International Journal of Pervasive Computing and Communications

On the application of Big Data in future large-scale intelligent Smart City installations

Sylva Girtelschmid Matthias Steinbauer Vikash Kumar Anna Fensel Gabriele Kotsis

Received 20 March 2014
Revised 20 March 2014
Accepted 8 April 2014

On the application of Big Data in future large-scale intelligent Smart City installations

Sylva Girtelschmid and Matthias Steinbauer
Institute of Telecooperation, Johannes Kepler University Linz, Linz, Austria
Vikash Kumar and Anna Fensel
The Telecommunications Research Center Vienna, Vienna, Austria, and
Gabriele Kotsis
Institute of Telecooperation, Johannes Kepler University Linz, Linz, Austria

Abstract

Purpose – The purpose of this article is to propose and evaluate a novel system architecture for Smart City applications which uses ontology reasoning and a distributed stream processing framework on the cloud. In the domain of Smart City, often methodologies of semantic modeling and automated inference are applied. However, semantic models often face performance problems when applied in large scale.

Design/methodology/approach – The problem domain is addressed by using methods from Big Data processing in combination with semantic models. The architecture is designed in a way that for the Smart City model still traditional semantic models and rule engines can be used. However, sensor data occurring at such Smart Cities are pre-processed by a Big Data streaming platform to lower the workload to be processed by the rule engine.

Findings – By creating a real-world implementation of the proposed architecture and running simulations of Smart Cities of different sizes, on top of this implementation, the authors found that the combination of Big Data streaming platforms with semantic reasoning is a valid approach to the problem.

Research limitations/implications – In this article, real-world sensor data from only two buildings were extrapolated for the simulations. Obviously, real-world scenarios will have a more complex set of sensor input values, which needs to be addressed in future work.

Originality/value – The simulations show that merely using a streaming platform as a buffer for sensor input values already increases the sensor data throughput and that by applying intelligent filtering in the streaming platform, the actual number of rule executions can be limited to a minimum.

Keywords Big Data, Ontology, Demand response, Semantic reasoning, Smart city

Paper type Research paper

This work is supported by the Austrian government and the City of Vienna within the competence center program COMET. It is part of the joint strategic project "Analysis on semantically structured data from smart buildings and smart grids" which is carried out by the Telecommunications Research Center Vienna (FTW) and the Institute of Telecooperation at the Johannes Kepler University (JKU), Linz.

International Journal of Pervasive Computing and Communications
Vol. 10 No. 2, 2014
pp. 168-182

1742-7371
DOI 10.1108/IJPCC-03-2014-0022

1. Introduction

As a response to the urgent call to lower emissions of carbon dioxide, the Smart Grid concept, closely tied to the concept of Smart Building/Home and eventually Smart City, all aimed at improving energy efficiency, has emerged.

A wide array of commercial product solutions, supporting the adoption of Smart Grid technology among energy providers (Stimmel and Gohn, 2012) as well as Smart Building technology among building constructors and managers are already available on the market (Talon *et al.*, 2011). Although these products are a sound starting point to achieve energy savings, the vision of a Smart City reaches much farther, including broader resource integration, increased ease of deployment and operation, automation, dynamic load adaptation and policy sharing. To achieve this vision, one of the integral parts of Smart Gird is the development of demand response (DR) market solutions that overcome the scalability issues of such an extensive market (Magoutas *et al.*, 2011). The use of effective DR implementations can greatly reduce electricity generation, and, in effect, Smart City could realize the goal of reducing carbon emissions. However, the vast scale of data to be processed to forecast the right energy need of a whole city raises the complexity of the DR mechanism. Moreover, the commercially available Smart Building Analytics Technologies are known to be cumbersome to setup and reconfigure, making them less likely to get adopted (Yu *et al.*, 2013). The main priority in further development is to improve the user experience. In the light of these problems, a wide area of research has formed around the Smart Grid/Building/Home concept, for readability purposes, further generalized and referred to as Smart City. The application of semantics is one such initiative.

Making use of semantic modeling to describe resources in various domains has been explored in numerous projects focusing among other topics also on Smart Cities (Pena and Penya, 2011; Zhou *et al.*, 2012; Kumar *et al.*, 2012). The widespread adoption of semantic technologies in the Smart City domain stems from the fact that it enables flexibility in system configuration and adaptation. Additionally, it can provide intelligence via reasoning over the system. This concept allows for interoperability of diverse distributed system components, such as sensor devices, smart meters and smart plugs. It can also provide the basis for higher-level interoperability such as rule translation and policy sharing (Kumar *et al.*, 2012). As a result, an ontology-based semantic model has become prevalent in the area of context-aware and Smart City technologies (Crapo *et al.*, 2011; Tomic *et al.*, 2012; Wagner *et al.*, 2010). Nonetheless, semantic processing is not capable of managing such vast amounts of data, as are expected in the Smart City scenario, in real-time (Crapo *et al.*, 2011; de Mues *et al.*, 2011; Espinoza *et al.*, 2011; Weijun *et al.*, 2007). Furthermore, in a study by Stimmel and Gohn (2012), the call for application of Big Data technology in Smart Grid analytics is emphasized, and, as suggested by Talon *et al.* (2011), cloud delivery of energy management solutions is largely becoming an accepted approach.

