Question 1. Short Answer Questions (25 points)

1. [4 points] Assume an erasure coding scheme LRC(12,2,2) . The data unit is divided into 12 fragments: a local parity fragment **px** is created from the first 6 fragments **x0 - x5** and another local parity fragment **py** is created from the next 6 fragments: **y0-y5**; two global parity fragments **p0** and **p1** are created from the entire 12 data fragments. Show an example failure pattern with four failed fragments and analyze the reconstruction cost. The failure pattern should not come from the original paper.
2. [6 points] Many cloud database services adopted the layered approach by separating the database layer and the storage layer. Identify THREE benefits of such approach, for each benefit, use a real world system as example to demonstrate the benefit.
3. [4 points] Use your own words, explain the difference between chunk version number and mutation serial number in GFS.
4. [5 points] Explain "shuffle" and how it happens in MapReduce and Spark framework respectively.
5. [6 points] Explain the following terms: "application", "job", "task" and "task attempt" in YARN. Some terms are shared by MapReduce and Spark framework while others may be only used by a single framework. If a shared term has different meanings in different frameworks, it should be highlighted.

Question 2. Virtualization and Containerization (20 points)

Assume you want to set up two isolated Jupyter notebook environments and you are going to use container technology to implement that. The images you will use are: **`jupyter/r-notebook`** and **`jupyter/pyspark-notebook`**. You have designated directory on host file system for each container and would like to save notebooks developed inside containers on those directories. You should make assumptions on the actual directory you want to use for each container. Assume all required images have been pulled and stored locally. The basic command to run a container with bind mount option is as follows:

```
docker run --rm -p 10000:8888 -v "$PWD":/home/jovyan/work jupyter/r-
notebook
```

You can use the `--name` option to give a name to your container. You can also replace the `"$PWD"` option with an absolute path in your file system such as `"/home/xyz123/c1"`.

