Virtual Earthquake and seismology Research Community e-science environment in Europe
Project 283543 – FP7-INFRASTRUCTURES-2011-2 – www.verce.eu – info@verce.eu

dispel4py: An Open Source Python Framework for Encoding, Mapping and Reusing Continues Data Streams

IN34A: Enabling Scientific Analysis, Data Reuse, and Open Science through Free and Open Source Software II

16 December 2015

Rosa Filgueira et al.
University of Edinburgh

Outline

- Introduction
- dispel4py features
- dispel4py basic concepts
- dispel4py advanced concepts
- VERCE project
- dispel4py workflows
- Evaluations
- Current work
- Conclusions and future work

Introduction – What it is dispel4py ?

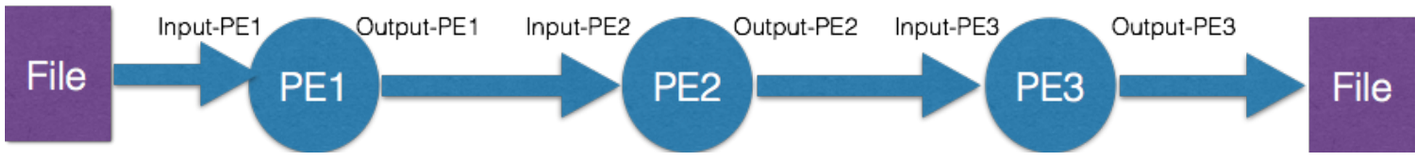
- User-friendly tool
- Develop scientific methods and applications on local machines
- Run them at scale on a wide range of computing resources without making changes

dispel4py features

- Python for describing tasks and connections
- Stream-based
  - Tasks are connected by streams
  - Multiple streams in & out
  - Optimisation based on avoiding IO
- Modular
- Multiple enactment systems

dispel4py basic concepts – Processing element

- PEs represent the basic computational unit
  - Data transformation, scientific method, service request
- PEs are the “Lego bricks” of tasks and users can assemble them into a workflow as they wish
- General PE features
  - **Consumes** any number and types of input streams
  - **Produce** any number and types of output streams

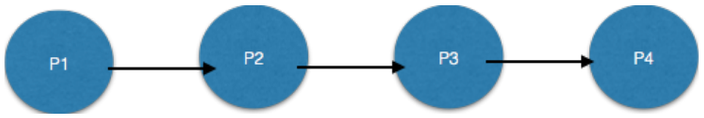


dispel4py basic concepts – Instance and graph

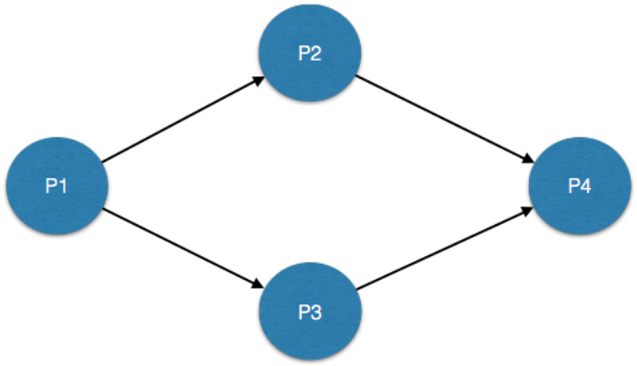
Graph

- Topology of the workflow: connections between PEs
- Users focus on the algorithm to implement or the service to use

