

DAQ Restart Scripts

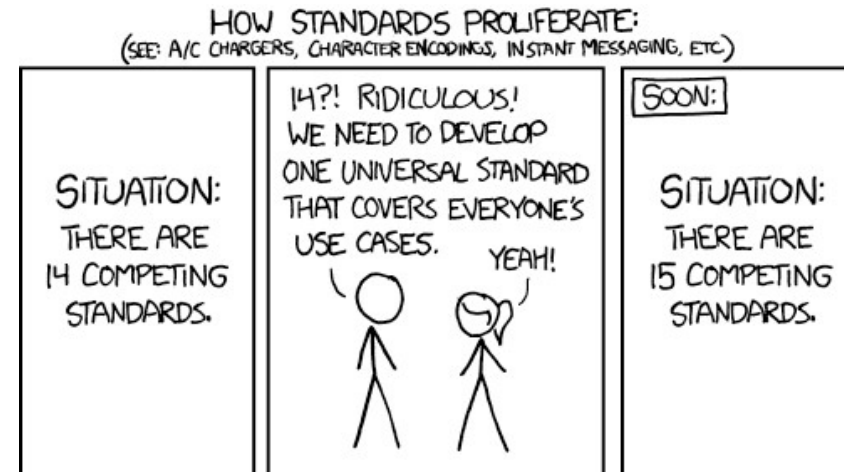
Oskar Hartbrich
B2 TRG/DAQ WS,
Yonsei University, 08/28/2019

DAQ Restart

- After power cycle (or fatal crash) of a given DAQ machine, we need to reliably bring it into a usable state for DAQ operations
 - Start core software: nsm2, Zabbix, ...
 - Run slowcontrol apps: runcontrold, hvmasterd, ...
 - Extra tools: run elog daemon etc.
 - Different machines need to start different software
- Old solution: restart_*.sh, bootnsm, bootcpr in daq_slc repository
 - Not well maintained, no common structure
 - Advantage: reads nsm ports from daq_slc configs
- New solution: daq_restart repository
 - 100% bash

daq_restart

- Agnostic to bash/csh due to some shell magic
- Nakao-san wrote very nice nsm2 starting tool
 - Acts as “single source of truth” for nsm2 ports on all relevant machines
- OH wrote tool to start/stop slowcontrol apps
 - relatively nice implementation for readout PCs + coppers
 - Stable and fast
 - Used by some people, but never officially introduced to DAQ and subdetector experts
 - Terrible logging, terrible user output



daq_restart – reloaded

- Learned so much about bash that I rewrote things from scratch
- Now much nicer logging, useful user output summary
 - Logging very close to default logging of daq_slc tools
 - User screen output only gives overview of success/fail in tree hierarchy
- Either works on current machine/subdetector only, or try to connect to all known machines in daqnet
- Modes:
 - start: start all missing/bad status processes
 - stop: stop all processes
 - restart: stop, then start
 - status: give slc status overview for machine(s)
- Integrated “daemonizer” to start nsm2cad (etc.) in background and log all outputs, no more screen/tmux needed
 - Full support of “env” command, can easily run with specific env variables

```
@top01:/log_bash
@top01:/log_bash 85x26
[b2top@top01 log_bash]$ ./test_main.sh
>>>> top01: test main:
>>>> top01->top01: start ropc:
>>>> top01->top01->cpr3004: this is a copper [ OK ]
>>>> top01->top01->cpr3003: this is a copper [ OK ]
>>>> top01->top01->cpr3002: this is a copper [ OK ]
>>>> top01->top01->cpr3005: this is a copper [ OK ]
>>>> top01->top01->cpr3002: this is a copper [ OK ]
>>>> top01->top01: ropc [ OK ]
>>>> top01->top02: start ropc:
>>>> top01->top02->cpr3009: this is a copper [ OK ]
>>>> top01->top02->cpr3010: this is a copper [ OK ]
>>>> top01->top02->cpr3007: this is a copper [ FAIL ]
>>>> top01->top02->cpr3006: this is a copper [ FAIL ]
>>>> top01->top02->cpr3008: this is a copper [ FAIL ]
>>>> top01->top02: ropc [ FAIL ]
>>>> top01->top03: start ropc:
>>>> top01->top03->cpr3011: this is a copper [ FAIL ]
>>>> top01->top03->cpr3014: this is a copper [ OK ]
>>>> top01->top03->cpr3015: this is a copper [ FAIL ]
>>>> top01->top03->cpr3012: this is a copper [ FAIL ]
>>>> top01->top03->cpr3016: this is a copper [ FAIL ]
>>>> top01->top03->cpr3013: this is a copper [ FAIL ]
>>>> top01->top03: ropc [ FAIL ]
>>>> top01: test main: [ OK ]
[b2top@top01 log_bash]$
```

```
@top01:/log_bash
@top01:/log_bash 52x26
[b2top@top01 log_bash]$ ./test_main.sh
>>>> top01: test main:
>>>> top01->top01: start ropc:
>>>> top01->top01->cpr3003: this i... [ FAIL ]
>>>> top01->top01->cpr3005: this i... [ OK ]
>>>> top01->top01->cpr3004: this i... [ OK ]
>>>> top01->top01->cpr3002: this i... [ FAIL ]
>>>> top01->top01->cpr3002: this i... [ OK ]
>>>> top01->top01: ropc [ FAIL ]
>>>> top01->top02: start ropc:
>>>> top01->top02->cpr3008: this i... [ FAIL ]
>>>> top01->top02->cpr3006: this i... [ FAIL ]
>>>> top01->top02->cpr3009: this i... [ FAIL ]
>>>> top01->top02->cpr3010: this i... [ OK ]
>>>> top01->top02->cpr3007: this i... [ OK ]
>>>> top01->top02: ropc [ FAIL ]
>>>> top01->top03: start ropc:
>>>> top01->top03->cpr3012: this i... [ OK ]
>>>> top01->top03->cpr3014: this i... [ OK ]
>>>> top01->top03->cpr3013: this i... [ OK ]
>>>> top01->top03->cpr3016: this i... [ OK ]
>>>> top01->top03->cpr3011: this i... [ OK ]
>>>> top01->top03->cpr3015: this i... [ FAIL ]
>>>> top01->top03: ropc [ FAIL ]
>>>> top01: test main: [ OK ]
[b2top@top01 log_bash]$
```

Status check

- nsm2 status: check process exists, parse nsminfo2 output
- slc apps status: check process exists, request version number nsm variable
 - Can check for correct version and process alive
- EPICs apps: check process exists, ... ?
- Other scripts (e.g. run elog): check process exists?
- Finally want to include status monitoring in Zabbix as well

Final Words...

- When is it available? Very soon. (just like the last times...)
 - Propose a training session for DAQ shifters and detector experts before beam runs resume
- Nils plans to use the same “framework” for HLT restart scripts
- We are totally reinventing a wheel here. Industry needs reliable job management/machine setup on far larger scales.
- Excellent FOSS software exists to deal with this issue, e.g. Apache Mesos/Aurora
 - A lot more features, very reliable! See Nils’ talk later today
 - Should consider this for the DAQ upgrade