SISTEMAS PARALELOS

Clase 9 - Tendencias en HPC

FACULTAD DE INFORMATICA

UNIVERSIDAD
NACIONAL
DE LA PLATA

Agenda de la clase anterior

- Diseño de algoritmos paralelos
  - Etapa de descomposición de tareas
  - Etapa de mapeo de tareas a procesadores
  - Métodos para reducir overhead de las interacciones
- Modelos de algoritmos paralelos

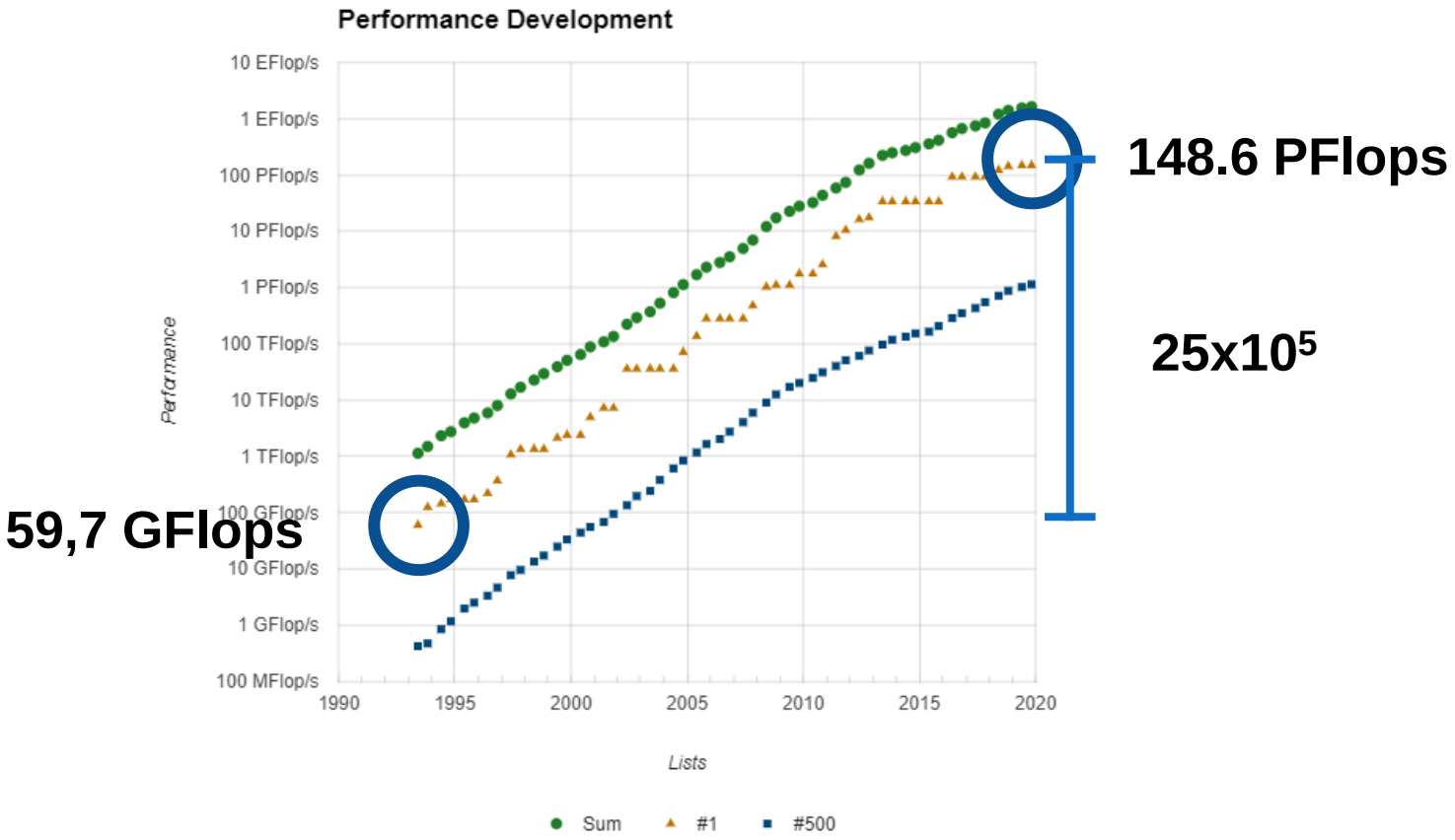
Agenda de esta clase

- Tendencias en HPC

TENDENCIAS EN HPC

Tendencias en HPC

- El objetivo de HPC ha sido incrementar el rendimiento y ocasionalmente el cociente precio/rendimiento



www.top500.org

Tendencias en HPC

- Notable incremento en el consumo de energía eléctrica

Summit
10.1MW
US$ 17200000

Sierra
7.4MW
US$ 12600000

Sunway TaihuLight
15.3MW
US$ 26000000

Tianhe-2A
18.4MW
US$ 31300000

www.top500.org

- Representa uno de los mayores obstáculos para alcanzar la escala de los Exaflops

THE GREEN 500™

TOP500

- Ranking que lista las 500 supercomputadoras más potentes del mundo.
  - Comenzó en 1993 manteniéndose hasta la actualidad.
  - Se actualiza 2 veces al año (junio y noviembre).
  - Para el cálculo de la potencia se emplea un benchmark específico llamado LINPACK.

TOP 10 Sites for November 2019

For more information about the sites and systems in the list, click on the links or view the complete list.

1-100 | 101-200 | 201-300 | 301-400 | 401-500

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
2	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta	1,572,480	94,640.0	125,712.0	7,438

www.top500.org

GREEN500

- Ranking que lista las 500 supercomputadoras más eficientes desde el punto de vista energético del mundo.
  - Comenzó en 2006 manteniéndose hasta la actualidad.
  - Se actualiza 2 veces al año (junio y noviembre).
  - Al igual que el TOP500, para el cálculo de la potencia se emplea un benchmark específico llamado LINPACK. Además, se debe medir el consumo energético durante su ejecución

Green500 List for November 2019

Listed below are the November 2019 The Green500's energy-efficient supercomputers ranked from 1 to 10.

