

基于遗传算法的云计算资源调度策略研究

徐文忠¹, 彭志平², 左敬龙²

(1. 江苏科技大学 计算机科学与工程学院, 江苏 镇江 212003;

2. 广东石油化工学院 计算机与电子信息学院, 广东 茂名 525000)

摘要: 对云计算环境中的资源调度问题进行了研究, 鉴于当前云计算环境中资源利用率不高, 节点负载不均衡的问题, 提出了一种新的基于遗传算法的关于虚拟机负载均衡的调度策略; 根据历史数据和系统的当前状态以及通过遗传算法, 该策略能够达到最佳负载均衡和减少或避免动态迁移, 同时还引入了平均负载来衡量该算法的全局负载均衡效果; 最后通过在 CloudSim 平台进行仿真实验, 结果表明, 该策略具有相当好的全局收敛性和效率, 当系统虚拟机被调度之后, 算法在很大程度上能够解决负载不均衡和高迁移成本问题, 并且极大地提高了资源利用率。

关键词: 云计算; 虚拟机资源; 负载均衡; 遗传算法; 调度策略

Research on Cloud Computing Resource Scheduling Strategy Based on Genetic Algorithm

Xu Wenzhong¹, Peng Zhiping², Zuo Jinglong²

(1. School of Computer Science and Engineering, Jiangsu University of Science And Technology,

Zhenjiang 212003, China; 2. School of Computer and Electronic Information, Guangdong

University of Petrochemical Technology, Maoming 525000, China)

Abstract: The resource scheduling problem in the environment of the cloud computing is studied. In view of the problem that resources utilization rate is not high and node load imbalance problem in current cloud computing environment, this paper presents a new scheduling strategy on load balance of VM (Virtual Machine) based on genetic algorithm. According to historical data and current state of the system and through genetic algorithm, this strategy can achieve the best load balance and reduce or avoid dynamic migration. At the same time, this paper also introduces average load to measure the overall load balance effect of the algorithm. Finally, the experiment simulated on CloudSim platform shows that this strategy has fairly good global astringency and efficiency, and the algorithm of this paper is, to a great extent, able to solve the problems of load imbalance and high migration cost after system VM being scheduled and improve resource utilization greatly.

Keywords: cloud computing; virtual machine resources; load balancing; genetic algorithm; scheduling strategy

0 引言

云计算是目前国内外科研机构和商业机构重点研究的一种新兴计算模式^[1-3], 主要研究的是关于如何将数据中心中的资源虚拟化、如何与调度器实现资源的逻辑整合和满足用户提交的任务需求以及最大化资源利用率等^[4-5], 云中的资源在不违反服务等级协议 (SLA) 的前提下以“按需使用, 按量付费”的原则提供给用户透明使用。随着云计算的日益普及, 云计算面临的一个主要挑战就是管理极大数量的服务请求。因此, 如何科学有效地管理资源以满足用户的需求, 实现负载均衡, 是提高云计算平台性能和服务质量的关键因素^[6]。

云计算中的资源调度^[7]是一个 NP-Hard 问题, 一般解决

这类问题, 人们常使用启发式算法, 如蚁群算法、遗传算法、模拟退火算法等。文献 [8] 在蚁群优化调度策略中引入了双向蚂蚁机制, 通过蚂蚁间的相互交流, 能够快速地发现合适的虚拟机资源, 从而使得 Master 节点能够快速地为用户任务分配虚拟机。文献 [9] 提出了一种具有双适应度的遗传算法, 通过此算法不但能找到总任务完成时间较短的调度结果, 而且此调度结果的任务平均完成时间也较短。本文在遗传算法的基础上, 提出了一种云计算资源调度策略, 能够实现最佳负载均衡以及减少或避免动态迁移。

1 遗传算法简介

遗传算法是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型, 是一种通过模拟自然进化过程搜索最优解的方法, 它最初由美国 Michigan 大学 J. Holland 教授与 1975 年首先提出出来, 后经 DeJong、Goldberg 等人归纳总结所形成的的一类模拟进化算法^[10-12]。遗传算法的基本运算过程如下。

