计算机测量与控制.2014.22(5)
Computer Measurement & Control

文章编号:1671-4598(2014)05-1537-03　　中图分类号:TP393　　文献标识码:A

基于CT－RAG和学习量子粒子群的云计算任务－资源分配算法

宗　苏
(赣南师范学院科技学院,江西 赣州　341000)

摘要:目前已有的云计算任务－资源分配算法仅针对独立任务进行同构资源分配,同时在分配时未考虑任务优先级;为了克服其缺点,提出了一种基于虚拟CT－RAG(Task－Resource Assignment Graph in Cloud Environment,CT－RAG)和学习量子粒子群的任务－资源分配模型;首先,定义了虚拟CT－RAG图和任务优先级,并描述了采用其获取任务－资源分配方案初始解的方法;然后采用具有学习能力的量子粒子群在可行解空间中寻优,通过为粒子安装学习机,粒子在每轮迭代的过程中根据适应度的变化情况自适应地调整动作选择概率,从而加快获取全局最优解和加快收敛速度;仿真实验表明:文中方法能有效地解决云计算环境下依赖型任务的异构资源调度,获取了全局最优解356.67,较其它方法具有较大的优越性。

关键词:任务－资源分配;云计算;量子粒子群;学习

Task－Resource Allocation Algorithm Based on CT－RAG and Study Quantum－behaved Particle Swarm Algorithm

Zong Su
(Gannan Normal University Institute of Technology, Ganzhou　341000, China)

Abstract: The given task－resource allocation algorithm only considers the independent task and homogeneous resource, and also do not mention task priority. In order to conquer their defects, a task－resource allocation model based on CT－RAG (Task－Resource Assignment Graph in Cloud Environment) and studying Quantum－behaved Particle Swarm was proposed. Firstly, the virtual CT－RAG and task priority was defined, and the initial solution was obtained by using CT－RAG. Then using the studying Quantum－behaved Particle Swarm to search the global optimum solution, every particle was installed studying machine, so it can change the action selection probability according to the fitness and finally converge to the global optimum solution. The simulation experiment shows the result in the solution in this paper can realize task－resource allocation in cloud environment, the optimal solution is 356.67, and compared with other methods, it has larger priority.

Key words: task－resource allocation; cloud computing; quantum－behaved particle swarm; study

0　引言

云计算(Cloud Computing)[1-2]是通过集群技术[3-4]将具有各类性能的资源联系起来,构建大规模的资源池[5-6]。任务－资源分配问题是云计算的一个至关重要度问题,其主要解决将云用户提交的n个任务,通过某种方法分配到处理节点集上执行以达到预定目标,分配的策略直接影响任务执行的效率甚至成败[7]。

现有的针对云计算环境的任务－资源分配工作[8-11]主要有:文献[8]提出了一种基于GA算法和ACO算法的资源分配方法。文献[9]设计了一种动态的任务－资源调度方法,通过应用的负载需求变化来动态和弹性地实现资源配置,为了降低能耗,实时减少物理服务器的资源,其不足是增加了虚拟机动态迁移的开销。文献[10]建立了一种基于免疫优化算法实现云计算任务－资源分配的方法。文献[11]采用服务器的温度控制资源分配方法,以最大化资源利用效率和云应用服务质量作为多目标,并研究了温度感知的应用负载迁移方法,其缺点是多个目标往往相互冲突,没有通过多目标优化来获得最佳的任务－资源分配模型。

上述工作均研究了云计算环境的任务－资源分配方法,但将任务看作互相独立的元任务,没有考虑到任务之间的相互关系,因此,本文设计了一种基于任务－资源分配图T－RAG(Task－Resource Assignment Graph,T－RAG)和学习量子粒子群的任务－资源分配方法。

1　问题描述

任务－资源分配的目标是将当前云用户提交的任务在满足任务优先级的情况下,将其分配给对应的资源来执行,并满足最大完成时间最短。

为了便于描述,假设以下条件成立:

(1) 所有任务均为独立元任务或依赖型任务;

(2) 任务不可再分,且将资源分配给任务开始执行后,任务的执行不可中断;

