Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Programming Distributed Computing Platforms with COMPSs

Rosa M. Badia, Javier Conejero, Daniele Lezzi

Workflows & Distributed Computing Group

19/09/2019

13th RES Users Conference

Agenda

14.30 Introduction

14:40 PyCOMPSs syntax

15:20 Overview of COMPSs runtime

15:30 Introduction to dislib

15:40 Break

15:55 Hands-on in MN4:

- PyCOMPSs simple use case in Python
- Use case calling external binaries
- dislib use case

17:00 Closing

www.bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Introduction

BSC vision on programming models

Applications

PM: High-level, clean, abstract interface

Power to the runtime

API

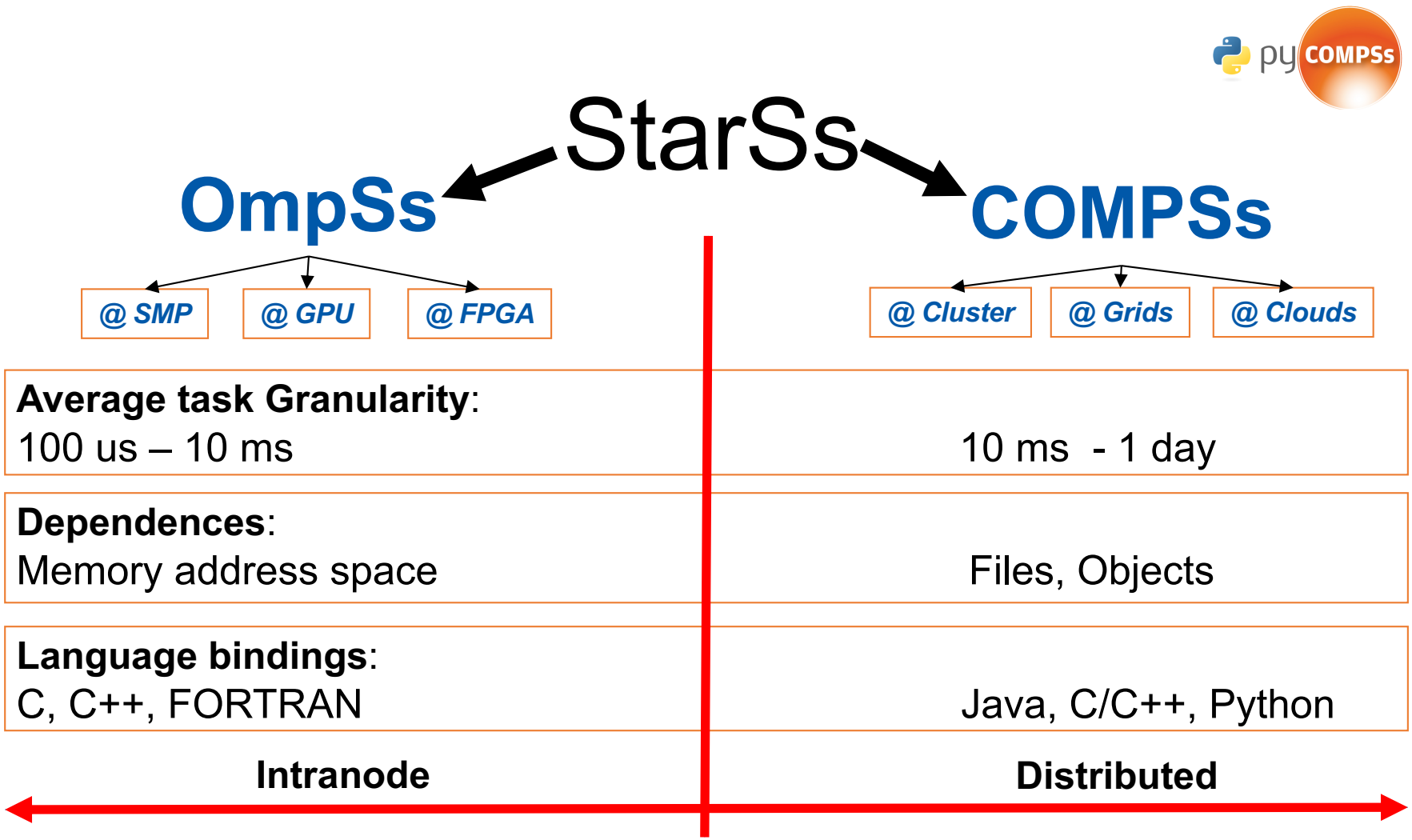
Program logic independent of computing platform

General purpose
Task based
Single address space

Intelligent runtime, parallelization, distribution, interoperability

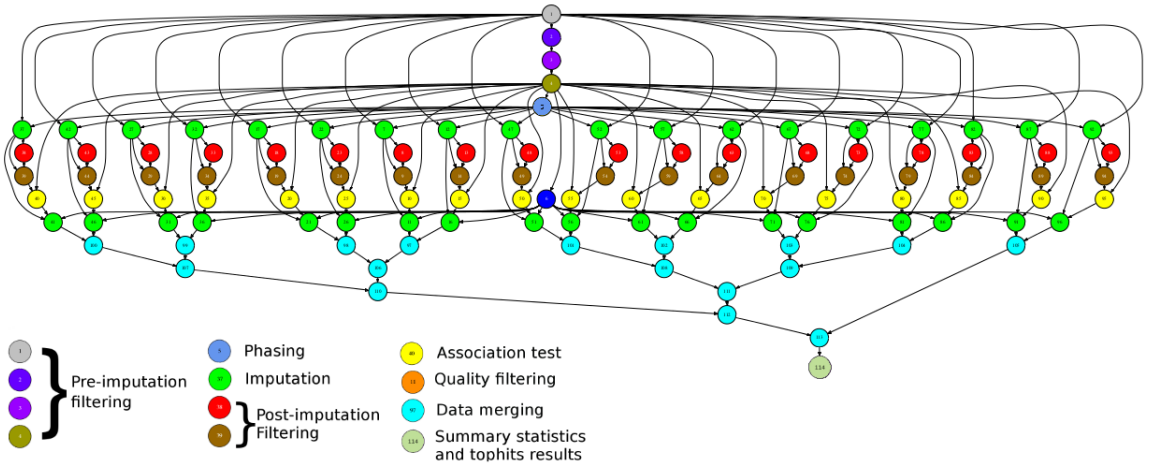
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

