

# MULTIOBJECTIVE QoS OPTIMAZATION BASED ON MULTIPLE WORKFLOW SCHEDULING IN CLOUD ENVIRONMENT

PaulRani.A<sup>1</sup>, M.GomathyNayagam<sup>2</sup>

PG Student, Department of CSE, Einstein College of Engineering, Tirunelveli, Tamilnadu, India<sup>1</sup>

Assistant Professor, Department of CSE, Einstein College of Engineering, Tirunelveli, Tamilnadu, India<sup>2</sup>

**ABSTRACT:** Cloud computing is an extension of parallel computing, distributed computing and grid computing. It provides secure, quick, convenient data storage and computing power with the help of internet. Cloud provides on-demand services based on user requirements. Whenever meet the different user with different QoS requirements scheduling the services is challenging one. Most of the existing papers for scheduling concentrate on cost or time or both. In this paper, the MQMCE schedule the services based on more than three QOS requirement such as time cost, reliability and availability. It evaluates performance for various test cases with different number of workflows and different set of QoS parameters for each workflow. The MQMCE results is the improved performance from the existing method such as reducing time effect, reducing cost effect as well as increase reliability and availability in a single objective manner.

**Keywords:** Cloud, Workflow, Qos, Scheduling.

## I. INTRODUCTION

A Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers[5]. Cloud computing brings a new level of efficiency and economy to deliver IT resources on demand and in the process it opens up new business models and market opportunities. The need of cloud computing is based on some of the requirements such as minimizing costs, business agility, reduce capital expenditure, eliminate over provisioning. Cloud computing is on demand services. These services are represented as workflows which is the collection of tasks that are processed based on service requirement. In cloud services provide to users at the same time or different users have different QoS requirements. Scheduling the services for different user requirement is difficult one. The scheduling strategy should be developed for multiple workflows with different QoS requirements. Mapping the multiple workflows to resources with multiple QoS requirement is NP hard. Scheduling of workflows is done by different algorithms to consider many QoS parameters. There are many[3][4][6][9][12] existing algorithms developed for different QoS such as cost based or time based or both. If consider more than two QoS parameters in to single objective function then the problem becomes more challenging. This paper, concentrated on multiple QoS like cost, time as well as reliability, where as no other algorithms concentrated on the more than two QoS parameters.

This proposed system use MQMCE algorithm which satisfies the multiple QOS such as reducing cost, time as well as increasing in reliability and availability in a single objective function.

Multi objective optimization is an area of multiple criteria decision making, that is concerned with mathematical optimization problems involving more than one objective function to be optimized simultaneously. For multiple-objective problems, the objectives are generally conflicting, preventing simultaneous optimization of each objective. Many, or even most, real engineering problems actually do have multiple objectives, i.e., minimize cost, maximize performance, maximize reliability, etc.

There are two general approaches to multiple-objective optimization. One is to combine the individual objective functions into a single composite function or move all butane objective to the constraint set. In the former case, determination of a single objective is possible with methods such as utility theory, weighted sum method, etc., but the problem lies in the proper selection of the weights or utility functions to characterize the decision-makers preferences. In practice, it can be very difficult to precisely and accurately select these weights, even for someone familiar with the problem domain. Compounding this drawback is that scaling amongst objectives is needed and small perturbations in the weights can sometimes lead to quite different solutions. In the latter case, the problem is that to move objectives to the constraint set, a constraining value must be established for each of these former objectives. This can be rather arbitrary. In both cases, an optimization method would return a single solution rather than a set of solutions that can be examined for trade-offs. For this reason, decision-makers often prefer a set of good solutions considering the multiple objectives.



The second general approach is to determine an entire Pareto optimal solution set or a representative subset. A Pareto optimal set is a set of solutions that are non dominated with respect to each other. While moving from one Pareto solution to another, there is always a certain amount of sacrifice in one objective(s) to achieve a certain amount of gain in the other(s). Pareto optimal solution sets are often preferred to single solutions because they can be practical when considering real-life problems

Genetic Algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution. GAs were introduced as a computational analogy of adaptive systems. They are modeled loosely on the principles of the evolution via natural selection, employing a population of individuals that undergo selection in the presence of variation-inducing operators such as mutation and recombination. Multiobjective genetic algorithm is an extension of genetic algorithms. The main work of MOGA is assignment of fitness. Once the fitness has been assigned to individuals, selection can be performed and genetic operators applied as usual. The MOGA proposed a Pareto-based ranking procedure of the individuals, where each individual is assigned a rank which counts how many individuals in the current population dominate them, In this way, on-dominated individuals are always assigned the same rank, independently of the trade-off surface.

