Workflow Scheduling for Intelligent Modern Medical System with Privacy Protection Constraints for Iaas Cloud Using Modified Grasshopper Algorithm

T R SARAVANAN
SRM Institute of Science and Technology
Kalaivani Jeyaraj (kalaivaj@srmist.edu.in)
SRM Institute of Science and Technology https://orcid.org/0000-0001-7300-3999
G Nagarajan
Sathyambama Institute of Science and Technology: Sathyabama Institute of Science and Technology

Research Article

Keywords: Privacy protection, Iaas Cloud, resources provisioning

Posted Date: February 2nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1303167/v1

Version of Record: A version of this preprint was published at Soft Computing on June 21st, 2023. See the published version at https://doi.org/10.1007/s00500-023-08736-2.

Abstract

Cloud computing considered as perhaps the most popular figuring models. The IaaS cloud offers different processing assets to the clients on request premise. A work process in the cloud is addressed by a Direct Acyclic Graph where the hubs address the assignments and edges address the between reliance of the errand. Distributed computing offers profoundly unique climate in which the framework burden and status of asset changes often. As due to technical enhancements many people are affected with diseases a new intelligent modern medical system is developed. As the responsibility increments with expansion in Cloud Services and customers there is a need to deal with these solicitations or occupations. Effective work booking instrument with security insurance imperatives simultaneously as limiting both execution time and cost, likewise the work process planning calculations with other improvement destinations, for example, limiting energy utilization will be thought of. Assessment is done dependent on CPU time, response time, Time for Total Execution.. Finally, the show of arranged Work Scheduling Algorithm evaluates use different replication scenarios and shows the effectiveness over the existing state-of-art protocols.

1. Introduction

Cloud computing is considered as perhaps the most well known figuring models. It acquired a great deal of consideration in both scholarly community and industry fields as it gives numerous advantages to clients and associations [1]. It empowers clients to store and deal with their unpredictable applications on an enormous datacenter. It additionally permits clients to get to their information distantly from wherever [2]. The cloud climate gives three kinds of administrations like Infrastructure as a Service, Platform as a Service and Software as a Service [3]. The IaaS cloud offers different processing assets to the clients on request premise [4]. To expand the incomes, an IaaS specialist co-op gives a superior QoS to the clients according to the help level understanding [5, 6] and augments the asset use. Work process booking is one of the conspicuous issues in the distributed computing which attempts to plan the work process errands to the VMs dependent on various utilitarian and non-useful necessities [7].

A work process in the cloud is addressed by a Direct Acyclic Graph (DAG) in which the hubs address the undertakings and edges address the between reliance of the assignment [8]. Work process applications are being utilized in a scope of spaces, like astronomy, bioinformatics, and debacle displaying and forecast [9]. These applications are the instrument of an enormous scope business measure execution, comprising of a bunch of occasions or assignments in which data is disseminated starting with one errand then onto the next dependent on some specialized principles, to accomplish an overall objective [10]. The work process application assignments are subject to one another, where the yield of certain errands is the contribution to another [11]. Additionally, confounded issues like complex logical applications are arising as of late through consolidating different strategies and methods in a solitary arrangement [12]. For such a need, this kind of uses has been executed on supercomputers, groups, and frameworks [13].

Luckily, with the coming of mists, such work process applications are executed in the cloud [14]. A basic test in incorporating work process frameworks with asset provisioning advances is to decide the perfect measure of assets needed for the execution of work processes [15][17]. The asset limit influences the absolute execution season of use work process and decides the monetary expense [16]. At that point, a proficient booking calculation is needed to ideally dispatch assignments to the cloud assets. It comprises in taking the choice for planning assignments to figuring assets by upgrading execution measurements, for example, time and cost execution [17, 18]. These assets are in units of virtual machines (VMs) cases. Errand planning is a NP-complete issue in the overall structure. Thus, different methodologies dependent on heuristics have been proposed for planning work processes [19]. By the by, this issue is as yet an open examination challenge and requires a few endeavors to arrive at an ideal arrangement. Each approach considers a few measurements to guarantee assets provisioning [20].

2. Literature Review

Y. Wang et al [21] has applied a profound Q-network model in a multi-specialist support getting the hang of setting to direct the booking of multi-work processes over foundation as-a-administration mists. To improve multi-work process consummation time and client's expense, they considered a Markov game model, which takes the quantity of work process applications and heterogeneous virtual machines as state input and the most extreme fulfillment time and cost[5] as remunerations. To approve their methodology, they direct broad contextual analyses dependent on various notable logical work process formats and Amazon EC2 cloud. The exploratory outcomes unmistakably recommend that their proposed approach outflanks customary ones, e.g., non-overwhelmed arranging hereditary calculation II, multi-target molecule swarm streamlining, and game-hypothetical based covetous calculations, as far as optimality of booking plans created.

Vishakha Singh et.al[23] proposed an Energy Efficient Algorithm for Workflow Scheduling in IaaS cloud.. An obvious measure of energy can be saved if a unique voltage scaling empowered climate. proposed conspire is demonstrated to be energy effective. Aside from this, it is additionally appeared to limit make length. It is refered as productive work process booking with a novel measure to decide the measure of energy which can be saved by considering a DVS-empowered climate. proposed plot performs better compared to the current calculations, for example, HCRO and numerous need lines hereditary calculation as far as different execution measurements including makespan and the measure of energy rationed. meaning of the proposed calculation is additionally decided through the examination of fluctuation test

H. Chen et.al[24] has presents an original task arranging framework for security sensitive work processes with three novel features. To begin with, they gave far reaching hypothetical investigations on how specifically copying an undertaking's archetype assignments was helpful for thwarting both the data transmission time and encryption time from delaying task's starting time. By then, they describe work process tasks' latest fruition time, and show that tasks can be done before endeavors' latest fulfillment time by using most affordable resources for diminish monetary cost without conceding tasks' substitutions' starting time and work cycles' makespans. Considering these examinations, they devise a

clever booking approach with explicit endeavors duplication, named SOLID, joining two huge stages: 1) task arranging with explicitly replicating prime example tasks to sit time portions on resources; and 2) widely appealing data encoding by effectively manhandling tasks' laxity time. They evaluate our response approach through intensive execution appraisal study using both erratically made work cycles and some authentic work process follows. They results show that the proposed SOLID procedure beats existing computations to the extent makespan, monetary costs and resource usefulness.

