DYNAMIC DISTRIBUTED STORAGE ARCHITECTURE ON SMART GRIDS

Joan Navarro[1], José Enrique Armendáriz-Iñigo[2] and August Climent[1]
[1]*Distributed Systems Research Group, La Salle, Ramon Llull University, 08022 Barcelona, Spain*
[2]*Dpto. Ing. Matemática e Informática, Universidad Pública de Navarra, 31006 Pamplona, Spain*

Keywords: Dynamic systems, Eventual consistency, Smart grids, Replication.

Abstract: Most of the power network services such as voltage control, asset management, or flow monitoring are implemented within a centralized paradigm. Novel distributed power generation techniques—eolian fields or local solar panels—are decentralizing this generation paradigm and forcing companies to change their traditional centralized infrastructure. This implies that intelligence and data sources are now spread over the whole network. Smart grids may enable to manage such a change although any standard architecture to deploy them on a power network exists. This challenges researchers to design and implement a new distributed storage system able to offer different levels of consistency and replication depending on the physical location of the smart sensor and according to the network needs. This paper reviews the requirements of smart grids and presents a new dynamic storage architecture following the flavor of cloud computing. This architecture is based on a variant of the primary copy scheme and is suitable to store all needed data and enable smart grids to solve the required functions in a distributed way. Moreover, it is able to offer high scalability and a consistency level similar to the one required by wireless sensor networks.

1 INTRODUCTION

Power networks are demanded to be high reliable and available because they have to supply all the infrastructures of a country at anytime and anywhere. This prevents power companies from updating and improving their systems because most of the changes may seriously affect critical services they are currently providing since novel devices might not be as tested as older ones. This leads to inefficient—due to their centralized nature—schemes which are expensive and even harder to maintain and scale.

With the growth of renewable energies the power network centralized model not only scales but also cannot work properly; the aforementioned renewable energy sources behave different than traditional sources. Moreover, current power networks are not able to remotely monitor power consumptions on the low voltage (LV) network which prevents companies from building new business strategies fitted to the end user needs (Brown, 2008). This situation claims to a substantial change which consists of decentralizing the power network and building a distributed system able to fulfill the current society requirements and technologies.

Recently, this new paradigm has also been referred to as smart grid (intelligent grid). The goal of a smart grid is to take advantage of the current digital technologies and build up an intelligent information system over all devices within the power network: from suppliers to consumers. This might allow companies to tune the power distribution and route energy where and when it is needed.

The purpose of this paper is to focus on the computer engineering field and propose a distributed architecture able to efficiently store and ease the computation of any data generated by the power network. This distributed storage architecture must be slightly different than the ones used on web services (Paz et al., 2010) or in cloud computing based storage (White, Tom, 2009; Palankar and et al., 2008) since smart grids demand a set of requirements have not been explored yet.

2 STORAGE REQUIREMENTS

Smart grids, as opposite to classical power networks, have become data driven applications since they own a management layer which takes decisions based on

Navarro J., Enrique Armendáriz-Iñigo J. and Climent A..
DYNAMIC DISTRIBUTED STORAGE ARCHITECTURE ON SMART GRIDS.
DOI: 10.5220/0003507702180221
In *Proceedings of the 6th International Conference on Software and Database Technologies* (ICSOFT-2011), pages 218-221
ISBN: 978-989-8425-76-8

the current status of the network. This forces to redefine the whole power network architecture and its specifications as now there is a need for storing and processing data besides supplying power. Next we review such needs and state the basics of our proposal.

Smart grids demand a trade off between static (Patiño-Martínez and et al., 2005) and dynamic (Aguilera and et al., 2009; Das et al., 2010) systems because they behave as a dynamic system but they may need some strong consistency (Vogels, 2009) requirements that typical cloud based techniques are currently unable to offer. Hence, our proposal is to build a hybrid system which take advantage of both distributed system schemes, static and dynamic. Furthermore, there are several applications (also referred to as smart functions) that run over the smart grid, such as power flow monitoring, under/over voltage monitoring, load shedding, or fault analysis. Each application has its own particular requirements so the proposed architecture must be flexible enough to support such variety of functions. Thus, the distributed storage architecture must provide the following:

Reliability. It must be fully tested since major changes on it may imply eventual denial of services.

Availability. It also has to ensure that there always be available data despite its level of consistency.

Fault Tolerance and Recovery. It has to be able to reconfigure its internal characteristics in order to keep supplying and storing data in case of failure.

Dynamic Consistency. Smart functions that require different levels of consistency. For example, on one hand, data needed to perform a load shedding requires strong consistency (Vogels, 2009) since it performs critical operations with the current values of the network. On the other hand, data needed to perform power monitoring might require a weaker consistency level since this function tolerates some kind of delay.

Minimum Message Exchange. It is important to keep a low network overhead in order to guarantee that there were no bottlenecks, and data will flow over the network in an efficient way.

To fulfill these requeriments, we propose a distributed storage architecture built on top of the power network able to afford the dynamic behavior of smart grids (e.g., a solar panel may stop supplying energy).

3 SYSTEM ARCHITECTURE

As depicted in Figure 1 a smart grid is seen as a set of clusters linked by a telecommunications network. A cluster is composed of up to ten devices placed on the same geographical area. Each device has limited

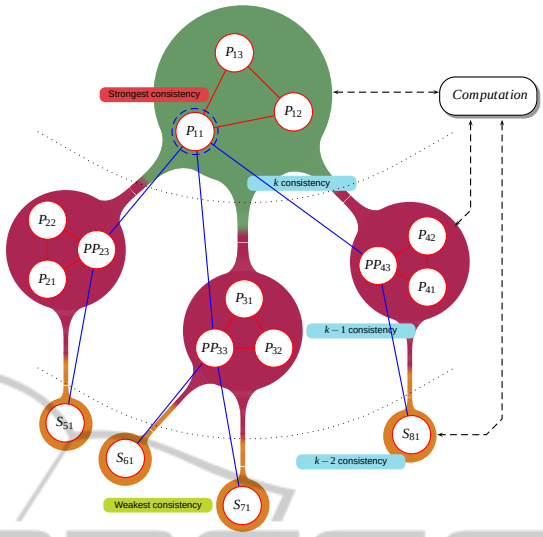

Figure 1: Proposed distributed storage system.

storage and computing capabilities and it might not be able to solve the whole required smart functions on its own. Smart meters are attached to these devices and report them their measurements from the smart grid.

Each device in the cluster is labeled as X_{ij} where X corresponds to the device role in the cluster (**P**rimary, **P**seudo-**P**rimary or **S**econdary); i is the cluster identifier, and j is the device identifier. In the same way, we define the ancestor of a cluster$_m$ as the node X_{ij} (that belongs to cluster$_i$ $(m \neq i)$) which is updating an arbitrary pseudo-primary k of this cluster (PP_{mk}). Figure 1 shows an example where we have that region 2 is formed by devices $P_{2k} \mid k = [1,3]$ where P_{21} is the primary of this region; P_{22} is a common device; P_{23} is the pseudo-primary, (that's why it is named $\mathbf{PP}_{23}$); and, its ancestor is P_{11}. Respectively, S_{61} is the only device on region 6 and its ancestor is PP_{33}.

Regarding data consistency, we define the replication depth r as the amount of different clusters that data are allowed to cross while being replicated. This value might be dynamically tuned according to the computation latency or the system performance.

Next, we describe the proposed architecture and explain how it solves the replication, consistency, and fault tolerance issues.

3.1 Architecture Overview

Although the number of smart sensors may substantially increase as time goes by, the number of devices that control them should not grow in the same way. The proposed architecture focuses on the devices instead of the smart meters which is an attempt to avoid scalability issues from the latter ones by hiding their dynamism. Any device belonging to a cluster

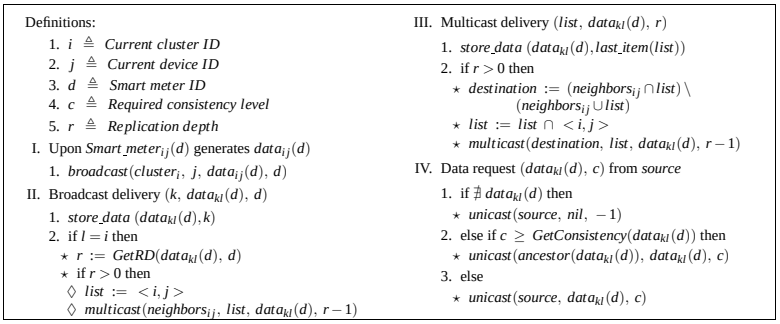

Figure 2: Replication protocol at smart device$_{ij}$.

may simultaneously adopt different roles according to the current situation: (1) primary master, (2) primary slave, (3) pseudo-primary. When a device is propagating data from their directly attached smart meters, it will act as a primary master and will treat the rest of devices in its cluster as their primary slaves. When a device receives data from another cluster it will be acting as a repeater (pseudo-primary). Blue lines just illustrate the particular case of P_{11} broadcasting data.

3.2 Replication

Replication provides availability and fault tolerance. However, it increases the number of messages since all replicas have to be synchronized which reduces the system throughput. Regarding the time when updates get propagated to the replicas there exist two major strategies; (1) eager replication (Bernstein and et al., 1987) provides strong consistency but poor scalability, and (2) lazy replication (Wiesmann and Schiper, 2005) provides higher scalability but has more difficulties to maintain consistency—i.e. replicas may diverge. Regarding the amount of sites that update data, there exist two major replication strategies; (1) active (Amir and Tutu, 2002) provides strong consistency since all replicas are synchronized but has low scalability, and (2) passive (Pedone et al., 2000) provides higher scalability but has some troubles on maintaining strong consistency since all replicas might be unsynchronized.

As shown in Figure 2, our proposal is a hybrid solution that performs (1) active and eager replication in the primary-master's cluster, and (2) passive and lazy replication in other clusters. This improves the scalability of classical architectures (Jiménez-Peris et al., 2002) and defines different consistency regions.

3.3 Consistency

Research on consistency protocols has been conducted for many years and several approaches have been proposed by the community. There are two major alternatives when defining the consistency properties of a system: (1) strong consistency and (2) weak consistency. Regarding our proposal, we take advantage of both strong and weak consistency strategies and propose a hybrid solution inspired by cloud-based storage and data stream warehouses (DeCandia et al., 2007; Golab and Johnson, 2011).

In the master's cluster we implement strong consistency between all replicas. This improves fault tolerance since another device of the cluster could easily take over from a primary-master's fault. Moreover, this avoids the typical single point of failure problem. Once data are strongly consistent in the master's cluster, devices start propagating them with a time stamp k to their pseudo-primaries. Therefore, we are currently implementing an eventually consistent system between the pseudo-primaries. To sum up, from the consistency point of view, we have shown how our hybrid architecture uses both strong and weak (actually k-weak) consistency techniques. Next we describe how our architecture deals with fault tolerance.

3.4 Fault Tolerance

Fault tolerance is the ability of the system to recover from a spontaneous site fault. Distributed systems are prone to different types of failures (Cristian, 1991). Since smart grids are hardly dependent on the communication network, we can assume that this channel will be reliable enough and focus our efforts on the distributed storage architecture. We also assume that any site may fail according to the crash model.

Regarding our proposal, there may exist two different failure cases: (1) the failure of a primary, and (2) the failure of a pseudo-primary. In the former, we inherit the advantages of the active replication techniques and are able to easily recover since any other primary-slave can immediately take over the situation. In the latter, as soon as the ancestor of the failed node belonging to another cluster, detects its unresponsiveness, it will select a new pseudo-primary.

4 DISCUSSION & CONCLUSIONS

As shown in the previous sections, our proposal takes benefit from many techniques used in distributed systems. However, these techniques have never been put together and neither tested and need to be discussed:

Master Cluster Reduction. Some members of the master region might be excluded from the active replication. Active replication does not scale well (Wiesmann and Schiper, 2005) and with the proper selection of representatives we could speed up this process.

Enhance the Takeover Process. A pseudo-primary could do active replication within its cluster. This role is not an exclusive one in the cluster, it can be also responsible for several smart meters and, thus, collaborate in the active replication protocol. Recall that there are not so many nodes in a given cluster.

Failure Detection. Active replication within a pseudo-primary cluster may (1) enhance the failure node detection process, and (2) speed up the synchronization of the new device in the replication chain.

Distributed Computing. Our proposed architecture allows to perform distributed computation on the read steps. Thanks to the fact that required data travel across the replication chain, each node might be able perform a piece of the computation required.

Dynamic Replication Depth Tuning. If we were able to dynamically adjust this value our system might adapt better to their requirements. Hence, we could use a cognitive system and apply some machine learning techniques (Mitchell, 1997) in order to (1) evaluate the whole system status and (2) predict the optimal value of the replication depth for each data item.

In this paper we have defined a way to distribute and store information across the network so that the computation needed for smart functions can be greatly reduced. This work aims to provide some insight into the world of smart grids from a data perspective. For the sake of simplicity during the presentation of our system, we have outlined simple scenarios about the replication policy or fault-tolerance issues that need to be treated in detail in further works.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union European Atomic Energy Community Seventh Framework Programme (FP7/2007-2013 FP7/2007-2011) under grant agreement n 247938 for Joan Navarro and August Climent and by the Spanish National Science Foundation (MEC) (grant TIN2009-14460-C03-02) for José Enrique Armendáriz-Iñigo.

REFERENCES

Aguilera, M. K. and et al. (2009). Sinfonia: A new paradigm for building scalable distributed systems. *ACM Trans. Comput. Syst.*, 27(3).

Amir, Y. and Tutu, C. (2002). From total order to database replication. In *ICDCS*, pages 494–.

Bernstein, P. A. and et al. (1987). *Concurrency Control and Recovery in Database Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Brown, R. E. (2008). Impact of Smart Grid on distribution system design. In *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, pages 1–4.

Cristian, F. (1991). Understanding fault-tolerant distributed systems. *Commun. ACM*, 34(2):56–78.

Das, S., Agrawal, D., and Abbadi, A. E. (2010). Elastras: An elastic transactional data store in the cloud. *CoRR*, abs/1008.3751.

DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., and Vogels, W. (2007). Dynamo: amazon's highly available key-value store. In *SOSP*, pages 205–220.

Golab, L. and Johnson, T. (2011). Consistency in a Stream Warehouse. In *CIDR*.

Jiménez-Peris, R., Patiño-Martínez, M., Kemme, B., and Alonso, G. (2002). Improving the scalability of fault-tolerant database clusters. In *ICDCS*, pages 477–484.

Mitchell, M. (1997). *An Introduction to Genetic Algorithms*. The MIT Press, Cambridge, Massachusetts.

Palankar, M. R. and et al. (2008). Amazon s3 for science grids: a viable solution? In *DADC '08: Proceedings of the 2008 international workshop on Data-aware distributed computing*, pages 55–64, New York, NY, USA. ACM.

Patiño-Martínez, M. and et al. (2005). MIDDLE-R: consistent database replication at the middleware level. *ACM Trans. Comput. Syst.*, 23(4):375–423.

Paz, A., Perez-Sorrosal, F., Patiño-Martínez, M., and Jiménez-Peris, R. (2010). Scalability evaluation of the replication support of jonas, an industrial j2ee application server. In *EDCC*, pages 55–60.

Pedone, F., Wiesmann, M., Schiper, A., Kemme, B., and Alonso, G. (2000). Understanding replication in databases and distributed systems. In *ICDCS*.

Vogels, W. (2009). Eventually consistent. *Commun. ACM*, 52(1):40–44.

White, Tom (2009). *Hadoop: The Definitive Guide*. O'Reilly Media, 1 edition.

Wiesmann, M. and Schiper, A. (2005). Comparison of database replication techniques based on total order broadcast. *IEEE Trans. Knowl. Data Eng.*, 17(4):551–566.