一种基于负载均衡的云资源调度方法

陈 斌[1],甘茂林[2],李 娟[1]
(1. 海军工程大学 电子工程学院,湖北 武汉 430033;
2. 海军驻广州地区通信军事代表室,广东 广州 510220)

摘 要:虚拟化技术可以实现云系统中资源的按需分配,但同时可能会造成部分物理服务器负载过重,从而导致系统性能降低。云系统负载均衡,是决定系统计算和服务能力的核心因素。在对云资源调度问题的形式化描述基础上,定义了负载均衡系数,构建了云资源调度负载均衡模型。针对云资源调度中的用户需求及服务器属性的差异性问题,提出了一种基于最大需求优先和最闲服务器被选原则的云计算资源调度方法,并进行了算法设计与分析,给出了系统负载均衡系数的动态计算公式。仿真实验结果表明,所提出的方法可得到理想的云计算负载均衡结果,与随机调度、轮转调度等传统算法相比具有优越的负载均衡性能和广泛的适应性。与此同时,所提出的方法可合理协调用户需求和服务器属性,以实现云资源负载均衡。

关键词:负载均衡;云资源调度;用户需求;服务器属性
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2017)06-0051-05
doi:10.3969/j.issn.1673-629X.2017.06.011

A Cloud Resource Scheduling Method Based on Load Balancing

CHEN Bin[1], GAN Mao-lin[2], LI Juan[1]
(1. College of Electronic Engineering, Naval University of Engineering, Wuhan 430033, China;
2. Representative Office on Military Communication, Navy in Guangzhou, Guangzhou 510220, China)

Abstract: Virtualization technology can achieve the distribution of cloud resources according to need. However, it also may cause the heavy load for some physical servers which make system performance reduced. Load balancing of cloud system is the core factor that determines the system computing and service capability. Based on the formal description of cloud resource problem, load balancing coefficient has been defined and load balancing scheduling model for cloud resource has been built. Then, aiming at differences issues such as user demands and servers properties involving in cloud resource scheduling, a method has been proposed focusing on the principle of maximum demand prior and the freest server selected. It has been implemented for algorithm design and analysis further. Dynamic calculation of load balancing coefficient has been put forward as well. The result of simulation experiment has proved that the method proposed has excellent performance for load balancing and wide adaptability especially comparing to the random scheduling and rotation scheduling algorithm. It has also shown that the method can achieve load balancing of cloud resources with reasonably coordinating user demands and the server properties.

Key words: load balancing; cloud resource scheduling; user demands; server properties

0 引 言

云资源[1-3]在物理上以分布式的共享方式存在,在逻辑上以单一整体的形式呈现给用户。云资源调度是将资源从资源提供方分配给用户的一个过程。先进的云计算资源调度技术[4-7]是提高云计算系统性能的关键之一。而云系统负载均衡[8-10],是决定系统计算和服务能力的核心因素。

传统的分配调度算法[11-12],如轮转法、最小负载优先法、哈希法等,虽然在一定程度上考虑了负载均衡的问题,但很难达到负载均衡的需求。原因来自两方面:一是用户需求的差异性,表现在用户可能在不同时刻提出不同的资源需求;二是服务器属性的差异性,在数据中心的服务器集群中,每个物理服务器的配置是不相同的。

收稿日期:2016-07-10 **修回日期**:2016-10-13 **网络出版时间**:2017-04-28
基金项目:军内科研计划项目(HJ-502-2015-35)
作者简介:陈 斌(1975-),男,博士生,副教授,研究方向为通信与信息系统。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20170428.1703.062.html

针对上述问题,在研究分析云资源调度方法的基础上,提出了一种基于最大需求优先和最闲服务器被选原则的云计算资源调度方法。仿真实验结果表明,该方法在云资源调度过程中通过合理协调用户需求和服务器属性,以达到云资源负载均衡的目的。

1 基本概念

云计算环境中,云资源调度流程如图 1 所示。

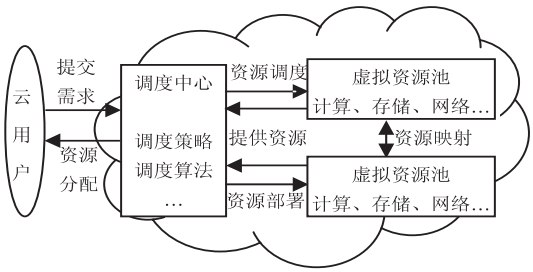


图 1 云资源调度流程

云用户向云平台提交资源申请后,云计算环境中的调度中心节点,根据云调度策略,按照相应的调度算法,进行云资源调度,建立虚拟资源与物理资源的映射,同时将用户所需资源部署到具体的物理节点上,最终将资源分配给用户,在用户申请期限内,提供给用户使用。

