

一种基于免疫克隆的偏好多维 QoS 云资源调度优化算法

孙大为¹, 常桂然², 李风云^{1,2}, 王 川¹, 王兴伟¹

(1. 东北大学信息科学与工程学院, 辽宁沈阳 110819; 2. 东北大学计算中心, 辽宁沈阳 110819)

摘 要: 针对云计算环境中高效资源调度问题, 首先从理论上对云资源调度进行了建模, 对用户应用偏好和多维 QoS 中的用户效用进行了量化, 给出了多维 QoS 优化的目标函数. 结合具有快速多目标优化能力的免疫克隆算法, 提出了一种云资源调度优化算法. 根据应用偏好信息为抗体分配偏好优先级, 据此进行抗体的免疫克隆操作, 提高抗体免疫基因操作的预见性, 改善了向最优解的高效收敛能力. 实验结果分析表明, 该算法能改善云资源调度系统的可用性、负载均衡偏差、有效时间等方面的性能, 满足了云计算环境的实际需求.

关键词: 资源调度; 免疫克隆; 多维 QoS; 应用偏好; 云计算

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2011) 08-1824-08

Optimizing Multi-Dimensional QoS Cloud Resource Scheduling by Immune Clonal with Preference

SUN Da-wei¹, CHANG Gui-ran², LI Feng-yun^{1,2}, WANG Chuan¹, WANG Xing-wei¹

(1. School of Information Science and Engineering, Northeastern University, Shenyang, Liaoning 110819, China;

2. Computing Center, Northeastern University, Shenyang, Liaoning 110819, China)

Abstract: Focusing on the problem of high efficiency and effectiveness resource scheduling in cloud computing environments, the model of cloud resource scheduling is systematically analyzed in theory at first, the application preferences and the user utility of multi-dimensional QoS is quantified, and the objective function of multi-dimensional QoS is given at last. Combining with the immune clonal algorithm of rapid multi-objective optimization, a heuristic cloud resource scheduling algorithm with application preference is proposed. The non-dominated antibodies are proportionally immune cloned according to their preference priority, which are defined by their cloud application preferences. It is beneficial to enhance the forecasting accuracy of the immune gene manipulation, and to increase the speed of finding the optimal solution based on the application preference. Experimental results conclusively demonstrate the efficiency and effectiveness of the improve system availability, load balancing deviation and valid time brought by the proposed algorithm in cloud computing environments.

Key words: resource scheduling; immune clonal; multi-dimensional QoS; application preference; cloud computing

1 引言

云计算 (cloud computing)^[1~4] 作为一种新的以商业为主要推动力的大规模分布式计算范式, 其利用抽象、虚拟化、瞬时部署、宽带网络等关键技术, 通过 Internet 实现资源的互联、互通和互操作, 以统一服务 (anything as a service) 的形式, 利用多终端 (multi-terminal)、多平台 (multi-platform)、多网络 (multi-network) 为用户以标准的浏览器访问方式在任意时间、任何地点内提供包括计算、存储、平台等在内的按需分配 (pay-as-you-go)、动态

配置、弹性扩展、价格低廉、高可用性和高可靠性的非平凡 QoS (quality of service) 服务. 从理论的角度而言, 云计算是通过网络数据中心和动态资源池来提供低成本、高效率、高可用性的各种服务, 高效的实现了效用计算的目标^[2,3]. 在实践中, 用户可以通过 Internet 以标准的浏览器方式在任意时间、任何地点访问所需要的服务, 而存储空间、计算能力、应用平台等资源不再是有限的. 为急剧增长的硬件成本、计算存储能力和服务计算以及 Web3.0 的发展所带来的各种新的需求的满足提供了高效的解决方案, 并受到政府、企业和科研机构的广泛

关注.云系统中的存储、集群、数据、软件、网络等构成云资源,是其基础支撑环境,针对云资源调度的特征,研究高效、经济、快速的资源发现及调度机制是本文研究的问题.

资源调度是云系统中的一个核心且富有挑战的重要组成部分,云计算与传统分布式计算范式之间存在以下几点主要区别^[2,3,5,6]:(1)灵活的网络访问机制.用户可以灵活的利用多种异构终端,通过 Internet 可以获取瞬时高效的服务;(2)服务的可量化.云系统为用户提供可量化的服务,并进行服务状态的实时监控与优化;(3)服务的快速供给.云系统通过虚拟化等相关关键技术为用户提供瞬时部署、动态配置的各种服务;(4)服务的按需分配.云系统根据用户的付费进行资源的配置和重配置,费用因子则是资源调度的一个重要影响因素;(5)动态的资源池.云系统根据不同层次的虚拟化技术,屏蔽云系统中的资源的分布、异构特征,为用户提供透明、统一的服务接口,快速的完成用户的服务请求.根据系统的特殊要求进行全面、深入、系统的分析和研究.一方面,需要根据应用的特征来高效的减少资源搜索空间,提高资源的发现效率和准确性,有效保证用户服务的性能 QoS.另一方面,针对用户的状态变迁,动态的进行资源选择和重定向,根据用户的应用偏好,高效的满足用户的经济 QoS,有效的提高用户满意度.

本文从理论上对云资源调度模型进行了系统的建模分析,提出了一种基于免疫克隆的偏好多维 QoS 的云资源调度 (cloud resource scheduling by immune clonal with preference, 简称 CICP) 优化算法,在云资源调度系统的可用性、负载均衡离差和有效时间等方面均实现了较好性能.本文的主要贡献:(1)从理论上提出了一种云资源调度 CRS 模型 (cloud resource scheduling, 简称 CRS 模型),并对应用偏好和多维 QoS 空间中的效用进行了量化分析;(2)针对量化的应用偏好,结合具有快速多目标优化能力的免疫克隆算法,实现了对 t -QoS 空间 (t dimensions QoS, 简称 t -QoS) 的多维 QoS 的改善;(3)完成了包括最终完成时间、响应时间、费用、可用性、安全性在内的 5-QoS 进行量化,实现了可用性、负载均衡离差和有效时间三个性能指标的改善.

