Jackson State University
Department of Computer Science

Course Number and Title: CSC425.01 Parallel Computing
Semester and Year: Fall 2014
Instructor: Dr. Sugnbum Hong
Office Location: JAP 330
Office Hours: MWF 3 to 5 PM (Fall 2014)
E-Mail: Sungbum.hong@jsums.edu
Telephone: 601-979-5838

Required Text(s):

- *Introduction to Parallel Computing, Second Edition*, Ananth Grama, George Karypis, Vipin Kumar, Anshul Gupta, Addison-Wesley, 2003, ISBN: 0201648652

Required Readings/Resources (should include current knowledge)

- Jack Dongarra, Geoffrey C. Fox, Kai Hwang, *Distributed and Cloud Computing : From Parallel Processing to the Internet of Thinks* Morgan Kaufmann Publishers Inc. ISBN-139780123858801
- S Hane Cook , *CUDA Programming: A Developer's Guide to Parallel Computing with GPUs* Morgan Kaufmann Publishers Inc. ISBN-0124159338

Course Description

CSC 425 (3) Parallel Computing. Prerequisite: CSC325 Operating Systems. A study of the hardware and software issues in parallel computing . Theoretical and practical survey of parallel processing, including a discussion of parallel architectures, parallel programming languages, and parallel algorithms. Programming on multiple parallel platforms in a higher-level parallel language. It should also be useful for those who want to learn programming multi-core processors.

Prerequisites (if this section is not used, delete it from syllabus)

CSC325 Operating Systems

Course Objectives (must be measurable student outcomes)

Each student who successfully completes this course should be able to:
CO-1: Understand parallel computer architecture.
CO-2: Become skilled at multi-core programming with Open MP.
CO-3: Become skilled on programming on massively parallel architecture (GPU) with CUDA
CO-4: Become skilled on message passing programming with MPI
CO-5: Analysis Performance of parallel algorithms.
CO-6: Understand fundamental parallel algorithms: Dense matrix, Sorting, Graph algorithm, Search Algorithms, Dynamic Programming, and Fast Fourier Transform.

Course Content and Assignment Schedule

(This section should include more than table of contents from textbook or chapter outlines. It should identify, when possible, content knowledge the student is expected to master)

Course Outline

No.	**Topic name**	**References**	**CO**	**Weeks**
1	Introduction: Need for Parallel Computing; Scope of Parallel Computing; Issues in Parallel Computing	Ch1.		1
2	Models of Parallel Computing: Taxonomy of Parallel Architectures; Dynamic Interconnection Networks; Static Interconnection Networks; Message Transfer; Reduction, Parallel Prefix; GPU thread model	Ch 2	CO1	1
3	Performance Modeling: Metrics; Granularity; Scalability; Overhead;	Ch 5	CO5	1
4	Programming Using the message-Passing Paradigm	Ch 6.		1
		Project #1 with MPI	CO4	
5	Shared Memory Programming with Open MP (**Exam 1**)	Ch 7	CO1	2
		Project #2 with Open MP	CO2	
6	Matrix Algorithms: Matrix Partitioning, Matrix Transposition Matrix Vector Multiply, Matrix Multiply	Ch 8	CO6	1
7	Linear Equations: LU(P) Decomposition	Ch8.3	CO6	1
		Project #2 with CUDA	CO3	
8	CUDA , Fundamental Parallel Algorithms	Hand out #1		0.5
9	Thrust	Handout #2		0.5
10	Sorting Algorithms: Sorting Networks, Bubble Sort; Quick Sort; Bucket and Sample Sort (**Exam #2**)	Ch 9	CO6	2
		Project #4 With CUDA	CO3	
11	Graph Algorithms: Minimum Spanning Tree: Prim's Algorithm; Single source Shortest Paths: Dijkstra's Algorithms; All pair Shortest Path Algorithm	Ch10 2	CO6	1.5
		Project #4 With CUDA	CO3	
12	Searching and Optimization, Parallel Depth First Search Parallel Best first Search;	Ch11.2-4	CO3	1.5
13	Dynamic Programming: Serial DP Formulation; Shortest-path problem; 0/1 Knapsack Problem.; Non Serial DP formulation.	Ch 12.2-4	CO6	1.5
		Project #3 CUDA	CO3	
14	Fast Fourier Transform: Serial Algorithm; Binary-Exchange Algorithms; Transpose Algorithms	Ch 13	CO6	1.5
		Project #5 with CUDA	CO3	

Instructional Strategies

All students are required to do homework and one course project. Three assignments of homework and the course project will be completed through teamwork and collaborative learning. In such assignments, teams of students will be formed and learn collaboratively relevant core concepts in the course subject for completing three assignments of homework and one course project. The team collaboration will take place on the online discussion board. Students in each team will be assigned with different roles for the discussion and will in turn take

different roles during the discussion. Participation in dissuasion will be part of required course participation for developing teamwork and communication skills. However, each student required completing and submitting his or her homework and course project individually. The requirement for the teamwork and grading criteria for participation of collaborative learning will be provided.

