

From Cloud to Fog and Sunny Sensors

Position Paper

Hannes Grunert¹, Björn Butzin², Martin Kasparick², Andreas Heuer¹, and Dirk Timmermann²

¹ University of Rostock, Database Research Group, 18051 Rostock, Germany

² University of Rostock, Institute of Applied Microelectronics and Computer Engineering, 18051 Rostock, Germany

Abstract. Assistive systems collect large amounts of data in the internet of things and compute behavior and intentions of users in the cloud. Our approach is to push these computations (interpreted as database queries) as close as possible to the local sensors of the internet of things. We aim at replacing privacy-compromising cloud-based computations by fog- or edge-based computations or even by processing on the local sensors directly. Not only can this approach solve privacy problems, but also results in a better performance and energy-efficiency of the whole system: sensor-based computations are the (privacy-respecting) sunny side of the cloud. This position paper will give a short motivation and state of the art in different areas (from databases to wireless sensor networks) and will present our approach in combining modern interfaces to different sensors and concepts of database theory such as query rewriting and query containment.

Keywords: Cloud, Fog and Edge Computing, Internet of Things, Sensor Networks, Privacy, Performance, Query Rewriting, Query Containment

1 Motivation

Assistive systems support the users at work (Ambient Assisted Working) while they can remotely controls their homes (Ambient Assisted Living, AAL). Through various sensors, information about the current situations and the actions of the users are collected. This data is stored by the system and linked with other data from the web, for example the Facebook profiles of the users. By designing models for intention and activity recognition from the connected data, the smart environment can react autonomously to meet the needs of the users.

In assistive systems [12], significantly more information than required is collected in the cloud – which raises questions about privacy. The users usually have no or only a very small influence on the storage and processing of their personal data. If the cloud service is not located in their native country, the users cannot be sure that the same laws apply as in their home countries. As a result, their right to informational self-determination is violated.

The introduction of data privacy mechanisms in assistive systems is seen very skeptical by the developers. It is feared that the anonymization of the data hinders system development. Anonymization or pseudonymization of the data may lead to loss of detail, so that the results of analytic processes become inaccurate and, in extreme cases, unusable.

Our idea is to support assistive systems in performing the necessary behavior and intention recognition algorithms, but to automatically push the analysis operations as far as possible from the cloud to the (local) sensors. In an AAL environment, this results in a sensor-based or fog-based (edge-based) instead of a cloud-based computation. Besides privacy aspects, sensor-based or fog-based computing can also increase the performance and energy-efficiency of the system.

1.1 Cyber-physical systems and wireless sensor networks

The often cited law of Gordon Moore is used many times to argue that we do not need to increase efficiency, one just has to wait for the next hardware generation and thus a new boost of computational power. In wireless sensor networks (WSN), cyber-physical systems (CPS) and the internet of things (IoT), considerations are different. Here, the main driving factor is the reduction in energy consumption. Even newly developed moting platforms like UC Berkeley's Firestorm only have 512kB ROM and 64kB of RAM [1] which is not significantly more than in devices of 2005 in terms of absolute numbers, e.g., TelosB: 48kB ROM and 10kB RAM. Instead, the idle power consumption has been reduced, in this example by about 55 percent from $5.1 \mu A$ to $2.3 \mu A$. Hence, the focus on developing them is different. To save as much energy as possible, moting devices are kept asleep as long as possible. If awake, the transmission of data is the most costly operation in WSN, thus, it is tried to avoid them. Data should only be sent when requested by others or local memory is going to exceed. Additionally, through aggregation, preprocessing and compression, the time to send the next data-set can be extended. At this point computational power as well as memory capacities have to be utilized in an efficient way. After reducing the amount and frequency of transmissions required, the next level is to keep the middleware and transmission protocol energy-efficient. This means to reduce protocol overhead but also stack sizes and dynamic memory consumption need to be taken into account.

The idea of privacy through locality and energy saving constraints of wireless sensor networks harmonize with each other. However, other requirements of databases might be contradictory to the requirements of WSN e.g. in terms of latency, reliability, and consistency.

1.2 Privacy and Performance

By reducing and pre-aggregating raw sensor data to its minimal essence of required information, it is possible to protect privacy in smart systems. Additionally, the overall performance of the system can be enhanced when less data is analyzed on frequently busy nodes.

As part of our research project, privacy concepts for processing queries in assistive systems are designed. However, these concepts are not placed on top of the existing analysis functions, but are integrated in close cooperation during the development process.