2 问题建模

2.1 云资源调度模型

由 n 个资源和 m 个用户所组成的 $N * M$ 云资源调度模型 (cloud resource scheduling, 简称 CRS 模型) 中,云资源调度问题可以描述为四元组,即 $M = (R, U, Of, \Theta)$. 其中, R 为 n 个资源所组成的资源集合; U 为 m 个用户所组成用户集合; Of 表示云调度系统的调度优化

目标函数;而 Θ 则表示调度优化算法.其具体特征描述如下:

云资源集合 $R = \{r_1, r_2, \dots, r_n\}$, 由 n 个资源构成,且第 i 个资源 $r_i = \{r_i(1), r_i(2), \dots, r_i(t)\}$, $0 \leq i \leq n$, 包含 t 个不同维度特征属性, $r_i(k)$, $0 \leq k \leq t$ 为第 i 个资源 r_i 的第 k 维特征属性. 在一个服务周期 T 内,资源 r_i 的可用服务时间服从 Poisson 分布^[7]且各个资源间相互独立,即 $\forall r_i \in R, \bigcap_{i=0}^n r_i = \emptyset$, 云资源集合 R 的可用服务率向量 $sr = \{sr_1, sr_2, \dots, sr_n\}$, 其中 sr_i ($0 \leq i \leq n$) 为资源 r_i 的可用服务率.

云用户集合 $U = \{u_1, u_2, \dots, u_m\}$, 由 m 个用户构成,且在任意时刻用户 u_j 仅有一个任务 ts_j , 且用户 u_j 的第 k 个任务 $ts_{jk} = \{ts_{jk}(1), ts_{jk}(2), \dots, ts_{jk}(h)\}$, $0 \leq j \leq m$, 包含 h 个不同维度的特征属性,云用户集合 U 的任务产生率向量 $tr = \{tr_1, tr_2, \dots, tr_m\}$, 其中 tr_j , $0 \leq j \leq m$ 为第 j 个用户 u_j 的任务产生率,且各个任务之间相互独立,即 $\forall ts_j \in U, \bigcap_{j=0}^m ts_j = \emptyset$.

$\forall r_i \in R, ts_j \in U$, 如果资源 r_i 可以满足任务 ts_j 的调度要求,即 $\Theta(r_i, ts_j) > 0$, 任务 ts_j 可以分配到资源 r_i 上来执行,通常满足任务 ts_j 的资源 r_i 往往有多个,为了实现系统的有效负载均衡,任务 ts_j 以概率 p_{ij} 被调度到资源 r_i 上,那么云系统概率矩阵 $P_{n \times m} = (p_{ij})_{n \times m}$, 且 $p_{ij} \geq 0$, $\sum_{i=0}^n p_{ij} = 1$. 云系统的资源调度矩阵 $S_{n \times m} = (x_{ij})_{n \times m}$, 如果 $\Theta(r_i, ts_j) > 0$, 则 $x_{ij} = 1$. 否则, $x_{ij} = 0$. 同时,为了保证任务 ts_j 顺利被完成,需要满足约束条件式(1);为了保证资源 r_i 所接受的任务在其可服务的能力范围内,需要满足约束条件式(2);为了保证云系统在一个稳定的环境中,需要满足约束条件式(3).

$$\sum_{i=0}^n p_{ij} \cdot x_{ij} = 1 \quad (1)$$

$$\sum_{j=0}^m p_{ij} \cdot x_{ij} \cdot tr_j \leq sr_i \quad (2)$$

$$\sum_{i=0}^n \sum_{j=0}^m p_{ij} \cdot x_{ij} \cdot tr_j \leq \sum_{i=0}^n sr_i \quad (3)$$

2.2 云应用偏好度量

本文是基于先验式偏好^[8,9]开展研究,先验式偏好分为刚性偏好 (rigidity preference) 和弹性偏好 (elasticity preference) 两种:刚性偏好是指该应用偏好必须在规定区间内完成,否则该任务将以失败告终.弹性偏好指该类偏好可以在用户 u_j 规定的区间内完成,也可在其区间外完成(此时用户效用降低).

当资源 r_i 的第 k 维特征属性 $r_i(k)$, $0 \leq k \leq t$ 满足刚性偏好,用户 u_j 对 $r_i(k)$ 的偏好区间为 $[r_i(k)_{low}, r_i(k)_{up}]$, 如果第 k 维特征属性的偏好实际值 $r_i(k)_{real}$ 越

小,用户 u_j 的满意度越高,则第 k 维特征属性的偏好值 $cp(k)$ 满足式(4);否则,满足式(5).

$$cp_{ij}(k) = \begin{cases} \frac{r_i(k)_{up} - r_i(k)_{real}}{r_i(k)_{up} - r_i(k)_{low}}, & r_i(k)_{low} \leq r_i(k)_{real} \leq r_i(k)_{up} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$cp_{ij}(k) = \begin{cases} \frac{r_i(k)_{real} - r_i(k)_{low}}{r_i(k)_{up} - r_i(k)_{low}}, & r_i(k)_{low} \leq r_i(k)_{real} \leq r_i(k)_{up} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

当资源 r_i 的第 k 维特征属性 $r_i(k)$, $0 \leq k \leq t$ 满足弹性偏好,如果第 k 维特征属性的偏好实际值 $r_i(k)_{real}$ 越小,用户 u_j 的满意度越高,则第 k 维特征属性的偏好值 $cp(k)$ 满足式(6);否则,满足式(7).

