Available online at: http://www.ijmtst.com/vol4issue5.html
International Journal for Modern Trends in Science and Technology
ISSN: 2455-3778 :: Volume: 04, Issue No: 05, May 2018

Effective Technique for Allocating Servers to Support Cloud using GPS and GIS

Kethineni Vinod Kumar[1] | Fayaz Dafedar[2]

[1] Faculty, RGUKT, RK Valley, Andhra Pradesh, India.
[2] Assistant Professor, SNIST, Hyderabad, Telangana, India.

To Cite this Article
Kethineni Vinod Kumar and Fayaz Dafedar, "Effective Technique for Allocating Servers to Support Cloud using GPS and GIS", *International Journal for Modern Trends in Science and Technology*, Vol. 04, Issue 05, May 2018, pp.-13-19.

ABSTRACT

Cloud computing is the on-demand delivery of compute power, database storage, applications, and other IT resources through a cloud services platform via the internet with pay-as-you-go pricing. Cloud is a group of servers. Cloud must be provided services to customers in all circumstances with high quality because it associated with market's applications that cannot be delayed, postponed for another time, etc because that will lead to considerable material losses. Under the could there is group of servers (data centers) may work as distributed system (like Google Apps engine, Amazon, etc) these servers can support each other rapidly and effective cost at peak time, failure, backup, etc to offer continuously services to clients.

A big problem appears when there is not idle server in the same data center, which has source server need the support, in this case need to decide any server from outside the data center site will allocate to that source server (which needed the support) in cloud rapidly with effective cost. This paper proposed a new technique for allocating servers to support cloud, i.e. determine the available servers which relatively have a higher idle (not busy) to support source servers using queue model at the same time employs GIS and GPS techniques via algorithm of Haversine equation to select the idle server which closer to satisfy lowest cost and reach the optimal throughput (performance). Implement and applying this proposed technique shows may not allocated the server which has the highest rate idle but may allocate the server with lowest rate idle to support the source server, and that will not cause a big difference/changing because all those idle servers in cloud are super computers with higher resources, reaching the aim of this new technique means an effective cost with optimal throughput when select the idle server (lowest busy relatively), which is closer to source server.

Keywords—*Cloud Computing; Steady State Queuing Model; Haversine Equation ; GPS and GIS; Data center.*

I. INTRODUCTION

The "Cloud" is a metaphor for the Internet and stripped of complex infrastructure.While "Computing" is a method of mathematical or business computer, providing IT-related capabilities are services, "computer within the Internet services" allow users of access to technology-related services from the Internet (within the cloud) without knowledge of testor control infrastructure that support them. Cloud without needs to provide the application on the user's computer, fewer security risks if what they were assurances the companies that provide these services realistic about the privacy and confidentiality, resources required gear, in addition to some of the fruits of activation of Cloud Computing CC inenterprises, the main factor is cost reduction and the catalystfor moving to the cloud, consequent reduced the need for staffalso equipment and reduce power consumption, in addition to speed construction of the system and the implementation of operations and technical support, meaning they summarize a lot of time [1].Google Apps engine is example of CC that provides common business applications through the Internet and can be accessed and used through a web browser as shown in fig1 while the remaining information is stored on the provider company. That means a computerized model assigns tasks to a group of communications, software and services that are accessible or used through the network. The Computing on this scale is that enables users to access the supercomputer level user influence "power-level", through the use of a small application for the "client" or entry points other "access point", for example mobile phones such as the iPhone or BlackBerry, GoogleAndroid, or even laptops be able to access to the cloud to get the resources which needed it.

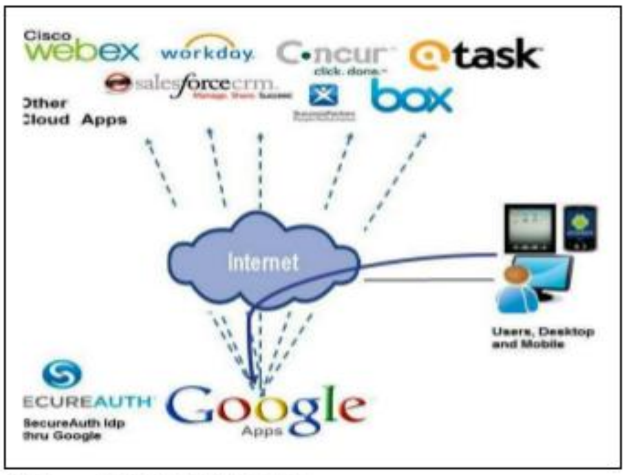


Fig.1. Shows Google Apps script

Infrastructure of CC is more efficient than locally host model, e.g. small and medium-size businesses cannot create the conditions on their own needed to get the most efficiency out of their mail servers.

