BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
WORK INTEGRATED LEARNING PROGRAMMES

COURSE HANDOUT

Part A: Content Design

Course Title	Systems for Data Analytics
Course No(s)	DSE* ZG517
Credit Units	5
Course Author	Prof. Shan Balasubramaniam
Version No	1
Date	26 / April / 2019

Course Description

Course Objectives

CO1	Introduce students to a systems perspective of data analytics: to leverage systems effectively, understand, measure, and improve performance while performing data analytics tasks
CO2	Enable students to develop a working knowledge of how to use parallel and distributed systems for data analytics
CO3	Enable students to apply best practices in storing and retrieving data for analytics
CO4	Enable students to leverage commodity infrastructure (such as scale-out clusters, distributed data-stores, and the cloud) for data analytics.

Text Book(s)

T1	Kai Hwang, Geoffrey Fox, and Dongarra. Distributed Computing and Cloud Computing. Morgan Kauffman
T2	

Reference Book(s) & other resources

R1	Nikolas Roman Herbst, Samuel Kounev, Ralf Reussner. Elasticity in cloud computing: What it is, and what it is not. 10th International Conference on Autonomic Computing (ICAC '13). USENIX Association.
R2	Mohammed Alhamad, Tharam Dillon, Elizabeth Chang.Conceptual SLA Framework for Cloud Computing.4th IEEE International Conference on Digital Ecosystems and Technologies. April 2010, Dubai, UAE.
R3	Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The Google File System. SOSP'03, October 19–22, 2003, Bolton Landing, New York, USA.
R4	Apache CouchDB. Technical Overview. http://docs.couchdb.org/en/stable/intro/overview.html
R5	Apache CouchDB. Eventual Consistency. http://docs.couchdb.org/en/stable/intro/consistency.html
R6	Seth Gilbert and Nancy A. Lynch. Perspectives on the CAP Theorem. IEEE Computer. vol. 45. Issue 2. Feb. 2012
R7	Werner vogels. Eventually Consistent. january 2009. vol. 52. no. 1 Communications of the acm.
R8	Eric Brewer.CAP Twelve Years Later: How the "Rules" Have Changed. IEEE Computer. vol. 45. Issue 2. Feb. 2012
R9	M. Burrows, The Chubby Lock Service for Loosely-Coupled Distributed Systems, in: OSDI'06: Seventh Symposium on Operating System Design and Implementation, USENIX, Seattle, WA, 2006, pp. 335–350.
R10	MATEI ZAHARIA et. al. Apache Spark: A Unified Engine for Big Data Processing .COMMUNICATIONS OF THE ACM | NOVEMBER 2016 | VOL. 59 | NO. 11.
R11	YASER MANSOURI, ADEL NADJARAN TOOSI, and RAJKUMAR BUYYA. Data Storage Management in Cloud Environments:Taxonomy, Survey, and Future Directions . ACM Computing Surveys, Vol. 50, No. 6, Article 91. December 2017

Modular Content Structure

Modular Structure

#	Topics
1	Introduction to Data Engineering
1.1	Systems Attributes for Data Analytics - Single System
	Storage for Data: Structured Data (Relational Databases) , Semi-structured data (Object Stores), Unstructured Data (file systems)

	Processing: In-memory vs. (from) secondary storage vs. (over the) network
	Storage Models and Cost: Memory Hierarchy, Access costs, I/O Costs (i.e. number of disk blocks accessed);
	Locality of Reference: Principle, examples
	Impact of Latency: Algorithms and data structures that leverage locality, data organization on disk for better locality
1.2	<u>Systems Attributes for Data Analytics - Parallel and Distributed Systems</u>
	Motivation for Parallel Processing (Size of data and complexity of processing)
	Storing data in parallel and distributed systems: Shared Memory vs. Message Passing
	Strategies for data access: Partition, Replication, and Messaging
	Memory Hierarchy in Parallel Systems: Shared memory access and memory contention; shared data access and mutual exclusion
	Memory Hierarchy in Distributed Systems: In-node vs. over the network latencies, Locality, Communication Cost
2	<u>Systems Architecture for Data Analytics</u>
2.1	<u>Introduction to Systems Architecture</u>
	Parallel Architectures and Programming Models: Flynn's Taxonomy (SIMD, MISD, MIMD) and Parallel Programming (SPMD, MPSD, MPMD)
	Parallel Processing Models:, {Data, Task, and Request}-Parallelism; Mapping: Data Parallel - SPMD, Task Parallel - MPMD, Request Parallel - Services/ Cloud, Client-Server vs. Peer-to-Peer models of distributed Computing.
	Parallel vs. Distributed Systems: Shared Memory vs. Distributed Memory (i.e. message passing) Motivation for distributed systems (large size, easy scalability, cost-benefit)
	Cluster Computing: Components and Architecture.
2.2	<u>Performance Attributes of Systems</u>
	Scalability - Speedup and Amdahl's Law; How to apply Amdahl's Law? (Relation to Barsis-Gustafson Law?)
	Impact of Memory Hierarchy on Performance: ● Shared Memory and Memory Contention ● Communication Cost ● Locality
	Reliability (for distributed systems): MTTF and MTTR, Serial vs. Parallel Connections, Single Point-of-Failure

