

# 基于特征尺度均衡的 Linux 系统双阈值任务调度算法

崔永君 张永花

(兰州交通大学电子与信息工程学院 兰州 730070)

**摘要** 在嵌入式 Linux 操作系统的设计和应用中,操作系统经过移植后运行在不同的硬件平台上,它需要一种有效的任务调度算法来实现进程管理和内存管理,以提高系统运行效率。提出了一种基于特征尺度均衡的 Linux 系统双阈值任务调度算法,分析了嵌入式 Linux 的内核结构,构建了系统任务调度模型。该算法对以往各类型任务的到达频率、执行时间等信息流进行统计,并提取特征尺度,在 Linux 系统的全局任务调度中心将所有任务进行融合,输入系统总调度器,得到尺度优化目标函数,进行特征尺度均衡处理;把均衡后的特征尺度时间轴划分成各个相邻但不重合的任务匹配平滑窗口,通过双阈值权衡判决实现对 Linux 系统的任务调度。仿真结果表明,采用该算法进行 Linux 嵌入式任务调度,具有较高的执行效率,CPU 利用率高,整体性能优于传统算法。

**关键词** 特征尺度, Linux 操作系统, 任务调度, 均衡

**中图分类号** TP302.7 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.6.039

## Linux System Dual Threshold Scheduling Algorithm Based on Characteristic Scale Equilibrium

CUI Yong-jun ZHANG Yong-hua

(School of Electronics & Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China)

**Abstract** In the design and application of embedded Linux operating system, operating system runs on different hardware platforms after transplantation, and it needs a task scheduling algorithm for effective implementation of process management and memory management to improve the operational efficiency of the system. Linux system dual threshold scheduling algorithm was proposed based on characteristic scale equilibrium. The kernel structure of embedded Linux was analyzed. The system task scheduling model was constructed. According to the various classifications of information such as task arrival rate, execution time, etc, the scale features are extracted. In the global task scheduling center, all the task data are integrated and input to the total system scheduler, and the scale optimization objective function is obtained. The feature scale balanced processing is taken. The characteristic time shaft is divided into the adjacent but not overlap task matching smoothing window, and the double threshold trade-off decision is used for task scheduling in Linux system. The simulation results show that the new algorithm has higher efficiency in Linux embedded task scheduling, utilization rate of CPU is better, and the overall performance is better than the traditional algorithm.

**Keywords** Characteristic scale, Linux operating system, Task scheduling, Equilibrium

## 1 引言

随着嵌入式系统在各种硬件应用设备中的植入运行, Linux 操作系统以其性能稳定可靠、代码资源开放、系统可裁剪的优点逐渐受到人们重视。嵌入式 Linux 操作系统可以支持多个文件和任务执行,对不同文件系统的操作和控制纳入一个统一的框架中,从而使用户可以通过一组标准的系统调用操作不同的文件任务系统,实现系统对并行处理任务的调度。在嵌入式 Linux 操作系统的设计和应用中,操作系统经过移植后运行在不同的硬件平台上,进行进程管理、内存管理、文件系统和设备管理等,这一过程是一个复杂的多任务的调度和集成处理过程,需要一种有效的任务调度算法实现进程管理和内存管理等,以提高系统运行效率。因此研究一种

高效的嵌入式 Linux 系统的任务调度算法,对提高操作系统的性能具有重要意义<sup>[1]</sup>。

由于 Linux 的开源性和无限扩展性,对嵌入式 Linux 操作系统中的任务调度算法进行优化设计一直是理论研究的难点和热点。传统方法中, Linux 系统的任务调度算法设计主要是采用分簇能耗调度算法、PSO 遗传进化任务调度算法及采用人工干预的半自动 Linux 任务整合算法,调动多芯片集成处理器进行分布式集成操作,实现任务调度优化。其中,文献[2]提出在 Linux 多处理器芯片上进行系统任务调度,采用的任务映射与调度在多处理器片上进行系统设计,算法不适用于集成任务调度的情况。文献[3]中李静梅、王雪等人提出一种改进的优先级列表任务调度算法,在 Linux 嵌入式系统中进行异构多核处理器任务调度高性能计算,然而该算法在

