SSM: Scheduling Security Model for a Cloud Environment

Abdullah Sheikh	Malcolm Munro	David Budgen
PhD Candidate	Emeritus Professor of Software Engineering	Professor of Software Engineering
Department of Computer Science	Department of Computer Science	Department of Computer Science
Durham University	Durham University	Durham University
DH1 3LE	DH1 3LE	DH1 3LE
+44 74430 75784		
abdullah.sheikh@dur.ac.uk	malcolm.munro@durham.ac.uk	david.budgen@durham.ac.uk

ABSTRACT

Scheduling in the cloud is a complex task due to the number and variety of resources available and the volatility of usage-patterns of resources considering that the resource setting is on the service provider. This complexity is compounded further when Security issues and Quality of Service (QoS) are also factored in. The aim of this paper is to describe a model that based on Security (SSM) as a key element that cloud services rely on which affects the performance, cost and time concerns within the security constraints of the cloud service. Definition of the Scheduling Security Model (SSM), and evaluation through worked example that can meet the customer requirements of cost and the quality of service in the required time.

CCS Concepts

Cloud Computing → Scheduling → Security Model

Keywords

Scheduling; Security; Model; Cost; Cloud Computing;

1. INTRODUCTION

Cloud computing gives convenient access to services that provide large resources (VMs) for application, software, data storage and executing tasks [1]. This comes with security issues that may affect the service such as data security, secure transactions, hacking and hijacking, and resource security. This paper defines a Scheduling Security Model (SSM) for a cloud environment that emphasises the security aspect.

Security is an important issue that needs to be considered for Quality of Service [3] and scheduling [16]. In this context, a service is a collection of tasks that are performed using some resource. Applying a certain level of security to the scheduling process is a very important factor that can ensure the execution of the tasks is secure [16]. In the SSM, the level of security will be applied at the task level to make the service itself more secure rather than just applying an overall security level to the service.

In addition, security is a shared responsibility between cloud providers and consumers to ensure that the level of security is at the desired level [11]. Che et al. [6] analysed the security concerns between cloud providers and consumers. For consumers security concerns include data security, user access, and data leakage. On the other hand, cloud providers security concerns include availability, long term system security, how to defend against hacking, data centre security, secure transaction, resources security, and access control and management system.

Consumers need to be aware of security from their side and protect their service from any threats. Cloud providers are able to afford better scalability by running multiple virtual machines on physical machines. They have to defend the service against any security risks from any unauthorised physical access, data security, security software, and resource security. Then any security risks in the virtualisation technology that allows co-occupant virtual machines to make unauthorised access could compromise information assets of consumers [11].

Scheduling is a process applied to minimise wasting limited resources by efficiently allocating them among all resources [9]. Resources or virtual machines (VMs) are assigned to customer for running a service and executing tasks [1]. Scheduling is a very complex operation in cloud computing, and used to allocate resources, improve server utilisation, enhance service performance, and execute tasks [14].

Resource scheduling is all about selecting the most appropriate resources to meet the needs of customers. This is a complex task due to the large number of resources available that need to be managed. This is further compounded when security issues and Quality of Service (QoS) are also factored in to achieve a satisfied level of the cloud service [3] [8].

1.1 Related Works

There are many models in cloud computing focusing in different areas. These areas include cloud service performance [10], service cost [7], and security [5]. However, most cloud security models focus on security of Data as a service (DaaS) [2] and cloud storage [4] and overall security [15]. A few have covered the infrastructure level in some aspects such as performance and scalability [12] but not in security especially resource scheduling. For example, executing tasks over virtual resources need to be secured especially when tasks have priority over each other and need to be running on different virtual resources. This process raises many security issues that need to be controlled such as avoiding processing sensitive data on public resource. So, there is a need to consider the security factor in order to improve resources scheduling within the cloud, and towards achieving an optimised resource scheduling model to meet the desired QoS for both providers and consumers.

2. THE SSM

This section presents the Scheduling Security Model (SSM) components and how the costs for executing a service can be calculated. It defines the SSM model and the scheduling function steps.

2.1 Security Definition

Before identifying the model components there is a need to define what security level means in this context. According to Watson [13] the overall security can be considered to be applied for executing tasks from trusted public resources to highly trusted private resources.

For example, a customer requests a service that includes two tasks. One task is to analyse general data with no security requirement. The other task is to save private data that requires higher security level. So, the system will require two resources one is trusted public resource with basic security feature to execute the first task, and another trusted private resource with security feature such as more secure fire-wall to execute the second task.

2.2 SSM Components

The SSM consists of the following components:

• Tasks Set T: The customer will specify all the tasks that need to be executed, where M is the total number of tasks.

$$T = \{t_1, t_2, t_3, ..., t_M\} \tag{1}$$

• Time cost: The elapsed time cost tm_i will be associated with the tasks to be executed over the allocated resource. The calculated total time cost is:

$$tm = tm_1 + tm_2 + tm_3 + ... + tm_M \tag{2}$$

The total time, tm, should be less than or equal to the customer maximum time limit, e.

$$tm \leq e \tag{3}$$

• Quality of service (QoS): SSM supports quality of service which allows the scheduler to adjust the service scheduling tasks to achieve the required quality of service q within the customer budget. Depending on the customer QoS target, q will be selected from a set of options associated with different costs. QoS can be a value in the range from 0, low quality, and incremented by 0.1 to reach 1.0 which is the highest quality level of service:

$$q \in \Big\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\Big\} \tag{4}$$

• Customer Budget: A customer will submit the overall budget, b for the service.

• Security Level: The security level in this context means that the required level of the security that can be applied to each task. So, the customer will submit the security level required for each task. The security level will be a number between 1 (low security) to 5 (high security). SSM will then map this security level to security weight, hw_i for task t_i.

$$h_i \in \Big\{1, 2, 3, 4, 5\Big\} \tag{5}$$

Then the associated weight for each level of security will be as follows by indexing the values:

$$hw_i = [0.0, 0.25, 0.5, 0.75, 1.0] \tag{6}$$

The SSM will map the associated security weight from the equivalent security level. So that, for example if h_1= 2 then hw_1 = 0.25. This will help to calculate the estimated cost to establish the service and to create the required resources with the security level for each resource (public with low security or private with higher security weight).

• Tasks importance: There are many techniques that can be used to prioritise tasks for execution, but in the SSM there will be three levels of task importance p_i which will help ordering tasks queue. These importance levels will be considered after classifying tasks based on the security levels given by the customer and to be ordered as first come first serve. The customer will submit the importance level required for each task:

$$p_i \in \{1, 2, 3\} \tag{7}$$

This means that tasks with a higher value of p_i = 3 is the most important for the customer. For example, A customer has submitted tasks t_1 with security level h_1 = 1 and importance p_1 = 2, and t_2 with security level h_2 = 1 and importance p_2 = 3. That means t_2 will be executed first then t_1 because t_2 has the highest importance.

2.3 Calculated Components

• Table 1 shows how the SSM will analyse the customer requirements for the service, and how to assign a task or set of tasks t_i to each resource.

• Resources Required for a set of tasks: First the model will categorise the tasks into groups depending on the security level. Then it creates, N, the number of resources required, which will be equal to the number of the groups and each resource will take the same security level of that group and it will mapped to the resources security weight Rw_i.

For example, if the customer submitted tasks (t_1: with h_{w1}=0.25, p_1=2 and t_2: h_{w2} =0.5,p_2=1), q=0.0, and e=60 min, the model will categorise these tasks into two groups with two different security levels. After that the model creates two resources N=2, R_1 takes security weight Rw_1 0.25, and R_2 with security weight Rw_2 0.5 then assign each group to the similar resources with same security weight as shown in Table 1.

Table 1. Example of a Service Required

Security Level(Weight)/Importance	1	2	3	R_i	RC_i
1 (0.00)					
2 (0.25)		t_1		R_1	20
3 (0.50)	t_2			R_2	20
4 (0.75)					
5 (1.00)					

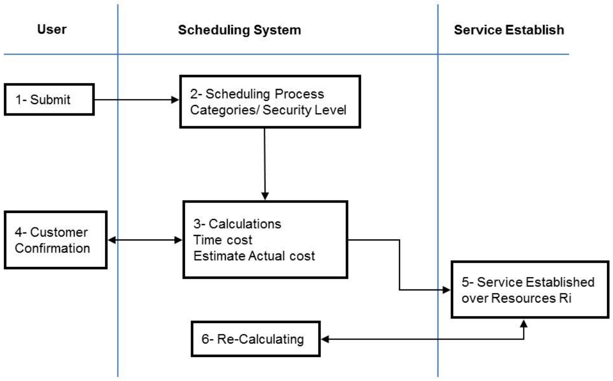

Figure 1. Scheduling Security Model (SSM).

