pingstats Documentation

Release 1.0.0

Ariana Giroux

Jan 21, 2020

Contents:

1 Readme **1**
- 1.1 Software Description . . . 1
  - 1.1.1 Why should you use it? . . . 1
- 1.2 Installation . . . 1
  - 1.2.1 Installing from source . . . 2
- 1.3 Usage . . . 2
  - 1.3.1 Command Line Arguments . . . 2
- 1.4 Examples of software in use . . . 3
- 1.5 Running the tests . . . 3
- 1.6 Contributing to the Project . . . 3
  - 1.6.1 Forking the project . . . 3
  - 1.6.2 Our issue tracker . . . 4

2 Authors **5**

3 Change log **7**

4 Modules **13**
- 4.1 pingstats.__init__ . . . 13
- 4.2 pingstats.ui . . . 14
  - 4.2.1 Base Widgets . . . 14
  - 4.2.2 Status Widgets . . . 15
  - 4.2.3 Plot Widgets . . . 16
  - 4.2.4 Pane Widgets . . . 16
  - 4.2.5 Creating your own Widget . . . 16
- 4.3 User Configuration . . . 17

5 MIT License **19**

6 Indices and tables **21**

Python Module Index **23**

Index **25**

CHAPTER 1

Readme

1.1 Software Description

This script provides a very simple CLI based ping visualization script by utilizing either hipsterplot or gnuplot/asciiplotlib.

The hipsterplot backend is a very simple and lightweight backend, meant to allow easier install and wider operating system reach. However, it is nowhere near as easy to read & understand as the gnuplot/asciiplotlib backend. It is highly recommended you make use of that backend if you can.

1.1.1 Why should you use it?

The long and short is that `pingstats` provides a simple and stable method to visualize average ping over time from the command line. Aside from that, it provides the following features:

- Provides a highly modular dynamic UI layout selector, allowing the user to modify the way the software presents data.
- Provides a simple API for implementing unique and new UI behaviour
- Automatically resizes and stays aware of viewport size
- Stays under a megabyte of total size

1.2 Installation

To install the base package (and use the simple backend, hipsterplot), installation has been made easy via `pip`.

```
pip3 install pingstats
```

If however, you are using Linux and can install the gnuplot library, first obtain the binary from `apt`:

```
sudo apt-get install gnuplot
```

Note: The pingstats package will automatically use gnuplot if it is available.

1.2.1 Installing from source

Installing from source can be achieved relatively simply by first cloning the repository and moving into the subsequent directory:

```
git clone https://git.eclecticmedia.space/public/pingstats.git && cd pingstats
```

Note: To checkout the newest, experimental changes ensure you are on the `master` branch

Once the repository is cloned, you can run the `setup.py` script to install the local files:

```
python3 setup.py install
```

1.3 Usage

This software was implemented with simplicity in mind, and only provides one point of access:

```
pingstats google.ca
```

1.3.1 Command Line Arguments

The software also has a variety of command line arguments avaialable:

```
usage: pingstats [-h] [-l LAYOUT] [--list-widgets] [-V] address

positional arguments:
  address               The address to ping. This could be either a web
                        address (i.e, "google.ca") or an IP address.

optional arguments:
  -h, --help            show this help message and exit
  -l LAYOUT, --layout LAYOUT
                        Specify a UI layout by listing ui elements in a comma
                        separated list where
                        'realtimepane,progstatus,averagepane' displays the
                        default layout. See the docs for more information.
  --list-widgets        Output a list of available widgets for '-l'
  -V, --version         show program's version number and exit
```

1.4 Examples of software in use

In this image, we can see two separate outputs. The top display is a display of the most recent actual return times, whereas the bottom display is the average return time for each sequence.

This display automatically scales to whatever window you have open, adding more lines and columns as necessary.

1.5 Running the tests

To run the tests, clone the repository:

```
git clone https://git.eclecticmedia.space/public/pingstats.git
```

Then simply run:

```
python3 setup.py test
```

1.6 Contributing to the Project

1.6.1 Forking the project

To fork the project, first clone it from the official repository:

```
git clone git://git.eclecticmedia.space/public/pingstats.git
```

Then, create your own public repository and set the origin url to that repository:

```
git remote set-url origin *your repository*
```

Once we have set your remote to origin, we can tell git where the upstream is:

```
git remote add upstream git://git.eclecticmedia.space/public/pingstats.git
```

Note: The upstream should be the official repository.

Note: For more on the git native forking workflow, see the git forking tutorial

1.6.2 Our issue tracker

We use a [Google Group] hosted mailing list (*pingstats@googlegroups.com*) to track and recieve new patches and bug reports.