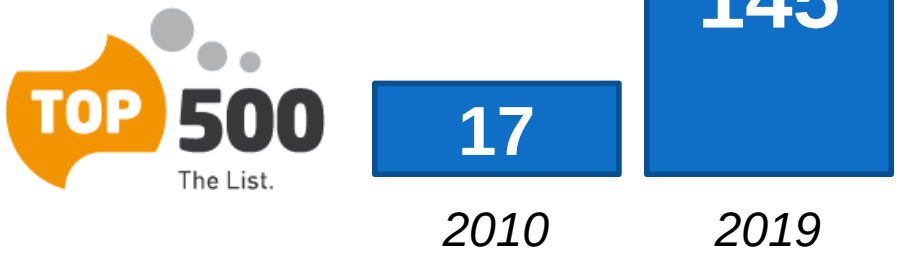
Note: Shaded entries in the table below mean the power data is derived and not meassured.

Rank	TOP500 Rank	System	Cores	Rmax (TFlop/s)	Power (kW)	Power Efficiency (GFlops/watts)
1	159	**A64FX prototype** - Fujitsu A64FX, Fujitsu A64FX 48C 2GHz, Tofu interconnect D , Fujitsu Fujitsu Numazu Plant Japan	36,864	1,999.5	118	16.876

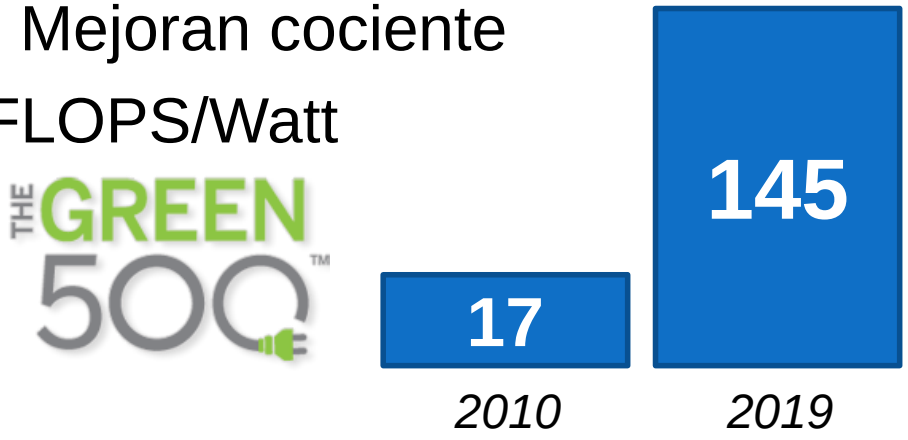
www.green500.org

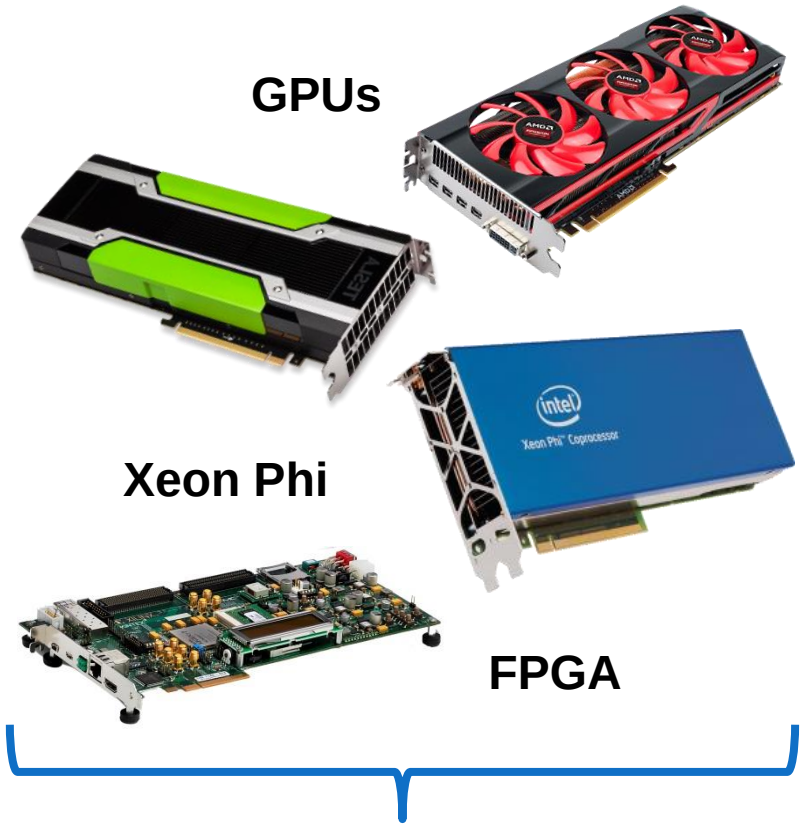
Tendencias en HPC

- Reciente consolidación de aceleradores en HPC



- Mejoran cociente FLOPS/Watt



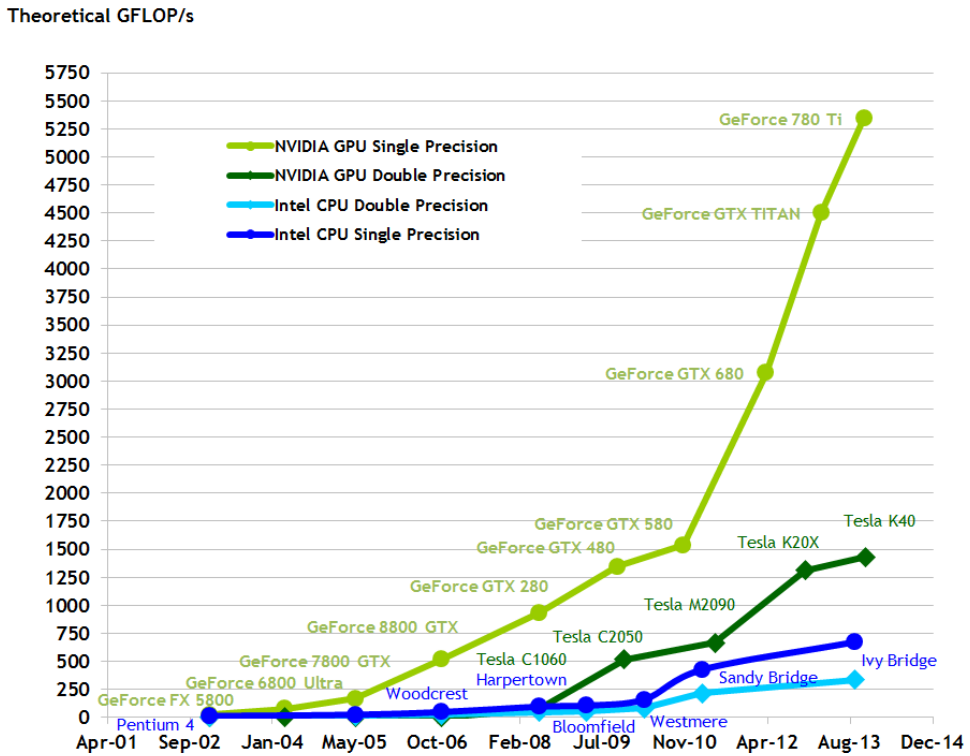


Desafío de programación

GPUs

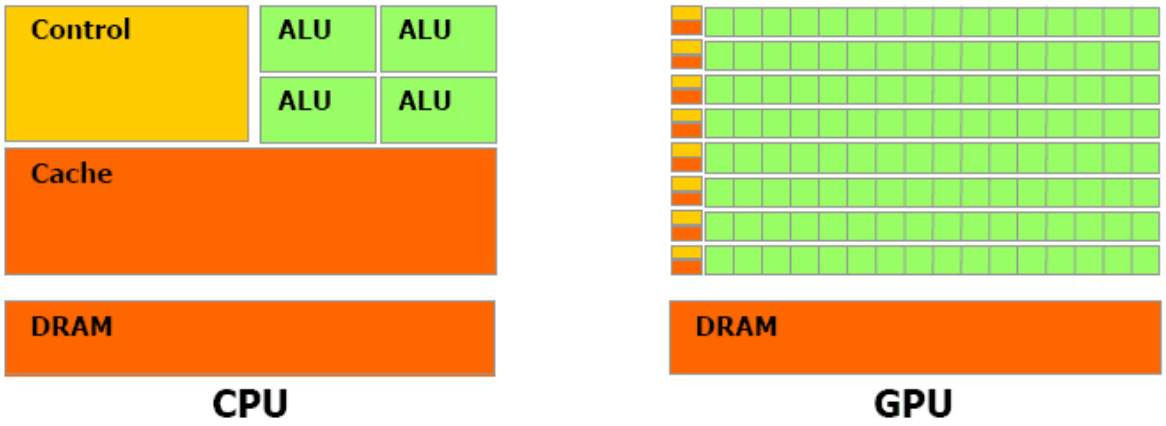
- Originalmente diseñadas para procesamiento de gráficos. Debido a su gran potencia de cálculo, las empresas fabricantes comenzaron a aumentar su grado de programación de las mismas

- Motivó el surgimiento de nuevas técnicas, lenguajes y herramientas para la programación de GPUs, lo cual permite utilizar a las mismas como arquitecturas paralelas para resolver problemas de propósito general



GPUs

