软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(2):266−278 [doi: 10.3724/SP.J.1001.2012.04143]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

随机任务在云计算平台中能耗的优化管理方法*

谭一鸣[1,2+], 曾国荪[1,2], 王 伟[1,2]

[1](同济大学 计算机科学与技术系,上海 200092)

[2](国家高性能计算机工程技术中心 同济分中心,上海 200092)

Policy of Energy Optimal Management for Cloud Computing Platform with Stochastic Tasks

TAN Yi-Ming[1,2+], ZENG Guo-Sun[1,2], WANG Wei[1,2]

[1](Department of Computer Science and Technology, Tongji University, Shanghai 200092, China)

[2](Tongji Branch, National Engineering and Technology Center of High Performance Computer, Shanghai 200092, China)

+ Corresponding author: E-mail: gszeng@tongji.edu.cn

Tan YM, Zeng GS, Wang W. Policy of energy optimal management for cloud computing platform with stochastic tasks. *Journal of Software*, 2012,23(2):266−278. http://www.jos.org.cn/1000-9825/4143.htm

Abstract: In the running process of cloud computing system, the idle compute nodes will generate a large amount of idle energy consumption. Furthermore, the unmatching task scheduling strategy will also cause a great waste of energy consumption. This paper presents a policy of energy optimal management for cloud computing system based on task scheduling strategy. First, use queueing system to model the cloud computing system for analyzing the mean response time, mean power consumption of cloud computing system, and constructing the energy consumption model of cloud computing system. In order to reduce waste of energy, a high service utilization task scheduling and a low execution energy task scheduling strategy are propsed, which are used to reduce idle energy and "luxury" energy respectively. Based on the idea of the strategies, an algorithm is designed which is called minimum expectation execution energy with performance constraints (ME^3PC). Repeated experiments show that this energy management strategy can reduce the energy consumption considerably while meeting performance constraints.

Key words: green cloud computing; stochastic task; energy management; task scheduling; queueing theory

摘 要: 针对云计算系统在运行过程中由于计算节点空闲而产生大量空闲能耗,以及由于不匹配任务调度而产生大量"奢侈"能耗的能耗浪费问题,提出一种通过任务调度方式的能耗优化管理方法.首先,用排队模型对云计算系统进行建模,分析云计算系统的平均响应时间和平均功率,建立云计算系统的能耗模型.然后提出基于大服务强度和小执行能耗的任务调度策略,分别针对空闲能耗和"奢侈"能耗进行优化控制.基于该调度策略,设计满足性能约束的最小期望执行能耗调度算法 ME^3PC(minimum expectation execution energy with performance constraints).实验结果表明,该算法在保证执行性能的前提下,可大幅度降低云计算系统的能耗开销.

关键词: 绿色云计算;随机任务;能耗管理;任务调度;排队论

中图法分类号: TP316 **文献标识码**: A

* 基金项目: 国家自然科学基金(61103068); 国家高技术研究发展计划(863)(2009AA012201); NSFC-微软亚洲研究院联合资助项目(60970155)

收稿时间: 2011-07-15; 修改时间: 2011-09-06; 定稿时间: 2011-11-14

目前,云计算作为一种新型的计算方式,以其高可扩展性和高可用性等优点迅速成为学术界和产业界的研究热点.例如,Google 推出了谷歌应用软件引擎(Google AppEngine,简称 GAE),IBM 推出了蓝云计算平台,Amazon 推出了弹性计算云(elastic compute cloud,简称 EC2).但是,要实现低成本、高效、安全、易用的云计算系统依然面临诸多挑战,其中,高能耗是云计算系统最为严重的问题之一.例如,Google 数据中心产生的能耗可相当于一个小型城市的总能耗[1].云计算系统中,除了处理任务时产生的必要能耗开销,其运行过程中还存在能耗浪费的现象,这表现在:(1) 由于计算任务达到的随机性,使得单位时间内到达的任务量时而稀疏,时而密集,而现有的云计算系统通常是长时间处于开启状态,等待计算任务的到达.但是当计算机处于空闲状态时,其空闲功率会占峰值功率的 50%~60%[2,3].因此,云计算系统会产生大量的空闲能耗.(2) 由于云计算系统中通常包含不同的计算机,实验结果表明,不同计算机对不同计算任务的执行功率和响应时间一般不同.例如,同一图像处理任务分别在 CPU 和 GPU 上的执行功率和响应时间不同,任务执行完成后,产生的总能耗也不同.因此,当未考虑能耗因素时,不匹配的调度方式会造成:本来用较低能耗就能解决问题,但却用了较高能耗.本文把由于任务的不合理调度而浪费的能耗称为“奢侈”能耗.因此,云计算系统中的空闲和“奢侈”能耗造成了极大的电能浪费,是造成云计算系统高能耗的原因之一,不仅带来了巨大的运营成本,而且高能耗还影响了系统的可靠性和稳定性.因此,云计算系统的能耗优化管理也就成为亟待解决的问题[4].

目前,分布式并行计算系统的能耗优化管理技术主要包括 3 类:关闭/休眠技术(resource hibernation)、电压动态调整技术(dynamic voltage scaling,简称 DVS)和虚拟化技术(virtualization).其中,关闭/休眠技术主要用来降低空闲能耗.基于该技术的能耗管理策略可分为 3 类:超时策略、预测策略和随机策略.其主要优点是可以最大限度地降低空闲能耗.其缺点是当使用计算机时需要较长的启动时间,导致系统性能一定程度的下降.2005 年,电子科技大学吴琦等人[5]研究表明:由于计算机系统业务请求具有自相似性,导致基于关闭/休眠技术的最优能耗管理策略为超时策略,并提出了当空闲时间长度服从 Pareto 分布时,基于截尾均值法小样本情况下,Pareto 分布形状参数的稳健有效估计算法和基于窗口大小自适应技术非平稳业务请求下的 DPM 控制算法.2009 年,图卢兹大学的 Costa 等人[6]设计了 GREEN-NET 框架.该框架包含 3 个部分:基于能耗感知的资源架构(energy aware resource infrastructure,简称 EARI)、可调整的资源管理系统(resource management system,简称 OAR)、信任评价组件.GREEN-NET 可用来对大规模分布式系统的能耗进行有效管理.

