Computación distribuida y procesamiento de grandes volúmenes de datos

Sergio Nesmachnow
(sergion@fing.edu.uy)

UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Computación distribuida

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Objetivos

- Presentar los principales conceptos del paradigma de computación distribuida y de cloud computing.
- Introducir el modelo de computación MapReduce y el framework Hadoop para el desarrollo de aplicaciones distribuidas y de procesamiento de grandes volúmenes de datos.
- Presentar Apache Spark para el procesamiento eficiente de grandes volúmenes de datos.
- Presentar ejemplos y aplicaciones de los modelos y herramientas utilizadas para la resolución de problemas prácticos.

Computación distribuida

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Contenido

1. Computación distribuida y computación cloud
2. Procesamiento de grandes volúmenes de datos
3. El modelo de computación Map-Reduce
4. El framework Hadoop y su ecosistema
5. Almacenamiento: HDFS y HBase.
6. Aplicaciones de Map Reduce sobre Hadoop: conteo, índice invertido, filtros
7. Procesamiento de grandes volúmenes de datos con Apache Spark
8. Ejemplos de aplicaciones en Spark y el lenguaje Scala
9. Análisis de datos utilizando Spark y el lenguaje R
10. Aplicaciones iterativas: Google Pregel y Apache Giraph

Computación distribuida

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Detalles

- Público objetivo: profesionales y estudiantes interesados en computación distribuida y procesamiento de grandes volúmenes de datos.
- Conocimientos previos exigidos: Fundamentos de informática.
- Conocimientos previos recomendados: Fundamentos de programación.
- Metodología de enseñanza:
  - Exposiciones teórico-prácticas para presentar los principales conceptos sobre computación distribuida y las metodologías de análisis y procesamiento de grandes volúmenes de datos.
  - Forma de evaluación: participación oral y actividades prácticas en clases (20%); trabajo final que aplique las técnicas de computación distribuida presentadas en el curso para resolver un problema específico de procesamiento de datos (80%).

Computación distribuida

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Detalles

- Horas de clase: 20 teórico, 5 práctico, 5 laboratorio, 5 consulta.
- Subtotal horas presenciales: 35.
- Horas estudio: 25, horas de ejercicios: 10, horas trabajo final: 20
- Total de horas de dedicación del estudiante: **90 horas**
- Bibliografía:
  - Kai Hwang, Jack Dongarra y Geoffrey C. Fox. 2011. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
  - George Reese. 2009. Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. O'Reilly Media, Inc.
  - Jeffrey Aven. 2016. Apache Spark in 24 Hours. Pearson Education, USA.
  - Holden Karau, Andy Konwinski y Patrick Wendell. 2015. Learning Spark: Lightning-Fast Big Data Analysis. O'Reilly Media, Inc.

Computación distribuida

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Sitio del curso en el Espacio Virtual de Aprendizaje (EVA) de la Facultad de Ingeniería

https://eva.fing.edu.uy/course/view.php?id=1154

Accesible desde eva.fing.edu.uy

Matriculación libre

Computación distribuida

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Contenido

1. Computación distribuida y computación cloud
2. Procesamiento de grandes volúmenes de datos
3. El modelo de computación Map-Reduce
4. El framework Hadoop y su ecosistema
5. Almacenamiento: HDFS y HBase.
6. Aplicaciones de Map Reduce sobre Hadoop: conteo, índice invertido, filtros
7. Procesamiento de datos en tiempo real: Apache Spark
8. Ejemplos de aplicaciones en Spark y el lenguaje Scala
9. Análisis de datos utilizando Spark y el lenguaje R
10. Aplicaciones iterativas: Google Pregel y Apache Giraph

Paradigmas de computación paralela y distribuida

PARADIGMAS de COMPUTACIÓN

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- 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 computación centralizada y computación distribuida.
- El campo de la computación paralela se superpone con el de computación centralizada y con el de la 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.
- 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

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- 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 disponibles desde terminales alternativas).
  - Son sistemas sobredimensionados para tolerar picos de utilización: usuarios tienen más prestaciones de 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 exceder la capacidad de procesamiento multiusuario) no es sencillo escalar las capacidades de cómputo y/o de almacenamiento
  - Costo elevado (millones USD).

