

成本约束下的云计算能耗资源调度优化算法^①

郝亮^{②*} 崔刚^{*} 曲明成^{*} 张康^{**}

(^{*} 哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

(^{**} 德克萨斯大学达拉斯分校计算机学院 美国达拉斯 75080)

摘 要 针对云计算能耗严重的问题,对云数据中心资源使用率、网络工作负载以及实时功率进行了深入研究,提出了在成本约束下的能耗资源调度优化算法,并提出根据世界时区性考虑电价成本,利用负载均衡的方法进行任务的调度执行。在此基础上设计了任务分层算法(TSA),通过增加任务执行的并行度与相关性,降低了数据中心节点的空闲概率,减少了数据中心间的传输能耗。当不满足约束成本时,根据此算法依次进行任务的迭代演算。模拟实验表明,基于成本约束的能耗优化算法能够在能耗优化的同时显著地节约服务的成本。

关键词 成本约束,云计算,资源调度,能耗优化,任务分层算法(TSA)

0 引言

随着全球气候变暖以及低碳经济和绿色能源的提出,云计算的能源消耗问题已成了一个关键问题^[1]。然而资源调度算法很少考虑云计算的能耗问题。尽管云计算可通过使用大型共享服务器和存储单元来节约能源^[2],但每年云数据中心部署的电脑越来越多,能源消耗有增无减^[3]。研究表明,一个 300W 的服务器运行一年的能耗相当于用于火力发电的煤炭产生 1300kg 二氧化碳释放的能量。最近有报告显示,2006 年美国的数据中心的云计算能耗占总耗电量的 1.5%,2011 年这一比例约为 3%,到 2015 年服务器操作的成本将超过购买的服务器硬件成本^[3]。同时,由此造成的网络流通负载以及网络带来的能量消耗在逐渐增多,因此如何使资源的调度算法能够解决云计算的能量消耗问题逐渐成为研究的重点。本文对云计算多数据中心的资源使用、当前能量消耗状况、区域能耗比等方面进行了深入研究,提出一种在低成本约束下,通过设计的云计算任务分层算法(task slicing algorithm, TSA)进行云计算过程中能耗的最小化计算;提出基于世界区

域性来研究云计算服务的调度问题。本文方法降低了云数据中心空闲概率,增加了任务执行的并行特性。通过仿真实验证明,在成本大幅度降低的同时,降低了使用中的能耗。

1 相关工作

当前对分布式计算有很多调度算法,如 Rajni 等人^[4]提出了利用网格资源调度的细菌觅食优化技术,设计了一种新的细菌觅食的超启发式资源调度算法,在网格环境中可利用资源有效地安排工作,与现有的启发式的调度算法结合,通过 gridsim 工具包对算法性能进行了评估。通过降低成本和完工时间把用户应用程序提交给网格。Lien 等人^[5]提出了一种云计算虚拟云桌面调度算法,该算法以优化资源的使用和提高客户的满意度为目的,通过优化,可以提高 29% 的资源利用率。当把空闲的在线冗余服务器池加入到系统负载时,可以提高 36.6% 的资源利用率,从而提高了效率。但上述资源调度算法只是优化了时间、资源利用率等服务质量参数。由于对能耗缺乏具体的研究,在云计算领域中,关于能耗的资源调度算法很少。

① 863 计划(2007AA041603,2006AA040202,2006AA04Z259),国家自然科学基金(61272382,61075076,61075077,60905047,60643005),国家重点实验室(SKLR200802B),黑龙江省自然科学基金(ZJG0709,F200909),广东省自然科学基金(8152500002000003,S2012010009963)资助项目。

② 男,1984 年生,博士生;研究方向:分布式计算,云计算资源管理等;联系人,E-mail: haoliangoor@163.com
(收稿日期:2013-04-01)

