LIGO

Data Acquisition, Diagnostics & Controls (DAQ)

Technical Status

Annual NSF Review of Advanced LIGO Project

April 30 – May 2, 2013

Rolf Bork, CIT

LIGO-G1300419
2013 April 30

LIGO

DAQ Functions

- Provide a global timing and clock distribution system to synchronize all realtime control and data acquisition.
- Provide a common Control and Data System (CDS) infrastructure design and standards for use in all aLIGO subsystem controls.
  - » Real-time applications development tools and code library
    - – Including “hard” real-time operating system, I/O drivers and inter-process communications.
  - » Computer and I/O standards
- Provide all software necessary to synchronously acquire and archive data.
- Provide all computing and networking hardware as necessary to collect data from the various subsystems, format the data and write the data to disk.
- Provide a standard set of diagnostic tools for use in all control subsystems, including ability to:
  - » Inject arbitrary waveforms into realtime control systems
  - » Set and acquire data from defined testpoints on demand
  - » Distribute both diagnostic data and acquired data channel to operator stations
  - » Provide data visualization and analysis tools in support of operations and commissioning.

LIGO

DAQ Functions (Continued)

- Provide computers, I/O hardware and software for the acquisition of Physical Environment Monitoring (PEM) data.
  - » New interfaces for existing PEM sensors
- Computers and infrastructure software for the Diagnostic Monitoring Tools (DMT)
  - » Specific application software provided by LSC members
- Control room computers and associated networking, including a common set of operations support software.
- Provide off-line test and development systems for both sites

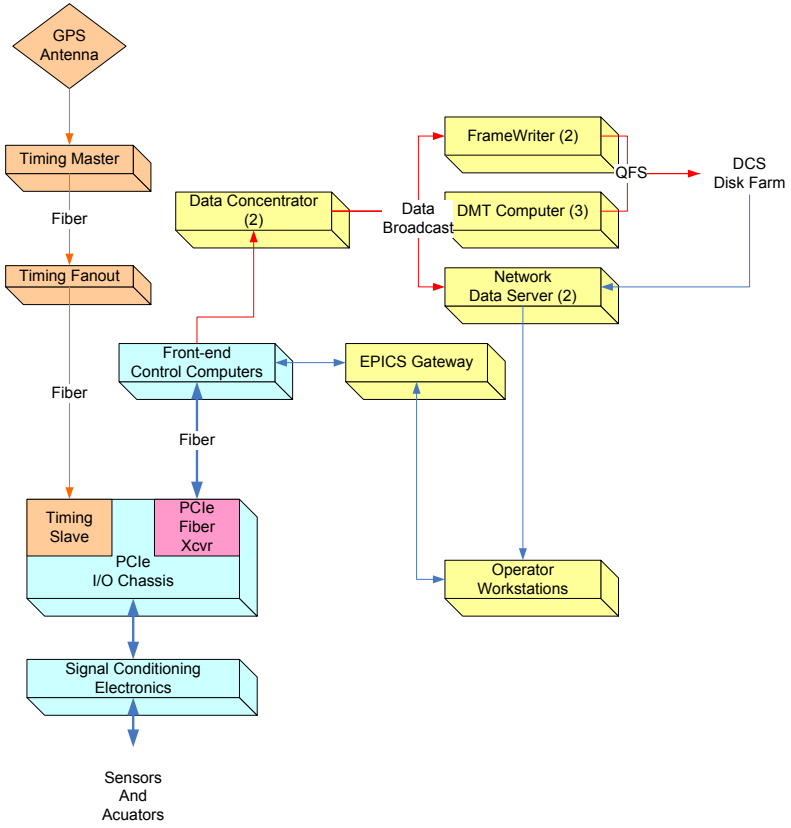
LIGO

DAQ System
Data Acquisition Requirements

- Provide a hardware design and software infrastructure to support real-time servo control applications
  - » Deterministic to within a few µsec.
  - » High performance to support servo loop rates from 2048Hz to 65536Hz
  - » Built-in diagnostic and data acquisition features
- Acquire and record up to 15MBytes/sec continuously from each interferometer.
  - » 'Fast' data channels at rates from 256 to 32768 samples/sec (Up to 3000/IFO)
  - » 'Slow' data channels at up to 16 samples/sec, with up to 70K channels per interferometer
- Provide capabilities to acquire (but not record) an additional 15MB/sec of diagnostic data.
- Write data in LSC/VIRGO standard Frame format to disk system provided by Data and Computing System (DCS).
  - » Provide local disk to allow up to two weeks of data storage
- Provide an internal data distribution system to communicate diagnostic and acquired data to operator stations and Diagnostic Monitoring Tool (DMT) computers.

