# A COST-EFFECTIVE RESOURCE MAINTAINANCE OF DATA DISTRIBUTION IN EDGE COMPUTING

**A PROJECT REPORT**

***Submitted by***


**BALAMURUGAN C (1813006)**

**BOOBALAN N (1813010)**

**GURU VIGNESH K (1813023)**


***in partial fulfilment for the award of the degree***

***of***

**BACHELOR OF ENGINNERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**


**K.S.R. COLLEGE OF ENGINEERING**

**(An Autonomous Institution, Affiliated to Anna University Chennai and Approved by AICTE)**

**TIRUCHENGODE– 637 215**


**JUNE 2022**

# K.S.R.COLLEGE OF ENGINEERING

# TIRUCHENGODE-637215

# ANNA UNIVERSITY, CHENNAI- 600025

## BONAFIDE CERTIFICATE

Certified that this project report "**A COST-EFFECTIVE RESOURCE MAINTAINANCE OF DATA DISTRIBUTION IN EDGE COMPUTING"** is the bonafide work of **"BALAMURUGAN C (1813006),BOOBALAN N (1813010),GURU VIGNESH K (1813023)"** who carried out the project work under my supervision.

| | |
|---|---|
| **SIGNATURE** | **SIGNATURE** |
| **Dr.A.RAJIV KANNAN M.E., Ph.D.,** | **Dr.N.S.NITHYA M.E., Ph.D.,** |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| **Professor** | **Professor** |
| Department of CSE, | Department of CSE, |
| K.S.R. College of Engineering | K.S.R College of Engineering |
| Tiruchengode-637215. | Tiruchengode-637215. |

.......................................................................................................................................................

Submitted for Project viva-voce held on .........................................................................

Internal Examiner External Examiner

# ACKNOWLEDGEMENT

We feel highly honoured to extend our sincere gratitude to our beloved Founder cum Chairman **Lion Dr. K.S. RANGASAMY MJF.,** K.S.R. Educational Institutions and our Chairman **Mr. R. SRINIVASAN BBM., MISTE.,** Aarthi Educational and charitable Trust for providing all facilities to complete this project work.

We would like to acknowledge the constant and kind support provided by our principal **Dr. P. SENTHILKUMAR M.E., Ph.D., (IITM),** who supported us in all the endeavours and been responsible for inculcating us all through our career.

We feel highly elated to thank our respectable Head of the Department **Dr. A. RAJIV KANNAN M.E., Ph.D., MISTE., MCSI.,** who guided us and was a pillar of support for the successful completion of the project.

We are thankful to our Project Coordinator **Dr. M. SOMU M.E., Ph.D.,** of our department for his valuable suggestions and guidance to our project.

We are the most fortunate in having the opportunity to work under the guide **Dr.N.S.NITHYA M.E.,PhD.,** express our sincere thanks to her. This project has brought out the hidden talent within us.

It is a pleasure to express our gratefulness to our beloved parents for providing their support and confidence to us for the completion of the project and our heartfelt thanks to our entire department faculty members, beloved friends, directly and indirectly who helped us during the tenure of the project.

**Temenos India Pvt. Ltd.**
OMR Bypass Road, MGR Salai,
Perungudi, Chennai

## Certificate for Excellence

This is to certify that **BALAMURUGAN C (1813006)**,**BOOBALAN N (1813010)** and **GURU VIGNESH K (1813023)** while pursuing their final year of B.E in Computer Science And Engineering at K.S.R College Of Engineering, Tiruchengode has successfully completed their academic project with the title "**A COST EFFECTIVE RESOURCE MAINTAINANCE OF DATA DISTRIBUTION IN EDGE COMPUTING**". During the period of November 2021 to February 2022(three months).We are really happy to work with these students and their performance was good. Wishing them the best for every step in their journey.

With Regards

Executive Manager

Temenos India Pvt. Ltd

# ABSTRACT

Cloud computing is rapidly growing and many more cloud providers are emerging. Cost efficiency and resource cost maximization become two major concerns of cloud providers to remain competitive while making profit. The profit maximization problem in federated cloud environments cooperate to increase the degree of multiplexing has been investigated. Outline novel economics-inspired resource allocation mechanisms to tackle the profit maximization problem from the perspective of a cloud provider acting solely. Existing abstractions for in-memory storage on clusters, such as distributed shared memory, key value stores, databases, and Piccolo, offer an interface based on fine-grained updates to mutable state (e.g., cells in a table). Itis fine-tuned to predict the load of its cluster. The final load of the whole grid is obtained by summing the loads of each cluster. The proposed method for load forecasting in Smart Grid has two major advantages. 1) Learning customer behaviours not only improves the prediction accuracy but also has a low computational cost. 2) SCCRF can effectively model the load forecasting problem of one customer, and simultaneously select key features to identify its energy consumption pattern.With this interface, the only ways to provide fault tolerance are to replicate the data across machines or to log updates across machines. Cost effective resource allocation based on following strategies are Cost Efficiency of the Cloud: Cost reductions and profit increases, Pay-as-you- go pricing, Implications of multi tenancy. Scheduling and resource allocation as a cost efficient solution Exploitation of application characteristics, explicit consideration of user experience/satisfaction. Admission control mechanisms tailored within a Profit management framework to maximize resource cost has been proposed.

## TABLE OF THE CONTENTS

## LIST OF TABLES

# LIST OF FIGURES

## LIST OF ABBREVIATIONS

| ACRONYMS | | ABBREVIATIONS |
|---|---|---|
| **CWC** | - | Central Warehousing Corporation |
| **DAG** | - | Directed Acyclic Graph |
| **DHI** | - | Door and Hardware Institute |
| **EC2** | - | Elastic Compute Cloud |
| **GAIN** | - | Graphic Arts Information Network |
| **ISP** | - | Internet Service Provider |
| **NASA** | - | National Aeronautics and Space Administration |
| **PCP** | - | Partial Critical Paths |
| **QOS** | - | Quality of Service |
| **RDD** | - | Resilient Distributed Database Sets |
| **TOF** | - | Tetralogy of Fallot |
| **VM** | - | Virtual Machine |

# CHAPTER 1

## INTRODUCTION

### 1.1 CLOUD COMPUTING

Cloud computing providing unlimited infrastructure to store and execute customer data and program. Customers do not need to own the infrastructure, they are merely accessing or renting; they can forego capital expenditure and consume resources as a service, paying instead for what they use. Benefits of Cloud Computing: Minimized Capital expenditure. Location and Device independence. Utilization and efficiency improvement. Very high Scalability. High Computing power. Using a rich set of operators. The main challenge in designing RDDs is defining a programming interface that can provide fault tolerance efficiently. Existing abstractions for in-memory storage on clusters, such as distributed shared memory, key value stores, databases, and Piccolo, offer an interface based on fine grained updates to mutable state (e.g., cells in a table). The only ways to provide fault tolerance are to replicate the data across machines or to log updates across machines.

Both approaches are expensive for data-intensive workloads, require copying large amounts of data over the cluster network, whose bandwidth is far lower than that of RAM, and incur substantial storage overhead. RDDs provide an interface based on coarse-grained transformations (e.g., map, filter and join) apply the same operation to many data items. Allows to efficiently provide fault tolerance by logging the transformations used to build a dataset (its lineage) rather than the actual data. If a partition of an RDD is lost, the RDD has enough information about how it was derived from other RDDs to recompute.

- Security a major Concern

- Security concerns arising because both customer data and program are residing in Provider Premises.

- Security is always a major concern in Open System Architectures

Professional Security staff utilizing video surveillance, state of the art intrusion detection systems, and other electronic means. When an employee no longer has a business need to access data center his privileges to access datacentre should be immediately revoked. All physical and electronic access to data centers by employees should be logged and audited routinely. Audit tools so that users can easily determine how their data is stored, protected, used, and verify policy enforcement**.** Data should be stored and processed only in specific jurisdictions as define by user. Provider should also make a contractual commitment to obey local privacy requirements on behalf of their customers, data-centered policies that are generated when a user provides personal or sensitive information that travels with that information throughout its lifetime to ensure that the information is used only in accordance with the policy.

Data store in database of provider should be redundantly store in multiple physical location. Data that is generated during running of program on instances is all customer data and provider should not perform backups. Control of administrator on databases. Sanitization is the process of removing sensitive information from a storage device. What happens to data stored in a cloud computing environment once it has passed its user's "use by date". Data sanitization practices does the cloud computing service provider propose to implement for redundant and retiring data storage devices as and when these devices are retired or taken out of service.

Denial of Service: where servers and networks are brought down by a huge amount of network traffic and users are denied the access to a certain Internet based service.

Like DNS Hacking, Routing Table "Poisoning", XDOS attacks QOS Violation: through congestion, delaying or dropping packets, or through resource hacking. Man in the Middle Attack: To overcome it always use SSL.IP Spoofing: Spoofing is the creation of TCP/IP packets using somebody else's IP address. Solution: Infrastructure will not permit an instance to send traffic with a source IP or MAC address other than its own.

## 1.2 RESOURCE ALLOCATION COST OPTIMIZATION

Cloud computing has emerged as important computing technology and its pay-as you-go cost structure enabled the providers to offer computing service on demand and pay for the resources just as utility computing. The rapid evolution of the technology makes the resources more cost effective consumer driven technology. The cloud consumer's important challenge is to find the most efficient way to utilize the rented cloud resources. Virtualization is the important process which allows the sharing of computing resources in online. The computing resources are of different types. These includes Infrastructure as a service (Iaas) which provides the capability to the consumer to provision network, storage and processing.

It can include the operating system and applications. E.g., Amazon EC, Open Nebula, Eucalyptus. Platform as a service(Paas)provides the capability to the consumer to acquire applications created using programming languages, deploy onto the cloud infrastructure and tools supported by the provider. E.g., Hadoop, Microsoft Windows Azure, Google App Engine.

Software as a service (SaaS) provides the capability to the consumer to use the applications of the provider which runs on cloud infrastructure. E.g., Google

Apps, SalesForce.com, Eye OS. Cloud providers provides these resources on demand to the users. When there is any requirement for the users in the cloud, the cloud system provides the required resources to the users by creating virtual machines (VM) in the host machine. The tasks of the users are in the form of workflow. The workflow applications are executed by the workflow scheduling. The workflow scheduling is the process which needs to map the tasks on the resources for the execution process of the workflow. The effective scheduling results in improving the resource utilization, reduce capital expenditure and reduce initial investment.

Security related to the information exchanged between different hosts or between hosts and users. This issues pertaining to secure communication, authentication, and issues concerning single sign on and delegation. Secure communications issues include those security concerns that arise during the communication between two entities. These include confidentiality and integrity issues. Confidentiality indicates that all data sent by users should be accessible to only “legitimate” receivers, and integrity indicates that all data received should only be sent/modified by “legitimate” senders. Solution: public key encryption, X.509 certificates, and the Secure Sockets Layer (SSL) enables secure authentication and communication over computer networks.

## 1.3 GROUP AND REAL WORKFLOW OPTIMIZATION ON CLOUD

A workflow is a depiction of a sequence of operations, declared the work of a person, work of a simple or complex mechanism, work of a group of persons, work of an organization of staff, or machines. Workflow may be seen as any abstraction of real work, segregated in work share, work split or whatever types of ordering. For control purposes, workflow may be a view on real work under a chosen aspect, thus serving as a virtual representation of actual work. The flow described often refers to a document transferred from one step to another. A workflow is a model to represent real work for further assessment, e.g., for describing a reliably repeatable sequence of operations.

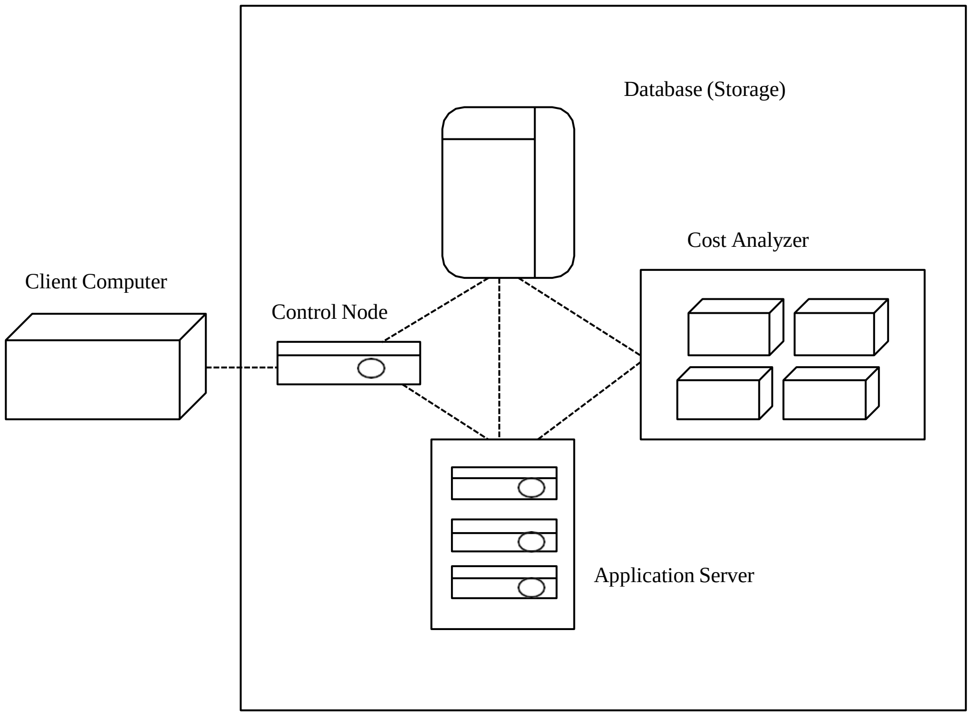



Fig 1.1: Information Security

More abstractly, a workflow is a pattern of activity enabled by a systematic organization of resources, defined roles and mass, energy and information flows, into a work process can be documented and learned. Workflows are designed to achieve processing intents of some sort, such as physical transformation, service provision, or information processing. An instance is a virtual machine offered by the cloud provider. Different types of instances can have different amount of resources such as CPUs and RAM and different capabilities such as CPU speed, I/O speed and network bandwidth.

