Programowanie równoległe i rozproszone
Kierunek Informatyka, WE, sem. VII

Laboratorium nr 6

Celem zajęć zapoznanie się z podstawowym mechanizmem komunikacji pomiędzy procesami z wykorzystaniem interfejsu MPI.

Wymagania:

- Instalacje platformy Docker
- Instalacja kontenera środowiska OpenMPI

Aby zaliczyć laboratorium **przygotuj sprawozdanie**, w którym umieść odpowiedzi do poleceń. Sprawozdanie **prześlij do odpowiedniego ćwiczenia** w iliasie.

A. Przygotowanie środowiska MPI

1. Instalacja platformy Docker

   Zainstaluj Docker Desktop lub zaktualizuj do najnowszej wersji
   Zalecana wersja (wersja >= 4.2.x).

2. Instalacja środowiska mpi na docker

   1) Pobranie obraz brute/brute_openmpi
      Źródło obrazu znajdue się pod
      adresem: https://hub.docker.com/r/brute/brute_openmpi

      Wykonaj komendę: docker pull brute/brute_openmpi

   2) Utwórz w wybranym katalogu plik Dockerfile.conf o następującej zawartości:

```
# bazowy obraz to brute_openmpi
FROM brute/brute_openmpi:latest
# czyszczenie cache dla apt-get
RUN apt-get clean
# update, instalacja serwera i klienta ssh oraz pakietu sudo
RUN apt-get update && apt-get install  openssh-server openssh-client net-tools sudo dnsutils  -y
EXPOSE 22
RUN  /etc/init.d/ssh start
CMD  ["/etc/init.d/ssh","start"]
```

Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

3) W komand line Wykonaj komendę utworzenia własnego obrazu o nazwie my_openmpi:
docker build --no-cache -t my_openmpi - < Dockerfile.conf

Uwaga! Ten obraz będzie nam potrzebny na kolejnych zajęciach. Proszę o tym pamiętać!

4) Sprawdź, czy w obrazach pojawił się obraz my_openmpi

3. Uruchomienie kontenera mpi-master na bazie utworzonego obrazu my_openmpi:

docker run --name mpi-master --hostname mpi-master -it my_openmpi /bin/bash

Sprawdź, czy kontener został uruchomiony.

4. Przeloguj się na konto studenta i przejdź do katalogu studenta, w którym utwórz podkatalog lab6

B. Programowanie z wykorzystaniem komunikatów MPI

1. W katalogu lab6 utwórz nowy plik przyklad1.c, w którym umieścisz kod programu przygotowany języku C:

```
#include "mpi.h"
#include <stdio.h>
int main(int argc, char *argv[])
{
 int numtasks, rank, rc;

 rc = MPI_Init(&argc,&argv);
 if (rc != 0)
  {
   printf ("Wystapil blad podczas startowania programu. Przerywam dzialanie.\n");
   MPI_Abort(MPI_COMM_WORLD, rc);
  }
 MPI_Comm_size(MPI_COMM_WORLD,&numtasks);
 MPI_Comm_rank(MPI_COMM_WORLD,&rank);
 printf ("Ilosc procesow= %d Jestem procesem= %d\n", numtasks,rank);

  /******* inna użyteczna praca *******/

 MPI_Finalize();
}
```

Do edycji plików z kodem użyj np. edytora nano

2. Skompiluj program

   mpicc -o przyklad1 przyklad1.c

3. Uruchom program przyklad1

   mpirun przyklad1

   Co otrzymałeś?

   A teraz uruchom program w parametrem -n:
   mpirun -n 3 przyklad1

   Jaka jest różnica pomiędzy tymi uruchomieniami?
   Uruchom program kilkakrotnie i zaobserwuj co się dzieje.
   Odpowiedź umieść w sprawozdaniu.

4. Skopiuj kod z przyklad1.c do pliku przyklad2.c i dopisz to tego kodu następującą linijkę kodu:

   MPI_Get_processor_name(nodename, &resultlen);

   Gdzie umieścisz tę linie kodu?
   Zanim skompilujesz program zdeklaruj zmienne nodename oraz resultlen.

   Co teraz uzyskałeś? Odpowiedź umieść w sprawozdaniu.

5. Co jest zadaniem następujących funkcji:

   - MPI_Init()
   - MPI_Comm_rank()
   - MPI_Comm_size()
   - MPI_Get_processor_name()
   - MPI_Finalize()

   Znajdź odpowiedzi I umieść je w sprawozdaniu.

6. Jaką role pełni (do czego służy) komunikator MPI_COMM_WORLD? Znajdź odpowiedzi na pytania. Odpowiedź umieść w sprawozdaniu.

Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

7. Utwórz kolejne plik i nazwij je odpowiednio:
   - przyklad3.c a następnie umieść w nim kod z send_recv z tutorialu: https://github.com/mpitutorial/mpitutorial/blob/gh-pages/tutorials/mpi-send-and-receive/code/send_recv.c
   - przyklad4.c a następnie umieść w nim kod z ping_pong z tutorialu: https://github.com/mpitutorial/mpitutorial/blob/gh-pages/tutorials/mpi-send-and-receive/code/ping_pong.c

   Następnie sformułuj odpowiedzi do pytań:
   1) Skompiluj kody, uruchom programy oraz dokonaj ich analizy. Co robią te programy?
   2) Jak działają procedury MPI_Send i MPI_Recv?
   3) Każdy z programów przedstaw w postaci schematu blokowego (zastosuj odpowiedni diagram UML).
   4) Zastanów się, czy procedury MPI_Send i MPI_Recv można zamienić na MPI_Isend and na MPI_Irecv. Czy taka operacja ma sens? Jakie jest działanie procedur MPI_Isend oraz MPI_Irecv. Jeśli uda się Tobie podmienić te funkcje opisz co należało zmienić i jak teraz działają programy? Zrzuty zmodyfikowanych programów (ich kodu) umieść w sprawozdaniu.

Dla podpowiedzi, standardowa procedura wysyłanie MPI_Send obejmuje:

MPI_Send(buf, count, datatype, dest, tag, comm)
- buf - zmienna zawierająca wysyłane dane
- count - liczba wysyłanych danych
- datatype - typ wysyłanych danych (patrz niżej)
- dest - rząd (numer) procesu celowego
- tag - identyfikator („pieczątka”) wiadomości
- comm – komunikator

gdy standardowa procedura odbierania MPI_Recv jest składową:

MPI_Recv(buf, count, datatype, source, tag, comm, status)
- buf - zmienna, do której przychodzą przyjmowane dane
- count - liczba elementów przychodzących danych
- datatype - typ przychodzących danych
- source - rząd (numer) procesora, który wysłał wiadomość
- tag - identyfikator wiadomości (jeżeli jest zaspecyfikowany explicite, zostanie odebrana wiadomość posiadająca ten właśnie identyfikator)
- comm - komunikator
- status - tablica o rozmiarze MPI_STATUS_SIZE zawierająca informacje o przychodzącej wiadomości

Jak „wyglądają” procedury MPI_Isend and na MPI_Irecv poszukaj już sam.

Projekt „SezAM wiedzy, kompetencji i umiejętności” jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Szukając odpowiedzi na inne pytania możesz również skorzystać z dostępnych tutoriali (np. http://staff.iiar.pwr.wroc.pl/wojciech.bozejko/Web_MPI/ lub innych)

Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój