迁移工作流中MI的迁移策略研究

吴修国
(山东财经大学管理科学与工程学院 济南 250014)

摘 要 迁移工作流是将移动agent计算模式应用于工作流管理的一项新技术。迁移实例(MI)、工作位置和迁移工作流管理引擎构成迁移工作流管理系统的三要素。其中,MI是以移动agent为计算范型构造的业务过程执行agent,它可以在工作位置之间移动并按照自身携带的工作流说明,就地利用服务执行一项或多项任务。MI的迁移策略是整个迁移工作流系统的核心问题之一。针对MI的服务定位问题,提出了一个分层结构的迁移工作流服务模式,在它对服务进行管理的同时也负责为MI导航;针对迁移目的地的选择问题在充分考虑了主机硬件可用度、主机资源以及迁移实例目标与主机提供服务的匹配程度的基础上,提出了包含静态和动态等要素在内的目的地主机可用度评估方法,用以确定MI的迁移目的地。最后,给出了实验过程描述和对结果的讨论分析。实验表明,分层结构的工作流服务组织形式以及目的地主机的可用评估方法,有效提高了系统效率。

关键词 迁移工作流,迁移策略,MI
中图法分类号 TP391 **文献标识码** A

Research on MI Migrating Strategy in Migrating Workflow

WU Xiu-guo
(School of Management Science and Engineering, Shandong University of Finance and Economics, Jinan 250014, China)

Abstract Migrating workflow is a new technology by which mobile agent is applied to workflow management. MI(Migrating Instance), work place and workflow engine are the main three elements which constitute a migrating workflow system. Among them, MI is based on mobile agent computing paradigm for the structure of the business process execution agent, which can move between work places and by following the their own workflow instructions, uses local service to implement one or more tasks. Migration strategy of MI is one of the most important issues in migrating workflow system. This paper presented a hierarchical structure of the migrating workflow service model aimed to the question for MI how to find services, also proposed a way for finding suit destination host including static and dynamic elements such as the availability of the host hardware, host resources. Finally, we gave a description of the experiment and the results of the discussion and analysis. Experiments show that hierarchical organization of workflow services, and the destination host assessment methods are available to effectively improve the efficiency of the migrating workflow system.

Keywords Migrating workflow, Migrating strategy, MI

1 引言

依照国际工作流联盟(WfMC)的定义,工作流是"业务流程的自动化和半自动化过程",在此过程中,文档、信息或者任务按照一定的过程规则流转,实现组织成员间的协调工作,以期达到业务的整体目标[1]。传统的工作流包括两个阶段,即建模阶段和执行阶段。建模阶段是对一个工作流进程进行定义;执行阶段则是工作流执行服务对工作流定义进行解释执行。由于一个工作流程在执行阶段会因为外部环境或者用户需求的变化而发生变化,因此工作流系统因缺乏动态性和实用性在实际的应用中具有一定的局限性。文献[2]参照移动计算范型给出了包括迁移工作流管理机、停靠站服务器、工作机网络以及迁移实例(MI)等的迁移工作流(Migrating Workflow, MWf)的概念定义,并提出了一个基于停靠站的迁移工作流管理系统结构。迁移工作流提高了工作流系统适应动态环境的灵活性,特别适合需要传递大量数据或需要大量调用远程服务的分布式业务并发处理过程。

在上述研究中,有关迁移实例的迁移策略,即迁移实例如何依据自身的任务、网络的软硬件环境和/或其他约束条件规划出最佳的迁移目的、迁移路线等问题,是迁移工作流系统的核心问题之一。迁移策略的优劣直接影响迁移实例任务的完成乃至迁移工作流系统的性能。因此,迁移实例的迁移策略受到学术界和工业界的广泛关注。

