High Performance Computing

Introduction

Fernando R. Rannou
Departamento de Ingeniería Informática
Universidad de Santiago de Chile

August 12, 2024

¿Qué es paralelismo?

En vez de definir qué es paralelismo, no acercaremos al concepto de paralelismo a través de varios de los elementos que lo conforman. Asi, al final la definición tendrá más sentido.

Velocidad secuencial

- Considere un procesador moderno de 3.0 Ghz el cual completa una instrucción en 3 ciclos. Luego

$$N = 3 \times 10^9 (\text{ciclos/s}) \cdot \frac{1}{3} (\text{inst./ciclo}) = 10^9$$

  instrucciones por segundo
- Es decir, en 60 segundos, el procesador ejecutaría, 60 mil millones de instrucciones.
- Consideraciones de acceso a memoria, memoria cache, contención del bus, y otros aspectos impiden alcanzar este rendimiento
- En grandes aplicaciones paralelas se requieren Trillones (10^{12}) de operaciones por segundo

Aplicaciones *Grand challenge*

Son problemas fundamentales en ingeniería y ciencias con gran impacto en la sociedad y economía. Generalmente son problemas intratables si no se usan computadores paralelos masivos para resolverlos.

- Modelamiento global del clima: Típicamente se modela con una grid en la que cada celda es de 300 Km. Un dia puede tomar 1800 segundos en un IBM SP
- El genoma humano
- Química computacional: Usando 16 nodos de un supercomputador IBM, una simulación de 1 nanosegundo de *dinámica de proteínas* demora 1 semana
- Cognición
- Visión
- Astrofísica

Supercomputadores tradicionales; el pasado

1. Tecnología
   - Procesador creado específicamente para un supercomputador
   - Max. performance posible con buen ancho de banda de memoria
2. Beneficios
   - Programación secuencial
   - Más de 30 años de desarrollo de herramientas de software
   - I/O es relativamente simple
3. Limitaciones
   - Procesadores dedicados extremadamente caros
   - Se requiere de sistemas sofisticados de enfriamiento
   - Rendimiento por procesador llegando a un límite

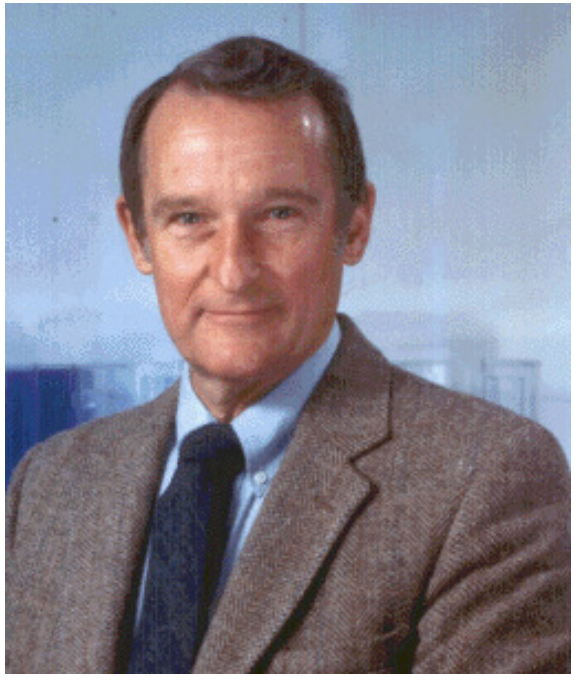
Supercomputadores paralelos actuales

1. Tecnología
   - Uso de muchos procesadores pequeños que trabajen en una parte del problema a resolver
   - Capitalización de la inversión de la industria de microprocesadores y redes
2. Beneficios
   - Variedad de procesadores de propósito general en el mercado
   - Buena capacidad de escalamiento
   - Más barato
3. Limitaciones
   - Nueva tecnología; programación paralela
   - Códigos secuenciales no sirven
   - Necesidad de nuevas tecnologías de software: compiladores, depuradores
   - I/O más complicado

Un poco de historia: Seymour Cray (1925-1996)

Considerado *el padre de la supercomputación*

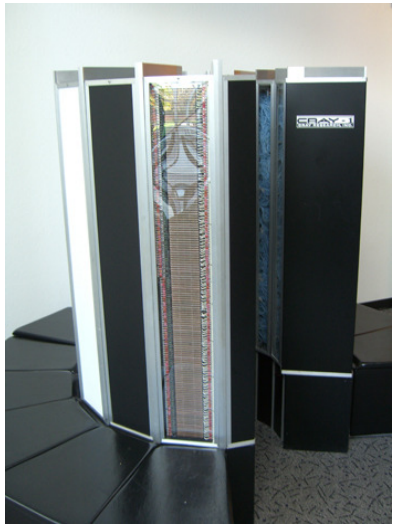
- En los años 50 participa en el diseñõ del UNIVAC 1103
- En 1957 construye el CDC 1604, uno de los primeros computadores comerciales en usar transistores en vez de tubos de vacio
- Diseña el computador más potente de su época, el CDC 6600
- En 1976 crea Cray-1, el primer supercomputador vectorial



