Performance Evaluation of Resource Management and Task Offloading Strategies in Mobile Cloud Computing

Olle Svensson, Ingrid Dahl, Erik Lind, Maja Bergman, Karin Holm*

Department of Electrical Engineering, Karlstad University, Karlstad, Sweden
Department of Computer Engineering, Karlstad University, Karlstad, Sweden
Department of Automation Engineering, Karlstad University, Karlstad, Sweden
Department of Telecommunications Engineering, Karlstad University, Karlstad, Sweden
Department of Robotics Engineering, Karlstad University, Karlstad, Sweden

ABSTRACT

Mobile cloud computing is a developing field in parallel processing and distributed computing region. Mobile cloud computing familiarity is exponentially greater because of its characteristics like on-request benefit, versatility, adaptability, and security. Cloud encourages both computational and storage service to its clients. This decreases maintenance and deployment cost support for any organization. Therefore, cloud computing has expanded significantly. To be specific, cloud service providers (CSP) necessities the resource utilization an ideal way. To make use of resource effectively, scheduling taskplays a significant role. Scheduling helps in allocating the tasks in the cloud environment. The task scheduler orchestrates tasks in queue for accessible associated assets. Furthermore, the created portable information movement has been violently developing and has turned into a serve load on versatile system administrators. To address such a confront in versatile systems, a successful approach is to managing data traffic by utilizing advanced technologies (e.g., Wi-Fi network, small cell network, so on) to accomplish portable data offloading This course of action benefits cloud service providers to accomplish most extreme execution in cost effective way. Here, a broad investigation of some scheduling algorithm that plans to diminish the energy consumption, while assigning different tasks in mobile cloud condition is finished. The merits and demerits of these existing algorithms are further identified.

Keywords: Deployment cost, Cloud service providers, Offloading, Task scheduling

I. INRTODUCTION

Mobile cloud computing is a backbone of variety of internet business using principle of virtualization.The main objective of mobile cloud computing environment is to optimally use available computing resources. Scheduling algorithms play an important role in optimization process. Therefore user tasks are required to schedule using efficient scheduling algorithm [1]. The scheduling algorithms usually have goals of spreading load on available processors and maximizing their utilization while minimizing total execution time. Users' requests are referred to as tasks. To enhance system performance, cloud service providers need to automate task and resource mapping process. This automation is done by task scheduling algorithms. Task scheduling procedures plays major role in deciding quality of service for users [2]. The energy required to complete scheduled task varies depending on which frequency a processor is operating. When frequency is high, then energy consumption is high and tasks are executed faster. In low frequency, energy consumption is low and tasks need more time to successfully execute. Task scheduling is one of the most famous combinatorial NP complete problem problems. The main purpose of scheduling is to schedule the tasks in proper sequence in which tasks can be executed under problem specific constraints. Many heuristic optimization algorithms have been developed and solved the task scheduling in cloud environment over the years.

II. NETWORK MODEL

The cloud framework comprises of numerous data centre that are disseminated all over globe and accessible using internet. Every data centre comprises of numerous processing and components and different resources. Components in data centre are associated by a high transfer speed communication network. Consequently unimportant correspondence delay is considered in this model. In the proposed design, client can get to cloud resources utilizing UI as depicted in figure 1. The proposed task scheduling module shows effective allotment of user tasks into various accessible sources with a target to upgrade energy consumption and time.

a. Centralized Task Scheduling

In centralized scheduling, one single scheduler is accessible for whole framework to plan different tasks. Centralized scheduling is anything but difficult to actualize; the fundamental downside of centralized scheduling is that entire framework ends when scheduler quits working. Fault tolerance and scalability to non-critical

failure of centralized task scheduling are low. This single point failure issue is overwhelmed by distributed task scheduling.

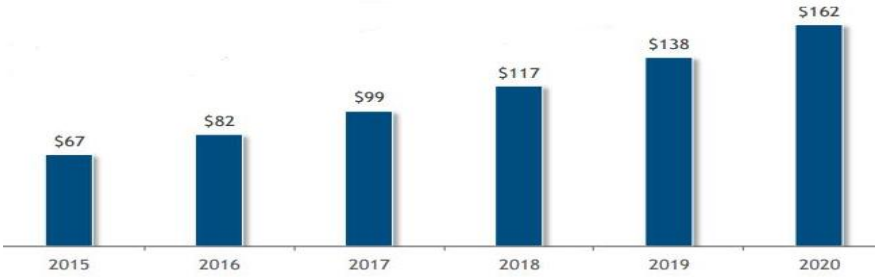

Fig 1 Graphical representation of cloud utilization over years

b. Distributed Task Scheduling

In distributed scheduling, accumulation of schedulers is associated with each other. At the point when any tasks originate from client to cloud framework, schedulers perform scheduling collectively i.e., workload is dispersed among different schedulers, hence saving significant measure of process cycle. Benefits of distributed scheduling are scheduling which exchanges workload with neighbour nodes.

