一种面向服务的网格工作流调度算法*)

郭文彩　杨　扬

(北京科技大学信息学院　北京 100083)

摘　要　面向服务的网格工作流的研究已成为网格领域的研究热点。由网格服务组成的工作流(GSF)的调度问题是一个典型的NP问题,由于遗传算法具有并行性和全局解空间搜索的特点,非常适合解决这个问题。因此,本文首先给出GSF的GA定义,然后提出基于遗传算法的网格服务工作流调度算法GSFGA,并通过应用实例验证了该算法优于传统的调度算法,作为结论本文指出了下一步的研究工作。

关键词　网格服务,工作流,调度,遗传算法

Genetic Scheduling Algorithm for Service Oriented Grid Workflow

GUO Wen-Cai　YANG Yang

(School of Information Engineering, Beijing University of Science and Technology, Beijing 100083)

Abstract　The service oriented grid workflow, GSF, has been a research focus in grid technology. As an NP problem, grid service scheduling is difficult to be solved by means of classic algorithms. Featured in searching concurrently and globally, genetic algorithm can be a better option for solving GSF scheduling problem, therefore, a GA based grid service scheduling algorithm, GSFGA, is provided for obtaining the best GSF instance with highest fitness in this paper. The experiment results prove it available and better than some traditional algorithms. As a conclusion, the further work is also pointed out.

Keywords　Grid service, Workflow, Scheduling, Genetic algorithm

1　引言

现在网格技术已成为计算机领域的研究热点。近年来,GGF(Global Grid Forum)组织借鉴Web服务的应用成果,提出开放的网格服务架构OGSA[1],并逐渐实现其中的重要服务。OGSA架构定义了面向服务的网格体系,使得所有的物理资源均可表示为服务,其优点是已有的网格服务或网络服务易于组合成新的服务实现复杂的应用,而这种组合可以通过工作流[2]技术实现。在一个业务过程中工作流的参与者通常是一个网络服务或网格服务,而该过程也可表示成一个组合服务,如此循环嵌套可以满足非常复杂的应用需求。

在网格服务组成的工作流中,服务与资源间的映射关系组成网格服务的调度方案,而这种调度往往是NP完全问题。在已有的网格资源调度算法中,主要侧重解决网格作业与网格资源间的调度问题,其中Buyya[3]提出基于经济学模型的优化调度模型,其目的是在资源提供者和使用者间建立一种"交易",以尽可能低的费用满足资源使用者进行计算任务的最低要求;Condor[4]采用一种基于ClassAd匹配的集中式方案,以形成资源调度器;Legion[5]提供了一个高度结构化的可扩展的调度方案。但是这些调度方案未能从网格服务的角度考虑网格服务工作流的调度。因此,本文介绍了一种基于遗传算法的网格服务工作流调度问题的求解方法。

2　网格服务工作流(GSF)及其调度问题

定义1(网格服务工作流,GSF)　网格工作流W由过程单元P和过程单元间变迁关系$Trans\subseteq P\times P$组成,其中,过程单元包括or-汇聚、and-汇聚、or-分支、and-分支节点以及由网格服务集S提供的活动集A组成,若存在双射$f: A\leftrightarrow S$,则称W为网格服务工作流,简称为GSF。

上述定义表明,网格服务工作流是网格工作流的一个特例,即组成网格工作流的任一活动是由网格服务集合中的唯一服务提供,而任意服务仅能提供一种活动,从而简化网格服务间的合作关系,以解决网格工作流应用。

2.1　GSF应用

一个网格服务工作流应用可以表示为一个有向无回路图(DAG),$I=(Nodes, Edges)$,其中,图的节点($Nodes$)代表组成工作流的服务集,边($Edges$)代表了有向的服务间的依赖关系,工作流服务分为两类:$Nodes=Nodes^{JS}\cup Nodes^{FT}$;

作业提交服务表示为$JS(z)\in Nodes^{JS}$,其中z是执行JS服务的网格节点;数据传输服务表示为$FT(z_1, z_2)\in Nodes^{FT}$,其中$z_1$、$z_2$是数据传输的源、目的节点;

