A dynamic resource management in mobile agent by artificial neural network

Yao-Tien Wang [a,b,*]

[a] Department of Information Management, Kainan University, Luzhu, Taoyuan County, Taiwan
[b] Department of Computer Science and Information Engineering, Kainan University, Luzhu, Taoyuan County, Taiwan

ARTICLE INFO

ABSTRACT

In this paper, a resource management for dynamic load balancing in mobile agent by artificial neural network scheme (ANN-DLB) is presented to maximize the number of the served tasks in developing high performance cluster. This dynamic load balance with the growth of the service type and user number in the mobile networks of the higher performance is required in service provision and throughput. Most of the conventional policies are used in load indices with the threshold value to decide the load status of the agent hosts by CPU or memory. The main factor influencing the workload is the competitions among the computing resources such as CPU, memory, I/O and network. There are certain I/O data of the intensive applications where load balancing becomes the important issue. This relationship between the computing resources is very complex to define the rules for deciding the workload. This paper proposed a new dynamic load balancing for evaluating the agent hosts' workload with the artificial neural network (ANN). By applying the automatic learning of the back-propagation network (BPN) model can establish the ANN model and also can measure the agent host loading with five inputs: CPU, memory, I/O, network and run-queue length. The structure of the load balancing system is composed of three design agents: the load index agent (LIA), the resource management agent (RMA) and the load transfer agent (LTA). These experimental results reveal that the proposed ANN-DLB yields better performance than the other methods. These results demonstrate that the proposed method has high throughput, short response time and turnaround time, and less agent host negotiation complexity and migrating tasks than the previous methods.

1. Introduction

The mobile agent (MA) contains software programs and data resources dispersed across the independent computers to connect through a communication network (Chung et al., 2008; Neeraj and Patel, 2007; Yu et al., 2009). Agent designs are increasingly prevalent in applications for computing. One important problem that arises in mobile agent is to improve the performance of a set of computing agent hosts (AHs) through dynamic workload redistribution. Some of the possible resources to be redistributed are computing power, data and hardware devices. To be more specific, load balancing is the process of redistributing the workloads submitted to a network of computers to avoid the situation where some AHs are idle while others are congested. It has been shown that a good load balancing algorithm produces considerable performance improvement. This is done by transferring tasks from MA to AH or over-loaded AHs to under-loaded AHs efficiently. Dynamic load balancing (DLB) reallocates the computing tasks, which are in accordance with the current

workload (Neeraj and Patel, 2007; Bahi et al., 2005; Chess et al., 1995; Kafeel and Ahmad, 1998; Fukushima, 1988; Ou et al., 2008). It is more suitable to be implemented in the real computing cluster environment (Patel and Garg, 2004; Raimundo et al., 2005). Hence, the proposed load balancing system is made in the dynamic model.

The ANN-DLB consists of (1) the load index agent (LIA), (2) the resource management agent (RMA) and (3) the load transfer agent (LTA) phases. The structure of a dynamic load balancing for mobile agent is composed of three design phases by applying ANN to them. (Chung et al., 2008; Neeraj and Patel, 2007). The load index agent indicates the amount of information about the system and the information gathering rules to be used in making the load redistribution decisions. The goal is to obtain sufficient information in order to make a decision whether an AH load is overloaded or not. The resource management agent selects the AHs to or from which tasks will be transferred when the load reallocation event takes place. The load transfer agent means how to transfer tasks from an AH to another one. The term task transferring implies the transfer of a task and continues the execution on a target AH.

Most conventional information policies use load indices with a threshold value to decide the load status of the AH by CPU or

memory (Neeraj and Patel, 2007). The main factor influencing the workload is competitions among the computing resources, such as CPU, memory, I/O and network. There are certain I/O data of the intensive applications where load balancing becomes an important issue. The relationship between the computing resources is very complex to define rules for deciding the workload. In this paper we propose a new dynamic load balancing for evaluating the MA and AHs workload with the artificial neural network (ANN). ANN has been proposed as a model for solving a wide variety of problems in such diverse fields as combinatorial optimization computing, vision, pattern recognition, etc. (Fukushima, 1988). Therefore, the ANN idea is used to solve the problem of estimating the load status of a computing host. Considering the load related information, namely CPU utilization, memory, I/O, network and run-queue length, the proposed LIA has the ability to represent the loads on all the resources of contention, which fluctuates in the execution time of computing tasks. By applying the automatic learning of the back-propagation network (BPN) model (Fukushima, 1988; Lin et al., 2009), we establish the ANN model to measure the agent host load with five input variables: CPU, memory, I/O, network, run-queue length and the three state tags as outputs: over-loaded, moderate-loaded and under-loaded. After ANN has been trained by an appropriate set of training data, the proposed scheme has the ability to reflect the instant workload as well as to predict it. Fig. 1 shows the block diagram of ANN-DLB. In this paper, an enhanced version of LTA with multi-task-transferring is proposed. The numbers of transferring tasks are decided by the run-queue length and a weighted value concerned with CPU utilization. It will involve more load balanced effects when the workload between the over-loaded and under-loaded agent host differs a lot. Load balancing is the capability of reducing the mean of the long-term computing tasks' execution time, especially when the system is over-loaded.