- 1) 初始化: 设置进化代数计数器 $t=0$, 设置最大进化代数 T , 随机生成 M 个个体作为初始群体 $P(0)$ 。
- 2) 个体评价: 计算群体 $P(t)$ 中各个体的适应度。
- 3) 选择运算: 将选择算子作用于群体。

收稿日期: 2014-08-01; 修回日期: 2014-10-13。

基金项目: 国家自然科学基金资助项目 (61272382); 广东省自然科学基金资助项目 (S2012010009963); 广东省高等学校科技创新项目 (2013KJJCX0132)。

作者简介: 徐文忠 (1991-), 男, 硕士, 主要从事机器学习和智能信息处理、云计算资源管理策略方向的研究。

彭志平 (1969-), 男, 教授, 博士, 主要从事多主体技术、机器学习、自主计算、语义 Web 服务和云计算等领域方向的研究。

- 4) 交叉运算：将交叉算子作用于群体。
- 5) 变异运算：将变异算子作用于群体。
- 6) 终止条件判断：若 $t = T$ ，则以进化过程中所得到的具有最大适应度个体作为最优解输出，终止计算。

遗传算法的流程如图 1 所示。

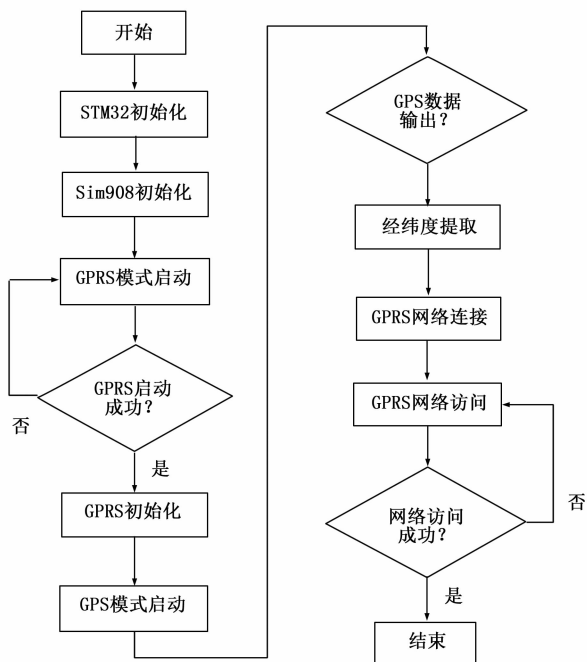


图 1 遗传算法流程

2 基于遗传算法的资源调度策略

2.1 虚拟机调度的模型设计

2.1.1 虚拟机模型

我们把虚拟机和物理机之间看成映射关系，系统中所有物理机的集合为 $P = \{P_1, P_2, \dots, P_N\}$ ， N 表示物理机的数量， $P_i (1 \leq i \leq N)$ 代表第 i 台物理机。我们在物理机 P_i 上命名虚拟机集合为 $V_i = \{V_{i1}, V_{i2}, \dots, V_{im_i}\}$ ，其中 m_i 表示第 i 台物理机上的虚拟机数量。假设我们现在需要部署虚拟机 V ，当把 V 部署给物理机之后，我们使用 $S = \{S_1, S_2, \dots, S_N\}$ 来表示映射方案集。当虚拟机 V 部署给物理机 P_i 之后， S_i 代表映射方案。

2.1.2 负载表达式

一台物理机的负载通常通过添加运行在其中的虚拟机的负载来得到。通过历史监测数据，我们假设最佳时间跨度为 T ，根据物理机负载的变化规律，我们可以把时间 T 分为 n 个时间段。因此，我们在此定义 $T = [(t_1 - t_0), (t_2 - t_1), \dots, (t_n - t_{n-1})]$ ，在定义中， $(t_k - t_{k-1})$ 代表第 k 个时间段。假设虚拟机的负载在每个时间段是相对稳定的，则我们定义虚拟机 i 在时间段 k 的负载为 $V(i, k)$ 。因此，我们可以得出结论，在周期 T 中，物理机 P_i 上的虚拟机 V_i 的平均负载为

$$\overline{V_i(i, T)} = \frac{1}{T} \sum_{k=1}^n V(i, k) \times (t_k - t_{k-1}) \quad (1)$$

