Review of ”Load Balancing in Cloud Computing: A State of the Art Survey - Mohammadreza Mesbahi, Amir Masoud Rahmani”

By Birta Alexandru

In this paper review I will present the survey done by Mohammadreza Mesbahi and Amir Masoud Rahmani on the various technologies used to achieve load balancing in the field of cloud computing most commonly used in today's systems. As they posed in the paper, cloud computing has taken to great heights utilizing various techniques to achieve load balancing in these virtual distributed and parallel systems bringing a very sought after quality of service for most big companies utilizing these technologies. Next in the review I will present the various necessities and requirements as well as the types of cloud balancing solutions explored in the paper. The main categories for the solutions proposed by the authors are artificial intelligence based and general algorithm based.

Cloud computing is a common and relevant concept in today's IT world. The primary purpose of cloud computing is to make better use of dispersed resources and apply them to obtain higher throughput, performance, and to solve large-scale computing challenges. In general, based on the *NIST* definition of cloud computing, we can conclude that one of the most significant aims of this model is to make the greatest effort to provide on-demand services while making the best use of available shared resources.

Load balancing is a significant obstacle and challenge in distributed systems such as grid computing and cloud computing. It is still a fresh problem in cloud computing that needs the creation of new architectures and algorithms to improve on previous ones. Because the system's dynamic on-demand network access feature makes it difficult to predict the arrival pattern, type of service, and other properties of many jobs in cloud computing, an efficient load balancing mechanism is critical to increasing the Service Level Agreement (*SLA*), delivering a robust service, and meeting other system requirements. Furthermore, load balancing is an important issue since it enables for other crucial factors like scalability.

Because of technological advancements in the computer world and the rapid growth of distributed systems in recent years, the load balancing problem has been recognized as a major concern in these systems more than ever before. The next section of the paper discusses some key principles and methods to load balancing mechanisms. Load balancing is the process of redistributing the general system work load among all nodes of a distributed system (network links, disk drivers, central processing units, etc.) in order to improve resource utilization and job response time while avoiding situations in which some nodes are overloaded while others are under-loaded or idle.

Load balancing is an important aspect of cloud computing and elastic scalability. It is commonly used to prevent system failure by controlling input traffic and ceasing to provide workload to resources that become overburdened and unresponsive. This is a trait carried over from grid-based computing to cloud computing. The main goals of load balancing advancements posed in major research to date are: *reducing task response time while maintaining acceptable delays*, *keeping the system stable*, *having the capacity to tolerate faults using load balancing to perform failover*, *improving overall system performance in order to achieve optimal resource use, maximum throughput, and minimize overload and improving and sustaining cloud system availability*.

There are several load balancing strategies and procedures, and most research has divided them

into two categories: static and dynamic. Static techniques usually contain prior information and assumptions about the global state of the system, such as task resource demands, communication time, processing power of system nodes, memory and storage device capacity, and so on.

A static approach is a method of assigning a set of tasks to a set of resources that can be either deterministic or probabilistic in nature. Furthermore, this strategy is commonly discussed in system design or implementation. The fundamental disadvantage of static load balancing algorithms is that they do not take into account the present state of the system while making choices, making them unsuitable for systems such as distributed systems, where the majority of the system's states change dynamically.

Approaches to dynamic load balancing take into consideration the current state of the systems on which their judgments are based. The primary advantage of dynamic load balancing systems, which may change dynamically dependent on the present state of the system, is that workloads can be dynamically transferred from an overloaded node to an underloaded one. Nonetheless, developing and executing a dynamic load balancing algorithm is significantly more extensive and challenging than locating a static solution; however, by employing dynamic mechanisms, we may get higher performance as well as more accurate and efficient solutions.

There are two types of dynamic load balancing algorithms: distributed and non-distributed. In distributed techniques, for example, the load balancing procedure can be carried out by all nodes in the system. Furthermore, with this strategy, all nodes can interact with one another to achieve a global objective in the system, which is known as cooperative, or each node can operate independently to achieve a local goal, which is known as non-cooperative. In a non-distributed design, however, the job of balancing the system burden is not shared by all system nodes. A single node can only perform the load balancing process across all nodes in a centralized method in a non-distributed scheme. In semi-distributed mode, the system is divided into divisions or clusters, with a single node doing load balancing in each partition. These two kinds' distributed, dynamic, and adaptive load balancing algorithms are better suited for large-scale distributed systems like cloud computing. The adaptive technique adapts load distribution to changes in system status by modifying their parameters and even their algorithms dynamically.

