# Programación Paralela y Distribuida

## *Introducción a la Programación HPC*

## Índice





## Introducción

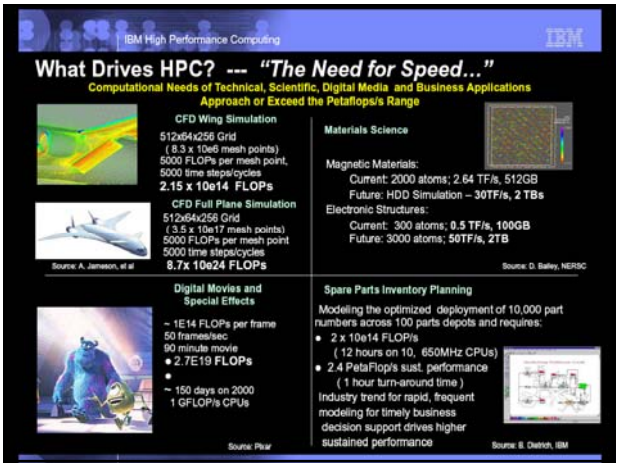
- Hay muchas aplicaciones científicas y de ingeniería que tienen tiempos de ejecución muy elevados
  - Dinámica molecular, química cuántica, estudio de nuevos materiales, **bioinformática**, defensa, finanzas, dinámica de fluidos, resistencia de materiales, electrónica, animación, **meteorología**, …
- Experimentos virtuales vs físicos
- La paralelización es la única forma de alcanzar tiempos de ejecución razonables y poder abordar experimentos más complejos.
- Aplicaciones “insaciable”, escalan más rápido que el Hw



## Introducción

- HPC: High Performance Computing (Computación de Alto Rendimiento)
- Busca obtener tiempos de ejecución razonables para estas grandes aplicaciones, permitiendo de este modo realizar experimentos que de otro modo serían inviables, por tamaño, complejidad y/o precisión.
- Usa máquinas diseñadas específicamente para este propósito (Centros de Supercomputación: CeSViMa)
- La programación de estas máquinas tiene sus peculiaridades, tanto por el hardware, de cierta complejidad y diversidad, como por las aplicaciones, programadas por científicos e ingenieros





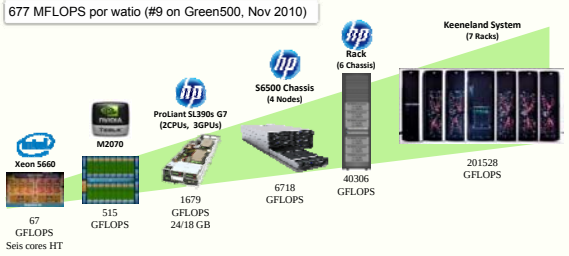

## Top500

- Máquinas con una gran potencia de cálculo que se emplean para ejecutar estas aplicaciones
- Las prestaciones se miden resolviendo un sistema de ecuaciones lineales denso, en 64 bits de precisión.
- El esquema general se basa en:
  - Múltiples nodos (memoria distribuida)
  - Cada nodo multiprocesador y cada procesador multicore (memoria compartida)
  - Conectados por una red de altas prestaciones.
- Frecuentemente se apoyan en coprocesadores para acelerar las rutinas más costosas.

