Distributed Shared Memory as an Approach for Integrating WSNs and Cloud Computing

Peter Langendoerfer, Krzysztof Piotrowski
IHP,
Im Technologiepark 25
15236 Frankfurt (Oder)
Germany
{langendoerfer,piotrowski}@ihp-microelectronics.com

Manuel Díaz, Bartolomé Rubio
Dpt. Lenguajes y C. Computación
Universidad de Málaga
SPAIN
{mdr,tolo}@lcc.uma.es

***Abstract*—In this paper we discuss the idea of combining wireless sensor networks and cloud computing starting with a state of the art analysis of existing approaches in this field. As result of the analysis we propose to reflect a real wireless sensor network by virtual sensors in the cloud. The main idea is to replicate data stored on the real sensor nodes also in the virtual sensors, without explicit triggering such updates from the application. We provide a short overview of the resulting architecture before explaining mechanisms to realize it. The means to ensure a certain level of consistency between the real WSN and the virtual sensors in the cloud is distributed shared memory. In order to realize DSM in WSNs we have developed a middleware named tinyDSM which is shortly introduced here and which provides means for replicating sensor data and ensuring the consistency of the replicates. Even though tinyDSM is a pretty good vehicle to realize our idea there are some open issues that need to be addressed when realizing such an architecture. We discuss these challenges in a tinyDSM independent way to ensure clear separation between concept and realization.**

Keywords-cloud computing, wireless sensor networks, distributed shared memory, security

I. INTRODUCTION

In the recent years the two fields; wireless sensor networks (WSN) and cloud computing have gained significant research interest and also became accepted very quickly, at least in the sense that there is a reasonable interest in using both these approaches also in commercial set-ups. Additionally, even though both approaches stem from different research areas they bear a certain potential for being combined to the benefit of the end users. But while the idea of having cloud computing just as a transparent layer on top of real world WSNs is appealing, its realization is challenging. The devices in the cloud are normally equipped with powerful processor, large memories and constantly power. In WSNs the devices look completely different, they come with constraint processing power, small memories and run out of a battery but are expected to work properly for months if not years. Thus, combining both worlds requires fully new concepts that help to do proper load balancing from the very beginning and by design.

Our concept is based on the well-known idea of distributed shared memory. The idea is as simple as it can be. By replicating the sensor readings in the WSN the whole network gets more reliable and broken links, battery outages etc. no longer prevent users from accessing the data, since it is still available from the replicas. In order to combine WSN and cloud computing we just need to extend the replication range so that the sensor readings are also stored in the cloud. This allows reducing access to the real sensor network to rare exceptions, since up-to-data information is already stored in the cloud so that all cloud users can work on that data. I.e. direct access to the WSN is needed only when data available in the cloud is out dated. The contributions of this paper are:

- Solid survey like state of the art concerning integration of WSNs and Cloud Computing.
- Discussion of the architectural concept to combine WSN and cloud computing via distributed shared memory.
- Introduction of basic means needed to implement our ideas.

The rest of this paper is structured as follows. We first provide a survey on the WSN and cloud computing integration efforts published so far. Then, in Section 3 we discuss our idea of representing a WSN as a set of virtual sensors as part of a certain cloud, based on the concept of distributed shared memory. In the following section we introduce tinyDSM, a middleware approach that was developed by us and provides a shared memory abstraction to WSNs. The middleware description is extended by the discussion on the open issues when using tinyDSM for WSN cloud computing integration.

II. STATE OF THE ART

To the best of our knowledge, first attempts to combine WSN and cloud computing have been publish at the beginning of 2009. In the following different proposals are analysed. We have classified them into four categories taking into account the main objectives of the approaches.

A. WSN Cloud Interaction

In this section we consider those proposals that are mainly focused on the way WSNs interact with the Cloud.

In [1] a Cloud Computing model mainly based on pipes and filters is proposed for the system hosted at the cloud infrastructure. Pipes do not transform data (coming from sensors), but generally buffer it and provide a uniform interconnection mechanism of filters. Filters do specific processing and transformation on the input data, such as data refinement or suppression. A simple filter could, for example, be responsible for storing the data in the database or deleting data below or above a certain threshold.

