Optimal Time And Cost Effectiveness In Performance Tuning Of Cloud Load Balancing Approach

R.Justin Kennedy[1], Dr.L. Jayasimman[2]

[1]Research Scholar, PG and Research Department of Computer Science, Bishop Heber College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli, Tamil Nadu.

[2]Assistant Professor, Department of Computer Science, Bishop Heber College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli, Tamil Nadu.

Abstract:
Cloud Computing is a large-scale distributed computing technology, in which a collection of dynamically-scalable and virtualized computing power, storage, and services are delivered to customers on demand over the internet. Load balancing is a process of distributing load. The load is distributed on individual nodes to maximize throughput, and to minimize the response time. Load balancing is a technique which uses multiple nodes and distributes dynamic workload among them so that no single node is overloaded. The main goal of load balancing includes optimal utilization of resources which increases the performance of the system in terms of time and cost. This paper provides the optimal time and cost effectiveness in performance tuning of cloud load balancing approach. In near future this paper will be extended with soft computing based load balancing approach for the higher bandwidth scaling in automated load balancing issues.

Keywords: Resource, Priority, Cloud, Load balancing, Cost.

I. INTRODUCTION

1.1 Cost effectiveness:

Cost-effectiveness analysis is a form of economic analysis that compares the relative costs and outcomes of different courses of action. Cost-effectiveness analysis is distinct from cost–benefit analysis as in fig-1, which assigns a monetary value to the measure of effect [1, 3]. Cost-effective methods or processes bring the greatest possible advantage or profit when the amount that is spent is considered.The definition of cost effective is something that is a good value, where the benefits and usage are worth at least what is paid for them.

Fig-1: Cost effectiveness Approach

1.2Time effectiveness:

In our rapidly changing, time-conscious world, we are forced to get more done with fewer people in less time. The quantity of time will not change. Time management is the process of planning and controlling how much time to spend on specific activities. Good time management enables an individual to complete more in a shorter period of time. Time management is the process of organizing and planning how to divide the time between different activities [2] as in fig-2.

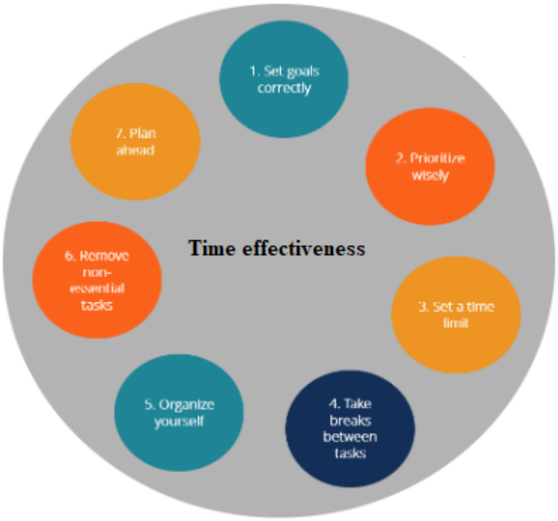


Fig-2: Time effectiveness tips

1.3 Load balancing:

Load balancing as in fig-3 is a technique used to distribute workloads uniformly across servers or other compute resources to optimize network efficiency, reliability, and capacity [4]. Load balancing refers to the process of distributing a set of tasks over a set of resources, with the aim of making their overall processing more efficient. Load balancing can optimize the response time and avoid unevenly overloading some compute nodes while other compute nodes are left idle. Load balancing is a networking solution that distributes traffic across multiple servers to improve application availability and prevent overload.

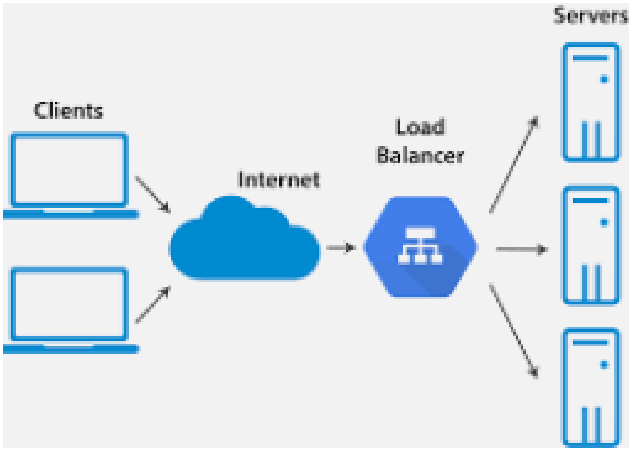


Fig-3: Loadbalancing role in Cloud Computing Model

II.PROPOSED METHODOLOGY

The following fig-4 represents the proposed methodology for the optimal cost and time effectiveness in cloud load balancing approaches.

