International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Performance and Turning Load Balance in Cloud Computing By Using Modified Particle Swarm Optimization (MPSO)

Rajasekhar Bandapalle Mulinti[1], G. A. Ramachandra[2]

[1]Research Scholar, SK University, Andhra Pradesh, Anantapur, India

[2]Professor, University, Andhra Pradesh, Anantapur, India

Abstract: ***Cloud computing is on demand access to virtualize Information Technology (IT) resources that are hosted in a data center, shared by others, simple to use, paid for via subscription, and accessed over the mobile cloud computing . In recent developments, complexity data intensive applications are increasing most of the users are accessing mobile devices through different range radio communication technologies to make them as cloudlets. So every mobile devices works either services provider or client provider with the corresponding services models. In this research paper we proposed modified particle swarm optimization(MPSO) for optimization of load balance in computing and categories into remote cloud service mode, direct cloud service mode and improves the performances of virtualization and reduce load balances in the cloud hosted services.***

Keywords: Modified Particle Swarm Optimization , offloading, load balance, remote cloud service mode, direct cloud service mode, virtualization

1. Introduction

Cloud computing allows renting infrastructure, runtime environments, and services on a pay- per-use basis. It finds several practical applications and then gives different images of cloud computing to different organizations. In this context, cloud computing an emerging model for represented the data intensive applications and highly design infrastructures and power consuming data centers to supporting the elasticity, scalability and increase the performance of user requirements. The fundamental problem is multidimensional resources such as CPU, storage, networking, etc. with dynamic load balance, energy efficiency. In order to achieve objectives of high performance, energy saving, and reduced costs, so data centers need to handle the physical and virtual resources in dynamic environment. The study of research is to identify best optimization techniques that will facilitate efficient management and scheduling of computing resources in cloud data centers supporting scientific, industrial, business, and consumer applications.

1.1 Workload Classification

Modern data centers provide different services like web applications hosting, Video on Demand (VoD), content sharing, and cloud computing facilities. These services have different computational and communicational requirements. The workloads at a data center may be classified into three major categories:

1.1.1 Balanced Workload

It is process of distributing workloads across multiple computing resources and reduces costs associated with document management systems and maximizes availability of resources. Balanced workloads are generated by applications that have both communicational and computational requirements such as, geographical information systems

1.1.2 Computation Intensive Workload (CIW)

CIW are generated by high performance computing (HPC) applications. Data centers hosting such applications have high demands of computational power (servers), while communicational requirements are minimal. Energy efficiency techniques in such data centers focus on switches as servers must be powered on to meet the computational requirements.

1.1.3 Data Intensive Workload (DIW)

DIW is the one generated by content and VoD applications. YouTube is one of the largest such user-generated content (UGC) application of VoD data. The DIW require high bandwidth and lesser end-to-end delay for efficient data transfer, while the computational requirements are minimal. Dynamic load balance techniques in such data centers focus on servers as the switches have to be powered on to meet the high data transfer requirements.

Various methods are to be used to make a better system by allocating the loads to the nodes in a balancing manner but due to network congestion, bandwidth usage, there were problems are occurred. To solve these problems a load balancing algorithm which is dynamic in nature does not consider the previous state or behavior of the system, it depends on the current behavior of the system. There were various goals that related to the load balancing such as to improve the performance substantially, to maintain the system stability etc. Depending on the current state of the system, load balancing algorithms can be categorized into two types they are static and dynamic algorithms. In the static algorithm there was prior knowledge of the system is needed and not depend on the current system. In the case of dynamic algorithm it is based on the current system and it is better performance than the static algorithm.

2. System Model

In this paper, we explain our assumption regarding customers and the fundamental model of service composition. Then we apply the penalty rule method to our model by formalizing drifty process and the procedure to calculate them following our model.

Today's Cloud data centers have become more flexible, more secure, and provide better support for on-demand allocation with the help of virtualization technology. It provides Cloud data centers have the ability to migrate an application from one set of resources to another in a non-disruptive manner. The main aims to efficiently have and manage extremely large data centers. One of the challenging scheduling problems in Cloud data centers is the allocation and migration of reconfigurable virtual machines (VMs) and the integrated features of physical machine (PM) hosting In load balance scheduling algorithms that consider only one physical server factor—such as CPU, Dynamic Resource Scheduling algorithm (DRSA) considers CPU, memory, and network bandwidth integrated for PMs and VMs.

Our contributions

- Providing a modeling approach to VM scheduling problems of capacity sharing by modifying traditional interval scheduling and considering life cycles and multidimensional characteristics of both VMs and PMs.
- Designing and implementing Modified Particle Swarm Optimization (MPSO) algorithms with computational complexity and competitive analysis.
- Providing performance evaluation of multiple metrics, such as make span, load efficiency, imbalance value, and make span capacity, to adjust make span capacity by simulating different algorithms.

In this model VM allocations as Modified Interval Scheduling Problems (MISPs) with fixed processing times. A set of requests 1,2,…,*n* where the *i*th request corresponds to an interval of time starting at s_i and finishing at f_i is associated with a capacity requirement c_i.

1) All tasks are independent. There are no predetermined constraints other than those implied by the start and finish times.
2) The required capacity of each request is a positive real number between (0, 1] and the capacity of a single PM is normalized to 1.
3) For each VM request is assigned to a single PM, thus interrupting a request and resuming it on another machine is not allowed, unless explicitly stated otherwise.
4) Each PM is always available, that is, each machine is continuously available in [0, ∞).

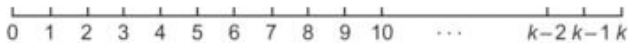


Figure 1: Time slots for N tasks.

3. Problem Description

In cloud architecture having heterogeneous nodes interconnected with high-speed links to perform different computationally intensive applications that have diverse computational requirements. It consists of m independent heterogeneous computing nodes M = {M1, M2, … Mm} nd n number of tasks with each task ti has an expected time to compute tij on node i.e T = {T1, T2, …, Tn}. The entire task has expected time to compute on m nodes. For load balancing problem is to assign each task to one of the node Mj so that the loads placed on all nodes are as "balanced" as possible.

Let A(j) be the set of jobs assigned to node Mj; and Tj be the total time machine Mj have to work to finish the entire task in A(j). Hence Tj = Σtj ∈A(j) tij; for all task in A(j). This is otherwise denoted as Lj and defined as load on node Mj. The basic objective of load balancing is to minimize make span, which is defined as maximum loads on any node (T = maxj:1:m (Tj). Let xij correspond to each pair (i, j) of node Mj Ɛ M and task ti Ɛ T

xij = 0; implies that task i not assign to node j.

$x_{ij} = t_{ij}$; will indicate load of task i on node j.

For each task ti $\sum_{j=1}^{m} x_{ij} = t_{ij}$; for all task ti Ɛ T

The load on node Mj can be represented as

$$L_j = \sum_{i=1}^{n} x_{ij} - t_{ij}$$

where xij = 0 whenever task ti not belongs to A(j). The load balancing problem aims to find an assignment that minimizes the maximum load. Let L be the load of a HDCS with m nodes. Hence the generalized load balancing problem on HDCS can be formulated as

Minimize L

∑mj=1xij = tij; for all task ti Ɛ T

∑ni=1.xij ≤.L, for all Mj Ɛ M

xij {0,tij }, for all ti Ɛ T and Mj Ɛ M

xij = 0, for all ti not belongs to A(j)

4. Algorithm for Modified Particle Swarm Optimization

- Step 1: Initialize random velocities and positions to all particles across m dimensions.
- Step 2: Evaluate the value of fitness function for all particles in all dimension.
- Step 3: Match the p_{best} of every particle with the present value of fitness function and, if better, then replaces p_{best} with new value, and best location with new location x_{new}.
- Step 4: Assign the index of particle with the best success to $g_{best.}$
- Step 5: Update the particles' positions as per equations:

$$V_{idnew} = W * V_{idold} + c_1 * r_1 * \left(P_{best} - X_{old}\right) + c_2 * r_2 * \left(G_{best} - X_{old}\right)$$

- Step 6: Update the particles' Velocity as per equations:

$$X_{new} = X_{old} + V_{new}$$

- Step 7: Repeat from Step 2 to till the stopping criteria is reached or maximum number of iterations are completed.

