

A hybrid genetic algorithm for optimization of scheduling workflow applications in heterogeneous computing systems



Saima Gulzar Ahmad^a, Chee Sun Liew^a, Ehsan Ullah Munir^b, Ang Tan Fong^a, Samee U. Khan^{c,*}

^a Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia

^b Department of Computer Science, COMSATS Institute of Information Technology, Quaid Avenue, Wah Cantt, Pakistan

^c Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND 58108-6050, USA

HIGHLIGHTS

- Proposed HGA (hybrid GA) to schedule workflow in heterogeneous environment.
- Simulation results are presented with synthesized and real-world workflows.
- HGA identifies lesser length of the workflows.
- HGA also improves load balancing.
- Significant improvement in schedule lengths as compared to existing work.

ARTICLE INFO

Article history:

Received 15 August 2014

Received in revised form

7 August 2015

Accepted 9 October 2015

Available online 19 October 2015

Keywords:

Workflow

Genetic algorithm

Heuristic

Directed Acyclic Graphs

ABSTRACT

Workflow scheduling is a key component behind the process for an optimal workflow enactment. It is a well-known NP-hard problem and is more challenging in the heterogeneous computing environment. The increasing complexity of the workflow applications is forcing researchers to explore hybrid approaches to solve the workflow scheduling problem. The performance of genetic algorithms can be enhanced by the modification in genetic operators and involving an efficient heuristic. These features are incorporated in the proposed Hybrid Genetic Algorithm (HGA). A solution obtained from a heuristic is seeded in the initial population that provides a direction to reach an optimal (makespan) solution. The modified two fold genetic operators search rigorously and converge the algorithm at the best solution in less amount of time. This is proved to be the strength of the HGA in the optimization of fundamental objective (makespan) of scheduling. The proposed algorithm also optimizes the load balancing during the execution side to utilize resources at maximum. The performance of the proposed algorithm is analyzed by using synthesized datasets, and real-world application workflows. The HGA is evaluated by comparing the results with renowned and state of the art algorithms. The experimental results validate that the HGA outperforms these approaches and provides quality schedules with less makespans.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Scientific communities are dealing with the experimentation and simulations that involve huge amount of data. Many scientific fields, such as Astronomy, Bio-informatics, Meteorology, Environmental Science, and Geological Sciences deal with large-scale data

[3]. Handling such an amount of data has become a great challenge. The increased complexity, the heterogeneity of the scientific applications and the execution platforms cause additional challenges in the big data management [4]. Scientists use workflow management systems (WMS) to handle such large-scale experiments. The WMS organize and manage large-scale distributed data. Moreover, the WMS simplify the complexity to run data-intensive applications [26].

Many WMS have been developed over the last decade and are widely used by the scientific world, such as Pegasus [11], Swift [23], Kepler [15], Taverna [13], Trident [5], and Konstanz Information Miner (KNIME) [6]. The workflow optimization is a

* Corresponding author.

E-mail addresses: saimagulzarahmad@siswa.um.edu.my (S.G. Ahmad), csliw@um.edu.my (C.S. Liew), ehsanmunir@comsats.edu.pk (E.U. Munir), angft@um.edu.my (T.F. Ang), samee.khan@ndsu.edu (S.U. Khan).

<http://dx.doi.org/10.1016/j.jpdc.2015.10.001>

0743-7315/© 2015 Elsevier Inc. All rights reserved.

vital part of all WMS that takes place during resource mapping, and scheduling in workflow life-cycle [10]. An efficient resource allocation strategy directly effects the WMS overall performance. Therefore, scientific community has been putting their efforts to propose novel techniques for the workflow optimization and overcome the challenges that emerge with the development in scientific applications.

The major contribution of this research is a new workflow scheduling algorithm, namely the Hybrid Genetic Algorithm (HGA). The HGA is a genetic algorithm (GA) guided by a heuristic to pursue an optimal (makespan) schedule in an efficient manner. Genetic algorithms (GAs) have the capability to search large problem spaces. The heuristic helps GA to enhance the performance. The genetic operators (crossover and mutation) are modified that help the HGA to explore most of the problem space in less amount of time and ultimately the HGA converges. The workflows are usually modeled as Directed Acyclic Graphs (DAGs) [24], in which the nodes or vertices represent tasks, and the edges represent the precedence or data dependencies among the tasks. The same workflow model has been adopted in our work. The behavior of the proposed algorithm is analyzed with the diverse datasets, including synthesized, and real-world application workflows e.g. Montage, and CyberShake.¹ Montage and CyberShake workflows are common choice of the researchers as benchmarks to validate the performance of workflow scheduling algorithms. These workflows have different characteristics and include most of the workflow patterns that are required to be analyzed during the performance evaluation of any novel scheduling algorithm. Random workflows provide grounds to check the behavior of proposed algorithm for different sizes and shapes of workflows having different values of out-degree and communication to computation ratio.

