List Scheduling in Cloud Computing with Trust Based Resource Selection

Pankaj Sareen[#1], Parveen Kumar[#2], Dr. Tripat Deep Singh[#3]
#1 Assistant Professor, SPN College Mukerian
#2 Assistant Professor, SPN College Mukerian
#3 Assistant Professor, SPN College Mukerian
[1]pankaj.sareen.mca@gmail.com
[2]parveenspn20@gmail.com
[3]tripatdua@yahoo.com

Abstract- **Cloud computing is network based computing using the internet, which is utility based, on demand computing with each client using other systems hardware/software/infrastructure in a cloud environment accessed through closed network. Cloud computing is computing service delivery over the Internet. Cloud services allow people and Businesses to use software/hardware managed by third parties in remote locations. Cloud computing Ensures access to information and computer resources from wherever a network connection is available. List scheduling is fundamental to achieving cloud computing efficiency. Most list scheduling Cloud computing methods consider task resource requirements for CPU and memory and not bandwidth requirements. List scheduling creates a jobs list based on priorities and the highest priority job is executed by assigning it to a suitable resource till a valid/optimal schedule is found. As many services are provided by unknown parties/enterprises, this study proposes a trust based model and reputation based scheme to select suitable resources to improve tasks scheduling performance in a cloud environment.**

Keywords-**Cloud Computing, Trust based resource selection, list Scheduling, Gravitational Search algorithm (GSA)**

I. INTRODUCTION

Cloud Computing is Internet Computing with resources being organized like clouds in the internet where end users access resources via Internet from anywhere and for any duration without knowledge of actual resources maintenance and management. All resources in clouds are dynamic and scalable. Cloud computing ensures sharing of resources and common infrastructure to offer services to users, so that operations meet applications needs. Cloud computing provides resources on demand by applications as per user need and services collect payment from end users according to resource usage. Some advantages of cloud computing are, improved performance at lower cost, reduced infrastructure and maintenance cost with increased computing power and unlimited storage capacity. Its limitations include its needing constant Internet connection, slow computing and cloud data may not be secure.

Resource allocation and scheduling are challenging in cloud computing as different cloud computers vary in resources and capacity. Some physical resources are processor, disk, bandwidth and special devices. Resource allocation depends on job requirements and users preferences. Jobs are distributed to remote computational nodes selected on parameters like computational power, line quality and network bandwidth. A computational node's QoS is represented by cost, completion time, reliability and network bandwidth. Many scheduling algorithms are available in literature. Selecting the best is complicated and hence existing scheduling algorithms are tailored to fit cloud environment.

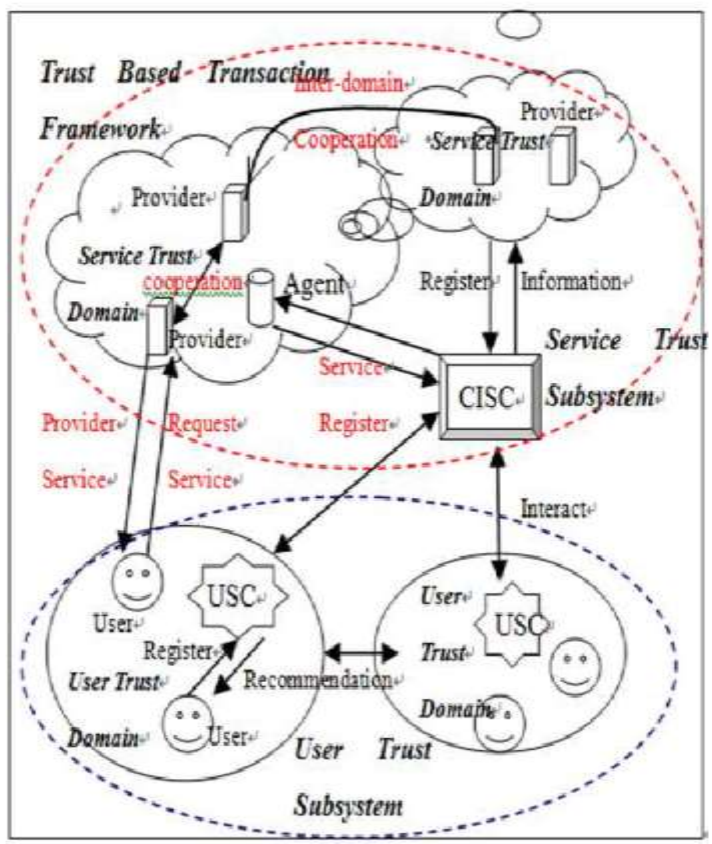

Fig1. Trust Based Transaction Framework on cloud Computing

Cloud environment scheduling has 3 stages including resource discovering and filtering, Resource selection and Task submission.