The data-avoidant passing of the information regarding sensors and context to the analytical tools of the assistive system will not only improve the privacy-friendliness of the system. By pre-compressing the data by means of selection, aggregation, and compression operators on the sensor itself it is possible to increase the efficiency of the system. The privacy rights and the information requirements of the analysis tools can be implemented as integrity constraints in the database system. Through the integrity constraints, the necessary algorithms for anonymization and preprocessing can be run directly on the database. Thus, a transfer of the local data to external programs or modules, that might be located on different computing units, is omitted.

Instead of using hundreds of thousands of computers in the cloud (e.g. Google or Amazon), we can also use hundreds of billions of sensors or devices in the IoT to perform the necessary computations for the behavior and intention recognition of assistive systems. This results in fog or edge computing [11, 4] and even in local data processing on sensors.

2 State of the Art

We now give a short overview of the research areas cloud, Big Data, IoT (especially middleware and embedded database systems) and fog computing.

Cloud and Big Data: In the era of Big Data, more and more information is stored and processed in Cloud environments like IBM’s Bluemix and similar platforms. Such systems offer a variety of services and possibilities for data storage, including services for the Internet of Things (e.g. APIs for REST and MQTT).

Unfortunately, privacy is often ignored or, at least, it is not guaranteed by cloud services. For example, nearly every service offered by IBM Bluemix states in the Terms of Use that the service “[...] does not comply with the US-EU [...] Safe Harbor Frameworks” (e.g. the Watson service “Driver Behavior” [7]).

Internet of Things — Middleware: IoT, CPS and WSN are distributed networks of small and heterogeneous applications. The service oriented architecture (SOA) approach has shown to be useful in such environments. SOA is well known for its capability to integrate different applications horizontally and vertically. Due to the restrictions of the devices in such scenarios, SOA approaches have been tailored to fit the needs for reduced overhead, descending memory footprint and less required computational power. Examples of those are the constrained application protocol (CoAP) and the devices profile for web services (DPWS). CoAP is a promising candidate for IoT applications, as it is a RESTful SOA

type with less overhead than HTTP and adds publish subscribe mechanisms. A comparison of different CoAP implementations can be found in [2]. Another competing protocol for networked embedded devices is Message Queue Telemetry Transport (MQTT). It is a pure publish subscribe protocol, that uses a centralized broker to manage the message flow. Often the cloud is used as broker. Thus, due to its centralized nature, MQTT is less optimal than CoAP for our proposed solution.

Data publishing of “traditional” database systems is performed on established interfaces like JDBC or ODBC. By these standardized interfaces, the programmer does not have to care about the actual implementation and can write code independent of the actual database system.

Embedded Databases: Besides standard database systems there exist several specialized databases like Berkeley DB [10] and TinyDB [9]. These systems are designed to run on resource limited devices like Raspberry Pis or even as embedded databases directly on the sensor. In [5], several approaches to a distributed database management on sensor networks are compared, TinyDB among them, here especially aiming at energy efficiency. Acquisitional query processing [5, 9] can push queries to sensors and select relevant sensors in a WSN to reduce the amount of sensors needed for a computation (sensor reduction). In the existing approaches mentioned in [5], this sensor reduction is completely done manually (by the programmer).

Fog Computing: What is missing in fog computing, is a database-centered approach to computation, that is, given a query representing a necessary computation on the sensor data of the IoT, how to automatically prevent to simply transfer the complete data sets to the cloud servers.

In [6] we introduced a framework for privacy aware query processing in layered networks of “traditional” databases. The query processor includes modules for query rewriting and transformation, detecting key-like combinations of attributes and different anonymization concepts like k-anonymity and slicing, which can be extended by including new concepts for querying modern hardware, e.g. by rewriting a SQL query to different data management layers, the lowest of these being only able to perform some simple filter operations (like selections against given attribute values).

3 Vision

We propose a layered architecture with four logically distinguishable layers (see Fig. 1). The *Sensor Layer* includes the sensors which are very resource constrained in terms of CPU, memory, and power. The *Personal Layer* consists of typically mobile devices or embedded systems with higher performance but also high power constraints, like smartphones or edge nodes of a WSN. Router, home automation control units, private servers, etc. build up the *Fog Layer*. As these

