August 30, 2010

CSE 487/587 **Data-intensive Computing** **Fall 2010**

I. Course Description

Data-intensive computing deals with storage models, application architectures, middleware, and programming models that address challenges in ultra-scale data. This course covers methods for transforming voluminous datasets (big data) into discoveries and intelligence for human understanding and decision making. Topics include: storage requirements of big data, organization of big data repositories such as Google File System (GFS), characteristics of Write-Once-Read-Many (WORM) data, semantic organization of data, data-intensive programming models such as MapReduce, fault-tolerance and performance, services-based cloud computing middleware, intelligence discovery methods, and scalable analytics and visualization. In particular, we will focus on methods for harnessing the cloud computing infrastructures such as Google App Engine (GAE), Amazon Elastic Compute Cloud (EC2), and Windows Azure.

II. Course outline: On completion of this course students will be able to analyze, design, and implement effective solutions for data-intensive applications with very large scale data sets. More specifically a student will be able to:

1. Describe data-intensive computing concepts
2. Compute with data-intensive computing concepts
3. Recognize a data-intensive problem.
4. Identify the scale of data.
5. Analyze the data requirements of a problem.
6. Describe the data layout and define the data repository format (Ex: GFS, Hive)
7. Decide the algorithms (Ex: MapReduce) and programming models (Ex: multi-core OpenMp)
8. Define application-specific algorithms and analytics (Ex: Finite State Element analysis)
9. Design the data-intensive program solution and system configuration.
10. Implement the data-intensive solution and test the solution for functional correctness and non-functional requirements.
11. Research algorithmic improvements and experiment with them.
12. Write a report summarizing the solution and results.
13. Incorporate services from cloud computing platforms.
14. Integrate semantic information in organizing data for context-awareness.
15. Apply collective intelligence methods for diverse data sources.
16. Formulate data-intensive visualization solutions for presenting the results.

III. Course Information

Website:	http://www.cse.buffalo.edu/~bina/cse487/fall2010
Instructor:	Bina Ramamurthy (bina@buffalo.edu)
Lecture Time:	MWF: 2.00-2.50PM
Lecture Location:	205 NSC
Office Hours:	TTh: 9.30-11.00AM
Office:	127 Bell Hall

IV. Grading Overall grade for the course will be based on the student's performance in: class attendance and participation (10%), 2 exams (40%), 4 projects (40%), presentation (10%)

V. Text Book There are two text books:

1. Cloud Application Architectures: Building Applications and Infrastructure in the Cloud (Theory in Practice) By George Reese, O'Reilly, **ISBN-10:** 0596156367
2. Algorithms of the intelligent web by H. Marmanis and D. Babenko, Manning Publications; 1 edition (June 3, 2009), **ISBN-10:** 1933988665.

VI. Tentative Schedule

Week of	Monday	Wednesday	Friday
30-Aug	Introduction to the course + the subject	Data-intensive computing problem space	Data-intensive computing solution space; ET0: Data structures and algorithms
6-Sep	Monday Labor day: No class	Cloud computing fundamentals; ET1: Web Services	Cloud computing models: demo amazon EC2 + Google App Engine
13-Sep	Project 1 Discussion: Performance through multi-core; exporting through web services	ET2: Windows Azure	Windows Azure web role, worker role, storage and working details.
20-Sep	ET3: Virtualization	Virtual appliances	Vmware;
27-Sep	ET4: Map-reduce programming model	Data mining with MapReduce	Analytics with MapReduce
4-Oct	Midterm review	Midterm	Makeup
11-Oct	ET5: Distributed data repositories: Hadoop distributed file system	Hama, Hive and other Hadoop related services	Project 2 discussion: Big-data computing through HDFS and MapReduce
18-Oct	Amazon EC2	S3 storage	Amazon cloud front
26-Oct	Google App Engine	Big table structure and API	AppSpot and templates
1-Nov	ET6: Open source cloud software: Eucalyptus	Cloud computing (contd.)	Cloud computing (contd.)
8-Nov	ET7: Apps and applications	Social networking data	large data sets and experiments
15-Nov	Performance; Reliability; Availability; and scalability	Privacy and security	Other issues in cloud computing: co-tenenacy, vulnerability; standardization
22-Nov	Monday only: Review	Thanks giving	Thanks giving
29-Nov	Intelligence discovery (ID)	ID applications	ID Applications
6-Dec	Review: project presentations	Review: Project presentations	Review: Project presentations

VII. Project Plans

Each project will involve complete installation of all the necessary toolkits, software packages and servers by each student (or group of students) in their workspace. Students will also write a detailed technical report on the project they implement. Students can work in groups of no more than 2 people. Choose the group members with complementary expertise.

