基于双适应度遗传算法的虚拟机放置的研究

黄兆年 李海山 赵 君
(武汉数字工程研究所 武汉 430074)

摘 要 减少数据中心产生的网络时延以及优化数据中心能源消耗和物理资源的浪费等越来越受到研究者的关注。主要关注数据中心的物理资源的浪费和数据中心产生的网络时延,并且建模一个多目标优化问题:最小化数据中心的物理资源以及数据中心的时延。通过改进型双适应度遗传算法将两个目标同时优化,将其结果与贪心算法进行比较,实验结果表明,此算法优于贪心算法,是云环境下有效的虚拟机放置算法。

关键词 数据中心,遗传算法,双适应度,虚拟机放置
中图法分类号 TP183 **文献标识码** A

Virtual Machine Placement Algorithm Based on Improved Genetic Algorithm

HUANG Zhao-nian LI Hai-shan ZHAO Jun
(Wuhan Digital Engineering Institute, Wuhan 430074, China)

Abstract Reducing the network delay and optimizing energy consumption and resource waste in the data centers have become increasingly important in the world. This paper focused on the resource waste and the network delay in the data centers and modeled the virtual machine placement to solve multi-objective optimization problems, such as minimizing physical machine resources and minimizing total network delay. Through the double-fitness genetic algorithm(CGA), we optimized the two objects at the same time. There is a contrast between CGA and FFD through simulation experiment, and the result is that CGA is better and it is an efficient virtual machine placement algorithm in the cloud environment.

Keywords Data centers, Genetic algorithm, Double-fitness, Virtual machine placement

1 引言

云计算应用[1]已经成为人们研究的重点,Amazon 弹性计算云、Google App engine、IBM 蓝云都提供了很多云计算服务。

作为一项刚刚兴起的技术,学术界虚拟机技术是云计算的基础,云计算[2]在运行成本和可靠性上很有优势。虚拟机的放置[4,6,8]是云计算中很重要问题,本文利用改进后的双适应度的遗传算法来研究虚拟机放置,并通过仿真实验验证该算法和贪心算法的性能差异。

2 云计算中的编程模型

考虑的因素有两个:物力资源的消耗,以及云环境的时延:用户访问部署在云环境下的应用时产生的时延,由两个部分组成:云环境内部处理并且响应服务请求的时延以及请求响应消息在云环境和用户之间的传输时延。

2.1 物理机资源浪费总和模型

为了充分利用多维资源,第 j 个物理机资源浪费定义为处理器利用率及内存利用率的函数,具体描述为:

$$W_j=\frac{|L_j^p-L_j^m|+\varepsilon}{U_j^p+U_j^m} \tag{1}$$

其中,W_j 为第 j 个物理机的资源浪费,U_j^p 为物理机的处理器利用率,U_j^m 为物理机的内存利用率,L_j^p 为物理机的处理器剩余部分,L_j^m 为物理机的内存剩余部分,ε 设置为 0.0001。

2.2 云环境下的时延

如图 1 所示,5 个不同的虚拟机布置到 5 个不同的数据中心上,虚拟机之间没有相互依赖的数据流,则单一服务器 i 的简化时延模型为:

$$Delay_i=D_{i,j+1}\times\frac{Dist_{i,j+1}}{Band_{i,j+1}} \tag{2}$$

其中,$Band_{i,j+1}$ 为虚拟机 i 和 $j+1$ 之间的带宽;$Dist_{i,j+1}$ 为 i 和 $j+1$ 两者之间的物理距离,如果两个虚拟机部署在同一个数据中心,那么物理距离就是 0,否则,两者之间的物理距离就以相互之间的交换机数量为度量;$D_{i,j+1}$ 为两者之间传输的数据量。该应用在云环境内部的总时延可以表示为:

$$f=\min\sum_{i=1}^{n}Delay_i \tag{3}$$

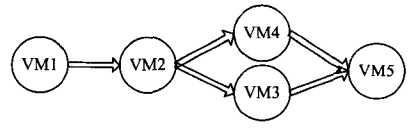


图 1 工作流

2.3 多目标优化

$$Min(W)=\min(\sum_{j=1}^{m}W_j)\sum_{j=1}^{m}(y_i\times\frac{|(\phi_j-\sum_{i=1}^{n}(x_{ij}\times rp_i))-(\varphi_j-\sum_{i=1}^{n}(x_{ij}\times rm_i))|+\xi}{\sum_{i=1}^{n}(x_{ij}\times rp_i)+\sum_{i=1}^{n}(x_{ij}\times rm_i)}) \tag{4}$$

