ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.10, October 2009, pp.2766−2775
doi: 10.3724/SP.J.1001.2009.03506

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

网格服务资源多维性能聚类任务调度*

陈志刚, 杨 博+

(中南大学 信息科学与工程学院,湖南 长沙 410083)

Task Scheduling Based on Multidimensional Performance Clustering of Grid Service Resources

CHEN Zhi-Gang, YANG Bo+

(School of Information Science and Engineering, Central South University, Changsha 410083, China)

+ Corresponding author: E-mail: rodmanyy2002@163.com

Chen ZG, Yang B. Task scheduling based on multidimensional performance clustering of grid service resources. *Journal of Software*, 2009,20(10):2766−2775. http://www.jos.org.cn/1000-9825/3506.htm

Abstract: Grid computing is currently an important research area and task scheduling is a basal part of it. The performance of task scheduling directly affects grid QoS. A task scheduling algorithm based on multidimensional performance clustering of grid service resources, MPCGSR (task scheduling algorithm based on multidimensional performance clustering of grid service resources), is proposed for shortening the completion time of task scheduling and improving task scheduling performance. In the algorithm, combined with the theory of small world, the multidimensional performance clustering of service resources is executed in advance based on the hypergraph model of grid service resources constructed according to characteristics of grid resources such as its huge numbers, heterogeneity and multiplicity. Tasks are matched to clustering resources and scheduled. Simulation results show that it is an effective grid task scheduling algorithm that is superior to other kindred algorithms.
Key words: grid; clustering; task scheduling; hypergraph

摘 要: 网格计算是当前一个重要的研究领域,其中任务调度是一个基本组成部分,其性能直接影响到网格服务质量.为了缩短任务调度完成时间,提高任务调度性能,提出了一种网格资源多维性能聚类任务调度算法 MPCGSR (task scheduling algorithm based on multidimensional performance clustering of grid service resources).该算法根据网格环境下服务资源数量庞大、异构、多样的特点,预先以构建的网格服务资源超图模型为基础,结合小世界理论对服务资源进行多维性能聚类,将任务与聚类资源相匹配并实施调度.模拟实验结果表明,算法较之同类算法具有优越性,是一种有效的网格任务调度算法.
关键词: 网格;聚类;任务调度;超图
中图法分类号: TP393 **文献标识码**: A

当前互联网发展迅速,其分布范围日益扩大,包含资源日益增多.然而在如此巨大的资源中,相当一部分未得到充分的共享和利用,资源间相互独立,“信息孤岛”现象严重.网格计算的目的就是通过实现多种资源的全面

* Supported by the National Natural Science Foundation of China under Grant Nos.60573127, 60773012 (国家自然科学基金)
Received 2008-03-10; Revised 2008-05-05; Accepted 2008-10-17; Published online 2009-06-09

共享,充分利用资源能力以形成虚拟的超级计算机,从而大幅度提升性能来完成各种计算任务,为用户提供高质量的服务.任务调度是网格计算的一个基本组成部分,任务调度的性能是影响网格 QoS 的重要因素,其中,对网格资源的选择与分配是提高任务调度性能需要解决的关键问题之一.Foster 提出的开放网格服务体系架构(open grid service architecture,简称 OGSA)[1]确立了 Web 服务作为网格资源新的抽象形式和构造基础,在 OGSA 下,所有资源均被建模为服务.2004 年,IBM,Globus 联盟和 HP 共同提出了 OGSA 的核心规范 Web 服务资源框架(WS-resource framework,简称 WSRF)[2],对有状态的服务资源及相关概念进行了定义.网格服务资源丰富多样,将任务分配到最适合的服务资源才能得到最佳的计算结果.网格环境下的任务调度包括元任务(独立任务)的调度和依赖任务的调度,元任务独立于其他任务,而依赖任务之间存在先后依赖关系.由于网格环境的特点,主要考虑元任务的调度,目前已有的元任务调度算法包括最大最小算法(max-min)、最小最小算法(min-min)、最大时间跨度算法(max-int)、快速贪吃算法(fast-greedy)[3]等.Kaya 等人提出了一种文件共享独立任务的启发式调度算法[4],Jones 等人提出以带宽为中心的任务调度模型与启发式算法[5],Kumar 等人从最大化商业价值角度研究了网格任务调度策略[6].这些算法的选择对象大多为全体服务资源,在资源相当丰富的网格环境下,性能受到限制,不能充分发挥其优越性.因此,缩小任务调度时资源选择的范围、提高资源选择的精度、缩短任务资源匹配与调度的时间,有助于提高任务调度的效率.其中,对资源的聚类预处理是一种有效的方式.资源聚类主要包括基于处理资源性能的聚类和基于网络性能的资源聚类,现有的聚类方法有层次式聚类算法、*K*-均值聚类算法[7]及基于密度的聚类算法[8,9]等.Fiolet 等人提出了一种网格分布数据的聚类方法[10],杜晓丽等人针对 DAG 图提出了基于模糊聚类的任务调度启发式算法,为元任务的聚类调度提供了借鉴[11].为了直观而有效地实现资源的聚类,本文引入了超图理论[12].超图是图论的一个分支,研究多元子集问题或有限集中各元之间的多元关系,由 Berge 于 20 世纪 70 年代首次提出.使用超图的原因在于超图所具有的多元特性符合聚类的特点,超图除了具有一般图论的概念,还可以解决一些一般图论无法解决的问题,如超边的合并与分解等[13].图论中所有概念可推广到超图中,并可获得更简洁的表达和更强的结果,相比普通图具有显著的优越性.已有的超图应用成果包括数据挖掘中的超图聚类[14]、基于有向超图的工作流支持资源分配优化方法[13]、基于超图的图像压缩[15]等.本文建立服务资源的超图模型,通过超图的聚类来完成资源的性能聚类,建立高效的资源查找结构优化任务的调度.