CHAPTER 2

Authors

Outlines all author details, including contributed versions.

Author Name Ariana Giroux

Author Email ariana.giroux@gmail.com

Author Website arianagiroux.ca

Author Twitter @ArianaGiroux

Contributed versions

1. V0.01
2. V0.1
3. V0.2
4. V0.3
5. V0.3.1
6. V0.3.2
7. V0.3.3
8. V0.3.4
9. V0.3.5
10. V0.4
11. V0.4.1
12. V0.4.2
13. V0.4.3
14. V0.5a1
15. V0.5b1

16. V0.5b2
17. V0.5b3
18. V0.5
19. V0.6
20. V0.6.1
21. V0.6.2
22. V0.7
23. V0.8
24. V0.8.1
25. V0.9
26. V0.9.1
27. V0.9.2
28. V0.9.3
29. V0.9.4
30. V1.0.0

Author Name Justyn Chaykowski

Author Email jbaergeist@gmail.com

Contributed versions

1. V0.01
2. V0.1
3. V0.2

CHAPTER 3

Change log

Outlines top level talking points of each major version change.

V0.1

- Initial functionality
- (tag: V0.1) Added some basic docs

V0.2

- (tag: V0.1) Added some basic docs
- Cleaned up formatting on README.rst
- Updated git submodule listings
- Implemented command line argument
- Updated docs for script
- (tag: V0.2) Implemented pip safe setuputils install

V0.3

- (tag: V0.2) Implemented pip safe setuputils install
- Added automated install script used by wget pipe
- Made install script prettier
- Made install script remove source after install
- (tag: V0.3) Updated install instructions to specify WGET method

V0.4

- (tag: V0.3) Updated install instructions to specify WGET method
- (tag: V0.3.1) Fixed install instructions
- Made screen update logic smarter

- Made keyboard interrupt cleaner
- (tag: V0.3.2) Updated versioning
- (tag: V0.3.3) Fixed a bug with versioning
- (tag: V0.3.4) Made graph scale to terminal size
- updated readme with usage examples
- Added an authors file
- Made install script handle git init
- Scaled down total packets displayed, fixed bug
- (tag: V0.3.5) Made script actually display lost packets
- Fixed UI scaling to better fit terminals
- Added a status bar
- Updated setup and authors to 0.4
- (tag: V0.4) Merge branch 'master' of gitlab.com:EclectickMediaSolutions/pingstats
- Added classifiers for pypi
- (tag: V0.4.1) Updated install instructions for pip
- Made setup grab readme
- (tag: V0.4.2) Updated documentation, fixed packaging bugs
- (tag: V0.4.3) Updated README for legacy install instructions

V0.5

- Made install script handle git init
- (tag: V0.4) Merge branch 'master' of gitlab.com:EclectickMediaSolutions/pingstats
- Added classifiers for pypi
- (tag: V0.4.1) Updated install instructions for pip
- Made setup grab readme
- (tag: V0.4.2) Updated documentation, fixed packaging bugs
- (tag: V0.4.3) Updated README for legacy install instructions
- (tag: V0.5a1) Changed hipsterplot dependency behaviour, fits pip
- (tag: V0.5b1) Made software display additional plot with average
- Fixed some display bugs from V0.5b2
- Docs now reference new functionality, version
- (tag: V0.5b2) Updated Authors
- Updated the way pypi gets version number
- Made setup.py call pytest
- Wrote some basic tests
- Added version to middle line
- Merge branch 'tests'

- Added setup.cfg alias for pytest
- (tag: V0.5b3) Added tests instructions, updated readme image
- (tag: V0.5) Graduate V0.5b3 to V0.5, add changelog

V0.6

- (tag: V0.5) Graduate V0.5b3 to V0.5, add changelog
- Made getpings handle nt search constraints
- Added nt safe Popen mechanics, nt ping grepping
- (tag: V0.6) Updated to version 0.6
- (tag: V0.6.1) Removed errant print statements
- (tag: V0.6.1.1) Fixed versioning
- (tag: V0.6.2) Replace _get_tty with stdlib get_terminal_size

V0.7

- (tag: V0.6) Updated to version 0.6
- (tag: V0.6.1) Removed errant print statements
- (tag: V0.6.1.1) Fixed versioning
- (tag: V0.6.2) Replace _get_tty with stdlib get_terminal_size
- (tag: V0.7) Added a more sane access approach