Cray computers

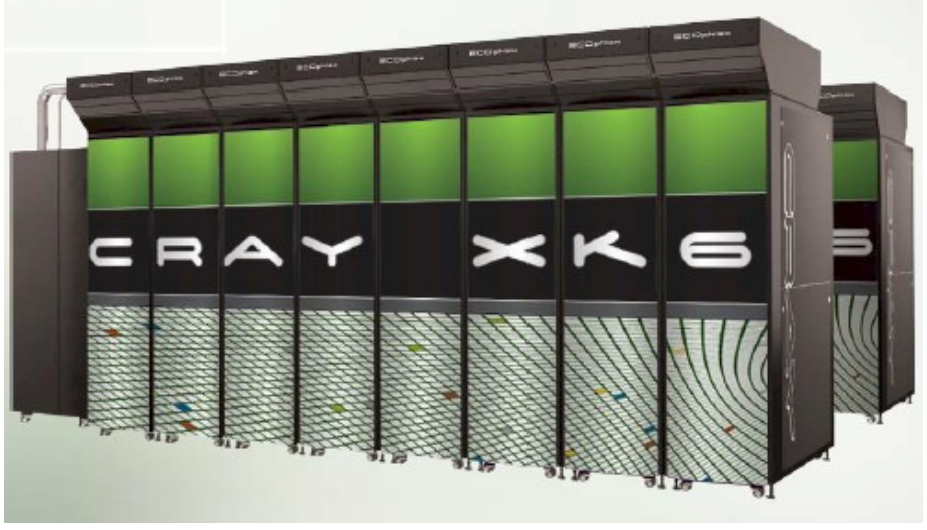
Cray-1 (1976):

- 64 Procesadores vectoriales, 80 Mhz
- 16 Mbytes memoria, 160 MPIS
- Sistema refrigeración por freón
- SO Cray UNICOS (variante de Unix)
- 5.5 toneladas



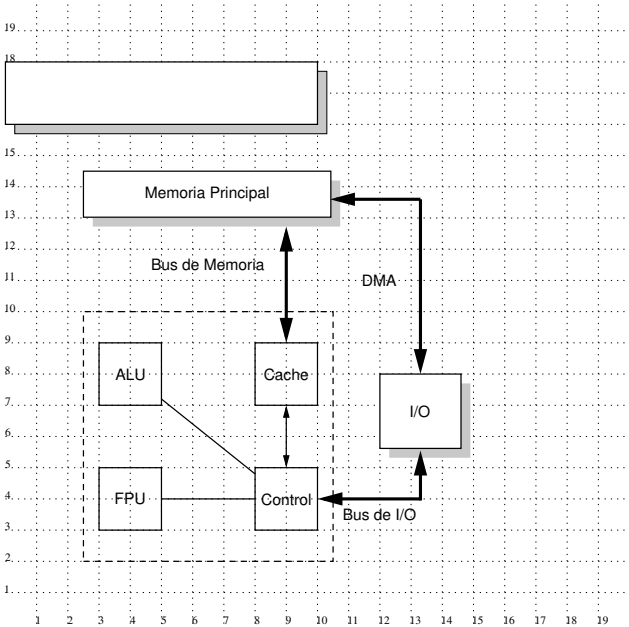
Cray xk6 (2011)

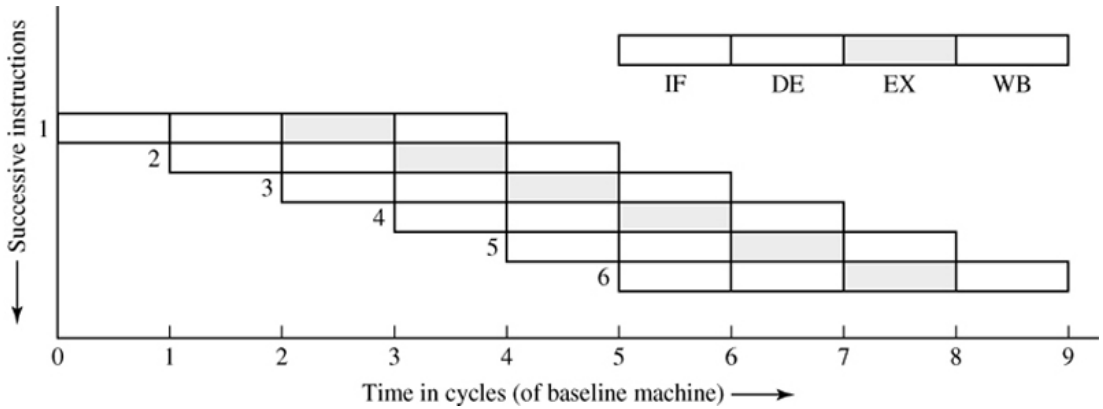
- Nodos de procesadores AMD 16 núcleos, 16GB o 32GB
- NVIDIA Tesla GPU, 6 GB
- Un cabinete tiene 96 nodos
- 70+ Teraflops por cabinete
- SO Cray Linux Environment



Procesador escalar

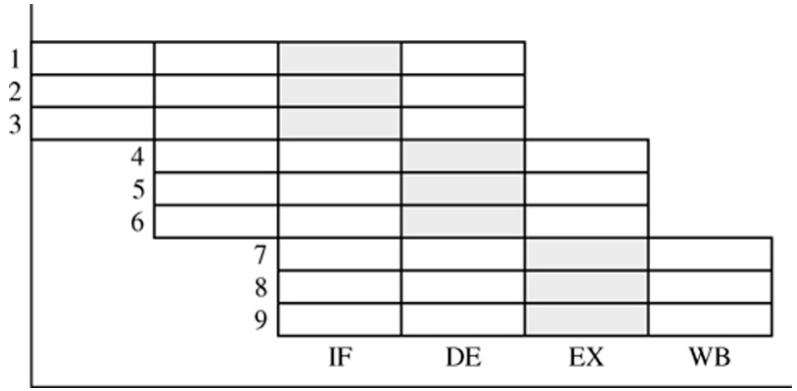
- Un *procesador escalar base* consiste de un procesador que "emite" una instrucción por ciclo de reloj
- También existe una o más unidades de I/O
- Los elementos se comunican a través de buses del sistema
- En general, los procesadores escalares están basados en pipelining
- El 99.2% de los computadores top 500 (2007) tienen procesadores escalares





