



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación



Programming Distributed Computing Platforms with COMPSs

Rosa M. Badia, Javier Conejero

Workflows & Distributed Computing Group

07/02/2023

Winter School, Barcelona

Supercomputers Hands-on



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Supercomputers Hands-on

- Execution in MareNostrum 4
- Exercise
- Tracing Analysis Overview

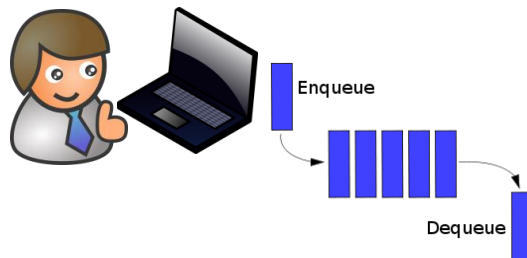
Execution in MareNostrum 4

- How to connect to MareNostrum?
 - `ssh -X nct01XXX@mn1.bsc.es`
- Load COMPSs:
 - `export COMPSs_PYTHON_VERSION=3`
 - `module load COMPSs/3.1`
- Where is the source code?
 - `cd`
 - `cp -r /gpfs/home/nct00/nct00016/source .`
- Available editors
 - `vi`
 - `emacs`



Job submission

- Submit PyCOMPSs jobs to MareNostrum 4:
 - All jobs should be submitted to the queuing system (SLURM)
 - We will use the **pycompss-cli**
 - Available after loading the COMPSs module
 - Useful commands:
 - `squeue` – This command shows the status of the job.
 - `scancel jobId` – This command kills a job with id 'jobId'.



Execution in MareNostrum 4 - HandsOn

- Submission sample (launch.sh):

```
#/bin/bash

export COMPSS_PYTHON_VERSION=3
module load COMPSS/3.1

pycompss job submit \
  --qos=training \
  --num_nodes=2 \
  --exec_time=10 \
  --reservation=NotOnlyFLOPS23-day2 \
  --lang=python \
  --tracing=true \
  --graph=true \
  /path/to/application.py parameter1 parameter2 ... parameter N
```

Load COMPSSs module
Main PyCOMPSSs command
PyCOMPSSs parameters
Application Parameters

- Parameters:
 - **num_nodes**: amount of nodes where to execute (1 master + 1 worker)
 - **exec_time**: maximum execution time (wall time)
- How to execute with PyCOMPSSs?
 - **chmod 755 launch.sh**
 - **./launch.sh**

Guided demo (kmeans)



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación

Exercise 1

Parallelize

Wordcount

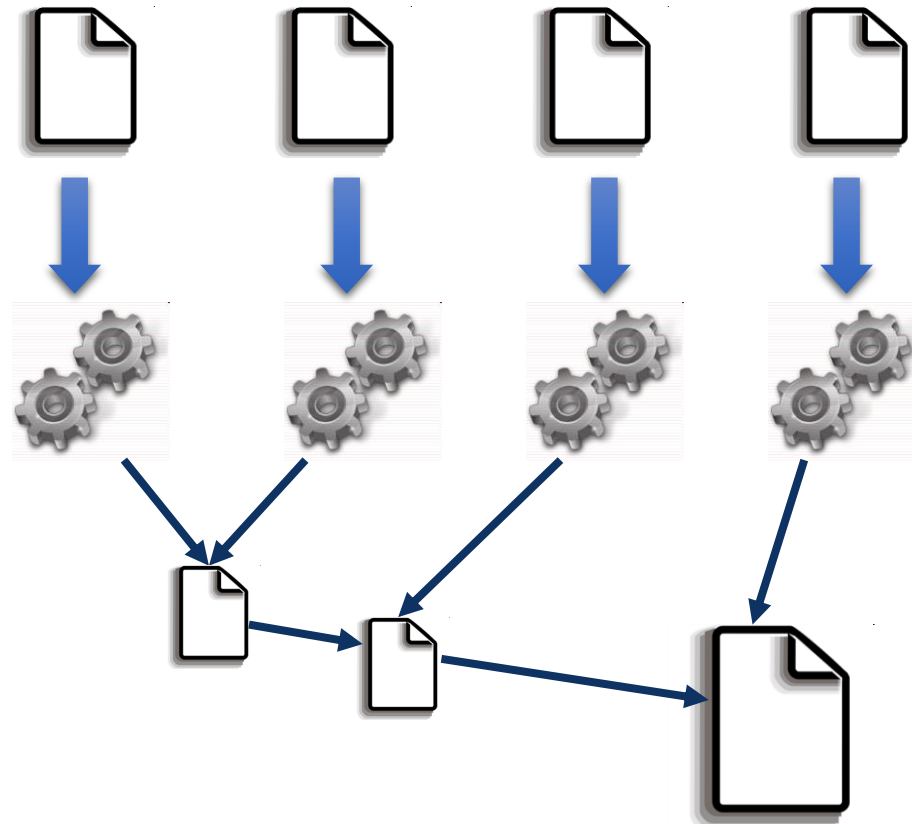


**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Wordcount Exercise

- Counting words of a set of documents
- Parallelization
 - Phase 1: Count words of a set of documents
 - Phase 2 : Reduce results accumulating the partial results



