ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse01942020.pdf
https://doi.org/10.30534/ijatcse/2020/01942020

A Cost Effective Scalable Framework for Dynamic Threshold Based Autoscaling in Cloud

C.Venish Raja*[1], Dr.L.Jayasimman[2]
[1]Research Scholar, Department of Computer Science, Bishop Heber College(Autonomous), Affiliated to Bharathidasan University, Trichy,Tamilnadu,India.
[2]Assistant Professor, Department of Computer Science, Bishop Heber College(Autonomous), Affiliated to Bharathidasan University, Trichy,Tamilnadu,India.

ABSTRACT

Cloud computing essentially has infinite resources from any single computing applications perspective, and supports those unlimited resources by on-demand scaling. An effective resource allocation strategy for client satisfaction and the maximization of profit for cloud providers is expected within the cloud paradigm.Previous resource allocation strategies are much focused on computation intensive task,distribution of task and not on the type and size of the task.Less focus is on data-intensive tasks in which resource management approaches are not as effective in minimizing costs and lead to over-resource provisioning.In this approach,resource allocation is studied at the events raised in the application,number of task,size of the task and the number of threshold users.A new architecture has been proposed based on task defined and service oriented resource allocation.The proposed architectural framework uses the dynamic threshold based auto scaling mechanism in allocating the resources. It also aims to find the maximum threshold values for the task related metrics with minimum resources. In addition the proposed work mainly focuses on the reduction of cost and provide better efficiency with minimum resources used. The resource allocation is based on the events raised by the cloud users while using the cloud services. The experimental results indicates that the proposed auto scaling approach is better in terms of cost and obtain maximum throughput with minimum resources utilized.

Key words: Resource allocation; cost;Auto scaling; server virtualization; Event

1. INTRODUCTION

Cloud computing provides users with scalable, on-demand, and virtualized resources. Users can use a shared computing resource pool provided with minimal management efforts. The evolution of cloud computing has its origin in cluster, grid and utility computing.A series of service models are provided for clouds, namely, software as a service(SaaS), platform as a service(PaaS) and infrastructure as a service(IaaS). In addition , cloud architecture models for clouds can be seen as private, collective, public and hybrid.

The workload in the cloud environment is highly dynamic.Scalability is the ability of the system to serve applications during varying workload conditions. Elasticity is the degree to which a system is able to adapt to workload changes. Such adaptation is possible through the availability of services and the de-provisioning. The auto-scaling mechanism ensures that the services given are as similar as possible to the current demand at any point in time. Auto-scaling mechanism facilitates the automatic addition or removal of resources for application data.

The static allocation of resources to cloud basedanapplication is not appropriate. If resources are allocated to the application, taking into account the peak workload, it may happen that most of the time resources are underused. This extra provision of resources is called over-provisioning. Over-supplying resources causes a waste of resources. It also increases the usage service charges.

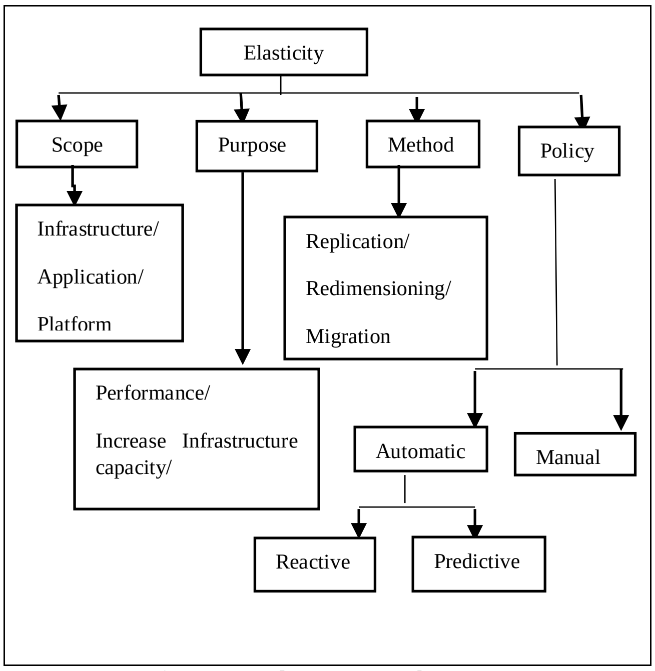

Figure 1 : Elasticity Mechanism

Similarly, if we allocate resources by considering the average application workload, it may happen that, during peak workload. The performance of the application is degraded (SLA infringement). This situation is called under-provisioning.

Figure 1 shows the classification of elasticity mechanism used in cloud computing. Scaling in the cloud can be done in two ways. They are vertical and horizontal scaling.Vertical scaling resizes the VM by changing its resources during scaling operations atruntime. One of the biggest problems regarding vertical scaling is that many service providers do not support reconfiguring the running VM. To dynamically perform these operations, modernhypervisors provide mechanisms such as CPU sharing and memory ballooning, to supportthe CPU and memory hot-plug . This approach is not suitable for highly scalableapplications because maximum scaling is possible up to the capacity of the single host.

In Horizontal scaling, VMs are added or removed during scaling operations. Most cloud service providers offer predefined VMs of different sizes at scaling. Horizontal scaling is supported by public service providers like Amazon, Right Scale etc. of resources.

Auto scaling mechanisms facilitates the scaling of application's resources by two differentways: Horizontally or Vertically. The horizontal scaling does the addition or removal ofVM instances as per the changing workload. The vertical scaling reconfigures the VM bychanging its resources as per the requirements .Auto scaling is also similar to dynamic provision of resources.Auto-scaling can be done by proactive or reactive scaling, where proactive is much more cost effective. The proactive approach used predicted demand to periodically allocate resources before they were needed. The reactive strategy led to immediate changes in demand until there was a regular forecast of demand. Both methods were necessary and required for effective management of resources in complex operating environments.Reactive scaling concentrates on network speed, RAM, bandwidth utilization and number of CPU's usage. Proactive scaling is much focused on responsetime, cost and energy consumption.

Threshold based resource allocation can be consider either static or adaptive threshold. In the static threshold whenever the number of task increases performance will be degraded. This can be solved using adaptive threshold. However, virtual machine migration issues will occur in adaptive threshold resource allocation.

The resource allocation models such as demand prediction,dynamic provisioning and time series model results in over provisioning or predefine allocation of resources before the task has been submitted which leads to increase in user's cost

This proposed work is aimed to achieve the following objective, to design a new architectural framework to address the dynamic threshold auto scaling issues in the cloud environment. The proposed framework implements the event capturing procedure at the application level .The paper introduces a cloud service for allocating the resourcesdynamically based on the events raised by the cloud user.

2.RELATED WORK

Efficient allocation of resources to reduce cloud usage costs is the greatest problem in today's world. Many resource management strategies exist to boost performance by reducing costs. This section presents some of the related in cloud resource allocationfield.

G.Park*et al*. [3] presented prediction based resource allocation to minimize the cost of cloud users and in addition to increase the profit for the service providers.But while allocating resources it does not concentrate on the service type, task size and the number of users request per hour which leads to over provisioning of resources. X.Chen*et al*. [5] proposed the self adaptive based resource allocation technique to improve the efficiency of QOS metrics. This approach does not support the dynamic workload.

M.T.Islam*et al*. [6] introduced the deadline based resource allocation model for big data applications in the cloud. This model is used to reduce the cost and minimum provisioning of resources without considering the type of application, size of the task and the number of users.J.R.Naha*et al*. [7] proposed the deadline based dynamic resource allocation technique to minimize the cost,response time and network utilization for the fog computing environment. This approach suits only the dynamic data service model or services available in the cloud.

Kholidy.A [8] proposed the swarm intelligent based prediction approach to reduce the cost and responsetime.This model does not concentrates on the type of task or service, task size and the number of users which results in the wastage of resources .Joseph C *et al*. [9] proposed dynamic interaction- aware resource allocation using micro services in the cloud datacenter. Since it's a new approach the implementation has to be testing using various applications. At present it supports only dynamic data service application.

