International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 04 | April – 2025 DOI: 10.55041/ISJEM02763
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

ENHANCING THE EFFECTIVENESS OF TASK SCHEDULING METHODS IN CLOUD ENVIRONMENTS

BABITA[1], DR. JUGNESH KUMAR[2]

[1]Research Scholar, FCEM College, Faridabad

[2]Proferssor,Department of Computer science & engineering, FCEM College, Faridabad

---

Abstract - Cloud computing combines the supremacy of parallel and distributed systems altogether. It is the first of its kind that delivered services (IaaS, PaaS and SaaS) on-demand. Its elasticity and scalability gains so much popularity worldwide, which is still one of the ongoing technologies. It charges in the form of utility computing based on a variety of pricing models: pay-per-use and reservation basis. Cloud is available in four basic forms that are: Public, Private, Community and Hybrid Cloud. Where, Amazon Web Services, Microsoft Azure, Google Cloud Platform are one of the top service providers. Still, research on different domains like: resource management, application modelling, Cloud security, interoperability, datacenters infrastructure and design are going on. Among them, resource management is one of the crucial and important areas where lots of research is still progressive. Resource management coordinates the resources in response to management actions. Due to the wide scalability, effective resource management becomes challenging now. It is the central part of Cloud environment consisting of four different phases that are: resource discovery, resource selection/allocation, task scheduling and resource monitoring.

Key Words: optics, Cloud Computing, Parallel Systems, Distributed Systems, On-Demand Services, Elasticity

1. INTRODUCTION

1.1 COMPUTING AND ITS EVOLUTION

The term "Computing" is defined as an activity that is responsible for executing a piece of code known as program. It required essential resources and communicates in a system to produce information intended for end user's. It has been evolved from supercomputing to distributed systems with the passage of time. The requirement and advancement in technologies given birth to following computing as discussed below:

1.1.1 Super Computing

Earlier system architecture was created with one goal that is to execute very large computational processes at much faster speed which gives the birth to super-computing. Super computer [4] is one that is capable of executing jobs which required huge processing power. The term "Super Computing" was used first time by New York City world broadsheet in 1929. Supercomputers were introduced in the 1960's. It was designed by Seymour Cray at CDC that later becomes Cray Research. The first product was CDC-6600 introduced in 1964. Then, researchers are improving gradually towards their processing power measured in petaFLOPS, as Supercomputer 2013 "Titan" built by Cray at Oak Ridge National Lab for use in different projects. It has processing speed 17.50 petaFLOPS with cost 97 million US $. Super computers are widely used in all kind of problems that required enormous amount of computing ranging from forecasting weather, stock market, metal strength, oil and gas explosion etc. to modeling several entities, nuclear study and almost all scientific and engineering problems. But its design issues like: one place computational hub for all the intended clients lead to issues especially remotely and its cost is still a barrier that gives the birth to other computing paradigms as discussed later. The latest one is being used widely from Kobe, Japan with name "Fujitsu Fugaku". 5

1.1.2 Parallel Computing

Another evolution in computing which gives birth to revolution in entire IT industry was the Parallel computing [5]. It was first introduced in year 1958 where first product official came in 1962 with name D825. It works as number of tasks from a given problem after decomposition based on graining and then they all are executed simultaneously. Parallel execution is based on the theory that outsized problems can frequently be separated into lesser modules as demand, that are next run at the same time as by identical yet capable resources. Parallel execution can be in the form of bit level to instruction level or data to task level. Even instruction pipelining is one of the true example that is inspired by parallel computing. In older days, computer programs have been designed and coded for serial computation but with the introduction of parallel computing they are replaced abruptly by parallel ones' where multiple tasks can be processed at the same time The paradigm become highlighted and a boon technology for rest of the world due to sudden advancement and increasing usage of multi-core processors along with Graphical Processing Units. GPU's efforts along with CPUs jointly amplify the throughput of data and the count of concurrent calculations within an application. Despite bundle of pros, it lacks in some department like: concurrency control from programmer's side, finding the right granularity:how much the fine grain required, scalability issues in hardware and software as well.

1.1.3 Distributed Computing

It came into existence at early 1960 with one of the big project that is: ARPANET. It is solely based on distributed system where multiple machines across the world are connected via

communication network through a message passing technique [1]. The ultimate idea behind distributed computing [7] was to accomplish a common goal utmost to facilitate resource sharing. The computing elements in a distributed system can be connected together as physically closed ones and either locally or geographically faraway using wide area network via web services [2]. It proves the several types of configuration that sometimes map with parallel computing and its paradigm. The aspiration of distributed computing is to build such a network that appears as a sole machine. The main thing behind such a computing is that they offer scalability and 6 redundancy in turn reliability actually which often lacks in centralized systems. Due to advancement in communication and inter-connecting devices, this architecture is quite popular and still in use that fulfils several types of problems from different domains. Its wide range of applications due to inherent characteristics makes a strong choice for its selection. On downside, it required efficient algorithms and programming techniques to distribute a problem as discussed in distributed system.

1.1.3.1 Characteristics of Distributed Computing

It is still one of the popular yet ongoing research areas because of its following inherent characteristics [1] [3] that are shown below: Resource Sharing: Resource sharing signifies the existing resources that are available across the world. In this, computers are having the ability to use any hardware devices like: disks and printers, software entities like: files and records or data anywhere in the system. Hardware resources are common for the purpose of reducing cost and ultimately for convenience. Where, data is shared for consistency and flow of information. Here, one node known as Resource Manager (RM) is used to control access of resources,concurrency among resources. This node is also responsible for giving naming scheme to identification purpose. Resource sharing model can be either Client-Server or Object-based. These models are used to describe following question like: how resources are provided, how they are used and how service provider and customer interact with each other.

 Scalability: Also known as expansion or growth. It concerned more regarding how the distributed system manages the expansion as the count of clients expands across the system. It is mostly done by adding machines in the network, in another way it is defined to accommodate new users without affecting the performance of existing system is called scalability. As the number of nodes increases,

components of the system should not require change. It means scalability does not affect the legacy and performance issues.

