# FOG-engine: Towards Big Data Analytics in the Fog

Farhad Mehdipour
Tech Futures Lab &
Unitec Institute of Technology,
Auckland, New Zealand
fmehdipour@unitec.ac.nz

Bahman Javadi
Western Sydney University,
Sydney, Australia
b.javadi@westernsydney.edu.au

Aniket Mahanti
University of Auckland,
Auckland, New Zealand
a.mahanti@auckland.ac.nz

***Abstract*— Existing platforms fall short in providing effective solutions for big data analytics while the demands for processing large quantities of data in real-time are increasing. Moving data analytics towards where the data is generated and stored could be a solution for addressing this issue. In this paper, we propose a solution referred as FOG-engine, which is integrated into IoTs near the ground and facilitates data analytics before offloading large amounts of data to a central location. In this work, we introduce a model for data analytic using FOG-engines and discuss our plan for evaluating its efficacy in terms of several performance metrics such as processing speed, network bandwidth, and data transfer size.**

***Keywords*— Cloud computing; Fog computing; Internet of Things; Big data; Data analytics***

## I. INTRODUCTION

Internets of Things (IoTs) deployments generate large quantities of data that need to be processed and analyzed in real time. Current IoT systems do not enable low-latency and high-speed processing of data, hence offloading data processing to the cloud (e.g., smart grid, oil facilities, supply chain logistics, and flood warning). The cloud allows access to information and computer resources from anywhere and facilitates virtual centralization of application, computing, and data. Although cloud computing optimizes resource utilization, it does not provide an effective solution for hosting big data applications [1].

Moving large amounts of data over the nodes of a virtualized computing platform may incur significant overhead in terms of time, throughput, energy consumption, and cost. There are several other technical issues which hinder adopting IoT-driven services, namely:

- The cloud may be physically located in a distant datacenter, so it may not be possible to service IoTs with reasonable latency and throughput.
- The current IoT development platforms are vertically fragmented. Thus, IoT innovators have to navigate between heterogeneous hardware and software services that do not always integrate well together.
- Processing large quantities of IoT data in real time will increase as a proportion of workloads in data centers, leaving providers facing new security, capacity, and analytics challenges.
- Incapability of current cloud for accommodating analytic engines for huge amounts of data (‘Big Data’) in efficient ways is a critical issue.

To address these challenges data analytics could be performed at the network edge - near where the data is generated - to reduce the amount of data and communications overhead [2]. Deciding what to save and what to use is as important as having the facility to capture the data. Rather than sending all data to a central computing facility such as the cloud, analytics at the edge of physical world where the IoTs and data reside introduces a middle layer between the ground and the cloud. The main question is that which data needs to be collected, which data needs to be cleaned and aggregated, and which data needs to be used for analytics and decision making.

We propose a solution that addresses the above challenges through:

- On-premise and real-time preprocessing and analytics of data near where it is generated,
- Facilitating collaboration and proximity interaction between IoT devices in a distributed and dynamic manner.

Through the proposed solution called the FOG-engine, IoT devices are deployed in a Fog closer to the ground that can have a beneficial interplay with the cloud and with each other. A user can use own IoT device(s) equipped with our FOG-engine to easily become a part of a smart system. Depending on the scale of user groups, several FOG-engines can interplay and share data with peers (e.g. via WiFi) and offload data on the associated cloud (via the Internet) in an orchestrated manner.

The rest of the paper is organized as follows. Section 2 describes the flow of processes in big data analytics. Section 3 describes how our proposed FOG-engine can be deployed in the traditional centralized data analytics platform and how it enhances existing system capabilities. Section 4 explains the system prototype and our plan for preliminary evaluation of the proposed solution. Section 5 concludes the paper.

## II. BIG DATA ANALYTICS

Big data refers to the large amounts of unstructured, semi-structured or structured data that flows continuously through and around organizations, including video, text, sensor data, and transactional records [3]. Big data processing can be performed either in batch mode or streamline mode. This means for some applications data is analyzed and the result is generated on a store-and-process paradigm basis [15]. Many time-critical applications generate data continuously and expect the processed outcome on a real-time basis such as stock market data processing.

Big data analytics describes the process of performing complex analytical tasks on data that typically includes grouping, aggregation, or iterative processes. Fig. 1 shows a typical flow for big data processing [3]. The first step is to perform collection/integration of the data coming from multiple sources. Data cleaning is the next step that may consume large processing time, although it may significantly reduce the data size that leads to less time and effort needed for data analytics. The raw data is normally unstructured that neither has a predefined data model nor is organized in a predefined manner. Thus, the data is transformed to semi-structured or structured data in the next step of the flow. Data cleaning deals with detecting and removing errors and inconsistencies from data to improve its quality [4]. When multiple data sources need to be integrated (e.g., in data warehouses), the need for data cleaning significantly increases. This is because the sources often contain redundant data in different representations.

One of the most important steps in any data processing task is to verify that data values are correct or, at the very least, conform to a set of rules. Data quality problems exist due to incorrect data entry, missing information or other invalid data. For example, a variable such as gender would be expected to have only two values (M or F), or a variable representing heart rate would be expected to be within reasonable range. A traditional ETL (Extract, Load and Transform) process extracts data from multiple sources, then cleanses, formats, and loads it into a data warehouse for analysis [5]. A rule-based model determines how data is handled by the data analytic tools.

A major phase of big data processing is to perform discovery of data, which is where the complexity of processing data lies. A unique characteristic of big data is the manner in which the value is discovered. It differs from conventional business intelligence, where the simple summing of known value generates a result. The data analytics is performed through visualizations, interactive knowledge-based queries, or machine learning algorithms that can discover knowledge [6]. Due to the heterogeneous nature of the data, there may not be a single solution for the data analytics problem so, the algorithm may be short-lived.
The increase in the volume of data raises several issues for analytic tools: (1) The amount of data is increasing continuously at a high speed, yet data should be up-to-date for analytics, (2) The response time of a query grows with the amount of data, whereas the analysis tasks need to produce query results on large data sets in reasonable amount of time [7].

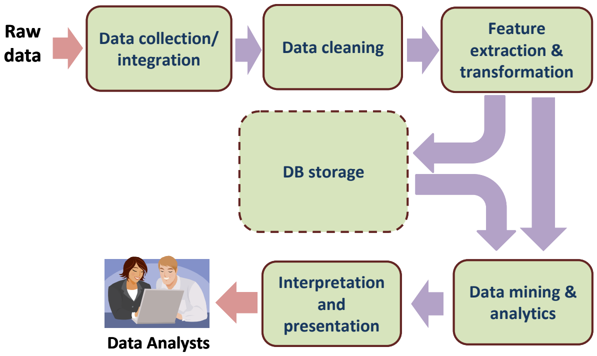

Raw data
Data collection/ integration
Data cleaning
Feature extraction & transformation
DB storage
Data mining & analytics
Interpretation and presentation
Data Analysts

Fig. 1. Typical data analytics flow

## III. Data Analytics in the Fog

While data size is growing very fast, decreasing the processing and storage costs and increasing network bandwidth make archiving the collected data viable for organizations. Instead of sending all data to the cloud, an edge device or software solution may perform a preliminary analysis and send a summary of the data (or metadata) to the cloud. For example, Google uses cloud-computing to categorize photos for its Google Photos app. For a picture taken and uploaded to Google Photos, the app automatically learns and classifies with respect to the photo's context. A dedicated chip referred to as Movidius with the capability of machine learning on the mobile devices, allows processing the information in real time, instead of in the cloud [8]. It is critical to decide what should be done near the ground, at the cloud, and in-between.

### *A. Fog computing*

The Fog extends the cloud computing paradigm to the edge of the network, thus enabling a new breed of applications and services [1]. It is a highly virtualized platform that provides compute, storage, and networking services between end devices and traditional cloud computing datacenters, typically, but not exclusively located at the edge of the network. Note that the Fog is neither the replacement of the cloud nor cannibalize the cloud. Fog computing brings about multiple benefits such as low latency, location awareness, widespread geographical distribution, mobility support, the predominant role of wireless access, the strong presence of streaming and real-time applications, heterogeneity, the orchestration of large-scale control systems, hierarchical networking, and computing structures. In general, it is an appropriate solution for the applications and services that fold under the umbrella of the IoTs [9-11].

### *B. FOG-engine*

We introduce an end-to-end solution called the FOG-engine that provides on-premise data analytic as well as the capabilities for IoT devices to communicate with each other and with the cloud. Fig. 2 provides an overview of a typical FOG-engine deployment. The FOG-engine is transparently used and managed by the end user and provides the capability of on-premise and real-time data analytics.

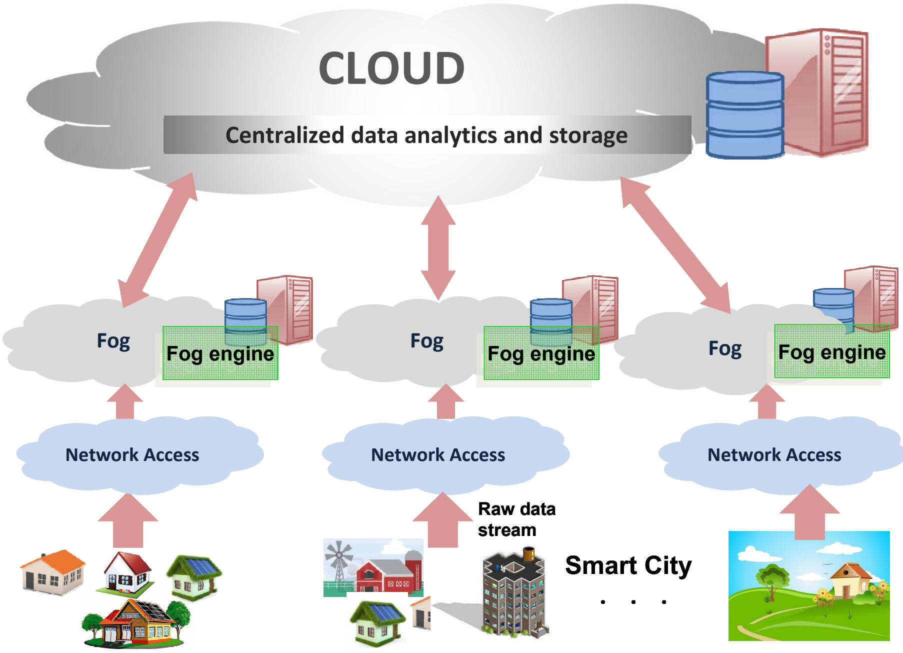


Fig. 2. Deployment of FOG-engine in a typical cloud-based computing system

FOG-engine is a customizable and agile heterogeneous platform that is integrated to an IoT device. The FOG-engine allows data processing in the cloud and in the distributed grid of connected IoT devices located at the network edge.

It collaborates with other FOG-engines in the vicinity, thereby constructing a local peer-to-peer network beneath the cloud. It provides facilities for offloading data and interacting with the cloud as a gateway. A gateway enables devices that are not directly connected to the Internet to reach cloud services. Although the term gateway has a specific function in networking, it is also used to describe a class of device that processes data on behalf of a group or cluster of devices. FOG-engine consists of modular APIs for supplying the above functionalities. Software-wise, all FOG-engines utilize the same API, also available in the cloud to ensure vertical continuity for IoT developers.

### *C. Data analytics using FOG-engines*

Fig. 3 shows on-premise data analytics being performed near the data source using FOG-engines before the data volume expands significantly. In-stream data is analyzed locally in FOG-engines while data of FOG-engines is collected and transmitted to the cloud for offline global data analytics. In a smart grid, for example, a FOG-engine can help a user decide on efficient use of energy. Whereas, the data of a town with thousands of electricity consumers is analyzed on the cloud of an energy supplier company in order to decide policies for energy use by the consumers. The analytics models employed in FOG-engines are updated based on the policies decided and communicated by the cloud analytics.

As the data preprocessed, filtered and cleaned in the FOG-engine prior to offloading to the cloud, the amount of transmitted data is most likely lower than the data generated by IoTs. Also, the analytics on FOG-engine is real-time while the analytics on the cloud is offline. FOG-engine provides a very limited computing power and storage compared with cloud, however processing on the cloud incurs higher latency. The FOG-engine offers a high level of fault tolerance as the tasks can be transferred to the other FOG-engines in the vicinity in the event of failure.

FOG-engine may employ various types of hardware such as multi-core processor, FPGA or GPU with fine granularity versus a cluster of similar nodes in the cloud. Each FOG-engine employs fixed hardware resources that can be configured by the user, whereas the allocated resources are intangible and out of user's control in the cloud. An advantage of FOG-engine is the capability of integration to mobile IoT nodes such as cars in an intelligent transportation system (ITS). In this case, multiple FOG-engines in close proximity dynamically build a Fog in which FOG-engines communicate and exchange data. Cloud offers a proven model of pay-as-you-go while FOG-engine is a property of user. Depending on the IoT application, in the case of a limited access to power sources, FOG-engine may be battery-powered which needs to be energy-efficient while cloud is supplied with a constant source of power. Table I compares FOG-engines with cloud computing.

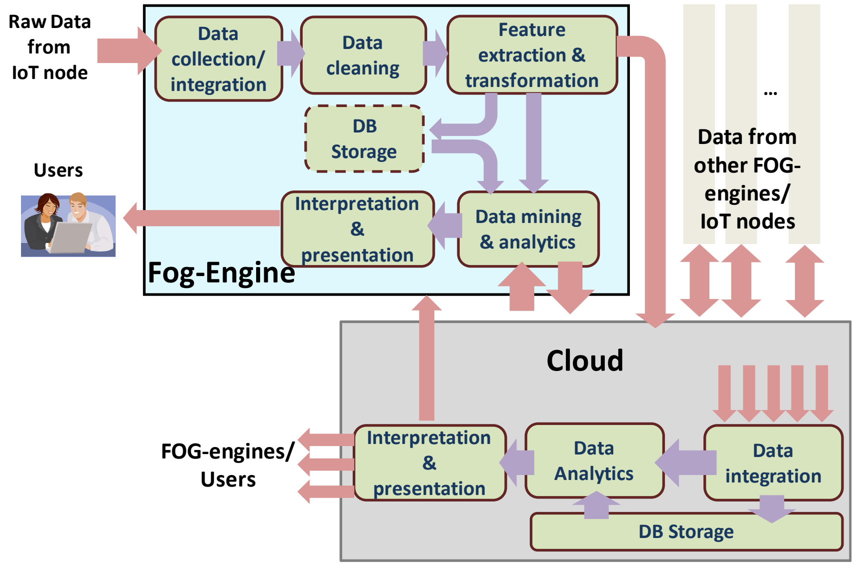

Raw Data from IoT node
Data collection/ integration
Data cleaning
Feature extraction & transformation
DB Storage
Users
Interpretation & presentation
Data mining & analytics
Fog-Engine
...
Data from other FOG-engines/ IoT nodes
Cloud
FOG-engines/ Users
Interpretation & presentation
Data Analytics
Data integration
DB Storage

Fig. 3. Data analytics using FOG-engine before offloading to the Cloud

### D. *Challenges and Issues*

There are few challenges against adopting FOG-engines in big data analytics that should be considered. The benefits of the proposed solution ought to be weighed against its costs and risks, which will vary from one use-case to another. Although sensors and IoT devices are normally cheap, the solution involving FOG-engines could be expensive if a large number of them over a wide area are involved. Thus, further research is required with respect to the scalability and the cost of the solution.

Security is another issue, as adding Fog as a new technology layer introduces another potential point of vulnerability. In addition, data management may need adjustments to address privacy concerns. Therefore, FOG-engines should be part of a holistic data strategy so there are clear answers to fundamental questions such as what data can be collected, and how long the data should be retained

Although FOG-engines can be configured as redundant resources, reliability still is an important issue where we have failures is different components of the system. Given that FOG-engine might be adapted for different applications, reliability mechanism should be changed based on the application requirements. As mentioned in Table I, FOG-engine can be battery operated so energy optimization will be a big challenge to optimize the lifetime and performance of the system. Executing data analytics in FOG-engine is a power consuming task, so energy efficiency must be implemented specially when there is large number of them deployed.

Last but not least, resource management is a challenging task for FOG-engine. Resource manager should be hierarchical and distributed where the first level of big data analytics conducted in the FOG-engine and the rest will be done in cloud. Therefore, provisioning of resources in FOG-engines for big data analytics with requested performance and cost will be a trade-off to solve by resource manager.

## IV. FOG-ENGINE PROTOTYPE AND EVALUATION

As the FOG-engine is integrated with the IoTs which mainly employ low-end devices, we need to ascertain that (a) it is agile and transparent (b) adding FOG-engine up to the IoT devices has no negative impact on the existing system. The FOG-engine is composed of three units, (i) an analytics and storage unit, (ii) a networking and communication unit and (iii) an orchestrating unit. 0(a) shows a general architecture of the FOG-engine.

Analytics and storage unit is responsible for preprocessing data (i.e. cleaning, filtering, etc.), data analytics as well as data storage. The communication unit consists of the network interfaces for peer-to-peer networking and communications to the cloud and the IoTs on the ground. Synchronizing FOG-engines with peers and the cloud is a functionality that is incorporated into every engine. This mechanism is able to orchestrate a network of distributed FOG-engines in their various forms of communications and keep them synchronized with an associated cloud as well. Moreover, for the IoT devices in the vicinity, FOG-engines may form a cluster and communicate to the cloud through a cluster head.

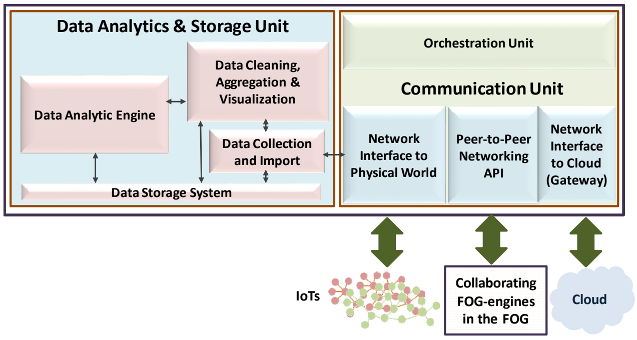
Fig. 4(b) shows an elaborated architecture for the FOG-engine. It uses a number of common interfaces for acquiring data through USB (Universal serial bus), WiFi for mid-range and Bluetooth for small-range communication with other devices, UART (Universal asynchronous receiver/transmitter), SPI (Serial peripheral interface bus) and GPIO (General-purpose input/output pins). The data may be obtained from sensor devices, other IoT devices, web or local storage. The raw or semi-structured data goes through preprocessing units such as cleaning, filtering, integration as well as ETL. A library keeps the rules which are used for data manipulation. For example, for the data generated by a smart meter on the energy consumption of a house, only positive values less than a few Kilo-Watts per hour are acceptable. The preprocessed data can be transmitted or interchanged with a peer engine via peer-to-peer networking interface unit. In a cluster of FOG-engines, one with higher processing capacity may act as a cluster head where other FOG-engines offload the data on. The orchestrating unit handles cluster formation and data distribution across a cluster of FOG-engines. The cloud interface module is a gateway that facilitates communication between the FOG-engine and cloud. All the above mentioned units are moderated by the FOG-engine scheduler and task manager.

Table I DATA ANALYTICS USING FOG-ENGINE AND THE CLOUD

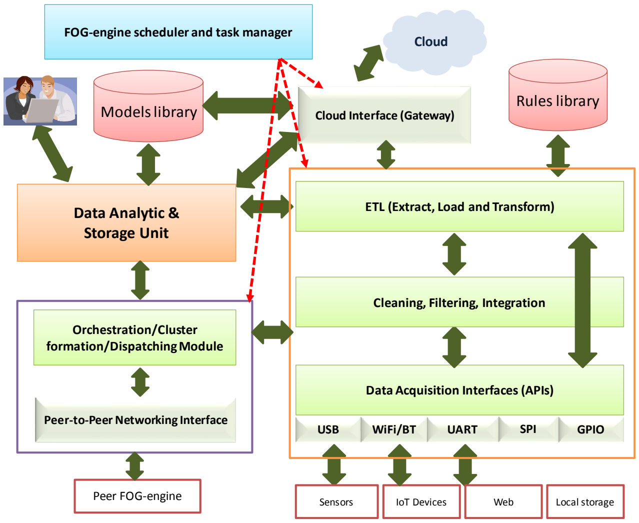
| Characteristic | FOG-engine | Cloud platform |
|---|---|---|
| Processing hierarchy | Local data analytics | Global data analytics |
| Processing fashion | In-stream processing | Batch processing |
| Computing power | GFLOPS | TFLOPS |
| Network Latency | Miliseconds | Seconds |
| Data storage | Gigabytes | Infinite |
| Data lifetime | Hours/Days | Infinite |
| Fault-tolerance | High | High |
| Processing resources and granularity | Heterogeneous (e.g.: CPU, FPGA, GPU) and Fine-grained | Homogeneous (Data center) and Coarse-grained |
| Versatility | Only exists on demand | Intangible servers |
| Provisioning | Limited by the number of FOG-engines in the vicinity | Infinite, with latency |
| Mobility of nodes | May be mobile (e.g. in the car) | None |
| Cost Model | Pay once | Pay-as-you-go |
| Power model | Battery-powered/Electricity | Electricity |

We examine the functionality of a FOG-engine in acquiring data, and interactions with peers, the cloud, and the user for various scenarios with respect to the throughput, latency, energy consumption, usage of network bandwidth, and the number of major transactions performed by the FOG-engines.

- Multiple receivers, multiple analyzers, and multiple transmitters scenario: In this scenario, each FOG-engine receives, analyses data and generates output.
- Multiple receivers, multiple analysers, and single transmitter scenario: Multiple FOG-engines receive and analyse data individually, but their data is transmitted to the cloud via one of them which acts as a cluster head.
- Multiple receivers, single analyser, and single transmitter scenario: Data is acquired by all engines and shared with one of them which has higher computing capacity. The results are sent back to the associated engines though only the analyser transmits entire data to the cloud.


Data Analytics & Storage Unit
Data Cleaning, Aggregation & Visualization
Data Analytic Engine
Data Collection and Import
Data Storage System
Orchestration Unit
Communication Unit
Network Interface to Physical World
Peer-to-Peer Networking API
Network Interface to Cloud (Gateway)
IoTs
Collaborating FOG-engines in the FOG
Cloud

(a)


FOG-engine scheduler and task manager
Cloud
Models library
Cloud Interface (Gateway)
Rules library
Data Analytic & Storage Unit
ETL (Extract, Load and Transform)
Cleaning, Filtering, Integration
Orchestration/Cluster formation/Dispatching Module
Peer-to-Peer Networking Interface
Data Acquisition Interfaces (APIs)
USB
WiFi/BT
UART
SPI
GPIO
Peer FOG-engine
Sensors
IoT Devices
Web
Local storage

USB: Universal serial bus
BT: Bluetooth
UART: Universal Asynchronous Receiver/Transmitter
SPI: Serial Peripheral Interface Bus
GPIO: General-purpose input/output pins

(b)

Fig. 4. (a) General architecture of FOG-engine (b) A detailed architecture for the communication unit

We will examine how this solution can partially undertake the burden on the network backbone and data analytics at utilities side, and reduce dependency on the cloud. Although computations are done locally, only a fraction of data that is cleaned and analyzed by FOG-engine is transferred to the cloud. Thus, we expect reduced amount of data transfers over the existing broadband network, which reduces network congestion and delays. The efficacy of FOG-engine will be demonstrated based on several factors such as the impact on the communication infrastructure in terms of cost, energy consumption, network vulnerability, transactions frequency, latency, and traffic volume. Finally, the opportunities for extending the solution in terms of size (scalability) and functionality to offer additional features for the applications with highly dynamic nature such as ITS will be explored.

Table II LIST OF IOT SOLUTIONS FROM FIVE MAJOR CLOUD PROVIDERS

| | **AWS** | **Microsoft** | **IBM** | **Google** | **Alibaba** |
|---|---|---|---|---|---|
| **Service** | AWS IoT | Azure IoT Hub | IBM Watson IoT | Google IoT | AliCloud IoT |
| **Data Collection** | HTTP, WebSockets, MQTT | HTTP, AMQP, MQTT and custom protocols (using protocol gateway project) | MQTT, HTTP | HTTP | HTTP |
| **Security** | Link Encryption (TLS), Authentication (SigV4, X.509) | Link Encryption (TLS), Authentication (Per-device with SAS token) | Link Encryption (TLS), Authentication (IBM Cloud SSO), Identity management (LDAP) | Link Encryption (TLS) | Link Encryption (TLS) |
| **Integration** | REST APIs | REST APIs | REST and Real-time APIs | REST APIs, gRPC | REST APIs |
| **Data Analytics** | Amazon Machine Learning model (Amazon QuickSight) | Stream Analytics, Machine Learning | IBM Bluemix Data Analytics | Cloud Dataflow, BigQuery, Datalab, Dataproc | MaxCompute |
| **Gateway Architecture** | Device Gateway (in Cloud) | Azure IoT Gateway (on-premises gateway, beta version) | General Gateway | General Gateway (on-premises) | Cloud Gateway (in Cloud) |

## V. RELATED WORK

Recently, major cloud providers are introducing new services for IoT solutions with different features and characteristics. Microsoft Azure Stack [12] is a new hybrid cloud platform product that enables organizations to deliver Azure services from own datacenter while maintaining control of datacenter for hybrid cloud agility. CardioLog Analytics[1] offers on-premise data analytics that run on user side servers. Oracle [13] delivers Oracle infrastructure as a service on premise with capacity on demand that enables customers to deploy Oracle engineered systems in their data centers. IBM Digital Analytics for on premises is the core, web analytics software component of its digital analytics accelerator solution. However, the analytic software is installed on high-performance IBM application servers. IBM PureData system for analytics is a data warehouse appliance which is powered by Netezza technology [14].

Cisco ParStream[2] specifically engineered to enable the immediate and continuous analysis of real-time data as it's being loaded. Cisco ParStream:

- features highly scalable, distributed hybrid database architecture to analyze billions of records at the edge,
- has patented indexing and compression capabilities that minimize performance degradation and process data in real time,
- integrated with R, Knime, and other ML engines to support advanced analytics, makes effective use of both standard multicore CPUs and GPUs to execute queries,

uses time-series analytics to combine analyzing streaming data with massive amounts of historical data,

- Uses alerts and actions to monitor data streams, create and qualify easy-to-invoke procedures that generate alerts, send notifications, or execute actions automatically
- Derives models and hypotheses from huge amounts of data by applying statistical functions and analytical models using advanced analytics

Although these solutions offer on-premise data analytic services, they lack in providing a holistic approach based on the fog concept with an intermediate layer between the IoTs and the cloud.

Table II shows the list of IoT solutions from five famous cloud providers. Data collection is one of the basic aspects for these solutions, which specify the communication protocols between the components of an IoT software platform. Since IoT systems might have millions of nodes, lightweight communication protocols such as MQTT and AMQP have been provided to minimize the network bandwidth. Security is another factor in these solutions where a secure communication is needed between IoT devices and the software system. As one can see in this table, link encryption is a common technique to avoid potential eavesdropping in the system. Integration is the process of importing data to the cloud computing systems and as mentioned in Table II, REST API is a common technique to provide access to the data and information from cloud platforms. After collecting data from IoT devices, data needs to be analysed to extract knowledge and meaningful insights. Data analytics can be done in several ways and each cloud provider has various packages and services including machine learning algorithms, statistical analysis, data exploration and visualizations.