COMPUTACIÓN CENTRALIZADA: HISTORIA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Dominó el mundo de la computación hasta la aparición de las computadoras personales en la década de 1980.
- Estuvo descartada como modelo de computación durante 15 años.
- Se comenzó a aplicar nuevamente para manejo transaccional de comercio electrónico y a partir del 2000 el desarrollo de Linux permitió implementar soluciones basadas en la utilización de (cientos de) máquinas virtuales en un único centro de cómputo … y las infraestructuras de computación centralizada volvieron a la vida.
- Alternativa para el acceso eficiente a recursos de cómputo mediante *clientes livianos* (no requieren instalar software pesado en el usuario).
- La evolución tecnológica ha seguido oscilando: los desarrolladores incluyen más lógica en los clientes para aprovechar el poder de cómputo de los dispositivos, implementando clientes ricos que reducen el cómputo remoto en los servidores.

COMPUTACIÓN CENTRALIZADA: HISTORIA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Aún es utilizada para aplicaciones críticas (financieras, 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
- Se utiliza también en datacenters que emplean modelos híbridos:
  - ciertas aplicaciones (e.g., aplicaciones web) ejecutan distribuidas, accediendo a servicios proporcionados por otras aplicaciones (en general, sistemas informáticos más complejos) que ejecutan de forma centralizada en un 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

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- 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

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Aplica un modelo de procesos concurrentes en ejecución simultánea, sobre una infraestructura computacional que en general es altamente acoplada y 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 DISTRIBUIDA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Aplica un modelo de procesos concurrentes y distribuidos en ejecución simultánea, sobre una infraestructura computacional débilemente acoplada que no 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 exclusivamente mecanismos de memoria distribuida (utilizando pasaje de mensajes explícitos).
- Un sistema computacional capaz de proveer el soporte para computación distribuida se denomina computador distribuido.

COMPUTACIÓN PARALELA y DISTRIBUIDA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- 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/distribuido
  - 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.

COMPUTACIÓN PARALELA y DISTRIBUIDA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- 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.

COMPUTACIÓN PARALELA y DISTRIBUIDA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Computador paralelo o distribuido
  - Conjunto de procesadores capaces de trabajar cooperativamente en la resolución de problemas computacionales.
  - La definición incluye un amplio espectro: supercomputadoras, procesadores masivamente paralelos, clusters, sistemas grid, cloud, etc.
  - Característica fundamental: disponibilidad de múltiples recursos de cómputo.
- Computación paralela y computación distribuida
  - Han dejado de ser exóticas para ser ubicuas.
  - Posibilitadas por avances en diferentes tecnologías:
    - Poder de procesamiento (microprocesadores).
    - Redes (comunicación de datos).
    - Desarrollo de bibliotecas e interfaces para programación.

COMPUTACIÓN PARALELA y DISTRIBUIDA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- 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).

COMPUTACIÓN de ALTO DESEMPEÑO

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Sistemas de computación de alto desempeño/alta performance (HPC)

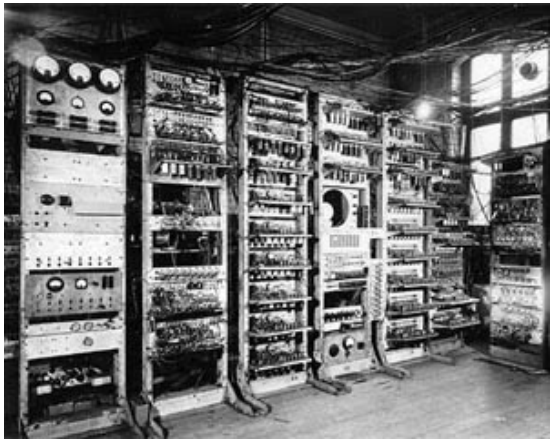
- Utilizados para computación científica.
- Enfatizan la importancia de la eficiencia (performance), considerando el número de operaciones realizadas por unidad de tiempo.
- Propulsados por las mejoras e innovaciones tecnológicas, han incrementado sus velocidades de procesamiento:
  - 1990: GFLOPS (10^9 operaciones de punto flotante por segundo)
  - 2010: TFLOPS (10^{12})
  - 2015: PFLOPS (10^{15})
  - antes de 2020: EFLOPS (10^{18})

- El número de usuarios de sistemas HPC es bajo (10%), pero con gran uso 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.
- Se enfocan en proporcionar servicios a la mayor cantidad de usuarios simultáneamente.
- En la actualidad, el desarrollo de sistemas de computación orientados a aplicaciones de mercado está más asociado con el paradigma de HTC que con el de HPC.

