

A TANTÁRGY ADATLAPJA

1. A képzési program adatai

1.1 Felsőoktatási intézmény	Babeş–Bolyai Tudományegyetem
1.2 Kar	Matematika és Informatika Kar
1.3 Intézet	Magyar Matematika és Informatika Intézet
1.4 Szakterület	Informatika
1.5 Képzési szint	Mesteri
1.6 Szak / Képesítés	Adatelemzés és modellezés / Analiza datelor și simulare

2. A tantárgy adatai

2.1 A tantárgy neve (ang)	Párhuzamos programozás, GPGPU programozás						
2.2 Az előadásért felelős tanár neve	Horváth Zoltán						
2.3 A szemináriumért felelős tanár neve	Horváth Zoltán						
2.4 Tanulmányi év	1	2.5 Félév	2	2.6. Értékelés módja	vizsga	2.7 Tantárgy típusa	választható – alap

3. Teljes becsült idő (az oktatási tevékenység féléves óraszámja)

3.1 Heti óraszám	3	melyből: 3.2 előadás	2	3.3 szeminárium/labor/praktika	1
3.4 Tantervben szereplő össz-óraszám	42	melyből: 3.5 előadás	28	3.6 szeminárium/labor	14
A tanulmányi idő elosztása:					Óra
A tankönyv, a jegyzet, a szakirodalom vagy saját jegyzetek tanulmányozása					42
Könyvtárban, elektronikus adatbázisokban vagy terepen való további tájékozódás					36
Szemináriumok / laborok, házi feladatok, portfóliók, referátumok, esszék kidolgozása					50
Egyéni készségfejlesztés (tutorálás)					28
Vizsgák					3
Más tevékenységek:					
3.7 Egyéni munka össz-óraszámja					158
3.8 A félév össz-óraszámja					200
3.9 Kreditszám					8

4. Előfeltételek (ha vannak)

4.1 Tantervi	Alapképzésből a C és az Assembly programozás
4.2 Kompetenciabeli	C programozás, Assembly programozás

5. Feltételek (ha vannak)

5.1 Az előadás lebonyolításának feltételei	.Az előadóterem kell rendelkezzen táblával illetve projektorral
5.2 A szeminárium / labor lebonyolításának feltételei	A laborokhoz javasolt, hogy a diákok olyan laptopokkal rendelkezzenek amelyeken Nvidia CUDA programok futtathatók, illetve szükséges hozzáférés nagyobb szerverekhez Nvidia grafikus kártyákkal (TESLA)

6. Elsajátítandó jellemző kompetenciák

Szakmai kompetenciák	<ul style="list-style-type: none"> • Párhuzamos programozás paradigmájának elsajátítása • Különböző algoritmusok optimális megírása párhuzamos architektúrákra • Optimalizációs problémák megértése grafikus kártyák programozása esetén
Transzverzális kompetenciák	<ul style="list-style-type: none"> • Párhuzamos programozás gondolkozásmódjának a megértése

7. A tantárgy célkitűzései (az elsajátítandó jellemző kompetenciák alapján)

7.1 A tantárgy általános célkitűzése	A tantárgy célja párhuzamos programozáshoz szükséges gondolkodásmód megtanítása, és ennek a gyakorlatba ültetése grafikus processzorokon
7.2 A tantárgy sajátos célkitűzései	<ul style="list-style-type: none"> • A párhuzamos programozás alapjainak a megértése • Grafikus kártyák programozása CUDA nyelvben (és egy bevezető más nyelvekhez is, pl. az OpenCL -hez) • A párhuzamos programozáshoz kötődő specifikus problémák megértése és kezelése, párhuzamosítás, hatékony memóriakezelés, szálak, ripple hatások, szinkronizálás, stb.

8. A tantárgy tartalma

8.1 Előadás	Didaktikai módszerek	Megjegyzések
1. hét Bevezetés a párhuzamos programozásba Alapfogalmak, utasítás-, adat-, és szál szintű párhuzamosság, párhuzamos programmodellek, SPMD programozási modell	tanári magyarázat, rávezetés, példa bemutatása	
2. hét Párhuzamos processzor architektúrák Párhuzamos processzorok története, felépítése és működése: CPU, GPU és MIC	tanári magyarázat, rávezetés, példa bemutatása	
3. hét Párhuzamos programozás CPU-n OpenMP, SSE/AVX intrinsic függvények, ISPC	tanári magyarázat, rávezetés, példa bemutatása	
4. hét CUDA programozási környezet CUDA program felépítése, példaprogram, GPU memória, kernel függvények és szálak	tanári magyarázat, rávezetés, példa bemutatása	
5. hét CUDA szálak Szálak, warp-ok és blokkok, ütemezés, szinkronizáció, skálázhatóság	tanári magyarázat, rávezetés, példa bemutatása	
6. hét Memóriakezelés CUDA-ban Memóriatípusok, helyfoglalás, adatok másolása, memória-hozzáférés hatékonysága	tanári magyarázat, rávezetés, példa bemutatása	

7. hét CUDA programok optimalizálása I. Teljesítmény mérése, memória-hozzáférések optimalizálása	tanári magyarázat, rávezetés, példa bemutatása	
8. hét CUDA programok optimalizálása II. Kihasznátság, szálak divergenciája, számítási hatékonyság növelése	tanári magyarázat, rávezetés, példa bemutatása	
9. hét Lebegőpontos műveletek CUDA-ban Lebegőpontos adattípusok, műveletek, teljesítmény és pontosság, beépített matematikai függvények	tanári magyarázat, rávezetés, példa bemutatása	
10. hét CUDA debuggolás és teljesítményelemzés Eszközök (Nsight, Visual Profiler), módszerek, példák	tanári magyarázat, rávezetés, példa bemutatása	
11. hét Párhuzamos programozási minták Scan, reduce, split, compact, stb.	tanári magyarázat, rávezetés, példa bemutatása	
12. hét Esettanulmány: sugárkövetés CUDA-val Bevezetés a sugárkövetésbe, egyszerű példa, CUDA implementáció	tanári magyarázat, rávezetés, példa bemutatása	
13. hét OpenCL programozási környezet Bevezetés, alapfogalmak, összehasonlítás CUDA-val, példaprogram	tanári magyarázat, rávezetés, példa bemutatása	
14. hét Aktuális kutatási témák Párhuzamos processzorok jövője, programozási modellek és nyelvek fejlődése	tanári magyarázat, rávezetés, példa bemutatása	

Könyvészet

1. Programming Massively Parallel Processors: A Hands-On Approach

David B. Kirk, Wen-Mei W Hwu (2010)

2. CUDA C Programming Guide

NVIDIA (2013)

3. Computer Architecture: A Quantitative Approach, 5th Edition

John L. Hennessy, David A. Patterson (2012)

4. CUDA by Example: An Introduction to General Purpose GPU Programming

Jason Sanders and Edward Kandrot (2011)

5. CUDA Application Design and Development

Rob Farber (2011)

8.2 Labor / Gyakorlat	Didaktikai módszerek	Megjegyzések
Labor:		
1. hét Mandelbrot fraktál kirajzoló program szekvenciális implementálása C/C++-ban	munkáltatás, demonstráció, példák	
2. hét Mandelbrot fraktál párhuzamos implementálása OpenMP-vel és intrinsic függvényekkel	munkáltatás, demonstráció, példák	
3. hét	munkáltatás,	

Ismerkedés a CUDA fordítóval, egyszerű CUDA programok megírása („Hello world”, vektorösszeg)	demonstráció, példák	
4. hét Mandelbrot fraktál implementálása CUDA-ban	munkáltatás, demonstráció, példák	
5. hét Mátrixszorzás hatékony implementálása CUDA-ban	munkáltatás, demonstráció, példák	
6. hét Egyszerű sugárkövető implementálása CUDA-ban	munkáltatás, demonstráció, példák	
7. hét Ismétlés, feladatok ellenőrzése	munkáltatás, demonstráció	

9. Az epiztemikus közösségek képviselői, a szakmai egyesületek és a szakterület reprezentatív munkáltatói elvárásainak összehangba hozása a tantárgy tartalmával.

- Az előadás anyaga a neten szabadon elérhető, Nvidia által közzétett kurzusok (<https://developer.nvidia.com/cuda-training>) anyagának megfelelő tudásanyagot mutat be, az itt található kurzusok közül a University Of Illinois : ECE 408 és a Stanford University: CS193G kurzusokkal összehangba hozható, vagyis a szakterület tekintélyei által felállított logikát követi.

10. Értékelés

Tevékenység típusa	10.1 Értékelési kritériumok	10.2 Értékelési módszerek	10.3 Aránya a végső jegyben
10.4 Előadás	Írásbeli vizsga a félév végén	Írásbeli vizsga	40%
10.5 Labor	Programozási feladatok bemutatása	A megoldások pontozása	60%

10.6 A teljesítmény minimumkövetelményei

Az elvárt minimális tudás:

- A párhuzamos programozáshoz szükséges gondolkodás megértése és elsajátítása
- Egyszerű CUDA programok írásának a készsége

Az átmenő jegy feltételei:

- A vizsgán legkevesebb a pontok felének összegyűjtése (5.0, kizáró jellegű)
- Az évközi jegyben legkevesebb a pontok felének összegyűjtése (5.0, kizáró jellegű)

Kitöltés dátuma

2018.04.24

Előadás felelőse

dr. Horváth Zoltán, professzor

Labor / praktika felelőse

dr. Horváth Zoltán, professzor

Az intézeti jóváhagyás dátuma

.....

Intézetigazgató

dr. András Szilárd, docens