3rd IEEE Workshop on Many-Task Computing on Grids and Supercomputers

MTAGS 2010

Improving Many-Task Computing in Scientific Workflows Using P2P Techniques

Jonas Dias
Eduardo Ogasawara
Daniel de Oliveira
Esther Pacitti
Marta Mattoso

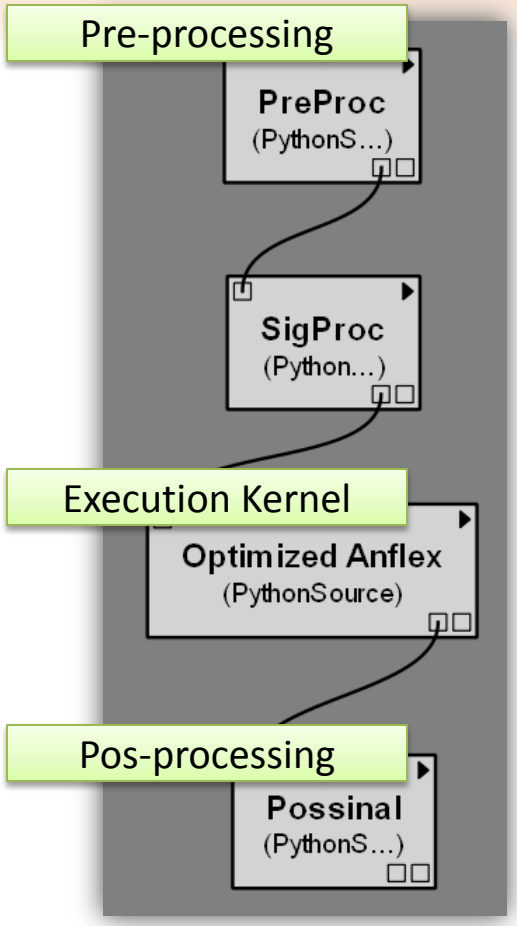
COPPE, Federal University of Rio de Janeiro, Brazil
INRIA & LIRMM, Montpellier, France

SC10 New Orleans, LA

COPPE UFRJ

Introduction

- Scientific Experiments
- Petascale Computing
  - Behavior of hundreds of thousands processors
  - Parallel Execution **failures**
- Scientific Workflows
  - Represent the chaining of activities of an experiment
  - Scientific Workflow Management Systems (SWfMS)



Typical Scientific Workflow

Experiment Execution

- The same workflow may run several times
  - 5000 parameter combinations to try
  - 3 workflow variations
  - Total of **15000** instances to be executed
- Motivation to parallelize
  - Accomplish the results timely
  - Clusters, Grids and Clouds
- Utility Computing model
  - Give the answer when they are still necessary

Difficulties in Workflow Parallelism

- MPI
  - Complex and legacy codes
  - Dynamic resource management
  - A job's process may fail
    - Compromise the whole execution
    - Resubmitting relies on the scientist manual control
      - Not feasible for a huge number of tasks
- Grid Schedulers
  - Submit many Jobs simultaneously
  - Waiting time on resource management queues

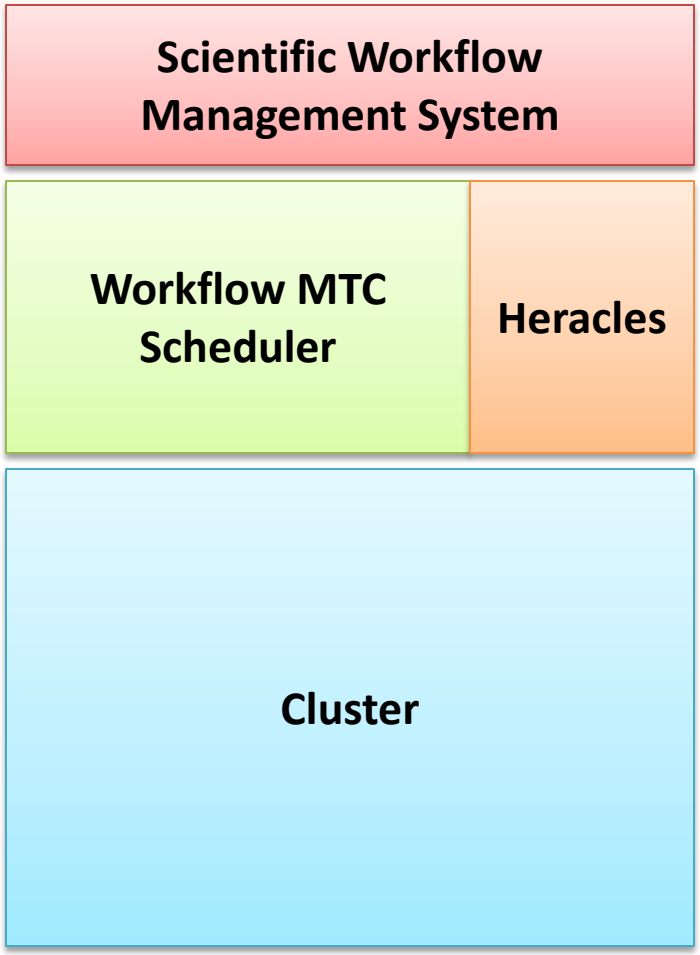
MTC Workflow Parallelism

- Many-task computing (MTC)
  - Improve Parameter Sweep and Data Parallelism
- HPC Cluster Systems
  - Not very easy to setup Jobs to be submitted
  - Centralized control
  - Compute nodes may fail
- Open Issues
  - Best approaches to **setup an experiment** execution
  - **Load balancing**
  - **Dynamic resource management**
  - Control the failures
    - What has **failed** and needs to be **rescheduled**?

