
Subhadarshini Mohanty, Department of Computer Science and Engineering, Siksha ‘O’ Anusandhan University, Bhubaneswar, India
Prashanta Kumar Patra, Department of Computer Science and Engineering, College of Engineering and Technology Bhubaneswar, Bhubaneswar, India
Subasish Mohapatra, Department of Computer Science and Application, College of Engineering and Technology Bhubaneswar, Bhubaneswar, India
Mitrabinda Ray, Department of Computer Science and Engineering, Siksha ‘O’ Anusandhan University, Bhubaneswar, India

ABSTRACT

Cloud computing is gaining more popularity due to its advantages over conventional computing. It offers utility based services to subscribers on demand basis. Cloud hosts a variety of web applications and provides services on the pay-per-use basis. As the users are increasing in the cloud system, the load balancing has become a critical issue. Scheduling workloads in the cloud environment among various nodes are essential to achieving a better Quality of Service (QOS). It is a prominent area of research as well as challenging to allocate the resources with changeable capacities and functionality. In this paper, a load balancing algorithm using Multi Particle Swarm Optimization (MPSO) has been developed by utilizing the benefits of particle swarm optimization (PSO) algorithm. Proposed approach aims to minimize the task overhead and maximize the resource utilization in a homogenous cloud environment. Performance comparisons are made with Genetic Algorithm (GA), Multi GA, PSO and other popular algorithms on different measures like makespan calculation and resource utilization.

KEYWORDS

Cloudlets, Genetic Algorithm (GA), Load Balancing, Makespan, Multi PSO (MPSO), Particle Swarm Optimization (PSO), Virtual Machine (VM)
INTRODUCTION

Cloud computing is extensively adopted through a wide range of users for providing a solution to massive length computational problems. Cloud environment includes heterogeneous computing resources along with processors, bandwidth, and memory. Cloud computing environment can create new application instances to provide distributed services within the form of infrastructure, platform, and software on demand basis with the help of virtualized resources. The VM migration is performed by moving the live VMs on execution from one physical machine to another machine to provide a better resources like large memory, high bandwidth, and computational power, (Jin et.al. 2011), (Mann et.al. 2015). The VM migration technique helps in remapping the digital machine and physical resources dynamically with flexible allocation and reallocation of the resources (Clark et.al. 2005), (Jun et.al. 2011).

The advantage of cloud computing includes location independence, availability, reliability, and optimized cost (Rimal et.al. 2009). To achieve the above benefits, the task needs to be properly scheduled among various resources. There are numerous issues in cloud computing system such as security, load balancing, performance monitoring, resource scheduling, scalability, identity management, optimal resource management, data transfer cost, and energy management. Among these issues, task scheduling is one of the promising issues in cloud computing environment. Load balancing ensures a better QOS by using optimizing the resource utilization and response time (Patel et.al. 2013). The request (task) generates from different users (user bases) and the amount of resource required for executing the tasks changes dynamically. Cloud data centers are highly dynamic to manage the unpredictable load behavior of the users (Randles et.al. 2010), (Kansal et.al. 2012).

Load balancing is achieved by proper task scheduling. Scheduling algorithms are implemented in task scheduler. The task scheduler may be centralized or distributed. The centralized scheduling algorithms in clouds are commonly supported by a common controller that balances VMs to the hosts which is shown in Figure 1. The central management algorithms for load balancing are simpler for implementation. In each execution process of the centralized algorithms, the statuses of all hosts are

Figure 1. Scheduling model
Related Content

Reliability-Aware Proactive Energy Management in Hard Real-Time Systems: A Motivational Case Study
[www.igi-global.com/chapter/reliability-aware-proactive-energy-management/63583?camid=4v1a](www.igi-global.com/chapter/reliability-aware-proactive-energy-management/63583?camid=4v1a)

Support for Dynamic Trading and Runtime Adaptability in Mobile Environments
[www.igi-global.com/chapter/support-dynamic-trading-runtime-adaptability/4222?camid=4v1a](www.igi-global.com/chapter/support-dynamic-trading-runtime-adaptability/4222?camid=4v1a)
Impact of Interest and FX Rate on Breach of DSCR in Project Finance
www.igi-global.com/article/impact-of-interest-and-fx-rate-on-breach-of-dscr-in-project-finance/188804?camid=4v1a

Hybridizing Shuffled Frog Leaping and Shuffled Complex Evolution Algorithms Using Local Search Methods
www.igi-global.com/article/hybridizing-shuffled-frog-leaping-and-shuffled-complex-evolution-algorithms-using-local-search-methods/109613?camid=4v1a