CCM readiness for VD coldbox testing

Pierre Lasorak

Introduction

- New features expected for the dunedaq-v2.8.1 for VD coldbox tests
- Run control - NanoRC
  - Run number database
  - Run configuration
  - (Commands distribution)
  - (Subsystems)
- Dashboards
  - Operational monitoring, ERS

Run DB

- Need an interface with the run database to save the run number when a tester enters “start” in NanoRC
- Right now, the user manually enters the run he/she decides is good
- E. Gamberini et al. created a flask server that interfaces with the run number database
  - Similar to what is used for ProtoDUNE
  - https://gitlab.cern.ch/ep-dt-di/daq/np04rn-rest
- NanoRC will use this server to get/save/increment the run number
  - PR: https://github.com/DUNE-DAQ/nanorc/pull/24
- Still needs work:
  - Hard coded IP/port in NanoRC (flask server runs in a tmux session on np04-srv-024) → make it a service and give it a name?
  - Authentification
    - How do we do that?

Run DB

nanorc

```
shonky rc> start
[11:00:00] INFO     Sending start to lr0 (http://np04-srv-022:3333/command)
           INFO     Response: <Response [202]>
           INFO     Received reply from lr0 to start
           INFO     Sending start to lr1 (http://np04-srv-022:3334/command)
           INFO     Response: <Response [202]>
           INFO     Received reply from lr1 to start
[11:00:00] Started run #64
```

Apps

name	host	alive	pings	last cmd	last succ. cmd
lr0	np04-srv-022	True	True	start	start
lr1	np04-srv-022	True	True	start	start

```
shonky rc> stop
[11:00:05] INFO     Sending pause to lr0 (http://np04-srv-022:3333/command)
           INFO     Response: <Response [202]>
           INFO     Received reply from lr0 to pause
           INFO     Sending pause to lr1 (http://np04-srv-022:3334/command)
           INFO     Response: <Response [202]>
           INFO     Received reply from lr1 to pause
```

Apps

name	host	alive	pings	last cmd	last succ. cmd
lr0	np04-srv-022	True	True	pause	pause
lr1	np04-srv-022	True	True	pause	pause

```
           INFO     Sending stop to lr0 (http://np04-srv-022:3333/command)
           INFO     Response: <Response [202]>
           INFO     Received reply from lr0 to stop
           INFO     Sending stop to lr1 (http://np04-srv-022:3334/command)
           INFO     Response: <Response [202]>
[11:00:06] INFO     Received reply from lr1 to stop
[11:00:06] Stopped run #64
```

Meanwhile, on the flask server…

```
10.73.136.67 - - [11/Oct/2021 11:00:00] "GET /runnumber/increment HTTP/1.1" 200 -
THREAD  Thread-65
  -> Using connection from pool: <cx_Oracle.Connection to externally identified user>
  -> Perform query done...
No rows...
  -> fetched  1  rows...
[[(64,)]]
10.73.136.67 - - [11/Oct/2021 11:00:00] "GET /runnumber/get HTTP/1.1" 200 -
THREAD  Thread-66
  -> Using connection from pool: <cx_Oracle.Connection to externally identified user>
  -> Perform query done...
No rows...
  -> fetched  1  rows...
[[(64,)]]
10.73.136.67 - - [11/Oct/2021 11:00:06] "GET /runnumber/get HTTP/1.1" 200 -
THREAD  Thread-67
  -> Using connection from pool: <cx_Oracle.Connection to externally identified user>
  -> Perform transaction done...
  -> Committed...
THREAD  Thread-67
  -> Using connection from pool: <cx_Oracle.Connection to externally identified user>
  -> Perform query done...
No rows...
  -> fetched  1  rows...
[[(datetime.datetime(2021, 10, 11, 11, 0, 0, 708964), datetime.datetime(2021, 10, 11, 11,
10.73.136.67 - - [11/Oct/2021 11:00:06] "GET /runnumber/updatestop/64 HTTP/1.1" 200 -
```

Run configuration

- We need a way to save the run configuration:
  - All the JSON files + run time variables
  - That can then be saved (how?) in the Run DB…
- Current implementation:
  - Save all the JSON files when user enters “start” + the runtime configuration in a directory called `RunConf_${run_number}`
  - Whenever a “resume” is issued, the runtime configuration can change, so save the the resume commands as well
  - `ConfigManager::runtime_start` now also returns the directory name in nanorc
    - That path can then be saved in the RunNumber DB on “stop”? Maybe out of scope for this time.
  - https://github.com/DUNE-DAQ/nanorc/pull/25

