

# Practica 5: Instalación de Dask

3 de mayo de 2022

Biblioteca disponible en:  
<https://dask.org/>

## 1. Objetivo de la practica.

Mostrar cómo se realiza la instalación de Dask y cómo se usa para la manipulación de grandes volúmenes de datos.

## 2. Conceptos.

Dask es una biblioteca flexible para computación paralela en Python.

Dask se compone de dos partes:

- a. **Programación dinámica** de tareas optimizada para el cálculo. Esto es similar a Airflow, Luigi, Celery o Make , pero optimizado para cargas de trabajo computacionales interactivas.
- b. **Colecciones de "Big Data"** como matrices paralelas, DataFrames y listas que extienden interfaces comunes como iteradores NumPy, Pandas o Python a entornos distribuidos o de mayor tamaño que la memoria. Estas colecciones paralelas se ejecutan sobre programadores de tareas dinámicos.

## 3. Herramientas a usar.

Computadora con acceso a internet.  
Los siguientes programas y librerías:

- a. Python versión 3.8.8
- b. Anaconda versión 4.10.1
- c. Jupyter-lab 3.0.14

## 4. Desarrollo.

### 4.1. Entender el problema.

Existen datasets muy grandes los cuales hacen casi imposible su procesamiento en una computadora convencional, incluso en computadoras más sofisticadas, debido a la gran carga computacional que representa cargar y manipular esos datos.

Para este tipo de casos, existen diferentes herramientas que ayudan al procesamiento óptimo de grandes volúmenes de datos. Tal es el caso de Dask, el cual es compatible con Python, además de compartir una sintaxis muy similar a la de pandas.

### 4.2. Criterio de evaluación.

El criterio consiste en la correcta instalación de Dask y que pueda cargar y manipular datos de manera adecuada, o sea, sin saturar la memoria de la computadora (18 CPU's y 32 GB de RAM).

### 4.3. Prerrequisitos.

- **Instalación de Anaconda.**

Lo primero que se tiene que hacer es instalar anaconda (solo en caso de que no esté instalado en el equipo). Para instalar anaconda, se puede ingresar a la página web <https://www.anaconda.com/products/individual> descargar el instalador necesario para su sistema.

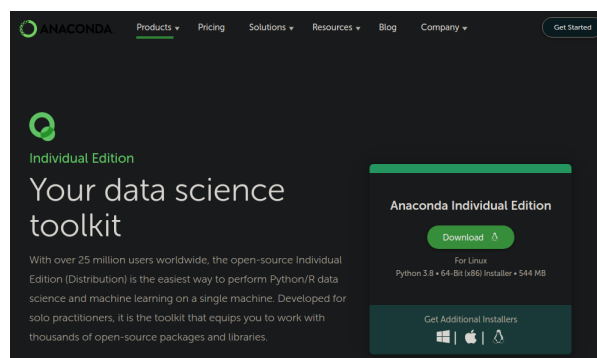


Figura 1: Página de descarga de Anaconda

En caso de contar con un sistema linux, se puede instalar anaconda desde terminal siguiendo estas instrucciones:

Use el siguiente comando para descargar el instalador de anaconda.

```
wget https://repo.anaconda.com/archive/Anaconda3-2021.05-Linux-x86_64.sh
```



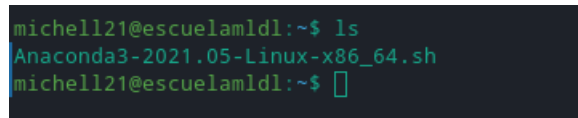
```
michell121@escuelamidl:~$ wget https://repo.anaconda.com/archive/Anaconda3-2021.05-Linux-x86_64.sh
--2021-11-02 23:27:40-- https://repo.anaconda.com/archive/Anaconda3-2021.05-Linux-x86_64.sh
Resolving repo.anaconda.com (repo.anaconda.com)... 104.16.130.3, 104.16.131.3, 2606:4700::6810:8303, ...
Connecting to repo.anaconda.com (repo.anaconda.com)|104.16.130.3|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 570853747 (544M) [application/x-sh]
Saving to: 'Anaconda3-2021.05-Linux-x86_64.sh'

Anaconda3-2021.05-Linux-x86_64.s 100%[=====] 544.41M 25.9MB/s in 20s
2021-11-02 23:28:00 (27.4 MB/s) - 'Anaconda3-2021.05-Linux-x86_64.sh' saved [570853747/570853747]
```

Figura 2: Descarga de Anaconda en terminal

Cuando la descarga haya finalizado, Verifique que el instalador se encuentre en su equipo usando el comando `ls`.