本文受中国科学院重大资助项目资助。

黄兆年(1990-),男,硕士生,主要研究方向为云计算、人工智能,E-mail:hzncd@126.com;**李海山**(1963-),男,博士,研究员,硕士生导师,主要研究方向为计算机网络、容错技术;**赵 君**(1979-),博士,高级工程师,主要研究方向为计算机网络、智能算法。

其中，y_j 表示物理机 j 是否被使用，x_{ij} 表示虚拟机 i 是否放置在物理机 j 上，ϕ_j 表示物理机 j 的处理性能阈值，φ_j 表示物理机 j 的内存性能阈值。

$$Min(\sum_{i=1}^{n} Dela y_i)=\min\sum_{i=1}^{n}(D_{i,j+1}\times\frac{Dist_{i,j+1}}{Band_{i,j+1}}) \tag{5}$$

3 改进遗传算法

遗传算法[3](Genetic Algorithm，GA)是 Holland 于 1975 年受生物进化论的启发而提出的，并行性和全局解空间搜索是 GA 的两个最显著的特点。为了能得到总任务执行时间和任务平均执行时间都较短的任务调度结果，本文对遗传算法作了一些改进，增加了一个适应度，用两个适应度来选择种群，即双适应度遗传算法(CGA)。

3.1 染色体表示

染色体编码有很多种方式，可以采用直接编码，即直接对任务的执行状态编码，也可以采用间接编码。本文采用虚拟机-物理机的间接编码方式。

比如有虚拟机(VM)m 个，物理机(PM)n 个($n\geqslant m$)，假设 $m=20$，$n=20$，染色体的长度为 20，染色体的基因号就是虚拟机的序列 01020304…，基因号为两位数，如果虚拟机序号小于 10，就在前面加 0。

染色体的长度为 20，每个基因值取值范围为 1 到 20，可以产生如下的染色体：

{01,05,04,02,07,08,09,11,14,15,19,20,16,17,18,03,12,06,10,13}，{05,05,12,14,15,14,17,01,02,07,04,13,17,14,15,07,08,04,05,06}

第 1 条染色体表示第 1 个虚拟机放置到第 1 个物理机上，第 4 个虚拟机放置到第 2 个物理机上，……，第 20 个虚拟机放置到第 13 个物理机上。第 2 条染色体表示第 1 和第 2 个虚拟机都放置到第 5 个物理机上，但是必须满足几个虚拟机的 CPU 内核数目加起来小于物理机的内核数，内存总量小于物理机内存。

3.2 初始种群生成

本实验中，染色体中基因的数量为 20，在 1 到 20 中随机取值，随机生成 500 个染色体，但是需要满足公式约束。

3.3 适应度函数

遗传算法是通过适应度函数来进行下一代的选择进化，通过不断迭代找到问题相对而言的最优解。这里研究的是两个目标的多目标优化问题，因此定义两个适应度函数：

$$f_1=1/resouce\cos t(W_j),1\leqslant j\leqslant n \tag{6}$$

表示第 j 个物理机的资源消耗。

$$f_2=1/time\cos t(V_i),1\leqslant i\leqslant m \tag{7}$$

表示第 i 个虚拟机产生的时延。这两个适应度函数有利于算法的收敛速度，最优解就是使物理资源消耗低、虚拟机时延小的解。

3.4 遗传操作

选择操作是通过评价个体染色体的适应度，计算个体染色体的选择概率和累计概率。通过式(6)、式(7)分别计算个体的选择概率 P_1 和 P_2，以 $c1*P_1+(1-c1*P_2)$ 作为选择概率，为下一代相对较优的个体提供优秀的基因基础。

3.5 交叉和变异操作

遗传算法的交叉操作模仿生物进化中的基因重组，保证现有的好基因遗传给下一代个体。本实验的交叉率为 50%，即把任意一个染色体分为两半，然后再进行基因的组合。

变异也是模仿生物种的变异，通过一定概率的变异，可以产生优秀的基因。随机产生一个 0 到 1 之间的数 $math.random()<f=e^{-2*|fitness|}$（$fitness$ 等于两个适应度函数的平均值），以小于 2%的概率进行变异。

