

云 workflow 任务调度的模拟退火遗传改进算法

黄婷婷, 梁意文

(武汉大学 计算机学院, 湖北 武汉 430072)

摘 要: 提出一种模拟退火(simulated annealing, SA)算法和遗传算法(genetic algorithm, GA)相结合的任务完成时间-可靠性的多目标优化算法. 该算法首先利用任务的影响程度生成合适的初始种群; 其次对交叉、变异等遗传操作产生出的个体分别进行模拟退火操作, 避免随机算法引起的早熟收敛问题; 最后, 在变异阶段引入失败率, 提高调度结果的可靠性. 实验分析表明, 该算法在解决早熟收敛和结果不可靠方面比 GA 性能更优.

关键词: 遗传算法; 云平台; 任务调度; 模拟退火算法; 多目标优化

中图分类号: TP338.8

文献标识码: A

文章编号: 1000-7180(2016)01-0042-05

An Improved Simulated Annealing Genetic Algorithm for Workflow Scheduling in Cloud Platform

HUANG Ting-ting, LIANG Yi-wen

(School of Computer Science, Wuhan University, Wuhan 430072, China)

Abstract: This paper proposed a multi-objective optimization algorithm combining simulated annealing algorithm with the genetic algorithm concerning makespan and reliability. Firstly, this algorithm makes tasks prioritization considering their influence degree tasks schedule policy to generates a suitable initial population. Secondly, for individuals which produced through crossover and mutation genetic operations to perform simulated annealing operation, in order to avoid the premature convergence problem caused by random algorithm. Then, it introduced failure rate in mutation phase to increase the reliability of the scheduling results. Experimental analysis indicates that the algorithm in solving the problem of premature convergence and result unreliable better than GA.

Key words: genetic algorithm; cloud platform; task scheduling; simulated annealing algorithm; multi-objective optimization

1 引言

云计算是一种新兴的商业分布式计算模型, 云平台下的 workflow 任务调度问题被认为是 NP 完全问题^[1]. 当前许多研究者将遗传算法和启发式算法应用到云平台的工作流任务调度中, 然而它们中的绝大多数只对任务完成时间^[1]或者结果可靠性^[2]中的一个目标进行优化. Wang X^[3]、Hakem M^[4]、Delavar A G^[5]等人提出了改进的遗传算法, 对任务完成时间和结果可靠性进行多目标的优化. 但是这些方法仍然存在早熟收敛和结果可靠性不高的问题.

模拟退火算法具有较强的局部搜索能力和跳出某局部最优解的优点, 易于向全局最优解快速收敛. 本文根据在遗传算法的搜索过程中融合退火算法的思想, 对任务完成时间-可靠性多目标优化的模拟退火遗传算法(Makespan and Reliability Simulated Annealing Genetic Algorithm, MRSAG)进行了改进, 主要在以下三个方面进行了改进: (1) 在获取初始种群阶段, 根据对其他任务的影响程度来确定任务的优先级, 影响程度越大的任务拥有越高的优先级并优先分配较可靠的资源, 生成一组合适的初始解, 加快种群的进化速度; (2) 对交叉、变异等遗传操

作所产生出的个体分别进行模拟退火操作,以其结果作为下一代群体中的个体,可以避免算法过早收敛于局部最优解;(3)在变异阶段选取失败率低的资源进行基因替换,提高调度结果的可靠性.通过与 GA 的对比实验,证明该算法性能更优.

2 问题建模和相关定义

2.1 工作流任务建模

工作流任务流图由存在相互依赖关系的子任务构成,这些子任务构成有向无环图(Directed Acyclic Graph, DAG),用二元组表示为 $G=(N, E)$,其中 N 代表节点集, E 代表连接节点的边,边上的数字代表边的权重.典型的 DAG 工作流任务模型如图 1 所示.

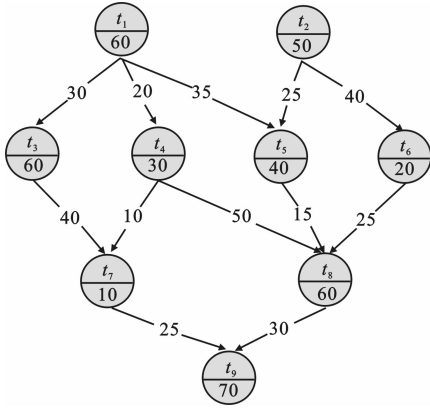


图 1 DAG 工作流任务模型图

在 DAG 图中除了开始节点和结束节点,每一个节点都有一个前驱节点和一个后继节点.没有前驱节点的称为开始节点,用 T_{entry} 表示;没有后继节点的称为结束节点,用 T_{exit} 表示.在图 1 中, $\{t_1, t_2\}$ 是开始节点, $\{t_9\}$ 是结束节点.

