

# 异构云中综合时间能耗成本的任务调度算法

李 君,殷小龙,万明祥

(南京邮电大学 计算机学院,江苏 南京 210003)

**摘 要:**针对异构云环境下不合理的任务-资源映射而导致依赖任务在运行过程中产生高能耗的问题,提出一种综合时间能耗成本的任务调度算法(Time and Energy Consumption Cost Scheduling,TECCS)。根据任务图逐层进行任务调度,面对同一层任务调度顺序只单独基于时间因素考虑而过于单一的问题,引入通信因子和计算因子,综合时间与能耗成本决定同一层任务的调度顺序;分析任务之间的依赖关系,自上而下,为任务分配计算节点,使得整个任务在期望完成时间条件下节省更多能耗。从性能和能耗方面将 TECCS 与 TUGS(Time Unify Greed Scheduling)、CATS(Communication-Aware Task Scheduling)、EETDS(Energy Efficient Task Duplication Scheduling)进行比较,结果表明 TECCS 在满足任务期望完成时间条件下能耗最少。

**关键词:**异构系统;云数据中心;依赖任务;层次;节能

**中图分类号:**TP393

**文献标识码:**A

**文章编号:**1673-629X(2014)08-0121-05

**doi:**10.3969/j.issn.1673-629X.2014.08.028

## Task Scheduling Algorithm Based on Time and Energy Consumption Cost in Heterogeneous Cloud

LI Jun, YIN Xiao-long, WAN Ming-xiang

(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

**Abstract:**Facing the problem of high energy consumption produced by unreasonable task-resource mapping in heterogeneous cloud, propose a task scheduling algorithm based on time and energy consumption (TECCS). Dividing the hierarchical of the tasks to determine the order of tasks, and facing the problem of the scheduling sequence of the same layer tasks based solely on time factors into consideration and too one-dimensional, introduce communication factor and computation factor, integrating time and energy consumption cost to determine the order of the same level tasks. Analysis of dependencies between tasks, based on hierarchical top-down to task scheduling, make the whole task completion time under the expected conditions to save more energy. Comparison on performance and energy consumption is conducted for TECCS with TUGS(Time Unify Greed Scheduling), CATS(Communication-Aware Task Scheduling), EETDS(Energy Efficient Task Duplication Scheduling). Experimental results show that TECCS saves much energy under the condition of task desired time.

**Key words:**heterogeneous system; cloud data center; dependent tasks; hierarchical; energy saving

## 0 引 言

随着云计算正在引领信息产业的新浪潮,云数据中心的能量消耗逐渐成为制约云计算发展因素之一。据统计,Google的云计算数据中心每年消耗的电能可为1亿千瓦,这相当于一个小型城市的总能耗。由IDC提供的报告显示,在最近的30年之中,由大规模数据中心所带来的能源消耗已经增长了400%,并且这一数字正快速地持续增长。目前,在全球70%的计算中心中,能耗开销已成为第二大运营开销<sup>[1-5]</sup>。为避免

云数据中心的高能耗问题成为制约云计算发展的瓶颈,高能耗问题亟须解决<sup>[6-7]</sup>。

云数据中心的硬件资源通常由大规模的异构计算节点组成,各计算节点之间通过具有不同传输率的链路连接而成,构成一个大规模异构云计算环境。任务调度是云计算的关键技术,是提高系统性能的重要手段<sup>[8-13]</sup>。云计算系统中的一些任务之间往往具有某种依赖关系,在异构云数据中心,同一个任务在不同计算节点的执行时间以及任务间的通信在不同传输路径

收稿日期:2013-10-18

修回日期:2014-01-20

网络出版时间:2014-05-21

基金项目:江苏省研究生科研创新计划项目(CXLX12\_0481)

作者简介:李 君(1988-),女,硕士研究生,研究方向为绿色计算、任务调度。

网络出版地址:<http://www.cnki.net/kcms/detail/61.1450.TP.20140525.1242.008.html>

中的传输时间均有所差异,在任务调度的过程中,不但要保证任务之间的依赖性不被破坏,还要保证整个任务在期望完成时间的条件下,能耗开销最小,故异构云数据中心中依赖任务的节能调度首要问题便是如何确定任务的调度顺序,其次是如何给任务分配计算节点。

