STUDY AND IMPLEMENTATION OF RESOURCE ALLOCATION ALGORITHMS IN CLOUD COMPUTING

PROJECT REPORT

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

Computer Science & Engineering

By

ANMOL RANA (121315)

ANKUSH SHARMA (121327)

Under the Supervision of

Dr. Hemraj Saini

to

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"STUDY AND IMPLEMENTATION OF RESOURCE ALLOCATION ALGORITHMS IN CLOUD COMPUTING"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2016 to June 2016 under the supervision of **Dr. Hemraj Saini** Assistant Professor (Senior Grade).
The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Anmol Rana 121315
Ankush Sharma 121327

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Hemraj Saini
Assistant Proffesor (Senior Grade)
Computer Science and Engineering
Dated:

ACKNOWLEDGEMENT

This project work would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost I offer my sincerest gratitude to my supervisor **Dr. Hemraj Saini** who has supported me throughout my project work with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my Masters degree to her encouragement and effort and without him this thesis, too, would not have been completed or written. One simply could not wish for a better or friendlier supervisor.

The work was performed at Department of Computer Science and Engineering at the University, and I would like to thank all the people there for their hospitality and support.

Last, but not the least, my family and the one above all of us, the omnipresent God.

Date:

ABSTRACT

Cloud computing is an attractive computing model since it allows for the provision of resources on-demand. Such a process of allocation and reallocation of resources is the key to accommodating unpredictable demands and improving the return on investment from the infrastructure supporting the Cloud. However, despite the recent growth of the Cloud Computing market, several problems with the process of resource allocation remain unaddressed. Resource allocation is an integral, evolving part of many data center management problems such as virtual machine placement in data centers, network virtualization, and multi-path network routing. Since the problems are inherently NP-Hard, most existing systems use custom-designed heuristics to find a suitable solution. However, such heuristics are often rigid, making it difficult to extend them as requirements change. In this work, the cloud based resource allocation approach is implemented with the effectiveness of the resources. The work is done on cloudsim that is Java based platform for cloud computing.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

Chapter-1

INTRODUCTION

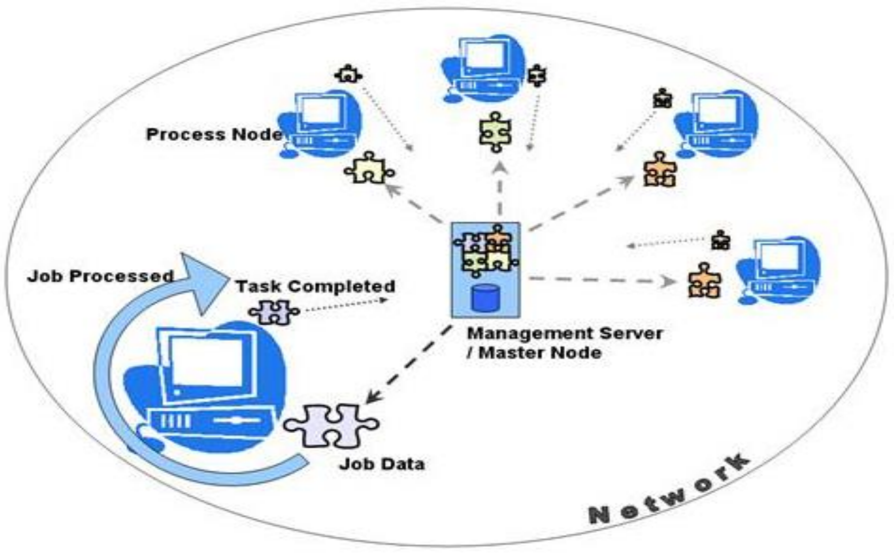
Cloud computing is a unique and effective way to use the computing infrastructure for effective communication and less overhead. The major technology behind grid as well as cloud computing is the virtualization technology. Using virtualization technology, the number or arbitrary and remote processors works in parallel for the execution of tasks in such a way that there is less complexity and higher integrity of the processes.

In cloud and grid architecture, there are number of remote computing devices and processors which work and execute the processes in parallel so that there is minimum load on the single node or machine. By this way, the overall load is balanced.

There are number of research areas in the domain of grid computing and cloud computing, still few areas are still in the preferences of the research community as these areas needs updations and enhancements in regular basis.

The major areas of research in grid and cloud architecture are –

- Load Balancing
- Power Optimization
- Energy Optimization
- Secured Routing
- Task Scheduling
- Resource Allocation and Optimization
- Confidentiality and Integrity Management
- Public and Private Key Cryptography ….. and many others



- **Figure 1.1 - Cloud Computing Architecture and Working Model [1]**

1.1 VIRTUALIZATION TECHNOLOGY

Virtualization technology is back bone in the grid and cloud platforms for effective management of the systems on demand for the jobs under execution. Using virtualization, there is no need of deployment of the dedicated systems or no0064es, rather the virtual machines take charge of all the operating system, platform and the resources to be used.

Virtual machines are classified into the two major taxonomies which depend on their usage and the degree of communication -

- **System Virtual Machine** – This system provides the full system environment to support the processes and execution of operating system in use.
- **Process Virtual Machine.** It is also called as Language Virtual Machine**.** Such machine is deployed and designed to process a single task or job or simply program. It means that it is dedicated for a single process.

Key Virtualization Technology Based Software Suite in the Corporate World

- **Windows as Host OS**
  - VMWare Workstation (Any Guest OS)
  - Virtual Box (Any Guest OS)
  - Hyper-V (Any Guest OS)

- **Linux as Host OS**
  - VMWare Workstation
  - Microsoft Virtual PC
  - VMLite Workstation
  - VirtualBox
  - Xen

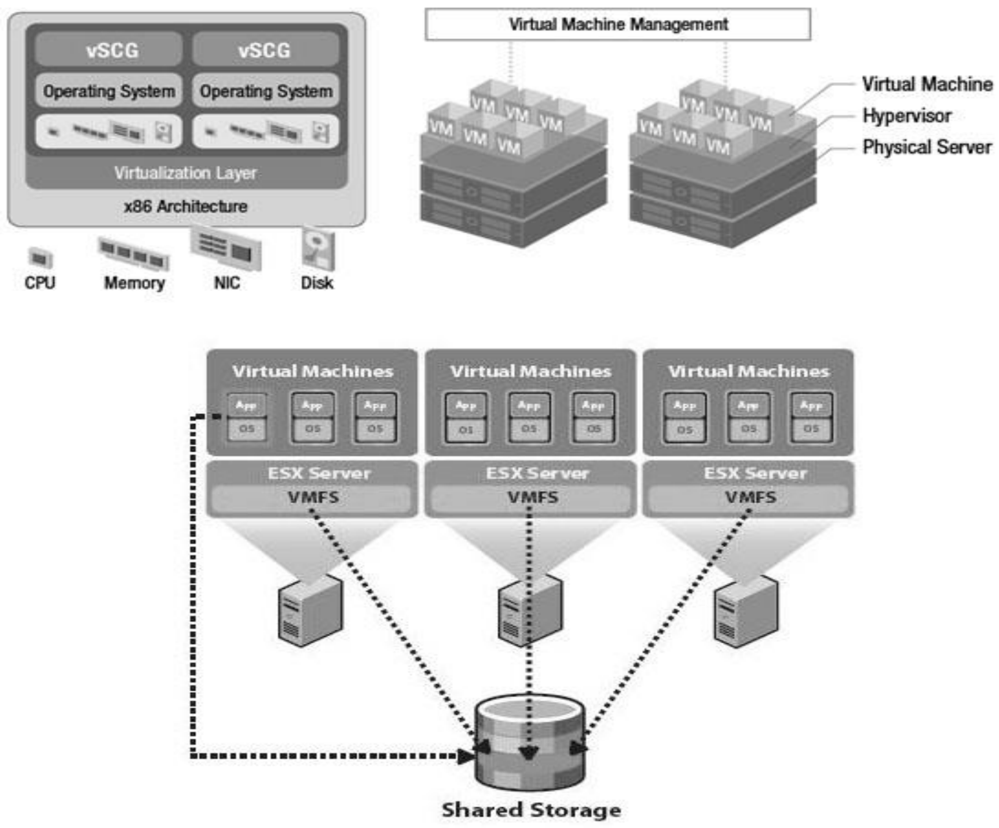


Fig. 1.2 - Virtualization Technology [2]

Hypervisor Type 1 Software Suite

- VMWare ESXi
- Citrix Xen
- KVM (Kernal Virtual Machine)
- Hyper-V

Hypervisor Type 2 Software Suite

- VMware Workstation
- VirtualBox

Table 1.1 - Hypervisors Used in Industry

Hypervisor	**Cloud Service Provider**
Xen	Amazon EC2 IBM SoftLayer Fujitsu Global Cloud Platform Linode OrionVM
ESXi	VMWare Cloud
KVM	Red Hat HP Del Rackspace
Hyper-V	Microsoft Azure

This research work is having the focus on development as well as implementation of a job scheduling algorithm for effective scheduling and prioritization. By this way and the proposed technique, the effective results in terms of less cost, higher performance and less execution time is obtained. In this research work, the authors have analyzed the performance of assorted algorithms based on new parameters and dimensions.

There is need to analyze the performance of job scheduling algorithms on cloud as well as grid environment as priority, load balancing and related aspects can affect the performance of cloud and grid to a huge extent.

The work done in this research includes to propose the implementation of algorithms on new parameters so that the hidden aspects and attributes of the algorithms can be analyzed. The work includes to perform the implementation / simulation and analysis on algorithms on new parameters beyond the work in the base literature.

1.2 Virtualization Technology and Cloud

Virtualization is the major technology that works with cloud computing. Actual cloud is implemented with the use of virtualization technology. "In Cloud computing, the dynamic virtual machines are created to provide the access of actual infrastructure to the end user or developer at other remote location.

A virtual machine or just VM is the item use of any enlisting device or machine or PC that executes the game plan of rules or undertakings as a physical (honest to goodness) machine. Right when a customer or originator manages a virtual machine, the advantages including all tasks presented on the remote machine are accessible using a specific plan of traditions. Here, for the end customer of the cloud advantage, the virtual machine acts like the authentic machine."

The term VM was at first proposed and described by Popek and Goldberg as "a viable, withdrew duplicate of a bona fide machine".

Virtual machines are perceived into two foremost courses of action, dependent upon their usage and level of correspondence to any bona fide machine:

System Virtual Machine – System Virtual Machine gives a complete structure organize that sponsorships the execution of a complete working structure (OS). These duplicate a present structure building plan, and are made with the inspiration driving either giving a stage to run programs where the veritable gear is not open for use or of having different instances of virtual machines inciting more gainful use of enrolling resources, both to the extent imperativeness usage and cost suitability (known as hardware virtualization, the route to a circulated registering environment), or both.

Process Virtual Machine (Language Virtual Machine) – This kind of virtual machine is proposed to execute a singular task which infers that it reinforces a single methodology. Such virtual machines are solidly suited to one or all the all the more programming lingos and fabricated with the inspiration driving giving task conveyability and versatility (amongst diverse things). A vital typical for a virtual machine is that the item running inside is limited to the benefits and reflections gave by the virtual machine - it can't piece or break out its virtual environment.
A hypervisor or virtual machine screen (VMM) is a touch of PC programming, firmware or gear that makes and runs virtual machines. A PC on which a hypervisor is running one or more virtual machines is portrayed as a host machine. Each virtual machine is known as a guest machine. The hypervisor presents the guest working systems with a virtual working stage and manages the execution of the guest working structures. Different cases of a grouping of working systems may share the virtualized

In the implementation and deployment of the cloud service, type 1 hypervisors are used. Hypervisors of type 1 are associated with the concept of bare metal installation. It means there is no need of any host operating system to install the hypervisor.

By this technology, there is no risk of getting the host operating system corrupt. These hypervisors are directly installed on the hardware without need of any other operating system. On this hypervisor, multiple virtual machines are created.

A Type-1 hypervisor is a sort of customer hypervisor that interfaces plainly with equipment that is being virtualized. It is completely free from the working framework, not in any way like a Type-2 hypervisor, and boots before the working structure (OS).

Beginning now, Type-1 hypervisors are being utilized by all the bona fide players in the desktop virtualization space, including however not constrained to VMware, Microsoft and Citrix.

The developed virtualization programming or sort 2 hypervisor is consistently introduced on any host working framework.

In the event that host working structure gets decay or walloped by any reason, the virtualization programming or sort 2 hypervisor will in addition be pulverized and obviously all virtual machines and unmistakable assets will be lost. That is the reason the advancement of hypervisor or uncovered metal establishment is amazingly without a doubt comprehended in the spread preparing world.

Sort 2 (Hosted) hypervisors execute inside of a standard working framework environment. With the hypervisor layer as a particular second programming level, visitor working structures keep running at the third level over the equipment. A Type-2 hypervisor is a sort of customer hypervisor that sits on top of a working structure.

Not in any way like a Type-1 hypervisor, a Type-2 hypervisor depends genuinely on the working structure. It can't boot until the working structure is beginning now up and running and, if for any reason the working framework crashes, all end-clients are affected.
This is a vital downside of Type-2 hypervisors, as they are for the most part as secure as the working structure on which they depend. Besides, since Type-2 hypervisors rely on upon an OS, they are not in full control of the end client's machine.

Hypervisor Type 1 Products

- VMWare ESXi
- Citrix Xen
- KVM (Kernal Virtual Machine)
- Hyper-V

Hypervisor Type 2 Products

- VMware Workstation
- VirtualBox

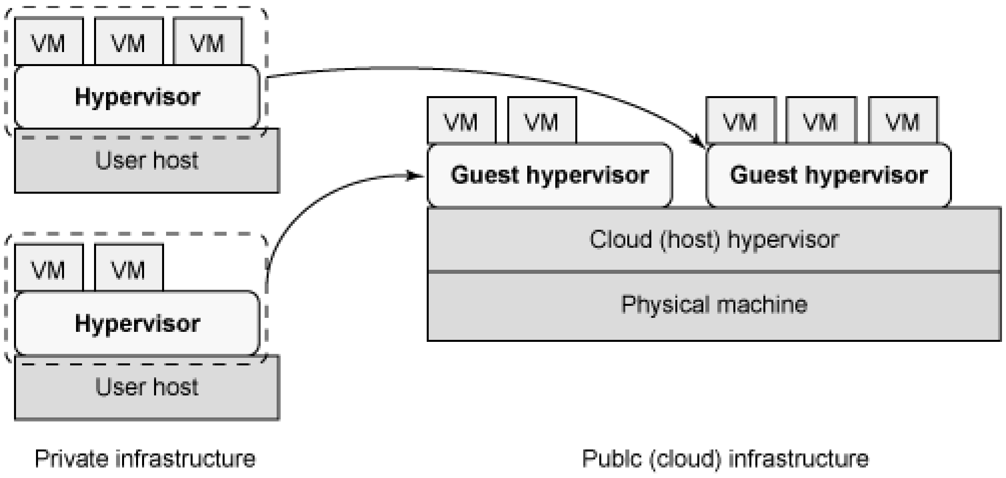


Figure 1.3 - Hypervisor Technology [6]

Data Centers and Uptime Tier Levels

As virtual machine is one of the mandatory aspects of the cloud computing, the term data center is also essential part of the technology. All the cloud computing infrastructures are located in the remote data centers that used to keep all the resources including computer systems and associated components, such as telecommunications and storage systems. Data centers classically includes redundant or backup power supplies, redundant data communications connections, environmental controls, air conditioning, fire suppression as well as security devices.