- La significativa diferencia de rendimiento que existe entre las CPUs y las GPUs se debe a que sus filosofías de diseño son muy distintas
  - CPUs destinan los recursos de silicio principalmente a memorias caché y a núcleos de compleja organización que permitan explotar ILP.
  - GPUs emplean la mayor parte del silicio disponible en unidades funcionales. Cada una de ellas tiene un conjunto de núcleos simples que comparten lógica de control, ejecutan instrucciones en orden y operan en grupos como si fueran un procesador vectorial.

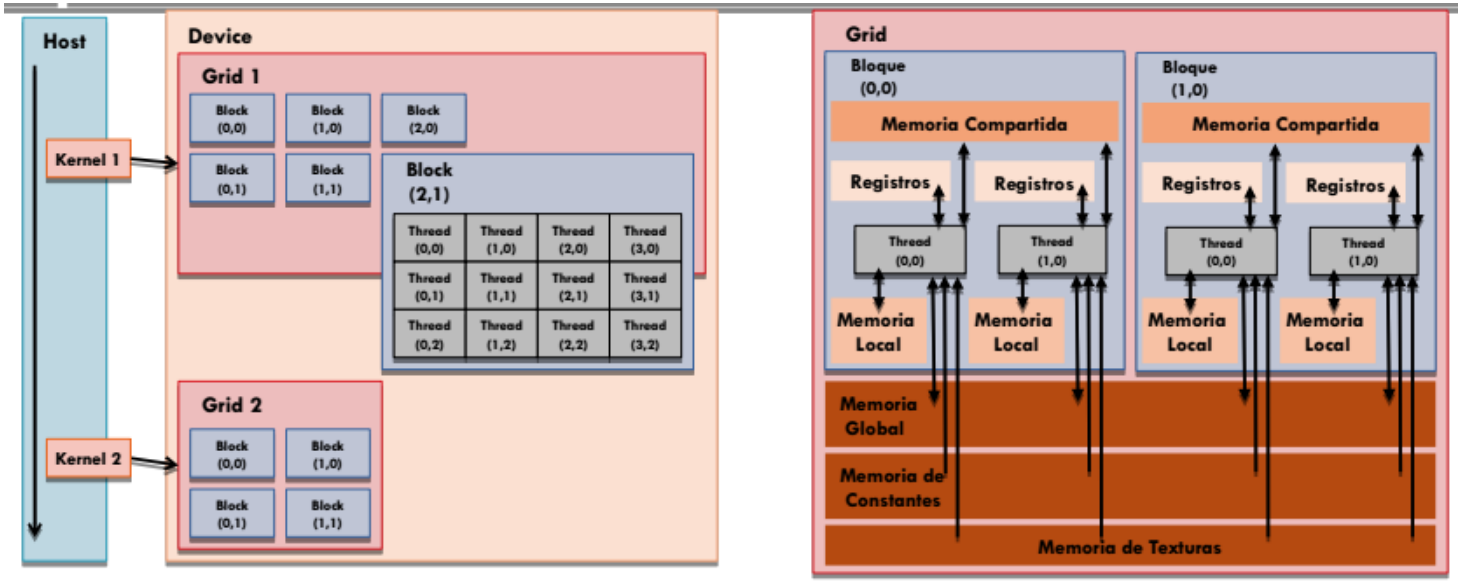


GPUs

- Las primeras aplicaciones no gráficas eran programadas en término de operaciones gráficas usando lenguajes como OpenGL o DirectX → Resultaba engorroso y propenso a errores
- Tanto la industria como la academia propusieron varios lenguajes que permiten abstraerse de los gráficos.
  - CUDA, OpenCL, OpenACC
- Al día de hoy, son 3 las empresas que comparten el mercado de las GPUs.
  - Aunque Intel es la más grande, sólo domina el segmento correspondiente a placas integradas y de bajo rendimiento.
  - En el segmento de alto rendimiento, AMD y NVIDIA son los únicos proveedores (NVIDIA supera ampliamente a AMD)

