

**AGH**AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Nazwa modułu: Systemy równoległe i rozproszone

Rok akademicki: 2018/2019 Kod: JIS-2-107-GK-s Punkty ECTS: 5

Wydział: Fizyki i Informatyki Stosowanej

Kierunek: Informatyka Stosowana Specjalność: Grafika komputerowa i przetwarzanie obrazów

Poziom studiów: Studia II stopnia Forma i tryb studiów: Stacjonarne

Język wykładowy: Polski Profil kształcenia: Ogólnoakademicki (A) Semestr: 1

Strona www: <http://www.fis.agh.edu.pl/~gronek/>

Osoba odpowiedzialna: dr inż. Gronek Piotr (gronek@agh.edu.pl)

Osoby prowadzące: dr inż. Dydejczyk Antoni (dydejcz@agh.edu.pl)
dr inż. Krupińska Grażyna (grazyna@fis.agh.edu.pl)
dr inż. Gronek Piotr (gronek@agh.edu.pl)

Krótką charakterystyka modułu

Zaznajomienie studentów z zagadnieniami tworzenia oprogramowania na komputerowe systemy równoległe i rozproszone, z pamięcią współdzieloną oraz rozproszoną.

Opis efektów kształcenia dla modułu zajęć

Kod EKM	Student, który zaliczył moduł zajęć wie/umie/potrafi	Powiązania z EKK	Sposób weryfikacji efektów kształcenia (forma zaliczeń)
Wiedza			
M_W004	Student zna podstawowe oraz wybrane zaawansowane idee i pojęcia związane z systemami równoległymi i rozproszonymi, w tym rozproszonymi systemami operacyjnymi	IS2A_W02	Egzamin
M_W005	Student zna podstawowe oraz wybrane zaawansowane narzędzia tworzenia systemów równoległych i rozproszonych	IS2A_W02	Egzamin
M_W006	Student zna zasady i metody przeprowadzania podstawowej analizy poprawności i wydajności systemów równoległych i rozproszonych	IS2A_W02	Egzamin
M_W007	Student rozumie podstawowe problemy projektowe w systemach równoległych i rozproszonych oraz zna wybrane sposoby ich rozwiązywania	IS2A_W02, IS2A_W05	Egzamin

Umiejętności			
M_U003	Student potrafi modyfikować, uruchamiać i śledzić wykonanie programów wykorzystujących podstawowe, aktualne narzędzia do tworzenia systemów równoległych i rozproszonych	IS2A_U03	Kolokwium, Sprawozdanie, Wykonanie ćwiczeń laboratoryjnych
M_U004	Student umie tworzyć proste systemy równoległe i rozproszone w wybranych technikach oraz środowiskach tworzenia oprogramowania	IS2A_U03	Kolokwium, Sprawozdanie, Wykonanie ćwiczeń laboratoryjnych, Wykonanie projektu
Kompetencje społeczne			
M_K003	Student rozumie potrzebę ciągłego dokształcania się oraz ma świadomość odpowiedzialności za pracę własną oraz w zespole	IS2A_K02, IS2A_K01	Wykonanie ćwiczeń laboratoryjnych, Wykonanie projektu

Matryca efektów kształcenia w odniesieniu do form zajęć

Kod EKM	Student, który zaliczył moduł zajęć wie/umie/potrafi	Forma zajęć										
		Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Inne	E-learning
Wiedza												
M_W004	Student zna podstawowe oraz wybrane zaawansowane idee i pojęcia związane z systemami równoległymi i rozproszonymi, w tym rozproszonymi systemami operacyjnymi	+	-	-	-	-	-	-	-	-	-	-
M_W005	Student zna podstawowe oraz wybrane zaawansowane narzędzia tworzenia systemów równoległych i rozproszonych	+	-	-	-	-	-	-	-	-	-	-
M_W006	Student zna zasady i metody przeprowadzania podstawowej analizy poprawności i wydajności systemów równoległych i rozproszonych	+	-	-	-	-	-	-	-	-	-	-
M_W007	Student rozumie podstawowe problemy projektowe w systemach równoległych i rozproszonych oraz zna wybrane sposoby ich rozwiązywania	+	-	-	-	-	-	-	-	-	-	-
Umiejętności												

