

Evaluating a Cloud Service using Scheduling Security Model (SSM)

Abdullah Sheikh¹, Malcolm Munro², David Budgen³
Department of Computer Science
Durham University
Durham, United Kingdom

Abstract—The development in technology makes cloud computing widely used in different sectors such as academic and business or for a private purposes. Also, it can provide a convenient services via the Internet allowing stakeholders get all the benefits that the cloud can facilitate. With all the benefits of cloud computing still there are some risks such as security. This brings into consideration the need to improve the Quality of Service (QoS). A Scheduling Security Model (SSM) for Cloud Computing has been developed to address these issues. This paper will discuss the evaluation of the SSM model on some examples with different scenarios to investigate the cost and the effect on the service requested by customers.

Keywords—Cloud computing; security; scheduling; evaluating; cloud models

I. INTRODUCTION

Cloud computing is significantly used by educational and business sectors and for personal use for various purposes. Due to the rapid growth and the development in technology and facilities that cloud services can provide it has added a fascinating transformation to the Information Technology (IT) industry. Also, cloud computing provides convenient services enabling access to different computing resources such as networks, storage, and applications.

Cloud computing includes services such as data services, storage services, scheduling services, accessing to applications via the Internet, on demand self-service, and service management. Data service is about all database services, processing, and data store. While storage services include using a cloud storage system to manage saving data remotely in different storage location. The scheduling services include allowing customers to execute tasks over virtual resources and trying to allocate these tasks to these resources efficiently.

All these services can be provided upon customer request without or with less service provider interaction. For example, a customer can request storage space by submitting the request to a provider website. Then the customer can get the service by finalising service payment without any interaction from the service provider.

Cloud services bring various benefits to stakeholders (providers and customers). These benefits include wide access to software and applications over the Internet without any need to install any software to the customer terminal device. Moreover, using cloud service can be cost effective and as the cloud computing environment depends on reducing infrastructure cost. Also, it comes with different kind of risks such as security risks and financial risks.

There are many kind of risks that are related to the use of cloud computing. Risks such as Security and Privacy are big concerns to all parties in a cloud environment as the cloud services must be trusted and secured. So, any breach or failure in security will cause loss of customers and the business. Another risks that makes customer aware of getting a service is that they will be locked into one provider until their service is finished. This makes the service provider more concern to provide a trusted service.

These considerations include the need of improving the Quality of Services (QoS) provided. QoS includes different aspects such as time, service performance, reducing cost, and some non-functional requirements like reliability and recovery [1]. The success of applying these QoS aspects will improve the cloud services to meet customer expectations.

With all the benefits of the cloud, security is still one of the main concerns that affect the use of the cloud service. Cloud providers will be subject to many threats at different level of the cloud. Similarly, customers have concern about security and they share some responsibility with the cloud providers to keep the service security at a high level.

For example, if a customer requests a cloud service with a set of tasks with different security levels, it is required to have a technique that can handle this request. This technique should be able to execute the tasks submitted in the right order combined with security and QoS aspects.

Executing tasks requires using a scheduling process that has security as the main category, then uses priority to put tasks in right order. Security as a feature will be applied to all parts of the service, and the QoS will be applied to make the service more reliable and more efficient while the service is running. This complex request should be cost effective because the customer needs a cloud service that is secure, reliable, and with a very competitive cost compared to other service provider.

A recent review [2] has investigated and described the current situation of the cloud models. It then discusses the requirement of having a cloud security model based on cost that can manage requests focusing on security as a main feature associated with QoS aspects to meet the customer requirements. Then to execute scheduled tasks over allocated resources.

Moreover, Sheikh et al. [3] define a Scheduling Security Model (SSM) for a Cloud Environment. The SSM considers Security, Quality of Service (QoS), and Scheduling to allocate

Resources for predefined tasks based on security as a key element for a cloud service request. The security aspect affects task scheduling process, service cost, and service time. QoS levels will be considered as a feature included for the service cost. The Scheduling process will be focused on serving Tasks with higher security and using Task Importance to ordering them depending on Tasks priority. Fig. 1 shows the position of the SSM combining these features for a cloud service.

This paper focuses on evaluating the SSM model on some examples with different scenarios to investigate the cost and the effect on the service requested by customers.

This paper is organised as follows. Section II gives an over view of the features that involved for developing the Scheduling Security Model (SSM) which are Scheduling, QoS, and Security.

Section III discusses the results of the examples from the SSM, then Section IV discusses the evaluation of the SSM answering the questions that helped to develop the model. Section V concludes explaining aspects that important to be addressed by the service.

