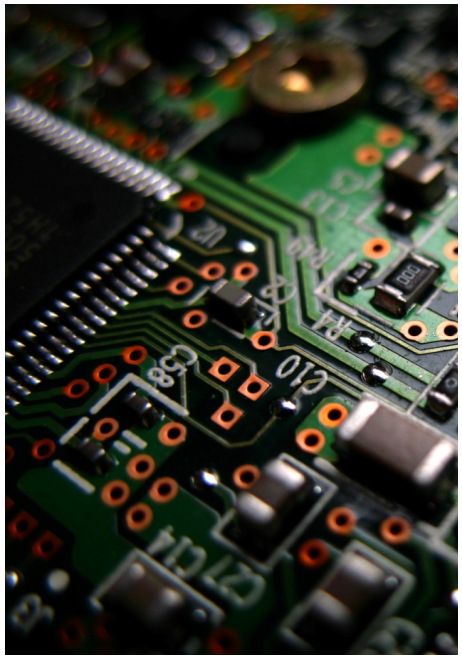
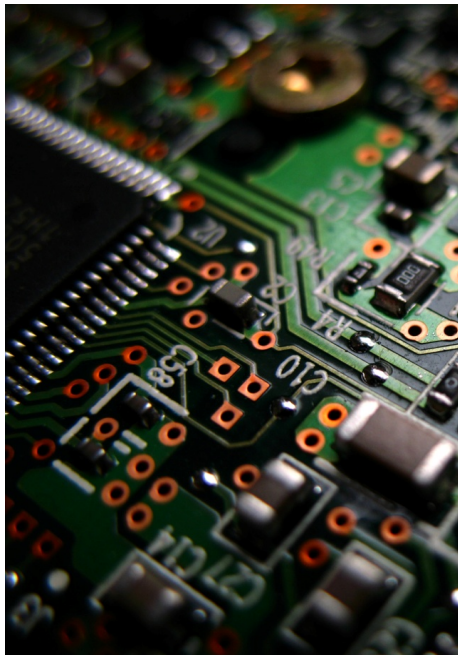
- Computación Paralela

César Pedraza Bonilla

Índice de Contenidos



Índice de Contenidos



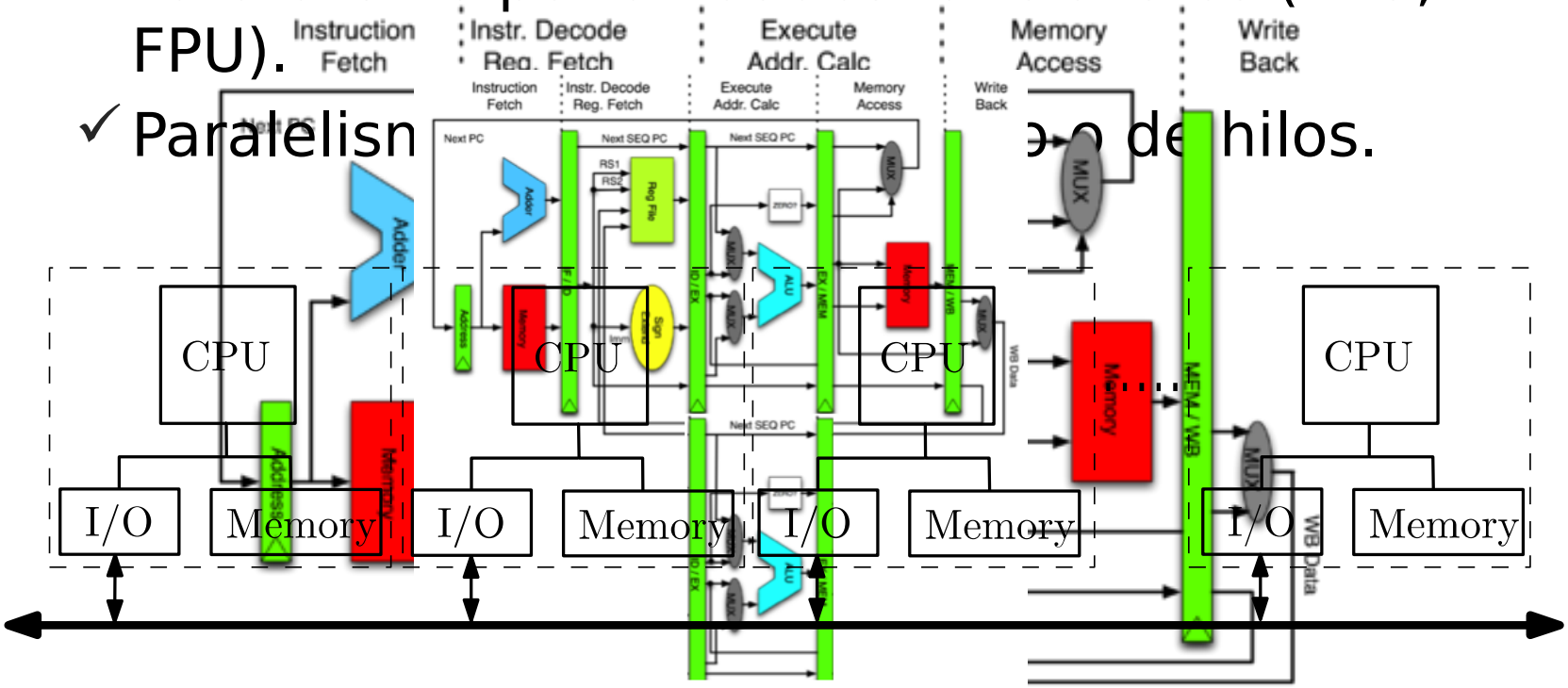
Introducción

- ¿Qué es computación paralela?
  - Es una técnica de la computación en la que se realizan operaciones de forma concurrente.
  - Se requieren múltiples unidades de procesamiento.
  - Funciona bajo la premisa de que algunas tareas pueden ser divididas en otras más pequeñas que se ejecuten de forma concurrente.

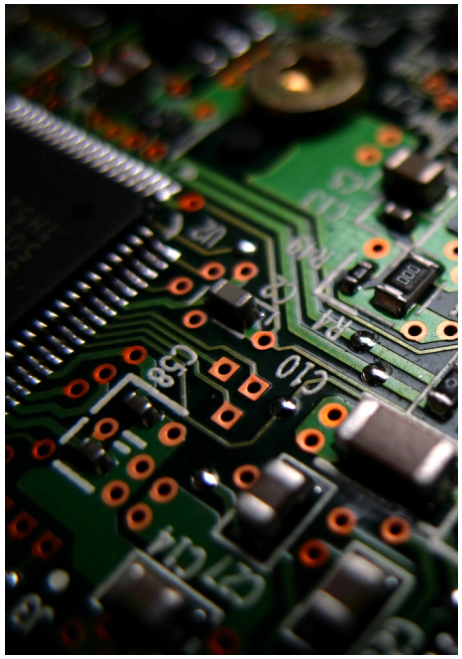
- ¿Por qué la computación paralela?
  - No se pueden aumentar las frecuencias de reloj de los procesadores actuales.
  - No se puede aumentar el consumo de potencia.
  - Se puede aumentar la densidad de transistores de los chips.

