Sustainable Computing: Informatics and Systems 3 (2013) 109–118

Contents lists available at SciVerse ScienceDirect

Sustainable Computing: Informatics and Systems

journal homepage: www.elsevier.com/locate/suscom

Energy-efficient task scheduling algorithms on heterogeneous computers with continuous and discrete speeds

Luna Mingyi Zhang[a,*], Keqin Li[c], Dan Chia-Tien Lo[b], Yanqing Zhang[d]

[a] Department of Computer Science, College of Engineering, Cornell University, Ithaca, NY 14853, USA
[b] Department of Computer Science and Software Engineering, Southern Polytechnic State University, Marietta, GA 30060-2896, USA
[c] Department of Computer Science, State University of New York at New Paltz, New Paltz, NY 12561, USA
[d] Department of Computer Science, Georgia State University, Atlanta, GA 30302-3994, USA

ARTICLE INFO

Article history:
Received 10 July 2012
Accepted 28 January 2013

Keywords:
Green computing
Task scheduling
Energy reduction
Power-aware methods
Pollution reduction

ABSTRACT

A large number of computing servers and personal electronic devices waste a tremendous amount of energy and emit a considerable amount of carbon dioxide, which is the major contribution to the greenhouse effect. Thus, it is necessary to significantly reduce pollution and substantially lower energy usage. Green computing techniques are utilized in a myriad of applications in energy conservation and environment improvement. New green task scheduling algorithms for heterogeneous computers with changeable continuous speeds and changeable discrete speeds are developed to reduce energy consumption as much as possible and finish all tasks before a deadline. A newly proven theorem can determine the optimal speed for tasks assigned to a computer with continuous speeds. This project seeks to develop innovative green task scheduling algorithms that have two main steps: heuristically assigning tasks to computers, and setting optimal or near-optimal speeds for all tasks assigned to each computer. Sufficient simulation results indicate that the algorithm with the best task schedule varied. Thus, two hybrid algorithms for continuous and discrete speeds are created separately to obtain the best task schedule among candidate task schedules. Potential research applications include incorporating energy-efficient software into mobile devices, sensor networks, data centers, and cloud computing systems.

1. Introduction

Green computing is an emergent technology that applies intelligent optimization algorithms and advanced computing techniques to minimize energy consumption and reduce pollution from computing resources [1–6]. It is important for various applications in power management, energy reduction, pollution control, and environment enhancement [1–10]. Specifically, minimizing energy consumption on cloud servers and significantly reducing pollution produced by computers is an imperative research problem as energy costs are rising and the use of computers is increasing.

A typical personal computer with 17-in. LCD monitor requiring 145 W, if left on every day for one year, would use around 1270 kilowatt hours (kWh) of electricity. In 2007, the Environmental Protection Agency (EPA) predicted that the total energy consumed by U.S. data centers will double by 2012 [7]. In the U.S., about 61 billion kWh were used to power data centers in 2006 ($4.5 billion). For 2010, Google's electricity consumption was about 2.26 million MWh [25]. The EPA estimates that data centers annually consume the output of 15 average-sized power plants. Also, the EPA predicts that power consumption of data centers will soon increase to 12 GW, leading to the equivalent output of 25 power plants [9].

Computer usage accounts for 2% of anthropogenic CO_2 emission. Data center activities are estimated to release 62 million metric tons of CO_2 into the atmosphere [9]. For example, Google states that one Google search may generate about 0.2 g of carbon dioxide [4]. In 2011, Google estimated that its total carbon emissions for 2010 were 1.46 million metric tons [25]. The use of 1270 kWh of electricity is enough to emit about 1720 pounds of CO_2 into the environment.

However, Google and General Electric (GE) have applied green computing techniques to save power and reduce costs [4–6,25]. The energy used for each Google search was very small, which was 0.0003 kWh [4]. Also, Google's data centers use 50% less energy than the typical data center [25]. GE saves $2.5 million a year by implementing power control methods of Windows operating system onto its computers [6]. Nonetheless, computers can be further improved to become more energy-efficient, which is essential. The fundamental reason for our research in task scheduling on computers is to minimize computing energy consumption and consequently substantially reduce pollution produced by computers. Therefore, it is important to develop effective green computing

* Corresponding author. Tel.: +1 678 654 7090.
E-mail addresses: mingyiluna@yahoo.com, lmz22@cornell.edu (L.M. Zhang), lik@newpaltz.edu (K. Li), clo@spsu.edu (D.C.-T. Lo), yzhang@cs.gsu.edu (Y. Zhang).

2210-5379/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.suscom.2013.01.002

techniques to considerably lower energy usage on heterogeneous cloud servers. Thus, this project is critical and beneficial for the global community.

