

An Improved Scheduling Strategy in Cloud Using Trust Based Mechanism

D. Sumathi, P. Poongodi

Abstract—Cloud Computing refers to applications delivered as services over the internet, and the datacenters that provide those services with hardware and systems software. These were earlier referred to as Software as a Service (SaaS). Scheduling is justified by job components (called tasks), lack of information. In fact, in a large fraction of jobs from machine learning, bio-computing, and image processing domains, it is possible to estimate the maximum time required for a task in the job. This study focuses on Trust based scheduling to improve cloud security by modifying Heterogeneous Earliest Finish Time (HEFT) algorithm. It also proposes TR-HEFT (Trust Reputation HEFT) which is then compared to Dynamic Load Scheduling.

Keywords—Software as a Service (SaaS), Trust, Heterogeneous Earliest Finish Time (HEFT) algorithm, Dynamic Load Scheduling.

I. INTRODUCTION

CLOUD computing enables ubiquitous, convenient, on-demand network, access a shared pool of configurable computing resources (networks, storage, servers, applications, and services). The resources are rapidly provisioned and released with minimum management effort or service provider interaction [1]. Various scheduling algorithms exist in cloud computing in a distributed computing system. Most can be applied in a cloud environment with verifications. The job scheduling algorithm's advantage is achieving high-performance computing and best system throughput. Traditional job scheduling algorithms cannot ensure scheduling in cloud environments. According to a simple classification, job scheduling algorithms in cloud computing are categorized into two groups; Batch Mode Heuristic scheduling Algorithms (BMHA) and online mode heuristic algorithms. In BMHA, jobs are queued and collected in a set when they arrive in the system. The scheduling algorithm starts after a fixed period of time. BMHA based algorithms are Round Robin (RR) scheduling algorithm, First Come First Served (FCFS) scheduling algorithm, Min-Min algorithm and Max-Min algorithm. Jobs are scheduled when they arrive in the system [2] by on-line mode heuristic scheduling algorithms

Scheduling in cloud is generalized into 3 stages namely [3]

- 1) Resource discovering and filtering - Datacenter Broker discovers resources in a network system and collects status information about them.
- 2) Resource selection - Based on certain parameters of task

and resource, target resource is selected. This is deciding stage.

- 3) Task submission - Task submitted to selected resource.

Security issues for cloud computing arise as it encompasses many technologies including databases, networks, operating systems, resource scheduling, virtualization, transaction management, load balancing, concurrency control and memory management. Security issues for many systems and technologies are applicable to cloud computing. The network that interconnects systems in a cloud must be secure [4].

Data Issues

Sensitive cloud computing data emerges as a major issue regarding security in cloud-based systems.

Secrecy Issues

A cloud computing service provider must ensure that customer's personal information is safe from other providers, customers, and users. As most servers are external, a cloud service provider should know who is accessing data and who maintains the server so that the provider protects a customer's personal information.

Infected Application

Cloud computing service provider should have total access to the server with all rights for monitoring and maintenance. This prevents a malicious user from uploading infected application onto a cloud which severely affects both customer and cloud computing service.

Security Issues

Cloud computing security must be at two levels. One is at the provider level, and the other is at user level. A cloud computing service provider should ensure that the server is secured from external threats it comes across [5].

Trust is a complex concept which has no universally accepted definition. "Trust is a psychological state, comprising of an intention to accept vulnerability, based on positive expectations of the intentions or behavior of another." [6] Trust establishes and maintains relationship between two entities for a long time. Applying trust models to scheduling lowers failure ratio and reassigning in cloud environments. Combining communication trust and data trust ensures a component or resource or service's overall trust while scheduling. Bayesian fusion algorithm computes overall resources trust [7]. Data trust decide resources list to calculate trust and threshold levels to separate trusty and untrustworthy nodes. Communication trust is calculated on client's bandwidth availability and resource centers. Bayesian model

Sumathi D is with Jayam College of Engineering and Technology, Dharmapuri, Tamilnadu, India (e-mail: sumathid.clo@gmail.com).

Poongodi P is with V.S.B. Engineering College, Karur, Tamilnadu, India (e-mail: poongodi.vsb@rediffmail.com).

is used for fusion of data and communication trusts [8].

