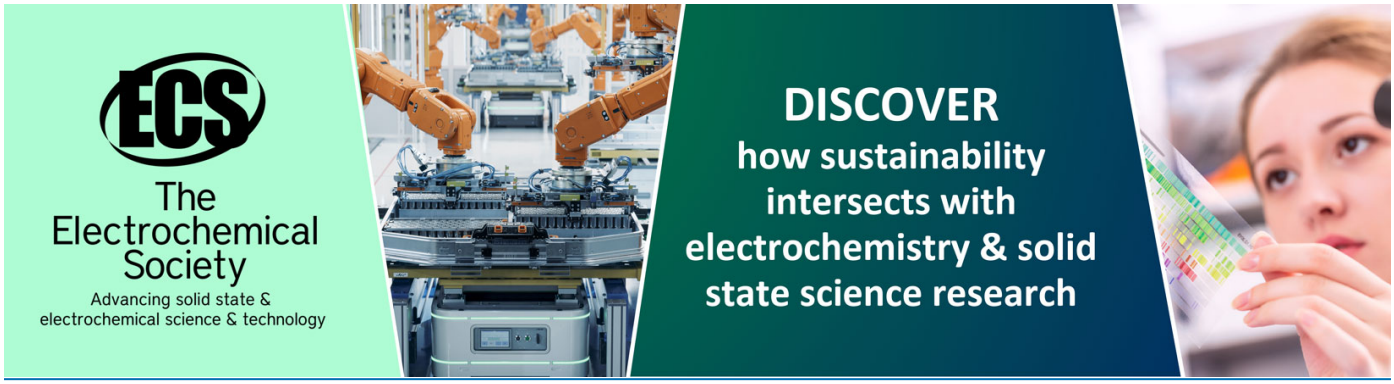
Indoor air quality analysis based on Hadoop



Indoor air quality analysis based on Hadoop

Wang Tuo[1],SUN Yunhua[2] ,TIAN Song[2], YU Liang[1] and CUI Weihong[1]

[1]Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100101, China
[2]China University of Mining & Technology, Beijing 100083, China

Email:tuow1105@gmail.com

Abstract. The air of the office environment is our research object. The data of temperature, humidity, concentrations of carbon dioxide, carbon monoxide and ammonia are collected peer one to eight seconds by the sensor monitoring system. And all the data are stored in the Hbase database of Hadoop platform. With the help of HBase feature of column-oriented store and versioned (automatically add the time column), the time-series data sets are bulit based on the primary key Row-key and timestamp. The parallel computing programming model MapReduce is used to process millions of data collected by sensors. By analysing the changing trend of parameters' value at different time of the same day and at the same time of various dates, the impact of human factor and other factors on the room microenvironment is achieved according to the liquidity of the office staff. Moreover, the effective way to improve indoor air quality is proposed in the end of this paper.

1. Introduction and related works

With continuous development in modern society, human production efficiency has been greatly improved. New scientific products make human's life better than before. However,they also bring damage to the natural environment, such as vehicle exhaust, various factory emissions. It has been found out that this phenomenon is especially serious in developing countries. Admittedly, many countries have built their own controlling network for atmosphere quality in urban areas. They do so no more than to macroscopically monitor outdoor air and research on indoor air is still not that available.

It is estimated that 70%-90% of the time of a person's life is spent indoors [1], so the quality of indoor air has a direct influence on a person's health. Indoor environment includes work environment (office) and living environment, for the former, one may spend 40% of the indoor time in. Thus, we select work environment as the study object.There are some computers, tables, chairs and air-conditions in the selected room. And about 6-13 persons work in the room. With the object to create a comfortable environment, temperature, humidity, CO_2 produced by human metabolism, ammonia released by building materials and CO of indoor are selected as monitoring objects. MapReduce is used to analyse the time series. The results will be visualized. The paper ends up with the analysis on the reasons for the indoor air quality change.

The rest of this paper is organized as follows. In section 2, the HBase and the storage design of air quality data are presented. In section 3, the crosswise and vertical comparison with these time series based on MapReduce is presented. In section 4, ana;ysis on the reasons for this results are carried out and followed by conclusions in section 5.