从虚拟资源的角度描述,云资源可以看作是若干种不同类型的集合,故有以下定义。

定义 1:云资源(Cloud Source,CS),含 n 类子资源,表示形式为:

$$cs = < cs_1, cs_2, \cdots, cs_n >, cs_i \in \{1, 2, \cdots, \infty\}, i \in \{1, 2, \cdots, n\}, CS = \{cs\}$$

其中,cs_i 表示云资源向量 cs 中含第 i 类子资源的数量值;CS 表示云资源集合。

在云资源集合 CS 定义运算"+""-"">""<":

$$x = < x_1, x_2, \cdots, x_n >, y = < y_1, y_2, \cdots, y_n >, x, y \in CS$$

$$x + y = < x_1 + y_1, x_2 + y_2, \cdots, x_n + y_n >$$

$$x - y = < x_1 - y_1, x_2 - y_2, \cdots, x_n - y_n >$$

$$x > y \Leftrightarrow (x_1 > y_1) \wedge (x_2 > y_2) \wedge \cdots \wedge (x_n > y_n)$$

$$x < y \Leftrightarrow (x_1 < y_1) \wedge (x_2 < y_2) \wedge \cdots \wedge (x_n < y_n)$$

从物理资源角度描述,云资源可以看作是若干物理服务器的集合,即服务器集群,每一台物理服务器上,可提供若干云资源。故有如下定义。

定义 2:物理服务器(Physical Machine,PM)位于云系统中,以 pm. s 表示位于服务器 pm 上的资源总数,pm. v 表示 pm 上的可分配资源,有:

$$pms = < pm.s_1, pm.s_2, \cdots, pm.s_n >$$

$$pmv = < pm.v_1, pm.v_2, \cdots, pm.v_n >$$

$$pm.s, pm.v \in CS, pm.v < pm.s$$

向量 pm. s 是对应服务器 pm 的性能属性,用以描述服务器的资源容量,其内容由物理服务器 pm 决定,在服务器正常工作中,不会发生变化。而向量 pm. v 表达的是对应服务器 pm 的实时状态,用以描述服务器在当前时刻可用资源总量,其内容由物理服务器 pm 当前资源余量决定,在服务器正常工作中,随资源的分配与释放实时变化。

定义 3:物理服务器集群(Physical Machine Cluster,PMC)位于云系统中,含 m 个服务器,表示为:

$$PMC = \{pm_1, pm_2, \cdots, pm_m\}$$

由定义 1、2、3 可知,在包含 m 台物理服务器的集群 PMC 中,其能提供的云资源总量 pmc. s 和可分配资源总量 pmc. v 为:

$$pmc.s = pm_1.s + pm_2.s + \cdots + pm_m.s$$

$$pmc.v = pm_1.v + pm_2.v + \cdots + pm_m.v$$

其中,$i \in \{1, 2, \cdots, m\}$,$pm_i \in PMC$。

2 问题模型

物理服务器是云资源的提供者,而资源的使用者是云系统用户(Cloud User,CU)。

从用户的角度描述,云资源可以看作是云系统满足用户实时需求的服务能力。用户在特定时刻的资源需求 ur 表示为:

$$ur = < cu, cs, ts, te >$$

其中,$cu \in CU$ 表示提出资源需求的云用户;$cs \in CS$ 表示资源需求量;$ts \in [0, \infty)$ 表示云用户提出资源需求的初始时刻;$te \in [0, \infty)$ 表示用户资源占有时间长度。

用户资源需求是一种实时变量,以 $UR(t)$ 表示在 t 时刻云用户的资源需求,则

$$UR(t) = \{ur \mid ur.ts = t\}$$

可以计算,在 t 时刻用户资源需求总量 UR(t). CS 为:

$$UR(t).CS = \sum_{ur_i \in UR(t)} ur_i.cs$$

若在 t 时刻,服务器的集群 PMC 中的可分配资源总量为 pmc. v(t),则:pmc. v(t) > UR(t). CS 为 t 时刻用户资源需求得以满足的必要条件。

