ORIENTAL JOURNAL OF
COMPUTER SCIENCE & TECHNOLOGY
An International Open Free Access, Peer Reviewed Research Journal
Published By: **Oriental Scientific Publishing Co., India.**
www.computerscijournal.org

ISSN: 0974-6471
March 2013,
Vol. 6, No. (1):
Pgs. 111-117

Data Analysis and Management Techniques in Wireless Sensor Networks

T. RAVI KUMAR[1] and K. RAGHAVA RAO[2]

Scholar, Associate Professor in CSE, SBIT, Khammam, India.
Professor in CSE, KL University, Vaddeswaram, Guntur, India.

(Received: January 01, 2013; Accepted: January 15, 2013)

ABSTRACT

Harvesting the benefits of a sensor-rich world presents many data analysis and management challenges. Recent advances in research and industry aim to address these challenges. Modern sensors and information technologies make it possible to continuously collect sensor data, which is typically obtained as real-time and real valued numerical data. Examples include vehicles driving around in cities or a power plant generating electricity, which can be equipped with numerous sensors that produce data from moment to moment. Though the data gathering systems are becoming relatively mature, a lot of innovative research needs to be done on knowledge discovery from these huge repositories of data. The data management techniques and analysis methods are required to process the increasing volumes of historical and live streaming data sources simultaneously. Analysts need improved techniques are needed to reduce an analyst's decision response time and to enable more intelligent and immediate situation awareness. Faster analysis of disparate information sources may be achieved by providing a system that allows analysts to pose integrated queries on diverse data sources without losing data provenance. This paper proposed to develop abstractions that make it easy for users and application developers to continuously apply statistical modeling tools to streaming sensor data. Such statistical models can be used for data cleaning, prediction, interpolation, anomaly detection and for inferring hidden variables from the data, thus addressing many of the challenges in analysis and managing sensor data. Current archive data and streaming data querying techniques are insufficient by themselves to harmonize sensor inputs from large volumes of data. These two distinct architectures (push versus pull) have yet to be combined to meet the demands of a data-centric world. The input of sensor streaming data from multiple sensor types further complicates the problem.

Key words: Sensor, Streaming, Historical, Data management, Analysis and data mining

INTRODUCTION

Sensor networks can be used to capture information about different aspects of an environment, by collecting readings that are conveyed over (typically) wireless networks to base stations for further analysis or action. Sensor data management supports the collection, analysis, integration and use of sensed data. Sensor data management is challenging for a range of reasons. For example, identifying which sensors to use and where to place them to support a particular data analysis requirement is by no means straightforward, and continuing to provide a reliable service when some sensor nodes have failed is important in a context where failures are common. And we have ongoing interests in the following topics: meeting quality of service requirements in sensor query processing; resilient query processing in sensor networks; and integrating complex analyses and query processing in sensor networks. Sensor database systems[1-3] attempt to fulfill this need. In fact, the reason of this evolution is similar to that of emergence of database systems a couple of decades ago, i.e. transition from application dependant data files to the application independent databases. The database community has been investigating new data management techniques[4-6] such as continuous queries, in-network aggregation, approximate query answers, and resource sharing. However, in spite of the fact that reliable transaction processing is the core functionality of a database management system (DBMS), transactional aspects are not explored for sensor database systems. Now that vast amounts of sensor data are being collected automatically, we face scalability problems in data analytics. In general, the problems of massive data analytics are twofold. The first one is how to compress the data effectively, without losing essential features of the data. The second one is how to optimize the runtime environments for analytic computer systems. This paper aims to expand the open questions mentioned above, and tries to find some answers. It particularly deals with concurrency control problem which arises with coexistence of update transactions and continuous queries. Firstly, section 2 gives our vision of large scale sensor database systems, and introduces different sensor transactions that can exist in these systems. The relevance of traditional ACID properties.