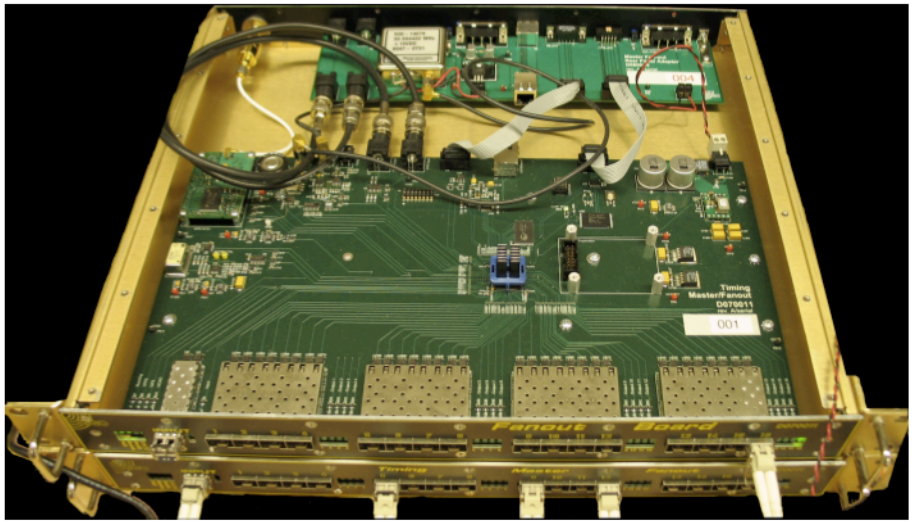
DAQ System Design Overview

- Timing system provides clocks to PCI Express (PCIe) modules in I/O chassis.
- PCIe modules interface to control computer via PCIe fiber link.
- Control computer acquires data and transmits to DAQ data concentrator (DC) via network.
- DC assembles data from all controllers and broadcasts full data blocks every 1/16 second.
- FrameWriter computers format data and write to disk (32sec. data frame)
- Network Data Server (NDS) provides data on demand either live or from disk.

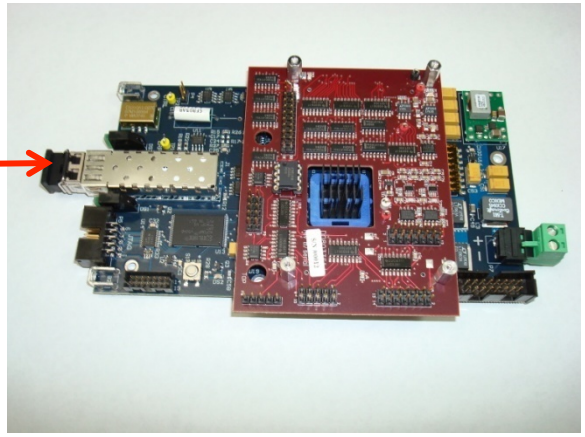


Timing Distribution System (TDS)

- Contracted to Columbia Univ. for manufacture and test after a joint development effort. Design described in the journal *"Classical and Quantum Gravity" under Imre Bartos et al., 2010 Class. Quantum Grav. Vol. 27, No. 8, 084025*



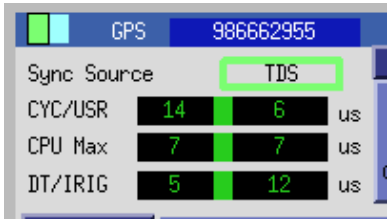
IRIG-B Timing Fanout
Provides accurate time information to computers.



Timing Slave provides accurate clocks At 65536Hz to ADC/DAC modules.

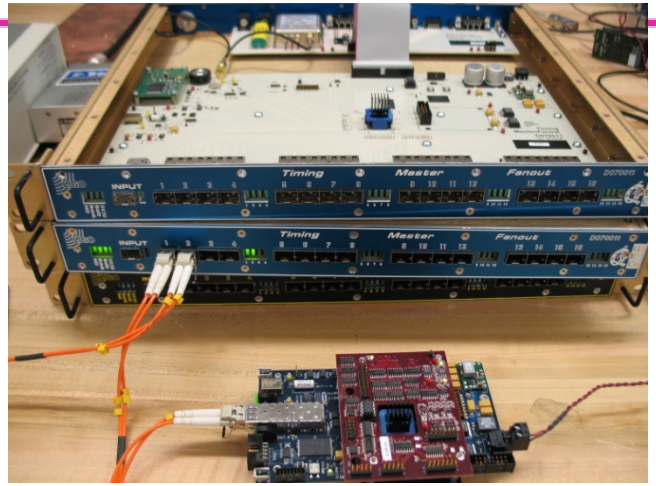
TDS
IRIG-B Distribution Unit

- IRIG-B system used to provide time information, in GPS seconds, to DAQ and control computers.
  - » Includes standard timing slave card to get time information from TDS.
  - » Outputs IRIG-B standard time code
    - – DC Level Shift format
  - » Commercial IRIG-B Receiver modules in computers for accurately setting time in GPS seconds.
  - » Time accuracy to better +/- 1 µsec.
  - » Second source of system time verification, along with duotone signal acquired from timing slave in I/O chassis.

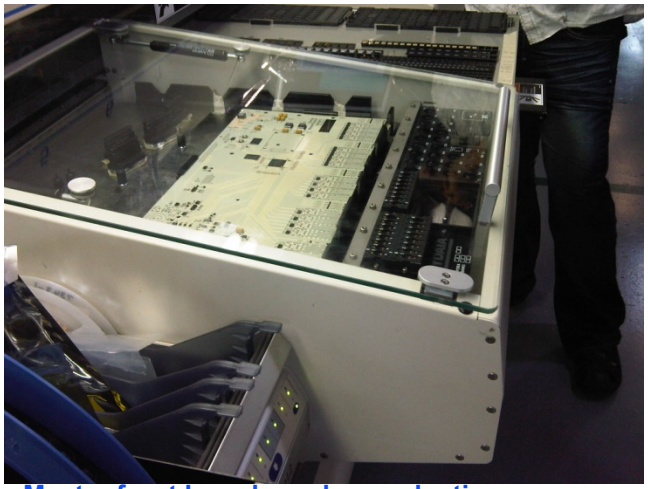


Timing Distribution System Status

- All components have been tested and delivered.
- Equipment installed and operational at both sites.