Tier level is considered as the rating or evaluation aspects of the data centers. Large data centers are used for industrial scale operations using as huge electricity consumption such as a small town. The standards are comprised of a four-tiered scale, with Tier 4 being the most robust and full featured.

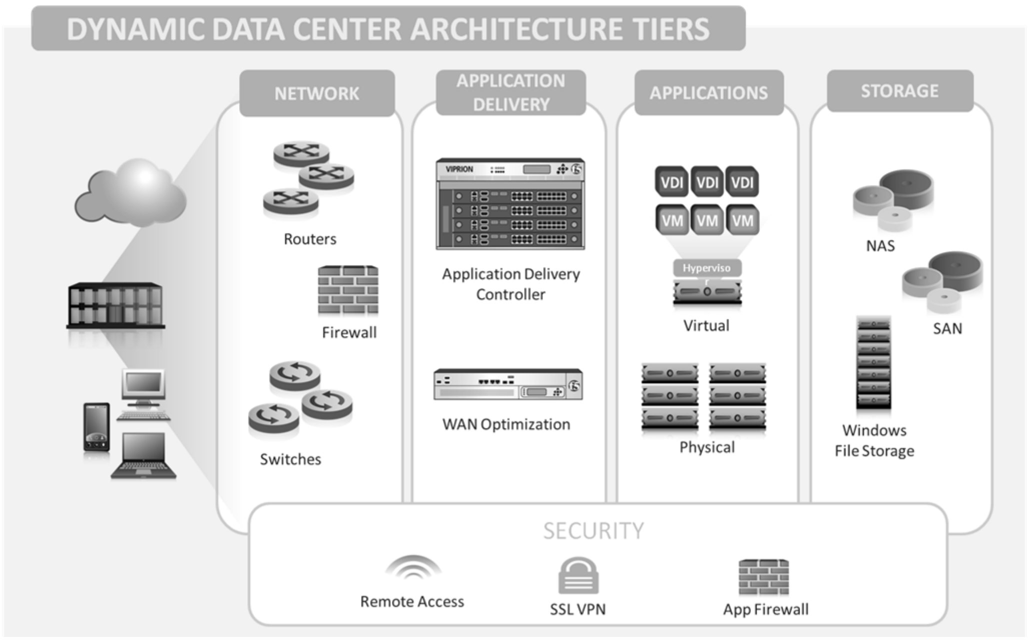


Figure 1.4 – Data Center Architecture Layers and Tiers [3]

1.3 Cloud Service Providers and their Services

Currently, numbers of cloud service providers are in the global market. Following is the list of cloud service providers in the domain of storage -

- JustCloud
- Zipcloud
- Dropbox
- Zoolz
- Livedrive
- Carbonite
- Backblaze
- 4shared
- Sosonlinebackup.com
- Mozy.com
- Crashplan.com
- Sugarsync.com
- Spideroak.com

- Mega.co.nz
- Google.com
- Onedrive.com
- Safecopybackup.com
- Bitcasa.com

Research Areas in Cloud Computing

Cloud computing and related services are very frequently taken as the research domain by the research scholars as well as academic practitioners. As cloud services are having number of domains, deployment models and respective algorithmic approaches, there are huge scope of research.

Following topics can be worked out by the research scholars as well as practitioners in the domain of cloud infrastructure -

- Energy Optimization
- Load Balancing
- Security and Integrity
- Privacy in Multi-Tenancy Cloud
- Virtualization
- Data Recovery and Backup
- Data Segregation and Recovery
- Scheduling for Resource Optimization

1.4 WORKING WITH DIFFERENT CLOUD PLATFORMS

The implementation is done in Eclipse IDE with Cloud and Grid Simulators in the External JAR Libraries for compatibility with Cloud and Grid Infrastructure. It is found and concluded from the results that EDSRTF algorithm is having effective results in cumulative way if consider all the parameters in investigation as a whole level.

Table 1.2 Key Cloud Service Providers - Storage as a Service

JustCloud	Zipcloud	Dropbox
Zoolz	Livedrive	Carbonite
Backblaze	4shared	Sosonlinebackup.com
Mozy.com	Crashplan.com	Sugarsync.com
Spideroak.com	Mega.co.nz	Google.com
Onedrive.com	Safecopybackup.com	Bitcasa.com

Table 1.3 Key Cloud Service Providers - Infrastructure as a Service (IaaS)

Amazon Web Services	AT & T Cloud Computing Services	Verizon
CloudScaling	DataPipe	ENKI
Enomaly	Eucalyptus Systems	GoGrid
HP	Joyent	LayeredTech
LogicWorks	NaviSite	OpSource
Rackspace	SAVVIS	Terremark

Table 1.4 Key Cloud Service Providers - Platform as a Service (PaaS)

ActiveState	Anaplan	AppearIQ
Apprenda	AppScale	AWS Elastic Beanstalk
Calipso Comunicaciones S.A.	Cloud Foundry	CloudControl
Cloudera	Distelli	Corvisa
Engine Yard	Google App Engine	Heroku
Hewlett Packard	IBM Bluemix	Jelastic
Microsoft Azure	Mendix	Morpheus
OpenShift	Oracle	OutSystems
Pivotal Software	Progress Software	QlikView
Ragic	Ricoh	Red Hat
Rollbase	Salesforce.com	Tsuru

A hypervisor or virtual machine monitor (VMM) is a piece of computer software, firmware or hardware that creates and runs virtual machines. A computer on which a hypervisor is running one or more virtual machines is defined as a host machine. Each virtual machine is called a guest machine. The hypervisor presents the guest operating systems with a virtual operating platform and manages the execution of the guest operating systems. Multiple instances of a variety of operating systems may share the virtualized hardware resources.

1.5 RESOURCE ALLOCATION

When developing a resource allocation system, one should think about how todescribe the resources present in the Cloud. The development of a suitable resourcemodel and description is the first challenge that an RAS must address. An RAS alsofaces the challenge of representing the applications requirements, called resourceoffering and treatment . Also, an automatic and dynamic RAS must be aware of thecurrent status of the Cloud resources in real time. Thus, mechanisms for resource discovery and monitoring are an essential part of this system. These two mechanismsare also the inputs for optimization algorithms, since it is necessary to know theresources and their status in order to elect those that fulfill all the requirements.Figure 5.5 shows how these challenges are related. First, the provider faces theproblems grouped in the Conception Phase , where resources must be modeledaccording to the variety of services the Cloud will provide and the type of resources thatit will offer. The next two challenges are faced in the scope of the Operational Phase .When requests for resources arrive, the RAS should initiate resource discovery todetermine if there are indeed resources available in the Cloud to attend the request.Finally, if resources are available, the RAS may select and allocate them to serve therequest.

RESOURCE MODELING AND DESCRIPTION

Resource modeling and description defines how the Cloud functions and how it dealswith infrastructural resources. This modeling is essential to all operations in the Cloud.Management, control, and optimization algorithms are all dependent on the resourcemodel chosen by the operator. Also, the services that a Cloud will provide to developersdepend on this concept.Network and computing resources may be described by several existingspecifications, such as Resource Description Framework (RDF) and Network Description Language (NDL). However, with Clouds it is very important that resourcemodeling takes into account schemas for representing virtual resources, virtualnetworks, and virtual applications. According to [Houidi et al. 2009], virtual resourcesneed to be described in terms of properties and functionalities much like as in the wayservices and devices/nodes are described in existing service architectures.It is important to note that if a particular model is very detailed (description at avery low level) it means that it's details are relevant and should not be disregarded,which in turn makes the optimization problem much more difficult to handle and solve.For example, if we consider that a request is composed of all the physical specificationsof each machine, we will have to deal with the little details that in another situationcould be disregarded or considered irrelevant. In this way, matching the request withavailable resources is rendered more difficult due to the peculiarities of each request,and achieving a generic solution is considerably more complicated. On the other hand,more details can give more flexibility and allow for a better usage of resources. Thisconcept is called the granularity of the resources description. Resource Offering and Treatment Once the resources are modeled, the Cloud provider may offer and handle them throughan interface . At a lower level, the RAS should handle the Cloud resources and, at ahigher level, address applications' requirements.The resource offering interface should provide some way for developers toclearly describe the requirements of their application. These requirements may berepresented by some form of a Service Level Agreement (SLA) that must be ensured bythe provider through continuous monitoring of QoS, due to the dynamic nature of theCloud [Patel et al. 2009][Yfoulis and Gounaris 2009]. Handling resources requires the implementation of solutions to control allresources available in the Cloud. Such control and management solutions would offer acomplete set of signaling protocols to set up hypervisors, routers, and switches.Currently, to delegate these control tasks, each Cloud provider implements their ownsolution that descends, in general, from datacenter control solutions. They also employvirtual datacenter solutions developed by vendors like VMWare or Citrix 33 . However, inthe future, new signaling protocols can be developed for integrated reservation of resources in the Cloud environment.

1.6 RESOURCE OFFERING AND TREATMENT

Once the resources are modeled, the Cloud provider may offer and handle them throughan interface . At a lower level, the RAS should handle the Cloud resources and, at ahigher level, address applications' requirements.The resource offering interface should provide some way for developers toclearly describe the requirements of their application. These requirements may berepresented by some form of a Service Level Agreement (SLA) that must be ensured bythe provider through continuous monitoring of QoS, due to the dynamic nature of theCloud [Patel et al. 2009][Yfoulis and Gounaris 2009]. Handling resources requires the implementation of solutions to control allresources available in the Cloud. Such control and management solutions would offer acomplete set of signaling protocols to set up hypervisors, routers, and switches.Currently, to delegate these control tasks, each Cloud provider implements their ownsolution that descends, in general, from datacenter control solutions. They also employvirtual datacenter solutions developed by vendors like VMWare or Citrix. It is important to highlight that resource modeling is not necessarily dependenton the way in which resources are offered to developers. For example, the Cloudprovider could model each resource individually, like independent items in a fine-grained scale (such as GHz of CPU or GB of memory), but can offer them like acoupled collection of items or a bundle, such as classes of VM (high memory and highprocessor types).As previously noted, in addition to traditional network requirements andcomputational requirements, new requirements for developers may be present inClouds. An example of such a requirement is the topology of the network , which maybe defined by developers. Developers are able to configure nodes' relationships andcommunication restrictions (down and uplinks, for example). Jurisdiction is anotherexample. It is related to where (physically) applications and their data must be storedand handled. Due to some restrictions – such as copyright laws – developers may solicitto store information in chosen places, such as specific countries or continents. The nodeproximity may be another restriction indicating that there should be a maximum (orminimum) physical distance (or delay value) between nodes or locations. Please notethat it may directly impact other requirements, such as topology. Resource Discovery and MonitoringResource discovery may be described basically as the task in which theprovider should find appropriate resources (suitable candidates) in order to comply withincoming developers requests. Also, more specific questions, like “How should onediscover resources with (physical) proximity in a Cloud?”, or “How does one causeminimal impact upon network traffic responsibility?” can be seen as a resourcediscovery responsibility and are not trivial to be answered, given the dynamic nature of the Cloud environment.Considering that one of the key features of Cloud Computing is the capability of acquiring and releasing resources on-demand [Zhang et al. 2010], resource monitoring should be continuous. This is because, in addition to serving as an information providerto handle new requests, it should also be informed of resources’ current status to makean informed decision of a possible reallocation, i.e., it must assist resource usageoptimization. The timing and the quantity of messages is a relevant issue to beemphasized, because the network bandwidth should be used rationally. In this way, acareful analysis should be done to find a suitable trade-off between the amount of messages and the frequency of information refreshing.The monitoring may be passive or active. It is considered passive when there isan entity (or a set of entities) that collects information from nodes continuously, i.e., thenodes are passive in relation to this entity. The entity may send messages to nodesasking for information or simply may retrieve information when necessary. When themonitoring is active , nodes are autonomous and may decide when to send stateinformation to some central entity. Clouds also make use of both alternativessimultaneously to improve the monitoring solution.A simple implementation of a resource discovery service is the discoveryframework used in an advertisement process as described in [Houidi et al. 2009]. Such aframework is proposed for a scenario based on a network virtualization environment,but it can be easily adapted for other scenarios. Such a framework can be used bybrokers to discover and match available resources, and typically is composed of distributed repositories, which are responsible for storing information about anddescriptions of physical and virtual resources.

1.7 RESOURCE SELECTION AND OPTIMIZATION

After acquiring information about available resources in the Cloud (during thediscovery phase), a set of appropriate candidates is highlighted. The resource selectionmechanism elects the candidate solution that fulfills all requirements and optimizes theusage of the infrastructure. In virtual networks, for example, the essence of resourceselection mechanisms is to find the best mapping between the virtual graph and thesubstrate graph. This should achieve the best load balancing in the substrate network resources, with respect to the capacity constrains [Houidi et al. 2008]. By this example,it is clear that selecting solutions from a set of available ones is not a trivial task due tothe dynamicity of the scenario and all the different requirements that must becontemplated by the provider.The resource selection may be done using an optimization algorithm. Manyoptimization strategies may be used, from simple and well-known techniques such assimple heuristics with thresholds or linear programming, to newer ones, such asLyapunov Optimization [Urgaonkar et al. 2010]. Moreover, traditional artificialintelligence algorithms – ant colony and game theory [Teng and Magouls 2010], forexample – are also viable for Clouds.One way to classify resource selection strategies is related to the moment whenoptimization techniques are applied: a priori and a posteriori.In the case of a priori techniques, the first allocation solution is already anoptimal solution. To achieve this goal, the optimization strategy should consider allvariables that influence the allocation process. For example, consider that VM instancesshould be allocated in the Cloud; the optimization strategy should determine theproblem, present a solution (or a set of possibilities) that satisfy all constraints, andattempt the goals (such as the minimization of resource reallocations) in an optimalmanner.In the case of a posteriori techniques, once an initial resource allocation is made,which can be a suboptimal solution, the RAS should manage its resources in acontinuous way. If necessary, decisions, such as to add more memory or storagecapacity to a VM, reallocate an already allocated application, or reallocate resources forother applications, should be taken in order to optimize the system utilization or tocomply with a developer’s requirements.Since the resource utilization and provisioning are dynamic (i.e., in addition tothe resources being requested and then allocated, they can also be de-allocated), it ismore interesting that the a posteriori optimization strategies reach an optimal allocationfirst, and are able to optimize the old requests and readjust them according to newdemand. In this case, the optimization strategy may also fit with the definition of apriori and dynamic classification.Furthermore, optimization strategies may also be applied to improve specificaspects in the Cloud such as energy saving or to cope with polices for security and fault-tolerance.

1.8 Taxonomy of Resource Allocation and Scheduling

In this section, the research portray our portrayal arrangement. The work acknowledge that occupation seizure is permitted. The work orchestrate booking counts into three classes based upon the available level of interprocessor development. The work in like manner perceive among three unmistakable classes of estimations based upon the open door with which needs may be dispensed. These two tomahawks of request are orthogonal to one another as in restricting a computation along one turn does not limit opportunity along the other. Thusly, there are 3 × 3 = 9 particular classes of booking estimations in this experimental order.

Movement Based Classification

Interprocessor movement has for the most part been forbidden ceaselessly systems for the going with reasons:

In various structures, the cost associated with each movement — i.e., the cost of trading a work's setting beginning with one processor then onto the following — can be prohibitive.

