OPQ Version 2: Architecture for Distributed, Real-Time, High Performance Power Data Acquisition, Analysis, and Visualization

Anthony J. Christe, Sergey I. Negrashov, Philip M. Johnson, Dylan Nakahodo, David Badke, David Aghalarpour
Collaborative Software Development Lab
University of Hawaii at Manoa

***Abstract*—OpenPowerQuality (OPQ) is a framework that supports end-to-end capture, analysis, and visualizations of distributed real-time power quality (PQ) data. Version 2 of OPQ builds on version 1 by providing higher sampling rates, optional battery backup, end-to-end security, GPS synchronization, pluggable analysis, and a real-time visualization framework. The OPQ project makes real-time distributed power measurements which allows users to see both local PQ events and grid-wide PQ events. The OPQ project is split into three major parts: back-end hardware for making power measurements, middleware for data acquisition and analysis, and a front-end providing visualizations. OPQBox2 is a hardware platform that takes PQ measurements, provides onboard analysis, and securely transfers data to our middleware. The OPQ middleware performs filtering on the OPQBox2 sensor data and performs high-level PQ analysis. The results of our PQ analysis and real-time state of the sensor network are displayed using OPQView. We've collected distributed PQ data from locations across Oahu, Hawaii and have demonstrated our ability to detect both local and grid-wide power quality events.**

I. Introduction

As power grids transition from a centralized generation model to a distributed model, maintaining a stability requires fine grained knowledge of grid's the state.[1] Monitoring PQ on a distributed generation smartgrid requires a consumer level distributed sensor network. How do we collect, analyze, and visualize power quality (PQ) on the power grid scale using data gathered from residential utility customers?

To answer this question, we developed and deployed an open source hardware and software system focused on consumer level monitoring across Oahu called Open Power Quality2(OPQ2). OPQ2 is able to aggregate distributed PQ measurements, perform high level real-time PQ analysis and classification, and display PQ at both the consumer the grid level. OPQ2 is a multi-layered framework consisting of a custom hardware for collecting distributed PQ data (OPQBox2), a middleware for filtering and event detection(OPQMakai), higher level PQ analysis(OPQMauka), and finally a front-end for consumer friendly grid level PQ visualizations(OPQView). OPQ2 architecture is shown in figure I.

Each OPQBox2 processes 2.7 billion samples per day. Using node level feature extraction OPQBox2's forward only 500,000 measurements to the cloud service in the same 24 hours. Further analysis and filtering yields roughly 1 distributed PQ event per day. These events contain data from every devices on the network sampled at full resolution and sampling rate. Through the use of our multi-tiered systems we aim to bring the operators and end users, useful, intuitive, actionable, real-time information about the power grid.

II. OPQ Version 2 Architecture

A. OPQBox2

OPQBox2 is a modern, high performance, power quality monitor built with distributed measurements in mind. OPQBox2 provides high resolution sampled data of up to 50kS/s at 16bits. Optional GPS synchronization and battery backup can be added to the OPQBox2 to tailor it to a specific power quality measurement.

While the sampling and GPS synchronization is controlled by the realtime DSP, all of the signal processing and communication is controlled by a Raspberry PI single board computer. The Raspberry PI collects 10 AC cycles of ADC measurements at a time and computes the utility frequency and V_{rms}. These values are sent to the cloud triggering broker via WiFi while the raw waveforms are buffered in the local Redis key value store.

If the triggering data stream indicated a power quality event, our middelware may request raw waveforms from the affected of OPQBox2s. When the triggering server requests data from an OpqBox2, it requests a time range. This data is queried from Redis, serialized into a single large data packet, and transfered back to the requesting server. All of the communication between the OPQBox2 and the cloud services are done over an encrypted ZeroMQ connection.[2]

Time synchronization of the OPQBox2 is performed via Network Time Protocol (NTP).[3] While OPQBox2's are able to synchronize via GPS, we found NTP to be suitable for our system. In this paper we verify the synchronization provided by Network Time Protocol (NTP) by comparing frequency measurements collected from two OPQBox2 devices and examining their phase difference.

B. OPQHub

OPQHub is OPQ's middleware system and is responsible for collecting raw and feature extracted data from OPQBox2's and for performing high level PQ analysis. OPQHub is split into two components, OPQMakai, a pluggable filtering and data

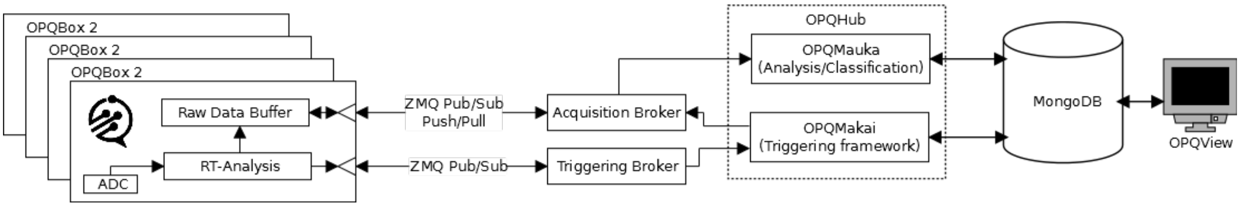

Fig. 1. OPQ System Architecture

request component and OPQMauka, a pluggable real-time PQ analysis component.

All communication between the OPQBox2 and OPQHub passes through two connection brokers. These brokers allow us to terminate encryption at the cloud boundary, thus simplifying the analysis framework. ZeroMQ is used throughout our cloud infrastructure to connect the analysis plugins to brokers and other plugins. This allows us to form ad-hoc analysis topologies, such pipeline processing, and map reduce pipelines without interfering with regular data acquisition.

Triggering stream consists of the feature reduced data(frequency and V_{rms}) for each device in the OPQ2 network. This stream is brokered via the Triggering broker, and analyzed by the set of analysis plugins called Makai. If multiple devices show temporally coherent deviation from the norm, Makai will request the raw data from OPQBox2 devices via the Acquisition broker.

Further analysis of the raw waveform is performed by OPQMauka analysis and classification system. Each of the plugins in OPQMauka are responsible for different PQ classification and analysis tasks. We currently implement plugins that detect and report PQ measurements, voltage sags and swells, frequency sags and swells, and perform ITIC event classification of voltage events. We've also developed plugins that, along with the filtering framework OPQMakai, allow us to detect these events on a distributed level. Data products from Mauka are stored in MongoDB and presented to clinets use OPQView.

C. OPQView

OPQView serves as the front-end user interface of the OPQ system. It is implemented in JavaScript using MeteorJS. OPQView serves as the end-point of OPQ2, where power data eventually propagates to for user consumption. As such, OPQView is responsible for displaying collected power quality data in a useful and meaningful way. OPQView maintains a communications interface with OPQHub via a MongoDB database, from which it is able to reactively retrieve real-time PQ data. This data includes voltages and frequencies measured by individual devices, as well as PQ events and their corresponding waveforms. OPQView does not only serve as a front-end to display real-time data. It can also display historical grid trends and power quality events. The real-time nature of our data allows for a wide potential of visualization techniques - such as real-time heat maps of device voltages and frequencies across the grid. OPQView also supplies the administrative interface for OPQBox2's, allowing device owners to adjust their device's privacy and sharing settings as needed.

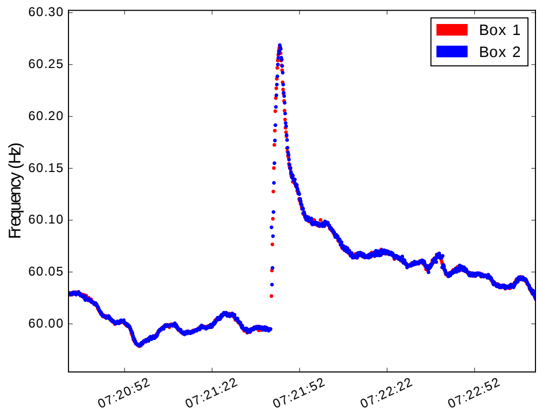

Fig. 2. Frequency event. March 1st 2017.

III. Results

Currently OPQHub is capable of detecting and recording frequency and voltage based PQ events. Voltage based events include sags, swells. Frequency events are limited to frequency deviation from the 60Hz nominal. Example of a triggering stream of a frequency event is shown in Figure 2. This event occurred during the lighting storm March 1st 2017, and is likely a lighting strike. Two devices were separated by 5 miles and were connected to different substations. Frequency deviations on the order of $\frac{1}{4}$Hz, can lead to load sheading.[4] A process where a section of the power grid is disconnected from the utility in order to preserve grid stability. Collection and analysis of events such as this one will ultimately lead to insights into power grid operation, and give utility customers a more active role in power grid monitoring.

References

[1] T. Peffer, "Evaluation of challenges and potential applications of building-to-grid implementation," *California Institute for Energy and the Environment Whitepaper*, Oct. 2010.

[2] P. Fengping and C. Jianzheng, "Distributed system based on zeromq," *Electronic Test*, vol. 7, no. 7, pp. 24–29, 2012.

[3] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on communications*, vol. 39, no. 10, pp. 1482–1493, 1991.

[4] W. C. New, "Load shedding, load restoration and generator protection using solid-state and electromechanical underfrequency relays," *General Electrics White Paper*, 2014.