ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Energy-Aware Mobile Application Scheduling using Backtracking Search Algorithm

Parveen Kaur
Dept. of computer engineering and technology
GNDU
Amritsar, Punjab, India

Anil Kumar
Dept. of computer engineering and technology
GNDU
Amritsar, Punjab, India

Abstract: Remote Cloud Computing conveys the information and registering assets through the web, on a compensation for use premise. By utilizing this, we can consequently refresh our product. We can utilize just the space required for the server, which decreases the overhead in terms of makespan. Undertaking planning is the principle issue in distributed computing which lessens the framework execution. To enhance framework execution, there is need of a productive way to decrease Makespan, Flowtime, latency and execution time associated with task allocation. In this paper, we present a proficient undertaking planning calculation, which presents detachable scheme by considering system data transmission along with backtracking. By this, we can assign the work process in view of the accessibility of VMs resource capacity. Our proposed work utilizes both serial and parallel schedules for work allocation and proposes hybridization of Cloud with Parallel schedule through backtracking algorithm. Significant improvement in result in terms of Makespan and Flowtime is seen though proposed approach.

Keywords: Remote Cloud Computing, framework, Makespan, Flowtime, backtracking

I. INTRODUCTION

[1]Cloud computing is used in order to provide resources for the machines having limited resources associated with them. Hence cloud computing becomes need of the hour. The resource allocation to VMs is critical. Henceforth scheduling is required. [2]The requirements of machines greatly depends upon the processes that are available to be executed on machines. The processes along burst time hence are considered in the proposed approach for the same. For parallel execution processes are divided into tasks.

Example 1: Let Processes with process Ids , Burst time and arrival time is given as shown in table 1.

Table 1: Listing Process Ids , Burst time and arrival time

Process Id	Burst time	Arrival time
1	100	1
2	20	2
3	98	3
4	67	4
5	90	5

Let time quantum of 10 is used then process with process ID 1 is divided into 10 tasks with resource required 10, 1 for each task. This division corresponds to equation 1.

$$Task_i = Burst_{Time}/Time_{Quantum}$$

Equation 1: Task Division Mechanism

Once number of tasks are obtained, [3]cloud VMs are searched for sufficient amount of VMs. Once found allocation of task to appropriate VM is made. If VMs with sufficient resources are not found then backtracking algorithm is used in order to find effective finish time used to locate earliest freeing VM. Rest of the tasks are allocated in such manner for the optimal allocation. Results are obtained in terms of [4]–[6]Makespan,[7]–[9]Flowtime, time consumption and [10]energy consumed.

In order to describe Makespa, we consider following example:

For instance, assume the "venture" is to bolster the goats. There are three goats to encourage, and there are two youngsters that can sustain them: ‘A’ bolsters every goat in 10 minutes and ‘B’ nourishes every goat in 12 minutes. A few timetables are conceivable:

In the event that we let ‘A’ bolster all goats, then the Makespan is 30 (3×10 for A, 0 for B);

On the off chance that we let ‘B’ nourish one goat and ‘A’ two goats, then the Makespan is 20 (2×10 for ‘A’, 12 for ‘B’);

In the event that we let ‘B’ nourish two goats and ‘A’ one goat, then the Makespan is 24 (2×12 for ‘B’, 10 for ‘A’);

In the event that we let ‘A’encourage all goats, then the Makespan is 36 (3×12 for ‘B’).

Rest of the paper is organised as under. Section 2 describes the existing work or background analysis , section 3 describes the proposed system, section 4 describes the performance analysis and result, section 4 gives conclusion and future scope and last section gives bibliography or references.

II. BACKGROUND ANALYSIS

[11]The appearance of Internet of Things and 5G applications renders the requirement for reconciliation of both concentrated distributed computing and rising versatile edge registering (MEC) with existing system frameworks to upgrade stockpiling, preparing, and storing capacities in incorporated as well as disseminated styles for supporting both postponement tolerant and missioncritical applications. This paper explores execution additions of brought together cloud and MEC empowered incorporated fiber remote (FiWi) get to systems (CM-FiWi). A novel unified asset administration plot consolidating both brought together cloud and MEC calculation offloading exercises into the basic FiWi dynamic data transfer capacity allotment process is considered. [12]Portable clients regularly have popularity

on restricted and area based data administrations. To dependably recover the confined information from the remote cloud, notwithstanding, has a tendency to be inefficient, which spurs mist registering. The mist registering, otherwise called edge figuring, develops distributed computing by sending limited processing offices at the start of clients, which pre-stores cloud information and appropriates to portable clients with quick rate neighborhood associations. [13]While High Performance Computing (HPC) focuses consistently develop to give all the more figuring energy to their clients, we watch a desire for the union between Cloud Computing (CC) and HPC stages, with the business would like to see CC foundations to in the long run supplant in-house offices[6], [14]. In the event that we prohibit the execution perspective where numerous past reviews highlight a non-insignificant overhead incited by the virtualization layer at the heart of each Cloud middleware when running a HPC workload, the subject of the genuine costeffectiveness is regularly left aside with the instinct that, most presumably, the cases offered by the Cloud suppliers are focused from a cost perspective.

[15]With explosive development of cell phones including PDAs, PDAs, and tablet PCs and the applications introduced in them, the portable Internet will keep up the improvement development slant as 4G correspondence system is broadly elevated to our lives. What clients of the cell phones and applications need is that portable Internet can give them the administration which is easy to understand, highspeed, and consistent. Dynamic Resource allocation proposed by [14] is Distributed computing. It enables business clients to scale here and there their asset use in light of requirements. Large portions of the touted picks up in the cloud show originated from asset multiplexing through virtualization innovation.[16] In this paper, we display a framework that utilizations virtualization innovation to distribute server farm assets progressively in view of use requests and bolster green processing by upgrading the quantity of servers being used. This [17], [18]proposes the idea of "skewness" to quantify the unevenness in the multidimensional asset use of a server. Mostly this scheme of thing is effective in detecting attacks. By limiting skewness, we can join distinctive sorts of workloads pleasantly and enhance the general usage of server assets. We build up an arrangement of heuristics that anticipate over-burden in the framework successfully while sparing vitality utilized. Follow driven recreation and test comes about show that considered calculation accomplishes great execution. Distributed computing as proposed by [19] is efficient in order for the jobs to execute efficiently in Cloud based system. Distributed computing offers utility-arranged IT administrations to clients around the world. In light of a compensation as-you-go display, it empowers facilitating of unavoidable applications from shopper, logical, and business spaces. Be that as it may, server farms facilitating Cloud applications devour colossal measures of electrical vitality, adding to high operational expenses and carbon impressions to the earth. Subsequently, we require Green Cloud figuring arrangements that can limit operational expenses as well as decrease the natural effect. In this paper, we characterize an engineering system and standards for vitality proficient Cloud registering. In view of this engineering, we show our vision, open research difficulties, and asset provisioning and allotment calculations for vitality proficient administration of Cloud figuring situations. The proposed vitality mindful portion heuristics arrangement server farm assets to customer applications in a way that enhances vitality proficiency of the server in cloud based system, while conveying the arranged Quality of Service (QoS). Effective Grid computing as suggested by [20] is used for job allocation in advanced computing system. Advanced computing involves legions of nodes or VMs of different configurations. Heterogeneous environment form grids.Nodes of same configuration are grouped into same batch known as cluster. Scheduling amon[21]g such nodes are considered effectively to reduce Makespan and Flowtime. Hybrid job scheduling is proposed by [22]–[24]. Ant colony optimization is suggested through the considered literature along with ordinal scheduling to enhance scheduling mechanism. Once scheduling of resources is made, task execution begins. Disadvantage of ant colony is rectified through the use of ordinal scheduling mechanism. Parameters considered in such environment are Makespan and Flowtime.[25]Job scheduling is a procedure of apportioning framework assets to various undertakings by utilizing working framework. The framework handles organized job lines that sit tight for CPU time and it ought to likewise figure out which job to be executed first from rundown of jobs. By utilizing the above criteria jobs can be executed in a reasonable way.

[26]Job scheduling is performed by utilizing job schedulers. Job schedulers are projects which empower scheduling and, now and again, track PC "cluster" jobs, or units of work as like the operation of a finance program. Job schedulers have the capacity that they can begin and control jobs consequently by running arranged job-control-dialect proclamations or by methods for comparative correspondence with a human administrator. By and large, the present-day job schedulers incorporate a graphical user interface (GUI) alongside a solitary purpose of control.[27], [28] Parallel processing is utilized as a part of job scheduling because of a few reasons, i.e. it give simultaneousness, spare time, tackle bigger issues, expand stack adjusting and make a decent utilization of parallel equipment engineering. In multiprocessor condition parallel processing has two sorts of processors heterogeneous and homogeneous, in heterogeneous the processors are of various sort of speed and cost while in homogenous there are same sort of processors in all points of view.[29], [30]The processor that utilized inside the parallel calculation could be of comparative sorts of different sorts. The comparative kind of processor utilized inside the system shape cluster processors and unique sorts of processor inside the system are known as vector processors. [31]The processors can be utilized inside the system executing guidelines either in serial or parallel way. The work has been redirected from people to machines. [32], [33]The human workload has been occupied in light of the fact that that much load can't be accommodated on the single client. Same is the situation with the machines. [34], [35]Presently days cloud is utilized frequently henceforth the heap on the cloud framework is expanding. The cloud will utilize server farms and with the expansion in number of clients the heap on server farms is expanding. All things considered if the server farm goes down then every one of the information put away over the VMs(Virtual Machine) will be lost. The

proposed paper will recommend the strategy for offloading the information to numerous server farms. In the proposed paper relocation will be performed by the utilization of live VM movement which implies that the relocation does not required to turn off the gadgets which is the situation in disconnected movement. Offloading is a viable strategy for augmenting the lifetime of handheld cell phones by executing a few parts of utilizations remotely (e.g., on the server in a server farm or in a cloud). To accomplish vitality sparing while at the same time fulfilling given application execution time necessity, we display a dynamic offloading calculation, which depends on Lyapunov enhancement. The technique expounded has low many-sided quality to take care of the offloading issue (i.e., to determine which programming segments to execute remotely given accessible remote system network).[17], [36]The offloading in versatile registering will fill in as system for putting away a bigger number of information inside the cell phone than its ability. The portable registering offloading ends up noticeably basic in a circumstance where bigger applications are need of great importance which are to be executed on the versatile framework. Calculation Offloading, sending computational errands to more ingenious servers, is turning into a generally utilized way to deal with spare restricted assets on cell phones like battery life, stockpiling, processor, and so forth. Given an application that is parceled into various assignments, the offloading choices can be made on each of them. Nonetheless, considering the postpone limitation and the additional expenses on information transmission and remote calculation, it is not unimportant to settle on improved choices. Existing works have planned offloading choice issues as either diagram apportioning or twofold number programming issues. The main approach can tackle the issue in polynomial time yet is not material to postpone requirements. The second approach depends on a whole number programming solver without a polynomial time ensure. We give a calculation, DTP (Deterministic postpone obliged Task Partitioning), to take care of the offloading choice issue with defer requirements. DTP gives close ideal arrangement and keeps running in polynomial time in the quantity of tasks. Going past earlier work on direct defer limitations that apply just to serial assignments, we sum up the postpone imperatives to settings where the reliance between errands can be portrayed by a tree. Besides, we give another calculation, PTP (Probabilistic defer obliged Task Partitioning), which gives more grounded QoS ensures. Recreation comes about demonstrate that our calculations are precise and vigorous, and scale well with the quantity of undertakings.

III. PROPOSED SYSTEM

Proposed approach uses allocation of resources parallel with backtracking mechanism in place for allocating resources by finding out spare virtual machines with sufficient resources. This section describes proposed system with parameters evaluation.

[37]Mobile Cloud Computing (MCC) develops distributed computing with the focal points of portability and remote systems to make another framework where cloud assumes control cell phones' duties of executing errands and putting away huge measures of information. Through offloading, all the real information preparing work happens in the cloud rather than the cell phones. The fundamental point of MCC is to accomplish a rich client encounter by empowering extensive variety of cell phones to execute rich versatile applications. Booking of errands require least consummation time, better execution, powerful usage of assets and brisk reaction time for which cloud utilizes virtualization idea. For undertaking assignment, cloud gives virtual machines which are adaptable yet booking them while effectively using the sit without moving benefit limits of the cell phones are still stays significant issue. In like manner, there are different issues confronted in MCC, for example, deficient asset, low network also, constrained vitality because of which using its full ability is a test. The existing application booking calculations in MCC don't take each undertaking's benefit or the general vitality utilization of cell phones into thought. Likewise it can't increment the benefit of the framework, which is an import focus for planning the undertakings in business portable cloud condition. In this paper, E-MACS (Energy-mindful Portable Application Consolidation and Scheduling) calculation is proposed to make the cell phones contribute their figuring and detecting capacities to achieve effective booking of utilization in half and half cloud display. The combination of application limits the general vitality utilization in cloudlet. The proposed framework limits the reaction inactivity, cost of use movement and it enhances nature of administration like all through and adaptability among assets utilizing stack adjusting methods by portable distributed computing.

Energy consumption in hybridization of [38]Energy Aware Mobile Application Consolidation and Scheduling with [39], [40]BSA(Backtracking search) identified the need of Resources. If the need is less hence categorize the job as sequential and job is executed on local machine. Otherwise VMs having appropriate resources are located through BSA. By following the purposed approach, the energy consumption is sufficiently reduced.

Algorithm for EMACBSA is as under:

```
ALGORITHM-EMACBSA
1. Deploy mobile cloud
   • Initialize Data centers, and allocate configuration to VMs
   • Arrange VMs with maximum power first and least power at last place.
2. Initialize Users
   • Assign users of mobile cloud with username and password
   • Validate the users according to authentication
   • If validated move to 3 step otherwise perform 2 step again
3. Load jobs externally
```

- Jobs are loaded from external file containing processes and burst time

4. Divide jobs into tasks.
   - $Tasks_i = Jobs_i / VM_{capacity}$
5. If available(resources)
   - Execute job sequentially selecting VMs according to resource requirement and obtain makespan

   Else
   - Execute jobs parallely and obtain makespan

   End of if
6. If(optimal(Makespan))

   Result=Makespan

   Else

   Perform Backtracking

   End of if
7. Obtain result in terms of Makespan

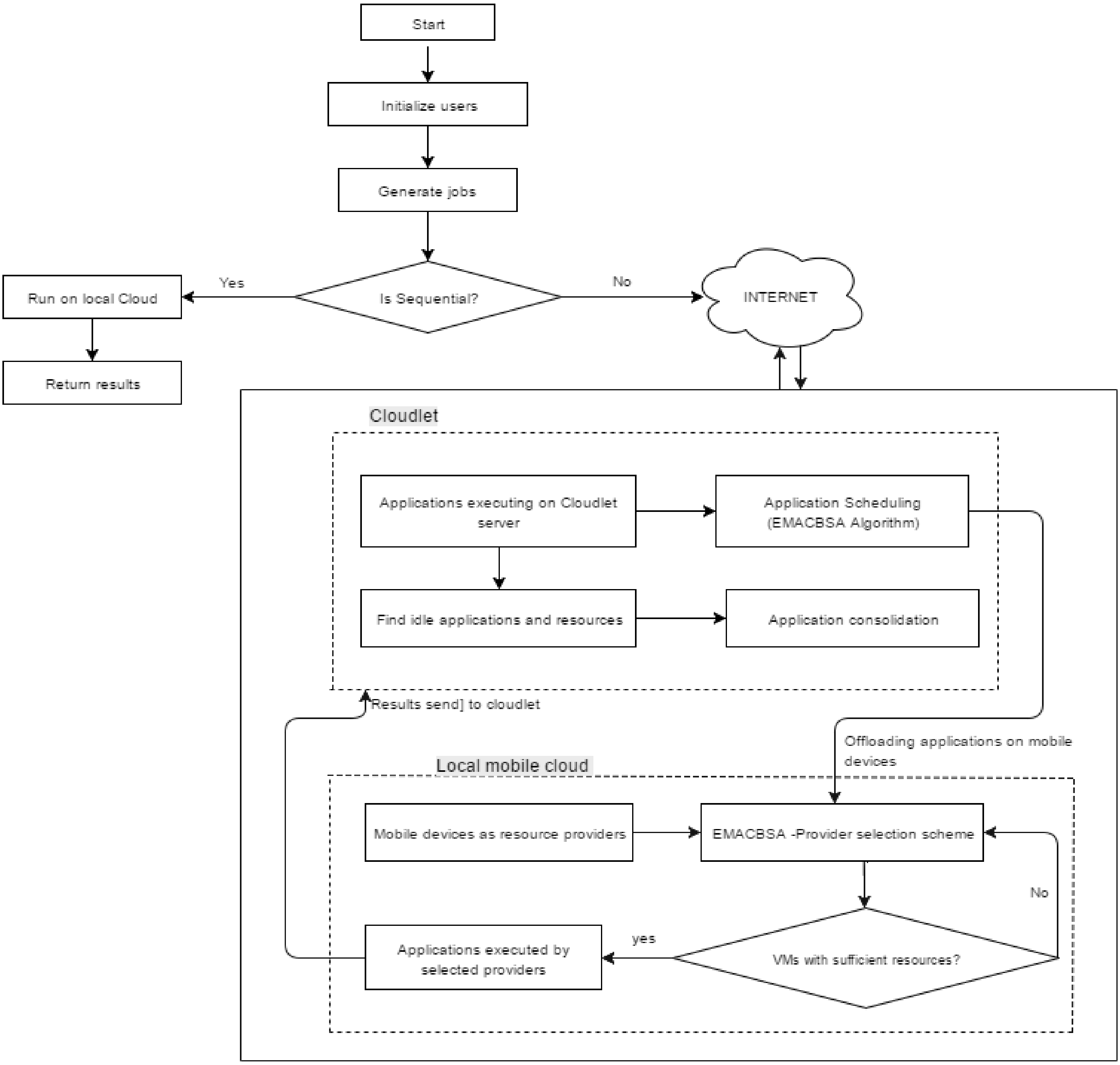
Start
Initialize users
Generate jobs
Is Sequential?
Yes
No
Run on local Cloud
Return results
INTERNET
Cloudlet
Applications executing on Cloudlet server
Application Scheduling (EMACBSA Algorithm)
Find idle applications and resources
Application consolidation
Results send] to cloudlet
Offloading applications on mobile devices
Local mobile cloud
Mobile devices as resource providers
EMACBSA -Provider selection scheme
No
yes
VMs with sufficient resources?
Applications executed by selected providers

Figure 1: Proposed Methodology

IV. PHASES OF EMACBSA

There are following phases associated with proposed literature

A. *Initilization*

The jobs are partitioned into tasks. These tasks are referred to as[22], [41] chromosomes. These chromosomes are in hunt of resources acting as food for chromosomes. Initially these chromosomes are set according to order presented through cloudlets. These cloudlets considered for proposed operation includes jobs with burst time and arrival time.

$$Chromosomes = Tasks(t_1, t_2, - - - -, t_n)$$

Equation 2: Chromosomes in terms of tasks.

B. *Objective function*

The objective function consist of Makespan. The BSA has the implication in reducing Makespan. The earlier approach(EMAC) gives Makespan which serves as baseline or threshold value. Main objective thus is to minimize the Makespan. The schedule is executed again and again until Makespan lesser than Makespan obtained through EMAC is obtained.

$$obj_{function} = \min(makspan_{BSA})$$

Equation 3: objective function

C. *Mutation and Crossover*

[41], [42]As the initialization is complete, process of allocating resources to task begins. Makespan is evaluated at each step. If the Makespan is more than Makespan obtained through EMAC then mutation with probability 0.5 is made. Flip bit mutation and random crossover is employed. Allocation is done again and Makespan is evaluated. This process continues until optimal schedule of allocation is accomplished.

D. *Parameter Evaluated*

Parameters evaluation is key aspect and prime parameter consider is Makespan and energy consumption. Objective of BSA is to minimize the both.

1. *Makespan*

Makespan is the sum of total burst time associated with the jobs. In other words, it is total time taken to execute last job.

$$Makespan = \sum Job_{BT_i}$$

Equation 4: Makespan Evaluation

2. *Flowtime*

This parameter is evaluated by considering the finish time of individual task.

$$Flowtime_i = BT_i$$

Equation 5: Flowtime evaluation which is equal to Burt time of schedule

3. *Energy consumption*

The finished time of each job is evaluated to calculate the energy consumption. As the job is executed on VM, energy is consumed. The finished time associated with task is added and then mean is obtained to calculate energy consumption.

$$Energy_i = \sum Finished_{Time_i}/total_{jobs}$$

Equation 5: Energy Consumption

4. *Latency*

Latency is given in terms of waiting time. Latency indicates the delay in execution of the jobs. This delay is calculated in terms of waiting time.

$$Latency_i = \sum Wt_i/total_{jobs}$$

Equation 6: Average latency evaluation

5. *Time Consumption*

This parameter indicated time it takes to execute individual task over the VM. More time consumption indicated more energy dissipation.

$$Time_{Consumption_i} = Finish_{time_i} - Start_{time_i}$$

Equation 7: Time Consumption Evaluation

V. PERFORMANCE ANALYSIS AND RESULTS

Performance analysis is done by evaluating the existing system i.e. EMAC and comparing it with EMACBSA. The performance analysis and results are describe in terms of the following parameters:

A. *Makespan*

Table 2: Makespan of Schedule

Number of jobs	*EMAC*	*EMACBSA*
10	2532	1559
8	1181	953
6	457	273
4	254	212

The figure 2 shows the comparison of makespan of the EMAC and EMACBSA algorithm. EMACBSA has better makespan because the concept of backtracking is used. If

VMs with sufficient resources are not found then backtracking is done.

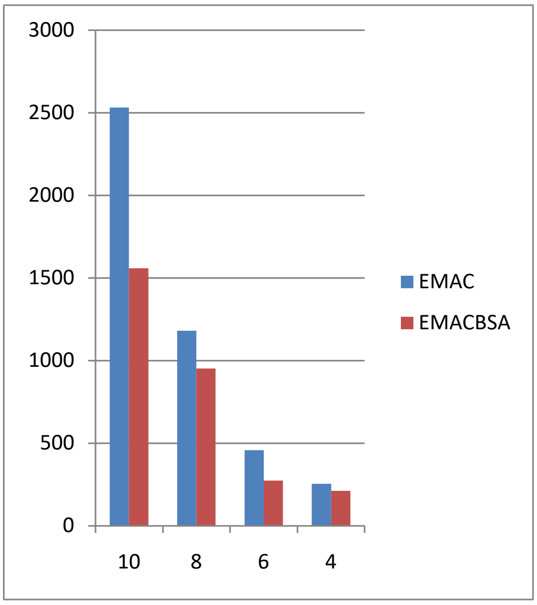

Figure 2: Plot of Makespan of EMAC and EMACBSA.

B. *Flowtime*

Table 3: Flowtime Evaluation of EMAC and EMACBSA.

Number of jobs	EMAC	EMACBSA
10	300	118
8	214	69
6	168	49
4	146	32

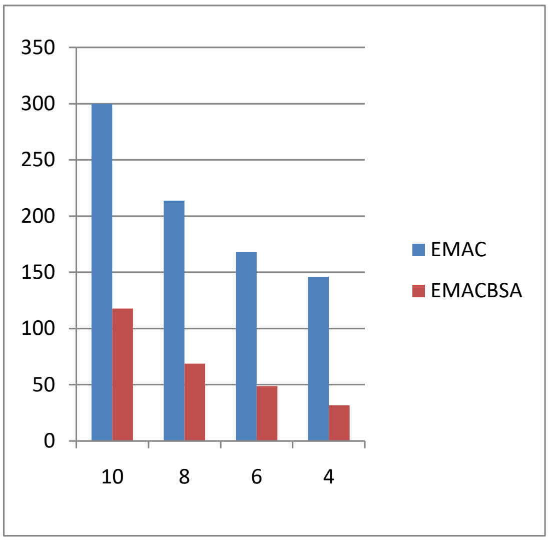

Figure 3: Plot of Flowtime

Flowtime and Makespan shows considerable improvement in case of EMACBSA.

C. *Energy Consumption*

Table 4: Energy Consumption of EMAC and EMACBSA

Number of jobs	***EMAC***	***EMACBSA***
10	267	83
8	195	54
6	62	26
4	39	17

Figure 4 shows the energy consumption of EMAC and EMACBSA. It shows that the purposed EMACBSA reduces the overall energy consumption. Plot of energy consumption of EMAC and EMAC BSA is as under

The time consumption and average latency values obtained from EMACBSA is also better. For 10 jobs EMACBSA has latency of 9ms where as EMAC has latency 51ms and time consumption of EMAC is 5ms where as time consumption EMACBSA is 2 ms. Overall performance on scale of distinct parameters, EMACBSA is better than its predecessor EMAC.

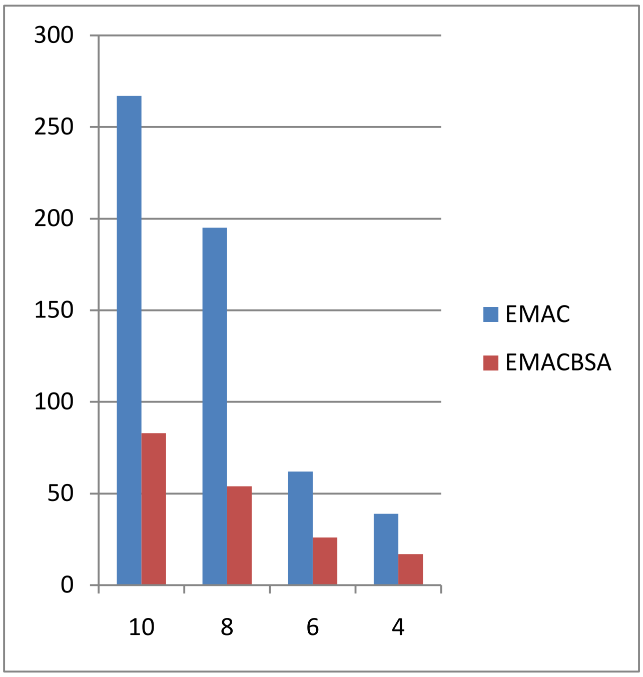

Figure 4: Energy Consumption of EMAC and EMACBSA

VI. CONCLUSION AND FUTURE SCOPE

The EMAC provide energy efficient approach for allocating jobs to virtual machines. However as resource decreases and requirement increases EMAC fails. Individual resource searching criteria is required under such situation. For this purpose BSA(Backtracking search algorithm) comes into picture which is hybridized with EMAC in proposed literature. Performance evaluation indicates significant improvement in results in terms of Makespan, Flowtime, energy consumption and latency.

The work can be further extended by using ant colony along with EMAC for enhancement in distinct parameters especially Makespan.

VII. REFERENCES

[1] *Commun. ACM*, vol. 51, no. 7, p. 9, 2010.

[2] Y. Sharma, B. Javadi, W. Si, and D. Sun, “Reliability and energy efficiency in cloud computing systems: Survey and taxonomy,” *J. Netw. Comput. Appl.*, vol. 74, pp. 66–85, 2016.

[3] K. Hwang, X. Bai, Y. Shi, M. Li, W.-G. Chen, and Y. Wu, “Cloud Performance Modeling with Benchmark Evaluation of Elastic Scaling Strategies,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 130–143, Jan. 2016.

[4] M. Wang and W. Zeng, “A comparison of four popular heuristics for task scheduling problem S. P. Mirashe and N. V Kalyankar, “Cloud Computing,”in computational grid,” *2010 6th Int. Conf. Wirel. Commun. Netw. Mob. Comput. WiCOM 2010*, pp. 3–6, 2010.

[5] C. R. Reeves, “A genetic algorithm for flowshop sequencing,” *Comput. Oper. Res.*, vol. 22, no. 1, pp. 5–13, Jan. 1995.

[6] Z. Zhu, G. Zhang, M. Li, and X. Liu, “Evolutionary Multi-Objective Workflow Scheduling in Cloud,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 5, pp. 1344–1357, May 2016.

[7] E. A. Siy, “A Sequencing Heuristic to Minimize Weighted Flowtime in the Open Shop,” vol. 4, 2016.

[8] J. Kołodziej and F. Xhafa, “A game-theoretic and hybrid genetic meta-heuristics model for security-assured scheduling of independent jobs in computational grids,” *CISIS 2010 - 4th Int. Conf. Complex, Intell. Softw. Intensive Syst.*, pp. 93–100, 2010.

[9] M. M. Alobaedy and K. R. Ku-Mahamud, “Scheduling jobs in computational grid using hybrid ACS and GA approach,” *Proc. - 2014 IEEE Comput. Commun. IT Appl. Conf. ComComAp 2014*, pp. 223–228, 2014.

[10] Z. Zhou, X. Xiang, and X. Wang, “An Energy-Efficient Data-Dissemination Protocol inWireless Sensor Networks,” in *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM’06)*, 2006, pp. 13–22.

[11] I. Zinno *et al.*, “Cloud Computing for Earth Surface Deformation Analysis via Spaceborne Radar Imaging : a Case Study,” vol. 7161, no. April 2014, pp. 1–35, 2015.

[12] Y. Xie, H. Wen, B. Wu, Y. Jiang, and J. Meng, “Transactions on Cloud Computing,” vol. 13, no. 9, 2015.

[13] L. A. Tawalbeh, “Mobile Cloud Computing Model and Big Data Analysis for Healthcare,” vol. 3536, no. c, 2016.

[14] Z. Xiao, W. Song, and Q. Chen, “Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1107–1117, Jun. 2013.

[15] J. Emeras, S. Varrette, V. Plugaru, and P. Bouvry, “Amazon Elastic Compute Cloud (EC2) vs . in-House HPC Platform : a Cost Analysis,” vol. 7161, no. c, pp. 1–14, 2016.

[16] R. Deng, R. Lu, S. Member, and C. Lai, “Optimal Workload Allocation in Fog-Cloud Computing Towards Balanced Delay and Power Consumption,” vol. X, no. X, pp. 1–11, 2016.

[17] T. Scholl, R. Kuntschke, A. Reiser, and A. Kemper, “Community Training: Partitioning Schemes in Good Shape for Federated Data Grids,” in *Third IEEE International Conference on e-Science and Grid Computing (e-Science 2007)*, 2007, pp. 195–203.

[18] S. Moradi, “A distributed method based on mobile agent to detect Sybil attacks in wireless sensor networks,” pp. 276–280, 2016.

[19] A. Beloglazov, J. Abawajy, and R. Buyya, “Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing,” *Futur. Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012.

[20] M. B. Qureshi *et al.*, “Survey on Grid Resource Allocation Mechanisms,” *J. Grid Comput.*, vol. 12, no. 2, pp. 399–441, 2014.

[21] B. P. Rimal, G. S. Member, and D. P. Van, “Mobile-Edge Computing vs . Centralized Cloud Computing over a Converged FiWi Access Network,” vol. 4537, no. c, 2017.

[22] S. D. Zahedani and G. Dastghaibyfard, “A hybrid batch job scheduling algorithm for grid environment,” *Proc. 4th Int. Conf. Comput. Knowl. Eng. ICCKE 2014*, pp. 763–768, 2014.

[23] D. Paul and S. K. Aggarwal, “Multi-objective evolution based dynamic job scheduler in grid,” *Proc. - 2014 8th Int. Conf. Complex, Intell. Softw. Intensive Syst. CISIS 2014*, pp. 359–366, 2014.

[24] S. C. Horng and S. S. Lin, “Integrating Ant Colony System and Ordinal Optimization for Solving Stochastic Job Shop Scheduling Problem,” *Proc. - Int. Conf. Intell. Syst. Model. Simulation, ISMS*, vol. 2015–Octob, pp. 70–75, 2015.

[25] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, “Mobile-Edge Computing for Vehicular Networks,” no. June, pp. 2–10, 2017.

[26] V. Hamscher, U. Schwiegelshohn, A. Streit, R. Yahyapour, R. Buyya, and M. Baker, “Evaluation of Job-Scheduling Strategies for Grid Computing,” *Grid Comput.*, vol. 1971, pp. 191–202, 2000.

[27] V. K. Sanap and V. M. Mane, “Region duplication forgery detection in digital images using 2D-DWT and SVD,” *Proc. 2015 Int. Conf. Appl. Theor. Comput. Commun. Technol. iCATccT 2015*, pp. 599–604, 2016.

[28] K.-L. Tsai *et al.*, “Cloud Encryption Using Distributed Environmental Keys,” *2016 10th Int. Conf. Innov. Mob. Internet Serv. Ubiquitous Comput.*, pp. 476–481, 2016.

[29] D. C. TRAN Dang Cong, WU Zhijian, WANG Zelin, “A Novel Hybrid Data Clustering Algorithm Based on Artificial Bee Colony, Algorithm and K-Means,” *Chinese J. Electron.*, vol. 24, no. 4, 2015.

[30] P. V. N. Reddy and A. R. N. System, “IMAGE ENCRYPTION AND DECRYPTION IN MODULI SET.”

[31] C. Fahn and H. Chen, “Detection System Based on Branchlet Features and.”

[32] X. Chen, “Decentralized Computation Offloading Game for Mobile Cloud Computing,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[33] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, and X. Peng, “Energy-efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks,” vol. 3536, no. c, pp. 1–10, 2016.

[34] D. Chen, “Research on traffic flow prediction in the big data environment based on the improved RBF neural network,” *IEEE Trans. Ind. Informatics*, vol. 3203, no. c, pp. 1–1, 2017.

[35] W. Xu, Y. Shen, N. Bergmann, and W. Hu, “Sensor-assisted Multi-view Face Recognition System on Smart Glass,” *IEEE Trans. Mob. Comput.*, vol. 1233, no. c, pp. 1–1, 2017.

[36] V. Biksham, “Query based computations on encrypted data through homomorphic encryption in cloud computing security,” pp. 3820–3825, 2016.

[37] L. Shakkeera and L. Tamilselvan, “Emerging Research in Computing, Information, Communication and Applications,” no. Mcc, 2016.

[38] C. Pahl and I. Centre, “Containerization and the PaaS Cloud,” 2015.

[39] L. Jourdan, M. Basseur, and E. G. Talbi, “Hybridizing exact methods and metaheuristics: A taxonomy,” *Eur. J. Oper. Res.*, vol. 199, no. 3, pp. 620–629, 2009.

[40] N. Rutar and J. K. Hollingsworth, “Software Analysis Techniques to Approximate Data Centric Direct Measurements.,” *First Int. Work. High-performance Infrastruct. Scalable Tools*, 2011.

[41] J. B. Kaper, H. Lockman, M. M. Baldini, and M. M. Levine, “A Recombinant Live Oral Cholera Vaccine,” *Bio/Technology*, vol. 2, no. 4, pp. 345–349, Apr. 1984.

[42] J. Xie, Y. Zhou, and H. Chen, “A novel bat algorithm based on differential operator and Levy flights trajectory,” *Comput. Intell. Neurosci.*, vol. 2013, 2013.

[43] L. Shakkeera and L. Tamilselvan, "Energy-Aware Application Scheduling and Consolidation in Mobile Cloud Computing with Load Balancing" Springer India 2016.