This is an imperative for a company as large as Google, which powers the email accounts of more than 4 million businesses who Google Apps.

Fig. 2 shows Email application in these two models.

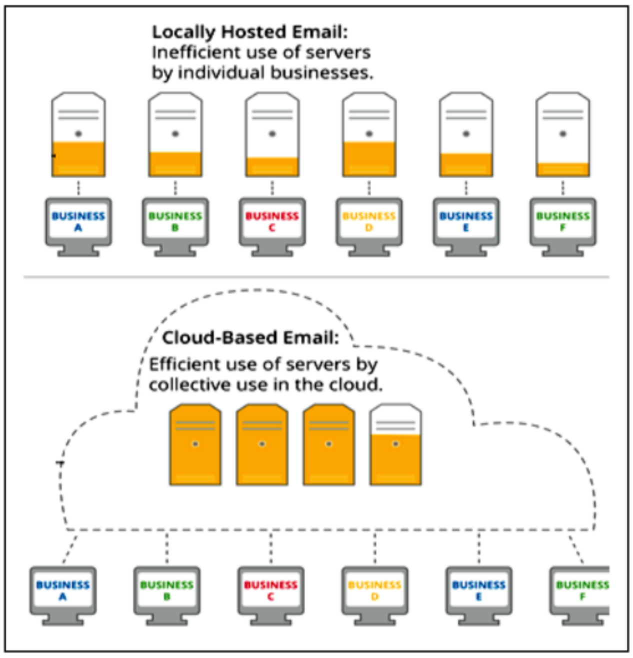


Fig.2. Shows the models of cloud and locally host Email

Currently the cloud consists of reliable services for the delivery of the new generation of data centers based on computers and technology used to stings virtualization. The user can access and use these services from anywhere in the world. And usually does not require the installation of any software to use applications or services [1].

Fig. 3 shows the idea of cloud service provider (as Google Apps engine), simply cloud service provider is customary based on the following group of services which work in
parallel:

1) SaaS: Is aptly referred to as software “on demand.

2) PaaS: it is the platform that hosts applications provided
in software as a service.

3) IaaS: enables businesses to move information from
onsite servers into the cloud.

The cloud must provide services (SaaS, PaaS, or IaaS) to customers in all circumstances and high quality because the cloud associated with such applications of market that can't be delayed, postponed for another time, etc that will lead to losses considerable.

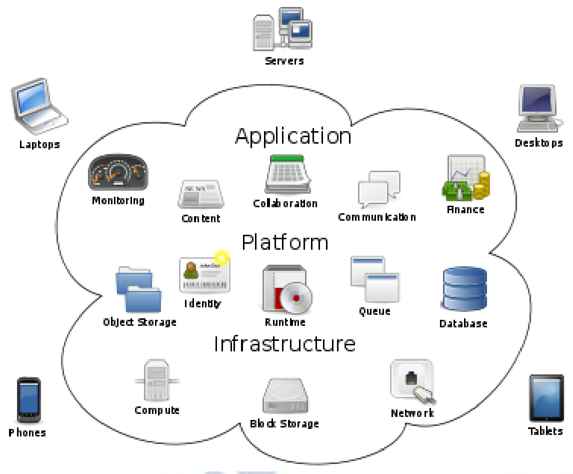


Fig.3.Shows cloud provider

The big problem appears when there is group of servers (data centers) working under cloud as distributed system, which can support each other rapidly with low cost (at peak time, failure, backup, etc) continuity in offering services to clients, the question appears which server from outside the site of data center when there isn't idle server in the same site of data center) will allocate to support the source server (which needs the support) rapidly with effective cost. Solving this problem must be done via technique for allocating servers to support cloud's servers under the cloud datacenter (cloud provider) to cover the services of clients in lowest cost (shortest path/distance) and at the same time allocate the idle servers (are not busy relatively) means the servers which are ready to support the source servers.