We contribute to this field of research by proposing and evaluating a novel system architecture for Smart City applications which use ontology reasoning and a distributed stream processing framework on the cloud. With our approach, the decision-making process is fully automatic and self-contained, and, at the same time, the system remains robust and time-efficient even in a large-scale domain. By lowering the processing load on the rule engine, we use not only computing resources but also real resources in Smart City installations more efficiently. As shown in this paper, intelligent preprocessing of sensor data (smart home sensors, smart meters and additional data such as weather and traffic information) will also lead to a lower amount of rules that actually fire and thus trigger actuator changes such as adapting heater settings and changing building lighting.

The remainder of this paper is organized as follows: in Section 2, we summarize current research related to the topics of ontology-based Smart City and processing of Big Data. Section 3 describes our proposed architecture and gives requirements for a real-world implementation. The prototype we have built to prove the concept is shown in Section 4, and the lessons learned and concluding remarks can be found in Sections 5 and 6, respectively.

2. Related work

Our work is closely related to ontology-based reasoning in the Smart City domain and to the processing of Big Data. In this section, we introduce the reader to the Smart City concepts and cover the related research. A concise description of the novelty of our architecture approach will be given at the end of this section.

2.1 Smart city

In the conventional power distribution systems, the customer (e.g. household) is viewed as a passive consumer of energy. With the Smart City paradigm, however, comes a different scenario, in which power plants need to interconnect with customers (also acting as prosumers) and with other distributed renewable energy sources. The challenge in coordinating this communication and fostering energy-saving and reuse calls for Information and Communications Technology (ICT) technologies to be applied.

Much research, concerning interactions of Smart Houses and Smart Grids, has already been conducted and many related projects have emerged. The report by Gungor *et al.* (2012) presents a summary of projects from around the world. In many cases, the work is also carried out by the industry. These initiatives promise to shed more light on the challenges of integrating heterogeneous participants and the ability to handle the large amounts of data needed to be reasoned over. However, the progress is rather slow, largely due to privacy and security issues (Simmhan *et al.*, 2011) and the power grid's high reliability needs. Given the fact that at this point there is little available to capitalize on in terms of the actual home-grid data interchange, we would like to contribute to the advance in the research by proposing a solution that we claim is general enough to support the energy coordination in a Smart City. More specifically, we claim our solution to be:

- adaptable to processing any range of data from various domains (i.e. in addition to power usage from smart meters, pricing and critical peak period signals, important information can be inferred from current weather reports, schedule information shared by customers, schedule about special city events or anything else that helps forecast and control power usage); and
- applicable to the coordination of the new power grids (often micro grids) and their distributed nature.

This leads us to two major requirements for our design:

(1) base our design on semantics to allow for seamless communication between diverse systems; and

(2) satisfy a sufficient real-time response of such a complex system by applying Big Data technology.

The next two subsections address the research related to these two requirements.

2.2 Ontology

Within the realm of computer science, ontologies are formal representations of concepts within a specific domain. Such models describe the relationships between pairs of these concepts, providing a common vocabulary for the given domain, and consequently, eliciting knowledge sharing. Ontologies can also include reasoning rules allowing processing of knowledge and deriving new information via inference (Uschold and Gruninger, 2004). The use of semantics in the Smart City problem domain has been researched in numerous works. For example, in Rohjans (2013), the work focuses on semantic-based architecture communication by applying standards of ICT. The author identified standards and specifications necessary for seamless data exchange among Smart Grid devices and systems and proposed a solution which relies on annotating semantic metadata to services to allow server and clients to share information.

Pena and Penya (2011) apply semantic tools in their distributed architecture approach within the ENERGOS project and propose that a unique global ontology, based on the standards from common information model (CIM), should be sufficient to represent the Smart Grid domain as a whole.

In Zhou *et al.* (2012), the authors integrate a Smart Grid information model and apply it to DR optimization application using complex event processing (CEP). A number of ontologies (electrical equipment, organization, infrastructure, weather and special and temporal ontologies) are integrated to represent a complete Smart Grid and the relationships among them have also been defined.

In our previous work (Kumar *et al.*, 2012), on which we base our current research, a semantically enabled Smart Building system was implemented, combining home automation techniques and data from smart meters, smart plugs and sensors. Complex rules and policies were created to monitor and administer the centrally stored data that were updated in near real-time. The system was deployed in two real-life buildings (a school and a factory floor), and data were collected over a period of several months resulting in almost 10 million triples. The measured data were obtained via REST and Web Services and stored in an OWLIM-Lite-based (Ontotext, 2014) semantic repository using Java SESAME 2.0 API. Custom rules implemented as SPARQL Construct queries were also used for reasoning in a number of use cases. Clients (tablets and phones) could query the repository via the OWLIM store REST interface. Complex rules and policies were created to monitor and administer the centrally stored data that were updated in near real-time. Several end-user services were developed:

- real-time feedback on power consumption;
- appliance status;
- comparison to historical consumption;
- remote shutdown for appliances; and
- a notification system alerting if an appliance was turned on although not in use. These services have polled for the repository changes at 15-minute intervals. There were almost 10 million triples collected during the experiments.

One can expect that in the context of Smart City where the amount of customers can reach a million or more, the amount of collected information multiplies into data on the order of terabytes which would potentially require daily processing and reasoning over (Simmhan *et al.*, 2011).

