BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, Pilani
Pilani Campus
AUGS/ AGSR Division

FIRST SEMESTER 2019-20
COURSE HANDOUT

Date: 18.08.2020

In addition to part-I (General Handout for all courses appended to the time table) this portion gives further specific details regarding the course

Course No. : **CS / SS G527**
Course Title **:** **Cloud Computing**
Instructor**:** **Hari Babu K (khari@pilani.bits-pilani.ac.in)**
Course Webpage: https://canvas.instructure.com/courses/2263144

1. Scope and Objective:

The primary objective of the course is to introduce the student to cloud computing from architectural and design perspectives. As such, the emphasis of the course would be on the underlying infrastructure and architecture of clouds, techniques for enabling services and the quality of such services, as well as issues in designing clouds. Select research issues in performance, security, and management would also be addressed. Programming on the cloud would be encouraged but not taught in class. Students are expected to learn and understand tools and techniques for using, designing, and implementing clouds and services via assignments and a term project.

2. Text and References:

a.Text Book: NONE

b.References:

R1. Kai Hwang , Jack Dongarra , Geoffrey C. Fox *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. Morgan Kauffman 2011.

R2. Jim Smith, Ravi Nair. *Virtual Machines: Versatile Platforms for Systems and Processes.* Morgan Kaufmann. 2005

AR. Additional references (papers) to be posted online.

3. Course Plan:

3a. Modules

Mo d- ule	Theme	Learning Objectives
I	Cloud Computing – Introduction, Principles, and Issues.	• To understand the motivation for Cloud Computing. • To understand the underlying (distributed) computing model.

II	Cloud Architecture – Resources and Virtualization	• To understand how to leverage and provision computing resources at different levels of abstraction. • To understand virtualization techniques at different levels of abstraction. • To understand how to architect a cloud to suit different requirements
III	Programming for the Cloud and Application Models	• To understand the execution of applications on the cloud • To understand how to develop & deploy applications for the cloud and the relevant tools & technologies
IV	Services, Service Models, and QoS.	• To understand how to use the cloud to deliver Software as a Service. • To understand how to deliver computing Infrastructure (e.g. processors, storage, network) as a Services • To understand Quality of Service issues and QoS support mechanisms for Services on the Cloud
V	Big Data	• To understand techniques and tools for handling large volumes of data; • To understand the structure and issues of file systems and databases for large volumes of data.
VI	Cloud Management, Performance and Security Issues	• To understand how to manage a cloud platform and a services ecosystem • To understand performance issues and techniques to enable performance of a cloud at different levels of abstraction • To understand security issues specific to cloud computing and solutions to address them.

[Note: Module III will not be covered in detail in class. It is to be learnt primarily through assignments and project. End of Note.]

3.b Lecture Schedule:

Lecture	Module	Topics	Readings
1	I	Evolution of Computing Systems – Mainframes to PCs to Networked Systems to Clouds. Cloud Computing and Services – User's Perspective: Economic / Business Motivation.	R1 Sec. 1.2 and 1.4.1
2		Spectrum of Parallel and Distributed Computers – Programmer's Perspective. Forms of Parallelism: Data Parallelism vs. Task Parallelism vs. Request Parallelism.	R1 Sec. 1.3.4

3		Distributed Computing Models – Message Passing, Failures, and Buffering.	NONE
4 - 6		Introduction to Computer Clusters: Structure and Components of a cluster – nodes, interconnect, and middleware. Design of a cluster: Single System Image, Design for availability, Fault-tolerance and Fault Recovery - Reliability and Availability.	R1 Sec. 2.1 to 2.3, AR
7		Computer Clusters: Task Management and Resource Management; Scheduling and Load Balancing.	R1 Sec. 2.4, AR
8		High Availability Clusters: Design Issues. Check-pointing and Recovery.	R1. Sec. 2.3.2 to 2.3.4
9-12	III	Programming on Clusters- Speedup and Scalability; Scale-out Clusters: Map-reduce Architectures; In-memory Distributed Programming; Platform Architectures and Programming Models	R1 Sec. 1.4.3 and 6.2.2, AR
13		Building Microservices – Programming Models and Platform Example	
14		Virtualization: Introduction, Different Levels of Abstraction, and Mechanisms for Virtualization. Process vs. System VMs, HLL VMs.	R2 Ch. 1, AR
15		Process VMs: Structure and Behavior.	R2 Sec. 3.1, AR
16		System VMs : Applications and Implementation Models.	R2 Sec. 8.1
17		System VMs: Resource Virtualization and ISA Virtualizability	R2 Sec. 8.2, AR
18	II	System VMs: Virtualization of Memory and I/O	R2 Sec. 8.3-8.4
19		Case Studies: VMWare/Xen	AR
20-21		Storage Systems and Storage Virtualization: Storage Devices, File Systems and Volumes, Storage Networks – NAS and SAN, NAS Internals, Virtual Storage and Storage on the Cloud. (Case Study: DropBox / Google Drive)	AR
22-23		Virtualized Networks and Virtual Clusters. Process Migration and VM Migration. Live Migration – Suspend and Resume.	R1 Sec. 3.4, R2 Sec. 10.2, AR
24		SDN in the Data Center	AR
25-26		Containers – Light-Weight VMs, Comparison with VMs; Case Study (Dockers/LXC). Scheduling and Load Balancing of VMs and Containers	AR
27		Elasticity	AR