Trusting in an individual is due to many analyses that generate a trust definition. Trust (or distrust) is a specific subjective probability level; an agent believes another agent or agents group will perform a specific action, which goes through a monitoration (independent of its ability to monitor) and where it affects its action [9]. Trust is a complex issue in a cloud-computing arena. Companies like Google and Amazon have implemented reputation based trust management system, and it helps users locate trustworthy resource providers for e-business securely and confidently. e-bay has a built-in centralized trust model [10].

Load balancing enhances resources, utilizing parallelism, exploiting throughput improvisation, to reduce response time through an appropriate application distribution. Load balancing is static and dynamic. In case of static load balancing, the decision regarding allotment of a task is known in advance [11]. In dynamic load balancing, task allotment decisions are taken at runtime. Dynamic load balancing improves systems performance through workload or requests distribution from various clients to different servers. Dynamic load balancing strategy's goal in cluster based systems is finding suitable servers to serve incoming requests without heavily loading servers by enhancing acceptance rate and reducing starvation period [12].

This study identified a Trust based scheduling to improve cloud security, and proposed modified HEFT algorithm and TR-HEFT (Trust reputation HEFT). Both are compared to Dynamic Load Scheduling. Section II identifies literature on the domain and section III identifies the methodology. Section IV describes the results and discussions, and section V concludes the paper.

II. RELATED WORKS

A scheduling method based on a model of time-cost-trust cloud computing resources, based on the subset tree algorithm was presented by [13]. This method meets users' concern for resources reliability. The algorithm has high efficiency and suits cloud computing resource scheduling.

A new cloud scheduling scheme using Service Level Agreement with trust monitor to ensure faster scheduling was presented by [14]. They also considered the problem of over flooding user request and to provide secure request processing. The proposed scheme provides advantages and better performance for both cloud user and service provider.

An approach to manage trust in Cloud IaaS was proposed by [15]. Trust enhances secure and reliable scheduling in Distributed, Grid, and Cloud environment. Scheduling is done for better resources allocation and to enhance the QoS provided to users.

A resources collaboration scheduling model to improve virtual resources collaboration scheduling efficiency proposed by [16], is based on virtual organization. It uses trust mechanism to estimate virtual organization credibility. Simulation results proved that the model analyzed trust and reputation of resources and improved virtual organization credibility.

Scheduling problem for workflow applications was formulated by [17] with trust constraints and a novel scheduling algorithm based on trust was presented. Results illustrated that the new heuristic scheduling algorithm was better than traditional algorithms for scheduling application workflows.

A trust into workflow's QoS target was presented by [18] which proposed a new customizable cloud workflow scheduling model. They introduced trust mechanism into multi-workflow scheduling and single workflow scheduling levels. Experiments showed that the new schema achieved relatively high execution success rate and user satisfaction compared to other solutions.

A new Bayesian method based cognitive trust model was proposed by [19]. They also proposed a trust Dynamic Level Scheduling (DLS) algorithm called Cloud-DLS by integrating the existing DLS algorithm. A benchmark is made to span a range of Cloud computing characteristics for its evaluation. Theoretical analysis and simulations proved that Cloud-DLS algorithm meets Cloud computing workloads requirement in trust.

The issue of trust management in multi-cloud environments based on a set of distributed Trust Service Providers (TSPs) was addressed by [20]. TSPs are distributed over clouds and elicit raw trust evidence from various sources and formats. Experiments showed that the new framework was effective and relatively stable, in differentiating between trustworthy and untrustworthy CSPs in multi-cloud environments.

An overview of literature and a comparative analysis of SLA monitoring regarding trust maintenance in cloud computing was presented by [21]. Maintaining trust in cloud computing is a challenge due to its dynamic nature and trust fragility. Trust is established through successful transactions and meeting all SLA parameters between two interacting parties.

A model on a statistical property based on reliability and reputation combined with a "trust" based design was presented by [22]. This algorithm handles QoS and is better than the current model. This paper proposes a near-optimal CQR based scheduling policy and minimum energy heuristics to locate a trust based Cloud character probability model that exploits heterogeneity over multiple data centers for a Cloud provider.

The cloud's open, distributed, and dynamic nature posing challenges for trust establishment between entities was presented by [23]. In such applications, the consumer usually knows little about trust or the service provider's reliability. A novel trust and reputation management system for cloud computing and WSNs integration was proposed by [24]. Considering Cloud Service Users (CSUs) and Cloud Service Providers (CSPs) attribute requirement and the cost, trust and reputation of CSP service and Sensor Network Providers (SNP), the new system achieves 2 goals: 1) calculating and managing trust and reputation regarding CSP and SNP service; 2) helping CSU choose CSP and assisting CSP in choosing SNP.

