

面向内容的有限状态网络服务模型

刘 勇¹, 杨建华²

(1. 浙江大学工业控制国家重点实验室, 浙江大学先进控制研究所, 浙江杭州 310027)

2. 浙江大学国防科学技术研究院, 浙江大学计算机科学与技术学院, 浙江杭州 310027)

摘 要: 提出了一个新的基于有限状态机的网络服务模型, 该模型以一种有限状态移动主体(FS Agent)作为服务的基本单元, 分别用服务内容状态机(Content FS)和服务计算状态机(MACM FS, 即移动主体计算模型)来定义网络服务中的服务内容和计算过程, 从而构建了一种面向内容的网络服务模型. 实验结果表明, 该网络服务模型具有良好的可扩展性和较高的网络服务效率.

关键词: 网络服务; 服务内容; 服务计算过程; 移动主体; 有限状态机

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2008) 10-1897-06

Content Oriented Finite State Based Network Service Model

LIU Yong¹, YANG Jianhua²

(1. State Key Laboratory of Industrial Control Technology, Institute of APC, Zhejiang University, Hangzhou, Zhejiang 310027, China;

2. College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang 310027, China)

Abstract: This paper provides a novel finite state based network service model, which takes the finite state mobile agent (FS Agent) as the fundamental service unit. In this model, the service content is described by a service content finite state machine (Content FS), and the service computing process is depicted by a service computation model, which is defined as a mobile agent computation model based on finite state (MACM FS). Therefore we construct a content-based service model prior to those based on traditional techniques. The experiment results demonstrate that the model is more scalable and effective.

Key words: network service; service content; service computing; finite state machine

1 引言

传统的面向实现技术的网络服务模型要求服务设计人员在熟知如何设计服务内容和业务流程的同时, 必须精通具体的服务实现技术细节, 因而对服务设计人员要求较高, 且服务设计及实现过程需全面考虑实现技术与服务内容, 客观上限制了服务架构下的分布式网络计算的发展. 网格技术^[1]的出现特别是虚拟组织(Virtual Organization)^[2]技术的兴起, 使得构架在具体实现技术之上的网络节点间的自组织和管理得到了有力的技术支撑. 从而也使得网络服务的虚拟化成为可能, 能够从原有的面向具体实现技术的网络服务模型向与技术无关的面向服务内容设计的网络服务模型进行转变. 基于远程执行、请求代码(COM, CORBA)等概念和思想的面向技术的传统网络服务实现构架, 往往因为必须完全由

设计者去考虑当前广域网中存在的各种复杂状况和问题(如网络拓扑结构不稳定、广域网中带宽不足^[4], 以及远程执行过程中的资源等待等), 故服务实现过程极其复杂和繁琐. 为了克服此问题, 面向服务内容设计的网路服务模型必须具有足够的鲁棒性, 能够应对各种复杂网络状况.

本文针对面向内容的网络服务设计需求, 将移动主体(Mobile Agent)技术引入到网络服务的发布和实现过程中, 并利用该技术给出了一个复杂网络下面向内容的网络服务解决模型. 在本文提出的基于有限状态机的网络服务模型中(以下简称有限状态服务模型), 通过三个有限状态机分别描述了服务内容定义、复杂网络拓扑下的服务计算过程(即移动主体计算模型), 以及服务执行中的状态转换过程(即有限状态移动主体, 也称为服务单元).

收稿日期: 2007-12-05; 修回日期: 2008-07-23

基金项目: 国家博士后科学基金(No. 20070420231); 国家自然科学基金(No. 60675049); 浙江省自然科学基金(No. Y106414); 国家 863 高技术研究发展计划(No. 2008AA04Z209); 111 引智项目(No. B07031)

2 服务构架概述

当前, Web Service 是最具代表性的网络服务技术, 它是一种快速发展的基于 Web 的计算技术, 起源于电子商务的需求推动. Web Service 具有松散耦合、普遍通信和统一的数据格式等特点. 在 Web Service 模型中, WSDL^[5]用以描述服务, UDDI^[6]在线注册、发布和动态发现 Web 服务. 此外, WSFL^[7]和 XLANG^[8]描述服务是如何组合在一起的工作方式. 所有这些协议都建立在 XML 技术基础之上.

