IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 6, Ver. II (Nov.- Dec. 2017), PP 59-67
www.iosrjournals.org

Smart Transaction in Services through Secured Framework for task scheduling in cloud environment

Prof. Ramesh Babu Inampudi[1], Mrs. Nahla Ahmed Farag[2]
[1,2](Department of Computer Science and Engineering)
[1]rinampudi@outlook.com
[2] nahlaalkhtry74@gmail.com

Abstract: *Managing multiple clouds for data transaction with proper tasks allocation and scheduling from decentralized place(service) is a big challenge in service oriented architectures which maintains virtual machines for data sharing and communications. Proper resources have to be allocated for data transactions and data divisions with proper security. This work proposes* ***STSASF(S****mart* ***T****ransaction in* ***S****ervices and* ***S****ecured* ***F****ramework) with 6 virtual machines communications in cloud environment. Based on the available bandwidth and long time waiting virtual machines will be identified by* ***STSASF*** *and will be allocated with tasks and scheduled for smooth data transactions. The data transactions would be in secured and maintained with proper log to avoid duplicate transactions. The switch mode is also proposed to start the free pool of virtual machines(6) at a time. So 6 virtual machines are in active state for data transactions. Rota(****radix****) algorithm is proposed for secure transactions. Depends on the capacities of the virtual machines STSASF will schedule the task of allocating the data divisions.*

Keywords*: -Transaction, Task Scheduling, Regex, Virtual Machine, pool, division.*

Date of Submission: 16-11-2017 Date of acceptance: 30-11-2017

I. Introduction

Cloud computing systems provide on pay-per-use access to computational resources [1].The information technology IT paradigm is the model of enabling parallel access to shared and configured resources(services, networks, storages, servers) which can provide rapidly with less effort of management over w3c(world wide web consortium).Cloud/service can provide various enterprise computational capabilities to compute, process and store the data in public and privately owned, third-party services/cloudswhich arelocated atdecen-tralized place called data center.Virtual machine [2]the emulation of a computer system in cloud computing is called "virtual machine". The virtual machines based on the computer systems architecture and serves as physical computer. The specialized hardware and hardware may be involved in the implementations of the virtual machines. System virtual machine (full virtualization VMs) is the substitution of real virtual machine. They provide the functionality to execute the whole operating system. A hypervisor uses local execution to manage and share the hardware, which allows multiple model of environments whoare isolated among them, still exists on the same physical machine. Scheduling policies are needed which ensures that both, the user and the service provider are benefited. Cloud computing uses virtualization to deal with such situation, where virtual machines are created as per user demand. Virtualization permits scalable cloud computing infrastructure to the user. Therefore, developing a model for scheduling user tasks on virtual machines is an important issue. Scheduling process in cloud can be generalized into three stages namely[3] i) Resource discovering and filtering –Datacenter Broker discovers the resources present in the network system and collects status information related to them. ii) Resource selection –Target resource is selected based on certain parameters of task and resource. iii) Task submission -Task is submitted to resource selected. The goal of scheduling algorithms in distributed systems is to schedule jobs to the flexible resources in accordance with flexible time, which includes finding out a proper sequence in which jobs can be executed under transaction logic constraints. The main advantage of task scheduling algorithm is to achieve a high performance computing and the best system throughput. Scheduling can be done at compile time (static scheduling)and/or at run time (dynamic scheduling) Static scheduling requires intelligent compilation support whereas dynamic scheduling requires sophisticated hardware support. In practice, dynamic scheduling is assisted by static scheduling to improve performance and to reduce hardware cost. On the other hand, static scheduling is often assisted by hardware interlocking to enforce the correctness of execution. Scheduling decisions can have a major impact on the performance of multiple-instruction-issue processors. The goal is to produce a code schedule that minimizes the execution time. In this paper, a task oriented scheduling model for the allocation of virtual machines in the cloud is proposed. Data centre has large number of requests in the form

of tasks of different nature. Some tasks require more RAM, others require high speed processors, high bandwidth or a combination of these attributes. A data centre has a number of virtual machines with different specification (RAM, MIPS etc.), with the aim is to allocate these virtual machines to the tasks by considering the nature of the tasks [4]. The Smart Transaction in Services through Secured Framework for task scheduling in cloud environment (STSASF) is used to make decisions for allocation of virtual machines on the basis of different attributes like BandWidth(BW), Time etc. of the tasks.