一些文献提出了云计算能耗的一些简单模型。如宋杰等人^[6]提出了一种云计算环境下的能效模型以及一种度量方法,可用以有效地计算单机以及云环境的计算机能效,计算结果和测量值非常接近。但是这种能效模型只考虑了 CPU 的任务量,没有考虑存储以及网络等辅助设备的能效,并且需要大量的 CPU 监控代理,没有考虑异构环境下的能效。Jayant 等人^[3]通过对在公有云和私有云中数据传输和转换的能耗进行了详细的测量和统计,对云计算软件作为服务、存储作为服务、平台作为服务三种服务的能耗情况进行了详细的分析,详细地列举了通过计算机、路由器、数据中心的能耗,并比较了三种服务的能耗状况。但是没有具体的算法来减少云计算中产生的能耗。Dzmitry^[7]设计了一个带有多个数据中心的云计算仿真环境,通过对工作负载的实际分布,设计了数据中心各服务器的转换、连接、传输等能量消耗,论证了电源控制如电压、频率、动态开关被应用到计算和网络中的影响。但是没有具体的调度算法去考虑服务质量参数性能影响。

在现阶段有一些考虑了能耗的算法。Anton 等人^[8]提出了一个云体系框架,并基于此框架设计了 3 种虚拟机移植算法:MM 最少移植算法,移植最少的虚拟机到 CPU 利用率最低的服务器上;迁移虚拟机的选择 (HPG) 算法,把当前 CPU 利用率最低的虚拟机迁移以解散当前虚拟机;RC 无目标优化法,以达到 CPU 利用率的最优化为前提,进而达到降低云能耗的目的。但是上述 3 种算法未考虑成本、时间等约束条件,缺少对虚拟机详细的算法推理过程,而且没有考虑不同云模块间地理位置的移植开销。Cardosa 等人^[9]从 Map reduce 模块虚拟机角度出发,把有相似起始结束时间的任务放在一起执行,以达到服务器的最高利用率,通过对空间中虚拟机摆放排列的算法达到 CMU 计算时间最小和能耗最小,但是未考虑任务迁移所需的能耗。Ivan 等人^[10]通过对虚拟机的移植、分配、合并来满足用户的服务,从 CPU 内存以及网络带宽的利用率来分析,并且利用不同 HPC 工作负载仿真评估了实验结果。这些能耗优化算法都是从虚拟机角度考虑的,但是虚拟机本身的添加和合并以及迁移本身消耗了大量的能量。

针对以上研究中的不足,本文在分布式计算调度算法的基础上进行了云计算能耗的研究,提出了一种云计算资源调度算法,该算法结合调度本身产生的成本进行约束,在减少调度能耗的同时,满足了

成本的约束,提高了系统的效率。

2 问题模型定义

在云计算模型下,任务的计算、数据的传输和存储、网络的带宽、传输的距离等所有资源都以服务形式存在,以下我们给出一些资源使用模型的定义。

定义 1 $T(t, e, w)$ 为随机任务的集合, t 为 n 个任务的集合, $0 \leq i \leq n, i$ 不等于 j, t_i 不等于 $t_j, e(i, j)$ 为任务之间的执行先后关系, w 为任务的工作量需求。

定义 2 云计算模型 (D, V, P_1, P_2) , 其中 D 为 d 个云服务器数据中心集合, $D = \{D_1, D_2, \dots, D_d\}$, V 为随机任务与计算节点对应矩阵, V_{ij} 表示任务 t_i 在处理器 p_j 上执行。 p_1 为节点空闲时功率, p_2 为节点峰值功率。

定义 3 云计算中各数据中心的网络带宽矩阵 $B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1d} \\ \vdots & \vdots & \vdots & \vdots \\ b_{d1} & b_{d2} & \dots & b_{dd} \end{bmatrix}$, 其中 b_{ij} 为云网络数据中心各节点之间的传输带宽。

定义 4 $T_1, T_2, \dots, T_m$ 为任务能耗分层算法得出的任务能耗层, T_1 的工作量大于 T_2 的工作量, 当不满足成本约束时, 从第一层任务依次调度执行。分层算法在 3.2 节给出。