图 1 所示的 DAG 图中共有 9 个节点, $t_1, t_2, \dots, t_9$, 每个节点包括两部分,上半部分表示节点名称,下半部分表示任务计算量.连接任意两个节点的边上的数字表示需要传输的数据.

2.2 相关定义

用 T 表示 DAG 图中的节点集合,用来描述任务集 $T = \{t_1, t_2, \dots, t_n\}$, n 表示工作流中的任务总数.用 E 表示连接节点的边,用来描述任务之间的依赖关系, $E = \{e_{ij} \mid e_{ij} = (t_i, t_j), e_{ij} \in T \times T\}$,任务 t_j 必须在其所有前驱任务都执行完成后才能开始执行.任务完成时间、失败率的定义和计算方法如下,其中相关符号说明如表 1 所示.

假设任务 t_i 在资源 r_u 上执行,任务 t_j 在资源 r_v 上

表 1 相关符号说明

符号	定义
T	DAG 中节点的集合,表示任务集
E	DAG 中连接节点的边的集合,表示任务间的依赖关系
R	云平台中的虚拟资源集合
t_i	任务集中的第 i 个任务
$\text{load}(t_i)$	任务 t_i 的权值,表示任务的计算量
$T_{\text{pre}}(t_i)$	任务 t_i 的前驱任务集合
$T_{\text{sub}}(t_i)$	任务 t_i 的后继任务集合
$\text{data}(t_i, t_j)$	任务 t_i 和任务 t_j 间传输的数据量
$T_{\text{start}}(t_i, r_u)$	任务 t_i 的开始执行时间
$T_{\text{end}}(t_i, r_u)$	任务 t_i 执行结束时间
$T_{\text{avail}}(t_i, r_u)$	资源 r_u 可为任务 t_i 所用的准备时间
$T_{\text{prend}}(t_i, r_u)$	任务 t_i 的所有前驱任务执行完成的时间
$T_{\text{priority}}(t_i)$	任务 t_i 的优先级
$\text{comm}(t_i, t_j, r_u, r_v)$	任务 t_i (由资源 r_u 处理) 和任务 t_j (由资源 r_v 处理) 之间数据传输的时间
$\text{comp}(t_i, r_u)$	资源 r_u 处理任务 t_i 的时间
$\text{cp}(r_u)$	资源 r_u 的计算能力
$\text{tp}(r_u)$	资源 r_u 的通信能力
r_{uMIPS}	资源 r_u 每秒执行指令数
r_{ufail}	单位时间内,在资源 r_u 上执行的任务的失败率

执行.式(1)表示任务 t_i 到 t_j 数据传输的通信时间,式(2)表示任务 t_i 在资源 r_u 上的执行时间.

$$\text{comm}(t_i, t_j, r_u, r_v) = \text{data}(t_i, t_j) * \text{tp}(r_u) \quad (1)$$

$$\text{comp}(t_i, r_u) = \text{load}(t_i) * \text{cp}(r_u) \quad (2)$$

DAG 图中的每一个任务必须在其前驱任务都执行完后才能执行,式(3)表示任务 t_i 的开始执行时间,其中式(4)表示任务 t_i 的所有前驱任务执行完成的时间.

$$T_{\text{start}}(t_i, r_u) = \max\{T_{\text{avail}}(t_i, r_u), T_{\text{prend}}(t_i, r_u)\} \quad (3)$$

$$T_{\text{prend}}(t_i, r_u) = \max\{T_{\text{end}}(t_k) + \text{comm}(t_k, t_i)\} \quad (4)$$

式中, $t_k \in T_{\text{pre}}(t_i)$

任务 t_i 的完成时间由三部分组成,前驱任务的完成时间,任务 t_i 在资源 r_u 上的执行时间和任务 t_i 与 t_j 间数据传输的时间.式(5)表示任务完成时间.

$$T_{\text{end}}(t_i, r_u) = T_{\text{prend}}(t_i, r_u) + \text{comp}(t_i, r_u) + \text{comm}(t_i, t_j, r_u, r_v) \quad (5)$$

由此可推出任务完成时间与结果可靠性双目标优化问题可以由式(6)和式(7)表示,其中 m 表示可用资源的个数.

$$\text{MinimizeTime}(S) = \max(T_{\text{end}}(t_i, r_u)) \quad (6)$$

$$\text{MinimizeFail}(S) = \sum_{u=1}^m \text{comp}(t_i, r_u) * \text{fail}(r_u) \quad (7)$$

3 MRSAG 调度算法

模拟退火算法较强的局部搜索能力,可以避免遗传算法在搜索过程中陷入局部最优解. 基于 SA 思想,提出了一种 MRSAG 算法.