从资源调度角度描述,云资源可以看作是云系统为云用户服务的能力。云系统将部署在服务器上的资源按用户需求分配给云用户。故资源分配(Source Assignment,SA)的任务是根据用户提出的资源需求,将用户需求的资源部署在物理服务器上,并分配给用户使用。资源分配的基本规则为:某个用户的一次资源需求中,所有的资源均应部署在同一个物理服务器上。即对于用户需求 ur,以下条件成立时,才可被满足。

$\exists pm \in PMC, pm.v > ur.cs$

以 sa 表示云系统满足 ur 的资源分配,有:

$sa = < ur, pm > = < cu, cs, ts, te, loc >$

其中, $< cu, cs, ts, te > \in UR$; $loc \in PMC$ 表示用户需求的资源所部署的物理服务器位置。

对应于 $UR(t)$,以 $SA(t)$ 表示在 t 时刻云系统满足所有用户的资源分配集合,则:

$SA(t) = \{ < cu, cs, ts, te, loc > | < cu, cs, ts, te > \in UR(t), loc \in PMC\}$

所研究的云资源调度算法以负载均衡为目标,与负载相关的定义如下。

定义 4:服务器负载率(Physical Machine Load Factor),以 f 表示。对于 pm,f 的计算方法如下:

$$pm.f = \frac{pm.s - pm.v}{pm.s}$$

定义 5:服务器集群负载率(Physical Machine Cluster Load Factor),以 F 表示。对于 PMC, F 的计算方法如下:

$$PMC.F = \frac{pmc.s - pmc.v}{pmc.s}$$

定义 6:服务器集群负载均衡系数(Physical Machine Cluster Load Balancing Factor)以 BF 表示。对于 PMC,BF 的计算方法如下:

$$PMC.BF = \sum_{pm_i \in PMC} | pm_i.f - PMC.F |$$

根据以上定义及描述,可对问题建立如下模型:

已知 PMC,UR;

求解 SA;

使得 min PMC.BF

$$\begin{cases} \underbrace{\forall ur \in UR \rightarrow \exists < ur, loc > \in SA \wedge loc \in PMC}_{(1)} \\ \underbrace{\forall pm \in PMC \rightarrow pm.v > 0}_{(2)} \end{cases}$$

其中,min PMC.BF 表示某时刻云系统服务器集群最小的负载均衡系数,(1)保证当前时刻所有的用户需求均被处理,(2)保证所有资源在物理服务器上是可得的。

3　算法设计及分析

3.1　算法策略

云应用中,导致服务器集群负载不均衡的因素有两个[13-15]:一是用户资源需求的随机性,云用户根据自己的实际情况向云系统申请规格不同的资源量,这些不同的资源需求将导致服务器负载的不均衡;二是服务器本身所能提供的资源量也是大小不一的,这种服务器属性同样导致负载的不均衡。针对这两点,提出以下策略。

(1)最大资源需求优先。

在资源部署时,资源量越大,越容易造成负载的不均衡。因而,采用“最大需求者优先”原则,先处理资源申请量大的用户需求。

因此,对应 t 时刻的用户需求集合 $UR(t)$,首先按其中元素资源需求量 ur.cs 根据从大到小进行排序。资源分配时,按排序后的用户资源需求顺序依次进行处理。

(2)最低利用率被选。

对应于每一个 ur,理论上可以由云系统中的任意一台服务器提供。所以对应 m 个服务器,有 m 种分配方案,为$<ur, pm_1>, <ur, pm_2>, \cdots, <ur, pm_m>$,对应这 m 种方案,提供资源的服务器相关参数更新为:

$$\begin{cases} pm_i.v = pm_i.v - ur.cs \\ pm_i.f = pm_i.f - \dfrac{ur.cs}{pm_i.s}, pm_i \in PMC \end{cases}$$

针对这些候选方案,选择 $ua = <ur, pm>$,其中 pm 使得 $pm_i.f$ 最小,即:

$ua = < ur, pm > \Leftrightarrow pm \in PMC \wedge$

$(\forall pm_i \in PMC \rightarrow pm.f \leqslant pm_i.f)$

3.2　算法步骤

根据“最大资源需求优先,最低利用率被选”策略,问题求解步骤如下:

(1)初始化服务器集群 PMC,用户资源分配队列 UA,时间值 t,确定算法终止条件 TEND。

(2)若满足终止条件,算法结束;否则继续步骤(3)。

(3)在 UA 队列中寻找已到期资源服务(即 ua.tb+ua.te< t),释放物理服务器 ua.loc 资源,并更新服务器负载率(即 pmv=pmv+ua.cs,pm.f=1-pmv/pms)。

