Fog Computing Made Easy with the Help of Citrix and Billboard Manager

Rajesh Bose
Ph.D. Scholar, Department of Computer Science & Engineering, University of Kalyani, West Bengal, India

Murari Krishna Saha
Assistant professor, Computer Science & Engineering, B. A. College of Engineering & Technology, Jamshedpur, Jharkhand, India

Debabrata Sarddar
Department of Computer Science & Engineering
University of Kalyani, Nadia, West Bengal, India

ABSTRACT

A model which allows storage of data, data processing and associated application on devices along the edge of a network of a cloud is called Fog Computing. This is also called fogging. In such a model, devices are not concentrated within a cloud. This, therefore, translates to data and information processing carried out on smarter, agile, and entirely local devices. In other words, the bulk of the data processing duties are taken away from the cloud. Fog Computing counts as one of the approaches which deal with the incessant and growing needs of devices which are connected to the Internet. Further, the approach is a key element in support Internet of Things in which components stay connected to each other without interference from humans or from other machines. In this paper, we put forward our proposed model which we call Billboard Manager based Fog Node Management Approach. This model will help user to cater to their processing needs from an appropriate fog node that is registered with the Billboard Manager. The Billboard Manager keeps track of all fog nodes that are currently available and in service. When a mobile user sends a request, the Billboard Manager receives that request and begins ascertaining the best-fit fog node that would be able to process that task. Upon selecting the fog node, the Billboard Manager hands over the processing duties to that fog node. We also discuss, in this paper, a detailed architecture of our proposed model involving Citrix XenApp.

Keywords

Cloud, Fog, Billboard Manager, Citrix, Internet of Things

1. INTRODUCTION

Not long ago, Cisco – one of the leading companies in the world to manufacture and market pioneering networking products – showed how fog computing can be made possible through connected Internet of Things (IoT). Cisco demonstrated how applications can be enabled and run on billions of such devices at the edge of the network [1]. Fog Computing bridges the gap between data centres running in a traditional cloud computing architecture, and end points. While doing so it also provides for a secure, virtual, thoroughly network integrated and a systematic platform which accords processing power, storage, and network service [2]. In this kind of model, applications and related data processing occurs at the very edge or periphery of the network in a cloud computing environment. The data processing and execution of the applications are no longer concentrated within the Cloud. In doing so, Fog Computing brings applications and data closer to end-users. This is the aim of Fog Computing. One of the key technologies used to enable Fog Computing is the drop [3]. A drop is a microcontroller which comes with its own built-in memory, an interface for data transfer, and wireless connection Mesh chip bundled together as a single unit. A drop is powered by a small battery which can last for a couple of years. Users are free to connect any type of sensors which can detect temperature, light and voltage. Fog Computing is layered beneath Cloud and acts as a gateway which permits optimization of transfer of data and services. The structure of Fog Computing in association with Cloud is shown in the following Figure 1.

Fog Computing is not intended to be a replacement for Cloud computing. In fact, it complements Cloud computing functionalities and enables optimized flow of data and information to users. In a scenario where Fog Computing is implemented, services are executed at the very edge of the network or even on devices that form the terminal points of the network. Such devices could be set-top-boxes or even access points, among others. Developed on current and existing technologies such as Content Delivery Networks (CDN), Conceptual Fog Computing improves upon those. Although based on Cloud Computing technologies, Fog Computing aims at delivering more complex services [4, 5].

1.1 Characterization of Fog Computing [2]

As a highly evolved form of a virtualized platform, Fog Computing extends processing power, storage needs, and networking services among end devices and data centers operating within a Cloud Computing framework. However, in Fog Computing concept, these are not the only devices to which it is limited to. Computing and storage devices and network resources constitute both Cloud and Fog. In order to understand the idea of the "edge of the network", it becomes essential to understand the following points which help in clarifying as to why Fog is not merely an extension of Cloud Computing.

1.1.1 Location awareness, low latency, and edge location:

The roots of Fog Computing emanate from initial observations and findings that posited support for endpoints that were rich in resources and sat at the periphery of the network. This also included applications with low latency specifications that needed to be extremely responsive. A few examples of such applications are those which include games, video streaming, etc.

