

COMPUTACIÓN DE ALTA PERFORMANCE

Curso 2025

Sergio Nesmachnow (sergion@fing.edu.uy)

Centro de Cálculo



OBJETIVOS DEL CURSO

- ⊕ Presentar los fundamentos de la computación de alto desempeño y su aplicación para la resolución eficiente de problemas con grandes requisitos de cómputo y en escenarios realistas.
- ⊕ Introducir los conceptos básicos de la computación paralela y distribuida.
- ⊕ Presentar conceptos, técnicas y herramientas de desarrollo de aplicación inmediata en la práctica.
- ⊕ Capacitar al estudiante para la resolución de problemas complejos aplicando técnicas de computación de alto desempeño.

CALENDARIO

Martes y jueves de 17 a 18:30 horas

Modalidad no presencial

Facultad de Ingeniería



Comienzo del curso: 11 de marzo de 2025

Fin del curso: julio de 2025

30 horas de exposiciones teóricas

20 horas de trabajo práctico

10 créditos

Eva del curso: <https://eva.fing.edu.uy/course/view.php?id=1064>



FACULTAD DE
INGENIERÍA
UDELAR

Enlaces de interés ▾ Cursos ▾



En este momento está usando el acceso para invitados

×

⋮

▾ Computación de Alta Perfo...

Novedades

Material

Práctico

Consultas

Formato de los proyectos

Novedades

▾ 11 y 13 de marzo de 2025

▾ 18 y 20 de marzo de 2025

▾ 25 y 27 de marzo de 2025

▾ 1 y 3 de abril de 2025

[Institutos](#) / [Computación](#) / [HPC](#)

Computación de alta performance

▾ Computación de Alta Performance

[Colapsar todo](#)

El objetivo del curso es introducir a los participantes en los conceptos de la computación de alto desempeño y su aplicación para la resolución eficiente de problemas científicos con grandes requisitos de cómputo y en escenarios realistas. Se presentan las técnicas de computación paralela y distribuida, se describen los diferentes tipos de arquitecturas de hardware existentes, enfatizando en arquitecturas de tipo cluster y técnicas de programación que permitan el uso de un conjunto de computadoras interconectadas en red como si fuera una única fuente de recursos computacionales. A lo largo del curso se presentan conceptos, técnicas y herramientas de desarrollo de aplicación inmediata en la práctica, que se ilustraran con ejemplos y proyectos concretos desarrollados en la Facultad de Ingeniería y en la Universidad de la República.

En las clases prácticas se presentan ejemplos y ejercicios utilizando la infraestructura de alto desempeño de la Facultad de Ingeniería: clusters de computadores, multiprocesadores de memoria compartida, servidores de cómputo

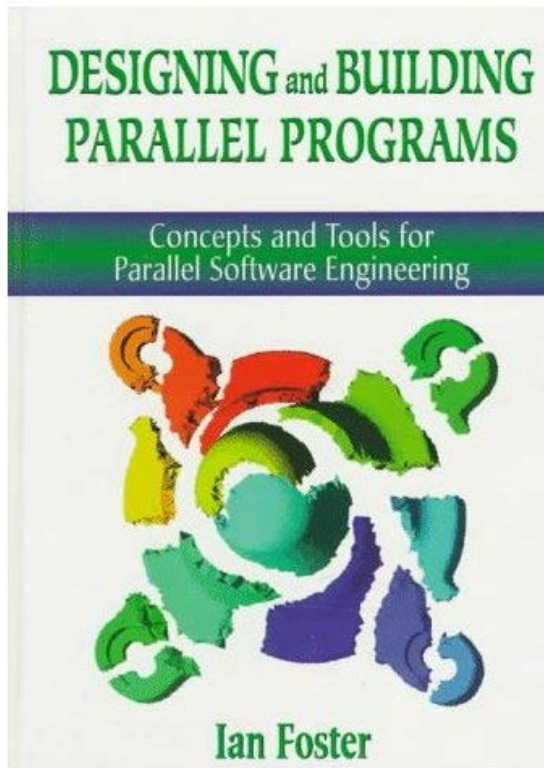
TEMARIO DEL CURSO

- El curso está dirigido a estudiantes, investigadores y profesionales de cualquier área técnica, que se enfrentan con problemas con altos requisitos de cálculo y procesamiento.
- El curso es **autocontenido** y puede ser realizado por participantes que no necesariamente hayan tenido contacto anterior con el tema.
- Temario:
 - Introducción
 - Descripción de arquitecturas paralelas
 - Modelos de programación paralela
 - Medidas de performance
 - Estudio de lenguajes, bibliotecas y entornos de desarrollo de aplicaciones paralelas y distribuidas
 - Computación distribuida en grid y cloud
 - Procesamiento de grandes volúmenes de datos (Big Data): MapReduce, Hadoop, Spark

- APROBACIÓN DEL CURSO
 - Trabajos prácticos (no obligatorios)
 - Ejercicios para aplicar las técnicas estudiadas.
 - Lenguaje C, bibliotecas OpenMP o MPI, otras tecnologías.
 - Proyecto final
 - Pequeño proyecto que aplicará las técnicas estudiadas.
 - Lenguaje C, bibliotecas OpenMP o MPI, otras tecnologías.
 - Artículo de síntesis
 - Descripción de actividades del proyecto.

BIBLIOGRAFÍA

- Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering, Ian Foster, Addison-Wesley, 1995.
- Disponible en <http://www-unix.mcs.anl.gov/dbpp/>



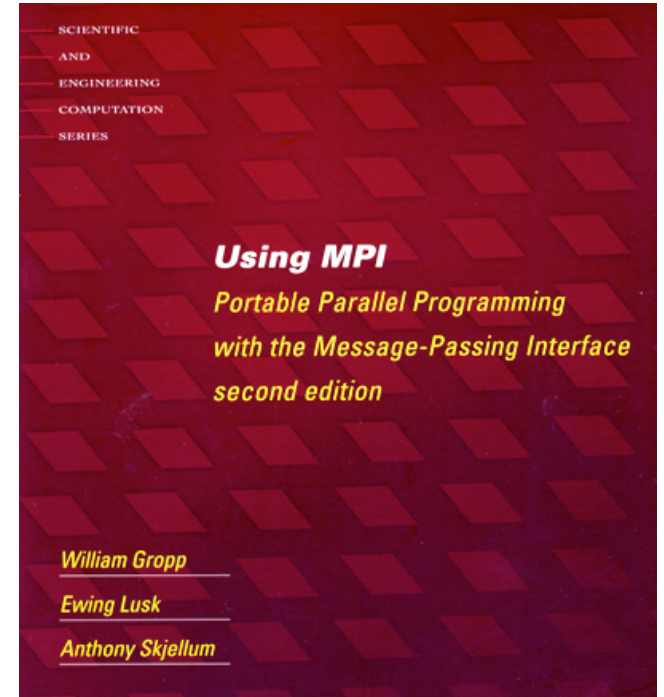
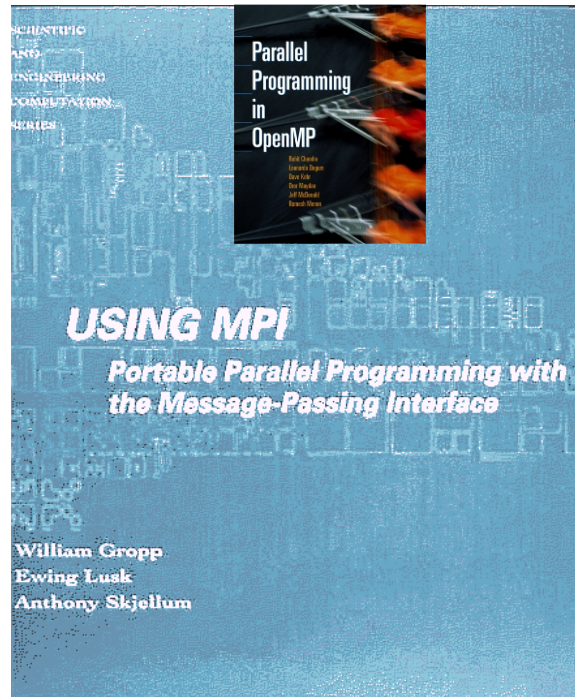
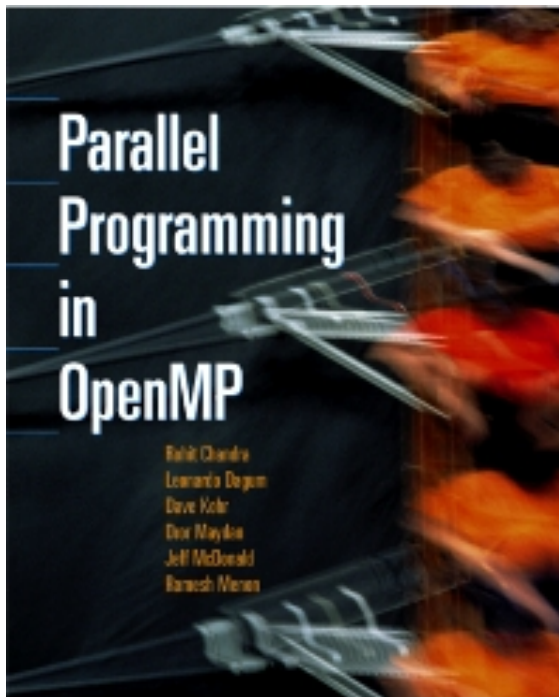
- El sitio forma parte de un proyecto que incluye:
 - El contenido del libro **Designing and Building Parallel Programs**.
 - Una colección de herramientas públicas para programación paralela.
 - Un conjunto de enlaces web para acceder a información sobre computación paralela.
 - Otros recursos educativos.

BIBLIOGRAFÍA



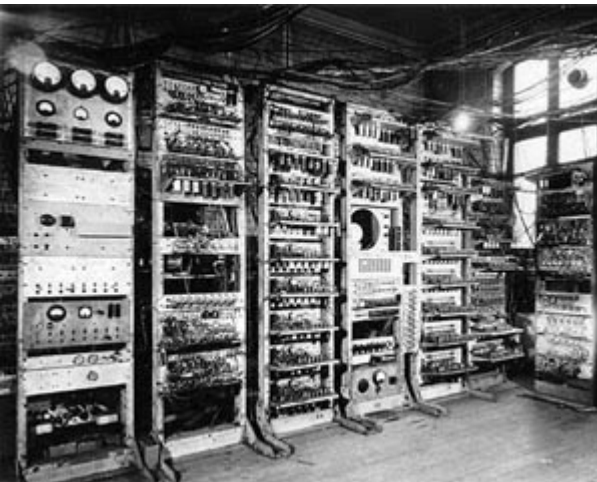
UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

- Parallel Programming in OpenMP. R. Chandra, R. Menon, L. Dagum, D. Kohr, D. Maydan, J. McDonald. Academic Press, 2000.
- Using MPI. Portable Parallel Programming with the Message Passing Interface, W. Gropp, E. Lusk, A. Skjellum, MIT Press, 1994 y 1999.

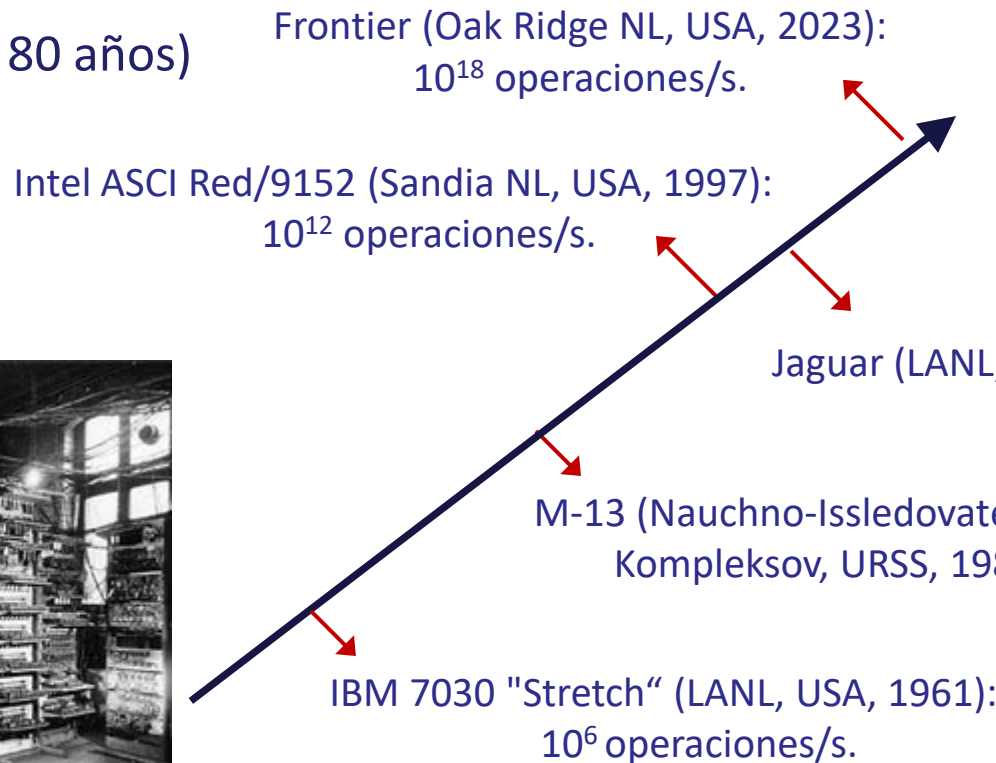


HPC timeline

(los asombrosos 80 años)



Colossus Mark 2 (UK, 1946):
5.000 operaciones/s.



El Capitan (USA, 2025):
 1742×10^{15} operaciones/s.

Jaguar (LANL, USA, 2010): 10^{15} operaciones/s.

M-13 (Nauchno-Issledovatesky Institute Vychislitelnyh
Kompleksov, URSS, 1984): 10^9 operaciones/s.

IBM 7030 "Stretch" (LANL, USA, 1961):
 10^6 operaciones/s.



TEMA 1

INTRODUCCIÓN a la COMPUTACIÓN de ALTA PERFORMANCE

1.1



CONCEPTOS PRELIMINARES

PARADIGMAS de COMPUTACIÓN

- Paradigmas:
 - Computación centralizada
 - Computación paralela
 - Computación distribuida
- **Categorías NO estrictamente disjuntas entre sí**
- Conceptualmente, tomando en cuenta la utilización de infraestructura y mecanismos de diseño, implementación y ejecución, los enfoques opuestos son los de **computación centralizada** y **computación distribuida**
- El campo de la computación paralela se superpone con el de computación distribuida.
- Los recientes paradigmas de computación en cloud son un tipo particular de computación distribuida.

COMPUTACIÓN CENTRALIZADA

- Caracterizada por poseer **todos los recursos centralizados** en un único sistema físico. Cómputo, memoria y almacenamiento están compartidos y se encuentran fuertemente acoplados con un único sistema operativo.
- El procesamiento de datos se realiza en una ubicación central, con acceso a la infraestructura mediante terminales conectadas al computador central.
- El manejo de los periféricos es directo desde el computador central (puede existir cierta independencia para manejarlos por un *servidor de terminales*), centralizado o distribuido en una red de área local.



COMPUTACIÓN CENTRALIZADA

- Principales ventajas:
 - **Gran seguridad**: provee un mecanismo de control del procesamiento y acceso a los datos centralizado en una ubicación física
 - Las terminales de acceso proveen un nivel básico de **tolerancia a fallos** (datos y procesamiento pueden estar disponibles desde terminales alternativas)
 - Son sistemas **sobredimensionados** para tolerar picos de utilización: usuarios cuentan con prestaciones superiores a las que necesitan sus aplicaciones
- Principales desventajas:
 - **Disponibilidad y fiabilidad** del computador central (controla procesamiento y acceso a los datos). El sistema completo resulta inaccesible e inutilizable ante una falla de control de la unidad central
 - El paradigma depende fuertemente de la administración y de los recursos provistos a los usuarios. Al alcanzar los límites de utilización (por limitaciones físicas o por exceder las capacidad de procesamiento multiusuario) **no es sencillo escalar** las capacidades computacionales y/o de almacenamiento
 - **Costo** elevado (millones USD)

COMPUTACIÓN CENTRALIZADA: HISTORIA

- Dominó el mundo de la computación hasta la aparición de las computadoras personales a inicio de la década de 1980
- Provee una solución muy útil para implementar sistemas multiusuario, mediante múltiples accesos simultáneos al recurso de cómputo centralizado
- Estuvo descartada como modelo de computación durante 15 años, y se comenzó a aplicar nuevamente para manejo transaccional de comercio electrónico
- A partir del 2000 el desarrollo de Linux permitió implementar soluciones de cómputo basadas en la utilización de (cientos de) máquinas virtuales en un único mainframe ... y las infraestructuras de computación centralizada volvieron a la vida ...

COMPUTACIÓN CENTRALIZADA: HISTORIA

- Resurgió desde 2010 como alternativa para el acceso eficiente a recursos de cómputo mediante *clientes livianos* que no requieren instalación de software pesado por parte del usuario
 - Permite acceder a cómputos realizados en un servidor central y provee amplia usabilidad de las aplicaciones desde múltiples dispositivos
- La evolución tecnológica ha seguido oscilando: los desarrolladores incluyen más lógica en los clientes para aprovechar el poder de cómputo disponible en los dispositivos, implementando clientes ricos que reducen el cómputo remoto en los servidores
- Los mainframes son aún utilizados para cierto tipo de aplicaciones críticas (transacciones financieras, aplicaciones de seguridad y defensa). El acceso ya no es mediante dispositivos de tipo terminal sino usando emuladores por software, a través de interfaces de aplicación web, o mediante protocolos web específicos.

- **Modelos híbridos:** la computación centralizada sigue utilizándose hoy en día en datacenters que emplean modelos híbridos
- En un modelo híbrido, ciertas aplicaciones (e.g., navegadores web) ejecutan localmente [distribuidas], accediendo a servicios proporcionados por otras aplicaciones (en general, sistemas informáticos más complejos) que ejecutan de forma centralizada en el datacenter
- El modelo de *hosted computing* aplica computación centralizada para alojar cómputo y almacenamiento en poderosos servidores de hardware, evitando a usuarios y organizaciones las responsabilidades de acceso, mantenimiento y seguridad de la información
 - Estos servicios se proveen bajo demanda y suscripción por parte de un proveedor de servicios de aplicación

COMPUTACIÓN PARALELA y DISTRIBUIDA

- En lugar de emplear el modelo estándar de computación utilizando un único recurso de procesamiento, aplican técnicas de **computación concurrente y paralelismo** para abordar problemas complejos utilizando **múltiples recursos de cómputo** simultáneamente
- Complejidad de problemas: gran escala y/o que manejan grandes volúmenes de datos
- Se trabaja sobre un conjunto de recursos de cómputo interconectados por una red de área local (LAN) o de área global (Internet)
- Los sistemas de computación paralela y distribuida están enfocados en la resolución de problemas con requisitos intensivos de cómputo (*CPU-intensive*) o con manejo de datos intensivo (*data-intensive*) y son sistemas basados en comunicación en redes (*network centric*)

COMPUTACIÓN PARALELA

- Aplica un modelo de **procesos concurrentes** en ejecución simultánea, sobre una infraestructura computacional **altamente acoplada** que se encuentra en una única ubicación física.
- La cooperación entre los procesos en ejecución, con el objetivo de resolver un problema global, se realiza mediante **comunicaciones** y **sincronizaciones**, utilizando algún recurso compartido (*mecanismos de IPC, memoria compartida*) o memoria distribuida (utilizando pasaje de mensajes explícitos).
- Un sistema computacional capaz de proveer el soporte para computación paralela se denomina **computador paralelo**. Los programas que ejecutan en un computador paralelo utilizando múltiples procesos simultáneos se denominan programas paralelos, por oposición a la computación secuencial tradicional. El proceso de desarrollar e implementar programas paralelos se denomina programación paralela.

COMPUTACIÓN PARALELA: MOTIVACIÓN

- Importancia de poder satisfacer los requisitos crecientes de poder de cómputo
 - Problemas inherentemente complicados
 - Modelos complejos
 - Grandes volúmenes de datos
 - Capacidad de respuesta en tiempo limitado (sistemas de tiempo real)
- Procesamiento paralelo
 - Varios procesos cooperan para resolver problema común
 - Aplicación de técnicas de **división de tareas** o de **datos** para reducir el tiempo de ejecución de un proceso o una aplicación, mediante la resolución simultánea de algunos de los subproblemas generados

- Computador paralelo
 - Conjunto de procesadores capaces de trabajar cooperativamente en la resolución de problemas computacionales
 - La definición incluye un amplio espectro: supercomputadoras, procesadores masivamente paralelos (MPP), clusters, etc.
 - Característica fundamental: disponibilidad de **múltiples** recursos de cómputo
- Computación de alto desempeño
 - Ha dejado de ser exótica para ser ubicua
 - Posibilitada por avances en diferentes tecnologías:
 - Poder de procesamiento (microprocesadores)
 - Redes (comunicación de datos)
 - Desarrollo de bibliotecas e interfaces para programación

- El tipo de problemas complejos para los cuales es apropiado aplicar el paradigma de computación paralela y distribuida incluye, entre otros:
 - simulaciones que involucran modelos complejos y de gran escala;
 - problemas cuya resolución demanda grandes requisitos de CPU y/o memoria;
 - problemas y aplicaciones que manejan y procesan grandes (inclusive enormes) volúmenes de datos;
 - aplicaciones que manejan y/o deben dar soporte a un gran número de usuarios;
 - aplicaciones y sistemas ubicuos y concurrentes basados en agentes.

- La clave consiste en la utilización de **múltiples recursos de cómputo** por parte de múltiples procesos que ejecutan **concurrentemente**, de modo **cooperativo** para resolver un problema complejo común
- La cooperación se logra a través de **comunicaciones y sincronizaciones**
- Los recursos de cómputo pueden organizarse en sistemas interconectados por redes de área local o de área global, orientándose a dos modelos específicos de computación paralela/distribuida:
 1. Sistemas de computación de alta performance (HPC)
 2. Sistemas de computación de alto rendimiento (HTC)

- **Sistemas de computación de alto desempeño/alta performance (HPC):**
 - utilizados para computación científica
 - enfatizan la importancia de la eficiencia (performance), considerando el número de operaciones realizadas por unidad de tiempo
- Han incrementado sus velocidades de procesamiento:
 - **1990:** GFLOPS (10^9 operaciones de punto flotante por segundo); **2010:** TFLOPS (10^{12}); **2015:** PFLOPS (10^{15}); **2022:** EFLOPS (10^{18}); bajo la demanda de poder de cómputo para aplicaciones científicas, ingeniería y manufactura.
- En la actualidad, el número de usuarios de sistemas de HPC es bajo (10%), pero con gran uso (80%) de recursos computacionales.
- Por otra parte, un mayor número de usuarios utiliza algún tipo de computación distribuida en Internet para ejecutar aplicaciones simples: búsqueda en la web, transacciones comerciales, redes sociales, etc.

- **Sistemas de computación de alto rendimiento**
(*High Throughput Computing* – HTC)
- Utilizados para aplicaciones de procesamiento transaccional masivo y aplicaciones comerciales a gran escala
- En la actualidad, el desarrollo de sistemas de computación orientadas a aplicaciones de mercado y de altas prestaciones está más asociado con el paradigma de HTC que con el de HPC

1.2



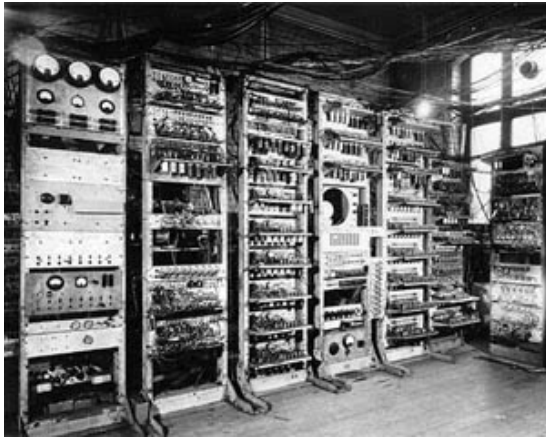
COMPUTACIÓN DE ALTO DESEMPEÑO

COMPUTACIÓN de ALTO DESEMPEÑO

- Sistemas de computación de alto desempeño/alta performance (HPC):
 - utilizados para computación científica
 - enfatizan la importancia de la eficiencia (performance), considerando el número de operaciones realizadas por unidad de tiempo
- Propulsados por las mejoras e innovaciones tecnológicas, han incrementado sus velocidades de procesamiento:
 - 1990: GFLOPS (10^9 operaciones de punto flotante por segundo)
 - 2010: TFLOPS (10^{12})
 - 2015: PFLOPS (10^{15})
 - 2022: EFLOPS (10^{18})



EVOLUCIÓN TECNOLÓGICA



Colossus 2 (UK), primer computador paralelo:
50.000 op/s

1938

1946

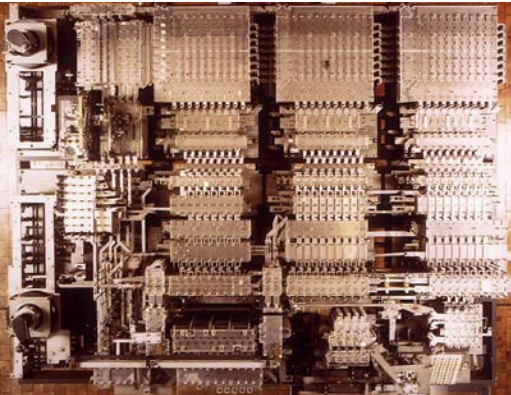
1948

IBM NORC (Columbia Univ, USA), reloj de 1 μ s., 67.000 op/s

1954

1964

Zuse Z1 (Ale), primer computador mecánico: 1 op/s



ENIAC (USA), 5.000 op/s

★
MEGAFLOP COMPUTER

MEGAFLOP COMPUTER
IBM 7030 "Stretch"
(LANL, USA), 1.2 MFLOPS



EVOLUCIÓN TECNOLÓGICA

★
GIGAFLOP
COMPUTER

Cray-2/8 (LANL, USA),

3.9 GFLOPS



1985-89



1984

1997

GIGAFLOP COMPUTER

M-13 (Nauchno-Issledovatesky Institute
Vychislitelnyh Kompleksov, URSS): 2.4 GFLOPS