The rest of the paper is organized as follows. Section 2 provides the background. The related work is summarized in Section 3. Section 4 describes the proposed algorithm. The performance analysis and discussion are presented in Section 5, and Section 6 concludes the paper.

2. Background

Workflows have simplified the management of data-intensive applications by decomposing the applications into smaller tasks that need to be completed in a sequence to reach the results. The concept of workflows has reduced the complexity of large-scale experiments [10]. The WMS have been designed to organize and manage large data using a step-wise approach [8]. The major stages of a WMS are shown in Fig. 1. Scientific experiments and simulations generate huge amounts of data. Different WMS mandate inputs in specific formats. The workflow composition stage constructs a high level workflow (abstract workflow), that identifies the software component and data required for the particular execution without any details of physical execution resources. In the next stage, the abstract workflow is mapped to the physical resources and provides an executable plan in the form of a concrete workflow. Resource allocation is optimized in the mapping stage. The mapped workflow is then scheduled on the physical resources and executed. The performance of the workflow execution is monitored and results are gathered. The resource allocation is optimized in the mapping stage and scheduling during the execution [4]. Efficient scheduling can enhance the system performance [16]. We focus on the latter stage of the optimization in this paper. The HGA is a scheduling algorithm that optimizes the makespan (schedule length) of the workflow and balances the load across all of the available resources.

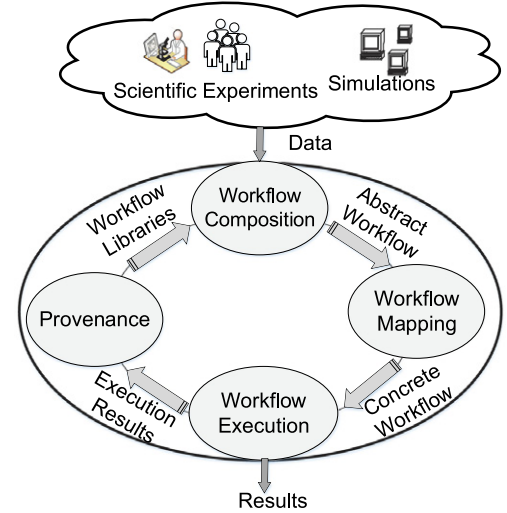


Fig. 1. Major stages of a generic workflow management system.

The paper is based on our previously proposed algorithm Performance Effective Genetic Algorithm (PEGA) [1]. In this paper we present a hybrid approach by using a schedule of well-known heuristic Heterogeneous Earliest Finish Time (HEFT) in the initial population. In-addition load balancing procedure enhances the equal load distribution among resources. These features provide strength to the proposed algorithm to provide better schedules with lower makespan. HEFT is a well-known list-based scheduling heuristic that outperforms a number of algorithms [21]. HEFT schedule is used as a seed in the initial population of the proposed hybrid approach. List-based scheduling is a two-phase process. In the first phase, the task-priority list is generated, and in the second phase, the processors are allocated. In the HEFT, a priority list is generated in the descending order of up-ward rank or the b -level. The processors allocation for tasks is based on the earliest finish time. Following are the attributes used in HEFT that are b -level, earliest start time, and earliest finish time. These attributes are defined and briefly discussed below.

- Upward rank (b -level):

$$rank_b(v_n) = \overline{w}_n + \max_{v_m \in succ(v_n)} \{\overline{d}_{nm} + rank_b(v_m)\}, \quad (1)$$

where \overline{w}_n is the average execution cost (it is the amount of time required to execute a task on execution node), \overline{d}_{nm} is the average communication cost (it is the cost of data transfer from a parent to child node in a workflow) between nodes n and m , and v_m is the successor of v_n . The up-ward rank of each of the task is calculated by the above recursive function and a task priority list is generated by a descending order of the corresponding upward rank values.

- Earliest start time (EST):

The earliest time of a processor p_k for a task v_n is defined as:

$$EST(v_n, p_k) = \max\{avail[k], \max_{v_m \in pred(v_n)} (AFT(v_m) + d_{nm})\}, \quad (2)$$

where $avail[k]$ is the time when processor k is ready to execute a new task, $AFT(v_m)$ is the actual finish time of task v_m on processor p_k , d_{nm} is the communication cost of tasks v_n and v_m , and the task v_m is predecessor of v_n . Task v_n can have more than one predecessors. Therefore, the maximum sum of the AFT and d_{nm} among the predecessors will be considered in Eq. (2), which will cause the maximum delay in the execution of the task v_n .

¹ www.confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator.

Download English Version:

<https://daneshyari.com/en/article/433001>

Download Persian Version:

<https://daneshyari.com/article/433001>

[Daneshyari.com](https://daneshyari.com)