Future Generation Computer Systems 78 (2018) 257–271

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Dynamic energy-aware scheduling for parallel task-based application in cloud computing

Fredy Juarez [a,b], Jorge Ejarque [a,*], Rosa M. Badia [a,c]

[a] Barcelona Supercomputing Center (BSC), Workflows and Distributed Computing Group, Barcelona, 08034, Spain
[b] Instituto Tecnológico Superior de Álamo Temapache, Xoyotitla, Veracruz, 92730, Mexico
[c] Artificial Intelligence Research Institute (IIIA), Spanish National Research Council (CSIC), Spain

HIGHLIGHTS

- Energy-aware run-time scheduler for task-based applications.
- Model for estimating the application Energy consumption.
- Methodology to automatically generate the required power consumption profile.
- Multi-heuristic resource allocation algorithm to get solutions in polynomial time.
- Energy saving/performance trade-off evaluation for different scenarios.

ARTICLE INFO

Article history:
Received 1 December 2015
Received in revised form
20 April 2016
Accepted 23 June 2016
Available online 5 July 2016

Keywords:
Distributed computing
Cloud computing
Green computing
Task-based applications
Energy-aware scheduling
Multi-heuristic resource allocation

ABSTRACT

Green Computing is a recent trend in computer science, which tries to reduce the energy consumption and carbon footprint produced by computers on distributed platforms such as clusters, grids, and clouds. Traditional scheduling solutions attempt to minimize processing times without taking into account the energetic cost. One of the methods for reducing energy consumption is providing scheduling policies in order to allocate tasks on specific resources that impact over the processing times and energy consumption. In this paper, we propose a real-time dynamic scheduling system to execute efficiently task-based applications on distributed computing platforms in order to minimize the energy consumption. Scheduling tasks on multiprocessors is a well known NP-hard problem and optimal solution of these problems is not feasible, we present a polynomial-time algorithm that combines a set of heuristic rules and a resource allocation technique in order to get good solutions on an affordable time scale. The proposed algorithm minimizes a multi-objective function which combines the energy-consumption and execution time according to the energy-performance importance factor provided by the resource provider or user, also taking into account sequence-dependent setup times between tasks, setup times and down times for virtual machines (VM) and energy profiles for different architectures. A prototype implementation of the scheduler has been tested with different kinds of DAG generated at random as well as on real task-based COMPSs applications. We have tested the system with different size instances and importance factors, and we have evaluated which combination provides a better solution and energy savings. Moreover, we have also evaluated the introduced overhead by measuring the time for getting the scheduling solutions for a different number of tasks, kinds of DAG, and resources, concluding that our method is suitable for run-time scheduling.

1. Introduction

Recent studies [1,2] have estimated that around 1.5%–2.0% of the total energy consumption is consumed by data centers, and this energy demand is growing extremely fast due to the popularization of Internet services and distributed computing platforms such as clusters, grids, and clouds. Regarding the efficiency of data centers, studies have concluded that, in average, around 55% of the energy consumed in a data center is consumed by the computing system and the rest is consumed by the support system such as cooling, uninterrupted power supply, etc. For that reason, green cloud

* Corresponding author.
E-mail address: jorge.ejarque@bsc.es (J. Ejarque).

http://dx.doi.org/10.1016/j.future.2016.06.029

computing is essential for ensuring that the future growth of cloud computing is sustainable [3].

There are several ways to reduce the energy consumed by an application when executed on a distributed platform: It includes the usage of low-power processor architectures or dynamic voltage frequency scaling (DVFS) [4], re-design of algorithms using energy-efficient patterns in compilers [5] or changing the scheduling policies for task-based applications on the available resources [6]. Traditionally, scheduling techniques have tried to minimize the total execution time of an application(makespan—$C_{\max}$) [7] without worrying about the energy consumed. However, there is a trade-off between energy consumed and the execution time, and sometimes increasing the performance for a faster execution implies a higher energy consumption.

The aim of our work is to offer resource providers and end-users more options for executing task-based applications in an energy conscious manner, giving the possibility of reducing energy consumption without a significant increase in total execution time or reducing the total execution time without a significant increase in energy consumption. In this paper, we present an energy-aware scheduling system for task-based applications. To decide which is the best scheduling solution according to the energy consumed, we propose a model for estimating the energy consumed by the application for a given application and a resource power consumption profile.