III. EXISTING METHODS AND THEIR PERFORMANCE

a. Energy aware genetic algorithm

Energy aware genetic algorithm for task scheduling attempts in diminishing the energy consumption caused due to resources. Energy consumption of task is evaluated utilizing Dynamic Voltage Scaling. This calculation assumes about small sets of approaching tasks as chromosomes. These chromosomes are introduced, and fitness functions for them are computed. The selection operation is then connected on these capacities. Therefore, calculation may not be successful in adaptable circumstance.

b. Energy Saving Task Scheduling Depending on Vacation Queue

The task scheduling scheme proposes an energy saving algorithm which relies upon queue [7]. The resources in virtual machine of cloud framework are not generally being used. So when there is no task in queue, the resources go to idle state lastly, it heads out to sleep mode where energy utilization is least [8]. When tasks arrive, it continues back to alive state. This traverse is called vacation. This task scheduler works on Dynamic Voltage Frequency Scaling (DVFS) and Dynamic Power Management (DPM) advancements to lessen total utilization by a task. Each server's power consumption is decreased by Dynamic Power Management voltage, and recurrence is controlled by Dynamic Voltage Frequency Scaling. So versatility of this algorithm from client point of view isn't great.

c. Toward Energy Efficient Scheduling for Online Tasks in Cloud Data Centres Based on DVFS

A cloud cluster is an accumulation of multi-centre homogeneous servers. Every processor has two kinds of power utilization. One is static power utilization which demonstrates consistent power utilization when processor is turned on. The other is dynamic power utilization which is reliant on frequency of the processor in which it is executing. This dynamic power can be controlled by the framework.Power utilization is corresponding to frequency of the processor in which it is working. From this, it can be determined that when a processor is running in high frequency then energy utilization by the processor is high.It is hard to isolate the centres of a multi-core processor and break down their frequency. They additionally accepted that deadline of a task will be analyzed when frequency of the processor is decreased to save power utilization.

d. An Innovative Energy-Aware Cloud Task Scheduling Framework

DVFS and VM reuse procedures are consolidated to control dynamic energy utilization, and tasks' deadline won't be influenced by their system. This scheduler structure is effectively adoptable and inventive one. Energy utilization of data centres can be limited by this scheduler [9] & [11]. Complicated algorithms should be executed instead of FFD or WRR (First Fit Decreasing (FFD) or Weighted Round Robin (WRR)) to get the fulfilling results.

e. Energy Aware Scheduling of HPC Tasks in Decentralized Cloud Systems

This algorithm spotlights to work in multi-cloud frameworks and where datacentres are decentralized. It takes its mapping decision by assuming the resource status. Tasks are organized in Directed Acyclic Graph (DAG) where task demand may have a few successors and predecessors. The static energy and dynamic energy devoured by processor is the aggregated energy utilization of processor [10]. Dynamic energy utilization is

subject to frequency on which the scheduler is running. In multicore framework for every core, energy utilization is computed. The deadline is considered for scheduling it just checks if task will met its deadline amid task allocation or not. It doesn't give higher priority for tasks with lesser deadline.

f. Activity-Based Costing Algorithm

This algorithm computes a priority list for each task that arrives in cloud framework. This priority is figured by considering number of resources required to finish the task, cost for getting to every resource [3], and the benefit that cloud service provider can gain on successfulfulfilment of that task. The ABC calculation does not think about time as a parameter. Thus, if there is large number of tasks in the HIGH list, the tasks in the LOW rundown need to wait for more extended time period. Such tasks may miss deadline bringing about client disappointment and hampering QoS.

g. Improved Activity-Based Costing Algorithm

This algorithm additionally considers about cost of individual resources. Like ABC, even here service provider's sets minimal effort for simpler and often executed tasks and for unique tasks the cost is set to high value. Total number of Million Instructions (MI) of each group must be not exactly or equivalent to the resource's execution limit. This limit is increase of resource's MIPS (Million Instruction for every Second resource can deal with) and prefixed granularity size. At the point when this defined criterion is coordinated, at that point the gathering is allotted to that resource.

h. Double Level Priority-Based Optimization Algorithm

Another sort of task scheduling system is proposed which work contingent on two levels of priorities. The task is put into the Fully Available category list if resources required by tasks are accessible in one data centre [5]. Nonetheless, the task is put into the Partially Available classification list if the required resources exist in excess of one data centre. Resources are arranged in turnaround time, and most elevated need task is allocated resource first.

i. Cost–Deadline-Based Algorithm

The scheduler allots priority for each tasks landing for execution and places them in queue. Admission to this queue is dictated by ascertaining tolerable delay and cost of service. This calculation is called as Task Priority Delay (TPD) calculation since three parameters are considered in this approach. They are due date, cost, and task length. To produce the tasks' deadline, priority queue, and task length is considered as in table I.

Table I: Existing schedulers with its merits and demerits

S.NO	SCHEDULERS	MERITS	DEMERITS
1	Energy aware GA	Power consumption is reduced	Other factors like deadline, cost makespan are not considered for better QoS to its users. GA's solution is bounded to local maxima.
2	Energy saving task scheduling on vacation Queuing	Energy consumption is reduced and moves to sleep mode when no task is available Makespan is reduced in this scheduling algorithm thus giving better efficiency for the system	Task performance is not considered. Needs better QoE value
3	Toward energy efficient scheduling for online tasks in cloud data centres based on DVFS	Energy consumption is low	Difficult to get homogenous servers. Deadline of task is assumed to be adjusted.
4	Innovative Energy-Aware Cloud Task Scheduling Framework	Easy to implement Task execution cost is minimized	The framework is not scalable. Deadline parameter is not considered. Task reaching deadline cannot be handled efficiently.