Introducción

- ✓ Tipos de paralelismo:
  - ✓ Paralelismo de bits.
  - ✓ Paralelismo por segmentación (segmentación).
  - ✓ Paralelismo por unidades funcionales (ALU, FPU).
  - ✓ Paralelisn[illegible] de hilos.



Índice de Contenidos

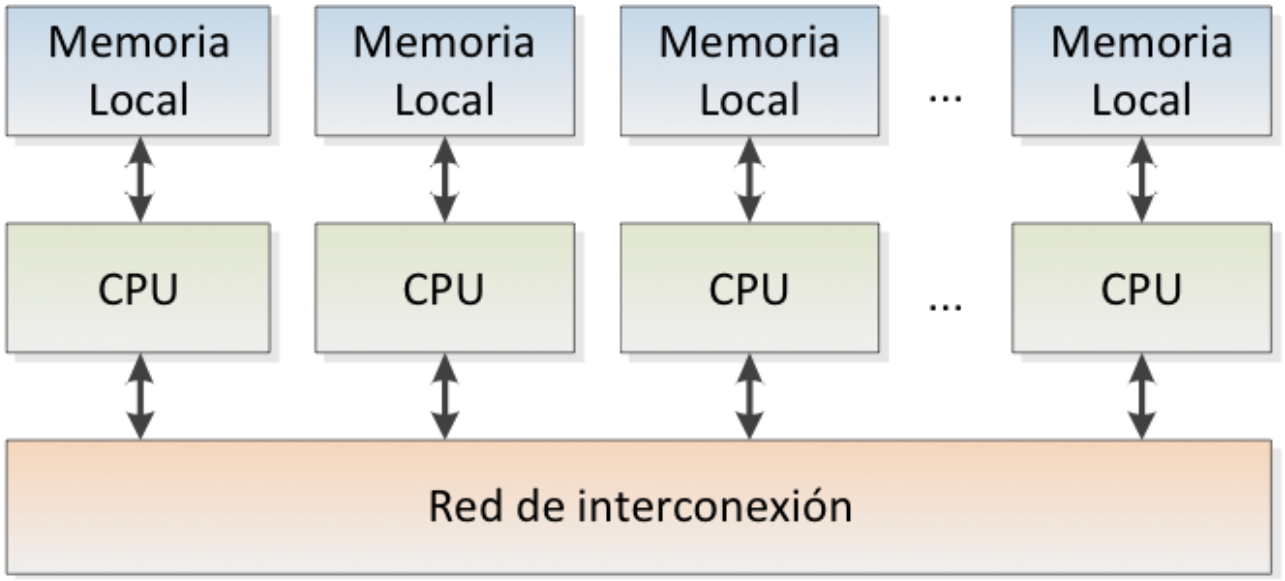


Arquitecturas paralelas

- Arquitecturas para computación paralela:
  - Memoria distribuida. Múltiples procesadores con su propia memoria física.
  - Memoria compartida. Múltiples procesadores comparten memoria física.
  - Heterogéneas.
    - Múltiples procesadores con memoria virtual compartida.
    - Múltiples procesadores con memoria compartida y distribuida.

Arquitecturas paralelas

- **Arquitecturas de memoria distribuida.**
  - Nodos conectados mediante una red de datos,
  - Cada nodo es una unidad con procesador, memoria y periféricos.



Arquitecturas paralelas

- **Arquitecturas de memoria distribuida.**
  - Clusters. Colección de computadores que se encuentran interconectados mediante redes de alta velocidad (Ethernet, SCI, Myrinet, Infiniband).

Arquitecturas paralelas

- **Arquitecturas de memoria distribuida.**
  - Clúster: Máquinas con procesador, memoria, s.o. independientes.

Pj. Summit IBM

- 4,608 nodos, 42TFlops c/u
- 9216 IBM POWER9 22C 3.07GHz
- 27648 NVIDIA Volta GV100
- 2,282,544 cores
- 122,300 TFlop/s (122,3 PFlops)
- Linux RHEL 7.4.
- Spectrum MPI

Linux

Arquitecturas paralelas

- **Arquitecturas de memoria distribuida.**
  - Clúster: Máquinas con procesador, memoria, s.o. indepen

Pj. Sunway TaihuLight

- 40.960 procesadores RISC SW26010
- 256+4 núcleos por procesador
- 10,649,600 cores
- 93,014.6 TFlop/s (93PFlops)
- Linux Sunway Raise OS.
- MPICH2

