

自适应 CPSO 算法在云计算任务调度中的应用

张晓丽

(西安航空学院 计算机学院, 陕西 西安 710077)

摘要: 云计算环境中的任务调度问题一直是云计算研究的重点。为了克服传统 PSO 算法易陷入局部最优的缺陷, 针对云计算的编程模型框架, 将混沌优化搜索技术应用于粒子群优化算法, 提出一种基于 Tent 映射的自适应混沌粒子群任务调度算法。该算法结合了 PSO 算法的快速收敛性和混沌运动的遍历性、随机性, 在初始化粒子的位置时采用混沌赋值的方式产生一系列初始解, 再根据每个粒子个体的适应值来自适应地调整其惯性权重, 对陷入局部最优的粒子群个体位置进行混沌更新, 帮助其跳出局部最优。通过在 CloudSim 平台进行仿真对比实验, 表明该改进算法有效地缩小了总任务的完成时间, 具有较好的寻优能力和实时性, 是一种有效的云计算任务调度算法。

关键词: 云计算; 任务调度; 粒子群优化算法; 混沌

中图分类号: TP393

文献标识码: A

文章编号: 1673-629X(2016)08-0161-05

doi: 10.3969/j.issn.1673-629X.2016.08.034

Application of Self-adaptive Chaos Particle Swarm Optimization in Task Scheduling for Cloud Computing

ZHANG Xiao-li

(School of Computer, Xi'an Aerotechnical University, Xi'an 710077, China)

Abstract: Tasks scheduling is an important issue to be resolved in cloud computing research. In order to overcome the defects of traditional PSO easy to fall into local optimum, in view of procedure model framework of cloud computing, the chaos optimization search technique is applied to the particle swarm optimization, and an adaptive chaotic particle swarm algorithm of task scheduling based on Tent mapping is presented. It combines the fast convergence of PSO and the ergodic property of chaotic motion, with chaotic assignment way in initializing particle position, and then adaptively adjusts its inertia weight according to the fitness value of each individual particle, updating chaos location for particle swarm individual to help themselves escape from local optima. Through simulation experiment on the CloudSim platform, the results show that the ACPOS, with a good real-time performance and optimization ability, significantly reduces the completion time of the task, which is an efficient task scheduling algorithm.

Key words: cloud computing; task scheduling; Particle Swarm Optimization (PSO); chaos

0 引言

云计算^[1-2]作为一种新型的计算服务商业模式,集成了网格计算、分布式计算和并行计算等技术,自提出以来就成为一个热点研究方向。云计算提供了基础设施、平台和服务,可以将用户提交的任务按照适当的算法分配到虚拟计算资源上进行合理调度,之后再将服务器的计算分析处理结果发送给用户。

由于云计算中存在诸多形态不同的云端,且面对的计算资源种类众多、提供服务效率不一、计算任务数量巨大,因此在“云”中如何将众多任务进行合理调度以满足用户对服务质量的要求,并且更有效利用云计

算系统中的资源,一直是一个热点问题^[2-3]。云计算任务调度是云计算研究的重点,它主要解决的问题是如何以最优的调度策略实现用户任务的有效调度。云计算任务调度问题实际上是一个 NP 难题^[4]。针对该问题,国内外学者对其进行了大量的研究,提出了众多云计算任务调度算法。如最小最小 (Min-Min)^[5]、Sufferage 算法^[6]、蚁群算法^[7]、遗传算法^[8]、粒子群优化算法^[9]等。

PSO 算法^[10]是一种基于群体的智能优化算法,从随机解出发,通过迭代寻找最优解。相比较于遗传算法或蚁群算法等全局优化算法,PSO 算法具有设置参

收稿日期:2015-10-28

修回日期:2016-01-28

网络出版时间:2016-06-22

基金项目:西安市科技计划项目资助(CXY1518(1))

作者简介:张晓丽(1980-),女,讲师,硕士,研究方向为云计算、分布式计算。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20160622.0844.032.html>

数少、易实现、精度高、收敛快等优点,在求解云计算任务调度问题上具有天然的优势。但是同其他一些智能优化算法一样,PSO 算法也存在着容易出现早熟收敛、陷入局部最优等缺点。

