IoT Lab4 Summary Report

Group 1

(By Hélder Monteiro and Saquib Herman)

● What do the acceleration values actually tell us about movement?

Acceleration values observed in x(pitch), y(roll) and z(yaw) directions tells about the movement in respective directions.

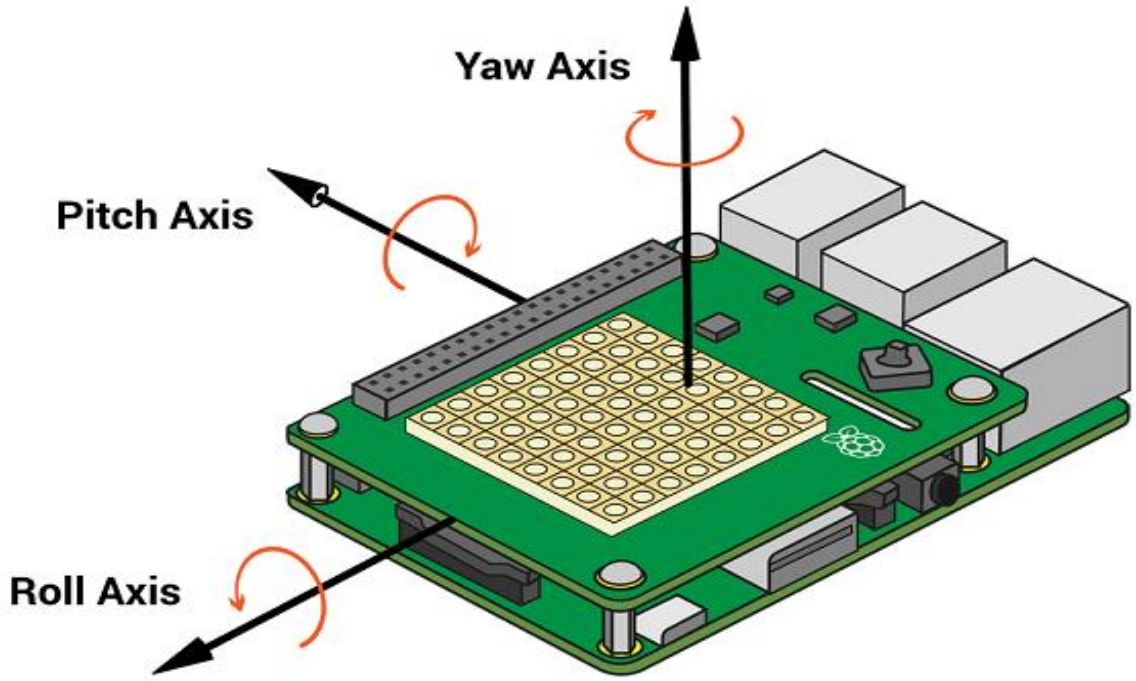

Figure 1: Raspberry Pi IMU movements supported by Sense Hat (Source: https://projects.raspberrypi.org/)

● What is the unit of acceleration values?

Unit of acceleration is **g** - earth mass acceleration (datasheet page 12).

● How is earth mass acceleration showing up in the accelerator values?

Acceleration values read using the API get_accelerometer_raw() tells us about the acceleration caused by the movement in x, y and z directions.

● How large is earth mass acceleration in accelerator values?

Below values of accelerations seen in respective x(accel_x), y(accel_y) and z(accel_z) directions while the board kept stationary.

{ "timestamp": 1614971094255, "device_id": "rasp1", "moved": 1, "accel_x": 0.0143043091521, "accel_y": 0.00291351089254, "accel_z": 1.01280403137, "direction": 78.5749128534 }.

● What is a reasonable trigger value/function for movement?

Movement is detected for slight change for accel_x > 0.1 (pitch), accel_y > 0.1(roll) and accel_z > 1.1(yaw) but the reasonable trigger values we figured out are accel_x > 1 , accel_y > 1 and accel_z > 1.1.

● How often do we need to sample to be sure to detect movement?

