缓存，9GB 硬盘	201-A	XYZ	15, 995	1	15,995	3,800	
128 MB 内存	8203	XYZ	11, 495	1	11,495		
14“ EPA SVGA 显示器	2600	XYZ	995	1	995		
带有 2xPentiumII 的小型服务器 2	SS12	XYZ	1, 195	1	1,195	999	
小计					29,680	4,799	
软件							
XYZ Unix	U123	XYZ	995		995		
超快 应用服务器	UFA1	美国广播公司	995	1	21000		
	RD123	.ODS	21, 000	1	10000	800	
卓越的数据库管理系统 1.0			10, 000	1			
小计					31,995	800	
总					61,675	5,599	
1 年拥有成本					67,274		
OPS@Std					25.34		
\$/OPS@Std					2654.85		

版本 1.0，最终版本，2001 年 5 月 29 日



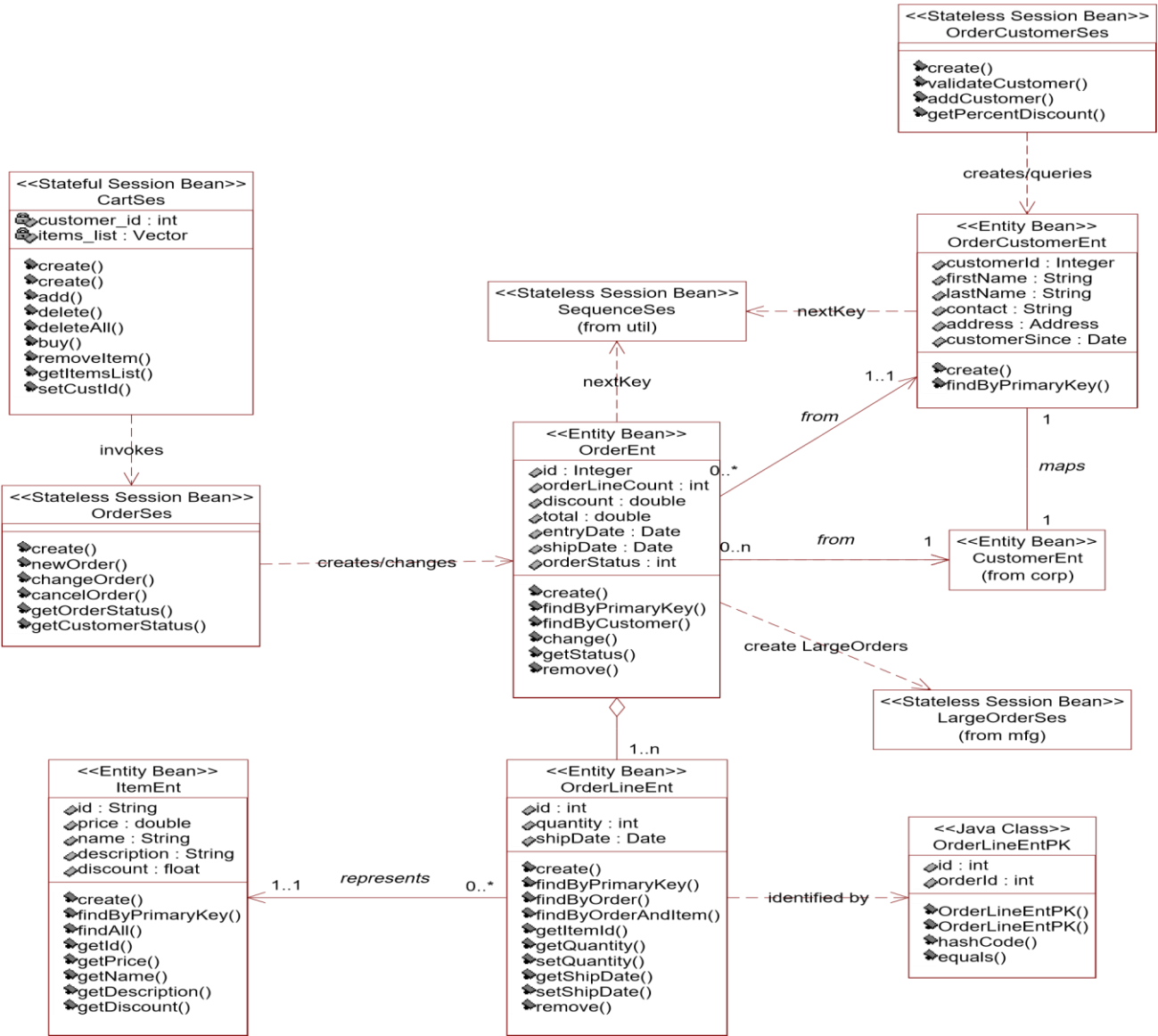
附录 C UML 图

