TCS training

(focused on troubleshooting)

LBTO SW Group Tech Talks #2

TCS philosophy

- TCS controls everything except instruments
- Instruments command TCS to perform observations
  - IIF, DDS - ICE public interface
- Only “Authorized instrument” can issues commands
  - Unauthorized instruments can read TCS variables (IIF GUI)

- Separate hardware control from GUIs
  - GUIs uses shared memory to get HW data
- TCSGUIs mostly for display..
  - And change few parameters
  - And do a few critical commands (initiate enclosure opening/closing,...)

TCS ingredients

- Reflective memory
- RPC calls
- MCSPU
- DSP for axis
- UMAC motion controllers for hexapods on OSS
- VxWorks (PMC)
- TPLC (Telescope PLC, powers axis)
- ECS PLC (Enclosure Control System PLC)
- HSB (HSB PLC)

Reflective memory

- IPC shared memory on nodes
- gshmserver allocates sharede memory
- gshmserver sends UDP packets with data changes
- Shared memory allocated into blocks by TCS components, ddeditor to see its structure
- tcslib keeps track of changed (dirty) blocks
- Dirty blocks are propagated to cluster gshmserver’s

RPC calls

- Homebrew TCS solution
- Uses TCP/IP as transport mechanism
- Opens new TCP/IP socket for every call, runs inside thread on server process
- Intra TCS - XML
  - Homebrew XML writer and parser
  - core/ArgumentList.cpp
  - commandreturn/CommandReturn.cpp

RPC calls - MCSPU

- Serializes as std::string, including binary data:
  - 989 // AZIMUTH
  - 990 azPolynomial.t0 = azTimestamp * SECINDAY;
  - 991 azPolynomial.a0 = azPMount;
  - 992 azPolynomial.a1 = azVMount;
  - 993 azPolynomial.a2 = azAMount;
  - 994 azPolynomial.seqNum = azElRotSeqNum;
  - .....
  - 1000 returnValue = azTrack(string((char*)&azPolynomial,sizeof(azPolynomial)));
  - 1001 azKey = returnValue.substr (0, 1);
  - 1002 azFound = mcspuErrorCodes_.find (azKey);

MCSPU

- Mount Control System, Processing Unit
  - Does not know about RA/DEC coordinates
  - Knows about axis Alt/Az
  - Does not know telescope modelling, .. -> all that is inside PCS
- Runs on jet computers
- mcstemp
  - Backend, holds connections to hardware
- mcsDisplay
  - Ncurses engineering display

TCS Polynomial

Reference to 4 element vector, describing:

- t - Timestamp
- p - Desired position
- v - Desired velocity
- a - Desired acceleration

Given Δt, demanded position P for axis is the calculated as:

- $P = p + v \times \Delta t + a \times \Delta t^2$

DSP for axis, UMAC, TPLC, ECS PLC & HBS PLC

- Axis DSP control axis motors from MCSPU jet's computer
  - Electrical lines from jet to motors
- UMAC motions controllers are primary used on OSS (Optical Support Structure)
  - M3 rotator, M2 hexapod
  - Network controlled
- PLCs provides intelligent control of relays
  - Switch off/on things
  - HBS management, ...
  - Provides extra security layer for better safety - PLC shall refuse command which can do a harm (axis interlocks, stow pins retraction/deployment,...)

VxWorks

- Real-time OS running on a specialized computer
- Control primary mirror movement
  - Pneumatic actuators
  - Air compressors
- Reads mirror sensor
  - Position
  - Temperature
- Sided (2x, SX and DX sides)

Chain of commands

- IIF - provides ICE interfaces
- PCS - generates polynomials for MCSPU
- MCSPU - executes polynomials received from PCS on axis, talks to TPLC
- PSF - calculates commands for PMC
- PMC - moves M1
- GCS - controls autoguiders and wavefront CCDs
- AOS - interfaces AO arbitrator