	Building Reliable Systems: Redundancy and Resilience; Failure Models in Distributed systems: Transient vs. Permanent Failures,
	Failure Recovery: Fail-over, Active Fail-over etc Process Migration
	Availability: Calculating Availability; Service Agreements and SLAs
	Elasticity: Resilient Performance and Scenarios; Calculating Elasticity; Achieving elasticity (via resource provisioning and virtualization)
3.	Data Storage and Organization for Analytics:
	File systems vs. Database systems. Vs. Object Stores
	Distributed File Systems - Basic architecture, Case Studies (GFS/HDFS)
	Unstructured Databases - Basic architecture, Case Study and Examples (Google BigTable, CouchDB / MongoDB)
	Consistency Models - Weak and Strong Consistency, Eventual Consistency, CAP Theorem - Result and Implications;
	Synchronization: Chubby Locking as a case study.
4.	<u>Distributed Data Processing for Analytics</u>
4.1	<u>(Re-)Designing Algorithms for Distributed Systems</u>
	Design Strategy: Divide-and-conquer for Parallel / Distributed Systems - Basic scenarios and Implications
	Parallel Programming Pattern: Data-parallel programs, and *map* as a construct
	Parallel Programming Pattern: Tree-parallelism, *reduce* as a construct
	Map-reduce model: Examples (of map, reduce, map-reduce combinations, Iterative map-reduce)
	Batch processing vs. Online Processing; Streaming - Systems-level understanding (input-output, memory model, constraints)
	Master-Slave Processing: Implications for speedup and communication cost
4.2	<u>Distributed Data Analytics</u>
	● Partitioning vs. Replication and Communication vs. Locality for Data Mining algorithms like k-means, DBSCAN, Nearest Neighbor ● Using data structures (such as kd-trees) for partitioning) ● Matrices and Locality - Row-major vs. Column major vs. Blocking in distributed context

<u>Learning Outcomes:</u>

No	Learning Outcomes
LO1	[to be done]
LO2	[to be done]
LO3	[to be done]
LO4	[to be done]

Part B: Contact Session Plan

Academic Term	
Course Title	Systems for Data analytics
Course No	DSE* ZG517
Lead Instructor	Prof. Anindya Neogi

Course Contents

Contact Session # **(2 hours / Session)**	**Topic # (from content structure in Part A)**	**List of Topic Title (from content structure in Part A)**	Reading / Reference
1	1.1	Systems Attributes for Data Analytics - Single System	
		Storage for Data: Structured Data (Relational Databases) , Semi-structured data (Object Stores), Unstructured Data (file systems)	T1 Sec. 1.2.3
		Processing: In-memory vs. (from) secondary storage vs. (over the) network	

		Storage Models and Cost: Memory Hierarchy, Access costs, I/O Costs (i.e. number of disk blocks accessed);	
2	1.1	Locality of Reference: Principle, examples	
		Impact of Latency: Algorithms and data structures that leverage locality, data organization on disk for better locality	
3	1.2	Systems Attributes for Data Analytics - Parallel and Distributed Systems	
		Motivation for Parallel Processing (Size of data and complexity of processing)	T1. Sec. 1.4.3
		Storing data in parallel and distributed systems: Shared Memory vs. Message Passing	
4	1.2	Strategies for data access: Partition, Replication, and Messaging	
5	1.2	Memory Hierarchy in Parallel Systems: Shared memory access and memory contention; shared data access and mutual exclusion	
		Memory Hierarchy in Distributed Systems: In-node vs. over the network latencies, Locality, Communication Cost	

6	2.1	Introduction to Systems Architecture Parallel Architectures and Programming Models: Flynn's Taxonomy (SIMD, MISD, MIMD) and Parallel Programming (SPMD, MPSD, MPMD)	T1 Sec. 1.4.3
		Parallel Processing Models:, {Data, Task, and Request}-Parallelism; Mapping: Data Parallel - SPMD, Task Parallel - MPMD, Request Parallel - Services/Cloud, Client-Server vs. Peer-to-Peer models of distributed Computing.	
7	2.1	Parallel vs. Distributed Systems: Shared Memory vs. Distributed Memory (i.e. message passing) Motivation for distributed systems (large size, easy scalability, cost-benefit)	T1 Sec. 1.4.3 T1 Sec. 2.1
		Cluster Computing: Components and Architecture.	T1 Sec. 2.2.1 to 2.2.4, Sec 2.3
8	2.2	Impact of Memory Hierarchy on Performance: ● Shared Memory and Memory Contention ● Communication Cost ● Locality	Additional Reading
		Reliability (for distributed systems): MTTF and MTTR, Serial vs. Parallel Connections, Single Point-of-Failure	T1 Sec. 1.5.2 and 2.3.3
		Mid Term Portion - Review	

