Sustainable Computing: Informatics and Systems 4 (2014) 252–261

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

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

A hierarchical approach for energy-efficient scheduling of large workloads in multicore distributed systems

Bernabé Dorronsoro [a,*], Sergio Nesmachnow [b], Javid Taheri [c], Albert Y. Zomaya [c], El-Ghazali Talbi [a], Pascal Bouvry [d]

[a] University of Lille, France
[b] Universidad de la República, Uruguay
[c] The University of Sydney, Australia
[d] University of Luxembourg, Luxembourg

A R T I C L E   I N F O

Article history:
Received 1 March 2014
Received in revised form 4 July 2014
Accepted 11 August 2014

Keywords:
Energy efficiency
Workflows
Multicore
Scheduling heuristics

A B S T R A C T

This article presents a two-level strategy for scheduling large workloads of parallel applications in multicore distributed systems, taking into account the minimization of both the total computation time and the energy consumption of solutions. Nowadays, energy efficiency is of major concern when using large computing systems such as cluster, grid, and cloud computing facilities. In the approach proposed in this article, a combination of higher-level (i.e., between distributed systems) and lower-level (i.e., within each data-center) schedulers are studied for finding efficient mappings of workflows into the resources in order to maximize the quality of service, while reducing the energy required to compute them. The experimental evaluation demonstrates that accurate schedules are computed by using combined list scheduling heuristics (accounting for both problem objectives) in the higher level, and ad-hoc scheduling techniques to take advantage of multicore infrastructures in the lower level. Solutions are also evaluated with two user- and administrator-oriented metrics. Significant improvements are reported on the two problem objectives when compared with traditional load-balancing and round-robin techniques.

1. Introduction

Nowadays, data-center facilities typically host a large number of com-putational resources. They usually include high performance clusters, large storage systems, and/or components of large grids or cloud systems. In any case, they are always composed of highly powerful computational resources (racks of computers, storage disks, routers, etc.) with increasing energetic demands. Therefore, energy consumption has become a major concern in data-centers [1].

There are different techniques aiming at reducing the energy consumption of data-centers [3,4], from low-level hardware solutions to high-level software methods, which are different according to the kind of infrastructure considered. All the existing sustainable (i.e., energy-aware) techniques are in conflict with the performance of the system, that can be quantified in terms of traditional scheduling metrics such as the makespan (i.e., the finishing time of the last scheduled task) or other Quality of Service (QoS) metrics. This means that, for an optimal solution, increasing its performance leads to an increase of the energy consumption, while lowering the latter involves worsening the former. This is the reason why multi-objective formulations are needed for accurate capturing all the features of the data-centers planning problem.

In this work, we focus on solving the problem of finding appropriate mappings of workflows into a set of available computing resources, in order to reduce the energy required to compute the tasks, while maximizing the quality of service too. The platform we target here is a distributed data-center, composed by a number of clusters that might be geographically distributed. Specifically, we target the reduction of the required execution time and energy consumption demanded to run very large sets of workflows in large-scale computing centers, typically composed by several dedicated clusters. This is the architecture of modern high performance and distributed computing systems, including big supercomputers, high performance computing centers, and cloud infrastructures, among others. In such systems, the energy consumption of the processors is a major concern, and reducing it will also allow lowering the cooling system operational cost too [2].

The proposed strategy consists in using a hierarchical two-level approach for scheduling static batches with a large number of workloads composed by tasks with dependencies (characterized

* Corresponding author. Tel.: +352 4666445374

http://dx.doi.org/10.1016/j.suscom.2014.08.003
2210-5379/

as DAGs) on an infrastructure that gathers large distributed data-centers composed by a heterogeneous set of clusters of multicore processors. The higher-level scheduler decides the mapping between jobs and data-centers, while the lower-level scheduling methods are applied to schedule each job within each data-center. We target the minimization of makespan, the global energy consumption, and the penalizations due to everdue deadlines associated to the submitted jobs.

Our main contributions in this work are: (1) defining and tackling a novel multi-objective problem for energy-efficient scheduling of large sets of workflows in distributed data-centers, (2) designing a hierarchical two-level scheduler that allows dividing the problem into a number of simpler and smaller subproblems, and (3) the evaluation and comparison of 16 different variants of the scheduler, by using ad-hoc heuristics based on combining both the makespan and energy consumption of solutions. The experimental evaluation demonstrates that accurate solutions are computed by the best performing schedulers, allowing the planner to achieve improvements of up to 46.8% in terms of makespan and 29.0% over a typical round-robin-based strategy. We propose a static solution to find the appropriate mapping, but it is suitable for real scenarios thanks to the extreme speed of our optimizers (they take around tenths of a second). Additionally, the proposed schedulers can be easily be extended to the dynamic case by implementing some dynamic heuristic in the higher level [5].

