EMERGING TECH CONFERENCE – Edge Intelligence

Volume 01, 2022, Page 87 – 94

**Proceedings of Emerging Tech Conference:
Edge Intelligence 2022**

Enabling the future smart grid: an edge computing approach

Eleftherios Mylonas*, Nikolaos Tzanis, Michael Birbas, Alexios Birbas

Electrical and Computer Engineering Department, University of Patras, 26504 Patras, Greece

e.mylonas@ece.upatras.gr

Abstract

Advances in the field of edge computing and the emergence of edge-cloud has enabled the monitoring and management of one of the most complex existing systems, the electric grid, which is undergoing a decentralized and 'smart transition' towards 'smart grid'. Advanced Monitoring Infrastructure (AMI) and sensing devices (PMUs, RTUs etc.) combined with advanced communications networking allow the monitoring of the grid parameters, while edge computing capabilities (including hardware acceleration engines, reconfigurable hardware, etc.) enable real-time Transient State Estimation (TSE) of the grid nodes and therefore enabling other smart grid applications, such as preventive maintenance, control and data analytics. In this work, we focus on the processing requirements of TSE and we present an edge-cloud architecture for efficient TSE application. The architecture leverages low-cost hardware acceleration engines at the edge (devices enhanced with FPGA resources) in order to solve the TSE equations effectively and in tight time thresholds, while reducing the transmission of traffic towards the cloud and thus saving bandwidth for more sophisticated smart grid applications.

1. Introduction

The advent of the digital age was mainly driven by the ubiquity of the internet and the rise of the cloud computing paradigm, which rapidly shifted the way people used to work until then. This change led to the emergence of many so-called smart applications and introduced new domains, such as Industry 4.0, Smart Health, Smart Agriculture, Smart Home, etc. However, this new paradigm did not come without introducing any concerns. Sensitive user data had to be disclosed to few cloud providers and transmitted at remote data centers far beyond the vicinity of the user, thus increasing risks of data privacy and security. In addition, those constant data transmissions at the cloud induced high latencies in the applications, therefore limiting the number and types of cloud services that data centers could provide and inhibiting the growth of smart applications. Furthermore, due to the nature of data centers which constitute of hundreds and hundreds of rack- type Commercial-off-the-self (COTS) servers, it was not possible to provide performance and power optimized services at the same time. All the above reasons gave birth to new computing paradigms, essentially edge computing, fog computing and nowadays edge-cloud computing. Overall, these paradigms are filling the missing parts of cloud computing by bringing smart applications closer to the network edge where the data sources are located. Together with the rise of Artificial Intelligence (AI) and Machine Learning (ML) algorithms, as well as the access to many open-source AI platforms and AI- enabled hardware commodities, this enables a wide range of new smart applications.

One application domain that has received a lot of attention lately and is often associated with the concept of edge-cloud computing and edge intelligence in general is that of smart grid. The term “smart grid” refers to the digitalization of the classical power grid and specifically its twin transition (digital and green) towards an enhanced Industry 4.0-oriented system. Such a power grid features a number of advantages over its predecessor, e.g. more robustness, reliability, and security throughout the power network by enabling smart, real-time management and control [1]. With the current energy crisis situation, the need for the inclusion of more Distributed Energy Resources (DERs) under a secure power grid is ever-increasing and the smart grid concept promises to provide solutions to this problem. One such solution that enables fast and reliable grid monitoring both at the transmission and at the medium-voltage distribution level is called Transient State Estimation (TSE). TSE is a set of state estimation algorithms that enable capturing transient events throughout the system under study. It has been proven to work under different kinds of systems with a time resolution of down to tens of microseconds, with the necessity for fast measurement units installed throughout it. With the current progress in Advanced Monitoring Infrastructure (AMI) for power grids and the development of fast recording devices named Phasor Measurement Units (PMUs), TSE is a possible solution for fast monitoring and control actions across the power network [2]. The usefulness of TSE is further highlighted when combined with the edge-cloud computing paradigm in order to form Distributed TSE, otherwise known as DTSE.

In this work, a proposed edge-cloud architecture for TSE is presented. In Section 2, we analyze the concept of TSE, we examine different implementation parameters and finally we present the DTSE approach used in our work for decreasing TSE’s complexity. In Section 3, the proposed architecture for TSE application in smart grids is presented and its vital components are also addressed. In Section 4, the proposed architecture is tested and evaluated in terms of performance and end-to-end (E2E) latency with the help of a modified IEEE 30- bus test system simulated at Simulink/Matlab. Lastly, Section 5 summarizes the benefits of the proposed architecture and concludes this study.