EVOLUCIÓN TECNOLÓGICA



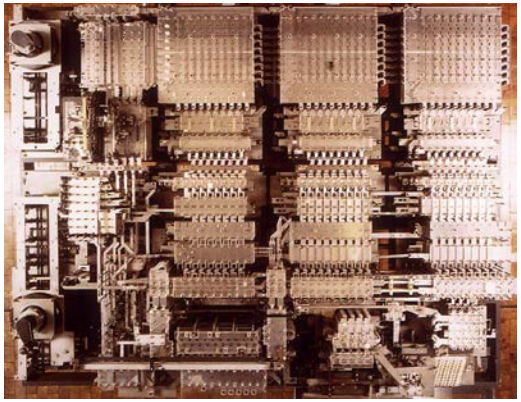
Collosus 2 (UK), primer computador paralelo: 50.000 op/s

MEGAFLOP COMPUTER

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

1938 — 1946 — 1948 — 1954 — 1964

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



ENIAC (USA), 5.000 op/s

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

EVOLUCIÓN TECNOLÓGICA

Cray-2/8 (LANL, USA),
3.9 GFLOPS

GIGAFLOP
COMPUTER

1984

1985-89

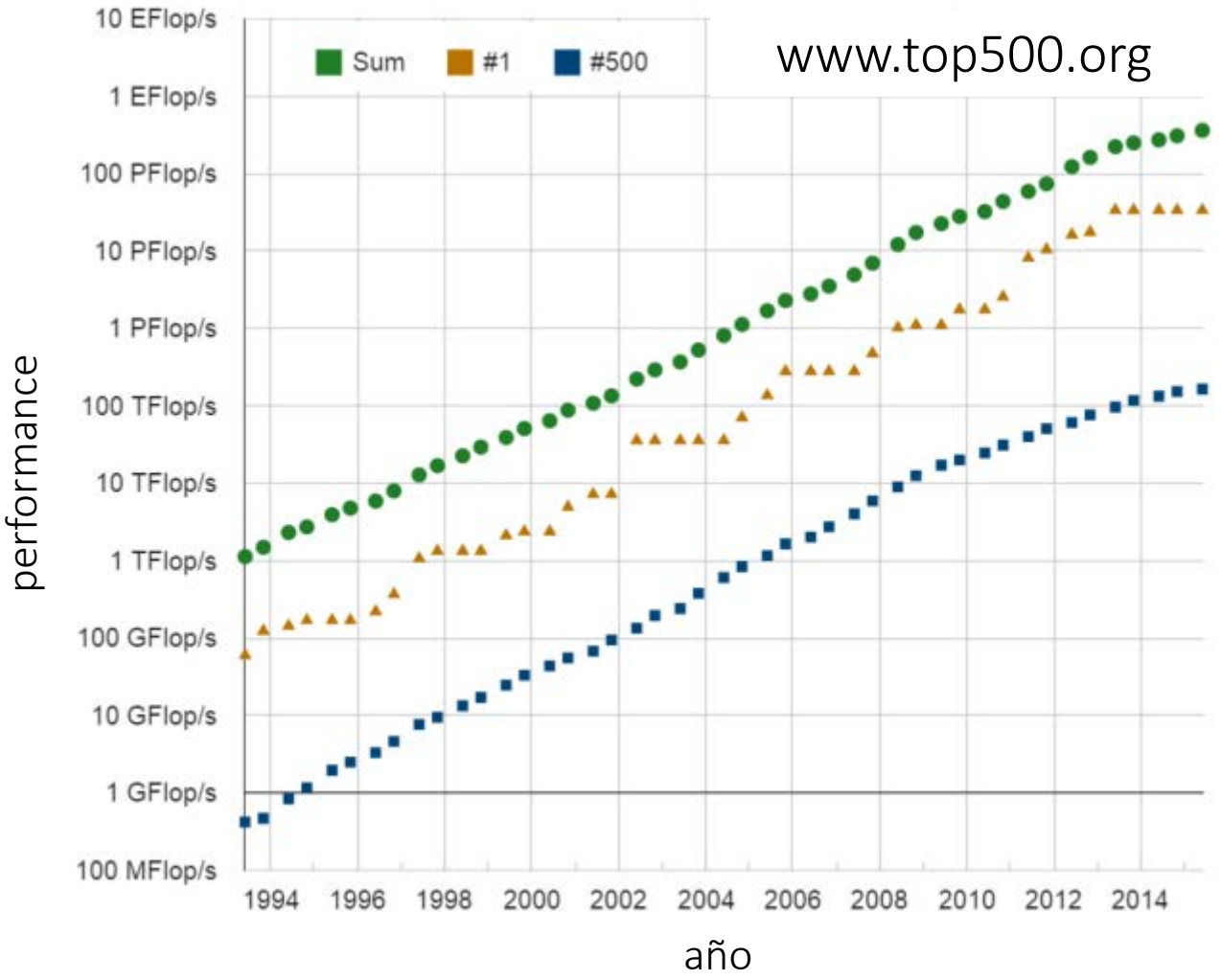
1997

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, escala logarítmica!, crecimiento exponencial)

EVOLUCIÓN TECNOLÓGICA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