A workflow can be represented by a directed graph represents data-flows connect loosely and tightly coupled (and often asynchronous) processing components. Monetary cost optimizations have been classic research topics in grid and cloud computing environments. Over grid computing, cost-aware optimization techniques have been extensively studied. Researchers have addressed various problems: minimizing cost given the performance requirements, maximizing the performance for given budgets and scheduling optimizations with both cost and performance constraints. Based on cloud computing, the pay-as-you-go pricing, virtualization and elasticity features of cloud computing open up various challenges and opportunities. For example, most cloud providers offer instance hour billing model. Partial-hour consumption is always rounded up to one hour. Some other billing models have been proposed (e.g., Google's IaaS service charges by minutes of use), hourly billing is still the most commonly adopted model.

Recently, there have been many studies on monetary cost optimizations with resource allocations and task scheduling according to the features of cloud computing Effectiveness in reducing the monetary cost, all of them assume static task execution time and consider only fixed pricing scheme (only on-demand instances in Amazon's terminology). Particularly, they have the following limitations. The resource allocation in cloud computing is much more complex than in other distributed systems like Grid computing platform. In a Grid system, it is improper to share the compute resources among the multiple applications simultaneously. Running atop it due to the inevitable mutual performance interference among them. Whereas, cloud systems usually do not provision physical hosts directly to users, but leverage virtual resources isolated by VM technology.

Elastic resource usage way adapt to user's specific demand, but also maximize resource utilization in fine granularity and isolate the abnormal environments for safety purpose. Some successful platforms or cloud management tools leveraging VM resource isolation technology include Amazon

EC2 and OpenNebula. Fast development of scientific research, users may propose quite complicated demands. For example, users may wish to minimize their payments when guaranteeing their service level such that their tasks can be finished before deadlines. Such a deadline-guaranteed resource allocation with minimized payment is rarely studied in literatures. Moreover, inevitable errors in predicting task workloads will definitely make the problem harder.

Based on the elastic resource usage model, we aim to design a resource allocation algorithm with high prediction error tolerance ability, also minimizing users' payments subject to their expected deadlines. Virtual machines (VMs) have emerged as the basis for allocating resources in enterprise settings and hosting centers. One benefit of VMs in these environments is the ability to multiplex several operating systems on hardware based on dynamically changing system characteristics. Such multiplexing must often be done while observing per-VM performance guarantees or service level agreements. Thus, one important requirement in this environment is effective performance isolation among VMs Resource provisioning in compute clouds often an estimate of the capacity needs of Virtual Machines (VMs).

The estimated VM size is the basis for allocating resources commensurate with demand. In contrast to the traditional practice of estimating the size of VMs individually, joint-VM provisioning approach in which multiple VMs are consolidated and provisioned together, based on an estimate of their aggregate capacity needs. New approach exploits statistical multiplexing among the workload patterns of multiple VMs, i.e., the peaks and valleys in oneworkload pattern do not necessarily coincide with the others, the unused resources of a low utilized VM can be borrowed by the other co-located VMs with high utilization. Compared to individual-VM based provisioning, joint-VM.

VMs obtain cycles and bandwidth on a demand-driven (best-effort) basis. Many hybrid approaches are also possible: for instance, a system may enforce fair sharing of resources between classes of VMs, which lets one overbook available resources while preventing starvation in overload scenarios. The key point is that both hypervisors and COSs incorporate sophisticated resource schedulers to avoid or minimize crosstalk. A general purpose desktop grid system must accommodate heterogeneous clusters of nodes running heterogeneous batches of jobs. The same computation may be performed on different input regions, such as n-body or weather calculations that differ only in spatial coordinates. The resource availability in Grids is generally unpredictable due to the autonomous and shared nature of the Grid resources and stochastic nature of the workload resulting in a best effort quality of service. The resource providers optimize for throughput and utilization whereas the users optimize for application performance.

Cost-based model where the providers advertise resource availability to the user community. Present a multi-objective genetic algorithm formulation for selecting the set of resources to be provisioned that optimizes the application performance while minimizing the resource costs, trace-based simulations to compare the application performance and cost using the provisioned and the best effort approach with a number of artificially generated work flow structured applications and a seismic hazard application from the earthquake science community The provisioned approach shows promising results when the resources are under high utilization and/or the applications have significant resource requirements.

## 1.4 TRANSFORMATION OPTIMIZATION FRAMEWORK

The transformation operations results in structural changes of the assignment of DAG. The transformation operations are classified as main schemes and auxiliary schemes. The main scheme aims to reduce the cost. The auxiliary schemes aim to change the form of workflow which is suitable for main scheme to reduce cost. The six basic workflow transformation operations are Merge, Demote, Split, Promote, Move and co-scheduling. The merge and demote operation comes under main scheme. The Split, Promote, Move and co-scheduling comes under the auxiliary scheme. Amazon EC2 provides different types of virtual machines (instances), each with different computational capabilities and prices. There are multiple pricing models in the cloud, such as on-demand, spot and reservation. Focus on the on-demand and spot pricing models in this paper.

Different from the on-demand pricing model where users pay a fixed price for unit time of instance usage, the spot price changes along time. To use spot instances, users need to bid the appropriate price they are willing to pay. The bid price is fixed once the instance is launched. If the bid price is higher than the spot price, the instance can be successfully launched and run; otherwise it waits. Amazon publishers update the spot price periodically and launch the waiting instances whose bid prices exceed the current spot price and terminate the instances whose bid prices are lower than that. Statistically analyzed the spot price history and found that, the spot price varies in both temporal and spatial dimensions and it is hard to predict the exact price in the future, the probabilistic distribution of the spot price is stable in a short time.

Spot price variance. The spot price has shown variances in both spatial and temporal dimensions, the spot price history of m1.medium and m1.large instance types in two Amazon EC2 availability zones. The spot price is not static, but changes along the time. The change of the spot price can be huge. The spot price of m1.medium instances in the us-east-1a zone increases from less than $0.1 to around $10 at the time of 10 hours, the variation of the spot prices is not constant. The spot price can be unchanged for some time (e.g., spot price of m1.medium in us-east-1a zone during 20 to 40 hours, highlighted with A changing dramatically for some other time (e.g., spot price of m1.medium in us-east-1a zone during 50 to 60 hours, highlighted with B. Thus, it is generally difficult or even impossible to predict the exact spot price, even in the very near future.

Spatial variation. On the spatial dimension, we have the following observations. The spot price variations of different instance types are different. For example, the spot price of m1.medium changes abruptly during 50 to 60 hours while the price of m1.large is unchanged, the spot price of a more powerful instance can be cheaper than a less powerful instance type at some time (e.g., m1.large and m1.medium).

The spot price variations of the same instance type indifferent availability zones are different. It is feasible and desirable to use the spotprice history to estimate the probabilistic distribution of the spot price in a short time. Implications to model design. Those observations have significant implications on our model design.

The temporal and spatial price variations require special design of fault-tolerant mechanisms for reliability, particularly critical for MPI applications, where the failure of one MPI process usually cause the failure of the entire MPI application. Leverage the redundancies in different instance types and availability zones of Amazon EC2 to increase the probability of using spot instances to reduce the cost. Second, the dynamics in spot prices is a norm. It is impractical or unreliable to predict the exact next spot price, the probabilistic distribution of spot prices is predictable in a short time and use the spot price distribution to estimate the expected monetary cost. Cloud dynamics from a real cloud provider(Amazon EC2) for the probabilistic models on I/O and network performance as well as spot prices. Three workflow applications on Amazon EC2 and on a cloud simulator.Our experimental results demonstrate the following two major results. The calibrations from Amazon EC2, Dyna can accurately capture the cloud dynamics and guarantee the probabilistic performance requirements predefined by the users.

The impractical or unreliable to predict the exact next spot price, the probabilistic distribution of spot prices is predictable in a short time and use the spot price distribution to estimate the expected monetary cost. Cloud dynamics from a real cloud provider(Amazon EC2) for the probabilistic models on I/O and network performance as well as spot prices. It is feasible and desirable to use the spotprice history to estimate the probabilistic distribution of the spot price in a short time. Implications to model design.

# CHAPTER 2

## LITERATURE REVIEW

### 2.1 SCHEDULE OPTIMIZATION FORDATA PROCESSING FLOWS ON THE CLOUD

**Herald Kllapi** and **Eva Sitaridi et al.,** has proposed in this paper **scheduling** data processing workflows (data flows) on the cloud is a very complex and challenging task. It is essentially an optimization problem, very similar to query optimization, that is characteristically different from traditional problems in two aspects: Its space of alternative schedules is very rich, due to various optimization opportunities that cloud computing offers; its optimization criterion is at least two-dimensional, with monetary cost of using the cloud being at least as important as query completion time. Scheduling of data flows that involve arbitrary data processing operators in the context of three different problems:

1) Minimize completion time given a fixed budget,
2) Minimize monetary cost given a deadline, and
3) Find trade-offs between completion time and monetary cost without any a-priori constraints.

Problems and present an approximate optimization framework to address them that uses resource elasticity in the cloud. Herald kllapi et all (2011) proposed the effectiveness of our approach, incorporate the devised framework into a prototype system for dataflow evaluation and instantiate it with several greedy, probabilistic, and exhaustive search algorithms. Finally, through several experiments that have conducted with the prototype elastic optimizer on numerous scientific and synthetic data flows, we identify several interesting general characteristics of the space of alternative schedules as well as the advantages and disadvantages of the various search algorithms. The overall results are quite promising and indicate the effectiveness of our approach. Workflow scheduling and resource provisioning algorithms can result in significant differences in the monetary cost of WaaS providers running the service on IaaS clouds. Considering the cloud dynamics, our goal is to provide a probabilistic scheduling system for WaaS providers, aiming at minimizing the expected monetary cost while satisfying users' probabilistic deadline requirements **[1].**

## 2.2 COST OPTIMIZEDPROVISIONING OF ELASTIC RESOURCES FOR APPLICATION WORKFLOWS

**Maciej Malawski,E.-K. Byun et al.,** has proposed in this paper large-scale applications expressed as scientific workflows are often grouped into ensembles of interrelated workflows. Address a new and important problem concerning the efficient management of such ensembles under budget and deadline constraints on Infrastructure-as-Service (IaaS) clouds. ET all (2011) proposed the algorithms based on static and dynamic strategies for both task scheduling and resource provisioning. Perform the evaluation via simulation using a set of scientific workflow ensembles with abroad range of budget and deadline parameters, taking into account uncertainties in task runtime estimations, provisioning delays, and failures.

The key factor determining the performance of an algorithm is its ability to decide which workflows in an ensemble to admit or reject for execution. Admission procedure based on workflow structure and estimates of task runtimes can significantly improve the quality of solutions. Gain insight into resource management challenges when executing scientific workflow ensembles on clouds. Address a new and important problem of maximizing the number of completed workflows from an ensemble under both budget and deadline constraints **[2].**

## 2.3 DISTRIBUTED SYSTEMS MEET ECONOMICS: PRICING IN THE CLOUD

**H. Wang, Q. Jing, R. Chen et al.,** has proposed in this paper cloud computing allows users to perform computation in a public cloud with a pricing scheme typically based on incurred resource consumption. While cloud computing is often considered as merely a new application for classic distributed systems, we argue that, by decoupling users from cloud providers with a pricing scheme as the bridge, cloud computing has fundamentally changed the landscape of system design and optimization. Amazon EC2 cloud service and on local cloud computing tested, have revealed an interesting interplay between distributed systems and economics related to pricing. New angle of looking at distributed systems potentially fosters new insights into cloud computing. Cloud-computing paradigm has transformed a traditional distributed system into a “two-party” computation with pricing as the bridge. A provider designs its infrastructure to maximize profit with respect to the pricing scheme, while a user designs her application according to the incurred cost**.[3].**

## 2.4 PROFILING, WHAT IF ANALYSIS, AND COST BASED OPTIMIZATION

**Herodotos Herodotouand S. Papadimitriou et al.,** has proposed in this paper Map Reduce has emerged as a viable competitor to database systems in big data analytics. Map Reduce programs are a wide variety of application domains including business data processing, text analysis, natural language processing, Web graph and social network analysis, and computational science. Map Reduce systems lack a feature that has been key to the historical success of database systems, namely, cost-based optimization. A major challenge here is that, to the Map Reduce system, program consists of black-box map and reduce functions written in some programming language like C++, Java, Python, or Ruby. Cost-based Optimizer for simple to arbitrarily complex Map Reduce programs. The optimization opportunities presented by the large space of configuration parameters for these programs.