MTC, Workflows and Clusters

- The Heracles Approach
  - Approach to execute workflow activities
    - More transparent setup
    - **Load Balancing**
    - **Quality of service**
    - **Distributed Provenance Gathering**
  - Uses the P2P model
    - To be implemented in a cluster scheduler
    - Not P2P infrastructure

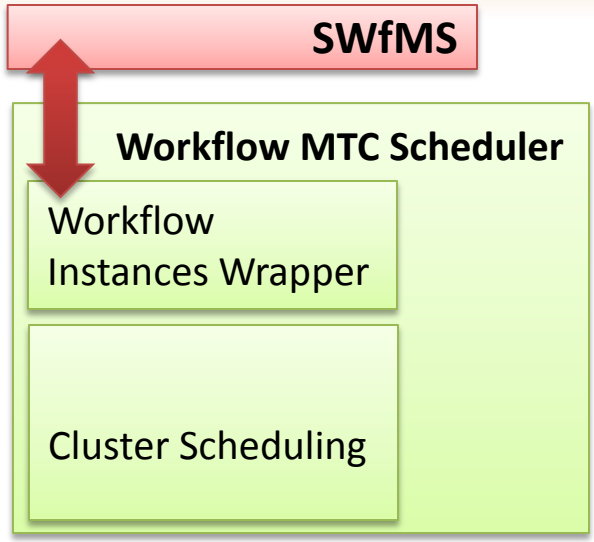
Heracles Overview



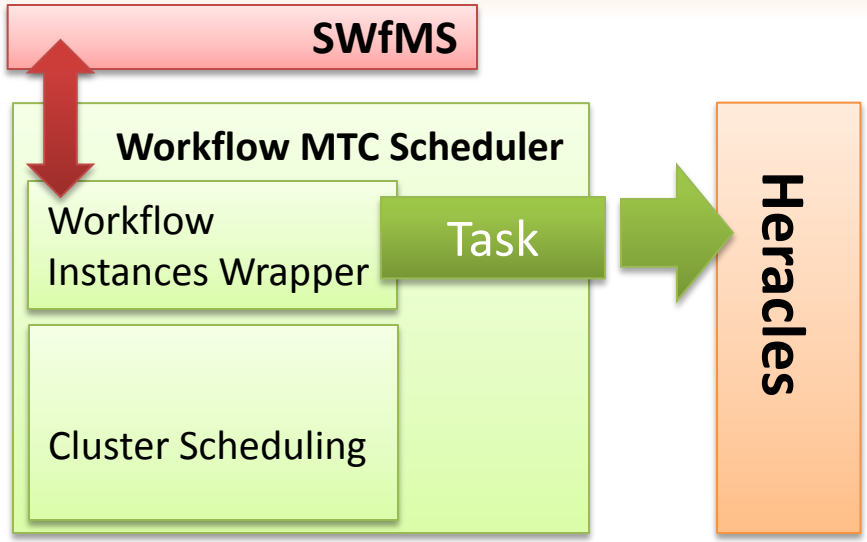
Heracles Structure

SWfMS

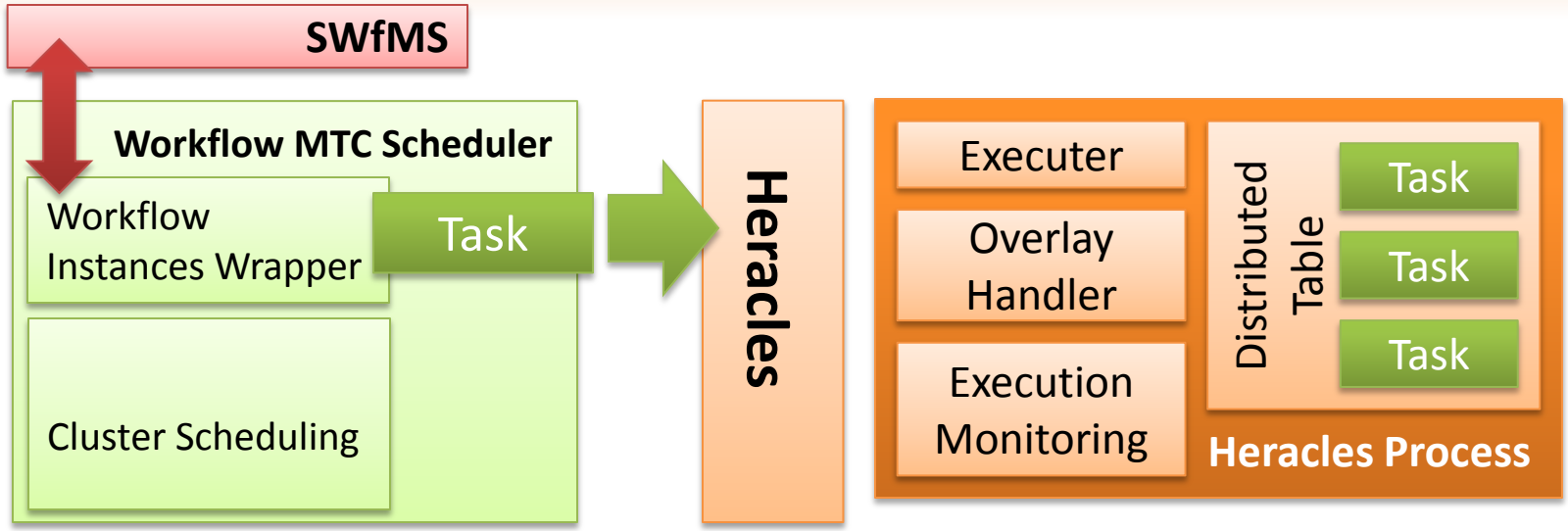
Heracles Structure



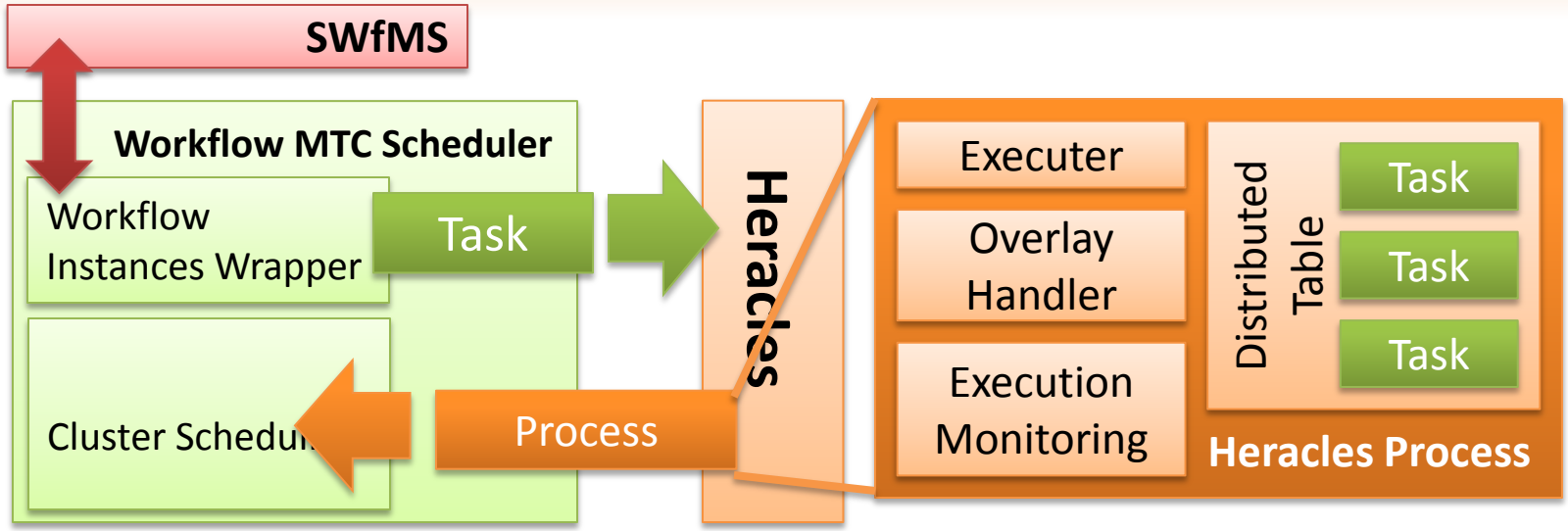
Heracles Structure



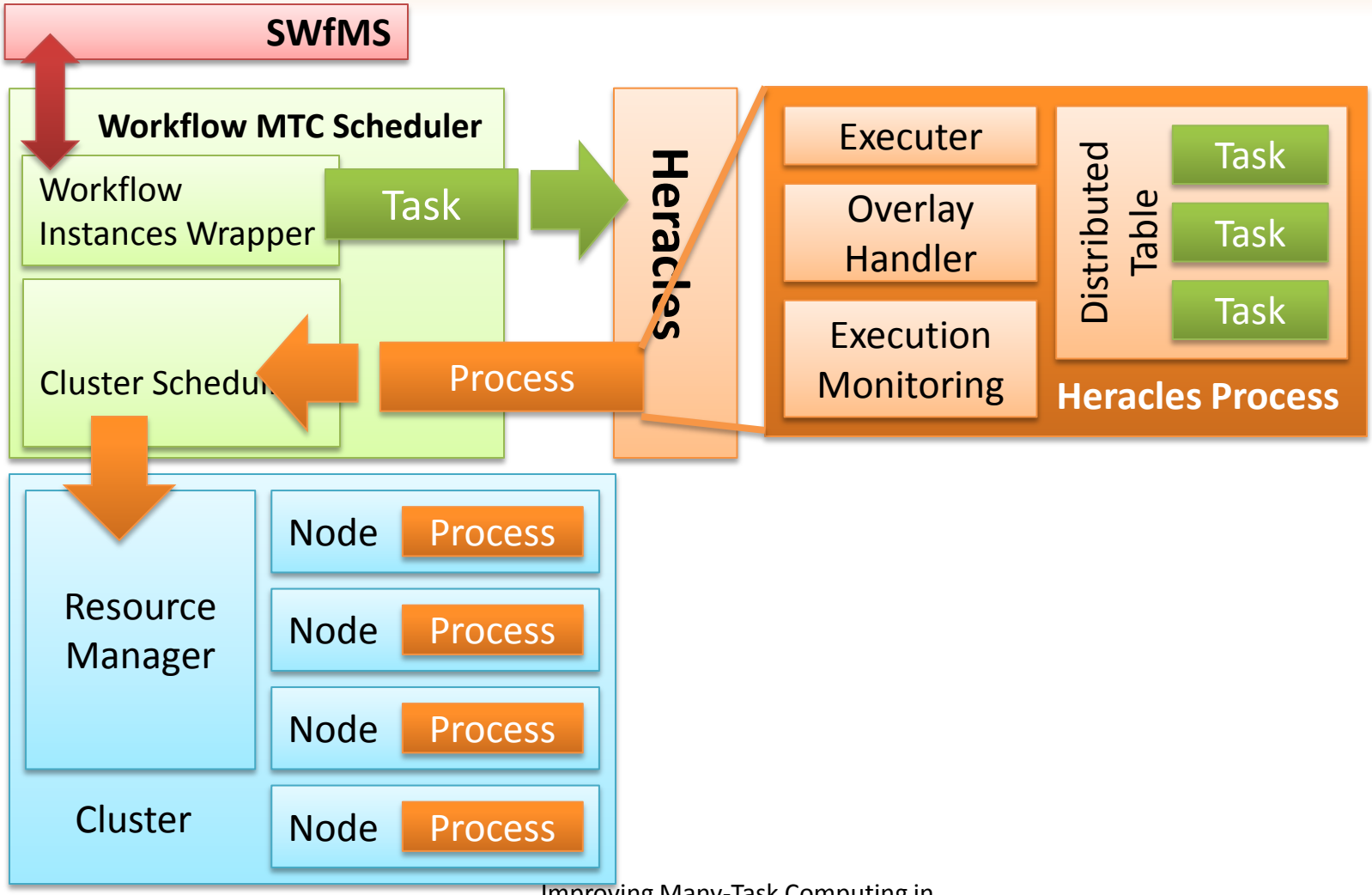
