

虚拟计算环境下面向应用的基于信任的资源匹配模型

朱春鸽^{1,2}, 刘欣然^{1,2}, 杨义先¹, 张鸿²

(1. 北京邮电大学 信息安全中心, 北京 100876; 2. 国家计算机网络应急技术处理协调中心, 北京 100029)

摘要:为了解决虚拟计算环境中的资源合理调度问题,提出了一种基于信任的资源匹配模型——“资源滑动窗口”模型。首先对资源的静态属性进行分类,然后依据基于时间窗口的贝叶斯信任模型对资源节点进行评价,同时考虑资源的负载,动态划分其实时性能。最后综合评估静态和动态属性,确定调度资源分配。该模型为不同任务和属性的资源调度策略奠定了基础,实现了“合适的资源服务于合适的任务”的目的。仿真实验表明所提的模型相对传统的调度算法,具有更高的系统任务执行成功率和资源利用率。

关键词:虚拟计算环境;信任;资源滑动窗口;资源匹配模型

中图分类号:TP393

文献标识码:A

文章编号:1000-436X(2013)09-0024-09

Application-oriented resource matching model based on trust for internet-based virtual computing environment

ZHU Chun-ge^{1,2}, LIU Xin-ran^{1,2}, YANG Yi-xian¹, ZHANG Hong²

(1. Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China;

2. National Computer Network Emergency Response Technical Team/Coordination Center, Beijing 100029, China)

Abstract: In order to solve the problem of reasonable resource scheduling in the virtual computing environment (iVCE), a resource matching model, based on trust “resource sliding window” model, was proposed. Firstly, the resources were classified in advance by their static properties. Then the resource nodes were evaluated by a trust model based on the time window and Bayesian algorithm, and the evaluated results as well as the resource load were fused to determine the levels of the dynamic real-time performance. Finally, the scheduling resources were allocated according to the comprehensive assessment of the static and dynamic properties of the resource nodes. The proposed model provides a basis of scheduling strategy for different resources with different tasks and attributes, thus achieving the goal of “proper resources’ serving proper tasks”. Simulation results show that, compared the previous scheduling algorithm, the proposed model has better performance in the success rate of task execution and the rate of resource utilization.

Key words: iVCE; trust; resource sliding window; resource matching model

1 引言

云计算的目标是将各种资源以服务的方式通过互联网交付给用户,实现“按需取用”的功能。从技术角度看,其计算平台主要依赖于虚拟计算技术来实现资源的部署、聚合和协同^[1]。虚拟计算环境(iVCE, internet-based virtual computing en-

vironment)是指建立在开放的互联网基础设施之上,通过对分布自治资源的集成和综合利用,为终端用户或应用系统提供和谐、安全、透明的一体化服务的环境^[2]。iVCE中包含海量、异构的自治资源也即自主元素(AE, autonomic element, 以下所称“自主元素”、“资源”、“节点”指同一个概念,不做区分),且自主元素动态变化,这给任务的调度

收稿日期:2013-05-02;修回日期:2013-07-17

基金项目:国家重点基础研究发展计划(“973”计划)基金资助项目(2011CB302605);国家自然科学基金资助项目(61121061)

Foundation Items: The National Basic Research Program of China (973 Program) (2011CB302605); The National Natural Science Foundation of China (61121061)

以及任务与资源的快速有效匹配提出了挑战。

目前,使用最广泛的任务调度算法是 Min-Min 算法和 Max-Min 算法^[3]以及它们的改进算法^[4]。Min-Min 算法是一个简单快速的方法,每次调度总是选择任务完成时间最短的,资源调度最短的任务,所以非常高效,然而对于长任务总会被延迟执行,而且有些资源总是被调度,而有些资源总是空闲,导致负载的不平衡。Max-Min 算法类似 Min-Min 算法,也是迭代计算任务最小完成时间,不过它先调度大任务,追求任务吞吐率,但该算法没有考虑用户对任务提出的服务质量要求。文献[4]提出的改进算法,进一步降低了执行时间跨度和降低负载,仍然没考虑任务的动态特性,这会引发任务的盲目调度。虽然所提调度算法开始主要用于网格服务中^[5],其他的分布式系统,如 P2P 网络^[6]和云计算环境^[7]中也使用了类似的调度算法,这些工作存在的缺陷仍然是任务和资源没有有效和动态的匹配,任务执行成功率和资源利用率有待进一步改进。

目前,任务调度中的资源匹配问题已经引起了一些研究者的注意^[8-11]。文献[8]提出了一种基于推荐机制的网格资源匹配算法,用基于奇异值分解的协同过滤算法匹配用户行为和资源使用频度的相关性,给出推荐资源集合。文献[9]提出了大尺度计算平台上的任务调度器,启发机制设置任务和机器之间的匹配值,设置一系列的调度计划,达到满足任务的服务质量要求,并优化资源的使用,但该算法没有考虑资源本身的动态属性,会引起一定程度的盲目调度。文献[10]和文献[11]分别提出了异构网络环境下基于遗传算法的任务匹配方法和云计算环境下可信的资源调度算法,但对资源和任务方面的行为动态属性和历史属性有欠考虑,匹配的精确和顽健性有待提高。

此外,以上资源匹配算法中都没有考虑应用类型,也即具体任务对资源需求的不同而对资源进行分类。文献[12]为了降低分组丢失率,将任务分为紧急任务、高优先级任务和普通任务,调度资源优先执行特殊任务,但其没对资源分类,负载仍存在不均衡。文献[13]根据资源的静态属性对资源进行模糊分类,给出了分类依据,但没有对资源根据实际情况进行动态更新。总之,针对资源的分类、匹配、合理调度仍然是个开放的课题。

