



Virtual Earthquake and seismology Research Community e-science environment in Europe  
Project 283543 – FP7-INFRASTRUCTURES-2011-2 – [www.verce.eu](http://www.verce.eu) – [info@verce.eu](mailto:info@verce.eu)



# The VERCE Science Gateway: Installing & Running dispel4py

(dispel4py training)

15-16 October 2014



# Outline

- Install dispel4py
- Dependencies
- Testing the installation
  - Sequential Mode
  - Parallel Mode
    - MPI
    - Multi-process
- Command line arguments

# Installing dispel4py

- Download the source and add it to your python path (PYTHON\_PATH):

```
git clone https://github.com/akrause2014/dispel4py.git
```

- or install in your python environment by using pip:

```
pip install git+git://github.com/akrause2014/dispel4py.git#egg=dispel4py
```

# Dependencies

- Dispel4Py has been tested with Python versions **2.7.5**, **2.7.2** and **2.6.6**.
- The following Python packages are required to run Dispel4Py:
  - **networkx** (<https://networkx.github.io/>)
- If using the MPI mapping:
  - Openmpi ( or MPICH2 ) :
    - Instructions: [http://www.itp.phys.ethz.ch/education/hs12/programming\\_techniques/openmpi.pdf](http://www.itp.phys.ethz.ch/education/hs12/programming_techniques/openmpi.pdf)
  - mpi4py (<http://mpi4py.scipy.org/>):
    - Instructions: `sudo easy_install mpi4py`
- If using the Storm mapping:
  - Python Storm module, available here:
    - <https://github.com/apache/incubator-storm/tree/master/storm-core/src/multilang/py>, to be placed in directory resources.
  - Python Storm thrift generated code, available here:
    - <https://github.com/apache/incubator-storm/tree/master/storm-core/src/py>

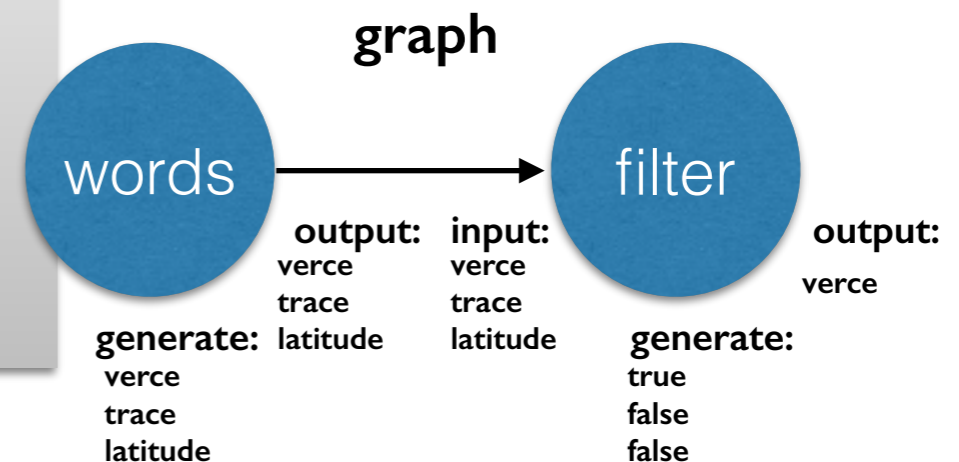
# Testing the installation

- **Create a 'filter\_graph.py' script:** The following script creates a dispel4py workflow called 'graph' which connects 2 Processing elements (PEs): 'words' and 'filter' objects. The first PE produces words and streams them to the 'filter' PE. The 'filter' PE generates randomly 'true' or 'false' values. Only when the 'filter' PE generates 'true', the word created by 'words' PE, is streamed out.

```
from dispel4py.workflow_graph import
WorkflowGraph
from dispel4py.examples.graph_testing.testing_PEs
import RandomWordProducer, RandomFilter
```

```
words = RandomWordProducer()
filter = RandomFilter()
graph = WorkflowGraph()
graph.connect(words, 'output', filter, 'input')
```

**filter\_graph.py**



# Testing the installation

## Sequential mode

- We assume that filter\_graph.py is located in the 'parent' directory ( 'dispel4py-master' )
- Execute the workflow in sequential mode. This type of execution is recommended for testing workflows in laptops or computers with one cpu.

- 1 iteration of the workflow

```
python -m dispel4py.simple_process filter_graph
```

- 10 iterations of the workflow

```
python -m dispel4py.simple_process filter_graph - i 10
```

# Testing the installation

## Parallel mode - Multi-process

- 1 Iteration of the workflow with 4 processes by using multiprocessing

```
python -m dispel4py.multi_process -n 4 filter_graph
```

- This type of execution is recommended when the workflow is executed in a shared environment: Single computing with several CPUs ( also called 'cores' ). For example, 'Terracorrelator machine'.

# Testing the installation

## Parallel mode - MPI

- 1 Iteration of the workflow with 4 processes by using mpi

```
mpiexec -n 4 python -m dispel4py.worker_mpi filter_graph
```

- This type of execution is recommended when the workflow is executed in a distributed (several computers) or hybrid (shared and distributed) memory architecture. For example in a 'cluster' like SuperMUC.
  - **\*\* MPI \*\*** can be used for executing workflows in shared memory environments, but it is not recommended for performance reasons.



# Command line arguments

- `<-n num>`: number of processes to run
- `<-a attr >`: name of graph variable in the module
- `<-f file>`: file containing the input dataset in JSON format
- `<-d data>`: input dataset in JSON format
- `<-i iter>`: number of iterations
- `<-s>`: force simple processing