• Actual Cost: The Actual Cost for the service calculated depends on the customer requirement.

- Resource Cost = Cost of resources per hour.

$$Resource\ Cost(RC) = \sum_{i=1}^{N} RC_i \tag{8}$$

Where RC_i is Resource cost for Resource i.

- Quality of Service = Resource Cost * Quality of Service required

$$QoS\ \ Cost = RC * q \tag{9}$$

- Security cost for each resource

$$SC_i \ \ = RC_i * Rw_i \tag{10}$$

- Security Cost for all Resources

$$SC = \sum_{i=1}^{N} SC_i = \sum_{i=1}^{N} RC_i * Rw_i \tag{11}$$

- Actual Cost AC = Sum for all resources (Resource Cost * Security weight of a Resource + Resource Cost) + QoS Cost

$$AC = RC + SC + QoS\ Cost$$

Therefore

$$AC = \sum_{i=1}^{N} (RC_i * R_{tmi}(1 + q + Rw_i)) \tag{12}$$

2.4 The Model

The main idea of the SSM is to categorise the submitted tasks on their security weight. The model has the following stages, shown in Figure 1:

1. Submit: A customer requests a service and provides information on all attributes of the requirement needs. They submit and identify attributes such as: Over all Time of the service, Number of tasks, tasks importance, budget, security level for each task, and QoS required.

2. Scheduling process: The scheduler analyses the customer requirements and categorises the tasks depending on the security weight of each task.

3. Calculations: In this stage, the scheduler calculates the Actual Cost and compare it to the customer budget. That will help to identify the service attributes and resources (VMs). The service attributes are sent to the customer for confirmation.

4. Customer Confirmation: The customer will receive an overview of all costs to take the final decision before establishing the service.

5. Establishing the service: After the customers confirmation, the system will establish the service and execute the tasks over the allocated resources.

6. Re-Calculating: At run time, for each task, the scheduler tries to optimise the service performance, time cost, and modify the resource requirements to reduce actual cost.

Considering the resource actual time after tasks execution process there is a chance to add some credit to the customer budget b to reduce the cost. So, the SSM will calculate the sum of actual times for each task in the resource, then reflect the actual time on the resource cost.

The SSM will check if the cost of actual resource time is greater than the resource cost or equal or less. So, if actual time is less than the required time there will be credit added to the customer, and if the actual time is more than the required time then the SSM will stop the service until the customer add more credit to continue with the service.

2.5 SSM Scheduling Function

SSM scheduling function f includes the following steps:

1. Categorise: from the attributes given by the customer this step uses the security level for each task to categorise tasks into groups.

2. Compare and Order: each group will be ordered to make the higher security task and with higher importance comes first. Then resources will be identified to be the same number of the group with equivalent security level.

3. Assign: after ordering each group will be assigned to the proper resource R that match the security level.

4. Execute: executing tasks over allocated resources.

2.6 SSM Algorithm

The SSM will use basic algorithm that divided into multiple steps starting from getting the customer inputs including budget, QoS, maximum time, and tasks information. This information will be used to calculate the total cost of the service. Also, the task information such as security weight will be used to categorise tasks into groups. The security weight has been set with constant level to calculate the cost. Otherwise, the security factor will be less important and the differences between level to another will be small with no effects.

2.7 Fast-Track

The Fast-Track technique is to give any task a high priority for execution depending on the security level required and it will be applied at the execution step if there dependencies between tasks. For example, if a customer submitted three tasks requiring the same security level t_1, t_2, t_3 and importance level 1,2,3 respectively. The normal executing order is t_3,t_2,t_1. If t_3 depends on t_1, then applying the Fast-Track technique and the executing order will be t_2, t_1, t_3.