A trust mechanism-based task scheduling model for cloud computing was proposed by [25]. Referring to social persons

trust relationship models, trust relationship is built among computing nodes and nodes trustworthiness is evaluated using the Bayesian cognitive method. Integrating nodes trustworthiness into a DLS algorithm, it proposes the trust DLS algorithm for cloud computing.

Abbadi and Alawneh [26] focused to identify the related challenges for establishing trust in a Cloud, and then proposing a foundation framework which addresses the identified challenges. This paper focuses on IaaS Cloud type and organizations as Cloud users.

Trust models that are proposed or implemented in Distributed and Grid environment was presented by [27] as to whether they fully fit cloud computing environments. A suitable trust model is proposed based on an existing model that suits trust value management for cloud IaaS parameters. A scheduling algorithm that further enhances QoS provided to users is also proposed

III. METHODOLOGY

Scheduling is a set of policies to control the order of work performed by a cloud. Cloud environment scheduling policies vary depending on the cloud's deployment model. Scheduling manages availability of cloud and resource grid, and good scheduling policy ensures maximum resource use. Job scheduling is a major activity in computing environments. Cloud computing has tremendous capabilities and to use limitless capabilities well, efficient scheduling algorithms are needed, as cloud computing is an upcoming technology. Scheduling strategy is the key to cloud computing [28]. A cloud scheduling process is summarized in 3 stages:

Resource discovery and filtering: Network system resources are discovered, and information about them is collected.

Resource selection: Target resource is selected based on task and resource parameters.

Task submission: Task to be executed is submitted to selected resource.

Heterogeneous Earliest Finish Time (HEFT) does not consider estimates of one task for locally optimal decisions but looks ahead in the schedule and considers information that affects the task's children by the decisions made. It is represented as a directed acyclic graph, on a bounded number of heterogeneous machines. Many different options for computing HEFT weights are considered. A weight is allocated to each node and graph edge, based on average computation and communication in HEFT. The graph at that point traverses upwards, and each node is assigned a rank value. Tasks are then scheduled, in order of rank value, on a machine with earliest finish time.

HEFT is an application scheduling algorithm for a bounded number of heterogeneous processors. It first constructs a priority tasks list and then makes locally optimal allocation decisions for each task based on a task's estimated finish time. Efficient scheduling aims to map tasks to the core processors and execution order is set so that task precedence requirements are satisfied with a minimum schedule length. HEFT algorithm is an effective solution for DAG scheduling problem on heterogeneous systems due to its robust

performance, low running time, and ability to ensure stable performance over various graph structures. The algorithm's limitation is that it uses static approaches to a mapping problem that assume static conditions for a period of time. Also, in complex situations it fails to find optimal scheduling [29].

The HEFT algorithm first calculates average execution time for a task and average communication time between two successive tasks resources. Let time (T_i, r) be execution time of task T_i on resource r and let R_i be the set of available resources to process T_i . Average execution time of task T_i is defined as

$$\bar{\omega} = \frac{\sum_{r \in R_i} \text{time}(T_i, r)}{|R_i|} \quad (1)$$

Let time (e_{ij}, r_i, r_j) be data transfer time between resources r_i and r_j which process tasks T_i and T_j respectively. Let R_i and R_j be the set of available resources to process T_i and T_j respectively. Average transmission time T_i to T_j is defined by:

$$\bar{c}_{ij} = \frac{\sum_{r_i \in R_i, r_j \in R_j} \text{time}(e_{ij}, r_i, r_j)}{|R_i| |R_j|} \quad (2)$$

Then tasks in a workflow based on a rank function are ordered in HEFT. For an exit task T_i , its rank value is:

$$\text{rank}(T_i) = \bar{\omega}_i \quad (3)$$

Rank values of other tasks are computed as:

$$(T_i) = \bar{\omega}_i + \max_{T_j \in \text{succ}(T_i)} (\bar{c}_{ij} + \text{rank}(T_j)) \quad (4)$$

where, $\text{succ}(T_i)$ is a set of immediate successors to task T_i . The algorithm then sorts a task by decreasing order of rank values. A task with higher rank value gets higher priority. In resource selection phase, tasks are scheduled according to priorities and a task is assigned to a resource that finishes it at the earliest [30].

