Parallélisme R

R est un thread unique, et en tant que tel il faut recourir à des bibliothèques pour paralléliser le travail.

Le Package *Parallel* est inclus dans R depuis *2011.* *(R V. 2.14.0).*
La fonction `parLapply` *permet de lancer les calculs parallèles en* ***cluster*** *via différents protocoles :*

PSOCK *notamment permet de faire des clusters de* ***plusieurs machines*** *:*

```
cl <- c( 'localhost', 'rstudio','xflr6')
makeCluster( cl, type='PSOCK' )
```

Sur une seule machine *on peut paralléliser entre les CPU en créant un cluster de type* ***FORK****.*

```
cl <- makeCluster(no_cores,type="FORK")
clusterExport( cl, c("maVar1", "maVar2", "maliste_a_traiter", "fonction1", "result_file") )
        … travaille en parallele dans un environnement partagé ...
stopCluster(cl)
```

*Le principe est d'exporter l'****environnement*** *(variables, functions etc.) nécessaire aux tâches qui seront réparties en le passant sous la forme d'une* ***liste****:* `clusterExport( cl, c("ground", "roof", "num", "result_file", "fibonacci") )`

En terme de code :

- La fonction `parLapply` *permet de* traiter en paralléle **une liste** et renvoie une liste de la même longueur.
Les fonctions à paralléliser doivent donc accepter une liste en entrée.

En pratique, les fonctions mono-thread traitée par la fonction `lapply` *peuvent être appelées par la fonction* `parLapply` *au sein d'un cluster. :*
Code pour mes tests:

```
system.time( capture.output( parLapply( cl, mylist, function (x) sort(mylist) ), file = result_file, append = TRUE ) )
```

La liste est générée par la ligne suivante *: mylist <- runif(num, ground, roof)*