Homework 1

Name	ID	Class
Qing Liu	6130116184	Class 164 in Computer Science and Technology

Edit with VSCode and convert to PDF by Chrome.

Q1: Characterize the following three cloud computing models.

- ### What is an IaaS cloud? Give one example system.

  IaaS, infrastructure as a service, which is defined in WIKIPEDIA:

  > Infrastructure as a service (IaaS), are online services that provide high-level APIs used to dereference various low-level details of underlying network infrastructure like physical computing resources, location, data partitioning, scaling, security, backup etc.

  Like a virtual machine, we only use it, regardless of how it is implemented. Many online virtual machines were provided to use for many people, and we just use it like programming with APIs, so it become the service and we need to pay for it. **Amazon EC2** is an example. We need to install the operating system and software by ourself. It only provides us with hardware we can not see and touch, we need to maintain the OS and software environment.

- ### What is an PaaS cloud? Give one example system.

  PaaS, platform as a service, which is defined in WIKIPEDIA:

  > Platform as a Service (PaaS) or Application Platform as a Service (aPaaS) or platform-based service is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app.

  According to the difinition, PaaS has already help us build the platform or environment for development. It not only provides the online hardware but also the computing platform and solution services. Developers only need to focus on their business logic. **Google App Engine** is a example for PaaS.

- ### What is an SaaS cloud? Give one example system.

  SaaS, software as a service, which is defined in WIKIPEDIA:

> Software as a service (SaaS) is a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted. It is sometimes referred to as "on-demand software", and was formerly referred to as "software plus services" by Microsoft. SaaS is typically accessed by users using a thin client, e.g. via a web browser.

IaaS is sometimes referred to as "on demand software", which is a software delivery model. In this delivery mode, cloud centralized hosting software and related data can be used only through the Internet, not through installation. Users usually use a streamlined client to access software as a service via a web browser. **Office 365** is an typical example. It's different from Office Home Edition and we used to call it as a service.

Q2: Basic Cloud Calculus

- ## Given a cloud system ... for this cloud system?

Because A = 98%, so I were left only 2% to down. So the answer is:
RMT = 0.02 x 24 x 30 (hours) = 14.4 (hours).

- ## Consider a cloud cluster of ... function of k and p.

If k is 1, A = 3 x p x (1-p) x (1-p)
if k is 2, A = 3 x p^2 x (1 - p)
if k is 3, A = p ^ 3
So, the A = C(3, k) x p^k x (1 - p)^(3 - k)

- ## Given that each server ... answer this question correctly.

A = C(n, k) x 0.985^k x 0.015^(n - k) > 0.95
A = n! / (k! x (n - k)!) x 0.985^k x 0.015^(n - k) > 0.95
  - When k is 1, A = n x 0.985 x 0.015^(n - 1)
    - n = 1, A = 0.985;
    - n = 2, A < 0.95
  - When k is 2, n(n - 1) / 2 x 0.985^2 x 0.015^(n - 2)
    - n = 1, A = 0;
    - n = 2, A = 0.97 > 0.95;
    - n = 3, A = 0.437 < 0.95
  - When k is 3, (n^3 - 3 x n^2 + 2n) / 6 x 0.985^3 x 0.015^(n - 3)
    - n = 1, A = 0;
    - n = 2, A = 0;
    - n = 3, A = 0.9556 > 0.95
    - n = 4, A = 0.0573 < 0.95

If k <= 3, the largest number of server is 3, the minimum is 1;