另外,电压动态调整技术和虚拟化技术主要用来降低执行能耗.执行能耗可定义为:任务在计算机上运行时,指令和数据驱动计算机硬件运转所产生的能耗.执行时硬件的耗电功率称为执行功率.同一个任务在执行过程中,其执行功率会随着运行阶段、执行特征的变化而变化[7].为了便于研究,本文假定执行功率为任务整个执行过程的平均耗电功率.根据 CMOS 电路动态功率公式 $P_{dynamic}\sim\alpha CV^2f$ 可知,动态功率与电压的平方成正比[8].因此,降低处理器的电压可以降低处理器的动态功率.但该方法的缺点是,随着电压的下降,处理器的性能会随之下降[9].2009 年,新加坡国立大学的 Lee 等人[9]针对嵌入式多处理器系统,提出了两种能耗感知的启发式任务调度算法 EGMS 和 EGMSIV.这两种算法在任务调度时同时考虑了任务调度顺序和电压动态调整,并采用能耗梯度作为任务调度的评价指标.EGMSIV 在 EGMS 的基础上实现了同一任务执行时的电压动态调整.2010 年,印第安纳大学的 Wang 等人[10]基于电压动态调整技术设计了一种启发式调度算法,用来降低并行任务在集群环境中执行时产生的能耗.该调度算法针对并行任务图中非关键路径上的任务,在不影响整个并行任务完成时间的条件下,降低非关键任务所调度处理器的电压来降低能耗.2010 年,佛罗里达大学的 Kang 等人[11]提出了一种基于电压动态调整的能耗优化算法.该算法针对任务预测执行时间不准确的问题,把因预测执行时间比实际执行时间要长而导致计算机空闲的时间段分配给新的任务或调整处理器电压以降低能耗.

虚拟化技术可实现多个任务在一个计算机的不同虚拟机上运行,通过提高计算机资源利用率,以减少所需计算机数量的方式降低能耗.2007 年,佐治亚理工学院的 Nathuji 等人[12]将能耗管理技术与虚拟化技术相结合,为大规模数据中心开发了一种能耗优化管理方法 VirtualPower.该方法支持虚拟机独立运行自己的能耗控制方法,并能够合理协调不同虚拟化平台之间、同一个虚拟化平台上不同虚拟机之间的能耗控制请求,实现对能耗的整体优化管理.2007 年,卡尔斯鲁厄大学的 Stoess 等人[13]开发了一个分布多层的能量控制系统.该系统包含两

个子系统:宿主级和用户级子系统.宿主级子系统从宏观上控制整个系统的能耗,根据所有用户请求对硬件资源进行合理分配,使得每个虚拟机的能耗不超过其规定的上限.用户级子系统在虚拟机层重新对虚拟硬件资源进行分配,使每个用户任务产生的能耗不超过其规定的上限.

上述 3 种能耗优化管理技术有不同的应用场景.其中,关闭/休眠技术的相关研究通常是针对计算机或处理部件的关闭/休眠时机进行设定或预测.但是对于包含有众多计算资源的云计算系统,如何根据单位时间到达的任务量决定要关闭的计算机数量,以及关闭哪些计算机等问题,都给关闭/休眠技术赋予了新的研究难题.例如,由于传统调度策略的缺陷,会导致计算机负载不平衡,甚至有计算机出现空闲的情况,此时如果再调用关闭/休眠技术,显然会严重影响整个系统的性能.电压动态调整技术的核心思想是:通过动态调整电压来使同一处理器具有不同的功率/性能"档位",用不同的档位来处理不同类型、不同计算量的任务,在降低执行能耗的同时又保证了执行性能.但是在云计算系统中,电压动态调整技术遇到以下几个问题:(1) 计算任务到达的随机性,导致很难预测下一个到达任务的类型;(2) 即使知道了任务类型,也很难准确分析该任务所适合的处理器电压"档位";(3) 电压动态调整技术主要是用来降低计算机中处理器的能耗,对整台计算机或整个云计算系统的能耗优化存在一定的局限性.虚拟化技术实现了计算机资源从物理实体向虚拟实体的迁移,提高了计算机资源的利用率.但虚拟化,特别是深层次的虚拟化本身也要付出高昂的效能代价,因为虚拟化技术通过对底层硬件部件到高层服务应用的层层虚拟,每一级的虚拟都造成了效能的损失.

本文针对云计算系统中的能耗浪费问题,通过任务调度的方式,对云计算系统的空闲和执行能耗进行优化控制,从而降低总能耗.

1 任务和系统模型

1.1 随机任务模型

由于用户提交服务请求在时间上是不确定的,导致任务到达云计算系统是随机的.例如,任务的到达间隔可能服从负指数分布、Erlang 分布等其他随机分布.另外,用户的服务请求具有一定的趋同性,例如,大量用户同时对一些网络新兴事物的关注,会导致短时间内涌现大量任务,即任务量有激增的现象.另外,用户提交的服务请求在形式上是多样的,导致到达云计算系统的任务具有不同的类型.任务类型是指任务对计算需求、执行模式的一种需求描述.例如,根据任务对计算机资源的需求特征,可分为计算密集型、通信密集型、数据密集型和 I/O 密集型等.不同类型任务要处理的数据形式和问题规模一般不同.为了便于研究,本文假定同一类型任务的计算量相同.由于用户服务请求的自主性、地域的分布性,导致不同用户提交的任务之间通常没有优先约束关系,即任务是独立的.

定义 1. 随机到达云计算系统的任务可表示为一个三元组(T,Λ,W),其中,$T=\{t_i|1\leqslant i\leqslant m\}$表示任务类型集合,$t_i$ 表示第 i 类任务,不同任务之间相互独立,即$\forall t_i\neq\forall t_j$,其中,$1\leqslant i,j\leqslant m$;$\Lambda=\{\lambda_i|1\leqslant i\leqslant m\}$表示任务的平均到达率集合,$\lambda_i$ 表示 t_i 类任务单位时间的平均到达数量,如果 $i\neq j$,则$\lambda_i\neq\lambda_j$,且 $1\leqslant i,j\leqslant m$;$W=\{w_i|1\leqslant i\leqslant m\}$表示任务的计算量集合,$w_i$ 表示 t_i 类任务的计算量.因此,第 i 类计算任务可表示为(t_i,λ_i,w_i),其中,$t_i\in T,\lambda_i\in\Lambda,w_i\in W$.

任务的平均到达率,可以根据云计算系统的大量监测数据,分析出任务到达间隔的随机分布,然后按照统计学的方法,例如χ^2 检验法,确定出属于哪种理论分布,并估计其参数值.

1.2 云计算系统模型

现有云计算平台的硬件基础设施通常是架构在大规模廉价服务器集群之上,系统中的不同服务器或计算机通常是由不同公司生产,有不同的硬件配置.这些计算机不仅有不同的功能和性能,其耗电的功率也不同,具体可表现为:(1) 功能异构.根据体系结构的不同,计算机可分为不同的类型,例如 PC、向量机、SIMD、MIMD 等.不同功能计算机对不同类型任务有截然不同的耗电功率和执行性能.例如,向量类型任务在非向量机器上执行性能很差,由于执行时间长也会导致执行总能耗较大.(2) 性能异构.计算机硬件配置不同,同样导致不同计算机的性能不同.(3) 空闲功率和峰值功率异构.由于计算机体系结构或硬件配置的不同,导致不同计算机处于空

闲或峰值性能状态时的耗电功率不同,见表 1[14].(4) 执行功率异构.不同类型任务在同一计算机上的执行功率不同,见表 2[15].究其原因是,不同类型任务的执行特征不同,导致对计算机各种硬件资源的需求和需求程度不同[16],因此执行能耗不同.另外,同一任务在不同计算机上的执行功率也不同.例如,绿色超级计算机 Top 500 的评比就是通过运行相同的 Linpack 基准程序来测量不同计算机的执行功率[17].