The *organization* of the paper is as follows. Section 2 presents literature review. Section 3, discusses the Architecture. In Section 4 presents Existing Approach. The Proposed framework is discussed in Section 5. The results and analysis are presented in section 6. Section 7 Conclusion and future work. Section 8 References.

II. Literature Review

In this section, related Smart Transactiontask scheduling algorithms put forward by existing researchers are discussed:D.I. George Amalarethinam et al.[5]:On-Demand service is proposed with pay-per-use schemes provided by services/cloud providers attracts the vendors to use and move from server technologies to service oriented technologies(cloud) environment. Both cloud service providers and vendors benefitted with enterprise level when the resources properly scheduled and the in time utilization of tasks from the service provider. Due to enterprise/commercialization the service/cloud environment leads to develop the new approaches for best economical factors. In this work customer facilitated cost-based (CFCSC) approach proposed to the favour of cloud customers with low cost. Khajehvand et al. [6] presented a scalable cost-time trade-off scheduling algorithm for grid computing environments with focused on task sizes, task parallelism, and heterogeneous resources as constraint towards evaluating the performance of their proposed algorithm. Dandhwani and Vekariya[7] presented a multi-objective scheduling algorithm for a cloud to locate and allocate clusters on best VMs based on machine heterogeneity. The result of performance shows improvement in execution time and makespan. Yue et al. [8] presented an Improved Multi-Objective niched Pareto Genetic (NPGA) method to minimized time consumption and financial cost of handling the user's cloud tasks.In Hua et al. [9], a PSO-based adaptive multi-objective task scheduling strategy is proposed that minimized both processing time and transmission time.Atul Vikas Lakra et al.[10]:proposed In distributed computing server farms apply server unification to improve the productivity of the assets. Numerous VMS are running on every datum focus to use the assets effectively. For the most part the cloud assets are underutilized because of poor planning of assignment in the server farm. Here we characterize how a multi-target undertaking planning calculation that maps to the errands to a VMS keeping in mind the end goal to enhance the throughput of the server farm and diminish the cost.Brototi Mondal et al.[11]:proposed another idea of virtualized PC assets. Distributed computing depicts a stage and sort of application. Servers in the cloud can be virtual machines spread over the system. Choosing hubs for executing an undertaking must be considered to abuse the adequacy of the assets. Neighbourhood advancement Stochastic Hill climbing is utilized for the assignment of the approaching occupations to the servers or virtual machines. As Cloud Analyst is a Cloud Sim-based Visual Model for breaking down distributed computing conditions and applications. A review is likewise made with Round Robin and FCFS calculations. Hao Yuan et al. [12]: proposed that virtual resource scheduling based upon a particle swarm scheduling algorithm, the paper introduces cellular automata theory to construct a new cellular particle swarm algorithm. This approach by mathematical modelling for virtual resource scheduling of the cloud computing and complete the final search configuration based on a directional optimization objective function. Experimental result shows that the proposed method has been more excellent scheduling performance, in the case of changes in resources, can be also kept as stable scheduling balance.

III. Architecture

First dashboard will select the data and after that all virtual machines will be started(Virtual machine1 to Virtual machine6) and centralized service(cloud service) will monitor the virtual machines to allocate the tasks to distribute the data depends on the capacity of each virtual machine. The distribution size depends on the data size. The task scheduler block will allocate the tasks to individual virtual machines and will be distributed the data to each virtual machine and log will be will updated by centralized service. Data will be encrypted before transaction(broadcast) using ROTA approach and will be broadcasted to destination. Again the log will be updated with this transaction (upload and download time).

