

Smart environmental data management system into a cattle building

Anass Majdoubi^{1,*}, Abdellatif El Abderrahmani¹, and Rafik Lasri¹

¹Advanced Science and Technology Laboratory, FP LARACHE, Abdelmalek Essaadi University, Morocco

Abstract. The climatic atmosphere in which cattle live is an essential parameter of their environment because of its critical role in their productivity. An adapted cattle building must help to mitigate the effects of climatic stress and allow the farmer to properly control the climatic atmosphere during the production cycle. The most important factors influencing the climatic atmosphere inside a cattle building are temperature, humidity, and greenhouse gas emissions. We propose a case study for a wireless sensor network model placed on a cattle farm, in which each measurement node “mote” collects environmental data (temperature, humidity, and emission gas), in order to control the building's climate, this data is stored and managed in a remote database. We will present HBase, a NoSQL database management system, based on the concept of distributed storage, a column-oriented database that provides the read/write access to data on the HADOOP HDFS file system in real-time. The storage results presented in this paper are obtained via a java code that can connect with the HBase database, in order to store the received data at every second from each node constituting the measurement system via HTTP requests.

1 Introduction

Because of the connection of millions of devices, we can support today, the Internet of Things is defined as a promising paradigm in the field of advanced technologies, these connected objects spread around us everywhere in the form of sensors connected to the internet through types of communication such as radio frequencies, Bluetooth and Wi-Fi ..., which all physical objects can exchange information with one other.

Over the past decade, we have seen the transition from connected PCs to connected smartphones, and more recently we have seen this transition from connected smartphones to connected objects which are more intelligent and which communicate with each other and also with their environment. Connected objects are seen as sophisticated tools that will help us improve our lives.

A connected object is a gateway between the physical world and the digital world. When we link an object connected to a smartphone with wireless connectivity or connection, these connected objects will use their sensors to collect information on the environment or on the individual, they can also have actuators that can interact with their environment.

The ecosystem of the Internet of Things is essentially based on three major parts, the object part, the connectivity part, and the application part.

The object part: through embedded sensors can collect information on the environment such as temperature, light, and humidity, etc. And these objects can also store this information or transfer it via the

connectivity part. The connectivity part will allow the object to connect through a wireless network to the network of telecom operators or the Internet and which in fact routes the information to the application part. The application part: in fact, this part has an essential role to store raw information, analyze it, and transform it into a service dedicated to customers or users.

Today sensors are ubiquitous and generate huge amounts of data, sensors can be found in traffic lights which allow traffic jams to be detected [1], transport trucks equipped with a geolocation sensor [2], the car parks equipped with weight sensors allow the driver to quickly know where there is a free space, the security cameras, agricultural fields and smart homes that can be equipped with motion, light, and temperature sensors [3]. All these measurement devices and all these sensors collect and transmit massive data that require storage and analysis at a later date or immediately in order to make the required modifications to computers, mobile terminals, or processes of all kinds.

There are three main forms of IoT data storage: local, distributed, and centralized. Local storage indicates that the data captured by a sensor is stored in the local storage unit of the sensor [4]. Distributed storage means that data is stored in certain nodes of a network via distributed technologies [5], in this case, intermediate mechanisms are used to access the data. The last form of storage is centralized storage, in this case the data collected by each node of the network is sent and stored in a data center [6]. Due to the limited storage capacity and limited power of the sensors, the two forms of local and distributed storage are not suitable for IoT

* Corresponding author: anassmajdoubi@gmail.com

applications that require large-scale data and intensive queries. In addition, these two methods are not practical for sharing data between separate applications [4].

Therefore, many research and applications are focused on centralized form. Our work also relates to this type of storage as part of our study on improving the living conditions of cattle, in a cattle farm located around the town of Ksar El Kebir - MOROCCO, which we focus on minimizing climatic stress by improving the climatic atmosphere of the cattle building. While adequate temperature control and compliance with humidity standards are factors that aim to promote ruminant productivity, avoid respiratory problems, maintain a good supply of litter, and reduce expenses for veterinary products. The authors in [7] found that methane emissions are optimized when the temperature varies between 10 °C and 15 °C, and they indicated that this optimal temperature range represents the welfare temperature of dairy cows. The authors of [8] indicate that milk production could decrease by around 2.8% due to climatic stress, cattle respiration rate can be increased up to 60% and rest time can be extended by 1 hour.