Q3: Write after Read

- ## What

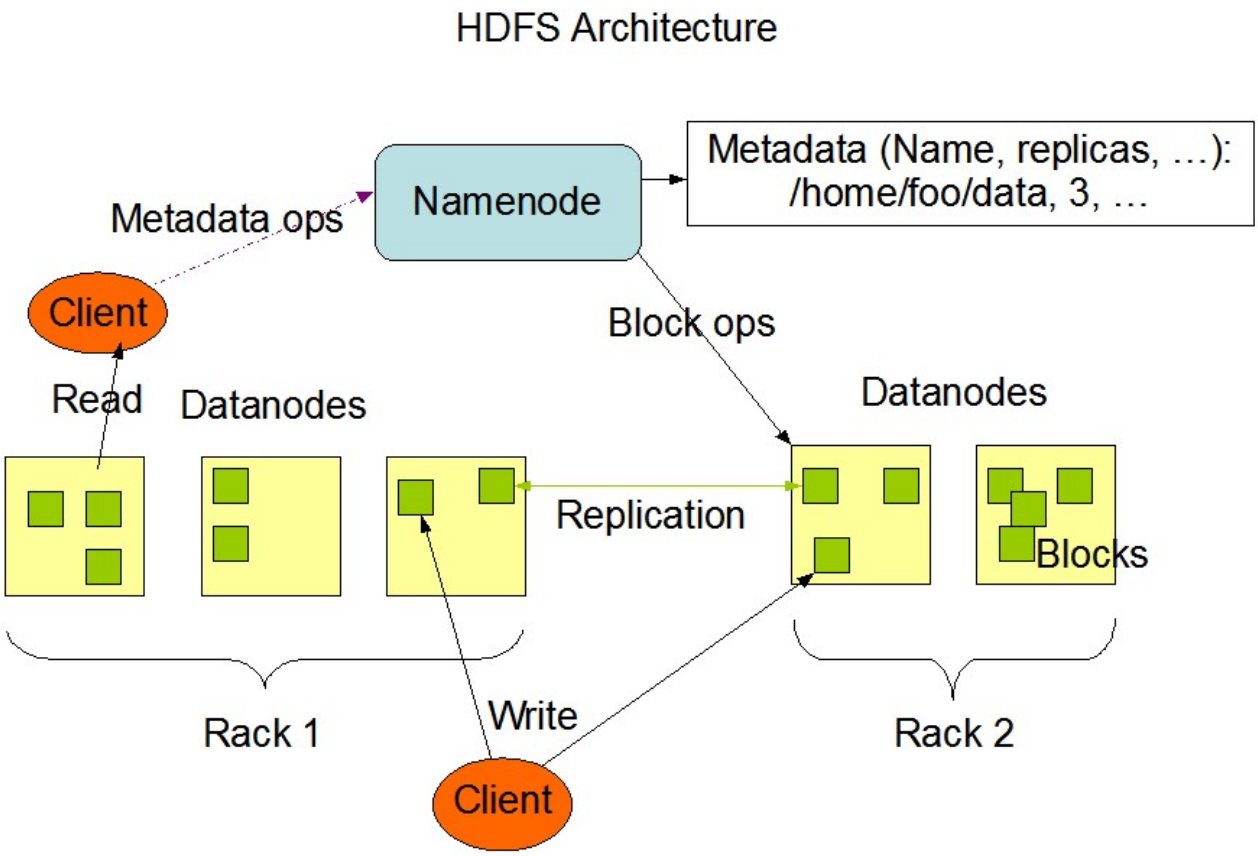
  This is an overview of **Hadoop File System**, which includes the motivation, arthitecture, design, inplementation and performance benchmarks.

- ## My Questions

  - ### What's the relationship between MapRedece, Hadoop, GFS and HDFS?

    MapReduce is a programming model which was developed by Google, and **GFS(Google File System)** is a patent of Google. Hadoop is consist of **HDFS(Hadoop Distrubuted File System)** and **MapReduce**, and it's open-source. **GFS** and **HDFS** are file system based on commodity servers, which are supported large distributed data-intensive applications.

  - ### What's the architecture of HDFS?