Table 1 Idle and peak power of different computers

表 1 不同计算机的空闲和峰值功率

生产厂商	服务器	CPU	空闲功率(W)	峰值功率(W)
IBM	x3400	Intel Xeon X5675	63.2	246
	x3200 M2	Intel Xeon E3110	75.2	117
Supermicro	2021+	AMD Opteron 2380	138	269
	1021+	AMD Opteron 2376HE	119	210

Table 2 Execution power of computer with different tasks

表 2 同一计算机不同任务的执行功率

服务器	测试程序	计算机利用率				
		20%	40%	60%	80%	100%
PRIMERGY TX150 S6	SPECpower_ssj2008	63.7W	73.2W	83.7W	92.5W	98.4W
	SPECweb2009_ASPX	150.3W	158.9W	166.4W	176.5W	183.8W

定义 2. 云计算系统可定义为六元组$(C, P_{m\times n}^{busy}, P^{idle}, P^{peak}, U_{m\times n}, S)$,其中,

- $C=\{c_i|1\leqslant i\leqslant n\}$表示云计算系统中计算机的集合,其中,$c_i$ 表示第 i 个计算机,n 为计算机的个数.
- $P_{m\times n}^{busy}=\{p_{ij}^{busy}|1\leqslant i\leqslant m,1\leqslant j\leqslant n\}$ 表示计算机的执行功率矩阵,其中, p_{ij}^{busy} 表示 t_i 类任务在计算机 c_j 上执行时的功率,如果 $i\neq h,j\neq k$,则 $p_{ij}^{busy}\neq p_{hk}^{busy}$.
- $P^{idle}=\{p_i^{idle}|1\leqslant i\leqslant n\}$表示计算机空闲功率的集合,其中, p_i^{idle} 表示计算机 c_i 处于空闲状态的功率,如果 $i\neq j$,则 $p_i^{idle}\neq p_j^{idle}$.
- $P^{peak}=\{p_i^{peak}|1\leqslant i\leqslant n\}$ 表示计算机峰值功率集合,其中, p_i^{peak} 表示计算机 c_i 处于峰值状态的功率,如果 $i\neq j$,则 $p_i^{peak}\neq p_j^{peak}$.
- $U_{m\times n}=\{\mu_{ij}|1\leqslant i\leqslant m,1\leqslant j\leqslant n\}$表示计算机平均服务率矩阵,其中,$\mu_{ij}$ 表示计算机 c_j 对 t_i 类任务的平均服务率,如果 $i\neq h,j\neq k$,则$\mu_{ij}\neq\mu_{hk}$.
- $S=\{s_{idle},s_{busy}\}$表示计算机状态的集合,其中,s_{idle} 表示计算机运行但处于空闲状态,s_{busy} 表示计算机处于执行状态.

与任务平均到达率的获取方法相同,通过对云计算系统大量监测数据随机分布的分析,可得到不同计算机处理不同类型任务的服务率矩阵 $U_{m\times n}$.P^{idle},P^{peak} 和 $P_{m\times n}^{busy}$ 则可通过测量的方式得到.

2 云计算系统性能和能耗分析

云计算系统任务调度的过程可描述为:不同类型任务以不同的速率随机到达系统,调度器则根据任务类型、任务到达时机、系统中所有计算机当前的执行状态等信息对任务进行合理映射,最后调度执行,如图 1 所示,其目标是降低系统运行过程中产生的空闲能耗和执行能耗.为了便于研究,假设对于系统中的每个计算机,任务的到达间隔相互独立,且服从同一参数的负指数分布.每个计算机对不同任务的服务时间也相互独立,且服从同一参数的负指数分布.而且到达间隔时间与服务时间相互独立.每个计算机通常都维护了一个局部的任务队列,因此,本文采用 n 个 $M/M/1$ 型排队模型对云计算系统进行建模.另外,本文暂不考虑云计算系统中与通信相关的能耗和时间开销.

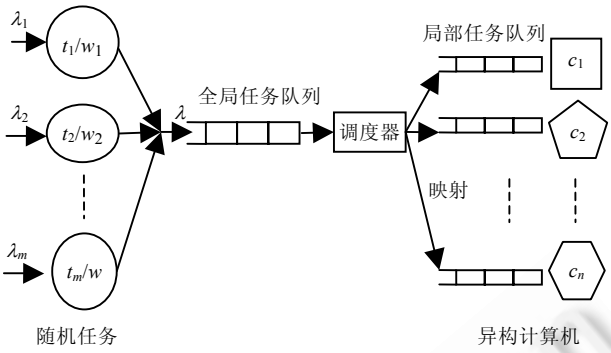


Fig.1 Task scheduling model of cloud computing system

图 1 云计算系统任务调度模型

2.1 系统的响应时间分析

根据对随机任务的定义,t_i 类任务的平均到达率为λ_i,则对于整个云计算系统,所有 m 类任务的总到达率为 $\lambda=\sum_{i=1}^{m}\lambda_i$. 由于不同计算任务之间、不同计算机之间存在差异,导致不同的调度方式会影响系统产生的能耗.因此,假设不同类计算任务以不同的概率调度到不同的计算机上,设 p_{ij} 表示 t_i 类计算任务调度到计算机 c_j 的概率,则计算机 c_j 的期望任务到达率可表示为 $E(\lambda_j)=\sum_{i=1}^{m}p_{ij}\times\lambda_i$. 由于计算机 c_j 对 t_i 类任务的平均服务率为μ_{ij},则平均服务时间为 $\frac{1}{\mu_{ij}}$, 计算机 c_j 对 t_i 类任务的服务强度可表示为 $\rho_{ij}=\frac{\lambda_i}{\mu_{ij}}$. 当考虑任务的调度概率时,计算机 c_j 对所有 m 类任务的期望服务强度可表示为

$$E(\rho_j)=\sum_{i=1}^{m}p_{ij}\times\rho_{ij} \tag{1}$$

则整个云计算系统对所有 m 类任务的期望服务强度可表示为 $E(\rho)=\sum_{j=1}^{n}E(\rho_j)$.

计算机 c_j 对 t_i 类任务的平均响应时间为 $E(t_{ij})=\frac{1}{\mu_{ij}-\lambda_i}$; 同样,当考虑计算任务的调度概率时,计算机 c_j 对 m 类任务的期望响应时间可表示为 $E(t_j)=\sum_{i=1}^{m}\frac{p_{ij}\lambda_i}{E(\lambda_j)}\times E(t_{ij})$; 整个云计算系统对 t_i 类计算任务的期望响应时间可表示为 $E(T_i)=\sum_{j=1}^{n}(p_{ij}\times E(t_{ij}))$. 由此,整个云计算系统对 m 类计算任务的期望响应时间可表示为

$$E(Time)=\sum_{i=1}^{m}\frac{\lambda_i}{\lambda}\times E(T_i) \tag{2}$$

2.2 系统的平均功率分析

根据云计算系统的定义,t_i 类任务在计算机 c_j 上的执行功率为 p_{ij}^{busy}, 当考虑任务的调度概率时,计算机 c_j 对所有 m 类任务的期望执行功率可表示为

$$E(p_j^{busy})=\sum_{i=1}^{m}\frac{p_{ij}\lambda_i}{E(\lambda_j)}\times p_{ij}^{busy} \tag{3}$$

由于任务的到达存在一定的间隔,且以一定的概率进行调度,导致系统中的计算机会以一定的概率处于空闲状态,则计算机 c_j 的期望功率(包含空闲功率和执行功率)可表示为

$$E(p_j)=P_j^{s_{idle}}\times p_j^{idle}+P_j^{s_{busy}}\times E(p_j^{busy}) \tag{4}$$

其中,$P_j^{s_{idle}}$ 表示计算机 c_j 稳态时处于空闲状态 s_{idle} 的概率,$P_j^{s_{busy}}$ 表示计算机 c_j 稳态时处于执行状态 s_{busy} 的概率,且 $P_j^{s_{idle}}+P_j^{s_{busy}}=1$.