Starting quite recently, routine ceaseless arranging theory did not have the strategies, instruments, and results to permit a point by point examination of structures that allow migration. In this manner, isolating has been the favored procedure as a result of the non-vicinity of achievable choice approaches.Recent enhancements in PC basic arranging, including single-chip multiprocessors and snappy interconnection frameworks over little zones, have realized the first of these stresses ending up being less of an issue. Appropriately, system organizers require no more block interprocessor development solely on account of use considerations, especially in solidly coupled structures. (In any case, it might regardless be alluring to strict overhead remembering the finished objective to decrease runtime overhead.) besides, generally tests demonstrate that arranging computations that allow development are forceful to the extent schedulability with those that don't migrate, even in the wake of merging movement overheads . This is a result of the way that systems exist that can be successfully occupied just if interprocessor development is allowedIn isolating among multiprocessor arranging figurings according to the level of migration allowed, this work consider the going with three characterizations:

No migration(Partitioned) – In distributed booking figurings, the game plan of assignments is partitioned into the same number of disjoint subsets as there are processors available, and each such subset is associated with an uncommon processor. All businesses created by the assignments in a subset must execute just upon the relating processor.

Restricted Migration

In this class of booking figurings, each occupation must execute by and large upon a lone processor. Regardless, different occupations of the same undertaking may execute upon particular processors. Likewise, the runtime setting of each occupation ought to be kept up upon onlyone processor; nevertheless, the task level association may be moved.

Full Migration – No constraints are set upon interprocessor movement.

Need Based Classification In isolating among arranging figurings as showed by the multifaceted design of the need arrangement, this work again consider three classes.

Static Priorities An intriguing need is joined with each task, and all jobs created by an endeavor have the need associated with that errand. In this way, if errand T1 has higher need than task T2, then at whatever point both have dynamic businesses, T1's occupation will have need over T2's occupation. An instance of a booking computation in this class is the RM figuring .

Job Level Dynamic requirements For every pair of occupations Ji and Jj, if Ji has higher need than Jj at some minute in time, then Ji constantly has higher need than Jj . An instance of an arranging count that is in this class, however not the past class, is EDF.

Unrestricted Dynamic Properties No restrictions are put on the needs that may be apportioned to businesses, and the relative need of two occupations may change at whatever point. An example booking count that is in this class, however not the

P fair scheduling - Starting late, much research has been done on overall multiprocessor booking computations that ensure sensibility. Proportionate-sensible (Pfair) booking, proposed by Baruah et al. , is in barely a second the principle known perfect procedure for booking irregular ceaseless assignments on a multiprocessor structure. Under Pfair arranging, each endeavor is alloted a weight that shows the rate at which that task should execute: an errand with weight w would ideally get w • L units of processor time over any interval of length L. Under Pfair arranging, endeavors are occupied by settled size task quantum so that deviation from an impeccable assignment is completely bounded.Currently, three perfect Pfair booking counts are known: PF , PF, and PD2 . Of these figurings, PD2 is the most starting late made and the most capable.

The crucial good position of Pfair arranging over isolating is the ability to arrange for any achievable irregular, sporadic, or rate-based task system3. Hereafter, Pfair arranging counts can perfectly handle element events, for instance, errands leaving and joining a structure. Also, sensible multiprocessor arranging figurings are ending up being more noticeable in view of the increase of web and sight and sound applications. For instance, Ensim Corp., an Internet organization supplier, has sent sensible multiprocessor arranging counts in its item advertising.

The essential weight of Pfair arranging is ruined processor inclination. Processor proclivity implies the slant of assignments to execute speedier when more than once anticipated the same processor. This slant is for the most part the delayed consequence of per-processor first-level putting away. Acquisitions and migrations, both of which tend to happen chronically under Pfair booking, limit the feasibility of these first-level stores and can incite extended execution times as a result of store misses. On the other hand, under allocating with EDF, there is no development and the amount of acquisitions on a processor is constrained by the amount of occupations on that processor (tolerating free endeavors).

Layer-Based Scheduling Algorithms - There has been a lot of investigation as for arranging counts for self-sufficient M-Tasks. Regardless, these arranging figurings can't adjust to need objectives between M-Tasks. This imprisonment can be refrained from using layer-based booking algorithms3 for MTasks with need goals. These counts utilize a contracting stage and a layering stage to separate a M-Task dag into sets of self-governing M-Tasks, called layers. The subsequent layer booking stage enlists a timetable for each layer in partition. Our extension system enables the blend of the contracting stage, the layering stage and the social event stage with an arranging figuring with the expectation of complimentary M-Tasks in the layer booking stage.

Layer Scheduling Algorithms

In this stage a M-Task calendar is registered for each developed layer VLi , i = 1, . . . , l in confinement. In the accompanying this work overlook the record i and use VL for the layer to be booked.

Two L-Level decides the aggregate execution time for every conceivable apportioning of the arrangement of accessible processors into K, K = 1, . . . , min(P, |VL|)

subgroups ˆg K,1, . . . ˆgk,k of about equivalent size3. The timetable for each of these partitionings is processed by receiving a rundown planning heuristic. In every progression of this heuristic the M-Task v € VL is appointed to assemble ˆg* €{ˆg K,1, . . . ˆg K, K }, where ˆg K is the first subgroup getting to be accessible and v is the M-Task with the biggest execution time. The last processor bunches g1, . . . , g K* are figured by a resulting gathering change venture from the gatherings ˆg K*,1, . . . , ˆgk*,k* , where K* indicates the parceling bringing about a base runtime.

Two L-Tree begins by building a tree for every M-Task v € VL comprising of a solitary node8. A dynamic programming methodology is utilized to locate every unordered pair of trees {t1, t2} with an equivalent profundity and disjoint arrangements of M-Tasks. For every pair {t1, t2} another tree t with another root hub and youngsters t1 and t2 is made. Every tree speaks to a calendar of the contained M-Tasks. The inward hubs of the trees are explained with an expense table containing the execution time of the entire subtree for all conceivable processor gathering sizes gs = 1, . . . , P. A second annotation characterizes whether the timetables spoke to by the offspring of the hub ought to be executed one after other or in parallel on disjoint processor bunches. At last, an arrangement of trees each containing all hubs of the present layer is developed, where each such tree symbolizes an alternate timetable. The yield calendar of the calculation is built from the tree which concedes the base execution time.

1.10 LLREF Scheduling Algorithm Model

This work consider worldwide planning, where undertaking relocation is not confined, on a SMP framework with M indistinguishable processors. This research consider the application to comprise of an arrangement of undertakings, meant T = {T1, T2, ..., TN}. Undertakings are expected to arrive intermittently at their discharge times ri. Every assignment Ti has an execution time ci, and a relative due date di which is the same as its period pi. The use ui of an undertaking Ti is characterized as ci/di and is thought to be under 1. This work expect that undertakings may be acquired whenever, and are free, i.e., they don't share assets or have any priorities.

This work consider a non-work moderating planning approach; subsequently processors may be sit out of gear notwithstanding when assignments are available in the prepared line. The expense of setting switches and undertaking relocations are thought to be unimportant .

Time and Local Time Execution

In the liquid booking demonstrate, every assignment executes at a steady rate at all times . The quantum-based Pfair booking calculation, the main known ideal calculation for the issue that this work consider here, depends on the liquid planning model, as the calculation always tracks the distributed undertaking execution rate through errand use. The Pfair calculation's achievement in building ideal multiprocessor timetables can be credited to decency—casually, all assignments get an offer of the processor time, and in this manner can all the while gain ground. P-reasonableness is a solid thought of decency, which guarantees that at any moment, no application is one or more quanta far from its due offer (or liquid calendar) [2, 5]. The centrality of the decency idea on Pfair's optimality is likewise upheld by the way that assignment earnestness, as spoke to by the errand due date is not adequate for building ideal timetables, as the work see from the poor execution of worldwide EDF for multiprocessors.

Toward outlining an ideal booking calculation, the work subsequently consider the liquid planning model and the reasonableness thought. To maintain a strategic distance from Pfair's quantum-based methodology, the work consider a reflection called the Time and Local Execution

Time Domain Plane (or condensed as the T-L plane), where tokens speaking to errands move after some time. The TL plane is propelled by the L-C plane deliberation presented by Dertouzos et al. in . the work utilize the T-L plane to portray liquid calendars, and present another planning calculation that can track the liquid timetable without utilizing time quanta.

For an undertaking Ti with ri, ci and di, the a 2-dimensional plane with time spoke to on the x-hub and the errand's remaining execution time spoke to on the y-pivot. On the off chance that ri is accepted as the birthplace, the spotted line from (0, ci) to (di , 0) shows the liquid timetable, the incline of which is −ui. Since the liquid timetable is perfect yet for all intents and purposes incomprehensible, the decency of a booking calculation relies on upon how much the calculation approximates the liquid calendar way.

Note that assignment execution is spoken to as a line whose incline is - 1 since x and y tomahawks are in the same scale, and the non-execution after some time is spoken to as a line whose slant is zero. It is clear that the Pfair calculation can likewise be spoken to in the T-L plane as a broken line taking into account time quanta.

At the point when N numbers of assignments are viewed as, their liquid timetables can be built, and a privilege isosceles triangle for every undertaking is found between each two back to back booking events2 and N triangles between each two successive planning occasions can be covered together. The work call this as the T-L plane TLk, where k is just expanding after some time. The extent of TLk may change over k. The base side of the triangle speaks to time. The left vertical side of the triangle speaks to the hub of a part of errands' remaining execution time, which the work call the neighborhood remaining execution time, li, which should be devoured before each TLk closes. Liquid calendars for every undertaking can be built as covered in each TLk plane, w

Genetic Algorithm

Genetic Algorithm (GAs) are request frameworks considering models of typical determination and inherited qualities (Fraser, 1957; Bremermann, 1958; Holland, 1975). The work start with a brief preface to essential inherited counts and related wording.

GAs encode the decision variables of a request issue into restricted length arrangement of letters all together of certain cardinality. The strings which are contender responses for the chase issue are suggested as chromosomes, the letters all together are insinuated as qualities and the estimations of characteristics are called alleles. For example, in an issue, for instance, the voyaging deals agent issue, a chromosome identifies with a course, and a quality may identify with a city. Rather than standard improvement systems, GAs work with coding of parameters, rather than the parameters themselves.

To create extraordinary game plans and to complete trademark decision, the work require a measure for perceiving incredible courses of action from dreadful game plans. The measure could be an objective limit that is a numerical model or a PC entertainment, or it can be a subjective limit where individuals pick better game plans over all the more horrendous ones. In a general sense, the wellbeing measure must choose a candidate course of action's relative health, which will thusly be used by the GA to control the improvement of good plans.

Chapter-2

LITERATURE SURVEY

To propose, defend and depict the novel research work, number of research articles, papers and conferences are investigated. Following is the list and explanation to the excerpts fetched from the multiple ad assorted research tasks done by the academicians and researchers.

2.1 LITERATURE REVIEW

Kune etl. al. [1] (2014) - In this paper, the authors proposed a Genetic Algorithm based scheduler for such Big Data Cloud where decoupled computational and information administrations are offered as administrations. The methodology is taking into account developmental systems focused on information conditions, computational assets and viable use of data transmission subsequently accomplishing higher throughputs

Mathew et. al. [2] (2014) – The authors displayed and presented a nitty gritty investigation of different undertaking booking routines existing for the cloud environment. A brief investigation of different booking parameters considered in these systems is additionally examined in this paper.

Singh et. al. [3] (2014) - In this paper, a hereditary calculation based planning methodology is proposed in which beginning populace is created with development rendition of Max Min by which the work can get more streamline results, regarding make compass. The execution of the proposed Modified Genetic Algorithm (MGA) and existing calculations have been assessed against the example information. Trial results demonstrate that proposed calculation shows adequate execution and beats the current calculations.

Verma et. al. [4] (2014) - Task planning and asset assignment are the key difficulties of distributed computing. Contrasted and framework environment, information exchange is a major overhead for cloud work processes. Along these lines, the expense emerging from information exchanges between assets and also execution costs should likewise be considered amid planning based upon client's Quality of Service (QoS) imperatives.

Javanmardi et. al. [5] (2014) - In this paper with the guide of hereditary calculation and fluffy hypothesis, the authors presented a half genetic based job planning methodology, which considers the heap adjusting of the framework and decreases downright execution time and execution cost. The work attempted to adjust the standard Genetic calculation and to decrease the cycle of making populace with the guide of fluffy hypothesis. The principle objective of this exploration is to appoint the employments to the assets with considering the VM MIPS and length of jobs. The new calculation relegates the employments to the assets with considering the employment length and assets limits. The work assess the execution of our methodology with some renowned cloud booking models. The aftereffects of the investigations demonstrate the effectiveness of the proposed approach in term of execution time, execution cost and normal Degree of Imbalance (DI).

Rodriguez et. al. [6] (2014) – The authors in this work proposed and presented an asset provisioning and planning system for investigative work processes on Infrastructure as a Service (IaaS) mists. This work exhibit a calculation taking into account the meta-heuristic improvement system, molecule swarm enhancement (PSO), which expects to minimize the general work process execution expense while meeting due date limitations. The heuristic is assessed utilizing CloudSim and different no doubt understood logical work processes of distinctive sizes. The outcomes demonstrated that the methodology performs better than the current best in class calculations.

Singh et. al. [7] (2014) - In this paper, a hereditary calculation has been recommended that timetables work process applications in questionable cloud environment and meet client characterized QoS limitations. A financial plan compelled time minimization hereditary calculation has been proposed which diminishes the disappointment rate and makespan of work process applications. It assigns those assets to work process application which are solid and expense of execution is under client spending plan. The execution of hereditary calculation has been contrasted and max-min and min-min planning calculations in inconsistent cloud environment.

Shojafar et. al. [8] (2015) - Job planning has been a standout amongst the most vital examination issues in circulated frameworks, especially cloud situations/processing. This work scientifically demonstrate our streamlining issue which is arched with surely understood investigative conditions (particularly, Karush–Kuhn–Tucker conditions). The authors in this work contrasted the execution of our methodology with a few other cloud booking models.

This paper presented the improvement of hereditary calculation way to deal with calendar undertakings on a multiprocessor framework. The goal is to minimize the make-compass i.e. the finishing time of all errands while keeping up the priority requirements inside of the undertaking chart. No between processor correspondence overheads are accepted. The exhibit information structure is utilized for string representation and a mixture determination technique for multiplication is embraced. The capacity of the genetic based scheduler to manage asset disappointments and aperiodic operations is likewise investigated.

Undertaking Scheduling is the distribution of assets over the long haul to perform an accumulation of assignments. Constant frameworks make utilization of planning calculations to augment the quantity of ongoing errands that can be prepared without damaging timing imperatives. A planning calculation gives a timetable to an undertaking set that appoints assignments to processors and gives a requested rundown of errands. The timetable is said to be achievable if the timing imperatives of the considerable number of undertakings are met. Planning methodologies can be arranged by entry time of undertakings into static and progressive and deterministic or stochastic booking. Assignment Scheduling in Multiprocessor is a term that can be expressed as discovering a calendar for a general errand diagram to be executed on a multiprocessor framework so that the timetable length can be minimized. Undertaking planning in multiprocessor frameworks otherwise called multiprocessor booking. Multiprocessor booking issues can be arranged into a wide range of classifications in view of attributes of the project and errands to be planned, the multiprocessor framework, and the accessibility of data. Multiprocessor booking issues may be separated in two classes: Static and element errand planning. In this paper creator talk about the Multiprocessor Environment utilizing hereditary Algorithm.

