MITSAT-An Automated Student Attendance Tracking System using Bluetooth and EyeOS

Srinivas Avireddy†, Prashanth Veerapandian†, Sundaravadanam Ganapati†, Maheshwar Venkat‡,
Prasanna Ranganathan† and Varalakshmi Perumal*
† Student, Department of Information Technology
‡Student, Department of Electronics and Communication Engineering
*Assistant Professor[Senior Grade], Department of Information Technology
Anna University,Chennai,India

***Abstract*—This paper proposes a novel solution MITSAT (Madras Institute of Technology Student Attendance Tracking system). MITSAT provides automation of student attendance tracking using wireless technology such as Bluetooth and scalability using cloud computing. Current day student attendance tracking systems employed in universities require a lot of human intervention. Also its storage for future use consumes a lot of memory. The proposed work identifies the potential use of Bluetooth and EyeOS cloud computing platform to track students' attendance and to efficiently store and retrieve the same. The challenges associated with the state-of-the-art technologies for automation are data heterogeneity, availability, consistency and fault tolerance. The problem with Schema based SQL database is that it is not scalable horizontally and provides poor fault tolerance to partitions. The issue with the schema free NoSQL databases is that it can provide either availability and fault tolerance or consistency and fault tolerance but not all three features together. The proposed work addresses these challenges by combining schema based MySQL and schema free mongoDB databases. The proposed work provides a scalable solution for the bottleneck of storing metadata for small files in Hadoop using a novel correlation based archiving technique.**

Keywords*—*Attendance Tracking; bluetooth; EyeOS; Hadoop

I. INTRODUCTION

Automation has become very essential for porting the legacy systems to the future generation due to the proliferation of THE INTERNET OF THINGs[1]. Current student attendance tracking systems require a lot of human intervention consuming a lot of paper work and time. Existing methodology requires manual tracking and then duplicating it to SQL databases. This can be replaced by automating the process. A wireless tracking system to monitor individual student present inside the campus will save a lot of time. The students can keep track of their individual attendance and take actions accordingly. The professors can track his classs attendance or individual students record with very less time.

According to CAP theorem [2], only two out of the three features, Consistency (C), Availability (A) and Partition tolerance (P) can be provided by distributed cloud based applications. SQL databases provide C and A, but poor tolerance to partitions. NoSQL databases provide either A and P or C and P but not all three. Combining A and C is difficult and NoSQL databases provide eventual consistency [3]. So,to provision all three features, the proposed architecture provides a novel solution of integrating SQL and NoSQL databases. In most of the tracking and monitoring systems, NoSQL databases are employed because the sensor data grows horizontally [4]. But in general it is observed that in most of the applications there are data items which scale vertically requiring schema based normalized SQL tables and some other data items scale horizontally requiring storage in NoSQL databases. State-of-the-art human tracking and monitoring solutions either use SQL or NOSQL databases for all data items. We identify the special requirement of each data item and store them in appropriate type of database thus providing a context aware usage of both SQL and NoSQL database. This helps in overcoming the limitation posed by CAP theorem where MySQL provides Consistency and MongoDB provides availability and fault tolerance.

The other major problem this paper addresses is the challenge faced during the backup of the student attendance information. The files are backed up and stored in Hadoop. Hadoop has HDFS, a Distributed File System which has a single namenode storing the metadata information of all files . Since the attendance information has abundant small files, the metadata storage in the namespace of the namenode leads to serious performance issues [5]. To overcome this challenge, the proposed system provides a novel solution by modifying the inbuilt archiving technique provided by HDFS called harballing [6] by considering the relationship and access patterns of these small files.

The major contributions to this paper are:

- Automating the student attendance tracking using Bluetooth and EyeOS cloud platform.
- Combining MySQL and MongoDB features to overcome problems identified by CAP theorem.
- Addressing the issue of small files in HDFS by adaptive archiving technique based on relationship and file access patterns.
- Evaluating the provided solution for storage efficiency and throughput.

The paper is organized as follows. Section II deals with the problem statement. Section III deals with the related work. Section IV deals with the Architecture and working of the proposed system. Finally section V provides the experimental results of the proposed system.

978-1-4673-5090-7/13/$31.00 ©2013 IEEE

1. Automating the student attendance tracking using Bluetooth and EyeOS cloud platform.
2. Combining MySQL and MongoDB features to overcome problems identified by CAP theorem.
3. Addressing the issue of small files in HDFS by adaptive archiving technique based on relationship and file access patterns.
4. Evaluating the provided solution for storage efficiency and throughput.

