Unification of Real-time Analytics and IoT Data

Uzma Anam Riaz[#1], Amtul Raheem Hafsa[#2], Bader Fatima Naaz[#3], Shaik Rasool[#4]

[#]*Information Technology Department, MJCET*
Hyderabad, India.

Abstract *Today, Internet of Things is the talk of the town. It revolves around increased machine-to-machine communication; it is built on cloud computing and networks of data-gathering sensors; it is mobile, virtual, and instantaneous connection; and is supposed to make everything in our lives from streetlights to seaports "smart". In this paper the proposed IoT system is capable of processing sensor data for real-time alerts. It does so with the help of Azure IoT Hub that performs two fundamental tasks. Firstly, it collects temperature sensor data from Raspberry Pi 2 device. Secondly, the Hub notifies any device or mobile app when temperature crosses a set limit with the help of Azure Stream Analytic, Event Hub and Cloud services. This concept can be applied to any type of data; enabling this system to be generic in nature.*

Keywords — *Internet of Things, Windows, Azure, Raspberry Pi, Analytics.*

I. INTRODUCTION

The Internet of Things (IoT) refers to the ever-growing network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between these objects and other Internet-enabled devices and systems. In general, the IoT promotes a heightened level of awareness about our world, and a platform from which to monitor the reactions to the changing conditions that said awareness exposes us to [1]. From the onset of Internet of Things (IOT), analytics technologies are critically known for turning this tide of streaming source data into informative, aware and useful knowledge. The real question however remains about the analysis of data as it streams nonstop from sensors and devices and how does this process differ from other analytical methods that are common today. In traditional analysis, data is stored and then analysed. However, with streaming data, the models and algorithms are stored and the data passes through them for analysis. This type of analysis makes it possible to identify and examine patterns of interest as data is being created in real time [2].

The Internet of Things (IoT) seamlessly bridges the physical and digital worlds using smart, connected devices, like sensors and actuators, that work together to send data across smart, connected systems. These connected devices provide a constant stream of data that is unique in its volume, velocity, and variety. A new report from Forrester Research called "IoT Upsets Application Development" looks at the importance of real-time IoT analytics in application development. The app development process is no longer linear, with inputs flowing through to outputs. Instead, streaming IoT data from hundreds or thousands of connected devices requires two-way communication and real-time reporting. IoT app developers must look at how to turn this flow of contextual information from connected devices and systems into useful, actionable data.

The term – 'Internet of Things' has been around for more than 15 years, though it only began gaining wide currency more recently. Still, nowadays "What is the Internet of Things?" is not the frequently asked question of the topic. Instead, "What is the business value of the data generated by the IoT? And what do we need to do to realize that value?" is what is mostly asked. Therefore, the problem lies in how can the IOT big data be fully realized for a maximum positive impact [3].

The real-time analytics makes the important real-time decisions that are faster and more honed for IOT systems leading to greater optimization. This way, greater value for individuals and businesses are created. The real-time analytics makes the important real-time decisions that are faster and more honed for IOT systems leading to greater optimization. This way, greater value for individuals and businesses are created. With advanced analytics techniques, data stream analytics can move beyond monitoring existing conditions and evaluating thresholds to predicting future scenarios and examining complex questions [4]. To assess the future using these data streams, high-performance technologies are needed that identify patterns in your data as they occur. Once a pattern is recognized, metrics embedded into the data stream drive automatic adjustments in connected systems or initiate alerts for immediate actions and better decisions. Essentially, this means the world can move beyond monitoring conditions and thresholds to assessing likely future events and planning for countless what-if scenarios.

II. PREVIOUS WORK

"Internet of things is an emerging trend where usage of Internet extends beyond just utilizing the World Wide Web. A novel method is presented in

this work through using IOT for sensing Temperature and Humidity with the help of scripting on Raspberry Pi. The advantages of Raspberry Pi have been enlisted along with the working methodology. The results are obtained in the graphical form, which are vividly represented. At the end of the section, paper concludes with a Conclusion note. This work is in the form of Review of the existing technology of using IOT for sensing and plotting Temperature and Humidity [5].

