

# 基于软件代理的学习资源检索技术研究与应用

吴 砥 严 鹤 蔡 蔚

(华中科技大学电子与信息工程系智能互联网技术湖北省重点实验室 武汉 430074)

**摘要** 学习资源检索是网络教育资源服务的重要内容,传统搜索门户式的资源检索方式在资源的个性化服务能力、自主检索能力等方面有所欠缺。软件代理技术为实现学习资源的分布式检索、个性化服务等功能上提供了便利的途径,使实现学习资源的自主服务成为可能。利用软件代理的自主性、协作性和移动代理的迁移性特点,提出了一种基于软件代理的学习资源检索策略,研究了采用代理通信语言实现代理间通信,将检索操作指令嵌入在通信原语中,实现多代理通信与代理迁移机制相配合的分布式网上学习资源检索机制,并在此基础上实现了一个应用原型系统。

**关键词** 网络教育,资源检索,软件代理

中图分类号 TP319 文献标识码 A

## Research and Application on Agent-based Learning Resource Discovery Technology

WU Di YAN He CAI Wei

(Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

**Abstract** Learning resource discovery is a very important service in E-Learning. Traditional resource searching service cannot provide an individualized discovery service. Software agent technology can help us to implement a distributed learning resource discovery and a individualized service. It makes it possible to search for learning resource independently. This paper introduced an agent-based learning resource discovery strategy, and implemented an experimental system. The strategy applied the agent-agent communication language and contained query instruction in the primitive performatives. Based on the strategy, we implemented a learning resource discovery system based on the strategy.

**Keywords** E-learning, Resource discovery, Software agent

### 1 背景简介

网络教育融合现代网络技术、多媒体技术、信息技术的优势,革新了传统教育模式,为人们提供了一个不受时空限制的全新教育环境,使实现大范围的教育资源共享成为可能。学习资源服务是网络教育领域的重要内容,其质量的高低直接影响了网络教育的实际实施效果。学习资源检索是网络教育中学习资源服务的重要内容,高效的学习资源检索技术是提高网络教育的学习资源服务质量的关键要素。

软件代理(Agent)技术的诞生和发展是人工智能技术(Artificial Intelligence, AI)和计算机网络技术发展的必然结果<sup>[1]</sup>。人工智能技术的发展加快了软件智能化的进程,越来越多的软件具备了面向目标、自主决策和自主行为的能力;另一方面,计算机网络技术的发展使网络成为软件的主要运行载体,越来越多的软件不仅仅在一台单独计算机环境上运行,而且在多台计算机组成的网络环境上进行分布式运行。软件的智能化和网络化发展的结果,导致了软件代理技术的迅速发展,成为一种新的在分布式环境下解决复杂问题的技术途径,为智能分布式应用系统的建立提供了直接支持。

目前 Agent 的定义尚未得到统一,但公认的 Agent 应该具备的几个特点是明确的,即<sup>[2]</sup>1)自治性(自主性),Agent 可以在完成大部分任务时具有自主行为能力,不需要人或者其他 Agent 的指导;2)协作性(社会性),Agent 可以在需要其他 Agent 或者人的帮助的时候与他们进行交互,也可以在必要的时候为其他 Agent 或者人提供帮助;3)响应性(反应性),Agent 可以感知外界环境的变化,并及时作出反应,对自身进行相应的改变以适应外界的变化;4)前瞻性(预知性),Agent 对周围环境有一定的预知能力,在合适的时候能够主动采取一些有目的的行为,即具有一定的目标驱动特征。

移动代理(Mobile Agent, MA)是软件代理中的一种,它除了具有 Agent 的上述基本特点外,还具备一个其他 Agent 不具备的特点,即移动性。这里的移动性指的是主动的移动,而不是被动的移动。很多分布式软件系统也可以实现代码的迁移,但那是在预设规则之下的被动迁移,而 MA 则能够根据自身意愿进行自主迁移。MA 技术被认为是一个综合性的技术,它使分布式系统的设计、实现和维护都简单许多<sup>[3]</sup>。基于 Agent 建立 Web 学习资源检索系统,可以有效地解决用户个性化服务、资源分布式检索等问题,实现较高的学习资源服

到稿日期:2008-05-13 本文受教育部 2006 年“高等学校科技创新工程重大项目培育资金”项目《国家知识服务体系支撑技术研究》课题《基于移动代理的标准化网络教育知识资源检索技术研究》专题(专题编号:70538-03)资助。

吴 砥(1978-),博士,讲师,研究方向为教育信息化技术标准、对等网学习资源分发、数字教育资源管理与知识服务, E-mail: wudi@hust.edu.cn.

务质量,从而提高网络教育资源服务的整体水平。

## 2 基于软件代理的检索机制

软件代理的自主性、协作性和移动代理的迁移性,可以为网络教育中学习资源的检索提供良好的支持。传统的学习资源检索方式是基于关键词的门户式检索,用户必须首先访问搜索门户网站,输入自己所要检索资源的若干关键词,然后由数据库系统据此对数据库中存储的资源数据进行检索。检索门户一般采用快照方式存储网络资源的 Web 镜像,这种方式的主要弱点是:

1)使用不便,用户必须首先访问搜索网站,然后才能开展检索。如果网站无法访问,或者用户网络状况不好,暂时无法连接网络,都无法完成操作;

2)检索结果不准确,由于用户仅仅能够输入有限的关键词,无法准确地描述自己所需,所以得到的检索结果往往并不是用户想要的资源<sup>[4]</sup>;

3)检索结果排序不准确,传统搜索系统根据关键词匹配程度对检索结果进行排序,但是这往往并不是用户希望看到的排序次序;

4)检索结果本身真实度不高,由于搜索网站一般采用有一定时间间隔的快照方式存储内容,所以往往在某时刻检索到一个资源,而实际上该资源已经不存在。

为解决上述问题,可以采用基于 Agent 的资源检索策略,利用 Agent 的自主性、协作性和 Mobile Agent 的迁移性,构建一个基于 Agent 的 Web 学习资源检索平台,实现一种自主迁移、多点通信协作的 Web 学习资源检索策略,如图 1<sup>[5]</sup>所示。

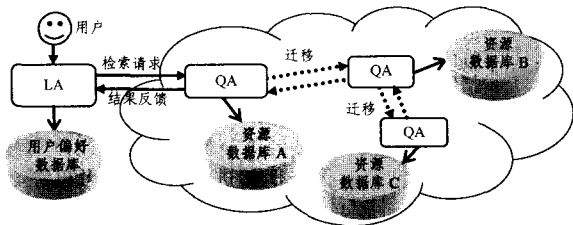


图 1 基于 Agent 的学习资源检索平台

图 1 是一个典型的多代理系统 (Multi-Agent System, MAS)。学习者代理 (Learner Agent, LA) 是驻留在用户机的 Agent,负责维护用户偏好数据库 (User Profile Database, UPD);检索代理 (Query Agent, QA) 是执行资源任务检索的 Mobile Agent,负责在不同资源库系统之间迁移,检索需要的资源,并返回结果。该检索平台的一般运转流程为:

- 1)用户向 LA 发送检索请求;
- 2)LA 根据用户请求和用户偏好,确定检索信息;
- 3)LA 创建一个 QA,并告知检索信息;
- 4)QA 访问资源数据库 A,检索是否有需要的资源;
- 5)如 QA 没有找到所需资源,则迁移至资源数据库 B。

可以看到,平台运转流程中涉及到大量的 Agent 之间的通信。有多种代理通信语言 (Agent Communication Language, ACL) 可以用来处理 Agent 之间的通信,其中最具有代表性的是知识查询操作语言 (Knowledge Query and Manipulation Language, KQML)。采用 KQML 定义的在 LA 和 QA

之间交互的资源检索请求和资源检索结果反馈消息如表 1<sup>[5]</sup>所列。

表 1 KQML 定义的检索请求和响应消息

请求消息	响应消息
request(	response(
:sender LA	:sender QA
:receiver QA	:receiver LA
:language Prolog	:language Prolog
:ontology KQML-Agents	:ontology KQML-Agents
:content REQUEST...	:content RESPONSE...
)	)

KQML 提供了多种消息原语 (performatives),以表示与内容无关的信息交换。无论采用什么样的具体通信方式,KQML 假定在 Agent 这个层次上,呈现的是一种点对点的消息通信方式,这种点对点消息传输方式屏蔽了底层通信机制,使代理间通信呈现出消息语义层面的单纯性。这种点对点消息传输抽象为外层基于 Agent 的程序通信提供一个简单、统一的模型,使基于 Agent 的程序和 API 更容易设计和实现<sup>[6]</sup>。

可以看出,使用 KQML 所定义的树状结构的语法规则和原语组织形式,非常适合于采用 XML 技术来绑定,以用于 Agent 之间的信息交换和传输。当然,必须注意的是 KQML 中有一些特殊字符 (如“#”、“\$”、“MYM”、“>”、“<”等)在 XML 中必须用编码形式表示出来<sup>[7]</sup>。采用 SOAP 作为传输协议,屏蔽底层的传输机制,采用 XML 作为格式绑定的载体,可以给出 KQML 的 XML 绑定模型,如图 2 所示。这样的绑定方式使 KQML 原语在 Agent 交互过程中可以借助 XML 解析技术得到有效处理。

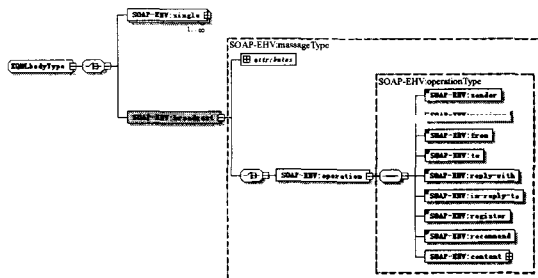


图 2 KQML 的 XML 绑定

基于 Agent 学习资源检索流程中涉及到多次 Agent 的状态迁移过程,以学习者代理 (Learner Agent, LA) 和检索代理 (Query Agent, QA) 为主要 Agent,实施资源的分布式检索,满足用户的个性化检索需求。参考曹军海等人的形式化描述方法<sup>[8]</sup>,给出基于 Mobile Agent 的资源检索流程的形式化描述如下<sup>[9]</sup>:

**定义 1** 序列 (Sequence) 是一个集合中所有元素的一个有序排列,可以表示为合式公式的形式。设  $A$  是一个集合, $p$  和  $q$  是  $A$  的两个元素,即  $p \in A, q \in A$ ,定义“ $p \rightarrow q$ ”表示“先  $p$  后  $q$ ”,“ $p \vee q$ ”表示“ $p$  与  $q$  并列”,“ $p \wedge q$ ”表示“ $p$  或  $q$ ”。

**定义 2** 动作 (Action) 是 Agent 内部造成 Agent 本身或者其环境的状态改变的最小执行单元。

**定义 3** 活动 (Activity) 是由事件激发或者由 Agent 主动发起的由多个动作组成的一个序列。

参考王红<sup>[10]</sup>的对 Agent 生命周期的描述,可以给出 QA 的生命周期和状态转移状况,如图 3 所示。

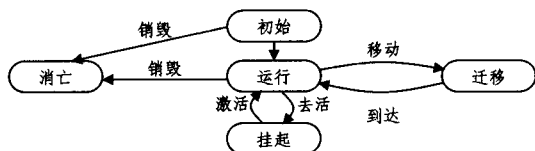


图 3 QA 的生命周期与状态转移图

根据前述检索流程,QA 的主要状态、动作和活动列表分别如表 2、表 3、表 4 所列。

表 2 QA 的主要状态

状态	涵义	状态	涵义	状态	涵义
S <sub>1</sub>	初始	S <sub>2</sub>	运行	S <sub>3</sub>	挂起
S <sub>4</sub>	迁移	S <sub>6</sub>	消亡		

表 3 QA 的主要动作

动作	涵义	动作	涵义
Act <sub>1</sub>	定位目的节点	Act <sub>7</sub>	比对权利证书
Act <sub>2</sub>	确认目的节点	Act <sub>8</sub>	保存学习资源
Act <sub>3</sub>	迁移目的节点	Act <sub>9</sub>	传送学习资源
Act <sub>4</sub>	确认用户身份	Act <sub>10</sub>	记录失败信息
Act <sub>5</sub>	检索所需资源	Act <sub>11</sub>	记录轨迹信息
Act <sub>6</sub>	检索权利证书		

表 4 QA 的主要活动

活动	涵义	活动	涵义
Atv <sub>1</sub>	节点迁移	Atv <sub>4</sub>	资源传送
Atv <sub>2</sub>	资源检索	Atv <sub>5</sub>	失败处理
Atv <sub>3</sub>	证书验证		

