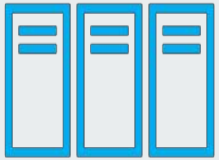
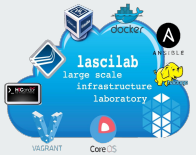


2 Seminario de Plataformas Computacionales

Computación de Alto Rendimiento y Reproducibilidad Experimental



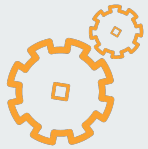
Introducción a OpenACC

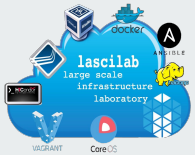
Aurelio Vivas - aurelio.vivas@correounivalle.edu.co

docker

Universidad del Valle

CIBioFi
Centro de Investigación e Innovación en Bioinformática y Fotónica





Lo que aprenderás hoy

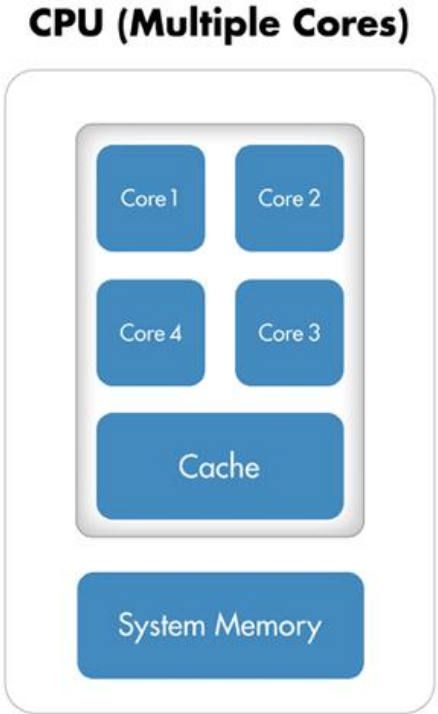
¿Cómo OpenMP puede mejorar el rendimiento de tus experimentos?

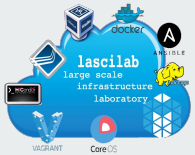
- **Resumen de la sección anterior**
- Computación Heterogénea
- ¿Por qué es importante la Computación Heterogénea ?
- OpenACC

Resumen

OpenMP

1. Programación en CPU.
    a. Basado en directivas de compilación.
    b. Directivas
        i. Parallel
        ii. For
        iii. Sections





Lo que aprenderás hoy

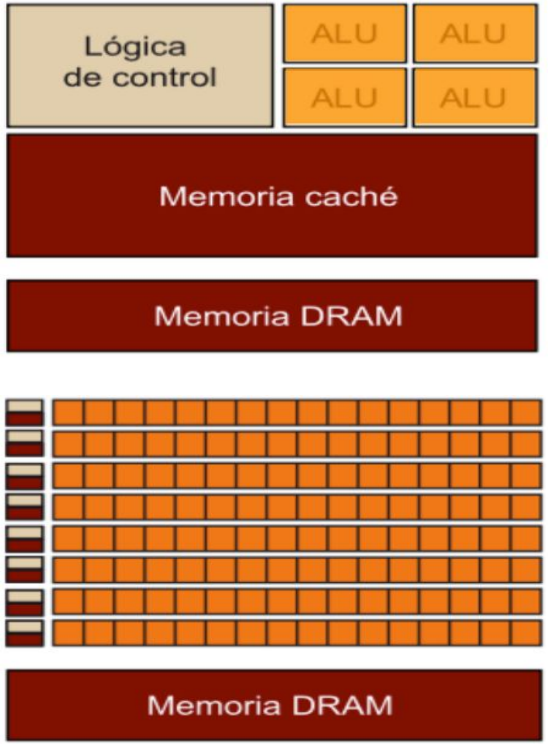
¿Cómo OpenMP puede mejorar el rendimiento de tus experimentos?

- Resumen de la sección anterior
- Computación Heterogénea
- ¿Por qué es importante la Computación Heterogénea ?
- OpenACC

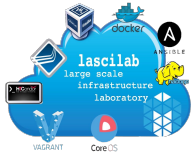
Computación Heterogénea

Qué es?

