Data Management for Sensor Networks

Johannes Gehrke*
Department of Computer Science
Cornell University

1 Introduction

One of the characteristics of the post-PC era is to push computation from desktops and data centers out into the physical world. This is an exciting time for systems research, as systems emerge with characteristics quite different from traditional environments. The area that we find especially interesting is networked sensors. Already today networked sensored can be constructed using commercial components using only a fraction of a watt in power on the scale of a few inches. Current prototypes measure already only $10cm^3$ [HSW+00], and application of Moore's law tells us that we will soon see components that measure 1 cm^3, and there is a plethora of research to scale down components to the 1 mm^3 range (about the size of a large piece of dust) integrated into the physical environment potentially powered by ambient energy [KKP99]. Instead of deploying preprogrammed sensor networks only for specific applications, future networks will have sensor nodes with different physical sensors for a wide variety of application scenarios and different user groups.

Sensor networks have the following physical resource constraints:

- **Communication.** The bandwidth of wireless links connecting sensor nodes is usually limited, on the order of a few hundred Kbps, and the the wireless networks connecting sensors provide only limited quality of service such as variable latency and dropped packets.
- **Power consumption.** Wireless sensors have limited supply of energy, and thus energy conservation is a major system design consideration. Current small batteries provide about 100mAh of capacity, powering a small Amtel processor for 3.5 hours (if no power management techniques would be applied). The current generation of sensor platforms uses about 2 microJ per bit of data transmitted. Note that future sensor nodes will have sophisticated power management features; current nodes already have three different sleep modes with several orders of magnitude different power usages [HSW+00].
- **Computation.** Sensor nodes have limited computing power and memory sizes that restrict the types of data processing algorithms that can be deployed and intermediate results that can be stored on the sensor nodes.

I advocate a database approach to sensor networks. Declarative queries are especially suited for sensor network interaction: Users and applications programs issue queries without knowing where and in which format the data is generated in the sensor network and how the data is processed to compute the query answer. Sophisticated catalog management, query optimization, and query processing techniques will isolate the user from the physical details of contacting the relevant sensors, potential in-network processing, and sending the results back to the user. Due to the large volumes of data produced in sensor networks and the above described physical constraints of sensor

*http://www.cs.cornell.edu/johannes; johannes@cs.cornell.edu

networks, data transmission back to a central node for offline storage, querying, and data analysis is infeasible for sensor networks of non-trivial size [EGHK99, PK00]. Thus information extraction from sensor networks has to perform non-trivial in-network data processing, data reduction, and aggregation, shortly, there is large potential for intelligent *data management*.

2 Challanges for the Database Community

Given the view of the sensor network as a huge distributed database system where each sensor node corresponds to a database site that holds part of the data, we would like or adapt existing techniques from distributed and heterogeneous database systems for the sensor network environment. But at close investigation, we can distinguish four major differences between sensor networks and traditional distributed and heterogeneous database systems.

Physical Characteristics. Sensor networks have physical characteristics that are very different from regular desktop computers or dedicated equipment in data centers. Sensors might fail at any time, the networking layer might only provide very weak quality of service, and the sensor nodes have strict resource limitations such as limited memory, computational and battery power. Query processing has to be aware of these physical constraints. One way of thinking about such constraints is the analogous interaction with the operating systems in traditional database systems [Sto81]. Database systems bypass the operating system buffer to have direct control over the disk. For a sensor network database system, the analogous resource is the networking layer, and for intelligent resource management we have to ensure that the query processing layer is tightly integrated with the networking layer.

We can distinguish several types of queries in a sensor network. *Long-running queries* deal with the status of the sensor network over a user-defined time period. Other queries are ad-hoc or *snapshot queries* that query the current status of the sensor network. Strategies for evaluating these two types of queries are likely to be very different: In long-running queries, we can pay up-front a higher cost that can be amortized over the lifetime of the query.

