Providing Quality of Service on Services Selection using Anycast Techniques

Lucas Junqueira Adami
University of Sao Paulo
LaSDPC - SSC - ICMC
Sao Carlos, Sao Paulo, Brazil
ljadami@icmc.usp.br

Julio Cezar Estrella
University of Sao Paulo
LaSDPC - SSC - ICMC
Sao Carlos, Sao Paulo, Brazil
ljadami@icmc.usp.br

***Abstract*—On the last years, the complexity and variety of available services on the Internet increased. This fact is leading to the search for efficient techniques of routing client requests to the best server available. A known technique is the application layer anycast (ALA). The main goal of this work is to elaborate efficient ways to provide ALA with quality of service in the context of cloud computing. To achieve this goal, a new algorithm was proposed (GALA, Global Application Layer Anycast). It inherits characteristics from another existing system and uses geolocation as a differential. The results of the experiments showed that the new system, compared to the inherited algorithm, maintains the clients request efficiencies and lowers substantially their latencies.**

***Keywords*-anycast; service selection; quality of service; distributed systems;**

I. Introduction

To answer the growing demand of the clients, the existent systems use servers replication [1]. This way, many hosts that offer the same service are available to the client. Thus, the requests load is shared among these replicas. The problem introduced by this approach is to know how to select the best available server. A solution is to use the anycast technique to route the client requests to the best replica.

Anycast is a technique used to deliver datagrams from one client to a server among many servers, being defined by Partridge et. al [2]. The definition says that an IP (Internet Protocol) address maps a group of servers that offer the same service. In Figure 1, there is an example of a datagram being routed to the IP 10.10.10.10. On the anycast communication, the client sends the packet to a specific address and the network is responsible to route it to the best server.

The original anycast definition describes this technique being used on the IP layer, and it is called IP layer anycast. Later, the application layer anycast concept is introduced [4][5][6] because of many problems of the previous technique. In IP layer anycast, a range of IP addresses must be allocated to anycast addresses. Also, the presence of routers that support the anycast is necessary. Yet, the metrics this technique uses are limited by the routers hops count, while in application layer anycast, this limitation does not exist. For example, the server load information can be used.

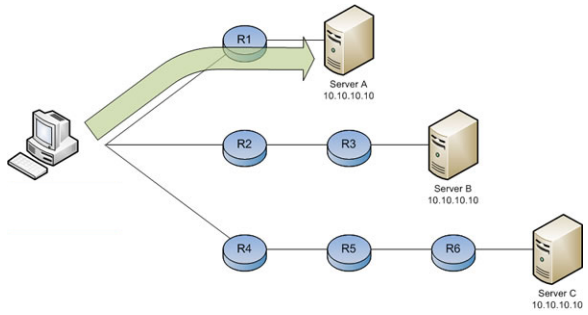

Figure 1. Anycast example [3]

The application layer anycast can be used in systems that utilize network distances (round trip time) to select the best server available. Furthermore, the closest server is selected to process a request. There are many situations where this is advantageous, like in content distribution networks, peer-to-peer, replication web servers architectures and cooperative games [7]. Also, multimedia distribution systems can utilize this type of anycast to balance the load in the content servers. In the work of [8], the authors consider the server load and the network distance as selection metrics. The Figure 2 shows the results of the anycast usage. The x axis represents the experiment over time while the y axis shows the average packet loss during the data transfer. Using anycast, the system achieved almost 0% of packet loss.

II. Related Works

The majority of the anycast systems uses the distance between components to select an appropriated server. OASIS [9] (Overlay-based Anycast Service Infrastructure) is an application layer anycast system where the proximity of the system nodes is calculated using their IP prefixes. Proxima [10] is an infrastructure of anycast based on NCs (network coordinates). In its systems, each node has a coordinate and the distance between two of them is calculated using these values. DOAT [7] (*Distribted Overlay Anycast Table*) is another anycast system. It uses spacial filling curves to find close system nodes.