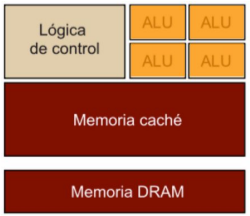
1. Nace con la necesidad de explotar el paralelismo de los procesadores de gráficos.
2. Se fundamenta en el uso de unidades de procesamiento de naturalezas distintas (GPU + CPU).
3. Nace el concepto de GPGPU (General Purpose Graphic Processing Unit)



Computación Heterogénea

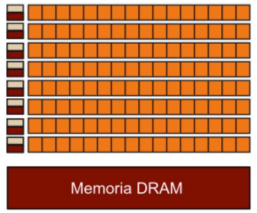


CPU



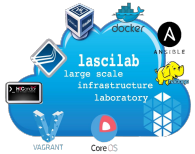
- Procesador Escalar.
- Paralelismo de tareas.
- Una instrucción opera sobre un dato.
- Alto rendimiento sobre un único hilo de ejecución.
- Modelo de programación muy conocido.
- MIMD.
- Poco paralelismo.
- Poca concurrencia.

GPU



- Procesador Vectorial.
- Paralelismo de datos.
- Una instrucción implica mucho trabajo.
- Alto rendimiento sobre gran cantidad de datos.
- Modelo de programación poco conocido.
- SIMD.
- Mucho Paralelismo.
- Mucha Concurrencia.

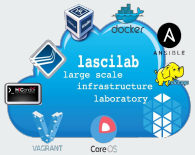
Computación Heterogénea



Casos Exitosos

- **Álgebra Lineal**
- **Procesamiento de Imágenes**
- Algoritmos de ordenamiento y búsqueda
- Procesamiento de consultas sobre bases de datos
- Análisis financiero
- **Mecánica de Fluidos**
- Predicción Meteorológica

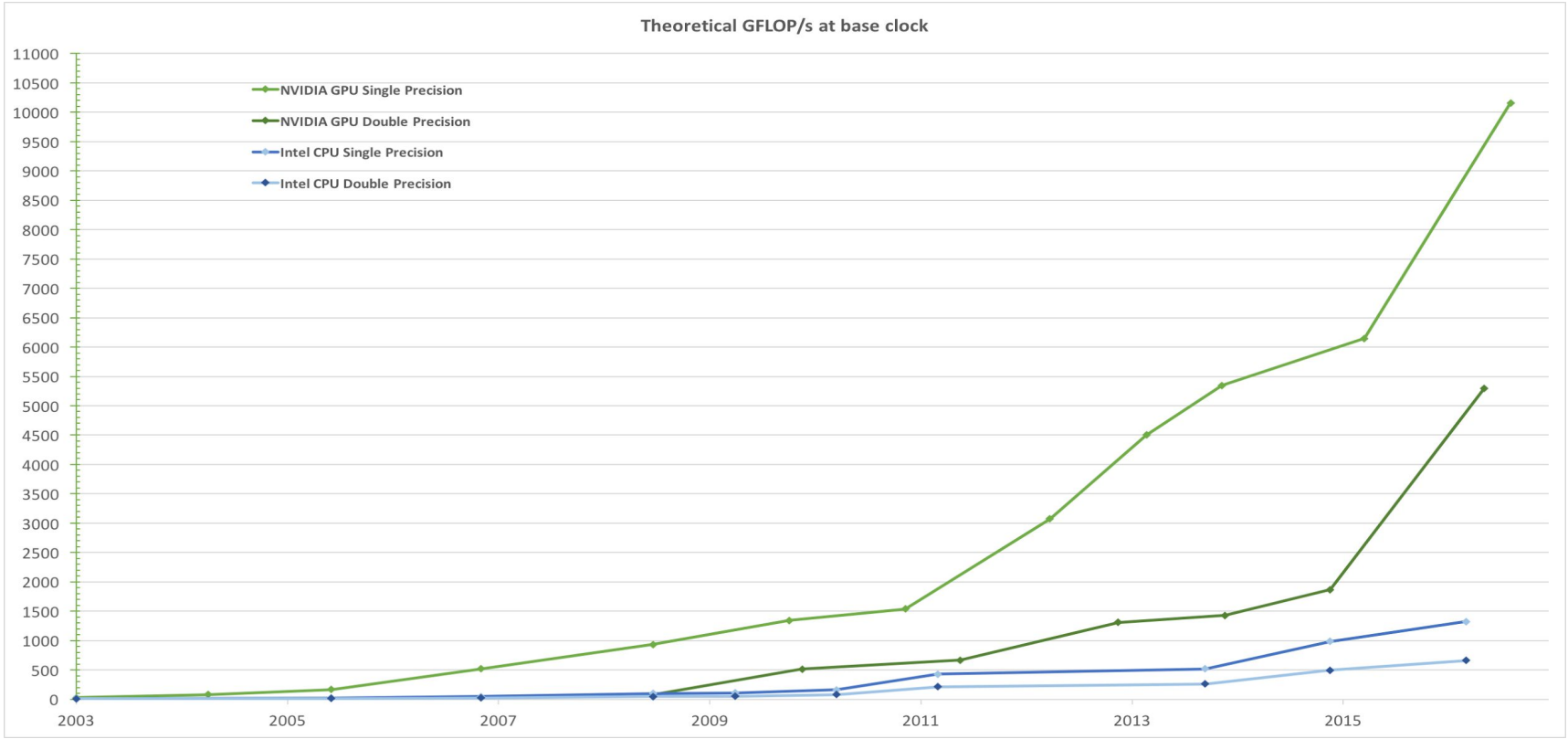
- **Simulaciones científicas**
- Análisis intensivo de datos
- Imágenes médicas
- **Procesamiento digital de audio y video**
- **Métodos númericos**
- Informatica biomédica
- etc



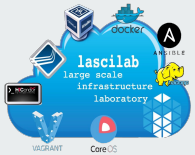
Lo que aprenderás hoy

¿Cómo OpenMP puede mejorar el rendimiento de tus experimentos?

- Resumen de la sección anterior
- Computación Heterogénea
- **¿Por qué es importante la Computación Heterogénea ?**
- OpenACC



¿Por qué es importante la computación heterogénea?

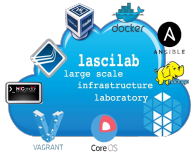


Lo que aprenderás hoy

¿Cómo OpenMP puede mejorar el rendimiento de tus experimentos?

- Resumen de la sección anterior
- Computación Heterogénea
- ¿Por qué es importante la Computación Heterogénea ?
- OpenACC

OpenACC



¿Qué es?

OpenACC (Open Accelerators) es un **modelo de programación** y un **lenguaje basado en directivas de compilación**.

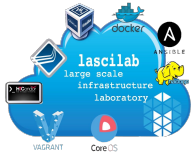
Permite expresar el paralelismo en las aplicaciones por medio de **anotaciones en el código.**

El programa paralelizado toma ventaja de los múltiples cores de una **GPU**.

¿Cómo se usa?

```
int main(void){
  #pragma acc parallel
  { // Start Parallel Region
    // some code .. to be executed in parallel
  } // End Parallel Region
  return 0;
}
```

OpenACC



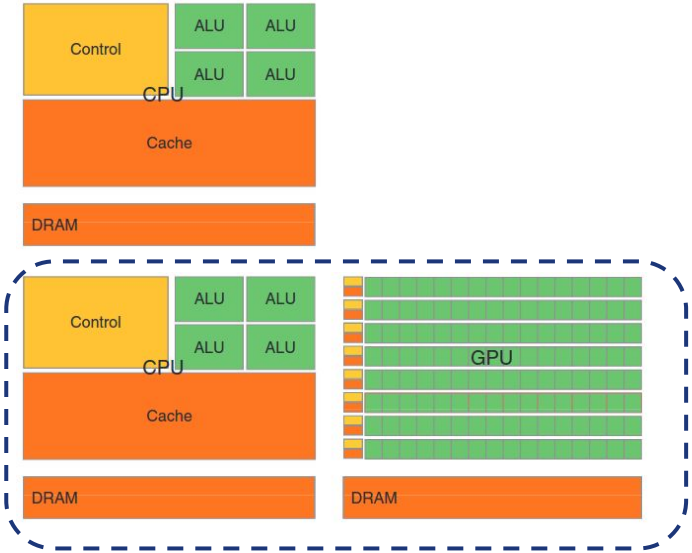
¿Qué es?

OpenACC (Open Accelerators) es un **modelo de programación** y un **lenguaje basado en directivas de compilación**.

Permite expresar el paralelismo en las aplicaciones por medio de **anotaciones en el código.**

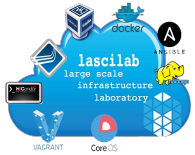
El programa paralelizado toma ventaja de los múltiples cores de una **GPU**.

¿Donde se usa?



Conozcamos nuestra CPU y GPU

OpenACC



¿Qué es?

OpenACC (Open Accelerators) es un **modelo de programación** y un **lenguaje basado en directivas de compilación**.

Permite expresar el paralelismo en las aplicaciones por medio de **anotaciones en el código.**

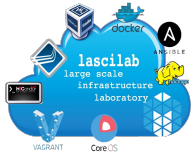
El programa paralelizado toma ventaja de los múltiples cores de una **GPU**.

¿Donde lo puedo encontrar?

Es una característica presente en los compiladores, puede ser habilitada usando banderas de compilación:

- **GNU gcc/g++**, **-fopenacc** (en desarrollo)
- **PGI compiler**, **-acc**
- Otros

OpenACC



¿Existen otras herramientas para paralelizar nuestras aplicaciones en GPU?

Cada lenguaje de programación posee un conjunto de herramientas para paralelizar nuestras aplicaciones.

Para el lenguaje de programación C/C++ vamos a usar **OpenACC** aunque existen más herramientas.