到稿日期:2014-08-13 返修日期:2014-11-13 本文受国家自然科学基金项目:静电感应晶体管栅源击穿电压的提高与关键工艺研究(61366006)资助。

崔永君(1973—),男,硕士,副教授,主要研究方向为计算机网络、Linux 操作系统, E-mail: 987654321\_cui@163.com; 张永花(1974—),女,硕士,副教授,主要研究方向为计算机操作系统、数据库原理研究。

任务分配阶段利用任务复制和区间插入技术,导致任务调度的特征尺度失衡,影响任务分配的准确性。文献[4]提出基于混合粒子群算法的 Linux 嵌入式系统任务调度算法,把 Linux 系统中的进程管理和内存管理系统进行任务调度整合,使用负载均衡的种群初始化进行任务执行算法设计,一定程度上节省了能耗开支。文献[5]提出基于免疫算法的多目标约束嵌入式任务调度,算法在整合精度上有改观,但是在通信实践方面有欠缺。文献[6]提出一种基于自适应能量补给和均衡的 Linux 系统进程生存期评估算法,采用遗传进化资源分配方案,对多台计算机同时执行任务,有效节省了资源开销,但算法没能对随机任务分配的多分簇节点实现能耗优化调度,还有待改进。文献[7]提出一种基于模糊聚类单阈值均衡尺度分解的嵌入式 Linux 系统任务调度策略,算法在进行任务信息流特征尺度分解和均衡设计中存在任务调度效益失衡的问题。

针对上述问题,本文提出一种基于特征尺度均衡的 Linux 系统双阈值任务调度算法,首先进行嵌入式 Linux 的内核结构分析与系统任务调度模型设计,提出 Linux 系统任务信息流特征尺度均衡算法,以此为基础实现 Linux 系统双阈值任务调度算法的改进,用仿真实验验证本文算法的有效性和可行性,最后进行结论分析。

## 2 Linux 嵌入式系统任务调度模型的构建与问题描述

### 2.1 嵌入式 Linux 的内核结构与系统任务调度模型

Linux 嵌入式操作系统是一种新型的网络操作系统,在 Linux 嵌入式系统设计中,其内核由几个重要的子系统组成,分别是进程管理、内存管理、文件系统、设备管理、网络系统等。其中进程管理子系统最为核心,在进程管理子系统中完成进程的创建、中止,进程间的通信及任务调度。嵌入式 Linux 系统由启动引导程序(BootLoader)、Linux 内核、文件系统、应用程序组成,完成任务的调度和执行。

本文通过研究 Linux 系统的任务调度算法,发现它可以降低 CPU 在任务执行中的负载,从而减少系统功耗。算法研究之初,需要构建嵌入式 Linux 系统的任务调度总体模型,为了实现嵌入式系统的任务调度,结合动态变化的分布式计算系统信息来评估任务调度的效用,提高任务调度性能。设定嵌入式 Linux 中执行计算任务的多台工作站具有不同的硬件配置(如 CPU、内存等)和软件环境(如操作系统、设计软件等),系统中的每个簇由一个簇首、若干个协同簇首和分簇成员组成<sup>[8-10]</sup>,Linux 嵌入式任务调度协同设计平台中,设系统中工作站的集合为:

$$P = \{p_1, p_2, \dots, p_m\}, m \in N \quad (1)$$

对多个任务流中的计算任务进行在线调度,在 Linux 嵌入式系统中,一台工作站相当于一台处理机,此时待分配的任务流为:

$$flow_k = \{n_1, n_2, \dots, n_q\}, q \in N \quad (2)$$

式中, $q$  表示多个任务流集合的节点位置, $n_q$  表示任务信息流的数据序列, $N$  表示任务总数。将嵌入式系统客户端创建好的任务流提交给服务器,得到一个具有  $n$  个输入控制参量、 $m$  个输出参量的分簇任务调度模型,表达为:

$$\begin{cases} x = (x_1, x_2, \dots, x_n) \\ y = F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T \end{cases} \quad (3)$$

