

基于粒子群优化算法的云计算资源调度策略研究

周丽娟¹ 王春影²

(长春工业大学信息传播工程学院 长春 130012)¹ (长春工业大学应用技术学院 长春 130012)²

摘要 针对移动互联网用户具有移动性的特点,采用移动云的概念来分担计算任务。粒子群算法能够有效地寻找移动互联网的计算资源,从而提高云计算中各个计算资源的分配速度和计算效率。采用粒子群算法,兼顾用户的服务质量,高效调度异构网络中的计算资源,完成具有大计算量的科学计算的云计算资源调度方案。仿真结果表明,所提策略能够提高资源调度的速度,并且能提高云计算的效率。

关键词 移动互联网,云计算,粒子群优化算法,资源调度

中图分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.6.058

Cloud Computing Resource Scheduling in Mobile Internet Based on Particle Swarm Optimization Algorithm

ZHOU Li-juan¹ WANG Chun-ying²

(School of Information Spreading Engineering, Changchun University of Technology, Changchun 130012, China)¹

(School of Applied Technology, Changchun University of Technology, Changchun 130012, China)²

Abstract According to the characteristics of the mobile internet users' mobility, the concept of mobile cloud is used to share the computing tasks. Particle swarm algorithm can effectively find the computing resources in mobile internet, so as to improve the allocation rate of each computing resources in cloud computing and computing efficiency. This article used the particle swarm optimization (PSO) algorithm, took the service quality of users into consideration, scheduled the heterogeneous network computing resources efficiently, and completed cloud computing resource scheduling scheme with a large amount of calculation of scientific computing. Simulation results show that the proposed strategy can improve the speed of resource scheduling, and improve the efficiency of cloud computing.

Keywords Mobile internet, Cloud computing, Particle swarm optimization algorithm, Resource scheduling

1 引言

随着移动业务的发展,移动应用多样化,越来越快的数据传输使得移动互联网已经成为人们生活不可缺少的一部分。传统的分布式计算均是建立在互联网的基础上的。移动互联网无法进行计算主要受到两方面限制,即节点之间的数据传输速度和移动终端的存储和计算能力。然而,随着近年来无线通信技术的发展,节点间的数据传输速度有了很大的提高;并且智能手机取代了传统手机,其存储与计算能力也成倍增长,这就为云计算在移动互联网中的应用提供了硬件保证^[1-3]。

云计算来源于以前的并行计算、分布式计算以及网格计算,但又有别于以上3种计算方式,其具有大型的计算数据中心,可以兼容各种网络形式,具有网络异构性的功能;并且,云计算对多种模型具有鲁棒性,采用虚拟化技术,可以将计算资源虚拟化,从而进行分配,具有更加高效的特点。

针对以上云计算的特点,将云计算引入移动互联网,具有很强的移植性,可以发挥云计算的高效计算能力;同时也为移动互联网的业务推广和创新提供了良好的算法基础。然而,随着云计算的深入研究,其计算资源庞大且分散,并且节点之

间的传输速率、存储和其他方面的条件依然受限,从而影响到云计算的效率。如何能够在其他条件受限的前提下,在数据中心快速地将虚拟数据计算资源分配给用户,以减少用户的运算时间,成为云计算在移动互联网中的一个核心应用问题。

此核心应用问题就是寻找高效的分配虚拟计算资源的计算方法。目前智能的计算方法有许多,如遗传算法、蚁群算法、支持向量机的方法以及本文重点研究的粒子群算法。

蚁群算法和遗传算法均是仿生学优化算法,特点是应用范围广、实用性较强,但是由于其算法效率不高,收敛速度慢,从而在优化的过程中会出现运算速度慢,计算复杂度较高,甚至陷入停止的现象。而支持向量机的算法在高维空间的映射优化算法中具有很强的优势,但是在大量的训练样本条件下,实施存在困难,所以在数据量庞大的移动互联网中,其实施将更加困难,不适用于移动互联网的云计算资源分配。

