

基于语义搜索引擎的云资源调度*

黎明¹, 吴跃^{1,2}, 陈佳^{1,3}

(1. 电子科技大学 计算机科学与工程学院, 成都 611731; 2. 电子科技大学无锡研究院, 江苏 无锡 214135;
3. 美国加州大学欧文分校 电子工程学院, 加州 92697)

摘要: 针对现有云资源调度存在的非自动调度、云资源的非优化使用等问题, 提出了采用语义搜索引擎(semantic search engine, SSE)实现云资源调度的有效方法。SSE是由美国加州大学欧文分校与电子科技大学联合开发的, 它利用云资源与云用户请求的语义信息构造相应本体, 进行资源的较优匹配, 自动、快速地为云用户需求建立一个优化的解决方案。首先, 给出了基于SSE的云资源调度体系结构; 其次, 给出了SSE的体系结构、SQDL & SCDL匹配器的本体构造方法。仿真实验表明, 基于SSE的云资源调度比传统的云资源调度的点对点通信性能提高了约15%, 子任务执行时间减少了约18%, 表明它是一种可行且高效的云资源调度方法。

关键词: 云计算; 资源调度; 语义搜索引擎; 本体; 对象关系数据库

中图分类号: TP393.07

文献标志码: A

文章编号: 1001-3695(2015)12-3735-03

doi: 10.3969/j.issn.1001-3695.2015.12.048

Cloud resource scheduling based on semantic search engine

Li Ming¹, Wu Yue^{1,2}, Chen Jia^{1,3}

(1. School of Computer Science & Engineering, University of Electronic Science & Technology of China, Chengdu 611731, China; 2. Wuxi Research Institute, University of Electronic Science & Technology of China, Wuxi Jiangsu 214135, China; 3. School of Electrical Engineering & Computer Science, University of California, Irvine 92697, USA)

Abstract: None of existing cloud systems is both being an automated scheduling and considering the optimal usage of resources. To address these problems, this paper proposed a cloud resource scheduling strategy using semantic search engine (SSECRS). SSE was a new type of service search engine developed by University of California, Irvine and University of Electronic Science and Technology of China. SSE took full advantage of the semantic information between cloud resources and cloud users' requirements to construct the appropriate ontology to complete the appropriate match. SSE built an optimal solution according to users' requirements automatically and quickly. Firstly this paper presented the architecture of cloud resource scheduling based on SSE as well as the architecture of SSE. Then it described the ontology construct in SQDL & SCDL match. The experiment shows point to point communication performance in SSECRS improves about 15% and reduces about 18% sub-tasks execution time comparing to traditional cloud resource scheduling (TCRS). Hence it is a feasible and efficient cloud resource scheduling method.

Key words: cloud computing; resource scheduling; semantic search engine (SSE); ontology; object-relational databases

0 引言

云计算^[1]是网格计算、分布式计算、并行计算^[2]、效用计算、网络存储、虚拟化、负载均衡等传统计算机技术与网络技术发展融合的产物。云计算的核心思想是将大量用网络连接的资源统一管理和调度, 构成一个资源池向用户按需服务^[3-6]。云计算资源^[7-10]具有分布性、异构性和动态性, 使得云计算资源调度比分布式计算环境或集群计算环境下的资源调度更加复杂, 不仅要支持跨组织或管理域的任务调度, 实时监控资源和作业执行的状态, 而且要维护局部的站点自治, 提供相应的QoS支持。因此, 对云资源调度的研究已成为云计算中的关键问题。

目前, 虽然已经存在一些开源的云资源调度系统, 如 Eucalyptus、Open Nebula、Nimbus^[11-13], 但是 Eucalyptus 使用贪婪

和轮盘赌算法, 该算法是随机的, 对于用户的请求, 它随机地选择可用资源, 不会考虑物理资源的最优使用; Open Nebula 使用排队系统, 提前预留以及抢占调度, 它未考虑物理资源的使用率; Nimbus 使用一些可定制的工具, 如 PBS 和 SGE, 它是基础的排队系统, 未能提供自动的优化调度, 未能考虑资源的最优使用。因此, 现有的云资源调度系统没有同时考虑自动调度和资源的优化使用。

