

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Zaawansowane zastosowania kart graficznych		Kod 1010515321010519522
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 1 / 2
Ścieżka obieralności/specjalność Sieci komputerowe	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obieralny
Stopień studiów: II stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 16 Ćwiczenia: - Laboratoria: 16 Projekty/seminaria: -		Liczba punktów 4
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) kierunkowy		(ogólnouczelniany, z innego kierunku) z danego kierunku
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne nauki techniczne		Podział ECTS (liczba i %) 4 100% 4 100%
Odpowiedzialny za przedmiot / wykładowca: dr inż. Witold Andrzejewski email: Witold.Andrzejewski@cs.put.poznan.pl tel. 61 6652965 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę o architekturze komputerów, o programowaniu obiektowym i proceduralnym oraz o metodach oceny złożoności algorytmów. Dodatkowo, powinien posiadać podstawową wiedzę dotyczącą geometrii obliczeniowej.
2	Umiejętności:	Powinien posiadać umiejętność rozwiązywania podstawowych problemów algorytmicznych, programowania w języku C/C++, oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.
3	Kompetencje społeczne	Powinien również rozumieć konieczność poszerzania swoich kompetencji i mieć gotowość do podjęcia współpracy w ramach zespołu. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
Cel przedmiotu:		
1. Przekazanie studentom podstawowej wiedzy z programowania procesorów kart graficznych (GPU), w zakresie: <ul style="list-style-type: none"> a. Wybranych aspektów architektury sprzętowej popularnych kart graficznych. b. Logicznego modelu współbieżności kart graficznych w kontekście API CUDA i OpenCL. c. Optymalizacji wykonania programów na kartach graficznych. d. Wybranych podstawowych algorytmów równoległych, w tym: map, reduce, compact, sort, scan, search, generowania ntych wyrazów ciągów. e. Wybranych struktur danych, w tym: macierzy, tablic hashowych i drzew CSS i algorytmów ich przetwarzania. f. Oceny złożoności algorytmów równoległych (w tym model PRAM) i ich związku z faktyczną złożonością programów wykorzystujących karty graficzne. g. Zastosowania kart graficznych do rozwiązywania praktycznych problemów związanych z przetwarzaniem i wizualizacją danych. 2. Rozwijanie u studentów umiejętności: <ul style="list-style-type: none"> a. Rozwiązywania problemów algorytmicznych z ukierunkowaniem na zrównoleglenie operacji przetwarzania danych. b. Optymalizacji programów wykorzystujących procesory kart graficznych. 		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		

<p>1. ma zaawansowaną i pogłębioną wiedzę z zakresu narzędzi i środowisk programistycznych użytecznych przy tworzeniu programów na procesory kart graficznych - [K2st_W2]</p> <p>2. ma zaawansowaną wiedzę szczegółową dotyczącą tworzenia algorytmów równoległych na procesory kart graficznych - [K2st_W3]</p> <p>3. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach informatyki w dziedzinie grafiki komputerowej i akceleracji obliczeń na GPU - [K2st_W4]</p> <p>4. zna zaawansowane metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich związanych z projektowaniem i optymalizacją niskopoziomową programów wykorzystujących procesory kart graficznych - [K2st_W6]</p>
<p>Umiejętności:</p> <p>1. potrafi posługiwać się technikami informacyjno-komunikacyjnymi wykorzystywanymi przy realizacji przedsięwzięć informatycznych potrafi posługiwać się technikami informacyjno-komunikacyjnymi wykorzystywanymi przy realizacji przedsięwzięć informatycznych - [K2st_U2]</p> <p>2. potrafi - przy formułowaniu i rozwiązywaniu zadań inżynierskich - integrować wiedzę z różnych obszarów informatyki oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne - [K2st_U5]</p> <p>3. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K2st_U6]</p> <p>4. potrafi dokonać krytycznej analizy istniejących algorytmów oraz zaproponować ich wersje równoległe potrafi dokonać krytycznej analizy istniejących algorytmów oraz zaproponować ich wersje równoległe - [K2st_U8]</p> <p>5. potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego, polegającego na budowie lub ocenie systemu informatycznego, lub jego składowych, wykorzystujących akcelerację GPU, w tym dostrzec ograniczenia tych metod i narzędzi - [K2st_U9]</p> <p>6. potrafi ? zgodnie z zadaną specyfikacją, uwzględniając aspekty pozatechniczne ? zaprojektować i zaimplementować algorytm wykorzystujący akcelerację GPU potrafi - zgodnie z zadaną specyfikacją, uwzględniając aspekty pozatechniczne - zaprojektować i zaimplementować algorytm wykorzystujący akcelerację GPU - [K2st_U11]</p>
<p>Kompetencje społeczne:</p> <p>1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K2st_K1]</p> <p>2. rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu informatyki w rozwiązywaniu problemów badawczych i praktycznych - [K2st_K2]</p>

<p>Sposoby sprawdzenia efektów kształcenia</p>
<p>Ocena formująca</p> <p>a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach</p> <p>b) w zakresie laboratoriów / ćwiczeń weryfikowanie założonych efektów kształcenia realizowane jest przez: okresową ocenę przygotowania studenta do poszczególnych sesji zajęć laboratoryjnych (sprawdzian wejściowy) oraz ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych</p> <p>Ocena podsumowująca:</p> <p>a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez ocenę wiedzy i umiejętności wykazanych na zaliczeniu pisemnym o formie testu wielokrotnego wyboru składającego się z ok. 30 pytań, łączna liczba punktów za prawidłowe odpowiedzi: 30, minimalna liczba punktów umożliwiających zaliczenie: 16</p> <p>b) w zakresie laboratoriów / ćwiczeń weryfikowanie założonych efektów kształcenia realizowane jest przez ocenę i obronę przez studenta sprawozdania z realizacji projektu.</p> <p>Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:</p> <ul style="list-style-type: none"> - efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu, - uwagi związane z udoskonaleniem materiałów dydaktycznych, - wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenie procesu dydaktycznego.
<p>Treści programowe</p>
<p>Program wykładu obejmuje następujące zagadnienia:</p> <ol style="list-style-type: none"> 1. Motywacja stojąca za wykorzystaniem kart graficznych do obliczeń. Dodatkowo: przedstawienie różnych rozwiązań technologicznych umożliwiających przetwarzanie równoległe, wprowadzenie podstawowych pojęć stosowanych w kolejnych wykładach. Omówienie podstaw modelu programistycznego wykorzystywanego w programach wykorzystujących procesory kart graficznych. 2. Architektura sprzętowa kart graficznych. Omówienie związków pomiędzy architekturą sprzętową a modelem programistycznym. Przedstawienie trików bitowych. 3. Omówienie hierarchii pamięci i metod wydajnego dostępu dla różnych architektur sprzętowych. Przedstawienie przykładowych metod optymalizacji wykorzystujących różne poziomy pamięci operacyjnej. Omówienie rozwiązań pozwalających na zapewnienie równoległego przesyłu danych pomiędzy komputerem a kartą graficzną i wykonywania obliczeń. 4. Omówienie mechanizmów komunikacji i synchronizacji wątków 5. Wprowadzenie podstaw teoretycznych oceny złożoności algorytmów równoległych. Omówienie obliczeń w modelu maszyny PRAM. Podstawowe algorytmy wykorzystywane podczas konstrukcji bardziej złożonych rozwiązań, w tym: map,

- gather, scatter, reduce, compact, search, scan. Szczegółowe omówienie algorytmu scan. Ocena złożoności wprowadzonych algorytmów w ich wersji sekwencyjnej i równoległej.
- Szczegółowe omówienie algorytmów : compact i reduce oraz algorytmów przeszukiwania i sortowania danych. Ocena złożoności przedstawionych algorytmów.
 - Algorytmy generowania elementów w sekwencji kombinacji, permutacji bądź w wyniku iloczynu kartezyjskiego wraz z ich przykładowymi zastosowaniami.
 - Algorytmy połączeń pionowych i poziomych w oparciu o algorytm merge-path
 - Problemy związane z wykorzystywaniem standardowych struktur danych. Omówienie wydajnych metod przetwarzania list (drzewo CSS), macierzy (w tym macierzy rzadkich), tablic haszowych oraz grafów. Oszacowanie złożoności algorytmów przetwarzania tych struktur.
 - Praktyczne zastosowania GPU do przetwarzania, eksploracji i wizualizacji danych.

Program laboratorium obejmuje następujące zagadnienia:

- Zapoznanie się z CUDA API.
- Ćwiczenia związane z prawidłową konstrukcją siatki obliczeń.
- Implementacja prostych algorytmów równoległych.
- Omówienie sposobów debugowania programów wykorzystujących karty graficzne.
- Realizacja ćwiczeń w których należy uwzględnić komunikację i synchronizację wątków.
- Testowanie wydajności dostępu do różnych dostępnych rodzajów pamięci operacyjnej na karcie graficznej.
- Zapoznanie się z biblioteką thrust. Omówienie podstawowych koncepcji stojących za API tej biblioteki. Przedstawienie podstawowych algorytmów w niej zaimplementowanych.
- Omówienie biblioteki CURAND. Ćwiczenia wykorzystujące bibliotekę thrust. Budowa złożonych algorytmów z podstawowych algorytmów takich jak: map, reduce, gather, scatter, search, scan.
- Omówienie API OpenCL oraz jego podobieństw i różnic w stosunku do NVIDIA CUDA
- Omówienie NVIDIA Optix API i jego zastosowań przy wizualizacji danych.

Część wyżej wymienionych zagadnień jest realizowanych przez studenta samodzielnie.

Metody dydaktyczne:

- wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, rozwiązywanie zadań.
- ćwiczenia laboratoryjne: rozwiązywanie zadań, ćwiczenia praktyczne, pokaz multimedialny, demonstracja.

Literatura podstawowa:

- Cuda w przykładach : wprowadzenie do ogólnego programowania procesorów GPU / Jason Sanders, Edward Kandrot
- OpenCL : akceleracja GPU w praktyce / Marek Sawerwain.
- NVIDIA: CUDA C Programming Guide: <http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>
- NVIDIA: CUDA C Best Practices Guide: <http://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html>
- NVIDIA: OpenCL Jumpstart Guide: http://www.cs.cmu.edu/afs/cs/academic/class/15668-s11/www/cuda-doc/OpenCL_Jumpstart_Guide.pdf
- NVIDIA: OpenCL Best Practices Guide: http://www.nvidia.com/content/cudazone/CUDABrowser/downloads/papers/NVIDIA_OpenCL_BestPracticesGuide.pdf
- NVIDIA Optix API documentation <http://raytracing-docs.nvidia.com/optix/index.html>

Literatura uzupełniająca:

- Jianlong Zhong* and Bingsheng He. Medusa: Simplified Graph Processing on GPUs. Accepted by TPDS 2013: IEEE Transactions on Parallel and Distributed System
- Bingsheng He and Jeffrey Xu Yu. High-Throughput Transaction Executions on Graphics Processors. Proceedings of Very Large Data Bases (VLDB) 2011
- Andrzejewski, Witold; Boinski, Pawel Efficient spatial co-location pattern mining on multiple GPUs Journal Article Expert Systems with Applications, 93 (Supplement C), pp. 465-483, 2018, ISBN: 0957-4174.
- Andrzejewski, Witold; Boinski, Pawel Parallel GPU-based Plane-sweep Algorithm for Construction of iCPI-trees Journal Article Journal of Database Management, 26 (3), pp. 1-20, 2015, ISSN: 1063-8016.
- Andrzejewski, Witold; Boinski, Pawel Parallel approach to incremental co-location pattern mining, Information Sciences, accepted for publication

Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
----------	--------------

1. udział w zajęciach laboratoryjnych	16	
2. przygotowanie do ćwiczeń laboratoryjnych	16	
3. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu	2 16	
4. przygotowanie do sprawdzianów / kolokwium	16	
5. udział w wykładach	18	
6. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi	16	
7. przygotowanie do zaliczenia wykładów		
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	100	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	36	2
Zajęcia o charakterze praktycznym	32	2