



Computing
For A
Changing
World.

MTAGS 2009

November 14-20, 2009
Oregon Convention Center
Portland, Oregon

Exploring Many Task Computing in Scientific Workflows

Eduardo Ogasawara

Fernando Seabra

Renato Elias
Braganholo

Alvaro Coutinho

Daniel de Oliveira

Carlos Barbosa

Vanessa

Marta Mattoso

Federal University of Rio de Janeiro, Brazil

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MTAGS '09 November 16th, 2009, Portland, Oregon, USA
Copyright © 2009 ACM 978-1-60558-714-1/09/11... \$10.00

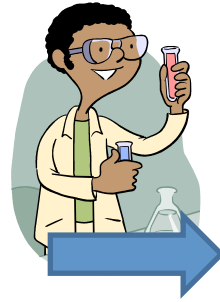


Agenda

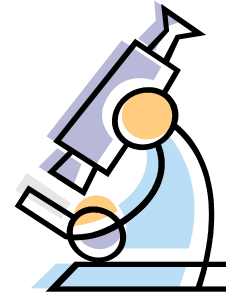
- Introduction
 - Scientific experiments
 - Scientific workflows
 - Experiments life cycle
- Hydra middleware
- Case study
- Related work
- Conclusion

Typical scenario: scientific experiment

This scenario demands the execution of many programs as a chain of activities that may be assisted by scientific workflows



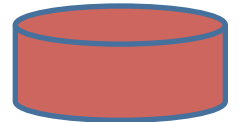
1. Data collection



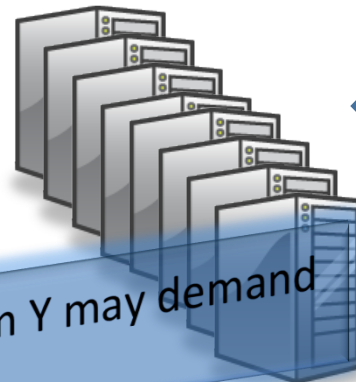
2. Data analyzed by program X



3. Large Volume of Data Produced ...



4. ...which need to be processed by program Y in a cluster



Program Y may demand HPC

5. Results are analyzed by program Z

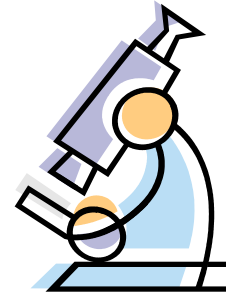


Variations of data or parameters

Many-task computing (MTC) is a very attractive paradigm to parallelize workflow activities in very large HPC



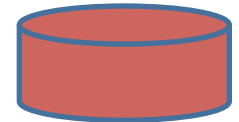
1. Data collection



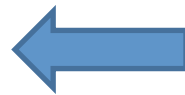
2. Data analyzed by program X



3. Large Volume of Data Produced ...



5. Results are analyzed by program Z



4. ...which need to be processed by program Y in a MTC environment

Current solutions

- Scientific Workflow Management Systems (SWfMS)
- SWfMS allow the execution of Scientific Workflows
 - Some SWfMS are strong in workflow design and provenance support (VisTrails, Kepler, Taverna)
 - Some SWfMS are strong in HPC support (Pegasus, Swift, Triana)
- Scientists should be free to choose the SWfMS that suits best for their needs
- This choice should not prevent the adoption of an MTC solution for executing one or more activities of a workflow

Parallelization difficulties

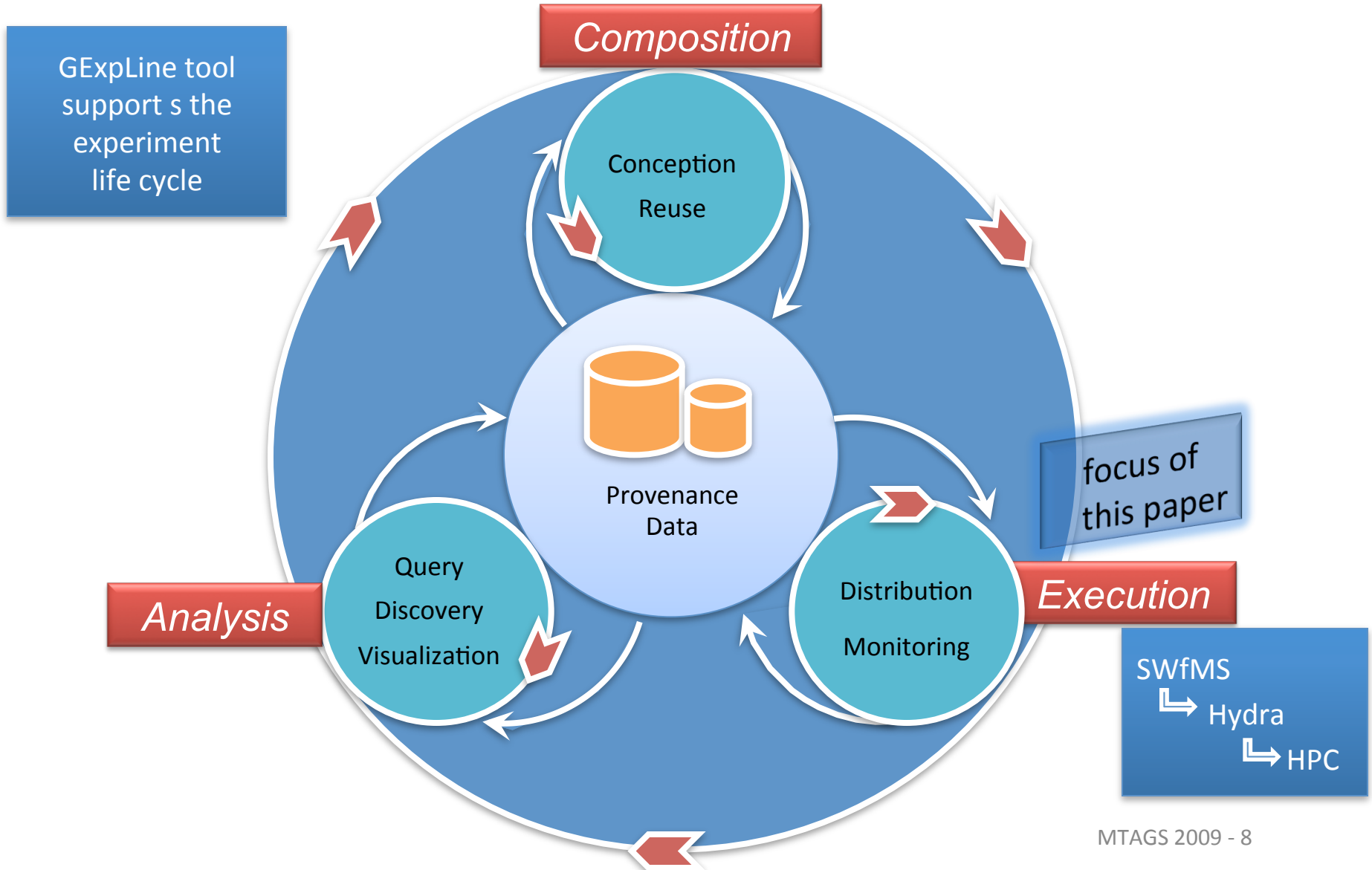
- Controlling parallel execution in distributed environments
- Steering activities in distributed environments
- Provenance gathering in distributed/heterogeneous environments

Provenance can support analyzing scientific experiments

- Before execution:
 - What programs may be used? Is there any alternative to explore?
 - Is there any dependency between activities? Which activities are mandatory?
- After execution:
 - What were the parameters that lead the best result?
 - What was the scientific workflow that lead to the desired result?
 - Where are the output files generated by the distributed activity A using the parameters P?
 - How many times the activity A in version V was used in the experiment E?

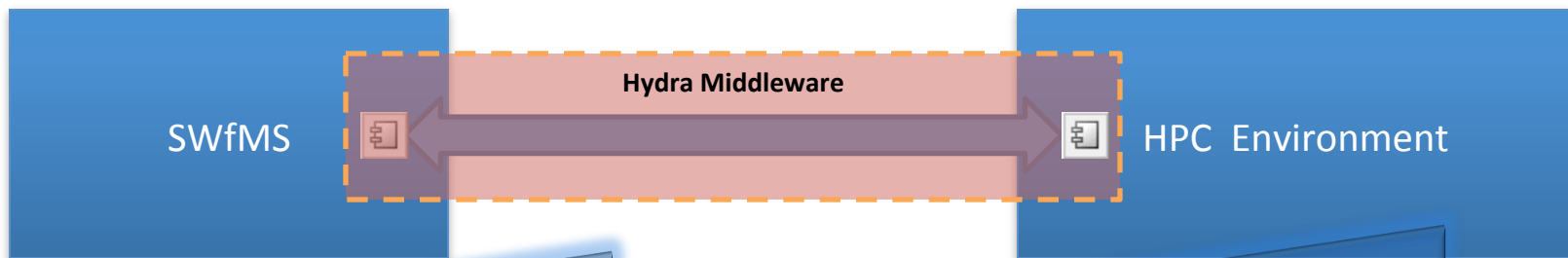
all these queries are related to the ability of reproducing and validating a scientific experiment

Our vision of the experiment life cycle



Hydra

- Middleware solution that bridges the SWfMS to the HPC supporting MTC parallelization strategies