在 M/M/1 排队模型中,空闲状态 s_{idle} 的概率可表示为 $P_j^{s_{idle}}=1-E(\rho_j)$,则 $P_j^{s_{busy}}=1-P_j^{s_{idle}}=E(\rho_j)$.公式(4)可转换为

$$E(p_j)=[1-E(\rho_j)]\times p_j^{idle}+E(\rho_j)\times E(p_j^{busy}) \tag{5}$$

则整个云计算系统对 t_i 类任务的期望功率可表示为 $E(P_i)=\sum_{j=1}^{n}p_{ij}\times E(p_j)$,整个云计算系统对所有 m 类任务的期望功率可表示为

$$E(Power)=\sum_{i=1}^{m}\frac{\lambda_i}{\lambda}\times E(P_i) \tag{6}$$

2.3 问题描述

根据第 2.1 节和第 2.2 节的分析,并结合能耗的计算公式 $E=P\times T$,则任意一个任务从进入云计算系统到执行完成离开所产生的期望能耗可表示为

$$E(Energy)=E(Power)\times E(Time)=\sum_{i=1}^{m}\left[\frac{\lambda_i}{\lambda}\times E(P_i)\right]\times\sum_{i=1}^{m}\left[\frac{\lambda_i}{\lambda}\times E(T_i)\right] \tag{7}$$

对公式(7)展开分析可知,期望能耗 $E(Energy)$与到达率集合Λ、服务率矩阵 $U_{m\times n}$、空闲功率集合 P^{idle}、执行功率矩阵 $P_{m\times n}^{busy}$ 和调度概率 p_{ij} 的值有关.但是在给定任务类型、确定云计算系统体系结构的条件下,Λ,$U_{m\times n}$,P^{idle},$P_{m\times n}^{busy}$的值是确定的,都可通过实验测量得到,只有调度概率 p_{ij} 的值是根据调度策略的不同而动态变化.可见,云计算系统的期望能耗与任务和计算机之间的调度策略有关.

虽然本文重点研究如何降低云计算系统的空闲能耗和执行能耗,但完成时间、负载平衡等性能因素依然是云计算系统重要的评价指标.因此,在降低能耗的同时必须考虑系统的执行性能.如果任务调度只考虑能耗,则可能会出现大量任务都调度到少数几个计算机上,导致负载严重不平衡的现象.更严重的是,可能会使计算机处于空闲状态,不仅产生了大量的空闲能耗,而且严重影响整个云计算系统的利用率和整体性能.因此,为了保证系统性能,必须使系统中计算机之间的负载保持平衡.为了刻画计算机的负载情况,令计算机 c_j 的负载程度表示为 $\omega_j=\dfrac{\sum_{i\in j\_local}w_i\times l_i}{\sum_{j=1}^{n}\sum_{i\in j\_local}(w_i\times l_i)/n}$,其中,$w_i$ 表示计算机 c_j 局部任务队列中 t_i 类任务的计算量;l_i 表示 t_i 类任务的个数,且 $0\leqslant\omega_j<1$.当 $\omega_j=0$ 时,表示该计算机当前处于空闲状态.并假设计算机轻载的阈值为 w_{light},重载的阈值为 w_{heavy}.一般地,负载平衡时,系统中计算机的负载应满足条件:$w_{light}<w_j<w_{heavy}$.在理想情况下,负载平衡表现为各个计算机的负载程度相等,即 $w_{light}<w_1=w_2=\ldots=w_n<w_{heavy}$.

因此,云计算系统的能耗优化管理问题可描述为:根据任务达到时机和类型、计算机类型和执行状态,对任务进行合理调度,使得任务从进入云计算系统到执行完成期间,在保证负载平衡的条件下,产生的期望能耗最小.用函数公式表示如下:

$$\begin{cases}\min E(Energy)=\sum_{i=1}^{m}\left[\frac{\lambda_i}{\lambda}\times E(P_i)\right]\times\sum_{i=1}^{m}\left[\frac{\lambda_i}{\lambda}\times E(T_i)\right]\\ 1\leqslant i\leqslant m;\ 1\leqslant j\leqslant n;\ w_{light}<w_j<w_{heavy};\ 0<\rho_{ij},E(\rho_j),E(\rho)<1\end{cases} \tag{8}$$

3 能耗优化管理

通过任务调度的方法进行云计算系统能耗优化管理的实质是:根据任务的达到时机和类型、不同计算机的

功率和性能、计算机实时的负载情况,对任务进行合理调度,使系统在满足一定性能的条件下,降低云计算系统运行过程中产生的空闲和执行能耗.公式(8)表示的能耗优化问题是 NP 难题,难以找到最优解.因此,本文设计了一个基于优先规则的启发式算法来进行问题的求解.

3.1 调度概率分析

设 $P_{m\times n}=\{p_{ij}|1\leqslant i\leqslant m,1\leqslant j\leqslant n\}$表示调度概率矩阵,其中,$p_{ij}$ 表示 t_i 类任务调度到计算机 c_j 的概率,如果 $i\neq h$, $j\neq k$,则 $p_{ij}\neq p_{hk}$.p_{ij} 的值受计算机 c_j 对 t_i 类任务的平均服务率和执行能耗、t_i 类任务的平均到达率、计算机 c_j 的空闲功率和当前负载情况等因素的影响.为了得到任务与计算机之间的调度概率,首先引入匹配度的定义.