本文重点考察了一类多应用的场景。虚拟计算环境中,资源和任务是异构的,用户的需求多样化,

需调度的任务可能是:1)具有条件约束的,比如某次调度需要 2 MB 的带宽、2 GB 的处理器能力;2)用户的不同资源偏好,比如用户可能更喜欢通过移动网络下载资源,用户更偏好存储优先型资源等;3)优先级别的差异,比如重要通信数据流的优先级要高等。在这样应用场景下,资源的聚合和匹配必须具有智能化的特性,才能实现优势资源的合理部署。

基于上述场景,本文提出了一种面向应用的基于信任的资源匹配模型——“资源滑动窗口”模型,对计算资源的静态属性预先进行分类,依据信任模型对资源的历史表现进行评价,同时综合考虑其负载情况,动态地划分资源的实时性能等级,为实现对不同任务和属性进行不同等级资源配置的调度策略提供基础,从而实现了“合适的资源服务于合适的任务”的目的。

2 贝叶斯信任模型

2.1 信任值求解

信任是一个非常复杂的概念,目前国内外对于信任还没有一个精确的、可广泛接受的定义。本文借鉴社会学中人际关系模型对信任的刻画,给出 iVCE 下的信任定义如下^[14]。

定义 1 信任是对一个实体(如 iVCE 中的自主元素)身份和行为的可信度的评估,包括在某些指定内容上(如任务执行成功率)对该实体过去行为的观察以及其他实体对该实体的推荐信息;信任是一个主观概念,取决于经验,用信任值表示信任等级的高低,信任值随实体的行为而动态变化,信任程度随时间递减。

为了模型的简捷并不失一般性,假定本文所考察的自主元素均已通过身份认证,即身份是可信的。由此,自主元素之间的信任度可简单刻画为目标节点的动态行为的度量,本文用基于历史行为的任务执行成功或失败的概率关系表示。

贝叶斯方法建立在主观概率的基础上。主观概率是一种见解,是合理信念的测度,它是某人对特定事件会发生的可能性的信念(或看法)的度量,即此人相信或认为事件将会发生的可能性的概率。文献[15]在网格环境中对节点的直接信任值采用了贝叶斯方法,依据先验知识对后验概率进行估算,这与本文 iVCE 计算环境下对自主元素信任值的推算场景类似,故在此本文借鉴了其方法,并根据贝

叶斯方法就上述的信任度值的获取进行分析。

本文用 ae_i 表示第 i 个自主元素，用 q 来表示 $iVCE$ 中 2 个自主元素 ae_i 和 ae_j 交互成功的概率，用 tv_i 表示第 i 个自主元素的信任值。对于 2 个自主元素间交互的信任度，可以用贝叶斯方法直接估计其值。

设 $iVCE$ 中 2 个自主元素为 ae_i 和 ae_j ，可用二项事件（成功/失败）来描述它们之间交互的结果，当其发生 n 次交互后，成功的次数为 u ，失败的次数为 v ，则可将信任度 $\$$ 定义为第 $n+1$ 次交互成功的概率。则自主元素 ae_i 和 ae_j 之间交互成功的后验概率服从 Beta 分布，其密度函数为

$$Beta(q | u, v) = \frac{G(u+v+2)}{G(u+1)G(v+1)} q^u (1-q)^v \quad (1)$$

$\$$ 为该密度函数的数学期望，定义为

$$\$ = E(Beta(q | u+1, v+1)) = \frac{u+1}{u+v+2} \quad (2)$$

其中， $0 < q < 1$ ， $u > 0, v > 0$ ， u 和 v 分别为自主元素 ae_i 执行任务成功和失败的次数。则第 i 个自主元素的信任值 tv_i 可定义为 $tv_i = \$$ 。文献[15]已对其正确性做了详细证明，在此不再累述。

由式 (2) 可知，信任度值的大小与目标节点以往提供服务的成功概率以及总的交互次数有关。同时，信任度也表达了目标节点在网络中提供可靠资源和稳定交互的能力，因此本文将信任度作为自主元素动态性能的一个重要参数。

2.2 时间衰减对信任值的修正

根据社会学中的人际关系模型可知，信任值有时间相关性，节点近期的行为应该比早期的更有说服力^[16]。本文采用时间窗口的概念，将自主元素的历史行为按时间窗口进行分组，并根据时间先后赋予不同的权值，再根据合成算法求得最终的信任值。信任值的更新根据时间窗口向前滑动，各时间窗口内的信任信息不停更新，进而自主元素的信任值随之更新。

在此，设 f_{ud} 为更新频率， tw_t 为第 t 个时间窗口， tv_{it} 表示第 i 个自主元素在第 t 个时间窗口内的信任值， w_t 为 tv_{it} 的权重。根据具体的用户任务对时效性的要求， f_{ud} 可以按照多个粒度进行设定，比如可设定为 1 min、1 h、1 天、1 个月等；同理，时间窗口的大小也可以按照分钟、小时、天、月等

多个粒度进行设定。而在实际应用中，以天为单位往往较为合理，既能及时反映信任度随时间的变化，又能使计算较为高效合理。

时间窗口如图 1 所示， NT 表示当前的这一天， tw_1 是第一个时间窗口，为最近 5 天， tw_2 是第二个时间窗口，为再早的 1 周， tw_3 是第三个时间窗口，为再早的 2 周。当前的信任值只考虑当前时间的前 26 天的信任信息，再早些的数据不予考虑。每过一天，每个时间窗口往前滑动，各个窗口内的信任值 tv_{it} 根据式 (2) 重新计算，则第 i 个自主元素当前的信任值最终修正为