本文为达到最小化完成时间等目标,提高网格 QoS,对资源建立超图模型,综合资源多维性能,并结合小世界理论[16]进行聚类.在此聚类基础上实现任务与资源的匹配与调度,提出了一种网格资源多维性能聚类任务调度算法(task scheduling algorithm based on multidimensional performance clustering of grid service resources,简称 MPCGSR).理论分析与实验表明,该算法提高了任务调度性能,与同类算法相比具有一定的优越性.

本文第 1 节建立任务与资源模型,描述相关概念.第 2 节对所提出的网格服务资源多维性能聚类任务调度算法进行阐述和分析,并在第 3 节中通过实验进行验证.第 4 节对全文进行总结并指出下一步研究方向.

1 调度模型

本文以 OGSA 为基础,将网格系统分为资源层、中间层、应用层 3 个层次,各层之间采用 OGSA 框架下的标准协议交互,既保持相对的独立性,又适合全球范围的资源组织,满足 Web 服务组织的自举性、分布性、无中心控制的要求以及较好的资源查找效率.应用层面向用户,在网格中间层的支持下,实现用户与服务提供者的交互机制,网格用户在任务调度过程中,通过应用层提交任务和接收任务执行结果.资源层是网格应用的基础,由主机、存储设备、大型仪器等各种资源构成,体现了网格环境的异构性、分布性和自治性特点.中间层主要解决跨组织域的共享与面向任务的资源协同[17],屏蔽网格资源层中计算资源的分布、异构特性,构建服务,向网格应用层提供透明、一致的使用接口,以支持网格应用的开发,对网格用户提供服务.本文的任务与服务资源模型及相应的任务调度机制在该层中实现.

1.1 任务模型

假设本文的任务均为计算任务,且只考虑元任务,任务之间相互独立,不考虑任务的跨资源节点执行,即用户提交的任务就是网格调度器进行任务分配的最小单位.任务模型描述如下:

$V=\{v_0,v_1,v_2,\ldots,v_{n-1}\}$代表任务集,$n=|V|$为任务集的大小,即任务数,$v_i(i\in[0,n-1])$表示第 i 个任务.$v_i=\{tID,tRr,tSta,tServ,tData\}$,各属性含义如下:

1) tID 为任务标识.
2) $tRr=\{tC_1,tC_2,\ldots,tC_k\}$代表任务的网格服务资源需求,$k$ 为该任务需求的资源能力数,$tC_j(j\in[0,k-1])$为完成该任务的资源应该具有的能力,能力级按由高到低顺序排列.
3) $tSta$ 为任务状态,$tSta=\{tFree,tAllo,tSche,tWait,tExec,tComp\}$.
4) $tServ$ 为任务调用的网格服务,包括服务名($tSName$)和服务方法($tSMethod$)及服务质量(tSQ)等.
5) $tData$ 为任务相关数据,包括任务的计算量 tC、输入数据 tI 以及输出数据 tO 等.

其中,$tSta$ 属性标记了任务在当前调度中所处的状态,方便任务的分配.$tFree$ 表示闲置状态,处于该状态的任务暂不能进行分配,待条件符合才能与资源进行匹配映射;$tAllo$ 表示待分配状态,当 $tFree$ 状态任务满足匹配允许条件后即转为该状态,可与资源进行匹配映射;$tSche$ 表示待调度状态,当 $tAllo$ 状态任务成功与资源匹配后转为该状态,可调度到物理资源;$tWait$ 表示等待状态,该状态任务已调度到物理资源,但处于等待队列中尚未执行;$tExec$ 表示执行状态,该状态任务正在执行尚未完成;$tComp$ 表示已完成状态,该状态任务已经执行完成.通过构建任务模型,将任务的特性与相关的服务联系起来,优化资源匹配过程.

任务的输入数据 tI 构成输入文件,输出数据 tO 构成输出文件,一个输入文件可以同时提供给一个或多个任务;而一个任务只有一个输出文件,在任务调度之前输入文件和输出文件的大小是已知的.

任务对服务资源的性能需求主要包括 3 个方面的能力:资源处理能力、通信能力和存储能力.我们设定 $tC_j(j\in[0,k-1])$由三元组$\langle Rr_{pro},Rr_{com},Rr_{sto}\rangle$构成,$Rr_{pro},Rr_{com},Rr_{sto}$ 分别代表任务对资源处理能力、通信能力和存储能力的需求.