定义 5 电费价格定义为 A 。因世界各地区域电费价格有所不同, 所以 A 的值有所不同, 同时因为执行的时间效率等不同因素影响, 能耗有所不同。本文在研究中假设拟定 6 个世界时区电价与能耗的研究: 亚太时区、欧洲时区、南北美洲时区、大洋洲及新西兰时区和非洲时区。每个时区假设 5 个数据中心, 由于时区的差别, 每个数据中心实时工作负载程度和网络状况以及电价成本有所不同, 研究中定义各时区价格分别为 $A = \{A_1, A_2, \dots, A_6\}$ 等。

网络传输距离定义为 L , 云计算任务的成本定义为 C , 任务的成本分别为 $C_1, C_2, \dots, C_n$, 传输功率定义为 P_3 , 能耗定义为 E 。其中 f 为节点 CPU 的频率, u 为节点 CPU 的使用率。成本约束能耗优化是指在云计算任务完成的过程中, 在执行成本约束条件下以能耗最小化为目标进行计算, 模型形式化描述为

$$\sum_{i=0}^n \sum_{j=1}^k X_{ik} A_{jk} = C \quad (1)$$

$$s. t. \sum_{k=1}^n X_{ik} = 1, \forall i \in V \quad (2)$$

$$X_{ik} \in \{0,1\} \quad (3)$$

$$O_j - O_i \geq \sum_{k=1}^n t_{ik} X_{ik} \quad (4)$$

$$\text{Min} E \quad (5)$$

其中 X 是布尔变量^[11], 表示任务每次只能选择一个服务去执行; i, k 分别表示当前任务与可选择服务个数, O_i 表示每一个任务的结束时间; 式(1)表示要给定的云服务的约定总费用; 式(2)表示每次任务只可以选择一个服务去执行, 其中 s. t. 为公式约束条件: 使得... 满足...; 式(3)表示 X_{ik} 是布尔变量; 式(4)表示任务满足偏序关系; 式(5)表示要最小化能耗。

3 算法的设计与分析

3.1 数据中心能耗分析

根据 Jayant 等人^[3]对空闲节点和峰值节点功率的比较可以计算得出, 云计算空闲计算节点的功率占峰值功率的 60% 左右, 这样空闲节点消耗了大量的能耗。而在数据中心满载运行时的计算节点也由于 CPU 使用率的关系完全没有达到峰值功率 P_1 , 这样空闲计算节点的能耗进一步占了总能耗的大部分。如何优化数据中心空闲节点能耗成了首要的问题。假设数据中心总计算节点数目为 N , 空闲的计算节点百分比为 X , 那么执行计算节点占有率为 $(1 - X)$, 总能耗 E 为

$$E = P_1 N f (1 - X) t_1 + P_2 N X t_2 + P_3 \frac{L_{ij}}{B_{ij}} \quad (6)$$

根据上式可以看出, 计算时间、峰值与空闲功率和节点总数均为常数, 能耗大小只和空闲节点百分比 X 有关, 对式(6)进行 X 变换得

$$E = P_1 f N + P_3 \frac{L_{ij}}{B_{ij}} + (P_2 N t_2 - P_1 f N t_1) X \quad (7)$$

由式(7)可见, 为了减少计算节点产生的能耗, 可以通过减少空闲节点百分比 X 来实现, 所以提出了一种基于任务能耗分层的策略。该策略首先在云数据中心进行分层数据采集, 并通过分层算法进行工作需求能耗分层, 优先调度首层任务到空闲服务器执行, 降低空闲计算节点百分比。