CUDA

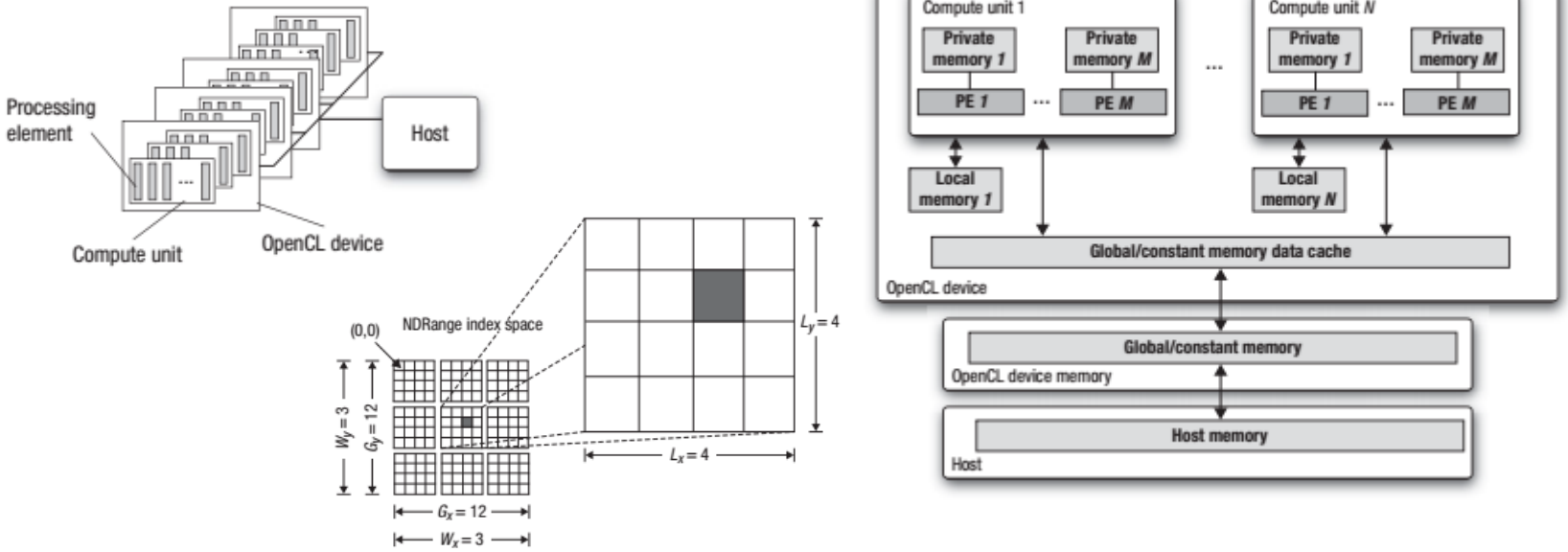
- Estándar de facto para programación de GPUs en HPC
- Modelo de ejecución (*host-device*) y arquitectura de memoria de CUDA



- El host es el responsable de administrar la memoria del dispositivos y sus transferencias, además de invocar la ejecución de los kernels.
- Un kernel es un trozo de código que ejecutan miles de hilos primitivos en paralelo en la GPU.

OpenCL

- Estándar para programación paralela multi-plataforma
- Modelo de ejecución (*host-device*) y arquitectura de memoria de OpenCL



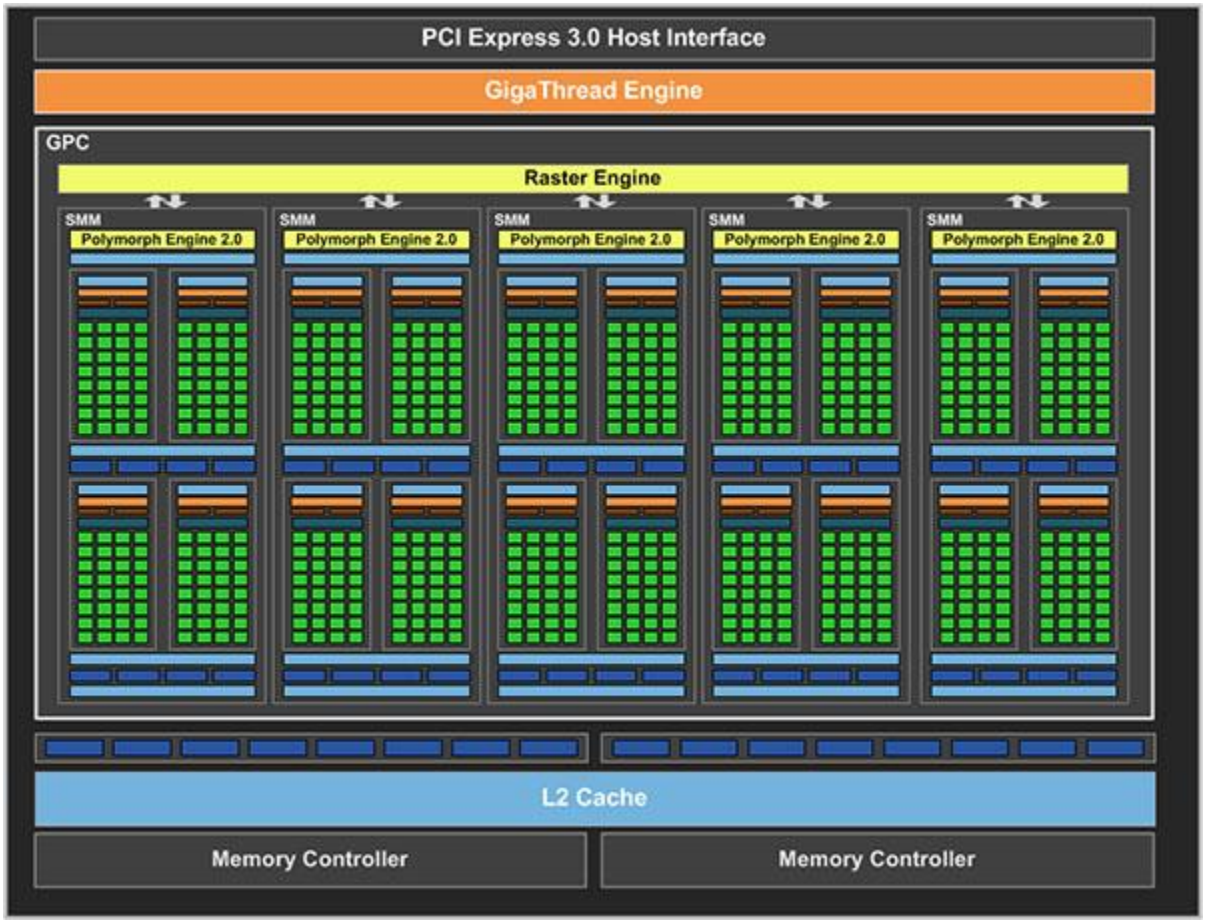
- Similar a CUDA pero con un enfoque más general ya que no sólo aplica a GPUs (CPUs, Xeon Phi's, FPGAs, DSPs, entre otros)

