LIGO

Using Simulated Data Streams to Develop Real-time Analysis Pipelines

John Zweizig
California Institute of Technology
LIGO Lab

For the LIGO Scientific Collaboration and Virgo Collaboration

LIGO-G09xxxxx-v1
LIGO G1300568-v2

GWADW Elba, 22 May 2013

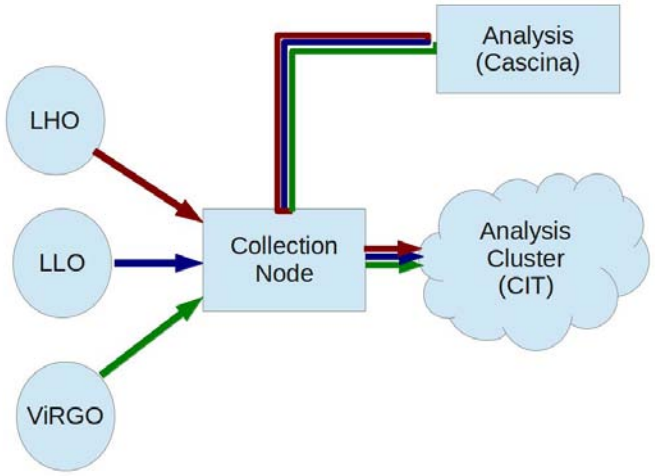
LIGO

Motivation

- Our overriding goal is early realization of full scientific potential of advanced detectors
- This will require:
  - a sensitive instrument
  - Rapid detector characterization
  - well understood analysis procedures
- Near real-time coherent data analysis will be necessary to exploit fully the sensitivity of the advanced detectors.
  - Immediately identify GW candidates
  - Focussed feedback to commissioners
  - Rapid EM follow-up

LIGO

Ideal Real-Time Analysis



- h(t) data available from all detectors (LHO, LLO, VIRGO)
- Route data to large analysis processor cluster
- Low latency O(0.5 – 10s)
- Perform coherent analysis

LIGO

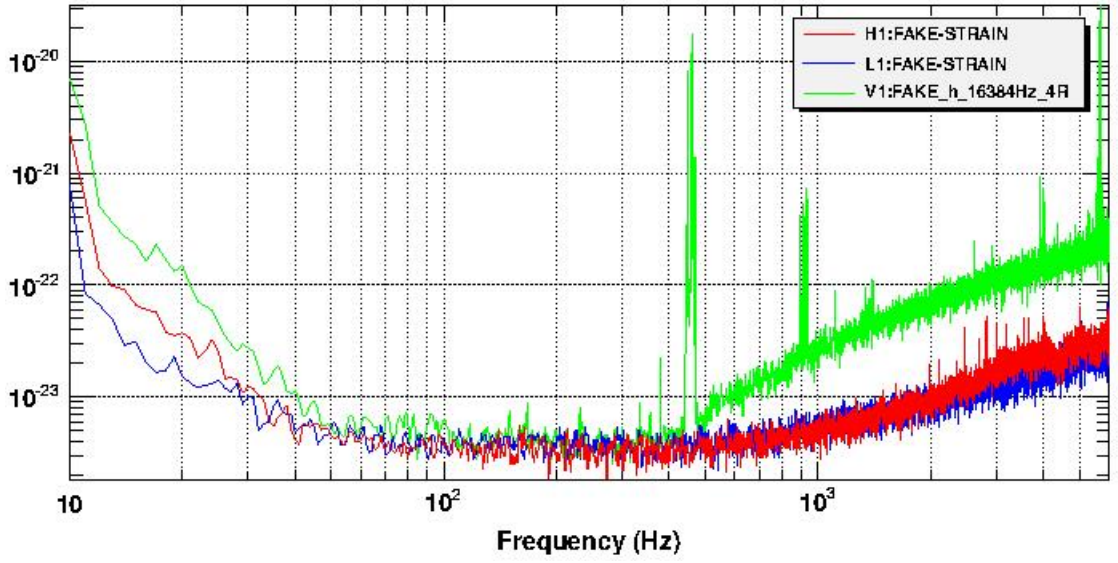
Real-Time Analysis Environment

- Interface to low latency data
- Use real-time data quality info
- Keep up with data stream
- Imperfections in data delivery
  - Dropped frames
  - Invalid data (e.g. NaNs)
  - Unanticipated state transitions (lock-loss)
  - Arrival time slewing
- Problems compounded when performing coherent multi-site analysis.

LIGO

Simulated Data Streams

- If the detector isn't available to provide data – Simulate!
- Noise spectrum generated according to design aLIGO (ZDHP) and aVirgo spectra.
- Inject coherent signal (modified for arrival time, antenna pattern) into each data stream.
- Add generated status channel (lock, calibrated, science bits)
- Package as 4s frames for low-latency delivery.



LIGO

Simulation Plan

- Set up data streams/delivery to be as similar as possible to science running
  - No predetermined science times
  - No synchronization of lock time or science mode between instruments
  - No synchronization of arrival times
  - Realistic data errors / drop-outs.
- Need to hand analysis pipelines simulated data NOW so that they can
  - Learn to interface to low-latency data
  - Implement error recovery
  - Check configuration/timing
  - Verify pipeline efficiency
- Transfer to central analysis clusters
  - Caltech Ligo condor-cluster
  - Cascina analysis node.
- Multiple parallel analysis and DQ pipelines (check interoperability)
- Watch Murphy's law in action!

LIGO

Low Latency Data Transfer

- Build low-latency frames at each observatory.
  - Simulated h(t) + injections, data quality channel
  - Archive frames at observatory, LDR to caltech
- Point-to-point frame transfer to CIT head node (ldas-grid)
  - Simple TCP link from LHO, LLO observatories.
  - FdIO from Virgo.
- UDP multicast from head node to all CIT cluster nodes.
  - Data access Via gds shared memory or /dev/shm
- FdIO service to send LLO, LHO data (+echoed Virgo data) to Virgo.
- Typical latency to CIT cluster nodes (from 4s frame start time):
  - <5s for LHO, LLO data
  - <9s for Virgo Data

Software Engineering Run (ER) Program

- Active 1-month runs twice per year (February, July).
  - Planned milestones (functional goals)
  - Participation of both LSC and VIRGO scientists
  - ~40% of CIT cluster nodes (> half of processor power)
- Low latency data
  - Data steams monitored and actively maintained
  - Both published and blinded injections
- Full implementation of analysis pipelines
  - CBC gStreamer pipeline
  - Coherent waveburst (cWB) pipeline
  - MTBF
  - Others to come?
- Event data services
  - Segment database
  - GW Event candidate database

LIGO

ER1 (Jan 18 - Feb 15, 2012)

- Goals
  - Deploy low-latency data transfer and access tools
  - Develop and test signal generation infrastructure
  - Begin developing science metrics for analysis
- Configuration
  - Noise floor read from Prepared frames (Ninja recolored S5)
- Results
  - O(1%) data losses
  - Latency ~5s (LLO, LHO), ~9s VIRGO
  - Problems when data streams stopped, memory issues

LIGO

ER2 (Jul - Aug 2012)

- Goals:
  - Reduce data loss to <0.1%
  - Closer connection to DAQ at LLO
    - Use ODC channel for DQ
    - Recolored PSL noise.
  - Test analysis event generation
- Configuration:
  - Ninja synthesized noise (recolored s5).
  - Late test using recolored PSL.
- Results:
  - < 0.06% of frames lost in transfer during run

LIGO

ER3 (Jan - Feb 2013)

- Goals
  - Generate noise floor from instrumental data (PSL).
  - Add proxy calibration pipeline (adaptive recoloring filter)
  - Improve ER2 Services
- Configuration:
  - Use recolored PSL PD signal for noise floor
  - Use gstreamer-based process for recoloring as proxy for calibration pipeline.
- Results
  - Used unstable PSL channel after PD failed.
  - Efficient reconstruction of CDC injections > 10Hz

LIGO

ER4 and Later

- Goals for ER4
  - Improved reliability, transparency, control
    - Dashboard
  - Use data from DRMI at LLO if available
  - Improve data recoloring algorithm
- Future improvements
  - Real-time detector characterization
  - Feed more complex data quality information (vetos, segments) to analysis.

LIGO

Summary

- Near real time analysis will be necessary to exploit detector sensitivity fully.
- Real time analysis environment will require specially designed pipelines to deal with online data issues.
- Simulated h(t) data streams are used to develop and test the low-latency data distribution system and the coherent analysis pipelines.
- A software engineering run program is now underway to drive development and testing of the pipeline implementations.