Middle-East Journal of Scientific Research 24 (Techniques and Algorithms in Emerging Technologies): 582-587, 2016
ISSN 1990-9233;
© IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.24.TAET23673

Cloud Security: Trust with Swarm Based Scheduling

[1]D. Sumathi and [2]P. Poongodi

[1]Department of Computer Science And Engineering,
Jayam College of Engineering and Technology, Dharmapuri, India
[2]Department of Electronic and Communication, Karpagam College of Engineering, Coimbatore, India

Abstract: Because huge-scale cloud computing is the next generation of computation and information platforms, new protocols are required for scheduling jobs on the trusty nodes for executing, assuring the high speed of communication, reducing job execution times, lowering ratio of failure execution and improving security of execution environment of important data. Security is the largest problem in cloud computing because when using storage services in remote locations, consumers are not aware of what happens to their information. The trust mechanism has proven to be an appropriate substitute to the aforesaid security issues as it establishes entities' relationship quickly and safely. Scheduling protocols minimize resource starvation and guarantee fairness amongst all parties utilizing resources. In this study, trust with swarm based scheduling is performed for obtaining the cloud security.

Key words: Cloud Computing · Security · Trust and Scheduling algorithms

INTRODUCTION

Trust is a complex concept which has no universally accepted definition. Broadly, it refers to the establishment as well as maintenance of relationships between two entities for a long duration. Employing trust models to scheduling decreases failure ratio as well as reassigning in a cloud environment [1].

Scheduling is utilized for distributing resources amongst parties that simultaneously as well as asynchronously demand it. Scheduling protocols minimize resource starvation apart from ensuring fairness amongst parties utilizing resources [2, 3]. Scheduling by way of Task Scheduler assists in the automation of routine tasks' performance by observing what criteria are chosen for initiating tasks and then executing tasks when certain criteria are fulfilled.

Scheduling allocates tasks to available resources based on a tasks qualities and need. Scheduling aims to increase the use of resources without affecting cloud provided services. Scheduling includes resource scheduling and job scheduling. Following are some cloud computing scheduling needs.

- Fair resource allocation – Scheduling is carried out such that resources allocation is fair.
- QoS – Resources and jobs are scheduled to achieve the quality of services.
- Resource utilization – the degree to which system resources are used. A good scheduling algorithm ensures maximum resource use.
- Energy consumption – degree to which system resources are consumed. A good scheduling algorithm saves energy.

Job scheduling is a major activity in all computing environments. Cloud computing is a latest technology developing drastically. To efficiently increase working of cloud computing environments, job scheduling is performed to gain maximum profit. The scheduling algorithms goal in distributed systems is spreading the load on processors and maximizing their utilization while reducing total task execution time. Job scheduling, a famous optimization problem, has a major role in improving flexible and reliable systems. To use the Cloud's tremendous capabilities, efficient scheduling algorithms are needed.

Scheduling algorithms are applied by cloud resource manager to optimally dispatch tasks to cloud resources. There are many scheduling algorithms to reduce tasks total completion time in distributed systems. The algorithms try to minimize tasks overall completion time by

Corresponding Author: P. Poongodi, Department of Electronic and Communication, Karpagam College of Engineering, Coimbatore, India.

finding most suitable resources to be allocated to tasks. Reducing the overall completion time of tasks does not lead to the reduction in an execution of individual tasks.

The purpose is scheduling jobs to adaptable resources in accordance with adaptable time, which necessitates locating a sequence where jobs are executed under transaction logic constraints. Scheduling algorithm includes static scheduling and dynamic scheduling algorithms and both have their advantages and limitations. Dynamic scheduling algorithms ensure higher performance compared to static algorithms but also lead to a lot of overhead [4].

The task workflow scheduling in a distributed computing platform is an NP-hard problem. It is even more complex and challenging when virtualized clusters execute many tasks in Cloud platforms. Cloud resources being heterogeneous and dynamic by nature makes scheduling in cloud NP-hard [5].

Approximate methods guarantee finding optimal solutions for good solutions in reduced time for NP-hard combinatorial optimization problems. Basic approximate methods distinguish between constructive and local search methods. As scheduling is NP-hard, it is handled by various heuristic methods which provide solutions for problem instances in a restricted manner. The resource allocation problem is NP-hard. Effectively scheduling dependent and independent tasks on distributed sources that could be virtualized clusters of servers in a cloud platform makes the issue more complex and challenging with guaranteed solution quality.

