Journal of Theoretical and Applied Information Technology
30th November 2014. Vol. 69 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HYBRID OPTIMIZED LIST SCHEDULING AND TRUST BASED RESOURCE SELECTION IN CLOUD COMPUTING

[1]V.SURESH KUMAR, [2] Dr. ARAMUDHAN

[1]Sree Narayana Guru Institute of Science and Technology, Master of Computer Application, Ernakulam, Kerala, India

[2]Perunthalaivar Kamarajar Engineering College, Department of Information Technology, Tamilnadu, India

E-mail: [1]sureshkumarv.clo@gmail.com

ABSTRACT

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. Task scheduling is fundamental to achieving cloud computing efficiency. But, it is a challenge to efficiently schedule algorithm design and implement it, as general scheduling is a NP-complete problem. Most task-scheduling cloud computing methods consider task resource requirements for CPU and memory, and not bandwidth requirements. This paper proposes to optimize scheduling using BAT-Gravitational hybrid algorithm with resources being chosen on trust.

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

1. INTRODUCTION

Cloud Computing encapsulates computing resources delivery as a service. It is an iteration of utility computing and returns to the 'renting' resources model. Terms 'cloud computing' and 'cloud' are accepted as part of industry lexicon and despite frequent misuse in advertising there is much research that underpins the area. Leveraging cloud computing is the de facto means to deploy internet scale systems. Much of the internet is tied to a limited number of cloud providers. Cloud computing advancement is intrinsic to development of next generation internet. Cloud computing started as a mixture of virtualization, distributed storage and service related architecture [1].

Cloud computing technology makes resource a single access point to clients and is implemented on a pay per usage model. Though cloud computing has advantages like prescribed and abstracted infrastructure, pay per consumption, totally virtualized environment, dynamic infrastructure, free of software and hardware installations, a major concern is how requests are satisfied. This evolves around resource scheduling. Resource allocation must be efficient and maximize system utilization and performance. Cloud computing is sold on demand based on time constraints specified in minutes/hours [2]. Scheduling should be such that resource must be used.

Cloud computing promotes on-demand model for IT resource provisioning where resources are virtual servers, services, or application platforms. Three major service offerings to defining Cloud computing: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). Infrastructure-as-a-Service ensures delivery of on-demand components to build IT infrastructure like storage, bandwidth and virtual servers, which are customized with required software stack to host applications.

Platform-as-a-Service providers deliver development/runtime environments for cloud hosted applications. They permit physical aspects abstraction of distributed systems by providing scalable middleware for managing application execution and dynamic resource provisioning. Software-as-a-Service providers ensure applications and services on-demand, accessible through the Web. SaaS applications are multi-tenant, composed by integrating available components over the net [3]. The offer of different models on which computing resources are rented creates new perspectives as to how IT infrastructures are used, as Cloud offers means to increase IT resource availability when necessary, and when resources

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

are needed, reducing costs linked to resource acquisition and maintenance.

Originating from IBM CP/CMS operating system, Virtual Machines (VMs) is a cornerstone for cloud computing. A VM is a software computer system implementation, running in isolation with other processes, but behave as physical systems. A single multi-processor server can run VMs, one per core (though cloud providers usually oversell CPUs) allowing one server to be used to capacity, reducing unused CPU cycles and lowering wasted energy. Virtualizing a computer system reduces management overhead and allows it to move between physical hosts and to instantiated or terminated. These properties create rapid elasticity and scalability underpinning cloud computing [1].

A solution between centralized and decentralized deployments is virtualization. Instead of purchasing and maintaining a computer for an application, each application is given its operating system, with all operating systems residing on single hardware. This ensures decentralization benefits like security and stability, while making most use of a machine's resources. Virtualization permits an operator to control guests operating system's use of CPU, memory, storage, and resources, so each guest receives only resources needed. This eliminates danger of one runaway process consuming available memory or CPU. It helps IT staff satisfy service level requirements for specific applications.

Virtualization comes in various implementations. Known as "full virtualization" in its basic form, hypervisor provides an emulated machine where an operating system runs. VMWare® is an example. The advantage of this approach is flexibility: one runs a RISC based S as guest on Intel-based host. While this is obvious, there are performance problems in emulating complete hardware in software. Even with optimization, it is difficult to get performance from fully virtualized environments.