3.2 分层算法

任务分层算法(TSA)是将任务按照给定的工作量 w 需求大小进行计算, 将计算出的结果按照工作量的指数分布分成 M 层, 每一层任务具有较大的相关性。TSA 任务增量形式化表达为

$$m = \frac{\sum_{i=1}^n \int_0^t f_i(t) u_i(t) dt}{\log_2 \sum_{i=1}^n \int_0^t f_i(t) u_i(t) dt} \quad (8)$$

$$\Delta w = \frac{\int_{t_{n-1}}^{t_n} f_n(t) u_n(t) dt - \int_{t_1}^{t_2} f_1(t) u_1(t) dt}{m} \quad (9)$$

$$T_i = T_{i-1} + \Delta w \quad (10)$$

其中, m 为任务层数, 任务工作需求最大的一层 T_1 为第一层, 其它排序依次为 $T_2, T_3, \dots, T_m$ 。调度算法以降低空闲节点百分比为目标, 首先调度首层任务到电价较小空闲的数据中心上执行。当调度需求再次到来时, 依次调度下一层任务到空闲或次空闲数据中心执行。分层算法通过增加任务的并行执行特性, 降低了完成时间和系统空闲概率, 从而降低了任务执行开销。

3.3 算法基本描述

成本约束的能耗资源调度优化算法 (energy consumption optimization algorithm with cost constraints, ECOC²) 给定一个任务的初始成本、能耗, 当初始成本小于最小约束的成本时, 算法结束, 否则, 按照任务分层进行任务的工作量需求进行分层调度。云数据中心进行分层采集计算, 同时进行时间区域成本的比较, 优先调度到成本较低且空闲的数据中心上, 较低的空闲服务器调度首个任务层 T_1 的任务执行, 结束后比较总消耗成本, 当成本小于等于成本约束时结束, 否则, 进行下一任务层的调度, 直到满足最小成本要求。算法步骤如下:

L1: for all data centers and incoming tasks

L2: generate C and C'

L3: end for

L4: if $C' \leq C$

L5: jump L28

L6: else if $C' \geq C$

L7: create $T_1, T_2, \dots, T_m$ using TSA

L8: for all data center $D_1, D_2, \dots, D_d$

L9: calculate P and A

L10: end for

L11: for ($i = 1; i < d; i++$)

L12: $j = i + 1$

L13: if $P_{D_i} < P_{D_j}$

L14: $P = P_{D_i}$ $A = A_{D_i}$

L15: else while $P_{D_i} = P_{D_j}$ calculate A_{D_i} and A_{D_j}

L16: if $A_{D_i} < A_{D_j}$

L17: $P = P_{D_i}$

```

L18:      end if
L19:      end while
L20:      scheduling  $T_i$  to data center  $D_i$ 
L21:      end if
L22:      calculate  $C'$ 
L23:      if  $C' \leq C$ 
L24:          jump 28
L25:      else continue loop
L26:      End for
L27:      end if
L28:      return E

```

3.4 算法分析

ECOC² 算法首先在给出的成本约束与最低成本进行比较,进行成本的调度调整,在调整成本的同时,做出能耗优化调整。能耗优化调整选择的规则为:优先选取空闲服务器,当空闲服务器满时,选取负载略低的节点进行处理。因为能耗等于功率乘以时间,所以当任务选择电量价格较小的区域时,消耗的成本计算如下:

$$(P_3 \frac{L}{B_{ij}} + P_2 t) A_j \quad (11)$$

公式(11)为当调度到新节点时的总的成本值。这样在选择距离较远而成本较低的云服务器区域后,产生的能耗差值与价格差值计算如下:

$$(P_3 \frac{L}{B_{ij}} + P_2 t) - P_1 t \quad (12)$$

$$P_1 t A_i - (P_3 \frac{L}{B_{ij}} + P_2 t) A_j \quad (13)$$

式(12)为调度到新节点后产生的能耗差值,式(13)为由此产生的成本差值。经过计算后,比较此时的成本与最低成本约束之间的差值,当此时的成本小于等于最低成本约束则结束,否则任务按照 TSA 依次进行算法迭代演算。

3.5 算法复杂度分析

ECOC² 算法的复杂度要决定于初始成本的规定与最低成本约束的差距,因为这决定了算法的迭代次数。由于每次算法迭代后都越来越接近最低成本,所以算法是收敛的。算法复杂度决定于云服务节点之间电价的比较,云服务节点越多,搜索比较难度越复杂。当初始成本逼近最低成本时,任务依次迭代,因此算法复杂度最大为 XN , X 为可选的云服务器区域节点, N 为任务的总数量。

