异构计算系统中弹性节能调度策略研究

朱晓敏 贺 川 王建江 江建清
(国防科学技术大学信息系统工程重点实验室 长沙 410073)

摘 要 目前,节能已成为异构计算系统中减少电量开销、提高系统可靠性和保护环境的重要研究内容.传统的节能调度策略侧重于研究如何节能而忽略了用户对任务完成时间的期望,使得任务执行效果受到较大影响.特别是当系统负载较重时,由于电压调节缺乏自适应性,导致在某些情况下(如应急服务)的任务执行效果不可容忍.文中提出一种弹性节能调度策略(Elastic Energy-Aware Scheduling,EEAS),用于动态调度异构计算系统中非周期、独立任务.EEAS策略根据系统负载情况在系统节能与用户期望之间进行权衡,即当系统负载较重时,EEAS优先考虑用户期望,通过动态调整计算节点局部队列中等待任务的执行电压提高任务完成率;当系统负载较轻时,EEAS在尽量满足用户期望的基础上最大限度地降低任务执行电压以实现节能.文中通过大量的模拟实验比较了EEAS、GEA、HVEA和LVEA的性能.实验结果表明,EEAS的调度质量优于其他策略,可有效提高系统弹性.

关键词 异构计算系统;调度;节能;弹性;动态电压调整
中图法分类号 TP393 **DOI号**: 10.3724/SP.J.1016.2012.01313

An Elastic Energy-Aware Scheduling Strategy for Heterogeneous Computing Systems

ZHU Xiao-Min HE Chuan WANG Jian-Jiang JIANG Jian-Qing
(*Key Laboratory of Information System Engineering, National University of Defense Technology, Changsha* 410073)

Abstract Energy saving has become a major issue for heterogeneous computing systems to minimize electricity cost, improve system reliability and protect environment. Conventional energy-aware scheduling strategies developed on heterogeneous computing systems concentrated on energy savings regardless of the user expected finish times of tasks while making scheduling decisions. As a result, the user expectations by such strategies will be affected greatly especially when the systems are heavily loaded, which results in inferior system adaptivity or, in some situations, (e.g., emergency service) it is even not tolerated. In this paper, we developed a novel dynamic scheduling strategy named Elastic Energy-Aware Scheduling (EEAS) for aperiodic, and independent tasks on heterogeneous computing systems with dynamic voltage scaling. The EEAS strategy aims at adaptively adjusting voltages according to the system workload, thereby making trade-offs between energy conservation and user expectation. i.e., when the system is under heavy workload, to meet user expectations, EEAS not only considers the voltage for a new task, but also takes the voltages to run tasks waiting in local queues into account; in contrast, EEAS degrades voltage levels to reduce energy consumption while holding higher user satisfaction rate in terms of user expected finish time. We conducted extensive experiments to compare our EEAS with three schemes—GEA, HVEA and LVEA. Experimental results show that EEAS significantly improves the scheduling quality of others, and is able to effectively enhance the system elasticity.

Keywords heterogeneous computing system; scheduling; energy-aware; elastic; dynamic voltage scaling

收稿日期:2011-07-06;最终修改稿收到日期:2012-02-13.本课题得到国家自然科学基金(61104180)资助.**朱晓敏**,男,1979年生,博士,讲师,中国计算机学会(CCF)会员,主要研究方向为系统优化与调度、实时系统.E-mail:xmzhu@nudt.edu.cn.**贺 川**,男,1985年生,博士研究生,主要研究方向为系统优化与调度.**王建江**,男,1986年生,博士研究生,主要研究方向为军事运筹、作战模型与模拟.**江建清**,男,1987年生,硕士研究生,主要研究方向为分布式调度.

1 引 言

随着计算机硬件设备的快速发展,大规模计算系统在成本和体积迅速下降的同时计算能力大幅提升,目前已被广泛用于计算密集型和数据密集型应用[1-2],尤其是由不同计算能力的节点组成的异构计算系统在实际工作中得到了广泛应用.这是因为一方面同一系统中的计算节点可能在不同时期获得,在计算能力上有所差异,同时出于经济效益考虑,那些仍然具有一定计算能力的节点会和新购买的计算节点放在一起;另一方面,系统中通信链路的传输能力可能有所不同,从而构成了一个异构计算系统[3].值得注意的是,大规模异构计算系统消耗的能量巨大.据统计,目前计算中心服务器消耗的能量大约占全球电力总消耗的 0.5%,如果照此速度继续发展,预计到 2020 年,其能量消耗将翻番[4].此外,系统运行过程中产生的热量需要制冷设备(如空调)进行降温调节以维持系统的正常运行环境,保证系统的可靠性,而制冷设备的投入又导致更多的能量消耗.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[5];高性能计算系统生命周期内维持正常运行所需的电力成本已经超出了系统的硬件成本[6].与此同时,通过燃料发电产生的大量 CO_2、SO_2 和粉尘,对环境带来极大污染[7].因此,研究异构计算系统中的节能调度问题,减少系统能量消耗已成为当前极具发展前景的研究方向.

现如今,许多处理器都可在不同的电压下运行(如 Transmeta TM5400 和 AMD Athlon64).不同电压支持不同的处理频率和不同的执行速度,即动态电压调整(Dynamic Voltages Scaling,DVS)技术.由于一个处理器的能量消耗与其执行速度满足严格的递增凸函数关系,因此可以通过降低 CPU 电压使其运行在一个较低的速度,从而降低系统的能量消耗.与此同时,系统的快速响应能力尤为必要,有时甚至是强制的[8-9].例如,对地观测卫星获取的图像数据在传到地面处理中心后,需要低延迟处理,特别是在应对地震、海啸等紧急情况下,系统的快速响应比节能更为重要[10].相反,如果任务不是很紧急,系统应在满足用户需求的基础上尽量减少系统能量消耗.因此,仅考虑能量节约或系统的快速响应能力是不够的,必须在异构系统的开发过程中研究弹性节能问题以提高系统的自适应性和灵活性.

另一方面,在实际应用中很多用户对任务完成时间没有特别严格的限制,允许任务的完成时间与预期完成时间存在一定误差.例如,一颗卫星获取的数据首先传到地面数据处理中心进行处理,分析结果用来引导下一次观测.由于数据执行时间的不确定性导致任务完成时间不固定,可能超出用户预期完成时间,但是数据处理结果对下一颗卫星仍然有用[11].而在传统的实时系统中,如果一个任务不能在截止期内完成,则该任务将被拒绝执行.对于非实时系统而言,通常忽略每个用户的期望完成时间,仅关注任务的整体完成时间或系统吞吐率.因此,为提高系统对用户的服务质量,应充分考虑任务的期望完成时间.

众所周知,调度是提高系统性能的有效手段.但是,目前大多数调度算法没有考虑异构计算系统中的弹性节能问题,因此研究新的节能调度策略,弥补该方面的不足具有重要的理论意义和应用价值.

本文提出了一种基于 DVS 技术的异构计算系统弹性节能调度策略 EEAS.该策略综合考虑了系统弹性和用户期望.EEAS 首先满足用户期望,并在此基础上尽量减少系统能量消耗.

本文的主要贡献包括:

(1)构建了一种新的调度器模型,在该模型中充分考虑了系统弹性和用户期望.

(2)提出了一种弹性节能调度策略 EEAS,用于异构计算系统中独立任务调度.

(3)通过模拟实验评估了 EEAS 策略的性能.

本文第 2 节回顾与本文相关的研究工作;第 3 节建立一种弹性节能调度模型;第 4 节介绍本文提出的 EEAS 节能调度策略,并对该策略进行分析;第 5 节通过大量的模拟实验测试 EEAS 策略的性能;第 6 节总结全文并给出下一步的研究工作.

2 相关工作

近几十年,人们对分布式计算系统中的调度问题进行了大量研究.实践表明,多机调度在大多数情况下仍属于 NP 完全问题[12],因此,在实际应用中常采用尽可能接近最优的启发式方法来解决此类问题[8,13].调度算法通常分为两大类,即静态(离线)调度和动态(在线)调度.在静态调度中,任务具有周期性特征,且到达时间已知[14-17].对于非周期到达任务,通常需要采用动态调度算法[18-23].本文提出的调度策略采用动态调度算法,用于调度非周期、独立任务.

Aydin 等人[14]提出了一种周期实时任务静态调度方法,可根据系统负载情况计算 CPU 的最优执行速度.Mishra 等人[15]提出了一种静态能量管理

模式，通过利用静态空闲时间槽节约能量. Yu 等人[16]研究了一种离线节能任务分配策略，该策略首先被描述为一个扩展的一般分配问题，之后通过线性启发式方法解决异构 DVS 实时系统中独立任务调度问题. Zhu 等人[17]引入了多处理器系统中空闲时间共享的概念来减少能量消耗，同时提出了两种基于空闲时间共享的节能调度算法，可分别用于调度多处理器系统中独立任务和有依赖关系的任务. 尽管这些调度方法可使系统有效节能，但是均属于静态调度算法，不具有处理动态到达任务的能力.