Optimal resource allocation or task scheduling in cloud decides optimal systems required in a cloud so that cost is minimized and SLA upheld. Cloud service scheduling is categorized at user and system levels. User level scheduling handles problems raised by service provision between providers and customers. System level scheduling handles resource management within data centers. Market-based and auction-based schedulers regulate cloud resources supply and demand. Market based resource allocation in cloud computing environment is effective as resources are virtualized and delivered to users as service.

Static scheduling allows pre-fetching required data and pipelining different task execution stages. Static scheduling uses reduced runtime overhead. In dynamic scheduling, information on job components/task is not known in advance [4]. So, execution time of tasks may be unknown and tasks allocation is done on the fly as the application executes. Service request scheduling strategies in 3-tier cloud structure, which has resource providers, service providers and consumers, must satisfy service providers and consumers objectives.

Trusting in an individual is due to many analyses, which generate definition of trust. Trust (or, symmetrically, distrust) is a specific level of subjective probability, 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 [5]. Reputation is defined in a scenario where there is not enough information to infer that an entity is unreliable [6]. To achieve the inference value, an entity asks other entities opinions. Based on information from questioned entities, requesting entity calculates reputation from own information based on its trust values and obtained information from third parties (degree of trust in them). Reputation calculation is got by analyzing an entity's behaviour over time [7].

Scheduling is used to distribute resources among parties which simultaneously and asynchronously ask for it. Scheduling algorithms are used for scheduling. Scheduling algorithms minimize resource starvation and ensure fairness among parties using resources. Scheduling deals with which resources need to be allocated for received task [8]. Many scheduling algorithms are available. Scheduling converts an outline plan for projects into time-based graphic presentations using time constraints.

Scheduling through Task Scheduler helps automatic routine tasks' performance on chosen schedules. Task Scheduler does this by observing whatever criteria is selected to initiate tasks and then to execute the task when criteria are met. With Scheduled Tasks, any script, document or program is scheduled to run at a specified time when the task was created. Task scheduling problem optimizes the application's overall performance and provides assurance for result correctness.

Conventional scheduling methods are infeasible in a cloud environment owing to its dynamical, distributed, and sharable properties. Tasks resource allocation is for services to meet performance targets. Many jobs demand different resources while operating simultaneously [9]. It is important

Journal of Theoretical and Applied Information Technology
30th November 2014. Vol. 69 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

for a cloud's efficient working to balance jobs on appropriate resources for optimal performance. So, various task parameters are considered for proper scheduling. Available resources should be used effectively without affecting service parameters. Cloud environment system scheduling is an NP-complete problem. As number of users increase, tasks to be scheduled increase proportionately. Hence, better algorithms are required to schedule tasks on such systems. Scheduling algorithms required are service-oriented and vary in varied environments.