设 Acts 为所有动作的集合,即

$$Acts = \{ Act_i \mid i \leq 10, i \in n \}$$

设 Atvs 为所有活动的集合,即

$$Atvs = \{ Atv_1, Atv_2, Atv_3, Atv_4 \}$$

则

$$Atv_1 = Act_1 \rightarrow (Atv_5 \wedge (Act_2 \rightarrow Act_3 \rightarrow Atv_2))$$

$$Atv_2 = Act_4 \rightarrow ((Act_5 \rightarrow (Atv_3 \wedge Atv_5)) \wedge Atv_5)$$

$$Atv_3 = Act_6 \rightarrow ((Act_7 \rightarrow (Atv_4 \wedge Atv_5)) \wedge Atv_5)$$

$$Atv_4 = Act_8 \rightarrow Act_9$$

$$Atv_5 = Act_{10} \rightarrow Act_{11} \rightarrow Act_1$$

以下用 Petri 网给出资源检索流程的形式化描述:

设  $P = \{ p_i \mid i \leq 12, i \in n \}$  为 QA 的有限状态集,  $Acts = \{ Act_i \mid i \leq 10, i \in n \}$  为 QA 的有限动作集,则  $Atv_1, Atv_2$  可分别用 Petri 网表示。如图 4、图 5 所示,类似可以给出  $Atv_3, Atv_4, Atv_5$  的 Petri 网表示。

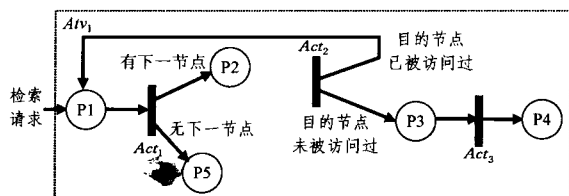


图 4 Atv<sub>1</sub> 的 Petri 网表示

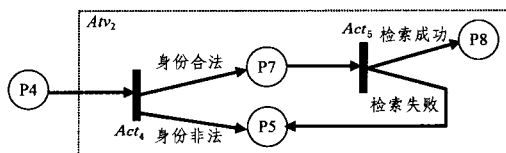


图 5 Atv<sub>2</sub> 的 Petri 网表示

由前述可见,在 Web 学习资源检索流程的各个环节中,Agent 的迁移条件是一致的,即 1) 在现有节点 Agent 不能完成所需的操作,可能的原因是没有检索到所需资源,或者虽然检索到所需资源但相应的数字权利证书不能匹配;2) Agent 还有下一目标节点没有迁移过,即在可知的节点中,尚有 Agent 没有迁移过的目标节点。Agent 可以通过自身携带的数据、程序和状态信息判断在当前节点能否完成所需操作,但是无法仅依靠自身力量得知是否还有下一目标节点没有迁移过,更无法进行权衡完成迁移选路操作,因此有必要在 MAS 中设置一个控制代理,即 Facilitator。Facilitator 保存并随时更新一个记录 MAS 中所有 Agent 状态和位置信息的映射表,每次 Agent 的迁移操作都根据该映射表进行选路,这就避免了 Agent 迁移中可能存在的盲目性问题。

### 3 基于软件代理的检索系统

Aglet 是 IBM 公司的移动代理开发解决方案,采用纯 Java 技术实现,为用户提供了一个简单、全面、实用的 MA 开发平台和执行环境。Aglet 主要基于 J2SE 开发,不仅提供了移动代理的实现机制,还提供了一个方便的运行环境。Aglet 的体系结构如图 6<sup>[4]</sup>所示。

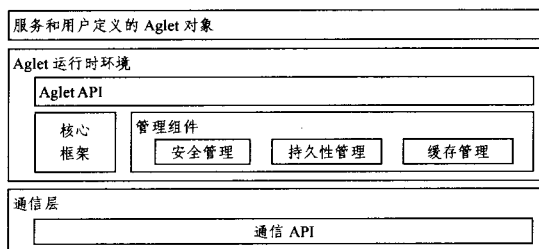


图 6 Aglet 体系结构

Java 技术曾经依靠 Applet 在网络上的良好移动性而获得极高的声誉。与 Applet 一样,Aglet 也具有有良好的移动性,Aglet 对象可以在网络上灵活移动。与 Applet 不同的是,Aglet 迁移的时候还迁移了状态信息。一个 Applet 是一个可以通过网络从服务器迁移到客户机的代码,而一个 Aglet 是一个运行中的 Java 程序(包括代码和状态),通过网络在不同主机之间的整体迁移<sup>[11]</sup>。另一方面,Java 天然所具有的跨平台特性,使 Aglet 代码的迁移有了更多的保障,使用户不必为平台的非一致性而担心。