Job scheduling algorithms are categorized into 2 groups in a cloud environment. They are Batch Mode Heuristic scheduling Algorithms (BMHA) and On-line Mode Heuristic Algorithms (OMHA). Jobs which arrive at the cloud environment are queued temporarily and form a set, and scheduling starts after a predefined time period in BMHA. Traditional scheduling algorithms like Round Robin (RR) scheduling algorithm, First Come First Served (FCFS) scheduling algorithm, Min–Min algorithm and Max–Min algorithm are BMHA algorithms. In OMHA scheduling, jobs are scheduled when they arrive at the system. As cloud computing environments have heterogeneous systems with dynamic services, OMHA scheduling algorithms suit cloud environments. Most fit Task Scheduling is an OMHA scheduling algorithm.

First come first service algorithm uses jobs arrival order to make the schedule. This algorithm is simple and fast. Round robin algorithm dispatches jobs like FCFS, but each is given to a processor fora predefined time period. Jobs needing limited time to execute are dispatched first to reduce small tasks waiting time in Min-Min algorithm. In Max-Min algorithm, job needing longer execution time are dispatched first to reduce large tasks waiting time.

Each job is assigned a priority either internally or externally in priority based scheduling and jobs are dispatched based on priority. Jobs with equal priority follow FCFS order. External priorities are set by user and internal priorities by the job's measurable quantities. Shortest job First (SJF) is a special internal priority scheduling algorithm. Priority is based on CPU burst of all jobs in queue. Most Fit task scheduling algorithms dispatch jobs which fit the queue best as first job, but this algorithm has a high failure ratio. Resource aware and reliability based scheduling algorithms are used in cloud systems. Presently, many optimization algorithms combine with traditional scheduling algorithms to tailor scheduling algorithms to suit cloud environments scheduling. Many scheduling algorithms consider trade-offs between cost and task execution time. Such algorithms assume that all cloud services are reliable when in reality some service providers are dishonest and malicious. If cloud environment is un-trust, then scheduling is uncertain.

Developing a model to measure trust minimizes uncertainty among open distributed system's computing nodes like grid and cloud environments. Use of trust in scheduling improves reliability and robustness in schedule. Reputation methods provide the computing systems past behavior details which help decide the computing system's trust. Rating mechanism is a method which uses user's feedback. Weighted rating gives varied weightage to feedback according to end user. Reputation based scheduling calculates progress score for every job execution in a computing node, which is considered the computing node's reputation. Maintaining records of progress, scores over a long duration ensuring scheduling decisions. The decision avoids failure prone nodes and time consuming computing nodes when scheduling.

List scheduling creates a job list by assigning priorities and executes the highest priority job by assigning a resource till a valid schedule is found. During selection, if suitable resource is not found, then the next job in the list is selected. Some lists scheduling algorithms are highest level first algorithm, critical path method, largest path algorithm, and heterogeneous earliest finish time scheduling for heterogeneous environment. This study proposes a list scheduling algorithm for cloud environment. For resources selection, trust based model is resorted to as the resource is heterogeneous and dynamic in cloud environment.

According to a current research carried out by Right Scale, Infrastructure as a Service (IaaS) grew 45 percent from 2013 to 2014. Currently it is estimated to be a $15 billion business and it expected to grow to about $31 billion by 2015 while in 2013, Software as a Service (SaaS) was a $13 billion market, but is now predicted to grow to over $30 billion by 2016. In 2008, revenue from worldwide cloud services was $46.4 billion; in 2014, it is expected to reach $150 billion, a jump of just over 225 percent In addition, a survey carried out by Avanade, businesses have increased investments in resources to secure, manage, and support Cloud Computing.