Since task allocation on distributed computing resources is a well known NP-hard problem in the general form [8], due to the time limitation required for run-time schedulers, the implementation of large-time optimization algorithms is not suitable. For a real time scheduler it is convenient to develop heuristic techniques for sub-optimal solutions, in order to build a scheduling algorithm that runs in polynomial-time without performing exhaustive search. Therefore, we propose the use of a multi-heuristic resource allocation (MHRA) that is essentially a faster local search algorithm for partial solutions. The algorithm is divided into two phases: the first phase combines a set of heuristic rules for ranking an eligible group of parallel tasks for a given Direct Acyclic Graph (DAG), based on the amount of data transfers, number of task predecessors or successors and execution time. The second phase combines a set of importance factors for the resource allocation algorithm that are used to determine which is the best position in the cloud for a specific task that minimizes energy (E_{flow}) and makespan ($C_{\max}$). The algorithm provides good real-time scheduling solutions in an affordable time scale.

The proposed scheduler has been designed to be applied to the COMP Superscalar (COMPSs) framework [9,10]. It provides an infrastructure-agnostic task-based programming model, which facilitates the development of parallel applications in distributed computing platforms. Developers can program their applications in a sequential fashion and without caring about the details of the underlying infrastructure. They just need to identify the tasks, which are the methods of the applications, to be executed in the distributed platform. At run-time, COMPSs detects data dependencies between tasks creating a DAG. Once the DAG is created, the COMPSs runtime will use the energy-aware scheduler for allocating and executing the tasks on the available computing resources in order to minimize energy or makespan.

The scheduler has been tested with different kinds of DAGs generated at random as well as on real COMPSs applications: embarrassingly parallel (EB), parallel reduction (PR), parallel increase/reduction (PIR), matrix multiplication (MT). Using three different size instances and importance factors. We have evaluated which combination of MHRA provides a better solution and energy savings and the execution time in each case, and the effect on the cloud elasticity. Moreover, we have also evaluated the introduced overhead by measuring the time for getting the scheduling solutions for a different number of tasks, kinds of DAG, and resources, concluding that it is suitable for run-time scheduling.

The rest of the paper is organized as follows: First, Section 2 presents the related work in energy-aware scheduling and Section 3 gives an overview of the COMPSs framework. Afterwards, Section 4 formulates the energy-aware scheduling problem, and the model used to estimate the application energy consumption, the profiling methodology and the multi-heuristic resource allocation algorithm are presented in Sections 5–7, respectively. In Section 8, we present the experiments performed to evaluate the proposed scheduler. Finally, Section 9 draws the conclusions and proposes guidelines for future work.

2. Related work

Traditional task scheduling algorithms for distributed platforms such as clusters, grids, and clouds, focus in minimizing the execution time [11,12] without considering energy consumption. Regarding specific work on energy-aware scheduling two main trends can be found in the literature: (1) pure scheduling software and (2) combined scheduling hardware/software. For combined scheduling, a commonly used technique is taking profit of the Dynamic Voltage Frequency Scaling (DVFS) feature which enables processors to reduce the energy consumption. By using DVFS, processors can run at different voltage, impacting on the frequency and energy consumption.

In [13] the authors present a scheduling heuristic for reducing power consumption of precedence-constrained parallel tasks in a cluster with DVFS, the model is proposed for applications that have slack time for non-critical jobs while they scale down supply voltage for reducing energy consumption and extend execution time of jobs. The model considers homogeneous nodes as processing elements (PEs) with the same processing speed and uses green SLA negotiation in order to accept a tolerable performance loss. In our work, we take into account makespan and energy consumption in a similar way, but in addition we take into account the cloud environment and setup times for creating and destroying VMs; in contrast for reduced energy consumption, we do not use DVFS, because our model minimizes energy consumption while striving to maintain application performance.

Another approach for combining DVFS in scheduling is proposed in [14]. In this case, the authors propose a scheduling algorithm in order to reduce power consumption by applying DVFS for enabling processors to run at low frequencies and low voltages. It is only applicable for those applications in which the performance is not important or can run under certain threshold frequency. The algorithm complies with the Service Level Agreement and obeys the SLA to assign resources for the job given to the consumers. The algorithm takes into account the maximum $Job(F_{\max})$ and minimum $Job(F_{\min})$ frequencies given for each job and the multiple servers S_i that are running at maximum $S_i(F_{\max})$ and minimum $S_i(F_{\min})$ frequencies. For specific jobs, the algorithm selects a server that runs between, $(F_{\min}, F_{\max})$ and guarantees the execution performance of the job while ensuring that the job does not overuse resources. In contrast, our algorithm detects parallelism at run-time by analyzing the task dependencies of a job (application) and allocates these tasks on specific VMs that have impact on the total energy and makespan.

Beloglazov et al. [6] propose an energy-aware allocation heuristic for client applications over the data center resources, considering QoS expectations. The green Cloud architecture proposed here, it is a power model that divides the energy level consumption in an idle server and running a server with the CPU utilization controlled by DVFS for different frequency and voltage utilization. This model proposes VM allocation divided in