Rozdział 10 Optymalizacja wykorzystania zasóbów komputera - obliczenia równoległe

Biblioteka ***parallel***

rm(list=ls()) #usuwanie danych  
library(parallel)  
library(knitr)  
rdzenie<-detectCores()

Na tym małym laptopie mamy 8 rdzeni. Aby sensownie na nim pracować zajmiemy 7 rdzeni pozostawiając jeden na inne zadania.

klaster <- makeCluster(rdzenie-1)

Wykorzystując klaster zostanie pokazane rozwiązanie zadania dopasowanie do modelu regresji na zbiorze Anscombe’a ([***https://pl.wikipedia.org/wiki/Kwartet\_Anscombe%E2%80%99a\_\_\_***](https://pl.wikipedia.org/wiki/Kwartet_Anscombe%E2%80%99a___)***), Utwórzona funkcja*** *\_alm()*\_\_, dla każdego wywołania, zbuduje próbę losową z powtórzeniami i wyznaczy na niej współczynniki dla modelu y3~x3. Następnie użyjemy funkcji ***parLapply()*** aby powtórzyć tę operację 100 000 razy.

N<-100000 #ilość powtórzeń  
alm <- function(i) {  
 tmp <- anscombe[sample(nrow(anscombe), replace = T),]  
 lm(y3~x3, data=tmp)$coef  
}  
wynik<-summary(system.time({  
 res <- lapply(1:N, alm)  
}))  
wynik<-as.data.frame(t(wynik))

Wynikiem funkcji ***parLapply()*** jest lista z wynikami poszczególnych wywołań funkcji ***alm()***.

w<-summary(system.time({  
 res <- parLapply(klaster, 1:N, alm)  
}))  
wynik<-rbind(wynik,t(w))  
rownames(wynik)<-c("1 rdzeń",paste(rdzenie-1,"rdzeni",sep=" "))  
  
kable(wynik)

|  |  |  |  |
| --- | --- | --- | --- |
|  | user | system | elapsed |
| 1 rdzeń | 53.36 | 0.00 | 53.50 |
| 7 rdzeni | 0.08 | 0.02 | 18.05 |

Funkcja ***parLapply()*** jako wynik zwraca listę wartości. Listę tę można uprościć czasem do wektora czasem do macierzy używając funkcji ***simplify2array()***. Aby nie kłopotać się tym dodatkowym wywołaniem, można zamiast ***parLapply()*** użyć ***parSapply()***. Po zakończeniu obliczeń równoległych należy pozamykać wiszące i zużywające RAM procesy. Aby zwinąć klaster można wykorzystać funkcję ***stopCluster()***.

stopCluster(klaster)

Jak widać obliczenia z wykorzystaniem 7 rdzeni znacznie przyspieszyło obliczenia.

# Ćwiczenia

1. Wybierz funkcję, która krótko się liczy (np. model liniowy na 11 wierszach). Wykonaj ją tysiące razy i zobacz jak wygląda czas obliczeń jako funkcja liczby użytych rdzeni.
2. Wybierz funkcję, która długo się liczy (np. model liniowy na tysiącach kolumn). Wykonaj ją kilkadziesiąt razy i zobacz jak wygląda czas obliczeń jako funkcja liczby użytych rdzeni.