The problem of scheduling many independent tasks in a heterogeneous distributed computing system has been studied. A new green scheduling algorithm for saving energy in cloud computing was proposed [11]. However, in some cases, energy consumption was not considered as a factor that should be minimized [12]. In [13], the advantages for having task assignments were discussed, and the voltage level was considered as a factor instead of the speed level for each processor. A linear combination of the minimum and maximum processor frequencies was used to decrease energy consumption [14] and optimal frequencies were analyzed in [15]. In [16], variable processor speeds were considered, and idle intervals were exploited for power minimization. The algorithms incorporated dynamic voltage and frequency scaling to save energy [17]. Task scheduling algorithms were proposed to lower energy consumption by using shared slack reclamation on variable voltage/speed processors for task sets with precedence constraints and those without precedence [18]. In [22], two new heuristic energy-efficient task scheduling algorithms were proposed; tasks are assigned from the front in a task queue to a computer with minimum energy in the computer queue and then each computer's speed, with constraints, is adjusted to reach an overall minimal energy usage. Conventional approaches utilize maximum speeds, which is not energy-efficient. In [23], six innovative green task scheduling algorithms were developed; as many tasks as possible are assigned to a cloud server with lowest energy and then the speeds, with constraints, are lowered for all assigned tasks, assuming that the speeds are continuous, not discrete. The minimum total energy consumption for a computer with multiple identical processors occurred when all independent tasks were executed with the same power (or at the same speed) [1].

This paper focuses on developing novel green task scheduling algorithms for completing sequential tasks on heterogeneous servers with variable continuous and discrete speeds to minimize energy consumption via energy consumption parameters in a cloud computing environment with a certain deadline. For continuous speeds, it focuses on real applications with heterogeneous processors with different parameters. Also, speed constraints for processors were not taken into account [1]. A new method in this project uses various speed constraints for heterogeneous computers to better model real systems. For discrete speeds, it focuses on real applications with heterogeneous processors with different parameters using discrete speeds. For example, a microcontroller such as MSP430 has discrete speeds (i.e. clock frequencies). To better model real systems, new green task scheduling algorithms for heterogeneous computers with variable discrete speeds and energy consumption parameters are developed to reduce energy consumption as much as possible and finish all tasks before a certain deadline.

The rest of the paper is organized as follows. In Section 2, all of the definitions and parameters are given. In Section 3, the optimization problem is given, and new energy-efficient task scheduling algorithms with continuous speeds are proposed. In Section 4, the optimization problem is given, and new energy-efficient task scheduling algorithms with discrete speeds are proposed. In Section 5, the simulations and performance analysis are discussed. In Section 6, conclusions and future works are given.

2. Definitions and parameters

Let n computers be used to finish m tasks by the deadline time T (s) where $m \geq n$. Assume that m_i tasks P_k^i for $k = 1, 2, \ldots, m_i$ are executed on computer i for $m = \sum_{i=1}^{n} m_i$. A changeable speed (discrete or continuous) for task P_k^i is denoted as S_k^i for $i = 1, 2, \ldots, n$, and $k = 1, 2, \ldots, m_i$. The speed is defined as the number of instructions per second. The number of instructions of task P_k^i is denoted as R_k^i. The execution time for task P_k^i on computer i is R_k^i/S_k^i, so the total execution time for m_i tasks P_k^i on computer i is $T_i = \sum_{k=1}^{m_i} R_k^i/S_k^i$. From [1], the energy (in J) for P_k^i on computer i is $E_k^i = C_i R_k^i [S_k^i]^{\alpha_i - 1}$ where C_i is a constant, $\alpha_i = 1 + (2/\phi_i) \geq 3$ for $0 < \phi_i \leq 1$, $i = 1, 2, \ldots, n$, and $k = 1, 2, \ldots, m_i$. The total energy is $E = \sum_{i=1}^{n} \sum_{k=1}^{m_i} C_i R_k^i [S_k^i]^{\alpha_i - 1}$. Let $\gamma_k^i = C_i [S_k^i]^{\alpha_i - 1}$. For a speed S_k^i, γ_k^i is called the energy slope (a constant for each computer and task). Then, the total energy is $E = \sum_{i=1}^{n} \sum_{k=1}^{m_i} \gamma_k^i R_k^i$.

3. Energy-efficient task scheduling algorithms with continuous speeds

The problem is to minimize $E = \sum_{i=1}^{n} \sum_{k=1}^{m_i} \gamma_k^i R_k^i$ with constraints: $1 \leq m_i \leq m - n + 1$, $m = \sum_{i=1}^{n} m_i$, $\sum_{k=1}^{m_i} R_k^i/S_k^i \leq T$ and $0 < a_i \leq S_k^i \leq b_i$, where a_i is the minimum speed and b_i is the maximum speed of computer i, for $i = 1, 2, \ldots, n$, and $k = 1, 2, \ldots, m_i$.

