USN | | | | | | | | | | |

Internal Assessment Test 3 – JULY 2022

| Sub: | Cloud Computing & Its Applications | | | | | Sub Code: | 18CS643 | Branch: | CSE |
|---|---|---|---|---|---|---|---|---|---|
| Date: | 11-07-2022 | Duration: | 90 mins | Max Marks: | 50 | Sem / Sec: | VI/ A&B&C | | OBE |

Answer any FIVE FULL Questions

| | | MARKS | CO | RBT |
|---|---|---|---|---|
| 1 | a) Map Reduce Programming model is used to analyze huge volumes of data but it is not applicable for all types of applications. Mention the variations and extensions of Map Reduce to overcome the above-mentioned Limitations.<br><br>b) List the parameters that can be controlled in Aneka Map Reduce Programming model.<br><br>**SOLUTION:**<br>a)<br>MapReduce exhibit few limitations on the abstractions. It might require considerable effort to design map and reduce for complex data. Few extensions have been provided. They focus on extending MapReduce application space and providing an easier interface to developers for designing the distributed algorithms.<br>1. Hadoop<br>2. Pig<br>3. Hive<br>4. Map-Reduce-Merge<br>5. Twister<br>Hadoop: Open source implementation of MapReduce framework supported by GFS. HDFS and Hadoop Map Reduce. Initially developed and supported by Yahoo. Robust community for developers and users. Yahoo runs the world's largest Hadoop cluster composed by 40000 machines and made it available for many institutions.<br>Pig: Platform for analysing the large data sets. Developed as an Apache project that consists of high-level language for expressing data analysis programs. It is coupled with infrastructure for evaluation. Consists a compiler that produces a sequence of MapReduce jobs that can be run on top of distributed infrastructure. Developers can express the data analysis programs in language called Pig Latin. Exposes SQL like interface.<br>Hive: Another Apache initiative provides data warehouse infrastructure on top of Hadoop MapReduce. Provides data summarization, easy automation and analysis of large data sets. Has the ability to scale out since it is Hadoop based framework.<br>b)<br>Partitions: Property stores an integer number containing the number of partitions into which final results will be divided. Also decides the number of reducer tasks. Default value is 10.<br>Attempts: Number of times that the runtime will retry to execute a task before declaring it failed. Default value is 3.<br>UseCombiner: Property stores a boolean value indicating whether a MapReduce runtime should add a combiner phase. This is used to reduce the number of intermediate files passed to reduce. Default value is true.<br>SynchReduce: Property stores a boolean value indicating whether to synchronize the reducers or not. Default value is true.<br>IsInputReady: Boolean property indicating whether the input files are ready. Or have to be uploaded by the client manager before execution. Default value is false.<br>Fetch Results: Boolean property indicating whether the client manager needs to download the results to the local computer. Default value is true. | [5+5] | 2 | L2 |
| 2 | a)Describe the importance of Data-Intensive Computing. And also discuss the open challenges associated with Data-Intensive Computing.<br>b) Other than analyzing huge volumes of data, List out various other problems that Map Reduce can solve.<br><br>**SOLUTION** | [5+5] | 2 | L2 |

| | | | | |
|---|---|---|---|---|
| | a) Data-intensive computing focuses on a class of applications that deal with a large amount of data. Several application fields like Computational science to social networking produce large volumes of data. This data need to be efficiently stored, made accessible, indexed and analyzed. These tasks becomes challenging as the quantity of information increases overtime at higher rates.<br><br>Ex1: Earthquake simulators, process a massive amount of data produced across the entire globe.<br>Ex2: Telecom Company.<br>Ex3: Social Networking sites and Games.<br>**Challenges:**<br><br>**1.** Metadata management that can scale to manage complex, heterogeneous and distributed data sources.<br>2. Need for scalable algorithms that can search massive datasets.<br>3. New approaches to software mobility.<br>4. Specialized hybrid interconnection architectures.<br>5. Flexible and high-performance software integration techniques.<br><br>b) The various problems that Map-reduce can solve are as follows:<br><br>1. **Distributed Grep**: Map reduce leverages a parallel and faster execution of this operation.<br>2. **Count of URL-Access Frequency:** Map [<URL,1>], Reduce [<URL:Count>]<br>3. **Reverse web-link graph:** Keeps track of all web pages might lead to a given link.<br>4. **Term-Vector per Host:** <word, frequency>; <host, document><br>5. **Inverted Index:** About the presence of the words.<br>6. **Distributed Sort:** parallelize the execution of a sort operation over a large number of records. | | | |
| 3 | Design and implement a simple program for Aneka that uses MapReduce for counting the words in file.<br><br>**SOLUTION**<br>Public abstract class Mapper <K,V>: MapperBase {<br><br>Public void Map(IMapInput Input, MapEmitDelegate emit){<br>……<br>}<br><br>Public override Type GetKeyType() {<br><br>Return typeof(K);<br>}<br><br>}<br>public class WordCounterMapper: Mapper<long, String> {<br>Protected override Map(IMapInput<long,string> input) {<br>String value = input.value;<br>String [] words = value.split (“\t\n!-={}.”.ToCharArray(), StringSplitOptions.RemoveEmptyEntries);<br>For each(String word in Words){<br>this.Emit(word,1);<br>} } } }<br>public abstract class Reducer <K,V>: ReducerBase {<br>Public void Reduce(IReduceInputEnumerator input, ReduceEmitDelegate emit)<br>{ …….. }<br>Public override Type GetKeyType() {<br><br>Return typeof(K);<br>}<br>public class WordCounterReducer: Reducer <String, int> {<br>Protected override void Reduce (IReduceInputEnumerator <int>input) { | [10] | 2 | L3 |