(4)读入当前用户资源需求集合。

(5)根据资源需求量大小对用户资源需求排序。

(6)按资源需求量大小,对每一个资源请求 ur:

(6.1)依次将其预分配到每一台服务器上,计算相应服务器均衡负载率。

(6.2)记录最小均衡负载率及相应服务器 pm。

(6.3)若 pm 合法,进行资源部署,更新 pm 状态,并创建资源服务项,将其加入资源服务队列中,转到步骤(7),否则转步骤(4)。

(6.4)无法满足 ur,资源分配失败。

(7) $t = t + 1$。

(8)转到步骤(2)。

算法描述如图 2 所示。

```
t=0,UA=NULL,初始化 PMS
    while(t<TEND)
    for all uain UA which ua. tb+te<t//用户资源服务期满
    pm=ua. loc,pm. v=pm. v+ua. cs, pm. f=1-pm. v/pm. s
    //释放用户资源
    read UR //读入当前用户资源请求集合
    sort UR from large to small according to ur. cs //根据资源请求大小对资源队列排序
    fmin=1,loc=-1
    for each ur in UR
    for each pm in PMC
          f=(pms-pmv+ur. cs)/pms
    if (f>0) & (f<fmin)
        fmin=f,loc=pm
      if loc>=0
        ua=new(<ur,loc>)
    pm=ua. loc,pmv=pmv-ua. cs ,pm. f=1-pmv/pms
    //部署资源到负载率最低的服务器
        add ua to UA
      elseSource Assignment Fail for ur
    t=t+1;
```

图 2 算法描述

3.3 算法分析

(1)算法时间复杂度。

设云系统中 PMC 的服务器数量为 m,CU 中用户总数为 u,用户提出资源需求的概率为 p,用户资源占有平均时长为 τ,则在每个时刻 t,UR 中元素平均个数,即需要处理的资源需求数量平均值为 up,服务集合 UA 平均长度为 up_τ。

算法从 $T=0$ 开始运行,至 $T=$TEND 终止,在每一轮运行中,主要完成遍历 UA、UR 排序、遍历 UR×PMC。故可计算其时间复杂度为:$\Theta(\mathrm{TEND}*\mathrm{up}(\tau+\log(\mathrm{up})+m))$。

(2)相关参数计算。

在算法执行过程中,随着资源的分配与释放,物理服务器及服务器集群的相关参数也会发生变化。

在时刻 t,首先系统释放到期服务资源,再为该时刻的用户需求进行资源分配。若:

$$\begin{cases}\mathrm{UA}(t-1)=\{<\mathrm{cu},\mathrm{cs},\mathrm{ts},\mathrm{te},\mathrm{loc}>\mid <\mathrm{cu},\mathrm{cs},\mathrm{ts},\mathrm{te}>\in\bigcup_{i<t}\mathrm{UR}(i),\mathrm{ts}<t,\mathrm{te}\geqslant t\}\\ \mathrm{UR}(t)=\{\mathrm{ur}\}\\ \mathrm{SA}(t)=\{<\mathrm{ur},\mathrm{loc}>\mid \mathrm{ur}\in\mathrm{UR}(t)\}\end{cases}$$

则:

$$\mathrm{UA}(t)=\mathrm{UA}(t-1)+\mathrm{SA}(t)-\{<\mathrm{cu},\mathrm{cs},\mathrm{ts},\mathrm{te},\mathrm{loc}>\mid <\mathrm{cu},\mathrm{cs},\mathrm{ts},\mathrm{te},\mathrm{loc}>\in\mathrm{UA}(t-1),\mathrm{te}=t\}$$

一轮算法执行完毕后,PMC 中服务器可用资源及负载率为:

$$\mathrm{pm}_i(t).v=\mathrm{pm}_i(t-1).v+\sum_{\substack{\mathrm{ua}_j.\,\mathrm{te}=t\\ \mathrm{ua}_j.\,\mathrm{loc}=\mathrm{pm}\\ \mathrm{ua}_j\in\mathrm{UA}}}\mathrm{ua}_j.\,\mathrm{cs}-\sum_{\substack{\mathrm{sa}_j.\,\mathrm{loc}=\mathrm{pm}\\ \mathrm{sa}_j\in\mathrm{SA}(t)}}\mathrm{sa}_j.\,\mathrm{cs},\mathrm{pm}_i\in\mathrm{PMC}$$