Current shared memory multicore and multiprocessor frameworks are nondeterministic. Every time these frameworks execute a multithreaded application, regardless of the possibility that supplied with the same information, they can deliver an alternate yield. This disappoints troubleshooting and limits the capacity to appropriately test multithreaded code, turning into a noteworthy hindrance to the quite required boundless selection of parallel programming. In this paper the work present the defense for completely deterministic shared memory multiprocessing (DMP). The conduct of a subjective multithreaded program on a DMP framework is just a component of its inputs. The center thought is to make between string correspondence completely deterministic. Past ways to deal with adapting to nondeterminism in multithreaded projects have concentrated on replay, a system valuable just for investigating. Conversely, while DMP frameworks are specifically helpful for investigating

2.2 FINDINGS FROM SURVEY

After analytical review and investigation of the research papers and manuscripts from various primary and secondary sources, it is found that the scheduling as well as routing is lacking the integration of specialized optimization techniques. These techniques can improve the existing results on the basis of multiple parameters. From the existing work and literature review, the results are effective but can be improved using genetic algorithm or related techniques for combinatorial optimization problems.

Work Extracted from the Literature Survey

Ref. No.	**Paper Name (Authors)**	**Technique Used**	**Advantages**	**Limitations**
1	Kune etl. al. [1] (2014)	Propose a unique implementation of Genetic Algorithm based scheduler for such Big Data Cloud	Resource Optimization based and Effective	Vulnerable and not good for general context
2	Mathew et. al. [2] (2014)	Deep investigation of different undertaking scheduling routines existing for the cloud environment. A brief investigation of different scheduling parameters considered in these systems is additionally examined in this work.	Effective Scheduling Approach and Time Aware	Complex for some cases
3	Singh et. al. [3] (2014)	The execution of the proposed Modified Genetic Algorithm (MGA) and existing calculations have been assessed against the example information. Simulation results depict that proposed calculation shows adequate execution and beats the current calculations.	Improved Optimization using MGA and having higher effectiveness	Resource Allocation not effective and complex

4	Verma et. al. [4] (2014)	The work is related to the key difficulties of distributed computing. Contrasted and framework environment, information exchange is a major overhead for cloud work processes.	Pragmatic review and analysis on the approaches and their remarks	Particular timeline followed
5	Javanmardi et. al. [5] (2014)	Exhibit of job planning methodology, that considers the heap adjusting of the framework and decreases downright execution time and execution cost. This work simulates the Genetic algorithm and to decrease the cycle of making populace with the guide of fluffy hypothesis.	Optimization of the cycle associated with the algorithm	Time Constraint
6	Rodriguez et. al. [6] (2014)	Proposes asset provisioning and planning system for investigative work processes on Infrastructure as a Service (IaaS) mists. The authors exhibit a detailed algorithmic implementation with PSO	Global Optimization and effective approach	Higher time complexity and cost factor
7	Singh et. al. [7] (2014)	The execution of genetic algorithm based implementation is measured on the cloud	Effective algorithm for scheduling is	Resources are required for all dimensions implementation

		based scheduling and integration The measured aspects contrasted and max-min and min-min planning calculations in inconsistent cloud environment.	used	
8	Shojafar et. al. [8] (2015)	The work scientifically demonstrates our streamlining issue arched with surely evident investigative conditions (particularly, Karush–Kuhn–Tucker conditions). The work contrasts the execution of our methodology with a few other cloud booking models.	Pragmatic review and effective analysis	Particular timeline is followed and not in general

Grid computing is a unique and effective way to use the computing infrastructure for effective communication and less overhead. The major technology behind grid as well as cloud computing is the virtualization technology. Using virtualization technology, the number or arbitrary and remote processors works in parallel for the execution of tasks in such a way that there is less complexity and higher integrity of the processes. In grid architecture, there are number of remote computing devices and processors which work and executes the processes in parallel so that there is minimum load on the single node or machine. By this way, the overall load is balanced.

2.3 AIMS AND PROPOSED WORK

This research work is having the focus on development as well as implementation of a job scheduling algorithm for effective scheduling and prioritization. By this way and the proposed technique, the effective results in terms of less cost, higher performance and less execution time is obtained.

In this research work, the implementations have analyzed the performance of assorted algorithms based on new parameters and dimensions. There is need to analyze the performance of job scheduling algorithms on cloud as well as grid environment as priority, load balancing and related aspects can affect the performance of cloud and grid to a huge extent. The work done in this research includes proposing the implementation of algorithms on new parameters so that the hidden aspects and attributes of the algorithms can be analyzed. The work includes to perform the implementation / simulation and analysis on algorithms on new parameters beyond the work in the base literature.

KEY POINTS AND CONCLUSIONS

- The proposed algorithm is implemented and integrated using genetic algorithm for optimization of the results including the cost and performance factor.
- The proposed system is generating efficient results in terms of the optimal solution when executed using genetic algorithm.
- The proposed technique is efficient also in terms of the execution and turnaround time despite of the number of iterations
- The limitations may be included regarding the proposed work in terms of its further enhancement using assorted metaheuristics.
- The proposed system gives better results in executed using genetic algorithm that is one of the prominent metaheuristic techniques.

Chapter-3

PROBLEM FORMULATION

3.1 PROBLEM IN EXISTING SYSTEM

In the proposed work of resource allocation in cloud, the work have built up a special and novel calculation making utilization of crossover and parallel methodology of taking care of and preparing the employments in succession. As in the exploration issue plan, the Grid is considered as a complete foundation for the sharing of asset. It is traditionally utilized for substantial scale information handling, a large number of the applications being experimental ones. Lattice booking is an indispensable segment of a Grid base.

3.2 PROBLEM FORMULATED

Resource Allocation Algorithms in Cloud Computing

3.3 OBJECTIVE OF THE PROBLEM

- To investigate the drawbacks and shortcomings in the classical resource allocation and optimization in cloud infrastructure
- To propose and implement a novel technique for the simulation of genetic algorithm based resource allocation in cloud

3.4 TECHNICAL IMPLEMENTATION AND APPROACH

=> Design computation cost matrix and communication matrix.

=> Assign priorities to tasks.

=>Calculate EST=0 & EFT=0 of task 1 on each processor, set process avail time for p1,p2,p3=0. Assign processor to task1 on which EFT is minimum

either p1=EFT (T1or task1) or p2=EFT (T1) or P3=EFT(T1)

=>for t2 to tn repeat above

Calculate if (parent of t1 executed on p1)

EST (tn) on p1=max (parent executed)

p2=max (parent completion+Communication cost), processor avail time)

EST (tn) on p3= same as above

EFT=EST+Computation cost

ADVANTAGES AND LIMITATIONS OF THE SYSTEM

- The proposed system is generating efficient results in terms of the optimal solution when executed using genetic algorithm.
- The proposed technique is efficient also in terms of the execution and turnaround time despite of the number of iterations
- The limitations may be included regarding the proposed work in terms of its further enhancement using assorted metaheuristics.
- The proposed system may give better results in executed using simulated annealing that is one of the prominent metaheuristic technique.

PLATFROM SETUP FOR THE IMPLEMENTATION

- JDK
- Eclipse IDE
- CloudSim
- GridSim
- JUnit
- JCharts

FORMULATION OF THE PARAMETERS

Time complexity => O (n^2)

=> O(1) + O (n) + O (n^2) + O (n) + O (n) + O (n) + O (n^2) + O (n^2) + n Log (n) + n Log (n) + n Log (n)

=> O (n^2) + n Log (n)

Space complexity

=> O (n) + O (n) + 8 * O(n) + 8 * O(n) + O(n) + n log (n) + 8 * O(n)
=> O (n)

Reliability
r => n * (1/t) * rnd
n -> Length of the Input
t -> Execution Time
rnd -> Random Fuzzy Random

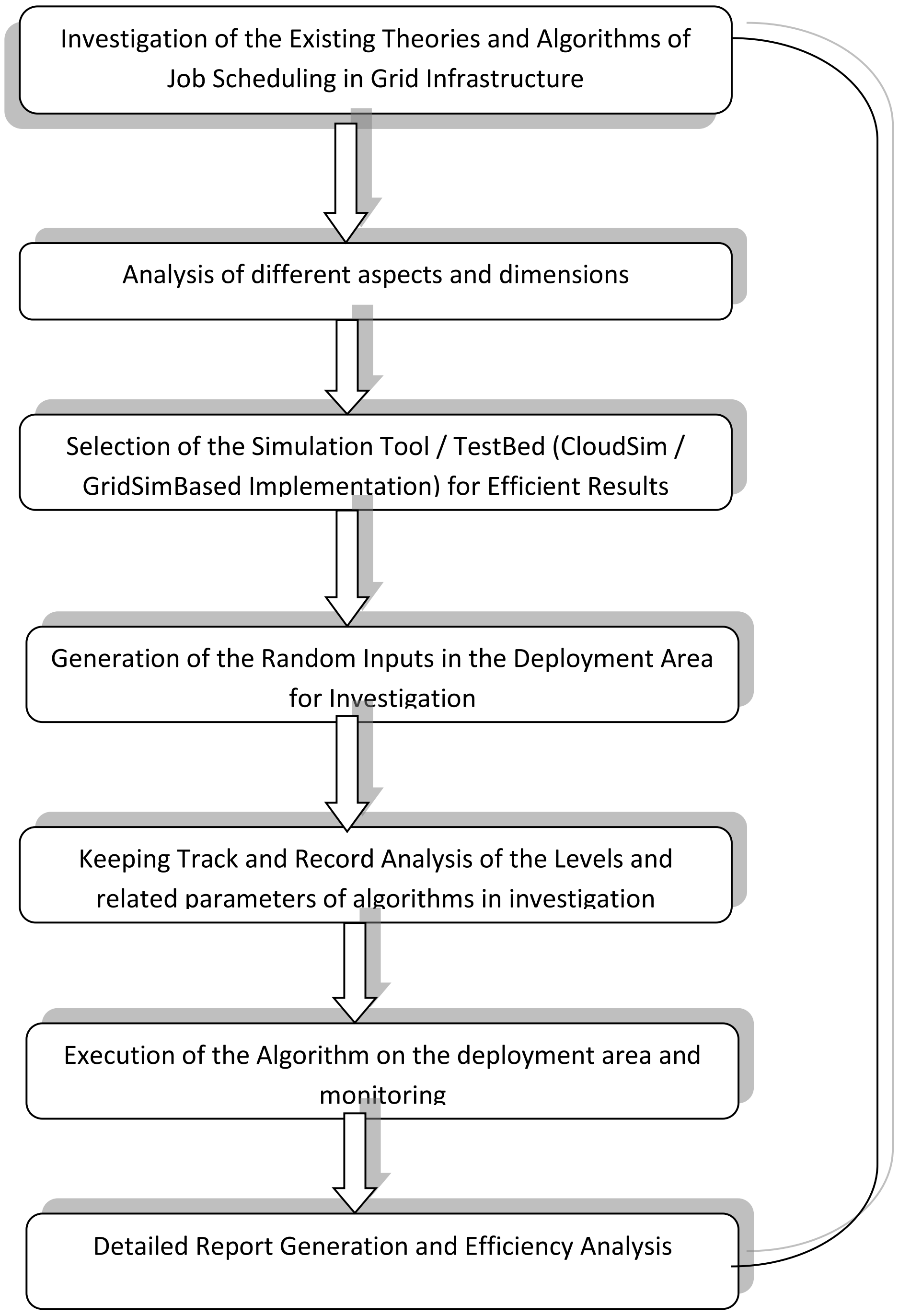
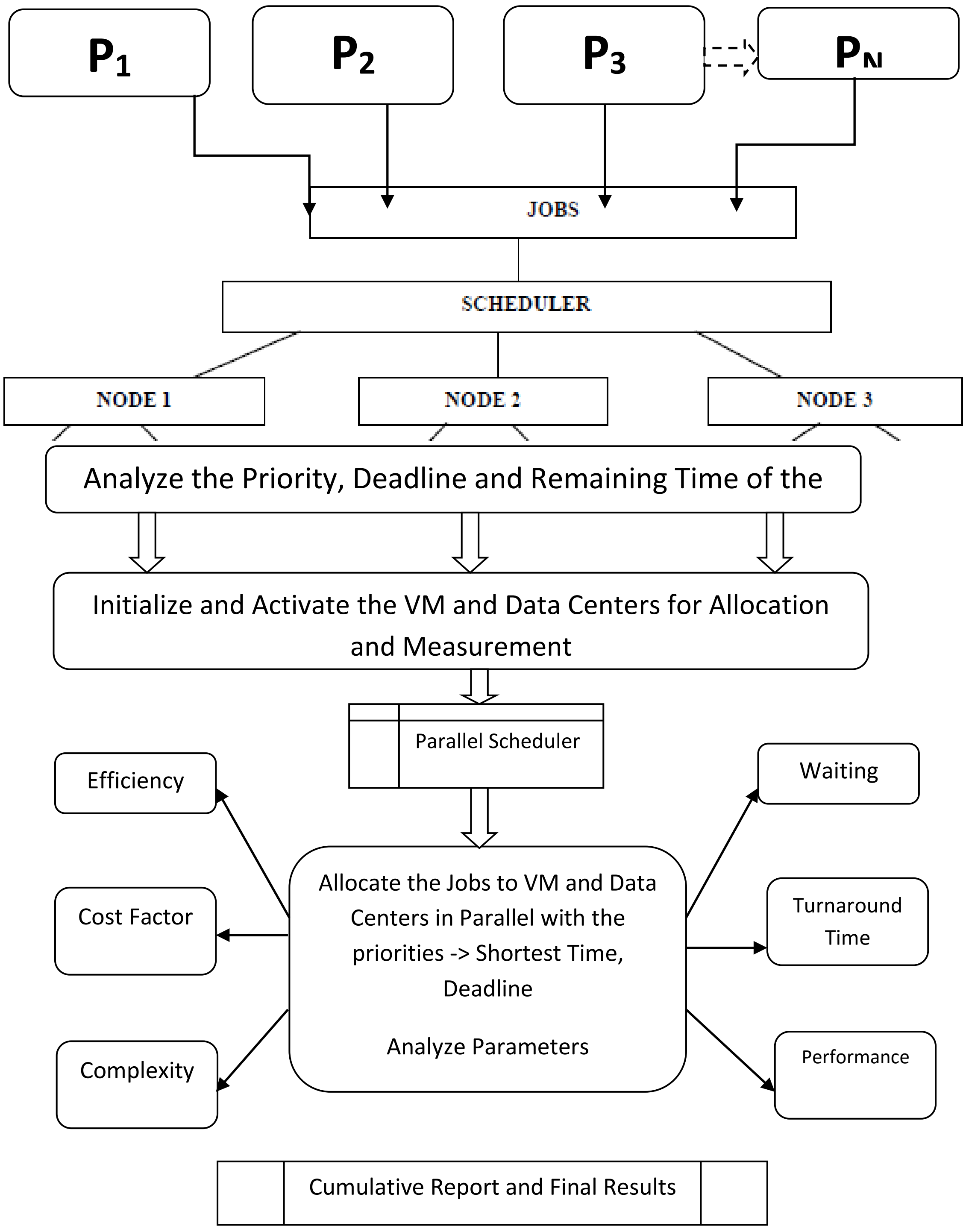

Investigation of the Existing Theories and Algorithms of Job Scheduling in Grid Infrastructure
Analysis of different aspects and dimensions
Selection of the Simulation Tool / TestBed (CloudSim / GridSimBased Implementation) for Efficient Results
Generation of the Random Inputs in the Deployment Area for Investigation
Keeping Track and Record Analysis of the Levels and related parameters of algorithms in investigation
Execution of the Algorithm on the deployment area and monitoring
Detailed Report Generation and Efficiency Analysis

Fig. 3.1 Proposed Algorithm Flow

Portability (p) and Efficiency (e) is directly proportional to reliability
Cost factor (c) ->
$c = 1/(r * (p + e)) * 100$
Reusability (ru) = r



P_1
P_2
P_3
P_N
JOBS
SCHEDULER
NODE 1
NODE 2
NODE 3
Analyze the Priority, Deadline and Remaining Time of the
Initialize and Activate the VM and Data Centers for Allocation and Measurement
Parallel Scheduler
Efficiency
Cost Factor
Complexity
Allocate the Jobs to VM and Data Centers in Parallel with the priorities -> Shortest Time, Deadline
Analyze Parameters
Waiting
Turnaround Time
Performance
Cumulative Report and Final Results

Fig. 3.2 – Allocation of Tasks to VM and related aspects

IMPLEMENTATION, RESULTS AND DISCUSSION

The implementation is done in Eclipse IDE with Cloud and Grid Simulators in the External JAR Libraries for compatibility with Cloud and Grid Infrastructure. It is found and concluded from the results that EDSRTF algorithm is having effective results in cumulative way if the work consider all the parameters in investigation as a whole level.