1.2 资源模型

资源模型相关定义如下:

定义 1. 星 $RHG(rv_i)$:对 $rv_i\in RV,RHG$ 中所有包含 rv_i 的边所导出的部分超图称为以 rv_i 为心的星 $RHG(rv_i)$.

定义 2. 性能距离 rPD:资源节点间的多维性能指标的差值.其中,rPD_{cmin} 为两个属于不同聚类的节点之间的性能距离的最小值.

定义 3. 核心节点:即为一个聚类的中心参考节点.

定义 4. $rWLen(i)$:即匹配到资源 rv_i 等待执行的任务总计算量.

定义 5. AW:即预期等待时间,$AW(i,j)$表示任务 v_i 执行前在资源 rv_j 上的等待时间.

定义 6. AE:即预期执行时间,$AE(i,j)$表示任务 v_i 在资源 rv_j 上的期望执行时间.

定义 7. AC:即预期完成时间,是指任务 v_i 在资源 rv_j 上的期望完成时间为 $AC(i,j)$.

为了达到最小化完成时间的目标,根据网格服务资源的性能特性,建立服务资源超图模型,基于该模型对网格资源进行预处理,提高任务匹配资源的效率,从而优化任务调度过程.服务资源超图 RHG 描述如下:

服务资源超图 $RHG=\langle RV,RE\rangle$,其中:

(1) $RV=\{rv_0,rv_1,rv_2,\ldots,rv_{rn-1}\}$代表处理资源集,$rn=|RV|$称为资源超图的阶,即处理资源数,$rv_i(i\in[0,rn-1])$表示第 i 个处理资源,$rv_i=\{rID,rCap,rServ,rData,rType\}$,各属性含义如下:

1) rID 为资源标识.
2) $rCap$ 为资源能力,$rCap=\{rCLev,rPro,rCom,rSto,rThr\}$,其中:
   - $rCLev$ 为处理资源的能力级.
   - $rPro$ 为资源的处理能力,即资源单位时间的计算量,其值越大,表明该资源的处理速度越快,处理能力越强.
   - $rCom$ 为资源的平均通信能力,即与该资源相连的链路通信能力的均值.
   - $rSto$ 为资源的存储能力,即资源单位时间的存储数据量.
   - $rThr$ 为资源的处理阈值,$rThr=\{recT,refT\}$,其中,$recT$ 为接收任务阈值,$refT$ 为拒绝任务阈值,该值

指定了资源的负载限度,以保持负载平衡.

3) *rServ* 为资源提供的网格服务,包括服务名(*rSName*)和服务方法(*rSMethod*)以及服务质量(*rSQ*)等.

4) *rData* 为资源相关数据,包括 *rID*,*rOD* 等,*rID* 和 *rOD* 分别代表输入带宽与输出带宽.

5) *rType* 为资源类型,如大型机、微型机和工作站等.

(2) $RE=\{re_0,re_1,\ldots,re_{rm-1}\}$:表示 *RHG* 的超边集,$rm=|RE|$为 *RHG* 的超边数,超边 re_h 的权值 rEW_h 代表该超边所包含资源节点的平均性能.

$$rEW_h=\frac{\sum_{rv_i\in re_h} rGP_i}{reo_h} \tag{1}$$

其中,rGP_i 为 rv_i 的综合性能值,reo_h 为超边 re_h 的秩,即所包含的资源节点数,即 $reo_h=|re_h|$.

任务调度的结果主要受服务资源的计算能力、通信带宽和存储效率的影响,因此在本文中,资源性能主要从处理能力、通信能力和存储能力 3 方面能力来考虑.表示资源节点 rv_i 的综合性能的 rGP_i 公式为

$$rGP_i=rdp_i\times\sqrt{\frac{rp_1(rPro_i)^2+rp_2(rCom_i)^2+rp_3(rSto_i)^2}{rp_1+rp_2+rp_3}} \tag{2}$$

其中,$rPro_i$,$rCom_i$,$rSto_i$ 分别为 rv_i 的处理能力、通信能力和存储能力;rp_1,rp_2,rp_3 分别为 *rPro*,*rCom*,*rSto* 对应的经验系数,*rdp* 为该节点完成任务的可靠性参数,其计算公式为

$$rdp_i=dp_1\times rCR_i+dp_2\times rOL_i \tag{3}$$

其中,rCR_i 为该节点的任务完成率,rOL_i 为该节点的在线率,dp_1 和 dp_2 分别为 rCR_i 和 rOL_i 对应的经验系数.

因此,两个资源节点 rv_i 和 rv_j 之间的性能距离 $rPD_{i,j}$ 为

$$rPD_{i,j}=|rGP_i-rGP_j| \tag{4}$$

其中,rGP_i 和 rGP_j 分别为 rv_i 和 rv_j 的综合性能.

