

What is Pywr, why was it written and what benefits does it have?

What is Pywr?

Pywr is a tool for solving network resource allocation problems.

- It is written in the Python programming language.

Pywr's principle design goals are that it is:

- Fast enough for use in advanced decision making methodologies;
- Free to use without restriction; and
- Extendable.

Origins

- Started 2014 by Joshua Arnott
 - An exercise in learning and curiosity to understand how water resource models behave.
 - Developed in to a generalised library for water resource modelling.

Pywr is a generalised network resource allocation model written in Python.

hydrology

water-resources

1 commit

44 branches

6 releases

7 contributors

GPL-3.0

Tree: a0f6a96fe0

New pull request

Find file

Clone or download



snorfalorpagus Initial commit.

Latest commit a0f6a96 on 14 Dec 2014

.gitignore

Initial commit.

4 years ago

pywr.py

Initial commit.

4 years ago

test.py

Initial commit.

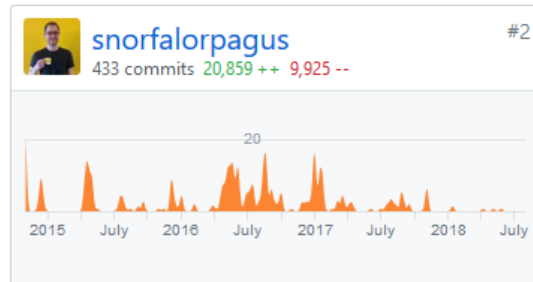
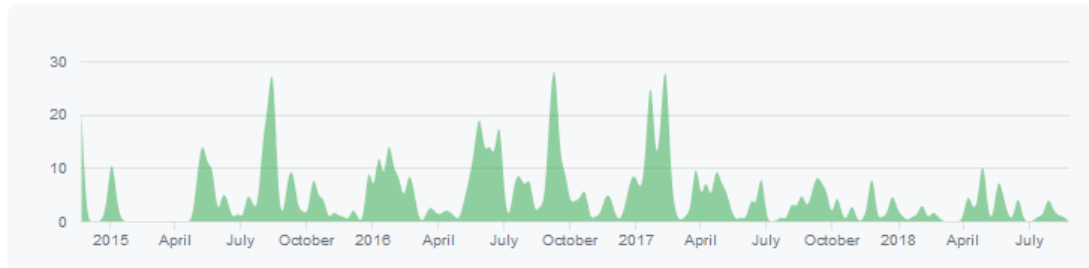
4 years ago

Today

Dec 14, 2014 – Sep 18, 2018

Contributions: Commits ▼

Contributions to master, excluding merge commits



- Development continued
 - Now over 1100 commits & 7 different contributors.
 - A regression test suite with over 300 unit tests.
 - Use in PR19 for several water companies.

How does it work?

- Pywr provides a way for modellers to :
 - Define a network of nodes and links;
 - Assign dynamic values to constraints and costs of transporting resource around the network;
 - Execute a simulation of many time-steps each computing an optimisation based on allocation of resource across the network;
 - Track and save useful information and model state during a simulation.