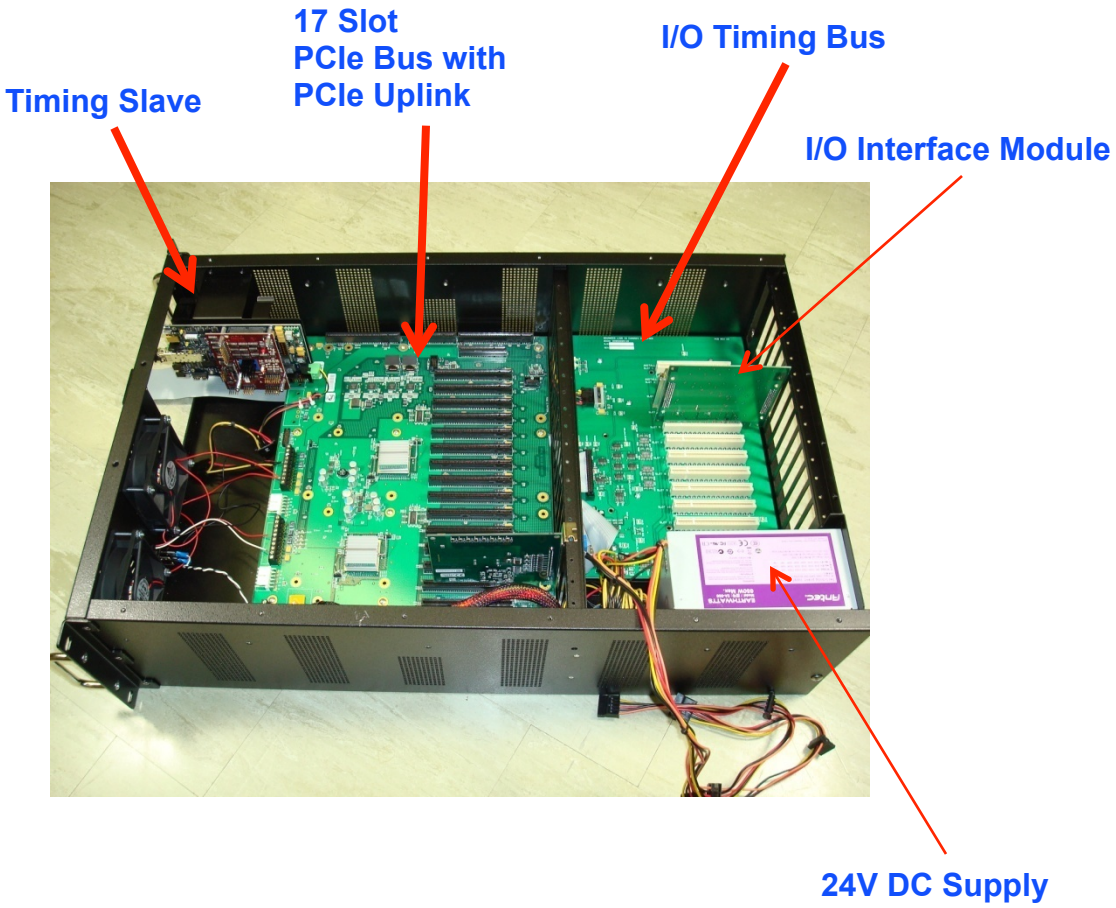
Slave-DuoTone pair being tested at Columbia



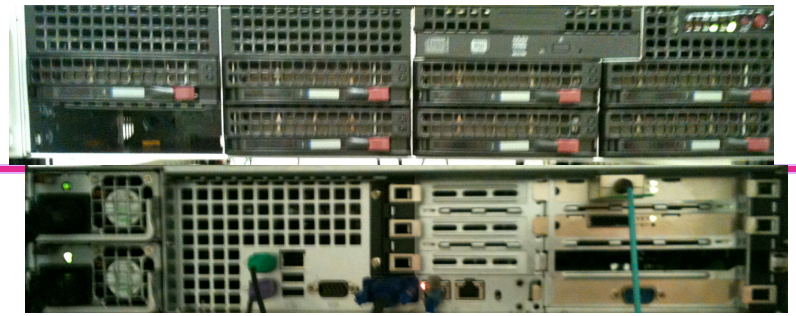
Master front boards under production

CDS Standard PCI Express I/O Chassis

- Commercial PCIe expansion motherboards.
- Custom I/O timing and interface backplane.
- I/O interface modules provide timing and interface between PCIe module connectors and field cabling.
- Two fiber optic links.
  - To timing distribution system via timing slave module.
  - To computer, via fiber optic PCIe link.



