# Sensor Data Pre-processing System for Storing Real-time Sensor Stream Data

Seungwoo Jeon[1], Bonghee Hong[1], Joonho Kwon[2], Yoon-sik Kwak[3], Seok-il Song[3]

1 Dept. of Computer Engineering, Pusan National University
Jangjeon-dong Geumjeong-gu, Busan 609-735, Korea
{i2825t, bhhong}@pusan.ac.kr
2 Institute of Logistics Information Technology, Pusan National University
Jangjeon-dong Geumjeong-gu, Busan 609-735, Korea
jhkwon@pusan.ac.kr
3 Dept. of Computer Engineering, Korea National University of Transportation
50 Daehak-ro, Chungju-si, Chungbuk, 380-702, Korea
{yskwak, sisong}@ut.ac.kr

**Abstract.** Nowadays, various monitoring systems are built using sensor network. One of them is warehouse monitoring system which used for monitoring temperature, humidity, and $CO_2$ level. The sensing data size may vary depending on how many property it has but it is relatively small, around 100 bytes. However, since there are several warehouses which are used for storing the products and every warehouse consist hundreds sensors, the collected sensing data can be very large. Instead of using the naïve approach, storing each data on a single file which can lead into an enormous number of data access into file system, our proposed data input system collect and combine several sensing data into a big chunk of file and save it to HDFS. Moreover, it will use MapRe-duce framework for accessing and processing the sensing data chunk inside HDFS.

**Keywords:** Distributed sensor stream data, HDFS

## 1 Introduction

In logistic area, sensor networks play an important role in monitoring the warehouse condition, such as: temperature, humidity, and $CO_2$ level in order to maintain the products quality inside it. Each sensor network may have thousands or more sensors and organized by a certain number of readers. The sensing data sent by sensor will be read by the reader every certain period of time called read cycle. Even if the data sent by the sensor to reader is very small, the collected data in the server side can be abundant, this is because the large number of sensor involved in the sensor networks. This can lead into bottlenecking system due to a very high data access to the file system. Therefore, instead of storing every single sensing data one-by-one, we propose a new technique that will combine several data to fit a predefined chunk of file to minimize the number of data access to file system.

The remaining of this paper is organized as follows. Section 2 discusses related works on HDFS and Wireless sensor networks. In Sec. 3, we define a problem in the target environment. In Sec. 4, we propose the system to solve the problem. A summary of the paper is presented in Sec. 5.

## 2 Related Work

Hadoop Distributed File System is designed for tackling a hardware failure and to be applicable with the commodity hardware [1]. A single cluster of HDFS consists of namenode and datanode. A client request the data read operation to the namenode, and then the namenode will look for the node which stores the data in the datanode. In the same way, whenever there is a data write operation, client send the data to the namenode, then it will be distributed to the available datanode inside the cluster by the namenode.

## 3 Target Environment and Problem Definition

### 3.1 Target Environment

This paper is targeting agriculture warehouse management system using wireless sensor networks. It consists of large number of warehouses and a single server. Each room inside the warehouse has lots of sensor nodes for monitoring the temperature, humidity, and $CO_2$ and a sensor hub for sending the sensing data to the server. All of the sensing data from each warehouse will be stored in the server for further historical data analysis. To be able to communicate with the hub, sensor node uses a short-range wireless technology and the hub uses 3G technology to communicate with the server.

### 3.2 Problem Definition

Processing stream data using existing distributed parallel processing brings difficulties for the system since the existing framework only designed for batch processing mechanism which is somewhat different with the stream data characteristic. Stream data arrives to the server continuously from the sensor hub. And the amount of data can be abundant which can lead very high frequency of data access to the file system. This will lead into performance degradation for the existing system since it processes and stores individual sensing data one by one as it arrives in the server.

## 4 Solution

### 4.1 System Architecture

The system architecture of distributed sensor stream data input system consists of 4 components which are:

1. **Data Measurement Component** measures sensor data generation rates per time slice and defines the amount of data transmitted.
2. **Storage Component** consists of two temporary storages and stores the stream sensor data collected from Data Measurement Component. At this time, if one of temporary storages goes on to move transfer phase, the other collects the stream sensor data and controls the system for preventing the acquisition and transmission.
3. **Configuration Control Component** enters maximum waiting time to collect data from user.