4 实验结果分析

4.1 实验设置

为了测试 ECOC² 算法在不同环境下的性能和运行效率,本文实验环境为 30 台系统 LINUX2.6.18 内核,CPU 为 Inter Core(TM)2 Duo 1.86GHz, RAM 为 2G,硬盘 500GB 的 PC 机,并采用 cloudsim 模拟了云计算仿真环境,算法采用 JAVA 语言编程。

为了比较 ECOC² 算法的性能,算法与文献[12]中的 ME³PC^[12] 进行比较,ME³PC 算法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。ME³PC 的基本思想是:优先从平均功率方面计算任务调度的顺序,把任务分配到低功率空闲计算节点去执行。当空闲节点满时,把任务分配到负载较少的计算节点去执行。实验选取了任务规模、数据中心节点数目、节点区域、成本约束 4 项参数。实验计算节点规模分别为 16,32,64,128 个节点,任务分别模拟了 {20, 40, ..., 120} 情况,分别模拟了 6 个区域节点的电价以及实时负载,约束成本区间选定在 [500, 800] 元。实验结果取不同规模下的平均值、算法选取平均成本、系统平均执行功率、执行平均总能耗等几方面综合以比较两种算法之间的关系。若两种算法在同一组数据的平均值进行比较,如平均能耗的比较,算法性能的提高率^[11] 计算为 $\frac{E_B - E_A}{E_A} \times 100\%$ 。

4.2 随机任务规模变化对结果的影响

图 1 是算法在不同任务规模下的系统执行平均能耗的比较。从实验结果计算可以得出,在开始阶

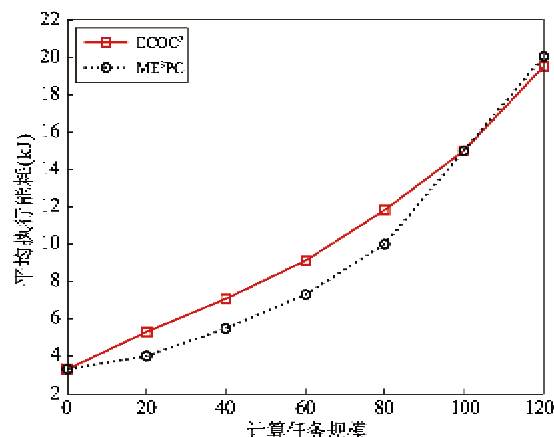


图 1 任务规模与能耗关系图

段, $ECOC^2$ 算法比 ME^3PC 算法在不同任务规模下的平均能耗略高一些, 经计算平均约为 6.27%。随着任务规模的逐渐增大, 这部分差距逐渐减小。这主要是因为, 在能耗优化的过程中, 采用成本约束的算法优先选取了距离相对较远的计算节点进行任务的计算, 这样虽然相对节约了成本, 但略为加大了能耗的传输开销。而随着任务规模的逐渐增大, 由于采用了任务分层算法, 使得相关度很大的一些任务都在同一数据中心执行, 在计算过程中减少了数据的传输能耗, 同时由于任务的并行度逐渐增大, 执行总时间相对减小, 所以平均能耗与 ME^3PC 算法逐渐接近。

图 2 是算法在不同任务规模下的系统执行平均成本的比较。从图中可以看出, 两种算法初始成本相同, 随着任务规模的逐渐增加, 两种算法出现不同的增长趋势。 ME^3PC 算法线性增加, $ECOC^2$ 呈水平抛物线形增加。这主要是因为 $ECOC^2$ 由于成本的约束, 优先把相关性较大的任务层调度到了电价较小的空闲数据中心去执行, 避免了成本增加过快, 逐渐趋于一个稳定增长线性数值。而 ME^3PC 算法随着任务规模的逐渐增大, 成本逐渐增加。经计算, $ECOC^2$ 算法在不同任务规模下成本平均降低了 14.79%。而随着任务规模的增大, 这个值还会继续增加。

