J. Parallel Distrib. Comput. 74 (2014) 1914–1929

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

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

Fair scheduling of bag-of-tasks applications using distributed Lagrangian optimization

Rémi Bertin [a,c], Sascha Hunold [b,c], Arnaud Legrand [b,c,*], Corinne Touati [a,c]

[a] INRIA, France
[b] CNRS, France
[c] University of Grenoble. LIG laboratory (MESCAL project), Montbonnot, France

HIGHLIGHTS

- A robust, fair, and optimal distributed scheduling algorithm for concurrent BoT applications with arbitrary CCR is proposed.
- Despite similarity, this context is more complicated than multi-path flow control.
- Non-trivial adaptations of Distributed Lagrangian Optimization are required.
- Experimental proof of convergence is given for large platforms.

ARTICLE INFO

Article history:
Received 20 September 2012
Received in revised form
11 June 2013
Accepted 15 August 2013
Available online 23 August 2013

Keywords:
Lagrangian optimization
Steady-state scheduling
Distributed scheduling
Grid computing

ABSTRACT

Large scale distributed systems typically comprise hundreds to millions of entities (applications, users, companies, universities) that have only a partial view of resources (computers, communication links). How to fairly and efficiently share such resources between entities in a distributed way has thus become a critical question.

Although not all applications are suitable for execution on large scale distributed computing platform, ideal are the Bag-of-Tasks (BoT) applications. Hence a large fraction of jobs in workloads imposed on Grids is made of sequential applications submitted in the form of BoTs. Up until now, mainly simple mechanisms have been used to ensure a fair sharing of resources among these applications. Although these mechanisms are proved to be efficient for CPU-bound applications, they are known to be ineffective in the presence of network-bound applications.

A possible answer resorts to Lagrangian optimization and distributed gradient descent. Under certain conditions, the resource sharing problem can be formulated as a global optimization problem, which can be solved by a distributed self-stabilizing supply and demand algorithm. In the last decade, this technique has been applied to design various network protocols (variants of TCP, multi-path network protocols, wireless network protocols) and even distributed algorithms for smart grids.

In this article, we explain how to use this technique for fairly scheduling concurrent BoT applications with arbitrary communication-to-computation ratio on a Grid. Yet, application heterogeneity raises severe convergence and stability issues that did not appear in the previous contexts and need to be addressed by non-trivial modifications. The effectiveness of our proposal is assessed through an extensive set of complex and realistic simulations.

1. Introduction

Large scale distributed computing infrastructures are now a reality. Production Grids like EGEE comprise hundreds of sites and several dozens of thousands of processing units. Volunteer computing infrastructures such as BOINC comprise over 580,000 hosts that deliver over 2300 TeraFLOP per day. In such systems, entities have only a partial view of the system, which makes fair and efficient sharing of resources (CPU, network links, storage, ...) among entities (network flows, users, applications, ...) particularly challenging since centralized algorithms do not scale well and distributed algorithms can only rely on local information. A possible approach resorts to the combined use of Lagrangian optimization and distributed gradient descent, which leads to distributed self-stabilizing supply and demand algorithms. In the last decade, this technique, which we call DLO (Distributed Lagrangian

* Corresponding author at: University of Grenoble. LIG laboratory (MESCAL project), Montbonnot, France.
E-mail addresses: sascha.hunold@imag.fr (S. Hunold), arnaud.legrand@imag.fr (A. Legrand), corinne.touati@imag.fr (C. Touati).

0743-7315/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jpdc.2013.08.011

Optimization) for short in the sequel, has been applied to design network protocols (variants of TCP [43], multi-path network protocols [51], wireless network protocols [28]) and even distributed algorithms for smart grids [18]. DLO is very appealing as it allows the choice of a wide variety of fairness criteria and achieves both optimal path selection and flow control.

A very large number of applications that are currently deployed on large scale distributed systems such as grids or volunteer computing systems are Bag-of-Tasks (BoT) applications. Up until now, mainly simple mechanisms have been used to ensure a fair sharing of resources among these applications. Although these mechanisms are proved to be efficient for CPU-bound applications, they are known to be ineffective in the presence of network-bound applications. The similitude between the grid context and the multi-path network context indicates Lagrangian-based algorithms as natural candidates for fairly and efficiently scheduling BoT applications.

We first review in Section 3 the context of BoT scheduling and the limitations of the existing approaches. Then, we introduce in Section 3 network protocol engineering techniques based on DLO. DLO has been widely used in the networking community and in particular to propose flow control mechanisms in multi-path networks, although to the best of our knowledge their efficiency has been evaluated only in very limited settings.

