

Zadania projektowe do przedmiotu PORR (na semestr zimowy 2019/2020 roku)  
prowadzący: Ewa Niewiadomska-Szynkiewicz (e-mail: [ens@ia.pw.edu.pl](mailto:ens@ia.pw.edu.pl), pokój 572A),  
konsultacje: wtorki godzina **14.10 – 16.00**

Dwa warianty projektów – **zakłada się 3-osobowe zespoły projektowe**

#### Wariant 1:

Należy zaproponować sposób zrównoleglenia i przygotować następujące aplikacje:

- wersja sekwencyjna działająca na pojedynczym procesorze
- wersja równoległa na maszynie z pamięcią wspólną
- wersja równoległa wykorzystująca procesory graficzne
- wersja równoległa na maszynie z pamięcią lokalną

#### Wariant 2:

Należy rozpoznać i zainstalować wskazane środowisko wspierające obliczenia równoległe i rozproszone, zrealizować zadany przykład i wykonać badania.

#### Informacje ogólne

Osoby zainteresowane realizacją wymienionych zadań projektowych mogą zgłaszać chęć wykonania projektu w ankiecie DOODLE

[https://doodle.com/poll/bkkggdvwqszcq6zp?utm\\_campaign=poll\\_created&utm\\_medium=email&utm\\_source=poll\\_transactional&utm\\_content=inviteparticipants-cta#email](https://doodle.com/poll/bkkggdvwqszcq6zp?utm_campaign=poll_created&utm_medium=email&utm_source=poll_transactional&utm_content=inviteparticipants-cta#email)

kolejne dni odpowiadają numerom zadań projektowych, np. Nov. 1 oznacza zadanie nr. 1.  
Można też zgłosić się bezpośrednio w czasie konsultacji. Zadania będą przydzielane zgodnie z kolejnością zgłoszeń.

Po dodatkowe materiały pomocnicze proszę się zgłaszać w terminie konsultacji.

**Zasady zaliczenia.** Za projekt można otrzymać maksymalnie 40 punktów. Projekt jest oddawany w dwóch etapach:

- 1) Wersja sekwencyjna i zrównoleglona (dyrektywy OpenMP) - 20 pkt.  
Termin oddania do **17.12.2018.**
- 2) Wersja równoległa (GPU i pamięć lokalna) - 20 pkt.  
Termin oddania do **20.01.2020.**

Za opóźnienie w oddawaniu projektu będzie obniżana ocena o **2 punkty** za każdy tydzień.

W celu oddania projektu zapraszam w terminie swoich konsultacji.

Wymagania: działający projekt + sprawozdanie z opisem i wynikami obliczeń (należy przesłać plik ze sprawozdaniem przed dniem oddawania projektu).

*Nazwiska przy zadaniu oznaczają, że zostało już ono przydzielone*

### *Zadania:*

#### **Wariant 1: Algorytmy - wersja sekwencyjna oraz zrównoleglenie z wykorzystaniem trzech mechanizmów:**

- zrównoleglenie z wykorzystaniem mechanizmu na maszynie z pamięcią wspólną: **OpenMP** (lub inne)
- zrównoleglenie z wykorzystaniem procesorów graficznych - wybrać jedno ze środowisk: **OpenACC, CUDA, OpenCL**
- zrównoleglenie z wykorzystaniem mechanizmu na maszynie z pamięcią lokalną: **MPI** (lub inne) lub realizacja w środowisku **Apache Hadoop** lub **Apache Spark**