2. HBase and the design of data storage

Each sensor of our indoor air quality monitoring system is designed to send one record every three seconds, causing a total number of 144 thousands records every day of the five sensors. However, the traditional mode poses much difficulty in data processing and storage expansion, because the data center will receive hundreds of millions records after running long time. Traditional processing method compresses data sequence by sampling, and analyzes and forecasts the trend with partial data. However, this sampling method is designed appropriately based on the desired questions in advance and the sampling results are only used for certain questions. With the development of sensors and big data, data could be completely collected and stored with new technologies. By analyzing these relevant data in-depth from different views, different problems can be solved other than doing random samplings which can only solve certain problem. So HBase is chosen to store the increasing data.

Hbase is an open source, non-relational, distributed database modelled after Google's BigTable and is written in Java [2-4]. It is a database with high reliability, high performance, column storage, scalable characteristics based on the HDFS distributed file system. Its goal is the hosting of very large tables -- billions of rows X millions of columns ---atop clusters of commodity hardware. HBase meets the mass data records storage requirement of air quality monitoring system after long time running and the reading and writing speed of HBase is faster than that of HDFS.

HBase is a database based on column storage. The table in HBase is a sparse persistent storage multi-dimensional, sorted mapping table. The index of the table is Rowkey, columns family and timestamp. Each table contains a series of row records. Rowkey is the unique identifier of a row in the table. The table is organized as key-value. A {row, column, version} tuple exactly specifies a cell in HBase. Cell content is interpreted bytes.

Table 1. The CO_2 table in HBase

Rowkey	timestamp	column family		
		value	type	unit
201206010834		390	CO_2	ppm

3. Air quality data analysis based on MapReduce

MapReduce is proposed by Google Labsa. It is a programming model for processing large data sets [5]. MapReduce is typically used to do distribute computing on clusters of computers. The model is inspired by the map and reduce functions commonly used in functional programming, although their purpose in the MapReduce framework is not the same as their original forms.

The dataset which contains 65 day's data are analysed in the experiment. The steps are as follows: (1) divide the dataset into two groups: weekdays and weekends. Since nobody works at weekends, and the air flows a long time at night, the concentration of each gas of the air in indoor and outdoor varies not much. The data of weekends are assumed to be the natural state as the reference group. (2) compare weekdays' data with weekends' data, find the start and end position of gas or temperature changes, remove the data whose position are before the start position and after the end position, reserve the weekdays' time series data which are different from those of weekends'. (3) cluster the weekdays' time-series data by similarity measurement , and give data support to the next section.

Take CO_2 data as an example for the following analysis.

Step 1: divide the raw data into two groups: weekdays and weekends

The program scans all the records and extracts the first eight characters of the Rowkey (the date string), and determine whether the current record belongs to the weekends group or not. All the records are written into two files (weekdays and weekends). Each file contains all the records belonging to the file group. Every row of the file is a time series of a day.

Map Stage: input :< file name, file context> output< Bool value, Rowkey&Value > (bool value: true weekday, false weekend). Reduce Stage: input<bool value, list (Rowkey&Value)>, output<file name&Rowkey, value>. The monitoring data are saved as a matrix.

$$\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix} \quad (1)$$

r_{ij}---the jth record of the ith day of CO_2 concentration

Step 2: compare the data of weekdays with those of weekends, and find out the change position (start and end). The judging standard is the difference between the value of one record in a weekday and the average value of all the records in the weekends at the same time. After that, the changed data are exaracted as input data for the next step (stage).

Map Stage: input :< file name, file context> output :< j,r_{ij} >, Reduce Stage: input <j, list (r_{ij}, Ave(r_{ij})>, output< deviation value, j>. After dictionary sorting, deviation is outputted to the results file. The start and end positions of the change can be easily found based on the sort results.

Ave (r_{ij}): the abbreviation of average value of the weekends' r_{ij}.

The following is a comparison chart of one weekday and the weekend's CO_2 concentration.

