Future Generation Computer Systems 36 (2014) 221–236

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Multi-objective energy-efficient workflow scheduling using list-based heuristics

Juan J. Durillo *, Vlad Nae, Radu Prodan

Institute of Computer Science, University of Innsbruck, Austria

HIGHLIGHTS

- Mono-objective greedy schedulers perform poorly in terms of energy efficiency.
- MOHEFT outperforms HEFT and greenHEFT in both makespan and energy consumption.
- MOHEFT increases energy efficiency with small hits on makespan.
- Energy consumption reduced by up to 34.5% with only 2% makespan overhead.

ARTICLE INFO

Article history:
Received 2 November 2012
Received in revised form
16 May 2013
Accepted 17 July 2013
Available online 6 August 2013

Keywords:
Energy-efficient scheduling
Workflow scheduling
Multi-objective optimisation

ABSTRACT

Workflow applications are a popular paradigm used by scientists for modelling applications to be run on heterogeneous high-performance parallel and distributed computing systems. Today, the increase in the number and heterogeneity of multi-core parallel systems facilitates the access to high-performance computing to almost every scientist, yet entailing additional challenges to be addressed. One of the critical problems today is the power required for operating these systems for both environmental and financial reasons. To decrease the energy consumption in heterogeneous systems, different methods such as energy-efficient scheduling are receiving increasing attention. Current schedulers are, however, based on simplistic energy models not matching the reality, use techniques like DVFS not available on all types of systems, or do not approach the problem as a multi-objective optimisation considering both performance and energy as simultaneous objectives. In this paper, we present a new Pareto-based multi-objective workflow scheduling algorithm as an extension to an existing state-of-the-art heuristic capable of computing a set of tradeoff optimal solutions in terms of makespan and energy efficiency. Our approach is based on empirical models which capture the real behaviour of energy consumption in heterogeneous parallel systems. We compare our new approach with a classical mono-objective scheduling heuristic and state-of-the-art multi-objective optimisation algorithm and demonstrate that it computes better or similar results in different scenarios. We analyse the different tradeoff solutions computed by our algorithm under different experimental configurations and we observe that in some cases it finds solutions which reduce the energy consumption by up to 34.5% with a slight increase of 2% in the makespan.

1. Introduction

Precedence-constrained parallel applications, also known as *workflows*, are a popular paradigm used by scientists for modelling large applications. Most of these applications have to deliver results as fast as possible and thus require parallel or distributed computation for decreasing their execution time. A major challenge in this case is to achieve the "best" *schedule* of the workflow tasks onto the available resources that minimises its execution time—a well-known NP-complete problem. Nowadays, high-performance parallel computing is available to almost any scientist or researcher. Along with their many advantages, these facilities introduce new challenges such as the considerable amount of energy required in today's data centres. Besides the green implications of energy savings, Hamilton [1] reported that the financial expenditure for the energy consumption of Amazon's data centres in a period of fifteen years accounts for 19% of its total budget and that 23% of the same budget represents the cost of maintaining the cooling infrastructure. Under these circumstances, energy efficiency is becoming an important objective for computing infrastructure providers due to its environmental and financial implications.

* Corresponding author. Tel.: +43 512 507 53284.
E-mail addresses: juan@dps.uibk.ac.at (J.J. Durillo), vlad@dps.uibk.ac.at (V. Nae), radu@dps.uibk.ac.at (R. Prodan).

0167-739X/$ – see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.future.2013.07.005

Reducing the energy consumption while maintaining the quality of the service in terms of performance has therefore become a major research challenge. The existing works mainly address this issue through the use of Dynamic Voltage and Frequency Scaling (DVFS), which enables online adjustment to voltage and frequency in CMOS circuits [2–4]. By applying this technique to different CPUs in a heterogeneous parallel system, it is possible to have resources working at their highest speed, and implicitly at peak energy consumption, while executing time-critical tasks. Conversely, they will be tuned to work at a lower speed with lower energy consumption while executing non-critical tasks. In such systems, the user is confronted with a set of resources working at different speeds and with different characteristic energy consumptions.

In these circumstances, workflow scheduling must be formulated as a *multi-objective optimisation problem (MOP)* aiming at optimising two possibly conflicting criteria: *makespan* and *energy* consumption for executing the workflow. The main characteristic of MOPs is that no single solution exists that is optimal with respect to all objectives, but a set of tradeoff solutions known as *Pareto set* or *Pareto front*, depending on if we refer to the domain or co-domain of the functions to be optimised. The property of the solutions on the Pareto set/front is that they cannot be simultaneously improved with respect to all objectives. In our particular case, it is likely that no single schedule simultaneously minimises makespan and energy consumption. Instead, there will be a set of solutions with a different tradeoff between makespan and energy consumption.