The organization of this paper is as follows. Section 2 describes an overview of the mobile agent and the agent host model and technology. The agent host workload measurement is discussed in Section 3. Section 4 discusses automatic learning of the load index agent with ANN. The proposed method of the resource management is in Section 5. The new tasks transferring is compared with the multitasks transferring of the load transfer agent in Section 6. The simulation model and results are given in Section 7. Concluding remarks are made in Section 8.

2. Mobile agent model and technology

The Platform for Mobile Agent Distribution and Execution with parallel computing using a system such as PMADE (Neeraj and Patel, 2007; Liu et al., 2005; Al-Jaroodi et al., 2003; Ephremides and Truong, 1990; Funabiki and Takefuji, 1993) may be approached from three fundamental viewpoints, based on the organization of the computing tasks. The most common model for PMADE applications can be termed crowd computing and the collection of the related processes closely and typically in executing the same code. They perform computations on the different portions of the workload, usually involving the periodic exchange of intermediate results. This paradigm can be further subdivided further more into two categories as follows:

(1) The MA–AH (or *host-node*) model in which a separate control program termed the MA is responsible for process spawning, initialization, collection and the display of the results, and perhaps the timing of functions. The AH programs perform the actual computation involved; they are either allocated their workloads by the MA (statically or dynamically) or perform the allocations themselves.
(2) The *AH-only* model where multiple instances of a single program are executed, with one process (typically the one initiated manually) taking over the non-computational responsibilities in addition to contributing to the computation itself.

Crowd computations typically involve three phases. The first is the initialization of the process group; in the case of AH-only computations, dissemination of group information and problem parameters, as well as workload allocation, is done typically within this phase. The second phase is computation. The third
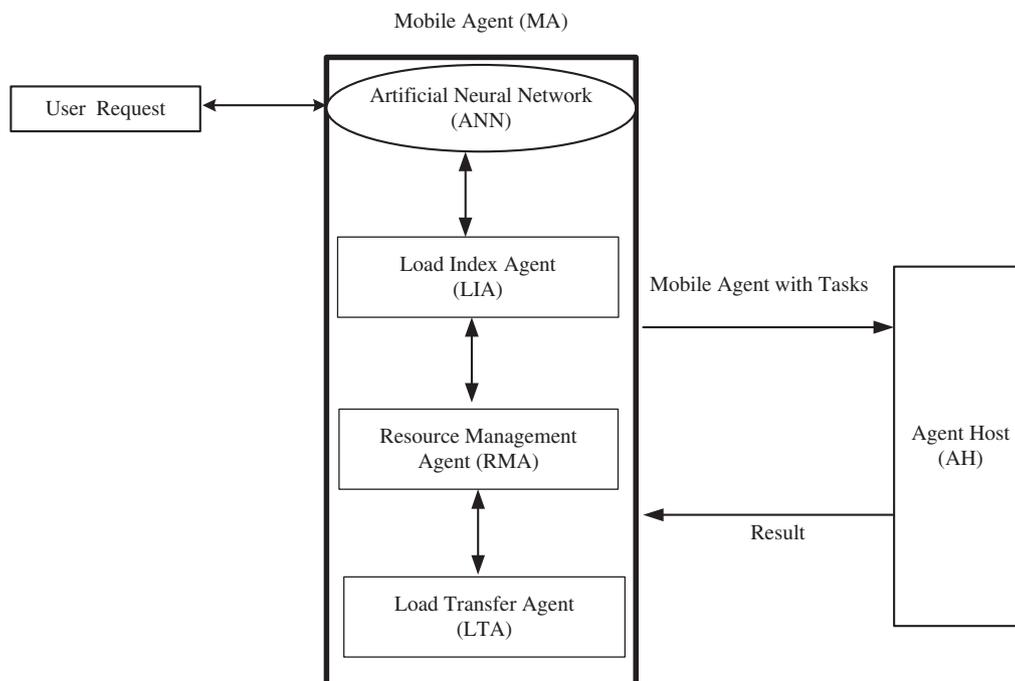

Fig. 1. Block diagram of ANN-DLB.