Gawali*et al*[10] proposed task scheduling and resource allocation procedure to minimize the cost and response time in cloud services. This model is based on demand prediction and assigns resources in advanced manner which may results in increase of cost and wastage of resources.Lu w *et al*. [11] presented cost effective resource provisioning in cloud environment to reduce the cost, bandwidth and response time. Similarly this model also uses the same prediction based approach which leads wastage of resources

SherzerE *at al*.[12] formulates the resource allocation problem in a geographically distributed cloud environment.This approach first study the capacity of servers in various geographical locations and uses mathematical model to reduce the cost and over provisioning of resources. This model does not focus on the type of service and task size and number so it is hard to accept this approach.

Li J *et al*. [13] proposed the resource and replica management strategy for optimizing the total cost and response in edge cloud computing services. This approach focuses on storage virtualization for large files.The only drawback in this model is it has not been fully tested in real time environment for various cloud services.

Khan H*et al*. [14] presented the efficient scheduling technique to enhance the performance and reduce the cost in cloud environment. This model does not concentrate on the type of service. Size of the task and the number users so it results in over provisioning of resources..

A Hierarchical edge-cloud SDN (HECSDN) controller device architecture is proposed by Lin F *et al.* in [15] to increase network scalability and the computation delay on SDN networks under Quality of Service (QoS) requirements. This model concentrates only dynamic data services not video sharing and social networking services.

Chen X*et al.* [16] proposed a prediction based feedback enabled decision making algorithm to reduce the cost and guarantee the QOS requirements in cloud based services. This algorithm does not focus on the varying size of the task and the threshold number of users.Li K *et al.* [17] proposed a multi-agent system for resource allocation to optimize the total cost and enhance the throughput in the cloud services. This approach mainly concentrates on the numeric type of service which means dynamic data services and does not support for other video sharing cloud services.

Chen L *et al.* [18] presented an integer programming to reduce the make span and cost while allocating the resources in cloud. This approach does not consider the size of the task and the number of users which results in over provisioning of resources used.Various task based scheduling techniques are discussed by Mahmoud M et al.[19].Weng C*at el.* [20] proposed a resource-constrained replication strategies for heterogeneous subtask to minimize the cost and response time. This model provides a better solution for dynamic data services where the task size is small and does not when the size of the task increases. Though numerous research have been studied in the field of resource allocation most of them are based on the assumption or prediction and mathematical model. Microsoft azure pricing details are discussed in [21].The amount of minimum bandwidth required to download or viewing video is discussed on [22] and [23].In addition it also discuss about the wastage of resource used The proposed framework uses the detailed study analyzed in [21]-[24] to implement the policy of allocating resources and procedures.In [25] , P.K.Vadla*et al*. proposed a residue based adaptive resource provisioning through multicriteria decision to allocate the resources based on Service Level Agreement(SLA).To enhance cloud security and to yield better performance a novel based approach is proposed by B.Mukhopadhyay*et al.* [26].

3.PROBLEM DEFINITION

.Auto scaling is not a new issue; it is similar to the dynamic provision of resources for cloud services that has been thoroughly studied in the past. The following are the issues in the traditional resource allocation policies or schemes.

- Traditional resource allocation schemes consider only random heterogeneous tasks and, on demand, a predictive model in the allocation of resources that may lead to over-resource provisioning and an increase in overall costs for cloud users.
- The existing resource allocation methods does not consider the parameters such as type of service, task size per request, number of task and the number of users.
- Current resource allocation strategies are studied at the application level.
- Most of the existing resource allocation procedures are experimented using computation intensive task or scientific task rather than data intensive task.
- Some of the resource allocation strategies used are predication based resource allocation, dynamic provisioning of resources, threshold based resource allocation and static based resource allocation.

3.1 Methodology

The proposed framework aims to reduce the cost by allocating the resources based on the events generated by the users. Before the resource allocation process service manager checks the availability of the requested service in the cloud. This architectural framework uses server virtualization to increase the resource allocation and to reduce the burden and complexity of computation from users. In the resource allocation process the framework uses automatic scaling and cloud bursting technique to provision the resources based on the events generated. In order to reduce the cost the framework mainly concentrates on the distribution of tasks, CPU usage,bandwidth usage, datacenters and so on. The cloud users using their various devices make their request to the proposed framework called Cost Effective Scalable Framework(CESF).The framework process the each request using the various components to allocate the resource and to reduce the cost and response time of the cloud users. The working procedure of various components of the framework are listed below

a)**Request Manager**:When a user submits a request for service, the Request Handler (RH) process analyzes the requests using request analyzer before deciding whether to accept or deny the request.It also analyzes the request generated by the existing user in terms of events. Request Monitor ensures that there is no overloading of request and calculates the number of request that requires service and the limit of the request queue.

b) **Service Manager:**In order to find the right service provider, the service manager must pick the services required for the customer. Service monitor maintains the list of active services in use.

c)**Event Manager:**It uses event analyzer to analyze the request and sent it to the appropriate resource allocator. Event monitor maintains the list of incoming request.

d)**Resource Manager:**Resource manager allocates the resources for the following events such as searching,trending,scrolling the web page, dynamic data, text information on chat. Resource monitor is use to keep track of available resources and used resources.Once the resources are located it transfers therequest to cache server.

e)**Replication Resource Manager**:Video uploading, watching video, broadcasting, and live TV are some of the events that get resource allocation using the resource replication manager. Replication resource Monitor does the same work as resource monitor.

f)**Response Manager**: It uses response handler to handle the request received from cache and replication cache server. Response handler uses response analyzer to analyze the service response and sent it to the appropriate user. Here the service response refers to the event type or service availability.

4. ARCHITECTURE OVERVIEW

Scalability and cost are the two essential features in the cloud that attracts more providersand customers. But most of the existing cloud providers focus on computation intensive task rather than data intensive task. The description of proposed cost effective scalable framework(CESF) is shown in figure-2.The proposed architectural framework is used to enhance the scalability in heterogeneous cloud environment. The proposed architectural framework provides an interface to the users to interact with different cloud providers using cloud-specific APIs.The proposed framework provides a flexible distribution of data and integrates heterogeneous data with various clouds. The following are the components used to

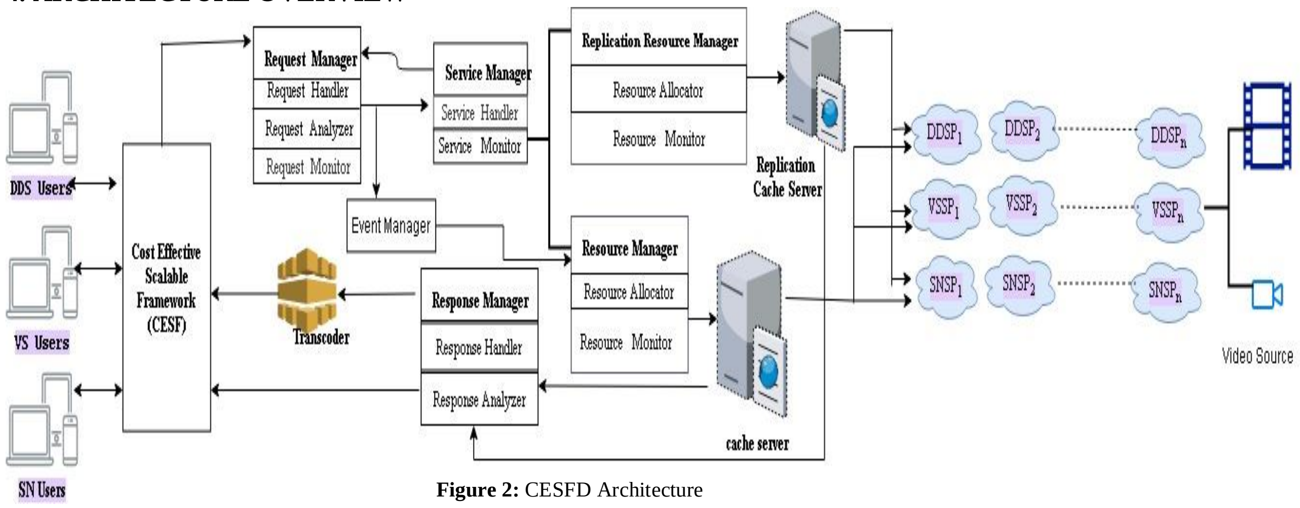

Figure 2: CESFD Architecture

various clouds. The following are the components used to construct the architecture:

User Devices: Cloud users from various devices used to submit their request to the proposed framework. If the service is available the proposed framework provides the cloud specific API to fulfill their needs.