其中, $x = (x_1, x_2, \dots, x_n)$  为系统中工作站的集合; $y = F(x)$  表示节点活性因子;假设任务  $n_i$  的种类为  $r_j$ ,则集合  $P(n_i) =$

$\{p_k | pr_{kj} = 1, k = 1, 2, \dots, m\}$  表示任务流的优先级属性,定义往返时延(Round-Trip Time, RTT)为衡量主机之间网络状况的一个重要指标。在 Linux 系统中,通过任务调度将其提供给网络管理部分,进而得到操作系统中的每一个 TCP 连接维护  $RTT_s$  的值,原  $RTT_s$  任务进行调度的功耗表达为:

$$RTT_s = (1 - \alpha) \times RTT_s + \alpha \times RTT \quad (4)$$

由此得到了嵌入式 Linux 系统的任务调度模型,为进行任务优化调度算法的设计提供模型依据和总体框架。

### 2.2 预备知识分析与问题描述

分析上述构建的总体模型可知,当 Linux 内核的处理机执行任务时,如果任务分配不当,将严重影响任务执行性能指标,因此需要进行任务调度算法优化设计。传统方法采用的是基于模糊聚类单阈值均衡尺度分解的嵌入式 Linux 系统任务调度策略,任务调度的关键技术描述如下。

假设 Linux 进程管理处理机  $p_i$  执行任务  $n_j$  所能获得的直接回报  $DR(p_i, n_j)$  定义为:

$$DR(p_i, n_j) = rwd_k \times PET(p_i, n_j) \quad (5)$$

上式描述了 Linux 系统中的多源进程节点的任务调度执行模型,可见在 Linux 内核系统中的任务调度系统为一个离散系统, $PET(p_i, n_j)$  表示处理机  $p_i$  执行任务  $n_j$  的预计执行时间,采用模糊聚类策略对任务调度管理的数据信息流进行特征重组和分析,特征分解的微分方差表达式为:

$$\frac{\partial z(t)}{\partial t \partial (t-1)} = f(z) \quad (6)$$

若  $n_j$  的任务类型为  $r_k$ ,以此设定阈值作为任务调度的判决,传统方法基于单阈值均衡尺度分解的任务调度迭代输出为:

$$x_n = x(t_0 + n\Delta t) = h[z(t_0 + n\Delta t)] + \omega_n \quad (7)$$

式中, $h(\cdot)$  表示单阈值系统响应函数。上述算法给出了基于模糊聚类单阈值均衡尺度分解的 Linux 系统任务调度算法,分析上述算法可知,传统方法不适用于集成任务调度的情况,没能有效地对随机任务分配的多分簇节点实现能耗优化调度,在进行任务信息流特征尺度分解和均衡设计中存在任务调度效益失衡的问题,因此需要进行算法的改进设计,以提高 Linux 系统的嵌入式任务调度性能。

## 3 特征尺度均衡算法的提出和双阈值任务调度算法的改进实现

### 3.1 Linux 系统任务信息流特征尺度均衡算法

在上述构架的嵌入式 Linux 系统的任务调度模型中,对传统的基于模糊聚类单阈值均衡尺度分解的嵌入式 Linux 系统任务调度算法进行改进设计,提出一种基于嵌入式任务调度信息流特征尺度均衡的任务调度优化算法。首先给出 Linux 系统的任务信息流模型构建和特征提取算法,然后进行尺度均衡以提高任务调度性能。