| | | | | |
|---|---|---|---|---|
| | Int sum = 0;<br>while(input.Movenext() {<br>Int value = input.current;<br>sum + = value;<br>} this.Emit(sum);<br>}}} | | | |
| 4 | Provide a historical perspective of the most important technologies supporting data-intensive computing.<br><br>**SOLUTION:**<br>**The Early Age: High-Speed Wide Area Networking :** 1989, high-speed networking as a support of remote visualization of scientific data. Two years later, used high-speed networking for TCP/IP based applications. Kaiser Project: Made available the remote data sources with the help of Wide Area Large Data Object (WALDO). Clipper Project: Collaborative effort of several scientific research laboratories for data-intensive computing.<br><br>**Data Grid:** With the advent of Grid computing, huge computational power and storage facilities could be obtained. Data grids emerge as infrastructures supporting data-intensive computing. It offers high performance and reliable transfer of moving large amounts of data. These data grids have their own data characteristics as follows:<br><br>1. Massive Datasets<br>2. Shared Data Collections<br>3. Unified Namespace<br>4. Access Restrictions<br><br>**Data clouds and Big Data:** Together with the diffusion of cloud computing technologies supporting data-intensive computations, the term “Big Data” has become popular. Big data sizes are a constantly moving target currently ranging from a terabytes to petabytes. Cloud technologies support Big data in several ways. Ex: Google File System (GFS), Hadoop Distributed File System (HDFS).<br><br>**Databases and Data-Intensive Computing:** Distributed databases are a collection of data stored at different sites of a computer network. A distributed database can be created by splitting and scattering the data of an existing database over different sites. These systems are very robust, and resources can be managed efficiently. | [10] | 2 | L2 |
| 5 | a) List and explain six major categories of currently available configuration for EC2 instances.<br>b) Describe how cloud computing technology can be applied to support remote ECG Monitoring.<br><br>**SOLUTION:**<br>**a)**<br>**Standard Instances:** Configuration that are suitable for most applications.<br>**Micro Instances:** These instances are suitable for the applications that consume limited amount of computing power and memory.Small web application with limited traffic.<br>**High-Memory Instances:** These instances are suitable for the applications that need huge workloads and require a large amount of memory. Three tier web applications<br>**High-CPU Instances:** This is applicable for computational intensive applications.<br>**Cluster Compute Instances:** Provides virtual clusters. These clusters have high CPU compute power, Large memory and high I/O and network performance.<br>**Cluster GPU Instances:** These instance are highly applicable for heavy graphic computations. They have high CPU compute power, Large memory and high I/O and network performance.<br><br>**Electrocardiogram (ECG) Analysis:** This produces a waveform that is repeated over time that represents the heartbeat. This is used to identify any kind of heart abnormalities. Cloud computing allows: Remote monitoring of patient’s heartbeat data. Performs Analysis in Minimum time. Notification of First-aid personnel. Risk of a patient can be monitored constantly | [5+5] | 2 | L2 |

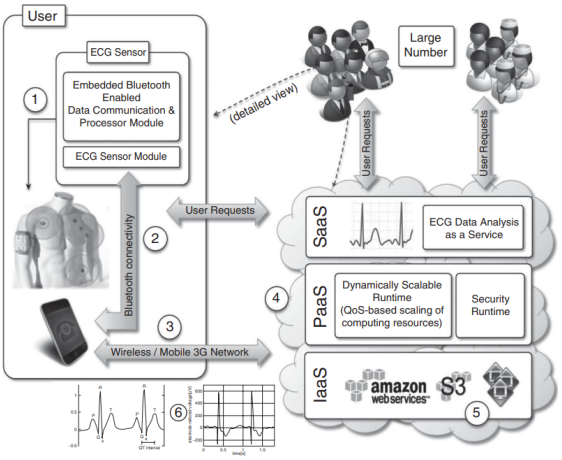


Fig. 10.1. Online Health-Monitoring System Hosted in the Cloud.

**Advantages:**

1. Elasticity of cloud infrastructure.
2. No need to invest more on the infrastructure.
3. Ubiquity
4. Easily accessible using web interfaces (SOAP and RestFul)
5. Delivers services with minimum or no downtime.
6. Pay-as-you-go basis.

| | | | | |
|---|---|---|---|---|
| 6 | Write short notes on the following Cloud Applications:<br>a) Animoto<br>b) Maya Rendering | [5+5] | 2 | L2 |

**SOLUTION:**

**ANIMOTO:** Animoto website provides an interface for creating the videos out of images, music and video fragments. Proprietary AI engine selects the animation and transition effects according to pictures and music. Allows 30 seconds video for free. Core functionality is implemented on the top of Amazon Web Services. Uses Amazon EC2 for worker nodes and S3 for storage of data.

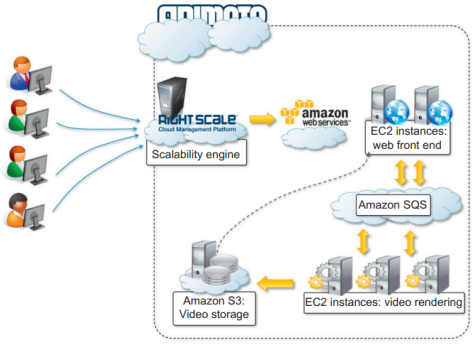


FIGURE 10.8
Animoto reference architecture.

**MAYA RENDERING:** Visualization of mechanical process is not only used at the end of design process but also used to improve the design. GoFront group: A private cloud solution for designing trains. Design process requires high quality 3D images. Includes a specialized interface used by GoFront engineers to enter the details of rendering process. Application is used to submit the rendering tasks to Aneka cloud.

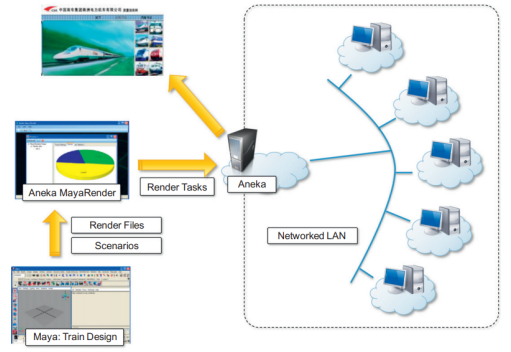


FIGURE 10.9
3D rendering on private clouds.

**CO PO Mapping**

| Course Outcomes | | Modules covered | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 | PSO4 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | Explain cloud computing, virtualization and classification of services of cloud computing | 1,2 | 2 | 2 | 2 | 2 | 0 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 2 | 2 | |
| CO2 | Illustrate architecture and programming examples in cloud | **2,3,4** | 2 | 2 | 0 | 2 | 2 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 2 | |
| CO3 | Describe the platforms for development of cloud applications with examples | **4,5** | 2 | 3 | 3 | 3 | 2 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 2 | |

| COGNITIVE LEVEL | REVISED BLOOMS TAXONOMY KEYWORDS |
|---|---|
| L1 | List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc. |
| L2 | summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend |
| L3 | Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover. |
| L4 | Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer. |
| L5 | Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize. |

| PROGRAM OUTCOMES (PO), PROGRAM SPECIFIC OUTCOMES (PSO) | | | | CORRELATION LEVELS | |
|---|---|---|---|---|---|
| PO1 | Engineering knowledge | PO7 | Environment and sustainability | 0 | No Correlation |
| PO2 | Problem analysis | PO8 | Ethics | 1 | Slight/Low |
| PO3 | Design/development of solutions | PO9 | Individual and team work | 2 | Moderate/ Medium |

| | | | | | |
|---|---|---|---|---|---|
| PO4 | Conduct investigations of complex problems | PO10 | Communication | 3 | Substantial/ High |
| PO5 | Modern tool usage | PO11 | Project management and finance | | |
| PO6 | The Engineer and society | PO12 | Life-long learning | | |
| PSO1 | Develop applications using different stacks of web and programming technologies | | | | |
| PSO2 | Design and develop secure, parallel, distributed, networked, and digital systems | | | | |
| PSO3 | Apply software engineering methods to design, develop, test and manage software systems. | | | | |
| PSO4 | Develop intelligent applications for business and industry | | | | |