4 算法仿真结果与分析

为了验证 CGA 算法的有效性，进行了仿真实验。选取贪心算法与 CGA 算法进行比较。在一台计算机(四核，2.33 GHz 的 CPU 主频，3.25GB 的内存)上采用 Java 语言编程进行仿真，实现了 CGA 算法和贪心算法，使用 Pareto 方法[9]来分析结果。

初始化条件：仿真实验使用 20 个虚拟机，20 个物理机，1 个宽带矩阵，1 个距离矩阵。结束条件：1)达到进化代数 2000；2)结果没变化，则认为算法收敛，终止算法。

从图 2 可以看出，随着迭代次数增加，产生的结果越来越好。从图 3 可以看出，改进后的遗传算法得到的结果比贪心算法好。

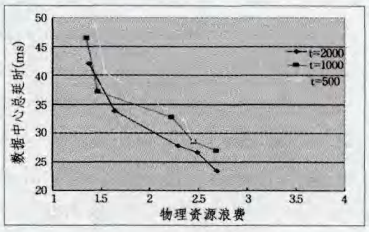


图 2 不同迭代次数

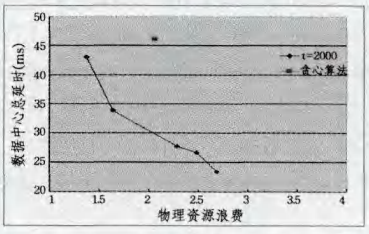


图 3 贪心算法和遗传算法

结束语 本文通过间接染色体编码，改进双适应度遗传算法，利用 Perato 结果分析方法来研究云环境下的数据中心中的网络时延和物理机资源浪费问题。通过改进后的遗传算法，可以得到一个相对较好的虚拟机放置最优解。

参 考 文 献

[1] 李进超，梁谨. 虚拟机动态资源分配及放置算法研究[D]. 上海：复旦大学，2014

[2] Mohammad H，Sun Xin，Sung Yu-wei，et al. Cloudward Bound：Planning for Beneficial Migration of Enterprise Applications to the Cloud[J]. Proceeding of Sigcomm，2010，40(4)：243-254

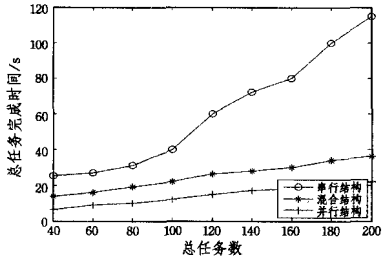
[3] 王小平，曹立明. 遗传算法[M]. 西安：西安交通大学出版社，2002

[4] 李剑锋，彭舰. 云计算环境下基于改进遗传算法的任务调度算法[J]. 计算机应用，2011，31(1)：1001-9081

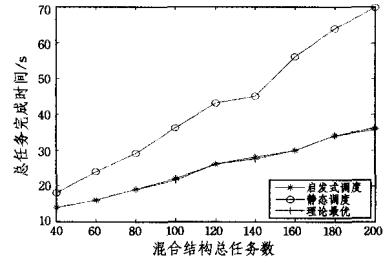
[5] Vasileios P，Zhang Li. Improving the Scalability of Data Center Network with Traffic-aware Virtual Machine[C]// Proc. of IEEE INFOCOM'10. San Diego，USA：IEEE Press，2010

(下转第 416 页)

较。静态执行方案中，采用中型标准实例执行所有任务。本文的算法整体来说接近最优，且方便可行。串行结构、并行结构，以及混合结构的总任务执行时间结果如图 3 所示。从对比结果中可以看出，在相同的资源限制条件下，所提的启发式调度方案，理论上接近于最优方案，理论最优通过数学仿真软件实现；相对静态方案来说，可以获得更短的执行时间，且随着任务数目的增多，两种方案之间的差距也更加明显，本文算法更适合大规模任务的执行。静态方案的任务完成时间往往很长，这是因为它只采用了一种 VM 来执行所有任务，任务不能以分布式的方式得到有效处理。另外从图 3(a)中可以看出，采用同样的方案执行所有任务，混合结构往往小于串行结构，大于并行结构，从而证实了程序的加速往往依赖于并行化所占的比例。并行化程度越高，程序的执行越快，但混合结构更适合于工作流应用。



(a)3 种结构对比



(b)混合结构不同算法对比

图 3

另外，本文还采用文献[16]中视频检索结构的工作流进行仿真，所有视频序列和数据库取自 TRECVID2009。如图 4 所示，同样可以看出所提最优方案处理视频任务时执行时间更短，且随着序列数的增大，两者之间的差距变大，说明本文所提方法更适合于大型多任务多媒体应用。