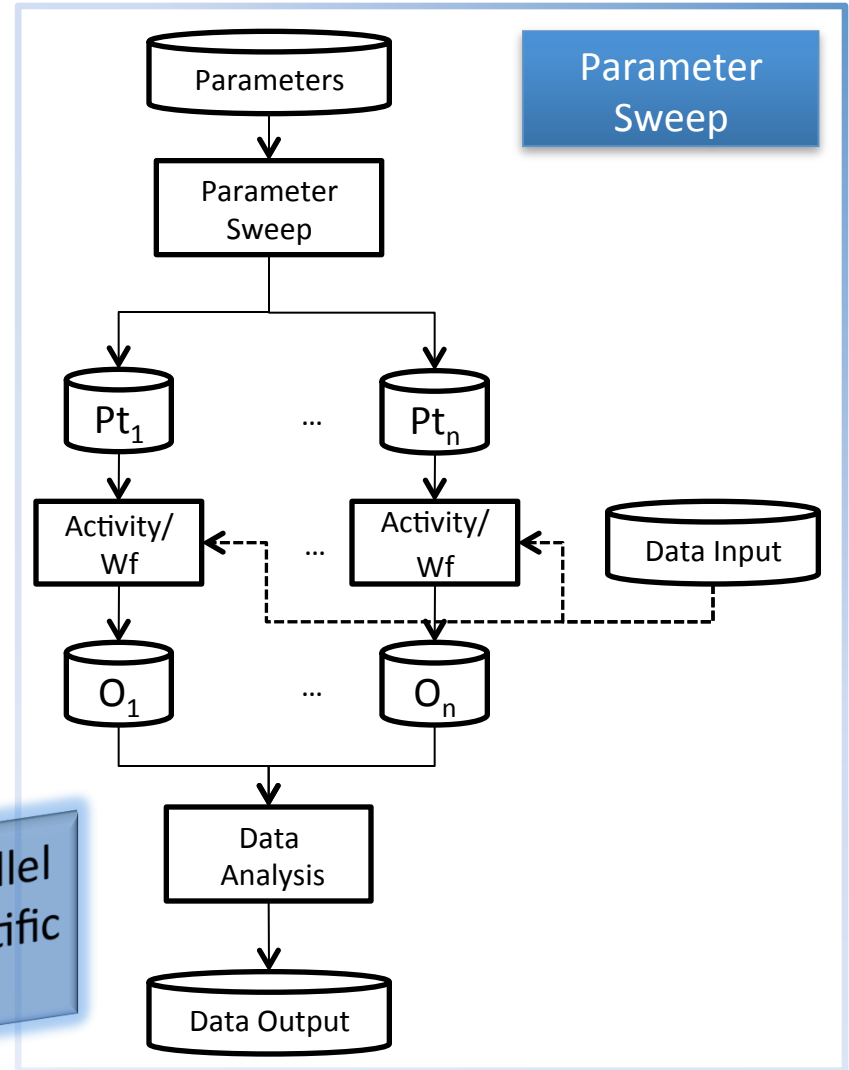
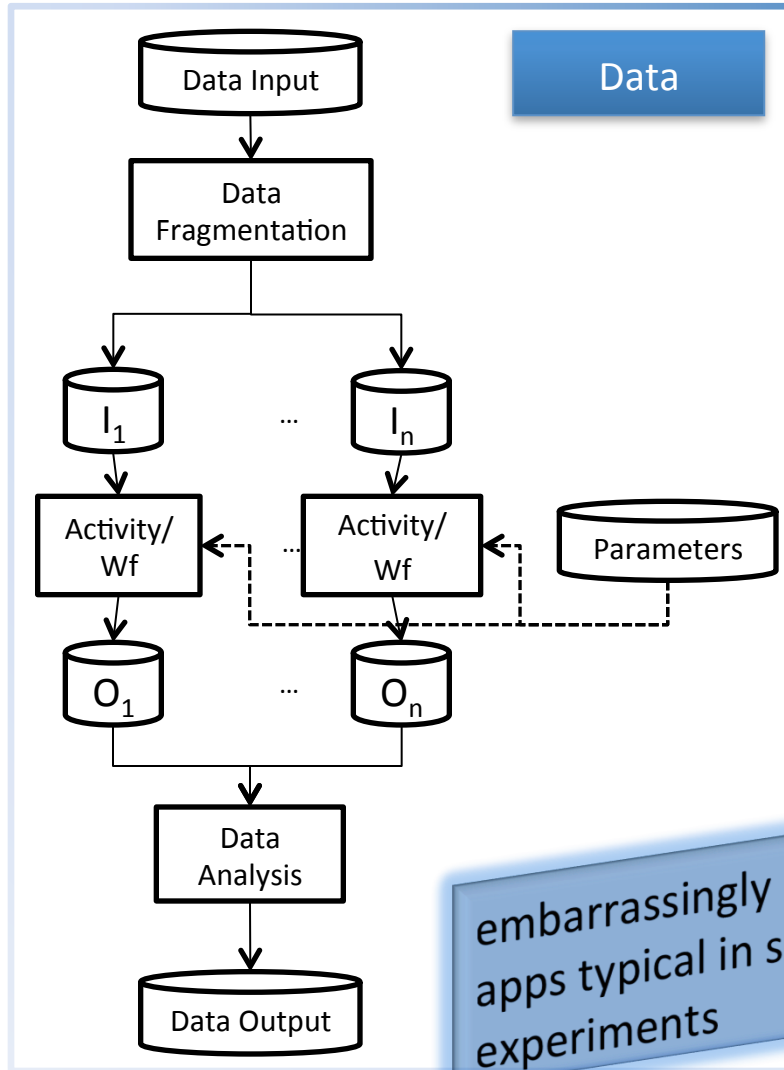


A set of components to be included on the workflow specification to control parallelization of activities as MTC

A set of components installed on the MTC environment that can register parallelization strategies to be reused with provenance gathering

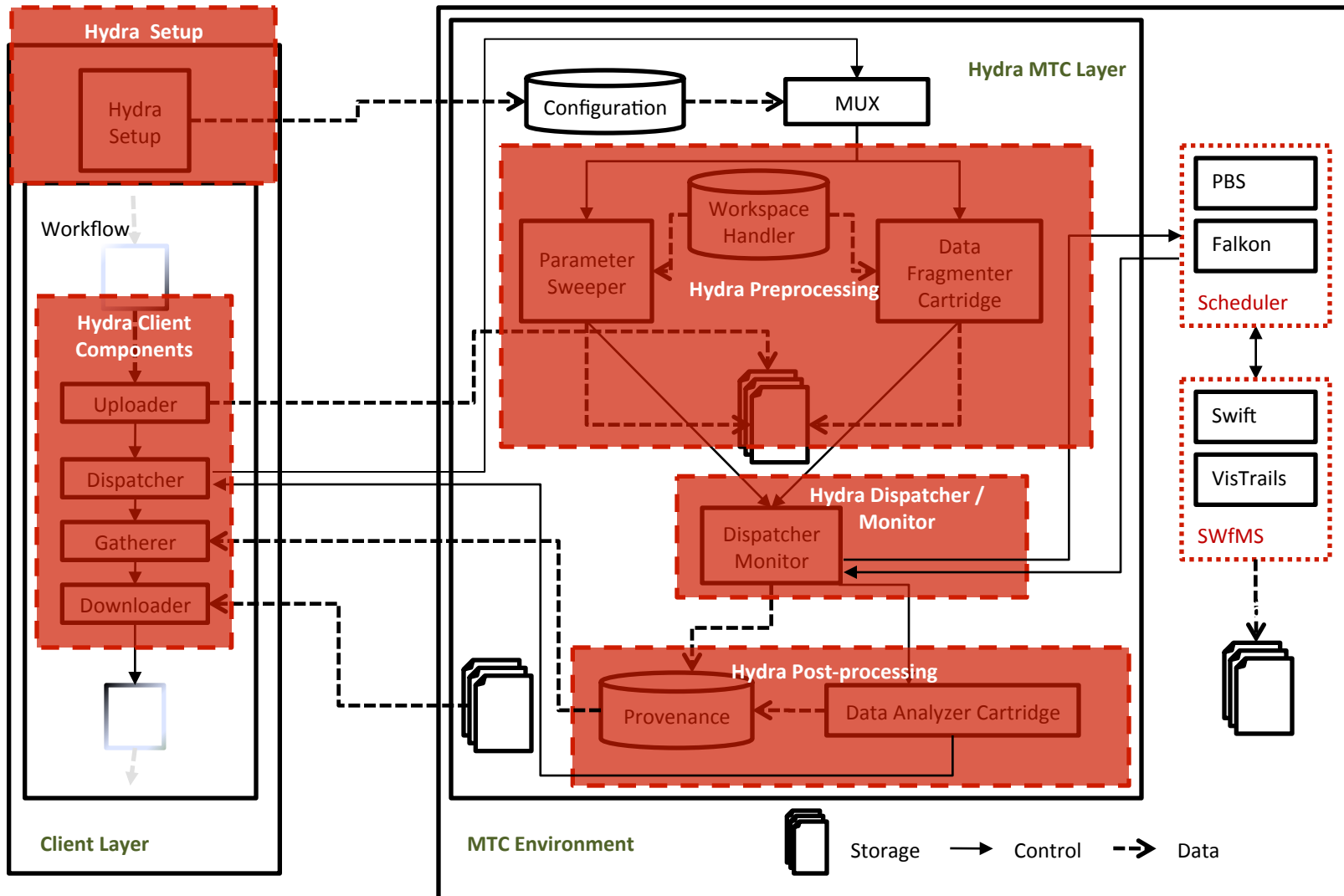
- Goal: reduce the complexity involved in designing and managing activity/workflow parallel executions while gathering distributed provenance data