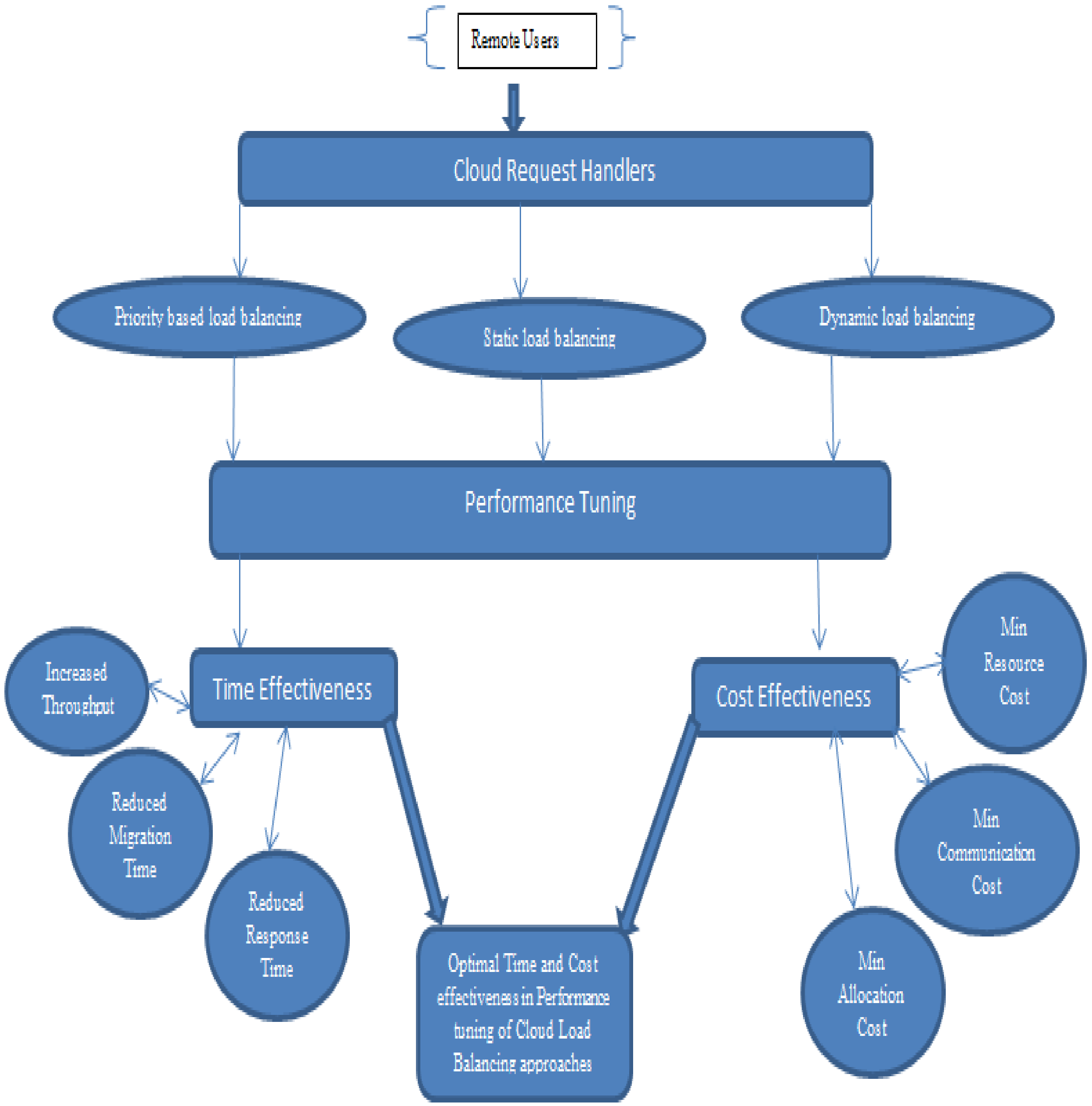


Fig-4: ProposedTime and Cost effectiveness approach for Cloud Load Balancing

The proposed methodology contains 3 sections; they are cloud load balancing types, Timeeffectiveness, and cost effectiveness for the performance tuning.

Proposed methodology Algorithmic approach:

The following algorithm describes the proposed methodology for the Effective time and cost effectiveness in cloud load balancing approaches.

Start

Step-1: Input the Cloud client requests

Step-2: Select the appropriate cloud load balancing approaches based on user requests as given below,

a. Priority based approach
b. Static approach

c. Dynamic approach

Step-3: Execute the Performance Tuning:

a. Time Effectiveness

b. Cost effectiveness

Step-4: Time effectiveness tuning

a. Increased Throughput

b. Reduced Migration Time

c. Reduced Response Time

Step-5:Cost effectiveness tuning

a. Minimize Resource Cost

b. Minimize Communication Cost

c. Minimize Allocation Cost

END

III.IMPLEMENTATION

Consider the sample cloud service as ABC cloud Services as in fig-5,

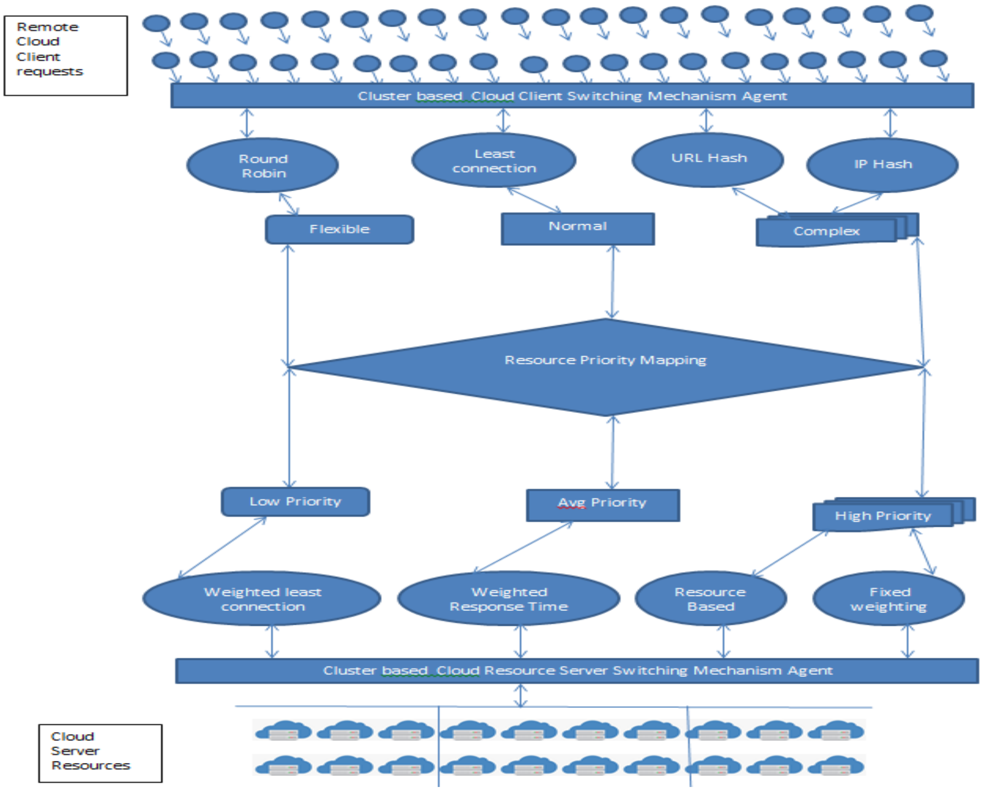

Remote Cloud Client requests
Cluster based Cloud Client Switching Mechanism Agent
Round Robin
Least connection
URL Hash
IP Hash
Flexible
Normal
Complex
Resource Priority Mapping
Low Priority
Avg Priority
High Priority
Weighted least connection
Weighted Response Time
Resource Based
Fixed weighting
Cluster based Cloud Resource Server Switching Mechanism Agent
Cloud Server Resources

Fig-5: Proposed Resource prioritization approach for Cloud Load Balancing

a. Priority based Cloud load balancing approach

The proposed methodology represents the effective resource prioritization in optimal cloud load balancing approach using cluster switching mechanism agent. The proposed methodology contains 3 phases .The initial or topmost phase focusing on to the client side request clustering, the second or bottom phase focusing on to the cloud server side clustering, the final or middle phase focusing on to the mapping of cloud client request to the proper server access in order to attain the optimal effective load balancing.

b.Static load balancing approach

Static load balancing algorithms in distributed systems minimize specific performance functions by associating a known set of tasks with available processors [6]. These types of load balancing strategies typically center on a router that optimizes the performance function and distributes loads. The following figure shows the static cloud load balancing approach as in fig-6[8].

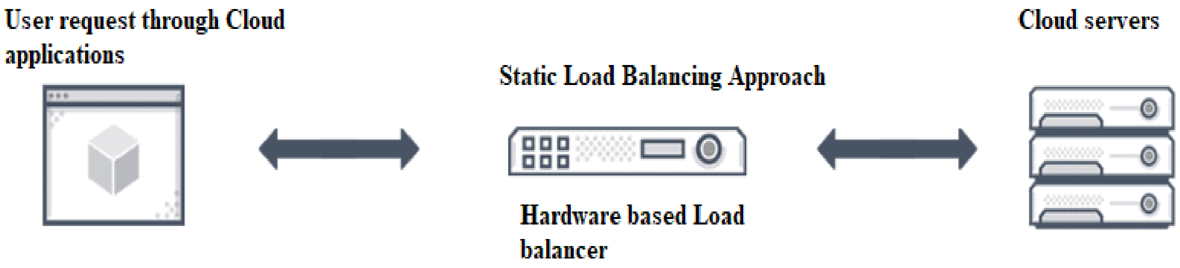

User request through Cloud applications
Static Load Balancing Approach
Hardware based Load balancer
Cloud servers

Fig-6: Static Load balancing Approach

c.Dynamic Load balancing approach

The following fig-7 represents the proposed methodology for optimal cloud access load balancing approach.

