
Evalys Documentation

Release 4.0.2

Olivier Richard

Dec 21, 2018

Contents

1	Evalys - Overview	1
1.1	Features	1
1.2	Examples	1
1.3	Gallery	2
2	Indices and tables	5
	Python Module Index	7

“Infrastructure Performance Evaluation Toolkit”

It is a data analytics library made to load, compute, and plot data from job scheduling and resource management traces. It allows scientists and engineers to extract useful data and visualize it interactively or in an exported file.

- Free software: BSD license
- Documentation: <https://evalys.readthedocs.org>.

1.1 Features

- Load and all [Batsim](#) outputs files
 - Compute and plot free slots
 - Simple Gantt visualisation
 - Compute utilisation / queue
 - Compute fragmentation
 - Plot energy and machine state
- Load SWF workload files from [Parallel Workloads Archive](#)
 - Compute standard scheduling metrics
 - Show job details
 - Extract periods with a given mean utilisation

1.2 Examples

You can get a simple example directly by running `ipython` and discover the evalys interface. For example:

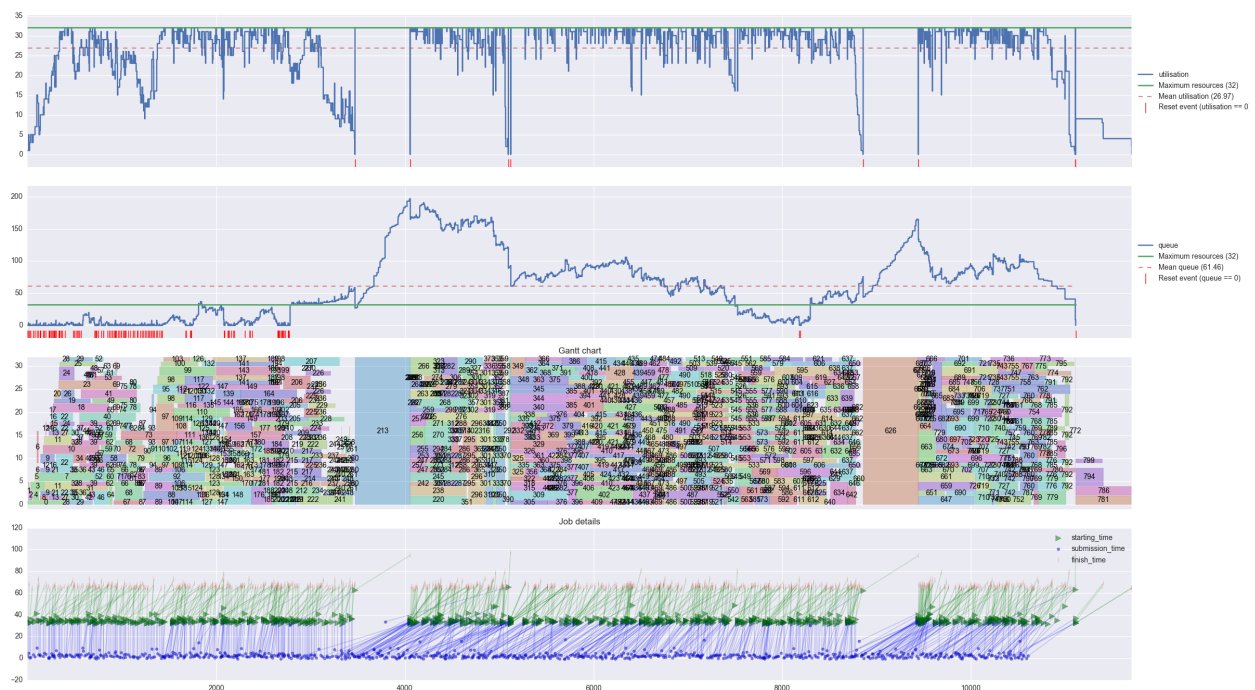
```
from evalys.jobset import JobSet
import matplotlib.pyplot as plt

js = JobSet.from_csv("evalys/examples/jobs.csv")
js.plot(with_details=True)
plt.show()
```

This also works for SWF files but the Gantt chart is not provided because job placement information is not provided in this format.