基于不同环境的假设,云计算中节能的依赖任务调度算法可分为基于同构和基于异构环境两种。文献[8-9]基于列表调度技术对依赖任务进行节能调度,其方法均是确定 DAG 任务图的关键路径任务,调节非关键路径上的任务进行节能调度,然而,其都是基于同构环境的假设,对于组成云数据中心的异构系统而言,任务的执行时间和任务间的通信时间都是不确定的,随着调度而改变。文献[10]面向高性能计算和绿色计算,给出并行任务在异构处理机上时间与能耗的启发式优化执行算法。但文献中假设任务执行时间与能耗之间存在线性关系,且算法性能受能耗时间归一因子的影响较大。文献[11]设计了异构分布式系统中能耗感知的任务调度算法,分析相互约束的并行应用程序,基于 DVFS 技术,利用进程间通信造成的 CPU 空闲时间槽进行电压调节从而减少能耗。文献[12]对通信感知的节能任务调度问题进行研究,任务的最早开始时间和任务最晚完成时间之和决定任务的优先级,最小优先级最先调度,并通过通信感知算法进行认证,目的是尽量减少计算节点间的通信量,从而减少节点间的通信能耗,当计算节点有松弛时间,通过电压调节减少任务的执行能耗开销。文献[13]面向异构系统中依赖任务的节能调度,将任务分组,在为任务分配计算节点时,通过任务复制减少任务间通信能耗,从而节省整个任务执行能耗。

文献[10-13]均是基于时间因素单独确定任务调度顺序,确定任务优先级的方法过于单一。首先,一个任务的调度顺序是由该任务的计算量及其前驱任务之间的通信量决定,对于依赖任务和异构云数据中心而言,任务的计算量和任务间通信量存在权重比较,不可直接相加而确定任务调度顺序;其次,对于节能调度,时间并不是唯一反映该任务优先级的属性,任务的调度顺序应由计算节点执行任务需要的执行成本(时间、能耗)和任务间通信需要的通信成本(时间、能耗)共同决定。

文中面对异构云数据中心中依赖任务的节能调度问题,提出了一种基于时间能耗成本的任务调度算法,在性能和能耗之间达到平衡。主要贡献:

(1)提出一种云计算节能任务调度模型;

(2)提出一种综合时间与能耗成本的方法确定同层中任务的调度顺序;

(3)通过分析任务之间的依赖关系,自上而下,为

任务分配计算节点,并结合 DVFS 技术进行电压调节;

(4)在任务期望完成时间条件下,节省更多能耗,实现了性能和能耗之间的平衡。

## 1 问题建模

由  $m$  个资源和  $n$  个任务组成的  $M * N$  云计算节能任务调度模型中,云计算任务调度模型可以描述为五元组,即  $M = (P, T, EN, \Theta, \vartheta)$ 。其中,  $P$  表示由  $m$  个资源所组成的资源集合;  $T$  为  $n$  个具有依赖关系的任务组成的任务集合;  $EN$  表示任务调度过程中的能耗开销;  $\Theta$  表示节能调度优化算法;  $\vartheta$  表示云节能调度系统的调度优化目标函数。其具体特征描述如下:

云资源集合  $P = \{p_1, p_2, \dots, p_m\}$  由  $m$  个异构计算节点组成,  $p_j = \{sv_j, env_j, vb_j, entr_j\}$  包含 4 个不同维度的特征属性,其中,  $sv_j$  表示  $p_j$  的计算能力,文中假设每个资源均支持 DVFS 技术,有不同的运行速度,因此资源  $p_j$  的计算能力  $sv_j$  进一步可以刻画为  $sv_j = \{s_{jk}, k \in 1, 2, \dots, q\}$ ,其中  $k$  表示节点执行速度的级别,  $s_{jmin} = s_{j1} < s_{j2} < \dots < s_{jq} = s_{jmax}$ ,在为任务选择合适的计算节点的过程中,初始一律选择计算节点的最大速度,即  $s_{jmax}$ ;  $env_j$  表示  $p_j$  的执行速度的能量消耗率,同样,在不同执行速度级别下的能量消耗率进一步刻画为:  $env_j = \{ev_{jk}, k \in 1, 2, \dots, q\}$ ,其中  $k$  表示节点执行速度能量消耗率的级别,  $ev_{jmin} = ev_{j1} < ev_{j2} < \dots < ev_{jq} = ev_{jmax}$ ;  $vb_j$  表示资源  $p_j$  与其他资源的数据传输率,  $vb_j = \{b_{j1}, \dots, b_{jk}, \dots, b_{jm}\}$ ,其中  $b_{jk}$  表示资源  $p_j$  与  $p_k$  之间的数据传输率;  $entr_j$  表示资源  $p_j$  与其他资源的数据传输能量消耗率,  $entr_j = \{t_{j1}, \dots, t_{jk}, \dots, t_{jm}\}$ ,其中  $t_{jk}$  表示资源  $p_j$  与  $p_k$  之间的数据传输能量消耗率。

