

面向服务簇的服务体系结构及服务发现

胡 强*, 杜玉越

(山东科技大学 信息科学与工程学院, 山东 青岛 266590)

(* 通信作者电子邮箱 huqiang200280@163.com)

摘 要:为了缩减服务查找空间,提高服务发现效率,提出服务簇的概念。将具有相似或相同服务功能的 Web 服务封装为一个服务簇,构建基于服务簇的服务请求/响应模式。给出了服务簇的形式化定义、面向服务簇的服务体系结构以及服务簇模式下的最优服务发现算法。仿真实验以 10000 个 Web 服务作为测试源,在服务簇数目不大于 1000 时,服务簇模式下的服务查找时间以及服务再查找时间均小于 600 ms,而当前服务响应模式下,上述时间均大于 900 ms。实验结果表明:基于服务簇的服务请求/响应模式可以显著提高服务发现效率,并能大幅度降低服务再查找的时间。

关键词: Web 服务;服务簇;服务发现;面向服务体系结构;服务质量

中图分类号: TP302 **文献标志码:** A

Service architecture and service discovery oriented to service clusters

HU Qiang*, DU Yuyue

(College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao Shandong 266590, China)

Abstract: In order to reduce the search space and improve the efficiency of service discovery, the concept of service cluster was put forward. Web services with the same or similar functions were encapsulated as a service cluster. The request/response schema based on service clusters was constructed. The formal definition of service cluster, the architecture and an algorithm to discover the optimal Web service in the service clusters schema were presented. The simulation experiment was conducted on 10 000 Web services. The discovery time and rediscovery time under service clusters schema were less than 600 ms when the number of service clusters was no more than 1 000. However, the above time was more than 900 ms in the current service response schema. The efficiency is greatly increased in discovering services under service clusters schema, and the rediscovery time is also greatly decreased.

Key words: Web service; service cluster; service discovery; Service-Oriented Architecture (SOA); Quality of Service (QoS)

0 引言

随着 Web 服务技术的发展与普及,Web 服务数量迅猛增加,目前 seekda.com 网站注册的 Web 服务数量已达到 3 万左右,这其中有大量的 Web 服务具备相同或者相似的功能,如果以关键字 stock 进行检索可以得到 391 个 Web 服务,如果以 weather 作为关键字检索可以得到 95 个 Web 服务。不同服务提供商发布了大量在功能上相同或者相似的 Web 服务,为服务请求者提供了更多的选择,同时也增加了用户查找和绑定最优服务的难度。

现有服务模式是单一服务请求/响应模式,即为用户的服务请求绑定一个 Web 服务,当用户需求发生细微改变或由于网络环境的变化造成当前响应服务失效时,需要重新在数目巨大的 Web 服务中再次查找合适的替换服务,这种再查找代价是巨大的。

为了提高查找速度,需要缩减服务查找空间。本文采取分簇的方法,将一组在功能上相同或者相似的 Web 服务映射为一个服务簇,并将服务簇作为服务响应的基本单位,构建一种基于服务簇的服务请求/响应模式(简称服务簇模式)。由

于服务簇是一组功能相同或者相似的 Web 服务的集合,服务响应粒度得到了加大,因此可以有效缩减服务查找空间,从而降低 Web 服务查找和绑定难度。当绑定的 Web 服务失效时,可以快速从响应服务簇中寻找一个替代服务。

在以往的研究中,出现了服务组^[1]、服务池^[2-3]、服务社区^[4]、服务容器^[5]和服务簇^[6-7]等类似的概念,这些概念均采用了服务分组的思想,但上述概念均要求包含在一组的 Web 服务必须具备相同的参数接口,因此服务功能单一,服务再查找时效率低下,并且上述研究没有给出具体的实施方案。本文中的服务簇概念并不要求簇内的 Web 服务具备相同的接口,因此,服务簇能够更好地满足具备细微差异的服务请求。