Web Service 作为一种集中式的服务体系模型, 大量的信息都集中在少数的 UBR 服务器上面, 故对服务器的运算能力、网络状况、存储能力等都提出了相当高的要求, 因此这些服务器很容易成为整体服务运行的瓶颈, 而且最为关键的是这种服务体系还是属于一种面向技术的服务模型, 无法提供更便捷的面向内容的服务设计模式.

目前普遍采用组合基本计算单元的方法实现面向内容的复杂服务设计, 这方面的研究工作有基于业务协议精化和组合的复杂业务过程设计与制定方法^[9]; 协议配置法^[12], 即提供全面的解决协议并依据实际情况配置特定的协议来实现不同类型服务的方法; 协议代数法^[13], 通过一组协议代数来描述和管理服务, 通过重用组合服务实现定制复杂服务的目的. 上述方法虽然能够在一定程度上灵活的合成复杂服务, 但是往往都存在以下弊端. 组合服务僵硬, 无法依据实际执行情况的变化进行动态适应. 此外由于对服务执行过程过于抽象, 没有考虑分布式执行环境下的资源失配等复杂网络情况, 因而对情况复杂的分布式网络适应性不强.

为此, 我们将移动主体技术和有限状态机理论引入了网络服务设计过程, 提出了一种面向服务内容的有限状态机网络服务模型. 在该网络服务模型中, 服务是由有限状态机上的一个或多个移动主体构成, 利用通用状态机的移动主体计算模型^[10], 即可实现面向内容的网络服务需求. 事实上, 更多的网络服务技术细节和复杂网络情况下的具体决策等问题, 都可由基于有限状态机的移动主体来实现. 由于有限状态服务模型是以移动主体作为执行单元的, 即服务的发布和执行都是以移动主体的实现而展开的, 故该服务模型是基于对等网络体系

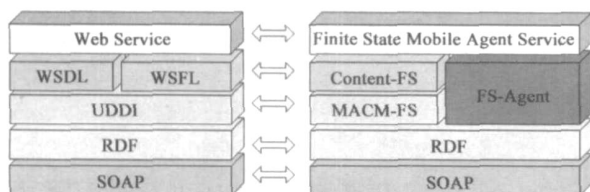


图1 有限状态服务模型分层结构

结构, 有效地避免了大量信息都集中于少数服务器上而造成的服务性能瓶颈问题, 即服务请求频繁时 UBR 服务器响应速度过慢.

3 基于有限状态机的网络服务模型

本文中的网络服务模型使用了三个状态机来描述整个服务的发布、执行等过程, 分别是有限状态移动主体(FS Agent)、服务计算状态机 MACM-FS(Mobile Agent Computation Model Finite State)和服务内容状态机 Content-FS, 其与 Web Service 中的协议对应关系如图 1 所示. 三个状态机之间的对应关系如图 2 所示, 其中, FS-Agent 对应于服务单元(Service Unit), 服务内容状态机对应于服务内容定义过程(Service Content), 服务计算状态机 MACM-FS 对应于服务计算过程(Service Computing).

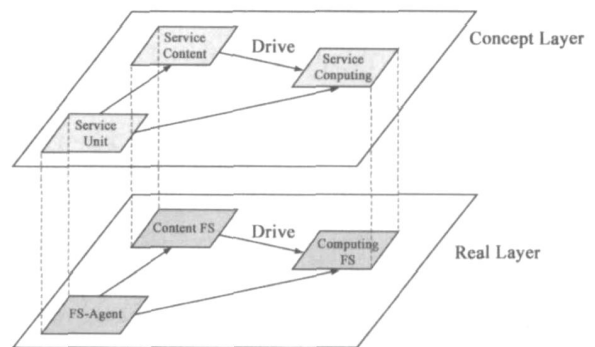


图2 有限状态服务模型原理示意图

在有限状态网络服务模型中, 网络服务的基本单位不再是某一种特殊的应用程序, 而是通用的移动主体. 通过 Content FS 来定义网络服务的计算内容(该服务内容与网络拓扑无关), 再通过 MACM-FS 执行对应的服务内容, 也称为服务计算过程(包含了在具体网络拓扑下的 Agent 迁移、执行、路径查找等).