5. Simulation Results

Cloud Analyst is a tool developed at the University of Melbourne whose goal is to support evaluation of social networks tools according to geographic distribution of users and data centers. It may also help in quickly highlighting any problems with the performance and accuracy of the simulation logic. By performing various simulations, the cloud provider can determine the best way to allocate resources, based on request which data center to be selected and can optimize cost for providing reliable services

5.1 Simulation Parameters

1) **User Base**: User Base models a group of users that is considered as a single unit in the simulation and its main responsibility is to generate traffic for the simulation. A single User Base may represent thousands of users but is configured as a single unit. The traffic generated in simultaneous burst represents the size of the user base.
2) **Internet Cloudlet**: An Internet Cloudlet is a grouping of user requests. The number of requests bundled into a single Internet Cloudlet is configurable in Cloud Analyst. The Internet Cloudlet carries information such as the size of a request execution command, size of input and output files, the originator and target application id used for routing by the Internet and the number of requests.
3) **Data Centre Controller**: The Data Centre Controller is probably the most important entity in the Cloud Analyst. It control the data center management activities such as VM creation, destruction and the routing of user requests received from User Bases via the Internet to the VMs.
4) **Vm Load Balancer**: The Data Centre Controller uses a VmLoadBalancer to determine which VM should be assigned the next Cloudlet for processing.
5) **Cloud Application Service Broker**: The traffic routing between User Bases and Data Center is controlled by a Service Broker that decides which Data Centre should service the requests from each user base.
6) **Service Proximity Based Routing**: In this case the proximity is the quickest path to the data center from a user base based on network latency. The service broker will route user traffic to the closest data center in terms of transmission latency.
7) **Performance Optimized Routing:** Service Broker actively monitors the performance of all data centers and directs traffic to the data center it estimates to give the best response time to the end user at the time it is queried.
8) **Dynamically Reconfiguring Router**: Proximity based routing, where the routing logic is very similar, but the service broker is entrusted with the additional responsibility of scaling the application deployment based on the load it is facing and increasing or decreasing the number of VMs allocated in the data center.

Dynamic task scheduling is performed using Modified Particle Swarm Optimization (MPSO). Each possible solution is represented as the particle. It represents the order in which the tasks are to be executed for a particular processor. Simulation involves maximum 5 processors and 50 jobs. The experimental data has been taken randomly. During the experiment period, the maximum number of iteration was set to 100. The results are compared for varying population size, where the size ranges from 10 to 300.

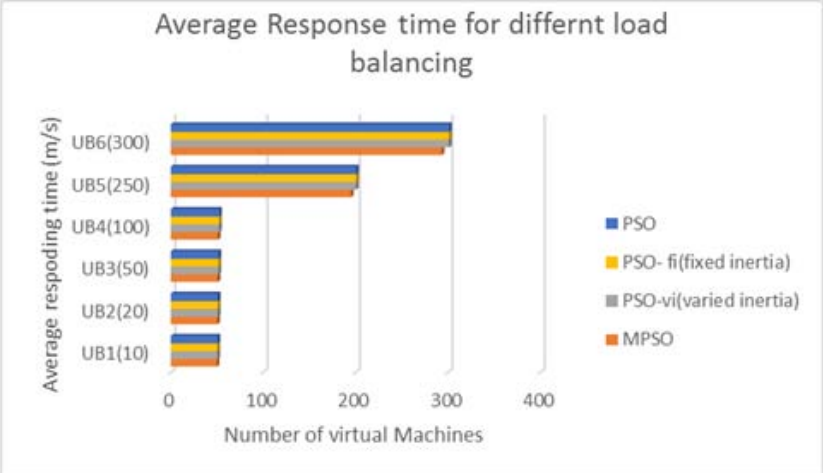
5.2 Results

We used CloudSim and cloud analyst in modeling and simulating Cloud computing environments. To evaluate the overhead in building a simulated Cloud computing environment that consists of a single data center, a broker and a user and performed series of experiments.

The number of hosts in the data center in each experiment was varied from 100 to 1000. As the goal of these tests were to evaluate the computing power requirement to instantiate the Cloud simulation infrastructure and monitoring user workload.

