TRUST BASED RESOURCE SELECTION AND LIST SCHEDULING IN CLOUD COMPUTING

V. Suresh Kumar[1], M. Aramudhan[2]
[1]Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, Tamil Nadu, India
[2]Associate Professor, PKIET, Karaikal, India

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 platforms hide underlying infrastructure's complexity and fine details from end users by providing simple Graphical User Interface (GUI) or Applications Programming Interface (API). 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: *List Scheduling, Cloud Computing, Trust based resource selection*

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 [1]. Resource's/device's location is unknown to network's end user. Users can also develop/manage cloud applications with the cloud making resources virtualization by maintaining/managing itself. Some important cloud computing [2] properties are as follows:

1) **User-centric computing**: When a new user is connected to cloud, documents, images and applications stored by new user are shared with other cloud users and data and devices of others are shared with the new user simultaneously.
2) **Task-centric computing**: Cloud computing focuses on the user's need, and how existing applications satisfy user's need.
3) **Powerful computing**: As cloud computing connects thousands of computers to form clouds, huge computing power is used for applications. When data is needed in cloud environment, users get data from multiple repositories simultaneously; hence users are not limited to a single source.
4) **Intelligent Computing**: As data is accessed from multiple data repositories, analysis and data mining gets information in intelligent way.
5) **Programmable**: Some cloud computing tasks are automatic. To maintain data integrity information in one computer is replicated with many in cloud. When a computer is offline/ disconnected, an automatic program redistributes available data to cloud computers.

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.