3.1 有限状态移动主体(FS Agent)

为实现基于有限状态机的移动主体计算模型, 进而由移动主体形成有限状态网络服务模型, 这就要求在系统中迁移执行的移动主体必须能提供在不同节点上的执行能力. 为此, 本文提出一个简化的有限状态移动主体概念, 并将数据和资源区分开来. 其定义分别如下:

定义 1 有限状态移动主体 这是一种基于资源驱动的介于强移动模式与弱移动模式之间的移动主体系统. 它实质上是一种在外部资源或数据变化的影响下自动进入对应执行状态的有限状态机.

定义 2 数据 与移动主体自身状态转换有关却与执行节点无关的所有局部数据.

定义 3 资源 分布在移动主体的执行节点上的所有数据、设备、软件环境的统称.

本文将位于节点上的所有运行时参数和设备都统称为资源, 以区别于影响移动主体自身发生状态转移的

数据. 这样就可暂时不考虑网络拓扑对移动主体执行状态的影响, 故可直接从移动主体的执行状态转移角度描述其具体执行过程. 实际上, 在有限状态移动主体中, 我们不必关心当前已经执行到了具体哪一个节点上, 而只需了解移动主体当前已执行到什么状态.

有限状态移动主体由有限状态集、外部输入条件集和状态转移关系组成. 记作

$$FS-Agent = \{A_r, S_r, U_r, F_r\}$$

其中, A_r 表示 $FS-Agent$ 的标识, 它在整个系统运行过程中保持不变, 其作用是用以区分不同的移动主体. 在有限状态移动主体体系中, 所有的移动主体都采用通用的状态集合及通用的状态转换关系, 个体之间通过 A_r 来区分, 在服务状态运行相应的服务内容. S_r 是有限状态集合, 包括请求(Request) 状态、服务暂停(Suspend) 状态、服务阻塞(Block) 状态、迁移状态(Migration) 和服务(Serve) 状态. U_r 包含两种条件: 一种是服务状态对应的资源(Resource); 另一种是服务状态对应的服务的执行时间(Time); F_r 是状态转移关系, 状态转换关系如图 3 所示.

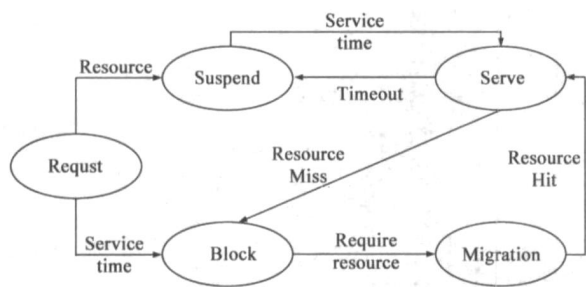


图3 有限状态移动主体的状态转换关系

由图 3 可知, 有限状态移动主体是在资源与服务时间共同驱动下的一个状态机. 在此状态机中, 从 Block 状态到 Serve 状态间加入了 Migration 状态, 实际上能起到如下作用: 当移动主体发现本地资源不足时, 就开始查找远程资源, 并迁移到具有足够下一阶段服务执行所需的资源的节点上.

