SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	**Babes-Bolyai University**
1.2 Faculty	**Mathematics and Informatics**
1.3 Department	**Informatics**
1.4 Field of study	**Informatics**
1.5 Study cycle	**Master**
1.6 Study programme / Qualification	**High Performance Computing and Big Data Analytics**

2. Information regarding the discipline

2.1 Name of the discipline (en) (ro)			**GPU and distributed architecture computing** **Programare pe arhitecturi GPU si distribuite**				
2.2 Course coordinator			**Conf. dr. Rareș Boian**				
2.3 Seminar coordinator			**Conf. dr. Rareș Boian**				
2.4. Year of study	**2**	2.5 Semester	**3**	2.6. Type of evaluation	**E**	2.7 Type of discipline	**Mandatory**
2.8 Code of the discipline		**MME8111**					

3. Total estimated time (hours/semester of didactic activities)

3.1 Hours per week	**3**	Of which: 3.2 course	**2**	3.3 seminar/laboratory	**1**
3.4 Total hours in the curriculum	**42**	Of which: 3.5 course	**28**	3.6 seminar/laboratory	**14**
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					**39**
Additional documentation (in libraries, on electronic platforms, field documentation)					**29**
Preparation for seminars/labs, homework, papers, portfolios and essays					**39**
Tutorship					**25**
Evaluations					**26**
Other activities:					

3.7 Total individual study hours	**158**
3.8 Total hours per semester	**200**
3.9 Number of ECTS credits	**8**

4. Prerequisites (if necessary)

4.1. curriculum	·
4.2. competencies	·

5. Conditions (if necessary)

5.1. for the course	· The requirements are posted here http://www.cs.ubbcluj.ro/~rares/course/pagd/
5.2. for the seminar /lab activities	· The requirements are posted here http://www.cs.ubbcluj.ro/~rares/course/pagd/

6. Specific competencies acquired

Professional competencies	· Define notions, concepts, theories and models of distributed and GPU architecture computing. · Critical analysis and use of the principles, methods and techniques work for quantitative and qualitative evaluation of the processes within distributed and GPU architectures · Apply basic concepts and theories in the field of distributed systems, programming methods and GPU architectures to professional project development
Transversal competencies	· Execution of the tasks required under specified requirements and the deadlines imposed, with the rules of professional ethics and moral conduct · Information and permanent documentation in its field · Seeking to improve business results by engaging in professional activities

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	· Introducing the students to general purpose GPU programming. The students should learn the following concepts: parallel architectures, GPU architecture, memory organization, parallelizable algorithms. The students should also learn to use GPU programming APIs and apply them practically in projects · Learning advanced concepts on GPU and distributed architecture computing · Distributed architectures
7.2 Specific objective of the discipline	· GPU o GPU architecture o Memory organization o Work scheduling o NVIDIA CUDA o OpenCL o Parallelizable algorithms · Distributed programming paradigms

8. Content

8.1 Course	Teaching methods	Remarks
Weeks 1 - 3 · Optimizing matrix multiplication · Flynn taxonomy; Examples and applications; GPU hardware architecture · Introduction to NVIDIA CUDA · Introduction to OpenCL	Exposition: presentation explanation, practical examples, demonstrations and case studies.	
Weeks 4 - 5 · Applications: Fractals · Applications: Ray tracing	Exposition: presentation explanation, practical examples, demonstrations and case studies.	
Weeks 6 - 8	Exposition:	

· GPU memory organization · GPU work scheduling: blocks, threads, atomicoperations, synchronization · Measuring performance	presentation explanation, practical examples, demonstrations and case studies.	
Weeks 9 - 10 Distributed programming paradigms Message passing, Client server, P2P, Message System, RPC/RMI/CORBA	Exposition: presentation explanation, practical examples, demonstrations and case studies.	
Weeks 11 - 12 Distributed programming paradigms Publish subscriber, Object Spaces	Exposition: presentation explanation, practical examples, demonstrations and case studies.	
Weeks 13 - 14 Distributed programming paradigms Mobile agents, Colaborative Applications	Exposition: presentation explanation, practical examples, demonstrations and case studies.	