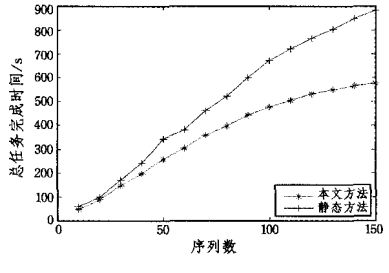


图 4　各方案处理视频任务的时间对比

结束语　本文对云中多媒体应用的最优任务级调度问题进行了研究。首先引入了有向无环图来描述任务之间的优先限制条件，并根据此模型，对 3 种结构中的任务进行了有效的任务优化调度。使得在资源成本限制下，总任务执行时间达到近似最优化。实验表明，在 VMs 资源有限的情况下，所提最优任务级调度方案能够有效满足任务调度实际需求，获得最短的执行时间，更适合于云中多媒体应用。

参 考 文 献

[1] 刘培松. 云计算环境下任务调度和资源分配策略的研究[D]. 上海：华东师范大学，2013

[2] Zhu W，Luo C，Li S. Multimedia cloud computing [J]. IEEE Signal Processing Magazine，2011，28(3)：59-69

[3] 晏婧. 适用于实例密集型云工作流的调度算法[J]. 计算机应用，2011，30(11)：1-3

[4] 晏婧. 云环境下基于 QoS 约束的工作流任务调度算法研究与实现[D]. 重庆：重庆大学，2011

[5] Nan X，He Y，Guan L. Optimization of workload scheduling for multimedia cloud computing [C]// IEEE International Symposium on Circuits and Systems(ISCAS). 2013：2872-2875

[6] Nan X，He Y，Guan L. Optimal resource allocation for multimedia cloud based on queuing model [C] // IEEE International Workshop on Multimedia Signal Processing(MMSP). 2011：1-6

[7] Zhang T J，Li J J，et al. Adaptive resource allocation for cloud computing environments under bursty workloads [C] // IEEE Performance Computing and Communications Conference. 2011：1-8

[8] Hong B，Tang R，Zhai Y. A resources allocation algorithm based on media task Qos in cloud Computing[C]// IEEE Software Engineering and Service Science(ICSESS). 2013：841-844

[9] Guo L，Zhao S，Shen S. Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm [J]. Journal of Networks，2012，7(3)：547-553

[10] Zhao Z，Zhao Y，Gao Z，et al. BUPT-MCPRL at TRECVID 2009 [C]// Proceedings of TRECVID 2009 Workshop. 2009：1-11

[11] 袁浩. 基于社会力群智能优化算法的云计算资源调度[J]. 计算机科学，2015，42(4)：206-208

[12] 王波，张晓磊. 基于粒子群遗传算法的云计算任务调度研究[J]. 计算机工程与应用，2015，51(6)：84-88

[13] Kelley J E. The critical-path method：Resources planning and scheduling [J]. Journal of Industrial Scheduling，1963，37(11)：108-111

[14] Gao Y，Ma H，Zhang H. Concurrency Optimized Task Scheduling for Workflows in cloud [C] // IEEE Sixth International Conference on Cloud Computing. 2013：709-716

[15] Amazon EC2[OL]. http：//aws. amazon. com/ec2/

[16] Over P，Awad G M，Fiscus，et al. TRECVID 2009-goals，tasks，data，evaluation mechanisms and metrics [OL]. http：//www-nlpir. nist. gov/projects/tvpubs/tv. pubs. org. html

(上接第 407 页)

[6] NelSon M，Lim B，Hutchins G. Fast Transparent Migration for Virtual Machines[C]// Proc. USENIX. 2005

[7] Jayasinghe D，Pu C，Eilam T，et al. Improving performance and availability of services hosted on iaas clouds with structural constraint-aware virtual machine placement[C]// IEEE SCC. 2011：72-79

[8] Chen Jian-hai，Kebin C，Ye De-shi. AAGA：Affinity-Aware Grouping for Allocation of Virtual Machines[C]// 27th International Conference on Advanced Information Networking and Applications. IEEE Press，2013

[9] Clark C，Fraser K，Hand S，et al. Live migration of Virtual machine[C]// Proceedings of the 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation. Boston，USA，2005