3.2 服务内容状态机(Content FS)

移动主体的计算模型主要用来描述服务在实际网络拓扑情况下执行的过程和路径. 而对于网络服务内容本身的表达和描述, 本文提出了一种描述服务流程的状态机, 服务内容状态机. 这样就可将面向具体应用的网络服务从复杂多变的网络拓扑环境和执行技术细节中解脱出来, 从而使得服务设计者能够更侧重于服务内容本身的流程控制. 服务内容状态机的定义如下:

定义 4: 服务内容状态机(Service Content Finite State Machine) 是一个六元组

$$SF = (S_0, Q, R, E, \theta, F)$$

其中, Q 是所有服务中执行的状态集合, Q 中状态代表

的是服务具体运行过程中的每一个在资源驱动下的执行片断, 在有限状态机的服务模型中也称之为状态(等同于移动主体计算模型中的状态); S_0 是服务状态机的初始执行状态, 即服务开始的入口, 这里有 $S_0 \in Q$; E 是终结状态集合, 即所有网络服务可以停止状态的集合, $E \subset Q$; R 是网络服务中所有驱动状态转换的资源集合; θ 是服务状态集合在驱动资源作用下的状态转移函数:

$$\theta: Q \times R \rightarrow Q$$

F 是状态机的约束规则集合, 通过约束规则来定义满足需求的特定服务.

由上述服务内容状态机的定义可知, 实际上可由它给出移动主体计算模型执行的逻辑控制关系, 每一个定义的服务内容状态机都驱动移动主体计算模型的计算和展开, 在不同的网络拓扑情况和网络资源分布情况下, 同一个服务内容状态机驱动下的移动主体计算模型的计算过程(即移动主体的服务和迁移路径)可能都不相同, 但其执行的服务功能都是同样的, 即都由服务内容状态机来定义和描述.

3.3 复杂服务的构成

在有限状态机网络服务模型中, 构成服务的基本单元是移动主体, 服务是由一个或多个移动主体组成的, 因此, 存在着由多个移动主体复合成为一个服务的过程, 即存在着多个服务内容状态机的合成问题. 本文总结出的主要复合规则如下:

规则 1 在复合成复杂服务的设计过程中, 任何已经定义的简单状态机 $SF(S_0, Q, R, E, \theta, F)$ 都可看作是目的服务状态机 $SF'(S'_0, Q', R', E', \theta', F')$ 中的一个状态 S_{SF} ;

规则 2 复合后的服务内容有限状态机可以展开为完全由纯执行状态和驱动资源组成的有限状态机, 展开规则为规则 3、规则 4、规则 5、规则 6 和规则 7;

规则 3 目的服务状态机 SF' 的状态集合转变为 $Q' = Q' \cup Q$;

规则 4 目的服务状态机 SF' 的资源集合变为 $R' = R' \cup R$;

规则 5 目的服务状态机 SF' 的终结状态集合变为 $E' = E' \cup E$;

规则 6 目的服务状态机 SF' 的约束规则集合变为 $F' = F' \cup F$;

规则 7 对于目的服务状态机的转移函数有:

(1) 若 $\forall S_i$ 为简单状态机 SF 在目的服务状态机中的前驱状态, 加入转移关系 $S_i \times r_i \rightarrow S_0$, 其中, S_0 为简单状态机的初始状态, r_i 为 S_i 状态转换到简单状态机的驱动资源;

(2) 若 $\forall S_i$ 为简单状态机 SF 在目的服务状态机中

的后继状态, 加入转移关系 $S_k \times r_k \rightarrow S_j$, 其中, $S_k \in E$, E 为简单状态机 SF 终结状态集合。

3.4 服务计算状态机(MACMFS)

在实际的网络执行状况下, 随着网络拓扑的变化, 以及网络节点上资源的消耗与生成, 同个服务内容状态机的具体执行轨迹是不同的, 为了能够描述和规范在虚拟组织之上的服务执行过程, 本文提出了一个服务计算状态机, 也称之为基于有限状态机的移动主体计算模型。

在已有虚拟组织^[10]的基础上, 本文给出分布式系统中的基于有限状态机的移动主体计算模型, 其定义如下:

定义 5 服务计算状态机(移动主体计算模型)
 $MACM-FS = (R, S, M, \Phi, v, E, \Lambda, \varepsilon)$, 其中, R 是节点集合; S 是移动主体的有限状态集合, 在实际的网络服务中 S 就代表着每段服务执行代码; $M \subset S$ 是移动主体中所有处理消息状态的集合, $M = \{M_s, M_a\}$, M_s 是发送消息的状态, M_a 是接收消息的状态; $v \in R$ 是移动主体初次产生的节点, 移动主体在节点 v 被发布, 进入其状态转换周期; $E \subset R$ 是移动主体注销的节点, 移动主体移动到 E 中的节点后注销并结束其生命周期; Λ 为移动主体的迁移状态, 表示移动主体处在开始移动到另一个节点上将要进入新的执行过程的状态; ε 为空状态, 表示移动主体不执行任何运行状态或迁移状态; Φ 是从 $R \times (S \cup \{\Lambda, \varepsilon\})$ 到 R 的转换函数, 即

$$\Phi: R \times (S \cup \{\Lambda, \varepsilon\}) \rightarrow R$$

使得

- (1) 对所有 $R_i, R_j \in R$, 若 $\Phi(R_i, \varepsilon) = R_j$, 则 $R_i = R_j$;
- (2) 对所有 $R_i, R_j \in R$, 若 $\Phi(R_i, \Lambda) = R_j$, 则 $R_i \neq R_j$;
- (3) 对所有 $R_i, R_j \in R, S_k \in S$, 若 $\Phi(R_i, S_k) = R_j$, 则 $R_i = R_j$;

- (4) 对所有 $R_i \in R$, 若 $\Phi(R_i, M_a) = R_j$, 则下一状态转换关系为 $\Phi(R_j, M_s) = R_i$ 。

移动主体计算模型中的转移状态 Λ 是建立在虚拟节点组织结构中的节点间的通信基础上的, 利用基于消息的通信算法^[10], 将移动主体从源节点迁移到目的节点。在虚拟节点的底层结构的支撑下, 移动主体的计算模型可实现高效的状态转换、位置迁移和消息通信等功能。

从上述移动主体的计算模型中不难发现, 移动主体的所有转移、通信(消息模式)及异地执行过程都被当作移动主体计算模型中的一个状态来处理。采用消息传递机制的传统移动主体系统普遍存在通信失效问题^[11], 这是由于移动主体具有自主迁移的能力, 移动主体迁移和消息发送的异步性导致移动主体可能在消息传输过程中离开当前节点, 从而无法收到该消息。类似地, 基于

广播等方式^[14]实现的移动主体寻址和通信算法都存在数据传输量过大, 在有限带宽条件下容易阻塞, 且可靠性较低等问题。在基于有限状态机的移动主体计算模型中, 因为通信过程被状态化, 并定义了相应的规则来保障消息通过程中的发送消息与接收消息操作的对称性, 因而可以保障通信过程与迁移过程的原子性, 从根本上保障了通信与移动是一个时序逻辑关系, 不存在通信失效问题。

4 实现

我们在 Java+XML 平台下开发了一个基于 Mobile Agent 的网络服务部署原型系统, 如下图 4 所示。该系统称之为 Agent Home, 部署在每一个支持 Mobile Agent 网络服务的网络节点上。Agent Home 的功能主要有两个部分组成, 分别是群组的管理(Virtual Group Management)和基于 Mobile Agent 的网络 P2P 服务管理。

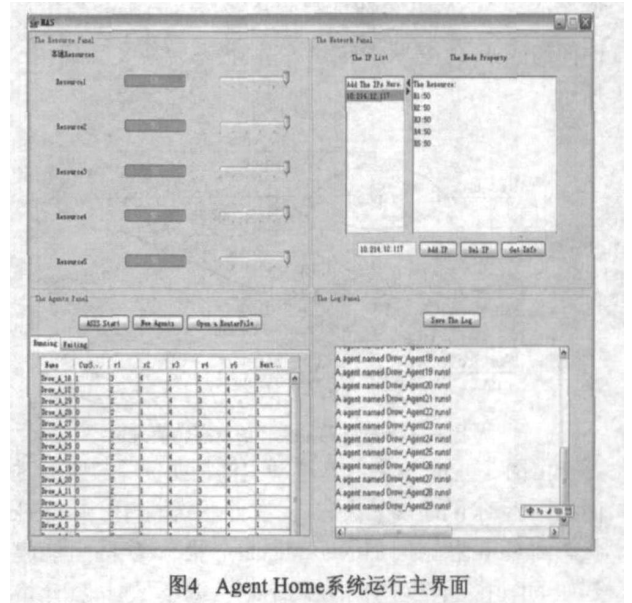


图4 Agent Home系统运行主界面

Agent Home 的群组管理功能实现了对网络节点(Peer)的分组管理以及节点的动态加入和离开等网络拓扑层的协议^[2], 此外, 各个网络节点间的网络通讯以及群组间的资源的查找都是由 Agent Home 代理实现的。

