A9

Elliot Gavrin

2023-03-13

https://github.com/19eag3/BIOL432_Assignment9

3.

```
library(doParallel)
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
library(parallel)
library(foreach)
```

4.

```
N <-detectCores()
N #number of cores
```

```
## [1] 8
```

There are 8 cores detected.

5-6

```
n_iter <- 4000
n_numbers <- 100000
mu <- 10
sigma <- 3
Time1 <-Sys.time()
for (i in 1:n_iter) {
  x <- rnorm(n = n_numbers, mean = mu, sd = sigma)
  mean_x <- mean(x)
}
Time2 <-Sys.time()
Etime <-(Time2-Time1)
print(Etime)
```

```
## Time difference of 59.28941 secs
```

It takes 60 seconds for the for loop to complete 7.

```
S <- 1.0 # Fraction of program that is serial
speedup <- (1/((1-S) + (S/N)))
speedup
```

```
## [1] 8
```

If the code was parallel, it would run 8 times faster than the serial code. It should take 4.275 seconds theoretically.

8.

```
Cores <- parallel::makeCluster(detectCores())
doParallel::registerDoParallel(Cores)
n_iterations <- 4000
n_samples <- 100000
Time1 <-Sys.time()
results <- foreach(i = 1:n_iterations, .combine = c) %dopar% {
  samples <- rnorm(n_samples, mean = 10, sd = 3)
  mean(samples)
}
Time2 <-Sys.time()
parallel::stopCluster(Cores)
Etime <-(Time2-Time1)
print(Etime)
```

```
## Time difference of 8.24 secs
```

The time difference of the theoretical is about 10 seconds longer than the actual run (7.4 seconds < 17.1 seconds). This is because other programs may be utilizing the cores, which causes it to not maximize all 8 cores in running the program. Because less than 8 cores are running the program, the actual time is slower than the theoretical.