You can find a lot of examples in the `/examples` directory.

1.3 Gallery



Contents:

1.3.1 Installation

You can install, upgrade, uninstall `evalys` with these commands:

```
pip install [--user] evalys
pip install [--user] --upgrade evalys
pip uninstall evalys
```

Or from git (last development version):

```
pip install git+https://github.com/oar-team/evalys.git
```

Or if you already pulled the sources:

```
pip install path/to/sources
```

Or if you don't have pip:

```
easy_install evalys
```

1.3.2 Evalys module documentation

Workload: Handle Feitelson's SWF

JobSet: Handle Batsim output file

Visualisation library

Metrics computation

`evalys.metrics.cumulative_waiting_time(dataframe)`

Compute the cumulative waiting time on the given dataframe

Dataframe a DataFrame that contains a "starting_time" and a "waiting_time" column.

`evalys.metrics.compute_load(dataframe, col_begin, col_end, col_cumsum, begin_time=0, end_time=None)`

Compute the load of the *col_cumsum* columns between events from *col_begin* to *col_end*. In practice it is used to compute the queue load and the cluster load (utilisation).

Returns a load dataframe of all events indexed by time with a *load* and an *area* column.

`evalys.metrics.load_mean(df, begin=None, end=None)`

Compute the mean load area from begin to end.

`evalys.metrics.fragmentation(free_resources_gaps, p=2)`

Input is a resource indexed list where each element is a numpy array of free slots.

This metrics definition comes from Gher and Shneider CCGRID 2009.

Utilities

`evalys.utils.bulksetattr(obj, **kwargs)`

Safely assign attributes in bulk.

For each keyword argument kw, the function checks that kw is the name of one of the object's attributes. If kw is not the name of an attribute, the function raises an `AttributeError`. Otherwise, the function assigns the value of the keyword argument to the attribute, provided the object allows it.

`evalys.utils.cut_workload(workload_df, begin_time, end_time)`

Extract any workload dataframe between *begin_time* and *end_time*. Datafram must contain 'submission_time', 'waiting_time' and 'execution_time' + 'jobID' columns.

Jobs that are queued (submitted but not running) before *begin_time* and jobs that are running before *begin_time* and/or after *end_time* are cut to fit in this time slice.

Example with `evalys.Workload`:

```
>>> from evalys.workload import Workload
>>> w = Workload.from_csv("./examples/UniLu-Gaia-2014-2.swf")
>>> cut_w = cut_workload(w.df, 500000, 600000)
```

Example with `evalys.JobSet`:

```
>>> from evalys.jobset import JobSet
>>> js = JobSet.from_csv("./examples/jobs.csv")
>>> cut_js = cut_workload(js.df, 1000, 2000)
```

1.3.3 Credits

- Olivier Richard <olivier.richard@imag.fr>
- Michael Mercier <michael.mercier@inria.fr>
- Millian Poquet <millian.poquet@inria.fr>
- Raphaël Bleuse <raphael.bleuse@uni.lu>
- Valentin Reis <valentin.reis@inria.fr>
- Steffen Lackner <lackner@cs.tu-darmstadt.de>

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

e

`evalys.metrics`, 3

`evalys.utils`, 3

B

`bulksetattr()` (in module `evalys.utils`), 3

C

`compute_load()` (in module `evalys.metrics`), 3

`cumulative_waiting_time()` (in module `evalys.metrics`), 3

`cut_workload()` (in module `evalys.utils`), 3

E

`evalys.metrics` (module), 3

`evalys.utils` (module), 3

F

`fragmentation()` (in module `evalys.metrics`), 3

L

`load_mean()` (in module `evalys.metrics`), 3