

Python

Cómputo Paralelo con MPI



Rogelio Ferreira Escutia

Profesor / Investigador
Tecnológico Nacional de México
Campus Morelia



Conceptos

MPI - Definición

- **MPI ("Message Passing Interface", Interfaz de Paso de Mensajes) es un estándar que define la sintaxis y la semántica de las funciones contenidas en una biblioteca de paso de mensajes diseñada para ser usada en programas que exploten la existencia de múltiples procesadores.**

MPI - Definición

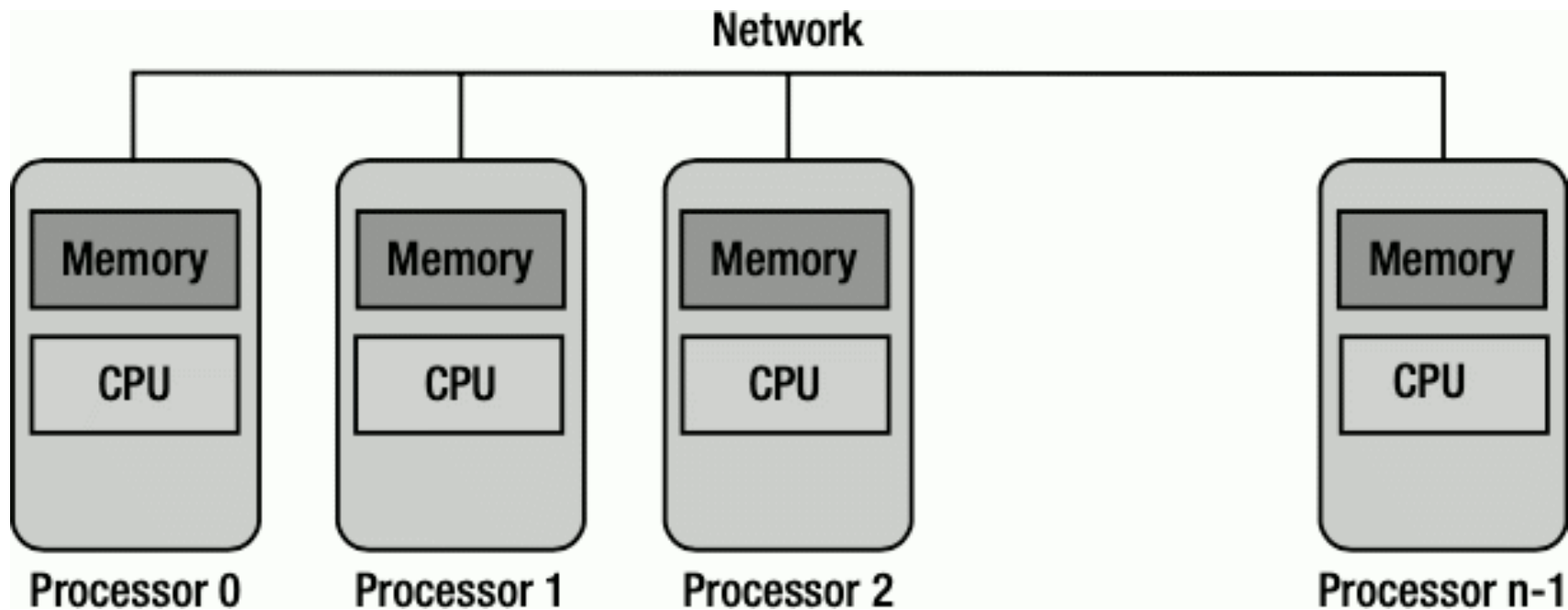
- **El paso de mensajes es una técnica empleada en programación concurrente para aportar sincronización entre procesos y permitir la exclusión mutua, de manera similar a como se hace con los semáforos, monitores, etc, y su principal característica es que no precisa de memoria compartida, por lo que es muy importante en la programación de sistemas distribuidos.**

MPI - Historia

- El esfuerzo para estandarizar MPI involucró a cerca de 60 personas de 40 organizaciones diferentes principalmente de EE.UU. y Europa.
- El proceso de estandarización comenzó en el taller de estándares para el paso de mensajes en un ambiente con memoria distribuida, patrocinado por el Centro de Investigación en Computación Paralela en Williamsburg, Virginia, Estados Unidos (abril 29-30 de 1992).
- El estándar final por el MPI fue presentado en la conferencia de Supercómputo en noviembre de 1993, constituyéndose así el foro para el MPI.