In summary, the use of ontologies has proven to be inevitable for Smart Building/Grid applications. However, some authors consider the use of ontologies in context-aware environments to be very sensitive to the size of the dataset and unreasonable to use for time-critical applications (Weijun *et al.*, 2007). Similarly, in (Espinoza *et al.*, 2011), the authors have reported a limitation in semantic processing, where the reasoner and rule engine would not be able to perform in real-time when applied to the full-semantics model. We too are skeptical about the performance in scenarios of Smart City where the context data are of large size, constantly changing and often incomplete. We turn to Big Data to remedy these issues.

2.3 Streaming big data

According to a study by Stimmel and Gohn (2012), the amount of sensors in Smart Grids, combined with those in Smart Homes/Buildings, will vastly increase the data influx in the near future. Moreover, the coordination of energy in Smart Cities is highly time critical and dependent on reliable data (Wagner *et al.*, 2010). This leads us to a situation in which the stream processing tools from Big Data technology are needed to ensure efficient processing of the generated data (Lynch, 2008).

Real-time streaming platforms are tools in Big Data that are able to handle large volumes of data, arriving to the system at high velocities, by using compute clusters to balance the workload. Those systems inherit some properties of message passing interfaces clusters, but add scalability to the feature set. They are able to rebalance the workload if too many messages need to be processed in a certain compute node. There are three systems that can be considered mature enough for productive environments:

(1) Project Storm (Storm, 2014) (developed at Twitter).

(2) S4 (2014) (developed at Yahoo).

(3) Project Spark (Spark, 2014).

All these systems have in common the ability to reliably process events or messages on distributed compute clusters. For all messages entering the system, they guarantee processing even if some of the compute nodes in the cluster fail. Finally, all systems can be dynamically reconfigured, making it possible to adjust the size of the cluster during runtime.

Our work stands apart from the previous related work in the combination of ontology-based reasoning with Big Data streaming methodology. We approach the integration of these two technologies by off-loading the basic data processing tasks (data cleansing, broken sensor detection, normalizing, threshold alerts, etc.) to the compute cluster and work with a reduced dataset (in terms of volume and throughput) in the ontology repository. As a result, we read from and, most importantly, write to the ontology only when necessary.

3. Proposed architecture

Based on what we found in literature (Stimmel and Gohn, 2012; Talon *et al.*, 2011; de Mues *et al.*, 2011; Weijun *et al.*, 2007) and our initial proof-of-concept performance testing on ontology repositories, we have concluded that moving Smart City applications (which heavily make use of ontologies and rule-based reasoning) to the cloud and integrating it with a real-time computation platform is desirable. On this account, we suggest architecture, featuring components for real-time processing and reasoning.

In the following, we discuss the proposed architecture as shown in Figure 1. The main components: streaming platform, ontology repository, rule engine and possible client applications are displayed. The figure also shows the general flow of information in the system.

Sensor data originating from any sensor found in a Smart City (e.g. smart meters and smart sensors) are sent to a real-time streaming platform. This platform is assembled into a cluster from many individual compute nodes. Due to the inherent feature of stream processing engines (working on streams of messages/events, they are able to balance these streams between compute nodes in a cluster), each of these nodes is handling streams of sensor data from arbitrarily many sensors, depending on the amount of data the individual sensors are generating. The streaming platform component is responsible for detecting any considerable changes in sensor data readings or failures in sensors and for accumulating sensor readings, where applicable. Further, data cleansing processes (handling of outliers, temporary sensor outages, normalization, calibration, etc.) are applied within the compute cluster to free underlying components from these tasks. Each node in the cluster can directly access the ontology repository to be able to make changes to it.

The ontology repository records the most recent readings and accumulated information such as the average temperature of the last hour or the average power consumption of the last week. Any changes in the ontology are triggering the rule engine to re-evaluate rules and take actions, if applicable.

Within the rule engine, a set of rules are stored and executed on a timely basis on user request and whenever the compute cluster is signaling a change event. The rule engine has several means for output of rule execution results:

- The engine can raise alerts. These are stored in the ontology repository for later querying and can be sent to clients that have registered to receive alerts of a certain category. The first option (querying) is especially useful when clients are polling the repository for alerts, which is the case in our earlier work. These legacy clients therefore remain compatible. Further, alerts are used to adapt the preprocessing that is performed in the compute cluster (update sensor readings more frequently, stop reading certain sensors, etc.).
- The outcome of rule execution can also just be an adaption of the preprocessing process (dependency between sensors has changed, sensor needs to be read less frequently, error correction needs to be adapted, etc.) to adapt to new situations.
- Last but not least, the rule execution results can just be sent back to clients, especially if rule execution was initiated by a client application.

Finally, client applications are able to register to receive results of rules stored in the rule engine (alerts, events, etc.), send queries to the rule engine for one time execution and store new rules in the rule engine for regular or event driven execution.

