Intelligent water drop algorithm for load balancing in cloud computing

Shibina T.
Department of computer science and engineering
Malabar Institute of Technology,
APJ abdulkalam university.
shibinaramith@gmail.com

Lijina S.S
Department of computer science and engineering
Malabar Institute of Technology
APJ abdulkalam university
lijina.nambiar@gmail.com

Abstract—As cloud users are increasing day by day and has become one of the important challenges for the cloud providers in terms of load balancing. In the previous works, many have focused on dynamic load balancing algorithms namely modified throttled,FCFS and Particle Swarm Optimization based on their performance using Cloud Analyst Simulator. Simulation outcomes are recorded in terms of the Response time and data center processing time of the three algorithms along with its performance and arrival cost details.

In my work I would like to propose a new algorithm “intelligent water drop” for load balancing in cloud.A natural river often finds good paths among lots of possible paths in its ways from the source to destination. These near optimal paths are obtained by the actions and reactions that occur among the water drops and the water drops with the riverbeds. The intelligent water drops (IWD) algorithm is a new swarm-based optimization algorithm inspired from observing natural water drops that flow in rivers.

I. INTRODUCTION

Cloud computing popularly termed as the computing system which offers internet based services on demand in parallel and distributed environment. Requests from different users are distributed to different processors randomly which imbalances the load assignment and this is considered as one of the biggest disadvantage of cloud computing. Thus, loads needs to be managed and it can be done in two different methods i.e. load balancing and load sharing. Load balancing is the strategy of dividing the workload between many computers or data centers equally in order to enhance the performance and makes the process faster. Many algorithms are proposed through which load can be distributed equally and with minimum Response time. Load Balancer is use to balance the load .Whereas Load sharing can be directed as the distribution of loads to different computers or data centers but doesn’t ensure uniform balance. Load Balancing concept can be further classified into two category i.e. static load balancing algorithm and dynamic load balancing algorithm. Static Load Balancing algorithm is an algorithm that checks the current state of the node algorithm and distributes the requests on a fixed set of rules depending on the input requests. The common static load balancing algorithms are Round-Robin, Weighted RR, and Shortest Job First Scheduling Algorithm. Another type is the dynamic Load Balancing algorithm which is called as self-adaptive algorithm. This algorithm checks the previous state as well as the current node and adjusts traffic distribution evenly in real time. Throttled Algorithm, Modified Throttled Load Balancing Algorithm, FCFS Algorithm, Particle Swam Optimization Algorithm, Genetic Algorithm, Clustering Algorithm, etc are some of the common dynamic algorithms. Cloud computing scenario cannot rely on the prior knowledge of nodes capacity, processing power, memory, performances and users requirements,so dynamic load balancing algorithm fits better than static load balancing algorithm as the former one takes run time statistics for load balancing.And another greater advantage in dynamic environment lies in the flexibility of user requirements, which may change at run time.

“Intelligent water drop(IWD) algorithm” for load balancing in cloud works on principle that a natural river often finds good paths among lots of possible paths in its ways from the source to destination. These near optimal or optimal paths are obtained by the actions and reactions that occur among the water drops and the water drops with the riverbeds.

II. RELATED WORK

In accordance with the two environments i.e. static and dynamic, many algorithms [1] have been analyzed. In that they proved PSO gives good response time with low implementation cost. In this section I would like to propose a new algorithm IWD [2] proposed by Hamed Shah-Hosseini.

III. LITERATURE SURVEY

Many dynamic algorithms have opposed for load balancing,some of them are,

A. *Throttled Load Balancing Algorithm*

Throttled Load Balancing Algorithm [1] is a dynamic algorithm which completely deploy on virtual machines. In this allocation, the client appeals the throttled load balancer to find the suitableVMto perform the operation. The VMs are grouped according to the requests they can manage. The moment the client sends the request, the load balancer immediately gets alert and searches for the required group which can manage easily. The issue of this allocation is that the load balancer has to search for the suitable VM, which would lead to delay in operation. Modified Throttled Load Balancing Algorithm [1] is algorithm is bit modified version of Throttled Load Balancing Algorithm. Modified Throttled Load Balancing Algorithm maintains a set of virtual machines named as VMindex table and stating the position of the VMs (i.e. Busy/Available).VM at first index is initially selected depending upon the state. If VM is available, then the request is assigned and if VM is not found then it returns (- 1) to the Data Centre Controller. When the next request arrives, the VM next to the already allocated VM is choosed .This process is repeated continuously until the index table size is reached, which is depicted in Figure below.