粒子群算法是根据动物的一些社会行为进行设计的智能算法。此算法可以用于优化非线性系统,并且也很容易与其他优化算法协作使用,故在移动互联网的云计算资源调度中可以广泛应用。

文献[4]提出了用传统的蚁群算法对传统互联网的云计算资源进行分配,但是由于其采用的算法收敛性差,并且随着

到稿日期:2014-12-19 返修日期:2015-01-12 本文受战略性新兴产业管理服务平台建设研究(吉发改投资[2013]1188号)资助。

周丽娟(1971—),女,硕士,副教授,主要研究方向为计算机应用,E-mail:110305606@qq.com;王春影(1971—),女,硕士,高级实验师,主要研究方向为计算机应用、电子商务。

网络节点的增加,分配资源的禁忌传输路径太多,从而使得算法计算时间长,甚至导致整个算法停止,无法达到资源分配的效果。文献[5]采用遗传算法,在保证服务质量的同时,对计算资源进行分配。这种算法以异构和基于服务方向存储作为搜索的起始点。文献[6]提出了优化的工作流任务集群算法,此算法只考虑了优化云计算的计算时间,而忽略了等待时间。故可以看出,此方法考量和优化的指标过于单一,从而整体性能并不能获得实质上的提高。

针对以上分析,本文采用改进的粒子群算法,综合考虑了整个移动互联网的性能,如计算速度、存储、吞吐量以及及时抖动等。另外,综合考虑不同服务业务的服务质量,其所对应的性能指标的要求也不尽相同,故对不同业务具有鲁棒性,也是所提算法要解决的问题。

2 问题描述

本节首先对需要优化和调度的计算资源建立模型,并通过这个模型,建立相关的优化标准。这个模型综合考虑了移动互联网的特点和拓扑结构容易变更的问题,并且在调度中综合考虑了各个优化指标的可变动性,从而适应移动互联网多种不同业务的需求。

2.1 计算资源模型

用 $W=(T,E)$ 代表有向非循环图,其中 $T=\{t_1, t_2, \dots, t_n\}$ 代表需要处理的计算任务集合, E 是有向链路集合。如果 t_i 和 t_j 之间有数据交换,对于 $\{t_i, t_j\}$ 存在联系,则建立链路 e_{ij} 。这使得 t_i 和 t_j 之间可以进行互相的资源调度,并且可以将任务加载到对方。基于这样的定义,子任务直到上级任务完成,才能够进行计算。另外,每个任务工作流程 W 有最后的时间 δ_w ,既为了保证业务的服务质量(Quality of Service, QoS),也是执行时间的限制^[7,8]。

移动互联网的云计算提供了一个多业务处理平台。其中每一个业务处理流程的时间限制不同。若需要对其进行资源调度,则对整个云计算的时间影响十分大,因此需要根据不断变化的拓扑结构,调度各个节点的计算资源,从而提高业务的服务质量。例如,如果一个业务需要 60s 之内完成,就需要对比各个节点的计算速度和传输能量^[9,10]。

设执行任务 t_i 的时间为 T_{t_i} ,其需要的计算量为 C_{t_i} ,此计算量以浮点计算为计算单位,因此可以根据式(1)得到两个变量之间的联系。

$$T_{t_i} = C_{t_i} / (P_i * \rho_i) \quad (1)$$

其中, P_i 为节点总和的计算能力, ρ_i 为节点计算能力中能够完成任务的比例。 TT_{t_i} 为传输所需要的时间,可以由式(2)表示:

$$TT_{t_i} = D_{t_i} / \beta \quad (2)$$

总的处理时间如式(3)所示:

$$PT_{t_i} = T_{t_i} + \sum_{j=1}^K TT_{t_j} * s_j \quad (3)$$

其中,代表参与分配并且共同完成计算任务的节点一共有 K 个, TT_{t_i} 为每个节点需要的时间的 s_j 的比例。

2.2 调度中优化指标的可变动性

自动计算资源调度和分配有不同的优化目标,比如,有些优化目标需要在满足最后的时间的条件下优化,有些优化目标需要执行成本的最小化。本文定义一个调度结果 $S=(R, M, TEC, TET)$,代表资源的一系列的分配解脱,对于一个计算任务,其资源的映射可以分为总的执行成本、总的执行时