Project 1: A simple Web Application deployment on Google Apps: See the prototype working at the links:
http://11.latest.healthmanage1.appspot.com/templates/main.html
http://healthmanage1.appspot.com/
Learning objective: Studying the capability of GAE cloud environment and its use data-intensive computing.
Project 2: Single node MapReduce data-intensive computing method using VMware virtualization; Introduction to Amazon EC2 environment; MapReduce on Amazon EC2.
Learning goal: Virtualization the enabling technology of the cloud; Amazon EC2 introduction. Data-intensive MapReduce with and without the cloud.
Project 3: Simple introduction to Amazon Elastic Compute Cloud (Amazon EC2).
Learning goal: Deploying common data-intensive applications on Amazon EC2 and creating AMIs.
Project 4: Introduction to Microsoft Azure: Development and production environment; Web role and worker role. Blob data-intensive storage
Learning goal: Azure cloud environment supporting data-intensive computing.

VIII. Suggested Reading List

1. K.A. Delic and M.A. Walker. Emergence of the Academic Computing Cloud. ACM Ubiquity, *Ubiquity Volume 9, Issue 31 (August 5 - 11, 2008), http://www.acm.org/ubiquity/volume_9/v9i31_delic.html, last viewed January 2009.*
2. J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December, 2004, 137-150, http://labs.google.com/papers/mapreduce.html, last visited December 2007.
3. J. Gray. Distributed Computing Economies. Computer Systems Theory, Technology and Applications, From Object-Relational Mappers Vol. 6, No. 3 - May/June 2008, http://www.acmqueue.org/modules.php?name=Content&pa=showpage&pid=545, last viewed December 2008.
4. The Hadoop Project. http://hadoop.apache.org/, last visited March 2008.
5. M. Johnson, R. H. Liao, A. Rasmussen, R. Sridharan, D. Garcia and B.K. Harvey. Infusing Parallelism into Introductory Computer Science Curriculum using MapReduce, EECS Department, University of California, Berkeley, April 2008, http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-34.html, Technical Report: UCB/EECS-2008-34.
6. B. McColl. Cloud N: New Directions in Massively Parallel Cloud Computing. http://cloudn.com. Last viewed January 2009.
7. T. White. Running Hadoop MapReduce on Amazon ECS and Amazon S3, http://developer.amazonwebservices.com/connect/entry.jspa?externalID=873, last visited March 2008.
8. I. Horrocks. Ontologies and the Semantic web. Communications of the ACM. pp.58-67, December 2008.
9. Google Code University: Material for Students and Educators: http://code.google.com/edu/parallel/index.html

IX. Course Policies

Attendance Policy: You are responsible for the contents of all lectures and recitations (your assigned section). If you know that you are going to miss a lecture or a recitation, have a reliable friend take notes for you. Of course, there is no excuse for missing due dates or exam days. We do, however, reserve the right to take attendance in both lecture and recitation. We may use this information to determine how to resolve borderline grades at the end of the course, especially if we see a lack of attendance and participation during lecture sessions. During lectures, we will be covering material from the textbook. We will also work out several of the problems from the text. Lecture will also consist of the exploration of several real world Operating System problems not covered in the book. You will be given a reading assignment at the end of each lecture for the next class.

Incomplete Policy: We only grant incompletes in this course under the direst of circumstances. By definition, an incomplete is warranted if the student is capable of completing the course satisfactorily, but some traumatic event has interfered with their capability to finish within the timeframe of the semester. Incompletes are not designed as stalling tactic to defer a poor performance in a class.

Academic Integrity Policy: UB's definition of Academic Integrity in part is, "Students are responsible for the honest completion and representation of their work". It is required as part of this course that you read and understand the departmental academic integrity policy located at the following URL:
http://www.cse.buffalo.edu/undergrad/policy_academic.php
There is a very fine line separating conversation pertaining to concepts and academic dishonesty. You are allowed to converse about general concepts, but in no way are you allowed to share code or have one person do the work for others. You must abide by the UB and Departmental Academic Integrity policy at all times. **NOTE:** Remember that items taken from the Internet are also covered by the academic integrity policy! If you are unsure if a particular action violates the academic integrity policy, assume that it does until you receive clarification from the instructor. *We reserve the right to check or question any portion of any work submitted at any time during the semester or afterwards.* If you are caught violating the academic integrity policy, you will minimally receive a ZERO in the course.

Exams Policy: There will be a midterm (Exam 1) that will be administered and graded before the resign date. Midterm material will cover all lecture and reading assignments before the exam, as well as concepts from the project assignments. Midterms are closed book, closed notes, and closed neighbor. The second exam (Exam 2) will be covering all lecture material after exam1 and all the projects. We do not give make up exams for any reason. If you miss an exam, you will receive a zero for that portion of the grade.

Students with Disabilities:

If you have special needs due to a disability, you must be registered with the Office of Disability Services (ODS). If you are registered with ODS please let your instructors know about this so that they can make special arrangements for you.