MICROSERVICE ARCHITECTURE TO MONITOR BODY TEMPERATURE USING INTERNET OF THINGS

Santhosh Thiyagarajan

Student, Master of Technology in Software Systems, Brila Institute of Technology, Pilani(Raj.), India.

---***---

Abstract - *In recent times there is a rapid evolution in the healthcare industry due to exponential growth in the smart technologies like cloud computing and Internet of Things (IoT). Advancement in the IoT device helps in assisting medical professional to keep track of patients continuously. IoT enables to monitor the patient's blood pressure, body temperature, heart rate, blood glucose level. This paper focus on the leveraging the microservice architecture to collect the data about patient's body temperature using aurdino system and send the data to the cloud for processing, anomaly detection and sending alerts.*

***Key Words*: Internet of Things, Smart Healthcare, cloud, IoT Tracking, microservice.**

1. INTRODUCTION

The rapid development in the wireless sensor technology leads to the development of the wearable device which can become a life savers. Most of time the patients are needed to be monitored continuously, but with the current patient to doctor ratio it will be difficult to monitor each patient individually. With the advancement of the technology it has become an elementary job to monitor the patients. These data obtained needs to be processed in a proper way to get the maximum information from them. The system should be a self-learning system which has the capability of improving over the time through a continuous feedback mechanism. The data which the system receives is so much critical such that there should not be any down time in the system and the system should be intelligent enough to auto scale horizontally based on the load. The microservice architecture comes handy here. The system developed with the principles of microservice provides high availability and scalability. The data is more sensitive so that the security should be top notch.

This paper gives the detail implementation of the sensor module which can measure the body temperature and transmit the data over the Wi-Fi network to a webserver. The webserver will provide the resource to accept the post request data and store it in a queuing system. Data from the persistent queue can be taken by various systems to further processing and anomaly detection. Once there is a any expected anomaly detected the system should be capable of transmitting the notifications to the authorized personnel. The system is `capable to work autonomously without any manual intervention. This intelligence can be trained by providing the past system metrics data.

2. RELATED WORK

This section employs the various existing work in the field of IoT sensors and the microservice architecture to handle the enormous data flow.

[1]The data monitored using the sensors are directly transmitted using the GMS modules. Anomaly detection is performed by a simple comparative operations as the aurdino system doesn't have the capability to perform the Central Processing Unit (CPU) extensive data science program. The data transmitted over the network is sensitive in nature hence the data should be highly encrypted. [2] System security is increased using two level authentication mechanism with SHA-2 encryption technology. This also increase the robustness of the system exponentially. [3]The sensors needs to be powered by renewable energy sources. Theses sensors can be powered by solar energy to provide uninterrupted data collection form the patient. These devices can be turned into a wearable device such as watches so that the user can wear it all the time without any discomfort. [4]The devices can be designed over M2M protocols so that the system can be able to process huge volume of data to come to a conclusion. Computing power will be limited by a device and the devices cannot be scalable horizontally and it is only possible to scale only vertically.

For the system to be horizontally scalable it should be able to process the data in distributed manner. The system should be intelligent enough to detect the data load and auto scale. [5]The smart IoT system coupled with the microservice architecture provides a solution to the scalability of the huge data flow system. [6]The complex computational system needs to be split down into sizeable small components such a way that they can be able to scale on their own. These individual components should serve only one purpose which it should be able to define in a single line. This decomposition can be either data driven or the domain driven. The healthcare system the data is key component hence the system can be decomposed by a data driven architecture. The should be maintained consistently across all the microservice available without any losses.

3. ARCHITECTURE

The patient's body temperature will be monitored using a temperature sensors. The body temperature data will be then posted from the device to the cloud computing system. The device will use the HTTP post protocols to communicate with the cloud microservice. Once the data is received at the computing endpoint the raw data will be processed in real time. The will be analysed and the alerting system will be set to notify the respective doctors those who will be monitoring the patient. This communication will be happening over a SMS notification or a push notification in a smart device.