目前，已有较多关于分布式计算系统中动态节能调度算法的研究. 例如，Ge 等人[18]研究了一种分布式 DVS 调度策略，该策略在不增加任务执行时间的情况下，通过改变任务粒度减少系统能量消耗. Nélis 等人[24]提出了两种节能调度算法，即离线算法 EDF$^{(k)}$ 和在线算法 MOTE. 两种算法均考虑了带有时间期限的实时任务. Hu 等人[19]提出了一种通过虚拟机实时迁移减少能量消耗的方法，该方法把负载分发到多个同构节点上以减少能耗. Laszewski 等人[25]研究了计算机集群系统中的虚拟机调度问题，通过 DVS 技术减少能量开销. Liu 等人[26]开发了一种集群系统中能量—性能折衷的任务复制调度算法 EPBTDCS，该算法可在任务分配到节点过程中减少通信能量消耗. Zong 等人[20]提出了两种节能复制调度算法，即节能复制算法 EAD 和性能—能量折衷复制算法 PEBD，用于提高系统性能并进行能量节约. 值得注意的是，这些算法均用于同构计算环境，不适用于具有不同节点计算能力的异构计算系统. 而在异构计算系统中，节能问题变得更复杂，因为某些处理器可能以较快的处理速度运行但消耗相对较少的能量.

也有学者对异构计算系统中的节能调度问题进行了研究. 例如，Hamano 等人[21]提出了一种调度策略，可通过调整加速参数来改善系统的节能效果. Zong 等人[27]研究了一种节能调度方法 EETDS，该方法可以灵活地减小通信能量开销，从而减少任务分配过程中的总能耗. Shekar 和 Izadi[28]研究了异构计算环境中的节能调度问题，提出了一种节能调度算法 EDLS，通过考虑各种开销因素，将任务分配到能量开销较小的节点上. 但是这些方法均没有考虑任务的完成时间限制.

与此同时，一些学者研究了异构计算系统中的实时任务节能调度问题. 例如，Zikos 和 Karatza[22]研究了 3 种基于最小排队的异构集群节能调度算法. Yan 等人[23]开发了一种基于集成 DVS 和自适应竞争的调度算法 ABB，该方法可用来优化动态能量消耗和静态能量消耗. Chen 等人[29]提出了一种实时任务节能调度策略，该策略可在符合最低满足度的情况下给出近似最优调度方案. 韩建军等人[30]针对多处理器系统，以最短任务优先调度为基础，结合共享时间回收等技术，在满足实时任务截止期的基础上尽量节能. 但是在这些实时调度算法中，如果一个任务不能在截止期内完成，该任务将被拒绝. 因此，对于有期望完成时间但不具有明确截止期的任务，这些算法并不适用.

在前期的研究工作中[31-33]，我们提出了一些异构多处理系统中的实时任务调度策略，但是这些调度策略都没有考虑任务的节能问题. 在本文中，我们针对非抢占、非周期、独立任务，提出了一种基于 DVS 技术的异构计算系统弹性节能调度策略. 该策略采用启发式算法，可根据系统负载情况自适应调整 CPU 电压，从而在最大化用户满意率的基础上减少系统能量消耗.

3　数学模型

本节主要介绍相关模型、概念和术语.

3.1　调度器模型

本文研究的异构计算系统中，计算节点可通过一个集合 $P=\{p_j, j=1\cdots|P|\}$ 表示. 假设系统中的节点是异构的，即所有节点的处理能力不同. 同时，通信能力也是异构的，即从调度器到各个处理节点的通信链路具有不同的传输能力. 图 1 给出了系统的调度器模型.

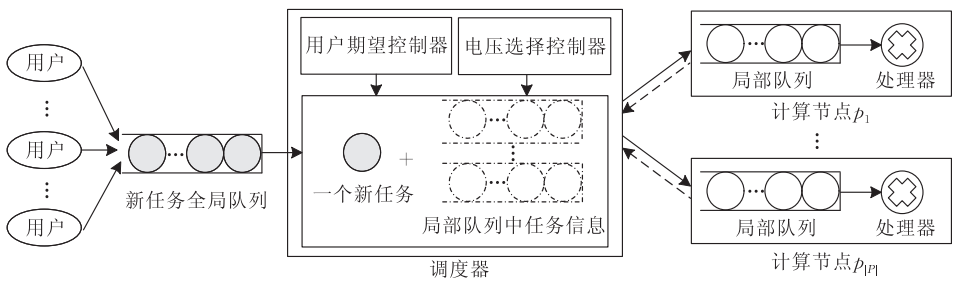

图 1　调度器模型

在如图 1 所示的调度器模型中，调度器包括一个用户期望控制器和一个电压选择控制器. 两个控制器共同工作，首先满足用户的完成时间期望，然后尽量减小系统的能量消耗. 当调度决策制定后，调度器为任务分配执行电压. 每个计算节点均有一个局部队列，任务在该队列中排队等待执行.

当一个新任务到达后，首先由调度器收集系统状态信息(例如，节点电压级别、正在执行任务的信息和节点局部队列中等待任务的信息等). 然后，调度器通过节能调度算法计算新任务是否可以在用户期望时间内完成. 在调度过程中，尽量将任务调度到具有较低执行电压的节点上. 系统的弹性主要体现在调度器可以调节新到任务和局部队列中等待任务的执行电压，从而提高调度质量.

3.2 任务模型

本文考虑一组非抢占、非周期、独立任务集合 $T=\{t_i, i=1\cdots|T|\}$，并假设任务之间没有通信. 每个任务只能在一个节点上执行，一个任务不能被分割和分发到多个节点上. 任务 t_i 用一组参数 $t_i=\{a_i, l_i, eft_i\}$ 表示，其中，a_i，l_i 和 eft_i 分别表示任务 t_i 的到达时间、任务大小和期望完成时间. 令 $\boldsymbol{AT}=(at_{ij})_{|T|\times|P|}$ 为任务的可获得时间矩阵，其中元素 at_{ij} 表示任务 t_i 在计算节点 p_j 上的可获得时间. 类似地，$\boldsymbol{ST}=(st_{ij})_{|T|\times|P|}$ 为开始时间矩阵，其中元素 st_{ij} 表示任务 t_i 在计算节点 p_j 上的开始时间. $\boldsymbol{FT}=(ft_{ij})_{|T|\times|P|}$ 为完成时间矩阵，其中元素 ft_{ij} 表示任务 t_i 在计算节点 p_j 上的完成时间. 由于节点和通信链路的异构性，令 $\boldsymbol{ET}=(et_{ij})_{|T|\times|P|}$ 为执行时间矩阵，其中元素 et_{ij} 表示任务 t_i 在计算节点 p_j 上的执行时间. $\boldsymbol{TT}=(tt_{ij})_{|T|\times|P|}$ 为传输时间矩阵，其中元素 tt_{ij} 为任务 t_i 从调度器到计算节点 p_j 的传输时间. $\boldsymbol{X}=(x_{ij})_{|T|\times|P|}$ 为任务分配矩阵，若任务 t_i 被分配到计算节点 p_j 上，则 $x_{ij}=1$；否则 $x_{ij}=0$. $\boldsymbol{O}=(o_{ij})_{|T|\times|P|}$ 为任务执行顺序矩阵，其中元素 o_{ij} 表示任务 t_i 在计算节点 p_j 上的执行顺序. $\boldsymbol{W}=(w_{ij})_{|T|\times|P|}$ 为任务等待矩阵，如果任务 t_i 在计算节点 p_j 的局部队列中等待，则 $w_{ij}=1$；否则 $w_{ij}=0$.

3.3 能量消耗模型

假设系统中的计算节点可进行动态电压调整，即具备 DVS 能力，那么每个计算节点中的处理器可在不同的电压和频率下工作. 目前，处理器中的数字电路大多采用 CMOS 集成电路，CMOS 集成电路的能量消耗主要由静态(泄露)能量消耗和动态能量消耗构成[23]. 尽管泄露能量消耗在微处理器设计中不能忽略，但由于其在总体能量消耗上所占的比例较少，通常将动态能量消耗作为节能研究的主要对象[4]. 因此，本文在构建能量消耗模型时仅考虑动态能量消耗.

令 ec_{ij} 为任务 t_i 在计算节点 p_j 上的能量消耗，该能量消耗可以表示为

$$ec_{ij}=ecr_j\cdot\tau_i \tag{1}$$

其中，ecr_j 为计算节点 p_j 上的能量消耗率，τ_i 为任务 t_i 的执行时间.

