ARTICLE IN PRESS

Simulation Modelling Practice and Theory xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Simulation Modelling Practice and Theory

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

Using a novel message-exchanging optimization (MEO) model to reduce energy consumption in distributed systems

Nik Bessis [a], Stelios Sotiriadis [a,*], Florin Pop [b], Valentin Cristea [b]

[a] School of Computing & Maths, University of Derby, Derby, United Kingdom
[b] University "Politehnica" of Bucharest, Bucharest, Romania

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Distributed systems
Message exchange optimization
Energy consumption in distributed systems
Inter-cloud

ABSTRACT

The concept of optimizing energy efficiency in distributed systems has gained particular interest. Most of these efforts are focused on the core management concepts like resource discovery, scheduling and allocation without focusing on the actual communication method among system entities. Specifically, these do not consider the number of exchanged messages and the energy that they consume. In this work, we propose a model to optimize the energy efficiency of message-exchanging in distributed systems by minimizing the total number of messages when entities communicate. So we propose an efficient messaging-exchanging optimization (MEO) model that aims to minimize the sum of requests and responses as a whole rather than only the number of requests. The view is to optimize firstly the energy for communication (e.g. latency times) and secondly the overall system performance (e.g. makespan). To demonstrate the effectiveness of MEO model, the experimental analysis using the SimIC is based on a large-scale inter-cloud setting where the implemented algorithms offer optimization of various criteria including turnaround times and energy consumption rates. Results obtained are very supportive.

1. Introduction

In recent years, users have increased the scale of the distributed systems due to the increased resource utilization. Clearly, the increased sum of messages that are exchanged between users and system entities have led to an increased energy consumption. Current solutions are focused on the optimization of the performance measures (e.g. resource discovery and scheduling) without focusing on the benefits that may derive from the introduction of a new message exchanging approach. This affects the amount of messages sent and received with regards to the energy consumption of nodes. This area emphasizes an increasing trend that shifts the focus from improving performance to optimizing the energy efficiency and performance of the system as a whole [5,8]. By planning an energy efficient computational model along with performance optimization we can reduce the consumption rates while at the same time increase the user satisfaction. In distributed systems, the performance includes the computational metrics (e.g. execution times) while the energy efficiency includes consumption of the interacting nodes.

Our contribution is by improving the energy consumption rates based on the minimization of the sum of messages that are exchanged during the resource management phase. We define as message exchange in distributed systems the communication among entities (nodes). To achieve this, we introduce a novel message-exchanging optimization (MEO) model in [18] to optimize performance of distributed systems. In addition, we focus on the energy efficiency and we present the performance evaluation of our approach in this direction. Current efforts include various nodes communicating with each

* Corresponding author.
E-mail addresses: n.bessis@derby.ac.uk (N. Bessis), s.sotiriadis@derby.ac.uk (S. Sotiriadis), florin.pop@cs.pub.ro (F. Pop), valentin.cristea@cs.pub.ro (V. Cristea).

1569-190X/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.simpat.2013.02.003

other in order to request for services by sending messages, however without considering the number of messages. We start with a related works section (Section 2) to present current approaches to and challenges in message exchanging in distributed systems. Based on this analysis, we conclude with a critical discussion of the key characteristics for messaging.

To demonstrate the message approach in a real-case scenario we implement an inter-cloud facility, where various services are submitted from users to clouds for execution. In Section 3, we present our message-exchanging optimization (MEO) model with a specific focus on how to optimize the performance and the energy efficiency of job distribution. We illustrate the mathematical representation based on graph theory in order to determine relationships and structures of nodes (entities of the distributed system). In Section 4, we present the algorithm pseudo-codes to show the processes of message exchanging along with the performance measures. The rest of the paper is organized as follows; the configuration of the setting and the inter-cloud cases (Section 5.1) are included in the simulation toolkit (SimIC) that implements the algorithms (Section 5.2) and produces the experimental results (Section 5.3). The study concludes with a critical evaluation and discussion of future work (Section 6).

2. Related works

Currently, most message solutions are focused on the number of packets that are moved among processors of clusters and grid settings. In such cases the Message Passing Interface (MPI) [11] has been introduced as a portable message passing standard. The MPI is related with the point-to-point communication and the collective communication approach. The first case (point-to-point) includes that the processor of a node sends a message to another along with some data. MPI processes are independent and they communicate to coordinate a job submission, so messages are sent between two processes. The actual operation includes that one process sends a message to another one that receives it, and then it replies to the sender. A typical rule is that for a point-to-point MPI case the number of messages that are sent and received should match (one receive per send).

Each message contains a number of properties that include the actual data, the data type of each element, the number of elements, a message tag, and the ranks of the source and destination process [1]. This kind of exchange could occur in synchronous or asynchronous mode. For instance, the synchronous mode includes the sending of complete information while the asynchronous contains data regarding the time that the message left from the first process. In addition, asynchronous allows high dynamicness of the system, so it could be applied successfully for cases of variable workloads [12]. Point-to-point has been considered as a flexible method for messaging, however for a large number of processes the sum of messages will be high; a fact that affects the overall system performance. This shortcoming is based on the request to response model and includes that for each request a process should always reply.

A different approach is the collective communication that involves the transfer of many processes at a time; so, we could have coordinated communication of a group of processes. The solution increases the design complexity as it encompasses the synchronization of processes. Nevertheless, it is a more advanced approach which in operation could be applicable for larger scale distributed infrastructures. The collective communication is always synchronous in the sense that collection will not be completed until all MPI nodes reach the same point [19]. In general, by using broadcasting named as "broadcast call" one node sends a message to all nodes of the group. The "reduce call" procedure is called by the MPI at the end of the process for collecting information from all nodes' processors and stores the result on one node [2]. The collective functions follow the basic MPI requirement that denotes that the amount of data sent should match the amount of data as specified by the receiver [15], thus this makes it an inflexible solution.

For the collective communication procedures, a variety of different routines have been introduced to implement different communication patterns [11]. This includes AllToAll, AllGather, BCast, Scatter and Gather, AllReduce and other functions. Initially, the AllToAll model allows complete information distribution among all the node processes of a group [19]. This model forms the basic communication pattern that is used from most distributed systems, see, [22–25]. This is because the focus of such systems is on the scheduling aspects rather than on communication. Based on that, we also focus on utilizing the AllToAll solution for our experimental analysis. AllGather, on the other hand, collects processes and then broadcasts (BCast) to each conducted node. However, as the number of nodes increases, the performance is compromised due to the congestion of network resources. Finally, the Scatter and Gather method allows collective processes to be distributed in a different process and gather again back to the origin processor. A different view of messaging could include the network algorithms for minimizing the network bottleneck.

A number of theoretical models have been developed in order to avoid network congestion [1], however from the perspective of packets that are exchanged among network entities. For example, the spreading simple algorithm allows a node to send data to node as node $(p + i)$ where p is the process and i the iteration and, receive data from node $(p - i + N) \bmod N$ (where mod is the division modulo) [1]. A different approach called the ring/bucket/circular algorithm is presented in [3]. Specifically, at each iteration i a process p sends data to a node with an index $(p - i + 1 + N) \bmod N$ to the right neighbour on the list. The recursive doubling algorithm [19] requires less time as the number of total transfers is reduced. The MPI makes use of the MPICH [11] (MPI Chameleon) to recursively reduce the number of messages by utilizing a criterion: when the number of processes is a power of 2 it uses recursive doubling for small message sizes. Next, for the rest of the messages (large size) it uses the ring algorithm to achieve message dissemination. However, this solution is aimed at small-scale [9] based parallel computing systems. Steffenel and Jeannot [2] propose that most of these algorithms have been designed for homogeneous and tightly coupled systems.

Please cite this article in press as: N. Bessis et al., Using a novel message-exchanging optimization (MEO) model to reduce energy consumption in distributed systems, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.02.003