因此，我们可以得出物理机 P_i 的负载为

$$P(i, T) = \sum_{j=1}^{m_i} \overline{V_i(j, T)} \quad (2)$$

当前需要部署的虚拟机是 V ，因为当前需要部署的虚拟机的资源信息已经被定义，我们可以根据相关信息估计虚拟机的负载为 V 。因此，当虚拟机 V 部署给物理机时，每台物理机的负载应该为

$$P(i, T)' = \begin{cases} P(i, T) + V \text{AfterDeploy} V \\ P(i, T) \text{Others} \end{cases} \quad (3)$$

通常，当虚拟机 V 部署给物理机 P_i 时，系统负载会有一些的变化。因此，我们需要进行负载调整，以实现负载均衡。当虚拟机 V 部署给物理机 P_i 之后，在时间段 T 中，映射方案 S_i 的负载变化为

$$\sigma_i(T) = \sqrt{\frac{1}{N} \sum_{i=1}^N (\overline{P(T)'} - P(i, T)')^2} \quad (4)$$

其中：

$$\overline{P(T)'} = \frac{1}{N} \sum_{i=1}^N P(i, T)' \quad (5)$$

2.1.3 数学模型

通过之前的分析，我们定义以下数学模型。

定义 1：在系统映射方案 S_i 下，每台物理机的负载为 $P(i, T)'$ ，在时间段 T 中的总负载变化（即平均负载的方差）定义为

$$\sigma_i(S_i, T) = \sqrt{\frac{1}{N} \sum_{i=1}^N (\overline{P(T)'} - P(i, T)')^2} \quad (6)$$

其中，

$$\overline{P(T)'} = \frac{1}{N} \sum_{i=1}^N P(i, T)' \quad (7)$$

定义 2：系统映射方案 S_i 的均衡映射方案为 S_i' ，那么映射方案集 S 的均衡映射方案集为 $S' = \{S_1', S_2', \dots, S_N'\}$ ，其中 S_i' 是使 $\sigma_i(S_i', T)$ 满足预定义负载限制的最佳映射方案。

定义 3：我们把虚拟机的数量 M 和在特定的映射方案中，需要迁移实现负载均衡的总虚拟机数 M 的比值定义为成本 ρ ，对于每个映射方案 S_i ，达到负载均衡 S_i' 所需的成本 $\rho(S_i)$ 定义为：

$$\rho(S_i) = \frac{M}{M} \quad (8)$$

本文的目的就是要找到最佳的映射方案 S_i ，从而达到最佳的系统负载均衡，或者说在负载均衡中最小化成本 $\rho(S_i)$ 。我们可以通过遗传算法从映射方案 S_i 中获得最佳的映射方案 S_i' 。

2.2 基于遗传算法的资源调度策略

遗传算法是一类借鉴生物界的进化规律演化而来的随机搜索方法，它是从代表问题可能潜在的解集的一个种群开始的，初代种群产生之后，按照适者生存和优胜劣汰的原理，逐代演化产生出越来越好的近似解，在每一代，根据问题域个体的适应度大小选择个体，并借助于自然遗传学的遗传算子进行组合交叉和变异，产生出代表新的解集的种群。这个过程将导致种群像自然进化一样，后代种群比前代更加适应于环境，末代种群中的最优个体经过解码，可以作为问题的近似最优解。因此，本文提出了一种新的基于遗传算法的资源调度策略，算法步骤如下：

- 1) 种群编码；

2) 初始化种群:

对于初始化种群, 本文主要使用生成树的方法, 有如下树的定义:

- (1) 树是由物理机集合和虚拟机集中的元素构造生成。
- (2) 树的根节点是预定义管理源节点。
- (3) 所有物理机节点和虚拟机节点都包含在树中。
- (4) 所有的叶子节点是虚拟机节点。

3) 适应度函数评估:

在本文中, 适应度函数为

$$f(S, T) = \frac{1}{A + B \times f_H} \quad (9) \quad f_H = \Phi(\sigma_i(S, T) - \sigma_0),$$

