

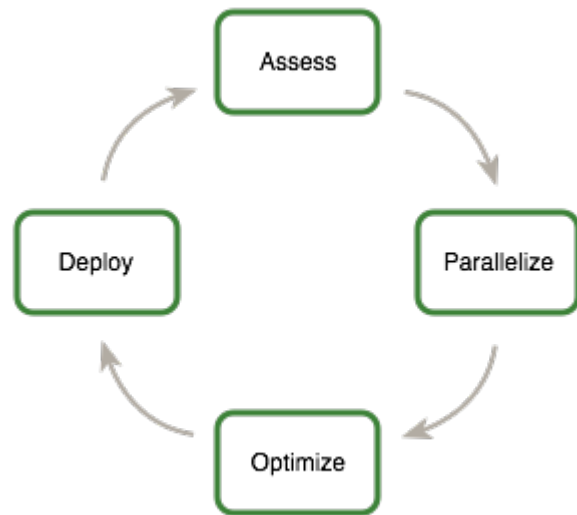
Programación paralela

APOD

José M. Cecilia

APOD

- ✓ APOD (*Assess, Parallelize, Optimize, Deploy*)
- ✓ Metodología de desarrollo para aplicaciones que se quieren acelerar.
- ✓ Ciclo de vida iterativo.
- ✓ 4 etapas: Evaluar, Paralelizar, Optimizar y Desplegar.



Evaluación

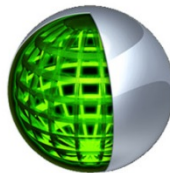
- ✓ Evaluar el **código objeto** de aceleración.
- ✓ Identificar dónde se consume la mayor parte del tiempo -> **Cuello de botella**.
- ✓ Ver si esa parte es **significativa** respecto al tiempo total de la aplicación.
- ✓ La evaluación es fundamental **en todos los pasos** de APOD.

Evaluación: Profilers



gprof

- ✓ Confirmar hipótesis.
- ✓ Usar múltiples **conjuntos heterogéneos** de datos.
- ✓ Usar cargas de **datos realistas**.
- ✓ Utilizar herramientas como **gprof** y **Vtune** para analizar el código.
- ✓ Temporizadores (**cudaEvent**) y **Nsight** para VS.
- ✓ El rendimiento de la aplicación cuando se ejecuta un profiling se puede ver afectado



NVIDIA®
Nsight™

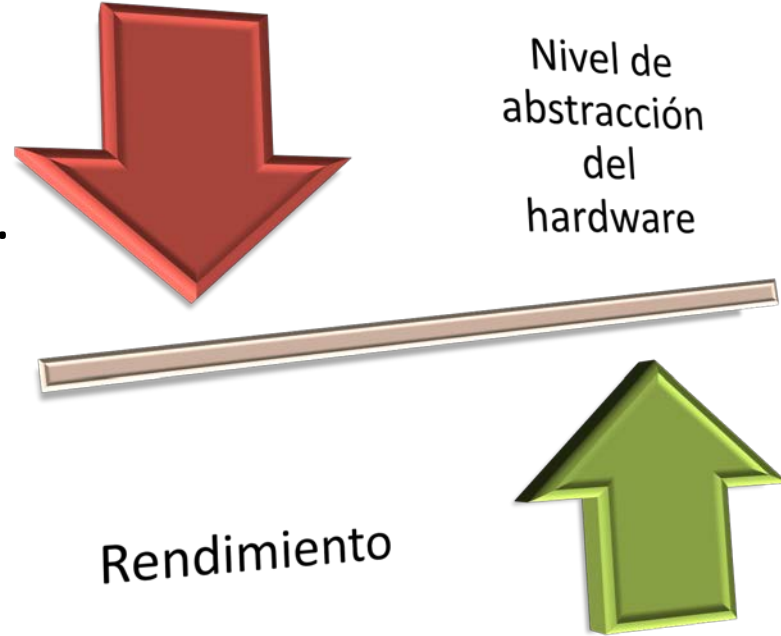


Escalado de las aplicaciones

- ✓ Si 1 core = 10 seg; ¿2 cores = 5 seg? ¿2x speed-up?
- ✓ Strong scaling o Ley de Amdahl -> Corto plazo
 - ¿Qué speed-up obtengo al añadir más procesadores?
- ✓ Weak scaling o Ley de Gustafson -> Largo plazo
 - ¿Qué speed-up obtengo al añadir más procesadores pero si vario las condiciones de entrada del problema?

Paralelización

- ✓ Elegir el **lenguaje de programación** (OpenMP, MPI, CUDA,...) -> depende de la plataforma objetivo.
- ✓ Paralelizar:
 - Usar librerías (BLAS, CUDA, FFT, cuSPARSE)
 - Usar compiladores paralelizantes (-O3, --Parallel).
 - Paralelización utilizando el lenguaje nativo.



Optimizar

- ✓ Todo es rendimiento -> **Medir rendimiento.**
- ✓ Seguir **prácticas adecuadas:**
 - Elección de algoritmos, movimiento de datos, conocimiento del hardware.
- ✓ Utilizar **profilers.**

Desplegar

- ✓ Ser evolucionario y no revolucionario.
- ✓ Obtener beneficios pronto y notables.
- ✓ Codificar pensando en los errores.

Información de depuración

GRACIAS