Sensor Data Management

Today's sensor motes (e.g: Horton et al. 2002) are full fledged computer systems, with a CPU, main memory, operating system and a suite of sensors. In the Tiny DB system (Madden et al. 2002; 2003; 2005), users connect to the sensor network using a workstation or base station directly connected to a sensor designated as the sink. Aggregate queries over the sensor data are formulated using a simple SQL-like language, then distributed across the network. Aggregate results are sent back to the workstation over a spanning tree, with each sensor combining its own data with results received from its children. As in conventional relational database systems, sensor data is represented by tuples which conform to a data schema. Queries are formulated according to that schema. Mostly, queries pertain to three parts of sensor data: meta-information of sensor (identification, location, type, unit of measurement.), sensor's measurement (temperature, pressure, GPS coordinates, RFID tags Id, etc.), and timestamp of the measurement. Continuous query operators execute on sensor measurement (e.g. sensors measuring less than 10).Sensor stream data is represented by a virtual table called measures. Queries are formulated according to a common global schema. A simple schema example could be as sensor stream = < sensorId,location,type,rate, unit, measurement, timestamp >.Sensor networks generate multidimensional data streams. Each stream has some common metadata, such as the organization responsible for the deployment (and basic facts about the deployment design), sensor type, location and physical context, calibration parameters, precision, accuracy, and maintenance history. However, the bulk of the information is often a time series of measurements (granted, we must take a liberal view of the term "time series"; in the case of camera networks, the observations taken in time are images). In one common operational model, a sensor reports a measurement averaged over a given time period; the measurement area (or measurement point) is often fixed and promoted to the metadata. So when asking who-what-when-where-why, the data stream is a where-when-what array or for fixed instruments, a when-what array. In actuated networks, sensors report data only when they have detected an event.

A database approach to managing data collected on sensor networks has been advocated [Yao and Gehrke 2002; Madden et al. 2005], with particular attention paid to efficient query processing for aggregate queries [Madden et al. 2002; Yao and Gehrke 2002; Zhao et al. 2003]. Sensor data base management systems [SDBSs]