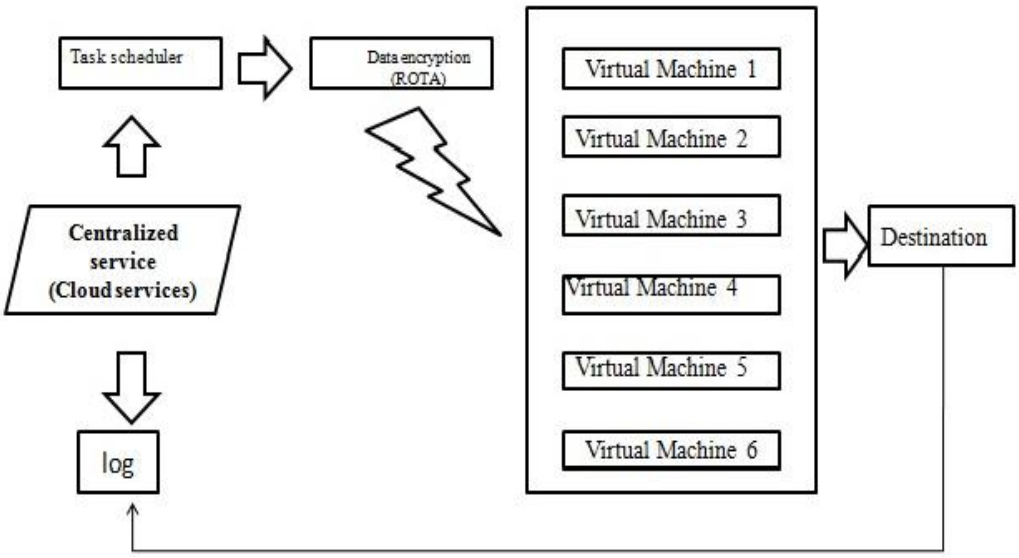

Figure 1: Over all flow and architecture

Assigning the task and scheduling is the common process in cloud computing. To allocate tasks proper resources to be utilized in service oriented architecture. Basically in the cloud environment multiple interconnected services will be interacting to share and process the data So service is which is decentralized data centre will maintain multiple virtual machines as pool. These virtual machines share the data before broadcasting and with encrypted format and shared and non distributed key methodology. At first each and every virtual machine will be allocated with fixed bandwidth and this bandwidth fluctuates depends on the communication and sharing the amount and type of the data.Once the data transmission is started the decentralized cloud or cloud service(data centre) will check all the virtual machine states with respect to bandwidth , wait time and finally type of the data. Once the cloud service fetches from the LOGwhich will be maintained cloud service , with continuous updating with each and every data transaction among the virtual machine till the data reaches to destination. Here based on the bandwidth and long waiting of the virtual machine(s) the tasks will be allocated with proper schedule time and proper data chunks sharing among them with secure manner.

IV. Existing Approach

Once the bandwidth allocated to all virtual machines by central cloud(cloud service). Client dashboard will choose the data and the data will be partitioned based on the random points and the bandwidths and waiting time is logged. Once the data is partitioned and allocated with all the virtual machines virtual machines transmits the data to the destination. Here the virtual machines will be allocated based on the bandwidth and there no proper encryption and distribution of the keys. Overall key is generated after the data slots allocated to virtual machines. But this key is shared among all which is with less security. Data slots/chunks are equally distributed and virtual machine waits to send the data depends on the waiting time. So data transmission delay will be more and no proper encryption with the data. No proper log to ignore duplicated transmission. So data duplication will be updated at the destination. So more time of upload and download is consumed.

V. Proposed Framework

STSASF is the approach proposed and here the cloud centre will check the bandwidth of each and every virtual machine and waiting time. So once the data is selected by client side the cloud centre will check the log to ignore the duplication of transmission. And the log will be checked with bandwidths and long waiting time to allocate chunks. Now data centre will partitions the data depends on the more bandwidth and more waiting time it allocate slots and OTP will be generated and once the transmission starts the OTP will be asked by dashboard and after checking the OTP the data will be encrypted using rota and will be broadcasted. So uploading and download time will be less and data is safe with encryption.