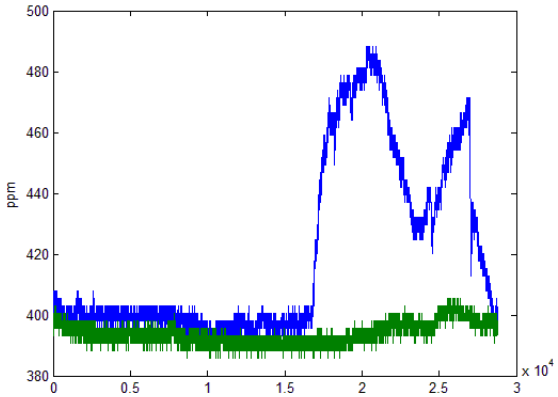


Figure 1. The unprocessed concentration time series

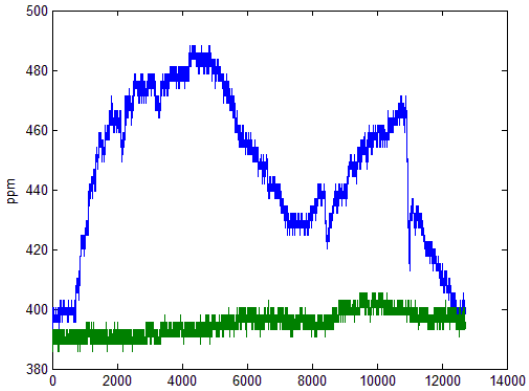


Figure 2. The result of the concentration time series

The blue curve (above) represents a weekday's CO_2 concentration change. And the green curve (below) represents the weekend's CO_2 concentration change.

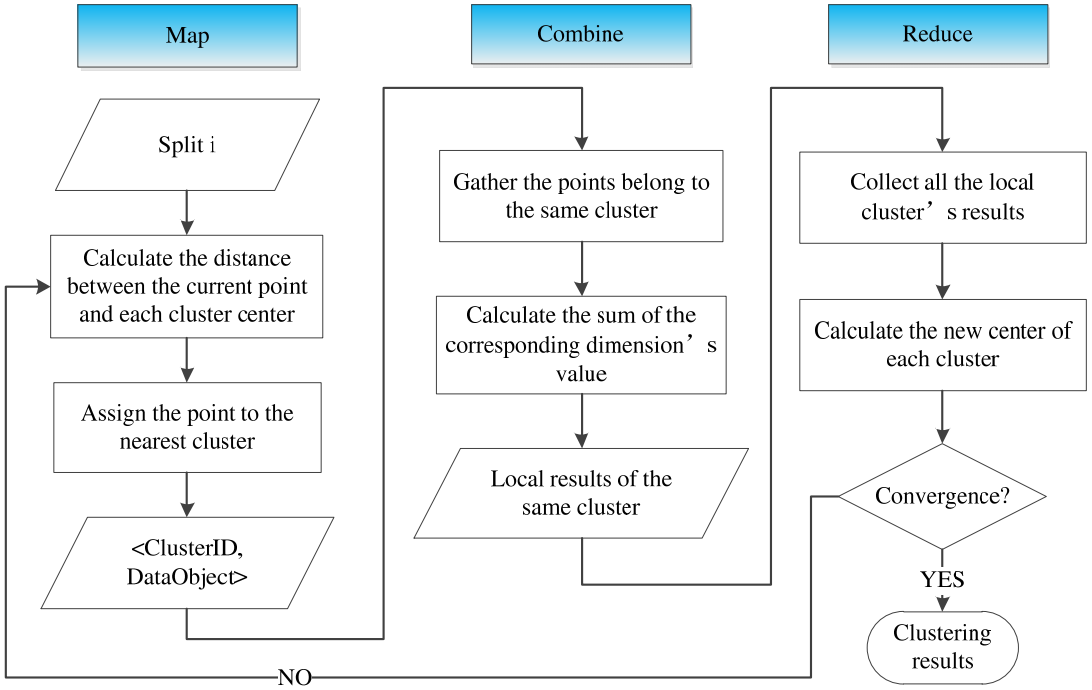


Figure 3.CO_2 concentration time series data processing flowcharts using MapReduce-based K-Means

Step 3: classify the data by similarity measurement

After the second step of the data processing, the weekdays' time series which should be analysed is limited to a relatively small range. In this step, the time series data are mapped to a point in the n-dimensional Euclidean space. In this case, without comparing with the weekends' time series, the similarity measurement of the data is converted to a measurement of the distance of the N-dimensional Euclidean space. All the time series data are classified by K-means clustering method [6].

