**Nazwa przedmiotu:**

Programowanie równoległe i rozproszone

**Koordynator przedmiotu:**

Ewa Niewiadomska-Szynkiewicz, Andrzej Karbowski, Bartłomiej Kubica

**Status przedmiotu:**

Fakultatywny ograniczonego wyboru

**Poziom kształcenia:**

Studia II stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne - zaawansowane

**Kod przedmiotu:**

PORR

**Semestr nominalny:**

4 / rok ak. 2012/2013

**Liczba punktów ECTS:**

5

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

125

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

3

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

2

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 30h |
| Ćwiczenia: | 0h |
| Laboratorium: | 0h |
| Projekt: | 30h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Znajomość języków programowania (C, C++).
Znajomość podstaw programowania równoległego.

**Limit liczby studentów:**

48

**Cel przedmiotu:**

Celem przedmiotu jest przedstawienie podstawowych wiadomości na temat prowadzenia obliczeń wektorowych, równoległych i rozproszonych na komputerach wielordzeniowych (wykorzystujących ewentualnie rozszerzenia strumieniowe SSE, karty graficzne), superkomputerach, a także w klastrach i gridach. W dalszej części wykładu omawiane są przykłady zastosowania obliczeń równoległych do rozwiązywania złożonych zadań analizy numerycznej - klasycznych oraz związanych z Internetem.
Celem projektu jest zdobycie podstawowych praktycznych umiejętności w posługiwaniu się równoległym środowiskiem do obliczeń oraz wykonanie przykładowych obliczeń na maszynach równoległych, wielordzeniowych (także z wykorzystaniem SSE i GPU), jak również w klastrze stacji roboczych. Przewidywane są zadania związane z:
1) Badaniem algorytmów synchronicznych z wykorzystaniem dyrektyw zrównoleglających kompilatora (OpenMP) oraz mechanizmu wątków (POSIX lub Java threads) na maszynie równoległej z pamięcią wspólną (4-procesorowy Sun);
2) Badaniem algorytmów rozproszonych w klastrze z wykorzystaniem oprogramowania: MPI, RPC, Java RMI, UPC;
3) Badaniem efektywności obliczeń hybrydowych - ze zrównolegleniem na wiele rdzeni oraz simdyzacją (SSE, GPU)
4) Oceną efektywności różnych narzędzi do zrównoleglania programów napisanych w: Javie, Matlabie, C# uruchamianych na maszynie wielordzeniowej pracującej pod kontrolą systemu UNIX/Linux albo w sieci PC pod kontrolą MS Windows

**Treści kształcenia:**

Zagadnienia podstawowe: klasyfikacja i architektura komputerów równoległych; procesory wielordzeniowe o strukturze symetrycznej i asymetrycznej (CELL), jednostki wykonawcze SSE/Altivec oraz GPU; obliczenia: wektorowe, współbieżne, równoległe, rozproszone, strumieniowe; rodzaje oprogramowania realizującego równoległość, istotne paradygmaty i modele programowania równoległego.
Miary oceny efektywności obliczeń równoległych (współczynniki przyśpieszenia oraz wydajności, prawa Amdahla i Gustafsona-Barsisa, sprawność i skalowalność).
Zagadnienia synchronizacji i wymiany informacji w obliczeniach równoległych, podstawowe mechanizmy: zamek, semafor, monitor, bariera klasyczna i dwuczęściowa, zmienne warunków, komunikaty (przesyłanie: synchroniczne, asynchroniczne, blokujące, nieblokujące, buforowane, itd.).
Wektoryzacja obliczeń we współczesnych komputerach opartych na architekturze x86, sposób wykorzystania jednostek wykonawczych SSE.
Podstawowe informacje o obliczeniach ogólnego przeznaczenia wykorzystujących karty graficzne (GPGPU), pojęcia strumienia i jądra; najważniejsze cechy środowisk oprogramowania: Brook, (MS-R) Accelerator, CUDA, OpenCL.
Elementy programowania na maszynach z pamięcią wspólną, narzędzia: klasyczne narzędzia systemu UNIX, programowania wielowątkowego (wątki POSIX, wątki w językach Java oraz C#), język dyrektyw OpenMP.
Elementy programowania na maszynach z pamięcią lokalną oraz w sieciach komputerowych, klastrach i gridach; narzędzia: środowisko MPI, rodzina narzędzi RPC.
Sposoby realizacji rozproszonych systemów z pamięcią wspólną: Linda, Co-Array Fortran, UPC.
Algorytmy synchroniczne: podstawowe algorytmy algebry liniowej w wersji równoległej, rozwiązywanie układów równań nieliniowych, równoległe metody optymalizacji.
Algorytmy całkowicie lub częściowo asynchroniczne: założenia, zbieżność, zastosowanie do rozwiązywania dużych układów równań liniowych i nieliniowych, optymalizacji statycznej, routingu, szeregowania linków w wyszukiwarkach, itp.

