Future Generation Computer Systems 74 (2017) 119–131

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

# Slack allocation algorithm for energy minimization in cluster systems

Yikun Hu [a], Chubo Liu [a,b], Kenli Li [a,b,c,*], Xuedi Chen [a], Keqin Li [a,d]

[a] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[b] National Supercomputing Center in Changsha, Hunan, China
[c] CIC of HPC, National University of Defense Technology, Changsha 410073, China
[d] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

## HIGHLIGHTS

- An energy-aware scheduling algorithm called EASLA is proposed.
- The main idea of the EASLA is to distribute slacks to tasks.
- The maximum set of independent tasks is involved.

## ARTICLE INFO

*Article history:*
Received 12 March 2015
Received in revised form
11 July 2016
Accepted 31 August 2016
Available online 11 September 2016

*Keywords:*
Cluster computing
Directed acyclic graph
Dynamic voltage/frequency scaling
Energy aware scheduling
Service level agreement

## ABSTRACT

Energy consumption has been a critical issue in high-performance computing systems, such as clusters and data centers. An existing technique to reduce energy consumption of applications is dynamic voltage/frequency scaling (DVFS). In this paper, we present a novel algorithm called EASLA for energy aware scheduling of precedence-constrained applications in the context of Service Level Agreement (SLA) on DVFS-enabled cluster systems. Due to the dependencies among tasks and makespan extension, there may be some underused slacks. The main idea of the EASLA algorithm is to distribute each slack to a set of tasks and scale frequencies down to try to minimize energy consumption. Specifically, it first finds the maximum set of independent tasks for each task, and then iteratively allocates each slack to the maximum independent set whose total energy reduction is the maximal. Randomly generated graphs and two real-world applications are tested in our experiments. The experimental results show that our scheduling algorithm can save up to 22.68% and 12.01% energy consumption compared with the GreedyDVS and EvenlyDVS algorithms respectively in homogeneous environments, and 12.33% energy consumption compared with the EES algorithm in heterogeneous environments.



## 1. Introduction

With the development of high-performance computing (HPC), more and more energy has been consumed by large-scale cluster systems, which raises various economical, environmental and system availability concerns [1,2]. According to statistics, the total energy consumption in information and communication technology (ICT) industry is about 868 billion, which is roughly 5.3% of the world total energy consumption in 2008. According to the current increasing trend, by 2025, the total energy consumption in ICT industry will be four-fold increase on 2008 levels [3,4]. Additionally, energy consumption translates into high carbon emissions and results in high cooling cost [5]. Furthermore, the temperature of computing systems will ascend sharply due to large amount of energy consumption. Evidence shows that the expected failure rate doubles for every 10 °C increased temperature [6]. This greatly affects the system reliability and availability, and eventually does harm to system performance. Therefore, it is important to study the strategy of reducing energy consumption in high-performance computing.

Energy awareness for parallel applications has been a growing concern for a decade. System-level power-saving methods including Dynamic Power Management (DPM) [7] and Dynamic Voltage/Frequency Scaling (DVFS) [1,8,9] have been investigated and

* Corresponding author at: College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China.
*E-mail addresses:* yikunhu@hnu.edu.cn (Y. Hu), liuchubo@hnu.edu.cn (C. Liu), lkl@hnu.edu.cn (K. Li), chenxuedi@hnu.edu.cn (X. Chen), lik@newpaltz.edu (K. Li).

http://dx.doi.org/10.1016/j.future.2016.08.022
0167-739X/

developed. The core idea of DPM is to turn off the idle computing nodes. However, this method is only suitable for the situation in which most of the idle periods are very long. For the scheduling of a precedence-constrained application, the idle time between the execution of two tasks is often short and DPM is not suitable. Different from DPM, the DVFS technique can assign different frequencies to each task, which gives us a useful way to minimize energy consumption of applications.

In this paper, we use DVFS technique to reduce energy consumption while considering service level agreement (SLA), which is measured by two metrics: *makespan* and energy consumption. Obviously, there exists a tradeoff between these two metrics. In our architecture model, a customer can negotiate with a service provider about Quality of Service (QoS) of his application by determining *makespan* extension rate. For example, the customer agrees to accept additional 10% of *makespan* to reduce more energy consumption. Under this situation, we propose an energy-performance tradeoff scheduling algorithm (EASLA), which uses DVFS technique to try to minimize energy consumption. Our scheme addresses the scheduling in a different way compared with most of the relevant algorithms for discrete voltage clusters. First, it finds the maximum set of independent tasks for each task to increase parallelism for using slacks. Second, it also allocates slacks to non-critical tasks to minimize energy consumption instead of only the critical ones.