9	2.2	Building Reliable Systems: Redundancy and Resilience; Failure Models in Distributed systems: Transient vs. Permanent Failures,	T1 Sec. 1.5.2 and 2.3.3
		Failure Recovery: Fail-over, Active Fail-Over etc Overview of Process Migration	T1 Sec. 1.5.2 and 2.3.3
		Availability: Calculating Availability;	T1 Sec. 1.5.2
10	3.1	File systems vs. Database systems. Vs. Object Stores	
		Distributed File Systems - Basic architecture, Case Studies (GFS/HDFS)	T1 Sec. 6.3.2 AR. Google File System paper
11	3.1	Unstructured Databases - Basic architecture, Case Study and Examples (Google BigTable, CouchDB / MongoDB)	T1 Sec. 6.3.3
12	3.1	Overview of Consistency Models - Weak and Strong Consistency, Eventual Consistency, CAP Theorem - Result and Implications;	AR - papers on consistency and CAP
Additional Content	3.1	Synchronization: Chubby Locking as a case study. [supplementary video to be added. Not to be done in Class]	AR - paper on Chubby
13	4.1	(Re-)Designing Algorithms for Distributed Systems	
		Design Strategy: Divide-and-conquer for Parallel / Distributed Systems - Basic scenarios and Implications	Notes

		Parallel Programming Pattern: Data-parallel programs, and *map* as a construct	T1 Sec. 6.2.1
		Parallel Programming Pattern: Tree-parallelism, *reduce* as a construct	T1 Sec. 6.2.2
14-15	4.1	Map-reduce model: Examples (of map, reduce, map-reduce combinations, Iterative map-reduce)	T1 Sec. 6.2.2
		Batch processing vs. Online Processing; Streaming - Systems-level understanding (input-output, memory model, constraints)	AR - Spark Paper
16	4.1	Master-Slave Processing: Implications for speedup and communication cost	Notes
	4.2	• Parallelization of Data mining algorithms like k-means, DBSCAN, Nearest Neighbor & identifying locality issues • Matrices and Locality - Row-major vs. Column major vs. Blocking in distributed context	AR - Notes

 # The above contact hours and topics can be adapted for non-specific and specific WILP programs depending on the requirements and class interests.

Select Topics for experiential learning

Topic No.	Select Topics in Syllabus for experiential learning	Resources (Need Weka or equivalent software)
1	Programming exercises on map-reduce	[Resources: Cloud Infra. Lab in Hyd.]

2	Setting up a simple 3-tier application on the Cloud	[Resources: Amazon student license]
3	Synchronization exercise on CouchDB	[Resources: Cloud Infra. Lab or Amazon student license]
4	Pen-and-paper exercise on Locality, Memory Contention, and Communication Requirement	
5	Pen-and-paper exercise on calculations of speedup, MTTF, and MTTR.	

Evaluation Scheme

Legend: EC = Evaluation Component

No	Name	Type	Duration	Weight	Day, Date, Session, Time
EC-1	Assignment	Take Home		10	To be announced
	Quiz 1	Take Home		5	To be announced
	Quiz II	Take Home		5	To be announced
EC-2	Mid-Semester Test	Open Book		30	To be announced
EC-3	Comprehensive Exam	Open Book		50	To be announced

Note *- Evaluation components can be tailored depending on the proposed model.*

Important Information

Syllabus for Mid-Semester Test (Open Book): Topics in Weeks 1-7
Syllabus for Comprehensive Exam (Open Book): All topics given in plan of study

Evaluation Guidelines:

1. EC-1 consists of an Assignment and two Quizzes. Announcements regarding the same will be made in a timely manner.
2. For Closed Book tests: No books or reference material of any kind will be permitted. Laptops/ Mobiles of any kind are not allowed. Exchange of any material is not allowed.
3. For Open Book exams: Use of prescribed and reference text books, in original (not photocopies) is permitted. Class notes/slides as reference material in filed or bound form is permitted. However, loose sheets of paper will not be allowed. Use of calculators is permitted in all exams. Laptops/ Mobiles of any kind are not allowed. Exchange of any material is not allowed.
4. If a student is unable to appear for the Regular Test/Exam due to genuine exigencies, the student should follow the procedure to apply for the Make-Up Test/Exam. The genuineness of the reason for absence in the Regular Exam shall be assessed prior to giving permission to appear for the Make-up Exam. Make-Up Test/Exam will be conducted only at selected exam centres on the dates to be announced later.

It shall be the responsibility of the individual student to be regular in maintaining the self-study schedule as given in the course handout, attend the lectures, and take all the prescribed evaluation components such as Assignment/Quiz, Mid-Semester Test and Comprehensive Exam according to the evaluation scheme provided in the handout.