间。每个资源 r_i 代表第 i 个任务所占用的计算资源块。根据式(4)和式(5)可以得到总的计算消耗资源的代价 TEC 和总的执行时间 TET 。

$$TEC = \sum_{i=1}^{|R|} C_{VM_{r_i}} * \left[\frac{(LET_{r_i} - LST_{r_i})}{\tau} \right] \quad (4)$$

$$TET = \max\{ET_{t_i} : t_i \in T\} \quad (5)$$

根据以上模型的定义,可以看出,对于计算资源的分配,可以得到资源的最优配置,从而最小化所用资源消耗,即 TEC ,同时保证所用的时间不能超过这个业务的最长时间。故优化的问题可以表示为:

$$\begin{aligned} \min \quad & TEC \\ \text{s. t. } \quad & TET \leq \eta \end{aligned} \quad (6)$$

3 改进粒子群分配算法

传统的粒子群算法是模仿动物种群觅食的行为并进行优化的。其在可行解的搜索当中,每个搜索的路线保持一定的粒度(即一定的方向和距离),同时彼此协作,从而达到了收敛速度快并且并行计算的效果;然而其最大的缺点也正是因为其搜索方向和搜索空间范围难于确定,因此搜索出来的可行解往往是局部最优解。遗传算法能够满足搜索全局最优的可行解,但是由于遗传基因的进化理论,执行时间较长,收敛性能较差。根据两个算法的优缺点,本文提出了一种改进的粒子群算法,此算法吸取了遗传算法搜索范围的设置优点,大大改进了计算资源的分配效率。

定义每一个粒子由搜索的速度和方向表示,每个粒子根据其最优解和局部最优解进行搜索。算法在每个搜索步骤,可以根据其搜索的速度和方向进行求解。在每次迭代中,其迭代的位置和速度可以由式(7)和式(8)表示:

$$x_i(n+1) = x_i(n) + v_i(n) \quad (7)$$

$$v_i(n+1) = \lambda v_i(n) + \beta_1 r_1 (x_i^*(n) - x_i(n)) + \beta_2 r_2 (x^*(n) - x_i(n)) \quad (8)$$

其中, λ 代表取自上个步骤的搜索速度的比重; r_1 和 r_2 为随机数,用于调节方向; β_1 和 β_2 代表加速系数; $x_i^*(n)$ 代表此次搜索的最优解; $x_i(n)$ 代表此次搜索的具体位置解。

在遗传算法中,最主要的是对取向矩阵进行设计,即对各个优化标准的最优解进行比较,将其最优解设置成为取向矩阵的运动方向,做到搜索最优解为全局最优解,避免了传统粒子群算法搜索的解陷入局部最优的情况。

在设置取向矩阵时,需要比较两个目标的重要程度,也就是更贴近全局最优的程度。根据需要,可以建立如式(9)所示的矩阵:

$$C = \begin{bmatrix} 0 & \dots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \dots & 0 \end{bmatrix} \quad (9)$$

为了表示方向,选取-1、0 和+1 作为矩阵元素的取值。若解 i 相对于解 j 更接近最优解,则对其赋值为+1;若为同时接近,对其赋值为0;若远离最优解,则赋值-1。故只有对其进行数次的关系确定,才可以得到取向矩阵,从而保证最优解的搜索方向正确。