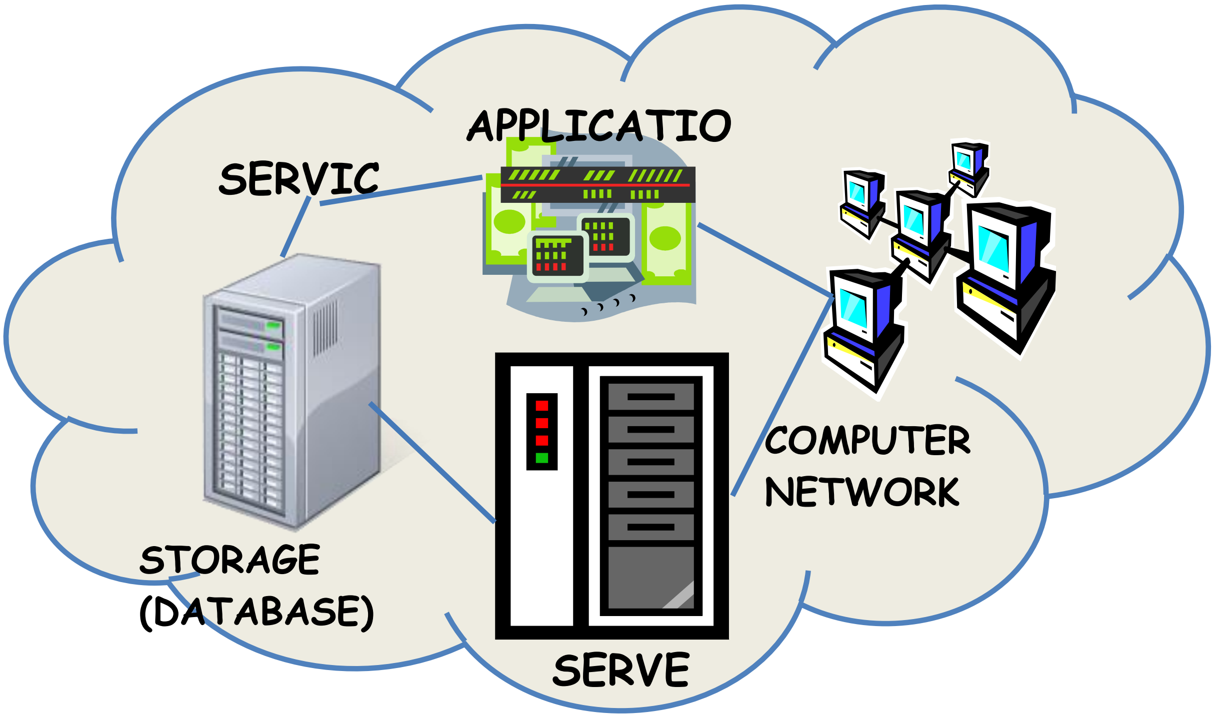

Figure 1: Basic structure of a cloud

Virtualization makes functionalities abstraction and isolation at lower level and underlying hardware enable functions portability at higher levels by physical resources [3]sharing and/or aggregation. Computing platform in clouds is by a virtual machine which helps users complete jobs cost effectively in reasonable time without affecting Quality of Service (QoS).

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 [4]. 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. 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) [5]. 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 for a 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 [6]. Resource aware [7] and reliability based [8] 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 [9]. 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 behaviour 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 [10] 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.

The rest of this study is organized as follows: Section 2 showcases related works in literature. Section 3 describes methods used in the proposed work; Section 4 talks about experiments and obtained results and Section 5 provide the conclusion.

II. RELATED WORKS

Hadoop is a parallel processing framework hiding processing nodes implementation and distribution, starting tasks, restarting failed tasks and consolidation of results. An improved scheduling algorithm for Hadoop Map Reduce in cloud environments was proposed by Raoand Reddy[11]. The authors implemented dynamic proportional scheduler, delay scheduler, and resource aware scheduler and deadline constraint scheduler for homogeneous computing nodes. Each scheduler's Pros and Cons were analyzed. Many scheduling algorithms were proposed in literature which increased complexity in selecting the best algorithm for adoption in a cloud environment.

Mohialdeen [12] studied 4 different scheduling algorithms like Round Robin, Random resource selection, Opportunistic load balancing time and minimum completion time algorithm for clouds list scheduling. These scheduling algorithms results were analyzed by QoS parameters and fairness in jobs allocation. Cloudsim simulator was used and results evaluated by parameters like makespan, throughput and total cost. It was seen that minimum completion time algorithm produced high throughput and reduced makespan with highest cost than others.

Developing service oriented infrastructure in cloud systems ensured computational resources to remote users. Parallel processing in cloud environments reduced jobs execution time. To improve resource use Li, et al., [13] suggested pre-emptive job scheduling. Dynamic scheduling algorithms were combined with feedback mechanism. Experiments with the proposed scheduling algorithm revealed that feedback improved scheduling performance especially in situations where there was resource contention.

Power consumption at resource centers/servers was critical in scheduling in a computing environment. Server work load consolidation and shutting off the machine when idle were undertaken to reduce power consumption, but workload consolidation was a NP-hard problem. Dynamic round robin scheduling for energy efficient virtual machine scheduling in cloud environments was proposed by Lin, et al., [14]. Greedy, round robin and power aware scheduling was implemented by authors and results compared with dynamic round robin scheduling. It revealed that dynamic round robin scheduling reduced power consumption compared to other 3 scheduling algorithms.

An efficient scheduling algorithm and resources management to resource use and minimizing total execution cost was proposed by Paul and Sanyal [15]. Credit based scheduling was through use of cost matrix generated by fairness of a task to be assigned to a specific resource. Fairness calculation is based on job arrival and resource waiting time. Cloud computing is utility computing where users do not need all resources at their site. They can acquire resources from other sites and pay as per usage. Scheduling algorithms based on tasks computational complexity and remote systems processors was proposed by Sindhuand Mukherjee [16]. Experiments were conducted with CloudSim simulator and these algorithms performed better for heavy loads. Issues in scheduling algorithms for cloud environment were addressed by Yang, et al., [17]. In heavy load situations no best scheduling algorithm considers clouds status. Also no mechanism existed to detect resource failures and recovery. The authors suggested a scheduling algorithm with reinforcement learning to increase fault tolerance and maximize resources use for a long duration in cloud environments.