1 服务簇

由于一个服务簇是由一组功能相似或者相同的 Web 服务组成,因此服务簇的定义与 Web 服务类似。

定义 1 Web 服务。Web 服务定义为 6 元组 $Ws = (W_{id}, W_{des}, I, O, Q, L)$ 。其中: W_{id} 是 Web 服务的标识; W_{des} 是 Web 服务的功能描述; I 和 O 分别是 Web 服务的输入和输出参数集

收稿日期: 2013-02-21; 修回日期: 2013-03-22。

基金项目: 国家 973 计划项目(2010CB328101); 国家自然科学基金资助项目(61170078); 山东省教育厅计划项目(J12LN11)。

作者简介: 胡强(1980-),男,山东邹城人,讲师,硕士,主要研究方向: 服务计算、形式化建模; 杜玉越(1960-),男,山东聊城人,教授,博士,主要研究方向: 计算机支持的协同工作、形式化技术、Petri 网。

合; Q 是 Web 服务的质量集合; L 是 Web 服务的 URI。

上述定义中 $W_{des} = (Ob, Ac)$, Ob 是 Web 服务的操作对象, Ac 是其操作对象^[8]。例如一个机票查询的 Web 服务可以将其 W_{des} 设置为 (Air_ticket, Query)。服务质量集合 $Q = \{q_i\}$, $q_i = \{N, C, V, U\}$, 其中: N 为服务质量参数名称, C 是比较运算符, V 是参数的值, U 是参数的单位。如果 $q = (ResponseTime, <, 10, ms)$, 表示该 Web 服务的响应时间低于 10 ms。

定义2 服务簇。服务簇定义为 7 元组 $S_{cluster} = (C_{id}, C_{des}, I, O, S_w, Q_c, f)$ 。其中: C_{id} 是服务簇的标识; C_{des} 是服务簇的功能描述; $I = I_c \cup I_p$, $O = O_c \cup O_p$, I_c 和 O_c 分别表示服务簇中 Web 服务的公共输入和输出参数集合, I_p 和 O_p 分别表示服务簇中部分 Web 服务的私有输入和输出参数集合; $S_w = \{ws_1, ws_2, \dots, ws_n\}$ 是服务簇中 Web 服务的集合; $f: O_p \rightarrow 2^{I_p}$, 是 O_p 与 I_p 中参数的映射函数, 且 $\forall o \in O_p, f(o) = \{ws. I \cap I_p \mid \forall ws \in S_w \wedge o \in ws. O\}$; $Q_c = \{q_i\}$, $q_i = \{N, C, [V_{min}, V_{max}], U\}$ 为服务簇质量参数集合。

由于服务簇的质量必须反映其所包含的所有 Web 服务的质量参数值, 因此采用区间法表示服务簇的质量参数值。如果一个服务簇的 $Q_c = \{(ResponseTime, <, [1, 3], ms), (ResponseRat, >, [97, 99], \%)\}$, 则表示该服务簇中的 Web 服务响应时间最低小于 1 ms, 最大小于 3 ms, 而服务响应率最低的 Web 服务小于 97%, 最高的 Web 服务大于 99%。

2 面向服务簇的服务响应模式及其体系结构

为了改进当前单一服务请求/响应模式的不足, 本文提出了面向服务簇的服务请求/响应模式, 如图 1 所示。在传统的业务模型层和物理资源层之间增加虚拟资源层, 用以存储服务簇。

物理资源层中存放具体的 Web 服务, 这些 Web 服务按照一定的算法聚类为服务簇, 用户的服务请求由业务模型层进行描述, 根据业务模型层中的服务请求为其在虚拟资源层查找满足其功能的服务簇, 最后再从服务簇中选择一个质量最优的 Web 服务返回给服务请求。需要说明的是, 用户的服务请求有时仅仅由一个 Web 服务无法完成, 此时就需要在业务模型层分解为服务组合, 然后在虚拟资源层选择一组服务簇进行响应, 最终在物理资源层会出现一个组合 Web 服务流程为其提供响应服务, 具体实现另文表述。