$$cp_{ij}(k) = \begin{cases} 0.5 + \min\left(0.5 \frac{r_i(k)_{low} - r_i(k)_{real}}{r_i(k)_{up} - r_i(k)_{low}}, 0.5\right), & r_i(k)_{real} \leq r_i(k)_{low} \\ \max(r_i(k)_{low}/r_i(k)_{real} - 0.5, 0), & \text{otherwise} \end{cases} \quad (6)$$

$$cp_{ij}(k) = \begin{cases} 0.5 + \min\left(0.5 \frac{r_i(k)_{real} - r_i(k)_{up}}{r_i(k)_{up} - r_i(k)_{low}}, 0.5\right), & r_i(k)_{up} \leq r_i(k)_{real} \\ \max(r_i(k)_{real}/r_i(k)_{up} - 0.5, 0), & \text{otherwise} \end{cases} \quad (7)$$

那么,用户 u_j 关于资源 r_i 的第 k 维特征属性的偏好权重 $w(k)$ 满足式(8).用户 u_j 关于资源 r_i 的偏好向量为 $w_{ij} = (w_{ij}(0), w_{ij}(1), \dots, w_{ij}(t))$, 且 $\sum_{i=0}^t w_{ij}(k) = 1$.

$$w_{ij}(k) = cp_{ij}(k) / \sum_{i=0}^t cp_{ij}(i), \quad 0 \leq k \leq t \quad (8)$$

2.3 多维 QoS 目标优化函数

用户 u_j 关于资源 r_i 的 t 个不同维度的特征属性的 QoS 向量 $q_{ij} = (q_{ij}(1), q_{ij}(2), \dots, q_{ij}(t))$ 构成 t -QoS 空间,其中 $q_{ij}(k)$, $1 \leq k \leq t$ 为用户 u_j 关于资源 r_i 第 k 个特征属性的 QoS. 用户 u_j 关于资源 r_i 第 k 个特征属性的用户效用为 $uti_{ij}(q_{ij}(k))$, $1 \leq k \leq t$. 因此,在 t -QoS 空间中,云资源调度多维 QoS 目标优化效用函数如式(9).

$$\begin{cases} Of(uti_{sys}) = \max \sum_{k=0}^t \sum_{j=0}^m \sum_{i=0}^n w_{ij}(k) \cdot uti_{ij}(q_{ij}(k)) \\ \text{s.t.} \quad 0 \leq w_{ij}(k) \leq 1, \sum_{i=0}^t w_{ij}(k) = 1 \end{cases} \quad (9)$$

其中,资源 r_i 第 k 个特征属性的效用函数 $uti_{ij}(q_{ij}(k))$, $1 \leq k \leq t$ 的确定是实现云系统效用最大化的关键,为了简化问题,本文仅针对云服务 5 个关键特征属性:最终完成时间(final completion time)、请求响应时间(request-to-response time)、费用(cost)、可用性(availability)、安全

性(security),进行 5-QoS 空间的效用优化.

任务 ts_j 的最终完成时间反映该资源 r_i 是否有能力完成相应的任务 ts_j ,其效用曲线如图 1(a)所示,相应效用函数满足式(10). d 为用户 u_j 所硬性规定的任务 ts_j 最晚完成时间.在区间 $[0, d]$ 内,效用为常量 1;否则,效用为 0.

$$uti_{ij}(t) = 1 \cdot [uti_{ij}(t) - uti_{ij}(t - d)] = \begin{cases} 1, & t \leq d \\ 0, & t > d \end{cases} \quad (10)$$

响应时间是提高用户 u_j 体验的重要指标,较快的响应时间往往给用户较高的 QoS,其效用曲线如图 1(b)所示,相应效用函数满足式(11). rt_1 和 rt_2 分别为两个不同层次的响应时间界限,在区间 $[0, rt_1]$ 内,效用为常量 1;在区间 $[rt_1, rt_2]$ 内,随着响应时间的不断增加,效用线性递减;否则,效用为 0.

$$\begin{aligned} uti_{ij}(rt) &= 1 \cdot [uti_{ij}(rt) - uti_{ij}(rt - rt_1)] + 1 \cdot \left(1 - \frac{rt - rt_1}{rt_2 - rt_1}\right) \\ &\quad \cdot [uti_{ij}(rt - rt_1) - uti_{ij}(rt - rt_2)] \\ &= \begin{cases} 1, & rt \leq rt_1, \\ 1 \cdot (1 - (rt - rt_1)/(rt_2 - rt_1)), & rt_1 < rt \leq rt_2, \\ 0, & rt > rt_2, \end{cases} \end{aligned} \quad (11)$$

云计算服务是一种‘pay-as-you-go’的服务模式,高的性价比对提高用户 u_j 的 QoS 很关键,其费用效用曲线如图 1(c)所示,相应效用函数满足式(12). b 为用户 u_j 所关于任务 ts_j 的费用预算,在区间 $[0, b]$ 内,随着费用的增加,效用线性递减;否则,效用为 0.

$$\begin{aligned} uti_{ij}(c) &= 1 \cdot \left(1 - \frac{c}{b}\right) \cdot [uti_{ij}(c) - uti_{ij}(c - b)] \\ &= \begin{cases} 1 \cdot (1 - c/b), & c \leq b, \\ 0, & c > b. \end{cases} \end{aligned} \quad (12)$$

可用性反映了用户 u_j 提交服务并成功返回处理结果的概率,较高的可用性,用户 u_j 的 QoS 可以得到较大的提高,其效用曲线如图 1(d)所示,相应效用函数满足式(13).在可用性 $a \geq 0$ 的前提下,随着 a 的不断加,效用呈现指数增长.