M_U003	Student potrafi modyfikować, uruchamiać i śledzić wykonanie programów wykorzystujących podstawowe, aktualne narzędzia do tworzenia systemów równoległych i rozproszonych	-	-	+	-	-	-	-	-	-	-	-
M_U004	Student umie tworzyć proste systemy równoległe i rozproszone w wybranych technikach oraz środowiskach tworzenia oprogramowania	-	-	+	+	-	-	-	-	-	-	-
Kompetencje społeczne												
M_K003	Student rozumie potrzebę ciągłego dokształcania się oraz ma świadomość odpowiedzialności za pracę własną oraz w zespole	-	-	+	+	-	-	-	-	-	-	-

Treść modułu zajęć (program wykładów i pozostałych zajęć)

Wykład

Systemy równoległe i rozproszone

A. Projektowanie algorytmów równoległych.

Pojęcia podstawowe. Modele programowania równoległego: zadanie-kanal, przekazywania komunikatów, równoległości danych, współdzielonej pamięci. Metodologia PCAM. Rodzaje dekompozycji. Schematy komunikacyjne. Efekty powierzchni do objętości. Algorytmy równoważenia obciążeń. Algorytmy kolejki zadań.

B. Miary efektywności zrównoleglenia. Klasyfikacja maszyn równoległych Współczynnik przyspieszenia. Prawo Amdahla. Prawo Gustafsona-Barsisa. Metryka Karpa-Flatta. Wydajność. Skalowalność i sprawność. Trendy rynku maszyn równoległych. Klasyfikacja maszyn równoległych. Taksonomia Flynn'a. Architektury pamięci. Topologie maszyn równoległych. Rozwiązania programowe. Systemy hybrydowe.

C. Narzędzia do programowania równoległego na maszynach z pamięcią współdzieloną (SM)

Elementy programowania równoległego. Paradygmaty programowania równoległego. Elementy programowania specyficzne dla maszyn SM, DM, dla klastrów. Narzędzia do programowania równoległego na maszynach z pamięcią współdzieloną (SM).

Tworzenie i likwidacja procesów. Operacje na uchwytach. Tworzenie i likwidacja wątków POSIX. Zmienne lokalne wątku.

D. Mechanizmy komunikacji i synchronizacji między procesami

Wspólna pamięć. Potoki anonimowe. Potoki nazwane. Kolejki komunikatów. Semaforey. Mechanizmy synchronizacji między wątkami jednego procesu. Obiekty typu mutex (zamki).

E. Dyrektywy zrównoleglające kompilatorów dla maszyn SM

Standard OpenMP. Blok zrównoleglany. Deklaracje zmiennych lokalnych i globalnych. Rozdzielenie sterowania pętli iteracyjnych. Bloki równoległe. Lokalna jednowątkowość. Zamek. Bariera. Obsługa zadań. Biblioteka procedur OpenMP.

F. Narzędzia do programowania na maszynach z pamięcią rozproszoną (DM)

Mechanizm gniazdek. Tworzenie gniazdek. Przesyłanie bezpołączeniowe. Komunikacja połączeniowa. Obsługa wielu gniazdek jednocześnie.

G. Message Passing Interface (MPI)

Interfejs MPI. Wywołania funkcji MPI. Operacje na grupach procesów. Operacje na komunikatorach. Przesyłanie komunikatów. Tryby komunikacji MPI. Przykłady komunikacji. Typy pochodne. Mapowanie typów. Funkcje pomocnicze. Ustanowienie i zwolnienie typu. Pakowanie i rozpakowywanie danych. Komunikacja kolektywna. Rodzaje operacji kolektywnych. Redukcja. Operacje zdefiniowane przez użytkownika. Topologie wirtualne. Rozszerzenia MPI. Dynamiczne tworzenie procesów.