Transparency: It should be seen as a single virtual computer and any user can access resources transparently. In simpler words, transparency is defined as a 7 single system having single framework. Transparency is not limited to access but can be: “Access transparency”, the first type that acts as bridge between the demonstrations of the data. Where, “Location transparency” is answerable to hide the real location of the resource to end users that seems to users as local one. Another one is “Relocation transparency” that is liable to hide the progress of the resources as mobility to different location while in use. It deals with mobility of entities without affecting the access and overall working in the system. “Replication transparency” is the act of duplicating the identical resources without knowing to users which resource they are using. “Concurrency transparency” represents the sharing of a resource that may be shared by a few dedicated clients. Last is, “Failure transparency”. It is the act of masking failed or unavailable resources used by client and directing them to available resources.

LITERATURE SURVEY

Cloud computing, a catchphrase of today’s that mingle the power of both parallel and distributed computing together. Cloud environment deals with dissimilar kinds of virtualized resources that are scattered in heterogeneous locations. So, to allocate and schedule resources efficiently it requires noticeable efforts. Such efforts are to be made on different phases under resource management [22] [43] of Cloud are: resource discovery, resource selection, task scheduling and resource monitoring. Among them, scheduling is important phase of Cloud that plays significant role under resource management. It can be seen as the finding an optimal assignment of set of task(s) known as meta-task [31] over the pool of available resources to obtain desired goals. As it is the one of decisive phase that mapped tasks (also known as cloudlets) to assigned resources based on conditions and different environmental parameters of cloud that are imposed for better performance as whole. Lack of efficient task scheduling makes under-utilized resources which in turns leads to higher completion time, thus contribute towards poor system’s performance. That’s why; scheduling is always seen as the core phase of resource management.

2.1 COMPONENTS OF TASK SCHEDULING

During task scheduling the ultimate goal is to reduce time (makespan, completion time, response time etc.) depending upon the client’s requirement that contribute the most. It is explained as the maximum time spent by a task on mapped resource that includes machine’s availability time [43] and task’s execution time [44] [46]. Variation in machine’s availability time or ready time will lead to unbalancing of load which in turn reduces the average resource utilization rate. The overall performance of meta-task should be enhanced by reducing the task’s completion time. This will be achieved by ensuring that selected resources are always used without being idle. Figure 2.1 shows the vital components of scheduling and their interaction with following entities written below: 40 User: An agent who is the ultimate customer that sends requests in the form of task (cloudlet) as input to service provide via broker (as per transparency) to carry out his/her work. This entity does not know the location and data services as per principle of transparency. It only involved in sending and receiving information for the purpose of accomplishment of task. Client can instruct various conditions as objective function to CSP. Such preconditioning leads to huge requirement of

2.2. objective based scheduling.

Broker: It acts as an intermediater that interacts with list of authorized and registered Cloud service provider’s and customers. Every incoming request comes first from client’s

side with dynamic requirements. It contains three essential components to complete the job scheduling process as shown below:

Scheduler: It is responsible for scheduling tasks to resources based on scheduling technique. It collects tasks known as meta-task and form a batch with predefined criteria based on optimization and then schedule the same using dispatcher. In case of online scheduling it is free to form any kind of batch and directly forward the task to provider.

 Dispatcher: Its key task is to dispatch the batch of user's tasks of a job to respective site's having machines using schedule decided by scheduler and after getting results back from service provider(s).

 Collector: When scheduled user's task is completed their respective job then collector will collect results. Now it depends on whether merging of result is required or standalone is enough for dispatcher to forward the result obtained after scheduling.

 Resource Pool: It comprises a bunch of machines equipped with processing elements and other vital components that are available for solving user's task according to scheduling policy decided by scheduler and dispatched by dispatcher. At this juncture each site indicates some location that contains a set of machines under its autonomy. Resources in Cloud can be from single organization or multiple in case they needed. Famous CSP are: Amazon Web Services, Microsoft Azure, Google Cloud, Salesforce, IBM and Alibaba Cloud.

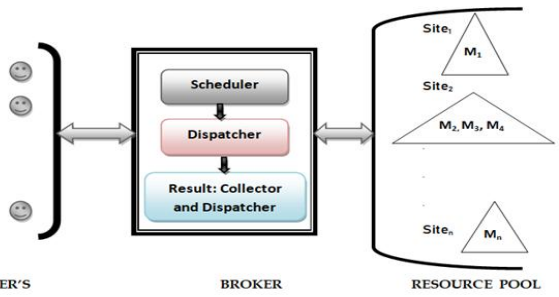

Figure 2.1: Components involvement during Task Scheduling

2.2.1 Taxonomy of Task Scheduling

Taxonomy is useful to classify certain types [46] and their immediate child under it to depict the scenarios in general. So, we are presenting taxonomy for Cloud computing environment under distributed system. Each category of scheduling helps the broker to choose the best and optimum scheduling policy for clients and CSP. The requirements to be met are dynamic and multi objectives are one of the tedious task. Independent and Cooperative scheduling techniques: The first category of scheduling technique is based on task dependency. A task that does not require any communication or dependency with other tasks is called independent task, whereas dependent tasks are different as they have order of precedence to be maintained while scheduling is going on. They are also known as cooperative tasks and can be shown by Directed Acyclic Graph. DAG can be further represented by three types like: Sequence, Parallel and Choice. In Sequence DAG, a chain of tasks in order. Here, a task will start after a previous task has completed. Parallelism denotes tasks that are running all together. In choice control, a task executed dynamically when it's associated evaluator criteria turns out to be true.

2.3. COMPARATIVE PERFORMANCE ANALYSIS OF INDEPENDENT TASK SCHEDULING HEURISTICS

Allocating user's cloudlet under heterogeneous environment like Cloud is a tiresome process. Therefore, researchers around the world have given birth to several algorithms from heuristic and meta-heuristic categories. Such techniques are proposed by keeping different performance parameters in mind. Different metrics are used for evaluating the efficiency of scheduling algorithms. In this chapter we have compared eleven scheduling algorithms based on parameters like: resource utilization rate, makespan, waiting time, fitness; flow time and machines variation as performance indicators. These metrics are simulated on three datasets that are: Standard using Braun twelve instances, Real dataset using Google work actual traces and artificial dataset to yield their performance in heterogeneous scenarios. These mapping techniques fall into a common category of heuristic mapping which is the fast in comparison to meta-heuristics approach. Under heuristic, we have gone through with two fundamental modes that are: online (immediate) and offline (batch) mode mapping where detailed analysis of their performance on above said parameters have been achieved.