$\Phi(X) = \begin{cases} 1, & X \leq 0 \\ r, & X > 0 \end{cases}$ (10) 其中, A 和 B 是在具体应用程序中定义的加权系数, σ_0 代表系统负载均衡中允许的偏差约束并且可以预定义, $\Phi(X)$ 是惩罚函数。

4) 选择操作:

在当前种群中, 我们通过适应度函数计算出个体的适应度, 并且让个体保持最高适应度进入子代种群, 然后根据它们的适应度值计算个体的选择概率

$$P_i(S) = \frac{f_i(S, T)}{\sum_{i=1}^D f_i(S, T)} \quad (11)$$

其中, $f_i(S, T)$ 代表种群中第 i

个成员的适应度, D 代表种群的规模。

5) 交叉操作:

交叉操作通过互交换和自交换两个选定的个体来产生新个体, 通过杂交, 遗传算法的搜索能力得到了巨大的改善。由于遗传算法采用树编码, 因此为了确保后代染色体的有效性, 这里的算法不能像遗传算法一样使用二进制杂交编码, 即简单地交换部分基因。本文模拟杂交过程, 以确保后代从亲本染色体摄入相同的基因, 并且保证后代树的有效性。操作过程如下:

- (1) 根据旋转选择算法, 选择两个亲本个体 T_1 和 T_2 。
- (2) 结合两个亲本个体形成一个个体树 T_0 。

(3) 对于两个亲本个体中不同的叶子节点, 首先根据每台虚拟机的负载计算出它们的选择概率 p , 然后基于 p 把它们分配给物理机集中具有最小负载的叶子节点, 直到分配完成。

- (4) 重复上述过程, 直到产生达到所需数量的个体。

6) 变异操作:

本文使用以下自适应变异概率

$P_m = \exp(-1.5 \times 0.5t) / D \times \sqrt{M}$ (12) 其中, t 为迭代次数, D 为种群规模, M 为虚拟机数量, 个体是根据变异概率随机选择变化的。

7) 调度策略:

本文的目的是要找到最佳的映射方案, 最大程度满足系统的负载均衡, 或使负载均衡的成本最低。通过遗传算法, 我们希望找到最佳调度方案。首先, 我们根据当前调度方案和最佳调度方案的比值计算出成本基因, 然后根据成本基因决定调度策略, 我们选择具有最低成本的调度方案作为最终调度方案, 以便调度之后对系统负载有最小的影响和达到负载均衡时成本最低。以这种方式, 我们就可以找到最佳调度策略。

3 实验仿真结果及分析

实验采用了澳大利亚墨尔本大学研发的云计算仿真平台

CloudSim3.0.3^[13], 并对其进行扩展, 重写了 DatacenterBroker、Cloudlet 等类, 对本文提出的算法进行了模拟仿真。在实验中, 我们选择一台物理机作为主机, 并且主机是根节点, 整个算法用 Java 语言实现, 所用的编程工具为 Eclipse3.7。实验的初始条件: 物理机数量为 5, 起始虚拟机数量为 15, 种群大小 $D = 50$, 交叉概率为 0.9, 变异概率为 0.1。

实验中分别采用了 CloudSim 中现有的轮询调度算法 (Round Robin, RR), 蚁群优化算法 (ACO) 和本文提出的算法, 在参数设置相同的情况下, 调度同等任务量所需执行时间的对比如图 2 所示, 资源利用率的对比如图 3 所示。

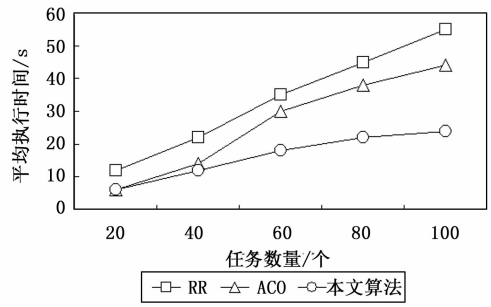


图 2 任务执行时间对比

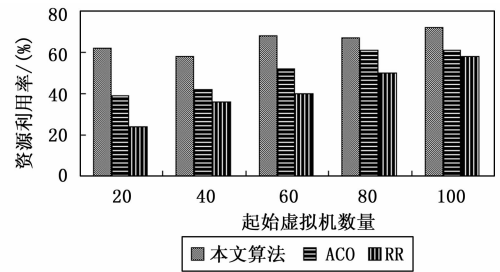


图 3 资源利用率对比

从图 2 可以看出, 当任务数量比较少的时候, 本文算法的优势并不明显, 但是随着任务的数量越来越多时, 本文算法时间增加的幅度要远小于 RR 和 ACO 算法, 因此, 本文提出的算法在任务数量较多时候执行效率较高, 而执行时间要较短。