V0.8

- (tag: V0.7) Added a more sane access approach
- (tag: V0.8) Improves API structure, adds minor features
- (tag: V0.8.1) Re-implement list constructors, rename args

V0.9

- (tag: V0.8.1) Re-implement list constructors, rename args
- (tag: V0.8.1.1) Fixed distultils binary pointer
- Made pingstats.get_pings modular
- Added tests for new pings object
- Updated global scale values to better scale ui
- Made pingstats.Pings object data names sane
- Re-implemented pingstats.get_pings
- Added deprecated calls to pingstats._run
- Finished implementing tests f/ logic
- (tag: V0.9b1) Re-implemented pingstats._run for legacy support
- Added tests f/ UI
- Made pingstats.get_pings more clear
- Updated documentation f/ project
- Added new docs via sphinx

- Removed old docs
- Updated changelog and authors f/ V0.9
- (tag: V0.9b2) Added dynamic ui layout manipulation
- Updated version tag
- Merge remote-tracking branch 'internal/0.9b3-docs'
- Added a basic readme back to the project
- (tag: V0.9b3) Fixed readme naming. . . .
- Fixed Justyn Chaykowski author credit
- (tag: V0.9b3.1) Various bugfixes, display fixes
- Added git hooks and configured repo hooks path
- Added a setup script for git hooks
- Removed old setup.sh
- Updated docs with hook install information
- Updated base readme
- Moved git hooks to more visible location
- (tag: V0.9b3.2) Sanitizes plot/widget sizing.
- Sanitizes plot/widget sizing.
- (tag: V0.9b4) Fix versioning
- Added status line, displays most recent raw data
- Adds tests for commit c93afd9
- Fixes scaling for windows environments
- Added a dropped packets status widget
- Extended DroppedStatus with docs and more functionality
- Added tests for DroppedStatus
- Adds documentation to RawStatus
- Updated tests to be more NT aware
- Merge remote-tracking branch 'internal/raw_status'
- Merge remote-tracking branch 'internal/dropped_status'
- (tag: V0.9b5) Added new status widgets, fixes windows scaling
- Updated docs to represent new changes
- Removed deprecated code (was deprecated in V0.8)
- Cleaned build directory for readthedocs
- Removed sub module
- (tag: V0.9) Merge remote-tracking branches 'internal/0.9-updated-docs' and 'internal/cleanup-deprecated'
- (tag: V0.9.1) Fix versioning. . . again
- Made widget object names more sane.

- Reimplement clear on ui loop
- Merge branches 'fix-ui-naming' and 'reimplement-clear' into rolling-release
- (tag: V0.9.2b1) Updated to V0.9.2b1 on rolling release
- Prune references to git tasks
- Merge branch 'prune_git_tasks' into rolling-release
- Cleans up changelog, adds changelog generation script
- Merge branch 'prune-changelog' into rolling-release
- Added example output to all status widget docstrings
- Merge branch 'widget-example-docstrings' into rolling-release
- Updated module autodocs to be more legible/sensible
- Updated Readme software description
- Removed scripts for old git tasks system
- Merge branches 'update_automodule', 'update_readme_description' and 'update-hooks' into rolling-release
- (tag: V0.9.2b2) Update to V0.9.2b2
- (tag: V0.9.2, origin/master, master) Update version to reflect V0.9.2b2 passing all tests
- Added config parsing capabilities via config.py
- Added config API to __init__.py
- Fixed a bug for nt
- Added some basic docs for config.py
- Added config.py docs to sphinx-docs
- Removed default docs theme to enable Readthedocs defaults
- Merge remote-tracking branch 'origin/user-preference-file' into rolling-release
- Added install instructions for installing from source
- Added an interface to provide ui.Runner with user derived widgets
- Remove reference to nix only installs
- Added long-description back to setup.py
- Added command line arguments to README
- Added a "widgets" property to ui.Runner to make things cleaner
- Updated tests to remove warnings from pytest
- (tag: V0.9.3) Added basic tests to ui.Runner
- Update versioning
- (tag: V0.9.3.1) Ensured Runner executed from setuputils
- Updated version
- Updated contribution guidelines to outline git bug workflow
- Fixed argv pass to run function
- Added tests for runtime and parser