OASIS, Proxima e DOAT uses predictions to calculate their components distances. But, even though the prediction

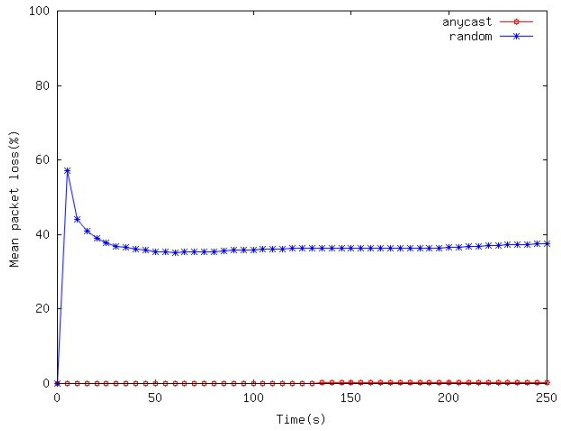

Figure 2. Packet loss in a system with and without anycast [8]

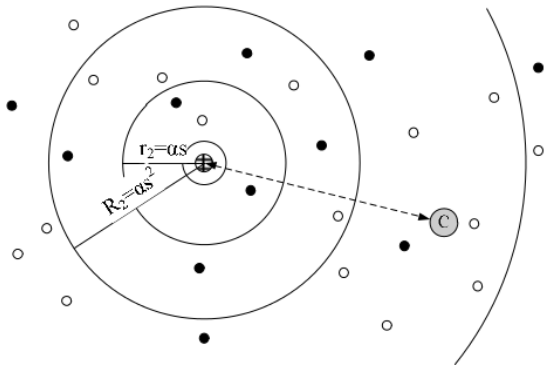

Figure 3. GAA node neighbourhood example [14]

can improve the anycast system performance, it can also make it worse than when the real distances are known [11]. In addition, the network topology has a considered influence in the accuracy of this prediction [12].

There are also works that use real measures trough round trip times. Using the RTT, the real network distance between two components can be obtained. AICN [13] (*Anycast for Information Centric Network*) integrates both IP layer anycast and application layer anycast. This system considers RTT and server loads on the service selection. The problem is that, by using IP layer anycast, all its disadvantages cited before appear on the system. For example, the AICN needs special routers in order to route the client requests.

GAA [14] (*Global Application-layer Anycast*) is also a scalable application layer anycast framework. It derives from another system, Meridian [15]. The difference between these works is that the first uses client centralized probing, while the second uses server centralized probes. In GAA, each node has a neighbourhood divided in rings. Figure 3 shows an example of a server node neighbourhood. When a client requests a service, a random server node is chosen and the request is routed trough the servers using this neighbourhood scheme. This random choice of the initial node consists into a problem. The disadvantage of this action is the fact that if a distant initial node is chosen, the algorithm take more steps to converge. Thus, the total request times increase.

As discussed, many works use prediction of the distance to find the best node available. But, researches concluded that the error introduced by this technique can harm the selection performance. Also, problems were identified into the others application layer anycast systems. This shows that anycast systems still have weak points that can be explored by new researches. On our work, we focus on developing a scalable and efficient anycast system that uses real distance values (by means of RTT). We use the idea of concentric rings neighbourhood from the GAA system and solves the initial server selection problem. The system proposed is called GALA (Global Application Layer Anycast).

III. Design Details

A. Geolocation Database

As said before, GALA utilizes the GAA algorithm to find the best server available. The main problem of this algorithm is the random initial server selection. Thus, it is necessary to define a strategy to select the best initial server as possible. To do this, we use geolocation. Geolocation is the action to map a geographic position using an IP address as key. Because there is correlation between RTT and the geographic distance between two hosts, as Figure 4 shows, this metric is adopted by our work.

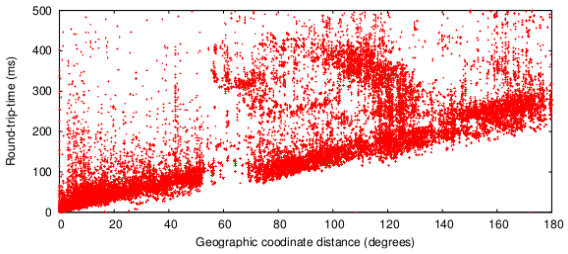

Figure 4. Correlation between RTT and geographic distance [9]

There are many ways to obtain latitude and longitude from a host IP [16]. The DNS (Domain Name System) can incorporate the IP location. An user can send a query to the *whois* database. Also, the user can use the *traceroute* information to map the routers of a path and then tabulate exhaustively IP addresses and their localities. In addition, many geolocation public databases can be used. Our system utilizes such public databases in order to find the initial server.