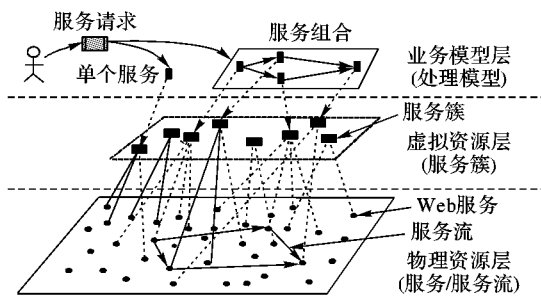


图1 面向服务簇的服务请求/响应模式

在进行服务查找时, 在一个服务簇中通常会出现一组可以满足服务请求的 Web 服务, 此时构建一个服务容器, 对这些 Web 服务进行暂存, 一旦需要寻找替代 Web 服务, 按照相关算法可以快速从服务容器中进行选取, 从而大幅度降低服务再查找的时间。

当前的 Web 服务体系结构包含三种角色, 服务请求者、服务提供者以及注册中心 (UDDI 或者一些 Web 服务注册网站)。相比上述体系结构, 面向服务簇的体系结构增加了一个服务代理器, 如图 2 所示。

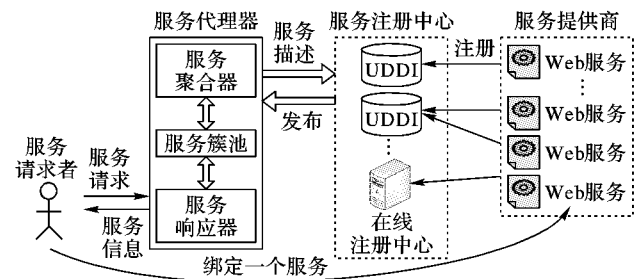


图2 面向服务簇的服务体系结构

图 2 中的服务请求者、服务提供者以及注册中心与当前 SOA 含义相同, 新增加的服务代理器由服务响应器、服务簇池以及服务聚合器组成, 各部分功能如下:

- 1) 服务响应器用来接收和处理服务请求; 从服务簇池中选择一个 (组) 合适的服务簇, 并为服务请求绑定一个最优 Web 服务 (流程)。
- 2) 服务簇池用来存放服务簇。为了便于服务簇的查找, 服务簇在服务簇池中按照逻辑上按照领域本体树进行存储。
- 3) 服务聚合器用来订阅 Web 服务和生成服务簇。类似于 RSS^[9], 服务提供者按照指定格式将 Web 服务发布, 服务聚合器从众多的服务注册中心自动订阅和更新自己领域相关的 Web 服务, 并按照相关算法生成服务簇以及更新服务簇池。

3 服务发现

在面向服务簇的服务请求/响应模式下绑定一个服务分为两步, 第一步寻找一个恰当的响应服务簇, 第二步在响应服务簇中绑定一个最优服务。由于不同的服务提供者定义 Web 服务时采用的语法不同, 因此在进行 Web 服务匹配时必须消除语法差异。本文采用基于本体的语义相似度来衡量 Web 服务之间的相似程度, 并假设所有的 Web 服务和响应服务簇都是建立在领域本体概念之上, 可以方便地进行语义相似度的计算。

语义相似度与语义距离密切相关, 语义距离的计算方法较多, 可以参见文献 [10-14], 不同的方法具备不同的效率和精确度, 本文采用文献 [13] 提出的方法计算语义距离。

定义3 语义相似度。CCP 是领域本体树中本体概念 O_1 和 O_2 最近的父节点, $Level(CCP)$, $level(O_1)$ 和 $level(O_2)$ 表示 CCP、 O_1 和 O_2 在本体树中的深度, 且根节点 O_{top} 的 $level(O_{top}) = 1$, 则 O_1 和 O_2 的语义距离为

$$dist(O_1, O_2) = \begin{cases} (2^{-level(CCP)} - 2^{-level(O_1)}) + (2^{-level(CCP)} - 2^{-level(O_2)}), & O_1 \neq O_2 \\ 0, & O_1 = O_2 \end{cases}$$