The paper is structured as follows. We present in the next section the main related works about energy-aware scheduling on data-centers. Section 3 presents the definition and the formulation of the proposed multi-objective scheduling problem. The software methods designed to tackle the problem are described in Section 4. The experimental analysis is reported in Section 5, including an exhaustive analysis of the numerical results for the proposed schedulers when solving a large benchmark set of problem instances composed of 1000 jobs with up to 132 tasks. Finally, Section 6 summarizes the conclusions of the research and proposes some lines for future work.

2. Related work

There are many works in the literature dealing with multi-objective scheduling problems for computational grids, targeting different performance objectives as makespan and flow-time [6,7], considering robustness of the solutions to optimize [8,9], minimizing scheduling cost for data transfer and processing of services [10–13], to name a few.

Energy efficiency is another important focus on multi-objective scheduling in recent approaches. Two main optimization strategies are established for energy-aware scheduling: independent and simultaneous. In independent approaches, a *best-effort* scheduling algorithm is either combined with or followed by a slack reclamation technique. In these techniques, because energy and performance constraints are assumed independent, new or existing scheduling algorithms – usually aimed to optimize performance – are adopted to become energy-efficient too [14,15]. The Maximum-Minimum-Frequency DVFS (MMF-DVFS) is among the most efficient algorithms for slack reclamation based execution of tasks. The authors proposed an approach to reclaim slack times of tasks by linear combination of the processor highest and lowest frequencies, and they mathematically calculated/proved what combination of frequencies should be used to minimize the energy consumption for executing a task through slack reclamation [15].

In simultaneous approaches, both QoS metrics and energy saving considerations are simultaneously targeted. In this case, the problem is modelled as a multi-constrained, biobjective optimization problem where the goal is to find Pareto optimal schedules; i.e., no scheduling decision can strictly dominate the other ones with lower makespan and energy consumption at the same time. Lee and Zomaya [16] studied a set of DVS-based heuristics to minimize the weighted sum of makespan and energy; the heuristics were modified to optimize both objectives. Because each scheduling decision could be confined/trapped to local minima, their algorithm included a makespan conservative local search technique to slightly modify scheduling decisions only when they do not increase energy consumption for executing jobs. Later on, Mezmaz et al. [17] improved the previous work by proposing a parallel bi-objective hybrid GA for the same objectives where the running time for getting a resolution is significantly reduced. The parallel model was based on the cooperative approach of the island model for parallel EAs combined with a multi-start parallel model using the farmer-worker paradigm. Pecero et al. [18] proposed a bi-objective algorithm with two phases based on the Greedy Randomized Adaptive Search Procedure (GRASP). During the first phase, a greedy evaluation function builds a feasible solution. This solution is then processed in the second phase – by a local search DVS-aware bi-objective algorithm – to not only improve its quality, but also to generate a set of Pareto solutions.

Kim et al. [19] addressed the task scheduling problem considering priority and deadline constrained in ad-hoc grids. They assumed batteries to have limited capacity that are equipped with DVS-enabled power management systems. A resource manager was designed to exploit the heterogeneity of tasks while managing the energy. Duplication technique is another technique that is usually adopted for finding reliable schedules of parallel tasks that minimize the overall completion time of applications. These techniques however usually yield large additional energy consumption and resources waste due to redundant computations. By studying several online and batch mode dynamic heuristics based on MinMin, Luo et al. [33] showed that batch mode dynamic scheduling outperforms online approaches though it requires significantly more computation time too. Li et al. [34] also introduced a MinMin-based online dynamic power management strategy with multiple power-saving states to reduce energy consumption of scheduling algorithms.

Pinel et al. [22] proposed a two-phase heuristic to schedule independent tasks on grids with energy considerations. First, a MinMin approach is applied to optimize the makespan, and then a local search is performed to minimize energy consumption. Through extensive simulation-based comparisons against a parallel asynchronous cellular GA, they showed that their proposed algorithm produces fairly comparable solutions to that of the GA, but within much lesser time.

Lindberg et al. [24] studied the task scheduling problem with the aim of minimizing makespan and energy subject to deadline constraint and tasks' memory requirements; eight heuristics (six greedy algorithms based on list scheduling and two GAs) were introduced to solve this problem using a DVS technique. The two GAs were found to be too slow compared to the heuristics, and they reported worse quality solutions. In a recent work, Iturriaga et al. [25] proposed a parallel multi-objective local search algorithm, based on Pareto dominance, to minimize energy consumption and makespan in the independent tasks scheduling problem. It was shown to outperform a set of fast and accurate two-phases deterministic heuristics based on the traditional MinMin.

In our previous work [26], we introduced an energy consumption model for multicore computing systems. Our approach did not apply DVS nor other specific techniques for power/energy management. Instead, we proposed an energy consumption model based on the energy required to execute tasks at full capacity, the energy when not all the available cores of the machine are used, and the energy that each machine on the system consumes in idle state. We proposed 20 fast list scheduling methods adapted to solve a