许多专家和学者沿用了移动agent的迁移策略。如文献[3]基于"旅行图"的概念,依据agent拥有的知识和以往的旅行经验可以自主地修改迁移路线的方法,通过感知模块和"资源监测agent"获取动态变化的环境,并在旅行图基础上减少MA在移动过程中的开销,给出了3种基于"旅行图"的移动

到稿日期:2011-07-29 返修日期:2011-10-17 本文受国家自然科学基金项目(60573169),山东经济学院博士基金项目(2010034)资助。
吴修国(1975—),男,博士,副教授,主要研究方向为移动计算、工作流等,E-mail:xiuguosd@163.com。

agent 迁移策略，充分体现了 agent 的反应性和自主性，有效地提高了移动 agent 的执行效率。该工作在移动 agent 迁移策略方面进行了有益的探索，提出很多行之有效的方法与策略，对提高移动 agent 系统的效率起到了非常重要的作用。但直接将其应用于迁移工作流系统中作为迁移实例的迁移策略，存在如下不足：(1)迁移实例迁移的目的是完成某项任务，所以不会出现强弱迁移的情况；(2)在多个主机都能实现任务的情况下随机选择目标主机，忽视了主机当前的忙闲等动态要素；(3)迁移实例可能携带一系列的任务，实现这些任务可能需要特定的顺序，这与移动 agent 仅实现某一个任务不同。针对上述问题，本文拟从两个方面解决迁移工作流中迁移实例的迁移策略问题：(1)针对 MI 的服务定位问题，提出了一个分层结构的迁移工作流服务模式，在对服务进行管理的同时也负责为 MI 导航；(2)针对迁移目的地的选择问题在充分考虑主机硬件可用度、主机资源以及迁移实例目标与主机提供服务的匹配程度的基础上，提出了包含静态和动态等要素在内的目的地主机可用度评估方法，用以确定 MI 的迁移目的地[5]。

2 迁移工作流基础

迁移工作流系统框架可示意为图 1，它由一个迁移工作流管理机和若干个已经建立友好信任关系的局域网互联组成。

定义 1 迁移工作流 MWF 是一个四元组(W_{id}，MI，WP，Engine)。其中，W_{id} 为迁移工作流标识；MI＝{mi_1，mi_2，…，mi_n}是迁移实例的集合，每个迁移实例 $mi\in$MI 执行一个目标相对独立的业务过程 BP；WP＝{wp_1，wp_2，…，wp_m}是所有迁移实例可能的工作位置集合；Engine＝{Gmi，Gwp}是面向业务流程目标的工作流引擎，其中 Gm_i 定义在迁移实例集合 MI 和工作位置 WP 上，它不仅掌握所有工作位置的资源状况与服务能力，能够依据业务流程的总目标和子目标，规划何时、何地与如何创建或派生迁移实例，而且掌握所有 $mi\in$MI 的工作需求、外部特征和当前状态，能够实施对迁移实例工作的协调和管理；Gw_p 定义在集合 WP 上，负责工作位置的组织和管理，并协调多个迁移实例以抢占同一个工作位置时的冲突。

定义 2 迁移实例 $mi\in$MI 是一个八元组(mi_{id}，TL，t，MP，p，S，ToL，MC)。其中，mi_{id} 为可认证的迁移实例标识；TL＝({⟨t_1，R_1，S_1⟩，⟨t_2，R_2，S_2⟩，…，⟨t_n，R_n，S_n⟩}，Schedule)是迁移实例携带的任务说明书，包括任务列表{⟨t_1，R_1，S_1⟩，⟨t_2，R_2，S_2⟩，…，⟨t_n，R_n，S_n⟩}和任务调度 Schedule 两部分。其中，任务 t_i 对应 BP 中的活动 a_i，i＝1，2，…，n；R_i 是任务 t_i 的资源需求；S_i 是任务 t_i 的服务需求，Schedule 是依照目标预先定义的任务执行关系 AR；t 为迁移实例 mi 当前正在处理的任务，$t\in$TL；MP 为允许 mi 迁移的工作位置集合，MP$\subseteq$WP；p 为迁移实例 mi 当前所处的工作位置，$p\in$MP；S 为迁移实例 mi 的当前状态；ToL 为迁移实例 mi 的生命周期；MC 为迁移实例 mi 的工作机，包括任务执行与中止、多任务协调、当前工作状态捕获、当前位置上资源与服务的可满足性检测、迁移查询、决策与迁移、自身安全保护等。