MPI

- **MPI es un biblioteca que se ejecuta en diferentes procesadores que se comunican por red:**



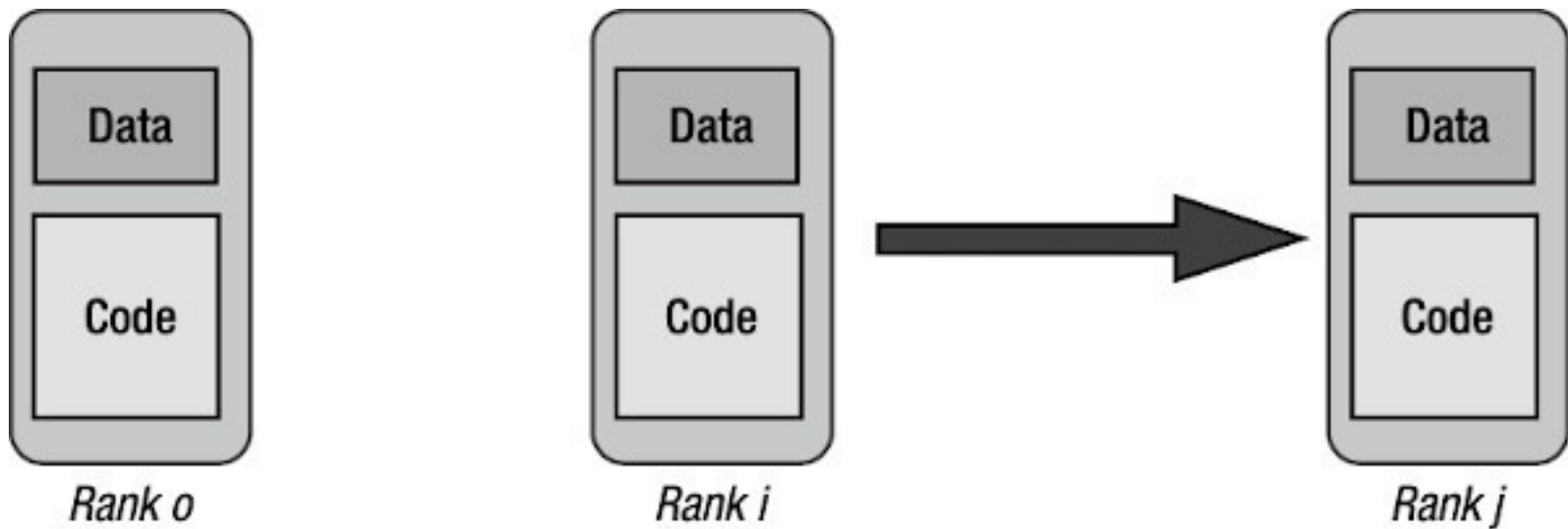
MPI - Características

- **Simple:** Se basa en el paradigma de comunicación por paso de mensajes.
- **General:** Puede ser usado en la gran mayoría de los sistemas paralelos.
- **Desempeño:** Puede igualar o superar implementaciones basadas en hardware dedicado.
- **Escalabilidad:** Un mismo programa puede ejecutarse en sistemas pequeños o grandes sin ningún cambio en el código.



MPI

- Programas corriendo de manera simultánea



MPI - Implementaciones

- **MPI es un estándar que tiene varias implementaciones, algunas de ellas son las siguientes:**
 - **MPICH**
 - **MP-MPICH (MP para multi-plataforma)**
 - **Winmpich**
 - **MPI BIP**
 - **HP's MPI**
 - **IBM's MPI**
 - **SGI's MPI**
 - **STAMPI**
 - **OpenMPI**



MPI4PY

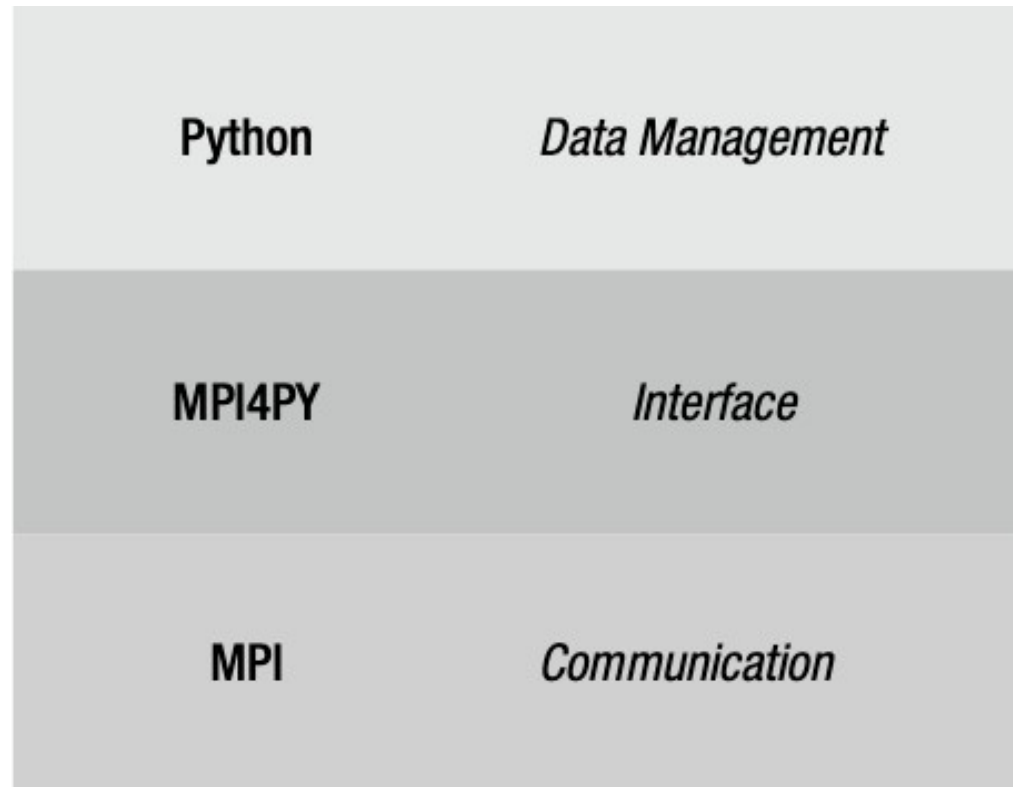
MPI4PY

- **MPI4PY es un estándar de MPI para Python y soporta las especificaciones de MPI versión 1, 2 y 3.**
- **Permite a los programas en Python ejecutarse en computadoras con múltiples procesadores.**



MPI4PY

- **Arquitectura de MPI4PY:**



MPI4PY – Funciones básicas

- **MPI rank (identificación de procesadores).**
- **Envío y recepción de datos.**
- **Etiquetado de datos.**
- **Broadcasting (envío de datos a todos).**

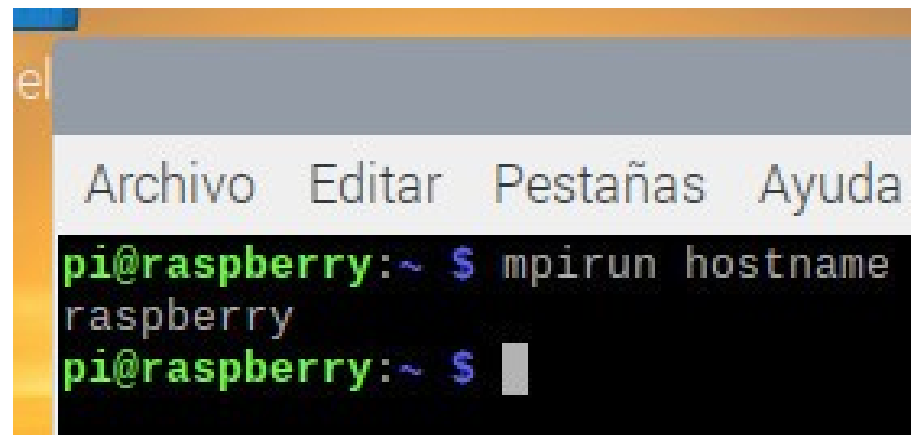
Instalación y prueba

MPI4PY – Biblioteca

- **Se requiere instalar “mpi4py”:**
- **En Linux (consola):**
 - **sudo apt-get install mpich python3-mpi4py**