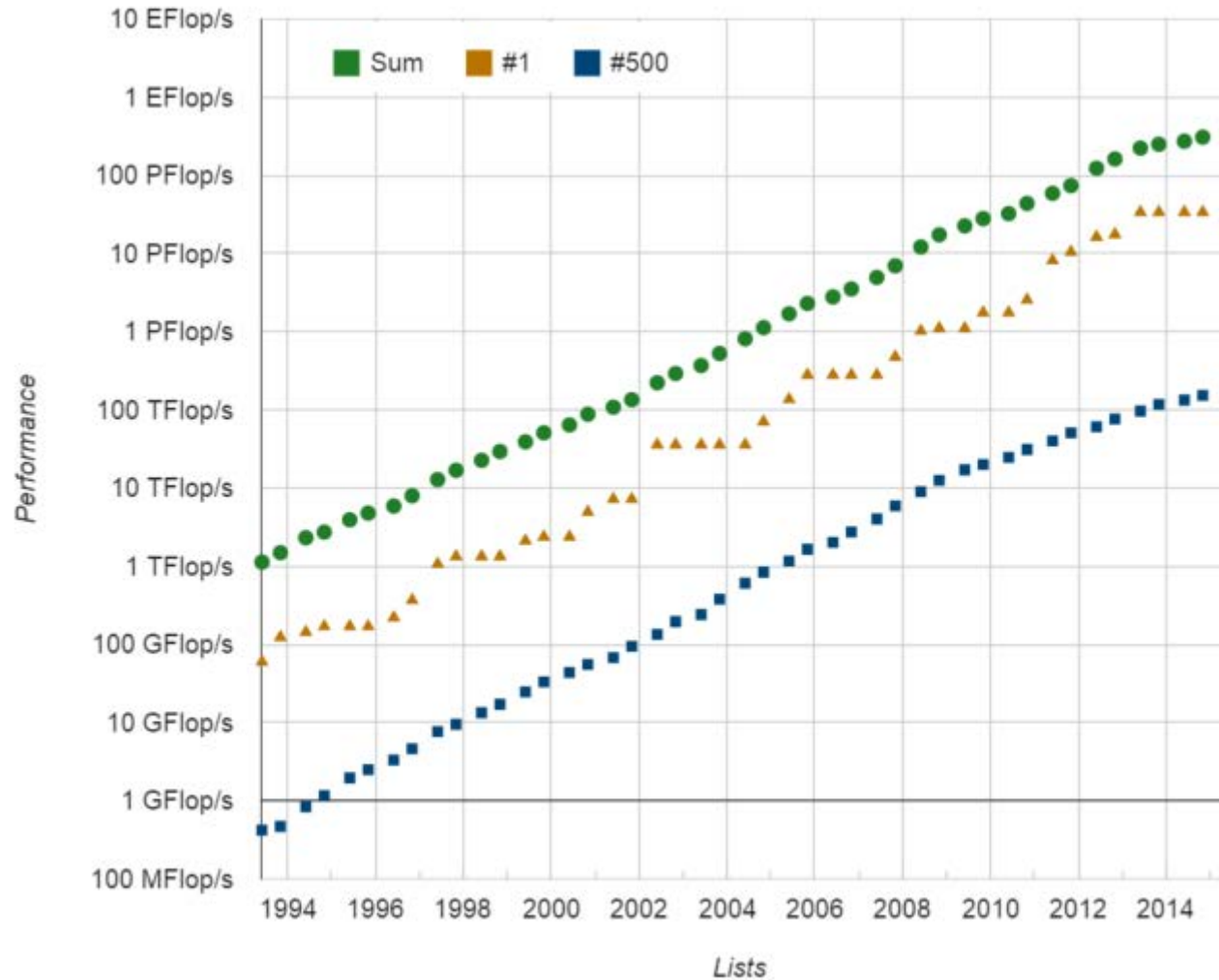


TERAFLOP COMPUTER
Intel ASCI Red/9152 (Sandia NL,
USA, 1997): 1.338 TFLOPS

TERAFLOP
COMPUTER



EVOLUCIÓN TECNOLÓGICA



Incremento de poder de cómputo (Top500, **escala logarítmica!**)

EVOLUCIÓN TECNOLÓGICA

- Similar comportamiento para otros indicadores
 - Frecuencia de relojes
 - Densidad de circuitos en chips de procesadores
 - Capacidad de almacenamiento secundario
 - Capacidad de transmisión por bus/red
- Siguen el mismo comportamiento **exponencial**, con diferentes pendientes



EVOLUCIÓN TECNOLÓGICA

- Junio de 2008: Petaflop supercomputer (Peta = 10^{15} = 1000000000000000000)
 - **Roadrunner** (LANL, USA), 1.026 petaflop/s
 - BladeCenter QS22 Cluster, con PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz
 - Híbrido: 6,562 dual-core AMD Opteron® y 12,240 Cell chips
 - 98 terabytes de memoria, 278 IBM BladeCenter® racks (560 m²)
 - 10,000 conexiones (Infiniband y Gigabit Ethernet), 90 km de fibra óptica



EVOLUCIÓN TECNOLÓGICA

- Julio de 2015: Tihanhe-2 (National University of Defense Technology, China)
 - Pico de desempeño real LINPACK: 33.86 petaflops
 - Intel cluster, pico teórico: 54.9 petaflops
 - 16.000 nodos, con dos procesadores Intel Xeon IvyBridge y tres Xeon Phi
 - 3.120.000 núcleos y 1.024 terabytes de memoria
 - Red propietaria TH Express-2, sistema operativo Kylin Linux



EVOLUCIÓN TECNOLÓGICA

- Julio de 2016: Sunway TaihuLight (National Supercomputing Center, China)
 - Pico de desempeño real LINPACK: **93 petaflops**
 - Pico teórico de desempeño: 125 petaflops
 - **40.960 nodos**, procesadores SW26010 manycore (260 núcleos), arquitectura ShenWei (RISC de 64-bits).
 - **10.649.600 núcleos** y 1.310 terabytes de memoria



Sunway TaihuLight

- 40.960 nodos, procesadores SW26010 1.45GHz manycore (256 núcleos de cómputo, 4 de administración), arquitectura ShenWei (RISC de 64-bits)
 - 10.649.600 núcleos, cada núcleo tiene 64 KB de memoria scratchpad (NUMA) para datos y 12 KB para instrucciones.
 - Los núcleos se comunican por red en chip, y no por caché jerárquica tradicional
- Red propietaria, Sunway Network: tecnología PCIe 3.0
 - 16 GB/s de pico de ancho de banda nodo a nodo, latencia de 1 μ s
 - Comunicaciones MPI a 12 GB/s (similar a InfiniBand EDR o 100G Ethernet)
 - Sistema operativo Raise OS 2.0.5, basado en Linux.
 - Incluye versión personalizada de OpenACC 2.0 para paralelización de código



Sunway TaihuLight

- Consumo energético: con carga máxima, 15.37 MW (6 GFLOPS/Watt)
 - Primeros lugares en Green500 en términos de performance/energía
- Totalmente construida en China, no usa procesadores Intel
- Aplicaciones: prospección de petróleo, ciencias de la vida, estudios climáticos, diseño industrial, investigación de fármacos
 - Tres aplicaciones científicas en TaihuLight han sido seleccionadas como finalistas del Gordon Bell Prize (mejor desempeño o escalabilidad, aplicada a problemas científicos y de ingeniería), alcanzando un desempeño de entre 30 y 40 petaflops
- 2016 marcó la primera vez que un país tiene más supercomputadores en Top500 que USA
 - China: 167, USA: 165



Summit

- Julio de 2018: Summit (Oak Ridge National Laboratory, USA)
 - Pico de desempeño real LINPACK: 143.5 petaflops.
 - Pico teórico de desempeño: 200,8 petaflops.
 - 4.608 nodos, procesadores IBM POWER9 (22 núcleos), 27.648 GPUs NVIDIA Volta V100s. Red InfiniBand 100G Mellanox.
 - 2.397.824 núcleos y >2PB de memoria RAM.



Fugaku

- Julio de 2020: Fugaku (RIKEN Center for Computational Science, Japón)
 - Pico de desempeño real LINPACK: 442 petaflops.
 - Pico teórico de desempeño: 537 petaflops.
 - 158.976 CPUs con procesadores Fujitsu A64FX (48 núcleos) arquitectura ARM,
 - 7.630.848 núcleos y >5PB de memoria RAM.
 - Desempeño de 1.42 exaflops en el benchmark HPL-AI



Frontier

- Junio de 2022: Frontier (Oak Ridge National Laboratory , USA)
 - Pico de desempeño real LINPACK: 1102 petaflops (exaflop computer).
 - Pico teórico de desempeño: 1685 petaflops.
 - 8.730.112 núcleos en CPUs con procesadores AMD Optimized 3rd Gen EPYC 64C 2GHz y aceleradores AMD Instinct 250X (GPUs)



Frontier (noviembre 2023)

- Noviembre de 2023: Frontier (Oak Ridge National Laboratory , USA)
 - Pico de desempeño real LINPACK: 1194 petaflops (exaflop computer).
 - Pico teórico de desempeño: 1685 petaflops.
 - 8.699.904 núcleos en CPUs con procesadores AMD Optimized 3rd Gen EPYC 64C 2GHz y aceleradores AMD Instinct 250X

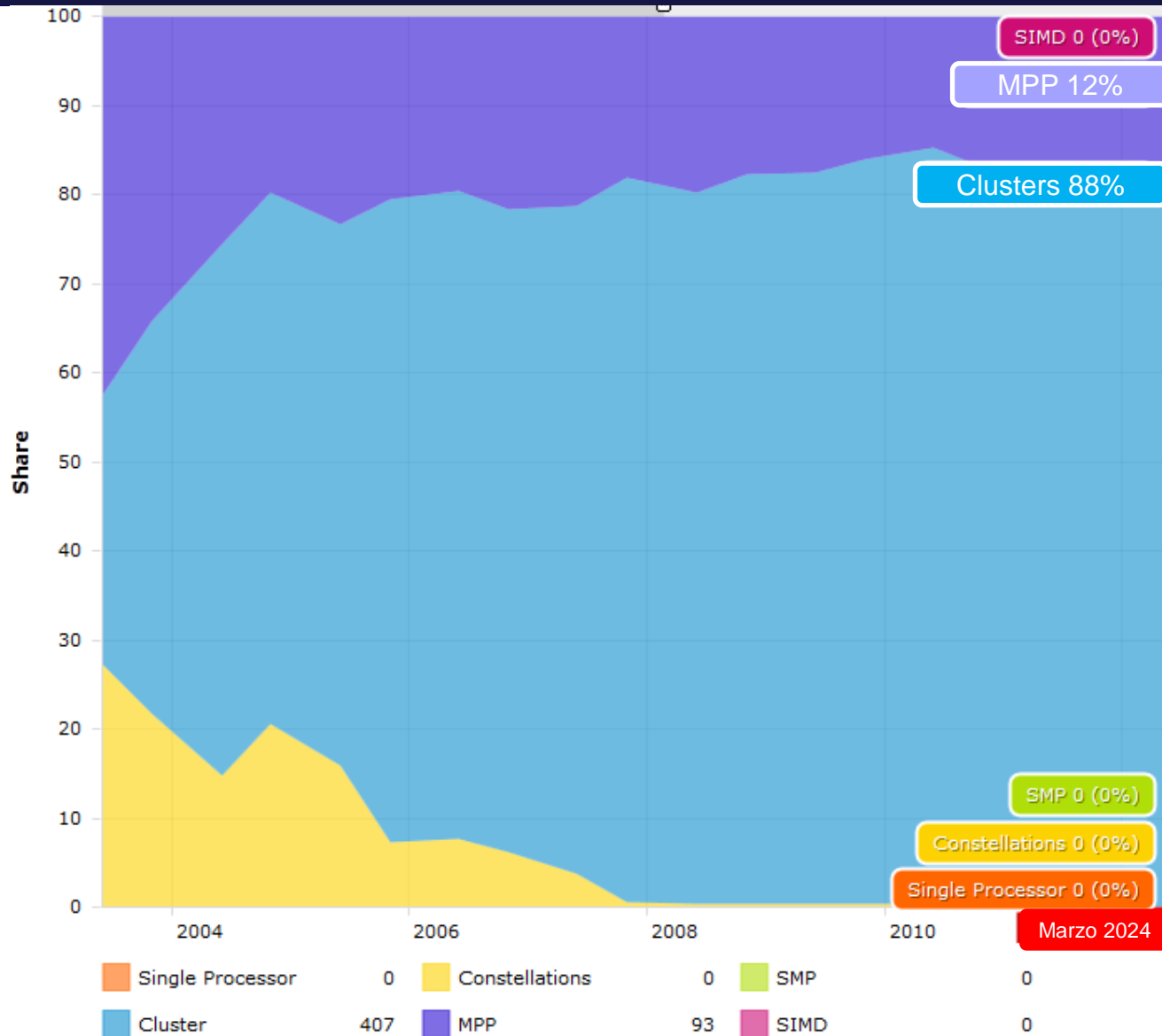


El Capitan (noviembre 2024)

- Noviembre de 2024: El Capitan (Lawrence Livermore National Lab, USA)
 - Pico de desempeño real LINPACK: 1.742 exaflops.
 - Pico teórico de desempeño: 2.746 exaflops.
 - 11.039.616 núcleos en CPUs con procesadores AMD 4th Gen EPYC 24C 1.8GHz y aceleradores AMD Instinct MI300A.

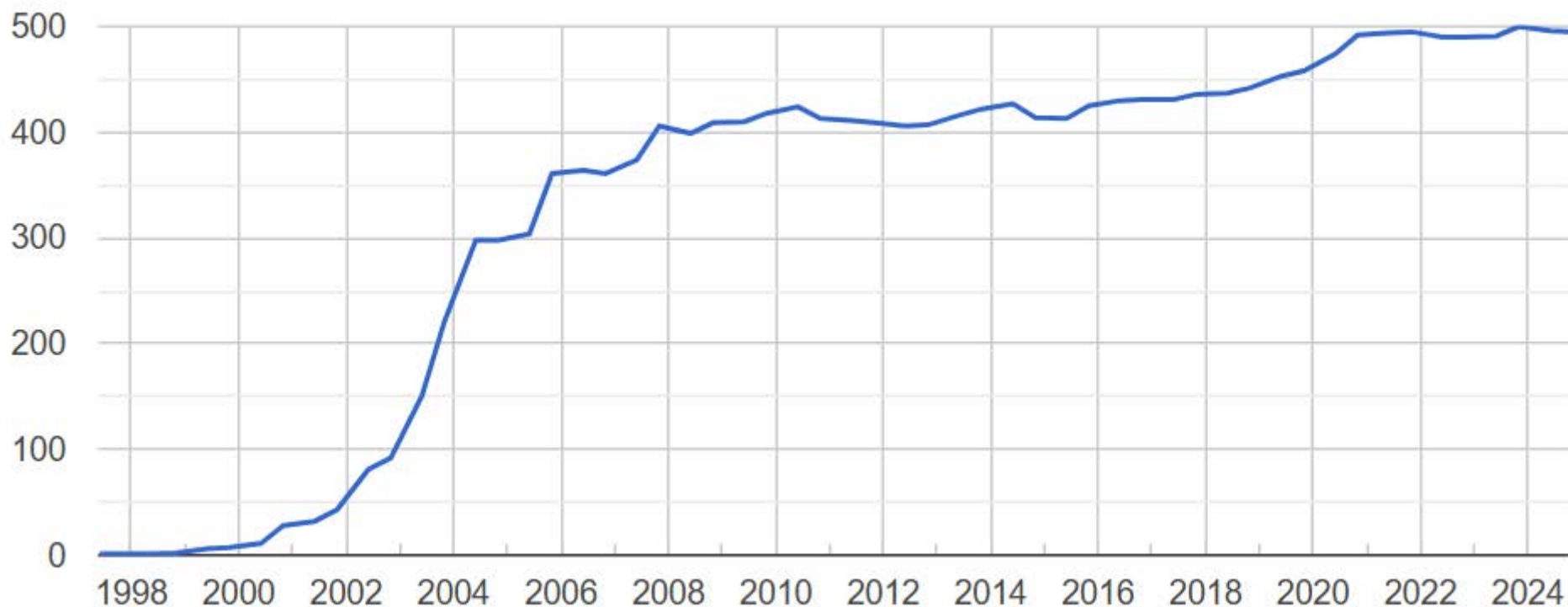


EVOLUCIÓN TECNOLÓGICA: ARQUITECTURAS

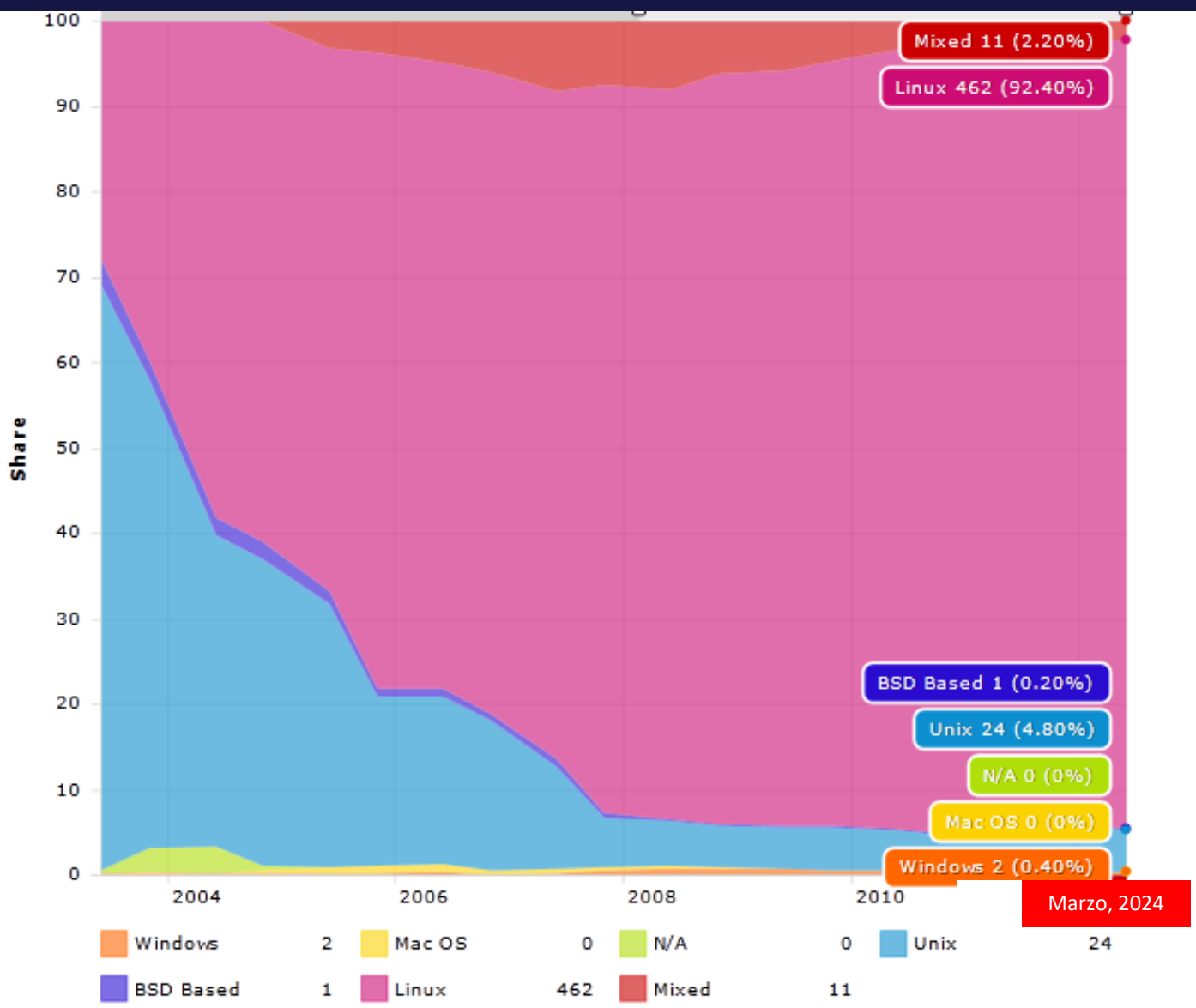


EVOLUCIÓN TECNOLÓGICA: Clusters

- En 2025, el 88% de los 500 sistemas más potentes son clusters (agregaciones de servidores de cómputo de alto desempeño)



EVOLUCIÓN TECNOLÓGICA: SISTEMAS OPERATIVOS

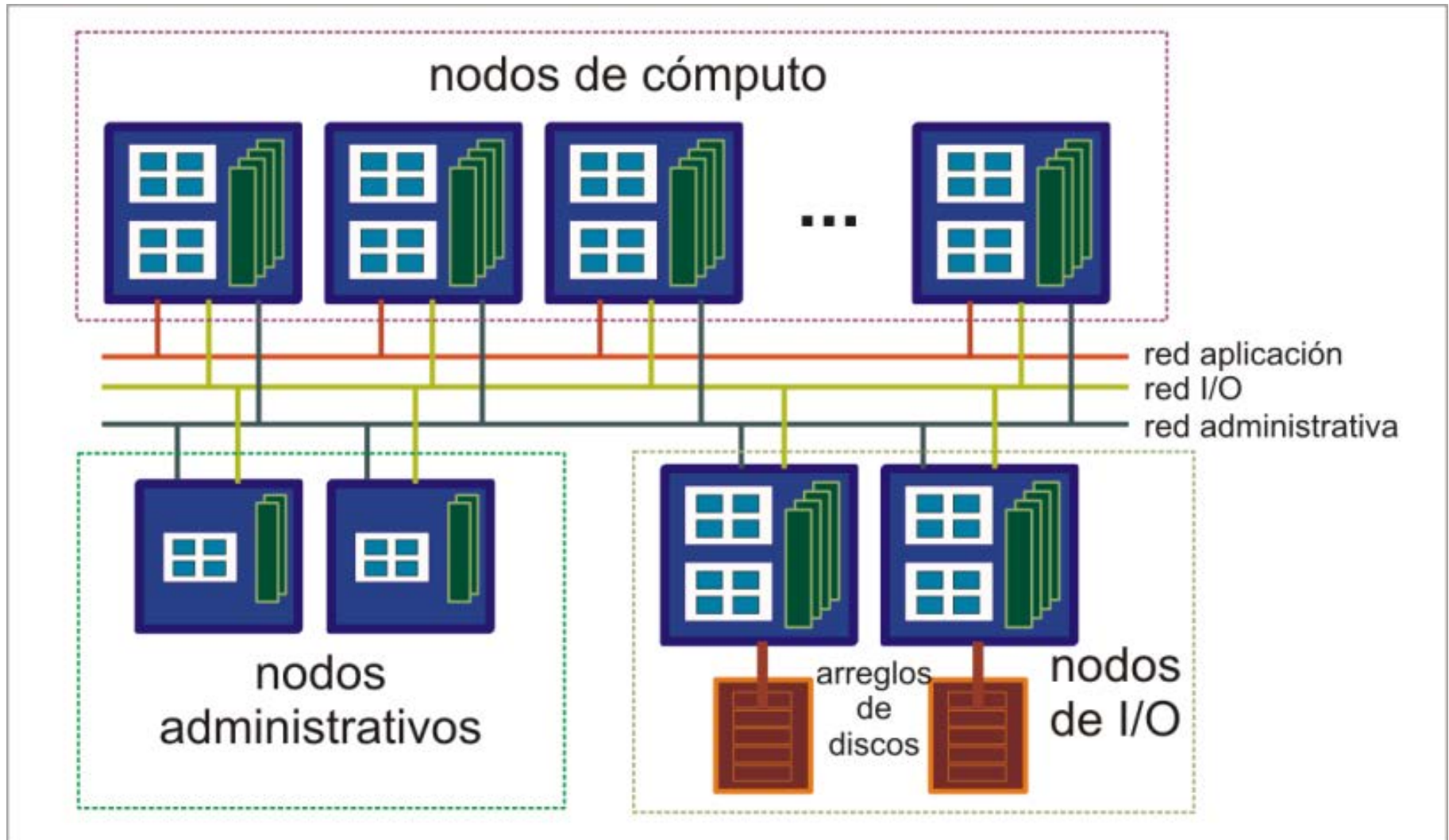


- La tecnología ha avanzado, permitiendo disponer de máquinas paralelas “caseras”
 - **Clusters** de computadores de bajo costo
- Internet surge como una fuente potencial de recursos de computación ilimitados
 - Internet 2 amplió la banda y la potencia de comunicación entre equipos
- Se desarrolló la tecnología **grid** y luego la tecnología **cloud**:
 - Permiten compartir recursos informáticos (locales o remotos) como si fueran parte de un único computador
 - Brinda capacidad de gestionar y distribuir la potencia de cálculo disponible en la mediana empresa
 - Empresas de renombre e investigadores trabajan en diseño de soluciones tecnológicas en este sentido

INFRAESTRUCTURA

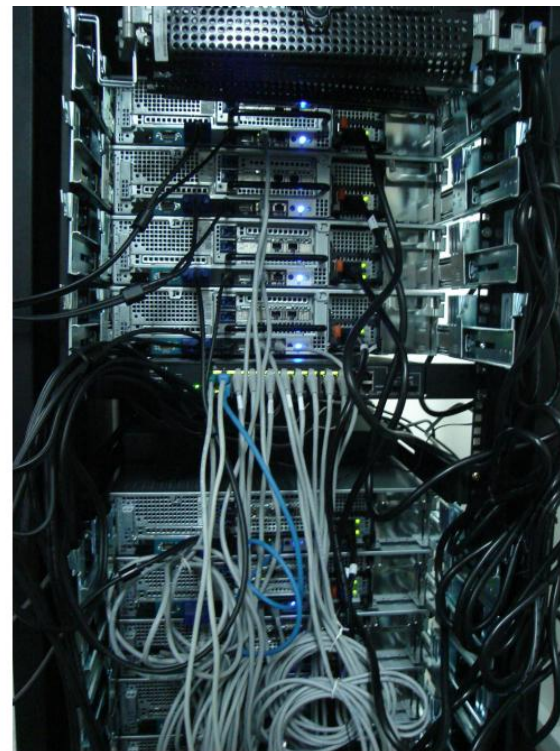
- Las alternativas mencionadas constituyen opciones realistas para tratar de lograr capacidad de cómputo competitivo
 - Obviamente, sin llegar a los límites de los mejores supercomputadores del Top500
- Sin embargo, permiten resolver problemas interesantes en los entornos **académicos**, **industriales** y **empresariales**, con una infraestructura de bajo costo