The work presented in [2] is focused on finding the shortest path between a sensor node (from the WSN) and a cloud server node (from the Cloud). An efficient ant colony optimization technique is used, so that all data from the sensor node can be uploaded to nearby cloud server, and if necessary efficient query execution is done when required by sensor nodes. Failures of sensor nodes or server nodes are also taken into account. Alternative paths can be selected for faster response time.

B. WSN in the cloud

A different vision of WSNs and Cloud integration is given in some approaches, where WSNs are considered as entities inside the Cloud, instead of a different system that interacts with the Cloud.

In [3] authors propose what they call Tangible Cloud Computing, which extends the current domain of the Cloud to include the physical world (WSNs). This means that networks of physical devices (sensor nodes) should be able to expose their functionality as standardized Cloud services. As first class entities in the Cloud, devices in the Tangible Cloud can also be used together with 3rd party cloud resources. In this approach, WSNs are inside the Cloud, instead of interacting with it from outside, like in other proposals.

Taken the previous work as background, in [4] authors demonstrate Cloud Computing capacity for supporting elastic sensing and modelling applications and show that it is feasible for sensor nodes to use and manage the Cloud-based extensible modelling resources. The elastic nature of Amazon EC2 has been shown as a perfect candidate to support the dynamic loads provided by integrated environmental monitoring and modelling applications.

C. WSNs-Cloud users interaction

The proposals of this category are mainly focused on the way the data collected from the sensors can be obtained by Cloud users.

In [5] a content-based publish/subscribe framework is proposed. Sensor data are coming through gateways to a publish/subscribe broker, which delivers sensor information to the consumers of SaaS applications. In order to match published sensor data or events with subscriptions efficiently, a fast and scalable event matching algorithm called Static Group Index Matching is proposed. On the other hand, the approach also allows dynamic collaboration between other cloud providers.

The Service Oriented Architecture (SOA) is considered in [6]. The proposed architecture consists of a layered service stack that has management, information, presentation and communication layers with all required services and repositories. The architecture uses SOA and features of cloud like virtualization to deal with heterogeneity. Services are used to allow the interaction between WSNs, subscribers and other clouds.

A different approach is presented in [7]. A data-channel software tool kit called Connector is defined integrating a program-side I/O and graphics package (Connector-API), a user-side GUI (Connector-GUI), a web-based GUI (Web Server) and a sensor-side data publisher (Sensor Server). The main design principles of this proposal are: facilitating Java FileInput/OutputStream-based uniform channels by hiding all underlying network protocol; separating connection set-up and data-sampling work from user code into an independent configuration file; and automating channel recovery and redirection upon a job/user migration.

The Sensor-Cloud infrastructure introduced in [8] virtualizes a physical sensor network as a virtual sensor network on the Cloud. Users need not worry about the real locations and the differences of multiple physical sensors. They can use and control virtual sensors with standard functions. The infrastructure also provides a user interface for registering or deleting of physical sensors, for requesting for provisioning or destroying virtual sensors, for controlling and monitoring virtual sensors, and for registering and deleting users.

D. Application perspective

Some proposals are currently appearing with the main objective of applying the integration of WSNs and Cloud Computing to different application scenarios. Some examples are the following:

In the application area of people healthcare have appeared several approaches, such as [4] and [9]. Besides of integrating WSNs and Cloud Computing, these proposals also take into account the security aspect in order to provide the data or information on the Cloud with a good protection. The systems proposed monitor human health, activities, and share information among doctors, care-givers, clinics, and pharmacies in the Cloud, so that users can have better care with low cost.

In [10] authors focused on introducing the latest technologies in sensors, wireless networks and Cloud computing to radically revise approaches to agriculture and conduct business feasibility studies to establish a hypothetical model of Cloud services that make a genuine contribution to agriculture. They conducted demonstration tests with the cooperation of two Japanese farming corporations.

An architecture for smart building control and energy management is designed and implemented in [11]. It is argued that a significant proportion (in US about 40%) of total worldwide energy is consumed by buildings. This way, making buildings more energy-efficient is an important step to reduce energy consumption in the combat with global climate change. A high-level system architecture is proposed supporting three main design principles: hierarchical sense and respond; a

reference semantic model that facilitates information exchange among the various building subsystems; and a subscription/usage based model over the Cloud in order to minimize the cost of IT over the lifecycle of a building.