4. **Data Manager Component** is composed of two processors: Data Combination Processor and Data Transmission Processor, for combining and transmitting data to HDFS. The system check the volume of the collected sensor stream data in real time and accordingly, when the volume reaches the predefined chunk size or passed over a certain time limit, the collected sensor data are transformed into a chunk and transmitted to HDFS. Eventually, we consider two ways depending on the amount of generated sensor data per time slice, as follows.

   a. **Huge Amount:** The default chunk size is 64MB, if an enormous amount of the stream data per time slice is generated, then the system will transform the stream data into a default sized chunk (64MB) or over (>64MB). It preserves the performance of the system by preventing frequent file system access.
   b. **Small Amount:** For real time data processing, user firstly set the length of the time slice to collect the data. Right after passing the predefined time slice, the system will combine the collected data and transmits the combined data to HDFS.

### 4.2 Processing Steps

There are seven steps for processing the input stream:

1. **Configure basic information:** User enters basic configuration, such as: HDFS IP address, and maximum waiting period before the data collection begins.
2. **Define data size:** The system firstly measures sensing data generation rate per time slice through data measurement component and defines data size to be transmitted at a time.
3. **Collect sensing data:** The data to be transmitted from sensor nodes are entered to a temporary storage.
4. **Check sensing data size:** The data measurement component compares the size of sensing data and configuration value. If the sensing data size is smaller than the configuration value, then directly move to *Combination phase*, step 6. Otherwise, move to next step.
5. **Check waiting time:** Check whether the waiting time has been exceeded. If it is already exceeded, then move to next step. Otherwise, move back to *Collecting sensor data phase*, step 3 and repeat the steps.
6. **Combine data:** This phase combine sensing data before being transmitted to HDFS.
7. **Transmit data:** In this phase the data will be transmitted to HDFS through data transmission processor in data manager component.

The following figure 1 depicts the seven steps explained above.

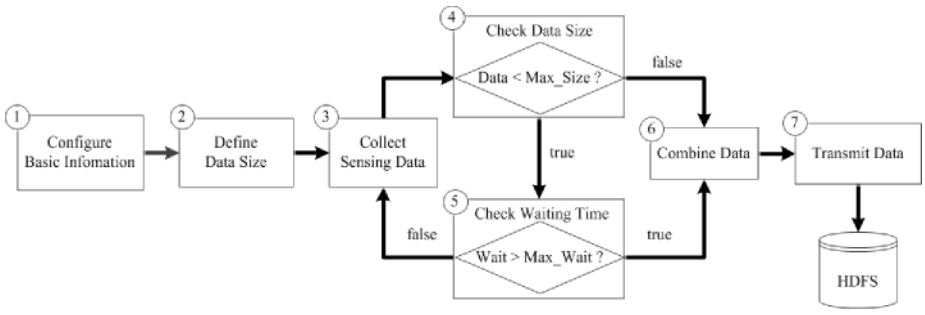


**Fig. 1.** Sensor stream data input processing.

## 5 Conclusion

A new sensor data input system is proposed in this paper. It is done by collecting and combining a single sensing data into predefined-size file. Therefore, the frequency of data access into the file system can be reduced, so that the system will perform well while handling an enormous amount of sensor stream data. First, the sensing data is collected into a buffer, then checked whether the size is already big enough to fit the predefined chunk size. If the collected data is smaller than the chunk size, then the system will wait until it exceed the predefined waiting time. The combining process will only begin after the collected data fits into the chuck size or the waiting time has already been exceeded.

## Acknowledgments

This research was supported by Bio-industry Technology Development Program, Ministry for Food, Agriculture, Forestry and Fisheries, Republic of Korea.

## References

1. The Apache Software Foundation, HDFS 0.21Documentation,http://hadoop.apache.org/ hdfs/docs/r0.21.0/.