The paper is organized as follows. Section II deals with the problem statement. Section III deals with the related work. Section IV deals with the Architecture and working of the proposed system.Finally section V provides the experimental results of the proposed system.

II. PROBLEM STATEMENT

To automate the Student Attendance Tracking process using Bluetooth and efficient storage by integrating MySQL and mongoDB and overcome the issue faced in storing metadata for small attendance files in namenode of Hadoop distributed file system.

III. RELATED WORK

Heterogeneous data collected from sensor sources are used in real time application pertaining to varied fields such as health, education, governance and business planning. The application considered here is Automated Student Attendance Tracking in education institutes.It involves mobile node localization and efficient storage of heterogeneous data. The localization mechanism of mobile nodes involves two steps namely signal measurement and position calculation [7]. Signal Measurement methodologies are classified into methodologies based on Time[8,9] ,Angle Of Arrival[10] and Received Signal Strength[11]. The Time based methods suffer from Time Synchronization problem. Angle Of Arrival methods have problems such as high cost, less precision and they are unsuitable for indoor localization. Received Signal Strength (RSS) overcome the problems of Time based and Angle Of Arrival based methods by measuring signal based on the attenuation produced by the propagation of the signal. We choose RSS because it based on pre-recorded database which can be sampled offline or matched online and does not need the help of reference nodes for Position Calculation. Position Calculation is done based on the signal measurement and co-ordinates of a reference node. Some notable methods available are Trilateration[12], Triangulation[13] or statistics based method such as Maximum Likelihood Estimation(MLE)[14]. The number of reference nodes required to estimate the position are three,two and none for trilateration,triangulation and MLE respectively.

Triangulation and Trilateration require many reference nodes[7] and hence we make use of MLE method for node localization which use RSS for signal measurement and is based on pre-recored database and not reference nodes. The reason is that these techniques can be adopted locally for each classrooms limiting the number of reference node required per room.

Another challenge associated with heterogeneous data is the storage format. Most sensor data storage use either NoSQL or SQL databases for storage. Due to the advent of elastic computing, event based computing and context awareness[15] we provide a novel solution of integrating the use of both SQL and NoSQL databases due to the scalability of heterogeneous data. NoSQL databases can be classified as column oriented ,document oriented and graph oriented[16]. The need for NoSQL database in MITSAT is storing each student information using dynamic allocation and reallocation[17] and hence document oriented mongoDB is chosen. Also mongoDB has several advantages in terms of flexibility, scalability and dynamic failover when compared to other NoSQL databases like Cassandra and couchDB[16].

The Hadoop Distributed File System is limited by the main memory of single namenode for storing metadata for files making it unsuitable for abundant small files and leads to inefficient file access patterns. Archiving Solution provided to mitigate this problem as in [5,6] do not consider relationship between files and access patterns. We overcome that by providing a depth wise archiving technique.

IV. PROPOSED SOLUTION

To understand the functionality of the proposed system we first describe the structure of Sample University and it is shown in Fig.1. Each university consists of several departments. Each Department consists of classrooms each of which is mapped to students of particular year. In general students study 4 year courses hence there are four classrooms needed to accommodate a department. Each classroom has an Access Point (AP) , which tracks the attendance of the class every hour locally and updates the remote server at the end of every working day.

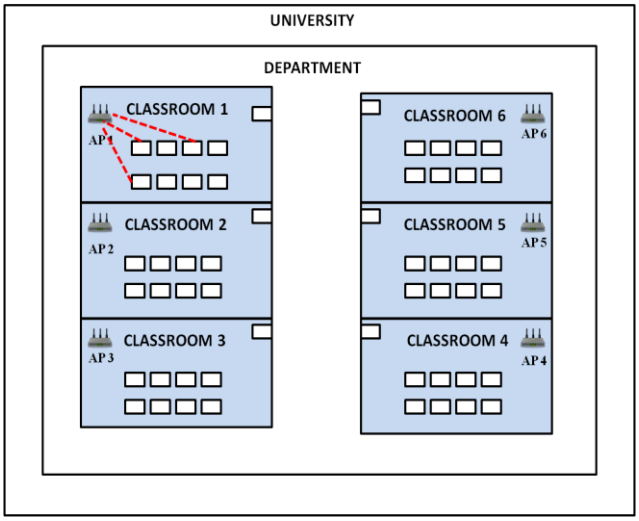

Fig. 1. Classroom Structure

The basic functionality of the proposed system is discussed as follows. Each student has a Bluetooth transmitter which transmits a unique registration ID on pairing itself with the access point in the classroom. Each classroom is provisioned with an access point to collect the student attendance information locally and a remote server is updated about the attendance information of every classroom. This follows a special type of client/server interaction style known as publish/subscribe model[18] or occasionally connected model. This interaction style is asynchronous, event driven and suitable for mobile nodes.

