

基于并行基因表达式编程的 网格资源分配算法

邓 松,王汝传,张 羽,张建风

(南京邮电大学计算机学院,江苏南京 210003)

摘 要: 网格下的资源分配属于 NP-难问题.为了更好地解决这个问题,文中首先提出了网格资源分配模型,并对资源个数与任务个数的三种不同情况进行了详细的分析,最后提出基于并行基因表达式编程的网格资源分配算法(Grid Resource Allocation Algorithm based on Parallel GEP, GRA-PGEP).该算法采用了基于资源与任务相关的非线性的编码方式和反转操作,同时应用粗粒度模型设计了该算法.仿真实验表明,GRA-PGEP 算法在优化成功率、平均收敛代数以及耗时方面都要优于传统的 GEP 和 GA 算法.

关键词: 基因表达式编程; 网格; 资源分配; 粗粒度模型

中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2009) 02-0272-06

Grid Resource Allocation Algorithm Based on Parallel Gene Expression Programming

DENG Song, WANG Ru-chuan, ZHANG Yu, ZHANG Jian-feng

(School of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China)

Abstract: Resource allocation of grid is part of optimization and NP-hard problem. In order to optimize resource allocation of grid, in the present research, it proposes a model of grid resource allocation, analyzes three different situations of the number of resources and tasks in detail, and then puts forward on a new algorithm which is called Grid Resource Allocation Algorithm based on Parallel GEP (GRA-PGEP). It adopts a nonlinear code based on resources and tasks and inversion operation, meanwhile, a coarse-grained model is applied to design the GRA-PGEP algorithm. By simulation experiment, it is showed that optimization successful rate, average convergent generation and consumptive time of GRA-PGEP have the advantage over traditional GEP and GA.

Key words: gene expression programming; grid; resource allocation; coarse-grained model

1 引言

目前,网格计算^[1]技术已经成为高性能计算领域发展的重要趋势,它的概念可以简单描述为:在动态变化的诸多异构环境中,共享资源和协同解决问题,目前已经被应用于科学计算以及系统集成等方面.网格计算的一个关键问题就是如何实现网格下的各种资源和任务调度的优化,使得资源的利用率能够达到最佳.显然,网格环境下的资源分配属于组合优化范畴,是一个 NP-难问题.对于解决 NP 问题,大部分情况下,利用传统的启发式搜索算法可以寻找问题的最优解,但是这类算法通常又难以避免局部最优的问题.

为了更好地解决这个问题,考虑到基因表达式编程^[2~5](Gene Expression Programming, GEP)是一种较好的

解决 NP 问题^[6]的算法,它具有天然并行性的特点,并且利用 GEP 的反转遗传操作可以较好地避免算法的局部最优,同时结合网格资源的分配特点,本文提出了基于并行 GEP 的网格资源分配算法(Grid Resource Allocation Algorithm based on Parallel GEP, GRA-PGEP).通过仿真实验表明,GRA-PGEP 算法在技术上是可行的,而且相对于传统的启发式搜索算法,大大提高了资源分配的质量,同时也克服传统启发式算法的局部最优问题.

本文所作的贡献如下:(1) 对于少资源多任务型,为了更好地处理提交给网格的任务,提出了基于 K 均值的任务聚类(Job Cluster based upon K-means, JC-Kmeans);(2) 对于多资源少任务型,提出了基于时间-费用最优化的网格资源选择算法(Grid Resource Selection based on Time-Cost Optimization, GRS-TCO);(3) 提出了基

收稿日期:2008-01-23;修回日期:2008-03-26

基金项目:国家自然科学基金(No. 60573141, 60773041);国家 863 高技术研究发展计划(No. 2006AA01Z201, No. 2006AA01Z219, No. 2006AA01Z439, No. 2007AA01Z404, No. 2007AA01Z478);江苏省高技术研究计划(No. BG2006001);南京市高科技项目(2007 软资 127);现代通信国家重点实验室基金(No. 9140C1105040805);江苏省高校科技创新计划(No. CX08B-085Z, CX08B-086Z)

于并行 GEP 的网格资源分配算法(Grid Resource Allocation Algorithm based on Parallel GEP, GRA-PGEP);(4) 通过仿真实验表明,相比与传统的 GEP 和 GA 算法, GRA-PGEP 算法的收敛速度要快,优化成功率要高且耗时少.