3.1 TASK SCHEDULING

The business growth [100] of new era with Cloud computing is expanding rapidly within few years. The reason for gaining widespread popularity is its prominent delivery model and services. It includes broadly three types of services that are: IaaS, PaaS and SaaS [101]. Before its emergence, to set-up huge infrastructure was a human hectic process. That required ample cost to set-up such agargantuan infrastructure of resources. But, its subsistence changes the definition from solving all such challenges by aggregating diversified resources around the globe as single point of service. Apart from its service delivery models, it's their phases, or we call stages which removes such barriers on the fly. It includes: resource discovery, selection and allocation of task for completion of jobs. Here, resource discovery is the first stage that searches list of available yet authentic resources without any conditions and impositions. Next one is resource selection that actually filtered out resources after their successful discovery. This filtration may involve stakeholders (end user, broker and SLA) requirements and constraints as well. Among them, task allocation or scheduling is treated as one the core phase [43] that plays a significant role in its true success. Where, an efficient scheduling algorithm is one that fulfils requirements from variety of users by keeping certain criteria's as well. Scheduling always earns a decisive role that entertains multipurpose constraints altogether. To run the system smooth, towards high performance, efficient technique(s) try to reduce certain time limits such as: task's completion time or makespan, waiting time, flow time and turnaround time etc. Other metrics that contribute in overall performance are: resource utilization rate, fitness, deviation rate among machines etc. But there is always a battle of reaching the

optimum point of dispatching cloudlets. It gives scope for further refinements to existing scheduling techniques. Even, it may yield the birth to latest yet efficient, multi-objectives [104] scheduling techniques. To do so, we need to acquire the properties and intrinsic nature of scheduling techniques. It requires a comparative analysis based on authentic metrics that can act as performance indicators to verify their efficacy. So, in this paper, authors focused on eleven heuristics scheduling techniques from both modes. Where, to evaluate performance, six metrics have been as used as benchmark during simulation. Those are: makespan, average resource utilization rate [99], flow time, average waiting time and fitness along with machine's standard deviation. Simulation is done on three input datasets that are: standard, real and artificial.

3.2 HEURISTIC BASED MAPPING TECHNIQUES

We have shortlisted eleven scheduling techniques for comparison to fetch out performance analysis. All falls into a common category that is heuristic mapping that is fastest in comparison to meta-heuristics and yet efficient scheduling approach. Under heuristics, we have gone through with two fundamentals modes that are: online (immediate) and offline (batch) mode mapping. Below table shows nature, dispatching category, parameters achieved along with their intrinsic nature. Scheduling algorithms are in chronological order of their proposed time that starts with FCFS and end up with stable and efficient technique that is TASA.

3.3 PERFORMANCE METRICS USED

To evaluate the performance of eleven scheduling under heuristic mapping, we have selected six metrics as performance indicators. These metrics are used to verify the inherent performance based on certain scenarios. Here, average resource utilization rate (ARUR) is used to check the status of occupied processing elements throughout dispatching cloudlets. Where, makespan is such an important metric that denotes the latest time of last dispatched task. Where, average waiting time (AWT) represents elapsed time to get the machine on average basis. Flow time denotes overall time of every task on mapped machines. This metric is used to represent overall machines contribution to verify QoS. Where, fitness represents a balanced value of makespan and flow time using a weighted value. At last, standard deviation is used to ensure variation among machines to verify their difference of their contribution. Except ARUR, a smaller value of rest of the five metrics denotes high efficacy.

3.4 DATASETS USED FOR SIMULATION

A good dataset mimic real environment that helps to match with expected outcomes in field based on actual results from simulation. For the purpose of testing above scheduling techniques, we have selected three different data sets. They include user's task in range 5 to 3125 with different instruction size. Where, 2 to 32 diversified virtual machines are used for simulation under testing environment. As mentioned in our previous work task's and VM's are generated using complete and consistent types of ETC matrices. We have studies and found seven types of data sets that are useful while measuring the performance of scheduling techniques.

4.1. I-MAXSTD: EFFICIENT BATCH MODE MAPPING FOR CLOUD COMPUTING ENVIRONMENT

Cloud computing can be seen as a service in foreground and a big infrastructure or pool of resources available to cater different services in the background. It offers various services including IaaS, PaaS and SaaS. During different phase of Cloud computing including resource discovery, resource selection and task scheduling. Task scheduling can be viewed as most difficult activity. It isthe one of decisive phase as scheduling mapped tasks to assigned resources based on conditions and different environmental parameters of cloud that are imposed for better performance as whole. It scheduled tasks to the list of available, authentic and assigned resources based on requirement(s). In this chapter, we have proposed an algorithm named: Improved-MaxStd (I-MaxStd) that refines the mapping process of conventional MaxStd. It yields an efficient output in terms of reduction in makespan and better resource average utilization rate without compromising its legacy. The validation of proposed work has been done for heterogeneous types of ETC matrices being used as dataset.