A. *Architecture of MITSAT*

The proposed architecture consists of four layers namely the hardware and sensor layer, the middleware layer, the cloud storage layer and the user interface layer. The architecture is shown in Fig. 2.

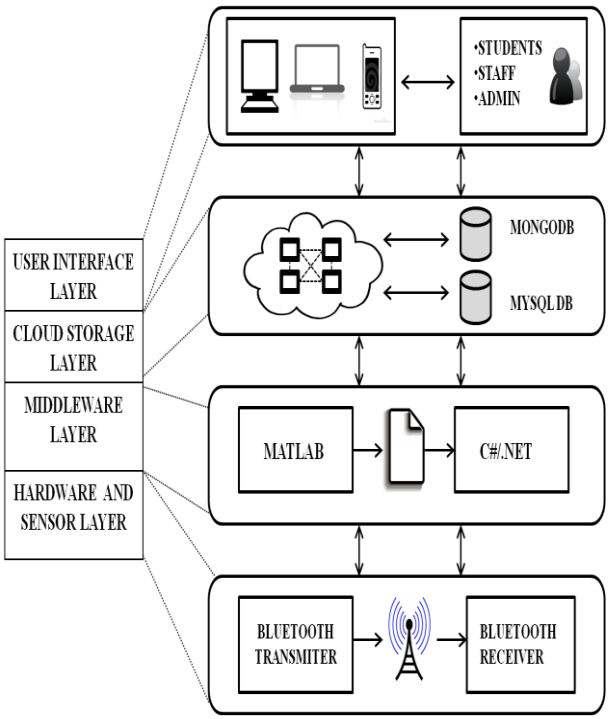

Fig. 2. Layered Architecture of MITSAT

*1) Hardware and Sensor Layer:*This layer consists of hardware and wireless tracking components required for MITSAT. It comprises of two components namely the Bluetooth bee and the Access point. Every student is identified by a Bluetooth bee transmitter registered with a unique ID. The Bluetooth bee Receiver is attached to the access point for tracking the students within the classroom. The Access point can a laptop/Desktop installed with Bluetooth module or it can be Router.

*2) Middleware Layer:*The middleware layer forms the intermediate layer between the sensor layer and the remote server. It consists of two major components namely the MATLAB module and C# module for pre-processing required for local disk storage of student attendance.

*3) Cloud Storage Layer:*The cloud storage layer forms the remote server layer. This layer comprises of the private cloud called MITSAT Cloud built using EyeOS cloud platform and is connected to the MySQL database and MongoDB collection, both serving as backend databases.

*4) User Interface Layer:*This layer form the User interface for the users to access the MITSAT system. The users may be administrators, staff or students. The admin has both read and write access to the database. All other users have only read access.We provide interoperability and making our system accessible using major platforms such as desktop, laptops and mobile phones by developing web application and porting it to different user devices using PhoneGap.

B. *Working*

The MITSAT consists of 4 phases for automating the attendance tracking process. They are sensing phase, pre-processing phase, updating phase and backup phase. These phases are explained in this section with respect to the interaction of the four layers drawn in the architecture diagram shown in Fig.2.

*1) Sensing Phase:*This phase involves the interaction of the hardware and sensor layer and the middleware layer.The layer 4 consists of the Bluetooth Bee Transmitter and receiver. The reason to choose Bluetooth technology is that the Bluetooth bee transmitter can be easily replaced by smartphones in future[19].