We designed a storage solution called IOTHNODE based on the NoSQL HBASE database, which has been successfully applied to the environment to cope with the huge amount of data, to solve the problem of storing and managing the massive environmental data generated on a cattle farm.

This article is structured as follows. In Sections 2, we looked at the related work regarding the storage of IoT data. Section 3 presents the design of the architecture of IOTHNODE. The choice of the "HBASE" database management system was given in section 4. Section 5 gives a description of the storage results. Conclusions are drawn in section 6.

2 Related works

The explosion of IoT applications in various fields poses a problem of heterogeneous data management. This data should be processed and analyzed in order to optimize the efficiency of the applications, thus tasks such as collecting, storing, and generating reports are managed by database management systems. Recently, several studies have proposed database management systems for storing, analyzing, and processing the massive data generated by IoT applications.

In this section, we discuss these studies, in particular, those which are representative of the state of the art and close to our work. Many researchers have addressed issues closely related to our work such as those in [12] [13] and [14].

The work in [9] focuses on the analysis, storage, and management of a massive amount of RFID data via a model called RFID-Cuboids. To manage the huge amount of data and make data recovery faster, the authors [10] proposed the conception, implementation, and evaluation of three data recovery approaches for RFID tags (sequential, parallel, alternative), these three approaches are characterized by the programming of the

sampling and sensor transmission operations. A data repository schema based on MongoDB which can effectively integrate and store IoT data from heterogeneous sources such as RFID, sensors, and GPS, [11].

Another database solution for IoT applications is provided by a distributed database [12], where each node on the network is capable of storing measurements and transmitting detected data at the request of the user. The author [13] provides a tool to facilitate the interconnection between the building design process and environmental data, taking into account the requirements of the new thermal regulation of 2020, this allows the creation of a SQL environmental database, based on environmental declarations, more suited to analyze. The authors in [14] propose a pervasive information system for a smart village based on a network of LoRAWAN connected objects and a NoSQL database for data management.

For various IoT applications, there is always the question of whether SQL or NoSQL databases prove to be better, and always remain without standard response. The authors in [15-17] propose a comparison study between SQL and NoSQL database management systems for IoT applications.

Several NoSQL systems are available. The widely-used open source systems are MongoDB, CouchDB, HBase, and Cassandra. For widely used non-open source systems there is BigTable from Google and Dynamo from Amazon. For our work, we used the HBase database management system, a distributed database based on the Hadoop HDFS distributed file system [18]. HBase was used for the possibility of real-time read/write access to very large amounts of data.

3 Proposed architecture

Our work is based on the control and management of storage of environmental data collected by a wireless sensor network placed in a cattle building, where each measurement node collects the temperature, relative humidity and gas emission of the environment studied, the three environmental measures, which determine the freshness and the quality of the area of each location covered by this type of object.

We have already created a prototype of an object locally connected via Wi-Fi, an object able to detect measurements of temperature, humidity, and gas emission. We built our prototype as shown in figure 1, which contains two sensors, the DHT11, and the MQ-2, connected with the NodeMCU-ESP8266 card, which controls the two sensors to determine and collect the three environmental measurements.

Among the critical features of this object is that it is able to transfer data as an HTML response after an HTTP request. Via this possibility of transfer, architecture has been designed for storing this environmental data in a remote database, see figure 2. This gives us the possibility of saving the data collected from each node constituting the network.

