A SOLUTION OF LOADING BALANCE IN CLOUD COMPUTING USING OPTIMIZATION OF BAT SWARM ALGORITHM

LAHEEB MOHAMMED IBRAHIM1,*, IBRAHIM AHMED SALEH2

1Software department, computer and Math. College, University of Mosul
Almalewa quarter, Mosul, Nineveh, Iraq
2Software department, computer and Math. College, University of Mosul
A-baker quarter, Mosul, Nineveh, Iraq
*Corresponding Author: Laheeb_alzubaidy321966@uomosul.edu.iq

Abstract

Loading balance is one of the most important problems in computer cloud environment because it needs to optimize task scheduling. The environment of cloud computing is not sufficient to give good quality for user’s service because the tasks schedule and allocation are not of the same loads. In this paper a new strategy is proposed to balance the loads using swarm Bat Algorithm (BA). Bat algorithm behaviour uses echolocation to detect their prey for creating load balancing model. The proposed strategy depends on Naïve-Bayes algorithm to classify virtual machines (VMs) that is used to update the migrated task for the information used by the bats. The migrated tasks assigned heavy loaded VMs to light loaded VM providing numbers of tasks have lowest priority in task’s queue. The tasks from heavy-loaded migrated virtual machines are migrating to lighted-loaded virtual machine, it is found that tasks have high-priority from minimizing of services in VM and avoid tasks have large number of services are scheduled to the same VM. The experimental results on loading balance bat algorithm (LBBA) were compared with traditional algorithms such as Dynamic load balance (DLB)or round robin (RR). It is clear that the LBBA is performing well better in regard to less response, completion, and migration times than the traditional methods. Further, the requirements of user’s service quality are best matched.

Keywords: Bat algorithm, Cloud computing, Load balancing, Naive Bayes, Quality of service QoS, Virtual machine.
1. Introduction

Cloud computing is a cluster of distributed computing resources (hardware and software) used as a service over a network. It is a model for enabling convenient on-demand network access to a shared pool of configurable computing resources or internet-based computing model. It provides data with lower cost, parallel computing, distributed computing and dynamic resources request. Virtual machines (VM's) are used as processing unit in a cloud environment that executes in parallel to reduce the execution time of task. Therefore, a reasonable appointment scheduling can improve user Quality of Service (QoS) and resources utilization rate [1]. To perform load balancing, the cloud computing task allocation is divided into three levels as shown in Fig. 1, the task request layer, resource management layer and task execution layer [2].

- **Task request layer**: Task request layer is also called user-oriented layer, in this layer the user interacts with the system to obtain their requirements.
- **Resource management layer**: Resource management layer uses map to reduce divide the task into logically independent subtasks and loading tasks. The equalization mechanism allocates subtasks to virtual resource nodes in a suitable physical machine perform parallel processing.
- **Task execution layer**: In this layer the execution time of subtasks should be as small as possible.

![Cloud computing task allocation levels](image)

**Fig. 1. Cloud computing task allocation levels [2].**

Resource management is the realization of cloud computing resources, which considers as the key to rational scheduling. It uses virtualization technology to put physics under cloud computing environment. The machine is splitting into a number of virtual resources, and these virtual resources are allocated on demand and automatic growth [3]. Load balancing is an important factor to measure task scheduling. Tasks are assigned in one or more virtual machines and run concurrently. The scheduler needs assistance adjust the task allocation to avoid some virtual machines overloaded, while others will be for long-term in the idle state. The load of resources with the implementation of the mandate constantly changes. The main goal of load balancing is to reduce task completion time, reduce the request ring should time and improve user service quality [4]. The two technologies for load balancing in cloud computing are static and dynamic technologies as follows:

- **Static load balancing technology**: Used for changes little situation from the task phase and node load.
• **Dynamic load balancing technology**: This technology is based on the system which determines how to assign tasks virtual machines to the status of current load. It depends on the present condition for the system and avoids previous state or behaviour of the system [5].