Software and technology

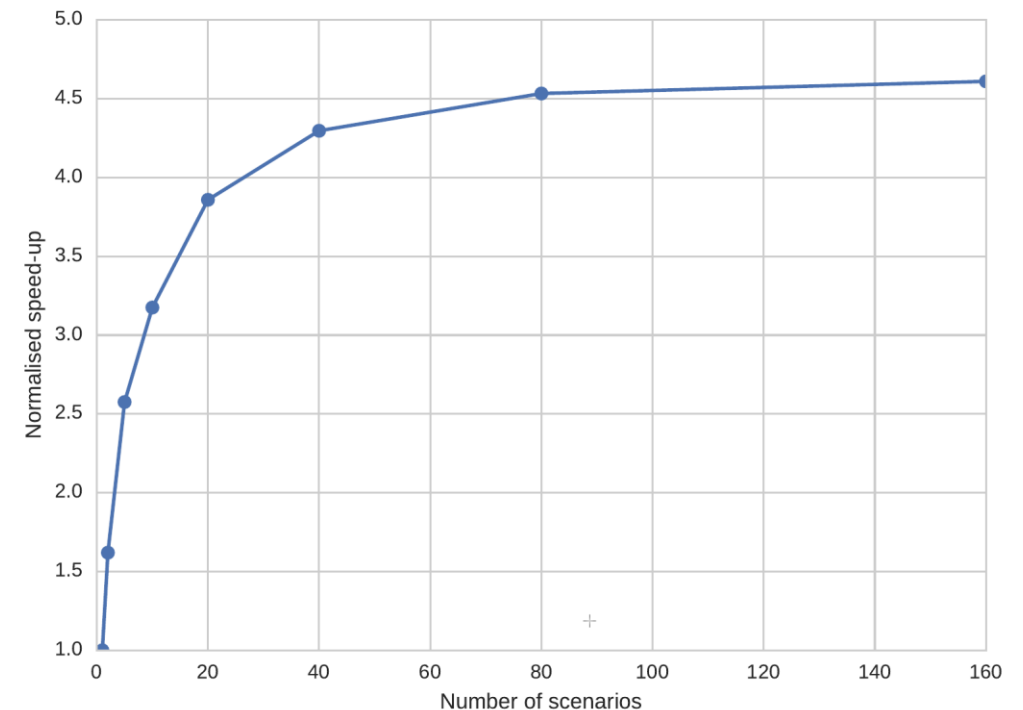
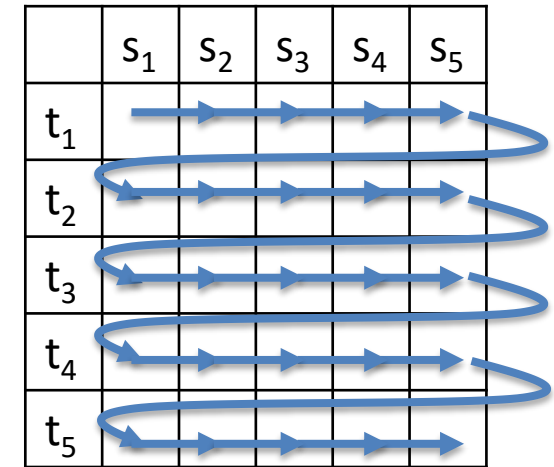
- Core technology is Python
 - Pywr is a Python library (or package)
 - Supported by the Cython static compiler for generating C extensions.
- 3rd party Python libraries, including:
 - Numpy, Scipy, Pandas, Matplotlib, Jupyter, Pytables
- GLPK as the linear programme solver
- A JSON file format is supported for defining models.

Fast

- Extensive use of Cython
 - Provides C extensions for Python.
 - All critical simulation code paths are “cythonised”.
- Direct interface to the GLPK library
 - Avoids overhead of a math programming library (e.g. Pyomo, GAMS)
- Efficient dependency tree for model components
 - Ensures evaluation only once per time-step.

Scenarios & RDM

- Built-in support for running multiple sets of scenario ensembles
 - E.g. 10 demand scenarios with 25 climate scenarios → 250 scenarios
- Execution model exploits commonality amongst scenarios.
- Tracked metrics can use different aggregation methods for the computation of robustness across scenarios.



Free

- Free to use without restriction;
- Licensed under the GNU General Public Licence;
- "Think free as in free speech, not free beer."
- *Disclaimer: I am not a lawyer*



Are in the input & output of a model covered by the GPL?

- Copyright of the output derives from the input.
 - Same principle as Microsoft Word/Excel and other software.
- If a user simply runs their data through Pywr the input and output copyright are retained by the user.
 - There is no requirement to distribute / release the input or output data.
- Extensions to Pywr would be considered derived works and *distributable* under the GPL.

Benefits of Free software

- Community of users (academic and commercial) and contributions;
- Everyone benefits from others' changes;
 - Enabling on-going innovation.
- Freedom to modify and study the code;
- No vendor lock-in;

Flexible & extendable

- Generic “parameter” system for defining complex dynamic behavior.
 - Multiple layers and dependencies across the model.
- Users can write new and custom functionality.
 - Standard Python can be used for rapid development and prototyping.
 - Custom code can be managed outside of Pywr as required.
 - Later cythonised if performance critical.
- Python can interact with many other languages, APIs and datasets
 - E.g. FORTRAN, NetCDF
 - Allows existing sub-models (e.g. rainfall runoff) to be linked to Pywr easily.

Deployable

- Free software, cross platform, decoupled from any UI
 - Can be deployed (installed and run) anywhere.
 - Has already been used on the Amazon EC2 cloud.
 - Can scale for large scenarios and/or optimisation studies.
- Docker containers packaging Pywr are easy to create.
- No licence files, servers or other restrictions on running 1 or 1000 models simultaneously.