定义 3. 任务与计算机之间的匹配度是指不同类型任务在不同计算机上执行效果的匹配程度.t_i 类任务与计算机 c_j 的匹配度 m_{ij} 可表示为

$$m_{ij}=\frac{\alpha\cdot\overline{p}_j^{idle}+\beta\cdot\rho_{ij}}{\chi\cdot\overline{e}_{ij}^{busy}+\gamma\cdot\omega_j} \tag{9}$$

其中,$\overline{p}_j^{idle}=\exp\left(\frac{p_j^{idle}-\max\limits_{1\leqslant j\leqslant n}\{p_j^{idle}\}}{\max\limits_{1\leqslant j\leqslant n}\{p_j^{idle}\}}\right)$表示空闲功率因子,且 $0<\overline{p}_j^{idle}\leqslant 1$,$\alpha$为权值;$\rho_{ij}=\frac{\lambda_i}{\mu_{ij}}$ 表示计算机 c_j 对 t_i 类任务的服务强度,且 $0<\rho_{ij}<1$,β为权值;$\overline{e}_{ij}^{busy}=\exp\left(\frac{e_{ij}^{busy}-\max\limits_{1\leqslant j\leqslant n}\{e_{ij}^{busy}\}}{\max\limits_{1\leqslant j\leqslant n}\{e_{ij}^{busy}\}}\right)$表示执行能耗因子,且 $0<\overline{e}_{ij}^{busy}\leqslant 1$, 其中,$e_{ij}^{busy}=p_{ij}^{busy}\times\frac{1}{\mu_{ij}}$ 表示 t_i 类任务在计算机 c_j 上的执行能耗,χ为权值;ω_j 表示计算机 c_j 的负载程度,且 $0\leqslant\omega_j<1$,γ为权值.

于是,t_i 类任务调度到计算机 c_j 的概率可表示为 $p_{ij}=\frac{m_{ij}}{\sum_{j=1}^{n}m_{ij}}$. 调度概率公式表明:当 t_i 类任务调度时,空闲功率和对该任务服务强度较大的计算机、执行能耗和负载程度较小的计算机,其调度概率较大.各个权值反映了各个量的重要程度,在不同情况下,可通过改变各个权值的大小来决定任务的调度策略.

3.2 能耗优化策略

云计算系统产生的能耗可分为空闲能耗和执行能耗,因此,本文采用不同的策略分别对空闲能耗和执行能耗分别进行优化.

3.2.1 空闲能耗优化策略

虽然计算机空闲功率 p^{idle} 是峰值功率 p^{peak} 的 50%~60%,但计算机执行时,由于受 CPU 利用率[16]、存储器访问率等因素的影响,性能往往达不到峰值性能.因此,计算机执行时的功率 p^{busy} 也就往往小于峰值功率 p^{peak}. 可见,计算机空闲功率p^{idle}占执行功率p^{busy}的比例将更大.因此,要降低云计算系统的总能耗,必须降低空闲能耗.

基于大服务强度的任务调度策略,任务从进入计算机 c_j 到执行完成所产生的期望空闲能耗可表示为

$$E(Energy_j)_{idle}=P_j^{s_{idle}}\times p_j^{idle}\times E(t_j) \tag{10}$$

根据对公式(10)的分析可知,要降低空闲能耗 $E(Energy_j)_{idle}$,可通过降低计算机 c_j 的空闲概率 $P_j^{s_{idle}}$ 来实现. 在 M/M/1 排队模型中,计算机 c_j 的空闲概率为 $P_j^{s_{idle}}=1-E(\rho_j)$, 因此,要降低 $P_j^{s_{idle}}$, 可以通过增大计算机 c_j 的期望服务强度 $E(\rho_j)$来实现.由 $E(\rho_j)=\sum_{i=1}^{m}p_{ij}\times\rho_{ij}$ 可知,要使 $E(\rho_j)$增大,可通过将在计算机 c_j 上有大服务强度ρ_{ij}的任务优先调度到计算机 c_j 上实现.该策略主要用于对空闲或轻载计算机的任务调度,目的是降低计算机空闲的概率. 总之,该策略可理解为:使系统中的所有计算机保持在忙碌状态,以降低系统中计算机空闲的时间.该策略的实施可通过增大调度概率 p_{ij} 计算公式中权值β的值来实现.

3.2.2 执行能耗优化策略

当任务的类型与计算机类型相匹配时,不仅有较好的性能,而且会有较小的执行能耗.即使相匹配计算机的执行功率较大,但由于有较短的执行时间,依然可以使总执行能耗最小.例如,图像处理任务分别在 CPU 和 GPU 上执行时,即使 GPU 的执行功率略大于 CPU,但 GPU 可以在较短时间内完成,导致 GPU 产生的总能耗依然要低于 CPU.因此,图像处理任务和 GPU 有较大的调度概率.

基于小执行能耗的任务调度策略:t_i 类任务在计算机 c_j 的执行能耗可表示为 $E(Energy_{ij})_{busy}=p_{ij}^{busy}\times\frac{1}{\mu_{ij}}$,则计算机 c_j 对所有 m 类任务的期望执行能耗可表示为

$$E(Energy_j)_{busy}=P_j^{s_{busy}}\times\sum_{i=1}^{m}\left(\frac{p_{ij}\lambda_i}{E(\lambda_j)}\times E(Energy_{ij})_{busy}\right) \tag{11}$$

根据对公式(11)的分析可知,要降低计算机 c_j 的期望执行能耗 $E(Energy_j)_{busy}$,可通过将在计算机 c_j 上有小执行能耗的任务优先调度到计算机 c_j 上来实现.即 t_i 类任务调度时,选择有 $\min\{E(Energy_{ij})_{busy}|1\leqslant j\leqslant n\}$ 的计算机.该策略主要用于对处于轻载和重载之间计算机的任务调度,目的是降低任务执行时所消耗的能耗.该策略的实施可通过减小调度概率 p_{ij} 计算公式中权值 χ 的值来实现.

3.3 性能约束的最小期望执行能耗调度算法ME^{3}PC(minimum expectation execution energy with performance constraints)

根据第 3.2 节提出的能耗优化策略,本文设计了一种满足性能约束的最小期望执行能耗调度算法 ME^{3}PC.该算法的主要思想是:根据云计算系统中计算机的负载情况,将所有计算机分成 3 个集合 $C=\{C_{light},C_{normal},C_{heavy}\}$,其中,$C_{light}=\{c_j|1\leqslant j\leqslant n,0\leqslant w_j\leqslant w_{light}\}$,$C_{normal}=\{c_j|1\leqslant j\leqslant n,w_{light}<w_j<w_{heavy}\}$,$C_{heavy}=\{c_j|1\leqslant j\leqslant n,w_{light}\leqslant w_j\leqslant 1\}$.当任务进行调度时,只要 $C_{light}\neq\varnothing$,则优先考虑集合 C_{light} 中的计算机,并采用基于大服务强度的任务调度策略,使对 t_i 类任务有大服务强度的计算机 c_j 有较大的调度概率,其中,$j\in C_{light}$.如果 $C_{light}=\varnothing$,且 $C_{normal}\neq\varnothing$,就考虑集合 C_{normal} 中的计算机,并采用基于较小执行能耗的任务调度策略,使对 t_i 类任务有较小执行能耗的计算机 c_j 有较大的调度概率,其中,$j\in C_{normal}$.如果 $C_{light}=\varnothing$,$C_{normal}=\varnothing$ 且 $C_{heavy}\neq\varnothing$,即 $w_{heavy}\leqslant\forall w_j$,其中,$1\leqslant j\leqslant n$,表示所有计算机都处于重载状态.这样,在任务调度时,把任务调度到负载最小的计算机上,该策略可通过降低调度概率 p_{ij} 计算公式中负载参数权值 γ 的值来实现.总之,ME^{3}PC 算法是针对不同的情况,通过改变参数 α,β,χ,γ 的值来调用相应的调度策略.性能约束的最小期望执行能耗调度算法 ME^{3}PC 的伪代码描述如下:

算法 1.

```
L1:  for all classes incoming tasks
L2:    generate Q_global using FCFS strategy;
L3:  end for
L4:  for each task in Q_global do
L5:    for all computer j do, where j∈C
L6:      calculate w_j;
L7:      create C_light, C_normal, C_heavy;
L8:    end for
L9:    if C_light≠∅
L10:     initialize the parameters α, β, χ, γ for class i, p←0;
L11:     for each computer j do, where j∈C_light
L12:       p_ij=Calculate(α_1,β_1,χ_1,γ_1);
L13:       If p<p_ij then
L14:         p=p_ij, c_j=j;
```

L15: scheduling t_i→computer c_j;
L16: end for
L17: else if (C_{light}=∅)∩(C_{normal}≠∅)
L18: initialize the parameters α, β, χ, γ for class i, p←0;
L19: for each computer j do, where $j \in C_{normal}$
L20: p_{ij}=*Calculate*(α_2,β_2,χ_2,γ_2);
L21: If $p<p_{ij}$ then
L22: $p=p_{ij}$, $c_j=j$;
L23: scheduling t_i→computer c_j;
L24: end for
L25: else if (C_{light}=∅)∩(C_{normal}=∅)∩(C_{heavy}≠∅)
L26: initialize the parameters α, β, χ, γ for class i, p←0;
L27: for all computer j do, where $j \in C_{heavy}$
L28: p_{ij} =*Calculate*(α_3,β_3,χ_3,γ_3);
L29: If $p<p_{ij}$ then
L30: $p=p_{ij}$, $c_j=j$;
L31: scheduling t_i→computer c_j;
L32: end if
L33: end for
L34: end if
L35: end for

定理 1. 满足性能约束的最小执行能耗调度算法 ME^{3}PC 的最坏时间复杂度为 $O(3mn)$,其中,m 为任务的类型数,n 为云计算系统中计算机的个数.

证明:第 5 步~第 8 步,对计算机负载量进行计算,并按照负载量对其进行分类,时间开销为 $O(2)$,一共有 n 个计算机,因此,总时间开销为 $O(2n)$.第 9 步~第 16 步、第 17 步~第 24 步、第 25 步~第 33 步,3 种执行情况执行的最坏时间复杂度均为 $O(n)$,但是对于一个任务,执行路径只可能是 3 种情况中的 1 种,因此,第 9 步~第 33 步的总时间开销依然为 $O(n)$,则从第 5 步~第 33 步的总时间开销为 $O(3n)$.最后,因为一共有 m 类任务,因此整个 ME^{3}PC 算法的最坏时间复杂度为 $O(3mn)$. □

4 实 验

4.1 实验环境的设置

为了验证 ME^{3}PC 算法的有效性,本文使用 Matlab 的离散事件模拟工具进行模拟实验.实验环境涉及的相关参数以及取值或取值范围见表 3.

实验中任务分为 4 类.第 i 类任务的到达间隔时间服从参数为 $1/\lambda_i$ 的负指数分布.该间隔时间可通过负指数分布函数 exprnd($1/\lambda_i$)来生成,其中,λ_i 的值在[10,15]区间随机生成.根据任务的到达间隔,利用函数 *cumsum*(·)可得到第 i 类每个随机任务到达系统的时刻.最终可确定所有 6 000 个任务到达系统的时刻.计算机 c_j 对 t_i 类任务的服务时间服从参数为 $1/\mu_{ij}$ 的负指数分布,同样,服务时间也通过函数 exprnd($1/\mu_{ij}$)来生成,参数 μ_{ij} 的值在[1,5]区间随机生成.

特别地,当 λ_i 和 μ_{ij} 的值随机产生时,需满足条件 $0<\rho=\frac{\lambda}{\mu}<1$, 其中,$\lambda=\lambda_1+\lambda_2+\lambda_3+\lambda_4$, $\mu=\sum_{j=1}^{16}\left(\frac{1}{4}\sum_{i=1}^{4}\mu_{ij}\right)$, 该条件保证了模拟系统的运行存在平稳状态.

Table 3 Parameters of simulation environment

表 3 模拟环境参数设置

参数	设置	说明
m	6 000	随机到达的任务总数
t_i	$1\leqslant i\leqslant 4$	任务类型的数量
n	16	系统中计算机的个数
λ_i	[10,15]	第 i 类任务的平均到达率
μ_{ij}	[1,5]	计算机 c_j 对 t_i 类任务的平均服务率
w_i	[1,10]	t_i 类任务的计算量
$(\alpha,\beta,\chi,\gamma)$	(1.1,5.2,1.5,1.3)	当 $C_{light}\neq\varnothing$时,调度概率计算公式中参数的取值
	(1.1,1.4,0.3,1.5)	当 $C_{light}=\varnothing$且 $C_{normal}\neq\varnothing$时,参数的取值
	(1.2,1.1,1.3,0.2)	当 $C_{light}=\varnothing$,$C_{normal}=\varnothing$且 $C_{heavy}\neq\varnothing$时,参数的取值
w_{light}	0.25	计算机轻载阈值
w_{heavy}	0.75	计算机重载阈值
p_i^{idle}	[50,60]	计算机 c_i 的空闲功率
p_{ij}^{busy}	[100,150]	计算机 c_j 对 t_i 类任务的执行功率

4.2 实验与结果分析

实验过程是:当每个任务进入系统时,记录其任务类型、进入时刻.根据设计的 ME^{3}PC 算法对任务进行调度,并记录任务调度到的机器序号.当任务进入该机器的局部任务队列,而队列为空,即该机器为空闲时,该任务的完成时刻等于其到达时刻与服务时间之和.当队列非空时,即该机器为忙碌时,其等待时间等于队列中前一任务的完成时刻减去该任务的到达时刻,该任务的完成时刻则等于其进入时刻与等待时间、服务时间之和.最后,任务的完成时刻减去进入时刻即为系统对该任务的响应时间.当 6 000 个任务全部执行完成时,模拟实验结束.所有任务响应时间的平均值即为任务在该系统中的平均响应时间 $Time_{avg}$.根据计算机 c_j 执行过的任务类型和相应任务个数、相应执行功率和服务时间,可计算出计算机 c_j 的执行时间 $Time_j^{busy}$ 和执行能耗 $Energy_j^{busy}$.在计算机 c_j 上,最后一个任务的完成时刻减去第一个任务的进入时刻为计算机 c_j 的完成时间 $Time_j^{total}$;计算机 c_j 的空闲时间 $Time_j^{idle}=Time_j^{total}-Time_j^{busy}$;计算机 c_j 的空闲能耗为 $Energy_j^{idle}=p_j^{idle}\times Time_j^{idle}$;则计算机 c_j 的总能耗为 $Energy_j^{total}=Energy_j^{busy}+Energy_j^{idle}$;计算机 c_j 的平均功率为 $Power_j=\dfrac{Energy_j^{total}}{Time_j^{total}}$.对于整个系统,所有任务的完成时间为 $\max\{Time_j^{total}\mid 1\leqslant j\leqslant n\}$,系统的平均功率为 $Power_{avg}=\dfrac{1}{n}\sum_{j=1}^{n}Power_j$.于是,单个任务在系统中执行时的平均能耗为 $Energy_{avg}=Power_{avg}\times Time_{avg}$.