Table 1: Average response time for different UBs in load balancing

Particle Size	*MPSO*	*PSO-vi (varied inertia)*	*PSO- fi (fixed inertia)*	*PSO*
UB1(10)	49.85	50.6	50.62	50.64
UB2(20)	50.42	51.07	51.14	51.15
UB3(50)	50.92	51.67	51.8	51.83
UB4(100)	51.21	52.4	52.48	52.49
UB5(250)	195.25	200.56	200.88	200.9
UB6(300)	293.5	301.38	301.49	301.56



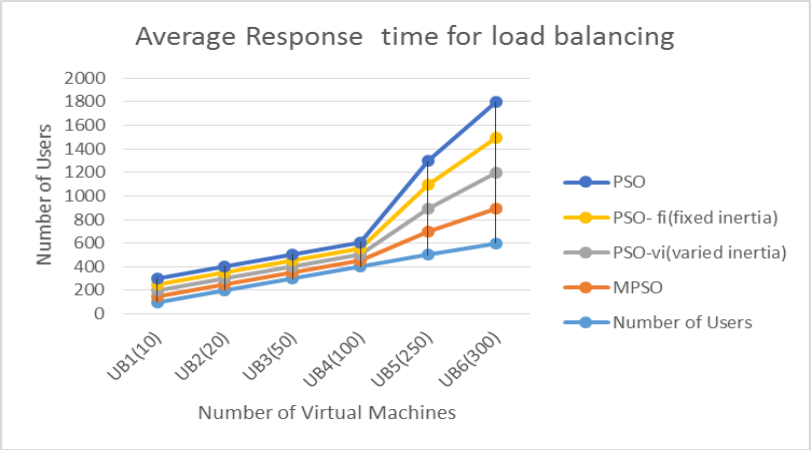


Table 2: Average processing time of DCs in load balancing

Data Centre with No. of VMs	MPSO	PSO-vi	PSO-fi	PSO
DC1(25)	0.689	0.783	0.785	0.786
DC2(50)	1.356	1.558	1.567	1.574
DC3(75)	1.938	2.095	2.1	2.1
DC4(100)	2.385	2.766	2.773	2.773

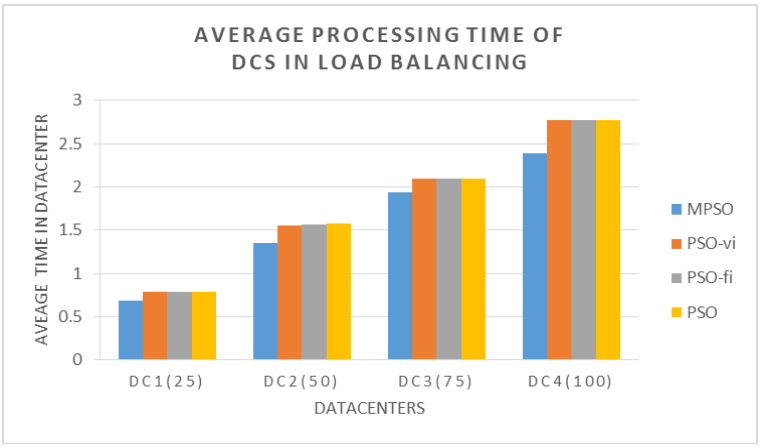
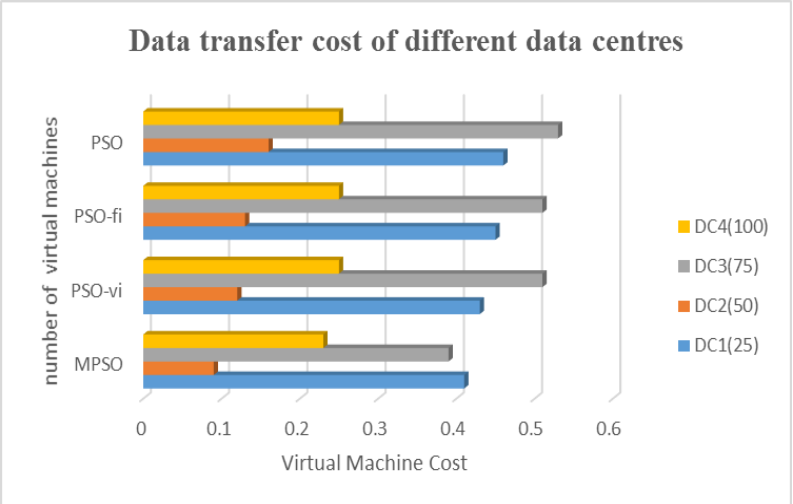