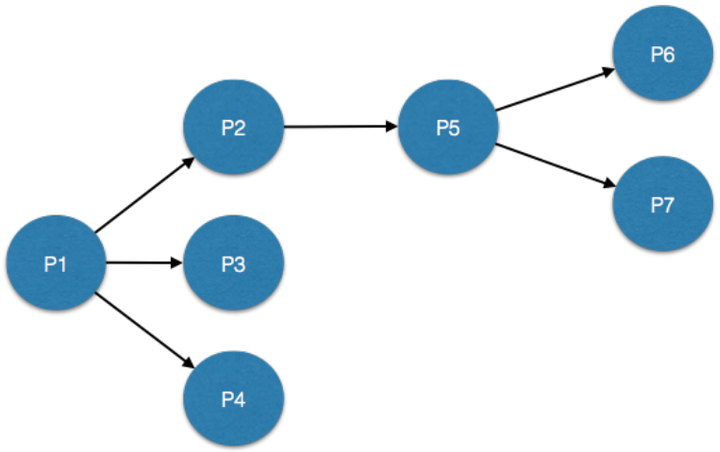
Pipeline



Split & Merge



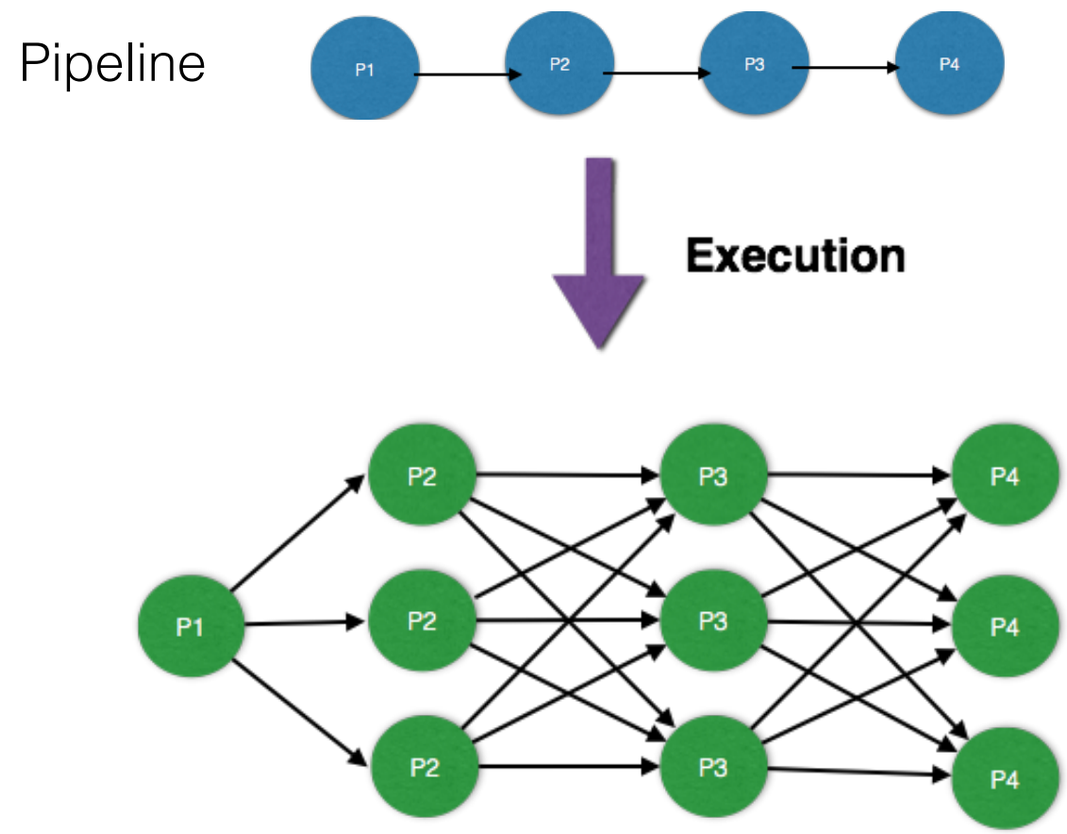
Tree



dispel4py basic concepts – Instance and graph

PE Instance

- Executable copy of a PE that runs in a process.
- Each PE is translated into one or more instances in run-time