SYCL

- Estándar para programación paralela multi-plataforma
- Basado en OpenCL pero buscando reducir esfuerzo de programación
  - Memoria Compartida Unificada
  - Reducciones paralelas (integradas)
  - Funciones a nivel de work-groups y sub-groups
  - Accessors
  - Interoperabilidad con otras APIs

GPUs de NVIDIA

- Una GPU de NVIDIA se compone de un conjunto de *Streaming Multiprocessors* (SM).



GPUs de NVIDIA

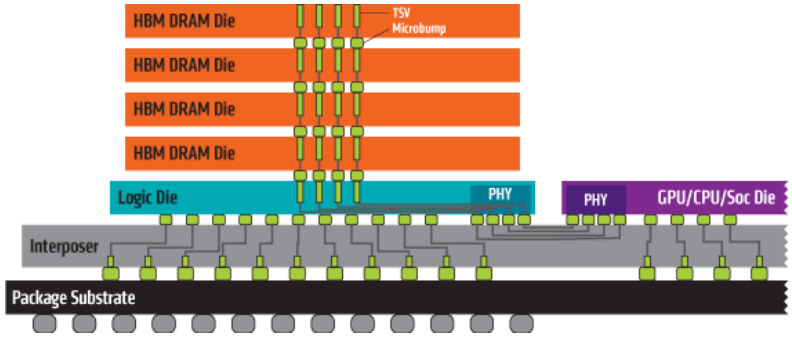
- Cada SM posee
  - Procesadores simples (*Scalar Processors* o núcleo CUDA):
    - Unidad de Punto Flotante.
    - Unidad Aritmético-Lógica.
  - SFUs (*Special Function Units*).
  - Una caché de instrucciones.
  - Una caché de sólo lectura.
  - Una memoria compartida (Shared).
  - Registros.

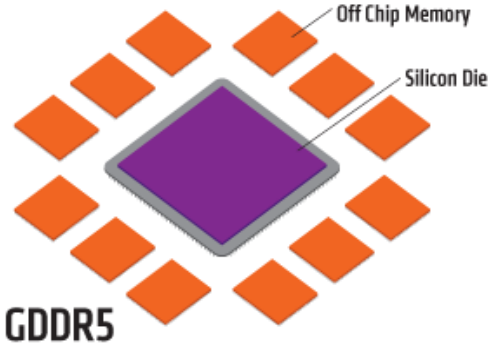
GPUs de NVIDIA

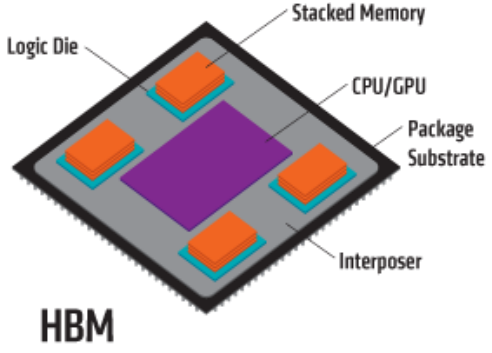
- Arquitectura Pascal en 2016
  - Orientada a mejorar la organización de la memoria y los buses de interconexión (adoptó HBM).
  - NVLINK, bus de alta velocidad (80 Gb/s) que reemplaza al PCIe (16 Gb/s).
  - Esquema de Memoria Unificada entre CPU y GPU para evitar reservas de memoria individuales (ya disponible en arquitectura Maxwell)
- Arquitectura Volta en 2017
  - Orientada al uso de Machine Learning → Incorpora soporte para precisión media (float de 16 bits) mediante núcleos específicos (Tensor cores)
  - Adopta HBM2
  - NVLINK 2.0
- Arquitectura Turing en 2019
  - Muy similar a Volta: Volta orientada al sector de alto rendimiento, Turing orientada al sector consumidor
  - Incorpora soporte específico para Ray-Tracing → Núcleos dedicados

Memoria de Alto Ancho de Banda (HBM)

- Una de las innovaciones más importantes de AMD en los últimos años es su tecnología Memoria de Alto Ancho de Banda (HBM)
  - HBM es un nuevo tipo de memoria RAM que organiza los chips de memoria en forma vertical y apilada (*memoria 3D*) y que puede ser aprovechado tanto por GPUs como CPUs.
  - Múltiples mejoras: significativo ahorro de espacio y considerables aumentos en la velocidad de comunicación y en la eficiencia energética
  - Incorporada en las GPUs con nombre clave Fiji de de AMD en 2015. También fue adoptada por NVIDIA para sus placas de la arquitectura Pascal en 2016.