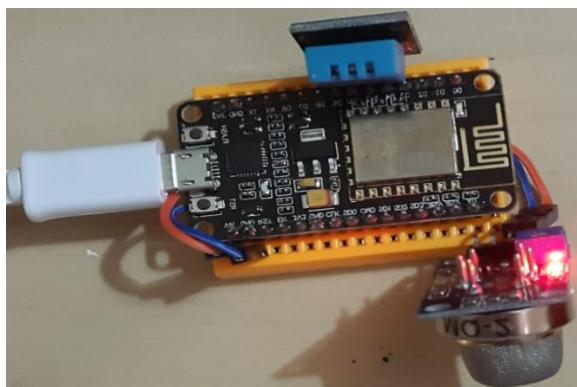


Fig. 1. Prototype of measurements

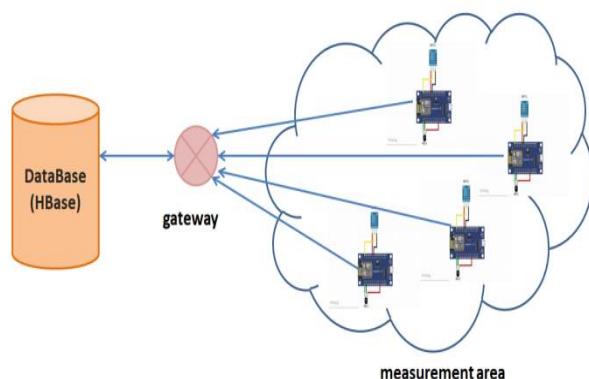


Fig. 2. The architecture of the proposed system.

4 Choice of the “Hbase” database

After the functional test of our prototype, this records the three measurements of temperature, humidity, and gas emission. Now we have to store these three measurement values. Knowing that we want to record every second the three measurements collected by each node constituting the measurement system. We are facing a high number of recordings, which requires a database management system that is capable of handling massive and voluminous data.

To solve the problem of recording data from our system, we chose the NoSQL "HBASE" database engine. This database system ensures the storage of very large volumes of data in real-time, which meets our needs with regard to the storage of the three values captured by the system of real-time measurements.

4.1 HBase data model

HBase is a distributed, non-relational (NoSQL) and column-oriented database management system designed for Big Data analyzes [18]. It is developed on top of the HDFS file system. It makes it possible to process very quickly and in real-time a very large volume of data from different sources and various structures.

The given HBase model is based on six concepts, which are:

- Table: in HBase, the data is organized in tables. The names of the tables are strings.

- Row: in each table, the data is organized in rows. A row is identified by a unique key (RowKey). The RowKey has no type, it is treated as an array of bytes.
- Column Family: The data within a row is grouped by the family of columns. Column families are defined when the table is created in HBase. The names of the column families are strings.
- Column Qualifier: Access to data within a family of columns is via the "column qualifier" or column. It is not specified when the table is created, but rather when the data is inserted. Like RowKeys, the column qualifier has no type, it is treated as an array of bytes.
- Cell: The combination of RowKey, a family of columns, and the column uniquely identifies a cell. Data stored in a cell are called the values of that cell. Values have no type, they're still considered byte arrays.
- Version: The values within a cell are numbered. The versions are identified by their TIMESTAMP (long type). The number of versions is configured via the column family. By default, this number is three.

4.2 Data storage characteristics

Data in HBase is stored as HFfiles, in columns, in HDFS. Each HFfile is responsible for storing data corresponding to a particular family of columns, see figure 3. Knowing that HBase does not have a predefined schema, except that it is necessary to define the families of columns when creating the tables as shown in table 1, because they represent the physical organization of the data in the form of collections of type key/value.

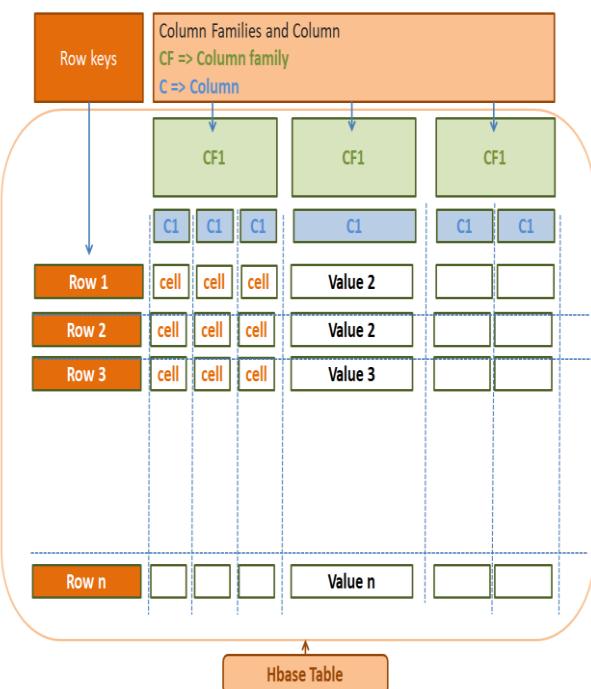


Fig. 3. Descriptive schema of Hbase.

