Chapter 5
Policies for Advanced Dynamic Load Balancing in Cloud Computing

Dharmesh Dhabliya
https://orcid.org/0000-0002-6340-2993
Department of Information Technology, Vishwakarma Institute of Information Technology, India

Sukhvinder Singh Dari
https://orcid.org/0000-0002-6218-6600
Symbiosis Law School, Symbiosis International University, India

Nitin N. Sakhare
https://orcid.org/0000-0002-1748-799X
Department of Computer Engineering, BRACT'S Vishwakarma Institute of Information Technology, India

Anish Kumar Dhablia
Altimetrik India Pvt. Ltd., India

Digvijay Pandey
https://orcid.org/0000-0003-0353-174X
Department of Technical Education, Government of Uttar Pradesh, India

A Shaji George
https://orcid.org/0000-0002-8677-3682
Almarai Company, Riyadh, Saudi Arabia

A. Shahul Hameed
Department of Telecommunication, Consolidated Techniques Co. Ltd., Riyadh, Saudi Arabia

Pankaj Dadheech
https://orcid.org/0000-0001-5783-1989
Swami Keshvaand Institute of Technology, Management, and Gramothan, India

ABSTRACT

Cloud computing involves virtualization, distributed computers, networking, software, and web services. Clouds have servers, datacenters, and customers. It has fault tolerance, high availability, scalability, flexibility, little user overhead,

DOI: 10.4018/979-8-3693-0819-6.ch005

low ownership cost, on-demand services, etc. These issues demand a robust load balancing mechanism. Balanced load distribution improves resource use and task response time by preventing some nodes from being completely loaded and others idle. Load balancers match processor and network node performance. A load balancing solution that improves throughput and latency for application-based virtual topologies with variable cloud sizes will apply the divisible load scheduling theorem.

INTRODUCTION

By dispersing the entire load throughout the numerous nodes that make up the collective system, this approach hopes to cut down on the amount of time needed to complete a task while simultaneously improving the effectiveness with which resources are put to use. According to Yu, Shucheng, et al. (2010), the goal of this method is to cut down on the amount of time needed to complete the work. Because of this, there is no longer a chance that some nodes may be overloaded as well as underloaded at the same time. This predicament may arise if a few of the nodes are carrying more weight than they should be at any given time (Goyal, S., et al. (2022)). These objectives can be accomplished by enhancing not only the responsiveness of the system but also the effectiveness with which it makes use of the resources that are already available (Boopathi, S., et al. (2023)). A dynamic load balancing technique is one that is defined by the fact that it is based on the behaviour the system is displaying at the moment the approach is being implemented, rather than on the state of the system or its behaviour at any previous time. This is in contrast to a static load balancing strategy, which is defined as one that is based on the behaviour the system exhibited at any earlier time. This indicates that the system's past state or behaviour is not taken into consideration in any way, and that this is the case regardless of the situation (Pramanik, S. et al. (2023)). When developing such an algorithm, some of the most important factors to take into account include the following: load estimation, load comparison, system performance, stability of various systems, node interaction, the kind of work that is to be transferred, node selection, and a whole host of other variables (Shang, Ning, et al. (2010)). This load that is being investigated may be assessed in terms of the quantity of memory that is being consumed, the latency, or the stress that is being placed on the network. All three of these metrics may be taken into consideration. These three strategies are all workable options. There is a possibility that each of these factors will be taken into consideration.