CDS Standard Computers



- Supermicro X8DTU-F Motherboards
  - » Fulfills BIOS PCI-e card mapping and real-time stability requirements
- Single Xeon X5680 processor with six cores at 3.33GHz
- Up to 4 full height + 1 half-height PCIe slots
- Two GigE Ethernet ports
  - » Separate EPICS/DAQ networks
- No disk drives installed in computers used for real-time control
  - » Operated as diskless-node from central boot server
- Operating Systems
  - » Gentoo with Linux kernel 2.16.34, plus LIGO RT patch
  - » Ubuntu Linux for CDS servers and other non-real-time computers

LIGO

Networking

- Ethernet backbones for most applications
  - » GigE switches with fiber uplinks from end stations
  - » GigE switches with 10G uplink options for corner station
    - – 10G uplink for DAQ and video connections
  - » 10G switches for DAQ Broadcasts
- Low latency networks for real-time data communications.
  - » Initial LIGO type reflected memory (for long runs to end stations)
  - » PCIe network, employing reflected memory software (corner station computers)

PCI Express (PCIe) Real-time Control Network

- Low Latency (1.25usec)
- High speed (10Gbit/sec)
- Cable or Fiber connections
  - CX-4 cable to 3 meters
  - Multi-core fiber to 100 meters
- Stackable 10 port Switches
- Reflected Memory Mode
  - Data broadcast to same memory location on each computer on the network.

Corner to End Station Real-time Control Network

- Loop topology
- Low Latency (700nsec/node)
- High speed (2Gbit/sec)
- Fiber connections
  - Up to 10km
- Bypass Switch provided at each location
- Reflected Memory
  - Data broadcast to same memory location on each computer on the network.

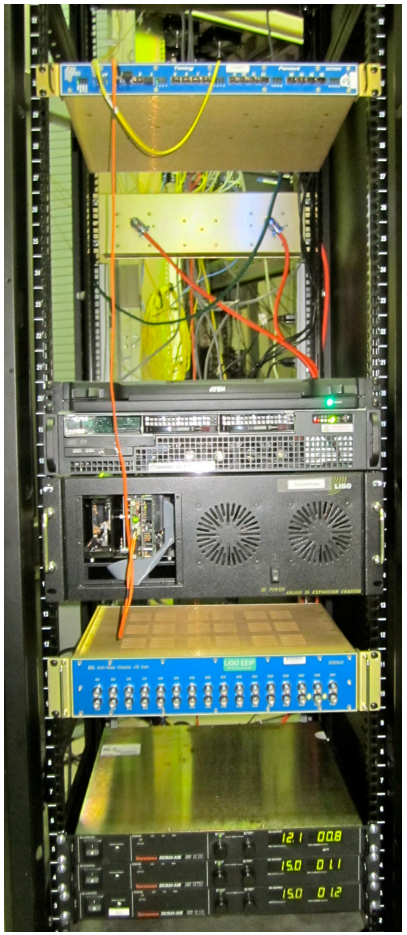
LIGO

Networking – Progress

- All networking equipment has been delivered and installed.
- Finalizing “as built” installation drawings.

Physical Environment Monitoring Infrastructure

- For aLIGO, PEM system will provide control as well as DAQ
  - » On-line Adaptive Filtering and feed-forward control.
- One computer + 1 I/O chassis at each station and at corner station.
- Re-use existing PEM sensors
- Up to 128 channels of ADC + 8 channels of DAC
  - » I/O connections via AA/AI chassis with BNC connections.
- Progress
  - » Computers, I/O chassis and ADC/DAC modules have all been procured and delivered.
  - » Systems installed and operational at both sites.



DAQ
Computing / Storage Equipment
(All Delivered and Installed)

- Data Concentrator (DC) (2)
  - » Collects data from all real-time control computers and broadcasts to 10GigE network.
  - » One unit on-line, second hot backup
- FrameWriter (2)
  - » Receive data from DC
  - » Format data into LVC standard Frame format
  - » Write data to disk
    - – Local
    - – Data Analysis group disk farm
- Network Data Server (NDS) (2)
  - » Provides real-time or stored data on request to various control room software tools
    - – NDS clients also developed for Perl, Python and Matlab
- Two computers running Solaris operating system to connect disk systems via QFS.
- 24 TByte Local Disk