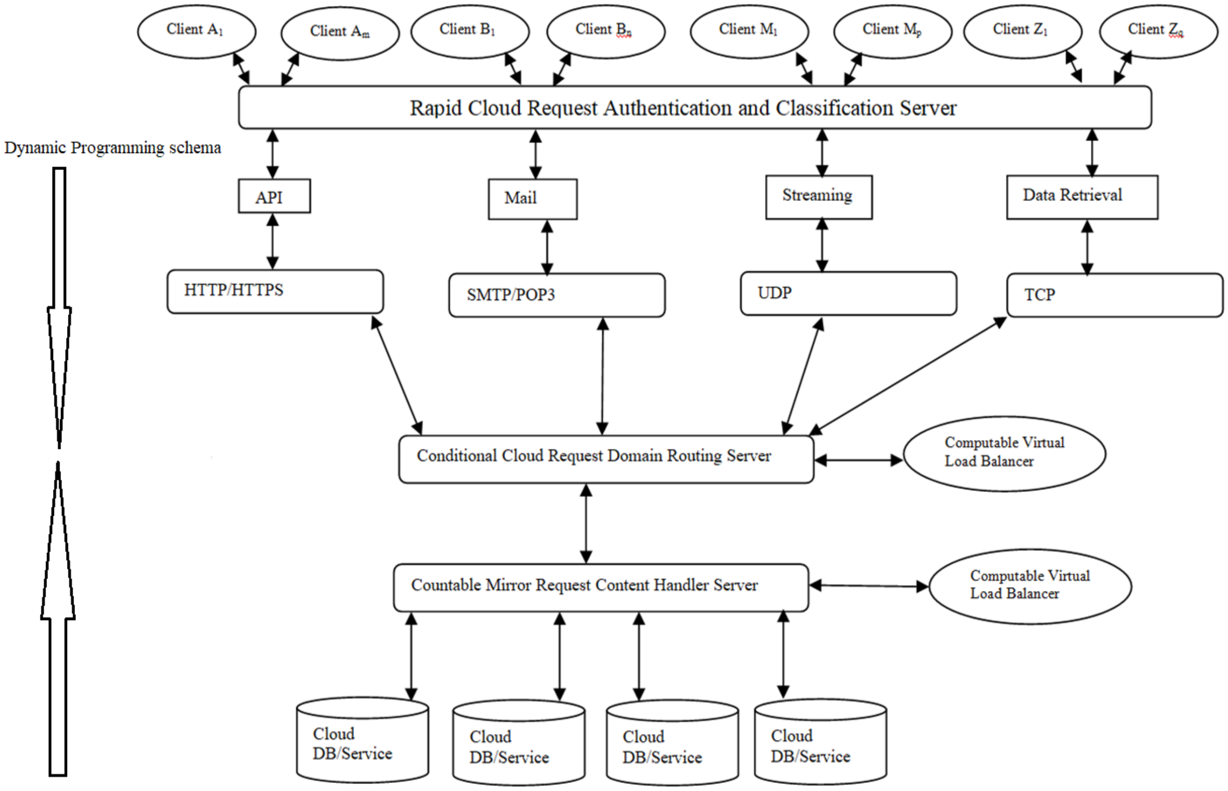

Client A_1
Client A_m
Client B_1
Client B_n
Client M_1
Client M_p
Client Z_1
Client Z_q
Rapid Cloud Request Authentication and Classification Server
Dynamic Programming schema
API
Mail
Streaming
Data Retrieval
HTTP/HTTPS
SMTP/POP3
UDP
TCP
Conditional Cloud Request Domain Routing Server
Computable Virtual Load Balancer
Countable Mirror Request Content Handler Server
Computable Virtual Load Balancer
Cloud DB/Service
Cloud DB/Service
Cloud DB/Service
Cloud DB/Service

Fig-7: Proposed Load Balancing using Dynamic Programming Approach

It includes the unification of 4 different sub servers for effective cloud based content or service accessing. They are

1. Rapid Cloud Request and authentication server.

2. Conditional cloud request domain routing server
3. Computable Virtual Load Balancer.
4. Countable mirror request Content handler server.

1. Rapid Cloud Request and Authentication Server:

The verification and validation of client requests are processed by registered username and passwords along with the associated security algorithms for data security. It processes the client requests through dynamic programming by classifying the request types into applications, mails, video streaming, and data retrievals [8].

2. Conditional Cloud Request Domain Routing Server:

The client request classifications are verified with their protocols, https for applications, SMTP and POP3 for mail transfers, UDP for video streaming and TCP for data retrievals [6].

3. Computable Virtual Load Balancer:

The level of significance for exact classification of client cloud requests are min 5% with inappropriate classification that can be dealt as a meta data with proper mapping for client side accurate data accessing including the Voice over internet protocol[7].

4. Countable Mirror Request Content Handler:

More than one request for the same cloud data or service at the same time will affect the service performance which can be handled with a single high capability server with virtual duplications and multiple physical servers with proper divergence of service towards specific content [5].

d. Time effectiveness tuning

1. Increased Throughput

The steps for increasing the throughput in cloud server network is as follows,

- ❖ Collect cloud server-wide TCP/UDP metrics via /proc/net/snmp and /proc/net/netstat.
- ❖ Aggregate per-connection metrics obtained either from ss -n --extended --info, or from callinggetsockopt (TCP_INFO)/getsockopt(TCP_CC_INFO) inside the cloud server.
- ❖ Tcptrace (1)’es of sampled TCP flows.
- ❖ Channelize Real User Monitoring metrics from the user cloud app/browser.

2. Reduced Migration Time

The steps for reducing the migration time in cloud load balancing is as follows,

1. During the first migration, all the memory pages of the selected cloud server are transmitted from the source node to the target node while the cloud server is still running (first migration transfer).

2. For subsequent migration transfers, the mechanism checks the dirty bitmap to determine which memory pages have been updated during the transfer. Only the newly updated pages are transmitted. The cloud server continues to run on the source node during these transfers.
3. Before transmitting in every transfer, the presence of dirty data is checked in an address-indexed cache of previously transmitted pages. If there is a cache hit, the whole page (including this memory block) is XORed with the previous version, and the differences are Run-Length Encoded (RLE). Only the differences from a previous transmission of the same memory data are transmitted.
4. For the memory data which is not present in the cache, apply a general-purpose quick compression technique.
5. When this mechanism is no longer beneficial, the cloud server is stopped on the source node, the remaining data (left pages, CPU registers and device states, etc.) is transmitted to the target node, and the cloud server is resumed.

Migration time=Data transfer time +Image creation time

3. Reduced Response Time

The least response time load balancing technique takes into account the current number of active connections on each server, plus the average response time. This load balancer forwards the new request to the server that is currently serving the lowest number of active connections and has the shortest average response time.

Response Time ~ Number of active connections + average response time

e. Cost effectiveness tuning

The cost effectiveness mainly depends on the resource, migration, and allocation costs.

1. Minimize Resource Cost

The steps for minimizing the resource cost without any additional storage is as follows,

- ❖ Make sure that when apps scale up to meet demand, they scale back down when demand drops.
- ❖ Implement load balancing to share workloads across resources.
- ❖ Always consider cloud scaling costs in conjunction with other cloud costs to determine where to host additional instances.
- ❖ Plot workflows to minimize the traffic charges that occur when components scale across different platforms -- either from the data center to the cloud, or from one cloud to another.
- ❖ Understand the pricing model of all your cloud providers to avoid accidentally adding in new cost items when you scale.

2. Minimize Communication Cost

When the underlying communication network is an arbitrary graph, we can get an O (log n) approximation by reducing this problem to an instance of a tree metric, by considering a probabilistic embedding of the metric into a distribution over tree metrics. The steps are as follows,

i) Develop a weighted graph G, by combining the query trees for all the queries.
ii) For each edge e = (x, y), decide which data sources and intermediate results move across that edge by solving an instance of the weighted graph using shortest path problem.
iii) Combine the local solutions for all the edges into data transfer states.