$$tv_i = \sum_{t=1}^k w_t \cdot tv_{it} \quad (3)$$

其中， $\sum_{t=1}^k w_t = 1$ ， $k \geq 1$ 。

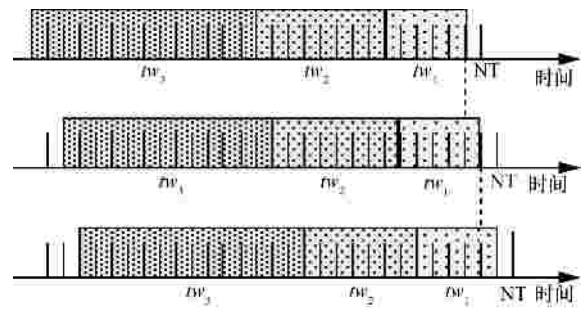


图 1 时间窗口与信任值的更新

3 资源匹配模型

3.1 资源的分类

为了更好地利用 $iVCE$ 中的资源，需要预先对资源的属性有充分的了解，并根据其属性及性能特点进行科学合理的分类。

目前 $iVCE$ 平台可以获取到的资源属性信息分为静态信息 (SI, static information) 和动态信息 (DI, dynamic information) 2 部分。其中静态信息包括该资源节点的 CPU 类型、CPU 的个数、内存大小、磁盘容量、操作系统类型、该节点所在地理位置 (精确至城市)、该节点所接入的 ISP 类型、接入方式 (目前支持 ADSL 动态接入和固网接入 2 种方式)、接入的网络带宽等；动态信息包括 CPU 利用率、内存利用率、磁盘利用率、网络的实际带宽、节点稳定性 (该节点信任值) 等。由此可见，资源节点的静态信息反映了该资源节点的固有属性，动态信息反映了该资源节点的当前状态。

静态信息中的各项参数构成一个一维向量 SI ，如表 1 所示。

向量名称	静态向量
CPU 类型	ct
CPU 数量	cn
内存大小	ms
硬盘容量	dc
OS 类型	ot
地域 (城市)	lc
运营商类型	it
接入方式	at
接入宽带	ad

在调度时，首先依据 SI 中的各向量参数值对资源节点进行分类，满足任务对 SI 要求的资源节点才可以进入到“资源窗口”中。比如某一个对地理位置及接入 ISP 有特定要求的任务，假定为北京移动，则只有满足 $lc = \text{“北京”}$ 、 $it = \text{“移动”}$ 的资源节点才能进入到“资源窗口”，这样可以有效提高任务调度的精准性。

同样，动态信息中的各项参数也构成了一个一维向量 DI ，如表 2 所示。

向量名称	动态向量
CPU 利用率	cu
内存利用率	mu
硬盘利用率	du
带宽	bw
稳定性 (信任值)	tv

动态信息 DI 中的各项参数都是实时变化的，实时反映资源节点的状态。其中节点的稳定性，即采用了在第 2 节中得出的信任值，依据该节点的历史表现给出一个信任值 tv ，以此来标示该节点的稳定性。 bw 是对实时带宽进行归一化之后的值，在此，以 100 MB 为基准。向量 DI 中的各项参数的取值区间均为 $[0, 1]$ 。

动态信息 DI 反映了该节点的实时性能，也即该节点当前的负载情况及其稳定性，在此依据 DI 的各项参数计算出每个节点的实时性能评价价值，首先计算负载，记为 $Load$ ，如式 (4) 所示。

$$load_i = a(1 - cu_i) + b(1 - mu_i) + c(1 - du_i) + dbw_i \quad (4)$$

其中， $a + b + c + d = 1$ 。节点的实时性能评价价值记为 pv ，如式 (5) 所示，该值也动态变化。

$$pv_i = e_1 load_i + e_2 tv_i \quad (5)$$

其中， $e_1 + e_2 = 1$ ， e_1 和 e_2 分别表示资源负载和资源信任值的权重。

由式 (5) 计算出第 i 个资源节点当前的性能评价价值， $pv \in [0, 1]$ 。其各项参数的具体权重值可根据具体任务对不同属性的要求进行调整。比如在计算负载时， a 、 b 、 c 、 d 为对应计算型任务、存储型任务、通信型任务的不同权重，如果该次任务要求的负载侧重点不同，可适当调整其对应某个参数的权重值。在计算综合性能评价价值时，信任值(对应资源节点的稳定性)和负载又可以对应不同的用户应用需求或用户偏好，也是通过调整权重 e_1 和 e_2 实现的。 pv 的值越大，表明该节点当前的符合特定应用的性能越好。该资源节点的性能等级，记为 pb ，依据 pv 的值计算得来。根据仿真系统经验值以及前述工作成果^[17]，采用分段函数来界定资源节点的性能等级，如式 (6) 所示。

$$pb = \begin{cases} 1, & pv \in [0.8, 1] \\ 2, & pv \in [0.5, 0.8) \\ 3, & pv \in [0.3, 0.5) \\ 4, & pv \in [0, 0.3) \end{cases} \quad (6)$$

3.2 “资源滑动窗口”模型

定义 2 资源滑动窗口是一组资源集合 U ，第 i 个资源(自主元素) $ae_i \in U$ ($i \in [1, n]$ ， n 为资源数目)的条件是 $rw_i = 1$ 。 rw_i 表示该资源是否在“资源滑动窗口”的状态， $rw_i = 1$ 表示该资源在可供调度的窗口中；反之， $rw_i = 0$ 表示该资源不可供调度。

rw_i 可以由以下方式确定：

$$rw_i = sm_i lm_i \quad (7)$$

其中， sm_i 表示满足向量 SI 的基本匹配要求的参数。 sm_i 定义为一个布尔值，即对于 $\forall sm_i, i \in [1, n]$ ，当满足向量 SI 的匹配要求时， $sm_i = 1$ ；否则 $sm_i = 0$ 。 lm_i 表示满足动态性能等级要求的参数。 lm_i 也是一个布尔值，即对于 $\forall lm_i, i \in [1, n]$ ，当满足式 (8) 时， $lm_i = 1$ ；否则 $lm_i = 0$ 。

$$pb_i \in [l_{\min}, l_{\max}], (l_{\min}, l_{\max} \in \{1, 2, 3, 4\}) \quad (8)$$