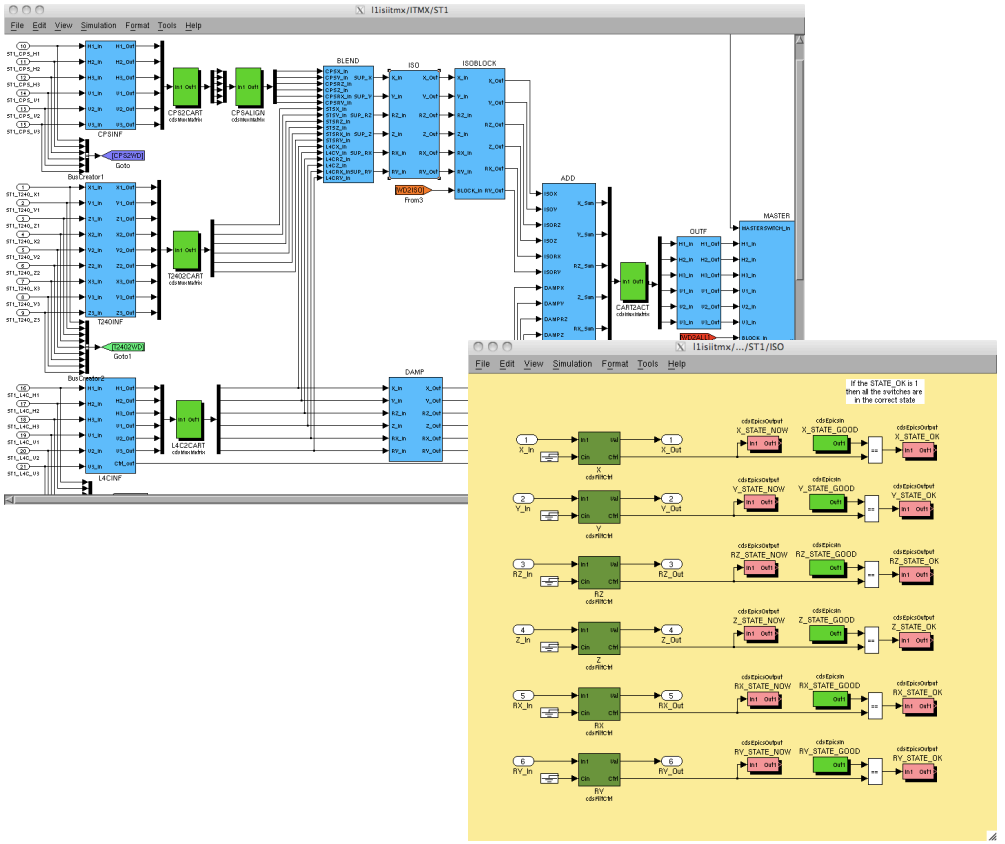
LIGO

Control Room and Global Diagnostic Systems

- iMac computers w/additional monitor chosen as the standard configuration for operator stations.
  - » Ubuntu Linux Operating System
- Two, dual CPU computers, similar to real-time control computers, in place for Global Diagnostic Monitoring Tool (DMT) applications.
  - » 24TByte disk drive provided for storage of DMT information.
- All equipment is installed and operational.

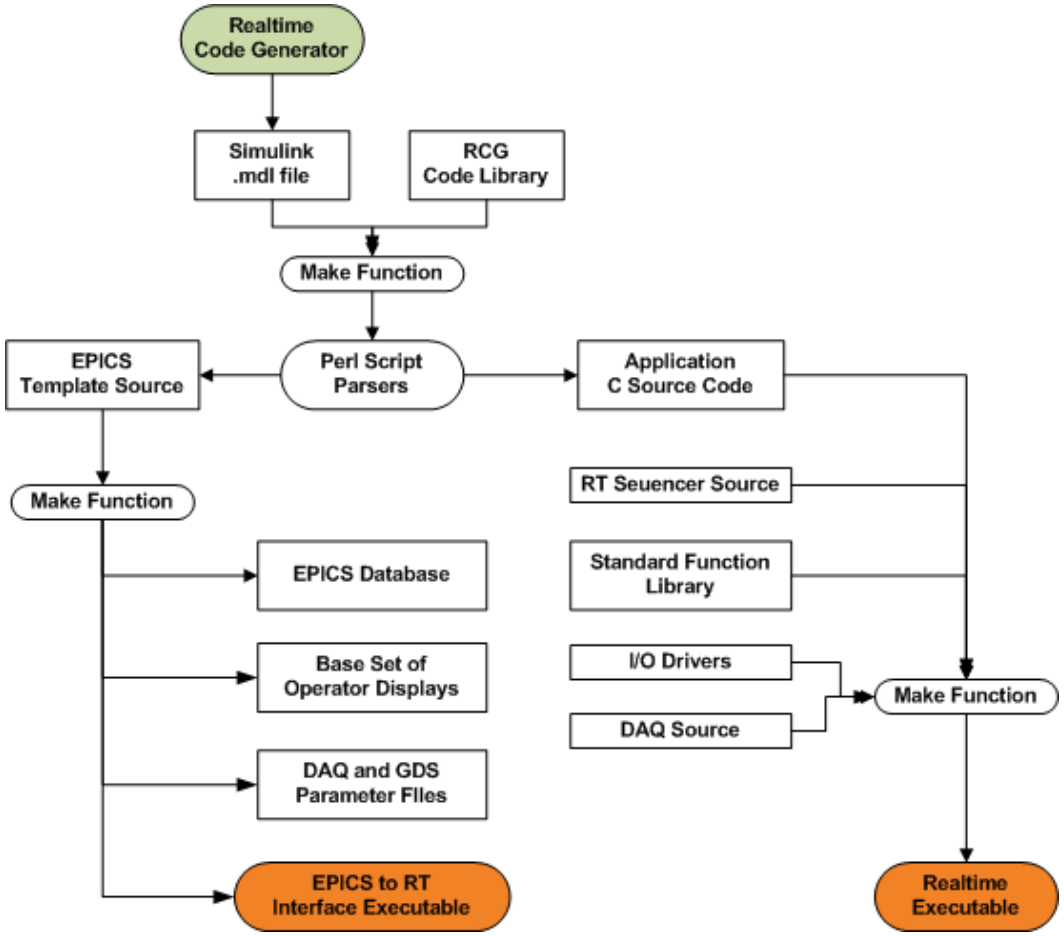
Software
Real-time Application Support

- Continued refinement of graphical tool for real-time code generation (“RCG”).
- Allows control application development and documentation without having to know a programming language.
- Allows programming staff to concentrate on development and test of common code modules.



