Paired Redundant IOCs with Redundant Hardware

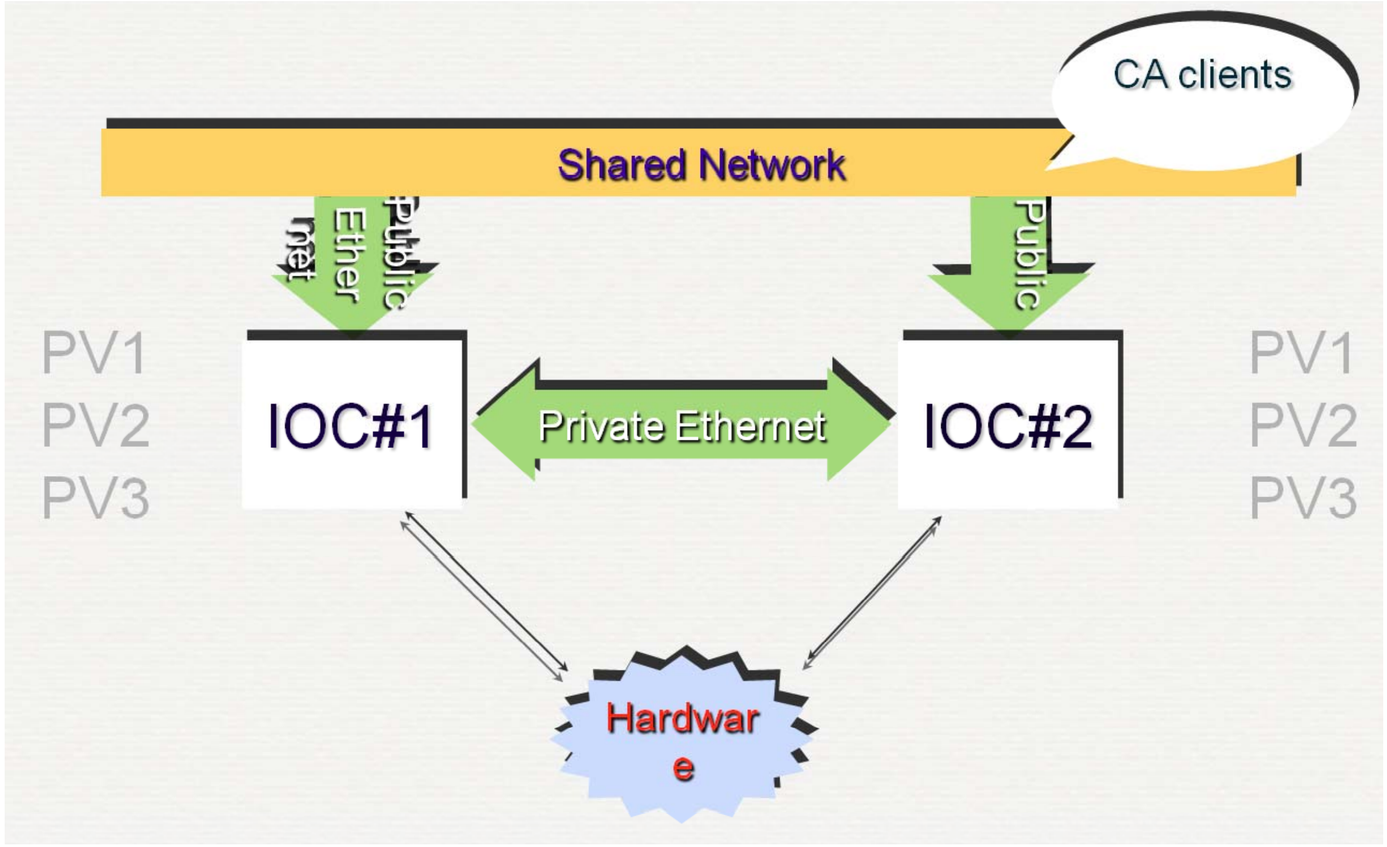
S. A. Baily and Eric Bjorklund

Why Does LANCSE Use Redundant IOCs For Its Timing System?

- Since the 1990s we've had redundant hardware for the master timer system.
- If the master timer stops putting out timing gates, all of our RF stands trip off.
  - It's not difficult to recover a single stand, but recovering all stands takes a lot of time.
  - Historically, dropping all RF power abruptly at 120 Hz caused city-wide power outages.
  - Power dispatch calls when our electricity usage drops quickly (at lower repetition rates).
- Without redundancy, maintenance opportunities on the timing system would be very limited.

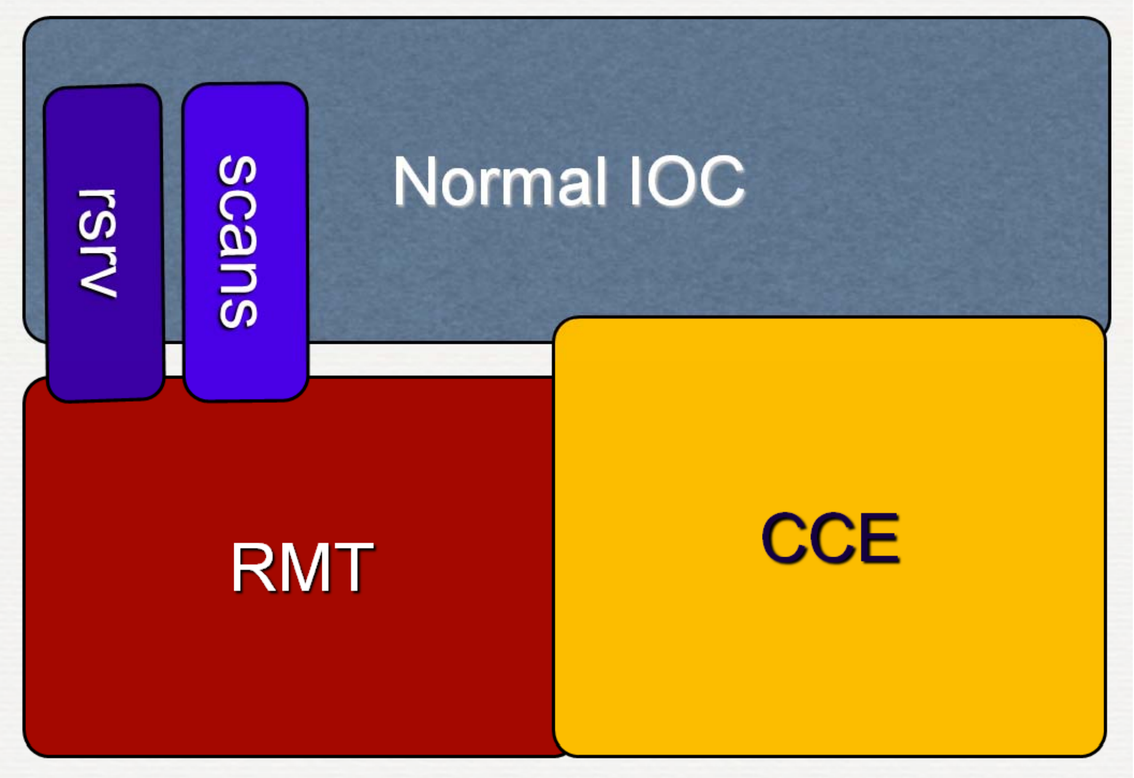
Redundant IOC software

- Developed by DESY in collaboration with SLAC
- Maintained by DESY.



How Redundant IOC Software Works

- Redundancy Monitoring Task
  - Monitors state health of network and drivers
- Continuous Control Executive
  - Synchronizes EPICS databases
- RMT Driver
  - Plugs in to RMT.
  - Reports state of health
  - Reports synchronization status
  - Accepts commands to enter master or slave states.