Supported parallelization types



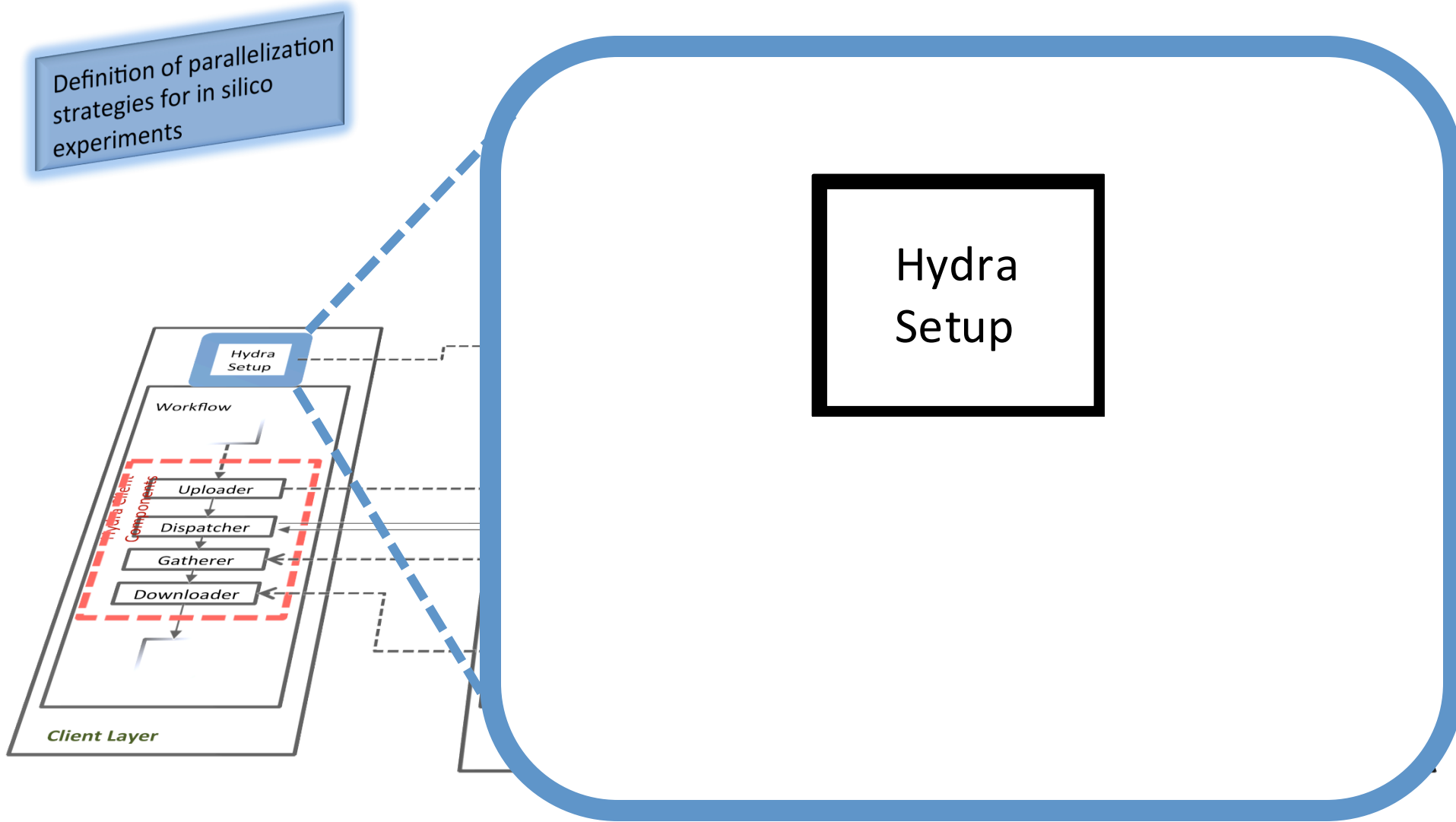
embarrassingly parallel
apps typical in scientific
experiments

Hydra Architecture



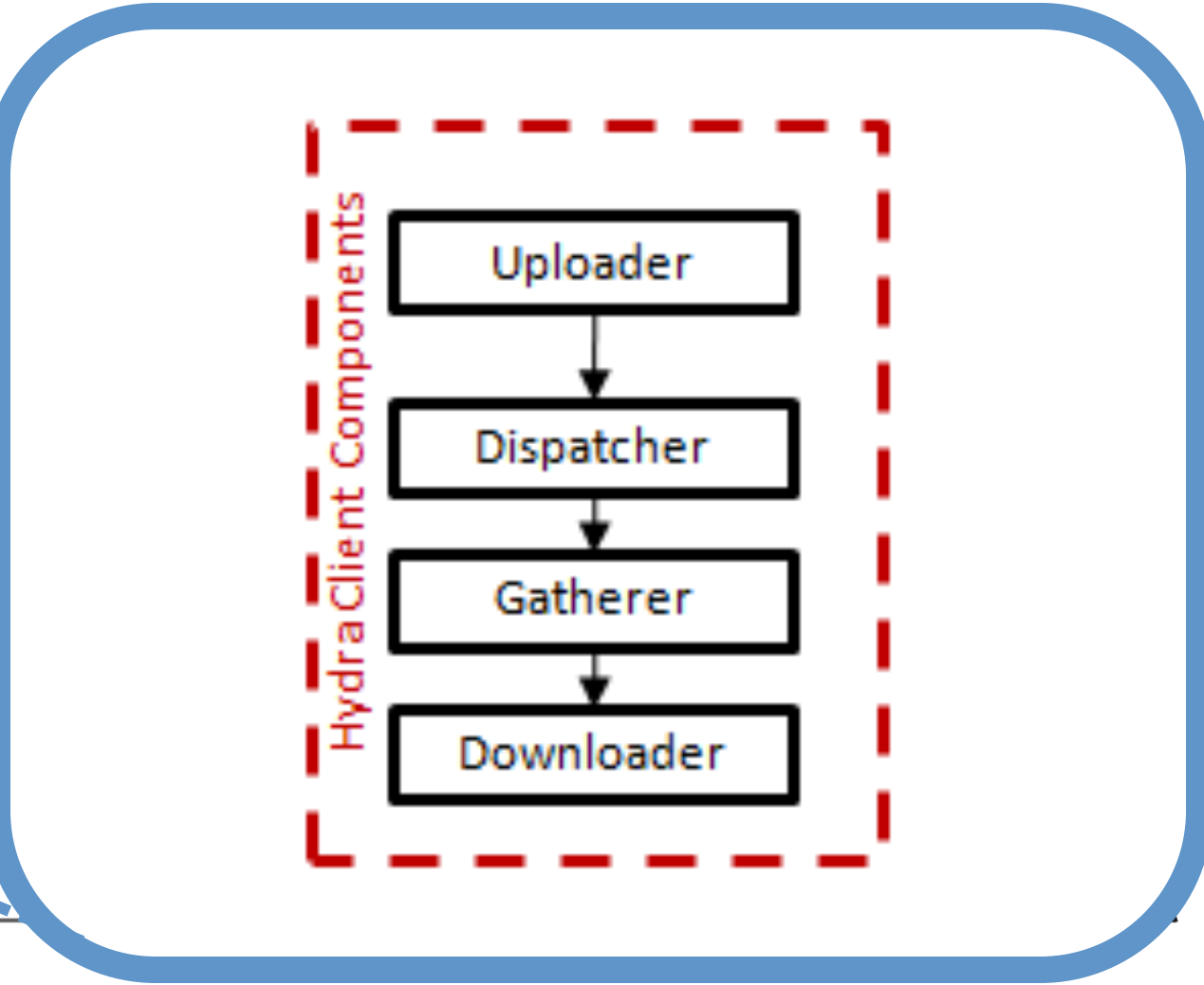
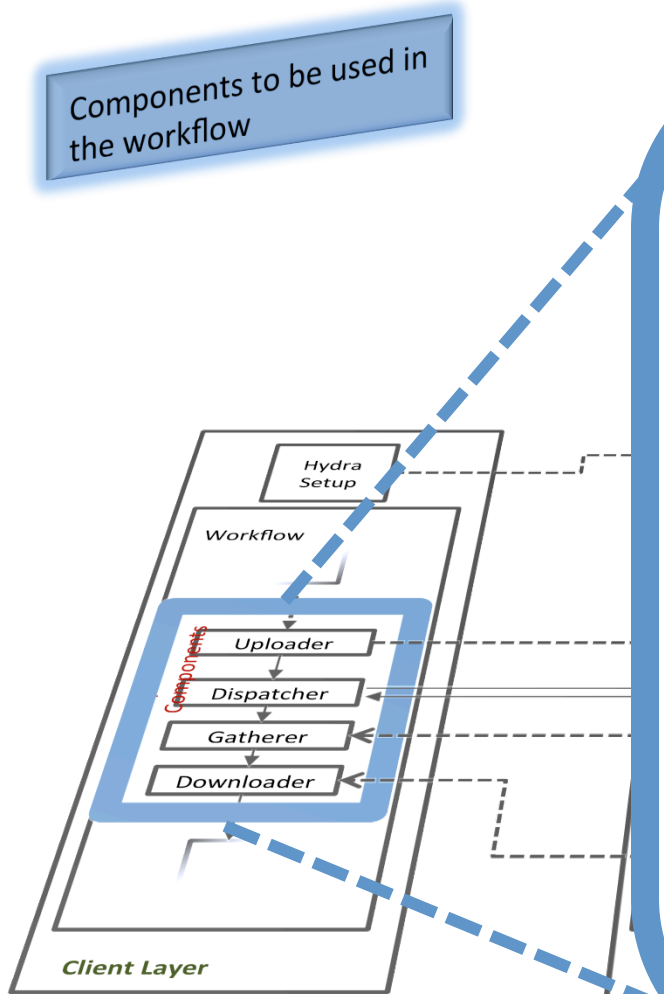
Hydra setup

Definition of parallelization strategies for in silico experiments

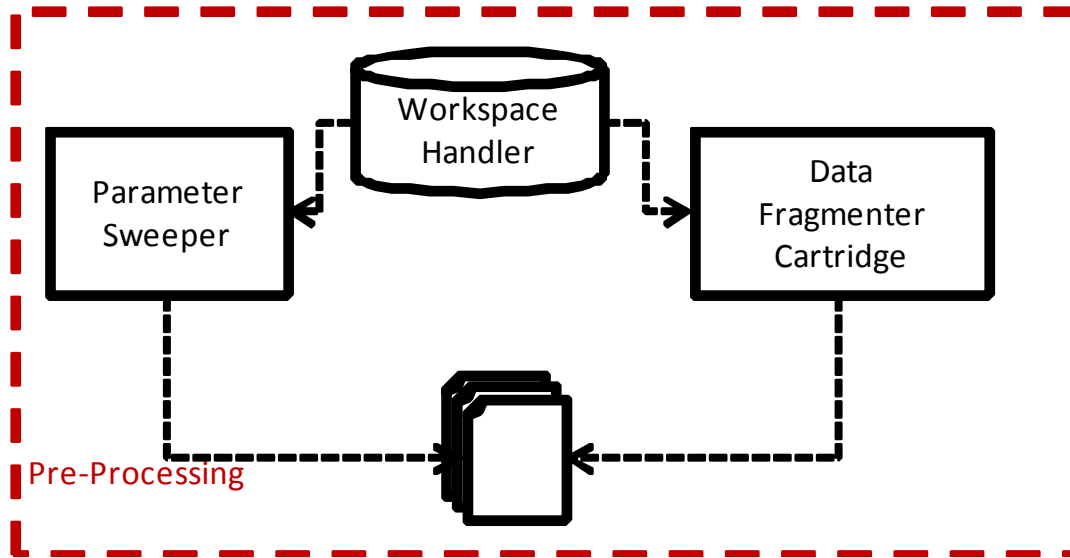


Hydra client components

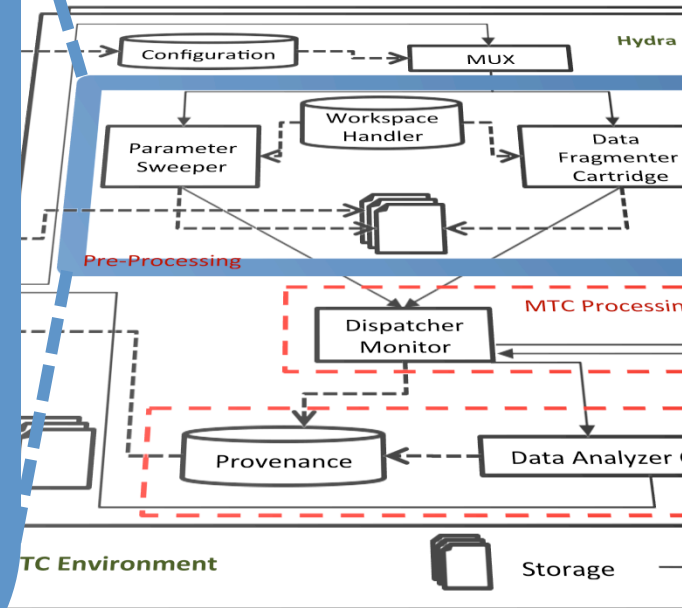
Components to be used in the workflow



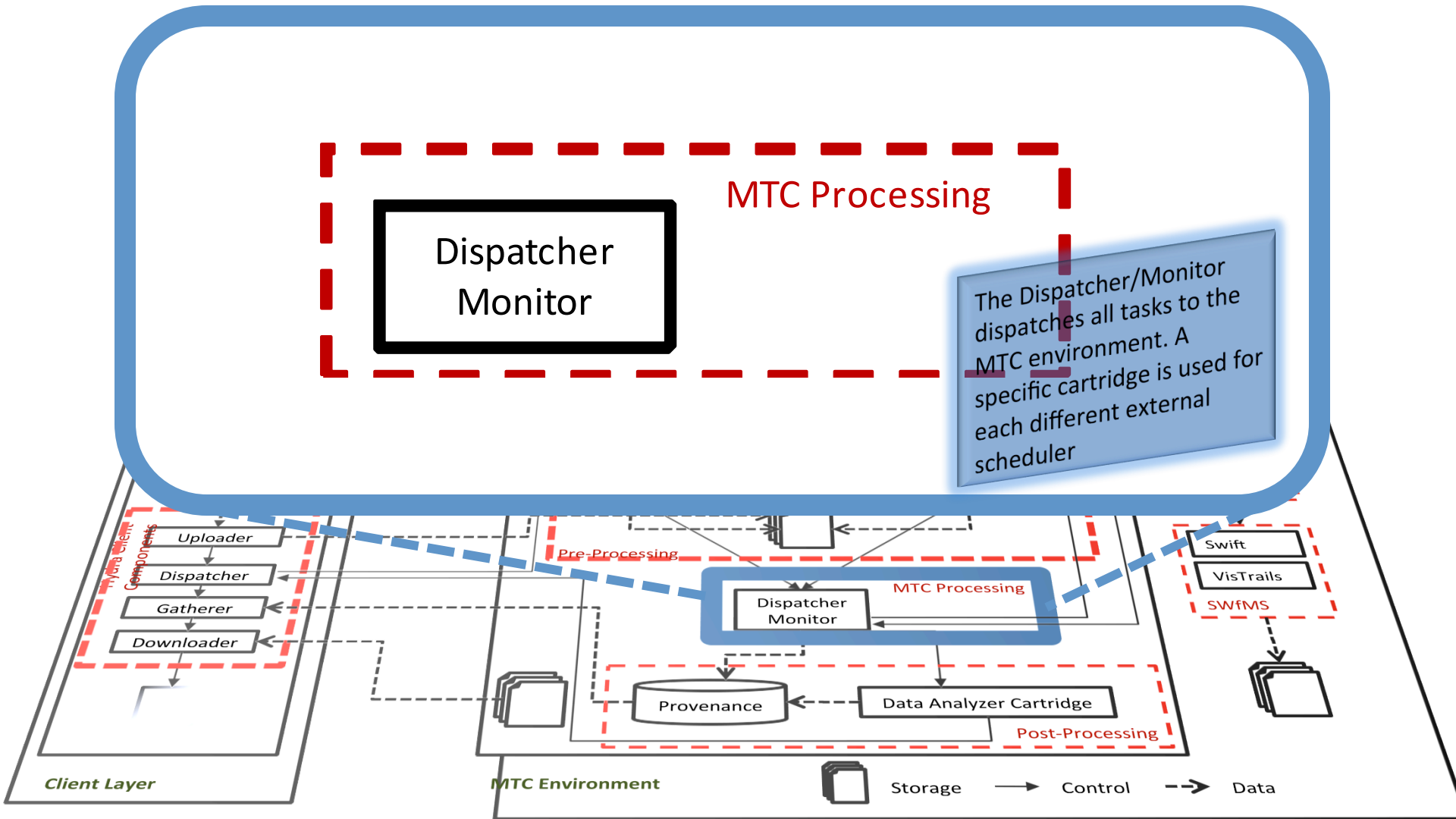
Hydra pre-processing components



Components that setup the distributed activity as many tasks. Each task has a workspace configuration

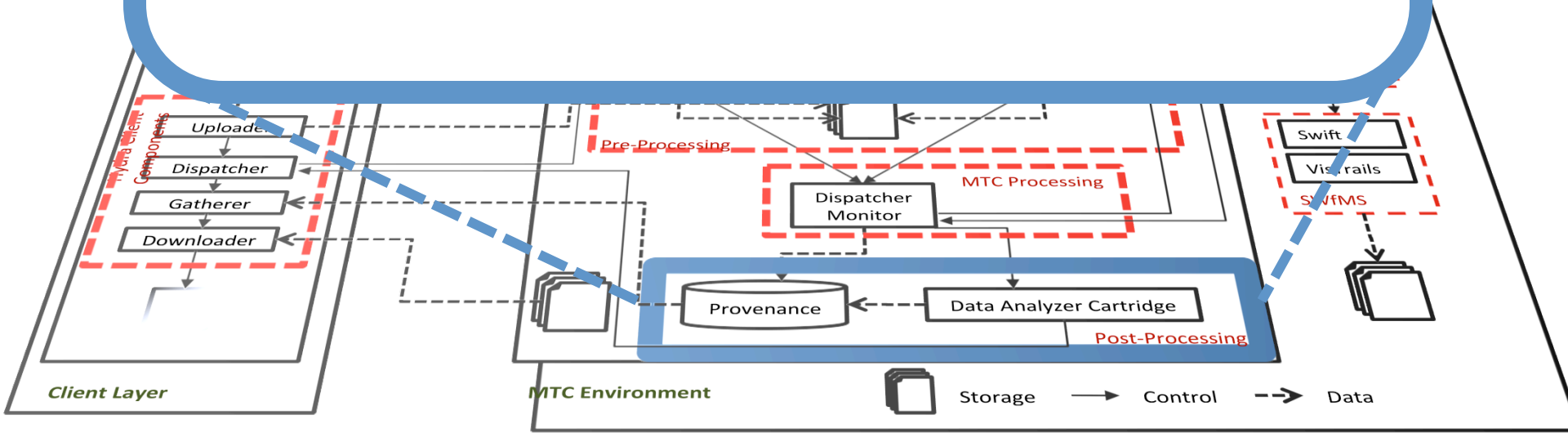
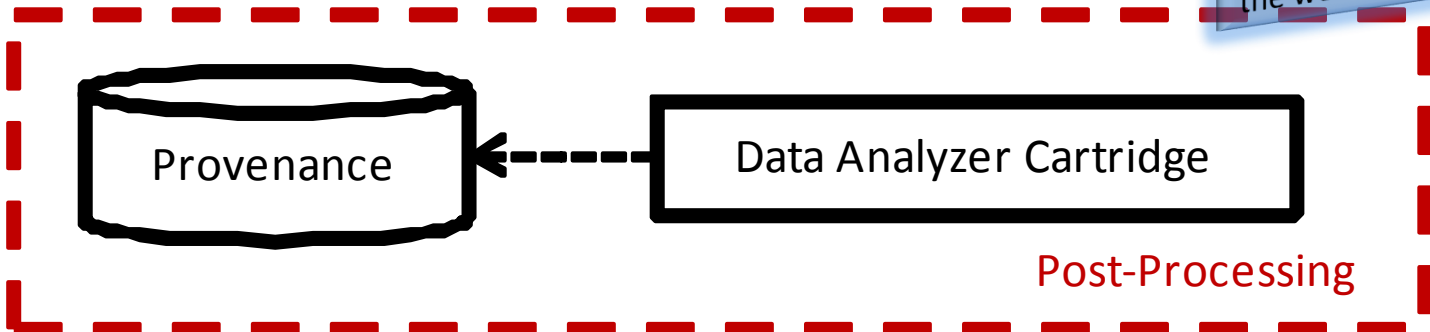


