# COMP Cloud Computing Sample Exam

## Part A

### Question I (30 marks)

1. **What is a parallel computing system?**

A parallel computing system is one with two or more processors, usually within a single system, in which several tasks can be completed at the same time, in parallel. A problem or task must be easily parallelizable to make it worth the effort to run on such a system. Can be split by task (operation) or data (chunks) and distributed to be run in parallel.

*Parallel Design:*

* 1. Partition – break the problem to split in parallel
  2. Communication – Co-ordinate tasks
  3. Agglomeration - group tasks
  4. Map – assigned to processor
  5. Join - for final result

1. **What is a distributed computer system?**

Any computing that involves multiple computers which are remote from each other but share a role in solving a computational problem or information processing. Involving two or more systems, they do not have to have the same hardware. The problem set does not easily parallelizable or even in size, since in a distributed system the load can be unevenly distributed depending on a number of factors (such as to machines with many more gpu’s vs cpu’s). Reliability is not assumed, since the cluster of machines may vary in size or availability. Usually a message passing system is used to co-ordinate the distributed computer system.

*Example - Folding@Home*

Folding@Home is a distributed computing project that simulates protein folding. It uses idle processing time of thousands of machines with very diverse hardware for this purpose. Users install a small piece of software which detects and makes available idle CPU cycles to the Folding@Home network.

1. **How do they compare?**

|  |  |  |
| --- | --- | --- |
| Characteristic | Parallel | Distributed |
| Overall Goal | Speed | Convenience |
| Interactions | Frequent | Infrequent |
| Granularity | Fine | Coarse |
| Reliability | Assumed | Not Assumed |

### Question II (30 marks)

1. **Give 3 examples of IaaS services**

Infrastructure as a Service (IaaS) is a lower level of services in the cloud. With IaaS you can rent: CPU, Storage Space, Server Space and Network Equipment. You can choose to run your own applications on hardware provided or you can install all your own hardware (also then responsible for upgrades).

*Example 1: Your own data centre*  
In this scenario, you just purely use the space provided by the IaaS provider to install all your own systems. Power will be provided, network can be, but may be up to you to arrange peering etc. A large amount of capital can be required.

*Example 2: Dedicated Virtual Servers*

In this scenario, you rent dedicated servers. The IaaS provider then provides all network and power capability, as well as server hardware upgrades. Can usually sign contract regarding hardware failure turnaround times. You then install your own software stack from the kernel (Linux, Windows) up to your application software.

*Example 3: Share Virtual Server*

In this scenario you rent a portion of a physical server from the IaaS provider. This is usually in the form of a virtual machine, or dedicated area of a SAN (Storage Area Network). The software stack is decided by you from a virtualized kernel up to your application. This is the cheapest of the IaaS examples, requires very little upfront capital.

1. **What are the differences between AWS SimpleDB and AWS S3?**

SimpleDB is a database product from Amazon, while S3 is a storage product.

SimpleDB is more like traditional SQL databases because it can be queried fast, but it is designed to store pre-sorted text, numbers and can (but shouldn’t really be used for) storing small binaries (in byte format usually). SimpleDB also has limits for storage based on the domain and the key.

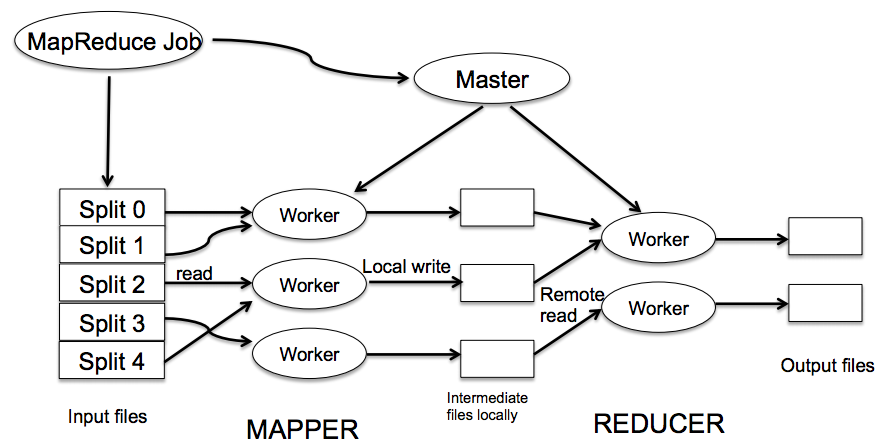
S3 is designed to store files, and can store large files. While S3 stores some metadata with each object (file), to query S3 properly you would have to retrieve every object in the bucket to see if your query is correct. This can get very expensive and time consuming with large buckets.