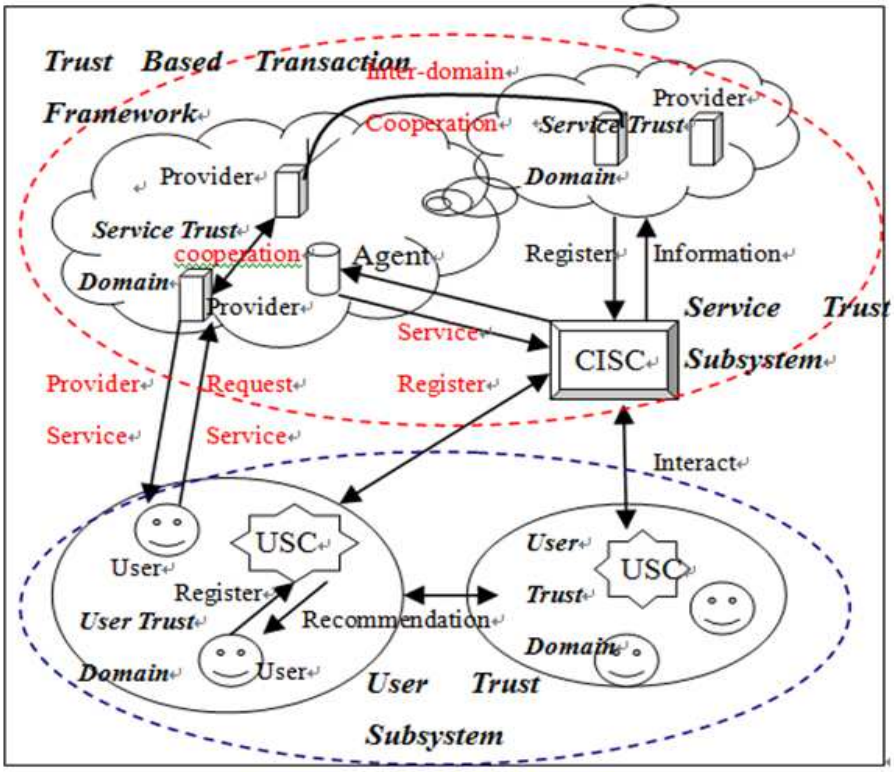

Figure 1 Trust Based Cloud Transaction Framework

Cloudsim software is used in this study to simulate performance of the proposed BAT-Gravitational hybrid algorithm and trust based resource selection. The reminder of the study is organized as follows: Section 2 discusses related works in literature. Section 3 describes methods used in proposed work; Section 4 talks about experiments and obtained results and Section 5 gives the conclusion.

2. RELATED WORK

An algorithm based on costs with user task grouping was proposed by Selvarani and Sadhasivam [10]. The new cloud scheduling approach uses an improved cost-based scheduling algorithm to make efficient tasks mapping to available cloud resources. The scheduling algorithm measures resource cost and computation performance, improves computation/ communication ratio by grouping user tasks according to specific cloud resources processing ability and sends grouped jobs to that resource.

Service request scheduling's problem in cloud computing systems was addressed by Lee et al [11]. A three-tier cloud structure includes resource providers, service providers and consumers. Service request scheduling strategies in such scenarios must satisfy objectives of service providers and consumers. A dynamic priority scheduling algorithm to solve this problem was suggested. The algorithm was better and optimal than FCFS and SPSA.

Two resource allocation and scheduling problems in cloud computing were identified by Elghoneimy et al [12]. Hadoop MapReduce and its schedulers described and presented recent research efforts including alternative schedulers and enhancements to current schedulers. The second scheduling problem is virtual machines

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

provisioning to cloud resources. A survey of different approaches to solve resource allocation problem was presented. It included research and standards for inter-connecting clouds and discussing the suitability of running scientific applications on a cloud.

An Ant Colony Optimization (ACO) based job scheduling algorithm, which adapts cloud computing's dynamic characteristics and integrates specific advantages of ACO in NP-hard problems was proposed by Song et al [13]. It minimized pheromone based job completion time. Results showed that it was a promising ACO algorithm for scheduling jobs in cloud computing environments.

A service flow scheduling with different Quality Of Service (QoS) requests in cloud computing was proposed by Liu et al [14] that adopted use of an ACO algorithm to optimize service flow scheduling. There is much research on scheduling in cloud computing, but most are about workflow and job scheduling. There is reduced research on service flow scheduling. In the proposed model, a default rate describes the ratio of cloud service providers breaking service level agreements (SLA). It introduces an SLA monitoring module to monitor cloud services running state.

Three types of meta-heuristic algorithms called firefly algorithm, bat algorithm and cuckoo search algorithm were used by Arora and Singh [15] to find optimal solutions. Firefly is inspired by flies' behaviour, bat algorithms are based on bats' echolocation behaviour while in cuckoo search, a pattern corresponds to a nest with each individual pattern attribute corresponding to a cuckoo-egg. Computational experiments were conducted using each algorithm. On results being analyzed it was observed that firefly algorithm performed better than bat algorithm and cuckoo searches. An improved Bat algorithm version with chaos is represented by Afrabandpey et al [16]. Difficulties of tackling real-world problems due to growing complexities motivated computer scientists to search for efficient problem solving approaches. Metaheuristic algorithms are examples of such approaches. Bat Algorithm (BA) is a new meta-heuristic optimization algorithm, in various optimization tasks recently. The new approach was based on substitution of Random Number Generator (RNG) with chaotic sequences for parameter initialization. Simulation on some mathematical benchmark functions demonstrated the new approach's validity in which Chaotic Bat Algorithm (CBA) outperformed classical BA.

