parallel R
in the cloud
two lines of code

no kidding!
why...

so i've go this problem...

insurance simulations
updated frequently for one month

on my laptop...

  each sim takes ~ 1 min
  10k sims * 1 min = ~ 7 days

no need for full map/reduce
embarrassingly parallel
you've seen "word count" demos...

segue has nothing to do with that

big cpu, not big data
my options...

- make the code faster
- build a cluster
  - type
    - snow
    - mpi
    - hadoop
  - location
    - self hosted
    - amazon web services
    - ec2
    - emr
    - rackspace

lowest startup costs
syntax...

require(segue)
myCluster <- createCluster()

congratulations. we've built a hadoop cluster!
more syntax...

parallel apply() on lists:

base R:

\texttt{lapply( X, FUN, \ldots )}

segue:

\texttt{emrlapply( clusterObject, X, FUN, \ldots )}
```r
example...

```

```r
estimatePi <- function( seed ){
  set.seed(seed)
  numDraws <- 1000000
  r <- .5
  x <- runif(numDraws, min=-r, max=r)
  y <- runif(numDraws, min=-r, max=r)
  inCircle <- ifelse( (x^2 + y^2)^.5 < r , 1, 0)
  return(sum(inCircle) / length(inCircle) * 4)
}

seedList <- as.list(1:1000)
require(segue)
myCluster <- createCluster(20)
myEstimates <- emrlapply( myCluster, seedList, estimatePi )
stopCluster(myCluster)
myPi <- Reduce(sum, myEstimates) / length(myEstimates)
format(myPi, digits=10)
```

https://gist.github.com/764370
how does it work?

createCluster()

- cluster object: list of parameters
- temp dirs: local, S3 for EMR
- bootstrap: update R, update packages

~ 10-15 minutes
how does it work?

emrlapply()

list is serialized to CSV and uploaded to S3 – streaming input file

function, arguments, r objects, etc are saved & uploaded

EMR copies files to nodes – mapper.R picks them up

CSV is input to mapper.R applies function to each list element

output is serialized into emr part-xxxxx files on s3

part files are downloaded to R and deserialized

deserialized results are reordered and put into a list object
createCluster( numInstances=2,
    cranPackages,
    filesOnNodes,
    rObjectsOnNodes,
    enableDebugging=FALSE,
    instancesPerNode,
    masterInstanceType="m1.small",
    slaveInstanceType="m1.small",
    location="us-east-1a",
    ec2KeyName,
    copy.image=FALSE,
    otherBootstrapActions,
    sourcePackagesToInstall)

numInstances  number of ec2 machines to fire up
cranPackages  cran packages to load on each cluster node
filesOnNodes  files to be loaded on each node
rObjectsOnNodes  R objects to put on the worker nodes
enableDebugging  start emr debugging
instancesPerNode  number of R instances per node
masterInstanceType  ec2 instance type for the master node
slaveInstanceType  ec2 instance type for the slave nodes
location  ec2 location name for the cluster
ec2KeyName  ec2 key used for logging into the main node
copy.image  copy the entire local environment to the nodes?
onotherBootstrapActions  other bootstrap actions to run
sourcePackagesToInstall  R source packages to be installed on each node
when to use segue...

embarrassingly parallel

cpu bound

apply on lists with many items

object size: to / from s3 roundtrip

each job has a fixed & marginal cost
downside of segue...
embarrassingly parallel failure
ways to fail...

if you use segue you will see:
unreproducible errors
clusters that never start
temp buckets in your s3 acct
clusters left running
i/o that takes longer than calcs

but... i've never had a "wrong" answer
imediate segue future...

maintenance issues:
  R releases change
  emr changes

vendor lock-in to amazon
whirr as solution?
foreach %dopar% backend?
imagine the future...

R objects backed by clusters
as.hdfs.data.frame(data)

operations converted to map reduce jobs transparently

abstractions...
segue project page
http://code.google.com/p/segue/

google groups
http://groups.google.com/group/segue-r

see also...
rhipe – program m/r in R
http://www.stat.purdue.edu/~sguha/rhipe/