Profiler to collect detailed statistical information from unmodified Map Reduce programs, and a What-if Engine for fine-grained cost estimation. All components have been prototyped for the popular Hadoop Map Reduce system. To Herodotus ET all (2011) proposed the effectiveness of each component is demonstrated through a comprehensive evaluation using representative Map Reduce programs from various application domains. Map Reduce is a relatively young framework—both a programming model and an associated run-time system—for large-scale data processing. Hadoop is a popular open-source implementation of Map Reduce that many academic, government, and industrial organizations use in production deployments. Hadoop is used for applications such as Web indexing, data mining, report generation, log file analysis, machine learning, financial analysis, scientific simulation, and bioinformatics research.

Cloud platforms makes Map Reduce an attractive proposition for small organizations that need to process large datasets, but lack the computing and human resources of a Google or Yahoo! to throw at the problem. Elastic Map Reduce, for example, is a hosted platform on the Amazon cloud where users can provision Hadoop clusters instantly toper form data intensive tasks; paying only for the resources used. A job is expressed as a workflow of tasks with precedence constraints. A job has a soft deadline. The deadline of a job as a probabilistic requirement. Suppose a workflow is specified with a probabilistic deadline requirement. Due to their ability on reducing monetary cost.EC2 spot instances have recently received a lot of interests **[4].**

## 2.5 COST-DRIVEN SCHEDULING OF GRID WORKFLOWS USING PARTIAL CRITICAL PATHS

**F. Busching, G. Berriman et al.,** has proposed in this paper, Clouds are rapidly becoming an important platform for scientific applications. The application was developed to process astronomy data released by the Kepler project, a NASA mission to search for Earth-like planets orbiting other stars. Workflow was deployed across multiple clouds using the Pegasus Workflow Management System. The clouds used include several sites within the Future Grid, NERSC's Magellan cloud, and AmazonEC2. The application was deployed, evaluate its performance executing in different clouds (based on Nimbus, Eucalyptus, and EC2), and discuss the challenges of deploying and executing workflows in a cloud environment. Pegasus ET all (2012) proposed was able to support sky computing by executing a single workflow across multiple cloud infrastructures simultaneously. Cloud management systems provide a service-oriented model for provisioning and managing computational resources.

Scientists can request virtual machine resources on-demand for their application. The ability to provision resources, however, is not sufficient to run a workflow application. The computational resources provided by clouds are basic and in many cases only the base OS, networking and simple configuration is included. What is missing for scientific workflows are job and data management service Pegasus and Condor top provide these services **[5].**

## 2.6 THE FIVE-MINUTE RULE TEN YEARS LATER, AND OTHER COMPUTER STORAGE RULES OF THUMB

**Jim Gray, Goetz Graefe et al.,** has proposed in this paper economic and performance arguments suggest appropriate lifetimes for main memory pages and suggest optimal page sizes. The fundamental trade-offs are the prices and bandwidths of RAMs and disks. The analysis indicates that with today's technology, five minutes is a good lifetime for randomly accessed pages, one minute is a good lifetime for two-pass sequentially accessed pages, and 16 KB is good size for index pages. Jim Gray et all (1997) has proposed the rulesof-thumb change in predictable ways as technology ratios change. Many sequential operations read a large data-set and then revisit parts of the data.

Database join, cube, rollup, and sort operators all behave in this way. Consider the disk access behaviour of Sort in particular. Sort uses sequential data access and large disk transfers to optimize disk utilization and bandwidth. Sorting the input file, reorganizes the records in sorted order, and then sequentially writes the output file. If the sort cannot fit the file in main memory, it produces sorted runs in a first pass and then merges these runs into a sorted file in the second pass. Hash-join has a similar one-pass two-pass behaviour **[6].**

## 2.7 INTERNET ECONOMICS: THE USE OF SHAPLEY VALUE FOR ISPSETTLEMENT

**Richard T.B. Ma, Dah-mingChiu et al.,** has proposed in this \paper current Internet, autonomous ISPs implement bilateral agreements, with each ISP establishing agreements that suit its own local objective to maximize its profit. Peering agreements based on local views and bilateral settlements, while expedient, encourage selfish routing strategies and discriminatory interconnections. From a more global perspective, such settlements reduce aggregate profits, limit the stability of routes, and discourage potentially useful peering/connectivity arrangements, thereby unnecessarily balkanizing the Internet.

The distribution of profits is enforced at a global level, then there exist profit-sharing mechanisms derived from the coalition games concept of Shapley value and its extensions that will encourage these selfish ISPs who seek to maximize their own profits to converge to a Nash equilibrium.Dah-Ming Chiu et all (2011) has proposed these profit sharing schemes exhibit several fairness properties that support the argument that this distribution of profits is desirable. The Nash equilibrium point, the routing and connecting/peering strategies maximize aggregate network profits, encourage ISP connectivity so as to limit balkanization **[7].**

## 2.8 COMPUTING WHILE CHARGING: BUILDING A DISTRIBUTED COMPUTING INFRASTRUCTURE --USING SMARTPHONES

**M. Y. Arslan, S. Abrishami et al.,** has proposed in this paper Utility Grids have emerged as a new model of service provisioning in heterogeneous distributed systems. Users negotiate with service providers on their required Quality of Service and on the corresponding price to reach a Service Level Agreement. One of the most challenging problems in utility Grids is workflow scheduling, i.e., the problem of satisfying the Quality of Service of the users as well as minimizing the cost of workflow execution.

New Quality of Service based workflow scheduling algorithm based on a novel concept called Partial Critical Paths That tries to minimize the cost of work flow execution while meeting a user-defined deadline. Sundaresan ET all (2012) has proposed the Partial Critical Paths algorithm has two phases: in the deadline distribution phase it recursively assigns sub deadlines to the tasks on the partial critical paths ending at previously assigned tasks, and in the planning phase it assigns the cheapest service to each task while meeting its sub-deadline. The simulation results show that the performance of the PCP algorithm is very promising **[8].**

## 2.9 A TAXONOMY OF WORKFLOW MANAGEMENT SYSTEMS FOR GRID COMPUTING

**JiaYu ,RajkumarBuyya et al.,** has proposed in this paper advent of Grid and application technologies, scientists and engineers are building more and more complex applications to manage and process large data sets, and execute scientific experiments undistributed resources. Such application scenarios requiremeans for composing and executing complex workflows. Many efforts have been made towards the development of workflow management systems for Grid computing. A taxonomy that characterizes and classifies various approaches for building and executing workflows on Grids. Several representative Grid workflow systems developed by various projects world-wide to demonstrate thecomprehensiveness of the taxonomy.

The taxonomy not only highlights the design and engineering similarities and differences of state-of-the-art in Grid workflow systems, but also identifies the areas that need further research. Grid workflow can be seen as a collection of tasks that are processed on distributed resources in a well-defined order to accomplish a specific goal.

Workflow management techniques have been developed forever 20 years, especially in business management and office automation, and production management many successful approaches can be applied to Grid workflow for scientific applications. There exist several differences between Grid-based scientific workflows and conventional workflows such as long lasting workflow execution, large data flow, heterogeneous resources, multiple administrative domains, and dynamic resource availability and utilization **[9].**

## 2.10 A HYBRID HEURISTIC FOR DAG SCHEDULING ON HETEROGENEOUS SYSTEMS

**P. R. Elespuru, S. Shakya,H. Zhao et al.,** has proposed in this paper motivated by the observation that different methods to compute the weights of nodes and edges when scheduling DAGs onto heterogeneous machines may lead to significant variations in the generated schedule. To minimize such variations, a novel heuristic for DAG scheduling, which is based upon solving a series of independent task scheduling problems. A novel heuristic for the latter problem. Both heuristics compare favourably with other elated heuristics.DAG scheduling heuristics exhibit a similar behaviour too; some results are included here in Section.

The approach followed rank the nodes of a graph may affect the quality of the schedule produced by list scheduling, exacerbated in heterogeneous environments as a result of the heterogeneity. The resulting variations in the makes pan may be so significant that it can be difficult to determine the baseline behaviour of a heuristic. More importantly, the sensitivity that the heuristics exhibit, with respect to the approach used to compute the weights, indicates that there is scope for improvement in their design **[10].**

## 2.11 PEGASUS: A FRAMEWORK FOR MAPPING COMPLEX SCIENTIFIC WORKFLOWS ONTO DISTRIBUTED SYSTEMS

**E. Deelman, G. Singh, M.-H. Su et al.,** has proposed in this paper network analysis is a quantitative methodology for studying properties related to connectivity and distances in graphs, with diverse applications like citation indexing and information retrieval on the Web. The hyperlinked structure of Wikipedia and the ongoing, incremental editing process behind it make it an interesting and unexplored target domain for network analysis techniques. Apply two relevance metrics, HITS and Page Rank, to the whole set of English Wikipedia entries, in order to gain some preliminary insights on the macro-structure of the organization of the corpus, and on some cultural biases related to specific topics. Wikipedia is structured as an interconnected network of articles. Each article can Hyperlink several other Wikipedia entries. It's up to the contributor to establish a hyperlink connection between a term that occurs in the article and the corresponding Wikipedia entry, provided that one exists; if the entry does not exist, it is possible to create a new empty ”stub” entry, and link it **[11].**

## 2.12 SCHEDULING SCIENTIFIC WORKFLOWS ELASTICALLY FOR CLOUD COMPUTING

**C. Lin and Jia Yu et al.,** has proposed in this paper grid technologies have progressed towards a service-oriented paradigm that enables a new way of service provisioning based on utility computing models, which are capable of supporting diverse computing services. It facilitates scientific applications to take advantage of computing resources distributed worldwide to enhance the capability and performance. Many scientific applications in area such as bioinformatics and astronomy require workflow processing in which tasks are executed based on their control or data dependencies. Scheduling such interdependent tasks on utility Grid environments need to consider users' QoS requirements. A genetic algorithm approach to address scheduling optimization problems in workflow applications, based on two QoS constraints, deadline and budget. For utility computing based services, users consume the services when they need to, and pay only for what they use.

With economy incentive, utility computing encourages organizations to offer their specialized applications and other computing utilities as services so that other individuals/organizations can access these resources remotely. It facilitates individuals/organizations to develop their own core activities without maintaining and developing fundamental infrastructure. Providing utility computing services has been reinforced by service-oriented Grid computing that creates an infrastructure for enabling users to consume services transparently over a secure, shared, scalable, sustainable and standard world-wide network environment **[12].**

## 2.13 MARKET-ORIENTED CLOUD COMPUTING:

**D. Datla, X. Chen et al.,** has proposed in this paper many online data sets grow incrementally over time as new entries are slowly added and existing entries are deleted or modified. As new data and updates are constantly arriving, the results of data mining applications become outdated over time. To keep away from the scenario it is necessary to refreshing mining results so that it can avoid the cost of re-computation from scratch. Map Reduce for efficient iterative computations, it is too expensive to perform an entirely new large-scale Map Reduce iterative job to timely accommodate new changes to the underlying data sets. The i2MapReduce is an extension to Map Reduce that supports key-value pair level incremental processing and also more sophisticated iterative computation, which is widely used in data mining applications .Incremental interactive processing, is a small number of updates which propagate to affect a large portion of intermediate states after a number of iterations. The technique helps in improving the job running time and reduces the running time of refreshing the results of big data **[13].**

## 2.14 PERFORMANCE ANALYSIS OF CLOUD COMPUTING SERVICES

**A. Iosup, RajkumarBuyya et al.,** has proposed in this paper A 21st century vision of computing; identifies various computing paradigms promising to deliver the vision of computing utilities; defines Cloud computing and provides the architecture for creating market-oriented Clouds by leveraging technologies such as VMs; provides thoughts on market-based resource management strategies that encompass both customer-driven service management and computational risk management to sustain SLA oriented resource allocation; presents some representative. Cloud platforms especially those developed in industries along with our current work towards realizing market-oriented resource allocation of Clouds by leveraging the 3rd generation Aneka enterprise.

Grid technology: reveals our early thought son interconnecting Clouds for dynamically creating an atmospheric computing environment along with pointers to future community research and concludes with the need for convergence of competing IT paradigms for delivering our 21st century vision **[14].**

## 2.15 CLOUD COMPUTING AND GRID COMPUTING 360-DEGREE COMPARED

**D. Datla, H. I. Volos, S. M. Hasan et al.,** has proposed in this paper cloud intelligence applications often perform iterative computations (e.g., Page Rank) on constantly changing data sets (e.g. graph). While previous studies extend Map Reduce for efficient iterative computations, it is too expensive to perform an entirely new large-scale Map Reduce iterative job to timely accommodate new changes to the underlying data sets. I2MapReduce to support incremental iterative computation.

The changes impact only a very small fraction of the data sets, and the newly iteratively converged state is quite close to the previously converged state. I2MapReduce exploits this observation to save re-computation by starting from the previously converged state, and by performing incremental updates on the changing data. I2MapReduce sees significant performance improvement over re-computing iterative jobs in Map Reduce **[15].**

## 2.16 AUTOMATED CONTROL IN CLOUD COMPUTING: CHALLENGES AND OPPORTUNITIES

**D. Datla, Harold C.Lim et al.,** has proposed in this paper advances in virtualization technology, virtual machine services offered by cloud utility providers are becoming increasingly powerful, anchoring the ecosystem of cloud services. Virtual computing services are attractive in part because they enable customers to acquire and release computing resources for guest applications adaptively in response to load surges and other dynamic behaviour's. "Elastic" cloud computing APIs present a natural opportunity for feedback controllers to automate this adaptive resource provisioning. Many recent works have explored feedback control policies for a variety of network services under various assumptions.