语义相似度定义为 $SeSim(O_1, O_2) = 1 - dist(O_1, O_2)$ 。

定义4 语义等价。对给定阈值 δ , 本体概念 O_1 和 O_2 称为语义等价, 当且仅当 $SeSim(O_1, O_2) \geq \delta$, 如果 O_1 和 O_2 语义等价, 记为 $O_1 \leftrightarrow O_2$ 。通常情况下, δ 可以取值为 0.9。

定义5 子集参数。 Pa 和 Pb 是两组参数, Pa 称为 Pb 的子集参数当且仅当:

1) $Num(Pa) \leq Num(Pb)$, $Num(p)$ 表示参数 p 的个数;

2) $\forall m_i \in Pa, \exists n_j \in Pb$, 满足 $m_i \leftrightarrow n_j$ 且 $Type(m_i) \subseteq Type(n_j)$ 。

$Type(m)$ 表示参数 m 的类型, 符号 \subseteq 表示参数类型兼容, 如果 Pa 是 Pb 的子集参数, 记为 $Pa \subseteq Pb$;

定义6 值覆盖参数。 Pa 和 Pb 是两组参数, Pa 称为 Pb 的值覆盖参数当且仅当:

1) $Pb \subseteq Pa$;

2) $\forall m_i \in Pb, \exists n_j \in Pa$, 满足 $Value(m_i) \subseteq Value(n_j)$, $Value(m)$ 表示 m 的取值范围。

如果 Pa 是 Pb 的值覆盖参数, 记为 $Pa \supseteq Pb$ 。

例如: r 为一个服务请求, $r.Q = \{(Rt, <, 2, ms), (Co, =, 0, \$), (Rr, >, 95, \%)\}$, s 是一个服务簇, $s.Qc = \{(Rt, <, [1, 7], ms), (Co, =, [0, 0.02], \$), (Rr, >, [92, 99], \%)\}$, 则 $s.Qc \supseteq r.Q$ 。

服务质量参数可以分为正向参数和负向参数, 正向参数值越大表示服务质量越高, 如吞吐量和可靠性; 负向参数值越小表示服务质量越高, 如响应时间、延迟时间以及费用等。本文基于 SAW 方法^[15] 对服务质量进行评分。

定义7 服务质量评分。 $S = \{ws_1, ws_2, \dots, ws_m\}$ 为一组 Web 服务, 假设 S 中的每个服务具有 n 个质量参数, $[q_{i1}, q_{i2}, \dots, q_{in}]$ 是服务 s_i 的质量参数向量, s_i 的服务质量评分为:

$$score(s_i) = \sum_{j=1}^n q_{ij} * w_j; \sum_{j=1}^n w_j = 1 \quad (1)$$

$$q_{ij} = \begin{cases} \frac{q_{ij}^{max} - q_{ij}^{min}}{q_j^{max} - q_j^{min}}, & q_j^{max} - q_j^{min} \neq 0 \\ 1, & q_j^{max} - q_j^{min} = 0 \end{cases} \quad (2)$$

$$q_{ij} = \begin{cases} \frac{q_{ij} - q_j^{min}}{q_j^{max} - q_j^{min}}, & q_j^{max} - q_j^{min} \neq 0 \\ 1, & q_j^{max} - q_j^{min} = 0 \end{cases} \quad (3)$$

式(1)用于计算 s_i 的服务质量评分, 其中的参数 w_j 用于设置各个质量参数的权重; 式(2)用于负向质量参数的量化; 式(3)用于正向质量参数的量化。

本文将服务请求形式化定义为一个4元组 $S_{request} = \{R_{des}, I, O, Q\}$, 其中: R_{des} 表示请求的服务功能描述, 其他各项具体含义和定义1相同。算法1描述了在服务簇响应模式下绑定一个服务质量最优服务的过程, 基于上述定义, 通过计算功能相似度, 可以获取预期响应服务簇, 如果服务簇和服务请求的功能相似度大于给定阈值 δ , 接口匹配(第3)~7)行)将会执行, 这样可以获取服务簇中在接口上匹配服务请求的所有 Web 服务集合, 然后对该集合中的 Web 服务进行服务质量评分(第8)~10)行), 找出得分最高的 Web 服务, 并将其返回即可。如果当前绑定的 Web 服务失效, 可以从集合 S_w' 快速为其选择一个替代服务。

算法1 服务簇模式下的 Web 服务发现。

输入: 服务请求 Sr , 服务簇集合 SC ;

输出: 质量最优的匹配 Web 服务。

- 1) for $\forall S \in SC$
- 2) if $SeSim(S.C_{des}, Sr.R_{des}) > \delta$
- 3) if $(S.Ic \subseteq Sr.I(Sr.O \subseteq S.O))$ and $(S.Qc \supseteq Sr.Q)$
- 4) if $(Sr.O \subseteq S.Oc)$ $S' = S$;
- 5) else $\{O' = Sr.O - S.Oc$;

6) for $\forall o_i, o_j \in O'$

7) if $\exists I' \in 2^{I_p}$, s. t. $f(o_i) = f(o_j) = I'$
then $S' = S$; }

8) for $\forall ws \in S'$. S_w

9) if $(ws.I \subseteq Sr.I(Sr.O \subseteq ws.O))$ $S_w' = S_w' \cup \{ws\}$

10) for $\forall ws \in S_w'$, $s = \{ws \mid \max(Score(ws))\}$

11) Return (s);

4 仿真实验

由于当前 Internet 中很少有基于领域本体构建的 Web 服务集, 因此本文采取仿真的形式进行实验验证, 首先建立一个深度为8宽度为6的本体树, 树的叶子节点表示本体概念, 实验中的 Web 服务涉及到的所有语法描述均基于该本体树中的概念, Web 服务和服簇采取如下形式构建:

1) 设置服务簇和 Web 服务的数量。令 N 是服务簇的数量, $N = \{50, 100, 200, 500, 1000, 2000\}$, Web 服务的数量设置为 10000, 对于给定数量的服务簇, 随机设置每一个服务簇的组成 Web 服务的数目, 将其表示为 M , 令 $1 \leq M \leq 10 * (10000/N)$ 。

2) 生成服务簇。对每一个服务簇, 其功能描述以及输入输出参数是随机选取本体树中的概念, 任意两个服务簇的公共参数要求不能相同, 服务簇的质量参数区间随机数生成。

3) 生成服务簇中的 Web 服务。对每一个服务中组成 Web 服务数量要求为 $1 \leq M \leq 10 * (10000/N)$, 通过随机分配私有输入输出参数以及设置服务质量参数形成对应的 Web 服务。

4) 扫描服务簇中的 Web 服务私有输入输出参数, 构建映射关系函数 f 。

实验采取如下硬件环境: CPU 为 i3 2120, 3.3 GHz, 内存 2 GB, 操作系统 Windows 7, 仿真程序采用 Java 进行开发。共进行 6 轮次, 从第 1 轮到第 6 轮次实验中服务簇的数目分别为 50, 100, 200, 500, 1000, 2000, 每一轮次共进行实验 10 次, 取其平均值作为实验数据, 实验结果分别如图 3 和图 4 所示。从仿真结果可以看出, 相比当前服务请求/响应模式, 面向服务簇的服务请求/响应模式具备较高的服务发现效率, 但随着服务簇粒度由大变小, 服务发现时间和再查找时间先降低后急剧增大, 因此服务簇的粒度与服务发现效率密切相关。

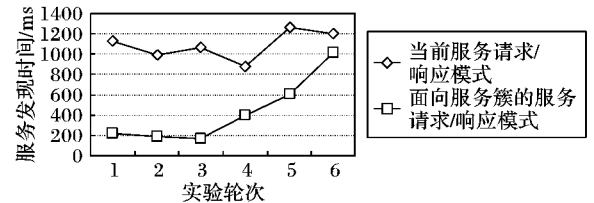


图3 两种服务模式下的服务发现效率比较

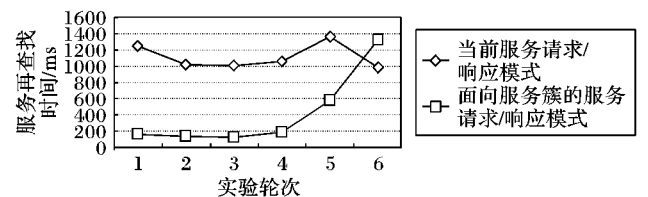


图4 两种服务模式下的服务再查找效率比较

5 结语

当前单一服务请求/响应模式下, 在众多的功能相似的 Web 服务中查找一个适合服务请求的质量最优 Web 服务是

一件困难的事情,特别是一旦当前绑定的 Web 服务失效,需要重新进行服务的查找,因此效率较低。为了缩减服务查找空间,提高查找效率,本文提出服务簇的概念以及面向服务簇的服务请求/响应模式。文中给出了服务簇的形式化定义、面向服务簇的服务请求/响应模式以及服务体系结构,设计了服务查找算法,并进行了仿真实验,结果表明面向服务簇的服务请求/响应模式在服务发现方面具有较高效率,能够大幅度地降低服务再查找时间。下一步的工作是对服务簇的粒度划分进行研究,发现粒度划分方案以及粒度与服务发现效率之间的关系。

参考文献:

- [1] MAGUIRE T, SNEELLING D. Web services service group 1.2[M]. London: OASIS, 2006: 1-42.
- [2] LIU X Z, HUANG G, MEI H. Discovering homogeneous Web service community in the user-centric Web environment [J]. IEEE Transactions on Service Computing, 2009, 2(2): 167-181.
- [3] 刘讚哲, 黄昱, 梅宏. 用户驱动的服务聚合方法及其支撑框架[J]. 软件学报, 2007, 18(8): 1883-1895.
- [4] SHENG Q Z, BENATALLAH B, MAAMAR Z, et al. Configurable composition and adaptive provisioning of Web services [J]. IEEE Transactions on Services Computing, 2009, 2(1): 34-49.
- [5] BENATALLAH B, SHENG Q Z, DUMAS M. The Self-Serv environment for Web services composition [J]. IEEE Internet Computing, 2003, 7(1): 40-48.
- [6] LIU J, WANG H Y, CUI L Z, et al. Method and supporting framework for business domain-oriented Web service discovery [J]. Journal of Southeast University: English Edition, 2008, 24(3): 369-371.
- [7] 陈科, 成毅, 谢明霞, 等. 基于服务簇的空间信息服务自动发现[J]. 计算机工程, 2012, 38(24): 182-190.
- [8] 栾文静, 杜玉越. 一种基于服务簇网元模型的 Web 服务发现方法[J]. 计算机科学, 2012, 39(8): 147-152.
- [9] 李庆斌, 左珊珊, 董振华, 等. 中文 RSS 信息自动检索与分类研究[J]. 计算机工程, 2011, 37(6): 79-81.
- [10] ALEXANDER B, HIRST G. Semantic distance in WordNet: an experimental, application-oriented evaluation of five measures [C]// Workshop on WordNet and Other Lexical Resources: Second Meeting of the North American Chapter of the Association for Computational Linguistics. Pittsburgh: [s. n.], 2001: 29-34.
- [11] LI Y, BANDAR Z A, MCLEAN D. An approach for measuring semantic similarity between words using multiple information sources [J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(4): 871-882.
- [12] GANJISAFFAR Y, ABOLHASSANI H, NESHATI M, et al. A similarity measure for OWL-S annotated Web services [C]// Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. Washington, DC: IEEE Computer Society, 2006: 621-624.
- [13] ZHONG J W, ZHU H P, LI J M, et al. Conceptual graph matching for semantic search [J]. Conceptual Structures: Integration and Interfaces, 2002 23(9): 92-106.
- [14] 孙萍, 蒋昌俊. 利用服务聚类优化面向过程模型的语义 Web 服务发现[J]. 计算机学报, 2008, 31(8): 1340-1353.
- [15] YALCIN N, BAYRAKDAROGLU A, KAHRAMAN C. Application of fuzzy multi-criteria decision making methods for financial performance evaluation of Turkish manufacturing industries [J]. Expert Systems with Applications, 2012, 39(1): 350-364.