BSC vision on programming models



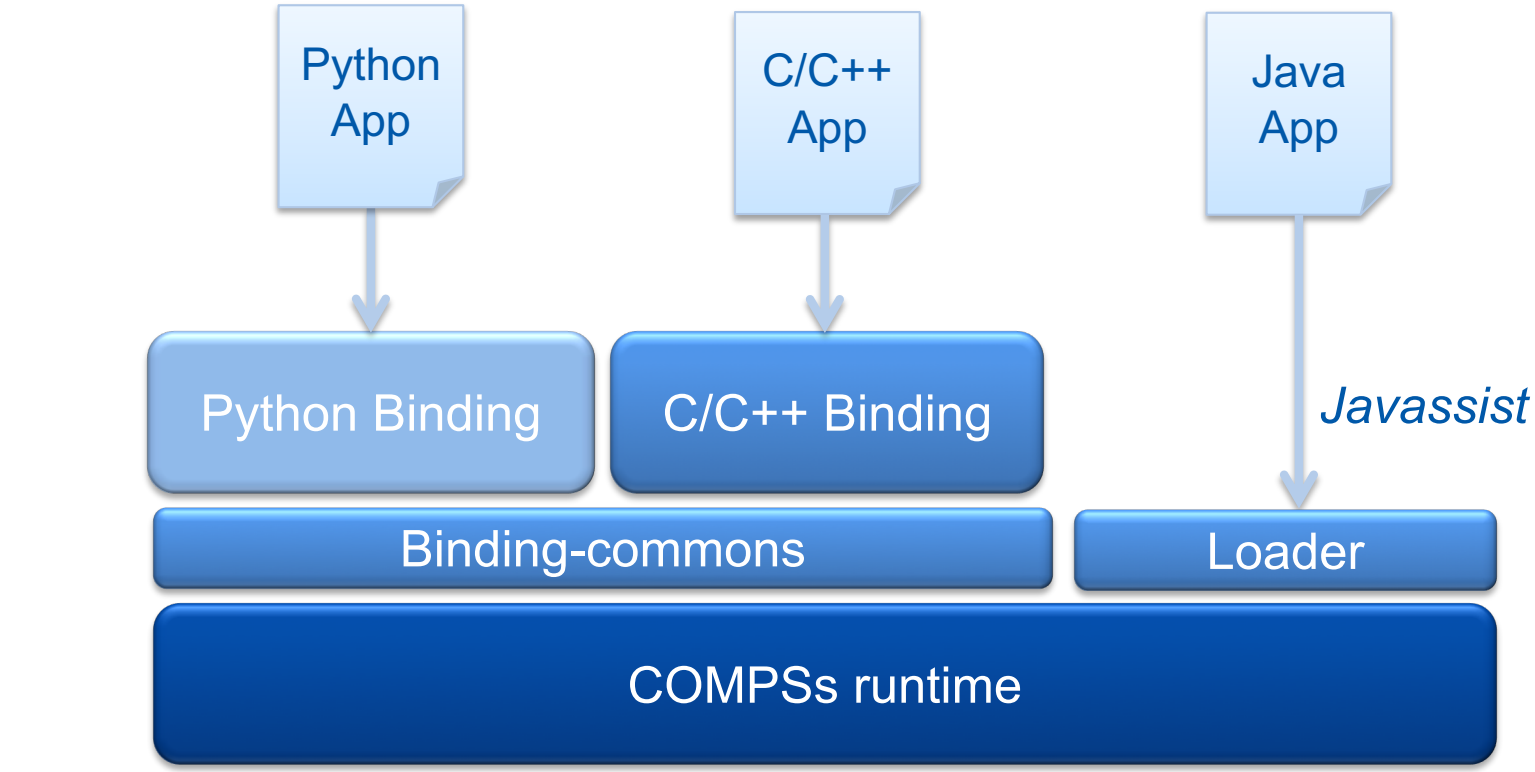
Programming with PyCOMPSs/COMPSs

- Sequential programming, parallel execution
- General purpose programming language + annotations/hints
  - To identify tasks and directionality of data
- Builds a task graph at runtime that express potential concurrency
- Offers a shared memory illusion to applications in a distributed system
  - The application can address larger data storage space: support for Big Data apps
  - Support for persistent storage
- Agnostic of computing platform
  - Enabled by the runtime for clusters, clouds and container managed clusters
- Available in MN4
  - module load COMPSs