为了进一步说明 ME^{3}PC 算法的有效性,本文又设计了最小执行功率调度算法(minimum execution power,简称 MEP),并将 ME^{3}PC 与 MEP、经典 MIN-MIN 算法进行比较.其中,MEP 的算法思想是:在任务调度时,将任务调度到执行功率最小的机器上,而不考虑该计算机当前的负载情况以及执行该任务的服务时间.MIN-MIN 是针对独立任务的动态调度算法,广泛应用于同构或异构分布式并行计算环境,有良好的调度性能.本文分别从系统执行任务的平均能耗、任务的平均响应时间、系统的平均功率、负载平衡和可扩展性这 5 个方面对 3 种算法进行对比分析.其中,算法的可扩展性分析是以任务总数 m=6000,单位时间到达任务数λ,计算机数 n=16 为基准,同比例地增加或减少 3 个参数的值,又做了 8 组模拟实验,每组实验可用参数(m,λ,n)来表示,则实验的参数值分别为 $(375,2^{-4}\times\lambda,1)$,$(750,2^{-3}\times\lambda,2)$,$(1500,2^{-2}\times\lambda,4)$,$(3000,2^{-1}\times\lambda,8)$,$(6000,\lambda,16)$,$(12000,2\times\lambda,32)$,$(24000,2^2\times\lambda,64)$,$(48000,2^3\times\lambda,128)$,$(96000,2^4\times\lambda,256)$.实验结果如图 2~图 6 所示.

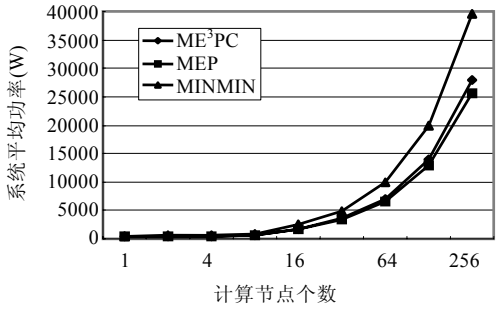


Fig.2 Average power of system

图 2 系统平均功率

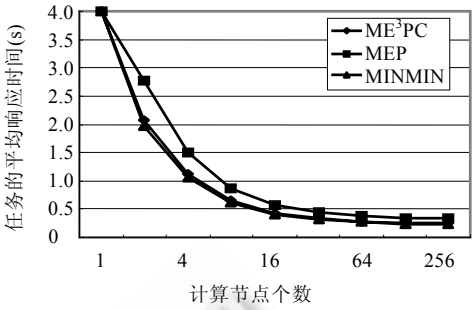


Fig.3 Average sojourn time of tasks

图 3 任务的平均响应时间

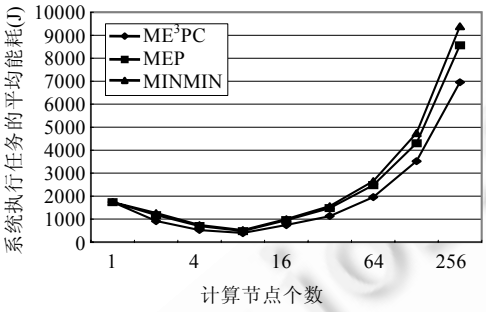


Fig.4 Average energy of one task running in system

图 4 系统执行任务的平均能耗

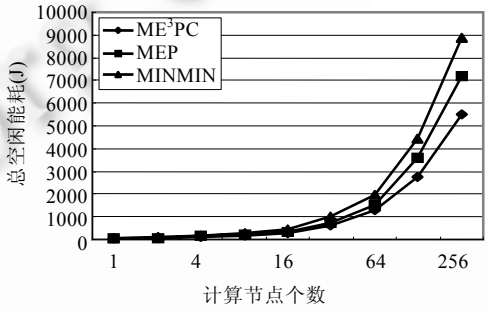
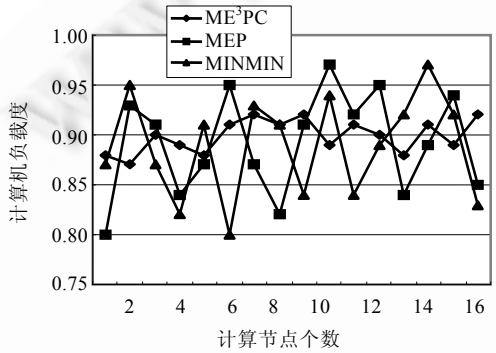
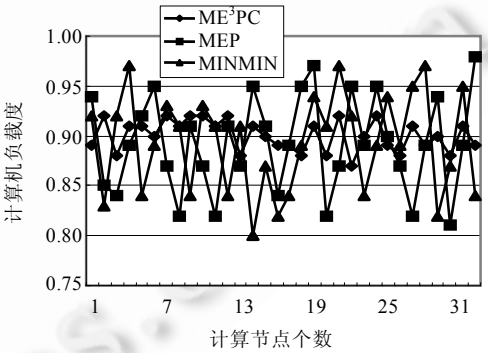


Fig.5 Total idle energy

图 5 系统总空闲能耗



(a) 16 个计算机的负载分布



(b) 32 个计算机的负载分布

Fig.6 Load distribution of computers

图 6 计算机的负载分布

从图 2 可以看出,采用 MEP 算法时系统的平均功率最小,ME^{3}PC 略大于 MEP,MIN-MIN 的功率最大,且远大于 ME^{3}PC 和 MEP.分析原因是,MEP 算法专注于系统执行功率的优化,任务调度时只将任务调度到执行功率最小的机器上,因此系统平均功率最小,但没有考虑调度的机器性能,因此总体性能最差.MIN-MIN 算法则相反,只专注于任务的完成时间,而不考虑能耗、负载平衡等其他因素,因此响应时间最小,但是系统平均功率最大.ME^{3}PC 算法则同时考虑了功率和性能因素,虽然系统功率较 MEP 算法平均增加了 6.4%,响应时间比 MIN-MIN 算法平均增加了 5.7%,但是任务在系统中的所产生的能耗却最小,如图 4 所示.究其原因是:(1) 由于 ME^{3}PC 算

法针对空闲或轻载计算机采用了大服务强度优先的调度策略,大大降低了系统中计算机出现空闲的概率,使系统的利用率高于 MEP 和 MIN-MIN.其中,采用 ME^{3}PC 算法时,系统产生的空闲能耗平均是 MEP 和 MIN-MIN 算法的 81%和 73%,如图 5 所示.(2) 由于优先对空闲和轻载计算机进行调度,最大限度地避免了负载不平衡的发生.因此,ME^{3}PC 算法保证了系统的负载平衡,如图 6(a)、图 6(b)所示.可见,只有同时考虑功率和性能因素才能真正降低云计算系统的能耗.

从实验结果中,我们发现了一个有趣的现象:当计算机个数为 1 时,3 种算法下的系统平均功率、任务平均响应时间和任务平均能耗自然都相等.但是随着计算机个数的增加,系统执行任务的平均能耗却呈线性下降.当计算机个数为 8 时,系统执行任务的平均能耗最小.随着计算机个数的继续增加,系统执行任务的平均能耗开始增大,且呈指数级增长.究其原因是:(1) 当计算机个数从 1 增加到 8 的过程中,任务平均响应时间呈指数级减少(如图 3 所示),但是系统的平均功率的增势却基本趋于平稳(如图 2 所示),这导致系统执行任务的平均能耗在该过程中是逐渐降低的,并且由于此时系统平均功率的绝对值较小,因此降低的趋势呈线性.(2) 当计算机个数从 8 增加到 128 的过程中,任务平均响应时间下降的趋势趋于平缓(如图 3 所示),但是系统的平均功率的增势却呈指数级增加(如图 2 所示),这导致系统执行任务的平均能耗在该过程中是逐渐增加的,并且由于此时系统平均功率的绝对值较大,因此增长的趋势呈指数级.由此可见,在等能耗的条件下,计算机个数为 8 时系统有最好的扩展性.对于实际的云计算系统,如何根据系统的体系结构、任务到达的规律确定系统中应该开启或关闭的机器个数,以及开启或关闭哪些机器进行能耗的优化控制,将作为下一步的研究内容.

5 结束语

本文针对云计算系统中存在的能耗浪费问题,利用不同计算机空闲功率、执行功率不同的现象,结合任务到达的随机性,计算机执行状态的动态变化特征,建立了随机任务模型和云计算系统模型,从性能和功率两方面对云计算系统进行分析,并建立了云计算系统的期望能耗模型.针对空闲能耗和执行能耗,分别提出了大服务强度和小执行能耗任务调度策略对其进行优化.大量实验表明,本文提出的能耗优化管理方法在保证其性能的前提下,大幅度降低了云计算系统的能耗.下一步的工作将研究在给定和真实的云计算系统体系结构下,如何根据任务到达率的大小和分布规律,决策系统中应该处于运行状态的计算机个数,结合关闭/休眠技术和电压动态调整技术,进一步对云计算系统的能耗进行优化控制,并且将研究的理论成果在实际云平台上进行评测,以验证其正确性.

致谢 在此,我们向对本文工作给予支持和建议的同行,尤其是 NASAC2011 年会上给本文提出宝贵意见的各位专家表示感谢.

References:

[1] Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging it platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 2009,25(6):599−616. [doi: 10.1016/j.future.2008.12.001]

[2] Kansal A, Zhao F. Fine-Grained energy profiling for power-aware application design. SIGMETRICS Performance Evaluation Review, 2008,36(2):26−31. [doi: 10.1145/1453175.1453180]

[3] Barroso LA, Holzle U. The case for energy-proportional computing. Computer, 2007,40(12):33−37. [doi: 10.1109/MC.2007.443]

[4] Lin C, Tian Y, Yao M. Green network and green evaluation: Mechanism, modeling and evaluation. Chinese Journal of Computers, 2011,34(4):593−612 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2011.00593]

[5] Wu Q, Xiong GZ. Adaptive dynamic power management for non-stationary self-similar requests. Journal of Software, 2005,16(8):1499−1505 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/1499.htm [doi: 10.1360/jos161499]

[6] Costa GD, Gelas JP, Georgiou Y, Lefevre L, Orgerie AC, Pierson JM, Richard O, Sharma K. The green-net framework: Energy efficiency in large scale distributed systems. In: Mei A, ed. Proc. of the Int'l Symp. on Parallel and Distributed Processing. Piscataway: IEEE Computer Society, 2009. 1−8. [doi: 10.1109/IPDPS.2009.5160975]

[7] Blume H, Livonius JV, Rotenberg L, Noll TG, Bothe H, Brakensiek J. OpenMP-Based parallelization on an MPcore multiprocessor platform—A performance and power analysis. Journal of Systems Architecture, 2008,54(11):1019−1029. [doi: 10.1016/j.sysarc.2008.04.001]

[8] Venkatachalam V, Franz M. Power reduction techniques for microprocessor systems. ACM Computing Surveys, 2005,37(3):195−237. [doi: 10.1145/1108956.1108957]

[9] Lee KG, Veeravalli B, Viswanathan S. Design of fast and efficient energy-aware gradient-based scheduling algorithms for heterogeneous embedded multiprocessor systems. IEEE Trans. on Parallel and Distributed Systems, 2009,20(1):1−12. [doi: 10.1109/TPDS.2008.55]

[10] Wang LZ, Laszewski GV, Dayal J, Wang FG. Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS. In: Parashar M, Buyya R, eds. Proc. of the 10th IEEE/ACM Int'l Conf. on Cluster, Cloud and Grid Computing. Melbourne: IEEE Computer Society, 2010. 368−377. [doi: 10.1109/CCGRID.2010.19]

[11] Kang J, Ranka S. Dynamic slack allocation algorithms for energy minimization on parallel machines. Journal of Parallel and Distributed Computing, 2010,70(5):417−430. [doi: 10.1016/j.jpdc.2010.02.005]

[12] Nathuji R, Schwan K. Virtualpower: Coordinated power management in virtualized enterprise systems. In: Bressoud TC, Kaashoek MF, eds. Proc. of the ACM SIGOPS Symp. on Operating Systems Principles. Washington: ACM Press, 2007. 265−278. [doi: 10.1145/1294261.1294287]

[13] Stoess J, Lang C, Bellosa F. Energy management for hypervisor-based virtual machines. In: Chase J, Seshan S, eds. Proc. of the USENIX Annual Technical Conf. Berkeley: USENIX Association, 2007. 1−14.

[14] All published SPECpower_ssj2008 results. 2011. http://www.spec.org/power_ssj2008/results/power_ssj2008.html

[15] All SPEC WEB2009 results. 2009. http://www.spec.org/web2009/results/web2009.html#JSP

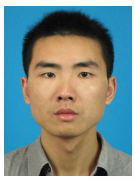
[16] Lien CH, Bai YW, Lin MB. Estimation by software for the power consumption of streaming-media servers. IEEE Trans. on Instrumentation and Measurement, 2007,56(5):1859−1870. [doi: 10.1109/TIM.2007.904554]

[17] The Green500 list-November 2010. 2010. http://www.green500.org/lists/2010/11/top/list.php

附中文参考文献:

[4] 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. [doi: 10.3724/SP.J.1016.2011.00593]

[5] 吴琦,熊光泽.非平稳自相似业务下自适应动态功耗管理.软件学报,2005,16(8):1499−1505. http://www.jos.org.cn/1000-9825/16/1499.htm [doi: 10.1360/jos161499]



谭一鸣(1982－),男,河南偃师人,博士生,主要研究领域为绿色计算,异构并行计算,可重构计算.

曾国荪(1964－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为异构并行计算,信息安全.

王伟(1979－),男,博士,讲师,CCF 会员,主要研究领域为分布并行计算,可信计算.