1. **Analiza tekstów - wyszukiwanie wzorca w tekście: Algorytmy Morrisa-Patta i Knutha-Morrisa-Patta do wyszukiwania wzorca w tekście.** Zaimplementować i porównać algorytmy MP i KMP <http://smurf.mimuw.edu.pl/node/569> w wersjach sekwencyjnej i równoległej. Zastosować te algorytmy do wyszukania wszystkich pozycji zadanego wzorca w tekście. Rozważyć dwa przykłady o różnym skomplikowaniu (dłuższy i krótszy wzorzec i różna liczba wystąpień). Ocenić wpływ zrównoleglenia na szybkość algorytmu. . (grupa 3 osobowa)
2. **Analiza tekstów - wyszukiwanie wzorca w tekście: Algorytm Boyera-Moore'a do wyszukiwania wzorca w tekście.** Zaimplementować algorytm Boyera-Moore'a <http://www.cs.utexas.edu/~moore/publications/fstrpos.pdf> w wersji sekwencyjnej i równoległej. Zastosować ten algorytm do wyszukania wszystkich pozycji zadanego wzorca w tekście. Rozważyć dwa przykłady o różnym skomplikowaniu (dłuższy i krótszy wzorzec i różna liczba wystąpień). Ocenić wpływ zrównoleglenia na szybkość algorytmu. . (grupa 3 osobowa)
3. **Analiza tekstów - budowa grafów i drzew oraz ich przeszukiwanie: Drzewo sufiksowe i graf podslów.** Zaimplementować:
  - drzewo sufiksowe <http://smurf.mimuw.edu.pl/node/578>
  - graf podslów <http://smurf.mimuw.edu.pl/node/581>W wersjach sekwencyjnej i równoległych. Rozważyć dwa przykłady o różnej długości tekstu (dłuższy i krótszy). Wyszukać najdłuższe podslowo, które występuje k=2, 5, 10 razy w danym tekście. Ocenić wpływ zrównoleglenia na szybkość algorytmu. . (grupa 3 osobowa)
4. **Optymalizacja: Metody przeglądu zupełnego i symulowanego wyżarzania do rozwiązania zadania komiwojażera.** Rozwiązać zadanie komiwojażera (wyznaczyć kolejność miast, które należy odwiedzić) stosując metodę przeglądu zupełnego i symulowanego wyżarzania. Zrealizować zadanie w wersji sekwencyjnej i równoległej. Sformułować i rozwiązać dwa zadania o różnym poziomie złożoności (duże i małe). Ocenić przyspieszenie obliczeń w zależności od wymiaru zadania i stopnia zrównoleglenia. Przedstawić graficznie zbieżność obu algorytmów w wersji sekwencyjnej – wartości funkcji celu w kolejnych iteracjach. . (grupa 3 osobowa)

5. **Optymalizacja: Strategia ewolucyjna - algorytm koewolucyjny.** Wersja sekwencyjna - strategia ewolucyjna z kodowaniem rzeczywistoliczbowym. Algorytm w wersji koewolucyjnej – uruchamiamy strategie ewolucyjne na kilku populacjach punktów, populacje co pewien okres czasu wymieniają między sobą informacje. Rozwiązać dwa zadania optymalizacji Zad. 1 i Zad. 2 dla  $n=2, 10, 20, 50, 100$ . Ocenic przyspieszenie obliczeń w zależności od stopnia zrównoleglenia. Zaprezentować graficznie przebieg algorytmu (sekwencyjny i równoległy) dla przypadku 2-wymiarowego – interfejs graficzny (wyświetlać populacje z kolejnych iteracji na poziomicach funkcji celu). Uwaga: przy badaniach wydajności nie wykorzystywać graficznego interfejsu. (grupa 3 osobowa)

Zad. 1:

$$\min_x \left( f(x) = \frac{1}{40} \sum_{i=1}^n (x_i^2) + 1 - \prod_{i=1}^n \cos\left(\frac{x_i}{i}\right) \right) \\ -40 \leq x_i \leq 40 \quad i = 1, \dots, n$$

Minimum globalne  $f_{min} = 0$  w punkcie  $x_i = 0$

Zad. 2:

$$\min_x \left( 20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e \right), \quad -30 \leq x_i \leq 30$$

Minimum globalne  $f_{min} = 0$  w punkcie  $x_i = 0$

6. **Optymalizacja: Algorytm świetlika (FA) i roju świetlików (GSO).**

Algorytm świetlika i roju świetlików (<http://pe.org.pl/articles/2014/6/37.pdf>, [http://home.agh.edu.pl/~slukasik/pub/021\\_Lukasik\\_KAEiOG2011\(presentation\).pdf](http://home.agh.edu.pl/~slukasik/pub/021_Lukasik_KAEiOG2011(presentation).pdf), [https://eti.pg.edu.pl/documents/176546/25263566/SZ\\_wyklad2.pdf](https://eti.pg.edu.pl/documents/176546/25263566/SZ_wyklad2.pdf)). Zrealizować wersje sekwencyjne i równoległe algorytmów. Rozwiązać dwa zadania optymalizacji Zad. 1 i Zad. 2 dla  $n=2, 10, 20, 50, 100$ . Ocenic przyspieszenie obliczeń w zależności od stopnia zrównoleglenia. (porównać obie metody). Przedstawić graficznie zbieżność obu algorytmów w wersji sekwencyjnej – wartości funkcji celu w kolejnych iteracjach. Uwaga: przy badaniach wydajności nie wykorzystywać graficznego interfejsu. (grupa 3 osobowa)