Figure-3 represents the proposed procedure of user requesting for dynamic data services using the proposed framework. The request manager handles the request using the request analyzer and it ensures the service availability.

Cloud Users: The users are classified in to three groups as follows Dynamic Data Service users, video sharing or watching users and social networking users. Banking, Stock Exchange,Healthcare,road traffic analysis and weather predictionare some of the services in which Dynamic data services are used YouTube, Netflix and daily motion are some of the service for video sharing. Face book and twitter for social networking.

Cache Server: This server mainly handles the request from the dynamic data users and it uses minimum resources to fetch the data in order to reduce cost and maximum throughput. In addition it also handles the request from other users based on the events such as searching, trending etc.

Replication Cache Server: This server focuses on data intensive tasks or huge volumes of data by allocating maximum resources in order to reduce the cost, maximum throughput and wastage of resources or over provisioning..Both caches servers act as an edge server which ideal for speeding up dynamic applications and reduce the work load of the cloud datacenters. In rare cases the cache server uses the cloud data centers to fetch data stores locally for future use. In addition it also acts a protection to datacenters by deviating all the request from the various users.

Cloud Servers: The cloud servers or data centers in the proposed architecture remains idle unless any cache server which doesn't have data to sent will communicate with cloud provider to fulfill the user needs. Only cache server can communicate with cloud servers.

Transcoder: It is a component present in the framework used to convert the video files to the device specific format.

Once if the service is available event manager transfers the request to resource manager to allocate the resources.The required data is fetch from the cache server with minimum resources and maximum throughput. In this proposed procedure cache server is the default server for Dynamic data services. So all the request are transferred to the cache server irrespective of the events generated.

Table 1: Mathematical Notation

SU	- Single User
TU	- Number of Users
Treq1,Treq2….Treqn	- Type of Request
Sreq1,Sreq2…..Sreqn	- Size of Request
Evnt1,Evnt2….Evntn	- Events generated by the request
Tser1,Tser2…….Tsern	- Type of service requested by

the SU	
TRC	- Total Resource Cost
$\sum_{i=1}^{n} TU =$	Total number of users
$\sum_{i=1}^{n} Trcs =$	Total number of resources available in cloud servers
$\sum_{s=1}^{n} Evn =$	Total number of Categorized Events
$\sum_{s=1}^{n} Evni =$	Total number of Series in the Events
$\sum_{s=1}^{n} Treq =$	Total number of type of request
$\sum_{s=1}^{n} CSPi =$	Total number of Cloud Service Providers

Mathematical Procedure for Dynamic Data Service (DDS) Resource Allocation to find the cost

1.$\sum_{i=1}^{n} CSPi =$ Total number ofCloud services
2.$\sum_{i=1}^{n} TUi =$ Total number of users
3.$\sum_{i=1}^{n} Totreqi =$ Total number of request
4.$\sum_{i=1}^{n} Tresi =$ Total number of resources
5.Request are categorized to find the type of data associated with cloud service $\sum_{i=1}^{n} Typreqi$
6.Check the validity of the request is made for dynamic data service. $\sum_{i=1}^{n} Vreqi$
7.Find the size of the valid request $\sum_{i=1}^{n} SVreqi$ FIFO(Fisrt in First out) is followed for each valid request.
8.Compare the size of the valid request to the predefined size in the DDS provider
 If (SVreqi<=3)
9.Allocate the minimum resources predefined by the CSP to complete the task
 RA=SVreqi
10.Calculate the cost for each user using the minimum resource cost
$\sum_{i=1}^{n} Totreq(i)$ for each Ui*minimum resource cost
11.Find the response time to complete each request
 RT=↓RT

Pseudo code 1:Dynamic Data Service

```
TRC_DDS ()
{
int $\sum_{s=1}^{n} TU, \sum_{r=1}^{n} Treq$,s=0,tsizemb=3;
For i=1;i<n;i++
{
Treqi=Treqi.Size;
If (Treqi==tsize)
{
Tresi=min_res();
SUi_start_time=current datetime;
Thread.Process.Start(Treqi);
R[s]=Reqi;
}
TRC=$\sum_{s=1}^{n} R[s] * min\_res\_cost * hrsused$ ;
Sui_endtime=Currentdatetime;
}
min_res()
{
intnvms=4,ram=256,bw=0.3;
inttcost=0.05$;
}
avail_res()
{
intnvms=400000,ram=2560000,bw=100;
}
```

Figure-3 illustrates the algorithmic flow of dynamic data service. The algorithm first analyzes the type of cloud service under which the user is requested. The system validates the user request and the quality of the service. The size of each request is measured and compared to the size of the request provider. If the state is less than or equal to the size of the provider, the resources are allocated. The system is used to look for the minimal resources available to complete the task with a faster response time . The average cost is determined by the amount of hours of operation used by the cloud customer.

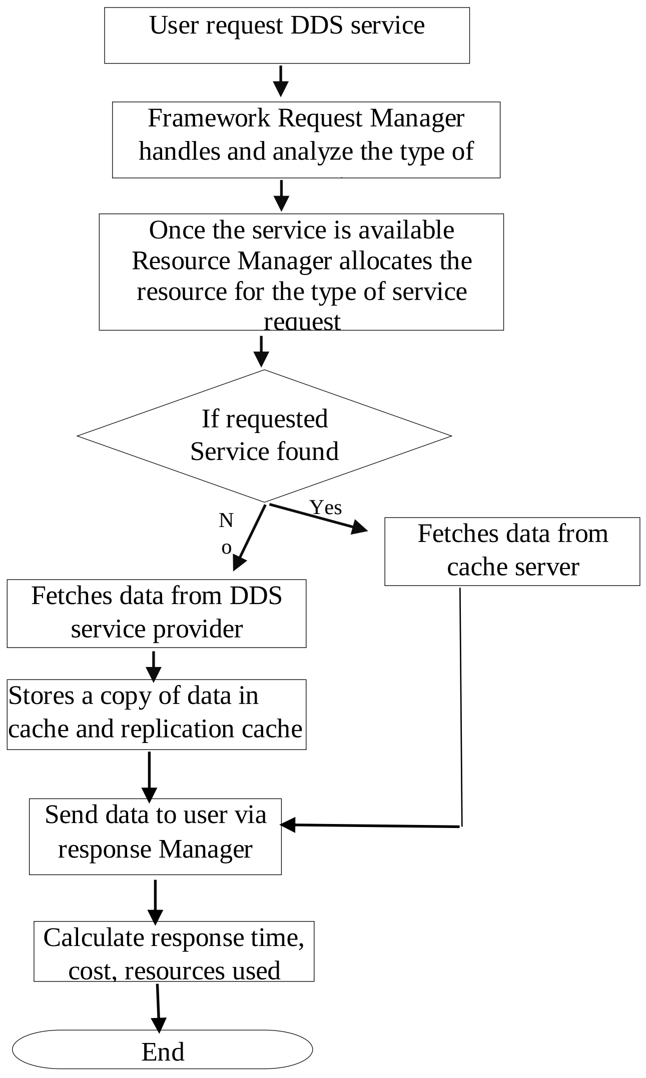
User request DDS service
Framework Request Manager handles and analyze the type of
Once the service is available Resource Manager allocates the resource for the type of service request
If requested Service found
Yes
No
Fetches data from cache server
Fetches data from DDS service provider
Stores a copy of data in cache and replication cache
Send data to user via response Manager
Calculate response time, cost, resources used
End

Figure 3: DDS Cost effective Workflow

In case of service or data unavailability the request is transferred to the cloud datacenter.Once the service is used by the user, it is calculated in terms of hourly computing resources.

Pseudo code 2: Video Sharing Service

```
TRC_VS ()
{
int $\sum_{s=1}^{n} TU, \sum_{r=1}^{n} Treq$,c=0,s=0,tsizemb=3;
For i=1;i<n;i++
{
Treqi=Treqi.Size;
If (Treqi==tsize)
{
Tminresi=min_res();
SUi_start_time=current datetime;
Thread.Process.Start(Treqi);
Emin[s]{SUi]=Evnti;
}
If (Treqi>tsize)
{
Tmaxresi=max_res();
Emax[s]{SUi]=Evnti;
}

TRC=
$\sum_{s=1}^{n} Emin[s]\{SUi]/hour + \sum_{s=1}^{n} Emax[s]\{SUi]/hour$
Sui_endtime=Currentdatetime;
}
min_res()
{
intnvms=4,ram=256MB/vm,bw=0.3mbps;
inttcost=0.5$;
}
max_res()
{
intnvms=4,ram=2Gb/vm,bw=25mbps;
inttcost=0.05$;
}
avail_res()
{
intnvms=400000,ram=2560000,bw=100;
inttcost=0.1$
}
```

Mathematical Procedure for Video Sharing Service (VSS) Resource Allocation to find the cost

1.$\sum_{i=1}^{n} CSPi =$ Total number ofCloud services
2.$\sum_{i=1}^{n} TUi =$ Total number of users
3.$\sum_{i=1}^{n} Totreqi =$ Total number of request
4.$\sum_{i=1}^{n} Tresi =$ Total number of resources
5.Request are categorized to find the type of cloud service $\sum_{i=1}^{n} Typreqi$
6.Check the validity of the request is made for Video sharing service. $\sum_{i=1}^{n} Vreqi$
7.Find the size of the valid request $\sum_{i=1}^{n} SVreqi$ FIFO(Fisrt in First out) is followed for each valid request.
8.Compare the size of the valid request to the predefined size in the VSS provider
If (SVreqi<=3)
9.Find the size of events generated by the each user request
SEvnti=VUreqi
10.Compare the size of the event with the providers predefined event size if (SEvnti>=maxsize)
11.Allocate the maximum resources predefined by the CSP to complete the task
MaxRA=SEvnti
Else
12.Allocate the minimum resources predefined by the CSP to complete the task
MinRA=SEvnti
13.Calculate the total cost for each user by adding maximum and minimum resources used.
TotCost=$\sum_{i=1}^{n} MaxRA(i)$ for each Ui*maximumresource cost +$\sum_{i=1}^{n} MinRA(i)$ for each Ui*minimumresource cost
14.Find the response time to complete each request or event based on their size.
RT=↓RT

Figure-4 shows the procedure of user requesting for video streaming or sharing service. The request manager interprets the request and ensures the availability of the service. Request analyzer handles the type of request and transfers to the event manager. Here the type refers to the new request from the user and also the request from the existing or cloud user. The request from the cloud user is based on the events generated. Event manager uses event analyzer to analyze the requested event and sets the flag equals to 1 for video sharing or watching or broadcasting which means more resources are needed to fulfill the demands and transfers it to the replication resource manager.For other events such as searching, trending scrolling and so on flag is set to zero which less resourcesare required to fulfill the demands and transfer it to the resource manager. Both managers will allocate the resources based on the events raised and transfers to the appropriate servers to fetch the data and send it to the cloud user through response manager. The total cost is calculated by adding the resources used in both servers in hourly basis. The average response time is obtained using the cloud analyst simulator.

The procedure of user request for Social Networking application is shown in figure-5.The request manager identifies the request and ensures the service's availability. Request analyzer handles the type of request and transfers to the event manager. The type here refers to the user's new request and also to the existing or cloud user's request.

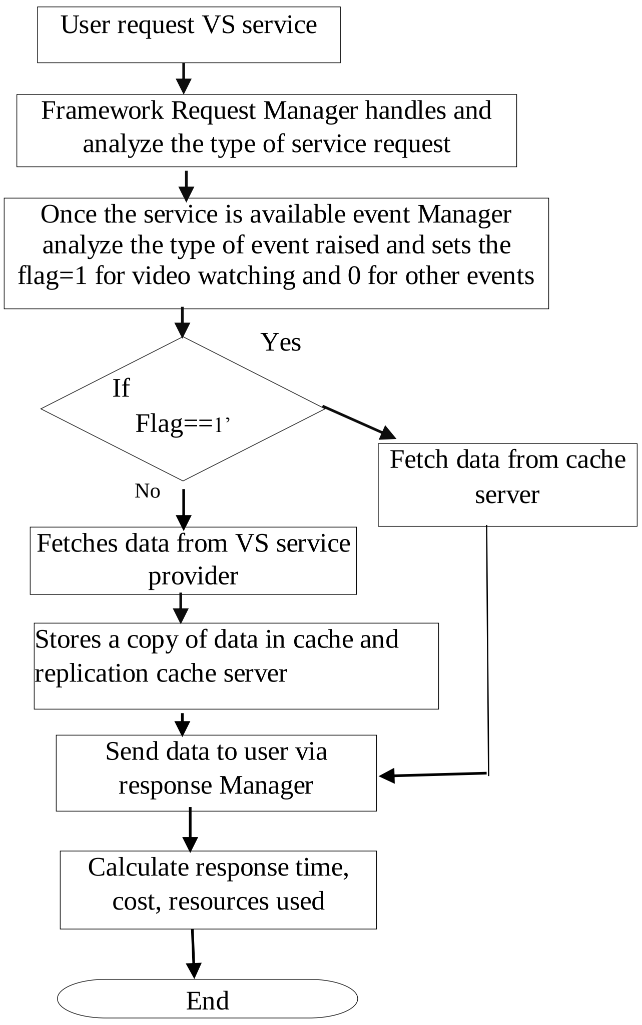
User request VS service
Framework Request Manager handles and analyze the type of service request
Once the service is available event Manager analyze the type of event raised and sets the flag=1 for video watching and 0 for other events
If Flag==1'
Yes
No
Fetch data from cache server
Fetches data from VS service provider
Stores a copy of data in cache and replication cache server
Send data to user via response Manager
Calculate response time, cost, resources used
End

Figure 4: VS Cost effective Workflow Mechanism

Mathematical Framework Procedure for Social Networking Service and other Services Resource Allocation to find the cost.