针对上述问题, 本文提出一种基于语义搜索引擎(semantic search engine, SSE)的云资源调度策略。语义搜索引擎是由美国加州大学欧文分校语义计算实验室与电子科技大学联合开发的。它实现了语义计算理论, 为用户提供了友好的问题驱动的界面来搜索资源, 自动、快速地根据云用户的需求找到所需且较优的云资源。它能自动处理和整合在云上公布的信息, 分别抽取云资源与云用户请求的语义信息来构造本体, 并

收稿日期: 2014-12-08; 修回日期: 2015-02-10 基金项目: 国家自然科学基金资助项目(61103111); 国家科技型中小企业创新基金资助项目(14C26213201050); 无锡市政产学研合作项目(CMES3C1101)

作者简介: 黎明(1977-), 男, 四川眉山人, 工程师, 博士, 主要研究方向为云计算、语义网络等(dawn33@163.com); 吴跃(1958-), 男, 四川南充人, 教授, 硕士, 主要研究方向为网络计算、数据库系统等; 陈佳(1980-), 女, 四川营山人, 副教授, 博士, 主要研究方向为云计算、语义计算等。

利用本体在已知的可用云资源中找出和提交的云用户请求最匹配的资源,以满足最优调度。

1 基于 SSE 的云资源调度体系结构

其体系结构分为三层,即云用户层、资源调度层以及云资源层,如图 1 所示。

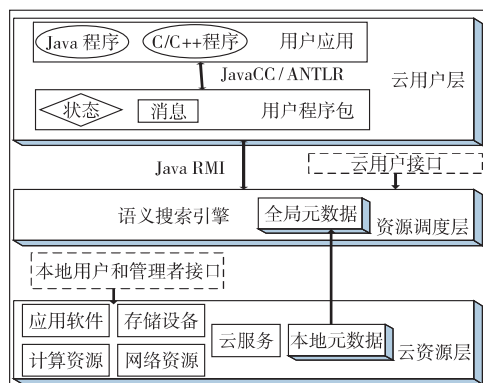


图 1 基于 SSE 的云资源调度体系结构

1) 云用户层 各种各样的应用都集中在这一层实现,该层的需求就是云系统要提供的功能,它直接影响着云系统要达到的目标。它处在云层次结构的最上层,通过应用支持环境和下层的实现交互。云用户层看到的只是应用支持环境提供给自己的视图,用户不需要了解云和各种服务的实现细节就可以开发自己的应用。

为实现云用户层的应用功能,需要将云用户层细分为两部分,即用户应用以及用户程序包。用户应用主要是用户提交的程序。在用户程序的连续计算块中必须插入诸如 For/While 的循环来表示相应状态。要达到插入状态标志的目的,就需要加入预编译工具: ANTLR 和 JavaCC。前者用于 C/C++,后者用于 Java 的预处理。对于用 Java 编写的用户程序,针对该模型分布式并行计算的特性,采用基于 Java 的执行方式,即采用 Java 虚拟机的方式运行于 JPVM 平台下,调用已经存在的 C/C++ 库模块和可执行代码,屏蔽节点异构特性,与各异构节点合作进行计算任务;对于用 C/C++ 编写的用户程序,有两种执行方式,一是调用 JNI 接口将其转换成相应的 Java 代码,按照上面的方式运行;二是采用本地执行方式,直接地使用 PVM 和 MPI 完成编译并直接执行代码。

用户程序包主要用来监控与其相对应的用户程序的执行状态以及从用户进程到其所有基于 TCP/IP 协议标准的消息通道。它必须提供状态捕获以及传递消息的方法,此方法能够在用户程序里被调用。为了达到这一目标,将用户程序划分成一个由多种方法构成的集合,每种方法命名为 func_n(n 是从 1 开始的整数)并且返回下一个方法要调用的索引值。用户程序包调用这些方法,并在每个方法调用结束时产生一个快照。

2) 资源调度层 它是本体系结构的核心层,由语义搜索引擎实现。

3) 云资源层 它是云系统的硬件基础,包括各种计算资源,如超级计算机、贵重仪器、可视化设备、现有应用软件等,这些计算资源通过网络设备连接起来。其基本功能就是控制局部的资源,向上提供访问这些资源的接口。它是云作用域内所有连接到网络上的信息。它包括主机名、IP 地址、体系结构类型、内存大小、主频等。通常使用局部元数据和全局元数据来

完成语义匹配。

2 语义搜索引擎(SSE)