India's population is reached beyond 1.2 billion and the population rate is increasing day by day then after 25-30 years there will be serious problem of food, so the development of agriculture is necessary. Today, the farmers are suffering from the lack of rains and scarcity of water. An automatic irrigation system saves time, money & power of the farmer. The traditional farm-land irrigation techniques require manual intervention. With the automated technology of irrigation, the human intervention can be minimized. Whenever there is a change in temperature and humidity of the surroundings these sensors sense the change in temperature and humidity and gives an interrupt signal to the micro-controller [6].

One of the topics sudden interests gaining popularity day by day is that of home because of its numerous advantages. One can achieve home automation by simply connecting home appliance and electrical devices to the internet or cloud. The reason for this surge in demand of network enabled home automation is reaching the zenith in recent days for its simplicity and comparable affordability. Platforms based on cloud computing help to connect to the things surrounding everyone so that one can find it easy to access anything and everything at any time and place in a user-friendly manner using custom defined portals. Hence, cloud acts as a front end to access IoT. In addition to that, with the never-ending growth of the Internet and its applications, there is much potential and scope for remote access and control and monitoring of such network enabled appliances. Even so, the exciting opportunities to increase the connectivity and relationship of home devices for home automation purposes to the internet are yet to be [7].

III. IMPLEMENTATION

For implementing this system, first, the PC requirements, of having the Windows 10 as the PC operating system and Visual Studio Community 2015 need to be installed along with few other prerequisites. Secondly, the Raspberry Pi2 has to be setup followed by the creation of an account to access one's Azure IoT Hub. Then, an UWP app is created for Raspberry Pi2. This application will send sensor data to IoT Hub and receive alerts from IoT Hub. Next, the filtering of IoT Hub and sending to Event Hub is accomplished by successful setup of Azure Stream Analytics. Finally, a cloud service is configured so that when temperature is high, alerts can be sent back to device through IoT Hub.

Step 1: Set up Your PC

i. First you need a Windows 10 PC running the public release of Windows 10 (version 10.0.10586) or better.
ii. Install Visual Studio Community 2015 or Visual Studio Professional 2015 or Visual Studio Enterprise 2015 with Update
   - Enable Universal Windows App Development Tools before updating.
iii. Install Windows IoT Core Project Templates (OS built for the IoT, enables development for raspberry pi 2)
iv. Enabling developer mode on windows 10 device
   - From the for-developer's settings dialog, choose the level of access that you need.
   - Read the disclaimer for the setting you chose, then click Yes to accept the change.

Step 2: Set up Raspberry Pi2

i. 5v Micro USB power supply with at least 1.0A current. If you plan on using several power-hungry USB peripherals, use a higher current power supply instead (>2.0A).
ii. 8GB Micro SD card - class 10 or better.
iii. HDMI cable and monitor.
iv. Ethernet Cable.
v. Micro SD card reader - due to an issue with most internal micro SD card readers, we suggest an external USB micro SD card reader.
vi. Download Windows 10 IOT core Dashboard and then set up a new device by providing Device name and password after that click on Download and Install.
vii. Install the Windows 10 IoT Core tools (It will automatically mount itself as a virtual drive, so you can access the contents)/
viii. When installation is complete, flash.ffu will be located at C:\ProgramFiles (x86) \Microsoft IoT\FFU\RaspberryPi2.
ix. Eject the Virtual CD when installation is complete - this can be done by navigating to the top folder of File Explorer, right clicking on the virtual drive, and selecting "Eject".
x. Insert a Micro SD Card into your SD card reader.
xi. Use IoTCoreImageHelper.exe to flash the SD card. Search for "WindowsIoT" from

start menu and select the shortcut "WindowsIoTImageHelper".

xii. After launch the IoTCoreImageHelper.exe and select your SD Card and the flash.ffu found in the directory.

xiii. Once the process has completed, you are ready to run Windows 10 IoT Core on your Raspberry Pi 2.

xiv. Safely remove your USB SD card reader by clicking on "Safely Remove Hardware" in your task tray, or by finding the USB device in File Explorer, right clicking, and choosing "Eject". Failing to do this can cause corruption of the image.

xv. Hook up your board

xvi. Insert the micro SD card you prepared into your Raspberry Pi 2.

xvii. Connect a network cable from your local network to the Ethernet port on the board. Make sure your development PC is on the same network

xviii. Connect an HDMI (High-Definition Multimedia Interface) monitor to the HDMI port on the board (Optional).

xix. Connect the power supply to the micro USB port on the board.

