Cloud-based Replication Protocol on Power 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
`jnavarro@salle.url.edu,enrique.armendariz@unavarra.es,augc@salle.url.edu`

Abstract. Cloud based storage systems are known to provide high scalability and reliability overcoming the traditional constraints of static distributed systems. The processing capacity over thousands of machines makes this approach especially suitable for many environments. In particular, we focus on power networks. These systems are currently decentralizing their architectures due to the growth of renewable sources and the increasing power demand which are obstacles to the traditional power network radial distribution. This new decentralized architecture which has computing abilities for network monitoring and improving customer services is denoted as power smart grid. This paper proposes a new scalable dynamic storage architecture able to store data with different consistency levels that fits the physical power smart grid topology and its demands.

Keywords: Dynamic systems, cloud computing, eventual consistency, smart grids, replication.

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.

---

* 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.

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 [2]. This situation urges 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 power smart 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.

Moreover, both energy and the IT system have to meet high availability and reliability. From the information system point of view, a centralized architecture is not suitable since it would not be able to scale out as required. Thus, we propose a distributed storage system inspired by the cloud paradigm which stores data combining classical replication techniques and permits to perform distributed computing following the MapReduce style.

The decentralization of a power network, actually the smart grid design, covers several disciplines such as (1) electricity, there are multiple power sources using different technologies; (2) networking, there must exist a secure communication between all the nodes which generate data on the system; (3) computer engineering, in the sense that this data must be stored and computed.

The purpose of this paper is to focus on the computer engineering field and propose an architecture able to efficiently store and ease the computation of any data generated by the power network inspired by the flavors of cloud computing. This distributed storage architecture is slightly different than the ones used on web services [11] or in pure cloud computing based storage architectures [14, 9] since smart grids demand a set of requirements that have not been explored yet.

2 System Architecture

Smart grids demand a trade off between static [10] and dynamic [1, 4] systems because they behave as a dynamic system but they may need some strong consistency [13] 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. As depicted in Fig. 1 a smart grid is seen as a set of clusters linked by a telecommunications network (i.e. power line communication, wireless network, wide area networks, etc.). A cluster is composed of up to ten devices placed on the same geographical area. Each device has limited storage and computing capabilities since it might not be able to solve the whole required smart functions on its own. Smart meters are attached to these devices and rely on them to report their measurements to the smart grid.

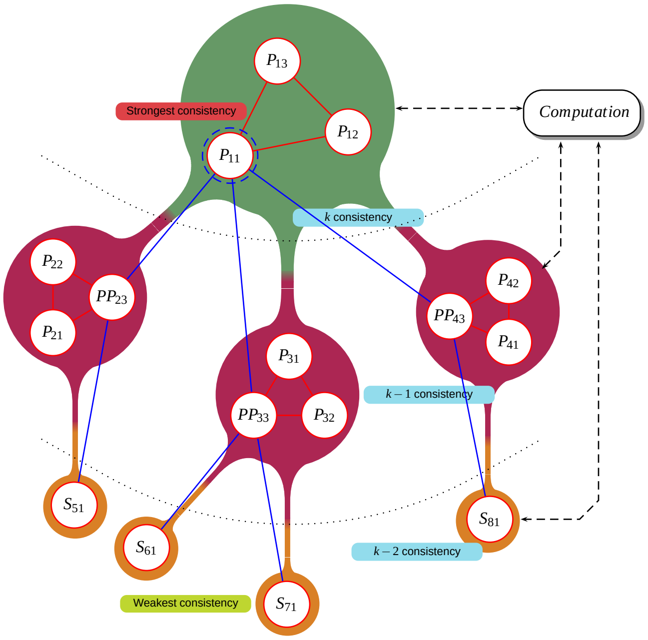

Fig. 1. Proposed distributed storage system.

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

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. However, the system must be robust against possible node failures which forces designers to implement some techniques commonly used in dynamic systems [6, 4, 1].

In order to provide a high available and fault tolerant system able to ease the smart functions distributed computation, we propose an architecture inspired on the Primary Copy [7, 15, 12] (also referred to as Primary Backup) scheme.

Any device belonging to an active cluster may simultaneously adopt different roles according to the current situation: (1) primary master, (2) primary slave, (3) pseudo-primary, and (4) secondary. 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 (in Fig. 1, P_{11}, marked with a dashed blue circle, is the primary master and P_{12}, P_{13} are their primary slaves). When a device receives data from another cluster it will be acting as a repeater (pseudo-primary) (in Fig. 1 PP_{23}, PP_{33}, and PP_{43} are the pseudo-primaries of P_{11}). Moreover, if a device receives updates from other clusters but it do not propagate them, it will be acting as a secondary (in Fig. 1, $S_{51}, S_{61}, S_{71}, S_{81}$ behave as secondary devices).

Smart grids need to compute many smart functions [3] indeed. Our architecture is flexible enough and able to adapt itself to the data freshness requirements [12] imposed by each smart function. Hence, the primary master of a region (i.e., P_{11}), after actively replicating its data to nodes within its cluster, must decide when to passively replicate its data to the pseudo-primaries of the neighboring regions (P_{21}, P_{31}, P_{41}). At the same time, each pseudo-primary has to take the same decision with its data and their pseudo-primaries or secondaries. These decisions must be taken according to (1) the function periodicity (i.e., flow monitoring will require faster updates than asset management), (2) link status and congestion, and (3) cluster status (i.e., a general master might decide to asynchronously replicate its data when it detects that there are very few alive nodes on its cluster).

Actually, once the primary master has sent its data to a pseudo-primary node of another cluster, a recursive process starts where each pseudo-primary looks for another pseudo-primary in another neighboring cluster to propagate its data. This process finishes when there are no more clusters or there is a cluster which has no more neighbors (i.e., S_{51}, S_{61}, S_{71}, and S_{81}). The decision process must be aware of not falling in cluster loops and ensure that in each cluster there is only one pseudo-primary node that contains data from the primary master.

Each time the computation unit of the smart grid needs to calculate the result of a given smart function, it first attempts to use data from its nearest neighboring cluster. If data contained on that cluster has a consistency level k greater than l, where l is the freshness level required by the function, it will use that cluster to perform computation. Otherwise, it will get redirected to its ancestor node with a freshness index $k-1$ and repeat the operation.

3 Discussion and Conclusion

Our proposal takes benefit from many techniques used in distributed systems. However, to the best of our knowledge, these techniques have never been put together nor tested. Hence, there are several aspects that have been intentionally left out and need to be discussed:

Master cluster reduction. Not all the members of the master region have to participate on active replication. Along this work, we have assumed that all nodes of a given primary cluster participate in the active replication of all data. In fact, this is not necessary at all: although the number of nodes belonging to a zone can be in the range of tens, we think that we can speed up the replication process by selecting a set of representatives for each subset of smart meters. It is well known that active replication does not scale well [15] and with the proper selection of representatives the rest can become secondaries of each representative.

Enhancing the takeover process. A pseudo-primary (PP) could do active replication within its associated 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.

Failure detection. It makes sense to think that inside the cluster of a given pseudo-primary (or secondary) the active replication among their nodes would detect the failure of the pseudo-primary (or another device). If so, they can agree with selecting a new pseudo primary in that cluster and notify the ancestor about this fact. Again, we have to reconstruct the new hierarchy tree in order to add the new pseudo-primary. However, this would overload the pseudo-primaries' clusters and might worsen the global performance of the system.

State transference under fault. We could perform a partial state transfer of data. This implies that the replication algorithm has to ensure that each pseudo primary has a set of secondaries in its associated cluster where the updates get also propagated asynchronously. Therefore, when a given pseudo-primary fails, it is only needed to transfer a much less amount of data to the new pseudo-primary than in the case of full state transfer. Nevertheless, this has to be tested and checked in order to find out the proper number of pseudo-primary slaves and the amount of data transferred per round.

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 (depending on the required consistency level k) each node might be able perform a piece of the computation required.

Actually, as suggested in [5] for cloud computing environments, with our architecture it would be possible to implement something similar to MapReduce; where the node owning the required data version run the *map* tasks and the rest of nodes in the replication chain continuously run the *reduce* tasks. Such distributed computation not only might reduce the size of the traveling data but also improve the computation throughput of the smart grid.

Dynamic replication depth tuning. We believe that if we were able to dynamically adjust this value our system might adapt better to their requirements. In fact, we have not specifically stated how the replication depth is set, neither augmented or decreased. It can be adjusted by the system administrator but it can also be dynamically adjusted as a function of the demands coming from the computation unit. Moreover, it can be tuned autonomously in case of disaster or rapid evaluation of certain functions (e.g. accounting). Further, there might exist certain information that would need to be replicated in all nodes as it is rarely modified.

To this end, we are thinking about a cognitive system [8] based on supervised learning in order to (1) evaluate the whole system status and (2) predict the optimal value of the replication depth for each data item. Actually, we are implementing a learning classifier system (e.g., XCS or UCS) [16] able to adapt itself to the system dynamism due its online nature.

In conclusion, this paper presents a novelty approach to take advantage of smart electric grids. We have defined a way to distribute and store information across the network so that the computation needed for certain 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.

References

1. Aguilera, M.K., et al.: Sinfonia: A new paradigm for building scalable distributed systems. ACM Trans. Comput. Syst. 27(3) (2009)
2. Brown, R.E.: 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. pp. 1–4 (2008), `http://dx.doi.org/10.1109/PES.2008.4596843`
3. Chuang, A., McGranaghan, M.: Functions of a local controller to coordinate distributed resources in a smart grid. In: Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE. pp. 1 –6 (2008)
4. Das, S., Agrawal, D., Abbadi, A.E.: Elastras: An elastic transactional data store in the cloud. CoRR abs/1008.3751 (2010)
5. Dean, J., Ghemawat, S.: Mapreduce: a flexible data processing tool. Commun. ACM 53(1), 72–77 (2010)
6. DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., Vogels, W.: Dynamo: amazon's highly available key-value store. In: SOSP. pp. 205–220 (2007)
7. Gray, J., et al.: The dangers of replication and a solution. In: Jagadish, H.V., Mumick, I.S. (eds.) Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, June 4-6, 1996. pp. 173–182. ACM Press (1996)
8. Holland, J.: Adaptation in natural and artificial systems. The MIT Press, second edn. (1992)
9. Palankar, M.R., Iamnitchi, A., Ripeanu, M., Garfinkel, S.: Amazon s3 for science grids: a viable solution? In: DADC '08: Proceedings of the 2008 international workshop on Data-aware distributed computing. pp. 55–64. ACM, New York, NY, USA (2008)
10. Patiño-Martínez, M., Jiménez-Peris, R., Kemme, B., Alonso, G.: Middle-r: Consistent database replication at the middleware level. ACM Trans. Comput. Syst. 23(4), 375–423 (2005)
11. Paz, A., Perez-Sorrosal, F., Patiño-Martínez, M., Jiménez-Peris, R.: Scalability evaluation of the replication support of jonas, an industrial j2ee application server. In: EDCC. pp. 55–60. IEEE Computer Society (2010)
12. Plattner, C., Wapf, A., Alonso, G.: Searching in time. In: SIGMOD Conference. pp. 754–756 (2006)
13. Vogels, W.: Eventually consistent. Commun. ACM 52(1), 40–44 (2009)
14. White, Tom: Hadoop: The Definitive Guide. O'Reilly Media, 1 edn. (June 2009)
15. Wiesmann, M., Schiper, A.: Comparison of database replication techniques based on total order broadcast. IEEE Trans. Knowl. Data Eng. 17(4), 551–566 (2005)
16. Wilson, S.: Classifier fitness based on accuracy. Tech. rep., The Rowland Institute for Science, 100 Edwin H. Land Blvd. Cambridge, MA 02142 (Apr 1995)