

基于 ESA 的网格资源管理模型

史文翀^{1,2}, 曾文华^{1,2}

(1. 厦门大学软件学院, 厦门 361005; 2. 智能信息技术福建省重点实验室, 厦门 361005)

摘 要: 提出了基于 ESA 的网格资源管理模型。该模型以 Service-Agent 为基础, 结合了 Agent 动态自主性和服务的松散耦合优点, 运用经济学的市场机制和交易理论, 提出该模型的资源发现机制、分配选择机制和交易机制, 以及基于议价交易机制的 Bargaining_Min_min 算法, 实现资源的优化管理, 使资源提供者 and 使用者满足各自利益, 达到网格市场上的双赢。

关键词: 网格资源管理; 经济机制; 服务代理; Bargaining_Min_min

Grid Resource Management Model Based on ESA

SHI Wenchong^{1,2}, ZENG Wenhua^{1,2}

(1. Software School, Xiamen University, Xiamen 361005;

2. Key Laboratory for Intelligent Information Technology of Fujian Province, Xiamen 361005)

【Abstract】A service-agent grid resource management model based on economic mechanisms is proposed. Based on Service-Agent, it combines the dynamic agent with loose coupling of services. The market mechanisms and trading theories such as resource discovery mechanism, resource allocation mechanism and Bargaining_Min_min based on bargaining mechanism, are also adopted to achieve the optimal allocation of grid resources so that providers and consumers can maximize their own interests and try to get the win-win objective in grid market.

【Key words】 Grid resource management; Economic mechanism; Service agent; Bargaining_Min_min

网格以资源共享为目的, 将属于不同个人和组织的资源从逻辑上组合在一起, 因此选择一个合适的模型来管理网格中的各类资源是非常重要的。网格资源管理模型按体系结构分为 3 类: 层次模型, 抽象所有者模型和经济/市场模型^[1]。在网格环境中, 资源提供者提供自身拥有的资源而获利; 资源使用者通过购买资源而完成在自己资源上无法完成的任务, 这种模式非常适合运用现实生活中的经济模型来管理网格资源。目前将经济机制应用于网格资源分配的研究非常活跃, 主要被研究的经济学模型^[2]包括拍卖模型、标价模型、招标/合同网模型和基于投标的均衡资源共享模型等, 并已在实际大型网格系统中应用, 如 Nimrod-G^[3]在资源管理体系中采用的就是典型的基于多商品的市场模型, 它重点研究完全竞争的网格经济环境下应用启发式调度算法。

Agent 技术起源于分布式人工智能的研究。基于多 Agent 的网格资源管理模型^[1]通过移动 Agent 收集资源的变化信息, 具有较强的可测量性和适应性, 用户只需要在资源请求代理上进行一级调度, 将任务提交给资源请求代理, 而代理背后对于用户来说都是透明的。

1 基于 ESA 的网格资源管理模型

网格资源管理的基本功能是接受来自网络范围内用户的资源请求, 并把特定的资源分配给资源请求者, 合理调度相应的资源, 使请求资源的任务顺利执行, 包括资源请求提交、资源发现、资源调度、资源分配、资源监测和资源注销等, 要求所有功能都要协调进行, 使网格系统获得较高性能。

1.1 ESA 模型功能框架

传统网格资源管理存在着一些问题, 资源在各层次上如何表示是比较突出的问题之一, 用户和资源之间容易出现识别、分配上的问题。而将异构的资源转化为表示统一、接口

一致的服务并在服务目录中进行注册是可以解决上述问题的。ESA 将服务的概念引入到网格资源管理, 采用细粒度服务来封装各类操作, 比如执行分配、调度、回收、授权和命名等操作。同时, ESA 将所有的资源都以网格服务的形式出现, 这样便于对服务的识别和协商调用。

如图 1 所示, ESA 模型主要分为 3 个功能模块:

(1) 服务创建功能: ESA 通过定义一系列的接口来支持服务代理实例的创建和映射。当用户需要提供所拥有的资源时, 只要对映射代理提出加入的申请, 通过服务工厂的 Create 操作, 创建一个新的网格服务实例, 并在服务目录中注册, 返回全局唯一的网格服务句柄 GSH 给映射代理供映射资源时使用。该用户资源得到服务标识后便被加入到资源池中, 服务代理携带生成服务的基本信息(比如句柄、底价等), 并发布到黑板代理上, 供给任务代理查找。