SSE 是由美国加州大学欧文分校语义计算实验室与电子科技大学联合开发的。SSE 实现了语义计算理论,为用户提供了一个友好的问题驱动的界面来搜索资源,自动、快速地根据用户的需求建立一个较优的解决方案。

2.1 语义搜索引擎体系结构

语义搜索引擎体系结构如图 2 所示。

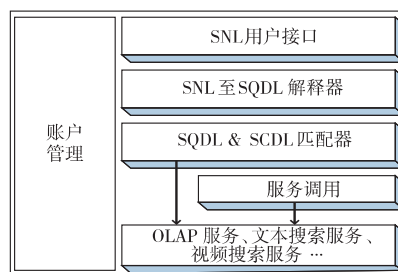


图 2 语义搜索引擎体系结构

1) SNL(structured natural language,结构化自然语言) 用户接口为用户提供了一个友好的查询界面,通过该接口,用户可以采用结构化的自然语言组合自己的查询。

2) SNL 到 SQDL 解释器转换一条 SNL 查询到 SQDL(service query description language,服务查询描述语言)。SQDL 是一种机器可处理的查询语言,它被 SSE(semantic search engine,语义搜索引擎)用来描述用户目的。通过使用 SQDL,用户查询可以统一进行以进一步处理。

3) SQDL & SCDL 匹配器匹配 SQDL 到 SCDL(service capability description language,服务能力描述语言)的描述。SCDL 是使用一种声明性语言来描述每个服务的能力。给定一个 SQDL 查询,匹配试图找到服务列表。如果没有一个单一的服务能满足这个要求,匹配器会分解 SQDL 查询成几个简单的查询,并设法找到一系列可能满足查询的服务。本系统中,抽取云资源与云用户请求的语义信息来构造本体,并利用本体在已知的可用云资源中找出与提交的云用户请求最匹配的资源,以满足最优调度。

4) 服务调用者调用匹配的服务,代表用户得到最终的解决方案。该调用是专为小型服务提供商设计的,因为他们未必能买得起昂贵的基础设施,但他们可以将自己的服务整合至基础设施。

5) 服务可以用来解决特定的问题。服务提供商使用关于每一个服务的 SCDL 描述,可以在 SSE 中注册他们的服务,并利用这些服务组成不同用户需求的解决方案。

6) 账户管理提供了访问控制和计费服务。SSE 使用云计算的计费策略。用户必须拥有一个有效账户来使用 SSE,当满足用户要求的服务被选择时,用户在使用它之前可能需要付费。

2.2 SQDL & SCDL 匹配器

SQDL & SCDL 匹配器是语义搜索引擎的核心,其匹配效率对整个引擎的性能起到决定性的作用。引入语义计算,充分利用云资源与云用户请求的语义信息来构造相应本体,进行资源的较优匹配。

各种本体的构造主要依赖于的一组规则,通过这些规则作用

于领域知识 构成领域本体。为进一步减少本体构造的时间, 提高本体构造的效率, 选择对象关系数据库作为本体构造的数据源。云资源都遵循该本体的规范进行描述, 使得云资源之间最大限度地实现共享与重用。

定义 1 本体可以用一个二元组 $O = (C, R)$ 表示, 其中: $C = (C_1, C_2, \dots, C_m)$ 表示概念的有限集合, $R = (R_1, R_2, \dots, R_n)$ 表示概念之间各种关系的有限集合, R_i 是一个有向关系, 是 C 的笛卡尔乘积的子集, 即 $R_i \in C \times C$ 。

定义 2 对象关系^[14] 记为 OR_i , 由一组属性 A_1, \dots, A_m 和元组 t_1, \dots, t_n 组成, 用 $Attr(OR_i)$ 表示对象关系 OR_i 中的所有属性, $Dom(A_i)$ 表示属性 A_i 的值域, $t_k(A_i)$ 表示元组 t_k 在属性 A_i 上的值; $OR_{id}(OR_i)$ 表示能唯一标志对象关系 OR_i 的一个对象, $referattr(OR_i)$ 表示一个特殊类型对象的参考属性。

1) 类及类层次构造规则

规则 1 对 ORDB schema 中的对象关系, 凡 OR_{id} 相等的对象关系都可以整合成一个本体类 C_i ; 凡 OR_{id} 不相等的对象关系, 则各自构成单独的本体类 C_i 。

