International Journal of Computational Intelligence Research
ISSN 0973-1873 Volume 13, Number 5 (2017), pp. 1283-1300

http://www.ripublication.com

Survey of Resource Provisioning Heuristics in Cloud and their Parameters

Kamali Gupta* and Vijay Katiyar**

**Department of Computer Science & Engineering, Maharishi Markandeshwar University, Ambala, Haryana, India.*
H.No 282, Sector 48 B, Kendriya Vihar society, Chandigarh-160047, India.

*** Department of Computer Science & Engineering, Maharishi Markandeshwar University, Mullana, India.*

Abstract

Deriving an optimal solution to demands lying in the vicinity of resource allocation mechanism is necessitated in Cloud computing. But, the underlying traditional methods find difficulty in obtaining solutions to NP-Hard problems, although, these methods are fast, deterministic and obtain exact solutions. With the aim to overcome such discrepancies, IT practitioners are focused towards exploring, developing and deploying heuristics and meta-heuristics techniques. This paper elucidates a wide range of study of existing resource management techniques and also explicates researches carried out in this area. A comparison among the batch-mode heuristics is presented in terms of makespan and the results have been verified using Cloudsim tool. Based on the analysis, future scope of the work has been proposed.

Keyword: Heuristics, Meta-Heuristics, Resource Management, Resource Scheduling, Makespan

1. INTRODUCTION (10 PT)

The resources [1] in a typical Cloud environment are pooled to serve multiple subscribers. Multi-tenancy models are being adopted by the providers to cater to the demands of tenants. Such assignments are provisioned via SLA agreements to outfit the multifaceted demands posed by users. The resources such as Disk space, CPU cores and network bandwidth must be sliced and shared among potential virtual machines for serving heterogeneous workloads. This distribution in Cloud can be termed as resource management. The definition in its scope is widely divided into three terms: - Resource Monitoring, Resource Allocation and Resource Discovery (Resource Provisioning). Therefore, resource management taxonomy is to be clearly stated to subsequently fulfill ongoing requirements. The taxonomy of resource management is illustrated in Figure 1.

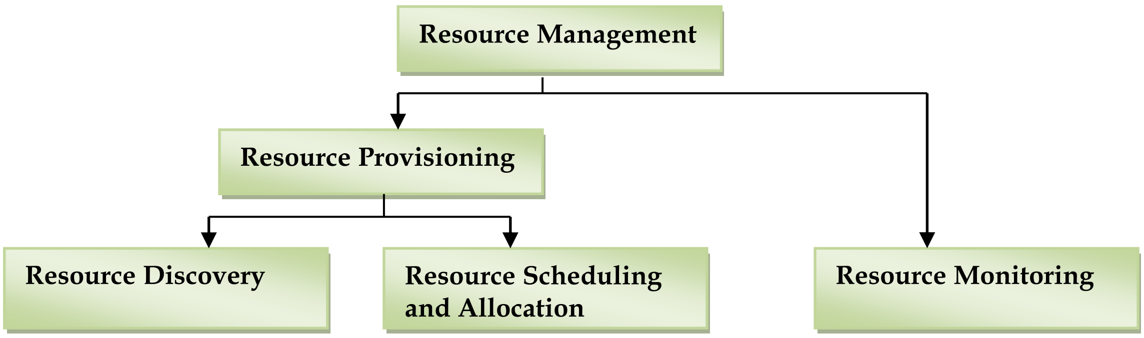

Figure 1. Taxonomy of Resource Management

The study discovered that the Resource Provisioning process enables provisioning of resources (vm instances) among the competing users. It comprises of two modules, namely, Resource Discovery and Resource Scheduling and Allocation.

The process of Resource Discovery, scheduling and allocation is carried out by resource broker or user broker in order to discover the available resources that best matches the application requirements. As mentioned in [2], resource discovery facilitates determination of state of resources that are incorporated and inter-operated in various resource management systems. The process identifies the physical hosts that are available to slice them into virtual machines that best suites the user requirements.

Resource Scheduling aims to identify the best resource from a set of available matched physical resources. After scheduling resource allocation is performed that aims to allocate the selected resource to the job. Resource allocation is a term that facilitates assignment of available virtual machine instances to the jobs submitted in a Cloud environment. The process in its broader terms is carried out by a broker who selects the best virtual machine (vm) instance against set of available vm instances for the purpose of meeting QoS requirements of each product operation. Resource allocation is followed by resource monitoring. Resource Monitoring [3] is a key tool that provision controlling and managing software and hardware infrastructures. It bestows and maintains key performance indicators that facilitate data collection to aid in decisions associated with resource allocation process. Besides, it monitors state of resources at the time of failure at physical or service layers. The phase is carried out to optimize the available resources.

This paper is focused on comprehensive study of existing scheduling mechanisms. A detailed comparison among existing policies is presented, pitfalls are revealed and improvisation methods are suggested in this direction. In section II, resource allocation strategies are exemplified. Section III epitomizes resource allocation models in Cloud paradigm. In section III existing heuristics have been explored and discussed.

2. ELEMENTS OF RESOURCE ALLOCATION STRATEGIES

From providers' perspective, predicting heterogeneous users and application demands and dynamic nature of users is impractical. A Cloud user aspires to obtain the bequeathed service request with in the prescribed time-frame with minimal cost [4]. Hence, efficient customized resource allocation systems that are capable of establishing a match between such perspectives are to be still established. The system should be capable of pacing with limitation of resources, heterogeneous user demands, geographical constraints of users and resources. For efficient resource allocation in a Cloud based environment, several strategies exist that have been explored and presented below:-

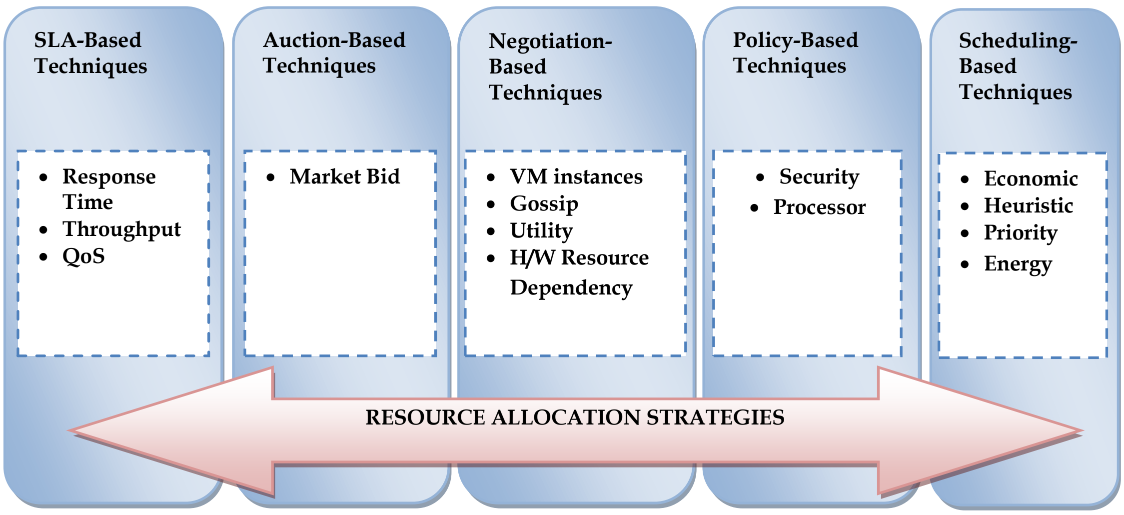

Figure 2. Elements of Resource Allocation Strategies along with their parameters

Figure 2 illustrates a number of resource allocation strategies explored in the research work. SLA Based Cloud scheduling focuses on achieving SLA requirements posed by user at the time of submission of service requests. These are the parameters/requirements of service asked by the user. The user-centric SLA parameters are- response time, deadline and Qos while provider centric SLA parameters are increased resource utilization (by enhancing up the throughput levels generated in a typical Cloud). Though a lot of progress is being made in offering satisfactory services to the consumers, still achieving SLA's are still in infancy stages for the providers. In [5], the author has addressed problem relating to profit oriented service request scheduling by focusing on both user centric and consumer centric parameters. The research work presented in [6] focuses on raising profit of SaaS providers by serving SLA driven user based QoS parameters. The researcher proposes optimization in resource allocation by reducing cost through mapping customer requests into infrastructure level parameters. The paper is directed towards calculating SLA violations in submitted service requests. The research work carried out in[7] signifies a resource management framework for SaaS providers in order to avoid SLA violations.

The study presented in above researches was oriented towards achieving SaaS providers benefits, reducing SLA violations and reducing resource wastage significantly. The scope of the present work carried out in this paper, apart from focusing on SLA violations and improving resource utilization, is oriented towards reducing makespan and meeting deadline requirements prescribed by the user.

Auction based Techniques in a Cloud environment is a market-based strategy that deals with finding the winners of auctions on the basis of their quoted bids and resource availability. Auction based resource allocation model is depicted in [8] and is presented as an improvement over CA-GREEDY (static provisioning method)algorithm by proposing a CA-PROVISION algorithm that dynamically provisions the resources. [9] presented combinatorial techniques for determination of winners in an auction-based Cloud market.[10] presented a CDA mechanism for determination of Cloud. The work further elaborated that the CDA mechanism is better than existing double auction mechanism as it continuously operates to sell unused resources at a reduced price.

Negotiation-based Techniques of resource allocation are broadly classified into four categories, namely vm instances, gossip, utility and hardware resource dependency. These techniques hand down the services by negotiating with the consumer on type of request generated by him.A more advancement to the theory stated above has been taken up by[11] in booting the resources at user end. This envisions payment by user for resources that has been used.Research work of [12] addressed that a number of algorithm exists for real-time task scheduling on multiprocessor systems, but still lacks in scalability.Different researchers have proposed different techniques for vm creation , placement and migration. Besides, the work presented illustrates allocation of resources based on VM cost and potential.

Gossip Based negotiations are agreed upon by collecting Peer information, Peer Resources and Experts advises.In [13], a gossip-based protocol is proposed and implemented that serves the user requesting time-dependent memory demands by dynamically maximizing a global utility function.Research work of[14] dealt with the issue of under-provisioning and over-provisioning of resources by adopting multiple strategies whenever a variation in user demands arrives.

Utility Based services for resource allocation focuses on response time, profit and application satisfaction. Many researchers [15-17] have conducted studies in this area to serve high priority applications.

The hardware dependent resources are allocated to demanding users on the basis of hardware requirements. One such study has been carried out in[18]. The research shows that support has been extended towards leasing of virtual and physical resources from a single point to offer support to heterogeneous applications on a shared infrastructure.

Policy-based Approaches are oriented towards the base of requests raised by the user. Some requests of users are security based while others are processor (computing and storage services)based. An adaptive resource allocation model has been presented in[19] that allocates the resources to users lying in the vicinity of the provider and is focused towards offering processor based services to the users. Research work of[20] comprises of security based services and has proposed loyalty-based trust mechanism that assesses the system according to its real-time condition. On the basis of it, the decision of resource allocation is done. Next section discusses various resource allocation models existing in cloud environment.

3. RESOURCE ALLOCATION MODELS IN CLOUD PARADIGM

The taxonomy of scheduling strategies is elucidated in Figure 3. Scheduling on Cloud platform can be pondered as an epitome of the research studies of Cloud; consequently, many studies have been attempted to give a precise and comprehensible definition of scheduling, still some variations exist .A lot of research is being conducted on heuristic and meta-heuristic techniques in Cloud to allocate resource in its fullest efficacy in order to cater to the heterogeneous demands of users present at diversified locations. Efforts are being made in this direction by scholars in grasping cognizance about the existing technologies.One such survey work conducted in the paper discusses meta-heurtic resource allocation techniques, namely, Particle Swarm Optimization, Ant Colony Optimization, Hill Climbing, Genetic Algorithm and Simulated Annealing[21]. The author outlined that the advantages of using meta-heuristic techniques in terms of computation time by finding approximate solutions to the problems is faster as compared to traditional exhaustive algorithms. Unfortunately, the techniques does not guarantee an optimal solution and takes longer time to execute. Author further suggested that two approaches that needs to be adopted in present and future scheduling is in direction of intensification(intensively searching a small region to find the possible best solution in that region) and diversification(implies searching a larger region better than local optima for obtaining a solution).

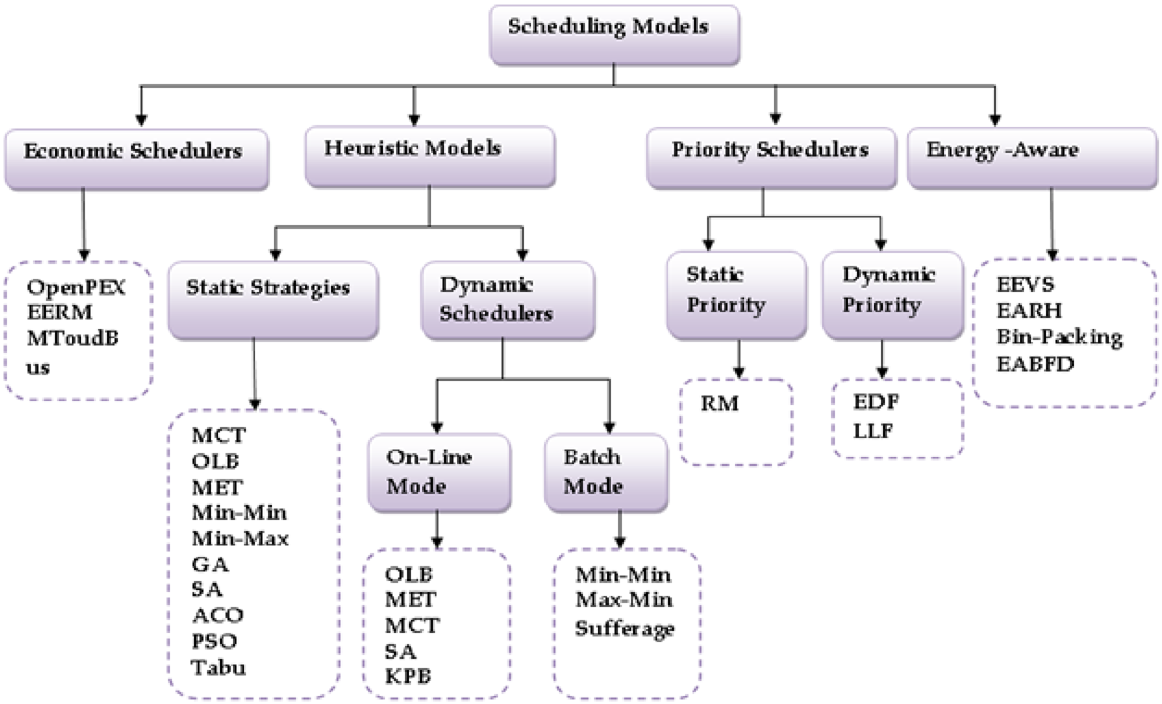

Figure 3. Taxonomy of Resource Scheduling

This section further discusses a number of existing scheduling techniques that have been explored and is presented in Table 1.

Table 1. Existing Scheduling Models Parametric Analysis

Sr. No	Algorithm/ Mechanism	Scheduling Strategy	Performance Metrics/ Parameters	Nature of Task	Simulation Environment / Simulator	Referenced Work
1.	Improved Min-Min Algorithm	Min-Min Heuristic	Resource Utilization, Reasonable Time , Cost	Independent	CloudSim	[22]
2.	X-Sufferage	Sufferage	Makespan, cluster level Sufferage value	Independent	CloudSim	[23]
3.	Task-Aware Scheduling Algorithm	Min-Min, Sufferage heuristic	Makespan, Number of Tasks	Independent	CloudSim	[24]
4.	Improved Max-Min	Max-Min Heuristic	Makespan	Independent	CloudSim	[25]
5.	Switcher Algorithm	Min-Min Heuristic,	Makespan, Load	Independent	CloudSim	[26]

		Max-Min Heuristic	balancing			
6.	Genetic Algorithm Coupled with Suffrage Heuristic	Sufferage Heuristic, Genetic Algorithm	Makespan	Independent	CloudSim	[27]
7.	Improved Sufferage Meta-Task scheduling Algorithm(M-Sufferage)	Sufferage Heuristic	Makespan, Resource Utilization, Minimum Completion time	Independent	Grid Environment	[28]
8.	Lottery Algorithm	Min-Min Heuristic	Makespan	Independent	Java-Based Simulation ToolKit	[29]
9.	Skewness-Based Min-Min Max-Min Heuristic	Hybrid Heuristic	Makespan	Independent	CloudSim	[30]
10.	-	FCFS Heuristic	Deadline, Priority of Jobs	Independent	CloudSim	[31]
11.	Enhanced Max-Min Task Scheduling algorithm	Max-Min Heuristic	Makespan, Load Balancing	Independent	CloudSim	[32]
12.	Community-Aware Scheduling Algorithm (CASA)	-	Average Job Waiting Time, Faster Response Time, Resource Utilization	Independent	MaGateSim	[33]
13.	Cloud-Aware Data Intensive Workflow Scheduling	-	Deadline, Cost	Workflow	Cycloid Grid	[34]
14.	Basic ACO	ACO	Makespan	Independant	CloudSim	[35]
15.	Basic ACO	ACO	Throughput, Response Time	VM Placement	Cloudsim	[36]
16.	Effective Scheduling Algorithm for Load Balancing	ACO	Load Balancing, SLA violation avoidance, Energy	Independent	CloudSim	[37]

	(SALB)		Consumption			
17.	Energy-Aware ACO based Workload Consolidation	ACO	Energy Conservation	VM Placement	Own Java-Based Simulation ToolKit	[38]
18.	-	ACO	Energy Conservation	VM Placement	CloudSimulation Environment	[39]
19.	-	ACO	Energy Conservation and Resource Utilization	VM Placement	CloudSimulation Environment	[40]
20.	Energy Efficient Mechanism	ACO	Energy Consumption, SLA, Throughput, Response Time	Independent	Proposed	[41]
21.	Parallel Genetic Algorithm	GA	Resource Utilization, Speed of resource allocation	Independent	CloudSimulation Environment	[42]
22.	modified aggregative genetic algorithm using the vaccination process(IGA)	GA	Makespan, Cost, Availability, Reliability, Energy Consumption	Worksflow	CloudSimulation Environment	[43]
23.	-	GA	Energy Conservation	VM Placement	Java	[44]
24.	Shadow Price Guided Genetic Algorithm (SGA)	GA	Energy Conservation	Independent	Microsoft C#	[45]
25.	Genetic Based Energy Aware Schedulers	GA	Makespan, Energy Conservation	Independent	HybridSim- G Grid Simulator	[46]
26.	Modified PSO Algorithm	PSO	Makespan	Independant	CloudSimulation Environment	[47]
27.	Integer PSO	PSO	Makespan, Cost	Independant	CloudSimulation Environment	[48]
28.	PSO Infrastructure level Cloud Scheduler	PSO	Throughput, Response Time	VM Placement	CloudSim	[49]
29.	Energy-Efficient	PSO	Energy Conservation	VM Placement	CloudSim	[50]

	Algorithm based on PSO					
30.	Bin Balancing Algorithm	Bin Packing	Energy Consumption, Task Deadline	VM Placement	CloudSim	[51]
31.	Deadline Constrained Heuristic based Genetic Algorithms	GA	Minimize execution Time, Deadline	Workflows	Synthetic workflows	[52]

Besides, the research work elaborated in [53] highlights a Banker's algorithm for resource allocation in cloud to prevent deadlock by putting restriction on number of users logging into the system. The research work presented in [54] explicates an integrated scheduling strategy that considers both the credibility of resources and user satisfaction as research parameters. Considering all the above heuristics and their parameters, next section discusses some of the existing batch-mode heuristics.

4. STUDY OF EXISTING HEURISTICS

This section outlines existing heuristics for resource allocation that operates in Batch-Mode. The batch-mode operation first collects the tasks requests of similar types and then allocates the asked resources. The existing batch-mode algorithms along with their constraints are enlightened here that becomes motivation for the current study. A list of notations used in the study is prescribed in Table 2.

Table 2. List of Key Notations

Symbol	Definition
MT	Meta-Task comprising of submitted tasks
VMT	Set of virtual machines in data center
T_i	Current task in Meta-Task
VM_j	Current vm instance in VMT
$CT_{i,j}$	Completion Time Matrix of Task T_i on VM_j
$ET_{i,j}$	Execution Time of Task T_i on VM_j
Rj	Ready Time of VM_j
MCT_i	Minimum Completion Time of Task T_i
Sec_MCT_i	Second Minimum Completion Time of Task T_i
T_u	Task with Minimum Completion Time
VM_v	Virtual Machine that takes minimum completion time

T_v	Task with Maximum Completion Time
SV_i	Sufferage value for task T_i
T_s	Task with maximum sufferag value

4.1. Min-Min Heuristic

Min-Min heuristic as elaborated in [4] initiates with the set MT [Meta-Task] comprising of all unassigned tasks and executes in two phases.

The first phase establishes minimum expected completion time (for each task in MT) on each machine. The second phase chooses the task boasting minimum expected completion time among the set of tasks in MT. Further, the chosen task is assigned to the matching resource (having minimum expected completion time). Finally, the assigned task is removed from MT and the process is replicated multiple times until each task in the MT is mapped. The mechanism is depicted in Figure 4.

Input: Meta-Task MT, Task Length Mi, Resource Speed MIPSj, Resources VMT, Execution Time Matrix ETi,j

Output: Mapped Schedule S:S(T1), S(T2),…..S(Tn)

Begin:

MT←{T_1,T_2,….T_n), VMT←{VM_1,VM_2,….VM_m)

While MT ≠ ϕ ***Do***

For $T_i \in$ MT ***Do***

For $VM_j \in$ VMT ***Do***

Compute $ET_{i,j} = M_i / MIPS_j$

End For

For $T_i \in$ MT ***Do***

For $VM_j \in$ VMT ***Do***

Compute $CT_{i,j}= ET_{i,j} + R_j$

End For

Find MCTi // Tu on VM_v, Sort($CT_{i,1}$,$CT_{i,2}$,…..$CT_{i,n}$)

Find S(T_u) for (T_u, VM_v)

Compute MT=MT- T_u //Delete task T_u from Meta-Task

R_u=$CT_{u,v}$ // Update ready time of machine v

For $T_i \in$ MT ***Do***

$CT_{i,v}$=$E_{i,v}$ + R_v

End For

End While

Fiure 4. Min-Min Heuristic

The Min- Min heuristic attempts to assign the tasks to the resources (virtual machines) that can execute them fastest. However, the heuristic fails to balance the load well as at the outset, scheduling of smaller tasks is done only. So, the Max-Min heuristic is discussed in the next section.

4.2. Max-Min Heuristic

Max-Min differs from Min-Min heuristic in second phase of its execution, where tasks(from MT) having overall maximum expected completion time is chosen and assigned to the virtual machine that takes minimum time to execute it. The algorithm is illustrated in Figure 5. In other words, Min-Min heuristic bestows precedence to tasks of shortest earliest completion time while Max-Min tends to schedule the longer task first at the time of each scheduling instance. However, the algorithm generates the schedule that is not much optimum in terms of makespan in comparison to the sufferage heuristic discussed in next section.

Input: Meta-Task MT, Task Length Mi, Resource Speed MIPSj, Resources VMT, Execution Time Matrix ETi,j

Output: Mapped Schedule S:S(T1), S(T2),…..S(Tn)

Begin:

MT←{T_1,T_2,….T_n), VMT←{VM_1,VM_2,….VM_m)

While MT ≠ ϕ ***Do***

For $T_i \in$ MT ***Do***

For $VM_j \in$ VMT ***Do***

Compute $ET_{i,j} = M_i / MIPS_j$

End For

For $T_i \in$ MT ***Do***

For $VM_j \in$ VMT ***Do***

Compute $CT_{i,j}= ET_{i,j} + R_j$

End For

Find MCTi // Tv on VM_v, //Sort($CT_{i,1}$,$CT_{i,2}$,…..$CT_{i,n}$) in descending

Find S(T_v) for (T_v, VM_v) //Assign task with maximum completion time on VM_V

Compute MT=MT- T_v //Delete task T_v from Meta-Task

R_v=$CT_{v,v}$ // Update ready time of machine v

For $T_i \in$ MT ***Do***

$CT_{i,v}=E_{i,v} + R_v$

End For

End While

Figure 5. Max-Min Heuristic

4.3. Sufferage Heuristic

The sufferage algorithm attains good reduction in makespan when evaluated against Min-Min and Max-Min. Sufferage heuristic [5] maps a virtual machine instance to the task that suffers most in terms of expected completion time. The mapping is performed according to the calculated sufferage value.

At the very outset, the completion time for each task of meta-task on each resource is computed. Subsequently, the two consecutive minimum completion times pertaining to each task is figured out followed by calculating the difference between these two generated values, also known as sufferage value. Later, the task with maximum sufferage value is mapped and assigned to the resource that takes minimum completion time among all machines for execution. Finally, the completion time for resources gets updated and in iteration, the above steps are performed until the entire tasks of Meta-Task get executed. The algorithm of Sufferage heuristic is depicted in Figure 6.

Input: Meta-Task MT, , Task Length Mi, Resource Speed MIPSj, Resources VMT, Execution Time Matrix ETi,j

Output: Mapped Schedule S:S(T1), S(T2),…..S(Tn)

Begin:

$MT \leftarrow \{T_1,T_2,\ldots.T_n)$, $VMT \leftarrow \{VM_1,VM_2,\ldots.VM_m)$

While $MT \neq \phi$ ***Do***

For $T_i \in MT$ ***Do***

For $VM_j \in VMT$ ***Do***

Compute $ET_{i,j} = M_i / MIPS_j$

End For

For $T_i \in MT$ ***Do***

For $VM_j \in VMT$ ***Do***

Compute $CT_{i,j}= ET_{i,j} + R_j$

End For

For $VM_j \in VMT$ ***Do***

Mark all machines as unassigned

End For

For $T_i \in MT$ ***Do***

Find MCT_i and Sec_MCT_i

Compute $SV_i = Sec\_MCT_i - MCT_i$

Compute VM_v

If VM_v is unassigned ***Then//*** $VM_v \in VMT$

Assign T_s on VM_v , mark it as assigned

Else If $SV_i(T_i)$ on $VM_v < SV_k(T_k)$ ***Then //*** $T_k \in MT$

Unassign T_i on VM_v

Find $S(T_k) : (T_k, VM_v)$

Compute $MT=MT- T_k$ //Delete task T_k from Meta-Task

EndIF

$R_v=CT_{k,v}$ // Update ready time of machine v

For $T_i \in MT$ ***Do***

$CT_{i,v}=E_{i,v}+ R_v$

End For

End While

Figure 6. Sufferage Heuristic

5. RESULTS AND ANALYSIS

A comparison among the algorithms in terms of makespan has been done on cloudsim 3.0.3 framework. The simulations have been conducted on testbed consisting of 100 CloudLets and 10 virtual machines in a DataCenter. The generated results are specified in Figure 7.

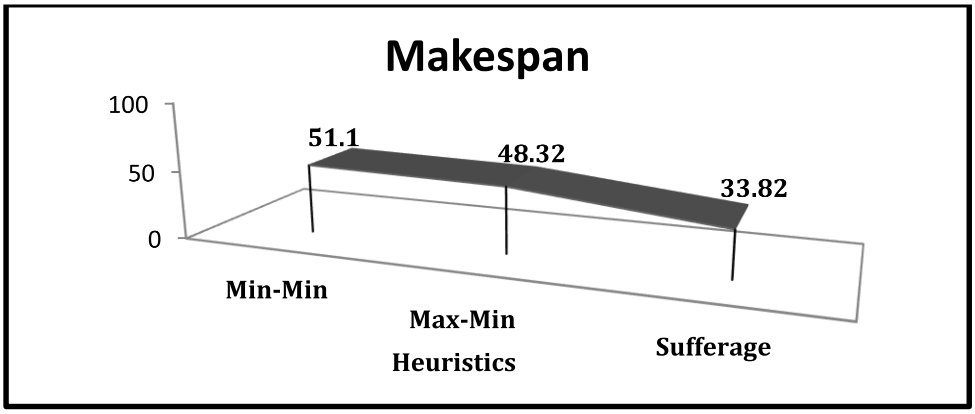

Figure 7. Comparison in makespan

It has been observed that in comparison to other batch-mode heuristics, sufferage heuristic is able to reduce the makespan value to a good extent.

6. CONCLUSION

A wide study of existing heauristics in the domain of cloud resource provisioning is presented in this research study. Keeping in consideration the above findings, it was observed that a heuristic based scheduling strategy can be derived that can outfit a good solution to the existing Cloud challenges in terms of constraints of deadline, makespan , cost and SLA violation avoidance as uptil now only one or two parameter have been considered as a part of research findings. Besides, the research work carried out up till now was providers-centric as it offers scheduling organizations to optimize providers goal, so a new strategy can be devised that is focused towards achieving users goal. Moreover, as sufferage heuristic is better in performance in terms of makespan, so the algorithm can be further improvised to integrate deadline and SLA parameters.

REFERENCES

[1] M.H. Mohamaddiah, A. Abdullah, S. Subramaniam, M. Hussin, "A Survey on Resource Allocation and Monitoring in Cloud Computing", *International Journal of Machine Learning and Comp*. pp. 31-38, 2014.doi: 10.7763/IJMLC.2014.V4.382

[2] K. Krauter, R. Buyya, M. Maheswaran M., "A taxonomy and survey of grid resource management systems for distributed computing", Software: Practice and Experience, 32(2), pp. 135-164, 2002.

[3] G. Aceto, A. Botta, W.D. Donato, A. Pescape, "Cloud monitoring:A survey", Computer Networks, Vol. 57, pp. 2093-2115, 2013.

[4] B. Rajasekar, S.K. Manigandan,"An Efficient Resource Allocation Strategies in Cloud Computing", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, pp.1239-1244, 2015.doi: 10.15680/ijircce.2015.0302083

[5] Y.C. Lee,"Project driven service request scheduling in Clouds", Proceedings of international symposium on Cluster and Grid computing, CC. Grid, Melbourne, Australia, 2010.

[6] L. Wu, S.K. Garg, R.K. Buyya, "SLA based resource allocation for Software as a Service Provider(SaaS) in Cloud Computing Environments", pp. 195-204,IEEE, 2011.

[7] T.B. Richard, T.B. Ma, D.M. Chiu, John, C.S. Liu, V. Misra, D. Rubenstein, "Resource Management for Cloud Users: a generalized Kelly Mechanism Approach", Electrical Engineering, 2010.

[8] S. Zaman, "A Combinatorial Auction-Based Mechanism for Dynamic VM Provisioning and Allocation in Clouds", IEEE transactions on Cloud computing, 2012.

[9] T. Sandholm, "Algorithm for optimal winner determination in combinatorial auctions", Artificial Intelligence 135 , pp. 1–54, Elsevier, 2002.doi: S0004-3702(01)00159-X

[10] S. Xuelin, K. Xu, "Continuous Double Auction Mechanism and Bidding Strategies in Cloud Computing Markets", IEEE Transactions, 2013.

[11] A. Inomata, T. Morikawa, M. Ikebe, S.M. Rahman, "Proposal & Evaluation of Dynamic Resource Allocation Method Based on Load of VMs on IaaS", IEEE, 2010.

[12] K. Kumar, J. Feng, Y. Nimmagadda, Y.J. Lu, "Resource allocation for real-time tasks using Cloud computing", IEEE, 2011.

[13] F. Uhib, R. Stadler, "Distributed monitoring and resource management for large Cloud environments", IEEE, 2011.

[14] P. Marshall, K. Keahey, T. Freeman, "Elastic Site", IEEE, pp. 42-46, 2010.

[15] D. Minarolli, B. Freisleben, "Utility based Resource allocations for virtual machines vin Cloud computing", IEEE, 2011.

[16] J. Akhani, S. Chaudhary, G. Somani, "Negotiation for Resource Allocation in IaaS Cloud", Bangalore Compute Conf., ACM, 2011.

[17] H.K. Mehta, E. Gupta, "Economy Based resource allocation in IaaS Cloud", International Journal of Cloud application Computing, 2013.

[18] L. Xiaoyi, J. Lin, L. Zha, Z. Xu, "Vega Ling Cloud: A resource single leasing point system to support heterogeneous application modes on shared infrastructure", IEEE, 2011.

[19] G. Jung, K.M. Sim, "Agent-Based adaptive Resource allocation on the Cloud Computing Environment", Parallel Processing workshops(ICPPW), 40^{th} International Conference, pp. 13-16, 2011.

[20] Y. Liu, S. Yang, Q. Lin, G.B. Kim, "Loyalty-Based Resource Allocation mechanism in Cloud Computing", Recent Advances in CSIE, Springer, 2012.

[21] C.W. Tsai, J. Rodrigues, "Metaheuristic Scheduling for Cloud: A Survey", IEEE Systems Journal, 2014.

[22] L. Gang, L. Jing, X. Jianchao, "An Improved Min-Min Algorithm in Cloud Computing, Proceedings of the 2012 International Conference of Modern Computer Science and Applications, Vol. 191, pp. 7-52, 2012.

[23] A.J. Ruby, B.W. Aisha, Shriram, "A Taxonomy and Survey of Scheduling Algorithms in Cloud: based on Task Dependency", *International Journal of Computer Applications*, ISSN 0975 – 8887, vol. 82(15), 2013.

[24] S.T. Dehkordi, V.K. Bardsiri, "TASA: A New Task Scheduling Algorithm in Cloud Computing", *Journal of Advances in Computer Engineering and Technology*, Vol. 1(4), 2015.

[25] S. Devipriya, C. Ramesh, "Improved Max-min heuristic model for task scheduling in Cloud", Green Computing, Communication and Conservation of Energy, IEEE Explore, 2014.

[26] M.M. Shoukat, M. Maheswaran, S. Ali, H.J.Siegel, D. Hensgen, R.F. Freund, "Dynamic mapping of a class of independent tasks onto heterogeneous computing systems", *Journal of Parallel and Distributed Computing*, Vol. 59, pp. 107–131, 1999.

[27] T. Goyal, A. Agrawal, "Host scheduling algorithm using genetic algorithm in Cloud computing environment", *International Journal of Research in Engineering & Technology*, Vol. 1(1), pp. 7-12, 2013.

[28] M. R Naglaa, "An Improved Sufferage Meta-Task Scheduling Algorithm in Grid Computing Systems", Research Gate, *International Journal of Advanced Research*, Vol. 3(10), pp. 123 -12, 20159.

[29] D. Elahe, A. Reza, "An Improved Approach for Scheduling Tasks Based on Lottery Algorithm in Cloud Computing Environment", Research Gate, *Global Journal on Technology*, Vol. 8, pp. 291-301, 2015.

[30] S.K. Panda, P. Agrawal, D. P. Mohapatra, "Skewness-Based Min-Min Max-Min Heuristic for Grid Task Scheduling", Proceedings of the 2014 Fourth International Conference on Advanced Computing & Communication Technologies, 2014.

[31] Saraswathi, Y.R. Kalaashri, S. Padmavathi, "Dynamic Resource Allocation Scheme in Cloud Computing", ELSEVIER, Procedia Computer Science 47, pp. 30-36, 2015.

[32] U. Bhoi, P.N. Ramanuj, "Enhanced Max-Min Task Scheduling Algorithm in Cloud Computing", *International Journal of Application or Innovation in Engineering & Management*, ISSN 2319-4847, Vol. 2(4), 2013.

[33] Y. Huang, N. Bessis, P. http://derby.openrepository.com/derby/simple-search?location=/&query=&filter_field_0=orcidId&filter_type_0=contains&filter_value_0=0000-0002-6013-3935&filter_field_1=abstract&filter_type_1=contains&filter_value_1=&submit_search=UpdateNorrington, P. Kuonen, "Hirsbrunner, Beat;Exploring decentralized dynamic scheduling for grids and Clouds using the community-aware scheduling algorithm", Future Generation Computer Systems, 2013.

[34] T. Ghafariana, J. Bahman, "Cloud-aware data intensive workflow scheduling on volunteer computing systems", Elsevier, Future Generation Computer Systems, Vol. 51, pp. 87–97, 2015.mailto:b.javadi@uws.edu.au

[35] M. Tawfeek, A. El-Sisi, A. Keshk, F. Torkey, "Cloud task scheduling based on ant colony optimization", 8th international conference compute eng system, 2013.

[36] E. Pacini, C. Mateos, C. Garcıa, "Balancing throughput and response time in online scientific Clouds via ant colony optimization", Elsevier, Adv Eng Software, Vol. 84, pp. 31–47, 2015.

[37] S. Khan, N. Sharma, "Effective scheduling algorithm for loadbalancing (SALB) using ant colony optimization in cloud computing", *International Journal of Advance Research Computer Science Software Engineering*, Vol. 4, 2014.

[38] E. Feller, L. Rilling, C. Morin, "Energy-aware ant colony based workload placement in Clouds", Proc 12th IEEE/ACM int conf grid computing. pp. 26–33, 2011. http://dx.doi.org/10.1109/Grid.2011.13.

[39] X. Liu, Z. Zhan, K. Du, W. Chen, "Energy aware virtual machine placement scheduling in Cloud computing based on ant colony optimization", Proc conference genetic evolution comput, ACM, pp. 41–7, 2014. http://dx.doi.org/10.1145/2576768.2598265.

[40] M.H. Ferdaus, M. Murshed, R.N. Calheiros, R.K. Buyya, "Virtual machine consolidation in Cloud data centers using ACO meta-heuristic", Euro-Par 2014 parallel process, Springer, pp. 306–17, 2014. http://dx.doi.org/10.1007/978-3-319-09873-9.

[41] L. Chimakurthi, S.K. Madhu, "Power efficient resource allocation for Clouds using ant colony framework", 2011, Available from arXiv:11022608.

[42] Z. Zheng, R. Wang, H. Zhong, X. Zhang, "An approach for Cloud resource scheduling based on parallel genetic algorithm", 3rd IEEE international conference on compute res dev, IEEE, 2011, http://dx.doi.org/10.1109/ICCRD.2011.5764170.

[43] K. Sellami, M. Ahmed-Nacer, P.F. Tiako, R. Chelouah, "Immune genetic algorithm for scheduling service workflows with QoS constraints in Cloud computing", *South African Journal Ind Eng*, Vol. 24, pp. 68–82, 2013.

[44] G. Wu, T. Maolin, Y.C. Tian, W. Li, "Energy-efficient virtual machine placement in data centers by genetic algorithm",Vmslv A, editor. Neural inf process,Springer, 2012.

[45] G. Shen, Y.Q. Zhang, "A shadow price guided genetic algorithm for energy aware task scheduling on Cloud computers", Adv Swarm Intell Notes Computer Science, Vol. 6728, Springer, 2011. http://dx.doi.org/10.1007/978-3-642-21515-5_62.

[46] J. Kołodziej, S.U. Khan, F. Xhafa, "Genetic algorithms for energy-aware scheduling in computational grids", Proc – international conference on P2P,

parallel, grid, Cloud internet computing, pp. 17–24, 2011. http://dx.doi.org/10.1109/3PGCIC.2011.13.

[47] S. Abdi, S.A. Motamedi, S. Sharifian, "Task scheduling using modified PSO algorithm in Cloud computing environment", International Conference on Machine Learning in Electrical & Mechanical Eng, pp. 37–4, 2014.

[48] A.S.A Beegom, M.S. Rajasree, "A particle swarm optimization based pareto optimal task scheduling in Cloud computing", Adv swarm intell notes computer science, Springer, pp. 79–86, 2014.

[49] E. Pacini, C. Mateos, C.G. Garino, "Dynamic scheduling based on particle swarm optimization for Cloud-based scientific experiments", *CLEI Electron Journal*, Vol. 14, pp. 1–12, 2014.

[50] A. Xiong, C. Xu, "Energy efficient multiresource allocation of virtual machine based on PSO in Cloud data center. Math Probl, 2014.

[51] J.M. Tang, L. Luo, K.M. Wei, X. Guo, X.Y. Ji, "A Heuristic Resource Scheduling Algorithm of Cloud Computing Based on Polygons Correlation Calculation", IEEE, pp. 365-370, 2015.

[52] A. Verma, "Deadline constraint heuristic-based genetic algorithm for workflow scheduling in Cloud", *International Journal of Grid and Utility Computing*, Vol. 5(2), 2014.

[53] S. K. Sood, "Dynamic Resource Provisioning in Cloud based on Queuing Model", *International Journal of Cloud Computing and Services Science,* vol. 2(4), pp. 313-320, 2013.

[54] W. Jiang, J. Zhang, J. Li, H. Hu, "A Resource Scheduling Strategy in Cloud Computing Based on Multiagent Genetic Algorithm", TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol. 11(11), pp. 6563-6569, 2013.