5	Energy aware scheduling of HPC tasks in decentralized cloud systems	Energy consumption is reduced Deadline miss is partly decreased	Does not follow prioritized scheduling.
6	ABC	Both resource cost overhead cost can be minimized	Algorithm may not be efficient in overhead. There is no time parameter.
7	Improved cost-based algorithm	Total profit is Increased Makespan is reduced	Tasks with lower costs have to wait long Quality Of Service attributes are not considered.
8	Double level priority-based Optimization	Cost is reduced Completion time improved	Deadline of task is not considered. Energy parameter for resources is not considered.
9	Cost–deadline-based Algorithm	Deadline of task is considered It predicts initial cost of task	Cost optimization is not up to Level. Low priority tasks may have to wait long due to availability of large tasks with high precedence. Non-pre-emptive type.

IV. OFFLOADING TECHNIQUES IN CLOUD ENVIRONMENT

The idea of mobile cloud computing includes offloading of task that will be executed by remote server [15]. Application that should be offloaded from mobile phone to cloud should be possible in two designs i.e., partial offloading or full offloading. In full offloading design, full application related to it has been offloaded to cloud where whole calculation happen and last outcomes has been sent back to mobile phone.In partial offloading architecture, application that consumes more vitality or has unpredictability regarding calculation has been offloaded to cloud. In this both mobile phone and cloud are responsible for calculation and last outcomes subsequent to blending outcomes of two calculations in mobile phone and at cloud. The partial offloading calculates tasks have been unpredictable because of lack of resources identifying with programming, infrastructure and so forth. Additionally, runtime complexities could likewise be stayed away from if code is offloaded to cloud for execution.

a) Offloading Models

i) MAUI Architecture

The MAUI architecture gives a structure to offloading from mobile phone to advanced mobile phone for saving energy. It has been recognized that the code ought to be offloaded or it ought to be executed locally, MAUI profiler is responsible for it. The choices of offloading is taken subsequent to considering factors like cell phone's example of energy consumption, running time of projects, accessible transfer speed and so forth.

ii) Clone- Cloud Based Models

The clone cloud framework has been produced for offloading from mobile phone to cloud naturally; MAUI doesn't need software engineer's assistance. It includes execution offloading in which offloading of process happens amid runtime. Clone cloud includes dynamic code partitioning, and state movement. At runtime code has been divided that resource accessibility is known around then and henceforth better choices could be taken to enhance migration cost and energy utilization.

iii) Cuckoo Design

The cuckoo configuration has been proposed for offloading from advanced mobile phones that keep running on android stage to cloud. This model assists both local and on-cloud execution of code contingent upon accessible resources [4]. Cuckoo resource supervisor screens reachable resources and give one if conceivable. Any framework that utilizations cuckoo model could offload to remote server that has java VM.

iv) MACS Architecture

The framework screens accessible resources and provides advanced solution that in the case of offloading should occur or not. This application could keep running on android telephone however it is likely to be offloaded to cloud on request premise. The framework focuses around accomplishing three components for effective and successful offloading which includes that memory use ought to be least, energy expended ought to be least and execution time ought to be least.

v) AHP and TOPSIS Techniques based Design

There is method that was proposed for cloud-path selection, which chooses which cloud ought to be offloaded on premise of analytical hierarchy process (AHP) and strategy for arrange inclination by likeness to ideal solutions (TOPSIS). AHP considers multi-criteria issue and it examinations and discover relative weight of all parameters to be considered for cloud selection accessibility, security, speed, and cost and transmission capacity. Fuzzy TOPSIS system is utilized here to discover ranking of clouds. The decision making has done by considering fuzzy approach. It has been observed that if information protection isn't prime concern then parameter of security doesn't assume important role while choosing cloud.

vi) Energy Aware Design for Workflows

It has been watched that offloading is influenced by two parameters:

- Network Connectivity& Communication Size
- Cloudlet Processing Speed& Computation Size

To enhance application execution and to save energy, an approach has been introduced in which mobile work process has been offloaded to closest accessible cloudlet. If more than two tasks have been offloaded to same cloudlet for if work flow is being spread on more than one phones then it could save energy as well established route could be thought about over and over.

vii) MCSOS Architecture

A structure has been proposed for offloading from smart phones to close-by cloudlets to enhance energy utilization called mobile cloud with smart offloading framework. MCSOS structure exploits Map Reduce programming model for enhancing energy consumption. For improving execution, master node i.e. advanced mobile phone, advances task to benefit service broker which isolates slave nodes around master hub for reason for parallel calculation of job. The slave nodes could run tasks locally or it could offload it to calculation related cloud (CA-cloud) contingent upon resources.

viii) Secured Offloading Design

A system has been proposed for elastic application that could have in excess of one cloudlets i.e., between smart devices and remote server. The model gives security to security for offloading on two sides that are cell phone and cloud on factors like confirmation to give secured environment of communication, secured migration of cloudlets and to approve cloudlets of cloud to utilize outside web services.

b) Requirements of Offloading

In cloud offloading three fundamental models are required named as cost display, energy model and weighted model. Prediction is critical in offloading in the event that takes some wrong forecast it produces a few errors in processing. Energy model is utilized to compute energy gain of offloading. It is characterized as aggregate sum of work performed over timeframe. Offloading of given tasks was computed. After this computation, perform offloading. Cost model calculate the total cost of offload application and furthermore ascertain memory gain of offloading.