The challenge of building an effective controllers a customer add-on outside of the cloud utility service itself. Such external controllers must function within the constraints of the utility service APIs. It is important to consider techniques for effective feedback control using cloud APIs, as well as how to design those APIs to enable more effective control. Proportional thresholding, a policy enhancement for feedback controllers that enables stable control across a wide range of guest cluster sizes using the coarse-grained control offered by popular virtual compute cloud services **[16].**

## 2.17 TOWARDSPREDICTABLE DATACENTER NETWORKS

**TH. Ballani, P. Costa, T. Karagiannis et al.,** has proposed in this paper the shared nature of the network in today's multi-tenant data centers implies that network performance for tenant scan vary significantly. Applies to both production data-centres and cloud environments. Network performance variability hurts application performance which makes tenant costs unpredictable and causes provider revenue loss. Motivated by these factors. The case for extending the tenant-provider interface to explicitly account for the network. Achieved by providing tenants with a virtual network connecting their compute instances. The key contribution of the design of virtual network abstractions that capture the trade-off between the performance guarantees offered to tenants, their costs and the provider revenue. To illustrate the feasibility of virtual networks, Octopus, a system that implements the proposed abstractions. Using realistic, large-scale simulations and an Octopus deployment on a 25-node two-tier test bed, demonstrate that the use of virtual networks yields significantly better and more predictable tenant performance. Simple pricing model, that the abstractions can reduce tenant costs by up to 74% while maintaining provider revenue neutrality **[17].**

## 2.18 CLOUDSIM: A TOOLKIT FOR MODELING AND SIMULATION OF CLOUD COMPUTING ENVIRONMENTS AND EVALUATION OF RESOURCE PROVISIONING ALGORITHMS

**R. N. Calheiros, R. Ranjan et al.,** has proposed in this paper a recent advancement wherein IT infrastructure and applications are provided as 'services' to end-users under a usage-based payment model. It can leverage virtualized services even on the fly based on requirements (workload patterns and QoS) varying with time. The application services hosted under Cloud computing model have complex provisioning, composition, configuration, and deployment requirements. Evaluating the performance of Cloud provisioning policies, application workload models, and resources performance models in a repeatable manner under varying system and user configurations and requirements is difficult to achieve. To overcome this challenge, CloudSim: an extensible simulation toolkit that enables modelling and simulation of Cloud computing systems and application provisioning environments. The CloudSim toolkit supports both system and behaviour modelling of Cloud system components such as data centres, virtual machines (VMs) and resource provisioning policies. It implements generic application provisioning techniques that can be extended with ease and limited effort.

Currently, it supports modelling and simulation of Cloud computing environments consisting of both single and inter-networked clouds (federation of clouds). It exposes custom interfaces for implementing policies and provisioning techniques for allocation of VMs under inter-networked Cloud computing scenarios. Several researchers from organizations, such as HP Labs in U.S.A., are using CloudSim in their investigation on Cloud resource provisioning and energy-efficient management of data centre resources. The usefulness of Clouds is demonstrated by a case study involving dynamic provisioning of application services in the hybrid federated clouds environment. The federated Cloud computing model significantly improves the application QoS requirements under fluctuating resource and service demand patterns **[18].**

## 2.19 A PERFORMANCE STUDY ON THE VMSTARTUP TIME IN THE CLOUD

**M. Mao and M. Humphrey et al.,** has proposed in this paper one of many advantages of the cloud is the elasticity, the ability to dynamically acquire or release computing resources in response to demand. However, this elasticity is only meaningful to the cloud users when the acquired Virtual Machines (VMs) can be provisioned in time and be ready to use within the user expectation. The long unexpected VM start-up time could result in resource under-provisioning, which will inevitably hurt the application performance. A better understanding of the VM start-up time is needed to help cloud users to plan ahead and make in-time resource provisioning decisions.

The start-up time of cloud VMs across three real-world cloud providers – Amazon EC2, Windows Azure and Rack space. The relationship between the VM start-up time and different factors, such as time of the day, OS image size, instance type, data center location and the number of instances acquired at the same time. The VM start-up time of spot instances in EC2, which show a longer waiting time and greater variance compared to on-demand instances **[19].**

## 2.20 TRANSFORMATION-BASED MONETARY COST OPTIMIZATIONS FOR WORKFLOWS IN THE CLOUD

**A. C. Zhou and B. He et al.,** has proposed in this paper recently, performance and monetary cost optimizations for workflows from various applications in the cloud have become about research topic. That most existing studies adopt ad hoc optimization strategies, which fail to capture the key optimization opportunities for different workloads and cloud offerings (e.g., virtual machines with different prices). ToF, a general transformation-based optimization framework for workflows in the cloud. Specifically, ToF formulates six basic workflow transformation operations.

An arbitrary performance and cost optimization process can be represented as a transformation plan (i.e., a sequence of basic transformation operations). All transformations form a huge optimization space. Develop a cost model guided planner to efficiently find the optimized transformation for a predefined goal (e.g., minimizing the monetary cost with a given performance requirement). ToF on real cloud environments including Amazon EC2 and Rack space. The effectiveness of ToF in optimizing the performance and cost in comparison with other existing approaches **[20].**

# CHAPTER 3

## SYSTEM ANALYSIS DESCRIPTION

### 3.1 EXISTING SYSTEM

Scientific applications partially or entirely shifting from traditional computing platforms (e.g., grid) to the cloud. Due to the pay-as-you-go computational behaviour, performance and (monetary) cost optimizations have recently become a hot research topic for workflows in the cloud. To address the limitations of current approaches, propose Profit Maximization, a transformation-based optimization framework for optimizing the performance and cost of workflows in the cloud. Profit Maximization models the cost and performance optimizations of workflows as transformations .It performance and monetary cost optimizations for workflows from various applications in the cloud have become a hot research topic. That most existing studies adopt ad hoc optimization strategies, which fail to capture the key optimization opportunities for different work resource costs and cloud offerings (e.g., virtual machines with different prices). WaaS providers charge users according to the execution of workflows and their QoS requirements. In this proposal, we argue that the WaaS provider should offer a probabilistic performance guarantee for users. Particularly, we can offer some fuzzy-style interfaces for users to specify their probabilistic deadline requirements, such as “Low”, “Medium” and “High”, Inside Dyna, we translate these requirements into probabilities of deadline. For example, the user may select the loose deadline of 4 hours with the probability of 96 percent. Ideally, the WaaS provider tends to charge higher prices to users when they specify tighter deadline and/or higher probabilistic deadline guarantee

### 3.1.1 DISADVANTAGES:

- This TOF Planning has tendency to make administration inflexible.
- There is no scope for individual freedom on performance and cost of Workflows in the cloud.
- Elaborate planning may create a false sense of security to the effect that everything is taken for granted.
- Therefore they cloud service may be fail to take up timely actions and an opportunity is lost.
- The application owners submit workflows with specified deadlines for QoS purposes.

## 3.2. PROPOSED SYSTEM

Proposed framework through large-scale simulations, driven by cluster-usage traces that are provided by Google. A PG-TOF based DHT scheduling algorithm that generates VM requests based on the user resource usage in these traces. Under-pricing conditions that are aligned with those of Amazon EC21, our admission control algorithms substantially increase resource cost for the provider. To maximize the profit, a service provider should understand both service charges and business costs, and how they are determined by the characteristics of the applications and the configuration of a resource allocation system. The problem of optimal resource allocation configuration for profit maximization in a cloud computing environment is studied. Pricing model takes such factors into considerations as the amount of a service, the workload of an application environment. The configuration of a resource allocation system, the service-level agreement, the satisfaction of a consumer, the quality of a service, the penalty of a low-quality service, the cost of renting, the cost of energy consumption, and a service provider's margin and profit. PG-TOF is to treat a resource allocation system is a queuing model, such that our optimization problem can be formulated and solved analytically. Two server speed and power consumption models are considered, namely, the idle-speed model and the constant-speed model.

The probability density function of the waiting time of a newly arrived service request is derived. The expected service charge to a service request is calculated. The expected net business gain in one unit of time is obtained. Numerical calculations of the optimal server size and the optimal server speed are demonstrated. Resource allocation approach is based on we find many risk in Profit Maximization on multiple clouds. Still, there are many practical and challenging issues for current multi-cloud environments. Issues include relatively limited cross-cloud network bandwidth and lacking of cloud standards among cloud providers. Relies on the assumption that all qualified nodes must satisfy Inequalities in existing system. To meet this requirement, we design a resource discovery protocol, namely pointer-gossiping PG-TOF, to find these qualified nodes. PG-TOF to adapt to the multidimensional feature. Traditional PG-TOF, each node (a.k.a., duty node) under PG-TOF is responsible for a unique multidimensional range zone randomly selected when it joins the overlay. Some of them are inherit in the process of planning like rigidity and other arise due to shortcoming of the techniques on multi cloud. Profit Maximization, a general transformation-based optimization framework for workflows in the cloud. Specifically, Profit Maximization formulates six basic workflow transformation operations.

An arbitrary performance and cost optimization process PG-TOF be represented as a transformation plan, a sequence of basic transformation operations including Amazon EC2 and Rack space.

### 3.2.1 ADVANTAGES:

- The effectiveness of Profit Maximization in optimizing the performance and cost in comparison with other existing approaches.
- Exhibitions are open to a large and sometimes diverse range of audiences (usually the general public).
- Provides you with a perfect platform to promote.
- This PG-TOF with multi-cloud or service to a broader group that may have better knowledge and co-operate with our services.
- Promote services with minimum cost. Better performance with lack of minimal resources at on demand services.

## 3.3 MODULE DESCRIPTION

- TASK PLANNING AND SCHEDULING
- WORKFLOW SCHEDULING AND MANAGEMENT
- WORKFLOW OPTIMIZER
- JOB SCHEDULER
- COST AND TIME ESTIMATION USING DAG

### 3.3.1 TASK PLANNING AND SCHEDULING MODULE

A task planning scheduling module based on evolutionary algorithms called TOF has been developed, it's able to optimize a given configuration of tasks and resources. It can efficiently exploit the resources you have, lower waste,in terms of costs and/or energy, and maximize efficiency. The task related to finding the most appropriate way to optimize productivity in product development and manufacturing processes can be highly complex even for quite small projects; scheduling problems are usually NP-hard. In their more generic form, they seek to respond to the following question: given a set of tasks/activities, a set of resources, and a metric to assess the performance, what is the best way to allocate the resources to the tasks in order to optimize the performance.

Cloud is by design a shared infrastructure, and the interference causes significant variations in the performance even with the same instance type. Significant variances on I/O and network performance. The assumption of static task execution time in the previous studies does not hold in the cloud. Under the static execution time assumption, the deadline notion is aS“deterministic deadline”. Due to performance dynamics, a more rigorous notion of deadline requirement is needed to cope with the dynamic task execution time.The application owners submit workflows with specified deadlines for QoS purposes. WaaS providers charge users according to the execution of workflows and their QoS requirements. In this proposal, we argue that the WaaS provider should offer a probabilistic performance guarantee for users. Particularly, we can offer some fuzzy-style interfaces for users to specify their probabilistic deadline requirements, such as “Low”, “Medium” and “High”, as illustrated in Fig. 2. Inside Dyna, we translate these requirements into probabilities of deadline. For example, the user may select the loose deadline of 4 hours with the probability of 96 percent. Ideally, the WaaS provider tends to charge higher prices to users when they specify tighter deadline and/or higher probabilistic deadline guarantee. The design of the billing scheme for WaaS is beyond the scope of this paper, and we will explore it as future work.

### 3.3.2 WORKFLOW SCHEDULING AND MANAGEMENT

The workflow scheduling strategy developed in order to allow tasks to only use a part of the resources. The methodology is based on a decision parameterization allowing to apply generic evolutionary TOF six workflow techniques to solve scheduling problems. The purpose of the research work targeted in the project was not intended to develop a problem specific algorithm but rather to investigate how a generic optimisation tool based on cloud can be used to solve task planning optimisation problems without major modifications to the optimisation algorithm itself. The genericity of the developments comes mainly from the separation into two modules: the work flow optimizer and the Job scheduler. The performance validated on a well-known job-shop scheduling problem of the literature showing promising results and has been integrated in the monetary cost analysis prototype through the software integration framework developed within the project.

Three parities in this scenario, namely the workflow application owner, WaaS provider and IaaS cloud provider. Different application owners submit a number of workflows with different parameters to WaaS and the WaaS provider rent resources from the cloud provider to serve the applications. The application owners submit workflows with specified deadlines for QoS purposes. WaaS providers charge users according to the execution of workflows and their QoS requirements. WaaS provider should offer a probabilistic performance guarantee for users. Particularly, some fuzzy-style interfaces for users to specify their probabilistic deadline requirements, such as “Low”, “Medium” and “High”. Inside Dyna, translate these requirements into probabilities of deadline. For example, the user may select the loose deadline of 4 hours with the probability of 96 percent. Ideally, the WaaS provider tends to charge higher prices to users when they specify tighter deadline and/or higher probabilistic deadline guarantee. The design of the billing scheme for WaaS is beyond the scope of this paper, and we will explore it as future work.

