

Homework 2

Problem 1

Python (Number of Cores = 8)

Number of Iterations = 2 (small)

```
In [1]: from joblib import Parallel, delayed
import time;

def time_consuming():
    time.sleep(5);

n = 2;

# Normal Computation
start1 = time.time();
for i in range(n):
    time_consuming();
end1 = time.time();
t1=end1-start1;

print("Execution Time for Normal Computation: ",t1);

# Embarrassingly Parallel Computation
start2 = time.time();
Parallel(n_jobs=8)(delayed(time_consuming)() for i in range(n))
end2 = time.time();
t2=end2-start2;

print("Execution Time for Embarrassingly Parallel Computation: ",t2)
sp=t1/t2;

print("Speedup: ", sp)
print("Average Efficiency: ", sp/8, "=", str(int((sp/8)*100)) + "%")

Execution Time for Normal Computation: 10.010491132736206
Execution Time for Embarrassingly Parallel Computation: 5.360119104385376
Speedup: 1.867587443074937
Average Efficiency: 0.23344843038436713 = 23%
```

Number of Iterations = 8 (medium)

```
In [36]: from joblib import Parallel, delayed
import time;

def time_consuming():
    time.sleep(5);

n = 8;

# Normal Computation
start1 = time.time();
for i in range(n):
    time_consuming();
end1 = time.time();
t1=end1-start1;

print("Execution Time for Normal Computation: ",t1);

# Embarrassingly Parallel Computation
start2 = time.time();
Parallel(n_jobs=8)(delayed(time_consuming)() for i in range(n))
end2 = time.time();
t2=end2-start2;

print("Execution Time for Embarrassingly Parallel Computation: ",t2)
sp=t1/t2;

print("Speedup: ", sp)
print("Average Efficiency: ", sp/8, "=", str(int((sp/8)*100)) + "%")

Execution Time for Normal Computation: 40.02420616149902
Execution Time for Embarrassingly Parallel Computation: 5.111095905303955
Speedup: 7.830846241794165
Average Efficiency: 0.9788557802242707 = 97%
```

Number of Iterations = 16 (large) (multiple of number of cores)

```
In [2]: from joblib import Parallel, delayed
import time;

def time_consuming():
    time.sleep(5);

n = 16;

# Normal Computation
start1 = time.time();
for i in range(n):
    time_consuming();
end1 = time.time();
t1=end1-start1;

print("Execution Time for Normal Computation: ",t1);

# Embarrassingly Parallel Computation
start2 = time.time();
Parallel(n_jobs=8)(delayed(time_consuming)() for i in range(n))
end2 = time.time();
t2=end2-start2;

print("Execution Time for Embarrassingly Parallel Computation: ",t2)
sp=t1/t2;

print("Speedup: ", sp)
print("Average Efficiency: ", sp/8, "=", str(int((sp/8)*100)) + "%")

Execution Time for Normal Computation: 80.0598349571228
Execution Time for Embarrassingly Parallel Computation: 10.080835103988647
Speedup: 7.941785986108018
Average Efficiency: 0.9927232482635022 = 99%
```

R (Number of Cores = 8)

Number of Iterations = 5 (small)

```
In [2]: library(tictoc)
library(doParallel)
library(scales)

n <- 5

time_consuming <- function() {
  Sys.sleep(5)
}

start1 <- Sys.time()
results = foreach(1:n) %do% time_consuming()
end1 <- Sys.time()
t1 <- end1-start1;

sprintf("Execution Time for Normal Computation: %f",t1);

myCluster <- makeCluster(8, type = "PSOCK")
registerDoParallel(myCluster)

start2 <- Sys.time()
results = foreach(1:n) %dopar% time_consuming()
end2 <- Sys.time()
t2 <- end2-start2;

stopCluster(myCluster)

sprintf("Execution Time for Embarrassingly Parallel Computation: %f",t2)
sp <- as.numeric(t1)/as.numeric(t2);

sprintf("Speedup: %f", sp)
sprintf("Average Efficiency: %f = %d%%", sp/8, as.integer((sp/8)*100))
```

'Execution Time for Normal Computation: 25.030765'

'Execution Time for Embarrassingly Parallel Computation: 5.021796'

'Speedup: 4.984425'

'Average Efficiency: 0.623053 = 62%'

Number of Iterations = 8 (medium)

```
In [58]: library(tictoc)
library(doParallel)
library(scales)

n <- 8

time_consuming <- function() {
  Sys.sleep(5)
}

start1 <- Sys.time()
results = foreach(1:n) %do% time_consuming()
end1 <- Sys.time()
t1 <- end1-start1;

sprintf("Execution Time for Normal Computation: %f",t1);

myCluster <- makeCluster(8, type = "PSOCK")
registerDoParallel(myCluster)

start2 <- Sys.time()
results = foreach(1:n) %dopar% time_consuming()
end2 <- Sys.time()
t2 <- end2-start2;

stopCluster(myCluster)

sprintf("Execution Time for Embarrassingly Parallel Computation: %f",t2)
sp <- as.numeric(t1)/as.numeric(t2);

sprintf("Speedup: %f", sp)
sprintf("Average Efficiency: %f = %d%%", sp/8, as.integer((sp/8)*100))

'Execution Time for Normal Computation: 40.056609'

'Execution Time for Embarrassingly Parallel Computation: 5.024011'

'Speedup: 7.973033'

'Average Efficiency: 0.996629 = 99%'
```

