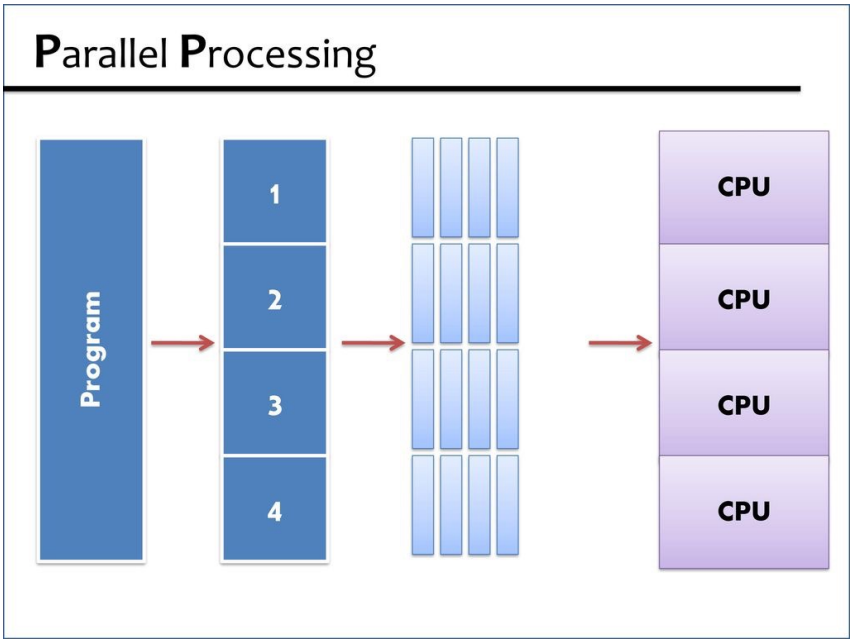
Harold Molina-Bulla DTSC-UC3M

PARALLEL PROGRAMING IN PYTHON



2021

Harold Molina-Bulla DTSC-UC3M

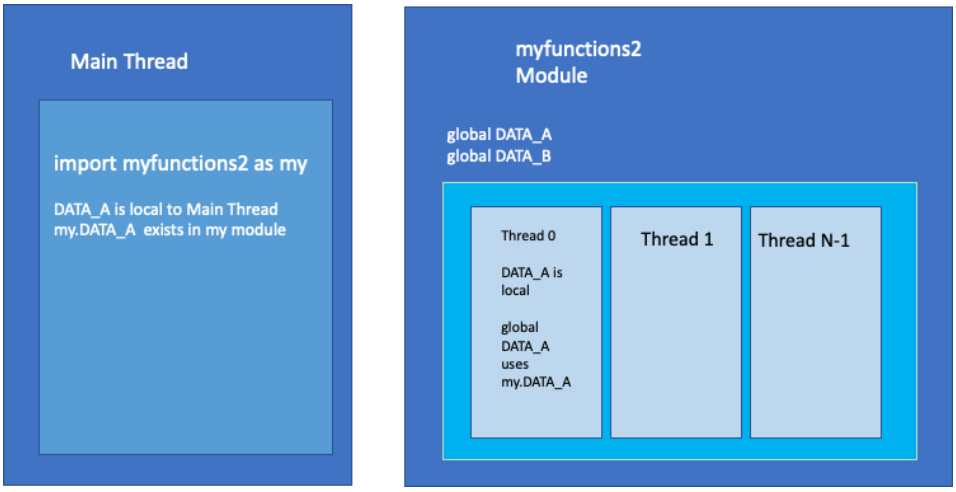
SHARED MEMORY

- Shared Memory: Memory with access from multiple threads
- Advantages:
  - Speed of access
- Disadvantages:
  - Memory Hazzards:
    - Read after write
    - Write after read
    - Write after write

MEMORY SCOPE

- The variable is defined in:
  - Module where it is defined
  - Function where it is defined
- Problems with global read only memory:
  - We must take care where the variable exists.

MEMORY SCOPE



Harold Molina-Bulla DTSC-UC3M

MEMORY HANDLE IN PYTHON

- Global variables must be used as read-only memory defined in python when we call inside a functions.
- We can access a global variable, referencing it inside the functions
- But, if we change the values of this variable, it will say it is not defined and create a memory exception
- Broadcasting Variables: Send the common values to all parallel processes, before they starts and handle it like read-only variables.

Harold Molina-Bulla DTSC-UC3M

IMAGE FILTERING

- Consider the image and a filter mask as a matrices of different sizes. To apply a filter mask to an image, we will apply the filter mask pixel by pixel.
- To filter a pixel, we will use the neighbours pixels (that means, the pixels around our pixel of interest)
- The filters will have an odd number columns and rows.
- The center cell of the filter mask corresponds to the coefficient of the pixel of interest.