Software
Real-time Application Build Process

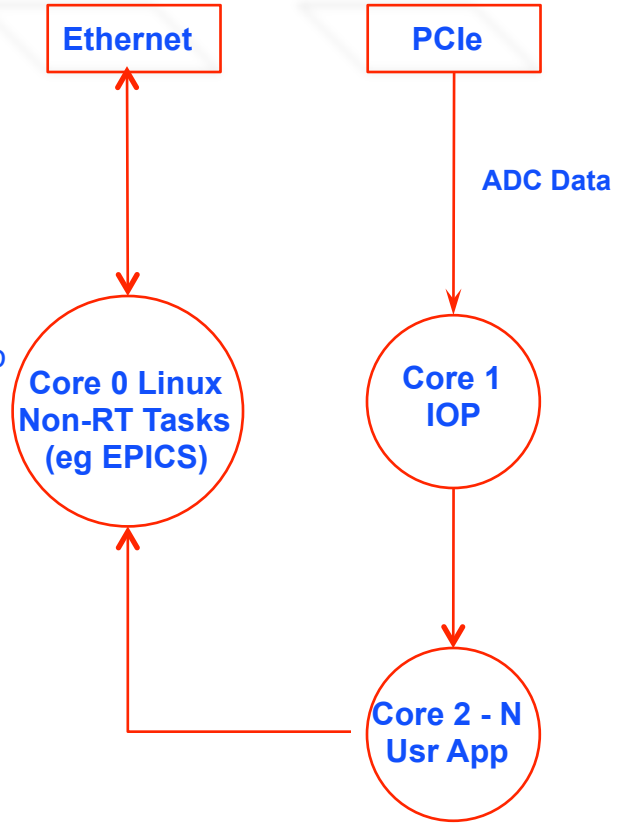
- Build and save RCG model.
- make 'modelName'
  - Perl scripts parse the model file to determine signal connections and code flow
  - Perl scripts generate EPICS and real-time source code.
  - Compiler is invoked to link common code libraries and produce real-time and EPICS executable software.
- make install
  - Moves executables to target directories for load onto real-time computers.
  - Channel descriptor files generated for use by DAQ and GDS
  - Basic set of operator displays generated.



LIGO

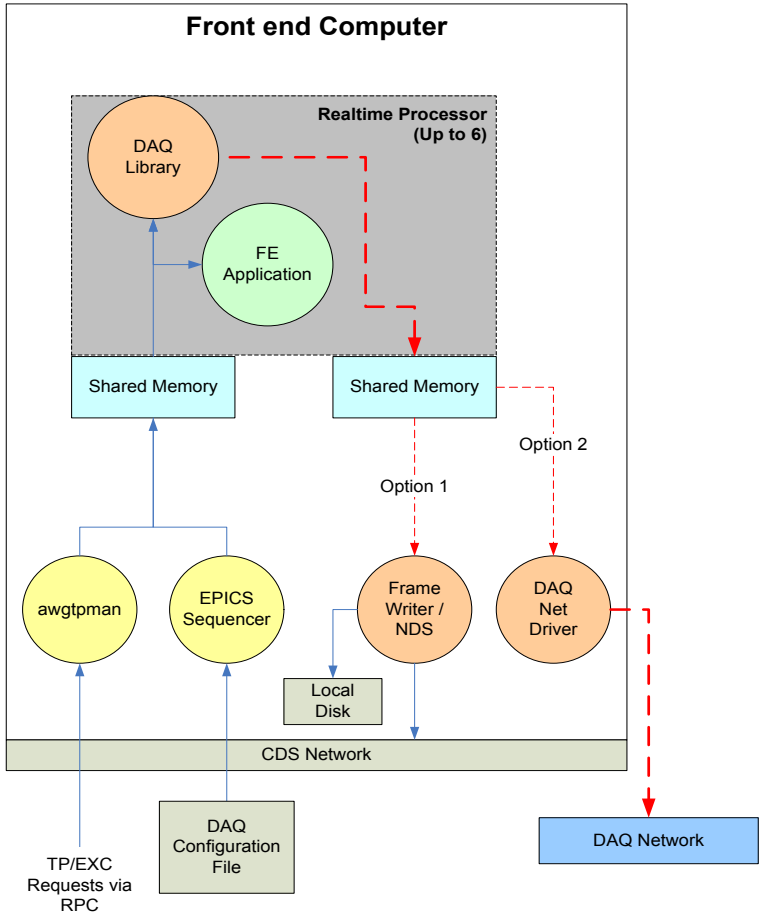
Real-time Core and Patch

- aLIGO Real-Time (RT) code not “traditional”
  - » No pre-emptive operating system scheduler
  - » No interrupts, semaphores, priorities, ensuing context switching, etc.
- Each RT app locked to its own CPU core
  - » Using custom patch to Linux kernel “play dead” routine
    - – Notifies Linux scheduler that CPU is going down and unavailable for interrupts/task assignment.
    - – Inserts RT app code instead of Linux idle routine.
    - – Removal of RT app brings the CPU “back to life” and reconnects to Linux as a useable resource.
  - » RT code runs in continuous loop
    - – Triggered by arrival of ADC data in local memory (polling or MONITOR/MWAIT CPU instructions)
      - ADC modules set up to automatically transfer data to computer memory on clock trigger
    - – Never switched out ie always resident on stack, in cache, memory
- For each RT computer, there is a special case model called an Input/Output Processor (IOP)
  - » Controls startup timing and synchronization.
  - » Maps and initializes all of the PCIe I/O interfaces
  - » Triggers and monitors user applications.
  - » Always running, allowing user apps to come and go, as necessary