(3) 任务数量远大于资源数量,采用动态批处理方式,以充分利用资源进行分配;

(4) 任务在某个资源上的执行时间和传输时间可以预先测定。

收稿日期:2014-01-17;　**修回日期**:2014-03-05。

作者简介:宗　苏(1984-),男,江西赣州人,硕士,实验师,主要从事云计算与数据挖掘方向的研究。

2 基于 CT－RAG 的任务－资源分配模型

2.1 CT－RAG 任务－资源分配图定义

定义 1：云计算的任务－资源分配模型可以表示为一个任务－资源分配图 CT－RAG（Task－Resource Assignment Graph in Cloud Environment，CT－RAG），即可以采用七元组表示为：$\Omega=\{T,R,A,E,W,C,ST\}$，并满足：

（1）$T=\{t_1,t_2,\ldots t_m\}$ 为 m 任务组成的任务集合；

（2）$R=\{r_1,r_2,\ldots r_n\}$ 为 n 个资源的集合；

（3）$V=\{v_1,v_2,\ldots,v_m\}$ 为云计算任务－资源分配图 CT－RAG 的顶点集，其中每个顶点 v_i 可以表示为 $v_i=\{t_k,r_f\}$，即将任务 t_k 映射到资源 r_f 上。

（4）E 为云计算任务－资源分配图 CT－RAG 的有向边集，表示任务之间的通信关系和优先级约束，对于任意边 $(v_i,v_j)\in E(1\leqslant i\neq j\leqslant m)$ 表示任务 v_i 为任务 v_j 的前驱，同时将任务 v_i 的前驱任务集记为 $PRE(v_i)$，将任务 v_i 的后继任务集记为 $SUUC(v_i)$

（5）$\boldsymbol{W}=\{\boldsymbol{WV},\boldsymbol{WE}\}$ 为权值向量矩阵，其中 $\boldsymbol{WV}=\{w(t_i)\mid 1\leqslant i\leqslant m\}$ 为任务的计算量矩阵，$\boldsymbol{WE}=\{we(r_j)\mid 1\leqslant j\leqslant n\}$ 为资源的计算能力矩阵。

（6）$\boldsymbol{C}=\{CD,CB\}$ 为数据量和带宽矩阵，CD 中的每个元素 $cd(t_i,t_j)$ 表示依赖型任务 t_i 和任务 t_j 之间的数据传输量，CB 中的每个元素 $cb(r_i,r_j)$ 表示依赖型任务 t_i 和任务 t_j 分配的资源 r_i 和 r_j 之间的网络通信带宽。

（7）$ST=\{st_1,st_2,\ldots st_n\}$ 为 n 个资源的最早可用时间。

一个包含 7 个任务和 6 个处理资源的 CT－RAG 可以表示为如图 1 所示。

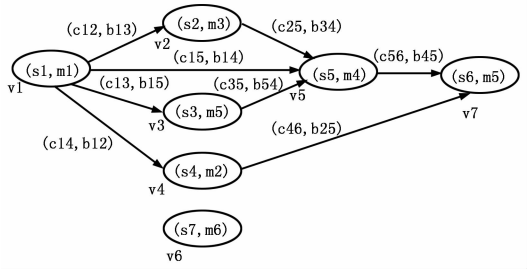

图 1 CT－RAG 任务－资源分配图

在图 1 中，每个顶点 v_i 中描述了任务和对应的资源，任务之间的依赖关系通过有向箭头表示，其中 s1～s5 为依赖型任务，s7 为独立型任务。

2.2 虚拟 CT－RAG 图

为了对依赖型任务和独立型任务进行统一调度，建立两个虚拟的节点即入节点 v_I 和出节点 v_o，并建立 v_I 到原 CT－RAG 图中所有入度为 0 节点的虚线有向边，建立原 CT－RAG 图中所有出度为 0 节点到 v_o 的虚线有向边，新的入节点 v_I 和新的出节点对应的数据量均为 0。图 1 对应的虚拟 CT－RAG 图可以表示为图 2。

