Improve Quality of Service in Optimization of Job Scheduling using a Hybrid Approach

Prof. Dilip Saini[1] , Dr. Sachin V. Solanki[2]
[1]Department of Computer science & Technology ,RAMA University , kanpur,
[2]Department of Information Technology, KDK College of Engineering, Nagpur

***Abstract*- Computing resources are scattered under unlike ownerships each one possesses their particular access tactic, outlay and a range of constraints. Now a day, the newest paradigm to come into view is Cloud computing environment that guarantees stable functionalities offered by high-end data centers which are constructed based on storage, computation and virtualization concepts. Users are informed and skilled to get information and use applications throughout the globe on claim from everywhere and anytime in cloud environment. Virtualization is an appellation that associates to the abstraction of computer resources. The functionalities which are normally called as "Cloud Services," are passed through various high-end data centers. To pick the most capable service or functionality from the appropriate set of services, Job Scheduling on basis of an optimized approach by hybridizing Genetic Algorithm and Tabu Search algorithm for the Cloud Environment using CloudSim Simulator is proposed in this research paper.**

Keywords: Job Scheduling, Cloud Computing, Genetic Algorithm, Tabu Search, Virtual Machine, Physical Machine.

I. INTRODUCTION

Cloud computing creates the whole thing more striking as functionality or service and its services to be more immensely reachable, extendable, and self-working [1]. A Job or (sometimes task) scheduling scheme performs a significant part in order to accomplish users' Job QoS needs and requirements and utilize resources competently in a cost-effective manner. To select most competent functionality or service from the appropriate various set of services, Job Scheduling based on Hybrid Algorithm HTSGA (hybridization of Tabu Search with Genetic Algorithm) is proposed as research work for the Cloud Computing Environment using CloudSim [14]. Job scheduling allots suitable jobs or tasks to hosts thus the implementation is finished to assure objective functions forced by users. In Cloud environment, VM provisioning is carried out at 2 stages: main, at the host stage and another, at the Virtual Machine stage. In the first and host stage, the virtual machine (VM) resources are mapped and scheduled to physical machines (or hosts) in accordance to get the most suited scheduled solution to attain the system load stability to the highest coverage. The foremost aim of mapping is to map compliant VMs to PMs (physical machines) appropriately with flexible time, that is basically, engrosses getting an appropriate sequence to enhance resource effectiveness. In Virtual Machine stage, the Virtual Machine consigns predetermined quantity of existing execution command to the separable application task units which are set up inside its computation engine. This research work proposes approaches at both the above mentioned levels to solve Job Scheduling problem to improve resource utility by hybridizing genetic algorithm and tabu search algorithm.

II. REVIEW OF LITERATURE

Numerous computing paradigms and research works of various researchers to provide the idea of computing utilities, cloud computing and presents novel initiatives for new Internet services are elaborated in this section.

1.1. Cloud Computing, Grid Computing and Job Scheduling based on QoS

A Berkeley outlook of Cloud environment [2] describes cloud computing, the idea of computation as a service, likely to modernize many IT industries. Volker Hamscher [3] discusses a characteristic scheduling structure that arises in computational grids. Zhiguang, S. [4] has given short description of the assessment of hierarchical task and job scheduling. Ajith Abraham, Rajkumar Buyya and Baikunth Nath [12] tried to handle the dynamic job scheduling to the extensively distributed computing jobs. Luqun Li [5] focused on the study of the differentiated QoS necessities of resources user's job and then constructing the analogous non preemptive precedence queuing method for the tasks.

1.2. Heuristic Algorithms

Di-Wei Huang and Jimmy Lin [6] implemented a Genetic algorithm to scale the population by MapReduce. Attila Csaba Marosi [7] notified existing connectivity and interoperability factors of Clouds, Grids and Clusters, and provided solutions to overcome these issues. A Gang Scheduling that is a specific example of parallel job/task scheduling is implemented by Zafeirios C. Papazachos and Helen D. Karatza [8]. Several approaches have been used to answer re-entrant flow shop scheduling problem where Danping and Lee [9] has a explained this problem and the associated approaches are proposed. Hegliang Shi, Guangyi Bai and Zhenmin Tang [10] discussed that Ant Colony Optimization can progress job scheduling system dynamically with probability matrix, hormone matrix and cost matrix, and then simulates. Seonho Kim and Jon B. Weissman [11] proposed new GA intensive system which automatically decays information into computation and communication resources. Jianhua Gu, etc. [13] presented a mapping process for load scheduling of Virtual Machine resources using GA. A new method have been proposed by Mohammad Ranjbar and Mojtaba Najafian Razavi [14] to parallel generate the outline and scheduling assessments in a job shop background.

III. THEORETICAL EVALUATIONS OF TECHNOLOGIES

3.1. The Emergence of Cloud Computing, Virtualization and Cloud Sim Overview

The foremost approach behind the Cloud structure is providing storage, computation and application "as a functionality or service". Virtualization is a technique that abstracts away the facts of physical hardware and offers virtualized resources for applications. A virtualized server is normally called a VM (virtual machine) is a virtual implementation of a system that executes applications and jobs like an actual system. The cloud decides how those virtualized resources are allocated delivered, and presented. A very common concern is "Experimental evaluation is too much of work and "expensive" for computing researchers and academicians". A more practicable substitute is the utility of simulation tools (ex. CloudSim). CloudSim simulator has been used in this research work that demonstrates mapping of CPU resources at 2 stages: Host stage and Virtual Machine stage. Time shared Cloudlet Scheduler policy has been used for the implementation in this research work for simulating tasks to VMs.

IV. SCHEDULING SYSTEM DESIGN

4.1. Scheduling Problem and Policy

The job of the scheduling system structure is to allocate resources to an application request for a definite extent of period. In the circumstance, scheduling works with the allotment of resources like processor nodes to user application requests for large computations. These requests are typically called jobs. To pick the most capable service from the fittest service sets, Job Scheduling based on Hybrid Algorithm HTSGA (such as Genetic Algorithm with Tabu Search) is a research work for the Cloud Computing Environment.

4.2. Objective Function

The most important intention of schedule/mapping system has to map jobs to the appropriate available VMs in malleable time that engages getting an appropriate succession in that jobs can be accomplished under particular logic restriction.

Definition1

The foremost objective of the job scheduling system has to establish an accurate mapping sequence so that jobs are performed following certain (logic) restraints. This problem comes under a class known as NP-complete problems, whose execution time for an exact solution raises with N as exp (const.× N), becoming quickly excessive in cost as N raises. So here, scheduling problem consider Jm user jobs m = {1,2,….M} on Rn dissimilar resources n={1,2,….,N} with a purpose of reducing the completing time and employing the resources efficiently. A task/job from Jm to be proceeded in system resource Rn, till accomplishment

V. SCHEDULING ALGORITHMS

5.1. Genetic Algorithm (GA)

An excellent scheduling process is estimated to generate very excellent, if not optimum schedules, in accordance to the objective function while not taking 'much' time and 'many' resources to decide the schedule. GAs are basically heuristic procedure, that mainly enlightens optimization problems depending on the genetic progression of biological organisms. A typical genetic algorithm is illustrated in below.

```
Stag 1 create initial population.
Stag 2 Estimate populations by calculating fitness value.
Stag 3 Select parents from Roulette Wheel List arranged by Roulette Wheel Selection
Stag 4 Use Crossover to produce children.
Stag 5 Use Mutation to children.
Stag 6 Choose parents and children to produce the firsthand population for the subsequent generation.
Stag 7 if finishing condition is come to an end, otherwise go to Step 2.
```

Algorithm 1: Basic GA

The initialization process is an additional significant concern in entirely GAs because it must produce an arbitrary preliminary population which extent in the entire search space.

Definition 2

Reign of an individual in the complete population can be determined by calculation of the fitness function. The fitness value (Eq. 1) measures importance of an individual.

$$F = UV/UC \qquad (1)$$

Where F = Fitness Value, UV = Utilized Value UC = Unique Count (i.e. the number of physical machines or hosts utilized)

$$UV = \sum (VMCi/PMCj) \qquad (2)$$

For i=0, 1, 2 …n VMs and j=0, 1, 2 ...m PMs Where VMC = Virtual Machine Capacity (or VM Capacity) PMC = Physical Machine Capacity (or PM Capacity)

Crossover operator

Arbitrary pair of chromosomes is selected for crossover operation, and then an arbitrary point in the initial chromosome is chosen. Roulette Wheel selection is used for a selection of a couple of chromosomes. A crossover operation (Algorithm 2) is performed for rejoining 2 strings to get a superior chromosome.

```
Choose randomly a crossover position p
Offspring1 content of parent1's gene[k], k □ [0; p − 1] and parent2's gene[v] not (□) gene[k], v □ [0; m]; Offspring2 is content of parent2's gene[k],k □ [0; p − 1]; And parent1's gene[v] not (□) gene[k], v □ [0; m];
```

Algorithm 2: Crossover Operation

Mutation

This process supplements new information in an arbitrary manner to the genetic search process and eventually supports not to get caught at local optima.

```
Choose randomly Gene g;
Get the first successor Suc(g) from g to the finish ;
Select randomly a Gene k □ (G, Suc(g)) ;
Get the first predecessor Pre (k) from Suc(g) to the start;
if Pre(k) > g then Exchange position of gene g and gene k;
Generate a new offspring;
            End if
```

Algorithm 3: Mutation Operation

5.2. Tabu Search (TS)

TS is a heuristic approach used to work out combinatorial optimization problems. The main concept behind TS is to follow local exploration when it finds a local optimal by letting non-enhancing moves; going back to already explored solutions is disallowed by using memories, that are tabu lists which store up to date account of the search.

```
Stag 1 Generate current solution y.
Stag 2 Declare the Tabu List.
Stag 3 Calculate fitness(y).
Stag 4 till set of neighbour solutions Y'' is not finished.
Stag 4.1 Produce neighbour solution y'' from existing solution y
Stag 4.2 Supplement y'' to Y'' only if y'' is not stored in tabu or if one of the Aspiration Criterion is fulfilled.
Stag 5 Pick the finest neighbour solution y* in Y''.
Stag 6 If fitness (y*) > fitness (y) then y = y*.
Stag 7 Modify Aspiration Criteria and Tabu List
Stag 8 If termination condition come to an end, otherwise go to Step 4.
```

Algorithm 4: Tabu Search

VI. HYBRID TABU SEARCH AND GENETIC ALGORITHM (HTSGA)

6.1. Need for the Hybrid Algorithm

Scheduling problems demonstrate such richness and diversity that no single scheduling method is satisfactory. The effectiveness of a local search in attaining a local optimum integrates the capability of a genetic algorithm in isolating the most capable basins of the

Search area. Hence, integration of a local search into a genetic algorithm can create in a proficient algorithm. During the hybrid search procedure, GA begins with a set of preliminary solution and produces a set of new solutions. On each set of new solution, TS executes a local search to enhance them. Then GA considers the improved solution of TS to carry on with parallel evolution. Integrating a local search approach also initiates an explicit enhancement operator which can produce high excellence solutions.

6.2 Hybridization of Tabu Search and GA

The core intent of the research paper has to decrease execution time, system cost, completion time, resource utilization and power consumption while satisfying all constraints. Hybridization of TS with GA formulates the algorithm more robust. In the hybrid TS and GA (HTSGA) technique, initialization, crossover and mutation in GA is substituted by initialization, crossover and Tabu search. In place of random mutation modifies each member of the population goes through a separate optimization procedure illustrated by a tabu algorithm.

```
Stag 1 create initial population.
Stag 2 Set the Tabu List
Stag 3 Evaluate populations by calculating fitness value.
Stag 4 Select parents from Roulette Wheel List arranged by Roulette Wheel Selection
Stag 5 Perform Crossover to create children.
Stag 6 Perform Mutation to children.
Stag 6.1    Initialize children as y
Stag 6.2    Till neighbour solutions set Y'' is not finished.
    Step 6.2.1 Create neighbour solution y'' from existing solution y
    Step 6.2.2 Supplement y'' to Y'' only if y'' is not stored in tabu or if minimum one
      Aspiration Criterion is fulfilled.
Stag 6.3 Pick the finest neighbour solution y* in Y''.
Stag 6.4 If fitness(y*) > fitness(y) then y = y*.
Stag 6.5 Modify Tabu List and Aspiration Criteria
Stag 6.6 if predefined state achieved, else go to Stag 6.2.
Stag 7 Choose parents and children to generate the firsthand population for the subsequent generation.
Stag 8 if finishing state is encountered, else go to Step 4
```

Algorithm 5: HTSGA

VII. EXPERIMENTS AND EVALUATION

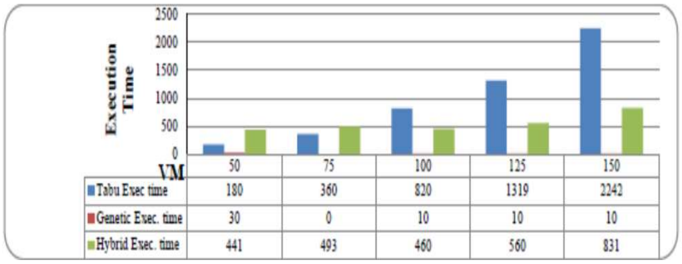
7.1. Experimental Evaluation

In this research work, by default VM Allocation Policy is replaced by our algorithms including Tabu Search, Genetic Algorithm and HTSGA. Time-shared policy has been used in the simulation of jobs to VMs. Numbers of experiments have been done considering different QoS parameters using three approaches tabu search, genetic algorithm and HTSGA. Their comparative results are also shown through different charts.

Experiment 1: Execution time comparison

Host is represented as an actual computation machine in a Cloud: it is allotted a reassigned execution competency (represented in MIPS), storage, memory, and a policy for assigning execution cores to VMs. Execution time is the time required for allocation of virtual machines to physical machines at Host level. Figures 7.1 shows the execution time required to schedule number of VMs (Ranging from

75, 100, 125…) to PMs (50, 75 …). Hence it is observed that in most of cases, time for genetic algorithm is smaller as compared to TS and HTSGA. GA. is able to perform a parallel search to determine the global search area. During the parallel search method GA gains meaningful data about what has been obtained from previous generations. GA looks for the solution from the entire population rather than a particular point. Tabu Search (TS) executes on the individual string, which are points on the solution area. Execution time for the hybrid algorithm HTSGA is the enhancement of the tabu search but not as compared to Genetic algorithm.

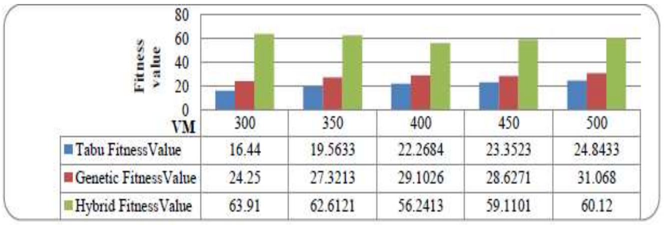


VM	50	75	100	125	150
🟦 Tabu Exec time	180	360	820	1319	2242
🟥 Genetic Exec. time	30	0	10	10	10
🟩 Hybrid Exec. time	441	493	460	560	831

Figure 7.1: Execution Time comparison for PM = 75

Experiment 2: Fitness Value Comparison

Better the fitness value, better the approach as fitness value determine the reliability that is needed to provide stable and reliable performance. Figure 7.2 shows the fitness value comparison for three approaches and it is observed that hybrid algorithm HTSGA has better fitness value in all the cases.

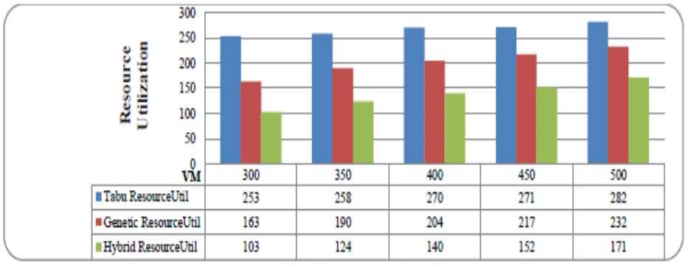


VM	300	350	400	450	500
🟦 Tabu FitnessValue	16.44	19.5633	22.2684	23.3523	24.8433
🟥 Genetic FitnessValue	24.25	27.3213	29.1026	28.6271	31.068
🟩 Hybrid FitnessValue	63.91	62.6121	56.2413	59.1101	60.12

Figure 7.2: Fitness Value Comparison for PM = 300

Experiment 3: Resource Utilization Comparison

The main objective of an efficient scheduling approach is to properly utilize physical resources. Figure 7.3 shows the comparisons among three approaches and in all in-stances hybrid algorithm HTSGA has properly utilized the physical resources and has less physical machine count as compared to others approaches. machines but GA and TS could not.

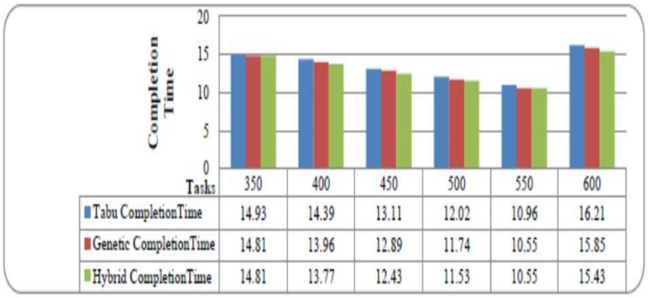


VM	300	350	400	450	500
🟦 Tabu ResourceUtil	253	258	270	271	282
🟥 Genetic ResourceUtil	163	190	204	217	232
🟩 Hybrid ResourceUtil	103	124	140	152	171

Figure 7.3: Resource Utilization Comparison (for PM = 300)

Experiment 4: Completion Time Comparison

Tasks required to be accomplished within a deadline in the real time environment. Figure 7.4 shows the completion time of different tasks on different VMs and ob-served that in most of the cases hybrid algorithm produces minimum completion time.



Tasks	350	400	450	500	550	600
🟦 Tabu CompletionTime	14.93	14.39	13.11	12.02	10.96	16.21
🟥 Genetic CompletionTime	14.81	13.96	12.89	11.74	10.55	15.85
🟩 Hybrid CompletionTime	14.81	13.77	12.43	11.53	10.55	15.43

Figure 7.4: Completion Time comparison (for PM=300 VM=400)

VIII. CONCLUSION

The main stress in Cloud Computing has to generate the schedules at an optimal ex-tent of time. Particularly as the demand rises, when the number of jobs and the re-sources starts towering up, traditional GAs become time consuming. Various QoS metrics have been defined for better performance. Considering different QoS parameters metrics have been defined for better performance. Considering different QoS parameters, various experiments are performed and it is observed that hybrid algorithm (HTSGA) performed much superior than genetic algorithm and tabu search. Comparative charts are given, showing the better performance of HTSGA.

IX. CONCLUSIONS

With the rapid growth of cloud computing platforms and services, cloud security is becoming a key priority for all players (i.e., individuals, companies, and cloud providers). Today, cloud computing is clearly one of the most enticing technology areas of the current times due, at least in part to its cost-efficiency and flexibility. When thinking about solutions to cloud computing adoption problem, it is important to realize that many of the issues are essentially old problems in a new setting, although they may be more acute. In this paper, we presented A study of Cloud Security: challenges and Concerns, illustrating their impact on the confidentiality, integrity, and availability properties.

REFERENCES

[1] Rajkumar Buyya, James Broberg, Andrzej Goscinski, “Cloud Computing: Principles and Paradigms” Copyright 2011 John Wiley & Sons, Inc.

[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, ‘‘Above the Clouds: A Berkeley view of Cloud Computing,’’ Univ. of California, Berkeley, Tech. Rep., 2009.

[3] Hamscher, V., et al., “Evaluation of Job-Scheduling Strategies for Grid Computing.” LECTURE NOTES IN COMPUTER SCIENCE, 2000, pp. 191-202.

[4] Zhiguang, S. and L. Chuang., “Modeling and Performance Evaluation of Hierarchical Job Scheduling on the Grids. in Grid and Cooperative Computing”, GCC 2007. Sixth Interna-tional Conference, 2007.

[5] Luqun Li, “An Optimistic Differentiated Service Job Scheduling System for Cloud Com-puting Service Users and Prividers”, Third International Conference on Multimedia and Ubiquitous Engineering, 2009: pp 295-299

[6] Di-Wei Huang, Jimmy Lin, “Scaling Populations of a Genetic Algorithm for Job Shop Scheduling Problems using MapReduce”, 2nd IEEE International Conference on Cloud Computing Technology and Science, 2010.

[7] Attila Csaba Marosi and P´eter Kacsuk, “ Workers in the Clouds”, 19th International Eu-romicro Conference on Parallel, Distributed and Network-Based Processing, 2011: pp 519-524

[8] Zafeirios C. Papazachos, Helen D. Karatza, “Performance evaluation of bag of gangs scheduling in a heterogeneous distributed system,” The J. of Syst. and Software, pp. 1–9, 2010.

[9] D. P. Lin and C. K. M. Lee, "A review of the research methodology for the re-entrant scheduling problem," International Journal of Production Research, 2010.

[10] Hengliang Shi, Guangyi Bai, Zhenmin Tang, “ACO Algorithm-based Parallel Job Scheduling Investigation on Hadoop”, International Journal of Digital Content Technology and its Applications. Volume 5, Number 7, July 2011

[11] Seonho Kim and Jon B. Weissman, “A GA-based Approach for Scheduling Decomposable Data Grid Applications” IEEE International Conference on Parallel Processing, 2004

[12] Ajith Abraham, Rajkumar Buyya and Baikunth Nath, “Nature’s Heuristics for Scheduling Jobs on Computational Grids”

[13] Jianhua Gu, Jinhua Hu, Tianhai Zhao, Guofei Sun, “A New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment”, JOURNAL OF COMPUTERS, VOL. 7, NO. 1, JANUARY 2012,pp42-52

[14] Mohammad Ranjbar & Mojtaba Najafian Razavi, “A hybrid met heuristic for concurrent layout and scheduling problem in a job shop environment”, Int J Adv Manuf Technol (2012) 62:1249–1260

[15] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajku-mar Buyya, “CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing En-vironments and Evaluation of Resource Provisioning Algorithms”, Journal of Software: Practice and Experience, Volume 41, Issue 1, Pages 23-50, January 2011.

[16] Hussin M. Alkhashai, Fatma A. Omara, “An Enhanced Task Scheduling Algorithm on Cloud Computing Environment”, International Journal of Grid and Distributed Computing, Vol. 9, No. 7 (2016), pp.91-100