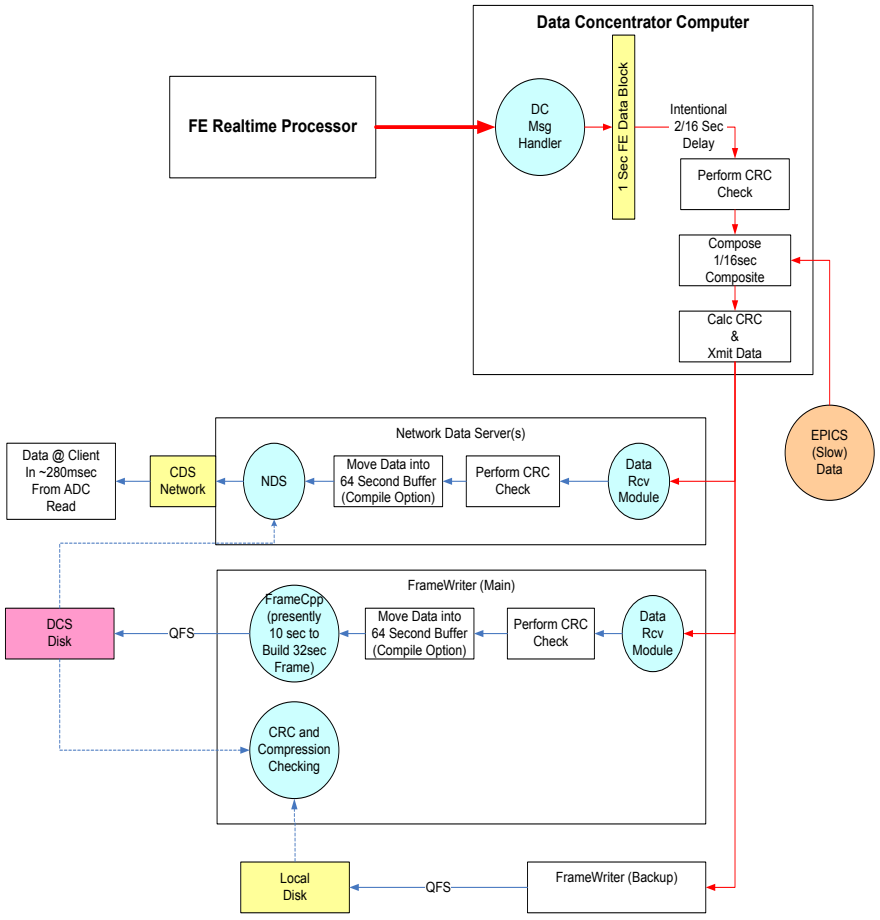
DAQ System
Front-End Software Design

- A common DAQ library is compiled into each FE application.
- Acquires data at user defined rates and transmits data as 1/16sec data blocks:
  - » For archive, as described in a DAQ channel configuration file.
  - » Test point and excitation channel data on demand
    - – As requested via the arbitrary waveform generator/ test point manager (awgtpman)
  - » Supports aggregate (DAQ+TP) data rate of 2MB/sec per FE processor
  - » CRC checksums and timestamps sent with all data blocks
- Supports various configurations
  - » (1) Data to FrameWriter/NDS software on same computer via shared memory
    - – Allows a complete stand-alone system to support various subsystem test stands
  - » (2) Data to shared memory, with separate network software
    - – Supports multiple FE applications on same computer
    - – Relieves RT front end code from network error handling and other possible delays



DAQ System
Backend Software Design

- Data Concentrator
  - » Collects ‘fast’ data from all FE computers via dedicated network
  - » Collects ‘slow’ (EPICS) data via CDS network
  - » Broadcasts combined data to upstream computers as 1/16 sec data blocks on to 10Gb Ethernet
- FrameWriter
  - » Format data into standard LIGO Frame using FrameCpp library, with data compression.
  - » Write data, via QFS, to DCS disk farm (32 second data file)
- Network Data Server (NDS)
  - » Provides live and archived data feeds, on request, to CDS operator stations



LIGO

Guardian

- Software tool set for implementation of control automation processes.
- Provides:
  - » Development Tools
    - – Scripting tools, with common API, to define states and state transitions.
    - – Methods to build a hierarchy of automation procedures.
  - » Runtime Tools
    - – Common operator graphical user interfaces.
    - – State monitoring and verification processes, with error reporting features.
    - – Ability to load state definition files and launch state transition scripts.

LIGO

Guardian Status

- Recent review meeting held to verify requirements and review present design (LLO April 24-25, 2013)
  - » Lead person identified to oversee the Guardian development/application process.
  - » While present software meets primary requirements of timing and synchronization, some additional requirements were identified.
  - » Guardian toolset developers to verify existing tools and provide software to meet additional requirements.
  - » Subsystem application developers to:
    - – Further define operational states and transitions.
    - – Migrate existing and add new automation scripts into the Guardian structure.

Software Development Process and QA (1)

- Basic review process and code style guidelines for initial code development provided in LIGO-T970004A.
- Additional documentation on software development process provided as code development moved into upgrade and maintenance phase, as outlined in T1300427.
- All software controlled under CDS SVN (LIGO-T0900531)
  - » Moved from previous CVS system.
- Bug reporting and new feature requests via Bugzilla (T1000496)
  - » Formal tracking and review for code release to use LIGO Engineering Change Request (ECR) procedures.