$$\begin{aligned} uti_{ij}(a) &= 1 \cdot (1 - 1/e^{-a}) \cdot uti_{ij}(a) \\ &= 1 \cdot (1 - 1/e^{-a}), \quad a \geq 0 \end{aligned} \quad (13)$$

安全性是保证用户安全体验的前提,较高的安全性,对应着较好的用户 u_j 的 QoS.其效用曲线如图 1(e)所示,相应效用函数满足式(14). s_1 和 s_2 分别为两个不

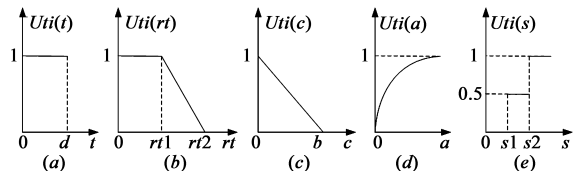


图1 5-QoS的效用函数

同层次的安全界限,在区间 $[0, s_1]$ 内,效用为常量 0;在区间 $[s_1, s_2]$ 内,效用为常量 0.5;在区间 $[s_2, +\infty)$ 内,效用为常量 1.

$$\begin{aligned} uti_{ij}(s) &= 0.5 \cdot [uti_{ij}(s - s_1) - uti_{ij}(s - s_2)] \\ &\quad + 1 \cdot [uti_{ij}(s) - uti_{ij}(s - s_2)] \\ &= \begin{cases} 0, & s \leq s_1, \\ 0.5, & s_1 < s \leq s_2, \\ 1, & s > s_2 \end{cases} \end{aligned} \quad (14)$$

3 免疫克隆偏好多维 QoS 云资源调度算法

免疫^[10,11]指对“自己(Self)”和“非己(Nonself)”的识别并排出非己的功能.免疫克隆的主要操作包括免疫克隆操作、免疫基因操作和克隆选择操作.

3.1 免疫克隆操作

用固定长度为任务数的整型字符串来生成抗体(任务资源调度方案)的染色体,其中,每个基因位代表任务所分配的资源编号.用户集合 U 的任务集的第 i 个抗体 CA_i 的第 0 代初始解满足式(15).而规模为 N_{Ab} 的第 0 代抗体群满足式(16).

$$CA_i(0) = (ca_i^1, ca_i^2, \dots, ca_i^{mt}), ca_i^k \in \{1, 2, \dots, n\} \quad (15)$$

$$CA(0) = (CA_1(0), \dots, CA_i(0), \dots, CA_{N_{Ab}}(0)), i \in \{1, \dots, N_{Ab}\} \quad (16)$$

其中, ca_i^k 为第 i 个抗体的第 k 个任务 ts_k 所分配的资源编号. mt 为用户集的任务数,且满足式(17).由于 Min-min 算法^[12]具有较好的负载均衡性,初始抗体则利用 Min-min 算法生成.

$$mt = \sum_{j=0}^m ts(j), j \in \{1, 2, \dots, m\} \quad (17)$$

其中, $ts(j)$ 为第 j 个用户的云任务数目.

抗体 CA_i 和抗原之间的适应度 $AgAb_i$ 满足式(18),反映了抗体 CA_i 满足抗原的程度.适应度 $AgAb_i$ 越大,抗体 CA_i 所实现的总效用则越大.

$$\begin{aligned} AgAb_i &= Of_i(uti_{sys}) \\ &= \sum_{k=0}^i \sum_{j=0}^{mt} w_{ca_i^j}(k) \cdot uti_{ca_i^j}(q_{ca_i^j}(k)), \\ &\quad i \in \{1, 2, \dots, N_{Ab}\} \end{aligned} \quad (18)$$

抗体 CA_i 和其他抗体间的亲和度 Ab_iAb_j 满足式(19),亲和度 Ab_iAb_j 越小,抗体间相似度越大,抗体间的抑制作用越强,当 $Ab_iAb_j = 0$ 时,两抗体完全相同.

$$\begin{aligned} Ab_iAb_j &= Avg\{D_{ij}\} = \frac{1}{N-1} \cdot \sum_{\forall i,j} \|CA_i - CA_j\| \\ &= \frac{1}{N-1} \cdot \sum_{\forall i,j} \sqrt{\sum_{k=0}^{mt} (ca_i^k - ca_j^k)^2}, \\ &\quad i \neq j; i, j \in \{1, 2, \dots, N_{Ab}\} \end{aligned} \quad (19)$$

其中, $\|CA_i - CA_j\|$ 为抗体 CA_i 和抗体 CA_j 的欧几里德距离, N_{Ab} 为抗体规模数.

抗体群 CA 的克隆操作 T_c^C 如式(20).

$$\begin{aligned} CA'(it) &= T_c^C(CA(it)) \\ &= (T_c^C(CA_1(it)), T_c^C(CA_2(it)), \dots, T_c^C(CA_{N_{Ab}}(it)))^T \end{aligned} \quad (20)$$

其中, $CA'_i(it) = T_c^C(CA_i(it)) = Q_i \times CA_i(it)$, Q_i 是元素为 1 的 q_i 维行向量,即为抗体 $CA_i(it)$ 的 q_i 克隆.

$$\begin{aligned} q_i(it) &= f(N_c, AgAb_i(CA_i(it)), Ab_iAb_j(CA_i(it))) \\ &= Int\left(N_c \cdot \frac{AgAb_i(it)}{\sum_{k=0}^{mt} AgAb_k(it)} \cdot Ab_iAb_j(CA_i(it))\right), \\ &\quad i \in \{1, \dots, N_{Ab}\} \end{aligned} \quad (21)$$