Optimization is need for improving the results in highly. In this work first adding more user request parameter then check the value independently then finally optimize the parameters in an effective manner .Finally produce the results in a highly when compared to previous methods.

## **II. RELATED WORK**

Currently there are two workflow scheduling algorithms[2] Best-effort-based scheduling and Qos constraint based scheduling. Best-effort-based scheduling algorithms derived from either heuristics or meta-heuristics. The heuristics based approach fit only for particular type of problem, while Meta-heuristics method which provide a general solution method for developing a specific heuristic to fit particular kind of problem. QoS constraint based scheduling based on user level Qos constraint. Scheduler must determine Qos constraint on each task in workflow, such that entire workflow is satisfied. Cost and Execution time is two constraints in Qos. Based on that constraint, two approaches are introduced. A compromised Time-Cost scheduling algorithm[4] uses Batch mode as scheduling method, two parameters passed for this algorithm to achieve QoS like cost and time. It is used to reduce cost and time with an array of workflow instances.

A Budget Constrained Scheduling of Workflow Applications on Utility Grids using Genetic Algorithms[9] indicate most of the applications are executed based on their control or data dependencies which minimize the execution time on grids. This paper focuses on budget constrained based scheduling. Grid technologies have progressed towards a service oriented paradigm. That enable a new way of service provisioning based on utility computing models .in such cases user uses "pay-per-use" Grids in which execution cost must be considered based on user's Qos constraint. Generally processing time and cost are two Qos constraint. This paper uses utility grids which concentrate on both the execution time and cost. This is used genetic algorithm based on scheduling heuristic to minimize execution time while meeting user's budget constraints. Using new way of genetic algorithm and test with new simulated testpad also used markov decision process. This way of utility grids very useful to users to consume that service transparently over a secure, shared and standard world wide h/w environment.

Improved cost-based algorithm[5] for task scheduling which uses Batch mode as scheduling method. Cost and performance are the two parameters passed for this algorithm. It is used to measure both resource cost and computation performance.

RASA[7] Workflow scheduling which uses Batch mode as scheduling method. The make span parameter passed for this algorithm. It is used to reduce make span.

The other algorithm known as SHEFT algorithm[3] which uses dependency mode as scheduling method and concentrates on the scheduling parameters such as execution time and scalability. It is used for optimizing workflow execution time as well enables resources to scale elastically during workflow execution.

Although the above mentioned algorithms have benefits of some QoS, None of the algorithms are concentrated on the multiple parameters such as scheduling success rate, cost, time, make span, availability and

reliability. In MQMCE algorithm reliability and availability is added as extra parameters to fulfil customer’s multiple requirements. It is used to schedule the workflow dynamically and is used to reduce the execution time, cost as well as increase availability and reliability.

### III. ARCHITECTURE AND METHODOLOGY

#### 1) Architecture

In cloud servers are maintain by two level i) storage server ii) computational server. Storage server provides the function of storage and modification. In computational server provide the function of mapping the cloud services to user based requirements. Fig1. shows the architecture for overall function of MQMCE .

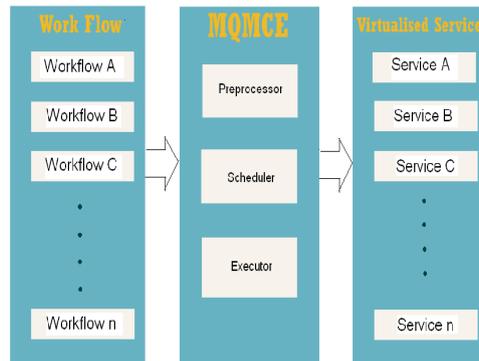


Fig 1. Overview of MQMCE

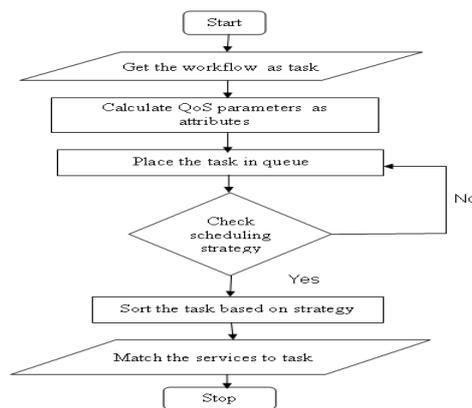


Fig2. MQMCE Scheduler Process

Number of workflows arrived from user and send to MQMCE scheduler. This performs all the process which is mentioned in Fig.2. Finally services are mapped to workflows based on three process i)pre-processor, ii)scheduler, iii)executer[5].In fig2 shows the scheduler process. First user requirements are arrived in the form of workflows, and then it is converted into tasks. These tasks are divided based on the QoS requirements. QoS requirements are calculated by preprocessing function using some formulas which is mentioned below. Second place the task in the queue and sorting the task based on scheduling strategy which is mentioned below. Finally match the best services for the task based on user QoS requirements such as cost, time, and reliability

#### 2) Methodology

Workflow scheduling discovers resources and allocates tasks on suitable resources. Proper scheduling can have significant impact on the performance of the system. Workflow  $W_i$  represented by 5 tuples  $W_i (T_{ij}, TQ_i, CQ_i, RLQ_i, AvK)$  where  $T_{ij}$  is number finite task.  $TQ_i$  is time quota for workflow.  $CQ_i$  is cost quota for workflow  $AvK$  is a overall availability of resources for user requirements.

$RLQ_i$  is reliability quota for Workflow. Let  $T$  is set of task and  $S$  set of available services, The MQMCE schedule the set of task by mapping task with suitable service by achieving of to reduce the turnaround time, reduce the waiting time ,increase response time ,increase resource utilization and reduce a cost as a single objective manner. Use some definitions for calculating QoS parameter which is used in the form of attributes in the algorithm. The time limit when the task is executed is calculated as below.

$$TQ(t_i) = \text{Min}(T(t_i, R)) + VT(t_i, R) * [5]$$

$$(QoS(\text{time}) - \text{Min}T(w)) / \sum VT(t, R)$$

Where  $\text{Min}(T(t_i, R))$  is the minimum time of the task  $t_i$  executed.  $VT(t_i, R)$  is the time variance which the task  $t_i$  executed on all services.  $QoS(\text{time})$  is the time attribute of QoS requirements.  $\text{Min}T(w)$  is the minimum time of the whole workflow executed,  $\sum VT(t, R)$  is the time variance sum of all tasks in the workflow. The cost limit when the task is executed calculated as below.

$$CQ(t_i) = \text{Min}(C(t_i, R)) + VC(t_i, R) * [5]$$

$$(QoS(\text{cost}) - \text{Min}C(w)) / \sum VC(t, R)$$

Where,  $\text{in}(C(t_i, R))$  is the minimum cost of the task  $t_i$  executed.  $VC(t_i, R)$  is the cost variance which the task  $t_i$  executed on all services.  $QoS(\text{cost})$  is the cost attribute of QoS requirements.  $\text{Min}C(w)$  is the minimum cost of the whole workflow executed,  $\sum VC(t, R)$  is the cost variance sum of all tasks in the workflow.

The reliability is calculated from different between the request per cloudlet and number of actual request from the user. The execution process is defined as below

$$RLQ(t_i) = m_s(t_i) / M_s(t_i)$$

$RLQ(t_i)$  is the reliability of the services  $S$ ,  $m_s(t_i)$  is the total number of tasks executed successfully on  $S$ ,  $M_s(t_i)$  is the total number of tasks executed on  $S$ .

The availability is calculated from first calculate the overall makespan then calculate the each task allocate for each resource then calculate the overall time taken for all the resources allocated for all the tasks. finally subtract the overall time of assignment resource to task from makespan, then calculate the overall availability of resource from different between makespan and product the results.

$$\text{Availability of a Resources } K : Av_K K := (\text{Makespan} - \sum \text{Time\_exe}(Ti / Rk)) / \text{Makespan}$$

The MQMCE scheduling strategy to sort all ready tasks based on below the consideration. i) The number of services in cloud is finite. And in most cases, the number of tasks waiting to be executed is larger than the service number. So the number of services in cloud is finite. And in most cases, the number of tasks waiting to be executed is larger than the service number. So a task with minimum available service number should be scheduled first. The reason is that the task would have not available services, if other tasks are scheduled first. ii) The tasks which belong to the workflow with minimum time surplus and cost surplus reliability surplus should be scheduled first. The reason is similar to the above. iii) The task with minimum covariance should be scheduled first. iv) The task with reliability executed by the sorting algorithm. It is considered by the reliability of the task. The minimum reliability will be scheduled first.

#### MQMCE Algorithm

There are two algorithms are used for mapping services to user requirements i) Task sort algorithm[5]. ii) Scheduling strategy algorithm[5].