Performance Analysis



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Wordcount @ Performance Analysis

- COMPSs can generate post-execution traces of the distributed execution of the application
 - Useful for performance analysis and diagnosis
- How it works?
 - Task execution and file transfers are application events
 - An XML file is created at workers to keep track of these events
 - At the end of the execution all the XML files are merged to get the final trace file
 - COMPSs uses Extrae tool to dynamically instrument the application
 - In a worker:
 - Extrae keeps track of the events in an intermediate file
 - In the master:
 - Extrae merges the intermediate files to get the final trace file

Wordcount @ Performance Analysis

----- Executing wc_reduce.py -----

Welcome to Extrae 3.5.3

Extrae: Generating intermediate files for Paraver traces.

Extrae: Intermediate files will be stored in /gpfs/home/nct01/nct01XXX/sources/examples

Extrae: Tracing buffer can hold 500000 events

Extrae: Tracing mode is set to: Detail.

Extrae: Successfully initiated with 1 tasks

[API] - Deploying COMPSs Runtime v3.1

[API] - Starting COMPSs Runtime v3.1

...

[API] - No more tasks for app 0

[API] - Getting Result Files 0

[API] - Execution Finished

...

Extrae: Application has ended. Tracing has been terminated.

merger: Output trace format is: Paraver

merger: Extrae 3.5.3

mpi2prv: Selected output trace format is Paraver

mpi2prv: Parsing intermediate files

mpi2prv: Generating tracefile (intermediate buffers of 745642 events)

mpi2prv: Congratulations! ./trace/wc_reduce.py_compss_trace_1453885329.prv has been generated.

Extrae starts before
the user application execution

COMPSs runtime starts

COMPSs runtime ends

The application finishes and
the tracing process ends

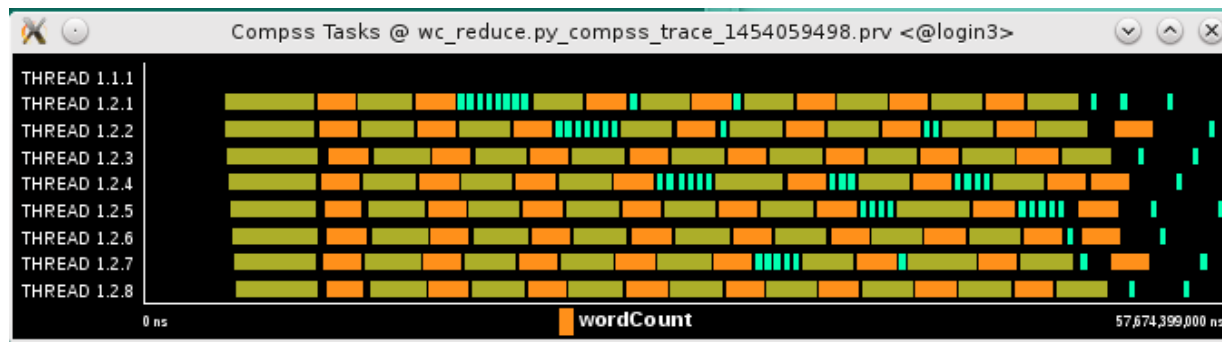
The merge process starts

Intermediate trace files
are processed

The final trace file is
generated

WordCount @ Performance Analysis

- Open Paraver
 - **module load paraver**
 - **cd \$HOME/.COMPSSs/wordcount.py_01**
 - **wxparaver trace/*.prv**
 - COMPSs provides some configuration files to automatically obtain the view of the trace
 - File/Load Configuration...
- `/gpfs/apps/MN4/COMPSSs/3.1/Dependencies/paraver/cfgs/compss_tasks.cfg`