3.1 染色体编码

本文将每一个可能的解都编码为一个染色体,每一个染色体都有 N 个基因, N 表示任务的个数. 假定工作流包含 9 个任务,云平台提供 6 个可用资源,则染色体的长度为 9,每个基因取值为 1~6 之间的随机数.

首先,图中任务的优先级是基于其对其他任务的影响程度来计算;其次,从云平台提供的可用资源中选取合适的资源,优先分配给影响程度较高的任务. 染色体的编码如图 2 所示,任务 t_1 在资源 r_1 上执行,任务 t_2 在资源 r_4 上执行,以此类推.

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
r_1	r_4	r_3	r_6	r_1	r_2	r_3	r_4	r_5

图 2 任务-资源染色体

3.2 任务优先级

在获取初始种群前,任务是按照各自的优先级排队进入工作流进行调度的. 然而,在复杂的工作流图中,不能简单的将任务进行分层,即不能认为 DAG 中任务所处层级越高,则任务优先级越高. 式 (8) 表示任务优先级 $Tpriority(t_i)$.

$$Tpriority(t_i) = load(t_i) + \max\{d(t_j) * data(t_j, t_i)\} \quad (8)$$

式中, $t_j \in T_{pre}(t_i)$, $d(t_j)$ 是任务 t_j 在工作流图关键路径中的深度.

3.3 种群初始化

普通遗传算法的初始种群是随机获取的,随机种群可能导致早熟收敛,或者最终结果不够理想. 为此,本文采用任务影响程度级别快速生成可用初始种群,步骤如下所示.

Step 1: 选择 DAG 中所有开始节点(如图 1 中 t_1, t_2) 作为调度候选任务集;

Step 2: 随机选择候选任务集中还没有分配资源的任务 t_i ;

Step 3: 计算任务 t_i 的优先级 $Tpriority(t_i)$ 和资源 r_u 的失败率,计算方式分别如式 (8)、式 (9); $fail(r_u) = r_u^{fail} / r_u^{MIPS}$ (9)

Step 4: 根据式 (9) 的计算结果,将失败率低的

资源 r_u 添加到图 2 的染色体编码中;

Step 5: 如果候选任务集不为空,则返回 Step 2,若为空,则执行 Step 6;

Step 6: 如果当前任务不是结束节点(如图 1 中 t_9),则重复 Step 2 到 Step 5;

上述步骤重复执行,直到产生一组初始种群,该初始种群是一组可行的调度方案.

3.4 适应度函数

适应度函数的选取直接影响到最优解的获取,本文将任务完成时间和调度结果可靠性的加权值作为适应度函数. 对于每一批进化的种群,用 $minTime$ 表示该批种群中任务完成时间最小值, $maxTime$ 表示任务完成时间最大值,用 $minFail$ 表示该批种群中结果失败率最小值, $maxFail$ 表示结果失败率最大值. 则适应度函数可由式 (10) 表示,其中 α 和 β 为任务完成时间和结果失败率的加权因子,且 $\alpha + \beta = 1$.

$$F(S) = \alpha * \frac{Time(S) - minTime}{maxTime - minTime} + \beta * \frac{Fail(S) - minFail}{maxFail - minFail} \quad (10)$$

3.5 交叉操作

本文采取单点交叉变异进行免疫基因操作,以增加种群的多样性,避免种群过早收敛. 选择两个父母染色体,交换选中的基因片段,形成两个新的个体. 图 3 所示为交叉变异操作的前后过程.

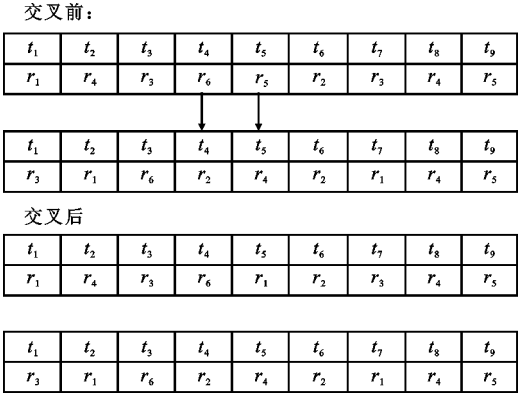


图 3 交叉操作

3.6 变异操作

基因变异操作可以促进基因交流,有利于丰富种群的多样性. 在该阶段引入失败率,可增加调度结果的可靠性. 步骤如下所示.

