SOLUTIONS

Module:	Cloud Computing		
Module Code	EBU750U	Paper	A
Time allowed	2hrs	Filename	Solutions_1920_EBU750U_A
Rubric	ANSWER ALL FOUR QUESTIONS		
Examiners	Dr Gokop Goteng	Dr Atm Alam	

Question 1

a) The Amazon Elastic Block Store (EBS) automatically replicates data within the availability zones of the region in which it is created to provide a highly availability block storage service to users. It takes on an average 10 seconds for an EBS that is attached to an Amazon Elastic Compute Cloud (EC2) node to be restored in case of any failures. On the average, this fault occurs once every 360 days. Use this information to answer the following questions:

[15 marks]

i) Give the mathematical formula for calculating High Availability (HA) for the EBS attached to the EC2 node as described in a).

(3 marks)

ii) Use the formula you have provided in ai) to calculate the High Availability (HA) for the EBS attached to the EC2 node.

(3 marks)

iii) 80% of the AWS EC2 nodes in a) are used for parallel processing and there is a total of 100 virtual Central Processing Units (vCPUs) being used for processing data. Use the **Fixed** workload and **Scaled** work load to calculate the **system efficiencies (fixed and scaled efficiencies)** of the AWS EC2 system.

(9 marks)

	Do not write in this column
Q1a i)	
High Availability (HA) = MTTF/(MTTF + MTTR) [2 marks], where MTTF = Mean Time To Failure and MTTR = Mean Time To Recovery [1 mark]	
Q1a ii)	
MTTF = 360 days = 360x24x60x60 secs = 25,920,000 secs [1 mark], MTTR = 10 secs	
HA = MTTF/(MTTF + MTTR) = 25,920,000 / (25,920,000 + 10) = 25,920,000 / 25,920,010 = 0.9999996141976797 =99.999961420% [2 marks]	
Q1a iii)	
Fixed workload efficiency, E = 1/[αn + 1- α] [2 marks]	
Parallel process, 1-α=80%=0.80, Serial process, α= 20%=0.20 [1 mark], number of processors, n=100	
Fixed workload system efficiency, E=1/[0.20 x 100 + 0.80] = 0.0480769=4.81% [2 marks]	

Scaled workload efficiency, E' = α/n + (1- α) [2 marks]	
Scaled workload system efficiency, E'= 0.20/100 + 0.80 = 0.802= 80.2% [2 marks]	
	15 marks

b) List the FOUR cloud deployment models.

[4 marks]

	Do not write in this column
Q1b	
Public cloud [1 mark]	
Private cloud [1 mark]	
Community cloud [1 mark]	
Hybrid cloud [1 mark]	
	4 marks

c) Describe Data Integrity and Data Confidentiality in computer network.

[4 marks]

	Do not write in

	this column
Q1c	
Data integrity: Is the process of ensuring that the data sent across networks and cloud systems are not tampered with or changed along the way [1 mark] meaning data sent should be the same data that are received. [1 mark]	
Data Confidentiality: This is the process of ensuring that only people and organisations that are supposed to see or use data can see it [1 mark] meaning data should be received by the right persons and not to find its way in the wrong hands [1 mark]	
	4 marks

d) Describe AWS CloudFront.

[2 marks]

	Do not write in this column
Q1d	
Amazon CloudFront is a fast content delivery network (CDN) [1 mark] service that securely delivers data, videos, applications, and APIs to customers globally with low latency and high transfer speeds. [1 mark]	
	2 marks

Question marking: $\frac{}{15} + \frac{}{4} + \frac{}{4} + \frac{}{2} = \frac{}{25}$

Question 2

a) This question is on Graphics Processing Units (GPUs) and Compute Unified Device Architecture (CUDA):

[15 marks]

i) Briefly state the FIVE main steps used in CUDA C programming model to address the heterogeneous nature of GPU programming architectures based on the host and device components.

