

## COMP Superscalar

The COMP Superscalar (COMPSs) framework is mainly composed of a task-based programming model which aims to ease the development of parallel applications for distributed infrastructures, such as Clusters, Clouds and containerized platforms, and a runtime system that exploits the inherent parallelism of applications at execution time. The framework is complemented by a set of tools for facilitating the development, execution monitoring and post-mortem performance analysis.

### Programming Model

With the objective of offering a programming environment with high productivity and portability, the COMPSs Programming model has the following key characteristics:

- **Sequential programming:** COMPSs programmers can develop their applications following the sequential programming paradigm. In this sense, the programmer does not need to take care of the parallelization and distribution aspects, such as thread creation and synchronization, data distribution, messaging or fault tolerance.
- **Infrastructure agnosticism:** COMPSs abstracts applications from the underlying distributed infrastructure. COMPSs programs do not include any detail that could tie them to a particular platform, like deployment or resource management. This makes applications portable between infrastructures with diverse characteristics.
- **Standard programming languages:** COMPSs applications can be developed in Java, Python and C/C++. The use of a general purpose programming language facilitates adoption, since these languages are between the more common and popular between programmers.
- **APIs:** In the case of COMPSs applications in Java, the model does not require to use any special API call, pragma or construct in the application; everything is pure standard Java syntax and libraries. With regard the Python and C/C++ bindings, a small set of API calls should be used on the COMPSs applications.

### Runtime System

The COMPSs programming model is supported by a runtime system which manages several aspects of the applications' execution. Besides, it keeps the underlying infrastructure transparent to the application. Some important functionalities implemented by the COMPSs runtime are:

- **Task Dependency Analysis:** tasks are the basis for the parallelism in COMPSs. The runtime automatically finds the data dependencies between tasks based on the direction of their parameters. With this information, it dynamically builds a task dependency graph.
- **Task Scheduling:** when tasks are free of dependencies, they are scheduled by the runtime in the available distributed resources.
- **Data synchronization:** data accesses from the main program of the application are automatically synchronized by the runtime, when necessary.

- **Resource Management:** for Cloud environments, the runtime features a set of pluggable connectors, each implementing the interaction of the runtime with a particular IaaS Cloud provider. This design enables interoperability and prevents vendor lock-in, since COMPSs can reserve and manage virtual resources coming from different Cloud providers, thus not being tied to a particular API. The number of reserved virtual resources can be elastically adapted to the task load that the runtime is processing.
- **Job & Data Management:** the runtime is also in charge of performing remote execution of tasks and the data transfers. It provides an extensible interface for supporting several protocols for Job and Data Management. In the current release two adaptors are implemented: the Non-blocking I/O, which offers high performance in secured environments; and the [GAT](#) adaptor, which offers interoperability with diverse kinds of Grid middleware.

## Citing COMPSs

Please, use the following references when citing COMPSs in your publications:

### Primary citation:

- ServiceSs: an interoperable programming framework for the Cloud, *Journal of Grid Computing*, March 2014, Volume 12, Issue 1, pp 67–91, Lordan, F., E. Tejedor, J. Ejarque, R. Rafanell, J. Álvarez, F. Marozzo, D. Lezzi, R. Sirvent, D. Talia, and R. M. Badia, DOI: 10.1007/s10723-013-9272-5

### Code reference:

- COMP Superscalar, an interoperable programming framework, *SoftwareX*, Volumes 3–4, December 2015, Pages 32–36, Badia, R. M., J. Conejero, C. Diaz, J. Ejarque, D. Lezzi, F. Lordan, C. Ramon-Cortes, and R. Sirvent, DOI: 10.1016/j.softx.2015.10.004

### PyCOMPSs reference:

- PyCOMPSs: Parallel computational workflows in Python, Enric Tejedor, Yolanda Becerra, Guillem Alomar, Anna Queralt, Rosa M. Badia, Jordi Torres, Toni Cortes, Jesús Labarta, *IJHPCA* 31(1): 66-82 (2017), DOI: 10.1177/1094342015594678

Barcelona Supercomputing Center - Centro Nacional de Supercomputación

---

**Source URL (retrieved on 5 Jul 2024 - 22:14):** <https://www.bsc.es/es/research-and-development/software-and-apps/software-list/comp-superscalar>