Design and Implementation of Embedded Cloud Storage

Kiran A. R.
Assistant Professor, Dept. of ECE
Yenepoya Institute of Technology
Moodbidre, Karnataka

Mohan Kumar M.
Assistant Professor, Dept. of ECE
Yenepoya Institute of Technology
Moodbidre, Karnataka

***Abstract:* Cloud storage is one of the features of cloud computing which is a model of networked online storage, where data is stored on multiple virtual servers. In this work Cloud storage is realized using embedded system and can be implemented for a factory sensor system.**

In present-day factory sensor system, each machine will have appropriate sensor which will be connected to a microcontroller, a LCD and a keyboard. Supervisor is required to monitor and collect the data of each and every sensor. As the numbers of machines get increased, number of sensors also increases; in turn number of LCD's and keyboards. This in turn increases the burden of supervisory work and cost. Also, some sensor data's may be lost due to defect in machines. Embedded Cloud Storage is a better solution for these issues.

***Key words* - Embedded cloud, Cloud Storage Controller, Storage Server, Master Client, RS 485.**

I. Introduction

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. Figure 1 shows working of cloud storage system [4]. For some computer owners, finding enough storage space to hold all the data they have acquired is a real challenge. Desperate computer owners might delete entire folders worth of old files in order to make space for new information [1]. But some are choosing to rely on a growing trend: cloud storage. Cloud storage is a model of networked online storage where data is stored on multiple virtual servers [2].

In embedded cloud storage, sensors will act as client server and sends its data to the cloud, where a controller node called as master cloud will guide the sensor client data to the storage servers.

The main part of this work is to develop an embedded cloud storage system.

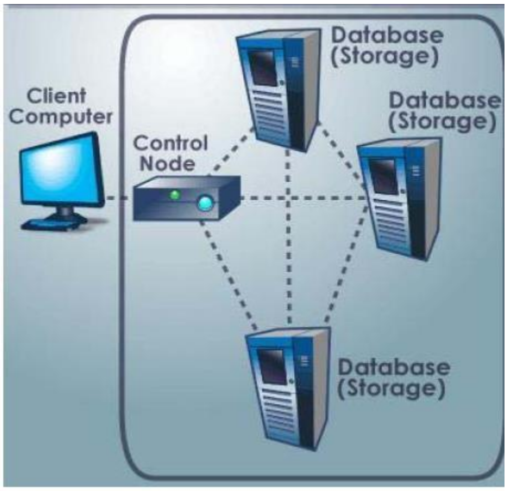

Figure 1: Typical cloud storage system architecture

II. Methodology

In an embedded cloud system, all sensors data will be stored at one place with date and time. Each sensor will have a unique id and will be registered before it starts communicating with the embedded cloud. Supervisor (users) can access the stored sensor data along with date and time. This system involves only one pair of LCD and keyboard to monitor all the sensor data and will have authentication as add on.

The main aim of project is to develop a centralized storage system; an ECS is as referred in figure 2.1. The main part of the work is to develop an embedded cloud system comprising of a master cloud controller and storage servers. The data received from sensor client module will be stored in the embedded cloud along with date and time. Storage servers comprising of storage servers and module to store data will take care of sensor data and provide timing facility too. The system also provides the user to check the stored data of sensors in cloud, using master client user can access the cloud to retrieve the data and also can get the current time