Among the many available databases in the Internet, we chose the MaxMind database. Comparative studies [17] [18] showed that none of the available databases have 100% accuracy. Also, their performance were better in a country level in comparison to a city level. The MaxMind

database obtained the best results comparing to the other free databases.

B. Anycast Query Workflow

First, we define the workflow of an anycast request. The Figure 5 shows the overview of the GALA system. According to it, there are four main steps in an anycast query workflow: start (1), mapping (2), resolving (3 and 3.1) and search (4).

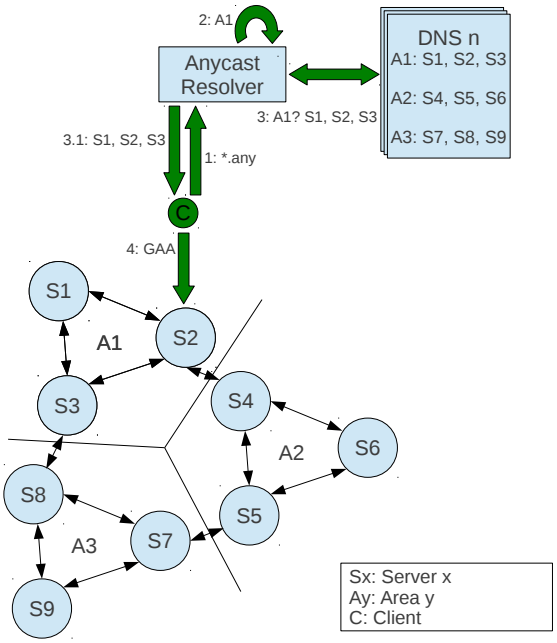

Figure 5. Overview of the GALA system

First, the client starts the anycast query sending a message to the local resolver. The anycast address is identified by the ".any" suffix. The local resolver knows the area it belongs, identified by the domain name A1. Therefore, it queries the DNS using this name. Then, the DNS answers with a list that contains the servers that are in that area, *S1, S2, S3*. After that, the local resolver sends this data to the client, which starts looking for the best server using a component from the list. The search is done as specified by the GAA algorithm. In the end, the algorithm will return the best server for the client.

C. Start of the Anycast Query

The first step is to resolve the anycast domain name into a traditional host name. The structure adopted for an anycast domain name is on the form "*.any". The client must query a DNS that supports this format. To do that, a specific resolver is proposed. The main features of this resolver are:

- Receive DNS packets, according to RFC 1035 [19];
- Identify anycast names;
- Forward name resolutions that are not anycast names;
- Answer queries with the anycast format.

D. Anycast Query Mapping

After identifying an anycast name, the resolver performs the query mapping into a traditional name. Using its IP address, the resolver extracts the geographic coordinates e maps them into a server name that represents that region. Figure 6 shows the mapping steps.

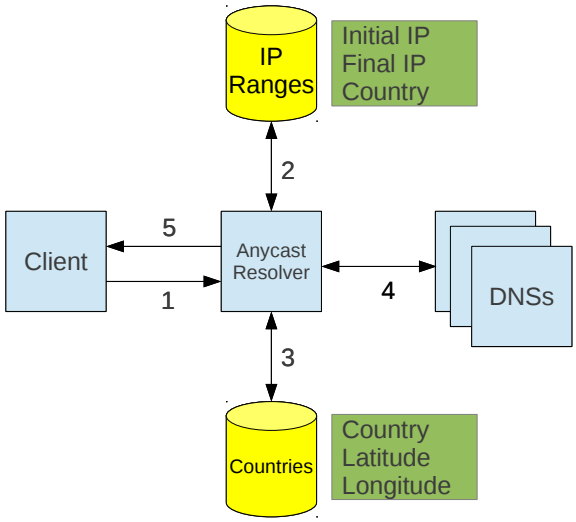

Figure 6. Resolver mapping steps

First, the client sends a DNS packet with the anycast domain name, such as *gala.com.any*, to the resolver (1). Then, the resolver, knowing its global IP address, maps its geographic coordinates using a geolocation database. This step can be achieved by querying two local databases, offered by the MaxMind service (2 and 3). Moreover, these actions can be done in background.

The first database is used to find the country of the given IP address (2). It contains IP ranges and their respective country codes, as specified by ISO3166 [20]. The second database contains the country codes and their latitude and longitude values. These registers were obtained from the World Factbook [21].

Yet, the resolver maps the coordinates obtained to a geographic region. The total number of regions is variable and defined by the user ("regionNodesCount"). Formulas 1 and 2 shows the calculus of the geographic region.

$$y = \frac{latitude + 90}{\frac{180}{regionNodesCount}}, \quad (1)$$

$$x = \frac{longitude + 180}{\frac{360}{regionNodesCount}}. \quad (2)$$

Figure 7 illustrates a mapping example considering 36 regions. The grid starts at the top left corner [0, 0] and ends at the bottom right corner [5, 5]. Brazil is located at [2, 3].

Finally, after obtaining x and y values, the resolver assembles a new host name in the format $xXyY$.name (for

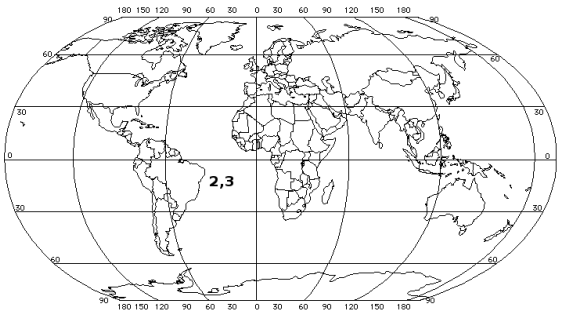

Figure 7. Mapping example considering 36 geographic regions [22]

example, $x2y3$.gala.com). Using this host name, it sends an anycast query to the DNS (4). The answer of this operation is forwarded to the client (5). The list obtained contains the group of servers on the same geographic region of the client.

E. GALA Servers Neighbourhood

The system servers are responsible for offering services to the clients. They communicate between themselves within a concentric neighbourhood. Figure 8 shows how this neighbourhood is formed. It is formed by R rings. The closest neighbours are located in the inner rings, while the distant ones are located in the outer rings. The proximity of a server is defined by the RTT.

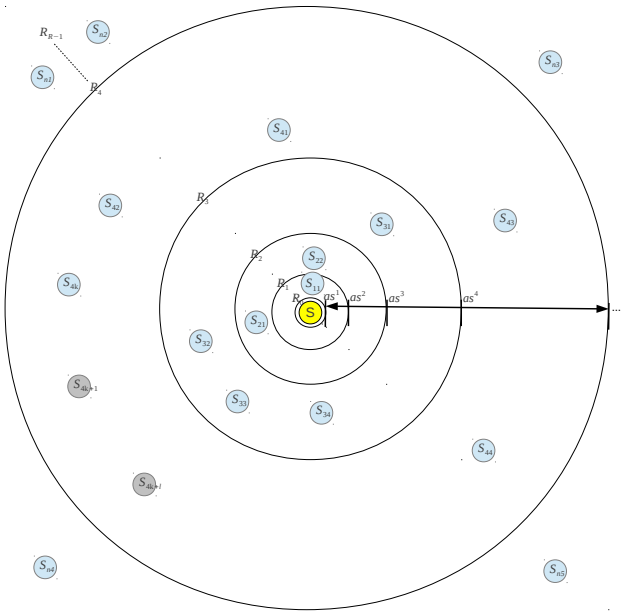

Figure 8. GALA neighbourhood

Using the RTT of its neighbour, a server places it on the suitable ith neighbourhood ring. The formula use is $R_i = as^i$. a and s are two parameters predefined by the system. Another parameter k is used to define the maximum ring nodes. Thus, the maximum number of neighbours of a server is kR.

When a server enters in the system, it chooses a host and inherits its neighbourhood. This choice is made using the same geolocation strategy in the client anycast query, previously described.

Periodically, each server communicates with their neighbours, in a process called gossip. First, the server chooses a random node from each ring, forming a G group of servers. Then, it sends this group to another node randomly chosen in each ring. Each receiver adds the servers in G to their neighbourhood, including the server who sent the message. For each server in the group, the receiver calculates the corresponding ring and updates its neighbourhood. If the server is new, the receiver adds it. If it already belongs to the neighbourhood, the receiver moves it to the new ring. In addition, if a server in G is new in the receiver neighbourhood, it sends all servers from the same ring to the new component.