2 基于并行 GEP 的网格资源分配算法

2.1 相关背景

现有的网格环境下集中式的资源管理和调度策略在网格环境中暴露出扩展性不好、资源负载不平衡等问题,而且不能保证任务的服务质量,造成了网格资源的浪费.传统的基于遗传算法和别的启动发算法的网格资源调度策略,通过一定的启发式操作来不断优化资源任务的匹配,但资源匹配任务的效率和成功率不高,而且容易陷入局部最优,从而导致求解失败.

基因表达式编程^[7-10](Gene Expression Programming, GEP)是遗传计算家族的革命性的新成员,是借鉴生物遗传的基因表达规律提出的知识发现新技术.利用 GEP 较强的随机搜索能力以及较好的遗传操作来求解网格资源任务匹配,可以使得种群不断向着全局最优解逼近,从而最终求解得到最佳的资源任务匹配序列.

2.2 问题描述

假设现在有 n 个任务, m 个资源,网格下的资源分配就是利用某种策略将 n 个任务映射到 m 个异构资源上,并且使得 m 个资源处理 n 个任务的总时间最小,这样可以提高网格资源的利用率.为了更好地描述资源处理任务的时间代价,文中采用了时间-费用矩阵的形式进行描述.

定义 1 时间-费用矩阵(Time-Cost Matrix) 设当前网格有 n 个任务, m 个可用的异构资源且每一个资源在某一段时间内只能处理一个任务,资源 i 处理任务 j 的时间为 c_{ij} ,其中 $i \in [1, m], j \in [1, n]$,那么每个

c_{ij} 就构成了一个矩阵 $C_{m \times n} = \begin{bmatrix} c_{11} & \dots & c_{1n} \\ \dots & \ddots & \dots \\ c_{m1} & \dots & c_{mn} \end{bmatrix}$,称矩阵

$C_{m \times n}$ 为时间-费用矩阵.

通常情况下,对于网格资源分配问题分为三种形式:(1)当 $n = m$ 时,表示一个资源只能处理一个任务,也既是资源与任务相同型;(2)当 $n > m$ 时,表示一个资源可以处理多个任务,也既是少资源多任务型;(3)当 $n < m$ 时,表示一个任务可以由多个资源来处理,也既是多资源少任务型.

网格资源分配问题是一个求极小值问题^[11].为了便于问题的求解,我们引入变量

$$x_{ij} = \begin{cases} 1, & \text{资源 } i \text{ 处理任务 } j \\ 0, & \text{否则} \end{cases}$$

则该网格资源分配问题就等价于:

$$\min_{i=1, j=1}^{m, n} c_{ij}x_{ij} \quad (1)$$

下面就网格资源分配的三种情况进行讨论:

(1)当 $n = m$ 时,也即当资源与任务个数相同时构造如下的约束条件:

$$\begin{cases} x_{ij} = 1, i = 1, 2, \dots, n \\ x_{ij} = 1, j = 1, 2, \dots, n \\ x_{ij} = 0 \text{ 或 } 1, j = 1, 2, \dots, n \end{cases} \quad (2)$$

第一个约束表示第 i 个资源只能处理一项任务,第二个约束表示第 j 个任务只能由一个资源来处理.此时把网格资源分配问题简化为多约束条件的极小值求解问题.而且当 $n = m$ 时,有:

性质 1 若从时间-费用矩阵 $C_{m \times n}$ 的每行(列)各元素中分别减去该行(列)的最小元素,得到的新矩阵 $C_{m \times n} = (c_{ij})$,则新矩阵求得的最优解和原矩阵求得的最优解相同,解的目标函数值只相差一个常数.