Use el siguiente comando para instalar anaconda.

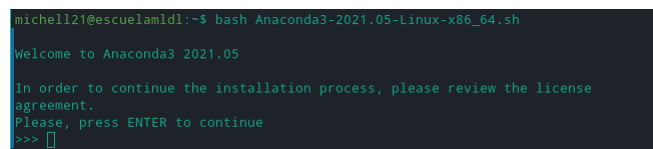


```
michell121@escuelamidl:~$ ls
Anaconda3-2021.05-Linux-x86_64.sh
michell121@escuelamidl:~$
```

Figura 3: Uso del comando `ls` para ver los archivos.

```
bash Anaconda3-<versión>-Linux-x86_64.sh
```

Y siga las instrucciones de muestra el instalador.



```
michell121@escuelamidl:~$ bash Anaconda3-2021.05-Linux-x86_64.sh
Welcome to Anaconda3 2021.05

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
```

Figura 4: Instalación de Anaconda.

Cuando pregunte si quiere activar conda init, seleccione “yes” y siga con la instalación.

```

Please answer 'yes' or 'no':
>>> yes

Anaconda3 will now be installed into this location:
/home/michell21/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/michell21/anaconda3] >>>
PREFIX=/home/michell21/anaconda3
Unpacking payload ...

```

Figura 5: Configuración de Anaconda.

Cuando la instalación haya finalizado, cierre la terminal actual y abra una nueva para que los cambios se apliquen en el sistema.

```

Thank you for installing Anaconda3!

=====

Working with Python and Jupyter notebooks is a breeze with PyCharm Pro,
designed to be used with Anaconda. Download now and have the best data
tools at your fingertips.

PyCharm Pro for Anaconda is available at: https://www.anaconda.com/pycharm

michell21@escuelamld1:~$ 

```

Figura 6: Instalación finalizada.

Una vez que haya abierto una nueva terminal, podrá ver *(base)* en la terminal. Para comprobar que todo se instaló correctamente, use el siguiente comando:

```
conda list
```

Y debería mostrar todas las librerías instaladas con anaconda.

```

(base) michell21@escuelamld1:~$ conda list
# packages in environment at /home/michell21/anaconda3:
#
# Name                   Version           Build    Channel
_ipyw_jlab_nb_ext_conf  0.1.0             py38_0  main
libgcc_mutex             0.1               pyhd3eb1b0_0
alabaster                0.7.12            pyhd3eb1b0_0
anaconda                 2021.05           py38_0
anaconda-client          1.7.2             py38_0
anaconda-navigator       2.0.3             py38_0
anaconda-project         0.9.1             pyhd3eb1b0_1
anyio                    2.2.0             py38h06a4308_1
appdirs                  1.4.4             py_0
argh                     0.26.2            py38_0
argon2-cffi              20.1.0            py38h27cfd23_1
asn1crypto               1.4.0             py_0

```

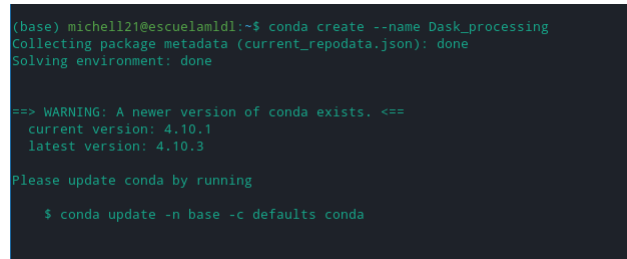
Figura 7: Librerías de anaconda.

- **Entornos virtuales con Conda.** Antes de instalar Dask, es recomendable crear un entorno virtual para trabajar únicamente con los recursos necesarios.

El comando para crear un nuevo entorno virtual es el siguiente:

```
conda create --name <nombre-del-entorno>
```

Al ingresar ese comando, anaconda configura todo lo necesario para crear el nuevo entorno virtual.



```
(base) michell21@escuelamld1:~$ conda create --name Dask_processing
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.10.1
  latest version: 4.10.3

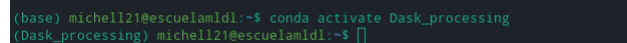
Please update conda by running

    $ conda update -n base -c defaults conda
```

Figura 8: Creación de un nuevo entorno virtual.

