

一种规避风险的网格信任调度模型

姚 军,何廷年,李 勇,曲伟丽,马满福

(西北师范大学 数学与信息科学学院, 兰州 730070)

(mamanfu@nwnu.edu.cn)

摘 要: 网格资源调度中,调度的可靠性不仅依赖资源本身,还受制于调度的环境。针对可靠性问题,将调度中环境带来的影响归纳为风险,以网络拥塞、病毒警报和系统可靠性为参数,给出了风险模型;提出了支撑该模型的体系结构;在信任度基础上,给出了规避风险的资源调度算法。实验表明,所提出的调度算法尽可能规避了存在的风险,大幅度提高了调度的成功率,实现了调度的优化。

关键词: 网格调度;风险模型;算法;体系结构

中图分类号: TP302 **文献标志码:** A

Grid resource scheduling model based on risk evasion

YAO Jun, HE Ting-nian, LI Yong, QU Wei-li, MA Man-fu

(College of Mathematics and Information Science, Northwest Normal University, Lanzhou Gansu 730070, China)

Abstract: Scheduling reliability not only depends on resources but also environment of correlation task and resource in grid. For the reliability problem, the author considered the influence of environment as risk, and proposed a risk model based on network congestion, virus alarm signal and system reliability and its risk infrastructure. The risk evasion scheduling algorithm of the model was put forward based on trust degree. Experiments show that the risk evasion scheduling algorithm is efficient in reducing risk and significantly improves the success rate of the scheduling.

Key words: grid scheduling; risk model; algorithm; architecture

当前网络安全和信任研究已经较为充分地考虑了基于用户身份、资源行为能力等方面对于调度安全的影响和采取的措施。但在网格环境下,一次调度是否能顺利完成不是一个独立依赖于单个资源的问题,是上下文相关的,即和环境密切相关。但针对上下文的研究在网格中目前还较少,也不存在一个规范的上下文标准。基于当前信任研究的基础^[1-3],本文进一步考虑调度中环境相关因素对成功调度产生的影响,包括病毒入侵、网络拥塞和系统可靠性等,从而尽可能规避调度中可能出现的风险,提高调度的成功率,维护调度的可靠性。

1 信任模型定义和描述

1.1 信任模型构成

经典的网格信任模型包括两部分:身份信任和行为信任^[4]。身份信任对实体对应的身份进行确认,主要采用 PKI、Kerberos 以及第三方的 CA 等技术进行身份的安全认证,同时包括用户的单点登录问题。行为信任则在实体之间发生调度关系时,依据对方在调度中体现出来的行为能力给对方做出评价,是对主体声称的能力可靠性的确认。在实现上,行为信任又可分为主观信任评价和客观信任评价。作为主观信任评价的典型方法,文献[5]将行为信任分为直接信任和间接信任,通过计算直接信任和间接信任的加权求和来获得信任的评价值 $T(A, B)$ 。文献[6]通过域的基础设施,依据每个实体调度的历史记录,进行客观的信任评价,这种评价结果称为声誉 RC 。

由上述可见,当前的信任模型是一个典型的二元组

$P(\text{Security}, \text{Trust})$, 即:

$$\left\{ \begin{array}{l} \text{PKI(X.509 V3/V2, CA/RA 操作协议, CA 管理协议,} \\ \text{CA 政策制定)} \\ T(A, B) = \alpha \mid G(\alpha^+ \rightarrow T(A, B) \geq \alpha) \wedge G(\alpha^- \rightarrow \\ T(A, B) < \alpha), \alpha \in [0, 1], \\ \text{或 } RC(x, y) = \alpha \cdot RC_{\text{old}}(x, y) + \\ \beta \cdot RC_{\text{new}}(x, y)^{[6]} \end{array} \right.$$