云任务集合由  $n$  个具有依赖关系的任务用有向无环图(Directed Acyclic Graph, DAG)  $T = (V, E)$  表示。顶点集合  $V$  表示任务,  $V = \{v_i \mid 1 \leq i \leq n\}$ ; 边集合  $E$  表示任务之间的依赖关系,  $E = \{e_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq n\}$ ,其中  $e_{ij}$  表示任务  $v_i$  传递给任务  $v_j$  的数据量。

$v_i = \{w_i, pred(v_i), succ(v_i), EST(v_i), ECT(v_i)\}$  包含 5 个不同维度的特征属性。其中,  $w_i$  表示任务  $v_i$  的计算量;  $pred(v_i)$  表示任务  $v_i$  的直接前驱任务集合,  $pred(v_i) = \{v_j \mid e_{ij} \in E\}$ ,没有前驱任务的任务记为入口任务;  $succ(v_i)$  表示任务  $v_i$  直接后继任务集合,  $succ(v_i) = \{v_j \mid e_{ij} \in E\}$ ,没有后继任务的任务记为出口任务;  $EST(v_i)$  为任务  $v_i$  的最早开始时间,入口任务的最早开始时间为 0;  $ECT(v_i)$  为任务  $v_i$  的最早完成时间。

云计算节能任务调度系统的能耗开销主要考虑计算能耗和通信能耗<sup>[11]</sup>,  $ec^{exec}(T, P)$  表示执行所有任务的计算能耗开销,  $ec^{comm}(T, P)$  表示执行所有任务的传

输能耗开销。云计算节能任务调度系统分配矩阵  $\mathbf{X} = \{x_{iq} \mid 1 \leq i \leq n, 1 \leq q \leq m\}$ ,  $x_{iq} = 1$  表示任务  $v_i$  分配到资源节点  $p_q$  上执行;  $x_{iq} = 0$  表示任务  $v_i$  不在  $p_q$  上执行。

$ec^{exec}(T, P)$  的计算公式如式(1):

$$ec^{exec}(T, P) = \sum_{j=1}^m \sum_{i=1}^n x_{ij} \cdot \frac{w_i}{s_{jk}} \cdot ev_{jk} \quad (1)$$

其中,  $s_{jk}$  表示计算节点  $p_j$  执行任务的处理速度;  $ev_{jk}$  为其能量消耗率。

$ec^{tran}(T, P)$  的计算公式如式(2):

$$ec^{tran}(T, P) = \sum_{s=1}^m \sum_{r=1}^m \sum_{j=1}^n \sum_{i=1}^n x_{ir} \cdot x_{js} \cdot \frac{e_{ij}}{b_{rs}} \cdot t_{rs} \quad (2)$$

根据式(1)和式(2)可知,总能耗的计算公式如式(3):

$$EN = ec^{exec}(T, P) + ec^{tran}(T, P) \quad (3)$$

记任务执行的用户期望时间为 FT, 整个依赖任务调度的时间用 Lspan 表示, 计算公式如式(4):

$$Lspan = \max\{ECT(v_i)\} - \min\{EST(v_i)\} \quad (4)$$

为了保证整个任务调度过程中能耗开销最小, 需要满足目标式(5), 为了保证任务在期望时间内完成, 需要满足约束条件式(6):

$$\text{minimize } EN = ec^{exec}(T, P) + ec^{tran}(T, P) \quad (5)$$

$$\text{subject to } Lspan - FT \leq 0 \quad (6)$$

## 2 一种综合时间能耗成本的任务调度算法

一种综合时间能耗成本的调度过程主要包括确定任务的调度顺序以及为任务分配计算节点。

### 2.1 任务调度顺序的确定

调度的顺序必须保证任务依赖性不被破坏, 文中对任务进行层次划分以确定任务调度次序, 针对同一层中任务调度次序只单独考虑时间因素而过于单一的问题, 引入计算因子和通信因子, 综合时间与能耗成本的方法确定调度次序。