It's been some time that the term cloud computing is being in the existence. But still it emerges as a new technology as lot of research is being progressed in this field. Cloud Computing provides an illusion of infinite available resources [4]. These resources are available with the resource owner (Res_O). These resources include Processing Elements (PEs), RAM, memory and bandwidth (BW).

II. EXECUTION OF THE TASK IN CLOUD COMPUTING

The Various steps are:

i. User assigns the task to be executed over cloud.
ii. The task is received by cloud coordinator (CC).
iii. Cloud coordinator forwards the task over datacenters (DC).
iv. Datacenters contains number of unfixed hosts consisting of pool of virtual machines (VM).
v. These hosts can be configured or deleted as per the demand.

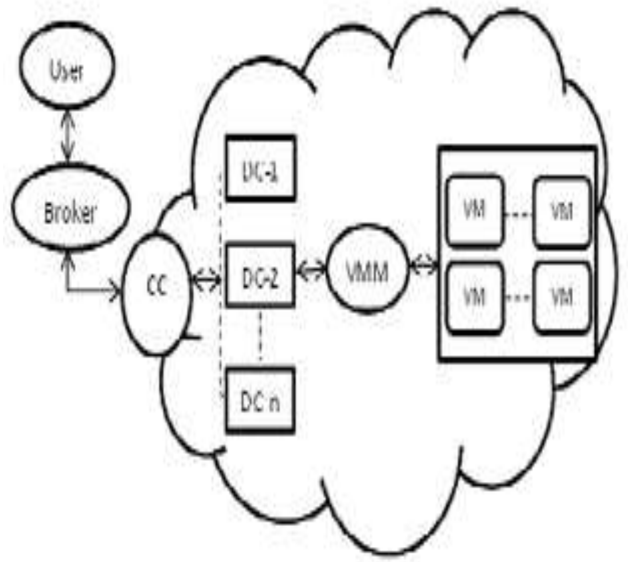

Fig.2 Task execution over cloud

A. Allocation of resources for execution of task:

i. VMM asks for resources from the resource provisioner by sending the task requirements.
ii. Resource provisioner checks the availability of resources with Resource Owner.
iii. If the resources are available, the resource owner grants the access permission to use the resources to resource provisioner.
iv. Resource provisioner further provides access of the resources for creation of virtual machines.

III. RESOURCE CLOUD COMMUNICATION PARADIGM

Fig. 3 shows how the resources in RCCP are allocated to VMM.

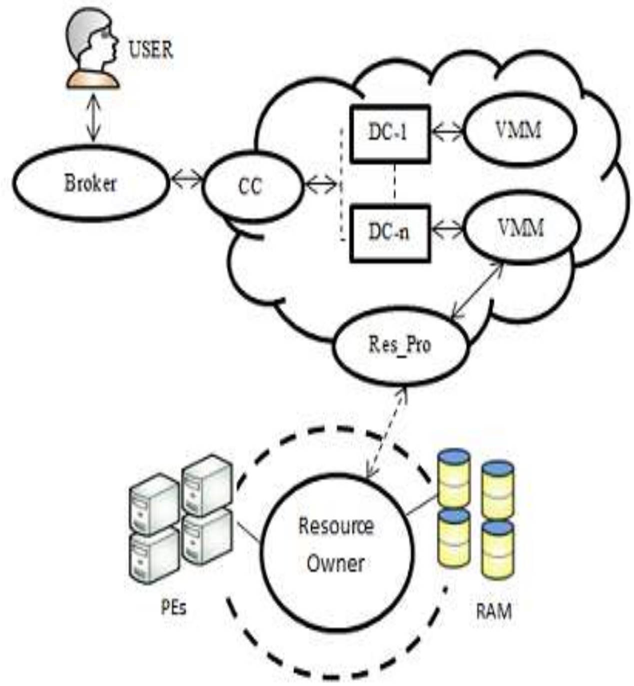

Fig. 3 Resource Cloud Communication Paradigm

A. Steps in Allocating Resources within RCCP

i. Res Owners provide details of available resources to *Resource Provisioner (Res_Pro)*.
ii. Res_Pro creates a ***Resource Allocation List (RAL)*** containing details of available resources of Res Owners.
iii. VMM asks for resources from the Res_Pro by sending the requirements.
iv. Res_Pro checks the RAL.
v. If resources are available in RAL, then Res_Pro provides access to resources.
vi. If resources are not available in RAL, then Res_Pro checks availability of resources with Res Owners.
vii. Res Owners, in return, sends resource availability acknowledgements to Res_Pro.
viii. Res Owner selects the resources considering the cost factor.
ix. Res_Pro requests for the selected resources from Res Owner.
x. Res Owner provides access to resources to Res_Pro.
xi. Res_Pro updates the RAL after getting access to resources.

xii. Res_Pro then provide access to VMM.