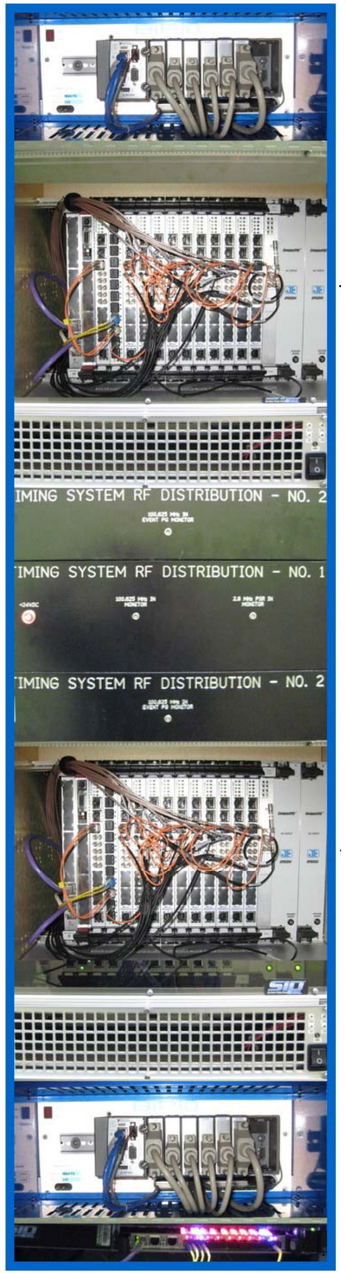


Why a Redundant IOC Pair?

- Master Timer
  - Timing Pattern Generator (TPG)
    - VME-64 IOC with solution and zero crossing detector.
    - Generates scheduled timing patterns and timing event link.
    - Includes RF gates and triggerable beam sequences.
      - MRF event generators.
  - Gate Enable, Inhibit, and Countdown Controller (GEICCO)
    - cRIO IOC determines which beam events get sent.
    - Triggers actual beam gates (slave event generators in the TPG) in response to operator switches, and counters.
      - cRIO with TTL Binary I/O.
- A second pair provides redundancy
- A fiber optic switch (MRF-fanout concentrator) selects which system is used.

Hardware Setup

- Two IOC pairs.
- Two single points of failure.
  - RF input (may not matter)
  - Event link (unavoidable)
- Systems monitor each other as part of synchronization.



Geicco-1

TPG-1

RF Dist.
RF Input
RF Dist.

TPG-2

GEICCO-2

Event-link switch

LANSCE Timing System IOC Functions

- Timing Pattern Generator (TPG)
  - Generates scheduled timing patterns.
  - Includes RF gates and triggerable beam sequences.
    - MRF event generators (VME-64).
- Gate Enable, Inhibit, and Countdown Controller (GEICCO)
  - Triggers actual beam gates (slave event generators in the TPG) in response to operator switches, and counters.
    - cRIO with TTL Binary I/O.

Differs From Traditional Redundant IOC

- Not two identical IOCs on the same bus.
- Has redundant hardware.
- IOCs should run continuously.
- IOCs should fail over in pairs.

Broadly Applicable Improvements

- Added support for syncing fields with SPC_DBADDR (waveform records)
  - Modified the Continous Control Executive (CCEXEC) to support larger fields.
- Made it possible to specify which records should be synced
  - Configurable via info nodes.
  - Modified e2db so that CapFast can be used to configure info nodes.

Why Not Sync All Records?

- IOC stats should be individual.
- Conserve bandwidth
- Redundant hardware read-back channels should come from the actual hardware.

Redundant Hardware Specific Improvements

- Added an option to pause only the channel access server instead of the entire IOC.
- Call post_event on the slave when the database becomes synced with the master (CCEXEC_SLV_INSYNC state) so that passive records can be made to process on the slave.
  - Uses event-driven fanout records.
  - Can control the processing order.
  - Particularly useful for commands.

Small “surprises”

- Database changes should be made live (as well as to the file).
  - If only the file is changed, a rebooted IOC will just resync with its master.
- Needed individual zero crossing delay record values to make timing match as close as possible.
- Failover cannot be made fast enough for our needs.
  - There’s a delay of approximately 1 second in addition to all the specified parameters.

Managing a pair of IOCs

- One of the health monitoring tasks monitors the paired IOC.
- Choose the IOC from which *command-to-fail* commands should be issued. The other monitors the switch position.
- Critical hardware fault failovers are handled by the program in the cRIO.
  - The redundant IOC software then follows.

RMT Drivers

- TPG
  - None, our software tasks
    - Monitors and reports its own health to GEICCO
    - Monitors GEICCO status/health (using asyn)
    - Monitors switch position (using asyn)
    - Issues *command-to-fail* to follow the switch position
- GEICCO
  - Monitors LabVIEW status/health
    - Our program sets the switch position when the IOC becomes master
  - Monitors TPG status/health

Results

- Redundant IOC software can be adapted to coordinate multiple IOCs.
  - Requires some custom software tasks to provide faster failover.
  - Our system can detect a dropped 120 Hz cycle, and respond in time for the next cycle.
- New synchronization control feature (*info nodes*)
- New *pause only channel access* feature
- New *post_event* feature
- New support for SPC_DBR field and large fields