显然,上述模型仍然是一个理想模型,仅仅考虑调度中实体的自身因素,而没有考虑实体所在环境中外部因素对调度的影响。调度的外部因素是一个复杂的系统,称为上下文,通常用集合表示为:

$$CS = \{CT_1, CT_2, \dots, CT_n\}$$

其中 CT_i 是包含在上下文中的各种因素,如时间、位置等一些具体的环境信息,也可能是抽象的概念。显然,上下文对调度本身有很大的影响。一个完善的信任模型应当包含上下文,由此,信任模型是一个三元组 $P(\text{Security}, \text{Trust}, \text{Context})$:

$$\left\{ \begin{array}{l} \text{PKI(X.509 V3/V2, CA/RA 操作协议, CA 管理协议,} \\ \text{CA 政策制定)} \\ T(A, B) = \alpha \mid G(\alpha^+ \rightarrow T(A, B) \geq \alpha) \wedge G(\alpha^- \rightarrow \\ T(A, B) < \alpha), \alpha \in [0, 1], \\ \text{或 } RC(x, y) = \alpha \cdot RC_{\text{old}}(x, y) + \\ \beta \cdot RC_{\text{new}}(x, y) \\ CS = \{CT_1, CT_2, \dots, CT_n\} \end{array} \right.$$

1.2 风险模型

上下文在网格信任领域目前缺乏统一和规范的定义,本文根据当前网络管理和网格监测提供的支持,将上下文具体

收稿日期:2008-12-04。 基金项目:教育部科学技术研究重点项目(208148),甘肃省科技攻关项目(2GS064-A52-035-03)。

作者简介:姚军(1967-),女,甘肃白银人,讲师,主要研究方向:计算机系统结构、网络计算; 何廷年(1978-),男,甘肃酒泉人,讲师,主要研究方向:网络计算; 李勇(1978-),男,甘肃庆阳人,讲师,主要研究方向:网络计算; 曲伟丽(1984-),女,甘肃平凉人,硕士,主要研究方向:网络计算; 马满福(1968-),男,甘肃甘谷人,副教授,博士,主要研究方向:计算机系统结构、移动计算。

定义为几个典型的参数:网络拥塞、病毒侵害和系统稳定性。由此,上下文表达式为: $CS = \{CT_1, CT_2, CT_3\}$ 。其中: CT_1 表示网络拥塞等级; CT_2 为病毒侵害等级或者系统受到安全威胁下的安全等级; CT_3 为系统可靠性参数。由于上述定义是一个不全面的上下文,同时参数均和调度的可靠性相关,这里称为风险模型。

在网络监测中,有多种方法用来描述拥塞程度和安全等级。如拥塞可用包平均延迟和丢包率来衡量,而安全可采用 TCSEC 准则,也可采用自定义准则。同样,网络涵盖了不同的网络,而这些网络有不同的监测和控制模型。鉴于此,将这些不同的标准转换到同一个等级标准上,并进行归一化处理,使其值在区间 $[0, 1]$ 按照严重程度单调递增。在模型中,假设网格域和网络管理域一一对应。

设 CT_1 为 $[0, 1]$ 之间连续变化的量,拥塞控制的监测由网络管理域中对应设施完成。 $CT_1^{t,r}$ 表示任务 t 到资源 r 的拥塞程度,经过的域依次为 D_1, D_2, \dots, D_m , 对应拥塞监测的结果为 $CT_1^1, CT_1^2, \dots, CT_1^m$, 由于拥塞不具有传递性和和多域之间的累计效应,则 $CT_1^{t,r}$ 按照取其中最大值得到,即 $CT_1^{t,r} = \max \{CT_1^1, CT_1^2, \dots, CT_1^m\}$ 。

CT_2 表示病毒侵害的状态,报警来自资源所在的系统。设 CT_2 定义的侵害级别依次为 G_1, G_2, \dots, G_l 。 CT_2^r 为资源 r 所在系统当前的病毒侵害状态,按照定义,当侵害级别大于 $G_h, 1 \leq h \leq l$ 时,系统处于高风险状态,此时不宜调度其所提供的资源。 CT_3^r 为资源 r 所在系统的可靠性,表现为一个成功调度资源的概率,由网络监测系统提供。风险模型可定义为一个分段函数:

$$Risk(t, r) = \begin{cases} 1, & CT_2^r > G_h \\ \alpha \cdot CT_1^{t,r} + \beta \cdot CT_2^r + \gamma \cdot (1 - CT_3^r), & CT_2^r \leq G_h \end{cases}$$