Different workflow scheduling and resource provisioning algorithms can result in significant differences in the monetary cost of WaaS providers running the service on IaaS clouds. Considering the cloud dynamics, goal is to provide a probabilistic scheduling system for

WaaS providers, aiming at minimizing the expected monetary cost while satisfying users' probabilistic deadline requirements.

### 3.3.3 WORKFLOW OPTIMIZER

There are a number of technical challenges in designing and implementing the planner. First, the transformation operations are composable. The order of applying transformation operations also matters for performance and cost optimizations. The searching space for an optimal transformation sequence is huge. Second, the optimization is an online process and should be lightweight. Find a good balance between the quality of the transformation sequence and the runtime overhead of the planner. Due to the huge space, a thorough exploration of the optimization space is impractical. Third, the planner should be able to handle different trade-offs on the monetary cost and performance goals.

Cost-aware optimizations. Workflow scheduling with deadline and budget constraints deadline assignment for the tasks within a job and used genetic algorithms to find optimal scheduling plans. Multi-objective methods such as evolutionary algorithms have been adopted to study the trade-off between monetary cost and performance optimizations for workflow executions. Those studies only consider a single workflow with on-demand instances only. Dynamic scheduling strategies for workflow ensembles. Auto-scaling techniquesbased on static execution time of individual tasks. Dyna is that it targets at offering probabilistic performance guarantees as QoS, instead of deterministic deadlines. Dyna schedules the workflow by explicitly capturing the performancedynamics (particularly for I/O and network performance) in the cloud. Calheiros, Buyya and Calheiros algorithm with task replications to increase the likelihood of meeting deadlines.

Due to their ability on reducing monetary cost, Amazon EC2 spot instances have recently received a lot of interests. Yehudaetal. Conducted reverse engineering on the spot price and figured out a model consistent with existing price traces. Javadi et al. developed statistical models for different spot instance types. Those models can be adopted to our hybrid execution. Introduced some check pointing mechanisms for reducing cost of spot instances, studies used spot instances with different bidding strategies and incorporating with fault tolerance techniques such as check pointing, task duplication and migration. Without offering any guarantee on meeting the workflow deadline like Dyna. Similar to Dyna, Chu and Simmhan hybrid method to use both on-demand and spot instances for minimizing total cost while satisfying deadline constraint. They did not consider the cloud performance dynamics.

### 3.3.4 JOB SCHEDULER:

Schedule workflows for periodic execution on a cloud server running for the job scheduling. It’s used within the Reporting suite Initial instance assignment. It considers multiple heuristics. Present three initialization heuristics for initial instance assignment, namely Best-fit, Worst-fit and Most-efficient. The Best-fit heuristic assigns each task with the most expensive instance type. Maximize performance but at the cost of a high monetary cost. Ideally, it should satisfy the deadline. Otherwise, we raise an error to the user. The Worst-fit heuristic first assigns each task with the cheapest instance type to minimize the cost. GAIN approach to repeatedly re-assign tasks to a better instance type.

GAIN is a greedy approach which picks the task with the largest benefit in execution time until the deadline requirement is met. The process of $A^\$$ search can be modelled as a search tree. In the formulated $A^\$$ search, we first need to clarify the definitions of the state and the state transitions in the search tree. A state is a configuration plan to the workflow, represented as a multi-dimensional vector. Each dimension of the vector represents the instance configuration of an on-demand instance type for each task in the workflow. This configuration is extended to hybrid instance configuration in the hybrid instance configuration refinement.

Workflow with three tasks is represented, meaning that task I (0 i 2) is configured with on demand instance type $t_i$. Starting from the initial state (root node of the search tree), the search tree is traversed by transmitting from a state to its child states level by level. At level l, the state transition is to replace the l th dimension in the state with all equally or more expensive instance types. Three on-demand instance types (type 0, 1 and 2 with increasing on-demand prices). From the initial state where all tasks are assigned to the cheapest instance type (instance type 0), we move to its child states by iterating the three available instance types for the first task

A search adopts several heuristics to enable its pruning capability. Particularly, $A^\$$ evaluates a state s by combining two distance metrics, which are the actual distance from the initial state to the state s and the estimated distance from the state sto the goal state, respectively. Also referred as g score and h score for s, respectively. . If the monetary cost of a states is higher than the best found result, its successors are unlikely to be the goal state since they have more expensive configurations than s. For example, assume state on the search tree in Fig. 4 has a high search cost, the grey states on the search tree are pruned since they have higher monetary cost. During the $A^\$$ search, we maintain two lists, namely the Open List and Closed List. The Open List contains states that are potential solutions to the problem and are to be searched later. States already been searched or with high search cost are added to the Closed List and do not need to be considered again during the $A^{\$}$ search.

Algorithm optimization process of the $A^{\$}$-based instance configuration algorithm. Iteratively, the Open List and add their neigh boring states into the Open List, feasible states that satisfy the probabilistic deadline guarantee. Estimate performance is used to estimate the feasibility of states, lowest search cost found during the search process as the upper bound to prune the unuseful states on the search tree. Function estimate cost returns the estimation for the h and g scores of states. When expanding the Open List, add the neighboring states with lower search cost than the upper bound.

### 3.3.5 COST AND TIME ESTIMATION USING DAG

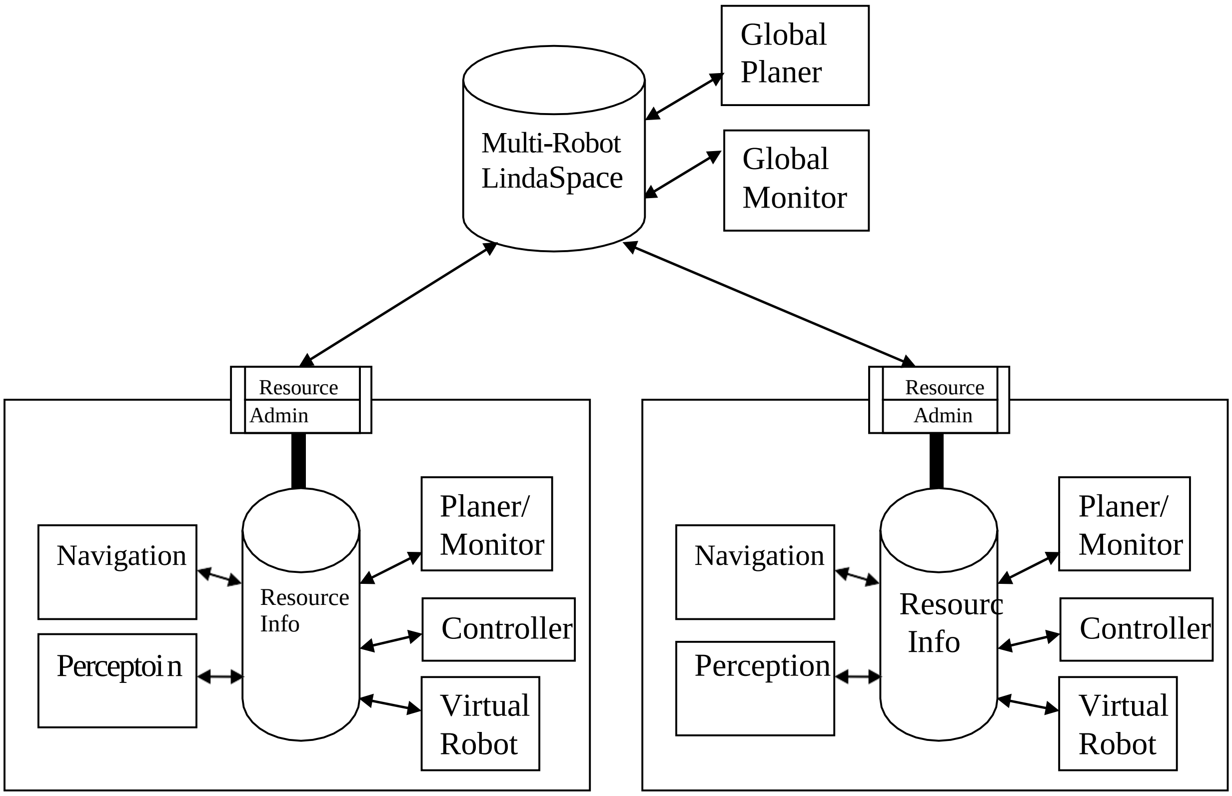
Effective cost models to estimate the cost and the time changes for applying one transformation operation on the instance DAG. Since an auxiliary scheme does not directly reduce the cost, estimate the potential cost saving of the main schemes after applying the auxiliary scheme. As for the time estimation, the changes of execution time need to be propagated to all the tasks with dependencies on the vertices affected by the transformation operation, the worst case for the change of execution time, since worst-case analysis usually can have simplified estimation process. Probabilistic distributions of the execution time, denoting the execution time distribution of Task 0, 1,..., n  1 to be $PDF_0$, $PDF_1$,..., $PDF_n1$ A hybrid instance configuration of a task is represented as a vector of both spot and on demand instance types. The last dimension in the vector is the on-demand instance type obtained from the $A^{\$}$-based instance.

The initial hybrid configuration contains only the on-demand instance type. Starting from the initial configuration, Spot instances at the beginning of the hybrid instance configuration to find better configurations. Add n spot instances (n is a predefined parameter). A larger n gives higher probability of benefiting from the spot instances while a smaller n gives higher probability of meeting deadline requirement and reduces the optimization overhead. Find that n ¼ 2 is sufficient for obtaining good optimization results.

A larger n greatly increases the optimization overhead with only very small improvement on the optimization results.

It is a challenging task to develop an efficient and effective approach for hybrid instance configuration refinement. First, coupled with the performance dynamics, it is a nontrivial task to compare whether one hybrid instance configuration. The overall execution time equals to the time that task T has run on the spot instance before it fails, $t_f$, plus the execution time of task T on the on-demand instance $t_o$, with the following probability.

**Fig 3.1**: Architecture of cost optimization



# CHAPTER 4

# SYSTEM REQUIREMENTS

## 4.1 HARDWARE REQUIREMENTS:

| | | |
|---|---|---|
| Processor Type | : | Intel Pentium i5/i7 |
| Speed | : | 3.40GHZ |
| RAM | : | 4GB DD2 RAM |
| Hard disk | : | 500 GB |

## 4.2 SOFTWARE REQUIREMENTS:

| | | |
|---|---|---|
| Operating System | : | Windows 10 |
| Front End | : | Netbeans 8.1 IDE |
| Coding Language | : | Java |
| Tools | : | Weka Tools |

# CHAPTER 5

## EXPERIMENTAL SETUP AND PROCEDURE

### 5.1 ABSOLUTE ERROR:

The absolute error is defined as the absolute value of the difference between the measured value and the true value. To demonstrate our solution we run a CPU steal case in a Cloudsim2.3.4 cluster of 2 PMs (PM1 and PM2). In our case, PM1 has 32GB or RAM while PM2 has 16 GB of RAM. After, we place one VM to PM1 and we run the Cloud simulator workload respectively. It shows the statistical CPU steal time distribution in an x, y plane. We can observe that the CPU steal time (default) is higher than our solution (that minimizes the overall steal time).We proposed and evaluated memory-aware cloud scheduling techniques, which Do not require any prior knowledge on the behaviours of VMs. This work shows that VM live migration can also be used to mitigate micro-architectural resource contentions, and the cloud-level VM scheduler must consider such hidden contentions. We plan to extend our preliminary design of TOF-aware scheduling for more Efficient TOF affinity supports with hot page migrations. Also, we will investigate a systematic approach based on a cost-benefit analysis for VM migrations and contention reductions.

Thus, let:

$e_a$ = the absolute error

$x_m$ = the measured value

$x_t$ = the true value

The formula for computing absolute error is:

$e_a = | x_m - x_t |$

**Table 1 Working Scenario**

| Work Scenario | Mean Absolute Error(%) | Relative Absolute Error(%) |
|---|---|---|
| **Existing System** | 0.325 | 55 |
| **Proposed System** | 0.075 | 16 |

### 5.1.1 MEAN ABSOLUTE ERROR

The *mean absolute error* function is given by

$$\mathrm{MAE}(t) = \frac{1}{n}\sum_{i=1}^{k} f_i \left|x_i - t\right| = \sum_{i=1}^{k} p_i \left|x_i - t\right|$$

As the name suggests, the mean absolute error is a weighted average of the absolute errors, with the relative frequencies as the weight factors.

Recall also that we can think of the relative frequency distribution as the probability distribution of a random variable X that gives the mark of the class containing a randomly chosen value from the data set. With this interpretation, the MSE (t) is the first absolute moment of X about t:

MAE (t) = E[|X - t|]

MAE (t) may seem to be the simplest measure of overall error when t is used to represent the distribution.

### 5.1.2 RELATIVE ABSOLUTE ERROR

You first need to determine absolute error to calculate relative error. Relative error expresses how large the absolute error is compared with the total size of the object you are measuring.

Relative error is expressed as a fraction or is multiplied by 100 and expressed as a percent.

