

# 基于移动代理在网格计算中的结构模型研究<sup>\*</sup>

韩光法<sup>1</sup> 王汝传<sup>1,2</sup>

(南京邮电学院计算机科学与技术系 南京210003)<sup>1</sup>

(南京大学计算机软件新技术国家重点实验室 南京210093)<sup>2</sup>

**摘要** 网格计算环境下的资源管理、作业调度和分配、负载均衡等问题成为网格计算发展的障碍。本文介绍了网格计算的体系结构和运用网格计算环境进行作业计算、资源管理、负载均衡等方法存在的问题,详细论述了在网格计算环境下引入移动代理技术对于解决网格计算问题的作用和优点。然后分别对网格计算中的移动 Agent 结构、资源管理模式、作业分配与调度方法、负载均衡等方面进行了研究。

**关键词** 网格计算,移动代理,层次模型,作业调度

## The Structure of Mobile Agent Based on Grid Computing

HAN Guang-Fa<sup>1</sup> WANG Ru-Chuan<sup>1,2</sup>

(Department of Computer Science and Technology, Nanjing University of Post and Telecommunications, Nanjing 210003)<sup>1</sup>

(State Key Laboratory for Novel Software Technology at Nanjing University, Nanjing 210093)<sup>2</sup>

**Abstract** The management of resource managing, task scheduling and distributing, and load balance become handicap of the development of grid computing. This paper discusses the system structure of the grid computing and the problems in task computing, resource management and load balance when operating on tasks base on the environment of grid computing. What's more, this paper also treatises the importance to introduce mobile agent technology to grid computing. Then we give the research base on the environment of grid computing with mobile agent as these aspects: the pattern of resource manage, the method of task distribute and schedule, the idea to balance load etc.

**Keywords** Grid computing, Mobile agent, Hierarchy model, Task schedule

## 1 引言

网格计算(Grid Computing)是一个分布式和并行计算的支持平台,是一种无缝、集成计算和协同环境。它可以作为虚拟的整体而使用在地理上分散的计算资源,如高速互连的异构计算机、数据库、科学仪器、文件和超级计算系统等。基于Internet的网格计算系统不但能使人们聚集分散的计算能力、形成超级计算的能力,解决诸如虚拟核爆炸、新药研制、气象预报和环境等重大科学研究和技术应用领域的问题,而且还能使人们共享和充分利用网络中的各种资源。

网格计算环境应具有如下几个方面的特征:(1)异构性:可以包含多种异构资源,如广域分布的多个管理域、多种类型的计算系统、不同类型的超级计算机体系结构、不同结构的操作系统和多层次应用软件等。(2)可扩放性(Scalable):随着用户的需求和各种计算资源的不断增加,网格计算能够不断地扩充自己的规模以适应这一需求。它可以从最初包含少数的资源发展到具有成千上万资源的大网格。(3)可适应性:由于网格计算环境中的各种资源较多,结构较为复杂,并且不同的资源在不同的时刻所表现的性能也不尽相同,网格的资源管理或应用应能动态地适应这种状况以合理地调用网格中可用的资源。(4)存在多级管理域:一个网格计算环境可能具有多个局域网、多个集群和成千上万个独立的计算资源组,并且由于构成网格计算系统的计算机资源通常属于不同的机构或虚拟组织(VO),它们使用不同的安全机制,因此它需要各个组

织共同参与解决多级管理域的问题。

正是由于网格环境下计算资源广域分布、异构、动态、有多个管理域、存在不同的存取花费模式等特点,使得资源的管理和调度十分复杂。目前没有一种科学、合理的管理模式能够处理所有的网格应用需求。大量的网格计算项目试图提供一个合适的资源管理结构,一般可分为集中式控制和本地应用控制两种类型。前者可以获得系统所有的资源信息以优化资源的使用,但扩放性(scalable)很差且存在单点失败;而后者则不能有效地了解全局资源信息,无法实现具有“网格意识”(grid-aware)的应用。因此,有必要为网格计算设计出一个较为科学的资源管理方法和合理的管理结构。

网格计算的自身特点,使得它拥有较为丰富的计算资源和作业处理能力,在其基础上调度思想也将由传统的以节省计算资源为主要因素的调度方法转到以优化调度策略、提高调度速度和可靠性、缩短作业处理时间为主要因素的调度方法上来。