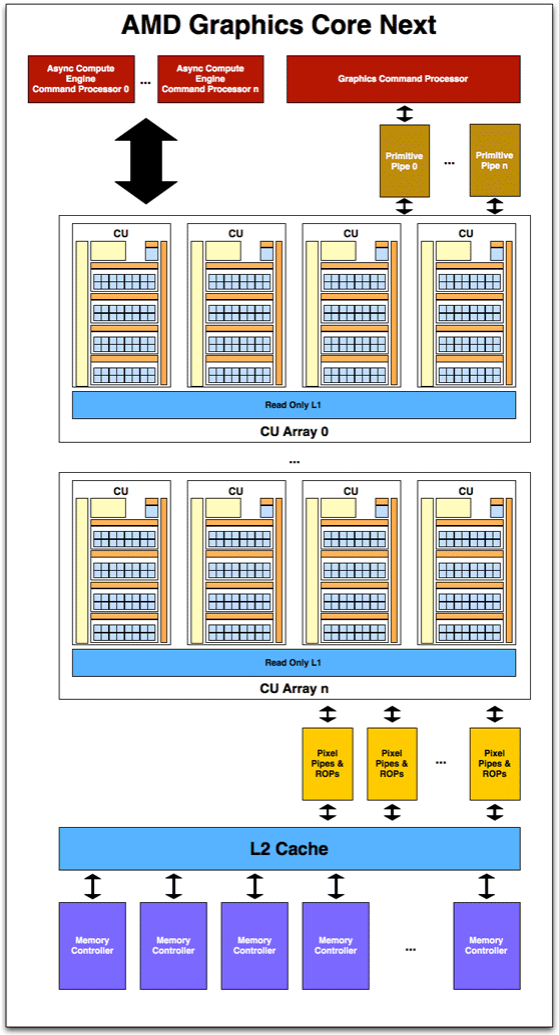






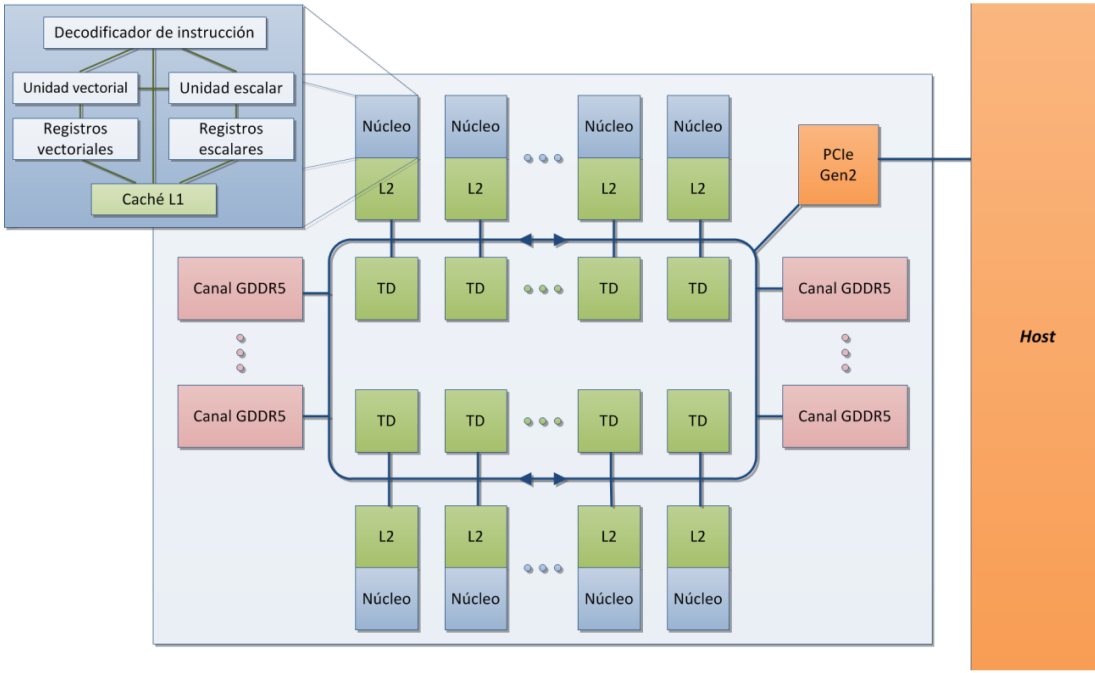
GPUs de AMD

- En el año 2006, AMD compró la empresa de placas gráficas ATI, lo que le permitió incrementar notablemente su capacidad de producir e innovar hardware gráfico.
- Una GPU de AMD se compone de una colección de arreglos de *Compute Units* (CUs)
- Cada CU contiene un conjunto de unidades SIMD, unidades para planificación de instrucciones, una unidad escalar y distintos bloques de memoria caché L1