- Added test to ensure path is set properly f/ console scripts
- (tag: V0.9.4) Moved custom_widget snippet to example file
- Pruned old hook files f/ git, unused and stale.
- Updated the contribution guidelines
- Made contribution guidelines more clear
- Updated docs to fix klaus-viewer rendering issues
- Added issue tracker to README.rst
- Updated contribution guidelines in sphinx-docs
- Updated primary repo url throughout docs
- Updated primary repo url throughout docs
- Merge remote-tracking branch 'origin/googlegroups-bug-tracker'
- (origin/master, origin/HEAD) Fixed a hyperlink error in README.rst
- Added import try/except block to handle missing gnuplot
- Added multi-backend render functions
- Makes UI Plot widgets intelligently choose backend
- Update test_ui.py
- Updated tests to reflect 739e898
- Moves a vim modeline to the bottom of the file
- Updated README to reflect both backend's install methods
- Added asciiplotlib to requirements.txt
- Updates sphinx-docs for V1.0, multi-backend support
- Removed old screenshot, linked to demo
- (tag: V1.0.0) Added link to readthedocs

CHAPTER 4

Modules

4.1 pingstats.__init__

Example Usage:

```
>>> import pingstats
>>> pings = pingstats.Pings('google.ca')
>>> for ping in pings:
...     pingstats.plot_pings(pings.realtime_data)
# HIPSTERPLOT OUTPUT
...     print(ping)
# The most recent return time in milliseconds
```

class pingstats.**Pings**(*address*, *realtime_data=[]*, *average_data=[]*, *realtime_data_length=20*, *average_data_length=200*)

Main logic object for obtaining ping data.

Example Usage:

```
>>> from pingstats import Pings
>>> pings = Pings('127.0.0.1')
>>> for this_time in pings:
...     print(pings.realtime_data)
...     print(pings.average_data)
...     print(pings.current_line)
...     print(this_time)  # current ping's realtime data in milliseconds
```

__init__(*address*, *realtime_data=[]*, *average_data=[]*, *realtime_data_length=20*, *average_data_length=200*)

Instantiates data used during the ping retrieval process.

Address The address to send ICMP ECHO requests to.

Realtime_data A python list object to continue to append data to.

Average_data A python list object to continue to append data to.

Realtime_data_length The maximum size of the realtime_data list

Average_data_length The maximum size of the average_data list

`pingstats.get_pings`(*address*)

Yields the last set of realtime and average data points, according to user specified maximum.

Address The address to send `ICMP ECHO` requests to.

`pingstats.plot_pings`(*pings*, *columns=70*, *rows=15*)

Provides high level bindings to `hipsterplot.plot` for use with *`pingstats.get_pings()`*.

Pings A list object containing ping return times as float values, normally from *`pingstats.get_pings()`*

Columns The number of columns to draw for the plot

Rows The number of rows to draw for the plot

4.2 pingstats.ui

Outlines UI behaviour.

UI is presented by outputting the same number of columns and rows as the connected tty.

The UI is constructed of several `pingstats.ui.PingsWidget` derivative classes. Each of these classes outputs different types of information to the display. Each type of derivative class has similar output:

Where objects with the following in their names have the listed effect:

- `Status`
  - A single line display of simple information. For example, the number of packets being displayed for a given measure (*`pingstats.ui.RealtimeStatus`* or `pingstats.ui.AverageStatusc`) or a simple display of ping target and program version.
- `Plot`
  - A plot based on `hipsterplot` plotting functionality. Each plot displays particular information (*`pingstats.ui.RealtimePlot`* displays a plot of realtime ping data, *`pingstats.ui.AveragePlot`* displays a plot of the average of those realtime data points.
- `Pane`
  - Some combination of `Status` and `Plot` widgets.

The `pingstats.ui.Runner` class provides functionality for dynamically executing the UI. For example, given the command line argument `-l realtimepane,progstatus,averagepane` produces the default display of the software.

—

4.2.1 Base Widgets

class `pingstats.ui.Widget`(*columns*, *rows*)

Base class for all UI elements. Defines columns and rows.

`__init__`(*columns*, *rows*)

Instantiates self.columns and self.rows

Columns The total number of columns available for display.

Rows The total number of rows available for display.

class pingstats.ui.**PingWidget**(*columns*, *rows*, *pings_object*)
Adds realtime and average data objects to self

__init__(*columns*, *rows*, *pings_object*)
Instantiate self.columns, self.rows, self.realtime_data, self.average_data

Columns The total number of columns available for display.

Rows The total number of rows available for display.

Pings_object An instance of `pingstats.Pings()`

4.2.2 Status Widgets

class pingstats.ui.**RawStatus**(**args*)
Displays the current data line from the original call to ping.

Example output:

```
>>> -----------64 bytes from 127.0.0.1: icmp_seq=1 ttl=128 time=0.115 ms------
↪------
```

class pingstats.ui.**AverageStatus**(**args*)
Displays information about average data, and if the connection was dropped.