为了克服传统 PSO 算法易陷入局部最优解的缺陷,文中通过对如何充分利用“云”中的资源使其中的任务进行高效合理的调度进行了研究,利用云模型的随机性和稳定性的特点,在传统 PSO 优化算法的基础上提出了一种基于云模型的自适应 CPSO 任务调度算法,采用混沌变量来优化粒子的位置和速度,并对每次混沌搜索的范围采用自适应调整方法,从当前群体中择优选择部分粒子进行混沌优化。

1 云计算中的任务调度模型

目前,大部分云计算平台都采用 Google 提出的 Map/Reduce 分布式计算编程模型^[11]。这是一种用于大规模数据集的并行编程架构,分为 Map 和 Reduce 两个阶段,实现将一个较大的任务切分为多个较小的子任务,交给 n 台计算机并行执行,再并行递归并返回结果,最后得到运算结果。

云计算环境下的任务调度是以任务为出发点,根据一定的任务调度策略,将相互独立的多个任务以最合理的方案分配到可用的计算资源上,使得任务的总执行时间跨度最短同时能够充分利用现有资源。因此云计算任务调度算法的优劣直接影响任务的执行效率,从而影响整个云的性能和用户的满意度。根据云计算任务调度的特点,定义如下的数学参数模型: $G = (T, R, F, P)$ 。

其中, $T = \{T_1, T_2, \dots, T_n\}$ 表示任务模型,包括 n 个相互独立的任务; $R = \{R_1, R_2, \dots, R_m\}$ 表示由 m 个可用于执行任务的高性能计算资源组成的计算资源模型,其中 $m < n$; F 表示云计算资源分配和任务调度的目标函数; P 为调度策略。

定义了以上的任务调度模型,使用 $n \times m$ 的 ETC 矩阵来描述各个计算资源上任务队列运行完成所需要的时间。其中, $ETC(i, j)$ 表示第 j 个计算资源上执行第 i 个子任务所需要的时间。

假设第 j 个资源上分配的的子任务的个数为 M , 那么资源 j 上的所有子任务执行完成所需要的时间应该为:

$$\text{Time}(j) = \sum_{i=1}^M \text{ETC}(i, j) \quad (1)$$

可以得到完成所有任务的总时间为:

$$\text{SumTime} = \max_{j=1}^m (\text{Time}(j)) \quad (2)$$

则任务的平均完成时间为:

$$T_{\text{avg}} = \text{SumTime}/n \quad (3)$$

2 标准粒子群算法

粒子群优化算法^[10]是 Kennedy 和 Eberhart 在 1995 年受飞鸟集群活动的规律性启发提出的一种全局优化进化算法。该算法通过目标函数适应度的值来分配相关资源。由于该算法简单易实现且没有太多参数需要调整,因而在求解云计算任务调度这个大规模、实时问题时能得到较好的效果。

PSO 同遗传算法类似,是一种基于迭代的优化算法。传统 PSO 算法中,每一个粒子代表一个可行解,且每一个粒子均有一个初始位置和速度,粒子的优劣程度取决于待优化问题目标函数确定的适应度值。系统初始化为一组随机解,通过迭代在解空间追随最优的粒子进行搜索,从而找到最优值。在每一次迭代中,粒子通过跟踪两个“极值”来更新自己,其中一个就是粒子本身所找到的最优解,即个体极值 $pbest$,另一个则是整个种群目前找到的最优解,即全局极值 $gbest$ 。在找到这两个最优值时,粒子根据式(4)和式(5)来更新自己的速度和位置:

$$v(t+1) = wv(t) + c_1 r_1 (pbest(t) - x(t)) + c_2 r_2 (gbest(t) - x(t)) \quad (4)$$

$$x(t+1) = x(t) + v(t+1) \quad (5)$$

其中, v 为粒子的速度; x 为粒子的位置; c_1 和 c_2 为加速因子; r_1 和 r_2 为 $[0, 1]$ 的随机数; w 为惯性权重。

3 自适应 CPSO 算法

3.1 CPSO 算法

混沌是存在于非线性系统中的一种无规则的运动现象,混沌优化算法的基本思想是把混沌变量从混沌空间映射到解空间。由于混沌变量具有遍历性、随机性和规律性,利用混沌变量进行优化搜索能加速算法搜索速度、避免算法陷入局部最优,从而提高算法的搜索性能。

