Title: Programming workflows with PyCOMPSs

Instructors: Rosa M Badia (Barcelona Supercomputing Center),
Javier Conejero (Barcelona Supercomputing Center)

Rosa M. Badia has a PhD on Computer Science (1994) from the Technical University of Catalunya (UPC). She is the manager of the Workkflows and Distributed Computing at the Barcelona Supercomputing Center (BSC) a Scientific Researcher from the Consejo Superior de Investigaciones Científicas (CSIC). She was involved in teaching and research activities at the UPC from 1989 to 2008, where she was an Associated Professor. From 1999 to 2005 she was involved in research and development activities at the European Center of Parallelism of Barcelona (CEPBA). Her current research interests are programming models for complex platforms (from multicore, GPUs to Grid/Cloud). She has published more than 150 papers in international conferences and journals in the topics of her research. She has participated in a large number of European projects and currently she is participating in the projects Euroserver, The Human Brain Project, EU-Brazil CloudConnect, the BioExcel CoE, NEXTGenIO, MUG, EUBra BIGSEA, TANGO, mf2C and it is a member of HiPEAC2 NoE. She has delivered multiple tutorials in conferences and events about the programming models at BSC (OmpSs and PyCOMPSs/COMPSs).

Javier Conejero: Senior Researcher at Barcelona Supercomputing Center. He holds a PhD Degree in Computer Science from the University of Castilla-La Mancha (2014). Javier worked at CERN for one year into WLCG software development and management. His current research interests are programming models, efficient exploitation of Cloud and HPC environments and QoS. He has been the main developer of PyCOMPSs since his arrival to BSC on 2014. He is also contributing to the European projects NEXTGenIO and MUG.

Abstract

Task-based programming models have proven to be a suitable approach to exploit large-scale parallelism by enabling a data-flow execution model and avoiding global synchronization.
COMPSs belongs to this category and is able to exploit the inherent concurrency of sequential applications and execute them in a distributed platform, like Clusters. This is achieved by annotating part of the codes as tasks, and building at execution time a task-dependency graph based on the data consumed/produced by the tasks. COMPSs runtime is able to schedule tasks in computing nodes taking into account facts like data locality and node heterogeneity. The recently released Python binding (PyCOMPSs) opens a new door to execute Python scripts in parallel in distributed platforms, especially with its recent integration with Jupyter notebook.

The objectives of the tutorial are to give an overview of the PyCOMPSs/COMPSs task based programming model syntax: how tasks are annotated, how tasks are synchronized; also, to demonstrate how to use PyCOMPSs/COMPSs to parallelize and run applications in clusters and clouds: parallelization strategies and sample codes to

illustrate how the framework words; and to give an overview of the COMPSs runtime: scheduling and policies, interface with execution platforms.

Agenda

The total estimated duration of the tutorial is 3 hours.

The tutorial will be composed of presentations and hands-on. The outline of the tutorial is the following:

- Welcome and introductions (15 min)
- Session 1: Introduction to PyCOMPSs (45 min)
  - Programming model
  - Python syntax
  - Demo: First Python example
- Section 2: Local environment hands-on (45 min)
  - Python Hands-on
  - Simple code without annotations (wordcount)
  - Tasks' annotation
  - Execution in local environment with Jupyter notebook
  - Overview of monitoring, tracing, trace analysis
- Session 3: COMPSs runtime (30 min)
  - Runtime components
  - Scheduling and policies
  - Platform interoperability
- Session 4: Demo: execution in large scale platform (25 min)
  - Execution of examples provided by the instructors in a large cluster (MareNostrum or MinoTauro)
  - Submission of PyCOMPSs applications in a cluster
  - Tracefile generation and analysis
- Conclusions (5 min)

Infrastructure

For the hands-on we will provide the attendees with a VirtualBox VM, with all software and examples installed. We can make this VM available before the tutorial. This can be added as a link in the website.
We will need good wi-fi connection.
Also, if possible (it is not a must), the availability of two screens/projectors helps when we do the hands-on or demos.

Software information: COMPSs webpage, including training material: *compss.bsc.es*

Limitations in the number of attendees: none