Information Sciences 221 (2013) 297–305

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Convergence of the dynamic load balancing problem to Nash equilibrium using distributed local interactions

Sameena Shah [a,*], Ravi Kothari [b]

[a] Thomson Reuters Corporate R&D, 195 Broadway, NYC, NY 10007, USA
[b] IBM India Research Lab, 4 Block C, Institutional Area, Vasant Kunj, New Delhi 110 070, India

ARTICLE INFO

Article history:
Received 13 July 2009
Received in revised form 2 September 2012
Accepted 9 September 2012
Available online 24 September 2012

Keywords:
Load balancing
Diffusion process
Nash equilibrium
Distributed algorithms
Agent based systems
Local information

ABSTRACT

Load balancers distribute workload across multiple nodes based on a variation of the round robin algorithm, or a more complex algorithm that optimizes a specified objective or allows for horizontal scalability and higher availability. In this paper, we investigate whether robust load balancing can be achieved using a local co-operative mechanism between the resources (nodes). The local aspect of the mechanism implies that each node interacts with a small subset of the nodes that define its neighborhood. The co-operative aspect of the mechanism implies that a node may offload some of load to its neighbor nodes that have lesser load or accept jobs from neighbor nodes that have higher load. Each node is thus only aware of the state of its neighboring nodes and there is no central entity that has the knowledge of the state of all the nodes. We model the overall mechanism of load balancing based on local interactions as a congestion game and show that convergence to the Nash equilibrium is possible using only local interactions. We derive worst case bounds on the number of transfers (time) required to achieve global load balancing under this setup. We also include simulation results to demonstrate emergent global load balancing based only on local interactions and local information.

1. Introduction

Load balancing distributes workload across multiple resources to allow for horizontal scalability and better availability. There are a large number of situations where load balancing is useful. For example, in a distributed computing environment load balancing may occur across servers, across processors in a single server in a super-computer, across network links or other resources.

Typically, a load balancer relies on global information about the state of all resources and uses this knowledge judiciously to assign a job to a resource in order to optimize a specified objective. The load balancing algorithm could be as simple as round robin (or one of it is several variants), it could be based on the content of the incoming requests (as in layer-7 load balancers), or any other variation. The specific algorithm, type of resources or exact nature of information used is not central to this paper. Our focus is on a crucial macro-level question, *whether global information is required to achieve global load balancing?*

Our motivation for investigation of necessity of using global information to achieve global load balancing is possibly obvious. Global information requires that each resource transmit its present state to the load balancer. This requires an extensive interconnect system to ensure that all nodes are able to update the state at the same time. If the interconnect topology has

* Corresponding author. Tel.: +1 646 822 2246.
E-mail addresses: sameena.shah@thomsonreuters.com (S. Shah), rkothari@in.ibm.com (R. Kothari).

0020-0255/$ - see front matter
http://dx.doi.org/10.1016/j.ins.2012.09.004

non-uniform paths to the load balancer, then it also requires dealing with states of the resources corresponding to different points in time. In either case, the amount of information related to load balancing increases in proportion to the number of resources. These issues along with the susceptibility to the faults and down-times of the load balancer itself are limiting for very large systems. We thus investigate whether global information is required to achieve global load balancing. To examine this in a more formal setting, we obtain performance characteristics in a game theoretic setting. In the sequel, we use a job (or task) to refer to work that must be performed by one of the resources (nodes). Jobs can migrate from one node to another. We do not consider the mechanics of the actual physical transfer.

We lay out the rest of the paper as follows: In Section 2, we review some of the existing approaches to load balancing. In Section 3, we state our assumptions, describe attributes of the model in a game theoretic setting and develop the appropriate game. In Section 4, we give several local interaction based approaches to load balancing (each resource can exchange load only with its immediate neighbors) and examine convergence properties in a game theoretic setting. We show that under a key condition, these algorithms converge to the Nash Equilibrium. We also compute the quality of equilibrium and rate of convergence. In Section 5, we provide simulation results for multiple interconnect models and balancing schemes. In Section 6, we present our conclusions.