A trust model in grid environment scheduling algorithms was proposed by XuandQu [18]. As many applications in real world are data-intensive, data being transferred increases task scheduling overhead. The authors suggested a Min-Min scheduling algorithm with trust based model which selected tasks based on the file server's trust degree and data transmission time. To calculate transmission time to remote resource, shortest path algorithm like dijikstra was used. The results showed that this algorithm improved task completion and completion time success rate in grid environments.

As a single cloud's capacity is limited, applications access other cloud's computational capacity over the internet. A resource collaborative scheduling to improve virtual resources use in cloud computing environment was proposed by Lu, et al., [19]. A virtual environment's available resource credibility is calculated as trust. Fuzzy linguistic representation represents trust by three dimensions which are system trust, user trust and collaboration trust. Malicious computing nodes were located through resources reputation improving the virtual organization's credibility.

An optimal workflow based scheduling for cloud computing environments was proposed by Tan, et al., [20]. Work flow scheduling ensured high clouds performance, but many cloud services were offered over the internet by third-party organizations. To solve services uncertainty and increase reliability, the authors suggested a Trust services-oriented Multi-Objectives Workflow Scheduling (TMOWS) model for cloud's work flow scheduling. The authors provided suggestions to optimize cost and execution time through a case study.

A trusted dynamic level scheduling algorithm to reduce task assignments failure probability and job execution in a security environment was proposed by Wang and Zeng [21]. Bayesian model found resources trust degree in cloud computing environment. Simulation showed that task execution failure rate reduced with increased time and cost. Dynamic trust scheduling in cloud environments was extended by Wang, et al., [22] through developing a new direct trust based model and the recommended trust from trusted resources/systems. When jobs were submitted to a cloud environment, they were in a queue and scheduler communicates with the advisor. The latter communicates with middleware which analyzes local transactions and trust models to find the cloud environment's most trustful resources.

Various work flow based scheduling algorithms useful for large applications like e-business and e-science were analyzed by Bardsiriand Hashemi [23]. These were distributed applications needing specialized tools for work in clouds. Meta-heuristic algorithms like Particle Swarm Optimization (PSO), Ant colony Optimization and hybrid optimization algorithms were analyzed and compared to conventional scheduling algorithms. Most of such algorithms aimed to meet deadlines and budget constraints. A reputation based work flow scheduling for grid computing nodes was proposed by Rahman, et al., [24]. As peer to peer networks were decentralized and large scale computing resources were available, resources might be unreliable. Resource reliability in grids was calculated by statistical measures on feedbacks/scores from resources users via grid service brokers. This reputation based scheme was dynamic considering changes in resources/services. Using grid environment traces, simulations showed that makespan was lowered up to 50 % compared to non-reputation based scheduling algorithms.

Cloud scheduling as a multi criteria decision making problem was proposed by Lawrance, et al., [25]. Potentially All Pair-wise Rankings of All Possible Alternatives (PAPRIKA) was used for QoS based resource scheduling. Jobs given to clouds were ordered according to QoS requirements and scheduled by PAPRIKA. The algorithm was simulated by CloudSim with results showing that PAPRIKA reduced jobs completion time through increased resource use.