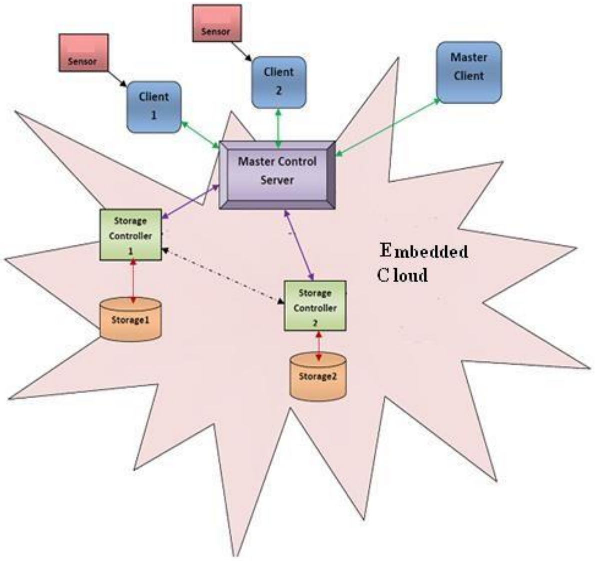

Figure 2: Embedded Cloud storage (Proposed System)

III. System hardware

The system mainly consists of following entities, namely:

- Storage server
- Sensor data acquisition system
- Master client
- Cloud storage controller

Block diagrams:

Embedded cloud block diagram is as referred in fig 3, which will comprise of cloud storage controller/ master cloud controller (MCC) which will receives packets from master client or sensor data acquisition system and provides write or read permission to the cloud and storage server is main module in cloud where sensor data will be stored along with data and time. Storage controller will communicate with MCC and also with EEPROM, where the sensor data will be stored and RTC will provide the timing facility.

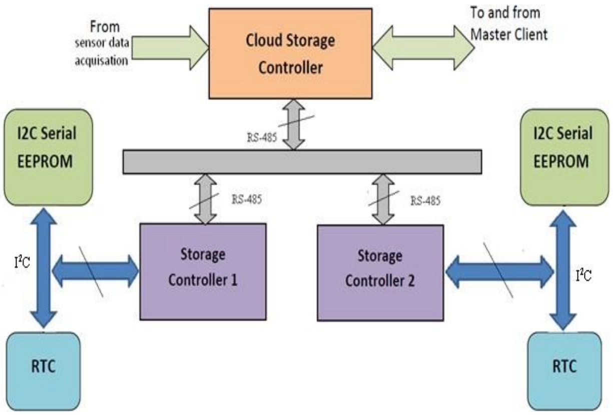

Figure 3: Embedded Storage Cloud

In sensor data acquisition system as referred in figure 4, sensor data will be converted in to digital data using ADC. This sensor data is framed into a packet format with access code, to gain storage access to embedded cloud. They respond to only one packet request from the cloud. Whenever master cloud controller sends get data request packet, the sensor client will respond with sensor data packet which will consist of updated average sensor data.

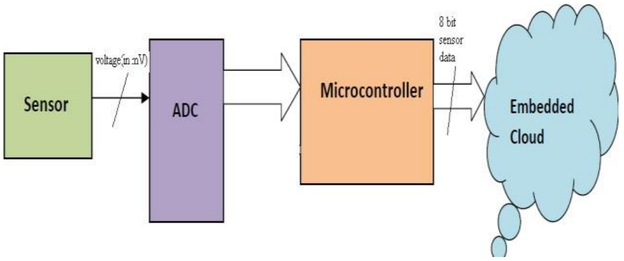

Figure 4: Sensor data acquisition system (DAQ) with cloud storage module

In master client system, a 4x4 keypad is used to browse the menu with the help of 16x 2 LCD which is as shown in figure 5. In menu system, supervisor (user) will have options to choose the required sensor data. Sensor data is accessed using its unique ID. An option to choose time of storage is also provided, which will allow user to browse current data as well as previously stored data.

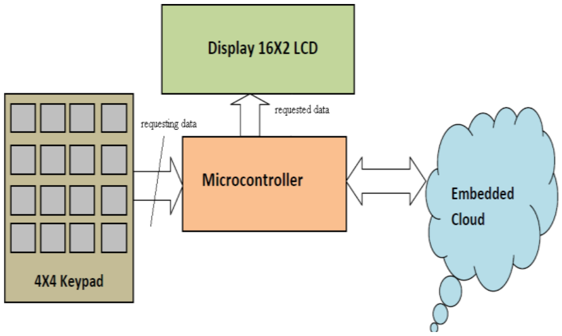

Figure 5: User Interface (Master Client) to access sensor data stored in the cloud

IV. Implementation of Master Cloud Controller and Storage Server

Master Cloud Controller (MCC):

The project is designed with a network using RS485 bus which uses 2 wire differential signal transmission for communication. Using MAX485, a multi-drop system is implemented via which many nodes were connected to RS485, but the care must be taken such that multiple nodes should not communicate simultaneously in order to avoid packet collision. To ensure avoidance of collision, a master cloud controller was designed which will take charge of complete bus and ensure collision free communication among the system.