2. Background literature on load balancing algorithms

The simplest algorithm for load balancing is round-robin allocation. Assuming N nodes that are indexed as $n_1, n_2, \ldots, n_N$ and the last task is assigned to node n_i, the next task is assigned to the node indexed as $1 + (i+1)\text{mod}N$. Though simple to implement, round-robin uses global information that has information about all resources. Variants of round-robin algorithm may require information beyond the availability of the resource, for example, the load on the resource.

In the context of jobs being decomposable, with sub-tasks being assigned to different resources, load balancing may be viewed as a graph coloring problem [30]. Assume that there is an undirected graph of T tasks. Color these nodes with N colors (nodes) so as to minimize a cost function J as follows,

$$J = J_{comm} + \lambda J_{comp} \tag{1}$$

where J_{comp} represents the computational aspect of the cost function and is minimized when each node has an equal amount of work. J_{comm} represents the cost of inter-node communication and λ represents the relative weight assigned to computation and communication costs. Here, global information is needed to construct the graph. An optimization procedure (for example, simulated annealing) may then be used to obtain the mapping of the sub-tasks onto given nodes.

In [1], a semi-distributed, two level load balancing strategy using global information was proposed. The network is divided into multiple spheres (or groups). At the first level, load is balanced amongst different spheres of the system. At the second level, load is balanced within the individual spheres in a centralized manner.

In the context of peer-to-peer systems, the notion of a Distributed Hash Table (DHT) is used [11,12] where each node is made responsible for a balanced portion of the DHT address space, and a random mapping of the requests ensures that a limited number of tasks end up at any node. Lee et al. [16,17] propose an algorithm based on symmetric balanced incomplete block design that distributes global information to nodes efficiently.

The Messor Algorithm [2,23,24] for load balancing using local information has been inspired from the concept of 'Stigmergy' or indirect communication by ants to disperse objects. Each ant moves independently between the nodes and transfers jobs between *seemingly* over-loaded and under-loaded nodes. A node *seems* over-loaded (or under-loaded) to an ant if it is over-loaded (or under-loaded) compared to the nodes most recently visited by it. Simulation results showed that using such a scheme, perfect load balancing can be achieved. However, no guarantees of convergence to a load balanced situation or time bounds were given.

Subsequently, using the macroscopic model of [18,19] to model complex agent based systems, Liu et al. [20] showed that the Messor algorithm will reach a stable state; but this stable state will correspond to a perfect load balanced state only in the limit that all ants have complete information about all the nodes. Hogg and Huberman [10] showed that since agents have incomplete information about nodes, load balancing converges under some conditions, but oscillates and is chaotic under other conditions.

In recent years, there has been an interest in analyzing distributed systems in terms of game theoretic and market oriented models [26]. Stability of the network is analysed in terms of reaching the Nash Equilibrium, a load distribution in which no single job can move to any other node with lesser number of jobs.

Khan and Ahmad [14] develop a non-cooperative game theoretic model for the replication of data objects across a system of multiple servers. Grosu and Chronopoulos [7] model the static distributed load balancing problem as a non-cooperative game and show existence of a unique Nash Equilibrium for the assumed load and node behavior. Grosu et al. [8] model static load balancing with load information and arrival times known *a priori*, and nodes co-operating with each other as a co-operative game. Subrata et al. [27] propose a semi-static load balancing algorithm that accounts for communication delays as a non-cooperative game and show its convergence to Nash Equilibrium.

A semi-static, cooperative game theoretic based solution for job allocation on a quality of service sensitive grid is proposed in [28]. Khan and Ahmad [13] simultaneously minimize make-span and energy consumption subject to deadlines and tasks' architectural constraints. They develop a cooperative model and derive the runtime complexity of the algorithm.