In this paper the simulation model used Bat Algorithm (BA) to balance the loads in cloud computing. BA is used to improve strategy of collecting mechanisms to the dynamic load balancing (DLB). According to bat behaviour mechanism to establishment new method to balance the loads for cloud computing. The load state of the virtual machine will be divided into three groups considering, task priority, resources and rational allocation of tasks. This method represents the cloud environment. In real situation, the traditional dynamic load balancing algorithm (DLB) and the static weighted wheel compared with the WRR (Weighted Round Robin) scheduling algorithm. WRR algorithm reduces response time, completion time, and improves user service quality. The second part of the paper illustrates an overview about related works, while the third part describes the loading balance concept. The fourth part proposes the bat algorithm for dynamic load balancing. Finally, the last and fifth part is devoted to the experiments results and the concluding remarks.

2. Related Work

Many researchers have a lot of related papers on load balancing technology in cloud computing, as shown in Table 1.

<table>
<thead>
<tr>
<th>No.</th>
<th>Authors</th>
<th>Year</th>
<th>Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ruixia and</td>
<td>2010</td>
<td>Researchers proposed a residual load rate based the combination of static and</td>
</tr>
<tr>
<td></td>
<td>Xiongfeng [6]</td>
<td></td>
<td>dynamic balance strategy after analysing static balance technology and dynamic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>balance technology to improve the performance of cluster systems. the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>researchers balanced in static load balancing to reduces times of dispatching</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>assignments between nodes, prevents the servers to be skew, and improves the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>performance of cluster systems</td>
</tr>
<tr>
<td>2</td>
<td>Kun et al. [7]</td>
<td>2011</td>
<td>The researchers proposed a cloud task scheduling is an NP-hard optimization</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>problem, and many meta-heuristic algorithms to solve it. They used ant colony</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>optimization (LBACO) algorithm in a cloud task scheduling policy. The main</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>contribution to balance the entire system load while trying to minimize the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>make span of a given tasks set</td>
</tr>
<tr>
<td>3</td>
<td>Rajoriya [8]</td>
<td>2012</td>
<td>This research explain in detail, load balancing technique in cloud computing,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>and also explain the different qualitative</td>
</tr>
</tbody>
</table>
metrics or parameters like performance, scalability, associated overhead, etc.

<table>
<thead>
<tr>
<th>No.</th>
<th>Authors</th>
<th>Year</th>
<th>Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Kumar et al. [9]</td>
<td>2012</td>
<td>The researchers applied ant colony optimization (ACO) algorithm for load distribution of workloads among nodes of a cloud computing. In modified algorithm the edge over the original approach in which each ant build own individual result set, However, ants continuously update a single result set rather than updating their own result set.</td>
</tr>
<tr>
<td>5</td>
<td>Redjeki et al. [10]</td>
<td>2014</td>
<td>This paper discuss how Cloud computing used to share data and provide many resources to users, how Cloud computing stores data in open environment where amount of data storage increases quickly in open environment. They used various types load in cloud computing like memory, CPU and network load. Load balancing is the process of finding overloaded nodes and then transferring the extra load to other nodes.</td>
</tr>
<tr>
<td>6</td>
<td>Akshada et al. [11]</td>
<td>2015</td>
<td>This paper includes a better load balance model for the cloud based on the cloud partitioning concept. A switch mechanism is chosen for different strategies and for different situations. The round robin and game theory algorithm improves the efficiency and performance in cloud environment LESS.</td>
</tr>
<tr>
<td>7</td>
<td>Athokpam et al. [12]</td>
<td>2017</td>
<td>This paper Reviews an overview on load balancing schemes in cloud environments. The researcher explore the diverse types of algorithms that are proposed by a number of researchers to solve the problem of load balancing in cloud computing.</td>
</tr>
<tr>
<td>8</td>
<td>Singh and Vashisht [13]</td>
<td>2018</td>
<td>Authors had proposed new version for “improved regulated algorithm (IRA)” to regulate load balancing approach for cloud computing and compared with equally spread, throttled and round robin approach. Better performance in terms of cost, response time and data processing time.</td>
</tr>
<tr>
<td>9</td>
<td>Manaseer et al. [14]</td>
<td>2019</td>
<td>The researchers applied MEMA technique with little steps adding to weighted round robin (WRR) and compares running time with WRR and sure give better result.</td>
</tr>
</tbody>
</table>
3. Load Balancing Mode

Load balancing is used to allocate processing with heavier load to smaller processing nodes for enhancing the performance of the system. In cloud computing environment load balancing is required to distribute the dynamic local workload evenly between all the nodes [15]. The overloaded (heavy load) occurs in a virtual machine. In this state some tasks must be migrated from virtual machine which has heavy load to virtual machine that has light load. When multiple tasks are migrated, there are multiple virtual machines can receive the available tasks. The selected and appropriate virtual machine priority based on the mechanism of bat algorithm to establish an equilibrium model in bats have the following:

- Bats search for food source by locating echoes in a sonar model used by bats to detect prey and avoid obstacles. These bats sound very loud and listen to the echoes from surrounding objects. Thus, the bat can calculate how far away it is from an object. In addition, bats can distinguish between obstacle and prey even in total darkness. All bats use echolocation to sense distance [16].
- The bats recognize the difference between food/prey and background barriers in some magical way. With this echo a bat can decide an object’s dimensions, shape, direction, duration, and movement. Bat behaviour control structure is shown in Fig. 2.

1. Initialize the population $x_i$ and velocity $v_i$, max number of iterations.
2. Define pulse frequency $f_i$ and $x_i$ and initialize pulse rare $n$ and loudness $A_i$
3. While ($t < \text{max number of iteration}$)
   a. Generate new solution by adjusting frequency, and updating velocities and location by using equation
   $$f_i = f_{min} + (f_{max} - f_{min}) * \beta$$
   $$v_i^{t+1} = v_i^t + (v_i^t - x_c) * f_i$$
   $$x_i^{t+1} = x_i^t + v_i^{t+1}$$
   where $\beta \in [0 \ 1]$ is random vector drawn from a uniform distribution, here $x_c$ is current global best solution. The value of frequency $f_{min} & f_{max}$ depended on domain size of the problem of interest.
   b. If (rand>n)  
   Select solution among the best solutions and generate location solution around it with equation
   $$x_{new} = x_{old} + \epsilon + A_{mean}$$
   c. Generate new solution.
   d. If (rand<$A_i$ and $f(x_c)<f(gbest)$)  
   Accept new solutions, increase $n$ and reduce $A_i$
   End if
   e. Rank the bat and find the current gbest.
   End While

Fig. 2. Bat optimization algorithm [16].
The model for corresponding between load balancing task scheduling and behaviour of bat algorithm under below:

i. Mission: bats

ii. Mission to be relocated: research on the social behavior of bats.

iii. Virtual machine: food (prey) resource.

iv. Task loading into virtual machine: bats use audio echo-location in detecting prey and finding himself located in the habitat of cracks in the dark.

v. Virtual machine overloaded: prey is not rich.

vi. Task scheduling to the appropriate virtual machine: bats look for new food sources.

vii. Migration task to update the virtual machine information: reconnaissance bats swing frequently by using echolocation of fixed signals to load balancing strategy for simulating bat behavior.

For loading balance model and to schedule the tasks, the following steps should be conducted:

• In a load balancing model, tasks are loaded into the virtual machine.

• If the current virtual machine loads is light, then the task is handled by this virtual machine.

• If the virtual machine loads heavier and there are multiple light-load virtual machines, then need to find the right virtual machine for this task,

• Match the most appropriate virtual machine to other tasks based on task priority which is not appropriate for this received task.

• The virtual machine waits and listens for other virtual machines updated by information for their tasks after the task is scheduled to the target virtual machine.

• Update the workload of the virtual machine and the number of different priority tasks to help waiting for the task in the queue to choose a lighter load and the highest number of high priority tasks virtual machine.

• When new tasks arrive, loop this around process until all tasks are assigned to the virtual machine and the system reaches load balancing status.

4. Load Balancing Bat Algorithm (LBBA)

Cloud computing is based on requests of user to assign tasks to virtual machine. Set of Dynamic resource are centralized and require different services requests from clients based on management policies which are sent to the terminal server. The common scheduling policies of the system are First-In-First-Out (FIFO) and Weighted Round Robin (WRR). However; these strategies eventually result in an unbalanced load on the virtual machine when they consume long task responds, waste of system resources and other issues [17].

Load balancing is designed to reduce response time for tasks and to make short span to it. Task scheduling is scheduling $n$ tasks to $m$ virtual machines. This paper assumes that the task set: $T = \{T_1, T_2, ..., T_n\}$ virtual machine set: $VM = \{VM_1, VM_2, ..., VM_m\}$, so the completion time is defined as:

$$
Timespan = \max\{CT_{ij}|i = 1,2, ... n; j = 1,2, ... m\}
$$

(1)
\( CT_{ij} \) represents the completion time of task \( T_i \) on the \( VM_j \)

### 4.1. Load status classification

During system operation, \( VMs \) are classified according to the load of the \( VMs \) three sets (\( L_{set} \) for light load, \( B_{set} \) for balanced load and \( H_{set} \) for heavy load) [15].

This paper applies Bayes algorithm in statistics to classification the virtual machine state. The load of virtual machine factors includes memory, CPU, network and hard disk [18-19].

Assume that the load attribute value \( L = (L_1, L_2, L_3, L_4) \) each value of the virtual machine \( (v) \) are expressed as:

i. **Memory attribute value**, \( L^v_1 = \alpha_1 M_1 + \alpha_2 M_2 \) where \( M_1 \) is assigned to the virtual machine \( v \) of the physical memory, \( M_2 \) for memory utilization, weight coefficient \( \alpha_1 + \alpha_2 = 1 \).

ii. **CPU attribute value**, \( L^v_2 = \rho_1 C_1 + \rho_2 C_2 \), where \( C_1 \) is the CPU of virtual machine \( v \) of the processing power, \( C_2 \) said CPU utilization, weight coefficient \( \rho_1 + \rho_2 = 1 \).

iii. **Network property value**, \( L^v_3 = \beta_1 N_1 + \beta_2 N_2 \), where \( N_1 \) is the network bandwidth of virtual machine \( v \) wide, \( N_2 \) is network delay, weight coefficient \( \beta_1 + \beta_2 = 1 \).

iv. **Disk property value**, \( L^v_4 = \omega_1 D_1 + \omega_2 D_2 \) where \( D_1 \) represents the virtual machine \( v \) disk access speed, \( D_2 \) for disk utilization, weighting factor \( \omega_1 + \omega_2 = 1 \).

Another time assume that the sample classification set \( K = \{K_i | i = 1, 2, 3\} \), \( K_1 \) indicates light load status, \( K_2 \) represents the state of equilibrium load, \( K_3 \) represents the state of heavy load, \( K(v) \) represents the virtual machine \( v \) load status.

According to the Bayesian theorem when the above attributes are independent of each other, classification result appears more accurate. For this reason, the value of each attribute is independent of each other.

Suppose that the total number of known samples is \( S \), The number of samples belonging to the category \( K_i \) in the sample is \( s_i \), the prior probability of the sample classification \( P(K_i) = \frac{s_i}{S} \) for the unknown sample \( u_x \), \( K(u_x) = K_i \), the conditional probability of is \( P(K_i | u_x) \), available from the Bayesian formula, the posterior probability of \( K_i \) is [16]:

\[
P(K_i | u_x) = \frac{P(K_i | u_x)}{P(u_x)} \quad (2)
\]