Agent Home 的另外一个主要功能是对网络 P2P 服务的管理, 即对 Mobile Agent 的部署、执行以及调度。而本文提出的基于有限状态机的 Mobile Agent 计算模型中通过 FS-Agent 状态机抽象出了所有 Mobile Agent 的执行过程, 这样就使得 Agent Home 对 Mobile Agent 的管理, 可以看作一个 FS-Agent 状态机的循环执行过程。

在 Mobile Agent 网络服务模型中, Mobile Agent 是通过一个 Content FS 来描述其具体的执行逻辑流程了, 在原型系统中, 我们使用 XML 技术实现对 Mobile Agent 的执行流程的描述和表达, 然后通过 Agent Home 中内嵌的

解析器, 分析并且根据 Content-FS 的逻辑流程来调度和发送迁移 Mobile Agent 服务. 图 5 给出了一个简单的 XML 格式的 Content-FS 描述. 图 5 中的描述文件主要由三部分构成, 分别是 Content-FS 的状态 (STATE 标签), 每一状态执行所需要的资源 (RDF 描述) 以及状态执行函数 (图中的 prepare_for_Mining 和 Mining_parallel).

```
<?xml version="1.0"?>
<Content FS>
  <STATE stateid="0" call="prepare_for_Mining" nstatenum="1">
    <NextS stateid="1">
      <rdf:RDF>
        <rdf:Description>
          rdf:about="http://icad.8866.org/Mining/localhost">
            <facilities:CPU>available</ facilities:CPU>
            <facilities:location>localhost</ facilities:location>
            <facilities:memory_less>256M</ facilities:memory_less>
          </rdf:Description>
        </rdf:RDF>
      </NextS>
    </STATE>
    <STATE stateid="1" call="Mining_parallel" nstatenum="1">
      <NextS stateid="2">
        <rdf:RDF>
          <rdf:Description>
            rdf:about="http://icad.8866.org/Mining/Rcmote">
              <facilities:CPU>available</ facilities:CPU>
              <facilities:location>localhost</ facilities:location>
              <facilities:memory_less>512M</ facilities:memory_less>
            </rdf:Description>
          </rdf:RDF>
        </NextS>
      </STATE>
    </Content FS>
```

图5 一个简单的Content-FS描述文件

5 讨论与实验

本文服务模型中的 Content FS 是建立在抽象的服务逻辑控制层, 故可以使用诸如文献[9~13]中的协议和组合方法, 具有很大程度的自由性. 同时 FS Agent 和 MACM-FS 给定了服务执行过程中遇到网络资源失配等复杂状况时的应对策略, 保证服务能够高效顺利的执行. 因此本文的服务模型即保留了 Web Service 中多协议组合模型的灵活性, 同时又可以根据分布式网络中的具体复杂情况进行自适应调节控制, 提高整个系统对外提供服务的总体性能. 下面将通过实验来验证本文模型的效率.

由于网络服务模型一般都按照移动主体执行“服务请求→查找资源→状态迁移→执行服务→服务请求”模式进行工作, 故移动主体执行的效能就决定了网络服务性能.

本文采用文献[3]中的仿真环境, 模拟一个网络结点随机分布的环境, 以此仿真虚拟组织^[2, 10, 15]的结点管理和通讯系统. 在此系统中, 采用文献[16]中给出的移动主体带宽耗费计算方法, 并且在其基础上给出了一个新的面向服务的移动主体性能评估参数—有效服务率.

