pyspark-plaso

Tool for Distributed Extraction of Timestamped Events from Files

User Guide

Marek Rychlý and Radek Burget

TARZAN Project Documentation
Faculty of Information Technology, Brno University of Technology

Last changed: December 28, 2019

PySpark Plaso User Guide

The PySpark Plaso extracting process is controlled by a Web service via a REST API. The service is running in Docker Spark Application container where the PySpark Plaso build artefacts were deployed (see the build process).

Command-line Interface (CLI)

There are several shell scripts in `deployment/scripts` to transfer data between a local file-system and a distributed file-system utilised by the distributed environment (HDFS) and to control the extraction process in the environment (those scripts require curl and zip or p7zip).

- `./client-ls.sh [--url=http://0.0.0.0:5432/] [path-to-list] [another-path ...]` -- to list the content of HDFS at a particular path
- `./client-rm.sh [--url=http://0.0.0.0:5432/] <path-remove> [another-path ...]` -- to remove a file or a directory from HDFS
- `./client-download-file.sh [--url=http://0.0.0.0:5432/] <path-where-to-download> <file-path-to-download> [another-file ...]` -- to download a file from HDFS into a local file-system
- `./client-download-into-zip.sh [--url=http://0.0.0.0:5432/] <path-where-to-download> <file-or-dir-path-to-download> [another-file-or-dir ...]` -- to download a file or a directory from HDFS as a ZIP file into a local file-directory
- `./client-upload-file-dir.sh [--url=http://0.0.0.0:5432/] <path-where-to-upload> <file-or-directory-to-upload> [another-file-or-dir ...]` -- to upload a file or a directory from a local file-system into HDFS
- `./client-upload-zip.sh [--url=http://0.0.0.0:5432/] <path-where-to-upload> <zip-file-to-extract-there> [another-file-or-dir ...]` -- to upload the content of a ZIP file from a local file-system into HDFS
- `./client-extract.sh [--url=http://0.0.0.0:5432/] [path-to-extract] [another-path ...]` -- to run the extraction process on a given path in the HDFS

REST Web API

In default configuration (see the `deployment/docker-compose/webapp.yml` docker-compose file) the REST Web API is running at http://0.0.0.0:5432/. The following operation are available in the Web API:

- to list the content of HDFS at a particular path (the path can be empty to list the content of a root directory); the response is a JSON array of all directories (suffixed with `/`) and files in the path recursively
  - `GET /ls/[path-to-list]`
- to remove a file or a directory from HDFS (the path is mandatory here)
  - `GET /rm/<path-to-remove>`
  - `DELETE /file/<path-to-remove>`
- to download a file from HDFS into a local file-system (the path is mandatory here)
  - `GET /file/<file-to-download>`
- to download a file or a directory from HDFS as a ZIP file into a local file-directory (the path can be empty to get the root directory)
  - `GET /zip/[file-or-dir-path-to-download]`
- to upload a file from a local file-system into HDFS (the path can be empty to upload into the root directory)
  - `PUT /file/[path-where-to-upload]`
  - `POST /file-form/[path-where-to-upload]` -- the uploaded file is read from `file` POST parameter (suitable as a target of HTML forms)
- to upload the content of a ZIP file from a local file-system into HDFS (the path can be empty to upload into the root directory)
  - `PUT /zip/[path-where-to-upload]`
  - `POST /zip-form/[path-where-to-upload]` -- the uploaded file is read from `file` POST parameter (suitable as a target of HTML forms)
- to run the extraction process on a given path in the HDFS (the path can be empty to extract events from the root directory)
  - `GET /extract/[path-to-extract]`