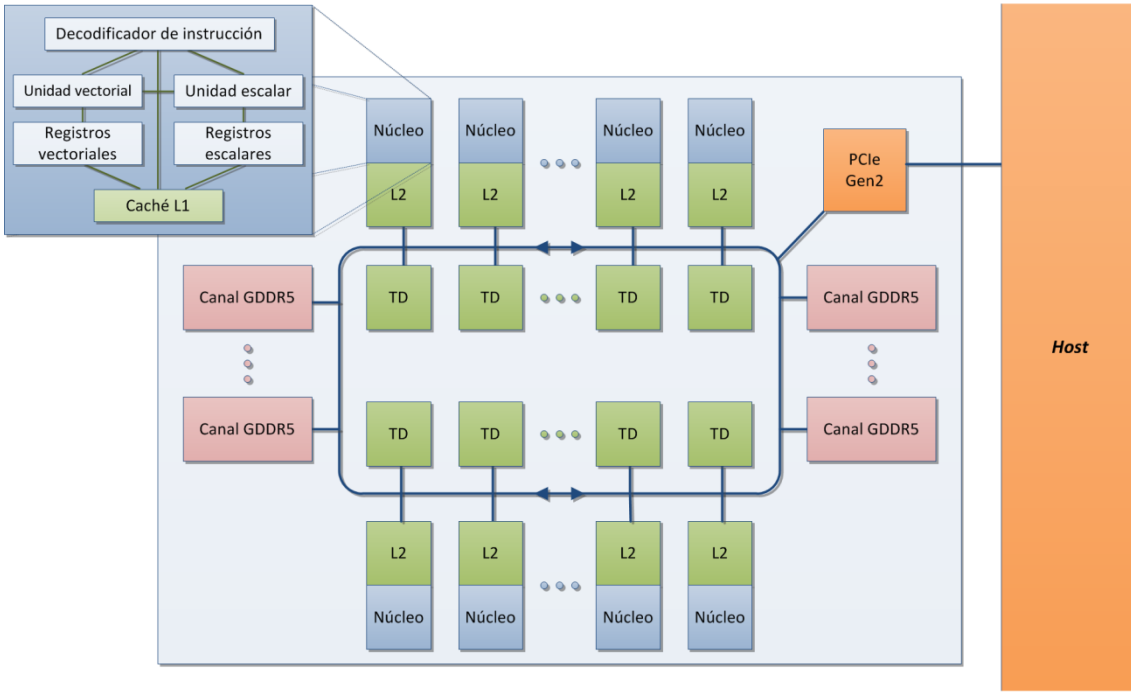
Intel Xeon Phi

- Arquitectura desarrollada por Intel para competir con las GPUs
- Primera generación lanzada en 2013 (Knights Corner, KNC)
  - Coprocesador de hasta 61 núcleos x86 con unidades vectoriales extendidas (512 bits) y SMT (4 hilos hardware por núcleo).
  - Caché L1 de 64 Kb + Caché L2 de 512 Kb
  - Interconexión en forma de anillo de alta velocidad
  - Conexión al host a través del bus PCIe Gen2



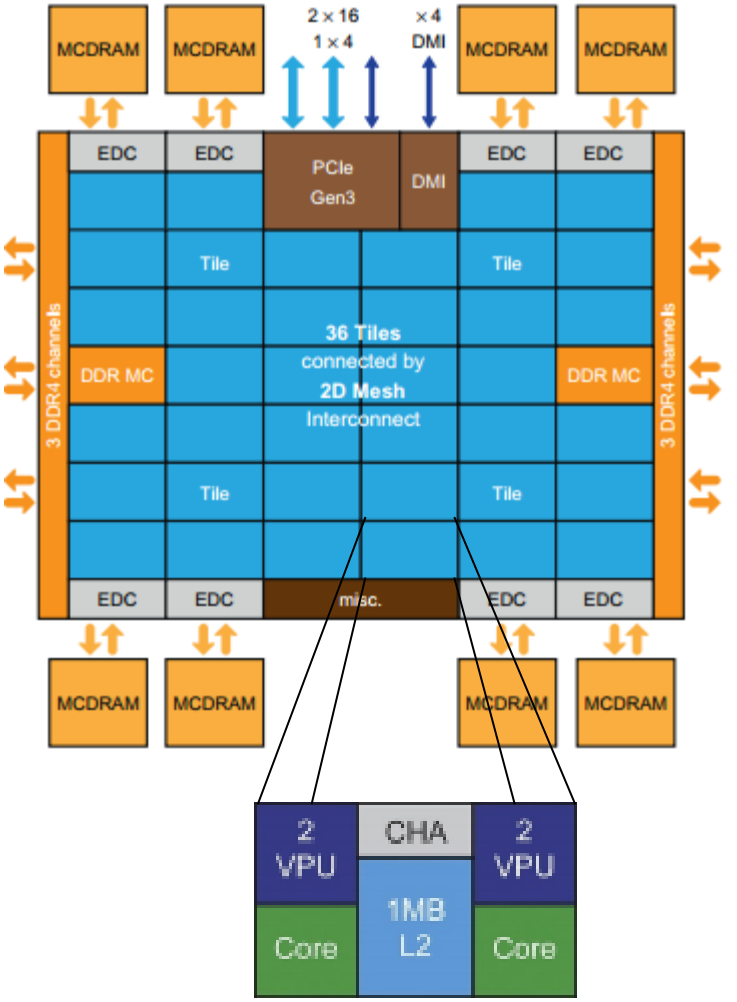
Intel Xeon Phi – Knights Corner

- Ejecuta un sistema operativo propio basado en Linux
- Compatibilidad de código con procesadores Xeon → requiere compilación cruzada
- Dos modos de ejecución:
  - Nativo
  - Offload



Intel Xeon Phi

- Segunda generación lanzada en 2015: Knights Landing (KNL)
- Capacidad de operar autónomamente
- Características arquitectónicas
  - Hasta 36 *Tiles* interconectados por malla 2D
  - Cada Tile incluye 2 núcleos:
    - Basados en la micro-arquitectura Intel Atom (fuera de orden, 4 hilos hw por núcleo)
    - 2 unidades vectoriales por núcleo
    - Caché L2 compartida de 1 MB
- Compatibilidad plena con arquitectura x86

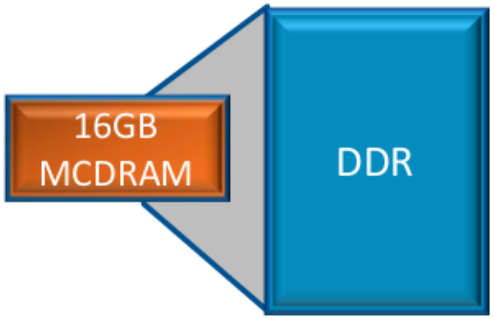


Intel Xeon Phi Processor High Performance Programming Knights Landing Edition . Jim Jeffers, James Reinders, Avinash Sodani.

