

REVIEW OF IMPLEMENTING FOG COMPUTING

Pranali More¹

¹MCA 3rd Year, Mumbai University, 10/303 SamtaNagar Opp. Raymond company, Thane (w)-400606

Abstract

Fog Computing is a paradigm that extends Cloud computing and services to the edge of the network. Similar to Cloud, Fog provides data, compute, storage, and application services to end-users. It is a model in which data, processing and applications are concentrated in devices at the network edge rather than existing almost entirely in the cloud. This document describes the various features of Fog Computing and a case study along with the actual implementation of fog computing in traffic analysis to understand how fog computing is applied to the edge environment. This document also contains the difference between the fog computing and cloud computing.

Keywords— Fog Computing, Characteristics of Fog computing, Application of Fog computing, Difference between Cloud computing and Fog Computing.

-----***-----

1. WHAT IS FOG COMPUTING?

Fog Computing enables a new breed of applications and services, and that there is a fruitful interplay between the *Cloud* and the *Fog*, particularly when it comes to data management and analytics. Fog Computing extends the Cloud Computing paradigm to the edge of the network. While Fog and Cloud use the same resources (networking, compute, and storage), and share many of the same mechanisms and attributes (virtualization, multi-tenancy) the extension is a non-trivial one in that there exist some fundamental differences that stem from the Fog raison d'être. The Fog vision was conceived to address applications and services that do not fit well the paradigm of the Cloud. They include:

- Applications that require very low and predictable latency—the Cloud frees the user from many implementation details, including the precise knowledge of where the computation or storage takes place. This freedom from choice, welcome in many circumstances becomes a liability when latency is at premium (gaming, video conferencing).
- Geo-distributed applications (pipeline monitoring, sensor networks to monitor the environment).
- Fast mobile applications (smart connected vehicle, connected rail).
- Large-scale distributed control systems (smart grid, connected rail, smart traffic light systems).

2. KEY CHARACTERISTICS OF THE FOG COMPUTING

- **Heterogeneity.** Fog Computing is a highly virtualized platform that provides compute, storage, and networking services between end devices and traditional Cloud Computing Data Centers, typically, but not exclusively located at the edge of network. Compute, storage, and networking resources are the building blocks of both the Cloud and the Fog. "Edge

of the Network", however, implies a number of characteristics that make the Fog a non-trivial extension of the Cloud. Let us list them with pointers to motivating examples.

- **Edge location,** location awareness, and low latency. The origins of the Fog can be traced to early proposals to support endpoints with rich services at the edge of the network, including applications with low latency requirements (e.g. gaming, video streaming, augmented reality).
- **Geographical distribution.** In sharp contrast to the *more centralized Cloud*, the services and applications targeted by the Fog demand widely distributed deployments. The Fog, for instance, will play an active role in delivering high quality streaming to moving vehicles, through proxies and access points positioned along highways and tracks.
- **Large-scale sensor networks** to monitor the environment, and the Smart Grid are other examples of inherently distributed systems, requiring distributed computing and storage resources.
- **Very large number of nodes,** as a consequence of the wide geo-distribution, as evidenced in sensor networks in general, and the Smart Grid in particular.
- **Support for mobility.** It is essential for many Fog applications to communicate directly with mobile devices, and therefore support mobility techniques, such as the LISP protocol, that decouple host identity from location identity, and require a distributed directory system.
- **Real-time interactions.** Important Fog applications involve real-time interactions rather than batch processing.
- **Predominance of wireless access.**
- **Interoperability and federation.** Seamless support of certain services (streaming is a good example) requires the cooperation of different providers. Hence, Fog components must be able to interoperate, and services must be federated across domains.

3. FOG COMPUTING CASE SCENARIO

A Smart Traffic Light System (TLS):

TLS calls for the deployment of a STL at each intersection. The STL is equipped with sensors that :

- (a) Measure the distance and speed of approaching vehicles from every direction;
- (b) Detect the presence of pedestrians and cyclists crossing the street.

The STL also issues “slow down” warnings to vehicles at risk to crossing in red, and even modifies its own cycle to prevent collisions.

The TLS has three major goals:

- (a) Accidents prevention;
- (b) Maintenance of a steady flow of traffic (green waves along main roads);
- (c) Collection of relevant data to evaluate and improve the system.

Note that the global nature of (b) and (c), in contrast with the localized objective (a). Also note the wide difference in timescales: (a) requires real time (RT) reaction, (b) near-real time, and (c) relates to the collection and analysis of global data over long periods.

- To be specific, consider some numbers. Let us say that the green wave is set at 64 km/h (40 miles/h). A vehicle moving at 64 km/h travels 1.7m in 100 ms. The policy requires sending an urgent alarm to approaching vehicles when collision with crossing pedestrians is anticipated. To be effective the local control loop subsystem must react within a few milliseconds—thus illustrating the role of the Fog in supporting low latency applications. Accident prevention trumps any other consideration. Hence, to prevent a collision the local subsystem may also modify its own cycle. In doing so it introduces a perturbation in the green wave that affects the whole system. To dampen the effect of the perturbation a re-synchronization signal must be sent along all the traffic lights. This process takes place in a timescale of hundreds of milliseconds to a few seconds.

Key Requirements Driven by the TLS:

This section discusses the requirements of a smart traffic light system; the requirements are also highlighted in (Table 1)

1. **Local subsystem latency:** In TLS a subsystem includes the traffic light, sensors and actuators in a local region such that the reaction time is on the order of <10 ms.
2. **Middleware orchestration platform:** The middleware handles a number of critical software components across the whole system, which is deployed across a wide geographical area. The components include:
 - The decision maker (DM), which creates the control policies and pushes them to the individual traffic lights. The DM can be implemented in a centralized, distributed or in a hierarchical way. In the latter, the most likely implementation, nodes with DM functionality of regional

scope must coordinate their policies across the whole system. Whatever the implementation, the system should behave as if orchestrated by a single, all knowledgeable DM.

- The federated message bus, which passes data from the traffic lights to the DM nodes, pushes policies from the DM nodes to the traffic lights, and exchanges information between the traffic lights.

3. **Networking infrastructure:** The Fog nodes belong to a family of modular compute and storage devices. However different the form factors, and the encasings (a ruggedized version is required for the traffic light), they offer common interfaces, and programming environment.

4. **Interplay with the Cloud:** In addition to the actionable real-time (RT) information generated by the sensors, and the near-RT data passed to the DM and exchanged among the set of traffic lights, there are volumes of valuable data collected by the system. This data must be ingested in a data center (DC)/Cloud for deep analytics that extends over time (days, months, even years) and over the covered territory.

In a cursory review of the potential value of this data we list:

- (a) evaluation of the impact on traffic (and its consequences for the economy and the environment) of different policies;
- (b) monitoring of city pollutants;
- (c) trends and patterns in traffic. We emphasize here the interplay between the Fog and the Cloud, which must operate in mutual support.

Table1. Attribute of Smart traffic light system

Attributes of smart traffic light system	
Geo-Distribution	Wide (across region) and dense
Low /predictable latency	Tight within the scope of interaction.
Fog-Cloud interplay.	Data at different time scales(sensors/vehicles at intersection, traffic info at diverse collection points)
Multi-Agencies orchestration	Agencies that run the system must coordinate control law policies in real time.
Consistency	Getting the traffic landscape demands a degree of consistency between collection of policies.

5. **Consistency of a highly distributed system:** Visualize the TLS as a highly distributed collector of traffic data over an extended geographically data. Ensuring an acceptable degree of consistency between the different aggregator points is crucial for the implementation of efficient traffic policies.

6. **Multi-tenancy:** The Fog vision anticipates an integrated hardware infrastructure and software platform with the purpose of streamlining and making more efficient the deployment of new services and applications. To run efficiently, the Fog must support multi-tenancy. It must also provide strict service guarantees for mission critical systems such as the TLS, in contrast with softer guarantees for say, infotainment, even when run for the same provider.

7. **Multiplicity of providers:** The system of traffic lights may extend beyond the borders of a single controlling authority. The orchestration of consistent policies involving multiple agencies is a challenge unique to Fog Computing.

4. HOW IT WILL HELP TO CONTROL THE TRAFFIC?

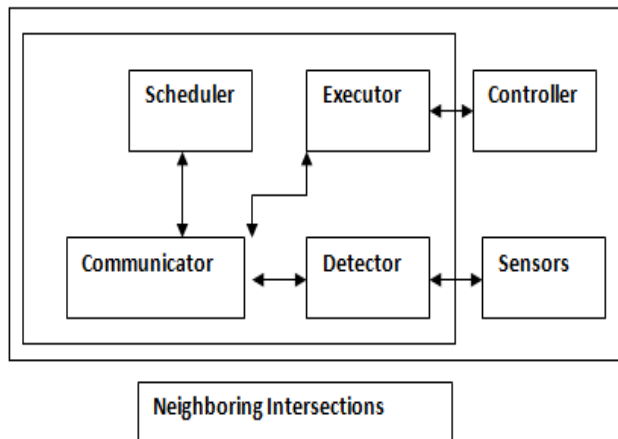


Fig 1 Block diagram of Traffic light system

This system consists of a:

- Scheduler which adjust the time.
- Executor which executed the instructions taken by the DM.
- Communicator, communicate with the neighbouring systems as well as to the decision maker.
- Detector detects the traffic.
- Sensors helps detector to detect the abnormal activity.
- Controller, controls the executor
- Decision Maker is nothing but the Local Server.

The above diagram shown is for the one signal similar systems are placed on each part of the heavy traffic area.

Working

- These systems will communicate with each other say every 15 minutes.
- The DM or the local server will communicate to the other local server servers in every 10 minutes.
- If traffic is detected in an area, the system attached to that area will communicate with the other systems with the help of communicator.
- And this is how the other systems will get information about the heavy traffic in that area.
- So the decision maker based on the data collected from the other systems will direct the coming traffic to the alternative road which will help to reduce the

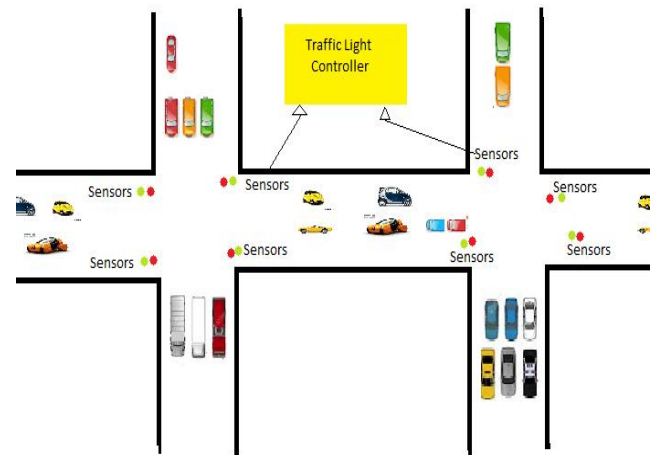


Fig 2 Block diagram of Implementation of Traffic light system

This was a small example of how the smart traffic system will help to reduce the heavy traffic in that system.

The other example is related to zebra crossing:

- The sensors will detect the number of vehicles on the zebra crossing.
- If the number of vehicles is more than the system will not allow the pedestrians to cross the zebra crossing unless there is a red signal.
- If the number of vehicles is less then it will give the red signal to them and then allow the pedestrians to cross the road.

What was the Role of FOG Computing in this Example?

- If the decision makers were on the cloud far away from the system location then it would have taken a huge time in taking the decision as well as it would cause a delay.
- Smart traffic light needs to be act in the real time.
- Therefore the Fog Computing concept resolves this issue.
- As mentioned earlier the Fog Computing benefits will help this Smart Traffic Light system to work efficiently in a real time.

5. WHAT IS THE DIFFERENCE BETWEEN THE CLOUD AND FOG?

There is "The Ground Difference between the cloud and fog

1. Tiered organization involving multiple administrations in the access.

2. Hierarchical control and management supporting interplay with the Cloud
3. Expanded mobility model
4. Real-time analytics at different tiers
5. Negligible latency
6. Geo-distribution of computational power with strong focus on service locality

7. Orchestration involving coordinated control and actuation in multi-tier settings
8. Unified exposure of virtualized resources (consolidated virtualization)
9. Distributed policy exposure and policy management involving multiple tiers.

- [4] Michael Enescu - From Cloud to Fog Computing and IoT | LinuxCon + CloudOpen North America 2014 .
<http://www.webopedia.com/TERM/F/fog-computing.html>

Table 2. Difference between Cloud & Fog Computing

Requirement	Cloud Computing	Fog Computing
Latency	High	Low
Delay Jitter	High	Very low
Location of server nodes	Within the internet	At the edge of the local network
Distance between the client and the server.	Multiple hops	One hop
Security	Undefined	Can be defined
Attack on data enroute	High probability	Very low probability
Location awareness	No	Yes
Geographical distribution.	Centralized	Distributed
Number of server nodes	Few	Very large
Support for mobility	Limited	Supported
Real time interactions	Supported	Supported
Type of last mile connectivity.	Leased line	Wireless

6. CONCLUSION

Fog Computing, is a hierarchical and distributed platform for service delivery consisting of compute, storage, and network resources. We examined key aspects of Fog computing, and how Fog complements and extends Cloud computing. We looked at use cases that motivated the need for Fog, emphasizing Fog's relevance to several verticals within IoT and Big Data space.

ACKNOWLEDGEMENTS

This work would not have been possible without the support of our colleagues, and our mentor Prof. Manish Deo. I would also like to thank my company CyberTech who helped me to understand the concept by providing the real life application based on Fog Computing.

REFERENCES

- [1] <http://www.sine-wave.com/blog/fog-computing-vs-cloud-computing-in-the-iot#.VTsW59Kqqko> (2014).
- [2] The Fog Computing Paradigm: Scenarios and Security Issue Ivan Stojmenovic SIT, Deakin University, Burwood, Australia 2013.
- [3] FOG Computing Mario Nemirovsky – ICREA/BSC 2014