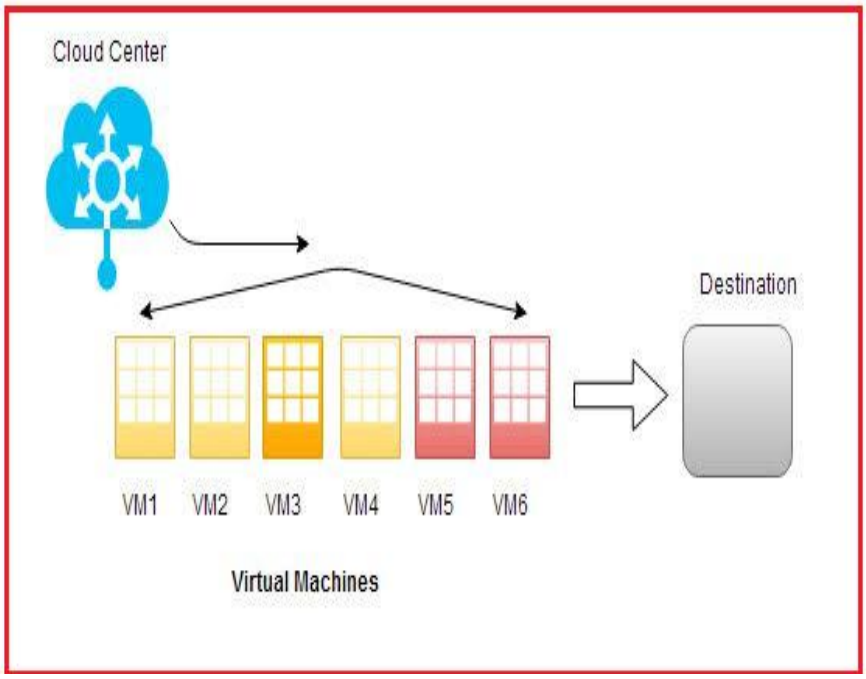

Figure 2: :Task scheduler

INPUT:

N <= 6 // number of virtual machines in cloud

$\sum Vm <= 6$ // virtual machines initialization

$\int_0^{n-1} Bw <= 1500\ MBPS$ //bandwidth allocation for all virtual machines

$\int Dc <= 0$ // data chunks

$\int L <= 0$ // log table

$T_s <= 0$ // task scheduling initialization

$B_c[T] <= 0$ // data broad cast

OUTPUT:

$T_s <= 0$ // scheduler

$B_c[T] <= 0$ // data broadcast

ALGORITHM:

// Start the virtual machine and bandwidth allocation

For i in 1 to n

START ($V_m(i)$)

ALLOCBW ($B_w(V_m(i))$)

End for

// Start broadcast

D_c = GETCHUNKS (V_m, $B_w(V_m)$) // Dc will be filled with data chunks depends on the bandwidth

T_s <= TASKSCH (D_c, V_m) // tasks will be allocated for each virtual machine

ENCR (Ts,Dc) // Encrypted the data

$B_c[i]$= BROAD CAST(D_c,V_m) // broadcast time

Figure 3**:** Task scheduler pseudo code

VI. Data Encryption Algorithm

ROTA(radix):This is a set of identical binary to strings(text) encoded models that indicates binary data with ASCII text format by converting/translating to RADIX-64 notations[13-17]. Normally rota models are widely used whenever there is a need of binary data encoding that needed to be encoded and transferred over the networks which are framed to deal with string based data. First time SangHoon [18] has proposed novel architectural idea to go for higher radix multiplier along with algorithmic approach. He has implemented a radix-64, 54x54 redundant binary parallel multiplier.This step is to ensure that data has to remain without alteration during broadcast. rota is ordinarily utilized as a part of a rota encoding takes the first double information and works on it by partitioning it into tokens of three bytes. A byte comprises of eight bits, so rota takes 24bits altogether. These 3 bytes are then changed over into four printable characters from the ASCII standard. The calculation's name rota originates from the utilization of these 64 ASCII characters. The ASCII characters utilized for rota are the numbers 0-9, the letter sets 26 lowercase and 26 capitalized characters in addition to two additional characters "+" and '/'.The initial step is to take the three bytes (24bit) of parallel information and split it into four quantities of six bits. Since the ASCII standard characterizes the utilization of seven bits, Base64 just uses 6 bits (comparing to 2^6 = 64 characters) to guarantee the encoded information is printable and none of the uncommon characters accessible in ASCII are utilized. The ASCII change of 3-byte, 24-bit gatherings is rehashed until the entire succession of unique information bytes is encoded. To guarantee the encoded information can be appropriately printed and does not surpass the farthest point. At the point when the quantity of bytes to encode is not distinct by 3 (that is, if there are just a single or two bytes of contribution for the last 24-bit square), then the accompanying activity is performed: Add additional bytes with esteem zero so there are three bytes, and play out the change to rota. On the off chance that there was just a single huge information byte, just the initial two base64 digits are picked (12 bits), and if there were two critical info bytes, the initial three base64 digits are picked (18 bits). "=" characters may be added to make the last piece contain four rota characters.