4. INTRODUCTION TO I-MAXSTD

We got motivation from one of the famous yet efficient scheduling technique under the batch mode heuristic that is, MaxStd [107]. This algorithm gives optimal outcomes in terms of compact makespan and enhanced average resource utilization rate during simulation results executed on HBIT-SIM as discussed in the Chapter-3. This scheduling is entirely based on one concept that is, standard deviation. It dispatched tasks to respective resource based on high value of standard deviation as a prime technique. Then the process will be iterated until all tasks from meta-task are assigned to allocated machines. So, the task which got maximum value of standard deviation on machine with minimum expected execution time will be scheduled first. For the very first time in year 2008, MaxStd use the Coefficient of Variance (CoV) ETC matrix for mapping tasks to machines. The reason behind choosing CoV type ETC was tighter control over tasks and 110 machines heterogeneity that makes it one of stable task scheduling technique. Experimental results are proven to be better in terms of less makespan and better resource utilization rate when compared with other heuristic techniques. MaxStd mapping heuristic is still one of the striking techniques among variety of batch mode approaches because it uses the concept of standard deviation for every task among assigned resources along with Coefficient of Variance (CoV) as ETC matrix. Here CoV is used as it considers the heterogeneity of tasks and resources to give better results. After analysis of MaxStd thoroughly, then we found that this works exceptionally well for CoV based ETC matrix only. Where, it behaves differently on other sort of ETC matrices which are used in to test real-life scenarios. This further boosts us to investigate other types of available ETC matrices for scheduling under Cloud used as input datasets. This gives us a scope of refinement in the standard MaxStd for better outcomes and is also a reason to propose an improved version of MaxStd which we named: I-MaxStd. In this chapter, we are presenting an I-MaxStd (Improved-MaxStd) that refines the mapping process of conventional MaxStd to yields an efficient output in terms of reduction in makespan and better

resource average utilization rate without compromising its legacy. The validation of proposed work has been done for heterogeneous types of ETC matrices being used is discussed next.

4.2. EFFICIENT TASK SCHEDULING UNDER BATCH MODE HEURISTIC IN CLOUD ENVIRONMENT

From solving complex computational problems to huge data intensive tasks, Cloud Computing is emerged as one the growing technology. It is not restricted to hand out gigantic problems but also gratifying day-to-day clients with variable requirements. To accomplish such a variety of users, Cloud always required an efficient task scheduling to cater such dynamism. So, this chapter presents a scheduling technique which we named: Efficient Task Scheduling under Batch Mode Heuristic (ETSBMH) that mapped cloudlets efficiently to reduce the makespan along with a new type of metric that depicts variation among heterogeneous virtual machines completion time. We named this new performance metric as: Machine Makespan Aware Completion time Variation (MMACV) that considered completion time variation among machines with respect to makespan which is extremely useful to find the best performer. Results demonstrate the effectiveness of ETSBMH (proposed) algorithm from existing scheduling approaches in terms of reduced makespan, less MMACV and improved average machine utilization rate.

5.1 BATCH MODE HEURISTIC MAPPING

Since Cloud computing emergence in market, its popularity graph is elevated due to consistent yet prominent delivery of various services like: IaaS, PaaS and SaaS [21]. Years ago, it was either tiresome to implement or required well enough investment to build such a big infrastructure of resources. But, its existence changes the definition of complex arrangements and monetary cost to handle all by combining variety of resources around the world as single giant pool of resources. It can be seen as amalgamation of heterogeneous technologies broadly came from parallel and distributed system. To build and manage such a huge network it always required an efficient task scheduling technique(s) to cater the requirements from variety of customers. Not only scheduling, there is an absolute need of efficacy at every phase from resource discovery to resource allocation and ultimately assignment of those resources in an optimum way. Well, there is always a race of reaching the optimum level of assignment that gives scope to devise efficient and multi objective scheduling techniques. Scheduling is plays a decisive role that entertain multipurpose constraints from clients and provider's both altogether. To run the system smooth, towards high performance, efficient technique try to reduce task's completion time or makespan in case of batch. Another parameter that contributes in overall performance that every provider wants to increase is resource utilization rate. As there are heterogeneous machines are available for task(s) which may sometime imbalance of load due to non-optimum assignment reduces the machine load index. Here we are presenting an efficient task scheduling technique that considered dual objectives with reducing makespan and increase machine utilization rate by applying relevant scoring mechanism.

5.2 PROBLEM DEFINITION

A number of scheduling techniques under heuristic category have been proposed on dual objectives: from minimizing makespan to increasing machine's load for batch mode mapping. Almost every technique performed impressive in favorable cases but in rest of cases it might lose its consistency due to heterogeneous environment. We got motivated by some existing famous yet fundamental techniques like: Min-Min, Max-Min, Suffrage, RASA, MaxStd and New Alg due to their low running complexity and inherent efficiency. All of them worked quite well on above discussed objectives in certain circumstances but lack in general. Min-Min [49] [66] performed well where number of smaller length tasks is in majority due to inherent selection criteria. In opposite, Max-Min [49] [66] outperformed when it accepts lengthy tasks executed in parallel with smaller ones. Sufferage [49] [66] on the other hand was the first who actually observed the expected completion time pattern. The criteria to select a task were based on most suffered value. RASA [67] was the first in heuristic category that uses the concept of hybridization. It picked Min-Min and Max-Min based on resource count. The cycle of both techniques were used to empower the best from each. Another approach that uses the application of statistics was MaxStd . It selects the task with most standard deviation. One more approach that uses the key properties from Sufferage technique was New Alg scheduling that considered minimum execution time as additional information for task selection. Almost every scheduling technique irrespective of category [66] applies MCT [49] for resource assignment to selected task. Here, we have proposed an efficient task mapping named it: Efficient Task Scheduling under Batch Mode Heuristic (ETSBMH). Next section discusses assumptions and objectives of proposed work along with its algorithms and outcomes.

5.3. QOS ENABLED TASK SCHEDULING APPROACH FOR CLOUD.

Due to constraints along with profit margins in background, service provider's sometime neglect to feed essential services to their respective clients. Such compulsion raises the demand for efficient task scheduling that can meet multiple objectives. But without any prior agreement, again makes a casual approach. So this dispute can be addressed when competent scheduling executes right over the Service Level Agreement. It acts as hotspots to define set of rules to assure quality of service. At this time, there is a huge demand of SLA opted scheduling that can produce profitable results from provider's and client's as well. This chapter presents an adaptive approach that can be applied to any existing independent scheduling techniques on the fly, named: Q followed by the applied scheduling technique. Result shows drastic improvement in terms of average waiting time (or response time due to non pre-emptive scheduling) and average turnaround time without comprising provider's cost margin at all along with maintaining fair policy in the system.

6.1 MOTIVATION BEHIND QOS

There are hundreds of scheduling techniques available, broadly split into two classes: heuristic and meta-heuristic mapping.