设  $M$  是  $d$  维的任务调度散布系统, $F$  表示为一矢量,具有光滑性,对于  $\Phi: M \rightarrow R^{2d+1}$ ,有:

$$\Phi(x) = (h(x), h(\varphi_1(x)), \dots, h(\varphi_{2d}(x)))^T \quad (8)$$

为实现网络分簇能耗的双阈值分配,降低任务调度功耗,当  $\omega_n = 0$  时, Linux 系统的任务调度序列  $\{x_n\}_{n=1}^N$  以如下向量在调度任务多源节点中形成新的映射:

$$x_n = [x(0), x(1), \dots, x(N-1)]^T \quad (9)$$

在 Linux 系统的全局任务调度中心将上述所有任务进行融合,输入系统总调度器,提取任务信息流的特征尺度,特征尺度提取采用文献[3]中提出的基于模糊聚类的层次分析特

征提取算法, Linux 系统对以往各类型任务的到达频率、执行时间等信息进行统计得到特征尺度:

$$AT(v) = \max_{u \in FI(v)} [AT(u) + delay(v)] \quad (10)$$

式中,在时间段  $T$  内,类型为  $r_k$  的任务的到达次数是一个随机变量(记作  $X$ ),  $A$  代表尺度幅度值,  $delay(v)$  表示时延函数。Linux 嵌入式内核系统中管理进程节点在执行任务调度时,在时间段  $T$  内,类型为  $r_k$ 、执行时间为  $t$  的任务出现的次数  $X$  服从参数为  $\lambda$  的泊松分布,因为  $X$  服从泊松分布,概率  $P\{X=m\}$  包含了组合数为  $m$  的任务出现情况的概率总和,任务信息流的特征尺度实现了效益均衡,从而实现了特征尺度的均衡算法设计,下一步将以此为基础通过双阈值权衡判决实现对 Linux 系统的任务调度。

### 3.2 Linux 嵌入式任务调度算法优化设计的实现

根据上述设计的 Linux 嵌入式任务信息流的特征尺度均衡设计结果,以提取的任务信息流的特征尺度作为调度模型的输入向量,在任务执行时间段  $T$  内, Linux 嵌入式任务调度处理机输入的尺度均衡后的特征向量为集群任务调度集合  $S$ , 表示如下:

$$S = (U, A, V, f), P \cap Q = f, A = P \cup Q \quad (11)$$

式中,  $U$  表示时间段  $T$  内任务出现的次数,  $A$  表示特征尺度向量空间,  $V$  表示空闲时间片长度,  $f$  为集群任务调度集合  $S$  的随机加权向量,在限定执行任务总量下,若时间段  $T$  内任务出现了  $m$  次,则最后一个出现的任务执行完成的时间可能会超出  $T$ ,因此从时间点  $t_0$  开始,以  $T_d$  为单位,将时间轴划分成一个个相邻但不重合的窗口  $W_i$ ,示意图如图 1 所示。

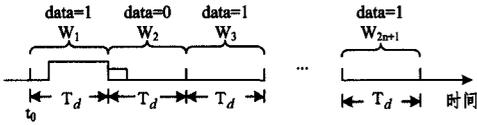


图 1 嵌入式任务调度时间窗口划分示意图

图 1 中,进行 Linux 任务调度的各空闲时间片长度分别记作  $x_1, x_2, \dots, x_{m+1}$ ,可以得到如下方程:

$$x_1 + x_2 + \dots + x_{m+1} = T + t - m \times t \quad (12)$$

在特征尺度均衡设计的基础上,采用双阈值判决条件设计,发送方需要发送数据时,时间轴划分成各个相邻的任务匹配平滑窗口模型,构建双阈值调度模型,假设每个任务节点  $v \in V$  都有一个延迟属性  $delay(v)$ ,得到的能耗分配的双阈值模型可以表示为:

$$AT(v) = \max_{u \in FI(v)} [AT(u) + delay(v)] \quad (13)$$

$$RT(v) = \min_{w \in RO(v)} [RT(w) - delay(v)] \quad (14)$$

最后根据 HList 返回到 IC Compiler 中进行任务调度,根据每个节点的时序信息,得到的整个 Linux 系统的任务节点双阈值调度的调度边界为:

$$bnr_{\beta}(X) = R_{\beta}X - R_{\beta}X_1 \quad (15)$$

由此得到的在特征尺度均衡条件下的 Linux 任务调度阈值判别式为:

$$\sum_{m=1}^n x_m^G \leq E_G, \sum_{m=1}^n x_m^T \leq E_T, \sum_{m=1}^n x_m^W \leq E_W, \sum_{m=1}^n x_m^L \leq E_L \quad (16)$$

通过双阈值判决,在不同的节点上能够提高整个任务的调度效率,效率优化总和为  $\sum_{\omega} \mu^{\omega} T_{\omega}^{\omega}$ ,其中  $\omega \subseteq (G, T, W, L)$ ,  $m \in [1, n]$ 。在系统应用时,平均距离值越小,代表算法收敛后解的性能越好,进而得到每个节点  $v \in V$  的时序裕度  $SL(v)$  表示 Spacing 指标。采用特征尺度均衡的双阈值任务调度,

Linux 系统的任务调度测试实现能耗的双阈值分配,用户给出了命令的路径,Shell 就沿着用户给出的路径进行查找,根据 Linux 嵌入式系统中的双阈值任务调度算法,建立 API 函数,使得软件能够访问硬件系统,又不会因为访问硬件系统而损坏操作系统的运行。通过本文改进算法的设计, Linux 中进行任务信息流特征尺度分解和均衡设计中存在的任务调度效益失衡的问题得到有效解决,最后通过仿真实验进行系统的性能验证。

## 4 仿真实验与结果分析

为了测试本文算法在 Linux 系统任务调度中的性能,需要进行一次仿真实验。仿真实验采用 NS-2.27 和 PDNS 软件进行数据模拟,采用随机数生成方法随机生成 1000 个任务调度节点。系统的软件设计以 Linux 2.6.32 内核为平台。实验过程中选择 3 台不同类型、不同配置的计算机作为多任务多处理器 Linux 内核任务调度平台,程序开发中,宿主机安装了 Windows 7 系统,要在 Windows 系统下使用 Linux 系统进行任务调度。在进行 Linux 嵌入式任务调度实验中,模拟 100 个系统请求任务,发出搜索等任务请求并通过日志对返回结果进行记录。时间段  $T$  内,  $m=3$  时的任务执行情况示例如图 2 所示。

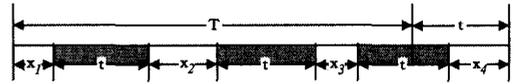


图 2 任务执行情况示意图

首先采用本文算法,构建系统任务调度模型,得到不同信噪比不同任务区节点下的 Linux 任务信息流,如图 3 所示。图 3 表示的是 3 个通道下的任务信息流,以此为信号模型原型,进行信息流的特征尺度均衡仿真,为实现 Linux 系统的双阈值任务调度提供数据基础。

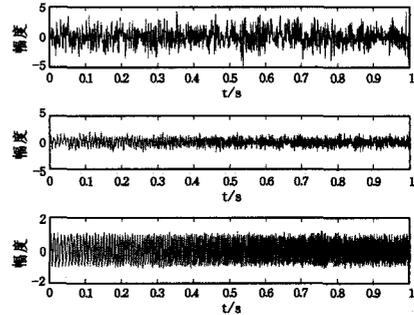


图 3 Linux 嵌入式任务信息流

以图 3 所示的 Linux 嵌入式任务信息流为信号原型,任务调度时长  $t$  取  $0 \sim 0.25$  s,采样频率为 2048 Hz,其中  $n(t)$  为高斯白噪声,现对任务信息流  $x(t)$  进行特征尺度分解,并采用特征尺度均衡算法,实现对三通道 Linux 任务信息流的特征分解均衡设计,得到特征尺度均衡结果,如图 4 所示。