VII. Sample Example For Rota Algorithm :

Example:

Text content	***M***		***a***	***n***
ASCII	***77***		***97***	***110***
Bit pattern	***0 1 0 0 1 1 0 1 0 1 1 0 0 0 0 1 0 1 1 0 1 1 1 0***			
Index	***19***	***22***	***5***	***46***
Base64-encoded	***T***	***W***	***F***	***u***

Padding:

The '==' sequence indicates that the last group contained only 1 byte, and '=' indicates that it contained 2 bytes.

Exapmple1:

Input:
any carnal pleasure.
Output: YW55IGNhcm5hbCBwbGVhc3VyZS4=

Exapmple2:

Input:
any carnal pleasure
Output:
YW55IGNhcm5hbCBwbGVhc**3VyZQ==**

Figure4: Rota algorithm example flow

DATA ENCRYPTION ALGORITHM:ROTA(radix): This is a set of identical binary to strings(text) encoded models that indicates binary data with ASCII text format by converting/translating to RADIX-64 notations.Normally rota models are widely used whenever there is a need of binary data encoding that needed to be encoded and transferred over the networks which are framed to deal with string based data. This step is to ensure that data has to remain without alteration during broadcast. rota is ordinarily utilized as a part of a rota encoding takes the first double information and works on it by partitioning it into tokens of three bytes. A byte comprises of eight bits, so rota takes 24bits altogether. These 3 bytes are then changed over into four printable characters from the ASCII standard.

The calculation's name rota originates from the utilization of these 64 ASCII characters. The ASCII characters utilized for rota are the numbers 0-9, the letter sets 26 lowercase and 26 capitalized characters in addition to two additional characters "+" and '/'.

The initial step is to take the three bytes (24bit) of parallel information and split it into four quantities of six bits. Since the ASCII standard characterizes the utilization of seven bits, Base64 just uses 6 bits (comparing to 2^6 = 64 characters) to guarantee the encoded information is printable and none of the uncommon characters accessible in ASCII are utilized. The ASCII change of 3-byte, 24-bit gatherings is rehashed until the entire succession of unique information bytes is encoded. To guarantee the encoded information can be appropriately printed and does not surpass the farthest point.