II.RELATED WORK

Ayad Ghany Ismaeel and Suha Adham Abdul-Rahman
[2012], proposed a new method of measuring TCP performance at real time of IP network (client-server model) using bio-computing, especially molecular calculation because it provides a wisdom results and it can exploit all facilities of phylogenic analysis. Applying the new method gives very close result of TCP performance comparing with TCP performance which obtains from queuing model as a single server queue (referring to as M/M/1 model) for same server-client model which is constructed to (Bio-computing). The difference percentage of server utility (Busy or traffic industry) and the idle time which are obtained from a new method base on Bio-computing comparing with queuing model was 0.13%[2].

[2]. Michele Mazzucco, Dmytro Dyachuk and Ralph Deters [2011], Said cloud providers, like Amazon, offer their data centers' computational and storage capacities for lease to
paying customers. High electricity consumption, associated with running a data center, not only reflects on its carbon footprint, but also increases the costs of running the data center itself. As a solution allocation policies which are based on the dynamic powering servers on and off are introduced and evaluated. The policies aim at satisfying the conflicting goals of maximizing the users' experience while minimizing the amount of consumed electricity. The results of numerical experiments and simulations are described, showing that the proposed scheme performs well under different traffic conditions. Assuming that jobs enter the system according to an independent Poisson process with rate λ, we model the number of jobs inside the system, for a fixed number of servers n, as the number of jobs in an Erlang loss (or Erlang-B) system with n trunks and traffic intensity $\rho = \lambda / \mu$. Thus, we can treat the resulting system as an M/GI/n/n queuing model (the 'M' stands for Markovian arrivals), which has independent and identically distributed (i.i.d.) service times with a general distribution (the 'GI') and independent of the arrival process, n servers, and no extra waiting spaces (e.g., if all servers are busy, further jobs are lost), augmented with The economic parameters introduced in Thus, can treat the resulting system as an M/GI/n/n queuing model (the 'M' stands for Markovian arrivals) [3]. process with rate λ, while service times – exponentially distributed with rate μ (mean $1/\mu$), $\lambda < \mu$ model the number.
The two techniques above are employing queuing theory with different characteristics, and used for the purposes of allocation server/servers and trying to solve the problem of performance and allocation/customization using theory mentioned, but still there are drawback and weakness caused by a lack of calculating alternative server (idle server) which support the source server, these two techniques are not take in consideration the distance between the location of source server and the alternative server in the cloud provider (data center).

The motivating an overcome the shortages above and to enhance the performance (throughput) for cloud servers that means needed to reach a new technique for allocating servers to support cloud by allocate servers relatively (higher %idle) to support source server using queue theory [2, 4], and at the same time employs GIS with GPS techniques [5] using Haversine equation [6], to reach lowest cost and optimal throughput (performance) addition to satisfy the follow:

1) Services to cover all users cloud.

2) Support for other server within the cloud, when needed
that at peak times (to avoid overflow of queue server and
reject/lost requests of clients).

3) The ability to replace the source server in stage of failure or performs the backup, etc.

4) Reliability and monitoring processes within cloud, that
can be sometimes done by checkup is there same data on thecloud to confirm the validity, integrity, etc because there is distributed system, so the data will put in more than one server and can use this feature to monitor the hackers because they can't reach all copies in multiple sites which contains the data, database, etc at the same time.

III. PROPOSED OF EFFECTIVE TECHNIQUE FOR ALLOCATION SERVERS TO SUPPORT CLOUD

The main tasks of the proposed technique for allocating
servers to support cloud reveals in the flowchart.

This proposed technique needs from the start (at the stage of initialize the data centers of cloud) must be determined the location (Longitude and Latitude) for each data center (group of server) for cloud provider using GPS, and then save these coordinates at table called Location as important table on Allocation database.

As reveals in Fig. 4, must be deciding is the selected idle server from the list of idle servers in the cloud at this time (using queue model) is the nearest to source server using Haversine equation or not, so the proposed technique of
allocation needed two important decisions to allocate specific
server/servers to support the source server in cloud as follow.