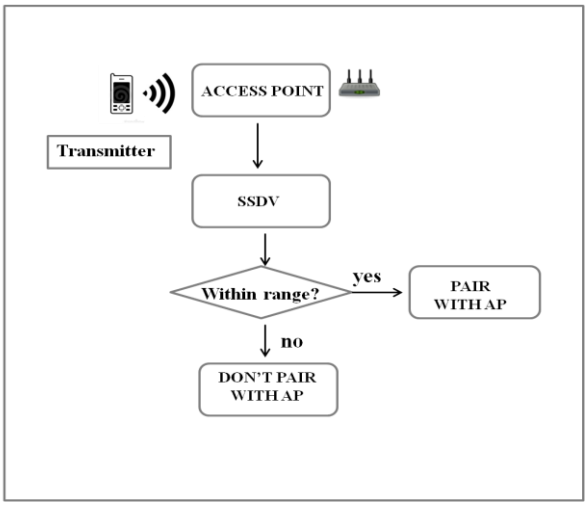

Fig. 3. Sensing Phase

During the sensing phase, once the student enters the classroom the Bluetooth bee transmitter of that student gets paired with the Access Point inside the classroom. The mobile

node localization of the Bluetooth bee transmitter is done using The Signal Strengthen Dynamic Value Scheme based on RSSI [20].If the transmitter is within the range and inside the class it is paired with the access point , the transmitter starts sending the Student ID. This information is sent via serial port communication to MATLAB using the BLUETOOTH API in MATLAB. The workflow of this phase is shown in the Fig.3.

*2) Pre-Processing Phase:*Once the sensing phase is over, the control is now passed on to the middleware layer. This layer has two major functions. One is to interface transmitter with the Bluetooth API in MATLAB and the other to process the student attendance information of every classroom in a particular hour. Once the transmitter gets paired with the Access Point, it starts sending 10 digit Student ID to the MATLAB module.

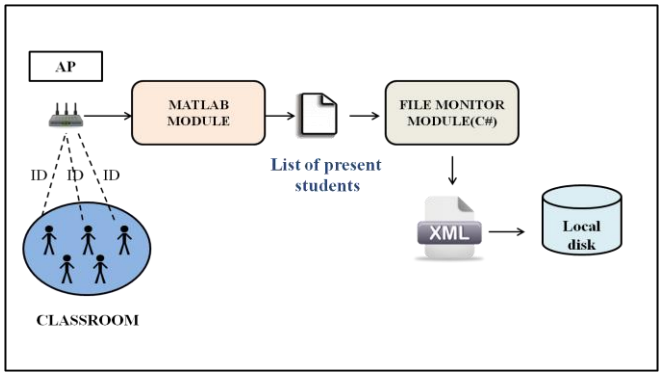

Fig. 4. Pre-Processing Phase

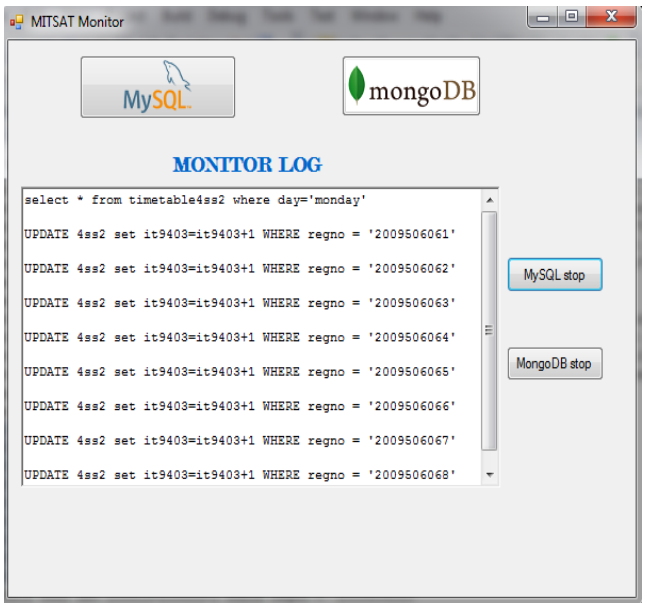

Fig. 5. File Monitor Module

The MATLAB module processes this information and generates a text file containing the list of students present .Once the text file is generated, the file monitor module written in C# language gets triggered and generates a XML file containing the student details along with their attendance information for that day. This updates local disk and sends updates to MySQL database only at the end of the day to avoid frequent round trip to remote server.