Heracles Structure



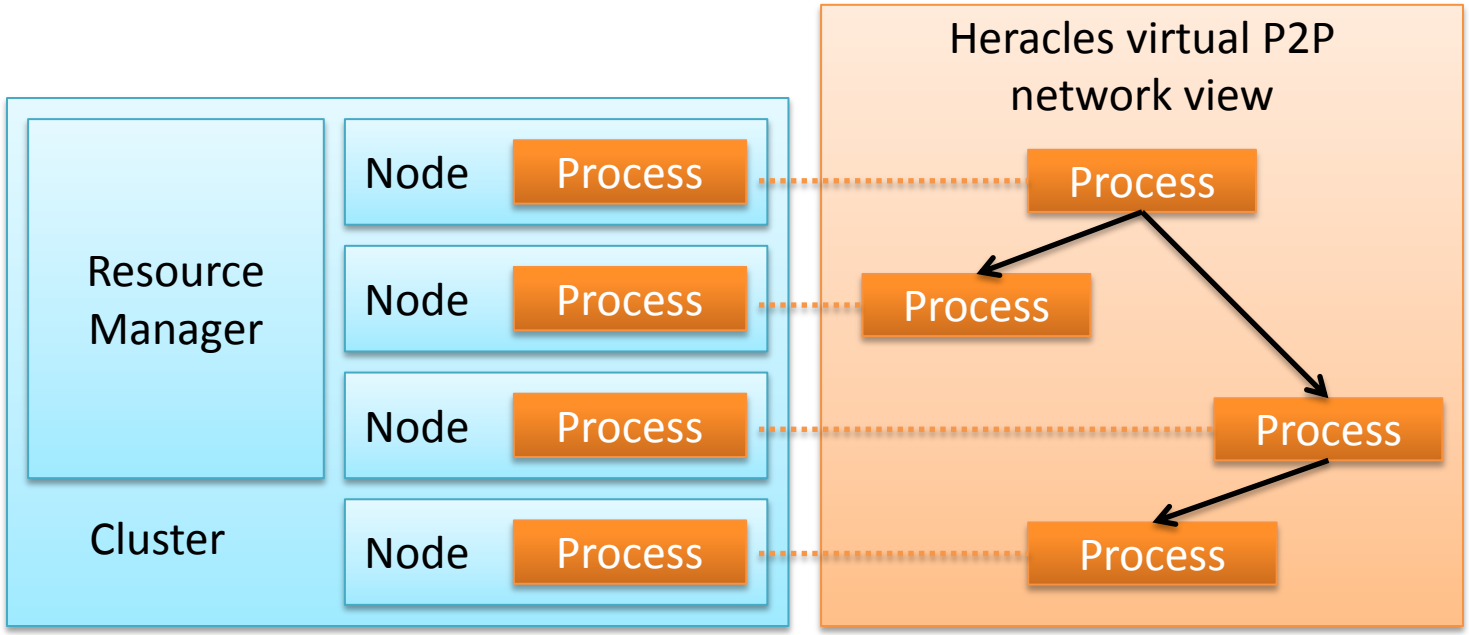
Heracles Structure



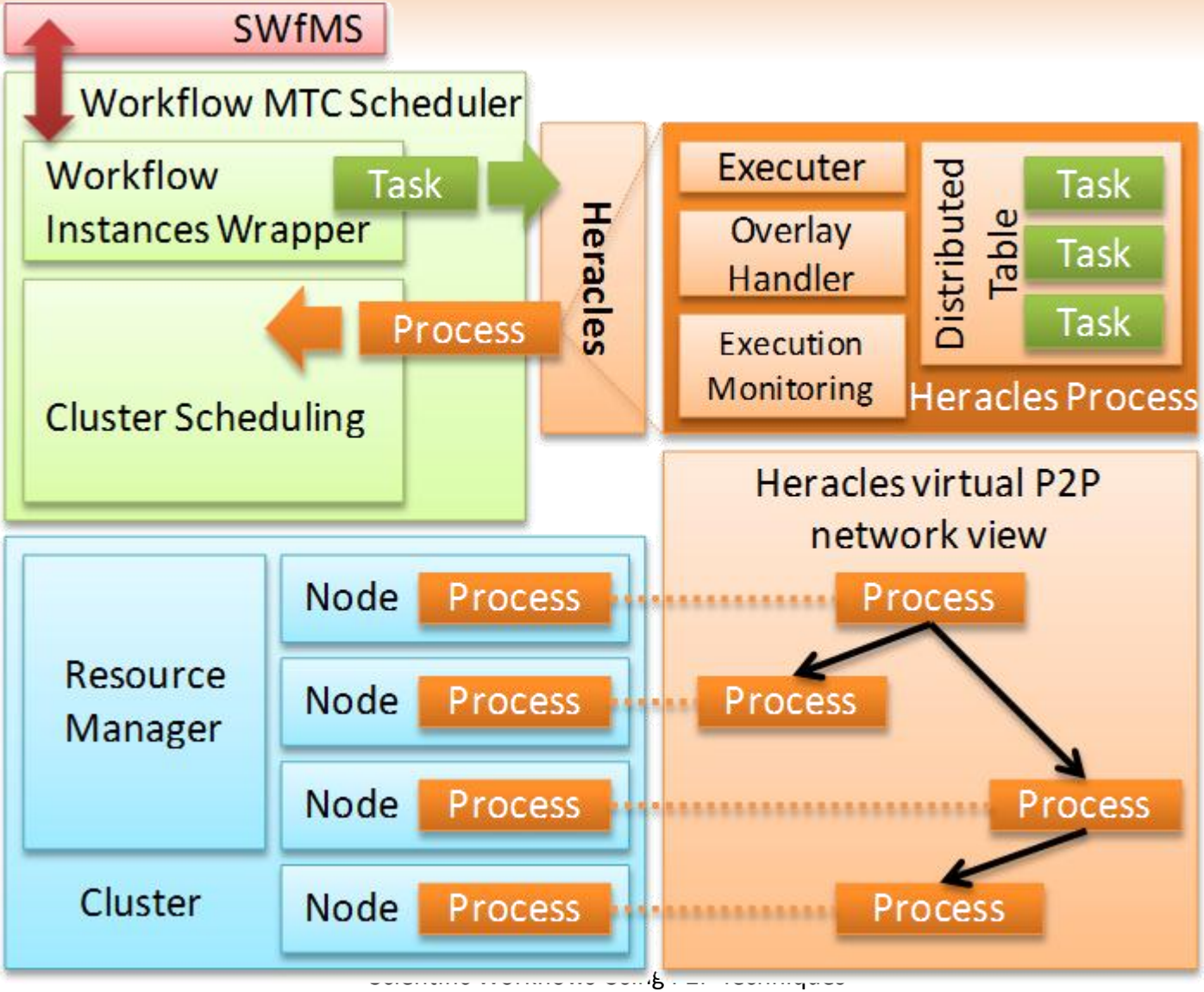
Heracles Structure



P2P view


Heracles virtual P2P network view
Resource Manager
Cluster
Node
Process
Node
Process
Node
Process
Node
Process
Process
Process
Process
Process

Heracles

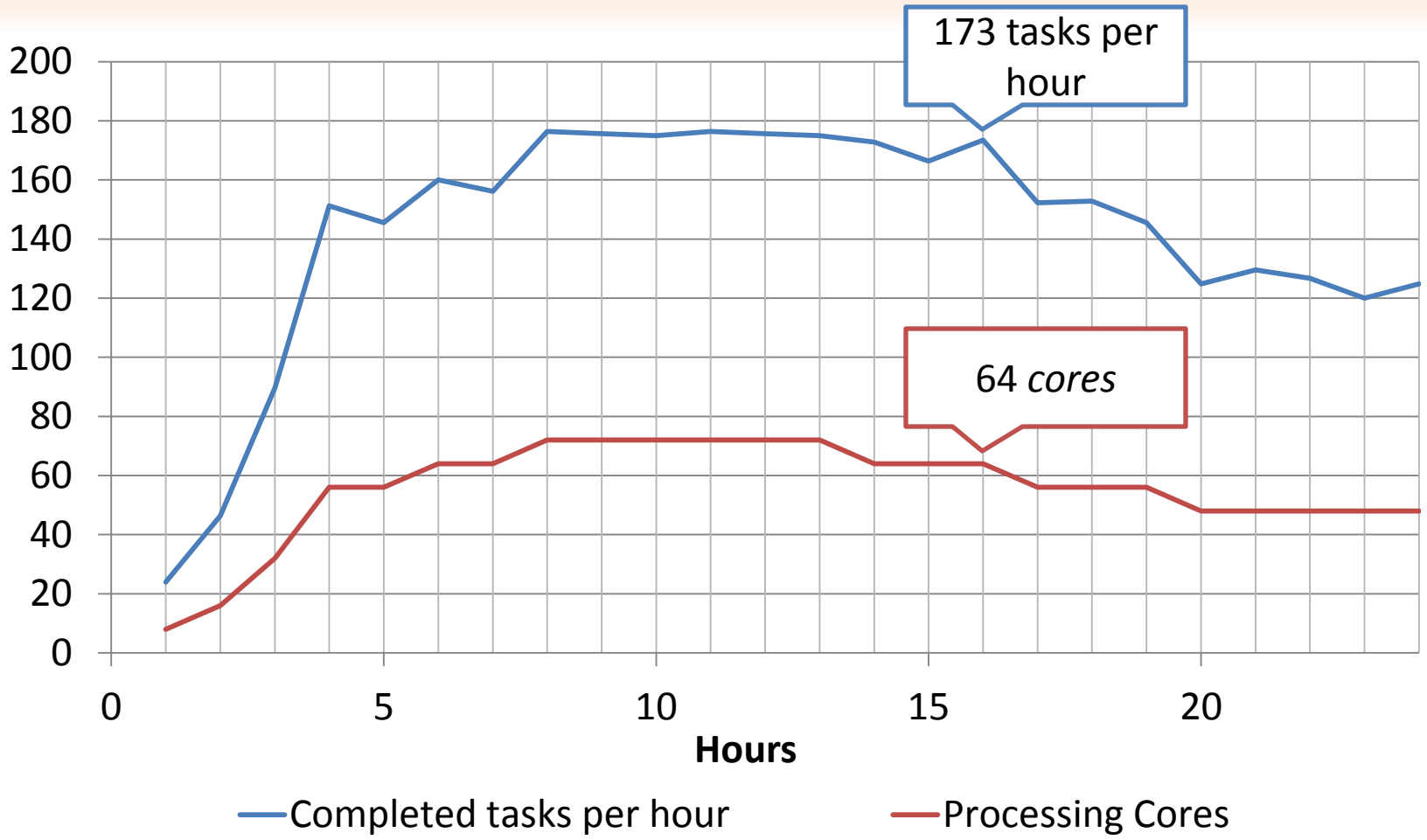

SWfMS
Workflow MTC Scheduler
Workflow Instances Wrapper
Task
Process
Cluster Scheduling
Heracles
Executer
Overlay Handler
Execution Monitoring
Distributed Table
Task
Task
Task
Heracles Process
Heracles virtual P2P network view
Process
Process
Process
Process
Resource Manager
Node
Process
Node
Process
Node
Process
Cluster
Node
Process