C.1 企业域类图



文件: /home/akara/ecperf2/ecperf.mdl 周五 1月 26 日 12: 08: 39 2001 类图: 公司 / 主页 1

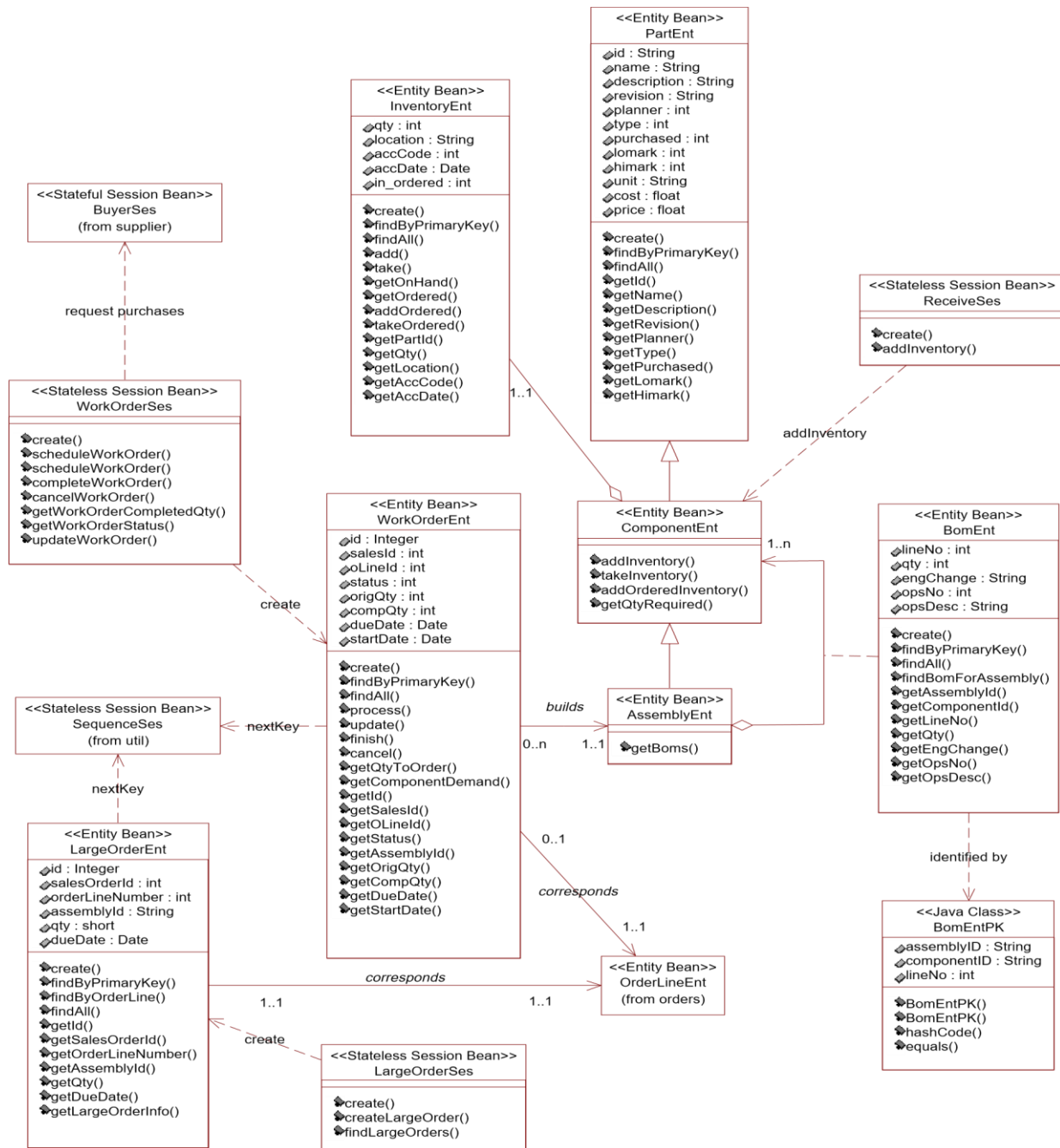
C.2 客户域类图



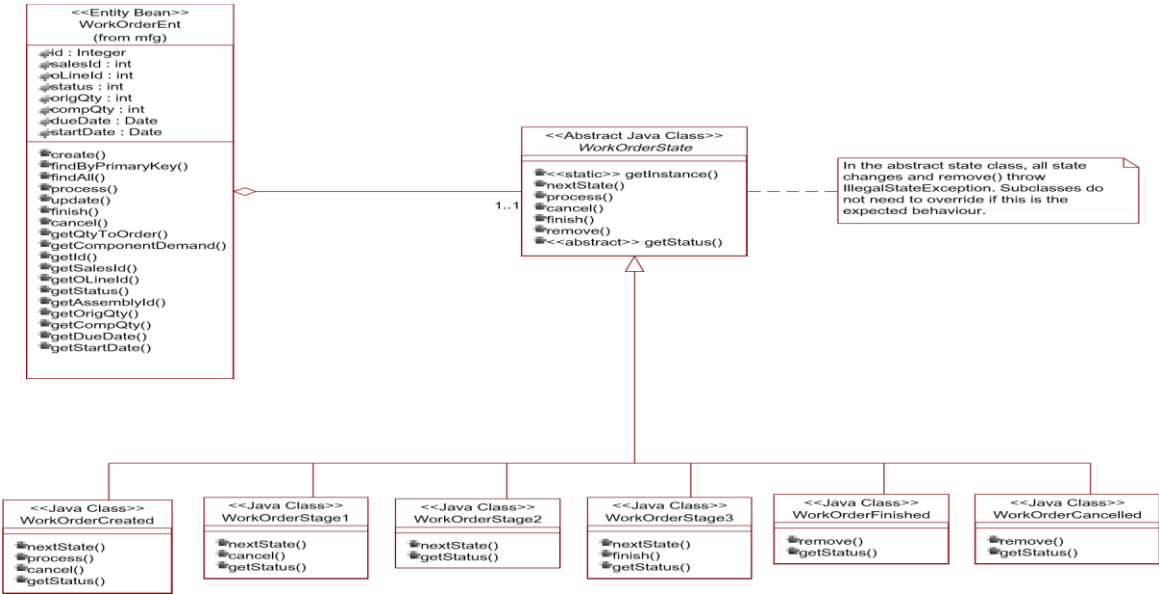
文件: /home/akara/ecperf2/ecperf.mdl 2001 年 1 月 26 日星期五 12: 12: 21 类图: 订单 / 主页 1