A server can receive many gossip packets over time. Hence, it will acquire new neighbours and the number of nodes in a ring will overflow the capacity k. So, another parameter l is used to define extra (or secondary) ring nodes. Periodically, the server prunes its rings in order to maintain $k + l$ nodes in each ring. The prune process is described as follows. If there are more than $k + l$ nodes in a ring, the server chooses secondary servers randomly and discards them until this condition is false. Then, it sends a probe packet to each neighbour on the ring. This message contains all servers in the ring. Each neighbour calculates its RTT to each component and returns to the server. Thus,the server will acquire a list with the ring distance vectors. Using this data, it uses a distance function to sort the neighbours. The k furthest servers will be placed as primary, and the l closest servers will be considered secondary. A ring will geographic distributed nodes offer more utility than a ring with clustered nodes [15].

The system servers use two distance functions: Chebchev and Euclidean. Formulas 3 and 4 shows how the distances are calculated.

$$D_{ij} = \max_{l=1}^{k} RTT_{il}, \tag{3}$$

$$D_{ij} = \sqrt{\sum_{l=1}^{k} RTT_{il}^2}. \tag{4}$$

F. Best Server Search

The client sends a packet to the initial system node, described previously. The distance between the initial node and the client is defined by D_{cs}. Then, the initial node reply the client with all its neighbours which distance, D_{sx_i}, satisfies $(1 - \beta)D_{cs} \leq D_{sx_i} \leq (1 + \beta)D_{cs}$. β parameter is defined by the system. After receiving the reply, the client calculates its distance to all the satisfying neighbours, D_{cx_i}. The client defines the closest node which satisfies

$D_{cx_i} \leq \beta D_{cs}$ as the new initial node and repeats this process. If no node suits the client, the algorithm defines the current initial node as the best server node, used to execute the client service [14].

IV. Performance Evaluation

A. Experiments Setup

We executed experiments using simulations to evaluate the GALA system compared to the GAA system. The main purpose of the experiments is to answer the hypothesis that the geolocation strategy to select an initial server will improve the system performance. We performed a complete factorial experiment, with two levels for each factor. For the analysis of the results, we used linear regression. Also, we generated the graphics using the Minitab software [23].

As response variables, we considered the hop count means, the efficiency means and the latency means. These values are obtained from the client requests. The hop count is the number of servers that participates in the request. The efficiency is the ration of the real closest server distance and the distance of the server found by the system. The latency is the request total time. Table I shows the experiments factors and levels. The input data are servers obtained from the PlanetLab [24] and GeoNames [25] data files.

Factor	Levels	Total
Algorithm	GAA, GALA	2
Distance Method	Chebchev, Euclideana	2
Input Data	PlanetLab e GeoNames	2

Table I
Experiment configurations.

Table II shows the simulation parameters. These values were set considering previous experiments made by Wong et. al and Ma et. al works. The RTT factor is used in the RTT calculus that uses a geodesic line to obtain this value [18].

Parameter	Value
Request turns	20
RTT factor	$5.128205 * 10^{-9}$
Ring nodes (k)	8
Secondary ring nodes (l)	2
Ring count (R)	11
a	1
b	0.5
s	2
Initial request turn	4
Regions count	64
Experiment replications	10

Table II
Simulation parameters.

B. Results

In the effects graphics (Figures 10, 12 and 14), the x axis represents the standardized effect for each factor combination, while the y axis is the result of the effects combination ordering by the formula:

$$\frac{i - 0.3}{n + 0.4}. \tag{5}$$

n is the number of combinations and i the index of the combination. The straight line in the graphic equals to $x = y$. If a point is to the left, it has a negative effect on the response variable. If a point is to the right, it has a positive effect in the response variable. To indicate if an effect is significant, Minitab uses the p-value test.

The bars graphics (Figures 9, 11 and 13) summarize the distribution of a response variable, showing its central tendency and its variability.

Figure 9 shows that the efficiency for all experiment configurations were greater than 95%. Figure 10 shows that the input data is the most significant factor. The algorithm is also significant. GAA results better efficiencies if compared to GALA. The distance method has a small positive significance, where the Euclidean method results in greater efficiencies if compared to Chebchev method.