有效服务率 (Service Availability, 缩写为 $SrvAvl$) 是某个移动主体在节点上的服务时间与它所消耗的总时间 (其迁移时间和搜索时间以及服务时间之和) 之比. 其计

算公式如下:

$$SrvAvl = \frac{T_{service}}{T_{service} + T_{migration} + T_{search}} \quad (1)$$

其中, $T_{service}$ 是移动主体在节点上的服务时间, $T_{migration}$ 是移动主体的迁移时间, T_{search} 是移动主体搜索目的节点所耗费的时间. 由公式 (1) 可知, 在服务时间保持不变的情况下, 减少移动主体在迁移和搜索过程中的时间消耗能有效地提高系统的服务率. 在移动主体模型中, 移动主体的迁移时间 $T_{migration}$ 可利用如下公式计算:

$$T_{migration}^{ij} = \frac{M_{agent}}{B} = \sum_{q=1}^{|R_i R_j|} \frac{M_{agent}}{B_q} \quad (2)$$

其中, M_{agent} 是移动主体的字节大小, B_q 是移动主体从节点 R_i 迁移到节点 R_j 过程中第 q 个网段中的传输速率. 一般情况下, 我们可以取节点 R_i 与 R_j 之间的平均传输速率 B . 单个移动主体的运行情况实验见文献[3], 本文中重点探讨多个移动主体情况下的服务运行性能.

为了衡量多个服务部署情况下的系统性能, 在实验中逐步增加移动主体 (Agent) 数目并验证其对整体性能的影响, 该实验采用 4 状态的内容状态机 (Content FS), 网络带宽设置为 512KB.

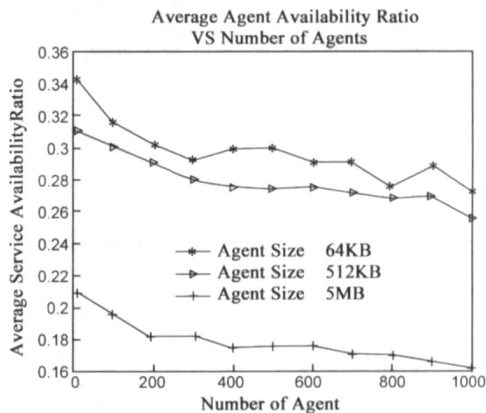


图6 Agent数目与平均服务有效率关系

实验结果如图 6, 图 7 所示. 从图中可以看出, 在 Agent 数目显著增加的情况下, 其平均有效服务率随之下降, 但保持比较平缓的趋势. 类似的, 在 Agent 平均执行

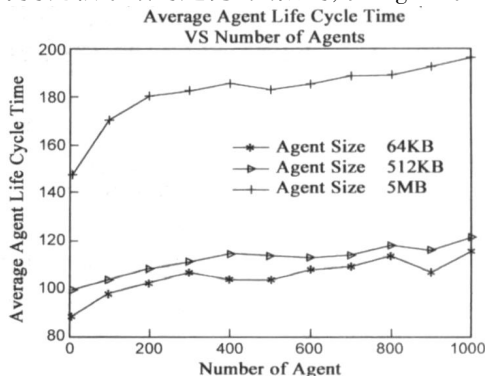


图7 Agent数目与平均执行总时间关系

总时间上,随着 Agent 数目的增加,其平均执行总时间平缓增加。从图 6 和图 7 的实验结果同样可以看出 Agent 的大小能够显著的影响其执行总时间和有效服务率。实验证明在多 Agent 执行情况下,系统整体性能仍能保持较满意的程度,该模型能够比较好的解决多个 Agent 间的执行通信损耗和资源竞争等问题。

6 结束语

本文给出了一种新的利用移动主体来实现面向内容的网络服务方法。由于将网络服务内容和具体的服务执行过程分离开来,分别用服务内容状态机(Content-FS)和服务计算过程状态机(MACM-FS)来表示,因此该模型可更好地实现与技术无关的网络服务设计和发布。有限状态服务模型以 FS Agent 作为执行单元,采用的是对等体系结构,故避免了传统的集中式网络服务模型中存在的很多问题。