IV. EXPERIMENTAL RESULTS

Implement and applying the proposed technique for allocating servers to support cloud referred to in section 3 will need several important software and technologies must be available as follow::

1) Microsoft visual studio 2010 Service pack 1.

2) Microsoft SQL server 2008 R: Used to create the database which needed this technique called Allocation as a relational database contains 3 tables (Location,arrival_service and Idle_server), the structure of these tables shown in Fig. 5; A, Fig. 5; B and Fig. 5; C respectively:

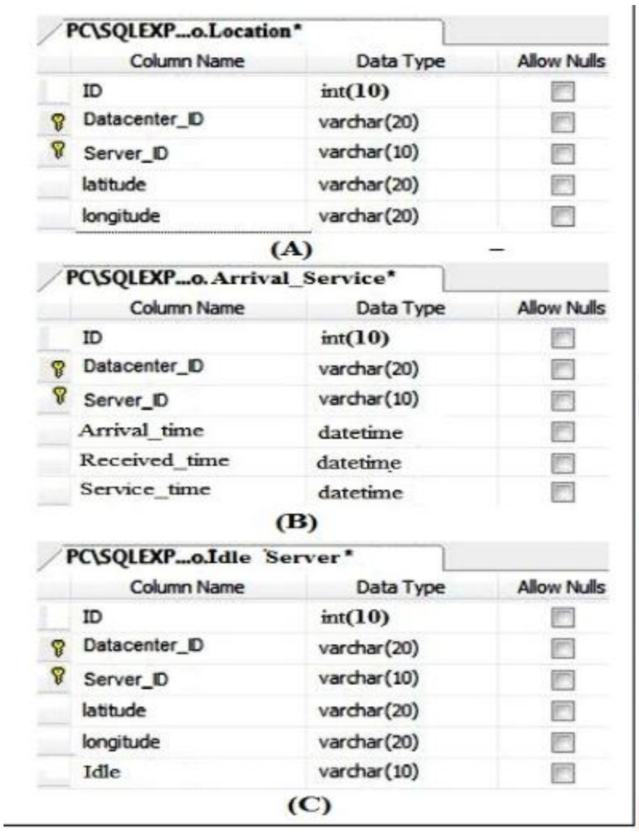
a) C# language.

b) Google Maps API V3.1.

c) Language-Integrated Query (LINQ): the data retrieved from Allocation database at using this LINQ.

Implemented the proposed technique of allocation server/servers to support cloud using C# language at visual
studio 2010 and based on Microsoft SQL server 2008 R, and
for the applying will suppose there are multiple data centers (at data center there is group of servers) distributed in the cloud (e.g. these data centers are Germany-Europe, Italy-Europe,Egypt-Africa, Pakistan-Asia, India-Asia and China-Asia) and previously determined the Latitude and longitude , for each data center with their servers and saved at Location table.

Applying the technique for allocating servers to support the
cloud shown as follow:



PC\SQLEXP...o.Location*

	Column Name	Data Type	Allow Nulls
	ID	int(10)	☐
🔑	Datacenter_ID	varchar(20)	☐
🔑	Server_ID	varchar(10)	☐
	latitude	varchar(20)	☐
	longitude	varchar(20)	☐

(A)

PC\SQLEXP...o.Arrival_Service*

	Column Name	Data Type	Allow Nulls
	ID	int(10)	☐
🔑	Datacenter_ID	varchar(20)	☐
🔑	Server_ID	varchar(10)	☐
	Arrival_time	datetime	☐
	Received_time	datetime	☐
	Service_time	datetime	☐

(B)

PC\SQLEXP...o.Idle Server*

	Column Name	Data Type	Allow Nulls
	ID	int(10)	☐
🔑	Datacenter_ID	varchar(20)	☐
🔑	Server_ID	varchar(10)	☐
	latitude	varchar(20)	☐
	longitude	varchar(20)	☐
	Idle	varchar(10)	☐

(C)

Fig.5. Structure of tables at *Allocation* database

A.Determine the Source server:

Supposing the server A is the source server (which needed support at this time) in cloud data center within India-Asia, e.g.this server reached to peak time (queue reach to overflow, i.e.L=overflow), after determining the Server_ID (A) which need support in the cloud, its Datacenter_ID (India-Asia), and base on these declarations the proposed technique will retrieve the coordinates of that server (which are saved at initialization)FROM LOCATION TABLE (E.G. N22.593726, E78.398438).