1.$\sum_{i=1}^{n} CSPi =$ Total number ofCloud services
2.$\sum_{i=1}^{n} TUi =$ Total number of users
3.$\sum_{i=1}^{n} Totreqi =$ Total number of request
4.$\sum_{i=1}^{n} Tresi =$ Total number of resources
Request are categorized to find the type of cloud service $\sum_{i=1}^{n} Typreqi$
5.Framework analyze the type of data
Typreqdi=num/char/media/static/scientific
6.Check the validity of the request is made for social networking or other available service. $\sum_{i=1}^{n} TypeVreqdi$
7.Find the size of the valid request $\sum_{i=1}^{n} SVreqi$ FIFO(Fisrt in First out) is followed for each valid request.
8.Compare the size of the valid request to the predefined size in the VSS provider
If (SVreqi<=3)
9.Find the size of events generated by the each user request
SEvnti=VUreqi

10.Compare the size of the event and type of data associated with the providers predefined event size if (SEvnti>=maxsize&&Typreqdi==’media’ || ‘scientific’)
11.Allocate the maximum resources predefined by the CSP to complete the task
MaxRA=SEvnti
Else
12.Allocate the minimum resources predefined by the CSP to complete the task
MinRA=SEvnti
13.Calculate the total cost for each user by adding maximum and minimum resources used.
TotCost=$\sum_{i=1}^{n} MaxRA(i)$for each Ui*maximumresource cost+$\sum_{i=1}^{n} MinRA(i)$for each Ui*minimumresource cost

14.Find the response time to complete each request or event based on their size.
RT=↓RT

Pseudo code 3: Social Networking Service

```
TRC_SNS ()
{
int $\sum_{s=1}^{n} TU, \sum_{r=1}^{n} Treq$,c=0,s=0,tsizemb=3;
For i=1;i<n;i++
{
Treqi=Treqi.Size;
If (Treqi==tsize)
{
Tminresi=min_res();
SUi_start_time=current datetime;
Thread.Process.Start(Treqi);
Emin[s][SUi]=Evnti;
}
If (Treqi>tsize)
{
Tmaxresi=max_res();
Emax[s][SUi]=Evnti;
}
TRC=
$\sum_{s=1}^{n} Emin[s]\{SUi]/hour + \sum_{s=1}^{n} Emax[s]\{SUi]/hour$
Sui_endtime=Currentdatetime;
}
min_res()
{
intnvms=4,ram=256MB/vm,bw=0.3mbps;
inttcost=0.5$;
}
max_res()
{
intnvms=4,ram=2Gb/vm,bw=25mbps;
inttcost=0.5$;
}
avail_res()
{
intnvms=400000,ram=2560000,bw=100;

}
```

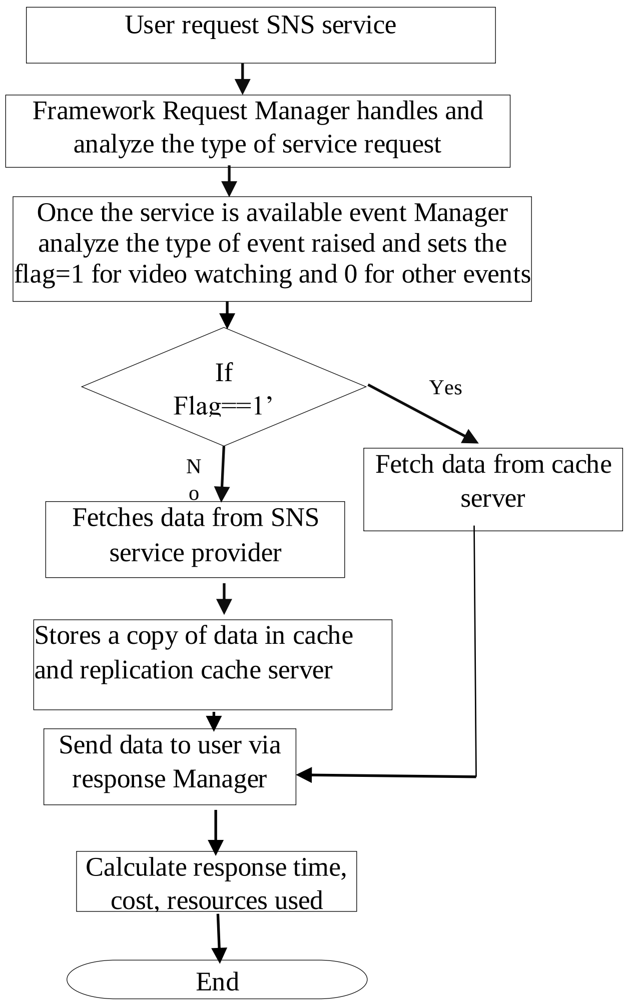
User request SNS service
Framework Request Manager handles and analyze the type of service request
Once the service is available event Manager analyze the type of event raised and sets the flag=1 for video watching and 0 for other events
If Flag==1'
Yes
No
Fetch data from cache server
Fetches data from SNS service provider
Stores a copy of data in cache and replication cache server
Send data to user via response Manager
Calculate response time, cost, resources used
End

Figure 5: SNS Cost Effective procedure

The request from the cloud user is focused on the defined events. The event manager uses the event analyzer to determine the requested event and sets the flag to 1 for video sharing or viewing or broadcasting, which means that more resources are required to satisfy the requests and pass them to the replication resource managerFor other events such as searching, trending, scrolling, adding comments, chatting, and so on flag is set to zero which requires less resources to meet the demands and transfer them to the resource manager. Both managers will allocate resources based on the raised events and transfer them to the appropriate servers to collect the data and send it to the cloud user via response manager. The net cost is determined by hourly adding the resources used on both servers.

5. IMPLEMENTATION

The Enhanced Cost Effective Scalable framework for data intensive service is proposed to provide following benefits:

- Provision for the integration of various cloud service providers using single cloud specific APIs.
- Server virtualization is used to enhance the scalability
- Auto scaling of resources using events triggered by the user
- Construction of a scalable framework by defining task, resources and instances for particular application.
- Predicting the number of resources required to handle the demand pattern for a particular application
- Reduced cost and over provisioning of the resources.
- Periodically classifying future demands and predicting resource requirements
- Better Quality of services in terms of response time.

5.1Experimental Setup:

The proposed methodology is deployed using the private cloud model.No federated system for assessing the technique is available in the present scenario.The proposed cost effective scalable framework (CESF) is implemented using visual studio.net 2013 using .net libraries.The framework components such as request manager, event manager, resource manager has been created using c#.Net functions.Cache servers have been created using Hyper V manager hypervisor.Various free cloud –specific open or common API's and data formare stored in two cache servers. When the cloud user submits the request the proposed framework analyzes the request and confirms the service availability. If the service request is for numeric data or dynamic data minimum resources are allocated using resource manager and fetches data from cache server. Similarly if the request is for video viewing or sharing or uploading more number of resources are allocated using replication cache server which follows the allocation policy discussed in[20][21].Private clouds are operated on windows 7 64 bit machine.The machine uses the Intel core(TM)2 Duo CPU T6500 T6500 running at 2.10 GHz with 4 GB of DDR3 RAM 100Mbps internet connection. To evaluate the scalability of the proposed schemes in large scale systems cloud analyst is used for simulationresults. In order to achieve the maximum throughput in terms of resource provisioning, response time and minimize cost the experiments are carried out by using threshold number of users, task and sizes. Three sets of experiments are carried out using the threshold values of users, task and its sizes to measure the response time and cost.Previous research have not focused too much on the task 's scale, the number of tasks per hour, or the number of users that contribute to an incorrect cost allocation estimate.The following resource configuration is made constant for the Dynamic Data Service. Experiments such as weather details in particular area, traffic details in particular place, stock exchange and banking transaction. Above are few instances where the proposed model is applicable.Data Center used-1, size of task is 3 MB/usr, 4 virtual machines of 256 MB RAM, bandwidth is 0.3Mbps for dynamic data, 1 CPU and default cost as specified [20][21].

Table 2 : Shows the comparisonof scalability features with other existing schemes using 200 tasks and 75 users

Features	TSARAHA (10)	GRPA(11)	Proposed (CESF)
No. of Tasks/hour	200	200	200
No. of users/hr	75	75	75
Avg.Resp.Time	0.37	0.35	0.39sec
RAM/VM	40GB	40GB	256Mb
Task Size per request(Mb)	3MB	3MB	3MB
Bandwidth	100Mbps	100Mbps	0.3Mbps
No. ofVms/user	20	6	4
Cost in dollars	15.75$	14.25$	0.4$

Table 2 shows the comparison of average response time and total cost of the proposed algorithm with other procedures. It is to be noted that average response time is slightly increased when compared with other procedures.This is due to the allocation of resources. At the same time cost is much better when compared to other procedures. The same process is repeated by defining number of task and number of users.