Students will have an optional opportunity to earn bonus points by participating in self-assessment on their learning and skill. The self-assessment processes would promote students to think of their learning activities and strategy utilization , remind them of other available learning activities and strategies that they may not try before an may help them, identify their weakness and strength, and reflect on their success and failure to adjust their learning strategies and efforts.

Student Activities

Homework and Programming Projects: Students will be given at the end of each major topic.

- Students should submit their homework before start class and also is should be typed.
- Programming Projects can be done in the server that the instructor provided.

Online Quizzes and Pop Quizzes: Students will be given quizzes at the class room and through the Black Board. It should be done within a term or at the place that the instructor requested.

Exams (Test 1 and Test 2 Final) : There will be three exams during the semester (the two tests and Final Exam). The exams will cover the topics discussed throughout the semester and will be given during the scheduled exam times.

Method of Student Evaluation

Grading Policy:

- Homework and Quizzes	20%
- Lab and Programming Projects	20%
- Midterm Exam (test 1 and Test2)	40%
- Final Exam	20%
- Total	100%

Grading Scale

Range	90% - 100%	80% - 89%	70% - 79%	60% - 69%	below 60%
Letter	A	B			

Method of Course Evaluation

- SIRS
- Evaluation by students on their ability to perform in each of the course outcomes during the semester using Tests , Homework, Programming Projects, and Quizzes (ABET).

Special Needs Learners

(If you have a disability for which you are or may be requesting an accommodation, you are encouraged to contact both your instructor and the Office of the Americans with Disabilities Act (ADA) Coordinator and Compliance Office, P.O. Box 17999, Jackson, MS 39217; (601) 979-2485 as early as possible in the term.)

Diversity Statement

(Jackson State University is committed to creating a community that affirms and welcomes persons from diverse backgrounds and experiences and supports the realization of their human potential. We recognize that there are differences among groups of people and individuals based on ethnicity, race, socioeconomic status, gender, exceptionalities, language, religion, sexual orientation, and geographical area. All persons are encouraged to respect the individual differences of others.)

Caveat (if needed)

- Not allow students come/go, Not allow students to use cell-phone after class starts.
- Not allow to bring any type of food into class.
- No make-up tests will be given, except in cases of verifiable emergencies.
- All assignments must be turned in at the beginning of class on the day due.
- Late assignments will be deducted 20% each week for which they are late. Assignments handed in more than 2 weeks (including week-end days) after the deadline will not be accepted, nor marked. No exceptions will be made.

Class Attendance Policy

Students at Jackson State University must fully commit themselves to their program of study. One hundred percent (100%) punctual class attendance is expected from each student for all the scheduled classes and activities. Instructors keep attendance records and any absence for which a student does not provide written official excuse is counted as an unexcused absence. With or without official excuses, students are responsible for the work required during their absences.

Academic Honesty Statement

- Assignments must be done individually; you may not work in groups. You may not copy another person's work in any manner (electronically or otherwise). Furthermore, you must not give a copy of your work to another person. We will be randomly checking for similarities between programs, and you may be asked to present and explain your program to the instructor.
- Cheating will not be tolerated. Students guilty of cheating on a test or program will be given an F in the course. Remember that allowing others to copy your work is considered cheating.

Bibliography/References (current knowledge)

Fayez Gebali John , *Algorithms And Parallel Computing* Wiley & Sons Inc ISBN-13:9780470902103

Peter S. Pacheco , *An Introduction to Parallel Programming* Elsevier Science Ltd ISBN-13: 9780123742605

Andrea Marongiu and Luca Benini. *Efficient OpenMP support and extensions for MPSoCs with explicitly managed memory hierarchy*. In Proceedings of the Conference on Design, Automation and Test in Europe (DATE '09). Belgium, Belgium, 809-814.

Bensoudane, and Gabriela Nicolescu. *Parallel programming models for a multi-processor SoC platform applied to high-speed traffic management*. In Proceedings of the 2nd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis (CODES+ISSS '04). ACM, New York, NY, USA

Dimitri Komatitsch , Gordon Erlebacher , Dominik Göddeke , David Michéa, *High-order finite-element seismic wave propagation modeling with MPI on a large GPU cluster,* Journal of Computational Physics, v.229 n.20, p.7692-7714, October, 2010.