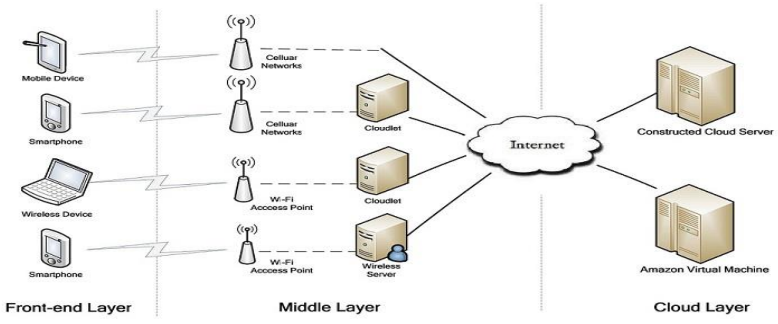

Fig 2Layers of cloud offloading

c) Cloudlet Architecture

A Droid Cloudlet device is either client device, which is any cell phone running an application that requirements to utilize any accessible additional resources; or a server device, which is any cell phone in vicinity of other customer mobile phones and having additional resources that it can give. Server Profiler, offloading Agent and Class Loader are significant modules that coordinate interactions amongst customers and servers, while different modules take assistant parts that help their capacities.

i. Server Profiler

The Server Profiler is in responsible for evaluating device resources and choosing whether it can take server part or not. It gets estimations of resources gathered by Resource Monitor, it makes decision based predefined Server Nomination Policies. Device prevailing to be server, Server Profiler registers designated device alongside its resources' estimations with help of Directory Service. Advanced device has adequate resources, its Server Profiler continues refreshing registry with new estimations of its resources' parameters.

ii. Offloading Agent

Offloading Agent lies amongst application and OS. Its significant part is to capture each application call to any remotable class and choose, as indicated by predefined offloading Policies, regardless to offload it to chosen server(s) or pass it to nearby OS. This block attempt happens just when device resource parameters are under threshold values. To settle on offloading choice, offloading Agent conjures Cost Model that evaluates remote and local performance called class. Cost Model expends readings from Resource Monitor, Energy Measurement and Historical Results Repository modules. To get data about cloudlet's resources, offloading Agent gets rundown of dynamic servers alongside their resources' parameters with help of Directory Service.

iii. Class Loader

The Class Loader keeps running on server when it is summoned by customer. Its significant part is to get classes and their parameters from offloading Agent of client, execute them on server's OS, lastly send comes about back to offloading Agent of client. There are two situations for task. If there should arise an occurrence of Private/Trusted Environment, Class Loader gets remote call from offloading Agent for server's pre-installed class. Just fully qualified class name and its parameters are required. Open/Un-trusted Environment, Class Loader gets pre-compiled (.dex) class alongside its parameters and executes it on server.

iv. Resource Monitor

Resource Monitor module is found in each Droid Cloudlet gadget. Its primary part is to device resources assets and convey them the request. Monitored resources are: CPU utilization rate, accessible memory, Wi-Fi accessibility alongside its average round trip latency to closest access point, battery level and whether it is charging or not.

v. Energy Measurement Module

Energy Measurement module is responsible of estimating energy consumed by local resources (CPU and Wi-Fi) while running application. These estimations are utilized for evaluating energy utilization while applying Cost Model. Neighbourhood energy estimations amid offloading sessions are additionally recorded by offloading Agent, which spares them into Historical Results Repository for later use by Cost Model to settle on offloading decisions.

vi. Historical Results Repository

This is repository that is sustained by offloading Agent after each Remotable class execution, with its execution result, regardless of whether it was local & remote. The major fields of each record in this store are: execution area, total execution time, transferring time, downloading time, holding up time, add up to energy utilization, transferring energy, downloading energy, holding up energy, sending size and results estimate.

vii. Offloading& Server Nomination Policies

DroidCloudlet is obliged by strategies for designating devices to be servers and for offloading Remotable class from feeble client to all the more power server(s). At first, all devices in DroidCloudlet are considered as independent devices [14]. In runtime, resources' parameters are assessed and those devices with predefined resource' qualities are assigned to play the part of servers. These attributes are characterized into Sever Nomination Policies when an application running on stand-alone device invoke Remotable class, offloading Agent checks for device resources; in the event that they are under predefined esteems, it observes that most intense accessible server(s), resource parameters esteems are over predefined esteems. Offloading advantage is evaluated utilizing Cost Model. These predefined values are characterized in offloading Policies.