Table 3: Shows the comparison ofscalability features with other existing schemes using 400 tasks and 150 users

Features	TSARAHA (10)	GRPA(11)	Proposed (CESF)
No. of Tasks/hour	400	400	400
No. of users/hr	150	150	150
Avg.Resp.Time	0.37	0.35	0.39sec
RAM/VM	40GB	40GB	256Mb
Task Size per request(Mb)	3MB	3MB	3MB
Bandwidth	100Mbps	100Mbps	0.3Mbps
No. ofVms/user	20	6	4
Cost (in dollars)	15.75$	14.25$	8.45$

The comparison of proposed procedure with 400 tasks is shown in table-2.Here the cost is measured for the total number of users using the resources for one hour.

Table 4: Shows the comparisonof scalability features with other existing schemes using 900 tasks and 250 users

Features	TSARAHA (10)	GRPA(11)	Proposed (CESF)
No. of Tasks/hour	900	900	900
No. of users/hr	250	250	250
Avg.Resp.Time	0.37	0.35	0.39sec
RAM/VM	40GB	40GB	256Mb
Task Size per request(Mb)	3MB	3MB	3MB
Bandwidth	100Mbps	100Mbps	0.3Mbps
No. ofVms/user	20	6	4
Cost (in dollars)	20.75$	19.25$	13.45$

Table 4 shows the threshold value for task and the number of users in the proposed algorithm.

The second set of experimentsis carried out for the media sharing services. The following are the configurations that remain constant in the proposed framework.Data Center used-2, task size as size of video streaming data is 400 MB per request, 5 virtual machines of 512 MB RAM, bandwidth is 100 Mbps, data transfer cost is 0.05$ per hour, cost per VM per hour 0.1$, memory cost 0.05$ per hour and storage cost is 0.1$ for play or watch event, 1 CPU and cost as specified in [2] [4][20][21].

Table 5: Shows the comparisonof scalability features with other existing schemes using 200 tasks

Features	TSARAHA (10)	GRPA(11)	Proposed (CESF)
No. of Tasks/hour	200	200	200
No. of users/hr	75	75	75
Avg.Resp.Time	1.5 sec	1.9 sec	2.1sec
RAM/VM	40GB	40GB	10GB
Task Size per request(Mb)	400MB	400MB	400MB
Bandwidth	100Mbps	100Mbps	25Mbps
No. ofVms/user	20	6	4
Cost (in dollars)	65.75$	65.25$	30.45$

The task size refers to the video file while uploading or downloading.The cumulative cost to maximum resources used by similar types of cloud users producing similar events such as replay or displaying video using the proposed system is illustrated in Table 2. The overall cost in table 2 depends on how many users obtain their response in a timely manner. The overall cost is not dependent on a single one-hour request from the customer. For the number of requests determined by the number of users, total cost is calculated based on the cost produced by the simulator.

Total user cost = Total simulator cost / number of users

Total cost per application = Total cost per user / request number.

Table 6: Shows the comparisonof scalability features with other existing schemes using 400 tasks and 150 users

Features	TSARAHA (10)	GRPA(11)	Proposed (CESF)
No. of Tasks/hour	400	400	400
No. of users/hr	150	150	150
Avg.Resp.Time	2.5 sec	2.6 sec	2.5sec
RAM/VM	40GB	40GB	10GB
Task Size per request(Mb)	400MB	400MB	400MB
Bandwidth	100Mbps	100Mbps	25Mbps
No. ofVms/user	20	6	4
Cost (in dollars)	100.75$	110.25$	50.45$

Table 6 provides a comparison of our proposed algorithm With other algorithms using 150users.

Table-7 provides a comparison of our proposed algorithm with other algorithms using 250 users and 400 tasks

Features	**TSARAHA (10)**	**GRPA(11)**	**Proposed (CESF)**
No. of Tasks/hour	900	900	900
No. of users/hr	250	250	250
Avg.Resp.Time	2.5 sec	2.6 sec	2.5sec
RAM/VM	40GB	40GB	10GB
Task Size per request(Mb)	400MB	400MB	400MB
Bandwidth	100Mbps	100Mbps	25Mbps
No. ofVms/user	20	6	4
Cost (in dollars)	221.75$	240.25$	80.45$

Table 8: Shows the comparison of scalability features with other existing schemes using 300 tasks

Features	**TSARAHA (10)**	**GRPA(11)**	**Proposed (CESF)**
No. of Tasks/hour	200	200	200
No. of users/hr	100	100	100
Avg.Resp.Time	2.5 sec	2.6 sec	2.5sec
RAM/VM	40GB	40GB	10GB
Task Size per request(Mb)	200MB	200MB	400MB
Bandwidth	100Mbps	100Mbps	25Mbps
No. of Vms/user	20	6	4
Cost (in dollars)	70.75$	75.25$	40.45$

Table 9: Shows the comparison of scalability features with other existing schemes using 300 tasks

Features	**TSARAHA (10)**	**GRPA(11)**	**Proposed (CESF)**
No. of Tasks/hour	300	300	300
No. of users/hr	175	175	175
Avg.Resp.Time	2.5 sec	2.6 sec	2.5sec
RAM/VM	40GB	40GB	10GB
Task Size per request(Mb)	200MB	200MB	200MB
Bandwidth	100Mbps	100Mbps	25Mbps
No. of Vms/user	20	6	4
Cost (in dollars)	90.75$	95.25$	50.45$

Table-7 provides a comparison of our proposed algorithm with other algorithms. It is also threshold value for task size and the number of users.

The third set of experiments is carried out for the Social Networking services. It uses both configurations as mentioned above in two sets of experiments based on the events triggered by the user. Suppose if the clouds user wants to add comment or texting first configuration from Dynamic Data Services (DDS) is used.

Similarly if the cloud user tries to upload the video file second set of configuration from the video sharing service is used. Then the total cost is calculated by adding two sets of resources used.

Table 8 provides a comparison of our proposed algorithm with other algorithms. It is also threshold value for task size and the number of users.

Table-10 shows the comparisonof scalability features with other existing schemes using 500 tasks

Features	**TSARAHA (10)**	**GRPA(11)**	**Proposed (CESF)**
No. of Tasks/hour	900	900	900
No. of users/hr	250	250	250
Avg.Resp.Time	2.5 sec	2.6 sec	2.5sec
RAM/VM	40GB	40GB	10GB
Task Size per request(Mb)	200MB	200MB	200MB
Bandwidth	100Mbps	100Mbps	25Mbps
No. ofVms/user	20	6	4
Cost (in dollars)	185.75$	195.25$	90.45$

Table-9 represents the comparison of our proposed algorithm ith other algorithms. It is also threshold value for task size and the number of users.

Table-10 represents the comparison of our proposed algorithm with other algorithms. It is also threshold value for task size and the number of users.

Acronyms used in Table 10,11 &12

NOU	Number of Users
NOT	Number of Tasks
RT	Request Type
TCO/HR	Total Cost for all users
TSARAHA	Task Scheduling and Resource Allocation Heuristic Approach Algorithm
GRPA	Global Resource Provisioning Algorithm
CESF	Cost Effective Scalable Framework
TS	Task Size in MB

Table 11: Comparison of scalability features of the proposed model in terms of number of tasks, size of the task, total cost and response time with existing modelsfor Dynamic Data Service from Tables 1,2&3

	Existing Cost Based Scheduling Schemes								**Proposed**			
S.NO	**TSARAHA [10]**				**GRPA[11]**				**CESSMS**			
	NOT	**NOU**	**Avg.RT(S)**	**TCO/HR ($)**	**NOT**	**NOU**	**Avg.RT (S)**	**TCO/HR ($)**	**NOT**	**NOU**	**Avg.RT (S)**	**TCO/HR ($)**
1	200	75	0.37	15.75	200	75	0.35	14.25	200	75	0.39	0.4
2	300	150	0.4	15.75	300	150	0.4	14.25	300	150	0.45	8.15
3	400	175	0.45	16.65	400	175	0.45	17.85	400	175	0.6	8.25
4.	500	200	0.9	17.85	500	200	0.93	18.55	500	200	1.0	9.25
5	600	210	1.0	17.95	600	210	0.99	18.79	600	210	1.5	9.17

Total cost: The amount includes the cost of resources used, storage costs and data transfer from the datacenter to the machine or device of the receiver.The total cost is calculated for total number of request per hour for the total number of users.