At the point when the quantity of bytes to encode is not distinct by 3 (that is, if there are just a single or two bytes of contribution for the last 24-bit square), then the accompanying activity is performed: Add additional bytes with esteem zero so there are three bytes, and play out the change to rota. On the off chance that there was just a single huge information byte, just the initial two base64 digits are picked (12 bits), and if there were two critical info bytes, the initial three base64 digits are picked (18 bits). "=" characters may be added to make the last piece contain four rota characters.

```
Input  ⟶  raw string
Output  ⟶ base64 encoded format
Step1: initialization
?ALPHABET = H 25 0 CAPS(65 - 90) + H 25 0 SMALL(97 - 122) +
H 9 0 NUMBERS(48 - 57) + H 2 0 SPC(43,67)
// all "ALPHABET" CONTAINS ASCII values for capital letters, small letters, sing
digit numbers, '+' and '/' characters.
Step2:
Functionality:
To convert alphabets to ASCII codes
Input      ⟵   all available characters
Output     ⟵   all equivalent ASCII values
n          ⟵   0
B0         ⟵   0
B1         ⟵   0
B2         ⟵   0
H n 0 buff[]   ⟵   0
Ũ ar       ⟵   0
π          ⟵   0
Iteration  ⟵   0
Loop statement:
Count=0
For each C in ALPHABET
toInt (count) = TOINT (ALPHABET (i))
Count++
End loop
Size=SIZE (buff)
Iteration  ⟵   (((size+2)/3)*4)
Do while n in iteration
B0         ⟵   buff
B1         ⟵   (i <size)?  buff++: 0
B2         ⟵   (i < size)?  buff++: 0
Masking
Mask=0X3F
ar = ALPHABET [(b0>>2) & mask]
ar= ALPHABET [(b0<<4) | ((b1&0XFF)>>4)) & mask]
ar= ALPHABET [(b1<<2) | ((b2&0XFF)>>6)) & mask]
ar = ALPHABET [b2 & mask]
End while
Padding:
If (size % 3=1)
ar   ⟵   "="
ar   ⟵   "="
Else if (size % 3=2)
ar   ⟵   "="
else
size %3=0
End if
```

figure5:Rota algorithm pseudo code

VIII. Results and analysis

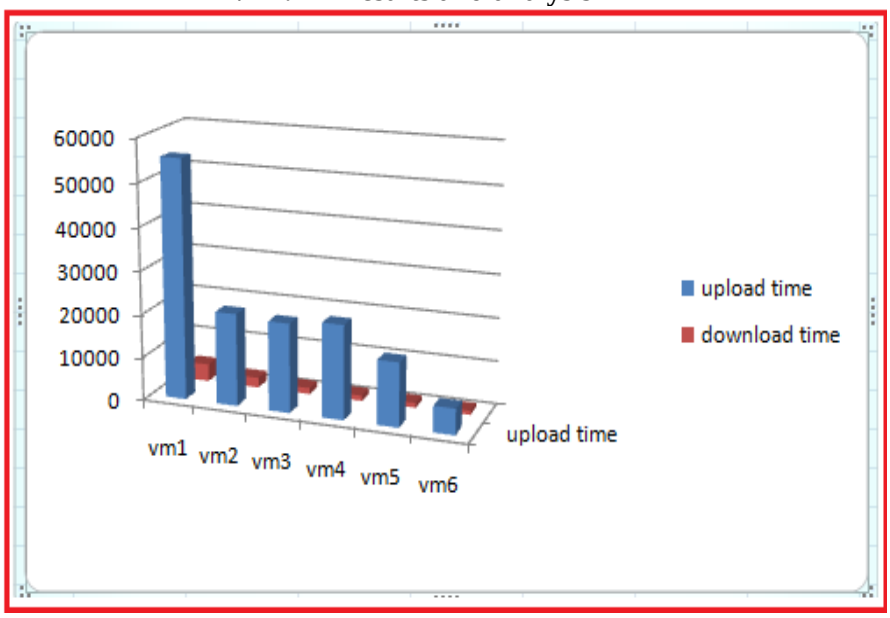

Figure 6: All virtual machines upload and download time after scheduling

Figure.6 shows the transactions for upload and download times with all the virtual machines. You can see the changes of each transaction with respect to all virtual machines of upload and download times. Vm6 consumed very less upload and download time and vm1 with high values. The allocation of bandwidth, RAM for each and every virtual machines showing in table 1

.

Virtual machine name	Bandwidth	Ram allocated
Vm1	789Mbps	43Gb
Vm2	2300mbps	75Gb
Vm3	2100Mbps	70Gb
Vm4	2500Mbps	80Gb
Vm5	4100Mbps	170Gb
Vm6	6780Mbps	200Gb

Table1: Bandwith and Ram allocated for Virtual machines

By observing the above table with Fig.6 vm6 is with high bandwidth to upload as well as download times. But these metrics will fluctuates each and every time with proper allocation from centralized service.