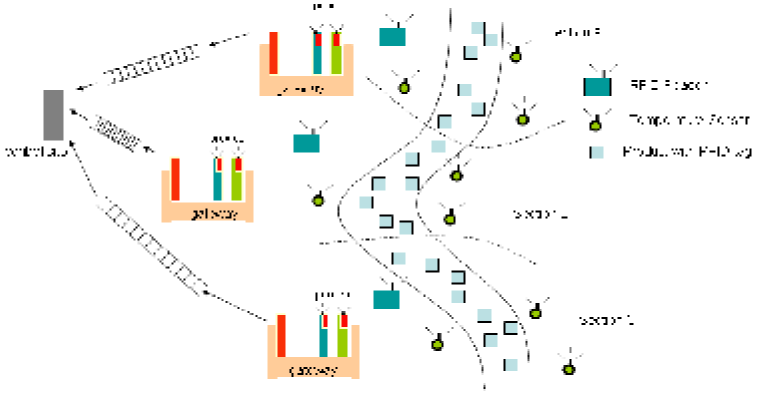

Fig. 1: Architecture and application scenario

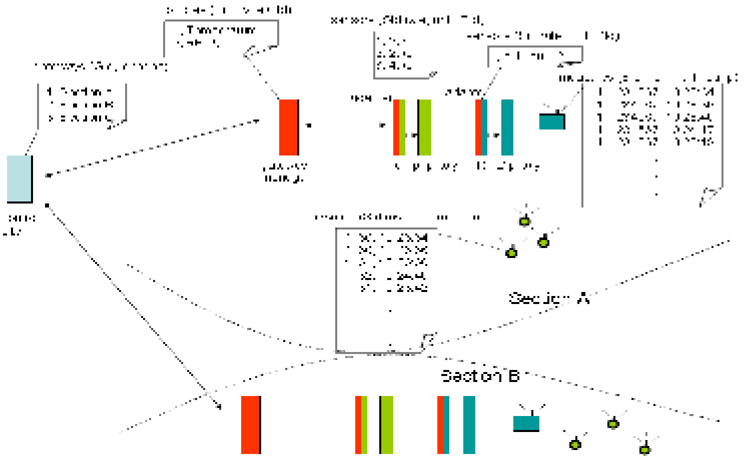

Fig. 2: Schema examples

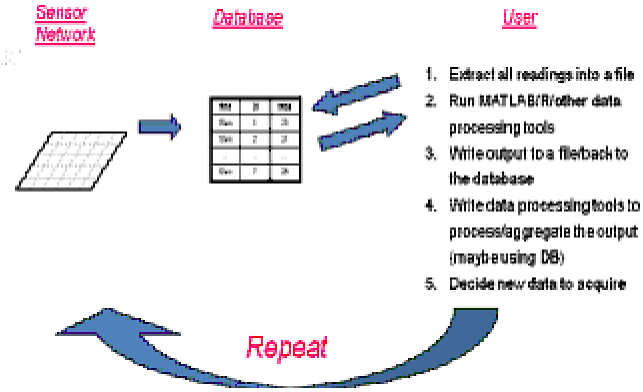
Sensor Network
Database
User
1. Extract all readings into a file
2. Run MATLAB/R/other data processing tools
3. Write output to a file/back to the database
4. Write data processing tools to process/aggregate the output (maybe using DB)
5. Decide new data to acquire
Repeat

Fig. 3: Sensor Data Processing

and RTDBSs have to deal with temporal issues. SDBSs are mostly conceived for monitoring sensor data in "quasi-real time", timely deliverance of sensor data gains importance. We believe that concurrency control protocols proposed for RTDBSs can certainly have reusable aspects for transactional sensor data management. In RTDBSs, optimistic protocols are mostly preferred against blocking protocols in order to deal with temporal constraints[7-9].

Data Aggregation

Data aggregation is defined as the process of aggregating the data from multiple sensors to eliminate redundant transmission and provide fused information to the base station. Data latency and accuracy are important in many applications such as environment monitoring, where the freshness of data is also an important factor. It is critical to develop energy-efficient and fast data-aggregation algorithms. Aggregation can be done in two approaches.

Direct delivery approach

Each sensor node sends sensed value to the sink. The Sink after receiving all messages computes the aggregated value. In fig 2a each node is labeled with the total number of hops/ messages needed to reach the sink.

Aggregation operators

Authors of TAG [14] classify aggregates according to four properties in Figure 3.

Duplicate sensitivity

Duplicate sensitive aggregates will change when it receives a duplicate reading from a single device. transmitted to the sink. This improves the energy efficiency of the network. In the rest of this subsection, we describe the different hierarchical data-aggregation protocols and highlight their advantages and limitations.

Sensor Data Analysis

Modern sensor and information technologies make it possible to continuously collect sensor data, which is typically obtained as real-time and real-valued numerical data. Examples include vehicles driving around in cities or a power plant generating electricity, which can be equipped with numerous sensors that produce data from moment to moment. Though the data gathering systems are becoming relatively mature, a lot of innovative research needs to be done on knowledge discovery from these huge repositories of data. This project focuses on developing knowledge discovery methodologies mainly for real-valued data generated in manufacturing industries such as the automotive and other heavy industries. to deal with raw sensor data—data that often requires a steep learning curve for interpretation, such as accelerometer data or 3D-coordinate location data[10] . Krause et al. describe an approach to learning context-dependent personal preferences using machine learning techniques to refine the behavior of Sensors, a context-aware mobile phone[11]. The behavior modifications, such as changing the state of the ringer volume from loud to silent, are known in advance sensors feed a continuous source of data, the collective sensor network is able to generate a map portraying the current situation[12].The key to successful sensor network algorithms is to infer as much coherent information as possible from an evidence base that may be noisy, corrupt, and erroneous. Using probabilistic techniques, this approach is able to account for limited and stochastic information and physical issues with sensor malfunction, providing information for the world state even in locales not directly observed. Sensor Data evaluation encapsulates two key tasks. The first, Evaluating a specification in elation to the data, is necessary in order to make the situation specifications consistent with annotated sensor data[11] multi-sensory analysis are the combined sensor monitoring with radio wave and optical sensors on the earth surface with vegetation and urban land use or sea/ice signature, combined passive and active sensor data both withradiowaves and optical lights for cloud/aerosol and precipitation in the atmosphere, and combined optical and radio wave sensor data analysis of atmospheric trace gases. Multiple satellite sensors are used to analyze physical processes that determine energy fluxes and their interaction at the urban surface. In environmental science, sensors are most commonly used for forecasting or monitoring environmental processes. The observations are usually collected on a per-project basis, therefore these measurements are often