Besides these academic proposals, several companies are launching products to take advantage of WSNs and Cloud Computing integration. Two examples are MicroStrain [12] and Digi International [13]. The former offers SensorCloud, a unique sensor data storage, visualization and remote management platform that leverages powerful cloud computing technologies to provide excellent data scalability, rapid visualization, and user programmable analysis. SensorCloud supports any web-connected third party device, sensor, or sensor network through a simple OpenData API. It is useful for a variety of applications, particularly where data from large sensor networks needs to be collected, viewed, and monitored remotely. Digi International provides us with iDigi Device Cloud, which is based on six pillars: device connectivity (iDigi utilizes an open device integration protocol), application integration through a variety of APIs, scalability, reliability, performance and security.

III. SENSOR CLOUD ARCHITECTURE

The core of our idea is to virtualize the wireless sensor network (WSN) and by that to create kind of a virtual sensor which is then becoming part of the cloud. As a result, the interaction between the cloud and the real world WSN is decoupled – at least to certain extent. This decoupling is mainly realized by the fact that all operations executed in the cloud are run with the data present in the virtual sensor. By that, the resource constraints of the real world sensor nodes can be neglected up to some extent. Additionally, it allows for splitting the security concept into two parts, i.e., the cloud vs. virtual sensor, and virtual sensor vs. real sensor nodes. The former can be handled by standard security means as they are already in place. For the latter we propose to research lightweight means in order to ensure that security features are not contradicting other system properties such as long life time of the real world sensor network.

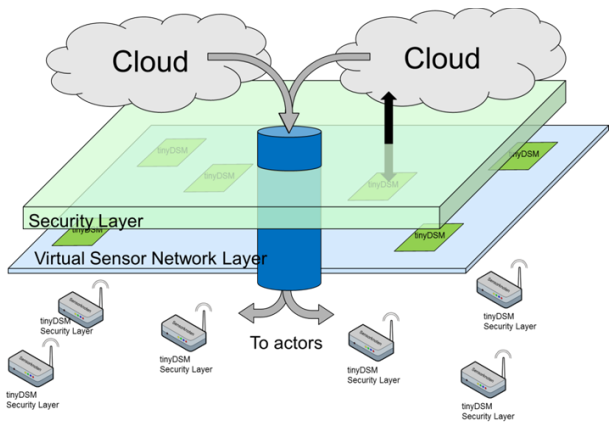

Figure 1: The sensor cloud architecture with the virtualization shortcut represented by the blue tube

In order to ensure a consistent and accurate view on the real world system in the virtual sensor layer we propose to use the tinyDSM approach which was designed to ensure consistent management of data replication in WSNs. Here the replication needs to be extended to include the virtual sensor network layer in the replication range.

In the system in question several security-related issues exist. First, in such an integrated solution there are two different domains with different abilities regarding computational power and energy. These domains also involve different kinds of foreseen user interactions and access methods. These possible user interactions induce also possible treats against the system and combining WSNs with cloud computing actually puts the relatively weaker WSN in danger. Thus, it is absolutely necessary to provide security means to protect the WSN part from unauthorized operations issued from the cloud. Additionally, to allow seamless and flexible integration of WSN in the cloud, adaptable security means are necessary that, on one hand, are strong enough, but on the other hand, do not put the energetic efficiency of the WSN in danger.

In order to ensure adaptation needed for real time behaviour, especially for emergency situations when the application running in the cloud needs to interact with the physical nodes to react on some specific sensor reading, we propose to provide a virtualization/security shortcut. And the idea here is NOT to compromise security for efficiency, but to allow direct communication from the application in the cloud to the real world sensors while relaxing the sensor virtualization. We propose to enable the direct cloud to real world sensor communication through the shortcut, if and only if predefined conditions are fulfilled and reflected in the virtual sensor network (emergency). In such a case, the security configuration is subject of transient change, i.e., as soon as the emergency situation is solved the original security and access regime is re-established.

