Energy-optimized Sensor Data Processing

Elena Chervakova[1], Daniel Klan[2], Tobias Rossbach[1]

[1] Institute for Microelectronic and Mechatronic Systems gGmbH
D-98693 Ilmenau, Germany
[2] Department of Computer Science & Automation
Ilmenau University of Technology, Germany

Abstract. We consider the problem of efficient data aggregation and processing within wireless sensor networks (WSN). The system we propose allows monitoring and sensor data processing by injecting bytecode into the network. Additionally we present AnduIN, a system whose goal it is to decide which parts of a given query should be processed within the sensor network.

1 Introduction

Measurements such as temperature, light, humidity, CO_2, etc. play a major role in monitoring and controlling ambient conditions within buildings. An evaluation of the measured data makes it possible to detect correlations and attributes, which in turn enable a recognition of contexts. Fig. 1 shows some characteristic fluctuations of the sensor data associated with particular user activities: opening a window, turning on the lights, and a meeting. This shows that utilizing computing abilities of sensor nodes and evaluating known correlations allows for the creation of so-called *virtual sensors* which may serve as *context sensors*. This concept is also ideally suited to shift some portions of data processing algorithms into the network, reducing the amount of data transmitted and thus power consumption.

ConSAS [1] is a system for WSNs based on TinyOS 2.x [2] which enables energy-efficient data aggregation and in-network processing via injection of node-specific algorithms "on the fly" (during operation).

In recent years, several middlewares addressing the challenges of processing streaming data and dealing with the limitations of sensor hardware have been developed. Data stream management systems (DSMS) like Aurora [3] or TelegraphCQ [4] use sensor networks as simple data sources but do not address the problem of in-network processing. Conversely, in-network query processors (INQPs) such as TinyDB [5] and Cougar [6] focus only on the in-network part. Due to the limitations of sensor networks, INQPs usually support only primitive operators, such as aggregations. To overcome this problem, Franklin et al. developed the HiFi [7] system, which combines TinyDB and TelegraphCQ. Nevertheless, in HiFi, the user decides which operators are processed in-network and which are processed on the central instance. Since TinyDB does not support complex operations, only primitives can be processed within the WSN.

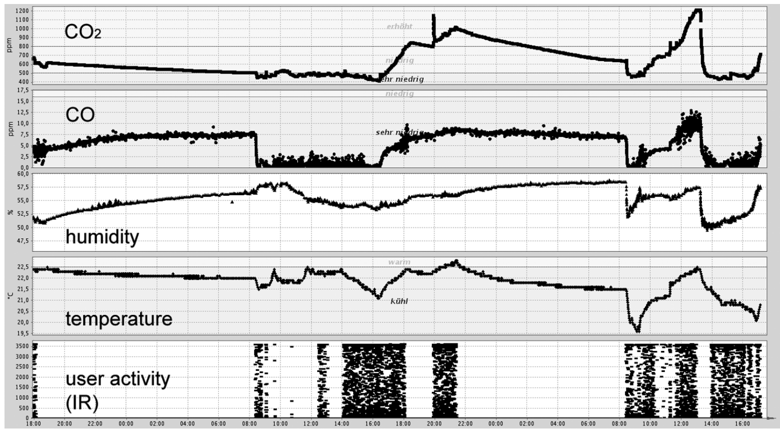

Fig. 1. Sensor data collected in an office room over a 24-hour period

To overcome this problems, we developed AnduIN (a combination of a DSMS and an INQP) [3]. In AnduIN, the query optimizer decides where operations will be processed.

2 ConSAS

ConSAS (Configurable Sensor and Actuator System) is a modular software system for sensor networks. It supports different types of sensor nodes and sensors and allows for easy integration of additional devices. Thanks to the system's modularity and configuration options, heterogeneous networks of different node types and sensors are supported, and so are a variety of routing protocols.

In our approach to in-network processing, we consider a system where physical sensors are inputs and virtual sensors are outputs. Output values are computed through mathematical operations involving one or more inputs. A virtual sensor y_i can be described mathematically as $y_i = F_i\left(x_n, \ldots, x_m\right)$, where $x_n \ldots x_m$ are physical sensors and F_i are mathematical operations on sensor data.

ConSAS supports representation of these operators as bytecode and its injection into the network "on-the-fly". Systems like TinyDB, which enable in-network processing and modification of the application, use over-the-air reprogramming, which implies the need to send entire new images for sensor nodes over the radio channel. However, sending/receiving over the radio channel and reprogramming of nodes both impose a very high power consumption. In our approach, we avoid this problem by changing the flow of execution using commands, representing the latter as bytecode. In order to inject new algorithms, a command is sent encapsulated in a data packet, thus allowing for significant energy

[3] http://www.tu-ilmenau.de/fakia/AnduIN.8598.0.html

savings compared to the transmission of entire firmware images. By equating in-network processing with the provisioning of virtual sensors, the set of commands can be limited to those required for the creation of the latter.