Intel Xeon Phi - Knights Landing

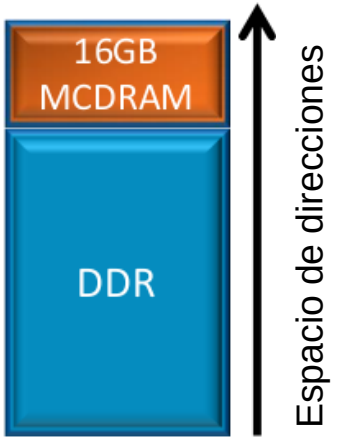
- Incorporación de memoria de alto ancho de banda (HBM) mediante tecnología MCDRAM

Modo cache



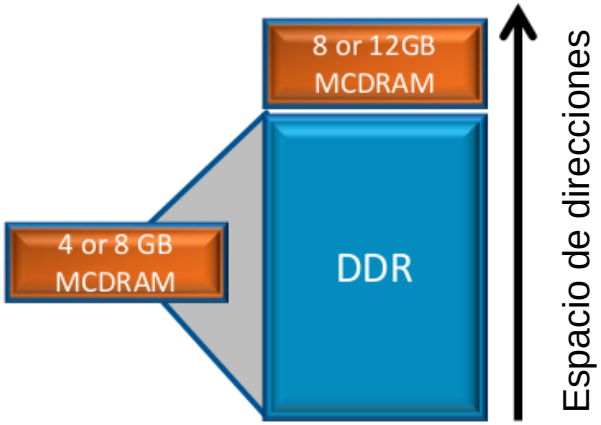
- ✓ Transparente al usuario
- ✓ No requiere cambios en el código fuente
- ✓ Puede sufrir pérdida de rendimiento

Modo *flat*



- ✓ Ofrece el mayor ancho de banda y la más baja latencia
- ✓ Usualmente requiere cambios en el código fuente

Modo híbrido



- ✓ Combinación de las dos anteriores

Knights Landing (KNL): 2nd Generation Intel® Xeon Phi™ Processor. Avinash Sodani.

Intel Xeon Phi

- Tercera generación lanzada en 2017: Knights Mill (KNM)
- Variante de KNL orientada a Deep Learning → Incorpora instrucciones que duplican/cuadriplican rendimiento en FP32/FP16 pero reducen a la mitad en FP64
- Intel discontinuó la línea Xeon Phi en 2019 para orientarse a otros productos

Intel Xe

- Primera GPU discreta de Intel (2021)
  - Variantes para gaming, datacenter y HPC

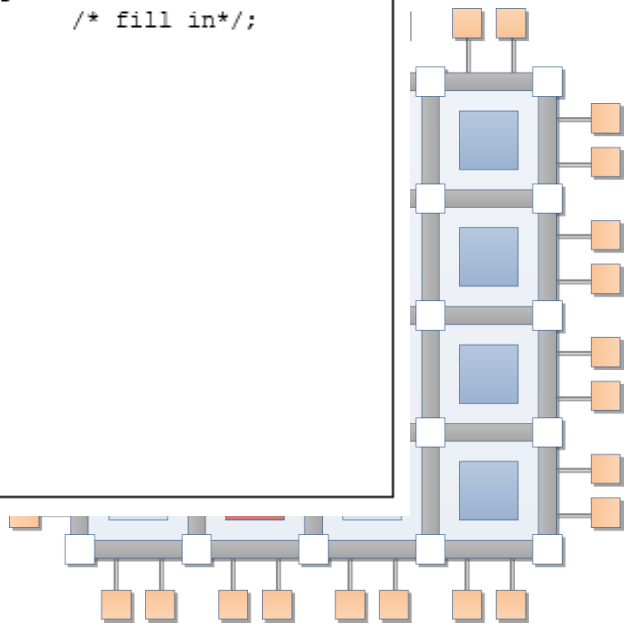
FPGAs

- Arqu
  - Ci
    in
    pr
- Capa
  obje
  - Pro
    De
    -
    -
  - Re
    per
    pro
    - SDK para OpenCL (Altera/Xilinx)

Ps

```
// Finite-State Machine for a simple elevator
module elevator (
 input    clk,        // clock (~1Hz)
 input    rst,        // asynchronous reset button
 input    go,         // 'go' button
 input      [3:0] dest_floor,
 output reg [3:0] curr_floor,
 output   door);      // indicate door open

  // State assignments
 parameter WAIT  = 0;
 parameter MOVE  = 1;
 parameter OPEN  = 2;

 reg [1:0] state, next_state;
 reg [3:0] goto_floor;
 reg [2:0] cnt;

 // latch the new destination floor
 always @(posedge clk)
   if (go)
     goto_floor <= dest_floor;

 // code curr_floor as a 4-bit up/down counter
 always @(posedge clk, posedge rst)
   if (rst)
     curr_floor <= 0;  // start at ground floor
   else if
     /* fill in */

 // add 5-second counter for door
 always @(posedge clk, posedge rst)
   if (rst)
     cnt <= 0;
   else if
     /* fill in */
```

```
// Sequential logic
always @(posedge clk, posedge rst)
  if (rst)
    state <= WAIT;
  else
    state <= next_state;

// Next-state logic
always @(*) begin
  next_state = state;
  case (state)
    WAIT: if (go) next_state = MOVE;
    MOVE:          /* fill in*/;
    OPEN:          /* fill in*/;
  endcase
end

// Output logic
assign door =      /* fill in*/;

endmodule
```



Comparación de rendimiento y eficiencia energética

		Intel Xeon E5-4669 v3	NVIDIA Tesla K40	Intel Xeon Phi 7120P	Xilinx Virtex7 XCTV2000T
Número de procesadores		18 núcleos (2 hilos hardware cada uno)	2880 núcleos CUDA	61 núcleos (4 hilos hardware cada uno)	2160 bloques DSP
Frecuencia del reloj (GHz)		2.1 - 2.9	0.745	1.238 - 1.333	< 0.741
Ancho SIMD		8	32	16	Configurable
Pico de FLOPS (GFLOPS)	**Precisión simple**	604.8	4290	2416	1636
	Precisión doble	302.4	1430	1208	671
TDP (Watt)		135	245	300	< 40
GFLOPS/Watt		4.48	17.51	8.05	> 40.9

Bibliografía usada para esta clase

- M. Giles and I. Reguly, Trends in high-performance computing for engineering calculations, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 372, no. 2022, p. 20130319, 2014.
- M. Vestias and H. Neto, Trends of CPU, GPU and FPGA for high-performance computing, in Field Programmable Logic and Applications (FPL), 2014 24th International Conference on, Sept 2014, pp. 1-6.
- E. Rucci, Evaluación de Rendimiento y Eficiencia Energética de Sistemas Heterogéneos para Bioinformática, Tesis Doctoral, UNLP, 2016.