Procesador super escalar

- Un *procesador super escalar* emite múltiples instrucciones por ciclo de reloj
- Por ejemplo, si el procesador tiene m pipelines, puede ejecutar m instrucciones en paralelo (si éstas son independientes)
- El número real de instrucciones ejecutadas por ciclo depende de la independencia entre las instrucciones.
- Uso de optimización en fase de compilación




```
for i=0 to N {
    a[i] = sin(30*i/PI)
}
```

```
a[0] = 0.5
for i=1 to N {
  a[i] = a[i-1]*sin(30*i/PI)
}
```

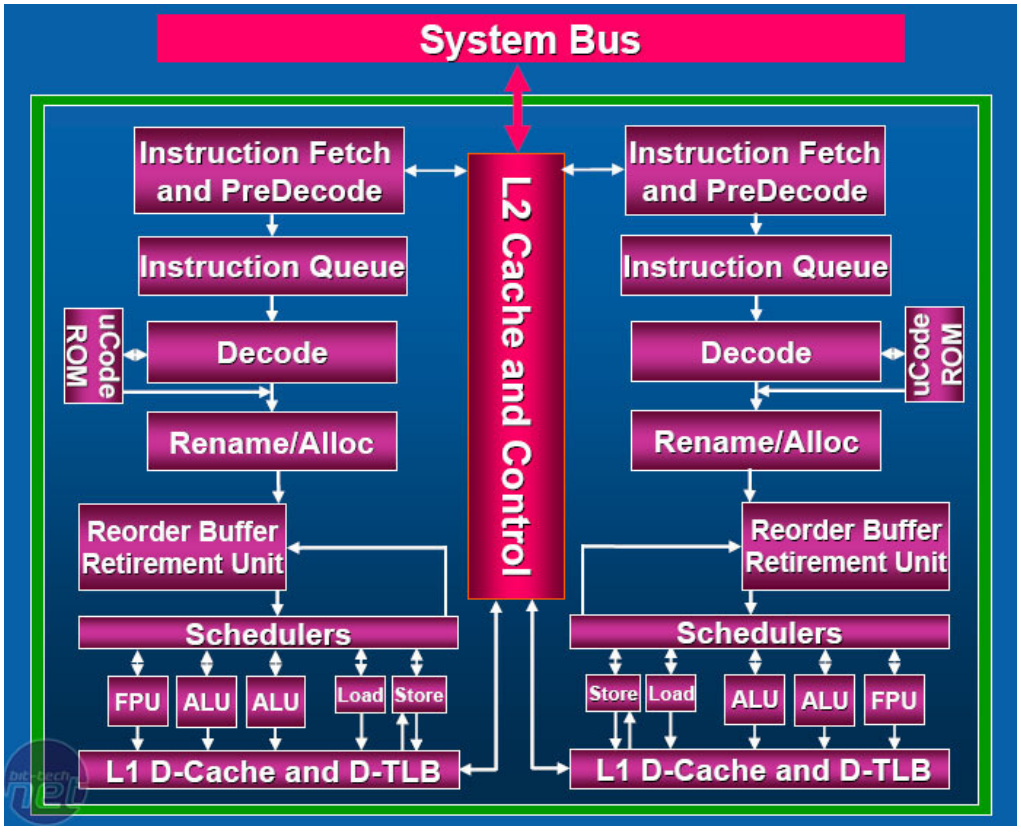
Procesador vectorial

- En un procesador vectorial paralelo (PVP) una instrucción opera simultáneamente sobre varios elementos de vectores o arreglos de datos
- Generalmente son muy caros y limitados en aplicaciones
- No hay mucho desarrollo en la actualidad (really...???)
- Apoyo del compilador para vectorizar loops (Fortran 90)
- Ejemplos: Cray SV1, Cray SV2, NEC Earth simulator
- El 0.8 % de los computadores top 500 (2007) tienen procesadores vectoriales

```
z(1:100) = x(1:100) + y(1:100)
```

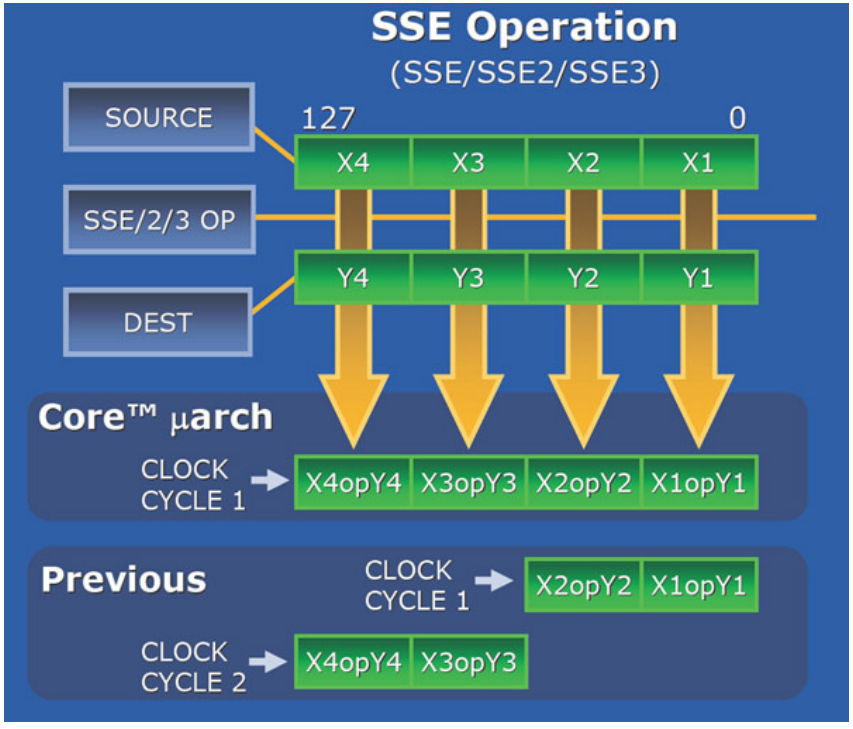
Intel Core 2 Duo

- El **ancho** del pipeline es 4, y el **largo** del pipeline es de 14 etapas
- *Smart Cache*: prefetching
- Cache L2 compartida entre los cores. ¿Cuáles son los beneficios?



