Heuristic-based Task Selection and Allocation Framework in Dynamic Collaborative Cloud Service Platform

Biao Song, Mohammad Mehedi Hassan, Eui-Nam Huh

CloudCom 2010

Presented By:
Prof. Eui-Nam Huh

Dep. of Computer Eng.
Kyung Hee University
Agenda

- Background of Dynamic Cloud Collaboration
- Task Selection and Allocation Problem
- New Metric to Improve Resource Utilization
- Heuristic Algorithm
- Simulation Result
- Conclusion
Agenda

- Background of Dynamic Cloud Collaboration
- Task Selection and Allocation Problem
- New Metric to Improve Resource Utilization
- Heuristic Algorithm
- Simulation Result
- Conclusion
Dynamic Collaborative Cloud Platform

- To address scalability & interoperability, Dynamic collaboration among Cloud providers is a key issue
- Consumers want to simultaneously use multiple cloud services
- Cloud providers have specialization in their resource and service supplement
- Dynamic Collaborative Cloud platform facilitates offering collaborative or portable cloud services to consumers
Dynamic Collaborative Cloud Market
Cloud Manager

Overall Architecture of Cloud Manager
Workflow of Dynamic Collaboration

Flowchart of dynamic collaboration

1. Requirements
2. Local task partition and allocation
3. Partner Selection
4. Negotiation
5. Bidding
6. Winning Auction
7. Service Supplement
Task Selection and Allocation Problem

- Cloud providers (CPs) are motivated by charging consumers for using their resources and services

- **Key Issue**
  - Improving resource utilization

- As CPs adopt virtualization technology, the task allocation problem becomes more challenging

- In collaborative environment, CPs also face task selection problem
Agenda

- Background of Dynamic Cloud Collaboration
- Task Selection and Allocation Problem
- New Metric to Improve Resource Utilization
- Heuristic Algorithm
- Simulation Result
- Conclusion
**Low Resource Utilization Scenario**

### Table.1 Resource of Primary Cloud Provider

<table>
<thead>
<tr>
<th>Physical Machine</th>
<th>CPU</th>
<th>Memory</th>
<th>Hard Disk</th>
<th>Network Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1000</td>
<td>800</td>
<td>700</td>
<td>500</td>
</tr>
<tr>
<td>P2</td>
<td>900</td>
<td>700</td>
<td>650</td>
<td>500</td>
</tr>
<tr>
<td>P3</td>
<td>800</td>
<td>600</td>
<td>600</td>
<td>500</td>
</tr>
</tbody>
</table>

### Table.2 Resource Requirement of Combinatorial Tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>CPU</th>
<th>Memory</th>
<th>Hard Disk</th>
<th>Network Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>450</td>
<td>400</td>
<td>400</td>
<td>200</td>
</tr>
<tr>
<td>T2</td>
<td>400</td>
<td>200</td>
<td>400</td>
<td>200</td>
</tr>
<tr>
<td>T3</td>
<td>100</td>
<td>300</td>
<td>500</td>
<td>400</td>
</tr>
<tr>
<td>T4</td>
<td>200</td>
<td>200</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>T5</td>
<td>350</td>
<td>500</td>
<td>100</td>
<td>500</td>
</tr>
<tr>
<td>T6</td>
<td>250</td>
<td>450</td>
<td>450</td>
<td>250</td>
</tr>
<tr>
<td>T7</td>
<td>500</td>
<td>200</td>
<td>250</td>
<td>300</td>
</tr>
<tr>
<td>T8</td>
<td>400</td>
<td>300</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>T9</td>
<td>200</td>
<td>200</td>
<td>500</td>
<td>150</td>
</tr>
<tr>
<td>T10</td>
<td>350</td>
<td>300</td>
<td>200</td>
<td>450</td>
</tr>
</tbody>
</table>

### Table.3 Allocation Process (using CPU-based Min−Min)

| Allocated Task | Physical Machine | Remaining Resource on Physical Machine | | |
|----------------|------------------|----------------------------------------|---|---|---|---|
| T3             | P1               | CPU 900                                | Memory 500 | Hard Disk 200 | Bandwidth 100 | Low Utilization |
| T4             | P2               | CPU 700                                | Memory 500 | Hard Disk 350 | Bandwidth 200 | Low Utilization |
| T9             | P3               | CPU 500                                | Memory 200 | Hard Disk 0   | Bandwidth 0   | Full Utilization |
| T8             | P1               | CPU 500                                | Memory 200 | Hard Disk 100 | Bandwidth 0   | Full Utilization |
| T10            | P3               | CPU 250                                | Memory 100 | Hard Disk 0   | Bandwidth 0   | Full Utilization |
Key Point: Resource BalancingTask

- In each physical machine, CPU, memory, hard disk and network bandwidth resources should be uniformly utilized
- It helps to improve the overall resource utilization for Cloud providers
- It can be achieved by using new metric when Cloud providers select and allocate tasks
Agenda

- Background of Dynamic Cloud Collaboration
- Task Selection and Allocation Problem
- New Metric to Improve Resource Utilization
- Heuristic Algorithm
- Simulation Result
- Conclusion
New Metric to Improve Resource Utilization

- Standard deviation is a simple but quite effective metric to tackle the resource balancing issue:

  Averaging Utilization:
  \[ au = \frac{(fc + fm + fh + fb)}{4} \]

  Balancing Condition:
  \[ bc = \sqrt{(au - fc)^2 + (au - fm)^2 + (au - fh)^2 + (au - fb)^2} \]

  \( fc, fm, fh \) and \( fb \) denote the utilization percentage of CPU, memory, hard-disk and network bandwidth respectively.
Tradeoff-metric

- Existing metrics also have many advantages, an existing metric could be:
  - Suppose a task $t_i$ allocated on $p_j$ can get
    \[ c_{ij} + fc_j \times (c_{ij} / (1-fc_j)) = c_{ij} / (1-fc_j) \]
    CPU capacity from $p_j$, where $c_{ij}$ is the minimum CPU capacity required for running $t_i$ on $p_j$
  - For $t_i$, the superiority of $p_j$ is defined as
    \[ S_{ij} = c_{ij} / (c_{ij} / (1-fc_j)) = 1-fc_j \]
- Combine both metrics using trade-off method
  \[ T(i, j) = \varphi_{ij} \times S_{ij} + (1- \varphi_{ij}) \times bc_{ij} \]
  - $\varphi_{ij}$ is related to the condition of remaining resources
Agenda

- Background of Dynamic Cloud Collaboration
- Task Selection and Allocation Problem
- New Metric to Improve Resource Utilization
- Heuristic Algorithm
- Simulation Result
- Conclusion
Heuristic Algorithms

- The trade-off metric can be directly applicable for many existing static or dynamic heuristic allocation algorithm, such as Min-Min, Max-Min, UDA or Sufferage.
- However, low resource utilization problem has not been fully solved since the task selection issue need to be addressed.
Threshold Method

- By setting proper threshold value for *Balancing* metric, we can prevent improper task selection from happening.
- Define $\xi$ as the scalable threshold value, any single allocation which causes $bc_{ij} > \xi$ is not acceptable.
Choosing Threshold Value

- In static environment, we assume the resource requirements of coming tasks are predictable.
  - By adopting different threshold value and iteratively run the heuristic algorithms, we can find the optimal value to maximize the overall resource utilization.

- In dynamic environment, the threshold value can be selected in a reactive way.

---

**Fig. 4(a) The peak locates in the side part**

**Fig. 4(b) The peak locates in the middle part**

\[ U \]

\[ \alpha \quad \beta \]

**Low Boundary** **Midpoint** **High Boundary**

**New Region** **New Region**

Internet Computing & Security Lab (ICNS Lab)
Agenda

- Background of Dynamic Cloud Collaboration
- Task Selection and Allocation Problem
- New Metric to Improve Resource Utilization
- Heuristic Algorithm
- Simulation Result
- Conclusion
Impact of Proposed Approach

Table.3 Allocation Process (using CPU-based Min-Min)

<table>
<thead>
<tr>
<th>Allocated Task</th>
<th>Physical Machine</th>
<th>Remaining Resource on Physical Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CPU</td>
</tr>
<tr>
<td>T3</td>
<td>P1</td>
<td>900</td>
</tr>
<tr>
<td>T4</td>
<td>P2</td>
<td>700</td>
</tr>
<tr>
<td>T9</td>
<td>P3</td>
<td>600</td>
</tr>
<tr>
<td>T8</td>
<td>P1</td>
<td>500</td>
</tr>
<tr>
<td>T10</td>
<td>P3</td>
<td>250</td>
</tr>
</tbody>
</table>

Table.4 Allocation Process (using proposed approach)

<table>
<thead>
<tr>
<th>Allocated Task</th>
<th>Physical Machine</th>
<th>Remaining Resource on Physical Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CPU</td>
</tr>
<tr>
<td>T4</td>
<td>P1</td>
<td>800</td>
</tr>
<tr>
<td>T2</td>
<td>P3</td>
<td>400</td>
</tr>
<tr>
<td>T7</td>
<td>P2</td>
<td>300</td>
</tr>
<tr>
<td>T1</td>
<td>P1</td>
<td>350</td>
</tr>
<tr>
<td>T8</td>
<td>P3</td>
<td>0</td>
</tr>
</tbody>
</table>
Simulation Results

Overall Resource Utilization
Agenda

- Background of Dynamic Cloud Collaboration
- Task Selection and Allocation Problem
- New Metric to Improve Resource Utilization
- Heuristic Algorithm
- Simulation Result
- Conclusion
Conclusion

- To make dynamic collaborative Cloud service platform truly applicable, we propose a general heuristic-based task selection and allocation framework.

- Our main target is to improve the overall resource utilization for Cloud providers.

- Considering the specialty of Cloud environment, we address the balancing issue by developing a new metric and apply it into existing heuristic-based allocation algorithms.

- By adding scalable threshold value, task selection is enabled to enhance heuristics.

- Simulation results prove that our proposed approach achieves better resource utilization.
Thank you

Email: bsong@khu.ac.kr