MATLAB
The Language of Technical Computing

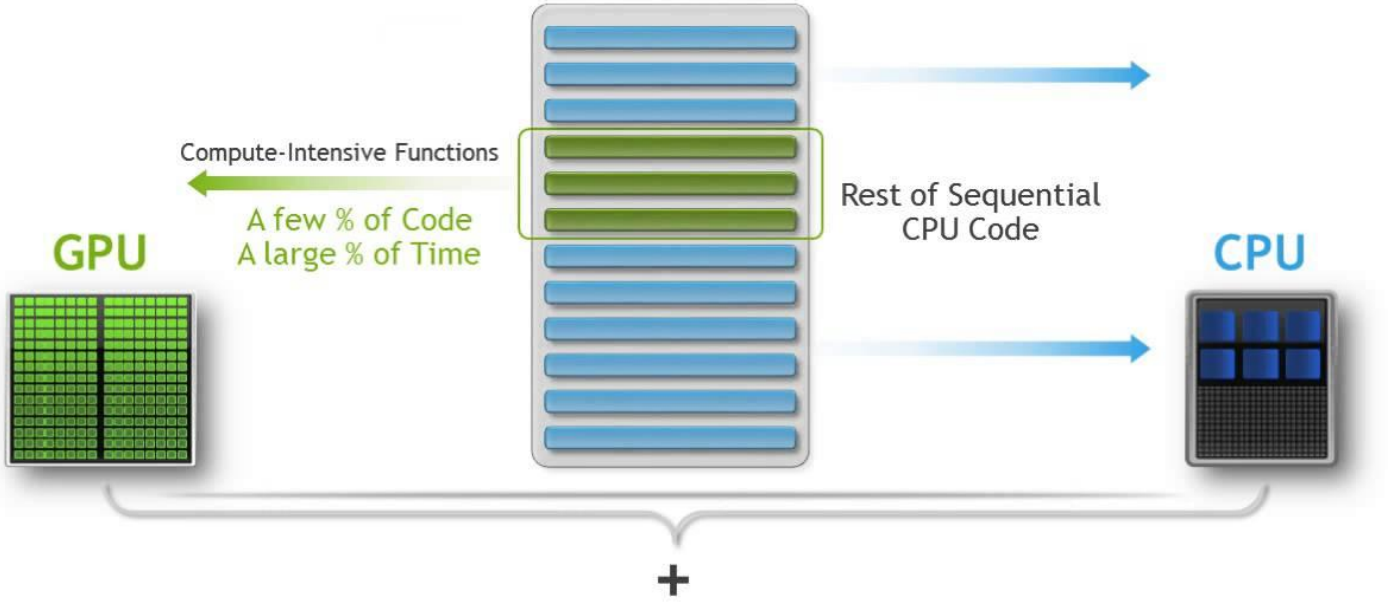
python™

NVIDIA
CUDA

OpenMP

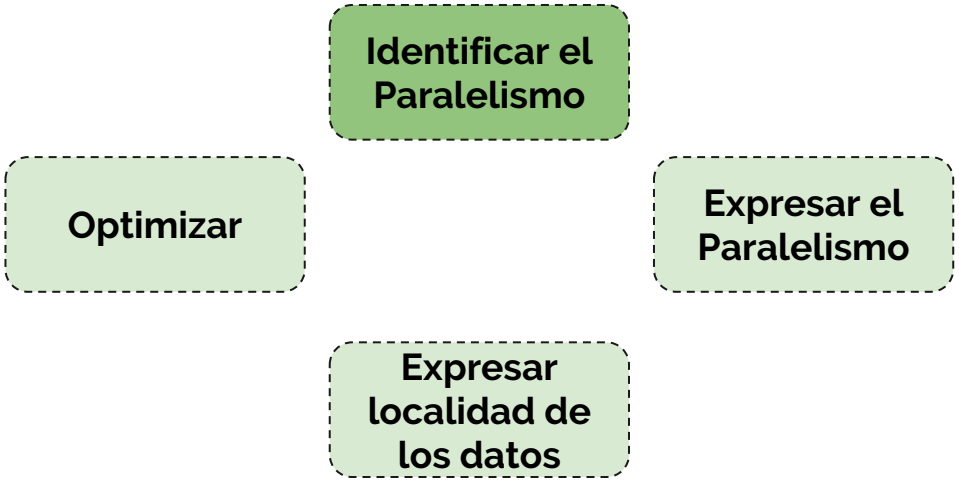
OpenACC

¿Cómo se ejecutaría un programa paralelizado en un CPU/GPU?



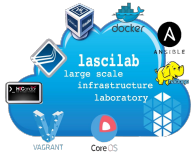
OpenACC

¿Qué pasos debo seguir para paralelizar mi aplicación en CPU/GPU?

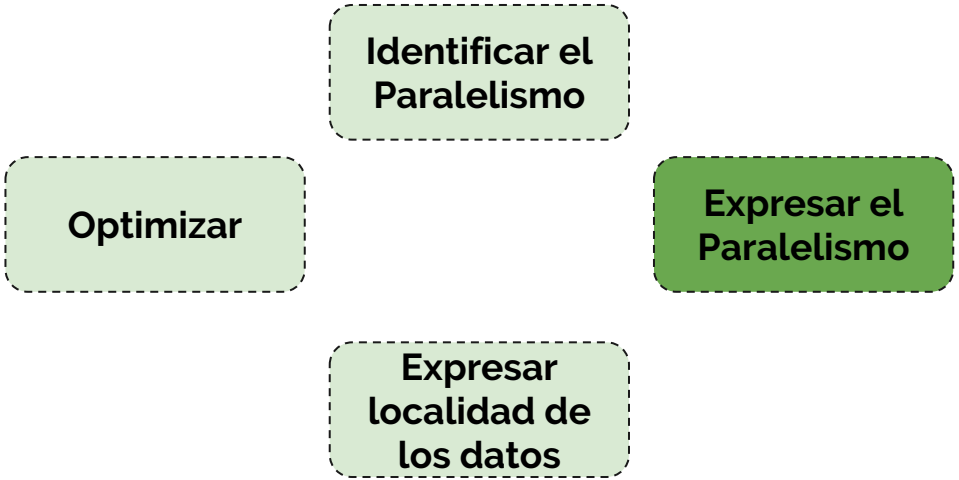


- “Identificar en qué partes se consume mayor tiempo de ejecución **(profiling).”**
- “Los ciclos (**loops**) con gran cantidad de iteraciones independientes son un excelente punto de partida”.

OpenACC



¿Qué pasos debo seguir para paralelizar mi aplicación en CPU/GPU?



- “Colocar **directivas OpenACC** en los ciclos identificados en el paso anterior.”
- “OpenACC consiste en brindar al compilador suficiente información para accelerar el código.”