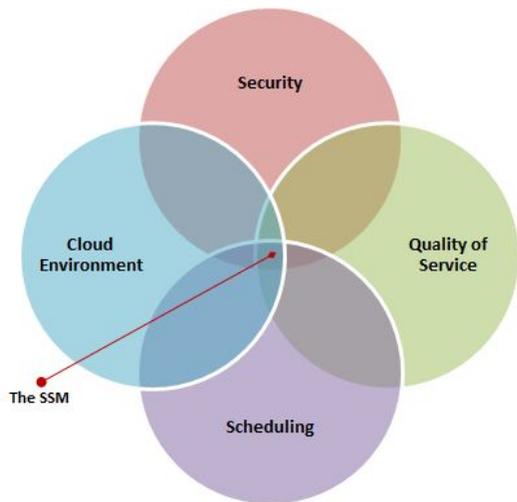


Fig. 1. SSM combined Features

II. BACKGROUND

This section serves as a background and a general view of the features that involved to develop the Scheduling Security Model (SSM) for a cloud environment which are Scheduling, QoS, and Security.

A. Scheduling

Scheduling is a process of decision making to deal with allocating resources to tasks within a certain amount of time [4]. There are many type of resources and it can be a machine in a workshop, resource in computing environment [5]. A scheduler is classified as follows [4]:

- Batch Scheduling: used to avoid any handling during the running time [34]. There are two types of batch scheduling, serial and parallel batching. In serial batching tasks with same setting can be executed one by one on a machine. In parallel batching a set of tasks can be grouped and executed at same time.
- Interactive: to allow decision making at running time take an immediate response.
- Real Time: the ability to schedule tasks with specific time requirements.
- Parallel: tasks or group of tasks executed at the same time in one VM or more [4].

There are three level of scheduling decisions:

- 1) Long Term: to control and decide what task execute first and to be supported once at anytime.
- 2) Medium Term: to control switching tasks for different criteria such as non active, fault, and low priority.
- 3) Short Term: to allow frequent interactive to take decisions in short time slot.

The main scheduling goals are:

- 1) Performance:
The scheduling algorithms should be able to consider the following measures in order to get good performance behaviour:
 - a) Maximise CPU Utilisation: to control the number of tasks that can be processed.
 - b) Maximise Throughput: to execute as many tasks as possible in a certain amount of time.
 - c) Maximise Scheduling Efficiency: to execute all tasks without interrupting.
 - d) Minimise Waiting Time: to reduce the amount of time that needed for executing tasks for users.
 - e) Minimise Energy: to control and reduce the power consumption of resources.
- 2) Fairness:
One of the important goals of scheduling is to treat all tasks to run in a reasonable time.
 - a) Equal CPU consumption: to allocate tasks the same processing time in the CPU.
 - b) Fair per(user, process, thread): giving all same characteristics for execution.
 - c) CPU bound, I/O bound: to allow direct priority to task from a user.
- 3) Unfair:
Sometimes the scheduling process tends to be unfair by giving advantage to one task over another for a specific aim.
 - a) Priority System: to run task with higher priority to run first then the less priority one.
 - b) Avoid starvation: to prevent that any task stays with no processing.

According to Yadav and Upadhayay [6], there are a number of existing scheduling algorithms as follows:

- 1) First-Come First-Served/ First-In First-Out (FCFS/FIFO):
Small tasks can be delayed behind large sized tasks.
- 2) Shortest Processing Time First/Shortest Job First (SPT/SJF):
The system just keep executing small tasks over large tasks.
- 3) Round Robin Scheduling (RR):
The scheduler allocates a fixed time to each task then executes tasks in a cycle way without any priorities.
- 4) Scheduling Priority (Priority):
Assigning priority to tasks will affect fairness of the system over other tasks. Also, tasks with less priority will never be executed and if so it will be delayed.
- 5) Multilevel Queue (MLQ) and Multi-level Feedback Queues (MLFQ):
Tasks queues can be divided into different categories where each class has its own scheduling requirements [6]. The difference is that the Multi-level Feedback Queues (MLFQ) can analyse the behaviour time of execution of the processing and according to changes on its priority.

B. Quality of Service (QoS)

Quality of Service (QoS) is one of the important factor that can help to improve any services, software, and applications [1]. So, the QoS means that the essential services features should meet all customer requirements.

According to Ramadan et al. [1], to have a good QoS there are some factors that need to be considered which affect the overall service as follows:

- Flexibility: It is all about managing any changes at the running time without any harms to the system.
- Maintainability and Readability: Similar to the flexibility but it is more focused on error correction and making any modification needed.
- Performance and Efficiency: It is all about the response time and making sure there is no delay or unexpected waiting time.
- Scalability: It is about responding to customers' activities in reasonable amount of time.
- Availability and Robustness: It is all about the availability all the time even if a failure has occurred.
- Usability and Accessibility: It is all about making the user interface the most visible side by making it very comfortable for the customer and easy to use.
- Platform Compatibility: For better quality the service should be running on as many different platforms as it can, with different system environments such as operating systems, and internet browsers.
- Security: It is the most important factor that needs to be considered in any service, and for QoS there is a need to apply security policies to make sure there are no security breaches at any level.