## 2 网格计算的体系结构

根据 Lan Foster 的观点,网格计算体系结构应具有以下三个标准:1)资源的协调而仅仅是集中控制。2)使用标准的、开放的、通用的协调和接口。3)提供安全、可靠、高性能的服务。

对于任何一个可提交任务的“实体”来说,用户通过它提交自己的计算任务并希望网格计算环境为用户提供快速、无

<sup>\*</sup> 本课题得到国家自然科学基金(60173037和70271050)、江苏省自然科学基金(BK2003105)、国家高科技项目八六三(2002AA712037)、江苏省计算机信息处理技术重点实验室基金(kjs03061)资助。王汝传 教授,博士生导师,主要研究方向是计算机软件、计算机网络、信息安全、移动代理和虚拟现实技术。韩光法 硕士研究生,研究方向为基于网络的计算机软件技术。

缝、透明的服务。也就是用户看来,整个网格环境就是一台具有海量处理能力的计算机,其全部资源均来自提交任务的“实体”,用户无须知道复杂的资源管理和作业调度的细节。

复杂的网格计算环境允许同时有多个复杂的计算任务存在,它们可能由不同的计算节点提出,也可能由不同的节点给出。对于不同节点提交的计算任务,网格计算环境应能合理地分配相应的资源以进行任务的计算和存储,而不是简单地将任务分隔放到各个计算单元中。同时,同一节点所提交的若干计算任务,计算环境应能提供一套机制以科学地进行任务迁移,将它们分配到各计算节点进行计算。这些节点可为同一局域网上的计算资源,也可以为不同局域网或异地的资源。因此,网格计算环境应能提供合理的路由策略以提交迁移的可靠性和安全性。

在处理海量数据的过程中,由于受到网格带宽等各方面的限制,我们必须对网络数据的通讯量进行有效地控制,使得系统在通讯量允许的情况下合理作业。同时,网格计算还应满足安全性、可靠性等要求,防止运行节点的非正常中断,最小化造成的损失等。

基于以上分析,我们得出网格计算结构应具有的特征如下:

- 每一个可能提交作业的实体应具有对等(P2P)的权限,即在满足各种协议和操作规程的前提下同等地使用网格计算资源。

- 可操作实体应能在各个不同管理域之间提供不同资源的“无缝”连接,并能动态、高效地获得与作业量相匹配的资源。

- 网格计算环境应能面向各级可操作实体的结构,以方便作业的处理和资源的有效分配、使用和协调资源的冲突。

- 各虚拟组织(Virtual Organization)负责各自的作业调度并负责管理该作业所需的各种资源;应能和其它虚拟组织和对应资源服务器动态地交互信息以适时了解各自的资源信息和作业处理信息,根据情况动态地修改自己的管理决策、协调资源“争用”、资源的“独占”和“开放性”等问题。

- 各个独立的资源节点对于异域的实体是不可见的,即独立资源的信息和使用权限只能通过其所在的局域网服务器授权获得。这样可以保证网格计算的有序性和局域网的“有限独立性”。

### 3 移动 Agent 在网格计算中的作用

移动代理是一个能在异构网络中自主地从一台主机迁移到另一台主机,并可与其他 Agent 或资源交互的程序,实际上它是 Agent 技术与分布式计算技术的结合体。移动代理适用于在地理上或逻辑上分布、自主或异构的节点间提供应用服务或中间件服务,对海量分布式信息进行搜索,建立快速的智能的搜索机制;在高度动态的环境下,能对多变的环境作出响应或自适应;Agent 应用程序能自主地处理失效或冲突,以进行再调度、再计划或资源再分配。同时,它还能与用户进行灵活的相互作用,在相互作用中智能地协助用户完成琐碎的工作。除此之外,移动 Agent 还能进行长期计划驱动的行为和短期实时响应行为,在复杂的或安全性很重要的应用程序中,还保证适宜的反映和应答时间。

在网格计算中引用移动 Agent 技术将会有以下优点:

- 1)在地域上分布的异构网格计算环境中能自主地将计算任务从一节点迁移到另一节点;并可与其它 Agent 或资源交互以实现作业和资源的管理和自适应。

- 2)移动 Agent 可以迁移到网格计算环境的各级客户服

务器或中央服务器上,与之进行本地高速通信,它不再占用网络资源,从而大大降低了网络的通讯量,并提高了网络资源的利用效率。

- 3)移动代理通过在各级客户服务器之间、客户服务器和服务器之间双向移动来传递对应的资源信息、负载信息、通讯量和任务执行序列等信息。这些信息作为资源管理、负载平衡、通讯调整、任务调度等的参考依据,移动 Agent 根据这些数据智能地判断管理域的情况并做出相应处理。这将大大改善系统的性能和智能化水平,提高网格计算的可靠性和执行效率。

- 4)移动 Agent 通过将服务器请求 Agent 动态移到各级 Server 端执行,使得此 Agent 较小依赖网络传输环节而直接面对要访问的服务器资源,从而避免了大量数据间的网络传送,降低了系统对网络带宽的依赖。

- 5)在网格计算中,移动 Agent 不需要统一的调度。由用户创建的 Agent 可以异步在不同计算节点运行,等任务完成再将结果传送给用户。同一用户或同一计算节点可创建多种 Agent,同时在一个或多个节点运行,形成并行求解的能力。

此外,移动 Agent 还具有自治性和智能路由等特性。基于规则的动态路由能满足复杂的网格计算结构下作业调度、资源管理、路由选择和非确定性的任务分解等问题。

因此,在网格计算中引入移动 Agent 技术将有助于解决网格计算中面临的各种问题。

## 4 基于移动代理用于网格计算的结构和模型

在网格计算环境下,采用基于移动代理的层次型管理模式能够较为方便地进行管理。在网格计算中,允许其中的任何授权节点提交计算任务,在对提交的作业进行处理的过程中,由于网格计算资源相对较为丰富,只要给它分配部分计算资源便可以完成计算任务,所以,为了有效地对作业进行管理,我们不允许将作业“平均”覆盖到网格计算的任一资源之中。如由 PC3提交的任务 Task,当 PC3和 PC4能按规定完成计算任务时,Task 只占用 PC3和 PC4资源,其他资源供其它任务调用;如果 PC3和 PC4无法完成规定的任务,系统将通过移动代理占用上一级部分或全部资源(如 PC5),如此类推直到满足处理作业的要求为止。

由此可知,各级服务器只负责监视系统性能和作业的调度,并不负责作业的处理,这样大大减少了服务器的负载,有助于提高网格计算的执行效率和性能。

使中央服务器产生中央管理代理 Center-Manager-Agent),并为各级客户服务器产生具体的客户服务管理代理(Client-Manager-Agent),它们用于管理自己所辖区域的资源信息和作业处理,并能与其它合法的 Agent 进行交互以改变自己的内部状态和外部特征以指导自己的行为。提交任务的计算节点向自己所属的 Client-Manager-Agent 提交自己的任务请求并产生一个任务请求代理(Task-Require-Agent)。Client-Manager-Agent 接收并分析该 Task-Require-Agent 请求,如果请求合法,它为该任务生成专用的任务代理(Task-Agent)来处理任务的请求。Task-Agent 评估该客户端服务器所辖计算资源的信息并合理地进行任务的分配。其示意如图1所示。

当该局域网无法按要求的指标完成作业处理时,上一级服务器必须报告并请求获得更多的计算资源。Task-Agent 向 Client-Manager-Agent 发出剩余作业信息,客户管理代理(Client-Manager-Agent)向上一级服务器派出一个作业处理请求,上一级客户管理代理 Client-Manager-Agent 接收并评