其中， l_{\min} ， l_{\max} 分别表示某次资源调度所要求的性

能等级的最大值和最小值。

从式(7)可以看出,当 sm_i 和 lm_i 的值同时为真时,也即 $sm_i=1$,且 $lm_i=1$ 时,该资源才进入“资源滑动窗口”,才能被调度。

3.3 模型分析

“资源滑动窗口”模型有 2 个显著的特点。首先,“窗口”的大小可动态“滑动”。该模型可以根据任务对资源多少的需要动态调整对资源性能等级 pb 值的选取来确定。由此,可以很好地对资源需求波动比较大的任务提供良好的弹性支撑;其次,“窗口”中的资源动态“滑动”。资源性能等级值 pb 的获取是由诸多实时性的数据计算得来,并且资源的稳定性 tv 的值还参考了节点的历史表现,因此,资源性能等级 pb 值的应用可以保证“优质资源”的及时更新和准确性。

从功能角度来看,所提模型满足多应用的功能。具有条件约束的调度任务,其条件的设置可以通过资源的静态属性筛选相应的资源到滑动窗口中。资源偏好型的应用可以通过设置动态参数中的权重向量提高对应的资源(例如计算型资源)的调度等级,实现优先调度。优先级资源的调度应用场景,如重要性的通信数据可以设置很高的信任值,在同等条件下优先得到调度。另外,信任作为节点行为动态的反映,使该算法具有一定的智能性。

从性能上看,该模型相对于传统的单纯地以追求最小完成时间的 Min-Min 算法和追求任务吞吐率的 Max-Min 算法,在性能上有了很大的提高。所提模型按需调度,避免了调度的盲目性,资源匹配精确性很高,任务执行成功率有所提高。另外,新模型实现了“合适的资源服务于合适的任务”,避免了负载不均匀的现象,提高了资源利用率。任务执行时间上,新模型提前对资源分类,从式(3)式(6)式(7)可以看出计算量大概需要 $2k$ 次乘法即可,相对于以往的调度采用迭代运算执行时间要小。另外通过中心化的资源管理调度,可进一步缩小任务执行时间。

3.4 基于“资源滑动窗口”模型的 iVCE 基本框架

本小节给出了基于“资源滑动窗口”模型的 iVCE 基本框架,如图 2 所示。

图 2 中,最底层为虚拟资源层,第二层分别为任务层和资源管理层,最顶层是应用层。其中,资

源管理层中,“资源滑动窗口”模型给出最符合任务需求的资源列表。

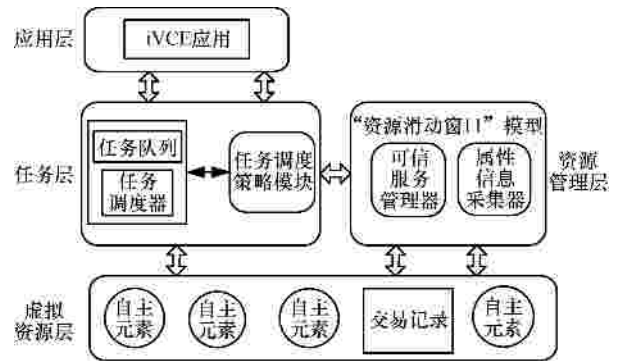


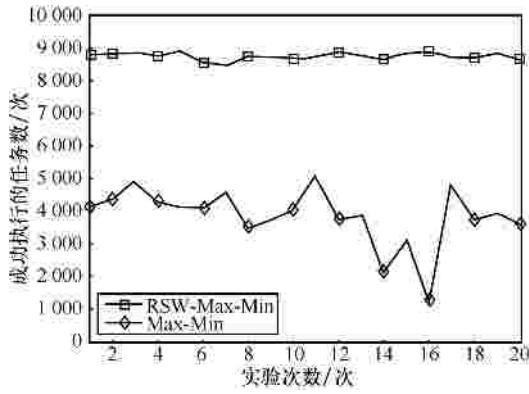
图 2 基于“资源滑动窗口”模型的 iVCE 基本框架

在基于“资源滑动窗口”模型的 iVCE 基本框架下,一个 iVCE 应用用户提交并执行一个任务的过程如下:1) 将自己的任务提交到任务队列中;2) 任务调度器从任务队列中依次取出任务,同时与任务调度策略模块进行交互;3) 属性信息采集器采集到自主元素的静态及动态信息,可信服务管理器通过分析交易记录获得资源的信任信息,均将之传递给“资源滑动窗口”模型,由“资源滑动窗口”模型计算出最符合任务需求的可供调度的资源列表;4) 任务调度策略模块与“资源滑动窗口”模型交互得到可供调度的资源列表;5) 任务调度策略模块将适合该任务的最优资源信息传递给任务调度器,由任务调度器将该任务分配到选中的资源上去执行。

