The Effects of Hotspot Detection and Virtual Machine Migration Policies on Energy Consumption and Service Levels in the Cloud

S. Sohrabi and I. Moser 1
Swinburne University of Technology, Melbourne, Victoria, Australia

ssohrabi,imoser@swin.edu.au

Abstract
Managing Cloud resources efficiently necessitates effective policies that assign applications to hardware in a way that they require the least resources possible. Applications are first assigned to virtual machines which are subsequently placed on the most appropriate server host. If a server becomes overloaded, some of its virtual machines are reassigned. This process requires a hotspot detection mechanism in combination with techniques that select the virtual machine(s) to migrate.

In this work we introduce two new virtual machine selection policies, Median Migration Time and Maximum Utilisation, and show that they outperform existing approaches on the criteria of minimising energy consumption, service level agreement violations and the number of migrations when combined with different hotspot detection mechanisms. We show that parametrising the the hotspot detection policies correctly has a significant influence on the workload balance of the system.

Keywords: Cloud, Scheduling, IaaS, Virtual Machine Migration, Hotspot Detection, CloudSim

1 Introduction

In the Cloud, the Virtual Machine Monitor (VMM) arbitrates the access to the real physical resources so that different operating systems in VMs can share the host infrastructure. Virtualisation first virtualises a task by assigning it to a VM (Workload Isolation) then allows several VMs to share a physical platform (Workload Consolidation) and if necessary, moves it to another server to balance the load (Workload Migration) [4].

Migration is generally seen as a two-step process consisting of the detection of an overloaded host and the choice of VM to migrate. Beloglazov and Buyya [1 2] have introduced policies that achieve both steps.
2 Previous Work

2.1 Overloaded Host Detection Policies

Beloglazov and Buyya first developed an adaptive threshold for determining an overloaded host in the system [1] and later developed and extended these policies [2].

1. THReshhold (THR) is an approach that sets upper and lower utilisation thresholds for hosts based on experience.

2. Median Absolute Deviation (MAD) specifies a lower threshold empirically, while the upper threshold is calculated using the median of the absolute deviation from the medians of the CPU usage data sets.

3. InterQuartile Range (IQR) is another approach to determine the upper threshold, while the lower threshold is determined empirically as before.

4. Local Regression (LR) is an approach that fits a curve that shows the trend in the data. A host is overloaded if the maximum migration time is closer than a safety margin to the trend line.

5. Robust Local Regression (RLR) compares the maximum migration time to an expected value and weights it before deciding whether a hotspot exists.

2.2 Selecting VMs for Migration

To select a VM for migration, Beloglazov and Buyya [2] proposed three alternative policies. Minimum Migration Time (MMT) migrates the VM with the lowest migration time, defined as the amount of RAM utilised by a $VM_i$ divided by the spare network bandwidth available for the given host. Maximum Correlation (MC) is inspired by an insight of Verma et al. [3] that high correlation between tasks and resource usage might lead to server overloading. MC uses the multiple correlation coefficient which corresponds to the squared correlation between the predicted and the actual values of the dependent variable.

3 Improved VM selection policies

The best VM selection policy in Beloglazov and Buyya’s work [2] was shown to be MMT which selects the VM with the least migration time. This may fail to resolve the hotspot and necessitate several migrations. The process of migration increases the task completion time by a minute amount due to the downtime it creates in the source and destination hosts. VM migration exacerbates energy consumption because of the load transfer through the network. Resolving the hotspot in fewer migrations often leads to a decrease in SLA violation. Therefore, this study proposes VM selection policies that resolve hotspots spending fewer migrations, resulting in lower energy consumption and SLA violation percentage.

Median Migration Time (MedianMT): The migration times of all candidate VMs on a hotspot are calculated and the VM with median migration time is migrated. Ties are resolved randomly.
Figure 1: Energy consumption caused by the three different migration policies MMT, MedianMT and MaxU in combination with the different hotspot detection policies.

Maximum Utilisation (MaxU): Choosing the VM to migrate from the hotspot based on the largest possible CPU usage can be expected to minimise the number of migrations.

4 Experimental Setup

The VM selection policies are compared with the best policy proposed by Beloglazov and Buyya [2], Minimum Migration Time. Each competing migration policy was combined with each of the five methodologies for hotspot detection in turn. Each of the five methodologies was used with a set of different variables as follows: THR \( u = [0.6, 0.7, 0.8, 0.9, 1] \), IQR \( s = [0.5, 1, 1.5, 2, 2.5, 3] \), MAD \( s = [0.5, 1, 1.5, 2, 2.5, 3] \), LR \( s = [1, 1.1, 1.2, 1.3, 1.4] \), RLR \( s = [1, 1.1, 1.2, 1.3, 1.4] \). The ten data sets provided with CloudSim were optimised to produce the results shown below. We obtained \( N = 270 \) data values by combining each of the parametric value of the hotspot detection policies with each of the 10 data sets.

5 Results

Following the approach by Beloglazov and Buyya [2], the averaged results over all hotspot detection mechanisms for each of the migration policies are given in Table [1]
Table 1: Descriptive statistics for energy consumption, number of VM migrations and SLA violation for the chosen migration policies.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Policy</th>
<th>N</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>MMT</td>
<td>270</td>
<td>180</td>
<td>34</td>
<td>108</td>
<td>301</td>
</tr>
<tr>
<td></td>
<td>MedianMT</td>
<td>270</td>
<td>171</td>
<td>34</td>
<td>98</td>
<td>296</td>
</tr>
<tr>
<td></td>
<td>MaxU</td>
<td>270</td>
<td>149</td>
<td>32</td>
<td>89</td>
<td>269</td>
</tr>
<tr>
<td>Migrations</td>
<td>MMT</td>
<td>270</td>
<td>31001</td>
<td>8072</td>
<td>19062</td>
<td>68540</td>
</tr>
<tr>
<td></td>
<td>MedianMT</td>
<td>270</td>
<td>27624</td>
<td>6577</td>
<td>17648</td>
<td>56123</td>
</tr>
<tr>
<td></td>
<td>MaxU</td>
<td>270</td>
<td>19133</td>
<td>4360</td>
<td>7972</td>
<td>32029</td>
</tr>
<tr>
<td>SLA violation</td>
<td>MMT</td>
<td>270</td>
<td>10.2</td>
<td>0.5</td>
<td>8.6</td>
<td>12.2</td>
</tr>
<tr>
<td></td>
<td>MedianMT</td>
<td>270</td>
<td>10.2</td>
<td>0.5</td>
<td>9.4</td>
<td>12.3</td>
</tr>
<tr>
<td></td>
<td>MaxU</td>
<td>270</td>
<td>9.9</td>
<td>0.6</td>
<td>8.4</td>
<td>12.9</td>
</tr>
</tbody>
</table>

Figure 2: Number of VM migrations caused by the three different migration policies MMT, MedianMT and MaxU in combination with the different hotspot detection policies.

Figure 1 confirms the superiority of MaxU and the fact that MedianMT performs better than MMT but worse than MaxU when minimising energy consumption which was suggested by Table 1. The graphs in Figure 1 seem to show that the performance of a hotspot detection policy is not affected by the choice of migration policy used. It is clear that the parameterisation of the policies has a more significant effect on the outcome than the choice of policy itself. Given the large spread of the box-and-whisker plots it is conceivable that the optimal parameters might be situation-dependent.

Figure 2 illustrates the numbers of VM migrations caused by these policies. In this case, we
Sohrabi and Moser

(a) THR

(b) IQR

(c) MAD

(d) LR

(e) RLR

Figure 3: SLA violation percentage caused by the three different migration policies MMT, MedianMT and MaxU in combination with the different hotspot detection policies.

can see that MaxU outperforms the other policies by a larger margin when paired with MAD, LR or RLR. The parameterisation of the algorithms does not have as large an effect. The fact that LR and RLR combined with MaxU perform very well suggests that predicting a hotspot and solving it by removing the 'biggest problem' immediately is a very effective strategy which pre-empts overuse. On SLAV (Figure 3), MaxU performs best, causing the smallest median number of SLAV regardless of hotspot detection policy and its parameterisation.

References