进一步的研究工作将集中于建立一个可重用的网络服务状态库,并针对一些特定的网络体系(如 Ad Hoc 网、移动通信网等)或特殊类型的网络服务制定相应的约束规则。此外随着语义网的普及和发展,如何提高 Agent 本身的智能性,更好的理解和识别网络资源,提高服务质量也是一个非常具有挑战性的研究方向,也是富有实用与理论价值的研究方向。

参考文献:

- [1] Foster I, Kesselman C, Tuecke S. The anatomy of the grid: enabling scalable virtual organizations [J]. International Journal of Supercomputer Applications, 2001, 15(3): 200–222.
- [2] Huang Li Can, Wu Zhao Hui, Pan Yur He. Virtual and dynamic hierarchical architecture for E science grid [J]. International Journal of High Performance Computing Applications, 2003, 17(3): 329–347.
- [3] Liu Yong, Xu Cong-Fu, Chen Yarr Yu, Pan Yur He. Simulating a Finite State Mobile Agent System [A]. In Proc. WAIM 2005 [C], 2005. 651–656.
- [4] Waldo J, Wyant G, Wollrath A, Kendall S. A note on distributed computing [R]. Technical Report TR 94 29, Sun Microsystems, November 1994.
- [5] Christensen, E., Cudra, F., Meredith, G., Weerawarana, S. Web services description language (WSDL) 1.1. W3C, Note15 [OL]. <http://www.w3.org/TR/wsdl>, 2001
- [6] UDDI. [OL] <http://www.uddi.org/>
- [7] Web services flow language (WSFL) Version 1.0 [OL]. <http://www4.ibm.com/software/solutions/WebServices/pdf/WSFL.pdf>

- [8] Web services for business process design[OL]. <http://www.gotdotnet.com/team/xml-wsspecs/xlangc/default.htm>
- [9] Desai N, Mallya AU, Chopra AK, Singh MP. Interaction protocols as design abstractions for business processes [J]. IEEE Trans. On Software Engineering, 2005, 31(12): 1015–1027.
- [10] Liu Yong, Xu Cong-Fu, Wu Zhao Hui and Pan Yur He. A finite state mobile agent computation model [A]. In Proc. AP Web 2004[C]. Springer Press, 2004. 152–157.
- [11] Jiannong Cao, Xinyu Feng, Jian Lu, Sajal K. Das. Mailbox Based Scheme for Designing Mobile Agent Communication Protocols [J]. IEEE Computer, 2002, 35(9): 54–60.
- [12] Bartolini C, Preist C, Jennings NR. Architecting for reuse: A software framework for automated negotiation [A]. In Proc. of the 3rd Int'l Workshop on Agent Oriented Software Engineering[C]. London: Springer Verlag, 2002. 87–98.
- [13] Benattallah B, Casati F, Toumani F. Representing, analysing and managing Web service protocols [J]. Data & Knowledge Engineering, 2005, 58(3): 327–357.
- [14] Murphy, A., Picco, G. P. Reliable communication for highly mobile agents [A]. In Proc. ASA/MA'99[C], 1999. 141–150.
- [15] Liu Yong, Xu Cong Fu, Wu Zhao Hui, and Pan Yur He. A predictable mobile agent computation model and its fabric architecture [A]. In Proc. AWCC'2004[C], Springer Press, 2004. 18–26.
- [16] Kotz D, Cybenko G, Gray R S, Jiang G, Peterson R A, et al. Performance analysis of mobile agents for filtering data streams on wireless networks [J]. Mobile Networks and Applications, 2002, 7, 163–174.

作者简介:



刘 勇 男, 1980 年 10 月生于江西萍乡, 博士, 浙江大学工业控制国家重点实验室先进控制研究所讲师。主要研究领域为智能计算、模式识别、海量信息处理、智能机器人。
E-mail: yongliu@ipc.zju.edu.cn



杨建华 男, 1973 年生于浙江临海, 博士, 浙江大学讲师。主要研究领域为人工智能、Web 服务。(本文通讯作者)
E-mail: jhyang@zju.edu.cn