batchtools Socket cluster

0. Modify batchtools

batchtools `clusterFunctionsSocket.R` makes a socket cluster with the local host only. To make a socket cluster with remote hosts, the initialization method for class Socket has to be modified. Github repository `sampoll/batchtools` commit 8fb8854 has the necessary modifications.

I. Make AMI

Begin with the batchtools SSH cluster AMI: ami-1480e86e

- Reinstall (force = TRUE) sampoll/batchtools from github
- Install snow

Result: ami-e85d3792

II. Modifications to run1.py and run2.py

The functions that instantiate the cluster for SSH need two small modifications. run1.py needs to open port 11600 (arbitrary) for communication within the cluster:

```
# lines 54-60 in run1-snow.py
p1 = { 'FromPort' : 22, 'ToPort' : 22, 'IpProtocol' : 'tcp' ,
     'IpRanges' : [ { 'CidrIp' : '0.0.0.0/0', 'Description' : 'Anywhere' } ] }
p2 = { 'FromPort' : 2049, 'ToPort' : 2049, 'IpProtocol' : 'tcp' ,
     'UserIdGroupPairs' : [{ 'GroupId' : sgid } ] }
p3 = { 'FromPort' : 11600, 'ToPort' : 11600, 'IpProtocol' : 'tcp' ,
     'UserIdGroupPairs' : [{ 'GroupId' : sgid } ] }
res_auth = sg.authorize_ingress(IpPermissions=[p1, p2, p3])
```

And run2.py has a different function for writing out the batchtools.conf.R file:

```
# from run2-snow.py
def write_batchtools_config(compute):
  file = open("batchtools.conf.R", "w")
  ss = 'cluster.functions = makeClusterFunctionsSocket(c('
  for c in compute:
    ss = ss + '"' + c['private'] + '"'
    if c != compute[-1]:
      ss = ss + ','
  ss = ss + '), port=11600)\n'
  file.write(ss)
  file.close()
```

III. Run batchtools with socket cluster functions

```r
# piApprox.R
piApprox = function(n)  {
    str <- Sys.info()["nodename"]
    nums = matrix(runif(2 * n), ncol = 2)
    d = sqrt(nums[,1]^2 + nums[,2]^2)
    res <- 4 * mean(d <= 1)
    return (c(str, res))
}

piReduce <- function(x, y)  {
  ss <- paste(x[1], y[1], sep='\n')
  xx <- as.numeric(x[2]) + as.numeric(y[2])
  return (c(ss, xx))
}
```

```r
library(batchtools)
source("piApprox.R")
reg <- makeRegistry()
batchMap(fun = piApprox, n = rep(1e6, 10))
submitJobs()
waitForJobs()
v <- reduceResults(piReduce)
v[1]                    # check that all jobs ran on the remote node(s)
as.numeric(v[2])/10     # should be about 3.1416
```