节点的能量消耗率 ecr_j 随着执行电压的改变而变化. 假设计算节点 p_j 有 k 个级别的执行电压，用 $V_j=\{V_{j1}, V_{j2}, \cdots, V_{jk}\}$ 表示，并满足 $V_{j1}<V_{j2}<\cdots<V_{jk}$. 任务 t_i 在计算节点 p_j 上的执行电压用 v_{ij} 表示，显然有 $v_{ij}\in\{V_{j1}, V_{j2}, \cdots, V_{jk}\}$. 任务 t_i 采用电压 v_{ij} 的能量消耗率记为 $ecr(v_{ij})$.

因此，任务 t_i 在计算节点 p_j 上执行的能量消耗可以计算为

$$ec_{ij}^{\text{node}}=ecr(v_{ij})\cdot et_{ij} \tag{2}$$

给定任务集合 T，计算节点集合 P，任务分配矩阵 $\boldsymbol{X}$，电压集合 V，任务执行时间矩阵 $\boldsymbol{ET}$，所有任务的整体能量消耗为

$$\begin{aligned} ec^{\text{exec}}(T,P,\boldsymbol{X},V,\boldsymbol{ET}) &= \sum_{j=1}^{|P|}\sum_{i=1}^{|T|} x_{ij}\cdot ec_{ij}^{\text{node}} \\ &= \sum_{j=1}^{|P|}\sum_{i=1}^{|T|} x_{ij}\cdot ecr(v_{ij})\cdot et_{ij} \end{aligned} \tag{3}$$

上式中，et_{ij} 可由式(4)计算：

$$et_{ij}=\frac{l_i}{s(v_{ij})} \tag{4}$$

其中，$s(v_{ij})$ 表示计算节点 p_j 采用电压 v_{ij} 执行任务 t_i 的处理速度.

值得注意的是，在式(3)中，没有考虑节点处于空闲状态时的能量消耗. 通常当节点处于空闲状态时，将其电压设定为最低值，即具有最低能量消耗率 $ecr(V_{j1})$. 因此节点空闲状态时的能量消耗可表示为

$$\begin{aligned} ec^{\text{idle}}(T,P,\boldsymbol{X},V,\boldsymbol{ET}) &= \sum_{j=1}^{|P|} ecr(V_{j1})\cdot\tau_j^{\text{idle}} \\ &= \sum_{j=1}^{|P|}\left(ecr(V_{j1})\cdot\left(\max_{i=1}^{|T|}\{ft_{ij}\}-\sum_{i=1}^{|T|}x_{ij}\cdot et_{ij}\right)\right) \end{aligned} \tag{5}$$

其中，τ_j^{idle} 为计算节点 p_j 的空闲时间，它等于 p_j 的调度长度减去 p_j 执行任务的时间. $\max_{i=1}^{|T|}\{f_{ij}\}$ 为最后一个在节点 p_j 上执行任务的完成时间，即计算节点 p_j 的调度长度.

异构计算系统的节点能量消耗可以通过式(3)

和式(5)得到

$$\begin{aligned} pec(T,P,\boldsymbol{X},V,\boldsymbol{ET}) &= ec^{\text{exec}}(T,P,\boldsymbol{X},V,\boldsymbol{ET})+ec^{\text{idle}}(T,P,\boldsymbol{X},V,\boldsymbol{ET}) \\ &= \sum_{j=1}^{|P|}\sum_{i=1}^{|T|}x_{ij}\cdot ecr(v_{ij})\cdot et_{ij}+ \\ &\quad \sum_{j=1}^{|P|}\left(ecr(V_{j1})\cdot\left(\max_{i=1}^{|T|}\{ft_{ij}\}-\sum_{i=1}^{|T|}x_{ij}\cdot et_{ij}\right)\right) \end{aligned} \tag{6}$$

在本文中，我们考虑了从调度器到各个计算节点的传输能量开销. 由于网络的异构性，令 ecr_j^{tran} 表示从调度器到计算节点 p_j 的传输能量消耗率. 因此，任务 t_i 从调度器到计算节点 p_j 的传输能量开销可计算为

$$ec_{ij}^{\text{tran}}=ecr_j^{\text{tran}}\cdot tt_{ij} \tag{7}$$

给定任务集 T，计算节点集合 P，任务分配矩阵 $\boldsymbol{X}$，电压集合 V，任务传输时间矩阵 $\boldsymbol{TT}$，传输所有任务的能量消耗为

$$\begin{aligned} ec^{\text{tran}}(T,P,\boldsymbol{X},\boldsymbol{TT}) &= \sum_{j=1}^{|P|}\sum_{i=1}^{|T|}x_{ij}\cdot ec_{ij}^{\text{tran}} \\ &= \sum_{j=1}^{|P|}\sum_{i=1}^{|T|}x_{ij}\cdot ecr_j^{\text{tran}}\cdot tt_{ij} \\ &= \sum_{j=1}^{|P|}\sum_{i=1}^{|T|}x_{ij}\cdot ecr_j^{\text{tran}}\cdot\frac{l_i}{tr_j} \end{aligned} \tag{8}$$

其中，tr_j 表示从调度器到计算节点 p_j 的通信链路传输率.

此外，通信链路空闲时的能量消耗可表示为

$$\begin{aligned} ec^{\text{link-idle}}(T,P,\boldsymbol{X},\boldsymbol{AT},\boldsymbol{TT}) &= \sum_{j=1}^{|P|}ecr_j^{\text{link-idle}}\cdot\tau_j^{\text{link-idle}} \\ &= \sum_{j=1}^{|P|}\left(ecr_j^{\text{link-idle}}\cdot\left(\max_{i=1}^{|T|}\{at_{ij}\}-\sum_{i=1}^{|T|}x_{ij}\cdot tt_{ij}\right)\right) \end{aligned} \tag{9}$$

其中，$ecr_j^{\text{link-idle}}$ 为调度器到计算节点 p_j 的通信链路在空闲时的能量消耗率，$\tau_j^{\text{link-idle}}$ 为调度器到计算节点 p_j 的通信链路空闲时间.

因此，通信链路的传输能量消耗可表示为

$$\begin{aligned} lec(T,P,\boldsymbol{X},\boldsymbol{AT},\boldsymbol{TT}) &= ec^{\text{tran}}(T,P,\boldsymbol{X},\boldsymbol{TT})+ec^{\text{link-idle}}(T,P,\boldsymbol{X},\boldsymbol{AT},\boldsymbol{TT}) \\ &= \sum_{j=1}^{|P|}\sum_{i=1}^{|T|}x_{ij}\cdot ecr_j^{\text{tran}}\cdot\frac{l_i}{tr_j}+ \\ &\quad \sum_{j=1}^{|P|}\left(ecr_j^{\text{link-idle}}\cdot\left(\max_{i=1}^{|T|}\{at_{ij}\}-\sum_{i=1}^{|T|}x_{ij}\cdot tt_{ij}\right)\right) \end{aligned} \tag{10}$$

最后，依据式(6)和式(10)可得到总体能量消耗为

$$\begin{aligned} tec(T,P,\boldsymbol{X},V,\boldsymbol{ET},\boldsymbol{TT}) &= \\ pec(T,P,\boldsymbol{X},V,\boldsymbol{ET})+lec(T,P,\boldsymbol{X},\boldsymbol{AT},\boldsymbol{TT}) \end{aligned} \tag{11}$$

3.4　调度目标

给定任务集 T，调度方案应尽可能使更多的任务在用户期望时间内完成，即

$$\max\sum_{j=1}^{|P|}\sum_{i=1}^{|T|}(x_{ij}\,|\,ft_{ij}-eft_i\geqslant 0) \tag{12}$$

为满足任务执行的用户期望，可能会导致系统能量消耗增加. 因此在调度过程中，应在满足用户期望的前提下，最大程度地降低系统能量消耗，即

$$\begin{aligned} \min\Bigg(&\sum_{j=1}^{|P|}\sum_{i=1}^{|T|}x_{ij}\cdot ecr(v_{ij})\cdot et_{ij}+ \\ &\sum_{j=1}^{|P|}\left(ecr(V_{j1})\cdot\left(\max_{i=1}^{|T|}\{ft_{ij}\}-\sum_{i=1}^{|T|}x_{ij}\cdot et_{ij}\right)\right)+ \\ &\sum_{j=1}^{|P|}\sum_{i=1}^{|T|}x_{ij}\cdot ecr_j^{\text{tran}}\cdot\frac{l_i}{tr_j}+ \\ &\sum_{j=1}^{|P|}\left(ecr_j^{\text{link-idle}}\cdot\left(\max_{i=1}^{|T|}\{at_{ij}\}-\sum_{i=1}^{|T|}x_{ij}\cdot tt_{ij}\right)\right)\Bigg) \end{aligned} \tag{13}$$

值得注意的是，仅追求最低能量消耗可能导致任务的完成时间推迟，使其不能满足用户期望. 因此，用户期望与系统节能在异构计算系统的调度过程中属于两个冲突目标. 本文提出一种弹性调度策略 EEAS，可根据系统负载情况在式(12)和式(13)之间进行权衡. 当系统负载较重时，EEAS 优先考虑满足用户期望；而当系统负载较轻时，EEAS 在满足用户期望的前提下尽量降低任务执行电压，以实现系统节能.