(5 marks)

ii) Write a simple CUDA C program that will print the message "Welcome to GPU Programming World!". Use myGPUKernel() as the kernel name for your CUDA C program and launch it using the <<< >>> delimiters to print the message "Welcome to GPU Programming World!" FIVE times in parallel (do NOT use any C iteration loop commands).

(10 marks)

	Do not write in this column
Q2a i)	
Declare and allocate host and device memory. [1 mark]	
Initialize host data. [1 mark]	
Transfer data from the host to the device. [1 mark]	
Execute one or more kernels. [1 mark]	
Transfer results from the device to the host. [1 mark]	
Q2a ii)	
__global__ void myGPUKernel() { [2 marks]	
Print ("Welcome to GPU Programming World! \n"); [1 mark]	
} [1 mark]	
int main() { [1 mark1]	
myGPUKernel<<<1,5>>>(); [3 marks]	
return 0; [1 mark]	
} [1 mark]	
	15 marks

b) Describe Amazon CloudWatch and AWS CloudTrail.

[4 marks]

	Do not write in this column
Q2b	
Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS. [1 mark] You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources. [1 mark]	
AWS CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of your AWS account. [1 mark] With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure including APIs calls. [1 mark]	
	4 marks

c) Describe the Model View Controller (MVC) and Front Controller design patterns. Give ONE example for each of these two design patterns where they are used in real-life applications.

[6 marks]

	Do not write in this column
Q2c	
Model View Controller (MVC):	
There is a clear separation and modularity in the codes that implement the model, controller and the view as they are loosely coupled [1 mark] and MVC improves maintainability as changes on any of the components do not require changes in the entire application. [1 mark]	
MVC is used in implementing e-commerce websites. [1 mark]	
Front Controller:	
This provides a centralised enterprise system which handles all tasks from a single point [1 mark] and so when there is problem, it will affect the entire system. [1 mark]	
Front Controller is a good method for secure systems and so it is used in implementing ATMs and Credit/Debit cards systems. [1 mark]	

		6 marks

Question marking: $\frac{}{15} + \frac{}{4} + \frac{}{6} = \frac{}{25}$

Question 3

a) The following pseudocode presents a Map/Reduce program that applies a vision recognition module (**recognizeAnimals**: returns a list of animals that appear in an image) to a large dataset of images collected from camera traps, which is used to monitor biodiversity and population density of animal species.

```
Map (String imageName, ImageData data) {
    String[] animals = recognizeAnimals(data);
    for(String animal: animals) {
        emit(animal, imageName)
    }
}
```

```
Reduce (String animal, List<String> images){
    emit(animal, images)
}
```

What will be the outcome of the Map/Reduce job? Explain it and illustrate the exchange of information between Mappers/Reducers.

[7 marks]

Q3a:

The job will emit the list of animals found in the dataset by the vision recognition program, and for each of them, the list of images that contain that animal [3 marks]. It is an inverted index algorithm. [1 marks].

Mappers will emit one key per found animal in an image and value the id of the image [1 mark]. All the images containing the same animal will be collected as a single input for a Reducer [2 marks]

b) Hadoop job execution involves many computation tasks. Indicate with **arrows** (→), which daemons (right part of **Figure 1**) are responsibilities for the Hadoop computation tasks (left part of **Figure 1**).