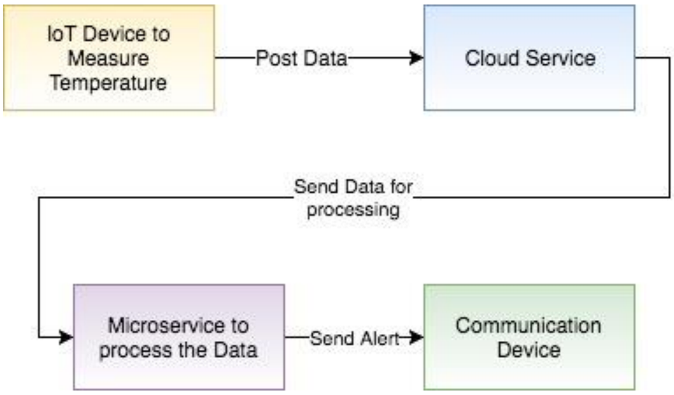

Fig -1: Block Diagram for Overall System

Fig -1 shows the overall block diagram of the system where the data collected using the sensors are transmitted over an IoT devices using the HTTP protocols and received at the computational server end. The computational system takes care of analysing the data and sending alerts to the respective persons. The alerts can be transmitted as SMS or a push notifications. This enables the doctors to pay attention to the patient in the emergency situations.

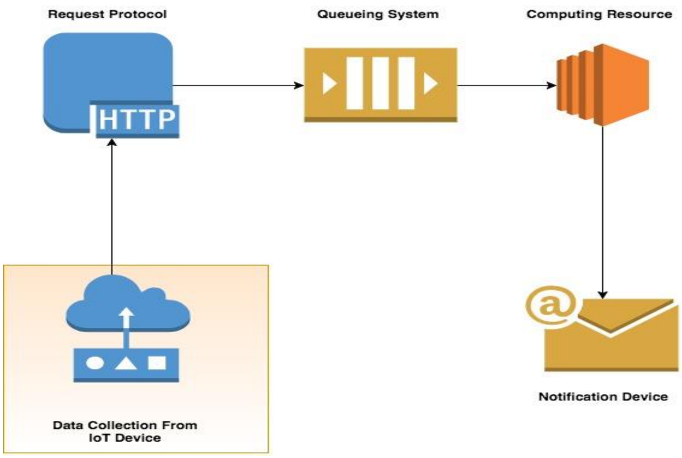

Fig -2: Microservice System Architecture

Fig -2 shows the architecture of the microservice. The data received through the HTTP request are stored on a processing queue. The queue will be listened continuously by an ever-running process which will take the data and process it for the anomaly detection. These instances can be scaled horizontally if there is huge data incoming. Once the anomaly is detected the notifications are sent to the respective persons through SMS/ Push notifications.

The given architecture is capable of adapting the various interfaces on demand by plug-in approach. As we read the data from the persistent queuing system various computation logics can be applied simultaneously on the data and take make various decision based on it.

4. IMPLEMENTATION

The implementation has two prats, one the Sensor circuit which is utilized to collect the body temperature and transmit the data over http protocol. Second part is the web service with microservice architecture which is scalable horizontally and robust in nature. This service is capable of anomaly detection and the notification service.

4.1 Sensor Circuit

The processing capability of the sensor circuit is powered by an Aurdino Chipset Board. This provides the high processing power to collect the data from the sensors. The LM35 sensor is used to make contact with the human body to measure the body temperature. The ESP86266-01 Wi-Fi Module is used to connect to the available network and transfer the sensor data over internet using HTTP protocol. The LCD Display is attached to the system to show the temperature in real time. The entire system will be powered by an 12V DC power supply.