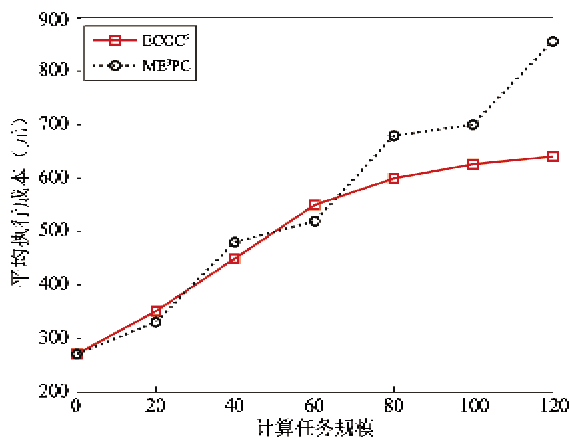


图2 任务规模与服务成本关系图

算法在不同任务规模下的系统执行平均功率的比较如图 3 所示。从图中可以看出, 两种算法平均功率随着任务量的增加呈同时上升的趋势。在任务量较小时, $ECOC^2$ 比 ME^3PC 平均功率要高, 中期时出现过略低情况, 随着任务规模的增大, 逐渐等于并小于 ME^3PC 算法。这主要是因为 $ECOC^2$ 在任务量

较小时成本还没有约束, 任务大部分都在同一数据中心执行。任务的并行度较低, 相对于 ME^3PC 空闲度较高, 所以平均功率较高。当任务逐渐增多时, 由于成本的约束, 相关度较高的一些任务层被调度到相对空闲的计算节点执行。由于任务相关度的原因节省了关键数据在数据中心各计算节点间相互传递的功率, 当这部分平均功率逐渐等于并小于调度数据产生的平均功率时, $ECOC^2$ 算法的曲线逐渐等于并小于 ME^3PC 算法。

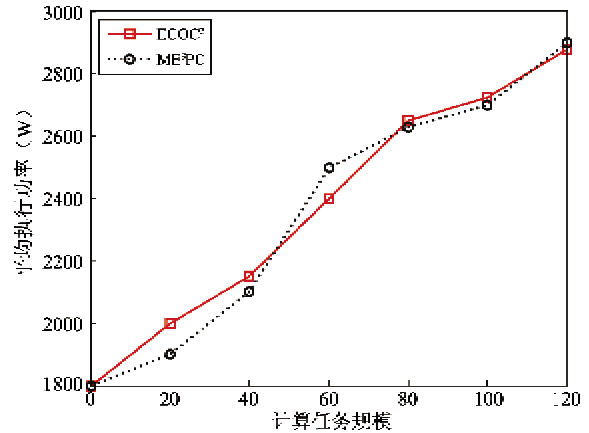


图3 任务规模与功率关系图

4.3 约束成本变化对能耗产生的影响

图 4 是在任务规模为 80, 数据中心规模为 64 时, 约束成本的变化对两种算法平均执行能耗的影响。从图中可以看到, ME^3PC 算法几乎不受成本的影响, 而 $ECOC^2$ 算法随着约束成本的逐渐增加, 系统平均能耗逐渐下降, 最后小于 ME^3PC 算法的能耗。这主要是由于约束成本较低时, 执行的任务需要调度的次数比较多, 系统传输能耗的增长大于任务并行执行时的数据传递能耗。当约束成本逐渐