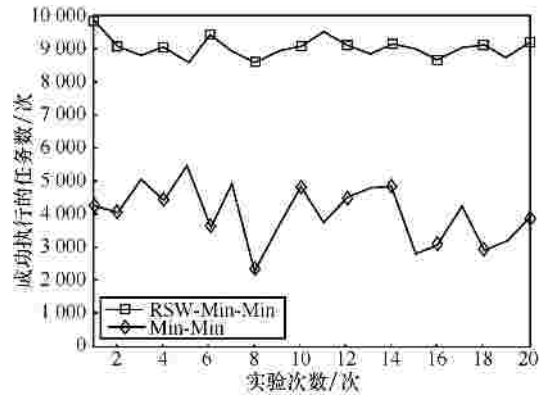
4 仿真实验与分析

4.1 实验环境设置

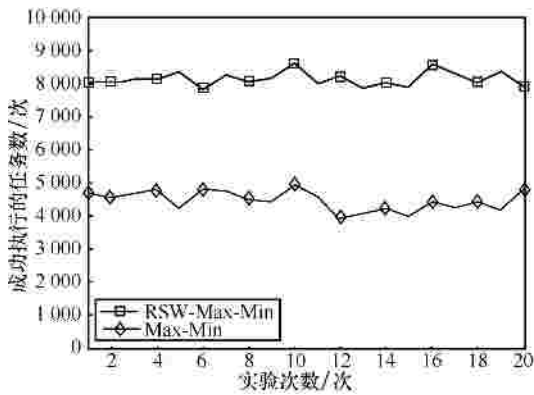
本文依据“资源滑动窗口”模型对传统的 Min-Min 算法和 Max-Min 算法进行了优化,优化后的算法分别为 RSW-Min-Min 和 RSW-Max-Min。为了评估优化后的调度算法的性能,本文在多应用需求下对所提模型进行了对比仿真实验。首先设置了 3 个仿真参数:1) 任务执行成功率,定义为成功执行调度的次数相对于总调度任务次数的比例,该参数可以反映所提模型调度的精确性,资源和任务匹配程度越高,其成功执行的概率越大;2) 系统资源利用率,定义为执行任务的节点的数目占总的资源节点的比例,该参数反映了资源的负载的均衡问题;3) 完成时间,定义为多次实验次数下,任务平均执行完成时间,反映了模型的资源调度的效率,完成时间越短越好。



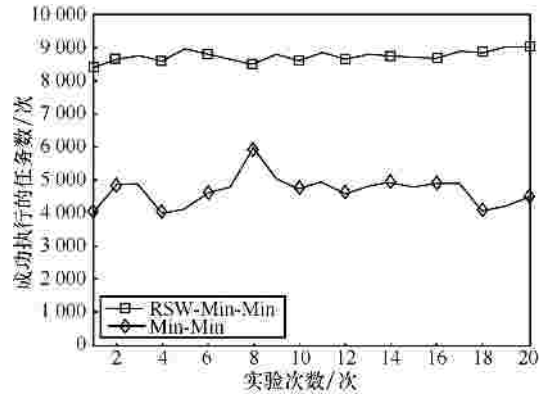
(a) 应用需求为(2, 2)时与 Max-Min 算法的执行成功率比较



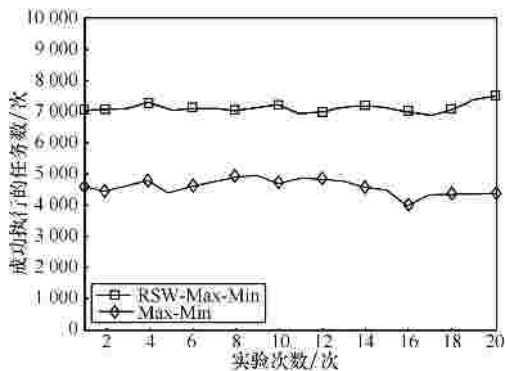
(b) 应用需求为(2, 2)时与 Min-Min 算法的执行成功率比较



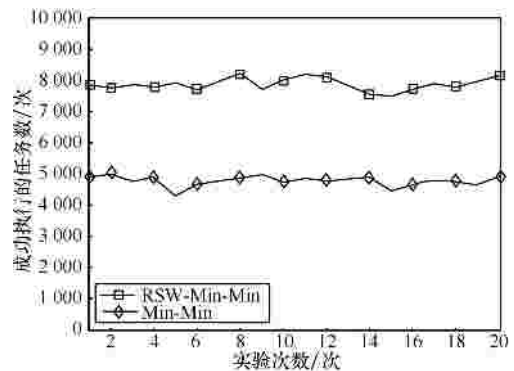
(c) 应用需求为(10, 4)时与 Max-Min 算法的执行成功率比较



(d) 应用需求为(10, 4)时与 Min-Min 算法的执行成功率比较



(e) 应用需求为(20, 5)时与 Max-Min 算法的执行成功率比较



(f) 应用需求为(20, 5)时与 Min-Min 算法的执行成功率比较

图 3 改进的算法与传统 Max-Min 和 Min-Min 算法在任务执行成功率上的比较

为了简化模型并更直观地说明多应用需求带来的差异性，在此设定资源的静态属性中仅对节点接入的地理位置和 ISP 的类型数量有不同的应用需求。本文使用云计算仿真软件 CloudSim 近似模拟云平台上的虚拟计算资源的任务调度。参考了文献 [12,13] 中设置的仿真次数，并不失一般性，本文设计了 1 000 个计算节点计算 10 000 个任务的几个不同场景，每种场景实验 20 次。每个节点的可信度

参数 t_V 以及 CPU 性能参数随机生成。

场景 1 节点地理位置要求数目为 2，ISP 的类型要求数量是 2(以下简称(2,2))。在这种情况下，将 1 000 个节点平均分成 4 组，每组 250 个节点，代表一个地域的一种类型 ISP 接入的节点。

场景 2 节点地理位置要求数目为 10，ISP 的类型要求数量是 4(以下简称(10,4))。在这种情况下，将 1 000 个节点平均分成 40 组，每组 25 个节点，

代表一个地域的一种类型 ISP 接入节点。

场景 3 节点地理位置要求数目为 20, ISP 的类型要求数量是 5(以下简称(20, 5))。在这种情况下, 将 1 000 个节点平均分成 100 组, 每组 10 个节点, 代表一个地域的一种类型 ISP 接入节点。