4　弹性节能调度策略 EEAS

本节我们给出一种弹性节能调度策略 EEAS，用于调度异构计算系统中非周期、独立任务. 此外，任务具有期望完成时间，但不具有明确的截止期. 下面首先介绍 EEAS 策略的一些性质.

性质 1.　在调度过程中，正在执行的任务不能被抢占.

$$\begin{aligned} &\forall t_i\in T,\ \sum_{j=1}^{|P|}x_{ij}=1, \\ &\forall t_i,t_k\in T,x_{ij}=1,[st_{kj},ft_{kj}]\cap[st_{ij},ft_{ij}]\neq\varnothing:x_{kj}=0 \end{aligned} \tag{14}$$

性质 2.　如果一个任务不能在用户期望完成时间内完成，该任务仍将被分配，并选择能量消耗最少的计算节点.

$$\forall t_i\in T,p_j,p_k\in P,ft_{ij}>eft_i:x_{ij}=1\,|\,ec_{ij}=\min\{ec_{ik}\} \tag{15}$$

性质 3.　对于一个新任务，首先采用最低电压进行试探，如果最低电压不能满足用户期望，逐步提高电压直到满足用户期望或者电压已经达到最高值.

$$\forall t_i \in T, p_j \in P:$$
$$(\exists \min\{v_{ij}\}: ft_{ij} \leq eft_i) \vee (v_{ij} = V_{jk}: ft_{ij} > eft_i) \quad (16)$$

性质 4. 调度在一个新任务到达后触发，即采用立即调度模式.

任务 t_i 在计算节点 p_j 上的可获得时间 at_{ij} 可计算为

$$at_{ij} = tst_{ij} + tt_{ij} = tst_{ij} + \frac{l_i}{tr_j} \quad (17)$$

其中，tst_{ij} 为 t_i 从调度器到计算节点 p_j 的传输开始时间.

任务 t_i 在计算节点 p_j 上执行的开始时间 st_{ij} 为

$$st_{ij} = \begin{cases} at_{ij}, & \sum_{i=1}^{|T|} w_{ij} \cdot x_{ij} = 0 \wedge r_j = 0 \\ at_{ij} + rt_j, & \sum_{i=1}^{|T|} w_{ij} \cdot x_{ij} = 0 \\ at_{ij} + rt_j + \sum_{o_{kj} < o_{ij}, w_{kj} = 1} et_{kj}, & \text{其它} \end{cases} \quad (18)$$

其中，$r_j = 0$ 表示没有任务正在计算节点 p_j 上执行. rt_j 表示在计算节点 p_j 上正在执行任务的剩余执行时间. $\sum_{i=1}^{|T|} w_{ij} \cdot x_{ij} = 0$ 表示没有任务在计算节点 p_j 的等待队列中排队.

任务 t_i 在计算节点 p_j 上的完成时间等于其开始时间 st_{ij} 与执行时间 et_{ij} 之和，即

$$ft_{ij} = st_{ij} + et_{ij} \quad (19)$$

EEAS 采用期望完成时间最早最优先策略对新到达任务和局部队列中等待任务进行排队，有如下性质.

性质 5. 当一个新任务被插入到局部队列时，可缩短局部队列中等待任务的执行时间以满足用户期望完成时间，从而需要重新计算队列中等待任务的开始执行时间.

假设任务 t_i 被插入到计算节点 p_j 的局部队列中，那么队列中等待任务 t_k 的新开始时间 st'_{kj} 可计算为

(1) 若 $o_{ij} = \min\{o_{mj} \mid w_{mj} = 1\}$，$o_{kj} = o_{ij} + 1$，则 $st'_{kj} = st_{kj} + et_{ij}$，如图 2 所示.

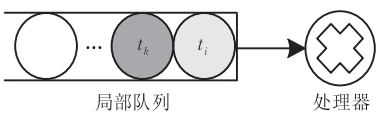

图 2 情况 1 示例

(2) 若 $o_{ij} = \min\{o_{mj} \mid w_{mj} = 1\}$，$o_{kj} > o_{ij} + 1$，则 $st'_{kj} = st_{kj} + et_{ij} - \sum_{o_{mj} < o_{kj}, o_{mj} \neq o_{ij}} (et_{mj} - et'_{mj})$，如图 3 所示.

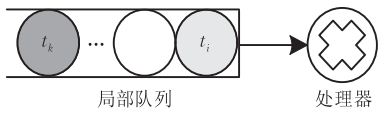

图 3 情况 2 示例

(3) 若 $o_{kj} = \min\{o_{mj} \mid w_{mj} = 1\}$，则 $st'_{kj} = st_{kj}$，如图 4 所示.

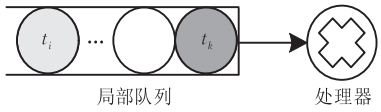

图 4 情况 3 示例

(4) 若 $o_{ij} \neq \min\{o_{mj} \mid w_{mj} = 1\}$，$o_{kj} \neq \min\{o_{mj} \mid w_{mj} = 1\}$，$o_{kj} < o_{ij}$，则 $st'_{kj} = st_{kj} - \sum_{o_{mj} < o_{kj}} (et_{mj} - et'_{mj})$，如图 5 所示.

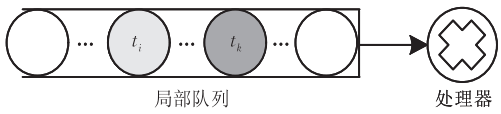

图 5 情况 4 示例

(5) 若 $o_{ij} \neq \min\{o_{mj} \mid w_{mj} = 1\}$，$o_{kj} \neq \min\{o_{mj} \mid w_{mj} = 1\}$，$o_{kj} > o_{ij}$，则 $st'_{kj} = st_{kj} - \sum_{o_{mj} < o_{kj}, o_{mj} \neq o_{ij}} (et_{mj} - et'_{mj}) + et_{ij}$，如图 6 所示.

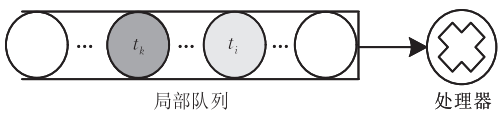

图 6 情况 5 示例

其中，et'_{mj} 表示任务 t_m 在计算节点 p_j 上的新执行时间.

任务 t_k 在计算节点 p_j 上的新完成时间可以计算为

$$ft'_{kj} = st'_{kj} + et'_{kj} = st'_{kj} + \frac{l_k}{s(v'_{kj})} \quad (20)$$

EEAS 调度策略采用启发式算法，在任务分配过程中尽量满足用户期望时间. 当一个新任务到达后，首先采用最低电压进行分配试探，然后 EEAS 计算任务在每个计算节点上的开始时间和完成时间，通过提升任务执行电压以满足用户时间期望. 如果最高电压仍不能满足用户时间期望，则调整局部队列中等待任务的执行电压，直到满足用户期望. EEAS 在执行过程中选择能量消耗和传输能量消耗最少的节点以减少系统的能量消耗.

EEAS 调度策略的伪代码如算法 1 所示.

算法 1.　EEAS 策略伪代码.

输入：任务及其属性、节点及其属性、链路及其属性、电压级别

输出：调度方案

```
1. FOR EACH new task t_i DO
2.   mSelectedNode ← NULL; nMSelectedNode ← NULL;
     furtherAdjust ← TRUE; energyCons ← ∞;
     meetExpectation ← FALSE;
3.   FOR EACH computational node p_j IN a heterogeneous computing system DO
4.     Calculate the transmission energy consumption ec_ij^tran using Eq. (8);
5.     Calculate the start time st_ij using Eq. (18);
6.     v_ij ← V_j1;
7.     IF st_ij + et_ij > eft_i THEN
8.       adjustPhase1();
9.       IF furtherAdjust == TRUE THEN
10.        adjustPhase2();
11.      END IF
12.      IF meetExpectation == FALSE THEN
13.        noMeetExpectation();
14.      END IF
15.    ELSE
16.      noNeedAdjust();
17.    END IF
18.  END FOR
19.  IF mSelectedNode ≠ NULL THEN
20.    Allocate t_i to mSelectedNode and update scheduling information;
21.  ELSE
22.    Allocate t_i to nMSelectedNode and update scheduling information;
23.  END IF
24. END FOR
```

EEAS 调度策略首先计算传输能量消耗和任务开始执行时间(见 4～5 行,算法 1). 之后 EEAS 试探系统采用最低电压是否可以满足新任务的用户期望(见 6～7 行,算法 1),如果试探结果为采用最低电压不能满足新任务用户期望,则调用函数 adjustPhase1().

算法 2.　adjustPhase1()函数伪代码.

