Toolkit for Management of Large-scale Monte Carlo Applications

D. Vudragović, A. Balaž, and V. Slavnić

Scientific Computing Laboratory, Institute of Physics Belgrade, University of Belgrade
Pregrevica 118, 11080 Belgrade, Serbia, http://www.scl.rs/
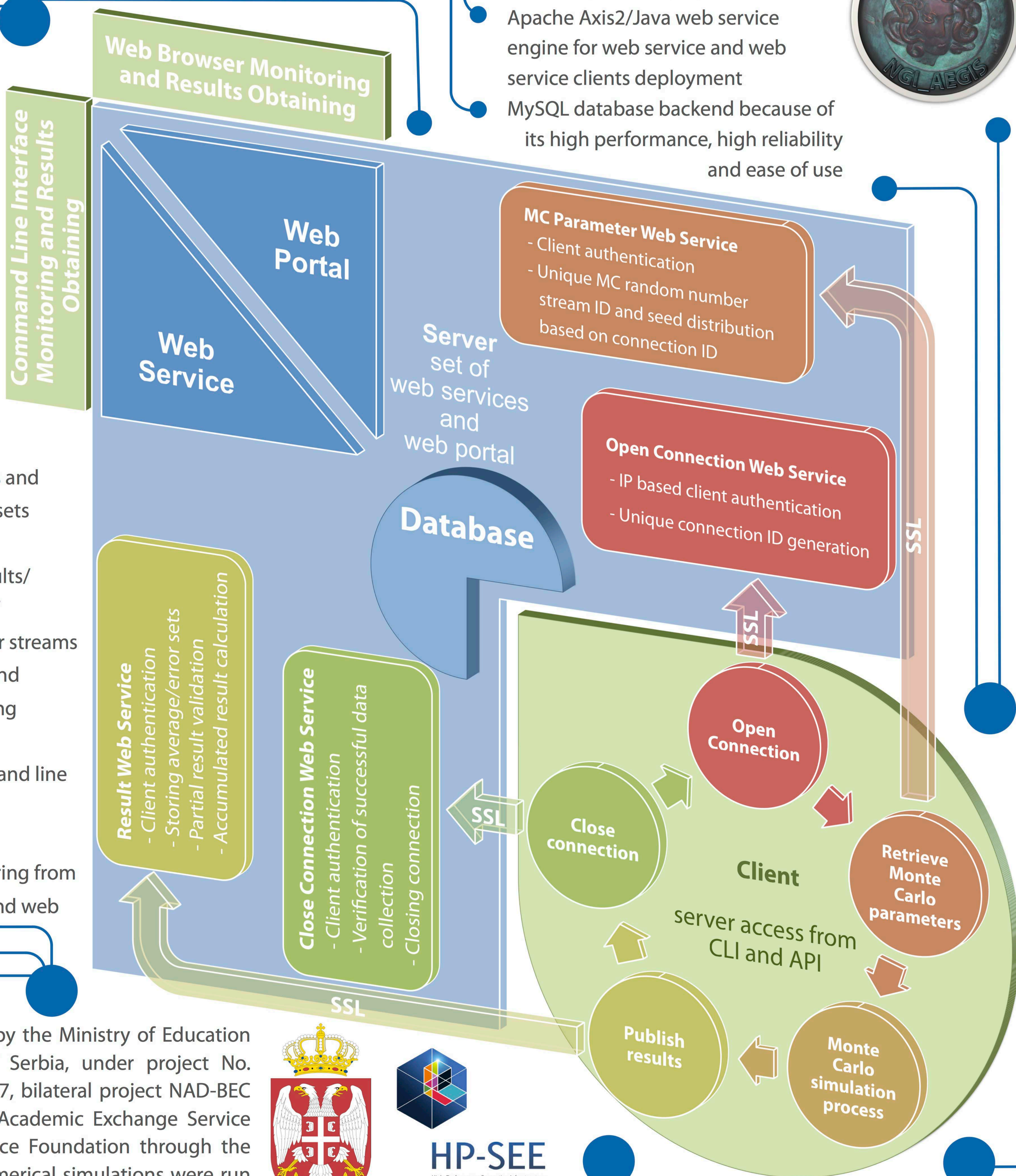
EGI-InSPIRE

Overview

Monte Carlo (MC) applications are inherently parallel and computationally-bound. In a distributed computing infrastructure, once MC application is initiated, it can be usually executed independently, i.e. with almost no interprocess communication. Such behavior allows for a usage of any kind of computing resources, but requires an integrated framework to simplify application management, and even most importantly, proper usage of the underlining parallel random number generator and reproducibility of numerical results. Here we present newly developed toolkit for management of large-scale Monte Carlo simulations on various kinds of computing resources, which includes management of parallel random number streams utilized by individual processes, as well as data gathering and post-processing functionality.

Implementation

- Client-server architecture; Server is deployed centrally; Client IP address is not required to be public
- Java programming language as general-purpose, concurrent, strongly typed, class-based object-oriented language
- Apache Tomcat open source servlet container for the set of web services and web portal deployment
- Apache Axis2/Java web service engine for web service and web service clients deployment
- MySQL database backend because of its high performance, high reliability and ease of use

Features

- Client is portable on almost any High Performance Computing (HPC) system
- Internet Protocol (IP) based client authentication
- Security communication over the Internet via Secure Sockets Layer (SSL) cryptographic protocol
- Distribution of the unique random number streams and seeds to the clients
- On-the-fly calculation of results and errors from accumulated data sets
- Validation of partial results, identification of suspicious results/ problems, and redistribution of corresponding random number streams
- Auto detection of a failed job and redistribution of a corresponding random number stream
- Server access both from command line interface (CLI) and application programming interface (API)
- Retrieval of results and monitoring from command line interface (CLI) and web portal

Acknowledgments