Resource Allocation for Service Composition in Cloud-based Video Surveillance Platform

M. Shamim Hossain, Mohammad Mehedi Hassan, M. Al Qurishi and Abdullah Alghamdi
College of Computer and Information Sciences
King Saud University, Riyadh, Saudi Arabia
Email: {mshossain, mmhassan, qurishi, ghamdi}@ksu.edu.sa

Abstract—Resource allocation play an important role in service composition for cloud-based video surveillance platform. In this platform, the utilization of computational resources is managed through accessing various services from Virtual Machine (VM) resources. A single service accessed from VMs running inside such a cloud platform may not cater the application demands of all surveillance users. Services require to be modeled as a value added composite service. In order to provide such a composite service to the customer, VM resources need to be utilized optimally so that QoS requirements is fulfilled. In order to optimize the VM resource allocation, we have used linear programming approach as well as heuristics. The simulation results show that our approach outperforms the existing VM allocation schemes in a cloud-based video surveillance environment, in terms of cost and response time.

Keywords—Cloud-based video surveillance, service composition, VM resource allocation and QoS

I. INTRODUCTION

In recent years, cloud-based video surveillance, also known as Video Surveillance as a Service (VSaaS) is emerging as a noteworthy technology [1] to be accessed from anywhere at any time. It can provide a flexible stack of powerful VM resources of cloud like CPU, memory, GPU (graphics processing unit), storage and network bandwidth on demand to manage traditional video surveillance services (VSS) and applications at lower cost. VSaaS allows media streams from cameras or other video sensors to be streamed to the cloud. In order to stream down the desired surveillance stream to the client of the respective subscribed users, different media processing services (e.g. transcoding, delivery, streaming, analysis, sharing, and payment) are used.

However, a single service offered from an application service executed inside a cloud platform may not cater the application demands of all users. In order to fulfill such demands, a composite media service is used rather than using a single service. A composite media service can be understood to combine or compose a set of media services (e.g. transcoding, delivery, streaming, sharing), which are provided by different application service providers executed in a cloud platform. One of examples of media services in VSaaS is Transcoding as a Services (Taas) provided by a application service provider is Encoding.com. It uses the Amazon Web Services (AWS) platform for transcoding saved surveillance videos into different formats in terms of resolution, frame rate, and bandwidth. Some of the applications, where VSaaS solutions are currently being used include airports, railway station, private homes, retail store, oil rigs and electricity substation [2].

One of these challenges is related to composition of different media processing services in VSaaS within the cloud, as a single service is often inadequate in its functionality to satisfy heterogeneous application requirements of emergency officials’ devices in terms of media processing and delivery. The composition media service in VSaaS is essential to provide customized video to the emergency officials’ handheld devices for quick decisions related to public safety concern in airports and other public places, hostage situation, fire-emergency and so on [3].

Recently, VaaS has been observed and studied mostly in industry [4], [5], [6], [7] and a few in academy [8], where utilization computational resources (e.g. storage, processing, delivery, application level QoS) is handled through services accessed from virtual machine (VM) resources (e.g. Hardware, software, and media application services). One essential issue here is to meet the QoS requirements of composite VSaaS services that require a different quantity of VM resources at run-time [9]. Inappropriate VM resource allocation in this environment may result in resource waste and QoS degradation. Therefore, there is a need to develop a cost effective and dynamic VM resource allocation model to meet the QoS requirements in the cloud-based video surveillance environment. The proposed approach is evaluated through simulation and implementation.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 describes proposed VM resource allocation model. Section 4 presents simulation results and performance comparisons. Finally, Section 5 concludes the paper.

II. RELATED WORK

Service composition in cloud-based video surveillance platform is becoming popular to augment its functionality. Most of the existing works [10], [11], [1] in this regard...
mainly focus on different methods of selecting the optimal service composition paths in cloud platform. For example, Qi et al presented a cloud-based multimedia service composition scenario in [10]. They studied a QoS-aware composition method for supporting cross-platform service invocation in cloud environment. Ye et al [11] investigated a genetic algorithm based approach for service compositions in cloud computing. However, to the best of our knowledge, only a few works consider an effective VM resource allocation model for composition of video surveillance services in cloud computing platform.

There are few researches [12], [13], [14] regarding VM resource allocation for media services in a multimedia cloud environment. Nan et al [12] proposed a cost effective resource allocation optimization approach for multimedia cloud that was based on a queuing model. Wen et al [13] presented an effective load-balancing algorithm (i.e. using round robin algorithm) for a cloud-based multimedia system, which can allocate and schedule VM resources for different user requests with minimum costs.