通过比较资源超图 $RHG=\langle RV,RE\rangle$中各超边的权值,即性能距离,将权值小于指定阈值的超边合并,而权值大于阈值的超边则分解成不同的超边,形成聚类超图 $RCHG=\langle RV,RCE\rangle$,其超边 $RCE=\{rce_1,rce_2,\ldots,rce_k\}$,其对应的权集为 $PW=\{PW_1,PW_2,\ldots,PW_k\}$,该聚类超图具有以下几个特征:

(1) $P_i\cup P_j=V;P_i\cap P_j=\varnothing(i,j=1,2,\ldots,k-1,i\neq j)$.

(2) P_k 是由前 $k-1$ 条超边中的核心节点构成的超边,是资源聚类超图 *RCHG* 的一个横贯,即

$$P_k\cap P_i\neq\varnothing(i=1,2,\ldots,k-1).$$

(3) 前 $k-1$ 条超边之间相关程度小,即超边之间的性能距离 $EPD_{i,j}$ 大于阈值.

$$EPD_{i,j}=|rEW_i-rEW_j| \tag{5}$$

其中,$EPD_{i,j}$ 表示超边 re_i 与超边 re_j 之间的性能距离,rEW_i 和 rEW_j 分别为超边 re_i 和 re_j 的性能权值.

(4) 前 $k-1$ 条超边内节点间相关程度大,即节点间性能距离小于阈值 rPD_{cmin}.

(5) 第 k 条超边包含其余各超边的核心节点,成为聚类超图 *RCHG* 的一个控制集.

聚类模型通过超边形成了基于多维性能特性的服务资源查找路径,有效地提高了资源查找效率.

任务与服务资源模型的建立,在任务与服务资源之间建立了松耦合的联系,MPCGSR 算法在此基础上根据两者的特性实现动态匹配,从而达到提高任务调度性能的目的.

2 调度算法

2.1 算法描述

由于网格具有广域性,在任务调度过程中,一次性地针对整个网格系统的资源进行处理复杂度较高,因此,在调度之前以网格自治域为单位基于服务资源超图模型聚类,形成各自治域的聚类超图 *RCHG*.自治域聚类算法如下:

AClus(){

初始超图为以本自治域内各资源节点 rv_i 为核心的星 $RHG(rv_i)$构成的超图;
集合 *CClus* 初始包含所有超图节点;
while (*CClus* 中存在资源节点){
选择 *CClus* 中度最大的一个资源节点 rv_j 作为核心节点,加入核心节点超边 *CoreEdge*;
将与 rv_j 的性能距离大于 rPD_{cmin} 的节点从星 $RHG(rv_j)$中分离;
将星 $RHG(rv_j)$所含节点从 *CClus* 中分离;
星 $RHG(rv_j)$成为一个新的以 rv_j 为核心节点的聚类;
}
}

通过各自治域分别执行 AClus 方法聚类,将各自治域内同类网格服务资源组织形成自治域内的服务资源组织超树(service resource organizing hypertree,简称 SROH).各自治域之间的同类聚类通过包含核心节点的超边相连通,整个网格系统的聚类形成一个层次结构,顶层即为包含各自治域聚类核心节点的超边,构成整个网格系统的 SROH.通过该组织超树,网格系统能够服务查找与定位得到服务资源的地址,并且能够获得全局的负载均衡,它记录了各自治域中的资源情况与负载信息,当有任务申请资源时,通过它将当前最适合的服务资源匹配给任务.

以资源聚类为基础,MPCGSR 算法的任务调度过程如下:

MPCGSR(){
//相关参数初始化,包括对 *V* 中任务按计算量降序排列
Initial();
for (*V* 中每一个任务 v_i){
if (v_i 有指定的资源需求){
通过 SROH 的超边由顶向下搜索到性能指标与需求相符合的聚类;
if (无满足要求的聚类资源)
将任务 v_i 从 *V* 中删除;
if (满足要求的聚类数>1)
在这些聚类中选择位于负载较轻的自治域的聚类;
将 v_i 调度到该聚类中综合性能 *rGP* 最佳的有效资源 rv_j 上;
}
else {
通过 SROH 的超边由顶向下搜索到具有有效资源的性能较高的聚类;
在这些聚类中选择负载较轻的自治域的聚类;
将 v_i 调度到该聚类中综合性能 *rGP* 最佳的有效资源 rv_k 上;
}
}
}

上述算法中,首先初始化任务调度的相关参数,包括对 *V* 中任务按计算量降序排列,赋予调度优先级,建立调度模型,然后以聚类结果为基础,将任务与资源相匹配并进行调度执行.由算法可见,具有预先指定服务资源需求的任务直接与符合该要求的聚类相匹配,并从聚类中找到性能最好且负载较轻的资源并映射到该资源;而没有具体资源需求要求的任务首先找到具有有效资源的性能较高的聚类,然后从聚类中找到性能最好且负载较轻的资源并映射到该资源.对于满足要求的多个聚类,优先选择负载低的进行匹配,从而有利于网格资源的负载均衡.由于以资源聚类为基础,且聚类按性能排序,因而能够有效地缩短资源查找的过程,从而缩短了任务调度的过程.