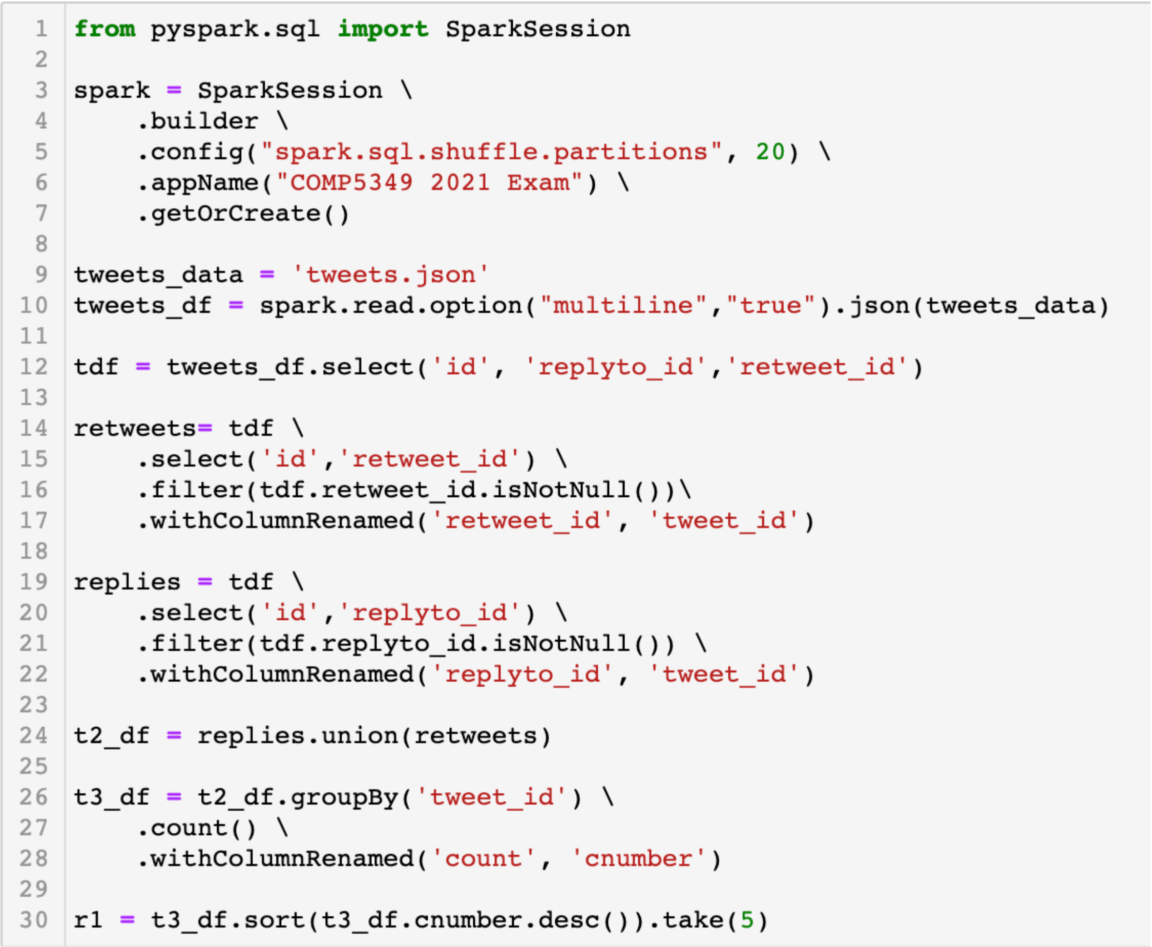
1. [2 points] Show the command to run a container based on **`jupyter/r-notebook`**, you should give the container a name and bind mounts a local directory into the container.
2. [2 points] Describe how you can access the jupyter notebook server running on this container.
3. [2 points] Show the command to list all processes running in this container.
4. [4 points] Assume the first container is running, show the command to run a second container based on **`jupyter/pyspark-notebook`**. You should give the second container a different name and bind mounts a different local directory into the container.
5. [2 points] Describe how you can access the jupyter notebook server on the second container.
6. [2 points] Assume you create a new folder called `"project"` under `"/home/jovyan"` and develope a few notebooks in this folder inside this container, where can you locate those notebooks on your local file system?
7. [6 points] Describe how you may achieve the isolated environment using virtualization techniques and highlight the difference between the two techniques using your own words.

Question 3. Spark Programming and Distributed Execution (**25 points**)

This question has several parts. All parts are related with the following PySpark application `app.py`. The application is submitted to a 5-node EMR cluster consists of one master node and four work nodes. Each worker node has 16G memory that can be used by YARN. Each node has 4 vCPUs. The program uses the same tweets data you have used in assignment 2. The data set contain many tweet objects. Each tweet object has many fields; only the following three fields will be used in this question:

- **id**: the unique id of the tweet. This field appears in all objects in the data set
- **retweet_id**: the **id** of the tweet it re-posts. This field appears in some object in the data set
- **replyto_id**: the **id** of the tweet it replies to. This field appears in some object in the data set

The size of the input file **`tweets.json`** is around 6MB.



```
from pyspark.sql import SparkSession

spark = SparkSession \
    .builder \
    .config("spark.sql.shuffle.partitions", 20) \
    .appName("COMP5349 2021 Exam") \
    .getOrCreate()

tweets_data = 'tweets.json'
tweets_df = spark.read.option("multiline","true").json(tweets_data)

tdf = tweets_df.select('id', 'replyto_id','retweet_id')

retweets= tdf \
    .select('id','retweet_id') \
    .filter(tdf.retweet_id.isNotNull())\
    .withColumnRenamed('retweet_id', 'tweet_id')

replies = tdf \
    .select('id','replyto_id') \
    .filter(tdf.replyto_id.isNotNull()) \
    .withColumnRenamed('replyto_id', 'tweet_id')

t2_df = replies.union(retweets)

t3_df = t2_df.groupBy('tweet_id') \
    .count() \
    .withColumnRenamed('count', 'cnumber')

r1 = t3_df.sort(t3_df.cnumber.desc()).take(5)
```