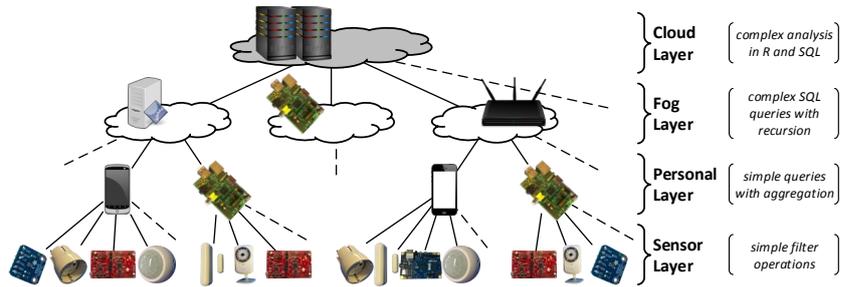


Fig. 1. Layered System Approach

devices have a wired power supply power saving is not as relevant as for the two lower layers. The *Cloud Layer* is built by powerful server farms without notable constraints according to power, CPU, or memory. Note that it is possible to have multiple layers within one of the four major layers, e.g. there could be several *Fog Layers* within a multi-tenancy or office building. From the top to the bottom layer resource constraints are increasing and the amount of possible (database related) functionalities and operations decreases. This layer approach has several advantages: In terms of privacy, each layer defines a strict transition where it can be defined which data is passed upwards and its granularity. This allows the fine-grained protection of critical personal data like health data, as the information can be stored and processed within the local parts of the system. Generally, the lower the layer, the higher is the ability of the user to control its own data.

As lower layers are more resource constrained than the upper ones, the middle layers provide functionalities for data processing as well as data transmission and additionally proxy functionalities. This enables optimized query execution according to the given resource constraints. The proxy functionality allows a reduction of the amount of communication. Especially within the *Sensor Layer* and between *Sensor* and *Personal Layer* this is a major aspect of power efficiency. The layered approach enables the power efficiency optimization of the overall system and not only for local nodes.

The constrained set of database operations at the lower layers can be compensated efficiently by the vertical fragmentation of queries into pushed-down and remainder queries. Thus, the privacy constraints of the users of these smart environments such as assistive systems are supported.

We apply our concept on machine learning algorithms to show how they will be transformed and pushed down in several steps resulting in simple filter operations.

A remaining open problem is to decide whether such queries can be performed on a resource-constrained device. If not, we have to check if the data can be sent to an upper layer without violating the privacy constraints of the users. This open problem results in a query containment problem that will be part of our future research.

Currently, only simple algorithms can be split up into their basic functions. The transformation of complex queries into simple fragments should be done automatically. By rewriting the complex query Q into Q_j and Q_δ , where Q_δ is executed outside of the protected system environment, we hope to only transfer data to the cloud that do not compromise privacy. We use extensions of the theory of query containment and query optimization for conjunctive queries [3, 8] to consider more complex queries (including complex statistical functions using aggregation and grouping) [6]. This approach can be extended to an IoT scenario with multiple layers, where the top layer is a cloud system while the bottom layer consists of embedded hardware.

The handling of data in IoT environments will be rethought fundamentally. Currently data is just pushed to the cloud while the layered approach enables new methods to store, process, and query data on the lower layers. To achieve this, IoT and database middleware have to be collated.

References

1. Andersen, M.P., Fierro, G., Culler, D.E.: System design for a synergistic, low power mote/BLE embedded platform. In: 2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). pp. 1–12
2. Butzin, B., Konieczek, B., Fiehe, C., Golasowski, F.: Applying the BaaS reference architecture on different classes of devices. In: 2nd International Workshop on Modelling, Analysis, and Control of Complex CPS (CPS Data). pp. 1–6 (Apr 2016), to be published
3. Chirkova, R.: Query containment. In: Encyclopedia of Database Systems, pp. 2249–2253. Springer US (2009)
4. Dastjerdi, A.V., Gupta, H., Calheiros, R.N., Ghosh, S.K., Buyya, R.: Fog computing: Principles, architectures, and applications. CoRR abs/1601.02752 (2016)
5. Diallo, O., Rodrigues, J.J.P.C., Sene, M., Mauri, J.L.: Distributed database management techniques for wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. 26(2), 604–620 (2015)
6. Grunert, H., Heuer, A.: Datenschutz im PArADISE. Datenbank-Spektrum 16(2), 107–107 (July 2016)
7. IBM: IBM Watson IoT Driver Behavior Service. <http://www-03.ibm.com/software/sla/sladb.nsf/sla/bm-7328-01?Open>, last access: 09.06.2016
8. Kolaitis, P.G., Vardi, M.Y.: Conjunctive-Query Containment and Constraint Satisfaction. 17. Symposium on Principles of Database Systems, Seattle pp. 205–213 (1998)
9. Madden, S.R., Franklin, M.J., Hellerstein, J.M., Hong, W.: TinyDB: an acquisitional query processing system for sensor networks. ACM Transactions on Database Systems (TODS) 30(1), 122–173 (2005)
10. Oracle: Oracle Berkeley DB 12c. <http://www.oracle.com/technetwork/database/database-technologies/berkeleydb/overview/index.html>, last access: 09.06.2016
11. Shi, W., Dustdar, S.: The Promise of Edge Computing. Computer 49(5), 78–81 (May 2016)
12. Weiser, M.: The Computer for the 21st Century. Scientific American 265, 94–104 (1991)