规则 2 对于对象关系 OR_i 以及 OR_j , $t_i(A_i)$ 表示元组 t_i 在属性 A_i 上的值, 对于在 OR_i 中的每一个 $t_i(A_i)$, 若 $t_i(A_i) = t_j(A_j)$ 成立, 则 $OR_i(A_i) \subseteq OR_j(A_j)$ 成立。若规则 1 不满足, 但是公式 $OR_i(OR_{id}(OR_i)) \subseteq OR_j(OR_{id}(OR_j))$ 及 $A_i \in Attr(OR_i)$ 成立, 则对应于 OR_i 的类是对应于 OR_j 的类的子类。

2) 属性构造规则

规则 3 对于两个对象关系 OR_i 以及 OR_j , 若 $A_i = referattr(OR_i)$ 成立, 以及 A_i 是 OR_j 的参考属性, 则对象属性 P 能够基于属性 A_i 建立。

规则 4 对于两个对象关系 OR_i 以及 OR_j , 若它们满足下面两个条件:

a) $|OR_{id}(OR_i)| > 1$;

b) $referattr(OR_i) \subset OR_{id}(OR_j)$, $referattr(OR_i)$ 是 OR_i 相对于 OR_j 的参考属性, 则两个本体对象属性 has-a 以及 is-a 能够独立建立。

规则 5 对于三个对象关系 OR_i, OR_j 以及 OR_k , 若 $A_k \in Attr(OR_k)$ 且 $(A_k \subset OR_{id}(OR_i)) \cap (A_k \subset OR_{id}(OR_j))$ 成立, 则两个对象属性 P_1 和 P_2 能够基于 OR_k 建立。

规则 6 对于本体类 C_i, C_j 的数据类型属性集合表示为 $DS(C_i)$, 若 C_i 对应于数据库中的对象关系 OR_1, OR_2, \dots, OR_i , 则对于在 OR_1, OR_2, \dots, OR_i 中的每一个属性, 若它不能被创建对象属性, 则它能被创建成数据类型属性。

根据上述本体构造规则, 产生相应的本体, 当 SSE 收到一个云用户请求的时候, SSE 就根据相应的本体, 搜索满足于该请求的最优或较优资源, 实现资源的优化调度。

3 仿真实验分析与结果

为了验证提出的基于语义搜索引擎的云资源调度的可行性以及有效性, 本文利用 CloudSim 云计算仿真工具进行了仿真实验, 选择 CloudSim 3.0.2 作为软件仿真工具。本文模拟 100 个云计算用户请求任务需要在部署有 10 个虚拟机的云计算系统中进行资源调度的情况。比较了基于语义搜索引擎的云资源调度模型 (semantic search engine-based cloud resource scheduling, SSECRS) 与传统的云资源调度 (traditional cloud resource scheduling, TCRS) 在点对点的通信性能以及子任务执行

时间两方面的性能。

在点对点通信实验中, 通过在两个不同的云计算系统中的计算节点来回地发送消息来比较点对点的通信性能, 消息大小在 1 ~ 1200 KB 进行变化。点对点通信性能比较结果如图 3 所示。从图 3 可以得出, SSECRS 策略比 TCRS 策略的点对点通信性能提高了约 15%, 尤其当消息大小超过 1 024 KB 时, 通信带宽增加的比率越大, 表明 SSECRS 策略比较适合云计算系统这种需要较大带宽的应用场合。

在子任务执行时间比较实验中, 子任务数目在 1 ~ 100 个进行增长, 两种云资源调度策略的子任务执行时间的比较结果如图 4 所示。从图 4 可以得出, 随着子任务数目的增加, SSECRS 策略的执行时间比 TCRS 策略的执行时间增长速度减慢; 尤其是当子任务数目达到 60 个以上时, SSECRS 策略比 TCRS 策略平均减少了约 18% 的子任务执行时间。

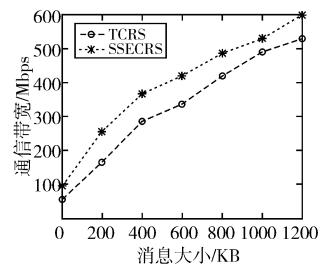


图3 点对点通信性能比较

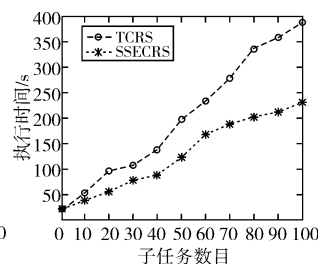


图4 子任务执行时间比较

以上仿真实验结果与结果分析充分说明, 语义搜索引擎的引入为多用户的云计算环境提供了一种可行且有效的资源调度策略, 它能够有效地提高其调度效率, 是云计算资源调度的一种较优方法, 很适合应用到真实的云计算环境中。

