

Cost and Effect of Using Scheduling Security Model in a Cloud Environment

1st Abdullah Sheikh
Dept. Computer Science
Durham University
Durham, United Kingdom
abdullah.sheikh@durham.ac.uk

2nd Malcolm Munro
Dept. Computer Science
Durham University
Durham, United Kingdom
malcolm.munro@durham.ac.uk

3rd David Budgen
Dept. Computer Science
Durham University
Durham, United Kingdom
david.budgen@durham.ac.uk

Abstract—Resource scheduling is concerned with selecting the most appropriate resources to meet the needs of consumers. This is a complex task in Cloud Computing because of the large amount of available resources such as applications and data storage facilities. This is compounded further when security issues and quality of service are also factored in. A Scheduling Security Model (SSM) for Cloud Computing has been developed to address these issues. This paper will apply the SSM model on some examples with different scenarios to investigate the cost and the effect on the service requested by customers.

Index Terms—Scheduling, Security, Model, Cost, Cloud Computing

I. INTRODUCTION

Cloud Computing is defined by the National Institute of Standard and Technology (NIST) [5] as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable resources (e.g. networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”. There are many models in cloud computing focusing on different areas such as cloud service performance [9], service cost [4], and security [3]. Furthermore, Subashini and Kavitha [8] bring in focus some security issues in cloud service models such as data security and network security. These issues make a significant trade-off to the service models to obtain a reliable, trusted and secure services.

For the cloud there are four deployment models, which specify the availability of using cloud service [5], are: Public, Private, Community, Hybrid [2]. Also, there are some cloud deployment models security concerns such as data privacy and trust, policies, and data transfer [1] make provider aware of security for cloud service. Then providers need to apply security policies to deal with data access and security.

The Scheduling Security Model (SSM) [7] has been developed to make security the main element. It enables scheduling with security as the main driver while at the same time producing a cost of using the resources on the Cloud.

According to Watson [10] the overall security can be considered to be applied for executing tasks from trusted public resources to highly trusted private resources shown in Fig 1. So, for the SSM the security levels as the following:

1) level 1: is trusted public resource with basic security that can execute public tasks.

- 2) level 2: is more trusted public resource with more security setting that can execute public tasks.
- 3) level 3: is highly trusted public resource with security that can execute public tasks.
- 4) level 4: is trusted private resource with security level that can execute private tasks.
- 5) level 5: is highly trusted private resource with more security setting that can execute private tasks.

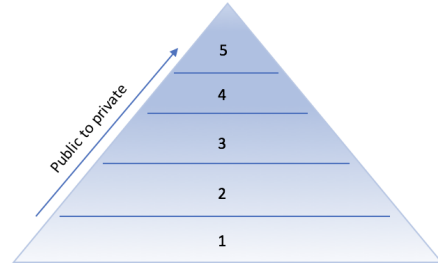


Fig. 1. Security Levels from Public to Private Resources

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.

The SSM is a service model that takes input from the Customer to allocate the requested tasks to resources that have the required security level. It calculates an initial cost of the use of the resources and then re-calculates the cost for a given usage of each task. The scheduling is also based on the importance level of each task. The SSM defines the following values: A service consists of a number of M tasks (t_1, t_2, \dots, t_M) , that can be run on a number N resources. The time for each task t_i is represented by tm_i . The actual cost (AC) for using N resources is:

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

Where, RC_i is the cost of using Resource i per hour.
 RT_i is the time spent using Resource i .
 Rw_i is the security weight for Resource i .
 q is the Quality of Service cost.

When, calculating the Actual Cost (AC):

- 1) Initially assume one hour usage for each resource:

$$RT_i = 1 \text{ for } i = 1, 2, \dots, N$$

- 2) For Re-Calculation: RT_i for R_i is the sum of the time for all tasks allocated to Resource i ,

$$RT_i = \sum_{j=1}^M (tm_j \mid t_j \in R_i)$$

R_i is the set of tasks t_j that run on Resource R_i . t_j is task j , and tm_j is time for task j .