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

COMPSs Architecture



Grid

Cluster

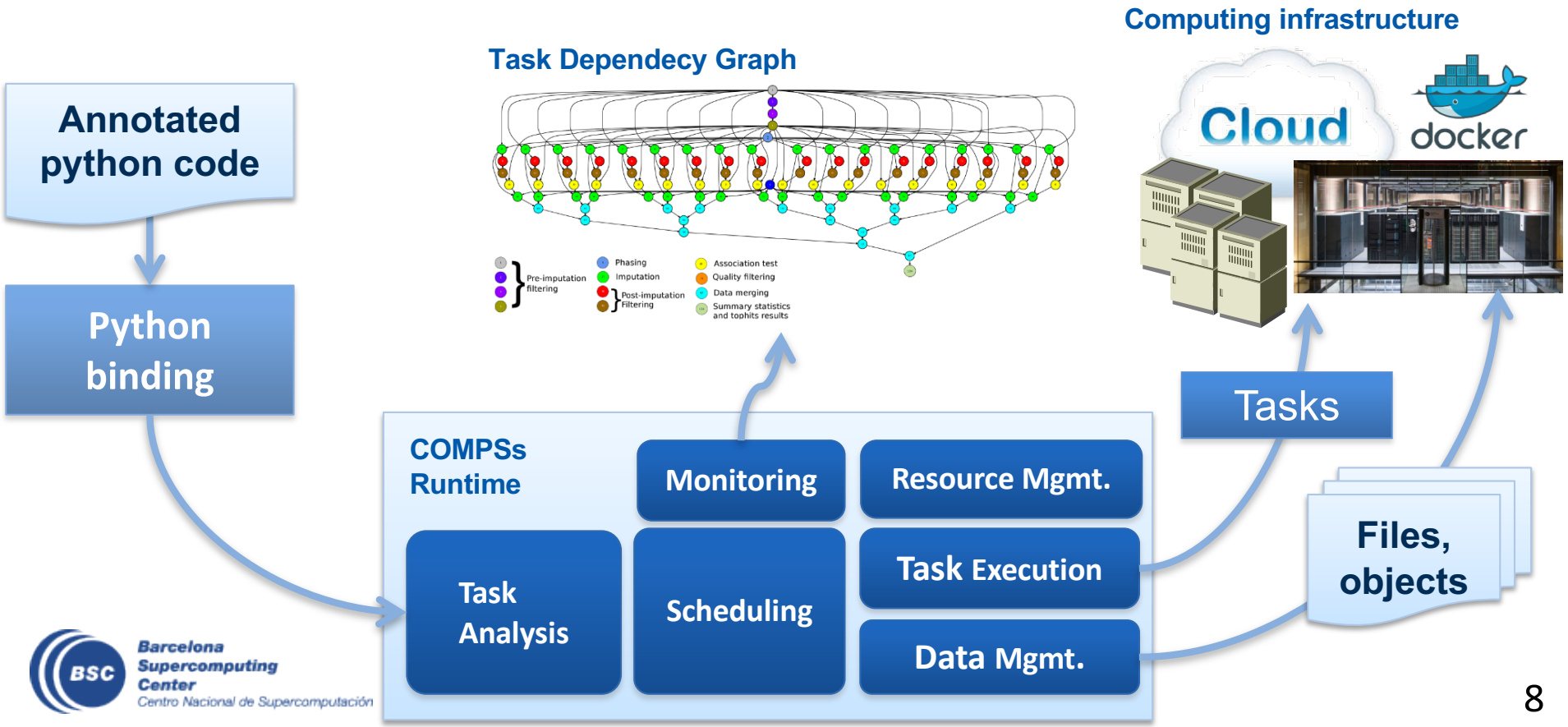
Cloud

Containers

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

PyCOMPSs/COMPSs runtime

- PyCOMPSs/COMPSs applications executed in distributed mode following the master-worker paradigm
  - Description of computational infrastructure in an XML file
- Sequential execution starts in master node and tasks are offloaded to worker nodes
- All data scheduling decisions and data transfers are performed by the runtime



www.bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Python Syntax (PyCOMPSs)

Tutorial with Jupyter notebooks

- Instructions in:
  - https://github.com/bsc-wdc/notebooks

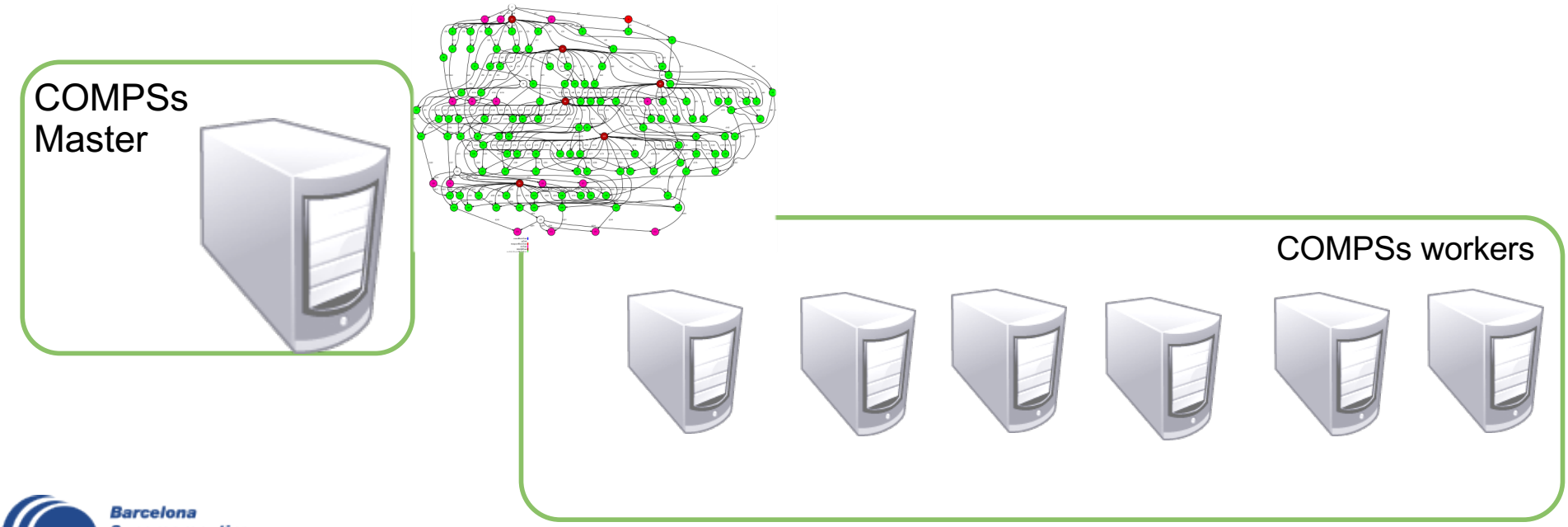
www.bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

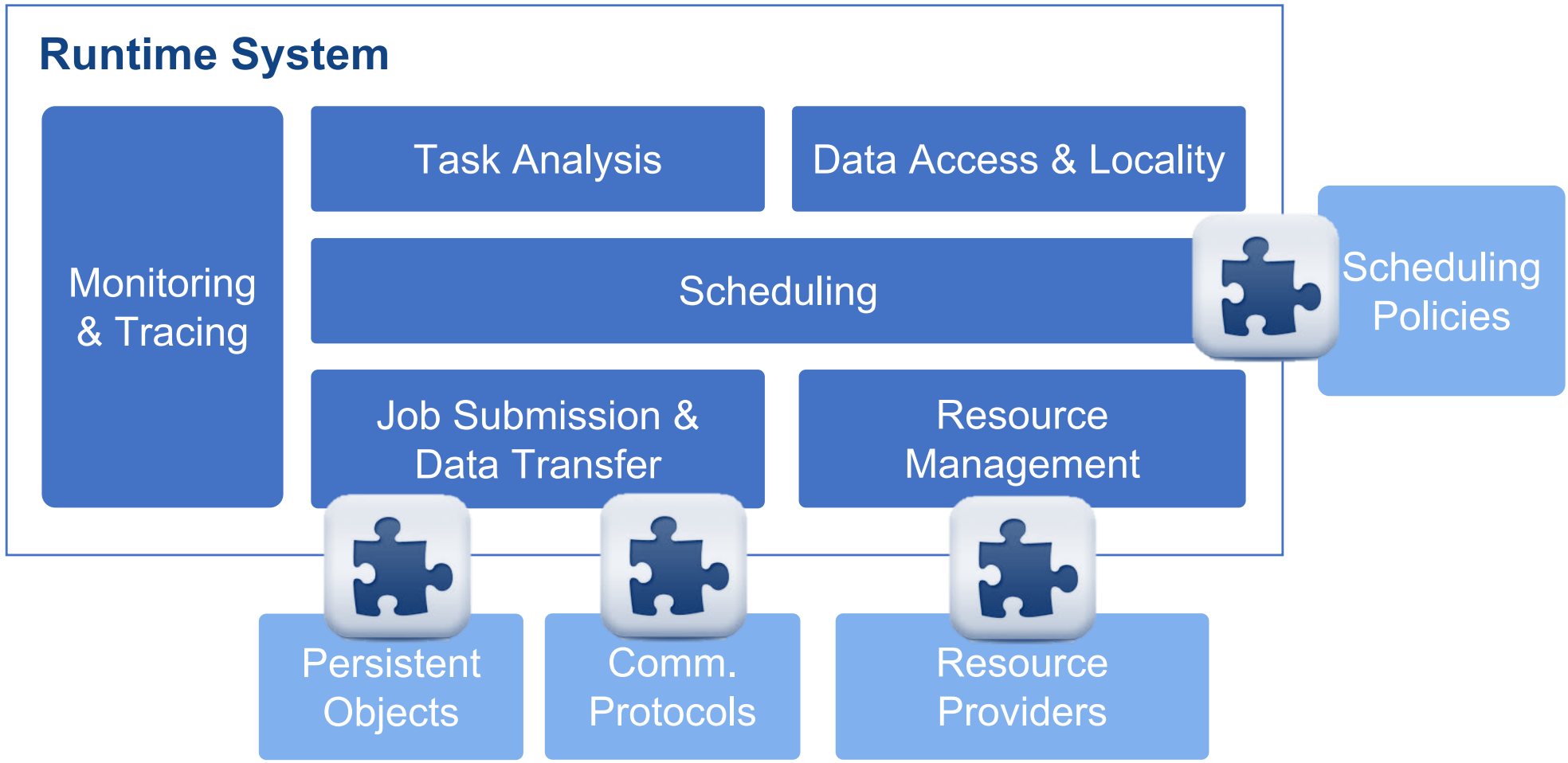
COMPSs runtime and execution environment

