

R with future - Parallel backend via batchtools

Fabian Freund, Brigitte Wellenkamp
(KIM Hohenheim, bwHPC)

July 26th, 2022
ZOOM



Excursus: R package batchtools

- ▶ This package distributes tasks across a HPC resource (usually) by sending the tasks as jobs to the job scheduler
- ▶ On bwUniCluster: Job scheduler=SLURM
- ▶ batchtools comes with a lot of functionality: e.g. rerunning failed jobs, logging tasks run, ...
- ▶ Can also setup makeshift clusters (see last unit today)
- ▶ Essentially, you prepare a general batch script to be executed by the scheduler for running R tasks, where all interesting settings (the SBATCH settings you may want to change from job to job) may be manipulated from the R session
batchtools is called from



future.batchtools: Use batchtools to set up a parallel backend

```
library(future.batchtools)
```

Loading required package: future

- ▶ Simply uses the mechanism of task distribution from batchtools to set up a parallel backend
- ▶ This is essentially it! Does not bring along all the functionality of batchtools



You need a SLURM template

See the text file `slurm_bt.tmpl` for a SLURM template to be used with `future.batchtools`.

Copy this text file to your current working directory



Again, the test function

```
test_node <- function(i,exportsth=NULL){  
  str(exportsth) #Do sth. cheap w. object  
  p1 <- date() #get date  
  sleep_t <- sample(10,1) #random sleep time  
  Sys.sleep(sleep_t) #sleep in R  
  x <- Sys.info() #System info, including host name  
  return(c(run=i,time_start=p1,time_end=date(),  
          pid=Sys.getpid(),host=x["nodename"],  
          sleep_time=sleep_t))}
```



An example plan via `future.batchtools`

```
library(future.batchtools)
library(future.apply)

#Start batchtools SLURM
plan(batchtools_slurm, template = "slurm_bt.tmpl",
      resources=list(nnodes=1,ncpus=1,
                    queue="dev_single",walltime=5))

future_sapply(1:8,test_node,future.scheduling=FALSE,
              future.seed=TRUE)
```

- ▶ What does this do? How many SLURM jobs are queued?
(Hint: look at `.future` in your current directory)
- ▶ How would you tweak the code (including SLURM template) to use another queue (e.g. `single`) and to use only 2 workers
- ▶ Which problem do you see on `bwUniCluster`?



How to use `future.batchtools` on `bwUniCluster`

- ▶ Issue walltime
- ▶ However, jobs already queued are not killed when the main R process is terminated
- ▶ Essentially, if your code sent to a single job (=one future) includes writing the output somewhere, as soon as all SLURM tasks are launched, you can simply stop your R main process
- ▶ Feel free to discuss with us how to set this up in detail



Outlook: Two-level plans

```
library(parallelly);library(future.apply)
#library(future.batchtools)
plan(list(tweak(batchtools_slurm,
               template = "slurm_bt.tmpl",
               resources=list(nnodes=1,ncpus=2,
                             queue="dev_single",walltime=5)),
         multisession))
see_resources <- function(){c(availableCores(),
                              Sys.info()["nodename"],
                              future_replicate(4,Sys.getpid()))}
future_replicate(2,see_resources())
```

- ▶ In some cases, you may want to submit multiple multicore jobs
- ▶ This can be done with a **nested plan**, use `tweak` to explicitly set resources
- ▶ Here: Submit 2 jobs in the queue both requesting 2 cores