If cloud environment is un-trusted, then scheduling is uncertain. Developing a model to measure trust reduces uncertainty among open distributed system's computing nodes like grid/cloud environments. Execution time and reliability determine trust degree. Scheduling logs store trust degree any time, sorting it decreasingly and computer slots are called according to those whose trust degree is greater. This algorithm is stable and reliable. Using trust in scheduling improves reliability and robustness. Reputation methods provide computing systems earlier behavior details which decide computing system's trust.

Literature Survey: Ko *et al*. [6] proposed a novel method of approaching conventional security and trust issues. A data-centric, detective approach is proposed to increase trust and security of data in the cloud. The proposed model, called TrustCloud, comprises a set of methods which address cloud security, trust as well as accountability from a detective method at every level of granularity.

Li *et al*. [7] focused on the trust computing requirement of multi-cloud collaborative services and developed a Data-driven and Feedback-Enhanced Trust (DFET) calculating pattern across various data centres with various innovative methods. An enhanced and hierarchical feedback mechanism is proposed that can effectively reduce networking risk while improving system dependability. Theoretical analysis showed that DFET pattern is highly dependable against garnished and bad-mouthing attacks.

Yang & Peng [8] formulated the scheduling problem for workflow applications with trust constraints and presented a novel scheduling algorithms based on trust. Experimental results illustrated that the suggested heuristic scheduling protocol is better than the conventional protocol for scheduling application workflow and can detect the most trustworthy execution flow effectively.

Tan *et al*. [9] proposed a trust service-oriented workflow scheduling algorithm. The scheduling algorithm adopted a trust metric that combines direct trust and recommendation trust. In addition, balance policies provided to enable users to balance different requirements, including time, cost and trust. A case work was conducted to illustrate the value of the proposed algorithm. The experimental results showed that the proposed approach is effective and feasible.

Li *et al*. [10] included trust in workflow's quality of service target and suggested a new customizable cloud workflow scheduling model. The novel framework split workflow scheduling into two phases: macro as well as micro multi-workflow scheduling. The simulation experiments showed that the novel schema had several benefits in reducing workflow's final completion times, attains relatively high execution success rates as well as consumer satisfaction when contrasted with other similar models.

MATERIALS AND METHODS

The mapping of application workflow tasks to distributed resources has many objectives. Reducing total computation cost of an application workflow was focused on. An application workflow denoted as a Directed Acyclic Graph (DAG) represented by G= (V, E), where V = {T1, ..., Tn} is set of tasks and E represents data dependencies between the tasks, that is, fj, k = (Tj, Tk) ◎ E is data produced by Tj and consumed by Tk. A set of storage sites S = {1, ..., i}, a set of compute sites PC = {1, ..., j}, as well as a set of tasks T = {1, ..., k}. The 'average' computation time of a task Tk on a compute resource PCj is assumed for a certain size of an input. Then, the cost of computing a task on a compute host is inversely proportional to the time taken for computation on that resource. It is assumed that the cost of unit data access di, j from a resource i to a resource j is known. The access cost is fixed by the service provider (Amazon Cloud Front). Transfer cost is calculated based on bandwidth between sites.

But, the cost to transfer unit data between sites, per second is non-negative, symmetric and fulfils the triangle inequality. Let Cexe (M)j be total cost of all tasks assigned to a compute resource PCj. This value is computed by adding all node weights of all tasks designated to every resource in mapping M. Let Ctx (M)j be total access cost (including transfer cost) between tasks assigned to a compute resource PCj and those not assigned to that resource in mapping M. This value is a product of output file size (given by edge weight ek1, k2) from a task k1 ◎ k to task k2 ◎ k and cost of communication from resource wherein k1 is mapped (M (k1)) to another resource where k2 is mapped (M (k2)). The average costs of transmission of unit data between two resources is expressed as dM (k1), M (k2). The cost of communication is applicable only when two tasks have file dependency between them, i.e. when ek1, k2> 0. For two or more tasks, executing on same resource, communication cost is zero [11].

$$C_{exe}(M)_j = \sum_k w_{kj} \qquad \forall M(k) = j$$
$$C_{tx}(M)_j = \sum_{k1\in T}\sum_{k2\in T} d_{M(k1),M(k2)} e_{k1,k2}$$
$$\forall M(k1) = j \text{ and } M(k2) \neq j$$
$$C_{total}(M)_j = C_{exe}(M)_j + C_{tx}(M)_j$$
$$Cost(M) = \max(C_{total}(M)_j) \quad \forall j \in P$$
$$Minimize(Cost(M)) \qquad \forall M$$

All tasks are not mapped to one compute resource. Initial cost maximization distributes tasks to all resources. Subsequent minimization of overall cost ensures that total cost is minimal after initial distribution. For an assignment M, total cost C_{total} (M)$_j$ for a compute resource PC$_j$ is the sum of execution cost and access cost. When estimating the total cost for all resources, the largest cost for all resources is minimized which indirectly ensures that tasks are not mapped to one resource and that there will be a cost distribution among resources.