The main contributions of this paper are summarized as follows:

- Unlike the most existing studies for DAG applications, we try to allocate a slack to tasks belonging to the maximum independent set of a task.
- We develop a novel energy-aware scheduling algorithm called EASLA, which can be adapted for a wide range of DAG applications.
- We carry out extensive experiments to verify the effectiveness of our algorithm under different circumstances: randomly generated graphs as well as graphs of real-world problems with various characteristics, homogeneous resources in terms of speed and energy consumption, or heterogeneous resources.

## 2. Related work

Many works have been investigated for energy reduction based on continuous DVFS assumption. Zhang et al. introduced a linear programming (LP) based formulation to solve the continuous voltage DVFS [6]. Kang et al. developed a path based DVFS algorithm to try to minimize energy requirement while meeting the deadline constraints at the same time [10]. Aupy et al. proposed several polynomial time scheduling algorithms, which try to minimize energy consumption under the condition of a prescribed *makespan* bound and a reliability threshold [11]. Baskiyar et al. used the heterogeneous earliest finish time (HEFT) heuristic as an initial scheduling algorithm to minimize *makespan*, then voltage scaling was performed to reduce power consumption without performance degradation [12,13]. However, these algorithms focus on continuous frequency and voltage systems. For discrete DVFS consideration, the problem becomes more complex.

For the discrete frequency and voltage situation, the authors in [14–18,1,8,19] considered the scheduling and proposed some algorithms. Kimura et al. proposed an energy reduction algorithm, in which the suitable frequency among the discrete set of processor's frequencies is chosen for each task according to its slack time [14]. Rizvandi et al. introduced a slack reclamation algorithm which uses a linear combination of the maximum and minimum processor frequencies to decrease energy consumption [15]. Chowdhury et al. allocated the slack time to tasks according to the decreasing order of their finish times [16]. Wang et al. evenly distributed the free slack obtained by makespan extension to critical tasks, and then distributed the slack occurred due to dependencies among tasks to non-critical tasks [17]. The core idea of all these algorithms is to utilize idle time slots (slack) to lower down supply voltage (frequency/speed). However, most of them ignore the variable energy profiles of tasks on different processors during slack allocation or do not take into account the non-critical tasks for using idle time slots. Due to the dependencies among tasks, there may be many situations in which the sum of energy reduction of several tasks (i.e., a task set that can execute in parallel whether the tasks belonging to the set are critical or not) can be higher than the energy reduction of critical tasks. Our scheme addresses the above issues and effectively allocates the slack to gain more energy reduction for precedence-constrained applications in discrete frequency and voltage systems.

Many studies on cluster and cloud computing use DVFS technique in the context of Service Level Agreement (SLA) [20–25]. SLA is an agreement between a service provider and customers, specifying the level of delivered services [26,27]. It can be decided by many different QoS parameters, such as deadline [21], response time [22], and so on. Wu et al. proposed an algorithm to assign resources for tasks obeying the SLA [20]. In this algorithm, the tasks are deployed to fewer computing nodes with lower SLA level, and then appropriate frequencies are selected for the computing nodes via DVFS. Kim et al. proposed DVFS scheduling algorithms for bag-of-tasks applications to try to minimize energy consumption with deadline constraints [21]. Gao et al. proposed a dynamic resource management scheme which takes advantage of both DVFS and server consolidation to achieve energy efficiency while satisfying response-time-based SLAs [22]. In our work, we also consider the energy reduction problem in the context of SLA.

The remainder of this paper is organized as follows. In Section 3, we introduce the system model and formulate the scheduling problem. In Section 4, we describe the proposed EASLA algorithm and discuss its performance. Section 5 gives the simulation results compared with GreedyDVS and EvenlyDVS algorithms in homogeneous environments, and the EES algorithm in heterogeneous environments. Finally, Section 6 presents the conclusion and future work.

## 3. System model and problem formulation

In this section, we introduce the system model and formulate the scheduling problem.

### 3.1. Architecture model

In this subsection, we introduce the architecture model of our scheduling environment. As shown in Fig. 1, the architecture model, which is similar to [17], consists of three layers: user layer, scheduling layer, and resource layer. The user layer is in charge of submitting tasks. The scheduling layer, which is used to assign tasks appropriately, consists of two components—a *makespan* and energy estimator and our scheduling strategy. The resource layer consists of multiple processing elements (PEs) and is responsible for task execution. The needed PEs are allocated to users exclusively for some duration of time.

Next, we describe the task execution process in the scheduling model mentioned above. First, tasks with computation time and dependency relations, and the needed number of PEs are submitted by users. Second, the information of tasks and PEs are sent to the scheduler and it uses an estimator to evaluate the *makespan* and energy consumption. Third, the estimator gives the evaluated results back to the user. Fourth, the user negotiates with the service provider about the *makespan* extension rate, which presents the