Relative error is determined by using the following formula:

**Relative Error = Absolute Error / Known Value**

## 5.2 VIRTUAL MACHINE CLOUD PLACEMENT

The prominent technology that drives the industry now-a-days is cloud computing. The growth of cloud computing has resulted in the setup of large number of data centers around the world. The data centers consume more power making it source for the carbon dioxide emission and a major contributor to greenhouse effect. This led to the deployment of virtualization. Infrastructure as a Service is one of the important services offered by cloud computing that allows virtualization and hardware to get virtualized by creating many instances of Virtual Machine (VM) on a single Physical Machine (PM) and helps in improving utilization of resources. VM consolidation includes method of choosing the more appropriate algorithm for migration of VM's and placement ofVMs to the most suitable host. VM placement is a part of VM migration. The effective placement of VM is aimed to improve performance, resource utilization and reduce the energy consumption in data centre without SLA violation. This work aims to focus on various VM placement schemes

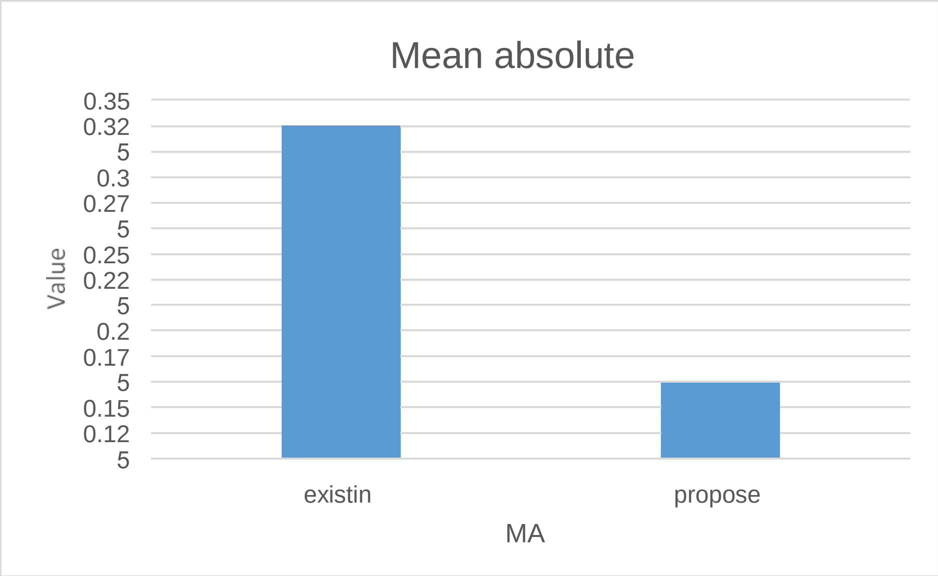



Fig 5.1: Mean Absolute Errors

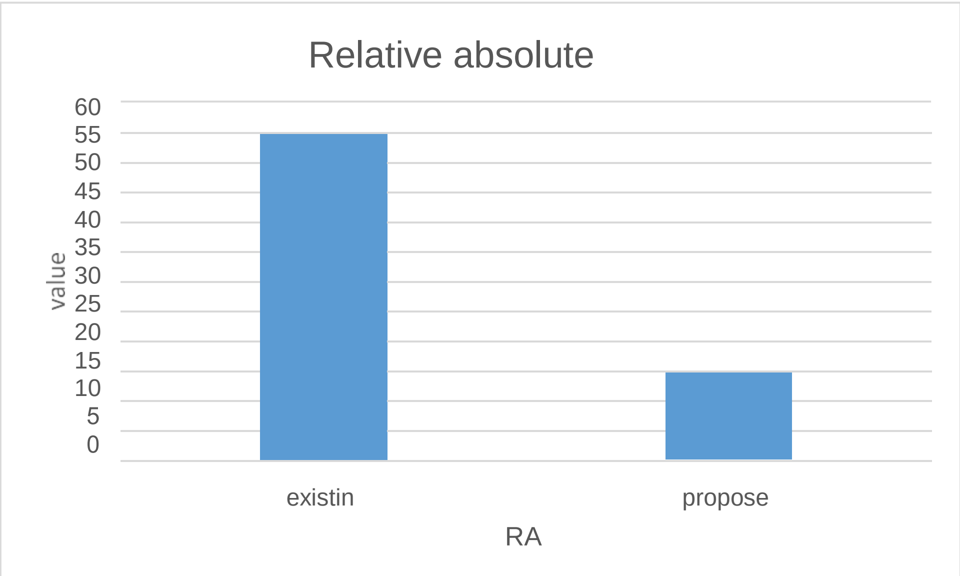



Fig 5.2: Relative Absolute Errors

# CHAPTER 6

## RESULTS AND DISCUSSION

### 6.1 VM EXPERIMENT:

Another experimental case involves execution of the Net beans 8.3 workload in a medium size VM that has been deployed in Cloudsim2.3.4. In particular, we run 100 inserts and 200 updates and we observe the CPU steal time. The time series in ”x” axis represent the time, while in ”y” axis the CPU steal time over the workload execution (its time point represent the measurement of the steal time in relation to the previous point, for example from 6.88 to 6.89 represents CPU steal time of 1%).

It demonstrates that during 10 minutes, the CPU steal time percentage was overall 10% (increased from 6.88 to 6.98). Based on this discussion we conclude that CPU steal time is an important factor to take in mind during VM scheduling as it can significantly affects VMs CPU utilization levels. A more refined VM scheduling can be based on predicting the CPU steal time according to the real time resource usage in order to perform scheduling that minimizes the CPU steal time.

| | |
|---|---|
| No. of .virtual machine | 16 |
| No. of. Physical machine | 20 |
| No. of classifiers | 02 |


Fig 6.1: VM Performance

The effectiveness of Profit Maximization in optimizing the performance and cost in comparison with other existing approaches. The cost calculated by, the lower the better from Computational overhead (time), measured by the time taken by an approach to find the solution, the lower the better. The processing algorithm reduce the cost and maximization the resources. The demonstration of how the changes in the setting parameters impacted their performance.

# CHAPTER 7

## CONCLUSION

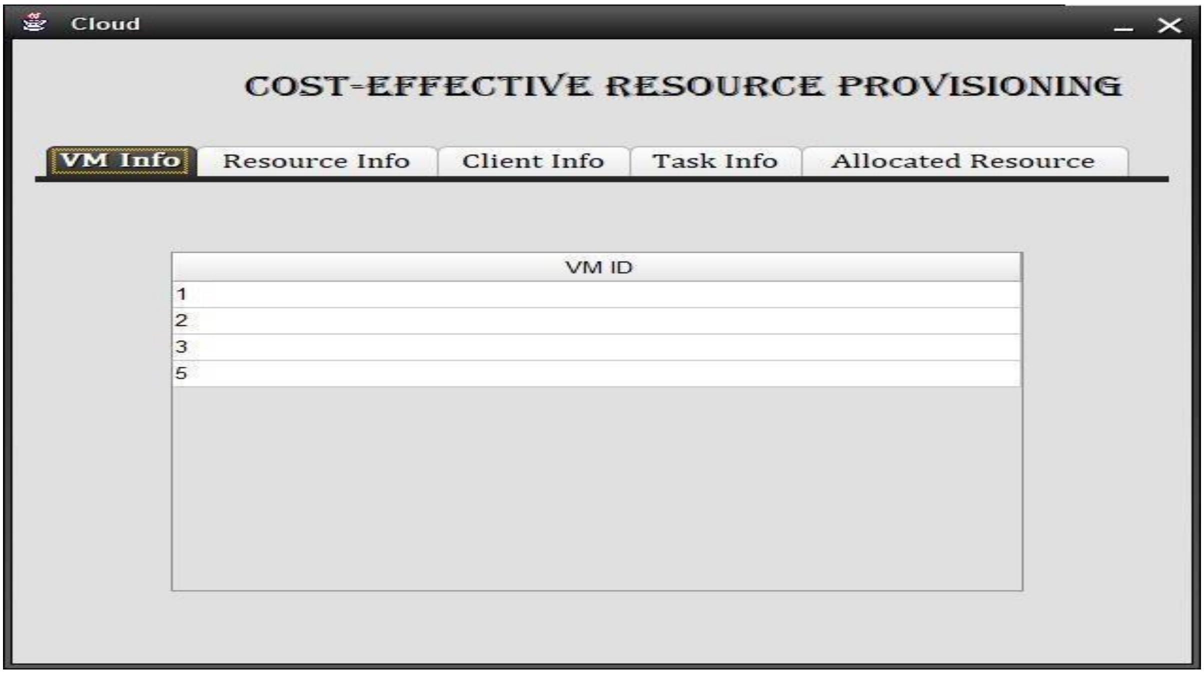
Building a distributed computing infrastructure using smart phones for enterprises, technical challenges in building such an infrastructure. Address many of them to design, a framework that supports such an infrastructure. The viability and efficacy of various components within novel scheme (ToF) for virtual resource allocation on a SOC, with three key contributions listed below. Optimization of task's resource allocation under user's budget. With a realistic monetary model, it proposes a solution which can optimize the task execution performance based on its assigned resources under the user budget. It proves its optimality using the CWC conditions in the convex-optimization theory. Maximized resource utilization based on ToF: In order to further make use of the idle resources, Design a dynamic algorithm by combining the above algorithm with ToF and the arrival/completion of new tasks. Give incentives to users by gaining an extra share of unused resource without more payment.

We formulated the edge data distribution strained optimization problem from the app vendor's perspective. Extensive experiments were conducted on a widely-used real-world dataset to evaluate the performance of the proposed approaches.

Mean Absolute Error is a model evaluation metric used with regression models. The mean absolute error of a model with respect to a test set is the mean of the absolute values of the individual prediction errors on over all instances in the test set. As the name suggests, the mean absolute error is a weighted average of the absolute errors, with the relative frequencies as the weight factors.

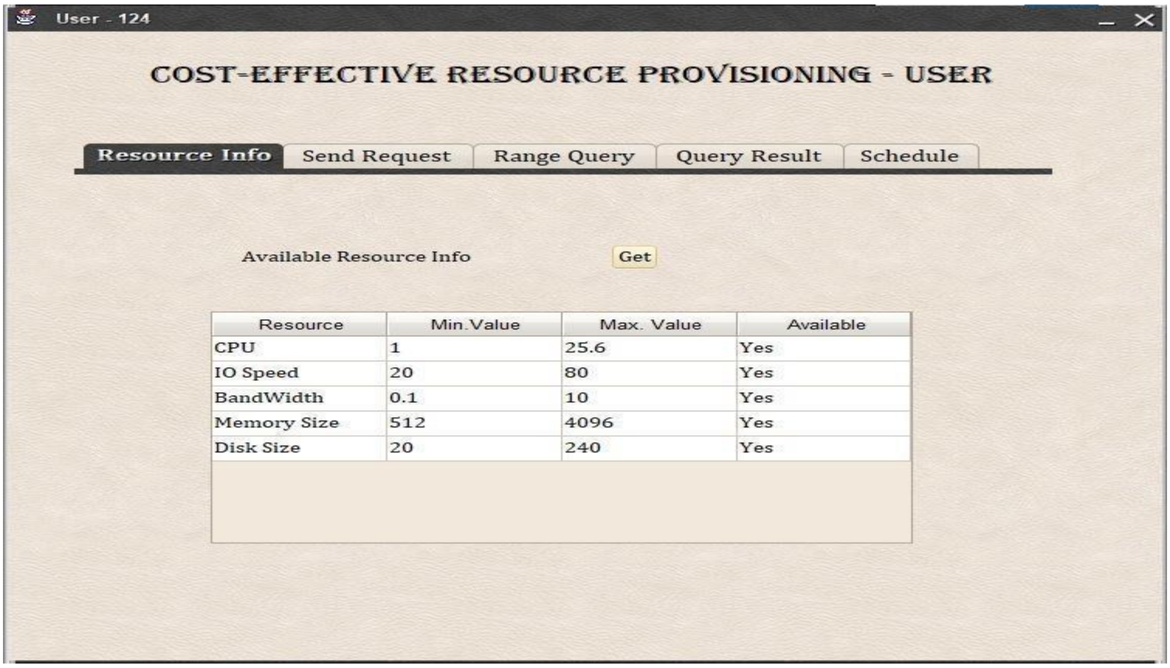
Relative error expresses how large the absolute error is compared with the total size of the object you are measuring. Relative error is expressed as a fraction or is multiplied by 100 and expressed as a percent. The demonstration of how the changes in the setting parameters impacted their performance.

# SCREENSHOTS

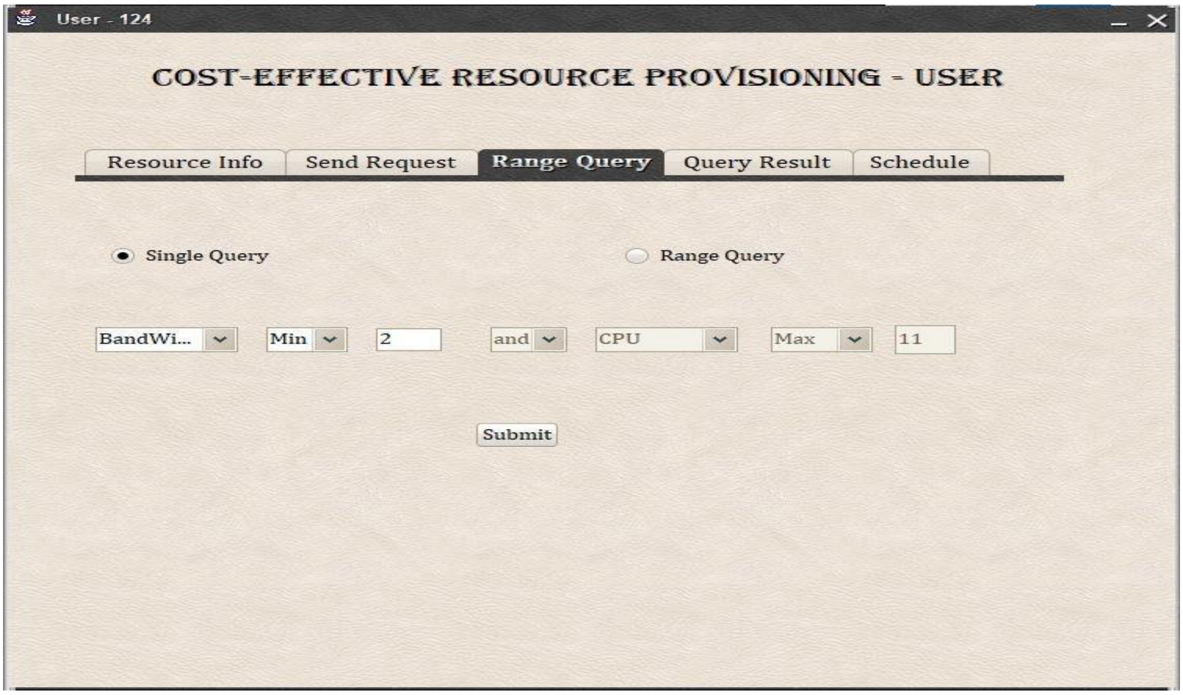



**CLOUD-EFFECTIVE RESOURCE PROVISIONING**

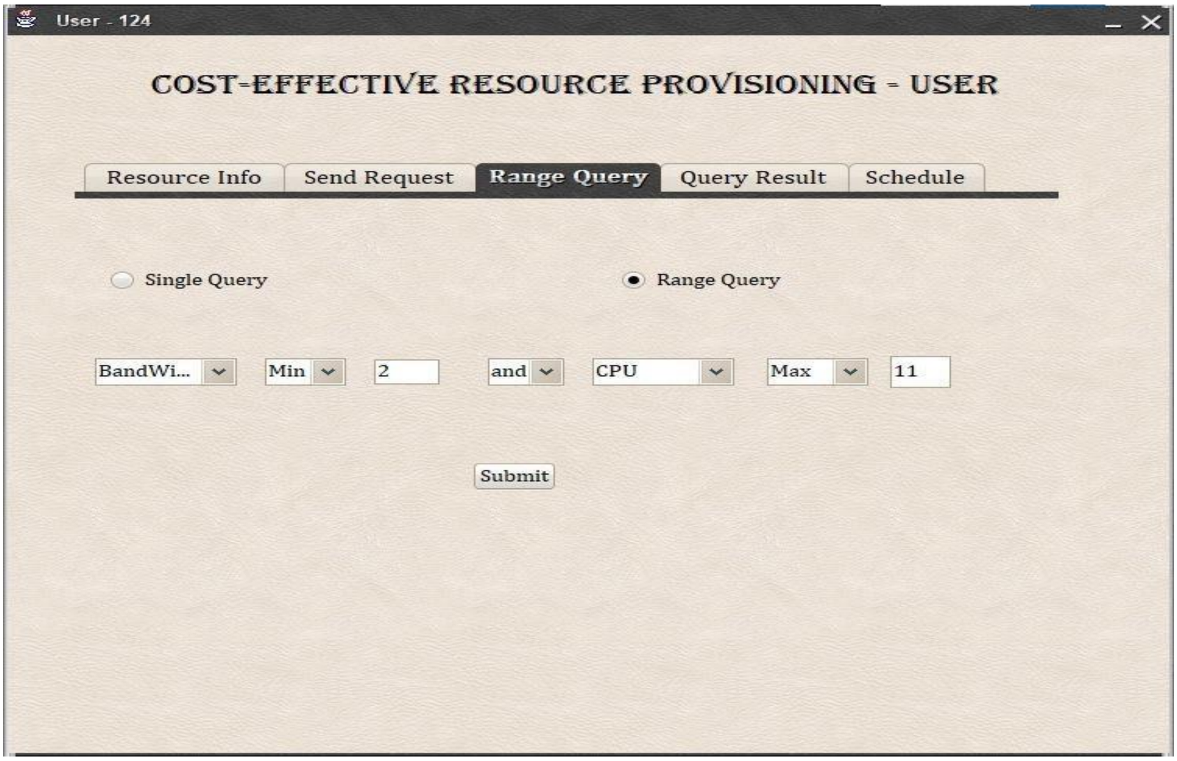

**COST-EFFECTIVE RESOURCE PROVISIONING-VM**



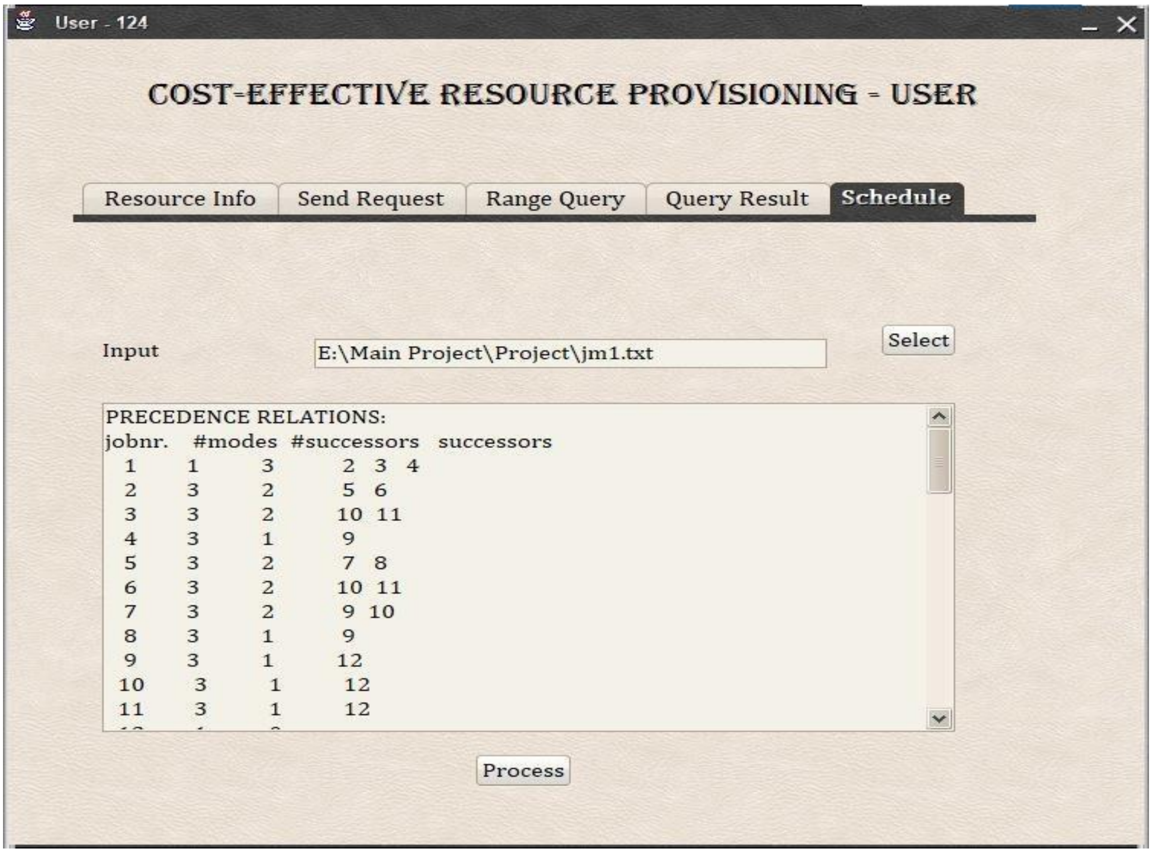

**COST-EFFECTIVE RESOURCE PROVISIONING-USER**



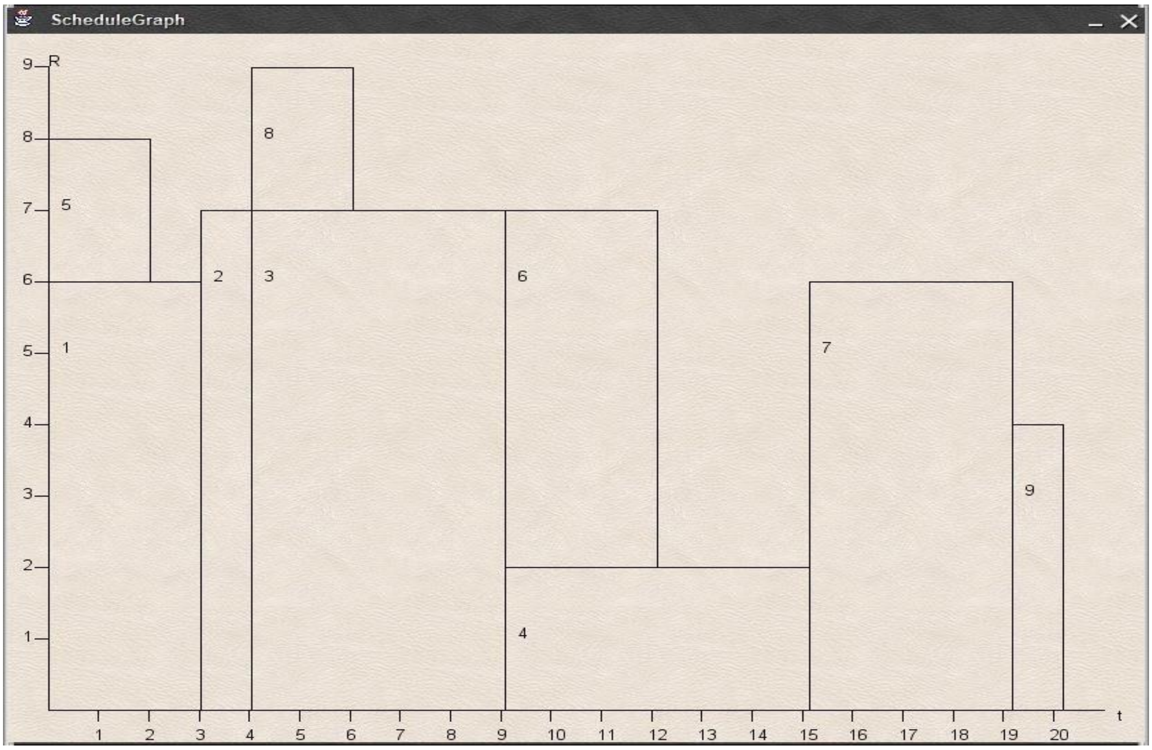

**USER-SINGLE QUERY**



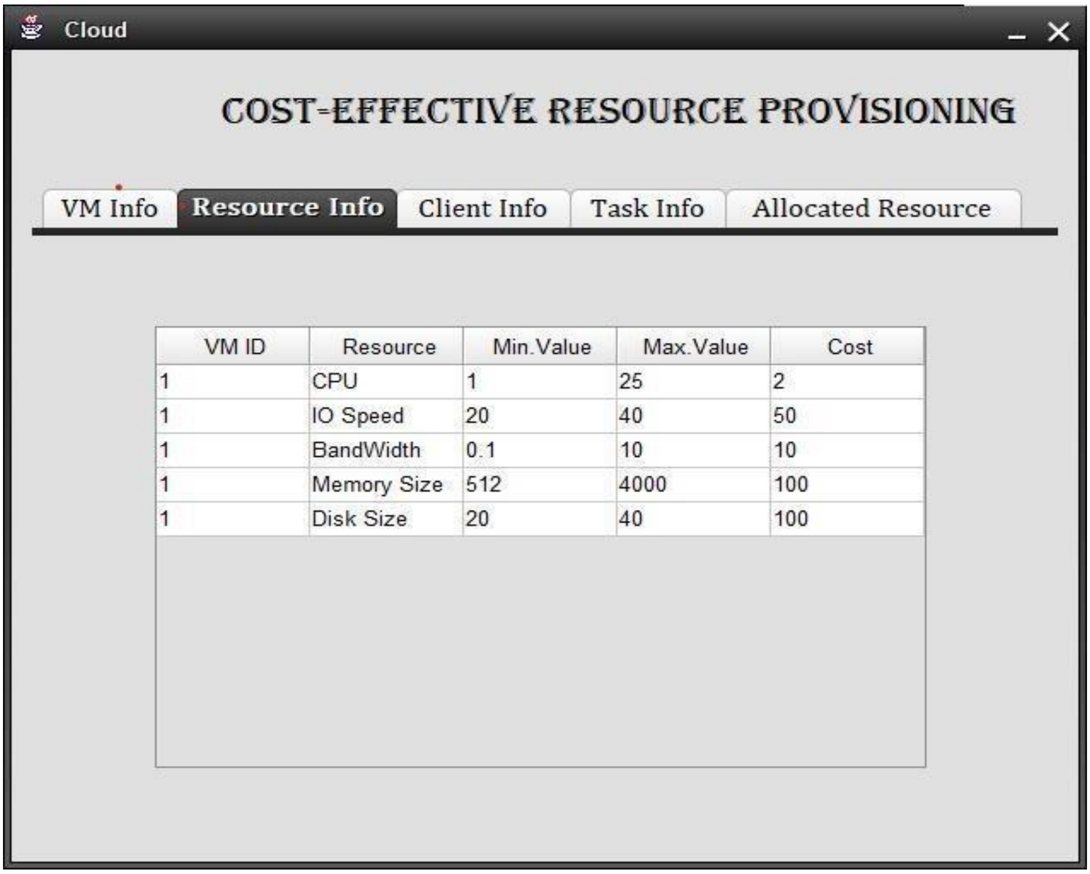

**USER-RANGE QUERY**




**SCHEDULE INPUT FOR USER**

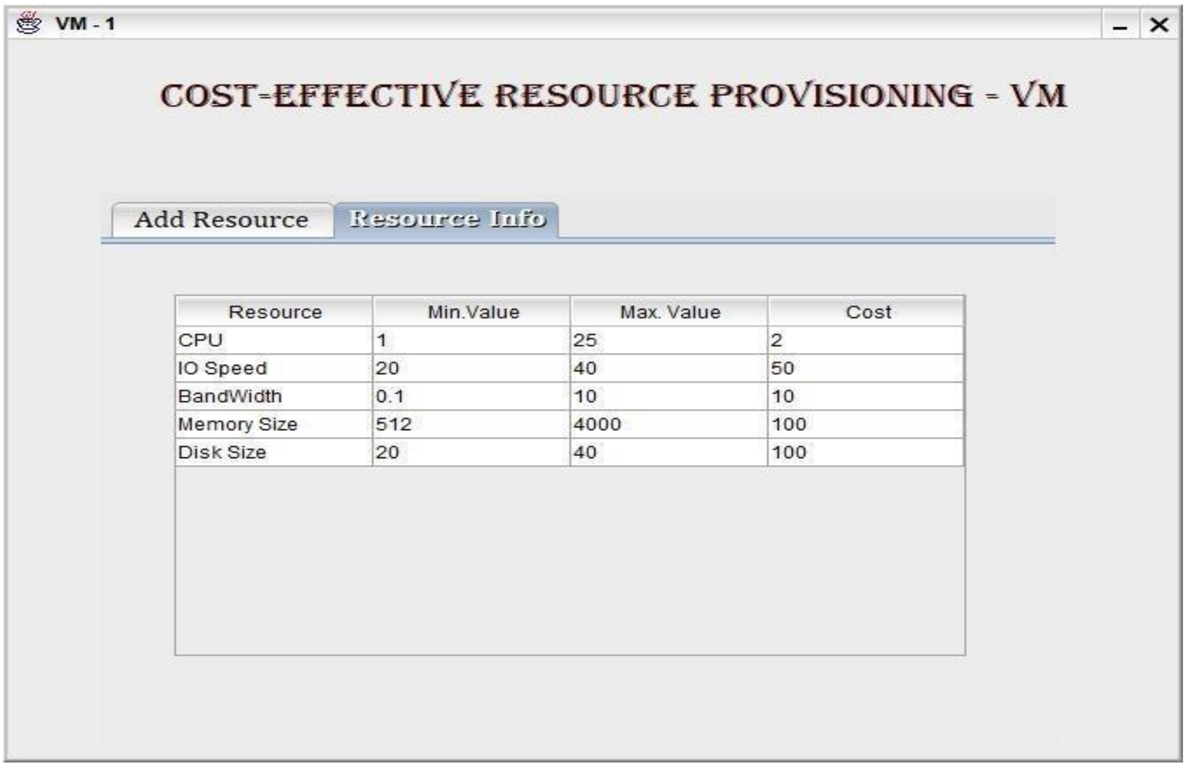



**SCHEDULE GRAPH**



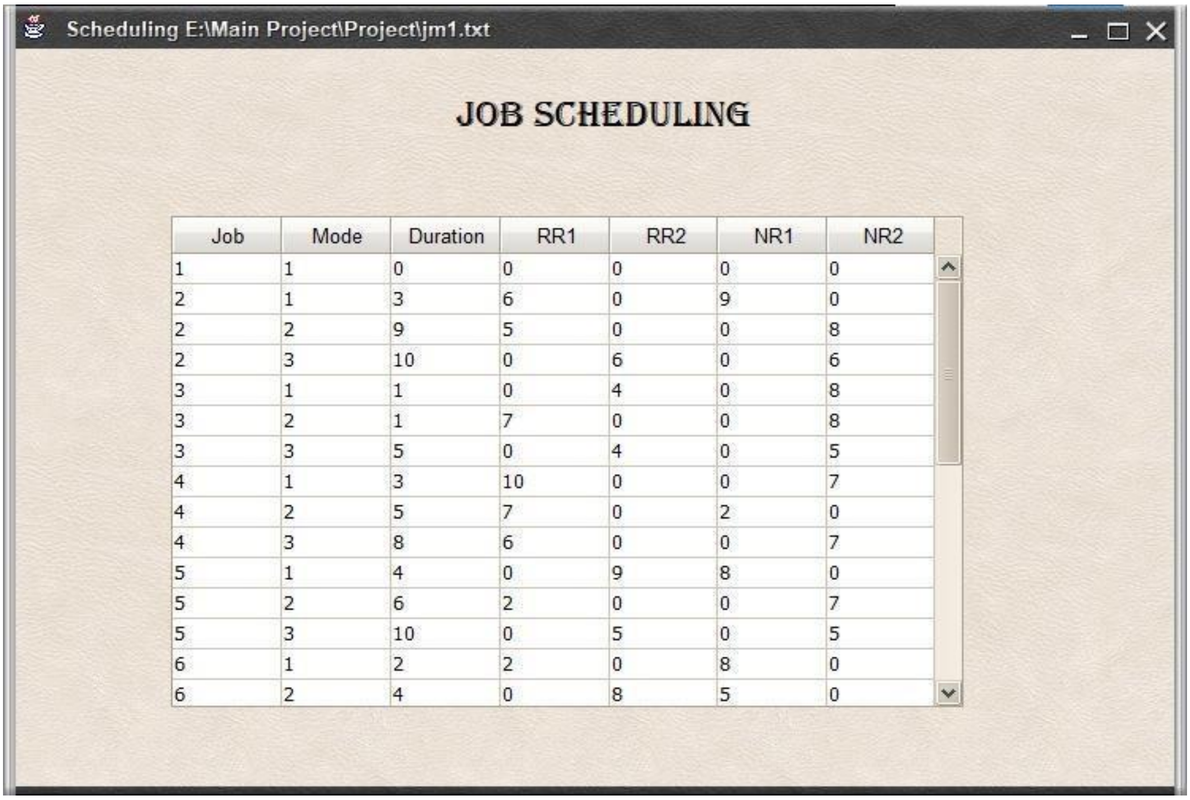

| VM ID | Resource | Min.Value | Max.Value | Cost |
|---|---|---|---|---|
| 1 | CPU | 1 | 25 | 2 |
| 1 | IO Speed | 20 | 40 | 50 |
| 1 | BandWidth | 0.1 | 10 | 10 |
| 1 | Memory Size | 512 | 4000 | 100 |
| 1 | Disk Size | 20 | 40 | 100 |

**COST-EFFECTIVE RESOURCE PROVISIONING**




| Resource | Min.Value | Max. Value | Cost |
|---|---|---|---|
| CPU | 1 | 25 | 2 |
| IO Speed | 20 | 40 | 50 |
| BandWidth | 0.1 | 10 | 10 |
| Memory Size | 512 | 4000 | 100 |
| Disk Size | 20 | 40 | 100 |

## VM RESOURCE INFORMATION




| Job | Mode | Duration | RR1 | RR2 | NR1 | NR2 |
|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 3 | 6 | 0 | 9 | 0 |
| 2 | 2 | 9 | 5 | 0 | 0 | 8 |
| 2 | 3 | 10 | 0 | 6 | 0 | 6 |
| 3 | 1 | 1 | 0 | 4 | 0 | 8 |
| 3 | 2 | 1 | 7 | 0 | 0 | 8 |
| 3 | 3 | 5 | 0 | 4 | 0 | 5 |
| 4 | 1 | 3 | 10 | 0 | 0 | 7 |
| 4 | 2 | 5 | 7 | 0 | 2 | 0 |
| 4 | 3 | 8 | 6 | 0 | 0 | 7 |
| 5 | 1 | 4 | 0 | 9 | 8 | 0 |
| 5 | 2 | 6 | 2 | 0 | 0 | 7 |
| 5 | 3 | 10 | 0 | 5 | 0 | 5 |
| 6 | 1 | 2 | 2 | 0 | 8 | 0 |
| 6 | 2 | 4 | 0 | 8 | 5 | 0 |

## JOB SCHEDULING

# REFERENCES


1. Xiaoyu Xia , Feifei Chen , Qiang He , Member, IEEE, John C. Grundy , Senior Member,Mohamed Abdelrazek, and Hai Jin(2021) “Cost-Effective App Data Distribution in Edge Computing” Transaction on parallel and Tribute Systems. vol 32,no. 1,pp. 31- 44.

2. Q. He et al., (2020) “A game-theoretical approach for user allocation in edge computing environment,” IEEE Trans. Parallel Distrib. Syst., vol. 31, no.3, pp. 515–529.

3. L. Chen, S. Zhou, and J. Xu,(2018) “Computation peer offlfloading for Energy-constrained mobile edge computing in small-cell networks,” IEEE/ACM Trans. Netw., vol. 26, no. 4, pp. 1619–1632.

4. Maciej Malawski**,** E.-K. Byun, Y.-S. Kee, J.-S. Kim, and S. Maeng, (2017) “Cost Optimized provisioning of elastic resources for application workflows” Future Gen. Comput. Syst., vol. 27, pp. 1011–1026.

5. F. Busching, G. Berriman, S. Schildt, and L. Wolf, (2017) “Cost-driven Scheduling of Grid Workflows UsingPartial Critical Paths,” in Proc. 32nd Int. Conf. Distrib. Comput. Syst. Workshop, vol. 22, pp. 114–117