Automatic Parallelization

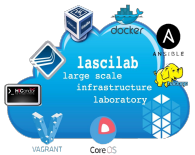
```
int main(void){
  #pragma acc kernels
  {
  for (...){
   for (...){
      ...
   }
  }
 }
 return 0;
}
```

Semi Automatic Parallelization

```
int main(void){
  ...
  #pragma acc kernels loop ...
  {
  for (...){
   #pragma acc loop ...
   for (...){
      ...
   }
  }
 }
  ...
  return 0;
}
```

Non Automatic Parallelization

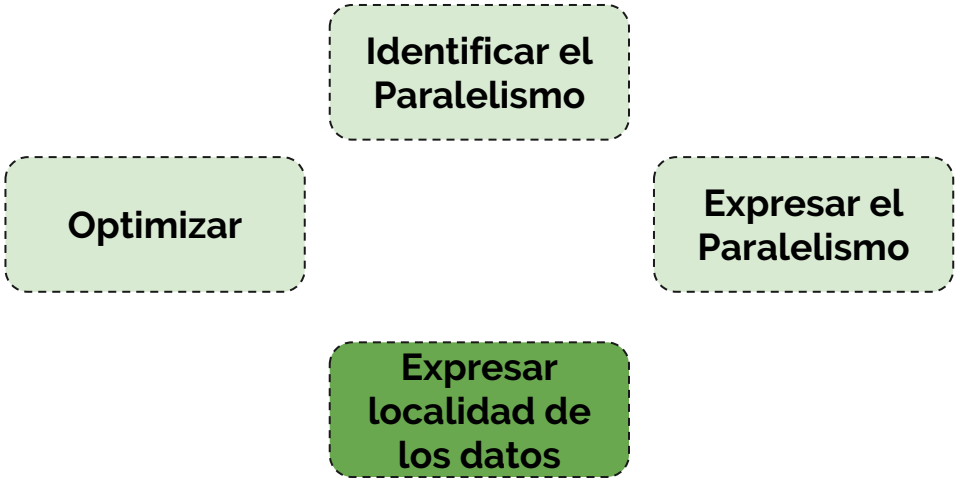
```
int main(void){
  ...
  #pragma acc parallel loop ...
  {
   for (...){
    #pragma acc loop ...
    for (...){
       ...
    }
   }
  }
  ...
  return 0;
}
```



Formas de **Expresar Paralelismo** en nuestras aplicaciones

OpenACC

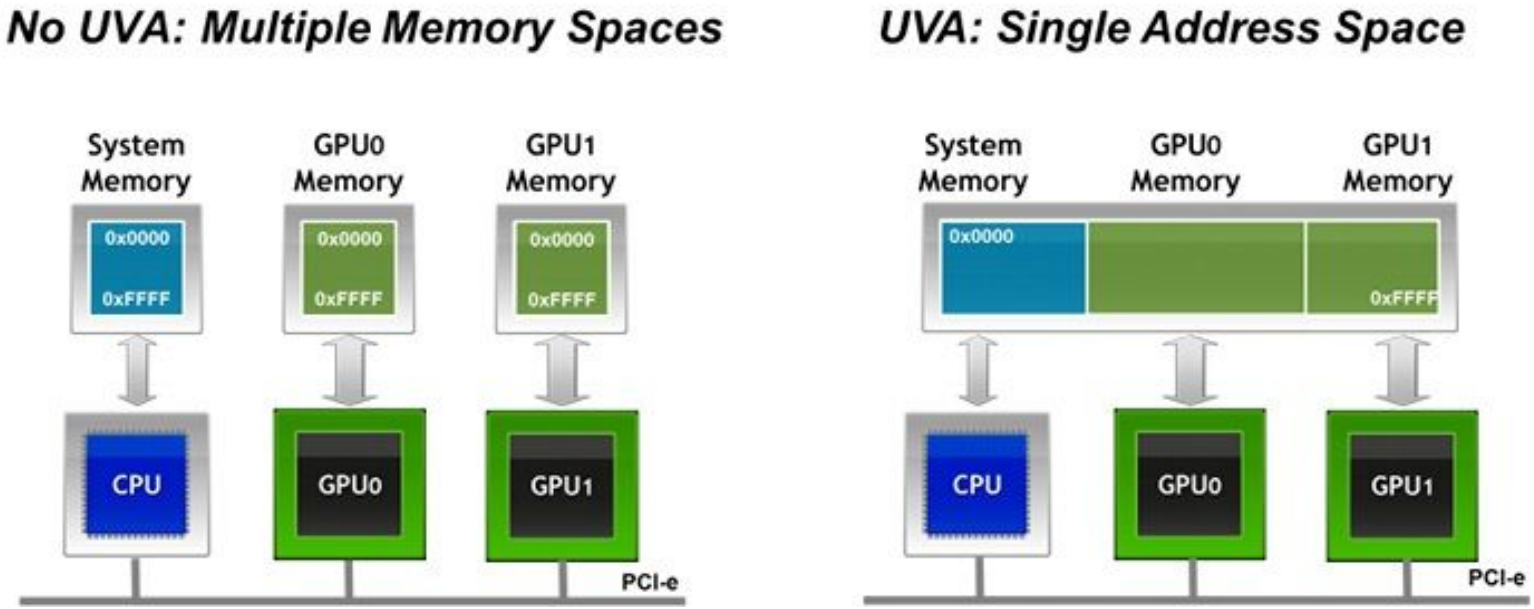
¿Qué pasos debo seguir para paralelizar mi aplicación en CPU/GPU?