The last row in Table II, is the gateway architecture which is the main scope of this paper. The gateway is the layer between IoT devices and cloud platform. Most of providers only provide a general assumptions and specifications about the gateway which will be located within the cloud platform. There are some early

[1] Intlock, http://news.intlock.com/on-premise-or-on-demand-solutions/

2 https://www.parstream.com/

stage developments for on-premises gateway from Microsoft and Google, but none of them has implemented that completely with appropriate integration. As mentioned earlier, FOG-engine can be a solution as the gateway that provides on-premise data analytic as well as the capabilities for IoT devices to communicate with each other and with the cloud platform.

## VI. Conclusions

The data analytics can be performed near where the data is generated to reduce the data communications overhead as well as data processing time. Through our proposed solution, FOG-engine, it is possible to enable IoTs with the capability of on-premise processing which results in multiple advantages such as lower latency, higher throughput, and less usage of network bandwidth. However, precise evaluations on various aspects of the efficiency are essential to be able to demonstrate the benefits of the solution over the current cloud-based solutions. Also, there are some challenges with respect to the extensibility, cost and security that require further investigations.

## References

[1] A.V. Dastjerdi, H. Gupta, R.N. Calheiros, S. K. Ghosh, and R. Buyya, Fog Computing: Principles, Architectures, and Applications, Book Chapter in Internet of Things: Principles and Paradigms, Morgan Kaufmann, Burlington, Massachusetts, USA, 2016.

[2] Satyanarayanan, Mahadev, et al., Edge analytics in the internet of things, Pervasive Computing, IEEE 14.2 (2015): 24-31.

[3] F. Mehdipour, H. Noori, B. Javadi, Energy-Efficient Big Data Analytics in Datacenters, Advances in Computers. Vol. 100, 2016, 59-101.

[4] E. Rahm, H. Hai Do, Data Cleaning: Problems and Current Approaches, IEEE Data Eng. Bull. 23.4, 3-13, 2000.

[5] Intel Big Data Analytics White Paper, Extract, Transform and Load Big Data with Apache Hadoop, 2013.

[6] B. Di-Martino, R. Aversa, G. Cretella, A. Esposito, Big data (lost) in the cloud, Int. J. Big Data Intelligence 1 (1/2) (2014) 3–17.

[7] M. Saecker, V. Markl, Big data analytics on modern hardware architectures: a technology survey, business intelligence, Lect. Notes Bus. Inf. Process. 138 (2013) 125–149.

[8] D. Schatsky, Machine learning is going mobile, Deloitte University Press, 2016.

[9] F. Bonomi, et al., Fog Computing and Its Role in the Internet of Things, MCC, Finland, 2012.

[10] A. Manzalini, A Foggy Edge, beyond the Clouds…, Business Ecosystems, Feb. 2013.

[11] T. Perry, What Comes After the Cloud? How About the Fog?, http://spectrum.ieee.org/tech-talk/computing/networks/what-comes-after-the-cloud-how-about-the-fog, 2013.

[12] J. Woolsey, Powering the Next Generation Cloud with Azure Stack, Nano Server & Windows Server 2016, Microsoft, 2016.

[13] Oracle infrastructure as a service (IaaS) private cloud with capacity on demand, Oracle executive brief, Oracle, 2015.

[14] L. Coyne, T. Hajas, M. Hallback, M. Lindström, C. Vollmar, , IBM Private, Public, and Hybrid Cloud Storage Solutions, Redpaper, IBM, 2016.

[15] B. Javadi, , B. Zhang, and M. Taufer. "Bandwidth Modeling in Large Distributed Systems for Big Data Applications." Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2014 15th International Conference on. IEEE, 2014.