任务的 level 可以通过下式计算：

$$level(t_i)=\begin{cases}0 & t_i\text{ 无前驱}\\ 1+\max(level(parent(t_i))) & \text{其它}\end{cases}\quad(1)$$

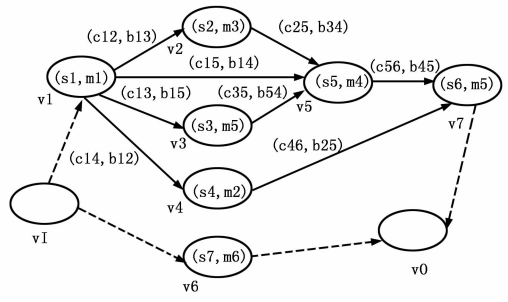

图 2 虚拟 CT－RAG 任务－资源分配图

3 基于虚拟 CT－RAG 和学习量子粒子群的任务资源分配

3.1 目标函数

目标函数为最小化最大完成时间，即：

$$Goal(T)=\min\max_{p}\Big(\sum_i\sum_j w_p(t_i,r_j)+ec_p(t_i,r_j)\Big)\quad(2)$$

式中，p 为关键任务，i 和 j 表示在关键路径上的所有任务和资源的标号，$w(t_i,r_j)$ 表示任务 t_i 在资源 r_j 上的等待时间。

3.2 基于 CT－RAG 图的初始解

采用虚拟 CT－RAG 图来生成初始解的过程为：

（1）根据任务的计算量，随机选择能对其处理的资源初始化 CT－RAG 中的顶点和有向边，建立统一的入节点 v_I 和出节点 v_o，得到虚拟的 CT－RAG 图。

（2）将虚拟 CT－RAG 图的入节点 v_I 从 CT－RAG 图中移除；

（3）寻找虚拟 CT－RAG 图中所有前驱 $PRE(v_i)$ 为空的节点 v_i，将其从虚拟 DAG 图中删除，并将其加入节点集 S；

（4）对任务节点集 S 中的任务按照式（1）计算优先级，将所有任务－资源对按照优先级排序得到就绪任务－资源对队列 R；

（5）取出就绪任务－资源对队列 R 中的队头元素，计算其执行时间和等待时间，得到各任务的完成时间，采用任务的完成时间来重新初始化对应资源的最早可用时间 st；

（6）判断虚拟 CT－RAG 图中是否只剩下出节点 v_o：

如果是，则算法结束，并输出先后执行的任务－资源分配向量；

否则转入步骤（2）继续进行迭代。

3.3 量子粒子群算法

采用学习量子粒子群算法对调度方案进一步寻优，在时刻 t，量子空间中的粒子 s_i 仅有位置向量：

$$x_i(t)=(x_{i1}(t),x_{i2}(t),\ldots,x_{iD}(t))\quad(5)$$

式中，每维分量均代表了任务对应的资源。在时刻 t 和 $t+1$ 时，粒子位置 $x(t)$ 通过蒙特卡洛方法进行模拟可以表示为：

$$\begin{cases}x(t)=x_o(t)\pm\dfrac{L}{2}\ln\left(\dfrac{1}{u}\right)\\ x_o(t)=\alpha p_i(t)+(1-\alpha)p_g(t)\\ L(t+1)=2\beta\mid x_o(t)-x(t)\mid\\ x(t+1)=x_o(t)\pm\beta\mid p_{avg}(t)-x(t)\mid\ln\left(\dfrac{1}{u}\right)\end{cases}\quad(6)$$

式中，$p_i(t)$、$p_g(t)$ 和 $p_{avg}(t)$ 分别表示个体的最优位置、全局最优位置和个体平均最优位置 $p_{avg}(t)$，可以通过下式获得：

$$\begin{cases} p_i(t) = (p_{i1}(t), p_{i2}(t), \ldots, p_{iD}(t)) \\ p_g(t) = (p_{g1}(t), p_{g2}(t), \ldots, p_{gD}(t)) \\ p_{avg}(t) = (\frac{1}{M}\sum_{i=1}^{M} p_{i1}, \frac{1}{M}\sum_{i=1}^{M} p_{i2}, \ldots, \frac{1}{M}\sum_{i=1}^{M} p_{iD}) \end{cases} \tag{7}$$