其中, $N_c > N_{Ab}$ 是与克隆规模有关的设置值, $Int(\cdot)$ 为向上取整.克隆规模根据适应度 $AgAb_i$ 和亲和度 Ab_iAb_j 进行自适应调整,当抗体间抑制越小,抗原刺激越大时,克隆规模越大,反之则反是.克隆后的种群变为:

$$\begin{aligned} CA'(it) &= (CA'_1(it), CA'_2(it), \dots, CA'_{N'_{Ab}}(it))^T, \\ N'_{Ab} &= \sum_{i=0}^{N_{Ab}} q_i \end{aligned} \quad (22)$$

其中,

$$\begin{cases} CA'_i(it) = \{CA'_{ij}(it)\} = \{CA_{i1}(it), CA_{i2}(it), \dots, CA_{iq_i}(it)\}, \\ CA'_{ij}(it) = CA_{ij}(it) = CA_i(it), j \in \{1, 2, \dots, q_i\} \end{cases} \quad (23)$$

3.2 免疫基因操作

免疫基因操作主要包括克隆重组和克隆变异,本文选用具有多目标优化性能较好的模拟二进交叉(Simulated Binary Crossover, SBX)和多项式变异(Polynomial Mutation, PM)^[13].SBX 重组如式(24).

$$ca_i'^k = \begin{cases} 0.5 \cdot [(1 + \beta_k) \cdot ca_i^k + (1 - \beta_k) \cdot ca_j^k], \\ \quad \text{if } r(0, 1) \geq 1/\alpha_k; \forall i, j, i \neq j; k \in \{1, 2, \dots, mt\}, \\ 0.5 \cdot [(1 - \beta_k) \cdot ca_i^k + (1 + \beta_k) \cdot ca_j^k], \\ \quad \text{otherwise; } \forall i, j, i \neq j; k \in \{1, 2, \dots, mt\} \end{cases} \quad (24)$$

其中, β_k 满足式(25), α_k 满足式(26), $r(0, 1)$ 为 $(0, 1)$ 区间高斯分布随机数, $ca_i'^k$ 是 ca_i^k 和 ca_j^k 重组后的结果.

$$\beta_k = \begin{cases} [\alpha_k \cdot u]^{\frac{1}{\eta_c + 1}}, \text{ if } u(0, 1) \leq 1/\alpha_k \\ [2 - \alpha_k \cdot u]^{-\frac{1}{\eta_c + 1}}, \text{ otherwise} \end{cases} \quad (25)$$

其中, u 为 $(0, 1)$ 区间高斯分布随机数, η_c 是重组指标参数, α_k 满足式(26).

$$\begin{aligned} \alpha_k &= 2 - \\ &\left(1 + 2 \cdot \left(\frac{\min(ca_u^k - \max(ca_i^k, ca_j^k), \min(ca_i^k, ca_j^k) - ca_l^k)}{\max(ca_i^k, ca_j^k) - \min(ca_i^k, ca_j^k)}\right)\right)^{-(\eta_c + 1)} \end{aligned} \quad (26)$$

其中, ca_u^k 和 ca_l^k 表示全部抗体的第 k 基因位的最大和最小值, η_c 越大, 重组后新抗体和原抗体越相似.

PM 变异的定义如式(27).

$$ca_i^k = ca_u^k + \delta_k \cdot (ca_u^k - ca_l^k), i \in \{1, 2, \dots, N'_{Ab}\} \quad (27)$$

其中, ca_i^k 和 ca_i^k 第 k 基因位变异前和变异后的值. δ_k 是 PM 变异微调参数, 且满足式(28).

$\delta_k =$

$$\begin{cases} \left[\frac{2 \cdot v + (1 - 2 \cdot v) \cdot \left(\frac{\max((ca_u^k - ca_i^k), (ca_i^k - ca_l^k))}{ca_u^k - ca_l^k} \right)^{\eta_m + 1}}{1} \right]^{\frac{1}{\eta_m + 1}}, & \text{if } v(0, 1) \leq \frac{1}{2}, \\ 1 - \left[\frac{2 \cdot (1 - ca_i^k) + 2 \cdot (ca_i^k - 0.5) \cdot \left(\frac{\max((ca_u^k - ca_i^k), (ca_i^k - ca_l^k))}{ca_u^k - ca_l^k} \right)^{\eta_m + 1}}{1} \right]^{\frac{1}{\eta_m + 1}}, & \text{otherwise} \end{cases} \quad (28)$$

其中, v 为 $(0, 1)$ 区间高斯分布随机数, η_m 是重组指标参数, η_m 越大, ca_i^k 的变异将越小. 当 ca_i^k 为 ca_u^k 或 ca_l^k 时, ca_i^k 则退化为区间 $[ca_l^k, ca_u^k]$ 的随机值.

PM 变异概率 p_m 的定义如式(29).

$$p_m = \begin{cases} (1 + p) \cdot p_{\min} - 2 \cdot p \cdot p_{\min} \cdot \left(\frac{it}{it_{\max}} \right), & \text{if } it \leq it_{\max}/2; p \in (0, 1), \\ p_{\min}, & \text{otherwise} \end{cases} \quad (29)$$

其中, $p_{\min}$ 为预定义最小变异概率, p 为预定义变异级值, it 为当前克隆代数, $it_{\max}$ 为预定义最大代数.

SBX 重组操作实现了抗体之间信息的交换, 随着抗体种群的进化而自适应收敛性, PM 变异操作对抗体进行了自适应微调, p_m 实现了在各代克隆过程中以适当的概率选取抗体进行 PM 变异.

3.3 免疫选择操作

克隆选择操作是从克隆操作和免疫基因操作的子代抗体种群中选择出适应度 $AgAb_i$ 最优抗体, 从而形成新的种群. 抗体群 CA'' 的克隆操作 T_s^C 如式(30).

$$\begin{aligned} CA'''(it + 1) &= T_s^C(CA''(it)) \\ &= T_s^C(CA_1''(it), CA_2''(it), \dots, CA_{N'_{Ab}}''(it))^T \\ &= (CA_1'''(it), CA_2'''(it), \dots, CA_{N'_{Ab}}'''(it))^T \quad (30) \end{aligned}$$

高频率的重组和变异容易导致优势模式的丢失, 为了使优势模式在子代抗体基因中得以保留, 算法设计时则根据适应度 $AgAb_i$ 的大小, 选择同父代个体数目相同的个体最优个体组成下一代抗体种群, 即 $CA(it + 1) = T_s^C(CA(it) + CA''(it))$.

3.4 CICP 算法描述

基于免疫响应中多样性抗体共生、少数抗体激活的事实, 结合有应用偏好的云资源调度多维 QoS 优化的特点, 利用免疫克隆操作、免疫基因操作以及克隆选

择操作, 提出免疫克隆偏好多维 QoS 的云资源调度 (cloud resource scheduling by immune clonal with preference, 简称 CICP) 优化算法, 如算法 1 所示.

算法 1 CICP algorithm

输入: cloud resource set R , cloud user set U , cloud probability matrix $P_{n \times m}$, preference interval, 5-QoS utility interval, minimal mutation probability $p_{\min}$, magnitude mutation p , and maximum generations $it_{\max}$;
输出: antibody population $CA(it_{\max})$.

1. Initialization $R, U, P_{n \times m}, p_{\min}, p, it_{\max}, it$, and initialize an antibody population $CA(0)$ by Min-min algorithm.
2. For each it such that $0 \leq it < it_{\max}$ do
3. For each $CA_i(it)$ in $CA(it)$ do
4. For each ca_i^k in $CA_i(it)$ do
5. Calculate preference vector w_{ij} according to preference interval by (8).
6. Update user utility according to 5-QoS utility interval by one of (10) ~ (14).
7. End for
8. Calculate affinity $AgAb_i$ by (18) and $Ab_i Ab_j$ by (19).
9. Clonal $CA_i(it)$ according to (20), and get new antibody population $CA_i'(it)$.
10. Perform simulated binary crossover by (24) to $CA_i'(it)$.
11. Calculate mutation probability p_m by (29).
12. If random(0, 1) < p_m then
13. Perform polynomial mutation by (27) to $CA_i'(it)$.
14. End if
15. Quick sort antibody population $CA_i'(it)$ according to affinity $AgAb_i$ by decreasing ordering.
16. For i such that $0 \leq i < N_{Ab}$ do
17. Selection $CA_i'(it)$ according to (30), and get new antibody population $CA'''(it)$.
18. End for
19. Update memory antibody population.
20. End for
21. $it = it + 1$.
22. End for
23. Return antibody population $CA(it_{\max})$.

CICP 算法的时间复杂度主要由免疫克隆、免疫基因和克隆选择组成. 设 M 是目标函数的个数, N_{Ab} 为抗体群 $CA(0)$ 的规模, 完成免疫克隆操作后的抗体规模为 N'_{Ab} , 免疫基因操作后的抗体规模为 $N'_{Ab}(1 + p) = N'_{Ab} + \lfloor N'_{Ab}/3 \rfloor$, $N_c > N_{Ab}$ 是与克隆规模相关的设置, 一般取 $N_c = 3 \cdot N_{Ab}$, 则在一代克隆过程中, 初始化种群并计算目标值的时间复杂度为 $O(M \times N_{Ab})$; 免疫克隆操作的时间复杂度最差为 $O(M \times N_c \times \log(N_c))$; 免疫基因操作的时间复杂度最差为 $O(M \times N'_{Ab} \cdot (1 + p))$; 克隆选择操作的时间复杂度为 $O(M \times \log(N'_{Ab}) + M \times N'_{Ab})$.

4 实验与性能分析

4.1 实验环境

基于 CloudSim^[14, 15] 云计算仿真平台, 建立 10 个数

据中心和600个用户的资源调度环境.其主要参数如表1所示.虚拟机数为100,其中每台虚拟机的VCPU数为 $[1,2]$ 之间,每个用户的任务数为 $[5,10]$ 之间,其计算量为 $[10000,20000]$ MI之间,数据通信量为 $[10,30]$ Mbps之间.各个任务间服从Poisson分布且相互独立.任务 ts_j 的最终完成时间 dt_j 满足式(31).

表1 数据中心参数设置

Datacenter ID	Machine number	PE/Machine	Cost (\$/MI)	Architecture and OS
DC.0, DC.1	2	1~4	3	X86/Linux
DC.2, DC.3	4	1~4	6	X86/Linux
DC.4, DC.5	6	4~8	9	X86/Solaris
DC.6, DC.7	8	4~16	12	X86/Solaris
DC.8, DC.9	10	8~16	15	X86/Solaris

$$dt_j = st_j + \eta \cdot \left(\frac{l_j}{1.1 \cdot \max(pe)}, \frac{l_j}{0.9 \cdot \min(pe)} \right) \quad (31)$$

其中, st_j 为任务 ts_j 的提交时间, η 为 $(0,1)$ 区间的正态分布随机数, l_j 为任务 ts_j 的计算量, pe 为各虚拟机的处理能力.

为了评价本文所提出优化算法CICP在云资源调度中的性能,与Max-min^[12]算法、Min-min^[12]算法和ICA(Immune Clonal Algorithm)算法^[16]进行了对比分析.其中测试分析的性能指标为可用性(Availability)、负载均衡离差(Load Balancing Deviation)^[17]和有效时间(Valid time)^[18].算法CICP的主要参数如表2所示.

表2 CICP算法参数

参数	描述
r, u, v	$(0,1)$ 之间高斯分布随机数
η_c, η_m	重组指标参数
$p_{\min}$	预定义最小变异概率
p	预定义变异级值
$it_{\max}$	预定义最大代数
it	当前克隆代数
η	$(0,1)$ 之间正态分布随机数

4.2 性能分析

(1)可用性.可用性 ur 反映任务 ts_j 首次被提交,并在最终完成时间 dt_j 之前成功返回的概率,在任务 ts_j 被选择资源 r_i 的前提下可用性 ur 满足式(32).

$$ur = \frac{1}{m} \sum_{j=0}^m \sum_{i=0}^n \xi_{ij}, \xi_{ij} = \begin{cases} 1, & \text{if } ts_j \text{ success in } r_i, \\ 0, & \text{otherwise} \end{cases} \quad (32)$$

如图2所示,CICP算法的可用性相对于同等条件下的标准的Max-min算法和Min-min算法提高8.3%.且当用户数大于400后,CICP算法可用性仍维持在 $[79\%, 83.6\%]$ 较高水平且保持稳定的范围内.

(2)负载均衡离差.负载均衡离差 σ 反映了资源根据其自身能力和应用偏好而实现资源公平利用率,避免资源负载不均.负载均衡离差 σ 满足式(33).

$$\sigma = \sqrt{\frac{1}{n-1} \cdot \sum_{i=0}^n (ld_i - \bar{ld})^2}, i \in \{1, 2, \dots, n\} \quad (33)$$

其中, ld_i 满足式(34).

$$d_i = \frac{c_i}{\sum_{i=0}^n c_i} \bigg/ \frac{l_j}{\sum_{j=0}^m l_j}, i \in \{1, 2, \dots, n\}; j \in \{1, 2, \dots, m\} \quad (34)$$

其中, c_i 为资源 r_i 的费用, l_j 为任务 ts_j 的计算量.

如图3所示,随着用户数的不断增加,负载均衡离差在不断的增加,但CICP算法同Max-min算法和Min-min算法相比,负载均衡离差维持在 $[0.05, 0.17]$ 之间,同比降低0.28,实现了较好的资源公平利用,并在用户数量大于400后,CICP算法可以维持在一个相对稳定负载均衡水平.

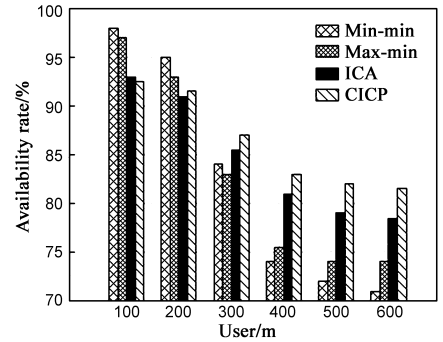


图2 可用性

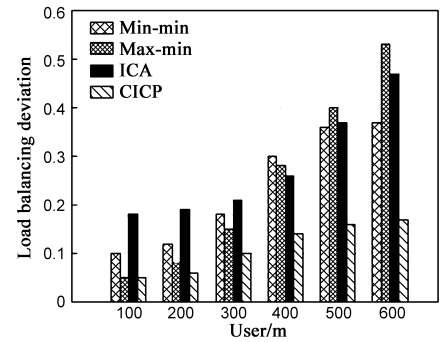


图3 负载均衡离差

(3)有效时间.有效时间 $\bar{vt}$ 反映了云系统中任务执行时间的有效利用率,如式(35)所示.其中时间的无效利用主要体现在任务在排队系统中等待被执行的时间.

$$\bar{vt} = \frac{1}{m} \cdot \sum_{j=0}^m \frac{l_j / pe_i}{rdt_j - rst_j}, j \in \{1, 2, \dots, m\}; i \in \{1, 2, \dots, n\} \quad (35)$$

其中, rdt_j 和 rst_j 分别为任务 ts_j 的实际提交时间和最终完成时间, pe_i 为任务 ts_j 所分配的资源 r_i 的实际处理能力.

如图4所示,随着用户数不断增加,用户的有效时

间 $\bar{v}_i$ 呈现不断下降的趋势,同 Max-min 算法和 Min-min 算法相比,CICP 算法的有效时间维持在 $[0.68, 0.95]$ 之间,提高近 0.39,表现较优的时间利用。

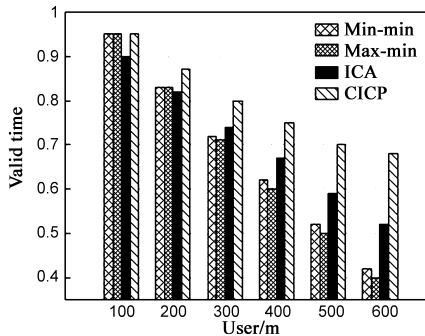


图4 有效时间

5 结语

云计算作为一种新的计算范式,提供非平凡 QoS 服务.资源调度是云系统中的一个核心且富有挑战的重要构成部分.针对该问题本文从理论上对云资源调度模型进行了系统的建模分析,提出了一种基于免疫克隆的偏好多维 QoS 的云资源调度优化算法 CICP,本文的主要贡献:(1)从理论上系统的提出了一种云资源调度 CRS 模型,并对应用偏好和多维 QoS 空间中的效用进行了量化分析;(2)针对量化的应用偏好,结合具有快速多目标优化能力的免疫克隆算法,实现了对 t -QoS 空间中的多维 QoS 的改善;(3)完成了包括最终完成时间、响应时间、费用、可用性、安全性在内的 5-QoS 进行量化,实现了可用性、负载均衡离差和有效时间三个主要性能指标的改善.实验结果及分析表明算法 CICP 满足了云计算系统的可伸缩性和自适应性需求,有效的保证了用的性能 QoS 和经济 QoS.在进一步的研究过程中,我们将更加深入、全面、系统的分析和完善相关性能指标的优化及改善,同时将该算法部署到真实的云计算环境中。