The areas of work will be mainly the following ones:

- Framework/architecture for integration of intensive data sensing applications in the cloud. This includes the sensor virtualization techniques and means to access the data on the sensor node from the cloud.
- Cost-effective computation models. This requires distributed and cooperative operations and effective resource management strategies.
- Timely and predictable communication. This includes both, the communication inside the private cloud and from the user to the cloud. SLA definition and mapping to execution guarantees.
- Adaptable security mechanisms that allow integration of resource constraint WSN devices with powerful cloud devices.

IV. TINYDSM

tinyDSM[14][15] is a middleware approach that allows to define and enforce data replication in WSN obeying a certain -

user defined - level of consistency. By that, the sensor reading can be retrieved from the wireless sensor node that did the reading itself, but also from those nodes that keep a replica of the data. By extending the replication range such that also replicas are held and updated by a computer running a cloud service, integrating the WSN into the cloud can be done transparently for the cloud user. The representation of the WSN nodes in the cloud is in this paper referred to as a virtual sensor node. In the following paragraphs we provide information on tinyDSM, before discussing open issues with realizing the close integration of WSN with cloud computing.

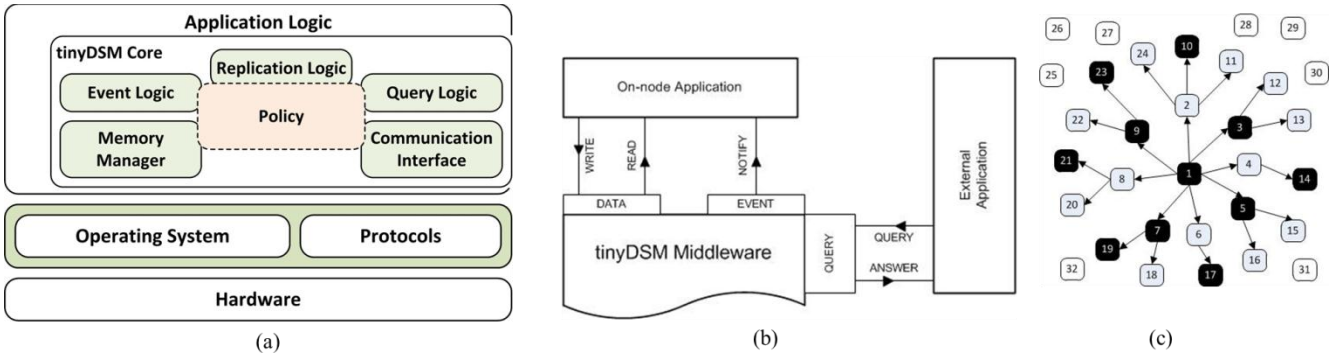

Figure 2: The tinyDSM middleware allows controlling the data replication to reduce the access to the sensor that is the actual source of the information; (a) tinyDSM components, (b) tinyDSM interfaces; (c) tinyDSM replication area

The concept behind the proposed programing framework is that the shared memory area is defined at the compile time. Its definition includes the definitions of the shared typed variables and the events that shall be detected, because the states of the individual events are also available in the shared memory. The access to the defined shared memory area and the notifications in case of event detections are provided by the middleware layer that is configured with these definitions. The middleware provides defined interfaces to the application and hides the message passing. The definition of each variable and event includes policy settings that specify the handling of the data. The policy parameters control the applied consistency enforcing mechanisms, but are also used to configure other parameters of the middleware. They specify the data identification details, specify the replication and consistency details, and enable reliability options and optimizations. The policy parameters specify also simple access rights, like read-only data or data writable by the source only.

The complete set of variables' and events' definitions represents the configuration of the middleware that is further used by the pre-compiler to generate the application-specific core of the middleware. The pre-compiler defines appropriate data structures and functions tailor-made for the application. And, in order to support heterogeneity in the underlying sensor network, the generated C code of the middleware is further encapsulated in an operating system adaptation layer, specific for the platform, it shall be used with. To allow easy adaptation to any operating system and hardware platform, the middleware uses a minimum set of system services and does not rely on any specific features or guarantees from the network stack. This built-in portability feature ensures easy adaptation of tinyDSM to powerful devices running the cloud stuff.

The encapsulated middleware can be compiled together with the application logic to obtain an image that can be installed on the nodes or a program that can be executed on the cloud devices.

