A Computing Architecture For IOT (Internet of Things) Applications And Devices

MD TARIKUL ISLAM
Department Of Computer Science And Engineering,
Brac University
Mohakhali, Dhaka - 1212, Bangladesh
md.tarikul.islam@g.bracu.ac.bd

ABSTRACT

The rapid growth of the Internet of Things (IoT) applications and their interference with our daily life tasks have led to a large number of IoT devices and enormous sizes of IoT-generated data.As the Internet of Things edges closer to mainstream adoption, with it comes an exponential rise in data transmission across the current Internet architecture. Capturing and analyzing this data will lead to a wealth of opportunities. However, this ungoverned, unstructured data has the potential to exhaust the resources of an already strained infrastructure. Analyzing data as close to the sources as possible would greatly enhance the success of the IoT. This paper proposes a distributed data processing architecture for edge devices in an IoT environment. Our approach focuses on a vehicular trucking use case. The goal is to recreate the traditionally centralized Storm processes on the edge devices using a combination of Apache MiNiFi and the user's custom built programs. Our approach is shown to preserve computational accuracy while reducing by upwards of 90 percent the volume of data transferred from edge devices for centralized processing.

1 INTRODUCTION

The "Internet of Things" is a paradigm in which objects such as household appliances, automobiles and even humans will be assigned IP addresses.As the enterprises store large volumes of unfiltered data in data centre,processing and filtering have been an obvious priority for worlds leading technology companies.Such centralized data storage limits organizations as they are unable to capitalize on timely insights.To tacle this problem we propose a distributed architecture which employed data filtering alogorithm, Hortonworks Data Flow (HDF) analytics platform and apache minify to evaluate a vehicular trucking scenario.

2 LITERATURE SURVEY

EdgeComputing and Fog Computing are new paradigms where data processing is executed on the edge of networks to mitigate cloud server load. However, EdgeComputing and Fog Computing still need powerful servers on the edge of networks. Here [6] proposes a platform called IFoT (Information Flow of Things).IFoT middleware which processes video streams in real- time and in a distributed manner by using computational resources of IoT devices. Krikkit is publish/subscribe mechanism where rules are registered on edge gateways that have visibility into and communicate with sensors.

3 ARCHITECTURAL IMPLEMENTATION

We evaluate a vehicular trucking scenario in which HDF analytics platform and Apache Minifi are used.Truck generates normal events include vehicle starting, vehicle stopping etc.Violation events include speeding, excessive accelaration, excessive braking and unsafe tail distance.The HDF analytics platform streams these events to storm.Storm filters violations and performs real-time analysis,detecting erratic behavior for a driver over a short period.If a driver creates five violations in a three-minute window, an alert is sent directly to the fleet manager immediately with a file containing a list of violations occured.
In HDF use case, NiFi ingests all data from edge devices and separates the data into two dataflows.In figure 1,

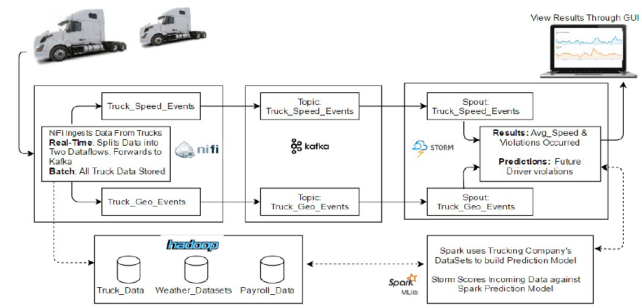

Figure 1: HDF Architecture with NiFi and Kafka Feeding Data into Storm Bolts for Processing. Apache Spark Creates Prediction Model

4 EDGE DATA PROCESSING

The Dataflow 1 calculates the average speed.Dataflow 2 enriches each unsafe event with weather attributes and prepares it for the Spark ML prediction model.It uses a RouteText processor to filter out all unsafe events which are then forwarded to the main Nifi server.Nifi server ingests and seperates incoming data from edge devices. Dataflow 2 is identical in scenario 1

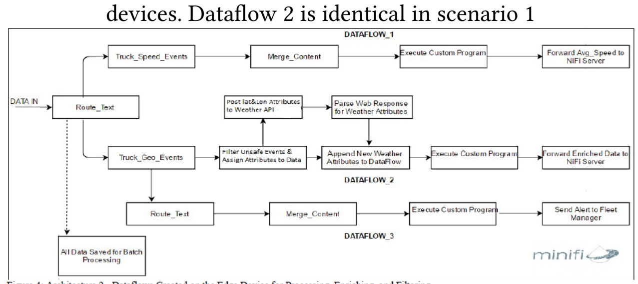

Figure 4: Architecture 2 - Dataflows Created on the Edge Device for Processing, Enriching, and Filtering

5 POTENTIAL CHALLENGES

There are some challenges to implement the idea.

- Computation and resource constraints.
- Data backup.
- Remote and resource management.
- Linguistic Privacy, cyber and physical security.
- Trust and authentication concerns.
- Reliability and fault tolerance.
- Scheduling and load balancing.

6 CONCLUSION

This enhanced distributed data processing architecture reduces significantly the need for centralized data processing while preserving computational accuracy.This architecture reduces data transmission by up to 98 percent.Future work combining MiniFi, NiFi and Apache Spark MLlib will enable us to utilize machine learning mechanism as the data is created.
[1–6].

REFERENCES

[1] 2017. *eclipse.org/krikkit/,*.

[2] B. Cheng A. Papageorgiou and E. Kovacs. 2015. *Real-time data reduction at the network edge of Internet-of-Things systems.* Tennessee Technological University. Retrieved February 28, 2015 from http://math.tntech.edu/rafal/cliff11/index.html

[3] C. Barbieru and F. Pop. 2016. *Soft Real-Time Hadoop Scheduler for Big Data Processing in Smart Cities.* Tennessee Technological University. Retrieved February 28, 2008 from http://math.tntech.edu/rafal/cliff11/index.html

[4] R. Dssouli I. Taleb and M. A. Serhani. 2015. *Big Data Pre-processing: A Quality Framework.* Tennessee Technological University. Retrieved February 28, 2015 from http://math.tntech.edu/rafal/cliff11/index.html

[5] G. Vetticaden. 2017. *github.com/georgevetticaden/hdp.* Tennessee Technological University. Retrieved February 28, 2017 from https://github.com/georgevetticaden/hdp/tree/master/referenceapps/iot-trucking-app

[6] Y. Arakawa H. Yamaguchi Y. Nakamura, H. Suwa and K. Yasumoto. 2016. *Middleware for Proximity Distributed Real-Time Processing of IoT Data Flows.* Stanford University. Retrieved March 2, 2005 from http://ccrma.stanford.edu/~jos/bayes/bayes.html