4 结束语

针对云资源的动态、异构、分布以及复杂等特点, 总结了现有云资源调度系统存在的非自动调度、云资源的非优化使用等问题, 提出一种基于语义计算理论的语义搜索引擎, 充分利用云资源与云用户请求的语义信息来构造相应的本体, 通过本体匹配来完成云资源的动态发现与调度策略。语义搜索引擎 (SSE) 是由美国加州大学欧文分校语义计算实验室与电子科技大学联合开发的, 它为用户提供了一个友好的问题驱动界面来搜索资源, 自动、快速地根据云用户的需求建立一个较优的解决方案。仿真实验结果表明, 基于 SSE 的云资源调度在点对点通信性能以及子任务执行时间方面均优于传统云资源调度, 是一种切实可行的调度方法。如何更好地设计 SSE 中的匹配算法, 将是笔者下一步研究的重点。

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 DC: IEEE Computer Society 2008.
- [2] Nurmi D, Wolski R, Grzegorzczak C. The eucalyptus open source cloud computing system [C]//Cluster Computing and the Grid. California: University of California 2009.
- [3] Chien A, Calder B, Elbert S, et al. Architecture and performance of an enterprise desktop grid system [J]. Journal of Parallel and Distributed Computing 2003 63(5): 597-610.
- [4] Rochwerger B. The reservoir model and architecture for open federated cloud computing [J]. IBM Journal of Research and Development, 2009 53(4): 1-17.

(下转第 3749 页)

图 2 对比了采用新方法和原方法时数据中心的 SLAV。由图可知,在绝大部分情况下,采用新方法进行虚拟机调度时,能将 SLAV 降得更低,即能更好地保证服务质量。但是在基于 2011 年 4 月 20 号的监测数据进行虚拟机调度时,新方法没有原方法好,这可能与过载检测算法、负载预测算法等的因素相关,将在下一步工作中深入分析这一问题。

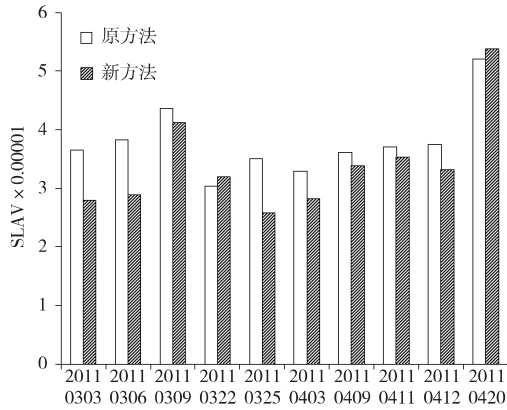


图 2 SLAV 对比

图 3 对比了采用新方法和原方法处理物理机过载时所迁移的虚拟机数量。由图可知,采用新方法处理过载时所迁移的虚拟机更少。与原方法相比,新方法在最好情况下可将虚拟机迁移次数减少 61.5%,在最差情况下也可减少 52.3%。

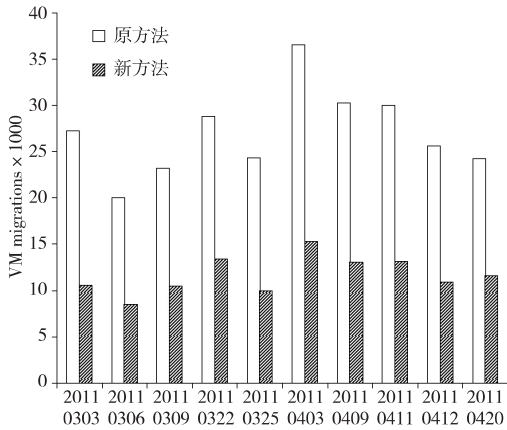


图 3 迁移次数对比

5 结束语

为了减少 MMT 策略在解除服务器过载时虚拟机迁移次数较高的问题,本文结合物理机容量特征和虚拟机负载特征,提出了基于容量感知和负载特征的虚拟机选择算法。该算法根据服务器的容量,采用不同的虚拟机选择策略,以期达到减少虚拟机迁移次数、降低数据中心的能耗并保证服务质量的目标。仿真结果表明,该方法可以达到既定目标。下一步,笔者将开展服务器负载预测及轻载服务器整合方面的研究,寻找进