The separation of the middleware and the application logic allows flexible application development. Single application logic can be used with different policy configurations to generate multiple versions with different data sharing features. It is also possible to generate multiple versions that support diverse levels of data transitioning between the virtual sensor and the cloud. This means that the contradictory requirements regarding the consistency and the delay of the data accesses, can be tuned to meet the resource constraints of a given application scenario.

A. Quality enforcing mechanisms

Several mechanisms to enforce the policy settings have been proposed within the tinyDSM framework. These mechanisms allow also to reduce the costs and to improve the scalability. And since the cost reduction usually comes together with fidelity reduction, these configurable mechanisms allow tuning the operation scope as well as the operation frequency in order to reduce the costs.

The spatial reduction is realized by the centralized management of the master copy and by limiting the forwarding of requests to a specified replication region. Dependent request forwarding further reduces the forwarding of the requests to the areas in the forwarding region where the data is concentrated. These mechanisms take advantage of the usual access locality and allow estimating the maximum costs of the operations.

Tuning the frequency of operations allows reducing the number of expensive actions to the defined minimum. This is realized by the instance filtering that updates the replicas only with the most important values and by the replication strategy, which reduces the frequency of the update verifications. All these mechanisms allow fine grained definition of the fidelity relaxation using the corresponding policy parameters.

B. Additional means required

In order to ensure proper reflection of the WSN in the cloud, it is required that updates of sensor readings are

propagated also into the virtual sensor node in the cloud. The currently available enforcement means do not allow for specifying individual nodes as "must be updated". This is fully correct since in the system set-up considered so far all nodes have been providing the same services, so no special roles needed to be obeyed. There are two principle strategies that might be applied, coming with different benefits and drawbacks:

- Proactive forwarding of sensor readings: This strategy would require that all updates are send to the virtual sensor, in the same way as it is done in the WSN. The problems are caused by the differences in the topology. The virtual sensors are located somewhere outside the WSN meaning that all updates need to go through gateway/base station devices. Such an approach comes with known problems such as depleting the battery of nodes close to the gateway by far faster than those of other nodes, making these nodes bottleneck, if the number of updates is significant. This situation becomes even worse, if it is required that the virtual sensor nodes are updated reliably, i.e., if they need to send acknowledgements. The load situation as well as the timing can be relaxed, if all sensor nodes holding a replica of the data to be updated are regarded as if they were representing just one single node. In that case, it is sufficient that one of these nodes retrieves the acknowledgement, and distributes it later, e.g., by piggy-backing it with other data.
- Reactive updating of virtual sensor: This strategy does not require new means for specifying tinyDSM. It needs means to define retrieval queries which can be used to periodically ask for updates from the WSN. The benefit of this approach is that it does not require any exchange of acknowledgements since the virtual sensor knows whether or not it received the update and can initiate a second request in case of a failure. The amount of needed data exchange can be kept minimal, following the strategy to consider all replica holders equivalent well suited to answer request from the virtual sensors. The major drawback is that state changes of the real world sensor are not visible in the virtual sensor for a certain time interval. In average, the delay is half of the update period, but in the worst case it might be the total period. The other issue is the situation when the updates are required periodically and the WSN is sometimes queried without obtaining new data, because the sensor reading did not change.

Exploiting the flexibility offered by the tinyDSM approach and extending the features it already supports, it is possible to define application specific configurations that provides the ideal data consistency of the replicas in the integrated WSN and cloud computing system. Such a data consistency may combine the above mentioned proactive and reactive update propagation to achieve the best results depending on the focus of the application, e.g., high energetic efficiency, low access delay.

V. CONCLUSIONS

Combining WSN and cloud computing is a very interesting research topic that has high potential of being taken up by industry and end users. In this paper we have introduced the concept of reflecting WSNs inside the cloud by virtual sensors. The underlying technology, i.e. distributed shared memory for wireless sensor nodes was developed by us and named tinyDSM. Currently it is not used in the context of cloud computing. The benefits of the tinyDSM approach compared to other concepts discussed in the literature are:

- The level of consistency between data measured and stored in the real WSN and their replicas in the virtual sensors in the cloud can be defined by the user and will be enforced by tinyDSM means.
- tinyDSM allows for different types of data i.e. data that is shared e.g. measurement values and management data that needs stronger protection.
- The reflection of data in the virtual sensor allows to create a valid picture not only of the measurement data but also of the topology, i.e. the whole WSN can be "virtualized". By that even decisions based on complete network knowledge can be taken in the cloud which offers the possibility to run by far more complex algorithms for decision taking.