viii. Directory Service Module

To dispose of overhead of distributed servers' disclosure procedure, centralized directory of dynamic servers exists inside cloudlet. This directory is overseen by Directory Service Module, which is light weight benefit that can be facilitated on any device inside cloudlet, and serves every single other devices [12]. Every single assigned server registers their IPs and their asset parameters inside this directory. All customers can look into this directory to discover accessible dynamic servers.

d) Mobile data offloading techniques

Data offloading procedures are arranged into four classifications, i.e., offloading through small cell systems (SCNs), Wi-Fi systems, opportunistic mobile networks or heterogeneous systems (HetNets), individually [16].

i) Data Offloading Through Small Cell Networks

Data offloading in cellular organizes by utilizing small cells, and extricated the primary necessities of content delivery that need to incorporate data offloading abilities into the present portable networks.

ii) Data offloading through Wi-Fi networks

Mobile data offloading plan by leasing remote data transfer capacity and reserve space of private 802.11 (Wi-Fi) APs, which can adequately diminish network congestion and enhance the client perceived network execution without over-overloading APs' backhaul links.

iii) Data offloading through opportunistic mobile networks

The opportunistic mobile network based mobile data offloading issue by numerous practical assumptions, e.g., content size, content lifetime, nodes' subscribing interests and constrained buffer size of nodes.

iv) Data offloading through heterogeneous networks

An ideal Opportunistic Mobile Network-based offloading plan is intended to enhance organize limit and nodes' energy effectiveness in Heterogeneous Wireless Networks. To limit the general cost through cellular systems [13], they proposed a model to offload a bit of the mobile data to Wi-Fi and Opportunistic Mobile Networks.

Table II: Mobile data offloading techniques and findings

S.No	TECHNIQUES	FINDINGS
1	Data Offloading Through Small Cell Networks	Content caching
2	Data offloading through Wi-Fi networks	Architecture and models Utility & costs Framework design
3	Data offloading through opportunistic mobile networks	Seed selection Load allocation Framework design
4	Data offloading through heterogeneous networks	Caching policy Content dissemination Framework design

e) Resource allocation

Resource management is an umbrella action including distinctive phases of resources and workloads from workload accommodation to workload execution. Resource management in Cloud incorporates two phases:

i) Resource provisioning
ii) Resource scheduling.

Resource provisioning is characterized to recognize sufficient resources for given workload in view of QoS prerequisites portrayed by cloud consumers while resource scheduling is mapping and execution of cloud consumer workloads in light of chosen resources through resource provisioning [6]. Resource scheduling is hotspot region of research in cloud because of substantial execution time and asset cost.

i) Need of Resource Scheduling

The primary goal of resource scheduling is to distinguish reasonable resources for planning proper workloads on time and to expand resource utilization effectively. Resources ought to be least for workload to maintain a

required level of service quality, or limit workload completion time (or boost throughput) of workload. For better resource scheduling, best resource workload mapping is required. The second target of resource scheduling is to distinguish suitable and adequate workload that assists of different workloads, to be competent to satisfy various QoS prerequisites, for example, CPU usage, accessibility, unwavering quality, security and so on for cloud workload. Along these lines, asset booking considers execution time of each unmistakable workload, however in particular, general execution is additionally in light of kind of workload i.e. with various QoS necessities (heterogeneous workloads) and with comparative QoS prerequisites (homogenous workloads). Resource scheduling for cloud is a procedure of dynamic allotment of assets to cloud workloads after resource provisioning.

i) Compute Resource:

The core processing capabilities are utilized to execute programming instructions in cloud. It involves CPU, multi-centre configuration, CPU cache and essential storage memory. Data centres farms regularly house a large number of servers containing compute resources.

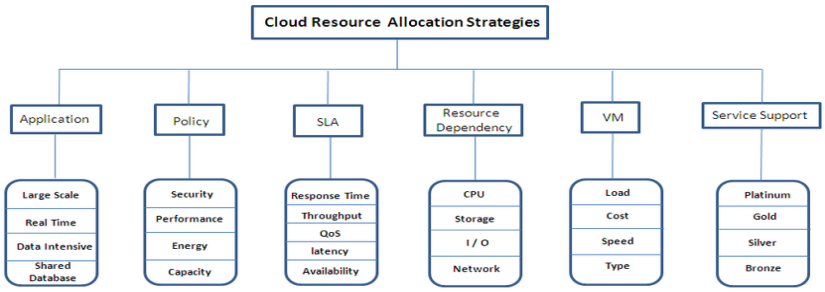

Fig 3 Flow representation of cloud resource allocation strategies

ii) Storage Resource:

Non-volatile secondary storage memory houses the data used by compute resources. This resource is typically cheaper than primary memory; many operating systems are able to use it as an extension of main memory, to temporarily swap out unused memory state. Many data centres will have server with access to internal storage as well as to Storage Area Network that consolidate and abstracts complexity of accessing storage throughout data centre.Non-volatile secondary storage memory houses the information utilized by compute resources. This resource is normally less expensive than essential memory; numerous operating frameworks can utilize it as an extension of main memory, to incidentally swap out unused memory state. Numerous data centres will have server with access to inner capacity and in addition to Storage Area Network that solidify and modified complex accessing storage all through data centre.

iii) Network Resource:

It incorporates network cards that interface into servers and in addition infrastructure elements that incorporate repeaters, load balancers, switches and firewalls. Networks can utilize distinctive topologies and protocols, which impact level of security, flexibility and Quality of Service.