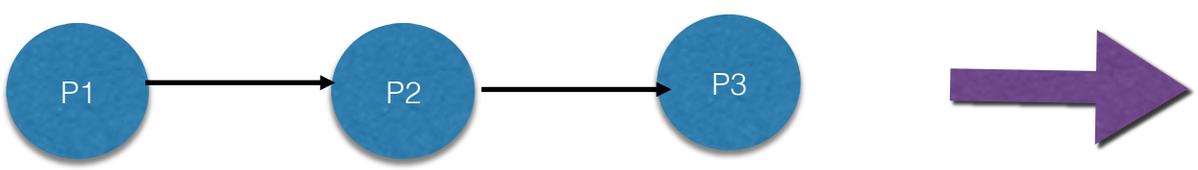


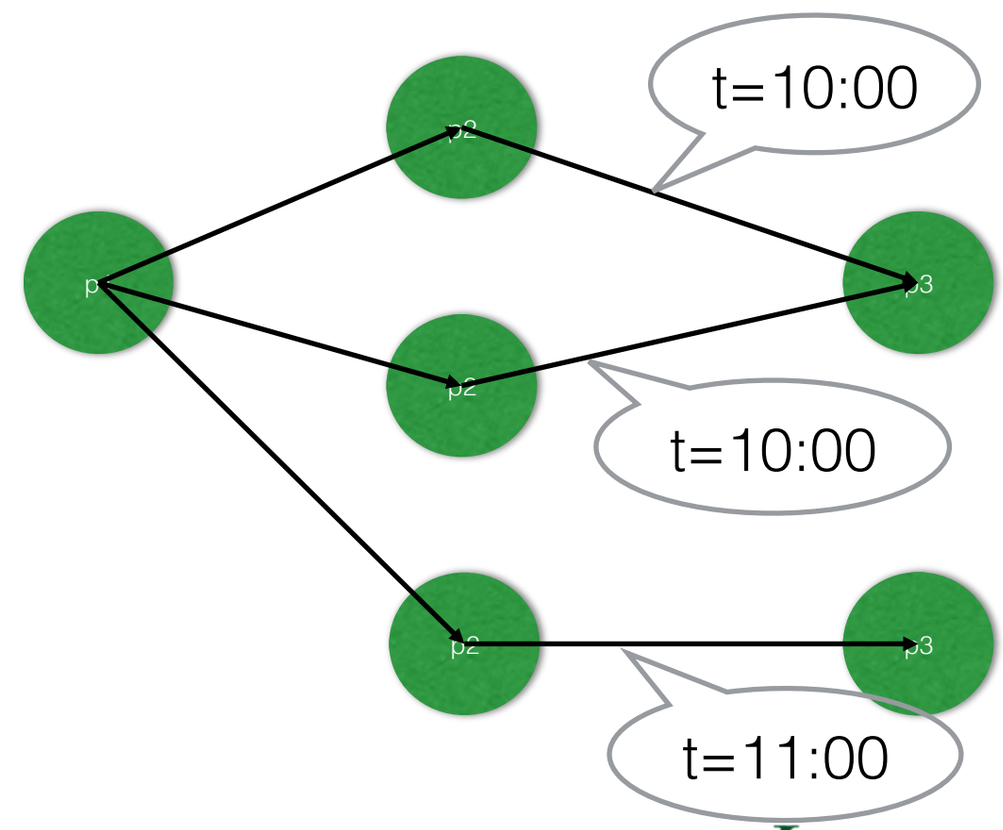
4 PEs & 10 processes

dispel4py basic concepts – Groupings

“Grouping by” a feature (MapReduce)

All data items that satisfy the same features are guaranteed to be delivered to the same **instance** of a PE



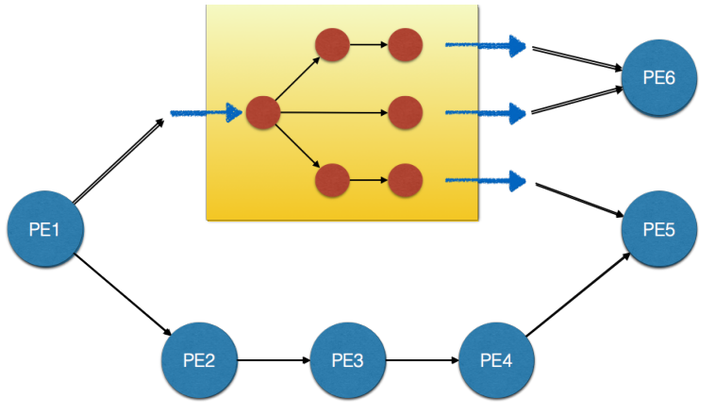


More groupings:
All-to-All, All-to-One, One-to-All

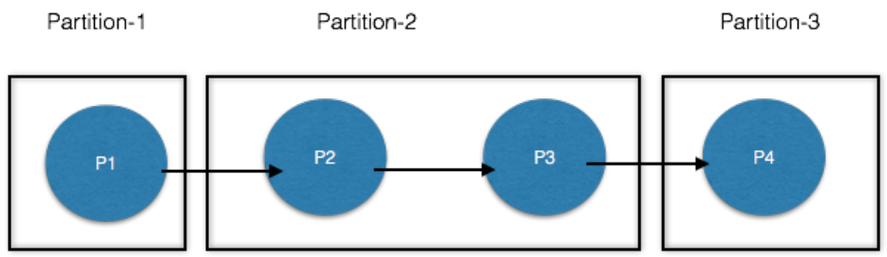
dispel4py basic concepts – Composite PE and Partition

- Composite PE
  - Sub-workflow in a PE
  - Hides the complexity of an underlying process
  - Treated like any other PE
- Partition
  - PEs wrapped together
  - Run several PEs in a single process