从图 4 结果可见,采用本文算法能从任务信息流特征尺度分解谱中明显地看出噪声信号所分布的前 3 个 imf 分量,因此去掉前面 3 个 imf 分量,对剩余的 imf 分量进行重构,实现了对任务的高效调度,有效解决了任务信息流特征尺度分解和均衡设计中存在的任务调度效益失衡的问题,从而实现了 Linux 系统的双阈值任务调度。为对比算法性能,采用本文算法和传统算法在相同条件下进行 Linux 系统任务调度仿真,测试任务调度的 CPU 利用率,采用文献[3,4]中的模糊聚

类调度算法、免疫调度算法和本文算法进行性能对比,得到任务调度的 CPU 利用率如图 5 所示。从图 5 可见,采用本文算法进行 Linux 嵌入式系统的任务调度时, CPU 利用率最高,调度性能较好,随着任务执行循环轮次的增加, CPU 利用率可以达到 100%,与传统方法相比具有较大幅度的提高。

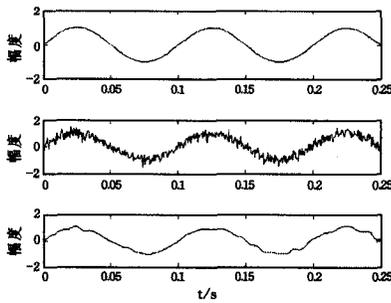


图 4 任务信息流特征尺度分解谱

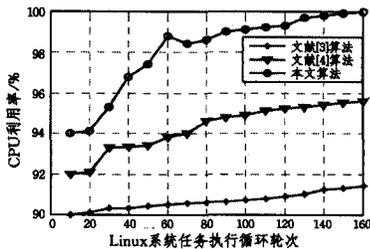


图 5 云平台任务调度 CPU 利用率对比

采用本文算法和传统方法进行性能对比,得到的结果如表 1 所列,从关键路径延时、相干功耗等指标进行任务调度分配的性能对比刻画,结果证明,采用本文算法具有较小的任务调度路径延时和能耗损失,系统功耗平均节约了 28%,展示了本文方法的优越性能。由计算任务调度时间的性能比较可知,传统方法用了 36ms 执行完成,而本文的方法仅需 8ms 即可执行完成相同的调度任务。

表 1 双阈值 Linux 系统任务调度算法的性能对比结果

	单阈值 Vt 方法	动态门限 算法	双阈值电压 分配算法
总逻辑门数	68956	68956	68956
low-Vt 单元数	63254	45637	53232
low-Vt 单元使用率/%	100	68.3	35.2
关键路径延时/ns	-0.202	-0.224	-0.223
相干功耗/nW	1.539	1.412	1.166
耗时/min	—	175.2	9.3

**结束语** 嵌入式 Linux 操作系统可以支持多个文件和任务执行,在嵌入式 Linux 操作系统的设计和应用中,操作系统经过移植后运行在不同的硬件平台上,进行进程管理、内存管理、文件系统和设备管理等,这一过程是一个复杂的多任务调度和集成处理过程,需要一种有效的任务调度算法实现进程管理和内存管理,以提高系统运行效率。因此研究一种高效的嵌入式 Linux 系统的任务调度算法,对提高操作系统的性能具有重要意义。本文提出了一种基于特征尺度均衡的 Linux 系统双阈值任务调度算法,进行了嵌入式 Linux 的内核结构分析与系统任务调度模型构建,进行了算法改进设计。算法模型研究和实验结果分析表明,采用本文算法能有效解决任务信息流特征尺度分解和均衡设计中存在的任务调度效益失衡的问题,提高 Linux 系统任务调度效率和性能,提高 Linux 系统的 CPU 利用率,节省任务执行时间和能耗等,其性能优于传统方案。