III. 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 [27] is given in the following.

```
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
                        ifno.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 [26]. 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 2.

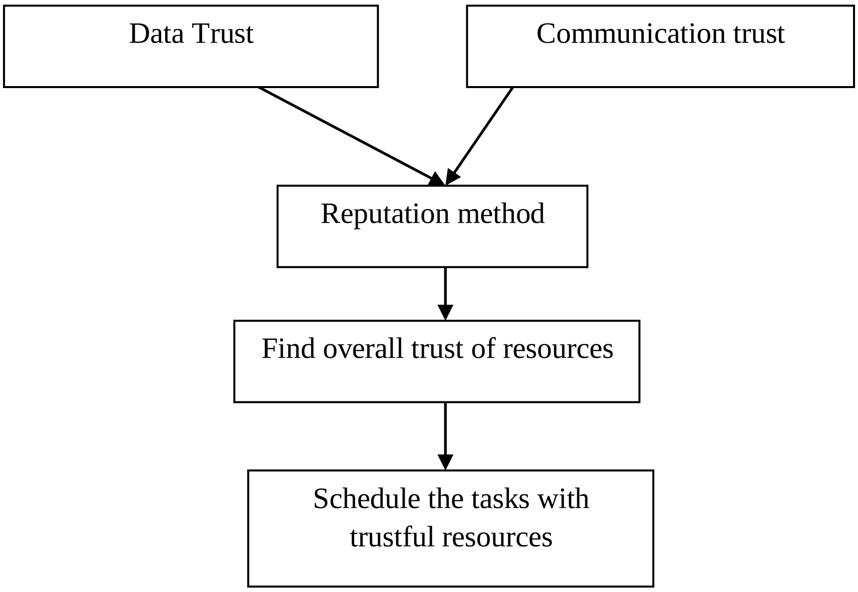

Figure 2: Flow chart

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,

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

where alpha represents number of jobs completed and beta represents unsuccessful jobs. R_{ij} is reputation for a resource n_i observed from neighbourhood resources n_j.

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

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,

$$R_{i,j} = N\left(\mu_{i,j}, \sigma_{i,j}^2\right)$$

$$T_{i,j} = \Pr ob\left\{\left|\theta_{i,j}\right| < \varepsilon\right\}$$

$$= \Pr ob\left\{-\varepsilon < \theta_{i,j} < +\varepsilon\right\}$$

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

where $\emptyset$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 n_i and observed by component n_j.

$$\mu_{i,j} \frac{\left(\mu_0 / \sigma_0^2\right) + \left(k\overline{y}_{i,j} / \tau^2\right)}{\left(1/\sigma_0^2\right) + \left(k / \tau^2\right)}$$

$$\sigma_{i,j}^2 = \frac{1}{\left(1/\sigma_0^2\right) + \left(k / \tau^2\right)}$$

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

$$\mu_{ij}^{new} = \frac{\left(\mu_0 / \sigma_0^2\right) + \sum_{s=1}^{m} \frac{\left(\mu_{l_s,j} + \mu_{i,l_s}\right)}{\left(\frac{1}{T_{i,l_s}} - 1\right)\alpha} + \left(k\overline{y}_{i,j} / \tau^2\right)}{\left(1/\sigma_0^2\right) + \sum_{s=1}^{m} \frac{1}{\left(\frac{1}{T_{i,l_s}} - 1\right)\alpha} + \left(k / \tau^2\right)}$$

$$\sigma_{ij}^{2new} = \frac{1}{\left(1/\sigma_0^2\right) + \sum_{s=1}^{m} \frac{1}{\left(\frac{1}{T_{i,l_s}} - 1\right)\alpha} + \left(k / \tau^2\right)}$$

New trust value between node n_i and n_j is updated by,

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

IV. EXPERIMENTS AND RESULTS

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 3.

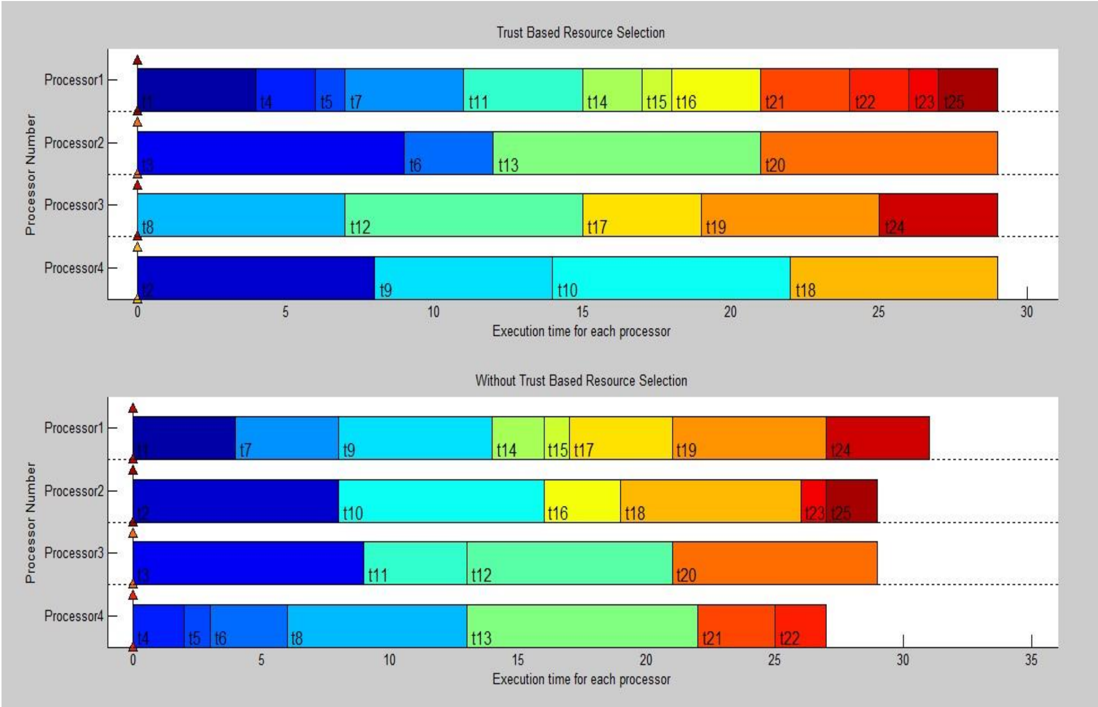

Figure 3: Execution time for trust based and without trust based scheduling

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

V. CONCLUSION

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] SrinivasaRao V, NageswaraRao N K, and E KusumaKumari, "CLOUD COMPUTING: AN OVERVIEW ", Journal of Theoretical and Applied Information Technology, 2008.
[2] Shivaji P. Mirashe and Dr. N.V. Kalyankar, "Cloud Computing", Journal Of Computing, Volume 2, Issue 3, March 2010.
