ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.8, August 2007, pp.2027−2037
DOI: 10.1360/jos182027

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

虚拟计算环境中的多机群协同调度算法*

张伟哲[1+], 田志宏[1,2], 张宏莉[1], 何 慧[1], 刘文懋[1]

[1](哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

[2](中国科学院 计算技术研究所,北京 100080)

Multi-Cluster Co-Allocation Scheduling Algorithms in Virtual Computing Environment

ZHANG Wei-Zhe[1+], TIAN Zhi-Hong[1,2], ZHANG Hong-Li[1], HE Hui[1], LIU Wen-Mao[1]

[1](School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

[2](Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: +86-451-86419652, Fax: +86-451-86413309, E-mail: wzzhang@hit.edu.cn, http://www.hit.edu.cn

Zhang WZ, Tian ZH, Zhang HL, He H, Liu WM. Multi-Cluster co-allocation scheduling algorithms in virtual computing environment. *Journal of Software*, 2007,18(8):2027−2037. http://www.jos.org.cn/1000-9825/18/2027.htm

Abstract: Based on the core mechanisms of Internet-based virtual computing environment (iVCE), a novel architectural framework for the multi-cluster task co-allocation is proposed by introducing the autonomic scheduling elements, domain scheduling commonwealth and meta-scheduling executor. A new multi-cluster task scheduling schema based on the multi-cluster task execution performance model is presented. Four multi-cluster heuristic scheduling algorithms are provided. Experiments indicate the scheduler schema and the algorithms are effective in the objective function of makespan and average utilization.

Key words: network computing; virtual computing environment; multi-cluster co-allocation; task scheduling; resource selection strategy

摘 要: 基于虚拟计算环境的核心机理,提出由自主调度单元、域调度共同体、元调度执行体为核心的多机群协同系统框架.剖析多机群任务并发运行性能模型,设计了多机群协同调度算法框架,提出最大空闲节点优先、最小网络拥塞优先、最小异构因子优先与最小异构空闲节点优先 4 种启发式资源选择策略.实验验证了协同调度模型与算法在任务集完成时间与系统平均利用率的测度上的有效性.

关键词: 网络计算;虚拟计算环境;多机群协同;任务调度;资源选择策略

中图法分类号: TP316 **文献标识码**: A

随着互联网基础设施的不断完善,封闭于单一组织的超级计算机已经难以满足科学计算与工程设计等大规模挑战性应用的需求.跨越多组织、多管理域的多机群共享与协同工作成为大规模并行计算的发展趋势[1].

* Supported by the National Natural Science Foundation of China under Grant No.90412001 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2006AA02Z334 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.G2005CB321806 (国家重点基础研究发展计划(973))

Received 2007-02-24; Accepted 2007-04-26

多机群协同取消了任务运行不能跨越机群边界的限制,允许并行任务的不同部分并发运行于分布在多个管理域的异构机群中,跨越多个机群运行的任务需要利用互联网进行同步与通信.

多机群协同工作需要跨域处理从资源监测、任务调度到可信保障等众多复杂问题,其中,任务调度是多机群资源管理中亟待解决的核心问题.高效的多机群共享模型与任务协同调度算法可以提高网络计算系统的处理能力,降低任务的响应时间,保障用户对服务质量的需求.即便在封闭的超级计算机中,并行任务调度也依然是一个极富挑战性的问题[2],而多机群计算系统中数量众多的机群资源分布在不同的地理位置,耦合度较低,计算能力存在较大差异,具备网络计算系统"成长性、自治性、多样性"等自然特征,从而向调度模型与算法的设计提出了新的挑战.

近年来,针对多机群协同调度问题开展了大量研究工作[3−17],但这些研究工作存在两方面的不足:

(1) 从体系结构角度来看,多数研究工作由于缺乏对网络计算系统成长性与自治性的认识,试图对每个机群都实施全局统一的控制和管理.事实上,每个机群都可能具有不同的本地管理策略,随时有可能因为本地占用或其他原因,撤销其共享的机群资源,网络中机群资源相关的全局资源信息很难获取和维护.

(2) 从算法设计角度来看,不同管理域的资源具有很强的异构性,而当前算法研究或假设不同管理域间机群均为同构,或忽略网络带宽变化对任务执行时间的影响,协同调度算法不具备异构适应性与网络适应性,难以在多机群协同环境中得到实际应用.

本文针对上述问题,在系统结构方面遵循虚拟计算环境的体系结构与核心机理,提出以元调度执行体、域调度共同体和自主调度单元为核心的多机群协同调度框架,解决多域、多组织引入的成长性、分布性与自治性的问题.在多机群协同调度算法方面,在前期工作的基础上[17,18],引入了基于异构因子与具备网络意识的多机群任务协同运行性能模型.而后,以性能模型为基础,构造了最大空闲节点优先、最小网络拥塞优先、最小异构因子优先与最小异构空闲节点优先算法这 4 种启发式任务调度策略,充分考虑了多机群计算环境的异构性与动态性.

本文第 1 节介绍相关工作,同时进一步阐明本文与相关工作的差异和深入研究的意义.第 2 节基于虚拟计算环境的核心机理,提出多机群协同系统框架.第 3 节详细阐述多机群任务并发性能模型,并提出多机群协同调度算法.第 4 节给出本文算法的实验设置以及实验结果.第 5 节对本文的工作进行总结并对后续工作进行展望.

1 相关工作

多机群协同的网络计算系统设计需要采用面向互联网的软件新技术.当前,国内外研究人员从不同角度广泛而深入地探讨了网络计算系统的核心机理,提出了网格计算(grid computing)[19]、虚拟计算环境(virtual computing environment)[20]、面向服务的计算(service-oriented computing)[21,22]和网构软件(Internetware)[23]等技术途径.其中,虚拟计算环境技术[20]通过提出自主元素、虚拟共同体和虚拟执行体等概念,支持开放环境下资源的按需聚合和自主协同,为终端用户或应用系统提供和谐、可信、透明的一体化服务,也为本文的多机群协同计算系统的构建奠定了理论基础.