证明 因为新矩阵 $C_{m \times n}$ 是由作为系数矩阵的时间-费用矩阵 $C_{m \times n} = (c_{ij})$ 中的每行(列)各元素分别减去该行(列)的最小元素得到的,而这种矩阵变化过程并没有影响到式(2)所表达的约束方程组中的 x_{ij} 的值,故由新矩阵 $C_{m \times n}$ 求得的最优解和原矩阵求得的最优解相同.

下面证明解的目标函数值只相差一个常数.对于特定的最优解向量 X ,代入式(1)且令 $F_1(X) = C_{m \times n}X$, $F_2(X) = C_{m \times n}X$, $U_{n \times n}$ 为时间-费用矩阵 $C_{m \times n}$ 中由每一行最小值所构成的矩阵, $V_{n \times n}$ 为时间-费用矩阵 $C_{m \times n}$ 中由每一列最小值所构成的矩阵.则由题意可知 $C_n \times n = C_n \times n - U_n \times n - V_n \times n$,从而 $F_1(X) - F_2(X) = C_n \times n \cdot X - C_n \times n \cdot X = (C_n \times n - C_n \times n) \cdot X = (U_n \times n + V_n \times n) \cdot X = C$,其中 C 为一常数.

原性质得证.

(2)当 $n > m$ 时,也即当少资源多任务时

当网格中可用资源的数目少于向网格中提交的任务数时,就会出现某几个任务提交给某一个资源进行处理的情况,那么任务之间可以按照加权公平排队规则进行排队等候,以便资源能够及时公平处理这些任务.为了方便使用 GEP 对资源任务进行编码,本文提出了基于 K 均值的任务聚类(Job Cluster based upon K-means, JC-Kmeans).

网格中的任务都有一些任务元数据进行任务本身的描述,在使用 K 均值聚类算法时,首先对描述任务的关键元数据进行量化,形成待聚类的任务序列,然后计

算各自的 z -分值,最后再进行聚类。

定义 2 任务 n -元数据序列 (n-Metadata Sequence of Job) 设 $x_{1j}, x_{2j}, \dots, x_{nj}$ 是任务 j 量化后的 n 个元数据值,则向量 $X_j = (x_{1j}, x_{2j}, \dots, x_{nj})$ 称为任务 j 的 n -元数据序列。

例 1 对于网格中的一个 BP 神经网络算法任务, BP 算法任务本身有一个 ID(惟一的标识符),BP 算法任务的其它一些参数,如:输入层节点数 In,隐层节点数 Hide,输出层节点数 Out,算法精度 Precision,迭代次数 Num 以及 BP 算法服务所在的 URL 等。对于这些参数通过量化后组成的数据序列 $X = (ID, In, Hide, Out, Precision, Num, URL)$ 称之为 BP 神经网络算法任务的七-元数据序列。

在通过 K 均值聚类时,需要计算每一个数据实例到聚类中心的距离,文中采用任务 z -分值的形式进行计算。

定义 3 任务 z -分值 (z-Division Value of Job)

设 $s_j = \frac{1}{n} (|x_{1j} - m_j| + |x_{2j} - m_j| + \dots + |x_{nj} - m_j|)$ 为任务 j 的绝对偏差均值,其中 $x_{ij}, i \in [1, n]$ 为任务 j 量化后的元数据值, $m_j = (x_{1j}, x_{2j}, \dots, x_{nj}) / n$ 为任务 j 的均值,则称 $z_{ij} = \frac{x_{ij} - m_j}{s_j}, i \in [1, n]$ 为任务 j 的 z -分值。

整个算法的形式化描述如算法 1 所示:

算法 1 基于 K 均值的任务聚类 (Job Cluster based upon K-means, JC-Kmeans)

Input: 待聚类的任务序列 $JobQueue (J_1, J_2, \dots, J_n)$;

Output: 聚类后的任务序列 $ClusterJobQueue (J_1, J_2, \dots, J_m)$;