Linux

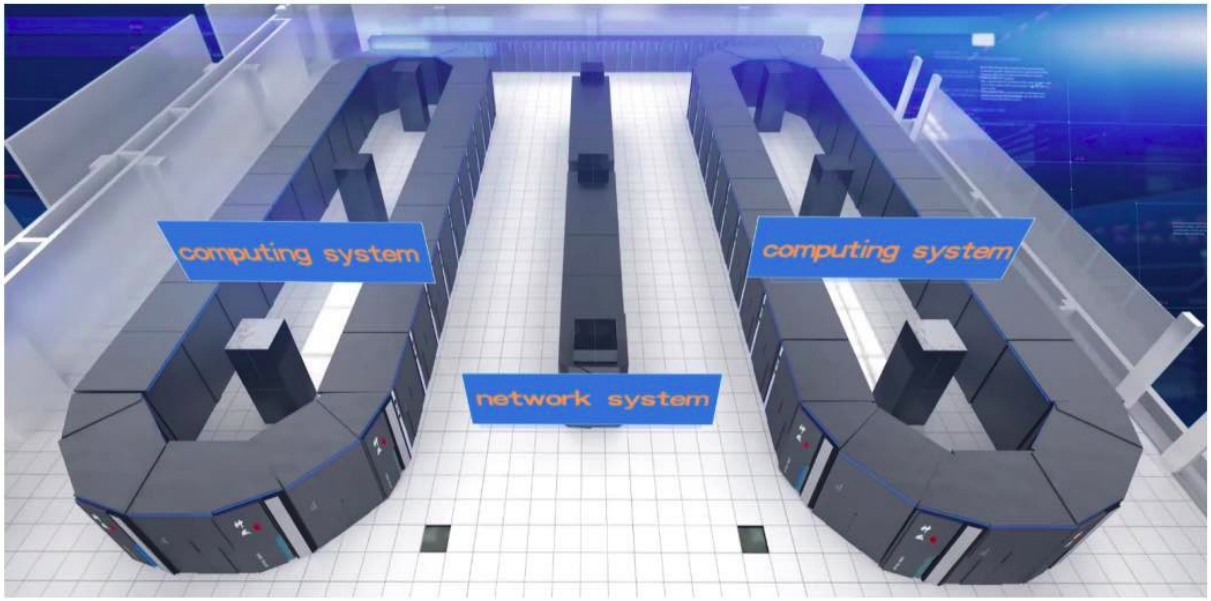
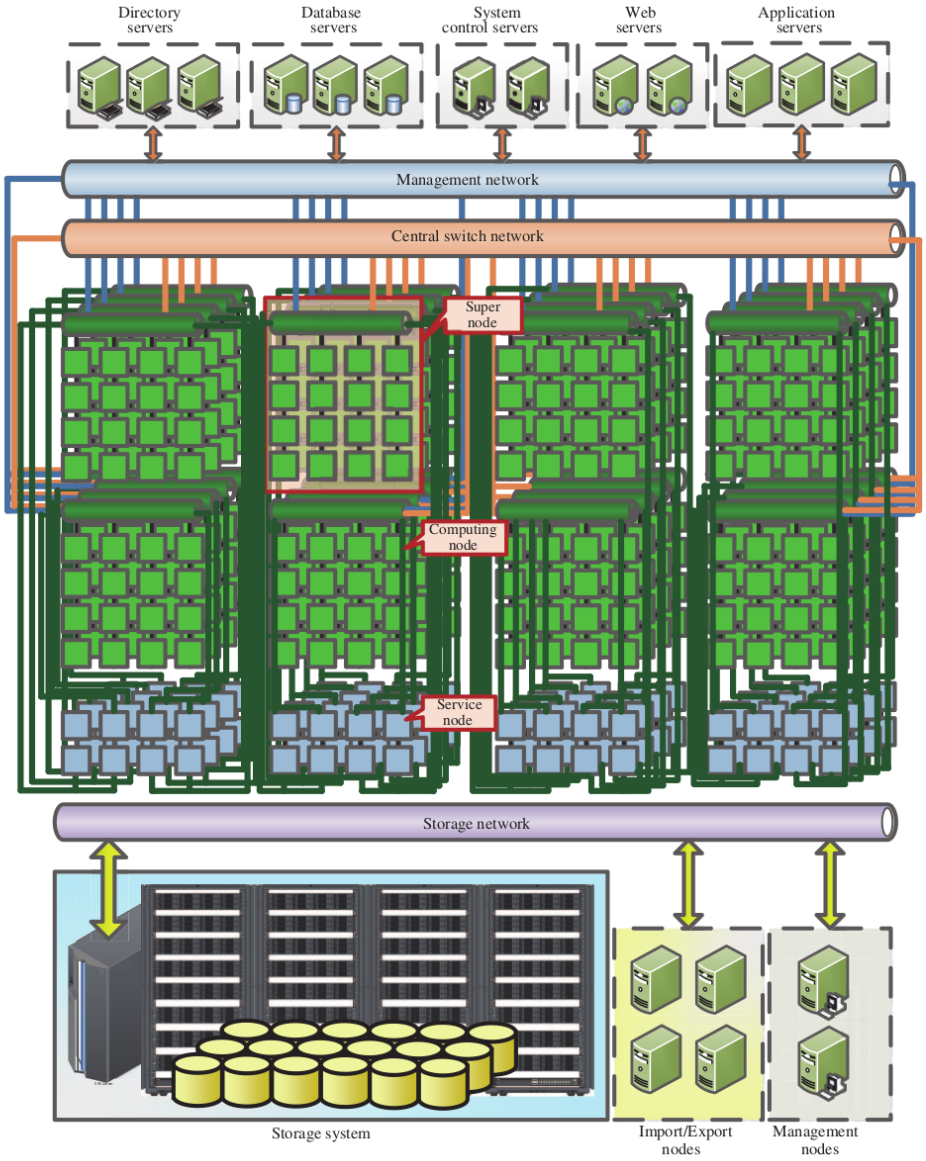


Figure 4: Overview of the Sunway TaihuLight System

Arquitecturas paralelas

Sunway cluster.

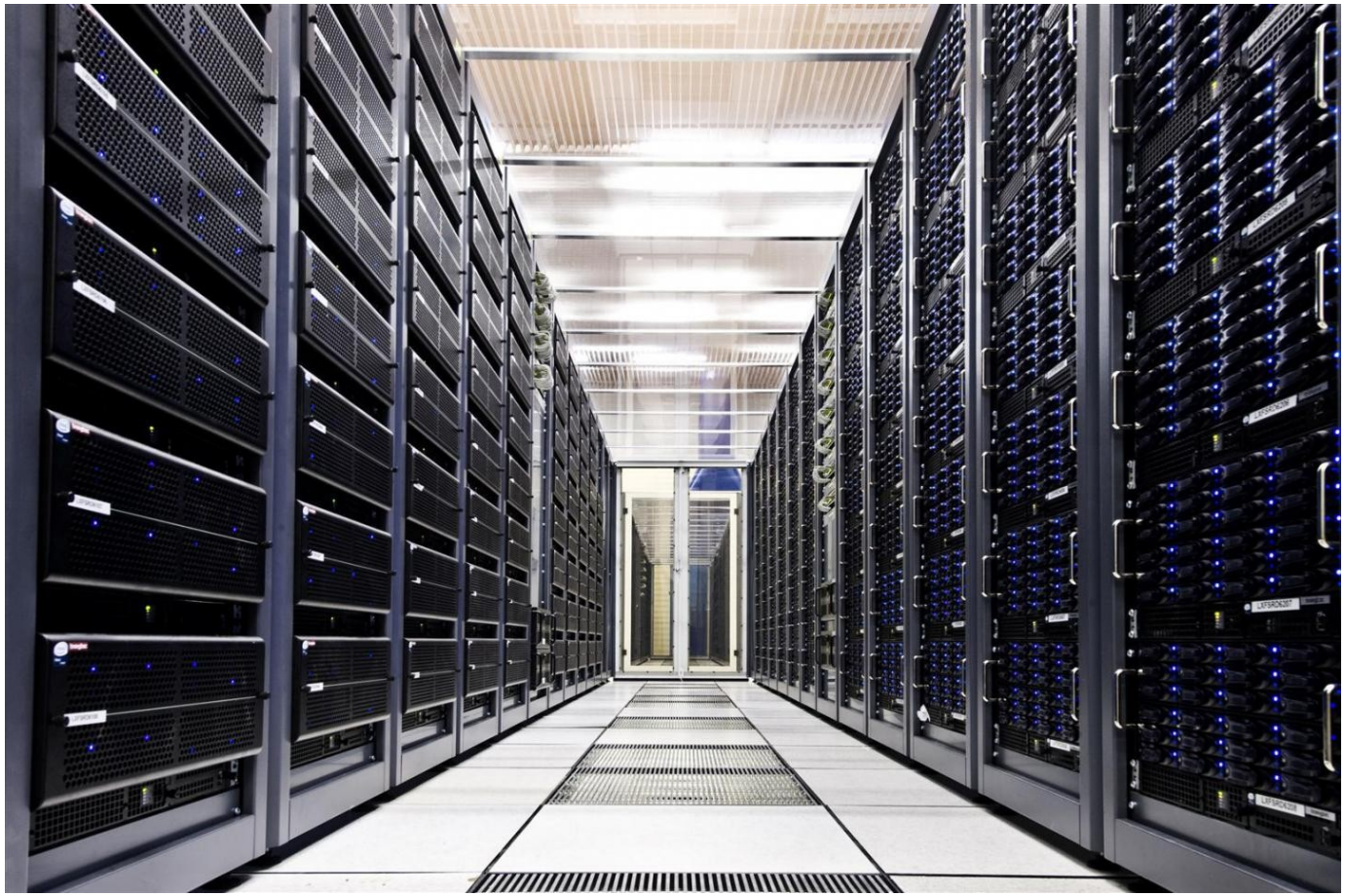


Arquitecturas paralelas

- **Arquitecturas de memoria distribuida.**
  - Grids. Computadores de múltiples dominios administrativos conectados para solucionar una tarea determinada.
  - Múltiples arquitecturas de computador.
  - Múltiples sistemas operativos.
  - Middleware. Globus toolkit.