Execution environment

- COMPSs runtime is able to execute in distributed computing platforms (clusters, clouds, contained manager clusters)
- Main program + runtime started in master node
  - Takes care of tasks scheduling, data transfers, etc
- Tasks executed in worker nodes



Runtime Architecture

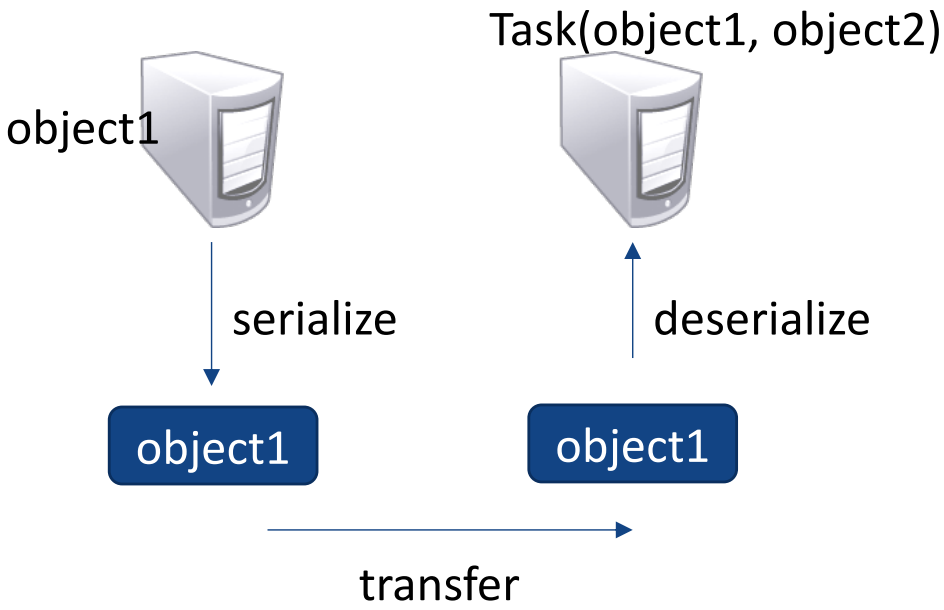


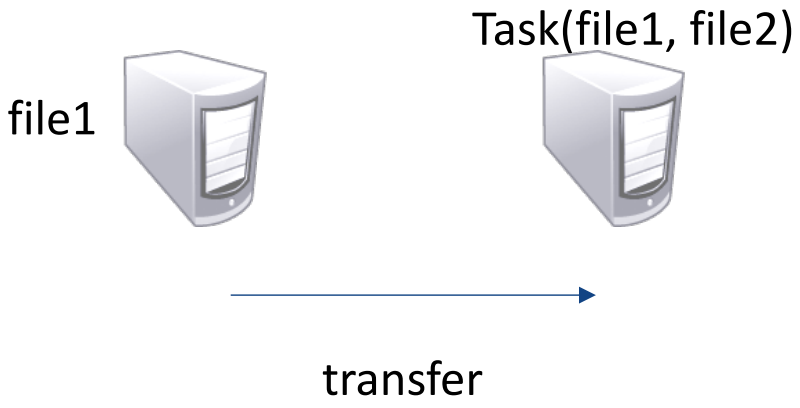
Data management in COMPSs

- COMPSs runtime offers the view of a single memory space and storage system
  - Can address memory spaces much larger than the initial space available in a single node
  - This larger memory space is made available to the application in a transparent way
- Data management
  - Objects can be either created by the main program or tasks
  - When accessed by tasks executed in different worker nodes, the runtime will be in charge of transferring the data between them
  - Files and/or objects can be renamed/versioned
  - Renaming enables further parallelism
  - Versioning reduces the number of required transfers
- Runtime supports shared and distributed filesystems
  - In the first case, no file transfers are required

Data management in COMPSs

- Serialization
  - Due to different address spaces between different nodes, objects in memory need to be transferred
  - Before being transferred are serialized
  - Only for objects, not for files
  - For Python, Pickle or Dill libraries are used





Runtime System

Application

Task Selection Interface

How do I select the execution platform?