CLUSTERS

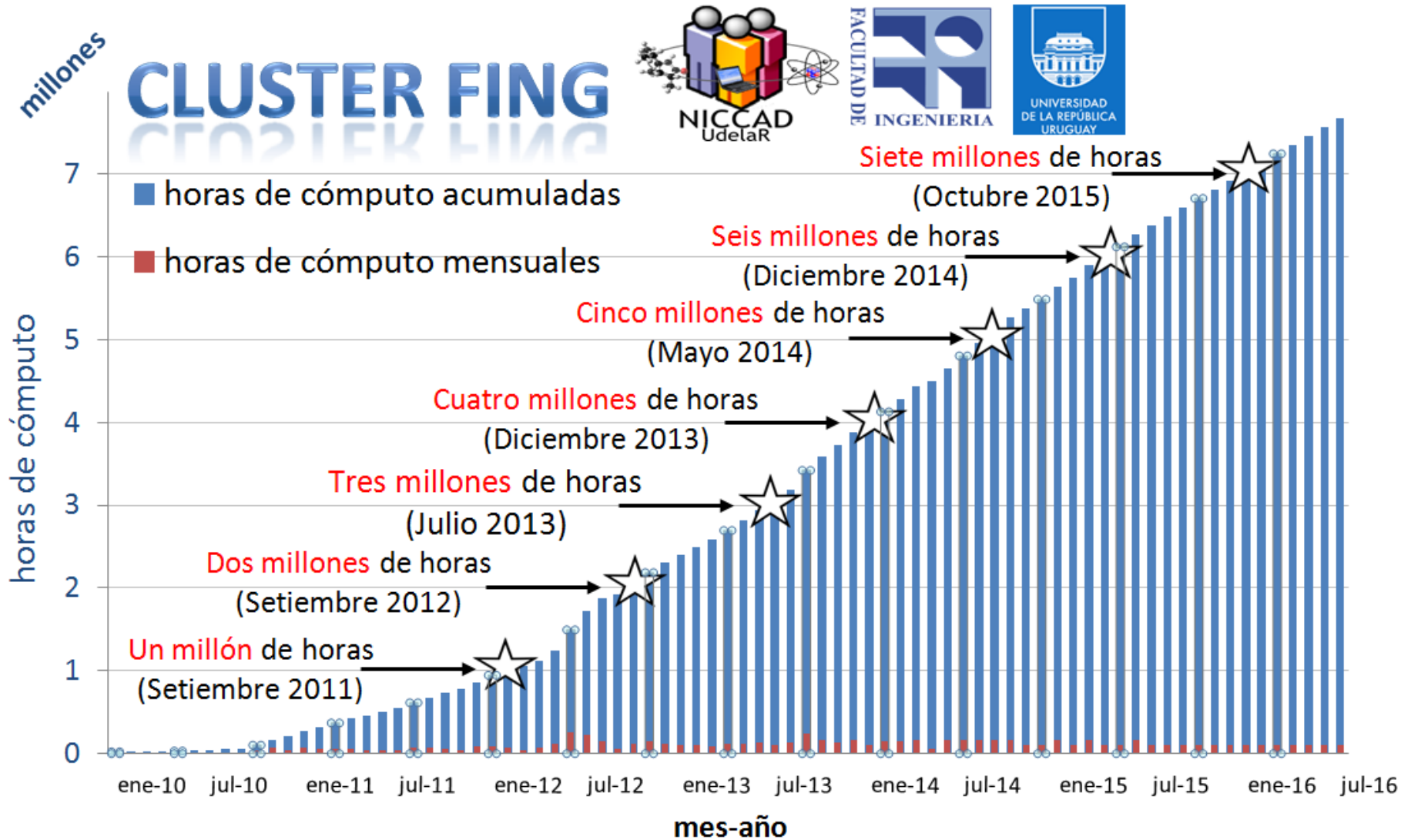


EL CLUSTER FING

- Infraestructura de cómputo de alto desempeño de la Facultad de Ingeniería
 - “Fortalecimiento de Equipamientos para la Investigación” (CSIC, 2008)
 - Objetivo: disponer de una plataforma computacional capaz de abordar eficientemente problemas complejos
 - Operativo entre 2009 y 2019.



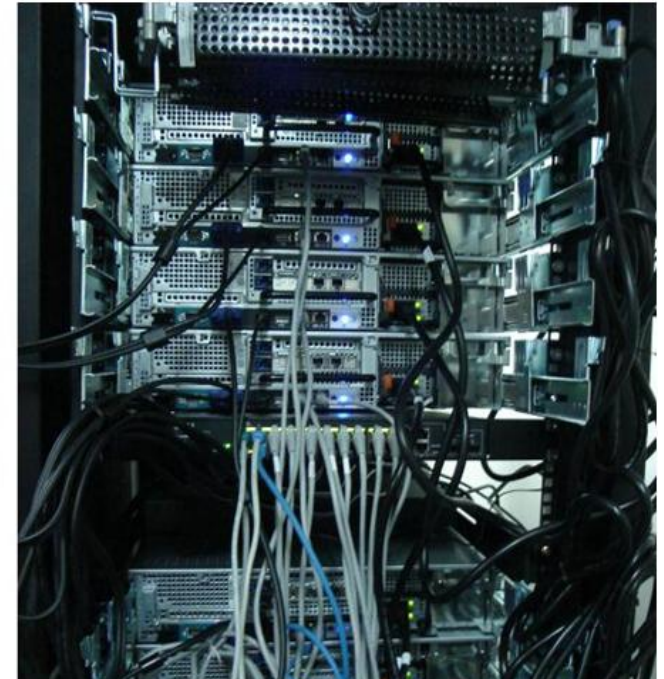
CLUSTER FING: UTILIZACIÓN



CLUSTER FING: UTILIZACIÓN



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



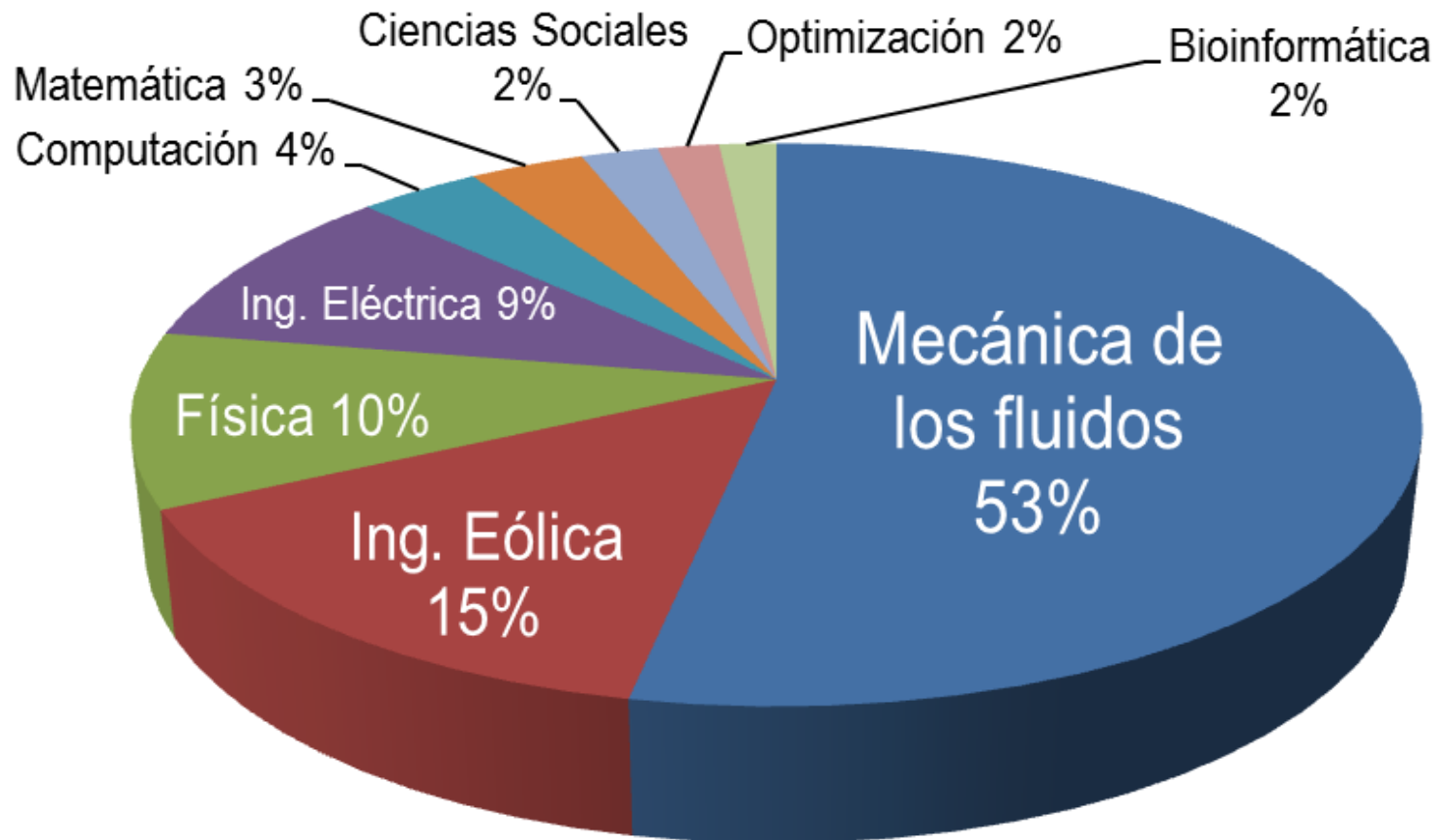
Cluster FING: 11.000.000 de horas de cómputo efectivo (2018)
<http://www.fing.edu.uy/cluster>

CLUSTER FING: UTILIZACIÓN

<i>Indicador</i>	<i>Valor</i>
Número de usuarios	> 500, de más de 20 países
Horas efectivas de cómputo	>11.000.000 horas (> 1255 años)
Proyectos académicos	> 200
Tesis de posgrado	> 80 Maestría y > 50 Doctorado
Grupos de investigación	> 50
Proyectos y trabajos de grado	> 300
Convenios	> 50
Atrículos publicados	> 1000

Estadísticas en el período 2009 – 2018

Áreas de aplicación (<http://www.fing.edu.uy/cluster>)



- Gran volumen de trabajos multidisciplinarios

Participación en infraestructuras distribuidas

- Infraestructura Grid Latinoamericano-Europea (GISELA)
 - Cluster FING implementa el Grid Nacional (Uruguay) y brinda soporte a Virtual Research Communities (e-infraestructura y servicios basados en aplicaciones)
- Ourgrid
 - Cluster FING integra la comunidad Ourgrid de computación voluntaria
- Redes de computación voluntaria con instituciones de enseñanza e investigación internacionales
 - Federación (OpenNebula) con Universidad de Buenos Aires
- Participación activa en la comunidad HPCLATAM



EL CENTRO NACIONAL DE SUPERCOMPUTACIÓN (Cluster-UY)

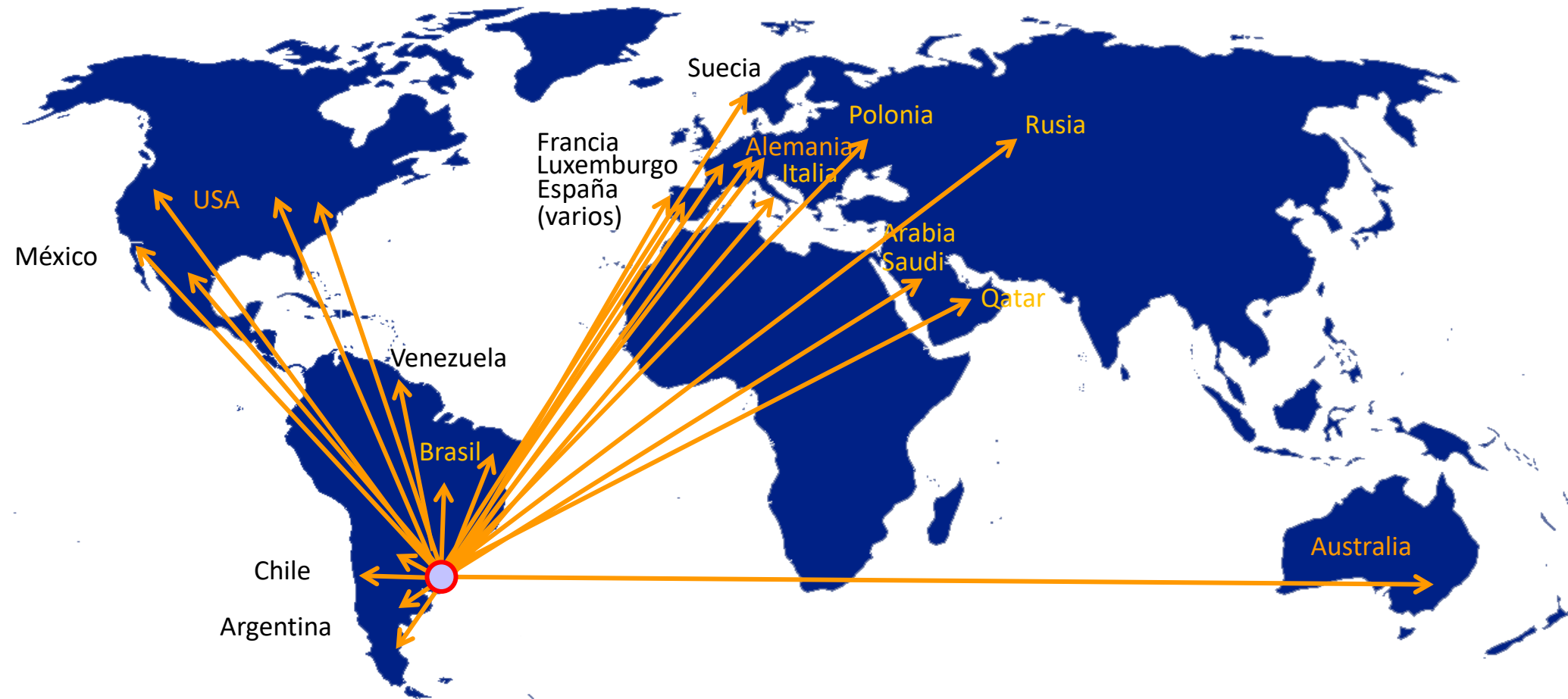
- Proyecto de grandes equipos (ANII-UdelaR, 2017)
 - Autofinanciado y autogestionado
- 32 nodos interconectados: más de 1700 núcleos de cómputo de CPU y 125.000 núcleos de cómputo de GPU de última generación
- Capacidad de procesamiento estimada en más de 100 TeraFlops (100 billones de operaciones por segundo)
- 20x Cluster FING