To set unknown sample \( u_x \) of \( F(u_x) = (F_{u_1}, F_{u_2}, F_{u_3}, F_{u_4}) \) because of the hypothesis in the text \( F_{u_j} (j = 1,2,3,4) \) independent of each other, therefore:

\[
P_r(u_x | G_i) = P_r \left( (L^u_{1x}, L^u_{2x}, L^u_{3x}, L^u_{4x} | G_i) \right) = \prod_{j=1}^{4} P_r \left( L^u_{jx} | K_i \right) \quad (3)
\]

where \( P_r(L^u_{jx} | K_i) (j = 1,2,3,4; i = 1,2,3) \) can be obtained by training the data set \( G_i \). The posterior probability can be expressed as
A Solution of Loading Balance in Cloud Computing using Optimization

\[ P_{r}(K_i|u_x) = \frac{P_r(K_i) \cdot \prod_{j=1}^{n} P_r(L_j^n|K_i)}{P_r(u_x)} \] (4)

The denominator \( P_r(u_x) \) does not depend on the classification \( K_i \), which is constant when all load attribute values are known, \( P_r(u_x) \) is constant \( Z \), then

\[ P_{r}(K_i|u_x) = \frac{1}{Z} P_r(K_i) \] (5)

According to the maximum a posteriori probability (MAP) decision criterion of naive Bayesian classifier, the definitions points are defined Class functions are:

\[ classify(u_x) = \arg \max \left\{ \frac{1}{Z} P_r(K_i) \cdot P_r(L_j^n|K_i) \right\} \] (6)

Which \( i = 1, 2, 3 \) so available

\[ classify(u_x) = \begin{cases} K_1 & \text{then } u_x \in L_{set} \\ K_2 & \text{then } u_x \in B_{set} \\ K_3 & \text{then } u_x \in H_{set} \end{cases} \] (7)

In order to load balance in the whole system, the part of the virtual machine in the \( H_{set} \) collection needs to be set tasks are migrated to the \( L_{set} \) collection [18]. The migrated tasks are based on load of virtual machines in \( L_{set} \). The task number and priority are selected target virtual machine. In the methods described here, the mission is to be reconstructed as bat echoes, the target virtual machine as the target prey source. The task information is updated after the migration to the target virtual machine includes (the load of all the virtual machines, the priority and number of tasks in the virtual machine, the number of virtual machines in each collection, and so on). The \( VM_i \) in \( B_{set} \) are not involved in the task migration, and the \( VMs \) in \( L_{set} \) and \( H_{set} \) are completed. After the task has been migrated, the load is balanced and partitioned into the \( B_{set} \), with all the virtual when the machines are in \( B_{set} \) collection; the entire system is loading balanced [19].

4.2. Evaluation of load balancing model

This paper assumes that all (virtual machines and tasks) are independent, parallel, and tasks are non-pre-emptive. The completion time (\( CT_i \)) for task \( T_i \) on the \( VMj \) processing time is défining processing time (\( PT_j \)) for all tasks on virtual machine (\( VM_j \)):

\[ PT_j = \sum_{i}^n PT_{ij}, j = 1, 2, ..., n \] (8)

The dynamic load balancing algorithm LBBA proposed not only balances the resource load, but also balances the number of different priority tasks in the virtual machine waiting queue. Tasks are scheduled from heavily loaded virtual machines to lightly loaded virtual machines when the machine is used, the virtual machine with the least load and the highest priority than the task is selected according to the existing information, so that the task can be executed as early as possible, thereby reducing the task response time.

The service capabilities (\( SC \)) and current load in virtual machine can be calculated based on relevant the number of processors, average execution speed of instructions per processor, and network bandwidth capability. The calculation formula of \( SC_j \) in virtual machine \( VM_j \)
\[ SC_j = P_j \times I_{mips_j} + BW_j \]  \hspace{1cm} (9)

where \( P_j \) represents the number of processors of the virtual machine \( VM_j \), \( I_{mips_j} \) represents million Instructions per second for each processor and \( BW_j \) represents the network bandwidth capacity of \( VM_j \). Therefore, the maximum service capacity of the system is:

\[ SC_{\text{max}} = \sum_{j=1}^{m} SC_j \]  \hspace{1cm} (10)

The load refers to the ratio of the sum of all the tasks on the virtual machine to the service rate.

\[ Load_j = \frac{\text{Task}_n}{S_{\text{rate}_j}} \]  \hspace{1cm} (11)

\text{Task}_n \) represents the number of tasks in the virtual machine \( VM_j \) service queue at a certain moment. \( S_{\text{rate}_j} \) represents the service speed of the virtual machine \( VM_j \) at the same time and its size is of the virtual machine.