Software Development Process and QA (2)

- Code Requirement and Design reviews
  - » Weekly software meetings, which include LIGO subsystem leads and other end users.
  - » Mailing list (cds_announce) to disseminate information and get feedback from a larger user community.
  - » Periodic face-to-face meetings, usually 2-3 days, with developers and end users to discuss focus topics.
    - – Latest held at LLO April 24-25, 2013 to review automation tools.
  - » Formal external reviews
    - – Latest held September, 2012 at Caltech.
    - – Down to the level of line-by-line review of key components.

Software Development Process and QA (3)

- Code development and test
  - » Second person assigned to review and test developer's code.
  - » Testing done per CDS Test Plan (T1000561)
    - – Automated test scripts have been, and continue to be, defined to perform nightly testing on the latest versions of software prior to release.
- Code documentation
  - » Code commentary written to use doxygen documentation generation tools.
  - » Documentation set part of nightly code build.
- Code Release
  - » Procedure provided in LIGO-T1100240

Software Failure Analysis and Test (1)

- CDS software not used in personnel safety systems.
- Equipment safety provided by hardware systems.
- Standard set of software built into every real-time control application to detect critical errors and take appropriate action.
  - » Standard diagnostics and actions listed in LIGO-T1100625
  - » On critical fault detection, basic sequence is:
    - – Take system to safe state by setting all controller outputs to zero (0V output from DAC modules)
    - – Report errors via EPICS channels for enunciation via alarm handlers and Guardian tools.
    - – Log errors to provide further diagnostic information.
    - – Exit from the real-time control process, if the software cannot, or should not, take further corrective action.

LIGO

Software Failure Analysis and Test (2)

- Standard watchdog code modules developed for use in individual control applications.
  - » Purpose is to allow software to detect errors before tripping hardware safety systems.
  - » Examples:
    - – DacKill part to force DAC outputs to zero. Actual error detection provided by separate input logic specific to a control application.
    - – Suspension watchdog for optics control monitoring.
- Testing
  - » Necessary hardware provided on LHO and Caltech off-line DAQ test system to run failure mode testing.
  - » Automated testing developed to run nightly using Jenkins tool.
    - – Latest code checkout from SVN repository.
    - – Control application code compiled, installed and restarted.
    - – Test software invoked.
    - – Test report generated, with doxygen format.
    - – Test pass/fail status recorded by Jenkins, along with detailed test report.

LIGO

Software Status

- "Final" code version tested and released.
  - » Any new code change requests / bug fixes are to be part of commissioning and operations activities.
- Software review, with external reviewers, held in September, 2012. Review findings contained in LIGO-M1200346. Primary recommendations are being addressed:
  - » Hierarchy of automated testing.
    - – Installed Jenkins continuous integration tool on test systems.
    - – Used to perform nightly SVN code checkouts and builds and initiate test scripts.
    - – Various test scripts/code have been, continue to be, developed to support various levels of software testing.
  - » Refactoring of large code blocks into more maintainable and well documented components.
    - – In progress ~80% complete.
  - » Additional code documentation and use of the doxygen tool
    - – About 75% of source code has been updated to use doxygen style commentary.
    - – A 'make doc' feature has been added to the RCG to produce on-line documentation using the doxygen tools. On software test systems, this is part of the nightly build process.

LIGO

DAQ System Acceptance Review Preparations

- Continuing to update DAQ document tree in DCC
  - » Top level is LIGO-E1200645
- Requirements/design documentation
  - » Performing final checks and updating, as necessary.
- Installation Documentation
  - » Completing “as built” drawing sets (90% complete)
- Software Development and Test Plans
  - » Recently updated and ready for review.
- Software Test Procedures and Test Data
  - » In process of automating test procedures and report generation (40% complete)

DAQ System Acceptance Review Preparations

- Internal Code Documentation
  - » 75% complete in moving code commentary to doxygen format for automated manual generation.
- User Guides
  - » Recently updated and being reviewed.
- System Diagnostics and Troubleshooting
  - » Ready for review.

NSF Review 2013 Concerns

- Concern:
  - The Project should implement procedures and controls to ensure that only realtime control software that has been tested on Caltech/MIT prototypes or another appropriate test stand can be uploaded for use in the control systems of critical components. The project should also take steps to ensure that the appropriate test stands can remain available for this purpose in the future.
- Action Taken:
  - Caltech/MIT and site test systems have been updated, to the extent possible, to use the latest aLIGO hardware and continue to be available for CDS core and user software testing.
  - CDS core software is now under aLIGO Engineering Change Request (ECR) control and review. Only approved and tested changes are allowed in code releases, and only these releases are allowed to run on the interferometers.
  - All subsystem control applications are under SVN control and, to the extent possible, tested offline. As many of these applications are becoming mature, an ECR will also be required for future updates.

LIGO

DAQ System Summary

- Software Development
  - » Code reviewed and action items being addressed.
  - » Documentation being updated for acceptance review.
- Equipment Procurement
  - » Complete
- Installation
  - » Complete
  - » “As built” installation drawings being completed for acceptance review.
- Storage of equipment for 3rd interferometer
  - » Preparing procurement documentation