The other function of the C# program is to populate the list of absentees per hour in the local disk. At the end of the day when there are no more transactions related to attendance tracking, the NoSQL database MongoDB is updated.The workflow of pre-processing phase is shown in Fig.4 and screenshot of the file monitor module in C# is shown in Fig.5

*3) Updating Phase:*This phase has two steps involved.First step is to update the MySQL database and the next is to update the MongoDB collection.To understand the working of this phase we first describe the structure of tables in MySQL database and also structure of each document of student in attendance collection stored in MongoDB.The database named "Attendance" consists of tables such a class,subject and timetable.The table class stores the number of hours a student is present for each subject.The subject table stores the subject ID,subject name and total number of hours handled for that subject.The Timetable table stores the information that maps each of the 8 hours of classes in a day to corresponding subject ID.

To understand the working of updating the MySQL Database, we explain with a real time scenario.Let us consider a clasroom C around 60 students. When the class starts the student information is tracked by the Access Point and processed by MATLAB module and a XML file is generated containing the list of students present and absent. A sample table after updating is shown in Fig.6. It has a list of Student ID's with the number of hours each student is present for each subject. The ID, '2009506001' before updating has 12 hours present for SUBJECT1 and it gets updated to 13 hours present since his attendance was automatically tracked and updated when he attends a class.

STUDENT ID	SUBJECT 1	SUBJECT 2	SUBJECT 3	SUBJECT 4	SUBJECT 5
2009506001	12	13	14	14	12
2009506002	13	12	13	14	12
2009506003	13	14	13	14	13
2009506004	13	13	12	14	14

Fig. 6. Example for Updating Phase

A sample MongoDB document for a student after updation is shown in Fig.7. It stores Student ID,Name,days absent for each subject.It is important to note that the number

of days present and absent can be found using MySQL database and details about each day absent or present can be found in MongoDB collection since it can grow horizontally without any predefined schema. Our novelty lies in segregating the use of SQL and NoSQL databases based on context and using it appropriately to provide memory efficiency and better response time. This is explained in the section VI. The overall working of this phase is shown in Fig.8.

```
>db.attendance4ss2.findOne()

{   _id: ObjectId("507c4e138fada716c89d0014"),
   name: abcde
   regno:2009506005
   IT9401:[ {day: 24-07-2012}, {day: 28-07-2012} ],
   IT9402:[ {day: 25-08-2012}, {day: 08-09-2012} ],
}
```

Fig. 7. A Sample Student mongoDB Document

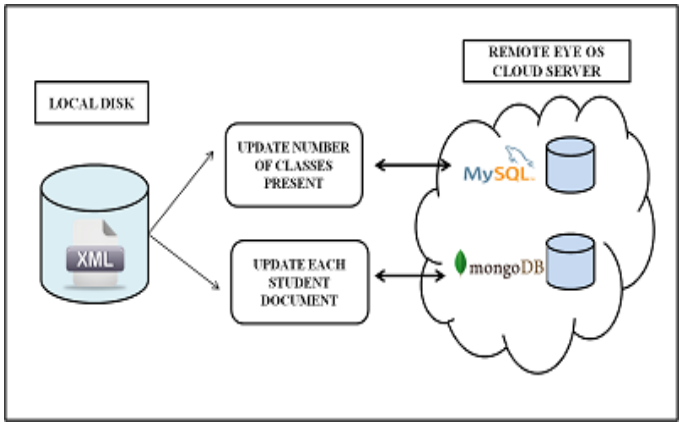

Fig. 8. Updating Phase

4) Backup Phase: After the updating phase, the database gets updated each day and remains consistent. To protect from hardware failures, backup is necessary and this marks the provision of fault tolerance feature.The backup is stored in Hadoop Distributed File System(HDFS). One major issue associated with HDFS is the limitation of memory of single namenode for metatdata storage.Since the files associated with attendance storage are small files ,storing it's metadata in the namespace of single namenode is a bottleneck. Hence to overcome this issue we provide an improved inbuilt archiving technique called harballing technique to archieve the metadata of small files.In general harballing for small files is not done based on any co relation between the files and access patterns. Since the frequent file access pattern in intra level of the tree is more frequent than inter levels, we archive the metadata information of each level into one, thus improving the usage of the harballing technique. We build relationship between these small files based on the tree structure as shown in Fig.9.Based on the tree we build a directed graph with the individual files forming the nodes of the graph and edges form the relationship between these files.the adjacency matrix shown in the fig depicts this depth wise or level wise inter relationship between these files. Thus using depth wise harballing technique provides reduce overhead on the namenode and prevents the small file metadata storage problem.Storing in HDFS provides not only availability but also fault tolerance.