We observed, continuously sampling can be done without giving any delay to detect the movement on raspberry pi.

● If we want to do the detection in google cloud, what are the cloud tools one could use to achieve this (explain the tools needed and how they connect)

We need to use all the tools as we used in detection in edge, only difference will be done in cloud side is, we must implement the code of movement detection in cloud function(index.js) . We need to remove the movement detection code from the edge. From the edge in every 10 seconds we need to send the payload instead of sending it continuously. Under the condition check in index.js we commit the changes in BigQuery.

```
if ( data['accel_x'] > 1 || data['accel_y'] > 1 || data['accel_z'] > 1.1) {
        var d = new Date()
        var cur_time = d.getTime()
        console.log(cur_time)
        console.log(data['timestamp'])
        var latency = cur_time - data['timestamp']
        console.log('Latency in miliseconds: ' + latency)
        bigquery
        .dataset(datasetName)
        .table(tableName)
```

Left side shows the code of detection in the edge whereas right side shows the code of detection in the cloud side.

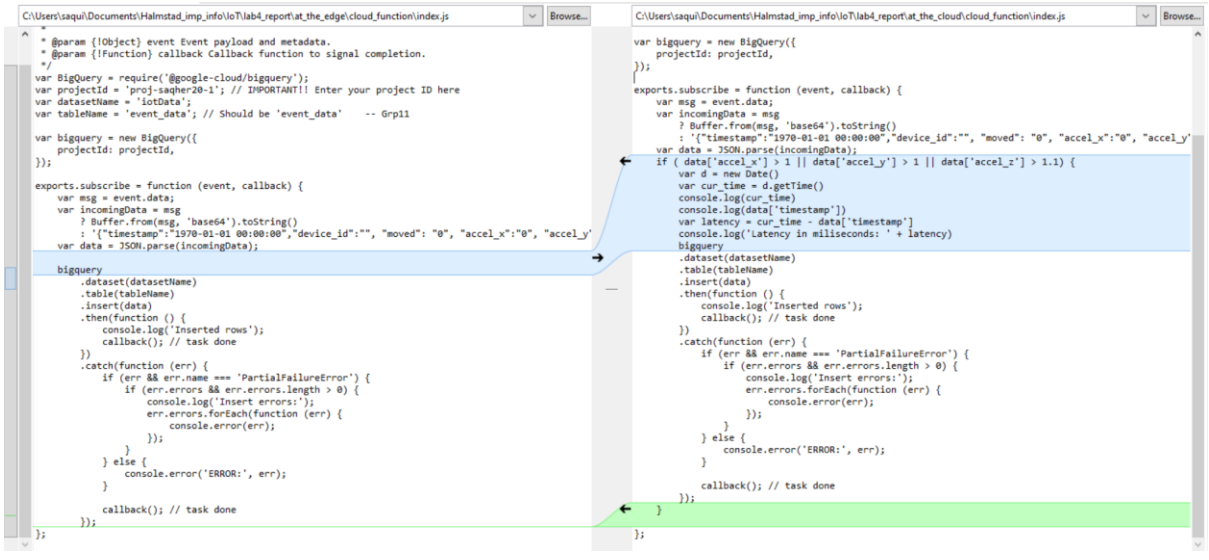

Figure 2: index.js file diff of edge vs cloud side implementation of movement detection.

and also we required to do below code changes in dt8030-lab4.py. Left side shows the code of detection in the edge whereas right side shows the code detection in the cloud side.