Majority of techniques focused on two factors only, makespan and resource utilization rate. Both metrics are from provider's point of view actually, where makespan is the largest availability time among assigned machines. Where, another parameter that takes care of load among resources is average resource utilization rate (ARUR) that defines time units used by each assigned resources to carry out the tasks. So, the objective of broker's site is to implement such a scheduling that outperforms in these departments mainly to attain the maximum profit. Despite such improvements in makespan and ARUR, broker site is still neglecting another side i.e., user's needs in terms providing basic Quality of Service (QoS) such as fast availability and less waiting time. QoS can be seen from multiple angles, where 138 from client's side it may includes: user's priority, task deadline, availability of resources, less waiting time and reliability etc. To such impositions, we have reviewed a variety of QoS enabled scheduling techniques under heuristic category where we found improvements in various QoS parameters by entertaining customer's basic needs of accepting priorities and meeting their deadlines. But, such improvements lack meet multi-objectives from clients and providers altogether. So, to address such challenges we can adopt Service Level Agreement (SLA) that provides quality assurance to clients and providers in the form of regulations and constraints during scheduling phase. So, today's where everything is dynamic, there is a huge demand of SLA based scheduling to cater quality of service. During literature survey, we have found one of the famous, oldest and efficient tasks scheduling under batch mode heuristic that is "Min-Min". This technique is unable to delivered QoS to its customers but have the ability to dispatch cloudlets having less completion time. It assigned tasks to their assigned resources in two phases that yields better result like: reduced makespan when compared with other existing approaches from same category. Each phase find out task with minimum completion time on available resources. Its schedule gives outstanding results when number of tasks with minimum execution time is higher than longer tasks (higher value of execution time).

6.2. COST AND TIME EFFICIENT HEURISTIC BASED IMMEDIATE TASK MAPPING

Cloud computing is first of its kind that delivered services on-demand and charge in the form of utility depending upon their consumption. Such elasticity and dynamic's gains popularity in Cloud so swiftly that catches the attention from worldwide. To serve clients better considering cost and quality at the same time with great competition among peers required efficient techniques. So, dispatching tasks to appropriate resource with constraints in background that can meet multiple objectives is always a hectic phase. That's why there is a need to embed optimization techniques to accomplish desired results. This chapter proposed a scheduling technique named: Cost and Time Efficient Task Scheduling (CTETS) that achieves noticeable saving in cost along with improvement gain in makespan for clients under immediate mode mapping. Proposed algorithm provides efficiency in cost and completion time both, although there is a tradeoff between them but still it gains both simultaneously as competing objectives. CTETS delivered outstanding results when compared with famous techniques from immediate mode. Even, it delivered stable yet prominent results in case of batch mode mapping during simulation.

7.1 MOTIVATION BEHIND PROPOSED TECHNIQUE

Cloud computing offers various services to clients such as: IaaS, PaaS, and SaaS in the form of utility computing . Among them, IaaS can be seen as one of the most prominent service that gives plenty of options to customers to utilize their underlying computational power that are scattered worldwide in the form of Web services. It offers CPU cycles ranging from low speed resources to high depending upon their availability and variety of pricing schemes where clients can formulate and solve their problems of variable length without any kind of infrastructure and special support. To do so, a site plays tremendous role for such involvements known as broker's site. It collects user's problems in the form of tasks and dispatched to available and assigned resources in such a way that meets multiple objectives. Broker scheduled client's tasks with the help of CIS that is Cloud Information System which supplies vital information about resources and traits of submitted tasks. So, objective of our broker's site is to implement such a scheduling technique that saves cost, gives quality to customers with keeping provider's profit on the same side. Scheduling in Cloud is broadly categorized into two modes [43] [66] depending upon the service that are immediate (online mode) and batch (offline mode). Every mode gives freedom to client's to consume such giant network that offers services and charged according depending upon variety of pricing models like: Amazon EC2offers resource instances on-demand ranging micro to large and Google Compute Engine-charge month wise depending upon assigned resources. During, literature work we found two major points, first one was trade-off between cost and time when cost efficient scheduling is being applied. Another point is lack of immediate mapping techniques that leads to cost effectiveness. So, in this research, proposed work studies how to allocate machines to tasks in immediate mode task mapping to reduce monetary costs along with time saving. We devised a cost and time efficient scheduling algorithm for heterogeneous machines and variable length tasks with linear pricing model based on pay-per-use. Proposed algorithm provides efficiency in completion time and costs despite their trade off as competing objective.

7.2 CTETS ALGORITHM (PROPOSED)

Proposed scheduling technique is devised to map the tasks to assigned machines under immediate mode mapping. It is also known as online mode scheduling as a task is dispatched immediately without forming a group of later tasks unlike batch mode mapping. After techniques like: FCFS [106], MET [49], MCT [49] there is not much research is done especially in area of cost efficiency becomes our motivation of research. So, this approach considers two main objectives that are: makespan and cost under consideration with certain assumptions.Not more than one task will arrive simultaneously. There will be at least a fraction of time difference in their arrival rate in case of immediate mode.

 Expected Time to Compute (ETC) matrix is used as input, it comprises twodimensional forms with row representing tasks

expected execution time on columns denoting heterogeneous assigned machines. CTETS approach is inspired by MEFT (Modified Earliest Finish Time) which in turn inspired from HEFT. It calculates a MEFT value for each individual machine. It considered summation of three factors as: first is the price of assigned machine(s), next is, task's completion time and third one is average execution time of each task. MEFT was developed dedicatedly for workflow scheduling only as it uses some sort of ranking for precedence of tasks. Our approach is specially devised for online task mapping due to its sequential ordering as discussed in assumption but not restricted to work on other categories of task scheduling available under Cloud. This technique uses a concept of machine's score that consist of only two parts unlike MEFT which includes additional third one as average task execution time.

8. CONCLUSION AND FUTURE SCOPE

Dispatching cloudlets under heterogeneous environment like Cloud is one of the core phases of Cloud systems. Therefore, researchers around the world have proposed several efficient algorithms from heuristic and meta-heuristic categories as well. Such scheduling techniques are proposed by keeping different performance parameters in their mind to achieve multi-objectives. So, keeping such views in mind, we have proposed and presented few scheduling techniques in this thesis.