(2) 服务调用功能: 用户向网格门户提出服务请求, 门户为其提供身份认证证书, 然后将服务请求和证书提交给经纪人代理。经纪人代理作为用户在系统中的唯一代表, 它根据 WSDL 定义生成用户代理, 将请求通过任务代理形成一个任务协作流。任务代理通过服务描述库识别经纪人代理请求的服务和黑板代理发布的资源服务描述, 查找到执行该任务所需的资源服务。任务代理可以对性能、服务提供地点、价格等各个方面提出相应的要求, 可以和各服务代理进行协商和交易, 最终完成用户的请求。

(3) 安全认证功能: 门户通过安全认证代理判断此请求用

基金项目: “985 工程” 智能化科技创新平台基金资助项目

作者简介: 史文翀(1981—), 男, 硕士生, 主研方向: 网格体系结构和资源管理; 曾文华, 博士、教授、博导

收稿日期: 2006-05-09 E-mail: xmu_wchong@126.com

户注册信息是否已经注册授权,如果是则直接返回该用户的授权证书;如果没有历史注册记录,则会生成临时用户的安全证书,用于查找、绑定服务时的权限,保证资源的合法使用。

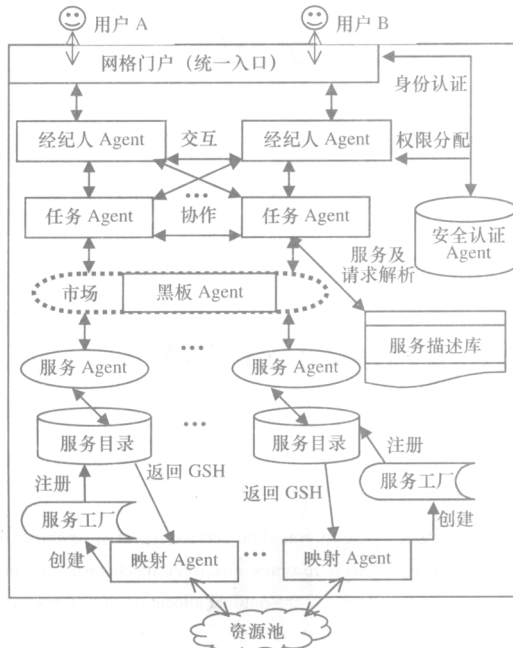


图1 ESA模型

1.2 基于ESA的资源发现机制

图2是基于ESA模型的资源发现机制框架:网格被划分为若干个虚拟组织,每个虚拟组织都由一个全局服务目录和多个结点组成。全局服务目录只存储网格资源的名称和位置信息,而结点可以是工作站、服务器,甚至是局域网。

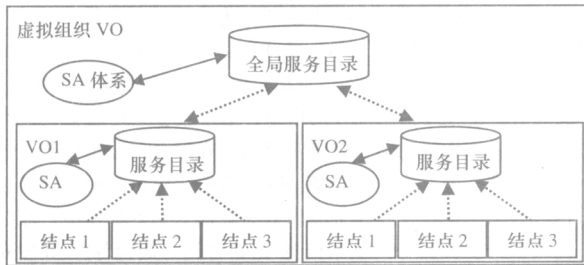


图2 基于ESA的资源发现机制

结点间按逻辑需要组成有层次的结构,每个结点都部署一个SA(Service-Agent)体系和一个本地服务目录。本地服务目录存储本结点的资源状态信息。为了提高网格资源发现的响应时间和执行效率,当本结点查询到其它结点的资源信息时,也将其存入本地服务目录中,并加盖有效时间戳。当接收到一个新的查询请求时,如果发现缓存中的资源信息仍在有效时间范围内,则直接从本地服务目录中读取数据并返回。如果发现缓存中的资源信息已过期,则到该虚拟组织的目录服务器中重新查询。如果还找不到,则由服务目录创建Agent到上一级或全局服务目录中查询。