令$succ(N)$为服务$N\in Nodes$的后继服务,即:$N_s\in succ(N)\Leftrightarrow\exists(N, N_s)\in Edges$,同样令$pred(N)$为服务$N\in Nodes$的前趋服务,即$N_p\in pred(N)\Leftrightarrow\exists(N_p, N)\in Edges$,若$pred(N)=\phi$,$\phi$为空集,则$N$为起始服务;若$succ(N)=\phi$,则$N$为终止服务。此外,服务$N$的$p$级前趋和后继可以表示为$pred^p(N)=pred(\cdots pred(N))$和$succ^p(N)=succ(\cdots succ(N))$。$\forall N_1, N_2\in Nodes$,若$N_1$与$N_2$相关,有且仅有$\exists p: N_1\in pred^p(N_2)\cup N_1\in succ^p(N_2)$。

2.2　GSF调度问题

*)基金项目:国家自然科学基金重大研究项目(No. 90412012)。郭文彩　博士生,主要研究方向:网络服务与工作流。杨　杨　教授,博士生导师,主要研究方向:计算机网络、计算机图像处理。

设 A 是一个 GSF 工作流应用，S 是组成 A 的网格服务集合，V 是完成网格服务的网格节点集合，对于 $\forall z_i \in S, i \in [1..n]$，$\exists V^{z_i} \in V$，一个 GSF 调度是一种映射：$S_A = S(A(z_1, \cdots, z_n)) = AI(e_1, \cdots, e_n)$，$\forall e_i \in V^{z_i}$，$\forall i \in [1..n]$

在 AI 内，一个作业提交调度表示为一种映射：$S_{JS(z_j)} = e_j$；一个文件传输调度的映射表示为：$S_{FT(z_k, z_l)} = (e_k, e_l)$，GSF 调度的目的是找到调度实例 AI 满足目标函数 $f = \min(\sum_{i=1}^{n} Exec(S_i))$，其中 $Exec(S_i)$ 是服务 S_i 的执行时间。

因此，求解 GSF 调度问题是一个 NP 完全问题，本文将借鉴启发式算法予以解决。

3 基于遗传算法的 GSF 调度问题求解

遗传算法[6]简称 GA，是由 John H. Holland 及其合作者在 20 世纪 70 年代提出的，模仿了自然界的自适应过程，提供一种搜索型算法。现在，随着机理上的不断完善和应用上的日趋广泛，遗传算法已成为求解调度问题在内的各类优化问题的有效优化算法之一。

GA 操作的对象是“染色体”，它代表了所有优化问题解的一个二进制代码串，“染色体群”即为优化问题解的集合，一般，GA 搜索最优解的过程就归结为寻找适应度高的个体染色体。为此，将染色体群置于问题的环境中，基于适者生存的原则，通过选择、复制、交叉、变异实现染色体的进化，按此方式，通过一代代不断地进化，搜索过程最后会收敛到最适应环境即适应度最高的一个染色体，即求得问题的一个最优解。

3.1 GSF 应用的 GA 定义

定义 2(GSF 染色体) 设 $A(z_1, \cdots, z_n)$ 表示一个 GSF 应用，其中，z_i 为服务变量，$\forall i \in [1..n]$，令 V^{z_i} 为 z_i 的数值集，z_i 即被定义为基因，等位基因是基因的实例化，即元素 $e_i \in V^{z_i}$；应用 A 的所有服务变量的完全排序组成 GSF 染色体，基因 z_i 在染色体中的排列序号 i 称作基因座。GSF 应用的实例 $AI(e_1, \cdots, e_n)$ 称作个体，其中，$e_i \in V^{z_i}$，$\forall i \in [1..n]$，GSF 应用的目标函数的倒数（即 f^{-1}）称为适应度函数，即 GSF 染色体的适应值越大，GSF 应用实例执行时间越短。

GSF 染色体的操作如下所述：

(1)种群初始化。具有固定种群数的 GSF 种群初始化是通过随机产生一组应用实例实现的，即把随机数值分配给服务变量：

$P = \{AI_i(e_1, \cdots, e_n) \mid e_j \in V^{z_j}, \forall j \in [1..n], \forall i \in [1..p]\}$。