B. Sequence Diagram illustrating resource allocation from Res Owners to VMMs

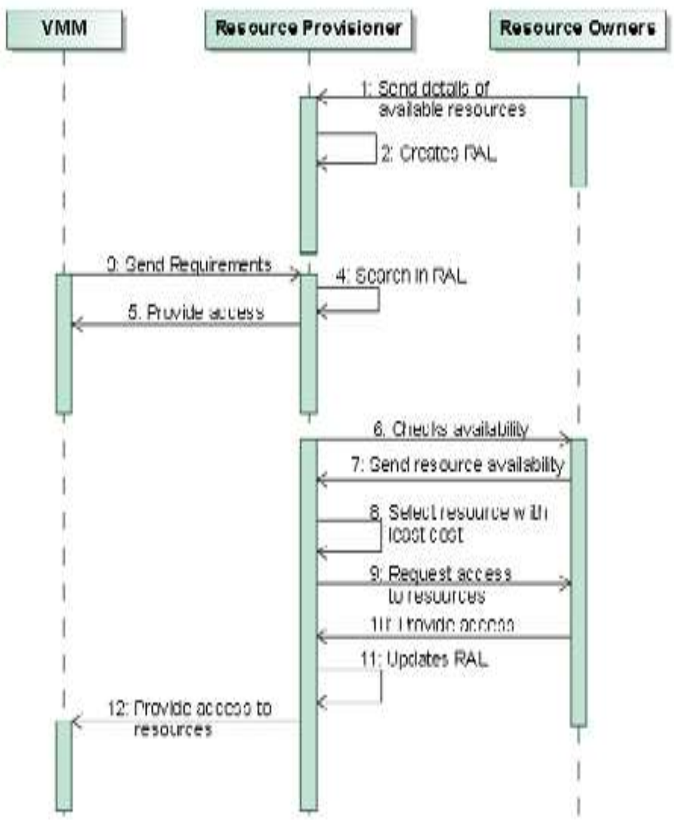

Fig. 4. Sequence diagram for Resource Allocation Process

IV. LIST SCHEDULING

A. Stages of List Scheduling process

Scheduling is used to allocate particular resources for a certain tasks in particular time. Task scheduling problem is a core and challenging issue in cloud computing. The Task execution time cannot be predicted in cloud computing. Hence the scheduler must be dynamic. The purpose of scheduling is to increase the utilization of resources.

Scheduling process in cloud can be generalized into three stages namely:

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. This is a deciding stage.
iii. *Task submission:* Task is submitted to resource selected.

B. Materials and Methods

Branch and Bound algorithm (BB) provides list scheduling. It usually gives an optimal schedule which cannot be prepared in polynomial time. Hence, heuristics based methods are combined to get an optimal schedule within polynomial time. List scheduling works with a Data Dependency Directed Acyclic Graph (DDD). In DDD, nodes represent operations and edges represent data dependencies between two operations. Each edge is given minimum and maximum timing associated with it which represent between 2 operations and dependences to form a constraint of scheduling. Data Ready Set (DRS) has all operations ready to be scheduled. An operation is data ready, when all operations it depends on are scheduled. From the DRS, the list scheduler finds the next operation for scheduling, based on a heuristic choice.

Pseudo code for traditional list scheduling algorithm is given below:

Input: DDD representing meta-block operations to be scheduled. DRS containing operations with no predecessors for each operation, the earliest and latest it may be scheduled.

Output: Instruction Schedule corresponding to input DDD.