Transparency

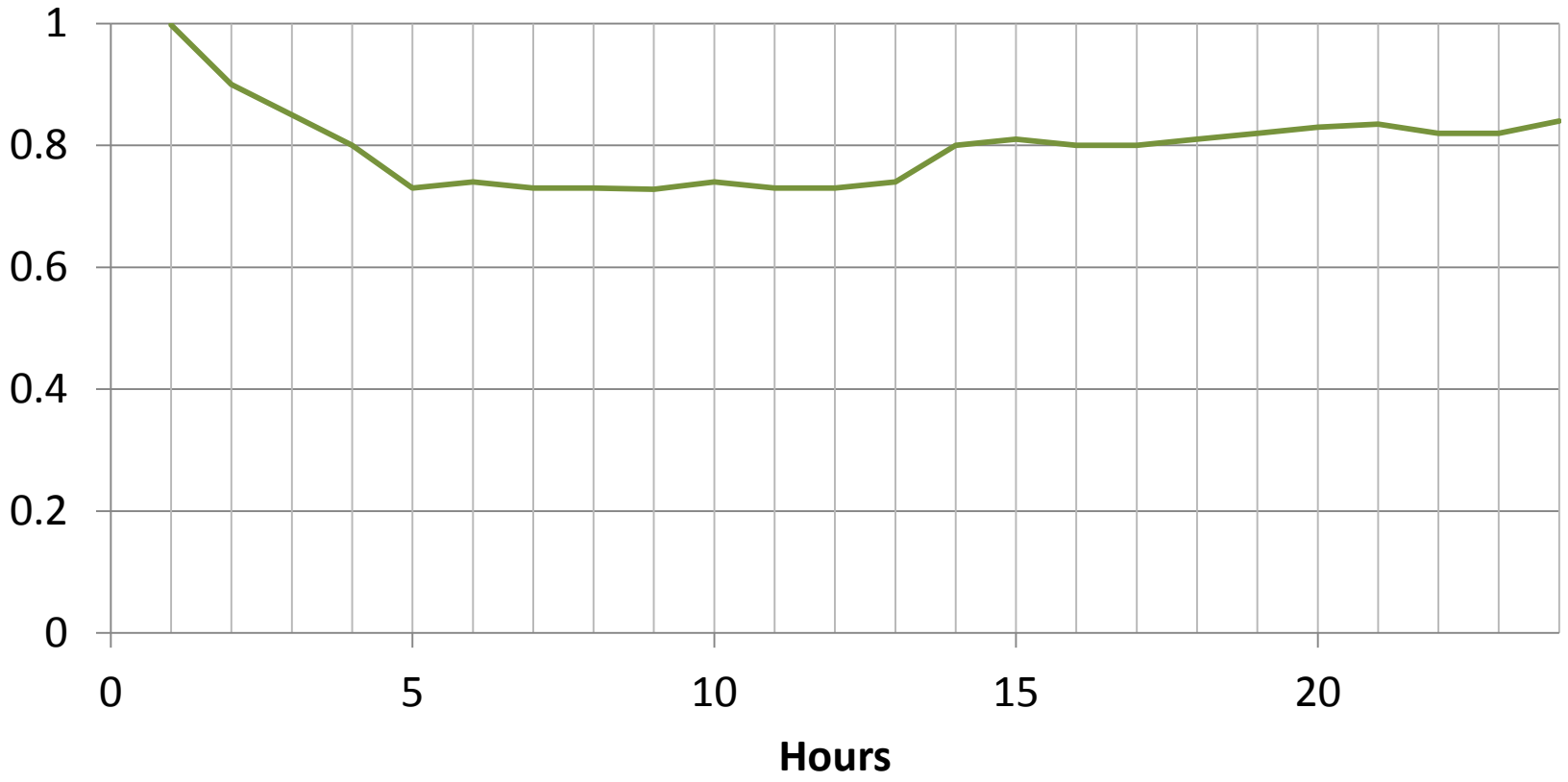
- Setup the *deadline*, not the number of nodes
- Heracles controls the number of involved nodes
  - Execution partial efficiency $$E_p^{'} = \frac{\sum_{i=1}^{n} T_i}{p.T_p}$$
  - Automatically refresh the number of necessary processors

$$p_{new} = \frac{1}{E_p^{'}}\left(\frac{p.(k-n)}{w.C_t^{'}}\right)$$

Dynamic Scheduling example

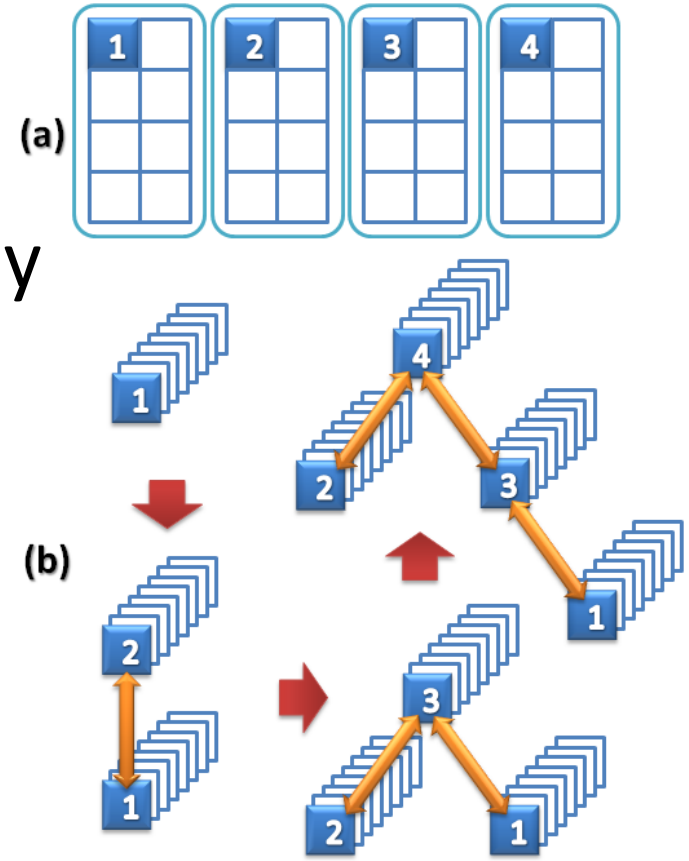


Efficiency



Load Balancing

- Clusters depend on the head node control.
- Tasks can have their autonomy
  - Like P2P dynamic control
- Hierarchical organization
  - Based on P2P hierarchical networks
  - Group leaders
  - Working nodes



Quality of Service

- Job's process failure
  - Hard to reschedule on traditional approaches
  - Manual reschedule not feasible
  - How to address it in the provenance collection
- P2P model can help
  - Autonomy of the nodes
  - Unfinished or failed tasks can be rescheduled
  - Provenance may register all execution attempts or the last execution attempt

When rescheduling?

- Group leaders are responsible for the decision
  - Distributed table data
- Status of the tasks on the distributed table
  - Pending, running or finished
- Average execution time of a task
- To reschedule means to change the status of the task to pending