- “El compilador necesita saber no solo el código a paralelizar, también necesita saber qué datos van a ser procesados.”
- **Los datos a ser procesados en GPU deben ser llevados a la GPU.**

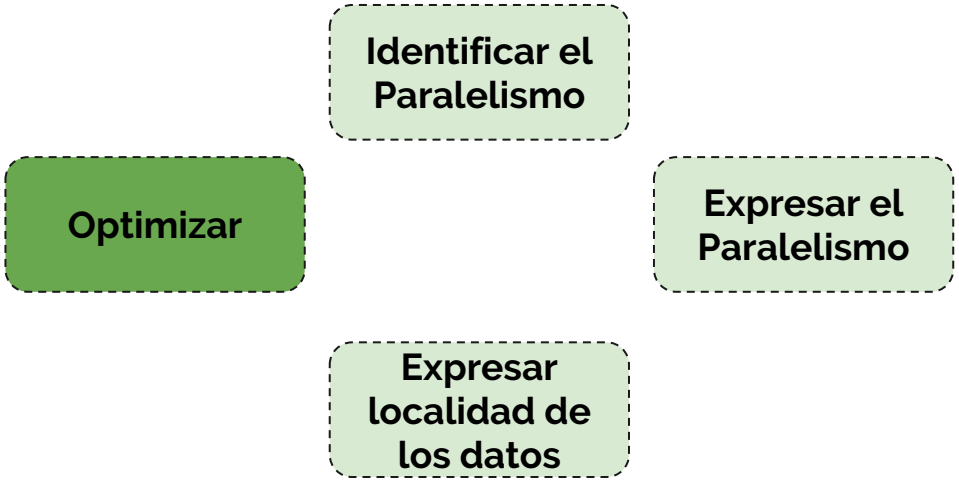
OpenACC

¿Qué pasos debo seguir para paralelizar mi aplicación en CPU/GPU?

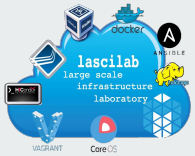


OpenACC

¿Qué pasos debo seguir para paralelizar mi aplicación en CPU/GPU?



- “El compilador usualmente hace un buen trabajo acelerando el código.”
- En la mayoría de los casos se puede obtener un mayor rendimiento dando más información de nuestro código al compilador.



Click Taller

- El taller es guiado, después de cada ejercicio se hará una retroalimentación.
- Los que terminan primero deben ayudar a sus compañeros.
- El taller sugiere preguntas para discutir durante la presentación.
- **IMPORTANTE: Si no entiende háganoslo saber**

Directiva “*Kernels*” - ejercicio

```
git clone git@github.com:DonAurelio/openacc-workshop.git

cd openacc-workshop/1.kernels
```

Directiva “*Parallel*” - demo - GoL

```
git clone git@github.com:DonAurelio/openacc-workshop.git
```

```
cd openacc-workshop/2.parallel/1.demo
```

Gracias ...