[3] Mladen A. Vouk, "Cloud Computing – Issues, Research and Implementations", Journal of Computing and Information Technology - CIT 16, 2008, 4, 235–246.
[4] Mayank Mishra, Anwesha Das, PurushottamKulkarni, and AnirudhaSahoo, "Dynamic Resource Management Using Virtual Machine Migrations", Sep 2012, 0163-6804/12, IEEE Communications Magazine, page no: 34-40.
[5] Vignesh V, Sendhil Kumar K S, and Jaisankar N, " Resource Management and Scheduling in cloud Environment **",** International Journal of Scientific and Research Publications, Volume 3, Issue 6, June 2013.
[6] PinalSalot, "A Survey Of Various Scheduling Algorithm in Cloud Computing Environment", IJRET, FEB 2013, Volume: 2 Issue: 2.
[7] SaeedParsa and Reza Entezari-Maleki," RASA: A New Task Scheduling Algorithm in Grid Environment" in World Applied Sciences Journal 7 (Special Issue of Computer & IT): 152-160, 2009.Berry M. W., Dumais S. T., O'Brien G. W. Using linear algebra for intelligent information retrieval, SIAM Review, 1995, 37, pp. 573-595.

[8] ArashGhorbanniaDelavar, Mahdi Javanmard ,MehrdadBarzegarShabestari and MarjanKhosraviTalebi "RSDC (Reliable Scheduling Distributed In Cloud Computing)" in International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.2, No.3, June 2012.
[9] Babanov A, Collins J, Gini M. Asking the right question: Risk and expectation in multi agent contracting. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 2003; 17(3):173–186.
[10] Tung Nguyen, and Weisong Shi, "Improving resource efficiency in data centers using reputation-based resource selection ", Sustainable Computing: Informatics and Systems , Elsevier, March, 2012.
[11] 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.
[12] IsamAzawiMohialdeen, "Comparative Study Of Scheduling Algorithms In Cloud Computing Environment" , Journal of Computer Science, 9 (2): 252-263, 2013.
[13] Li, J., M. Qiu, J. Niu, W. Gao and Z. Zonget al., 2010. Feedback dynamic algorithms for pre emptable jobscheduling in cloud systems. Proceedings of the International Conference on IEEE Web Intelligence andIntelligent Agent Technology, August 31.
[14] Lin, C.C., P. Liu and J.J. Wu, 2011. Energy-aware virtual machine dynamic provision and scheduling for cloud computing. Proceedings of the 4th International Conference on Cloud Computing, Jul.4-9, IEEE Xplore Press, Washington, DC., pp: 736- 737.
[15] Paul, M. and G. Sanyal, 2011. Task-scheduling in cloud computing using credit based assignment problem.International Journal of Computer Science and Engineering, 3: 3426-3430.
[16] Sindhu, S. and S. Mukherjee, 2011. Efficient task scheduling algorithms for cloud computing environment. Communications and Computing Information Science, Springer, 169: 79-83.