对于各种应用问题，应根据试验确定种群数，此外，通过人工干预也可插入应用实例，以改善计算效率。

(2)GSF 染色体的选择。选择最佳的应用实例通过选择算子（再生算子）可以生成新的种群，设 P 为一个种群，它的种群数为 p，平均适应度为 $\overline{F}$，使用重置余数随机采样法[7]选择模型产生新的种群：$P' = P_1 \cup P_2$

选择步骤如下：

1)期望值模型：与适应度成比例的选择应用实例，并且消除随机采样误差，得：

$P_1 = \bigcup_{i=1}^{p} \bigcup_{j=1}^{\left\lfloor \frac{F(AI_i)}{\overline{F}} \right\rfloor} clone_j(AI_i)$

2)$P_2 = \bigcup_{i=1}^{s} clone_j(AI_j)$，其中 $s = |P| - |P_1|$，$r_i \in [0,1]$ 是一个随机数使得：

$$\frac{\sum_{k=1}^{i-1}\left\{\frac{F(AI_k)}{\overline{F}}\right\}}{\sum_{k=1}^{p}\left\{\frac{F(AI_k)}{\overline{F}}\right\}} < r_i \leqslant \frac{\sum_{k=1}^{i}\left\{\frac{F(AI_k)}{\overline{F}}\right\}}{\sum_{k=1}^{p}\left\{\frac{F(AI_k)}{\overline{F}}\right\}},$$

$|P|$ 为集合 P 的势，利用每个个体适应度与平均适应度间的比例，模仿赌轮方法填充第 1)步遗留的空闲位置。

(3)GSF 染色体的交叉。在遗传算法中使用交叉算子实现局部最大值的搜索，算法采用单点交叉算子，使用随机函数定义该算子：

$\otimes_r : V^A \times V^A \to V^A \times V^A$，

$AI_1(e_1, \cdots, e_n) \otimes AI_2(e'_1, \cdots, e'_n) = (AI'_1, AI'_2)$

其中：

$AI'_1 = AI'_1(e_1, \cdots, e_r, e'_{r+1}, \cdots, e'_n)$，

$AI'_2 = AI'_2(e'_1, \cdots, e'_r, e_{r+1}, \cdots, e_n)$，

且 $r \in [1, n-1]$ 是一个随机数，交叉操作如图 1(a)所示。

设 $P = \{AI_1, \cdots, AI_n\}$ 表示一个 GSF 应用实例，p_c 为交叉概率，p_c 的值根据不同的问题域通过实验确定，交叉后的应用实例子集表示为：$P_c = \bigcup_{i=1}^{n} AI'_i$

(4)GSF 染色体的变异。变异操作使得算法能够跳过其它的搜索空间，避免种群的局部停滞，应用于网格应用的实例的变异算子表示为：

$\otimes : V^A \to V^A$，$\oplus(AI(e_1, \cdots, e_n)) = AI'(e'_1, \cdots, e'_n)$

其中：

$$e'_i = \begin{cases} e''_i, r_i < p_m; \\ e_i, r_i \geqslant p_m; \end{cases}$$

p_m 是基因的变异概率，取值由试验确定，$r_i \in [0,1]$ 是一个随机数，$e''_i \in V^{z_i}$ 是一个随机选择的等位基因，$\forall i \in [1,n]$。单基因变异的染色体实例如图 1(b)所示。

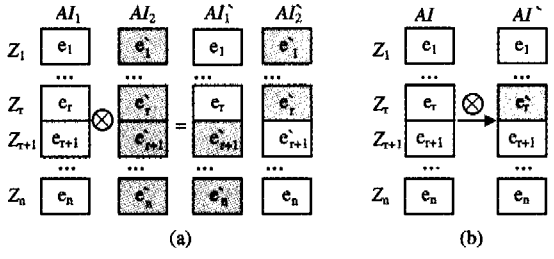

图 1 GSF 染色体的交叉与变异

3.2 GSFGA 算法描述

结合上述 GSF 应用的 GA 定义，针对 GSF 调度问题，我们提出 GSFGA 算法，算法描述如下：

输入：1)GSF 应用 A；2)适应度函数 F；3)种群数目 p；4)交叉概率 p_c；5)变异概率 P_m；6)种群代数最大值 max_gen；7)稳态比例；8)适应度定标系数 C_{mult}；9)最优保存策略进化模型

输出：GSF 应用 A 的最佳实例 AI；

1. 初始化种群内的所有个体；
2. $generation = 1$；
3. repeat
   选择并复制 A 的实例；
   以概率 p_c 实现应用实例的交叉；
   以概率 p_m 实现任务变量的变异；
   $generation = generation + 1$；
   until 满足收敛判据 or 稳态 or generation $\geqslant$ max_gen
4. return 适应度最大的应用 A 的实例.

