Information Sciences 192 (2012) 228–243

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Swarm scheduling approaches for work-flow applications with security constraints in distributed data-intensive computing environments

Hongbo Liu [a,b,c,*], Ajith Abraham [c,d], Václav Snášel [c,d], Seán McLoone [e]

[a] School of Information, Dalian Maritime University, 116026 Dalian, China
[b] School of Computer, Dalian University of Technology, 116023 Dalian, China
[c] Machine Intelligence Research Labs, Auburn, WA 98071, USA
[d] Department of Computer Science, VŠB-Technical University of Ostrava,708 33 Ostrava-Poruba, Czech Republic
[e] Department of Electronic Engineering, National University of Ireland Maynooth, Maynooth, Co. Kildare, Ireland

ARTICLE INFO

Article history:
Available online 9 January 2012

Keywords:
Swarm intelligence
Particle swarm
Scheduling problem
Work-flow
Security constraints
Distributed data-intensive computing environments

ABSTRACT

The scheduling problem in distributed data-intensive computing environments has become an active research topic due to the tremendous growth in grid and cloud computing environments. As an innovative distributed intelligent paradigm, swarm intelligence provides a novel approach to solving these potentially intractable problems. In this paper, we formulate the scheduling problem for work-flow applications with security constraints in distributed data-intensive computing environments and present a novel security constraint model. Several meta-heuristic adaptations to the particle swarm optimization algorithm are introduced to deal with the formulation of efficient schedules. A variable neighborhood particle swarm optimization algorithm is compared with a multi-start particle swarm optimization and multi-start genetic algorithm. Experimental results illustrate that population based meta-heuristics approaches usually provide a good balance between global exploration and local exploitation and their feasibility and effectiveness for scheduling work-flow applications.

1. Introduction

Advances in high performance computing, computational grid and cloud computing platforms are enabling researchers to explore increasingly computationally complex problems in domains such as chemistry, meteorology, high-energy physics, astronomy, biology, human brain planning and sensor networks [31,24,62,5,23]. A key consideration when developing applications for these platforms is how to schedule application tasks in order to optimize performance with respect to the available resources. However, scheduling in distributed data-intensive computing environments differs substantially from conventional scheduling. Jobs and resources in data-intensive applications have to meet specific requirements including process flow, data access/transfer, security constraints, completion cost, flexibility and availability, adding greatly to the complexity of the scheduling problem. In addition, all the components in an application can interact with each other directly or indirectly. Scheduling algorithms in traditional computing paradigms rarely consider the data transfer problem when mapping computational tasks, but such an omission can be very costly in the case of distributed data-intensive applications [23,14].

The particle swarm paradigm [27,11,30] is inspired by the social behavior patterns of organisms that live and interact within large groups. In particular, it incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms

* Corresponding author at: School of Information, Dalian Maritime University, 116026 Dalian, China.
E-mail addresses: lhb@dlut.edu.cn (H. Liu), ajith.abraham@ieee.org (A. Abraham), vaclav.snasel@vsb.cz (V. Snášel), sean.mcloone@eeng.nuim.ie (S. McLoone).

0020-0255/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.12.032

of bees, and even human social behavior. It can be easily implemented and applied to solve various function optimization problems and, by extension, to problems that can be transformed into function optimization problems. As an algorithm its main strength is its fast convergence, which compares favorably with many other global optimization algorithms [15,6,1,13,34,2,3,46].

In this paper, several meta-heuristics adaptations to the particle swarm optimization algorithm are investigated for scheduling work-flow on distributed data-intensive computing environments. In particular, a novel variable neighborhood search strategy is introduced that helps prevent particle swarms getting trapped in local minima.

The remainder of the paper is organized as follows. Related research on scheduling algorithms in distributed data-intensive computing environments is presented in Section 2. The scheduling problem and security constraint model are formulated in Section 3. Section 4 then introduces the particle swarm based schedule optimization algorithm and a number of extensions targeted at enhancing performance. Experimental results and discussions are provided in Section 5 and finally in Section 6 conclusions are presented.