目前,国内外基于Internet的任务调度策略研究主要有两类:应用级任务调度和作业级任务调度.应用级任务调度由传统机群环境下基于任务图(DAG)的调度问题演化而来,通过将计算密集型应用抽象为粗粒度约束任务图,采用经济模型和数学规划策略将其映射到网络计算资源,从而提高应用性能.由于当前网络环境存在高延迟、低带宽等缺点,这方面的研究工作仅局限于参数扫描与松耦合迭代应用.澳大利亚墨尔本大学的Rajkumar Buyya等人[24]和美国加州大学的Grail实验室[25]是应用级任务调度研究的代表.国内杜晓丽等人针对网格环境下DAG图调度问题提出了基于模糊聚类的任务调度启发式[26].

作业级任务调度研究对象是独立作业集的性能优化问题,是高性能机群作业调度研究在网络计算环境下的延伸,也是本文主要研究的内容.机群任务调度与网络环境中的任务调度显著不同.高性能机群多为由同构节点采用高速链路互连,任务调度以提高系统吞吐率和资源利用率为目的.而网络环境是由不同管理域异构机群通过Internet互连,机群规模具有动态性和成长性.国外历年的作业调度策略专题工作组(workshop on job

scheduling strategies for parallel processing)*与异构计算专题工作组(international heterogeneous computing workshop)**中收录了许多此方面的重要工作.其中,多机群协同方面的研究与本文关系最为密切.Abawajy与Dandamudi提出了一种动态在线作业调度策略,通过多机群间作业流动态分配,提高作业响应时间与系统利用率[3].同时,Sabin等人提出多机群并发请求策略,将作业请求并发提交到多个独立机群,一旦开始运行后立即取消冗余请求,可以有效地降低作业平均响应时间[4].Ernemann等人的研究发现,利用分布在不同时区的机群协同工作,作业的平均响应时间较之位于相同时区的多机群系统可以降低 30%[5].Yahyapour等人的模拟结果表明,即使当附加通信开销达到原程序运行开销的 25%时,同构机群间多址协同算法也仍然优于单址协同算法[6].而后,进一步对多址协同算法针对作业分片限制进行优化[7],实验表明,在所有机群中,计算资源总数恒定的前提下,机群数目和每个机群内所含资源数目不会影响多址协同算法的性能[8].Epema等人在同构环境下设计了动态多机群协同调度服务框架[9],实现了基本的调度模块[10,11],并评估了多址协同算法中作业结构和大小、机群大小、机群内部与机群间网络通信比等因素变化对作业平均响应时间的影响[12,13].William等人的研究关注任务分配后机群间带宽变化对作业运行时间的影响,提出以带宽为中心的任务调度模型与启发式算法[14].林伟 伟[15]、丁箐[16]等人也分别针对树型网格计算环境与用户QoS保障方面开展了任务调度策略的研究.我们之前的研究工作[17,18]也是通过引入网格环境下作业级多址任务调度模型与性能模型,提出多址任务协同调度算法框架.以最优和贪心资源选择策略为核心,提出两种作业级多址协同调度算法.

当前,多机群协同算法的研究存在局限性:首先,部分调度策略[3−5,15,16]允许作业迁移至非本地机群执行,但不允许并行任务的不同部分跨越多个机群并发执行,忽视了多机群协同对降低作业响应时间和提高资源使用率的影响,不具备完备的协同性;其次,许多研究工作假设所有高性能机群均为同构[6−14],与真实的多机群计算环境有较大的差距,在异构性方面有所欠缺.更重要的是,多数工作[3−13]在建立多机群任务运行模型时均采用静态预估模型刻画任务通信时间,未考虑网络传输产生拥塞时任务执行时间的变化,缺乏网络意识.因此,多机群协同系统框架与任务调度算法设计亟待深入研究.

2 多机群协同计算系统框架

虚拟计算环境(Internet-based virtual computing environment,简称iVCE)[20]从构建资源的主体化模型、构建利益共同体、支撑资源间自主协同的角度提出了 3 个重要概念:自主元素(autonomic element)、虚拟共同体(virtual commonwealth)、虚拟执行体(virtual executor).自主元素是iVCE中的基本资源管理单位,是具有自主行为能力的资源管理者.虚拟共同体是指一组具有共同兴趣、遵从共同原则的自主元素构成的集合.虚拟执行体是指协同承担同一任务的相关自主元素,为完成该任务而形成的状态空间的总和.在此基础上,提出以资源层、资源虚拟层、聚合层、自主协同层和应用层为核心的网络计算系统体系结构.

基于虚拟计算环境的核心机理,iVCE 中多机群协同(iVCE for multi-cluster co-allocation)计算系统框架如图 1 所示,主要元素包括高性能机群、自主调度单元、域调度共同体、元调度执行体与独立作业集.

高性能机群处于多机群协同调度系统资源层.机群内部节点同构,由高速总线互连,连接稳定.机群间由于节点性能和负载不同,整体性能存在差异,且通过广域网互连,传输低速、不稳定.站点间可以交换用户提交的任务,共享计算资源.

自主调度单元为虚拟计算环境中自主元素的物化,位于 iVCE for multi-cluster co-allocation 中的资源虚拟层.自主调度单元通过统一资源映像,将机群整封装成一台虚拟计算机,屏蔽机群内部节点细节.自主调度单元负责感知机群内部资源的状态变化情况,向元信息服务器注册新加入的计算节点并反馈本地机群中节点的使用情况.同时,通过动作部件接收用户提交的任务调度请求,进行任务分发与结果收集.此外,自主调度单元还具备一定程度的自主决策能力,通过比对调度请求与掌握的资源情况进行最低需求过滤,从而减轻上层任务调度

* Workshops on Job Scheduling Strategies for Parallel Processing. 2005. http://www.cs.huji.ac.il/~feit/parsched/

** The Int'l Heterogeneous Computing Workshop. 2005. http://www.cs.umass.edu/~rsnbrg/hcw2005/hcw05_prev_workshops.html

系统的负担.

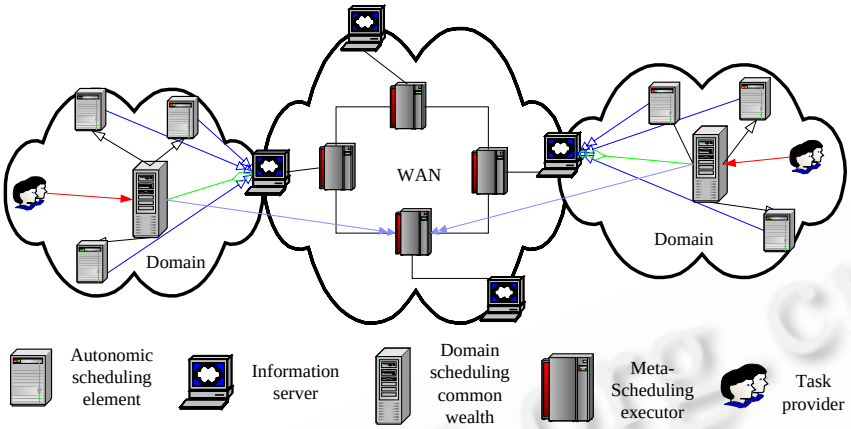


Fig.1 Framework of multi-cluster co-allocation task scheduling for iVCE

图 1 虚拟计算环境中的多机群协同任务调度系统框架

域调度共同体位于 iVCE for multi-cluster co-allocation 的聚合层,是具有并发作业协同能力的自主调度单元的集合,其核心部件是信息服务系统与域调度器.信息服务系统负责资源元信息的组织与发现.域调度器同时拥有域和全局资源选择与映射能力,以降低用户应用执行时间为目标,并不具备协作和提高系统吞吐量的能力.域调度器数目依赖于域的规模.当用户向域调度器提交任务后,域调度器通过信息服务器收集机群信息.然后,采用决策模块确定资源调度集,发送元调度请求至元调度器.最后,根据元调度器结果自动分发任务至自主调度单元.

元调度执行体是 iVCE for multi-cluster co-allocation 中自主协同层的虚拟执行体,也是多机群协同调度系统的核心组件,主要目的是协同各域调度的调度行为,避免域调度器过于关注最优调度结果而引发的资源争用、整体性能下降等负面效应.元调度接受域调度器的调度请求,通过与其他元调度器的协同,提高系统吞吐量.

3 多机群协同调度算法

本节首先提出了多机群任务并发执行的性能模型.而后,面向元调度执行体提出了最大空闲节点优先、最小网络拥塞优先与最小异构因子优先等 4 种启发式任务调度策略.最后,分析了多机群协同调度算法的时间复杂性.

3.1 多机群任务并发性能模型

网络环境下异构、多机群任务执行性能模型的建立是调度算法设计与评估的基础.设任务在队列中等待调度时间为T_W,计算时间为T_E,通信时间为T_C,则任务总运行时间T为

$$T=T_W+T_E+T_C \tag{1}$$

任务运行需要占用多个机群中的多个节点,而机群间计算能力的差异影响任务计算部分所占用的时间T_E.定义异构因子h描述不同站点计算能力的差异对任务计算部分的影响程度.异构因子受站点的计算能力、体系结构和存储方式等多种因素的影响,难以通过理论计算确定站点异构因子.然而,可以通过提取标准测试程序的代码"骨骼"[27],在不同站点上运行,获得相对某站点的异构因子.选择某机群作为基准机群,其计算能力为h_0,计算时间为T_0,其余机群异构因子为h_i,计算时间为$T_0\times h_i/h_0$.由于大多数多机群任务并发时需保持同步,所以,其运行时间应不短于性能最低的机群独立运行该任务的时间.设$j,k,\ldots,m$为参与任务运行的机群序号,则多机群执行计算部分时间T_E为

$$T_E=T_0\times\max(h_j,h_k,\ldots,h_m)/h_0 \tag{2}$$

任务的通信时间受触发事件、网络拓扑结构、网络带宽和并行应用的通信模型等多种因素的影响,因此,

通信模型应具备动态、时变的特征.触发事件可以分为任务运行结束与新任务调度执行两类,事件触发后,由于任务重新分配对带宽产生影响,所有任务的通信时间会发生变化,因此,通信时间由触发事件分隔的若干时间段组合而成,如图 2 所示.

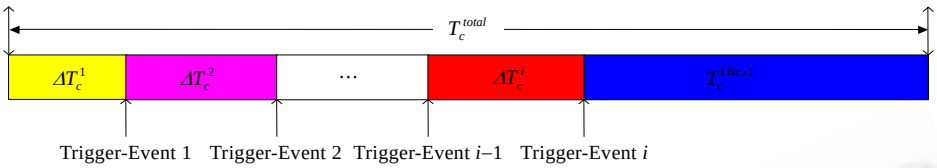


Fig.2 Dynamic communication time model for multi-cluster co-allocation task

图 2 多机群协同任务通信时间动态模型

在图 2 中, ΔT_c^i 表示触发事件 i–1~I 之间任务的通信时间, $T_c^{(Re,i)}$ 表示事件 i 发生后预估的剩余通信时间.借鉴William等人的工作[14],可以得出多机群执行计算部分时间T_c为

$$T_c = \sum_{m=1}^{i} \Delta T_c^i + T_c^{(Re,i)} \tag{3}$$

$T_c^{(Re,i)}$ 的计算需要深入分析网络带宽占用变化情况对通信时间的影响.假设机群内部为全互联紧耦合结构,机群间为星型连接,且任务通信模式为all-to-all.设任务k并发执行所需总节点数为n_t,与链路j相连的机群内部分配节点数为 n_k^j ,机群间两个节点发送与接受消息占用带宽(band width between nodes)为 BWBN,忽略机群内部节点间的通信开销,则任务 k 执行时需占用链路 j 的总带宽 BW_k^j 为

$$BW_k^j = BWBN \times n_k^j \times (n_t - n_k^j)/(n_t - 1) \tag{4}$$

在触发事件 i 发生后,需重新计算链路 j 带宽占用引起的拥塞情况.设链路 j 的固有带宽为 BW_{const}^j ,触发事件 i 发生时链路 j 的拥塞因子(congestion factor)为

$$CF_i^j = BW_{Const}^j \Big/ \sum_{\forall k \in Job} BW_k^j \tag{5}$$

其中,k 为并发过程中需要占用链路 j 的作业.不同时刻拥塞因子的变化必然引发任务剩余通信时间的变化,这里定义,若任务 i 所跨越的所有链路其 CF 值均大于 1.0,则任务剩余通信时间不受网络拥塞影响;反之,如果存在某个链路 CF 值均小于 1.0,则此链路拥塞延长此任务后续的通信时间.为此,进一步引入伸缩因子(flex factor)为

$$FF_i = \begin{cases} 1, & \text{if } \forall j \in Link_i,\ CF_i^j \geq 1.0 \\ \min(CF_i^j), & \text{if } \forall j \in Link_i,\ CF_i^j < 1.0 \end{cases} \tag{6}$$

综上,由公式(4)~公式(6)可以得出剩余通信时间 $T_c^{(Re,i)}$ 为

$$T_c^{(Re,i)} = (T_c^{(Re,i-1)} - \Delta T_c^i) \times FF_{i-1} \times (FF_i)^{-1} \tag{7}$$

3.2 多机群协同调度算法

网络环境下异构机群任务调度为NP-hard问题,难以获得多项式时间内的最优调度算法.分析多机群任务并发运行性能模型(公式(1)~公式(3))可以发现,影响任务执行时间的主要因素为任务在队列中的等待时间(T_W)、机群异构因子的差异(h_i)和任务的通信开销(BWBN),因此,本节提出了 4 种启发式调度算法,从降低任务等待时间、优先选择较低异构因子资源和网络意识的角度对任务调度进行优化.

3.2.1 基调度策略

算法思想:系统接受任务调度请求,将其放入等待队列并触发调度事件.调度时,依次检查等待队列中的任务.如果资源需求可以满足,则进行资源分配并启动任务执行;反之,则跳过此任务,尝试调度其后的任务.系统定时检查任务完成情况,若有任务完成,则回收资源并触发新的调度事件.

算法 1. 多机群协同调度算法框架.

输入:(1) 任务等待队列;(2) 机群集合.

输出:(1) 映射结果.

(2) 中间变量:(a) 调度间隔Δt;(b) 任务队列中未调度首作业 *CurrentJob*;(c) 分配的节点集合 *Sites CurrentJob*.

步骤 1:初始化,启动调度程序.

步骤 2:检查运行队列,若有任务完成,则释放资源,重新执行步骤 2;若无任务完成,则更新任务的剩余完成时间.

步骤 3:检查等待队列,若队列为空,等待Δt 调度间隔,重新检测任务队列状态;若队列非空,则从任务队列与本地调度器中分别收集任务请求信息与机群状态信息.

步骤 4:映射.针对当前队列中所有未分配资源的任务请求,将当前任务队列首任务赋予 *CurrentJob*.

(a) 资源的选择.调用资源选择策略*AllocateJob*(*CurrentJob*),若调度成功,则获得任务执行的节点集*Sites*.

(b) 更新任务队列与机群状态.

步骤 5:返回映射结果并从步骤 1 重新执行.

3.2.2 资源选择策略

算法 2. 资源选择启发式算法框架.

输入:(1) 任务等待队列;(2) 机群集合.

输出:(1) 映射结果.

(2) 中间变量:(a) 调度间隔Δt;(b) 任务队列中当前要处理的作业 *CurrentJob*;(c) 按照自由节点数目排序的机群集合 *SortedQueue*;(d) 所需剩余的节点 *RemainSite CurrentJob*;(e) 当前机群的所有可用节点 *AvailSites*;(f) 当前机群的异构因子 *HeterRate*;(g) 机群关联链路拥塞度 *Saturation*.

步骤 1:如果 *Currentjob* 为空,则将 *Currentjob* 指向等待对列的第 1 个任务,找到等待队列 *Currentjob* 的下一个任务,将其作为 *Currentjob*.若已经到等待队列的末尾,则退出;将等待队列按照性能模型中的影响因素进行排序,生成 *SortedQueue*;设置 *RemainSite=CurrentJob* 的所需节点.

步骤 2:如果任务要求的节点比所有资源还多,则没有满足条件的节点集,放弃,转到步骤 1.

步骤 3:如果任务要求的节点比所有可用资源还多,则暂时没有满足条件的节点集,转到步骤 1.

步骤 4:遍历 *SortedQueue*;若当前的机群可用节点 *AvailSites* 少于 *RemainSite*,则将该机群的所有可用节点分配给 *CurrentJob*,*RemainSite=RemainSite–AvailSites*;若当前的机群可用节点 *AvailSites* 大于 *RemainSite*,则将该机群中的 *AvailSites* 分配给 *CurrentJob*.

步骤 5:更新任务队列与机群状态,转到步骤 1.

资源选择策略是多机群任务协同调度的核心组件,根据其为机群生成 *SortedQueue* 时采用的启发式信息的不同,可以衍生出一族多机群协同调度算法.本文中对算法名称作如下约定:若等待队列按照机群空闲节点数目 *AvailSites* 进行排序生成 *SortedQueue*,则称此算法为最大空闲节点优先启发式算法(biggest free node priority);若等待队列按照机群关联链路的拥塞程度 *Saturation* 进行排序,则称此算法为最小网络拥塞优先启发式算法(smallest network congestion priority);若等待队列按照机群的异构因子 *HeteroRate* 进行排序,则称此算法为最小异构因子优先算法(smallest heterogeneous factor priority);若等待队列按照机群空闲节点与异构因子综合作用 *AvailSites/HeterRate* 效果进行排序,则称此算法为最小异构空闲节点优先算法(smallest heterogeneous free node priority).算法主体部分是对任务队列遍历与采用快排序算法多机群集合按不同性能指标进行排序,因此,算法时间复杂度为 $O(n\log n+m)$,其中,n 为机群数目,m 为任务队列包含的任务数.

4 仿真实验结果及性能分析

4.1 仿真实验环境及算法设置

本文开发的多机群网络计算系统仿真平台以国家重点基础研究发展计划(973)支持的“虚拟计算环境实验床”的基础设施为蓝图.“虚拟计算环境实验床”是由国家计算机网络应急技术处理协调中心(CNCERT/CC)和哈

尔滨工业大学(HIT)协作建设,以国家计算机网络应急技术处理协调中心遍布全国31个省份的网络基础设施及计算资源为基础,对分布自治资源进行集成和综合利用,构建起的一个开放、安全、动态、可控的大规模虚拟计算环境实验平台,研究并验证虚拟计算环境聚合与协同机理.

在前期研究工作[17,18]的基础上,我们对机群规模做了进一步的扩充.机群数目为3个,共计552个节点,其中,北京360个节点、上海128个节点、哈尔滨64个节点.机群间采用千兆带宽星型互联,不同机群计算能力相异.工作负载采用Feitelson等人收集的Cornell Theory Center高性能计算机群IBM RS/600 SP共11个月的并行负载工作日志***,对任务规模进行扩充以适应多机群协同计算模型.并行负载为相互独立的并行任务集,其中并行任务是调度的原子对象,不能分解为子任务.任务宽度固定且运行过程中不允许被抢先.任务对节点的使用是独占式的,运行过程中任务不具备可塑性.在节点数量满足请求宽度的前提下,任务可以在任意机群节点集上执行,可以通过机群内部与机群间网络传递信息,但并行任务对所分配节点必须同时占用与释放.

仿真实验将主要采用任务集合的完成时间(Makespan)与系统平均利用率(average utilization)这两个测度,同时从用户与系统的角度验证算法性能.这两个测度定义如下:

定义1(任务集完成时间(Makespan)).

$$Makespan=\max(T_1,T_2,\ldots,T_{n-1},T_n),$$

其中,T_i表示任务i的完成时间.

定义2(系统平均利用率(AU)).

$$AU=\sum(N_{used}/N_{total})/Num,$$

其中,N_{used}表示每秒有作业运行的节点数目,N_{total}表示节点总数,Num为检测总次数.

多机群协同调度算法中,基调度策略的时间间隔Δt设置为 1 秒.由于多机群协同调度算法相关工作中较少考虑机群计算机能力异构与网络拥塞对任务性能模型的影响,因此采用Yahyapour[4]与Epema[10]等人提出的单机群协同调度算法与本文提出的多机群协同调度算法进行性能比较,同时选用William[14]等人提出的非共享调度算法与理想调度算法作为参照算法,以给出性能的上、下边界.其中,单机群协同(single cluster co-allocation)算法要求任务可以在不同机群间迁移,但并行任务不能跨越机群边界执行;非共享调度算法(No-Share)仅允许任务在提交的本地机群执行,不允许迁移或多机群执行;理想调度算法(Idea)忽略机群间网络开销,所有机群均为全互联.

4.2 实验结果及性能分析

本节首先将本文提出的多机群协同算法与Yahyapour等人提出的单机群协同调度算法(SCCA)[4,10]及两种参照算法(No-Share和Idea)[14]从任务集合的完成时间(Makespan)与系统平均利用率(average utilization)两个测度进行性能比较,而后进一步考察了异构因子与任务请求节点数目变化等因素对多址协同算法的影响.

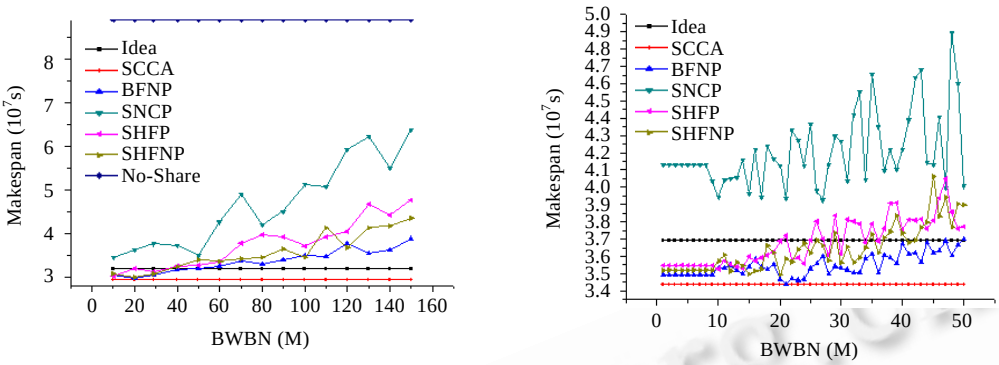
4.2.1 任务集完成时间的比较

设置3个机群的异构因子分别为1.0,2.0和4.0,表示机群1的计算能力是机群2的2倍,是机群3的4倍.网络之间的带宽为1 000M,任务请求的平均节点数为40.图3为各种算法随任务平均带宽需求(BWBN)变化的效率对比图.图3(a)为BWBN从1变化到200时的任务集完成时间(含No-Share算法);图3(b)为BWBN从1变化到50时的任务集完成时间(不含No-Share算法).

从图3(a)中可以看出,参考算法中性能最优的算法是SCCA算法,Idea算法的作业完成时间是SCCA算法的1.5倍,最差的是No-Share算法,与SCCA算法相差近3倍.SCCA算法将无法本地执行的任务迁移到其他机群,具备一定程度的单机群协同特征.Idea算法将全部机群等效成一个机群,可以将任务分配给多个机群执行,具有较完备的协同性.但由于忽略异构性影响,任务在多个机群上执行,最终的完成时间取决于计算时间最长的机群.所以,将任务分配到的协作机群数越多,其最终完成的时间可能会越长.而在No-Share算法中,因为某机群资源暂时无法满足当前提交的任务请求,该任务只能在本地站点继续等待,无法转移到其他站点或与其他站点协

*** Feitelson D. Parallel Workloads Archive. URL, 2002. http://www.cs.huji.ac.il/labs/parallel/workload/

作.因此,考虑多机群协作的算法降低了任务在队列中等待调度的时间,优势比较明显.



(a) BWBN varies from 1M to 200M with No-Share
(a) BWBN 从 1M 变化到 200M(含 No-Share 算法)

(b) BWBN varies from 1M to50M without No-Share
(b) BWBN 从 1M 变化到 50M(不含 No-Share 算法)

Fig.3 Comparisons of makespan for different multi-cluster co-allocation scheduling algorithms

图 3 不同多机群协同调度算法任务集完成时间的比较

在启发式算法中,BFNP 算法性能最优,随着 BNBW 的变化,Makespan 增长的趋势比较平缓,并且只有当 BNBW 低于 80M 时,BFNP 均优于 Idea 算法.其主要原因在于,BFNP 算法优先选取空闲节点最多的机群作为调度对象,使得任务在单机群执行的概率最大,将网络发生拥塞的概率降到最低.SHFNP 算法性能次之,SHFP 算法处于第 3 位,而 SNCP 算法性能最差.4 种多机群协同算法随着 BNBW 的增长,性能逐渐下降.然而从图 3(b)中可以看出,当 BNBW 小于 40M 时,任务集完成时间优于 Idea 算法.考虑到参考算法采用的性能模型与真实网络环境存在较大差异,因此,本文提出的多机群协同调度算法族在实际应用中具备更大的潜力.

4.2.2 系统平均利用率的比较

如图 4 所示,缺乏机群间协同的 No-Share 调度算法的系统平均利用率最低,仅为 53.26%.其次是部分协同的 SCCA 算法,其利用率为 83.42%.而本文提出的 4 种启发式算法系统利用率始终维持在 97%以上,且随 BNBW 而发生变化,利用率总体趋势保持稳定.因为 No-Share 算法只能利用本地站点的节点,所以会造成其他站点即使有空闲节点,但任务仍然无法利用,导致整体利用率下降;而 SCCA 算法则仅进行迁移,在单个站点上执行,不能利用多个站点上的可用节点,所以利用率也不高;而本文提出的网络环境下的多机群协同调度算法族充分利用其他机群的空闲资源,保障任务完成时间的同时最大化系统利用率,体现出较大的优势.

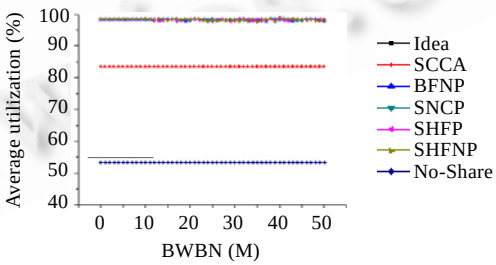


Fig.4 Comparisons of average utilization for different multi-cluster co-allocation scheduling algorithms

图 4 不同多机群协同调度算法系统平均利用率的比较

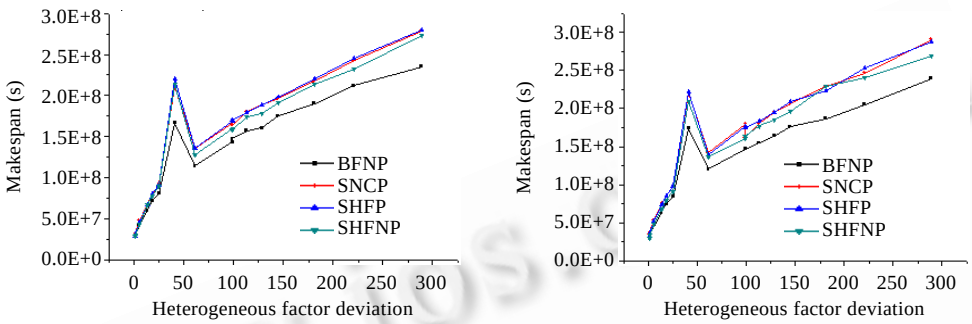
4.2.3 异构因子的变化对多机群协同调度算法的影响

异构因子衡量机群间计算能力的差异.多机群系统计算能力的差异可用异构因子方差(heterogeneous factor deviation)来衡量,方差越大,表示不同机群间计算能力的差异越大.在实验中,设异构因子方差变化范围为 0~250,以测试机群计算能力的差异对调度算法性能的影响.

由图 5(a)与图 5(b)可以得出如下结论:

1) 在 BWBN 为 20M 或 50M 的情况下,随着异构因子方差的增大,机群间的计算能力越发不均衡,所有多机群协同调度算法性能均有所下降,机群异构性对调度算法设计具有重要影响.

2) 随着异构因子方差的增大,多机群调度算法中性能的排序仍然是 BFNP 最优,其次是 SHFNP 算法,而后依次是 SNFP 和 SNCP 算法.异构因子方差的变化没有对算法性能排序造成影响,以降低任务等待时间为目的 BFNP 依然排在首位,说明在诸多影响因素中采用多机群协同降低任务等待时间 *Tw* 的权重要高于异构因子的适应性.