+ Example of Composite PE



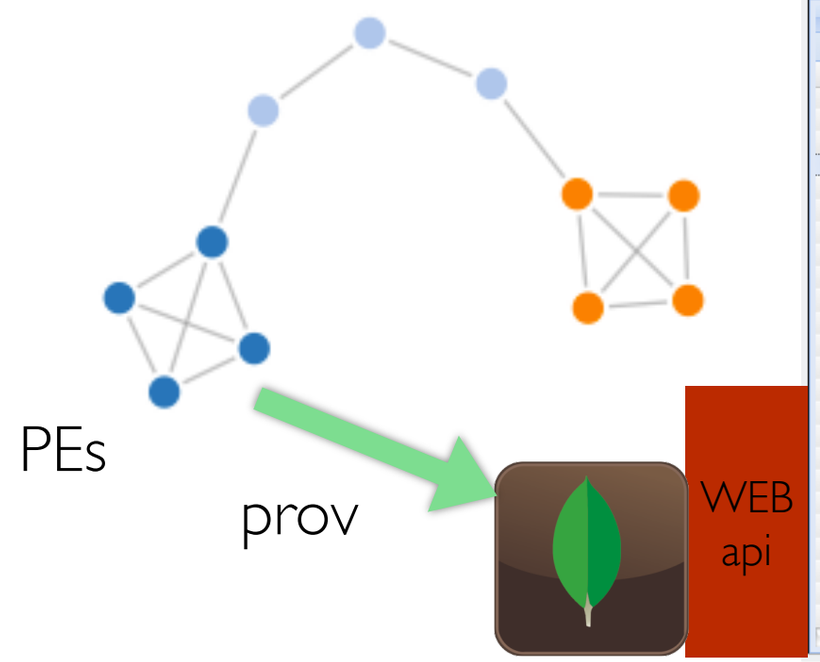
+ Example of Partition

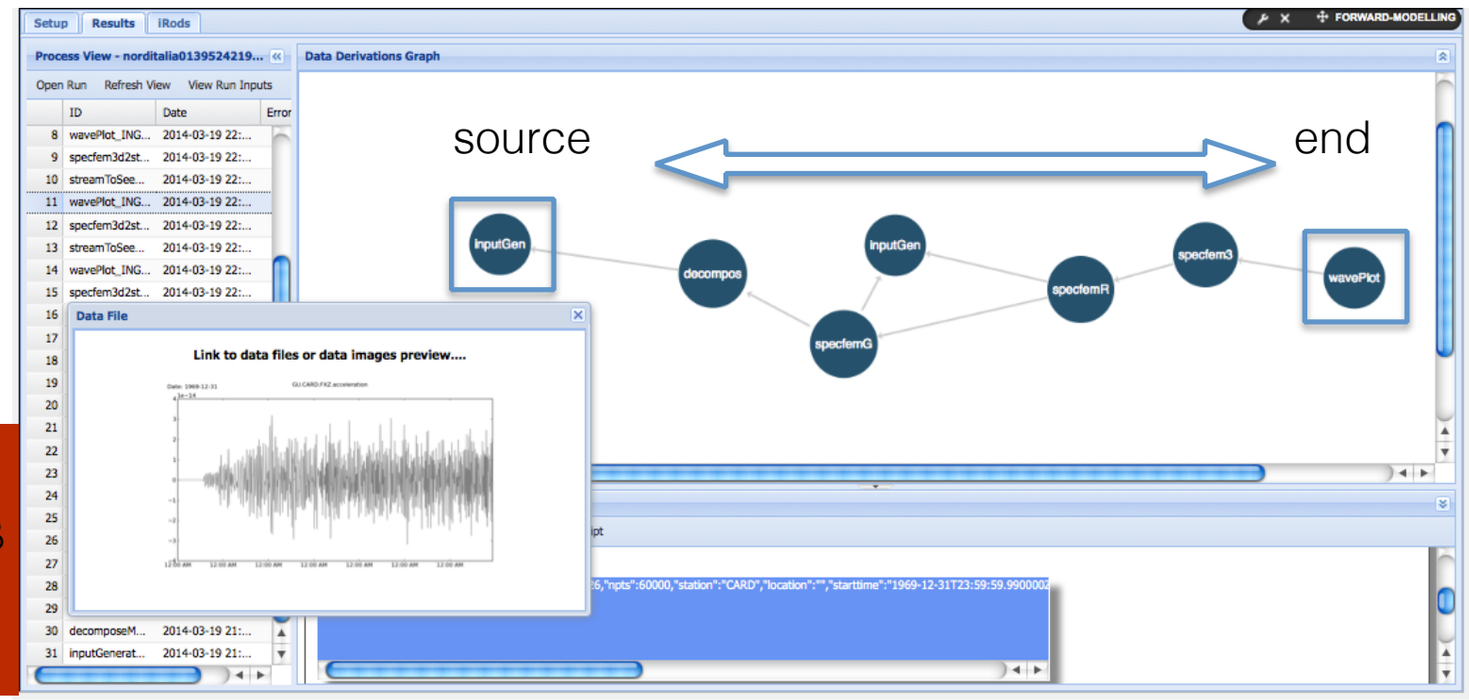


dispel4py advanced concepts – Mappings

- **Sequential**
  - Sequential mapping for local testing
  - Ideal for local resources: Laptops and Desktops
- **Multiprocessing**
  - Python's multiprocessing library
  - Ideal for shared memory resources
- **MPI**
  - Distributed Memory, message-passing parallel programming model
  - Ideal for HPC clusters
- **STORM**
  - Distributed Real-Time computation System
  - Fault-tolerant and scalable
  - Runs all the time
- **SPARK (Prototype)**
  - HDFS, Layer on top of Hadoop

dispel4py advanced concepts – Provenance





Users can select which metadata to store
Searches over products metadata within and across runs
Data download and preview
Capturing of Errors for Diagnostic purposes
Data Fabric: Multi directional navigations across data dependencies
W3C PROV-DM as reference model.