Main job of master cloud controller is to get data from sensors (one by one) and communicate it to the storage server. Storage server will read the time from RTC and then store the sensor data with time. When data is stored successfully, master cloud controller will check for requests from master client: if any request is received, then MCC will process the request by communicating with storage server and sends reply to the master client. Master client display the results on the LCD screen to the user. The flow chart of MCC is as referred in figure 6.

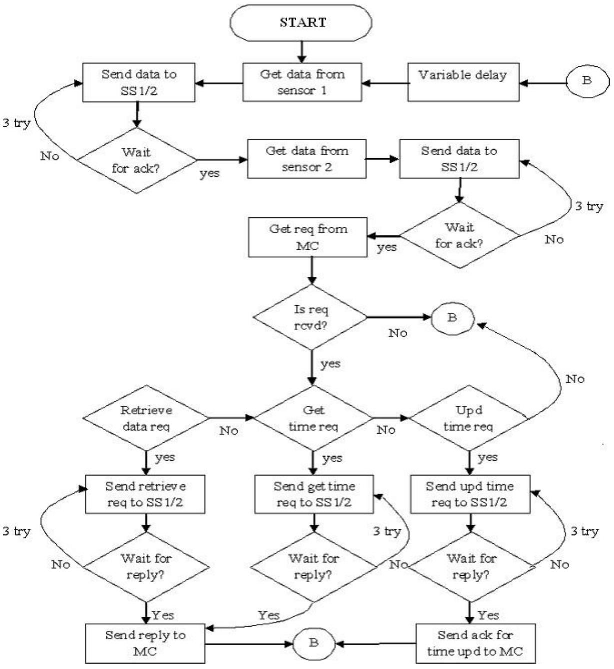

Figure 6: Flow Chart of MCC

Storage Server:

Storage server is the main module of the embedded cloud which will act as centralized storage system. Storage controller is a part of the storage server module, which controls the communication with EEPROM and RTC and also with MCC. The storage controller along with EEPROM and RTC will make a storage server module as referred in figure 7.

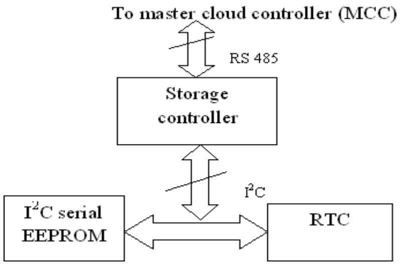

Figure 7:Block Diagram of one Storage Server

EEPROM will be used as storage part and RTC will provide timing facility. The main function of storage server is:

- To store sensor data to the EEPROM
- To send the requested data
- To update the time of RTC
- To get the current time

Real Time Clock will provide the timing facility and EEPROM will store sensor data along with sensor id, time and day no. While storing data, if memory becomes full the storage server will roll back to start address and then start overwriting the old data. When the memory reaches 80% of completion and message will be displayed to the user indicating memory full. Storage server keeps track of time using RTC with backup of 3.3V battery. The main operation performed by storage server is to store the sensor data, to retrieve the data for requested time, to get the current time or to update the time. Figure 8 shows the Flow chart of storage server:

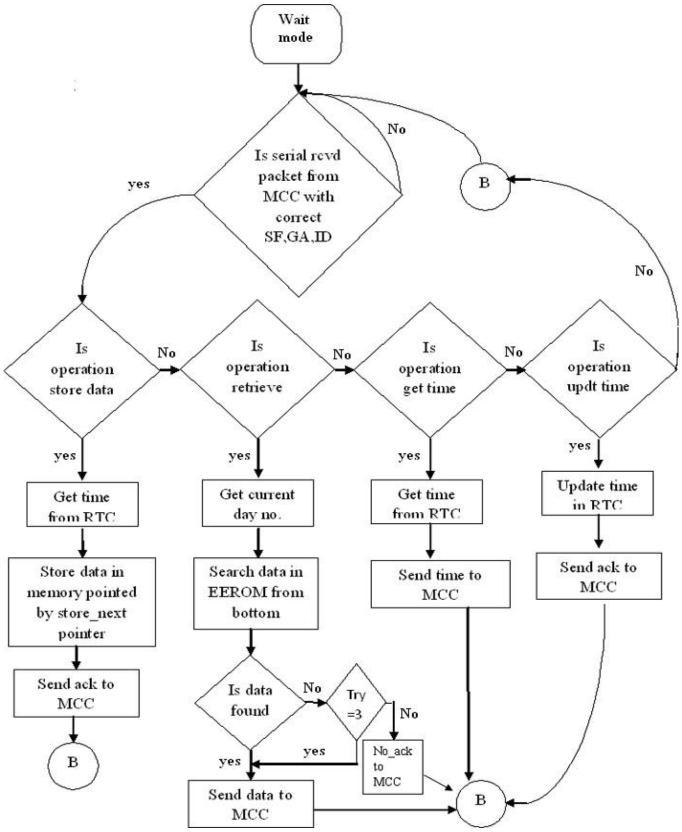

Figure 8: Flow Chart of Storage Server

In EEPROM Sensor data is stored as:

Sensor ID	data1	data2	Day No	Hour	Min

Consider an example to calculate the Day No:

For 5th May: Day no will be 31+30+5 = 66.

So if time stored is 66 14 20, it will be equivalent to 5th May 14:20.

V. Implementation of Master Client and Sensor Data Acquisition System (DAS)

Master Client:

Master client is equipped with human machine interfacing; it has LCD display and 4x4 matrix keypad as input. Full fledge menu based system is designed for ease of user. For security purpose master client is provided with a password. User can navigate to menu using up/down (A-B) keys and select the required menu, once a menu is selected as per the requirement of that operation, user is requested to enter information and the request is formatted and placed in the buffer and master client waits for cloud controller to get request packet; in other words it waits for its connection to embedded cloud.

After connection is established; master clients request will be sent to the cloud controller, cloud controller will process the request and returns the result to the master client and master client displays the appropriate result on the LCD screen. All communication between master client and cloud controller is done using packets. The working of master client is as shown in flow chart referred in figure 9.

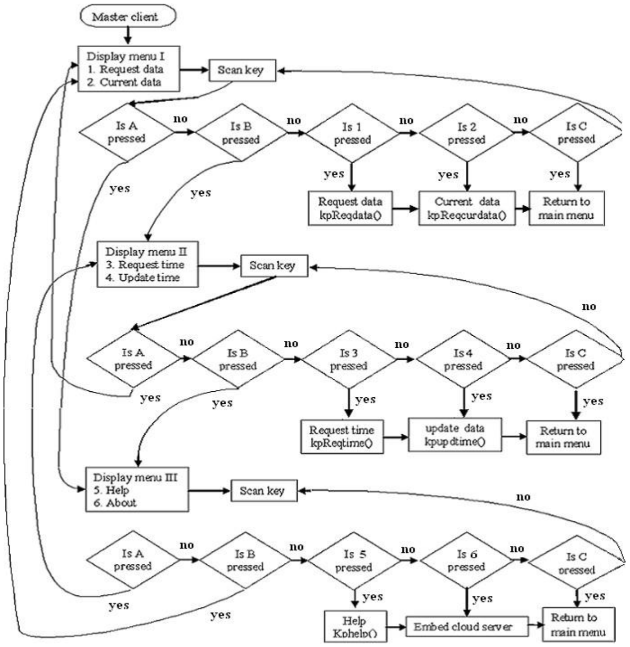

Figure 9: Flow chart of Cloud Storage Controller

Sensor Data Acquisition System (Sensor Client):