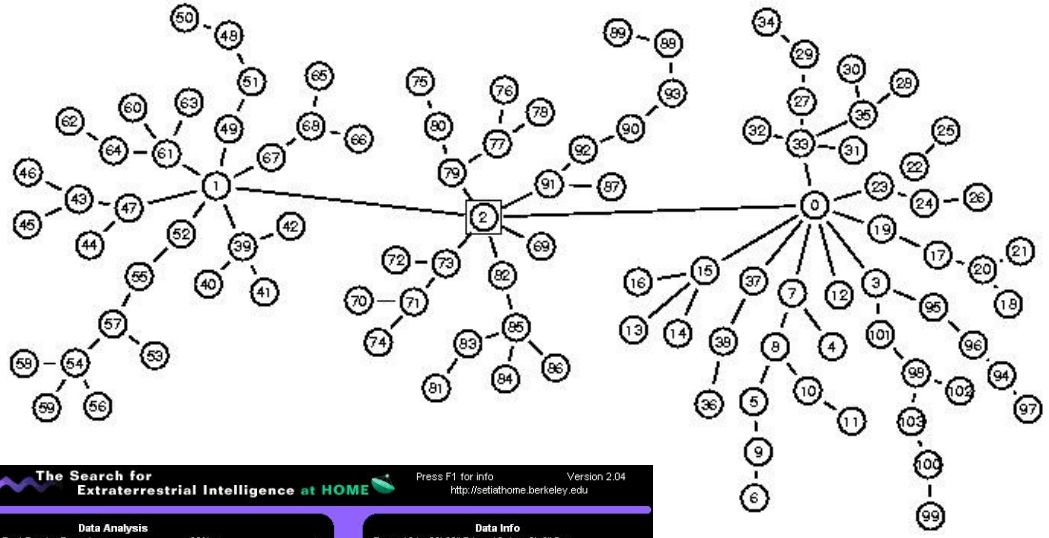
Arquitecturas paralelas

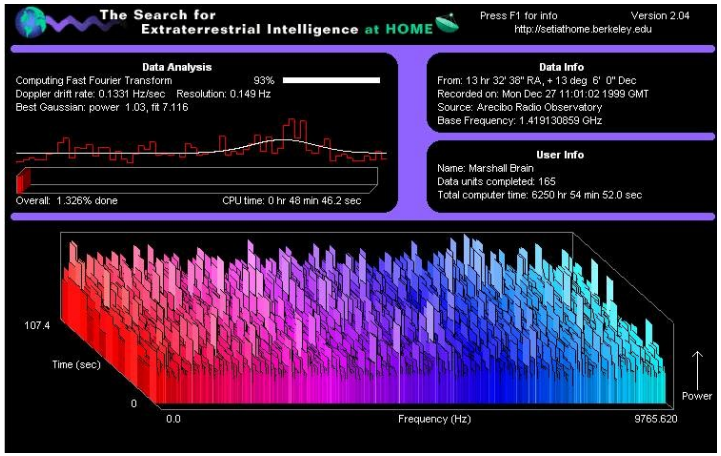
The Worldwide LHC Computing Grid



Arquitecturas paralelas

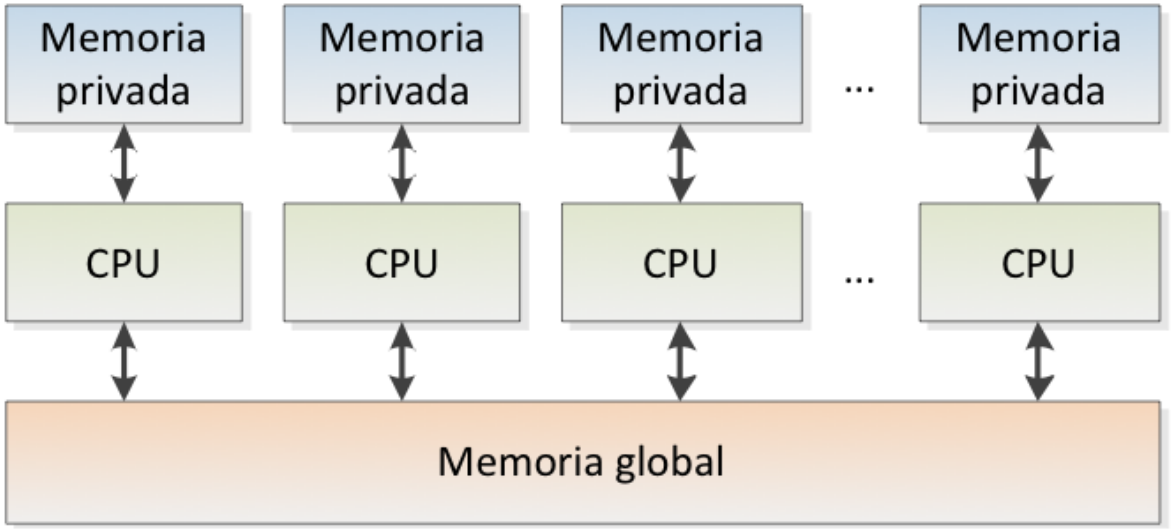
- **Arquitecturas de memoria distribuida.**
  - Grids.





Arquitecturas paralelas

- **Arquitecturas de memoria compartida.**
  - Son sistemas de múltiples núcleos de procesamiento que comparten cierta cantidad de memoria física (memoria global)



Arquitecturas paralelas

- **Arquitecturas de memoria compartida.**
  - GPU. (Graphics Processing Unit)
    - Es una arquitectura dedicada para el procesamiento gráfico.
    - Se encuentra basada en el concepto de múltiples unidades de procesamiento en un mismo chip.
    - Las comunicaciones se realizan mediante memorias compartidas.