There are also open issues that need additional investigation. Load balancing and energy consumption triggered by updating the virtual sensors in the cloud is one of these topics.

We are planning to realize our idea with our own sensor nodes running tinyDSM in a testbed which we will use then to research open issues especially the security aspects and the different types of update mechanisms.

REFERENCES

[1] W. Kurschi, W. Beer. "Combining Cloud Computing and Wireless Sensor Networks". Proceedings of the 11th International Conference on Information Integration and web-based Applications and Services (iiWAS'2009), pp. 512-518. Kuala Lumpur, Malaysia, December 14-16, 2009.

[2] R. Sen. "Exchanging of Information between Cloud Computing Server and Sensor Node for Effective Application Development". *Proceedings of the International Symposium on Devices MEMS, Intelligent Systems and Communication (ISDMISC'2011)*, pp. 31-34. Gangtok, India, April 12-14, 2011.

[3] K. Lee, D. Hughes. "System Architecture Directions for Tangible Cloud Computing". *Proceedings of the 1st ACIS International Symposium on Cryptography and Network Security, Data Mining and Knowledge Discovery, E-Commerce and Its Applications and Embedded Systems (CDEE'2010)*, pp. 258-262. Qinhuangdao, China, October 23-24, 2010.

[4] X.H. Le, S. Lee, P.T.H. Truc, et al. "Secured WSN-integrated Cloud Computing for u-Life Care". *Proceedings of the 7th Annual IEEE Consumer Communications and Networking Conference (CCNC'2010)*, pp. 1-2., Las Vegas, NV, USA, January 9-12, 2010.

[5] M-M. Hassan, B. Song, E-N. Huh. "A Framework of Sensor-Cloud Integration; Opportunities and Challenges". *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication (ICUIMC'2009)*, pp. 618-626. Suwon, S. Korea, January 15-16, 2009.

[6]] S.V. Patel, K. Pandey. "Design of SOA Based Framework for Collaborative Cloud Computing in Wireless Sensor Networks". *International Journal of Grid and High Performance Computing, 2(3)*, pp. 60-73. July-September 2010.

[7] J. Melchor, M. Fukuda. "A Design of Flexible Data Channels for Sensor-Cloud Integration". *Proceedings of the International Conference on Systems Engineering (ICSEng'2011)*, pp. 251-256. Las Vegas, NV, USA, August 16-18, 2011.

[8] M. Yuriyama, T. Kushida. "Sensor-Cloud Infrastructure: Physical Sensor Management with Virtualised Sensors on Cloud Computing". *Proceedings of the 13th International Conference on Network-Based Information Systems (NBIS'2010)*, pp. 1-8. Takayama, Japan, September 12-16, 2011.

[9] J. Jayashree, J. Vijayashree. "Ubiquitous Life Care Integrates Wireless Sensor Network and Cloud Computing with Security". Global Journal of Computer Science and Technology, 11(15). September 2011.

[10] M. Hori, E. Kawashima, T. Yamazaki. "Application of Cloud Computing to Agriculture and Prospects in other Fields". Fujitsu scientific and technical journal, 46(4), pp. 446-454. October 2010.

[11] H. Chen, P. Chou, S. Duri, H. Lei, J. Reason. "The Design and Implementation of a Smart Building Control System". *Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE'2009)*, pp. 255-262. Macau, China, October 21-23, 2009.

[12] MicroStrain.http://www.microstrain.com/.

[13] Digi 2011. http://www.digi.com/.

[14] K. Piotrowski, P. Langendoerfer, and S. Peter, "tinyDSM: A highly reliable cooperative data storage for Wireless Sensor Networks," in Proc. of Collaborative Technologies and Systems, 2009. CTS '09. International Symposium on, 18-22 May 2009, Baltimore, MD, USA, pp. 225-232.

[15] K. Piotrowski, "Assessment of the Feasibility of Distributed Shared Memory and Data Consistency for Wireless Sensor Networks",Ph.D. Thesis, Brandenburg University of Technology, Cottbus, December 2011.