Table 1. The storage structure.

		NodeMCU1		
DateTime	temperature	humidity	Gaz emission	
20-05-2020 00:00:00	19	40	70	
20-05-2020 00:00:01	18	40	70	
20-05-2020 00:00:02	18	41	72	
20-05-2020 00:00:03	17	41	75	

5 Description and storage results

To deal with the management of voluminous environmental data generated every second by all the nodes that make up the distributed measurement system, we have developed a java code that can connect with the HBase database, in order to store the data received from the measurement system via an HTTP request at every second.

We have proposed a simple algorithm based on the client/server architecture, see figure 4. Where we can make HTTP requests to all the measurement nodes, in order to get the HTML responses for each measurement node, then we extract the associated collection information from each node, as the last step we store this data in HBase column families, where each column family represents a measurement node and this gives the flexibility to add other measurement nodes with measurements appropriate to the storage system without any standard structure constraints such as relational database engines.

Pseudocode : measurement Storage THG

INPUT : N measurement nodes, T temperature, H humidity, G gaz
OUTPUT : HM hbase table of measurement

```

Begin
    While network is available do
        for each node  $Ni$  from  $N$  do
            1 – HTTPRequest( $Ni$ );
            2 – HTMLResponse( $Ni$ );
            3 –  $T, H, G = getMesures(Ni)$ ;
            4 –  $HM.put( dateTme, \{Ni : temperature' : 'T',
                                'Ni : humidity' : 'H,
                                'Ni : gaz': 'G\})$ 
        end for
    end while
end

```

Fig. 4. The environmental data storage algorithm.

FIG. 5 represents an example extracted from the HBase database management system installed in a LENOVO computer operating under the Ubuntu Linux operating system, and we used two measurement nodes to collect information from two separate places (see Table 2).

Table 2. The tools used in the experiment.

Hardware	Description
Laptop	Lenovo Ram 8Go, Intel Core i7-5500U 2.40GHZ 2.40GHZ, Linux (Ubuntu 14.04), 64 bits.
Two measurement nodes	NodeMCU-ESP8266 card * DHT11 Sensor * MQ2 Sensor
Wi-Fi gateway	Huawei B310s-22

The result obtained via the « scan‘measures’ » command, of which ‘measures’ is the name of the measuring table for each node making up the distributed measurement system. As we said earlier, each family of columns represents the node information such as the IP address, the location, the description and the measures associated with the DateTme key of these measures.

```

hduser@anass: ~
2019-09-06 06:53:24 column=N:gaz, timestamp=1567749204936, value=104.0
2019-09-06 06:53:24 column=N:humidity, timestamp=1567749204936, value=63.0
2019-09-06 06:53:24 column=N:ip, timestamp=1567749204936, value=192.168.1.109
2019-09-06 06:53:24 column=N:region, timestamp=1567749204936, value=req2
2019-09-06 06:53:24 column=N:temperature, timestamp=1567749204936, value=23.0
2019-09-06 06:53:25 column=N:description, timestamp=1567749205679, value=modification 0609
2019-09-06 06:53:25 column=N:gaz, timestamp=1567749205679, value=104.0
2019-09-06 06:53:25 column=N:hunedity, timestamp=1567749205679, value=63.0
2019-09-06 06:53:25 column=N:ip, timestamp=1567749205679, value=192.168.1.109
2019-09-06 06:53:25 column=N:region, timestamp=1567749205679, value=req2
2019-09-06 06:53:25 column=N:temperature, timestamp=1567749205679, value=23.0
2019-09-06 06:53:26 column=N:description, timestamp=1567749206168, value=modification 0609
2019-09-06 06:53:26 column=N:gaz, timestamp=1567749206168, value=104.0
2019-09-06 06:53:26 column=N:humidity, timestamp=1567749206168, value=63.0
2019-09-06 06:53:26 column=N:ip, timestamp=1567749206168, value=192.168.1.109
2019-09-06 06:53:26 column=N:region, timestamp=1567749206168, value=req2
2019-09-06 06:53:26 column=N:temperature, timestamp=1567749206168, value=23.0
2019-09-21 13:53:30 column=node99:description, timestamp=1569070415536, value=ueeo
2019-09-21 13:53:30 column=node99:gaz, timestamp=1569070415536, value=90.0
2019-09-21 13:53:30 column=node99:humidity, timestamp=1569070415536, value=64.0
2019-09-21 13:53:30 column=node99:ip, timestamp=1569070415536, value=192.168.1.104
2019-09-21 13:53:30 column=node99:region, timestamp=1569070415536, value=req2
2019-09-21 13:53:30 column=node99:temperature, timestamp=1569070415536, value=22.0
2019-09-21 13:53:35 column=node99:description, timestamp=1569070415917, value=ueeo