定义1: 任务的层次为其直接前驱任务中最大层次加1, 用  $level(v_i)$  表示。表达式如下:

$$\begin{cases} 0, & \text{if } \text{pred}(v_i) = \emptyset \\ \max(level(\text{pred}(v_i))) + 1, & \text{if } \text{pred}(v_i) \neq \emptyset \end{cases} \quad (7)$$

定义2: 计算因子  $\delta$  为调度过程中任务计算量权重。

定义3: 通信因子  $\vartheta$  为调度过程中任务间通信量的权重。

同一层中的不同任务, 按照优先级从大到小的次序依次分配计算节点, 同层中任务  $v_i$  优先级  $PL(v_i)$  按下式确定:

$$PL(v_i) = \vartheta \cdot \max\{e_{ji}\} + \delta \cdot w_i \quad (8)$$

其中,  $\max\{e_{ji}\}$  为任务  $v_i$  与其各直接前驱任务之间的通信量的最大值;  $w_i$  为任务  $v_i$  的计算量。

计算因子  $\delta$  和通信因子  $\vartheta$  反映整个任务调度过程中任务计算量和任务间通信量的重要程度, 即  $\delta + \vartheta = 1$ 。欧几里得空间常用于确定权重系数, 文中采用欧几里得距离确定  $\vartheta$  和  $\delta$ , 具体方法如下:

记  $T_{ran}$  为整个 DAG 任务节点的平均通信时间,  $T_{exc}$  为平均计算时间,  $E_{exc}$  为整个任务执行的平均计算能耗,  $E_{tran}$  为平均通信能耗; 记  $\alpha_1$  和  $\alpha_2$  分别为整个有向无环图任务节点的平均通信时间和平均计算时间占整个任务执行时间的比例; 记  $\beta_1$  和  $\beta_2$  分别为整个有向无环图任务节点的平均通信能耗和平均计算能耗占整体能耗开销的比例; 则有:

$$\alpha_1 = \frac{T_{ran}}{T_{ran} + T_{exc}} \quad (9)$$

$$\alpha_2 = \frac{T_{exc}}{T_{ran} + T_{exc}} \quad (10)$$

$$\beta_1 = \frac{E_{tran}}{E_{tran} + E_{exc}} \quad (11)$$

$$\beta_2 = \frac{E_{exc}}{E_{tran} + E_{exc}} \quad (12)$$

记点  $A(\alpha_1, \beta_1)$ ,  $B(\alpha_2, \beta_2)$ , 综合时间与能耗成本确定计算因子和通信因子, 分别计算  $A$  与  $B$  大小, 如式(13)、(14):

$$d_A = \sqrt{\alpha_1^2 + \beta_1^2} \quad (13)$$

$$d_B = \sqrt{\alpha_2^2 + \beta_2^2} \quad (14)$$

$$\text{则有 } \delta = \frac{d_B}{d_A + d_B}, \vartheta = \frac{d_A}{d_A + d_B}。$$

### 2.2 任务计算节点的选取

针对 DAG 任务图中某一个任务  $v_i$ , 根据其直接前驱任务的个数分别讨论, 为其选择计算节点:

1) 当前任务  $v_i$  只有一个直接前驱任务  $v_j$ , 从所有可用计算节点中选择使当前任务  $v_i$  的最早开始时间  $EST(v_i)$  最小的计算节点作为当前任务  $v_i$  的计算节点; 将当前任务  $v_i$  分配给任意一个计算节点  $p_k$  时的最早开始时间  $EST(v_i)$  按照式(15)确定:

$$EST(v_i) = \max\{ECT(v_j) + tt(v_j, p_m; v_i, p_k), M_{wait}(p_k)\} \quad (15)$$

式中,  $ECT(v_j)$  为当前任务  $v_i$  的直接前驱任务  $v_j$  的最早完成时间;  $p_m$  为任务  $v_j$  所分配的计算节点;  $M_{wait}(p_k)$  表示计算节点  $p_k$  中等待任务队列中所有任务的最早完成时间的最大值;  $tt(v_j, p_m; v_i, p_k)$  为任务  $v_j$  和  $v_i$  之间的通信量  $e_{ji}$  在计算节点  $p_m$  和  $p_k$  之间传输所需的时间, 则  $tt(v_j, p_m; v_i, p_k) = \frac{e_{ji}}{b_{mk}}$ 。

任务  $v_i$  在计算节点  $p_k$  中的最早完成时间  $ECT(v_i)$



按照下式确定:

$$ECT(v_i) = EST(v_i) + \frac{w_i}{s_{kmax}} \quad (16)$$

2) 当前任务  $v_i$  有至少两个直接前驱任务, 则采取前驱任务节点预分配原则为任务  $v_i$  分配计算节点。任务  $v_i$  的直接前驱任务分配好计算节点后, 这些前驱任务分别为其后继任务  $v_i$  分配一个预计算节点, 则在这些预计算节点中选择一个计算节点作为  $v_i$  的计算节点。

若  $v_i$  的直接前驱任务有 2 个, 记为  $v_s$  和  $v_u$ 。当对  $v_s$  分配计算节点后, 给其直接后继任务  $v_i$  分配一个计算节点, 从所有可用计算节点中选择使其直接后继任务  $v_i$  的最早开始时间  $EST_1(v_i)$  最小的计算节点作为任务  $v_i$  的预分配计算节点; 将任务  $v_i$  分配给任意一个计算节点  $p_k$  时的最早开始时间  $EST_1(v_i)$  按照下式确定:

$$EST_1(v_i) = \max\{ECT(v_j) + tt(v_u, p_m; v_i, p_k), Mwait(p_k)\} \quad (17)$$

同理,  $v_u$  分配结束后从所有可用计算节点中选择使其直接后继任务  $v_i$  的最早开始时间  $EST_2(v_i)$  最小的计算节点作为任务  $v_i$  的预分配计算节点, 将任务  $v_i$  分配给任意一个计算节点  $p_k$  时的最早开始时间  $EST_2(v_i)$  按照下式确定:

$$EST_2(v_i) = \max\{ECT(v_j) + tt(v_u, p_m; v_i, p_k), Mwait(p_k)\} \quad (18)$$

任务  $v_i$  的执行节点是其直接前驱任务给其分配的使其具有最大最早开始时间节点, 如式 (19):

$$EST(v_i) = \max\{EST_1(v_i), EST_2(v_i)\} \quad (19)$$

### 2.3 节点中待执行任务的电压调节

当确定任务  $v_i$  的计算节点为  $p_k$  后, 计算该计算节点的空闲时间  $slacktime(p_k)$ , 令

$$slacktime(p_k) = EST(v_i) - \max(wait(k, ECT(v_n))) \quad (20)$$

当  $slacktime(p_k) > 0$ , 表明在计算节点  $p_k$  中等待任务队列中所有任务完成之后, 任务  $v_i$  还没有到达计算节点  $p_k$ , 此时存在空闲时间, 基于 DVFS 技术, 调节计算节点  $p_k$  中等待执行任务的电压。

## 3 实验及结果分析

将文中提出的算法与文献 [10] 中的时间归一贪婪算法 (TUGS)、文献 [12] 中的通信感知的节能调度算法 (CATS) 以及文献 [13] 中基于任务复制的节能调度算法 (EETDS) 进行比较。由于文中目的是通过任务调度实现节能, 任务完成时间是评价一个任务调度算法的性能标准, 整个任务调度过程中系统产生的总

能耗能直观表现调度算法节能程度, 故评价指标如下:

(1) 任务完成时间: 为具有最晚完成时间的任务与最早开始时间任务之差;

(2) 系统总能耗: 任务调度中系统产生的能耗。

### 3.1 实验环境及参数设置

仿真实验使用 SimGrid 模拟异构云数据中心, 设置 3 种类型的计算节点分别为  $Node_1$ 、 $Node_2$ 、 $Node_3$ , 计算节点速度和计算节点能量消耗率服从均匀分布。各节点计算速度分别为 200–500 kbps、400–800 kbps、700–1 000 kbps; 能量消耗率为 1–2 kJ/s、2–4 kJ/s、4–6 kJ/s; 通信链路传输为 NET1, 传输率范围为 2.5–10 Mbit/s, 链路传输能量消耗率范围为 5–10 kJ/s; 能量消耗率与通信链路传输率服从均匀分布。异构云数据中心的计算节点个数  $m$  选为 12, 每组实验中随机选取计算节点和通信链路体现云数据中心的异构性。

### 3.2 不同任务个数对算法性能的影响

本组实验测试任务大小对算法性能的影响。在  $CCR=2.0$ , 任务数目  $n=\{50, 100, 300, 500, 700, 1\ 000\}$  的条件下进行实验, 通过获得的任务完成时间、系统总能耗开销验证 TECCS 在不同任务数下的适应性。实验获得的任务完成时间如图 1 所示。

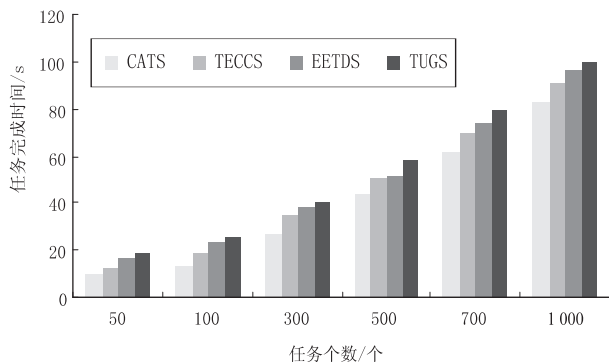


图 1 不同任务个数时各算法的任务完成时间

由图 1 可知, TECCS 任务完成时间比 CATS 大 8.75%, 比 TUGS 小 7.4%, 比 EETDS 小 6.1%; 算法 CATS 的任务完成时间优于算法 TECCS, 因为在为任务分配调度顺序的时候, CATS 仅仅从时间角度考虑优先分配当前最需要调度的任务。

由图 2 可知, TECCS 系统总能耗开销比 CATS 小 15.81%, 比 TUGS 小 8.2%, 比 EETDS 小 4.2%; TECCS 能耗开销优于算法 CATS、TUGS 和 EETDS。

### 3.3 任务类型 (CCR) 对算法性能的影响

本组实验验证不同任务类型对算法性能的影响。CCR<sup>[14]</sup> 表示整个任务的平均通信时间与平均计算时间的比率, 反应任务类型。取  $n=200, m=12, CCR=\{0.25, 0.5, 0.75, 1, 2, 2.5\}$ 。通过实验获得任务完成时间和系统总能耗开销验证算法的性能。获得的任务完成时间如图 3 所示。

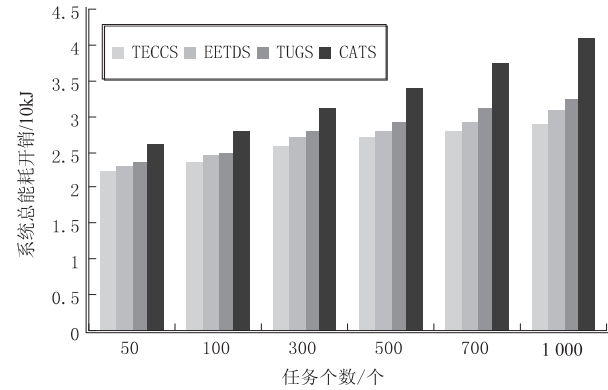


图2 不同任务个数时各算法的总能耗开销

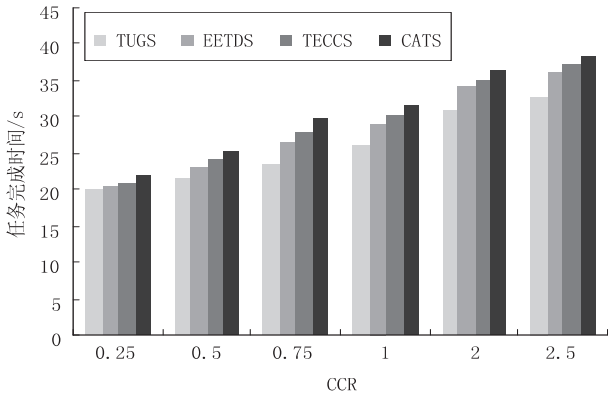


图3 不同 CCR 时各算法的任务完成时间

由图3可知,TECCS的平均任务完成时间比CATS小3.8%,比EETDS大4.11%,比TUGS大7.3%;随着CCR值的增加,任务间的通信增加,各算法的任务完成时间均增加。在CCR值很小,为0.25时,由于任务间的通信较少,算法TECCS的任务完成时间略大于算法CATS。