Hydra dispatcher/monitor components

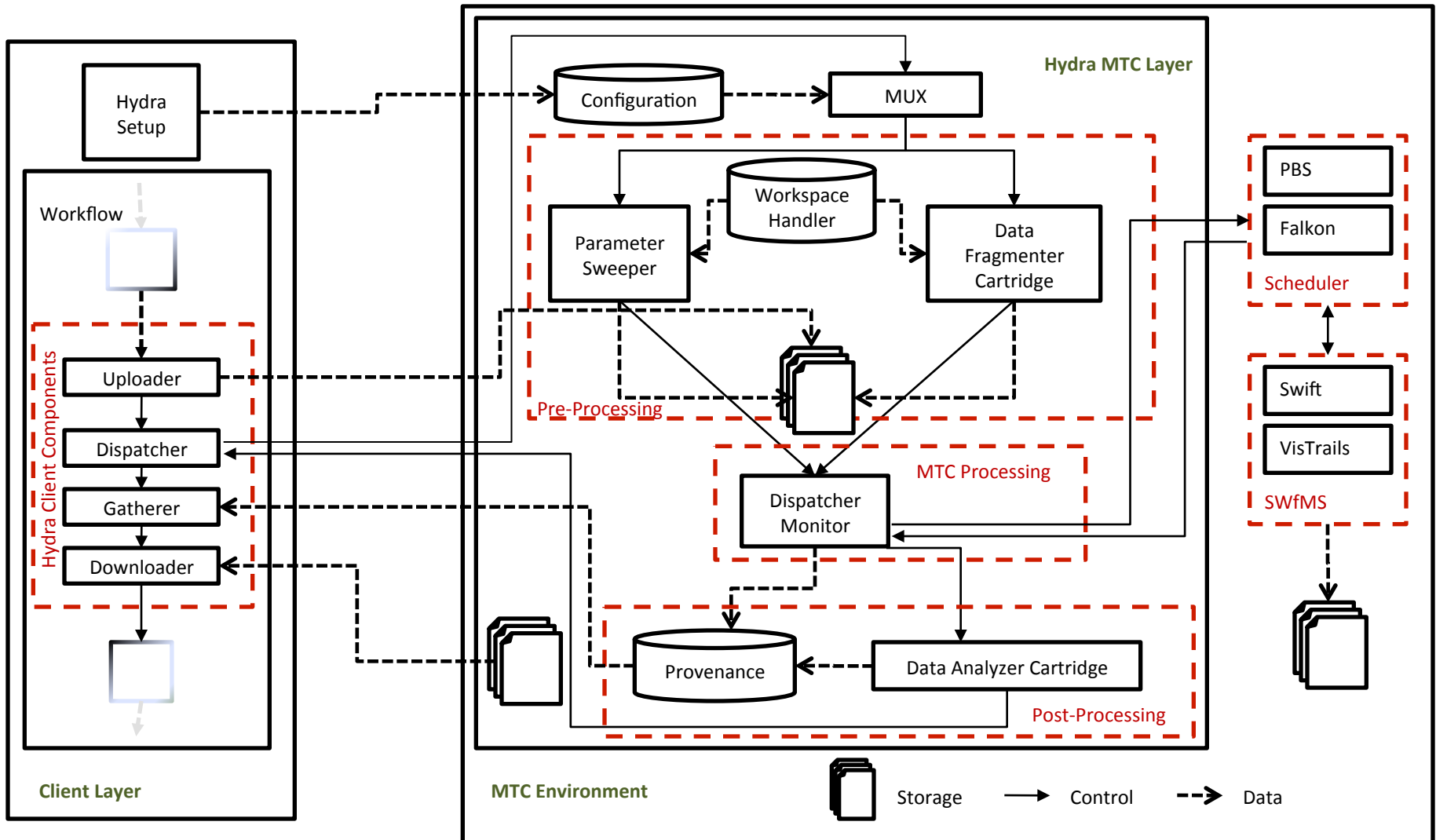


Hydra post-processing components

The distributed provenance is collected and made available to be gathered by the workflow

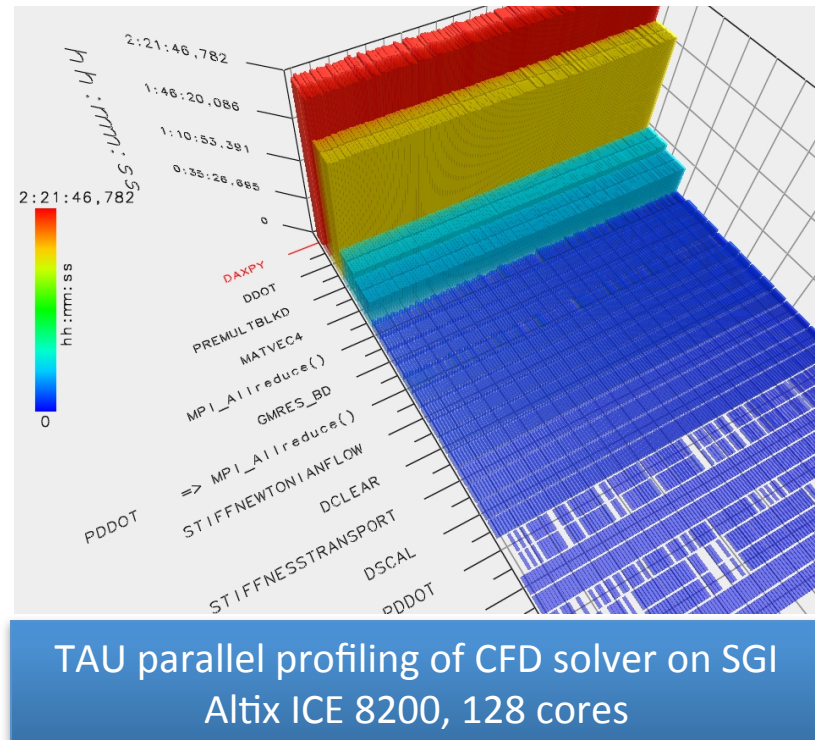


Hydra Architecture

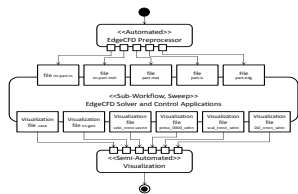


Case study

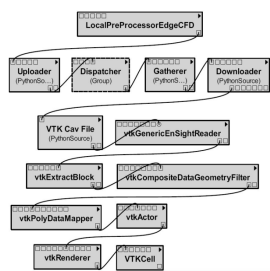
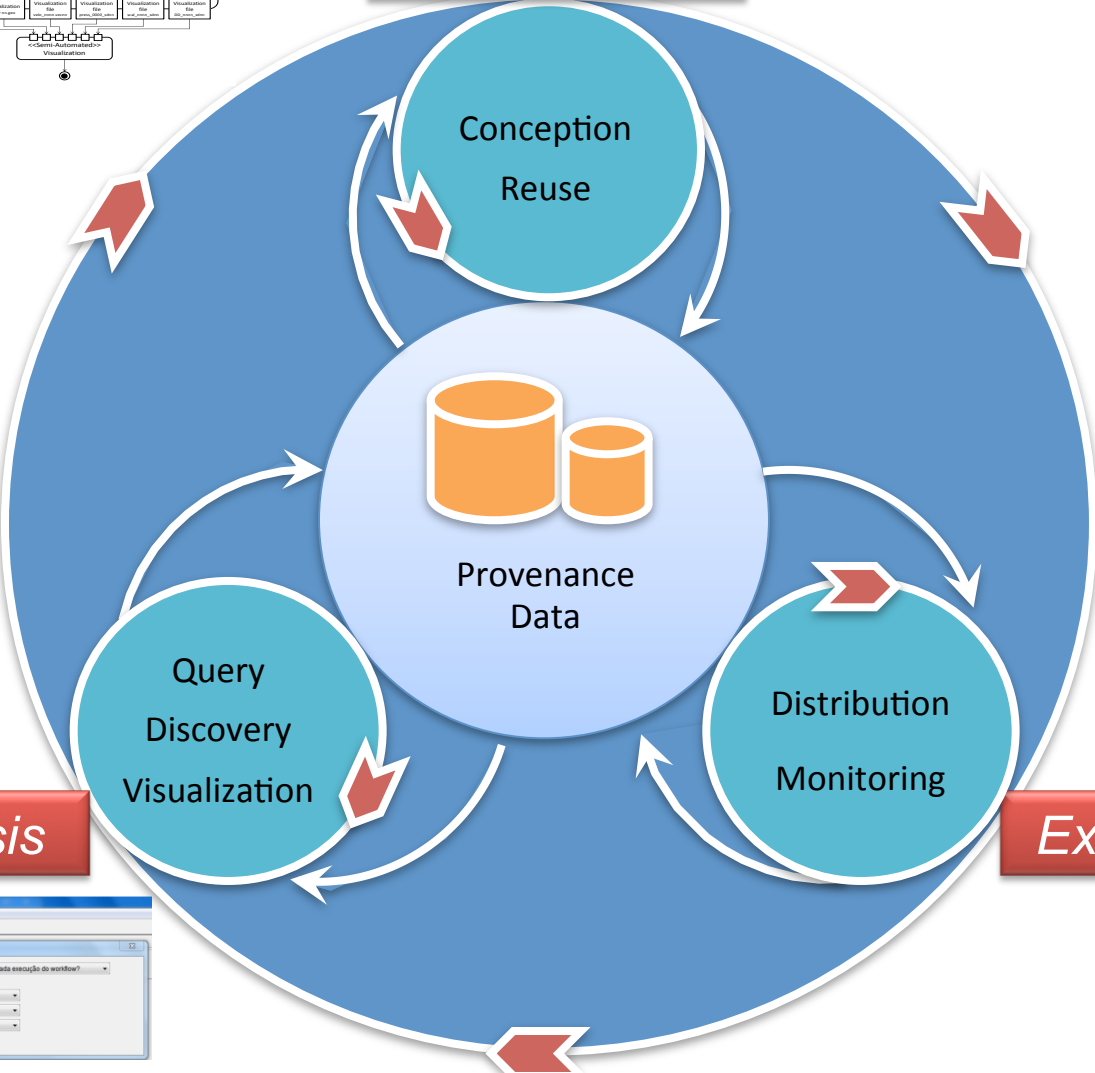
- Computational Fluid Dynamics (CFD)
- EdgeCFD: a parallel stabilized finite element incompressible flow solver
- Synthesized in four steps:
 - Modeling
 - Preprocessing
 - Solution
 - Visualization



