Quake-Catcher Network – Client and Server Software

Carl Christensen carlgt1@yahoo.com
Chief Software Architect, Quake-Catcher Network
School of Earth Sciences, Stanford University
The Quake-Catcher Network

Prof. Jesse Lawrence (Stanford); Prof. Elizabeth Cochran (UC-Riverside)

- Sensors report seismic events ("triggers") over the Internet to our servers via internal (laptop) or external (USB) sensors
- "Opposite" the usual volunteer computing projects with high computing (100% CPU) requirements
- The sensor monitoring is low CPU, but want to optimize network speed for fast trigger reporting (i.e. earthquake detection)
- http://qcn.stanford.edu
Client Software

- Really two clients –
  - the standalone QCNLive program – for educational initiatives, or local monitoring (can save continuous stream of sensor data)
  - the BOINC-enabled client for live monitoring and reporting to our servers and earthquake detection if numerous triggers from an area are recorded
QCNLive

- Based on the QCN/BOINC screensaver with added features for user intervention
- 4 views – quake globe, 2d/3d sensor, shake cube
- Earthquake information updates hourly (or manually)
- Now uses Qt (formerly wxWidgets), C++, cross platform (Windows, Mac OS X, Linux)
- Can “brand” with your school/museum logo
- Freely available: http://qcn.stanford.edu/downloads
QCN with BOINC

Runs as “non-compute intensive” (nci) so coexists with full CPU jobs e.g. other BOINC projects

Easy access to QCN website & location page

Uses the “trickle” feature in BOINC for reporting potential seismic events (lot of “false positives”)

User can set to run when idle (laptop esp) or run continuously (they have a mounted, external USB sensor)

Attach to http://qcn.stanford.edu/sensor
Drivers for laptops & external devices

- Only two laptop styles supported – Mac laptops, and Lenovo Thinkpads (Windows)
- Manufacturers have been reluctant to allow access to their on-board accelerometer (for various reasons)
- Even external USB manufacturers often provide “kludgy” solutions (simulated COM ports over USB) that require much work to integrate into QCN
Location

3-step Location System:

- Estimate location based on last known router (geoip). Often accurate within several kilometers.

- Participants provide their “favorite five locations” using a Google Maps Interface.

- Linked to IP or set a default location/address.

*Future: a “Where Were You?” website*
Timing Issues

- As well as location, very important for computing quake location
- The QCN/BOINC and QCNLive clients both utilize an external ntpdate program
- Gets an offset between our server and local time and stores locally for adjustments
- Does not interfere with host machine date/time features
- We seem to get $1/100^{th}$ of a second resolution, but need to do more studies
**Triggering Algorithm**

**Significance Level Filter:**
- Compare each point to the standard deviation of the signal in the long-term window prior to the trigger

\[ SL_i = \frac{|y_i|}{\sigma_i} \]

- SL = \(|y_i|/\sigma_i\) provides the confidence level (e.g. SL > 2 gives a 95\% confidence) that an emerging signal is statistically not representative of the prior long-term average (Gaussian)
- Extremely Fast, all CPU in the sensor monitoring loop (50 Hz)
- Computer records 1 minute before and 2 minutes after trigger to disk (SAC file I/O)
- Now sends a follow-up trickle 4 seconds later
>1100 active sensors (3/2011)

- Mac PowerPC 10
- Mac Intel 320
- Thinkpad 159
- JWF8 USB 380
- JWF14 USB 260
- MN USB 6

- Laptop 489
- USB 646 – more USB than laptops!

Thousands of new sensors on order thanks to NSF grant
Roughly 30-50 triggers per day from each laptop
Number of triggers is roughly the same throughout the day (no large day/night difference)
BOINC “trickles” – turnaround time ~3-5 s
Most (99.99%) triggers are “false positives” – we want that .01% though!
QCN Client - Mobile

- Just started Google Android development!
- Relying on OpenGL ES (Embedded Systems) if performance is OK
- Initiate BOINC web service calls when docked
- Most Android devices have accelerometer & GPS
- Anonymously “sign up” via phone # for tracking
- Possibility for external USB sensors?
- Use for “RAMP” (Rapid Aftershock Mobilization Program) to get a lot of low-cost devices in areas
Server Software

- 5 components on 3 servers:
  - Database
  - BOINC scheduler
  - Web pages
  - Data upload (& analysis) server
  - Event detection

- The servers are basic Dell Linux servers i.e. 16GB RAM, 4 CPU, 2 gigabit network if, 1TB disk (cost about US $6K each)
Added tables to BOINC – IP/lat/Ing lookup, geoip, earthquakes, triggers

New mysql – memory tables for new triggers “cached” a few minutes for quake detection

Archiving system – only recent two months of triggers “live” due to size (60 million triggers over three years)
Server – BOINC Scheduler

- Handles BOINC client requests for new jobs (workunits), trickles/triggers, etc.
- Customized for QCN to process trickles immediately (usually go to a trickle table).
- Few thousand lines of extra code for QCN, 4 tables for lookups etc.
- Incoming trickle (trigger) gets matched via IP address to a latitude/longitude, and inserted into trigger table.
- New triggers are put into a memory table (new mysql feature) for fast earthquake event detection.
- ~10 simultaneous events in a region then we have identified an earthquake (still working on detection algorithms & output).
Carried over all BOINC web pages, just customized a few to show triggers etc.

Very important for setting location of your computer (and optional IP address matchup of different locations e.g. laptops).

In near future will show fuller display of detected quakes (currently show sensor location & USGS-detected quakes).
Server – QCN Web Pages
After a “trigger” the software uses BOINC to send a “trickle” message immediately.

This is a small (~300 byte) packet of XML – event time, host time offset, x/y/z sensor reading, significance threshold etc.

If we determine that this machine was involved in a seismic event, we can send a request to upload the SAC (full waveform) files for this event (BOINC “intermediate upload” feature).

Also have a “continual” project that sends full SAC streams via BOINC every 10 minutes (building studies, high-risk areas, sensor testing etc).

Now also send a follow-up trickle after 4 seconds with peak event info.

QCN SAC files are small – 2MB for 3-axes per hours, .5MB compressed.
Server – Data Upload - SAC

9 QCN “stations” in Christchurch NZ
Processes scan the new trigger (mysql memory table) every 0.2 seconds (5Hz)

Simple process – compares USGS-detected/reported events with our sensor network

More complex – detect from our sensor network only

Still a “work in progress” as we fine tune and test the detection algorithms

We seem to get reliable events with as little as 6-8 sensors reporting
Server – Event Detection

March 14th 2011 07:29:28
22 USB Sensors
172.70, -43.59

QCN students/postdocs
Installed ~100 USB sensors
In Christchurch after the
Major quake few months ago
Summary

- QCN different from other volunteer computing projects as it’s low-CPU “distributed sensor”
- QCN client uses BOINC API & features with no modifications – heavily relies on trickle & intermediate uploads
  - Potential mod – mirror trickles to multiple QCN servers (i.e. Taiwan trigger goes to server in Taiwan, and Stanford server)
- QCN has had to modify BOINC on the server, mainly for trickle/trigger processing and “live” detection/reporting
  - Future – QCN servers in different countries, with database (mysql) replication of crucial tables & trickle mirroring to Stanford & UC-Riverside
- Result – a low-cost and reliable system for remote event detection