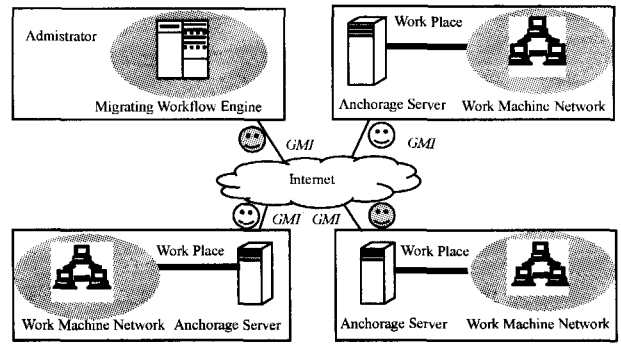

图 1 迁移工作流系统结构图

3 迁移实例的服务定位

按照迁移实例就地利用工作位置服务执行任务的工作流管理模式，迁移实例需要知道哪里存放着可以利用的服务，哪里可以提供最佳的服务。迁移实例如何在工作流网络中迅速有效地找到其需要的服务，称为服务定位问题。这是其需首要解决的问题，否则迁移实例的迁移就没有意义。参照移动 agent 中较常用的方法，将与服务定位有关的代码置于迁移实例内部，在派遣出之后，在整个网络中独立地搜索目标服务。显然，这种方法会使迁移实例的体积增大，增加迁移时间，而且让所有的迁移实例携带功能相似的代码也是一种资源浪费，增加了网络负载。更重要的是，迁移实例和服务平台以及其他迁移实例之间缺乏足够的交互性，最终限制了服务定位的性能。本文提出的服务定位方法与移动 agent 独立搜索服务不同，将服务定位代码从迁移实例中分离出去，使其变为工作流管理引擎和工作流管理机的组成部分，由它们提供诸如服务注册/定位的支持，从而可为迁移实例导航[6]。

3.1 迁移工作流系统的服务管理

为避免概念混淆，我们预先定义一些术语，这些术语将在后面的叙述中用到。(1)运行环境(MI Environment，MIE)：迁移实例运行时所处的环境，通常由迁移工作流平台提供。(2)工作机(Work Machine，WM)：网络中的计算机，能为 MI 提供服务。(3)停靠站服务器(Anchorage Server，AS)：能注册服务并接受来自 MI 查询的主机。(4)服务空间(Service Space，SS)：由停靠站服务器管理的一组主机。通常在一个 SS 中仅包含一个 AS。(5)服务项(Service Entry，SE)：关于服务特征的信息。每个 SE 代表一个可用服务。(6)可用服务列表(Available Service List，ASL)：存储 SE 的数据表。

迁移实例的服务定位模型是由 AS，SS，WM 和 MIE 共同组成的分级结构，如图 2 所示。

在图 2 所示的模型中，每个停靠站服务器 AS 负责为当前服务空间内的工作机管理可用服务列表。例如，在服务空间 A 中，停靠站服务器 A 可被 3 台工作机(工作机 A，B，C)提供的服务注册。因而，停靠站服务器能在任何时候了解到关于这些服务的信息。

在该模型中，工作机能动态加入或者退出一个服务空间。假设一台新主机 N 企图加入服务空间 A，首先向停靠站服务

器 A(服务空间 A 的管理者)派遣一个携带服务项的注册请求 RR(步骤 1)。当 RR 抵达 A 后,它向 A 送出一个注册请求(步骤 2)。停靠站服务器 A 依据其负载情况和当前管理的主机数量决定是否接受此请求(步骤 3)。如果停靠站服务器 A 接受,它向工作机 N 送回一个 Accept 应答,并将该服务添加进 SL 中。这样一来 N 和它所提供的服务就成为了 SSJ 的成员。

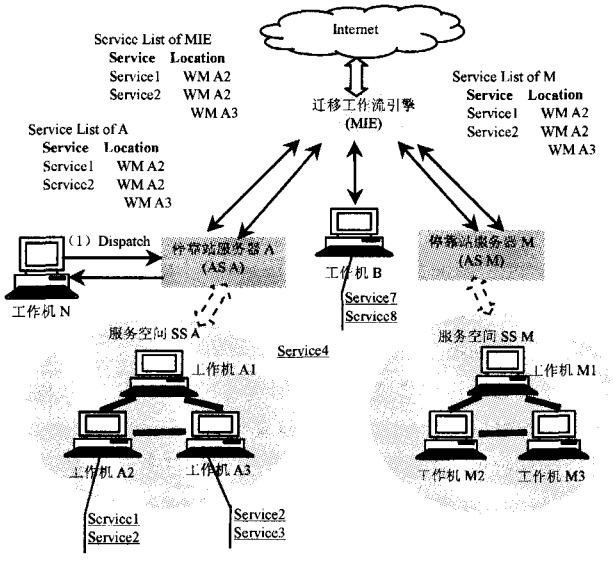

图 2 迁移示例服务定位模型

但也存在另一种情况,即由 N 派遣出的注册请求会被 AS A 拒绝。假如 AS A 的负载或是其管理的主机数量超出可承载的程度,AS A 会将该注册请求 RR 导向上级 AS(比如图 2 中的 MIE)。一旦 MIE 接收到这样一个转移而来的 RA,则把注册请求在当前的 SS 范围内广播(步骤 4)。如果有任何下级 RS 都返回 Accept 应答(步骤 5),A 将此应答连同该 RS 的地址信息一起送至 N。随后,N 和它所提供的服务可以向接受请求的目标 RS 注册。另一方面,若所有下级 RS 都拒绝该请求,A 会尝试自己接受该请求(步骤 6)。如果能成功的话,A 发送 Accept 应答给 N。若 A 也拒绝该请求,它将向 RA 导向上级 RS 去处理。

算法 1 服务注册算法

```
RegisterMachine(s,WM,SS)
{
  WM. dispatch(req,s);
  If(SS. Accept(req)) {
    SS. AppendToSL(s,req);
    SS. Reply(WM,ACCEPT,s);}
  Else {
    While(NotTimeOut(req) &&(Exist(SS. rs)) {
      SS. Forward(s,WM);
      SS. Parent. Broadcast(s,WM);
      WaitForAck(ACK);
      If(ACK==Accept){
        P_rs. Reply(WM,Accept,ACK);
        Goto Stop;}
      ElseIf(Map(s,p_rs. sl){
        P_rs. Reply(h,Accept,p_rs. addr);
        Goto Stop;}}
Stop:
      Kill(ra. reqs);}}
```

尽管与传统概念上的服务有区别,“服务注册”也可被看作是由特殊主机(AS)提供的一类特殊服务(S)。因此,每个主机都可向其上级进行服务注册。一旦某个主机不能正常提供某项服务,或者注册服务器发现某个主机提供的服务达不到预先设定的要求,就要更新该系统提供的服务内容。这种情况包括:(1)主机本身负载过重,不能及时提供该服务 s;(2)停靠站服务器经检测发现该主机提供的服务未达到预先设定的要求;(3)其它原因。这种情况下,基本的处理的方法包括:(1)找到该主机不能正常提供服务的原因,并进行修复;(2)直接将该主机提供的服务 s 从服务列表中删除。在实际应用中,方法(2)较为简单。