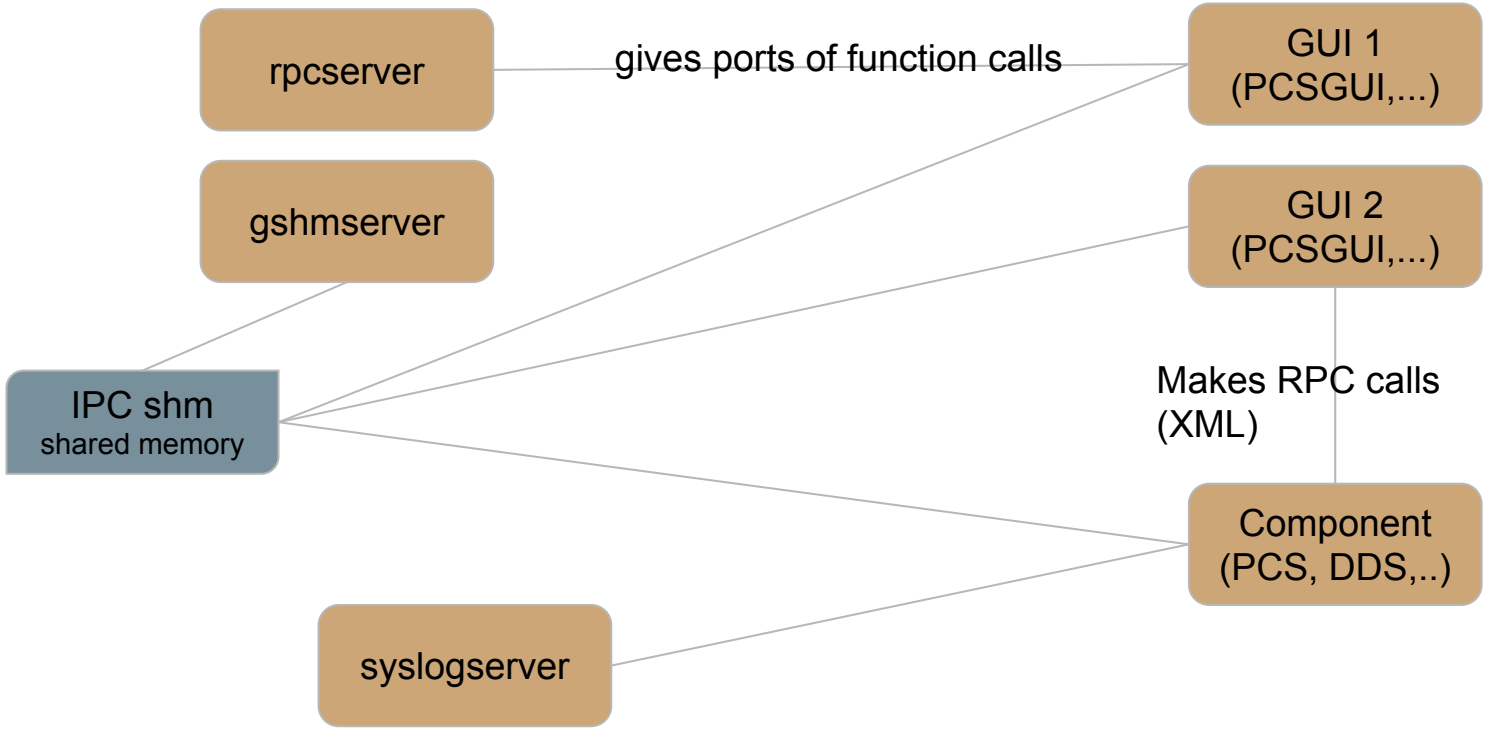
Auxiliary services

- LSS - display events, write events to a disk file, produces audible warnings
- DDS - provides read-only access to selected shared memory variables
- ENV - weather
- ECS - enclosure, building AC, ..
- MCS - MCSPU GUI
- OSS - optical arms