MPI4PY – Prueba

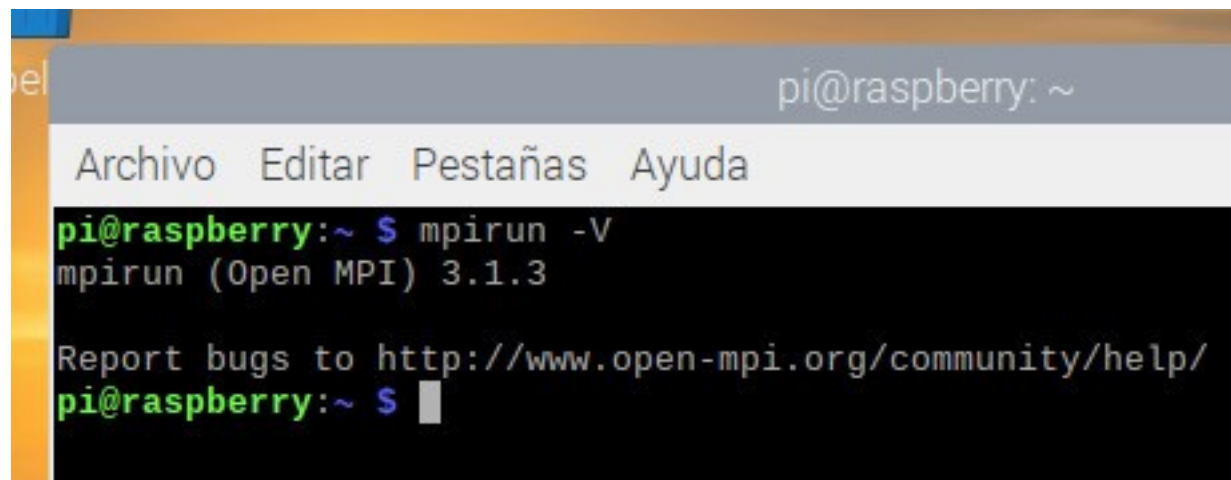
- Ejecutar el siguiente comando para probar si quedó instalado:
- En Linux (consola):
 - \$ mpirun hostname



```
el
Archivo  Editar  Pestañas  Ayuda
pi@raspberrypi:~ $ mpirun hostname
raspberrypi
pi@raspberrypi:~ $ █
```


MPI4PY – Versión

- Ejecutar el siguiente comando para revisar la versión instalada:
- En Linux (consola):
 - `$ mpirun -V`



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~ $ mpirun -V  
mpirun (Open MPI) 3.1.3  
Report bugs to http://www.open-mpi.org/community/help/  
pi@raspberrypi:~ $
```

Funciones Básicas

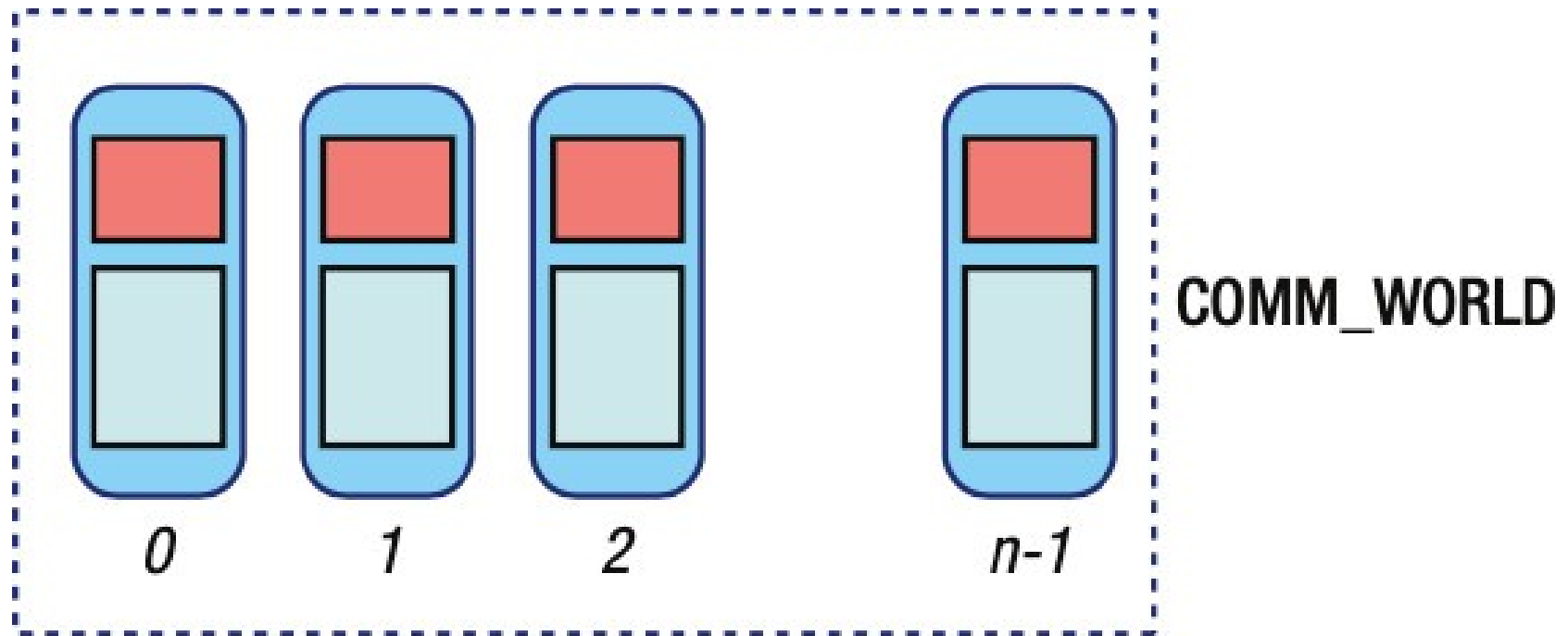
MPI4PY – Cargar librería

- Se inicia cargando la librería mpi4py.
- En Python:

```
from mpi4py import MPI
```

MPI

- Enumerando los procesadores



MPI4PY – Tamaño

- Se debe saber la cantidad de procesadores disponibles, para poder determinar el tamaño del Cluster.
- En Python:

```
size = MPI.COMM_WORLD.Get_size()
```

MPI4PY – Rank

- **Se debe saber el Rank (rango o estatus) de nuestro procesador con respecto a los demás, para saber si somos maestro ó esclavo (el cero es el maestro).**
- **En Python:**

```
rank = MPI.COMM_WORLD.Get_rank()
```

MPI4PY – Nombre

- Se puede obtener el nombre de nuestro host (computadora).
- En Python:

```
name = MPI.Get_processor_name()
```

Hola Mundo

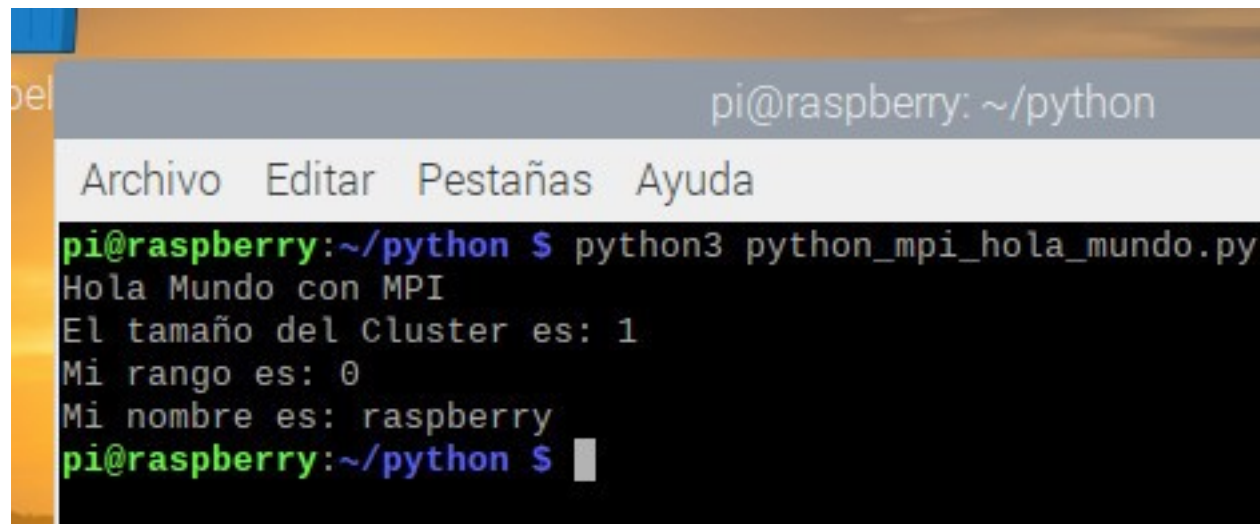
MPI

- **Hola Mundo**

```
from mpi4py import MPI
```

```
size = MPI.COMM_WORLD.Get_size()  
rank = MPI.COMM_WORLD.Get_rank()  
name = MPI.Get_processor_name()
```

```
print("\nHola Mundo con MPI")  
print("El tamaño del Cluster es: " + str(size))  
print("Mi rango es: " + str(rank))  
print("Mi nombre es: " + str(name))
```



```
del pi@raspberrypi: ~/python  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~/python $ python3 python_mpi_hola_mundo.py  
Hola Mundo con MPI  
El tamaño del Cluster es: 1  
Mi rango es: 0  
Mi nombre es: raspberry  
pi@raspberrypi:~/python $
```





Rogelio Ferreira Escutia

Profesor / Investigador
Tecnológico Nacional de México
Campus Morelia



rogelio.fe@morelia.tecnm.mx



rogeplus@gmail.com



xumarhu.net



[@rogeplus](https://twitter.com/rogeplus)



[https://www.youtube.com/
channel/UC0on88n3LwTKxJb8T09sGjg](https://www.youtube.com/channel/UC0on88n3LwTKxJb8T09sGjg)



[rogelioferreiraescutia](https://www.linkedin.com/in/rogelioferreiraescutia)