C. Security

Security is a concept that the process protect from physical or digital unauthorised use of any asset [7]. Also, security is a critical feature for any Service. The service must be secure and trusted for both customer and provider as they have both agreed in the Service Level Agreement (SLA) [8]. Security issues can affect Data, Networks, Communications, Privacy, unauthorised access and most things connected via the internet.

All of these aspects need to be protected, and each one requires a different way of security. So, these aspects can be classified into different security levels from high to low. This classification depends how valuable information is included in each asset. For example, storing very important government data requires a very high security level. This security level includes physical security measures and secure network connection and secure encrypted data storage. Also, it may require a limited access control to manage the process of retrieving and storing this data.

Security is also a very critical point that needs to be aware of all kind of information for all levels such as individuals, academic, business, and government even if it is digital or non digital materials.

Cloud computing as defined by the National Institute of Standard and Technology (NIST) [9] as "a model for enabling convenient, on-demand network access to a shared pool configurable computing resources (e.g. networks, servers, storage, application, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". Customers need to communicate with a service provider to request a cloud service. This communication process makes the customer and the provider reach an agreement of the level of the service. This agreement referred to Service Level Agreement (SLA) [10]. This SLA is the basis for the expected level of the service between the customer and the provider. The provider of a cloud architecture can offer various services to a customer. Quality of Service (QoS) refers to cloud stakeholders expectation of obtaining a desirable service meeting requirements such as timeliness, scalability, high availability, trust and security specified in the Service Level of Agreement SLA [11]. These services can vary both in terms of functionality (such as storage capacity or processor count) or in terms of the Quality of Service (QoS) provided [12]. In terms of the QoS a provider will offer a defined SLA which the customer can use when determining the 'best' provider for their needs.

As a result of this review, the proposed model by Sheikh et al. [3] has considered the overall security discussed by Watson [13] to develop Scheduling Security Model (SSM) to address the issues found in other approaches such as security and cost. The SSM use a following equation to calculate the cost [3]:

$$AC = \sum_{i=1}^N (RC_i * RT_i (1 + q + R w_i)) \quad (1)$$

Where

- RC_i is the cost per hour of Resource i .
- RT_i is the time used on Resource i .
- RC_i is the Quality of Service.

- Rw_i is the security weight of Resource i .
- N is the number of Resources.

III. EXAMPLES AND SCENARIOS

This section will discuss examples from Sheikh et al. [14]. The examples and scenarios will discuss aspects that are important to be addressed by the service which are Security, Time, and Cost.

In all examples and scenarios Tasks Security is identified by the customer and the value that added to the service request is to know all service requirements. Which is better than just requesting a service in a certain level of security with Resources then requesting a different Resource with different security level.

One of the main important issues for any cloud services is to be in a convenience time for any customer requests. Also, the service provider should manage to finish the cloud service on requested time or less.

It is very important that the customer identified all service requirements for any service request, because it would help the service provider to analyse the requirements and establish the service without any issues or asking the customer for more details.

A. Example: 1

A customer submits a service request with the details showing in Table I.

TABLE I. SSM CUSTOMER REQUIREMENT FOR EXAMPLE 1

Security Level(Weight)/Importance	1	2	3	Resource	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)					

For this example the customer entered the Quality of Service $q = 0.0$

Tasks submitted as follows:

- Task t_1 : $p_1 = 3$, $hw_1 = 0.25$
- Task t_2 : $p_2 = 2$, $hw_2 = 0.50$

After analysing the customer inputs, the SSM created a Resource for each Task as follows:

- R_1 for t_1
- R_2 for t_2

1) Scenario: 1.1: The calculated AC will be as follows:
From equation 1

$$AC = ((20*1) * (1+0+0.25)) + ((20*1) * (1+0+0.50))$$

$$AC = 25 + 30 = £55$$

For Example 1, Table I: the actual running time each Resource is as follows:

- $R_1 : t_1$ and $tm_1 = 18$ minutes
- $R_2 : t_2$ and $tm_2 = 13$ minutes

Tasks time line for Scenario 1.1 is shown in Fig. 2. Here both actual time and the elapsed time will be equal because there are no dependencies between the tasks. As a result the AC will be the cost of actual running time for each resource. The actual running time for each Resource:

- $tm_1 = 18$, RT_1 : Time for $R_1 = 18$ minutes
- $tm_2 = 13$, RT_2 : Time for $R_2 = 13$ minutes

So, the SSM will use the actual running time for all Resources to Re-Calculating the Actual Cost.