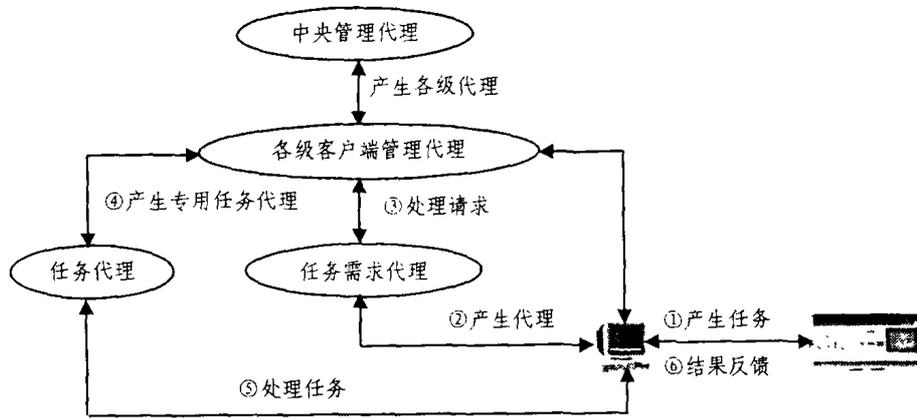


图1 移动代理创建示意图

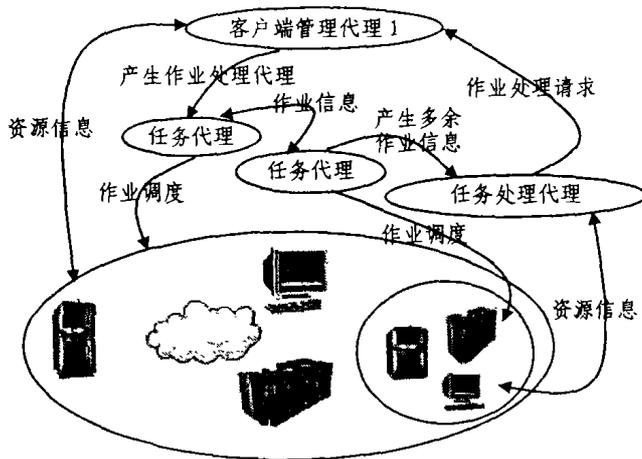


图2 多级 Agent 任务处理示意图

估该请求,然后为该请求产生出专用的任务处理代理,任务处理代理与本服务器管理代理交互得到区域可用资源信息并计算获得最佳任务调度方案,然后经过路由将任务调度到指定的计算节点中,如图2所示。这样便通过各种 Agent 将提交的任务调度到各种对应计算节点中。

在很多情况下,一个网格计算环境会有多个计算任务,在同一个局域网下也可能有多个计算任务存在,并且它所处理的任务可能为多个总任务中间的若干部分。因此,用单独的 Client-Manager-Agent 管理资源信息和资源的负载显然不是一个好的方法,因为这样做不但增加 Client-Manager-Agent 工作量和网络通讯量、使负载平衡造成混乱、任务无法正确调度等严重问题。

在网格计算中解决该类问题的一般方法根据应用的技术不同而分为轮询 DNS、硬件解决方案、协商式处理和流量分发等。DNS 方法在 DNS 服务器中设定对同一个 Internet 主机

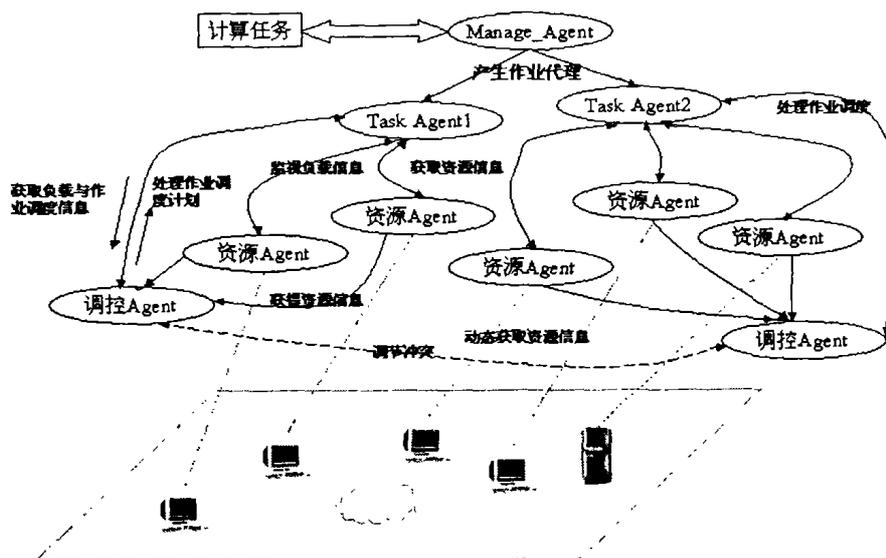


图3 Agent 调控任务/资源示意图

名的多个 IP 地址的映射,在 DNS 收到查询主机的请求时,系统就会循环地将所有对应的 IP 地址逐个返回。但是它有两个致命的缺点:只能实现对基于 Internet 主机名请求的负载均衡,而不是 IP 地址;DNS 服务器会不加识别地将可能存在节点故障的 IP 地址返回给查询方。

