Medición y Visualización Meteorológica con Azure-IoT

Adrian J. Fernandez[1], Ariel Aizemberg[2]

[1] ITBA, Buenos Aires, Argentina,
adrian.fernandez@microsoft.com
[2] Depto. de Ingeniería Informática, ITBA, Buenos Aires, Argentina,
aaizemberg@itba.edu.ar

Resumen Este proyecto implementa un prototipo que realiza la medición y visualización de temperatura en tiempo real utilizando tecnologías Open Hardware y Microsoft Azure. Se utilizan distintos tipos de sensores junto a un microcontrolador para capturar datos y almacenar la información en la nube, para luego analizar y visualizar los flujos de datos resultantes en tiempo real.

Keywords: Open Hardware, Microsoft Azure, Big Data, IoT

1. Descripción del trabajo

Internet of Things (IoT), se basa en objetos electrónicos físicos interconectados con otros, creando una red de dispositivos que producen información en tiempo real sobre Internet. En este contexto, los sensores están capturando eventos de nuestro entorno produciendo grandes volúmenes de información de la vida real.

En este trabajo presentamos un prototipo que utiliza tecnología Open Hardware, Microsoft Azure y utiliza sensores físicos de temperatura, humedad y luminosidad pero se podria extender otros como velocidad del viento, dirección, pluviómetro, presión, etc., conectados a dispositivos microcontroladores para capturar y enviar datos de sensores a internet para ser procesados, analizados y crear visualizaciones del flujo de datos en tiempo real, luego procesada por un clúster Hadoop a efectos de encontrar patrones de variación de datos meteorológicos.

1.1. Arquitectura

La arquitectura se presenta en la Figura 1, los sensores (*Weather Shield*) son conectados a un microcontrolador Open Hardware programable Arduino, enviando los datos a un Gateway (Raspberry Pi B+) que ejecuta una distribución Raspbian de Linux, el cual retransmite los datos a un Hub de eventos en Internet (Azure Events Hubs).

Los sensores crean un flujo constante con datos meteorológicos en formato raw los cuales son pre-procesados, formateados y enviados a un Hub de eventos de en Internet a razón de 200 a 400 mensajes por segundo por sensor.

Una vez en la nube, el flujo de datos o *data stream* se introduce en un motor de análisis en tiempo real (Azure Stream Analytics), para generar cálculos analíticos y alertas a través de todos los dispositivos. Estos flujos de datos en tiempo real y alertas se visualizan con gráficos interactivos desde un sitio de control o monitoreo, se necesita sólo un navegador moderno que soporte HTML5.

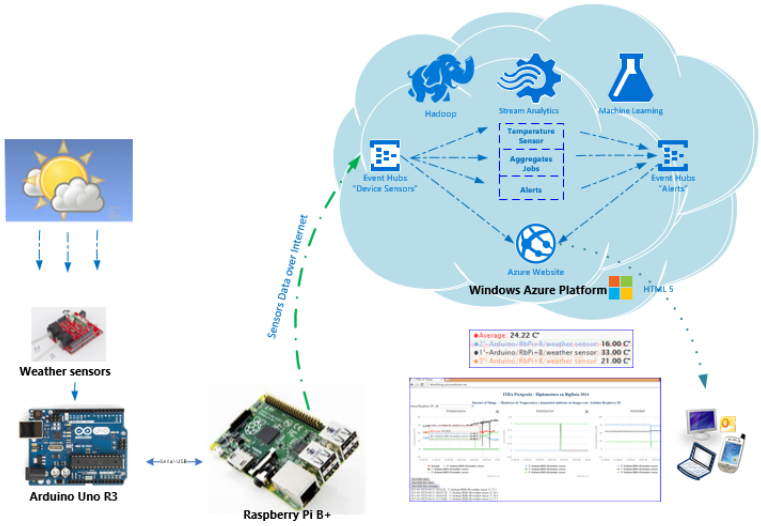

Figura 1.

1.2. Escalabilidad

El prototipo implementado transmite mensajes o ráfagas de datos que llegan de los sensores a los hubs de eventos a razón de 200 a 400 mensajes por segundo. La implementación actual podría soportar de 8000 a 12000 msg/s aproximadamente; basado en estos cálculos aproximados con la configuración actual sería posible escalar de 20 a 40 sensores similares dependiendo de las variables (medidas) que se definan a transmitir.

1.3. Visualización

Se realizaron gráficos de líneas múltiples para las principales variables: temperatura, humedad y luminosidad ambiente.

En la Figura 2, se observa la representación gráfica de los datos que ingresan en tiempo real y muestra en detalles la medición de temperatura en grados Celsius, con datos en tiempo real de cuatro dispositivos, tres de los cuales son simulados y uno es el prototipo real.