Step 1: 随机选择一个染色体;

Step 2: 随机选择染色体中的两个任务(t_i 和 t_j) 并交换它们;

Step 3:计算任务 t_i 和 t_j 的所有候选资源的失败率,若当前分配给任务的是失败率最低的资源则不做任何改变,否则,用候选资源中失败率最低的资源替换当前已分配的资源。

变异操作的示例如图 4 所示。

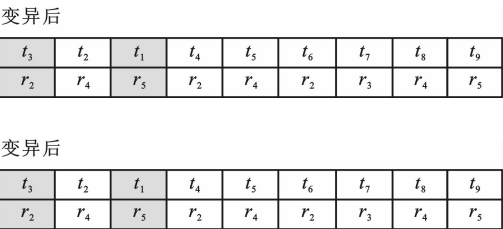


图 4 变异操作

假定任务 t_1 的候选资源为 $\{r_1, r_2, r_4, r_6\}$, 其中 r_2 的失败率最低;任务 t_3 的候选资源为 $\{r_3, r_4, r_5\}$, 其中 r_5 的失败率最低。

3.7 算法描述

模拟退火遗传算法结合 GA 算法和 SA 算法的优点,能够避免陷入局部最优解. 其中,SA 的初始解来自于 GA 的进化结果,被退火操作接受的种群成为 GA 进一步进化的初始种群. 步骤如下所示。

Step 1:初始化算法参数,设定初始温度 T_i , 进化代数 $i=0$;

Step 2:获取初始种群 G_i ,根据第 3.3 节介绍的初始种群方式生成一组可行的调度方案;

Step3:交叉和变异操作,利用遗传算法对种群 G_i 进行交叉和变异操作得到种群 G_i' ,其中,根据第 3.5 节介绍的单点交叉算子进行交叉运算,根据第 3.6 节介绍的步骤进行变异操作;

Step 4:适应度计算,按照式(10)分别计算种群 G_i 和种群 G_i' 的适应度函数值,分别为 $F(G_i)$ 和 $F(G_i')$;

Step 5:计算增量,公式为 $\Delta F' = F(G_i') - F(G_i)$,若 $\Delta F' < 0$,则接受 G_i' 为新的进化种群,否则以概率 $\exp\left(-\frac{\Delta F'}{T_i}\right)$ 接受 G_i' 为新的进化种群;

Step 6:退温,执行 $T_{i+1} = \lambda T_i, G_{i+1} = G_i', i = i + 1$,其中 $0 < \lambda < 1$;

Step 7:判断是否满足进化结束条件,如不满足,转 Step 3 以 G_{i+1} 为进一步进化的初始种群,否则转 Step 8;

Step 8:输出当前最优个体。

4 实验结果与分析

本文使用开源云仿真平台 CloudSim^[6] 来执行

workflow 任务调度 MRSAG 算法的仿真实验,并在相同实验条件下与 GA 算法进行比较,以证明该算法的正确性与有效性. 在该实验平台上分别测试了上述两种算法的适应度函数的值、任务完成时间和失败率,实验仿真参数如表 2 所示。

表 2 仿真参数

参数名	参数值
任务数	10~100
资源数	30
资源计算能力/MIPS	500~1 000
资源间带宽/(Mb/s)	10~100
资源失败率/h	$10^{-4} \sim 10^{-3}$
初始种群大小	10
迭代次数	10
交叉概率	0.5
变异概率	0.5
α, β	0.5

实验 1:适应度值对比测试。

从图 5 可以看出 GA 算法较早地出现了局部最优解,而 MRSAG 算法则有效利用了退火算法局部搜索的能力,使得它能概率性地跳出局部最优解并最终趋于全局最优。

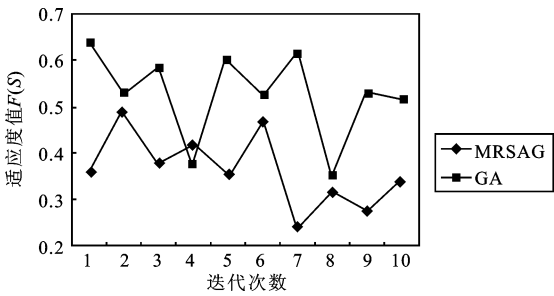


图 5 两种算法的适应度值对比

实验 2:任务完成时间对比测试。

任务完成时间是指调度开始到所有任务调度结束的时间,通过式(6)进行计算,值越小,表示任务的完成时间越短. 本文和 GA 算法进行比较,测试结果如图 6 所示. 从图 6 可以看出, MRSAG 算法 10 次实验的任务完成时间整体低于 GA 算法;可见 MRSAG 算法的任务完成时间更短,性能更优。