对于粒子群算法中的加速系数,可以根据其优化目标随机生成,其主要是对优化目标的执行时间和传输时间进行矩阵化,从而根据其分布,选取概率最大的初值作为初值,并根据其范数决定搜索加速系统。其资源分配的初始值主要有以下两个方面:执行成本和执行时间之间的关系以及传输时间。

对于执行时间,可以用式(10)来表示:

$$ETIME = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{m1} & t_{m2} & \cdots & t_{mn} \end{bmatrix} \quad (10)$$

其中,行代表计算任务,列代表资源,其中的元素值代表计算任务在此资源下的执行时间。定义传输的时间矩阵如式(11)所示:

$$TT = \begin{bmatrix} rt_{11} & rt_{12} & \cdots & rt_{1n} \\ rt_{21} & rt_{22} & \cdots & rt_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ rt_{m1} & rt_{m2} & \cdots & rt_{mn} \end{bmatrix} \quad (11)$$

其中每一行和每一列表示任务转移的时间。而同一个任务转化所需的时间为 0,所以其对角线元素为 0。

根据式(10)和式(11),可以计算加速系数 λ 、 β_1 、 β_2 ,如式(12)一式(14)所示:

$$\beta_1 = \frac{\| (TT)_i \|_2^2 - \| (TT)_j \|_2^2}{\| (TT)_j \|_2^2} \quad (12)$$

$$\beta_2 = \frac{\| (TT)_i \|_\infty^2 - \| (TT)_j \|_\infty^2}{\| (TT)_j \|_\infty^2} \quad (13)$$

$$\lambda = \frac{\| (ETITE)_i \|_\infty^2 - \| (ETITE)_j \|_\infty^2}{\| (ETITE)_j \|_\infty^2} \quad (14)$$

根据以上的设计,可以总结出其资源分配算法,具体执行流程如图 1 所示。

1. 输入:设置本次计算资源分配的任务 T;
初始化资源
选取单位矩阵作为初始化资源,此资源分配方式兼顾了资源分配的公平性。
2. 计算调度算法
 - (1)初始化调度元素;
 - (2)采用式(3)计算执行计算任务的代价。
3. 当 $i=0$ 到 $i=T-1$ 时
 - (1)利用式(10)计算执行时间矩阵;
 - (2)利用式(11)计算传输时间矩阵;
 - (3)利用式(14)计算 λ ;
 - (4)利用式(12)和(13)计算 β_1 和 β_2 ;
 - (5)利用式(7)和式(8)计算搜索向量;
 - (6)利用式(9)计算取向矩阵;
 - (7)筛选搜索向量,确定搜索方向。
4. 利用式(4)计算 TEC。
5. 利用式(5)计算 TET。
6. 计算分配结束。

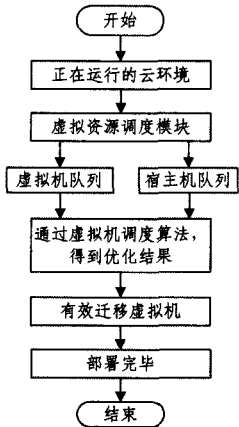


图 1 资源分配执行流程

4 实验结果

本文在计算和仿真的云计算模型为移动互联网的情况下,需要搭建一个云的仿真平台,此云模型平台采用云计算仿真的通用平台 CloudSim,在分布式并行计算的基础上发展而来,利用此平台可以通过计算机的资源模拟数据的存储和传递,但是缺少拓扑变化的环节。根据这个情况,本实验对此进行了修改,根据拓扑图修改其传输的数据和传输的时间。仿真的环境中,计算机的配置环境为:操作系统为 windows7,内存为 8G,硬盘容量为 1T。