Recently developed bat algorithm was used by Yang et al [17] to solve topology optimization problems. Results revealed that distribution of different topological characteristics like materials was achieved efficiently. The bat algorithm was tested by solving nonlinear design benchmarks. Results suggested that bat algorithm was efficient in solving nonlinear global optimization problems and topology optimization.

Hybridization of an improved Gravitational Attraction Search (as a local search algorithm) with Imperialist Competitive Algorithm was introduced by Jula et al [18] which applied a new memetic algorithm to gain optimal/near optimal response time and execution fees simultaneously, for cloud computing service composition. Using roulette wheel selection algorithm to ensure well-advised and non-blind decisions to choose the countries in each empire that should apply a local search has assisted hybrid algorithm to achieve better solutions.

A hybrid GSA with sequential quadratic method to solve dynamic economic dispatch of generating units with valve point effect was presented by Swain et al [19]. A two phase optimization scheme was proposed. In the first phase, GSA explored solution space directly. In second phase, SQP was applied to improve GSA algorithm. Thus, SQP guided GSA algorithm to better performance in complex solution space. To validate the proposed method's effectiveness, a 10-unit test system was taken for results comparison with other heuristic optimization methods. GSA-SQP can find global optimum results with reduced computational time and with increased effectiveness and robustness.

An optimized Genetic Algorithm (GA) based algorithm to schedule independent and divisible tasks adapting different computation and memory requirements was proposed by Zhao et al [20]. The algorithm was prompted in heterogeneous systems, where resources (including CPUs) have computational/communication heterogeneity. Dynamic scheduling is also considered. Though GA was meant to solve combinatorial optimization problems, it is inefficient for global optimization. So, the authors concluded in favour of optimized genetic algorithm.

An enhanced cloud simulating system DartCSim based on CloudSim with a more user-friendly interface was developed by Li et al [21], that allows users to configure all data of simulation environment with visual interface, including network cloudlets configurations, network topology and algorithms to manage cloud center.

The cloud simulation platform, CloudSim was extended by Long et al [22] by adding file striping and data replica management functions, making it a

Journal of Theoretical and Applied Information Technology
30th November 2014. Vol. 69 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

simulation platform to compute and store. The expanded CloudSim can implement different data layout and replicate management strategy. And it can easily add functionality to meet other needs. The design's detailed description has proved its significance through examples.

An enhanced cloud simulation platform called DartCSim+ that supported energy-aware network simulation and network-aware live migration was designed and implemented by Li et al [23]. A resubmit mechanism was implemented for packets transmission to ensure real network behaviour to solve transmission failures caused by migration/network failure. Finally, three groups of experiments demonstrated the effectiveness of DartCSim+.

3. METHODOLOGY

3.1 BAT Algorithm

Xin-She Yang developed the Bat algorithm in 2010 which exploits bats echolocation. Bats use sonar echoes to detect obstacles. It is known that sound pulses are transformed into a frequency which is reflected by obstacles. Bats navigate using time delay from emission to reflection. They emit short, loud sound impulses. Pulse rate is defined as 10 to 20 times per second. After hitting and reflecting, bats transform their pulse into useful information to gauge the distance from prey. The bats use wavelengths ranges from 0.7 to 17 mm or with inbound frequencies of 20-500 kHz. To utilize the algorithm, pulse frequency and rate are to be defined. Pulse rate is determined in the range from 0 to 1, where 0 means no emission and 1 means that bats' emit their maximum [24, 25]. For simplicity, we use approximate or idealized rules [26]:

1. Bats use echolocation to sense distance, and 'know' difference between food/prey and background barriers in some way;
2. Bats fly randomly with velocity v_i at position x_i with a fixed frequency f_{min}, varying wavelength λ and loudness A_0 to search for prey. They automatically adjust wavelength (or frequency) of emitted pulses and adjust rate of pulse emission r ε [0, 1], depending on target proximity;
3. Though loudness varies in many ways, it is assumed that loudness varies from large (positive) A_0 to minimum constant value A_{min}.