输入：任务 t_i 在节点 p_j 上的当前选择电压 v_{ij}

输出：*furtherAdjust*、*meetExpectation* 和 *mSelectedNode* 的取值

```
1.  WHILE v_ij ≤ V_jk DO
2.    Increase one supply voltage level v'_ij ← v_ij ++;
3.    IF st_ij + et'_ij < eft_i THEN
4.      furtherAdjust ← FALSE;
5.      meetExpectation ← TRUE;
6.      Calculate the node energy consumption ec_ij^node using Eq. (2);
7.      IF ec_ij^tran + ec_ij^node ≤ energyCons THEN
8.        energyCons ← ec_ij^tran + ec_ij^node;
9.        mSelectedNode ← j;
10.       BREAK;
11.     END IF
12.   END IF
13. END WHILE
```

在函数 adjustPhase1()中,EEAS 逐渐提高新任务的执行电压直到满足其用户期望(见 2～5 行,算法 2). 之后,EEAS 选择能量消耗最少的节点(见 6～11 行,算法 2),WHILE 循环之后,如果变量 *furtherAdjust* 等于 TRUE,表明采用最高电压也不能满足用户期望,那么 EEAS 调用函数 adjustPhase2(),进一步调整局部队列中等待任务的执行电压.

算法 3.　adjustPhase2()函数伪代码.

输入：节点 p_j 局部队列中等待任务及其属性、任务 t_i 及其属性

输出：*meetExpectation* 和 *mSelectedNode* 的取值

```
1.  FOR EACH task t_m IN the local queue of p_j DO
2.    WHILE v_mj ≤ V_jk DO
3.      Increase one supply voltage level v'_mj ← v_mj ++;
4.      Calculate t_m's new start time st'_mj and execution time et'_mj;
5.      IF st'_mj + et'_mj ≤ eft_m AND st'_ij + et_ij ≤ eft_i THEN
6.        meetExpectation ← TRUE;
7.        Calculate the new node energy consumption:
          pec_j = ec_ij^node + Σ_{w_mj=1} ec_mj^node';
8.        IF ec_ij^tran + pec_j ≤ energyCons THEN
9.          mSelectedNode ← j;
10.       END IF
11.       BREAK;
12.     ELSE
13.       v'_mj = v_mj − −;
14.       BREAK;
15.     END IF
16.   END WHILE
17. END FOR
```

对于局部队列中的等待任务,如果其执行电压没有达到最高,EEAS 首先提高其执行电压(见 2～3 行,算法 3). 如果电压提高后不能使新任务和队列中等待任务满足用户期望,则将电压调回原值(见 12～13 行,算法 3);否则,说明新任务可在用户期望时间内完成(见第 6 行,算法 3). 之后,EEAS 寻找能量消耗最少的计算节点(见 7～10 行,算法 3).

如果函数 adjustPhase1( )和 adjustPhase2( )均不能使变量 *meetExpectation* 为 FALSE,即所有电压调整均不能保证新任务在用户期望时间内完成,则调用函数 noMeetExpectation( ).

算法 4. noMeetExpectation( )函数伪代码.

输入:任务 t_i 及其属性、节点 p_j 及其属性、t_i 到 p_j 链路及其属性

输出:*nMSelectedNode* 的取值

1. IF $ec_{ij}^{\text{tran}}+ec_{ij}^{\text{node}}(v_{ij}=V_{jk})\leq energyCons$ THEN
2. $nMSelectedNode\leftarrow j$;
3. END IF

在函数 noMeetExpectation( )中,EEAS 选择最高电压但具有最少能量消耗的计算节点(见 1～3 行,算法 4).

如果新任务采用最低电压可以满足用户期望,则调用函数 noNeedAdjust( ).

算法 5. noNeedAdjust( )函数伪代码.

输入:任务 t_i 及其属性、节点 p_j 及其属性、t_i 到 p_j 链路及其属性

输出:*mSelectedNode* 的取值

1. IF $ec_{ij}^{\text{tran}}+ec_{ij}^{\text{node}}(v_{ij}=V_{j1})\leq energyCons$ THEN
2. $mSelectedNode\leftarrow j$;
3. END IF

在函数 noNeedAdjust( )中,EEAS 选择具有最少能量消耗的计算节点.

回顾算法 1,如果变量 *mSelectedNode* 不为空,表明某些计算节点可以在满足用户期望的条件下执行新任务,则 EEAS 将任务分配到能量消耗最少的计算节点上.若没有计算节点可在满足用户期望的条件下执行该新任务,那么 EEAS 仍选择一个能量消耗最少的计算节点(见 19～23 行,算法 1).

定理 1. EEAS 的时间复杂度为 $O(|P||Q||K|)$,其中,$|P|$ 为计算节点数,$|Q|$ 为局部队列中的等待任务数,$|K|$ 为电压级别数.

证明. 函数 adjustPhase1( )的时间复杂度为 $O(|K|)$,adjustPhase2( )的时间复杂度为 $O(|Q||K|)$,noMeetExpectation( )和 noNeedAdjust( )的时间复杂度为 $O(1)$,其它行的时间复杂度为 $O(1)$.因此,EEAS 的时间复杂度为 $O(|P|)(O(|K|)+O(|Q|)O(|K|))=O(|P||Q||K|)$. 证毕.

5 实验测试

本节通过大量模拟实验测试 EEAS 节能调度策略的性能,将其与贪婪节能(Greedy Energy-Aware,GEA)调度算法、最高电压节能(Highest Voltage Energy-Aware,HVEA)调度算法和最低电压节能(Lowest Voltage Energy-Aware,LVEA)调度算法进行比较.

GEA,HVEA 和 LVEA 的算法描述如下:

GEA:通过调整新到任务的执行电压,尽量满足用户的期望完成时间,同时减少能量消耗;

HVEA:所有计算节点以最高电压运行.当分配一个新任务时,选择能量消耗最少的计算节点;

LVEA:所有计算节点以最低电压运行.当分配一个新任务时,选择能量消耗最少的计算节点.

本文主要从以下几个方面比较了 EEAS、GEA、HVEA 和 LVEA 的性能:

(1)用户满意率(User Satisfaction Rate,USR):$USR=$满足用户期望的任务数/全部任务数$\times 100\%$;

(2)总能量消耗(Total Energy Consumption,TEC):能量消耗之和;

(3)调度跨度(Makespan):任务集合中最后一个任务的完成时间.

本文的总能量消耗为归一化能量消耗,类似的处理方法如文献[19]和[25].

5.1 模拟方法和参数

异构计算系统中所有计算节点的电压级别变化从 0.9 V～1.5 V,步长为 0.1 V.相对应的处理器能量消耗率分别为 140 MW 到 1000 MW.处理器能量消耗率按电压等级服从均匀分布.此外,假设 0.9 V 和 1.5 V 分别对应处理速度 350 Kbps 和 1000 Kbps[23].处理速度的分布类似于能量消耗率分布.

下面给出实验中所用的模拟参数.

(1)为体现系统中节点计算能力的异构性,令 *minSpeed* 为最小执行电压下的处理速度范围,各节点最低电压下的处理速度在 *minSpeed* 内均匀分布;类似地,*maxSpeed* 为最高执行电压下的处理速度范围,各节点最高电压下的处理速度在 *maxSpeed* 内均匀分布.为体现网络传输速度的异构性,令参数 *bandWidth* 为网络传输率范围,从调度器到各个计算节点的传输率在 *bandWidth* 内均匀分布.

(2)参数 *processingPower* 用于反映计算节点从最低电压到最高电压的能量消耗率范围.参数 *transmissionPower* 为从最差通信链路到最佳通信链路的传输率范围,且能量消耗率与通信链路传输率服从均匀分布.

(3)参数 *taskSize* 表示任务大小.在实验中测试了 3 种大小的任务,即小任务、中等任务和大任务.例

如，0 KB～500 KB 属于小任务，500 KB～1000 KB 属于中等任务，1000 KB～1500 KB 属于大任务.

(4) 参数 $expectedTimebase$ 用于控制任务的期望完成时间. 式(21)中用户期望完成时间 eft_i 的计算类似于文献[34].

$$eft_i = a_i + (1 + expectedTimeBase) \times e_i^{\max} \quad (21)$$

其中，$e_i^{\max}$ 为任务 t_i 的最长执行时间：

$$e_i^{\max} = \max\{e_{ij}(V_{j1})\} \quad (22)$$

(5) 单位时间内到达的任务数服从泊松分布，参数 $intervalTime$ 表示两个连续任务间的平均时间间隔.