为验证基于软件代理的学习资源检索机制的可行性和有效性,我们基于 Aglet 开发了学习资源服务管理原型系统。系统包括两个部分:一个是基于 Aglet 的资源检索模块,另一个是基于 Web 的用户交互模块。主要思路是基于 Aglet 实现分布式检索,基于 Web 实现用户交互。

基于 Aglet,结合 J2EE 技术实现的学习资源检索系统如图 7 所示。该系统具备了将一个用户检索请求编码为 XML

格式信息,并在后台进行自主迁移和检索的能力。用户检索请求信息被内嵌在 KQML 的内容层中,由 Aglet 携带进行数据通信和代码迁移,学习资源的分布式检索流程通过 Aglet 对象的迁移而实现。

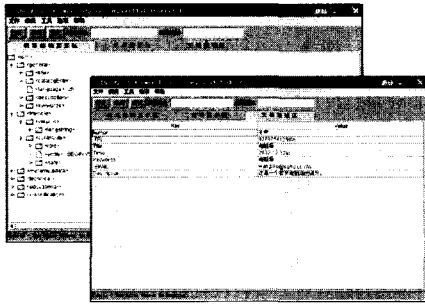


图7 基于 Aglet 实现的资源检索系统

实际运行中,当一个用户检索请求到达时,一个 Aglet 实例由后台 Servlet 创建,并自主迁移到网络中不同的数据库系统,检索用户所需的资源,返回结果到页面。对用户来说,并不知道自己已经访问了不在同一台服务器上的多个数据库,所有的迁移操作都是在后台完成的。

为验证该检索策略是否对于提升系统的运行性能具有实际作用,作者在采用 Aglet 机制的分布式检索和不采用 Aglet,仅仅通过 SOAP 消息通信实现同样功能的学习资源检索模块之间进行性能比对实验。在同一个 100 兆以太网局域网内,采用 PC 机模拟两台数据库服务器,先后采用 SOAP RPC 消息通信方式和 Aglet 方式,实现对同一条学习资源数据的分布式检索。首先检索第一台数据库服务器,如没有检索到想要的学习资源,则转而检索第二台数据库服务器(实验实际资源样本设计为第一台数据库服务器中检索不到,而第二台数据库服务器中可以检索到的学习资源,以便确保检索过程形成二次检索)。实验硬软件配置情况如表 5、表 6 所列。

表 5 实验硬件配置

服务器 1		服务器 2	
CPU	P4 2.40G	CPU	P4 2.40G
IP	192.168.1.144	IP	192.168.1.145
内存	512M	内存	1G

表 6 实验软件配置

Web 服务器	Tomcat 5.0.16
数据库管理系统	MySQL 3.21
SOAP 开发平台	SAAJ 1.1.02, JAXMI. 1.01
Agent 开发平台	IBM Aglet2.0.2

在采用软件代理的实现方案中,系统首先由与用户直接交互的 JSP 页面调用后台 Servlet,由 Servlet 在后台创建一个 Aglet 实例,检索第一个资源数据库,如没有,则 Aglet 自主迁移到第二台数据库服务器进行检索,直到找到想要的学习资源,然后携带检索结果提示信息返回,并将检索结果回显到 JSP 页面。

在采用 SOAP 消息通信的实现方案中,两台数据库服务器都将自身的资源检索服务对外发布为 Web 服务,供 SOAP 消息远程访问调用。检索过程中,系统首先由 JSP 页面调用后台 Servlet,向第一台数据库服务器发送检索消息,调用 Web 服务进行检索,如检索不到,则向第二台数据库服务器

发送检索消息,调用 Web 服务进行检索,并将检索结果回显到 JSP 页面。

检索案例:以 10000 条不同的学习资源数据作为检索对象,以检索包含特定词条的数据进行测试。分别进行了 10 次测试,最后将测试数据取平均值。实验过程中发现,无论是利用 Aglet 还是利用 SOAP,第一次检索的耗时都相对较多。主要原因是 Java Servlet 程序发布后第一次运行时要先进行编译,因此耗时较多,所以在取样时不作为数据样本列入。

实验主要依据在同等硬软件条件和网络环境下的耗时进行对比。Aglet 迁移总时间由 Aglet 从本机开始游走时计算,到 Aglet 返回结果集到本机后为止,不包括本机在 Servlet 中通过 Socket 创建和驱动 Aglet,以及从本机通过 Socket 返回结果集到 Servlet 的时间。