Zad. 1:

$$\min_x \left( f(x) = \frac{1}{40} \sum_{i=1}^n (x_i^2) + 1 - \prod_{i=1}^n \cos\left(\frac{x_i}{i}\right) \right) \\ -40 \leq x_i \leq 40 \quad i = 1, \dots, n$$

Minimum globalne  $f_{min} = 0$  w punkcie  $x_i = 0$

Zad. 2:

$$\min_x \left( f_1(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \right) \quad f_{min} = 0 \\ \sum_{i=1}^n (x_i - i)^2 \leq n \cdot 10 \\ -40 \leq x_i \leq 40$$

Minimum globalne  $f_{min} = 0$

7. **Optymalizacja: Metoda optymalizacji rojem cząstek (PSO) do wyznaczenia optymalnej ścieżki dojazdu do celu dla robota mobilnego.**  
 PSO ([http://atol.am.gdynia.pl/~tomera/publikacje/PTETiS\\_2015a\\_Tomera.pdf](http://atol.am.gdynia.pl/~tomera/publikacje/PTETiS_2015a_Tomera.pdf), <http://aragorn.pb.bialystok.pl/~wkwedlo/EA6.pdf>). Robot porusza się po określonych trasach. Zadanie polega na wyznaczeniu najkrótszej (w sensie odległości) ścieżki dojazdu do wskazanego punktu w przestrzeni roboczej. Do punktu prowadzi kilka ścieżek złożonych z odcinków o różnych długościach (robot porusza się po grafie). Rozwiązać dwa zadania o różnej złożoności. Zrealizować zadania w wersji sekwencyjnej i równoległej. Ocenić przyspieszenie obliczeń w zależności od stopnia zrównoleglenia (porównać obie metody). Przedstawić graficznie zbieżność obu algorytmów w wersji sekwencyjnej – wartości funkcji celu w kolejnych iteracjach. . (grupa 3 osobowa)
  
8. **Optymalizacja: Algorytm mrówkowy (ACO – Ant Colony Optimization) do wyznaczania routingu pakietów w sieciach komputerowych.** Zaimplementować algorytm ACO ([http://atol.am.gdynia.pl/~tomera/publikacje/PTETiS\\_2015a\\_Tomera.pdf](http://atol.am.gdynia.pl/~tomera/publikacje/PTETiS_2015a_Tomera.pdf)) w wersjach sekwencyjnej i równoległej. Zastosować ten algorytm do wyznaczenia optymalnych tablic routingu pakietów w sieciach komputerowych. Rozważyć dwa przykłady sieci (mniej i bardziej złożone). Ocenić wpływ zrównoleglenia na szybkość algorytmu. Przedstawić graficznie zbieżność obu algorytmów w wersji sekwencyjnej – wartości funkcji celu w kolejnych iteracjach. . (grupa 3 osobowa)
  
9. **Algorytm szyfrujący AES** (*Advanced Encryption Standard*) roboczo nazywany Rijndael. Jest to symetryczny algorytm z kluczem. Szyfruje i deszyfruje dane w 128-bitowych blokach, korzystając z 128, 192 lub 256 bitowych kluczy. Należy zapoznać się z algorytmem, zrealizować w wersji sekwencyjnej. Wybrać tryb szyfrowania najlepszy dla zrównoleglenia. Następnie zaproponować sposób zrównoleglenia i zrealizować w dwóch wersjach. . (grupa 3 osobowa)

## **Wariant 2: Środowiska wspierające obliczenia równoległe i rozproszone**

10. **Symulacja optymalnej alokacji maszyn wirtualnych w środowisku do modelowania i symulacji chmur obliczeniowych CloudSim** <http://www.cloudbus.org/cloudsim/>. Rozpoznać środowisko i je zainstalować. Zaproponować architekturę przykładowej chmury i zdefiniować listę zadań do wykonania na zadanym horyzoncie - każde zadanie to maszyna wirtualna (VM). Zasyмуляć działanie tej chmury. Porównać różne metody alokacji maszyn wirtualnych. Rozważyć możliwość migracji maszyn – wykonać symulacje różnych scenariuszy.
  
11. Środowiska Apache Hadoop i Apache Spark (<http://www.i3s.unice.fr/~jplozi/cmpt732/>, <https://courses.cs.washington.edu/courses/cse490h/08au/projects/original/assignment1.pdf>). Rozpoznać środowiska. Zbudować prostą sieć społeczną: klucz- ID, wartość – osoba. Zrealizować w obu środowiskach algorytm MapReduce zwracający liczby przyjaciół każdej z osób w sieci. Ocenić wygodę i łatwość implementacji w obu środowiskach. . (grupa 3 osobowa)