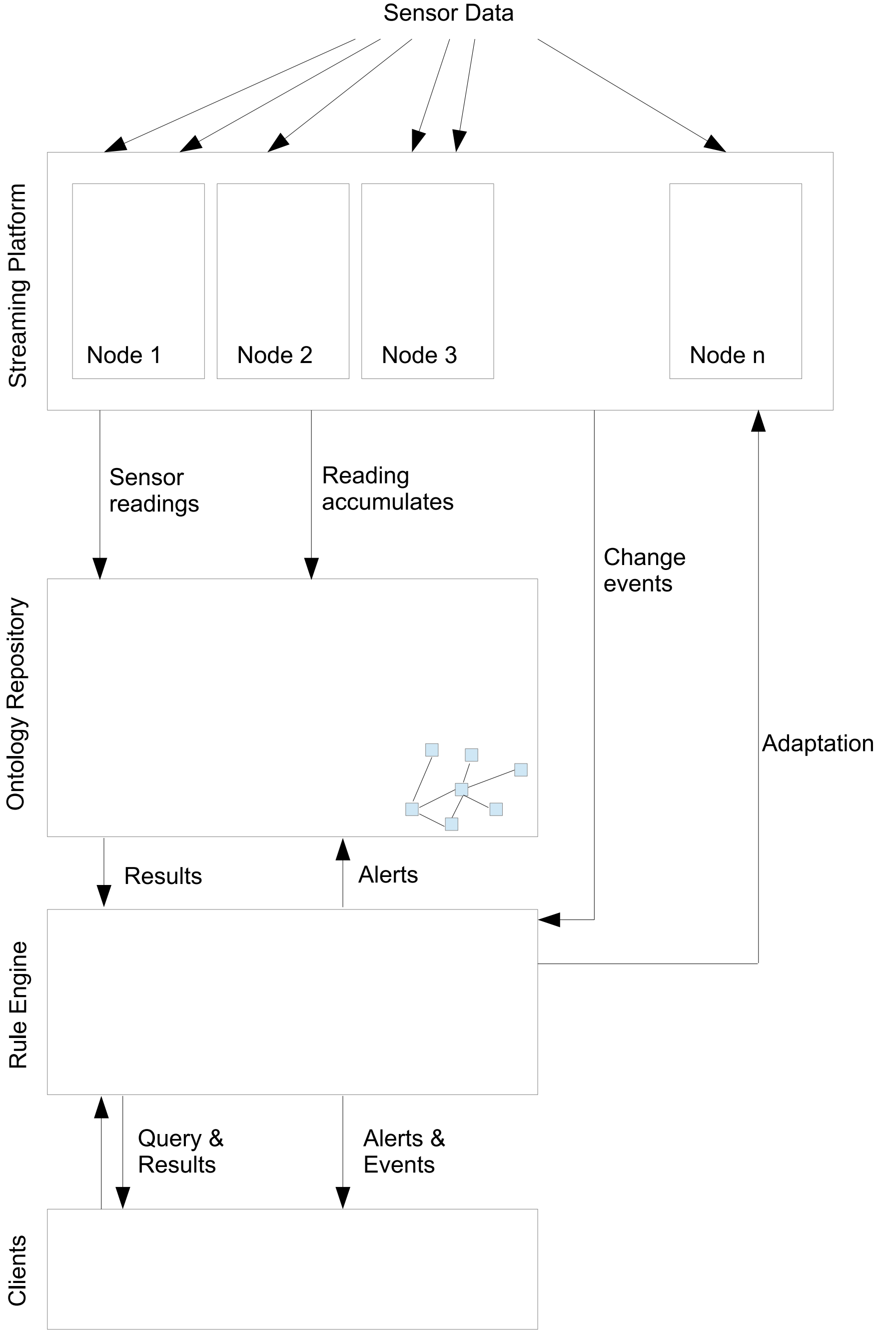

Figure 1.
High-level overview of the Smart City data processing architecture

4. Prototype

To show the technical feasibility of our approach and to get a first evaluation of this architecture, a prototype implementation was created. In accordance with the proposed architecture a cloud-based real-time stream processing platform and an ontology repository along with a rule engine were needed. In the following, we motivate our decisions for the software packages and frameworks we have used in our implementation.

For the distributed streaming platforms several options are available. Although S4 would provide the advantage of integrating with Spring configuration (Spring, 2014) and Spark would already provide tight integration with other Big Data tools such as Hadoop and Hive, we have chosen to use Storm as the basis for the prototype implementation. The deciding factor for our choice was the availability of a simple administrative interface for monitoring the cluster and the possibility to specify upper bounds for parallelism. This, in our case, can be used to vary the cluster size for running experiments.

As already explained to some extend in the previous section, Storm is a distributed and fault-tolerant real-time computational platform. It knows two basic processing primitives: Spouts and Bolts. Spouts are used to stream data to the system. They connect to the data sources, in our case to Redis-backed (Redis, 2014) sorted sets of sensor readings and pass this data encapsulated in so called Tuples to the processing units (Bolts). These Bolts are able to consume and emit streams of Tuples. They build the main building blocks of a Storm system. It is in these Bolts where our preprocessing takes place. Spouts and Bolts are tied together in a so-called Topology that describes the flow of messages within Storm. It allows specifying groupings, and the number of concurrent instances for a certain Bolt or Spout can be bounded in the Topology.

The ontology repository, describing a representational subgroup of facets typical for a Smart City model, is mostly reused from the previous work of Kumar *et al.* (2012), where ontologies of Smart Buildings and the connector software talking to the hardware (sensors, smart meters, smart plugs, etc.) were already put to test. However, to prove our concept in this article, we implement our stripped-down ontology as a set of simple Java classes as is required by our rule engine. The repository does not store extensive data (sensor readings, timings, etc.), as this was the case in (Kumar *et al.*, 2012). Instead, only the most recent readings and accumulated information are being recorded, effectively reducing the work complexity done by the reasoning engine.