参考文献

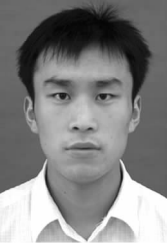
- [1] Foster I, Zhao Y, Raicu I, et al. Cloud computing and grid computing 360-degree compared[A]. Proc of the Grid Computing Environments Workshop, GCE 2008 [C]. New York: IEEE Press, 2008. 1 – 10.
- [2] Buyya R, Yeo C S, Venugopal S, et al. Cloud computing and e-emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility[J]. Future Generation Computer Systems, 2009, 25(6): 599 – 616.
- [3] Armbrust M, Fox A, Griffith R, et al. A view of cloud computing[J]. Communications of the ACM, 2010, 53(4): 50 – 58.
- [4] Mell P, Grance T. The NIST definition of cloud computing[J]. Communications of the ACM, 2010, 53(6): 50.
- [5] Wei G Y, Vasilakos A V, Zheng Y, et al. A game-theoretic method of fair resource allocation for cloud computing services[J]. Journal of Supercomputing, 2010, 54(2): 252 – 269.
- [6] 王铁军, 刘恒, 孙明, 等. 资源定位服务的分布式生成树模型及算法研究[J]. 电子学报, 2011, 39(2): 364 – 369.
Wang Tiejun, Liu Heng, Sun Ming, et al. Research on the model and algorithms based on distributed spanning tree for resource location service[J]. Acta Electronica Sinica, 2011, 39(2): 364 – 369. (in Chinese)
- [7] Zhao G, Liu J, Tang Y, et al. Cloud computing: A statistics aspect of users[A]. Proc of the First International Conference of Cloud Computing, CloudCom 2009 [C]. Heidelberg: Springer Verlag Press, 2009. 347 – 358.
- [8] Shen X, Guo Y, Chen Q, et al. A multi-objective optimization evolutionary algorithm incorporating preference information based on fuzzy logic[J]. Computational Optimization and Applications, 2010, 46(1): 159 – 188.
- [9] 杨咚咚, 焦李成, 公茂果, 等. 求解偏好多目标优化的克隆选择算法[J]. 软件学报, 2010, 21(1): 14 – 33.
Yang Dongdong, Jiao Licheng, Gong Maoguo, et al. Clone selection algorithm to solve preference multi-objective optimization[J]. Journal of Software, 2010, 21(1): 14 – 33. (in Chinese)
- [10] Ulker E, Arslan A. Automatic knot adjustment using an artificial immune system for B-spline curve approximation[J]. Information Sciences, 2009, 179(10): 1483 – 1494.
- [11] Gao X Z, Wang X, Ovaska S J. Fusion of clonal selection algorithm and differential evolution method in training cascade-correlation neural network[J]. Neurocomputing, 2009, 72(10 – 12): 2483 – 2490.
- [12] Singh M, Suri P K. QPS max-min min-min: A QoS based predictive max-min, min-min switcher algorithm for job scheduling in a grid[J]. Information Technology Journal, 2008, 7(8): 1176 – 1181.
- [13] Chen J Y, Lin Q Z, Ji Z. A hybrid immune multiobjective optimization algorithm[J]. European Journal of Operational Research, 2010, 204(2): 294 – 302.
- [14] 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 & Experience, 2011, 41(1): 23 – 50.
- [15] Belalem G, Tayeb F Z, Zaoui W. Approaches to improve the resources management in the simulator CloudSim[A]. Proc of the First International Conference of Information Computing and Applications, ICICA 2010 [C]. Heidelberg: Springer Verlag Press, 2010. 189 – 196.
- [16] Shi G, Jing Y. Research of improved immune clonal algorithms and its applications[A]. Proc of the 2009 IEEE International Conference on Computational Intelligence for Mea-

surement Systems and Applications, CIMS A 2009 [C]. New York: IEEE Press, 2009. 212 – 215.

[17] Xiao L, Zhu Y, Ni L M, et al. Incentive-based scheduling for market-like computational grids [J]. IEEE Transactions on Parallel and Distributed Systems, 2008, 19(7): 903 – 913.

[18] Li C, Li L. A system-centric scheduling policy for optimizing objectives of application and resource in grid computing [J]. Computers & Industrial Engineering, 2009, 57 (3): 1052 – 1061.

作者简介



孙大为 男, 1985 年生于安徽六安. 东北大学信息科学与工程学院博士研究生. 研究方向为云计算、可信计算.
E-mail: sundaweicn@163.com



常桂然 男, 1946 年生于河北曲周, 博士. 现任东北大学计算中心教授, 博士生导师. 研究方向为高性能计算、计算机网络、信息安全.
E-mail: chang@neu.edu.cn

李凤云 女, 1981 年生于吉林公主岭. 东北大学信息科学与工程学院博士研究生, 东北大学计算中心助教. 研究方向为物联网、云计算.

王 川 男, 1980 年生于辽宁沈阳. 东北大学信息科学与工程学院博士研究生. 研究方向为云计算、搜索引擎.

王兴伟 男, 1968 年生于内蒙古包头, 博士. 现任东北大学信息科学与工程学院教授, 博士生导师. 研究方向为下一代互联网、移动互联网等.