Pseudo code of the bat algorithm (BA)

Objective function f(x), $\mathrm{x} = (x_1, ..., x_d)^T$
Initialize the bat population $\mathrm{x}_i (i = 1, 2, ..., n)$ *and* v_i
Define pulse frequency f_i *at* x_i
Initialize pulse rates r_i *and the loudness* A_i
while*(t <Max number of iterations)*
Generate new solutions by adjusting frequency, and updating velocities and locations/solutions
if*(rand >*r_i*)*
Select a solution among the best solutions
Generate a local solution around the selected best solution
end if
Generate a new solution by flying randomly
if$(rand < \mathrm{A}_i \,\&\, \mathrm{f}(\mathrm{x}_i) < \mathrm{f}(\mathrm{x}_*))$
Accept the new solutions
Increase r_i *and reduce* A_i
end if
Rank the bats and find the current best x_*
end while
Postprocess results and visualization

3.2 Gravitational Search Algorithm (GSA)

GSA introduced by Rashedi et al. in 2009 intended to solve optimization problems. This population-based heuristic algorithm is based on law of gravity and mass interactions. The algorithm comprises of a collection of searcher agents interacting with each other through gravity force [27]. Agents are considered objects and their performance measured by masses. Gravity force causes a global movement where objects move to other objects with heavier masses. Slower movement of heavier masses guarantees exploitation of the algorithm corresponding to good solutions. The masses obey the law of gravity as seen in Equation (1) and law of motion in Equation (2).

$$F = G\,(M1M2 \,/\, R2) \quad (1)$$

$$a = F / M \quad (2)$$

Journal of Theoretical and Applied Information Technology
30th November 2014. Vol. 69 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

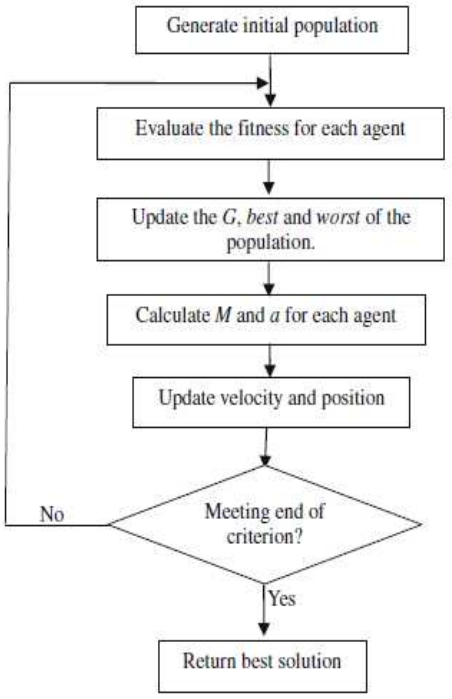

Figure 2 General Principle Of GSA

GSA starts with an agents set, selected randomly or based on criteria, with certain positions and masses representing solutions to a problem, iterated by changing positions based on values like fitness function, velocity and acceleration that are updated with each iteration. To relate those values and parameters, the relations among them should be demonstrated. In a system with N agents, position of ith agent is defined as in equation (3) [28]:

$$X_i = (x_i^1, ..., x_i^d, ..., x_i^n) \text{ for i=1,2,...N} \quad (3)$$

Where x_i^d presents position of ith agent in dth dimension, and ***n*** is dimension of search space. At time t a force acts on mass i from mass j. This force is defined as shown in equation (4):

$$F_{ij}^d = G(t)\frac{M_{pi}(t)\times M_{aj}(t)}{R_{ij}+\varepsilon}(x_j^d(t)-x_i^d(t)) \quad (4)$$

Where M_{aj} is active gravitational mass of agent j, M_{pi} is the passive gravitational mass agent for i, G(t) is the gravitational constant at time t, ε is a small constant, and $R_{ij}(t)$ is Euclidian distance between two agents i and j:

$$R_{ij}(t) = \| X_i(t), X_j(t) \| \quad (5)$$

Total force acting on mass i in the dth dimension in time t is given as in equation (6):

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i}^{N} rand_j F_{ij}^d(t) \quad (6)$$

Where $rand_j$ is a random number in interval [0, 1], K best is set of first K agents with best fitness value.