Algorithm

```
While DRS not empty
        Heuristically select best node from DRS
        Scheduled = FALSE
         Compute_Schedule_Range (operation)
       current_instruction=operation. Earliest
      While (current_instruction≤operation.latest AND
Not Scheduled
                if no.conflictsv(operation.Current_instruction)
then
                Schedule(operation.current_instruction)
                        Scheduled = TRUE
       else
                    current_instruction = next_instruction
if (Not Scheduled)
                    Compaction Failed
   Update successors Timings
   Update Data Ready Set
```

Generally Trust is used to establish/maintain relationship between two entities for a long time. Applying trust models to scheduling decreases failure ratio and reassigning in cloud environments. Combining communication trust and data trust locates a component/resource/service's overall trust while scheduling. Bayesian fusion algorithm computes overall resources trust. Here direct trust and indirect trust formed by recommendations of trusted components find a component's overall trust. The proposed algorithm's flow chart is given in figure 5.

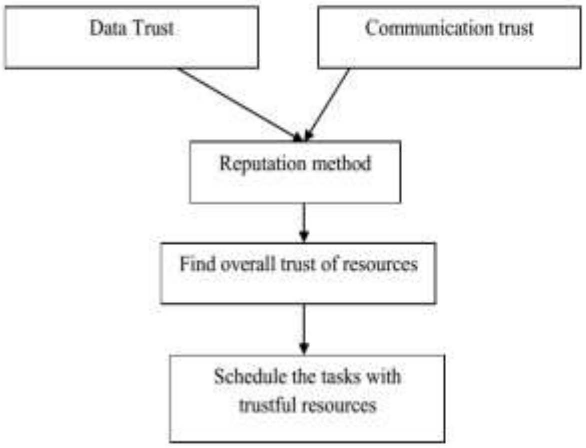

Fig. 5. Flow chart for the algorithm

Data trust decide resources list to be considered to calculate the trust/threshold levels to separate trustful and untrusted nodes. Communication trust is calculated on client's bandwidth availability and resource centers. For fusion of data and communication trusts, Bayesian model is used. Reputation ratings are calculated by beta reputation based on probability density functions given by,

$$f(p|\alpha,\beta)=\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}p^{\alpha-1}(1-p)^{\beta-1}$$

Where alpha represents number of jobs completed and beta represents unsuccessful jobs. Rij is reputation for a resource ni observed from neighbourhood resources nj.

$$R_{ij}=Beta\left(\alpha_{ij}+1,\beta_{ij}+1\right)$$

Then trust value is calculated using expected value of reputation.

$$T_{ij}=E\left(R_{ij}\right)=E\left\{Beta\left(\alpha_{ij}+1,\beta_{ij}+1\right)\right\}=\frac{\left(\alpha_{ij}+1\right)}{\left(\alpha_{ij}+\beta_{ij}+2\right)}$$

Reputation is always updated by new alpha and beta values.

$$R_{ij}^{new}=Beta\left(\alpha_{ij}^{new},\beta_{ij}^{new}\right)$$

New communication trust is updated by following formulae,

$$\alpha_{ij}^{new}=\alpha_{ij}+\frac{2*\alpha_{ik}*\alpha_{kj}}{\left(\beta_{ik}+2\right)*\left(\alpha_{kj}+\beta_{kj}+2\right)+\left(2*\alpha_{ik}\right)}$$

$$\beta_{ij}^{new}=\beta_{ij}+\frac{2*\alpha_{ik}*\beta_{kj}}{\left(\beta_{ik}+2\right)*\left(\alpha_{kj}+\beta_{kj}+2\right)+\left(2*\alpha_{ik}\right)}$$

$$T_{ij}^{new}=E\left(R_{ij}^{new}\right)=E\left\{Beta\left(\alpha_{ij}^{new}+1,\beta_{ij}^{new}+1\right)\right\}=\frac{\left(\alpha_{ij}^{new}+1\right)}{\left(\alpha_{ij}^{new}+\beta_{ij}^{new}+2\right)}$$

Data trust is calculated from distributions of mean and error reports variance about a resource observed in clouds. Data trust reputation is calculated by the following formulae,

$$\begin{aligned}R_{i,j}&=N\left(\mu_{i,j},\sigma_{i,j}^{2}\right)\\T_{i,j}&=\Pr ob\left\{\left|\theta_{i,j}\right|<\varepsilon\right\}\\&=\Pr ob\left\{-\varepsilon<\theta_{i,j}<+\varepsilon\right\}\\&=\phi\left(\frac{\varepsilon-\mu_{i,j}}{\sigma_{i,j}}\right)-\phi\left(\frac{-\varepsilon-\mu_{i,j}}{\sigma_{i,j}}\right)\end{aligned}$$

where ∅is cumulative probability distribution used to map trust value within range $[-\varepsilon, \epsilon]$ and $\mu i,j$ and $\sigma i,j$ are mean and error variance generated by component ni and observed by component nj.

