Future Generation Computer Systems 45 (2015) 114–122

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Anticipating resource saturation in Federated Grids☆

Katia Leal

Universidad Rey Juan Carlos, Departamento de Sistemas Telemáticos y Computación, Escuela Superior de Ingeniería de Telecomunicación, Camino del Molino SN, 28943 Fuenlabrada, Madrid, Spain

HIGHLIGHTS

- Self-adjusting resource-sharing policies save time and communication bandwidth.
- Our policies map jobs proportional to the performance of the resources.
- Our algorithms perform a mapping of jobs adjusted to resources real saturation.
- We do not need to negotiate with the rest of participating infrastructures.
- Our strategies can reach the best makespan within different saturation level scenarios.

ARTICLE INFO

Article history:
Received 15 April 2014
Received in revised form
14 October 2014
Accepted 25 October 2014
Available online 6 November 2014

Keywords:
Resource sharing
Performance model
Decentralized
Non-coordinated scheduling
Federated Grids

ABSTRACT

In a dynamic and geographically distributed Federated Grid where resources are shared between system participants, there is a lack of mechanisms capable of reallocating already scheduled tasks based on grid infrastructure owners' current internal needs. In this paper we propose a set of policies for both, users and owners, that aid owners to satisfy internal peak demands and users to achieve the best makespan despite the circumstances. As in our previous work, the main purpose is to do so in the least intrusive way possible to maintain software stack independence of all participants, and to save time and communication bandwidth by anticipating grid resources saturation. These strategies suppose a novel approach for decentralized and non-cooperative workflow scheduling in a federation of heterogeneous grid infrastructures. We evaluate and prove the feasibility of our policies through a set of simulations that reflect the worst case where all resources are saturated. The results show that, in the worst scenario, our scheduling mechanism is beneficial to big infrastructure owners since they can achieve their own internal objectives, as well as to small users since they can reach the best possible completion time.

1. Introduction

Resource management and application scheduling over a Federated Grid is quite complicated due to large system size, infrastructures' heterogeneity, domain-specific policies and dynamic environment. All in all, in the last decade there have been a set of efforts focused on scheduling as well as on decentralized cooperative or non-cooperative mechanisms. However, there are no examples in the literature on how to anticipate grid resources saturation in order to lighten the load of grid infrastructures so owners can better respond to internal needs, and to reduce the utilization of other resources including computing cycles and network bandwidth, thus enabling users to achieve the best makespan possible. Our previous investigation [1] demonstrated that applying self-adjusting resource-sharing policies to a performance-based scheduling strategy can contribute to reduce the number of jobs migrations, and thus saving time and improving application makespan while keeping Federated Grid participating infrastructures' owners autonomy. Now, this work is extended to provide a full set of actions to make a decision when a resource is near to saturation or owners have internal jobs peak demand. Specifically, in this paper we propose two new owner's actions and one new user's action.

Section 2 compares the novelty of the proposed resource-sharing strategies with respect to the existing systems. Majority of brokering approaches solve Federated Grid scheduling problem based on a decentralized model. As mentioned in [2], these systems can be further classified into two main areas: workload scheduling and coordinated mechanisms.

In Section 3 we briefly review our decentralized [3], performance-based algorithm [4] for scheduling independent tasks in

☆ Thanks to the GSyC/Libresoft Research on Free Software Engineering group at Rey Juan Carlos University for the hardware support to run the simulations.
E-mail address: katia.leal@gmail.com.

http://dx.doi.org/10.1016/j.future.2014.10.025

Federated Grids. Also, this section describes the full set of self-adjusting resource-sharing policies proposed. In fact, with this work we want to demonstrate that our sharing policies can contribute to reduce the number of jobs migrations, and thus saving time and improving application makespan while keeping Federated Grid participating infrastructures' owners autonomy.

Simulations and experiments setup are described in Section 4. Five different experiments are performed to capture the behavior of the proposed resource-sharing strategies. In all cases, the environment setup is exactly the same; the experiments only differ in the resource-sharing policies implemented at the meta-scheduler level. In addition, applications are submitted in specific simulation time instants that correspond with different infrastructures' saturation levels, or what is the same, with a different number of free computing elements. Thus, the five experiments coexist in a *low saturation*, *medium saturation* and *high saturation* level scenario. This *high saturation* level scenario reflects the worst case where all resources are saturated.

Section 5 provides simulation results of the five experiments. We analyze the makespan achieved (completion time), the objective function of each algorithm (number of jobs submitted to each participating infrastructure), number of jobs migrated, rejected, suspended, canceled and anticipated, and the behavior of the different algorithms through graphics.

Finally, Section 6 summarizes the benefits of applying self-adjusting resource-sharing policies to performance based scheduling strategies in Federated Grids. Guidelines for our current research activities are also provided.