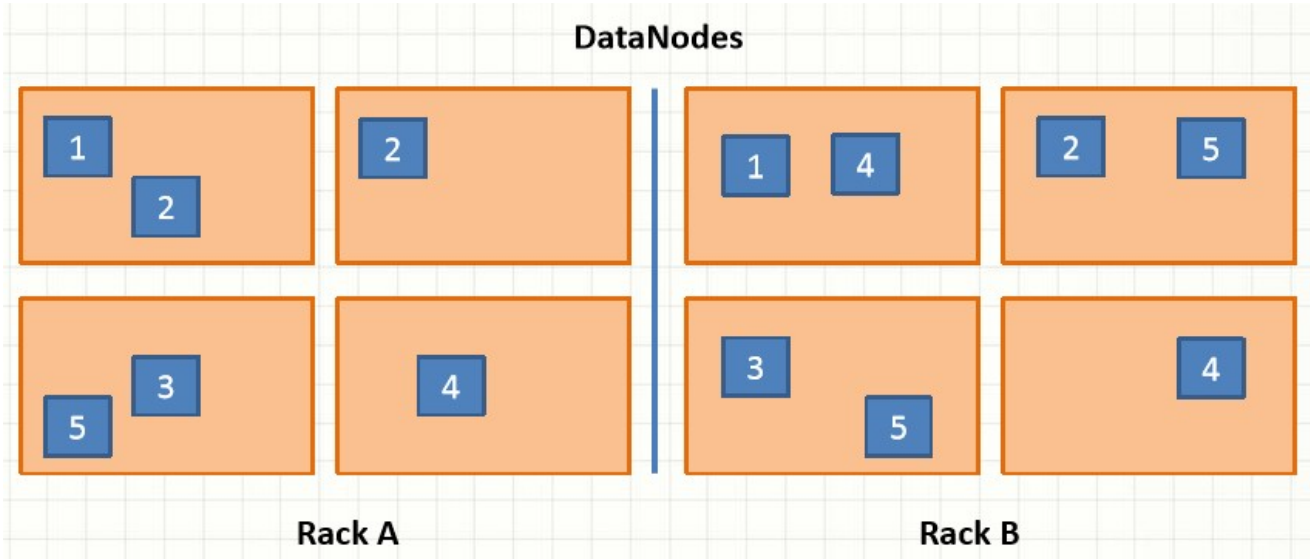
There are 3 components - namenode, datanode and client in the architecture.
It's a typical master/slave architecture. Namenode is a central server that manages the namespace of the file system and client access to files. A Datanode in a cluster is usually one node, responsible for managing its storage on all nodes. HDFS exposes the namespace of the file system and enables users to store data in the form of files. Internally,

a file is actually divided into one or more data blocks, which are stored on a set of DataNodes. Namenode performs namespace operations on file systems, such as opening, closing, renaming files or directories. It is also responsible for determining the mapping of data blocks to specific Datanode nodes. Datanode handles read and write requests from file system clients. Create, delete and replicate data blocks under the unified scheduling of Namenode.

- **Why is HDFS reliable and how to achieve fault tolerance?**

  HDFS is designed to reliably store large files across machines in a large cluster. It stores each file as a series of data blocks, all of which are the same size except the last one. For fault tolerance, all data blocks of the file will have copies. The data block size and copy coefficients of each file are configurable. The application can specify the number of copies of a file. Copy coefficients can be specified at the time of file creation or can be changed later. The files in HDFS are written at one time, and there is a strict requirement that only one writer should be available at any time.

- **As for the Block Placement, why the default strategy ensures that no Datanode contains more than one replica of any block and no rack contains more than two replicas of the same block?**



I got some information in https://hadoop.apache.org/docs/r1.0.4/cn/hdfs_design.html.
The storage strategy of HDFS is to store one copy on the node of the local rack, one copy on the other node of the same rack, and the last copy on the node of different racks. This strategy reduces data transmission between racks, which improves the efficiency of write operation. Rack errors are far fewer than node errors, so this strategy will not affect the reliability and availability of data. At the same time, this strategy reduces the total network transmission bandwidth required to read data because the data blocks are only placed on two (not three) different racks. In this strategy, replicas are not evenly distributed on different racks. One-third of the copies are on one node, two-thirds are on one rack, and the rest are

evenly distributed on the remaining racks. This strategy improves write performance without compromising data reliability and read performance.

- # Problems in HDFS Design

To be honest, if I were to design this filesystem, I even don't know where to start. I don't know much about this field at present. So as for the problems in HDFS, I search for them in Google. And I found out an blog in JueJin and another blog in CSDN.

HDFS can store data of TB or even PB size on the premise that the data should be large files first, and NameNode memory should be large enough.NameNode is HDFS that stores metadata information of the entire cluster, such as all file and directory information. Moreover, when metadata information is abundant, NameNode will start slowly and trigger GC operations more easily. Obviously, when the data reaches a certain level, metadata management will become a bottleneck of HDFS, which is why it is suitable for storing large files. If we solve the problem of metadata management, in fact, HDFS can support a large number of small files.

The solution is Ozone, an object storage service implemented by Hortonworks based on HDFS, which was called HDDS too. It aims at data Node storage based on HDFS, supports larger data object storage, supports various object sizes and has the reliability, consistency and availability of HDFS.

At present, HDFS extensibility mainly focuses on the following two points:

1. Heavy Block Task
2. Huge memory namespace storage

For the first point, HDDS provides a container-based block service. The second point is the storage of metadata. HDFS currently stores all metadata in memory. Although this query responds quickly, with the increase of metadata, the memory consumption of NameNode nodes is amazing. HDDS introduces a third-party more efficient K-V database for metadata storage, which currently supports RocksDB and LevelDB.