To demonstrate working of embedded cloud storage two sensors client were implemented, actually this could have done by simulating sensor client packets using terminal software and PC, but to imitate a real time scenario; two sensor clients based on light and temperature have been implemented. Two sensor clients were developed which will communicate with the master cloud controller (MCC) in the network. Sensor client do not know about number of storage servers present in the embedded cloud, the only work of sensor client is to send the data to MCC. Whenever MCC sends get data request packet, the sensor client will respond with sensor data packet which will consist of updated average sensor data. Temperature sensor (LM35) and light sensor (LDR) modules are developed.

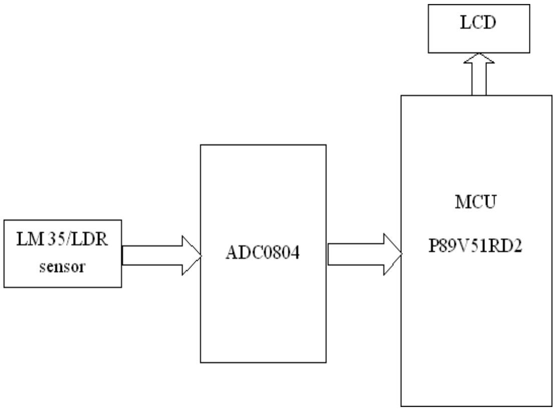

Figure 10: Block Diagram of Sensor Client

LCD was used initial to check working of sensor data acquisition module, to ensure proper working of hardware and software, which is shown in figure 10. Sensor client will respond only to cloud controller and only when cloud controller request data, it (sensor client) will give data back in packet form. Till then sensor client will keep on measuring parameter and average and update average at every cycle. Each and every time sensor client has to check request from master cloud controller; detailed working is as shown figure 11. Both the sensor client will communicate with master cloud controller using the RS 485 network. Figure 11 shows the flow chart of sensor client.

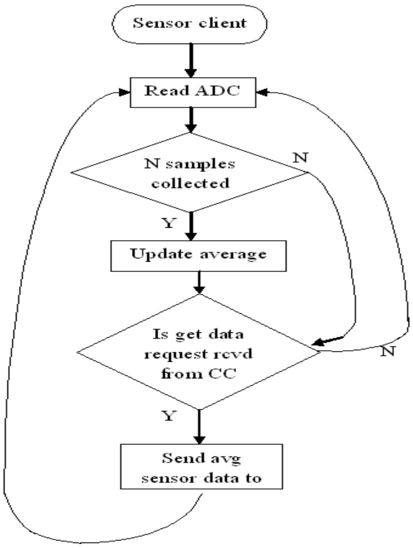

Figure 11: Flow chart for sensor client

VI. Implementation of Packet Communication and Packet Description

Master cloud controller controls all the clients in the network. Communication between the modules in the network is done with the help of packets. Proprietary packet communication protocol has been designed, which uses RS485 bus for communication. Network has distinct Start of Frame 0xFA abbreviated as SF and End of Frame FB abbreviated as EF. Packet is send as a multiple byte grouped together; all important information is embedded or sandwiched between SF and EF. The general frame format is shown below:

S F	Group Selection	Destination Address	Operation if required	Data 1*	Data n*	E F

Note: *means optional

Group Selection: For easier communication, storage servers, master client, sensor client modules are grouped separately and a hex address will be assigned to each group. This will save the time, as and when packet will be received group address will be checked first before the destination address.

Destination Address: The address where packet has to be delivered.

Operation: It provide the information about the operation to be performed like to request data, time, to update time.

Packets Used To Store Data

At first master cloud controller (MCC) will request the sensor client to send data:

SF	Sensor Group	Sensor id	Get Data	EF

As there will be multiple sensors, each will be allocated a unique id.