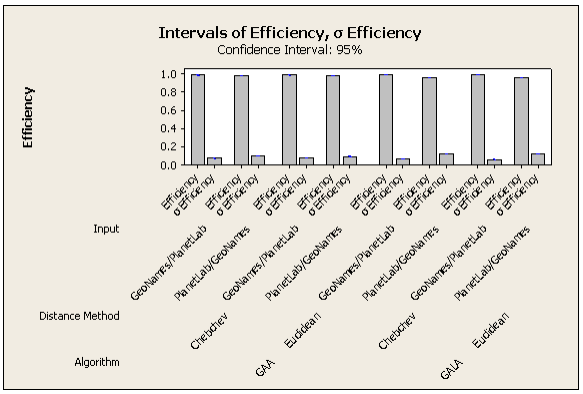

Figure 9. Efficiency: bars graphics.

According to Figure 11, we affirm that the GALA algorithm results in much smaller latencies than GAA. Figure 12 reveals that the algorithm is the only significant factor, for this response variable.

It is important to see that the latency standard deviation is not small. This indicates that the request times are disperse from the observed means. Table III exposes the overall minimum and maximum request latencies for each factor combination. According to this data, the maximum values for the GALA algorithm are always lesser than GAA values. In the latter, the randomness of the initial server choice explains this variation. For the GALA algorithm, the variation can be explained by the fact that there are clients remotely located in the map. These clients are located in geographic

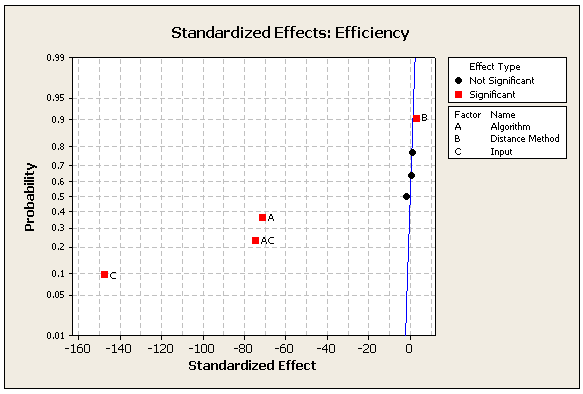

Figure 10. Efficiency: effects graphics.

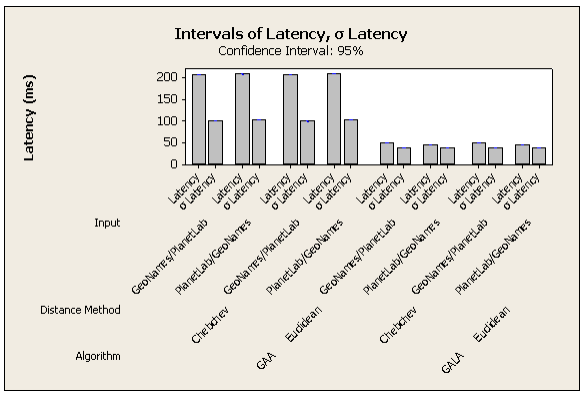

Figure 11. Latency: bars graphics.

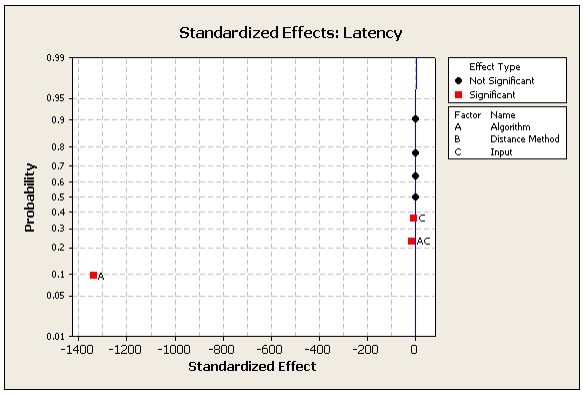

Figure 12. Latency: effects graphics.

regions where there are no servers available. Therefore, when they start a request, servers from the neighbour regions are used. Thus, the requests generated by these remote clients results into bigger latencies.