- The existing and base multiprocessor scheduling approaches are having huge pitfalls more execution time
- The basic solution obtained without swarm intelligence is not efficient in terms of the turnaround time and optimal results
- To investigate the drawbacks and shortcomings in the classical multiprocessor scheduling
- To propose and implement a novel technique for the simulation of genetic algorithm based multiprocessor scheduling

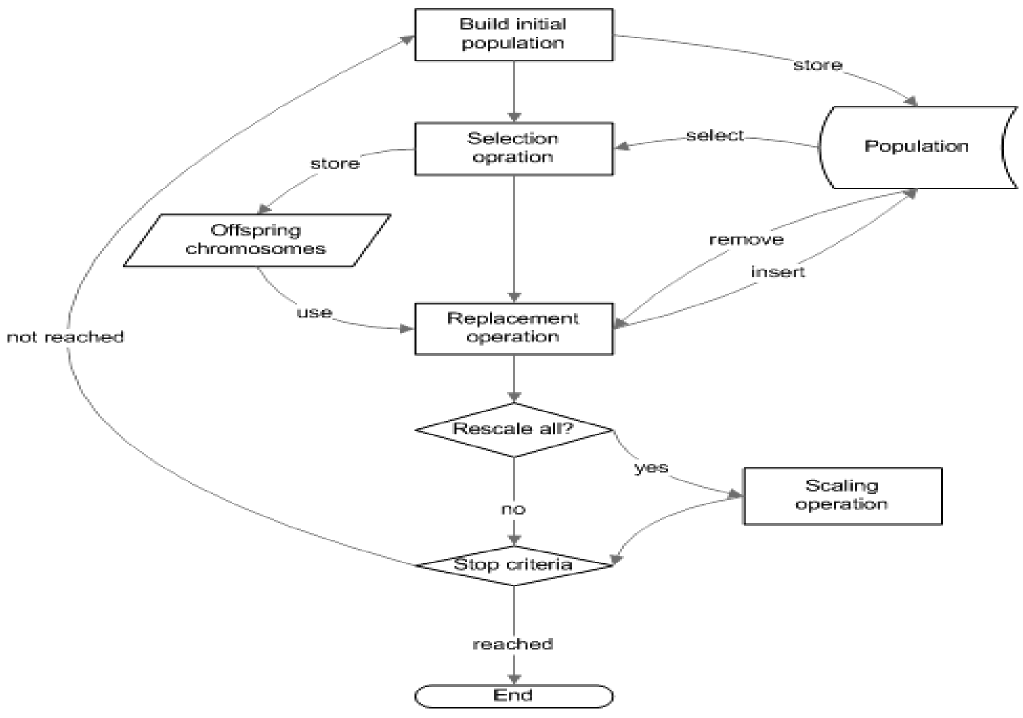


Fig. 3.3 – Genetic Algorithm Modular Approach [3]

- The proposed algorithm is implemented and integrated using genetic algorithm for optimization of the results including the cost and performance factor.
- The proposed system is generating efficient results in terms of the optimal solution when executed using genetic algorithm.
- The proposed technique is efficient also in terms of the execution and turnaround time despite of the number of iterations
- The limitations may be included regarding the proposed work in terms of its further enhancement using assorted metaheuristics.
- The proposed system gives better results in executed using genetic algorithm that is one of the prominent metaheuristic techniques.

PROPOSED ALGORITHM

1. *To design the matrices and blocks of the computation cost as well as communication.*
2. *Assignment of priorities to tasks.*
3. *Calculate EST=0 & EFT=0 of task 1 on each processor, set process avail time for p1,p2,p3=0. Assign processor to task1 on which EFT is minimum either p1=EFT (T1or task1) or p2=EFT (T1) or P3=EFT(T1)*
4. *Activation of the Genetic Algorithm and its aspects*
    a. *for t2 to tn repeat above*
5. *Calculate if (parent of t1 executed on p1)*
6. *EST (tn) on p1=max (parent executed)*
    a. *p2=max (parent completion+Communication cost), processor avail time)*
7. *EST (tn) on p3= same as above*
8. *EFT=EST+Computation cost*
9. *Effective Comparison between the classical approach and proposed Approach*

Chapter-4

IMPLEMENTATION AND RESULTS

4.1 Methodology Used

- Collection of Training or Process Data Set for Deep Analysis
- Implementation of the Computation Cost Matrix
- Applying Genetic Algorithm on the Multiprocessor Scheduling
- Applying proposed model on Training data set for effective results
- Fetch Results
- Data Interpretation

In the complete implementation and research task, following aspects and reference is used regarding classical and proposed approach

4.2 Results

4.2.1 Implementation of Proposed Algorithm

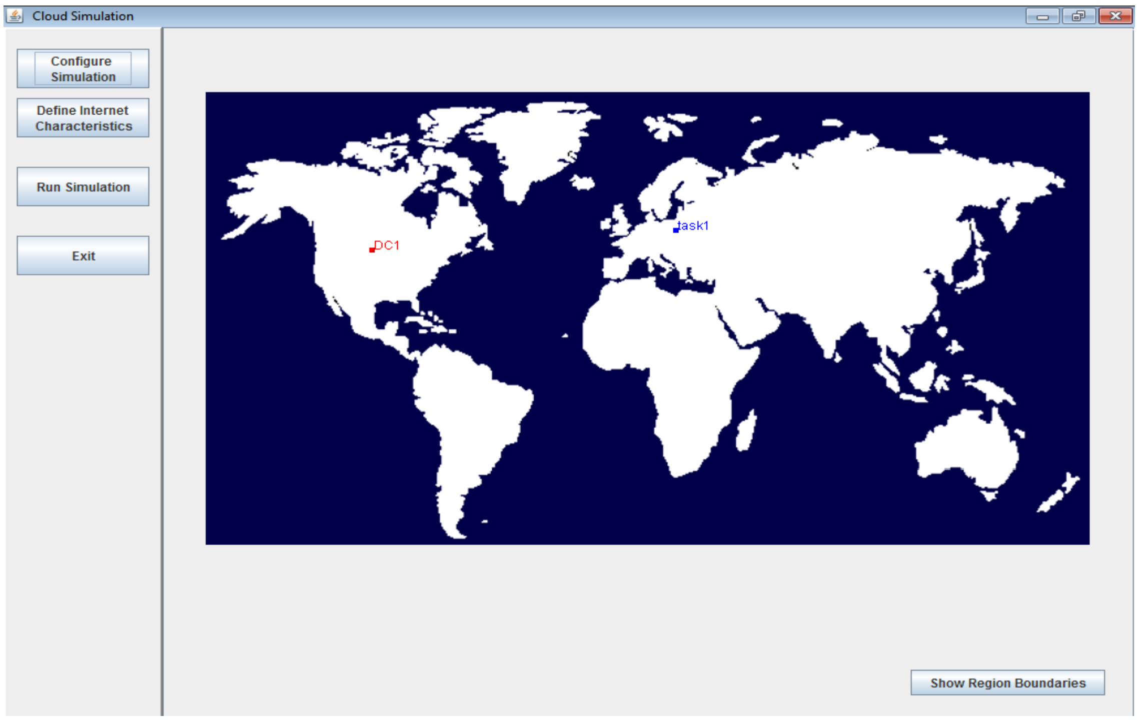
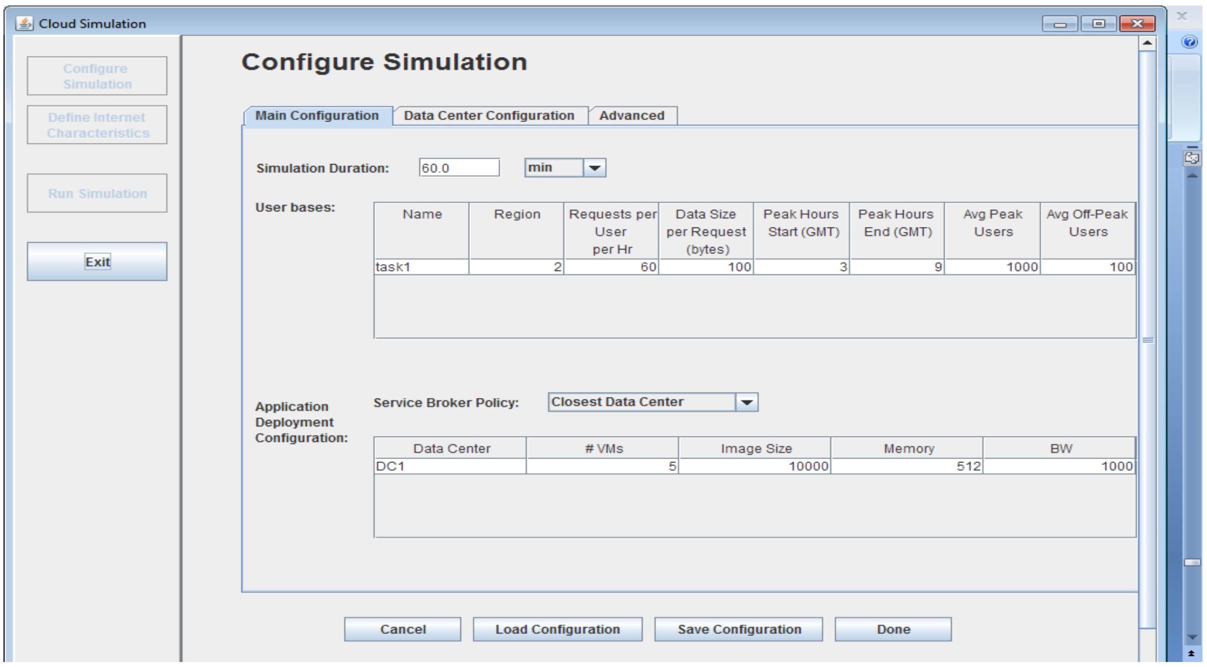


Fig. 4.1 – Simulation Scenario



Configure Simulation

Simulation Duration: 60.0 min

User bases:

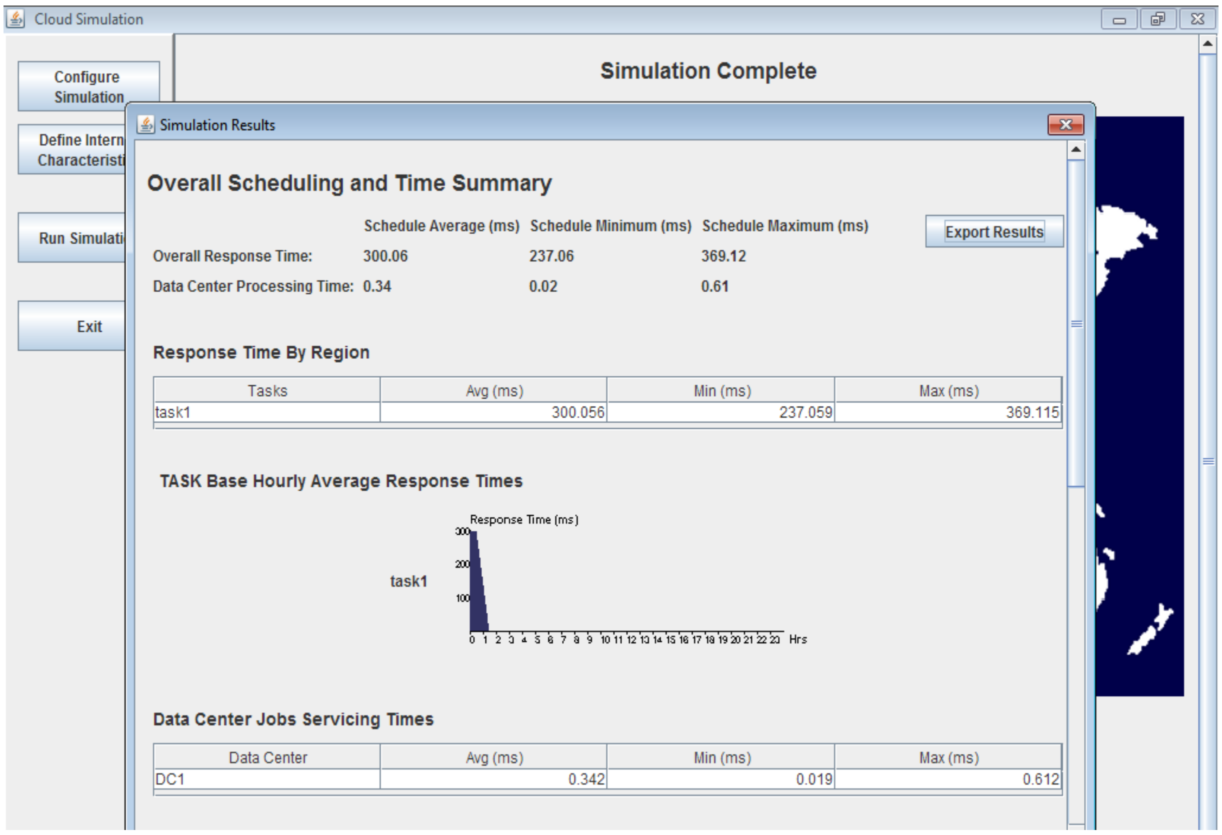
Name	Region	Requests per User per Hr	Data Size per Request (bytes)	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg Peak Users	Avg Off-Peak Users
task1	2	60	100	3	9	1000	100

Application Deployment Configuration:

Service Broker Policy: Closest Data Center

Data Center	# VMs	Image Size	Memory	BW
DC1	5	10000	512	1000

Fig. 4.2 – Simulation Configuration Panel



Simulation Complete

Overall Scheduling and Time Summary

	Schedule Average (ms)	Schedule Minimum (ms)	Schedule Maximum (ms)
Overall Response Time:	300.06	237.06	369.12
Data Center Processing Time:	0.34	0.02	0.61