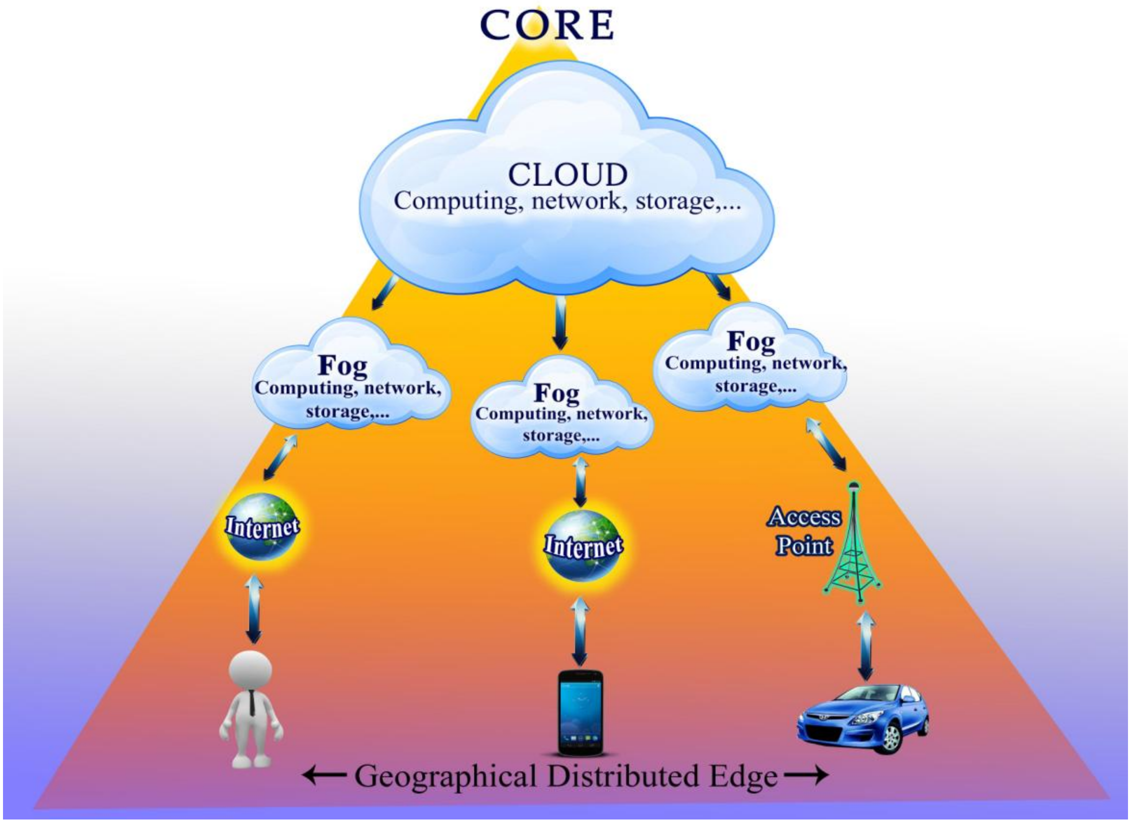
CORE
CLOUD
Computing, network, storage,...
Fog
Computing, network, storage,...
Fog
Computing, network, storage,...
Fog
Computing, network, storage,...
Internet
Internet
Access Point
Geographical Distributed Edge

Figure 1: Fog Computing Architecture

1.1.2 Geographical distribution:

Set against the backdrop of the usual more centralized Cloud Computing framework, Fog Computing model requires services and applications to seek out distributed deployments in order to be relevant and effective. For instance, where high quality streaming to moving vehicles is required, Fog Computing can be useful to deliver the content at the required speeds with little or no loss in quality and flicker-free transmission through utilization of proxy and access point devices strategically located along highways, roads, and railway tracks.

Networks that are built to monitor the environment, including Smart Grid and other forms of distributed systems, benefit greatly through implementation of Fog Computing. Sensor networks that are installed in diverse, and often geographically distant locations, can be optimized through Fog Computing model.

1.1.3 Mobility support:

It is often imperative that Fog applications need establishing communications directly with mobile devices. Therefore, support for mobile technologies, for instance the LISP protocol 1, which require decoupling of the host identity from the location one, require a directory system that is distributed.