8.1 CONCLUSION

Scheduling algorithms are simulated according to modules using programming languages like Core Java and Python. Where, HBITS-SIM own developed simulator in Java is used to evaluate the performance of exiting task scheduling techniques under heuristic approach. Following techniques have been proposed from immediate and batch mode mapping under heuristic mapping as:

 I-MaxStd: Efficient Batch Mode Mapping for Cloud: In this, we have proposed an efficient batch mode mapping technique from heuristic category of scheduling. Here, MaxStd is refined in such a way that further reduces makespan and improves the resource utilization rate. I-MaxStd (Proposed) algorithm is an improvement over standard MaxStd algorithm. It uses an addition phase where single task is rescheduled to the least loaded machine on the basis of makespan in its final phase. As it is devised to obtain dual objectives that are makespan and average resource utilization rate, it gives significant improvement in compact makespan in all the four scenarios from 7.14% to 18.4%. Where, a noteworthy gain of above 13% is achieved for ARUR. For various types of ETC matrices, IMaxStd performance will either outperform the standard MaxStd or behaves congruent to MaxStd. This improvement is without any degration in its performance as maintained its legacy.Efficient Task Scheduling under Batch Mode Heuristic (ETSBMH): We have proposed a heuristic technique named: ETSBMH for scheduling a batch of independent tasks along with proposed metric that is Machine Makespan Aware Completion time Variation (MMACV). Proposed approach outperformed with six famous heuristic mapping like: Min-Min, Max-Min, Sufferage, MaxStd, RASA and New Alg in terms of compact makespan as prime objective next to MMACV along with improved average machines load. During experiments ETSBMH achieved a noteworthy improvement in makespan from 1.27% to 41% in different workloads. It mapped tasks based on only two types of information that are: static (as priori) and dynamic (MCT) after each phase. Usage of only two types of input is one of the key features of ETSBMH. Here, compared techniques shares identical time complexity of O(nm) in the best case without considering precedence of tasks. Even, we have compared this technique with I-MaxStd (proposed). Where, we found some interesting facts as both scheduling techniques outperformed in their own input. But, when tested on standard input using Braun twelve instances, ETSBMH achieved significant improvement over I-MaxStd. QoS Enabled Task Scheduling Approach in Cloud computing as per SLA: Here, we have proposed an approach that drives QoS under batch mode mapping technique from heuristic category. Proposed approach denoted by 'Q' provides a new dimension to QoS by reducing average waiting time (AWT) and consequently average turnaround time (ATT) without affecting underlying arguments. The best part of proposed approach is of mapping.now users having lowered computational demands need not wait for longer period of time due to higher computational cloudlets which improves the overall system. Here, we are actually removing the convoy effect to give more availability to rest of user's that are waiting for their turn on same resource. Proposed technique does not alter the occupied machine of their respected tasks is 160 the limelight feature that leads to makespan and its average resource utilization rate unchanged. It gives guarantee without any loss from provider's perspective is the key of its success.

8.2 FUTURE SCOPE

Presented independent task scheduling techniques from heuristic mapping in this thesis are prominent and efficient. Here, entire contribution is focused on dispatching task with non-preemptive and non-precedence nature of scheduling. Thesis work achieved performance metrics like: reduced makespan, improved average resource utilization rate, compact waiting and turnaround time along with lower down cost as multi-objectives. This research work improves overall performance of Cloud environment and advances the state-of-the art through presented contribution. Still, there is always scope for further refinements and improvements to make the system more efficient. Further, it required refinements through additional testing by integrating more efficient approaches to meet the ultimate goal. Following points will depict future work as,Although, we have tested proposed schemes using simulator through standard and some real input datasets, still to verify the actual performance it required to map on real world scenarios on actual system.

 We have taken average of waiting time and turnaround time as top QoS parameters, where proposed scheme can further include other QoS parameters like: flow time, matching proximity, tardiness and security.

 We have proposed an algorithm for cost and time together for immediate mode exclusive but not limited to it. It can be further refined to deliver prominent results for batch mode mapping too. To test its robustness, it required equipping with fault tolerance mechanism in case of partially completed work,

so that a task can be resumed on other capable machine as an additional feature.

 We can also customize our research to collaborative environment, where different administrative domains have their full control over scheduling cloudlets in the future.

 Our entire research is focused on immediate (online) and batch mode (bag/offline) under heuristic category of task mapping. It reveals some interesting facts, but, further clubbing efficient algorithms from meta-heuristic as a hybrid approach needs to be tested. It may give some exciting outcomes to be investigated in the future.

REFERENCES

1. Lamport L. Time, clocks, and the ordering of events in a distributed system. InConcurrency: the Works of Leslie Lamport 2019 Oct 4 (pp. 179-196).

2. Xiaohui Z, Huayong W, Guiran C, Hong Z. An autonomous system-based distribution system for web search. In2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236) 2001 Oct 7 (Vol. 1, pp. 435-440). IEEE.

3. Nadiminti K, De Assunçao MD, Buyya R. Distributed systems and recent innovations: Challenges and benefits. InfoNet Magazine. 2006 Sep;16(3):1-5.

4. Cook JS, Gupta N. History of Supercomputing and Supercomputer Centers. InResearch and Applications in Global Supercomputing 2015 (pp. 33-55). IGI Global.

5. Navarro CA, Hitschfeld-Kahler N, Mateu L. A survey on parallel computing and its applications in data-parallel problems using GPU architectures. Communications in Computational Physics. 2014 Feb;15(2):285-329.

6. Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I. Spark: Cluster computing with working sets. HotCloud. 2010 Jun 22;10(10-10):95.

7. Franz J, Gerber M, Gruetzner M, Spruth W, inventors; International Business Machines Corp, assignee. Providing computing service to users in a heterogeneous distributed computing environment. United States patent US 8,140,371. 2012 Mar 20. 163

8. Anderson DP, Korpela E, Walton R. High-performance task distribution for volunteer computing. InFirst International Conference on e-Science and Grid Computing (e-Science'05) 2005 Jul 5 (pp. 8-pp). IEEE.