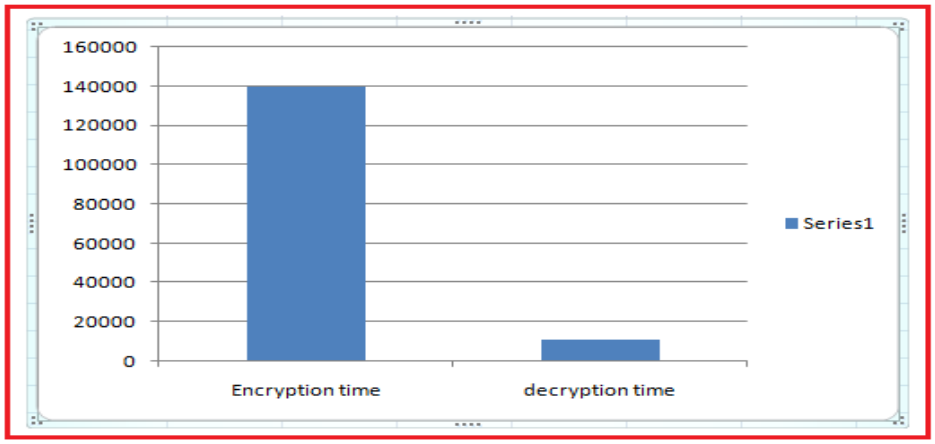

Figure 7**:**Encryption and decryption of consolidated data at dashboard and service side

Figure7 shows the data security of encryption and decryption at only service side and these virtual machines cannot be given access to data without encryption and decryption facilities. Here encryption time is

more coz all chunks from all virtual machines needs to be individual encryption and merged. But decryption is on single slot.

IX. Conclusion And Future Work

Resources have to be allocated for data transactions and data divisions with proper security. This work proposes STSASF(Smart Transaction in Services and Secured Framework) with 6 virtual machines communications in cloud environment. Based on the available bandwidth and long time waiting virtual machines will be identified by STSASF and will be allocated with tasks and scheduled for smooth data transactions. The data transactions would be in secured and maintained with proper log to avoid duplicate transactions. The switch mode is also proposed to start the free pool of virtual machines(6) at a time. So 6 virtual machines are in active state for data transactions. rota(radix) algorithm is proposed for secure transactions. depends on the capacities of the virtual machines STSASF will schedule thetask of allocating the data so we can access resources in a cloud in secure manner.

X. Future work

The extension and future of this work is to deploy the service as centralized service on fixed IP where that IP can be made as private IP so that external services can be adopted with this service to serve to multiple virtual machines concurrently in cluster model. This cluster model will clusters the data and checks the maximum waiting virtual machines to release the data. The main data will be shuffled randomly for allocation to relevant virtual machine before broadcasting. The encryption technique of the data would be DNA methodology. In security part, the knowledge of type and level of encryption required can save a considerable amount of computing because the process of encryption/decryption needs and overhead of computing. Providing unnecessary security to a task which does not require such level of security is just a waste of computing resources.

References

[1]. Luiz F. Bittencourt, Carlos R. Senna, and Edmundo R. M. Madeira, "Enabling Execution of Service Workflows in Grid/Cloud Hybrid Systems ",2010 IEEE/IFIP Network Operations and Management Symposium Workshops,pp.343-349.

[2]. Susanta, Nanda and Tzi-cker, Chiueh. "A survey on virtualization technologies," Experimental Computer Systems Lab, Feb 2005.

[3]. Rajkumar kannan, Raihan Ur Rasool, Haijin,Sr.R.Balasundaram " Managing and Processing Big Data in Cloud Computing"2016 pp.195-197.

[4]. T. L. Saaty, "How to make a decision: the analytic hierarchy process," Eur. J. Oper. Res., vol. 48, no. 1, pp. 9–26, 1990.

[5]. D.I. George Amalarethinam, T. Lucia Agnes Beena,"Customer Facilitated Cost-based Scheduling (CFCSC) in Cloud" International Conference on Information and Communication Technologies (ICICT 2014) Science Direct Published by Elsevier B.V.© 2015.