对于整体网格系统和每个虚拟组织内部的资源管理体系框架都如图1所给出的ESA模型。在虚拟组织内部,各结点任意加入退出的过程就是ESA模型中服务创建和注销的过

程。而虚拟组织之间通过移动Agent进行交流通信。

1.3 基于ESA的服务分配选择机制

ESA模型中采用市场机制来实现服务的分配:虚拟组织中各类结点可以向外出售服务,如CPU时间、存储空间等。相反,如果系统中的服务不能满足各自请求,它们也可以派遣代理从外部查找资源服务,并可以根据需求从资源服务拥有者那里购买或交换所需资源服务。

服务选择机制的原理如下:

- (1)服务代理周期性地检测最新服务目录信息,并发布到黑板代理上予以通告更新,方便任务代理进行查询搜索。
- (2)任务代理根据经纪人代理的请求需要,在黑板代理上查询服务代理发布的服务信息。
- (3)任务代理与服务代理进行协商,抉择是否选择购买该服务。如果协商成功,任务代理与服务代理分别代表各方的利益进行交易,并将最终执行结果返回。

1.4 基于ESA的资源交易机制

ESA模型中,服务代理携带相关信息,根据市场供需等情况,做出是否与任务代理交易的决策;而任务代理体现着用户的利益,根据服务的质量和价格等情况做出交易决策;黑板代理起到市场和仲裁人的作用,一方面使交易模型接近实际市场交易形式,同时又可以使任务代理和服务代理的议价算法最终能收敛或停止。

ESA模型提出了基于议价仲裁交易机制的算法。下面给出了代理的简单议价仲裁算法和相关定义:

- (1) $TA = \{TA_1, \dots, TA_i, \dots, TA_n\}$: n 个任务代理队列;
- (2) $SA = \{SA_1, \dots, SA_j, \dots, SA_m\}$: m 个服务代理队列;
- (3) $ETC(TA_i, SA_j)$: TA_i 使用 SA_j 后的预期执行时间。 n 个任务代理使用 m 个不同服务代理后的预测执行时间是一个 $n \times m$ 的ETC矩阵;
- (4) $DeadLine_i$: 完成 TA_i 的最终期限,如果完成时间超出 $DeadLine_i$, TA_i 将不会与 SA_j 进行议价;
- (5) BP_i : 用来表示 TA_i 的当前出价;
- (6) HP_i : 表示 TA_i 最高的心理价位,如果 SA_j 的标价高于 HP_i , TA_i 将拒绝与 SA_j 进行交易;
- (7) MP_j : SA_j 所出的市场价格,即市场标价;
- (8) FP_j : SA_j 的底价,如果 TA_i 的出价低于 FP_j , SA_j 将拒绝与 TA_i 进行交易;
- (9) Acq_num_i : 当前 TA_i 可以选择获得SA的数量,这个参数直接影响双方协商的价格, Acq_num_i 越多, TA_i 选择余地越大,则每次 TA_i 出价就会越低;
- (10) Req_num_j : 当前请求 SA_j 的TA数量,这个参数直接影响双方协商的价格, Req_num_j 越多, SA_j 选择余地越大,则每次 SA_j 的标价就会越高;
- (11) QoS_{TA_i} : TA_i 的服务质量,包括该任务代理的信用度、交易诚意及交易历史等;
- (12) QoS_{SA_j} : SA_j 的服务质量,包括资源稀缺程度、服务的性能及可靠性等;
- (13) $y = f(x)$: 值为域为(0,1)的边际协商价格函数,自变量 x 为还价次数;
- (14)定义TA, SA 出价计算公式

$$BP_{i_k} = MP_{j_{k-1}} - \Delta BP_{i_k} \quad (1)$$

$$MP_{j_k} = BP_{i_{k-1}} - \Delta MP_{j_k} \quad (2)$$

$$\Delta BP_{i_x} = f(\alpha \times \frac{Acq_num_i}{x \times QoS_{SA_j} \times Req_num_j}) \times \Delta MP_{j_{x-1}} \quad (3)$$

$$\Delta MP_{j_x} = -f(\beta \times \frac{Req_num_j}{x \times QoS_{TA_i} \times Acq_num_i}) \times \Delta BP_{i_{x-1}} \quad (4)$$