Another major question when aiming for a reduction of energy consumption through a reduction of data traffic is which parts of a given query should be processed within the sensor network — this issue is addressed by AnduIN.

3 AnduIN

As described above, AnduIN consists of a DSMS part and an INQP. Fig. 2 shows the overall architecture of AnduIN. The DSMS provides a query processor

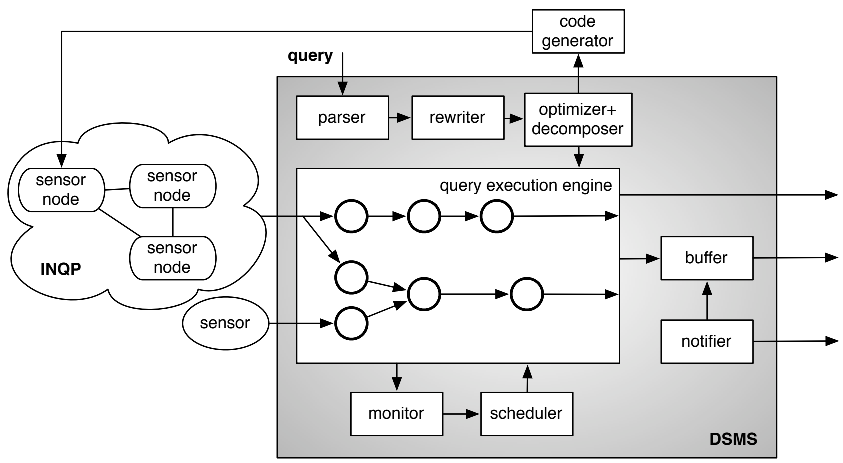

Fig. 2. Architecture of AnduIN

supporting the Continuous Query Language (CQL) [8]. In addition to standard CQL features, AnduIN provides support for synopses operators for, e.g., complex data stream analytics.

After a query has been successfully parsed it is translated into a logical query tree, which is subsequently optimized by a rule-based optimizer. One logical operator may have a set of physical representations.

In AnduIN, we additionally have the choice between a locally-executed operator and in-network-processed operators. From this, we can translate one logical operator tree into a (large) set of physical plans.

In order to choose the best plan, AnduIN enumerates all possible physical plans which can be derived from the logical plan. Subsequently, the overall sensor network power consumption can be estimated for each plan. For more details on our cost model, see [9]. With this, AnduIN chooses the plan that promises the maximum network lifetime. Both the in-network and central portions of the resulting plan are realized. After decomposition, sensor nodes can be reconfigured over-the-air.

4 Conclusion

The combination of TinyOS, the IMMS-developed ConSAS application software system (in combination with a suitable data acquisition system, such as the Gateway Application Framework [10]) and the approach of AnduIN allows for optimizing the energy consumption of WSNs used for sensor data collection. We are currently working on the integration and tests of various context sensors as well as the implementation of a replaceable localization module — a plug-in for ConSAS which allows to detect user activities and replace localization algorithms depending on application requirements.

5 Acknowledgment

Portions of this work are funded by the German Federal Ministry of Education and Research (BMBF) (support code 03WKBD3C) [11].

References

1. E. Chervakova, T. Rossbach, Wireless Sensor Networks for Building Automation, In: Proceedings of the EuroSSC 2008.
2. TinyOS, `http://www.tinyos.net`
3. D. Carney et al., Monitoring streams: a new class of data management applications, VLDB '02 Hong Kong, China.
4. S. Chandrasekaran et al., TelegraphCQ: Continuous Dataflow Processing for an Uncertain World, CIDR '03.
5. S .R. Madden et al., TinyDB: an acquisitional query processing system for sensor networks ACM '05.
6. Y. Yao, J. Gehrke, The cougar approach to in-network query processing in sensor networks, ACM '02.
7. O. Cooper et al., HiFi: A unified architecture for high fan-in systems, In VLDB - Demo '04.
8. A. Arasu et al., The CQL Continuous Query Language: Semantic Foundations and Query Execution, University of Stanford (2003).
9. C. Franke et al., In-Network Detection of Anomaly Regions in Sensor Networks with Obstacles, BTW 2009.
10. T. Rossbach, M. Götze, E. Chervakova, Architekturaspekte beim praktischen Einsatz drahtloser Sensornetzwerke In: Proceedings of the IWK Mittweida 2008.
11. CBS Customer Bautronic System, `http://www.customerbautronic.de/`