其中, $0 \leq \alpha, \beta, \gamma \leq 1$, 且 $\alpha + \beta + \gamma = 1$ 。显然,风险体现为 $[0, 1]$ 内的一个量,最大值为 1, 表示资源所在系统病毒侵害级别超过临界值,处于高风险状态,不能进行该系统资源的调度。 α, β, γ 为主观值,一般地按照调度中的危险性,存在 $\beta > \alpha > \gamma$ 。

2 体系结构

支撑风险评估的网格监测结构如图 1 所示。图 1 中,假定网络管理域和网络管理域完全一致。在每个管理域内,部署网络监测系统,以最终获取系统可靠性参数;同时部署网络监测,实现网络运行中拥塞的检测;在每个资源所在的系统中,安装了病毒检测软件,实现病毒报警和防范。上述监测信息按照一定的时间周期定时在网格中的监测服务器汇集,通过 Agent 实现多个域之间的数据交换。显然,通过本域的数据汇集和 Agent 的交换,在每个域服务器上,存放了一个完全版本的数据快照,供本地任务选择资源时查询。数据处理模型如图 2 所示。

其中数据监测和汇集在域内完成,处理主要是对本地数据按照模型要求进行级别对应和归一化,使之在全网格具有统一的含义;交换实现各域间的监测数据互换,由 Agent 完成;发布则在本域发布全局数据。需要说明的是,Agent 从网络监测处获取数据需要调用相应的 API,并经过授权。

3 风险规避调度算法

设满足(符合 QoS 需求)任务 t 的所候选资源集合 $S_k =$

$\{r_1, r_2, \dots, r_l\}$, 在 S_k 中, 逐个计算 t 针对每个资源的信任度 $T(t, r_i)$ 和调度每个资源的风险 $Risk(t, r_i)$ ($i = 1, \dots, l$)。如果出现 $Risk(t, r_i) = 1$ 的资源, 则从 S_k 中直接删除。之后, 计算信任度和风险, 按照 $Risk(t, r_i)/T(t, r_i)$ 从小到大的顺序排列 S_k , 再按顺序协商并调度资源, 算法描述如下:

```
task-to-resource ( $t, S_k$ )
{
     $S = \emptyset$ ; Scheduler = false;
    For every  $r_i \in S_k$  do
    {
         $T(t, r_i) = \alpha$ ; //计算任务  $t$  对资源  $r_i$  的信任度
         $Risk(t, r_i) =$ 
        {
            1,  $CT_2^r > G_h$ 
             $\alpha \cdot CT_1^{t,r} + \beta \cdot CT_2^r + \gamma \cdot (1 - CT_3^r)$ ,  $CT_2^r \leq G_h$ 
        } //计算任务  $t$  对资源  $r_i$  的风险
        if ( $Risk(t, r_i) = 1$ ) then delete  $r_i$  from  $S_k$ 
        else insert  $r_i$  to  $S$  as increase by  $Risk(t, r_i)/T(t, r_i)$ 
        //按照风险最小、信任度最高排序
    }
    if  $S = \emptyset$  then abort("no resource is available") else
    //没有满足任务需求的资源
    For  $i = 1$  to  $|S|$  do
    {
        Scheduling  $t$  to  $r_i$  //按  $S$  中的顺序逐个进行协商
        Scheduler = true;
        break;
    }
    if Scheduler = false then
        abort("no resource is available");
        //所有资源协商没达成协议,调度失败
}
```