El mayor poder de cómputo disponible en el país

<http://cluster.uy/>



COLABORACIONES ACTUALES en HPC



- Más de 25 colaboraciones internacionales activas con proyectos en curso

1.3



COMPUTACIÓN DE ALTO RENDIMIENTO

- Presta mayor atención al manejo de **grandes flujos de datos** (búsquedas en Internet y uso de servicios web), realizados en forma masiva por **muchos usuarios simultáneamente**
- A diferencia de HPC, el objetivo deja de ser lograr un alto desempeño en número de operaciones y tiempo de ejecución
- Busca proporcionar el mejor **rendimiento**, evaluado por el número de **tareas que el sistema puede completar por unidad de tiempo**, permitiendo la atención de un mayor número de usuarios
- HTC busca mejorar en términos de procesamiento neto de tareas ejecutadas fuera de línea (procesamiento batch), y también se enfoca en proveer buenas soluciones en línea considerando los problemas y restricciones de costo monetario, ahorro energético, seguridad y confiabilidad, en especial para infraestructuras de múltiples centros de cómputo y repositorios de datos distribuidos

- Sistemas de computación de alto rendimiento:
 - Redes peer-to-peer (P2P)
 - Datacenters distribuidos
 - Sistemas de computacion cloud
- Redes P2P: creadas para compartir y distribuir contenidos, fueron punto de inflexión en el desarrollo de sistemas distribuidos escalables
 - Integran múltiples clientes (*peers*) distribuidos, de modo eficiente y escalable, para compartir datos bajo los conceptos de HTC
 - Constituyeron el paso inicial para crear redes computacionales globales como los sistemas *cloud* actuales

- Actualmente: era de computación en Internet (millones de usuarios cotidianamente). Los centros de supercomputación y datacenters deben proveer servicios de HPC y HTC para contemplar requisitos de un gran número de usuarios concurrentes
- Los centros de cómputo han evolucionado, utilizando metodologías de computación paralela y distribuida para incluir servicios de cómputo, almacenamiento y redes de comunicación de datos al servicio de los usuarios y las aplicaciones transaccionales que realizan
- Objetivo actual: aprovechar redes avanzadas para diseñar sistemas ubicuos que puedan utilizarse desde nuevos dispositivos y usando nuevas tecnologías

1.4



COMPUTACIÓN DISTRIBUIDA

COMPUTACIÓN DISTRIBUIDA

- Estudia el desarrollo de aplicaciones sobre **sistemas distribuidos**
- Un sistema distribuido consiste en la agrupación de **múltiples elementos de procesamiento autónomos**, cada uno conteniendo su propia memoria (por lo que la memoria global del sistema se encuentra distribuida).
- Los elementos de procesamiento están conectados a través de una **red de interconexión** y el intercambio de información (comunicaciones y sincronizaciones) se realiza mediante **pasaje de mensajes o invocaciones remotas** a procesos o servicios
- Los programas desarrollados bajo este paradigma se denominan programas distribuidos. El proceso de desarrollar e implementar programas distribuidos se denomina programación distribuida

COMPUTACIÓN DISTRIBUIDA

- Los sistemas distribuidos se popularizaron en la década de 1990. Comenzando con redes de workstations, clusters, y redes P2P, han avanzado hasta consolidar infraestructuras computacionales de amplia aplicabilidad y uniendo recursos de diversas ubicaciones geográficas
- Los sistemas globales fueron originalmente concebidos como *grids computacionales* o *grids de datos*
- El siguiente paso en la evolución de los sistemas distribuidos consistió en alcanzar la ubicuidad, independencia y transparencia al usuario al instrumentar infraestructuras *cloud*, principalmente enfocadas en el procesamiento de grandes volúmenes de información
- Sobre los sistemas cloud se implementa una abstracción del modelo de computación en redes de workstations, que se ha extendido y ha evolucionado para contemplar el desarrollo de la computación orientada a servicios y manejo de grandes volúmenes de datos en datacenters

- Computación en la nube (*cloud computing*)
 - Un cloud de recursos computacionales aplica estrategias de computación paralela o distribuida según el caso, o una mezcla de ambos paradigmas
 - Las infraestructuras cloud pueden ser construidas utilizando dispositivos físicos o máquinas virtuales, implementando grandes datacenters centralizados o distribuidos
 - Se considera como un tipo especial del modelo de computación utilitaria o computación basada en servicios, donde los recursos computacionales (procesamiento y almacenamiento) es suministrado a demanda
 - El desarrollo del paradigma de computación cloud ha estado impulsado por:
 - la creación del paradigma de Arquitectura orientada a Servicios (SOA) para diseñar y desarrollar aplicaciones distribuidas
 - el desarrollo de la Web 2.0, focalizada en los conceptos de compartir información, interoperabilidad, diseño centrado en el usuario y la colaboración
 - los avances en la tecnologías de virtualización

COMPUTACIÓN DISTRIBUIDA

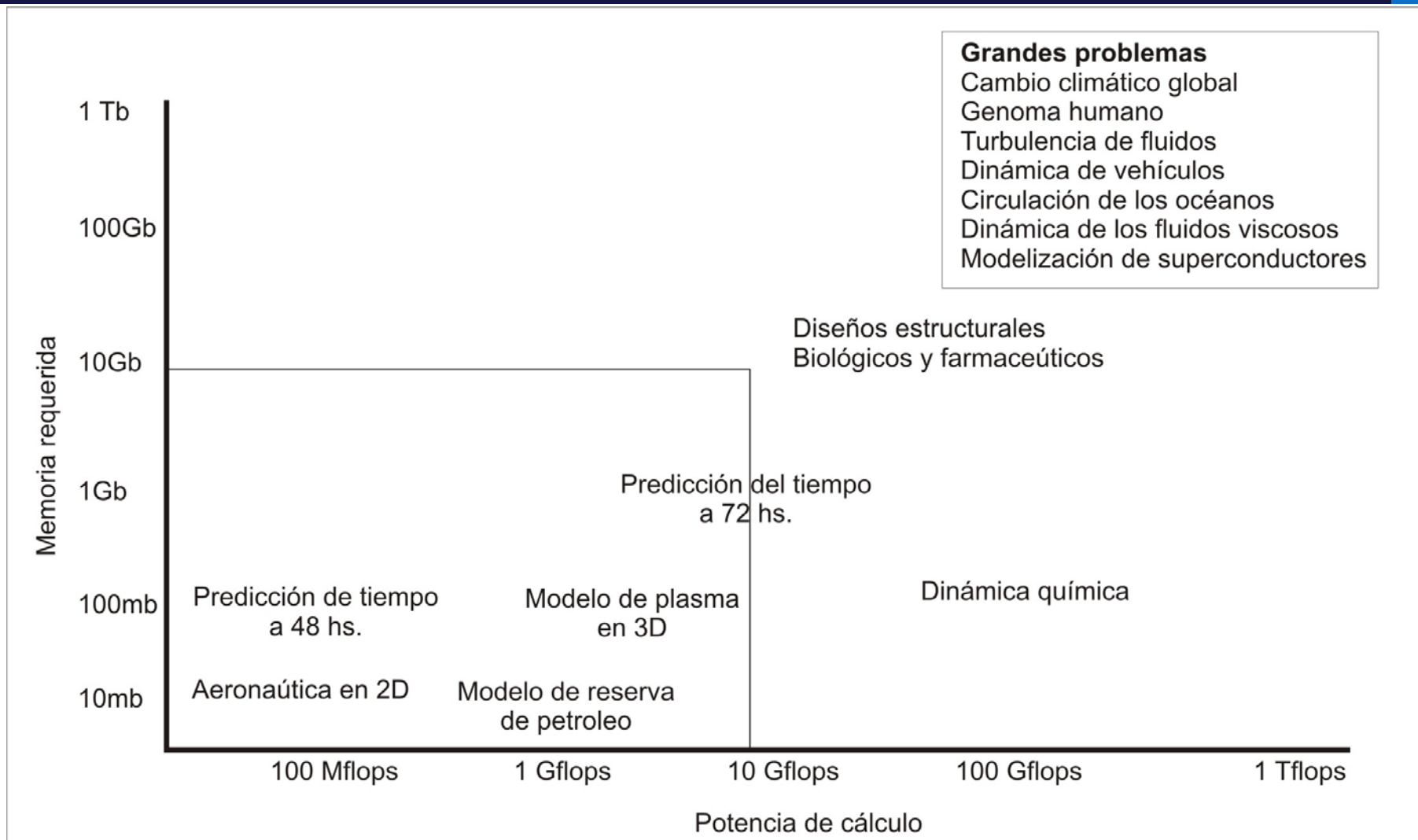
- Terminologías alternativas
 - *Computación concurrente*: unión de computación paralela y distribuida, ambos modelos basados en explotar la concurrencia para mejorar el rendimiento y aumentar la escalabilidad de las aplicaciones
 - *Computación ubicua o pervasiva*: modelos de computación distribuida que utilizan dispositivos en diversas ubicaciones geográficas simultáneamente, conectándose entre sí mediante redes cableadas o inalámbricas
 - *Internet de las cosas* (Internet of Things–IoT): interconexión digital de objetos cotidianos (computadores, sensores, dispositivos electrónicos, e inclusive humanos!). Utiliza Internet para implementar la computación ubicua con cualquier objeto conectado, en cualquier momento y en cualquier lugar. Se basa en los avances en la identificación por radiofrecuencia (radio-frequency identification–RFID), en los sistemas de posicionamiento satelital (Global Positioning System–GPS) y en el desarrollo de las redes de sensores.
 - *Computación en Internet* (Internet computing–IC): término genérico utilizado para designar todos los paradigmas de computación que hacen uso de Internet como mecanismo de interconexión

1.5



COMPUTACIÓN PARALELA Y DISTRIBUIDA: PRINCIPIOS

LOS PROBLEMAS TAMBIÉN CRECEN



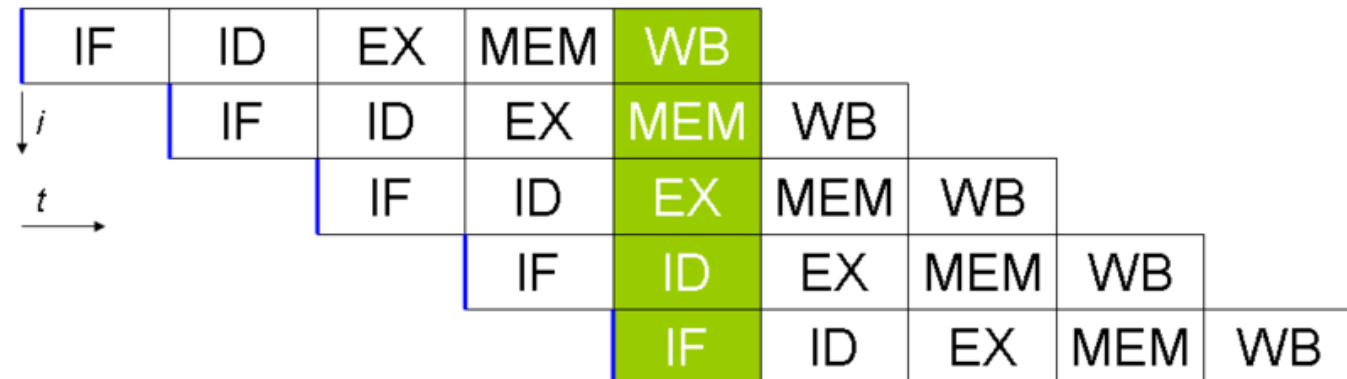
Requerimientos computacionales de problemas complejos