Due to inherent resource limitations of sensor networks, users should be able to trade off the accuracy of a query answer versus the quantity of resources used to compute the query answer. As a simple example, assume that the sensor network consists of N temperature sensor nodes. To accurately compute the average temperature, all sensor nodes need to be contacted and their temperatures aggregated. But the user might be sufficiently confident in the average of $M << N$ sensor readings, given that the sensors chosen are a random sample of the overall set of sensors. Computing the average of M sensor readings requires much less energy since only a small subset of all sensors is contacted. This is analogous to the computation of an aggregate through a sample in a database [OR86]. Further research is necessary to understand these tradeoffs in detail as it is not obvious how to select a truly random sample of sensor nodes that satify a given geographic constraint without complete knowledge about the sensor network at a central query optimization node.

Data Streams. Sensors produce data continuously in data streams, and sensor nodes have only limited memory and computational resources. We need to develop new query processing techniques for the online processing of data streams that do not assume that relations are materialized on secondary storage. Important for data stream processing will be intelligent data reduction at individual sensor nodes through the computation of stream aggregates. In addition to the computation of such statistics, we need to be able to process these synopsis data structure themselves when we combine synopsis data from several sensors. Conceptually, we can think about the data structures as new base relations; research has to address how to design operators that work directly on these and similar summary data structures.

Uncertainty Through Physical Measurements. Inherent to data that result from a physical measurement is *uncertainty* regarding the true value of the measured quantity. This uncertainty is often most properly described by a *continuous probability distribution function (*p.d.f.*)* over the possible measurement values. For example, consider a temperature sensor in your office that reports an estimate $\widehat{T}$ of the current temperature T; let this estimate be $\widehat{T} = 68°$ Fahrenheit (F). Given this measurement, do we believe that the temperature in your office is exactly 68° F? Assuming that the error introduced by the sensor has a gaussian distribution with a known standard deviation of σ°F, we can compute the probability that the true temperature T lies in the range $[T_1, T_2]$.

Probabilistic data models from the literature handle only discrete probability distribution functions, but the uncertainty associated with physical measurements is best described by a continuous probability distribution function such as a Gaussian but we need a data model for representing continuous probability distribution functions (p.d.f.'s) such as gaussians. Surprisingly, none of the numerous probabilistic data models described in the literature handles continuous p.d.f.'s—all models deal with discrete p.d.f.'s

Distributed Triggers. Many sensor networks will include *actuators* — devices that allow manipulation of properties of the physical world; simple examples are temperature controls, door locks, or light switches. Scalable, distributed trigger management is a considerable research challenges for large-scale monitoring and control sensor systems.

3 Conclusions

This paper outlines a research program that addresses fundamental problems in sensor networks: Data streams, uncertainty about sensor measurements, query processing, and trigger management. While developing techniques that address the three problems above, we must not forget that scalability of the techniques with the size of the network, the data volume, and the query workload is an intrinsic consideration to any design decision. I believe that sensor networks is a research area with challenging data management problems for years to come.

References

[ACM99] ACM SIGMOBILE. *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom-99)*. ACM Press, 1999.

[EGHK99] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. [ACM99], pages 263–270.

[HSW+00] J. Hill, R. Szewczyk, A. Woo, D. Culler, S. Hollar, and K. Pister. System architecture directions for networked sensors. *ACM SIGPLAN Notices*, 35(11):93–104, November 2000.

[KKP99] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for "smart dust". [ACM99], pages 271–278.

[OR86] F. Olken and D. Rotem. Simple random sampling from relational databases. *VLDB 1986*, pages 160–169. Morgan Kaufmann, 1986.

[PK00] G. J. Pottie and W. J. Kaiser. Embedding the Internet: wireless integrated network sensors. *CACM*, 43(5):51–51, May 2000.

[Sto81] M.Stonebraker. Operating system support for database management. *CACM*, 24(7):412–418, 1981.