Each task is given a security level (between 1 and 5 inclusive) and an importance value (between 1 and 3 inclusive). The scheduling will be carried out on the basis of the security level and the importance level of the tasks and resources. When calculating the costs the security level is mapped onto a security weight which is then used. The contribution of this paper is to evaluate the SSM model by working through a number of examples and scenarios to show how the costs change.

II. EXAMPLES OF COSTS

This section discusses the Re-calculation step considering the elapsed time of the tasks compared to the running time. The following examples are to show re-calculation for different scenarios. It assumes the Quality of Service cost q is 0, and that all the times are in minutes.

A. Example: 1

A customer submitted 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)					

1) *Scenario: 1.1:* For Example 1 Table I: the running time each resource as follows:

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

The calculated AC will be as follows:

$$tm_1 = 18, RT_1: \text{Time for } R_1 = 18$$

$$tm_2 = 13, RT_2: \text{Time for } R_2 = 13$$

$$AC = ((20 \times 1) \times (1+0+0.25)) + ((20 \times 1) \times (1+0+0.50))$$

$$AC = 25 + 30 = 55$$

The 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.

Re-Calculation:

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

$$AC = 7.5 + 6.5 = 14$$

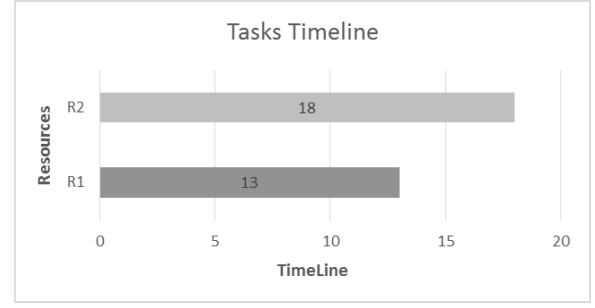


Fig. 2. Tasks Time Line for Scenario 1.1

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

1) *Scenario: 2.1:* For Example 2 Table II: Scenario: 2.1, assuming the running time 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 system 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 AC will be as follows:

$$tm_1 = 18, tm_2 = 15, RT_1: \text{Time for } R_1 = 18+15 = 33$$

$$tm_3 = 13, tm_4 = 10, RT_2: \text{Time for } R_2 = 13+10 = 23$$

$$tm_6 = 10, tm_5 = 5, RT_3: \text{Time for } R_3 = 10+5 = 15$$

Here there will be waiting time, so it will be added to RT_3 . Time for $R_3 = 13 + 10 + 5 = 28$. 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-Calculation:

$$AC = 13.75 + 13.42 + 18.67 = 45.84$$

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

$$tm_1 = 18, tm_2 = 15, RT_1: \text{Time for } R_1 = 18 + 15 = 33$$

$$tm_3 = 13, tm_4 = 10, RT_2: \text{Time for } R_2 = 13 + 10 = 23$$

$$tm_6 = 10, tm_5 = 5, RT_3: \text{Time for } R_3 = 10 + 5 = 15$$

Re-Calculation:

$$AC = 13.75 + 13.42 + 10.00 = 37.17$$

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

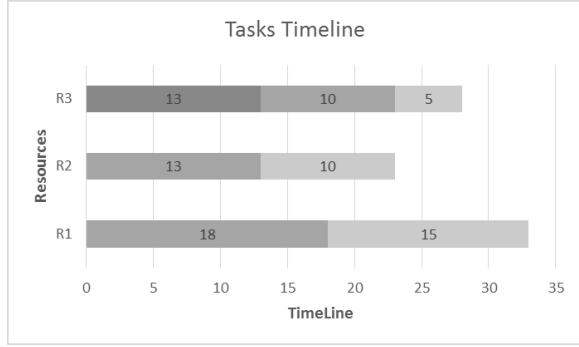


Fig. 3. Tasks Time Line for Scenario 2.1

2) *Scenario: 2.2:* For Example 2 Table II: Scenario 2.2, one of dependant tasks finishes before the other task that relates to a higher security level. So, the system will start executing the higher security task but in a different order depending on what dependant tasks finishes first:

- $R_1 : t_1^{FT}$ and $tm_1 = 8$, t_2 and $tm_2 = 3$
- $R_2 : t_3^{FT}$ and $tm_3 = 10$, t_4 and $tm_4 = 4$
- $R_3 : t_6$ and $tm_6 = 5$, t_5 and $tm_5 = 7$

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

$$tm_1 = 8, tm_2 = 3, RT_1: \text{Time for } R_1 = 8 + 3 = 11$$

$$tm_3 = 10, tm_4 = 4, RT_2: \text{Time for } R_2 = 10 + 4 = 14$$

$$tm_6 = 5, tm_5 = 7, RT_3: \text{Time for } R_3 = 5 + 7 = 12$$

Here there will be waiting time, so it will be added to RT_3 Time for $R_3 = 8 + 7 + 5 = 20$. The reason for adding tm_1 not tm_3 is that tm_1 less than tm_3 which can let the related task tm_5 start just after it finishes and is different from Scenario 2.1, the time line is shown in Fig 4.

Re–Calculation:

$$AC = 4.58 + 8.17 + 13.33 = 26.08$$

If, the waiting time is not considered then the elapsed time is calculated as:

$$tm_1 = 8, tm_2 = 3, RT_1: \text{Time for } R_1 = 8 + 3 = 11$$

$$tm_3 = 10, tm_4 = 4, RT_2: \text{Time for } R_2 = 10 + 4 = 14$$

$$tm_6 = 5, tm_5 = 7, RT_3: \text{Time for } R_3 = 5 + 7 = 12$$

Re–Calculation:

$$AC = 4.58 + 8.17 + 8.00 = 20.75$$

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

3) *Scenario: 2.3:* For Example 2 Table II: Scenario 2.3, there are two tasks dependent on the same task. So, the system will start executing tasks in same scheduling order considering the task importance.

- $R_1 : t_1$ and $tm_1 = 5$, t_2 and $tm_2 = 5$
- $R_2 : t_3^{FT}$ and $tm_3 = 5$, t_4 and $tm_4 = 5$

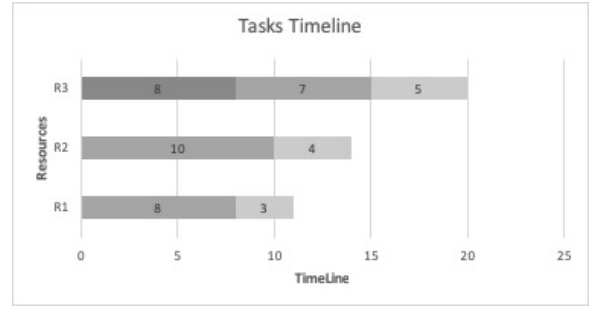


Fig. 4. Tasks Time Line for Scenario 2.2

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

The dependencies are: t_5 depends on t_3 and t_6 depends on t_3 . If the system considers the running time for each tasks, there will be a delay of executing tasks t_6 and t_5 because of the dependencies. In this case the calculated AC will be as follows:

$$tm_1 = 5, tm_2 = 5, RT_1: \text{Time for } R_1 = 5+5 = 10$$

$$tm_3 = 5, tm_4 = 5, RT_2: \text{Time for } R_2 = 5+5 = 10$$

$$tm_6 = 5, tm_5 = 5, RT_3: \text{Time for } R_3 = 5+5 = 10$$

Here there will be waiting time, so it will be added to RT_3 . Then, $RT_3 = 5+5+5 = 15$. The time line is shown in Fig 5.

Re–Calculation:

$$AC = 4.17 + 5.83 + 10.00 = 20$$

If, the system does not consider the waiting time and just calculates the elapsed time as follows:

$$tm_1 = 5, tm_2 = 5, RT_1: \text{Time for } R_1 = 5+5 = 10$$

$$tm_3 = 5, tm_4 = 5, RT_2: \text{Time for } R_2 = 5+5 = 10$$

$$tm_6 = 5, tm_5 = 5, RT_3: \text{Time for } R_3 = 5+5 = 10$$

Re–Calculation:

$$AC = 4.17 + 5.83 + 6.67 = 16.67$$

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

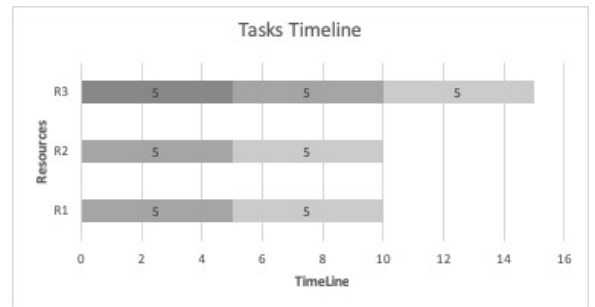


Fig. 5. Tasks Time Line for Scenario 2.3

C. Example: 3

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

1) *Scenario: 3.1:* For Example 3 Table III: Scenario 3.1, one of the dependant tasks finishes before the other tasks that are related to a higher security level. So, the system will

TABLE III
SSM CUSTOMER REQUIREMENT FOR EXAMPLE 3

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_7	t_6		R_3	20

start executing the higher security task in a different order depending on what dependant tasks finishes first:

- $R_1 : t_1^{FT}$ and $tm_1 = 8$, t_2 and $tm_2 = 3$
- $R_2 : t_3^{FT}$ and $tm_3 = 10$, t_4 and $tm_4 = 4$
- $R_3 : t_6$ and $tm_6 = 5$, t_5 and $tm_5 = 7$, t_7 and $tm_7 = 7$

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

$$tm_1 = 8, tm_2 = 3, RT_1: \text{Time for } R_1 = 8+3 = 11$$

$$tm_3 = 10, tm_4 = 4, RT_2: \text{Time for } R_2 = 10+4 = 14$$

$$tm_6 = 5, tm_5 = 7, tm_7 = 7, RT_3: \text{Time for } R_3 = 5+7+7 = 19$$

Here, there will be waiting time, so it will be added to $RT_3 = 8 + 7 + 5 + 7 = 27$.

The reason for adding tm_1 not tm_3 is that tm_1 less than tm_3 which can let the related task tm_5 start just after tm_1 finishes.

Re-Calculation:

$$AC = 4.58 + 8.17 + 18.00 = 30.75$$

If, the waiting time is not considered then the elapsed time is calculated as:

$$tm_1 = 8, tm_2 = 3, RT_1: \text{Time for } R_1 = 8 + 3 = 11$$

$$tm_3 = 10, tm_4 = 4, RT_2: \text{Time for } R_2 = 10 + 4 = 14$$

$$tm_6 = 5, tm_5 = 7, tm_7 = 7, RT_3: \text{Time for } R_3 = 5+7+7 = 19.$$

The time line is shown in Fig 6.

Again, as a result of calculating the elapsed time the AC is less than calculated AC with the running time. But, what if the system lets t_7 run first to reduce the total running time as follows: The waiting time will be less in both cases, in running time: $(8-7) + 7 + 5 + 7 = 20$

Re-Calculation:

$$AC = 4.58 + 8.17 + 13.33 = 26.08$$

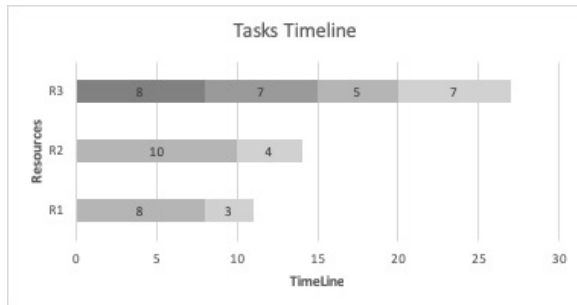


Fig. 6. Tasks Time Line for Scenario 3.1

2) *Scenario: 3.2:* For Example 3 Table III: Scenario 3.2, one of the dependant tasks finishes before the other tasks that are related to a higher security level. So, the system will start executing the higher security task in a different order depending on what dependant tasks finishes first:

- $R_1 : t_1^{FT}$ and $tm_1 = 8$, t_2 and $tm_2 = 3$
- $R_2 : t_3^{FT}$ and $tm_3 = 10$, t_4 and $tm_4 = 4$
- $R_3 : t_6$ and $tm_6 = 5$, t_5 and $tm_5 = 7$, t_7 and $tm_7 = 12$

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

$$tm_1 = 8, tm_2 = 3, RT_1: \text{Time for } R_1 = 8+3 = 11$$

$$tm_3 = 10, tm_4 = 4, RT_2: \text{Time for } R_2 = 10+4 = 14$$

$$tm_6 = 5, tm_5 = 7, tm_7 = 12, RT_3: \text{Time for } R_3 = 5+7+12 = 24$$

Here there will be waiting time, so it will be added to $RT_3 = 8+7+5+12 = 32$.

The reason for adding tm_1 not tm_3 is that tm_1 is less than tm_3 which can let the related task t_5 start just after tm_1 finishes.

Re-Calculation:

$$AC = 4.58 + 8.17 + 21.33 = 34.08$$

Time line for Scenario 3.2 shown in Fig 7. If, the waiting time is not considered then the elapsed time is calculated as:

$$tm_1 = 8, tm_2 = 3, RT_1: \text{Time for } R_1 = 8 + 3 = 11$$

$$tm_3 = 10, tm_4 = 4, RT_2: \text{Time for } R_2 = 10 + 4 = 14$$

$$tm_6 = 5, tm_5 = 7, tm_7 = 12, RT_3: \text{Time for } R_3 = 5+7+12 = 24$$

Here, if the system lets t_7 run first there will be a delay for running t_5 and t_6 with no waiting time.

Re-Calculation:

$$AC = 4.58 + 8.17 + 16.00 = 28.75$$

Also, in this scenario, t_6 will run after t_7 then t_5 as it has a higher security level, and it is depending on task t_3 .

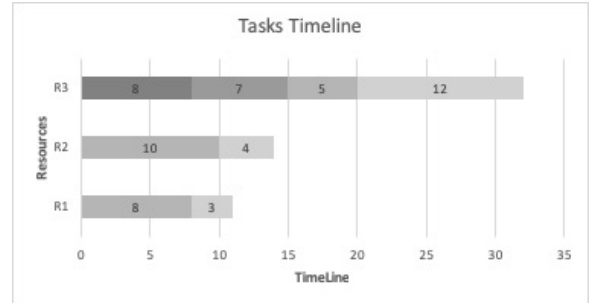


Fig. 7. Tasks Time Line for Scenario 3.2

D. Example: 4

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

1) *Scenario: 4.1:* For Example 4 Table IV: Scenario 4.1 where more than one task depend on each other. The dependencies are: t_5 depends on t_3 , and t_3 depends on t_2 . Also, t_6 depends on t_1 , then tasks will be allocated over resources as the following:

TABLE IV
SSM CUSTOMER REQUIREMENT FOR EXAMPLE 4

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

- R_1 : t_1^{FT} and $tm_1 = 8$, t_2^{FT} and $tm_2 = 3$
- R_2 : t_3^{FT} and $tm_3 = 6$, t_4 and $tm_4 = 4$
- R_3 : t_6 and $tm_6 = 5$, t_5 and $tm_5 = 7$

The implication here is that t_2 has higher importance than t_1 , then if the system lets t_2 run first it will cause a delay to run related tasks. That will cause more delay to t_3 then to t_5 and t_6 . The system run t_2 then t_1 , see Fig 8 for tasks time line, and the calculating time will be:

- R_1 : t_1^{FT} and $tm_1 = 8$, t_2^{FT} and $tm_2 = 3$, $RT_1 = 8+3 = 11$
- R_2 : t_3^{FT} and $tm_3 = 6$, t_4 and $tm_4 = 4$, $RT_2 = 3+6+4 = 13$
- R_3 : t_6 and $tm_6 = 5$, t_5 and $tm_5 = 7$, $RT_3 = 11+5+7 = 23$

The reason of adding tm_2 to RT_2 is that tm_3 is waiting for t_2 to finish. Also, adding RT_1 to RT_3 because R_3 can not start running tasks until R_1 is finished.