Algorithm JC-Kmeans (int m , String[] $JobQueue (J_1, J_2, \dots, J_n)$);

Begin{

 选择资源个数 m 为聚类个数;

 在待聚类任务序列 $JobQueue (J_1, J_2, \dots, J_n)$ 中随机选择 m 个任务作为初始聚类中心;

 根据量化过的任务元数据,计算它们的 z -分值,然后在分别计算各自的 z -分值到初始聚类中心的 z -分值之间的距离,再根据最近原则,将剩余任务序列赋给与它们之间的 z -分值距离最小的聚类中心;

 再使用每个聚类中心的实例来计算每个类新的平均值;

 若新的平均值等于上次迭代的平均值,终止该过程。否则,用新平均值作为新的聚类中心,返回第 3 步继续执行;

 return $ClusterJobQueue (J_1, J_2, \dots, J_m)$;}

End

通过 K 均值聚类可以把少资源多任务型归并到资源与任务个数相同型中进行处理。

(3) 当 $n < m$ 时,也即当多资源少任务时

当网格中可用资源的数目大于向网格中提交的任务数时,算法根据时间-费用矩阵,从 m 个可用资源中,按照资源执行任务时间最少的策略选择出 n 个资源来

处理这 n 个任务,然后构造成资源任务相同型。为此文中提出了基于时间-费用最优化的网格资源选择算法 (Grid Resource Selection based on Time-Cost Optimization, GRS-TCO)。整个算法的形式化描述如算法 2 所示:

算法 2 基于时间-费用最优化的网格资源选择算法 (Grid Resource Selection based on Time-Cost Optimization, GRS-TCO)

Input: 时间-费用矩阵 $C_m \times n = \begin{pmatrix} c_{11} & \dots & c_{1n} \\ \dots & \ddots & \dots \\ c_{m1} & \dots & c_{mn} \end{pmatrix}$;

Output: 最优化的资源列表 $RList$;

Algorithm GRS-TCO (double[] $C_m \times n$)

Begin {

 1. Input ($C_m \times n$) ; // 输入时间-费用矩阵;

 2. $i = 1, List = RList$;

 3. while ($i \leq n$) {

 4. for (int $j = 1; j \leq m; j++$) {

 5. 求解 $\min (C_{ij})$ 中对应的资源标号 j ;

 6. $RList.add(j)$; // 把资源标号 j 加入到最优化的资源列表 $RList$ 中;

 7. $C_m \times n -> C_m \times n$; // 去掉第 i 个任务所在的行和资源 j 所在的列;

 8. $C_m \times n = C_m \times n$; // 使用处理后的矩阵 $C_m \times n$ 代替 $C_m \times n$;

 9. $i = i + 1$;

 10. return $RList$; // 返回最优化的资源列表;}

由于整个算法最后求解出来的最优化的资源列表取决于相应的时间-费用矩阵 $C_m \times n$ 的值,若该值在算法开始时已经确定了,那么任务扫描的先后顺序都不影响最后得出的最优资源列表。

整个算法的时间复杂度约为 $O(n * m)$ 。通过对 m 个资源按照时间-费用最优化的原则进行选择,使得多资源少任务型也转化为资源与任务个数相同型进行处理,从而可以简化整个算法的复杂度。

2.3 GEP 编码

编码^[2]方法在很大程度上决定了如何进行群体的遗传进化等运算以及这些运算的效率,因此是挖掘过程中一个很关键的步骤。在基于并行 GEP 的网格资源分配算法中,采用了资源和任务相结合的非线性编码方式,整个 GEP 的染色体既能反应提交给网格的任务,又能体现处理这些任务的网格资源。

(1) 对于 $n = m$ 和 $n < m$ 两种情况

当 $n < m$ 时,通过算法 2 从 m 个异构资源中选择 n 个资源以更好地处理提交给网格的 n 个任务。首先把它转化为资源任务个数相同型,这样方便 GEP 编码。下面以资源任务个数相同为例说明 GEP 是如何编码的。