So, the full load of all the virtual machines \( i \)

\[ Load_{\text{sum}} = \sum_{j=1}^{m} Load_j \]  \hspace{1cm} (12)

The processing time \( (PT_j) \) of all tasks in the virtual machine \( VM_j \) is defined as the \( load_j \) is the load of \( VM_j \) Service Capability \( SC_j \) ratio:

\[ PT_j = \frac{Load_j}{SC_j} \]  \hspace{1cm} (13)

The average processing time on all virtual machines is the ratio of the total load to the total service size of the virtual machine service:

\[ PT_{\text{avg}} = \frac{Load_{\text{sum}}}{SC_{\text{max}}} \]  \hspace{1cm} (14)

The standard deviation is calculated to reflect the load difference of each virtual machine. The standard deviation is calculated as:

\[ \text{Std}_{\text{Load}} = \sqrt{\frac{1}{m} \sum_{j=1}^{m} (PT_j - PT_{\text{avg}})^2} \]  \hspace{1cm} (15)

From Eq. (9), it can be observed that the smaller the \( \text{Std}_{\text{Load}} \), the more balanced load system, and by setting \( \text{Std}_{\text{Load}} \) threshold \( \alpha \in [0, 1] \), then when \( \text{Std}_{\text{Load}} < \alpha \), the system load reaches equilibrium, otherwise, the system is negative or unbalanced.

### 4.3. Task migration

The calculation of the maximum service capacity \( (SC_{\text{max}}) \) of the system from equation (9–12) and the current total work \( (Load_{\text{sum}}) \), if \( Load_{\text{sum}} > SC_{\text{max}} \), then the final system cannot reach the load balance, otherwise, load balancing operations can be performed by the following:
• First, according to Naive Bayes Method, divide all virtual machines into sets \( LSet, BSet, \) and \( HSet \), and calculate each set in \( LSet \). The resources available in the virtual machine (Supply resource) \( \text{Sup.res}_j = \text{S.cap}_j - \text{Load}_j \). Resources required for each virtual machine in \( HSet \) (Demand resource) \( \text{Dem.res}_j = \text{Load}_j - \text{S.cap}_j \);

• Second, \( LSet \) and \( HSet \). The virtual machines are sorted in descending order by \( \text{Sup.res} \) and \( \text{Dem.res} \), respectively;

• Finally, the virtual tasks on the plane are sorted by priority from the highest to the lowest, making it easy to be scheduled and find the most suitable target virtual machine.

Assume that all tasks have a total of \( k \) priorities from high to low \( p_1 > p_2 > \ldots > p_i > \ldots > p_k \) and set the task of migration degree \( T_{pi} \in VMs \in Hset \), the target virtual machine \( VM_d \in Lset \) should meet the following equation:

\[
\min \left( \sum T_{p_j} \right) \in VM_d \text{ and } p_j > p_i 
\]  

The target virtual machine has priority \( p_i \). The highest total number of tasks guaranteed \( T_{pi} \) can be executed as soon as possible. The pseudo code of LBBA algorithm to simulate bat algorithm behaviour as shown in Fig. 3.

Step1: Initialize;
Step 2: calculate \( S\_cap, \text{Load, Std\_Load} \) from equation (3-9)
Step 3: if (\( \text{Std\_Load} \leq \alpha \)) break;
Step 4: else if (\( \text{Load\_sum\_max} > S\_cap \)) then Stop receives tasks or increase resource;
else Classify VMs into, \( Lset, Bset, Hset \) based on naive Bayesian;
Step 5: Calculate \( \text{Sup\_res} \) for each \( VM \) in \( Lset \); Calculate \( \text{Dem\_res} \) for each \( VM \) in \( Hset \);
Sort VMs in set \( L \) and set \( H \) by descending order;
Step 6: While (\( \text{Set} LSet \neq \emptyset \) and \( Bset \neq \emptyset \)) {
    For (\( s = 1; s \leq Hset\_size; s++ \)) {
        Sort tasks by priority from high to low;
        For (\( T_{pi} \in VM \in Hset \) need to be scheduled) {
            Find \( VM_d \in Lset \) && \( \min \left( \sum T_{p_j} \right) \in VM_d (p_j > p_i) \)
            Apply bat algorithm to update \( Load, task\_num \) and priority
        }
    }
}
Step 7: Update, \( LSet, BSet, HSet \);
Step 8: Sort VMs in \( LSet \) and \( HSet \) by descending order

Fig. 3. Pseudo code of LBBA algorithm to simulate bat algorithm.

As shown in Fig. 3, step (3) if the standard deviation is less than the set threshold, it indicates that the system reaches load balancing, then exit the algorithm. \( \alpha \) value can be adjusted according to the specific circumstances section. In step (4), if the system does not reach equilibrium situation, but the virtual machines total load is greater than the system capacity, it need to stop receiving tasks or increase system resources; otherwise, Load balancing can be performed.

In the end of step 4 dummy Bayes algorithm used to classify \( VM \). In step 5 line sorts the virtual machines in \( Hset \) in descending order by \( \text{Dem\_res} \).
relieve the load on the heavily loaded virtual machines; sort \( Sup_{res} \) in descending order of \( L_{set} \) virtual machines and prioritize the tasks in the virtual machine from highest to lowest,

Finally, in step 6 line task migration operation, according to conditions will the target virtual machine and update the virtual machine information.

5. Experiments and Results Analysis

The experimental environment used in this paper, 2.1GHz CPU, 2GB of memory, 32Bit Windows 7 operating system, OPNET 14 model programming tools. The Main function of this system is to calculate completion time, request time, response time and number of tasks being migrated. Figure 4 illustrates cloud computing simulation platform. The LBBA algorithm in this paper uses the same experimental loop as the DLB and WRR algorithms Habitat, experimental comparison of parameter values for the following:

i. Tasks number (50,100,150,200,250,300,350,400).

ii. Total demand of CPU (2.4 GHz).

iii. Total demand of RAM(2 GB)

![Figure 4: Cloud computing simulation.](image)

For the compression of the completion time of load balancing before and after LBBA algorithm, there are more balanced for system load and less the task completion time, the experimental data are shown in Table 2.

In order to evaluate the performance of LBBA algorithm, in the same experimental environment, comparison of the data in Table 2, it can be seen that the use of LBBA algorithm results in that the task completion time is reduced by nearly 50%. Figure 5 shows the comparison results between them.
Table 2. Task completion time before and after using LBBA.

<table>
<thead>
<tr>
<th>No. of task</th>
<th>Before (s)</th>
<th>After (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>57</td>
<td>29</td>
</tr>
<tr>
<td>100</td>
<td>111</td>
<td>57</td>
</tr>
<tr>
<td>150</td>
<td>191</td>
<td>107</td>
</tr>
<tr>
<td>200</td>
<td>278</td>
<td>159</td>
</tr>
<tr>
<td>250</td>
<td>310</td>
<td>172</td>
</tr>
<tr>
<td>300</td>
<td>371</td>
<td>194</td>
</tr>
<tr>
<td>350</td>
<td>469</td>
<td>236</td>
</tr>
<tr>
<td>400</td>
<td>575</td>
<td>287</td>
</tr>
<tr>
<td>450</td>
<td>648</td>
<td>335</td>
</tr>
</tbody>
</table>

Fig. 5. Task completion time before and after used LBBA.

Also, from the comparison experiments results among algorithms (LBBA, DLB, and WRR), take the average of multiple results which are the completion time, request response time and moved shift the number of tasks respectively. The experimental results are shown in Table 3.