针对传统遗传算法的收敛性能差和早熟收敛等缺点，本算法做了如下改进：

(1)为增强灵活性，GSFGA 算法定义了三种收敛判据，这三种判据可以组合使用：

a)满足目标函数定义的收敛判据，例如，适应值超过上限；

b)超过预定义的最大种群代数；

c)达到稳态停滞，不能继续提高适应值，这种状态可以在预定义数量的种群代中使用滑动窗口检测最优个体的适应度

来确定。

(2)采用最优保存策略[8]实现跨代保留

反复地执行交叉与变异操作,有可能导致删除最佳应用实例,影响应用的优化求解。设 P_G 是 GSF 应用的第 G 代种群,$AI_B^G \in P_G$ 是当前的最优应用实例(即 $F(AI_B^G) \geqslant F(AI)$, $\forall AI \in P_G$),P_{G+1}是下一代种群,最优保存策略的目的是强制跨代保留最优应用实例:

$$P'_{G+1}=\begin{cases} P_{G+1}, & AI_B^{G+1} \geqslant AI_B^G; \\ P_{G+1}-AI \cup AI_B^G, & AI_B^{G+1} < AI_B^G, \end{cases}$$

其中,$AI \in P_{G+1}$是随机删除的个体。最优保存策略有可能导致算法的未成熟收敛。

(3)通过适应度定标[9]防止早熟收敛

传统的染色体选择方法存在下列问题:

1)在算法执行的初期,往往有可能出现几个超级个体主宰了后代的产生,并且导致算法的未成熟收敛;

2)在算法执行的后期,种群的平均适应度经常接近最佳适应度,在这种情况下,个体间竞争力减弱,最佳个体和其它大多数个体在选择过程中有几乎相等的选择机会,从而使有目标的优化过程趋于无目标的随机漫游过程。

设 $\bar{F}$ 为种群的平均适应度,线性定标定义了 GSF 应用实例的新的定标函数:

$$F'=a \cdot F+b$$

其中,通过求解下列方程可以得出 a 和 b:

$$\begin{cases} a \cdot \bar{F}+b=\bar{F}' \\ a \cdot F_{\max}+b=C_{mult} \cdot \bar{F} \end{cases}$$

这 2 个方程保证了实现遗传算法正确收敛的两个关键特点:

1)定标后的平均适应值 $\bar{F}'$等于原来的平均适应值,这是因为每个平均应用实例都要为下一代产生子孙;

2)适应值为 $F_{\max}$ 的最佳应用实例将为下一代产生 C_{mult} 个子孙,这样可以减少先辈中超级个体和平均个体间的差距,从而避免未成熟收敛;增加后代中这种差距以确保必要的竞争。

4 实验结果

为验证 GSFGA 算法,我们模拟了由 5～30 个网格服务组成的工作流,其中每个服务都有 2～5 个网格节点可以执行该服务。初始种群数为 100,变异概率是 0.04,交叉概率是 0.8,本文算法与 SJLF[10]和 LJLF[10]两个常用算法比较,实验结果如图 2 所示,实验表明,采用 GSFGA 算法所产生的所有网格服务的全部执行时间均小于 SJLF 和 LJLF 算法产生的执行时间。