(a) BWBN=20M (b) BWBN=50M

Fig.5 Effect of multi-cluster scheduling algorithms with different heterogeneous factor deviations

图 5 异构因子变化对多机群协同调度算法的影响

4.2.4 任务请求节点数目的变化对多机群协同调度算法的影响

如图 6 所示,当任务请求节点数目较小时,系统能够很容易地将其分配到一个或多个机群;相反,如果任务请求节点数目增大,系统没有足够的节点供分配,则该任务只能继续等待,延长了任务的整体完成时间;同时,过多的负载容易被分配到多个站点运行,容易产生通信开销,造成网络拥塞,从而降低系统整体的效率.

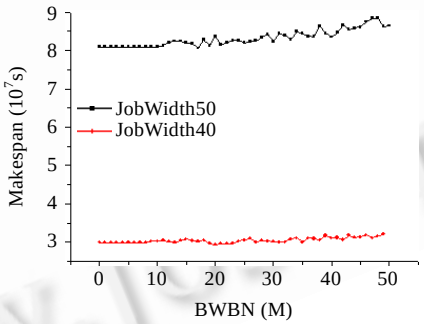


Fig.6 Effect of multi-cluster co-allocatio scheduling algorithms with different job width

图 6 任务请求节点数目变化对多机群协同调度算法的影响

实验环境设置为:提交任务的平均请求节点数为 50 个节点,比前面的实验增加了 10 个节点,选择性能最好的 BFNP 算法为验证对象.将实验结果与第 4.4.1 节的实验结果进行比较分析发现,所有调度算法完成时间是平均请求节点数为 40 时完成时间的 2~3 倍.

5 结 论

本文针对网络计算环境下多机群协同计算系统与任务调度算法进行深入研究,主要贡献如下:

1) 通过分析网络计算环境的自然特征,基于虚拟计算环境的核心机理,提出由自主调度单元、域调度共同体、元调度执行体为核心的多机群协同计算系统框架;

2) 引入具有异构适应性与网络意识的多机群任务并发运行性能模型;

3) 提出了多机群协同调度算法框架,并进一步给出最大空闲节点优先、最小网络拥塞优先、最小异构因子优先与最小异构空闲节点优先 4 种启发式资源选择策略.

本文在多机群任务调度方面做了一些初步的工作.今后,我们将进一步深入分析任务等待时间、异构因子与网络拥塞对算法性能的影响,开发自适应的多机群任务调度策略.

References:

[1] Feitelson DG, Rudolph L, Schwiegelshohn U. Parallel job scheduling—A status report. In: Feitelson DG, Rudolph L, Schwiegelshohn U, eds. Proc. of the Job Scheduling Strategies for Parallel Processing. LNCS 3277, Berlin: Springer-Verlag, 2004. 1−16.

[2] Dong FP, Akl SG. Scheduling algorithms for grid computing: State of the art and open problems. Technical Report, 2006. http://www.cs.queensu.ca/TechReports/Reports/2006-504.pdf

[3] Abawajy JH, Dandamudi SP. Parallel job scheduling on multicluster computing systems. In: Proc. of the IEEE Int'l Conf. on Cluster Computing (CLUSTER 2003). Oakland: IEEE Computer Press, 2003. 11−18.

[4] Sabin G, Kettimuthu R, Rajan A, Sadayappan P. Scheduling of parallel jobs in a heterogeneous multisite environment. In: Feitelson DG, Rudolph L, Schwiegelshohn U, eds. Proc. of the Job Scheduling Strategies for Parallel Processing. Berlin: Springer-Verlag, 2003. 87−104.

[5] Ernemann C, Hamscher V, Yahyapour R. Benefits of global grid computing for job scheduling. In: Buyya R, ed. Proc. of the 5th IEEE/ACM Int'l Workshop on Grid Computing in Conjunction with SuperComputing 2004. Oakland: IEEE Computer Press, 2004. 374−379.

[6] Hamscher V, Schwiegelshohn U, Streit A, Yahyapour R. Evaluation of job-scheduling strategies for grid computing. In: Proc. of the Grid Computing (Grid 2000) at 7th Int'l Conf. on High Performance Computing (HiPC 2000). LNCS 1971, Berlin: Springer-Verlag, 2004. 191−202.

[7] Ernemann C, Hamscher V, Streit A, Yahyapour R. Enhanced algorithms for multisite scheduling. In: Proc. of the 3rd IEEE/ACM Int'l Workshop on Grid Computing (Grid 2002) at Supercomputing 2002. LNCS 2536, Berlin: Springer-Verlag, 2002. 219−231.

[8] Ernemann C, Hamscher V, Yahyapour R, Streit A. On effects of machine configurations on parallel job scheduling in computational grids. In: Proc. of the Int'l Conf. on Architecture of Computing Systems. Springer-Verlag, 2002. 169−179.

[9] Sinaga JMP, Mohamed HH, Epema DHJ. A dynamic co-allocation service in multicluster systems. In: Feitelson DG, Rudolph L, Schwiegelshohn U, eds. Proc. of the Job Scheduling Strategies for Parallel Processing. LNCS 3277, Berlin: Springer-Verlag, 2005. 194−209.

[10] Bucur AID, Epema DHJ. The maximal utilization of processor co-allocation in multicluster systems. In: Werner B. ed. Proc. of the 17th Int'l Parallel and Distributed Processing Symp. (IPDPS 2003). Oakland: IEEE Computer Press, 2003. 60−69.

[11] Bucur AID, Epema DHJ. The performance of processor co-allocation in multicluster systems. In: Lee S, Sekiguchi S, eds. Proc. of the 3rd IEEE/ACM Int'l Symp. on Cluster Computing and the Grid (CCGrid 2003). Oakland: IEEE Computer Press, 2003. 302−309.