Step 3: Set Up Azure IoT Hub

i. Create and configure IoT Hub in Azure
- Login to your new azure portal, click on browse -> select IoT Hub and click on Add icon.
- Give name for the IoT Hub, select pricing, enter resource name and region and finally click on create.
- Copy the host name which will be used in connection string at the time of registering device identity.
- Now click on All Settings -> Select Shared Access Policies -> select iothubowner. Copy primary access key which will be used in later steps.

ii. Device identity Registry
- Download pre- built version of device explorer.
- After downloading it is present here C:\ProgramFiles(X86) \Microsoft\DeviceExplorer
- Under device explorer go to configuration tab, enter your IoT Hub connection string and click on Update, after updating go to Management tab and then click on Create.
- Finally save the device Id and Primary Key values which will be used in later steps.

iii. Create and configure Event Hub
- Go to azure portal, click on browse select Event IoT Hub and click on Add icon.
- Enter the details then click on create.

Step 4: Create an UWP app for Raspberry Pi 2 Which will Send Sensor Data to IoT Hub and Receive Alerts from IoT Hub

i. After setting up the Development PC, Raspberry Pi and Azure IoT Hub, Open Visual Studio and create a new project by selecting File -> New -> Project -> Installed -> Templates -> Visual c# -> Windows -> Universal –> Blank App.

ii. Write the name of a project, and then click on OK.

iii. Now add reference of Windows 10 IoT Extensions to the recently created project.
- Solution Explorer (Right side of the Visual Studio) -> Reference -> Add Reference -> Universal Windows -> Extensions -> Windows IoT Extensions for the UWP (Select an appropriate version of IoT Extensions, Raspberry Pi 2 has the same version of Windows 10 IoT core installed.) -> OK.

iv. Also add Microsoft.Azure.Devices.Client reference from NuGet Manager.
- Solution Explorer -> App (Right click) -> Manage NuGet Packages -> Click on Browse tab -> Search Microfoft.Azure.Devices.Client -> Click on Install.

v. Reading Temperature values
- In this system we are reading temperature, pressure and altitude of a room.
- For temperature we used BPM280 V1 sensor which comes with Adafruit starter kit.
- Create helper class which has all functionality to Initialize and read values from BPM280 Sensor.

vi. Sending sensor data to IoT Hub
- The application reads data from sensors and send it to IoT Hub periodically. Following is the timer event code to read sensor data and send to IoT Hub.

vii. Write Device side code for receiving alerts from IoT Hub
- Timer method will send the temperature data for every time interval. After initiating timer method, you have to call the following method to receive alerts from IoT Hub.

viii. Deploy UWP app in to Raspberry Pi

- Set Project build type to Debug, device architecture to ARM and Device as Remote Machine.
- Remote Machine Configuration window enter the unique name of your device or IP address.
- The unique name is the name which you will create while setting up the device.

Step 5: Set Up Azure Stream Analytics to filter IoT Hub data and send it to Event Hub

i. To create a Stream Analytics Job in azure, log in to your azure classic portal. Select Stream Analytics from left side list and click on New plus icon on bottom of the page. Select Data Services->Stream Analytics->Quick Create

ii. Enter Job Name, Select Region and related region storage account. If these details already exist, then it automatically selects it, otherwise we have to create a new storage account by specifying name for the storage account. Finally click on Create Stream Analytics Job which is located at the bottom of New dialog screen.

iii. Select the newly created stream analytics, click on Inputs tab and select Add Input option available on the page.

iv. On Add an Input Popup window select Data Stream as the input and click on next.

v. Select IoT Hub as the data stream input and click next.

vi. Enter Input stream alias name which will be used in later steps, select subscription, choose an IoT Hub which we created in Step 3 Set Up Azure IoT Hub and select iothubowner as shared access policy name and click on next.

vii. Select Event serialization format as JSON and Encoding as UTF8 then click on create.

viii. Here temperaturealertsjob is my stream analytics job name and bmpsensordata is the alias name for my IoT Hub Input.

ix. Now add an output for the stream analytics job so that we can process the data which is coming from input stream and send it to the list of supported outputs.

x. Here we are using Event Hub as an output to the stream analytics job.

xi. Enter output alias name, event name and name space under which you want to create Event Hub. Leave the remaining fields with default values, and click on next.

xii. In the next page, just click on tick mark on bottom of the popup and leave all input fields with their default values. Now this creates a Service bus and a Event Hub in it.