仿真流程如图 2 所示,根据虚拟任务和调度来实现模型,其核心算法为开发编写的,调度间隔根据仿真环境的不同,需要单独设置。

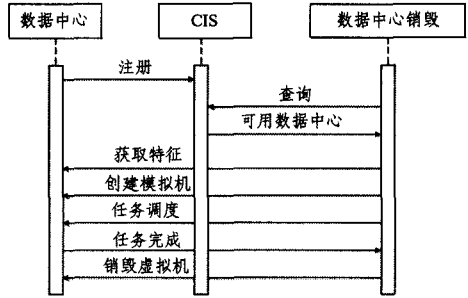


图 2 云计算仿真流程

从图 3 可以看出本文所提的改进型方案在虚拟成本方面的优势,即根据不同的拓扑结构和执行时间,本文所提方案所用的成本均低于传统的粒子群方案;并且随着任务计算量的增加,其优势更加明显。

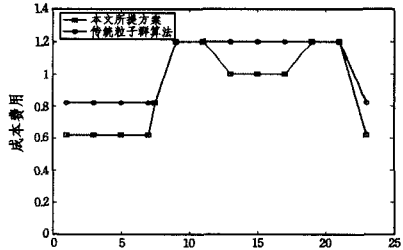


图 3 成本费用对比

图 4 说明了所提算法在资源利用率方面高于传统的粒子群方法。所提方法由于资源利用率高,能够充分发挥分布式计算的优势,从而节约计算成本。

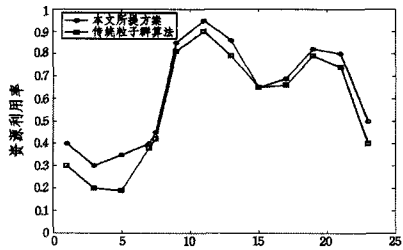


图 4 资源利用率的对比

结束语 本文根据粒子群算法的缺点,即搜索容易造成得到局部最优点,借用遗传算法的优势,通过定义取向矩阵,保证了粒子搜索的范围和方向,从而能够得到全局最优解,提高了移动互联网的云计算的资源分配效率。仿真结果表明了所提方案的优势。

(下转第 292 页)

Lab 色彩空间采用分层聚类来定位统一颜色的文本行。对同一个图像库使用统一的参数检测,对比的文章使用的参数均根据其推荐值确定,本文所提出算法的参数根据经验设定。实验结果如表 1 所列。

表 1 ICDAR 图像库实验结果

Method	R	P	F
Edge1	0.53	0.61	0.57
Edge2	0.67	0.33	0.44
Color	0.60	0.44	0.45
本文提出算法	0.77	0.58	0.62

注:R-召回率,P-精确度,F-综合指数。

通过表 1 中结果可得知,本文提出的基于主动轮廓模型的方法有较好的检测效果,因为在 ICDAR 库中其召回率有较突出的表现。本文在精确度方面还有上升空间,但是由于文本与非文本来自同一个场景,消除非文本时很容易将文本一同删掉,因此这并非易事。

结束语 本文提出了一种主动轮廓模型的文本检测方法,该方法在自然场景和人工场景中均可应用。所提出的方法的初始轮廓不再是一个简单的矩形框,而是让初始轮廓就能够反映图像中的文本信息。这样做不但能得到较好的检测效果,而且检测后将文本单独框出,有利于后续分割识别的进行,使整个文本检测识别过程更加紧密。sobel-laplacian 与 gaussian-laplacian 两步骤的文本预处理,使文本与非文本界限突出,提升了检测结果。实验结果表明,本文所提出的方法效果良好。

参考文献

[1] Liu C, Wang C, Dai R. Text detection in images based on unsupervised classification of edge-based features [C]// Document Analysis and Recognition. 2005;610-614

[2] Mariano V Y, Kasturi R. Locating uniform-colored text in video frames [C]// 15th International Conference on Pattern Recognition. 2000;539-542

[3] Cai M, Song J, Lyu M R. A new approach for video text detection [C]// 2002 International Conference on Image Processing.

