A Grid Computing Environment for Design and Analysis of Computer Experiments

Yann Richet[1,2,*] , David Ginsbourger[1,3], Olivier Roustant[1,4], Yves Deville[1,5]

1. DICE consortium, "Deep Inside Computer Experiments", France
2. IRSN, Radioprotection and Nuclear Safety Insitute, Fontenay-aux-Roses, France
3. CHYN, Institute of Geology and Hydrogeology, University of Neuchatel, Switzerland
4. ENSM-SE, Graduate School of Engineering, Saint-Etienne, France
5. Statistical consultant, Chambery , France

* Contact author: yann.richet@irsn.fr

Keywords: grid computing, GUI, Rserve, kriging, EGO

Promethee[1] is a flexible and generic grid computing environment designed for numerical engineering. As a key feature, this tool provides a Graphical User Interface (GUI) for the management of parametric computing tasks with heavy software. From the user point of view, Promethee frontend GUI allows to:

1. edit and parametrize input files (of a given computing software),
2. launch remotely and simultaneously all calculations (thanks to Promethee backend grid),
3. parse and store all software output results.

This basic usage of Promethee holds interesting day-to-day benefits for engineering, and above all, turns the computing software usable as a math function, thus suitable for design of computer experiments frameworks. Latest developments of Promethee include the direct integration of algorithms included in R packages, such as **sensitivity**, **DiceDesign**[2], **DiceKriging**[2] and **DiceOptim**[2]. The key point of this R integration is to easily bring cutting-edge algorithms inside Promethee environment, so that any engineer using Promethee as its grid computing frontend may access powerful algorithms without any knowledge in R programming. This fast and easy wrapping of R packages inside one's grid significantly shortens the time-to-market delay for algorithms development projects, and thereby increases industrial interest and support for these R&D services. Obviously a proper use of most of these algorithms relies on users understanding of underlying mathematical background, and could not be dissociated from well suited documentation (mainly provided by packages documentation and vignettes) and training courses (to be embedded in service deliverables).

Technically speaking, Promethee embeds **Rserve**[3] *java* client as a library dependency, and the access to R codes and results is performed through the **Rserve** protocol, thus allowing remote execution (and a possible backend centralization) of R functions.

As an example, the integration of EGO (Efficient Global Optimization[4,5]) algorithm through **DiceKriging** and **DiceOptim** packages inside Promethee grid environment at IRSN is illustrated by a real-world application in nuclear safety assessment based on an industrial Monte-Carlo neutronic simulator[6].

References

[1] Promethee project (2009). "*Unobtrusive grid computing, efficient parametric modeling and deep robust design"*,
http://promethee.irsn.org/.

[2] DICE consortium (2006-2009). "*Deep Inside Computer Experiments"*, http://www.dice-consortium.fr/

[3] Rserve. "*Binary R Server"*, http://www.rforge.net/Rserve/

[4] D. Ginsbourger, R. Le Riche, and L. Carraro (2009), chapter "*Kriging is well-suited to parallelize optimization*", to appear in Computational Intelligence in Expensive Optimization Problems, Studies in Evolutionary Learning and Optimization, Springer.

[5] D.R. Jones (2001), "*A taxonomy of global optimization methods based on response surfaces"*, Journal of Global Optimization, 21, 345-383.

[6] J. Miss, O. Jacquet, F. Bernard, B. Forestier, W. Haeck, Y. Richet, (2008), *"First validation of the new continous energy version of MORET5 Monte Carlo code"*, PHYSOR 2008, Interlaken, Switzerland