PYTHON ASSIGNMENT 3

1.
When you load data from an external source, you load it into a suspense table. You can then review the data in the suspense table and modify it. To load data into the suspense table, position the source file or tape, specify the location of the source, and run the appropriate load external data process.
2.
Reading JSON Files using Pandas
To read the files, we use read_json() function and through it, we pass the path to the JSON file we want to read. Once we do that, it returns a “DataFrame”(A table of rows and columns) that stores data. If we want to read a file that is located on remote servers then we pass the link to its location instead of a local path.
3.
If you have larger-than-memory data, you can use Dask to scale up your workflow to leverage all the cores of your local workstation, or even scale out to the cloud.

This article will discuss:

How Dask can help parallelize your data science computations,
Behind-the-scenes workings of Dask with schedulers and workers, and
Dask’s diagnostics dashboard and resources for scaling to the cloud.

A prominent feature of Dask is its familiar API. As Matthew Rocklin, the creator of Dask wrote in A Brief History of Dask:

“One pain point we heard time and time again was that people worked with data that fit comfortably on disk but that was too big for RAM, and accelerating NumPy was a common feature request to Continuum/Anaconda at the time. To this end, the purpose of Dask was originally to parallelize NumPy so that it could harness one full workstation computer, which was common in finance shops at the time. There were two technical goals, and a social goal:
4.
We used the dask.delayed function to wrap the function calls that we want to turn into tasks. None of the inc, double, add, or sum calls have happened yet. Instead, the object total is a Delayed result that contains a task graph of the entire computation. Looking at the graph we see clear opportunities for parallel execution. The Dask schedulers will exploit this parallelism, generally improving performance (although not in this example, because these functions are already very small and fast.
5.
Features of Cassandra
Apache Cassandra is an open source, user-available, distributed, NoSQL DBMS which is designed to handle large amounts of data across many servers. It provides zero point of failure. Cassandra offers massive support for clusters spanning multiple datacentres.

There are some massive features of Cassandra. Here are some of the features described below:

Distributed:

Each node in the cluster has has same role. There’s no question of failure & the data set is distributed across the cluster but one issue is there that is the master isn’t present in each node to support request for service.