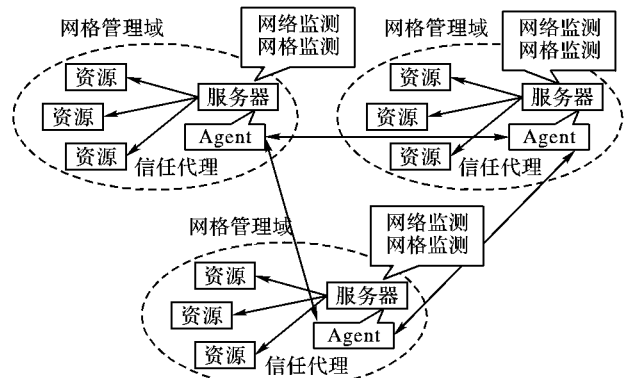


图 1 网络监测结构



图 2 风险数据处理模型

4 实验评价和分析

4.1 实验设计

实验以 Nimrod-G 提供的 GRACE (GRid Architecture for Computational Economy)^[7] 为平台实现了原型系统, 共包括 40 个节点, 每个节点提供多种类型的多个资源, 节点既是资源的提供者, 也是任务的宿主者。所有节点分为 4 个网络管理域, 在每个管理域内除部署了资源管理系统外, 按照文献[8]提供的方法进行资源信任度评价。同时, 实现了 Agent, 按照体系结构进行部署, 实现域间数据的传递。

实验中需要的监测参数采用随机数生成的方式提供。按照仿真需要, 拥塞数据以域为单位提供, 与域标识绑定; 病毒报警数据和系统可靠性数据与资源绑定。产生数据均规约在

$[0,1]$,为了更真实地模仿实际系统,降低病毒报警频率,病毒数据在 $[0,0.7]$ 内视为无报警。报警分为四级:无报警 $[0,0.7]$ 、初级警报 $(0.7,0.8]$ 、中级警报 $(0.8,0.95]$ 和高级警报 $(0.95,1]$,当发生高级警报时,不再调度该系统提供的资源。相应地,在资源所在的系统中,以系统可靠性参数为概率提供服务的成功率。任务的平均执行周期定义为 100 ms ,风险数据的刷新周期为 400 ms 。在风险计算中, α, β, γ 分别取值为 $0.3, 0.5, 0.2$ 。

在比较算法上,将文献[7]在经济模型下提出的非最小化(None Minimisation)算法加入信任度,改造为信任度调度,具体为:

- 1) 按照不超出时间和预算的约束来选择资源;
- 2) 按照信任递减的顺序排序,按顺序调度资源;
- 3) 在信任度相等时,对于费用更低的资源,按照任务完成时间的反比分配任务;
- 4) 重复上述步骤,直到任务得到资源或失败。

4.2 结果及分析

在实验中,采用符合泊松分布的随机函数进行调度和资源的登录以及退出。为了便于采集实验数据,控制调度在低频率范围进行。在系统运行较长时间,获得了较为稳定的状态,即信任模型对资源进行了多次评价,且 Agent 之间完成了多次刷新后进行数据采集。

实验针对两种算法在任务的平均完成时间、吞吐量、资源调度成功率和资源利用率上取得了实验结果,如表1所示。

表1 实验结果

算法	吞吐量/次	平均完成时间/ms	成功率/%	资源利用率%
信任度调度	9675	117	55.59	9.35
风险规避调度	9834	102	83.56	9.12

信任度和风险规避算法比较,在系统吞吐量上,分别完成的调度次数为9675和9834,差异不到1.61%,无明显变化。资源利用率也显示出相似的特征。在任务平均完成时间上,风险规避算法比信任度算法提高了12.8%,其原因在于,一方面风险规避算法考虑了不同域中通信的拥塞,选择通信状况良好的资源调度;另一方面,选择资源所在的系统可靠性较高,完成情况也会更好,而信任度算法以信任度为唯一的依据,没考虑上述因素。从调度的成功率上看,信任度算法为55.59%,而风险规避算法达到了83.56%,相比提高了29.97%。这是因为从通信、资源的执行环境以及资源本身的可靠性方面,风险规避算法都进行了选择,特别是尽可能避免了影响权值比较大的($\beta = 0.5$)病毒报警对任务执行带来的干扰,从而基本保证了一个调度的良好环境。

实验还针对三个风险参数进行了单独的评估。即在四个域 D_1, D_2, D_3, D_4 中,对应不同域,病毒和拥塞参数取值大于0.8,而可靠性小于0.2,实验结果如表2所示。

