A Proposal to Address the Big Data Challenges for the Internet of Things

Tania Khalafbeigi[1], Steve Liang[2]

1 Geomatics Engineering, University of Calgary, tkhalafb@ucalgary.ca
2 Geomatics Engineering, University of Calgary, steve.liang@ucalgary.ca

Abstract

Billions to trillions of small sensors and actuators will be embedded in everyday objects and connected to the Internet forming a concept called the Internet of Things (IoT) (Evans, 2011). These internet connected sensors register their observations frequently to the IoT cloud services and the cloud service must respond to a very large number of different queries about those observations. As a result handling IoT sensor data is considered as one of the major big data challenges. In addition big data is not only about large volume, but also about variety and velocity of data. To address IoT's big data variety challenge, Open Geospatial Consortium (OGC) SensorThings standard was proposed (GeoSensorWeb Lab, 2013a). In this paper, we propose an architecture to address the volume, variety and velocity challenges in the Internet of Things.

Background and Relevance

Based on standard interfaces for sensors in IoT, like the OGC SensorThings application programming interface (API) standard (GeoSensorWeb Lab, 2013a), a coherent service can be developed to serve a wide variety of sensors all over the world , like the prototype service developed in (GeoSensorWeb Lab, 2013a; Khalafbegi & Liang, 2014a). Such service needs to handle a very high throughput rate (*i.e.*, a very large number of requests per second), since those many sensors will frequently register their observations and also many requests will be sent from applications using these sensors data for different applications. Moreover, the service should be able to scale with the growing amount of data it stores without sacrificing performances. Thus, handling IoT sensor data is considered as a classic big data problem.

Volume, Velocity and Variety are called three "V"s of big data (Russom, 2011). Big data volume in IoT is about managing a very large amount of sensors and actuators data. Very soon billions to trillions of small sensors and actuators will be embedded in everyday objects and connected to the Internet forming IoT. As predicted by CISCO (Evans, 2011), the number of these internet-connected objects will reach 50 billion by 2020. Each of these sensors registers its observations frequently and we need to manage all the data from all these things. As a result, we face big data volume that we need to manage.

Big data variety in IoT is about managing data comes from different resources with different structures. For example there are different kinds of temperature sensors from different vendors and their observations needs to be organized in a consistent fashion by IoT. Each of these sensors has their own IoT service provider that develops and uses its own proprietary software interface, encodings, and ontologies. That means the number

of proprietary interfaces are growing as the number of IoT devices increases. The IoT sensor data may come from different sources with different interfaces and structures (or even unstructured). This aspect of big data variety is also a major challenge that needs to overcome.

Big data velocity in IoT means managing continuous sensor observations streams. There will be billions of sensors and each one of them registers their observation frequently. For example sound pressure sensors typically register their readings every second. Moreover, IoT cloud service needs to answer the queries about these data streams that flow into the service. As a result, managing data streams and responding to real-time queries are another major challenge.

To address big data variety we propose to use the OGC SensorThings API (GeoSensorWeb Lab, 2013a). OGC SensorThings API is an open standard-based interoperable Web Application Programming Interface (API) that allows all IoT sensing devices and applications to interoperate. The goal of this API is to capture the observations and controlling capabilities from IoT devices and makes them easily available through data aggregation portals (e.g., cloud services for IoT devices) (Khalafbegi & Liang, 2014a).

In this paper, we focus on addressing the big data volume and velocity challenges. The objectives of this research is to design an architecture for global IoT devices that can efficiently store and retrieve large amount of sensors and actuators data and also can handle large amount of real-time queries with different types in a very efficient manner. The proposed architecture scales with the growing amount of data without sacrificing performance.

Methods and Data

To achieve the research goals, we propose an architecture based on the Lambda Architecture (Marz & Warran, 2013) in order to implement the SensorThings services.

Lambda Architecture is a generic, scalable and fault-tolerant data processing architecture that can be used for a real time big data processing. It contains three layers. Batch layer manages the master datasets and create batch views. Serving layer indexes the batch views and prepares them for querying. And Speed Layer deals with real-time data processing and query answering. (Khalafbegi & Liang, 2014b)

The bath layer responsibilities are: 1) storing an immutable, constantly growing master dataset, and 2) computing arbitrary functions on that dataset and creating batch views. The serving layer indexes these pre-computed batch views so that it can be efficiently queried. In other words, serving layer makes batch views query-able. The serving layer continuously swaps in new versions of a batch view that are periodically computed by the batch layer. Since the batch layer processing takes at least a few hours, the serving layer is updated at most every few hours. (Marz & Warran, 2013)

Batch layer together with serving layer is used for addressing the big data volume challenge. Different kinds of technology can be used to organize the master dataset and

batch views which contains large amount of data from sensors and actuators. Apache Hadoop (Hadoop, 2014; HBase, 2014) is the canonical example of a batch processing system (Marz & Warran, 2013).

The serving layer updates whenever the batch layer finishes precomputing batch views. As a result, data that came during batch views precomputation are not represented in batch views. If the service answers the queries only based on service layer, the response does not contain real time data. To overcome this problem, Lambda architecture has a fully real-time data system – that is, arbitrary functions computed on arbitrary data in real-time – named speed layer.

One of the strengths of the Lambda Architecture is that once data makes it through the batch views and is loaded into the serving layer, the corresponding results in the real-time views will be removed from speed layer. This means *no longer needed* real-time view is discarded from speed layer frequently. It makes the architecture more fault-tolerant, since the speed layer is much more complex than the batch and serving layers and the probability of fault is more in that. However, even if a fault occurs in speed layer, very soon it will be replace with correct information in batch views.

To address big data velocity challenge, speed layer is used to provide real-time sensor and actuator data processing and query answering.

Results

We propose using Lambda Architecture to implement SensorThings standard API. As a result, the developed system has the potential to efficiently store and retrieve large amount of sensors and actuators data and also can handle large amount of queries with different types in a very efficient manner. We will present some preliminary experimental results in the presentation.

Conclusions

To address big data challenges in Internet of Things, we propose using Lambda architecture in IoT services. Using Lambda architecture makes IoT services scalable with the growing amount of sensors and actuators data without sacrificing performance of answering queries about them. The implementation of the proposed architecture on SensorThings prototype (GeoSensorWeb Lab, 2013b) is still ongoing with Apache Hadoop (Hadoop, 2014) technologies.

For the future work, first of all different technologies can be used for implementation such as document based databases. Moreover, the architecture can be tuned to be more efficient for geospatial queries.

References

University of Calgary GeoSensorWeb Laboratory (2013a). *SensorThings API*, http://ogc-iot.github.io/ogc-iot-api/index.html , Accessed 7/22/2014

University of Calgary GeoSensorWeb Laboratory (2013b). *SensorThings Prototype Service*, http://demo.student.geocens.ca:8080/SensorThings_V1.0 , Accessed 7/22/2014

Khalafbeigi, T. & Liang, S. (2014a). *A Prototype Implementation of the Open Geospatial Consortium (OGC) SensorThings Service*, SKI Canada 2014, Banff.

Evans, D. (2011). *The Internet of Things: How the next evolution of the internet is changing everything*. CISCO white paper.

Russom, P. (2011). Big Data Analytics. *TDWI Best Practices Report, Fourth Quarter*.

Marz, N., & Warren, J. (2013). *Big Data: Principles and best practices of scalable realtime data systems*. O'Reilly Media.

Khalafbeigi, T. & Liang, S. (2014b). *A Big Data Architecture for the Open Geospatial Consortium (OGC) SensorThings API*. OGC Academic Summit, Calgary, Canada.

Apache Hadoop, http://hadoop.apache.org/ , Accessed 10/30/2014

Apache HBase, http://hbase.apache.org/ , Accessed 10/30/2014