(上接第2142页)

4 结语

本文研究了异构云中面向集群负载的任务调度问题,通过融入“二八”定律,对任务调度过程加入了大客户满意度概念,提出了一种更利于提升云服务提供商利润的新模型。模型充分考虑了异构云环境下各节点固有能力之间的差异,把用户对集群节点可靠性和安全性的偏好程度加入到策略中来,以求达到在保证集群负载均衡的情况下,最小化任务执行时间的同时,尽可能提高大客户满意度的目标。并提出了基于改进自适应变异粒子群算法的实现,新算法对原 AMPSO 算法做了改进,通过将全局最优位置和适应值较低的粒子进行自适应变异,使得算法具有更强的跳出局部最优的能力。通过仿真实验分析表明,改进的自适应变异粒子群算法在解决本文提出的问题时,比基本 PSO 算法和未改进的 AMPSO 算法具有更高的效率。但本文的重心放在了新模型和策略的阐述上,并没有对可以实现此策略的其他算法进行深入的研究和比较,故无法得出一个最适合本模型和策略的实现算法,这将是以后重点研究的方向。

参考文献:

- [1] 刘晔, 沈潇军, 刘摩西, 等. 基于云模式的资源调度与负载均衡研究[J]. 电脑知识与技术, 2011, 7(33): 8208-8210.
- [2] 杨锦, 李肯立, 吴帆, 等. 异构分布式系统的负载均衡调度算法[J]. 计算机工程, 2012, 38(2): 166-168.
- [3] 凌云, 周华锋. 面向异构集群系统的动态负载均衡技术研究[J]. 计算机工程与设计, 2008, 29(12): 3068-3070.
- [4] 张春艳, 刘清林, 孟珂, 等. 基于蚁群优化算法的云计算任务分配[J]. 计算机应用, 2012, 32(5): 1418-1420.
- [5] 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.
- [6] MARTINO D, MILLOTTI M. Sub optimal scheduling in a grid using genetic algorithms [J]. Parallel Computing, 2004, 30(5): 553-565.
- [7] ANDREW P, THOMAS N. Framework for task scheduling in heterogeneous distributed computing using genetic algorithms [J]. Artificial Intelligence Review, 2005, 24(3): 415-429.
- [8] 张铃, 张钊. 遗传算法机理的研究[J]. 软件学报, 2000, 11(7): 945-952.
- [9] 吕振肃, 侯志荣. 自适应变异的粒子群优化算法[J]. 电子学报, 2004, 32(3): 416-420.
- [10] 秦勇, 宋继光, 蔡昭权, 等. 基于关系矩阵编码的粒子群负载均衡算法研究[J]. 计算机应用与软件, 2011, 28(5): 126-128.
- [11] QIN X, XIE T. An availability-aware task scheduling strategy for heterogeneous systems [J]. IEEE Transactions on Computers, 2008, 57(2): 188-199.
- [12] FANG Y Q, WANG F, GE J W. A task scheduling algorithm based on load balancing in cloud computing [C]// Web Information Systems and Mining, LNCS 6318. Berlin: Springer, 2010: 271-277.
- [13] TIAN W H, ZHAO Y, ZHONG Y L, et al. A dynamic and integrated load-balancing scheduling algorithm for Cloud datacenters [C]// CCIS 2011: Proceedings of the 2011 IEEE International Conference on Cloud Computing and Intelligence Systems. Piscataway: IEEE, 2011: 311-315.
- [14] HU J H, GU J H, SUN G F, et al. A scheduling strategy on load balancing of virtual machine resources in cloud computing environment [C]// PAAP 2010: Proceedings of the 2010 Third International Symposium on Parallel Architectures, Algorithms and Programming. Piscataway: IEEE, 2010: 89-96.
- [15] 杨维, 李歧强. 粒子群优化算法综述[J]. 中国工程科学, 2004, 6(5): 87-94.