H. Narzędzia do programowania układów GPGPU .

Środowiska programowe dla GPGPU. Model programowy CUDA. Zarządzanie pamięcią GPU. Przesyłanie danych CPU - GPU. Kernele - wykonanie kodu na GPU. Kompilacja programu C CUDA. Przegląd narzędzi dla innych języków programowania. Porównanie z modelem działania akceleratora Intel Xeon Phi (MIC). Dyrektywy kompilatora obsługujące akceleratory - OpenMP 4.0.

I. Rozszerzenia języków programowania na maszyny DM (PGAS)

Model Partitioned Global Address Space (PGAS).

Unified Parallel C. Rozproszona pamięć współdzielona. Dystrybucja danych. Podział iteracji między wątki. Wskaźniki. Alokacja pamięci. Operacje kolektywne. Projekt UPC++. Środowisko GPI-2/GASPI.

J. Java Remote Method Invocation. Java Native Interface

Aspekty obiektowego programowania rozproszonego. Struktura oprogramowania rozproszonego. Model klient - serwer.

Schemat komunikacji Java RMI. Ograniczenia RMI. Interfejs RMI. Menedżer obiektów. Własności interfejsu Remote. Mechanizm serializacji. Java Native Interface. Schemat tworzenia aplikacji RMI/JNI. Generacja szkieletu RMI i nagłówek JNI.

K. CORBA - Common Object Request Broker Architecture

Model referencyjny CORBA. Interfejs CORBA. Język IDL. Typy IDL. Definicje operacji. Operacje wykonywane przez referencje. Portable Object Adapter. Cykl realizacji żądania w POA. Implementacje CORBA. Przegląd wybranych serwisów CORBA. Schemat tworzenia aplikacji CORBA.

Ćwiczenia laboratoryjne

PRACOWNIA KOMPUTEROWA

Wprowadzenie do środowiska programowania równoległego MPI.

Efekty kształcenia:

- student potrafi zakodować, skompilować i uruchomić prosty program MPI w języku C,
- student potrafi zrealizować przesyłanie i odczytywanie statusu komunikatów MPI,
- student potrafi zmierzyć czas wykonania programu MPI.

Podstawy komunikacji kolektywnej w MPI.

Efekty kształcenia:

- student potrafi zrealizować prostą komunikację kolektywną MPI,
- student potrafi wykonać rozgłaszanie i redukcję danych,
- student potrafi tworzyć komunikatory pochodne.

Interfejs wizualizacyjny MPE dla MPI.

Efekty kształcenia:

- student potrafi wykonać podstawowe operacje równoległego interfejsu graficznego MPE,
- student potrafi rejestrować zdarzenia w aplikacji MPI za pomocą funkcji biblioteki MPE,
- student potrafi wykonać wizualizację logu MPE aplikacji MPI za pomocą

odpowiedniego oprogramowania.

Profilowanie aplikacji równoległej MPI.

Efekty kształcenia:

– student potrafi zrealizować profilowanie operacji MPI za pomocą usług biblioteki MPE.

Działanie aplikacji w środowisku rozproszonym Java RMI / JNI.

Efekty kształcenia:

– student potrafi skonstruować prosty interfejs RMI,

– student potrafi wygenerować szkielet serwera RMI,

– student potrafi uruchomić katalog serwerów w środowisku JDK,

– student potrafi zaprogramować prosty interfejs JNI,

– student potrafi zakodować, skompilować i uruchomić prostą aplikację klient – serwer korzystającą z usług RMI/JNI w systemie Linux.

Działanie aplikacji w środowisku rozproszonym CORBA.

Efekty kształcenia:

– student potrafi skonstruować prosty interfejs w języku CORBA IDL,

– student potrafi wygenerować szkielet serwera CORBA w języku C++,

– student potrafi uruchomić prosty serwer usługi CORBA,

– student potrafi zaprogramować prostego klienta CORBA w języku C++,

– student potrafi zakodować, skompilować i uruchomić prostą aplikację klient – serwer korzystającą z implementacji środowiska CORBA w systemie Linux.

