OPQ Version 2: An 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
Oahu, Hawaii
(achriste, sin8, johnson, dylankhn, davidrb, aghalarp) @ hawaii.edu

***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 triggering on the OPQBox2 sensor network 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 distribution model to a distributed model, maintaining a stable grid requires fine grained knowledge of how distributed renewables are affecting the state of the grid. 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.

Our system is able to aggregate distributed PQ measurements, perform high level real-time PQ analysis and classification, and display PQ at both the consumer level and the grid level. OPQ is a multi-layered framework consisting of a custom hardware for collecting distributed PQ data, a middleware for filtering and event detection, higher level PQ analysis, and finally a front-end for consumer friendly local and grid level PQ visualizations. The architecture is visualized in 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 1 distributed PQ event per day. Through the use of our multi-tiered systems we aim to bring the operators and end users, useful, intuitive, actable, real-time information about the state of the 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.

The cloud service may request raw waveforms from the collection of OPQBox2s if the triggering data stream indicated a power quality event. When the triggering server requests data from an OpqBox2, it requests a range of data. This data is queried in Redis, reserialized into a single large data packet, and transfered back to the requesting server over the same ZeroMQ channel that it arrived on. All of the communication between the OPQBox2 and the cloud services are done over an encrypted ZeroMQ connection.

Time synchronization of the OPQBox2 is performed via Network Time Protocol (NTP). While OPQBox2 has the capability to use 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.

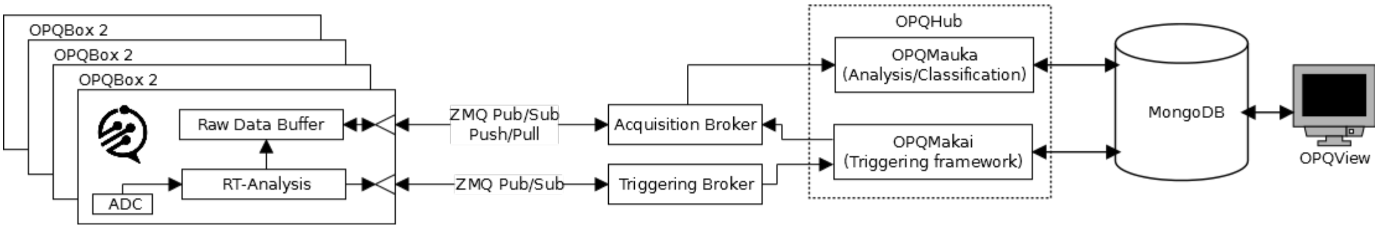

Fig. 1. OPQ System Architecture

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

All communication between the OPQBox2 and the analysis stack 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 as pipeline of 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 the 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 triggering 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 via MeteorJS.

OPQView serves as the end-point of this system, 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 manner. OPQView does not receive its data directly from OPQBoxes, but rather through the intermediary OPQHub. OPQView maintains a communications interface with OPQHub through the use of a MongoDB database, from which it is able to reactively retrieve real-time power quality data. This data includes voltages and frequencies measured by individual devices, as well as PQ events and their corresponding waveform data. OPQView not only serves as a front-end to display real-time data, but also a platform to safely and securely store a complete history of PQ data. This allows users to display historical PQ event data. In addition to local event data, users can also view distributed events. Over time, OPQHub aims to determine patterns in these distributed events, which would allow OPQView to determine 'communities' of devices with similar PQ characteristics. The real-time nature of our data allows for a wide potential of visualization techniques - including real-time heat maps of device voltages and frequencies across the grid. OPQView also supplies the administrative interface for OPQBoxes, allowing device owners to adjust their device's privacy and sharing settings as needed.

III. Results

Curretly OPQHub is capable of detecting and recording frequency and voltage based events. Voltage based events include sags swells and flicker. Frequency events are limited to frequency deviation from the nominal. Example of a frequency event is shown in Figure 2, this event occurred during the lighting storm March 1st 2017. Two devices were separated by 5 miles and were connected to different separate substations.

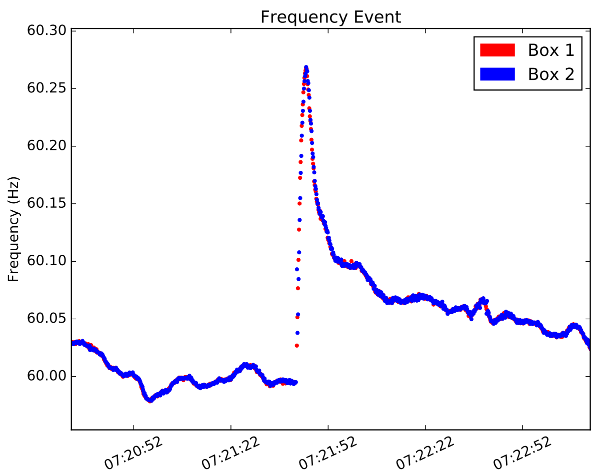

Fig. 2. Frequency event.