Grid

Cluster

Cloud

Execution Environments Configuration

Specifies:
- Resources and Project files
- Scheduler, Comm. Adaptor
- Persistent object storage

runcompss options

Infrastructure Description
- Describe the available resource in the infrastructure
- Describe Cloud Providers: Images and VM Templates

Runtime System

Exec. Mngmt & Data Transfers

Persistent Object Storage: DataClay, Hecuba, Redis

Communication Adaptor: NIO, GAT

Schedulers

Resources

Cloud Connector: jClouds, rOCCI

Slurm, Docker, Mesos

resources.xml

project.xml

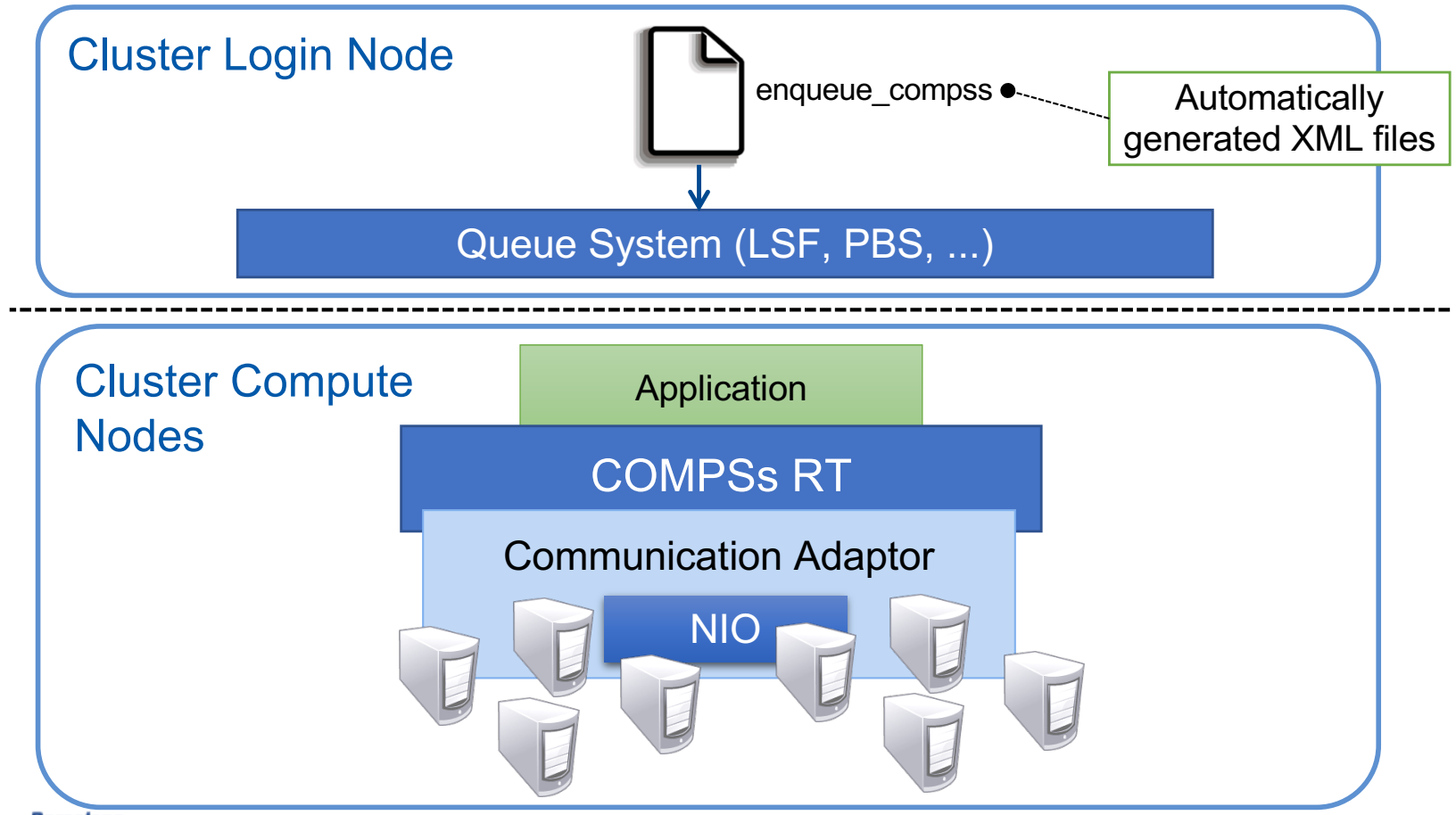
Master-Worker Comm. Mechanism
- GAT: Restrictred environments (only ssh access) and Grid Middleware
- NIO: Efficient Persistent workers implementation in controlled and secured environments

Resource Scalability
- Provide interaction with resource providers to create and destroy new computing resources

Application Exec. Desc.
- Selection of resources
- Application Code Location
- Working directory
- Provided as execution command argument

COMPSs@Cluster

- Execution divided in two phases
  - Launch scripts queue a whole COMPSs app execution
  - Actual execution starts when reservation is obtained



COMPSs in a Cluster

- Use of enqueu_compss command (instead of runcompss)
- Generates project.xml and resources.xml
- Launches application in the allocation

```
enqueue_compss \
  --exec_time=10 \
  --num_nodes=5 \
  --tasks_per_node=16 \
  --master_working_dir=. \
  --worker_working_dir=scratch \
  --lang=python \
  --comm=integratedtoolkit.nio.master.NIOAdaptor \
  --tracing=true \
  --graph=true \
  /home/bsc19/bsc19776/SC16/wordcount/wc_merge.py \
 /gpfs/projects/bsc19/COMPSs_AP PS/wordcount/data/dataset_64f_16mb
```

- Type “enqueue_compss” to see help and options

Configuration with Job scheduler: Resources Specification

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ResourcesList>
 <SharedDisk Name="gpfs" />
 <SharedDisk Name="gpfs2" />
 <ComputeNode Name="s12r1b62-ib0">
  <Processor Name="MainProcessor">
   <ComputingUnits>24</ComputingUnits>
   <Architecture>Intel</Architecture>
   <Speed>2.6</Speed>
  </Processor>
  <Memory>
   <Size>92</Size>
  </Memory>
  <OperatingSystem>
   <Type>Linux</Type>
   <Distribution>SMP</Distribution>
   <Version>3.0.101-0.35-default</Version>
  </OperatingSystem>
  <Software>
   <Application>JAVA</Application>
   <Application>PYTHON</Application>
   <Application>EXTRAE</Application>
   <Application>COMPSS</Application>
  </Software>
…
```

Generated automatically by enqueue_compss command in MN4

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

```
    ...
 <Adaptors>
      <Adaptor Name="es.bsc.compss.nio.master.NIOAdaptor">
        <SubmissionSystem>
          <Interactive/>
        </SubmissionSystem>
        <Ports>
          <MinPort>43001</MinPort>
          <MaxPort>43002</MaxPort>
          <RemoteExecutionCommand>none</RemoteExecutionCommand>
        </Ports>
      </Adaptor>
      …
    </Adaptors>
    <SharedDisks>
      <AttachedDisk Name="gpfs">
        <MountPoint>/gpfs/</MountPoint>
      </AttachedDisk>
      <AttachedDisk Name="gpfs2">
        <MountPoint>/.statelite/tmpfs/gpfs/</MountPoint>
      </AttachedDisk>
    </SharedDisks>
  </ComputeNode>
  <ComputeNode Name="s12r1b63-ib0">
    <Processor Name="MainProcessor">
      <ComputingUnits>48</ComputingUnits>
      <Architecture>Intel</Architecture>
      <Speed>2.6</Speed>
    </Processor>
    <Memory>
      <Size>92</Size>
    </Memory>
    <OperatingSystem>
      …

</ResourcesList>
```

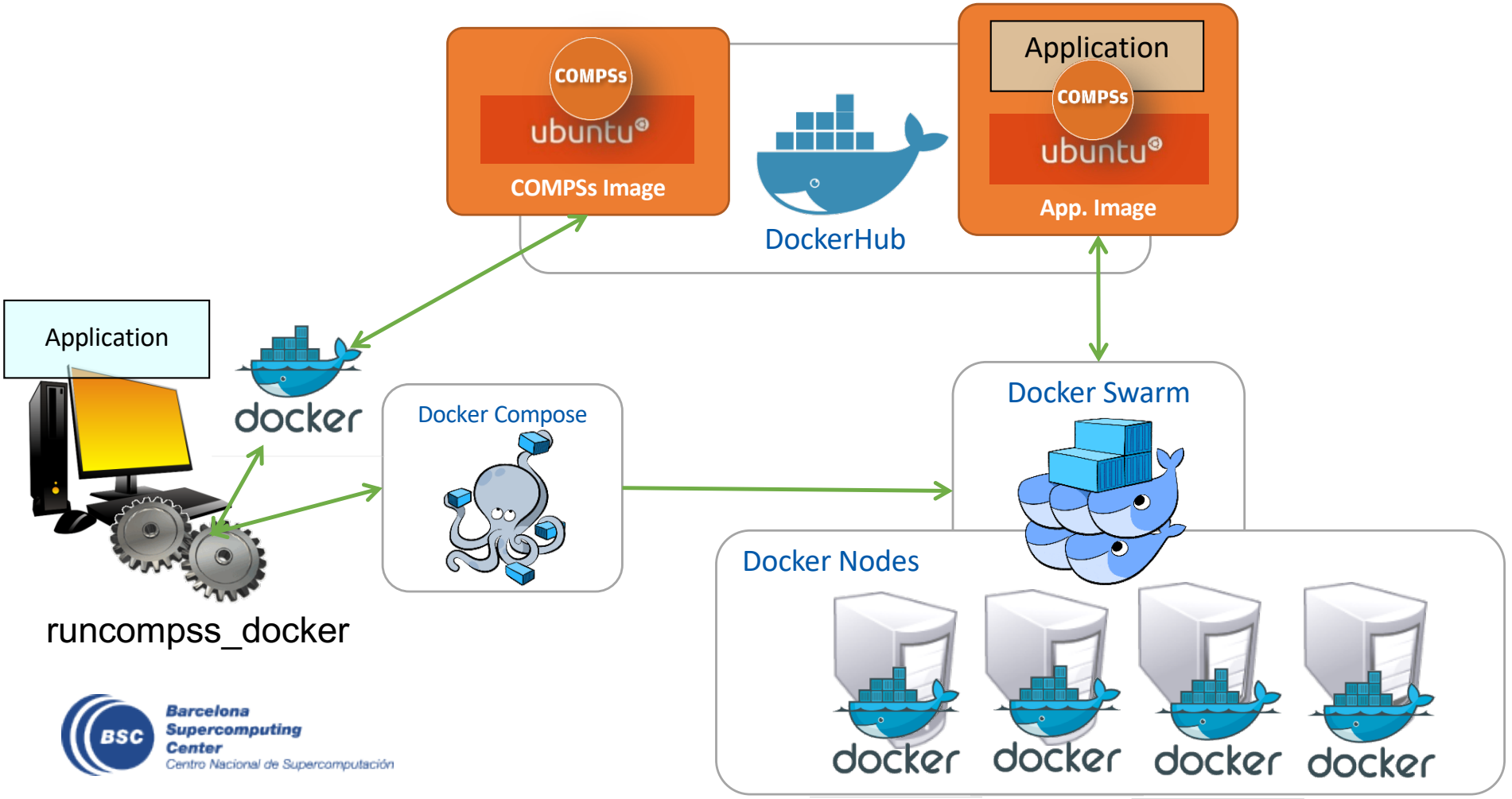
Configuration with Job scheduler: Project Specification

```
<Project>
  <MasterNode>
    <SharedDisks>
      <AttachedDisk Name="gpfs">
        <MountPoint>/gpfs/</MountPoint>
      </AttachedDisk>
      <AttachedDisk Name="gpfs2">
        <MountPoint>/.statelite/tmpfs/gpfs/</MountPoint>
      </AttachedDisk>
    </SharedDisks>
  </MasterNode>

  <ComputeNode Name="s12r1b62-ib0">
    <InstallDir>/apps/COMPSs/2.5</InstallDir>
    <WorkingDir>/home/nct00/nct00002/examples/clustering_comparison/tmp.Ynl9AcAh8D</WorkingDir>
    <Application>
      <LibraryPath>/home/nct00/nct00002/examples/clustering_comparison</LibraryPath>
    </Application>
  </ComputeNode>

 <ComputeNode Name="s12r1b63-ib0">
    <InstallDir>/apps/COMPSs/2.5</InstallDir>
    <WorkingDir>/home/nct00/nct00002/examples/clustering_comparison/tmp.Ynl9AcAh8D</WorkingDir>
    <Application>
      <LibraryPath>/home/nct00/nct00002/examples/clustering_comparison</LibraryPath>
    </Application>
  </ComputeNode>