2.2 算法性能分析

在网格任务调度过程中,任务数据传输的通信开销为

$$LT(i,j)=\frac{tDI_i}{rCom_j}=\frac{tI_i+tO_i}{rCom_j} \tag{6}$$

其中,$LT(i,j)$表示任务 v_i 在资源 rv_j 上的通信开销,tDI_i 表示任务 v_i 的传输信息量,包括任务 v_i 在资源 rv_j 上的输入数据量和输出数据量,$rCom_j$ 为传输链路的通信能力.

任务在资源上的预期执行时间为

$$AE(i,j)=\frac{tC_i}{rPro_j} \tag{7}$$

其中,$AE(i,j)$表示任务 v_i 在资源 rv_j 上的预期执行时间,tC_i 为任务 v_i 的计算量,$rPro_j$ 为资源 rv_j 单位时间的计算量.$AE(i,j)=\infty$表示任务 v_i 不能在资源 rv_j 上执行.

在任务执行过程中会产生各种数据,包括中间数据、最后结果等,这些数据需要进行相应存储,因此存储时间为

$$AS(i,j)=\frac{tSD_i}{rSto_j} \tag{8}$$

其中,$AS(i,j)$表示任务 v_i 在资源 rv_j 上的数据存储时间,tSD_i 为任务 v_i 需存储的数据量,$rSto_j$ 为资源 rv_j 单位时间的存储量.$AS(i,j)=\infty$表示任务 v_i 不能在资源 rv_j 上存储.

因此,任务 v_i 在资源 rv_j 上的预期完成时间 $AC(i,j)$为

$$AC(i,j)=LT(i,j)+AE(i,j)+AE(i,j)=\frac{tDI_i}{rCom_j}+\frac{tC_i}{rPro_j}+\frac{tSD_i}{rSto_j} \tag{9}$$

由式(9),任务调度的主要目标是最小化完成时间 $\min\limits_{v_i\in V,rv_j\in RV}\{AC(i,j)\}$,即 $\min\limits_{v_i\in V,rv_j\in RV}\left\{\frac{tDI_i}{rCom_j}+\frac{tC_i}{rPro_j}+\frac{tSD_i}{rSto_j}\right\}$.

由此可见,除了资源的处理能力以外,最优目标还体现在通信带宽和数据存储的利用效率上.因此,与许多算法只考虑资源处理能力有所不同,在公式(2)中,我们主要考虑了资源的 3 项主要能力——处理能力、通信能力和存储能力,分别用 $rPro$,$rCom$,$rSto$ 表示.综合考虑资源的这 3 方面能力,能够更准确地估计任务的完成时间,选择更为匹配的资源来执行任务.对应于资源这 3 项能力,任务相应的需求分别为 Rr_{pro},Rr_{com} 和 Rr_{sto}.我们设定如下约束条件:

$$rPro\ge Rr_{pro},\ rCom\ge Rr_{com},\ rSto\ge Rr_{sto}.$$

该式表示了用户提交任务所需求的资源能力必须满足的约束条件,是资源允许分配给任务的必要初始条件,在满足该约束条件的资源的基础上再作进一步择优选择.

在用户提交的任务中,不同任务对不同资源的需求量不同,对资源的各项能力的需求也不同.针对资源的 3 项能力,我们设定对应的经验系数来定量地表示对各项能力的需求,在公式(2)中表示为 rp_1,rp_2,rp_3.基于经验系数,对 3 项能力进行加权平均计算,能够有效地量化资源综合性能,同时便于计算资源之间的性能距离,提高资源性能聚类的效率.另外,面向网格环境的动态性,我们在公式(2)中加入了可靠性参数 rdp.历史统计信息往往能够反映事件的发展趋势,我们从资源节点的历史情况出发,通过统计资源节点的任务完成率与在线率来计算资源节点的可靠性参数.rdp 由公式(3)计算获得.在公式(3)中,rCR_i 为资源节点的任务完成率,rOL_i 为该节点的在线率,dp_1 和 dp_2 分别为 rCR_i 和 rOL_i 对应的经验系数.rCR_i 即为在该节点上按时完成的历史任务数占该资源接收任务历史总数的比值,表明了完成任务的历史情况.任务完成率越高,按时完成新提交任务的可能性越大,该资源也越可靠.rOL_i 为处理资源处于有效状态的时间占总时间的比值,在线率越高,该资源越可靠.由于网格环境的动态多变,资源的不稳定情况直接影响到任务的顺利执行,选择了不可靠资源可能会使任务执行进度大大延期,因此,将资源的可靠性作为考虑的指标之一.

通过以上方法,将任务需求与资源特性联系在一起,有助于任务与资源间的合理匹配与调度.公式(2)将资源

的多维性能影响因子进行综合计算,通过 AClus 方法将资源按综合性能聚类,网格服务资源多维性能聚类任务调度算法基于该聚类进行任务调度,符合任务调度的优化目标.

AClus 方法以超图的星为单位实现网格自治域的资源聚类.方法中选择当前聚类集合中度最高的节点作为核心节点符合无尺度特性与小世界特性.研究表明,Internet 等大型网络的节点度分布服从无尺度(scale-free)[18]特性和小世界(small world)[16]特性.无尺度特性是指网络服从幂规律 $p(d)\sim d^{-\tau}$,其中分布函数 $p(d)$用来表示节点度的分布,即一个任意节点正好有 d 条边的概率,$1<\tau<\infty$.无尺度特性直观地体现为网络被少数连接度较大的节点所支配,而多数节点的度较低.通过少数连接度大的节点查找信息效率较高.小世界特性是指网络拓扑具有高聚集度和低特征路径长度的特性,节点之间的平均距离随远程连接个数的增加呈指数级下降.在小世界网络中,平均距离 $L\sim\ln(N)/\ln(D)$,其中,L 为平均距离,N 为节点数,D 为节点度数.小世界概念说明,在大多数网络中,尽管其规模很大,但任意两个节点之间总有一条相当短的路径.小世界最为通用的表现形式是"六度分离"概念,同时,小世界网络具有相对较高的集群系数[19].连接度高的节点代表了该性能层次的典型水平.由这些自组织特性可知,连接度高的节点适合作为聚类的核心节点.因此,通过选择当前待聚类集合中度最高的节点作为核心节点,可以提高不同性能水平的节点聚类的准确程度,减少迭代次数,且能提高聚类中节点的查找效率.

根据性能距离阈值,AClus 方法在选择核心节点的同时进行性能聚类,在最坏情况下的时间复杂度为 $O(mk)$,其中,m 为自治域内资源数,k 为聚类数.由于每次聚类即将该聚类中资源从待选资源集中排除,减少了后继循环的选择范围,从而降低了聚类的时间复杂度,优于同类算法.通过在生成的 RCHG 中对核心节点超边 CoreEdge 中的节点按性能降序排序,实现了对各核心节点所在的聚类超边性能的降序排序.由于自治域节点数远小于整个网格的节点数,各自治域资源聚类并行进行,极大地缩短了网格聚类的总执行时间.

MPCGSR 算法在资源性能聚类的基础上进行任务调度.通过聚类得到的 SROH,调度算法可以快速查找到性能匹配的资源并将任务调度执行,在最坏情况下的时间复杂度为 $O(nm)$,其中,n 为任务数,m 为自治域内资源数.通过对服务资源的超图聚类,任务调度不必每次都检测所有的资源,算法根据聚类结果优先选择综合性能最合适的资源分配给当前任务.由于资源超图聚类过程是进行任务调度前的准备工作,当有大量同类任务在网格系统上计算时仅需执行 1 次,明显降低了算法的 Makespan,有效提高了任务调度的执行效率.

3 实 验

3.1 实验内容与设置

我们将 MPCGSR 算法与同类元任务调度算法 Min-Min 以及 KPB(k-percent best)[20]算法分别进行模拟调度实验进行比较.实验环境为 PIV 3GHz,2G 内存,操作系统为 Windows XP.模拟仿真系统主要由任务提交系统、网格资源环境以及调度器 3 部分构成.

在模拟仿真系统中,由任务提交系统生成任务.任务相关参数包括任务节点个数、任务计算量等.任务以 Possion 分布到达.仿真系统包含不同类型的资源实体,不同带宽的链路将资源相连接,构成了分布、异构的网格环境.相关参数包括网格拓扑结构(包括完全连接与任意连接)、资源类型、性能值、个数及异构率等,拓扑节点分布满足 small world 网络和幂规律特征.调度器根据调度算法负责将任务映射并调度到相应网格资源执行.

在本实验中,我们考察 250~1 500 个服务资源组成的网格系统对 200~2 000 个任务的调度情况.实验通过一些控制参数对任务和资源的构成进行调节,生成一批独立的任务请求和资源节点,以下是几个主要控制参数:

hp_{task}:任务差异度.任务差异度体现的是任务计算量的差别大小.任务计算量的取值为$[TaskH_{\min}, hp_{task}\times TaskH_{\min}]$之间的任意一个随机数,其中,$TaskH_{\min}$为任务计算量的最小值.

hp_{pro}:资源处理能力差异度.资源处理能力差异度体现的是网格资源间处理能力的差别大小.资源处理能力的取值为$[ProH_{\min}, hp_{pro}\times ProH_{\min}]$之间的任意一个随机数,其中,$ProH_{\min}$为网格中资源处理能力的最小值.

hp_{com}:通信能力差异度.这一数值体现的是网格资源间连接带宽的差别大小.资源之间网络通信能力的取值为$[ComH_{\min}, hp_{com}\times ComH_{\min}]$之间的任意一个随机数,其中,$ComH_{\min}$为网格中资源通信能力的最小值.

hp_{sto}:存储能力差异度.资源存储能力差异度体现的是网格资源间存储能力的差别大小.资源存储能力的取

值为[$StoH_{min}$,hp_{sto}×$StoH_{min}$]之间的任意一个随机数,其中,$StoH_{min}$ 为网格中资源存储能力的最小值.

通过调节控制参数,我们模拟各种不同情况下的任务与网格服务资源环境,同时通过服务资源的随机失效等方法,体现网格环境的动态性.

3.2 实验结果与性能分析

优化 Makespan 是任务调度的主要目标,因此,实验将不同网格服务资源环境下任务完成的平均调度长度 Makespan 作为考察算法性能的主要指标.实验结果如图 1 所示.

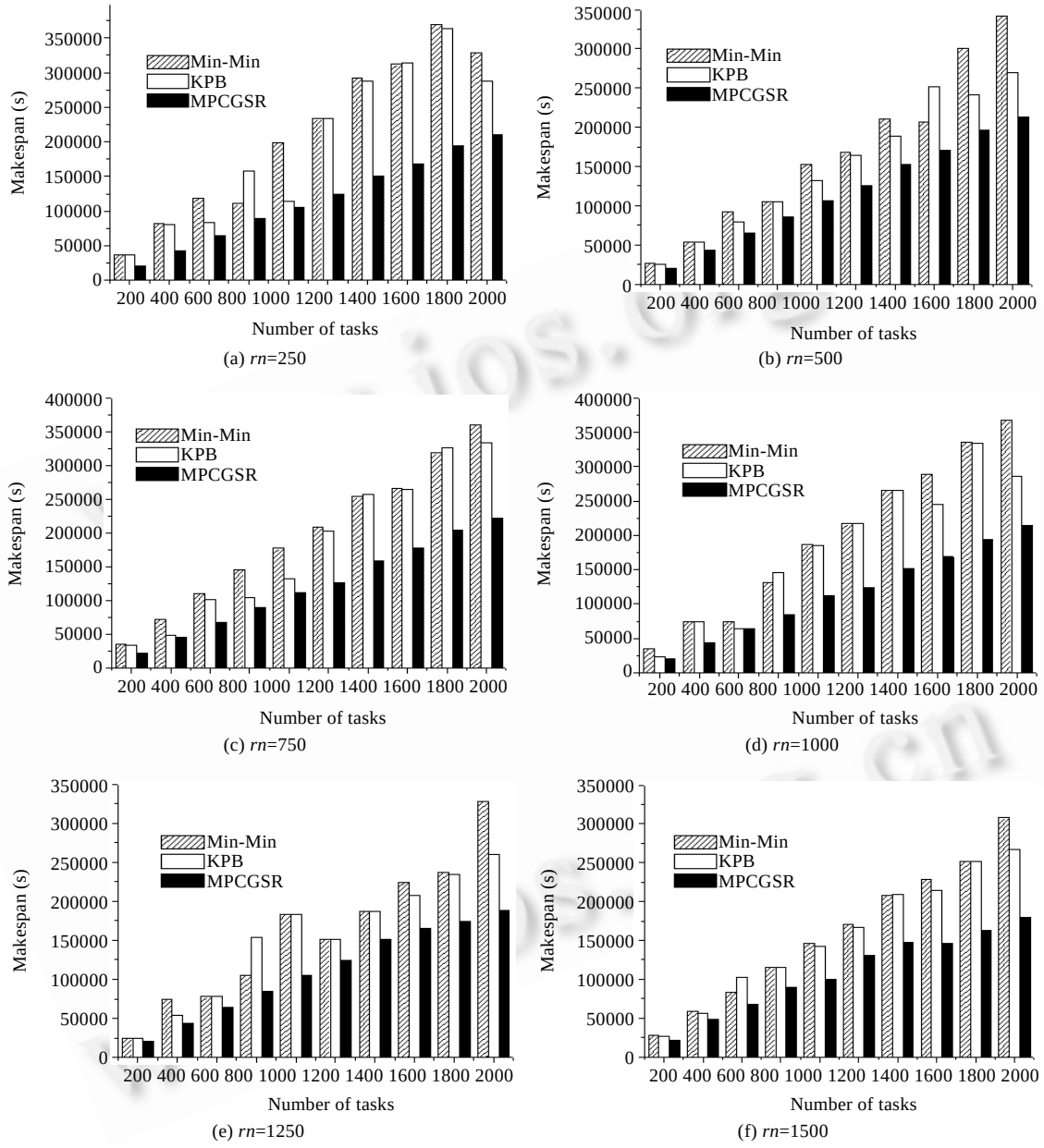

Fig.1 Makespan comparison

图 1 Makespan 比较

其中,*rn* 表示服务资源数,图 1(a)~图 1(f)分别为不同服务资源环境下 3 种算法完成任务的情况,其资源数逐步递增,步长为 250,分别为 250,500,750,1 000,1 250,1 500 个服务资源.在各子图中,各种算法获得的 Makespan 均随着任务数的增多而明显上升,其中,Min-Min 算法与 KPB 算法获得的 Makespan 相差不大,KPB 算法略优于 Min-Min 算法,MPCGSR 算法的调度结果明显优于 Min-Min 算法与 KPB 算法,体现出其更优越的调度性能.

4 结束语

本文面向任务调度,基于 OGSA 建立任务模型和网格服务资源超图模型,根据资源节点的多维性能特性并结合小世界理论进行超图聚类,在聚类资源基础上完成任务的映射与调度,提出了一种网格服务资源多维性能聚类任务调度算法 MPCGSR.理论分析与实验结果表明,算法通过资源超图聚类,有效提高了任务与资源的匹配效率,缩短了任务完成时间,提高了任务调度的性能.下一步工作的重点是对调度模型在安全性、动态性等方面作进一步完善和扩展,并在模型基础上完善现有调度算法以及提出新的更有效的调度策略,提高网格 QoS.

References:

[1] Foster I, Kishimoto H, Savva A, Berry D, Djaoui A, Grimshaw A, Horn B, Maciel F, Siebenlist F, Subramaniam R, Treadwell J, Von Reich J. The open grid services architecture. 2005. http://www.ogf.org/documents/GWD-I-E/GFD-I.030.pdf

[2] Czajkowski K, Ferguson DF, Foster I. The WS-resource framework. 2004. http://www.globus.org/wsrf/specs/ws-wsrf.pdf

[3] Gui XL. Grid Computing Technology. Beijing: Beijing University of Posts and Telecommunications Press, 2005. 144−169 (in Chinese).

[4] Kaya K, Ucar B, Aykanat C. Heuristics for scheduling file-sharing tasks on heterogeneous systems with distributed repositories. Journal of Parallel Distributed Computing, 2007,67(3):271−285.

[5] 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.

[6] Kumar S, Dutta K, Mookerjee V. Maximizing business value by optimal assignment of jobs to resources in grid computing. European Journal of Operational Research, 2009,194(3):856−872.

[7] Agrawal A, Casanova H. Clustering hosts in P2P and global computing platforms. In: Proc. of the 3rd IEEE/ACM Int'l Symp. on Cluster Computing and the Grid (CCGRID 2003). Tokyo: IEEE Computer Society, 2003. 367−373. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1199389

[8] Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis E, Han JW, Fayyad UM, eds. Proc. of the 2nd Int'l Conf. on Knowledge Discovery and Data Mining. Portland: AAAI Press, 1996. 226−231.

[9] Zhang WZ, Hu MZ, Liu KP. Computing grid hosts clustering based on network performance. Journal of Computer Research and Development, 2004,41(12):2135−2140 (in Chinese with English abstract).

[10] Fiolet V, Toursel B. A clustering method to distribute a database on a grid. Future Generation Computer Systems, 2007,23(8): 997−1002.

[11] 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

[12] Berge C, Wrote; Bu YH, Zhang KM, Trans. Hypergraph: Combinatorics on Finite Sets. Nanjing: Southeast University Press, 2002 (in Chinese).

[13] Sun XD, Xu XF, Wang G. Resource allocation balancing of workflow base on directed hypergraph. Acta Electronica Sinica, 2005, 33(8):1370−1374 (in Chinese with English abstract).

[14] Wang B, Zhang MW, Zhang B, Wei WJ. An effective hypergraph clustering in multi-stage data mining of traditional Chinese medicine syndrome differentiation. In: Proc. of the 6th IEEE Int'l Conf. on Data Mining—Workshops (ICDMW 2006). 2006. 848−852. http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F4063580%2F4063581%2F04063744.pdf&authDecision=-203

[15] Gillibert L, Bretto A. Hypergraphs for near-lossless volumetric compression. In: Proc. of the Int'l Conf. on Information Technology (ITNG 2007). Washington: IEEE Computer Press, 2007. 229−233. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4151689

[16] Newman MEJ. Models of the small world. Journal of Statistical Physics, 2000,101(3-4):819−841.

[17] Hu CM, Huai JP, Sun HL. Web service-based grid architecture and its supporting environment. Journal of Software, 2004,15(7): 1064−1073 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/1064.htm

[18] Goh KL, Oh E, Jeong H, Kahng B, Kim D. Classification of scale free networks. Proc. of the National Academy of Sciences of the United States, 2002,99(20):12583−12588.

[19] Dong P, Zhu PD, Lu XC. A mathematical model for network self-organized evolvement. Journal of Software, 2007,18(12): 3071−3079 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/3071.htm

[20] Maheswaran M, Ali S, Siegel HJ, Hensgen D, Freund RF. Dynamic mapping of a class of independent tasks onto heterogeneous computing systems. Journal of Parallel and Distributed Computing, 1999,52(2):107−131.

附中文参考文献:

[3] 桂小林.网格技术导论.北京:北京邮电大学出版社,2005.144−169.

[9] 张伟哲,胡铭曾,刘凯鹏.基于网络性能的计算网格主机聚类.计算机研究与发展,2004,41(12):2135−2140.

[11] 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

[12] Berge C,著;卜月华,张克民,译.超图——有限集的组合学.南京:东南大学出版社,2002.

[13] 孙雪冬,徐晓飞,王刚.基于有向超图的工作流资源分配均衡优化方法.电子学报,2005,33(8):1370−1374.

[17] 胡春明,怀进鹏,孙海龙.基于 Web 服务的网格体系结构及其支撑环境研究.软件学报,2004,15(7):1064−1073. http://www.jos.org.cn/1000-9825/15/1064.htm

[19] 董攀,朱培栋,卢锡城.一种网络自组织演化的数学模型.软件学报,2007,18(12):3071−3079. http://www.jos.org.cn/1000-9825/18/3071.htm

陈志刚(1964－),男,湖南益阳人,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机网络,数据库技术.

杨博(1979－),男,博士生,CCF 学生会员,主要研究领域为网格计算,网络与分布式计算.