标准 PSO 算法是一种简洁高效的智能优化算法,但是由于算法随机性很大,仍有很多不完善的地方,例如容易出现收敛过早、陷入局部最优等问题。为弥补该缺陷,文中将混沌思想引入粒子群优化算法,提出一种混沌粒子群优化算法。

在算法初始化时利用混沌赋值的方式对粒子群进行初始化,同时在算法搜索过程中将混沌范围扩大到优化变量的取值范围中,充分利用 PSO 算法收敛速度快以及混沌运动遍历性、随机性和规律性的优点,对种群中处于最差位置的粒子个体进行混沌优化,从而提高传统粒子群算法的收敛速度和寻优效率。不同的混沌映射方程对混沌寻优过程有很大的影响。文献

[12]通过比较指出,相比 Logistic 映射,Tent 映射具有更快的迭代速度和更高的遍历均匀性,因此文中采用 Tent 映射作为混沌优化模型,其表达式为:

$$z_{k+1} = \begin{cases} 2z_k, & 0 \leq z_k \leq 1/2 \\ 2(1 - z_k), & 1/2 < z_k \leq 1 \end{cases} \quad (6)$$

文中提出的基于 Tent 映射的混沌粒子群(CPSO)算法的基本思想是:当 PSO 算法出现早熟收敛时,引入混沌 Tent 映射,对全局最优粒子进行混沌搜索,通过改变粒子个体位置的更新策略来引导粒子群进行一定次数的混沌更新,使其快速跳出局部最优。当算法出现早熟收敛时,使用式(7)对位置变量进行更新:

$$x(t+1) = x_{\min} + z_{k+1} \times (x_{\max} - x_{\min}) \quad (7)$$

3.2 自适应惯性权重

PSO 算法的性能优劣与参数惯性权重 w 的取值有着密切的关系。 w 取值越小越有利于局部搜索,并且能得到较为精确的解,但搜索时不易跳出局部极值点;相反, w 取值越大越有利于全局搜索,但不易得到精确的解。对此,为了加强算法在全局和局部搜索之间的平衡,文中结合 CPSO 优化算法,提出一种基于粒子个体适应值的自适应惯性权重:

$$w(t) = w_{\min} + (w_{\max} - w_{\min}) \times 2^{\exp(\frac{t}{T-G})} \quad (8)$$

其中, w_{\max} 为惯性权重的最大值; w_{\min} 为惯性权重的最小值; t 为当前的迭代次数; G 为最大迭代次数。

可见,通过式(8)来计算每个粒子的惯性权重系数,对每次混沌搜索的范围采用自适应调整方法。在算法初期 w 取较大的值,使算法具有较强的全局搜索能力;而算法后期 w 使用较小的值,从而提高算法的局部搜索能力。

3.3 适应度函数的选取

在 PSO 算法中,适应度函数的选取是相当重要的。适应度函数直接影响算法的收敛速度与最优解的查找,适应度值越大,表示相对应粒子质量越好。云计算任务调度目标是总任务完成时间最短,文中根据任务总完成时间来定义适应度函数:

$$fit = \frac{1}{SumTime} \quad (9)$$

文中采用类似遗传算法的选择思想,优先选择适应度高的粒子。由式(9)可知,任务总完成时间越小的粒子,适应度值越大,则越容易被选择。

3.4 算法流程

文中将混沌优化思想用于粒子群优化算法,并结合动态惯性权值因子,提出一种自适应的混沌粒子群优化算法。以粒子变量为基础,在初始化粒子群中粒子的位置时采用混沌赋值的方式,对粒子产生混沌扰动形成混沌序列,以此替换原有粒子,提高粒子性能。算法在初始化粒子位置时采用混沌赋值的方式产生一

系列初始解,从中择优选出初始群体,提高粒子群的多样性和粒子搜索的遍历性。其次在粒子群优化过程中,根据每个粒子个体的适应值来自适应地调整其惯性权重,对粒子群个体位置进行混沌更新,帮助其跳出局部极值区间,提高了算法的收敛速度和精度。

具体实现步骤如下:

(1)采用混沌理论产生初始群体。利用混沌优化的随机性、遍历性特点对粒子群算法进行混沌初始化,从中择优选择粒子作为初始粒子,在保证随机性的前提下使初始粒子尽可能分布均匀。