## 参考文献

- [1] 张松慧,熊锦江.一种针对非平稳网络任务调度防冲突算法研究[J].科技通报,2013,29(10):143-145  
Zhang Song-hui, Xiong Jin-jiang. Non-Stationary Network Scheduling Anti-Collision Algorithm Research [J]. Bulletin of Science and Technology, 2013, 29(10): 143-145
- [2] Miorandi D, Sicari S, Pellegrini F D, et al. Internet of things: vision, applications and research challenges [J]. Ad Hoc Networks, 2012, 10(7): 1497-1516
- [3] 李静梅,王雪,吴艳霞.一种改进的优先级列表任务调度算法[J].计算机科学,2014,41(5):20-23  
Li Jing-mei, Wang Xue, Wu Yan-xia. Improved Priority List Task Scheduling Algorithm [J]. Computer Science, 2014, 41(5): 20-23
- [4] Chong S K, Gaber M M, Krishnaswamy S, et al. Energy conservation in wireless sensor networks: a rule-based approach [J]. Knowledge and Information Systems, 2011, 28(3): 579-614
- [5] 许丞,刘洪,谭良. Hadoop 云平台的一种新的任务调度和监控机制[J]. 计算机科学, 2013, 40(1): 112-117  
Xu Cheng, Liu Hong, Tan Liang. New Mechanism of Monitoring on Hadoop Cloud Platform [J]. Computer Science, 2013, 40(1): 112-117
- [6] 李仁发,刘彦,徐成.多处理器片上系统任务调度研究进展评述[J].计算机研究与发展,2008,45(9):1620-1629  
Li Ren-fa, Liu Yan, Xu Cheng. A Survey of Task Scheduling Research Progress on Multiprocessor System-on-Chip [J]. Computer Research and Development, 2008, 45(9): 1620-1629
- [7] 刘少伟,孔令梅,任开军,等.云环境下优化科学 workflows 执行性能的两阶段数据放置与任务调度策略[J].计算机学报,2011,34(11):2021-2130  
Liu Shao-wei, Kong Ling-mei, Ren Kai-jun, et al. A Two-Step Data Placement and Task Scheduling Strategy for Optimizing Scientific Workflow Performance on Cloud Computing Platform [J]. Chinese Journal of Computers, 2011, 34(11): 2021-2130
- [8] 孟宪福,解文利.基于免疫算法多目标约束 P2P 任务调度策略研究[J].电子学报,2011,39(1):101-107  
Men Xian-fu, Xie Wen-li. Research on P2P Task Scheduling with Multi-objective Constraints Based on Immune Algorithm [J]. Acta Electronica Sinica, 2011, 39(1): 101-107
- [9] 李文娟,张启飞,平玲娣,等.基于模糊聚类的云任务调度算法[J].通信学报,2012,33(3):146-154  
Li Wen-juan, Zhang Qi-fei, Ping Ling-di, et al. Cloud Scheduling Algorithm Based on Fuzzy Clustering [J]. Journal on Communications, 2012, 33(3): 146-154
- [10] 周浩,高远,朱昌平.基于双门限能量检测的选择式协作频谱感知[J].计算机仿真,2014,31(1):199-203  
Zhou Hao, Gao Yuan, Zhu Chang-ping. Alternative Cooperative Spectrum Sensing Based on Double Threshold Energy Detection [J]. Computer Simulation, 2014, 31(1): 199-203
- [11] 史少锋,刘宴兵.基于动态规划的云计算任务调度研究[J].重庆邮电大学:自然科学版,2012,24(6):687-692  
Shi Shao-feng, Liu Yan-bing. Cloud Computing Task Scheduling Research Based on Dynamic Programming [J]. Journal of Chongqing University of Posts and Telecommunications: Natural Science Edition, 2012, 24(6): 687-692