WAVEFORM SERVER:
WEB-BASED ACCESS TO NEAR REAL-TIME AND ARCHIVED HIGH-DENSITY SEISMIC WAVEFORM DATA: CYBER INFRASTRUCTURE CHALLENGES & DEVELOPMENTS IN THE OPEN-SOURCE WAVEFORM SERVER

Reyes, J.C. <reyes@ucsd.edu>, Newman, R.L. and Vernon, F.L.

Array Network Facility, Scripps Institution of Oceanography, University of California San Diego, La Jolla, CA 92093-0225

Introduction

The Waveform Server is an interactive web-based interface to multi-station and multi-channel high-density seismic waveforms stored in Center for Seismic Studies (CSS) 3.0 schema relational databases (Newman et al., 2009). In the last twelve months, based on expanded specifications and current user feedback, both the server-side infrastructure and client-side interface have been extensively rewritten.

In addition to supporting the single database model, the Python server-side code has been fundamentally modified to retrieve data using Python Deferred Objects in a multi-threaded architecture and access data stored in cluster-based databases. This allows interactive web-based access to high-density (+200Hz) waveform data that can span multiple years; the common lifespan of broadband seismic networks.

The client-side interface expands on it's use of simple JSON-based AJAX queries to now incorporate a variety of User Interface (UI) improvements including standardized calendars for defining time ranges, applying on-the-fly data calibration and unit representation , and time-zone correction.

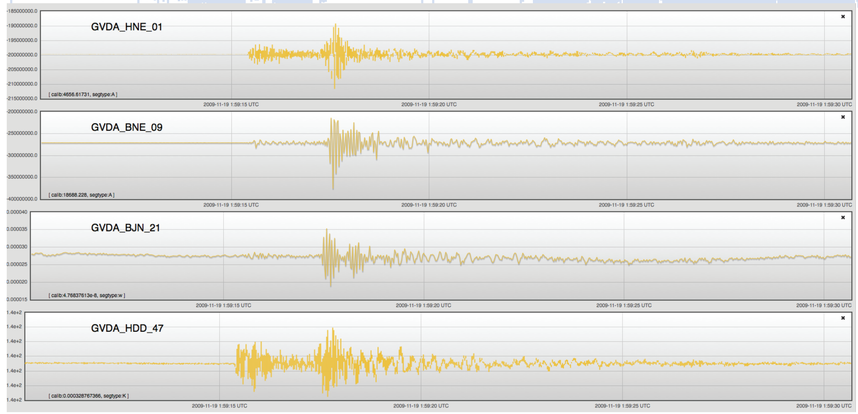

Figure 5. High and Low Density Plots

Efficient rendering of high and low density traces. Channels BNE and BJN are 40 Hz channels returned as JSON object less than 25 KB containing 819 2-tuple elements in the format [time,value]. Channels HNE and HDD are 200 Hz channels returned as JSON objects of less than 33 KB containing 819 3-tuple elements in the format [time,max-v,min-v].

Data for plot provided by the Institute for Crustal Studies, University of California Santa Barbara, Santa Barbara, CA 93106-1100

Development

JSON

JavaScript Object Notation (JSON) is a lightweight text-based open standard designed for human-readable data interchange. Originally derived from the JavaScript programming language, now it is language-independent with parsers available for many programming languages. It lacks semantical meaning by definition but JSON has a much smaller grammar and maps more directly onto the data structures used in modern programming languages. This translates to faster parsing and processing of the data structure.

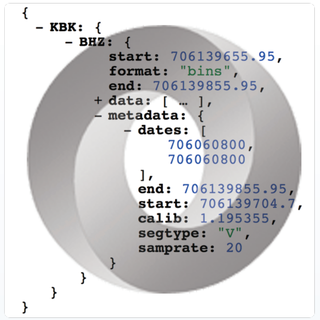

Figure 1. JSON format

Example of JSON object return to the client after each query. The "data" element contains the 2-tuple or 3-tuple arrays for the traces depending on the "format" value which could be set to bins or lines. The server automatically determines the type comparing the total points for the requested trace to the configure maximum. The "metadata" element contains all the information available in the database for that channel.

Dbcentral

Many institutions have divided their datasets into multiple independent databases, for example databases segmented by day, week, month or year. The dbcluster schema provides the logic to reference those databases. To properly handle this paradigm the server will initialize the database in a new Dbcentral object capable of returning the database pointer that will correspond to the asynchronously requested data.

jQuery

jQuery is a cross-browser JavaScript library that provides abstraction into low-level client-side scripting and enables easy traversal and modification of Document Object Model (DOM) elements, Cascading Style Sheets (CSS) manipulation, event handling, window effects and asynchronous requests of data. In the new version we expanded the use of jQuery with the implementation of the jQuery User Interface (jQuery UI).

jQuery UI provides advanced interactive plugins for high-level visual effects and themeable widgets that are flexible and user-friendly. The control window (Figure 4), the station and channel selection windows (Figure 3) and the calendar (Figure 2) are some examples of the use of this library.

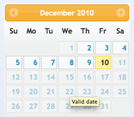

Figure 2. jQuery UI calendar

Figure 3. Source Selection Window

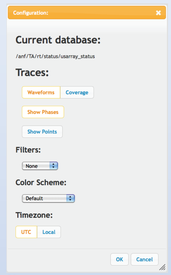

Figure 4. Configuration Window

Deferred Objects

A key concept in the Twisted application model is the deferred object. This object encapsulates an instruction that will produce a result in the future and contains a series of functions that will be applied to the returned value. Our new implementation of the server leverages on the use of deferred objects and threaded function calls to spawn a child process for every asynchronous request. The Deferred object then hands the results of each callback or errback function to the next function in the chain and returns results to the client allowing parallel processing of asynchronous requests. This multiplexed analysis of queries allows a non-blocking processing of client requests.

AJAX

All data and metadata is retrieved using the Asynchronous JavaScript and XML (AJAX) technique from the client to the server. All of this background bidirectional communication does not interfere with the behavior of the loaded page. AJAX leverages the HTML and CSS mark up and style information to dynamically display, and allow the user to interact with, the information presented. The asynchronous, callback-style retrieval of data allows the client to populate each waveform independently; drastically decreasing the total loading and rendering time of the application.

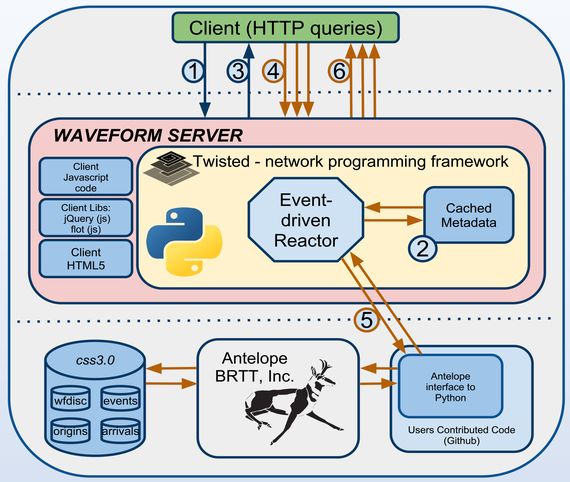

Figure 6. Query life-cycle

1) Client (i.e. web browser) initiates the process by querying the server with a user provided URI.

2) The Reactor interprets the URI using the cached metadata and produces a meta-query object that contains references to the requested data.

3) This meta-query object gets injected in the HTML application that will return to the browser.

4) JavaScript code on the browser prepares the application (HTML5) through the modification of DOM and CSS elements and initiates independent asynchronous queries to the server to populate the canvas elements with the waveforms.

5) The Reactor will produce Deferred Objects by sequencing the necessary steps to produce the data. Each Deferred Object gets its own thread and the data retrieval process continues out of the Reactor.

5) Each Deferred Object will return independent JSON objects in parallel to the client.

Performance

Parallelization of the retrieval and rendering of the data proved to be the key element in achieving satisfactory speed of the application. Multi-threaded analysis of the queries by the server enables the system to support multiple users simultaneously. The implementation of commonly used libraries enables the use of content distribution networks, i.e. Google Libraries API, for the collection of widgets used by the application. This decreases the total amount of code that needs to be maintain and updated.

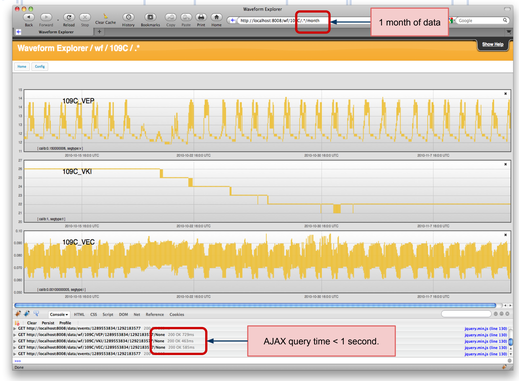

Figure 7. Improve Performance

Future development

1. Real time live interface to streaming waveforms from a Object Ring Buffer (ORB).
2. Automate the installation of server-side library dependancies.
3. Promote the development of new clients that can use the server as a gateway to the databases.

Download

You can download the code used in this presentation from the online Git repository hosted by Github. http://github.com/antelopeusersgroup/antelope_contrib

References

Newman, R.L., Clemesha, A., Lindquist, K.G., Reyes, J.C., Steidl, J.H. and Vernon, F.L., (2009)The Waveform Server: A web-based interactive seismic wavefrom interface, AGU, IN41A-1133

Lindquist, K.G., Clemesha, A., Newman, R.L. & Vernon, F.L., (2008), The Python Interface to Antelope and Applications, Eos Trans. AGU, 89(53), Fall Meet. Suppl., Abstract G43A-0671

Flot; http://code.google.com/p/flot

jQuery; http://www.jquery.com

Python; http://www.python.org

Twisted; http://www.twistedmatrix.com

JSON; http://www.json.org/

Google library API; http://code.google.com/apis/libraries/

Acknowledgements

Funding for this project was provided by the Incorporated Research Institutions of Seismology (IRIS) and the Institute for Crustal Studies at the University of California Santa Barbara (UCSB).