Real-time Interactions:

Mission-critical Fog Cloud application engages in real-time interactions rather than depend on batch processing modes.

1.1.4 Preponderance of wireless communication and access:

Fog Computing enables heterogeneity: Nodes used in Fog Computing infrastructure might comprise various form factors. All these will be deployed in environments which are diverse and unique.

1.1.5 Federation and interoperability:

Components in a Fog Computing model must have capabilities to interoperate. Their services must also be federated spanning several domains. Seamless and uninterrupted support of certain services requires active support and participation of different providers. One such example is a video streaming service.

1.2 Zones and farm in Citrix technology [6]

In a Citrix farm comprising several Citrix servers, zones form the logical segments. Within a zone resides a data collector. In a zone, servers will communicate with the zone's data collector. Data collectors of individual zones will also share information on the respective zones. Any change within a farm is distributed to Citrix server so that the change in settings is recorded. More the number of logical changes, greater the amount of traffic generated among Citrix servers. In harmony with the available bandwidth, the quantum of changes is sufficient to segregate a farm into zones. Each zone requires a data collector. Although a server can serve as a data collector, logically this role requires that some resources be free and available to perform tasks. The data collector is a role that functions on a Citrix XenApp server. It serves to collect, maintain and manage runtime information on the farm and the respective zones. The data collector tries to ensure that a user is connected to the server that is least busy. Every Citrix XenApp server can support a server role which requires resources. Zones serve the following purposes:

- In a hierarchical structure, zones collect data from member servers.
- Zones also propagate changes to all servers in the farm as efficiently as possible.

Every zone houses a server that is designated to function as its data collector. Such data collectors collect and store information on the servers and applications that are published in the respective zones. Data Collectors also double up as communication gateways between two or more zones in farms where more than a single zone exists.

The following Figure 2 shows a sample layout of a server farm linking two zones and interconnected data collectors over WAN.

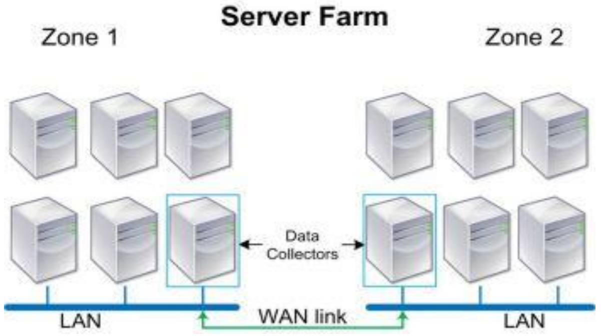

Figure 2: A sample diagram of a server farm linking two zones and data collectors over WAN

Data collectors must act to replicate changes throughout the farm to all other data collectors. However, this produces an effect whereby consumption of bandwidth and network traffic also increases in proportion with the number of zones in a farm. In communicating with each other almost on a constant basis, data collectors tend to produce a high amount of network traffic. This is brought about as a result of the following:

Every data collector in a zone has an open connection to all other data collectors within the farm. Member servers notify data collector on requests made and data changed during a zone update. Data collectors pass on changes to other related data collectors. As a result, data collectors end up maintaining session information for all zones.

The rest of our paper is segregated into several sections. We discuss related work in section 2. In section 3, we discuss our proposed work. In section 4, we present an algorithm for our proposed method. We conclude our paper in section 5. In sections 6, 7 and 8 we present our acknowledgements, references, and authors' profiles, respectively.

2. RELATED WORK

In their work, B. Ottenwalder et al., presented a method for placement and migration of Cloud and Fog resource providers [7]. The authors devised a model wherein network utilization can be reduced by planning migration in advance. Further, the model also ensures that latency restrictions which are of end-to-end type and defined by applications are in place. The authors also present how the bandwidth required for migrating virtual machines can be cutback through enabling of application knowledge of complex event processing system. While computationally intensive operators can be put directly in Cloud, operators that are network-intensive by nature are placed on distributed devices in a Fog model. In ensuring low expected network utilization that lasts for sufficiently long time, select migration targets are chosen whereby migration costs are gradually reduced. However, this work does little to optimize workload mobility because Fog devices are also able to perform tasks that are computation-intensive. Nor does it optimize the size of control information or mobility overhead. Further, network control policies for locating efficient paths for various applications are not described.

A unique procedure was demonstrated by J. Zhu et al. while applying existing web optimization methods [8]. Within the context of Fog computing, these methods can be connected to knowledge that is unique and only available with Fog devices. Increasing runtime adaptations to user's conditions can also be achieved with knowledge that is specific and resides at the edge of the network. Consequently, the rendering performance at user-level is improved through implementing those methods on the web server.

The authors, BE TaaS [9], proposed "local cloud" of gateways to replace Cloud as the resident in cases of machine-to-machine applications. In their work, the authors have presented a model that is made up of smart devices connected to the Internet to act as a "local cloud". These smart devices could be smart phones, home routers, and even road-side units. Such local clouds would be able to empower applications that are constricted by time and space, to demand only simple interactions that repeat over and over again. This also enables applications to respond in a manner that is consistent.

The authors, K. Hong et al. proposed Mobile Fog computing [10] to put forward a high-level programming model that is aligned for large scale, latency-sensitive, geo-spatially distributed Internet applications of the future. As would be evident from the logical structure presented in Figure 1, low-latency processing takes place near the edge of Cloud networks. The large scope aggregation which is also latency tolerant is performed on resources that are powerful and reside at the heart of the Cloud network. Mobile Fog is built up of a set of event handlers and functions that can be called by an application. This model is not generic in nature. Rather, it is targeted at specific applications while excluding features and functionalities that are required to overcome the technical hurdles in processing primitives related to images. Fog Computing approach as presented by these authors are seen to reduce latency and network traffic.

3. PROPOSED WORK

In this paper, we present a Fog and a Cloud-based architecture that is buttressed by Citrix XenApp technology. In this architecture, we consider a cloud node a farm, and a fog node as a zone. This is analogous to Citrix XenApp technology. Our proposed model is actually placed in between zones or fog nodes which are situated at the very edge of mobile networks. Different fog nodes or zones are part of cloud nodes or farm. Each of these communicates with each other at regular intervals. We posit a Billboard Manager framework in which several Billboard Managers are each connected to respective fog nodes. The Billboard Managers, too, are inter-connected.

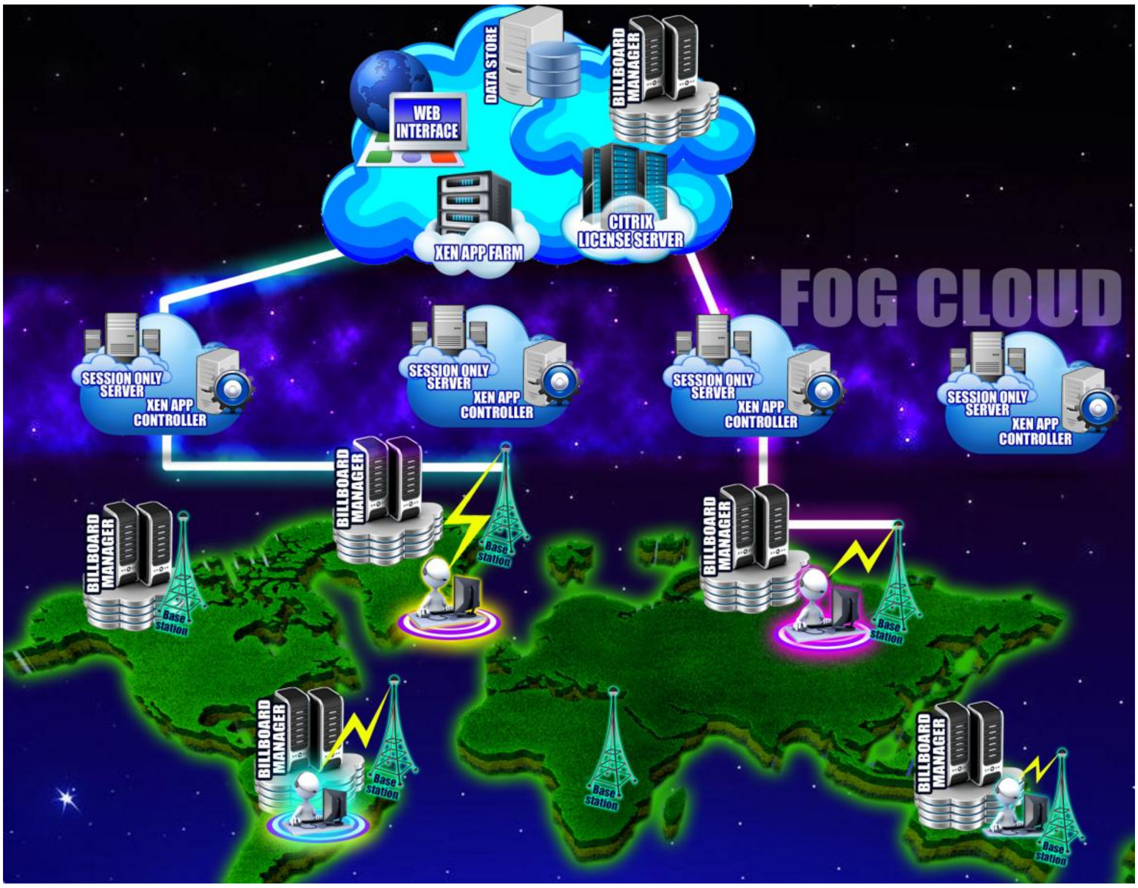

Figure 3: Pictorial depiction of our proposed work

In our model, a user only has to issue a search request to access a particular cloud-based application. The search request is taken up by the nearest Billboard Manager. All fog nodes in the region where the user is situated, is registered to this Billboard Manager. Based on certain conditions and parameters, the Billboard Manager routes the request to the nearest fog node. Thus, the user is directly connected to this node and is able to access data locally.

The Billboard Manager framework consists of a central Billboard Manager at the heart of the Cloud. The central Billboard Manager keeps track of all the other Billboard Managers connected to fog nodes.

When a particular search request cannot be serviced by a fog node Billboard Manager, a query is placed with the central Billboard Manager. The central Billboard Manager then looks up in its search index for the nearest Billboard Manager that can service the request. Once located, the user is directly connected to the fog node.

Service requests may not be fulfilled by user’s local Billboard Manager for reasons ranging from unavailability of fog nodes, to where applications are not found anywhere within the local fog node.

In Figure 3, we present a layout of our proposed architecture detailing therein the various Fog Cloud zones managed by Billboard Managers.

4. PROPOSED ALGORITHM

1. The central Billboard Manager keeps track of all connected Billboard Managers in fog nodes. The central Billboard Manager maintains an index and a database in which it records channel bandwidth, last ping response times, IP addresses, list of applications hosted by the Billboard Managers at fog node level, and free nodes available under the respective Billboard Managers. The data is constantly updated at preset frequent intervals such that the index and the table maintained by the central Billboard Manager is up-to-date at a given point of time.
2. When a user initiates a request, the local fog node Billboard Manager immediately makes a note of the geographical position of the user and hardware details of the device from which the request was initiated.
3. The Billboard Manager then looks up the location in its database.
4. The Billboard Manager then checks the nearest available fog in its area.
5. In case an appropriate fog node is available to service the request, a handover is initiated and the user is connected directly to the resource by the Billboard Manager.
6. In case the Billboard Manager is unable to service the request, it contacts the central Billboard Manager.

7. The central Billboard Manager stores and updates periodically information related to fog cloud nodes. Such information relates to capacity available, IP address, shortest node distances, channel bandwidth, applications hosted, and last ping response times. Using this dataset, the central Billboard Manager zeroes in on the best fit fog node that can service the user's request.
8. Once a fog node has been identified, the user is connected directly to the fog node.
9. In case the request can still not be serviced owing to unavailability of the application being sought, the Billboard Manager places the request on the Cloud for further processing.

5. CONCLUSION

In our work we have examined how Citrix XenApp technology can be used to support our proposed Billboard Manager based Fog Computing model. We have explained how, with the use of Fog Computing concept, data storage and processing can be taken to a level where efficiency can be markedly improved. Efficiency is boosted through use of local devices connected to Billboard Managers in a Fog Cloud that enables users to connect to resources faster and, thereby, reducing latency and the need for higher bandwidth. With the help of Citrix technology, application hosting cost is significantly rationalized. The Billboard Manager framework optimizes application selection and data storage on Fog Cloud computing platform.