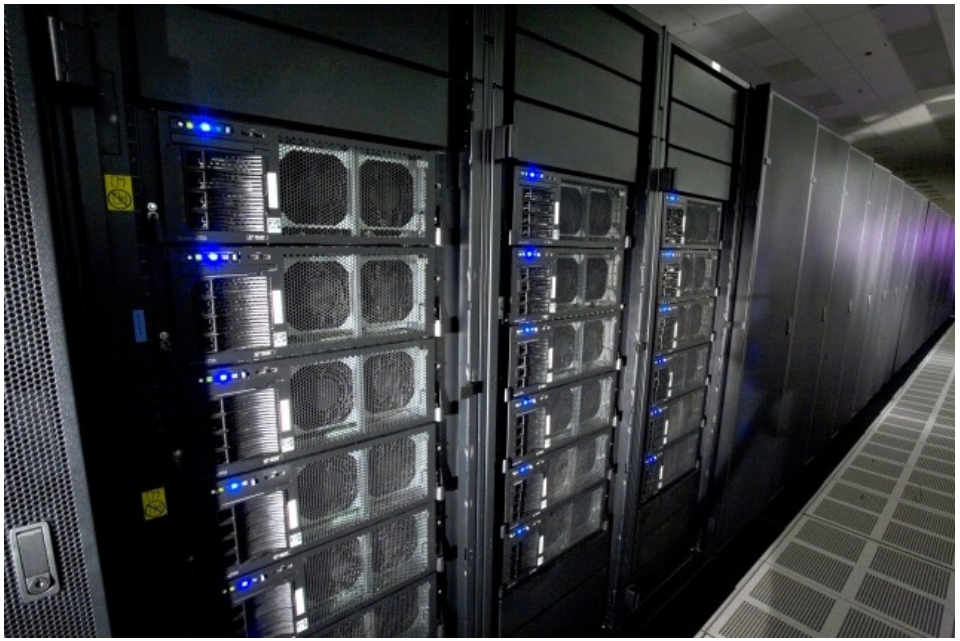
- Similar comportamiento para otros indicadores:
  - Frecuencia de relojes.
  - Densidad de circuitos en chips de procesadores.
  - Capacidad de almacenamiento secundario.
  - Capacidad de transmisión por bus/red.

- Siguen el mismo comportamiento exponencial, con diferentes pendientes.

EVOLUCIÓN TECNOLÓGICA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Junio de 2008: Petaflop supercomputer (Peta = 10^{15} = 1000000000000000).
  - 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 m^2).
    - 10,000 conexiones (Infiniband y Gigabit Ethernet), 90 km de fibra óptica.