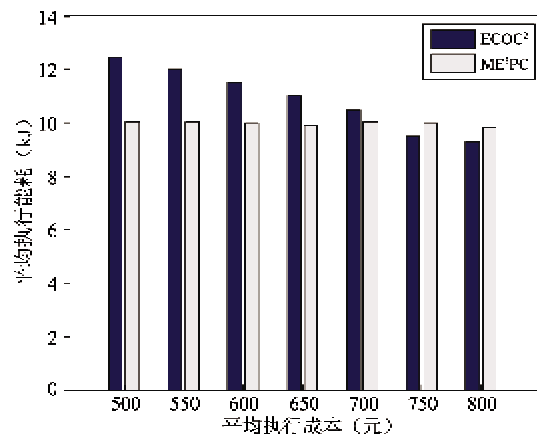


图4 成本变化与能耗关系图

增加时,任务调度次数相对减少,这部分能耗逐渐降低,执行能耗逐渐减少,并在一定规模时趋于稳定。

4.4 数据中心规模变化对结果的影响

图5和图6分别是数据中心规模变化对两种算法的平均响应时间和负载平衡的比较示意图。从图中可以看出,随着数据中心规模的逐渐增大,ECOC²算法比ME³PC算法的系统平均响应时间、系统负载平衡都要低一些。这主要是因为ECOC²算法在任务执行的过程中,由于任务分层算法使得任务具有相关性,所以在任务调度过程中节省了大量开销,增加了任务执行的并行特性,降低了空闲计算节点的开销,降低了任务的平均响应时间,所以ECOC²算法的系统负载平衡要低于ME³PC算法。

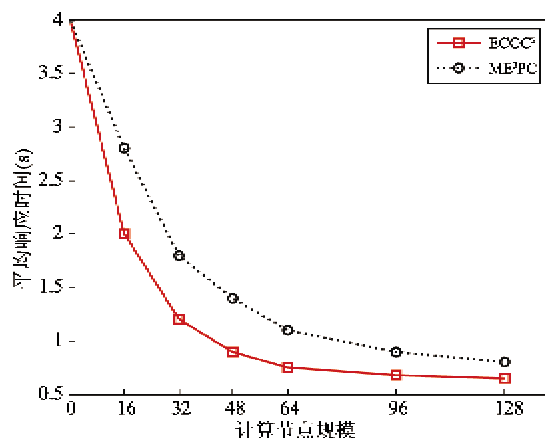


图5 节点规模与响应时间关系图

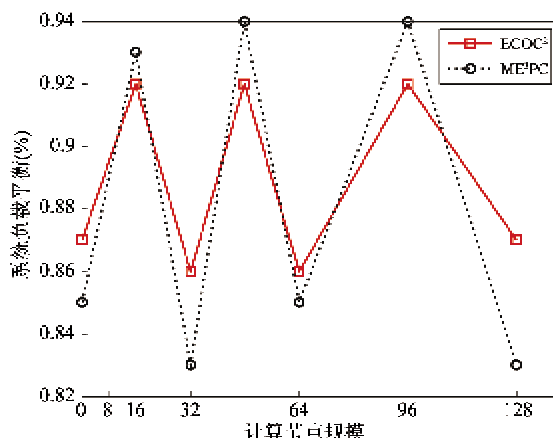


图6 系统负载平衡示意图

4.5 实验总结

在任务规模变化范围0-120、6个区域节点的电价以及计算节点规模在128时,ECOC²算法比ME³PC算法在不同任务规模下的平均能耗提高了约为6.27%。随着计算任务的逐渐增多,这部分差距

逐渐变小,而成本比ME³PC算法节约了约14.79%,平均功率变化逐渐相等;在任务规模为80,数据中心规模为64时,当约束成本在闭区间[500,800]变化时,ECOC²算法平均能耗逐渐减少并趋于平稳;在数据中心规模变化范围为0-128,电价在6个区域节点的变化,任务规模不变时,ECOC²算法的系统平均响应时间以及系统负载平衡等都要明显低于ME³PC算法。

通过实验结果可以得出,ECOC²在调度过程中优先选择了远端成本较低的空闲节点,增加了少量的调度开销,但是同时体现出了成本上的巨大优势,不但解决了能耗浪费严重的问题,同时提高了系统效率,并使节约成本成为可能。