The hop count is lesser for the GALA algorithm, if compared to GAA. Figure 13 illustrates this behaviour. Analysing Figure 14, we say that the input and the algorithm were significant for this response variable. This is true because in GALA, the request starts at a close server. Thus,

Algorithm	Input Data	Dis. Method	Min & Max Latency
GAA	GeoNames/PlanetLab	Chebyshev	3 & 494 ms
GALA	GeoNames/PlanetLab	Chebyshev	3 & 254 ms
GAA	GeoNames/PlanetLab	Euclidean	3 & 494 ms
GALA	GeoNames/PlanetLab	Euclidean	3 & 227 ms
GAA	PlanetLab/GeoNames	Chebyshev	3 & 559 ms
GALA	PlanetLab/GeoNames	Chebyshev	3 & 397 ms
GAA	PlanetLab/GeoNames	Euclidean	3 & 553 ms
GALA	PlanetLab/GeoNames	Euclidean	3 e 383 ms

Table III
MINIMUM AND MAXIMUM EXPERIMENT LATENCIES.

the algorithm converges in less steps.

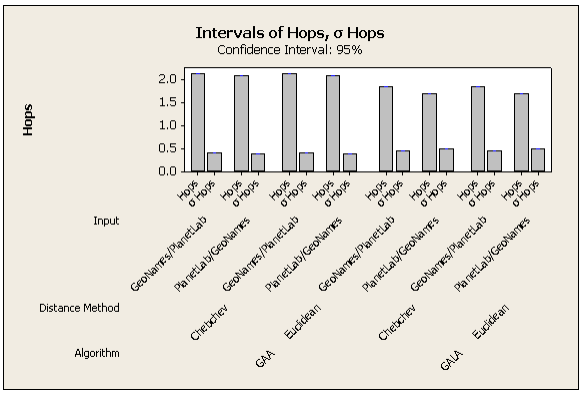

Figure 13. Hop count: bars graphics.

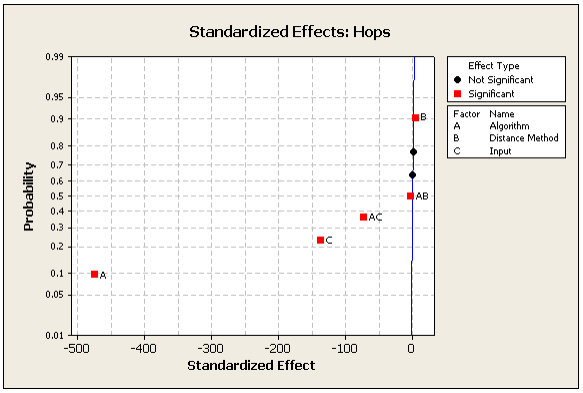

Figure 14. Hop count: effects graphics.

V. CONCLUSION

We conclude that the GALA algorithm has a better performance than GAA algorithm. Geolocation decreased the number of steps to find the best server. Thus, the time of the client requests also decreased. In addition, the GALA algorithm always starts its search in a close region to the client. This fact helped in the decrease of the request times. Yet, the results showed worse efficiencies on GALA, comparing to GAA. However, the input data was the main significant factor considering the requests efficiencies.

Therefore, we affirm that the proposed algorithm improves significantly the anycast system.

In the future, we plan to consider the server loads in the select process. This way, no overloaded servers will be selected to answer the clients requests. Also, we will replicate our experiments in a real environment.

ACKNOWLEDGMENT

We would like to thank FAPESP (Fundacao de Amparo a Pesquisa do Estado de Sao Paulo) for the monetary support of this research. Also, we would like to thank ICMC-USP (Instituto de Ciencias Matematicas e de Computacao) and the LaSDPC (Laboratorio de Sistemas Distribuidos e Programacao Concorrente) for offering the equipments and laboratories necessary for this study.

REFERENCES

[1] S. Chellouche, D. Negru, E. Borcoci, and E. Lebars, "Anycast-based context-aware server selection strategy for VoD services," in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, 2010, pp. 1513–1517.

[2] T. M. C. Partridge and W. Milliken, "Host Anycasting Service," 1993, rFC 1546.

[3] P. H. Piper, "Anycast DNS - Part 1, Overview," http://netlinxinc.com/netlinx-blog/45-dns/118-introduction-to-anycast-dns.html, 2013, accessed in 02/12/2013.

