GRID PROTECTION ALLIANCE

Pending Improvements

openXDA Version 2.3 ; Open PQ Dashboard Version 2.1

Stephen Wills & Billy Ernest

April 25, 2018

Pending Improvements with EPRI Funding

- Enhancement of the openXDA file-watcher
- Large reduction in openXDA data base size.
- Loading data from PQ View into openXDA

Enhancement of openXDA Filewatcher

- For Windows server hosted file-shares, openXDA uses OS functionality as the primary process to detect and process new files.
- For file-shares hosted by other OS's, openXDA enumerates all files on a schedule (e.g., every 5 minutes) and discovers any files that it has not already processed.
- This file enumeration process has been stable for some storage appliances, but in at least one instance it "sticks" from time-to-time.
- To enable the openXDA service to oversee (monitor, alarm and take action) file enumeration, this process has been pulled out as a separate Windows service – with it's own console and command set.
- All configuration data is hosted in the openXDA database and loaded via the webAPI – so that the filewatcher service can run on a separate server without direct access to the openXDA database.
- Status: Enhancement is code complete and being tested.

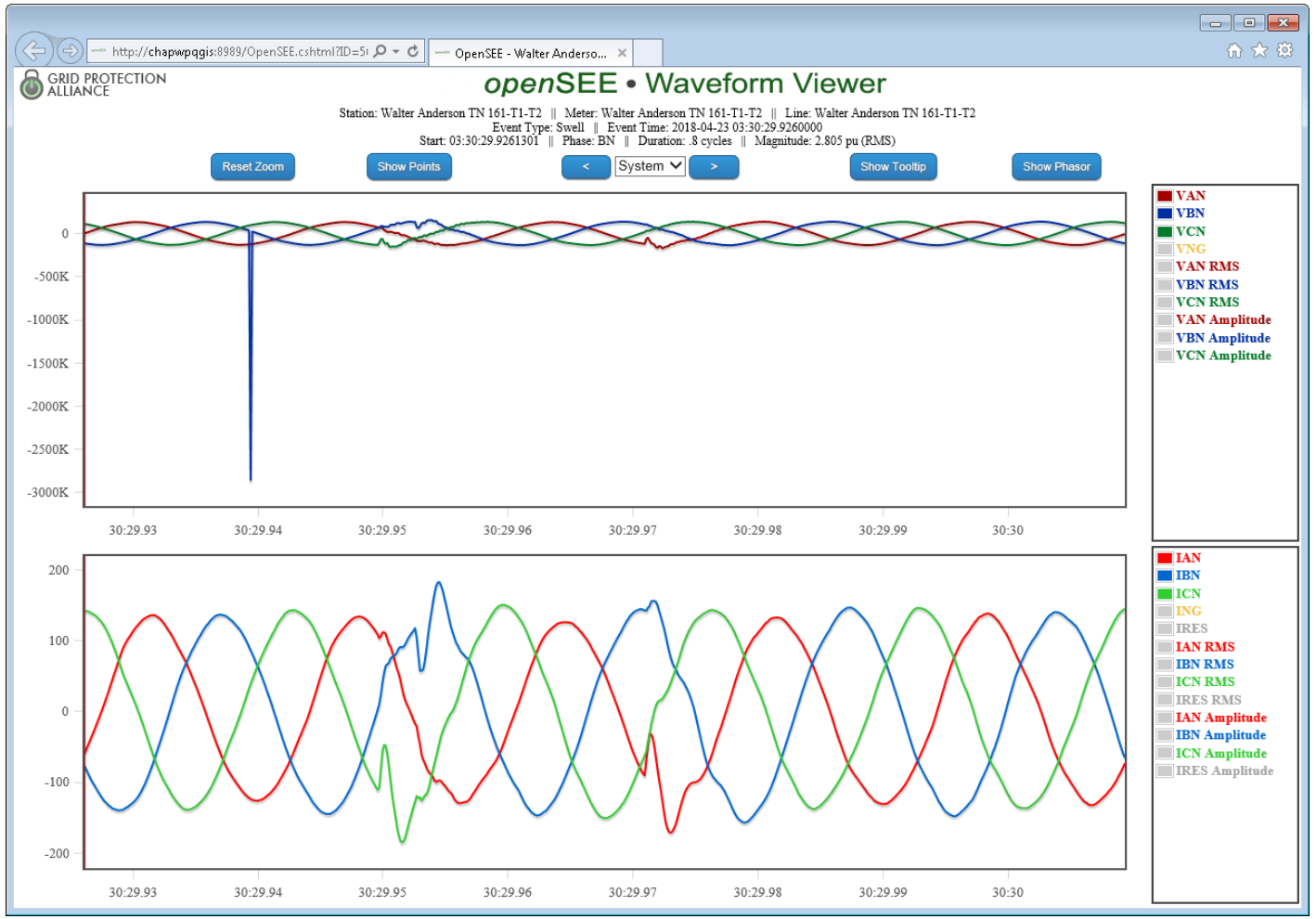
openXDA Database Storage Requirement Reduction

- Frequency domain data (e.g. RMS values) will be computed at the time of rendering and are no longer stored in the data base.
- Time domain data is saved using a new compression algorithm.
  - Saving this time domain data facilitates “reprocessing”
- Data base space reductions will exceed 85%.
  Our bench for ~50K files, from 114 GB to 14GB (88%)
- Current impact – about 1.5 seconds in openSEE data load times
- Future impact – near zero. openSEE will load and render data progressively.
- Status: Code complete and tested.

Loading data from PQ View into openXDA

- In early April, Dranetz released an updated version of the PQ View 4 SDK with documentation.
- Since then, GPA has worked with TVA to load openXDA using this SDK.
- The data flow is purposely strongly decoupled to avoid impacts to either PQ View or openXDA. The GPA's PQ-View-Data-Exporter (PQ-View-DE) tool is an independent service.
  - PQ View "site IDs" where data transfer is desired are loaded into openXDA
  - PQ-View-DE queries PQ View via the SDK to load configuration data for meters at these sites
  - On a schedule (e.g., every 5 minutes),
    - PQ-View-DE looks for new data in PQ View
    - It produces PQDIF formatted data as it pulls data from PQ View.
    - It places the PQDIF "blob" of data in the openXDA database where openXDA processes it immediately following load.

TVA PQ View Data Loaded into openXDA



Observations on use of the PQ View SDK

- The documentation is not great. (But who are we to be critical?)
- From testing, the SDK is only fully compatible with 32-bit assemblies
- Some actions are slower – e.g., 1 minute.
  - Querying the event list for all sites. e.g., what happened yesterday?
- Some actions are fast – e.g., sub-second.
  - Querying the event list for a single site.
- Licensing for use is a bit complex
  - The SDK requires a (1) “valid” license applied to the server where the code will be running and (2) a user-provided certificate.
  - Obtaining a “valid” license requires running a tool on this server – and ElectroTek will return a license for this server.
  - In addition, the application assembly must be signed with a certificate whose public key is included in the license. For testing, ElectroTek provides this certificate.
  - This approach assures that the SDK can only be used on servers of which they are aware and only by developers of which they are aware.

So how could this work in practice?

- GPA hopes to make the PQ-View-DE service available as a closed-source solution to any one that requests it.
- The user will run PQ-View tools on the desired hosting platform.
- GPA will provide the key that was used to sign our assembly.
- The user will request a license from Dranetz – based on the GPA key and PQ View tool result.