2. Related work

Non-coordinated decentralized meta-schedulers perform scheduling strategies without taking into account resources current load or specific resources owners internal demands [5,6], leading to suboptimal schedules that can cause coexistence problems in the Federated Grid. This lack of cooperation exacerbates the utilization of different types of resources, including computing cycles and network bandwidth. Moreover, current brokering systems have evolved around centralized or hierarchical models. In fact, key functionalities such as resource discovery are delegated to centralized server machines. This is important since, as pointed out by Ranjan et al. [7], centralized and hierarchical information services (such as R-GMA [8] and MDS-2,3,4 [9]) have several design limitations including: single point of failure, lack of scalability, high network communication cost, and computational power to serve queries. In addition, the studies conducted by Zhang et al. [10] verified that existing systems including R-GMA, MDS, and Hawkeye fail to scale beyond 300 concurrent users. Regarding to the possibility of anticipating resource saturation, the GridWay [11,12] workflow system includes a ''Performance Slowdown Detection'' mechanism. Thus, the scheduler requests a migration when an intolerable performance loss for a job is detected. The scheduler acts on behalf of their users, but the owners of the resources do not have a way to express their needs. Also, UNICORE [13] has a ''Service Orchestrator'' layer which is responsible to executing the individual tasks in a workflow, handling job execution and monitoring on the Grid. However, this service does not include resource-sharing strategies.

On the other hand, decentralized coordinated scheduling schemes negotiate resource conditions with the other application level schedulers in the system. In this case, the negotiation may lead to a large number of messages generated per job before being scheduled. Negotiation among all the participants can be based on the well-known coordination protocol contract net [14]. This protocol have been widely used for Grid scheduling, as shown by several works [15–17]. However, negative effects of contract net communication overhead on job execution time in a multi-agent Grid computing system have also been studied [18]. In their work, authors claim that the main overhead in the system is due to the execution of the Contract Net protocol. More specifically in the amount of messages exchanged required for each request. In [19] authors propose a different approach for decentralized and cooperative workflow scheduling. In this case, the participants in the system work together to enable a single cooperative resource-sharing environment. This approach derives from a Distributed Hash Table (DHT) based d-dimensional logical index space that serves as a blackboard system where distributed participants can post and search complex coordination objects that regulate system-wide scheduling decision making. Nevertheless, this system does not provide users neither owners with specific mechanisms to anticipate and to react a resource saturation.

Although our solution can also be classified as non-coordinated decentralized, the main axis of our proposed algorithms is that they consider the performance of the infrastructures forming the Federated Grid, not only their state. So, any changes in infrastructures' behavior will be reflected in their performance and will be considered in order to determine the number of jobs to submit to them. At the same time, our solution does not strongly depend on resource discovery services. In addition, since our sharing strategies are self-adjusting, they do not need to negotiate with the rest of participating infrastructures, and thus, they do not flood the network with negotiation messages. Therefore, our policies save time and communication bandwidth. Also, we do not need to deploy agents or specialized software across the different infrastructures what allows us to maintain software stack independence of all participants.

3. Decentralized non-coordinated self-adjusting resource-sharing policies

In our previous work [3] we proposed the decentralized model depicted in Fig. 1 as an alternative to centralized, application-centric or ad-hoc solutions to the scheduling problem in Federated Grids. The model puts a meta-scheduler at the top level of each grid infrastructure, over the workload managers. This new layer presents a queue on which jobs have to wait to be scheduled. However, the experimental results demonstrated that, even with the delays introduced by the different queues jobs have to pass through, this model is still very efficient. Instead of having a unique centralized global scheduler to map the jobs of the distinct Grids, each one has its own meta-scheduler. The aim of the mapping strategy implemented on the meta-scheduler at each Grid infrastructure of the Federated Grid was to reduce the makespan of its applications and to increase the performance of its own Grid infrastructure.

Subsequent investigations [4] considered three algorithms for this model: ARAE (All Resources Are Equal), PT-AR (Per Type-All Resources), and PT-RR (Per Type-Resources with Results). These algorithms are based on a performance model [20] that allows us to parameterize and compare different grids.

The performance model is based on the parameters proposed by Hockney and Jesshope [21]: r_∞ (asymptotic performance), which is the maximum rate of performance in tasks executed per second, and $n_{1/2}$ (half-performance length), which is the number of tasks required to obtain the half of the asymptotic performance. These parameters, which characterize the average behavior of the system, are obtained by means of a first-order adjustment to the function representing the number of tasks completed as a function of time.

With every completed job, the corresponding algorithm recalculates the performance of the individual Grid infrastructure where it was executed, by updating its r_∞ and $n_{1/2}$ values [20]. Finally, to characterize the performance of the whole Federated