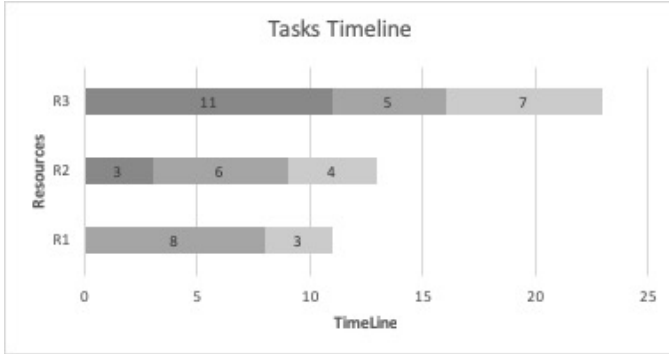


Fig. 8. Tasks Time Line for Scenario 4.1

III. SSM VS OTHERS

Comparing SSM against similar work will be useful to see similarities and differences and effect on cost.

For example, Tripathy and Patra [6] give an overview of scheduling tasks (jobs) with priorities. Their aim is to run all high priority tasks first over the number of resources required. Because the task priority is the main consideration it has been justified to have five level to order tasks to be executed over resources where 1 is minimum priority and 5 is the highest. They indicated that a job can be executed on multiple resources R_i . The following example explains how they execute jobs over resources. As a start, the task identified with the following attributes $job_i(j,k,l)$: i for job or task id, j for resources required, k for task duration, and l for task priority. Then the following service request given:

- $job1(2,5,1)$
- $job2(6,10,5)$
- $job3(2,5,4)$
- $job4(2,5,2)$

Fig 9 shows how tasks are running over resources.

If the SSM is used for this service to calculate the cost assuming that the security level is the minimum and as same as the Quality of Service then the actual cost will be the total cost for all resources. Also, all resources have the same running time = 15. So, the cost for each resource = $(15*20)/60 = 5$. There for the total cost for all resources = $5 * 6 = 30$.

IV. DISCUSSION

The worked examples show different cost that meet the customer requirements of cost and quality of service in the required time. For all examples, each resource can be a single task or a set of tasks. Also, each example ends up with a different cost that is less than the initial cost of establishing the service and show the different cost by calculating actual running time and calculating the elapsed time.

Example 1 Scenario 1.1, shows a service request with two different tasks at a different security level. So, the SSM allocated them to two resources, the basic cost was the cost per hour. But in the Recalculation step the system calculates the actual running time of the resources and as a result the AC becomes cheaper. Also, there is no big effect here in the cost because there is no difference between AC with elapsed time and AC with actual running time.

Example 2, shows a customer request of a service with more tasks and different security levels. Then it presents different possible scenarios with tasks dependencies to show the difference in cost and the differences that could affect it in the Re-calculating step.

In Scenario 2.1, the system compares the time of the tasks with the Fast Tracked tasks t_1 and t_3 for the dependant tasks t_5 and t_6 . If $tm_3 > tm_1$, the system lets the task with the less time run first. The effect is shown in Re-calculating the cost using either the elapsed time or the actual running time. As a result, the AC of the elapsed time is less than the AC of the running time.

Scenario 2.2, shows that the dependant task t_1 has tm_1 less than tm_3 , but the system lets t_1 run first because the total delay time will be less than if the system lets t_3 run first. This is the only situation that the system lets a task with less importance run first to avoid any possible delay to the service.

In Scenario 2.3, the system considers the running time for each task, and there will is a delay in executing tasks t_6 and t_5 because of the dependencies.