- Los sistemas de HPC y de HTC se enfocan en aspectos relevantes del **rendimiento y disponibilidad**, y deben ser capaces de satisfacer las crecientes demandas de poder de procesamiento
- Para lograr estos objetivos, deben contemplarse los siguientes principios de diseño:
 - La **eficiencia**, que en sistemas de HPC evalúa **tiempos de ejecución y uso de recursos** al explotar el paralelismo masivo, mientras que en sistemas de HTC se relaciona con el **número de transacciones, servicios y/o usuarios que se pueden atender por unidad de tiempo (throughput)**, el acceso eficiente a los datos, las estrategias de almacenamiento y la eficiencia energética
 - La **confiabilidad**, que evalúa la robustez y capacidad de auto-administración, desde bajo nivel (chip) hasta el más abstracto (aplicaciones). El objetivo es proveer sistemas de alto rendimiento que permitan asegurar altos niveles de calidad de servicio (QoS), aún bajo escenarios de alta demanda o situaciones de falla. Para lograrlo se aplican técnicas de **tolerancia a fallos**

- Para lograr estos objetivos, deben contemplarse los siguientes principios de diseño (continuación):
 - La *adaptación de los modelos de programación*, que determina la capacidad de soportar requerimientos de millones de tareas, potencialmente trabajando sobre repositorios masivos de datos y sistemas distribuidos virtualizados, utilizando diferente tipo de recursos físicos y modelos de servicio
 - La *flexibilidad en el desarrollo de aplicaciones*, que evalúa la capacidad de los sistemas de trabajar tanto en el modelo de alto desempeño (orientado a aplicaciones científicas, de ingeniería, industriales) como en el modelo de alto rendimiento (orientado a aplicaciones comerciales y procesamiento transaccional)

PROCESAMIENTO PARALELO

- En este contexto se ha desarrollado activamente el procesamiento paralelo
 - Basado en el estudio en Universidades e Institutos.
 - Aplicado directamente en la industria, organismos científicos y las empresas.
- La evolución de la aplicación del paralelismo puede resumirse en:
 - Paralelismo a nivel de bits (4, 8, 16 bits).
 - Se reduce a partir de 32 bits (hoy 64 bits).
 - Paralelismo a nivel de instrucciones.
 - Pipelining, superescalar, ejecución fuera de orden, ejecución especulativa, predicción de saltos.



PROCESAMIENTO PARALELO

- Evolución de la aplicación del paralelismo
 - Paralelismo a través de hilos (*multithreading*)
 - Programación paralela
 - Sobre supercomputadores
 - Sobre máquinas paralelas de bajo costo
- A partir de 1990, el desarrollo de las redes de computadoras permitió otro avance importante:
 - Procesamiento distribuido
 - Grid computing y cloud computing



PROCESAMIENTO PARALELO

- Ventajas:
 - Mayor capacidad de proceso
 - Permite ampliar objetivos y campo de trabajo
 - Permite abordar problemas de mayor complejidad
 - Permite mejorar calidad y fiabilidad de los resultados
 - Aumento directo de competitividad
 - Menor tiempo de proceso
 - Proporciona más tiempo para otras etapas de desarrollo del producto
 - Permite hacer frente a sistemas críticos
 - Reducción de costos
 - Aprovechar la escalabilidad de recursos **en el entorno local**

PROCESAMIENTO DISTRIBUIDO

- Principales conceptos
 - Procesadores independientes
 - Autonomía de procesamiento
 - Interconexión
 - Habitualmente mediante redes
 - Cooperación
 - Para lograr un objetivo global
 - Datos compartidos
 - Varios “repositorios” de datos
 - Sincronización
 - Frecuentemente a través del **pasaje explícito de mensajes**



PROCESAMIENTO DISTRIBUIDO

- Grados de distribución
 - Hardware y procesamiento
 - Datos o Estado
 - Control
- La distribución puede ser compleja de manejar, frecuentemente se necesitan herramientas especializadas:
 - Sistemas Operativos de Red
 - Sistemas Operativos Distribuidos
 - Bibliotecas de desarrollo

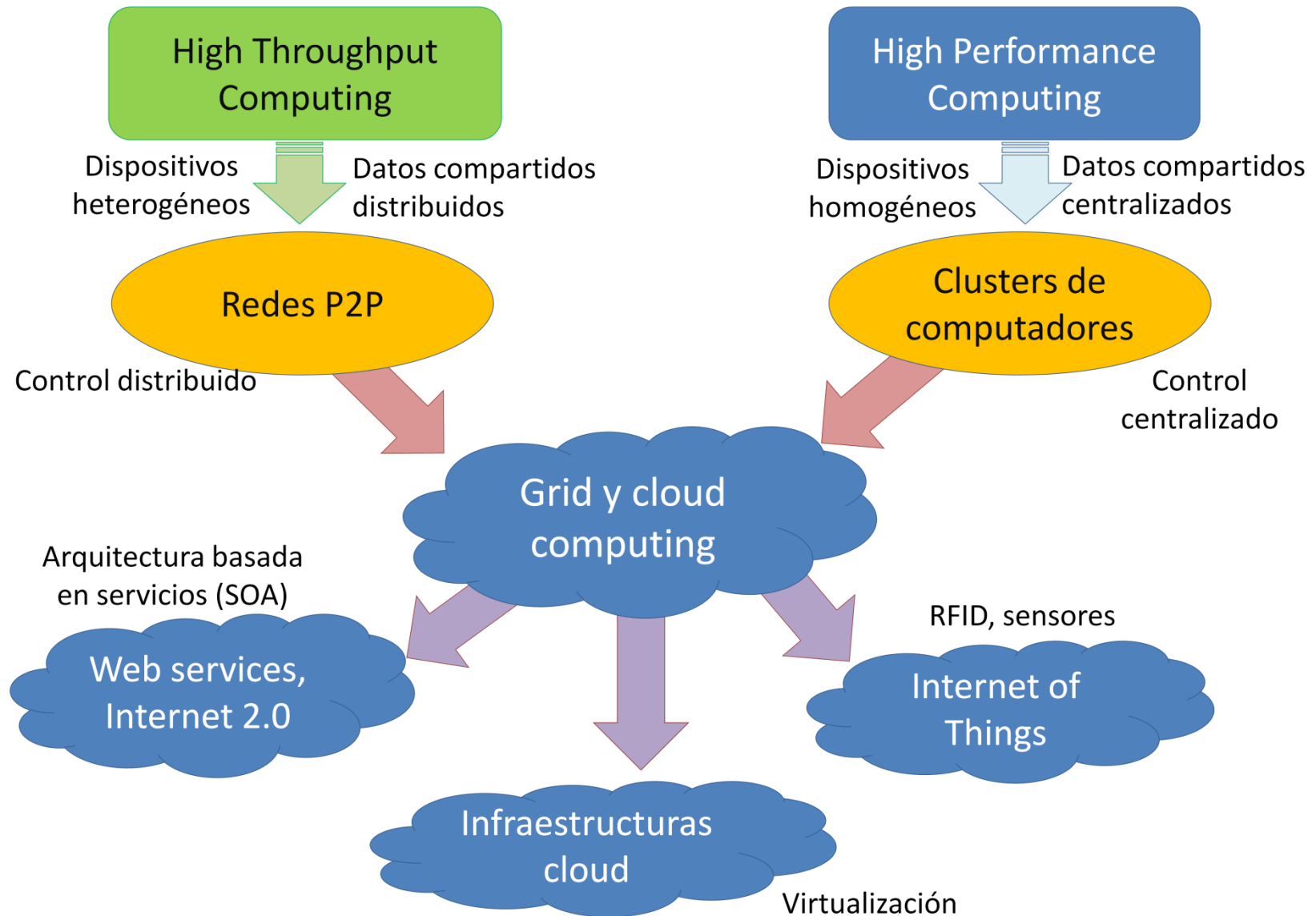


MPI

PROCESAMIENTO DISTRIBUIDO

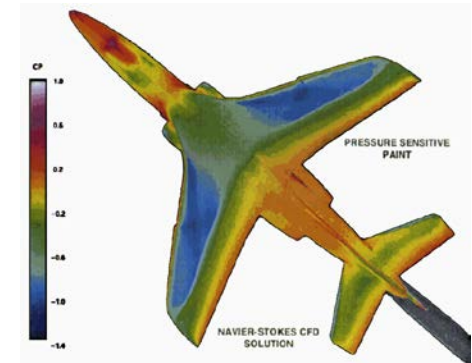
- Ventajas:
 - Mejora en desempeño:
 - Al disponer de mayor cantidad de procesadores
 - Robustez
 - Dada por la mayor disponibilidad de recursos
 - Seguridad no centralizada
 - Deben manejarse cuidadosamente las políticas
 - Una situación caótica evidentemente es una desventaja
 - Permite el acceso transparente a los datos no locales
 - Mecanismos y protocolos para compartir y acceder a la información
 - Escalabilidad
 - Potencialmente **ilimitada** en la red

EVOLUCIÓN TECNOLÓGICA



APLICACIONES

- Amplia aplicabilidad en problemas científicos
- Industriales
 - Química y bioingeniería
 - Estudio de estructuras moleculares, simulación de reacciones, espectroscopía
 - Fluidodinámica
 - Análisis de flujos, turbulencias y simulaciones
 - Mecánica Industrial
 - Diseño asistido. Modelos de elementos finitos
 - Medicina
 - Estudio del genoma, medicina farmacéutica, radioterapia
 - Electromagnetismo
 - Diseño de dispositivos de grabación, instrumentos médicos, tubos de rayos X, pantallas planas



APLICACIONES

- Comerciales
 - Telecomunicaciones
 - Análisis de tráfico, desempeño y calidad de servicio
 - Redes de control inteligentes
 - Comercio electrónico
 - Manejo transaccional en línea
 - Servicios web
 - Buscadores paralelos (metabuscadore)
 - Sistemas de tiempo real
 - Bases de datos paralelas
 - Análisis de datos
 - Data mining
 - Análisis de mercado, series temporales, etc



APLICACIONES

- Investigación

- Simulaciones espaciales
- Estudios atómicos
- SETI
- Inteligencia artificial



- Recreación

- Simulaciones tridimensionales y realidad virtual
- Cine: “actores virtuales”
- Multimedia: procesamiento de voz e imágenes
- Computación gráfica y videojuegos



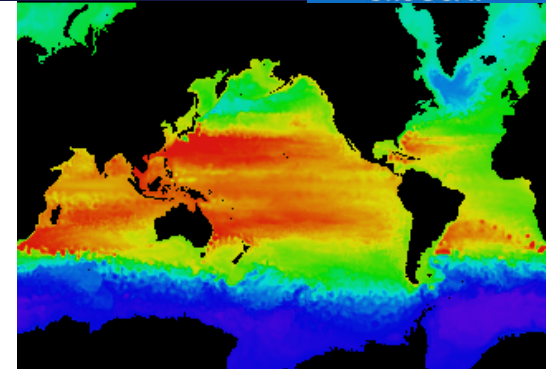
CASO DE ESTUDIO 1

PREDICCIÓN CLIMÁTICA



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

- Modelos climáticos globales
 - Dividir el mundo en una grilla (por ej., de 10 km de paso)
 - Resolver las ecuaciones de fluidodinámica para cada punto y tiempo
- Requiere un mínimo de 100 Flops por punto por minuto
- Predicción del tiempo (7 días, cada 24 horas): 56 GFLOPS
- Predicción climática (50 años, cada 30 días): 4.8 TFLOPS
- Perspectiva:
 - En un computador tradicional con procesador de 3GHz (≈ 10 GFLOPS) la predicción climática demandaría del orden de 100 años de tiempo de cómputo.