5 结论

本文对云数据中心资源使用率、网络传输能耗以及系统效率进行了深入研究,进而提出了云计算在成本约束下的能耗资源调度算法,并通过定义世界范围的电价,利用网络负载率进行了实验验证。由实验结果可以得出,随着计算任务数量的增加,本文提出的算法由于成本的约束,在平均执行功率增加过程中避免了系统瞬时能耗增加过快的缺点。当在任务量较小时,通过增加少量传输能耗,节省大量的服务成本;而在任务量较大时,通过任务分层算法进行任务的相关性分层调度,增加了任务并行特性,降低了数据中心计算节点空闲概率,同时由于任务的相关性降低了数据中心间数据传递能耗,降低了系统响应时间及系统负载率。可以在节约能耗的同时节约成本,同时提高系统的性能,这就为云计算服务商提供了更好的选择。

在未来的工作中,我们拟设计节省更多空闲的虚拟机技术来降低数据中心的能耗,通过研究优化的副本部署算法来满足虚拟机关闭以及迁移带来的数据变化,进而达到降低系统能耗目的同时提高系统的效率。

参考文献

- [1] Silva M, Morais H, Vale Z. An integrated approach for distributed energy resource short-term scheduling in smart grids considering realistic power system simulation. *Energy Conversion and Management*, 2012, 64 (3): 273-288
- [2] Young C L, Albert Y Z. Energy efficient utilization of resources in cloud computing systems. *Journal of Supercomputing*, 2012, 60 (4): 268-280

- [3] Jayant B, Robert W A, Kerry H, et al. Green cloud computing: balancing energy in processing. *Storage and Transport*, 2011, 99(1): 149-167
- [4] Rajni L, Inderveer C. Bacterial foraging based hyper-heuristic for resource scheduling in grid computing. *Future Generation Computer Systems*, 2013, 29 (1): 751-762
- [5] Lien D, Bert V. Efficient resource management for virtual desktop cloud computing. *Journal of Supercomputing*, 2012, 62(1): 741-767
- [6] 宋杰, 李甜甜, 闫振兴. 一种云计算环境下的能效模型和度量方法. *软件学报*, 2012, 23(2): 200-213
- [7] Dmitry K, Pascal B, Samee U K. Green cloud: a packet-level simulator of energy-aware cloud computing data centers. *Journal of Supercomputing*, 2012, 62(1): 1263-1283
- [8] Anton B, Jemal A, Rajkumar B. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 2012, 28(1): 755-768
- [9] Michael C, Aameek S. Exploiting spatio-temporal tradeoffs for energy-aware MapReduce in the cloud. *IEEE transactions on computers*, 2012, 61(12): 1731-1751
- [10] Ivan R, Hariharasudhan V, Eun K L, et al. Energy-efficient thermal-aware autonomic management of virtualized HPC cloud infrastructure. *Journal of Grid Computing*, 2012, 10(1): 447-473
- [11] 苑迎春, 李小平, 王茜. 成本约束的网格 workflow 时间优化方法. *计算机研究与发展*, 2009, 46(2): 194-201
- [12] 谭一鸣, 曾国荪, 王伟. 随机任务在云计算平台中能耗的优化管理方法. *软件学报*, 2012, 23(2): 266-277

A cost constrained resource scheduling optimization algorithm for reduction of energy consumption in cloud computing

Hao Liang*, Cui Gang*, Qu Mingcheng*, Zhang Kang**

(* School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

(** School of Computer Science, University of Texas at Dallas, Dallas 75080)

Abstract

In order to solve the serious energy consumption problem in cloud computing, a cost constrained resource scheduling optimization algorithm for reduction of the energy consumption in cloud computing was proposed based on the intensive study of the resource utilization of cloud data centers, the network workloads and the real-time power. Then the electricity cost was considered according to the world time zones, and the tasks were scheduled and performed by using the load-balancing method. Based on the above researches, the task slicing algorithm (TSA) was designed to reduce the idle probability of data centers and the energy consumption of data transmission between the centers through increasing the parallelism degree and dependency of tasks. When the cost constraint was not satisfied, tasks were iteratively calculated according to the algorithm. The results of the simulating experiments show that the algorithm can significantly save the service cost while optimizing the energy consumption.

Key words: cost constraints, cloud computing, resource scheduling, energy optimization, task slicing algorithm (TSA)