$$\mu_{i,j}\ \frac{\left(\mu_{0}/\sigma_{0}^{2}\right)+\left(k\bar{y}_{i,j}/\tau^{2}\right)}{\left(1/\sigma_{0}^{2}\right)+\left(k/\tau^{2}\right)}$$

$$\sigma_{i,j}^{2}=\frac{1}{\left(1/\sigma_{0}^{2}\right)+\left(k/\tau^{2}\right)}$$

Where k is number of reported errors of computing node ni observed from node nj and τ is known error value. Mean and variance new values are updated by the following formula.

$$\mu_{ij}^{new}=\frac{\left(\mu_{0}/\sigma_{0}^{2}\right)+\sum_{s=1}^{m}\frac{\left(\mu_{I_{s},j}+\mu_{i,I_{s}}\right)}{\left(\frac{1}{T_{i,I_{s}}}-1\right)\alpha}+\left(k\bar{y}_{i,j}/\tau^{2}\right)}{\left(1/\sigma_{0}^{2}\right)+\sum_{s=1}^{m}\frac{1}{\left(\frac{1}{T_{i,I_{s}}}-1\right)\alpha}+\left(k/\tau^{2}\right)}$$

$$\sigma_{ij}^{2new}=\frac{1}{\left(1/\sigma_{0}^{2}\right)+\sum_{s=1}^{m}\frac{1}{\left(\frac{1}{T_{i,I_{s}}}-1\right)\alpha}+\left(k/\tau^{2}\right)}$$

New trust value between node ni and nj is updated by,

$$T_{i,j}^{new} = \phi\left(\frac{\varepsilon - \mu_{i,j}^{new}}{\sigma_{i,j}^{new}}\right) - \phi\left(\frac{-\varepsilon - \mu_{i,j}^{new}}{\sigma_{i,j}^{new}}\right)$$

V. TRUST BASED RESOURCE SELECTION

CloudSim software is used for simulation with twenty five tasks assigned to Cloud with 15 resources. Each resource has 1 cpu with 256 Mb RAM. Each task is of size between 1 and 9 units. Trust based method for the selection of resources is used in scheduling. The execution time of these tasks is compared with non-trust based resource selection. Results are shown graphically in the following figure 6.