V. Arabnejad et.al [22] has presented another heuristic planning calculation, Budget Deadline Aware Scheduling (BDAS) that tends to eScience work process booking under spending plan and cutoff time boundaries in Infrastructure as a Service (IaaS) mists. The oddity of their work was fulfilling both financial plan and cutoff time limitations while presenting a tunable expense time compromise over assorted cases. What's more, they study the soundness and power of their calculation by performing affectability investigation. The outcomes exhibit that generally speaking BDAS tracks down a feasible timetable for in excess of 40000 experiments achieving both characterized limitations: financial plan and cutoff time. Also, their calculation accomplishes a 17.0-23.8 percent upper achievement rate when contrasted with cutting edge calculations.

W. Guo et.al [25] has proposed a booking methodology for a cutoff time compelled logical work process across different mists. To limit the execution cost of the work process while fulfilling its time constraint, their procedure uses the discrete molecule swarm advancement strategy, and embraces arbitrarily two-point hybrid administrator and haphazardly single point transformation administrator of the hereditary calculation. Additionally, the technique streamlines the exhibition for both calculation cost and information move cost across numerous mists. Their methodology was assessed through notable work processes, and trial results show that it performs better compared to other best in class methodologies.

Z. Zhu et.al [27] has displayed the work process booking issue which smoothes out both make reach and cost as a Multi-target Optimization Problem (MOP) for the Cloud conditions. They propose a developmental multi-target enhancement (EMO)- based calculation to tackle the work process planning issue on a foundation as a help (IaaS) stage. Novel plans for issue explicit encoding and populace introduction, wellness assessment and hereditary administrators were proposed in the calculation. Broad investigations on certifiable work processes and arbitrarily created work processes show that the timetables delivered by our developmental calculation present greater steadiness on a large portion of the work processes with the case based IaaS registering and estimating models. The outcomes additionally show that their calculation can accomplish essentially preferable arrangements over existing cutting edge QoS enhancement booking calculations by and large. The directed tests depended on the on-request occurrence kinds of Amazon EC2; nonetheless, the proposed calculation were not difficult to be reached out to the assets and evaluating models of different IaaS administrations.

S. Esteves et.al[26] has presented another planning measure, Quality-of-Data (QoD), which depicts the necessities about the information that are deserving of the setting off of assignments in work processes. In view of the QoD idea, they propose a novel help situated scheduler organizer, for consistent

information preparing work processes, that is fit for implementing QoD imperatives and guide the booking to achieve asset effectiveness, generally controlled execution and assignment prioritization.

3. Proposed Methodology

Cloud Computing offers wide calculation and asset offices for execution of different application work processes. A wide range of assets engaged with execution of single work process. Distributed computing offers exceptionally unique climate in which the framework burden and status of asset changes oftentimes. As the responsibility increments with expansion in Cloud Services and customers there is a need to deal with these solicitations or occupations. The execution of cloud work processes faces numerous unsure factors in apportioning and planning responsibility. Work process booking model will plan occupations so that every one of the positions will get executed taking insignificant conceivable time, keep up QoS and fulfill customer's necessities.

3.1. Problem Identification

Work process planning for cloud computing gets one of the noticeable issues, which is NP-hard and pointed toward discovering appropriate plans for planning the work process assignments to broadly disseminated assets dependent on various utilitarian and non-practical prerequisites. Despite the fact that there are numerous papers identified with work process planning, almost no examination consider the security insurance necessities when settling on asset allotment choices for enormous information applications, e.g., security assurance requirements which confine related undertakings or information of cloud clients to being handled in explicit at least one cloud datacenters because of some security and protection reasons. Creators in [28] have proposed a canny water drop (IWD) calculation for work process planning task, nonetheless, the creator in [29] says that the principle algorithmic segments of IWD are rearrangements or uncommon instances of insect province improvement (ACO), and in this way, IWD is essentially a specific launch of ACO and the wellspring of motivation of IWD, is superfluous, deluding and dependent on unconvincing suppositions of waterway elements and soil disintegration that do not have a genuine logical reasoning intricacy of the calculation is high. These above measures spurred us to do the exploration around here.

3.2. Framework Architecture

In this paper we intend to propose an efficient work scheduling mechanism with privacy protection constraints at the same time as minimizing both execution time and cost, in addition

The above Fig. 1 clearly illustrates the proposed modified Flow architecture. The workflow scheduling algorithms[6] with other optimization objectives such as minimizing energy consumption will be considered. One of the thought-provoking disputes in the cloud data focal point is to choice the lowest numeral of virtual machine (VM) case in point to accomplish the tasks of a workflow in the interior a time limit. The intentions of such a approach are to diminish the over-all accomplishment period of a workflow and progress resource exploitation. As a result, we formulate an operative VM obligation strategy to

progress effectual possessions application. At this time we employ a reformed variety of metaheuristic algorithm [30] to accomplish higher resource consumption and curtail the make span in the interior the agreed deadline and constraints. The first influence of our proposed algorithm is to find numerous fractional critical paths (PCPs) of a workflow which helps in verdict appropriate VM instances. The second contribution is a scheduling[18] policy for PCP-VM obligation for turning over the VM instances. To conclude, the proposed algorithm is estimated over and done with a number of simulation runs with synthetic datasets and a number of routine metrics.