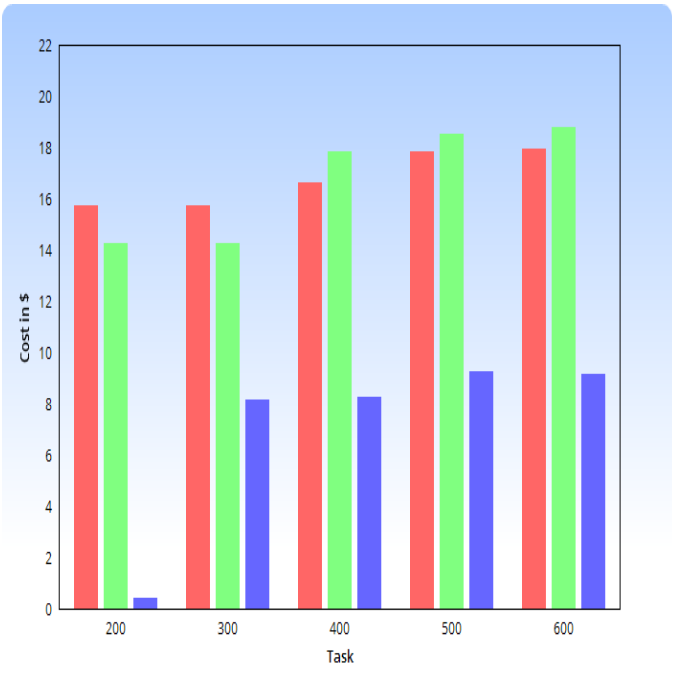

Figure 6: Comparison of CESF using DDS with other models

Figure-6 provides a comparison of existing models with the proposed cost-effective scalable framework(CESF) for Dynamic Data Services. The tasks are performed in the x-axis while the overall cost (price) per hour runs in the y-axis. It is noted that in terms of capital, response times and costs the scheme proposed performs better. That is because the resources are reserved until they are distributed to the current methods. In the proposed model allocation is based on type of data and events triggered.

Average Response Time: It is the time taken to submit the each request to the cache server and forward the response to the users device

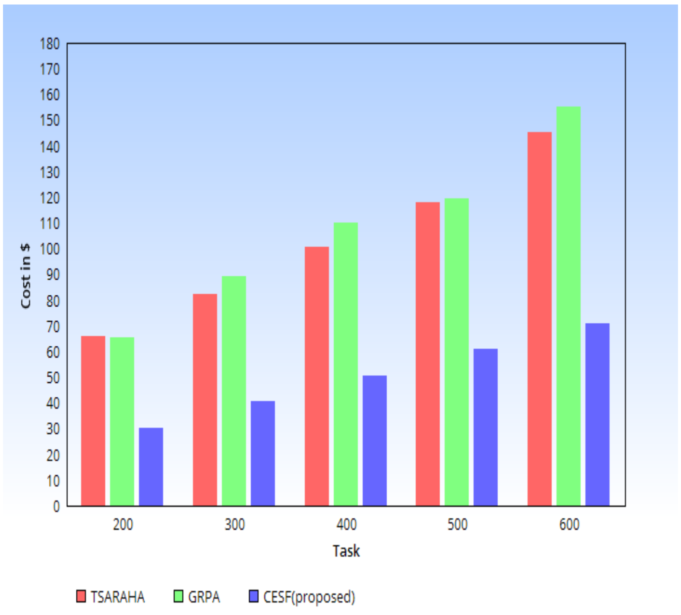

Figure 7: Comparison of CESF using VS with other models

Table 12: Comparison of scalability features of the proposed model in terms of number of tasks, size of the task, total cost and response time with existing models for Video Sharing Service from Table 4, 5 & 6

	Existing Cost Based Scheduling Schemes								**Proposed**			
S.NO	***TSARAHA [10]***				***GRPA[11]***				***CESSMS***			
	NOT	***TS***	***Avg.RT(S)***	***TCO/HR ($)***	***NOT***	***TS***	***Avg.RT (S)***	***TCO/HR ($)***	***NOT***	***TS***	***Avg.RT (S)***	***TCO/HR ($)***
1	200	400	1.5	65.75	200	400	1.9	65.25	200	400	2.1	30.45
2	300	400	2.4	82.55	300	400	2.6	89.25	300	400	2.5	40.45
3	400	400	2.5	100.75	400	400	2.6	110.25	400	400	2.5	50.45
4.	500	400	2.7	117.85	500	400	2.8	119.55	500	400	2.8	60.75
5	600	400	2.8	145.25	600	400	2.8	155.25	600	400	2.8	70.85

Table 13: Comparison of scalability features of the proposed model in terms of number of tasks, size of the task, total cost and response time with existing models for Dynamic Data Service from Table 10,11 & 12

	Existing Cost Based Scheduling Schemes								**Proposed**			
S.NO	***TSARAHA [10]***				***GRPA[11]***				***CESSMS***			
	NOT	***TS***	***Avg.RT(S)***	***TCO/HR ($)***	***NOT***	***TS***	***Avg.RT (S)***	***TCO/HR ($)***	***NOT***	***TS***	***Avg.RT (S)***	***TCO/HR ($)***
1	200	200	2.5	70.75	200	200	2.6	75.25	200	200	2.7	40.45
2	300	200	2.6	80.75	300	200	2.6	82.25	300	200	2.7	42.75
3	400	200	2.6	90.75	400	200	2.6	95.25	400	200	2.7	50.45
4.	500	200	2.6	100.25	500	200	2.6	102.25	500	200	2.7	59.75
5	600	200	2.7	127.25	600	200	2.7	132.25	600	200	2.8	69.75

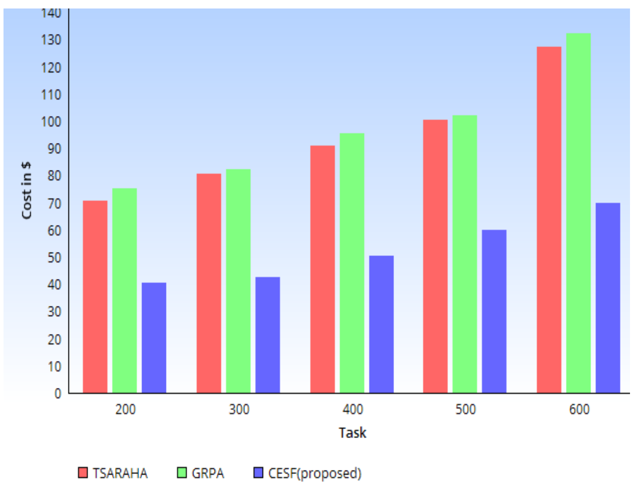

Figure 8 : Comparison of CESF using VS with other models

Figure 7 provides a comparison of existing models with the proposed cost-effective scalable framework(CESF) for Video Sharing Services. It is noted that in terms of capital, response times and costs the scheme proposed performs better.Figure 8 provides a comparison of existing models with the proposed cost-effective scalable framework(CESF) for Social Networking Services. It is noted that in terms of capital, response times and costs the scheme proposed performs better.