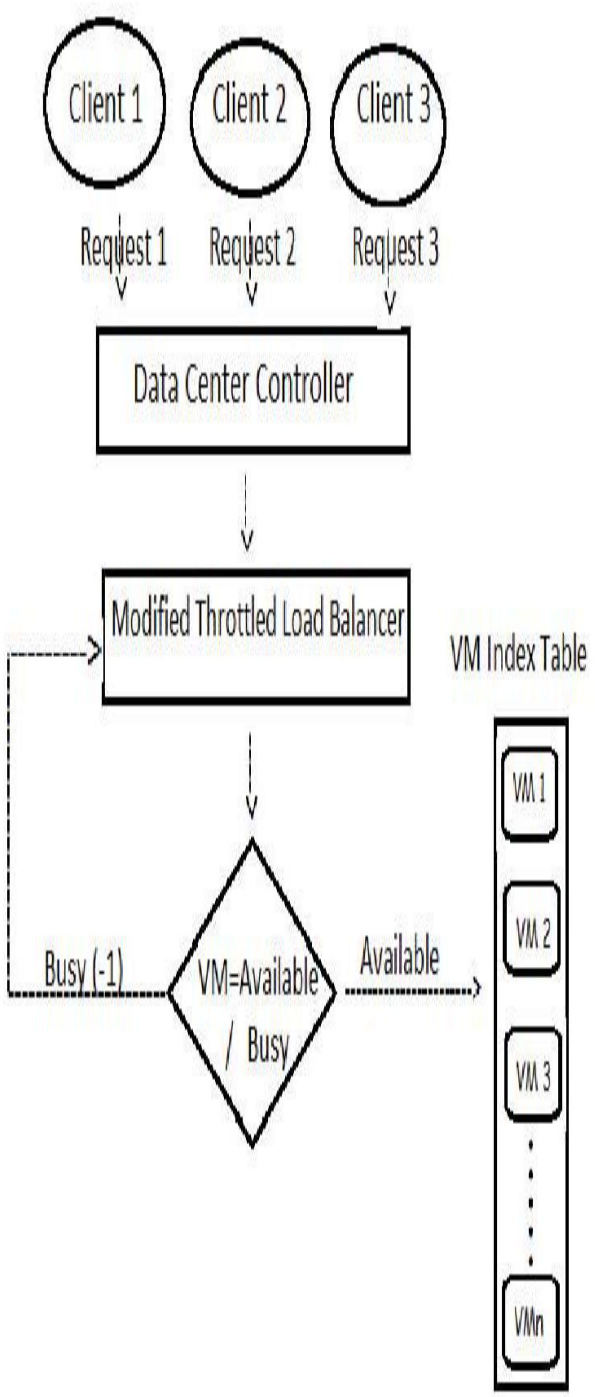


Fig.Modified Throttled Algorithm

A. FCFS Algorithm

FCFS Algorithm [1]is the simplest parallel task ordering dynamic load balancing algorithm. The processing takes place by choosing the right order of jobs. With this scheme, the user request which comes first to the data centre controller ,would only be allocated with the VM for first execution. The implementation of FCFS policy[2] is easily managed with FIFO queue. The datacenter controller searches for virtual machine which is in free or overloaded. Then the 1strequest from the queue is removed and is passed to one of the VM through the VMLoadBalancer.The allocation of request takes place by two ways : Firstly the requests can be arranged in a queue manner and secondly by allocating heavy load node less work and low load node more work manner. Many function parameters can be taken into consideration in order to calculate the current real load weighing value and the complex load weighing value. The whole mechanism of the algorithm is depicted in the below Fig.

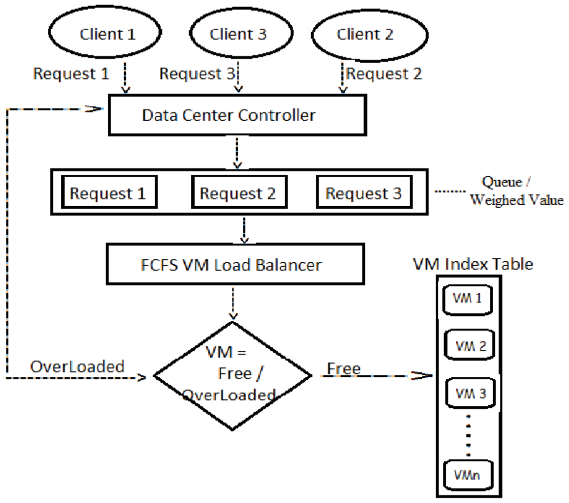
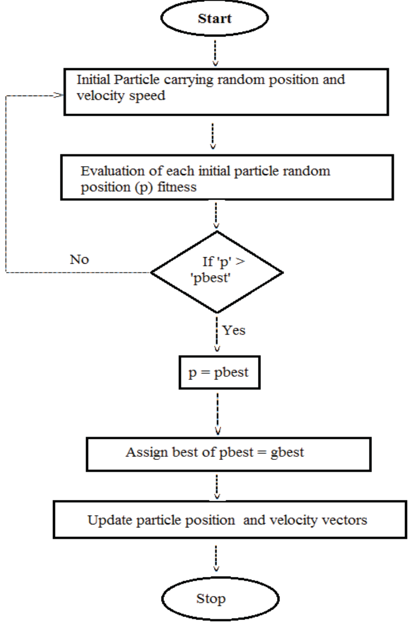