Response Time By Region

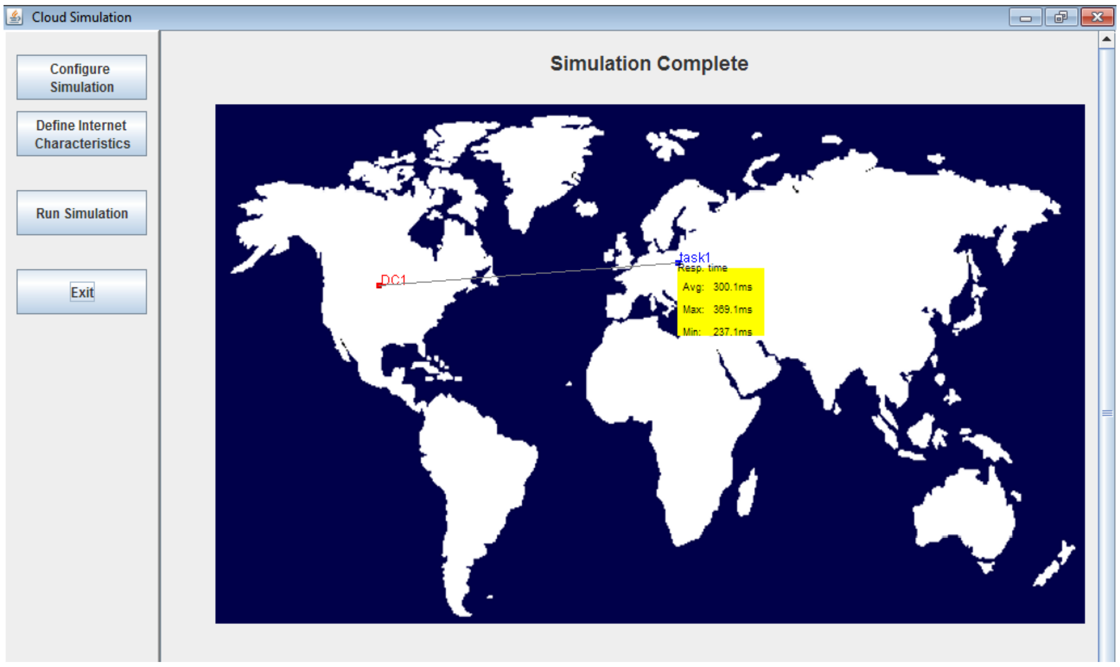
Tasks	Avg (ms)	Min (ms)	Max (ms)
task1	300.056	237.059	369.115

TASK Base Hourly Average Response Times

Data Center Jobs Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.342	0.019	0.612

Fig. 4.3 – Simulation Completion Status



Resp. time
Avg: 300.1ms
Max: 369.1ms
Min: 237.1ms

Figure 4.4 – Simulation Completion Status

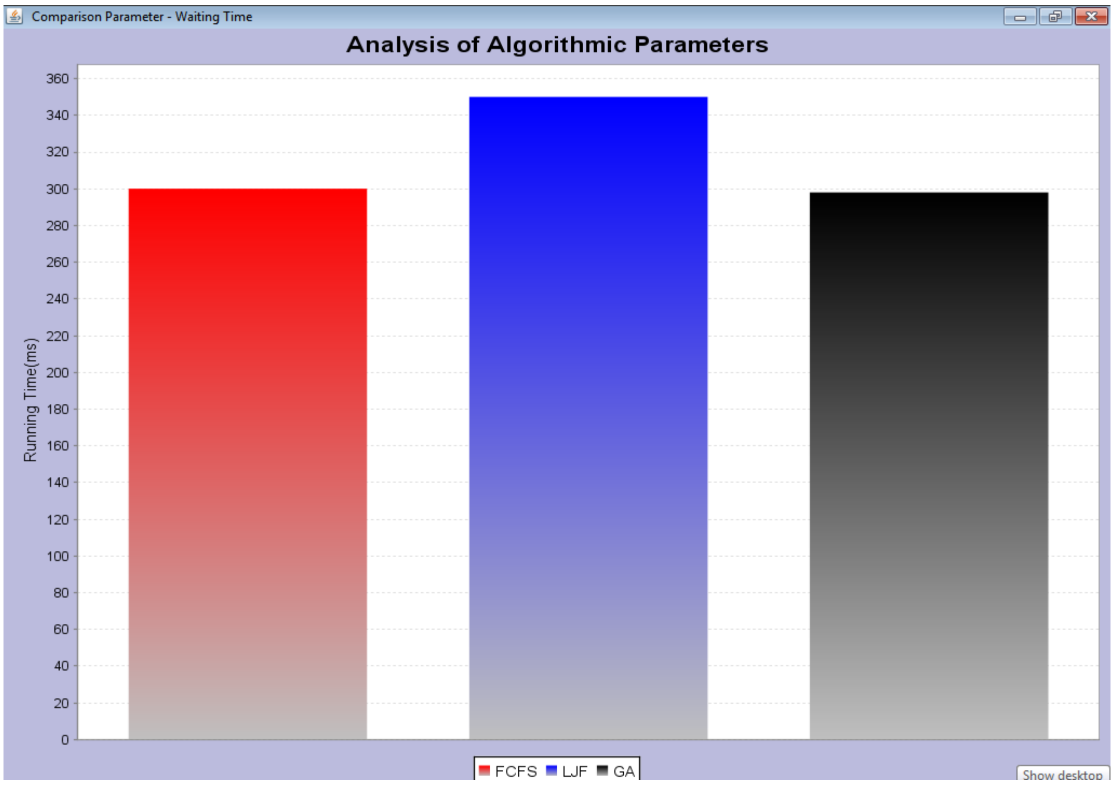


Fig. 4.5 – Comparison between classical and proposed approach in terms of Waiting Time

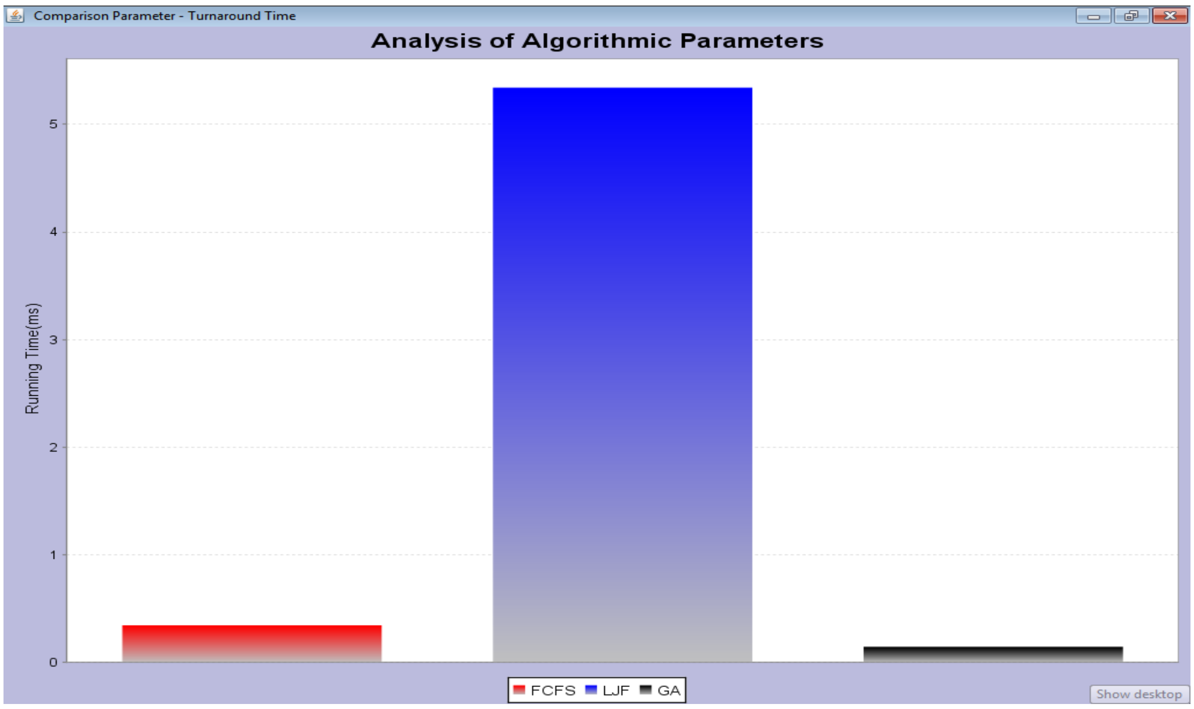


Fig. 4.6 – Comparison between classical and proposed approach in terms of Turnaround Time

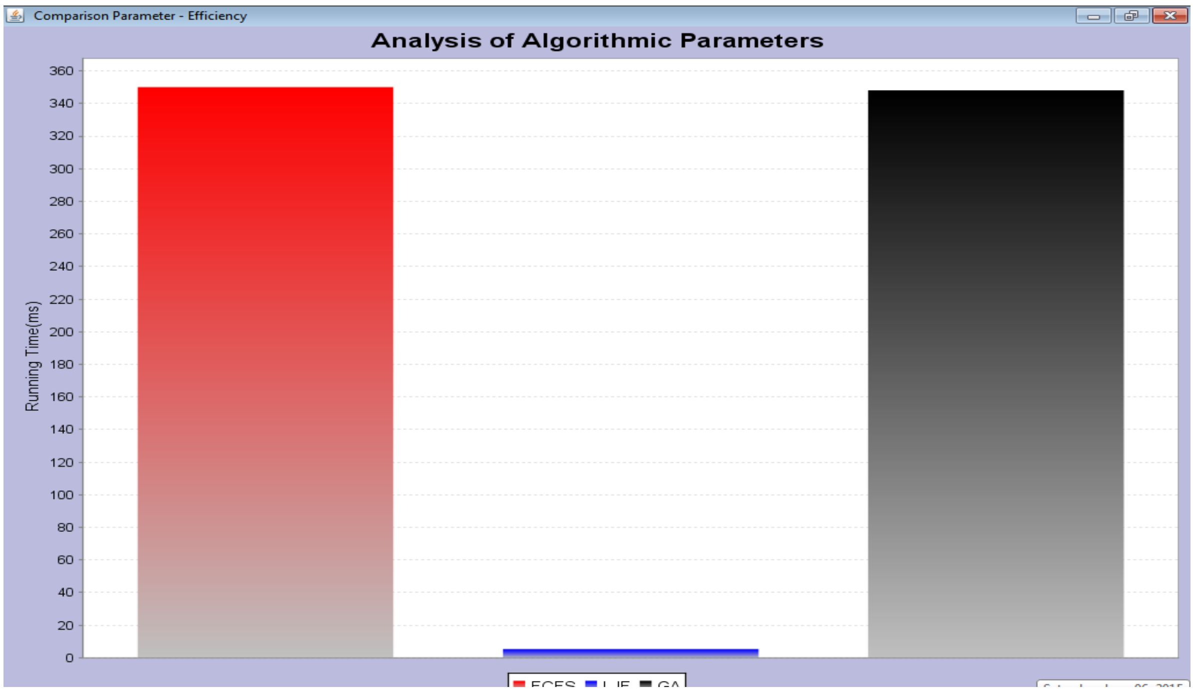


Fig. 4.7 – Comparison between classical and proposed approach in terms of Efficiency

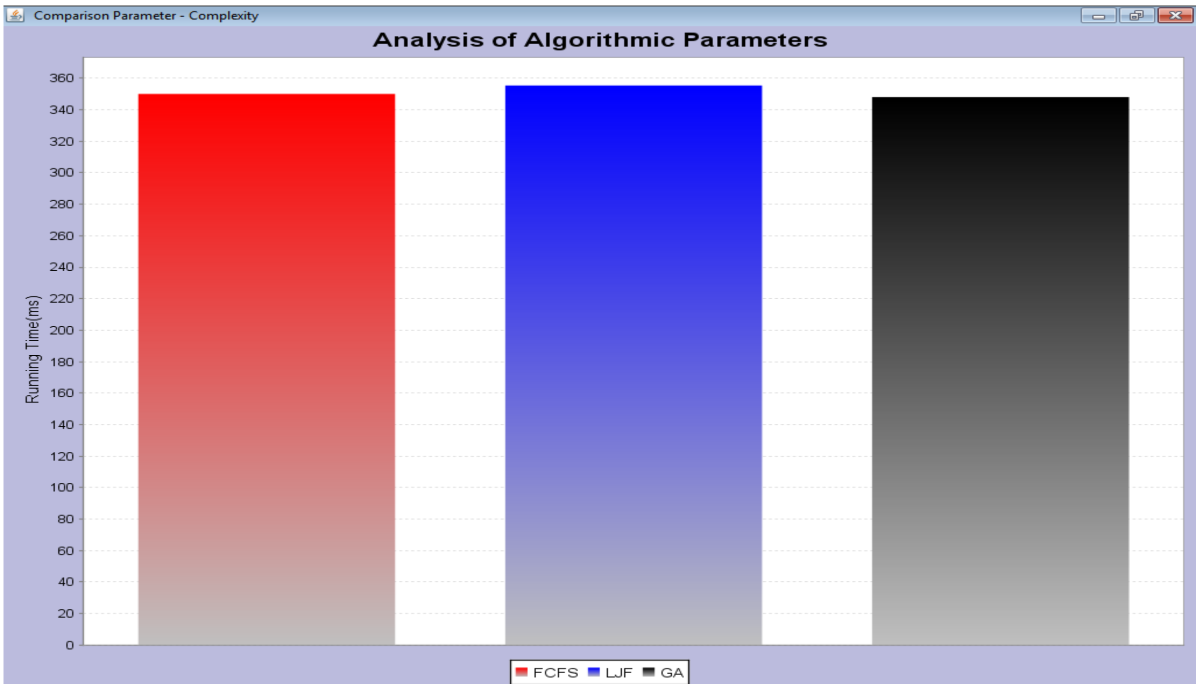


Fig. 4.8 – Comparison between classical and proposed approach in terms of Complexity

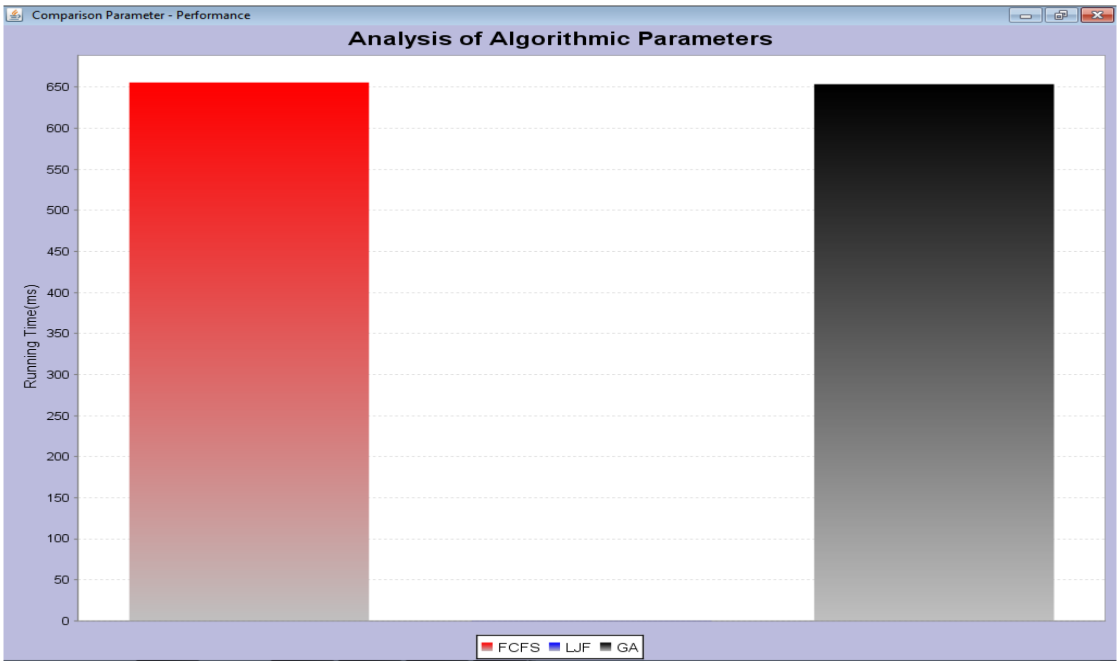


Fig. 4.9 – Comparison between classical and proposed approach in terms of Performance