3. Minimize Allocation Cost

The minimized resource allocation cost achievement for the cloud requests in load balancing approach includes the following steps,

[1] Computethe number of cloud requests in S.
[2] Compute the number of cloud servers in T.
[3] Compute the number of mirror cloud server clusters in k.
[4] Perform the inter-task communication cost matrix.
[5] Apply assignment problem approach on Cloud service requests.
[6] Store cluster information's.
[7] Modify execution time of tasks in each cluster.
[8] Put inter-communication cost equal zero between the tasks which are on same cluster.
[9] Divide modified n column matrix, which includes execution cost of each work tasking S.
[10] Sort each column matrix.

IV.RESULTS AND DISCUSSION

Consider the sample cloud client requests with a count of 1500 requests from 127 clients and 17 servers for accessing the cloud services.

A.Throughput Efficiency improvement:

Consider the fig-8 a and b for data distribution in the cloud server with 100 data server addresses,80 data units initially stored in the normal random format allocation in (a) as 80 different data clusters requirement but with Metrics aggregation with TCP traces requires only 20 different data clusters as in (b).

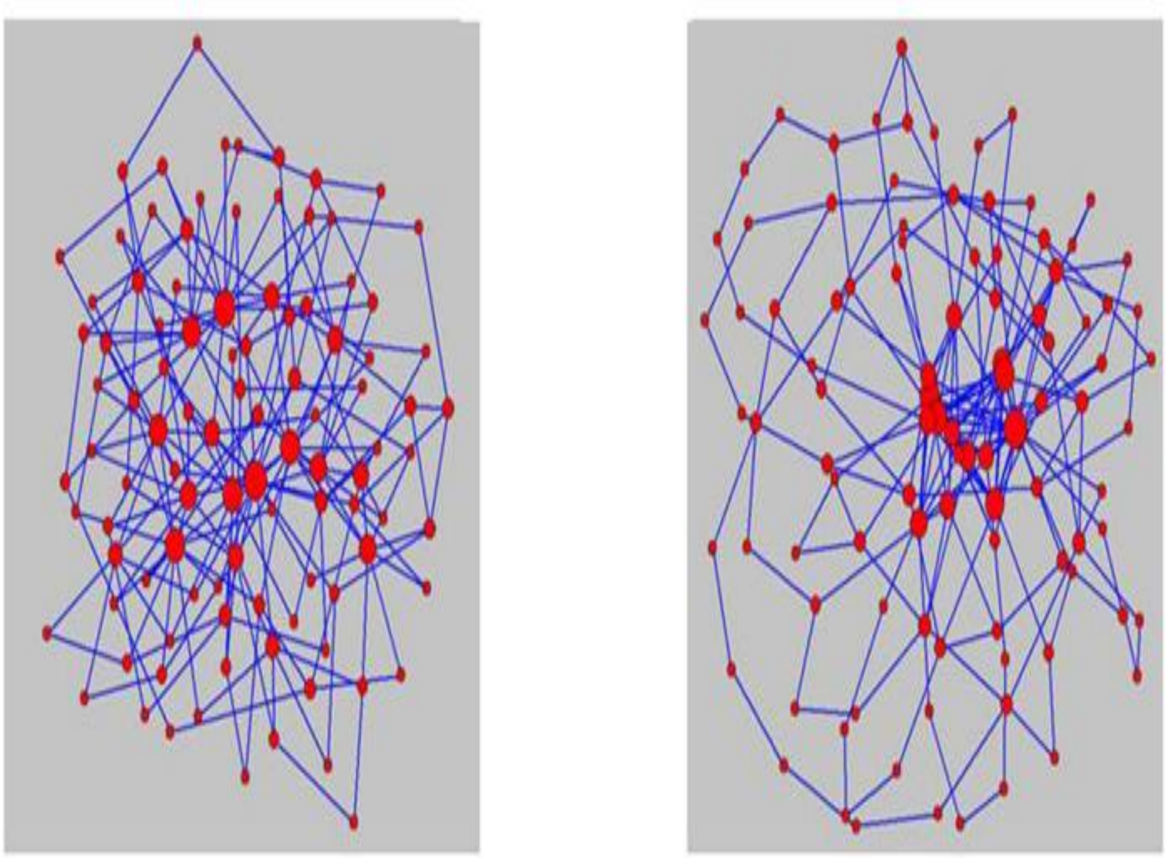


Fig-8 (a) and (b): Throughput efficiency improvement after TCP metrics aggregation

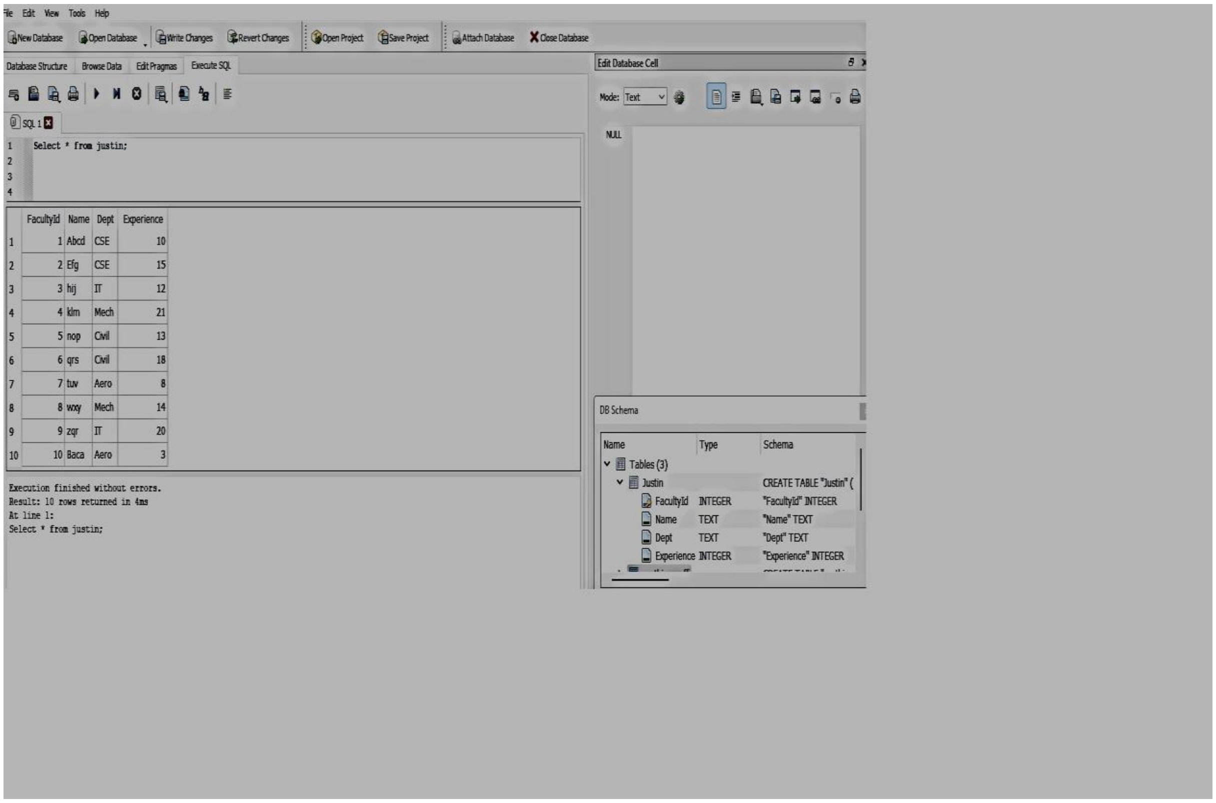
Fig-8-(a)-Normal case
Data elements=N=100
Server Location addresses=100
Data Allocation Cluster=D=80
Throughput Efficiency=N/D=100/80=1.25

Fig-8-(b)-After threshold tuning
Data elements=N=100
Server Location addresses=100
Data Allocation Cluster=D=20
Throughput Efficiency=N/D=100/20=5.0

B. Reduced Migration time Sample Computation Result:

The following 3 figures fig-9, fig-10, fig-11 shows the sample reduced data migration time computation.