Dynamic Level Scheduling (DLS) protocol is a compile time, static list scheduling heuristic for allocating a DAG-structure application to a set of heterogeneous machines to reduce application execution time. A task machine's dynamic level, (v_i, m_j) is defined as in equation (2):

$$DL(v_i, m_j) = SL(v_i) - \max\{t_{i,j}^A, t_j^M) + \Delta(v_i, m_j) \qquad (2)$$

where *SL(v$_i$)* is called static level of a task, $\{t_{i,j}^A, t_j^M)$ 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_j^M denotes a time when machine m_j is available for task execution $v_i.\Delta(v_i, m_j) = t_i^E, t_{i,j}^E$ reflects computing performance of a machine, t_i^E denotes execution time of task v_i on all free machines and $t_{i,j}^E$ represents execution time of task v_i on machine m_j.

For offsetting the neglecting of resource node trustworthiness in cloud systems, trust-dynamic level scheduling algorithm in Cloud environment (Cloud-DLS) is developed and trust dynamic level is defined as in equation (3):

$$TDL_S(v_i, n_j) = T_S(v_i, n_j)^{a_i} * (SL(v_i)\max\{t_{i,j}^A, t_j^M\} + \Delta(v_i, n_j)) \qquad (3)$$

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 θ discussed above. α_i is a QoS factor of v_i, satisfying $0 \leq \alpha_i \leq 1$ and $\Sigma\, a_i = 1$. To one task machine pair (v_i, n_j), when α_i is increased, it implies needs of task v_i in trust is increased, so scheduling priority is decreased accordingly. Hence, the protocol is scalable and meets various QoS requisites. By adjusting, α_i users' different trust requirements are satisfied.

Trusted dynamic scheduling protocol is executed as middleware for plugging into cloud systems, by which tasks are implemented on trust nodes in an efficient manner. The basic integrated model has its basis in Cloud-DLS as given in Figure 1. This model has 4 tiers, which are resource and infrastructure tier, basic middleware, trustworthy scheduler as well as the client.

In trust scheduling based systems, the procedure of task submission and execution involves: Tasks are submitted to a task queue; a task scheduler fetches tasks from the queue and communicates with a schedule advisor; schedule advisor communicates with a trust model; the trust model analyses local transactions, communicates with trust middleware, obtains detailed trust resource data of tasks and transfers it to task schedulers; Who then executes the task on a most trustworthy Cloud resource node.

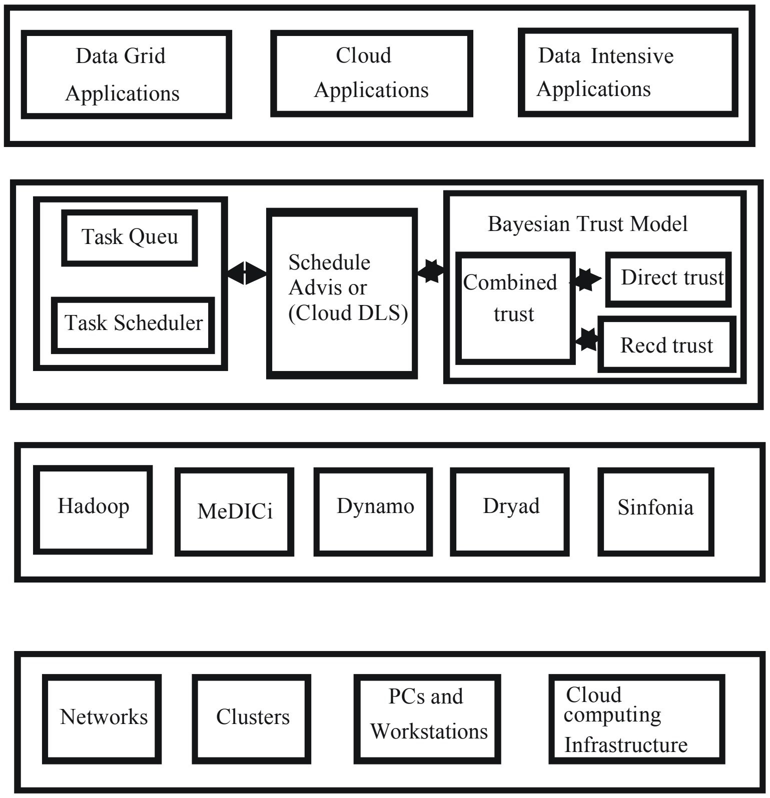

Fig. 1: Trusted dynamic scheduling framework.