HEFT algorithm

- 1: compute average execution time for a task $t \in \Gamma$ according to (1)
- 2: compute average data transfer time between tasks and successors according to (2)
- 3: compute rank value for a task according to (3) and (4)
- 4: sort tasks in a scheduling list Q by decreasing order of task rank value
- 5: while Q is not empty do
- 6: $t \leftarrow$ remove first task from Q
- 7: $r \leftarrow$ find a resource which completes t at earliest
- 8: schedule t to r .
- 9: end while

Dynamic Level Scheduling (DLS) algorithm is a compile time, static list scheduling heuristic to allocate a DAG-structure application to a set of heterogeneous machines to reduce application execution time [31], [32]. At every scheduling step, the DLS algorithm chooses a next task to

schedule and on which machine the task is to be executed by finding the ready task and machine pair with the highest dynamic level. A task machine's dynamic level, (v_i, m_j) is defined as

$$DL(v_i, m_j) = SL(v_i) - \max\{t_{i,j}^A, t_{i,j}^M\} + \Delta(v_i, m_j) \quad (5)$$

where $SL(v_i)$ is called static level of a task, $\{t_{i,j}^A, t_{i,j}^M\}$ is time when task v_i can begin execution on machine m_j , $t_{i,j}^A$ denotes time when data is available if task v_i is scheduled on machine m_j , and $t_{i,j}^M$ denotes a time when machine m_j is available for task execution, $v_i \cdot \Delta(v_i, m_j) = t_{i,j}^E$ reflects computing performance of a machine, $t_{i,j}^E$ denotes execution time of task v_i on all free machines, and $t_{i,j}^E$ denotes execution time of task v_i on machine m_j .

When making a decision to schedule, DLS algorithm considers heterogeneous machines, which adapt resources heterogeneous characteristics in a Cloud environment, but it neglects resource nodes trustworthiness in a Cloud system. When a task is scheduled to be executed on a machine, nodes trustworthiness reflects the reliability of the service it supplies. To offset this, a trust-DLS algorithm in Cloud environment (Cloud-DLS) is developed, and trust dynamic level is defined as:

$$TDL_s(v_i, n_j) = T_s(v_i, n_j)^{\alpha_i} * (SL(v_i) \max\{t_{i,j}^A, t_{i,j}^M\} + \Delta(v_i, n_j)) \quad (6)$$

where $T_s(v_i, n_j)$ is trustworthiness evaluation of n_j when v_i is scheduled by n_s on n_j , equal to $\hat{\theta}$ discussed above. α_i is a QoS factor of v_i , satisfying $0 \leq \alpha_i \leq 1$ and $\sum \alpha_i = 1$. To one task machine pair (v_i, n_j) , when α_i is increased, it means requirements of task v_i in trust is increased and scheduling priority lowered accordingly. Thus, the algorithm is scalable and meets different QoS requirements. By adjusting α_i , users' different trust requirements are satisfied.

Trusted DLS algorithm is implemented as middleware to plug into a Cloud system, by which tasks are executed on trust nodes efficiently. On one hand, the failure task execution ratio is reduced; on the other, data executive environment security is improved. This provides a basic integrated framework based on Cloud-DLS (Fig. 1). This framework has 4 tiers: the first is the resource and infrastructure tier including network, cluster or Cloud computing infrastructure, the second is the basic middleware for Cloud computing, the third is the trustworthy scheduler; and the last is the client. Schedule Advisor in a trustworthy scheduler is developed based on Cloud-DLS, and a Trust Model is based on a Bayesian trust evaluation model.

In a trust scheduling based system, the process of task submission and execution is as follows: (1) tasks are submitted to a task queue; (2) a task scheduler fetches tasks from the queue and communicates with a schedule advisor; (3)

schedule advisor communicates with a trust model; (4) the trust model analyzes local transactions, communicates with a trust middleware, gets detailed trust resource information of task and transfers it to a task scheduler; (5) who then executes the task on a most trustworthy Cloud resource node.