4. Results And Discussion

The output effects of both current and proposed (Modified Grasshopper Algorithm) strategies for safe access to cloud PHR data are discussed in here. The recommended approaches are tested here and contrasted with some of the current techniques. The simulation is carried out for the processes involved in the proposed work for secure accessing. In the cloud database, the simulation analysis takes place by LAMP (Linus, Apache, MySQL, and PHP) server in Amazon Web Service (AWS) Cloud. The personal health record information is retrieved by using the MySQL database. The results is obtained for the proposed method along with the existing methods of Dynamic Searchable Symmetric Encryption (DSSE), Security Hadoop Distributed File System Sec (HDFS) and Tree Based Access Control (TBAC) with the proposed Modified Grasshopper Algorithm.

4.1 CPU Time

The Table 1 gives the CPU time of the proposed system. The CPU time keeps on increasing in a linear way with the increase of the size of the key. The measurement of the CPU time is in milli-seconds. The overall processing time of 5067 milli-seconds is obtained for 256 bits to 2048 bits size of the key. For example, the key size of 256 bits produces 26 milli-seconds for CPU time. The CPU time for the key size of 2048 bits is 4419 milli-seconds. All the above metrics if for the proposed system.

Table 1
Evaluation of CPU Time

Size of the Key	CPU Time (in Milli-seconds)			
	DSSE	Sec (HDFS)	TBAC	Proposed
256 Bits	33	35	29	26
512 Bits	86	85	76	73
1024 Bits	612	583	559	549
2048 Bits	5789	4912	4443	4419
Overall CPU Time	6520	5615	5107	5067

The similar metrics is obtained for the existing methods also for DSSE, Sec (HDFS) and TBAC. Of all the available methods in Table 1, the proposed method is better than the existing methods which is considered for comparison.

The Table 1 and Figure 1 indicates the CPU time by different key sizes as like 33 milli-seconds for DSSE, 35 milli-seconds for Sec (HDFS), 29 milli-seconds for TBAC and 26 milli-seconds for the proposed with key size as 256 Bits. In case of 512 Bits the evaluation is as 86 milli-seconds for DSSE, 85 milli-seconds for Sec (HDFS), 76 milli-seconds for TBAC and 76 milli-seconds for the proposed. The next is by considering the key size of 1024 Bits, 612 milli-seconds for DSSE, 583 milli-seconds for Sec (HDFS), 559 milli-seconds for TBAC and 549 milli-seconds for the proposed. Finally, for the key size of 2048 Bits, 5789 milli-seconds for DSSE, 4912 milli-seconds for Sec (HDFS), 4443 milli-seconds for TBAC and 4419 milli-seconds for the proposed. In Addition to the above key sizes the CPU time evaluation is consolidated as Overall CPU time as in Figure 2, as 6520 milli-seconds for DSSE, 5615 milli-seconds for Sec (HDFS), 5107 milli-seconds for TBAC and 5067 milli-seconds for the proposed. Of the final evaluation by considering the values of Table 1, Figure 1 and 2 it is evident and clear that the proposed system outperforms all the existing methods considered.

4.2 Response Time

The Table 2 gives the Response time of the proposed system. The Response time also keeps on increasing in a linear way with the increase of the size of the key as like Response time. The measurement of the Response time is also in milli-seconds. The overall response time of 6711 milli-seconds is obtained for 256 bits to 2048 bits size of the key. For example, the key size of 256 bits produces 29 milli-seconds for Response time. The Response time for the key size of 2048 bits is 4619 milli-seconds. All the above metrics if for the proposed system. The similar metrics is obtained for the existing methods also for DSSE, Sec (HDFS) and TBAC. Of all the available methods in Table 2, the proposed method is better than the existing methods which is considered for comparison.

Table 2
Evaluation of Response Time

Size of the Key	Response Time (in Milli-seconds)			
	DSSE	Sec (HDFS)	TBAC	Proposed
256 Bits	35	36	32	29
512 Bits	164	160	161	155
1024 Bits	2287	2290	2126	2108
2048 Bits	5109	5008	4981	4619
Overall Response Time	7595	7494	7300	6911

The Table 2 and Figure 3 indicates the response time by different key sizes as like 35 milli-seconds for DSSE, 36 milli-seconds for Sec (HDFS), 32 milli-seconds for TBAC and 29 milli-seconds for the proposed

with key size as 256 Bits. In case of 512 Bits the evaluation is as 164 milli-seconds for DSSE, 160 milli-seconds for Sec (HDFS), 161 milli-seconds for TBAC and 155 milli-seconds for the proposed.

The next is by considering the key size of 1024 Bits, 2287 milli-seconds for DSSE, 2290 milli-seconds for Sec (HDFS), 2126 milli-seconds for TBAC and 2108 milli-seconds for the proposed. Finally, for the key size of 2048 Bits, 5109 milli-seconds for DSSE, 5008 milli-seconds for Sec (HDFS), 4981 milli-seconds for TBAC and 4619 milli-seconds for the proposed.

In Addition to the above key sizes the response time evaluation is consolidated as Overall response time as in Figure 4 as 7595 milli-seconds for DSSE, 7494 milli-seconds for Sec (HDFS), 7300 milli-seconds for TBAC and 6911 milli-seconds for the proposed. Of the final evaluation by considering the values of Table 2, Figure 3 and 4 it is evident and clear that the proposed system outperforms all the existing methods is taken for consideration.

4.3 Time for Total Execution

The Table 3 depicts the time in milli-seconds for total execution of the proposed method with existing DSSE, Sec (HDFS) and TBAC approaches. The time for total execution keeps on increasing with increase in number of users of users. The overall execution time of 4441 milli-seconds is obtained for 5 users to 25 user numbers. For example, if the number of users is 5 and the execution time is 120 milli-seconds consequently for 10 users, the execution time is 360 milli-seconds, 15 users the execution time is 790 milli-seconds, 20 users the execution time is 1365 milli-seconds, and 25 users the execution time is 1806 milli-seconds.

Table 3
Evaluation of Time for Total Execution

Number of Users (In Count)	Time for Total Execution (in Milli-seconds)			
	DSSE	Sec (HDFS)	TBAC	Proposed
5 Users	143	135	139	120
10 Users	437	408	402	360
15 Users	1097	909	923	790
20 Users	1588	1501	1514	1365
25 Users	2189	2023	2019	1806
Overall Time for Total Execution	5454	4976	4997	4441

All the above measurements are for the proposed system.

The Table 3 and Figure 5 indicates the time for total execution by different users as like 143 milli-seconds for DSSE, 135 milli-seconds for Sec (HDFS), 139 milli-seconds for TBAC and 120 milli-seconds for the proposed with 5 uses. In case of 10 users the evaluation is as 437 milli-seconds for DSSE, 408 milli-

seconds for Sec (HDFS), 402 milli-seconds for TBAC and 360 milli-seconds for the proposed. The next is by considering the 15 users, 1097 milli-seconds for DSSE, 909 milli-seconds for Sec (HDFS), 923 milli-seconds for TBAC and 790 milli-seconds for the proposed.

The next is for the 20 users, 1588 milli-seconds for DSSE, 1501 milli-seconds for Sec (HDFS), 1514 milli-seconds for TBAC and 1365 milli-seconds for the proposed. Finally, for 25 users, 2189 milli-seconds for DSSE, 2023 milli-seconds for Sec (HDFS), 2019 milli-seconds for TBAC and 1806 milli-seconds for the proposed. In Addition to the values by number of users, the time for total execution evaluation is consolidated as Overall time for total execution as in Figure 6, as 5454 milli-seconds for DSSE, 4976 milli-seconds for Sec (HDFS), 4997 milli-seconds for TBAC and 4441 milli-seconds for the proposed. Of the final evaluation by considering the values of Table 3, Figure 5 and 6 it is evident and clear that the proposed system outperforms all the existing methods considered for the evaluation.

5. Conclusion

In Works stream booking The yield impacts of both current and proposed (Modified Grasshopper Algorithm) procedures for safe admittance to cloud PHR information are talked about. Proposed calculation gives better outcomes as far as execution measurements CPU time, reponse time, Time for Total Execution. least number of virtual machine (VM) occasions to execute the assignments of a work process inside a period limit challenge has been resolved. The Intelligent Medical system for Work flow scheduling is more effective comparing existing methodologies. Proposed calculation is checked for number of clients regarding milliseconds. The reaction is acceptable contrasting and existing systems. From the recreation results, we notice the utilizing engineered datasets and different execution measurements are beating the current DSSE, HDFS and TBAC strategies.

Declarations

Funding: The authors did not receive financial support from any organization for the submitted work.
Conflicts of interest/Competing interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Availability of data and material: 'Not applicable', Authors' contributions: 'Not applicable' Code availability: 'Not applicable',

Consent to participate: ‘Not applicable’

Ethics approval: Compliance with Ethical Standards

Consent for publication: Authors give consent to Soft Computing Journal to publish their article.

References

[1]Mei, L.; Chan, W.K.; Tse, T.H.: A tale of clouds: paradigm comparisons and some thoughts on research issues. Proc. APSCC 2008, 464–469 ,2008

[2] Haijun, Z.; Cao, X.; Ho, J.K.L.; Chow, T.W.S.: Object-level video advertising: an optimization framework. IEEE Trans. Ind. Inform. 13(2), 520–531, 2017

[3] Haijun, Z.; Llorca, J.; Davis, C.C.; Milner, S.D.: Nature-inspired self-organization, control, and optimization in heterogeneous wireless networks. IEEE Trans. Mob. Comput. 11(7), 1207–1222,2012

[4] J. Yu, R. Buyya, and K. Ramamohanarao, “Workflow scheduling algorithms for grid computing,” in Metaheuristics for scheduling in distributed computing environments, pp. 173–214, Springer, 2008.

[5] A. Verma and S. Kaushal, “Cost-Time Efficient Scheduling Plan for Executing Workflows in the Cloud”, Journal of Grid Computing, vol. 13, no. 4, pp. 495–506, 2015.

[6] H. Ji, W. Bao, and X. Zhu, “Adaptive workflow scheduling for diverse objectives in cloud environments,” Transactions on Emerging Telecommunications Technologies, vol. 28, no. 2, Article ID e2941, 2017.

[7] A. M. Manasrah, “Dynamic weighted VM load balancing for cloud-analyst,” International Journal of Information and Computer Security, vol. 9, no. 1-2, pp. 5–19, 2017.

[8] W.-N. Chen and J. Zhang, “An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements,” IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 39, no. 1, pp. 29–43, 2009.

[9] A. K. M. K. A. Talukder, M. Kirley, and R. Buyya, “Multiobjective differential evolution for scheduling workflow applications on global Grids,” Concurrency and Computation: Practice and Experience, vol. 21, no. 13, pp. 1742–1756, 2009.

[10] M. Wieczorek, A. Hoheisel, and R. Prodan, “Towards a general model of the multi-criteria workflow scheduling on the grid,” Future Generation Computer Systems, vol. 25, no. 3, pp. 237–256, 2009.

[11] P. Li, J. Li, Z. Huang, C.-Z. Gao, W.-B. Chen, and K. Chen,(2017) “Privacy-preserving outsourced classification in cloud computing”, Cluster Computing, pp. 1–10, 2017.

[12] C. Stergiou, K. E. Psannis, B.G. Kim, and B. Gupta, “Secure integration of IoT and Cloud Computing”, Future Generation Computer Systems, vol. 78, pp. 964–975, 2018

[13]Navjot Kaur, T. S “Comparison of workflowScheduling Algorithms in Cloud Computing”, International Journal of Advanced computer Science and Applications , 2-

10,2011

[14] Cheng, B. , “Hierarchical Cloud ServicesWorkflow scheduling Optimization schema usingHeuristic Generic Algorithm”,vol. 5, no. 2, pp. 789-795,2012.

[15] Fan Zhang, J. C., “Ordinal Optimized scheduling ofscientific workflow in Elastic Compute Clouds”,vol. 59, no. 3, pp. 371-383,2010

[16] K. Agarwal, A. B, “Scheduling Algorithms forlinear workflow optimization”, IEEE Conference , no.1 .pp.89-95, 2010

[17] Ashutosh Ingole, S. C An optimized Algorithmfor task scheduling based on activity based costing inCloud Computing. International Journal of Computer Applications(IJCA) .vol. 59, no. 3, pp. 371-383,2011

[18] Tayal, S, “Tasks Scheduling Optimization for theCloud Computing System”, International Journal of Advanced Engineering Science and Technologies (IJAEST), 111-115,2011

[19] S. Abrishami, M. Naghibzadeh, Deadline-constrained workflow scheduling in software as a service cloud, Sci. Iranica, Trans. D: Comput. Sci. Eng., Electr. Eng. 19 , 680–689,2011

[20] M.A. Rodriguez, R. Buyya, Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds, IEEE Trans. on Cloud Comput.2 , 1–14 ,2013

[21]Y. Wang et al., "Multi-Objective Workflow Scheduling With Deep-Q-Network-Based Multi- Agent Reinforcement Learning", IEEE Access, vol. 7, pp. 39974-39982, 2019.

[22]V. Arabnejad, K. Bubendorfer and B. Ng, "Budget and Deadline Aware e-Science Workflow Scheduling in Clouds", IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 1, pp. 29-44, 2019.

[23]Vishakha Singh, Indrajeet Gupta, Prasanta K. Jana , “An Energy Efficient Algorithm for Workflow Scheduling in IaaS Cloud,” Journal of Grid Computing vol.18, pp.357–376.2020

[24]H. Chen, X. Zhu, D. Qiu, L. Liu and Z. Du, "Scheduling for Workflows with Security- Sensitive Intermediate Data by Selective Tasks Duplication in Clouds", IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 9, pp. 2674-2688, 2017.

[25]W. Guo, B. Lin, G. Chen, Y. Chen and F. Liang, "Cost-Driven Scheduling for Deadline-Based Workflow Across Multiple Clouds", IEEE Transactions on Network and Service Management, vol. 15, no. 4, pp. 1571-1585, 2018.

[26]S. Esteves and L. Veiga, "WaaS: Workflow-as-a-Service for the Cloud with Scheduling of Continuous and Data-Intensive Workflows", The Computer Journal, vol. 59, no. 3, pp. 371-

383, 2015

[27]Z. Zhu, G. Zhang, M. Li and X. Liu, "Evolutionary Multi-Objective Workflow Scheduling in Cloud", IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 5, pp. 1344-1357, 2016.

[28]M. Adhikari and T. Amgoth, "An intelligent water drops-based workflow scheduling for IaaS cloud", Applied Soft Computing, vol. 77, pp. 547-566, 2019

[29]C. Camacho-Villalón, M. Dorigo and T. Stützle, "The intelligent water drops algorithm: why it cannot be considered a novel algorithm", Swarm Intelligence, 2019.

[30] Saremi, Shahrzad, Seyedali Mirjalili, and Andrew Lewis. "Grasshopper optimisation algorithm: theory and application." Advances in Engineering Software 105 (2017): 30-47,2017

Figures

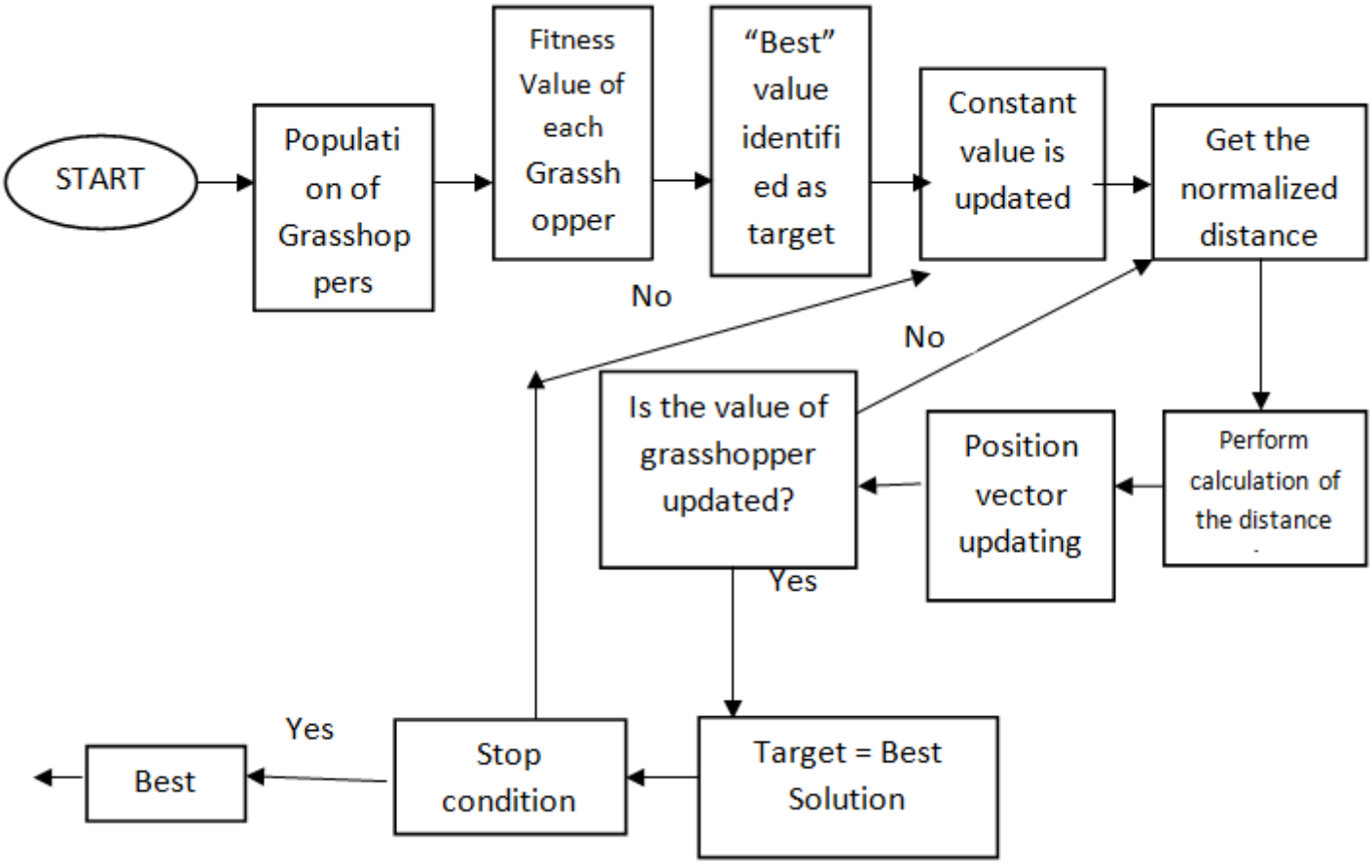

Figure 1

Proposed Workflow Scheduling Architecture

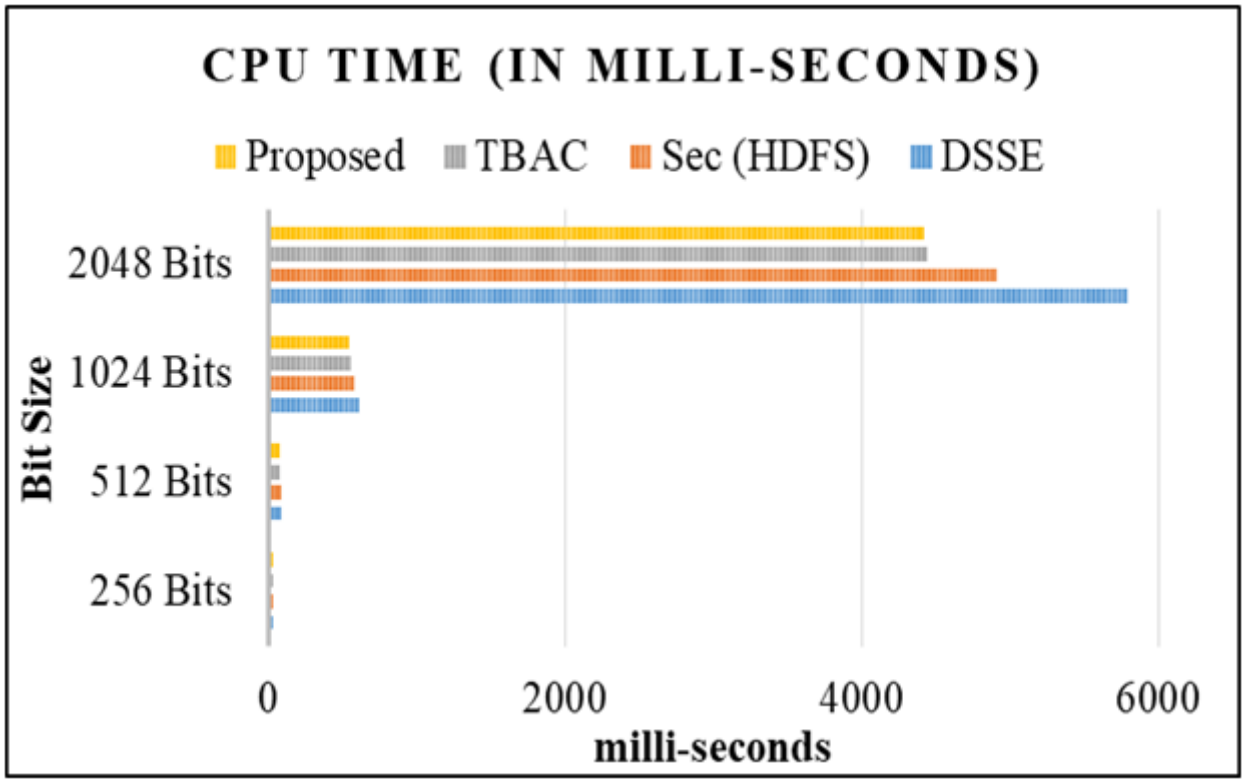

Figure 2

CPU Time in milli-seconds for varying Bit sizes

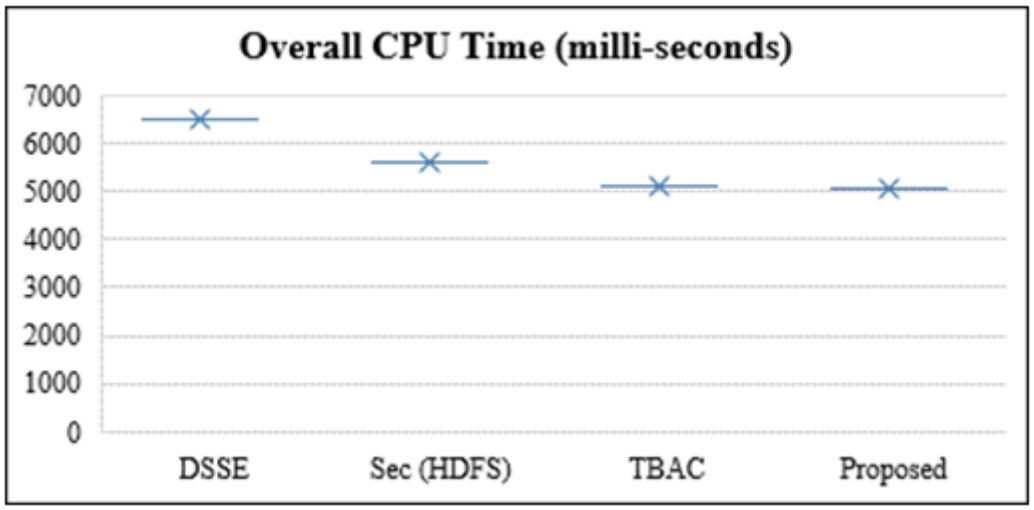

Figure 3

Overall CPU Time in milli-seconds for varying methods

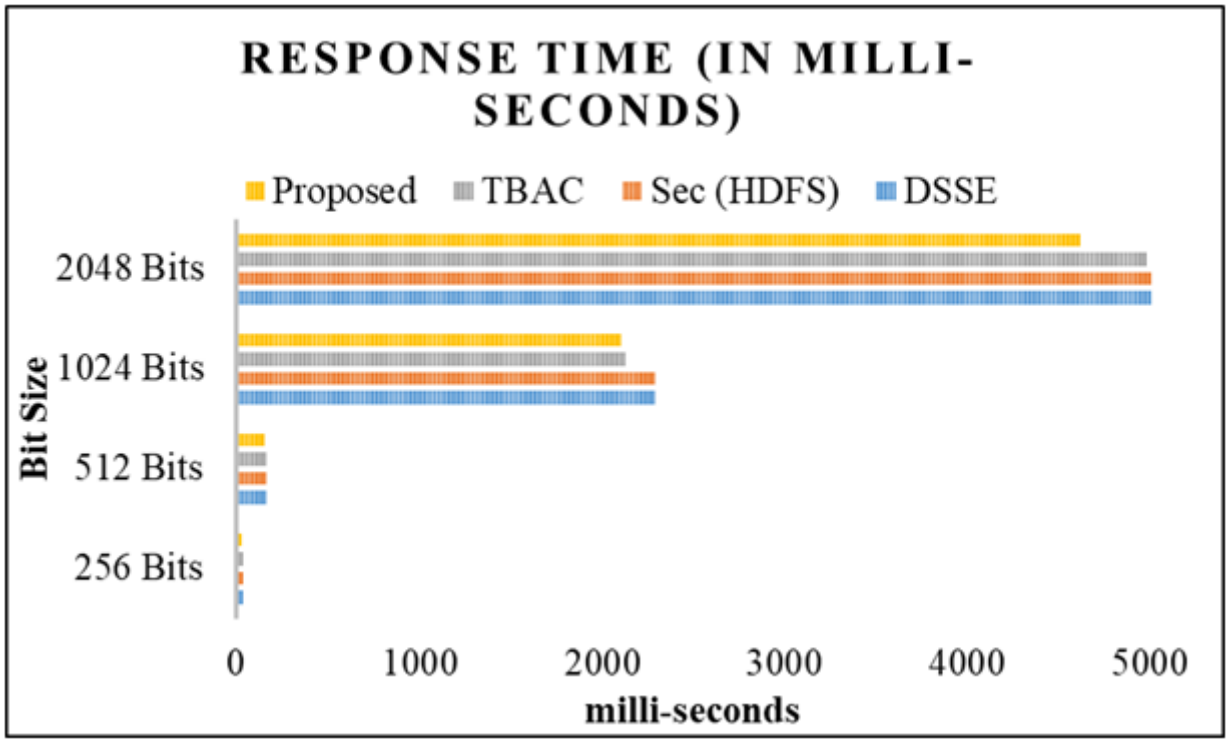

Figure 4

Response Time in milli-seconds for varying Bit sizes

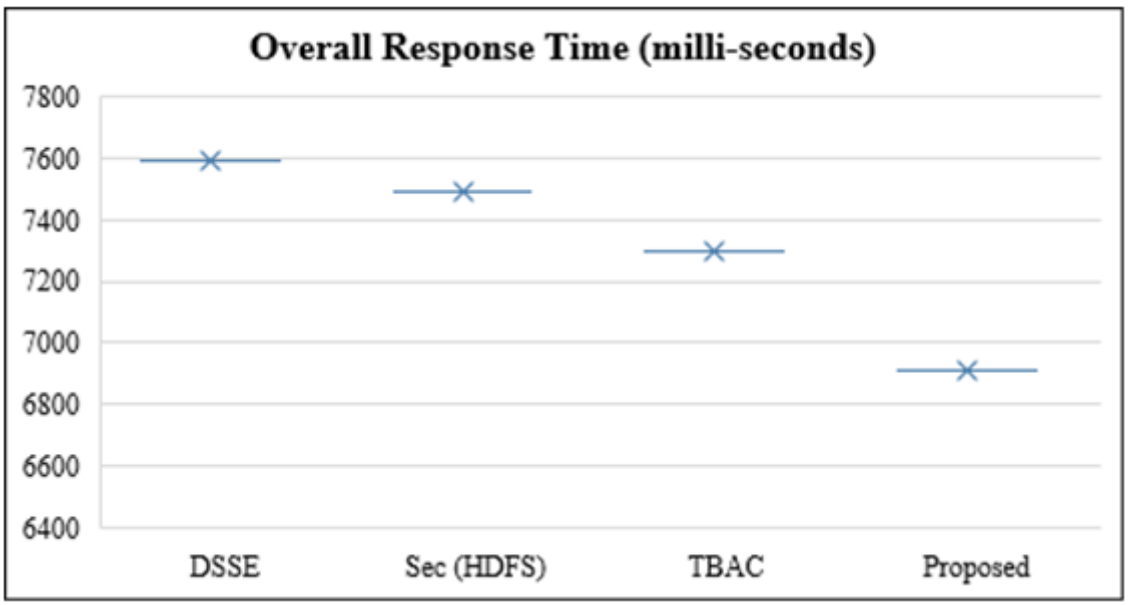

Figure 5

Overall Response Time in milli-seconds for varying methods

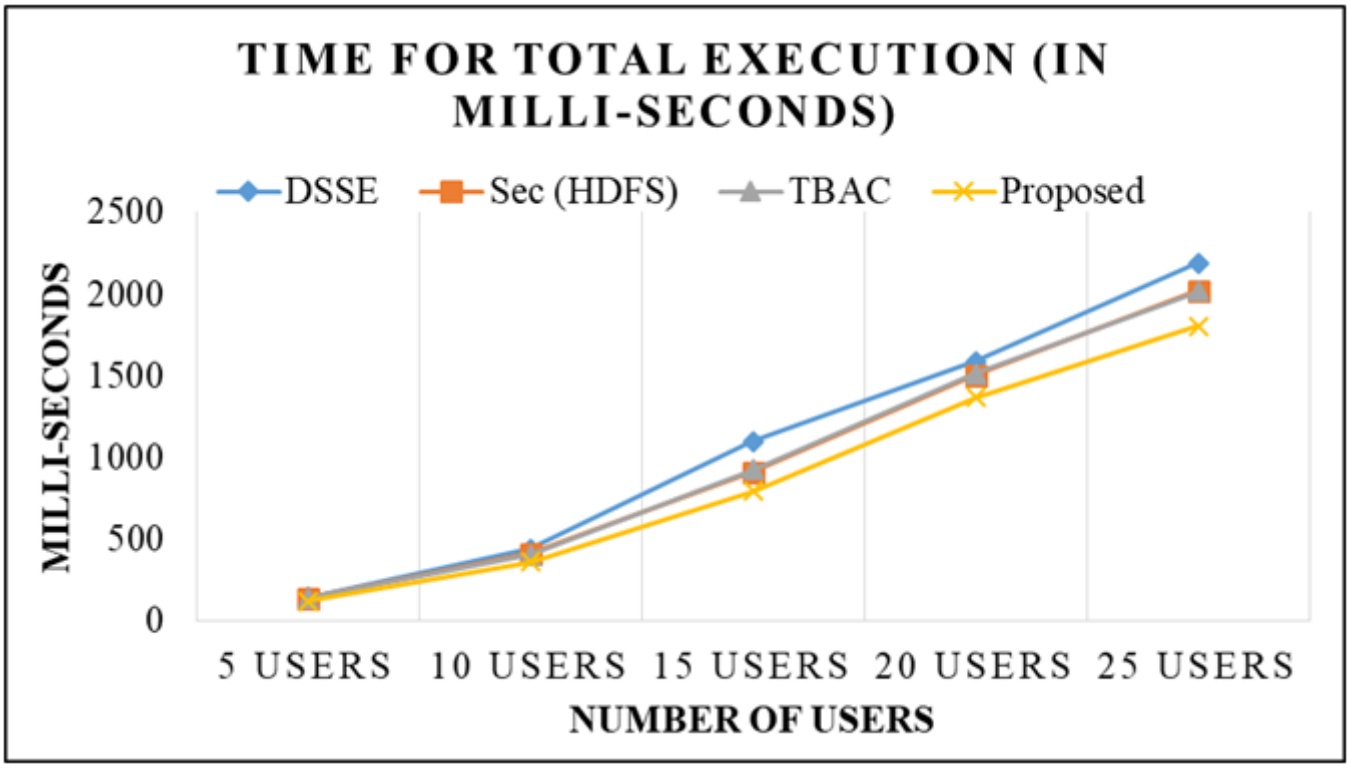

Figure 6

Time for Total Execution in milli-seconds for varying number of users

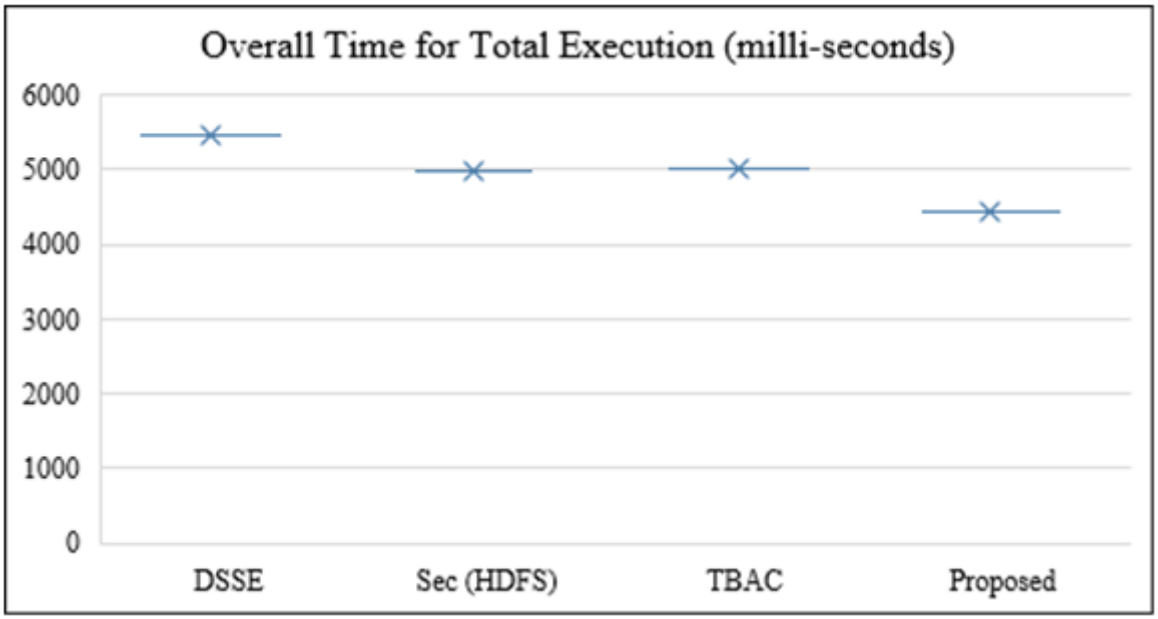

Figure 7

Overall Time for Total Execution in milli-seconds for varying methods