Map Stage: read the value of the time series data each time a row, convert the data to points of N-dimensional space, calculate the distance between each cluster centre and the point, and assign the point to a specific cluster which is nearest to the point. The output is < cluster ID, time series data>.

In order to reduce the amount of calculation of Reduce stage, the algorithm adds a Combine stage: sum the value of the corresponding dimension. the output is < cluster ID, Sumvaluetimeseriesdata>

Reduce Stage: accept the Combine output of each node, recalculate a new cluster centres, compare with the previous cluster centre points, if they are the same or the offset is controlled within the threshold, output the result to the file, otherwise process repeatedly until convergence or it reaches the set number of iterations.

The data are divided into two sections: morning section and afternoon section, and use K-means clustering algorithm to deal with each data section. The results are as follows:

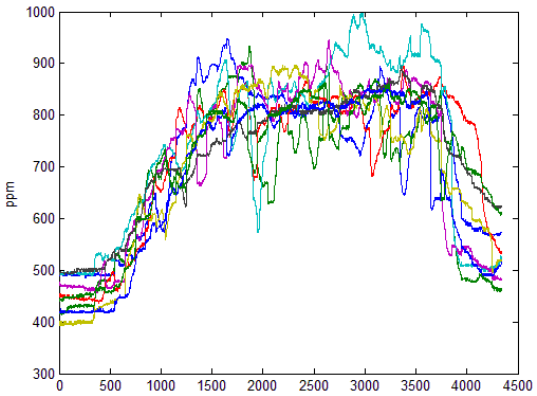


Figure 4. Category 1 of morning section

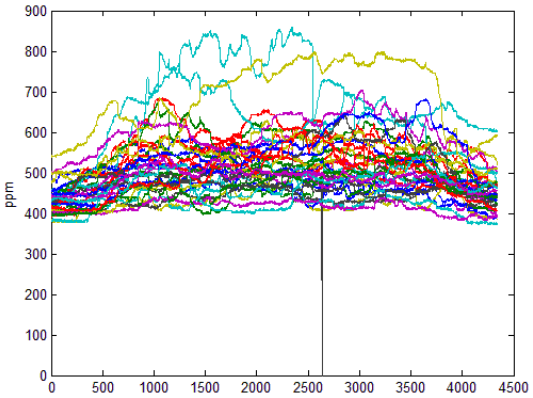


Figure 5. Category 2 of morning section

Table 2. Clustering analysis result of morning section

Category	Date
Category 1	2012/12/14、2012/12/17、2012/12/19、2012/12/20、2012/12/21 2012/12/25、2012/12/27、2012/12/31、2013/01/04
Category 2	2012/06/08、2012/06/11、2012/06/12、2012/06/13、2012/06/14 2012/06/15、2012/06/18、2012/06/19、2012/06/20、2012/08/15 2012/08/16、2012/08/17、2012/08/20、2012/08/21、2012/08/22 2012/08/23、2012/08/24、2012/08/27、2012/08/28、2012/08/29 2012/08/30、2012/08/31、2012/09/04、2012/09/05、2012/09/06 2012/09/10、2012/09/26、2012/09/27、2012/09/28、2012/10/11 2012/10/12、2012/12/18、2012/12/24、2013/01/07

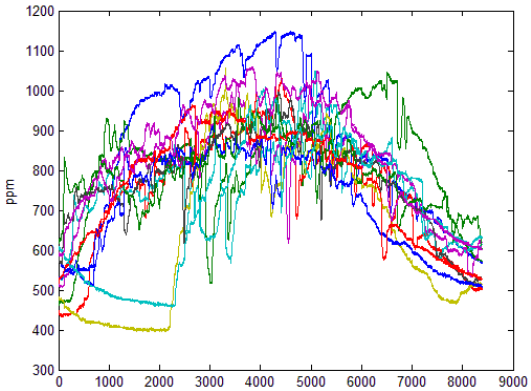


Figure 6. Category 1 of afternoon section

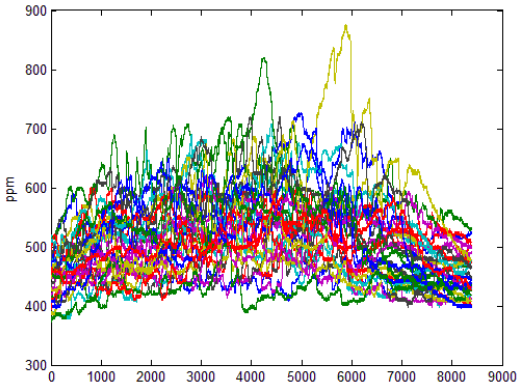


Figure 7. Category 2 of afternoon section

Table 3. Clustering analysis result of afternoon section

Category	Date
Category 1	2012/12/14、2012/12/17、2012/12/18、2012/12/19、2012/12/20 2012/12/21、2012/12/24、2012/12/25、2012/12/27、2012/12/31 2013/01/04、2013/01/07
Category 2	2012/06/08、2012/06/11、2012/06/12、2012/06/13、2012/06/14 2012/06/15、2012/06/18、2012/06/19、2012/06/20、2012/08/15 2012/08/16、2012/08/17、2012/08/20、2012/08/21、2012/08/22

	2012/08/23、2012/08/24、2012/08/27、2012/08/28、2012/08/29 2012/08/30、2012/08/31、2012/09/04、2012/09/05、2012/09/06 2012/09/10、2012/09/26、2012/09/27、2012/09/28、2012/10/11 2012/10/12

4. Analysis of results

It can be readily found that the value of CO_2 concentration included in category 1 change greatly but not so much in category 2. Almost all the monitoring data in category 1 are acquired in December, while the monitoring data in category 2 are mainly from June to October. The CO_2 aggregation effect is more obvious in the winter because air conditionings do not function and the windows are closed. We compare two classification results and find the data of the three days (2012/12/18, 2012/12/24, and 2013/01/07) are abnormal. The value of the data is different from that of the adjacent time, because our team had a meeting in another room in the morning. Measurement equipment is deployed in the table next to the door. When the stuffs get in and out of the door, the data gathered by the sensors fluctuate within a certain range. We tried in vain to find a function to describe the relationship between carbon dioxide concentration and temperature. The cures of the data in category 1(in winter) show that the CO_2 concentration decreases rapidly during the lunch time and after work, which suggests that the residence time of CO_2 is short. The closed windows in winter make the indoor and outdoor air flow slowly. The indoor staff respiration in winter basically unchanged with respect to that in summer. This brings about the high-value level of the CO_2 concentration in winter. Ignoring CO_2 concentration change in the air during a day, it is better to keep the doors and windows on between the office and the corridor or lower the population density of the room to reduce CO_2 concentration of the office.

5. Conclusion

This paper analyses the relationship between the CO_2 concentration and the number of people in the room qualitatively. All the CO_2 concentration of the time series data are stored in HBase. K-means clustering algorithm based on MapReduce is used to analyse the data. According to the classification results of CO_2 concentration, It can be found easily that the direct cause of the indoor CO_2 concentration changes is human respiration. The CO_2 concentration increases with the amount of the people in the room, especially in winter. The residence time of the CO_2 is very short.

Acknowledgment

This work is supported by the National Natural Science Funds of China (NO. 71150001).

References

[1] Liu Nan 2009 *Ecological Economy*. **3** 191-193.
[2]Apache 2012 Apache HBase Reference Guide http://hbase.apache.org/book.html#datamodel
[3]L. George 2011 *HBase: The Definitive Guide* (California: O'Reilly Media)
[4]F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes and R.E. Gruber 2008 *ACM Transactions on Computer Systems (TOCS)*, **26** 4
[5]J. Dean and S. Ghemawat 2008 *Communications of the ACM* **51** 107-113
[6]S. Owen, R. Anil, T. Dunning and E. Friedman 2011 *Mahout in action* (New York: Manning)