The workloads utilized in the benchmark range across several data intensive computing features in Cloud computing. It have implemented six applications for web and image data analysis as benchmarks for the data intensive computing, which comes from previous work [12].

RESULTS AND DISCUSSION

Simulations are carried out using 10 VM and variable number of jobs with computing power of 100-900 jobs. The Trust-max-min, Trust-HEFT, Trust - IWO, Trust - BPSO and Dynamic Level Scheduling (DLS) are evaluated. The average schedule length with trust, ratio of successful execution and number of recommendations are shown in Table 1 to 3. Figure 2 to 4 shows the same.

Table 1: Average Schedule Length with Trust

Number of tasks	TR-Max-Min	TR-HEFT	TR-CS	TR-BPSO	DLS
100	341	322	300	302	330
300	1074	993	932	946	1022
500	1805	1677	1622	1564	1719
700	2484	2330	2153	2187	2394
900	3201	2983	2809	2702	3054

Table 2: Ratio of Successful Execution

Number of tasks	TR-Max-Min	TR-HEFT	TR-CS	TR-BPSO	DLS
100	0.83	0.87	0.92	0.89	0.86
300	0.84	0.86	0.89	0.88	0.84
500	0.78	0.83	0.87	0.85	0.82
700	0.78	0.81	0.86	0.82	0.8
900	0.74	0.8	0.83	0.81	0.79

Table 3: Dynamic Trust Value

Number of recommendations	Trust Value, a=0.5	Trust Value a=0.75	Trust Value a=1
2	0.44	0.45	0.48
4	0.52	0.55	0.59
6	0.68	0.74	0.78
8	0.72	0.79	0.84
10	0.74	0.83	0.89
12	0.78	0.87	0.92
14	0.81	0.91	0.94
16	0.83	0.93	0.96

From the Figure 2, it can be observed that the TR-CS has lower average schedule length with trust by 12.79%, 14.15%, 10.67%, 14.27% & 13.04% for TR-max min, by 7.07%, 6.33%, 3.33%, 7.89% & 6% for TR-HEFT, by 0.66%, 1.49%, 3.64%, 1.56% & 3.88% for TR-BPSO and by 9.52%, 9.21%, 5.8%, 10.6% & 8.35% for DLS when compared with 100, 300, 500, 700 and 900 number of tasks respectively.