6. M.Y. Arslan, S. Abrishami, Jia Yu, S. Singh, H. V. Madhyastha, K. Sundaresan, and S. V. Krishnamurthy, (2013) “Computing while charging: Building a distributed computing infrastructure --using smartphones,” in Proc. 8th Int. Conf. Emerging .Netw. Experiments Technol., vol. 05, pp. 193–204.

7. JiaYu , RajkumarBuyya, and L. Wolf, (2013)“A Taxonomy of Workflow Management Systems for Grid Computing,” in Proc. IEEE Int. Conf. Green

Comput. Commun. IEEE Internet Things and IEEE Cyber, Phys. Social Compute .vol. 11, pp. 1986–1991.

**8.** P. R. Elespuru, S. Shakya,H. Zhao and S. Mishra, (2013) “A Hybrid Heuristic for DAG Scheduling on Heterogeneous Systems” in Proc. 7th IFIP WG 10.2 Int. Workshop Software . Technol. Embedded Ubiquitous Syst., vol. 28, pp. 168–179.

**9.** E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G.Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, andD. S. Katz, (2015) “Pegasus: A framework for mapping Complexscientificworkflows onto distributed systems,” Sci. Program., vol. 13, pp. 219–237.

**10.** C. Lin and Jia Yu, (2011) “Scheduling scientific workflows elastically for cloud computing,” in Proc. IEEE Int. Conf. Cloud Comput., vol. 26, pp. 746–747.

**11.** D. Datla, X. Chen, Nayana.N.Kumar, S. Raghunandan, S. M. Hasan, J. H. Reed, B. Fette, C. B. Dietrich, J.-H Kim, and T. Bose,(2017)“Market-Oriented Cloud Computing:Vision, Hype, and Reality for Delivering IT Services as Computing Utilities” IEEE Commun. Mag., vol. 50, no. 1, pp. 144–152.

**12.** A. Iosup, RajkumarBuyya, N. Yigitbasi, R. Prodan, T. Fahringer, andD. Epema, (2015) “Performance analysis of cloud computing services formany-tasks scientific computing,” IEEE Trans. Parallel Distrib.Syst., vol. 22, no. 6, pp. 931–945.

**13.** D. Datla, H. I. Volos, S. M. Hasan, J. Raicu, and T. Bose, (2011) “Cloud Computing and Grid Computing 360-Degree Compared” Analog Integr. Circuits Signal Process. vol. 69, nos. 2/3, pp. 341–353.

**14.** D. Datla, Harold C.Lim, H. I. Volos, S. M. Hasan, J. H. Reed, and T. Bose, (2017) “Automated Control in Cloud Computing:Challenges and Opportunities” Ad Hoc Netw. Special Issue Cognitive Radio Ad Hoc Netw., vol. 10, no. 5, pp. 845–857,.

**15.** R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R.Buyya, (2011) “Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning a lgorithms,” Softw. Pract. Expert. vol. 41, pp. 23–50.

**16.** M. Mao and M. Humphrey, (2012) “A performance study on the VMstartup time in the cloud,” in Proc. IEEE 5th Int. Conf. Cloud Comput., vol. 03, pp. 423–430.

**17.** A.C. Zhou and B. He, (2013) “Transformation-based monetary cost optimizations for workflows in the cloud,” IEEE Trans. Cloud Computing.,vol. 2, no. 1, pp. 85–98.

# PUBLICATION

1. Nithya N.S., Balamurugan C, Boobalan N, Guru Vignesh K (2022), “A Cost- Effective Resource Maintenance Of Data Distribution In Edge Computing’’ inInternational Journal of Health Sciences (IJHS), Acceptance received.