Example 3, shows a customer request of a service with different security levels with tasks that do not depend on any other tasks with a higher security level. Then it presents different possible scenarios to show the difference in cost and how it affects it in the Re-calculating step.

In Scenario 3.1, the system will start executing the higher security task but with different order depending on what dependant tasks finish first. Here, the system compares tm_7

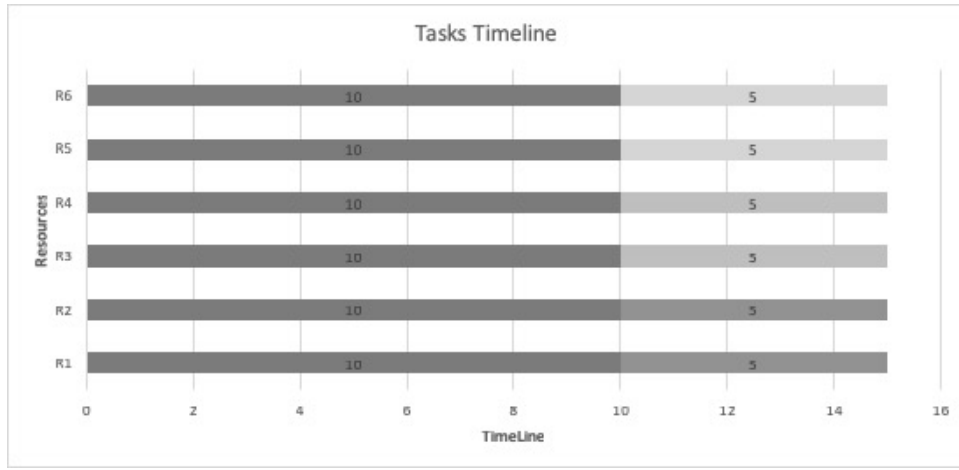


Fig. 9. Tasks Time Line for Tripathy and Patra [6]

with tm_1 and tm_3 , and it found tm_7 is the smallest. So, the system lets t_7 run first because the total waiting time will be less than if it waits for the other tasks to finish.

In Scenario 3.2, the system found that one of the dependant tasks t_1 with less time tm_1 finishes before the other tasks that are related to a higher security level. So, the system lets t_1 run first, and that lets the tasks in the higher security resource start running after it finished. As a result, the effect is the waiting time becomes less and the cost is cheaper than the cost of establishing the service.

Example 4, shows a customer request of a service with different dependencies which is more complex. Then it shows how the system will respond to this kind of request in possible scenarios.

Scenario 4.1, shows how the system works with more complex dependencies. The system compares task times to avoid any delay in the waiting time. Here, the system found $tm_3 > tm_1$. The main issue here is that t_2 has higher importance than t_1 . If the system lets t_2 run first it will cause a delay running related tasks. This will cause more delay to t_3 then to t_5 and t_6 . So, the system lets t_2 run first to make t_3 run next then t_1 and then all tasks in R_3 to run after. The effect here is that the cost becomes cheaper and the system chooses the possible time without any extra waiting time.

Comparing SSM with Tripathy and Patra [6] gives different cost with different effects. For the example given by Tripathy and Patra the cost is 30. But if the SSM take the service request and apply the model with same tasks, the cost will be different. Because, in the SSM there will be just one resource to run all tasks in same security level. Then the SSM will execute all the tasks in the order given with their times. The difference is that the resource will be reserved for each task at the time until it finishes, not like Fig 9 showing a delay and each resource is on hold until the task finishes. So, calculating the Actual Cost by SSM will be the cost of total time for one resource. The total time = $10 + 5 + 5 + 5 = 25$, considering the security level is 1 and q is 0, then $AC = 25 * 20 / 60 = 8.33$.

There are five main aspects that are considered for comparing the SSM and other approaches. These aspects are Security, QoS, Priority, Time, and Cost. Table V shows the comparison between the SSM and Tripathy and Patra [6]. The SSM cover more aspects than others. Both SSM and Tripathy and Patra [6] consider Priority to run tasks over resources. But in the SSM Priority is a secondary factor to order tasks for execution and the Security is the main driver to allocate tasks to an available Resource. Tripathy and Patra [6] allow multiple resources in one level different from the SSM, which adds more cost to the service.