However, all these works cannot be directly applicable in VSaaS platform since they do not consider the composite media service that can affect the response time as well as overall utilization of VM resources. Also all these works assume that the pool of VM resources is homogeneous which is not practical. This paper work has been motivated by Ferreto et al [15], even though they did not consider composite media service scenario. The authors in [15] proposed an LP formulation and heuristics for server consolidation with migration control for virtualized data centers. We extend their approaches and consider more constraints such as delay of composite media services, resource utilization and CPU utilization to reduce the response time, improve the overall resource utilization and avoid frequent VM migration respectively for cloud-based video surveillance system.

III. SYSTEM MODEL AND PROPOSED RESOURCE ALLOCATION APPROACH

Fig. 1 depicts the system architecture of a general VSaaS service composition in a cloud platform. In this platform, video streams are delivered to the cloud from video capturing services. The users (e.g. fire fighter, security personnel) can obtain different VSS services from video surveillance service directory such as transcoding service, analysis and sharing service, streaming service, payment service etc. and request for their different compositions through a web browser interface. The users’ composite service requirements are then sent to the cloud system manager, which finds out the suitable configuration of VM resources that are based on SLA. The resource allocation manager then allocates the VM resources to a set of physical machines to run the mobile media service tasks. The mobile media service tasks outputs (i.e. display updates, composition results etc.) are finally transmitted to the user through the web browser. After the media applications or services are started, the system monitoring and metering function tracks the VM resource usage that are attributed to users. It can also notify the resource and system managers for a quick response, and the resource configuration to assure that the correct VM resources are distributed to suitable mobile users. Therefore, in order to correctly allocate resources and to deploy VM images, an efficient, cost effective and optimal VM resource allocation algorithm is necessary for a resource manager.

In the next sections, we will present our proposed VM allocation approaches in detail.

A. Linear Programming Formulation

The proposed VM resource allocation problem is mapped to the multidimensional bin-packing problem [16], which is NP-complete. In this problem, we have to map several items into the smallest number of bins as possible. Here, each item denotes a tuple, which contains its dimensions. In our scenario, we consider each VM as an item and the dimensions like CPU, memory, storage, network bandwidth and GPU, as its capacities. The target is to find a set of physical machine to host the VMs in an optimized way. The basic concern of a VM allocation is that a physical machine must have enough capacity for hosting the VMs. To reduce the hosting cost, the number of active physical machines needs be minimized. To avoid frequent VM migration, certain amount of CPU capacity needs to be preserved as backup resource for handling workload burst. To reduce the response time, the delay of the composite service needs to be controlled. According to above considerations, we design a linear programming (LP) model for quantitatively optimizing VM allocation into physical servers. The input parameters and variables used in the linear programming formulation are presented in Table I.

For any composite video surveillance service $I$ that needs to be allocated in the cloud, the LP model is presented in
The objective function in (1) aims at minimizing the number of required physical servers. The constraint in (2) guarantees that each virtual machine is mapped to a single physical server. Equation (3) guarantees that the virtual machine demands allocated in each physical server do not overload its capacity. The constraint in (4) guarantees that the delay of composite video surveillance service \( I \) does not exceed a certain threshold value \( T \). Equation (5) helps to improve the overall resource utilization. The constraint in (6) can reduce the chance of CPU overload and can potentially balance the CPU utilization among all physical servers.

The delay of composite video surveillance service \( delComp_I \) is defined differently in different scenarios as follows: 1) In the asynchronous composition case, where the VMs have no inter-communication with each other, \( delComp_I \leq T \) can be viewed as the combination of delay constrains provided by all VMs. 2) In the synchronous composition case, where the screen update from \( m \) VMs is synchronized, \( delComp_I \leq T \) means that the most strict delay constraint among all VMs is applied to every VM. 3) In the sequential composition case, where the output of the service running on a predecessor VM is the input of the service running on a successor VM, \( delComp_I \leq T \) denotes that the delay constraint on the predecessor VM is incrementally applied to the successor VM.

### IV. Simulation Results

We study the performance of the proposed LP and heuristic-based solutions through simulations. We compared their performances with two existing algorithms: VM allocation with fractional Knapsack problem (FracKnap) [17], and a round-robin allocation. The results include the performance of cost reduction and response time reduction. Table II shows simulation workload parameters similar to [15], where \( HI, ACU, AMU, APU, \) and \( ANU \) represents heterogeneity index, average CPU utilization(\%), average memory utilization(\%), average GPU utilization(\%), and the average network bandwidth utilization(\%) respectively. \( HI, ACU \) and \( AMU \) were retrieved from the Google workload as described in [15], which are normally used by researchers and students to execute computational experiments.

Table III specifies the delay settings of the workload, where \( IDC, SDC, IDT, \) and \( SDT \) represents the individual delay on atomic video surveillance service, the se-
quential delay on composite video surveillance services, the individual delay on a single server, and the sequential delay on connected servers respectively. Using the above simulation parameters, we randomly generate video surveillance service requests in each workload group. In addition, we assume each group of composite video surveillance service contains 1-5 services, which can be atomic, synchronous or sequential. Initially, the capacities of physical servers were assumed to be identical. In this simulation, the number of physical server was fixed to 100.

### Table II
**DETAILS OF WORKLOAD GROUP**

<table>
<thead>
<tr>
<th>Number of traces</th>
<th>HI</th>
<th>ACU</th>
<th>AMU</th>
<th>AGU</th>
<th>ANU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>264</td>
<td>0.02</td>
<td>6.08%</td>
<td>5.79%</td>
<td>30%</td>
</tr>
<tr>
<td>Group 2</td>
<td>125</td>
<td>0.22</td>
<td>9.53%</td>
<td>19.03%</td>
<td>50%</td>
</tr>
</tbody>
</table>

### Table III
**DETAILS OF WORKLOAD DELAY**

<table>
<thead>
<tr>
<th></th>
<th>IDC (ms)</th>
<th>SDC (ms)</th>
<th>IDT (ms)</th>
<th>SDT (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>24-32</td>
<td>24-35</td>
<td>4-28</td>
<td>6-16</td>
</tr>
<tr>
<td>Group 2</td>
<td>17-43</td>
<td>23-43</td>
<td>4-30</td>
<td>6-15</td>
</tr>
</tbody>
</table>

At first, we did several sets of experiments to measure the cost optimization capability of the proposed allocation methods. The number of video surveillance service requests was fixed to 100. Figs. 2 and 3 show the simulation results derived from the two workload groups at low and heterogeneity environments in terms of the number of active physical servers to host the VMs. From the figs., we have found that the proposed LP and heuristic approaches tend to produce more similar results. As shown in Fig. 2 the performance of FracKnap is almost similar to our approaches (require 35 active physical servers) since the low heterogeneity requests do not overuse any type of resource. The resource utilization threshold we used in our proposed approaches do not provide further optimization in this environment. The round-robin method performs worse than any other solution as it does not provide any optimization method. It randomly chooses physical server for each request.

From Fig. 3, we can see that due to the resource utilization threshold and high heterogeneity environment, our proposed approaches outperform existing algorithms by avoiding overuse of any resource. The performance of Fracknap method does not degrade so much, as allocating a large group of requests is easier than allocating several small groups of requests. As the requests consumes more resources, the result of round-robin algorithm increases to 68 active servers.

In the second set of simulations, we explored the actual response time in different environments. We varied the number of video surveillance service requests from 1-150.

In Figs. 4 and 5, the average delay (response time) of the video surveillance services achieved by each solution are
illustrated. As can be seen from these figs, the proposed LP and heuristic approaches show their superiority in both low and high heterogeneity environment. The reason why the FracKnap and round-robin performs worse than our proposed approaches is because they do not consider the delay optimization during the VM allocation process. Since we clearly define the delay model for both atomic and composite video surveillance services, we are capable to address the dependency issue among the services and the physical machines. Consequently, we can conclude that our delay model is effective in the composite video surveillance service allocation.

We implemented a prototype of composite video surveillance service in Amazon cloud platform. It takes pictures by sensing motion and then sending it to the Amazon cloud where different compositions are possible. Currently, we simply allow composition of transcoding service, editing service and sharing service. The tools we utilized were; Amazon Web Services (AWS), SQL server and Visual Studio 2010. A standard web cam was used as camera sensor.

V. CONCLUSION

In this paper, we presented a resource allocation scheme for composite media stream in a cloud-based video surveillance environment, where archived as well as live surveillance video stream captured from camera are transferred to elastic cloud platform to be used by subscribed emergency officials (fire fighters and security) for enhanced security decisions. We have tested our initial prototype in a limited extent inside Amazon cloud. In future, we will describe our proposed approach of VaaS inside an edge cloud platform as well as in our settings. In addition, we will provide more performance comparisons.

REFERENCES