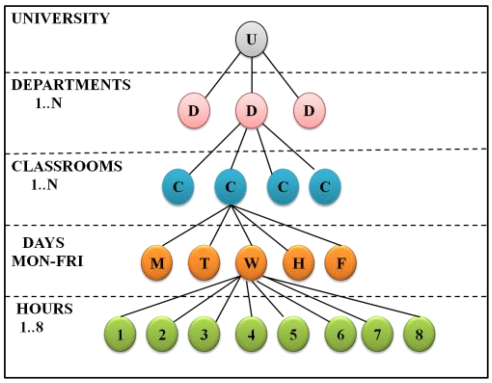

Fig. 9 .Tree Structure of Attendance Files

V. EXPERIMENTATION AND RESULTS

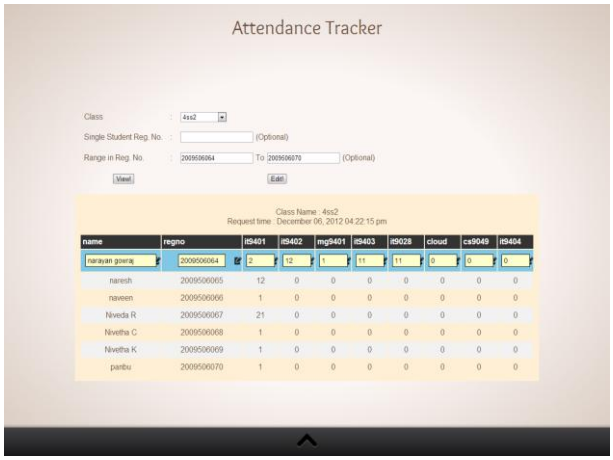

Fig. 10. Screenshot of Web application

The implementation screenshots of the User Interface(UI) of MITSAT provided to the users are shown in Fig. 10 and Fig. 11. Fig. 10 shows the UI of the web application built using EyeOS[21] and Fig. 11 shows the UI of android application developed using PhoneGap[22] to support all forms of user devices providing interoperability and platform independence.Three departments Instrumentation(E&I), IT and CSE each having 60,120 and 180 students respectively are considered. These departments' student attendance information are considered to evaluated the efficiency of the proposed system.The parameters considered

for evaluation are memory optimization, query execution time and metadata storage optimization.

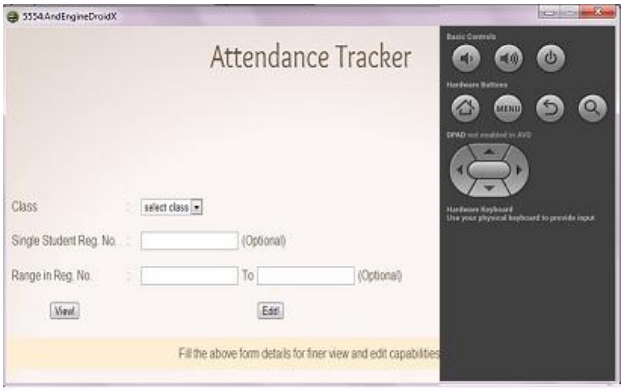

Fig. 11. Screenshot of Android Application

1) Memory Optimization:

For all statistical data of students like number of hours present and absent can be stored in MySQL table. But, horizontally scalable attributes of students such as details of list of dates absent and present can be stored and retrieved efficiently using mongoDB. The graph shown in Fig.12 shows that there is a 60% more efficiency by integrating MySQL and mongoDB databases than using only MySQL database for horizontally growing data attributes like student details.

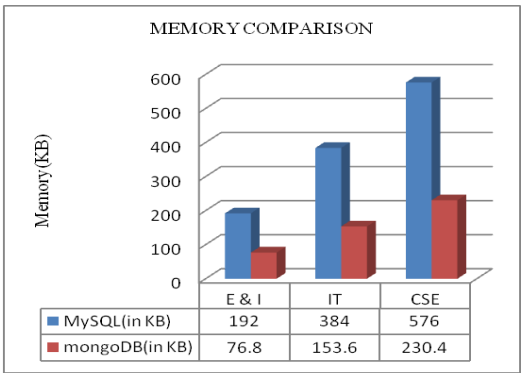

Fig. 12 Comparison of Memory usage between MySQL and mongoDB

2) Query Execution Time

For simple queries without involving joins and aggregation it is found that the Query Execution time is in the ascending order of MySQL with indexed attributes,MySQL without indexed attributes, mongoDB with indexed attributes and mongoDB without indexed attributes.Here MySQL with index has lowest execution time and mongoDB without index has the highest execution time. It is shown in the Fig.13.

