alien.py
Using the alien.py client for site operations

Adrian Sevcenco, ISS, RO

https://gitlab.cern.ch/jalien/xjalienfs

https://pypi.org/project/alienpy/

Target version: 1.4.4

Where to find

- **Requirements**
  - websockets, xrootd, async-stagger, pyOpenSSL
  - addition of “rich” (formatting), “requests” (safe and easy http access)

- **xjalienfs** package in alidist recipes: aliBuild build xjalienfs

- **Packaged for Pypi** repository: https://pypi.org/project/alienpy/
  - So, just “pip install alienpy”

- **Singularity container**
  - singularity run oras://registry.cern.ch/asevcenc/alienpy:[latest|tag] [cmd]
    - Latest (master) is not always up to date

Reference version (this presentation) : 1.4.4
Main documentation link: alien.py commands reference guide*
Last "tutorial" style presentation
***please send feedback on topics that would be helpful to have better/longer explanations**

Site admin tools

- Storage tools (XRootD based)
  - xrd_config
  - xrd_stats
  - xrd_ping
  - xrdstat
  - pfn
  - pfn-status
  - listSEs
  - getSE
  - listSEDistance
  - testSE
- Reporting/Information tools
  - queryML
  - listCEs
- Job inspection tools
  - ps
    - -jdl
    - -trace

Storage tools

- xrd_config [-v | -verbose] FQDN | ML Storage name (or part of) | ML ID
  - get the “xrdfs query config” output
  - verbose will add the 2nd line of information

```
alien.py xrd_config -verbose ISS::EOS
Site/XrdVer: ALICE::ISS::EOS/v5.5.0 ; TPC status: 1 ; role: none ; CMS: NOT_SET
Chksum type: 0:adler32 ; Bind max: 15 ; PIO max: 5 ; Window/WAN window: 87380/NOT_SET ; readv_{ior,iov}_max: 2097136/1024
```

  - the output json can be processed to jq

```
alien.py -json xrd_config ISS::EOS | jq '.results[0].version'
"v5.5.0"
```

- xrd_stats [-xml | -xmlraw | -compact] FQDN | ML Storage name (or part of) | ML ID
  - print the “xrdfs query stats a”
  - either in native xml (-xmlraw: compact, native to xrdfs) or pretty printed (-xml)
  - otherwise translated to json (and pretty printed) or compacted with -compact

```
alien.py xrd_stats fst07.spacescience.ro:1095 | jq '.stats.sched|.jobs,.threads'
"676434710"
"107"
```

- xrd_ping [-c count] FQDN | ML Storage name (or part of) | ML ID
  - XRootD ping (so we can have some RTT even when no ICMP)

```
alien.py xrd_ping ISS
XRootD ping(s): 3 time(s) to:
ALICE::ISS::EOS                 rtt min/avg/max/mdev (ms) = 2.806/3.067/3.264/0.236
ALICE::ISS::FILE                rtt min/avg/max/mdev (ms) = 2.746/2.881/3.000/0.128
```

Storage tools

- xrdstat [-d [-i]] [-v] [-p PID,PID,...] [-s SE1,SE2,...] [-c] <filename1> [<or UUID>]
  - server based (central services command)
  - complex checking (including downloading and validation of the file/GUID)
- pfn lfn
  - print only the recorded PFNs for a LFN (simplified output of whereis)
- pfn-status pfn | lfn
  - query the XRootD server for the GUID status
- listSEs [-qos filter,by,qos] [-s] [SE name] [SE name]
  - server based (central services command)
  - list the informations for registered SEs
- getSE <-id | -name | -srv> identifier_string
  - return/translate any king of identifier to id/name/manager fqnd of SEs
- listSEDistance -site | -read | -qos
  - server based (central services command)
  - Returns the closest working SE for a particular site
- testSE [-v | -c | -t | -a] <some SE names, numbers or @tags>
  - server based (central services command)
  - Test the functional status of Grid storage elements

Site and jobs tools

Reporting/Information tools

- queryML
  - interface to MonaLisa provided REST endpoint (alimonitor/REST)
  - format of path: <FARM>/<CLUSTER>/<NODE>/<PARAMETER>
  - time range can be inserted before the parameter with the form: /T_BEGIN/T_END (unix time, ms)
    - if negative times: X ms since now
  - as usual, -json will return the json form of the answer (if not, notify me)

- listCEs
  - server-side implemented
  - get a listing of CE names, status, max {running,queued}, TTL, Type, Host
  - N.B. CE name is the tag associated to the submitting vobox
- getCE (similar to getSE) : in a next version

Site admin tools

Job inspection tools

- ps
  - have the potential to list all jobs running on site:
    - alien.py ps -a -f r -s SITE_NAME
      - -a (all users) is required or a user specified
  - -jdl : print the job jdl
    - almost json: with some processing it can be proper json and the be possible to be filtered
  - -trace : print the trace for job operations
    - bulk text only
    - BatchId is the local job id, so the rest can be followed
      - ?? Would be possible to provide as json fields:
        - BatchId (list as it can be more than one)
        - workdir
        - Job Agent version
        - Worker node hostname
        - State
    - Q: Is there any CE identifier (ARC-CE, HTCondor) that can be also reported in similar manner?

Site admin tools

Job inspection tools

- More informations can be only found only as privileged user for resource managers access
  - as personal TODO, i do have a plan to gather together information from central services and link it with information from RM (with ssh access)
    - “The AliEn job id X : where is on my site (JobID, host, PID(s) list), what it is doing/ did, how many resources were reported by RM as being consumed”
    - this would need specific information for each RM type (how to list the job information)
  - not sure if this would be of general interest

Support Info

- JAliEn issues - the place to report JAliEn related problems (alien.py included)
- JAliEn channel on CERN Mattermost - @asevcenc for direct message

Costin.Grigoras@cern.ch - for JAliEn features
Adrian.Sevcenco@cern.ch - alien.py features

THANK YOU !