由图4可知,算法TECCS的平均系统总能耗开销比TUGS小17.9%,比EETDS小11.2%,比CATS小9.1%;随着CCR的增加,算法TECCS的系统总能耗开销明显优于CATS。

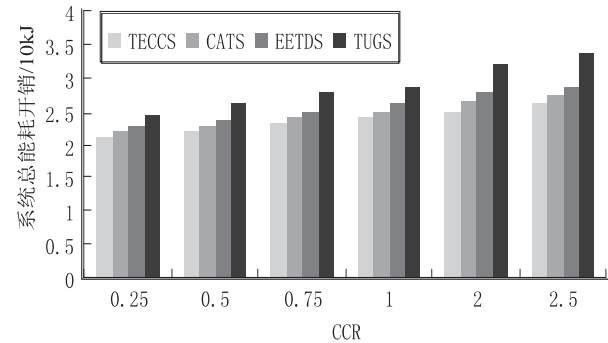


图4 不同 CCR 时各算法的总能耗开销

4 结束语

文中研究了异构云环境下依赖任务的节能调度问题,仿真结果表明,TECCS 调度策略在任务完成时间和能量消耗上能有效地达到平衡。

参考文献:

[1] Jifang. 云计算与能源管理[EB/OL]. 2011. <http://www.jifang360.com/news/2011112.html>.

[2] 刘 鹏. 云计算[M]. 北京:电子工业出版社,2011.

[3] 周洪波. 云计算技术、应用、标准和商业模式[M]. 北京:电子工业出版社,2011.

[4] 云计算[EB/OL]. 2013. <http://baike.baidu.com/view/1316082.htm>.

[5] Chinadaily. 打造绿色节能的云计算数据中心[EB/OL]. 2011. [http://www.chinadaily.com.cn/microreading/tech/2011.5.10/content\\_2561229.html](http://www.chinadaily.com.cn/microreading/tech/2011.5.10/content_2561229.html).

[6] 张 法, Anta A F, 王 林, 等. 网络能耗系统模型及能效算法[J]. 计算机学报, 2012, 35(3): 603-615.

[7] 林 闯, 田 源, 姚 敏. 绿色网络和绿色评价: 节能机制、模型和评价[J]. 计算机学报, 2011, 34(4): 593-612.

[8] Zhang Y M, Hu X B. Task scheduling and voltage selection for energy-minimization[C]//Proceedings of DAC. New Orleans, LA; [s. n.], 2010.

[9] Kang J, Ranka S. DVS based energy minimization algorithm for parallel machines[C]//Proc of IEEE international symposium on parallel and distributed processing. Miami, FL: IEEE, 2008: 1-12.

[10] Huang Qingjia, Su Sen, Li Jian, et al. Enhanced energy-efficient scheduling for parallel applications in cloud[C]//Proceedings of the 2012 12th IEEE/ACM international symposium on cluster, cloud and grid computing. Ottawa, ON: IEEE, 2012: 781-786.

[11] Liu Yongpeng, Zhu Hong, Lu Kai, et al. A power provision and capping architecture for large scale systems[C]//Proc of 2012 IEEE 26th international parallel and distributed processing symposium workshops & PhD forum. Shanghai: IEEE, 2012: 954-963.

[12] Varatkar G, Marculescu R. Communication-aware task scheduling and voltage selection for total systems energy minimization[C]//Proceedings of the 2003 international conference on computer aided design. [s. l.]: IEEE, 2003: 510-517.

[13] 张建军, 李庆华, 瞿 勇. 基于任务复制的调度算法[J]. 计算机工程与设计, 2009, 30(8): 1896-1899.

[14] 施步青. 基于能量优化的网络资源调度算法研究[D]. 武汉: 武汉理工大学, 2009.

# 异构云中综合时间能耗成本的任务调度算法

作者：[李君](#)，[殷小龙](#)，[万明祥](#)，[LI Jun](#)，[YIN Xiao-long](#)，[WAN Ming-xiang](#)

作者单位：[南京邮电大学 计算机学院](#)，[江苏 南京](#)，[210003](#)

刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：[2014\(8\)](#)

本文链接：[http://d.wanfangdata.com.cn/Periodical\\_wjfz201408028.aspx](http://d.wanfangdata.com.cn/Periodical_wjfz201408028.aspx)