(2)混沌初始化各粒子的位置和速度,根据式(9)计算粒子群的适应度值。

(3)根据式(8)自适应地调整粒子的惯性权重。

(4)对于每个粒子的适应度值与原来的个体极值 pbest 和全局极值 gbest 进行比较,如果粒子的适应度值优于 pbest,那么将其作为当前的个体极值;如果粒子的适应度值优于 gbest,则将其作为当前的全局极值点。

(5)对每个粒子,按式(4)更新速度,按式(5)和式(7)更新位置。

(6)根据适应度值的大小从中选出最优的前 m 个粒子,对其进行混沌优化。将 $gbest(i) (i = 1, 2, \dots, m)$ 映射到 Tent 方程的定义域 $[0, 1]$ 内进行迭代,产生混沌变量序列 $z_i^m (m = 1, 2, \dots)$,然后通过逆映射将混沌序列返回原解空间,得到原解空间每一个可行解 $g_{best}^m = (g_{best}^m(1), g_{best}^m(2), \dots)$,计算其适应值,从而得到性能更优的 m 个可行解 g_{best}^* 。

(7)用 g_{best}^* 取代当前群体中按照适应度值排序最差的 m 个粒子的位置。

(8)判断是否满足停止条件,如果满足则停止搜索,输出全局最优位置,否则返回步骤(4)。

4 实验结果及分析

为了检测文中算法的有效性,采用墨尔本大学开发的云仿真平台 CloudSim^[13]来模拟一个云计算的局部环境,在相同实验条件下,对传统 PSO 算法和文中算法进行了云计算环境下的仿真对比实验,分别从收敛性和任务完成时间两个方面对算法性能进行对比。

(1)算法的收敛性对比。

在云计算资源数为 10,调度任务数为 100 的环境下,基本参数设置一致,对文中算法和基本 PSO 算法调度方案的收敛性进行仿真实验,收敛曲线如图 1 所示。

由图 1 可知,相较于基本 PSO 算法而言,文中方法的收敛速度明显占据优势,并且比基本 PSO 算法优

先达到最优方案。这是因为在迭代过程中,由于文中方法引入混沌机制对基本的粒子群进行混沌扰动,避免粒子陷入局部最优,并且在迭代中采用了动态自适应的惯性权重机制,提高了算法的寻优能力。

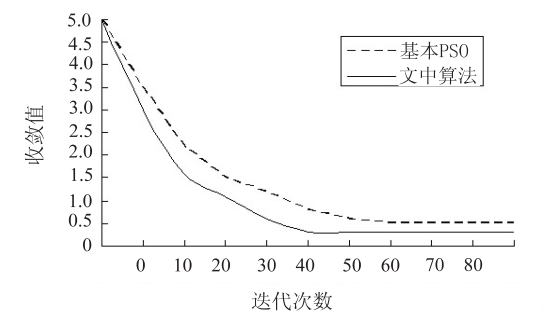


图 1 算法改进前后的收敛曲线图

(2)总任务完成时间对比。

考虑到云环境任务具有动态性的特点,分别模拟 10 个计算资源、50 个子任务以及 10 个计算资源、500 个子任务两种实验环境,对基本 PSO 算法和文中算法的性能进行了对比实验。两种算法分别运行 100 次,取 100 次实验的平均结果作为实验的最终结果。根据文献[14]的参数调整原则,设置两种算法的基本参数,如表 1 和表 2 所示。

