

The Study of Latency in Fog Computing

Aditi Jha¹ and Prof. Pravesh Patel²

¹Master of Technology/Computer Engineering, Mehsana, India

Email: first.author@hostname1.org

²Ganpat University – U. V. Patel College of Engineering, Mehsana, India

Email: pravesh.patel@ganpatuniversity.ac.in

Abstract— Cloud providing services like storage, computing, and processing through the internet. IoT produces large amount of data day by day, to store those data on cloud and access it which leads to high latency. To overcome this problem, Fog computing comes in the scenario, the fog network resides the edge of network which can allow direct access of the data. In this paper, we discuss about the fog computing and approaches through which we can estimate latency in better way.

Index Terms— fog computing, estimate latency, NCS algorithm, Round Trip Time and Vivaldi algorithm.

I. INTRODUCTION

In the recent technologies, cloud computing is one of the most popular due to its capability in diverse. Cloud computing is based on data networks (DNs) which are centralized servers responsible for computation and storage. The request and resource demands are analyzed and compute within the DNs[1]. However, the technology rises, the amount of data are also risen. On the other side, the IoT technology is becoming popular.

In 2012, the total revenue of IoT based application is \$4.8 billion. It is statically estimated that by 2020, around 50 billion devices will be connected to the internet. CISCO also estimates the global profit will also increase by 21 percent[2]. The data center (DCs) exhaust due to their massive amount of data which is release enormous amount of greenhouse gases (GHGs), mostly Co₂. This will increase the atmospheric pollution. The technology IoT is based on cloud computing. The data of IoT devices are to be processed within the cloud DCs.

The vehicular system, smart driving, smart grids are on great demand they need to be real time, low latency services. Since the large amount to data need to processed, compute and storage within DCs with arises large amount of network traffic and experienced the large network bottleneck, high service latency and poor quality of service (QoS). The fog computing is emerged to solve all the large amount of data issues of cloud computing. It serve the on demand services to the latency sensitive applications.

In 2012, the term coined by the CISCO is a distributed computing paradigm[3]. The fog computing brings the cloud closer to the IoT devices. The fog data processing and storage locally at the IoT devices rather than sending them to the cloud. Both the cloud and fog provides the storage computing and network resources[4]. The purpose of fog computing to reduce the amount data transferred to the cloud for storage, compute and analyze. Therefore, the data collected by sensors will send to the fog network edge for processing and temporary storing with will reduce the network traffic and latency.

The latency is the execution time of task in other words how much a packet will need to travel from one location to another location. The fog computing gives the more choices to the processing data, it may be processed as quickly as possible so fog computing can create low latency between node. It will also reduce bandwidth as data does not need to send to the cloud DCs. Low latency is the big issue for delay sensitive application such as live streaming, video surveillance and online data analysis. While the distance from sensing and processing is reduced by fog computing. The cross-fog latency is yet to be determined.

Some IoT application are delay sensitive applications and real time. In other words, sensing data should immediately processed to trigger respectively actuator without any delay. For example, connected traffic vehicle system need to instantly process to prevent accidents, even a millisecond delay will affect the life and assets. Now a days, latency calculation is primarily conduct thought an assistance of Network Coordinate System (NCS), which create a virtual positioning system for every node.

II. LATENCY ESTIMATION TECHNIQUES

NCS estimates latency based on known network coordinates in virtual geometric vector. NCS plots the network node as per the Euclidean distance by mapping the delay into the measured distance[5]. The latency prediction formula for a dimensional Euclidean nodes,

$$dP_{A-B} = \sqrt{\sum_{i=1}^n (A_i - B_i)^2}$$

Where, dP_{A-B} is the prediction delay and, A_i and B_i are coordinates.

The absolute relative error (RE) is a performance metrics,

$$RE = \frac{|E - RTT|}{\min(E, RTT)}$$

Here, E is the estimated latency and RTT is the actual distance. Various NCS developed based on OWD (One way delay) and RTT (Round trip time). OWD techniques required highly qualified system with synchronized clocking to aid the delay. However, its utilization is limited because of having an undesired error. That is the reason, most of the NCS use RTT to measured latency.

A. Landmark based Coordinate System

a. Global network positing

GNP has small number of landmark nodes to compute synthetic coordinates[5]. It has two faces to positioning nodes.

i. Landmark phase

In landmark phase, each landmark measure RTT to other landmark.

ii. Ordinary host phase

Ordinary host measure its RTT to landmark to work out its coordinates.

The ordinary host gets its coordinate using simplex downhill method. Its performance relies on distribution of landmark and great with perfectly chosen landmark, degrades when landmark is poor.

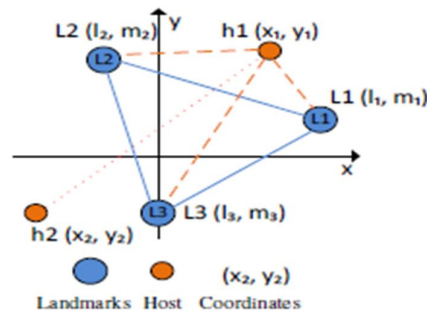


Figure 1. GNP

b. Network Positioning System

NCS allow any arbitrary node to act as landmark managed by independent membership server. Any node can query its membership server for more information like list refreshing point than nodes involves in probing

process to update position with respect to refreshing point. Until its position is stabilized. As a result position consistency achieved.

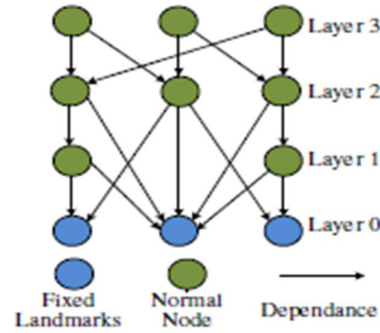


Figure 2. NPS[5]

B. Distributed Network Coord- Inate System

a. VIVALDI

Unlike landmark based NCS, this does not need any predefined landmark or infrastructure to predict latency[6]. The natural length of spring assumed to be distance between two nodes locally. If spring is stretched, RTT is over-estimated while spring is compressed RTT is under-estimated. Likewise, the natural length of RTT is exactly estimated. According to the Hooke's law, the spring length between nodes a, b, c and d is distance between them. Vivaldi set the synthetic coordinates so it estimate RTT with less error.

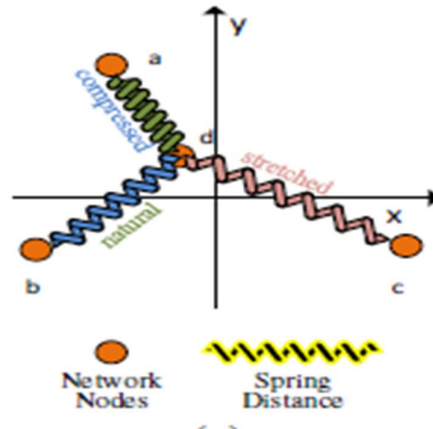


Figure 3. Vivaldi

The squared error function is,

$$E = \sum_i \sum_j \left(RTT_{ij} - \|x_i - x_j\| \right)^2$$

Where, RTT_{ij} is the actual latency, $\|x_i - x_j\|$ is the estimated distance between nodes i and j .

During the process of minimizing error, nodes are push and pull to get reduced error. When the new add in Vivaldi cluster, it will assign new random coordinates as it communicate with other nodes. It will get its exact coordinates. Vivaldi nodes move by a small time step δ , with each movement reduce error with respect to the other node. to chose the accurate time step is still a open issue for Vivaldi.

b. PHAROS

Vivaldi is maintaining global spring system with significantly chargeable distance, which become hard to handle. Pharos divide nodes into two different nodes, both short and long distance coordinates are assign to each node.

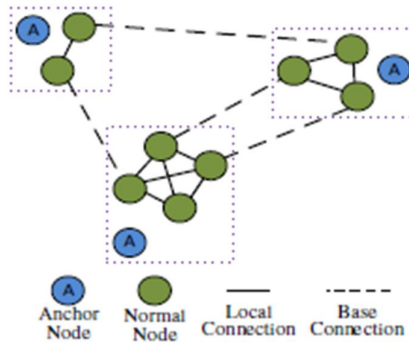


FIGURE 4. Pharos

As a result, both short link cluster and medium or long link cluster are separated. When new node joined the system, it broadcast its latency to locate the nearest node. And joined cluster. At the end, Vivaldi is used to attain the two set of coordinates for both overlay.

III. THE COMPARISON

Every NCS has merits and demerits. Landmark system required stable landmark to serve other nodes, it must be well distributed and available. The change in distance may not update which lead inaccurate estimation. A good landmark based NCS algorithm can accurately predict latency with direct communication with other nodes. On the hand, distributed NCS does not need any specific device. It relies on each to adaptive positioning which does not benefit to low end nodes but those nodes response best to the network, changes to help us up-to date latency estimation.

IV. CONCLUSION

The point is we can only minimize latency rather than eliminate it. Latency prediction is important to calculate latency. By using landmark and distributed NCS algorithm, one can prefer to calculate latency between the nodes and things. We hope the above simple explanation will help in concerned work.

REFERENCES

- [1] 이동만 (KAIST), "Fog Computing 동향과 응용 사례 분석," *Futur. Internet Forum Newsl.*, pp. 2–5, 2015.
- [2] R. Mahmud, K. Ramamohanarao, and R. Buyya, "Latency-aware application module management for fog computing environments," *ACM Trans. Internet Technol.*, vol. 19, no. 1, 2018.
- [3] H. J. Hong, "From cloud computing to fog computing: Unleash the power of edge and end devices," *Proc. Int. Conf. Cloud Comput. Technol. Sci. CloudCom*, vol. 2017-Decem, pp. 331–334, 2017.
- [4] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," *Proc. - 3rd Work. Hot Top. Web Syst. Technol. HotWeb 2015*, pp. 73–78, 2016.
- [5] J. Li, T. Zhang, J. Jin, Y. Yang, D. Yuan, and L. Gao, "Latency estimation for fog-based internet of things," *2017 27th Int. Telecommun. Networks Appl. Conf. ITNAC 2017*, vol. 2017-Janua, pp. 1–6, 2017.
- [6] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," *Comput. Commun. Rev.*, vol. 34, no. 4, pp. 15–26, 2004.