[6]. V. Khajehvand, H. Pedram, and M. Zandieh, "SCTTS: Scalable CostTime Trade-off Scheduling for Workflow Application in Grids. KSII Transactions On Internet and Information Systems,7(12), pp.3096-3117,2013

[7]. V. Dandhwani, and V. Vekariya, "Multi-Objective Task Scheduling using K-mean Algorithm in Cloud Computing. International Journal of Innovative Research in Computer and Communication Engineering, 4(11), pp.19521í19524, 2016.

[8]. P. Yue, X. Shengjun, L. Mengying, "An Improved Multi-Objective Optimization Algorithm Based on NPGA for Cloud Task Scheduling. International Journal of Grid and Distributed Computing, 9(4) (2016), pp.161-176.

[9]. H. Hua, X. Guangquan, P. Shanchen, and Z. Zenghua, "AMTS: Adaptive Multi-Objective Task Scheduling Strategy in Cloud Computing. China Communications 13(4), pp. 162–171, 2016.

[10]. Atul Vikas Lakra, Dharmendra Kumar Yadav, "MultiObjective Tasks Scheduling Algorithm for Cloud Computing Throughput Optimization" International Conference on Intelligent Computing, Communication & Convergence (ICCC- 2015)Conference Organized by Interscience Institute of Management and Technology,Bhubaneswar, Odisha, India Published by Elsevier B.V. © 2015.

[11]. Brototi Mondal, Kousik Dasgupta, Paramartha Dutta, "Load Balancing in Cloud Computing using Stochastic Hill ClimbingA Soft Computing Approach" Published by Elsevier B.V. © 2011.

[12]. Hao Yuan, Changbing Li and Maokang Du,"Cellular Particle Swarm Scheduling Algorithm for Virtual Resource Scheduling of Cloud Computing" International Journal of Grid Distribution Computing Vol. 8, No. 3, (2015), pp. 299-308 ISSN: 2005-4262 IJGDC Copyright Ⓒ 2015 SERSC.

[13]. Gensuke Goto, Atsuki Inoue, Ryoichi Ohe, Shoichiro Kashiwakura, Shin Mitarai, Takayuki Tsuru, and Tetsuo Izawa., "A 4.1 ns Compact 54X54 - B Multiplier Utilizing Sign-Select Booth Encoders", IEEE J. Of Solid-State Circuits, Vol.32, No.11, pp. 1676- 1682,Nov. 1997.
[14]. Wen-Changyeh and Chein-wei Jen, "High-speed Booth encoded parallel multiplier design" IEEE Transaction On Computers, Vol. 49,No. 7,pp. 692-701, July 2000.
[15]. Nario Ohkubu,Makoto Suzuki,Toshinobu shinbo "A 4.4 ns CMOS 54x54-b Multiplier using Pass-Transistor Multiplexer " IEEE J. Of Solid-State Circuits, Vol.30, No. 3, pp. 251-257,March 1995.
[16]. Gensuke Goto, Tomio Sato, Masao Nakajima, And Takao Sukemura, "A 54X54 Regular Structured Tree Multiplier", IEEE J. Of Solid State Circuits, Vol.27, No.9, pp. 1229-1236, September 1992.
[17]. Luigi Ciminiera And Paolo Mantuschi, "Carry Save Multiplication Schemes Without Final Addition" IEEE Transaction On Computers, Vol. 45,No. 9,pp. 1050-1055, September 1996. [7] P. Song And G.De Micheli, "Circuit And Architecture Tread Off For High Speed Multiplication" IEEE J. Of Solid State Circuits, Vol.2, No.9, November 1997.
[18]. Sang-Hoon Lee,Seung-Jun BAE,Hong-June PARK, "A Compact Radix-64 54x54 CMOS Redundant Binary Parallel Multiplier" IEICE Trans. Electron., Vol. E85-C, No. 6, pp.1342-1350, June 2002.

Prof. Ramesh Babu Inampudi Smart Transaction in Services through Secured Framework for task scheduling in cloud environment." IOSR Journal of Computer Engineering (IOSR-JCE) , vol. 19, no. 6, 2017, pp. 59-67.