Arquitecturas paralelas

AMD Radeon Instinct

- 2304 cores
- 12,3 TFlops

GeForce GTX 1080

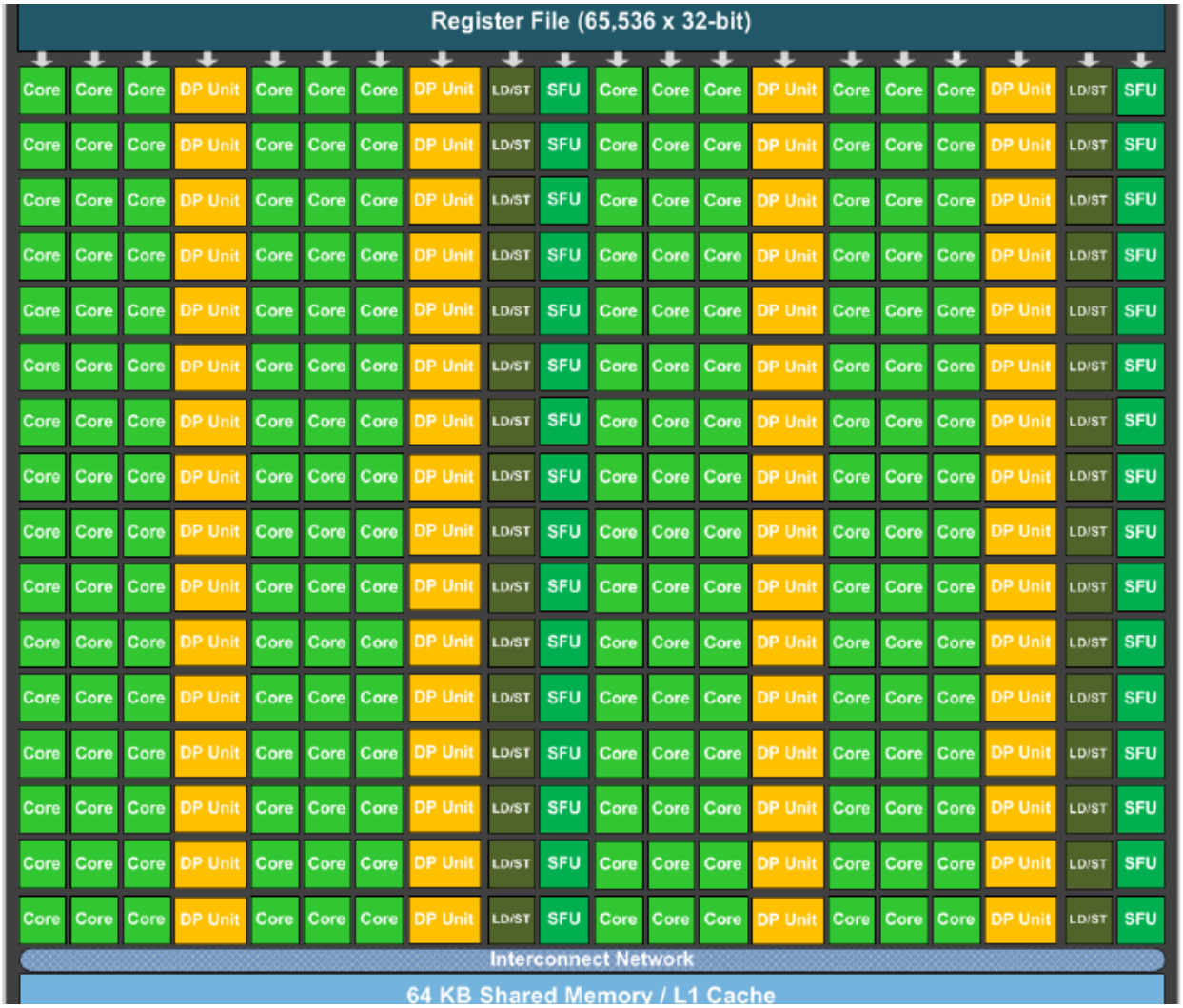
- Núcleos CUDA 3584
- 332 GFlops single
- 10609 GFlops double

Volta V100

5120 CUDA cores. 7 Tflops (double), 15.7 (single) Tflops

Arquitecturas paralelas

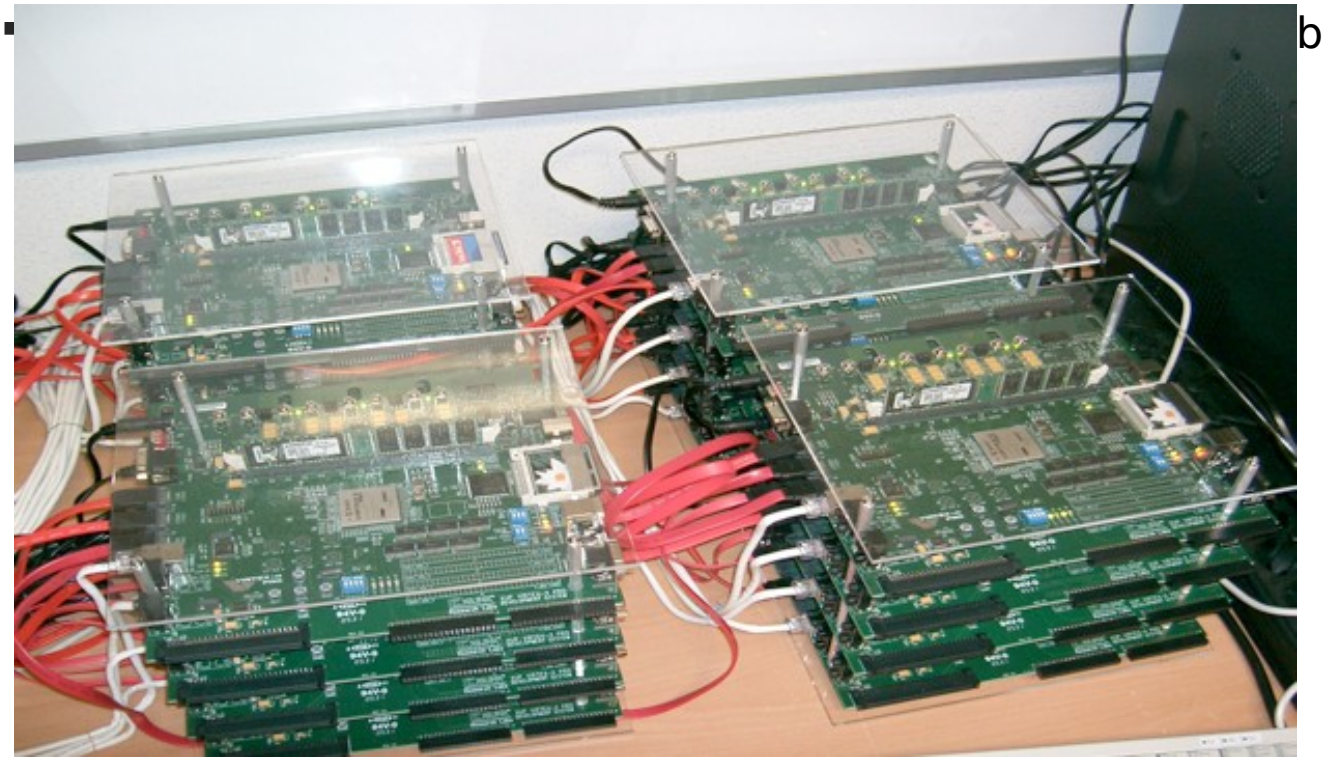
- SMX: 192 single-precision CUDA cores
- 64 double-precision units.
- 32 special function units (SFU)
- 32 load store units (LD/ST).



NVIDIA, Kepler SMX

Arquitecturas paralelas

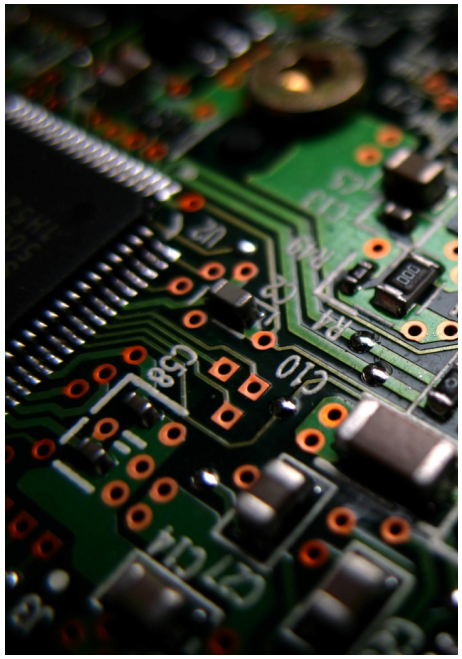
- **Arquitecturas de memoria heterogéneas.**
  - Mezclan el concepto de memoria compartida y distribuida.
    - bles, etc.