EVOLUCIÓN TECNOLÓGICA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- 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

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- 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

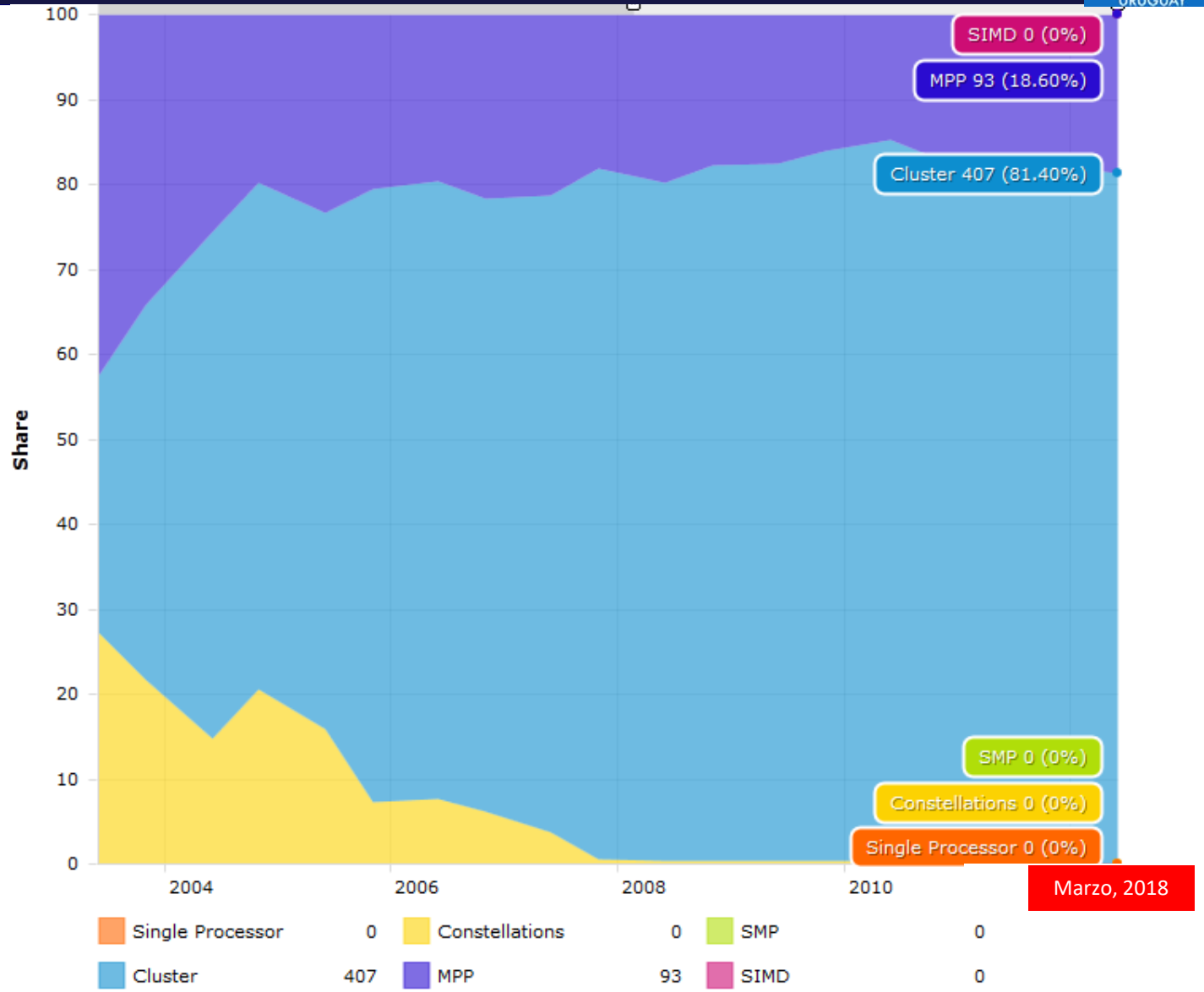
- 40.960 nodos, procesadores SW26010 1.45GHz manycore (256 núcleos de cómputo, 4 de administración), arquitectura ShenWei (RISC de 64-bits).
  - 10.649.600 núcleos, cada núcleo tiene 64 KB de memoria scratchpad (NUMA) para datos y 12 KB para instrucciones.
  - Los núcleos se comunican por red en chip, 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 µs.
  - Comunicaciones MPI a 12 GB/s (similar a InfiniBand EDR o 100G Ethernet).
  - Sistema operativo Raise OS 2.0.5, basado en Linux.
  - Incluye una 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 cientificas 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 el Top500 que USA.
  - China: 167, USA: 165.

