Efficient PMU Data Dissemination in Smart Grid

Kedar Khandeparkar, Pratik Patre,
Swadesh Jain, Krithi Ramamritham
Indian Institute of Technology, Mumbai, India
{kedark,pbpatre11,swadeshjain,krithi}@cse.iitb.ac.in

Rajeev Gupta
IBM Research, India
grajeev@in.ibm.com

ABSTRACT

Next-generation Smart Grid will be a highly data driven system with sensors deployed across the grid and analytics being performed on the data collected for intelligent and timely decision making. This paper proposes novel techniques for filtering as much data as possible early in the dissemination network so that applications get the required data without imposing exorbitant bandwidth requirements while ensuring low latency.

1. INTRODUCTION

A power grid is a large complex network of power generation, transmission, and distribution systems. The demands for increased reliability, distributed generation, integration of renewable sources, and requirements of making the grid more stable are pushing the development of the Smart Power Grid. A large number of monitoring and sensing devices, Phasor Measurement Units (PMUs), are being deployed throughout the transmission network that provide a continuous stream of informative data by providing GPS synchronized measurements of various electrical signals such as voltage, current, phase angles, frequencies, etc. Phasor Data Concentrators (PDCs) at one or more levels integrate and aggregate the PMU data. Lower level PDCs (LPDCs) aggregate data from PMUs which are geographically located at different places, time align the data, and send the aggregated data to the applications at higher levels or super PDCs (SPDCs). Different monitoring and control applications perform computations on the sensed data to make intelligent decisions and ensure efficient utilization and reliable operation of the grid. However given the large volume and velocity of the data generated by the sensors, collecting all the data at the centralized site continuously for application execution is not scalable. It may result in wastage of bandwidth and increase the latencies, which can affect the fulfillment of QoS requirements associated with the applications.

In this paper,we primarily focus on the question, *How do we design data dissemination techniques to efficiently share the communication infrastructure between applications with vastly different QoS requirements such that the QoS requirements of all applications are satisfied?* We address the problem of executing smart grid applications by designing application specific PMU data dissemination techniques without affecting the QoS requirements of the applications. The proposed techniques not only reduce the processing time at the application site but also reduce the network traffic by adopting data filtering techniques. We describe how application semantics can be used to design intelligent data dissemination techniques. We take angular stability monitoring as an example application and show the effectiveness of semantic aware data dissemination using real PMU data for Northern Indian grid.

e-Energy'14, June 11–13, 2014, Cambridge, UK.
ACM 978-1-4503-2819-7/14/06.
http://dx.doi.org/10.1145/2602044.2602079 .

2. SEMANTICS-AWARE DATA FILTERING AND COMMUNICATION

We need the semantics-aware distributed query processing approach to achieve the better utilization of communication infrastructure. Semantics are of two kinds: data semantics– involving physical meaning of data, their characteristics, etc., and application semantics– whether the application is monitoring and control application or an operator display application, what the application priorities are, how the application semantics maps to data semantics, etc.

We now summarize the three high level approaches we consider in this paper. These approaches differ with respect to the distribution of application execution and the nature of data forwarded to higher level PDCs.

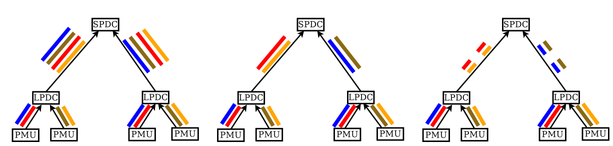

Figure 1: Message transmission in CUT, CQT, and DQT

2.1 CUT

A typical approach for application execution is the **C**entralized execution with **U**nqualified data forwarding **T**echnique (CUT). Here applications execute at a single site (say, SPDC) and, as shown in Figure 1(a), LPDCs forward all the data received from downstream PMUs to SPDC, irrespective of whether the data is relevant to the application. CUT requires static bandwidth reservation. Clearly this approach wastes data bandwidth. We need to design application data needs aware communication solutions which would flexibly share the bandwidth between various applications while meeting the data requirements of various applications.

2.2 CQT

Exploiting knowledge of which data is required for which application leads to the **C**entralized execution with **Q**ualified data forwarding **T**echnique (CQT). In this case, LPDCs transfer only some of the streams to the higher level PDCs. For CQT, we need to identify the data requirements at every application node. These requirements are disseminated to various PDCs and routers. An LPDC receives all the data from PMUs. This data can be seen as a number of streams from each PMU, say, one each for 3 voltage phases, frequency, phasors, etc. As part of CQT, each PDC or router will forward only a sub-set of these streams depending on the application semantics. Thus, each router selects a sub-set of streams, aggregates the data based on their time-stamp, if possible, and these combined packets are forwarded to the higher level PDCs or SPDC.

2.3 DQT

In **D**istributed execution with **Q**ualified data forwarding **T**echnique (DQT), the application at SPDC is divided into coordinated sub-applications (sub-queries) and executed in a distributed manner at LPDCs. We show that it is sufficient to monitor these sub-queries, over one data stream or data from a limited geographical region, to monitor the application query. These sub-applications forward the relevant filtered and aggregated data from LPDCs to SPDC, as shown in Figure 1(c). This helps in filtering a large amount of data before it is collected at the site. This in-network query processing is application semantics-aware and hence achieves high amount of filtering when *the system is steady* and also ensures detection of any unstable situation when the application is state is changing. In the next section we describe angular stability as the example application and show how the DQT scheme can be implemented for this application.

3. ANGULAR STABILITY

In angular stability application, we need to monitor multiple buses and send alert if the difference in the phase angles is above the specified threshold. It should be noted that, if f is the frequency of the electrical signal, phase angle θ_t at time t can be modeled as:

$$\theta_t = f \times t \times 360^\circ + \theta_0 \; (mod\, 360^\circ)$$

where, f is the phasor frequency in Hz, t is measurement time in seconds, and θ_0 is the phase angle at time $t = 0$. If the frequencies for PMU_i and PMU_j, corresponding to θ^i and θ^j, are equal ($f_i = f = f_j$), individual values of phase angles, with time, will be parallel to each other.

In DQT, rather than sending all the phase values to the application, thresholds to the phase angles are determined, such that LPDC needs to sends the phase angle values to the SPDC only if it crosses the specified thresholds. Since, phase angle values vary with time, rather than a fixed threshold, we need to have time varying thresholds over individual values of phase angles which bounds the angles with an upper bound and a lower bound. These two upper and lower bound constraints should vary linearly with time, should be parallel to each other, and the distance between them should be less than or equal to the value of phase angle difference threshold (say, θ_{Th}).

Let us assume that, for a pair of phase angles at two ends of a bus, initially θ^i is greater than θ^j. If β_i is the distance of the upper bound from linear equation estimating θ^i_t and β_j is the distance of the lower bound from linear equation estimating θ^j_t, we get,

$$\beta_i + \beta_j = \theta^{\mathbf{ij}}_{\mathbf{Th}} - (\theta^i_0 - \theta^j_0) \qquad (1)$$

We can then set different values for β_i and β_j. Thus the linear constraints, for both θ^i and θ^j, are:

$$\theta^j_0 - \beta_j \le (\theta^i - f \times t \times 360)\; (mod\, 360^\circ) \le \theta^i_0 + \beta_i$$
$$\theta^j_0 - \beta_j \le (\theta^j - f \times t \times 360)\; (mod\, 360^\circ) \le \theta^i_0 + \beta_i \qquad (2)$$

The values of β_i and β_j can be dynamically obtained using various methods outlined in [1]. If the frequencies of the buses are different, the linear estimation models of phase angles at both the buses are not parallel. In this method, we derive the equation of the linear bounding constraints based on the average frequency ($f_{avg} = (f_i + f_j)/2$), i.e., use f_{avg} in the above equation while dividing the β values in proportion to the corresponding frequency values.

4. PERFORMANCE EVALUATION

For our experiments, we emulated the PMUs and PDCs using a grid simulator [2]. We implemented angular stability monitoring queries at SPDC where the SPDC receives data from 2 local PDCs and determined whether the difference in the phase angles is above the specified threshold. We used PMU data from a Northern-Indian grid line for these experiments. This data included the data when grid was stable as well as during a blackout. The buses (PMUs) were located at: Agra, Bassi, Dadri, Kanpur, Moga, Mumbai, and Hisar; mostly in northern Indian towns which were affected by a blackout during 30^{th} and 31^{st} July 2012. PMU data rate was 50 Hz and data frame size was 74 bytes. which had 16 bytes of header and 58 bytes of payload (3 voltage phasors, 3 current phasors, 1 frequency, 1 angle). We emulated the three methods of data dissemination, CUT, CQT, and DQT.

Average data rates from LPDC to SPDC, in Kb/s were, 586.4, 186.4, 158.4 for the three approaches CUT, CQT, and DQT, respectively. As expected, CUT approach incurs the highest bandwidth consumption compared to CQT and DQT. Average latency to get the application results at SPDC was 2881 μsecs for CUT. Reduction in data requirement also leads to reduction in total latency for CQT and DQT: CQT approach reduces latency by 15% whereas the DQT approach reduces average latency by as much as 80%.

5. CONCLUSION AND FUTURE WORK

We designed various in-network query processing techniques which allow for flexible bandwidth sharing of real time applications. The techniques are application semantics-aware due to which high data filtering is achieved when the grid was stable. This leads to significant message reduction which translated to latency reduction for critical applications. In the future, we will be exploring whether other aggregation processing techniques can be explored depending on the specifics of a given query over smart grid data

Acknowledgments

We would like to acknowledge contributions by Gopal, Prof. Soman, and Prof. Kulkarni from EE department at IIT Bombay in understanding the electric grid.

6. REFERENCES

[1] R. Gupta and K. Ramamritham, "Distributed execution of continuous queries," in *ICDE 2014*, 2014.

[2] K. Khandeparkar and N. Pandit, "Design and implementation of IEEE c37.118 based phaser data concentrator and pmu simulator for wide area measurement system," *Technical Report, Department of Electrical Engineering, IIT, Bombay.*, May, 2012.