```

Fig. 5. Data record extract.

The table below shows an example of the storage of measurement data of a node named "node2" at an IP address "192.168.1.103" with other information regarding this node.

Table 3. An array of measurement data of node "node2".

Name	Ip address	T	H	G	area	DateTime
Node2	192.168.1.103	23	49.0	102	Ksar	2019-08-28 06:25:10
Node2	192.168.1.103	23	49.0	102	Ksar	2019-08-28 06:25:11
Node2	192.168.1.103	23	49.0	102	Ksar	2019-08-28 06:25:12
Node2	192.168.1.103	23	49.0	102	Ksar	2019-08-28 06:25:13
Node2	192.168.1.103	23	49.0	102	Ksar	2019-08-28 06:25:14
Node2	192.168.1.103	23	49.0	102	Ksar	2019-08-28 06:25:15
Node2	192.168.1.103	23	49.0	102	Ksar	2019-08-28 06:25:16

The storage results obtained in this article are based on a local Wi-Fi connection model, in our case, due to the absence of a supplier of the associated networks for the IoT applications in Moroccan territory, in a similar French project [14] the authors propose a simple pervasive information system adaptable to small non-urban territories based on a network of connected objects using a LoRAWAN network. Until now this type of the network is unavailable for our case, which generally poses constraints regarding the deployment of connected objects in areas that are not covered by the network.

6 Conclusions

The necessity of IoT and its various applications become including all areas of our daily and professional life. With this development, there is a major problem concerning the processing, analysis, storage, and management of massive amounts of data, which requires the parallel development of relevant solutions capable of intervening with this kind of data.

In this work, we proposed a storage solution for environmental data generated by prototypes of measurements placed in a cattle building. Where we put and treated a case study of a distributed measurement system, which collects environmental data of temperature, humidity, and gas emission, stores them in

a database using a non-relational "HBase" database management system. This meets our needs regarding the storage and management of a huge amount of data collected.

Finally, we are always impatient to improve our work, especially regarding performance and intelligence. We would like in future work to interact with case studies performing more powerful recording solutions that integrate remote processing and proximity storage technologies, based on "cloud computing" techniques, in order to optimize communications between a large number of connected objects contributing to a network of distributed measurements.

