Chapter 69
An Efficient Server Minimization Algorithm for Internet Distributed Systems

Swati Mishra
Department of CSE and IT, Veer Surendra Sai University of Technology, Burla, India

Sanjaya Kumar Panda
Department of CSE and IT, Veer Surendra Sai University of Technology, Burla, India

ABSTRACT

The increasing use of online services leads to an unequal distribution of the loads among the servers. As a result, the problem is to balance the loads among the servers such that the total number of active servers is minimized. One of the possible solutions is to transfer the loads from the underutilized server to a suitable server and make the underutilized server to sleep mode. In this paper, a server minimization algorithm (SMA) is proposed for the solution of server minimization and the load balancing problem. The proposed algorithm reduces the number of servers by merging the loads of the two least loaded servers. Then it determines the standard deviation of the server loads for load balancing. The proposed SMA is compared with an existing load balancing algorithm using the number of minimized servers, load standard deviation and load factor. The simulation results show the efficacy of the SMA.

INTRODUCTION

In Internet Distributed System (IDS), the autonomous computers are combined together to share a large pool of computing resources among themselves (Nishida, & Nguyen, 2013). These computers are mainly classified into clients and servers. The interaction between client and server takes place via request and response messages. In general, a client request for a service through a request message and a server acknowledges the corresponding message through a response message. In order to fulfill all the client requests, a server may request the resources to other servers by changing its type to the client. However, this request is served if and only if the other servers are not overloaded. This message com-

DOI: 10.4018/978-1-7998-5339-8.ch069

munication between clients and servers aim to balance the loads of the servers by equally distributing the client requests. It is essential as a server can accommodate only a limited number of client requests at a particular time (Li et al., 2012; Panda, & Jana, 2015a). This is referred as the load balancing problem in client-server assignment problem and it is a well-known NP-Hard problem (Shi, & Malik, 2000; Fei, Ammar, & Zegura, 2002; Nishida, & Nguyen, 2013; Ba, Nguyen, Bose, & Tran, 2014). On the other hand, load balancing does not imply minimization of servers. For instance, the loads are equally distributed among the servers, but all the servers are under loaded. In this case, it is better to sleep (or switch off) few servers and transfer their corresponding loads to the other servers. It results following advantages. (1) The total number of active servers is minimized. As a result, the energy consumption, resource management and costs are minimized. (2) The active resources are properly utilized.

The problem of server minimization is discussed as follows. Given a set of m servers with their corresponding loads, the primary objective is to minimize the total number of active servers such that the total load of the active servers is approximately equal. An algorithm called a server minimization algorithm (SMA) is introduced to the solution of the above discussed problem. SMA consists of a two-phase process, namely load balancing and server minimization. The basic idea of SMA is explained as follows. (1) Load Balancing: SMA determines the total load of each server. Then it finds the standard deviation of the total loads of the m servers. (2) Server Minimization: Next it merges the total loads of two least loaded servers and makes the least loaded server in sleep mode. If two least loaded servers are holding same load, then one of them is randomly selected to break the tie for sleep mode. Then it again finds the standard deviation of the total loads of the $m - 1$ servers. (3) This process is repeated until the total number of servers is reduced to three. Here, a set of server loads which result in minimum standard deviation is selected for server minimization. Note that minimum standard deviation indicates that the variation of the loads are minimized. The evaluation process of SMA is carried out using ten different datasets. The performance is determined using three different metrics, number of minimized servers, load standard deviation and load factor. The outcomes of the simulation show that the SMA balances the loads of the servers and minimizes the number of active servers in compare to the existing client-server load balancing algorithm (Nishida, & Nguyen, 2013; Bharati, Naidu, Kiran, Khune, & Vyas, 2014; Mishra, Panda, & Das, 2017).

Our major contributions are described as follows.

- Development of a server minimization algorithm for IDS
- Simulation of the proposed SMA with a wide set of clients and servers
- Evaluation of the proposed SMA in terms of three performance metrics, namely number of minimized servers, load standard deviation and load factor
- Comparison of the proposed SMA with an existing client-server load balancing algorithm

The remainder of this paper is organized as follows. The next section describes the related work in client-server load balancing problem. Then the system model and problem statement section presents the Internet distributed system model and the server minimization and load balancing problem. Then the proposed algorithm section presents the proposed algorithm SMA with a step-by-step illustration followed by various performance metrics to compare the existing algorithm and the proposed SMA in the performance metrics section. Simulation results are presented in the next section of the performance metrics section. Finally, the conclusion is presented in the last section with some future research directions.

Related Content