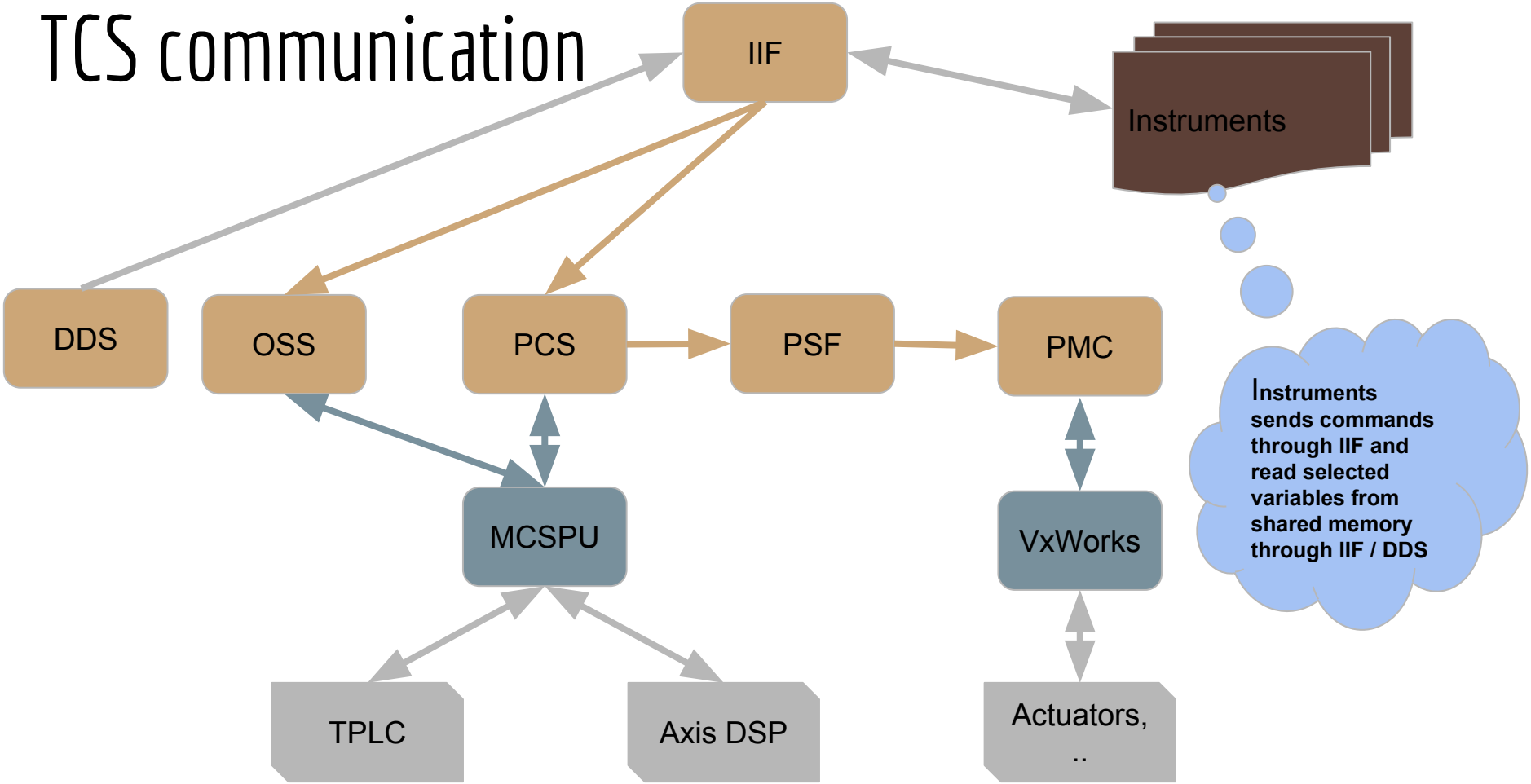
TCS cluster


jet
TCS1
TCS sys
TCS2
TCS sys
TCS3
laptop
obs
desktops
mcstemp
mcsDisplay
PCS, MCS, ..
TCSGUI, PCSGUI, ...
ECS