表 1　实验参数表

参数	取值(固定值)-(变化值)
节点数	(32)-(8,16,32,56,64,96,128)
任务数	(512)
$minSpeed$/Kbps	([250,450])-([300,400]),([250,450]),([200,500])
$maxSpeed$/Kbps	([900,1100])-([950,1050]),([900,1100]),([850,1150])
$bandWidth$/Kbps	([1250,1400])
$intervalTime$/s	(2.0)-(2.0,4.0,6.0,8.0,10.0,12.0,14.0,16.0)
$taskSize$/KB	([500,1000])-([0,500],[500,1000],[1000,1500])
$expectedTimeBase$/s	(2.0)-(2.0,4.0,6.0,8.0,10.0,12.0,14.0,16.0)
$processingPower$/s	([140,1000])
$transmissionPower$/s	([10,20])

5.2　节点数对性能的影响

在本组实验中，为测试 EEAS 策略的可伸缩性，将节点数从 8 变化到 128. 图 7 显示了 GEA、HVEA、LVEA 和 EEAS 在用户满意率、总能量消耗和调度跨度 3 个方面的性能对比.

图 7(a)显示了所有策略的用户满意率随着节点数的增加而提升，这是因为更多的节点可以保证更多的任务在用户期望时间内完成. 从图 7(a)可以发现，HVEA 一直比其它策略具有较高的用户满意率，因为 HVEA 始终将电压设定为最高值，导致任务具有较短的执行时间和较早的开始时间，因此满足用户期望的概率较高. 相反，由于 LVEA 采用了最低电压，导致最低的用户满意率. 此外，图 7(a)显示了 EEAS 调度策略在用户满意率方面优于 GEA 算法，这可以解释为 EEAS 充分利用了节点局部队列中的任务信息，可以提高等待队列中任务的执行电压以提高用户满意率，而 GEA 仅对新到达任务的执行电压进行调节，忽略了局部队列中的任务信息.

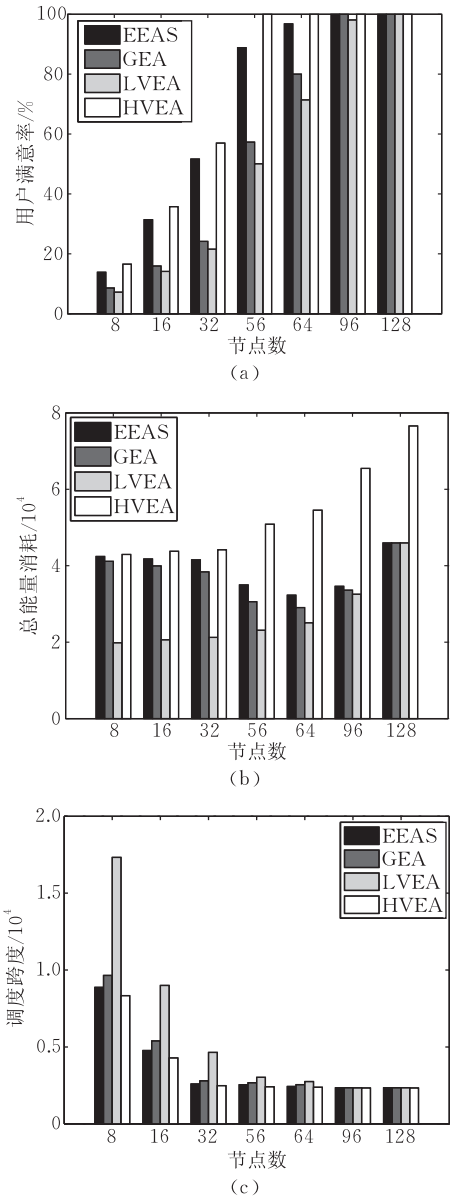

图 7　节点数对性能的影响

从图 7(b)中可以发现，尽管 HVEA 可以提供最高的用户满意率，但其节能效果最差(见图 7(a))；相反，LVEA 具有最少的能量消耗，但其用户满意率最低. 上述实验结果表明两种策略均不具有弹性，缺乏在用户满意率和总能量消耗之间进行权衡的能力. 此外，从图 7(b)可以发现，当计算节点数小于 96 时，尽管 GEA 比 EEAS 的节能效果好，但是大量任务的用户期望无法得到满足(见图 7(a))，这是因为当系统负载较重时，EEAS 为提高用户满意率增加了能量消耗. 此外，当计算节点数大于 96 时，EEAS 和 GEA 具有相似的能量消耗，这是由于系统负载较轻时，EEAS 在保证系统提供较高用户满意率的基础上减少了系统能量消耗. 值得注意的

是,当节点数从 8 变化到 64 时,EEAS 的总能量消耗逐渐减少,因为此时系统负载较轻,EEAS 在保证系统具有较高用户满意率的基础上减少了系统能量消耗. 当节点数大于 64 时,EEAS 的总能量消耗增加,这是因为尽管系统以最低电压运行,但随着节点数的增加,较多节点处于空闲状态. GEA 具有和 EEAS 类似的趋势,但是它只能调整新到任务的执行电压而不能调整等待队列中任务的执行电压,因此当系统负载较重时,不能提供比 EEAS 更优的用户满意率.

图 7(c)的结果表明,LVEA 具有最差的调度跨度,因为 LVEA 始终选择最低电压,使得任务具有最长的执行时间,因此产生最长的调度跨度; 相反,HVEA 的调度跨度最小,但是它的能量消耗一直高于其它策略的能量消耗. 这一结果表明,异构计算系统采用该策略不具备任何弹性. 当节点数小于 96 时,EEAS 比 GEA 具有更优的调度跨度,这可以解释为当系统负载较重时,为获得更高的用户满意率,EEAS 可通过提高局部队列中等待任务的执行电压来缩短任务的执行时间,但是 GEA 缺乏对局部队列中任务执行电压的调节能力. 因此,采用 EEAS 调度策略的任务相对采用 GEA 策略的任务具有较短的执行时间. 当计算节点数目大于 96 时,即使采用最低电压,系统负载仍然较轻,因此两者的任务执行时间基本相同.

5.3 任务到达率对性能的影响

为了测试任务到达率对性能的影响,在本组实验中,将参数 *intervalTime* 的值从 2 变化到 16,步长为 2. 图 8 显示了 GEA、LVEA、HVEA 和 EEAS 的性能.

图 8(a)显示了当参数 *intervalTime* 较小时,任务到达速度较快,使得较多任务在局部队列中排队,因此一些晚到达任务可能错失其期望完成时间. 随着 *intervalTime* 的增加,在局部队列中等待任务的数量减少,任务具有较早的开始时间,此时系统负载较轻,使得系统具有较高的用户满意率. 此外,HVEA 和 LVEA 分别具有最高和最低的用户满意率. EEAS 的用户满意率高于 GEA 的用户满意率,其结果与图 7(a)中所示情况的解释一致.

图 8(b)显示了 HVEA 消耗的能量最多,而 LVEA 消耗的能量最少. 值得注意的是 EEAS 具有特殊的变化特性. 例如,当参数 *intervalTime* 小于 8 时(即系统负载较重),EEAS 尽量通过增加节点局部队列中等待任务的执行电压提高用户满意率. 当

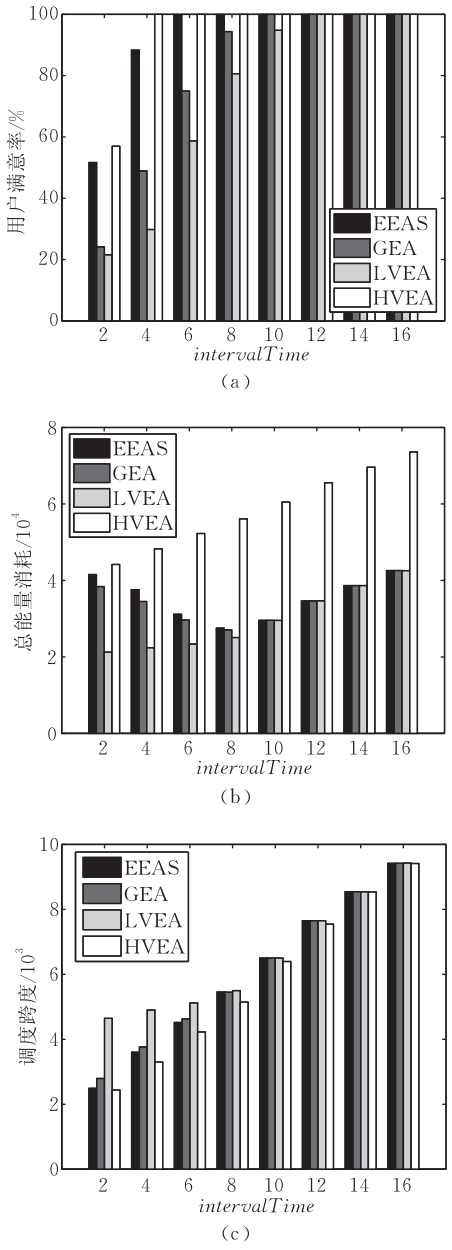

图 8 任务到达率对性能的影响

intervalTime 大于 8 时,任务到达速率较低(即系统负载较重)时,EEAS 的总能量消耗增加,这是因为系统负载变轻,空闲时间增加,增加的空闲时间使得系统的总能量消耗略微增加.