例 2 假设网格中有 5 个任务,5 个资源,利用 GEP 对该 5×5 组合优化问题进行编码,随机产生的编码形式如下所示:

0 1 2 3 4 0 1 2 3 4
 3 5 2 4 1 E C B A D

在该 GEP 染色体中有两个基因组成,第一个基因是由 Head = {1,2,3,4,5} 组成,代表 5 个任务,第二个基因是由 Tail = {A,B,C,D,E} 组成,代表 5 个资源.两者之间如何联系成了 GEP 解决这种问题的重点.图 1 就很好地说明了两个基因之间的联系.

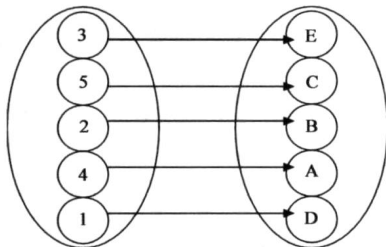


图1 GEP染色体中两个基因间的联系

由图 1 可以很清楚地看出,任务 3 由网格资源 E 完成,任务 5 由网格资源 C 完成,依此类推.

(2) 对于 $n > m$ 这种情况

当 $n > m$ 时,通过算法 1 使得 n 个任务聚成 m 类,假设原来有任务集合 $T = \{1,2,3,4,5,6\}$,资源集合 $R = \{A,B,C,D,E\}$,通过 JC-Kmeans 算法把 6 个任务聚成 5 类,形成 $T = \{1,2,3,4,5\}$,此时少资源多任务的编码也就转化为资源任务相同型的编码形式.

先假设有 GEP 编码为:

0 1 2 3 4 0 1 2 3 4
 4 1 3 2 5 D B C A E

那么原来任务集合 T 和资源集合 R 两基因之间对应的关系如图 2 所示.

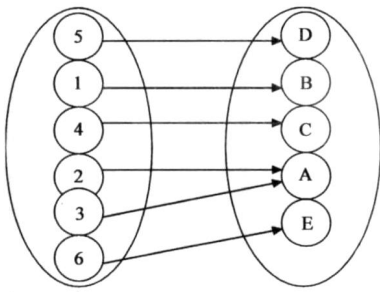


图2 少资源多任务的GEP编码中基因之间的对应关系

很明显,对于资源和任务个数不同的网格资源分配问题需要不同的染色体结构以及多基因之间的关系,而这种染色体结构是网格资源分配问题的基本编码方式.

2.4 基于并行 GEP 的网格资源分配算法

并行 GEP 可以克服传统的 GEP 算法的不足,从而提高了 GEP 的求解速度和质量.目前并行算法的设计主要有主从式、粗粒度以及细粒度三种模型^[12].本文采用了粗粒度模型.首先将 GEP 核心操作算法分别部署到相应的各个处理器上,客户端首先负责初始化种群,然后按照处理器个数进行 GEP 种群的分割并传输到各个处理器上,同时根据已有的网格资源的时间-费用矩阵来并行执行网格资源任务的分配,然后把最优的网格资源分配组合序列返回到客户端.

利用并行 GEP 对 $n \times m$ 的网格资源分配问题进行优化,首先根据资源与任务的特点进行非线性编码,最后运用文献[2]中提出的反转遗传操作,一代代进化直到求出最优解.基于并行 GEP 的网格资源分配算法的形式化描述如算法 3 所示.

算法 3 基于并行 GEP 的网格资源分配算法 (Grid Resource Allocation Algorithm based on Parallel GEP, GRA-PGEP)

Input :任务个数和资源的个数以及时间-费用矩阵 $C_m \times n$;
 Output :最优的网格资源分配组合 Best-ResTaskString;
 Begin {
 1. 输入待求解的资源个数 m 和任务个数 n 以及时间-费用矩阵 $C_m \times n$;
 2. if ($n < m$) {
 3. GRS-TCO(double[] $C_m \times n$); //按照时间-费用最优化原则从 m 个资源中选择 n 个资源;}
 4. else ($n > m$) {
 5. JC-Kmeans(int m ,String[] J_1, J_2, \dots, J_n); //把 n 个任务按照 K 均值聚类成 m 类;}
 6. Initial Population S ; //由客户端初始化种群;
 7. MappingSubPopulation(S, n); //把初始种群 S 分割成 n 个子种群,映射到相应的 n 个处理器上;
 8. While ($fitness < MaxFitness$) or ($gen < MaxGen$) { //服务端 n 个子种群同时执行开始;
 9. 解码并计算种群中每一个个体的适应度函数值 $fitness$;
 10. 向主线程返回最优个体;
 11. if ($epoch \bmod t = 0$) { //是否执行各子种群间交换;
 12. select emigrants; //选择子种群中最佳个体进行交换;}
 13. Inversion($MGF, point1, point2, rate$); //在随机选择 MGF 中,按照一定的概率对 $point1$ 到 $point2$ 之间的基因串进行反转;
 14. $gen = gen + 1$;}
 15. return Best-ResTaskString; //返回最优的网格资源分配组合序列;}

整个 GRA-PGEP 算法当到达预定的代数或种群已经收敛或达到最大适应度值时,停止进化,并输出最优的网格资源分配组合序列.

3 仿真实验

为了证明 GRA-PGEP 算法的有效性,在实验室 4 台机器所构成的环境中做了仿真实验.整个实验平台为 Windows2000 操作系统 + 512M + P41.86,所有算法程序由 Java 语言实现.

实验 1 用产生随机数的方法产生时间-费用矩阵 $C_m \times n$,对于资源与任务相同型,在实验中取 $n = m = 10$,分别使用 GRA-PGEP、GA、GEP 三种算法来求得最优的资源任务组合 Best-ResTaskString,三种算法分别运行 100 次,3 种算法的平均收敛代数和成功率如表 1 所示.

从表 1 中可以看出,当资源与任务个数相等时,GRA-PGEP 算法的平均收敛代数比 GEP 提高了 1 倍多,

比 GA 要提高了 3 倍多,而且该算法的优化成功率也比 GEP 约高 10%,比 GA 约高 21%。同时在实验 1 中,对 n 或者 m 的个数从 12 递增到 30,步长为 1,比较了三者的成功率,结果如图 3 所示。

表 1 三种算法的平均收敛代数数和成功率比较

算法名称	平均收敛代数	成功率
GA	230	76.45 %
GEP	120	87.43 %
GRA-PGEP	70	97.23 %

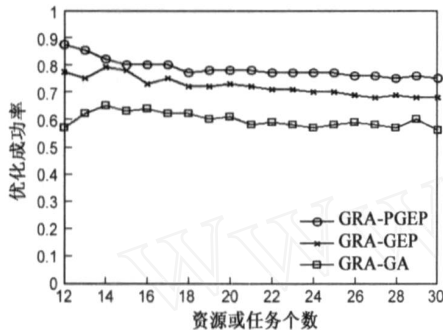


图 3 随着资源或任务个数的递增三种算法的优化成功率比较

从图 3 中可以看出,随着 n (或者 m)的增大,GRA-PGEP 的优化成功率最大比 GEP 提高了约 9%,比 GA 提高了约 19%。

对于当 $n < m$ 时,也即当多资源少任务型时,首先使用 GRS-TCO 算法进行资源的选择,然后再根据 $n = m$ 这种情况进行优化处理。实验中取 $n = 4, m = 5$,随机生

成时间-费用矩阵 $C_{5 \times 4} = \begin{pmatrix} 1 & 11 & 2 & 9 \\ 2 & 3 & 1 & 1 \\ 5 & 2 & 4 & 2 \\ 7 & 9 & 5 & 1 \\ 9 & 1 & 7 & 3 \end{pmatrix}$,图 4 显示了