Es necesario disponer de estrategias más potentes para el análisis

CASO DE ESTUDIO 2

ANÁLISIS DE DATOS



- Hallar información “oculta” en grandes cantidades de datos
- ¿Qué motivos existen para “husmear” en grandes cantidades de datos?
 - ¿Existen dolencias inusuales en los habitantes de una ciudad?
 - ¿Qué clientes son más propensos a tratar de hacer fraude al seguro de salud?
 - ¿Cuándo conviene poner en oferta la cerveza?
 - ¿Qué tipo de publicidad enviar a un cliente?
- Recolección de datos:
 - Sensores remotos en un satélite
 - Telescopios
 - Microarrays generando data de genes
 - Simulaciones generando terabytes de datos
 - Espionaje

CASO DE ESTUDIO 2

ANÁLISIS DE DATOS



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

- La información se “descubre” mediante un proceso sistemático
- Análisis estadístico de los datos, comparaciones y relaciones para detectar tendencias, identificar situaciones o hechos inusuales
- El tiempo de procesamiento es creciente con respecto al volumen de datos
- Ciertos problemas pueden ser inabordables con los algoritmos de la computación secuencial tradicional

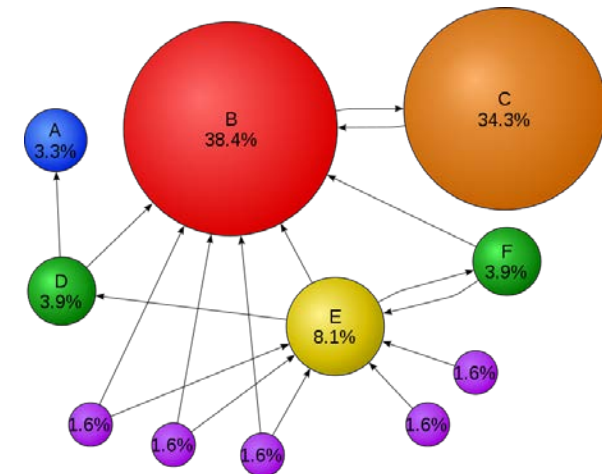
Es necesario disponer de métodos más potentes para el análisis

CASO DE ESTUDIO 3

ANÁLISIS DE DATOS: PAGERANK DE GOOGLE

- **PageRank**: familia de algoritmos utilizados para asignar numéricamente la relevancia de los documentos (o páginas web) indexados por un motor de búsqueda.
- Se basa en la “naturaleza democrática de la web”
 - Un enlace de una página A a una página B se interpreta como un voto, de la página A, para la página B.
- También se analiza la página que emite el voto
 - Los votos emitidos por las páginas “importantes” [con PageRank elevado], valen más, y ayudan a hacer a otras páginas "importantes".

$$PR(A) = (1 - d) + d * \sum_{i=1}^n \frac{PR(i)}{C(i)}$$



CASO DE ESTUDIO 3

ANÁLISIS DE DATOS: PAGERANK DE GOOGLE



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

$$PR(A) = (1 - d) + d * \sum_{i=1}^n \frac{PR(i)}{C(i)}$$

factor de amortiguación ≈ 0.85 ,
probabilidad de que un navegante
continúe pulsando links

valores de PageRank que tienen las
páginas que enlazan a A

número de enlaces salientes de la
página i (sean o no hacia A)

- El PageRank no se actualiza instantáneamente, ni siquiera diariamente, [tarda varios días en completarse]. Las últimas actualizaciones del Pagerank fueron en 4/2 y 5/12 de 2013 [oficial] y julio de 2014 [extraoficial].
- Datos:
 - 1.000 millones de páginas en 1999 (estimado)
 - 30.000 millones en 2005 (Yahoo)
 - 90.000 millones en 2007 (estimado Google)
 - Google dejó de reportar luego de indicar que indexaba 8.000 millones
 - +50.000 millones de páginas indexadas (estimado 2014)

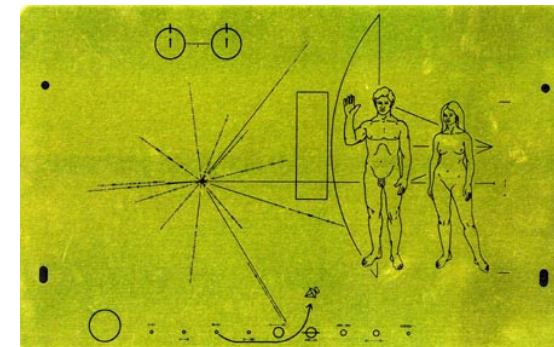
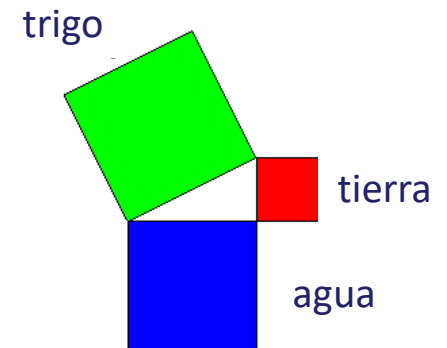
CASO DE ESTUDIO 4

SETI @HOME



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

- Giordano Bruno (1548-2000): “hay vida en otros mundos”.
Resultado: la hoguera !!
- Carl Gauss (1777-1855): “comunicación con la luna”.
Resultado: sin financiación.
- Joseph Von Litron (1840): “círculo de fuego”
Resultado: sin financiación.
- Charles Cros (1869): “espejo gigante”.
Resultado: sin financiación.
- Voyager (1977): placa de oro



CASO DE ESTUDIO 4

SETI @HOME



- Avances en SETI
 - Corrección del efecto Doppler coherente
 - Ancho de canal más fino, incrementa la sensibilidad
 - Resolución variable de ancho de banda y tiempo
 - Búsqueda de múltiples tipos de señales
 - Análisis de distribución Gaussiana
 - Búsqueda de pulsos repetidos
- **Problema: requiere TFLOPs de procesamiento**
- **Solución: computación paralela/distribuida**
- SETI@HOME: usa tiempo de cómputo donado voluntariamente por usuarios en todo el mundo para ayudar a analizar los datos recabados por los radiotelescopios

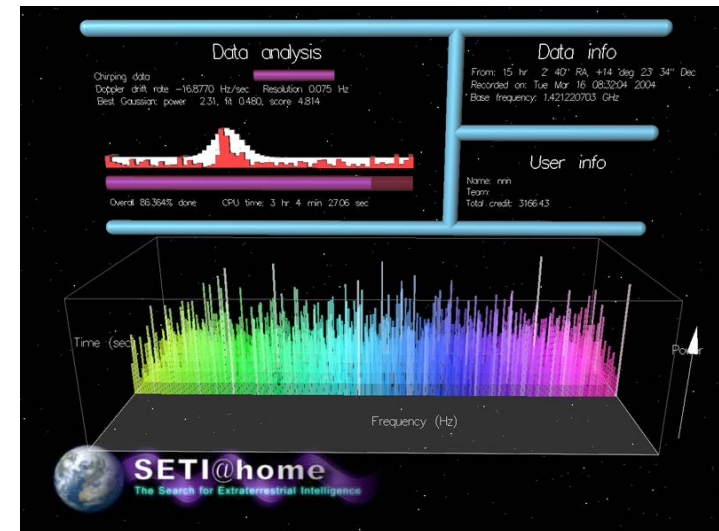
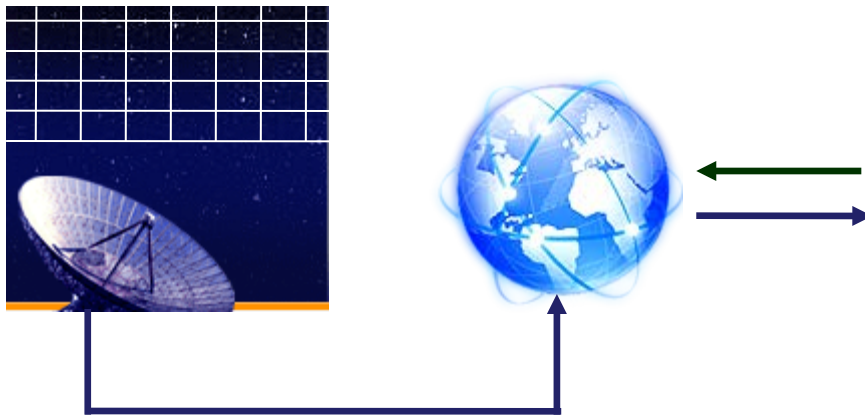
CASO DE ESTUDIO 4

SETI @HOME



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

- División del dominio de cómputo
- Distribución de datos
- Análisis distribuido
- Reporte de resultados



CASO DE ESTUDIO 4

SETI @HOME



- Estadísticas (2010)
 - > 5 millones de usuarios (mayor número para un proyecto de computación distribuida)
 - > 3 millones de computadores en 253 países
 - > medio millón de personas participan diariamente
 - En 2001, SETI@home sobrepasó el número de 10^{21} operaciones de punto flotante (el cómputo más largo de la historia según Guinness World Records).
 - En 2009, SETI@home tenía una capacidad de cómputo mayor a 800 TFLOPS
 - 1000 años de tiempo de cómputo por día
 - Más de dos millones de años de tiempo de cómputo agregado
 - Se procesan señales 10 veces más débiles que las de 1980-1990
- Ha sido el punto de partida para muchos proyectos similares
 - Folding@home, Einstein@home, MilkyWay@home, Rosetta@home, etc.

CASO DE ESTUDIO 5

INTERACCIÓN ENTRE PARTÍCULAS



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

- N-body problem
 - Predecir el movimiento de un conjunto de partículas y sus interacciones

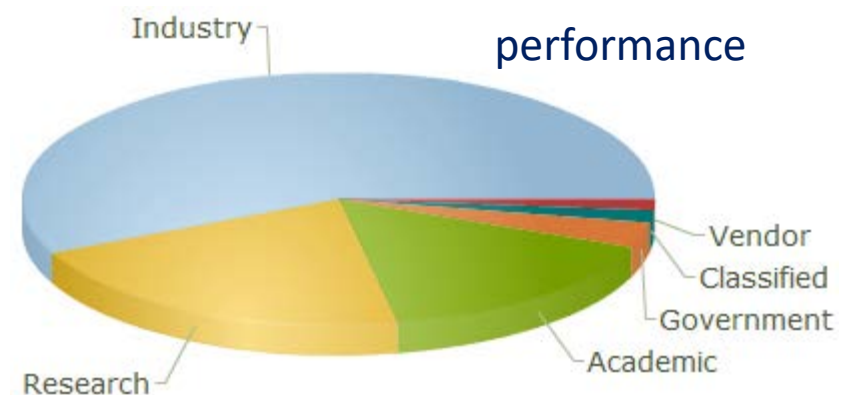
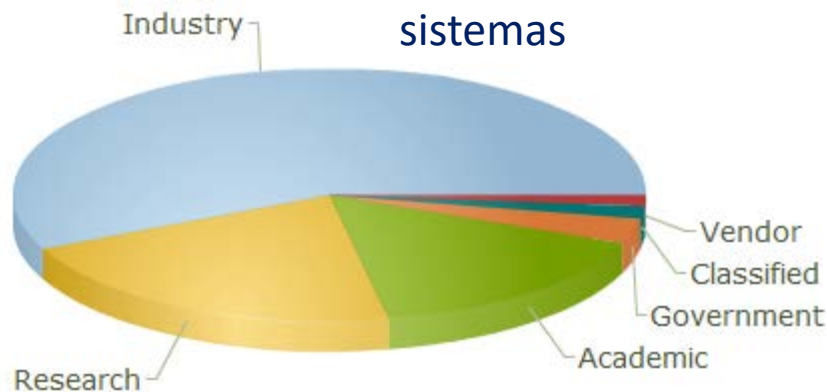
$$m_j \ddot{\mathbf{q}}_j = G \sum_{k \neq j} \frac{m_j m_k (\mathbf{q}_k - \mathbf{q}_j)}{|\mathbf{q}_k - \mathbf{q}_j|^3}, j = 1, \dots, n$$

- Aplicaciones:
 - Astronomía (movimiento de cuerpos celestes) y física (medios granulares)
 - Computación gráfica (iluminación)
 - Termodinámica (radiación térmica)
- Inconveniente: sistemas complejos involucran millones de partículas.
- Solución: paralelismo de procesamiento simétrico
 - Computación multithreading
 - Utilizando coprocesadores (XeonPhi) o procesadores gráficos (GPU)

APLICACIONES

- Sectores de aplicación (Top500)

Segments	Count	Share %	Rmax Sum (GF)	Rpeak Sum (GF)	Processor Sum
Academic	79	15.80 %	10258602	15254518	1205160
Classified	8	1.60 %	752813	974331	100464
Government	16	3.20 %	1060789	1686243	154460
Industry	285	57.00 %	15222240	25767492	2450854
Research	105	21.00 %	31113640	40809541	3813010
Vendor	7	1.40 %	521941	687823	55976
Totals	500	100%	58930025.59	85179949.00	7779924



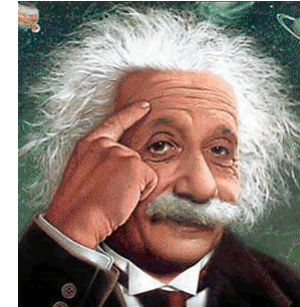
- Utilizar herramientas de desarrollo, simulación y optimización que utilicen paralelismo permite:
 - Reducir el tiempo necesario para desarrollar, analizar y optimizar diversas alternativas de diseño
 - Obtener resultados más precisos
 - Abordar casos realistas y escenarios extremos
 - Analizar alternativas de diseño que en otro caso resultarían intratables
- En definitiva, las técnicas de procesamiento paralelo/distribuido posibilitan obtener resultados más precisos de un modo eficiente en la resolución de instancias difíciles de problemas complejos

APLICACIONES



- RESUMEN

- Procesamiento paralelo de gran porte
 - Aplicaciones científicas
 - Manejo de enormes volúmenes de datos
- Procesamiento paralelo de mediano porte
 - Aplicaciones comerciales
 - Procesamiento transaccional financiero
 - Bases de datos distribuidas
- Programas multithreading
 - Aplicaciones de escritorio
- Procesamiento distribuido
 - Internet, grid y cloud, web services



CONSIDERACIONES IMPORTANTES

- **DISEÑO del HARDWARE**
 - Tecnología, poder y cantidad de los elementos de procesamiento
 - Conectividad entre elementos
- **TÉCNICAS de PROGRAMACIÓN**
 - Abstracciones y primitivas para cooperación
 - Mecanismos de comunicación

La clave es la integración de estos aspectos
para obtener un mejor desempeño computacional
en la resolución de aplicaciones

- Aspectos relevantes:
 - Arquitecturas de computadores paralelos
 - Modelos de programación
 - Diseño de algoritmos eficientes
 - Medidas para evaluar los algoritmos paralelos
 - Lenguajes y bibliotecas para programación paralela y distribuida
 - Tendencias tecnológicas actuales de diseño y programación

**SERÁN LOS ASPECTOS A ABORDAR
A LO LARGO DEL CURSO**