9. Motta G, Sfondrini N, Sacco D. Cloud computing: An architectural and technological overview. In2012 International Joint Conference on Service Sciences 2012 May 24 (pp. 23-27). IEEE.

10. Garrison G, Wakefield RL, Kim S. The effects of IT capabilities and delivery model on cloud computing success and firm performance for cloud supported processes and operations. International Journal of Information Management. 2015 Aug 1;35(4):377-93.

11. Marinos A, Briscoe G. Community cloud computing. InIEEE International Conference on Cloud Computing 2009 Dec 1 (pp. 472-484). Springer, Berlin, Heidelberg.

12. Satyanarayanan M. The emergence of edge computing. Computer. 2017 Jan 5;50(1):30-9.

13. Pan J, McElhannon J. Future edge cloud and edge computing for internet of things applications. IEEE Internet of Things Journal. 2017 Oct 30;5(1):439-49.

14. Bonomi F, Milito R, Zhu J, Addepalli S. Fog computing and its role in the internet of things. InProceedings of the first edition of the MCC workshop on Mobile cloud computing 2012 Aug 17 (pp. 13-16).

15. Stojmenovic I, Wen S. The fog computing paradigm: Scenarios and security issues. In2014 federated conference on computer science and information systems 2014 Sep 7 (pp. 1-8). IEEE. 164

16. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M. A view of cloud computing. Communications of the ACM. 2010 Apr 1;53(4):50-8.

17. https://www.inforisktoday.com/5-essential-characteristics-cloud-computing-a- 4189.

18. Gong C, Liu J, Zhang Q, Chen H, Gong Z. The characteristics of cloud computing. In2010 39th International Conference on Parallel Processing Workshops 2010 Sep 13 (pp. 275-279). IEEE.

19. Dillon T, Wu C, Chang E. Cloud computing: issues and challenges. In2010 24th IEEE international conference on advanced information networking and applications 2010 Apr 20 (pp. 27-33). IEEE.

20. Bohn RB, Messina J, Liu F, Tong J, Mao J. NIST cloud computing reference architecture. In 2011 IEEE World Congress on Services 2011 Jul 4 (pp. 594-596). IEEE.

21. Wei Y, Blake MB. Service-oriented computing and cloud computing: Challenges and opportunities. IEEE Internet Computing. 2010 Nov 1;14(6):72-5.

22. Mustafa S, Nazir B, Hayat A, Madani SA. Resource management in cloud computing: Taxonomy, prospects, and challenges. Computers & Electrical Engineering. 2015 Oct 1;47:186-203.

23. Endo PT, de Almeida Palhares AV, Pereira NN, Goncalves GE, Sadok D, Kelner J, Melander B, Mangs JE. Resource allocation for distributed cloud: concepts and research challenges. IEEE network. 2011 Jul 18;25(4):42-6. 165

24. Zarrin J, Aguiar RL, Barraca JP. Resource discovery for distributed computing systems: A comprehensive survey. Journal of parallel and distributed computing. 2018 Mar 1;113:127-66.

25. Houidi I, Louati W, Zeghlache D. A distributed virtual network mapping algorithm. In2008 IEEE International

Conference on Communications 2008 May 19 (pp. 5634-5640). IEEE.

26. Nassif LN, Nogueira JM, de Andrade FV. Resource selection in grid: a taxonomy and a new system based on decision theory, case-based reasoning, and fine-grain policies. Concurrency and Computation: Practice and Experience. 2009 Mar 10;21(3):337-55.

27. Gholami A, Arani MG. A trust model for resource selection in cloud computing environment. In2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI) 2015 Nov 5 (pp. 144-151). IEEE.

28. Shen W, Li Y, Ghenniwa H, Wang C. Adaptive negotiation for agent-based grid computing. Journal of the American Statistical Association. 2002 Jul 15;97(457):210-4.

29. Liu W, Shi F, Du W, Li H. A cost-aware resource selection for data intensive applications in cloud-oriented Data Centers. IJITCS. 2011 Aug;3(1):10-7.

30. Anitha N, Basu A. A dynamic resource allocation based on multi attributes scoring in collaborative cloud computing. Global Journal of Computer Science and Technology. 2015 Oct 5.

31. Elzeki OM, Reshad MZ, Elsoud MA. Improved max-min algorithm in cloud computing. International Journal of Computer Applications. 2012 Jan 1;50(12). 166

32. Christodoulopoulos K, Sourlas V, Mpakolas I, Varvarigos E. A comparison of centralized and distributed meta-scheduling architectures for computation and communication tasks in Grid networks. Computer Communications. 2009 May 28;32(7-10):1172-84.

33. Henzinger TA, Singh AV, Singh V, Wies T, Zufferey D. Static scheduling in clouds. memory. 2011 Jun 14;200(o1):i1.

34. Xhafa F, Abraham A. Computational models and heuristic methods for Grid scheduling problems. Future generation computer systems. 2010 Apr 1;26(4):608- 21.

35. Alshathri S. Contemporary Perception of Task Scheduling Techniques in Cloud: A Review. In2018 2nd European Conference on Electrical Engineering and Computer Science (EECS) 2018 Dec 20 (pp. 201-205). IEEE.

36. https://www.researchgate.net/post/What_are_the_differences_between_heuristics_a nd_metaheuristics

37. Annette J R, Banu W A, Shriram S. A taxonomy and survey of scheduling algorithms in cloud: based on task dependency. In IJCA 2013 Nov (Vol. 82, No. 15, pp. 20-26).

38. Mangla N, Singh M. Workflow Scheduling In Grid Environment. InIJERA, March, 2014.

39. Page AJ, Naughton TJ. Dynamic task scheduling using genetic algorithms for heterogeneous distributed computing. In19th IEEE international parallel and distributed processing symposium 2005 Apr 4 (pp. 8-pp).IEEE. 167

40. Chen H, Fu X, Tang Z, Zhu X. Resource monitoring and prediction in cloud computing environments. In 2015 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence 2015 Jul 12 (pp. 288-292). IEEE.

