
dsat Documentation

Release stable

January 19, 2015

1	References	1
2	Content	3
2.1	Status handling	3
2.2	State Machine	3
2.3	Messaging	3
2.4	Topological connection settings	3
3	Version	5
4	Changes	7
4.1	0.5.0	7

References

DSat is a comprehensive distributed solution made of satellites communicating with a messaging system.

The basic:

- a process is a directed graph with alternates fixed points (routers) and moving parts (workers);
- the output of a worker are the input of the next worker;
- the satellite set of worker/processor does the monitoring of the load of the pool of workers process and propagate messages outside the satellite;
- the Status of a pipelined task can be followed either dynamically (cf status_for) or statically by requesting the State(s) object
- the workers are encapsulated in a state_machine wrapper ensuring the reporting of the finite state machine corresponding to the task
- the solution provides native load balancing with round robin

Content

2.1 Status handling

Status are stored in a backend that ensures we are threadsafe and that we have serialization.

2.2 State Machine

A simple wrapper around a process that provides communication around the execution and (should) handle time outs.

2.3 Messaging

normally not needed outside of satlive (except parse_event)

Ensures a consistency between message emission and reception

2.4 Topological connection settings

state_machine.get_connection set the connections of your step magically according to the imperative of the graph.
Because, a distributed system is mainly processes connected in a directed graph.

Version

This a **safe** subset of <http://www.python.org/dev/peps/pep-0440/#version-scheme> Given it is broken for now I use a safe subset : Given a version $X.Y.Z$ where X Y Z are strings by convention X and Y are numbers Z is alphanumeric lower case default “0”

- X is the major version number. $X+=1 \Leftrightarrow$ big change in philosophy API
- Y is the minor version number. $Y+=1 \Leftrightarrow$ API has changed (ascendant compatibility is broken)
- Z is a bugfix/feature addition: it shall not break any program already using the API

$V1 > V2$ should be true has much as in number as in alphanumeric sens.

- Z belongs to $[a-z0-9]$
- X and Y belongs to $[0-9]^+$

Yes we are limited to 36 versions per API if needed and you must extends Z remember to keep string and number sorting consistent.

Changes

4.1 0.5.0

As perfect as an alpha can be :)