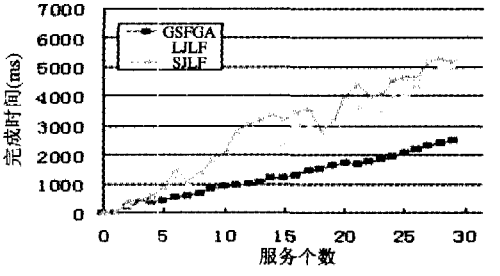

图 2 GSFGA 与 LJLF、SJLF 算法比较

结语 本文针对网格服务工作流调度问题,提出基于遗传算法的网格服务工作流调度算法 GSFGA,并采用灵活的收敛判据、最优保存策略和适应度定标来克服传统遗传算法收敛性差和早熟收敛的缺陷,并通过 GSF 应用实例验证了该算法的有效性。为简化研究,本文定义的网格服务工作流假设每个服务仅提供一种活动,而实际的网格服务应用却复杂得多;此外,网格环境的动态性将很大程度地影响 GSF 调度算法地实施,因此,在下一步工作中,我们将针对复杂的 GSF 应用,研究 GSF 的动态调度问题。

参 考 文 献

1 Kesselman F I, Nick C J, Tuecke S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Globus Project, 2002. http:// www. globus. org/research/papers/ogsa. pdf.

2 van der Aalst W, van Hee K. Workflow Management Models, Methods, and Systems. The MIT Press. Mar. 2004

3 Buyya R, Abramson D, Giddy J. An economy driven resource management architecture for global computational power grids1 Int'l Conf on Parallel and Distributed Processing Techniques and Applications, Las Vegas, 2000

4 Frey J, Tannenbaum T, Foster I, Livny M, Tuecke S. Condor-G: A computation management agent for multi institutional grids. Cluster Computing, 2002, 5: 237～246

5 Chapin S, Karpovich J, Grimshaw A. The Legion resource management system. In: 5th Workshop on Job Scheduling Strategies for Parallel Processing, Apr. 1999

6 Goldberg D E. Genetic Algorithms in Search, Optimization 6 Machine Learning. Reading. Addison-Wesley, Massachusetts, 1989.

7 Geist G A, Heath M T, Peyton B W, Worley P H. A user's guide to PICL: a portable instrumented communications library. Technical Report ORNL/TM-11616, Oak Ridge National Laboratory, Oak Ridge, Tennessee, Jan. 1992

8 Bhandari D, Murthy C A, Pal S K. Genetic Algorithm with elitist model and its convergence. Int. J. Pattern Recognition Artif. Intell. 10, 1996

9 Kreinovich V, Quintana C, Fuentes O. Genetic algorithms: What finess scaling is optimal?. Cybernetics and Systems, 1993, 24(1): 9～26

10 Di Martino V, Mililotti M. Sub-optimal scheduling in a grid using genetic algorithms. Parallel Computing, 2004, 30(5-6): 553～565

(上接第 114 页)

17 Tripathy S, Panda B. Post-Intrusion Recovery Using Data Dependency Approach. In: Proc. of the 2001 IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, June, 2001

18 蔡亮,杨小虎,董金祥. 信息战下的数据库安全-我国的持殊需求分析和对策. 计算机研究与发展, 2002, 39(5): 568～573

19 Sandhu R, Samarati P. Access control: principles and practise [J]. IEEE Communications, 1994, 32(9): 40～48

20 Lebkicher M. Role Based Access Control [EB/OL]. http:// www. giac. org/practical/GSEC/Michael. Lebkicher. GSEC, pdf, 2000.

21 彭文灵,王丽娜,等. 基于角色访问控制的入侵容忍机制研究. 电子学报, 2005, 33(1): 91～95

22 Leonard J. Lapadula State of the Art in Anomaly Detection and Reaction: [Technical report]. MITRE, Bedford, Massachusetts, 1999

23 Javitz H S, Valdes A. The sri ides statistical anomaly detector. In Proceedings IEEE Computer Society Symposium on Security and Privacy, Oakland, CA, May 1991

24 Lee W, Xiang D. Information-theoretic measures for anomaly detection. In: Proc. 2001 IEEE Symposium on Security and Privacy, Oakland, CA, May 2001

25 Samfat D, Molva R. Idamn: An intrusion detection architecture for mobile networks. IEEE Journal of Selected Areas in Communications, 1997, 15(7): 1373～1380

26 Sekar S, Bendre M, Bollineni P. A fast automation-based method for detecting anomalous program behaviors. In: Proc. 2001 IEEE Symposium on Security and Privacy, Oakland, CA, May 2001