Si todo se ejecuta correctamente, el entorno se puede activar usando:

```
conda activate <nombre-del-entorno>
```



```
(base) michell21@escuelamld1:~$ conda activate Dask_processing
(Dask_processing) michell21@escuelamld1:~$ []
```

Figura 9: Activar el nuevo entorno virtual.

Y se desactiva usando:

```
conda deactivate <nombre-del-entorno>
```

Ya con todo esto instalado, ahora es posible instalar Dask en el equipo para empezar a trabajar con él.

#### 4.4. Instalar la librería Dask.

Al tener anaconda instalado, se pueden usar los comandos de conda para hacer una instalación más fácil. Antes de instalar Dask, asegúrese de tener el entorno virtual, previamente creado, activo.

Para instalar alguna librería con conda, se usa el siguiente comando:

```
conda install <nombre-del-entorno>
```

Por lo tanto, la instalación de Dask quedaría de la siguiente forma

```
conda install dask
```

```
(base) michell21@escuelamld1:~$ conda activate Dask_processing
(Dask_processing) michell21@escuelamld1:~$ conda install dask
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.10.1
  latest version: 4.10.3

Please update conda by running

  $ conda update -n base -c defaults conda
```

Figura 10: Instalación de Dask.

Durante el proceso, se instalarán también todas las demás librerías que Dask necesita para funcionar correctamente, se acepta la instalación de estas y se espera hasta que la instalación termine.

```
The following NEW packages will be INSTALLED:

_libgcc_mutex           pkgs/main/linux-64::_libgcc_mutex-0.1-main
_openmp_mutex           pkgs/main/linux-64::_openmp_mutex-4.5-1_gnu
blas                    pkgs/main/linux-64::blas-1.0-mkl
bokeh                   pkgs/main/linux-64::bokeh-2.4.1-py39h06a4308_0
bottleneck              pkgs/main/linux-64::bottleneck-1.3.2-py39hdd57654_1
ca-certificates         pkgs/main/linux-64::ca-certificates-2021.10.26-h06a4308_2
certifi                 pkgs/main/linux-64::certifi-2021.10.8-py39h06a4308_0
click                   pkgs/main/noarch::click-8.0.3-pyhd3eb1b0_0
cloudpickle             pkgs/main/noarch::cloudpickle-2.0.0-pyhd3eb1b0_0
cytoolz                 pkgs/main/linux-64::cytoolz-0.11.0-py39h27cfd23_0
dask                    pkgs/main/noarch::dask-2021.9.1-pyhd3eb1b0_0
dask-core               pkgs/main/noarch::dask-core-2021.9.1-pyhd3eb1b0_0
distributed             pkgs/main/linux-64::distributed-2021.9.1-py39h06a4308_0
freetype                pkgs/main/linux-64::freetype-2.11.0-h70c0345_0
fsspec                  pkgs/main/noarch::fsspec-2021.8.1-pyhd3eb1b0_0
giflib                  pkgs/main/linux-64::giflib-5.2.1-h7b6447c_0
heapdict                pkgs/main/noarch::heapdict-1.0.1-pyhd3eb1b0_0
intel-openmp            pkgs/main/linux-64::intel-openmp-2021.4.0-h06a4308_3561
jinja2                  pkgs/main/noarch::jinja2-3.0.2-pyhd3eb1b0_0
jpeg                    pkgs/main/linux-64::jpeg-9d-h7f8727e_0
lcms2                   pkgs/main/linux-64::lcms2-2.12-h3be6417_0
ld_impl_linux-64        pkgs/main/linux-64::ld_impl_linux-64-2.35.1-h7274673_9
libffi                  pkgs/main/linux-64::libffi-3.3-he6710b0_2
```

Figura 11: Librerías a instalar.

Cuando la instalación haya terminado, use de nuevo el comando `conda list` para comprobar que las librerías están instaladas correctamente

#	Name	Version	Build	Channel
1	_libgcc_mutex	0.1		main
2	_openmp_mutex	4.5	1_gnu	
3	blas	1.0	mkl	
4	bokeh	2.4.1	py39h06a4308_0	
5	bottleneck	1.3.2	py39hdd57654_1	
6	ca-certificates	2021.10.26	h06a4308_2	
7	certifi	2021.10.8	py39h06a4308_0	
8	click	8.0.3	pyhd3eb1b0_0	
9	cloudpickle	2.0.0	pyhd3eb1b0_0	
10	cytoolz	0.11.0	py39h27cfd23_0	
11	dask	2021.9.1	pyhd3eb1b0_0	
12	dask-core	2021.9.1	pyhd3eb1b0_0	
13	distributed	2021.9.1	py39h06a4308_0	
14	freetype	2.11.0	h70c0345_0	
15	fsspec	2021.8.1	pyhd3eb1b0_0	

Figura 12: Librerías instaladas.

