Prediction technique for resource allocation in Micro Data Center

Cong-Thinh Huynh  
Department of Computer Engineering  
Kyung Hee University  
Yongin, South Korea  
Email: thinhhc@gmail.com

Eui-Nam Huh  
Department of Computer Engineering  
Kyung Hee University  
Yongin, South Korea  
Email: johnhuh@khu.ac.kr

Abstract—Cloud Computing (CC) represents an attractive and cost-efficient of server-based computing and application service provider models. Virtualization technology enables dynamically allocate resources based on workload fluctuations from users’ needs. The current paradigm of Data Center (DC) or Mega Data Center (MegaDC) has changed to new approach where Micro Data Centers (MDC) are distributed to different locations, Cloud services and solutions are deployed onto those MDCs in order to provide services to enterprise, home or mobile users. Therefore, by identifying future resources needed before scaling up or down the MDC’s resources pool is one of the most challenge research. In this paper, we propose the resources prediction technique based on historical data at MDC level for allocating the appropriate resources. The experiment results show that by using this technique we can predict the future resource needed for efficient resource provisioning in Cloud Computing environment.

I. INTRODUCTION

Recently, Cloud Computing is a computing paradigm where virtual machines (VM) are simulated by software running on one or many real physical machines [2]. In this paradigm, Cloud Service Providers can be able to deploy their services and solutions onto those VMs e.g. web services, online collaboration, messaging & monitoring, communication, finance, scientific computational and so on. End users use their devices such as laptop, smart phones, tablets, smart watches can access to their scalable resources such as files, data, programs, computational processes from enterprise, home or mobility on the move via the internet connection as seen on Figure 1.

In this paradigm, Micro Data Center (MDC) keeps an important role as intermediate server, where cloud services are deployed on and these MDCs can be placed at many different locations and connected to a Mega Data Center (MegaDC) through internet connection and managed by legacy routers [4]. In CC environment, by allowing Converged Services and Simplified Branches, complex IT infrastructure in enterprise groups will meet some challenges about limiting factors such as floor space, energy consumption, high density zones, datacenter infrastructure management and deployment. Those aforementioned challenges demand new disruptive technology to meet the business objectives in a efficient manner. MDC comprises of standard, high density, cost effective and energy efficient in a single rack size area, is a good solution in this scenario. MDC comes with onboard cooling, security and fire prevention, it can accommodate servers, networking storage equipment and connects through Wi-Fi, WiMax, 4G wireless and satellite for communication. Especially, MDCs are mobile and can be moved with the equipment mounted in a variety of indoor/outdoor conditions where only power and internet connection is available. Theses MDCs are connected to each others and connected to MegaDC. In which, each MDC may provide one or many services and data from these services can be transfered, migrated within network to perform multi-service needs and stored into main data center at MegaDC.

The problem is when a peak load occurs, current number of VMs on these MDCs cannot handle fluctuation workloads, the need of identifying the future workloads needed to allocate resources provisioning is important. To address this challenge, we propose the resources prediction technique using seasonality forecast and time analysis based on historical data [1]. The rest of the paper is organized as follows: Section 2 presents the problem statement and paradigm. The prediction model is described in section 3. Experiment results are described and discussed in section 4. We draw a conclusion and future work in section 5.

II. PROBLEM STATEMENT

In virtualization environment in Cloud Computing, more particularly at MDC level in this scenario, cloud services instances are deployed into one or many virtual machines, based on hypervisor from physical server resources capacity.
VMs have to handle all the requests from users also share the resource pools such as RAM memory and CPUs. One of the most attractive features of CC for Cloud Service Providers is the ability to access computing resource elastically (by scaling up or down) according to dynamic resource demands. Users use their devices are continuously sending requests to cloud services, which are monitored by an Automated Scaling Listener (ASL). After receiving those requests, ASL will send away the request information such as the number of requests, request rates, Service Level Agreement (SLA) to Intelligent Automation Engine (IAE). An IAE script will be generated using the workflow logic, depends on how much requests are coming. The script runs the workflow logic that notifies the hypervisor to allocate more resources from the resource pools, share additional CPUs and RAM memory to the virtual server. The paradigm of this approach is represented in Figure 2.

Fig. 2. Intelligent Automation Engine allocating resource pool based on workflow logic

In this architecture, Intelligent Automation Engine has to produce the desired future resource results to resource pools in order to interact with the hypervisor and return the appropriate number of CPU and RAM resources at runtime. Whole the process has to ensure appropriate resources via dynamic allocation before thresholds at VMs are met. On the other hand, launching a VM instance takes several tens of seconds to minutes, that is why the predictive-driven approach is proposed in order to quickly adapt the new coming requests or release the resource pools to save cost or energy. We propose this prediction model at Intelligent Automation Engine level so that our systems can predict the future requests based on historical data which we retrieve from Automated scaling listeners.

III. PREDICTION MODEL

The main approaches of this prediction model are based on user request as time series and their seasonality/periodical behavior characteristics. The fact is that, from the human’s point of view, we usually use devices to connect to internet at enterprise for working, at home for entertainment, on the move for relax or so on, the number of requests gain the highest volume at this time while the traffic is decreased in the night. Besides, this number will be much higher on working days than on weekends, we also achieve this trend on holidays compare with others.

We deploy and collect the request analytics from a website on virtual hosting in first 4 days based on time series, the graph is depicted in the Figure 3. In which, most time series patterns, every 4 blocks in this scenario, can be described in terms of two basic classes of components: trending pattern and seasonality. Trending pattern represents a general systematic linear or nonlinear component that changes over time and does not repeat or at least does not repeat within the time range captured by our data. Whereas, the seasonality may have a formally similar trend, same fluctuation trends and it repeats itself in systematic intervals over time. We can formulate this problem as following approaches.

1) User request as time series: The requests from users are denoted as a time series: \( \{X(t); t \in T\} \), where \( T \) is an index of the time fragment, and \( X(t) \) is the random variable, which represents the total number of requests that arrive into the system in the \( t \) time fragment. Let the set \( (X(t), X(t−1), ..., X(t−k+1)) \) is the current and last \( k−length \) observed request values which we get from Automated Scaling Listener, we will predict the future value \( X(t+n) \) for the next \( n−length \) fragment of time.

2) Seasonality or Periodical behavior: From above analysis about seasonality of requests during observed time. The actual number of requests to the servers, denoted by \( N_{actual} \), has seasonality characteristics. The same correlation pattern when plotted, by deseasonalizing this data using Linear Regression approach [3], we can identify the moving average and predict the future value, denoted by \( N_{forecast} \).

3) Evaluation of this prediction model: Given \( N_{actual} \), \( N_{forecast} \), \( n \) and \( MAD \) is the actual number of requests to the servers, future forecasted requests, number of forecast errors and Mean Absolute Deviation, respectively. We have:

\[
MAD = \frac{\sum |N_{actual} - N_{forecast}|}{n} = \frac{\sum Errors}{n} \tag{1}
\]

Depends of the SLA, we can identify this \( MAD \) value in order to get higher value compare to actual value to ensure that workflow logic determines enough resources to handle all the future requests from users.

4) Method: This prediction model is represented as below steps:
   i. Determine the seasonal period.
   ii. Develop a Moving Average forecast, smooth the centered
Moving Average (MA).

iii. Find the ratio of each observation to the MA forecast:

\[ r_t = \frac{Y_t}{S_t} = \frac{observed\_value}{MA} \]  

(2)

iv. Find the average of the ratio for each season or periodic unit.

v. Divide each of the ratio by the average of the ratio. This will set the average index to 1, which will identify the adjusted season indexes.

vi. For each observation, we divide the observation by its adjusted seasonal index:

\[ S'_t = \frac{Y_t}{season\_indexes\_of\_t} \]  

(3)

IV. EXPERIMENT RESULTS AND DISCUSSION

We deploy and collect the request analytics from a website on virtual hosting in first 4 days as seen on Figure 3. The Intelligent Automation Engine actively monitors the traffic every minutes and makes the decision to resource pool every 6-hour through workflow logic. Because these requests have seasonality characteristics, we apply our prediction model to first 4 days data to predict the value of requests on the fifth day.

<table>
<thead>
<tr>
<th>Day</th>
<th>Seasonal</th>
<th>Requests (x100,000)</th>
<th>De-seasonalize</th>
<th>Predict</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>4.8</td>
<td>4.57</td>
<td>4.89</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4.1</td>
<td>4.89</td>
<td>4.52</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6.0</td>
<td>5.49</td>
<td>6.06</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>6.5</td>
<td>5.69</td>
<td>6.50</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>5.8</td>
<td>6.22</td>
<td>5.44</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>5.2</td>
<td>6.21</td>
<td>5.01</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>6.8</td>
<td>6.22</td>
<td>6.70</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>7.4</td>
<td>6.47</td>
<td>7.18</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>6.0</td>
<td>6.44</td>
<td>5.99</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>5.6</td>
<td>6.68</td>
<td>5.50</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>7.5</td>
<td>6.86</td>
<td>7.35</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>7.8</td>
<td>6.82</td>
<td>7.85</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>6.3</td>
<td>6.76</td>
<td>6.54</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>5.9</td>
<td>7.04</td>
<td>6.00</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>8.0</td>
<td>7.32</td>
<td>7.99</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>8.4</td>
<td>7.35</td>
<td>8.52</td>
</tr>
<tr>
<td>17</td>
<td>5</td>
<td></td>
<td></td>
<td>7.09</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td></td>
<td></td>
<td>6.49</td>
</tr>
<tr>
<td>19</td>
<td>3</td>
<td></td>
<td></td>
<td>8.63</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td></td>
<td></td>
<td>9.19</td>
</tr>
</tbody>
</table>

Fig. 4. Actual and predicted values analysis

Experiment in Figure 3 shows that our prediction model can predict the trend and the future request from historical data in which the blue dots are actual values and orange dots are predicted values.

V. CONCLUSION

In this paper, we have proposed prediction based model in Cloud Computing to dynamically scale resources in response. By using this technique, we can predict the future usage of resources on every MDC, based on retrieved results from workflow logic, our system can automatically supply new resources from resources pool or release them based on users workloads. Hence, this technique shows the cost effective and energy efficient in MDC architecture. This model can be designed so that the Intelligent Automation Engine can directly send its scaling request via the VIM instead of to the hypervisor. We can apply some other Cloud components such as: Cloud Usage Monitor to collect resource usage information on IT resources before, during, and after scaling, to help define the future processing capacity thresholds of the virtual servers. Pay-Per-Use Monitor to collect resource usage cost information as it fluctuates with the elastic provisioning and Resource Replication to generate new instances of the scaled IT resources, which will be a topic for future investigations.

ACKNOWLEDGMENT

This research was supported by the MSIP (Ministry of Science, ICT&Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2014(H0301-14-1020)) supervised by the NIPA (National IT Industry Promotion Agency). This research was also supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No.NRF-2013R1A1A2013620). Professor Eui-Nam Huh is corresponding author.

REFERENCES