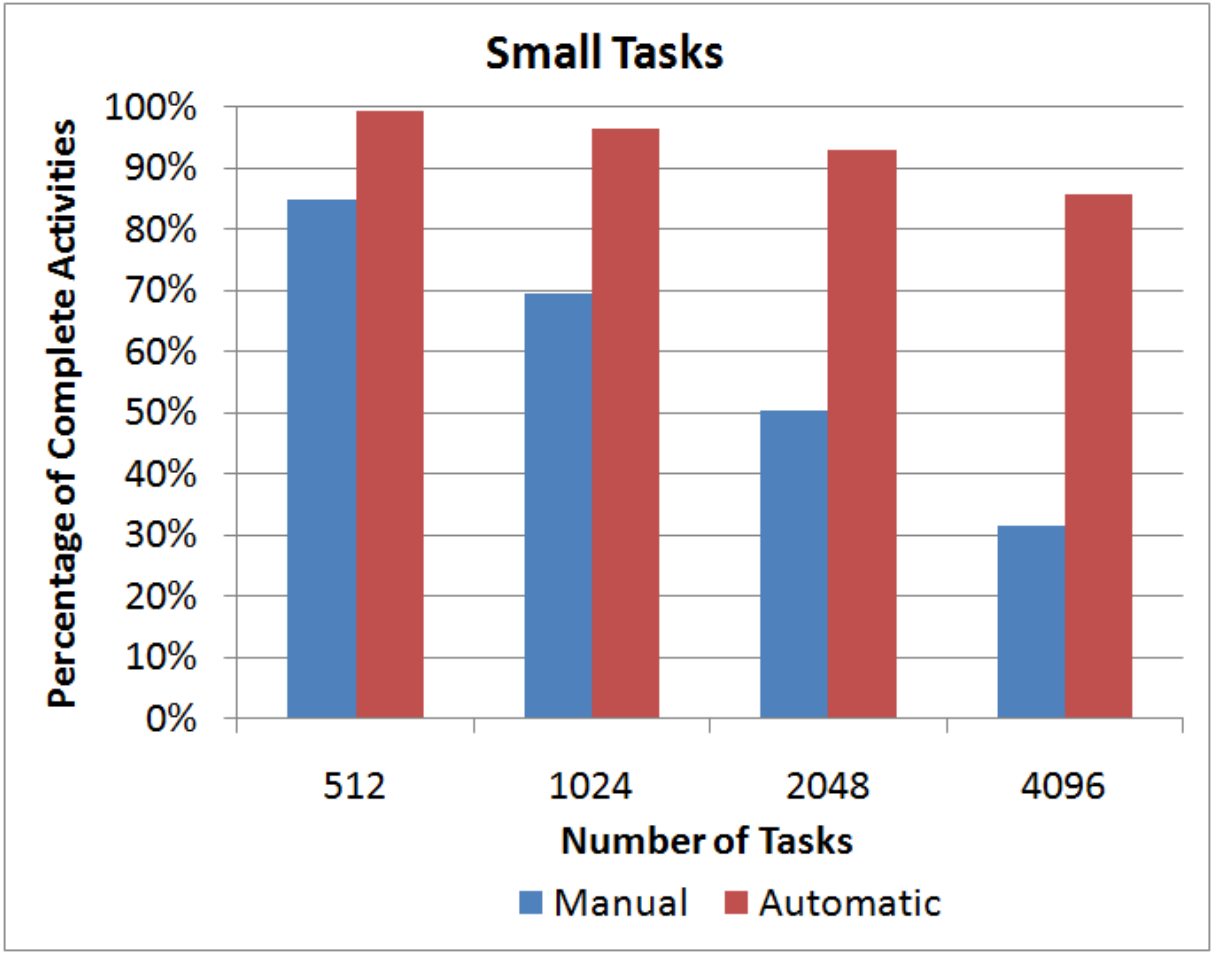
Case Study

- Analyze the impact of churn events on tasks execution on clusters
  - Many workflow activities to be executed
  - Activities are decomposed into tasks
    - Suffer with churn events
  - Activities producing 512, 1024, 2048 and 4096 tasks
  - Tasks is classified as small, medium and large
  - Seven days simulated
  - Calibrated using real experiment data