Table 3: Data transfer cost of different data centers in LB

Data Centre with No. of VMs	MPSO	PSO-vi	PSO-fi	PSO
DC1(25)	0.41	0.43	0.45	0.46
DC2(50)	0.09	0.12	0.13	0.16
DC3(75)	0.39	0.51	0.51	0.53
DC4(100)	0.23	0.25	0.25	0.25



6. Conclusion

In this paper we have implemented MPSO algorithm for dynamically schedule the task in a heterogeneous environment. Using this approaches for solving the dynamic task scheduling using PSO has been tried namely PSO with fixed inertia, PSO with variable inertia. The experimental results show that MPSO and its variants perform better than the PSO. In future, work can be carried out by using other hybridization techniques with MPSO to achieve better result.

References

[1] Fan, Z., & Shen, H. (2013). Simulated Annealing Load Balancing for resource allocation in cloud environments. Proceedings of 14th International Conference on Parallel, Distributed Computing, Application and Technology (PDCATB) (pp. 16-18).
[2] Fang, Y., Wang, F., & Ge, J. (2010). A task scheduling algorithm based on load balancing in cloud computing. In Web Information Systems and Mining. Lecture Notes in Computer Science (Vol. 6318, pp. 271–277). doi:10.1007/978-3-642-16515-3_34
[3] Ge, Y., & Wei, G. (2010). GA-Based Task Scheduler for the Cloud Computing Systems. Proceedings of The IEEE International Conference on Web Information Systems and Mining, (pp. 181-186). doi:10.1109/WISM.2010.87
[4] Goldberg, D. (1989). Genetic Algorithms in search, optimization, and machine learning. Addison-Wesley Publishing Corporation, Inc.
[5] Guo, L., Zhao, S., Shen, S., & Jiang, C. (2012). Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm. Journal Of Networks, 7(3), 547–553. doi:10.4304/jnw.7.3.547-553
[6] Guo-ning, G., Ting-Iei, H., & Shuai, G. (2010). Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment. Proceedings of IEEE International Conference on Intelligent Computing and Integrated Systems (pp. 60-63).
[7] Hu, J., Gu, J., Sun, G., & Zhao, T. (2010). A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. Proceedings of Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP) (pp. 89-96).
[8] Jang, S., Kim, T., Kim, J., & Lee, J. (2012). The Study of Genetic Algorithm-based Task Scheduling for Cloud Computing. International Journal of Control and Automation, 5(4), 157–162.
[9] Junwei, G., & Yongsheng, Y. (2013). Research of cloud computing task scheduling algorithm based on improved genetic algorithm. Proceedings of 2nd International Conference on Computer Science and Electronics Engineering (pp. 2134-2137).
[10] Kennedy, J., & Eberhart, R. (2001). Swarm intelligence. San Francisco, CA: Morgan Kaufmann Publishers, Inc.
[11] Kumar, R., & Sahoo, D. (2013). Load balancing using ant colony in cloud computing. International Journal of Information Technology Convergence and Services, 3(5), 1–5. doi:10.5121/ijitcs.2013.3501
[12] Li, K., Xu, G., Zhao, G., Dong, Y., & Wang, D. (2011). Cloud Task scheduling based on Load Balancing AntColonyOptimization. Proceedings of Sixth IEEE Annual ChinaGrid Conference (pp. 3-9).
[13] Liu, H., Liu, S., Meng, X., Yang, C., & Zhang, X. (2010). LBVS: A load balancing strategy for virtual storage. In the Proceedings of International Conference on Service Sciences (ICSS) (pp. 257-262). IEEE.
[14] Liu, J., Luo, X., Zhang, X., Zhang, F., & Li, B. (2013). Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm. IJCSI International Journal of Computer Science Issues, 10(3), 134–139.
[15] Liu, X., Pan, L., Wang, C., & Xie, J. (2011). A lock-free solution for load balancing in multi-core environment. Proceedings of 3rd IEEE International Workshop on Intelligent Systems and Applications (ISA) (pp. 1-4).