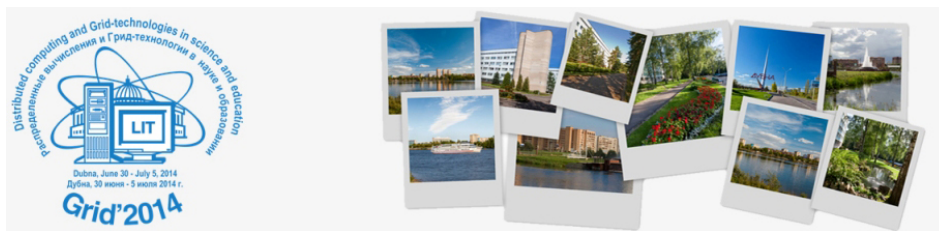


The 6th International Conference "Distributed Computing and Grid-technologies in Science and Education"



Contribution ID: 52

Type: **sectional reports**

Development of Distributed Computing Applications and Services with Everest Cloud Platform

Monday, June 30, 2014 3:25 PM (20 minutes)

The ability to effortlessly use and combine existing computational tools and computing resources is an important factor influencing research productivity in many scientific domains. However, scientific software often requires specific expertise in order to install, configure and run it that is beyond the expertise of an ordinary researcher. This also applies to configuration and use of computing resources to run the software. Finally, researchers increasingly need to combine multiple tools in order to solve a complex problem, which brings an important issue of application composition.

The report presents Everest, a cloud platform that addresses these problems by supporting publication, sharing and reuse of scientific applications as web services. The underlying approach is based on a uniform representation of computational web services and its implementation using REST architectural style pioneered by MathCloud project [1].

In contrast to traditional service development tools, Everest follows the Platform as a Service cloud delivery model by providing all its functionality via remote web and programming interfaces. A single instance of the platform can be accessed by many users in order to create, run and share services with each other without the need to install additional software on users' computers.

Another distinct feature of Everest is the ability to run jobs on external computing resources and connect services to arbitrary sets of resources. A computing resource can be attached to the platform by any user. A resource owner can configure a policy for accessing the resource. Any allowed user can bind the resource to any service. It is also possible to bind multiple resources to a service, or override default resources by providing another resource for running a job.

While the platform doesn't provide its own infrastructure to run compute jobs as classic PaaS examples, it can handle the problems of resource allocation, job scheduling, data transfer and so on without the interference of users. All interaction with external computing resources is performed by the Compute subsystem that represents a generic metascheduling framework. It manages execution of computational jobs, consisting of one or more tasks, on remote resources and performs all routine actions related to staging of task input files, submitting a task, monitoring a task state and downloading task results. Compute subsystem also monitors the state of resources attached to the platform and uses this information during job scheduling.

The platform's application programming interface (REST API) is implemented as a RESTful web service. It serves as a single entry point for all clients, including the web user interface. The latter is implemented as a JavaScript application and provides a convenient graphical interface for interaction with the platform. A Python API is implemented on top of REST API in order to support writing programs that access services and combine them in arbitrary workflows.

The work is supported by the Russian Foundation for Basic Research (grant No. 14-07-00309 A).

[1] Afanasiev A., Sukhoroslov O., Voloshinov V. MathCloud: Publication and Reuse of Scientific Applications as RESTful Web Services // Victor E. Malyshev (Ed.): Parallel Computing Technologies (12th International Conference, PaCT 2013, St. Petersburg, Russia, September 30 —October 4, 2013). Lecture Notes in Computer Science Volume 7979. Springer 2013. pp. 394-408

Primary author: Dr SUKHOROSLOV, Oleg (IITP RAS)

Co-authors: Mr RUBTSOV, Anton (IITP RAS); Mr VOLKOV, Sergey (IITP RAS)

Presenter: Dr SUKHOROSLOV, Oleg (IITP RAS)

Session Classification: Section 1 - Technologies, architectures, models, methods and experiences of building distributed computing systems. Consolidation and integration of distributed resources

Track Classification: Section 1 - Technologies, architectures, models, methods and experiences of building distributed computing systems. Consolidation and integration of distributed resources