Re-Calculating:
From equation 1

$$AC = ((20*18/60)*(1+0+0.25)) + ((20*13/60)*(1+0+0.50))$$

$$AC = 7.5 + 6.5$$

$$AC = £14$$

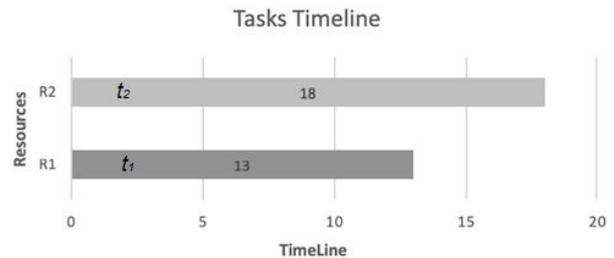


Fig. 2. Tasks Time Line for Scenario 1.1

B. Example: 2

A customer submitted a service request with the details shown in Table II.

TABLE II. SSM CUSTOMER REQUIREMENT FOR EXAMPLE 2

Security Level(Weight)/ Importance	1	2	3	Resource	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

For this example the customer entered the Quality of Service $q = 0.0$

1) Scenario: 2.1: For Example 2, Table II: Scenario: 2.1, the running time for each Resource, and the Tasks with FT indicates that is has been Fast Tracked as follows:

- $R_1 : t_1^{FT}$ and $tm_1 = 18$, t_2 and $tm_2 = 15$
- $R_2 : t_3^{FT}$ and $tm_3 = 13$, t_4 and $tm_4 = 10$

- $R_3 : t_6$ and $tm_6 = 10$, t_5 and $tm_5 = 5$

The dependencies are: t_5 depends on t_1 and t_6 depends on t_3 . If the SSM considers the running time for each Task, there will be a delay in executing Tasks t_6 and t_5 because of the dependencies. In this case, the calculated running time for each Resource will be as follows:

- $tm_1 = 18$, $tm_2 = 15$, RT_1 Time for $R_1 = 18+15 = 33$ minutes
- $tm_3 = 13$, $tm_4 = 10$, RT_2 : Time for $R_2 = 13+10 = 23$ minutes
- $tm_6 = 10$, $tm_5 = 5$, RT_3 : Time for $R_3 = 10+5 = 15$ minutes

Here there will be waiting time, so it will be added to RT_3 : Time for $R_3 = 13 + 10 + 5 = 28$ minutes. The reason for adding tm_3 not tm_1 is that tm_3 is less than tm_1 which can let the related Task t_6 start just after it finishes.

Re-Calculating: From equation 1

$$AC = 13.75 + 13.42 + 18.67$$

$$AC = \text{£}45.84$$

This is illustrated in Tasks time line in Fig. 3. If, the SSM does not consider the waiting time and just calculates the elapsed time as follows:

- $tm_1 = 18$, $tm_2 = 15$, RT_1 Time for $R_1 = 18 + 15 = 33$ minutes
- $tm_3 = 13$, $tm_4 = 10$, RT_2 : Time for $R_2 = 13 + 10 = 23$ minutes
- $tm_6 = 10$, $tm_5 = 5$, RT_3 Time for $R_3 = 10 + 5 = 15$ minutes

Re-Calculating: From equation 1

$$AC = 13.75 + 13.42 + 10.00$$

$$AC = \text{£}37.17$$

As a result of calculating the elapsed time the AC is less than calculating the AC with the running time.

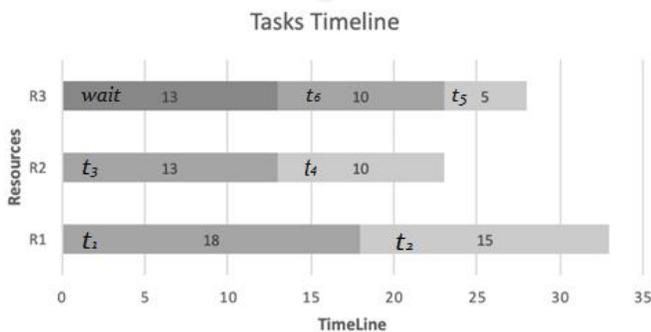


Fig. 3. Tasks Time Line for Scenario 2.1

IV. EVALUATION DISCUSSION

A. Evaluation Questions

The following questions are the evaluation questions that will be used to see how this research achieved its aims:

- 1) How does the SSM improve the security aspects of the cloud service?
The SSM applying security to the Tasks Level to make the customer to be more specific on the service requirement. After that the SSM applies to the Resource level to help running the cloud service at a trusted level.
- 2) How does the SSM impact Resource scheduling and performance and security?
Performance is a big issue, and currently the SSM is allowing for each Resource to run a single Tasks or a set of Tasks but there is a need to investigate how it impact the service from the performance prospective.
- 3) How well does the SSM help to achieve QoS?
The SSM applies different levels of QoS but it still need to clarify and inspect these QoS levels and how QoS affecting factors considered in these levels.