SOAP 通信时间包括与服务器 SOAP 和与服务器 1 通信的总时间,从通过 SOAP 发送要检索的信息开始到 SOAP 接收到返回的结果集为止,SOAP 与服务器 2 通信的总时间从通过 SOAP 发送要检索的信息开始到 SOAP 接收到返回的结果集为止。

由实验结果得到,利用 Aglet 检索两个数据库的时间平均值为 0.9408 秒,而利用 SOAP 检索两个数据库的平均时间为 1.0606 秒,如图 8 所示。可以看到 Aglet 因为迁移次数少,不需要频繁的消息交互操作,所以在效率上优于 RPC 方式的纯 SOAP 消息交互处理机制。

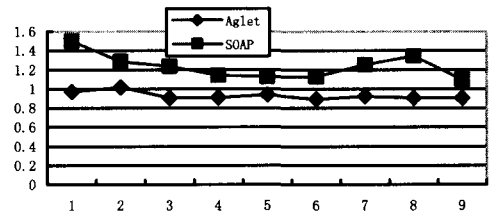


图8 两种检索方式的对比

结束语 软件代理技术对于提高学习资源检索效果,提高网络教育的学习资源服务质量具有重要作用。本文讨论了软件代理技术在网络教育学习资源检索中的作用和实际应用策略,并基于 J2EE 技术和 IBM 公司的 Aglet 平台实现了一个原型系统,验证了此种检索机制的有效性。

随着国内网络教育资源开发量的急剧增加,资源积累越来越丰富,面对众多的教育资源,如何提高学习资源的检索有效性,是我们越将将要面临的问题。利用移动代理技术自主性、迁移性的特征,可以建立一个有效的学习资源检索机制,提高学习资源的服务质量。

## 参考文献

- [1] 张云勇,刘锦德. 移动 agent 技术[M]. 北京:清华大学出版社, 2003:2
- [2] Jennings N, Wooldridge M. Software agents. IEE Review, January 1996:17-20
- [3] Satoh I. Selection of Mobile Agents // Proceedings of the 24th International Conference of Distributed Computing System (ICDCS'04), March 2004:484-493
- [4] Quah J T S, Leow W C H, Chen Y M. Mobile Agent Assisted E-Learning // the First International Conference on Information

Technology & Applications. ICITA2002, March 2002;595-600

[5] Wu Di, Yang Zongkai, Cheng Wenqing. A Mobile Agent Assisted Learning Resource Service Framework Based on SOAP. Web-based Learning: Technology and Pedagogy, July 31-August 3, 2005;125-134

[6] Labrou Y, Finin T. A Proposal for a New KQML Specification. February 1997;3

[7] 冯志勇,洪卫林. 基于 SOAP 协议的 KQML 语言通信实现[J]. 计算机工程, 2003, 29(6): 97-98

[8] 曹军海,张和明,熊光楞. 多 Agent 仿真中 Agent 行为的形式化描述方法[J]. 系统仿真学报, 2004, 16(11): 2399

[9] 吴砥. 学习资源管理与服务关键技术研究[D]. 武汉: 华中科技大学, 2006; 100

[10] 王红. 移动 agent 关键技术研究[D]. 北京: 中科院计算所, 2002; 15

[11] Venners B. The Architecture of Aglets. <http://www.java-world.com/javaworld/jw-04-1997/jw-04-hood.html>, April 1997

(上接第 174 页)

数据文件的大小为 500M, 数据网格系统中共有 1000 个均匀分布在不同网格节点上的数据源文件。

基于上述模拟环境, 分析副本数  $R$  的选取, 以及不同在线概率对一致性维护性能的影响, 并比较分析了读写请求量、请求执行时间等因素对本文提出的基于自适应冲突协调机制的一致性算法和 Andrea Domenici<sup>[4]</sup> 提出的基于分布锁的数据网格副本一致性方案的性能影响。

#### 实验 1: 副本数 $R$ 对读一致性的影响

本文算法在至少有一个有效节点在线的情况下满足读一致性, 而在网格环境下满足该条件的概率为  $P_{Read} \geq 1 - (1 - P_{OnlineCorrect})^R$ 。

因此, 通过改变同一逻辑副本的物理副本数  $R$  分析在不同在线概率  $p_{Online}$  下对读一致性的影响。可以为  $R$  选取一个合适的值, 使得在低在线概率情况下, 可以满足读一致性。