3.3 Resource Selection based on Trust

Communication Trust (CT) means trust value representing bandwidth availability between client and data centre. Beta reputation system is based on Beta probability density function, *Beta* (α, *β*), and is as in equation (7) [29],

$$f(p \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \quad (7)$$

For each node n_j, a reputation R_{ij} is carried by a neighbouring node n_i. This reputation is embodied in Beta model and carried by 2 parameters α_{ij} and β_{ij}. α_{ij} represents number of successful jobs completed for node n_i had with, or observed about n_j, and β_{ij} number of unsuccessful transactions. Reputation of node n_j, maintained by node n_i, is seen in following equation (8),

$$R_{ij} = Beta(\alpha_{ij}+1, \beta_{ij}+1) \quad (8)$$

Trust is defined as expected value of reputation and given in the equation (9),

$$T_{ij} = E(R_{ij}) = E\{Beta(\alpha_{ij}+1, \beta_{ij}+1)\} = \frac{(\alpha_{ij}+1)}{(\alpha_{ij}+\beta_{ij}+2)} \quad (9)$$

4. EXPERIMENTAL RESULTS

The simulations are conducted using 25 VM with computing power in the range of 1000 MIPS to 3000MIPS and Bandwidth availability of 1000 Mbps to 6000 Mbps. 100 tasks with the size of each task being 30000 MI are scheduled. Resources are selected randomly and based on the proposed trust model. The proposed scheduling mechanism was implemented.

Journal of Theoretical and Applied Information Technology
30th November 2014. Vol. 69 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1 Time Required For Completing The Task

Time required to complete all task	Time (second)
Random resource selection with BAT scheduling	391.14
Trust based resource selection with BAT scheduling	371.26
Random resource selection with GSA scheduling	396.34
Trust based resource selection with GSA scheduling	388.84
Random resource selection with BAT-GSA scheduling	372.77
Trust based resource selection with BAT-GSA scheduling	364.69

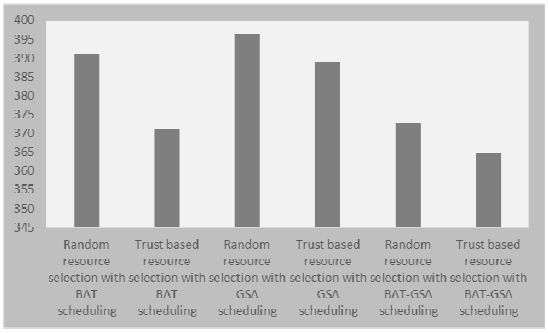

Figure 3 Time Required To Complete All Tasks

Trust based resource selection with BAT-GSA scheduling decreases the time essential to complete all tasks by 8.68% when compared with Random resource selection with GSA scheduling.

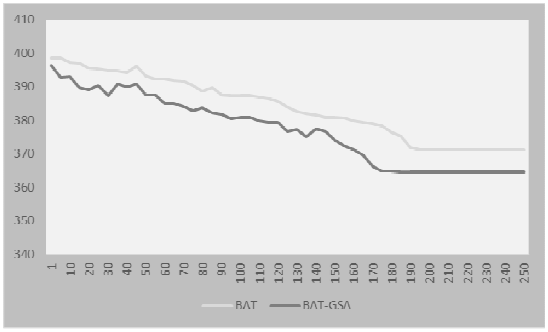

Figure 4 Best Fitness Achieved

5. CONCLUSION

Cloud computing is a high performance computing environment with heterogeneous, large scale, autonomous systems and flexible computational architecture collection. To improve overall cloud computing performance with deadline constraints, a task scheduling model is established to reduce cloud computing's system power consumption and to improve service providers' profit. By applying a trust model to scheduling decreases task failure numbers making task reassignment and restart unnecessary. Cloudsim software simulates performance of the new BAT-Gravitational hybrid algorithm and trust based resource selection. Simulations results prove that trust based resource selection with BAT-GSA scheduling decreases time to complete tasks by 8.68% compared to Random resource selection with GSA scheduling.

REFRENCES

[1] Ward and Barker "A Cloud Computing Survey: Developments and Future Trends in Infrastructure as a Service Computing", 2013.

[2] Abirami, S. P., &Ramanathan, S. (2012). Linear scheduling strategy for resource allocation in cloud environment. International Journal on Cloud Computing: Services and Architecture (IJCCSA), 2(1), 9-17.