式(1)表示, TA_i 的第 x 次出价是 SA_j 前一次标价减去此次的协商价格; 式(2)的意义与式(1)相似, 即 SA_j 的第 x 次标价等于 TA_i 前一次出价减去此次的协商价格; 式(3)、式(4)计算双方的协商价格, 表示双方的协商价格与对方前一次的协商价格有关, 并且通过一个边际协商价格函数反映了其他相关参数与双方协商价格的关系, 其中, α 、 β 为边际系数。

(15) 任务代理议价算法

```
Request for BBA (Service Description)
Query the Service on BBA
Receive from BBA ( $MP_{j0}$ ) //从黑板 Agent 得到对方出价
While ( $MP_{jx} < HP_i$  & ETC ( $TA_i, SA_j$ ) < DeadLine)
//有议价空间
{
    Receive from BBA ( $MP_{jx}$ , arbitrate)
     $\Delta BP_{i_x} = f(\alpha \times \frac{Acq\_num_i}{x \times QoS_{SA_j} \times Req\_num_j}) \times \Delta MP_{j_{x-1}}$ 
     $BP_{i_x} = MP_{j_{x-1}} - \Delta BP_{i_x}$ 
    Send to BBA ( $BP_{i_x}$ ) //送出自己报价
}
if (arbitrate == 1)
    Allocate the  $SA_j$  to  $TA_i$ ; //分配资源
```

(16) 服务代理议价算法

```
Receive from BBA ( $BP_{i0}$ ) //从黑板 Agent 得到对方出价
while ( $BP_{i_x} > FP_j$ ) //有议价空间
{
    Receive from BBA ( $BP_{i_x}$ , arbitrate)
     $\Delta MP_{j_x} = -f(\beta \times \frac{Req\_num_j}{x \times QoS_{TA_i} \times Acq\_num_i}) \times \Delta BP_{i_{x-1}}$ 
     $MP_{j_x} = BP_{i_{x-1}} - \Delta MP_{j_x}$ 
    Send to BBA ( $MP_{j_x}$ ) //送出自己报价
}
if (arbitrate == 1)
    Allocate the  $SA_j$  to  $TA_i$ ; //分配资源
```

(17) 黑板代理仲裁算法

```
arbitrate=0 //裁决结果, 初始为 0, 表示可以议价
While (Listening to SA & TA)
{
    Receive from TA ( $BP_{i_x}$ )
    Receive from SA ( $MP_{j_x}$ )
    if ( $|MP_{j_x} - BP_{i_x}| < \delta$  & Exchange_Time <  $\mathcal{E}$ )
        arbitrate=1; //仲裁  $TA_i$  获得  $SA_j$ ;
    else if ( $|MP_{j_x} - BP_{i_x}| > \delta$  & Exchange_Time >  $\mathcal{E}$ )
        arbitrate=-1; //协商失败
    Send to TA ( $MP_{j_x}$ , arbitrate) //送出他方的出价
    Send to SA ( $BP_{i_x}$ , arbitrate) //以及裁决结果
}
```

2 基于议价交易机制的调度策略

网络调度算法研究^[4]表明, Min_min 算法既能得到相对较好的性能, 而且运算速度也比 GA 和 A* 更快, 可以适应大规模的计算环境, 是网络调度算法的研究基础之一。

本文对 Min_min 算法进行改进, 提出基于议价交易机制

的 Bargaining_Min_min 调度策略。该调度策略既考虑资源对任务的执行效率, 又兼顾网格市场上资源提供者和请求者双方的利益, 起到均衡负载的效果。

2.1 调度策略参数定义

本文假设在 ESA 模型下, 任务代理通过服务代理在黑板代理上发布的信息发现该服务资源, 并通过双方议价得到协商价 Negotiate_price[TA, SA]。

对每个请求服务的任务代理, 定义以下参数:

(1) MCT[i, j]: 第 i 个任务代理使用第 j 个服务代理后预测最小完成时间, 矩阵表示各任务代理使用服务代理后不同预测完成时间。

(2) Get_Service_Time[j]: 表示当前各任务代理可以获得第 j 个服务代理的等待时间;

(3) Complete_Time[i, j]: 任务代理在动态环境下的实际完成时间。如果 TA_i 使用 SA_j 完成任务, 则定义 Complete_Time[i, j] = ETC[i, j] + Get_Service_Time[j]。

2.2 Bargaining_Min_min 算法描述

核心算法描述如下:

当需要调度的任务代理队列 U 非空时, 反复执行如下操作直至任务代理队列 U 为空:

(1) 对于非空任务代理队列 U 中每一个等待分配的 TA_i , 计算出所有 TA_i 可请求 SA 的最小完成时间 Min_Time 和次小完成时间 Second_Min_Time。假设该任务代理使用第 k 个服务代理后的完成时间为最小(或次小), 则记为

$$Min_Time[TA_i] = MCT[TA_i, SAK]$$

$$(或次小 Second_Min_Time[TA_i] = MCT[TA_i, SAK])$$

再计算两者差值

$$Diff_Value[TA_i] = Second_Min_Time[TA_i] - Min_Time[TA_i]$$

其中, Diff_Value[i], Min_Time[i], Second_Min_Time[i]

都定义为一维数组;

(2) 对于每个 TA, 计算 $\frac{Min_Time[TA_i]}{Negotiate_price[TA_i, SA_j] \times Diff_Value[TA_i]}$, 选择值最小的任务代理, 将其请求的服务代理分配给它;

(3) 从需要调度的非空任务代理队列 U 中把该任务代理删除, 同时更新 MCT 矩阵;

(4) 循环该过程, 直至需要调度的队列 U 为空。

3 模拟实验结果

本文采用 SimGrid 对算法进行模拟实验, 它提供了一系列 API 来随机生成不同的参数模拟网络异构环境。首先模拟 TA 和 SA 议价的过程, 得到协商价格。议价过程中, TA 和 SA 进行双向选择, 性能和价格两个因素起到了“杠杆”的作用。图 3 显示了 TA_6 和 SA_4 基于议价仲裁交易算法的议价过程。图 4 显示了所有 TA 匹配 SA 的成交价对比。可以发现, 由于 SA_1 、 SA_3 性能较好, 使大多数 TA 执行效率都比较高, 因此市场需求竞争比较激烈, 其相应的成交价格基本没有降低; 而其他 SA 在不低于底价的前提下, 都有或多或少的降低以满足 TA 的购买请求, 达到最终成交。

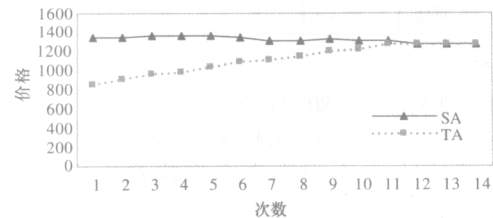


图3 协商价格曲线

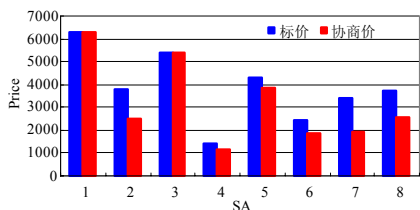


图4 协商成交价

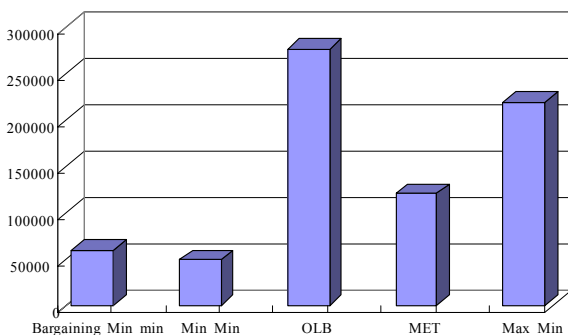


图5 makespan 对比图

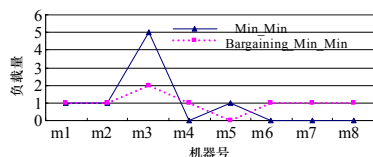


图6 机器负载均衡对比图

Bargaining_Min_min 算法是基于议价仲裁交易机制的改进 Min_min 算法。从图 5 和图 6 可以看到，本算法除了保持原有 Min_min 算法 makespan 情况较好的优点外，也提高了