TCS cluster node



TCS communication



TCS log files & telemetry

- Log files in /lbt/log/YYYY/MM/YYYYMMDD*
  - Most important are YYYYMMDD.events (events log) and YYYYMMDD.log (“syslog”)
- Telemetry under /lbt/telemetry_data/tcs
  - Hdf5 format, needs special libraries to read (Python: pyh5)
  - See http://telemetry.lbto.org/visualization/telem_vis.html
  - See https://wiki.lbto.org/bin/view/Software/TCSTelemetryGraphTool

(Re) starting things

Manager, mechanics and software engineer travelling in a car. Car broke in middle of nowhere. Looking for ways to finish the journey:

Manager: let's call service to tow us, and rent a car

Mechanics: let's try to fix the car, should be simple enough

Software engineer: let's all step out and back in the car

(Re) starting things

- Most common strategy (after collecting enough evidence)
- netconfig start PCS
  - Try netconfig start PCS on local
  - Or even better, do valgrind netconfig start PCS on local
- Look for core dumps
  - On TCS machines, in /home/tcs
  - On Jet machine, in /home/telescope

[tcs@tcs1 ~]$ file core*

core.20420: ELF 64-bit LSB core file x86-64, version 1 (SYSV), SVR4-style, from 'PCS NONE'

....

(Re) starting things

- PDUs on summit
  - https://wiki.lbto.org/bin/view/InformationTechnology/PDUandKVMList
- Restarting TCS processes
  - netconfig stop PCS
  - netconfig kill PCS
  - netconfig start PCS
- TCSGUI (demonstration)

When on call..

When SW on call

Expect to receive 0-2 calls per month!

(night staff get trained a lot dealing with common issues)

Know what is potentially unsafe

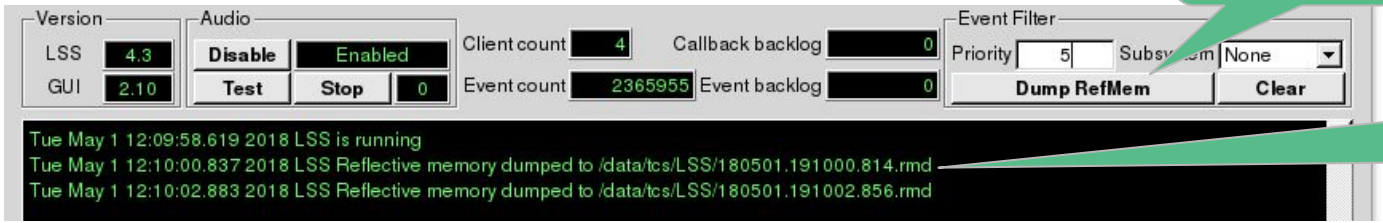
- Move telescope so it can harm somebody - check with TO
- DM (Deformable Mirror) limits
- Watch for M1 safety (don’t disable too many actuators)
- Everything else seems to be safe - you actions shall not cause permanent damage *(you will need to modify PLC code to put telescope in danger, for which we aren’t trained and don’t have the rights)*

Checklist before starting SW on call duty

- **Being able to connect to VPN from your home**
- Being able to ssh into tcs1 and tcs2
- Being able to run GUIs on tcs1 and tcs2
- Being able to ssh into jet computer and run mcsDisplay there
- Knows TCS subsystems and communication patterns
- Knows log files location
- Knows troubleshooting checklists
- Know the tools (netconfig, rpcconfig, DDAccess, DDBrowser)
- Understand how devices are commanded (serial line, network,..)
- Look for know issues
- Read https://wiki.lbto.org/bin/view/Software/GeneralTips
- Read other documentation, talk with your peers
- **Make sure mountain can contact you**
  - **Add mountain numbers to Priority, so they will ring even in silence mode (Android)**
- Check you have all relevant numbers in your cell phone (UA Network tech, ..)