图 8(c)中的结果表明,当参数 *intervalTime* 从 2 变化到 8 时,所有策略的调度跨度增加. 这是因为随着 *intervalTime* 的增加,任务到达时间间隔变长,因此,任务的开始时间延长导致最晚到达任务的完成时间较晚,相应的调度跨度增加. 此外,HVEA 的调度跨度最小,而 LVEA 正好相反. 该原因和图 8(c)中的解释类似. 当参数 *intervalTime* 小于 8 时,EEAS 在调度跨度方面优于 GEA,这可解释为当系

统负载较重时，EEAS 优先满足用户期望，使任务具有较短的执行时间，因此调度跨度减小.

5.4　期望完成时间对性能的影响

本组实验的目的是为了测试期望完成时间对 GEA、HVEA、LVEA 和 EEAS 性能的影响. 参数 *expectedTimebase* 的变化从 2～16. 图 9 显示了实验中 4 种方法的性能.

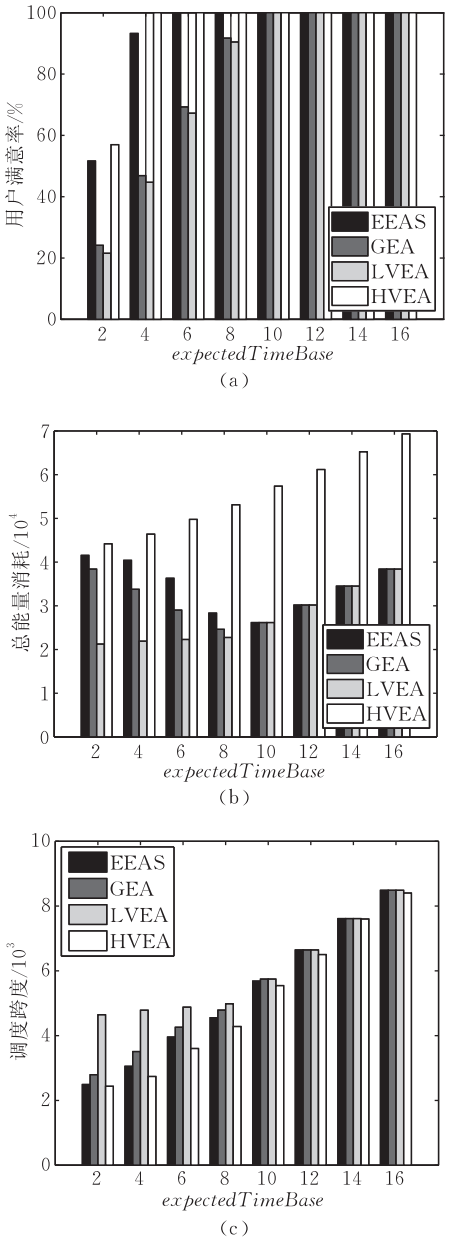

图 9　期望完成时间对性能的影响

从图 9(a)中可以发现，随着 *expectedTimebase* 的增加(即期望值变得宽松)，GEA、HVEA、LVEA 和 EEAS 的用户满意率随之增加，这是因为任务的期望完成时间限制变得宽松，使得任务可以较晚完成. 此外，图 9(a)显示了 HVEA 和 LVEA 分别具有最高和最低的用户满意率. EEAS 的用户满意率高于 GEA 的用户满意率，这与前面实验的结果类似(见图 7(a)和图 8(a)).

参数 *expectedTimebase* 的增加使得系统负载变轻. 从图 9(b)中可以发现，当 *expectedTimebase* 的值小于 10 时，采用 EEAS 策略的能量消耗逐渐减少，因此具有较好的弹性；而当 *expectedTimebase* 的值大于 10 时，其实验结果与前述实验结果类似. 尽管 GEA 与 EEAS 的变化趋势类似，但当系统负载较轻时，其用户满意率低于 EEAS. 此外，无论系统负载如何变化，HVEA 和 LVEA 均不能减少系统能量消耗.

图 9(c)显示了 LVEA 具有比其它算法较长的调度跨度，这是因为 LVEA 始终采用最低的任务执行电压，导致任务具有最长的执行时间，而 HVEA 恰好相反. EEAS 具有比 GEA 更好的调度跨度，其原因是为了保证较高的用户满意率而缩短了部分任务的执行时间.

5.5　任务大小对性能的影响

本组实验测试了任务大小对性能的影响. 3 种任务大小的设计如表 1 所示. 假设任务大小均匀分布，图 10 显示了小任务、中等任务和大任务对 GEA、HVEA、LVEA 和 EEAS 性能的影响.

图 10(a)显示了当任务为小任务时，由于任务执行时间较短，所有方法均提供较高的用户满意率. 当任务为中等任务或大任务时，任务的执行时间变长，因此用户满意率下降. 从图 10(a)中可以看出，HVEA 始终具有最高的用户满意率，而 LVEA 具有最低的用户满意率，这是因为 HVEA 和 LVEA 均属于静态算法，不具备根据系统负载情况自适应调整电压的能力. EEAS 具有比 GEA 较高的用户满意率是因为在系统负载较重时，EEAS 可以通过增加能量消耗来提高用户满意率.

图 10(b)显示了随着任务粒度的增加，采用 4 种不同策略的系统能量消耗均增加，这是因为大任务需要更长的执行时间，因此消耗的能量最多. 当任务为中等或大任务时，EEAS 的总能量消耗略高于 GEA 的总能量消耗，因为 EEAS 在系统负载较重时优先考虑用户满意率. 但是，当任务为小任务时，EEAS 和 GEA 基本上具有相同的能量消耗.

图 10(c)显示了采用 4 种不同调度策略时，任务大小对调度跨度的影响. 实验结果表明 EEAS 具有最好的弹性.

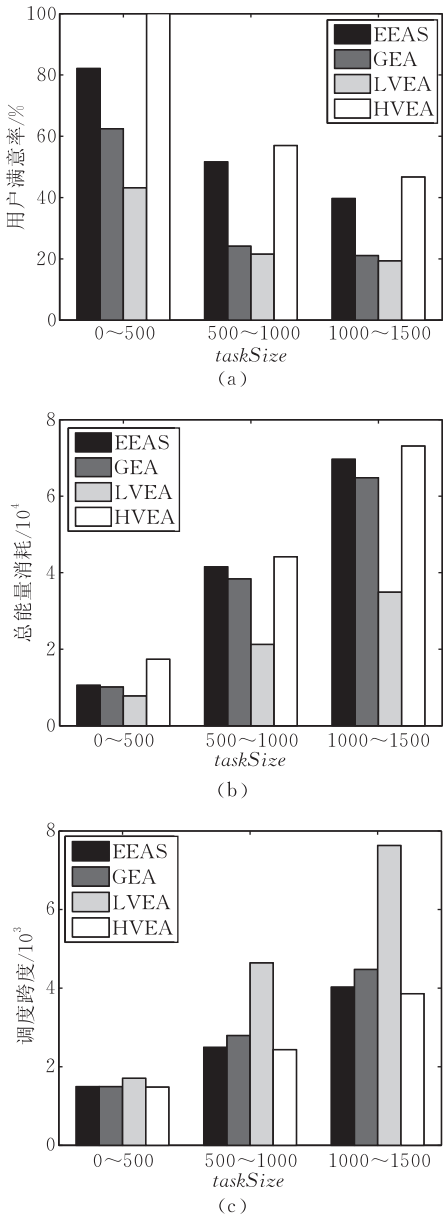

图 10 任务大小对性能的影响

6 结 论

本文研究了异构计算系统中独立任务节能调度问题,其调度目标是在用户期望与系统节能之间进行权衡.为了达到这一目标,本文提出了一种弹性节能调度策略 EEAS.该策略可有效提高系统灵活性,根据系统负载情况对新到达任务和局部队列中等待任务进行执行电压自适应调整.为了评估 EEAS 策略的性能,本文通过大量模拟实验对其进行了测试.实验结果表明在系统负载的较大变化范围内,EEAS 调度策略相比其它算法具有更好的弹性,可以为动态环境下异构计算系统提供良好的适应性.

下一步的研究工作主要包括 3 个方面:(1)将现有调度策略扩展到存储系统的节能设计中;(2)改进现有调度策略以处理异构计算系统中的并行任务;(3)进一步考虑系统中存储能力的异构性.

参 考 文 献

[1] Goller A, Leberl F. Radar image processing with clusters of computers. IEEE Aerospace and Electronics Systems Magazine, 2009, 24(1): 18-22

[2] Zheng Kan, Wang Jian-Feng, Huang Lin, Decarreau G. Open wireless software radio on common PC//Proceedings of the 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC'06). Helsinki, Finland, 2006: 1-5

[3] Qin Xiao, Jiang Hong. A dynamic and reliability-driven scheduling algorithm for parallel real-time jobs executing on heterogeneous clusters. Journal of Parallel and Distributed Computing, 2005, 65(8): 885-900

[4] Lin Chuang, Tian Yuan, Yao Min. Green network and green evaluation: Mechanism, modeling and evaluation. Chinese Journal of Computers, 2011, 34(4): 593-612(in Chinese)
(林闯, 田源, 姚敏. 绿色网络和绿色评价: 节能机制、模型和评价. 计算机学报, 2011, 34(4): 593-612)

[5] Li Yu, Liu Yi, Qian De-Pei. An energy-aware heuristic scheduling algorithm for heterogeneous clusters//Proceedings of the 15th International Conference on Parallel and Distributed Systems (ICPADS'09). Shenzhen, China, 2009: 407-413

[6] Guo Min-Yi. Green computing: Connotation and tendency. Computer Engineering, 2010, 36(10): 1-7(in Chinese)
(过敏意. 绿色计算: 内涵及趋势. 计算机工程, 2010, 36(10): 1-7)

[7] Guo Bing, Shen Yan, Shao Zi-Li. The redefinition and some discussion of green computing. Chinese Journal of Computers, 2009, 32(12): 2311-2319(in Chinese)
(郭兵, 沈艳, 邵子立. 绿色计算的重定义与若干探讨. 计算机学报, 2009, 32(12): 2311-2319)

[8] Xie Tao, Qin Xiao. An energy-delay tunable task allocation strategy for collaborative applications in networked embedded systems. IEEE Transactions on Computers, 2008, 57(3): 329-343

[9] Kang Qin-Ma, He Hong. A novel discrete differential evolution algorithm for task scheduling in heterogeneous computing systems//Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC'09). San Antonio, USA, 2009: 5006-5011

[10] Wang Jun-Ming, Li Ju-Fang, Tan Yue-Jin. Study on heuristic algorithm for dynamic scheduling problem of earth observing satellites//Proceedings of the 8th ACIS International Con-

ference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'07). Qingdao, China, 2007: 9-14

[11] Wang Hui-Lin, Li Jian-Jun, Huang Wei, Qiu Di-Shan. Area census-oriented electronic reconnaissance satellites scheduling technique under uncertain space-frequency domain environments//Proceedings of the International Conference on Electronics and Information Engineering (ICEIE'10). Weihai, China, 2010: 443-448

[12] Coffman E G. Computer and Job-Shop Scheduling Theory. Hoboken, New Jersey, USA: John Wiley & Sons, 1976

[13] Braun T D, Siegel H J, Beck N, BÄolÄoni L L, Maheswaran M, Reuther A I, Robertson J P, Theys M D, Yao Bin. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. Journal of Parallel and Distributed Computing, 2001, 61(6): 810-837

[14] Aydin H, Melhem R, Mosse D, Meja-Alvarez P. Power-aware scheduling for periodic real-time tasks. IEEE Transactions on Computers, 2004, 53(5): 584-600

[15] Mishra R, Rastogi N, Zhu Da-Kai, Mosse D, Melhem R. Energy aware scheduling for distributed real-time systems//Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'03). Nice, France, 2003

[16] Yu Yang, Prasanna V K. Power-aware resource allocation for independent tasks in heterogeneous real-time systems//Proceedings of the 9th International Conference on Parallel and Distributed Systems (ICPADS'02). Chungli, China, 2002: 341-348

[17] Zhu Da-Kai, Melhem R, Childers B R. Scheduling with dynamic voltage/speed adjustment using slack reclamation in multiprocessor real-time systems. IEEE Transactions on Parallel and Distributed Systems, 2003, 14(7): 686-700

[18] Ge Rong, Feng Xi-Zhou, Cameron K W. Performance-constrained distributed DVS scheduling for scientific applications on power-aware clusters//Proceedings of the ACM/IEEE Conference on Supercomputing (SC'05). Washington, USA, 2005: 34-44

[19] Hu Li-Ting, Jin Hai, Liao Xiao-Fei, Xiong Xian-Jie, Liu Hai-Kun. Magnet: A novel scheduling policy for power reduction in cluster with virtual machines//Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER'08). Tsukuba, Japan, 2008: 13-22

[20] Zong Zi-Liang, Manzanares A, Ruan Xiao-Jun, Qin Xiao. EAD and PEBD: Two energy-aware duplication scheduling algorithms for parallel tasks on homogeneous clusters. IEEE Transactions on Computers, 2011, 60(3): 360-374

[21] Hamano T, Endo T, Matsuoka S. Power-aware dynamic task scheduling for heterogeneous accelerated clusters//Proceedings of the 4th Workshop on High-Performance, Power-Aware Computing (HPPAC'09). Rome, Italy, 2009: 1-8

[22] Zikos S, Karatza H D. Performance and energy aware cluster-level scheduling of compute-intensive jobs with unknown service times. Simulation Modelling Practice and Theory, 2011, 19: 239-250

[23] Yan Le, Luo Jiong, Jha N K. Joint dynamic voltage scaling and adaptive body biasing for heterogeneous distributed real-time embedded systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2005, 24(7): 1030-1041

[24] Nélis N, Goossens J, Devillers R, Milojevic D, Navet N. Power-aware real-time scheduling upon identical multiprocessor platforms//Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'08). Taichung, China, 2008: 209-216

[25] Laszewski G, Wang Li-Zhe, Younge A J, He Xi. Power-aware scheduling of virtual machines in DVFS-enabled clusters//Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER'09). New Orleans, USA, 2009: 1-10

[26] Liu Wei, Li Hong-Feng, Shi Fei-Yan. Energy-efficient task clustering scheduling on homogeneous clusters//Proceedings of the 11th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'10). Wuhan, China, 2010: 381-385

[27] Zong Zi-Liang, Qin Xiao, Ruan Xiao-Jun, Bellam K. Energy-efficient scheduling for parallel applications running on heterogeneous clusters//Proceedings of the International Conference on Parallel Processing (ICPP'07). Xi′an, China, 2007: 19-26

[28] Shekar V, Izadi B. Energy aware scheduling for DAG structured applications on heterogeneous and DVS enabled processors//Proceedings of the International Conference on Green Computing (ICGREEN'10). Athens, Greece, 2010: 495-502

[29] Chen Jian-Jia, Yang Chuan-Yue, Kuo Tei-Wei, Shih Chi-Sheng. Energy-efficient real-time task scheduling in multiprocessor DVS systems//Proceedings of the 12th Asia and South Pacific Design Automation Conference (ASP-DAC'07). Yokohama, Japan, 2007: 342-349

[30] Han Jian-Jun, Li Qing-Hua, Liao Tian-Peng, Essa A A. Dynamic scheduling algorithms for the savings of power consumption in real-time multi-processor systems. Mini-Micro Systems, 2006, 27(4): 691-694 (in Chinese)
(韩建军，李庆华，缪天鹏，Essa A A. 实时多处理器系统中基于能量节约的动态调度算法. 小型微型计算机系统，2006，27(4)：691-694)

[31] Zhu Xiao-Min, Qin Xiao, Qiu Mei-Kang. QoS-aware fault-tolerant scheduling for real-time tasks on heterogeneous clusters. IEEE Transactions on Computers, 2011, 60(6): 800-812

[32] Zhu Xiao-Min, Lu Pei-Zhong. Multi-dimensional scheduling for real-time tasks on heterogeneous clusters. Journal of Computer Science and Technology, 2009, 24(3): 434-446

[33] Zhu Xiao-min, Lu Pei-Zhong. A two-phase scheduling strategy for real-time applications with security requirements on heterogeneous clusters. Computers and Electrical Engineering, 2009, 35(6): 980-993

[34] Qin Xiao, Jiang Hong. A novel fault-tolerant scheduling algorithm for precedence constrained tasks in real-time heterogeneous systems. Journal of Parallel Computing, 2006, 32(5): 331-356

ZHU Xiao-Min, born in 1979, Ph.D., lecturer. His research interests include system optimization and scheduling, as well as real-time systems.

HE Chuan, born in 1985. Ph. D. candidate. His research interests include system optimization and scheduling.

WANG Jian-Jiang, born in 1986, Ph. D. candidate. His research interests include military operations research, and combat modeling and simulation.

JIANG Jian-Qing, born in 1987, M. S. candidate. His research interest is distributed scheduling.

Background

This work presented in this paper was supported by the National Natural Science Foundation of China under Grant No. 61104180. Nowadays, increasing attention has been drawn towards the problem of energy conservation in the context of heterogeneous computing systems. However, conventional energy-aware scheduling strategies developed on these systems do not sufficiently exploit the system elasticity and adaptability for maximum energy savings, and do not take account of user expected finish time either, thereby having no capability of system elasticity. In this paper, we develop a novel scheduling strategy named elastic energy-aware scheduling EESA strategy for aperiodic and independent tasks on DVS-enabled heterogeneous computing systems. The EEAS strategy significantly improves the system elasticity while making trade-offs between user satisfaction rate and energy saving.