图 3 反映了不同算法下云环境中资源利用率的情况, 从图中可以看出本文所提出算法的资源利用率最高, 其次是蚁群优化算法, 利用率最低的是轮询调度算法。

4 结束语

本文提出了一种新的基于遗传算法的关于虚拟机负载均衡的调度策略, 该策略根据历史数据和系统的当前状态, 当所需的虚拟机资源部署给每个物理节点之后, 提前计算其对系统的影响, 然后选择对系统具有最低影响的虚拟机部署方案, 最终能够获得最佳负载均衡, 并且能够减少或避免动态迁移, 从而解决了传统调度算法引起的负载不均衡和高迁移成本问题。实验结果表明该算法具有较短的时间跨度, 能够保证任务尽快地完成, 具有良好的负载均衡能力, 有效地提高了资源利用率。下一步的研究工作是结合其他优化算法以及人工神经网络, 希望能进一步缩短用户任务的执行时间和提高资源利用率。

参考文献:

[1] 孙香花. 云计算研究现状与发展趋势 [J]. 计算机测量与控制,

2011, 19 (5): 998-1001.

[2] Foster I, Zhao Y, Raicu I, et al. Cloud computing and grid computing 360-degree compared [A]. Proceedings of the 2008 Grid Computing Environments Workshop [C]. Washington, D. C: IEEE Computer Society, 2008; 1-10.

[3] Armbrust, Fox, Griffith, et al. Above the Clouds: A Berkeley View of Cloud Computing [EB/OL]. [2-010-01-25]. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/E ECS-2009-28.pdf>.

[4] 王庆波, 金 津, 何 乐, 等. 虚拟化与云计算 [M]. 北京: 电子工业出版社, 2009.

[5] Garg S K, Yeo C S, Buyya R, et al. Energy-Efficient Scheduling of HPC Application in Cloud Computing Environment [EB/OL]. (2009-09-07). <http://www.cloudbus.org/reports/EE-Scheduling-AcrossClouds-2009.pdf>.

[6] Cherkasova L, Gupta D, Vahdat A. When virtual is harder than real: Resource allocation challenges in virtual machine based IT environments [R]. California: Technical Report HPL, 2007.

[7] 熊聪聪, 冯 龙, 陈丽仙, 等. 云计算中基于遗传算法的任务调

度算法研究 [J]. 华中科技大学学报 (自然科学版), 2012, 40 (增刊 I): 1-4.

[8] 刘 永, 王新华, 邢长明, 等. 云计算环境下基于蚁群优化算法的资源调度策略 [J]. 计算机技术与发展, 2011, 21 (9): 19-23.

[9] 李建锋, 彭 舰. 云计算环境下基于改进遗传算的任务调度算法 [J]. 计算机应用, 2011, 31 (1): 184-186.

[10] Holland J H. Adaptation in natural and artificial systems [M]. Ann Arbor University of Michigan Press, 1975.

[11] DeJong K A. The analysis of the behavior of a class of Genetic adaptive systems [D]. Ann Arbor University of Michigan, 1975.

[12] Goldberg D E. Genetic algorithms in search optimization and machine learning [M]. Boston: Addison - Wesley Longman Press, 1989.

[13] CloudSim: A Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services Introduction [EB/OL]. <http://www.buyya.com/gridbus/>.