dispel4py advanced concepts – Registry

127.0.0.1

VERCE Registry Admin Welcome, **admin.** Change password / Log out

Home › Vercereg › PEs › Add PE

Add PE

Workspace: admin: SampleWorkspace

Pckg: test.package

Name: SamplePE

User: admin

Creation date: Date: 2015-03-23 Today | Time: 09:51:04 Now |

Note: You are 2 hours ahead of server time.

Description:

A sample dispel4py PE definition.
PE definitions can potentially have multiple implementations, which can be provided at a later time.

Kind: Abstract

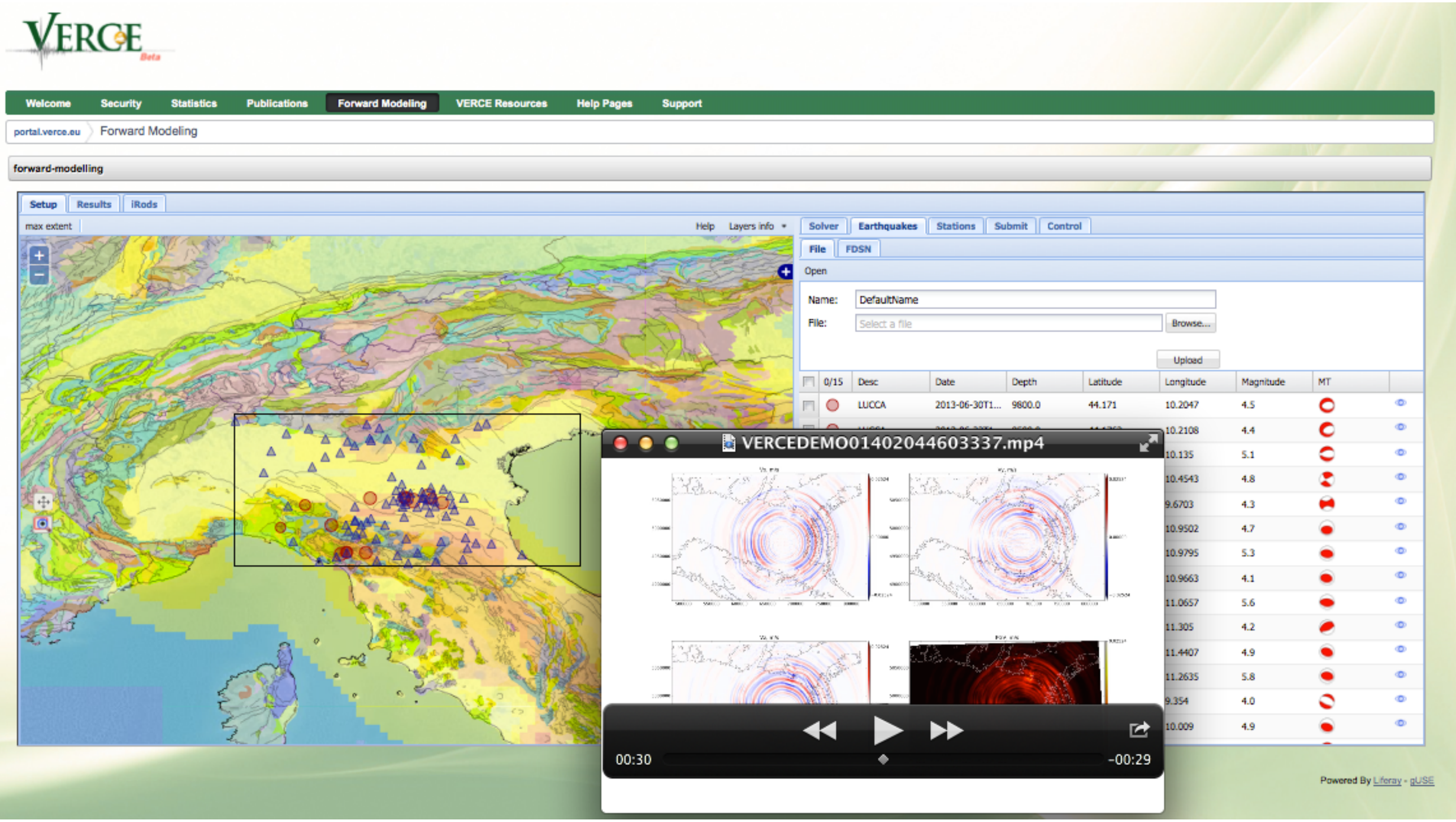
Connections

Kind	Name	S type	D type	Comment	Is array	Modifiers	Delete?
In	Input1	str			☑	modifier1:modifier2	
Out	Output1	int			☐	modifier3	

