6. IMPROVED EFFICIENT DYNAMIC RESOURCE OPTIMIZATION ALGORITHM

6.1 INTRODUCTION

Computing resources are presented as virtual machines. In such a scenario, scheduling algorithms play an important function where the intention is to schedule the tasks effectively so as to reduce the response and execution time and improve the resource usage. This is achieved in EDFRLS and EDRAP algorithms. But still improve the cloud performance an IEDRO algorithm is used.

The proposed IEDRO algorithm uses CloudSim toolkit for experimentation. The experimental results indicate that the proposed algorithm shows better performance under heavy loads. The primary aim of the proposed method is to improve the efficiency of the cloud. The focus of this chapter is to optimize the resource, increase the speed of cloud computing and minimize the response and the execution time. In the previous chapters, the presentations about the parameters and the results are as follows:

- Latency, execution and response time is better in EDFRLS
- Delay, memory reliability and failure VM identification is better in EDRAP.

These parameters give better performance in cloud computing. Therefore, an attempt is made to improve more resource and speed. To achieve this, an algorithm IEDRO has been proposed. This chapter deals with the resource utilization and the speed in cloud computing. The Improved Efficient Dynamic Resource Optimization gives efficient result in simulation analysis.

6.2 IMPROVED EFFICIENT DYNAMIC RESOURCE OPTIMIZATION

A cloud computing uses many data centers with many VMs based on the physical resources of cloud computing. The virtual machines are mapped as a VM.
Each VM has properties used for processing the VM. The proposed algorithm uses VM properties with parameters. The IEDRO algorithm allocates the load in efficient VMs based on the time consuming and resource level in a VM. These two parameters give more effective solutions. Initially, the fastest response time VM is placed based on the time limit and specified a priority value. After specifying the priority value, the resource level percentage is counted and the first load is allocated to first priority VM and thus along. This gives faster load and efficient outcome than the existing two algorithms.

6.2.1 Parameters used in IEDRO

- $t_{\text{limit}}$ denotes the limit of the time 1 to 1.5 ms
- $M^\text{max}$ maximum free memory space
- $\text{max}_{\text{COUNT}}$ denotes the reliable VM count
- $T^U$ denotes the upper limit of the time 1.5 ms
- $T^L$ denotes the lower limit of time 1 ms.
- $SP_i$ denotes the CPU speed
- $M_i$ denotes the memory size
- $RLP$ denotes the resource level percentage.

The proposed algorithm IEDRO is derived from the two existing algorithms and produces efficient results. The IEDRO includes all the efficient parameters in the EDFRLS and EDRAP algorithms. The proposed algorithm IEDRO uses five parameters, resource optimization, speed, delay, identifying failure VMs and efficiency. The purpose of this research is to improve the fastness of the cloud and balanced load to the cloud data center. To reach these finishes, three algorithms have been developed EDFRLS, EDRAP and IEDRO. The IEDRO is derived from EDFRLS and EDRAP. Figure 6.1 illustrates the requirements of the IEDRO algorithm.
FIGURE 6.1: IEDRO ALGORITHM REQUIREMENTS

1. Latency
2. Execution Rate
3. Resource Level Percentage

1. Bandwidth
2. Resource Utilization
3. Response time
4. Reliability access mechanism

1. Latency
2. Delay
3. Response time
4. Execution Time
5. Resource level Percentage
6. Priority Identification
Resource Optimization

The first parameter is the resource optimization. In this algorithm, it is worn that the resources are used in an effective way. The RLP parameter optimizes the resource and loads the data in a better way.

\[
\text{Latency Rate} = \frac{\sum_{t=0}^{n} (CT_t - AT_t) - BT_t}{n}
\]

\[
\text{Mean Execution Rate} = \frac{\sum_{t=0}^{n} (PT_t - ST_t)}{\sum_{t=0}^{n} 1}
\]

\[
RRL = \sum_{P} \frac{P \cdot Max_x + e^{\log_2 1} + e^{\log_2 L}}{2 \cdot e^{\log_2 (L+1)}} + \frac{E}{n}
\]

Resource Reliability (RR)

Reliability of any resource is evaluated along a reliable scale, hence helping in allocating efficient resources to virtual machine speed. The speed is calculated between middleware and the VM. The threshold value is specified as the lower limit of 1 and the upper bound of 1.5 ms. The load is allocated to the fast response VM within the time limit and the remaining VMs are required subsequently.

Delay

The delay is minimized in the proposed IEDRO algorithm. The delay is taken at the time of data loading in the VM. The fast response VM gives less delay.

Efficiency

The efficiency is achieved by a better way than the EDFRLS and EDRAP. The efficiency is computed by using the efficient parameters. The time limit is evaluated based on the CPU utilization of the VM. The time limit taken in the proposed methodology is between 1 to 1.5 ms.
The VM responds to middleware based on the theme. The equation (1) gives the result of VM response within the time $1$

$$\sum_{i=1}^{1.5} \max\left(t^U(t)\right) = M^L(t_{limit}) \leq t_{limit} \quad (1)$$

The equation (2) gives the result of VM response exceeding the time.

$$\sum_{i=1}^{1.5} \max\left(t^U(t)\right) = M^L(t_{limit}) \geq t_{limit} \quad (2)$$

The equation (3) gives the result of two or more VMs responses at the same time.

$$\sum_{i=1}^{1.5} \max\left(t^U(t)\right) = M^L(t_{limit}) \sum_{i=1}^{n} t_{limit} = VM1 = VM2 = VMn \quad (3)$$

The equation (4) shows the resource level percentage for the VM in the cloud data enter.

$$RLP = \sum \frac{p \log_2 c + \log_2 L}{n} \quad (4)$$

The resource utilization is efficiently applied in the proposed load balancing parameter resource level percentage. The proposed methodology has the combination of both EDFRLS and EDRAP algorithms.