因此,在产生确定的 Task Agent 对作业处理的同时,系统将随之产生相应的资源代理来管理本计算任务占用的资源和所用计算节点的负载情况。这样在多任务处理的网格计算环境下,会同时有相同数量的 Task Agent 和资源 Agent 执行

相应的工作,资源 Agent 通过监视任务的执行和资源使用情况来发现负载异常,一旦平衡过载、或作业没有完全调度完毕的情况下某个计算节点资源的使用率低于期望的值,资源 Agent 就会向反馈负载信息。Task Agent 接受信息并将负载与作业调度信息送给一个成为调控 Agent 的代理,它根据这些信息进行资源的再分配、负载平衡以及任务的再分配。每一个调控 Agent 专门负责某个确定的任务并由该任务激活。

当多个调控 Agent 进行工作的过程中,可能会发生资源征用、路由等各种冲突,因此应当建立一定的机制来避免冲突

的发生,这就要求调控 Agent 之间具有良好的交互以协调各自的工作。如图3所示。

## 5 实验

为了测试上述方法的可行性,我们采用图3的网格计算模型,并设由 PC3提交一个计算任务,采用 RIPS(Runtime incremental parallel schedule)任务调度策略进行作业调度。在执行调度任务的过程中,可能会产生新的任务。当系统进行进一步调度时,上一轮调度没有执行完的任务与新产生的任务一同调度,两个阶段交替进行,直到全部的任务执行结束,为了简化问题,假定提交的任务有  $n$  个相互独立并有一定执行序列的子任务  $A_i(i=1, \dots, n)$ 。

建立一个 Task Agent 来完成子任务执行的同步,如果操作  $A_i$  依赖于操作  $A_j$ ,那么执行操作  $A_i$  的 Agent 在执行操作  $A_i$  前通知 Task Agent 等待操作  $A_j$  的完成,此时 Agent 即通知处于等待状态的 Agent:操作  $A_j$  已完成,可以继续执行。

为此,我们定义两个消息:Wait( $A_i$ )——它用于等待操作  $A_i$  的完成;Complete( $A_i$ )——用于表示操作  $A_i$  已完成。

假定执行操作  $A_i$  和  $A_j$  的 Agent 分别为 Agent <sub>$i$</sub>  和 Agent <sub>$j$</sub> ,协调者为 Agent <sub>$k$</sub> 。则给出的 Agent <sub>$i$</sub>  和 Agent <sub>$j$</sub>  的执行计划如下:

```
Agent $i$ :
Char * name; //站点名称
LOAD loadnum; //站点当前负载
LOAD maxloadnum; //站点最大负载
Int stanum; //站点编号
.....
Send Agent $k$  Wait( $A_j$ )
Wait Complete( $A_j$ )
Execute  $A_i$ 
...
Agent $j$ :
Char * name; //站点名称
LOAD loadnum; //站点当前负载
LOAD maxloadnum; //站点最大负载
Int stanum; //站点编号
.....
Execute  $A_j$ 
Send Agent $k$  Wait( $A_i$ )
...
```

其中,Agent <sub>$k$</sub>  需要提供的处理过程如下:

```
On Complete( $A_j$ )Do
Send Agent $i$  Complete( $A_j$ )
At Priority(xxx)
```

当所有的子任务调度、执行完毕之后,集成子任务的执行结果作为一组既得的目标知识,它即为整个任务的结果。

通过作为协调者的 Agent 向所有的 Task Agent 搜集它们执行操作所获得的知识,便可以得到任务的执行结果。

**结束语** 本文介绍了网格计算的体系结构和运用网格计算环境进行作业计算、资源管理、负载平衡等方法存在的问题,详细论述了在网格计算环境下引入移动代理技术对于解决网格计算问题的作用和优点。然后分别对网格计算中的移动 Agent 结构、资源管理模式、作业分配与调度方法、负载平衡等方面进行了研究。