一步降低数据中心能耗的方法。

参考文献:

- [1] 刘永,王新华,王朕,等. 数据中心节能算法研究综述[J]. 微型机与应用,2012,31(7): 1-5.
- [2] Barroso L A, Holze U. The case for energy-proportional computing[J]. Computer, 2007, 40(12): 33-37.
- [3] Fan Xiao, Weber W D, Barroso L A. Power provisioning for a warehouse-sized computer[C]//Proc of the 34th Annual International Symposium on Computer Architecture. San Diego, USA: ACM Press, 2007.
- [4] Wu Linlin, Garg S K, Buyya R. SLA-based resource allocation for software as a service provider (SaaS) in cloud computing environments[C]//Proc of IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. 2011: 195-204.
- [5] Beloglazov A, Buyya R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers[J]. Concurrency and Computation: Practice and Experience, 2012, 24(13): 1397-1420.
- [6] Kusic D, Kephart J O, Hanson J E, et al. Power and performance management of virtualized computing environments via look ahead control[J]. Cluster Computing, 2009, 12(1): 1-15.
- [7] Vu H T, Hwang S. A traffic and power-aware algorithm for virtual machine placement in cloud data center[J]. International Journal of Grid & Distributed Computing, 2014, 7(1): 21-32.
- [8] Graubner P, Schmidt M, Freisleben B. Energy-efficient management of virtual machines in eucalyptus[C]//Proc of IEEE International Conference on Cloud Computing. [S.l.]: IEEE Press, 2011: 243-250.
- [9] Luo Jianping, Xia Li, Chen Mingrong. Hybrid shuffled frog leaping algorithm for energy-efficient dynamic consolidation of virtual machines in cloud data centers[J]. Expert Systems with Applications, 2014, 41(13): 5804-5816.
- [10] 姜建华,刘渝,王丽敏,等. 数据云中基于启发式反向蜂群的虚拟机选择节能算法[J]. 吉林大学学报:理学版,2014,52(6): 1239-1248.
- [11] 曾凯,余堃,敬思远. 云环境下基于功耗感知的虚拟机博弈迁移算法[J]. 计算机应用研究,2013,30(6): 1668-1671.
- [12] 叶可江,吴朝晖,姜晓红,等. 虚拟化云计算平台的能耗管理[J]. 计算机学报,2012,35(6): 1262-1285.
- [13] Calheiros R N, Rajiv R, Beloglazov A. 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] Park K S, Pai V S. CoMon: a mostly-scalable monitoring system for PlanetLab[J]. ACM SIGOPS Operating Systems Review, 2006, 40(1): 65-74.

(上接第 3737 页)

- [5] 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): 2411-2415,2421.
- [6] 左利云,曹志波. 云计算中调度问题研究综述[J]. 计算机应用研究,2012,29(11): 4023-4027.
- [7] Armbrust M, Fox A, Griffith R. A view of cloud computing[J]. Communications of the ACM, 2010, 53(4): 50-58.
- [8] 李宝禄,张伟. 基于 Hadoop 平台的并行特征匹配算法研究[J]. 计算机应用研究,2014,31(11): 3320-3323.
- [9] 宁彬,谷琼,吴钊,等. 云计算环境下的混沌萤火虫的资源负载均

衡算法[J]. 计算机应用研究,2014,31(11): 3397-3400.

- [10] 李邈,姚晔,李铁. 基于改进型人工萤火虫算法的云资源研究[J]. 计算机应用研究,2013,30(8): 2298-2300,2333.
- [11] Nurmi D, Wolski R, Grzegorzczak C, et al. The eucalyptus open-source cloud-computing system[C]//Proc of IEEE/ACM International Symposium on Cluster Computing and the Grid. 2009.
- [12] OpenNebula[EB/OL]. (2010). <http://www.opennebula.org>.
- [13] Nimbus[EB/OL]. (2010). <http://nimbusproject.org>.
- [14] 张劲松,王启富,万立. 基于本体的产品配置建模研究[J]. 计算机集成制造系统,2003,9(5): 344-350.