



Contribution ID: 338

Type: **Poster**

## Executor framework for DIRAC

*Tuesday, 22 May 2012 13:30 (4h 45m)*

DIRAC framework for distributed computing has been designed as a group of collaborating components, agents and servers, with persistent database back-end. Components communicate with each other using DISET, an in-house protocol that provides Remote Procedure Call (RPC) and file transfer capabilities. This approach has provided DIRAC with a modular and stable design by enforcing stable interfaces across releases. But it made complicated to scale further with commodity hardware.

To further scale DIRAC, components needed to send more queries between them. Using RPC to do so requires a lot of processing power just to handle the secure handshake required to establish the connection. DISET now provides a way to keep stable connections and send and receive queries between components. Only one handshake is required to send and receive any number of queries. Using this new communication mechanism DIRAC now provides a new type of component called executor. Executors process any task (such as resolving the input data of a job) sent to them by a task dispatcher. This task dispatcher takes care of persisting the state of the tasks to the storage backend and distributing them amongst all the executors based on the requirements of each task.

In case of a high load, several executors can be started to process the extra load and stop them once the tasks have been processed. This new approach of handling tasks in DIRAC makes executors easy to replace and replicate, thus enabling DIRAC to further scale beyond the current approach based on polling agents.

**Student? Enter 'yes'. See <http://goo.gl/MVv53>**

no

**Primary author:** CASAJUS RAMO, Adrian (University of Barcelona (ES))

**Co-author:** GRACIANI DIAZ, Ricardo (University of Barcelona (ES))

**Presenter:** CASAJUS RAMO, Adrian (University of Barcelona (ES))

**Session Classification:** Poster Session

**Track Classification:** Distributed Processing and Analysis on Grids and Clouds (track 3)