式中，D 为粒子维数，具体根据粒子编码位数来确定，M 为粒子种群中的粒子数。

3.4　加入学习能力的量子粒子群

在经典量子粒子群算法中，粒子没有学习能力，为了让粒子具有自学习能力，首先为粒子增加一种新的运动方式：高斯运动：

$$x(t) = \frac{ap_i(t) + bp_g(t)}{a+b} \tag{8}$$

式中，a 和为满足高斯分布的随机数。

粒子的学习过程可以描述为：

（1）初始化每个粒子在时刻 t 对应的动作概率向量为［1，1］；

（2）根据式（2）计算粒子在时刻 t 的适应度 $Goal_t$，选择动作概率向量中概率最高的动作为 a，得到 $t+1$ 时的适应度 $Goal_{t+1}$：

如果 $Goal_{t+1} \geqslant Goal_t$，则对动作 a 施加奖励 g，否则对动作 a 施加惩罚－g，将动作选择概率向量中动作 a 的选择概率更新为：

$$pro_j(t+1) = \begin{cases} pro_j(t) + g[1 - pro_j(t)] j = i \\ (1-g)\, pro_j(t) \; \forall\, j \neq i \end{cases} \tag{9}$$

式中，i 表示当前粒子。

（3）重复执行步骤（2），直到所有粒子的适应度变化范围小于某阀值为止。

3.5　基于学习量子粒子群的算法任务－资源分配

初始化：奖励变量 g，惩罚变量－g，高斯随机数 a 和 b，粒子种群规模 M，学习误差阀值 th，当前迭代次数 t，最大迭代次数 T；

（1）随机初始化任务－资源分配对，建立多个虚拟 CT－RAG 图，根据 3.2 获得多个初始解；

（2）将所有任务－资源分配对编码成粒子，粒子维数为任务个数，粒子位置的每维分量的值即为对其分配的资源。

（3）运行 3.4 节的学习算法，获得粒子在各状态的动作概率向量，初始化 $t=1$；

（4）根据当前状态，在对应的动作概率向量中选择的动作，并根据选择的动作即式（6）或（8）来更新粒子的位置，同时更新粒子个体的最优位置 $p_i(t)$、粒子种群的全局最优位置 $p_g(t)$ 以及粒子个体平均最优位置 $p_{avg}(t)$；

（5）$t=1+t$，并重复步骤（4），直到达到最大迭代次数 T 为止，此时算法结束，输出全局最优解 $p_g(t)$。

4　仿真实验

采用云计算仿真工具 CloudSim 对文中方法进行仿真，实验环境参数设置如下：

文中算法的参数设置如下：奖励变量 $g=5$，惩罚变量 $-g=-5$，高斯随机数 $a=2$，$b=1$，粒子种群规模 $M=200$，学习误差阀值 $th=0.1$，最大迭代次数 $T=50$。

采用文中方法求取任务－资源分配方案，并与文献［8］和文献［10］所示的蚁群算法和免疫克隆算法进行比较：蚁群算法的参数如下：蚁群规模 $M=200$，信息素挥发因子 $\rho=0.6$，信息素权重因子 $a=0.6$，启发式信息权重因子 $b=0.4$，迭代次数 $t_{max}=50$。免疫克隆算法的参数如下：抗体种群规模 $M=200$，记忆集 $N=20$，交叉率为 0.53，变异率为 0.62，迭代次数最大值 $t_{max}=200$。

表 1　实验参数

参数名称	取值范围
任务个数(个)	100
资源个数(个)	30
计算速度(MHz)	3 000～4 000
网络速度(Mbps)	500～800
系统负载(%)	20～220
内存负载(%)	30～110
空闲磁盘空间(GB)	80～180
资源单位时间费用(美分)	5～100

