Biocep, an e-Science Computational Platform for the Cloud

Karim Chine
Cloud Era Ltd
Cambridge, UK
e-mail: karim.chine@m4x.org

Abstract
Biocep builds on top of the highly popular statistical environment R, an e-platform for computing and data analysis.

Keywords**:** e-computing; e-statistics; HPC; cloud computing; distributed computing; application virtualization; Web Services; workflows; cloudbursting; large scale data mining; collaborative data analysis; open source

R is becoming the *lingua franca* of data analysis and statistical computing. It has a very powerful graphics system as well as cross-platform capabilities for packaging any computational code. Hundreds of available R packages, exponentially growing in number, implement the most up-to-date computational methods and reflect the state-of-the-art of research in various fields. R packages are likely to become a reproducible research enabler because they enable functions and algorithms to be reused and shared. There is no obstacle to a large-scale deployment of R on public grids since it is a GPL software. However, R is not multithreaded. It does not operate as a server and it has only a low-level non-object-oriented API. GUI development for R remains non-standardized. R's potential as a computational back end engine for applications and service-oriented architectures has yet to be fully exploited. While its user base is growing at a high rate, this growth rate would be significantly higher in the presence of a user-friendly and rich workbench.

Biocep is a general unified open source Java solution for integrating and virtualizing the access to R engines/servers. It aims to become a federative user-friendly computational e-platform for research, finance and education. The Biocep virtual workbench provides a framework enabling the connection of all the elements of a computational environment:
1. The computational resource (whether it is a local machine, a cluster, a grid or a cloud server) via a simple URL.
2. The computational components via the import of R packages.
3. The GUIs via the import of plugins from repositories or the design of new views with a drag-and-drop GUI editor.
Several dockable built-in views allow users to work interactively with R engines running at any location. The views include a console, highly interactive remote graphic devices (with built-in zooming, scrolling, real coordinate tracking..), PDF and SVG viewers, R data inspectors, linked plots and spreadsheets that are fully integrated with R functions and data.

Biocep enables collaborative R sessions − multiple web users can connect simultaneously to an R server running anywhere and analyze data collaboratively via a set of broadcasted views. For example, the console log is sent in real time to all users. Chatting is enabled and a graphic device is synchronously updated for all. Biocep includes an editable collaborative spreadsheet that retains data on the server, removing limits on client machines. Distributed and linked statistical graphics based on a refactored *iplots* package (www.iplots.org) enable the collaborative highlighting and color brushing of various linked plots.

Biocep frameworks and tools make it possible to use R as a Java object-oriented toolkit or as an RMI server. All the standard R objects have been mapped to Java and user defined R classes can be mapped to Java on demand. Calls to R functions from java locally or remotely cope with local and distributed R objects. An easy-to-use Web Services generator is provided to enable automatic exposure of R functions and packages as Web Services. They can be seamlessly integrated as nodes into workflows. They can be stateless (an anonymous R worker performs the computation) or stateful (an R worker reserved and associated with a session ID is used and can be reused until the session is destroyed). The statefulness solves the overhead problem caused by the transfer of intermediate results between workflow nodes. A stateful R-SOAP API exposes the full capabilities of the platform and enables an efficient integration of R into data analysis pipelines using PERL, C, C++, C#, Java or R. R-SOAP clients are provided for each language.

Biocep provides a remote resources pooling framework (RPF) allowing pools of R engines to be deployed on heterogeneous nodes. These engines are managed and used via a simple borrow/return API for multithreaded web applications and web services, for distributed and parallel computing, for dynamic content on-the-fly generation (analytic results, tables and graphics in various formats for thin web clients) and for R virtualization in a shared computational resources context. RPF enables transparent cloudbursting: Amazon EC2 virtual machines running R servers can be fired up or shut down to scale up or scale down according to the load in a highly scalable web applications deployment.

Biocep has built-in Python and Groovy scripting facilities both on server and on client sides. The bridging of R and the scripting interpreters is bi-directional. R objects can be exported to Python/Groovy and vice versa and the scripts can embed seamlessly any R code. Scripting with R as a component becomes easier than ever using either the Biocep APIs or the workbench’s views. User Java code can be dynamically loaded by R servers and used for scripting.

In Summary, Biocep combines the capabilities of R and the flexibility of a Java based distributed system to create a tool of considerable power and utility. A Biocep based R virtualization infrastructure has been successfully deployed on the British National Grid Service, demonstrating its usability and usefulness for researchers. Biocep could

become an essential building block of a new generation of distributed or web-based statistical software. The virtual workbench enhances the user experience and the productivity of anyone working with R. As Biocep is extensible, it enables the emergence of repositories of plugins. The interoperability, coupled with a large-scale deployment of virtualization infrastructures on various grids democratizes R based HPC and enables users from within their browsers to compute and visualize data with unprecedented flexibility and performance. The adoption of the new platform would be a step forward in the direction of interoperability, reusability and seamless integration of research resources (and therefore a reproducible research enabler). Finally, Biocep may work as an enabler of a new computing business model that would synergize the utility computing model (resources) and the pay-per-use software model (components/GUIs).