[4] S. Bhattacharjee, M. Ammar, E. Zegura, V. Shah, and Z. Fei, "Application-layer anycasting," in *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE*, vol. 3, 1997, pp. 1388–1396.

[5] Z. Fei, S. Bhattacharjee, E. Zegura, and M. Ammar, "A novel server selection technique for improving the response time of a replicated service," in *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, 1998, pp. 783–791.

[6] E. Zegura, M. Ammar, Z. Fei, and S. Bhattacharjee, "Application-layer anycasting: a server selection architecture and use in a replicated Web service," *Networking, IEEE/ACM Transactions on*, vol. 8, no. 4, pp. 455–466, 2000.

[7] E. Mykoniati, L. Latif, R. Landa, B. Yang, R. Clegg, D. Griffin, and M. Rio, "Distributed overlay anycast tables using space filling curves," in *INFOCOM Workshops 2009, IEEE*, 2009, pp. 1–6.

[8] S. Chellouche, D. Negru, E. Borcoci, and E. Lebars, "Context-aware distributed multimedia provisioning based on anycast model towards Future Media Internet," in *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, 2011, pp. 880–885.

[9] M. J. Freedman, K. Lakshminarayanan, and D. Mazières, "OASIS: anycast for any service," in *Proceedings of the 3rd conference on Networked Systems Design & Implementation - Volume 3*, ser. NSDI'06. Berkeley, CA, USA: USENIX Association, 2006, pp. 10–10.

[10] G. Wang, Y. Chen, L. Shi, E. K. Lua, B. Deng, and X. Li, "Proxima: Towards Lightweight and Flexible Anycast Service," in *INFOCOM Workshops 2009, IEEE*, 2009, pp. 1–2.

[11] R. Zhang, C. Tang, Y. Hu, S. Fahmy, and X. Lin, "Impact of the inaccuracy of distance prediction algorithms on internet applications - an analytical and comparative study," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, 2006, pp. 1–12.

[12] C. Xing and M. Chen, "Impact of network topology on distance prediction accuracy," in *Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on*, 2008, pp. 1425–1429.

[13] S. Yuan, T. Lin, G. Zhang, Y. Li, H. Tang, and S. Ci, "A future anycast routing scheme for information-centric network," in *Communications (APCC), 2012 18th Asia-Pacific Conference on*, 2012, pp. 173–178.

[14] Z.-Y. Ma, J. Zhou, and L. Zhang, "A Scalable Framework for Global Application Anycast," in *Information and Computing Science, 2009. ICIC '09. Second International Conference on*, vol. 1, 2009, pp. 250–253.

[15] B. Wong, A. Slivkins, and E. G. Sirer, "Meridian: a lightweight network location service without virtual coordinates," New York, NY, USA, pp. 85–96, 2005.

[16] V. N. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for internet hosts," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 173–185, aug 2001.

[17] Y. Shavitt and N. Zilberman, "A geolocation databases study," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 10, pp. 2044–2056, 2011.

[18] B. Huffaker, M. Fomenkov, and K. Claffy, "Geocompare: a comparison of public and commercial geolocation databases," May 2011.

[19] P. Mockapetris, "Domain Names - Implementation and Specification," 1987, RFC 1035.

[20] ISO 3166, "ISO 3166 Codes (Countries)," "http://www.iso.org/iso/country_codes/iso_3166_code_lists/country_names_and_code_elements.htm", 2013, accessed in 08/08/2013.

[21] CIA, "CIA World Factbook," https://www.cia.gov/library/publications/the-world-factbook/, 2013, accessed in 08/08/2013.

[22] B. Jones, "The world," "http://cse.ssl.berkeley.edu/segwayed/lessons/search_ice_snow/ski.2b1BigMap.html", 1998, accessed in 10/02/2013.

[23] Minitab Inc, "Software for Statistics, Process Improvement, Six Sigma, Quality - Minitab," "http://www.minitab.com", 2013, accessed in 12/06/2013.

[24] Princeton University, "PlanetLab," https://www.planet-lab.org/, 2012, accessed in 03/09/2013.

[25] Marc Wick, "GeoNames," http://download.geonames.org/export/dump/, 2013, accessed in 11/13/2013.