For a single computer case, the superscript i is omitted, so the energy for one computer is then $E = \sum_{k=1}^{m} C R_k [S_k]^{\alpha - 1}$, where R_k denotes the number of instructions and S_k denotes the execution speed (the number of instructions executed per second) for the kth task. Thus, the problem is to minimize $E = \sum_{k=1}^{m} C R_k [S_k]^{\alpha - 1}$ subject to $\sum_{k=1}^{m} R_k/S_k \leq T$ and $0 < a \leq S_k \leq b$ for $k = 1, 2, \ldots, m$, where a is the minimum speed and b is the maximum speed of the computer [22,23]. The minimal E occurs when each of the computers consumes energy at its minimal level to finish its tasks on time.

Theorem 1. *After m tasks with the same speed* $\bar{S}$ *for* $0 < a \leq \bar{S} \leq b$ *are successfully assigned to a computer for* $\sum_{k=1}^{m} R_k/\bar{S} < T$, *the minimal total energy consumption is* $E = C[S^*]^{\alpha - 1} \sum_{k=1}^{m} R_k$ *for* $\alpha = 1 + 2/\phi \geq 3$ *and* $0 < \phi \leq 1$ *when all tasks are executed with the same optimal speed* S^* *where* $S^* = \text{maximum}\,(a, \sum_{k=1}^{m} R_k/T)$ [22].

Proof. Because $\sum_{k=1}^{m} R_k/\bar{S} < T$, speeds S_k for $0 < a \leq S_k \leq b$ and $k = 1, 2, \ldots, m$ can be optimized to minimize the total energy consumption $E = \sum_{k=1}^{m} C R_k [S_k]^{\alpha - 1}$ [22].

The Lagrangian function [24] is defined as $L = \sum_{k=1}^{m} C R_k [S_k]^{\alpha - 1} - \lambda (T - \sum_{k=1}^{m} R_k/S_k)$.

$\partial L/\partial S_k = C R_k (\alpha - 1)[S_k]^{\alpha - 2} - \lambda (R_k/S_k^2) = 0$ for $k = 1, 2, \ldots, m$. The first order equalities are $\lambda (T - \sum_{k=1}^{m} R_k/S_k) = 0$ for $k = 1, 2, \ldots, m$. Also, $\sum_{k=1}^{m} R_k/S_k \leq T$, $0 < a \leq S_k \leq b$, and $\lambda \geq 0$ for $k = 1, 2, \ldots, m$. □

Case 1. When $\lambda > 0$, we have $C R_k (\alpha - 1)[S_k]^{\alpha - 2} - \lambda (R_k/S_k^2) = 0$. Thus, $S_k = (\lambda / C(\alpha - 1))^{1/\alpha} = S$ where S is a constant speed for $k = 1, 2, \ldots, m$. Because $\partial^2 L/\partial^2 S_k \big|_{S_k = S} = C R_k (\alpha - 1)(\alpha - 2)[S]^{\alpha - 3} + 2\lambda (R_k/S^3) > 0$ for $\alpha = 1 + 2/\phi \geq 3$ and $0 < \phi \leq 1$, we obtain the minimum value of $E_k = C R_k [S_k]^{\alpha - 1}$ when $S_k = S$. We have $T - \sum_{k=1}^{m} R_k/S = 0$, so $S = \sum_{k=1}^{m} R_k/T$. Since $\sum_{k=1}^{m} R_k/\bar{S} < T$, $\sum_{k=1}^{m} R_k/T < \bar{S} \leq b$. Hence, $0 < S < \bar{S} \leq b$.

Case 1.1. If $0 < a < S < \bar{S} \leq b$, then the optimal speed is $S = \sum_{k=1}^{m} R_k/T$.

Case 1.2. If $0 < S \leq a$, then $0 < \sum_{k=1}^{m} R_k/T \leq a$. Thus, $0 < \sum_{k=1}^{m} R_k/a \leq T$, meaning that every task can be executed at the minimum speed a so the optimal speed is $S = a$.

From Cases 1.1 and 1.2, the optimal speed is $\text{maximum}\,(a, \sum_{k=1}^{m} R_k/T)$.

Case 2. When $\lambda = 0$, we have $C R_k (\alpha - 1)[S_k]^{\alpha - 2} = 0$. Thus, $S_k = 0$ which is invalid since $0 < a \leq S_k \leq b$.

From Cases 1 and 2, the optimal speed is $S^* = \text{maximum}\,(a, \sum_{k=1}^{m} R_k/T)$ [22].