Przegląd działania narzędzi PGAS.

Efekty kształcenia:

– student potrafi opracować, skompilować i uruchomić prostą aplikację UPC za pomocą narzędzi pakietu Berkeley UPC,

– student potrafi opracować, skompilować i uruchomić prostą aplikację PGAS za pomocą narzędzi pakietu GPI-2/GASPI,

Przegląd działania narzędzi CUDA.

Efekty kształcenia:

– student potrafi uruchomić prostą aplikację wykorzystującą GPU za pomocą narzędzi pakietu CUDA,

Ćwiczenia projektowe

PROJEKT

W ramach ćwiczeń projektowych studenci wykonują aplikację projektową w środowisku równoległym MPI i wybranym środowisku programowania rozproszonego. Projekty programistyczne oceniane będą zgodnie z kryteriami zamieszczonymi na stronie internetowej przedmiotu oraz przedstawionymi na pierwszych zajęciach. Projekty są wykonywane w zespołach.

Efekty kształcenia:

– student potrafi zaprojektować implementację równoległą problemu obliczeniowego w środowisku obliczeń równoległych MPI, w oparciu o opis problemu algorytmicznego zawierający wskazówki dotyczące strategii zrównoleglania,

– student potrafi uruchomić aplikację równoległą działającą w środowisku MPI,

– student potrafi zaprojektować implementację równoległą problemu obliczeniowego w środowisku usług przetwarzania rozproszonego Java RMI lub CORBA, lub CUDA, w oparciu o opis problemu algorytmicznego zawierający wskazówki dotyczące strategii zrównoleglania,

– student potrafi uruchomić aplikację rozproszoną działającą w środowisku Java RMI lub CORBA, lub CUDA.

Sposób obliczania oceny końcowej

W ramach zajęć laboratoryjnych studenci wykonują aplikację projektową w środowisku równoległym MPI i wybranym środowisku programowania rozproszonego. Projekty programistyczne oceniane będą zgodnie z kryteriami zamieszczonymi na stronie internetowej przedmiotu oraz przedstawionymi na pierwszych zajęciach. Ocena zaliczeniowa jest obliczana jako średnia z ocen cząstkowych wykonanych projektów.

Egzamin zakłada udzielenie odpowiedzi na zadaną liczbę pytań testowych, wybranych przez egzaminatora z zawczasu uzgodnionej i udostępnionej na stronie internetowej przedmiotu, listy.

Ocena końcowa z modułu obliczana jest jako średnia ważona z oceny zaliczeniowej zajęć projektowych (z wagą 40%) i oceny z egzaminu (z wagą 60%).

Wymagania wstępne i dodatkowe

- Programowanie w językach C/C++ i Java

Zalecana literatura i pomoce naukowe

- Andrzej Karbowski, Ewa Niewiadomska-Szynkiewicz [red.], „Obliczenia Równoległe i Rozproszone”, Politechnika Warszawska 2001
- M. Herlihy, N. Shavit „The Art of Multiprocessor Programming” Elsevier, 2008 (wydanie polskie „Sztuka programowania wieloprocesorowego”, PWN 2010)
- Michael J. Quinn, “Parallel Programming in C with MPI and OpenMP”, McGraw-Hill 2004
- William Gropp, Ewing Lusk, Anthony Skjellum, “Using MPI: Portable Parallel Programming with the Message-Passing Interface”, MIT Press 1999
- Ian Foster, “Designing and Building Parallel Programs”, Addison-Wesley 1995
[<http://www-unix.mcs.anl.gov/dbpp/>]
- Marc Snir, Steve Otto, Steven Huss-Lederman, David Walker, Jack Dongarra, “MPI: The Complete Reference”, MIT Press 1998
- Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, “Introduction to Parallel Computing”, Addison-Wesley / Pearson Education 2003
- G. Coulouris et al., “Distributed Systems. Concepts and Design” (4th ed.), Addison Wesley, 2005 (wydanie polskie “Systemy rozproszone. Podstawy i projektowanie”, WNT 1998)
- S. Tanenbaum, “Distributed Systems. Principles and Paradigms” (2nd ed.), Prentice Hall 2002
- George Em Karniadakis, Robert M. Krby II, “Parallel Scientific Computing in C++ and MPI”, Cambridge University Press 2003, 2008
- Geoffrey C. Fox, Roy D. Williams, Paul C. Messina, “Parallel Computing Works!”, Morgan Kaufmann Publishers 1994 [<http://www.netlib.org/utk/lisi/pcwLSI/text/>]
- Victor Eijkhout, et al., “Introduction to High Performance Scientific Computing”, lulu.com 2011 [<http://www.tacc.utexas.edu/~eijkhout/istc/istc.html>]

Publikacje naukowe osób prowadzących zajęcia związane z tematyką modułu

- Naming boys after U.S. presidents in 20th century / K. KUŁAKOWSKI, P. KULCZYCKI, K. MISZTAŁ, A. DYDEJCZYK, P. GRONEK, M.J. KRAWCZYK // Acta Physica Polonica. A ; ISSN 0587-4246. — 2016 vol. 129 no. 5, s. 1038-1044
- No more presidents in my family / Krzysztof KUŁAKOWSKI, Piotr KULCZYCKI, Krzysztof Miształ, Antoni DYDEJCZYK, Piotr GRONEK, Małgorzata KRAWCZYK // W: FENS : 8th Polish symposium on Econo- and sociophysics FENS : 4-6 November 2015, Rzeszów, Poland : book of abstracts. — ISBN10: 83-89585-42-1. — S. 25
- If others jump to the queue front, how long I will wait? / M.J. KRAWCZYK, P. GRONEK, M. NAWOJCZYK, K. KUŁAKOWSKI // Acta Physica Polonica. A ; ISSN 0587-4246. — 2015 vol. 127 no. 3-A, s. A-95-A-98 — 7th Polish Symposium of Physics in Economy and Social Sciences (FENS) : Lublin, Poland, 14-17 May 2014
- How many parameters to model states of mind? / Krzysztof KUŁAKOWSKI, Piotr GRONEK, Antoni DYDEJCZYK // W: 27th European Conference on Modelling and Simulation ECMS 2013 : May 27th-May 30th, 2013, Ålesund, Norway : proceedings / eds. Webjørn Rekdalsbakken, Robin T. Bye, Houxiang Zhang. — [Norway : s.n.], 2013 + CD. — ISBN: 978-0-9564944-6-7 ; e-ISBN: 978-0-9564944-7-4. — S. 895-899.
- Zaller-Deffuant model of mass opinion / Krzysztof MALARZ, Piotr GRONEK, Krzysztof KUŁAKOWSKI // JASSS Journal of Artificial Societies and Social Simulation ; ISSN 1460-7425. — 2011 vol. 14 iss. 1 art. no. 2, s. [1-20]

Informacje dodatkowe

- Dopuszcza się nieobecność na co najwyżej 2 zajęciach laboratoryjnych (w tym co najwyżej 1 nie usprawiedliwiona).
- W/w nieobecność nie ma wpływu na uzyskanie oceny cząstkowej z dotyczącej formy zajęć.

Nakład pracy studenta (bilans punktów ECTS)

Forma aktywności studenta	Obciążenie studenta
Udział w wykładach	30 godz
Udział w ćwiczeniach laboratoryjnych	26 godz
Przygotowanie do zajęć	20 godz
Wykonanie projektu	30 godz
Udział w ćwiczeniach projektowych	4 godz
Samodzielne studiowanie tematyki zajęć	15 godz
Egzamin lub kolokwium zaliczeniowe	2 godz
Sumaryczne obciążenie pracą studenta	127 godz
Punkty ECTS za moduł	5 ECTS