If you need large files with fast query, you can store the metadata in SimpleDB with a key to get the item from S3, for an efficient hybrid approach.

1. **What are the main security concerns in cloud computing?**

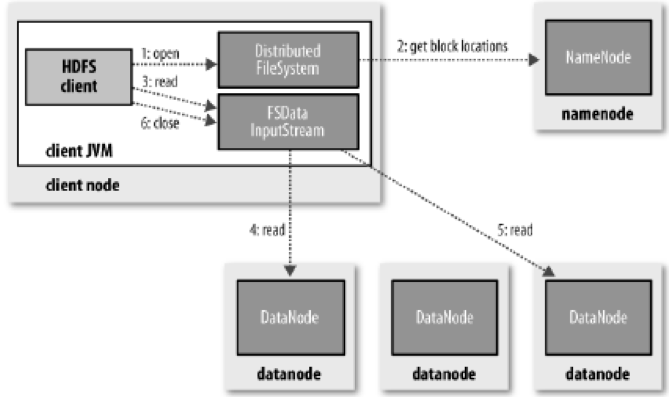
|  |  |
| --- | --- |
| Vulnerability | Examples |
| Shared Technology Vulnerabilities | Gaining access through hypervisor, escaping virtual machine/sandbox etc |
| Insecure interface and APIs | Allows hijacking of resources or accounts (like leaking of Amazon private key) |
| Abuse of cloud services | Hogging CPU, RAM, Launching DDOS and Spam attacks etc |
| Data Loss and Leakage | Faulty backups, incomplete backups, storing other customers data beside yours so both are retrievable |
| Data Breaches | Insecure data storage, database vulnerability (no saneitized SQL lookups etc) |
| Denial of Service | Unable to access critical services, loos of business, loss of trust |
| Malicious Insider | Can be paid to leak vital data to competitors |

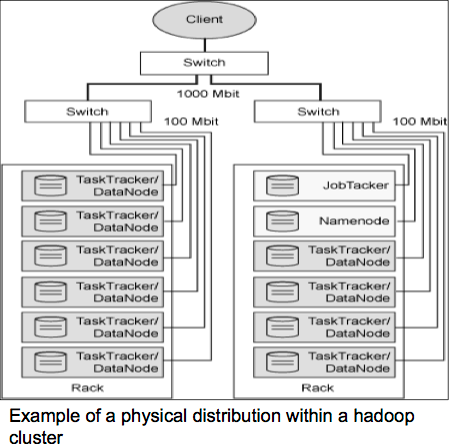
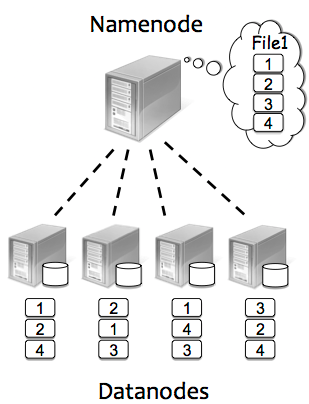
### Question III (10 marks)

1. **What is MapReduce? Describe it’s model**MapReduce is a data parallel programming model designed for high scalability and resiliency.  
     
   A **map** function processes a key/value pair to generate a set of intermediate key/value pairs.  
     
   The **reduce** function merges all intermediate values associated with the same intermediate key.  
     
   *The system takes care of:*1. Partitioning the data (Split)  
   2. Scheduling the data and execution (on Workers)  
   3. Node failures, replication and re-submission  
   4. Co-ordination among nodes  
     
   ****

1. **Describe the general architecture of HDFS**

Hadoop File System (HDFS) is a highly scalable and fault-tolerant file system for Hadoop. You have two main types of node: NameNode and DataNode. Commodity hardware can be used, since the architecture is managed in the software stack. When storing data, metadata is stored on a NameNode and the actual data is stored across two or more DataNodes. This ensures fault-tolerance and reliability. When retrieving data, the NameNode is queried to find out on which DataNodes the data should be retrieved on. This is suitable for very large files.