$$\mathrm{pm}_i(t).f=1-\frac{\mathrm{pm}_i(t).v}{\mathrm{pm}_i.s}$$

PMC 的可用资源、负载率及负载均衡系数为:

$$\mathrm{PMC}(t).\mathrm{v}=\mathrm{PMC}(t-1)+\sum_{\substack{\mathrm{ua}_j.\,\mathrm{te}=t\\ \mathrm{ua}_j\in\mathrm{UA}}}\mathrm{ua}_j.\,\mathrm{cs}-\sum_{\mathrm{sa}_j\in\mathrm{SA}(t)}\mathrm{sa}_j.\,\mathrm{cs}$$

$$\mathrm{PMC}(t).\mathrm{F}=1-\frac{\mathrm{PMC}(t).\mathrm{v}}{\mathrm{PMC}.\mathrm{s}}$$

$$\mathrm{PMC}(t).\mathrm{BF}=\sum_{\mathrm{pm}_i\in\mathrm{PMC}}\mid \mathrm{pm}_i(t).\mathrm{f}-\mathrm{PMC}(t).\mathrm{F}\mid$$

4 实验及结果分析

为进一步研究算法效果,通过仿真实验分析算法演化过程及结果。实验中,取服务器集群数量为 6,仅考虑一种资源的情况,服务器资源参数如表 1 所示。

表 1 服务器资源参数

云资源	第一组	第二组
$\mathrm{pm}_1.\mathrm{s}$	1 000	1 400
$\mathrm{pm}_2.\mathrm{s}$	1 000	1 250
$\mathrm{pm}_3.\mathrm{s}$	1 000	1 100
$\mathrm{pm}_4.\mathrm{s}$	1 000	900
$\mathrm{pm}_5.\mathrm{s}$	1 000	750
$\mathrm{pm}_6.\mathrm{s}$	1 000	600

其余参数设定为:CU 中用户总数为 20,用户提出资源需求的概率为 0.5,用户资源需求量均匀分布在(0,50)之间,用户资源占有时长均匀分布在(0,10)之间,算法演化时长为 100。算法演化过程中,服务器负载率及服务器集群负载率变化如图 3 所示。

两组参数对应的服务器集群负载均衡系数为:

$\mathrm{PMC}^{(1)}.\mathrm{BF}=0.062\ 21$

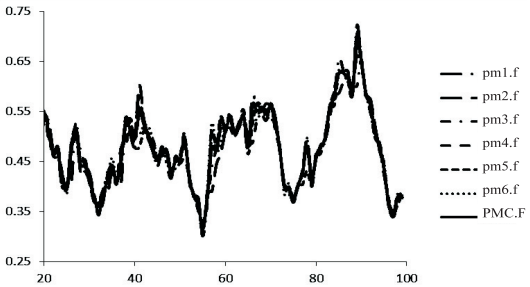
$\mathrm{PMC}^{(2)}.\mathrm{BF}=0.055\ 72$

实验结果表明,服务器集群中,无论服务器属性是否有差异性,提出算法均可达到较好的负载均衡性能。

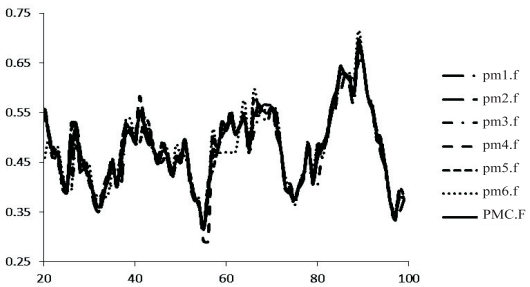
为进一步分析算法优越性,以随机分配方案、轮转分配方案、最低利用率被选方案(与所提算法相比,不对资源需求进行排序)作为对比,在相同两组服务器资源参数下进行相同的仿真实验。不同方案下,服务器集群负载均衡系数值如图 4 所示。

实验结果表明,提出的负载均衡资源调度算法相比于传统资源调度算法具有明显的优越性,特别是在服务器属性差异性较大的情况下。表 2 列出了第二组参数下,提出的“最大资源需求优先,最低利用率被