**Identify Failure VM**

The failure VM identification is used at the time data loading. The client transmits data to middleware. The middleware forward that request to all VMs in the datacenter. The VM which contains an effective CPU utilization gives fast response to the middleware. The middleware identifies the failure VM based on failure message.
FIGURE 6.2: FLOW CHART OF IEDRO
6.2.2 Steps in IEDRO

Step 1: The load balancer gets the data from CSP. The load balancer is a middleware intermediate to CSP and data center.

Step 2: The middleware initially sends the request to data center to identify the failure and reliable VM within the time limit 1 to 1.5 ms. In data center there are multiple VMs are available. The data center receives the request form middleware and transfers the request to all VMs in their data center. All the VMs respond to the middleware depending upon the status such as total memory, used memory, free memory, latency rate and execution rate are calculated.

Step 3: The middleware lists the VM based on the response time except failure VM. The middleware allocates the load based on the priority. The priority is assigned

- If the VMs respond within the time, the less time limit will take first and so on.
- If two or more VMs respond at the same time limit, the less memory utilization VM needs as the next priority.
- If the VM exceeds the time limit will assign as last priority and is the two or more VMs respond exceed the time limit the priority is assigned based on the IP address.

Step 4: Resource level percentage is calculated percentage. This process is done for each VM separately.

Step 5: Using the fuzzy rules the resource level is identified as low, medium and high.

Step 6: The load is assigned to the VM based on by with the percentage identified by the resource level percentage parameter.
6.2.3 Pseudo Code IEDRO

1. For (Every VM identifies RLP and allocate load based on priority)
2. $T^U \leftarrow$ denotes the upper limit of time 1.5 ms
3. $T^L \leftarrow$ denotes the lower limit of time 1 ms
4. $t_{\text{limit}} \leftarrow$ Time limit for response of VM within time
5. $\text{for} \ (\text{all VM } i=1 \text{ to } n) \text{ identifies the response time}$
6. $P_i \leftarrow$ identify the priority based on the VM response to request.
7. $F_i \leftarrow$ identify the failure VM
8. $\text{RLP} \leftarrow$ calculate the resource percentage with execution and latency for all VM
9. Update the status to middleware
10. $\text{max}_\text{COUNT}$ denotes the reliable VM count
11. $\text{RLP}$ denotes resource level percentage.
12. vmLoad$\leftarrow$ h.getBw()+h.getRam()+h.getMips();
13. allVmLoad get the vmLoad;
14. VM select the load BrokerLoad.add(allVmLoad);
15. brokerLoad.add(allVmLoad);
16. resourceOptimization().setVisible(true);
17. for(int $i=0$;$i<$demand.size();$i++)
18. if (load reach RLP value transfer the load to next VM.
19. endif
20. endif
6.3 EXPERIMENTAL RESULTS OF IEDRO

The experimental results of the proposed IEDRO algorithm show better outcomes than the EDFRLS and EDRAP. The analysis of IEDRO is shown below.

<table>
<thead>
<tr>
<th>Virtual machine</th>
<th>VM Status</th>
<th>Reconciliation Computation</th>
<th>Reliability in Milliseconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM1</td>
<td>True</td>
<td>True</td>
<td>1.142</td>
</tr>
<tr>
<td>VM2</td>
<td>True</td>
<td>True</td>
<td>1.477</td>
</tr>
<tr>
<td>VM3</td>
<td>True</td>
<td>False</td>
<td>1.695</td>
</tr>
<tr>
<td>VM4</td>
<td>False</td>
<td>False</td>
<td>1.916</td>
</tr>
<tr>
<td>VM5</td>
<td>False</td>
<td>True</td>
<td>1.213</td>
</tr>
<tr>
<td>VM6</td>
<td>True</td>
<td>True</td>
<td>1.119</td>
</tr>
<tr>
<td>VM7</td>
<td>True</td>
<td>False</td>
<td>1.743</td>
</tr>
<tr>
<td>VM8</td>
<td>True</td>
<td>True</td>
<td>1.456</td>
</tr>
<tr>
<td>VM9</td>
<td>True</td>
<td>True</td>
<td>1.235</td>
</tr>
<tr>
<td>VM10</td>
<td>True</td>
<td>True</td>
<td>1.428</td>
</tr>
</tbody>
</table>

Table 6.1 indicates the status of all the VMs in the cloud data center. Before processing the load in VMs, the middleware gets the status for each VM on the basis of time limit lower bound of 1 and the upper bound of 1.5 ms.
<table>
<thead>
<tr>
<th>Virtual Machine</th>
<th>VM Status</th>
<th>Reconciliation Computation</th>
<th>Reliability in Milliseconds</th>
<th>Priority of VM to allocate Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM1</td>
<td>True</td>
<td>True</td>
<td>1.142</td>
<td>2</td>
</tr>
<tr>
<td>VM2</td>
<td>True</td>
<td>True</td>
<td>1.477</td>
<td>6</td>
</tr>
<tr>
<td>VM3</td>
<td>True</td>
<td>False</td>
<td>1.695</td>
<td>7</td>
</tr>
<tr>
<td>VM6</td>
<td>True</td>
<td>True</td>
<td>1.119</td>
<td>1</td>
</tr>
<tr>
<td>VM7</td>
<td>True</td>
<td>False</td>
<td>1.743</td>
<td>8</td>
</tr>
<tr>
<td>VM8</td>
<td>True</td>
<td>True</td>
<td>1.456</td>
<td>5</td>
</tr>
<tr>
<td>VM9</td>
<td>True</td>
<td>True</td>
<td>1.235</td>
<td>3</td>
</tr>
<tr>
<td>VM10</td>
<td>True</td>
<td>True</td>
<td>1.428</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 6.2 shows a reliable VM list based on operation. The table is derived from the equations (1), (2), (3) and (4). The failure VM is not taken from this list. The load is allocated based on the priority of the VM which satisfies the conditions. The following are the process of IEDRO:

- To calculate and assign the priority for each request based on the threshold value and allocate the service to each request’s.

- Understand the client request data within the time and requested server for each request and the tasks is determining the time priority value based on the predefined conditions. Assign priority value to each task for the client’s request.

- For each request and its tasks find the node priority value based on the predefined conditions Assign priority value to each task for the client’s request. For each client’s input data check whether it is within the threshold value or not If the input value is within the threshold limit and total node <= available node). Add respective computed time and node priority value.
The proposed IEDRO gives the efficient result for load balancing algorithm than the EDFRLS. For better understanding, the efficiency of the proposed algorithm has been compared with the EDFRLS and EDRAP and the new derived algorithm IEDRO gives better outcomes than the other two algorithms. The IEDRO gives an increase in speed and decrease in delay than EDFRLS and EDRAP.

In the first phase of the proposed EDFRLS, the load allocated to all VM in a balanced way. In the second phase of the proposed EDRAP, a single VM is identified for reliability to process the client request among multiple VMs where the client data is replicated. In the third phase of the proposed IEDRO method, the load is allocated to the particular VM. The proposed IEDRO method follows the more VM response same time limit and the VM which is first will be allocated based on IP address. The IEDRO gives the efficient outcome than the other two algorithms. In the proposed IEDRO, the speed increases than the EDFRLS algorithm. These efficiency gives the best effect.

**TABLE 6.3: COMPARISON OF DELAY**

<table>
<thead>
<tr>
<th>VM</th>
<th>DELAY ANALYSIS</th>
<th>EDRAP</th>
<th>IEDRO</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td></td>
<td>0.19</td>
<td>0.10</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>0.29</td>
<td>0.12</td>
</tr>
<tr>
<td>75</td>
<td></td>
<td>0.32</td>
<td>0.19</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>0.28</td>
<td>0.21</td>
</tr>
<tr>
<td>125</td>
<td></td>
<td>0.29</td>
<td>0.24</td>
</tr>
<tr>
<td>150</td>
<td></td>
<td>0.31</td>
<td>0.25</td>
</tr>
<tr>
<td>175</td>
<td></td>
<td>0.25</td>
<td>0.15</td>
</tr>
<tr>
<td>200</td>
<td></td>
<td>0.26</td>
<td>0.11</td>
</tr>
</tbody>
</table>
The three proposed algorithms improve the efficiency of public cloud computing. In public cloud, many users share a single data center and allocate the load where the free space is available in the VM. All the status of the load allocation is monitored by middleware and the middleware maintains the status of the data center process provides efficient load monitoring process. Table 6.3 shows the comparative analysis of the proposed algorithms. The derived algorithm IEDRO result shows the reduction in delay compared to EDRAP. The IEDRO gives efficient results than the other two algorithms.

**FIGURE 6.3: DELAY ANALYSIS**

Figure 6.3 identifies the delay analysis of the proposed algorithms. The IEDRO gives the efficient outcome than the EDFRLS algorithm.
Figure 6.4 shows the comparison of the speed analysis with existing PISA and DAP with the proposed algorithm IEDRO. The result shows that the IEDRO gives more speed than other two existing algorithms. The speed is calculated based on the execution time and response time of the VM. The VM speed improvement gives more efficient result in cloud computing.
### Table 6.4: Numerical Analysis of Response Time in EDFRLS and IEDRO

<table>
<thead>
<tr>
<th>VM</th>
<th>RESPONSE TIME</th>
<th>DIFFERENCE IN % DECREASE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EDFRLS</td>
<td>IEDRO</td>
</tr>
<tr>
<td>5</td>
<td>33112.76</td>
<td>31245.79</td>
</tr>
<tr>
<td>10</td>
<td>35129.09</td>
<td>21265.71</td>
</tr>
<tr>
<td>15</td>
<td>26972.98</td>
<td>14367.52</td>
</tr>
<tr>
<td>20</td>
<td>20365.22</td>
<td>10123.55</td>
</tr>
<tr>
<td>25</td>
<td>16342.32</td>
<td>10012.01</td>
</tr>
<tr>
<td>50</td>
<td>12560.26</td>
<td>9973.49</td>
</tr>
<tr>
<td>75</td>
<td>10986.89</td>
<td>9812.34</td>
</tr>
<tr>
<td>100</td>
<td>9809.78</td>
<td>8762.90</td>
</tr>
<tr>
<td>125</td>
<td>7905.65</td>
<td>6712.76</td>
</tr>
<tr>
<td>150</td>
<td>6890.46</td>
<td>5127.08</td>
</tr>
<tr>
<td>175</td>
<td>5256.67</td>
<td>4312.12</td>
</tr>
<tr>
<td>200</td>
<td>2985.90</td>
<td>1432.09</td>
</tr>
</tbody>
</table>

Table 6.4 reveals the numerical analysis of response time for 200 VMs. By comparing the IEDRO gives efficient results than the EDFRLS algorithms. The response time decreases when the number of VMs is gradually increased. The result shows efficiency in speed and delay in the proposed IEDRO.
FIGURE 6.5: RESPONSE TIME ANALYSIS

Figure 6.5 identifies the response time analysis of the proposed algorithms. The IEDRO gives efficient results than the EDFRLS algorithm.
Table 6.5 reveals the numerical analysis of execution time for 200 VMs. By comparing the IEDRO gives efficient results than the EDFRLS algorithms. The execution time decreases when the number of VMs is gradually increased and. The result shows efficiency in speed and delay in the proposed IEDRO.
### TABLE 6.6: NUMERICAL ANALYSIS OF DELAY IN EDRAP AND IEDRO

<table>
<thead>
<tr>
<th>VM</th>
<th>DELAY</th>
<th></th>
<th>Difference in % Decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EDRAP</td>
<td>IEDRO</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.09</td>
<td>0.02</td>
<td>-78</td>
</tr>
<tr>
<td>10</td>
<td>0.10</td>
<td>0.05</td>
<td>-50</td>
</tr>
<tr>
<td>15</td>
<td>0.12</td>
<td>0.08</td>
<td>-33</td>
</tr>
<tr>
<td>20</td>
<td>0.15</td>
<td>0.09</td>
<td>-40</td>
</tr>
<tr>
<td>25</td>
<td>0.19</td>
<td>0.10</td>
<td>-47</td>
</tr>
<tr>
<td>50</td>
<td>0.29</td>
<td>0.12</td>
<td>-59</td>
</tr>
<tr>
<td>75</td>
<td>0.32</td>
<td>0.19</td>
<td>-41</td>
</tr>
<tr>
<td>100</td>
<td>0.28</td>
<td>0.21</td>
<td>-30</td>
</tr>
<tr>
<td>125</td>
<td>0.29</td>
<td>0.24</td>
<td>-17</td>
</tr>
<tr>
<td>150</td>
<td>0.30</td>
<td>0.25</td>
<td>-12</td>
</tr>
<tr>
<td>175</td>
<td>0.25</td>
<td>0.15</td>
<td>-40</td>
</tr>
<tr>
<td>200</td>
<td>0.26</td>
<td>0.11</td>
<td>-58</td>
</tr>
</tbody>
</table>

Table 6.6 reveals the numerical analysis of delay for 200 VMs. By comparing with IEDRO, the proposed algorithm gives efficient results than the EDRAP algorithms. The delay is decreased when the number of VM is increased. The result shows efficiency delay in the proposed IEDRO.
TABLE 6.7: AVERAGE SPEED FOR IEDRO IN 10 COMPUTING CYCLES WITH 200 VMS

<table>
<thead>
<tr>
<th>VM</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6.03</td>
<td>6.23</td>
<td>7.03</td>
<td>6.02</td>
<td>6.52</td>
<td>6.50</td>
<td>6.35</td>
<td>7.52</td>
<td>7.83</td>
<td>6.53</td>
</tr>
<tr>
<td>50</td>
<td>9.00</td>
<td>8.00</td>
<td>6.00</td>
<td>7.00</td>
<td>8.58</td>
<td>8.08</td>
<td>8.68</td>
<td>7.98</td>
<td>6.98</td>
<td>9.00</td>
</tr>
<tr>
<td>75</td>
<td>9.78</td>
<td>10.70</td>
<td>10.00</td>
<td>11.2</td>
<td>10.02</td>
<td>10.20</td>
<td>9.20</td>
<td>8.90</td>
<td>8.09</td>
<td>9.78</td>
</tr>
<tr>
<td>100</td>
<td>9.07</td>
<td>9.50</td>
<td>10.06</td>
<td>11.76</td>
<td>9.70</td>
<td>10.60</td>
<td>10.9</td>
<td>10.80</td>
<td>11.80</td>
<td>9.07</td>
</tr>
<tr>
<td>125</td>
<td>11.35</td>
<td>10.32</td>
<td>11.42</td>
<td>11.00</td>
<td>11.22</td>
<td>10.22</td>
<td>11.12</td>
<td>11.20</td>
<td>10.90</td>
<td>11.35</td>
</tr>
<tr>
<td>150</td>
<td>10.98</td>
<td>11.97</td>
<td>10.78</td>
<td>11.38</td>
<td>10.08</td>
<td>12.28</td>
<td>11.00</td>
<td>10.60</td>
<td>10.80</td>
<td>11.97</td>
</tr>
<tr>
<td>173</td>
<td>10.32</td>
<td>11.34</td>
<td>11.35</td>
<td>10.04</td>
<td>10.45</td>
<td>11.50</td>
<td>12.15</td>
<td>11.10</td>
<td>10.65</td>
<td>11.34</td>
</tr>
<tr>
<td>200</td>
<td>12.43</td>
<td>12.41</td>
<td>11.51</td>
<td>10.41</td>
<td>12.18</td>
<td>11.08</td>
<td>11.18</td>
<td>12.58</td>
<td>12.50</td>
<td>12.41</td>
</tr>
</tbody>
</table>
Table 6.7 shows the numerical analysis of speed in 10 computing cycle with 200 VM. The efficiency of the proposed IEDRO algorithm gives increase in speed.

6.4 SIMULATION RESULTS

The proposed algorithms use CloudSim for analysis of the results. The three algorithms EDFRLS, EDRAP and IEDRO are compared and efficient solution is attained. The simulation uses 200 VMs and achieves the result. The IEDRO gives the following results

- The resource is optimized efficiently
- Increases speed
- Reduces delay time.
- Efficient load balancing
- Efficient, reliable VM identification

The speed is a specific parameter in cloud computing. To increase the performance, such as speed and resource utilization, many algorithms have been developed. The proposed algorithm gives more efficient solution than the existing algorithms. Increasing speed and resource utilization brings more efficiency in the cloud.

The primary aim is to ensure efficient load balancing with reliability and to improve speed in cloud computing. This chapter explains the overall view of the existing PISA and DAP with EDFRLS, EDRAP and IEDRO requirements, performance analysis, simulation results and the overall framework.

The performance analysis has been performed in 200 VMs with different computing cycles. The proposed algorithms are implemented in CloudSim 3.0. Each simulation is performed for different VMs at different properties. The cloud simulator can set this configuration and process the metrics and record the outcome.
**TABLE 6.8: NUMERICAL ANALYSIS OF CPU PROCESSING TIME FOR IEDRO**

<table>
<thead>
<tr>
<th>Computing Cycle</th>
<th>CPU Processing Time (Average)</th>
<th>Efficiency Based on Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean (Average)</td>
<td>Mean %</td>
</tr>
<tr>
<td>1</td>
<td>9.39784</td>
<td>93.9784</td>
</tr>
<tr>
<td>2</td>
<td>9.47666</td>
<td>94.7666</td>
</tr>
<tr>
<td>3</td>
<td>9.38921</td>
<td>93.8921</td>
</tr>
<tr>
<td>4</td>
<td>9.41597</td>
<td>94.1597</td>
</tr>
<tr>
<td>5</td>
<td>9.42795</td>
<td>94.2795</td>
</tr>
<tr>
<td>6</td>
<td>9.43318</td>
<td>94.3318</td>
</tr>
<tr>
<td>7</td>
<td>9.42868</td>
<td>94.2868</td>
</tr>
<tr>
<td>8</td>
<td>9.48309</td>
<td>94.8309</td>
</tr>
<tr>
<td>9</td>
<td>9.55468</td>
<td>95.5468</td>
</tr>
<tr>
<td>10</td>
<td>9.6478</td>
<td>96.4780</td>
</tr>
<tr>
<td>Entire Computing Cycle</td>
<td>9.4655</td>
<td>94.6550</td>
</tr>
</tbody>
</table>

Table 6.8 records the simulation analysis of CPU processing time for IEDRO algorithm. The simulation results proves increase in efficiency by 94%.

### 6.5 CREATION OF CLOUD ENVIRONMENT

Cloudsim environment consists of

**Datacenters**

Data center class is a cloud resource whose host list are virtualized. It deals with processing of VM queries (i.e., handling of VMs) instead of processing Cloudlet-related questions. Thus, even though an allocation policy will be instantiated, it will
not be applied, as processing of tasks are handled by the Cloudlet scheduler and processing of virtual machines are treated by the VM allocation policy.

**Cloud brokers and Cloudlet(jobs)**

Cloud broker actually establishes the link between data centres and the virtual machines and routines as the interface between them. It enables the VM management, submitting Cloudlets to the respective VMs. The Cloudlet is actually referring to cloud tasks or tasks that are being taken to the virtual machines.

**Virtual Machines**

VM stands for virtual machines. It moves within a host, sharing the host list with other VMs. It processes Cloudlets. This processing takes place according to a policy, defined by the Cloudlet Scheduler. Each VM has an owner, which can submit Cloudlets to the VM to be accomplished.

**Host**

Host consists of physical components that communicate with one another using logical component (software and protocols).

1) Resource Utilization

   a. Based on processing power of virtual machines

   b. Processing power, includes

      i. RAM

      ii. Speed

      iii. Processing time and Execution time (CPUs)

2) Resource Optimization

   a. Load computation based on the above parameters.

   b. Usage of resources based on load of each VM.
6.6 OVERALL FRAMEWORK

Figure 6.6 depicts the overall functioning of the proposed algorithms. The operation is computed using CloudSim and the result is efficiently accomplished by three algorithms: EDFRLS, EDRAP and IEDRO.
FIGURE 6.7: FLOW CHART FOR PROPOSED ALGORITHMS

START

For each VM

Identify the Load

NO

Is Active VM

YES

Allocate Load

Execute EDFRLS, EDRAP and IEDRO for improving cloud speed

End
6.7 PERFORMANCE ANALYSIS OF SIMULATION RESULTS PROPOSED ALGORITHMS

The carrying out of the proposed algorithms proves efficient. The proposed algorithms EDFRLS, EDRAP and IEDRO give better performance than the existing algorithms. These algorithms use 200 VMs in a single cloud data centre with three CPUs.

The proposed methodology consists of three phases. In cloud system, it is possible that some nodes to be heavily charged and the other is lightly loaded. This state of affairs can contribute to poor execution. The goal of load balancing is distributing the load among nodes in a cloud environment.

Load balancing is one of the key topics in cloud computing. For better resource utilization it is necessary to optimize the resource on cloud. Therefore, a load balancing algorithm tries to balance the entire system load by taking resource level percentage and assign priority based on the threshold time limit.

When considering performance from the point of view, the metric involved is often the response time of the operations. However, when the operation is studied from the research point of view, the metric involved is total system throughput to response time, throughput is concerned with ascertaining that all users are treated fairly and that all are making advancement.

To improve the performance of the system and high resource allocation ratio to load balancing mechanism in the host. The characteristics of load balancing are:

- Distribute load evenly across all the nodes.
- To achieve a high user satisfaction.
- Improving the overall performance of the system.
- To reduce response time.
Table 6.9 illustrates simulation metrics used in the proposed algorithms. The simulation analysis uses 200 VMs for computation. The time threshold is 1 to 1.5 ms based on the intermediate performance of 200 VMs. The memory allocation of these is taken from the physical memory of the cloud data center. In the proposed methodology the CloudSim uses the single data center with 200 VM. The VM allocation policy shares the data from one data center to another data center.

In the first phase, a new algorithm EDFRLS has been derived from PISA. The analysis has been performed to improve speed, reduce response time, execution time and reduce delay and it has been compared with the performance of Priority Scheduling Algorithm. The simulation result shows that EDFRLS improves the focal ratio and reduces delays in the cloud environment.

In the second phase, a new algorithm EDRAP has been derived from DAP. The proposed algorithm reduces the delay and increases the speed. The failure VM is identified with this method. The reliable VM is identified efficiently.
In the third phase, a new algorithm is derived from EDFRLS and EDRAP called as IEDRO. The IEDRO gives a more efficient result, such as increases in speed, decrease in delay and efficiently optimizing the resources.

The aim of this thesis is to provide the best result in load balancing and reliability algorithms as well as to improve the performance of the cloud. As a result, this research work proposes a new algorithm to give better result. From the simulation result, it is proved that IEDRO provides better result in all cloud performance.

In cloud the performance, such as an increase in speed, increase in response and execution time and efficient resource utilization is achieved. Therefore, it offers the best result in terms of all parameters than the existing PISA and DAP. Hence, it has reached the goal.

**TABLE 6.10: VM SPEED ANALYSIS WITH PISA, EDFRLS AND IEDRO**

<table>
<thead>
<tr>
<th>VM</th>
<th>SPEED in MS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PISA</td>
</tr>
<tr>
<td>5</td>
<td>2.0054</td>
</tr>
<tr>
<td>15</td>
<td>2.5456</td>
</tr>
<tr>
<td>25</td>
<td>3.0007</td>
</tr>
<tr>
<td>50</td>
<td>3.6005</td>
</tr>
<tr>
<td>75</td>
<td>4.1351</td>
</tr>
<tr>
<td>100</td>
<td>4.7000</td>
</tr>
<tr>
<td>125</td>
<td>4.9011</td>
</tr>
<tr>
<td>150</td>
<td>5.2412</td>
</tr>
<tr>
<td>175</td>
<td>5.7010</td>
</tr>
<tr>
<td>200</td>
<td>6.0001</td>
</tr>
</tbody>
</table>
Table 6.10 shows the speed analysis of PISA, EDFRLS with IEDRO. The simulation is experimented and the operations of the proposed EDFRLS and IEDRO are compared with the existing PISA.

**TABLE 6.11: VM DELAY ANALYSIS WITH DAP, EDRAP AND IEDRO**

<table>
<thead>
<tr>
<th>VM</th>
<th>DELAY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DAP</td>
</tr>
<tr>
<td>5</td>
<td>0.10</td>
</tr>
<tr>
<td>15</td>
<td>0.17</td>
</tr>
<tr>
<td>25</td>
<td>0.32</td>
</tr>
<tr>
<td>50</td>
<td>0.33</td>
</tr>
<tr>
<td>75</td>
<td>0.4</td>
</tr>
<tr>
<td>100</td>
<td>0.36</td>
</tr>
<tr>
<td>125</td>
<td>0.38</td>
</tr>
<tr>
<td>150</td>
<td>0.41</td>
</tr>
<tr>
<td>175</td>
<td>0.34</td>
</tr>
<tr>
<td>200</td>
<td>0.42</td>
</tr>
</tbody>
</table>

Table 6.11 shows the delay analysis of DAP, EDRAP with IEDRO. The simulation is experimented and the operations of the proposed EDRA and IEDRO are compared with the existing DAP algorithm. The proposed algorithms EDRAP and IEDRO give better performance. The EDFRLS and EDRAP give efficient result in individual execution. The EDFRLS parameters such as speed, response, execution time, resource allocation gives efficient results. The EDRAP parameters such as delay
and resource utilization gives efficient results. The IEDRO uses these parameters and achieves the outcome better than the EDFRLS and EDRAP.

**TABLE 6.12: VM RESPONSE TIME ANALYSIS WITH PISA, EDFRLS AND IEDRO**

<table>
<thead>
<tr>
<th>VM</th>
<th>RESPONSE TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PISA</td>
</tr>
<tr>
<td>5</td>
<td>42567.98</td>
</tr>
<tr>
<td>15</td>
<td>33792.89</td>
</tr>
<tr>
<td>25</td>
<td>18665.23</td>
</tr>
<tr>
<td>50</td>
<td>14345.56</td>
</tr>
<tr>
<td>75</td>
<td>12456.34</td>
</tr>
<tr>
<td>100</td>
<td>10976.23</td>
</tr>
<tr>
<td>125</td>
<td>8964.67</td>
</tr>
<tr>
<td>150</td>
<td>7980.45</td>
</tr>
<tr>
<td>175</td>
<td>5897.98</td>
</tr>
<tr>
<td>200</td>
<td>4789.78</td>
</tr>
</tbody>
</table>

Table 6.12 shows the response time analysis of EDFRLS with IEDRO. The simulation is experimented and the operations of the proposed EDFRLS and IEDRO are compared with the existing PISA algorithm.
**TABLE 6.13: VM EXECUTION TIME ANALYSIS WITH PISA, EDFRLS AND IEDRO**

<table>
<thead>
<tr>
<th>VM</th>
<th>EXECUTION TIME</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PISA</td>
<td>EDFRLS</td>
<td>IEDRO</td>
</tr>
<tr>
<td>5</td>
<td>35654.09</td>
<td>36754.67</td>
<td>34394.2</td>
</tr>
<tr>
<td>15</td>
<td>25438.71</td>
<td>24980.08</td>
<td>23948.9</td>
</tr>
<tr>
<td>25</td>
<td>18120.67</td>
<td>15342.12</td>
<td>15058.3</td>
</tr>
<tr>
<td>50</td>
<td>14123.78</td>
<td>11980.05</td>
<td>11746.8</td>
</tr>
<tr>
<td>75</td>
<td>11345.12</td>
<td>10867.13</td>
<td>9995.49</td>
</tr>
<tr>
<td>100</td>
<td>10127.01</td>
<td>8956.12</td>
<td>8587.4</td>
</tr>
<tr>
<td>125</td>
<td>8120.78</td>
<td>6970.23</td>
<td>6790.95</td>
</tr>
<tr>
<td>150</td>
<td>6981.89</td>
<td>5980.78</td>
<td>5833.21</td>
</tr>
<tr>
<td>175</td>
<td>5237.78</td>
<td>4980.56</td>
<td>4598.25</td>
</tr>
<tr>
<td>200</td>
<td>4209.12</td>
<td>2890.78</td>
<td>3017.4</td>
</tr>
</tbody>
</table>

Table 6.13 shows the execution time analysis of EDFRLS with IEDRO. The simulation is experimented and the operations of the proposed EDFRLS and IEDRO are compared with the existing PISA algorithm.
FIGURE 6.8: VM EXECUTION TIME ANALYSIS FOR PISA, EDFRLS WITH IEDRO

FIGURE 6.9: VM RESPONSE TIME ANALYSIS FOR PISA, EDFRLS WITH IEDRO
FIGURE 6.10: VM DELAY ANALYSIS FOR PISA, EDFRLS WITH IEDRO

FIGURE 6.11: VM SPEED ANALYSIS FOR IEDRO, EDRAP AND EDFRLS
Figures 6.8, 6.9, 6.10 and 6.11 shows execution time, response time, speed and delay analysis of EDFRLS, EDRAP with IEDRO and are compared with existing load balancing and reliability algorithm.

The following summarizes the results of simulation of the existing algorithm with the proposed algorithms EDFRLS, EDRAP and IEDRO in percentage.

Response Time (in MS)

The proposed EDFRLS and IEDRO algorithms for the research work shows decrease in response time.

- On 5 VM EDFRLS (-22.212%) IEDRO (-27.525%)
- On 10 VM EDFRLS (-24.072%) IEDRO (-39.464%)
- On 15 VM EDFRLS (-6.321%) IEDRO (-46.733%)
- On 20 VM EDFRLS (-8.907%) IEDRO (-50.290%)
- On 25 VM EDFRLS (-12.445%) IEDRO (-38.735%)
- On 50 VM EDFRLS (-12.034%) IEDRO (-20.594%)
- On 75 VM EDFRLS (-11.797%) IEDRO (-12.690%)
- On 100 VM EDFRLS (-10.627%) IEDRO (-15.671%)
- On 125 VM EDFRLS (-11.813%) IEDRO (-20.089%)
- On 150 VM EDFRLS (-13.658%) IEDRO (-25.591%)
- On 175 VM EDFRLS (-10.873%) IEDRO (-17.968%)
- On 200 VM EDFRLS (-37.661%) IEDRO (-52.0382%)
Execution Time (in MS)

The proposed algorithms EDFRLS and IEDRO for the research work shows decrease in execution time

- On 5 VM EDFRLS (-3.087 %) IEDRO (-5.638%)
- On 10 VM EDFRLS (-3.994 %) IEDRO (-8.897%)
- On 15 VM EDFRLS (-1.803 %) IEDRO (-31.341%)
- On 20 VM EDFRLS (-7.87 %) IEDRO (-23.450%)
- On 25 VM EDFRLS (-15.334 %) IEDRO (-38.735%)
- On 50 VM EDFRLS (-15.178 %) IEDRO (-20.594%)
- On 75 VM EDFRLS (-4.911 %) IEDRO (-17.968%)
- On 100 VM EDFRLS (-31.321 %) IEDRO (-52.038%)

Delay

The proposed algorithms EDRAP and IEDRO for the research work show reduce in delay

- On 25 VM EDRAP (-6.25%) IEDRO (-47%)
- On 50 VM EDRAP (-12.12%) IEDRO (-59%)
- On 75 VM EDRAP (-20%) IEDRO (-41%)
- On 100 VM EDRAP (-22.22%) IEDRO (-30%)
- On 175 VM EDRAP (-26.47%) IEDRO (-40%)
- On 200 VM EDRAP (-35%) IEDRO (-58%)
### Table 6.14: Comparative Analysis of EDFRLS and EDRAP Algorithms with IEDRO Algorithm

<table>
<thead>
<tr>
<th>Analysis Values</th>
<th>EDFRLS</th>
<th>EDRAP</th>
<th>IEDRO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response time</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Execution time</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Delay</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Speed</td>
<td>High</td>
<td>High</td>
<td>Very high</td>
</tr>
<tr>
<td>Efficient parameter</td>
<td>Resource Level Percentage</td>
<td>Memory Reliability</td>
<td>Both</td>
</tr>
<tr>
<td>processing time</td>
<td>1.5</td>
<td>1.3</td>
<td>1.2</td>
</tr>
<tr>
<td>Priority based</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Scalability</td>
<td>Medium</td>
<td>Not taken</td>
<td>High</td>
</tr>
<tr>
<td>Performance in %</td>
<td>91%</td>
<td>92%</td>
<td>94%</td>
</tr>
<tr>
<td>Time limit</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Techniques used</td>
<td>Load Balancing</td>
<td>Reliability based on memory</td>
<td>Both</td>
</tr>
<tr>
<td>Nature</td>
<td>Dynamic</td>
<td>Real time</td>
<td>Real time</td>
</tr>
</tbody>
</table>

Table 6.14 shows the overall comparative analysis of proposed algorithms EDFRLS, EDRAP and IEDRO. The IEDRO gives more efficient than EDFRLS and EDRAP.
EDFRLS Metrics with 200 VM

**Figure 6.12: EDFRLS Speed**

**Figure 6.13: EDFRLS Response Time**

**Figure 6.14: EDFRLS Execution Time**
EDRAP Metrics with 200 VM

**FIGURE 6.15: EDRAP DELAY**

**FIGURE 6.16: EDRAP SPEED**

**FIGURE 6.17: EDRAP EFFICIENCY**
IEDRO Metrics with 200 VM

**FIGURE 6.18: IEDRO DELAY**

**FIGURE 6.19: IEDRO RESPONSE TIME**

**FIGURE 6.20: IEDRO SPEED**
6.8 SUMMARY

The load balancing is the major challenge in cloud computing. The existing algorithms give less performance in cloud computing. However, the issue in cloud is load balancing in busy time. The existing algorithms give less solution.

Compared to grid computing major advantage of cloud computing is the ability to provide more reliable and flexible resources are available. In cloud computing the major challenge is resource provisioning and allocation in load to dynamically changing work loads. Currently, most of these methods focused on the optimization of allocating physical resources to their associated virtual resources and migrating virtual machines to achieve load balance and increase resource utilization.

To propose a time limit based dynamic resource allocation strategy for cloud computing that dynamically allocate the virtual machines among the cloud computing applications based on their load changes instead of allocating resources required to satisfy peak demands and can apply the threshold method to optimize the decision of resource reallocation.

Resource utilization includes automatic load balancing. A cloud system may cause an unexpected number of processes gives more processing time. If, the algorithm utilize resources more that give more efficiency to under loaded processors. IEDRO algorithms have lesser resource utilization for load balancing methods to delegate tasks to processors in order to achieve minimized response time am execution time gives better performance. The dynamic load balancing algorithms give better resource utilization than the static load balancing.

Dynamic IEDRO load balancing algorithms are more reliable as processes can be loaded in efficient virtual machine in case of failure occurs IEDRO load balancing algorithms are adaptive towards every situation whether numbers of procedures are paid off or varying one. Dynamic load balancing algorithms may take in relatively higher response time as sometimes redistribution of processes takes place. Some time is being consumed during task loaded. The proposed algorithm gives more effective
results than the existing algorithm. Therefore, an attempt is made to increase the speed and decrease the delay in cloud VM using three different algorithms EDFRLS, EDRAP and IEDRO. Among all the proposed algorithms, IEDRO improves the performance of the cloud computing environment.

IEDRO load balancing algorithms are totally dependent. The active load balancing algorithms are constantly safer than existing techniques by way of fault tolerant, resource usage, response time, reaction time, delay and resource accessibility. It enables an algorithm to keep running the right way in the case of any failure. Think if the performance of algorithm decreases gives the error, but the IEDRO algorithm gives an increase in performance gives no error. Because of this the load is efficiently used and give continuous services to the user without any outages in cloud computing.

All the proposed algorithms are experimented through CloudSim tool. The requirements, flow chart, architecture and performance analysis of the proposed algorithms have been discussed in this chapter. It is concluded that the IEDRO provides better results than EDFRLS and EDRAP. Thus, the proposed algorithm IEDRO is the efficient method for improving the cloud performance and more suited for load balancing in a cloud computing environment.

PUBLICATIONS