## 4.5. Hacer pruebas.

Una de las pruebas más fáciles que se puede hacer en Dask, es comparar el tiempo que tarda en cargar un dataset grande, en comparación con pandas.

Para medir el tiempo de respuesta de Dask y pandas. El dataset que se va a usar, es el dataset de datos climaticos de la Red Universitaria de Observatorios Atmosféricos (RUOA), iniciando en 2015-08-01 hasta 2021-06-31 con un lapso de una hora entre cada medición.

Primero se tiene que hacer es importar las librerías.

```
import os
import glob
import zipfile
import pandas as pd
import dask.dataframe as dd
```

Los archivos dentro de este dataset, vienen divididos en un archivo csv por mes. Para que la medición sea lo más justo posible, estos archivos se deben unir en uno solo. El siguiente código lee cada uno de los archivos csv contenidos en la carpeta “Datos\_ruoa.” en la dirección *Path* y los concatena en un solo archivo llamado “Data\_complete”. Es importante remarcar que en cada archivo se descartan las primeras 6 líneas ya que éstas no aportan información, la línea 8 se elimina por la misma razón.

```
Path = "/dirección/a_los/archivos/"
```

```
csv_files = glob.glob(Path+'Datos_ruoa/*.csv')
csv_files.sort()
```

```
list_data = []
```

```
for filename in csv_files:
    data = pd.read_csv(filename, skiprows=5)
    data.drop(0, axis=0, inplace=True)
    list_data.append(data)
```

```
df = pd.concat(list_data, ignore_index=True)
df.to_csv(Path+"Data_complete.csv")
```

“Data\_complete.csv” contiene los datos de todas las mediciones desde 2015-08-01 hasta 2021-06-31. Ahora es momento de hacer la comparación entre pandas y Dask.

Para medir el tiempo que tarda pandas en cargar el archivo se usa *%time*.

```
%time df_pandas = pd.read_csv(Path+"Data_complete.csv")
```

El resultado sería este:

```
CPU times: user 81.7 ms, sys: 19.9 ms, total: 102 ms
Wall time: 101 ms
```

Figura 13: Tiempo de respuesta de Pandas

Y se puede comprobar que los datos se han cargado correctamente usando `head` En el caso de Dask, sería el siguiente.

Unnamed: 0		TIMESTAMP	Temp_Avg	RH_Avg	WSpeed_Avg	WSpeed_Max	WDir_Avg	Rain_Tot	Press_Avg	Rad_Avg
0	0	2015-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1	2015-08-01 01:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	2	2015-08-01 02:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	3	2015-08-01 03:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	4	2015-08-01 04:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figura 14: Datos cargados con Pandas

```
% time df_dask = dd.read_csv(Path+"Data_complete.csv")
```

Y muestra el siguiente tiempo de respuesta.

```
CPU times: user 12 ms, sys: 4.28 ms, total: 16.3 ms
Wall time: 15.2 ms
```

Figura 15: Tiempo de respuesta de Dask

Dask maneja los datos de manera muy similar a pandas, por lo tanto el uso de `head` se implementa de la misma manera. Como se puede apreciar en las

Unnamed: 0		TIMESTAMP	Temp_Avg	RH_Avg	WSpeed_Avg	WSpeed_Max	WDir_Avg	Rain_Tot	Press_Avg	Rad_Avg
0	0	2015-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1	2015-08-01 01:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	2	2015-08-01 02:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	3	2015-08-01 03:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	4	2015-08-01 04:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figura 16: Datos cargados con Dask



imágenes 13 y 15, Dask tiene mucho mejor tiempo de respuesta, esto ayuda a la optimización del proceso. Otra característica es que Dask puede paralelizar el procesamiento de datos, pero eso se verá en próximas prácticas.

#### **4.6. Implementación**

Ver el Notebook de Jupyter llamado *P5-Dask.ipynb*