Number of Iterations = 14 (large) (not a multiple of number of cores)

```
In [10]: library(tictoc)
library(doParallel)
library(scales)

n <- 14

time_consuming <- function() {
  Sys.sleep(5)
}

start1 <- Sys.time()
results = foreach(1:n) %do% time_consuming()
end1 <- Sys.time()
t1 <- (end1-start1)*60;

sprintf("Execution Time for Normal Computation: %f",t1);

myCluster <- makeCluster(8, type = "PSOCK")
registerDoParallel(myCluster)

start2 <- Sys.time()
results = foreach(1:n) %dopar% time_consuming()
end2 <- Sys.time()
t2 <- end2-start2;

stopCluster(myCluster)

sprintf("Execution Time for Embarrassingly Parallel Computation: %f",t2)
sp <- as.numeric(t1)/as.numeric(t2);

sprintf("Speedup: %f", sp)
sprintf("Average Efficiency: %f = %d%%", sp/8, as.integer((sp/8)*100))

'Execution Time for Normal Computation: 70.069006'

'Execution Time for Embarrassingly Parallel Computation: 10.028468'

'Speedup: 6.987010'

'Average Efficiency: 0.873376 = 87%'
```

Problem 2

Code (Number of Cores = 8)

```
clear;

if isempty(gcp())
    parpool();
end

% Number of Iterations
n = 2;

% Normal Computation
tic
for i=1:n
    timeconsumingfun;
end
t1 = toc;

x = sprintf("Execution Time for Normal Computation: %f", t1);
disp(x);

% Embarrassingly Parallel Computation
tic
parfor i=1:n
    timeconsumingfun;
end
t2 = toc;

y = sprintf("Execution Time for Embarrassingly Parallel Computation: %f", t2);
disp(y);

sp = t1/t2;
z = sprintf("Speedup: %f", sp);
disp(z);
v = sprintf("Average Efficiency: %f = %d%%", sp/8, int8((sp/8)*100));
disp(v);

function timeconsumingfun
    pause(5)
end
```

Number of Iterations = 2 (small) (without disturbance)

Execution Time for Normal Computation: 10.006140

Execution Time for Embarrassingly Parallel Computation: 5.215397

Speedup: 1.918577

Average Efficiency: $0.239822 = 24\%$

Number of Iterations = 8 (medium) (without disturbance)

Execution Time for Normal Computation: 40.042790

Execution Time for Embarrassingly Parallel Computation: 5.198092

Speedup: 7.703364

Average Efficiency: $0.962920 = 96\%$

Number of Iterations = 15 (large) (without disturbance)

Execution Time for Normal Computation: 75.066611

Execution Time for Embarrassingly Parallel Computation: 10.204815

Speedup: 7.355999

Average Efficiency: $0.919500 = 92\%$

Code (Number of Cores = 8)

```
clear;

if isempty(gcp())
    parpool();
end

% Number of Iterations
n = 2;

t1 = 10.006140; % Execution Time of Normal Computation

% Embarrassingly Parallel Computation
tic
parfor i=1:n
    timeconsumingfun;
end
t2 = toc;

y = sprintf("Execution Time for Embarrassingly Parallel Computation:
%f", t2);
disp(y);

sp = t1/t2;
z = sprintf("Speedup: %f", sp);
disp(z);
v = sprintf("Average Efficiency: %f = %d%%", sp/8, int8((sp/8)*100));
disp(v);

function timeconsumingfun
    pause(5)
end
```

Number of Iterations = 2 (small) (with disturbance)

Execution Time for Normal Computation: 10.006140
Execution Time for Embarrassingly Parallel Computation: 5.984629
Speedup: 1.671973
Average Efficiency: 0.208997 = 21%

Number of Iterations = 8 (medium) (with disturbance)

Execution Time for Normal Computation: 40.042790
Execution Time for Embarrassingly Parallel Computation: 7.080541
Speedup: 5.655329
Average Efficiency: 0.706916 = 71%

Number of Iterations = 15 (large) (with disturbance)

Execution Time for Normal Computation: 75.066611
Execution Time for Embarrassingly Parallel Computation: 12.380736
Speedup: 6.063179
Average Efficiency: 0.757897 = 76%

Result: While executing the code with disturbance (6 Chrome tabs with 8K HDR 60FPS FUHD Videos), the speedup and the average accuracy are affected.

Problem 3

Code (Number of Cores = 8)

```
clear;

if isempty(gcp())
    parpool();
end

% Number of Iterations
n = 2;

% Normal Computation
tic
for i=1:n
    timeconsumingfun;
end
t1 = toc;

x = sprintf("Execution Time for Normal Computation: %f",t1);
disp(x);

% Embarrassingly Parallel Computation
tic
parfor i=1:n
    timeconsumingfun;
end
t2 = toc;

y = sprintf("Execution Time for Embarrassingly Parallel Computation:
%f",t2);
disp(y);

sp = t1/t2;
z = sprintf("Speedup: %f", sp);
disp(z);
v = sprintf("Average Efficiency: %f = %d%%", sp/8, int8((sp/8)*100));
disp(v);

function timeconsumingfun
    pause(randi([1,5],1,1))
end
```


Number of Iterations = 2 (small)

Execution Time for Normal Computation: 8.010428

Execution Time for Embarrassingly Parallel Computation: 4.134746

Speedup: 1.937345

Average Efficiency: $0.242168 = 24\%$

Number of Iterations = 8 (medium)

Execution Time for Normal Computation: 25.035484

Execution Time for Embarrassingly Parallel Computation: 5.188583

Speedup: 4.825110

Average Efficiency: $0.603139 = 60\%$

Number of Iterations = 15 (large)

Execution Time for Normal Computation: 47.063417

Execution Time for Embarrassingly Parallel Computation: 9.145267

Speedup: 5.146205

Average Efficiency: $0.643276 = 64\%$