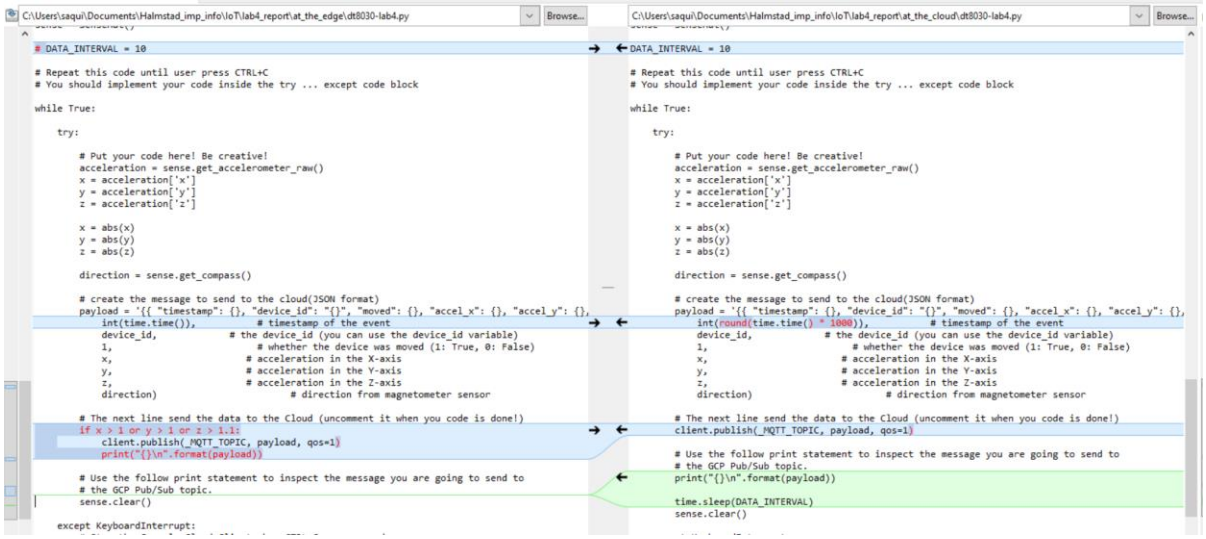

Figure 3: dt8030-lab4.py file diff of edge vs cloud side implementation of movement detection.

The estimate of the reduction in data transmission with our movement detector at the edge, compared with having the same detector in the cloud is in the order of **100ms**.

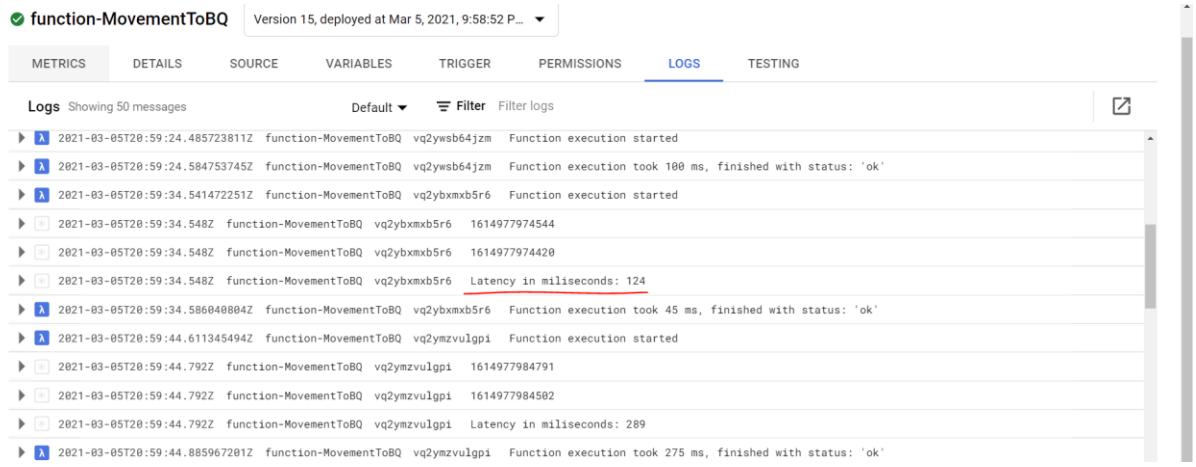

Figure 4: Log message of cloud side implementation of movement detection shows delay in milliseconds

Code changes both on the edge side(dt8030-lab4.py) and the cloud side(index.js) are attached in the zip file for both types of implementations.