2. TSE Analysis

This section describes the TSE concept in full detail, the basic computational and networking requirements it imposes, the different ways to decrease its complexity and increase its overall performance, and finally the DTSE approach which holds the most promise.

2.1 TSE Description

As described earlier, TSE methods focus on capturing the system state under very tight time granularity constraints. The events they capture can be in the order of tens of microseconds. This is rather useful for the power grid as it enables dealing with power electronics transients that inject harmonics at the power network and/or cause overvoltage/undervoltage situations, which are a threat to grid stability. That is, TSE output data can be fed to controllers or other smart grid services, enabling advanced power grid applications. Although TSE was considered too expensive and essentially unfeasible for the grid, current PMUs with reporting rates faster than 10ms have changed this fact. Nowadays, more and more PMUs are getting installed at the transmission and medium-voltage level, providing great sources of valuable information.

In general, TSE methods involve the design of a time domain system solution and the dynamic formulation of system components, with the latter usually based on the Numerical Integration Substitution (NIS) method [3] that stands out as a relatively simple and effective transformation. The TSE problem can be generally expressed as:

$$z = Hx + \varepsilon \tag{1}$$

where z is a (m × 1) vector of measured quantities, x is a (n× 1) vector of state variables (unknown quantities) for which the equation must be solved, H is a (m × n) measurement function relating the known quantities to state variables and ε is the vector of measurement errors. To build the above equation, a dynamic model representing the system is needed for which the Dommel approach [4] is adopted. This method gives a numerical solution for the transient behaviour of a dynamic system at discrete time points by applying an NIS algorithm in order to solve the differential equation describing the behaviour of each lumped circuit element connected between two network nodes k, m (Hkm) thus leading to their equivalent circuit. In this way, the H matrix in (1) is gradually filled and remains constant for all iterations [1]. The H matrix cells can be calculated as shown in Table 1 for a classic power grid. The TSE equation is solved for the calculation of state variables (node voltages).

2.2 TSE Implementation Requirements

TSE methods can be rather computationally expensive if not properly designed. One vital part of TSE is arguably the NIS algorithm or integrator chosen for discretizing the system components and equations. There are different NIS algorithms that can be used for implementing TSE. The most simple and common integrators are the forward Euler, backward Euler and trapezoidal. From those three integrators, the latter is recommended as it is relatively simple and accurate enough due to its better tracking abilities. Table 2 describes the discrete components for the basic circuit elements in accordance to the trapezoidal integrator, respectively the effective conductance Geff and the current History Term. Another fact that one should consider before actually implementing TSE is the computation of the inverse H matrix. In most cases, the H matrix is sparse and non-square meaning that no inverse matrix can be calculated or it will probably be ill-conditioned. A valid solution to this problem is the use of Singular Value Decomposition (SVD) algorithm, which deals with ill- conditioned matrices and enables the calculation of a pseudo- inverse H matrix. SVD is relatively robust and copes with under-determined systems [5], making it ideal for TSE.

The challenge of solving TSE equations does not lie only on the actual type of arithmetic computations that need to be solved, but rather on the large number of TSE equations and the tight time threshold in which the task must be performed. For this reason, the adoption of hardware acceleration engines (GPUs, FPGA) was proposed, due to their ability to tackle hard real-time computing tasks by leveraging, among others, their inherent parallelism [6]. Although this is an overall valid solution computation-wise, it cannot ensure stable TSE operation on a real power grid scenario. This is if we consider how power grids operate and how they integrate PMUs in the overall system architecture. As shown in Fig. 1, PMUs are installed at different locations in the grid and monitor the system state at a local level. When they capture new data, they convert them to their respected phasor units and transmit them to the upper layers, where special Phasor Data Concentrators (PDCs) collect them, format them to an appropriate Protocol Data Unit (PDU) and forward them towards the Wide Area Monitoring System (WAMS). This process induces delays which in some circumstances may be disastrous for a TSE- enabled application, such as real-time fault/event detection. For this reason, the TSE concept does not fit the cloud computing paradigm.

Measurement Type	Measurement vector [z]	Measurement Matrix [H]	State variables [x]
Resistor	$v_k(t)$	[1 0]	[$v_k(t)$]
Inductor	$v_k(t)$- $v_m(t)$	[1 -1]	[$v_k(t)$ $v_m(t)$]
Capacitor	$i_{km}(t)_{History}$	[Geff -Geff]	[$v_k(t)$ $v_m(t)$]

Table 1: TSE matrix construction

Element	Geff	History Term
Resistor	1/R	-
Inductor	$\Delta t/2L$	$i_{mn}(t-\Delta t)$ + Geff * $v_{mn}(t-\Delta t)$
Capacitor	$2C/\Delta t$	$i_{mn}(t-\Delta t)$ + Geff * $v_{mn}(t-\Delta t)$

Table 2: Norton equivalent of basic elements

2.3 DTSE Description

In order to tackle this problem, the DTSE approach is proposed. DTSE is based on the partitioning of the power grid under study into a number of subgroups (hereafter, DTSE groups). Each DTSE group is modelled as an independent TSE problem as described earlier, has much lower computational complexity and therefore can be solved near the edge of the power grid in low-cost embedded devices. This alleviates the problem of installing power-hungry expensive equipment at the WAMS, as well as minimizing network delays and increasing bandwidth availability in the overall system. The most ideal equipment for the DTSE deployment is low-cost embedded devices featuring heterogeneous System-on-Chips (SoCs), as their hardware acceleration capabilities can be utilized for timely TSE operation, while their general-purpose Central Processing Unit (CPU) can execute other smart grid applications, such as generating alarm signals for a local controller in a substation or higher-level controllers.

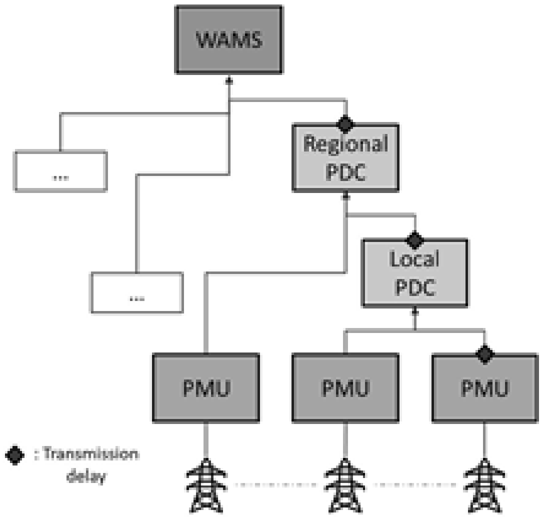

Figure 1: Power Grid Architecture

3. Edge-cloud Architecture for TSE

This section describes the proposed TSE-oriented edge- cloud architecture and explains the major TSE elements that were chosen for hardware acceleration.

3.1 Proposed architecture

The edge-cloud architecture for TSE application in smart grids is shown in Fig. 2. The power grid under study is partitioned into a number of DTSE groups. Each group and its respected PMUs are associated with a specific PDC which is located inside a local substation. The PMU measurements are sent to the PDC via fiber or 5G connection for low latency transmission and are then forwarded to the hardware acceleration engines inside the substation. There, the DTSE group model is solved, and alarm signals arise in case of abnormal state detection. The alarm signals can be used internally at the substation to trigger a controller action (edge) or be sent to higher-level controllers at the WAMS level for a more system-wide action (cloud). For this architecture to work, we assume that PMUs are operating as in [2]. That is, each PMU packet contains synchronized measurements of 80 consecutive timestamps, equally distributed in time. That means that for a reporting rate of 10ms, each measurement data is time-sampled at 125μs.

3.2 Hardware-accelerated tasks for DTSE

In order to solve the DTSE equations at the tight time threshold of 10ms and enable other smart grid applications, two tasks are implemented in the hardware accelerators, namely the DTSE task and the Event Capturing task. For each task, a dedicated hardware module is created, whose functionality is summarized below. The DTSE module implements the core functionality of the solution. The relation matrix H (1) of each DTSE subgroup is centrally computed prior to the service deployment and is forwarded to the respective edge resource in the form of a hardware description file. This is feasible because the relation matrix remains constant during the service execution if no major change in the power grid is performed. The precomputed relation matrix of each DTSE group is stored and for every new set of measurements, the algorithm produces a new set of outputs representing the transient state of the power grid group.

The TSE application receives synchronized measurements at a certain time point and provides estimation for system quantities such as voltage and current at different network nodes for the same time point, bringing information from non- observable nodes as well. This information is then fed to the Event Capturing module which is responsible for identifying whether the system is operating normally or under transient conditions (towards a network fault). When capturing an abnormal scenario, the Event Capturing module forwards the information to a local controller for fault mitigation at the local level, leading to substantial reduction in response time of the controller. Measurements and events are logged at a local database and are forwarded to the WAMS cloud periodically for later use (for other smart grid applications or system-wide fault mitigation).

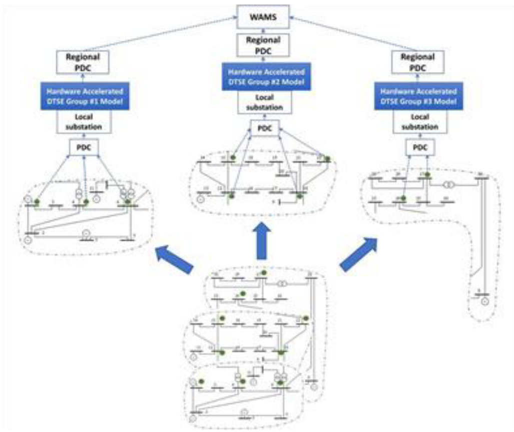

Figure 2: Proposed Architecture for TSE

4. Case Study, Experimentation & Results

This section describes the experimentations carried out for validation purposes of the proposed TSE-oriented edge-cloud architecture and the hardware-accelerated components of TSE. Firstly, we analyze the benchmark model we used in our tests, then we describe the different software and hardware platforms used and finally we provide the relevant results.

4.1 Benchmark model

The modified IEEE 30-bus test system, whose single-phase diagram is shown in Fig. 2, is used to validate the proposed architecture. As it can be seen in Fig. 2, PMU devices (green circles) are placed throughout the grid in such an order so as to obtain full observability of the system. The system is simulated in Mathworks Simulink employing the Simscape library [7]. The quantities of interest are logged throughout the simulation and are divided into two categories. The first category comprises the measured quantities (nodal voltages, branch currents) of observable nodes and represent the measurements originating from the measuring units. The second category comprises the unobservable locations and serves for the evaluation of the DTSE algorithm. The system is segmented into three DTSE subgroups, each one served by a different H matrix and a different set of measurement units.

4.2 Experimentation Setup

In this specific test scenario, we implement the DTSE and Event Capturing tasks in two different types of platforms, one at the cloud and one at the edge of the power grid and proceed to test them in terms of latency and accuracy. For this analysis, we use a standard COTS server with an Intel Core i7-4800MQ and 32GB RAM in order to simulate a cloud server located at the WAMS level. In addition, we select the Zybo Z7-20 board as our edge device. Zybo Z7-20 is centered around the Zynq- 7000 System-on-Chip (SoC) and combines a dual-core ARM Cortex-A9 with a traditional FPGA, 1GB RAM and a Gigabit Ethernet interface [8]. The ARM processor can support both bare metal applications and RTOS–based ones for the execution of hard real-time computations. The FPGA serves as hardware acceleration engine of the proposed TSE algorithm. For the emulation of the measurement units, we use the Simulink simulator in playback mode, which transmits packets of logged measurements to the cloud or the edge.

4.3 Results

Following the TSE formulation procedure, the H matrices of each DTSE group are computed and implemented in hardware logic in Zybo Z7-20 edge devices. The two hardware- accelerated modules are described in VHDL and DTSE operations are mapped into 34-bit fixed-point arithmetic operations. The same H matrices are also used at the cloud server, where all tasks run in software with 64-bit floating- point precision. Fig. 3 illustrates the difference between the two solutions for the three phase voltages and currents at a node of interest. Both solutions exhibit similar results (average difference less than 0.1 Volts and Amperes for RMS values of 7KV and 2.8KA respectively), showcasing the adequate accuracy of the hardware implementation.

The results of the mapping of the proposed TSE architecture on FPGA hardware resources are shown in Table 3. Table 4 gives the execution time for the processing of 80 instantaneous measurements included in a single PMU packet. The time is measured from the point where measurements are received and stored at the input buffer of the DTSE module, so the communication delay between the Processing System (PS) and the Programmable Logic (PL) is not taken under consideration. We can observe a significant 10000X speedup between the fully software (cloud) and hardware-accelerated (edge) approach. Hence, it is clear that the proposed edge-cloud architecture is valid for TSE calculations, as the computation- intensive tasks are offloaded to the hardware-acceleration engine at the edge, while the PS is free for other smart grid tasks.

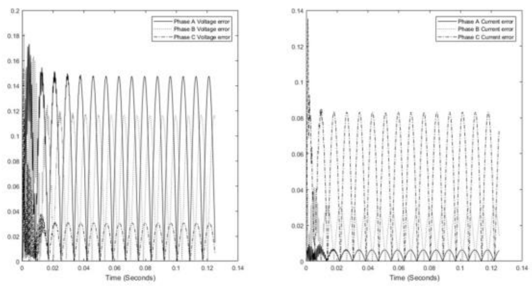

Figure 3: Accuracy comparison between cloud & edge TSE solutions

Resource Type	Used	Available	Utilization (%)
Slice LUTs	42,862	53,200	80.57
Slice Registers	3570	106,400	3.36
DSPs (DSP48E1)	96	220	43.64

*Clock Frequency = 28 MHz

Table 3: Norton equivalent of basic elements

Execution Environment	Execution Time
x86 cloud server	30.4ms
Zybo Z7-20 board	0.0029ms

Table 4: Timing Summary

5. Conclusion

We have shown that TSE which is required for the real time monitoring of the transient behavior of modern power grids presents excessive, communication and processing needs. Advanced metering devices, such as PMUs, and modern communication technologies respond to the metering and communications requirements by feeding the TSE application with massive, synchronized measurements in a timely manner. In this paper, we focused on the computing requirements of TSE and presented an edge-cloud architecture featuring hardware acceleration engines at the edge for efficient TSE application. We showed that by applying our solution at a lowcost Zybo Z7-20 MPSoC board, and by leveraging the inherent parallelism provided by FPGAs, we can achieve a 10000X speedup for the calculation of the transient state of a IEEE 30- bus system thus meeting real-time requirements, without sacrificing accuracy. This is achieved with minimum overhead to the PS of the system, allowing it to serve other smart grid applications. The efficiency of the proposed solution is more evident when the complexity of the power grid scales up (and the complexity of the H matrix increases) since the edge computation latency remains relative constant as long there are adequate hardware resources.

Acknowledgments

This work has been partially supported by the projects: 5GVICTORI (grant agreement No. 857201) & EnerMan (grant agreement No. 958478).

References

[1] N. Tzanis, N. Andriopoulos, G. Proiskos, M. Birbas, A. Birbas and E. Housos, "Computationally efficient representation of energy grid-cyber physical system," 2018 IEEE Industrial Cyber-Physical Systems (ICPS), St. Petersburg, 2018, pp. 679-683.

[2] Yang X (2013) Distributed state estimation with the measurements of phasor measurement units. University of Birmingham.

[3] Watson, N.R., Arrillaga, J.: 'Power systems electromagnetic transients simulation IEE Power Energy Ser. 39, 449 (2003).

[4] Dommel, H.W. EMTP Theory Book; Microtran Power System Analysis Corporation: Vancouver, BC, Canada, 1996.

[5] I. Molina-Moreno, A. Medina, R. Cisneros-Magaña and O. Anaya-Lara, "A Methodology for Transient State Estimation Based on Numerical Derivatives, Optimal Monitoring, and Filtered Measurements," in IEEE Transactions on Power Delivery, vol. 33, no. 4, pp. 1527-1535, Aug. 2018.

[6] J. Mahseredjian, "Computation of power system transients: overview and challenges," in 2007 IEEE Power Engineering Society General Meeting, 2007, pp. 1–7.

[7] Simscape, "The math works inc. simscape electrical—user's guide (specialized power systems)," https://www.mathworks.com/help/pdfdoc/physmod/sps/powersys.pdf, [Accessed; 22 August 2022].

[8] Xilinx, "Ug-585 zynq-7000 soc technical reference manual," https://www.xilinx.com/content/dam/xilinx/support/documentation/user guides/ug585-Zynq-7000-TRM.pdf, 2021, [Accessed; 22 August 2022].