Fig. 4 shows the SSM features that are considered for a cloud service request. These features are Security, Priority, QoS, Time, and Cost per Resource. QoS contributes to the overall service cost and any change in QoS levels will affect the service cost. Security applied to the service cost as well and used in the scheduling process by Categorising the Tasks by the Task Security Level. Priority will be used in the scheduling process to put the Tasks in the right order for each Resource. Time will be calculated initially before establishing the service and after to show the different between the elapsed time and the actual running time. Resource Cost will be calculated before establishing the service and after receiving customer confirmation it will be Re-Calculating to have the Actual Cost.

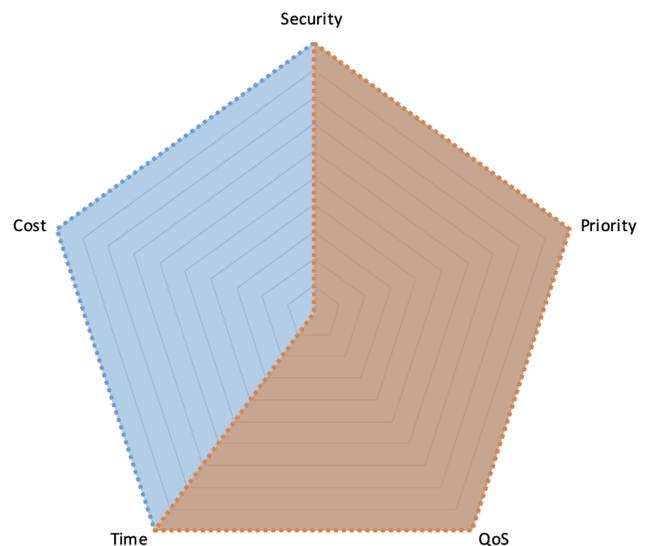


Fig. 4. Service Features of Applying the SSM

Table III shows the changes to the service requested on the SSM features in Example 1. The Features are Security, QoS, Priority, Time, and Cost. In this example, there are no change in the Security, Priority, and QoS. The change means there are no values reduced or increased that could cause any effects to the service. Also, the reason the starting points for all of this three features is in the centre of Fig. 4.

TABLE III. EXAMPLE 1.1

SSM Features	Example 1	
	Before	After
Security	Applied to Categorise Tasks	✓
QoS	Customer Input	–
Priority	Applied to Order Tasks	✓
Time	Initial Time Calculated for 1 hour or 60 minutes	18 minutes
Cost	Initial Cost Calculated £55	£14

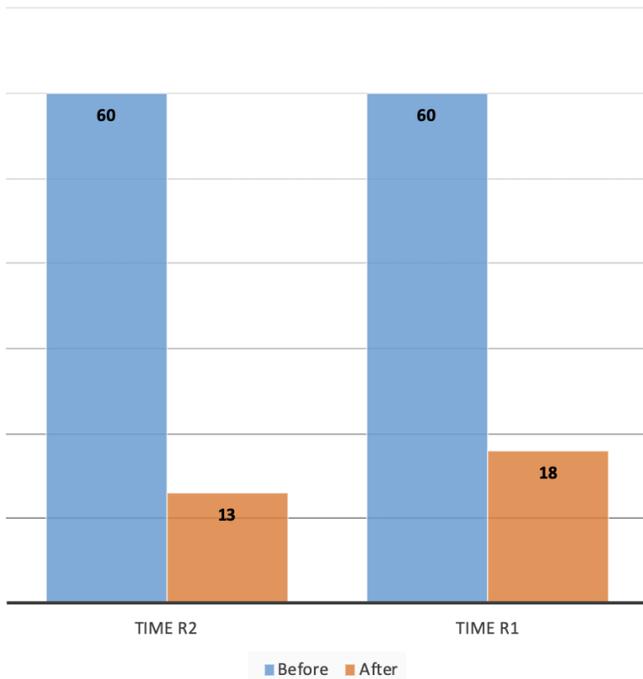


Fig. 5. Example 1 Scenario 1: Changes in Service Time Before and After SSM Applied

As shown in Fig. 5, the calculated time is less than the initial time after establishing the service. The Cost calculated in Pounds (£) as a cost unit for this example. As seen in Fig. 6, the initial cost before establishing the service is £55. Then it reduces after running the service to £14. Service time is calculated initially for 1 hour or 60 minutes, but the SSM will Re-calculating the service time after getting the Customer Confirmation to establish the service.

So, in Example 1 the service time calculated for each Resource and for the first Resource the time is 13 minutes and for the second Resource is 18 minutes. As a result, both Resources did not take more than the higher Time which is the time for the second Resource, and it reflected on the total or the Actual Cost (AC).

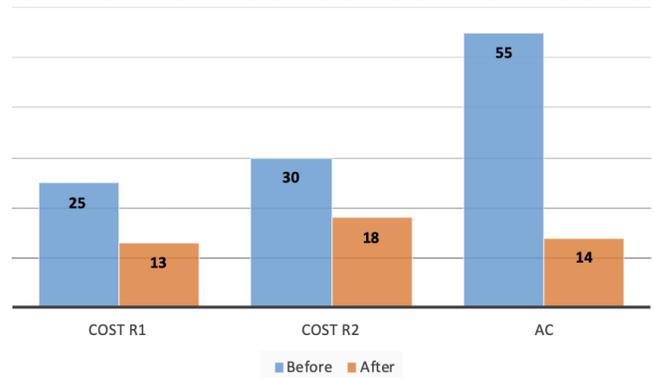


Fig. 6. Example 1 Scenario 1: Change in Service Cost Before and After SSM Applied

Table IV. Example 2 shows that in the Re-Calculating step the cost using either the elapsed time or the actual running time are less than the initial cost. Fig. 8 shows the change in service time for each Resource before and after applying the SSM. As a result, the AC of the elapsed time is less than the AC of the running time.

TABLE IV. EXAMPLE 2

SSM Features	Example 1	
	Before	After
Security	Applied to Categorise Tasks	✓
QoS	Customer Input	–
Priority	Applied to Order Tasks	✓
Time	Initial Time Calculated	✓
Cost	Initial Calculated	✓

The change in cost for each Resource and then the different between the initial cost and AC shown in Fig. 7. Also, this example has shown how the SSM works with the Tasks dependencies by using the Fast-Track technique with its benefits for scheduling the dependent Tasks with high Security Level.

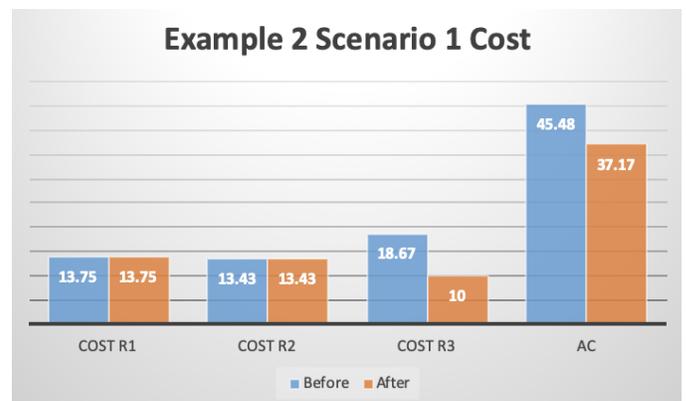


Fig. 7. Example 2 Scenario 1: Change in Service Cost Before and After SSM Applied

B. Compare with Other Approaches

In order to identify more implications and to clarify issues in the SSM. As a result, the comparison showed that the SSM and other model have different cost and with different effects on service time. The SSM has more features than other models, but one of the implications that have been founded is the SSM does not allow more than one Resource in the same Security Level. This can affect the total service time and it might cause a delay to execute other Tasks in the same Resource.

However, there is a lack of similar work with security as main feature for scheduling Tasks over allocated Resources. Table V. shows what the SSM provides against other Cloud Models. Also, it shows that there are shared features but there are for different purpose. For example, the Cloud Trust Model [30] is including security but it serves DaaS. So, the SSM adds to Scheduling and QoS the features Security, Cost, Service Availability, and IaaS all together to improve the cloud service.

V. CONCLUSION

This paper provided a discussion of the results produced from the SSM, which helped to discuss the evaluation questions to give better understanding to the SSM. It presented how the SSM improves the security aspect of the Cloud service by implementing the security as a main feature for executing Tasks and the effects on the Cost. Then it showed the differences in time and cost before and after establishing the service.

ACKNOWLEDGMENTS

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

REFERENCES

[1] H. Ramadan, and D. Kashyap, *Quality of Service (QoS) in Cloud Computing*, International Journal of Computer Science and Information Technologies (IJCSIT), vol. 8, No. 3, pp. 318–320, 2017.
[2] A. Sheikh, M. Munro and D. Budgen, *Systematic Literature Review (SLR) of Resource Scheduling and Security in Cloud Computing*, International Journal of Advanced Computer Science and Applications (IJACSA), IEEE, vol. 10, no. 4, pp. 35–44, 2019.

[3] A. Sheikh, M. Munro and D. Budgen, *Scheduling Security Model (SSM) for a Cloud Environment*, Conference of Cloud Computing, ACM, vol. 2, no. 1, pp. 1–15, 2018.
[4] CS. Pabla, *Completely fair scheduler*, Linux Journal, ACM, vol. 1, no. 184, pp. 68–72, 2009.
[5] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, Springer, pp. 1–10, 2008.
[6] RK. Yadav and A. Upadhyay, *A fresh loom for multilevel feedback queue scheduling algorithm*, International Journal of Advances in Engineering Sciences (IJAES), vol. 2, no. 3, pp. 21–23, 2012.
[7] CP. Pfleeger and SL. Pfleeger, *Security in computing*, Prentice Hall Professional Technical Reference, 2002.
[8] Tutorials Point, *Cloud Computing Quick Guide*, http://www.tutorialspoint.com/cloud_computing/cloud_computing_quick_guide.htm [Accessed in May 2019].
[9] P. Mell and T. Grance, *The NIST definition of cloud computing*, Computer Security Division, Information Technology, Laboratory, National Institute of Standards and Technology, Gaithersburg, 2011.
[10] P. Patel, A. Ranabahu and A. Sheth, *Service level agreement in cloud computing*, 2009.
[11] F. Panzieri, O. Babaoglu, S. Ferretti, V. Ghini, and M. Marzolla, *Distributed computing in the 21st century: Some aspects of cloud computing*, Springer, 2011.
[12] R. Shelke and R. Rajani, *Dynamic resource allocation in cloud computing*, International Journal of Engineering Research and Technology, ESRSA, vol. 2, no. 10, 2013.
[13] P. Watson, *A multi-level security model for partitioning workflows over federated clouds*, Journal of Cloud Computing, Springer, vol. 1, no. 1, pp. 1–15, 2012.
[14] A. Sheikh, M. Munro and D. Budgen, *Cost and Effect of Using Scheduling Security Model in a Cloud Environment*, International Conference on Cyber Security and Cloud Computing and Edge Computing and Scalable Cloud (CSCloud/EdgeCom), IEEE, pp. 95–101, 2019.
[15] A. Bessani, M. Correia, B. Quaresma, F. André and P. Sousa, *DepSky: dependable and secure storage in a cloud-of-clouds*, ACM Transactions on Storage (TOS), ACM, vol. 9, no. 4, pp. 12–33, 2013.
[16] M. Vrable, S. Savage, GM. Voelker, *BlueSky: a Cloud-Backed File System for the Enterprise*, Proceedings of the 10th USENIX Conference on File and Storage Technologies, USENIX Association, vol. 12, no. 10, pp. 237–250, 2012.
[17] R. Kotla, L. Alvisi and M. Dahlin, *SafeStore: a durable and practical storage system*, USENIX Annual Technical Conference, pp. 129–142, 2007.
[18] M. Alzain, B. Soh and E. Pardede, *A new model to ensure security in cloud computing services*, Journal of Service Science Research, Springer, vol. 4, no. 1, pp. 49–70, 2012.
[19] Tu. Hu, HCH. Chen, P. Lee and Y. Tang, *NCCloud: Applying Network Coding for the Storage Repair in a Cloud-of-Clouds*, FAST, vol. 12, no. 10, pp. 265–272, 2012.
[20] K. Bowers, A. Juels and A. Oprea, *HAIL: a high-availability and integrity layer for cloud storage*, Proceedings of the 16th ACM conference on Computer and communications security, ACM, pp. 187–198, 2009.
[21] C. Cachin, R. Haas and M. Vukolic, *Dependable storage in the intercloud*, IBM research, 3783, pp. 1–6, 2010.
[22] AJ. Feldman, WP. Zeller, MJ. Freedman and EW. Felten, *SPORC: Group Collaboration using Untrusted Cloud Resources*, OSDI, vol. 10, pp. 337–350, 2010.
[23] P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, M. Walfish, *Depot: Cloud storage with minimal trust*, Transactions on Computer Systems (TOCS), ACM, vol. 29, no. 4, pp. 79–117, 2011.
[24] W. Wongthai, F. Rocha, and A. Van Moorsel, *Logging Solutions to Mitigate Risks Associated with Threats in Infrastructure as a Service Cloud*, International Conference on Cloud Computing and Big Data (CloudCom-Asia), IEEE, 163–170, 2013.
[25] A. Shraer, C. Cachin, A. Cidon, I. Keidar, Y. Michalevsky, and D. Shaket, *Venus: Verification for Untrusted Cloud Storage*, Proceedings of the 2010 ACM workshop on Cloud Computing Security Workshop, ACM, pp. 19–30, 2010.

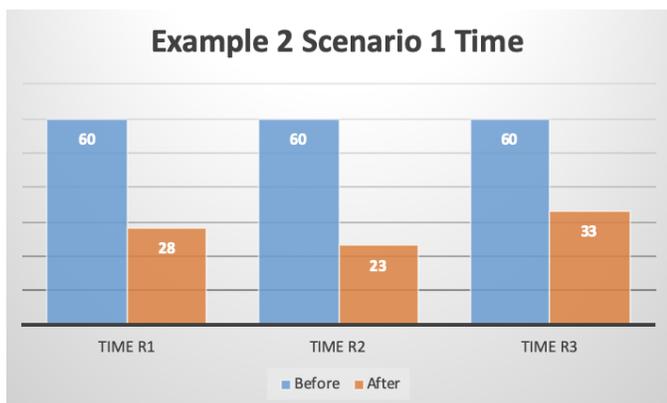


Fig. 8. Example 2 Scenario 1: Change in Service Time Before and After SSM Applied

TABLE V. THE SSM AND OTHER CLOUD MODELS

Models	Category				Main Focus													
	DaaS	SaaS	IaaS	Cloud storage	Availability	Confidentiality	Reliability	Intrusion	Integrity	Fault tolerance	Recovery fail	Cost	Scalability	Performance	Accountability	Latency	Security	Authentication
DepSky [15]	✓				✓	✓												
Bluesky [16]	✓																	
SafeStore [17]	✓					✓												
NetDB2-MS [18]	✓				✓			✓	✓									
NCCloud [19]				✓						✓	✓	✓						
HAIL [20]				✓	✓													
ICStore [21]				✓				✓										
SPORC [22]				✓	✓													
Depot [23]				✓	✓						✓					✓		
Logging Solutions [24]			✓									✓	✓	✓				
Venus [25]				✓					✓									
TCCP [26]			✓		✓				✓									
CCM [27]																		✓
Hexagon Model [27]																		✓
MTCM [27]																		✓
CSA [27]																		✓
Mapping model[27]																		✓
Separation Model [28]										✓								
Migration Model [28]									✓									✓
Availability Model [28]					✓													✓
Tunnel Model [28]									✓									✓
Cryptography Model [28]						✓												✓
NDSM [29]	✓			✓														✓
Cloud Trust Model [30]	✓																	✓
DSM [31]	✓					✓												✓
DSSM [32]	✓			✓														✓
SC [33]														✓				
SSM [3]			✓		✓							✓						✓

[26] N. Santos, KP. Gumjadi and R. Rodrigues, *Towards Trusted Cloud Computing*, HotCloud, vol. 9, no. 9, pp. 3–8, 2009.

[27] V. Chang, D. Bacigalupo, G. Wills and D. De Roure, *A categorisation of cloud computing business models*, Proceedings of the IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, IEEE Computer Society, pp. 509–512, 2010.

[28] G. Zhao, C. Rong, MG. Jaatun, FE. Sandnes, *Deployment models: Towards eliminating security concerns from cloud computing*, International Conference on High Performance Computing and Simulation (HPCS), IEEE, pp. 189–195, 2010.

[29] Sh. Ajoudanian and Mr. Ahmadi, *A novel data security model for cloud computing*, International Journal of Engineering and Technology, IACSIT Press, vol. 4, no. 3, pp. 326–329, 2012.

[30] H. Sato, A. Kanai and S. Tanimoto, *A cloud trust model in a security aware cloud*, International Symposium on Applications and the Internet (SAINT), IEEE, vol. 10, pp. 121–124, 2010.

[31] Z. Xin, L. Song-qing and L. Nai-wen, *Research on cloud computing data security model based on multi-dimension*, International Symposium on Information Technology in Medicine and Education (ITME), IEEE, vol. 2, pp. 897–900, 2012.

[32] HB. Patel, DR. Patel, B. Borisaniya and A. Patel, *Data storage security model for cloud computing*, International Conference on Advances in Communication, Network, and Computing, Springer, pp. 37–45, 2012.

[33] L. Tripathy and R.R. Patra, *Scheduling in cloud computing*, International Journal on Cloud Computing: Services and Architecture (IJCCSA), vol. 4, no. 5, pp.21–27, 2014.

[34] M. Mathirajan and AI. Sivakumar, *A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor*, The International Journal of Advanced Manufacturing Technology, Springer, vol. 29, no. 9, pp. 990–1001, 2006.