3.2 服务项的内容与标识

服务项的实质是集中描述一项服务,因此对服务定位模型的性能来说,服务项的内容描述详细与否直接决定搜索命中率的高低。尽管可以在服务项内包含尽可能多的服务特征,但那样会额外占用存储空间和搜索时间。因此,需要在存储搜索代价和定位性能之间找到折衷。一个服务项一般包含以下内容:

(1)服务名称 Name:设置名称只是为了方便引用,不需全局唯一。

(2)服务标识 ID:与名称不同,ID 是最重要的关于服务的信息,也是服务调用函数的句柄,必须全局唯一。

(3)服务目标描述 Goal:描述调用该服务后达到的预期目标。

(4)服务类型 Type:当前服务所属类型,方便分组与搜索。

(5)服务期望值 Goal:MI 调用服务后期望得到的结果,此域的值在不同情况下可能有不同的形式。

(6)服务质量 Quality:服务满足用户需要的程度,可以作为在一组相似服务中排序的标准。

(7)服务位置 Location:服务提供者的网络地址,可以是域名或者 IP。

对服务定位模型来说,一个关键问题是怎样标识网络上的众多服务,从而使得它们能被正确有效地定位。在此引入下述方法:

服务 ID 由两部分组成:共享域地址(主机地址+端口号)和本地名称(Local_ID)。

假设有工作机 h(202.93.73.9;Port:4389)提供一项服务 s,服务 s 的 ID 标识为 202093073009_4389_s。由于每台主机的共享域地址是唯一的,只要确保每个服务有一个本地唯一名称,因此这种方法就能保证该服务完整 ID 的全局唯一性。

3.3 服务定位

迁移实例服务定位的基本步骤可以描述为:MI 首先向停靠站服务器 AS 送出关于目标 g 的查询(que_g)。如果 que_g 内包含的 SE 与目标 g 吻合,则一个肯定应答连同 s 的位置信息

将被送回给 MI，否则查询会被转向上一级的工作流引擎。当上一级接收到这样一个转移过来的查询，则在当前的 AS 中进行广播。如果有任何 AS 回应一个肯定回答，则将该应答连同查询到的服务地址信息一并送回 MI。若在当前 AS 内没有回应，则该查询将转向上一级 AS 进行处理[6,7]。

3.4 相关研究

Travel Plan(TP)是整合的注册服务搜索模型，核心思想是顺序访问，每个域保存一个列表顺序，存储该域内主机的地址。由于采用的是盲目搜索，TP 的性能可从概率分析得出。假设迁移实例 MI 企图在规模为 n 台主机的网络中搜索服务 s，则该 MI 在第一次访问某台主机时就能找到服务 s 的概率为 $1/n$，这一数值随网络规模扩大而减小。在算法 1 中，搜索总是在一个较小的范围内(当前 SS)被执行，只有在当前 AS 返回拒绝应答时，搜索才在一个更广的范围内执行(上级 SS)。此外，与 TP 中的顺序访问所有主机不同，在我们的模型中，MI 可仅通过一次查询而明确得知当前 AS 内是否存在目标任务。假定有规模为 n 的网络，其中的主机和 RS 平均分布于 m 个 AS 内，则每个 AS 内有$(\frac{n-1}{m}-1)$个主机和一个 RS(假设存在一个顶级 RS 与所有下级 RS 相连)。MI 在首次查询就可发现目标服务的概率为$((\frac{n-1}{m}-1)\frac{1}{(n-1)-m})=\frac{1}{m}$，该值的大小仅取决于 SS 的数量。即使网络中加入了更多主机，只要 AS 的数量保持不变，该概率就不会减小。

4 迁移实例的目的地选择

在第 3 节提出的迁移实例目标服务定位算法是为 MI 的任务进行服务定位。在实际的应用环境中，服务定位的结果可能有不止一台目标主机可以满足 MI 的要求，提供其需要的服务。MI 面对的问题就是如何从这些潜在的目的主机中选择一个作为其迁移目的地。而做此选择的一个通用标准是选择拥有最高“可用度”的主机，使得 MI 尽可能获得最佳运行性能[8]。