Eduardo Gurgel Pinho and Francisco Heron De Carvalho, Junior. *An object-oriented parallel programming language for distributed-memory parallel computing platforms*. Sci. Comput. Program. 80 (February 2014), 65-90.

Garnett Wilson and Wolfgang Banzhaf. *Deployment of parallel linear genetic programming using GPUs on PC and video game console platforms*. Genetic Programming and Evolvable Machines 11, 2 (June 2010), 147-184.

Guangming Tan, Ninghui Sun, and Guang R. Gao. *A parallel dynamic programming algorithm on a multi-core architecture*. In Proceedings of the nineteenth annual ACM symposium on Parallel algorithms and architectures (SPAA '07). ACM, New York, NY, USA, 135-144.

Guojing Cong. 2004. *On the Design and Implementation of Parallel Algorithms for Graph Problems on Shared-Memory Machines*. Ph.D. Dissertation. University of New Mexico, Albuquerque, NM, USA. Advisor(s) David A. Bader. AAI3156106.

Kengo Nakajima and Hiroshi Okuda. *Parallel Iterative Solvers for Unstructured Grids Using an OpenMP/MPI Hybrid Programming Model for the GeoFEM Platform on SMP Cluster Architectures*. In Proceedings of the 4th International Symposium on High Performance Computing (ISHPC '02), Hans P.

Koji Nakano. *An optimal parallel prefix-sums algorithm on the memory machine models for GPUs*. In Proceedings of the 12th international conference on Algorithms and Architectures for Parallel Processing - Volume Part I (ICA3PP'12), Springer-Verlag, Berlin, Heidelberg, 99-113.

J. Steven Kirtzic. 2012. *A Parallel Algorithm Design Model for the Gpu Architecture*. Ph.D. Dissertation. University of Texas at Dallas, Richardson, TX, USA. Advisor(s) Ovidiu Daescu.

Pablo Toharia, Oscar D. Robles, José L. Bosque, and Angel Rodríguez. *Video shot extraction on parallel architectures*. In Proceedings of the 4th international conference on Parallel and Distributed Processing and Applications (ISPA'06), Springer-Verlag, Berlin, Heidelberg, 869-883.

P Pacheco , *Parallel Programming with MPI* Morgan Kaufmann Publishers Inc. ISBN-13: 9781558603394

S. Tabik, E. M. Garz, I. Garc and J. J. Fern. *High performance noise reduction for biomedical multidimensional data*. Digit. Signal Process. 17, 4 (July 2007), 724-736.

Serban Georgescu , Peter Chow, *GPU accelerated CAE using open solvers and the cloud*, ACM SIGARCH Computer Architecture News, v.39 n.4, September 2011

Sayyed Ali Mirsoleimani, Ali Karami, and Farshad Khunjush. 2013. *A parallel memetic algorithm on GPU to solve the task scheduling problem in heterogeneous environments*. InProceeding of the fifteenth annual conference on Genetic and evolutionary computation conference (GECCO '13), Christian Blum (Ed.). ACM, New York, NY, USA, 1181-1188.

Srinivas Vadlamani. 2007. *Tools and Techniques for Efficient Parallelization on Modern Parallel Microprocessors.* Ph.D. Dissertation. California State University at Long Beach, Long Beach, CA, USA. Advisor(s) Stephen Jenk

T. K. Ralphs, L. L, danyi, and M. J. Saltzman. 2004. *A Library Hierarchy for Implementing Scalable Parallel Search Algorithms*. J. Supercomput. 28, 2 (May 2004), 215-234.

Tomislav Matić , Željko Hocenski, *Parallel processing with CUDA in ceramic tiles classification*, Proceedings of the 14th international conference on Knowledge-based and intelligent information and engineering systems: Part I, September 08-10, 2010, Cardiff, UK

Tomohiro Okuyama, Fumihiko Ino, and Kenichi Hagihara. 2008. A Task Parallel Algorithm for Computing the Costs of All-Pairs Shortest Paths on the CUDA-Compatible GPU. In Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA '08). IEEE Computer Society, Washington, DC, USA, 284-291.

Tabitha L. James, Reza Barkhi, and John D. Johnson. *Platform impact on performance of parallel genetic algorithms: Design and implementation considerations*. Eng. Appl. Artif. Intell.19, 8 (December 2006), 843-856.

Vikram S. Adve and Mary K. Vernon. 2004. *Parallel program performance prediction using deterministic task graph analysis*. ACM Trans. Comput. Syst. 22, 1 (February 2004), 94-136