Add another Connection

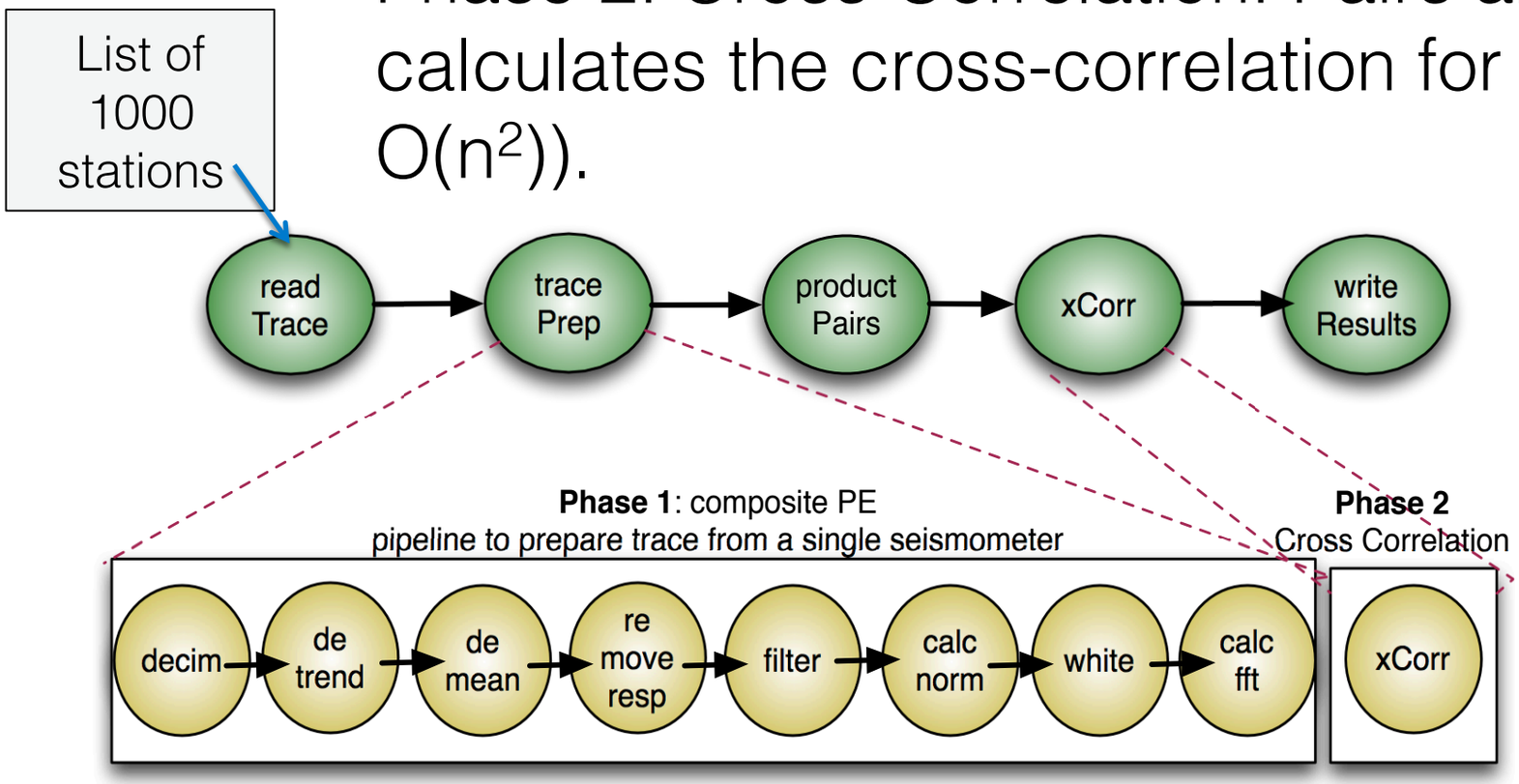
VERCE

Virtual Earthquake and Seismology Research Community in Europe



dispel4py workflows- Cross Correlation

- Data intensive problem and it is commonly used in seismology
  - Phase 1- Preprocess: Time series data (traces) from seismic stations are preprocessed in parallel
  - Phase 2: Cross-Correlation: Pairs all of the stations and calculates the cross-correlation for each pair (complexity $O(n^2)$).