比如，假设当前 MI 需要定位的任务集合 $t=\{t_1,t_2,\cdots,t_m\}$，任务 t_i 的服务定位结果为$\{H_{i1},H_{i2},\cdots,H_{im}\}$，则 MI 应该怎样评估这些主机的可用度来确定哪一台最适合运行，且可以作为其迁移目的地。

迁移实例的目的地选择问题，一方面涉及运行环境的各个方面，比如 I/O 负载、CPU 处理能力、内存数量、磁盘效率以及网络状况等硬件条件；另一方面还涉及迁移目的地能否尽可能多地提供目标服务，以满足目标集合中的多个目标，减少迁移实例迁移的次数。针对这一问题，一个常用的解决方案是随机选择主机 H_i，让 MI 迁移到该工作机上。这种方法的缺点是随机性太强，有时会导致迁移失败。另一个解决方案是利用主机特征参数作为标准进行可用度评估，但是到目前为止还没有比较行之有效的方法用来衡量每项特征参数在整体评估中占的比重。一般来讲，根据特征参数值和 MI 运行性能之间关系的经验可以为每种特征值分配一个系数，进而通过计算特征值的加权和得出一个综合数值，该值在某种程度上反映了主机运行环境的可用度。这种方法的缺点在于其主观性有时会引起误差，使得结果与实际情况相背离。在充分考虑主机硬件可用度、主机资源以及迁移实例目标与主机提供服务的匹配程度的基础上，提出了一种迁移实例目的地可用度评测模型：

$$V(H_i)=aS(H_i)+bD(H_i) \quad (1)$$

式中，$V(H_i)$表示主机 H_i 的评测度值；$S(H_i)$是主机 H_i 的计算机特征值，是一个静态值，它在某种程度上反映了主机 H_i 的运行环境的硬件配置；$D(H_i)$是对主机 H_i 运行环境的未来变化趋势进行的某种程度的预测，它是一个动态值，反映了当前主机 H_i 的资源耗用量，即忙/闲程度；a,b 分别表示两个特征值在 $V(H_i)$中所发挥的作用，称为权重。一般来说，要求两个值的和为 1。

4.1 $S(H_i)$

$S(H_i)$用于描述主机硬件可用度，它主要涉及运行环境的各个方面，比如 I/O 负载、CPU 处理能力、内存数量、磁盘效率以及网络状况等，这些都是能影响迁移实例运行性能(通常用运行时间衡量)的关键因素。

迁移工作流引擎 ME 在它创建的每种类别的 MI 内部嵌入一个神经网络函数和一个训练集文件。该函数的输入是主机特征参数值，输出是预测的 MI 在该特征环境下的运行时间，其目的是用来确定主机特征值和 MI 运行时间之间的关系。当 MI 离开出发地抵达目的地之后，先从目的主机采集一组特征参数，随后启动运行，并在完成运行后计算出所花费的时间。当它返回出发地后，采集到的信息被添加进训练集，用来训练网络。随着越来越多 MI 的返回，积累的信息也逐渐增多。当网络被充分训练后，可确定从特征参数值到此类 MI 运行时间的映射关系。对于以后派遣同类 MI，只需从潜在目的主机采集同样的一组特征参数，输入神经网络。通过比较网络对于不同主机输出的预测运行时间，本文选择 MI 时可以采用最短运行时间的主机作为衡量主机硬件可用度的度量[8,9]。

4.2 $D(H_i)$

$D(H_i)$用于描述对于运行在动态变化的主机环境中的 MI 程序来说当前主机 H_i 的状态，即反映主机当前能否提供足够的资源。4.1 节提出的神经网络方法并不适宜用来确定 $D(H_i)$，因为主机的状态是动态变化的，频繁的主机探测状态要牺牲不小的性能代价[10]。

