Coalition : a simple and useful tool to distribute R-works on a set of computers

Marie-Pierre Etienne[1,2,*], **Cyril Corvazier**,[3] , **Benjamin Legros**[3]

1. AgroParisTech, UMR 518, Mathématiques et Informatique Appliquées, F-75005 Paris, France
2. INRA, UMR 518, Mathématiques et Informatique Appliquées, F-75005 Paris, France
3. Mercenaries Engineering, Paris, France

* Contact author: marie.etienne@agroparistech.fr

Keywords: Distributed Computing, Resource Manager,

Intensive simulations, Monte Carlo methods or hierarchical Bayesian estimation can be highly CPU intensive and, in order to be solved in a decent amount of time, have to be run on a set of computers. More and more computations are distributed on dedicated servers, the difficulty is then to manage the computation tasks. Most of the available tools for distributed computing with R are R specific (see for instance SNOW package, Rossini el al, 2009) and/or require strong network programming skills.

We propose a simple solution called Coalition to distribute shell commands on a set of computers. From the user point of view, it is as simple as using R (or whatever he wants) in command line and Coalition manages the computational resources very similarly to SLURM (S. M. Balle and D. Palermo, 2007).

Coalition consists of two main Python scripts : `server.py` and `worker.py`. `server.py` is run on the master computer, each computer of the set (worker) runs the `worker.py` script.
A user adds a command on the server either using a web interface or a command line through the `control.py` script. The so called 'job' is added to the server job queue. When a worker is free, it asks the server for a job. As an answer, the server attributes the next job present in the job queue to the worker. If the command is successfully achieved (exit code 0), the task is marked as finished in the job queue. To use R in this framework, one just has to submit some `Rscript` commands to the server.

A job consists of a command line but also of a priority level, some affinity flags and dependencies on other jobs. The job queue is sorted according to the priority levels of the submitted jobs. The affinity flags are used to constraint a job to be run on a subset of workers. For instance, a R job may require a particular package not available on all the workers. In this case the affinity flag may indicate this requirement and only workers with the suitable installation of R can execute the job command. The dependencies of a job, say A, consists of a set of jobs that must be finished before running A. A friendly web interface is available to submit jobs, manage the job list, watch and control the state of the workers.

Because a job command is executed on a worker, all the data required for this command must be available from any worker. This availability is straightforward with a network file-system and may be achieved through for instance FTP commands in other cases. If properly configured, the server can use a LDAP directory to authenticate the users. In this case, the workers are able to run the jobs' commands with the submitter's rights. This is a simple and safe way for the workers to access the users' files over the network.

To take advantage of multi-core CPU, one simply has to run on a single computer as many workers as cores.

A major limitation of this method leads in the necessity to split up the computing task in several independent jobs. This may be an issue for some application. But, with a suitable task splitting, this framework can achieve nearly optimal parallel performances. It has been widely tested in simulation studies context with R.

Coalition is distributed under GNU GPL license and available at `http://coalition.googlecode.com`.

References

Luke Tierney, A. J. Rossini, Na Li, H. Sevcikova (2009) SNOW Package : Simple Network of Workstations `http://cran.r-project.org/web/packages/snow/index.html`

S. M. Balle and D. Palermo (2007) Enhancing an Open Source Resource Manager with Multi-Core/Multi-threaded Support, *Job Scheduling Strategies for Parallel Processing*.