## Part B

**Answer any one of the following:**

### Question IVa (30 marks):

**Write a pig script that counts the number of occurrences of each word in a text file input.txt**

//load file as lines

lines = LOAD 'input.txt' AS line;

//Tokenize words

words = FOREACH lines GENERATE FLATTEN(TOKENIZE(line)) AS word;

// Convert to lowercase

small\_words = foreach words generate LOWER(word) AS word;

// filter to only words by regex

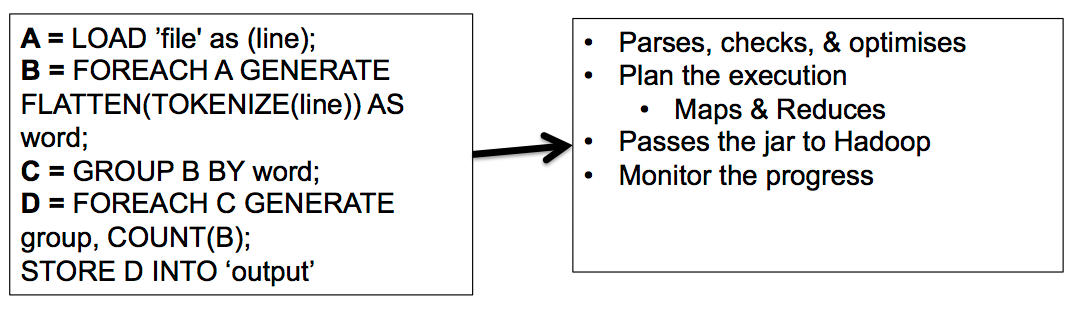
filtered\_words = FILTER small\_words BY word MATCHES '\\w+';

// Group words that are the same

grouped\_words = GROUP filtered\_words BY word;

// Count the words

word\_count = FOREACH grouped\_words GENERATE group, COUNT(filtered\_words) AS count;



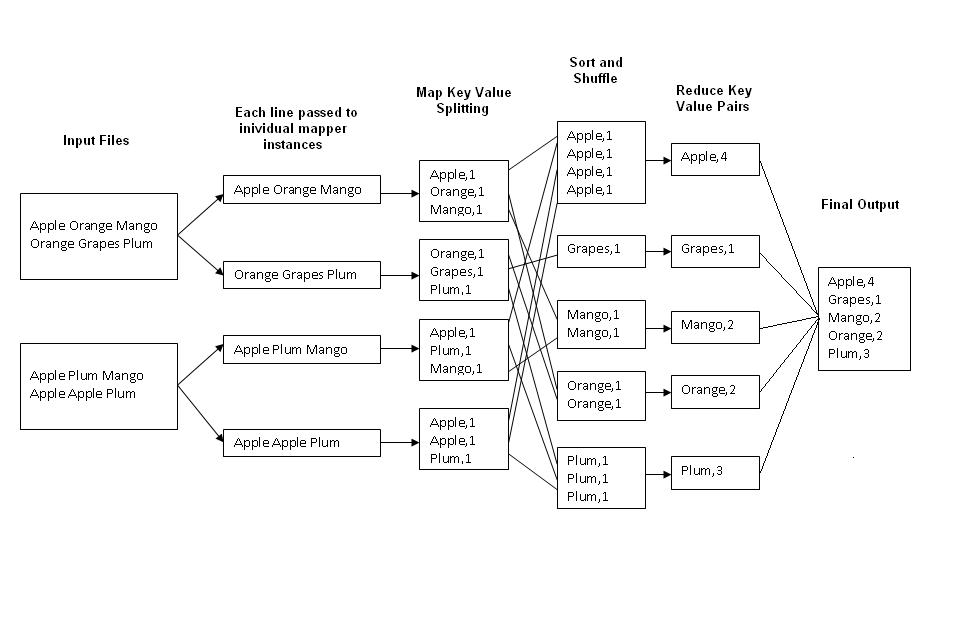
### Question IVb (30 marks):

**Let’s Consider the problem of counting occurrences of each word in a text file:**

1. **Define input and output of the Map and Reduce functions**

[**https://developer.yahoo.com/hadoop/tutorial/module4.html**](https://developer.yahoo.com/hadoop/tutorial/module4.html)

[**http://kickstarthadoop.blogspot.ie/2011/04/word-count-hadoop-map-reduce-example.html**](http://kickstarthadoop.blogspot.ie/2011/04/word-count-hadoop-map-reduce-example.html)



#### AN EXAMPLE APPLICATION: WORD COUNT

A simple MapReduce program can be written to determine how many times different words appear in a set of files. For example, if we had the files:

**foo.txt:** Sweet, this is the foo file

**bar.txt:** This is the bar file

We would expect the output to be:

sweet 1

this 2

is 2

the 2

foo 1

bar 1

1. **Write Map and Reduce function**

**mapper** (filename, file-contents):

**for each** word **in** file-contents:

**emit** (word, 1)

**reducer** (word, values):

sum = 0

**for each** value **in** values:

sum = sum + value

**emit** (word, sum)