Post-O2 software plans

Jameson Graef Rollins

Commissioning F2F
LLO
February 22, 2017

Short term (to be ready post O2)

- Long range PCIe network to replace RFM (ECR pending)
  - Allows more corner↔end IPC channels
  - Faster real-time read/write
  - Allows all Dolphin (PCIe) equipped computers to communicate directly regardless of location.
  - Include improved IPC error monitoring (latching) (bugzilla #793)
- Improved global settings monitoring (see below)

Longer term: features

- Control room NDS2 proxies. Allow seamless retrieval of data (including test points) for all times (including “online”) via a single NDS2 server.
  - proxy NDS1 servers with NDS2 interface
    → **eliminate need for NDS1 client support**
  - transparently handle data retrieval from LDAS NDS2 servers
  - automatic server fail-over and load balancing
- RCG: allow BUS connections between model sub-block levels
- DAQ: support string types in frames
- state space parts

Longer term: features

- RCG "simulation" modes → no required hardware
  - Allow IOP models and DAQ to function without ADCs
  - Direct DAQ→ADC connections in IOP
  - Allow compiling models into user space processes

Longer term: core maintenance

- Distribution of all CDS software through package management for supported “reference operating systems” (Debian and Scientific Linux)
- NDS2 server re-architecture for improved scalability and robustness
- DAQ source/build reorganization/modernization
- RCG
  - support for newer Linux kernels
  - rework parser (more extensible, python-based)
  - support new Matlab file format (.slx)
  - investigate alternate design GUIs

Longer term: core maintenance

- Replace aging software:

  `dataviewer` need NDS2 support

  `MEDM` end of life, no longer supported

  `MX/OpenMX` (RCG→DAQ network) end of life, no longer supported

  `EPICS` upgrade to 3.15 (4.0?)

Global settings monitoring

- Track all settings in real time and allow comparing current settings against arbitrary references:
  - snapshot files
  - times from past
  - previous instance of Guardian state
- Unified monitoring of all settings globally, regardless of IOC source, and not exposing arbitrary front-end partitioning.
- Allow viewing differences for any subset of settings channels.
- Allow creation of new references, either as snapshot files or as tagged times.
- Allow different users to compare against different references at the same time.
- Allow restoring (arbitrary subset) of settings to those in reference.

Global settings monitoring

- During Observing: provide continuous real-time feedback of settings differences compared to a static reference, and expose those differences via e.g. EPICS so that they may be recorded and integrated into “Observation Ready” status.

- During commissioning: provide capability to compare current settings against those from the last time the system was in the same Guardian state.
    - Primary behavior: when ISC_LOCK guardian comes to rest, look at current state and find last instance of same state. Retrieve settings from that time and compare.
    - Work for other guardian nodes and setting subsets (e.g. SEI_ITMX guardian state and L1:ISI-ITMX_* channels).

Types of settings

Types of settings:

static not changed by automation

automated change by automation (e.g. changed during lock acquisition then reset to initial/"safe" value at lockloss).

secular change in a secular way (e.g. alignment offsets)

actuated actively actuated via "feedback" of some sort

System should facilitate categorizing all settings appropriately and should allow filtering by type.

Are there other types?

Conlog?

Is conlog needed?

All current conlog functionality can be duplicated with combinations of NDS calls (prototype implementation exists). Question is relative performance.

- NDS-only implementation can compare ~ 20k channels in ~ 1.5 min (on site)

If NDS-only implementation performance is within $\sim 2\times$ conlog performance, is conlog worth the effort/cost?

Still need efficient way to find previous Guardian states.

- NDS search of this type is slow but not necessarily significantly slower than conlog would be.
- Needed for “lockloss” tools as well.

More testing required...

Noise budget web

https://ldas-jobs.ligo.caltech.edu/~noisebudget/

- Common web space for noise budget data from both sites, with interactive plots
- Provided scripts will format site NB data into common data format (HDF5) and upload.
  - HDF5 is a structured data format that is universal supported and trivial to import in any language/platform.
- Pages and plots are auto-generated.