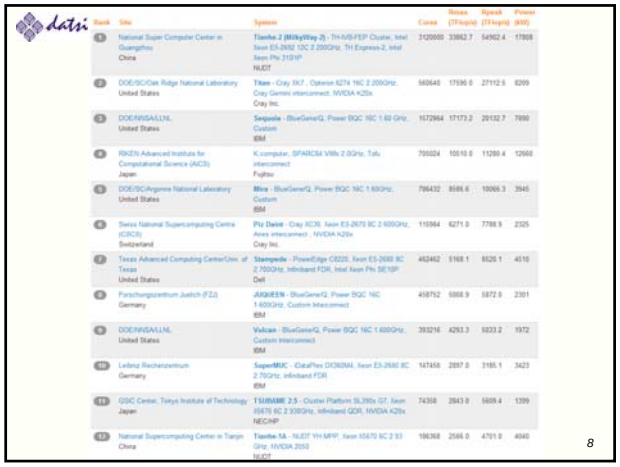
## Ejemplo de cluster HPC

201 TFLOPS en 7 racks

677 MFLOPS por watio (#9 on Green500, Nov 2010)







## Green 500

- El consumo energético es muy importante
- Por coste y por la potencia (PFlop/s) alcanzada
- Los primeros dan unos 2 GFlop/w
- Un Kwh (10-12 céntimos) permite 7,2 PFlop ¡¡7200 billones de flop!!
- Importancia creciente porque se quiere alcanzar el Exaflops (1000 Petaflops):
  - ¡¡500 MW es inaceptable!!



## Green 500

| Green500 Rank | MFLOPS/W | Site* | Computer* | Total Power (kW) |
|---|---|---|---|---|
| 1 | 4,503.17 | GSIC Center, Tokyo Institute of Technology | TSUBAME-KFC - LX 1U-4GPU/104Re-1G Cluster, Intel Xeon E5-2620v2 6C 2.100GHz, Infiniband FDR, NVIDIA K20x | 27.78 |
| 2 | 3,631.86 | Cambridge University | Wilkes - Dell T620 Cluster, Intel Xeon E5-2630v2 6C 2.600GHz, Infiniband FDR, NVIDIA K20 | 52.62 |
| 3 | 3,517.84 | Center for Computational Sciences, University of Tsukuba | HA-PACS TCA - Cray 3623G4-SM Cluster, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband QDR, NVIDIA K20x | 78.77 |
| 4 | 3,185.91 | Swiss National Supercomputing Centre (CSCS) | Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x<br>Level 3 measurement data available | 1,753.66 |
| 5 | 3,130.95 | ROMEO HPC Center - Champagne-Ardenne | romeo - Bull R421-E3 Cluster, Intel Xeon E5-2650v2 8C 2.600GHz, Infiniband FDR, NVIDIA K20x | 81.41 |
| 6 | 3,068.71 | GSIC Center, Tokyo Institute of Technology | TSUBAME 2.5 - Cluster Platform SL390s G7, Xeon X5670 6C 2.930GHz, Infiniband QDR, NVIDIA K20x | 922.54 |
| 7 | 2,702.16 | University of Arizona | iDataPlex DX360M4, Intel Xeon E5-2650v2 8C 2.600GHz, Infiniband FDR14, NVIDIA K20x | 53.62 |
| 8 | 2,629.10 | Max-Planck-Gesellschaft MPI/IPP | iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband, NVIDIA K20x | 269.94 |
| 9 | 2,629.10 | Financial Institution | iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband, NVIDIA K20x | 55.62 |
| 10 | 2,358.69 | CSIRO | CSIRO GPU Cluster - Nitro G16 3GPU, Xeon E5-2650 8C 2.000GHz, Infiniband FDR, Nvidia K20m | 71.01 |



## Programación HPC

- La arquitectura subyacente tiene una gran importancia para maximizar el rendimiento
- Uso de herramientas de análisis (*profilers*) para saber dónde optimizar y paralelizar.
- Memoria distribuida:
  - Se programa con MPI (paso de mensajes)
  - Hw: Destaca la red de interconexión
- Memoria compartida:
  - Se programa con OpenMP (*threads*)
  - Hw: Destaca sincronización y coherencia caches



## Programación HPC

- Coprocesador Intel Xeon Phi:
  - Se programa con OpenMP y Vectorización
  - Hw: Destaca uso de directorios y unidades vectoriales
- Coprocesador GPU (Nvidia y AMD):
  - Se programa con Cuda (a veces tb OpenCL)
  - Hw : Destaca la jerarquía de memoria y ejecución síncrona (SIMT)
- Se combinan todos entre sí: MPI+OpenMP, MPI+Copro, OpenMP+Copro, MPI+OpenMP+Copro