4.2 实验 1: 任务执行成功率比较

图 3 表示了当应用需求分别为(2,2)、(10,4)和(20,5)时, 改进的优化算法与传统 Min-Min 算法和 Max-Min 算法在任务执行成功率方面的比较。

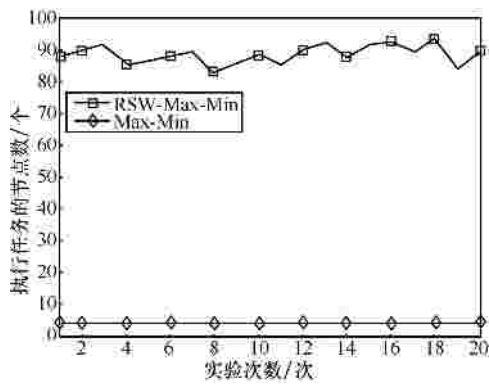
由图 3 可以看出, 当应用需求比较少时, 传统的 Min-Min 算法与 Max-Min 算法表现出比较大的波

动性, 任务执行成功率偶尔也可以达到 60%左右; 随着应用需求不断增加, 则趋于平缓, 稳定在 50%以下。而改进后的优化算法, 则一直表现出比较高的任务执行成功率, 因为改进后的算法每次调度都选择了最适合的资源提供服务, 资源匹配程度高。

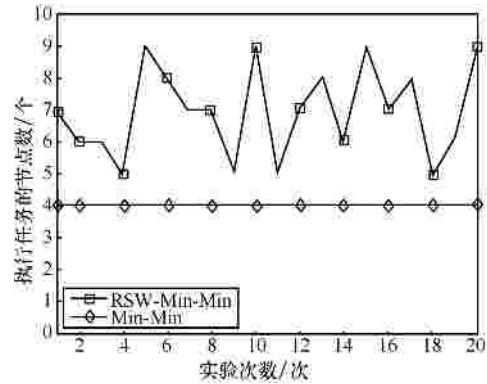
4.3 实验 2: 系统资源利用率比较

图 4 所示为当应用需求分别为(2,2)、(10,4)和(20,5)时, 改进的优化算法与传统 Min-Min 算法和 Max-Min 算法在系统资源利用率方面的比较。

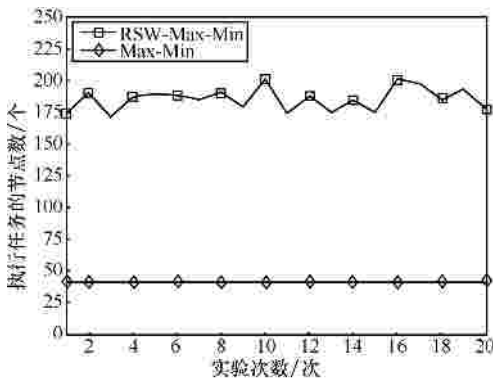
由图 4 可以看出, 改进后的优化算法比传统的 Min-Min 算法与 Max-Min 算法更有效地提升了系统



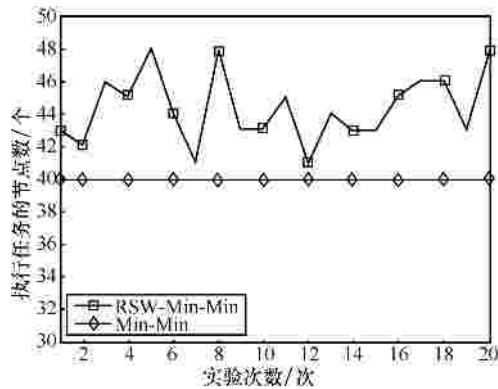
(a) 应用需求为(2, 2)时与 Max-Min 算法的资源利用率比较



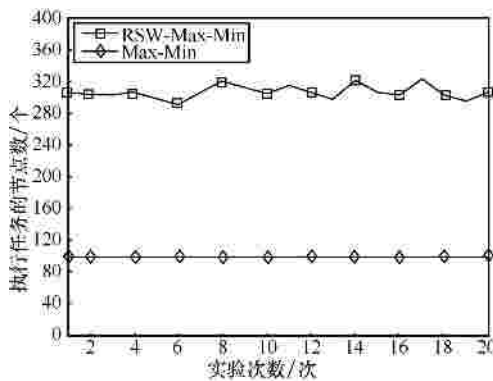
(b) 应用需求为(2, 2)时与 Min-Min 算法的资源利用率比较



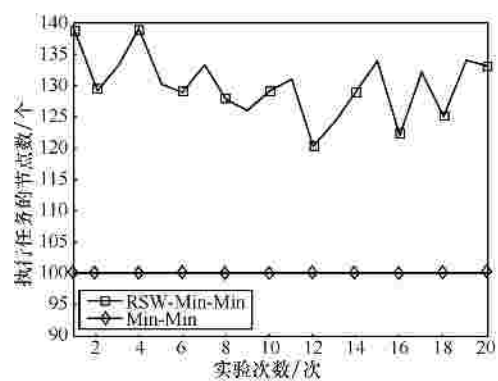
(c) 应用需求为(10, 4)时与 Max-Min 算法的资源利用率比较



(d) 应用需求为(10, 4)时与 Min-Min 算法的资源利用率比较



(e) 应用需求为(20, 5)时与 Max-Min 算法的资源利用率比较



(f) 应用需求为(20, 5)时与 Min-Min 算法的资源利用率比较

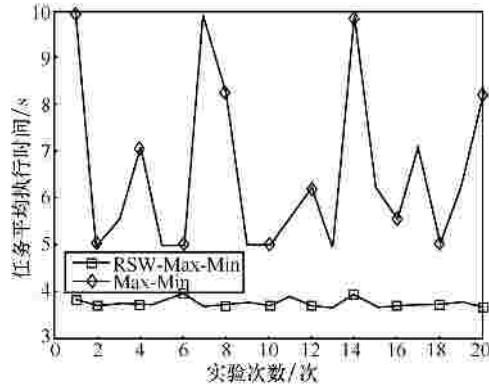
图 4 改进的算法与传统 Max-Min 和 Min-Min 算法在系统资源利用率上的比较