The actual data transfer time requires 4 ms for entire table data transmission as in fig-9.



```
Select * from justin;
```

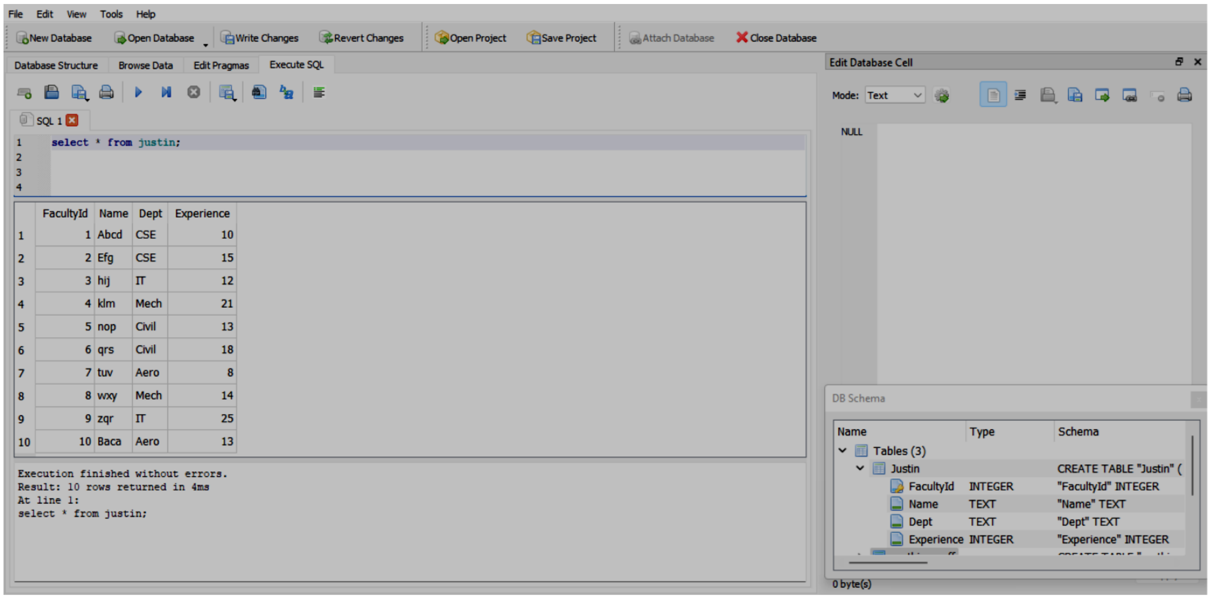
	FacultyId	Name	Dept	Experience
1	1	Abcd	CSE	10
2	2	Efg	CSE	15
3	3	hij	IT	12
4	4	klm	Mech	21
5	5	nop	Civil	13
6	6	qrs	Civil	18
7	7	tuv	Aero	8
8	8	wxy	Mech	14
9	9	zqr	IT	20
10	10	Baca	Aero	3

```
Execution finished without errors.
Result: 10 rows returned in 4ms
At line 1:
Select * from justin;
```

Name	Type	Schema
Tables (3)		
Justin		CREATE TABLE "Justin" (
FacultyId	INTEGER	"FacultyId" INTEGER
Name	TEXT	"Name" TEXT
Dept	TEXT	"Dept" TEXT
Experience	INTEGER	"Experience" INTEGER

Fig-9: Actual results Migration time

After modifying two records in the table the data transfer time requires 4 ms for entire table data transmission as in fig-10.



```
select * from justin;
```

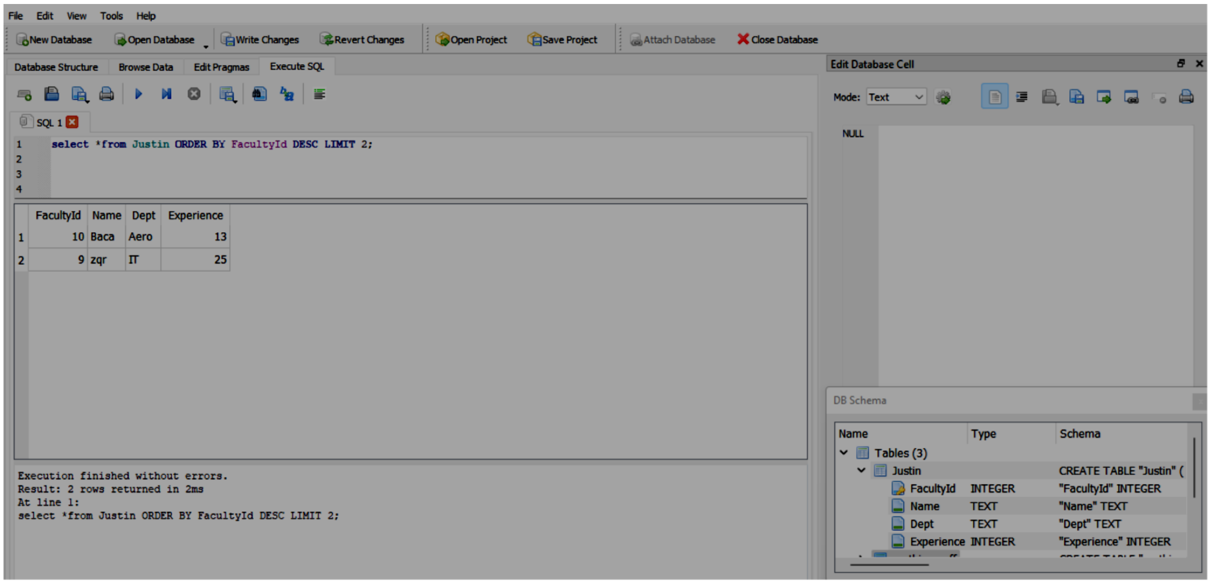
	FacultyId	Name	Dept	Experience
1	1	Abcd	CSE	10
2	2	Efg	CSE	15
3	3	hij	IT	12
4	4	klm	Mech	21
5	5	nop	Civil	13
6	6	qrs	Civil	18
7	7	tuv	Aero	8
8	8	wxy	Mech	14
9	9	zqr	IT	25
10	10	Baca	Aero	13

```
Execution finished without errors.
Result: 10 rows returned in 4ms
At line 1:
select * from justin;
```

Name	Type	Schema
Tables (3)		
Justin		CREATE TABLE "Justin" (
FacultyId	INTEGER	"FacultyId" INTEGER
Name	TEXT	"Name" TEXT
Dept	TEXT	"Dept" TEXT
Experience	INTEGER	"Experience" INTEGER

Fig-10: Modification with full results Migration time

Instead of migrating the entire table (since the table data already exists), only the updated records are migrated to the server component as in fig-11 which requires only 2 ms.



```
select *from Justin ORDER BY FacultyId DESC LIMIT 2;
```

	FacultyId	Name	Dept	Experience
1	10	Baca	Aero	13
2	9	zqr	IT	25

```
Execution finished without errors.
Result: 2 rows returned in 2ms
At line 1:
select *from Justin ORDER BY FacultyId DESC LIMIT 2;
```

Name	Type	Schema
Tables (3)		
Justin		CREATE TABLE "Justin" (
FacultyId	INTEGER	"FacultyId" INTEGER
Name	TEXT	"Name" TEXT
Dept	TEXT	"Dept" TEXT
Experience	INTEGER	"Experience" INTEGER

Fig-11: Updated results Migration time

C.Reduced response time:

Consider the load balancing scenario in the "XYZ" cloud computing environment as in table-1.

Table-1: Sample Cloud server's response time

Server Name	Active connections with Max capacity 25 connections	Average Response time in ms
A	8	100
B	5	150
C	11	75
D	6	20
E	12	50

If the number of requests = 51 new connections. The random allocation of balancing uses the following linear approach as follows,

Allocating the server A with (25-8) = 17 connections=17*100=1700 ms requirement of response time.

Allocating the server B with (25-5) = 20 connections=20*150=3000 ms requirement of response time.

Allocating the server C with (25-11) = 14 connections=14*75=1050 ms requirement of response time.

Total response time=5750 ms for the new requests.

But by the proposed load balancing reduced response time approach initially performs the following sorted order of servers for allocating the requests based on minimal average response time computation. So the order is, D(20),E(50),C(75),B(100) and A(150).

Allocating the server D with (25-6) = 19 connections=19*20=380 ms requirement of response time.
Allocating the server E with (25-12) = 13 connections=13*50=650 ms requirement of response time.
Allocating the server C with (25-11) = 14 connections=14*75=1050 ms requirement of response time.
Total response time=2080 ms for the new requests.

The Time effectiveness according to the proposed methodology implementation, the results are as follows in table-2,

Table-2: Cloud server resources with Time effectiveness

Cloud servers Priority	**Normal Case Performance**	**Proposed methodology for Time effectiveness**
Throughput	1.25 units/Second	5 units/Second
Migration time	8 milliseconds	6 milliseconds
Response time	5750 milliseconds	2080 milliseconds

The following fig-12 shows cloud time effectiveness results.

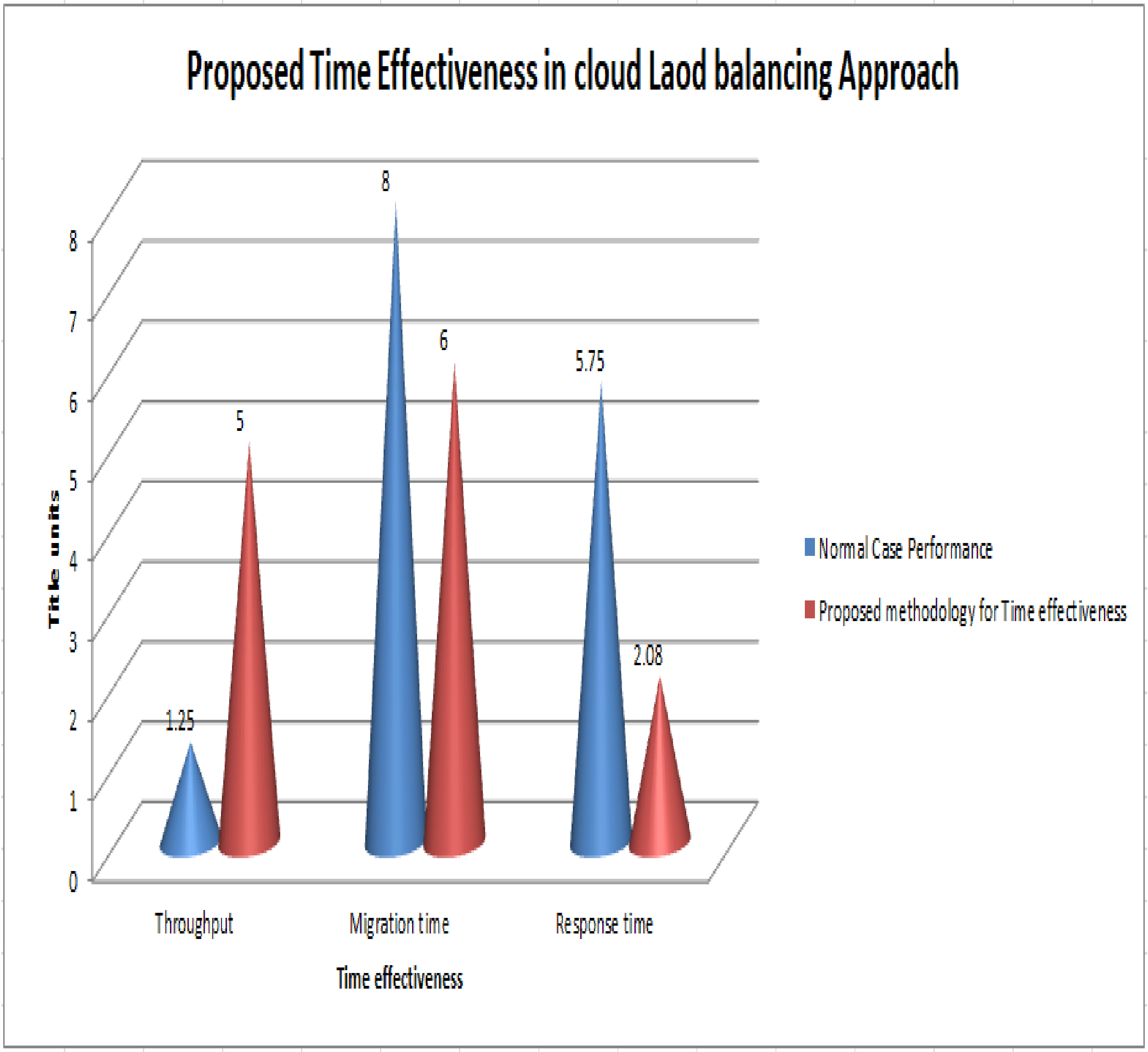
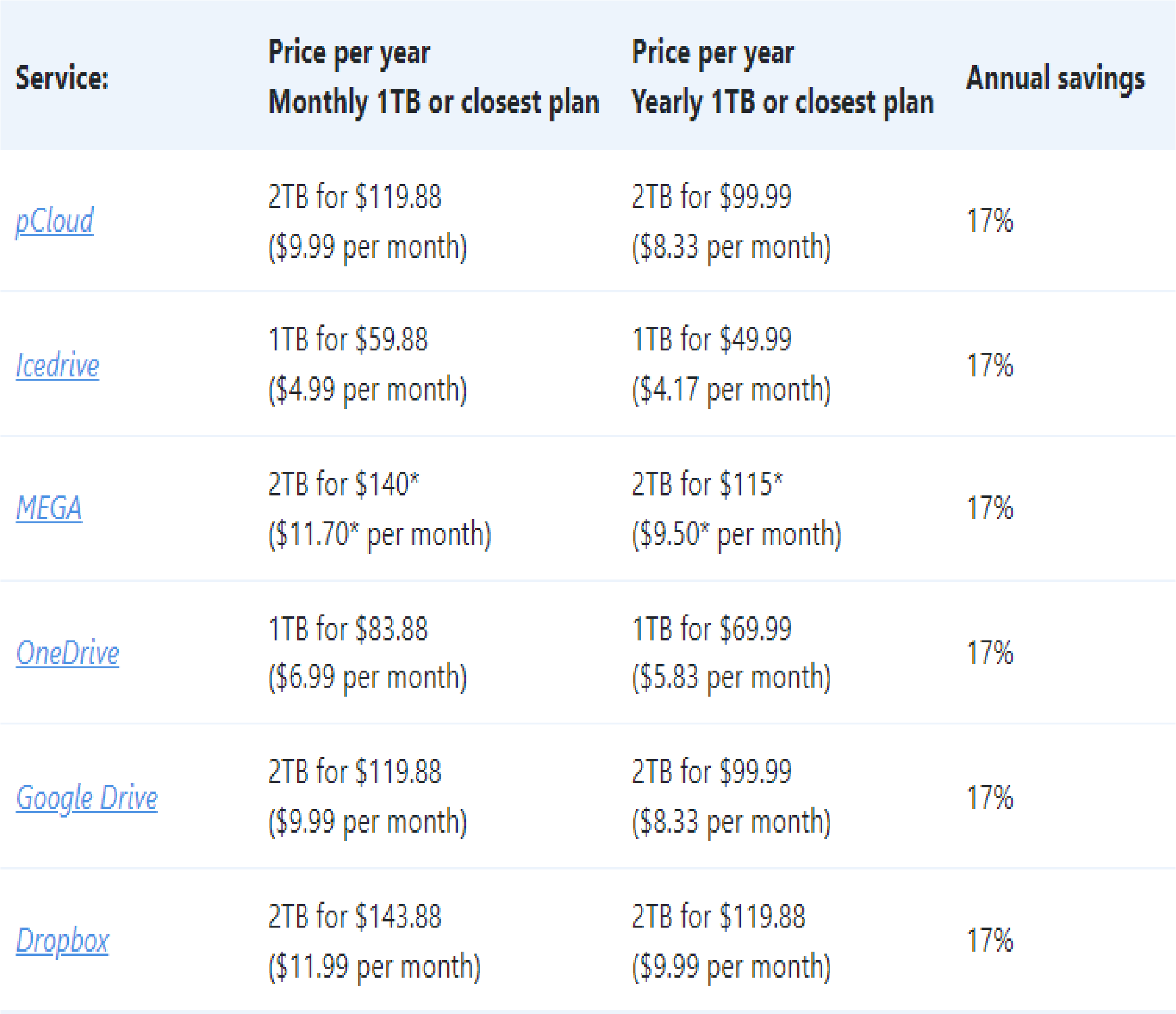


Fig-12: Proposed Methodology Time effectiveness results

D.The resource Cost effectiveness:

The cloud storage costs are referred [12] in the following fig-13



Service:	Price per year Monthly 1TB or closest plan	Price per year Yearly 1TB or closest plan	Annual savings
pCloud	2TB for $119.88 ($9.99 per month)	2TB for $99.99 ($8.33 per month)	17%
Icedrive	1TB for $59.88 ($4.99 per month)	1TB for $49.99 ($4.17 per month)	17%
MEGA	2TB for $140* ($11.70* per month)	2TB for $115* ($9.50* per month)	17%
OneDrive	1TB for $83.88 ($6.99 per month)	1TB for $69.99 ($5.83 per month)	17%
Google Drive	2TB for $119.88 ($9.99 per month)	2TB for $99.99 ($8.33 per month)	17%
Dropbox	2TB for $143.88 ($11.99 per month)	2TB for $119.88 ($9.99 per month)	17%

Fig-13: Cloud server resource cost

The Resource cost computations are in table-3

Table-3: Resource Cost effectiveness

Cloud servers Cost	Normal Case Performance with additional storage of 1 TB+1 TB	Proposed methodology for Resource cost effectiveness without additional storage but by load balancing and app

		scaling with 1 TB alone
Resource Cost	Min Ice drive $49.99*2=$99.98	Min Ice drive $49.99*1=$49.99

E.Reduced communication cost:

Consider the following data transmission network in the “Xyz” cloud server as in fig-14.

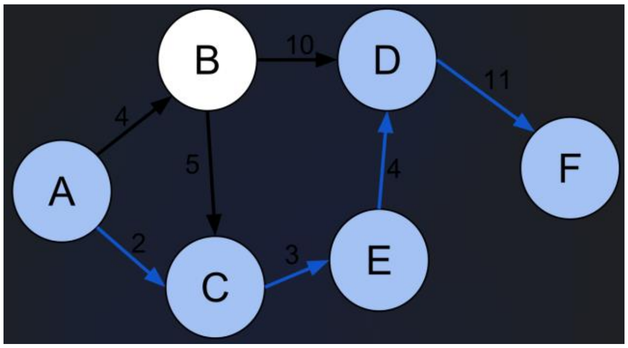


Fig-14: Data transmission network in cloud

For establishing the successful communication the cost computations for A to F are as follows,

The communication cost computations are in table-4

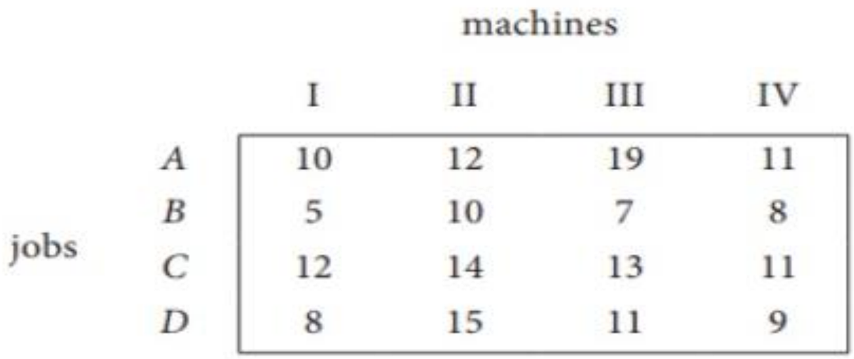
Table-4: Communication Cost effectiveness

Cloud servers Cost	**Normal Case Random paths**	**Proposed methodology for Communication cost with shortest path problem using critical path method**

Communication Cost	A-B-D-F=4+10+11=25 $
	A-B-C-E-D-F=4+5+3+4+11=27 $
	Therefore Min=25 $
		A-C-E-D-F=2+3+4+11=20 $

F.Reduced Allocation Cost:

Consider the sample allocation cost for the cloud server component machines such that the cell values represent cost of assigning job A, B, C and D to the machines I, II, III and IV as in Fig-15.



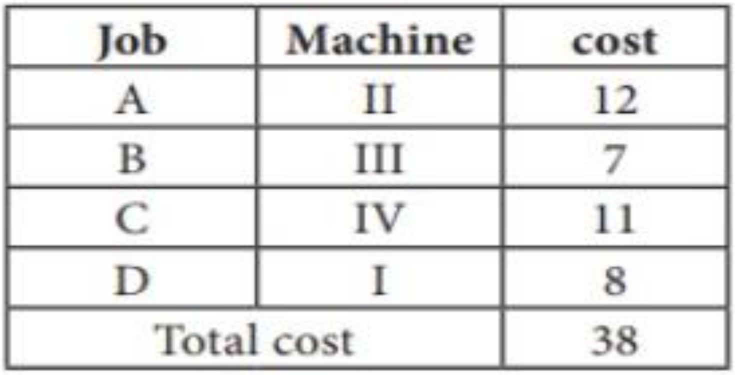
		machines			
		I	II	III	IV
jobs	*A*	10	12	19	11
	B	5	10	7	8
	C	12	14	13	11
	D	8	15	11	9

Fig-15: Task to Cloud server machine allocation request with cost values

The normal case of random assignment is A-I, B-II, C-III and D-IV

Therefore the total allocation cost is=10+10+13+9=$42

By applying the assignment problem approach in the resource allocation method in the cloud server environment, the four assignments have been made. The optimal assignment schedule and total cost is as in fig-16.