As our reasoning engine in the prototype, we are using the Drools Fusion module of Drools from JBoss (Drools, 2014). Drools is an object-oriented business rule management system (BRMS) with a forward chaining inference-based rules engine, i.e. a production rule system. Drools uses an enhanced implementation of the Rete algorithm and is very flexible when it comes to adapting to any problem domain. Drools Fusion is the Drools module for enabling CEP capabilities, offering features such as temporal reasoning and reasoning over an absence of events (Fusion, 2014). We have chosen this rule engine, thanks to it being rated as one of the fastest and most flexible open-source rule engines available as of now, which allowed us to rapidly build our test scenarios.

All of our components are running inside of virtual machine containers on top of an OpenNebula (2014) cloud installation set up at one of our cloud computing labs. This cloud is featuring 24 CPU cores, 72 GB of RAM which allows us to deploy test installations and to conduct experiments with varying cluster sizes. All virtual machines are connected to a 1-GB switch in bridged networking mode which sets the upper bound of our internal bandwidth limit. A diagram illustrating the setup is shown in Figure 2.

Being fault-tolerant and scalable, Storm guarantees that each emitted Tuple is processed by the correct order of Bolts and the cluster adapts to the addition and removal of compute nodes dynamically. Thanks to these properties, it is possible to seamlessly grow the system further (from just handling data for a number of buildings up to the size of whole cities) without drastically affecting the performance of the system.

4.1 Evaluation

To carry out the performance evaluation, we simulated large-scale operations of a Smart City with three city size setups. The data used in our experiment have been collected from the work of Kumar *et al.* (2012). This very same data were extrapolated and slight offsets (timestamps were shifted and time frames were compacted at random) and variations (slightly higher or lower measures and sensor failures) were introduced. The resulting data were then streamed at high velocity to the experiment cluster to simulate the activities in a virtual Smart City.

The simulation to test our prototype is driven by a configurable Smart City model created anew for each test run. This simple model takes care of generating energy-consuming entities (for the prototype testing so far, it generates buildings with sensors and appliances in their rooms). This model can be readily extended to include additional sensor data or other entities that play a role in a Smart City, once their data are available to us.

The test setup is able to run in two general setups:

(1) The sensor data arriving at the system are either directly forwarded (going through no pre processing at all) to the task responsible for updating the ontology which will result in reevaluation of all the affected rules eventually.

(2) Or they are preprocessed in a number of steps that filter jitter, cope with failed sensors and skip values for obvious outliers.

In the later case, the ontology is updated only selectively, causing the rules not to be evaluated as often. Every ontology update triggers the rule engine, which is making simple decisions for every building.

The goal of our evaluation was to show that sensor data can be streamed at a very high velocity to the cluster and our system is capable of scaling in size such that it can adapt to larger compute demands of larger city sizes. To show the feasibility of our approach, several experiments were run. For this a cloud-based experiment, the launcher Util was created. This tool is able to facilitate arbitrary cloud stacks in experiment runs. Before each run for each of the clusters components, virtual machine

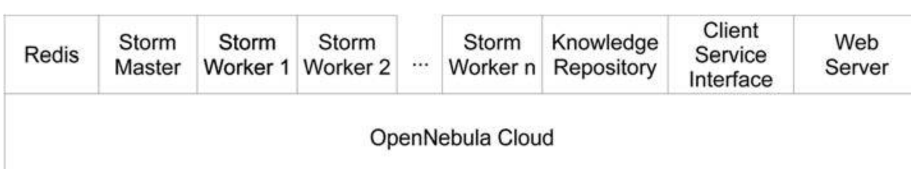

Figure 2.
Stack of the components as setup on top of OpenNebula

instances were automatically created and configured on the cloud. The front-end machine responsible for managing the smart city model and evaluating Drools rules was configured with 4 virtual CPUs and 8 GB of memory. The worker nodes of the storm cluster were configured with 4 virtual CPUs and 2 GB of memory each. After each experiment run, we evicted all used virtual machines from the cloud to make sure no caching effects or something of the like had any influence on our results.

We ran experiments for configurations of 100 (small), 200 (medium) and 300 (large) buildings, where the exact layout of each building (number of rooms, number of users, number of sensors, etc.) was always determined at random. For each of these city configurations, we streamed artificial sensor data to the storm cluster as described above and recorded how many Drools rules evaluations were necessary to handle the sensor events. For each configuration, there were three experiments executed with filtering-enabled and three runs with filtering disabled. Further, each configuration was run on different Storm cluster sizes of 2, 3, 4 and 5 compute nodes respectively. With these city configurations we observed no significant changes in the execution time for Drools rules. As long as the rules only considered a fraction of the data model, the execution time remained stable. Single rules executed in 24,000 *ns* on average on the test setup.

As for the different runs in filtered and unfiltered mode, we can clearly show that intelligent filtering of sensor data on a Big Data platform leads to less workload in the reasoning engine. In Figure 3 a plot of all experiment runs is shown. It is clearly visible that larger cluster sizes lead to two interesting results.

(1) Larger cluster sizes allow the system to accept more sensor data in parallel which increases the overall systems bandwidth. The top three lines in the plot show the number of necessary Drools rules evaluation for the small, medium and large city configuration. The system was able to handle spikes in sensor data throughput better so that on larger clusters more Drools runs could be executed.

(2) However, the overall goal is to execute only as much rule evaluations as possible, which refers to filtered mode in our system.

The lower three lines show the number of necessary Drools rules evaluations for the small, medium and large city configuration. The plot in Figure 3 also makes it obvious that the number of rule evaluations can be reduced to a minimum when our filtering mechanisms are in place. Only sensor data changes that matter to rule outcomes are passed to the rule engine; thus, rule evaluation becomes necessary only in the case of significant sensor data changes.

This means that the assumption that Big Data streaming platforms are beneficial in Smart City DR systems holds. The evaluation clearly shows that simple pre-filtering mechanisms can drastically reduce the amount of semantic rule evaluations and the system gain scalability. Additionally, one has to take into account the fact that in a real-life application, the rules are likely to be much more complex as well as more numerous, making the case for the use of preprocessing within a streaming platform even more sound. We also believe that our simulation proves the point even though the data used are not directly related to the data that would need to be processed in a real-life Smart City DR system. We claim this is relevant because in both cases the streaming data are essentially timestamp-value Tuples.

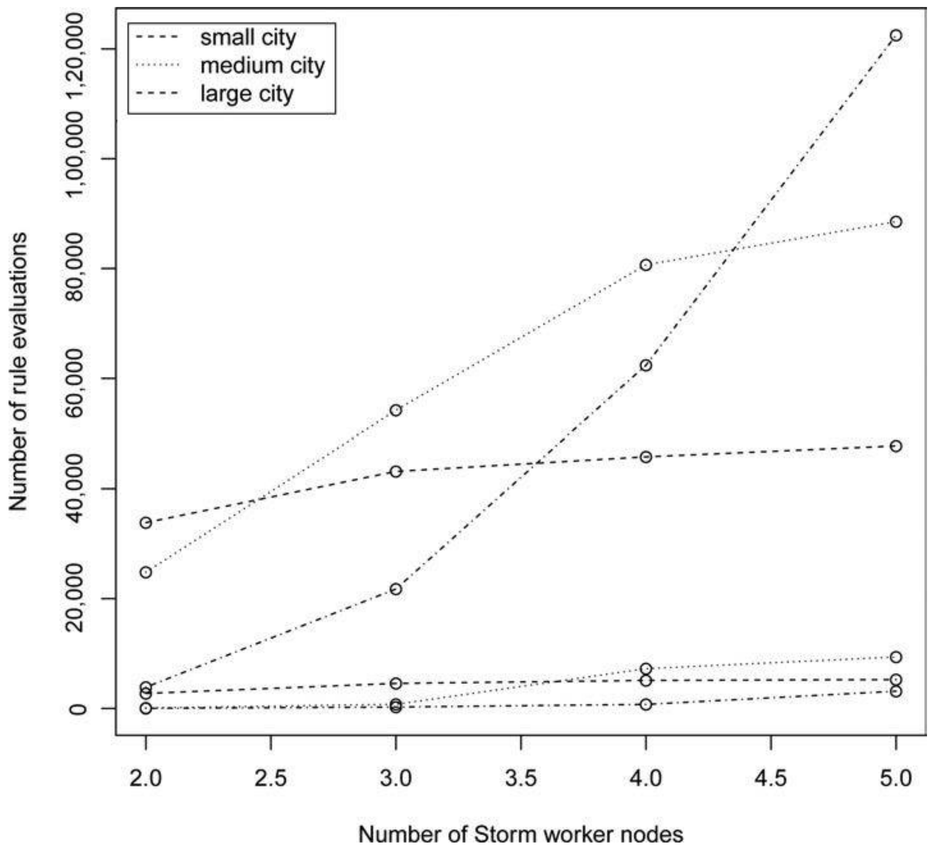

Figure 3.
Experiment runs

5. Lessons learned

The prototype shows that our approach is both feasible for the problem scope and scalable to larger problem sizes. Still, there are issues that need to be addressed to make this architecture applicable in real-world scenarios.

During our experiment runs, we have clearly seen that the model size that one can run Drools rules on is mainly restricted by operating memory and not by computation. This means while Drools was a good candidate to run our experiments where we could still keep all our data model in memory, this approach will fail for real-world cities where the city size might vastly differ from the cities used in our experiments. In these cases, different approaches with semantic models stored in semantic databases will need to be used. In this case, the effects described in Section 4.1 will have even more impact.

One such matter is the lack of any security issue considerations in both our architecture and implementation. High standards in security and privacy are by all means indispensable (Wagner *et al.*, 2010; Simmhan *et al.*, 2011) in Cloud-enabled Smart City deployments. To just point out a single example, users of such a system could quite easily run denial of service attacks by creating rules that consume a large amount of computing resources, and, moreover, real-world resources of a Smart City. As the aim of this experiment was strictly to prove the indispensability of including some sort of a streaming platform in the semantically enabled Smart City architectural decisions, we did not concern ourselves with security and policy issues. Future work on realizing our prototype will take these measures into account and investigate novel security implementations applicable to Smart City solutions hosted on Cloud.

Our work faces the problem that there are no complete datasets publicly available in the Smart City category that possess Big Data properties. For further improvements of our system, it will be necessary to work with more realistic datasets as well as appropriate rules to evaluate situations and derive decisions in the scope of demand and response in the Smart Grid. Additional work is planed with using weather information as a one way to increase the complexity of decision-making.

In continuation, we are also planning to devote substantial attention to improving the streaming platform component where more advanced filtering, failure detection and data cleansing needs take place.

6. Conclusion

In this paper, we have proposed and presented an architecture for efficient processing of sensor data from Smart City installations. We concluded that in such larger-scale scenarios, the influx of data, needed to be processed to optimize energy usage, requires smart reasoning mechanisms, such as ontologies, to live up to its full potential. Because the current ontology-based knowledge databases would cause performance bottlenecks in large-scale installations, we have addressed this matter by combining ontology-based reasoning with Big Data processing. Namely, our architecture uses Big Data streaming clusters for basic processing needs, while the time-consuming ontology-driven reasoning is applied only when necessary.

References

Crapo, A., Piasecki, R. and Wang, X. (2011), "The smart grid as a semantically enabled internet of things", in *Implementing Interoperability Advancing Smart Grid Standards, Architecture and Community: Information Interoperability Track, 2011*, *Phoenix, AZ*.

de Mues, M.O., Alvarez, A., Espinoza, A. and Garbajosa, J. (2011), "Towards a distributed intelligent ICT architecture for the smart grid", in *Industrial Informatics 2011 Ninth IEEE International Conference*, *Lisbon*.

Drools (2014), "Drools: business logic integratin platform", available at: www.jboss.org/drools/ (accessed 20 March 2014).

Espinoza, A., Ortega, M., Fernandez, C., Garabjosa, J. and Alvarez, A. (2011), "Software-intensive systems interoperability in smart grids: a semantic approach", in *Industrial Informatics 2011 Ninth IEEE International Conference*, *Lisbon*.

Fusion (2014), "Drools fusion: event processing", available at: www.jboss.org/drools/drools-fusion.html (accessed 20 March 2014).

Gungor, V.C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C. and Hancke, G.P. (2012), "Smart grid and smart homes: key players and pilot projects", *Industrial Electronics Magazine, IEEE*, Vol. 6 No. 4, pp. 18-34.

Kumar, V., Fensel, A., Lazendic, G. and Lehner, U. (2012), "Semantic policy-based data management for energy efficient smart buildings", in *On the Move to Meaningful Internet Systems: OTM 2012 Workshops, Rome*, Springer, Berlin, pp. 272-281.

Lynch, C. (2008), "Big data: how do your data grow?", *Nature*, Vol. 455 No. 7209.

Magoutas, B., Apostolou, D. and Mentzas, G. (2011), "Situation-aware demand response in the smart grid", in *Intelligent System Application to Power Systems, 2011 16th International Conference*, *Crete*.

Ontotext (2014), "OWLIM | ontotext", available at: www.ontotext.com/owlim (accessed 20 March 2014).

OpenNebula (2014), "Flexible enterprise cloud made simple", available at: http://opennebula.org (accessed 20 March 2014).

Pena, A. and Penya, Y.K. (2011), "Distributed semantic repositories in smart grids", in *Industrial Informatics 2011 Ninth IEEE International Conference*, *Lisbon*.

Redis (2014), "Redis: data structure server", available at: http://redis.io (accessed 20 March 2014).

Rohjans, S. (2013), "A standard-compliant ICT-architecture for semantic data service integration in smart grids", in *Innovative Smart Grid Technologies Procedings*, *Copenhagen*, IEEE.

S4 (2014), "Apache S4: distributed stream computing platform", available at: http://incubator.apache.org/s4/ (accessed 20 March 2014).

Simmhan, Y., Kumbhare, A.G., Baohua, C. and Prasanna, V. (2011), "An analysis of security and privacy issues in smart grid software architectures on clouds" in *Cloud Computing 2011 IEEE International Conference*, *Washington, DC*, IEEE.

Spark (2014), "Apache spark – lightning-fast cluster computing", available at: http://spark.apache.org (accessed 20 March 2014).

Spring (2014), "Spring framework", available at: http://spring.io (accessed 20 March 2014).

Stimmel, C.L. and Gohn, B. (2012), "Smart grid data analytics: smart meter, grid operations, asset management, and renewable energy integration data analytics: global market analysis and forecasts", *Executive Summary*, Pike Research, Boulder, CO.

Storm (2014), "Apache storm, distributed and fault-tolerant realtime computation", available at: http://storm.incubator.apache.org (accessed 20 March 2014).

Talon, C., Jaffe, S. and Nicholson, R. (2011), "IDC marketscape: worldwide smart building energy analytics 2011", IDC Energy Insights Distributed Energy Stragegies, Framingham, MA.

Tomic, S., Fensel, A., Schwanzer, M., Veljovic, M.K. and Stefanovic, M. (2012), "Semantics for energy efficiency in smart home environment", *Applied Semantic Web Technologies*, CRC Press, Taylor & Fancis Group, Berlin Heidelberg.

Uschold, M. and Gruninger, M. (2004), "Ontologies and semantics for seamless connectivity", in *Sepcial Interest Group Management of Data International Conference*, *Paris*.