According to the authors' earlier findings, dynamic load balancing procedures are more appropriate solutions in large scale distributed systems such as grid and cloud computing. Dynamic load balancers require information about the status of the system in order to make decisions depending on the current burden of the system. As a result, specific components for obtaining and managing this sort of information are required for a dynamic load balancing method. Next I will introduce the four main components of making a dynamic load balancer according to the paper.

The information strategy component of a dynamic load balancer is in charge of gathering information about the state of resources in a system. The load balancing mechanism in a distributed system will be able to perform efficiently based on the whole information acquired from each processing node. There are numerous techniques for gathering information from system nodes. Broadcasting, centralized polling, and agents are three of the most prevalent approaches.

A triggering strategy component is a component of a dynamic load balancer that identifies the right moment to begin a load balancing operation. Resources are categorized into two types based on the type of trigger policy: sender and recipient. As a result, load balancing techniques will be referred to as sender-initiated and receiver-initiated, respectively.

The transfer strategy component is a component that is in charge of selecting a job for transfer to another resource. The last-received task and all-current-tasks methods are two general ways for determining which task should be transferred. In the first method, which is a basic solution, just a newly arrived task is chosen for transfer. However, in the following step, an intelligent judgment is made based on specific factors to choose a job from among all existing node jobs.

The location strategy component determines a target resource and a computing node for task transfer. There are several methods for picking a location, including probing, random, and negotiation.

A local node often probes the system nodes to discover a suitable destination in the probing strategy. A location method that uses a random approach chooses a destination node at random. Finally, in negotiation techniques, nodes bargain with one another for the load balancing process.

Now we have reached the second part of the paper review in which I will summarize in a few examples the major two types of cloud load balancing algorithms which are in two categories: *general algorithm based (abbreviated GAL in this paper)* and *artificial intelligence based*.

General algorithm based load balancing mechanisms are those that present a comprehensive method and examine all aspects of the load balancing algorithm therein. Based on the suggested technique, a load balancing system may be developed or simulated in this category. These techniques often suggest a load balancing solution without taking into account specific cloud architecture and are given in a generic manner. In other words, certain GAL-based load balancing techniques are some traditional load balancing methods that are analogous to allocation and scheduling methods in operating systems, such as Round-Robin, First Come First Serve (FCFS), Minimum Execution Time (MET), and so on. Now I will briefly review some methods.

Roundrobin is a well-known and easy mechanism for allocating workloads to servers, and it often performs well in systems with modest workload.

Weighted round-robin is a method, which is adapted from classic round-robin, outperforms traditional round-robin by assigning more weights to servers with superior performance. As a result, the more competent computer nodes will receive greater workloads.

Least-connection is a method which dynamically counts the connections connected with each server and then selects the server with the lowest count and allocates new incoming workloads to the server with the lowest connection.

Shortest expected delay is a method that considers the last response time for each server and then selects the server with the shortest response time as the next appropriate server.

Load balancing mechanisms based on artificial intelligence are load balancing methods whose fundamental principles are based on Artificial Intelligence concepts. In other words, AI-based techniques present a solution for balancing workloads in cloud computing environments by exploiting similarities between established artificial intelligence algorithms and methodologies and cloud computing components and ideas. Arch-based AI-based load balancing techniques can be developed.

An AI-based load balancing algorithm example is Kumar Nishant et al.'s proposed modified Ant Colony Optimization (*ACO*). *ACO* is used for cloud computing load balancing to ensure that the system continues to work properly even during peak usage hours. This AI-powered solution includes a head node that creates ants. Ants travel the width and length of the cloud network in order to determine the location of under- or over-loaded nodes. These ants' movements update a pheromone database, which stores information about resource consumption. These ants can move either forward or backward and loads will be moved from overloaded nodes to underloaded nodes based on the above moves and the pheromone table.

In conclusion, in recent years, load balancing in cloud computing data centers has been a major concern and an important topic of research. The paper's authors offered a study of current load balancing approaches and solutions that have been proposed solely for cloud computing settings in this research. Based on the author's findings, cloud load balancing mechanisms may be divided into two major classes based on their design perspectives: load balancing techniques based on general algorithms and artificial intelligence. Also each of these approaches are deeply explored in the paper with their detailed implementations, use cases and the system requirements that are needed for their efficient usage.