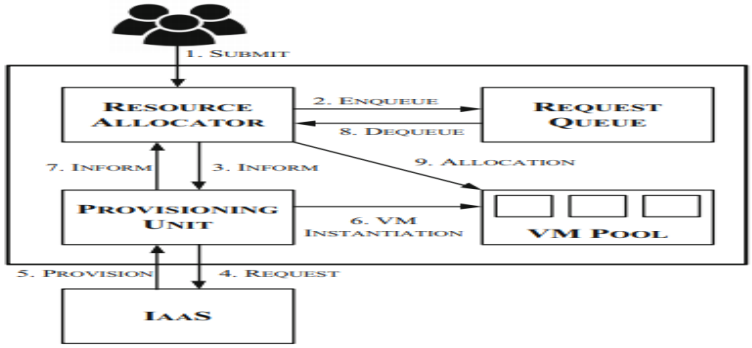

Fig. 4 Block diagram of resource allocation in cloud

iv) Virtual Resource

It is a deliberation included onto compute, network resources and storage. It empowers slicing resources into littler chunks that can be scaled vertically or on a level plane. Normally virtualisation is utilized as a part of data centre to cut data centre calculate resource into Virtual machines, and possibly to show logical processors by mapping these onto a solitary physical processor. Network cards and capacity are additionally virtualised and introduced as individual devices to VMs.

v) Service Management Resource (SMR)

It is a knowledge library where IPs store management objectives, policies, evaluating and coordination data.

vi) Management Tools

They are utilized by IPs to arrangement, screen, and reconfigure reinforcement and re-establish the framework.

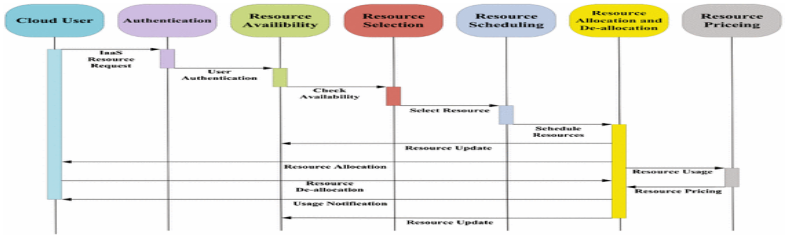

Fig 5 Time slots of resource allocation

f) Resource allocation algorithms

i) Ant Colony Optimization algorithm

Ant Colony Algorithm depends on conduct of ants gathering food. Ants groups to pursuit and gather food dependably. The essential standard of this calculation is conduct of ant insect while moving from source i.e. source to destination i.e. nourishment and from source to destination. This calculation checks accessible resources, select set of optimal nodes in investigating factors like response time and after that allots the employments to suitable nodes.

ii) Bee's Algorithm

This algorithm is based on action of bee's to get their food. In this algorithm, a meta-scheduler finds a job with lowest memory, input-output, and processor requirement. This job acts as a scout bee to find suitable site. The scout job is sent to location at which task requires resource at present. The scout job identifies location using fitness function. This fitness function runs task in particular instance and evaluates that task is memory dependent or processor dependent. Fitness is progress of particular job with assigned resources. After identifying resources and location, scout job returns back to meta-scheduler and perform waggle function. Waggle function segregates tasks which are present in meta-scheduler on basis of information provided by scout job such as cost, processor and memory requirements.This calculation depends on activity of honey bee's to get their nourishment. In this procedure, a meta-scheduler finds job with least memory, input-output, and processor necessity. This activity goes about as a scout honey bee to discover appropriate site. The scout job is sent to location at which task requires resources. The scout work recognizes location utilizing fitness function. This fitness function runs with specific instances and evaluates memory dependent task or processor dependent. Fitness is advance of specific occupation with allottedresources.

iii) Priority Algorithm

The dynamic resource allocation for pre-empt capable occupations in cloud is portion of resources to clients as per their requests, Priority based scheduling calculation performs superior to Cloud min-min scheduling procedure. In priority algorithm when job arrives at Cloud scheduler, it partitions it tasks as per their conditions and after that calculation is called to form lists as per their priorities. The calculation chooses suitable virtual machine and asset and assigns them to tasks in list.

iv) Bin-Packing Algorithm

Bin Packing problem involves objects packing of provided size into bins of given capacity. In one-dimensional bin packing size of each object is real number among 0& 1 and size of every bin is similar, provided that sum of objects in bin must not exceed 1. Bin packing algorithms utilized best fit procedures of resources in cloud. A formal BPP definition can be distinct as providedobjects list and their bin size, weights, find least numb of bins so that all resources are allocated to bin.

v) First Come First Serve Algorithm

Job in queue thatcomes first is served. This procedure is fast and simple.