**Metody oceny:**

Zaliczenie egzaminu.
Zaliczenie projektu (wykonanie dwóch zadań).

**Egzamin:**

tak

**Literatura:**

Praca zbiorowa pod red. A. Karbowskiego i E. Niewiadomskiej-Szynkiewicz, Programowanie Równoległe i Rozproszone (pozostali autorzy: J. Błaszczyk, M. Kamola, B. Kubica, K. Malinowski, M. Warchoł), Oficyna Wydawnicza PW, 2009.
Grama A., Kumar V., Gupta A., Karypis G., Introduction to Parallel Computing, 2/E, Addison Wesley , 2003.
Dongarra J, et al., The Sourcebook of Parallel Computing, Morgan Kaufmann (Elsevier), 2003.
Foster I. and Kesselman C. (Eds.), The Grid 2: Blueprint for a New Computing Infrastructure (Second edition), Morgan Kaufmann (Elsevier), 2004.
Bertsekas D.P. i Tsitsiklis J.N.,Parallel and Distributed Computation: Numerical Methods, Athena Scientific, 1997.

**Witryna www przedmiotu:**

https://studia.elka.pw.edu.pl/priv/11Z/PORR.A/

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt Wpisz opis:**

Ma wiedzę na temat prowadzenia obliczeń wektorowych, równoległych i rozproszonych na komputerach wielordzeniowych (wykorzystujących ewentualnie rozszerzenia strumieniowe SSE, karty graficzne), superkomputerach, a także w klastrach i gridach. Zna podstawowe narzędzia i srodowiska do programowania równoległego i rozproszonego. Ma wiedzę dotyczacą sposobów zrównoleglania zadań analizy numerycznej - klasycznych oraz związanych z Internetem.

Weryfikacja:

Egzamin obejmujący materiał z wykładu. Wykonanie zadań projektowych.

**Powiązane efekty kierunkowe:** K\_W03, K\_W04, K\_W05, K\_W07, K\_W08, K\_W09, K\_W10

**Powiązane efekty obszarowe:** T2A\_W04, T2A\_W05, T2A\_W02, T2A\_W04, T2A\_W05, T2A\_W07, T2A\_W03, T2A\_W03, T2A\_W04, T2A\_W05, T2A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt Wpisz opis:**

Potrafi pozyskiwać informacje z literatury krajowej i zagranicznej oraz dostępnych baz danych. Potrafi zaprojektować i wykonać system oprogramowania w równoległym i rozproszonym środowisku sprzętowym. Potrafi przeprowadzić testy wykonanego oprogramowania, dokonać jego oceny i przedstawić wyniki w postaci odpowiednio przygotowanej dokumentacji.

Weryfikacja:

Ocena wykonanego oprogramowania i dokumentacji z wynikami badań.

**Powiązane efekty kierunkowe:** K\_U01, K\_U03, K\_U04, K\_U05, K\_U08, K\_U09, K\_U11, K\_U13

**Powiązane efekty obszarowe:** T2A\_U01, T2A\_U03, T2A\_U05, T2A\_U07, T2A\_U09, T2A\_U12, T2A\_U11, T2A\_U16, T2A\_U18

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt Wpisz opis:**

Rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe. Rozumie potrzebę i zna możliwości dalszego dokształcania się. Potrafi pracować indywidualnie i w zespole informatyków, w tym także potrafi zarządzać swoim czasem oraz podejmować zobowiązania i dotrzymywać terminów.

Weryfikacja:

Ocena pracy podczas wykonywania projektu.

**Powiązane efekty kierunkowe:** K\_K01

**Powiązane efekty obszarowe:** T2A\_K06