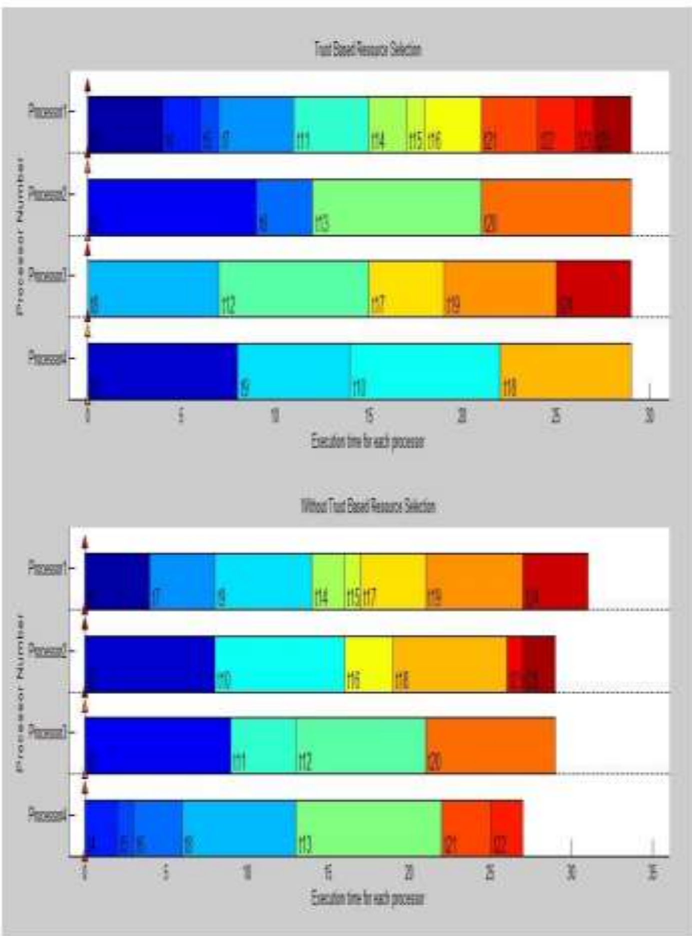

Fig. 6. Execution time for trust based and without trust based scheduling

From the above figure, it is observed that the trust based scheduling reduced the total execution time of given jobs up to 10 seconds.

VI. Advantages and Limitations of Resource Allocation Strategies

A. Advantages:

i. The first major benefit of resource allocation is that user neither has to install software nor hardware to access the applications, to develop the application and to host the application over the internet.

ii. The next major benefit is that there is no limitation of place and medium. We can reach our applications and data anywhere in the world, on any system.

iii. The user does not need to expend on hardware and software systems.

iv. Cloud providers can share their resources over the internet during resource scarcity.

B. Limitations:

i. Since users rent resources from remote servers for their purpose, they don't have control over their resources

ii. Migration problem occurs, when the users wants to switch to some other provider for the better storage of their data**.** It's not easy to transfer huge data from one provider to the other.

iii. In public cloud, the clients' data can be susceptible to hacking or phishing attacks. Since the servers on cloud are interconnected, it is easy for malware to spread.

iv. Peripheral devices like printers or scanners might not work with cloud. Many of them require software to be installed locally. Networked peripherals have lesser problems.

v. More and deeper knowledge is required for allocating and managing resources in cloud, since all knowledge about the working of the cloud mainly depends upon the cloud service provider.

In Appendix A, various resource allocations strategies and their impact are listed

VII. CONCLUSION

Cloud computing is a high performance computing environment with heterogeneous, large scale, autonomous systems and flexible computational architecture collection. Applying a trust model on scheduling decreases task failure numbers, so that a task's reassignment and restart is unnecessary. This study combines resources communication trust and data trust to find a component/resource/service's overall trust while scheduling. Bayesian fusion algorithm computes resources trust. Both direct and indirect trust formed by recommendations of trusted components finds a component's overall trust. Reputation based method updates trust value dynamically. CloudSim software is used to simulate with 25 tasks assigned to Cloud from 15 resources. Performance evaluation is through execution time.

Results revealed that total execution time is reduced in trust based scheduling significantly.

REFERENCES

[1] B.ThirumalaRaoand Dr.L.S.S.Reddy, "Survey on Improved Scheduling in HadoopMapReduce in Cloud Environments ", International Journal of Computer Applications (0975 – 8887) Volume 34– No.9, November 2011.

[2] samAzawiMohialdeen, "Comparative Study Of Scheduling Algorithms In Cloud Computing Environment" , Journal of Computer Science, 9 (2): 252-263, 2013.

[3] Li, J., M. Qiu, J. Niu, W. Gao and Z. Zonget al., 2010. Feedback dynamic algorithms for preemptable jobs cheduling in cloud systems. Proceedings of the International Conference on IEEE Web Intelligence and Intelligent Agent Technology, August 31.

[4] Elghoneimy, E., Bouhali, O., &Alnuweiri, H. (2012, January). Resource allocation and scheduling in cloud computing. In Computing, Networking and Communications (ICNC), 2012 International Conference on (pp. 309-314). IEEE.

[5] Song, X., Gao, L., & Wang, J. (2011, June). Job scheduling based on ant colony optimization in cloud computing. In Computer Science and Service System (CSSS), 2011 International Conference on (pp. 3309-3312). IEEE.

[6] Liu, H., Xu, D., & Miao, H. (2011, December). Ant colony optimization based service flow scheduling with various QoS requirements in cloud computing. In Software and Network Engineering (SSNE), 2011 First ACIS International Symposium on (pp. 53-58). IEEE.

[7] Arora, S., & Singh, S. (2013, August). A conceptual comparison of firefly algorithm, bat algorithm and cuckoo search. In Control Computing Communication & Materials (ICCCCM), 2013 International Conference on (pp. 1-4). IEEE.