[3] Gade, A. H. (2013). A Survey paper on Cloud Computing and its effective utilization with Virtualization. International Journal of Scientific & Engineering Research, 4(12), 357-363.

[4] Chawla, Y., &Bhonsle, M. (2012). A Study on Scheduling Methods in Cloud Computing. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 1(3), 12-17.

[5] Gambetta Diego. (2000). "Can We Trust Trust?", in Gambetta, Diego (ed.) Trust: Making and Breaking Cooperative Relations, electronic edition, Department of Sociology, University of Oxford, chapter 13, 213-237.

[6] Patel, Jigar. "A Trust and Reputation Model for Agent-Based Virtual Organizations". Thesis of Doctor of Philosophy. Faculty of Engineering and Applied Science. School of Electronics and Computer Science. University of Southampton. January. 2007.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[7] Canedo, " Trust Measurements Yeld Distributed Decision Support in Cloud Computing", 2012.

[8] Kaleeswaran, A., Ramasamy, V., &Vivekanandan, P. (2012). Dynamic scheduling of data using genetic algorithm in cloud computing. International Journal of Advances in Engineering & Technology, 5(2).

[9] Bilgaiyan, S., Sagnika, S., & Das, M. (2014). An Analysis of Task Scheduling in Cloud Computing using Evolutionary and Swarm-based Algorithms. International Journal of Computer Applications, 89.

[10] Selvarani, S., &Sadhasivam, G. S. (2010, December). Improved cost-based algorithm for task scheduling in cloud computing. In Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on (pp. 1-5). IEEE.

[11] Lee, Z., Wang, Y., & Zhou, W. (2011, August). A dynamic priority scheduling algorithm on service request scheduling in cloud computing. In Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on (Vol. 9, pp. 4665-4669). IEEE.

[12] 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.

[13] 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.

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

[15] 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.

[16] Afrabandpey, H., Ghaffari, M., Mirzaei, A., &Safayani, M. A Novel Bat Algorithm Based on Chaos for Optimization Tasks.

[17] Yang, X. S., Karamanoglu, M., & Fong, S. (2012, December). Bat algorithm for topology optimization in microelectronic applications. In Future Generation Communication Technology (FGCT), 2012 International Conference on (pp. 150-155). IEEE.

[18] Jula, A., Othman, Z., &Sundararajan, E. (2013, April). A hybrid imperialist competitive-gravitational attraction search algorithm to optimize cloud service composition. In Memetic Computing (MC), 2013 IEEE Workshop on (pp. 37-43). IEEE.

[19] Swain, R. K., Meher, K. C., & Mishra, U. C. (2012, December). Dynamic economic dispatch using hybrid gravitational search algorithm. In Power, Control and Embedded Systems (ICPCES), 2012 2nd International Conference on (pp. 1-6). IEEE.

[20] Zhao, C., Zhang, S., Liu, Q., Xie, J., & Hu, J. (2009, September). Independent tasks scheduling based on genetic algorithm in cloud computing. In Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on (pp. 1-4). IEEE.

[21] Li, X., Jiang, X., Huang, P., & Ye, K. (2012, October). DartCSim: An enhanced user-friendly cloud simulation system based on CloudSim with better performance. In Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on (Vol. 1, pp. 392-396). IEEE.

[22] 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.

[23] 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.

[24] Gandomi, A. H., Yang, X. S., Alavi, A. H., &Talatahari, S. (2013). Bat algorithm for constrained optimization tasks. Neural Computing and Applications, 22(6), 1239-1255.

[25] FisterJr, I., Fister, D., & Yang, X. S. (2013). A hybrid bat algorithm. arXiv preprint arXiv:1303.6310.

[26] Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010) (pp. 65-74). Springer Berlin Heidelberg.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[27] Sabri, N. M., Puteh, M., &Mahmood, M. R. (2013). A Review of Gravitational Search Algorithm. Int. J. Advance. Soft Comput. Appl, 5(3).

[28] Eldos, T., &Qasim, R. A. (2013). On The Performance of the Gravitational Search Algorithm. International Journal of Advanced Computer Science & Applications, 4(8).

[29] Momani, M., Challa, S., &Alhmouz, R. (2010). Bayesian fusion algorithm for inferring trust in wireless sensor networks. Journal of Networks, 5(7), 815-822.