DSC 204A: Scalable Data Systems

PA0 Discussion Session: Setting up AWS and Dask

Rohit Ramaprasad

Agenda

1. Fundamentals of Dask

2. Demo
   a. Setting up AWS
   b. Demonstration of some common Dask functions
   c. Some tips!

Dask: Overview

- Parallel computing framework that scales existing Python frameworks.

np.zeros((10000,10000,10000)) -> OOM!
dask.array.zeros((10000,10000,10000))) -> SUCCESS!

- Breaks up work into tasks and executes them in task parallel manner.
- Dask provides APIs to create task graphs (DAG).
- Dask provides a scheduler that runs this DAG by assigning tasks to workers.

Dask: APIs

Low-level APIs:

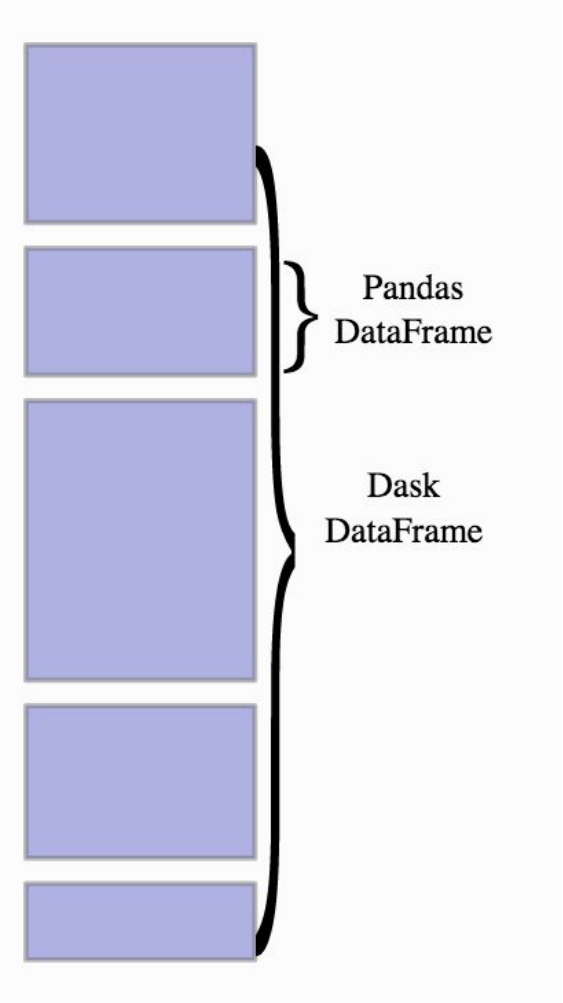
- Dask Delayed (Parallel lazy objects)
- Dask Futures (Parallel eager objects)

High-level APIs:

- Dask Array (Parallel NumPy)
- Dask DataFrame (Parallel Pandas)
- Dask Bag (Parallel Dictionary)
- Dask ML (Parallel Scikit-Learn)

DataFrame APIs enough for this assignment, feel free to check out other APIs if needed.

Dask: DataFrame API



How Dask Represents A DataFrame -

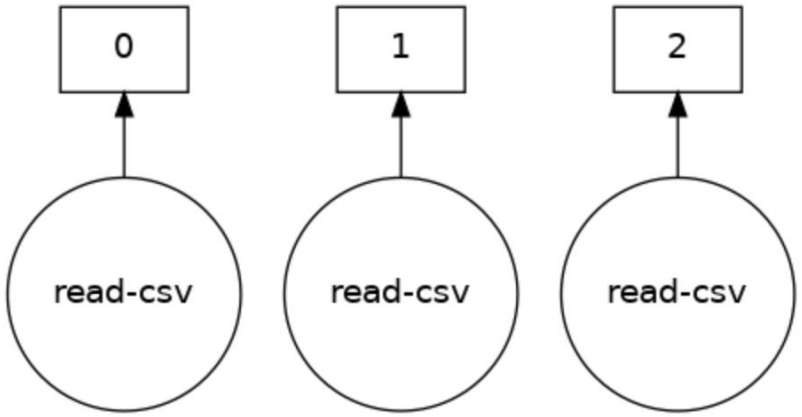
- Dask DF consists of multiple smaller Pandas DFs
- Partition size can be manually set or left to Dask to decide based on the available memory and number of cores on the machine, up to a max size of 64MB.
- Operation on Dask DF triggers operations on smaller Pandas DFs. Partitions are loaded onto RAM to perform computations and later memory is released for remaining partitions to be loaded.
- Similar APIs as Pandas DF APIs!