B. Find the idle servers in the cloud:

The next step in this technique will determine the servers that have idle rate (idle%) in data centers worldwide cloud, which can support the source server (A), and using the data which recorded for each server in cloud at Arrival_Service table within the database which called Allocation to compute (λ) and (μ) using steady state queuing model as referring to in section 3; A, to find (ρ) traffic intensity (busy), then (π0) (idle% or not busy) to support that source server (A). This computation of queuing model done automatically for each server and the result will put on the Idle-server table at same database as shown in Table 1. that means at any time the server which needed support in the cloud can obtain the idle servers which supported it from that table (Idle_Server) it gives list of idle servers indexed sequentially, e.g. in the case of server (A) within India-Asia data center shows there is 5 idle servers (in Pakistan-Asia, China- Asia, Germany-Europe, Egypt-Africa and Italy-Europe) can support server A (the source server WHICH IS NEEDED SUPPORT NOW) AS SHOWN IN FIG. 6.

TABLE I. LIST OF IDLE SERVERS IN DATA CENTRES WORLDWIDE CLOUD

No	Datacenter_ID	Server_ID	GPSCoordinate Latitude	Longitude	Idle%
1	China-Asia	Server B	N37.1603170	E102.304688	65.9
2	Pakistan-Asia	Server C	N28.613459	E66.445313	60.4
3	Germany - Europe	Server D	N50.064192	E10.195313	55.6
4	Egypt-Africa	Server E	N27:99:44:01	E28:12:50:00	53.5
5	Italy-Europe	Server F	N44.263200	E11.440300	51.3

C. Select the nearest idle server:

This case study shows the suitable selection for idle serverto support source server in cloud which has the higher idle ratiois server (B), means the server (B) in data center of China-Asiaas shown in Table 2. when the idle factor taken in consideration only, but when taken in consideration another factor the distance/location between the source server and the idle server, so will select the server which has idle ratio at the same time must has the nearest/shortest distance (i.e. lowest cost, allocate the nearest idle server plays role to avoid the problems of network more than other servers which have large distance, fast reply, etc). Compute the nearest/distance between the source server and the idle server done using the algorithm of Haversine equation as referring to in subsection 3.

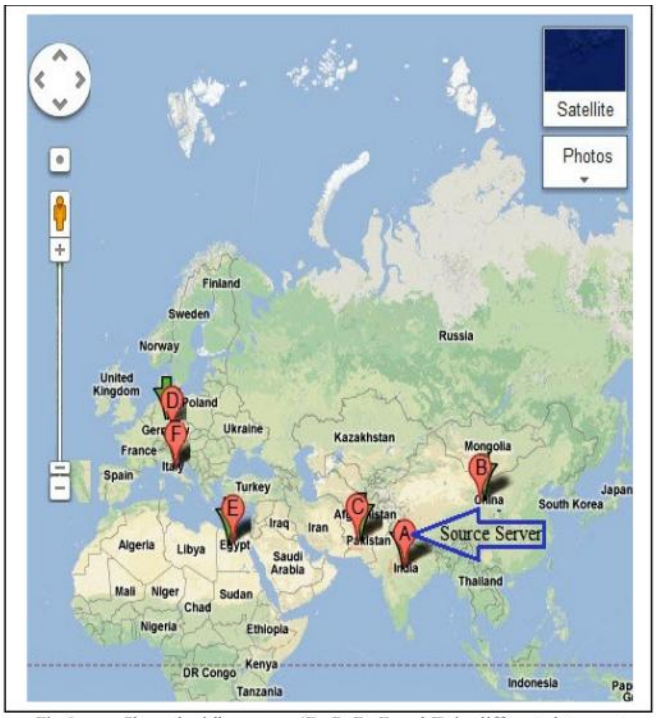


Fig.6. Show the idle servers (B, C, D, E and F) in different data centers of the cloud which can support the source server.