1. [3 points] How many times the input file will be scanned when executing this application? Describe possible improvement to avoid multiple scan and re-computation.
2. [5 points] Identify all variables referring to a DataFrame or an RDD between line 14 and line 30. Describe the record/element structure of each DataFrame or RDD. For DataFrames with the same structure, you only need to describe the structure once.
3. [9 points] Assume The default resource configuration for Spark application is:

   Drive memory: 1G;
   Application Master Memory: 2G
   Executor Memory: 8G;
   Executor Core: 4

   The submit script is:
   ```
    spark-submit \
            --master yarn \
            --deploy-mode cluster \
            --num-executors 3 \
            app.py \
   ```

   Describe the process YARN uses to allocate resources for this application. Executing this application will generate a number of tasks; each task needs to be allocated to an executor. Show also a possible task allocation plan.

4. [8 points] The given application produces top 5 results for tweets having retweets and/or replies. Assume we are only interested in tweets having both retweets and replies, you are asked to implement a workload using **PySpark API** to produce similar top 5 results ONLY for those tweets. You may reuse code in the given program. In doing so, you need to indicate the lines that you will reuse. You are encouraged to design more efficient operation sequence to produce the top 5 results.

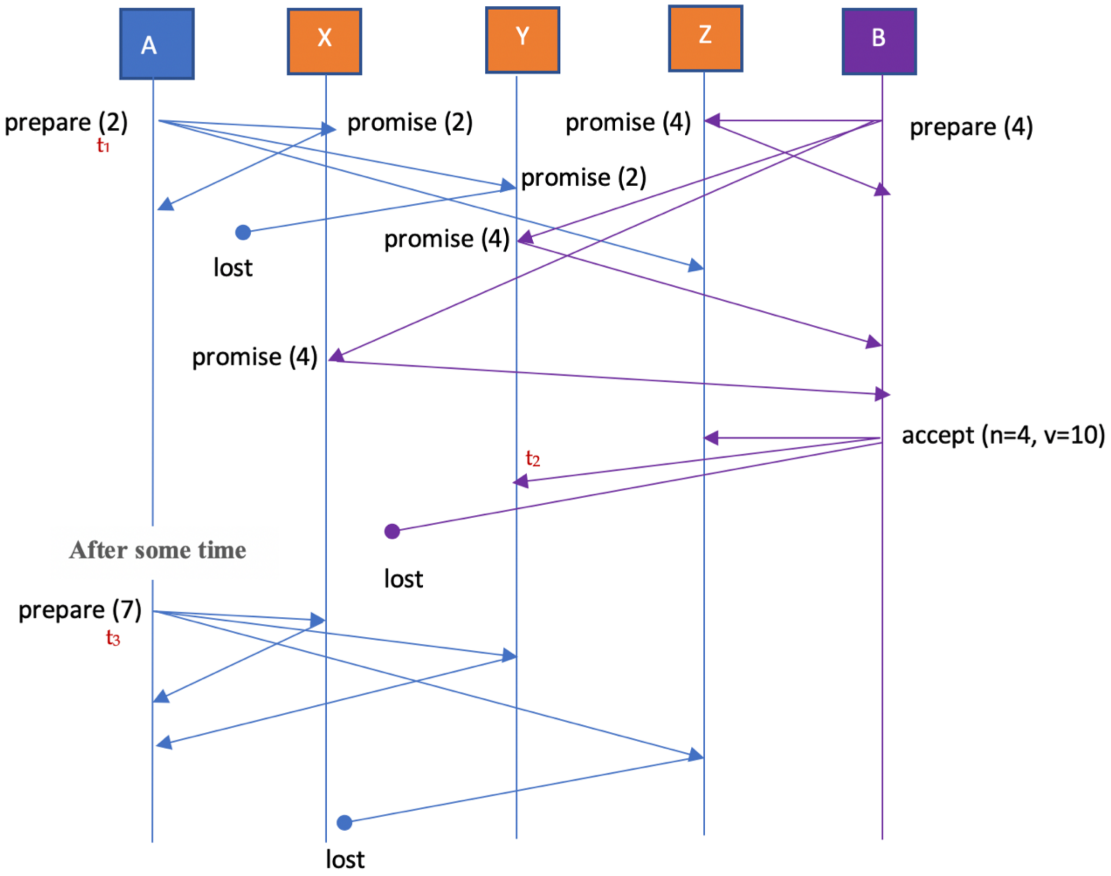
Question 4. Distributed Data Consistency (20 points)

