



# ALICE XRootD setup

Adrian Sevcenco, ISS, RO

https://github.com/adriansev/alicexrd

# Outline

- Software requirements and installation
- xrd.sh : environment variables
- xrd.sh : usage options
- xrd.sh : ALICE monitoring of xrootd metrics
- XRootD configuration options
- Storage tuning

# Software requirements

- Repositories : epel and WLCG
- EL6 no longer supported (by me, in relation to xrd.sh)
  - EL8 is already GA :), soonish we will have Centos 8
- curl, bind-utils, bzip2 : used by the script
- “alicexrdplugins” meta-package will pull the rest of dependencies

# Environment variables

- Location variables
  - Internal XRDSHDIR is the location of xrd.sh
  - XRDCONFDIR – defaults to ${XRDSHDIR}/xrootd.conf/
  - XRDRUNDIR – defaults to ${XRDSHDIR}/run/
  - XRDCONF – defaults to ${XRDCONFDIR}/system.cnf
    - All these should be always set (use .bashrc)
- Functional variables
  - XRDSH_DEBUG – enable output of network detection
  - XRDSH_NOWARN_ASLIB – if the script is sourced, do not warn
  - **XRDSH_NOAPMON – do not use apmon perl script**
    - **Use mlsensor agent for reporting, more info later**
  - XRD_DONOTRECONF – the configuration file will NOT be recreated at each invocation of xrd.sh
  - XRDREADONLY – if set AND configuration recreated it will disallow writes (and report ReadOnly status to redirector)
    - Or just change all.export declaration from “writable” to “notwritable”

# Command options

- -getkeys : no longer needed/used, they are part of xrootd-alicetokenacc
- -addcron : (re)install the user cron for service checking
- -logs : compress the logs; does not send signal, log files are not rotated
- -k : kill running processes
- -c : check is process is running, if not, restart it
- -f : restart services (kill + start)
- -limits : generate the limits files for the current user
- **-conf** : without arguments it recreates the XRootD conf file
  - One argument (file) : write configuration file to this file (default template)
  - Two arguments : <template> <conf file> : write the specified configuration file using the specified template
- -f, -c, -k : have the same options
- -systemd : generate systemd service file for xrootd and cmsd services
  - These are linked and ordered

# ALICE XRootD monitoring

- There are 2 sources of information :
  - IO monitoring : send to local VOBOX on UDP/9930
  - MonaLisa agent – 2 choices
    - servMon.sh script (use ApMon perl bindings to send information)
    - MIsensor – java agent that is packaged as a system service
      - Preferred choice
      - Is a service managed by system init and it have no functional dependencies on other services

# XRootD configuration options

- Quite a few knobs (here only most important IMHO)
  - xrd.timeout : it manages the connections timeouts
    - The defaults does not close the idle connections
  - cms.space : selects the server selection for writing
  - xrd.sched : threads management
    - Defaults are : mint 8 maxt 2048 avlt 512 idle 780
    - Because of defaults we had 2k+ load on IOWait
      - We reduced to a small multiple (2) of system threads
  - xrootd.async : ALICE used it as off
    - We began to use as : force limit 8 maxsegs 8 maxstalls 4 maxtot 2048 segsz 64k syncw
    - The writing (fsync) is still synchronous
  - cms.sched : load balancing
    - cpu 50 io 50 refreset 1800
    - Equal weights of cpu and io when computing load score

# Storage tuning

- Just to share what we use
- 3 subsystems : Kernel tuning, Block device tuning and Network tuning
- Kernel tuning - usually vm related, see sysctl-explorer
- Block device tuning
  - It is always best if “performance” governor is used
  - The scheduler
    - Since 5.0 there are not many options
    - Mq-deadline a good coverall (multi-queue + sorting)
    - echo 2048 > ${blk}/queue/nr_requests
    - echo 4096 > ${blk}/queue/read_ahead_kb
    - echo 512 > ${blk}/queue/max_sectors_kb
      - this should be <= ${blk}/queue/max_hw_sectors_kb
- generic file system tuning
  - noatime, align(make aware) the file system on stripe,stride

# Storage tuning

- Network tuning
  - The packet scheduler : net.core.default_qdisc = fq
  - Congestion protocol : net.ipv4.tcp_congestion_control = bbr
  - Kernel network infrastructure tunables :
    - Quite a lot (and some debatable as the effect depends on the kernel version)
    - We use these sysctl configurations
      - Any comment/view/opinion on these are more than welcome
      - It would be great to have more specific guidelines and maybe some starting points (not templates as everyone situation is unique)
    - N.B. we consistently use the mainline from ELRepo

# Thank you!

adrian.sevcenco@cern.ch