表 1 文中算法主要参数设置

项目	取值
种群规模	300
最大迭代次数 G	200
w_{\min}	0.3
w_{\max}	0.9
$c_1 = c_2$	2

表 2 PSO 算法主要参数设置

项目	取值
种群规模	300
最大迭代次数 G	200
惯性权重 w	0.9
$c_1 = c_2$	2

在参数设置基本一样的情况下,分别进行 200 次迭代,两种调度算法的对比结果如图 2 和图 3 所示。

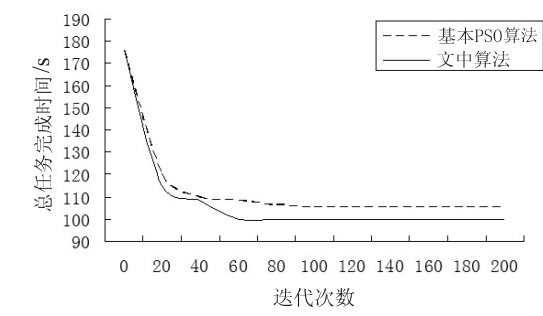


图 2 任务较少时,两种算法总任务完成时间比较

从图 2 和图 3 可以看出,基本 PSO 算法和文中算法在迭代初期任务的总完成时间相差不大,随着迭代次数的增大,两种算法的任务总完成时间都在不断减小。相比而言,由于基本 PSO 算法只重视总任务的完成时间,在迭代初期收敛速度快,但在运行过程中由于无法有效跳出局部最优而出现了早熟收敛,过早地收敛到局部最优的任务调度结果。因此采用基本 PSO 算法的调度优化效果,迭代初期比较明显,但是随着迭代次数的增加,探索能力不足,在进化前期迅速地探索到局部最优解后就没再跳出。而文中算法将混沌机制引入基本 PSO 算法,在迭代过程中不断优化调度结果。对比图 2 和图 3 可以看出,当任务数增加到 100 时,使用文中算法得到的总任务完成时间明显优于使用基本 PSO 算法调度的总任务完成时间,且收敛速度也优于传统的 PSO 调度算法。

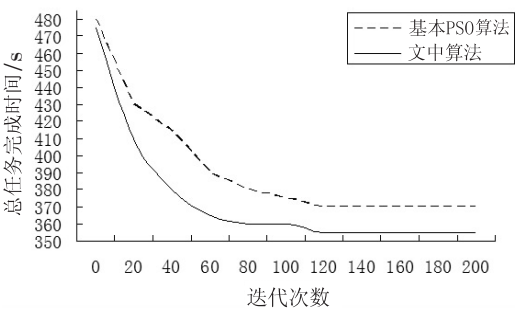


图 3 任务较多时,两种算法总任务完成时间比较

(3)任务平均完成时间对比。

为了更好地验证和评价文中算法的性能,将其与改进遗传算法 (GA)^[7]、改进蚁群算法 (ACO)^[15] 进行对比试验。设置计算资源节点数为 10,待调度的子任务数为 100,在相同的实验环境下,分别测试三种算法迭代 100 次后得到的任务平均完成时间,对比结果如图 4 所示。

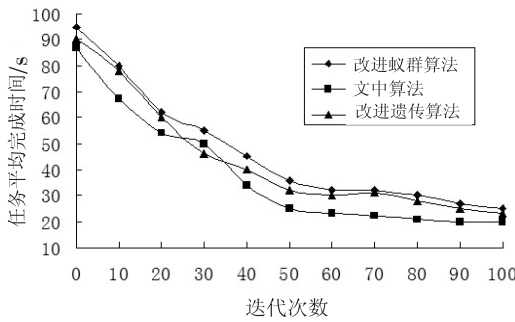


图 4 任务平均完成时间比较

从图中可以看出,文中的 CPSO 算法结合了 PSO 算法的快速收敛能力以及混沌优化算法的遍历性和随机性特点,能够有效降低算法陷入局部极小的概率,在迭代过程体现了均衡的探索能力和开发能力,在三种调度算法中表现最优。ACO 算法在整个迭代过程中,迭代初期优化效果比较明显,但是在迭代后期优化效

果变弱,任务平均完成时间大于遗传算法和文中算法,调度效果相对较差。GA 算法在整个迭代过程中优化能力比较均衡,但是收敛速度相对较慢。因此,无论是任务的平均完成时间还是算法的收敛速度,文中提出的 CPSO 算法都优于 ACO 算法和 GA 算法。

5 结束语

针对当前云计算任务调度算法存在收敛速度慢、资源利用率不足等缺陷,利用 PSO 算法控制参数少、易于实现、计算简单等优点,文中提出一种云计算环境下基于混沌粒子群优化的任务调度算法。在 PSO 算法中引入混沌搜索,采用混沌赋值的方式初始化种群,并使用自适应的惯性权值控制粒子的搜索范围,使粒子自适应地进行搜索。通过对比实验可知,文中提出的基于 Tent 映射的自适应 CPSO 算法很好地解决了标准 PSO 算法容易“早熟”的问题,提高了任务和资源之间的匹配度,并且有效缩短了总任务的完成时间,具有较好的寻优能力,能够有效地解决云计算环境下的任务调度问题。

参考文献:

[1] Foster I,Zhao Yong,Raicu I,et al. Cloud computing and grid computing 360-degree compared[C]//Proc of grid computing environments workshop. Washington D. C. ,USA:IEEE Computer Society,2008:1-10.