的资源利用率。所有子图中的改进算法执行任务的节点数较原算法大，意味着节点资源得到充分利用。传统的调度算法因其本身固有的缺陷使有些资源总是被调度，而有些资源总是空闲，导致负载的不平衡。而加入了基于“资源滑动窗口”的资源匹配模型之后的改进算法可有效提升系统的负载均衡和资源利用率，因为不同的资源偏好和条件调度会使各种异构的资源得到调度的机会，从而使整个系统负载均衡。

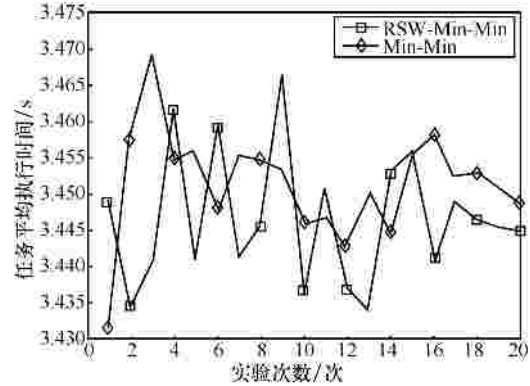
4.4 实验 3：完成时间比较

如图 5 所示，分别为当应用需求为(2,2)、(10,4)和(20,5)时，改进的优化算法与传统 Min-Min 算法和 Max-Min 算法在任务执行完成时间方面的比较。

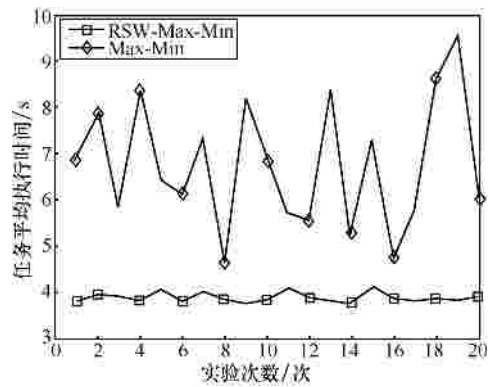
从图 5 可以看出，相对于传统的 Min-Min 算法，改进的算法当应用需求比较少时，平均完成时间差别不大；随着应用需求的不断增加，改进的算法会比传统的 Min-Min 算法有少量时间损耗的增加。而相对于原始 Max-Min 算法，改进算法有效减少了每



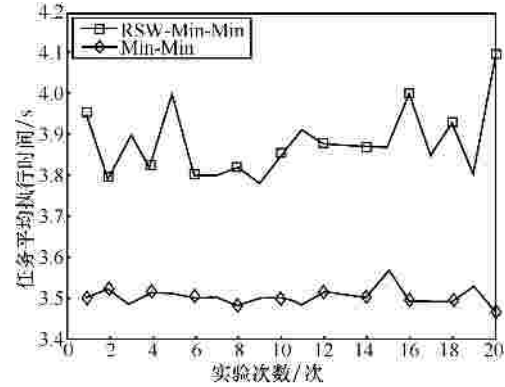
(a) 应用需求为(2, 2)时与 Max-Min 算法的执行时间的比较



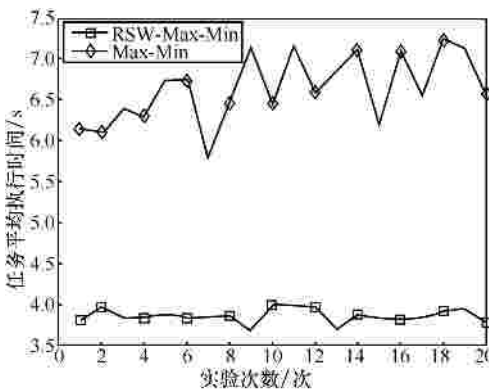
(b) 应用需求为(2, 2)时与 Min-Min 算法的执行时间的比较



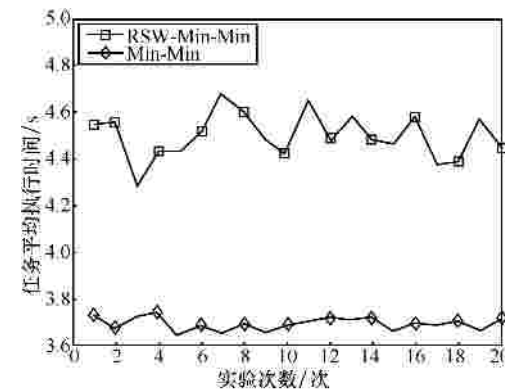
(c) 应用需求为(10, 4)时与 Max-Min 算法的执行时间的比较



(d) 应用需求为(10, 4)时与 Min-Min 算法的执行时间的比较



(e) 应用需求为(20, 5)时与 Max-Min 算法的执行时间的比较



(f) 应用需求为(20, 5)时与 Min-Min 算法的执行时间的比较

图 5 改进的算法与传统 Max-Min 和 Min-Min 算法在任务平均执行完成时间方面的比较

一个任务的平均执行时间。而且由于对资源事先有所预判,不像原始 Max-Min 算法对资源无法掌控,故而优化算法在任务平均执行时间方面耗时比较稳定,而原始 Max-Min 算法则表现出比较大的波动性。