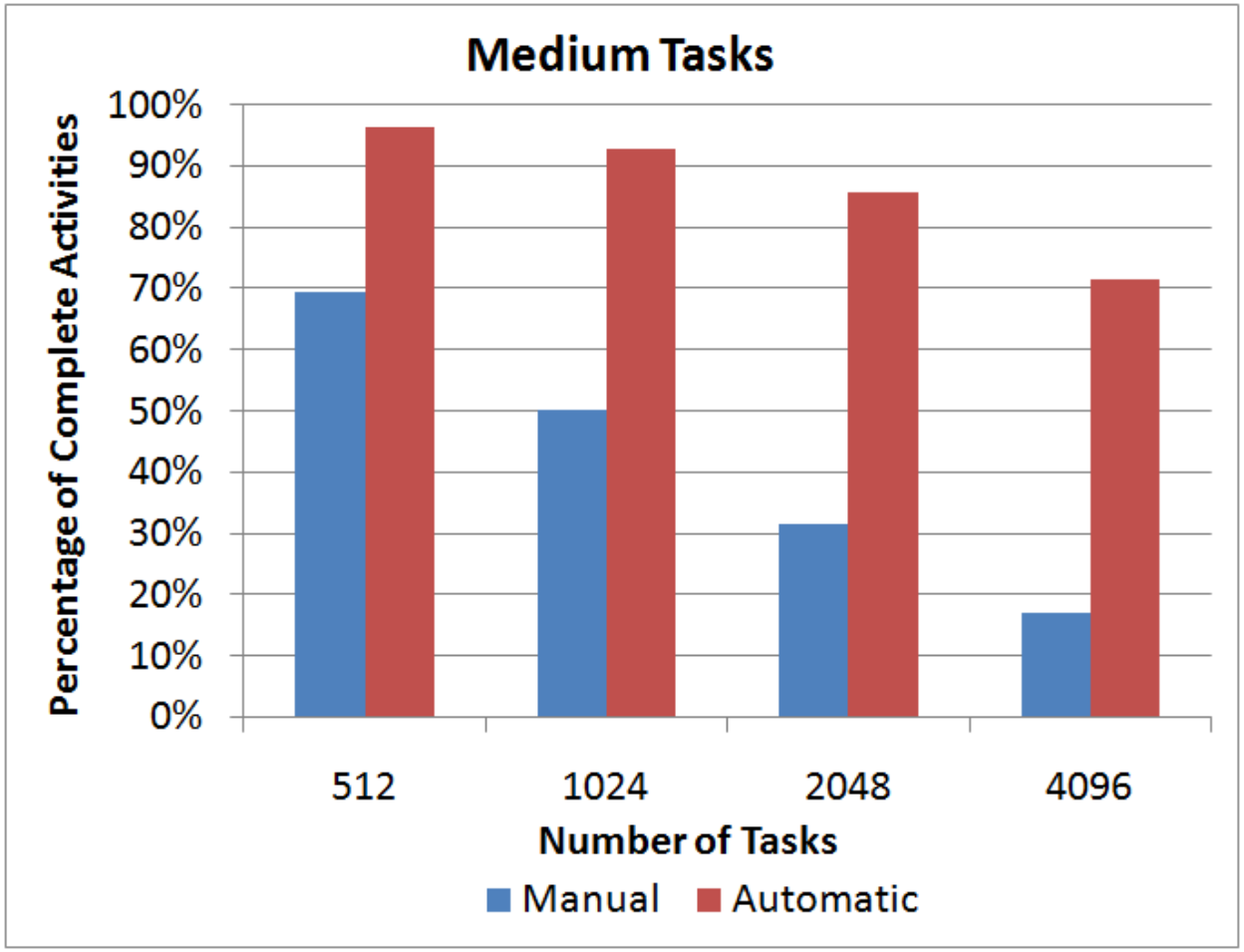
Rescheduling Types

- Manual Rescheduling
  - Scientists checks activity status every twelve hours
  - If a failure happens, all the tasks of the activity are rescheduled
- Automatic Rescheduling
  - Only the task that has failed is rescheduled

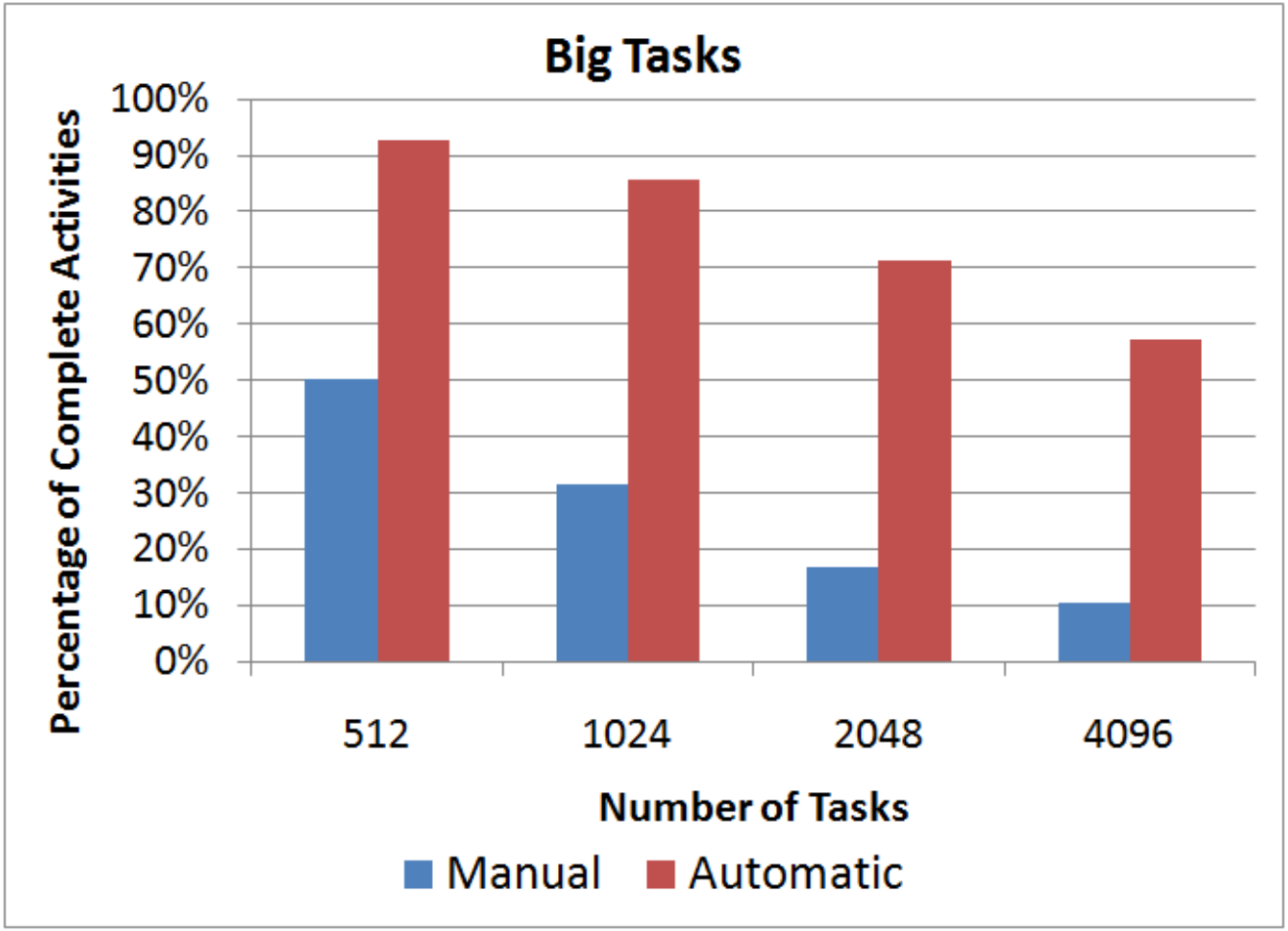
Small Tasks


Small Tasks
Percentage of Complete Activities
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
512
1024
2048
4096
Number of Tasks
Manual
Automatic

Medium Tasks



Big Tasks


Big Tasks
Percentage of Complete Activities
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
512
1024
2048
4096
Number of Tasks
Manual
Automatic

Conclusions

- Empowering scientific experiments execution
  - Scientific Workflow parallelization on huge clusters
  - Many task computing
  - Process failures, poor load balancing, usability issues
- Heracles Approach
  - Transparency, load balance and quality of service
  - Using P2P model on clusters
- Case study showed the gains with automatic rescheduling

Future Work

- Analyze the advantages that MTC schedulers can achieve when using full Heracles approach
- Using Heracles on real experiments
  - Implementing it on real schedulers such as Hydra
- Evaluate other fault tolerant mechanisms such as redundant executions

Acknowledgements

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

INRIA

NACAD
High Performance Computing Center

CAPES

Improving Many-Task Computing in Scientific Workflows Using P2P Techniques

THANK YOU!

COPPE, Federal University of Rio de Janeiro, Brazil

INRIA & LIRMM, Montpellier, France