##### i) Task sort algorithm

Assume the workflow as tasks  $T$  and resources as a services  $S$ . i) Get all the tasks and if any task is ready task in the all tasks and assume that the ready task as  $t$ . ii) sort the all ready task in the queue while any ready task in the queue then it is insert in the queue. Insert task  $t$  into queue  $q$  according to the strategy which is mentioned above methodology.  $\text{getReadyTasks}()$  The method is executed until all the set of ready tasks is in ready.  $\text{sort Task}()$  The method inserts the task until the set of ready tasks scheduled insert  $\text{Task}()$  The method used to insert the task into queue according the scheduling strategy

##### ii) Task scheduling algorithm:

```

schedule(q,S)
while
{ q is not empty
t←first task in q
s←get service(t,S)
schedule task t on service s
q=q-{t}
S=S-{s}
}end while
get Service(t,S)

```

select s.S, and the execution time, cost and reliability not exceed the time, cost, availability and reliability quota of task t return s to assign suitable services to task.

#### IV. EXPERIMENTAL RESULTS

The proposed MQMCE algorithm was run to evaluate its performances for various test cases with different workflows with different set of QoS parameters for each workflow. The cloud environment is simulated using cloudsim and communication between them is achieved by using java programming, Eclipse. Taking four workflows as a tasks and map the graphs for different user QoS parameter values and obtained the values of time cost and reliability of success rate from the fig.3, fig.4, fig.5, fig.6 respectively. From these graph MQMCE algorithm meet the QoS requirements of the workflow by achieving to reduce the cost and reduce the time and increase the reliability and availability when compared to user required values. And also compared with Round Robin algorithm, which is considering the Qos users time, cost , availability and reliability. Also it is compared with the MQMW algorithm which is considered the QoS parameters time and cost but not the availability and reliability. But MQMCE consider time, cost, availability and reliability. From this analysis, The MQMCE algorithm used to reduce the cost, to reduce the time and increase the reliability and availability which results better than Round Robin algorithm.

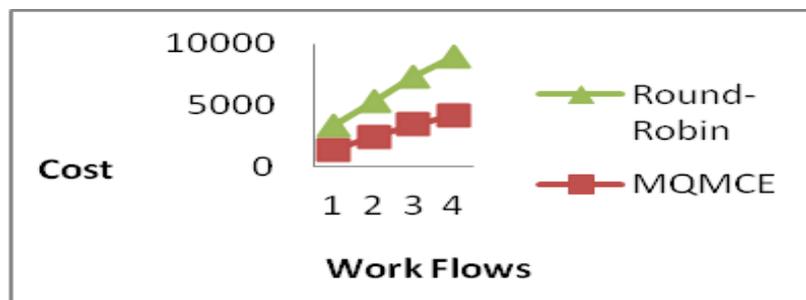


Fig3. MQMCE(Vs)RR cost

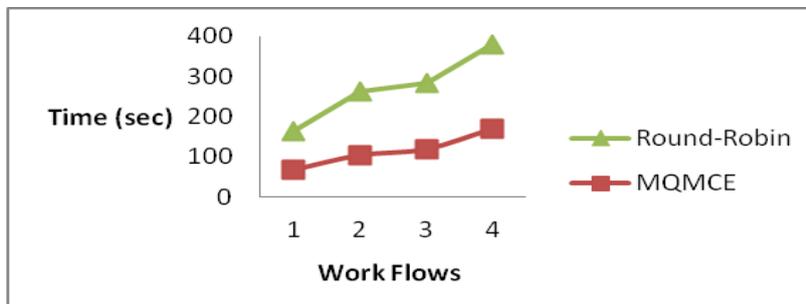


Fig4. MQMCE(Vs)RR Time

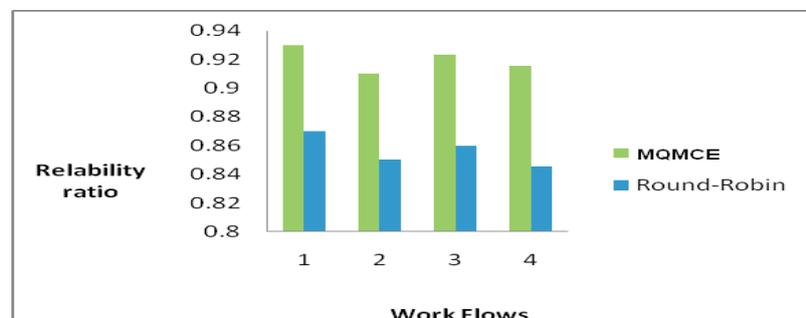


Fig 5. MQMCE(Vs)RR Reliability values.

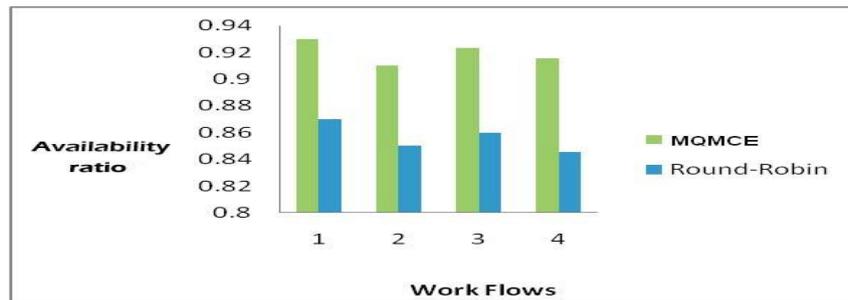


Fig 6. MQMCE(Vs)RR Availability values.

## V. CONCLUSION AND FUTURE WORK

The Cloud Computing led to the possibility to provide anything as a service over the Internet. It service for different users at the same time with different QoS requirements, the scheduling strategy should be developed for multiple workflows with different QoS requirements. The Multiple QoS Constrained Scheduling Strategy of Multi-Workflows (MQMC) attains the customer's multiple QOS constraints such as execution time, executions cost as well as schedule the workflows dynamically. In existing algorithms, they concentrate only on cost or time or both but not concentrated on reliability and availability. In this proposed algorithm satisfy the multiple QOS like time, cost as well as reliability and availability. This lead the scheduling is done by customer's multiple QOS requirements in a single objective manner. In future work combine MQMCE with MOGA for optimize the above QoS parameters highly based on more user requirements.

## REFERENCES

1. Mohsen Amini Salehi, Bahman javadi and Rajkumar Buyya, "QoS and pre-emption aware scheduling in federated and virtualized Grid Computing Environments, Elsevier July 2012"
2. Cui Lin, Shiyong Lu, "Scheduling Scientific Workflows Elastically for Cloud Computing" in IEEE 4th International Conference on Cloud Computing, 2011.
3. K. Liu; Y. Yang; J. Chen, X. Liu; D. Yuan; H. Jin, " A Compromised-Time- Cost Scheduling Algorithm" in SwinDeW-C for Instance-intensive Cost-Constrained Workflows on Cloud Computing Platform, International Journal of High Performance Computing Applications, vol.24 no.4 445-456, May, 2010.
4. Mrs.S.Selvarani I; Dr.G.Sudha Sadhasivam, Improved cost-based algorithm for task scheduling in Cloud computing, IEEE 2010.
5. Ming Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi, "A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud computing, IEEE, April 2009.
6. Saeed Parsa and Reza Entezari-Maleki, " RASA: A New Task Scheduling Algorithm in Grid Environment" in World Applied Sciences Journal 7 (Special Issue of Computer & IT): 152-160, 2009. Berry M. W., Dumais S. T., O'Brien G. W. Using linear algebra for intelligent information retrieval, SIAM Review, 1995, 37, pp. 573-595.
7. Rajkumar Buyya, CheeShinYeo, Srikumar Venugopal, JamesBroberg, and Ivona Brandic. "Cloud Computing forms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility", Future Generation Computer Systems, Elsevier Science, Amsterdam, June 2009, Volume 25, Number 6, pp.599-616.
8. H. Zhao and R. Sakellariou, "Scheduling multiple dags onto heterogeneous systems," in Proceedings of the 15<sup>th</sup> Heterogeneous Computing Workshop (HCW), Rhodes Island, Greece, April 2006.
9. Jia Yu and Rajkumar Buyya, "A Budget Constrained Scheduling of Workflow Applications on Utility Grids using Genetic Algorithms", Proceedings of the 15th IEEE International Symposium on High-performance Distributed Computing (HPDC 2006), IEEE CS Press, Los Alamitos, CA, USA, June 19-23, 2006, Paris, France.
10. J. Blythe et al., "Task scheduling strategies for workflow-based applications in grids," in Proceedings of the IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05), Cardiff, UK, 2005.
11. A. Mandal et al., "Scheduling strategies for mapping application workflows onto the grid, "in Proceeding of the 14th International Symposium on High Performance Distributed Computing (HPDC'05). IEEE, 2005, pp. 125-134.
12. Jia Yu, Rajkumar Buyya and Chen Khong Tham, "Cost-based Scheduling of Scientific Workflow Applications on Utility Grids", In 1st IEEE International Conference on e-Science and Grid Computing, Melbourne, Australia, Dec. 5-8, 2005.