1. Spiegare quelli che sono i componenti principali di Hadoop

   Hadoop è costituito da due componenti principali: un file di archivio e un sistema di elaborazione distribuita HDFS (Hadoop Distributed File System) quest'ultimo consente di memorizzare i file attraverso un insieme di server in cluster. HDFS consente di elaborare direttamente i data-nodes senza dover trasferire i dati al sistema computazionale.
   • LETTURA: Per effettuare una scrittura il client HDFS contatta il namenode, che restituisce i vari datanode che possiedono una copia dei primi blocchi. Le operazioni di lettura sono effettuate direttamente sui datanode interessati ai blocchi, interrogandoli singolarmente. I datanode vengono forniti ordinati in base alla distanza dal client: se il client è eseguito su un nodo del cluster che già possiede una copia del blocco, leggerà la sua copia locale. Man mano che la lettura del file procede, il client richiede al namenode la locazione dei blocchi successivi.
   • SCRITTURA: Per effettuare una scrittura, un client HDFS comunica la creazione di un file al namenode, il quale ne controlla la non-esistenza ed i permessi di accesso. Se la scrittura viene autorizzata, il namenode crea un record sui metadati per il file. Il namenode alloca quindi i blocchi, fornendo al client una lista di possibili datanode su cui posizionarli; la taglia di questa lista è pari al livello di replica desiderato.

2. Spiegare cos'è il "meccanismo" di MapReduce

   E' un sistema di elaborazione parallela di dati, include una componente software chiamata job scheduler, responsabile della scelta dei server che eseguono il lavoro di ogni singolo utente. Il client Hadoop fornisce il job e le configurazioni al JobTracker il quale si occupa di distribuirli ai vari nodi per l'esecuzione. Il JobTracker determina il numero di parti in cui l'input deve essere distribuito e attiva alcuni TaskTracker in base alla loro vicinanza ai nodi che contengono i dati di interesse. I TaskTracker estraggono poi i dati dalla parte di loro competenza e attivano la funzione map che produce coppie chiave/valore. Una volta terminata la fase di map, i TaskTracker notificano al JobTracker il completamento del loro lavoro. Il JobTracker può cosi attivare la fase di reduce, nella quale i TaskTracker ordinano i risultati dei mapper per chiave, li aggregano ed eseguono la funzione reduce al fine di produrre l'output, salvato in un file diverso per ciascun TaskTracker. Un job MapReduce è costituito da 4 componenti:
   - I dati di input, su HDFS;
   - Una funzione map, che trasforma i dati di input in una serie di coppie chiave/valore;
   - Una funzione reduce che, per ogni chiave, elabora i valori ad essa associati e crea, come output, una o più coppie chiave valore. L'esecuzione della funzione reduce è preceduta da una fase di raccolta delle coppie chiave/valore prodotte dalla funzione map. Le coppie sono ordinate per chiave e i valori con la stessa chiave sono raggruppati;
   - L'output, scritto su un file HDFS.

3. Cos'è Pig Latin?

   Pig è un linguaggio di scripting per accedere velocemente a dataset estesi; infatti, consente di scrivere poche dozzine di righe direttamente da console, per avere subito i risultati attesi. Possiede numerosi comandi sofisticati, già pronti per risolvere la maggior parte delle comuni esigenze. Come per MapReduce, i dati sono elaborati via batch. I suoi casi d'uso classici sono i seguenti:
   - Data Sampling: riduzione di un ampio dataset, per esplorare una porzione rilevante. Possibilità di analizzare il subset, con strumenti che non scalano bene.
   - Analisi di Web Log: estrazione di informazioni utili da file di log e da Web Server.

- Elaborazione ETL: estrazione, Trasformazione e Caricamento dati in un DWH.

4. Immaginando di avere un file con il seguente contenuto

   Dear, Bear, River, Car, Car, River, Deer, Car ,Bear

   mostrare in modo concettuale come andrebbe a lavorare MapReduce.
   (NON FATE CODICE VOGLIO SOLAMENTE SAPERE QUALI SONO I PASSAGGI E COME VENGONO FATTI IN PSEUDOCODICE)

   Map:

   Dear:1, Bear:1, River:1, Car:1, Car:1, River:1 , Deer:1, Car:1 ,Bear:1

   Shuffle & Sort:

   - Dear:1
   - Bear:1,1
   - River:1,1
   - Car:1,1,1
   - Deer:1

   Reduce:

   - Dear:1
   - Bear :2
   - River:2
   - Car:3
   - Deer:1

5. Cos'è ZLIB?

   ZLIB è una libreria open source per compressione e decompressione file, senza rischiare di avere delle perdite. Ha 10 livelli di compressione da 0 a 9, dove 0 non prevede nessuna compressione,1 veloce ma basso livello di compressione,9 molto letto ma livello di compressione elevato. Il livello di default è 6 .

6. Definizione e utilizzo del K-MEANS

Il **k-means clustering** è tra i più semplici algoritmi non supervisionati di machine learning, utile per la segmentazione del dataset. L'obiettivo è raggruppare **data points** simili, dividendo il dataset in un numero *k* di **clusters**. Un **cluster** è quindi un insieme di osservazioni che condividono caratteristiche simili.

Il **k-means** è un partiotioning algorithm, questo significa che il **k-means** divide il dataset in numero *k* di *clusters* **non sovrapposti** e **indipendenti**, privi di strutture interne o labels, tali per cui le **osservazioni** di un cluster siano **simili** tra loro e **dissimili** da quelle presenti nei restanti insiemi.

come funziona?

La **similarità** tra i samples è usata per dare forma ai cluster, facendo in modo che osservazioni simili finiscano nello stesso insieme. Il **k-means** tenta quindi di massimizzare la distanza **inter-cluster** tra i samples e minimizzare quella **intra-cluster**.

le formule per il calcolo della distanza che è possibile usare sono:

- Cosine similarity
- Euclidean Distance
- Average Distance
- Minkowski Distance

Per scegliere quale misura sia più corretta, è fondamentale prendere in esame parametri quali il **domain knowledge** del dataset, e la tipologia di attributi che lo compongono.

Il funzionamento operativo dell'algoritmo si basa sul concetto di **centroide**, centro di ogni cluster, e richiede che venga specificato a priori il numero *k* di **clusters**. Esistono due approcci all'inizializzazione dei centroidi. Il primo prevede la scelta casuale di un numero pari a *k* di samples, usati come partenza. Il secondo individua sempre un numero pari a *k* di punti, questa volta scelti però in modo completamente casuale, e non appartenenti al dataset.

A ogni iterazione, è calcolata la distanza di ciascun sample dai **centroidi**. In questo processo definiamo errore la distanza totale, intesa come sommatoria, di ciascun sample dal centroide (è fondamentale minimizzare l'errore). Prima dell'iterazione successiva, ogni centroide è spostato nel punto medio delle distanze calcolate. Il ciclo ricomincia e continua fintantoché la posizione dei centroidi non si stabilizza, cioè rimanga la medesima da un'iterazione alla successiva.

Le prestazioni dell'algoritmo sono influenzate dalla scelta iniziale dei **centroidi**. È consuetudine testare diverse varianti dell'algoritmo, con condizioni di partenza ovviamente differenti. L'algoritmo è veloce, quindi non ci sono grosse problematiche.

7. Quali sono i file utilizzabili in Hive?

   E' stato creato per mettere a disposizione degli utilizzatori di HBase un'interfaccia **SQL like** orientata ai dati e può agire o da shell o da software applicativo, ma non trasforma il cluster Hadoop in un RDBMS, Il cluster non è un Database Server. È importante sottolineare come Hive non lavori solamente con Hadoop ma può compiere tutta una serie di operazioni come creare tabelle locali.