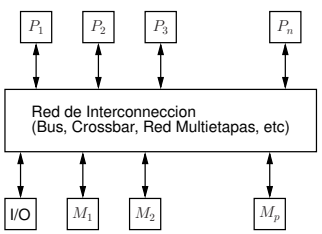
Inte Core 2 Duo (cont)

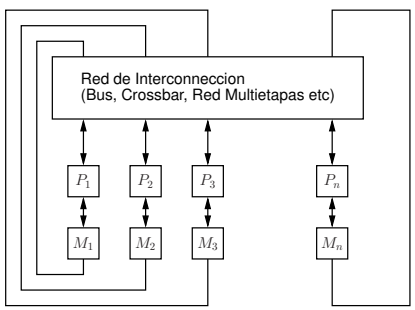
- Unidades SSE dedicadas a operaciones SIMD enteras y flotantes, por ALU!
- SSE = Streaming SIMD Extensions
- 4×32 bits o 2×128 bits, en un ciclo de reloj



Multiprocesadores de memoria compartida

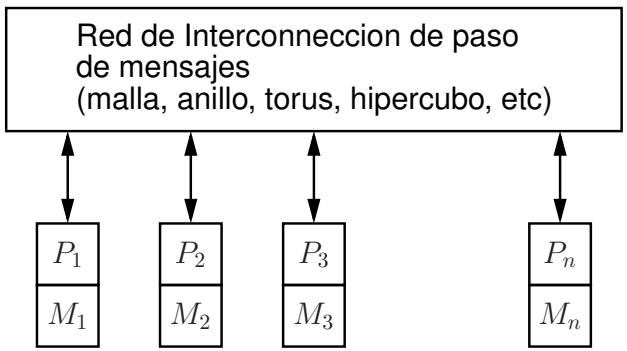
- Un multiprocesador de memoria compartidad contiene dos o más procesadores que comparten el espacio de memoria. Dos modelos:
  1. UMA (Uniform memory-access): la memoria física es compartida por todos los procesadores; por lo tanto el tiempo de acceso es el mismo para todos.
  2. NUMA (Non-uniform memory-access): la memoria física esta distribuida (local a cada procesador) y por lo tanto el tiempo de acceso a ella no es uniforme.
- La colección de todas las memorias locales forman el espacio global de direcciones
- Programación multihebra usa eficientemente esta aquitectura





Multiprocesadores de memoria distribuida

- Cada nodo consiste de un procesador autónomo con memoria local, y dispositivos de I/O.
- Los nodos se comunican mediante una red de paso de mensajes, la cual provee conexiones punto-a-punto y estáticas
- La memoria local a cada nodo sólo puede ser accesada por dicho nodo
- Variadas topologías de interconección: anillo, torus, hypercubo, etc.
- Librerías nativas para comunicación proveen mejor rendimiento que librerías de propósito general



Cluster de computadores

- Cada nodo es un computador separado del resto, ej: PCs, estaciones de trabajo, etc
- Los nodos pueden o no ser heterogéneos (procesador, velocidad, memoria, SO, otros)
- Se conectan a través de un red de paso de mensajes
- Sencillos de construir, actualizar; baratos
- Limitada escalabilidad
- I/O puede ser complicado

Tendencias actuales

- Memoria distribuida compartida; necesidad de protocolos de coherencia de memoria cache
- Procesadores Many-core
- HPC cloud computing
- GPGPU
- Virtualization

Taxonomía Flynn

Propuesta para clasificar las arquitecturas de acuerdo a la multiplicidad de los datos e instrucciones

1. Single-Instruction, Single-Data (SISD)
   - Modelo tradicional, escalar
   - Puede tener pipelining
2. Single-Instruction, Multiple-Data (SIMD)
   - Arquitectura vectoriales
3. Multiple-Instruction, Multiple-Data (MIMD)
   - Mutliprocesadores y multicomputadores
   - MIMD de memoria compartida
   - MIMD de memoria distribuda
4. Single-Instruction, Multiple-Data (SIMD)

Supercomputadores modernos; el presente (2018)

1. Tecnología
   - Uso de muchos procesadores pequeños que trabajen en una parte del problema a resolver
   - Capitalización de la inversión de la industria de microprocesadores y redes
2. Beneficios
   - Variedad de procesadores de propósito general en el mercado
   - Buena capacidad de escalamiento
   - Más barato
3. Limitaciones
   - Nueva tecnología; programación paralela
   - Refactorización códigos secuenciales
   - Necesidad de nuevas tecnologías de software: compiladores, depuradores
   - I/O más complicado

Los computadores más veloces: www.top500.org

- Ranking de los computadores más veloces del mundo
- LINPACK benchmark

$$Ax = b$$

- A is $n \times n$ dense matrix.
- FLOP(n) $= \frac{2}{3}n^3 + 2n^2$
- Rendimiento peak teórico, *Rpeak*
- Rendimiento peak real, *Rmax*