Example output:

```
>>> "Displaying the average of 12 total packets from 127.0.0.1"
                                - OR -
>>> "Connection dropped!"
```

class pingstats.ui.**DroppedStatus**(**args*)
Displays how many packets have been dropped if any, otherwise the program name.

Example Output:

```
>>> -----------64 bytes from 127.0.0.1: icmp_seq=1 ttl=128 time=0.115 ms--
↪----------
```

- or -

```
>>> -----------64 bytes from 127.0.0.1: icmp_seq=1 ttl=128 time=0.115 ms--
↪----------
```

class pingstats.ui.**ProgStatus**(**args*)
Displays Program information, and a separator bar.

Example output:

```
>>> "pingstats V0.9 ------------------------ ping data from 127.0.0.1
```

class pingstats.ui.**RealtimeStatus**(**args*)
Displays information about real-time data, and packets dropped.

Example output:

```
>>> "Displaying 12 packets from 127.0.0.1"
                                - OR -
>>> "3 packets dropped of 12"
```

4.2.3 Plot Widgets

class pingstats.ui.**AveragePlot**(**args*)
Displays average data on a plot.

class pingstats.ui.**RealtimePlot**(**args*)
Displays real-time data on a plot.

4.2.4 Pane Widgets

class pingstats.ui.**AveragePane**(*columns*, *rows*, *pings*)
Displays average data on a plot, along with the output of `pingstats.ui.AverageStatus()`.

class pingstats.ui.**RealtimePane**(*columns*, *rows*, *pings*)
Displays real-time data on a plot, along with the output of `pingstats.ui.RealtimeStatus()`.

4.2.5 Creating your own Widget

To create your own widget, simply inherit from the topmost relevant widget class, override the init class, and then define your output to the `stdout`. For example:

```
from pingstats.ui import Runner, PingWidget  # Import resources


class MyStatus(PingWidget):  # Define our widget
    def __init__(self, *args):  # Overide init, collecting any args
        super(MyStatus, self).__init__(*args)  # Pass args to the original function
        print('hello world')  # Perform output


if __name__ == '__main__':
    # instantiate runner, pass a list containing our Widget and the equivalent of sys.
↪argv
    runner = Runner([MyStatus], ['-l', 'droppedstatus,mystatus,realtimestatus',
↪'google.ca'])
    runner()  # Run logic loop
```

Warning: Arguments are passed to the widget classes internally. You can mirror the arguments of the parent class init function, or pass `*args`

Note: The runner class uses the name of the widget to determine how much space each should take on screen, where status widgets take one line and plot/pane widgets take *n* lines. Therefore, you should specify what kind of widget it is in the object name.

4.3 User Configuration

Provides ease of use configuration capabilities to the project.

Parses files with arguments for the project split line by line, such that:

-l droppedstatus,averagepane,rawstatus

Would be appended onto the end of sys.argv.

Warning: The config files do not accept the setting of targets

For *Nix users, the project searches for configuration files under the following paths:

```
~/.pingstats.rc
```

For NT users, the project searches for configuration files under the following paths:

```
%LOCALAPPDATA%\Pingstats\pingstats.rc
```

CHAPTER 5

MIT License

Copyright (c) 2017 Ariana Giroux

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

pingstats, 13
pingstats.config, 17
pingstats.ui, 14

Index

Symbols

`__init__()` (*pingstats.Pings method*), 13
`__init__()` (*pingstats.ui.PingWidget method*), 15
`__init__()` (*pingstats.ui.Widget method*), 14

A

`AveragePane` (*class in pingstats.ui*), 16
`AveragePlot` (*class in pingstats.ui*), 16
`AverageStatus` (*class in pingstats.ui*), 15

D

`DroppedStatus` (*class in pingstats.ui*), 15

G

`get_pings()` (*in module pingstats*), 14

P

`Pings` (*class in pingstats*), 13
`pingstats` (*module*), 13
`pingstats.config` (*module*), 17
`pingstats.ui` (*module*), 14
`PingWidget` (*class in pingstats.ui*), 15
`plot_pings()` (*in module pingstats*), 14
`ProgStatus` (*class in pingstats.ui*), 15

R

`RawStatus` (*class in pingstats.ui*), 15
`RealtimePane` (*class in pingstats.ui*), 16
`RealtimePlot` (*class in pingstats.ui*), 16
`RealtimeStatus` (*class in pingstats.ui*), 15

W

`Widget` (*class in pingstats.ui*), 14