Part **1- 4** of this question refer to the following message sequence scenario of a Paxos instance with five agents: two proposers A and B; three acceptors X, Y and Z. Assume time proceeds from top to bottom and there is no chosen value at the beginning of the scenario. Both A and B try to propose a value following Paxos algorithm.

Proposer A has two proposals, with sequence number 2 and 7 respectively. Proposer B only has one proposal with sequence number 4. A message with arrow end indicates it is correctly sent. A message with dot end indicates a lost message, aiming for the nearest agent in the message direction. There are three lost messages in the scenario: the `promise(2)` message sent from acceptor Y to proposer A; the `accept(n=4,v=10)` message sent from proposer B to acceptor X; the response message from acceptor Z to proposer A's `prepare(7)` message.

Note that not all messages are shown in the scenario. Some are left for questions.

For part 4 and part 5, a textual description is preferred; if you want to use diagram, please draw the diagram using a drawing software. Hand-drawn diagram will NOT be marked.



1. [**2 points**] Following Paxos algorithm, will proposer **A** proceed with the second phase of proposal 2 by sending accept message to available acceptors? If yes, what will be the content of the message and which acceptor(s) it will send the message to; if no, explain the reason.
2. **[3 points]** What are the response message of each acceptor to proposer **A**'s `prepare(7)` message?
3. **[3 points]** Proposer **A** wants to propose value 5. Now assume the response message for message `prepare(7)` from acceptor **Z** is lost, will proposer **A** proceed with the second phase by sending accept message to available acceptors? If yes, what will be the content of the message and which acceptor(s) it will send the message to; if no explain the reason.
4. [**4 points]** Assume proposer A sends `prepare(7)` message before t_2, describe a scenario that will end up with value 5 from proper A to be chosen in proposal 7.
5. [**8 points**] This question is related with a Chubby lock service consists of five nodes: A, B, C, D, E. A Chubby service can only have one master at any time. A master has a lease that lasts for a few seconds. Once the lease expires, the system will elect a new master with a new lease. The master's identity and the expire time of the lease are selected(chosen) using Paxos algorithm as a tuple value: **(master, expire_time**). All nodes in the cluster take all three roles: *proposer*, *acceptor* and *learner* in any Paxos process in the system. This means any node can make a proposal, accept a proposal or try to learn the chosen value. Assume that at **t1**, all nodes know the current master is A and its lease expires at **t5**. This is achieved through an accepted proposal: **(n=5, v=(A, t5))**. The proposal number is **5**, and chosen value is **(A,t5)**. Now assuming at **t2**, which is before **t5**, node C loses contact with node A and believes that A is dead. C plans to make a proposal to elect itself as the new master, with a lease expire time set to **t7**. Assume the communication between B and C, D, E are normal and there will be no message loss.

   Describe the process C will use to elect itself. You description should include all messages with sender node, receiver node and detailed content. Indicate the sequence of messages as well.

Question 5. Integral question (10 points)

This unit covers many distributed systems, such as GFS, Bigtable, Dynamo, Azure storage, Auroral, YARN and Kubernetes, all systems provide some fault tolerance mechanisms. Use your own words to describe and compare 2 fault tolerance mechanisms used in two difference systems covered in this unit. Your description should contain enough detail on each fault tolerance mechanism. When comparing two mechanisms, you should clearly identify which aspect(s)/feature(s) of the faulty tolerance mechanism you are comparing. In particular, you should focus on aspect(s)/feature(s) supported in both mechanisms.