(上接第 95 页)

小区域都至少有 4 个测试点，记录它们的实际位置(坐标)和接收到的信号强度。

虽然实验的结果很大程度上受到物理区域的实际环境影响，但表 1 显示，在实验的特定环境中，MMCCPA 比 MCCPA 在算法定位的精确度上有所提高。改进后的算法与原来的最大区别在于选择子群的过程。现在综合考虑了定位支持线的数量和历史路径两个因素；另外在计算时，只考虑了上次测量位置附近的区域，因此大大减少了计算量，提高了计算速度，同时也提高了定位的精确度。其问题是，后一次的定位结果很大程度上依赖于前面的定位结果，所以提高第 1 次定位的准确性对定位系统的整体性能至关重要。

表 1 实验结果对照

方法 \ 区域识别	正确次数	错误次数	正确率(%)	平均误差距离(m)
MCCPA 算法	82	22	78.8	2.3
MMCCPA 算法	98	6	94.2	1.5

5 总结

本文研究在基于 802.11 的定位系统中，假设在定位前一切初始数据都已经形成，因此没有详细介绍诸如 AP 如何摆放、参考点的选择以及参考点信号强度的获取等问题，而是将重点放在了定位算法的研究上。

MMCCPA 仅仅通过每一个移动终端测量到的信号强度

资源的利用率，能够充分均衡现有的资源使请求者更好地执行任务。

4 总结

本文提出了基于 ESA 的网格资源管理模型。ESA 模型以 Service-Agent 为基础，依靠市场机制和交易理论来实现资源的优化管理。基于 ESA 模型的资源发现、分配和基于议价交易机制的 Bargaining_Min_min 算法完全模仿了现实市场上买卖双方的交易机制，使资源提供者和使用者在网格虚拟市场上满足各自利益，达到双赢。因此，基于 ESA 的网格资源管理模型及其机制是一种比较有效的网格资源管理调度解决方案。

参考文献

- Li Chunlin, Lu Zhengding, Li Layuan. Apply Market Mechanism to Agent-based Grid Resource Management[J]. International Journal of Software Engineering & Knowledge Engineering, World Scientific Publishing, 2003, 12(3): 327-340.
- 邹 锐, 李之棠, 杨德志. 网络计算环境中经济模型研究[J]. 计算机应用研究, 2005, 22(2): 82-84
- Buyya R, Abramson D, Giddy J. Nimrod-G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid[C]//Proc. of the 4th International Conf. on HPC in Asia-Pacific Region. 2000
- Braun T D, Siegel H J, Beck N. A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems[J]. Journal of Parallel and Distributed Computing, 2001, 61(6): 810-837.

对该物体定位。该方法代价低，不需要对移动终端和服务提供端这两个通信装置做大的改动。并且和其它根据信号强度定位的方法相比，MMCCPA 充分利用了相邻参考点信号强度的综合信息以及位置历史信息等，提高了抗干扰能力和定位的精确度。

参考文献

- Jin Minghui, Liao Yunbin, Liao Huichun, et al. 802.11-based Positioning System for Context Aware Applications[C]//Proc. of IEEE Global Telecommunications Conference. 2003: 929-933.
- Bahl P, Balachandran A, Padmanabhan V N. Enhancements to the RADAR User Location and Tracking System[R]. Microsoft Co., 2000-02.
- Mitchell S. Context-aware Multimedia Computing in the Intelligent Hospital[C]//Proc. of the 9th ACM SIGOPS European Workshop. ACM Press, 2000: 13-18.
- Bahl P, Padmanabhan V N. RADAR: An RF-based In-building User Location and Tracking System[C]//Proc. of IEEE Infocom'00. 2000.
- Garmin Corporation. About GPS[EB/OL]. 2001. <http://www.garmin.com/aboutGPS/>.
- Perry R, Vaddiraju A, Buckley K. Multitarget List Viterbi Tracking Algorithm[C]//Conference Record of the 32nd Asilomar Conference on Signals, Systems & Computers. 1998: 436-440.