[12] Bucur AID, Epema DHJ. The influence of the structure and sizes of jobs on the performance of co-allocation. In: Feitelson DG, Rudolph L, eds. Proc. of the 6th Workshop on Job Scheduling Strategies for Parallel Processing. LNCS 1911, Berlin: Springer-Verlag, 2000. 154−173.

[13] Bucur AID, Epema DHJ. The influence of communication on the performance of co-allocation. In: Feitelson D, Rudolph L, eds. Proc. of the 7th Workshop on Job Scheduling Strategies for Parallel Processing. LNCS 2221, Berlin: Springer-Verlag, 2001. 66−86.

[14] William J, Walter L, Louis P, Daniel S. Characterization of bandwidth-aware meta-schedulers for co-allocating jobs across multiple clusters. Journal of Supercomputing, 2005,34(2):135−163.

[15] Ding J, Chen GL, Gu J. A unified resource mapping strategy in computational grid environments. Journal of Software, 2002,13(7): 1303−1308 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/13/1303.pdf

[16] Lin WW, Qi DL, Li YJ, Wang ZY, Zhang ZL. Independent tasks scheduling on tree-based grid computing platforms. Journal of Software, 2006,17(11):2352−2361 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/2352.htm

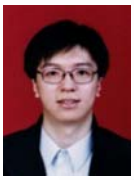
[17] Zhang WZ, Albert C, Hu MZ. Multisite co-allocation algorithms for computational grid. In: Proc. of the 3rd High-Performance Grid Computing Workshop (HPGC 2006), Associated with the Int'l Parallel and Distributed Processing Symp. 2006 (IPDPS 2006). New York: IEEE Press, 2006. 8.

[18] Zhang WZ, Fang BX, Hu MZ, Liu XR, Zhang HL, Gao L. Multisite co-allocation scheduling algorithms for parallel jobs in computing grid environments. Science in China (Series E), 2006,36(10):1240−1262 (in Chinese with English abstract).

[19] Foster I, Kesselman C, Tuecke S. The anatomy of the grid: Enabling scalable virtual organizations. Int'l Journal of High Performance Computing Applications, 2001,15(3):200−222.
[20] Lu XC, Wang HM, Wang J. Internet-Based virtual computing environment (iVCE): Concepts and architecture. Science in China (Series E), 2006,36(10):1081−1099 (in Chinese with English abstract).
[21] Foster I. Service-Oriented science. Science, 2005,308(5723):814−817.
[22] Huai JP, HU CM, Li JX, Sun HL, Wo TY. CROWN: A service grid middleware with trust management mechanism. Science in China (Series E), 2006,36(10):1127−1155 (in Chinese with English abstract).
[23] Mei H, Huang K, Zhao HY, Jiao WP. A software architecture centric engineering approach for Internetware. Science in China (Series E), 2006,36(10):1100−1126 (in Chinese with English abstract).
[24] Buyya R, Abramson D, Giddy J, Stockinger H. Economic models for resource management and scheduling in grid computing. Special Issue on Grid Computing Environments, Journal of Concurrency and Computation: Practice and Experience (CCPE), 2002,14(13-15):1507−1542.
[25] Dail H, Berman F, Casanova H. A decoupled scheduling approach for grid application development environments. Journal of Parallel and Distributed Computing, 2003,63(5):505−524.
[26] Du XL, Jiang CJ, Xu GR, Ding ZJ. A grid DAG scheduling algorithm based on fuzzy clustering. Journal of Software, 2006,17(11): 2277−2288 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/2277.htm
[27] Sodhi S, Subhlok J. Skeleton based performance prediction on shared networks. In: Moreira E, ed. Proc. of the 4th IEEE Symp. on Cluster Computing and the Grid (CCGrid 2004). Washington: IEEE Computer Press, 2004. 723−730.

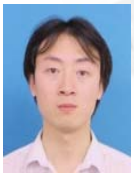
附中文参考文献:

[15] 丁箐,陈国良,顾钧.计算网格环境下一个统一的资源映射策略.软件学报,2002,13(7):1303−1308. http://www.jos.org.cn/1000-9825/13/1303.pdf
[16] 林伟伟,齐德昱,李拥军,王振宇,张志立.树型网格计算环境下的独立任务调度.软件学报,2006,17(11):2352−2361. http://www.jos.org.cn/1000-9825/17/2352.htm
[18] 张伟哲,方滨兴,胡铭曾,刘欣然,张宏莉,高雷.计算网格环境下基于多址协同的作业级任务调度算法.中国科学(E 辑),2006,36(10): 1240−1262.
[20] 卢锡城,王怀民,王戟.虚拟计算环境 iVCE:概念与体系结构.中国科学(E 辑),2006,36(10):1081−1099.
[22] 怀进鹏,胡春明,李建欣,孙海龙,沃天宇.CROWN:面向服务的网格中间件系统与信任管理.中国科学(E辑),2006,36(10): 1127−1155.
[23] 梅宏,黄罡,赵海燕,焦文品.一种以软件体系结构为中心的网构软件开发方法.中国科学(E辑),2006,36(10):1100−1126.
[26] 杜晓丽,蒋昌俊,徐国荣,丁志军.一种基于模糊聚类的网格 DAG 任务图调度算法.软件学报,2006,17(11):2277−2288. http://www.jos.org.cn/1000-9825/17/2277.htm



张伟哲(1976－),男,黑龙江哈尔滨人,博士,讲师,CCF 会员,主要研究领域为网络计算,网络安全.

何慧(1974－),女,博士,副教授,CCF 会员,主要研究领域为网络计算.



田志宏(1978－),男,博士,讲师,CCF 会员,主要研究领域为网络安全,网络计算.

刘文懋(1983－),男,硕士生,主要研究领域为网络计算.

张宏莉(1973－),女,博士,教授,博士生导师,CCF 会员,主要研究领域为并行计算,网络安全,拓扑发现.