Real-Time Big Data Processing

Lab 3 – Aggregating Data Streams

Overview

In this lab, you will extend Azure Stream Analytics solution that you created in lab 2 by adding static reference data and aggregating the streamed data over a temporal window.

What You'll Need

To complete the labs, you will need the following:

- A web browser
- A Microsoft account
- A Microsoft Azure subscription
- A Windows, Linux, or Mac OS X computer
- The lab files for this course
- The Azure resources created in the previous labs

Important: If you have not completed labs 1 and 2, or you have deleted the event hub, storage account, IoT hub, and stream analytics jobs you created, complete the previous labs now.

Using Static Reference Data

Many real-time data processing solutions use static reference data to augment the streaming data. In this exercise, you will add a static dataset containing details of the devices submitting readings to your streaming solution.

Upload Reference Data to Azure

The device details data is provided as a text file, which you will upload to your Azure blob storage container.

1. In the folder where you extracted the lab files, open the **devices.csv** file in a text editor or spreadsheet application.
2. Review the device data, noting that the first value in each row is a device ID (in the format dev*n*) and the second value is the full name of the device (in the format Device *n*). Then close the file without saving any changes.
3. Start Azure Storage Explorer, and if necessary, sign into your azure subscription using your Microsoft account.
4. Expand your storage account, and then expand **Blob Containers**.
5. Double-click the **device-readings** container, and then in the **Upload** drop-down list, click **Files**.

6. Browse to the **devices.csv** file. Then in the **Blob type** list, ensure that **Block Blob** is selected, and in the **Upload to folder** box type **static-data**. Click **Upload** to upload the file.
7. Verify that **devices.csv** file is now stored in a folder named **static-data** in your blob storage container.

Modify a Stream Analytics Job

Now that you have uploaded the static reference data you can modify the query in the first of your Stream Analytics jobs to look up the device name.

1. In the Azure portal, browse to the first Stream Analytics job you created, which reads device data from an IoT hub input, and routes the processed results to a blob storage output and an event hub output.
2. In the blade for your Stream Analytics job, in the **Job Topology** section, click the **Inputs** tile.
3. In the **Inputs** blade, click **Add**.
4. In the **New input** blade, enter the following settings, and then click **Create**:
    - **Input alias**: DeviceDetails
    - **Source Type**: Reference data
    - **Subscription:** Use blob storage from current subscription
    - **Storage account**: *Select your storage account*
    - **Container:** device-readings
    - **Path pattern:** static-data/devices.csv
    - **Partition key column:** *Leave blank*
    - **Event serialization format:** CSV
    - **Delimiter:** comma (,)
    - **Encoding:** UTF-8
5. Wait for the input to be created and tested.
6. In the blade for your Stream Analytics job, in the **Job Topology** section, click the **Query** tile.
7. Verify that the existing query looks like this:

```
WITH [AllReadings] AS
(SELECT * FROM [DeviceData])

SELECT device, reading, EventEnqueuedUtcTime
INTO [DeviceReadings]
FROM [AllReadings]

SELECT device, reading, EventEnqueuedUtcTime
INTO [HighReadings]
FROM [AllReadings]
WHERE CAST(reading AS float) > 0.5
```

8. Modify the query as shown below, joining the streaming data source to the static reference data:

```
WITH [AllReadings] AS (
 SELECT strm.*, stat.DeviceName
 FROM [DeviceData] AS strm
 JOIN [DeviceDetails] AS stat
 ON strm.device = stat.DeviceID
)

SELECT device, DeviceName, reading, EventEnqueuedUtcTime
```

```
INTO [DeviceReadings]
FROM [AllReadings]

SELECT device, DeviceName, reading, EventEnqueuedUtcTime
INTO [HighReadings]
FROM [AllReadings]
WHERE CAST(reading AS float) > 0.5
```

9. Save the query, and then close the query pane.

View the Job Diagram

Your Stream Analytics job now consists of two inputs, connected to two outputs by two steps. You can verify this by viewing the job diagram.