Rpeak = `Nsockets` x `Ncores` x Ciclos/s x `OP per ciclo`

Rmax = `FLOP(n)/Tiempo Ejecución`

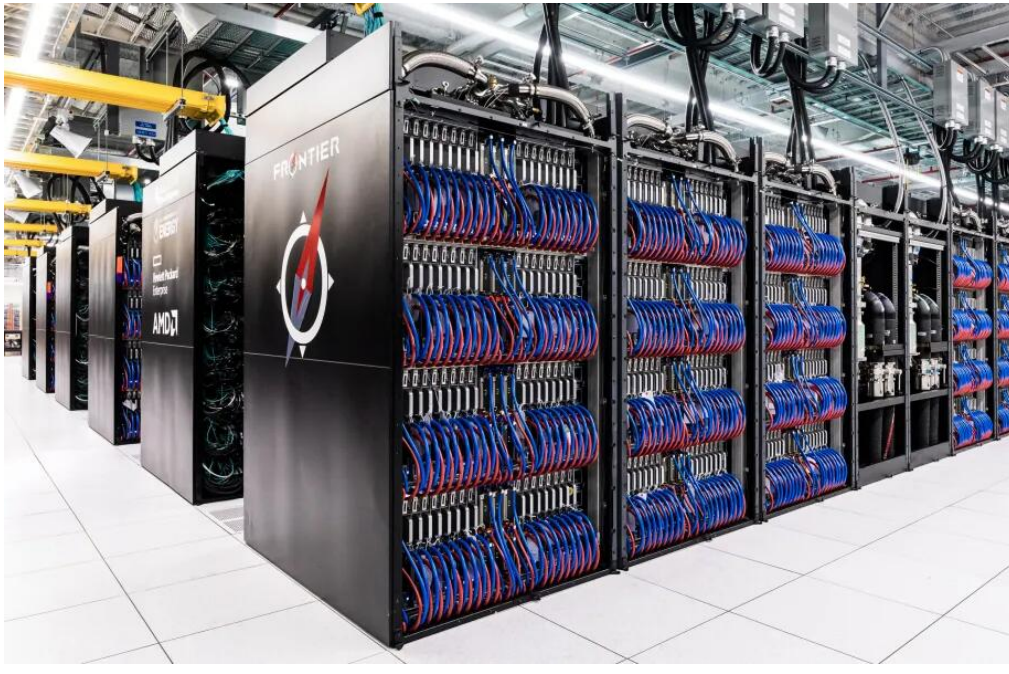
Ejemplos *Rpeak* (Single Precision)

$$Rpeak = \text{Nsockets} \times \text{Ncores} \times f \times \text{OP/ciclo}$$

Procesador	Sockets	Cores per socket	f	OPs/c	*Rpeak*
Cray-1 (1976)	1	1	80 MHz	2	160 MFLOPS
Xeon E5-2680	2	10	2.80 GHz	8	448 GFLOPS

1ero : Frontier: HPE Cray 1206/1714 petaflops

- First exascale supercomputer
- AMD EPYC
- 64 cores/socket, 2 Ghz
- 8.699.904 cores
- Oak Ridge National lab, USA

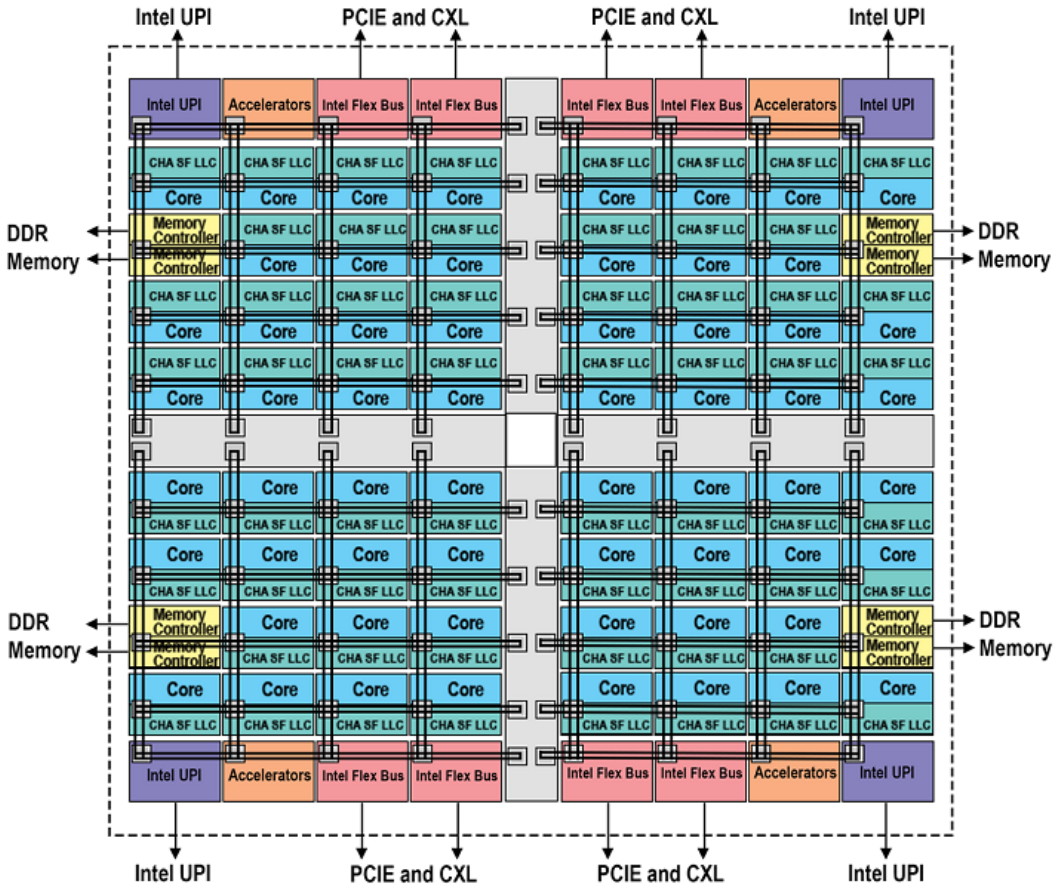


1 PFLOPS $= 10^{15}$ ops/sec

2nd : Aurora, HPE Cray 1012/1980 petaflops

- Xeon CPU Max 9474
- 52 cores/socket, 2.4 Ghz
- 9.264.128 cores
- Argonne National Lab, USA