[8] Long, S., & Zhao, Y. (2012, December). A toolkit for modeling and simulating cloud data storage: an extension to CloudSim. In Proceedings of the 2012 International Conference on Control Engineering and Communication Technology (pp. 597-600). IEEE Computer Society.

[9] Li, X., Jiang, X., Ye, K., & Huang, P. (2013, June). DartCSim+: Enhanced CloudSim with the Power and Network Models Integrated. In Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on (pp. 644-651).IEEE.

[10] A.Meera, S.Swamynathan, "Agent based Resource Monitoring system in IaaS Cloud Environment", International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA), 2013

[11] Weiwei Lin, James Z. Wang, Chen Liang, and Deyu Qi, "A Threshold-based Dynamic Resource Allocation Scheme for Cloud Computing", Procedia Engineering volume 23, 2011, Pages 695–703

[12] K C Gouda, Radhika T V, Akshatha M, "Priority based resource allocation model for cloud computing", International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 1, January 2013

[13] Ewnetu Bayuh Lakew, Francisco Hernandez-Rodriguez, Lei Xu and Erik Elmroth, "Management of Distributed Resource Allocations in Multi-cluster Environments", Performance Computing and Communications Conference (IPCCC), IEEE 31st International, 2012

[14] Diptangshu Pandit, Matangini Chattopadhyay, and Nabendu Chaki, "Resource Allocation in Cloud using Simulated Annealing", Applications and Innovations in Mobile Computing (AIMoC), Feb. 27 2014 – March 1 2014, Pages 21-27

[15] A.Singh, M.Korupolu and D.Mohapatra. Server-storage virtualization:Integration and Load balancing in data centers. In Proc.2008 ACM/IEEE conference on supercomputing (SC'08) pages 1- 12, IEEE Press 2008.

[16] AndrzejKochut et al.: Desktop Workload Study with Implications for Desktop Cloud Resource Optimization, 978-1-4244-6534-7/10 2010 IEEE.

Appendix A

S.No	Resource Allocation Strategy	Impacts
1	Based on the estimated execution time of job .(Advanced Reservation, Best effort and immediate mode)	Estimation may not be accurate. If job could not finish its execution in estimated tine, it will affect the execution of other jobs.
2	Matchmaking strategy based on Any-Schedulability criteria.	Strategy mainly depends upon the user estimated job execution time of a job.
3	Based on role based security policy.	Follows decentralized resource allocation.
4	Most Fit Processor Policy.	Requires complex searching process and practical to use in real system.
5	Based on cost and speed of VM.	Allows the user to select VM.
6	Based on the load conditions specified by the user.	Instances of resources can be added or removed.
7	Based on gossip protocol (resources allocated by getting information for other local nodes)	It used decentralized algorithm to compute resource allocation and this prototype is not acceptable for heterogeneous cloud environment.
8	Utility function as a measure of profit based on live VM migration.	Focused on scaling CPU resources in IaaS.
9	Based on the utility function as a measure of price.	Allocate resources only in the lowest level of cloud

		computing and considered only CPU resource.
10	Utility function as a measure of response time.	Lacks in handling dynamic client requests.
11	Based on utility function as a measure of application satisfaction.	Relies on two-tier architecture.
12	Based on the CPU usage of VM, active user requests are served. Adaptively new VM spawns, when the CPU usage reaches some critical point.(VR)	There is a limitation in the number of concurrent user monitor and the prototype is not capable of scaling down as the number of active user decreases.
13	Based on hardware resource dependency.	Considered only CPU and I/O resource.
14	Auction mechanism.	Not ensure profit maximization.
15	Based on online resource demand predication.	Prediction may not be accurate and leads to over provisioning or under provisioning.
16	Based on workflow representation of the application.	The application logic can be interpreted and exploited to produce an execution schedule estimate. Again estimation may not be accurate.
17	Based on the machine learning technique to precisely make decisions on resources.	This prototype reduces the total SLA cost and allocate resources considering the both the request rates and also the weights.
18	Simulated annealing algorithm.	Lacks in handling dynamic resource request.
19	Based on constant needs of client and GPS.	Solution can be improved by changing the resource allocation and lacks in handling the large changes in parameters.
20	Network game theory approach.	Lack in dynamic cooperative organization formation