1. In the blade for your Stream Analytics job, click **Settings**.
2. In the **Settings** blade, click **Job diagram**.
3. In the **Job diagram** blade, verify that your job consists of an IoT Hub input and a blob storage input, followed by two query steps, followed by a Blob Storage output and an Event Hub output.
4. Close the **Job diagram** blade and the **Settings** blade.

Start the Jobs

Now you're ready to start both jobs and test the streaming topology.

1. Start both analytics jobs and wait for them to start – this can take a minute or so.
2. When the jobs have started, in the Node.JS console, in the **iotdevice** folder, enter the following command to run device simulation script and start submitting messages to the IoT hub:

   ```
   node iotdevice.js
   ```

3. While the script is running, start Azure Storage Explorer, and if necessary, sign into your azure subscription using your Microsoft account.
4. Expand your storage account, and then expand **Blob Containers**.
5. Double-click the **device-readings** container, and then browse through the **readings** folder, and the year, month, and date folder to view the most recent blob that has been generated by your job.
6. Download the blob to open it in a text editor or spreadsheet application, and verify that it contains the device ID and name for each reading.
7. Close the downloaded file, and in the Node.JS console, press CTRL+C to stop the script.

Stop the Jobs

When you want to stop processing events, you can stop the jobs.

1. In the Azure portal, stop both stream analytics jobs.

Aggregating Events Over a Temporal Window

In most cases, data analysis involves aggregating individual data observations. When dealing with an unbounded stream of events, this aggregation is usually performed over temporal windows. For example, you find the average value of a field or the number of events within a specific time-period.

In this exercise, you will modify the query for the second Stream Analytics job to log alerts whenever a device experiences two or more readings greater than 0.5 over a 10 second period.

Modify a Stream Analytics Job

The second Stream Analytics job in your solution currently creates an alert for every reading greater than 0.5. This results in a high volume of alerts. You need to improve the solution so that an alert is only generated if a device experiences two or more high readings within a 10 second window.

1. In the Azure portal, browse to the second Stream Analytics job you created, which reads device data from an event hub input, and routes the processed results to a blob storage output.
2. In the blade for your Stream Analytics job, in the **Job Topology** section, click the **Query** tile.
3. Verify that the existing query looks like this:

```
SELECT
    *
INTO
    [DeviceAlerts]
FROM
    [HighReadings]
```

4. Modify the query as shown below, adding fields to show the start and end of each 10-second window in which a device experienced more than one high reading:

```
SELECT DateAdd(second,-10,System.TimeStamp) AS WinStart,
       System.TimeStamp AS WinEnd,
       DeviceName,
       AVG(CAST(reading as float)) AS AvgReading,
       COUNT(*) AS Alerts
INTO
    [DeviceAlerts]
FROM
    [HighReadings] TIMESTAMP BY EventProcessedUtcTime
GROUP BY DeviceName, SlidingWindow(second, 10)
HAVING COUNT(*) > 1
```

5. Save the query, and then close the query pane.

Start the Jobs

Now you're ready to start both jobs and test the streaming topology.

1. Start both analytics jobs and wait for them to start – this can take a minute or so.
2. When the jobs have started, in the Node.JS console, in the **iotdevice** folder, enter the following command to run device simulation script and start submitting messages to the IoT hub:

```
node iotdevice.js
```

3. While the script is running, start Azure Storage Explorer, and if necessary, sign into your azure subscription using your Microsoft account.
4. Expand your storage account, and then expand **Blob Containers**.
5. Double-click the **device-readings** container, and then browse through the **alerts** folder, and the year, month, and date folder to view the most recent blob that has been generated by your job.
6. Download the blob to open it in a text editor or spreadsheet application, and verify that it contains average readings and counts for high readings in 10-second windows.
7. Close the downloaded file, and in the Node.JS console, press CTRL+C to stop the script.

Stop the Jobs

When you want to stop processing events, you can stop the jobs.

1. In the Azure portal, stop both stream analytics jobs.

Note: You will use the resources you created in this lab when performing the next lab, so do not delete them. Ensure that all stream analytics jobs and Node.js scripts are stopped to minimize ongoing resource usage costs.