41. Sookhak M, Talebian H, Ahmed E, Gani A, Khan MK. A review on remote data

auditing in single cloud server: Taxonomy and open issues. Journal of Network and Computer Applications. 2014 Aug 1;43:121-41.

42. Espadas J, Molina A, Jiménez G, Molina M, Ramírez R, Concha D. A tenant

based resource allocation model for scaling Software-as-a-Service applications over cloud computing infrastructures. Future Generation Computer Systems. 2013 Jan 1;29(1):273-86.

43. Banga P, Rana S. Heuristic based independent task scheduling techniques in cloud computing: a review. Int. J. Comput. Appl. 2017 May;166(1):0975-8887.

44. Cui H, Liu X, Yu T, Zhang H, Fang Y, Xia Z. Cloud service scheduling algorithm research and optimization. Security and Communication Networks. 2017 Jan 1;2017.

45. Fang Y, Wang F, Ge J. A task scheduling algorithm based on load balancing in cloud computing. InInternational conference on web information systems and mining 2010 Oct 23 (pp. 271-277). Springer, Berlin, Heidelberg.

46. Mazumder AM, Uddin KA, Arbe N, Jahan L, Whaiduzzaman M. Dynamic task scheduling algorithms in cloud computing. In2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA) 2019 Jun 12 (pp. 1280-1286). IEEE. 168

47. Armstrong R, Hensgen D, Kidd T. The relative performance of various mapping algorithms is independent of sizable variances in run-time predictions. InProceedings Seventh Heterogeneous Computing Workshop (HCW'98) 1998 Mar 30 (pp. 79-87). IEEE.

48. Freund RF, Gherrity M, Ambrosius S, Campbell M, Halderman M, Hensgen D, Keith E, Kidd T, Kussow M, Lima JD, Mirabile F. Scheduling resources in multiuser, heterogeneous, computing environments with SmartNet. InProceedings Seventh Heterogeneous Computing Workshop (HCW'98) 1998 Mar 30 (pp. 184- 199). IEEE.

49. Maheswaran M, Ali S, Siegel HJ, Hensgen D, Freund RF. Dynamic mapping of a class of independent tasks onto heterogeneous computing systems. Journal of parallel and distributed computing. 1999 Nov 1;59(2):107-31.

50. Sharma G, Banga P. Classifier MCT for immediate mode independent task scheduling in Computational Grid. International Journal of Engineering Trends and Technology. 2013 June 1(4):2722-6.

51. Xu M, Cui L, Wang H, Bi Y. A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing. In2009 IEEE International Symposium on Parallel and Distributed Processing with Applications 2009 Aug 10 (pp. 629- 634). IEEE.

52. Chen H, Wang F, Helian N, Akanmu G. User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing. In2013 national conference on parallel computing technologies (PARCOMPTECH) 2013 Feb 21 (pp. 1-8). IEEE. 169

53. He X, Sun X, Von Laszewski G. QoS guided min-min heuristic for grid task scheduling. Journal of Computer Science and Technology. 2003 Jul 1;18(4):442- 51.

54. Munir EU, Li J, Shi S. QoS sufferage heuristic for independent task scheduling in grid. Information Technology Journal. 2007 Aug;6(8):1166-70.

55. Huang QY, Huang TL. An optimistic job scheduling strategy based on QoS for cloud computing. In2010 International Conference on Intelligent Computing and Integrated Systems 2010 Oct 22 (pp. 673-675). IEEE.

56. Suresh P, Balasubramanie P. User demand aware grid scheduling model with hierarchical load balancing. Mathematical Problems in Engineering. 2013 Jan 1;2013.

57. Thomas A, Krishnalal G, Raj VJ. Credit based scheduling algorithm in cloud computing environment. Procedia Computer Science. 2015 Jan 1;46:913-20.

58. Ali HG, Saroit IA, Kotb AM. Grouped tasks scheduling algorithm based on QoS in cloud computing network. Egyptian informatics journal. 2017 Mar 1;18(1):11- 9.

59. Nasr AA, El-Bahnasawy NA, Attiya G, El-Sayed A. A new online scheduling approach for enhancing QOS in cloud. Future Computing and Informatics Journal. 2018 Dec 1;3(2):424-35.

60. Barani, R. and Suguna, S. Deadline Aware Prioritized Task Scheduling Algorithm in Cloud Computing. Int. J. of Recent Tech. & Engg. 2019 8(2S11), pp.815-818.

61. Hussain A, Aleem M, Iqbal MA, Islam MA. SLA-RALBA: cost-efficient and resource-aware load balancing algorithm for cloud computing. The Journal of Supercomputing. 2019 Oct 1;75(10):6777-803. 170

62. Dong F, Luo J, Gao L, Ge L. A grid task scheduling algorithm based on QoS priority grouping. In2006 Fifth International Conference on Grid and Cooperative Computing (GCC'06) 2006 Oct 21 (pp. 58-61). IEEE.

63. Keat NW, Fong AT, Chaw LT, Sun LC. Scheduling framework for bandwidthaware job grouping-based scheduling in grid computing. Malaysian Journal of Computer Science. 2006 Dec 1;19(2):117-26.

64. Suresh P, Balasubramanie P. Grouping based user demand aware job scheduling approach for computational grid. International Journal of Engineering Science and Technology. 2012 Dec;4(12):4922-8.

65. Sharma A, Sharma S. Credit based scheduling using deadline in cloud computing environment. Int. J. Innov. Res. Comput. Commun. Eng.(IJIRCCE). 2016;4(2).

66. Braun TD, Siegel HJ, Beck N, Bölöni LL, Maheswaran M, Reuther AI, Robertson JP, Theys MD, Yao B, Hensgen D, Freund RF. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. Journal of Parallel and Distributed computing. 2001 Jun 1;61(6):810-37.

67. Saeed P, Entezari-Maleki R. RASA: A new task scheduling algorithm in grid environment. World Applied Sciences Journal of Special Issue of Computer and IT. 2009:152-60.

68. Sharma G, Banga P. Task aware switcher scheduling for batch mode mapping in computational grid environment. International Journal of Advanced Research in Computer Science and Software Engineering. 2013 Jun;3. 171