vi) Round Robin algorithm

In round robin planning, forms are dispatched in FIFO way and provided with constrained measure of CPU time called time-slice or quantum. A procedure does not finish before its CPU-time lapses, CPU is acquired and given to next process holding up in queue. The pre-empted procedure is then put at back of ready list.

vii) Min–Min algorithm

This algorithm selectslesser tasks to be executed initially, which in turn huge task delays for long time.

viii) Max – Min algorithm

This algorithm selectshuge tasks to be executed initially, which in turn small task delays for long time.

ix) Most fit task scheduling algorithm

In this algorithm task which fit best in queue are implemented first. This procedure has high failure ratio.

x) Priority scheduling algorithm

The essential idea is simple: everyprocedure is assigned a priority, and priority is authorized to run. Equal-Priority procedures are scheduled in FCFS order. Shortest-Job-First (SJF) algorithm is of common priority scheduling procedure. An SJF algorithm is merely a priority algorithm where priority is inverse of (predicted) subsequent CPU burst. Longer CPU burst, lower priority and vice versa. Priority can be distinct either internally or externally. Internally proved priorities utilizecertain measurable quantities or qualities to calculate priority of procedure.

xi) Resource-Aware-Scheduling algorithm (RASA)

It is collected of two traditional scheduling procedures; Min-Min and Max-Min. RASA uses Max-min and Min-min algorithms advantages and covers their disadvantages. Deadline of every task, arriving task rate, task execution cost on every resource, communication cost are not measured.

xii) RSDC (Reliable Scheduling Distributed In Cloud Computing)

Major Job is partitioned to sub jobs. To poisejobs, request and acknowledge times are computed separately. Scheduling of every job is done by computing request and acknowledges time in shared job. Consequently, efficiency of system is enhanced.

xiii) An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment

Priority is allocated to every admitted queue. Admission of every queue is selected by computing service cost and tolerable delay. The proposed cloud architecture has acquired very high (99%) service completion rate with guaranteed QoS. It provides highest precedence for highly paid user service-requests, complete servicing cost for cloud also enhances.

xiv) A Priority based Job Scheduling Algorithm in Cloud Computing

This scheduling algorithm consist of three level of scheduling: object level, attribute level and alternate level. In this algorithm priority can be set by job resource ratio. Then priority vector can be compared with each queue. This algorithm has higher throughput and less finish time.

xv) Extended Max-Min Scheduling

Extended Max-min algorithm is grounded on expected execution time. Petri nets are utilizedon concurrent nature of distributed systems. Max-min illustratesattaining schedules with contrast lower makespan rather than RASA and unique Max-Min.

xvi) An Optimistic Differentiated Job Scheduling System for Cloud Computing

One web application is generated to do certain activity such as file uploading and downloading and a need of proficient job scheduling procedure. Qos requirements of cloud computing user and highest profits of cloud computing service provider are acquired with this algorithm.

xvii) Improved Cost-Based Algorithm for Task Scheduling

This scheduling algorithm partitions all user tasks based on priority of every task into three diverse lists. This scheduling algorithm calculates both computation performance and resource cost, it enhances computation/communication ratio.

xviii) Performance and Cost evaluation of Gang Scheduling in a Cloud Computing System with Job Migrations and Starvation Handling

This scheduling strategy can be efficiently deployed on Clouds, and cloud platforms can be feasible for HPC or high recital enterprise applications.

xix) Heuristic-PSO based Scheduling Algorithm (H-PSO)

It decreasesmakespan and enhances memory utilization. The Heuristic-PSO techniquemaximizesprocessing speed and provides an optimal solution.

xx) Hybrid Cuckoo Algorithm

Hybrid Cuckoo algorithm is anamalgamation of Cuckoo search and Genetic Algorithm. Itenhances resource utilization and decreases energy consumption.

xxi) Credit Based Scheduling Algorithm

Cloud providers have limited resources, and to maximize utilization. This scheduling producedinitiates user priority and task length. Credits are allocated to task priority and task length. This algorithm works more efficiently than previous methods. This scheme is enhanced to consider other factors such as deadline.

Table III: Existing scheduling algorithm and its findings

S.NO	SCHEDULING ALGORITHM	SCHEDULING METHOD	SCHEDULING PARAMETER	Findings
1	Resource-Aware Scheduling algorithm (RASA)	Batch Mode	Make Span	Utilized to reduce makespan
2	RSDC (Reliable Scheduling Distributed In Cloud Computing)	Batch Mode	Processing time	It is used to reduce processing time.
3	An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment	Batch Mode	Quality of Service, Service request time	It is efficient for load balancing.
4	A Priority based Job Scheduling Algorithm in Cloud Computing	Dependency mode	Priority to each queue	High QoS
5	Extended Max-Min Scheduling Using Petri Net and Load Balancing	Batch Mode	Load balancing, finish time	High throughput
6	An Optimistic Differentiated Job Scheduling System for Cloud Computing	Dependency mode	Quality of service, Maximum profit	Less finish time
7	Improved Cost-Based Algorithm for Task Scheduling	Batch Mode	Cost, Performance	Utilized for load balancing.
8	Performance and Cost evaluation of Gang Scheduling in a Cloud Computing System with Job Migrations and Starvation Handling	Batch Mode	Performance, Cost	To remove limitation of max-min procedure
9	Earliest Feasible Deadline First	Deadline Based Scheduling	To reduce time complexity	Used for Real Time Systems utilization
10	A Priority based Scheduling Strategy for VM Allocation	Priority Based VM Scheduling	Priority of Jobs	Maximize service provider and enhance resource utilization
11	Heuristic-PSO based Scheduling Algorithm	Particle Swarm Optimization Technique	Number of Particles in Swarm, Number of iterations individual ability & Learning rate regards to social ability	Reduce makespan and enhance memory utilization.
12	Hybrid Cuckoo Algorithm	Genetic Algorithm and Cuckoo Search	Priority, Workload Type, Temperature	Increase resource utilization and