## C.3

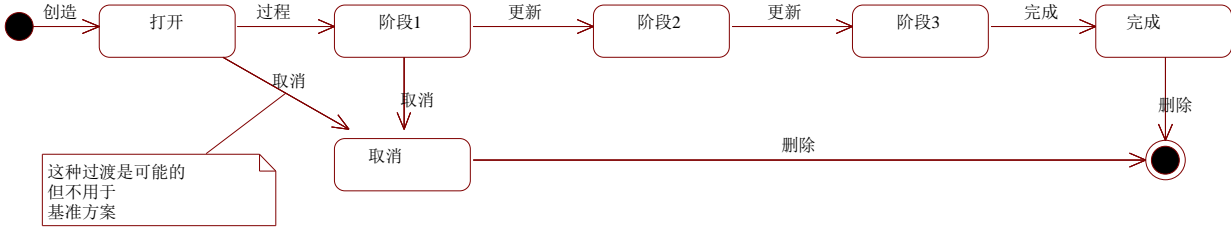
## 制造域类图



C.4 制造状态类

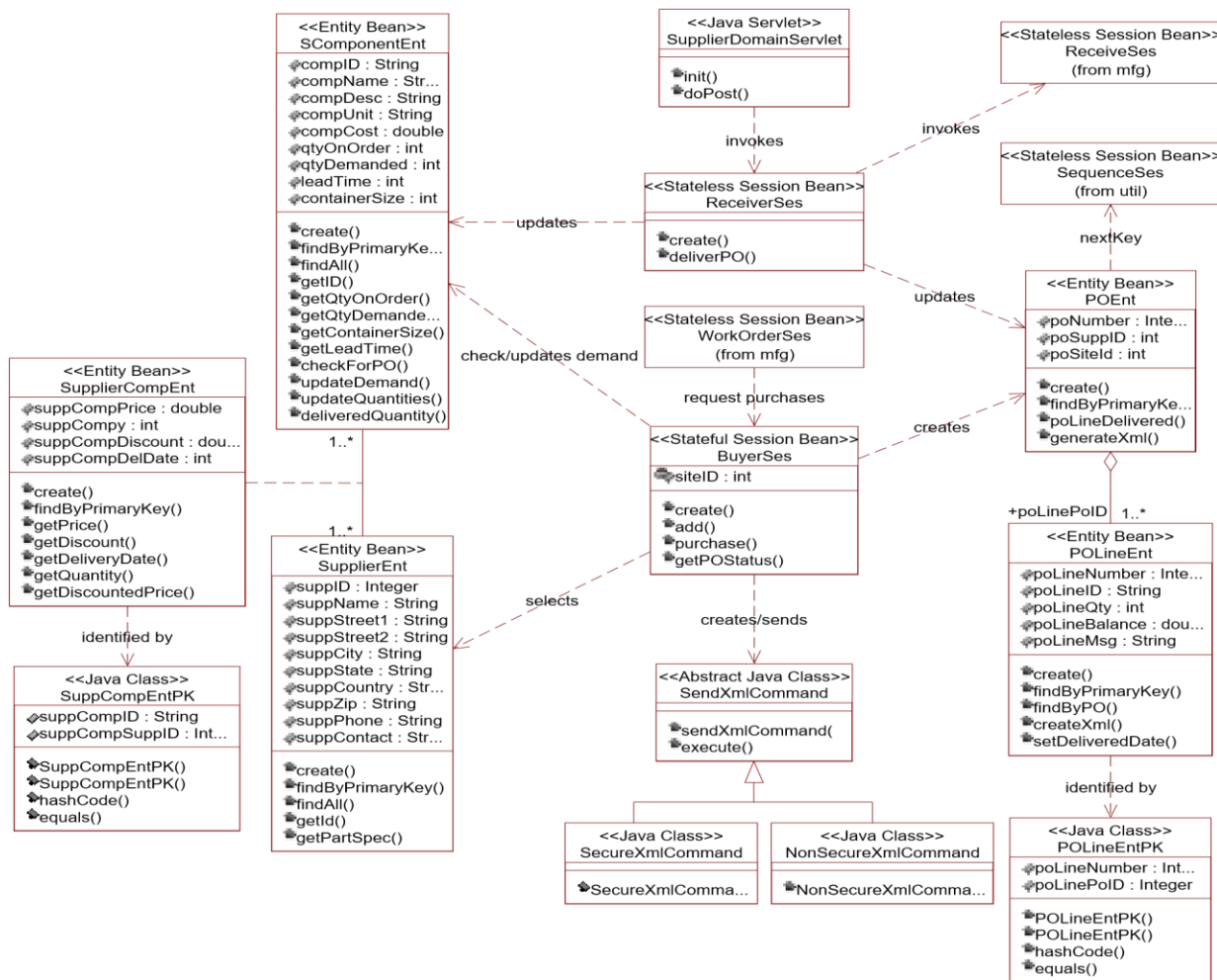


C.5 工作订单 状态转换



## C.6

## 供应商域类图



## C.7 密钥生成实用程序类图