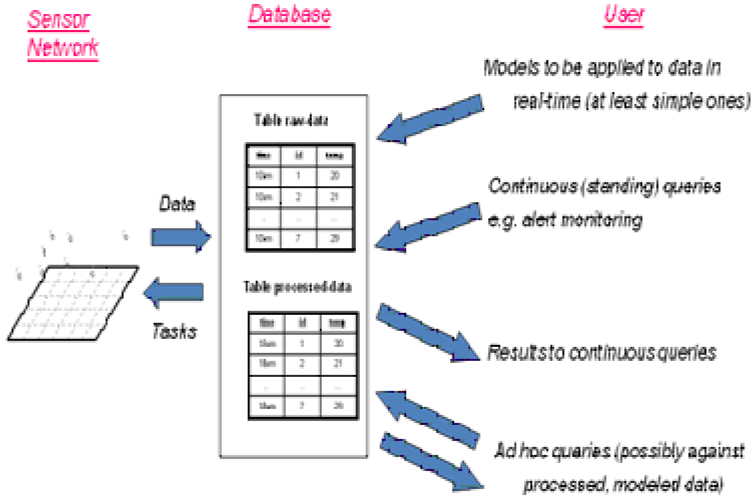
Sensor Network
Database
User
Models to be applied to data in real-time (at least simple ones)
Table raw data
Continuous (standing) queries e.g. alert monitoring
Data
Tasks
Table processed data
Results to continuous queries
Ad hoc queries (possibly against processed, modeled data)

Fig. 4: Sensor Data Processing for what we want

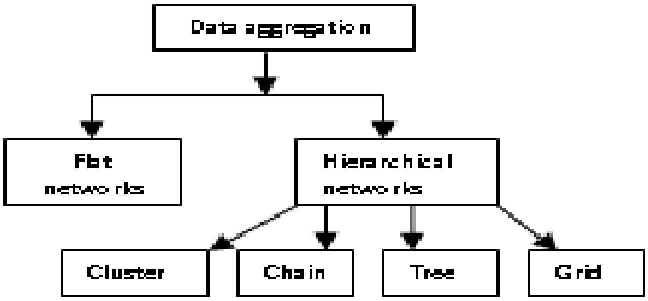

Fig. 5: Network architecture

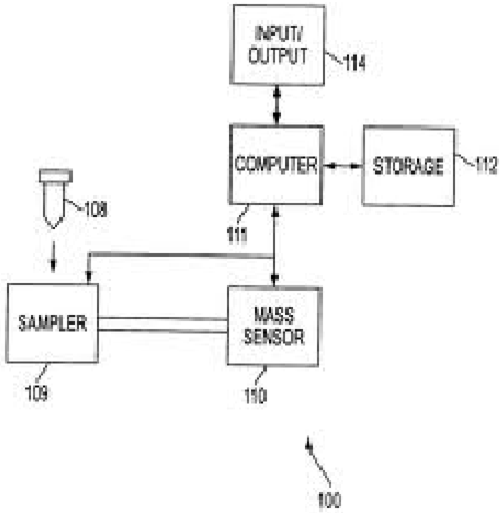

Fig. 6: Temporal profile analysis of mass data

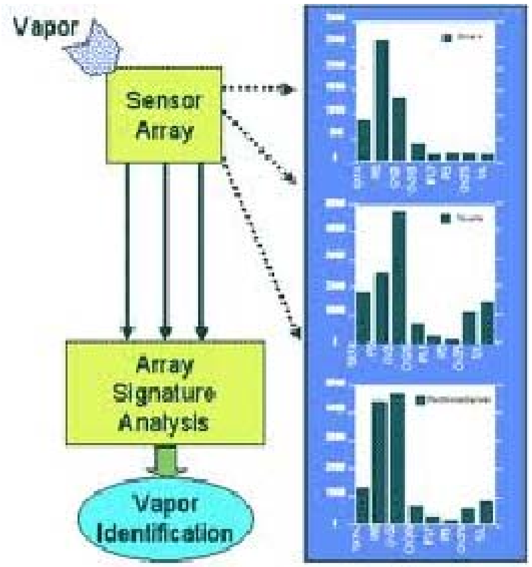

Fig. 7: Data Analysis for Multi-Sensor Arrays (171). Analysis Flow

duplicated between projects running at multiple organizations. A step in the right way to avoid this duplication is to introduce sensor networks, as they not only allow researchers to perform real-time data analysis, but enable sensor data sharing as well. However, in order to draw accurate conclusions or validate new models using this automatically collected data, metadata needs to be stored that gives meaning to the recorded observations. The sensor data generated by a sensor network depends on several influences, like the configuration and location of the sensors or the aggregations performed on the raw measurements.