B. The distance between the source server and each idle server in the cloud can computing using e.g. ArcGis 9.2 which need many steps to transform the coordinates (using UTM calculator) and other processing steps (e.g. create file using MS-Excel, SPSS, etc for coordinates which can read by ArcGis 9.2 package) to compute the distances between each idle server and the source server [5], while Google Maps API V3.1 can reach to distance directly (fast way) by using the (Longitude and Latitude) without transformation or any other processing steps so will select Google Maps instead of ArcGis packages. The selection of the idle server for this case study is server (C) at Pakistan-Asia data center in the cloud as shown in Table 2.

2.TABLE II.S HOWS SELECTED OF IDLE SERVER WHICH IS NEAREST TO SOURCE SERVER

No	Source server	Idle server	Idle %	Distance in km
1	Server A	Server B from china data center	65.9	7322
2	Server A	Server C from pakistan data center	60.4	2129
3	Server A	Server D from Germeny data center	55.6	8129
4	Server A	Server E from Egypt data center	53.5	5204
5	Server A	Server F from Italy data center	51.3	8520

D. Discussion of the results

Comparing the important results of the proposed technique for allocating servers to support cloud with other techniques, as shows in Table 3.

V. CONCLUSIONS AND FUTURE WORK

A. Conclusions

The conclusions which obtained from the technique for
allocating servers to support cloud are summarized as follow:

1) The proposed technique for allocating servers to support cloud is more effective cost when it merges between more than one feature (like idle server and which is closer to source server which needed support (using Haversine algorithm).

2) When there is no server in same site of source server can support it This proposed technique is flexible, because i can find the list (more than one option) of servers which are idles outside of data center site (as shown in Table 1) and at the same time select from it the nearest server (means this technique is economic).

3) This proposed technique for allocating servers shows may not be allocated the highest rate idle server (i.e. may allocate the lowest rate idle server) to support the source.

4) server, and that will not cause a big difference/changing because all those idle servers are supper and with higher resource.

5) The proposed technique for allocating servers to the cloud satisfy high response relative to other techniques.

Feature	Proposed Technique of Allocation Servers to Support Cloud	Ayad Ghany Ismaeel and Suha Ad	Michele Mazzucco, and others
Allocation of idle server by using	Queue theory for determining the idle% server and the shortest	Queue theory only	Queue theory only

	distance between the idle server and source server using Haversine equation		
Using GPS and GIS via Google Maps	Yes	No, it don't care to distance between idle sever and source server	No, it don't care to distance between idle sever and source server
Results (selecting the idle% server for supporting the source server)	More an effective because addition to determine the idle server to source server, must be the selected server is nearest (economically means lowest cost and more avoiding the problems of network, satisfy fast reply)	Based on the idle server only	Based on the idle server only
Using Google Maps	Shows more fast in finding the distances between idle server and source server than GIS packages	No, uses	No, uses
Response to support the cloud	Higher relative to other techniques, because it used Haversine equation to compute the nearest idle server	Do not bother to nearest distance between the idle sever and source server	Do not bother to nearest distance between the idle sever and source server

because it take in consideration when allocate the idle server which is closer to the source server, i.e. the throughput and the performance better than other techniques which don't care about the closer server when allocated.

B. Future work

To enhance this proposed technique for allocating servers to support cloud by take in consideration other feature addition to idle server and the closer/nearest server to source server, these feature like size of buffers or capacity, number of processers, etc.
For example the queue model M/M/1/K where K is the size of buffer/capacity [4], this feature is playing role when selected the server to support e.g. video server in the cloud rather than chat server, email server, etc.

REFERENCES

[1] Sun Microsystems, Inc., " Introduction to Cloud Computing architecture", White paper 1 st Edition, June © 2009. Pages 1-35. URL: Web sun.com.

[2] Ayad Ghany Ismaeel and Suha Adham Abdul-Rahman, " NEW METHOD OF MEASURING TCP PERFORMANCE OF IP NETWORK USING BIO-COMPUTING", International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.3, May 2012, Pages 167-183.

[3] Michele Mazzucco, Dmytro Dyachuk and Ralph Deters , "Maximizing Cloud Providers Revenues via Energy Aware Allocation Policies "arXiv:1102.3058v1 [cs.DC] 15 Feb 2011, http://www.eia.doe.gov/.

[4] János Sztrik, "Basic Queueing Theory", University of Debrecen, Faculty of Informatics, publication Debrecen,2012.URL:http://irh.inf.unideb.hu/~jsztrik/education/16/SOR_Main_Angol.pdf