Common incident - something is not working

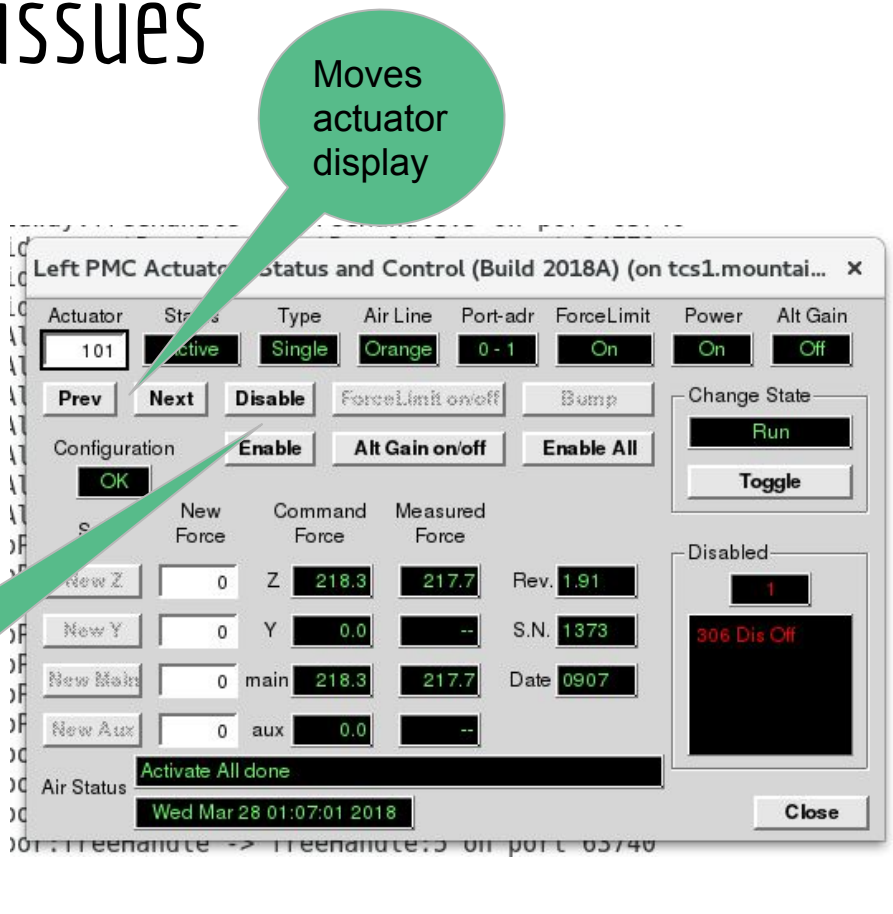
1. Ask TO (or DIY) to save reflective memory data



2. Pinpoint which TCS code is misbehaving *(probably the most difficult part)*
3. Suggest a solution
    a. Restarting component
    b. Modifying component behaviour from GUI
    c. Check for component connectivity
    d. Anything else..be creative!

Common incident - PMC issues

- Try to isolate fault component (see PMC demonstration)
- Mirror is safe without VxWorks
  - It is safe to reboot VxWorks anytime, mirror will rest on support spring
- You might want to try disable actuators
  - Safe when mirror is down/panicked



Common incident - PCS complains

- Reduce PCS complexity
  - Disable “Update from Ephemeris”
  - Keep in mind co-pointing checks can be disabled in PCS

Disabling this will mute “copointing” complains. (at the cost of observatory efficiency - script might hit the hard limits, mirror will panic)

Disabling this might save you from NS tracking issues. Costing you possible problems on fast moving (fast rate changing) objects.

Common incidents - firmware

- **You aren’t supposed to change firmware alone, always consult Engineering!**
- PMC has firmware in every accuator, you can update it from VxWorks (see PMC VCan document)
- UMAC motion controllers runs firmware (there is a Wiki/VCan with update instructions)
- Think twice before starting firmware update
    - You can easily do more damage than good (chances to brick the device, so it must be removed and and reprogram on a bench - particularly high for PMC actuators)