Data mining of sensor data

Data mining as a process has already been received as an effective way to reduce time in the time consuming and memory consuming processes of obtaining knowledge from a given set of attribute values[13]. Data mining is treated as an algorithmic process that has a sensor data as an input, and as a result generates patterns for future prediction of hydrologic phenomena. The fundamental concepts that we use for the sensor data mining model are: sensor class, time interval for sampling, and threshold value. The sensor class is used to determine the sensor type by its location and sensing type. The time interval can have discrete sampling values or continuous interval values. The threshold value is given for the narrowing of the interesting values from the total set of values. In order to develop an effective sensor data mining model we subsequently used the three basic data analysis:outlier extraction, pattern generation, and prediction analysis.

CONCLUSIONS

There are a lot of aspects that can to be taken into account when creating a workflow for real-time sensor data processing. This ranges from the requirements for online retrieval of sensor data to the requirements that need to be met in order for the end user to analyze the retrieved data. By describing the whole workflow from the beginning to the end, a clear list of requirements has originated that can be used as reference material for the development and assessment of real-time processing systems. Comparisons with activities in data intensive science areas such as high energy physics and astronomy show that the data volume for all these activities is certainly challenging (hundreds of Petabytes) but, as has been seen, this is not an unmanageable data volume. Significant filtering of the data is a key component of any data collection activity. Sometimes this has to be done at the data source and in other cases can be done retrospectively. The key issue is not the availability or development of hardware; there seems to be ample capability in this regard both in the development of data sources (sensors) and data storage media. What does seem to be lacking is an adequate investment in software, so that the analyst can keep pace with the impressive developments to date in wide area surveillance

REFERENCES

1. P. Bonnet, J. Gehrke, and P. Seshadri. Towards sensor database systems. Lecture Notes in Computer Science,

2. L. Gurgen, C. Labb´e, V. Olive, and C. Roncancio. SStreaM: A model for representing sensor data and sensor

queries. In International Conference on Intelligent Systems And Computing: Theory And Applications (ISYC), Cyprus (2006).

3. C. Intanagonwiwat, R. Govindan, D. Estrin,J. Heidemann, and F. Silva. Directed diffusion forwireless sensor networking. IEEE/ACM Transactions on Networking, **11**(1): 2–16 (2003).
4. N. Koudas and D. Srivastava. Data stream query processing. In ICDE, page 1145, (2005).
5. L. Golab and M. T. Ozsu. Issues in data streammanagement. SIGMOD Rec., **32**(2): 5-14 (2003).
6. S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tinydb: an acquisitional query processing system for sensor networks. ACM Trans. Database Syst., **30**(1): 122-173 (2005).
7. X. C. Song and J. W. S. Liu. Maintaining temporal consistency: Pessimistic vs. optimistic concurrency control. IEEE *Transactions on Knowledge and Data Engineering,* **7**(5): 786-796 (1995).
8. J. R. Haritsa, M. J. Carey, and M. Livny. On being optimistic about real-time constraints. In PODS '90: 331–343, NY, USA (1990).
9. J. Huang, J. A. Stankovic, K. Ramamritham, and D. Towsley. Experimental evaluation of real-time optimistic concurrency control schemes. In VLDB'91.
10. K. Henricksen, A framework for context-aware pervasive computing applications, Ph.D. thesis, The School of Information Technology and Electrical Engineering, University of Queensland (2003).
11. Adrian K. Clear , Thomas Holland , Simon Dobson, Aaron Quigley,Ross Shannon, Paddy Nixon Situvis:A sensor data analysis and abstraction tool for pervasive computing systems.
12. L. Guibas. "Sensing, Tracking and Reasoning with Relations". IEEE Signal Processing Magazine. **19**(2): (2002).
13. Hluchy, L.; Habala, O.; Ciglan, M.; Tran, V.D.: "Mining and Integration of Environmental Data", IEEE International Conference on Computational Cybernetics, ICCC 2008, 27-29 Nov. pp. 247 – 252 (2008).
14. S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: a Tiny AGgregation service for ad- hoc sensor networks," In 5th Annual Symposium on Operating Systems Design and Implementation (OSDI), December Pages: 131-146 (2002).