Arquitecturas paralelas

- **Arquitecturas de memoria heterogéneas.**
  - Cloud. Computación en la nube.
    - Aplicaciones que se pueden ejecutar en máquinas virtuales conectadas a internet.
    - Sistema paralelo y distribuido de máquinas virtuales.
    - Actualmente se ofrecen sistemas de cómputo paralelo.
    - Amazon Elastic Compute Cloud EC2
    - Google Compute Engine.

Índice de Contenidos

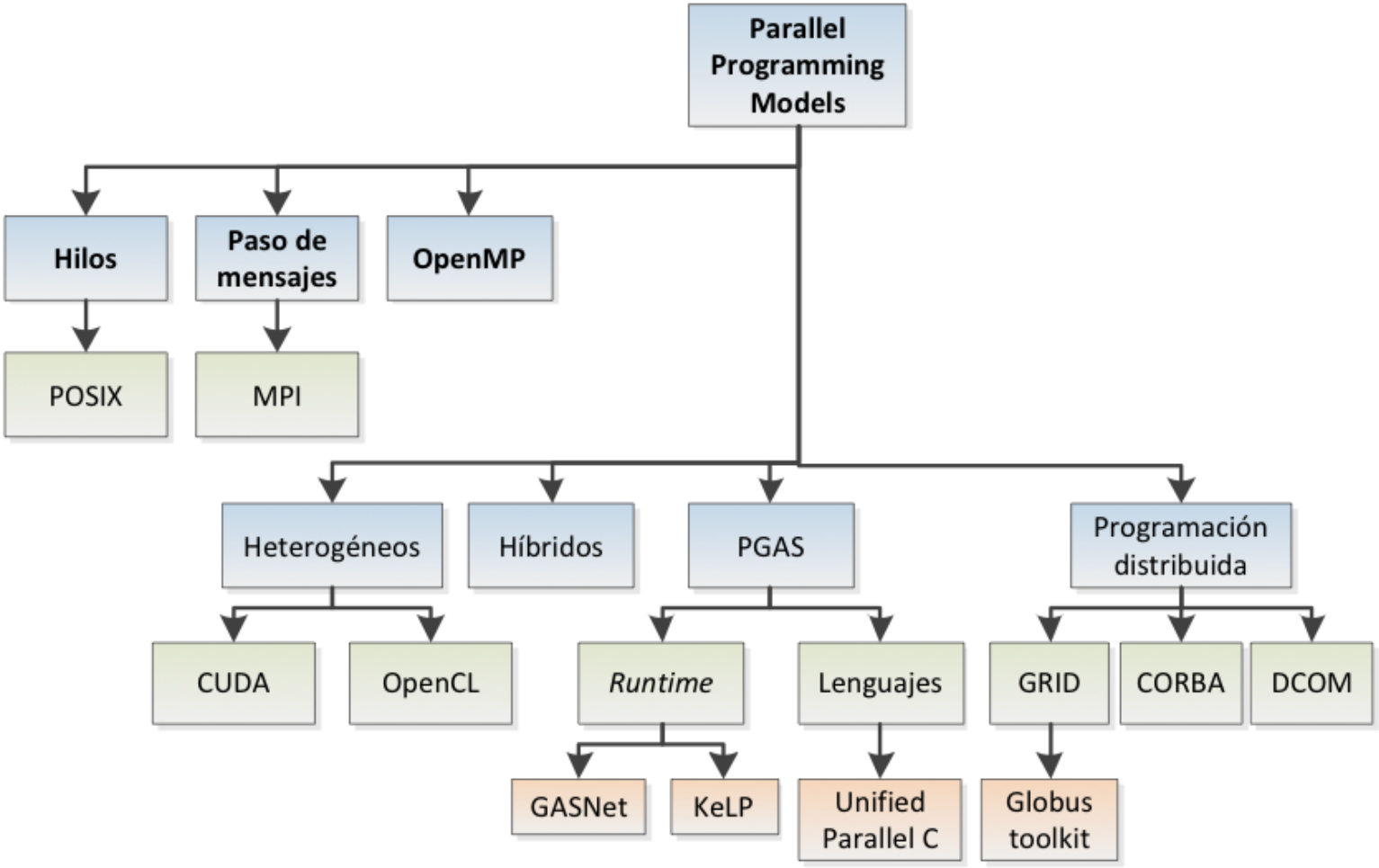


Programación paralela

- Auto paralelismo.
  - Los programas secuenciales son automáticamente paralelizados a nivel de instrucción. Se hace uso de un compilador.
- Programación paralela.
  - El programador realiza pasos para la paralelización de las tareas y asigna tareas a unidades de procesamiento.

Programación paralela

- **Programación paralela.**



Programación paralela

- **Hilos**
  - **Son unidades de procesamiento que comparten segmentos de memoria de datos, archivos abiertos y otros recursos.**
  - **Procesos ligeros.**
  - **POSIX -> Pthreads.**
  - **Son difíciles de mantener cuando el programa es complejo.**

Programación paralela

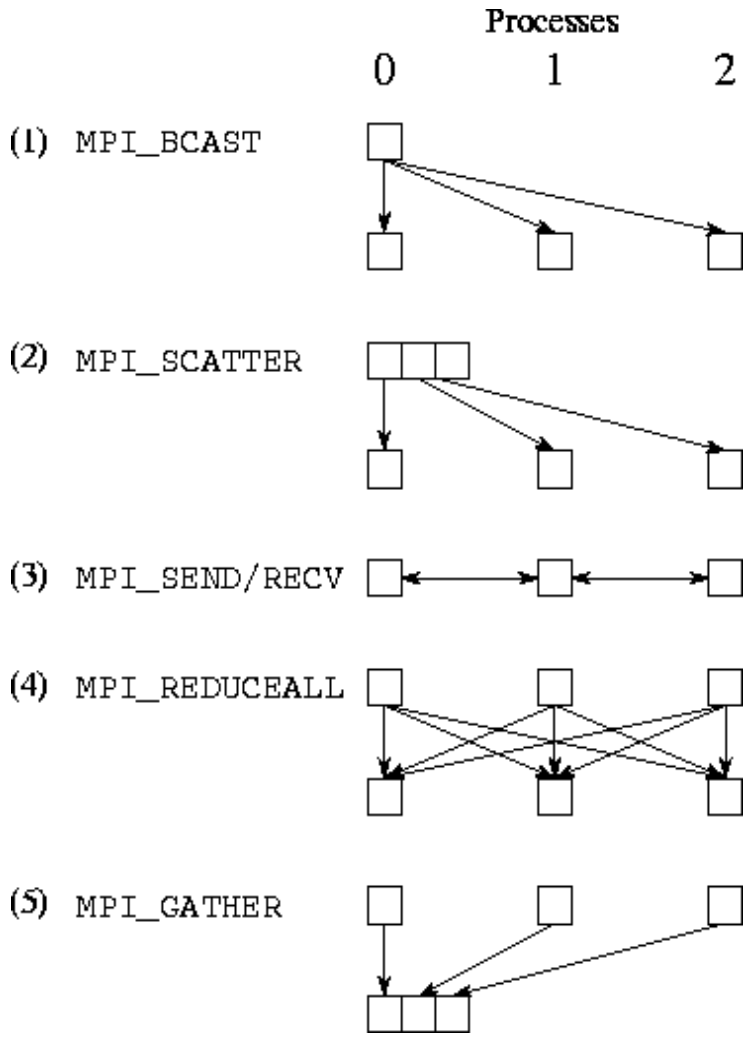
- **OpenMP. Última versión Nov 2015.**
  - **Es un modelo basado en hilos.**
  - **API. Comprende directivas para compilador.**
  - **La paralelización se hace mediante el modelo fork() join(). PARALLEL; END PARALLEL;**

```
#include <stdio.h>

int main(void)
{
  #pragma omp parallel
    printf("Hello, world.\n");
  return 0;
}
```

Programación paralela

- **Paso de mensajes.**
  - **OpenMPI.** **Última versión Sept 2017** Es una librería para simplificar los procesos de las comunicaciones entre nodos o máquinas en una plataforma paralela basada en cluster.

