

pyJAliEn client

Adrian Sevcenco, ISS, RO
https://github.com/adriansev/jalien_py

Outline

- **Why**
- **How**
 - Python
 - Websockets
 - Asyncio
 - JSON messages
- **What (can it do)?**
 - Metadata related manipulation
 - Job Management
 - File operations

Why

- New Java based central services – JAliEn
 - There is an jalien.sh client (Java based)
 - Alternative client needed for flexibility - pyJAliEn
- The new JAliEn services offer websocket endpoints for communication
 - Websocket is HTTP with some important extras:
 - Full duplex, connection-oriented communication
 - Stateful protocol
 - Optimized, low-latency communication

How

- Python (3.6 or higher)
 - Universal
 - Simple, natural language
 - Requirements : pyOpenSSL, websockets, pyxrootd
 - Waiting for xrootd upstream fixes to trickle down to pypi repository
- Websocket connection
 - SSL connection to upstream
 - Serialized queries (send/receive dialogue)
 - Simple communication, optionally compressible

How (cont.)

- Asyncio
 - Communication infrastructure (transport and protocol abstractions, multiplexing I/O access over sockets)
 - Single threaded concurrent code (synchronization primitives)
 - Coroutine* paradigm (async def – coroutine functions, await – suspend execution coroutine completion until return data is fetch and returned)

*Coroutine: functions/components that can be suspended/resumed – it can (a)wait for the results

How (cont.)

- JSON
 - Easy serialization of “items” and concepts (dictionaries, lists, maps)
 - Native serialization/deserialization of python objects
 - Space efficient (w.r.t. xml)
 - Easier to use in developing

What (can it do)?

- API/metadata operations
 - Catalog manipulation (files metadata) (examples in command mode)
 - `alien.py ls -l`
 - `alien.py mkdir my_new_dir`
 - `alien.py stat /my_alien_file`
 - Job management
 - `alien.py ps -A`
 - `alien.py kill <job_id>`

What (can it do)?

- XRootD operations (cp)
 - Copy from local to grid (upload)

```
alien.py cp file:///testfile /alice/cern.ch/user/a/asevcenc/test_upload.txt
jobID: 1/2
jobID : 1 ; processed: 12499, total: 12499
STATUS: OK ; MESSAGE: [SUCCESS]
jobID: 2/2
jobID : 2 ; processed: 12499, total: 12499
STATUS: OK ; MESSAGE: [SUCCESS]
```

- Copy from grid to local (download)

```
alien.py cp /alice/cern.ch/user/a/asevcenc/testfile file:///download.txt
jobID: 1/1
jobID : 1 ; processed: 12499, total: 12499
STATUS: OK ; MESSAGE: [SUCCESS]
```

- NO grid to grid copy! (replication is done centrally)

Uploading (writing) still a WIP. Do not rely on it yet.

What (can it do)?

- Shell interaction
 - ‘Sh:’ - will execute everything after the tag in the shell
 - ‘|’ will pipe the result from the left side to the shell commands of the right side
- Command line or interactive mode
 - Anything after ./alien.py will be taken as JAliEn command
 - Interactive mode if there is no argument

Debugging

For those interested in under-the-hood mechanics:

- Environment variables (set to anything):
 - JALIENPY_DEBUG : printouts of sent messages
 - JALIENPY_XRDDEBUG : printouts of command and parameters related to XRootD commands
- Enabling debug by the script name
 - If script have _json in name : it will print raw json messages (full answer of the server)
 - If script have _json_all in name : it will print raw json messages (full answer of the server) including the metadata section of the answer

Future development

- Recursive copy
 - Uploading/Downloading directories
- Smart interactive interface : history file, Up/Down, Left/Right keyboard input
 - Ådne Nerheim : will work on full interactive shell implementation in Python

Examples

Command mode examples

Get the stat information of a file

```
alien.py stat /alice/cern.ch/user/a/asevcenc/testfile
File: /alice/cern.ch/user/a/asevcenc/testfile
Type: f
Owner: asevcenc:asevcenc
Permissions: 755
Last change: 2019-03-24 16:29:46.0 (1553441386000)
Size: 12499 (12.21 KB)
MD5: e48b12de59d6251c18156e0232c79a7c
GUID: a7f1c690-4e49-11e9-8a64-0242266c4d21
    GUID created on Sun Mar 24 16:29:46 CET 2019 (1553441386873) by 02:42:26:6c:4d:21
```

Listing a file

```
alien.py ls -l /alice/cern.ch/user/a/asevcenc/testfile4
-rwxr-xr-x  asevcenc asevcenc          0 Apr 01 12:10  testfile4
```

Find all files in specified directory with the pattern

```
alien.py find /alice/cern.ch/user/a/asevcenc test*
/alice/cern.ch/user/a/asevcenc/testfile
/alice/cern.ch/user/a/asevcenc/testfile2
/alice/cern.ch/user/a/asevcenc/testfile3
/alice/cern.ch/user/a/asevcenc/testfile4
```

Examples

Interactive mode examples

List first 3 jobs of user aliproduct

```
jsh: /alice/cern.ch/user/a/asevcenc/ > ps -u aliproduct | head -n 3
aliproduct 834810241  SP AOD_merge.sh
aliproduct 921515484  SP AOD_merge.sh
aliproduct 921538600  D  AOD_merge.sh
```

Print/use an environment variable within shell

```
alien.py
jsh: /alice/cern.ch/user/a/asevcenc/ > sh:echo "$TMPDIR"
/home/adrian/tmp
```

Within client, list the files and then use shell constructs to filter and count

```
jsh: /alice/cern.ch/user/a/asevcenc/ > ls | grep '^test' | wc -l
2
```

Command mode, chain commands example:

For each jobID of user adobrin (but with a limit of 1) print the jdl

```
for id in $(alien.py ps -u adobrin -A -l 1 | awk '{print $2}'); do alien.py ps -jdl ${id}; done
InputDataListFormat = "xml-single";
Validationcommand = "/alice/cern.ch/user/a/adobrin/treePass118r/datapass1_validation.sh";
Executable = "/alice/cern.ch/user/a/adobrin/treePass118r/datapass1.sh";
```

Feedback

- A feedback on what features/commands are missing at this point would be really useful
 - https://github.com/adriansev/jalien_py/issues
- This is the best time for improvements/additions in both JAliEn server code and also in clients code

Costin.Grigoras@cern.ch - for JAliEn features

Adrian.Sevcenco@cern.ch *- alien.py features

THANK YOU !