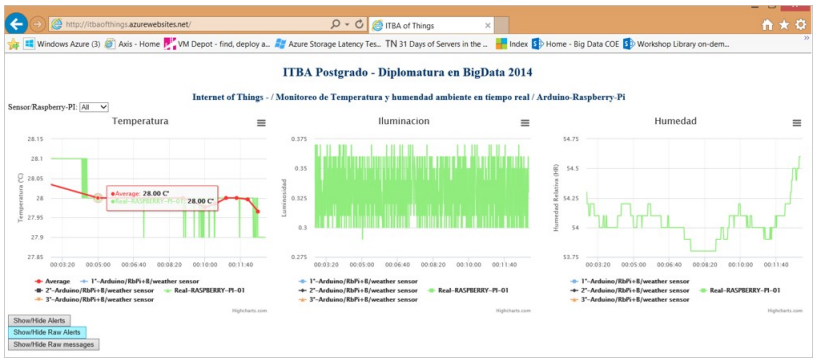

Figura 2.

La figura 3, detalla la medición de la temperatura y se incluyen los datos de tres sensores programados del tipo virtual: “1er-Arduino/RbPi+B/weather sensor”, “2do-Arduino/RbP+B/weather sensor” y “3er-Arduino/RbPi+B/weather sensor”. Los sensores simulados generan el mismo tipo de registro que el sensor real y nos permiten experimentar como escalaría el sistema con múltiples dispositivos.

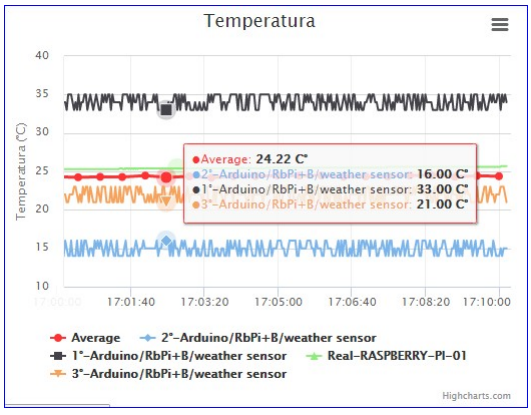

Figura 3.

El eje x en la figura3 representa el tiempo, medido en horas, minutos y segundos, con una ventana de visualización de 10 minutos, el eje Y representa la variables de los distintos sensores (Temperatura, Luminosidad y Humedad).

Para realizar una prueba de esfuerzo de las capacidades de la arquitectura propuesta, se analiza el tráfico y los eventos generados por un solo sensor o dispositivo de captura, la figura 4 muestra la carga de Hub de eventos de un dispositivo enviando 300 msg/seg.

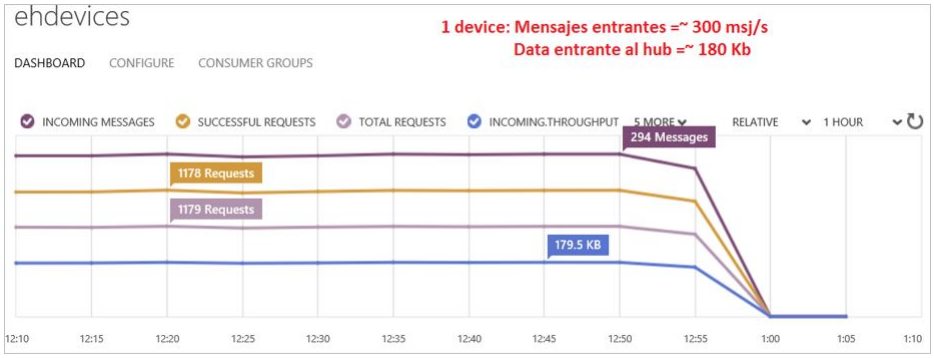

Figura 4.

2. Conclusiones y trabajo futuro

El proyecto incluyó la implementación del prototipo físico (Arduino-Raspberry) y el desarrollo de sensores virtuales (generadores de datos), con los cuales fue posible realizar pruebas avanzadas para los cálculos analíticos y las pruebas de esfuerzo y estimar la escalabilidad de la arquitectura implementada.

El prototipo desarrollado soporta el procesamiento de los datos en tiempo real utilizando Apache Storm, los datos históricos se almacenan en una tabla relacional y en simultáneo en tablas HBase y DocumentDB. Toda la solución está montada sobre Microsoft Azure.

Para el análisis avanzado de mediciones meteorológicas, se realizarán experimentos con Azure Machine Learning, por ej. *Anomaly detection on real time stream of data using Azure ML anomaly detection service*[8], detección de valores atípicos y predicción meteorológica básica.

3. Agradecimientos

Agradecemos al ITBA y a Microsoft por el apoyo brindado para poder implementar este proyecto sobre servidores en la nube de Azure.

Referencias

1. Membrey P., Hows D.: *Learn Raspberry Pi with Linux.* Editorial Apress (2013)
2. Roden T.: *Building the Realtime User Experience.* Editorial O'Reilly (2010)
3. Sarkar D.: *Pro Microsoft HDInsight Hadoop on Windows.* Editorial Apress (2012)
4. Barlow M.: *Real-Time Big Data Analytics emerging Architecture.* Editorial O'Reilly, Strata (2013)
5. *WiFi Weather Station.* `https://goo.gl/ZbOfgf`
6. *Cloud-Connected Weather Station with the Arduino Yun Temboo.* `https://goo.gl/yw1r9k`
7. *Arduino Weather Station Data Analysis.* `http://goo.gl/URCybo`
8. *Machine Learning Anomaly Detection Service* `https://goo.gl/kUg0Lj`