选”资源调度策略和随机调度策略下,各服务器及服务器集群的平均负载率数据。

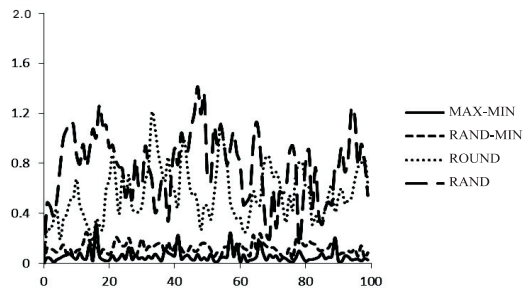


(a)第一组实验结果

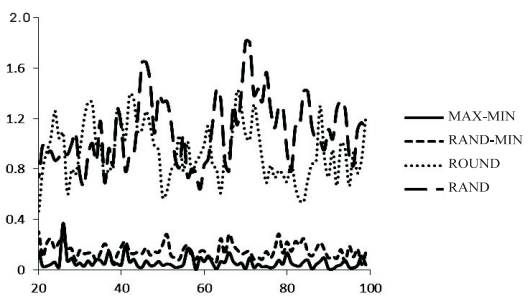


(b)第二组实验结果

图 3　服务器及服务器集群负载率



(a)第一组实验结果



(b)第二组实验结果

图 4　不同策略下服务器集群负载均衡系数对比

表 2　两种策略下的服务器负载率

负载率	所提方案	随机方案
$pm_1.f$	0.476	0.219
$pm_2.f$	0.477	0.354
$pm_3.f$	0.474	0.434
$pm_4.f$	0.479	0.446
$pm_5.f$	0.481	0.576
$pm_6.f$	0.477	0.618
PMC.F	0.477	0.477

5　结束语

在云资源调度中,为保证服务器集群负载均衡,在构建云资源调度负载均衡模型的基础上,提出了一种基于最大需求优先,最闲服务器被选原则的云计算资源调度方法。实验结果表明,与传统资源调度算法相比,提出方法具有优越的负载均衡性能和广泛的适应性。该方法可合理协调用户需求和服务器属性,以实现云资源负载均衡。

参考文献:

[1] 陈　星,张　颖,张晓东,等. 基于运行时模型的多样化云资源管理方法[J]. 软件学报,2014,25(7):1476-1491.

[2] 洪　斌,彭甫阳,邓　波,等. 云资源状态监控研究综述[J]. 计算机应用与软件,2016,33(6):1-6.

[3] 赖培源,马卫民,刘　艺,等. 云资源池网络自动化部署技术研究与实践[J]. 电信科学,2015,31(7):96-103.

[4] Singh S, Chana I. A survey on resource scheduling in cloud computing: issues and challenges[J]. Journal of Grid Computing, 2016, 14(2): 217-264.

[5] Singh S, Chana I. QRSF: QoS-aware resource scheduling framework in cloud computing[J]. Journal of Supercomputing, 2015, 71(1): 241-292.

[6] 魏　蔚,刘　扬,杨卫东. 一种通用云计算资源调度问题的快速近似算法[J]. 计算机研究与发展,2016,53(3):697-703.

[7] 李媛祯,杨　群,赖尚琦,等. 一种 Hadoop Yarn 的资源调度方法研究[J]. 电子学报,2016,44(5):1017-1024.

[8] 孟　蒙,茅　苏. 基于云计算的可反馈负载均衡策略的研究[J]. 计算机技术与发展,2014,24(10):135-139.

[9] Milani A S, Navimipour N J. Load balancing mechanisms and techniques in the cloud environments: systematic literature review and future trends[J]. Journal of Network and Computer Applications, 2016, 71: 86-98.

[10] 陶永才,张丹丹,石　磊,等. 基于 Maxdiff 直方图的 MapReduce 负载均衡研究[J]. 小型微型计算机系统,2016,37(3):417-421.

[11] Levitin A. 算法设计与分析基础[M]. 潘　彦,译. 第 3 版. 北京:清华大学出版社,2015.

[12] 王永明,尹红丽,秦开大. 作业车间调度理论及其优化方法研究[M]. 北京:科学出版社,2013.

[13] Pushpalatha K, Shaji R S, Jayan J P. A cost effective load balancing scheme for better resource utilization in cloud computing[J]. Journal of Emerging Technologies in Web Intelligence, 2014, 6(3): 280-290.

[14] 宋　浒. 面向用户服务需求的云计算管理机制研究[D]. 合肥:中国科学技术大学,2013.

[15] Madni S H H, Latiff M S A, Coulibaly Y, et al. Resource scheduling for infrastructure as a service (IaaS) in cloud computing: challenges and opportunities[J]. Journal of Network and Computer Applications, 2016, 68(6): 173-200.