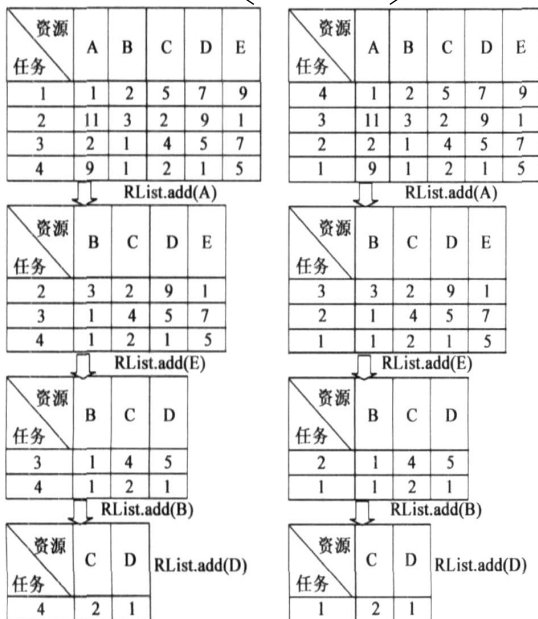


图 4 GRS-TCO算法中时间-费用矩阵的变化过程

在基于时间-费用优化的资源选择算法中的时间-费用矩阵的变化过程。

从图 4 中可以看出,通过时间-费用最优原则进行资源的选择,形成新的处理任务的资源集合,从而保证任务完成时间的最优。并且从图 4 的 a 和 b 中可以看出,对于确定的时间-费用矩阵,人为改变任务扫描的先后顺序并不能改变最优资源列表的内容。

实验 2 对于 $n > m$,首先按照算法 1 对所给任务数进行聚类,在实验中首先对所有任务之间按照 z 分值距离进行聚类判别,然后对判别后的结果使用 GRA-PGEP 算法进行优化处理。表 2 显示在这种少资源多任务型的资源分配中,随着 n 和 m 规模的增大,GRA-PGEP、GEP 以及 GA 算法的收敛代数数和耗时情况。

表 2 三种算法的实验结果对比

问题规模		各种分配算法	实验结果				
资源个数	任务个数		1	2	3	4	
5	12	GRA-PGEP	收敛代数	7	9	5	7
			耗时(分)	0.23	0.23	0.24	0.22
		GEP	收敛代数	13	14	11	12
			耗时(分)	1.3	1.2	1.5	1.2
		GA	收敛代数	20	19	22	20
			耗时(分)	2.3	2.1	2.2	2.1
40	95	GRA-PGEP	收敛代数	37	44	43	51
			耗时(分)	5.2	5.4	5.6	5.5
		GEP	收敛代数	88	92	99	102
			耗时(分)	78.1	77.3	79.2	77.1
		GA	收敛代数	113	116	115	123
			耗时(分)	111.3	114.2	113.1	114.4
100	150	GRA-PGEP	收敛代数	71	73	78	74
			耗时(分)	12.2	12.4	12.2	12.9
		GEP	收敛代数	243	249	245	244
			耗时(分)	192.1	194.4	193.2	193.6
		GA	收敛代数	421	429	434	443
			耗时(分)	255.2	255.3	255.4	255.3

从表 2 中可以看出,随着问题规模的增大,GRA-PGEP 算法比 GEP 算法的平均收敛速度最大提高了约 2 倍,比 GA 算法的平均收敛速度最大提高了约 6 倍;GRA-PGEP 算法比 GEP 的平均耗时最大降低了约 16 倍,比 GA 算法的平均耗时最大降低了约 21 倍。由此可见,随着任务 n 和资源 m 个数的增大,GRA-PGEP 算法对于资源优化分配有着良好的性能。

4 总结

网格资源优化分配是网格中间件中需要解决的关键问题之一,随着资源和任务个数的增加,网格资源优化问题将以指数级进行增长,因此研究高效健壮的网格资源分配算法是解决该关键问题的重要前提。本文对资源个数与任务个数的三种不同情况进行了详细的