## 参考文献

- 1 Chess D. Security Issues in Mobile Code Systems, in: Giovanni Vigna (ED.), Mobile Agent Security, LNCS 1419, Springer, 1998. 1~14
- 2 Karnik N M, Tripathi A R. Design Issues in Mobile Agent Programming Systems. IEEE Concurrency, 1998, 6(3): 52~61
- 3 Wilhelm U G, Staamann S M, Buttyan L. A Pessimistic Approach to Trust in Mobile Agent Platforms. IEEE COMPUTING, SEPTEMBER/Oct. 2000. 40~48
- 4 Eckel B. Thinking in Java, 2nd Edition, Release 11. Prentice-Hall, 2000
- 5 Caro G D, Dorigo M. Two Ant Colony Algorithms for Best-effort Routing in Datagram Networks. In: Proc. of the 10<sup>th</sup> IASTED Intl. Conf. on Parallel
- 6 Schoonderwoerd R, Holland O, Bruten J. Ant-like Agents for Load Balancing in Telecommunications Networks. In: Proc. of the First Intl. Conf. on Autonomous Agents, ACM Press, 1997. 209~216
- 7 Schaaf M, Maure F. Integrating Java and CORBA: A Programmer's Perspective. IEEE Internet Computing, Jan.-Feb. 2001. 72~78
- 8 Farmer W M, Gutman J D, Swarup V. Security for Mobile Agents: Authentication and State Appraisal. In: The 4th European Symp On Research in Computer Security. Rome, Italy, 1996. 18~130
- 9 Eckel B. Thinking in Java, 2nd Edition, Release 11. New Jersey: Prentice-Hall, 2000
- 10 Davies W, Price W L. Security for Computer Networks. Chichester: John Wiley & Sons, 1989
- 11 Pfleeger C P. Security in Computing, Second Edition. Upper Saddle River, New Jersey: Prentice Hall, 1997
- 12 Wooldridge M. Intelligent Agents: Theory and Practice, Knowledge Engineering Review, 1995, 10(2)
- 13 Muller J P. The Design of Intelligent Agents, A Layered Approach, 1996
- 14 Dunham M H, Helal A. Mobile computing and databases: anything new? ACM SIGMOD Record, 1995, 24 (4)
- 15 Villinger K, Burger C. Generic mobile agents for electronic markets. In: The 4th Conf. of the Internet Society for Decision Support Systems, Switzerland, 1997
- 16 OMG, The common object request broker, architecture and specification. July 1995
- 17 Kraus S, Sycara K, Evenchik A. Reaching agreement through argumentation: A logical model and implementation. Artificial Intelligence, 1998, 104(1-2): 1~69
- 18 何炎祥,陈莘萌. Agent 和多 Agent 系统的设计与应用. 武汉:武汉大学出版社, 2001
- 19 张云勇. 移动 Agent 及其应用. 北京:清华大学出版社, 2002

(上接第171页)

- And Adaptation, COMPSAC 2003, Dallas, Nov. 2003
- 6 Laddaga R. Creating robust software through self-adaptation. IEEE Intelligent Systems, 14, May/June 1999. 26~29
  - 7 Laddaga R, et al. Introduction to Self-adaptive Software: Applications. Lecture Notes in Computer Science, 2001, 1936: 1~5
  - 8 Kephart J O, Chess D M. The Vision of Autonomic Computing. IEEE Computer, 2003, 36(1): 41~50
  - 9 Laddaga R, et al. Results of The First International Workshop on Self Adaptive Software, Lecture Notes in Computer Science, 2001, 1936: 242~247
  - 10 Kokar M M, Baclawski K, Eracar Y A. Control theory based foundations of self controlling software. IEEE Intelligent Systems, 1999, 14(3): 37~45
  - 11 Diao Y, Hellerstein J L, Parekh S, Bigus J P. Managing Web

Server Performance with AutoTune Agents. IBM Systems Journal, 2003, 42(1). URL. www.research.ibm.com/journal/sj/421/diao.pdf

- 12 Oreizy P, et al. An architecture-based approach to self-adaptive software. IEEE INTELLIGENTSYSTEMS, MAY/JUNE 1999
- 13 Garlan D, Schmerl B. Model-based Adaptation for Self-Healing Systems. WOSS'02, 2002
- 14 Ledoux T. OpenCorba: A Reflective Open Broker. In: Cointe P, ed. the 2nd Intl. Conf. on Reflection, LNCS 1616. Heidelberg: Springer-Verlag, 1999. 197~214
- 15 Klefstad R, Schmidt D C, O'Ryan C. Towards Highly Configurable Real-time Object Request Brokers. In: Fifth IEEE Intl. Symposium on Object-Oriented Real-time Distributed Computing. Washington, D. C., USA., 2002. 347~447