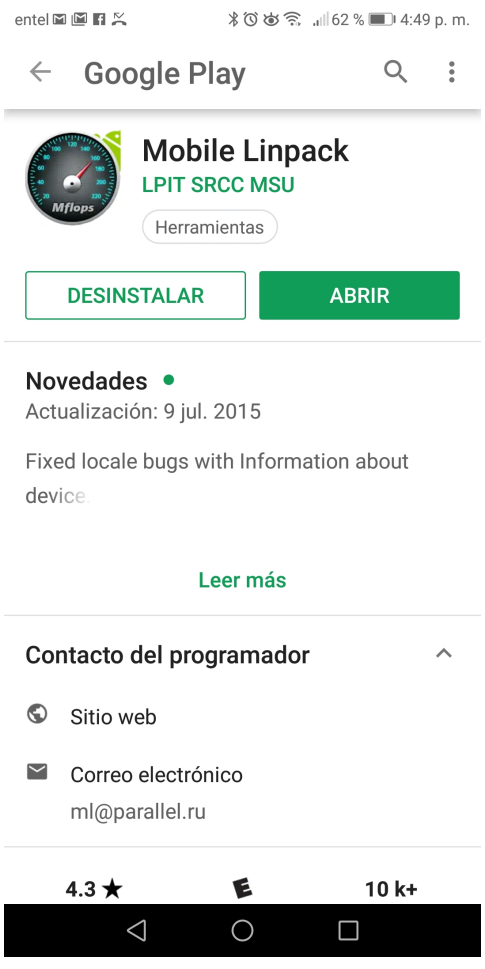
Xeon Max series (9474)

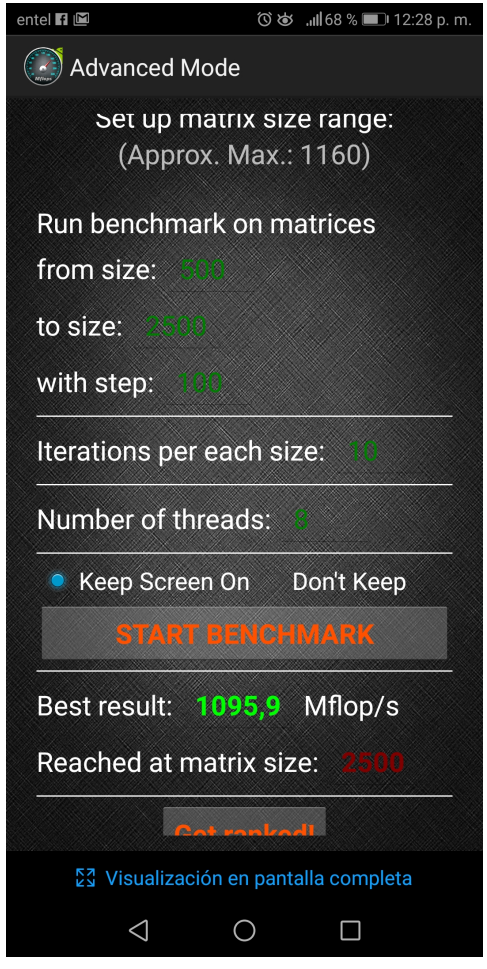


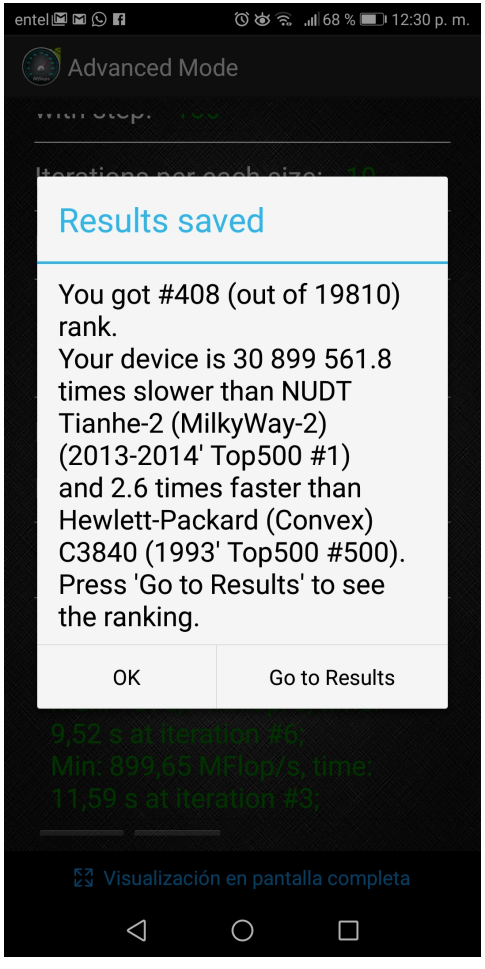
3ero : Eagle, Microsoft NDv5 561.2/846.84 petaflops

- Xeon Platinum 8480C
- 48 cores/node, 2 Ghz
- 2.073.600 cores
- RIKEN Center for Computational Science, Japan

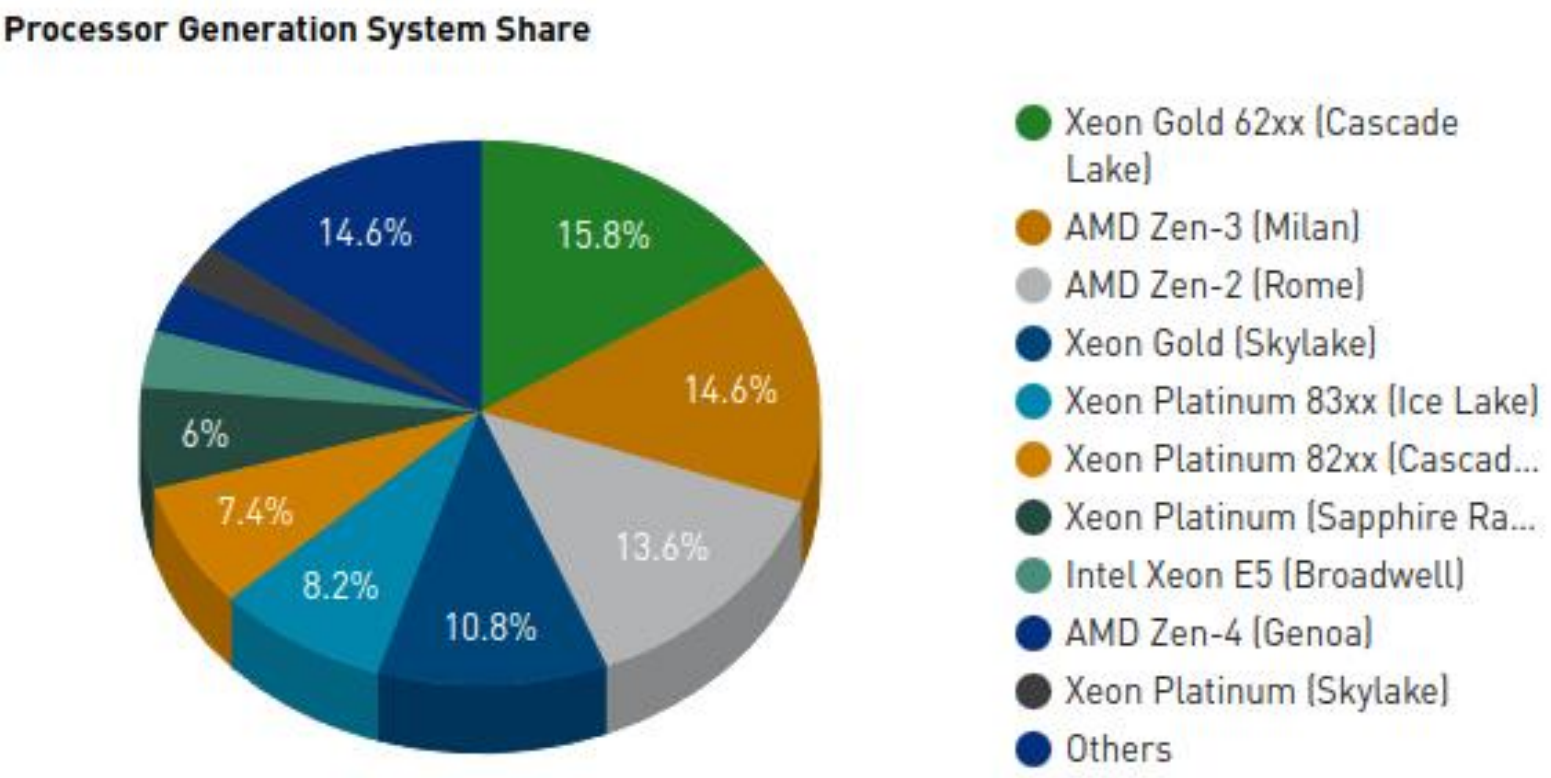
¿Y mi teléfono celular?