For complex queries involving aggregation and multiple table joins it is found that the query execution is in the ascending order of mongoDB with index,mongoDB without index, MySQL with index and MySQL without index. Here mongoDB with index has the lowest execution time and MySQL without index has the highest execution time.This is shown in Fig.14.

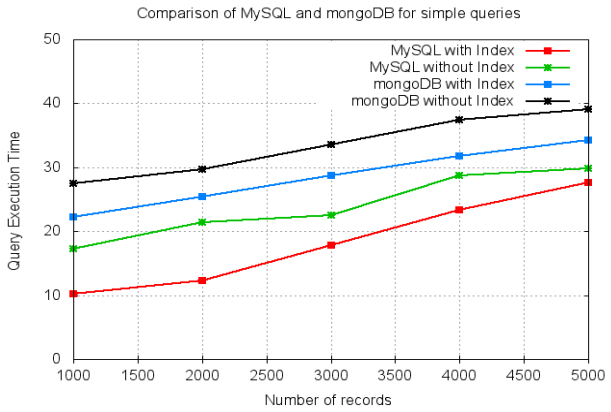

Fig. 13. Execution Time for simple queries

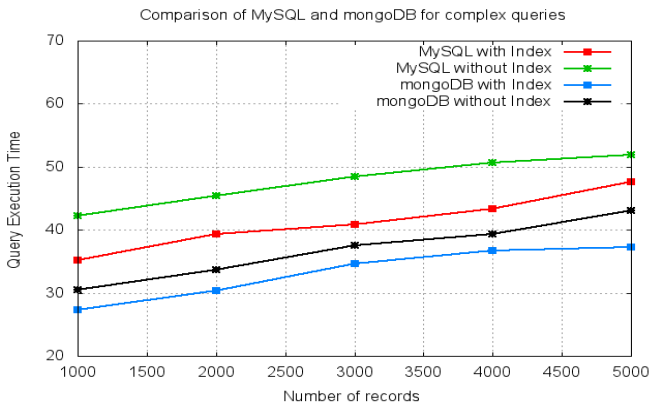

Fig. 14. Execution Time for Complex Queries

3) Metadata Storage Efficiency:

By applying the depth wise harballing technique for the backup files of student attendance information in HDFS, it is found that the metadata storage requires 50% less storage than when the depth wise harballing technique is not employed. It is plotted in the graph shown in Fig.15.

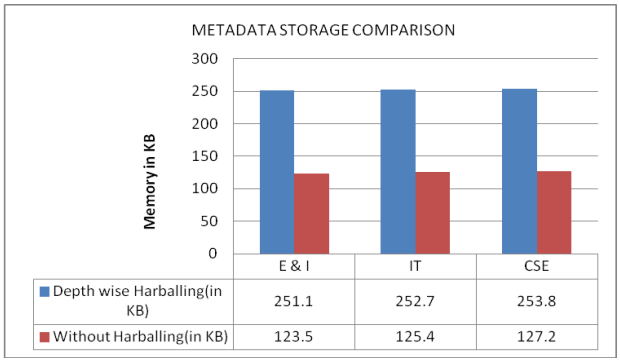

Fig. 15.Metadata Storage Efficiency

VI. Conclusion and Future Work

The proposed novel Student Attendance Tracking System MITSAT is automated using Bluetooth. Memory Efficiency of 60% is obtained by integration of SQL and NoSQL databases. The Query Execution time analysis is also done for both MySQL and mongoDB. Also the proposed depth wise harballing technique also provides 50% lesser metadata storage. In our future work we would like to integrate other details of the students such as fees payment and exam mark automation into the existing system.

References

[1] van der Veen, J.S.; van der Waaij, B.; Meijer, R.J.; , "Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual," *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on* , vol., no., pp.431-438, 24-29 June 2012.

[2] E. A. Brewer, "Towards robust distributed systems," *Symposium* on Principles of Distributed Computing, 2000.

[3] W. Vogels, "Eventually consistent," Communications of the ACM, 2009.