采用 3 种方法对表 1 所示的实验参数环境进行仿真，重复 10 次并对其进行平均得到的结果如表 2 所示。

表 2　仿真结果比较

算法	平均值	最差解	最优解	迭代次数
蚁群算法	1 531.43	2 056.21	923.67	165
免疫克隆算法	1 078.46	1 517.82	657.89	132
文中算法	723.92	1 076.34	356.67	59

从表 2 中可以看出，文中方法在迭代次数仅为 59 次时就趋于收敛，获取了全局最优解 356.67，而蚁群算法和免疫克隆算法分别在 165 次和 132 次时才获取全局最优解，且仍未收敛。这是因为文中方法采取虚拟 CT－RAG 图构造初始解，采用量子粒子群在初始解的基础上进行搜索，同时粒子具有学习能力，从而加快了粒子在可行解空间中搜索最优解。

5　结语

为了解决云计算环境下的非独立元任务的异构资源分配问题，以最小化关键路径的最大完成时间为目标，将 CT－RAG 图所获取的解作为初始解，采用具有学习能力的量子粒子群在可行空间不断搜索最优解，实现对初始解的不断改进，最终获的全局最优解作为最终的任务－资源分配方案。仿真实验表明了文中方法能有效地实现云计算环境下的任务－资源分配。下一步的工作是研究满足多个互斥目标的云计算环境下的依赖型任务调度问题。

参考文献：

［1］Ian F，Yong Z，Ioan R，et al. Cloud computing and grid computing 360－Degree compared［A］. Grid Computing Environments Workshop［C］，IEEE，2008.

［2］罗军舟，金嘉晖，宋爱波，等. 云计算：体系架构与关键技术［J］. 通信学报，2011，32（7）：3－21.

［3］刘伯成，陈庆奎. 云计算中的集群资源模糊聚类划分模型［J］. 计算机科学，2011，38（10A）：157－160.

［4］钱琼芬，李春林，张小庆，等. 云数据中心虚拟资源管理研究综述［J］. 计算机应用研究，2012，29（7）：2411－2415.

（下转第 1567 页）

检测结果。

3　实验结果与分析

实验中，首先对系统的调相功能进行仿真，图 4 为时钟存在相位差时的校正过程，其中，BD _ GPS _ 1PPS 为标准时钟信号，OUT _ 1PPS 为调整后的输出时钟，C 为相差检测信号，存在相差时 C 输出高电平，state 为当前状态机的状态。

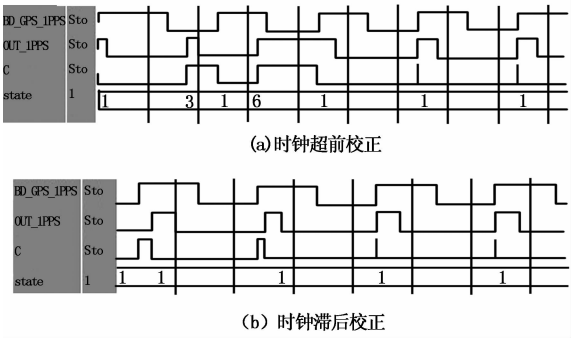

图 4　时钟校正过程

由图 4（a）可知，当系统存在超前相位差时，对输出进行超前调整，在经过两个调整周期后，输出时钟的上升沿即可与标准时钟信号对齐，实现同相输出。

由图 4（b）可知，系统存在滞后相位差时，对输出进行滞后调整，经过两个时钟周期后，输出时钟上升沿可与标准时钟信号同相位，实现滞后校正。

在实际测量中，经过调频和调相处理的输出信号与标准信号的相位比较如图 5 所示。

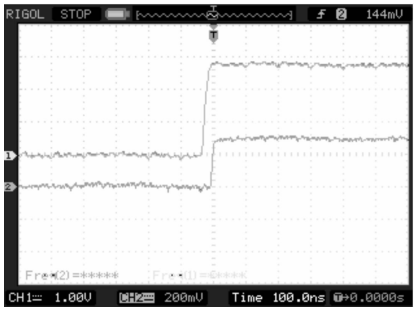

图 5　输出秒脉冲与 GPS 时标的相位比较

由图 5 可知，经过调整后的输出与标准时钟之间的时间误差仅为 20 ns 左右，即 FPGA 的一个工作周期，精确度较高。

表 1 为实际校正过程输出频率和相位差的变化情况。

由表 1 数据可知，该系统具有良好的校正性能，能较快速的减小输出与标准信号的相位差，最终达到与 GPS/北斗标准时钟信号完同步，实现精确守时。

表 1　守时系统的校正过程

校正次数	频率(Hz)	相位差(ns)
1	1.010	$2*10^8$
2	0.990	1 200
3	1.000	40
4	1.000	20
5	1.000	20

经过反复试验，当 GPS/北斗时钟失锁后，系统输出信号的长时间时间误差不大于 0.65 μs/min，符合守时系统不大于 0.92 μs/min 的精度标准。

4　结论

利用 FPGA 和有限状态机的守时系统能够实现对本地时钟信号频率和相位的校正，使其作为标准时间为系统提供精准的秒信号，并能在 GPS/北斗信号失锁后保持长时间的稳定。系统设计中采用的高精度的恒温晶振能够提供稳定的频率输出，有效保证系统的时间精度。同时，FPGA 的使用增强了系统的可靠性和抗干扰能力，简化系统的构成的同时提高了精度，降低系统运行和维护的成本，有很好的实用价值。

参考文献：

[1] 郭　彬. 基于北斗/GPS 双模授时的电力系统时间同步技术研究 [D]. 长沙：湖南大学，2010.
[2] 王新军. 应用 GPS 系统的卫星授时装置研究 [D]. 济南：山东大学，2008.
[3] 邓金根，周　彬，王　健，等. 1588 透传中晶振引入误差的一种改进方法 [J]. 计算机测量与控制，2012，20 (11)：3060－3062.
[4] 原玉磊，夏天倚，陈　渊. 基于单片机晶振的守时研究 [J]. 电子测量技术，2011，34 (11)：20－22.
[5] 李　展，张　莹，周　渭. 基于单片机和 GPS 信号的校频系统 [J]. 时间频率学报，2005，28 (1)：68－75.
[6] 周启民. 高精度守时电路设计及其在导航接收机的实现 [D]. 武汉：武汉理工大学，2008.
[7] 廖　瑛. 基于 GPS 技术与 FPGA 技术的时间统一系统研究与设计 [D]. 武汉：武汉大学，2005.
[8] 刘　进，黄秋元，沈　兵. 自主导航用户机高精度守时电路的设计 [J]. 电子器件，2007，30 (5)：1623.
[9] 郭　峰. 基于 FPGA 的作战系统时统设计 [J]. 电子技术，2009，10：9－11.
[10] 陈贵军. 基于北斗和 GPS 授时系统的研制 [D]. 沈阳：沈阳工业大学，2011.

(上接第 1539 页)

[5] Andreolini M, Casolari S, Colajanni M. Dynamic load management of virtual machines in a cloud architecture [J]. Department of Information Engineering, 2010：201－204.
[6] 王意洁，孙伟东，周　松，等. 云计算环境下的分布存储关键技术 [J]. 2012，23 (4)：962－986.
[7] 李　乔，郑　啸. 云计算研究现状综述 [J]. 计算机科学，2011，38 (4)：32－37.
[8] 王永贵，韩瑞莲. 基于改进蚁群算法的云环境任务调度研究 [J]. 计算机测量与控制，2011，19 (5)：1203－1211.
[9] Lee Y C, Zomaya A Y. Energy efficient utilization of resources in cloud computing systems [J]. The Journal of Super computing, 2010, (53)：1－13.
[10] 孙大为，常桂然，李风云，等. 一种基于免疫克隆的偏好多维 QoS 云资源调度优化算法 [J]. 电子学报，2011，39 (8)：1824－1831.
[11] Tang Q, Gupta S K S, Varsamopoulos G. Energy efficient thermal aware task scheduling for homogeneous high－performance computing data centers：A cyber－physical approach [J]. IEEE Trans. Parallel Distrib. Syst., 2008, 19 (11)：1458－1472.