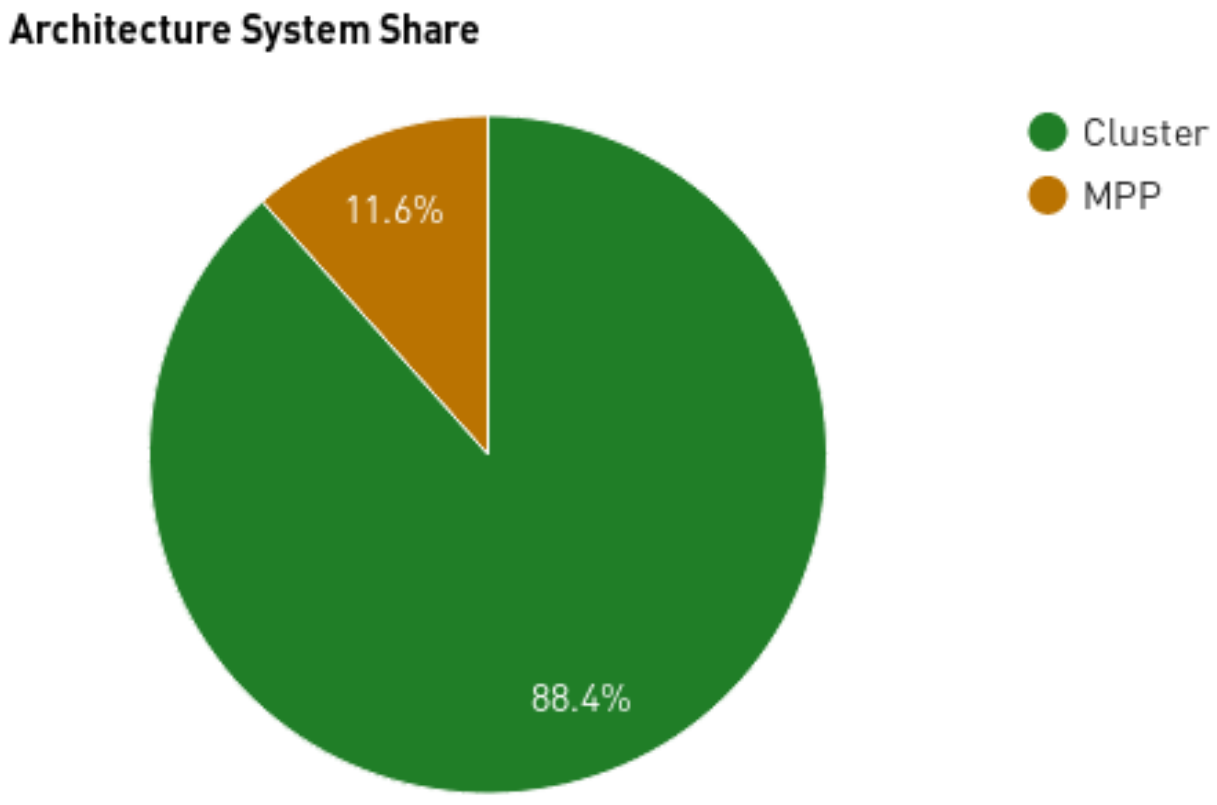




Top500 : Distribución por procesador

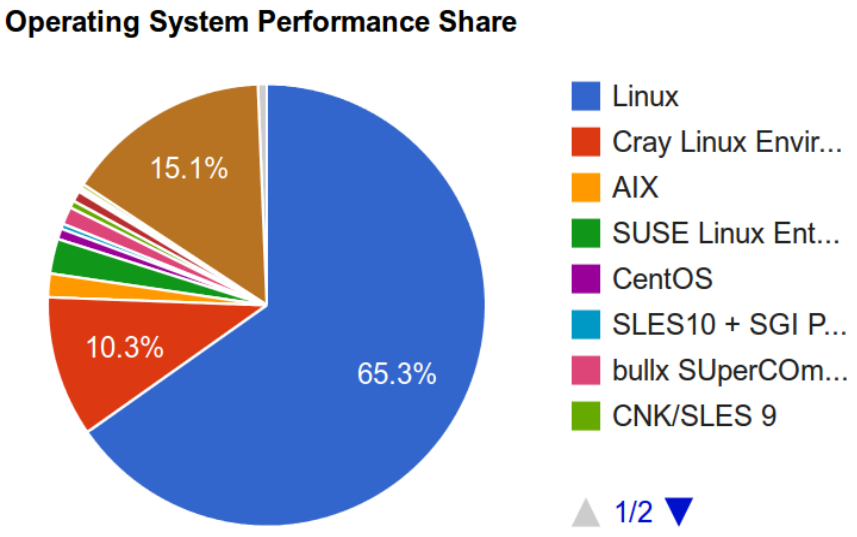


Top500 : Distribución por arquitectura

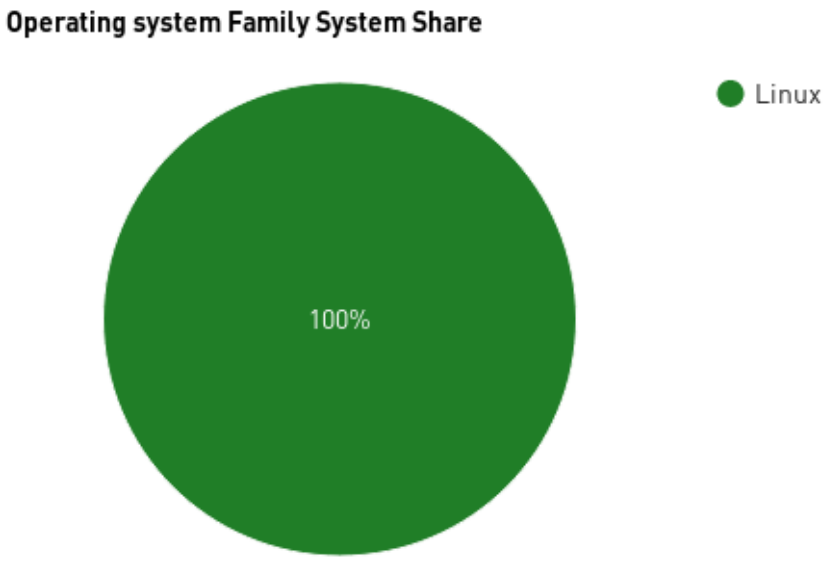


Distribución por familia de SO

2012



2024



Número de operaciones por ciclo

$$Rpeak = \text{Nsockets} \times \text{Ncores} \times f \times \text{OP/ciclo}$$

¿Qué es paralelismo?

- Una estrategia compuesta de elementos de hardware y software para resolver complejos problemas computacionales, en forma más rápida.
- En términos simples, paralelismo se logra realizando las siguientes etapas
  1. Dividir el problema en tareas más pequeñas
  2. Asignar las tareas a un conjunto de procesadores que trabajen simultáneamente
  3. Coordinar a los procesos
- Una solución secuencial sólo se preocupa de encontrar un algoritmo que resuelva el problema.
- Paralelismo implica además considerar la infraestructura paralela.