Run configuration

nanorc

```
shonky rc> start 123456
[11:26:38] INFO     Sending start to lr0 (http://np04-srv-022:3333/command)
           INFO     Response: <Response [202]>
           INFO     Received reply from lr0 to start
           INFO     Sending start to lr1 (http://np04-srv-022:3334/command)
           INFO     Response: <Response [202]>
           INFO     Received reply from lr1 to start
[11:26:38] Started run #123456
           Saving run data in /nfs/home/plasorak//RunConf_123456/
```

...

```
shonky rc> resume
[11:26:44] INFO     Sending resume to lr0 (http://np04-srv-022:3333/command)
           INFO     Response: <Response [202]>
           INFO     Received reply from lr0 to resume
           INFO     Sending resume to lr1 (http://np04-srv-022:3334/command)
           INFO     Response: <Response [202]>
           INFO     Received reply from lr1 to resume
                                Apps
┏━━━━━━┳━━━━━━━━━━━━━┳━━━━━━━┳━━━━━━━┳━━━━━━━━━┳━━━━━━━━━━━━━━┓
┃ name ┃ host        ┃ alive ┃ pings ┃ last cmd ┃ last succ. cmd ┃
┡━━━━━━╇━━━━━━━━━━━━━╇━━━━━━━╇━━━━━━━╇━━━━━━━━━╇━━━━━━━━━━━━━━┩
│ lr0  │ np04-srv-022 │ True  │ True  │ resume   │ resume         │
│ lr1  │ np04-srv-022 │ True  │ True  │ resume   │ resume         │
└──────┴──────────────┴───────┴───────┴──────────┴────────────────┘
shonky rc> stop
```

Saved run env

```
np04-srv-022:~/RunConf_123456 > l
total 48K
-rw-r--r-- 1 plasorak np-comp 1.7K Oct  8 17:36 boot.json
-rw-r--r-- 1 plasorak np-comp   86 Oct  8 17:36 conf.json
drwxr-xr-x 2 plasorak np-comp 4.0K Oct 11 11:26 data/
-rw-r--r-- 1 plasorak np-comp   86 Oct  8 17:36 init.json
-rw-r--r-- 1 plasorak np-comp   88 Oct  8 17:36 pause.json
-rw-r--r-- 1 plasorak np-comp   90 Oct  8 17:36 resume.json
-rw-r--r-- 1 plasorak np-comp  220 Oct 11 11:26 resume_parsed_2.json
-rw-r--r-- 1 plasorak np-comp  220 Oct 11 11:26 resume_parsed.json
-rw-r--r-- 1 plasorak np-comp   88 Oct  8 17:36 scrap.json
-rw-r--r-- 1 plasorak np-comp   88 Oct  8 17:36 start.json
-rw-r--r-- 1 plasorak np-comp  344 Oct 11 11:26 start_parsed.json
-rw-r--r-- 1 plasorak np-comp   86 Oct  8 17:36 stop.json
```

Commands distribution

- Idea is that some commands can happen in parallel, others can't
  - WIB, Timing, …
- Some commands take a very long time to execute (c.f. WIB config at PD1)
- Implementation
  - Need to be able to specify order when requested from the user
  - Only started on Friday afternoon… But I'm hopeful I can get this to work by the end of the week
  - Idea is that `ApplicationManager::send_command` has a flag to wait for the response of the command or not, and command can be executed both sequentially or in parallel.

Dashboards

- Work by Marco Roda
- Instructions:
  - https://github.com/DUNE-DAQ/minidaqapp/wiki/Graphic-process-monitoring
- Dashboard repository (any dashboard can be specified in JSON)
  - https://github.com/DUNE-DAQ/grafana-dashboards
  - np04-srv-009 will have updated dashboards before VD coldbox tests start
- Valuing feedback from users, urgent or not! Contacts:
  - Sammy Valder: S.L.Valder@sussex.ac.uk
  - Marco Roda: mroda@fnal.gov
- Presentation at the CCM meeting this Wednesday on how to create uniform style, organisation

Conclusion

- I started working last week on NanoRC:
  - NanoRC / Run number DB interface in a PR
  - NanoRC config saving in a PR
  - More work needed for parallelisation
  - More work needed to get a `np04-nanorc` (with all the DB authentification, write permissions in place) and a `test-nanorc` (when people want to play around)
  - Haven't had a look at the subsystems (2 config dirs etc) yet.
- Dashboards:
  - Developed by Marco and Sammy
  - Going to be updated on srv-009
  - Documented
- More discussion about these 2 topics on Wednesday at the CCM meeting!