Philipp Gampe, Simon Gehring – KivS PA

KivS 2014/2015 PA Exercise 3: Manual to Exercise 2

Inhaltsverzeichnis

Requirements

- python (2.7) usually available as python2.7 with modules:
  - urllib2
  - re
  - os.path
  - numpy
  - matplotlib.pyplot
  - argparse
- bash
- date

How to use

Run the script `exercise2.py` from the `Tools/` folder. The script takes various options, available via the `-h` command line option.

By default, it downloads the `PA.log` file into the `Data/` folder, unless the file is already present there. Afterwards it processes the file and plots a graph over all data points. Then it segments the data based on the TTL and plots a boxplot graph based on the determined segments.

The graphs are written out to `Data/pa_graph.svg` and `Data/pa_boxplot.svg`.

All paths my be changed via command line options. On top of that, the analyzed segment may be limited in to specific time frame given with --start and --end options. Both of them take a time stamp as value.

Command line options

```
usage: exercise2.py [-h] [--start START] [--end END] [--log-url LOG_URL]
                    [--log-file LOG_FILE] [--log-graph LOG_GRAPH]
                    [--log-boxplot LOG_BOXPLOT]

Analyze data from PING measurement.

optional arguments:
  -h, --help            show this help message and exit
  --start START         The start time for the plots. Default: 0
  --end END             The end time for the plots. Default: 2147483647
  --log-url LOG_URL     The URL do download. Default:
                        ftp://rogue-01.informatik.uni-bonn.de/PA.log
  --log-file LOG_FILE   The file to save the log file contents. Default:
                        ../Data/PA.log
  --log-graph LOG_GRAPH
                        The file to save the generated linegraph SVG image.
                        Default: ../Data/pa_graph.svg
  --log-boxplot LOG_BOXPLOT
                        The file to save the generated boxplot SVG image.
                        Default: ../Data/pa_boxplot.svg
```

Implementation details

The program is separated into multiple functions to allow reuse of the program. All options are taken from the command line and have sane defaults if omitted (see above). The main function parses the command line options and runs all steps.

All functions have an inline ReST-style documentation.