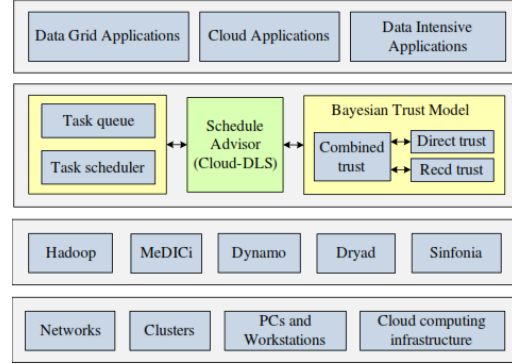


Fig. 1 Trusted dynamic scheduling framework

IV. EXPERIMENTAL RESULTS

Simulations are carried out using 10 VM and variable number of jobs with computing power of 500 to 1500 MIPS. The makespan results of proposed Trust HEFT are compared with the HEFT.

TABLE I
MAKESPAN ACHIEVED

Number of jobs	HEFT	Proposed Trust HEFT
50	494	482
100	892	870
150	1398	1373
200	1817	1780
250	2324	2265
300	2862	2793
50	494	482

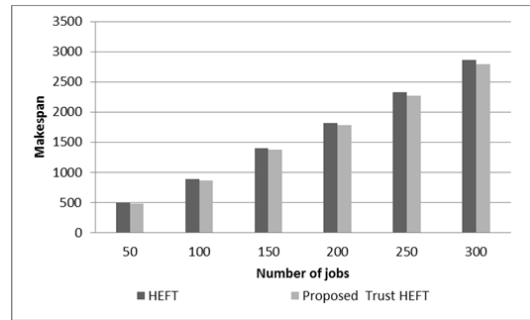


Fig. 2 Makespan Achieved

The proposed HEFT takes less time than compared with the normal HEFT algorithm. When the number of jobs is 50, the proposed HEFT decreased by 2.459% when compared to HEFT. When the number of jobs is 300, the proposed HEFT decreased by 2.44% when compared to HEFT.

V.CONCLUSION

Cloud computing ensures ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources. Trust establishes and maintains relationship between two entities for a long time. Applying trust models to scheduling lowers failure ratio and reassigning in cloud environments. This study focused on Trust based scheduling to improve cloud security by modifying the HEFT algorithm and by proposing a Trust reputation HEFT. Results show that the new Trust HEFT algorithm is time efficient compared to the normal HEFT algorithm.