		Algorithm	Capacity of Machine.	decrease energy consumption.

V. CONCLUSION

Scheduling is the most significant tasks in mobile cloud computing situation. This paper have examineddiverse scheduling algorithm and tabulated diverse parameter. Existing scheduling procedureprovides cost effective and high throughput but they do not consider consistency and accessibility. Consequently,designing of novelalgorithm enhances reliability and availability in cloud computing environment. As well, Mobile data offloading is still very novel and hot research area. Mobile data offloading is an effectual approach to ease the burden of cellular network, which not only offerseffectual mobile data management solutions to service providers for relieving congestion, but it offers new business chances. The methods involved strategies of mobile offloading and comparison with other techniques to evade the drawbacks

REFERNCES

[1] Xiao, Jing, &Zhiyuan Wang. "A Priority Based Scheduling Strategyfor Virtual Machine Allocations in Cloud ComputingEnvironment" Cloud and Service Computing (CSC), 2012 InternationalConference on IEEE, 2012.

[2] Behzad, Shahram, Reza Fotohi, and Mehdi Effatparvar. "Queue basedJob Scheduling algorithm for Cloud computing" International ResearchJournal of Applied and Basic Sciences ISSN (2013): 3785-3790.

[3] Gupta, Gaurav, et al. "A simulation of priority based earliest deadlinefirst scheduling for cloud computing system" Networks & SoftComputing (ICNSC), First International Conference on. IEEE,2014.

[4] Aujla, Sumandeep, and AmandeepUmmat. "Task scheduling in CloudUsing Hybrid Cuckoo Algorithm" International Journal of ComputerNetworks and Applications (IJCNA) 2.3: 144-150.

[5] Li, Ji, LonghuaFeng, and Shenglong Fang "A greedy-based jobscheduling algorithm in cloud computing" Journal of Software 9.4(2014): 921-925.

[6] Huang L, Chen H, Hu T (2013) “Survey on resource allocation policy and job scheduling algorithms ofcloud computing” J Softw 8(2):480–487

[7] Shuja J, Bilal K, Madani SA, Othman M, Ranjan R, Balaji P, Khan SU (2014) Survey of techniques andarchitectures for designing energy-efficient data centers. IEEE Syst J 99:1–13

[8] Ahmad RW, Gani A, Hamid SHA, Shiraz M, Yousafzai A, Xia F (2015) “A survey on virtual machinemigration and server consolidation frameworks for cloud data centers” J NetwComputAppl 52:11–25

[9] Cheng, C., Li, J., Wang, Y. “An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing” Tsinghua Sci. Technol. 20(1), 28–39 (2015)

[10] Alsughayyir, A., Erlebach, T “Energy aware scheduling of HPC tasks in decentralized cloud systems” In: 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), pp. 617–621 (2016)

[11] Alahmadi, A., Che, D., Khaleel, M., Zhu, M.M., Ghodous, P.“An innovative energy-awarecloud task scheduling framework” In: 8th IEEE International Conference on Cloud Computing(ICCC), pp. 493–500 (2015)

[12] Patil, S., Kulkarni, R.A., Patil, S.H., Balaji, N. “Performance improvement in cloud computing through dynamic task scheduling algorithm” In:1st International Conference on Next Generation Computing Technologies, pp. 96–100 (2015)

[13] Sidhu, H.S.: Cost–deadline based task scheduling in cloud computing. In: Second International Conference on Advances in Computing and Communication Engineering, pp. 273–279 (2015)

[14] EI-Derini, M. N., Aly, H. H., EI-Barbary, A. E. H. G., & EI-Sayed, L., "DroidCloudlet: Towards cloudlet-based computing using mobile devices". The 5th International Conference on Information and Communication Systems (1CICS), IEEE, pp.1-6, April 2014.

[15] Loke, S. W., Napier, K., Alali, A, Fernando, N., &Rahayu, W., "Mobile Computations with Surrounding Devices: Proximity Sensing and Multi-Layered Work Stealing," ACM Transactions on Embedded Computing Systems (TECS), 14(2), 22, 2015.

[16] Schildt, S., Busching, F., Jorns, E., & Wolf, L., "CANDIS: Heterogeneous Mobile Cloud Framework and Energy CostAware Scheduling," In Green Computing and Communications (GreenCom), and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, IEEE, pp. 1986-1991, August 2013.