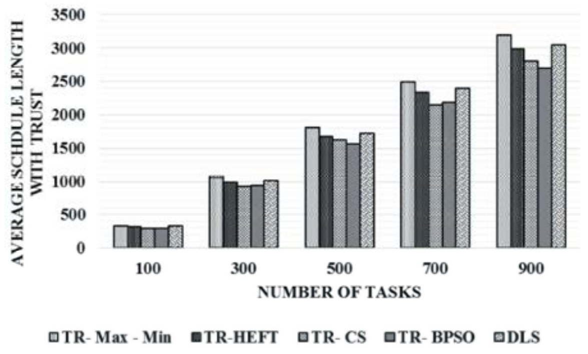

Fig. 2: Average Schedule Length with Trust

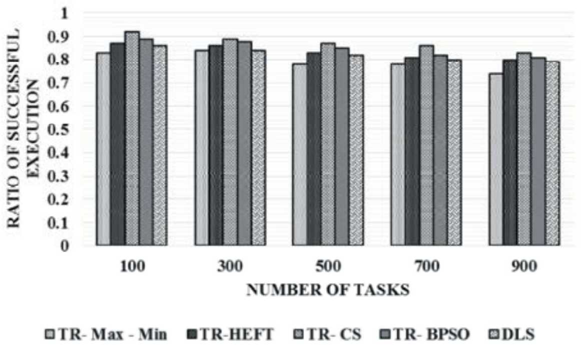

Fig. 3: Ratio of Successful Execution

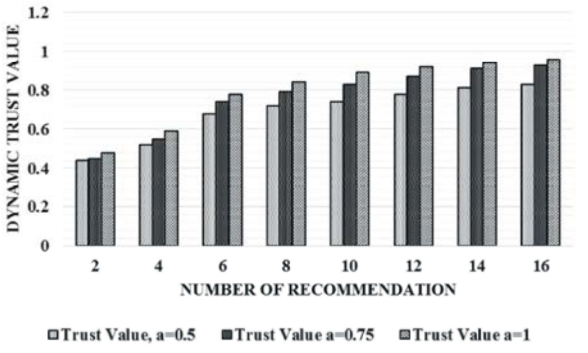

Fig. 4: Dynamic Trust Value

From the Figure 3, it can be observed that the TR-CS has higher ratio of successful execution by 10.28%, 5.78%, 10.9%, 9.75% & 11.46% for TR-max min, by 5.58%, 3.42%, 4.7%, 5.98% & 3.68% for TR-HEFT, by 3.31%, 1.12%, 2.32%, 4.76% & 2.43% for TR-BPSO and by 6.74%, 5.78%, 5.91%, 7.22% & 4.93% for DLS when compared with 100, 300, 500, 700 and 900 number of tasks respectively.

From the Figure 4, it can be observed that the trust value a=1 has higher dynamic trust value by 12.61%, 15.38%, 16.47% & 14.52% for trust value, a=0.5 and by 7.01%, 6.13%, 5.58% & 3.17% for trust value a=0.75 when compared with 4, 8, 12 & 16 number of recommendations.

CONCLUSION

Trust refers to the establishment as well as maintenance of relationships between two entities for a long duration. Employing trust models to scheduling decreases failure ratio as well as reassigning in a cloud environment. This work focused on Trust based scheduling to improve cloud security by max-min, HEFT, IWO, BPSO algorithm and by proposing a DLS. Results show that the TR-CS has higher ratio of successful execution by 10.28%, 5.78%, 10.9%, 9.75% & 11.46% for TR-max min, by 5.58%, 3.42%, 4.7%, 5.98% & 3.68% for TR-HEFT, by 3.31%, 1.12%, 2.32%, 4.76% & 2.43% for TR-BPSO and by 6.74%, 5.78%, 5.91%, 7.22% & 4.93% for DLS when compared with 100, 300, 500, 700 and 900 number of tasks respectively.

REFERENCES

1. Alhmouz, R., S. Challa and M. Momani, 2010. Bayesian fusion algorithm for inferring trust in wireless sensor networks.
2. Kaleeswaran, A., V. Ramasamy and P. Vivekanandan, 2012. Dynamic scheduling of data using genetic algorithm in cloud computing. International Journal of Advances in Engineering & Technology, 5(2).
3. Kumar, V.S., 2014. Hybrid Optimized List Scheduling And Trust Based Resource Selection In Cloud Computing. Journal of Theoretical & Applied Information Technology, 69(3).
4. Salot, P., 2013. A survey of various scheduling algorithm in cloud computing environment. International Journal of research and engineering Technology (IJRET), ISSN, 2319-1163.
5. Jacob, L., V. Jeyakrishanan and P. Sengottuvelan, 2014. Resource Scheduling in Cloud using Bacterial Foraging Optimization Algorithm. International Journal of Computer Applications, 92(1): 14-20.
6. Ko, R.K., M. Kirchberg and B.S. Lee, 2011. From system-centric to data-centric logging-accountability, trust & security in cloud computing. In Defense Science Research Conference and Expo (DSR), 2011 (pp: 1-4). IEEE.
7. Li, X., H. Ma, W. Yao and X. Gui, 2015. Data-driven and Feedback-Enhanced Trust Computing Pattern for Large-scale Multi-Cloud Collaborative Services.
8. Yang, Y. and X. Peng, 2013. 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.

9. Tan, W., Y. Sun, L.X. Li, G. Lu and T. Wang, 2014. A trust service-oriented scheduling model for workflow applications in cloud computing. Systems Journal, IEEE, 8(3): 868-878.
10. Li, Q., 2012. Applying stochastic integer programming to optimization of resource scheduling in cloud computing. Journal of Networks, 7(7): 1078-1084.
11. Pandey, S., L. Wu, S.M. Guru and R. Buyya, 2010. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In Advanced information networking and applications (AINA), 2010 24th IEEE international conference on (pp: 400-407). IEEE.
12. Calheiros, R.N., R. Ranjan, C.A.F. De Rose and R. Buyya, 2009. CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services, Technical Report, GRIDS-TR-2009-1, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia.