Australian Journal of Basic and Applied Sciences, 9(27) August 2015, Pages: 416-424

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

3DHBPTG: 3D Hybrid Bin Packing with task Grouping Based Cloud Schedulingfor Improving Availability

[1]S.K. Senthil Kumar and [2]P. Balasubramnie

[1]*Assistant Professor, Department of Computer Science and Engineering, Jansons Institute of Technology, Coimbatore, Tamil Nadu, India*
[2]*Professor, Department of Computer Science and Engineering, Kongu Engineering College, Erode, Tamil Nadu, India.*

ARTICLE INFO

Article history:
Received 23 June 2015
Accepted 25 August 2015
Available online 2 September 2015

Keywords:
Cloud Computing, Cloud Reliability and Availability, Cloud Scheduling with Task Grouping, Bin packing, 3 Dimension Bin Packing, Healthcare – as – a Service (HaaS), Task Grouping and Cloud Scheduling.

ABSTRACT

Cloud Computing is the present technology which sets the milestone for Information and Communication Technology (ICT). Several researchers developed several methods to produce the reliability of cloud services and Availability of Cloud Resources. Now Cloud computing plays a vital role in all business, Education and other applications. When we try to apply the Cloud computing for Healthcare application we find number of challenges. Out of which the Reliability and Availability are seems to be important challenge. To improve and maintain the Reliability of cloud services and Availability of cloud resources, we propose a new modified 3 Dimensional Hybrid Bin Packing Algorithms with Task Grouping (3DHBPTG) for maximize the utilization of cloud Resources by schedule and allocate the multiple lite tasks in one VM (Resource). Wen multiple tasks are allocated in one VM if possible then the resource availability can be improved. We compare our proposed algorithm with available algorithm and find the better results.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: S.K. Senthil Kumar and P. Balasubramnie, 3DHBPTG: 3D Hybrid Bin Packing with task Grouping Based Cloud Schedulingfor Improving Availability. ***Aust. J. Basic & Appl. Sci.,*** *9(27): 416-424, 2015*

INTRODUCTION

Cloud computing is present technology of computing that promises to provide more flexibility, less expense, and more efficiency in IT services to end users. It offers potential opportunities for improving nearly all the businesses, Education services, online digital libraries, real world problem solving and research. Cloud computing can be perceived as an on-demand, self-service Internet infrastructure that makes the user to utilize computing resources anytime from anywhere (Mell, 2011). It is a new model of delivering computing resources, such as CPU, Memory and other hardware, platform, data and software as services represented Everything-as-a Service (EaaS). With these application success stories, the cloud also can be applied on health care applications (Kagadis, 2013; Ahuja, 2012; Vilaplana, 2013) Healthcare Education, and for data storage for public people's Health and Medical Records. We are very much convinced in proposing the full-fledged application of cloud computing based Healthcare called HOSPTIONE a concrete utility service as Healthcare – as – a Service (HaaS). By using available advantages of cloud such as massive scalable computing, resources available on demand, elimination of a setup cost commitment by users, and payment for use on a short-term basis as needed (Ahuja, 2012). (Mell, 2011) By analyzing several research and application on evolutionary algorithms (Binitha, 2012; Bitam, 2012; Kaur, 2010; Wedde, 2014; Zhang, 2014), we understand that the evolutionary algorithms are static in nature and in any form of those algorithms are more iterative and may not be fit for dynamic system like cloud computing. With this conclusion, we follow only simple linear non-iterative algorithm for better and efficient scheduling to achieve availability of cloud services and reliability of cloud services. Bin packing problem (https://en.wikipedia.org/wiki/Bin_packing_problem.) is a classical combinatorial optimization problem which has been studied extensively (Coffman Jr, 1996). In the classical bin packing problem, given a set of items with different volumes, the objective is to pack the items into a minimum number of bins such that the total size of the items in each bin does not exceed the bin capacity C. There are many variations of this problem, such as 2D (Berkey, 1987), 3d (Martello, 2000) and multiple dimensions packing, linear packing, packing by weight, packing by cost, and so on. They have many applications, such as filling up containers, loading trucks with

Corresponding Author: S.K. Senthil Kumar, Assistant Professor, Department of Computer Science and Engineering, Jansons Institute of Technology, Coimbatore, Tamil Nadu, India
E-mail: sksenthilkumar@jit.ac.in

weight capacity constraints, creating file backups in media and technology mapping in Field-programmable gate array semiconductor chip design.

Related work:

There are several research were done on cloud scheduling for various objectives. We represent some of them here. The author (Yusen, 2014) proposed a resource scheduling algorithm called as MinTotal DBP for gaining the minimum cost for cloud utilization in Gaming cloud. Their Main Objective is to maintain the Minimum total cost for using the Cloud for gaming. The author (Bein, Doina) proposed an algorithm for online cloud storage by their clients' request to store in the cloud servers' hard dive. As every hard drives has the same size, this algorithm assumed that each hard drive as a bin and the data to be stored in this drive as item so they followed the classical Bin Packing algorithm to achieve the minimum storage. The author (Feller, 2011) proposed a multi dimension based bin packing model with Ant Colony based optimization for workload consolidation problem for energy gain in the cloud. Author (Javanmardi, 2014) proposed a hybrid job scheduling approach based on the aid of Genetic Algorithm (GA) with Fuzzy Algorithm with consideration of the load balancing of system. They proposed the above system with the objective in mind to assign the jobs to the resources by considering the VM MIPS and length of the job.The author (Zitzler) proposed a Multi-Objective Genetic cloud resource scheduling Algorithm (MO-GA) to minimize energy consumption and maximize the profit of service provides under the constraints of deadlines. Some researchers accomplished research on (Sandholm, 2015; Mathew, 2014) batch scheduling algorithm using FCFS first come and first service based scheduling algorithm was proposed to utilization with and without backfilling model. The authors (Liu, 2015) proposed a Gang scheduling is an alternative to batch scheduling. It is a scheduling algorithm for parallel systems that schedules related processes or threads to run simultaneously on different processors. Gang scheduling is based on ouster out matrix. Gang scheduling is useful for applications where performance of application degrades when any part of the application is not running. The author (Zhang, 2003) proposed a Backfilling and Migration Gang Scheduling for better utilization of resource and to improve the cloud's performance. Authors in (Singh, 2013), studied several job scheduling algorithm and compare them and concluded as the good cloud job scheduling algorithm is supposed to schedule the resources in optimized use of the resource. The Author (Verma, 2014) proposed a deadline constrained heuristic based Genetic Algorithm (HGA) to schedule the resources to tasks with bottom-level (B-Level) and Top-Level (T-Level) priority. Author (Song, 2014) proposed a Bin Packing based cloud resource scheduling called VISBP variable item Size Bin Packing algorithm with one, two and multiple Dimensional Bin Packing and evaluated.

In cloud, the requested tasks may classified as lite task when the task require very less memory to accommodate its data and instructions and very less CPU requirement to complete the services. A task may be classified as data intensive task when it requires the more memory to store data and very less CPU utilization. On other hand the task may be called processor intensive, when the task require less memory to store and high CPU requirement to complete the service. Finally, a task may be complex, when it require more memory to hold the data and instruction and more CPU requirement to complete the service. Same as a may be atom task, if the task cannot be divided. Some task can be called as molecule, if we can split it into two or more sub tasks and execute.

Bin packing problem:

Bin Packing Problem (BPP) is very long studied problem. The applications of BPP have numerous success stories in solving variety of research and real world problems. The Main objective of the BPP is to pack any objects with identical or different volumes in to the fixed or variable size least number of bins (boxes or Packets) by satisfying given constraints.

The mathematical formulation of BPP (https://en.wikipedia.org/wiki/Bin_packing_problem) is, in a given bin S, the size of V and list of n objects with sizes a_1, a_2, a_3 … a_n to pack, find an integer number of bins B and a B-partition S_1 U S_2 U .. S_B of the set {1,2,…,n} such that$\sum_{i\in Sk} ai \leq V$ for all k = 1, …, B.

A solution is optimal if it has minimal B. The B-value for an optimal solution is denoted OPT below. The possible integer linear programming of problem is:

Minimize B= $\sum_{i=1}^{n} yi$ subject to $\sum_{j=1}^{n} aj\ xij \leq V\ yi, \forall i \in \{1,2,..,n\}$, $\sum_{j=1}^{n} xij = 1\ \forall i \in \{1,2,..,n\}$, $y_i \in \{0,1\}\ \forall i \in \{1,2,..,n\}$, $x_{ij} \in \{0,1\}\ \forall i \in \{1,2,..,n\}$ where yi = 1 if bin i is used and xij = 1if then j is put into bin i.

To achieve the objective of the BPP there are several algorithms were studied (Senthil Kumar, 2012). The BPP can be broadly divided as fixed size Bin and Variable Size Bin. In fixed size Bin, the bin size if fixed for the entire bin. With that constraint, we try to pack all the objects in least number of bins. In case of variable size BPP, the each Bin has different size; accordingly we need to pack the objects. The other main classification of the BPP (Johnson, 1973) is i) Online BPP and ii) Offline BPP. In online BPP, the objects are packs as the input sequence without knowing the next objects' properties (size) which may not give the optimum packing. In the Offline BPP, the objects are received and with the knowledge of all objects are packed. So

there is a possibility of packing in the optimized manner. For the offline BPP there are different fitting algorithms (Berkey, 1987) were studied by several researchers such as First-Fit, First-Fit increasing order, First-Fit Decreasing order, Best-Fit, Next-Fit and Worst-fit algorithms. In First-Fit, the objects are packed without changing order. The first object is packed in first Bin and then we try to pack the second object with the same bin. If not possible we open another bin to pack the second one. Third object may be packed in any of the two bins if possible, if not third bin may be opened. Likewise the bins are used to pack all the objects. The main problem is that there are lots of fragment spaces in one or more bins at last. In the first-fit increasing order, the objects are ordered in ascending order of given sizes of each objects and then try to pack multiple objects in least number of bins. For the First-Fit decreasing, same as then previous model, the objects are ordered in descending order and try to pack them in the least bins. For the above two algorithm the fragments spaces may be reduced but cannot be fully eliminated. Best-Fit algorithm provides the best optimum packing among all the fitting algorithms. Here the each objects packed in the bin in such a way that no or very less fragment. Worst fit need not be described because this will not give the packing as we expect.

Proposed system:

We propose a new modified Multi-pass 3D Hybrid Bin Packing with Task Grouping(3BHBPTG) algorithm to achieve reliability incloud service and availability of cloud resources.In our proposed system we make use of hybrid BPP. We mean our proposed system follow a partial offline model to achieve the objective. As we know the cloud is a dynamic system we cannot make use of offline BPP. The cloud will have continues task request. If we try to follow the online BPP, we may not get the optimum usage of VMs (maximum utilization). So we follow a time slice model called FQTS (Flexible Quantum Time Slice) (Senthil Kumar, 2012). In this model, we receive tasks in FQTS time and wait for 20% of FQTS to get any other task arrived to cloud and waiting of 20% time until no task received or 3 times reached. Once the FQTS + waiting routine of time reached, the tasks are taken further processing. We also follow variable size bin (VM).Since, we have different configuration of VMs in the resource pool to do service such as Memory size and CPU cores with clock speed. To achieve the best optimum task scheduling and allocation, we make use of several historical Data (Senthil Kumar, 2012) as listed in terminologies. This algorithm is a customized from classical Bin Packing algorithm, to schedule the resources (Virtual Machine) to the requested tasks by clients to achieve the high availability of resources for fourth-coming tasks accomplishing by improving the reliability and minimize the number of Virtual Machine by maximizing the to achieve the maximum utilization of the resources. Using our proposed algorithm we also try to minimize the cost of the cloud services also.

In our proposed algorithm, each Virtual machine is assumed as a bin (hereafter bin and VM refers the same) and requested tasks are assumed as items (hereafter task and object refers the same) that will be packed in the VM. In our algorithm, we follow neither static nor dynamic. Instead, we follow a new hybrid style of scheduling system, in which the task receiver will wait for a time slice called FQTS (Flexible Quantum Time Slice) (Senthil Kumar, 2012) to achieve our objective. We define the terminologies and notations used in this paper in table 1.

Table 1: Terminologies used in this paper.

TERMINOLOGIES FOR 3 DIMENSION HYBRID BIN PACKING WITH TASK GROUPING (3DHBPTG)		
Task		
ReqT	:	All Requested Tasks by cloud.
TP	:	Task Pool.
TT	:	Total Tasks – where, n Tasks selected to get service for this iteration in TP. Referred as $T_1,T_2,T_3\ldots\ldots T_n$, where n denotes the total Tasks in TP.
T_i	:	a Particular Task in TP.
TGj	:	Task Group for j^{th} Vm, where one or more Tasks from TP whose RT = VMs RT and sum of CPU% and MEM required <= MCPUj and MMEMj appropriately.
Task Attributes		
DLT_i	:	Deadline time of i^{th} Task set by user. (Values either Immediate or any valid Time (greater than present time).
PTi	:	Priority of i^{th} Task set by user.
RT_i	:	Resource Type required for Task i.
CPU_i	:	CPU % required to get service for Task i.
MEM_i	:	Memory required to get service for Task i.
$Size_i$	:	Size of Task i (Expected Execution Time to complete).
AT_i	:	Arrival Time of i^{th} Task.
DT_i	:	Departure Time of i^{th} Task.
WT_i	:	Waiting time of i^{th} Task in Queue.
$RelT_i$	:	Release Time of i^{th} task from j^{th} VM.
ST_i	:	Service Time of i^{th} Task in j^{th} VM.
SS_i	:	Success Score of i^{th} Task (% of successful completion in S_i)
TOTTi	:	Total Time Taken to get Service of ith Task

TurnT	:	Turn Around Time of Assigned Tasks
Resource / VM		
AVM	:	All VMs – all resources in cloud.
RP	:	Resource Pool.
TVM	:	Total VM - m VMs (Resources) free and readily available to do service in RP. VM_1, VM_2, $VM_3 \ldots VM_m$, where m denotes the total VMs in RP.
VM_j	:	a Particular VM (resource) in RP.
Resource /VM Attributes		
VMTj	:	VM Type of j^{th} VM (when $RT_i = VMT_j$ then $VM_j \leftarrow T_i$).
MCPUj	:	Maximum CPU (Number of Cores) of j^{th} VM.
MMEMj	:	Possible Maximum Memory allocation for Tasks of j^{th} VM.
User		
UP_i	:	User Privilege of i^{th} Task.

The mathematical formulation of our proposed algorithm is follows:

The Bin B we assume that without loss of generality that follow the constraints.

Constraint 1: $\sum_{j=1}^{j=m} CPU_i <= MCPU_J$,

Constraint 2: $\sum_{j=1}^{j=m} MEM_i <= MMEM_i$,

Constraint 3: $Size_i <=$ Remaining Size of VM_J(if possible without opening a new VM. In order to make the resources available)

Constraint 4: $p_i = VMP_j$.

To define our new algorithm, that3BHBPTG= $\sum_{j=1}^{j=m} cpu_i \odot MMEM_i <= MCPU_j \odot MMEM_j$ and $Size_i <$Remaining Size of VM_J, PTi = VMP_j.In this3D Hybrid Bin Packing with Task Grouping Algorithm, each VM has some attributes such as Maximum CPU utilization in percentage called as MCPU (assumed 80% as optimum), Maximum Memory utilization in percentage called as MMEM (assumed 80% as optimum), Remaining time that opened VM needs to process that has already allocated tasks, represented as Rs and platform of the VM as VMP (Operating System or Environment). Every task has the attributes such as CPU utilization for the task to complete, in percentage defined as CPU, memory utilization for the task to complete called as MEM in percentage, the size of the task that is expected execution time to complete called as **s** and platform required by VM to process this task denoted as VMP.

The algorithm assigns the task to the VM, by which satisfied all the above constraintsexcept constraint 3. When some VMs are working with tasks and remaining time in any of those VMs are greater than or equal to the new task's size then the task is assigned in that VM. If not the task is assigned to near smaller remaining time available VM. The Main objective of this constraint is to allocate the task to VM without extending the processing time of that VM. By which we can try to make the more number of VM available in the resource pool to allocate in the future.

3 DIMENSION HYBRID BIN PACKING With TASK Grouping (3DHBPTG) ALGORITM.		
Step 1	:	Set FQTS Timer Value in ms (as 0 / 200 / 400 / 600 / 800 / 1000 / 1200).
Step 2	:	i) FQTS←0 (initialize the FQTS Timer). ii) Start Timer.
Step 3	:	iii) Receive each Task with AT, iv) Collect DLT and PT from user for the above Task Received. v) Collect CPU %, RT, MEM, Size from Task database for the above Task Received. vi) Get UP from SLA. vii) Add the above Received Task in ReqT. (For new Task, Set CPU%, MEM, Size ← appropriate Maximum, RT ← set by Cloud Administrator).
Step 4	:	If FQTS reaches the assigned value do Step 5 to Step 10 simultaneously do steps 2 to step 4.
Step 5	:	Simultaneously call Resource-coordinator
Step 6	:	Order (Sort) ReqT by DLT (immediate followed by low to high), UP and PT (low to high), SS (high to low) of the task.
Step 7	:	Select RP from AVM, find m
Step 8	:	Select first m Tasks in ReqT Queue as TP. i.e m=n.
Step 9	:	Order (Sort) TP by RT and Order (Sort) MEM, CPU % in Descending order
		Order (Sort) RP by RT and Order (Sort) MMEM, MCPU in Descending order.
Step 10	:	i) Select top VM from RP we refer as VM_j ii) Make TG_j from TP. iii) Schedule and Allocate – VMj ← TGj. iv) Simultaneously call follow-up do next step.
Step 11		Remove VMj from RP and TGj from RP and TP respectively
Step 12		Do steps 10 and 11 until TP or RP is nil.
Step 13	:	If RP has VM surrender to AVM and TP has task return to ReqT.
Step 14		Go-to Step 2.

Resource-Coordinator		
Step 1	:	If FQTS reaches the assigned value do steps 2 to 5
Step 2	:	Get status of each VM

Step 3	:	Count number of VMs are free and ready to serve
Step 4	:	Get number of VMs under service with more space in memory and CPU % of each Resource type (RT)
Step 5	:	Transfer the values to 3DHBP.
Step 6	:	Go-to step 1

Follow-up		
Step 1	:	Check any VM completes its service then do steps 2 to 8 else do step 1.
Step 2	:	Calculate Waiting time of T_i as $WT_i = DT_i - AT_i$
Step 3	:	If any task T_i released from VM_j then do step 3
Step 4	:	i) Calculate Service Time $ST_i = RelT_i - DT_i$ ii) Find Total Time of $TOTT_i = WT_i + ST_i$ iii) Find total turnaround time = $\sum TOTT_i$
Step 5	:	Write Log Report for each event
Step 6	:	Find Success Score SSi
Step 7	:	Update all VMs and Tasks Attributes.
Step 8	:	Do Step 1

Experimental setup:

We implement our proposed multi-pass 3 Dimension Hybrid Bin Packing with Task Grouping algorithm using the Java 1.7.0.75 and also we implement the VISBP [compare 1] algorithm, MinTotal Dynamic Bin Packing [compare 2], andBottom-level and top-level GA (BTGA) [compare 3] in the following hardware configurations. We have 3 modules of IBM Blade server with Xeon processor with 8 GB RAM and 1.5 TB storage act as cloud cluster with 26 systems of Intel I3 core – 64 bit processors with 4 GB RAM and 500 GB Storage. We assigned 10 system as cluster 1, 8 system belongs cluster 2 and cluster 3. We installed VMware to make Virtual Machine. Each Physical machine has 3 VMs of 2 1GB RAM with 1 CPU core and 1 VM with systems 2 GB RAM and 2 CPU Cores. In our implemented system, we have 20 VM with 1 GB RAM and 1 CPU Core and 10VMs with 2GB RAM 2 CPU cores in cluster 1. 16 VM with 1 GB RAM and 1 CPU Core and 8 VMs with 2GB RAM, 2 CPU cores in cluster 1 and 16 VM with 1 GB RAM and 1 CPU Core and 10VMs with 2GB RAM, 2 CPU cores in cluster 3. As a whole we have 78 VMs.

Solution specification:

We formulate a test bed to evaluate the proposed algorithm. First we try to find the optimum FQTS for our set of 5 test tasks generated randomly. We evaluate our proposed 3D Hybrid Bin Packing with Task Grouping algorithm with all possible scheduling methods such as First Fit (FF), Any Fit (AF), Best Fit (BF) and Worst Fit (WF) with various FQTS such as FQTS =0, 200 m.sec, 400 m.sec, 600 m.sec, 800 m.sec and 1000m.sec.

First we test our algorithm with FQTS with 0. This means, algorithm works as a pure dynamic Bin Packing algorithm. For testing, we follow the FF (First Fit), the task arrived in a Poisson distribution, are assigned in the VM which opened earlier if the constraints satisfied. If no VM is fit for assign then a new VM is opened to process. We test this algorithm with our 5 set of task list generated. The total time to complete to process all the tasks by our algorithm is compared and a graph drawn.The algorithm is tested for FQTS = 0 and AF (Any Fit), the received task directly assigned to any randomly chosen VM. If the chosen VM could not satisfy the constraints then the algorithm tries to make use of any other available a new VM is opened to process. The total time to complete to process all the tasks by our algorithm is compared and a graph drawn. For Best fit(BF) for FQTS = 0, the received task find the available VM whose CPU percentages and memory percentage is exactly or very nearly matches, in order to enhance the utilization of the resource. For Worst Fit (WF), the received task is assigned to the VM whose CPU and Memory is least used for other tasks. If no VM is available then a new VM is opened to process the task.We test our algorithm with FQTS = 200 m.sec is fixed to activate the hybrid type; here the receiving system receives all the arrived tasks for the period of 200 m.sec. After 200 m.sec the queued tasks are sorted on priority and maximum to minimum size of the tasks and assign to the VM based on methods such as FF, AF, BF and WF. The total time to complete is noted. Same as the FQTS = 400,600,800 and 1000 m.sec for all the assigning methods.

From our evaluation of our proposed algorithm 3DHBPTG with FQTS = 0, 200, 400, 600, 800, and 1000, we plot the graphs which are all shown in figure 1 (a), (b), (c), (d) and (e). By carefully analyzing those graph we understand, our algorithm yields better optimum lowest lime to complete the service for the tasks given. From this we conclude that our proposed algorithm can produce expected results at FQTS = 600. From our evaluation of our proposed algorithm 3DHBPTG with FQTS = 0, 200, 400, 600, 800, and 1000, we plot the graphs which are all shown in figure 1 (a), (b), (c), (d) and (e). By carefully analyzing those graph we understand, our algorithm yields better optimum lowest lime to complete the service for the tasks given. From this we conclude that our proposed algorithm can produce expected results at FQTS = 600.

Comparison:

Our 3d hybrid Bin packing with Task Grouping algorithm is compared to system already proposed by other researchers. First we compare with MinTotal DBP [15] Minimum total Dynamic Bin Packing

proposed by Yusen Li and *et al*. Second we compare our proposed algorithm with HGA [26] algorithm with B-level and T-level priority scheduling algorithm proposed by Verma, Amandeep, and Sakshi Kaushal. Third, we compare with VISBP [27] a multi-dimensional Bin Packing algorithm proposed by Song, Weijia and *et al*.

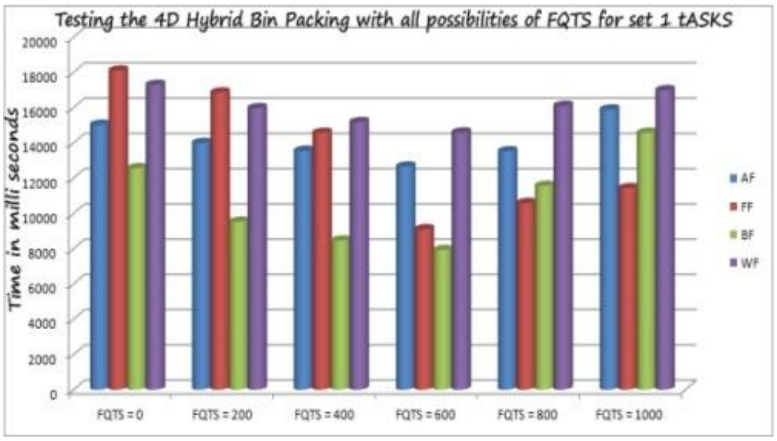

Fig. 1(a): Set 1 Tasks

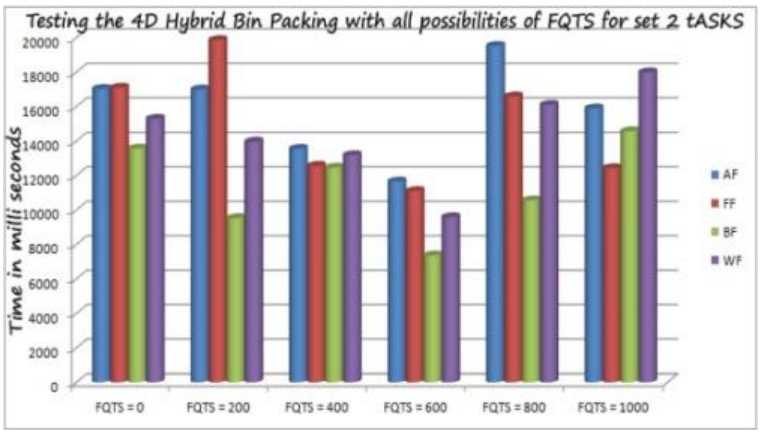

Fig. 1(b): Set 2 Tasks

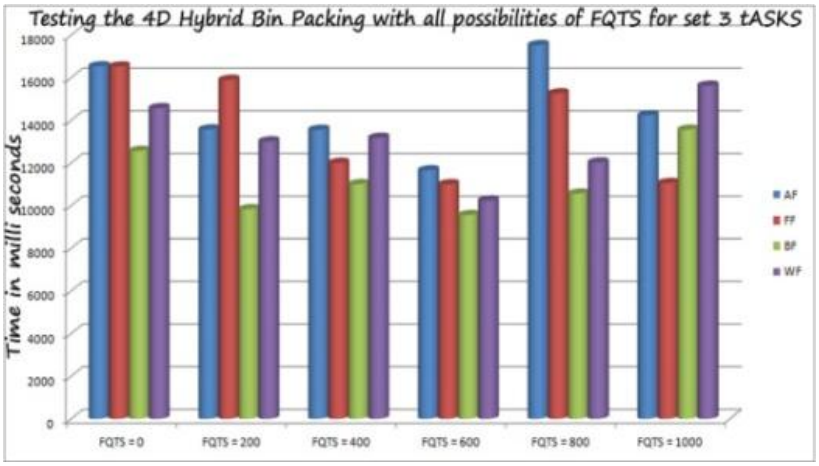

Fig. 1(c): Set 3 Tasks

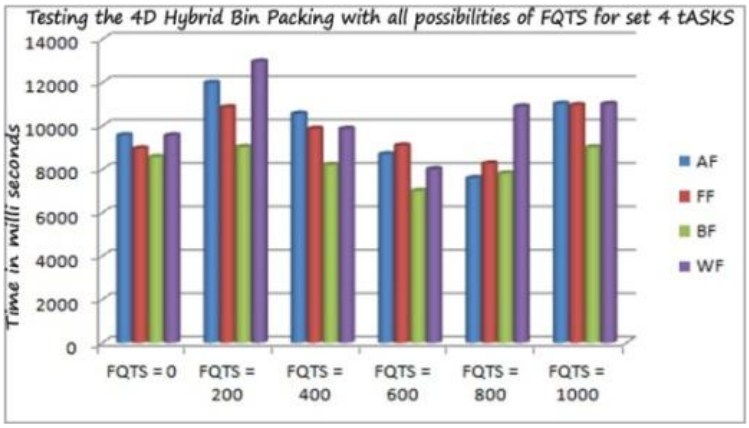

Fig. 1(d): Set 4 Tasks.

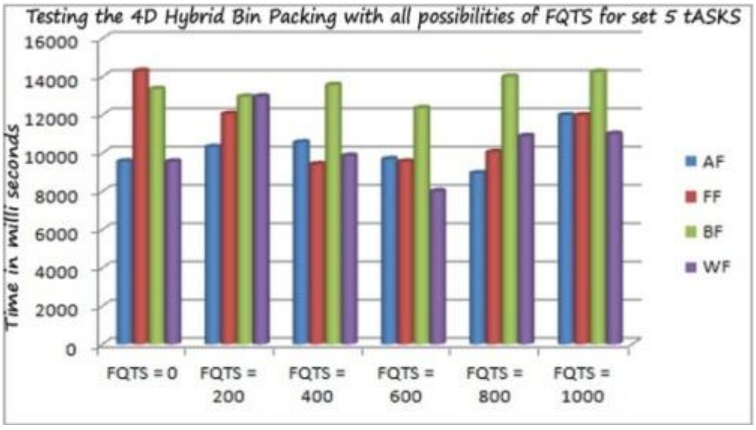

Fig. 1(e): Set 5 Tasks

Each task (item)irequested by clients is associated with a 3-tuple (CPU(i), MEM(i), and RT(i) where CPU(i), MEM(i), and RT(i), which denotes the CPU utilization percentage, memory utilization percentage, and Resource Type (such as Operating System and or environment to execute) required to execute respectively. We calculate, on all the Scheduling implemented, parameters such as arrival timeAT(i), and the departure time DT(i) with this we find the length of waiting time of the task WT(i), Service Time in the Cloud resource and total time of the task for measuring the performance of the cloud system.

From the graph (Figure2) shows the performance of our proposed system. The Graph (figure 3) shows the performance of our system with other three systems. From the graph we understand that our proposed 3DHBPTG algorithm schedule the tasks with better reliability and availability compare with other systems.

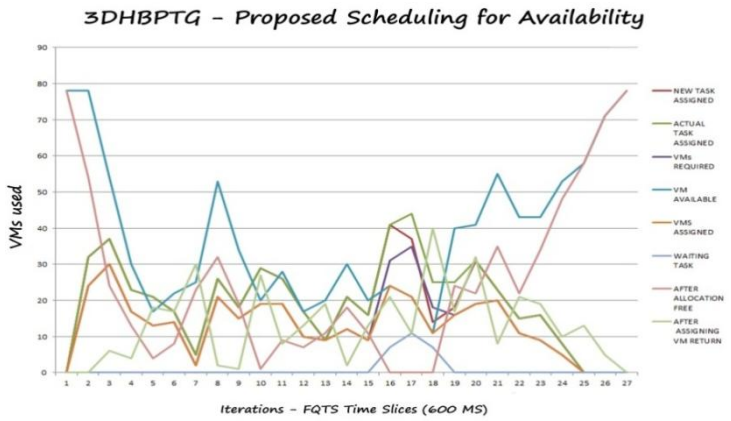

Fig. 2: Proposed Algorithm – 3DHBPTG.

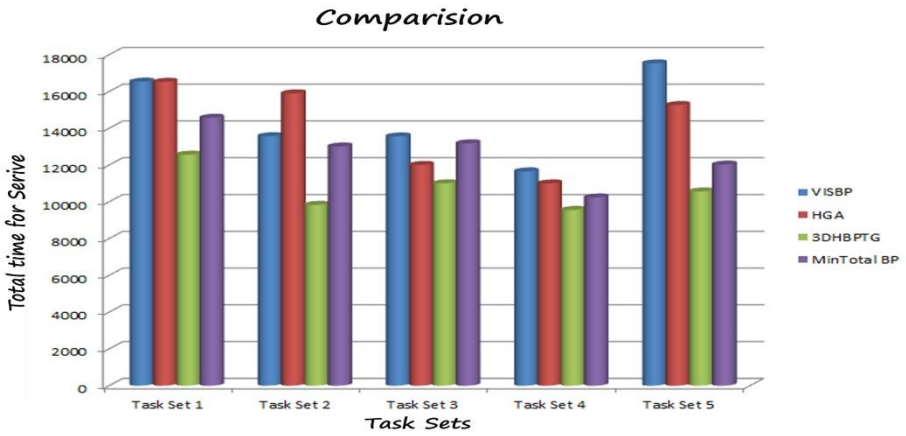

Fig. 3: Performance comparison with other Scheduling algorithms.

Conclusion:

Our Proposed algorithm 3 Dimension Hybrid Bin Packing with Task Grouping algorithm (3DHBPTG) can accommodate many lite Tasks in one VM (Resource) and effectively improves the Reduces' utilization into maximum possible. On proper analysis and evaluation, we understand that our proposed system performs better than other scheduling algorithm proposed by other researchers. To improve more we are planning improve the scheduling algorithm, we have a plan to add a monitoring phase to understand and follow the cloud traffic status slightly before the situation. So we can minimize the waiting time and total turnaround time of the requested tasks.

REFERENCES

, Yusen, Li., Xueyan Tang and Wentong Cai, 2014. "On dynamic bin packing for resource allocation in the cloud." In *Proceedings of the 26th ACM Symposium on Parallelism in Algorithms and Architectures*, pp: 2-11. ACM.

Ahuja, Sanjay, P., Sindhu Mani and Jesus Zambrano, 2012. "A survey of the state of cloud computing in healthcare." *Network and Communication Technologies*, 1(2): 12.

Bein, Doina, Wolfgang Bein and Swathi Venigella, "Cloud storage and online bin packing." In *Intelligent Distributed Computing V*, pp. 63-68. Springer Berlin Heidelberg.

Berkey, O. Judith and Y. Pearl Wang, 1987. "Two-dimensional finite bin-packing algorithms." *Journal of the operational research society*, pp: 423-429.

Binitha, S. and S. Siva Sathya, 2012. "A survey of bio inspired optimization algorithms." *International Journal of Soft Computing and Engineering*, 2(2): 137-151.

Bitam, Salim, 2012. "Bees life algorithm for job scheduling in cloud computing", In *Proceedings of the Third International Conference on Communications and Information Technology*, pp: 186-191.

Coffman Jr, G. Edward, R. Michael Garey and S. David Johnson, 1996. "Approximation algorithms for bin packing: a survey." In *Approximation algorithms for NP-hard problems*, pp. 46-93. PWS Publishing Co.,

Feller, Eugen, Louis Rilling and Christine Morin, 2011. "Energy-aware ant colony based workload placement in clouds." In *Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing*, pp: 26-33. IEEE Computer Society.

https://en.wikipedia.org/wiki/Bin_packing_problem.

Javanmardi, Saeed, 2014. "Hybrid job scheduling algorithm for cloud computing environment." Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014. Springer International Publishing.

Johnson, S. David, 1973. "Near-optimal bin packing algorithms." PhD diss., Massachusetts Institute of Technology.

Kagadis, C. George, Christos Kloukinas, Kevin Moore, Jim Philbin, Panagiotis Papadimitroulas, Christos Alexakos, G. Paul Nagy, Dimitris Visvikis and R. William Hendee, 2013. "Cloud computing in medical imaging." *Medical physics*, 40(7): 070901.

Kaur, Kamaljit, Amit Chhabra and Gurvinder Singh, 2010, "Heuristics based genetic algorithm for scheduling static tasks in homogeneous parallel system."*international journal of computer science and security*, 4(2): 183-198.

Liu, Yu., Changjie Zhang, Li. Bo and Jianwei Niu, 2015. "DeMS: A hybrid scheme of task scheduling and load balancing in computing clusters." *Journal of Network and Computer Applications*.

Martello, Silvano, David Pisinger and Daniele Vigo, 2000. "The three-dimensional bin packing problem." *Operations Research*, 48(2): 256-267.

Mathew, K. Tojo, Chandra Sekaran and Jithin Jose, 2014. "Study and analysis of various task scheduling algorithms in the cloud computing environment." In*Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on*, pp: 658-664. IEEE, 2014.

Mell, P., T. Grance, 2011. The NIST definition of cloud computing. NIST Special Publication, pp: 145–800.

Sandholm, Thomas, Julie Ward, Filippo Balestrieri and A. Bernardo Huberman, 2015. "QoS-Based Pricing and Scheduling of Batch Jobs in OpenStack Clouds."*arXiv preprint arXiv:1504.07283*.

Senthil Kumar, S.K. and P. Balasubramanie, 2012. "Dynamic scheduling for cloud reliability using transportation problem," Journal of Computer Science, 8(10): 1615–1626.

Singh, R.M., K.S. Sendhil Kumar, N. Jaisankar, 2013. Comparison of Probabilistic Optimization Algorithms for resource scheduling in Cloud Computing Environment, International Journal of Engineering and Technology (IJET), 5(2): 1419 – 1427.

Song, Weijia, Zhen Xiao, Qi Chen and Haipeng Luo, 2014. "Adaptive resource provisioning for the cloud using online bin packing." *Computers, IEEE Transactions on*, 63(11): 2647-2660.

Verma, Amandeep and Sakshi Kaushal, 2014. "Deadline constraint heuristic-based genetic algorithm for workflow scheduling in cloud." International Journal of Grid and Utility Computing, 5(2): 96-106.

Vilaplana, Jordi, Francesc Solsona, Francesc Abella, Rosa Filgueira and Josep Rius, 2013. "The cloud paradigm applied to e-Health." *BMC medical informatics and decision making*, 13(1): 35.

Wedde, F. Horst, Muddassar Farooq and Yue Zhang, 2014. "BeeHive: An efficient fault-tolerant routing algorithm inspired by honey bee behavior." In *Ant colony optimization and swarm intelligence*, pp: 83-94. Springer Berlin Heidelberg.

Zhang, Min-Xia, Bei Zhang and Yu-Jun Zheng, 2014. "Bio-Inspired Meta-Heuristics for Emergency Transportation Problems." *Algorithms*, 7.1: 15-31.

Zhang, Yanyong, Hubertus Franke, Jose Moreira and Anand Sivasubramaniam, 2003. "An integrated approach to parallel scheduling using gang-scheduling, backfilling, and migration." *Parallel and Distributed Systems, IEEE Transactions on*, 14(3): 236-247.

Zitzler, Eckart, Kalyanmoy Deb and Lothar Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results." *Evolutionary computation*, 8(2): 173-195.