分析,并给出了相对应的策略来解决其中的关键技术问题,在此基础上结合基因表达式编程(GEP)的特点,提出了基于并行 GEP 的网格资源分配算法(Grid Resource Allocation Algorithm based on Parallel GEP, GRA-PGEP),通过仿真实验证明了该算法大大地提高了求解最优资源组合优化序列的质量和速度,并比较了在相同的条件下,GRA-PGEP 算法比传统的 GEP 和遗传算法 GA 的收敛速度要快,耗时要少,求解的质量要高。

参考文献:

- [1] Foster I, Kesselman C. The Grid:Blueprint for a New Computing Infrastructure [M]. San Francisco: Morgan Kaufmann, 1999.
- [2] Candida Ferreira. Gene Expression Programming: A new adaptive algorithm for solving problems [J]. Complex Systems, 2001, 13(2): 87 - 129.
- [3] Ferreira C. Gene expression programming in problem solving [A]. Invited Tutorial of the 6th Online World Conference on Soft Computing in Industrial Applications [C]. Berlin: Springer, 2001, 9: 10 - 24.
- [4] Candida Ferreira. Automatically defined functions in Gene expression programming [J]. Genetic Systems Programming: Theory and Experiences, Studies in Computational Intelligence, 2006, 13: 21 - 56.
- [5] Candida Ferreira. Function finding and the creation of numerical constants in Gene expression programming [A]. The 7th Online World Conference on Soft Computing in Industrial Applications [C]. England: Springer, 2002.
- [6] Fernandez-Baca D. Allocating Modules to Processors in a Distributed System [J]. IEEE Transactions on Software Engineering, 1989, 15A(11): 1427 - 1436.
- [7] 蒋思伟,蔡之华,曾丹等. 基于模拟退火的并行基因表达式编程算法研究 [J]. 电子学报, 2005, 33(11): 2017 - 2021. Jiang Si-wei, Cai Zhi-hua, Zeng Dan, et al. Parallel Gene expression programming algorithm based on simulated annealing method [J]. Acta Electronica Sinica, 2005, 33(11): 2017 - 2021. (in Chinese)
- [8] Chang-an Yuan, Chang-jie Tang, et al. Intelligent function model discovery system based upon Gene expression programming [J]. Journal of Computational Information Systems, 2006, 2(4): 1299 - 130.

- [9] Zuo Jie, Tang Changjie, Zhang Tianqing. Mining predicate association rule by gene expression programming [A]. Proc of the 3rd Int'l Conf for Web Information Age 2002 (WAIM02). LNCS 2419 [C]. Berlin: Springer-Verlag, 2002. 92 - 103.
- [10] Zuo Jie, Tang Changjie, Li Chuan, et al. Time Series Prediction based on Gene Expression Programming [A]. Proc of the 5th Int'l Conf for Web Information Age 2004 (WAIM04). LNCS 3129 [C]. Berlin: Springer-Verlag, 2004. 55 - 64.
- [11] 姚恩瑜,何勇,陈仕平,等. 数学规划与组合优化 [M]. 浙江杭州: 浙江大学出版社, 2001.
- [12] 陈国良,王照法,庄镇泉等. 遗传算法及其应用 [M]. 北京: 人民邮电出版社, 1996.

作者简介:



邓松 男, 1980 年生于安徽合肥, 南京邮电大学博士生, 主要研究方向为网格计算、数据挖掘、计算机软件等。

E-mail: dsylc2006@yahoo.com.cn



王汝传 男, 1943 年生于安徽合肥, 南京邮电大学教授、博士生导师, 主要研究方向为计算机软件、计算机网络、信息安全、移动代理技术等

E-mail: wangrc@njupt.edu.cn

张羽 男, 1978 年生于陕西铜川, 南京邮电大学硕士生, 主要研究方向为网格计算、计算机软件、计算机网络等。

张建凤 女, 1983 年生于江苏连云港, 南京邮电大学硕士生, 主要研究方向为网格计算、计算机软件、计算机网络等。