Storage platform for data sharing

The Big Data challenge responds to the growing need for combining disparate data sources to gain new insights from data.

The goal of this research line is to realize a convenient way of sharing data, both for consumers and for providers, in order to motivate data sharing and foster the creation of new knowledge and services that would otherwise be impossible to provide.

Summary

Sharing data so that researchers, students, application developers, and citizens in general can build new applications or services based on it is nowadays a trend. Open Data initiatives have this same goal, but the fact that they deal with public data makes these mechanisms not applicable when more sensitive data is to be shared. As a result, organizations end up by sharing only a very limited part of their data, thus reducing the possible impact on society.

Since data is one of the most valuable assets for companies, sharing it in such a way that governance is kept and at the same time the possibilities of exploitation by third parties are maximized, poses new technological challenges to be investigated.
In this research line we address these issues by raising the abstraction at which data is manipulated. In particular, we propose the concept of self-contained objects, a new abstraction where data, methods, and policies are encapsulated to enable data sharing between providers and third-party consumers. In this way, issues such as privacy and security are guaranteed by data objects themselves, facilitating both data sharing among independent organizations, and offloading data and computation to the cloud.

- **Self-contained objects.** A self-contained object is a piece of data that also contains all the logic needed to process it (methods), and the policies that manage its behaviour with respect to security, integrity, etc.
- **3rd-party enrichment.** The future of sharing data consists of building new services by iteratively enriching the available data, that is, adding additional information and/or ways of processing it. We will research mechanisms to enable 3rd parties to create/modify how data is seen by applications without moving any data around and without affecting performance. These new views will share the infrastructure with the original data without compromising the security or integrity of the original data, but will allow to increase its value.
- **Data and computation offload.** Offloading computation to third-party infrastructures is clearly a need in cases where computation is very costly. In this research line we will investigate how self-containment properties can enable objects to move from one infrastructure to another without losing any of their properties. The objective is to perform this offloading and still guarantee data security and integrity because not just the data is moved, but also its methods and expected behaviour.

**Objectives**

The objective of this research line is to deliver a platform based on the previous concepts that will improve both the experience of data sharing and the exploitation of shared data. This will be achieved by enabling the enrichment of both data and code in the same infrastructure, as well as to process such data using any available resources, and not only the ones owned by the data provider, without his losing control on the data.

An additional goal of this research is on the usability side, and intends to reduce the effort required for data consumers to build new applications. This is achieved, again, by the fact that the platform is based in the concept of objects, dealing with data in the same way as if it was all in memory. In this way, application developers do not need to learn yet another interface to access persistent data, and do not have to deal with data format issues. In addition, data stored in the platform will be accessible both from Java and from Python applications, independently of the programming language used to create the data and the classes that allow to manipulate it.

Finally, it is also an objective of this research line to improve access performance and usability of large data sets. In this respect, the platform should be able to work both independently and in combination with the (Py)COMPSs programming model, in order to exploit parallelism when possible.