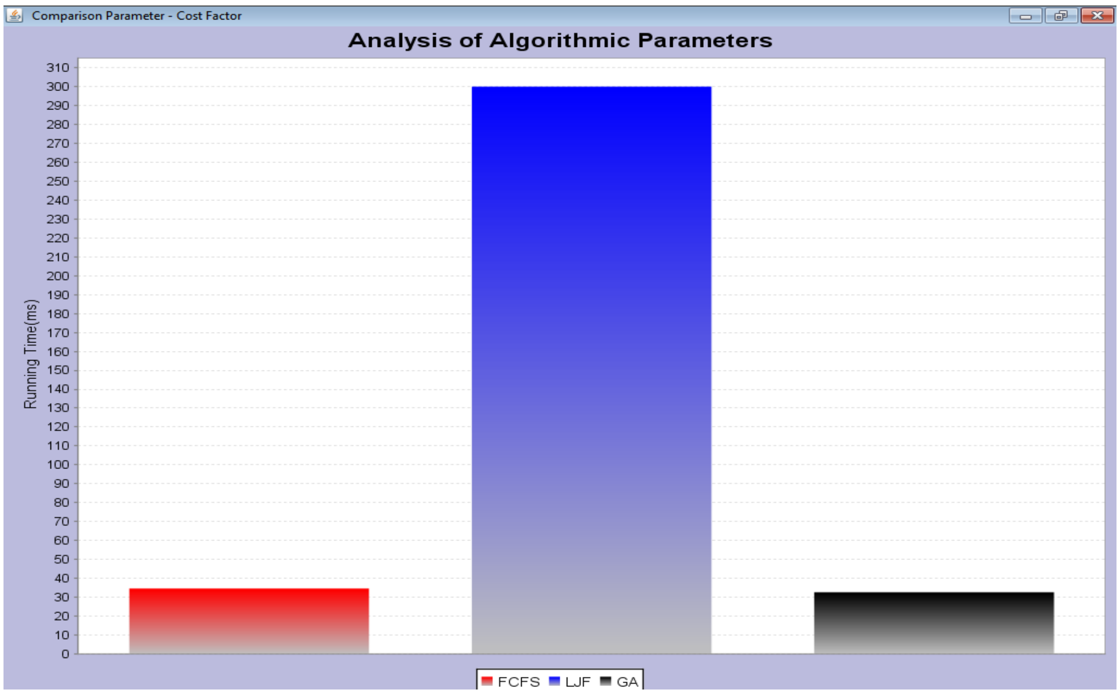


Fig. 4.10 – Comparison between classical and proposed approach in terms of Cost Factor

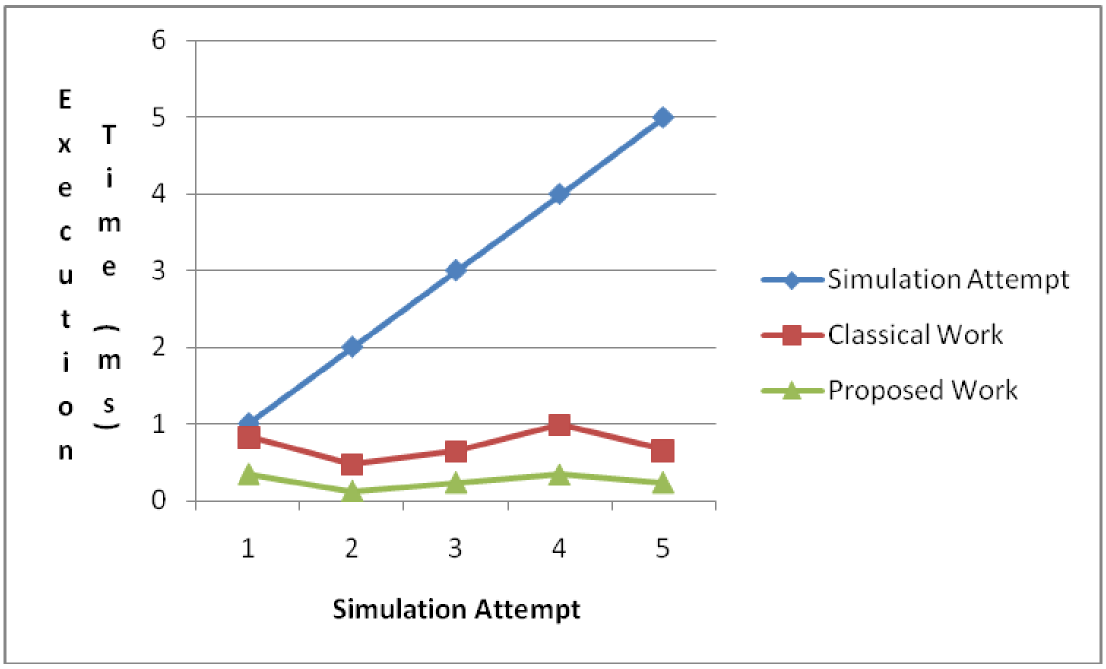


Figure 4.11 - Line Graph Analysis of the Classical and Proposed Approach

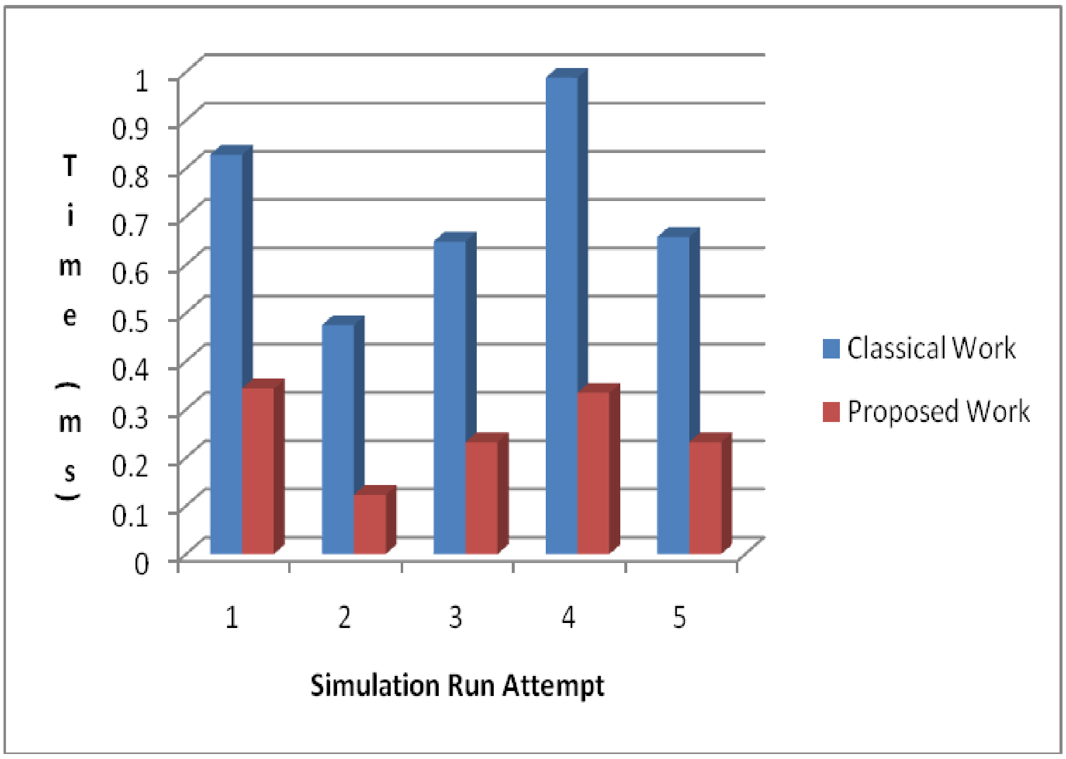


Figure 4.12 –Bar Graph Analysis of the Classical and Proposed Approach

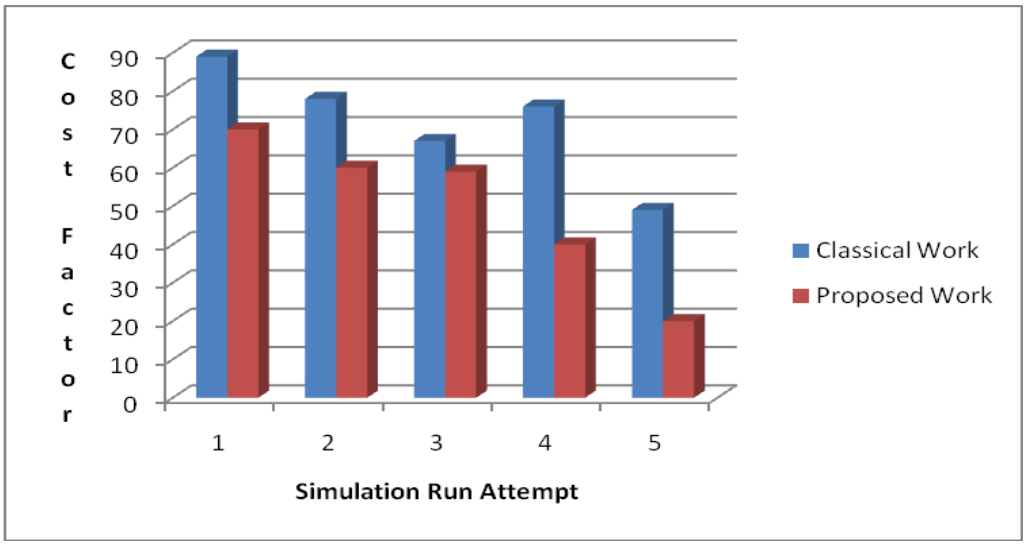


Figure 4.13 – Cost Factor Graph Analysis of the Classical and Proposed Approach

Simulation Attempt	Classical Work	Proposed Work
1	89	70
2	78	60
3	67	59
4	76	40
5	49	20

Comparison between classical and proposed approach in terms of Cost Factor

Classical / Existing Approach is not having effectiveness and efficiency as compared to GA based approach. The classical work is taken as the implementation without integration of metaheuristic based simulation.

Data Centers - 10

Cloudlets – 20 in each module

Tasks executed without GA and then with the integration of GA to evaluate the efficiency and related cost factor.

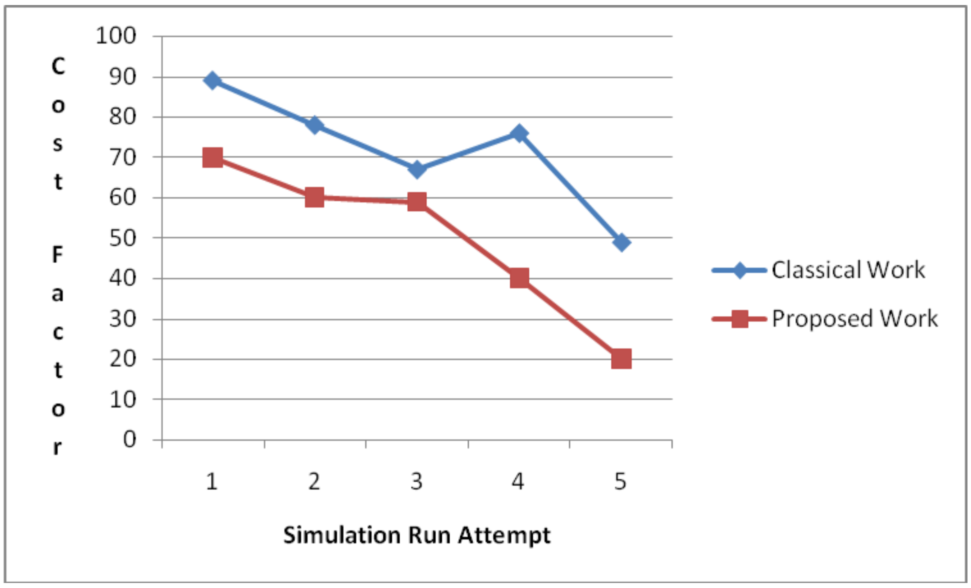


Figure 4.14 – Cost Factor Line Graph Analysis of the Approaches

It is evident from the graphical results that the cost factor in the proposed research approach that is very less when compared to the existing algorithmic approach. The execution time in the classical work is taking higher units as compared to the proposed work.

Data Centers - 5

Cloudlets - 40 in each module

Tasks executed without GA and then with the integration of GA to evaluate the efficiency and related cost factor.

Table 4.1 – Classical and Improved Approach

Existing Base Work	Proposed Approach
50	70

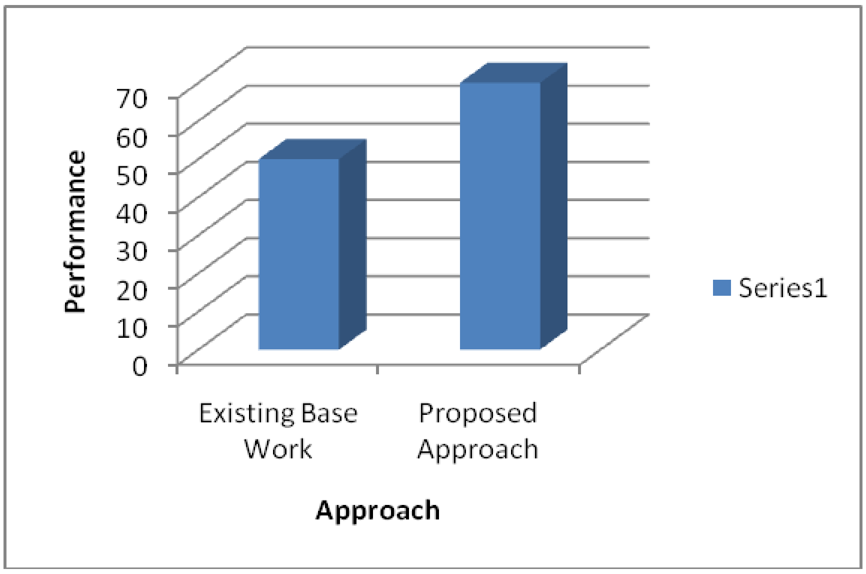


Figure 4.15 – Classical and Proposed approach

Data Centers - 20

Cloudlets – 50 in each module

Tasks executed without GA and then with the integration of GA to evaluate the efficiency and related cost factor.

Table 4.2 - Classical and Improved Approach

Existing Base Work	Proposed Approach
90	60

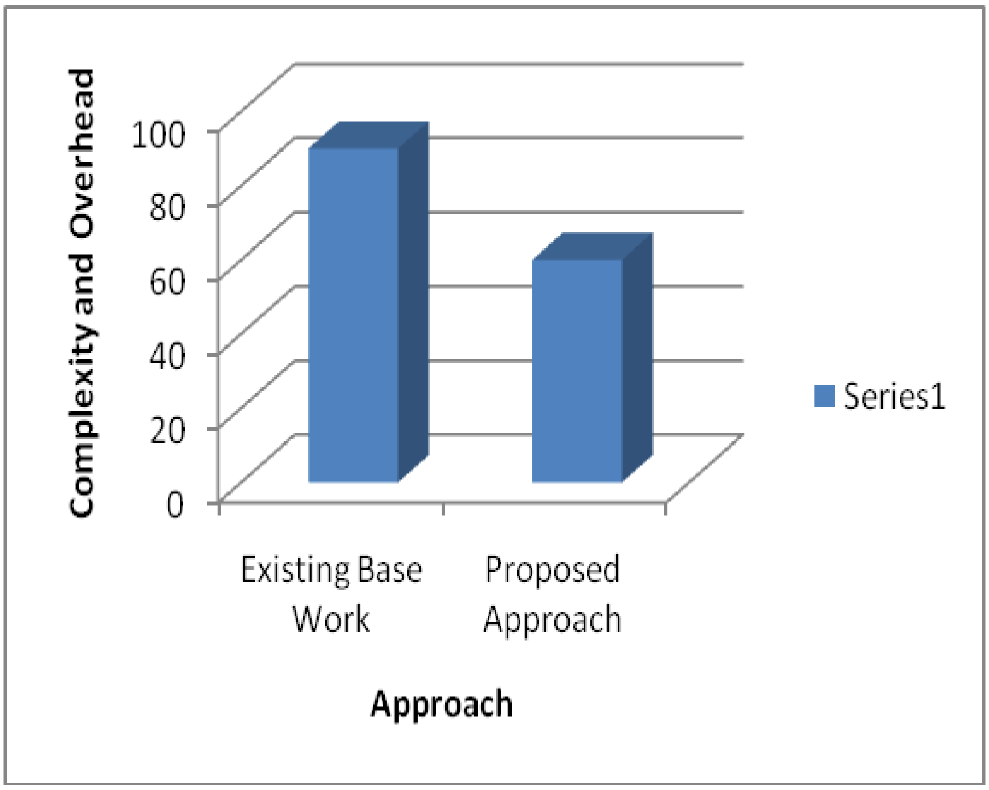


Figure 4.16 – Comparison of Classical and Proposed approach

Data Centers - 2

Cloudlets – 20 in each module

Tasks executed without GA and then with the integration of GA to evaluate the efficiency and related cost factor.

