

Activity 22.3: Simulating Parallel Processing (45:00)

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>)

☐ (<https://classroom.emeritus.org/courses/9296/modules/items/1556615>)

☐ (<https://classroom.emeritus.org/courses/9296/modules/items/1556616>)

Live Support <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 simulate running parallel operations in DASK across multiple machines.

To complete this activity, follow these steps:

1. In a Terminal window, run the command below to install the DASK *library* locally:

```
pip install dask
```

Provide a screenshot to show that you successfully installed DASK.

2. Depending on what operating system you are using, open the Anaconda Prompt (for Windows) or the Terminal (for Mac) and run the `dask-scheduler`. Provide a screenshot to show that you successfully ran the `dask-scheduler`.

3. Take note of the address of the for the `dask-scheduler`. Open two additional Anaconda command prompts and start two `dask-worker` processes. Pass the address of the `dask-scheduler` to each of the `dask-worker` processes when you run them. Provide two screenshots to show that you successfully started both of the `dask-worker` processes by passing the correct address to the `dask-scheduler`.
4. Create a Jupyter Notebook to simulate a *client* program that has complex computation. Add the following code to your Jupyter Notebook to import the required DASK *libraries*:

```
import dask.array as da
from dask.distributed import Client
```

Provide a screenshot to show that you created a Jupyter Notebook and successfully imported the DASK *libraries*.

5. Complete the following code to create a 50,000 by 50,000 matrix of random numbers in DASK. Compute the mean and assign the value to the `y` variable:

```
x = da.random.random((50000, 50000))
y = da.exp(x).????
```

Provide a screenshot to show that you successfully executed the command to create a matrix, compute the mean, and assign the value to the `y` variable.

6. Now create a DASK `client` that passes the address of your `dask-scheduler`. Call the `compute` *function* on the `y` variable to instruct DASK to execute the command:

```
client = Client("????????")
y.??
```

Provide a screenshot to show that you successfully computed your calculations using DASK using the `compute` *function*.

You have completed this activity and practiced simulating running complex operations across multiple machines using DASK.

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 that you successfully installed DASK.
2. Provide a screenshot to show that you successfully ran the `dask-scheduler`.

3. Provide two screenshots to show that you successfully started both of the `dask-worker` processes by passing the correct address to the `dask-scheduler`.
4. Provide a screenshot to show that you created a Jupyter Notebook and successfully imported the DASK *libraries*.
5. Provide a screenshot to show that you successfully executed the command to create a matrix, compute the mean, and assign the value to the `y` variable.
6. Provide a screenshot to show that you successfully computed your calculations using DASK using the `compute` *function*.

Additional Details:

- **Estimated time:** 45 minutes
- *This is a required activity and counts toward course completion.*

Activity 22.3

Criteria	Ratings		Pts
1. Provide a screenshot to show that you successfully installed DASK.	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
2. Provide a screenshot to show that you successfully ran the dask-scheduler.	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
3. Provide two screenshots to show that you successfully started both of the dask-worker processes by passing the correct address to the dask-scheduler.	5 pts Complete Both correct screenshots have been included in the submission.	0 pts Incomplete Both screenshots have not been included in the submission or are the incorrect screenshots.	5 pts
4. Provide a screenshot to show that you created a Jupyter Notebook and successfully imported the DASK libraries.	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
5. Provide a screenshot to show that you successfully executed the command to create a matrix, compute the mean, and assign the value to the y variable.	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 to show that you successfully computed your calculations using DASK using the compute function.	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