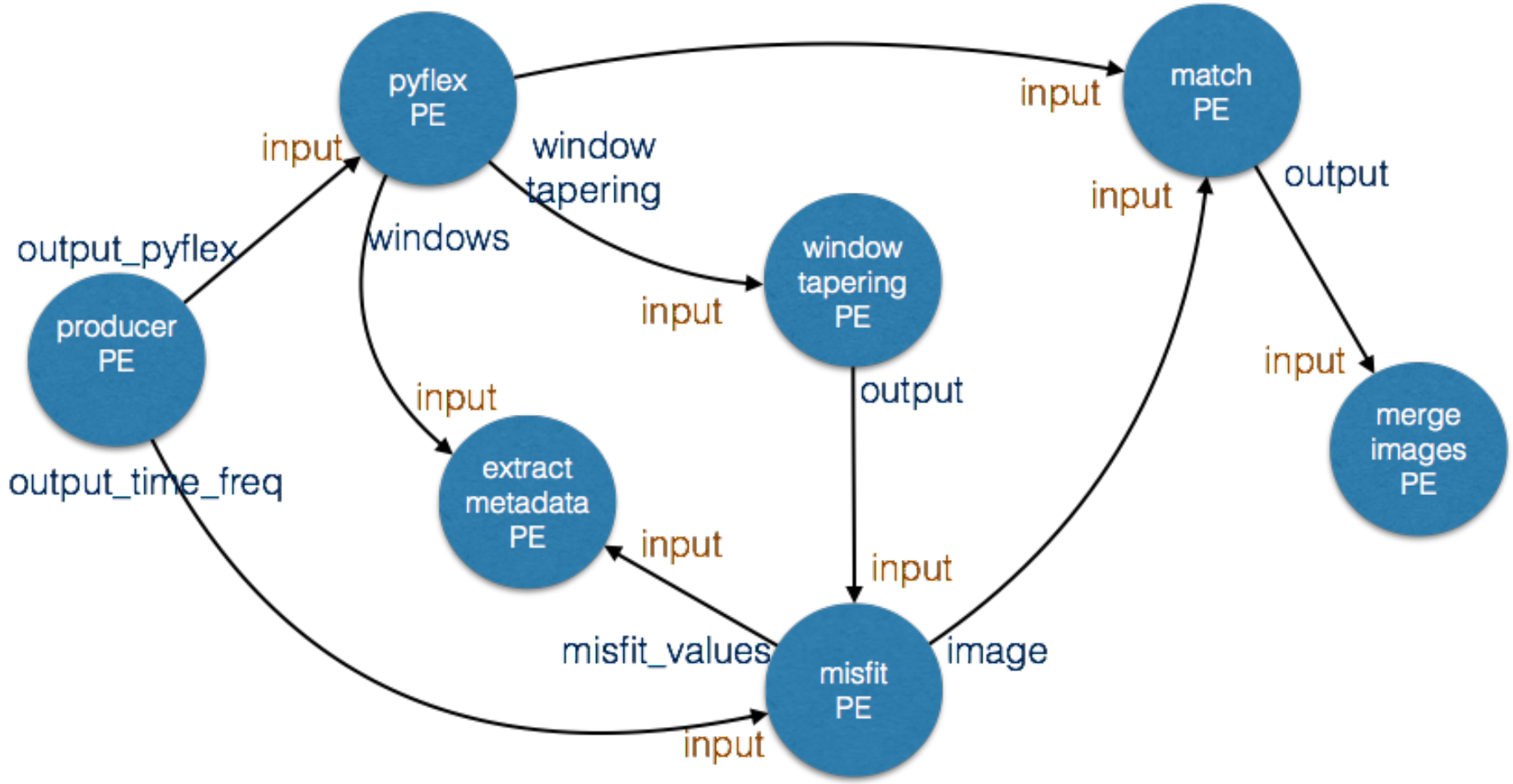


Input data:
1000 stations as input data (150MB)
Output data:
499,500 cross-correlations (39GB)

xcorr

dispel4py workflows- Misfit Computation

- Phase 1 – Preprocess: Align and prepare traces
- Phase 2 – Misfit: Compare synthetic and observed data



Evaluations – Cross Correlation

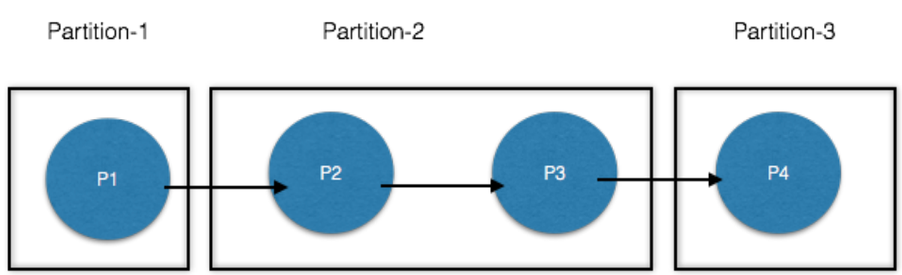
Computing Resources	Terracorrelator	SuperMUC	Amazon EC2	EDIM1
Type	Shared-memory	Cluster	Cloud	Cloud
Enactment Systems	MPI, multi	MPI, multi	MPI, Storm, multi	MPI, Storm, multi
Nodes	1	16	18	14
Cores per Node	32	16	2	4
Total Cores	32	256	36	14
Memory	2TB	32GB	4GB	3GB

Mode	Terracorrelator	SuperMUC	Amazon EC2	EDIM1
MPI	1501.32 (~25minutes)	1093.16 (~19minutes)	16862.73 (~5hours)	38656.94 (~11 hours)
multi	1332 .20 (~23minutes)			
Storm			27898.89 (~8 hours)	120077.123 (~33 hours)

1000 stations
Input 150MB
Output 39GB

Current work

- Diagnosis tool
  - How to partition the workflow automatically
  - How many processes execute each partition
- Run-time Stream Adaptive Compression



Current work – diagnosis tool

ConfigurationAmyRosaPartitions-Cross-New

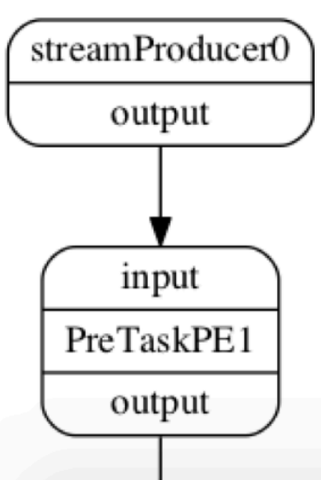
Overview

- **Started:** 2015-09-17T15:38:09.940
- **Completed:** 2015-09-17T15:38:26.024
- **Total runtime:** 0:00:16.084
- **Mapping:** multi
- **Number of processes:** 16
- **PE processes:**
  - SimpleProcessingPE15: [8, 9, 10, 11, 12]
  - SimpleProcessingPE17: [13, 14, 15]
  - SimpleProcessingPE16: [0, 1, 2, 3, 4, 5]
  - SimpleProcessingPE19: [6]
  - SimpleProcessingPE18: [7]

Monitoring

- Summary
- Timeline
- Diagrams
- Diagnostics

Workflow



Diagnosis tool – 4 elements:

1. Information collection
2. Mondgodb database
3. GUI to display the stored information
4. Tuning decisions

Current work – diagnosis tool

Partitioning

Suggested partitions:

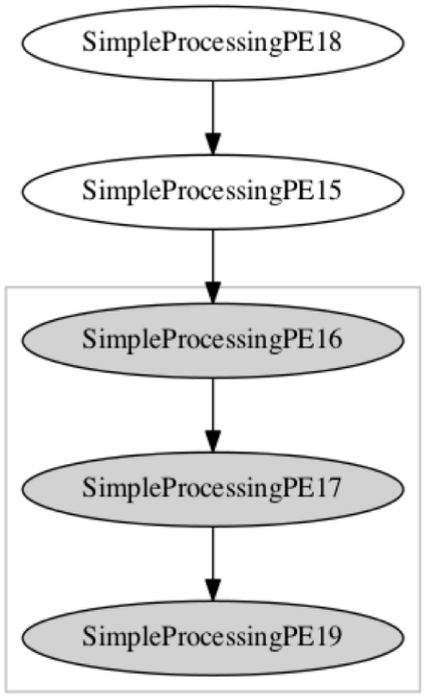
Show 50 entries

Search:

Partition	Number of processes
[SimpleProcessingPE18]	1
[SimpleProcessingPE15]	7
[SimpleProcessingPE16, SimpleProcessingPE17, SimpleProcessingPE19]	8

Showing 1 to 3 of 3 entries

Previous 1 Next



New algorithm -- tuning decisions:

1. Partitions
2. Number of processes per partition

Conclusions and Future work

- Python library for **streaming and data-intensive processing**
  - Users express their computational activities
  - Same workflow executed in several parallel systems
  - Easy to use and open
- **Future**
  - Support for PE failures
  - Select the best computing resource and mapping

Installations and Links

- This is all you need:

pip install dispel4py

- Web site http://dispel4py.org/
- GitHub: https://github.com/dispel4py/dispel4py
- Documentation: http://dispel4py.org/documentation/

Thanks and Questions

- Contact emails
  - Rosa Filgueira: rosa.filgueira@ed.ac.uk
  - Amy Krause: a.krause@epcc.ed.ac.uk
  - Malcolm Atkinson: Malcolm.Atkinson@ed.ac.uk