Dask: DataFrame API

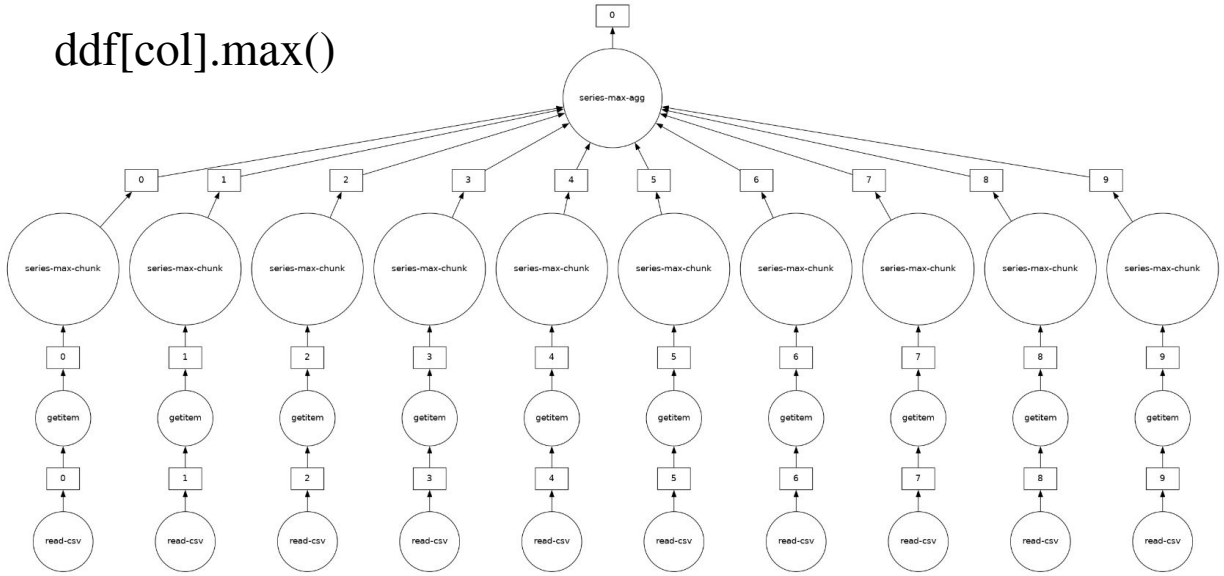
How Computations Are Executed ?

- Dask operations are evaluated **lazily:**
  - Not evaluated immediately.
  - Constructs a graph of smaller tasks on each partition needed to go from input to output.
  - Evaluation starts only when calling .compute() on the dask dataframe.

dd.read_csv(“my_huge_file”.csv)



ddf[col].max()



AWS and Dask Demo

Assignment: Dataset Description

Column name	Column description	Example
reviewerID	ID of the reviewer	A32DT10X9WS4D0
asin	ID of the product	B003VX9DJM
reviewerName	name of the reviewer	Slade
helpful	helpfulness rating of the review	[0, 0]
reviewText	text of the review	this was a gift for my friend who loves touch lamps.
overall	rating of the product	1
summary	summary of the review	broken piece
unixReviewTime	summary of the review	1397174400
reviewTime	time of the review (raw)	04 11, 2014

Table 1: Schema of Reviews table

Assignment Tasks: PA 0

Create a new *users* dataframe using only the *reviews* table with the following schema:

Column name	Column description
reviewerID (PRIMARY KEY)	ID of the reviewer
number_products_rated	Total number of products rated by the reviewer
avg_ratings	Average rating given by the reviewer across all the reviewed products
reviewing_since	The year in which the user gave their first review
helpful_votes	Total number of helpful votes received for the users' reviews
total_votes	Total number of votes received for the users' reviews

Table 2: Schema of users table

Grading Scheme

1. Accuracy (80 points)
    - 5 columns
    - 16 points per column
    - Error tolerance 1%

2. Runtime (20 points)

Absolute single node runtimes	**Points**
Under 20 mins	20
Between 20 mins and 30 mins	12
Between 30 mins and 1 hr	8
Anything above 1 hr	0

Table 1: Grading Scheme

Some tips!

- Some helpful dask APIs
  - groupby
  - map_partitions
  - apply

- While performing groupby aggregations with large no. of groups (millions or more), use **split_out** to split the output into multiple partitions to avoid worker error (see this).

- Do not call compute() many times.

- Try to avoid writing custom aggregation functions. They are typically very slow. Use built-in aggregation methods like sum(), mean() etc.

Infra Management

- Use private GitHub repo if possible for versioning code. Or simply download your notebook to your local machine at regular intervals.

- Terminate the AWS instance every time after usage to save budget. Launch new Spot instance and read from S3 again next time.

Some Helpful Links

- https://saturncloud.io/docs/troubleshooting/package-support/dask/dask_groupby_aggregations/
- https://distributed.dask.org/en/latest/memory.html.
- https://distributed.dask.org/en/latest/manage-computation.html
- https://docs.dask.org/en/latest/dataframe-best-practices.html