



Karta przedmiotu

Nazwa i kod przedmiotu	Systemy obliczeniowe wysokiej wydajności, PG_00054279						
Kierunek studiów	Informatyka						
Data rozpoczęcia studiów	luty 2027 r.	Rok akademicki realizacji przedmiotu			2026/2027		
Poziom kształcenia	II stopnia	Grupa zajęć			Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			polski		
Semestr studiów	1	Liczba punktów ECTS			3.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca	Wydziały Politechniki Gdańskiej -> Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Architektury Systemów Komputerowych						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr hab. inż. Paweł Czarnul				
	Prowadzący zajęcia z przedmiotu		dr inż. Adam Brzeski dr hab. inż. Paweł Czarnul mgr inż. Robert Kałaska dr inż. Mariusz Matuszek				
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	15.0	0.0	0.0	30
W tym liczba godzin zajęć na odległość: 0.0							
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów		Udział w konsultacjach		Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	30		10.0		35.0	75
Cel przedmiotu	poznacenie sposobów i technologii zrównoleglenia i optymalizacji algorytmów w wydajnych środowiskach równoległych						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K7_U12] potrafi w pogłębionym stopniu analizować działanie elementów, układów i systemów związanych z kierunkiem studiów oraz mierzyć ich parametry i badać charakterystyki techniczne, a także planować i przeprowadzać eksperymenty związane z kierunkiem studiów, w tym symulacje komputerowe, interpretować uzyskane wyniki i wyciągać wnioski	potrafi skonfigurować środowisko równoległe, uruchamiać i profilować wykonanie aplikacji równoległych	[SU1] Ocena realizacji zadania
	[K7_U03] potrafi zaprojektować, zgodnie z zadaną specyfikacją, oraz wykonać typowe dla kierunku studiów złożone urządzenie, obiekt, system lub zrealizować proces, używając odpowiednio dobranych metod, technik, narzędzi i materiałów, korzystając ze standardów i norm inżynierskich, stosując właściwe dla kierunków studiów technologie i wykorzystując doświadczenie zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską	potrafi zaprojektować i zaimplementować skalowalną aplikację równoległą	[SU1] Ocena realizacji zadania
	[K7_U04] potrafi wykorzystywać posiadaną wiedzę z zakresu metod i technik programowania oraz dobrać i zastosować właściwe metody i narzędzia programistyczne w tworzeniu oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, charakterystycznych dla danego kierunku studiów, dokonując oceny i krytycznej analizy wykonanego oprogramowania, a także syntezy i twórczej interpretacji prezentowanych za jego pomocą informacji	potrafi optymalizować implementację równoległą	[SU1] Ocena realizacji zadania
[K7_W08] zna i rozumie w pogłębionym stopniu fundamentalne dylematy współczesnej cywilizacji, główne trendy rozwojowe dyscyplin naukowych istotnych dla kierunku kształcenia	potrafi znajdować zastosowania dla programowania równoległego	[SW1] Ocena wiedzy faktograficznej	
Treści przedmiotu	Treści przedmiotu - wykład 1 Podanie zasad zaliczenia 2 Wprowadzenie do przetwarzania w środowiskach równoległych: praktyczny model aplikacji równoległej: podstawowe parametry aplikacji opartej o procesy i wątki. Praktyczny model systemu równoległego i rozproszonego, grafy połączeń. Kryteria odwzorowań. Opóźnienie komunikacyjne: latency i bandwidth. Przykłady wartości w rzeczywistych systemach. 3 Architektury równoległe: Shared Memory, Distributed Shared Memory, Distributed Memory. 4 Paradygmat master-slave, paradygmat SPMD, przetwarzanie potokowe, dziel-i-zwyciężaj, przykłady. 5 MPI: model aplikacji. Podstawowe API, prosta aplikacja. 6 MPI: uruchamianie, różne implementacje: MPICH, OpenMPI, uruchamianie na klastrach, superkomputerach, systemy kolejkowe 7 MPI: Tryby operacji send: klasyczny, rsend, bsend, ssend, przykłady. Komunikacja nieblokująca. Komunikacja blokująca. Komunikacja grupowa a punkt-punkt. tworzenie typów i pakowanie 8 MPI: Komunikatory, dynamiczne tworzenie procesów 9 MPI: dynamiczne równoważenie obciążenia: repartycjonowanie, „ghost nodes”. 10 Przykłady aplikacji w MPI – wydajność na klastrach TASK (różne tryby komunikacji (SSend, lsend, Rsend, Send), nakładanie obliczeń i komunikacji w MPI) 11 MPI: dynamiczne uruchamianie procesów, komunikacja jednokierunkowa 12 Checkpointing aplikacji równoległych 13 Wątki a MPI, OpenMPI 14 Zaawansowane techniki zrównoleglenia algorytmów: nakładanie komunikacji i obliczeń, inne techniki ukrywania kosztów komunikacji. 15 OpenMP 16 Równoległe operacje wejścia/wyjścia (Parallel I/O w MPI) 17 Przykłady aplikacji HPC: aplikacje medyczne etc. 18 Podstawy programowania z wykorzystaniem CUDA 19 Podstawy programowania z wykorzystaniem OpenMP dla Xeon Phi		
Wymagania wstępne i dodatkowe	znajomość języka C, znajomość podstawowych algorytmów sekwencyjnych i struktur danych		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa ocena końcowej
	Kolokwia w czasie semestru	50.0%	50.0%
	Ćwiczenia praktyczne	50.0%	50.0%

Zalecana lista lektur	Podstawowa lista lektur	1 Dokumentacja MPI 2. Dokumentacja OpenMP 3. Dokumentacja CUDA 4. P. Czarnul. Parallel Programming for Modern High Performance Computing Systems. Taylor & Francis. 2018 ISBN 9781138305953
	Uzupełniająca lista lektur	Nie ma wymagan
	Adresy eZasobów	
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania		
Zajęcia praktyczne w ramach przedmiotu	Nie dotyczy	

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.