[4] Jianwen Wei; Yusu Zhao; Kaida Jiang; Rui Xie; Yaohui Jin; , "Analysis farm: A cloud-based scalable aggregation and query platform for network log analysis," *Cloud and Service Computing (CSC), 2011 International Conference on* , vol., no., pp.354-359, 12-14 Dec. 2011

[5] Xuhui Liu; Jizhong Han; Yunqin Zhong; Chengde Han; Xubin He; , "Implementing WebGIS on Hadoop: A case study of improving small file I/O performance on HDFS," *Cluster Computing and Workshops, 2009. CLUSTER '09. IEEE International Conference on* , vol., no., pp.1-8,Aug.31,2009-Sept.4,2009.

[6] Mackey, G.; Sehrish, S.; Jun Wang; , "Improving metadata management for small files in HDFS," *Cluster Computing and Workshops, 2009. CLUSTER '09. IEEE International Conference on* , vol., no., pp.1-4, Aug. 31 2009-Sept. 4 2009.

[7] Da Zhang; Feng Xia; Zhuo Yang; Lin Yao; Wenhong Zhao; , "Localization Technologies for Indoor Human Tracking," *Future Information Technology (FutureTech), 2010 5th International Conference on* , vol., no., pp.1-6, 21-23 May 2010.

[8] C. Falsi, D. Dardari, L. Mucchi, M. Z.Win, "Time of Arrival Estimation for UWB Localizers in Realistic Environments", *EURASIP Journal onApplied Signal Processing*, 2006, Article ID 32082, 13 pages.

[9] Y. Takabayashi, T. Matsuzaki, H. Kameda, and M. Ito, "Target Tracking Using TDOA/FDOA Measurements in the Distributed Sensor Network", *SICE Annual Conference*, August 2008 , Japan. Pages: 3441-3446.

[10] Chunhua Yang, Yi Huang, Xu Zhu, "Hybrid TDOA/AOA method for indoor positioning systems", *IET Seminar on Location Technologies*, Dec.2007, Pages:1-5.

[11] Jie Zhan, Hongli Liu, Bowei Huang," A New Algorithm of Mobile Node Localization Based on RSSI," *Wireless Engineering and Technology*,2011,2,112-117.

[12] G. Han, D. Choi, W. Lim, "A Novel Reference Node Selection Algorithm Based on Trilateration for Indoor Sensor Networks", *IEEE Intl Conf. on Computer and Information Technology, 2007*,Pages:1003-1008.

[13] Hossain, I.S.; Ariffin, S.H.S.; Fisal, N.; Hassan, N.S.A.; Latiff, L.A.; Choong Khong Neng; , "Performance analysis indoor location tracking framework with SIP on IPv6," *Modeling, Simulation and Applied Optimization (ICMSAO), 2011 4th International Conference on* , vol., no., pp.1-7, 19-21 April 2011

[14] Xianzhong Tian, Yonggang Miao, Tongsen Hu, Bojie Fan, Jian Pan,Wei Xu, "Maximum Likelihood Estimation Based on Time Synchronization Algorithm for Wireless Sensor Networks", *ISEC International Colloquium on Computing, Communication, Control, and Management* (CCCM),August2009,Pages:416-420.

[15] http://www.ics.uci.edu/~singhv/Publications_files/TechReport-EventShop.pdf.

[16] Chad DeLoatch and Scott Blindt, "NoSQL Databases: Scalable Cloud and Enterprise Solutions", August 2, 2012.

[17] Shidong Huang; Lizhi Cai; Zhenyu Liu; Yun Hu; , "Non-structure Data Storage Technology: A Discussion," *Computer and Information Science (ICIS), 2012 IEEE/ACIS 11th International Conference on* , vol., no., pp.482-487, May 30 2012-June 1 2012.

[18] Abdullahi, M.B.; Guojun Wang; , "Secure Publish-Subscribe-Based In-Network Data Storage Service in Wireless Sensor Networks," *Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on* , vol., no., pp.297-299, 16-18 May 2012.

[19] Hernandez-Capel, C.; Belmonte-Carmona, A.; Roca-Piera, J.; Alvarez-Bermejo, J.A.; , "Integrated architecture to track and organize tasks in corporations and institutions," *Next Generation Web Services Practices (NWeSP), 2011 7th International Conference on* , vol., no., pp.158-163, 19-21 Oct. 2011.

[20] http://www.eyeos.com/

[21] http://phonegap.com/

,