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

Marie-Pierre Etienne\textsuperscript{1}, Cyril Corvazier\textsuperscript{2} and Benjamin Legros\textsuperscript{2}

1. AgroParisTech - INRA
2. Mercenaries Engineering

User! 2009 Conference
marie.etienne@agroparistech.fr
What is Coalition?

Overall principle

Coalition is a task scheduler

Coalition

A small but beautiful job manager by Mercenaries Engineering - GPL licence
Coalition principle

- One server schedules tasks running with server.py script
- Workers execute tasks running with worker.py script

Coalition is available at http://code.google.com/coalition/ under GNU General Public License v3
How to start with Coalition

1. Running the Server
   python server.py

2. Running the Worker
   python worker.py
How to use Coalition?

1. Using Web interface

2. Using a Python script control.py

```python
python control.py -c "ls" -t "UseRDemo2"
http://localhost:19211 add
```
How to use Coalition with R?

Use of Rscript to run R in command line.

Factorial.R is located /home/metienne/DemoCoalition and contains:

```r
## the wrong way to compute
## factorial of a given argument
args <- commandArgs(TRUE)
m.max <- type.convert(args[1])
file.out <- paste("factorial",
                   m.max,".txt", sep="")
factorial <- numeric(m.max)
for( i in 1:m.max)
{
  prov <- 1
  for(j in 1:i)
  {
    prov <- prov*j
  }
  factorial[i] <- prov
}
# write results in file
write.table(factorial,file.out)
```

1. Using Web interface

   Using Coalition Server Interface:
   
   - Run Coalition Server: `start Coalition Server`
   - Add a job: `Add a Job`
   - Set the job details:
     - Job title: `Factorial`
     - Command: `Rscript Factorial.R 1000`
     - Directory: `/home/metienne/DemoCoalition`
   - Save and submit the job.

2. Scripting with control.py

   - Run Coalition Server: `start Coalition Server`
   - Add a job: `Add a Job`
   - Set the job details:
     - Job title: `Factorial`
     - Command: `python control.py -c "Rscript Factorial.R 1000" -d "/home/metienne/DemoCoalition"`
   - Save and submit the job.

http://localhost:19211 add
Other basic functions

Every function may be controlled using either web interface or control.py script.

- Show logs
- Remove a job/a selection of jobs
- Reset a job/a selection of jobs
- Control workers
Priority, Dependency and Affinity system

- **Priority.** To give some priority to pressing jobs, change the priority level: Jobs are submitted according to their priority level.

- **Dependency.** If your job, say number 10, needs the results of another one say 8, use dependency option. Since submitting the job, precise the job ID of the required one.

- **Affinity.** You can tag workers with affinities. A job requiring specific affinities will be run only on a worker meeting all the specified affinities. It is a way to manage R-packages availability on a pool of computers.
To fully exploit the multicore capacities of the processor, one simply runs a worker per core available on the computer. Be sure to have enough RAM for all the processes.
Coalition deals also with

- LDAP authentication
- Windows and GNU/Linux support
- iPhone support ...

And aims to deal with

- Kinship notion and progress status
- Sampling next task according to priority level
To conclude

- Coalition is a solution to share computational resources.
- If you can divide your work in independent tasks, Coalition allows an optimal usage of your computational resources.
- Using OS scheduler, worker may be launched during inactivity time.
- And last but not least, Coalition is very simple to deploy.

Thanks to Hamid Aichoune, our IT, for his advices and his beta testing work.