5 结束语

本文通过分析虚拟计算环境及其应用所呈现出的特点,针对其海量、异构的资源与多样化应用之间的资源聚合问题,提出了一种面向应用的

基于信任的资源匹配模型——“资源滑动窗口”模型。该模型实现了“合适的资源服务于合适的任务”的调度目的。仿真实验结果表明本文所提模型可以有效提高系统的任务执行成功率和资源利用率。相对于传统的 Max-Min 算法,有效地缩短了任务的平均执行时间;而对于传统的 Min-Min 算法,牺牲了少量的时间损耗,确保了任务执行成功率和资源利用率的有效提升。

本文所提模型为解决基于条件约束、资源偏好、不同优先级别的多应用任务的合理有效的调度问题提出了可行的解决方案。下一步的工作主要是考虑对该资源匹配模型在资源聚类技术方面进一步优化,对计算类、存储类、带宽类等不同属性资源进行划分,以更好地满足不同的应用对资源的不同偏好需求。

参考文献:

- [1] 陈滢,王庆波,金津等. 虚拟化与云计算[M]. 北京:电子工业出版社,2009.
CHEN Y, WANG Q B, JIN X, *et al.* Virtualization and Cloud Computing[M]. Beijing: Electronic Industry Press, 2009.
- [2] 卢锡城,王怀民,王戟. 虚拟计算环境 iVCE: 概念与体系结构[J]. 中国科学(E 辑): 信息科学, 2006, 36(10):1081-1099.
LU X C, WANG H M, WANG J. Internet-based virtual computing environment iVCE: concepts and architecture[J]. Science in China Ser F-Information Sciences, 2006, 36(10):1081-1099.
- [3] ETMINANI K, NAGHIBZADEH M. A min-min max-min selective algorithm for grid task scheduling[A]. 2007 3rd IEEE/IFIP International Conference in Central Asia on Internet, ICI 2007[C]. Tashkent, Uzbekistan, 2007. 1-7.
- [4] PANDA S K, BHOI S K, KHILAR P M. A semi-interquartile min-min max-min (SIM2) approach for grid task scheduling[A]. Proceedings of International Conference on Advances in Computing, ICAdC 2012[C]. Bangalore, Karnataka, India, 2012. 415-421.
- [5] WEI X M. Study of ant colony hybrid algorithm in grid task scheduling[J]. Advances in Information Sciences and Service Sciences, 2012, 4(15):325-331.
- [6] 林伟伟,齐德昱. 基于概率冗余调度的可靠 P2P 计算模型[J]. 华南理工大学学报(自然科学版), 2010, 38(6):84-88.
LIN W W, QI D Y. Reliable P2P computing model based on probabilistic redundant scheduling[J]. South China University of Technology (Natural Science), 2010, 38(6):84-88.
- [7] XU X, HU H, HU N, *et al.* Cloud task and virtual machine allocation strategy in cloud computing environment[J]. Communications in Computer and Information Science, 2012, 345:113-120.
- [8] 简源,罗四维,杨麟儿. 基于推荐机制的网格资源匹配算法研究[J]. 计算机研究与发展, 2009, 46(11):1814-1820.
LIN Y, LUO S W, YANG L E. Recommendation based grid resource matching algorithm[J]. Journal of Computer Research and Development, 2009, 46(11):1814-1820.
- [9] BARAGLIA R C, GABRIELE D P, PAGANO G. A multi-criteria job scheduling framework for large computing farms[J]. Journal of Computer and System Sciences, 2013, 79(2):230-244.
- [10] CHIANG C W. Two novel genetic operators for task matching and scheduling in heterogeneous computing environments[J]. Journal of Internet Technology, 2012, 13(5):773-784.
- [11] ABBADI I M, RUAN A. Towards trustworthy resource scheduling in clouds[J]. IEEE Transactions on Information Forensics and Security, 2013, 8(6):973-984.
- [12] WAN B T. Qos min-min scheduling strategy[J]. International Journal of Distributed Sensor Networks, 2009, 5(1):13.
- [13] 李文娟,张启飞,平玲娣等. 基于模糊聚类的云任务调度算法[J]. 通信学报, 2012, 33(3):146-154.
LI W J, ZHANG Q F, PING L D, *et al.* Cloud scheduling algorithm based on fuzzy clustering[J]. Journal of Communications, 2012, 33(3):146-154.
- [14] 朱春鸽. 网格环境下信任的研究与应用[D]. 北京:北京邮电大学, 2006.
ZHU C G. Research and Application on Trust in Grid Environment[D]. Beijing: Beijing University of Posts and Telecommunications, 2006.
- [15] 王伟,曾国荪. 一种基于 Bayes 信任模型的可信动态级调度算法[J]. 中国科学(E 辑): 信息科学, 2007, 37(1):285-296.
WANG W, ZENG G S. Trusted dynamic level scheduling algorithm based on bayesian trust model[J]. Science in China Ser F Information Sciences, 2007, 37(1):285-296.
- [16] JOSANG A, ISMAIL R. The beta reputation system[A]. Proceedings of the 15th Bled Conference on Electronic Commerce[C]. Bled, Slovenia, 2002. 324-337.
- [17] ZHU C G, LIU X R, YANG Y X, *et al.* An application-oriented scheduling optimization algorithm for internet-based virtual computing[A]. 2013 IEEE 7-th International Symposium on Service-Oriented System Engineering[C]. CA, USA, 2013. 347-352.

作者简介:



朱春鸽(1981-),女,河南许昌人,北京邮电大学博士生,主要研究方向为虚拟计算、云计算、网络安全等。

刘欣然[通信作者](1971-),男,黑龙江鸡西人,北京邮电大学教授、博士生导师,主要研究方向为分布式计算、信息安全等。E-mail: lxr@isc.org.cn。

杨义先(1961-),男,四川盐亭人,北京邮电大学教授、博士生导师,主要研究方向为密码学、网络与信息安全等。

张鸿(1976-),男,陕西西安人,国家计算机网络应急技术处理协调中心高级工程师,主要研究方向为云计算、信息技术、网络安全等。