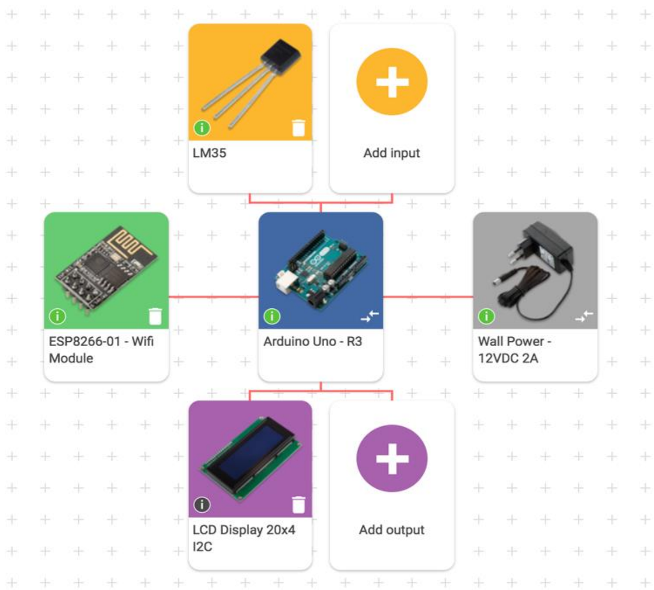

Fig -3: Components Diagram

The circuit connections are made as given in the Fig -4. The 12V DC supply will power the aurdino board and the remaining components will get the power supply from the output of the aurdino pins. The ESP8266-01 is a 3.3V 100 amp powered module so the power distributor is used to maintain the steady current flow to the module. The LCD display is connected to the digital pins of the aurdino board. The LM35 sensor is powered by 5V DC supply from the aurdino board and the voltage pin is connected to the analog pin in the board. This is used to make the voltage difference based on the temperature of the human body. This voltage difference will be sent out in the middle pin of the sensor.

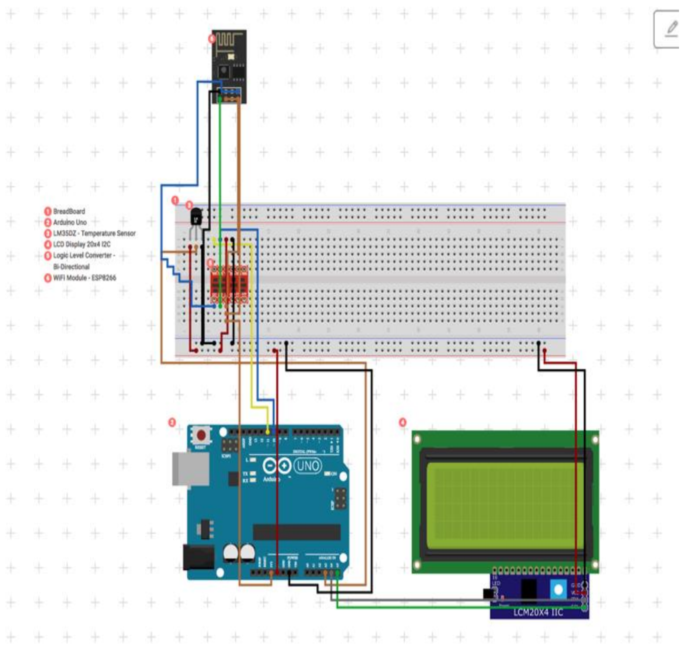

Fig -4: Circuit Diagram

The programmed in such a way that the voltage difference from the sensor is converted in to equivalent analog value then from the analog value the temperature is measured. Then the data will be transmitted over the network every second to the webserver. This data transfer uses the HTTP protocol over TCP so that the data is secured and transferred without any data loss

$$Temp = sensordata * \left[\frac{5V * 1000}{1024}\right] / 10$$

The analog data from the sensor is multiplied with the input voltage to sensor data ratio, this give the temperature measured by the sensor. This data will then be transmitted over the network to the webserver from the ESP8266-01 module.

The temperature measured is also sent to a programmed LCD display to track the patient in the real-time. This enables the doctors to monitor the patient condition in the hospitals.

4.2 Microservice Web Server

There are three microservice involved in the design. This enables the system to be an pluggable architecture, These microservice can be able to scale horizontally based on the demands. The designing of the microservice needs to be followed by 12 principles of microservice development. The three micro services are 1) the webserver API to handle HTTP post request 2) Queueing system to handle incoming data 3) The computational microservice with anomaly detection logic and notification mechanism.