从图 2 可以看出, 随着  $R$  的增大, 对在线概率的要求越低。当  $R=60$  时, 在线概率为 20% 即可使读一致性概率达到 90%。考虑到副本越多开销越大, 因此, 可选取  $R$  为 60。

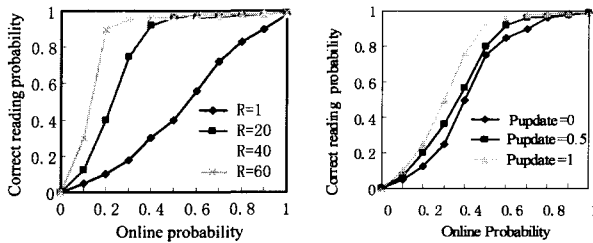


图 2 副本数  $R$  对读一致性的影响 图 3  $P_{update}$  对读一致性的影响分析

#### 实验 2: $P_{update}$ 对读一致性的影响

当  $R=10$  时, 副本节点在离线与在线状态之间转换的过程中发生更新的概率  $P_{update}$  也是对读一致性影响的重要因素。

从图 3 可以看出, 副本节点在离线与在线状态之间转换过程中发生更新比不发生更新的读一致性概率提高了 20%。因此, 写概率越高, 网格系统的读一致性概率就越高。

#### 实验 3: 分布锁算法的读写请求量与请求执行时间关系的比较

在  $R=60$ , 低在线概率  $P_{online}=0.2$  的条件下, 本文算法与 Andrea Domenici<sup>[4]</sup> 提出的基于分布锁的副本一致性算法进行比较分析。

从图 4 可以看出, 本文算法的执行请求的总时间明显低于基于分布锁的数据网格副本一致性算法。因此, 本文算法对于读写请求量具有更好的可扩展性, 性能较优。

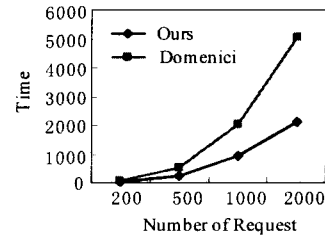


图 4 本文算法与 Domenici 分布锁算法的比较

**结束语** 针对网格系统高度动态异构的特点, 基于动态投票机制, 提出了副本一致性维护算法, 通过对全局有序性和读一致性的分析, 证明了算法的正确性。经仿真实验表明: 在实验确定的低在线概率满足读一致性副本数  $R$  下, 与 Andrea Domenici 所提出的基于分布锁机制的副本一致性算法相比, 具有更好的可扩展性。

下一步研究工作将研究与容错相合的副本一致性维护算法, 考虑失效副本的一致性恢复。

#### 参考文献

[1] Düllmann D, Hoschek W, Jaen-Martinez J, et al. Models for replica synchronization and consistency in a data grid[C]// Tenth IEEE Symposium on High Performance and Distributed Computing (HPDC-10). San Francisco, CA, August 2001

[2] Proti'c J, Tomasevi'c M, Milutinovi'c V. Distributed Shared Memory, Concepts and Systems[C]// IEEE Computer Society Press: Los Alamitos, CA, 1997

[3] Levy E, Korth H F, Silberschatz A. An optimistic commit protocol for distributed transaction management// James C, Roger K, eds. Proceedings of the ACM SIGMOD International Conference on Management of Data. New York, ACM Press, 1991; 88-97

[4] Jim\_enez-Peris R, Pati\_no-Mart\_enez M, Alonso G, et al. Are quorums an alternative for data replication? [J]. ACM Trans. Database System, 2003, 28; 257-294

[5] Adzilacos V H, Toueg S. Fault-tolerant broadcasts and related problems// S. Mullender, ed. Distributed Systems, second edition. New York, ACM Press, 1993; 97-145

[6] Schiper A, Sandoz A. Uniform reliable multicast in a virtually synchronous environment// Proceedings of the 13th International Conference on Distributed Computing Systems (ICDCS-13). Pittsburgh, Pennsylvania, USA, IEEE Computer Society Press, May 1993; 561-568

[7] Domenici A, Donno F, Pucciani G. Replica consistency in a Data Grid[J]. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 2004, 534(1/2); 24-28

[8] Herlihy M, Wing J. Linearizability: A correctness condition for concurrent objects[J]. ACM Trans Programming Languages and Systems, 1990, 12(3); 463-492