2002;117-120

[4] Shivakumara P, Phan T Q. A laplacian approach to multi-oriented text detection in video [J]. IEEE Transactions on Software Engineering, 2011, 33(2)

[5] Zhang Kai-hua, Zhang Lei, Song Hui-hui. Active contours with selective local or global segmentation: a new formulation and level set method [J]. Image and Vision Computing, 2010, 28(4):668-676

[6] Zhu G P, Zhang S H Q, Zeng Q S H, et al. Boundary-based image segmentation using binary level set method [J]. Optical Engineering, 2007, 46(5):050501

[7] Li C M, Xu C Y, Gui C F, et al. Level set evolution without re-initialization: a new variational formulation [C]// IEEE Conference on Computer Vision and Pattern Recognition. San Diego, 2005;430-436

[8] Li C M, Kao C, Gore J, et al. Implicit active contours driven by local binary fitting energy [C]// IEEE Conference on Computer Vision and Pattern Recognition. 2007

[9] Xu C Y, Yezzi J R A, Prince J L. On the relationship between parametric and geometric active contours [C]// Processing of 34th Asilomar Conference on Signals Systems and Computers. 2000;483-489

[10] 刘海波, 沈晶, 岳振勋. Visual C++ 数字图像处理技术详解 [M]. 北京:机械工业出版社, 2014

Liu Hai-bo, Shen Jing, Yue Zhen-xun. Digital image processing using visual C++ [M]. Beijing: China Machine Press, 2014

[11] 蒋先刚. 数字图像模式识别工程软件设计 [M]. 北京:中国水利水电出版社, 2008

Jiang Xian-gang. Digital image pattern recognition engineering software design [M]. Beijing: China WaterPower Press, 2008

[12] Lucas S M. ICDAR 2005 text locating competition results [C]// Eighth International Conference on Document Analysis and Recognition, 2005(ICDAR 2005). 2005

[13] Chen D, Odobez J M, Thiran J P. A localization / verification scheme for finding text in images and video frames based on contrast independent features and machine learning [J]. Image Communication, 2004, 19:205-217

(上接第 281 页)

参考文献

[1] Dinh H T, Lee C, Niyato D, et al. A survey of mobile cloud computing: architecture, applications, and approaches [J]. Wireless communications and mobile computing, 2013, 13(18):1587-1611

[2] Garg S K, Versteeg S, Buyya R. A framework for ranking of cloud computing services [J]. Future Generation Computer Systems, 2013, 29(4):1012-1023

[3] Iosup A, Epema D. On the Gamification of a Graduate Course on Cloud Computing [C]// The International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2013

[4] Venkata Krishna P. Honey bee behavior inspired load balancing of tasks in cloud computing environments [J]. Applied Soft Computing, 2013, 13(5):2292-2303

[5] Ryan M D. Cloud computing security: The scientific challenge, and a survey of solutions [J]. Journal of Systems and Software, 2013, 86(9):2263-2268

[6] Szymanski T H. Low latency energy efficient communications in global-scale cloud computing systems [C]// Proceedings of the 2013 workshop on Energy efficient high performance parallel and distributed computing. ACM, 2013;13-22

[7] Pérez O, Amaya I, Correa R. Numerical solution of certain exponential and non-linear Diophantine systems of equations by using a discrete particle swarm optimization algorithm [J]. Applied Mathematics and Computation, 2013, 225:737-746

[8] Mandal D, Kar R, Ghoshal S P. Digital FIR filter design using fitness based hybrid adaptive differential evolution with particle swarm optimization [J]. Natural Computing, 2014, 13(1):55-64

[9] Belmecheri F, Prins C, Yalaoui F, et al. Particle swarm optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows [J]. Journal of intelligent manufacturing, 2013, 24(4):775-789

[10] Katherasan D, Elias J V, Sathiyar P, et al. Simulation and parameter optimization of flux cored arc welding using artificial neural network and particle swarm optimization algorithm [J]. Journal of Intelligent Manufacturing, 2014, 25(1):67-76