(上接第 1648 页)

在 $31 \mu s$, 而本文算法则在 $25 \mu s$ 。很显然, 本文算法对于同步误差的精确度的改善较为明显, 较 RBS 提升了 168%, 较 ERBS 则提升了 24%。

通过改变 WIA-PA 网络中节点的数量, 分别对三中同步算法进行仿真, 1 小时后得到能量损耗模型中的能量矩阵, 通过分析矩阵各个因子, 得到了 3 种算法对应的能量损耗如图 5 所示。

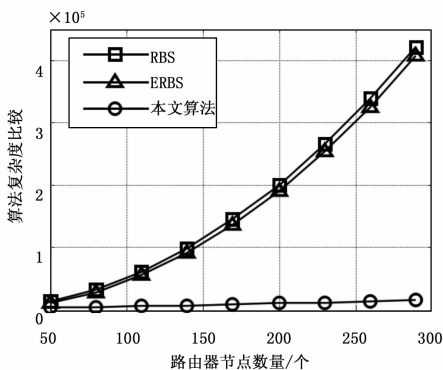


图 5 3 种算法的算法复杂度比较

从图 5 中可以看出 RBS 和 ERBS 的能耗很接近, 因为 ERBS 算法随着节点数目的增加, 只对不相邻的两个节点完成时间信息交换, 节点间交换信息的数量相对于 RBS 算法而言有所减少, 但减少的交换次数甚少。由于本文算法只与相邻的最相关的节点进行信息交换, 所以能耗将大大降低。由图 5 可以看出本文算法的能耗大大低于 RBS 算法和 ERBS 算法, 并且, 随着节点数目的不断增加, RBS 和 ERBS 算法能耗的增长率越来越高, 而本文算法能量消耗的增大趋势则渐进平稳, 很符合 WIA-PA 网络的实际应用需求。

4 结论

本文提出了一种基于能量有效性的局部最优化时间同步算法, 解决了 WIA-PA 网络中大规模节点同步通信的问题。本文算法与传统的 RBS 算法以及改进型 ERBS 算法不同, 该算

法在采用“接收方-接收方”同步的前提下, 节点只与之相邻的节点进行时间戳交换, 并根据时间偏差求取偏差的平均值, 继而对此偏差值进行最大后验估计, 获得时间偏差的估计值。本文算法在通过 OMNeT++ 构建的逼近真实 WIA-PA 网络的环境中进行实测, 取得了与较好的效果。不仅同步的精确度有了明显提升, 同步过程中的能量损耗也被降到了极低的水平, 适合大规模工业应用。

参考文献:

[1] 于海斌, 梁 炜, 曾 鹏. 工业无线网络技术体系与 WIA 标准 [J]. 自动化博览, 2009 (1): 17-20.

[2] Zennaro D, Dall'Anese E, Erseghe T, et al. Fast clock synchronization in wireless sensor networks via ADMM-based consensus [A]. in Proc. 9th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wireless Netw [C]. Princeton, 2011: 148-153.

[3] 杨照峰, 时合生, 樊爱宛. 基于 ZigBee 的低延时 PDA 网络系统设计与实现 [J]. 计算机测量与控制, 2014 (7): 2182-2184.

[4] Wang Y, Qian Z, Wang G, et al. Research on Energy-efficient Time Synchronization Algorithm for Wireless Sensor Networks [J]. Journal of Electronic and Information Technology, 2012 (9): 2174-2179.

[5] Samanta A K, Mukherjee A, Hossein S M. Efficient time synchronization by KD tree in WSN [A]. Computer and Automation Engineering (ICCAE2010) [C]. 2010: 655-659.

[6] Wang Y, Nunez F, Doyle III F J. Energy-efficient pulse-coupled synchronization strategy design for wireless sensor networks through reduced idle listening [J]. IEEE transactions on signal processing, 2012 (10): 127-134.

[7] Elson J, Girod L, and Estrin D. Fine-grained network time synchronization using reference broadcasts [A]. The Fifth Symposium on Operating Systems Design and Implementation, Boston [C]. USA, 2002: 147-163.

[8] 许婧祺, 王 敏. 一种基于网格的无线传感器网络动态分簇路由协议 [J]. 计算机测量与控制, 2012 (6): 1729-1732.

[9] 王 沁, 赵金东. WIA-PA 网络中的无冲突资源调度算法 [J]. 小型微型计算机系统, 2010 (9): 1726-1730.