batchtools: Tools for R to work on batch systems

9 November 2016

Summary

The `R` [@R] package `batchtools` is the successor of the `BatchJobs` package [@batchjobs_2015]. It provides an implementation of a Map-like operation to define and asynchronously execute jobs on a variety of parallel backends:

- Local (blocking) execution in the current session or in a spawned separate `R` process (intended for debugging and prototyping)
- Local (non-blocking) parallel execution using `parallel`'s multicore backend or `snow`'s socket mode
- Execution on loosely connected machines using SSH
- Docker Swarm
- IBM Spectrum LSF
- OpenLava
- Univa Grid Engine (formerly Oracle Grind Engine and Sun Grid Engine)
- Slurm Workload Manager
- Torque/PBS Resource Manager

Extensibility and customizability is very important for scheduled clusters because configuration of these systems is often heavily tailored towards very specific requirements or special hardware. Thus the interaction with the different schedulers is templated for improved flexibility. Furthermore, custom functions can be hooked into the package to be called at certain events. As a last resort, many utility functions simplify the implementation of a custom cluster backend from scratch. The communication between the master `R` session and the computational nodes is kept as simple as possible and runs completely over the file system. This greatly simplifies the extension to more parallel platforms like cloud computing services.

The `batchtools` package also comes with an abstraction mechanism to assist in conducting large-scale computer experiments. The mechanism is similar to the implementation in BatchExperiments which `batchtools` also supersedes: After defining problems and algorithms, both can be parametrized with arbitrary parameters to define jobs which are then in a second step submitted to one of the parallel backends. The `data.table` package acts as an in-memory database

to keep track of the computational status of all jobs, unique job seeds ensure reproducibility across systems, log files can conveniently be searched using regular expressions and jobs can be annotated with arbitrary tags. Jobs can be chunked to be executed sequentially or in parallel using multiple cores of the computational node in order to reduce the overhead induced by job management and starting/stopping `R`. All in all, the provided tools allow to work with many thousands of jobs in an organized and efficient way.

References