[17] Yang, B., X. Xu, F. Tan and D.H. Park, 2011. An utilitybased job scheduling algorithm for cloud computingconsidering reliability factor. Proceedings of the 2011 International Conference on Cloud and ServiceComputing, Dec. 12-14, IEEE Xplore Press, Hong Kong, pp: 95-102.
[18] YujiexXu and WenyuQu, "A Trust Model-Based Task Scheduling Algorithm for Data-Intensive Application ", Proceedings in the Sixth annual conference on IEEE chinagrid, 2011.
[19] Kun Lu, Hua Jiang, Mingchu Li, and Sheng Zhao, " Resources Collaborative Scheduling Model Based on Trust Mechanism in Cloud", IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2012.
[20] Wenan Tan, Yong Sun, Guangzhen Lu, Anqiong Tang, and LinShan Cui, "Trust Services-Oriented Multi-Objects Workflow Scheduling Model for Cloud Computing", Pervasive computing and Networks World, Springer, Volume 7719, 2013, pp 617-630.
[21] Wang, W., and Zeng, G. S. (2010). Dynamic trust evaluation and scheduling framework for cloud computing, ACSA 2010 (Security and Communication Networks), Gwangju, Korea, December 9–11.
[22] Wei Wang, GuosunZengDaizhong Tang,and Jing Yao," Cloud-DLS: Dynamic trusted scheduling for Cloud computing ",Expert Systems with Applications, Elsevier, Volume 39, 2012, pp 2321–2329.
[23] Amid KhatibiBardsiri and Seyyed Mohsen Hashemi, "A Review of Workflow Scheduling in Cloud Computing Environment ", International Journal of Computer Science and Management Research Vol 1 Issue 3 October 2012.
[24] Mustafizur Rahman , Rajiv Ranjan, and RajkumarBuyya , " Reputation-based dependable scheduling of workflow applications in Peer-to-Peer Grids ",Computer Networks Elsevier, Volume 54 (2010), pp 3341–3359.
[25] Hilda Lawrance et al.."EfficientQos Based Resource SchedulingUsing PAPRIKA Method for CloudComputing ", International Journal of Engineering Science and Technology (IJEST), Volume 5 No.03 March 2013.
[26] Mohammad Momani, SubhashChalla and Rami Alhmouz ," Bayesian Fusion Algorithm forInferring Trust in Wireless Sensor Networks" , JOURNAL OF NETWORKS, VOL. 5, NO. 7, JULY 2010.
[27] Michael J. Bourke , " Frequency-Based List Scheduling: An Extension of List Scheduling to Incorporate Frequency Information", Technical Report.

AUTHOR'S BIOGRAPHY

V Suresh Kumar is M.Tech in Computer Engineering. Currently doing his PhD in Computer Engineering at MS University under the guidance of Dr.M Aramudhan. Prof V Suresh Kumar has more than 15 years of experience in the field of Computer Engineering. His research area is Cloud computing. He is currently working as Dean, Engineering in SNGIST, N.Paravoor, Kerala.