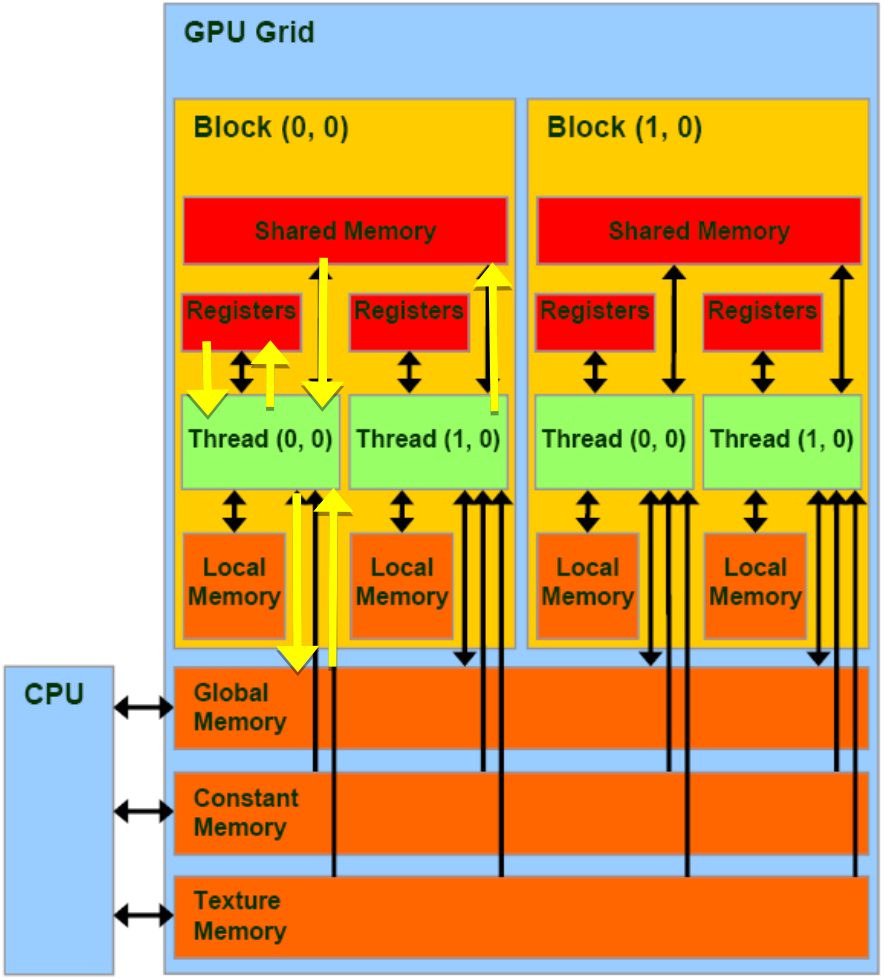


Programación paralela

- **CUDA**. **Última versión 9.1 dic 2017**
  - Es una plataforma de software para el desarrollo de aplicaciones de propósito general sobre arquitecturas basadas en GPUs de NVIDIA.
  - Librerías, compiladores etc.

Programación paralela

- **CUDA**.
  - Jerarquía de memoria: se distinguen tres tipos de memoria:
  - Registros: La más rápida y cercana a cada core.
  - Compartida. Común para los cores de un bloque y que permite las comunicaciones entre estos.
  - Global: es la que permite comunicarse con el sistema CPU y con cualquier core de la GPU.



Programación paralela

- **CUDA**.
  - Jerarquía de hil

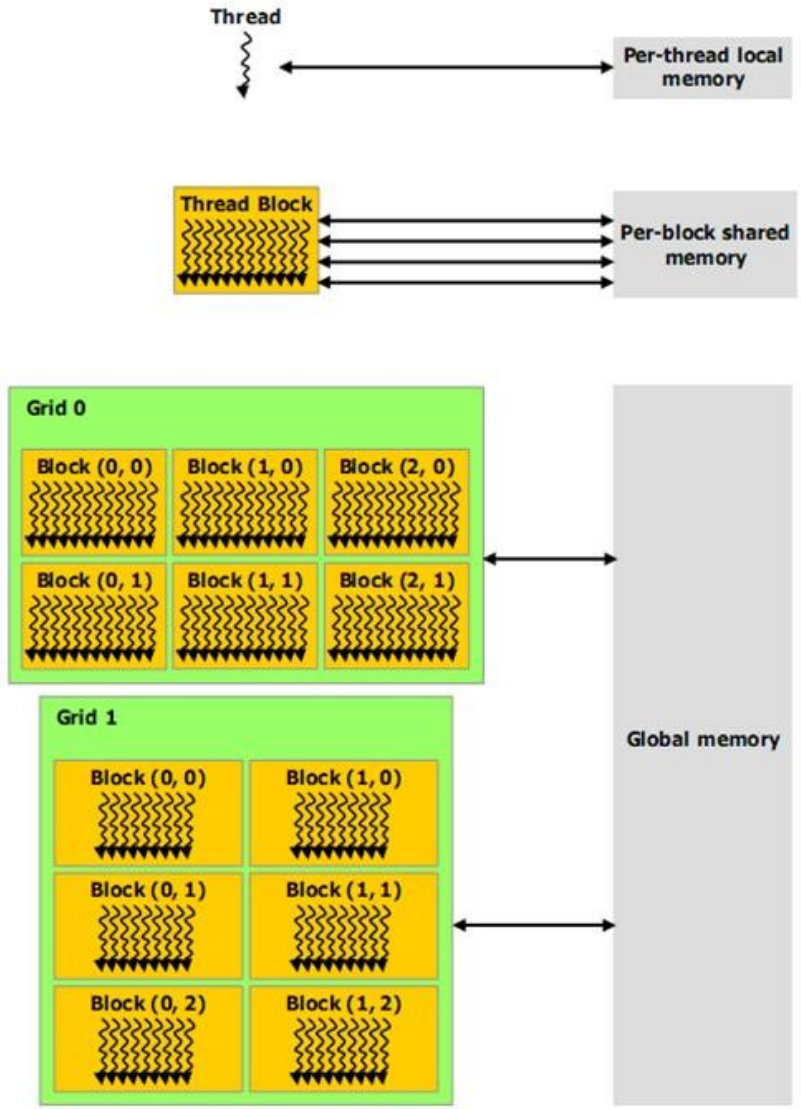


Figure 11 Cuda Memory

Programación paralela

- **CUDA**.
  - Programa paralelo:

CPU program

```
void increment_cpu(float *a, float b, int N)
{

    for (int idx = 0; idx<N; idx++)
        a[idx] = a[idx] + b;
}
```

CUDA program

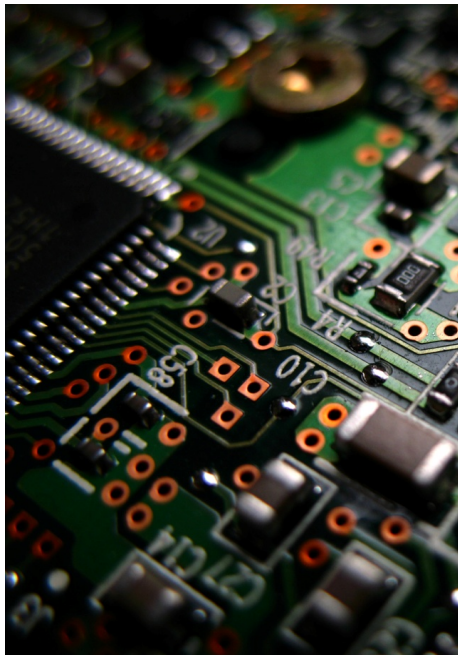
```
__global__ void increment_gpu(float *a, float b, int N)
{
    int idx = blockIdx.x * blockDim.x + threadIdx.x;
    if (idx < N)
        a[idx] = a[idx] + b;
}
```

kernel

Programación paralela

- **OpenCL.** **Última versión 2.0 Julio 2013.**
  - Conjunto de herramientas para desarrollar programas para sistemas basados en CPUs, GPUs y DSPs.
  - Basado en *kernels* con C99.

Índice de Contenidos



Metodologías de diseño

- ¿Cómo se diseña un programa para una arquitectura paralela?
  - No todos los algoritmos o programas computacionales son paralelizables!
  - No todos los algoritmos funcionan mejor en una plataforma paralela.
  - Existen metodologías para el diseño de un programa paralelo.
    - Metodología Foster.

Metodologías de diseño

Método de diseño Foster.

- ✓ Particionamiento:
  En el dominio de los datos o de funciones.

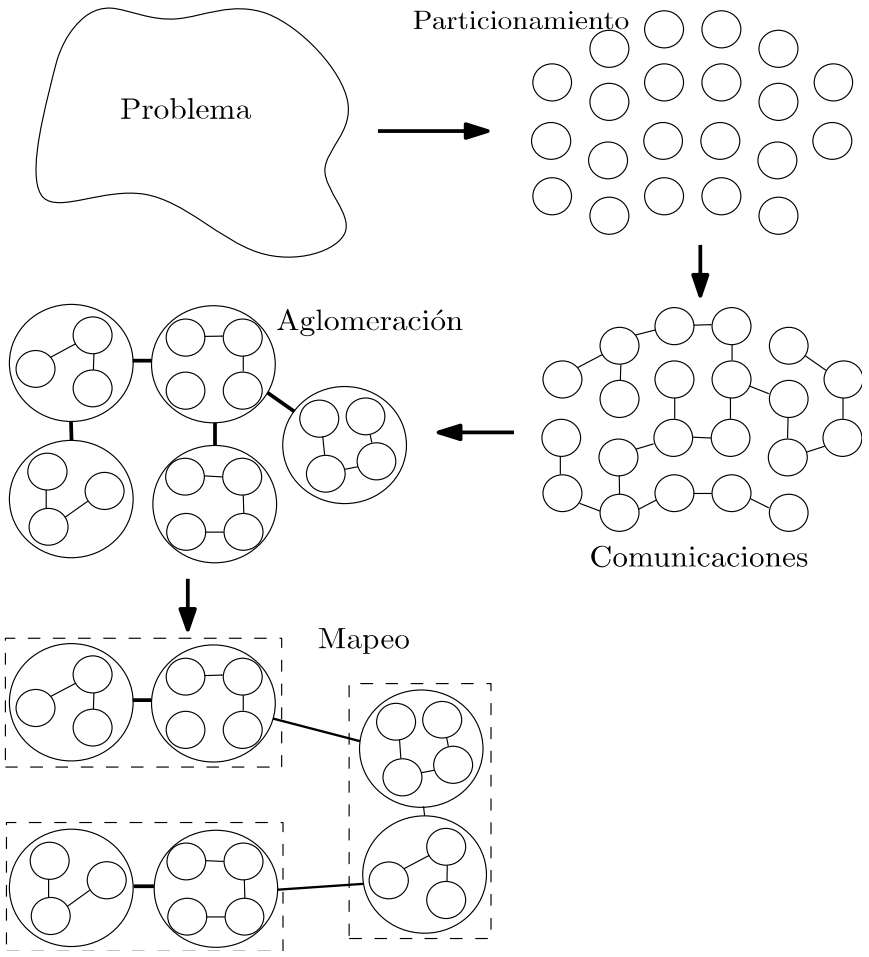
- ✓ Comunicaciones:
  Distintos medios o paradigmas:
  Memoria.
  Paso de mensajes.

  - ✓ Aglomeración:

  Tareas o datos son agrupados teniendo en cuenta posibles dependencias.

  - ✓ Mapeo:

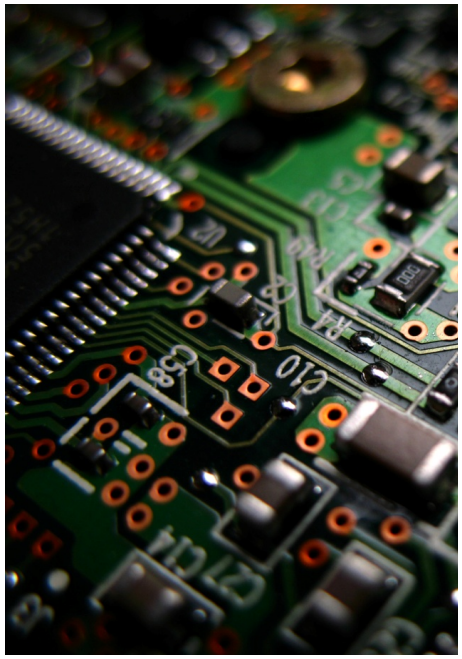
  Los grupos son asignados a una máquina o core.



Metodologías de diseño

- Cuando se diseña un algoritmo paralelo es necesario tener en cuenta:
  - Los tiempos de las comunicaciones.
  - Maximizar el procesamiento en cada nodo o unidad de procesamiento.
  - Los costes de implementar el algoritmo.
  - Tiempos de planificación (scheduler).

Índice de Contenidos



Tendencias

- ✓ Sistemas híbridos. Producir software que haga transparente la programación en sistemas híbridos.
- ✓ Software de desarrollo para clústeres más sencillo.
- ✓ Sistemas híbridos, múltiples plataformas. Computadores de escritorio, tabletas, móviles, entre otros.
- ✓ Confiabilidad y tolerancia a fallos.
- ✓ Computación en la nube. Unificación de APIs ó integración. Seguridad.
- ✓ Reducir los tiempos de comunicaciones en las arquitecturas. Diseñar programas limitando las comunicaciones.