我们用与 MI 运行相关的主机资源的耗用量(或可用度)来衡量主机状态，比如当前 CPU 利用率、网络带宽、串口数量等。一般认为，只有当主机状态中涉及的所有资源都可用时，该主机状态才适合 MI 运行。这样一来，关键的一个问题是 MI 如何判断一种资源对于自身运行是否可用。用 r^{th} 来表示资源 r 的可用度阈值(Availability Threshold)。当一种资源的耗用度低于 r^{th} 时认为不可用，而当耗用度超出该阈值时认为该资源对于 MI 运行不可用。后一种情况下，MI 有必要迁移到另一台更合适的主机继续运行。

我们将系统时间看作是由微小单位时间(Unit Times)构成的。这样的一个单位时间是 agent 计算资源耗用情况并做迁移决定的常规周期。尽管实际中的网络是连续时间系统，但只要取的单位时间足够小，在此基础上得到的结果将会接

近对一个连续时间系统直接分析得出的结果。

p_{ri}：是资源 ri 的耗用量在某个单位时间内落在$[r_i^{min}, r_i^{th}]$内的概率。

P_{ri}：表示在 MI 的剩余运行时间内(假设为 l 个单位时间)，资源 ri 的耗用量总是在范围$[r_i^{min}, r_i^{th}]$内的概率。显然，如果对应于每一单位时间的 p_{ri} 都相同的话，则 $P_{ri}=(p_{ri})^l$。

利用一个概率模型来实现 MI 对迁移的判定，该方法基本思想如下：对于主机状态涉及到的每种资源，用耗用量的概率分布对其变化趋势建模。在每个时间步 T_i，算法可获得关于近期资源耗用的最新统计数据，并在此基础上对概率函数进行修正。同时，MI 预测其剩余运行时间(用单位时间的数量来衡量)，并将该数值传给主机。主机可计算对于每种资源的 p_{ri} 和 P_{ri}。此外，P_C^S 和$\overline{P_C^S}$也同时被计算。$\overline{P_C^S}$是 MI 做出迁移决定的参考。假如它比预先规定的基准低，MI 可认为在其剩余时间内主机状态的变化不会对其产生影响。否则，MI 应该迁移到另外一台更合适的主机来完成剩余的任务。本文采用预测任务完成时间作为目标主机动态状况的度量。

5 实验

本节将通过实验验证本文提出的迁移策略的有效性。实验主要通过对主机性能的评价，说明迁移策略与随机选择方法的优劣。

作为初步研究，我们设计让 MI 计算两个 100 阶矩阵的乘积，并将所有中间数据写入一个磁盘文件。虽然这只是一种较简单的情况，但从实验得出的结论同样适用于复杂的情况。

首先在每台主机上随机启动一些程序，用以区分不同的工作及性能；依次在选定的 6 台工作机 $H_1,\cdots,H_6$ 上进行评价，记录 $S(H_i)$和 $D(H_i)$。实验时，式(1)中 a 与 b 的取值均为 1。

表 1 各目标机可用度评测结果

实验次数	$S(H_i)$	$D(H_i)$	$V(H_i)$
H_1	41.8	26.1	67.9
H_2	40.7	23.3	64
H_3	44.6	24.9	73.9
H_4	45.7	28.2	73.9
H_5	51.4	26.4	77.8
H_6	48.6	28.3	73.9

从表 1 可以看出，不同的主机其静态性能指标和动态性能指标具有较大的差异。与采用随机方法相比，在实际迁移时选择 $V(H_i)$较小的目的主机能有效地提高系统的效率。

结束语 迁移实例的迁移策略在迁移工作流系统的整体性能表现中起非常关键的作用。本文针对迁移实例迁移时遇到的服务定位、迁移目的地选择等问题提出了新的解决方案：在服务定位方案中，提出了一个模型让 MI 对网络上的服务进行定位。该模型是分层结构的注册服务器集合，它们负责管理服务并为 MI 导航；在目的主机可用度评测方案中，提出了一个包含静态性能和动态性能的评价方法。该方法的有效性通过实验得到了验证[11,12]。