      ...
</Project>
```

Generated automatically by enqueue_compss command for MN4

COMPSs with Docker

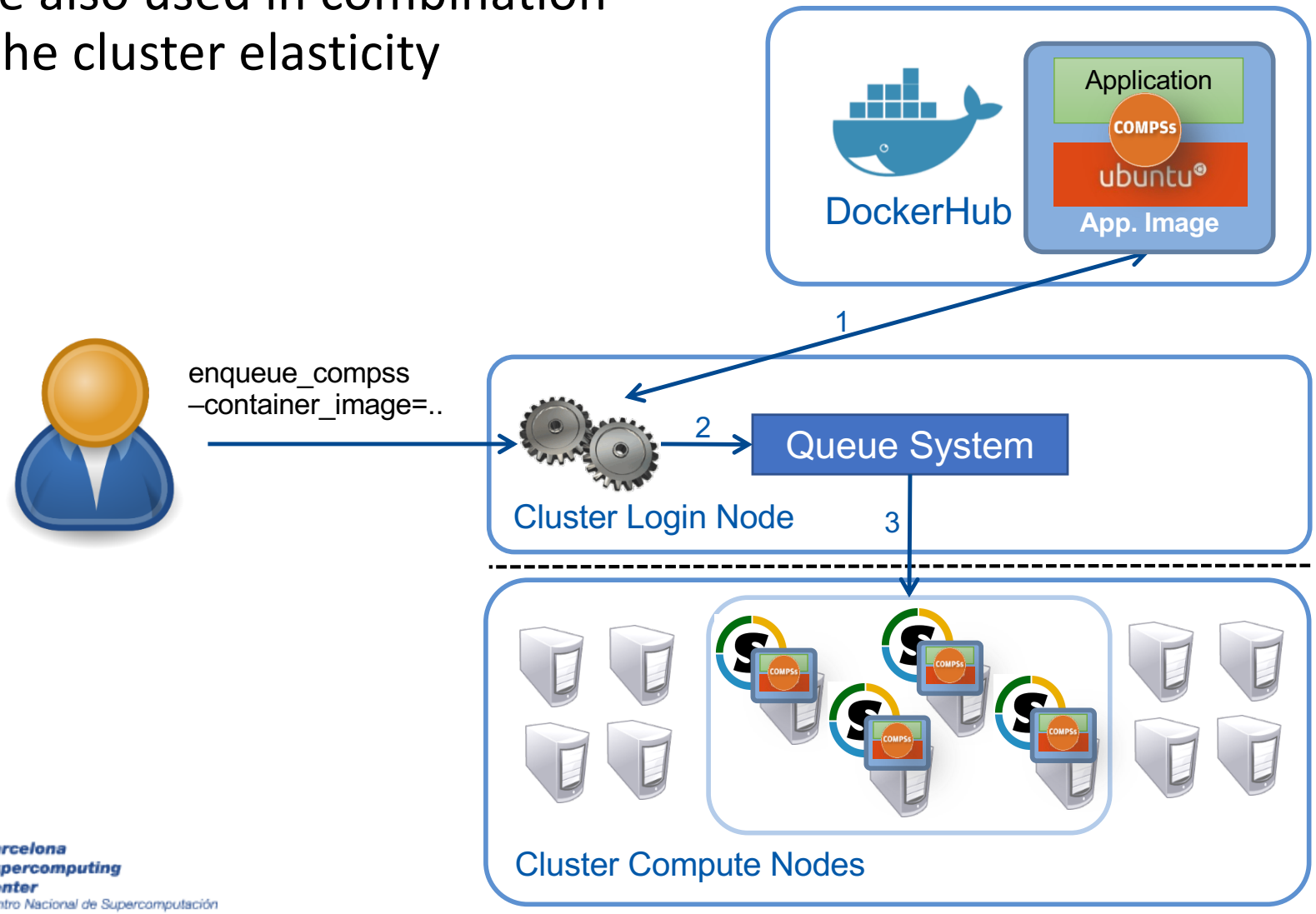
- Keep as transparent for the user as possible
- Same as running a local COMPSs application (runcompss command)
- Deploy applications as a set of docker container



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

COMPSs@Singularity

- Execute applications from a container image in HPC cluster
- Can be also used in combination with the cluster elasticity



www.bsc.es

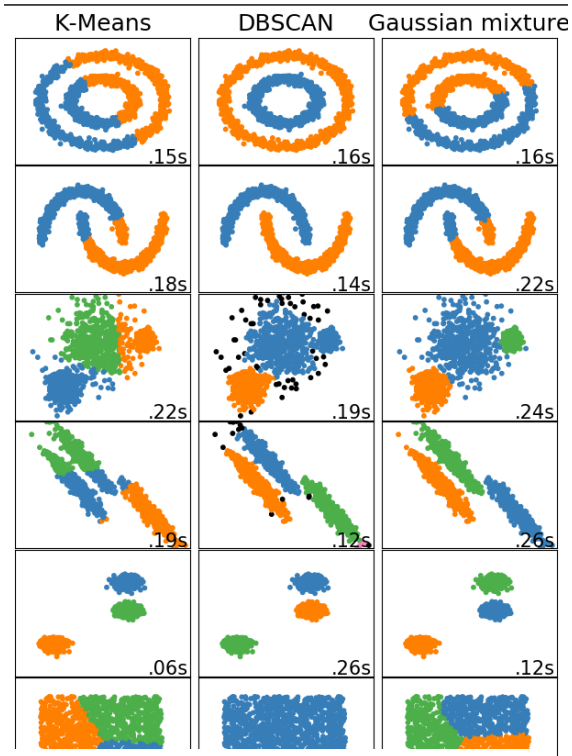
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

dislib

dislib

- dislib: Collection of machine learning algorithms developed on top of PyCOMPSs
  - Unified interface, inspired in scikit-learn (fit-predict)
  - Unified data acquisition methods and using an independent distributed data representation
  - Parallelism transparent to the user – PyCOMPSs parallelism hidden
  - Open source, available to the community



dislib.bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

FUJITSU

DISLIB | Distributed Computing Library

dislib library

dislib

```
from dislib.cluster import KMeans
from dislib.data import load_txt_file

file_ = "~/my_input_file.csv"

dataset = load_txt_file(file_, 10, 780)

kmeans = KMeans()
kmeans.fit(dataset)
centers = kmeans.centers
```

scikit-learn

```
from sklearn.cluster import KMeans
import numpy as np

file_ = "~/my_input_file.csv"

arr = np.load_txt(file_, delimiter=",")