We explain in Section 4 the similarities between the flow control problem and the problem of fairly sharing communication and computation resources between multiple BoT applications in a grid environment. We show in Section 5 how DLO can be used to design a hierarchical and distributed algorithm. This algorithm only requires local information at each worker process and at each buffer of the network links.

We demonstrate in Section 6 through a carefully designed set of simulations that applying simply DLO to the grid context is effective if and only if all applications are identical: application heterogeneity raises very complex practical convergence issues. Surprisingly, this issue had been completely overlooked in the previous works that rather focused on the lack of strict convexity of the global objective function (, which complicates the proof of the convergence but never appeared as a practical issue in our experiments).

To address application heterogeneity, we detail in Section 7 a set of non-trivial adaptations that are required to ensure convergence. In Section 8, we prove their effectiveness in a fully heterogeneous setting through an extensive set of simulations. We briefly illustrate in Section 9 the ability of the algorithm to adapt to the departure of critical nodes. We believe that our thorough analysis enables the reader to deeply understand the potential benefits as well as the limitations of DLO in the context of grid computing.

The contributions of this article can be summarized as follows:

- a *robust*, *fair* and *optimal distributed scheduling algorithm for concurrent BoT applications with arbitrary communication-to-computation ratio* on Grids. Popular existing infrastructures do not offer support for applications with such characteristics. The effectiveness of this algorithm is assessed in a wide variety of complex scenarios.
- our algorithm is based on DLO but requires a set of non-trivial adaptations compared to more classical approaches that can be found in the network protocol engineering literature. We provide an experimental proof that although a naive adaptation of DLO is effective when all applications are identical, it is bound to fail when applications have different characteristics. Hence, heterogeneity makes BoT scheduling on grid computing platforms significantly more complex than the flow control in multi-path networks.
- we provide an in-depth understanding of our algorithm and of the convergence issues raised in our context by presenting a general introduction to DLO and a comprehensive survey on how it has been used to design network protocols.

2. Scheduling bag-of-tasks on grid platforms

Not all applications are suitable for execution on large scale distributed computing platforms but ideal are the Bag-of-Tasks (BoT) applications. Hence a large fraction of jobs in workloads imposed on Grids is made of sequential applications submitted in the form of BoT. Despite their suitability for such platforms, scheduling such applications is complexified by several theoretical and practical aspects among which are platform heterogeneity, management of both data and computations, the presence of several users (and even sometimes virtual organizations), fault tolerance, the difficulty to predict workload characteristics as well as their evolution over time and across users. This has led to a significant amount of efforts on workload characterization, on the design of BoT management infrastructures and on scheduling theory.

In this article, we seek to design a *fair* and *optimal hierarchical scheduling algorithm* for applications with *arbitrary communication-to-computation ratio*. The key characteristic of our work is the consideration of mixtures of CPU-bound and network-bound applications in a multi-user context where a fair sharing of resources needs to be ensured. The Large Hadron Collider Computing Grid (LCG) [33] is a system with such needs. The Large Hadron Collider (LHC) produces roughly 15 Petabytes of data annually that are accessed and analyzed by thousands of scientists around the world. The resulting computation tasks have a much larger communication-to-computation ratio than typical distributed computing applications and their efficient management is still an open problem.

2.1. BoT scheduling infrastructures

The most well-known systems specifically tailored for BoT in the Grid context are APST [13], Nimrod/G [1], Condor [38], MyGrid [16], Cigri [21] and Glite Workload Management System [40]. All these infrastructures work on a best-effort basis, with no performance guarantees. At a different scale, BOINC [2] is a centralized scheduler that distributes tasks for participating applications, such as SETI@home, ClimatePrediction.NET, World Community Grid or Einstein@Home. Most of the existing systems are client–server oriented and are proved to be efficient for applications with a very small communication-to-computation ratio (CCR). This is a key simplifying hypothesis as it enables to serve clients regardless of their connectivity and avoids server overload. It also enables to rely on very simple sharing mechanisms. For example the BOINC sharing policy fairly shares on each client the CPU resource among projects to which the volunteer subscribed [2]. Yet, it has been proved [36] that such a simplistic and local approach leads to resource waste whenever communication links become critical resources.

Most existing workload studies [41,25,26] ignore communications as such information can generally not be traced at the batch scheduler level. In most currently deployed infrastructures, the file manager and the batch scheduler work in a best effort mode, trying to prefetch the data or to schedule computations near data whenever possible. The interaction between efficient data management and scheduling is still not well understood, especially at large scale and under fairness constraints. Hence, simple and pragmatic strategies are used in practice although this lack of understanding motivates a lot of research work in scheduling theory.

2.2. Scheduling theory

The classical approach for BoT scheduling is to minimize the completion time of a single batch while the main issue is to select resources and manage the data. Such problems are generally solved with list scheduling heuristics like min–min, sufferage, or similar