Bibliography

1. Ananth Grama, George Karypis, Vipin Kumar, Anshul Gupta, Introduction to Parallel Computing, Addison-Wesley; 2 edition (January 26, 2003)
2. Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional; 1 edition (July 29, 2010)
3. Nicholas Wilt, The CUDA Handbook: A Comprehensive Guide to GPU Programming, Addison-Wesley Professional; 1 edition (June 11, 2013)
4. Aaftab Munshi, Benedict Gaster, Timothy G. Mattson, James Fung, Dan Ginsburg, OpenCL Programming Guide, Addison-Wesley Professional; 1 edition (July 23, 2011)
5. Matthew Scarpino, OpenCL in Action: How to Accelerate Graphics and Computations, Manning Publications (November 17, 2011)
6. Russ Miller, Laurence Boxer , Algorithms Sequential & Parallel: A Unified Approach, Cengage Learning; 3 edition (December 20, 2012)
7. BACON J. Concurrent Systems: Operating Systems, Database and Distributed Systems - an integrated approach. Addison-Wesley, 1998
8. BOIAN F.M. Programare distribuita în Internet; metode si aplicatii. Ed. Albastra, grupul Microinformatica, Cluj, 1997
9. BOIAN F.M. FERDEAN C.M., BOIAN R.F., DRAGOS R.C. Programare concurenta pe platforme Unix, Windows, Java. Ed. Albastra, grupul Microinformatica, Cluj, 2002
10. Gerassimos Barlas, Multicore and GPU Programming: An Integrated Approach, Morgan Kaufmann; 1 edition (December 1, 2014)
11. Raphael Couturier, Designing Scientific Applications on GPUs (Chapman & Hall/CRC Numerical Analysis and Scientific Computing Series), Chapman and Hall/CRC (November 21, 2013)

8.2 Seminar / laboratory	Teaching methods	Remarks
1. CUDA programming examples	Explanation, examples, case studies, dialog	
2. OpenCL programming examples	Explanation, examples, case	

	studies, dialog	
3. GPU Fractals	Explanation, examples, case studies, dialog	
4. GPU Ray tracing	Explanation, examples, case studies, dialog	
5. Distributed system project	Explanation, examples, case studies, dialog	
6. Distributed system project	Explanation, examples, case studies, dialog	
7. Project grading	Explanation, examples, case studies, dialog	

Bibliography

1. Ananth Grama, George Karypis, Vipin Kumar, Anshul Gupta, Introduction to Parallel Computing, Addison-Wesley; 2 edition (January 26, 2003)
2. Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional; 1 edition (July 29, 2010)
3. Nicholas Wilt, The CUDA Handbook: A Comprehensive Guide to GPU Programming, Addison-Wesley Professional; 1 edition (June 11, 2013)
4. Aaftab Munshi, Benedict Gaster, Timothy G. Mattson, James Fung, Dan Ginsburg, OpenCL Programming Guide, Addison-Wesley Professional; 1 edition (July 23, 2011)
5. Matthew Scarpino, OpenCL in Action: How to Accelerate Graphics and Computations, Manning Publications (November 17, 2011)
6. Russ Miller, Laurence Boxer , Algorithms Sequential & Parallel: A Unified Approach, Cengage Learning; 3 edition (December 20, 2012)
7. BACON J. Concurrent Systems: Operating Systems, Database and Distributed Systems - an integrated approach. Addison-Wesley, 1998
8. BOIAN F.M. Programare distribuita în Internet; metode si aplicatii. Ed. Albastra, grupul Microinformatica, Cluj, 1997

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- Learning the theoretical and methodological concepts and the practical aspects included in the course, students acquire the knowledge required by the Grid on - RNCIS for the partial competences
- The course follows the IEEE and ACM Curricula Recommendations for Computer Science studies.
- The course exists in the curricula of similar departments in Romania

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	Understand and be able to use the concepts and methods taught by the course	Written exam	40%

10.5 Seminar/lab activities	Ability to correctly solve practical problems from the course curricula	GPU project	30%
		Distributed project	30%
10.6 Minimum performance standards			
Ø Minimum final grade of 5			

Date	Signature of course coordinator	Signature of seminar coordinator
20.04.2018	Assoc.prof. Rareş Boian	Assoc. prof. Rareş Boian

Date of approval	Signature of the head of department
....	Prof.dr. Anca Andreica