EXAMPLE

- We have a filter mask of size 3x3 (rows 0,1,2 and columns 0,1,2)
- We have an image of size, for example, 14x12x3. That means: size of 14x12 and 3 colour layers.
- Several cases:
  - Pixel in the middle of the image, for example position 10,10 in the layer 0
  - The coefficient in the filter mask cell (1,1) will multiply the pixel in the position (10,10,0) in the image

Harold Molina-Bulla DTSC-UC3M

Filter mask

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Image Matrix

(6,6,0)	(6,7,0)	(6,8,0)	(6,9,0)	(6,10,0)	(6,11,0)	(6,12,0)
(7,6,0)	(7,7,0)	(7,8,0)	(7,9,0)	(7,10,0)	(7,11,0)	(7,12,0)
(8,6,0)	(8,7,0)	(8,8,0)	(8,9,0)	(8,10,0)	(8,11,0)	(8,12,0)
(9,6,0)	(9,7,0)	(9,8,0)	(9,9,0)	(9,10,0)	(9,11,0)	(9,12,0)
(10,6,0)	(10,7,0)	(10,8,0)	(10,9,0)	(10,10,0)	(10,11,0)	(10,12,0)
(11,6,0)	(11,7,0)	(11,8,0)	(11,9,0)	(11,10,0)	(11,11,0)	(11,12,0)
(12,6,0)	(12,7,0)	(12,8,0)	(12,9,0)	(12,10,0)	(12,11,0)	(12,12,0)
(13,6,0)	(13,7,0)	(13,8,0)	(13,9,0)	(13,10,0)	(13,11,0)	(13,12,0)
(14,6,0)	(14,7,0)	(14,8,0)	(14,9,0)	(14,10,0)	(14,11,0)	(14,12,0)

$$
\begin{aligned}
Fp_{10,10,0} &= Img_{10,10,0} * F_{1,1} + \\
&= Img_{9,9,0} * F_{0,0} + Img_{9,10,0} * F_{0,1} + Img_{9,11,0} * F_{0,2} + \\
&= Img_{10,9,0} * F_{1,0} + Img_{10,11,0} * F_{1,2} + \\
&= Img_{11,9,0} * F_{2,0} + Img_{11,10,0} * F_{2,1} + Img_{11,11,0} * F_{2,2}
\end{aligned}
$$

SPECIAL CASES

- In the borders:
  - Right border

$$
\begin{aligned}
Fp_{10,12,0} &= Img_{10,12,0} * F_{1,1} + \\
&= Img_{9,11,0} * F_{0,0} + Img_{9,12,0} * F_{0,1} + Img_{9,12,0} * F_{0,2} + \\
&= Img_{10,11,0} * F_{1,0} + Img_{10,12,0} * F_{1,2} + \\
&= Img_{11,11,0} * F_{2,0} + Img_{11,12,0} * F_{2,1} + Img_{11,12,0} * F_{2,2}
\end{aligned}
$$

SPECIAL CASES

- Left border

$$
\begin{aligned}
Fp_{10,0,0} &= Img_{10,0,0} * F_{1,1} + \\
&= Img_{9,0,0} * F_{0,0} + Img_{9,0,0} * F_{0,1} + Img_{9,1,0} * F_{0,2} + \\
&= Img_{10,0,0} * F_{1,0} + Img_{10,0,0} * F_{1,2} + \\
&= Img_{11,0,0} * F_{2,0} + Img_{11,0,0} * F_{2,1} + Img_{11,1,0} * F_{2,2}
\end{aligned}
$$

SPECIAL CASES

- Top border

$$
\begin{aligned}
Fp_{0,10,0} &= Img_{0,10,0} * F_{1,1} + \\
&= Img_{0,9,0} * F_{0,0} + Img_{0,10,0} * F_{0,1} + Img_{0,11,0} * F_{0,2} + \\
&= Img_{0,9,0} * F_{1,0} + Img_{0,11,0} * F_{1,2} + \\
&= Img_{1,9,0} * F_{2,0} + Img_{1,10,0} * F_{2,1} + Img_{1,11,0} * F_{2,2}
\end{aligned}
$$

Harold Molina-Bulla DTSC-UC3M

SPECIAL CASES

- Lower Border

$$
\begin{aligned}
Fp_{14,10,0} &= Img_{14,10,0} * F_{1,1} + \\
&= Img_{13,9,0} * F_{0,0} + Img_{13,10,0} * F_{0,1} + Img_{13,11,0} * F_{0,2} + \\
&= Img_{14,9,0} * F_{1,0} + Img_{14,11,0} * F_{1,2} + \\
&= Img_{14,9,0} * F_{2,0} + Img_{14,10,0} * F_{2,1} + Img_{14,11,0} * F_{2,2}
\end{aligned}
$$

SPECIAL CASES

- What happens in the corners??
- What happens if the filter is not 3x3, but 5x5 or bigger?
- What happens if the filter is not square (AxA), but rectangular (AxB) where AxB are odds.