Table 3. Completion time and response time for algorithms

<table>
<thead>
<tr>
<th>No. of tasks</th>
<th>Completion time(ms)</th>
<th>Response time(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WRR</td>
<td>DLB</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>29</td>
</tr>
<tr>
<td>100</td>
<td>98</td>
<td>60</td>
</tr>
<tr>
<td>150</td>
<td>174</td>
<td>115</td>
</tr>
<tr>
<td>200</td>
<td>251</td>
<td>170</td>
</tr>
<tr>
<td>250</td>
<td>259</td>
<td>206</td>
</tr>
<tr>
<td>300</td>
<td>270</td>
<td>242</td>
</tr>
<tr>
<td>350</td>
<td>379</td>
<td>301</td>
</tr>
<tr>
<td>400</td>
<td>480</td>
<td>360</td>
</tr>
<tr>
<td>450</td>
<td>528</td>
<td>411</td>
</tr>
</tbody>
</table>

By comparing the experimental results of the algorithms LBBA, DLB, and WRR it can be seen in Table 3 that when the algorithm LBBA is used the task
completion time is less and with the task as the number increases. The advantage is more obvious as the number of tasks increases.

Again, the second part of Table 3 shows that the response time is less when it compare with traditional algorithms and increases in the number of tasks. It compares response time values for these three algorithms. Due to the load of the virtual machine LBBA algorithm divided into three categories \(L_{set}, H_{set}, B_{set}\), the task scheduling is only performed between \(H_{set}\) and \(L_{set}\). The scheduled task in \(H_{set}\) needs only to consider the set \(L_{set}\) when selecting the target virtual machine. So, the LBBA algorithm in searching for the optimal solution in the entire virtual machine group, it can reduce the request response time more effectively, reduces the time spent searching for the target virtual machine and reduces the task waiting time when compared the number of task migration for Table 4.

<table>
<thead>
<tr>
<th>Virtual machine number</th>
<th>Number of tasks to be transferred (migrate 100 tasks)</th>
<th>Number of tasks to be transferred (migrate 200 tasks)</th>
<th>Number of tasks to be transferred (migrate 300 tasks)</th>
<th>Number of tasks to be transferred (migrate 400 tasks)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LBB A</td>
<td>DL B</td>
<td>LBB A</td>
<td>DL B</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>38</td>
<td>52</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>32</td>
<td>42</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>26</td>
<td>35</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>16</td>
<td>30</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>18</td>
</tr>
</tbody>
</table>

Table 4 shows the comparison of the number of migrated tasks as increase the number of virtual machines. In order to have a more balanced system load, the number of tasks that are needed to be scheduled in LBBA algorithm, is less than number of DLB algorithm, which indicates better results than DLB algorithm in the system load balancing. It also shows that the more virtual machines, fewer tasks scheduled.

Table 5 illustrates the compare of running time between LBBA and MEMO in response time with numbers of request tasks [14].

<table>
<thead>
<tr>
<th>Number of request</th>
<th>MEMO algorithm running time (ms)</th>
<th>LBBA algorithm running time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>29</td>
<td>72</td>
</tr>
<tr>
<td>200</td>
<td>61</td>
<td>123</td>
</tr>
<tr>
<td>300</td>
<td>83</td>
<td>164</td>
</tr>
<tr>
<td>400</td>
<td>104</td>
<td>254</td>
</tr>
</tbody>
</table>

Despite of running time in MEMO is less than LBBA, But it does not serve argent requests and LBBA is more speed for migrated tasks for one resources to other also LBBA is more organized and efficiently working.
6. Conclusions

This paper proposes a load balancing strategy based on bat algorithm mechanism in cloud computing environment. It aims to balance the system load to a certain extent. The naive Bayesian algorithm is used to divide the state of the virtual machine so that the task assignment is more rapid and reasonable. For heavy load, the virtual need task distribution more quickly and reasonably. The factor of task priority is considered when selecting the target virtual machine. The mission to be migrated on the aircraft as a bat echolocation to acquaint with perceives target for bat interest is to find task priority as a target virtual machine. The load balancing technique reduces the response time and completion time for task request also reduces number of tasks being migrated. By comparing the LBBA algorithm with the existing one, the results show that the proposed algorithm is of better performance, better resource utilization and better response time without adding system overhead. It considers the task as independent, non-preemptive, and the QoS considered is only task priority.

References