Wagner, A., Speiser, S. and Harth, A. (2010), "Semantic web technologies for a smart energy grid: requirements and challenges", in *International Semantic Web Conference*, *Shanghai*, Springer, Berlin.

Weijun, Q., Yuanchun, S. and Yue, S. (2007), "Ontology-based context-aware middleware for smart spaces", *Tsinghua Science and Technology*, Vol. 12 No. 6, pp. 707-713.

Yu, D.Y., Ferranti, E. and Hadeli, H. (2013), "An intelligent building that listens to your needs", in *Symposium on Applied Comuting 2013 Proceedings of the International Conference*, *Coimbra*, ACM, New York, NY.

Zhou, Q., Natarajan, S., Simmhan, Y. and Prasanna, V. (2012), "Semantic information modeling for emerging applications in smart grid", in *Information Technology: New Generations 2012 Ninth International Conference*, *Las Vegas, NV*.

About the authors

Sylva Girtelschmid is a part-time Researcher at the Institute of Application Oriented Knowledge Processing at JKU. She studied at Old Dominion University in Norfolk, Virginia, from where she received her bachelor's degree (2003) in Computer Science and Mathematics and master's degree (2005) in Computer Engineering with program specialization on Modeling and Simulation. For a few years, during and after her studies, she worked in Norfolk in the areas of medical and military simulation. In 2007, she joined the Graphics and Parallel Processing Institute at JKU, where she worked as an Assistant Researcher and Lecturer in Grid Computing and Visualization. She is an author and co-author of several published papers in the areas of Medical Simulation, Grid Computing, Visualization and Big Data. After maternity leave, she returned to JKU, shortly joining the Telecooperation Institute and later the Institute of Application Oriented Knowledge Processing. She is currently pursuing her PhD, orienting her research in Information Extraction with focus on Social Media crowdsourcing and its applications to Crisis Management.

Matthias Steinbauer is a Researcher and Lecturer at the Institute of Telecooperation, JKU, Linz, and a lecturer at the University of Applied Sciences of Upper Austria. He joined the Institute in 2012 and focuses on Cloud Computing systems and Big Data systems. He aims to leverage the computational power of such systems for socially intelligent applications such as enabling machines to determine social structures of very large work groups. He has organized the 2013 editions of the International Conference on Information Integration and Web-based Applications and Services and the International Conference on Advances in Mobile Computing and Multimedia. He has authored and co-authored several papers in international conferences and journals and he has been reviewing manuscripts for Massachusetts Institute of Technology (MIT) Press and other international publishers. Before joining JKU he studied Computer Science as a part-time student. He worked for several enterprises in different fields. Ranging from mobile-banking applications, Web-based banking applications over to real-estate management systems that are provided as Software as a Service application. He further worked as a Senior Consultant in the area of Business Process Management. Matthias Steinbauer is the corresponding author and can be contacted at: matthias.steinbauer@jku.at

Vikash Kumar is an Information Scientist at Inland Revenue, New Zealand. He received his doctoral degree from JKU, Linz, Austria (2013), while working as a Researcher at the FTW. In his PhD, he investigated the adaptation and re-use of semantic rules in ubiquitous semantic applications. He received his MS in Computational Logic jointly from Vienna University of Technology, Austria, and the Free University of Bolzano, Italy (2009), and a BTech in Computer Science from VIT University, Vellore, India (2006). For the past 6+ years, he has been actively contributing to research in the application of semantic technologies across various fields including compliance analysis, fraud detection, energy efficiency, e-commerce, location intelligence and natural language processing. His experience comes from participating actively in several European Union's (EU) and national projects, while working for leading research organizations in Germany, Austria, Italy and New Zealand. Most of his contributions can be searched online, mainly as papers published in various journals, conferences and workshop proceedings.

Anna Fensel is a Senior Researcher at FTW and at the University of Innsbruck, Austria. She has a doctoral degree in Informatics from the University of Innsbruck (2006) and a university degree in Mathematics and Computer Science from Novosibirsk State University, Russia (2003). She has been active in research on semantic technologies for more than a decade while working in Austria, UK and Ireland, led and took part at numerous international and national projects with industry involvement and (co-)authored more than 70 refereed papers in international journals and conference and workshop proceedings. She has been a co-organizer or a Program Committee member of more than 60 scientific events, a reviewer for numerous journals and a project proposals evaluator for European Commission. Her research interests include knowledge engineering and acquisition, data value chain, semantic services, social media and software and application of semantic technologies in fields such as energy efficiency, mobile communications, tourism and media.

Gabriele Kotsis received her master's degree (1991, honored with the Award of the Austrian Computer Society), her PhD (1995, honored with the Heinz-Zemanek Preis) and the venia docendi in Computer Science (2000) from the University of Vienna. She was working as a Researcher and Teacher at the University of Vienna (1991-2001), at the Vienna University for Economics and Business Administration (2001) and at the Copenhagen Business School (2002). Since December 2002, she is holding a full professor position at the Telecooperation Department at the JKU, Linz. Her research interests include performance management of computer systems and networks, workgroup computing, mobile and internet computing, telemedia and telecooperation. Kotsis is the author of numerous publications in international conferences and journals and is co-editor of several books. From April 2003 to April 2007, she was the President of the Austrian Computer Society. Since October 2007, she is Vice Rector for research at the JKU, Linz.