2. Related work

The job scheduling problem in distributed computing systems has been drawing researchers' attention worldwide, not only because of its practical and theoretical importance, but also because of its complexity. It is an NP-hard optimization problem [7,38], which means that the amount of computation required to find optimum solutions increases exponentially with problem size. Different approaches have been proposed to solve this problem. Ranganathan and Foster [40] defined a general and extensible scheduling framework within which a wide variety of scheduling algorithms are instantiated. Khoo et al. [29] proposed a distributed resource-scheduling algorithm capable of handling multiple resource requirements for jobs that arrive in a grid computing environment. Yu and Marinescu [58] proposed a set of divisible load scheduling algorithms for different data staging strategies. Their algorithm generates the optimal mapping from process groups to parallel systems, computes the data allocated to each system, and guarantees the shortest makespan. Venugopal and Buyya [52] considered the problem of scheduling an application composed of a set of independent tasks, each of which requires multiple data sets that are each replicated on multiple resources. They break this problem into two parts: (1) to match each task (or job) to one resource for executing the job and one storage resource for accessing each data set required by the job; and (2) to assign the set of tasks to the selected resources. They extend the MinMin and Sufferage algorithms to schedule the set of distributed data-intensive tasks. Roshanaei et al. [41] proposed a variable neighborhood search algorithm for the problem which has also proven to be effective.

Zhang and Wu [60] proposed a hybrid simulated annealing algorithm based on the immune mechanism for the job shop scheduling problem. The hybrid optimization algorithm is subsequently tested on a number of job shop instances. Computational results for different-sized instances show that the proposed hybrid algorithm performs effectively and converges quickly to satisfactory solutions. Chung et al. [10] proposed a modified genetic algorithm approach to deal with distributed scheduling models with maintenance consideration, aiming to minimize the makespan of the jobs. They also tested the influence of the relationship between maintenance repair time and machine age on the performance of scheduling of maintenance during distributed scheduling in the studied models. Wen et al. [53] investigate a heuristic-based hybrid genetic-variable neighborhood search algorithm for the minimization of makespan in the heterogeneous multiprocessor scheduling problem. The empirical results on benchmark task graphs of several well-known parallel applications show that the hybrid algorithm significantly outperforms several related algorithms in terms of the schedule quality.

Work-flows play an important role in distributed computing environments. For the job scheduling problem, the work-flow constraints need to be considered. Tao et al. [47] presented a grid work-flow scheduling based on reliability cost. The performance evaluation results demonstrate that the approach improves the dependability of work-flow execution and success ratio of tasks with low reliability cost.

With the development of large scale distributed data-intensive computing systems, issues of reliability and security have increasingly become the focus of attention. Song et al. [45] proposed a space–time genetic algorithm for trusted job scheduling. According to their security model, a job can possibly fail if the site security level is lower than the job security demand. They further proposed six risk-resilient scheduling algorithms to assure secure grid job execution under different risky conditions. These risk-resilient job scheduling schemes can upgrade grid performance significantly at only a moderate increase in extra resources or scheduling delays in a risky grid computing environment [43]. Xie and Qin [55] proposed a security-aware real-time heuristic strategy for clusters, which integrates security requirements into the scheduling for real-time applications on clusters. Experimental results show that their approach significantly improves security. They also introduced the concept of security heterogeneity for their scheduling model in the context of distributed systems. Based on the concept, they proposed a heuristic scheduling algorithm, which strives to maximize the probability that all tasks are executed without any risk of being attacked [56]. Wu and Sun [54] proposed a genetic algorithm for job scheduling to address the problem of heterogeneity of fault-tolerance mechanisms in a computational grid. They assume that the system supports four different fault-tolerance mechanisms, including job retry, job migration without checkpointing, job migration with checkpointing and job replication mechanisms. The risky nature of the grid environment is taken into account in the algorithm.

Recently, swarm intelligence and multi-agent techniques have attracted the attention of parallel computing researchers. Liu et al. [36] presented a complete multi-agent framework for dynamic job shop scheduling considering robustness and