REFERENCES

- [1] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.
- [2] Salot, P. (2013). A Survey of Various Scheduling Algorithm In Cloud Computing Environment. *International Journal of Research in Engineering & Technology (IJRET)*, 2(2), 131-135.
- [3] Gupta, H., Singh, D., & Gupta, B. K. Scheduling Techniques in Cloud Computing: A Systematic Review.
- [4] Hamlen, K., Kantarcioglu, M., Khan, L., & Thuraishingham, B. (2010). Security issues for cloud computing. *International Journal of Information Security and Privacy (IJISP)*, 4(2), 36-48.
- [5] Jain, P. (2012). Security Issues and their Solution in Cloud Computing. *International Journal of Computing & Business Research*.
- [6] Rousseau, D. M., Sitkin, S. B., Burt, R. S., & Camerer, C. (1998). Not so different after all: A cross-discipline view of trust. *Academy of management review*, 23(3), 393-404.
- [7] Kumar, V. S., & Aramudhan, M. (2014). Trust Based Resource Selection and List Scheduling in Cloud Computing. *International Journal of Advances in Engineering & Technology*, 6(6).
- [8] Alhmouz, R., Challa, S., & Momani, M. (2010). Bayesian fusion algorithm for inferring trust in wireless sensor networks.
- [9] Gambetta, D. (2000). Can we trust trust. *Trust: Making and breaking cooperative relations*, 213-237.
- [10] Hwang, K., & Li, D. (2010). Trusted cloud computing with secure resources and data coloring. *Internet Computing, IEEE*, 14(5), 14-22.
- [11] Anitha, T. N. A Novel Approach to Balance The Dynamic Load Using Task Allocation On Distributed Content Based Cluster Servers.
- [12] Kumar, S. M., Mathur, T. P., & Antoine, M. Dynamic Load Scheduling Optimization of Power Plants.
- [13] Zhong-wen, G., & Kai, Z. (2012, December). The research on cloud computing resource scheduling method based on time-cost-trust model. In *Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on* (pp. 939-942). IEEE.
- [14] Daniel, D., & Lovesum, S. J. (2011, July). A novel approach for scheduling service request in cloud with trust monitor. In *Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference on* (pp. 509-513). IEEE.
- [15] Goyal, M. K., Aggarwal, A., Gupta, P., & Kumar, P. (2012, December). QoS based trust management model for Cloud IaaS. In *2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing* (pp. 843-847).
- [16] Lu, K., Jiang, H., Li, M., Zhao, S., & Ma, J. (2012, June). Resources collaborative scheduling model based on trust mechanism in cloud. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on* (pp. 863-868). IEEE.
- [17] Yang, Y., & Peng, X. (2013, October). Trust-Based Scheduling Strategy for Workflow Applications in Cloud Environment. In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference on* (pp. 316-320). IEEE.
- [18] Li, W., Zhang, Q., Wu, J., Li, J., & Zhao, H. (2012, September). Trust-based and QoS Demand Clustering Analysis Customizable Cloud Workflow Scheduling Strategies. In *Cluster Computing Workshops (CLUSTER WORKSHOPS), 2012 IEEE International Conference on* (pp. 111-119). IEEE.
- [19] Wang, W., Zeng, G., Tang, D., & Yao, J. (2012). Cloud-DLS: Dynamic trusted scheduling for Cloud computing. *Expert Systems with Applications*, 39(3), 2321-2329.
- [20] Fan, W., & Perros, H. (2014). A novel trust management framework for multi-cloud environments based on trust service providers. *Knowledge-Based Systems*, 70, 392-406.
- [21] Hussain, W., Hussain, F. K., & Hussain, O. K. (2014, January). Maintaining Trust in Cloud Computing through SLA Monitoring. In *Neural Information Processing* (pp. 690-697). Springer International Publishing.
- [22] Anandharajan, T. V., & Bhagyaveni, M. A. (2014). Trust Based VM Consolidation in Cloud Data Centers. In *Recent Trends in Computer Networks and Distributed Systems Security* (pp. 103-114). Springer Berlin Heidelberg.
- [23] Wang, X., Su, J., Hu, X., Wu, C., & Zhou, H. (2014). Trust Model for Cloud Systems with Self Variance Evaluation. In *Security, Privacy and Trust in Cloud Systems* (pp. 283-309). Springer Berlin Heidelberg.
- [24] Zhu, C., Nicanfar, H., Leung, V., Li, W., & Yang, L. T. (2014, June). A trust and reputation management system for cloud and sensor networks integration. In *Communications (ICC), 2014 IEEE International Conference on* (pp. 557-562). IEEE.
- [25] Wang, W., Zeng, G., Zhang, J., & Tang, D. (2012). Dynamic trust evaluation and scheduling framework for cloud computing. *Security and Communication Networks*, 5(3), 311-318.
- [26] Abbadi, I. M., & Alawneh, M. (2012). A framework for establishing trust in the Cloud. *Computers & Electrical Engineering*, 38(5), 1073-1087.
- [27] Gupta, P., Goyal, M. K., Kumar, P., & Aggarwal, A. (2013, January). Trust and reliability based scheduling algorithm for cloud IaaS. In *Proceedings of the Third International Conference on Trends in Information, Telecommunication and Computing* (pp. 603-607). Springer New York.
- [28] Xu, M., Cui, L., Wang, H., & Bi, Y. (2009, August). A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing. In *Parallel and Distributed Processing with Applications, 2009 IEEE International Symposium on* (pp. 629-634). IEEE.
- [29] Chen, W., & Deelman, E. (2012, October). Workflowsim: A toolkit for simulating scientific workflows in distributed environments. In *E-Science (e-Science), 2012 IEEE 8th International Conference on* (pp. 1-8). IEEE.
- [30] Bala, R., & Singh, G. (2014). An Improved Heft Algorithm Using Multi-Criterion Resource Factors.
- [31] Dogan, A., & Ozguner, F. (2000). Reliable matching and scheduling of precedence-constrained tasks in heterogeneous distributed computing. In *Parallel Processing, 2000. Proceedings. 2000 International Conference on* (pp. 307-314). IEEE.
- [32] Dogan, A., & Ozguner, F. (2002). Matching and scheduling algorithms for minimizing execution time and failure probability of applications in heterogeneous computing. *Parallel and Distributed Systems, IEEE Transactions on*, 13(3), 308-323.

Sumathi D. is with Jayam College of Engineering and Technology, Dharmapuri, Tamilnadu. She is currently pursuing her doctorate in India.

Poongodi P. is currently working in V.S.B. Engineering College, Karur, Tamilnadu, India.