In this paper, we present a holistic approach for an energy-efficient workflow scheduler for heterogeneous systems, capable of computing a set of tradeoff solutions in a single run. Our method called MOHEFT is an extension of the Heterogeneous Earliest Finish Time (HEFT) algorithm [5], one of the most popular algorithms for workflow scheduling. HEFT is based on the assumption that the execution time of each workflow task is known on each resource, which is not always realistic for highly dynamic distributed systems consisting of a large number of multi-core processors sharing resources (e.g. bus, caches) and affected by the external load caused by other concurrent tasks. To overcome this drawback, MOHEFT relies on empirical models for execution time of workflow tasks and their entailed energy consumption. These models are based on the knowledge extracted from historical executions, reflecting the behaviour of real multi-core CPUs with different levels of energy consumption depending on the number of cores used and their individual level of utilisation, and predicting it for new unseen resources. While we do look into the external load generated by different workflow tasks, we do not consider multi-tenancy in our approach, as scientific applications, including workflows, are typically highly specialised and customised applications owned by individual scientists and tuned for their particular temporary research needs.

The contributions of this paper in the area of energy-efficient scheduling are as follows.

1. *Identification of fine-grained levels of energy consumption in a multi-core CPU*. Thorough extensive experimentation, we measure the energy consumption and performance of different multi-core CPUs with different number of cores used. The experiments show that the performance of individual cores decreases with the concurrent number of cores used, while the energy efficiency increases.
2. *Use of empirical models for energy consumption and performance based on real data*. We build neural network-based empirical models for time and energy consumption of CPU-intensive tasks based on historical executions on heterogeneous sets of machines. Our approach considers external load coming from tasks of the same as the one being modelled; however, we plan in future work to extend it with training information about the types of tasks defining the external load.
3. *A multi-objective energy-efficient scheduling algorithm*. We present an algorithm capable of computing workflow schedules representing tradeoffs between energy consumption and makespan. We prove thorough an extensive experimentation that our algorithm computes solutions with the same or shorter makespan than HEFT and with lower energy consumption than HEFT and greenHEFT, an ad-hoc greedy algorithm for energy optimisation. MOHEFT also outperforms the most popular multi-objective optimisation algorithm, NSGA-II [6], computing solutions with shorter makespan and lower energy consumption.
4. *Analysis of the impact of different workflow characteristics and different resources on the tradeoff solutions*. We carry out a thorough evaluation of our proposal under different circumstances: number of tasks, homogeneous resources in terms of speed and energy consumption, or heterogeneous resources. Our aim is to analyse the performance of MOHEFT versus other algorithms and the impact of different scenarios on the computed tradeoff solutions.

The rest of this paper is structured as follows. The next section formally describes our problem and introduces some background on multi-objective optimisation. Related works on energy modelling, multi-objective workflow scheduling and energy-efficient scheduling are analysed in Section 4. After that, we describe the modelling approach followed in this paper. Section 6 introduces MOHEFT. The evaluation of our algorithm and the analysis of the obtained results is included in Section 7. Finally, we present the main conclusions and future work in Section 8.

2. Formalism

Our problem consists in scheduling a workflow application's tasks on a set of available heterogeneous resources in such a way that the makespan and the energy consumption of its execution are minimised. We introduce in the remainder of this section a simple but realistic formalism that defines the workflow, resource environment, and the metrics targeted.

2.1. Workflow application

We model a *workflow application* as a directed acyclic graph (DAG), $W = (A, D)$ consisting of n tasks or activities: $A = \bigcup_{i=1}^{n} \{A_i\}$, interconnected through control flow and data flow dependences, D, defined as

$$D = \left\{ \left(A_i, A_j, Data_{ij}\right) \mid \left(A_i, A_j\right) \in A \times A \right\},$$

where $Data_{ij}$ represents the size of the data needed to be transferred from activity A_i to activity A_j. In the remainder of this paper, we use the terms activity and task interchangeably. We use $pred\,(A_i) = \{A_k | \forall\,(A_k, A_i, Data_{ki}) \in D\}$ to denote the set of *predecessors* of activity A_i (i.e. A_i's parents to be completed before starting it). Every activity $A_i \in A$ is characterised by its length (or workload) measured for example in the total number of instructions, which affects its execution time and the energy consumption.

2.2. Resource environment

We assume a hardware platform consisting of m heterogeneous resources $R = \cup_{j=1}^{m} R_j$. Each resource $R_j \in R$ is described by a set of nine different characteristics which influence the number of machine instructions per second it is able to process, and the energy it consumes during this. A brief description of these nine characteristics is summarised in Table 1. We use $sched(A_i)$ to denote the resource on which activity A_i is scheduled to be executed.