实验 3:调度结果失败率对比测试。

调度结果的可靠性是指用户提交到云平台的任务被成功完成的概率,失败率越低,调度结果越可

靠. 通过式(7)计算调度结果的失败率,该值越小,表明调度结果越可靠. 从图 7 中可以看出,MRSAG 算

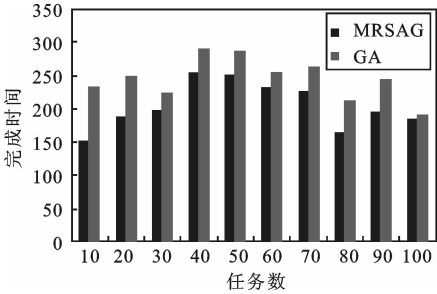


图 6 两种算法的任务完成时间对比

法 10 次实验的平均值为 0.37, GA 算法 10 次实验的平均值为 0.46;其中,MRSAG 算法的失败率在 0.4 以下的实验占 80%,而 GA 算法在 0.4 以下的仅占 50%;可见 MRSAG 算法能获得更多可靠的调度解.

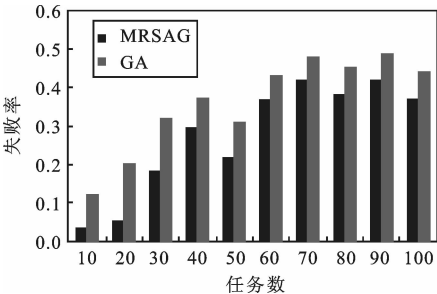


图 7 两种算法的失败率对比

5 结束语

本文在云环境下,提出了一种对任务完成时间和结果可靠性多目标优化的 MRSAG 算法. 通过 CloudSim 的仿真实验结果表明,MRSAG 算法相较于 GA 算法任务完成时间更短,失败率更低. 本文考虑了任务完成时间和结果可靠性的双目标优化,没

有考虑云平台下资源的负载均衡问题,这也是本文下一步的研究重点.

参考文献:

[1] Delavar A G, Aryan Y. HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems[J]. Cluster computing, 2014, 17(1): 129-137.

[2] Sonnek J, Chandra A, Weissman J B. Adaptive reputation-based scheduling on unreliable distributed infrastructures[J]. Parallel and Distributed Systems, IEEE Transactions on, 2007, 18(11): 1551-1564.

[3] Wang X, Yeo C S, Buyya R, et al. Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm[J]. Future Generation Computer Systems, 2011, 27(8): 1124-1134.

[4] Hakem M, Butelle F. Reliability and scheduling on systems subject to failures[C]// Parallel Processing, 2007. ICPP 2007. International Conference on. China, X'ian, IEEE, 2007: 38-38.

[5] Delavar A G, Aryan Y. A goal-oriented workflow scheduling in heterogeneous distributed systems[J]. Int J Comput Appl, 2012, 52(8): 27-33.

[6] Calheiros R N, Ranjan R, De Rose C A F, et al. Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services [J]. Computing Research Repository, 2009, 9(3): 2-10.

作者简介:

黄婷婷 女,(1990-),硕士研究生. 研究方向为人工免疫学. E-mail:huangtingting@whu.edu.cn

梁意文 男,(1962-),教授. 研究方向为人工免疫学、计算机血缘.

(上接第 41 页)

[7] Hogervorst Ron, Tero John P, Eschauzier, et al. A compact power-efficient 3 V CMOS rail-to-rail input/output operational amplifier for VLSI cell libraries[J]. IEEE Journal of Solid-State Circuits, 1994, 29(12): 1505-1513.

[8] Wang Wenbo, Mao Luhong, Xiao Xindong, et al. A differential automatic gain control circuit with two-stage - 10 to 50 dB tuning range VGAs[J]. Journal of Semiconductors, 2013, 34(2):5-9.

作者简介:

马建平 男,(1990-),硕士研究生. 研究方向为模拟/射频集成电路及系统设计. E-mail:majianping@mail.sim.ac.cn.

袁 晨 男,(1989-),博士研究生. 研究方向为模拟/射频集成电路及系统设计.

袁圣越 男,(1986-),硕士研究生. 研究方向为模拟/射频集成电路及系统设计.

田 彤 男,(1968-),博士研究生. 研究员,研究方向为模拟/射频集成电路及系统设计.