References

1. S.Faye, C.Chaudet, and I.Demeure, *Controle du traffic routier urbain par un réseau fixe de capteur sans fil*, [Research Report], TELECOM ParisTech, hal-00781140, pp. 5-10, 2018
2. Z.Maldizan, *Suivi et localisation des marchandises en temps réel à partir des camions de transport sur longue distance*, [Research Report], école polytechnique de Montréal, pp. 28-44, 2012.
3. A.Yassine, et al, *IoT big data analytics for smart homes with fog and cloud computing*, Future Generation Computer Systems (2018), <https://doi.org/10.1016/j.future.2018.08.040>.
4. N. Tsiftes, and A. Dunkels, *A Database in Every Sensor*, in Proceedings of the 9th ACM Conference on Embedded Network Sensor Systems, pp. 316-332, 2011, <https://doi.org/10.1145/2070942.2070974>.
5. G. Weimin and Z. Lingzhi, *Distributed Data Storage in Wireless Sensor Networks*, International Journal of Database Theory and Application Vol.8, No.4, pp.179-182, 2015, <http://dx.doi.org/10.14257/ijdta.2015.8.4.18>.
6. S. R. MADDEN, *TinyDB: An Acquisitional Query Processing System for Sensor Networks*, ACM Transactions on Database Systems 30(1), pp. 122-173, 2005, <https://dl.acm.org/doi/10.1145/1061318.1061322>.
7. S.Hempel, D.Willink, D.Janke, C.Amon, B.Amon, T.Amon, *Methane Emission Characteristics of Naturally Ventilated Cattle Buildings*, Sustainability **2020**, 12, 4314.
8. S.Hempel , C.Menz, S.Pinto, E.Galán, D.Janke, F.Estellés, T.Müschner-Siemens, X.Wang, J.Heinicke, G.Zhang, B.Amon, A.del Prado, and T.Amon, *Heat stress risk in European dairy cattle husbandry under different climate change scenarios – uncertainties and potential impacts*, Earth Syst. Dynam., 10, 859–884, <https://doi.org/10.5194/esd-10-859-2019>, 2019.
9. H. Gonzalez , Jiawei Han , Xiaolei Li and D. Klabjan, *Warehousing and Analyzing Massive RFID Data Sets*, 22nd International Conference on Data

- Engineering (ICDE'06), 2006,
<https://ieeexplore.ieee.org/document/1617451>.
10. Y. SU, A. Wickramasingh and D. C. Ranasingh *Investigation sensor data retrieval shemes for multi-sensors passive RFID tags*, 2015 IEEE International Conference on RFID (RFID), San Diego, CA, pp. 158-165, 2015,
<https://ieeexplore.ieee.org/document/7113087/>.
 11. Y. Kang, I. Park, J. Rhee and Y. Lee, *MongoDB-Based Repository Design for IoT-Generated RFID/Sensor Big Data*, in IEEE Sensors Journal, vol. 16, no. 2, pp. 485-497, Jan.15, 2016,
<https://ieeexplore.ieee.org/document/7279070>.
 12. O. Diallo, J. J. P. C. Rodrigues, M. Sene and J. Lloret, *Distributed Database Management Techniques for Wireless Sensor Networks*, in IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 2, pp. 604-620, Feb. 2015,
<https://ieeexplore.ieee.org/document/6582416>.
 13. A. Jerome, *Integration of LCA into the building design processes*, Dissertation, 2019.
 14. T. Antoine-Santoni, B. Poggi, E. Vittori, H. V. Hieu, D. Araujo, et al, *Vers un système d'information pervasif pour un Smart Village, Evolution des SI : vers des SI Pervasifs ?*, Université Paris 1 Panthéon-Sorbonne, Jun 2019, Paris, France. fhal 02152730v2ff.
 15. Azad, P., Navimipour, N.J., Rahmani, A.M. et al. *The role of structured and unstructured data managing mechanisms in the Internet of things*, Cluster Comput (2019).
<https://doi.org/10.1007/s10586-019-02986-2>
 16. L. Incipini, A. Belli, L. Palma, R. Concetti and P. Pierleoni, *Databases Performance Evaluation for IoT Systems: the Scrovegni Chapel Use Case*, 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2019, pp. 463-468, doi: 10.23919/MIPRO.2019.8756813.
 17. Ahmad, Azhar & Said, Nur & Hamid, Roslina & Zainuddin, Fauziah & Mokhtar, Rahmah. *Influence of Network Speed on Structured Query Language (SQL) Database Data Transmission Performance in an Internet of Things (IoT) Sensing Device on Single Board Computers (SBC)*, in Conference on Green Engineering Technology & Applied Computing, Bangkok, Thailand, 2019
 18. K.Krishnan, *Building Big Data Applications*, Chapter 2- Infrastructure and technology, Academic Press, 2020, pp 17-71, ISBN 9780128157466, <https://doi.org/10.1016/B978-0-12-815746-6.00002-8>.