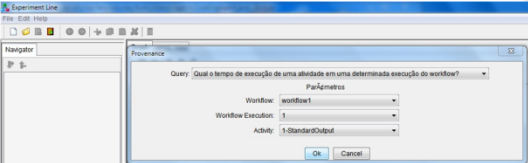
EdgeCFD experiment life cycle



Composition

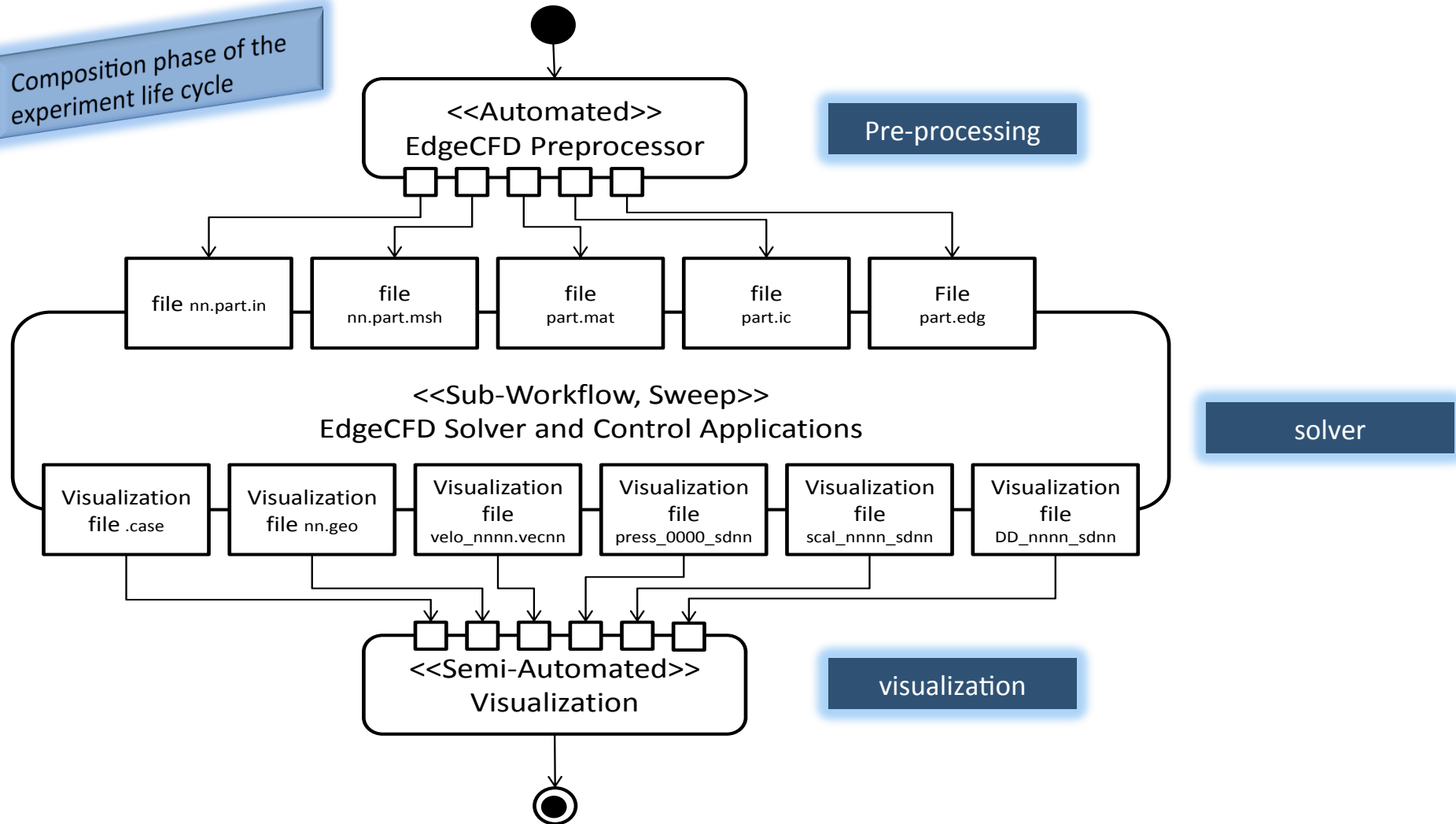


VisTrails & Hydra

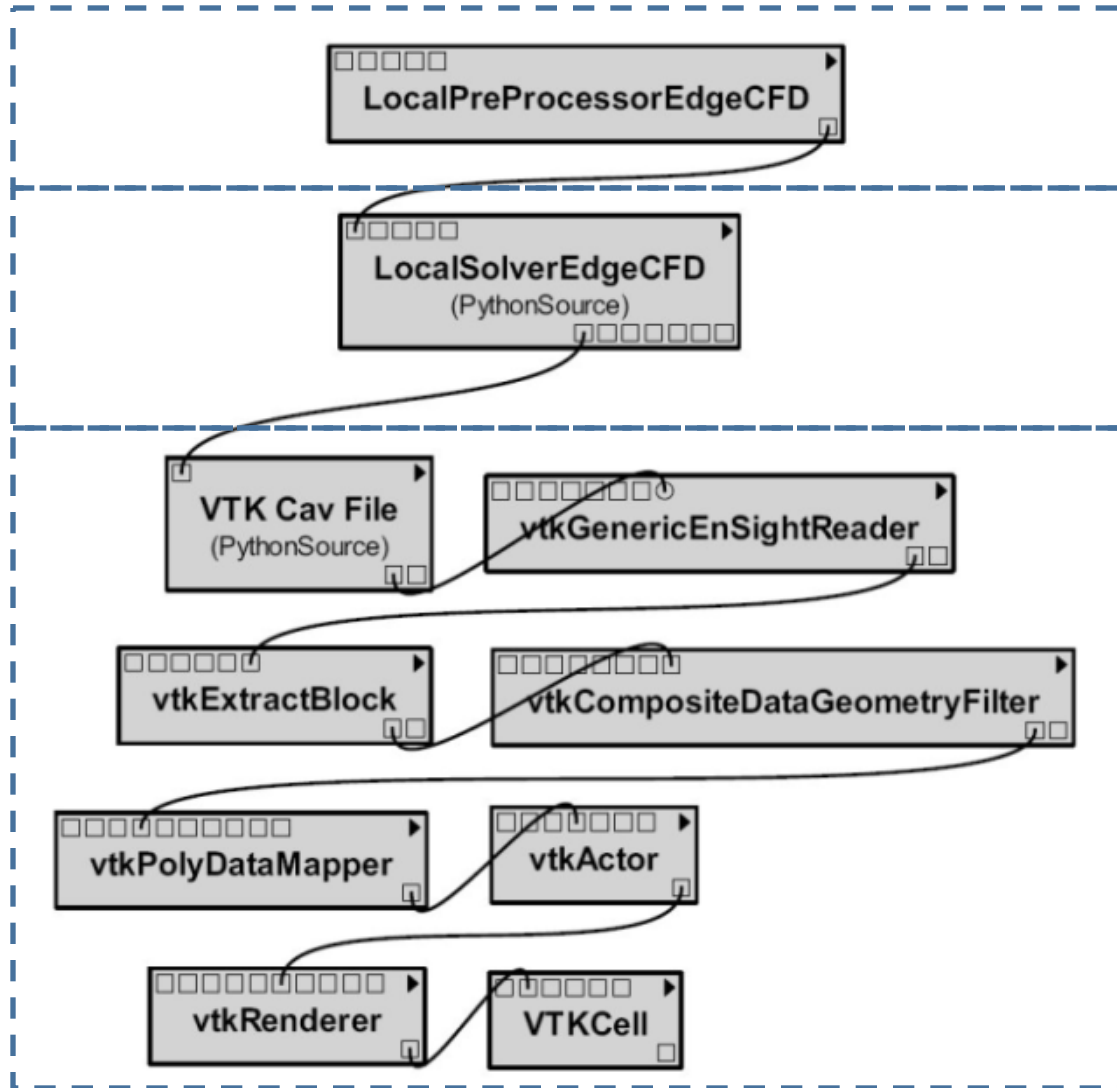


Workflow modeled in UML

Composition phase of the experiment life cycle



Sequential workflow

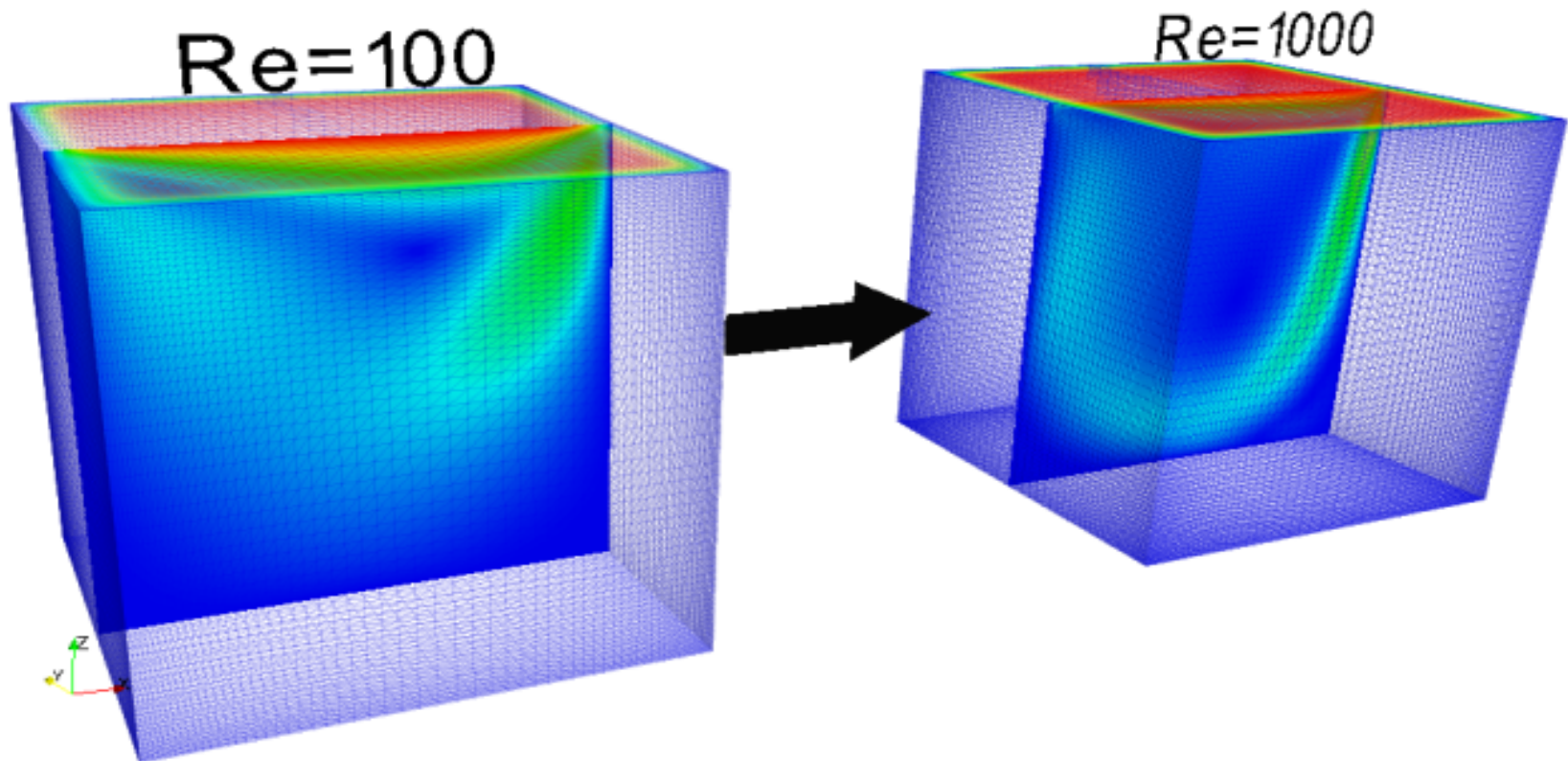


Pre-processing

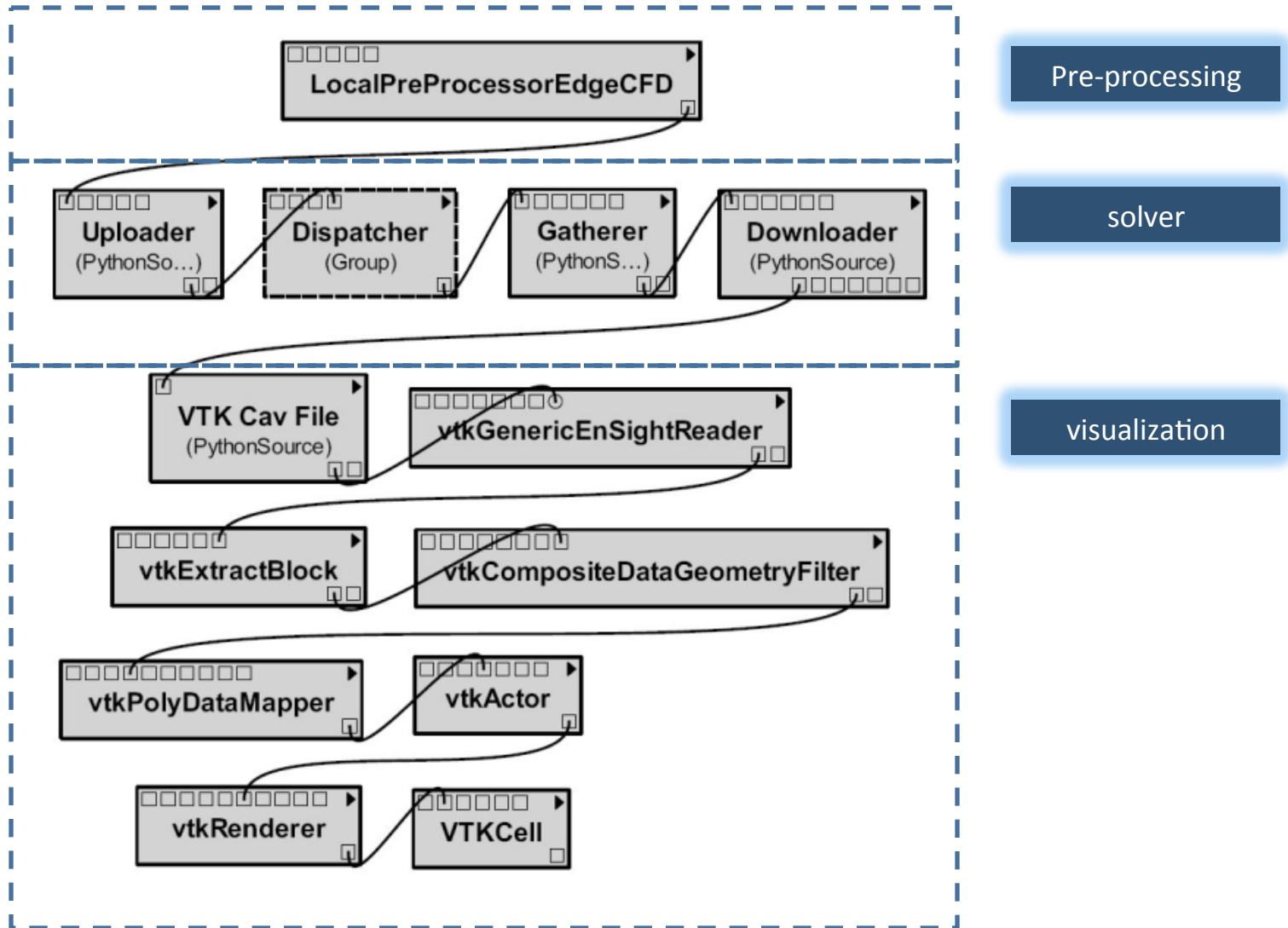
solver

visualization

Parameter sweep scenario



Workflow with parameter sweep using Hydra

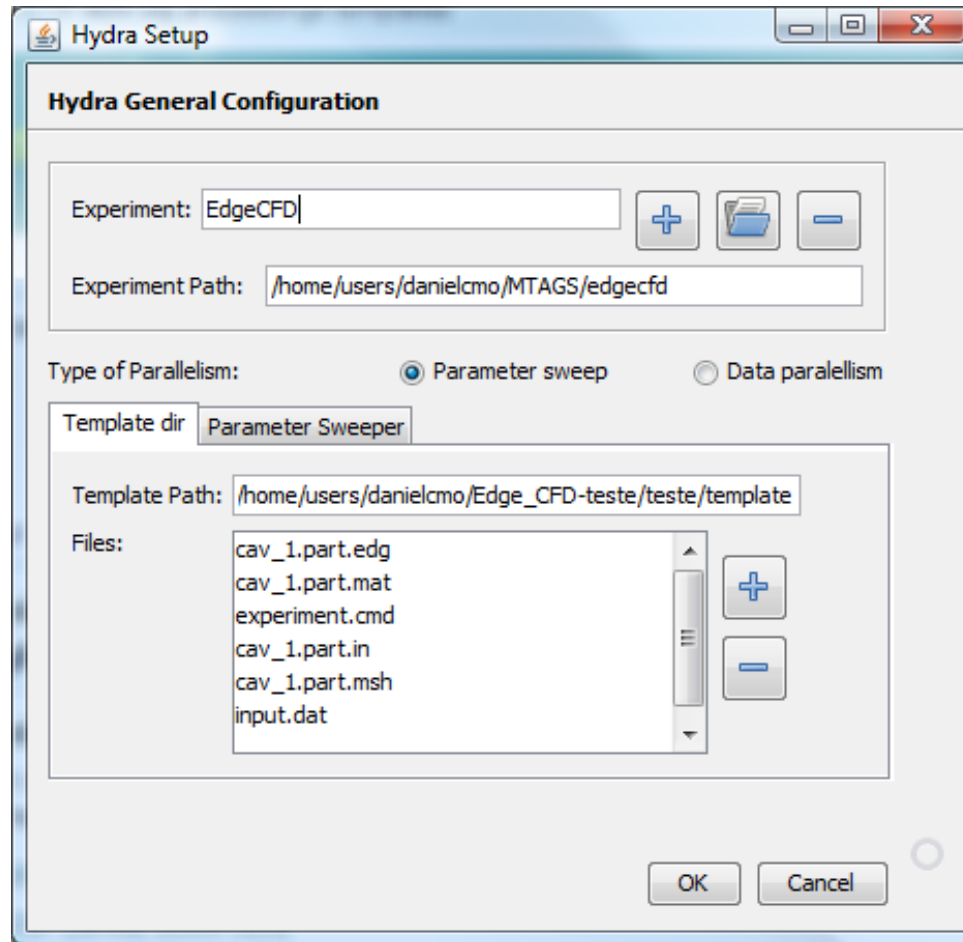


Pre-processing

solver

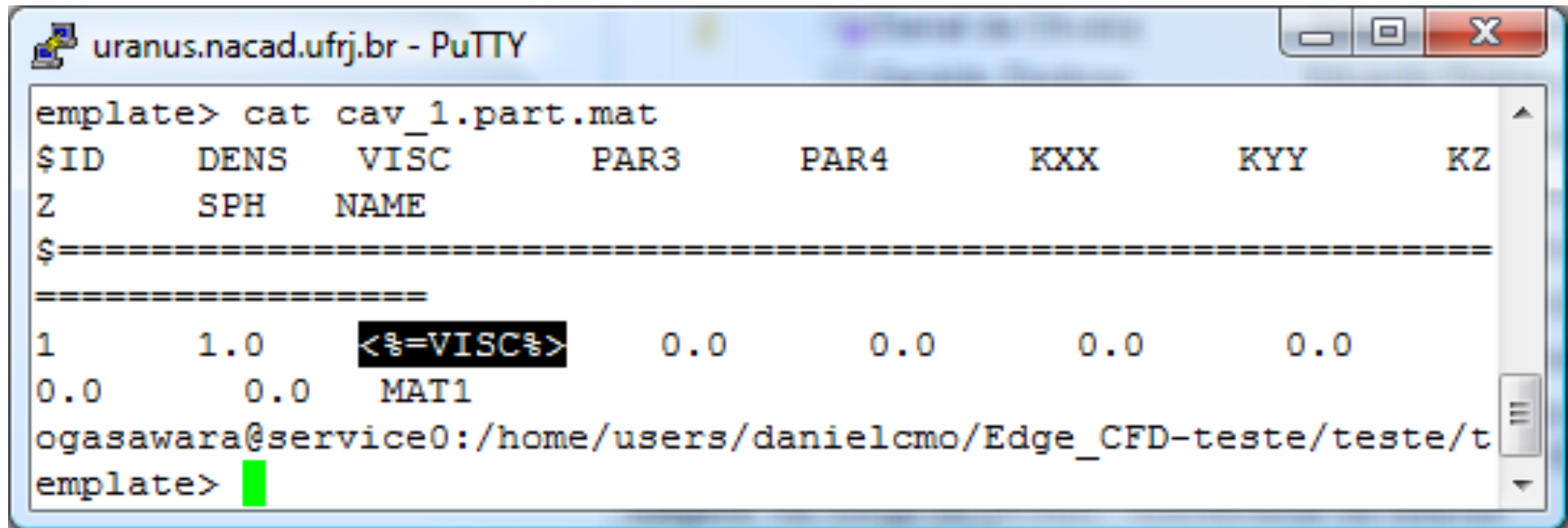
visualization

Hydra client setup for the solver activity



Prospective provenance for distributed environment

Instrumentation of files for the experiment



```
uranus.nacad.ufrj.br - PuTTY
emplate> cat cav_1.part.mat
$ID      DENS    VISC      PAR3      PAR4      KXX      KYY      KZ
Z        SPH     NAME
$=====
=====
1        1.0     <%=VISC%>  0.0       0.0       0.0       0.0
0.0      0.0     MAT1
ogasawara@service0:/home/users/danielcmo/Edge_CFD-teste/teste/t
emplate>
```

Instrumentation of the