6. FUTURE ENHANCEMENT

Since users rent resources for their purpose from remote servers, they have no control over the resources. There is a migration issue if users want to switch to another provider to ensure their data are better stored. Transferring large data from one provider to the other is not easy. In addition, the detailed study of energy consumed in this approach can be extended as future work.Mobility, data deduplication and unstable bandwidth challenges have not been discussed in this approach.

8.CONCLUSION

In this research, a cost-effective scalable framework is proposed based on a dynamic threshold based on an auto scaling resource allocation policy. The proposed framework focuses on events raised by cloud users while using the services, the size of the task per request, the number of tasks and the threshold number of users, along with other parameters such as CPU utilization, bandwidth, cost and average response time. Experimental results show that the proposed framework provides the best solution for data intensive and computing-intensive resource allocation in cloud services. In addition, the proposed framework also reduces costs and response time for cloud users.

REFERENCES

[1] C.Venish Raja, Dr.L.Jayasimman**A Cost Effective Scalable Scheme for Dynamic Data Service in Heterogeneous Cloud Environment**, *International Journal of Advanced Science and Technology*, Vol.28. No.20, 2019, pp.764-776.

[2] MS.C.Kamatchi, R.Pooja, S.Serishma, R.Vanitha**Data Deduplication Security with Dynamic Ownership Management,** *International Journal of Computer Science Trends and Technology*, Volume 5, Issue 2, Mar-Apr 2017, pp.252-256.

[3] G. Park and M. Song, **Prediction-based Resource Allocation using LSTM and Minimum Cost and Maximum Flow Algorithm,***in proc. International Conference on Process Mining (ICPM)*, Aachen, Germany, 2019, pp. 121-128, doi: 10.1109/ICPM.2019.00027.

[4] C.Venish Raja, Dr.L.Jayasimman**Enhanced Cost Effective Scalable Scheme for Media Streaming in Heterogeneous Cloud**,*Sustainable Humanosphere*,vol.16, no.1, pp.1–12, 2020.

[5] X. Chen, H. Wang, Y. Ma, X. Zheng and L. Guo, **Self-adaptive resource allocation for cloud-based software services based on iterative QoS prediction model,***Future Generation Computer Systems*, vol. 105, pp. 287-296, 2020 https://doi.org/10.1016/j.future.2019.12.005

[6] M. T. Islam, S. Karunasekera and R. Buyya, **dSpark: Deadline-Based Resource Allocation for Big Data Applications in Apache Spark**, *in proc. IEEE 13th International Conference on e-Science (e-Science)*, Auckland, 2017, pp. 89-98. https://doi.org/10.1109/eScience.2017.21

[7] JR. Naha, S. Garg, A. Chan and S. Battula**Deadline-based dynamic resource allocation and provisioning algorithms in Fog-Cloud environment**, *Future Generation Computer Systems*, vol. 104, pp. 131-141, 2020.

[8]H. Kholidy, **An Intelligent Swarm Based Prediction Approach For Predicting Cloud Computing User Resource Needs***Computer Communications*, vol. 151, pp. 133-144, 2020. Available: 10.1016/j.comcom.2019.12.028

[9] C. Joseph and K. Chandrasekaran, **IntMA: Dynamic Interaction-aware resource allocation for containerized microservices in cloud environments**, *Journal of Systems Architecture*, vol. 111, p. 101785, 2020. https://doi.org/10.1016/j.sysarc.2020.101785

[10] Gawali, M.B., Shinde, S.K. **Task scheduling and resource allocation in cloud computing using a heuristic approach**. *Journal of Cloud Computing* 7, 4 (2018). https://doi.org/10.1186/s13677-018-0105-8.

[11] L. Wu, R. Ding, Z. Jia and X. Li, **Cost-Effective Resource Provisioning for Real-Time Workflow in Cloud**, *Complexity*, vol. 2020, pp. 1-15, 2020.

[12] E. Sherzer and H. Levy, **Resource allocation in the cloud with unreliable resources**, *Performance Evaluation*, vol. 137, p. 102069, 2020.

[13]C. Li, J. Bai, Y. Chen and Y. Luo, **Resource and replica management strategy for optimizing financial cost and user experience in edge cloud computing system**, *Information Sciences*, vol. 516, pp. 33-55, 2020. https://doi.org/10.1016/j.ins.2019.12.049

[14]H. Khan et al., **An Efficient Scheduling based cloud computing technique using virtual Machine Resource Allocation for efficient resource utilization of Servers**, *in proc. International Conference on Engineering and Emerging Technologies (ICEET)*, Lahore, Pakistan, 2020, pp. 1-7

[15] F. P. Lin and Z. Tsai, **Hierarchical Edge-Cloud SDN Controller System With Optimal Adaptive Resource Allocation for Load-Balancing**, *in IEEE Systems Journal*, vol. 14, no. 1, pp. 265-276, March 2020.

[16]X. Chen, F. Zhu, Z. Chen, G. Min, X. Zheng and C. Rong, **Resource Allocation for Cloud-Based Software Services Using Prediction-Enabled Feedback Control with Reinforcement Learning,** in *IEEE Transactions on Cloud Computing*,pp 1-1,May 2020.

[17]K. Li, Q. Liu and Z. Zeng, "**Distributed optimisation based on multi-agent system for resource allocation with communication time-delay**," in *IET Control Theory & Applications*, vol. 14, no. 4, pp. 549-557, 5 3 2020. https://doi.org/10.1049/iet-cta.2019.0020

[18]L. Chen, X. Li and R. Ruiz, "**Resource Renting for Periodical Cloud Workflow Applications**," in *IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 130-143, 1 Jan.-Feb. 2020.

[19]H. Mahmoud, M. Thabet, M. H. Khafagy and F. A. Omara, **A Comparative Study of Heterogenous Task-based Scheduling Techniques in a CloudEnvironment**, *in proc. International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)*, Aswan, Egypt, 2020, pp. 1-6.

[20]W. C. Ao and K. Psounis, **Resource-Constrained Replication Strategies for Hierarchical and Heterogeneous Tasks**, in *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 4, pp. 793-804, 1 April 2020.

[21] **Pricing - Windows Virtual Machines | Microsoft Azure,***Azure.microsoft.com*, 2019. *Available:https://azure.microsoft.com/en-in/pricing/details/virtual machines/windows/.*

[22]Manish Ranjan, Abdul Kayum, M.suri, **Broadband definition and standardization of Measurement method in the context of developing countries like India having large rural population***Telecom Engineering Center, Department of Telecommunications, Government of India.* Tec.gov.in, 2019. [Online]. Available:http://tec.gov.in/pdf/Studypaper/Study%20 paper%20on%20BB%20Definition.pdf.

[23] "What is Bandwidth? - Definition and Details", Paessler.com,2019.[Online].Available: *https://www.paessler.com/it-explained/bandwidth*.

[24] N. binti Muhamad Shaari, T. Ang, L. Por and C. Liew, **Dynamic Pricing Scheme for Resource Allocation in Multi-Cloud Environment**, *Malaysian Journal of Computer Science*, vol. 30, no. 1, 2017, pp. 1-17. https://doi.org/10.22452/mjcs.vol30no1.1

[25]P. Vadla, K.B.Prakash**,Residue Based Adaptive Resource Provisioning through Multi-Criteria Decision and Horizontal Scaling,***International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 2, pp. 1610-1622, 2020. https://doi.org/10.30534/ijatcse/2020/108922020

[26]B. Mukhopadhyay,.**A Novel Approach to Load Balancing and Cloud Computing Security using SSL in IaaS Environment**, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 2, pp. 2362-2370, 2020. https://doi.org/10.30534/ijatcse/2020/221922020

[27] How to create video streamingwebiste and type of parameters to be focused https://codetiburon.com /create-video-streaming-website-like-netflix-amazon-hulu/