下一步研究将对本文提出的主机性能评测方法增加任务与服务之间的匹配程度来进行度量；同时赋予 MI 通过学习来智能选择特征参数进行评测的功能。

参 考 文 献

[1] Workflow Management Coalition. Workflow Management Coalition Terminology & Glossary WFMC-TC-1011[R]. 1996

[2] 曾广周，党妍. 基于移动计算范型的迁移工作流研究[J]. 计算机学报，2003，26(10)：1343-1349

[3] 刘大有，杨博，杨鲲. 基于旅行图的移动 Agent 迁移策略[J]. 计算机研究与发展，2003，40(6)：838-845

[4] 陈洪娜，祖旭，周峰. 工作流技术研究发展状况、研究内容及趋势[J]. 重庆工学院学报：自然科学版，2006(02)：65-69

[5] Wang J，Rosca D. Dynamic Workflow Modeling and Verification [J]. Lecture Notes in Computer Science，2006，4(1)：303-311

[6] 高丛. Mobile Agent 的迁移策略[D]. 青岛：中国海洋大学，2004

[7] 范玉顺，吴澄. 工作流管理技术研究与产品现状及发展趋势[J]. 计算机集成制造系统-CIMS，2000，6(1)：1-7

[8] 罗海滨，范玉顺，吴澄. 工作流技术综述[J]. 软件学报，2000，11(07)：899-907

[9] 孙瑞志，史美林. 支持工作流动态变化的过程元模型[J]. 软件学报，2003，14(1)：62-67

[10] 丁柯，金蓓弘，冯玉琳. 事务工作流的建模和分析[J]. 计算机学报，2003，26(10)：1304-1311

[11] 吴修国，曾广周，韩芳溪，等. 迁移工作流中的目标规划研究[J]. 计算机科学，2008，35(01)：147-150

[12] 唐卓，刘国华，李肯立. 多域环境下工作流访问控制时序策略组合研究[J]. 计算机科学，2011，38(01)：125-129

(上接第 203 页)

[6] Yao Jin-Shing，Lin F T. Fuzzy critical path method based on signed distance ranking of fuzzy numbers [J]. IEEE Trans on Systems，Man and Cybernetics，2000，30(1)：76-82

[7] Reisig W. Petri Nets：An Introduction [M]. Berlin，Heidelberg：Springer-Verlag，1985：17-135

[8] 袁崇义. Petri 网原理与应用 [M]. 北京：电子工业出版社，2005：114-188

[9] 吴哲辉. Petri 网导论 [M]. 北京：机械工业出版社，2006：27-100

[10] van der Aalst W M P，van Hee K. Workflow Management Models，methods and systems [M]. Cambridge：MIT Press，2002：22-115

[11] Murata T. Petri Nets：Properties，Analysis and Applications [J]. IEEE Trans on Software Eng，1987，77(4)：541-580

[12] 蒋昌俊. 离散事件的动态系统的 PN 机理论 [M]. 北京：科学出版社，2000：129-170

[13] 叶剑虹. Petri 网若干关键技术的研究与应用[D]. 成都：电子科技大学计算机科学与工程学院，2009

[14] van der Aalst W M P，Basten T. Inheritance of workflows：an approach to tackling problems related to change [J]. Theoretical Computer Science，2002，270：125-203

[15] Lautenbach K. Reproducibility of the Empty Marking [C]// the 23rd International Conference on Applications and Theory of Petri Nets. LNCS 2360. Adelaide，Australia：Springer，2002：237-253

[16] 叶剑虹，宋文，孙世新. 空标识可再生网的运算和性质分析 [J]. 计算机研究与发展，2009，46(8)：1378-1385