6. ACKNOWLEDGMENTS

Authors gratefully acknowledge to CSE Department of University of Kalyani for providing lab and related facilities for do the research.

7. REFERENCES

[1] F. Bonomi, "Connected vehicles, the internet of things, and fog computing," in The Eighth ACM International Workshop on Vehicular Inter Networking (VANET), Las Vegas, USA, 2011.

[2] F. B ONOMI , R. M ILITO , J. Z HU , S. A DDEPALLI , Fog Computing and Its Role in the Internet of Things. Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, (2012), pp.13–16.

[3] E. R UDENKO , Fog Computing Is a New Concept of Data Distribution. (2013, December 5). Retrieved from CloudTweaks: http://www.cloudtweaks.com/2013/12/fogcomputing-is-a-new concept-of-data-distribution/

[4] R. B UEST, Fog Computing: Data, Information, Application and Services need to be delivered more efficiently to the end user. (2013, October 19). Retrieved from cloud user.de: http://clouduser.de/en/analysis/fogcomputing-data-information-application and services needs to be delivered more efficient to the end user 22362

[5] B. K LEYMAN , Welcome to Fog Computing: Ex tending the Cloud to the Edge. (2013, August 23). Retrieved from Data CenterKnowledge:http://www.datacenterknowledge.com/archives/2013/08/23/welcome to the fog a new type of distributed computing/

[6] Designing Zones for a XenApp Deployment: http://support.citrix.com/proddocs/topic/xenapp65-planning/ps-planning-zones-wans-v2.html

[7] B. Ottenwalder, B. Koldehofe, K. Rothermel, and U. Ramachandran, "Migcep: Operator migration for mobility driven distributed complex event processing," in *Proceedings of the 7th ACM International Conference on Distributed Event-based Systems*, ser. DEBS'13. ACM, 2013, pp. 183–194.

[8] J. Zhu, D. Chan, M. Prabhu, P. Natarajan, H. Hu, and F. Bonomi, "Improving web sites performance using edge servers in fog computing architecture," in *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*, March 2013, pp. 320–323.

[9] BETaaS, "Building the environment for the things as a service," BETaaS, Tech. Rep., Nov. 2012.

[10] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the internet of things," in *Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing*, ser. MCC'13. ACM, 2013, pp.15–20.

8. AUTHOR PROFILE

Rajesh Bose is currently pursuing Ph.D. from Kalyani University. He is an IT professional employed as Senior Project Engineer with Simplex Infrastructures Limited, Data Center, and Kolkata. He received his degree in M.Tech. in Mobile Communication and Networking from WBUT in 2007. He received his degree in B.E. in Computer Science and Engineering from BPUT in 2004. He has also several global certifications under his belt. These are CCNA, CCNP-BCRAN, and CCA(Citrix Certified Administrator for Citrix Access Gateway 9 Enterprise Edition),CCA(Citrix Certified Administrator for Citrix Xen App 5 for Windows Server 2008).His research interests include cloud computing, wireless communication and networking.

Murari Krishna Saha, is an Assistant Professor in the Department of Computer Science and Engineering, B.ACollege of Engineering and Technology, Jamshedpur, Jharkhand, INDIA. He completed his M. Tech in Computer Science & Engineering from Kalyani Government Engineering College, West Bengal in 2010, his MCA in Computer Science & Engineering from University of Kalyani, Kalyani, Nadia in 2008 and B.Sc(pure) from Abhedananda Mahavidyalaya under University of Burdwan, Burdwan, West Bengal. His research interest Cloud computing..

Debabrata Sarddar, Assistant Professor in the Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, INDIA. He has done Ph.D. He completed his M. Tech in Computer Science & Engineering from DAVV, Indore in 2006, and his B.E in Computer Science & Engineering from NIT, Durgapur in 2001. He has published more than 75 research papers in different journals and conferences. His research interest includes wireless and mobile system and WSN, Cloud computing.