Project Home Page: www.biocep.net

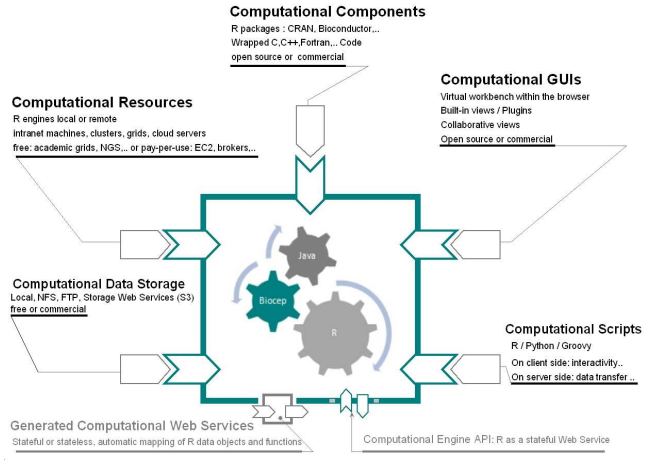

Figure 1 - Biocep computational open platform

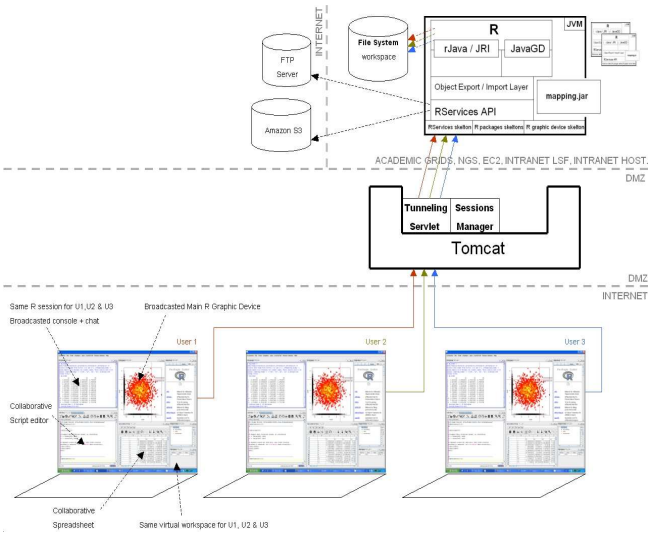

Figure 2 – Virtual and collaborative R

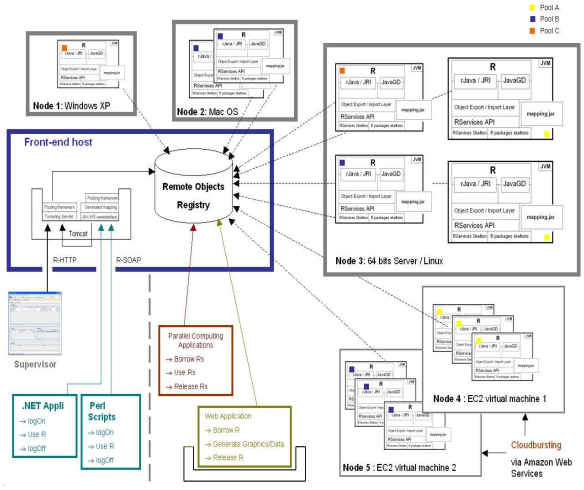

Figure 3 - R engines pools deployment - Cloudbursting

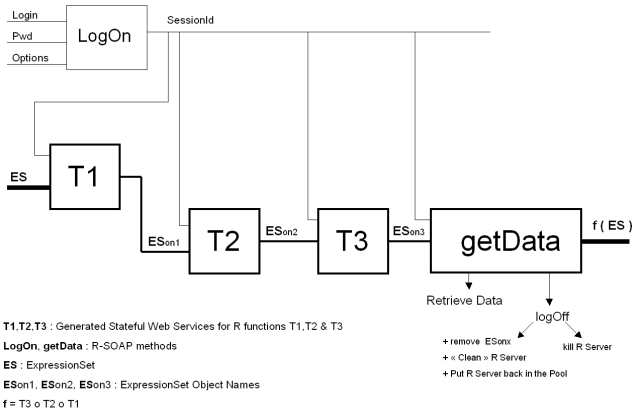

Figure 4 - generated stateful Web Services workflows

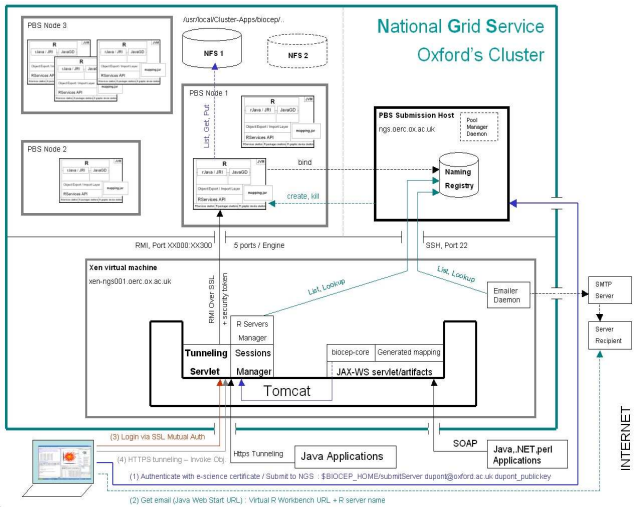

Figure 5 – R virtualization on the National Grid Service