3. EXAMPLE OF COSTS

This section presents simple examples of calculating costs (Actual Cost, Time).

3.1 Example:

1. Customer has requested and submitted information as shown in Table 1, q=0.0 and e = 60 min.

2. Scheduling process: Resources: N = 2 resources. For the tasks submitted the resources associated as follows: First task t_1 will be allocated to resource R_1 with same security weight 0.25. The other task t_2 will be allocated to resource R_2 with security weight 0.5.

3. Calculation: the system will start to calculate the initial cost for each resource and send it to the customer as follows:

- Resource Cost (RC) = 20 + 20 = 40

- QoS Cost = 40*0.0 = 0.0

- Security Cost for resources (SC) = 5 + 10 = 15

- Actual Cost = 40 + 15 + 0 = 55

- Check if less than or equal to b

4. Customer Confirmation: t_1 requires one resource (R_1) and t_2 requires one resource (R_2) with Time e = 60 min, and the cost =55 is less than or equal to b = 100.

5. Service Established over Resources R: After receiving the confirmation from the customer, the system will start to execute the tasks over the allocated resources. Because there is no dependency R_1 and R_2 can be established at the same time.

6. Re-calculating: In this example the time has been calculated after the establishing the service for the tasks:

$tm_1 = 1.67 \quad min\, , tm_2 = 3.33 \quad min$

So, the SSM will re-calculate the Actual Cost for t_1 over R_1 and t_2 over R_2, and then the Actual Cost will be as the following:

- RC = 0.56 + 1.11 = 1.67

- QoS Cost = 1.67 * 0.0 = 0

- SC = 0.14 + 0.56 = 0.7

Then AC = 1.67 + 0.7 + 0 = 2.37

Therefore, after the re-calculated step the AC is less than the AC at the calculation step.

This example shows that grouping tasks depending on the security level required make different cost to resources. But the cost can be reduced if there is no dependencies between tasks. Also, the re-calculation step consider the elapsed time to calculate the Actual Cost and can be reflected on the customer budget and can add more credit to it.

3.2 Example:

1. Customer has requested and submitted information as shown in Table 2 (t_6 depends on t_3, t_5 depends on t_1), q=0.0 and e = 60 min:

Table 2. SSM Analysing Customer Requirement

Security Level(Weight)/Importance	1	2	3	R	RC_i
1 (0.00)					
2 (0.25)	t_1	t_2		R_1	20
3 (0.50)					
4 (0.75)	t_3	t_4		R_2	20
5 (1.00)	t_5	t_6		R_3	20

2. Scheduling process: The SSM will analyse the request from Table 2. The number of resources is N =3, and each task allocated to the relevant resource.

3. Calculation: After analysing the requirement, the system sends the details to the customer to get the final confirmation to establish the service.

- Resource Cost (RC)= 20 + 20 + 20 = 60

- QoS Cost = 60*0.0 = 0.0

- Security Cost for resources (SC)= 5 + 15 + 20 = 40

- Actual Cost = 60 + 40 + 0 = 100

- Check if less than or equal to b

4. Customer Confirmation: After the scheduling process the system will send the details of the service to the customer. The customer needs to confirm to establish the service.

5. Service Established over Resources R:

• The normal tasks execution order depends on the security and importance levels only:

- $R_3 : t_6, t_5$, $R_2 : t_4, t_3$, and $R_1 : t_2, t_1$

• But with dependencies this order will change and the SSM will use the fast-tracking technique, and the new execution order will be as follows:
- $R_2 : t_3^{FT}, t_4$, $R_1 : t_1^{FT}, t_2$, and $R_3 : t_6, t_5$
• The system will start running tasks in the fast-track list for each resource. Then it will use time-out technique to ensure there will not be huge delay in the service and not affecting the other tasks.

Example 2 shows that scheduling process with taking security level as main driver to execute tasks over different resources is very complex with dependencies between tasks. But with applying the fast-track technique to simplify the process when dependencies exist, and not causes any delay to other tasks.

4. FUTURE WORK

Further investigations need to be done on the SSM. Also, it will be examined to see how it works under a significant load. For example, if there a large number of tasks need to be executed with different security levels and the customer demands low costs in terms of the available budgets. What is the impact in this case on identifying resources and possible cost?

5. CONCLUSIONS

This paper has described the Scheduling Security Model (SSM). It has presented an overview of the SSM components in each stage of the service and what components submitted from a customer and how the SSM will calculate the costs and resources identified. Furthermore, it showed how the SSM creates the groups of the security level required for each task and the task importance. Then it explains very simple examples of costs and how the model works and how it reflected of the costs. Finally, as a result of the worked calculated examples suggested that the complex examples would clarify other cases related to costs and identifying resources.

6. ACKNOWLEDGMENTS

All thanks to Taif University in Kingdom of Saudi Arabia for the funding.

7. REFERENCES

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., and Stoica, I. Above the clouds: a berkeley view of cloud computing. *University of California at Berkeley* (2009).

2. Bessani, A., Correia, M., Quaresma, B., André, F., and Sousa, P. Depsky: dependable and secure storage in a cloud-of-clouds. *ACM Transactions on Storage (TOS) 9*, 4 (2013), 12.

3. Buyya, R., Ranjan, R., and Calheiros, R. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. *International Conference on Algorithms and Architectures for Parallel Processing* (2010), 13–31.

4. Cachin, C., Haas, R., and Vukolic, M. Dependable storage in the intercloud. Tech. rep., Research Report RZ, 3783, 2010.

5. Chang, V., Bacigalupo, D., Wills, G., and De Roure, D. A categorisation of cloud computing business models. *Proceedings of the 10th ieee/acm international conference on cluster, cloud and grid computing* (2010), 509–512.

6. Che, J., Duan, Y., Zhang, T., and Fan, J. Study on the security models and strategies of cloud computing. *Procedia Engineering 23* (2011), 586–593.

7. Hu, Y., Chen, H. C., Lee, P. P., and Tang, Y. Nccloud: applying network coding for the storage repair in a cloud-of-clouds. *FAST* (2012), 21.

8. Jackson, D., Snell, Q., and Clement, M. Core algorithms of the maui scheduler. *Workshop on Job Scheduling Strategies for Parallel Processing* (2001), Workshop on Job Scheduling Strategies for Parallel Processing pp.87–102.

9. Kim, Y. G., Wang, Y., Park, B., and Choi, H. H. A heuristic resource scheduling scheme in time-constrained networks. *Computers & Electrical Engineering 54* (2016), 1–15.

10. Li, Q., Hao, Q., Xiao, L., and Li, Z. Adaptive management of virtualized resources in cloud computing using feedback control. *First International Conference on Information Science and Engineering,IEEE* (2009), 99–102.

11. Nallur, V., and Bahsoon, R. A decentralized self-adaptation mechanism for service-based applications in the cloud. *IEEE Transactions on Software Engineering 39*, 5 (2013), 591–612.

12. Patil, R., and Singh, R. Scaling in cloud computing. *International Journal of Advance Research, IJOAR 1* (2013), 21–27.

13. Paul, W. A multi-level security model for partitioning workflows over federated clouds. *Journal of Cloud Computing, Springer 1*, 1 (2012), 1–15.

14. Tripathy, L., and Patra, R. R. Scheduling in cloud computing. *International Journal on Cloud Computing: Services and Architecture (IJCCSA) 4*, 5 (2014), 21–7.

15. Wongthai, W., Rocha, F., and Van Moorsel, A. Logging solutions to mitigate risks associated with threats in infrastructure as a service cloud. *Cloud Computing and Big Data (CloudCom-Asia), International Conference on IEEE* (2013), 163–170.

16. Xie, T., and Qin, X. Scheduling security-critical real-time applications on clusters. *IEEE Transactions on Computers 55*, 7 (2006), 864–879.

Authors' Background

Abdullah Sheikh	PhD candidate	Cloud Computing, Security	
Malcolm Munro	Professor of Software Engineering	Reverse Engineering, Program Comprehension, Software and System Visualisation, Documentation, Impact Analysis	http://community.dur.ac.uk/malcolm.munro/
David Budgen	Professor of Software Engineering	Evidence-based software engineering (EBSE), Evidence-informed software development (EISD), Software design, Healthcare Informatics	http://community.dur.ac.uk/david.budgen/