configuration file
varying the viscosity

Hydra provenance

Table 1 - Summary of a Hydra MTC execution using 4 nodes with 8 cores obtained from the provenance repository

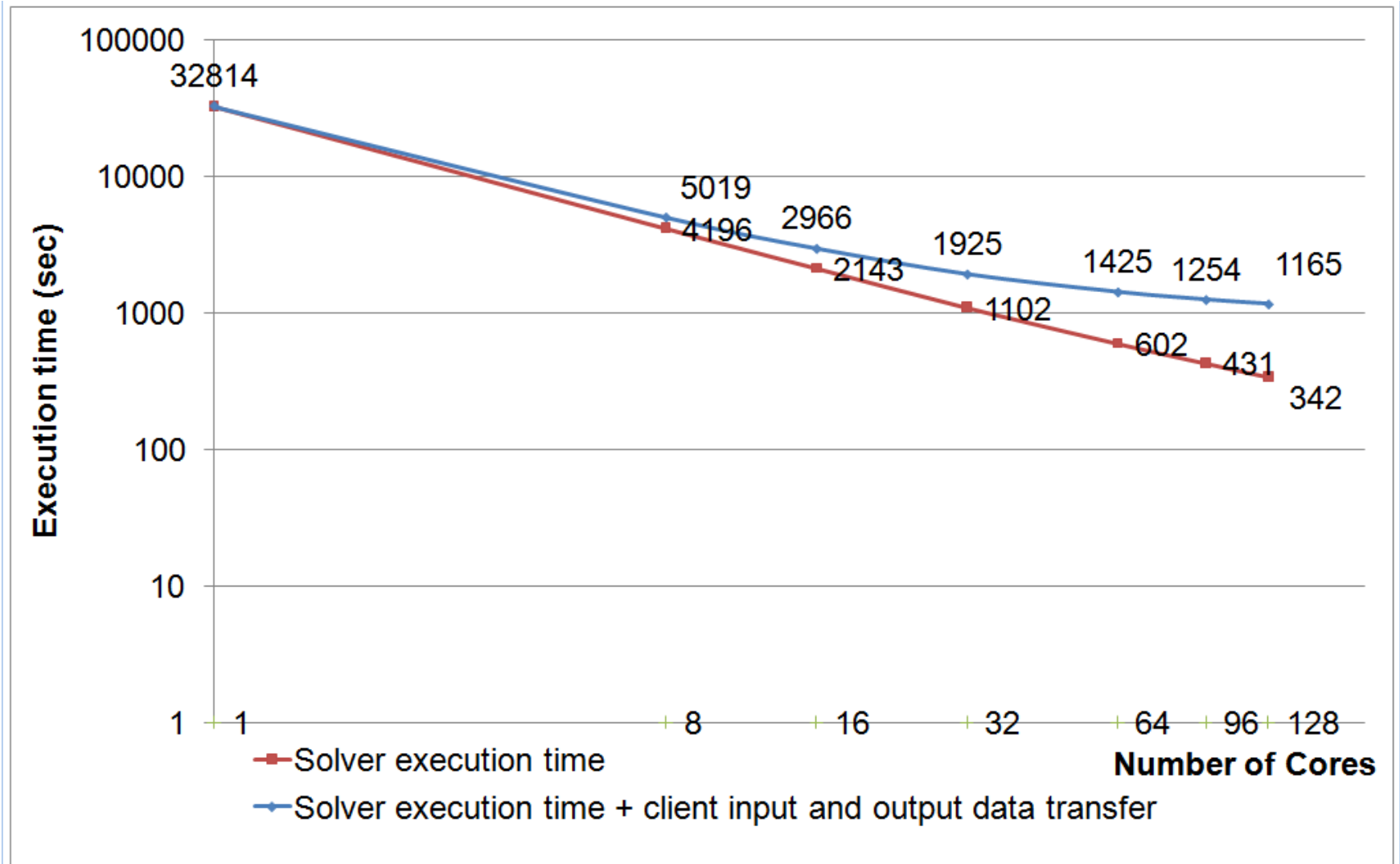
Number of activities (parameters explored)	128
Start date/time	2009-08-31 01:40
Uploader time	0.8 min
HPC Execution Time (Dispatcher + Gatherer)	1,671.2 min
Downloader time	11.6 min
Total Execution Time	1,683.6 min
Speedup	31.0
Number of errors	0

27 hours

Retrospective provenance
for distributed environment

An equivalent sequential
scenario would need 36 days

Evaluation of a small experiment



Related work

- Swift/Falcon
 - Provides MTC support from Swift SWfMS
- MyCluster
 - Supports PBS with transient fault support over remote sites
- Dryad
 - Supports data parallelization with high scalability
- Sawzal
 - It is a framework for MTC that explore data parallelism

Conclusions

- Experiments life cycle must be managed as a whole:
 - Composition: experiment is modeled in a workflow abstraction level until being deployed into a specific SWfMS
 - Execution: some activities demand HPC with monitoring facilities and provenance gathering
 - Analysis: uses both information from the composition (prospective provenance) and from execution (local and distributed - retrospective provenance)
- Hydra can be a bridge between the SWfMS and the HPC environment
 - Supports workflow data and parameter sweep parallelization
 - Evaluated in a real case CFD solver with little overhead
 - Supports distributed provenance gathering

Future work

- Evaluate different kinds of applications (e.g. blast, uncertainty quantification)
- Model distributed activities that are actually sub-workflows
- Run experiments in HPC with more cores



Computing
For A
Changing
World.

MTAGS 2009

November 14-20, 2009
Oregon Convention Center
Portland, Oregon

Exploring Many Task Computing in Scientific Workflows

Thank you!

Eduardo Ogasawara

Fernando Seabra

Renato Elias

Braganholo

Alvaro Coutinho

Daniel de Oliveira

Carlos Barbosa

Vanessa

Marta Mattoso

Federal University of Rio de Janeiro, Brazil

Please visit our site
<http://gexp.nacad.ufrj.br>

