

## Przetwarzanie współbieżne – programowanie równoległe

### Laboratorium 13.

Cel:

Opanowanie podstaw programowania z przesyłaniem komunikatów MPI.

Kroki:

1. Utworzenie katalogu roboczego (np. lab13).
2. Wygenerowanie kluczy ssh w celu połączenia bez podawania hasła:  

```
>> ssh-keygen  
>> cd ~/.ssh  
>> cp id_rsa.pub authorized_keys
```
3. Połączenie przez ssh z każdym komputerem, który będziemy chcieli dołączyć do klastra w celu dodania go do bazy kluczy.
4. Stworzenie pliku hostfile wg. składni:  

```
idkomp1 slots=liczba_procesów  
idkomp2 slots=liczba_procesów
```
5. Skopiowanie przykładowego programu (mpi\_simple.c), jego kompilacja (mpicc, lub /usr/lib64/openmpi/bin/mpicc) oraz uruchomienie (mpiexec lub mpirun, ew. /usr/lib64/openmpi/bin/mpirun) na co najmniej trzech maszynach (adresy komputerów z pliku hostname przekazywane do mpiexec w ramach argumentu -machinefile; np. mpiexec -machinefile hostname -np 3 a.out).
6. Uzupełnienie programu o przesyłanie w tablicy znaków adresu internetowego węzła nadawcy i wypisywanie adresu u odbiorcy (adres można pobrać funkcją gethostname).
7. Opracowanie programu propagującego liczbę w konwencji pierścienia (sztafeta). Wysyłana liczba powinna być równa dziesięciokrotności numeru procesu ją wysyłającego, tj proces numer  $n$  powinien odebrać od procesu  $n-1$  liczbę  $(n-1)*10$  i przysłać do procesu  $n+1$  liczbę  $n*10$ . Ostatni proces powinien przesłać dane do procesu pierwszego.
8. Program powinien wyświetlić podstawowe informacje zgodnie z przykładem (dla dziesięciu procesów):  

```
...  
Proces 2 odebrał liczbę 10 od procesu 1 na .....  
Proces 2 wysłał liczbę 20 do procesu 3 na .....  
...  
itp.
```

Dalsze kroki:

Zadanie z pkt. 7 rozwiązać stosując komunikację nieblokującą.

Warunki zaliczenia:

1. Obecność na zajęciach i wykonanie kroków 1-7.
2. Oddanie sprawozdania z opisem zadania, kodem źródłowym programów, wynikami i wnioskami.