Table 4.3 – Difference between Classical and Improved Approach

Existing Base Work (Overall Effectiveness)	Proposed Approach (Overall Effectiveness)
50	60
60	88
70	89
50	69

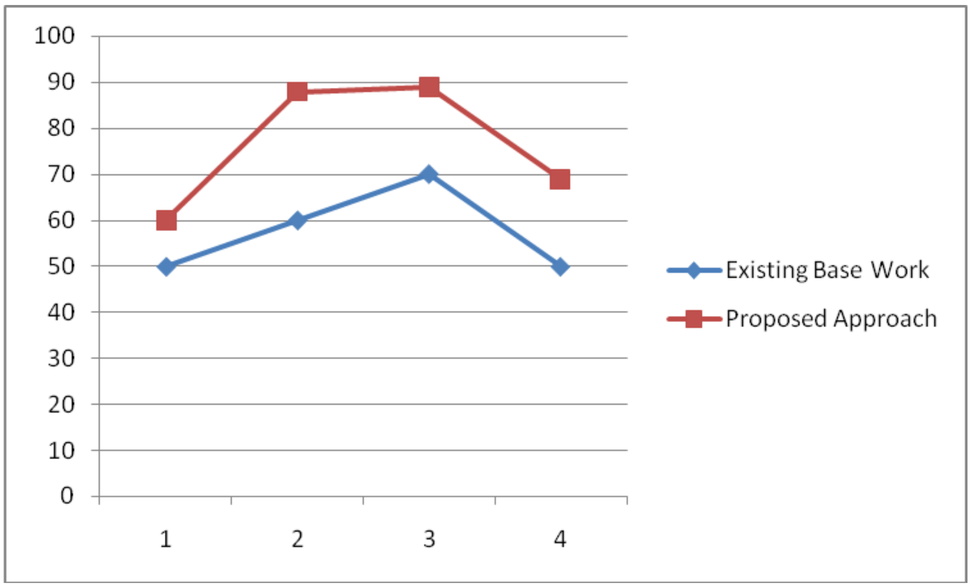


Figure 4.17 – Effective Comparison of Classical and Proposed Algorithm

Table 4.4 - Tabular Comparison of the Results Obtained

	W	T	E	P	C	CF
FCFS	10	50	75	85	690	35
LJF	14	550	20	40	20	300
EDSRTF	8	20	70	87	680	20
GA	7	18	95	96	670	18

It is evident from the simulation results and Table 1 that the cumulative result based on all the parameters are effective and better in the proposed approach name GA.

W – Waiting Time

The amount of time a process has been waiting in the ready queue in the process of execution.

T – Turnaround Time

Amount of the time that is taken to complete a specific process.

E – Efficiency

The numbers of processes which completes its execution or processor / time.

P – Performance

Performance (P) is directly associated with the degree of Efficiency (e)

C – Complexity

Complexity of a structured program is defined with reference to the control flow graph of the program, a directed graph containing the basic blocks of the program, with an edge between two basic blocks if control may pass from the first to the second.

The complexity C is then defined as

$C = E - N + 2P,$

where

E = the number of edges of the graph.

N = the number of nodes of the graph.

P = the number of connected components.

$C = E - N + P.$

CF – Cost Factor

r => n * (1/t) * rnd

n -> Length of the Input

t -> Execution Time

rnd -> Random Fuzzy Random

c = 1/(r * (p + e)) * 100

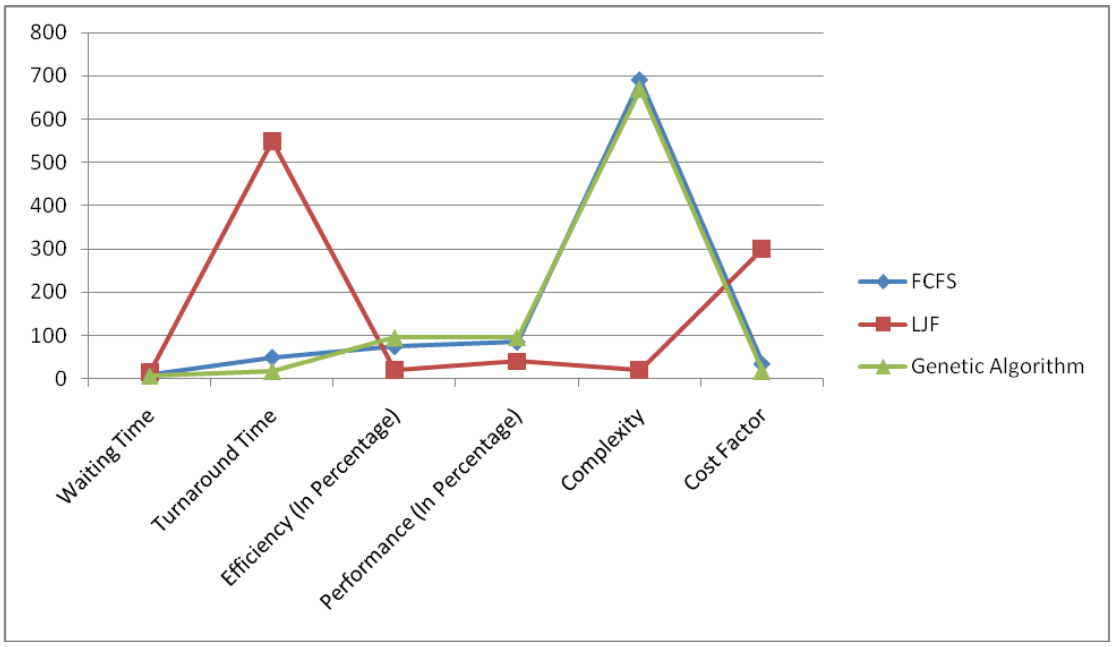


Figure 4.18 – Detailed Analysis of the Algorithm

The Classical or Existing Approach is not having effectiveness and efficiency as compared to GA based approach. The classical work is taken as the implementation without integration of metaheuristic based simulation.

Chapter-5

CONCLUSION AND FUTURE WORK

The existing problems in the multiprocessor scheduling have been removed using genetic algorithm and optimal results has been obtained. The further work in this area can be improved by using the other metaheuristics including ant colony optimization, simulated annealing, and honeybee algorithm. These algorithms are very prominent in terms of solving the combinatorial optimization problems. The multiprocessor scheduling algorithm can be passed with assorted swarm intelligence techniques to get the multiple results and from which the optimal result be obtained.

As a result of the advances in the innovation, the issues and further extension in science and building are turning out to be more convoluted than any time in recent memory. To tackle these muddled issues, network figuring turns into a well known apparatus. A framework domain gathers, coordinates, and uses heterogeneous or homogeneous assets scattered the world over by a rapid system. A lattice situation can be ordered into tw sorts: processing networks and information frameworks. This exploration work concentrates on employment booking calculations and their execution on numerous parameters in the lattice environment. In processing matrix, occupation booking is a critical undertaking. A decent booking calculation can allocate occupations to assets productively and can adjust the framework load.

For future scope of the work, following techniques can be used in hybrid approach to better and efficient results –

- Particle Swarm Optimization
- HoneyBee Algorithm
- Simulated Annealing
- Genetic Algorithmic Approaches

REFERENCES / BIBLIOGRAPHY

[1] Kune, R., Konugurthi, P. K., Agarwal, A., Chillarige, R. R., & Buyya, R. (2014, December). Genetic Algorithm based Data-aware Group Scheduling for Big Data Clouds. In Proceedings of the 2014 IEEE/ACM International Symposium on Big Data Computing (pp. 96-104). IEEE Computer Society.

[2] Mathew, T., Sekaran, K. C., & Jose, J. (2014, September). Study and analysis of various task scheduling algorithms in the cloud computing environment. In Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on (pp. 658-664). IEEE.

[3] Singh, S., & Kalra, M. (2014, November). Scheduling of Independent Tasks in Cloud Computing Using Modified Genetic Algorithm. In Computational Intelligence and Communication Networks (CICN), 2014 International Conference on (pp. 565-569). IEEE.

[4] Verma, A., & Kaushal, S. (2014). Deadline constraint heuristic-based genetic algorithm for workflow scheduling in cloud. International Journal of Grid and Utility Computing, 5(2), 96-106.

[5] Javanmardi, S., Shojafar, M., Amendola, D., Cordeschi, N., Liu, H., & Abraham, A. (2014, January). Hybrid job scheduling algorithm for cloud computing environment. In Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014 (pp. 43-52). Springer International Publishing.

[6] Rodriguez, M. A., & Buyya, R. (2014). Deadline based resource provisioningand scheduling algorithm for scientific workflows on clouds.Cloud Computing, IEEE Transactions on, 2(2), 222-235.

[7] Singh, L., & Singh, S. (2014). A Genetic Algorithm for Scheduling Workflow Applications in Unreliable Cloud Environment. In Recent Trends in Computer Networks and Distributed Systems Security (pp. 139-150). Springer Berlin Heidelberg.

[8] Shojafar, M., Javanmardi, S., Abolfazli, S., & Cordeschi, N. (2015). FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method. Cluster Computing, 18(2), 829-844.

[9] Intisar A.Majied Al-Said et. al. (2015). Evolution of grid computing architecture and grid adoption models. *IBM Systems Journal*, *43*(4), 624-645.

[10] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A. & Warfield, A. (2003). Xen and the art of virtualization. ACM SIGOPS Operating Systems Review, 37(5), 164-177.

[11] Al-Fares, M., Loukissas, A., & Vahdat, A. (2008). A scalable, commodity data center network architecture. ACM SIGCOMM Computer Communication Review, 38(4), 63-74.

[12] Robinson, J., Sinton, S., & Rahmat-Samii, Y. (2002). Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna. In Antennas and Propagation Society International Symposium, 2002. IEEE (Vol. 1, pp. 314-317). IEEE.

[13] Sailer, R., Valdez, E., Jaeger, T., Perez, R., Van Doorn, L., Griffin, J. L., ... & Berger, G. S. (2005). sHype: Secure hypervisor approach to trusted virtualized systems. *Techn. Rep. RC23511*.

[14] Gu, J., Hu, J., Zhao, T., & Sun, G. (2012). A new resource scheduling strategy based on genetic algorithm in cloud computing environment. Journal of Computers, 7(1), 42-52.

[15] Pandey, S., Wu, L., Guru, S. M., & Buyya, R. (2010, April). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on (pp. 400-407). IEEE.

[16] Kune, R., Konugurthi, P. K., Agarwal, A., Chillarige, R. R., & Buyya, R. (2014, December). Genetic Algorithm based Data-aware Group Scheduling for Big Data Clouds. In Proceedings of the 2014 IEEE/ACM International Symposium on Big Data Computing (pp. 96-104). IEEE Computer Society.

[17] Verma, A., & Kaushal, S. (2014). Deadline constraint heuristic-based genetic algorithm for workflow scheduling in cloud. International Journal of Grid and Utility Computing, 5(2), 96-106.

[18] Javanmardi, S., Shojafar, M., Amendola, D., Cordeschi, N., Liu, H., & Abraham, A. (2014, January). Hybrid job scheduling algorithm for cloud computing environment. In Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014 (pp. 43-52). Springer International Publishing.

[19] Ye, H. (2015). Research on Emergency Resource Scheduling in Smart City based on HPSO Algorithm. city, 5, 6.

[20] Pawar, A., Scholar, M. T., & Kapgate, P. D. (2014). A Review on Virtual Machine Scheduling in Cloud Computing. vol, 3, 928-933.

[21] Shojafar, M., Javanmardi, S., Abolfazli, S., & Cordeschi, N. (2015). FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method. Cluster Computing, 18(2), 829-844.

[22] Zhang, F., Cao, J., Li, K., Khan, S. U., & Hwang, K. (2014). Multi-objective scheduling of many tasks in cloud platforms. Future Generation Computer Systems, 37, 309-320.

[23] Quang-Hung, N., Tan, L. T., Phat, C. T., & Thoai, N. (2014). A GPU-Based Enhanced Genetic Algorithm for Power-Aware Task Scheduling Problem in HPC Cloud. In Information and Communication Technology (pp. 159-169). Springer Berlin Heidelberg.

[24] Tsai, C. W., & Rodrigues, J. J. (2014). Metaheuristic scheduling for cloud: A survey. Systems Journal, IEEE, 8(1), 279-291.

[25] Lin, W., Liang, C., Wang, J. Z., & Buyya, R. (2014). Bandwidth-aware divisible task scheduling for cloud computing. Software: Practice and Experience,44(2), 163-174.

[26] Kumar, M., & Doegar, A. (2014). Reliable and Efficient Task Scheduling based on Genetic Algorithm in Cloud Computing Environment.

[27] Frîncu, M. E. (2014). Scheduling highly available applications on cloud environments. Future Generation Computer Systems, 32, 138-153.

[28] Yang, T., & Gerasoulis, A. (2014, June). Author retrospective for PYRROS: static task scheduling and code generation for message passing multiprocessors. In 25th Anniversary International Conference on Supercomputing Anniversary Volume (pp. 18-20). ACM.

[29] Leena, V. A., & Rajasree, M. S. (2016). Genetic Algorithm Based Bi-Objective Task Scheduling in Hybrid Cloud Platform. International Journal of Computer Theory and Engineering, 8(1),

[30] Xu, Y., Li, K., Hu, J., & Li, K. (2014). A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues.Information Sciences, 270, 255-287.

[31] Lin, J., Zhong, Y., Lin, X., Lin, H., & Zeng, Q. (2014). Hybrid Ant Colony Algorithm Clonal Selection in the Application of the Cloud's Resource Scheduling. arXiv preprint arXiv:1411.2528.

[32] Zdenek Konfrst, " Parallel Genetic Algorithms: Advances, Computing Trends, Applications and Perspectives", 18th International Parallel and Distributed Processing, 2004.

[33] Marin Golub, Leo Budin, "An Asynchronous Model of Global Parallel Genetic Algorithms" Unska 3, HR-10000 Zagreb, Croatia

[34] Multiprocessor booking Algorithm Based On Genetic Algorithm (Intisar A.Majied Al-Said, Nedhal Al-Saiyd, Firas Turki Attia)

[35] Multiprocessor Environment Using Genetic Algorithm (Volume 2, Issue 5, May 2012 ISSN: 2277 128X, Sandeep Jain, Shweta Makkar

[36] DMP: Deterministic Shared Memory Multiprocessing (Joseph Devietti Brandon Lucia Luis Ceze Mark Oskin Computer Science & Engineering, University of Washington)