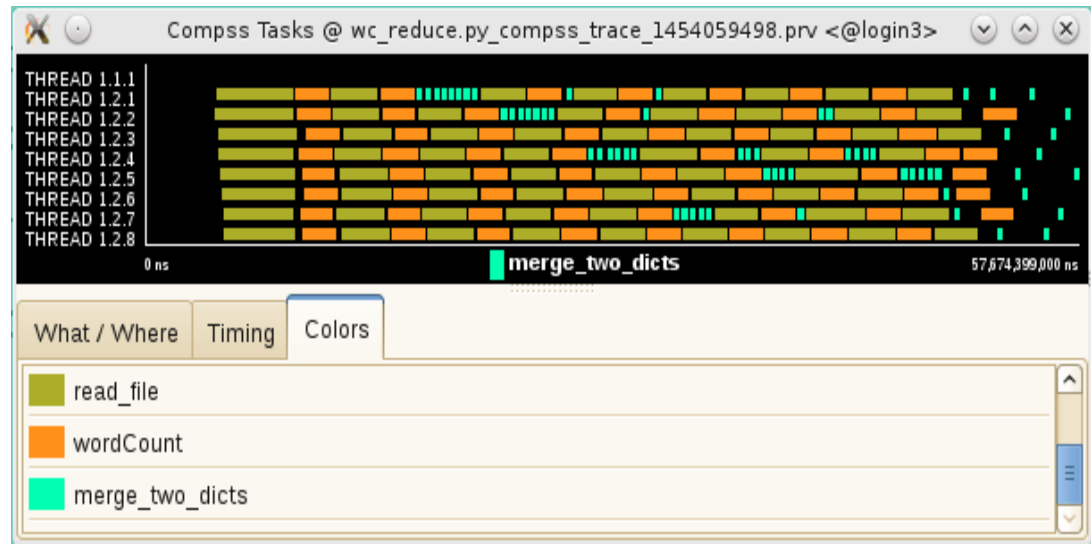


Wordcount @ Performance Analysis

- Summarizing:
 - Lines in the trace:
 - THREAD 1.1.X are the master threads
 - THREAD 1.X.Y are the worker threads

- Meaning of the colours:
 - Black: idle
 - Other colors: task running
 - see the color legend

- Flags (events):
 - Start / end of task



Exercise 2

Parallelize

Wordcount with

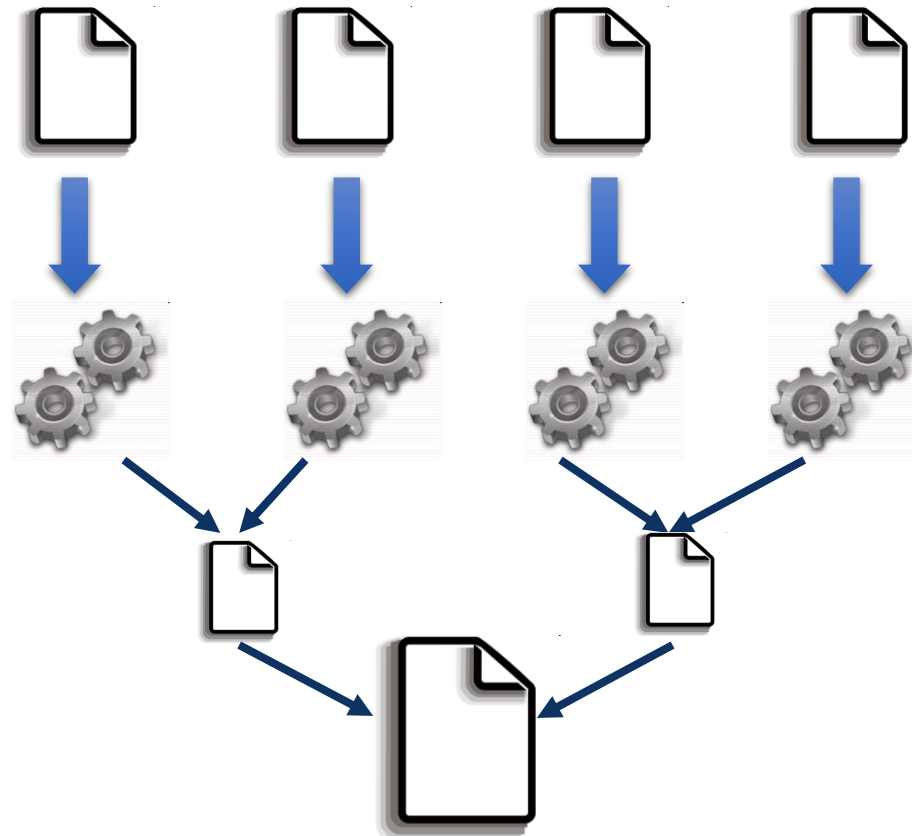
Merge



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación

Wordcount Exercise with merge

- Counting words of a set of documents
- Parallelization
 - Phase 1: Count words of a set of documents
 - Phase 2 : Reduce results merging partial results





**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación



EXCELENCIA
SEVERO
OCHOA

THANK YOU!

support-compss@bsc.es

www.bsc.es