xiii. Now go to Query tab and write a query to filter the data coming from IoT Hub input stream and pass it onto Event Hub. Write query to filter the temperature greater than 50 degrees. This is where you can apply your business rules/logic for the IoT data.

xiv. Now click on start icon from the bottom of the screen. If stream analytics fails to run, then go to Dashboard of your stream analytics job, select Operational Logs which is located right side of the page under Management Service.

xv. Select the log and click on Details icon which is located under bottom of the page

Step 6: Sending Alerts back to device through IoT Hub when the Temperature is High

i. This newly created Stream Analytics job will send temperature data to Event Hub when the temperature is greater than 50 degrees. Now we will create a cloud service that reads this data from Event Hub and sends it back to IoT Hub. And IoT hub will send it to the device back (Raspberry Pi). To achieve this, we need to configure Cloud Service.

ii. Configure a Cloud Service
- Select Cloud Service from left hamburger menu in Azure Classic portal
- Select New option located at the bottom of the page and enter the name for the service
- Now create a new project in Visual Studio of type Cloud Service, for this you need Visual Studio 2015 Update 2 and Azure SDK 2.9.
- Now select Worker Role from left side list and add it to the right-side list. Then click on OK.
- After successful creation of the solution, notice that there are two projects inside the solution, one is for the worker role library and another is for cloud service which will be published to azure.
- Update the WorkerRole.cs code to read data from Event Hub, and send the data back to IoT Hub.
- We will use App settings to store the Event Hub, storage account and IoT Hub keys and will use those in the Worker Role class with the help of ConfigurationManager class.
- Add two new classes in to WorkerRole project SensorEventProcessor.cs class and SimpleTemperatureAlertData.cs model class.

iii. Publish Cloud Service

- Now build the code and publish it to azure. Right click on Cloud Service project and select Publish option. On publish window, login using your Azure account and select appropriate subscription, then click on Next button.
- Make sure to select the cloud service that was created in previous step. keep all other settings to pre-populated values (Don't change them).
- Now click on Advanced settings. In that, cross check to make sure that the storage account which is selected, is the same that was mentioned in app.config. Now click on Next Button.
- Now click on Publish Button.

IV.4. RESULTS

A LED glows when you receive alert from IoT Hub on the event of the temperature being greater than the threshold temperature set. This is event is generated after it has been received from the Cloud service, post the real time analytics operation.

V. CONCLUSIONS

This system is a simple and effective demonstration of how to leverage the power of Azure IoT Hub and Stream Analytics to send real time alerts based on business rules, a phenomenon that truly maximises the array of benefits Internet of Things has to offer the smart, digitised world of today. Future enhancements to this system will include replacing the temperature sensors by using temperature and pressure sensors. Alternatively, application domain can be changed rather than an extension by involving the use of a radiofrequency circuit detector at the input side, and have RF levels of surroundings gathered and observed carefully.

REFERENCES

[1] Abhilasha Jayaswal, Ayushi Sharma, "A Study on Internet of Things", IJIRST, Vol. 3 Issue 4, September 2016

[2] James Nunns, "Real time data analytics – which tech giants are making serious plays in streaming big data analytics", December 2015 https://www.cbronline.com/big-data/analytics/real-time-data-analytics-which-tech-giants-are-making-serious-plays-in-streaming-big-data-analytics-4760742/

[3] Vishweshwar Hegde," IoT from end-point to end-user interactions and use cases", February 2017 https://www.linkedin.com/pulse/iot-from-end-point-end-user-interactions-use-cases-vishweshwar-hegde/

[4] Abhilash N., "How can the Internet of Things benefit your industry?" July 2015 https://www.linkedin.com/pulse/how-can-internet-things-benefit-your-industry-abhilash-nasre/

[5] Arshdeep Singh, Gurashish Singh, "Review on Temperature & Humidity Sensing using IOT" International Journal of Advanced Research in Computer Science and Software Engineering, February 2016

[6] Karan Kansara, Vishal Zaveri, Shreyans Shah, Sandip Delwadkar, Kaushal Jani, "Sensor based Automated Irrigation System with IOT: A Technical Review", International Journal of Computer Science and Information Technologies, Vol. 6 (6) , 2015

[7] H. Santhi, Gayathri. P, "A Review of Home Automation using IoT Applications", International Journal of Computer Science & Engineering Technology, Vol. 7 No. 07 Jul 2016