[2] Armbrust M,Fox A,Griffith R,et al. Above the clouds;a berkeley view of cloud computing[EB/OL]. 2009-11-21. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>.

[3] 邹永贵,万建斌. 云计算环境下的资源管理研究[J]. 数字通信,2012(4):39-43.

.....

(上接第 160 页)

[27] 孔艺权. 基于语义云实验资源共享平台的研究[J]. 实验室研究与探索,2012,31(7):216-219.

[28] 张继东. 云计算环境下的数字图书馆语义服务架构研究[J]. 图书情报工作,2013,57(16):130-134.

[29] 张琼妮. 网络环境下区域协同创新平台模式与机制及政策研究[D]. 杭州:浙江工商大学,2014.

[30] Di Modica G,Tomarchio O. A semantic model to characterize pricing and negotiation schemes of cloud resources[C]//Proc of the third international conference on cloud computing, GRIDs, and virtualization. Nice, France:IARIA,2012:18-23.

[31] Sheth A,Ranabahu A. Semantic modeling for cloud computing, part 2[J]. IEEE Internet Computing,2010,14(4):81-84.

[32] Dastjerdi A V,Tabatabaei S G H,Buyya R. An effective architecture for automated appliance management system applying

[4] Arfeen M A,Pawlikowski K,Willig A. A framework for resource allocation strategies in cloud computing environment[C]//Proc of IEEE 35th annual computer software and applications conference workshops. [s. l.]:IEEE,2011:261-266.

[5] 罗红,慕德俊,邓智群,等. 网格计算中任务调度研究综述[J]. 计算机应用研究,2005,22(5):16-19.

[6] Buyya R. Economic-based distributed resource management and scheduling for grid computing[D]. Melbourne, Australia: Monash University,2002.

[7] 王芳,李美安,段卫军. 基于动态自适应蚁群算法的云计算任务调度[J]. 计算机应用,2013,33(11):3160-3162.

[8] 林剑柠,吴慧中. 基于遗传算法的网格资源调度算法[J]. 计算机研究与发展,2004,41(12):2195-2199.

[9] 季一木,王汝传. 基于粒子群的网格任务调度算法研究[J]. 通信学报,2007,28(10):60-66.

[10] Kennedy J,Eberhart R. Particle swarm optimization[C]//Proc of IEEE international conference on neural networks. Piscataway,NJ:IEEE,1995:1942-1948.

[11] Dean J,Ghemawat S. MapReduce: simplified data processing on large clusters[C]//Proceedings of the 6th symposium on operating system design and implementation. New York:ACM,2004:137-150.

[12] 单梁,强浩,李军,等. 基于 Tent 映射的混沌优化算法[J]. 控制与决策,2005,20(2):179-182.

[13] Calheiros R N,Ranjan R,Beloglazov A,et al. CloudSim:a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms[J]. Software: Practice and Experience,2011,41(1):23-50.

[14] 段海滨,张祥银,徐春芳. 仿生智能计算[M]. 北京:科学出版社,2011:63-85.

[15] 李建锋,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):184-186.

.....

ontology-based cloud discovery[C]//Proc of 2010 10th IEEE/ACM international conference on cluster,cloud and grid computing. Melbourne, Australia:IEEE,2010:104-111.

[33] Ngan L D,Kanagasabai R. OWL-S based semantic cloud service broker[C]//Proc of 2012 IEEE 19th international conference on web services. Honolulu,HI:IEEE,2012:560-567.

[34] Fortis Teodor-Florin,Munteanu V I,Negru V. Towards an ontology for cloud services[C]//Proc of 2012 sixth international conference on complex,intelligent, and software intensive systems. Palermo:[s. n.],2012:789-791.

[35] Giakoumis D,Mavridou E,Votis K,et al. A semantic framework to support the management of cloud-based service provision within a global public inclusive infrastructure[J]. International Journal of Electronic Commerce,2015,20(1):142-173.