Fig. 2.FCFS algorithm

B. Particle swarm optimization algorithm

Particle Swarm Optimization Algorithm [1] Particle Swam Optimization is a heuristic speculative escalation technique based on swarm intelligence. Particle Swam Optimization bids on the idea of social synergy of birds and fish flock behaviour. Particle Swam Optimization idea was put forward by Dr.Kennedy and Eberhant in 1995. Particle swarm is mainly of 'n' particles and the position of each particle is spaced in N-dimensional region which keeps track of its coordinates in accordance to the fitness as far as achieved (analysis of particle swarm algorithm in different areas). Particle Swarm Optimization Algorithm main idea lies in modification of each particle in position. This modification mainly depends on the


Start
Initial Particle carrying random position and velocity speed
Evaluation of each initial particle random position (p) fitness
No
If 'p' > 'pbest'
Yes
p = pbest
Assign best of pbest = gbest
Update particle position and velocity vectors
Stop

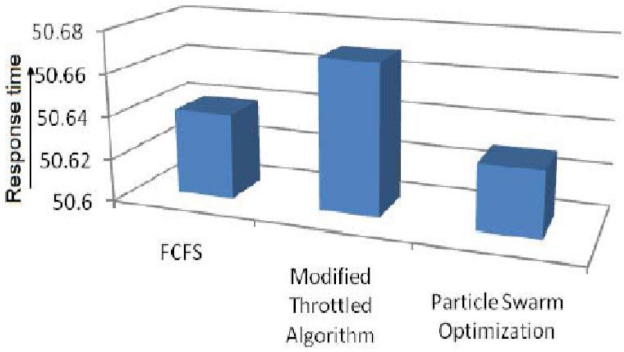
current position, current velocity vectors,(dist:current position->pbest) and ist:current position—-> gbest) where pbest - best value procured so far by any particle in its own coordinate solution space, gbest - best value procured so far by any particle in the community of the particle concern. The flow of the algorithm is depicted in Fig. 3. below.

Fig. 2.3: Particle Swarm Optimization

Here initialisation of each particle carrying random position and velocity speed takes place. After each particle gets intialised, evaluation of particle is done resulting a fitness value(p). If the fitness(p) is greater than fitness(pbest) than p = pbest and the best value of pbest is assign as gbest with updating the particle position and velocity vectors. In the other way, if it doesnt agree with the condition then again the same process of initialization starts.

IV. ANALYSIS OF EXISTING ALGORITHMS

comparison of three major dynamic load balancing algorithms namely, Modified Throttled Load Balancing Algorithm, FCFS Algorithm and Particle Swarm Optimization Algorithm it’s found that the Response time of Particle Swarm Optimization Algorithm is efficient one as compared to other two algorithms.



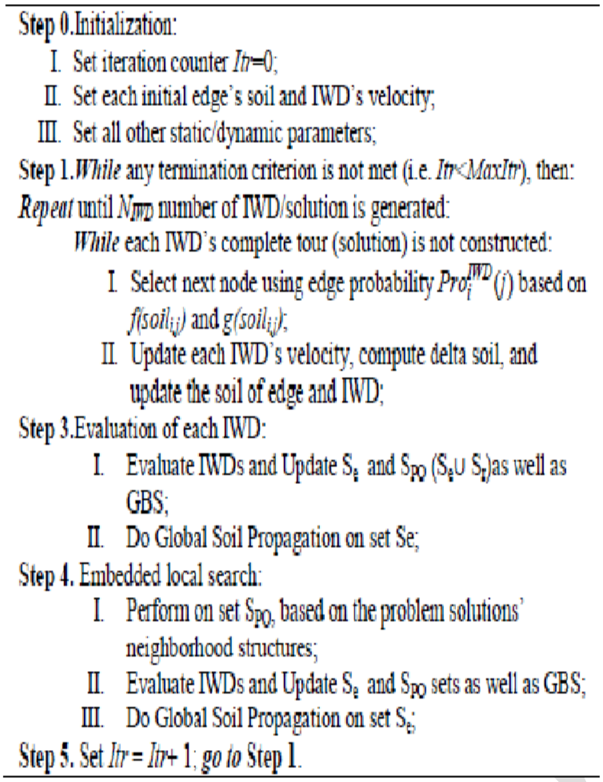
V. PROPOSED SYSTEM

I would like to propose IWD algorithm for load balancing in cloud computing environment.

In IWD algorithm [2] and [3] uses property of Water drops that flow in rivers, lakes, and seas are the sources of inspiration for developing the IWD. This intelligence is more obvious in rivers which find their ways to lakes, seas, or oceans despite many different kinds of obstacles on their ways. In the water drops of a river, the gravitational force of the earth provides the tendency for flowing toward the destination. If there were no obstacles or barriers, the water drops would follow a straight path toward the destination, which is the shortest path from the source to the destination. However, due to different kinds of obstacles in their way to the destination, which constrain the path construction, the real path has to be different from the ideal path and lots of twists and turns in the river path is observed. The interesting point is that this constructed path seems to be optimum in terms of distance from the destination and the constraints of the environment.

1. IWD ALGORITHM

Basic principles of the IWD algorithm Water drops that flow in rivers, lakes, and seas are the sources of inspiration for developing the IWD. This intelligence is more obvious in rivers which find their ways to lakes, seas, or oceans despite many different kinds of obstacles on their ways. In the water drops of a river, the gravitational force of the earth provides the tendency for flowing toward the destination. If there were no obstacles or barriers,the water drops would follow a straight path toward the destination, which is the shortest path from the source to the destination. However, due to different kinds of obstacles in their way to the destination, which constrain the path construction, the real path has to be different from the ideal path and lots of twists and turns in the river path is observed. The interesting point is that this constructed path seems to be optimum in terms of distance from the destination and the constraints of the environment. IWD could give better response time than PSO or any other Genetic algorithms.



```
Step 0.Initialization:
    I.   Set iteration counter $Itr$=0;
    II.  Set each initial edge's soil and IWD's velocity;
    III. Set all other static/dynamic parameters;
Step 1.While any termination criterion is not met (i.e. $Itr<MaxItr$), then:
Repeat until $N_{IWD}$ number of IWD/solution is generated:
        While each IWD's complete tour (solution) is not constructed:
            I.  Select next node using edge probability $Pro_i^{IWD}(j)$ based on
                $f(soil_{i,j})$ and $g(soil_{i,j})$;
            II. Update each IWD's velocity, compute delta soil, and
                update the soil of edge and IWD;
Step 3.Evaluation of each IWD:
        I.   Evaluate IWDs and Update $S_E$ and $S_{PO}$ ($S_E$∪ $S_T$)as well as
             GBS;
        II.  Do Global Soil Propagation on set Se;
Step 4. Embedded local search:
        I.   Perform on set $S_{PO}$, based on the problem solutions'
             neighborhood structures;
        II.  Evaluate IWDs and Update $S_E$ and $S_{PO}$ sets as well as GBS;
        III. Do Global Soil Propagation on set $S_E$;
Step 5. Set $Itr$ = $Itr$+ 1; go to Step 1.
```

1.

VI. CONCLUSION

The greatest challenge in cloud computing world is minimization of Response time and cost in order to balance the load and increase business performance with customer satisfaction. comparison of three major dynamic load balancing algorithms namely, Modified Throttled Load Balancing Algorithm, FCFS Algorithm and Particle Swarm Optimization Algorithm it's found that the Response time of Particle Swarm Optimization Algorithm is efficient one as compared to other two algorithms.But IWD is best algorithm ever proposed in swarm intelligence, hence it could give better result than PSO algorithm for load balancing in cloud computing.

References

[1] Prasant Kumar Pattnaik Priyadarashini Adyasha Pattanaik,Sharmistha Roy, Performance study of some dynamic load balancing algorithms in cloud computing environment, 2nd International conference on Signal Processing and Integrated Networks (SPIN) (2015).

[2] Hamed Shah-Hosseini, Intelligent water drops algorithm a new optimization method for solving the multiple knapsack problem, International Journal of Intelligent Computing and Cybernetics **1** (2008), no. 2, 193–212.

[3] Ehsan Teymourian S. Sheikh Zahra Booyavi, G.M. Komaki, An improved optimization method based on the intelligent water drops algorithm for the vehicle routing problem, (2014).