Sensor will reply data back to MCC:

SF	Master Cloud Controller	SN1	Data1	Data2	EF

The data send from sensor client will be processed to storage server, where data will be stored:

SF	Storage Server Group	SS id	Store Data	Sensor id	Data 1	Data 2	E F

An acknowledgement will be send back to MCC from storage server indicating data stored successfully:

SF	Master cloud controller	MC 1	ACK	EF

Packets Used to Process the Request from Master Client:

At first MCC will send a packet to get request from master client:

SF	M/S Client Group	MC1	Get Request	EF

If the request from master client is to retrieve data, then user (master client) will send following packet to MCC:

SF	Master Cloud Controller	Request data	Sensor id	Elapsed Day	HH	MM	EF

This packet is send by master client to MCC in order to request stored sensor data for a particular time. Sensor ID will be unique for each sensor module and elapsed day means: if the user requesting today data, elapsed day will be 0, and if yesterday data means it will be and so on. HH will indicate hour and MM will be minute.

MCC will process this request to storage server:

S F	Storage Server Group	SS id	Req Data	Sensor id	Elapsed Day	H H	M M	E F

The packet will be similar to that of request packet by master client but storage server and destination addressed will be changed by MCC.

Data found will be sent to the master cloud controller:

SF	Master Cloud controller	Sensor id	Data1	Data2	EF

The MCC sends the sensor data in above format to master client (user)

SF	M/S client Group	MS1	Sensor id	Data1	Data2	EF

If the request from master client is to update the time, then user (master client) will send following packet to MCC**:**

S F	Master Cloud Controller	Update Time	DD	MM	YY	HH	MM	E F

Once the time is updated an acknowledgement will be send by the MCC to master client in the following format:

SF	Master client group	MC 1	ACK	EF

The user will have an option to view current time of the storage server; to process this request a packet will be sent from master client to MCC:

SF	Master Cloud Controller	Get time	EF

MCC will get the time from storage server and send the current time to master client:

SF	M/S client Group	MC1	DD	HH	MM	EF

In EEPROM Sensor data is stored as:

Sensor ID	data1	data2	Day No	Hour	Min

VII. Conclusion

It was challenge to implement a system similar to cloud storage using hardware chips and embedded system. First task was to design the storage servers as centralized storage system, where sensor data will be stored safely along with time. Then design of master client, which decreases the number of LCD and Keypad used in the conventional system and also work of the supervisor to monitor the data, will be eased. For communication between the modules a network was developed using RS-485 bus. A proprietary packet communication protocol was also developed for the system. There was a chance that data packets of multiple servers and client on the system can cause clashes and data may be lost, hence master cloud controller was designed, which control the complete bus. Also two sensor clients (temperature and light sensors) were developed to check working of embedded cloud.

The developed system can be used for a factory sensor system which reduces the hardware cost and work of the supervisor. The data stored will be safe in embedded cloud and can be easily accessed by the user.

ACKNOWLEDGMENT

We thank the Management, the Director/ Principal and Staff of Channabasaveshwara Institute of Technology, Gubbi, Tumkur (Dist.), Karnataka, India for encouraging us for this research work.

References

[1] Zhang Jian Hua , Cloud Computing – based Data Storage and Disaster Recovery”, International Conference on Future Computer Science and Education, 201

[2] Jagan Srinivasan, Ting Yu , EMFS: Email-based Personal Cloud Storage, Sixth IEEE International Conference on Networking, Architecture, and Storage, 2011

[3] Qinlu He ,Xiao Zhang , Analysis of the Key Technology on Cloud Storage , International Conference on Future Information Technology and Management Engineering, 2010

[4] Jiyi Wu, Lingdi Ping , Cloud Storage as the Infrastructure of Cloud Computing , International Conference on Intelligent Computing and Cognitive Informatics, 2010

[5] http://www.swaroopch.com/notes/Distributed_Storage_Systems