kmeans = KMeans()
kmeans.fit(arr)
centers = kmeans.centers
```

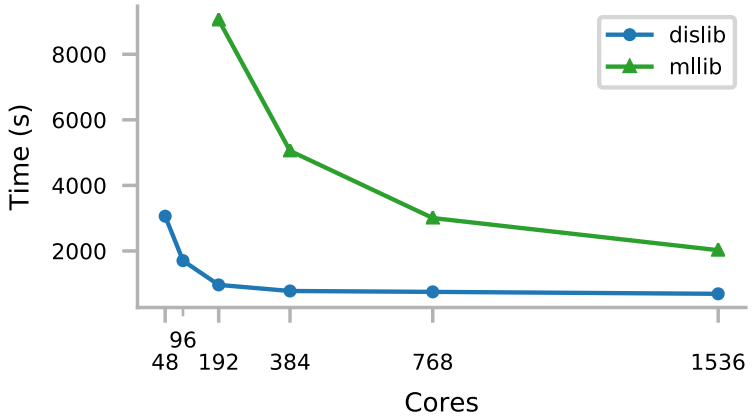
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

FUJITSU

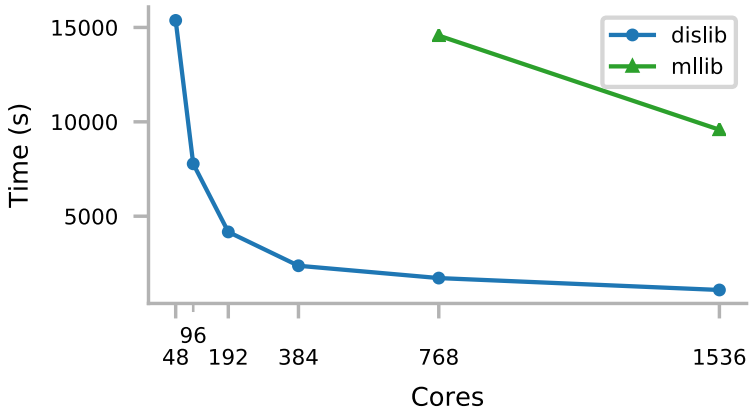
DISLIB | Distributed Computing Library

Comparison with DASK and MLib

- Gaussian mixtures (execution time) – 100 features, 50 components



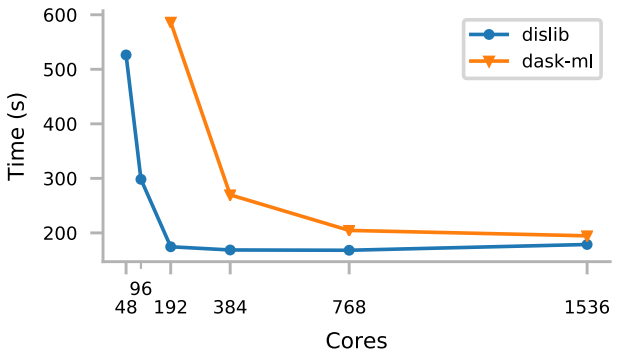
100 million samples



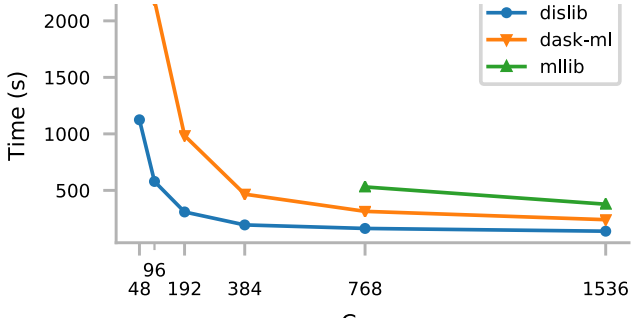
500 million samples

Comparison with DASK and MLib

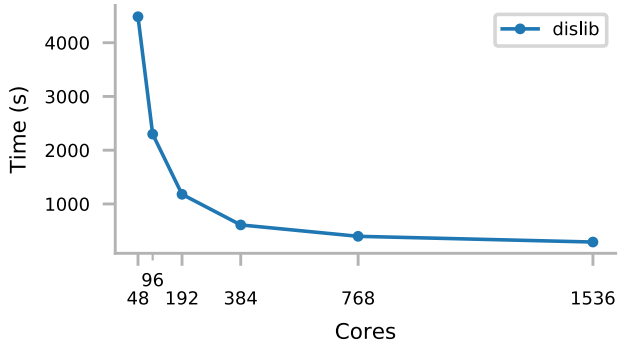
- K-means (execution time) –



1 billion samples with low granularity (50 features and 50 clusters)



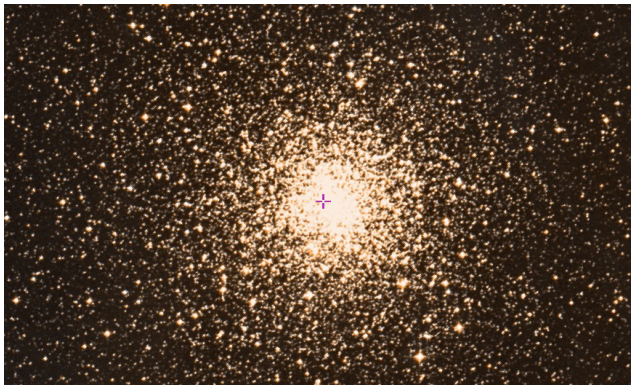
500 million samples with high granularity (100 features and 500 clusters)

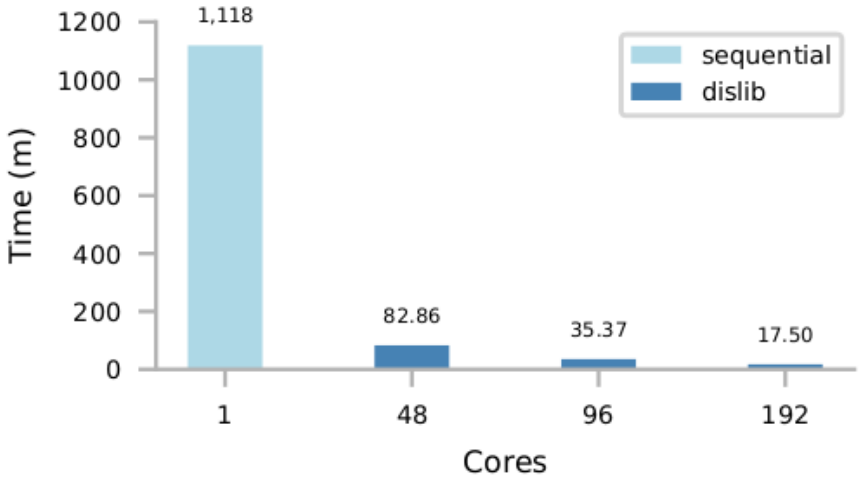


2 billion samples with high granularity (100 features and 500 clusters)

Looking for objects in the sky with dislib

- Gaia satellite data: Sample scientific application:
  - Looking for open clusters in the sky with DBSCAN clustering
  - Subset of astrometric data from 2.5 million stars
    - Total data is 10^9 stars
  - Execution of 6,145 DBSCANs in parallel





Demo dislib

- Jupyter notebook

We are hiring!

COMPSs runtime developer - Junior developer (RE1)

Job Reference	Position	Closing Date
243_19_CS_WDC_RE1	COMPSs runtime developer - Junior developer (RE1)	Sunday, 13 October, 2019

Reference: 243_19_CS_WDC_RE1
Job title: COMPSs runtime developer - Junior developer (RE1)

Barcelona Supercomputing Center
Centro Nacional de Supercomputación


Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación
EXCELENCIA
SEVERO
OCHOA
THANK YOU!
support-compss@bsc.es
www.bsc.es