

Activity 22.2: Using DASK to Read and Analyze Multiple Files in Parallel (60:00)

Start Assignment

Due Nov 13 by 11:29am **Points** 20 **Submitting** a file upload
Available after Nov 6 at 8am

☒ (<https://classroom.emeritus.org/courses/9296/modules/items/1556599>)

☒ (<https://classroom.emeritus.org/courses/9296/modules/items/1556601>)

☒ (<https://classroom.emeritus.org/courses/9296/modules/items/1556602>)

☒ (<https://classroom.emeritus.org/courses/9296/modules/items/1556603>)

☐  (<https://classroom.emeritus.org/courses/9296/modules/items/1556604>)

☒ (<https://classroom.emeritus.org/courses/9296/modules/items/1556606>)

☒ (<https://classroom.emeritus.org/courses/9296/modules/items/1556607>)

☒ (<https://classroom.emeritus.org/courses/9296/modules/items/1556608>)

☒ (<https://classroom.emeritus.org/courses/9296/modules/items/1556609>)

☐  (<https://classroom.emeritus.org/courses/9296/modules/items/1556610>)

☒ (<https://classroom.emeritus.org/courses/9296/modules/items/1556611>)








☐  (<https://classroom.emeritus.org/courses/9296/modules/items/1556612>)

☐ (<https://classroom.emeritus.org/courses/9296/modules/items/1556613>)

 ☐  (<https://classroom.emeritus.org/courses/9296/modules/items/1556614>)

Live Support

[s://classroom.emeritus.org/courses/9296/modules/items/1556615](https://classroom.emeritus.org/courses/9296/modules/items/1556615)

- ☐ (<https://classroom.emeritus.org/courses/9296/modules/items/1556616>)
- ☐  (<https://classroom.emeritus.org/courses/9296/modules/items/1556617>)
- ☐  (<https://classroom.emeritus.org/courses/9296/modules/items/1556618>)
- ☐  (<https://classroom.emeritus.org/courses/9296/modules/items/1556619>)
- ☐ (<https://classroom.emeritus.org/courses/9296/modules/items/1556621>)
- ☐ (<https://classroom.emeritus.org/courses/9296/modules/items/1556622>)
- ☐  (<https://classroom.emeritus.org/courses/9296/modules/items/1556623>)
- ☐ (<https://classroom.emeritus.org/courses/9296/modules/items/1556624>)
- ☐  (<https://classroom.emeritus.org/courses/9296/modules/items/1556625>)
- ☐ (<https://classroom.emeritus.org/courses/9296/modules/items/1556626>)
- ☐ (<https://classroom.emeritus.org/courses/9296/modules/items/1556628>)
- ☐ (<https://classroom.emeritus.org/courses/9296/modules/items/1556629>)
- ☐  (<https://classroom.emeritus.org/courses/9296/modules/items/1556630>)
- ☐  (<https://classroom.emeritus.org/courses/9296/modules/items/1556631>)
- ☐  (<https://classroom.emeritus.org/courses/9296/modules/items/1556632>)
- ☐ (<https://classroom.emeritus.org/courses/9296/modules/items/1556633>)
- ☐  (<https://classroom.emeritus.org/courses/9296/modules/items/1556634>)
- ☐ (<https://classroom.emeritus.org/courses/9296/modules/items/1556636>)

Learning Outcome Addressed:

2. Run parallel operations in DASK.

In this activity, you will practice reading multiple files in parallel using DASK. A common task for data engineers is reading and processing large numbers of files. These files can come from many sources. Some files may contain data that needs to be preprocessed and then added to a database. In other cases, large files may need to be scanned and transformed before being sent to another application. Regardless of the application, reading and analyzing files is a time-intensive and thus relatively slow task. In order to speed up this process, it is desirable to read files in parallel. DASK provides a simple and easy to implement solution for parallel processing and parallel read operations.

Prior to beginning this activity, be sure that you have watched and understood the lessons in [Video 22.4 \(https://classroom.emeritus.org/courses/9296/pages/using-dask-to-create-multiple-files-in-parallel-04-10\)](https://classroom.emeritus.org/courses/9296/pages/using-dask-to-create-multiple-files-in-parallel-04-10) and [Video 22.5 \(https://classroom.emeritus.org/courses/9296/pages/using-dask-to-read-and-analyze-multiple-files-in-parallel-05-44\)](https://classroom.emeritus.org/courses/9296/pages/using-dask-to-read-and-analyze-multiple-files-in-parallel-05-44). Also, ensure that you have completed [Activity 22.1 \(https://classroom.emeritus.org/courses/9296/assignments/218533\)](https://classroom.emeritus.org/courses/9296/assignments/218533) as a prerequisite to this activity.

In [Mini-Lesson 22.4 \(https://classroom.emeritus.org/courses/9296/pages/mini-lesson-22-dot-4-dask-30-00\)](https://classroom.emeritus.org/courses/9296/pages/mini-lesson-22-dot-4-dask-30-00), you should have already installed DASK on your device. Reference this mini-lesson to ensure that you have successfully installed DASK before you begin this activity.

To complete this activity, follow these steps:

1. First, create a folder titled `Activity_22.2`. Download the [GenerateFilesWithDask.py \(https://classroom.emeritus.org/courses/9296/files/2493745/download\)](https://classroom.emeritus.org/courses/9296/files/2493745/download) file to the `Activity_22.2` folder. Provide a screenshot to show the `GenerateFilesWithDask.py` file in the `Activity_22.2` folder.
2. Run the `GenerateFilesWithDask.py` file. This should create a `/data` folder with some large files. Provide a screenshot to show that the `/data` folder has been successfully created.
3. Navigate out of the `/data` folder, within the `Activity_22.2` folder, and create a new Python file called `Activity22-2.py`. Provide a screenshot to show the `Activity22-2.py` file is in the `Activity_22.2` folder.
4. In the `Activity22-2.py` file, import the necessary DASK *libraries* using the command below:

```
import dask.dataframe as ddf
from dask import delayed
```

Provide a screenshot to show that the correct DASK *libraries* have been imported into the `Activity22-2.py` file.

5. Use a wildcard to read all of the files generated in the `/data` folder. A wildcard is a designated symbol or character which helps pattern match specific words. In this case, the `*` symbol directs the CSV *reader* to grab all files as long as the beginning starts with “data/2000” and ends with “.csv”. Add the following command to read all of the CSV files:

```
df = ddf.read_csv("data/2000*.csv")
```

Then, display the data that you just read into the DASK *dataframe* using the commands below:

```
df.compute()
print(df.head())
```

Run the `GenerateFilesWithDask.py` Python file. Provide a screenshot to show the head of the DASK *dataframe* and display that the DASK *dataframe* correctly displays the first five rows.

6. Next, process the data by calculating and displaying the mean of the `x` column using the code below:

```
mean = df['x'].mean().compute()
print(f'mean: {mean}')
```

After you have entered the above code, run the file. Provide a screenshot of your Terminal window to show the output after you have *printed* the computed mean of the *dataframe*.

7. Compute the number of columns in the *dataframe* using the code below:

```
cols = len(df.columns)
print(f'columns: {cols}')
```

After you have entered the code above, run the `GenerateFilesWithDask.py` Python file again. Provide a screenshot of your Terminal window to show the number of columns in the *dataframe*.

8. Compute the number of rows in the *dataframe* using the code below:

```
rows = len(df.index)
print(f'rows:{rows}')
```

After you have entered the code above, run the `GenerateFilesWithDask.py` Python file again. Provide a screenshot of your Terminal window to show the number of rows in the *dataframe*.

Submission Instructions:

Your submission for this activity should be a Word document that includes the following screenshots, each labeled for the step that the screenshot represents:

1. Provide a screenshot to show the `GenerateFilesWithDask.py` file in the `Activity_22.2` folder.
2. Provide a screenshot to show that the `/data` folder has been successfully created.
3. Provide a screenshot to show the `Activity22-2.py` file is in the `Activity_22.2` folder.
4. Provide a screenshot to show that the correct DASK *libraries* have been imported into the `Activity22-2.py` file.
5. Provide a screenshot to show the head of the DASK *dataframe* and display that the DASK *dataframe* correctly displays the first five rows.
6. Provide a screenshot of your Terminal window to show the output after you have *printed* the computed mean of the *dataframe*.
7. Provide a screenshot of your Terminal window to show the number of columns in the *dataframe*.
8. Provide a screenshot of your Terminal window to show the number of rows in the *dataframe*.

Additional Details:

- **Estimated time:** 60 minutes
- *This is a required activity and counts toward course completion.*

Activity 22.2

Criteria	Ratings		Pts
1. Provide a screenshot to show the GenerateFilesWithDask.py file in the Activity_22.2 folder.	2 pts Complete The correct screenshot has been included in the submission.	0 pts Incomplete The screenshot has not been included in the submission or is the incorrect screenshot.	2 pts
2. Provide a screenshot to show that the /data folder has been successfully created.	2 pts Complete The correct screenshot has been included in the submission.	0 pts Incomplete The screenshot has not been included in the submission or is the incorrect screenshot.	2 pts
3. Provide a screenshot to show the Activity22-2.py file in the Activity_22.2 folder.	2 pts Complete The correct screenshot has been included in the submission.	0 pts Incomplete The screenshot has not been included in the submission or is the incorrect screenshot.	2 pts
4. Provide a screenshot to show that the correct DASK libraries have been imported into the Activity22-2.py file.	2 pts Complete The correct screenshot has been included in the submission.	0 pts Incomplete The screenshot has not been included in the submission or is the incorrect screenshot.	2 pts
5. Provide a screenshot to show the head of the DASK dataframe and display that the DASK dataframe correctly displays the first five rows.	3 pts Complete The correct screenshot has been included in the submission.	0 pts Incomplete The screenshot has not been included in the submission or is the incorrect screenshot.	3 pts
6. Provide a screenshot of your Terminal window to show the output after you have printed the computed mean of the dataframe.	3 pts Complete	0 pts Incomplete	3 pts

Criteria	Ratings		Pts
	The correct screenshot has been included in the submission.	The screenshot has not been included in the submission or is the incorrect screenshot.	
7. Provide a screenshot of your Terminal window to show the number of columns in the dataframe.	3 pts Complete The correct screenshot has been included in the submission.	0 pts Incomplete The screenshot has not been included in the submission or is the incorrect screenshot.	3 pts
8. Provide a screenshot of your Terminal window to show the number of rows in the dataframe.	3 pts Complete The correct screenshot has been included in the submission.	0 pts Incomplete The screenshot has not been included in the submission or is the incorrect screenshot.	3 pts
Total Points: 20			

Top Questions



It's all empty here!

If you have any questions ask one