4.2.1 API Webserver

The API server need to be highly available and scalable in nature. The high availability is required to receive the patient data and no data loss can be acceptable. The webserver is built with a load balancer and the deployments should be made as a blue green deployment mode. The received at the API end should be sent to the queueing system for further processing.

4.2.2 Queueing System

The queueing system should be able to handle the priority data and persist the data for long time. It should also provide the users to replay the data in a exact insertion order. The queuing system should be highly scalable as there will be huge amount of patient data will be gathered each and every second. The queueing system should also provide a compressing technique to persist the data.

4.2.3 Computation and Notification Service

The computation and notification microservice should able to read the data from the queues and process them in a timely manner. The picking from the queue should able to handle the priority to make sure the emergency patients gets priority attention. This system should be capable of detecting the anomalies based on the past data or a human input. The feedback mechanism should be ensured to keep the system updated. Once there is an anomaly detected the authorized personnel should get a notification about the patient. All the patient data should be stored in the sliceable database with proper archiving techniques to maintain the past history of the patient.

All these microservice need to be containerized and enabled with proper monitoring and metrics collection system to make sure they are always up and running in a healthy condition

5. RESULTS

The temperature data measured is transmitted ever second from the sensor circuit to the web server. The performance testing is done on the web server microservice. The load of the server is tested with number of simultaneous requests. The configuration of the machine is 2 core CPU with 4 GiB of Memory.

Table 1: Performance testing with simultaneous requests

Number Simultaneous Requests in Sec	Total Time Taken	Number of request per second
100	3 s	33.33
1000	37 s	27.02
10000	400 s	25.64

Table 1 shows that there is a deterioration in the performance with increase in the number of simultaneous request when the system is developed with single instance. When the system is built with auto scaling feature with microservice the following result in Table 2 is observed.

Table 2: Performance testing with simultaneous requests and auto scaling

Number Simultaneous Requests in Sec	Total Time Taken	Number of request per second
100	0.5 s	200
1000	0.4 s	2500
10000	0.6 s	16666

This shows that the microservice architecture will help in the IoT system which will generate huge simultaneous data in same time and the processing the data will happen in a matter of time.

6. CONCLUSION

The Healthcare system evolves with the growth in the technology as a smart healthcare technology which has become a life saver in the recent days. The proper monitoring and maintenance of the patient data with trained anomaly detection will lead to alerting the doctors at the right time to save a patient life. To handle the huge flow of the data the system need to be scaled horizontally. The microservice architecture will enable the system to handle huge data flow in a robust manner. The future work can be carried out to detect the nearest possible hospital for emergency and the availability of the doctors. As the system is designed with pluggable architecture various sensors can be used to collect the data such as heart rate monitoring, blood pressure monitoring.

ACKNOWLEDGEMENT

The author acknowledges Dr. Priyanka Parthiban for providing the vital support to develop the body temperature monitoring system with the medical background.

REFERENCES

Aziz, Kahtan, et al. "Smart real-time healthcare monitoring and tracking system using GSM/GPS technologies." Big Data and Smart City (ICBDSC), 2016 3rd MEC International Conference on. IEEE, 2016.

Yeh, Kuo-Hui. "A Secure IoT-Based Healthcare System With Body Sensor Networks." IEEE Access 4 (2016): 10288-10299.

Wu, Taiyang, et al. "An autonomous wireless body area network implementation towards IoT connected healthcare applications." IEEE Access 5 (2017): 11413-11422.

Alkhomsan, Mashail N., et al. "Situation Awareness in Ambient Assisted Living for Smart Healthcare." IEEE Access 5 (2017): 20716-20725.

Krylovskiy, Alexandr, Marco Jahn, and Edoardo Patti. "Designing a smart city internet of things platform with microservice architecture." Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on. IEEE, 2015.

Richardson, Chris. "Microservices Pattern: Microservice Architecture Pattern." Microservices.io, microservices.io/patterns/microservices.html.