Multi-Objective Optimization for Dynamic Resource Provisioning in a Multi-Cloud Environment using Lion Optimization Algorithm

To be Virtually Presented At

20th IEEE International Symposium on Computational Intelligence and Informatics (CINTI 2020) planned to be organized on November 5-7, 2020 in Budapest, Hungary

By

Shivani Agrawal

Authors

T Chaitra
Shivani Agrawal
Jeny Jijo
Arti Arya
AGENDA

• Problem Statement
• System Architecture
• Proposed Algorithm
• Lion Optimization Algorithm
• Multi-Objective Optimization
• Results and Discussion
• Conclusion and Future Scope
• References
In recent years, developments in multi-cloud have been prominent because of its indispensable use in our day-to-day life.

Many recent nature inspired algorithms like grasshopper optimization, hybrid bacteria foraging, whale optimization, etc, are used to solve multi-objective optimization problems like Process Selection, Machine Allocation, Finding the feasible solution and other maximization/minimization problem.

Therefore, in this paper we are using Lion Optimization Algorithm (LOA) for optimizing multi-objectives like minimizing the cost, maximizing the revenue, etc, in order to improve dynamic resource provisioning performance in multi-cloud environment.

As compared to Particle Swarm Optimisation (PSO), we have observed that LOA takes much less time with increasing no. of tasks.
SYSTEM ARCHITECTURE

Input the cloud parameters through Graphical User Interface

Task 1  Task 2  ...  Task N

Lion Optimization Algorithm for optimal mapping of numerous tasks to various virtual machines in Data Centres

Cloud (α)  Cloud (β)  Cloud (χ)

DC1
VM1
VM2
...
DCm
VM1
VM2
...
DCM
VM1
VM2
...
**PROPOSED ALGORITHM**

**PROVISIONING OF TASKS TO VMs USING LOA**

*Require: Parameters P,*
*Ensure: Provisioned tasks to VMs inside clouds*

1. Get Parameters P, from the user through the GUI
2. Get or generate dataset
3. Create multi-cloud environment by creating clouds, data centers and VMs.
4. Call LOA function based on parameters.
5. The output of LOA function, i.e., best positions of lions is mapped to VMs.
6. Cloud Simulation starts
7. Provisioning of resources to tasks is done by broker
8. Cloud Simulation stops
9. Display results of Provisioned tasks

---

**CLOUD PARAMETER VALUES**

<table>
<thead>
<tr>
<th>Cloud Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of tasks</td>
<td>10-100</td>
</tr>
<tr>
<td>No. of Data Centers</td>
<td>5-25</td>
</tr>
<tr>
<td>VM Size</td>
<td>5000-25000 (MB)</td>
</tr>
<tr>
<td>VM RAM</td>
<td>256-512 (MB)</td>
</tr>
<tr>
<td>VM MIPS</td>
<td>150-500</td>
</tr>
<tr>
<td>VM Bandwidth</td>
<td>500-5000 (MB/sec)</td>
</tr>
<tr>
<td>No. of CPUs</td>
<td>1-4</td>
</tr>
<tr>
<td>File Size</td>
<td>100-500 (MB)</td>
</tr>
<tr>
<td>Output File Size</td>
<td>200-500 (MB)</td>
</tr>
</tbody>
</table>
LION OPTIMIZATION ALGORITHM

- Generate Population
- Hunting
- Move to Safe Place
- Roaming
- Mating
- Defence
- Migration

LOA PARAMETER VALUES

<table>
<thead>
<tr>
<th>LOA Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>npop</td>
<td>50</td>
</tr>
<tr>
<td>prideNo</td>
<td>4</td>
</tr>
<tr>
<td>percentNomad</td>
<td>0.2</td>
</tr>
<tr>
<td>roamingPercent</td>
<td>0.2</td>
</tr>
<tr>
<td>mutateProb</td>
<td>0.2</td>
</tr>
<tr>
<td>sexRate</td>
<td>0.8</td>
</tr>
<tr>
<td>mateProb</td>
<td>0.3</td>
</tr>
<tr>
<td>migrateRate</td>
<td>0.4</td>
</tr>
</tbody>
</table>
MULTI-OBJECTIVE OPTIMIZATION

• Makespan

\[ \text{makespan} = \max (\text{endTime} - \text{startTime})_{\text{task}_i} \]

• Average Response Time (ART)

\[ \text{ART} = \frac{\sum (\text{endTime} - \text{startTime})_{\text{task}_i}}{n} \]
where \( n \) is the total no. of tasks

• Cost

\[ \text{cost} = \sum (C_i \times T_i)_{\text{resource}_i} \]
where '\( C_i \)' is the cost of resource \( i \)
'\( T_i \)' is the total time taken by resource \( i \)

• Completion Time (CT)

\[ \text{CT} = \sum (\text{endTime} - \text{startTime})_{\text{task}_i} \]

• Average Resource Utilization (ARU)

\[ \text{ARU} = \frac{\sum (\text{finishTime} - \text{startTime})_{\text{task}_i}}{\text{makespan} \times m} \]
where \( m \) is the total no. of tasks
RESULTS AND DISCUSSION

For the given parameters, No. of tasks = 50, No. of Data centers = 15, VM size = 15000 MB, VM Ram = 512 MB, VM MIPS = 400, File Size = 300 MB, VM BW = 4000 MB/sec, Output File size = 400 MB, No. of CPUs = 2
RESULTS AND DISCUSSION

Performance Comparison of LOA over PSO

<table>
<thead>
<tr>
<th>Objectives</th>
<th>PSO</th>
<th>LOA</th>
<th>PC(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT (ms)</td>
<td>18564</td>
<td>4117</td>
<td>77.82</td>
</tr>
<tr>
<td>Makespan (ms)</td>
<td>2070</td>
<td>1192</td>
<td>42.41</td>
</tr>
<tr>
<td>Cost (Rs)</td>
<td>323</td>
<td>74</td>
<td>77.089</td>
</tr>
<tr>
<td>ART (ms)</td>
<td>1294</td>
<td>745</td>
<td>42.42</td>
</tr>
<tr>
<td>ARU (%)</td>
<td>6.25</td>
<td>2.08</td>
<td>66.72</td>
</tr>
</tbody>
</table>

\[
PC\% = \frac{(Value_{LOA} - Value_{PSO})}{Value_{PSO}} \times 100
\]
CONCLUSION AND FUTURE SCOPE

• This paper addresses performance, efficiency and reliability issues of resource provisioning.

• The proposed method using LOA outperforms dynamic resource allocation for all considered objectives.

• The result of the proposed method shows better percentage of improvement as compared with the traditional PSO algorithm.

• In future, we have planned to consider the performance of the same parameters used in LOA and simulate them in the federated cloud environment.

• There will be a study on how to incorporate Live VM migration as an effective mechanism to balance the load.
REFERENCES:


THANK YOU!
Q/A