Job	Machine	cost
A	II	12
B	III	7
C	IV	11
D	I	8
Total cost		38

Fig-16: Optimal assignment of cloud server tasks

The optimal assignment (minimum) cost= $ 38

The allocation cost computations are in table-5

Table-5: Allocation Cost effectiveness

Cloud servers Cost	Normal Case Performance with random allocation	Proposed methodology for Resource cost effectiveness without additional storage but by load balancing and app scaling with 1 TB alone
Allocation Cost	$ 42	$ 38

The Cost effectiveness according to the proposed methodology implementation, the results are as follows in table-6,

Table-6: Cloud server resources with cost effectiveness

Cloud servers Priority	Normal Case Performance cost units	Proposed methodology for Time effectiveness cost units
Resource Cost	99.98 $	49.99 $
Communication Cost	25 $	20 $
Allocation Cost	42 $	38 $

The following fig-17 shows cloud cost effectiveness results.

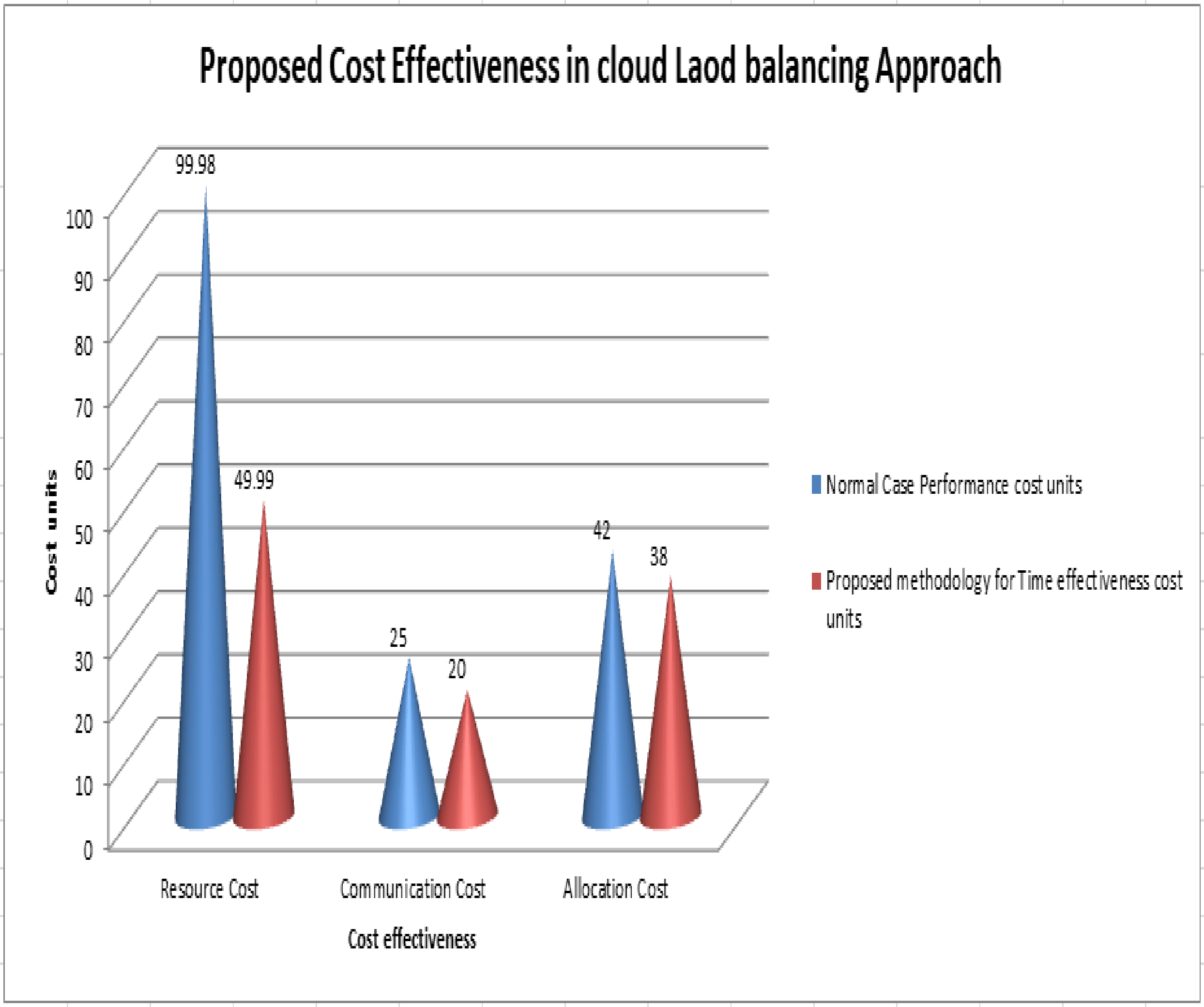


Fig-17: Proposed Methodology Cost effectiveness results

The proposed methodology provides the effective Time and cost for the optimal load balancing which will enforce the cloud service with the maximum level of satisfaction.

V.CONCLUSION

Cloud load balancing is an important task for our current dependency in cloud service networks. The process of cloud load balancing efficiency is computed through its time and cost effectiveness for its performance improvements. This paper performs the Time effectiveness improvement through the increased throughput, reduced migration time and reduced response time along with the cost effectiveness improvements handling through minimizing the resource cost. Communication cost and allocation cost.The proposed methodologyprovides the effective time and cost for the optimal load balancing which will enforce the cloud service with the maximum level of satisfaction. In near future this paper will be extended with the implementation of soft computing approacheswith statisticaltechniquesfor future requirement in load balancing in cloud computing server service for the automated client side accessing.

References

1. Bermes E. Convergence and interoperability: A linked data perspective. In: IFLA World Library and Information Congress. Vol. 77. 2011. pp. 1-12
2. Hidalgo-Delgado Y, Xu B, Marino-Molerio AJ, Febles-Rodriguez JP, Leiva-Mederos AA. A linked data-based semantic interoperability framework for digital libraries. RevistaCubana de CienciasInformáticas. 2019; 13(1):14-30
3. Alakeel AM. A guide to dynamic load balancing in distributed computer systems. International Journal of Computer Science and Information Security. 2010; 10(6):153-160
4. Khan RZ, Ali MF. An efficient diffusion load balancing algorithm in distributed system. International Journal of Information Technology and Computer Science. 2014; 6(8):65-71
5. Khatchadourian S, Consens MP. ExpLOD: Summary-based exploration of interlinking and RDF usage in the linked open data cloud. In: Extended Semantic Web Conference; May 2010. pp. 272-287
6. Schwarte A, Haase P, Hose K, Schenkel R, and Schmidt M. Fedx: Optimization techniques for federated query processing on linked data. In: International Semantic Web Conference. Berlin, Heidelberg: Springer; October 2011. pp. 601-616
7. Kumar B, Richhariya V. Load Balancing of Web Server System Using Service Queue Length. Match Scholar (CSE) Bhopal. Vol. 5(5). 2014. Available from: http://www.ijetae.Com/files/Volume4Issue5/IJETAE_0514_14.pdf
8. Chen C, Bai Y, Chung C, Peng H. Performance measurement, and queuing analysis of web servers with a variation of webpage size. In: Proceedings of the International Conference on Computer Applications and Network Security. 2011. pp. 170-174
9. Zhang Z, Fan W. Web server load balancing: A queuing analysis. European Journal of Operational Research. 2008; 186(2):681-693
10. Singh H, Kumar S. WSQ: Web server queuing algorithm for dynamic load balancing. Wireless Personal Communications. 2015a; 80(1):229-245
11. Singh H, Kumar S. Analysis,& minimization of the effect of delay on load balancing for efficient web server queuing model. International Journal of System Dynamics Applications. 2014; 3(4):1-16
12.https://www.cloudwards.net/understanding-cloud-storage-pricing/