从表2可以看出,在域 D_1 中,由于拥塞的发生,和其他域比较,资源调度率急剧下降,从平均的10%左右下降到2.8%;在 D_2 中,下降到了1.1%,原因在于风险模型中 $\beta = 0.5$,相对其余参数,权值较大,影响也更明显;在 D_3 中,资源调度率下降到了3.6%,而除此之外的其他域在参数不变的情况下显得比较稳定。可见,风险模型在各自的参数上尽可能地避免了风险的发生,选择了当前网格中可靠性高、通信畅通和病毒危害小的资源和路径,达到了规避风险的目的,实现了调度的优化。

和信任度调度比较,风险数据的采集、交换以及调度均增

加了系统负载。但数据的采集和交换在时间和设施上是独立于调度进行的,不影响调度的执行;在调度中,风险参数处理模型简单、计算复杂度低,不会带来很大负担。从表1的任务平均完成时间上看出,计算的负载被优化选择完全覆盖,而且有所提高。可见,计算负载不影响调度的效率。

表2 不同参数设置及对应结果

网格域	参数设置	资源调度率
D_1	拥塞参数 >0.8 ,其余不变	$D_1 = 2.8\%$, $D_2 = 11.3\%$, $D_3 = 10.6\%$, $D_4 = 10.8\%$
D_2	病毒报警 >0.8 ,其余不变	$D_1 = 10.3\%$, $D_2 = 1.1\%$, $D_3 = 11.1\%$, $D_4 = 10.6\%$
D_3	可靠性 <0.2 ,其余不变	$D_1 = 11\%$, $D_2 = 10.7\%$, $D_3 = 3.6\%$, $D_4 = 11.2\%$

5 结语

本文在当前信任模型的基础上,将调度风险纳入到资源选择过程中。以网络拥塞、病毒警报和系统可靠性为参数,给出了风险模型。由此,资源选择时尽可能避开存在风险的域和系统,选择一个从通信到执行环境良好的资源。实验证明,信任结合风险模型,大幅度提高了调度的成功率,实现了调度的优化。本文所考虑的内容是网格调度上下文中的一部分,对风险也仅仅是一个初步的理解,在将来应当考虑更多可采集的参数、影响调度的因素,并将它们全部纳入进行风险控制,使得调度的成功率进一步提高,规避可能出现的风险,增强系统的可靠性。

参考文献:

- [1] EYMANN T, KÖNIG S, MATROS R. A framework for trust and reputation in grid environments [J]. Grid Computing, 2008, 6(3): 225–237.
- [2] SELVI S T, BALAKRISHNAN P, KUMAR R, et al. Trust based grid scheduling algorithm for commercial grids [C]// International Conference on Computational Intelligence and Multimedia Applications: ICCIMA 2007. Washington, DC: IEEE Computer Society, 2007, 1: 545–558.
- [3] PAPALILLO E, FREISLEBEN B. Towards a flexible trust model for grid environments [C]// International Conference on Grid Services Engineering and Management: GSEM 2004, LNCS 3270. Berlin: Springer-Verlag, 2004: 94–106.
- [4] RUOHOMAA S, KUTVONEN L. Trust management survey [C]// iTrust International Conference, LNCS 3477. Berlin: Springer-Verlag, 2005: 77–92.
- [5] AZZEDIN F, MAHESWARAN M. Evolving and managing trust in grid computing systems [C]// IEEE Canadian Conference on Electrical and Computer Engineering: CCECE'02. Washington, DC: IEEE Press, 2002, 5: 1424–1429.
- [6] 马满福, 吴健, 胡正国, 等. 网格计算资源管理中的信誉度模型 [J], 计算机应用, 2005, 25(1): 61–64.
- [7] BUYYA R, ABRAMSON D, GIDDY J. Nimrod/G: An architecture or a resource management and scheduling system in a global computational grid [C]// Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region: HPC ASIA 2000. Washington, DC: IEEE Computer Society, 2000.
- [8] AZZEDIN F, MAHESWARAN M. Integrating trust into grid resource management systems [C]// The International Association for Computers and Communications. Washington, DC: IEEE Computer Society, 2002: 47–54.