TABLE V
COMPARE SSM WITH ANOTHER MODEL

Model	Security	QoS	Priority	Time	Cost
SSM	✓	✓	✓		✓
Tripathy and Patra [6]			✓	✓	

In all the scenarios, the system has applied the following time calculating steps:

- For each task ask if it has dependants .
 - If no dependants then add task time to RT_i time.
 - If dependants then go to dependant task and ask again.
- There will be different scenarios for calculating the time if dependant tasks are on different Resources R_i . So, time can be calculated from one of the following:
 - The dependant task for the higher security task has less or equal time than the others. So, it will be run first then add its time to the correspondent resource time RT_i .
 - The dependant task for the higher security task requires more time than the others. So, the system will run the task with less time and then return to it and add both times to RT_i .

- In the case, where the dependant task with tm_1 also has a dependant task t_2 with tm_2 , the system will add tm_2 to the resource task with tm_1 . Then both times will be added to the correspondent resource R_i .
- In the case, where two or more tasks with dependant tasks, the system will check the time for the dependant task with higher security and then decide which dependant task will run first then add its time to the correspondent R_i .

V. FUTURE WORK

Further investigations suggested and need to be done on the SSM. The results from the examples in this paper will be compared against similar work. More investigations on the Quality of Service levels of a service requested by a customer to study possible changes in cost and effects. That will help for the evaluation and to do more analysis on the SSM.

VI. CONCLUSIONS

This paper has examined the Scheduling Security Model (SSM) through worked examples of customer submitted service requests. Each example is presented with some possible scenarios. The SSM showed the Actual Cost (AC), then it applied the time calculating steps in the Re-calculating stage to present the cost of both elapsed time and actual running time. Furthermore, it discussed the cost and effect of the service in each scenario. The main effect is that both the AC of the elapsed time and AC of actual running time are cheaper than the cost of establishing the service. Also, by applying the time calculating steps the service will run with less waiting time. Finally, as a result of these worked examples and scenarios, it is suggested that comparing the SSM against similar scheduling models in same area would identify more implications and clarify other cases related to cost and effects on the service.

ACKNOWLEDGMENT

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

REFERENCES

- [1] T. Dillon, C. Wu, and E. Chang, "Cloud computing: issues and challenges", International Conference on Advanced Information Networking and Applications, IEEE, 27–33, 2010.
- [2] D. Goutam, A. Verma, N. Agrawal, "The performance evaluation of proactive fault tolerant scheme over cloud using CloudSim simulator", International Conference on the Applications of Digital Information and Web Technologies (ICADIWT), IEEE, 171–176, 2014.
- [3] V. Chang, D. Bacigalupo, G. Wills and D. De Roure, "A categorisation of cloud computing business models,"FAST, pp. 509–512, 2010.
- [4] Y. Hu, HCH. Chen, PPC. Lee and Y. Tang, "NCCloud: applying network coding for the storage repair in a cloud-of-clouds,"FAST, pp.21, 2012.
- [5] 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.
- [6] L. Tripathy and R.R. Patra, "Scheduling in cloud computing", International Journal on Cloud Computing: Services and Architecture (IJCCSA), 4, 5, pp.21-7, 2014.
- [7] A. Sheik, M. Munro and D. Budgen, "SSM: Scheduling Security Model for a Cloud Environment,"ACM, pp. 11–15, 2018.
- [8] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing", Journal of network and computer applications, Elsevier, 34, 1, 1–11, 2011.
- [9] L. Qiang, H. Qinfen, X. Limin and L. Zhoujun, "Adaptive management of virtualized resources in cloud computing using feedback control," First International Conference on Information Science and Engineering, IEEE, pp. 99–102, 2009.
- [10] P. Watson, "A multi-level security model for partitioning workflows over federated clouds," Journal of Cloud Computing, Springer, 1, 1, pp 1-15, 2012.