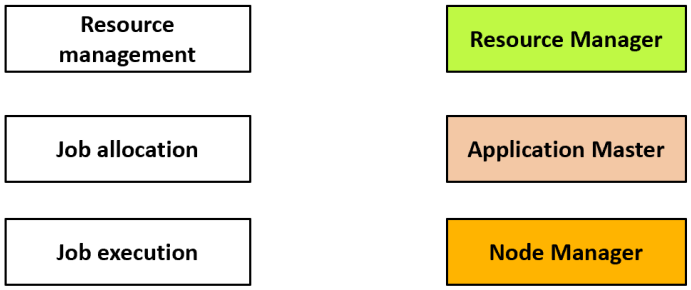
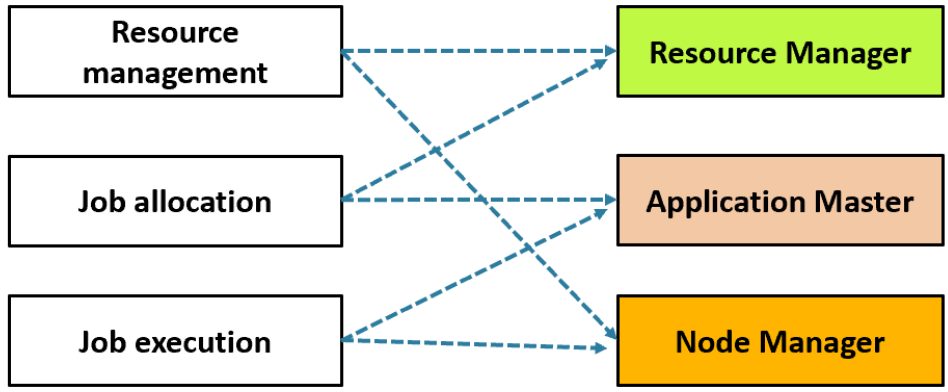


Figure 1

Note that you can answer to this question either below or on the diagram above.

[3 marks]

Q3b:



[3 marks = 0.5 mark for each correct connection. Also, -0.5 mark for each wrong connect until 0 mark]

c) This question relates to data filtering in Map-Reduce patterns.

[5 marks]

i) What is the goal of data filtering in the context of Map-Reduce job? Give an example of data filtering.

(3 marks)

ii) Why is the data filtering "Mapper only job"?

(2 marks)

Q3c:

(i) The goal of the data filtering pattern is to filter out records/fields that are not of interest for further computation. [2 marks]

- *Distributed grep (text search).*
- *Tracking a thread of events (logs from the same user)*
- *Data cleansing*

[1 mark for any reasonable data filter example]

(ii) No need to aggregate because only removing data with data filtering pattern. [2 marks]

d) This question is about **performance** in Map/Reduce.

[10 marks]

i) Describe **Amdahl's Law**, and the distinction between sequential and parallel parts of computation. Name one stage in Hadoop that must be performed sequentially.

[3 marks]

ii) If **95%** of a computational job must be performed **sequentially**, what is the maximum speedup achievable when running this job across 8 processors?

Again, for the same job, what is the maximum speedup achievable when running this job across 1000 processors? Justify your answer.

You should use **Amdahl's Law** to answer this question. Show the working out.

(7 marks)

Q3d:

(i) Signals that only parts of a computation can be parallelised [2 Marks]

*Sequential examples include **loading data**, **pre-processing** and **data splitting*** [1 Mark for any]

(ii) Speedup (8 processors) = 1/(0.95 + (1-0.95)/8) = 1.046 [2 Marks]

Speedup (1000 processors) = 1/(0.95 + (1-0.95)/1000) = 1.053 [2 Marks]

Overall, the speedup is tiny [1 Mark]*. There is only little improvement in speedup or more/less the same* [1 Mark] *even if the number of processors has been increased significantly, which is because only small fraction (5%) of the total job can be executed in parallel* [1 Mark].

Question 3 marking: $\frac{}{7} + \frac{}{3} + \frac{}{5} + \frac{}{10} = \frac{}{25}$

Question 4

a) This question is about **Big Data** platforms beyond Map/Reduce

[10 marks]

i) What is in-memory processing? Discuss main performance limitations of Hadoop Map/Reduce when compared to modern in-memory processing systems such as Apache Spark. Illustrate the difference with an example.

(4 marks)

ii) What is a **resilient distributed dataset (RDD)** in the context of Apache Spark? Explain two types of RDD operations with an example for each operation, i.e., how RDDs are created and modified via programmatic operations.

(6 Mark)

Q4a:

(i) Data is already loaded in memory before starting computation.

***Or** Processing data stored in memory*

***Or** In-memory processing defines as, instead of storing data in some slow disk drives, the data is kept in random access memory(RAM) to be processed in parallel.*
[1 mark for the definition]

Map/Reduce reads and writes data to distributed storage after every Map/Reduce stage [1 mark].

This becomes very inefficient in any computation that needs to iterate over data or hold state, whereas in-memory processing systems allow to reuse these distributed data structures once they are loaded in memory. [1 mark]

For any loop-style algorithm, Map/Reduce will incur on extra overhead for HDFS read, write, Map/Reduce shuffle network transfer, as well as job setup, whereas a platform such as Spark can implement a loop on the same data structure. [1 mark]

(ii) RDDs are immutable [1 mark] collections distributed data across the nodes of the cluster, and it can be

- *Can be **rebuilt** if a partition is lost [0.5 mark]*
- *Can be **cached** across parallel operations [0.5 mark]*

RDDs are operated upon with functional programming constructs and Two types of operations can be applied to RDDs:

- ***Transformations***
  - ***Transformations** are applied to an existing RDD to create a new RDD **Or** Lazy operations to build RDDs from other RDDs [1 mark]*
  - *For example, applying a filter operation on an RDD to generate a smaller RDD of filtered values. Or (e.g. map, filter, groupBy, join) [1 mark for any example]*
- ***Actions***
  - *Actions are operations that actually return a result back to the Spark driver program—resulting in a coordination or aggregation of all partitions in an RDD. **Or** Return a result or write it to storage. [1 mark]*
  - *For example, count, collect, save, etc. [1 mark for any example]*

b) What is DNS caching in Content Delivery Networks? Mention two benefits of the DNS caching.

[4 marks]

Q4b:

Once (any) name server/resolver learns mapping, it caches mapping the information on domain name/IP address mappings.

***Or** A DNS name server can store DNS query results for a period of time and this is called DNS caching.*

[2 marks]

Benefits of DNS caching:

Reduces the burden on the root servers
Reduces DNS traffic across the Internet (i.e., reduces overhead)
Increases performance in Internet applications

[2 marks for any two]

c) The following questions relate to **Cloud Databases**:

[11 marks]

i) Explain the following techniques that are used to achieve data partitioning and replication in the context of cloud databases: **Memory Caches**, **Separating Reads from Writes**, **High Availability Clustering**, and **Data Sharding**.

(8 Marks)

ii) The SQL databases provide **strong consistency** and **availability** at the expense of **partition tolerance** while different NoSQL databases make different **CAP-based trade-offs**. What trade-offs does **Amazon Dynamo Systems** make?

(3 marks)

Q4c:

*(i) **Memory Caches** can be seen as transient, partly partitioned and replicated in-memory databases, since they replicate most frequently requested parts of a database to main memories of a number of servers* [2 mark]

- *Fast response to clients*
- *Off-loading database servers*

***Separating reads from writes**:*
One or more servers are dedicated to writes (master(s)) [0.5 mark]*, and a number of replica servers are dedicated to satisfy reads (slaves)* [0.5 mark]*. The master replicates the updates to slaves* [1 mark]*.*

- *If the master crashes before completing replication to at least one slave, the write operation is lost.*
- *Otherwise the most up to date slave undertakes the role of a master.*

High-availability (HA) clusters *are groups of computers that support server applications and the HA clusters operate by harnessing redundant computers in groups to provide a continued service when system components fail* [1 mark].

- *When a HA cluster detects a hardware/software fault, it immediately restarts the application on another system without requiring administrative intervention (**failover**)* [0.5 mark]
- *HA clusters use redundancy to **eliminate single points of failure** (SPOF)* [0.5 mark]

Data Sharding *is data partitioning in such a way that:*

- *Data typically requested and updated together reside on the same node* [1 mark]*, and*
- *The workload and storage volume are roughly evenly distributed among servers* [1 mark].

*(ii) Dynamo systems provide **availability*** [1 mark] *and **partition tolerance*** [1 mark] *at the expense of **strong consistency*** [1 mark]**.**

Question 4 marking: $\frac{}{10} + \frac{}{4} + \frac{}{11} = \frac{}{25}$