28	IV	Services on the Cloud. Software as a Service; SOA; REST; Web Services;	R1 Sec. 5.1, AR
29- 30		Infrastructure as a Service (IaaS), Storage as a Service (StaaS) Platform as a Service – Features, Programming Models (e.g. Serverless Architectures, AWS MapReduce, Amazon EBS, S3, SimpleDB)	AR
31		Resource Scaling and Capacity Management: Managing VMs Resource Provisioning. Quality of Service – Models, Parameters, and Metrics. SLAs.	AR
32-33	V	Big Data: Consistency Protocols- 2-Phase and 2-Phase Commit Protocol, Paxos Protocol. Case Study: Chubby locking	AR
34-35		Big Data: Consistency Protocols – Implementation and Practical Platforms. Case Studies: Zookeeper, , Raft	AR
36		Big Data File Systems- Case study: Google FS / Hadoop Distributed FS;	AR
37-38		Big Data: CAP Theorem; PACELC theorem; NoSQL Databases.	AR
39		Handling Large Data: Big Data Issues; Programming and Storage Models: GFS/HDFS, NoSQL, Google BigTable, Google Spanner, / AWS DynamoDB etc.	
40-41	VI	Cloud Security	AR
42		Cloud Economics – Service Pricing	AR

3.c. Practicum

Practical work by students is expected to carried out at the students' own convenience via assignments and term project. The *assignments* are primarily meant for students to *learn implementation skills with some design*. On the other hand, *term project* is meant for the student to *explore a problem and/or solution through readings, carry out a new design, and implement as well*.

Both assignments and term project will require students to *work in a team*, *set up developmental / experimental platform(s)* and *conduct performance studies* on their implementations. The term project will also require students to *articulate their solutions and results* through interactions with the instructor and through design / implementation documents.

4. Evaluation

4. a. Evaluation Scheme:

Component	Weight	Date & Time	Remarks
Test1	10%	<TEST_1>	Closed Book
Test2	10%		Closed Book
Test3	10%		Open Book
Class Tests / Quizzes	10%	Announced in class	Small exercises either take-home or In-class
Assignments (3)	30%	TBA	Programming, Take Home
Comprehensive Examination	30%	As per A0055GSD Timetable	

4. b. Make-up Policy:

- **Assignments / Term Project:**
  - **No Make-up will be available for assignments or term project. Late submissions will be evaluated at 25% less weight for that component for a delay of up to 24% hours after which no submissions will be accepted.**
- **Test:**
  - **Prior Permission of the Instructor-in-Charge is usually required to get a make-up for a test/quiz.**
  - **A make-up test shall be granted only in genuine cases where - *in the Instructor's judgment* - the student would be physically unable to appear for the test. Instructor's decision in this matter would be final.**
- **Comprehensive Exam:**
  - **Make-up for the comprehensive exam may be applied only with the Associate Dean of Under-Graduate Studies or Associate Dean of Graduate Studies / Research.**

4.c. Fairness Policy:

- **Students are expected to work within their team on assignments / project expect where explicitly instructed / permitted otherwise.**
- **When students are allowed to consult/discuss with other students/teams such consultation/discussion should be explicitly acknowledged and reported to the instructor prior to evaluation.**
- **When students are expected to collaborate within a team:**
  - **Individual contributions should be identified and documented in qualitative and quantitative terms by each team member.**
  - **Instructor may assess and mark each individual in a team separately.**

- **The instructor's assessment of the contributions in this matter would be final.**

5. Consultation Hours: To be notified (via the course website on *nalanda*)

6. Notices: All notices concerning this course will be displayed online only (i.e on the course website). If there is a need, email would be used on short notice (12 hours) – only BITS Pilani email would be used.

Instructor –In- Charge
CS / SS G527