Chapter 7

Mutation-Based Glow Worm Swarm Optimization for Efficient Load Balancing in Cloud Computing

Avtar Singh
https://orcid.org/0000-0001-7526-6813
National Institute of Technology, Jalandhar, India

Shobhana Kashyap
National Institute of Technology, Jalandhar, India

ABSTRACT

Cloud computing has evolved as an innovation that facilitates tasks by dynamically distributing virtual machines. User has to pay for the resources as per the demand. This is a challenging task for cloud service providers. The problems caused in load balancing are selecting random solutions, low speed convergence and picking up the original optima. To attain the best result, a mutation-based glow worm swarm optimization (MGWSO) technique is proposed. With this method, the makespan is reduced for a single work set across multiple datacentres. The work is motivated to decrease the consumption of resources in dynamic contexts while simultaneously increasing their availability. The simulated result shows that the suggested load balancing method dramatically reduces makespan in comparison to mutation-based particle swarm optimization.

INTRODUCTION

The cloud is the next computer paradigm and the next step in the evolution of information technology. It provides access to vast amounts of online storage and processing power on a pay-as-you-go basis. Problems like finding what you need; handling failures, spreading the load evenly, and keeping your data secure are magnified in the cloud (Dave et al., 2016). Distributing the burden or job evenly among the

DOI: 10.4018/979-8-3693-0900-1.ch007

nodes or servers is a significantly difficult and crucial function, for which load balancing is essential. It is the key difficulty with cloud computing.

The word "load balancing" refers to distributing the workload among multiple processors. It helps to utilize computing resources efficiently. It evenly distributes the workload, preventing any server from being overloaded while others are idle. This maximizes resource utilization and reduces costs (Kumar & Kumar, 2019). Its additional goals include prioritizing activities that need immediate execution above those that can wait and enabling scalability & flexibility for growing-in-size applications over time and so need more resources. In order to adjust load balancing, it must be able to reduce energy consumption, prevent bottlenecks, provide assistance, and fulfil quality of service (QoS) standards (Chen et al., 2017). The load balancer divides client demands into many servers. Virtual machine management (VMM) and hypervisors are used to implement the virtualization concept.

The three cloud computing deployment options are SaaS, PaaS, and IaaS. In SaaS, users rent software or applications from a cloud service provider. Users can utilize the provider's platform to build and launch their own applications using Paas model. IaaS provides computing resources over the internet, allowing users to acquire processing power, data storage, and other infrastructure components. Load balancing can be performed in static or dynamic cloud environments. Static load balancing maintains constant inputs and outputs throughout execution, while dynamic load balancing adapts to changing resource availability and workload demands. Static load balancing methods may not be suitable for dynamic environments, which require load changes during execution (Kaur, 2017).

Glow Worm Swarm Optimization (GWSO) is a Particle Swarm Optimization (PSO) variant that is utilized for optimization issues in a variety of disciplines, including cloud computing. It is more scalable because the control decentralised at various nodes. GWSO is based on how glow worms behave. GWSO is based on the behaviour of glow-worms. A glow worm with a high amount of light production (high luciferin) is better positioned and has a higher objective function value. Each glow worm chooses a neighbour with a higher luciferin value than its own and moves in that direction, according to probabilistic estimation. These movements are entirely based on local knowledge. As a result, the glow worms can form smaller groups, making it possible to discover various optima for the stated goal function. The GWSO algorithm starts by randomly distributing glow worms over the workspace, each with an equal amount of luciferin. A GWSO algorithm includes four phases: initialization, luciferin updating, movement, and local radial range updating. The algorithm is population-based. GSO is a metaheuristic method inspired by glow worm activity.

The proposed approach tackles the cloud computing load balancing issue by lowering the makespan time with the MGWSO technique. The result of this work shows that MGWSO has a lower makespan than the MPSO method. The fitness feature is also enhanced further it eliminates dummy jobs.

1. The chapter identifies and addresses the challenges faced by cloud service providers in load balancing, including the issues of selecting random solutions, low convergence speed, and the difficulty in identifying the original optima. By highlighting these challenges, the chapter establishes the necessity for innovative load balancing techniques in dynamic cloud environments.
2. The chapter introduces a mutation-based technique for load balancing, leveraging the GWSO metaheuristic. This approach represents a novel contribution to the field, as it enhances traditional load balancing methods by incorporating mutation strategies inspired by natural processes. The integration of GWSO introduces adaptability and self-learning capabilities, enabling the system to explore and exploit the solution space effectively.

Evaluating the Performance of Monolithic and Microservices Architectures in an Edge Computing Environment
Nitin Rathoreand Anand Rajavat (2022). *International Journal of Fog Computing (pp. 1-18).*
www.irma-international.org/article/evaluating-the-performance-of-monolithic-and-microservices-architectures-in-an-edge-computing-environment/309139