EVOLUCIÓN TECNOLÓGICA

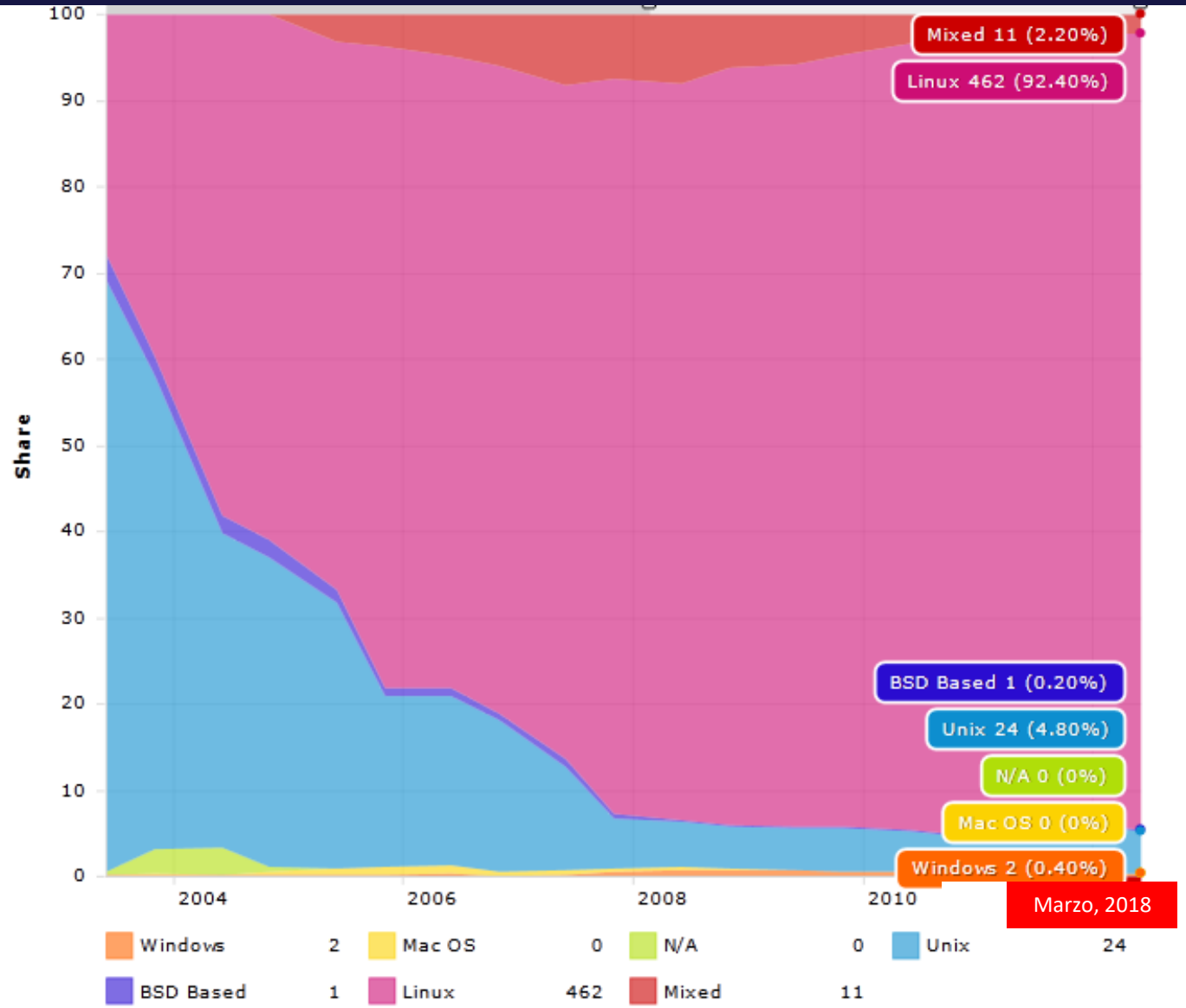
ARQUITECTURAS



EVOLUCIÓN TECNOLÓGICA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

SISTEMAS OPERATIVOS



INFRAESTRUCTURA

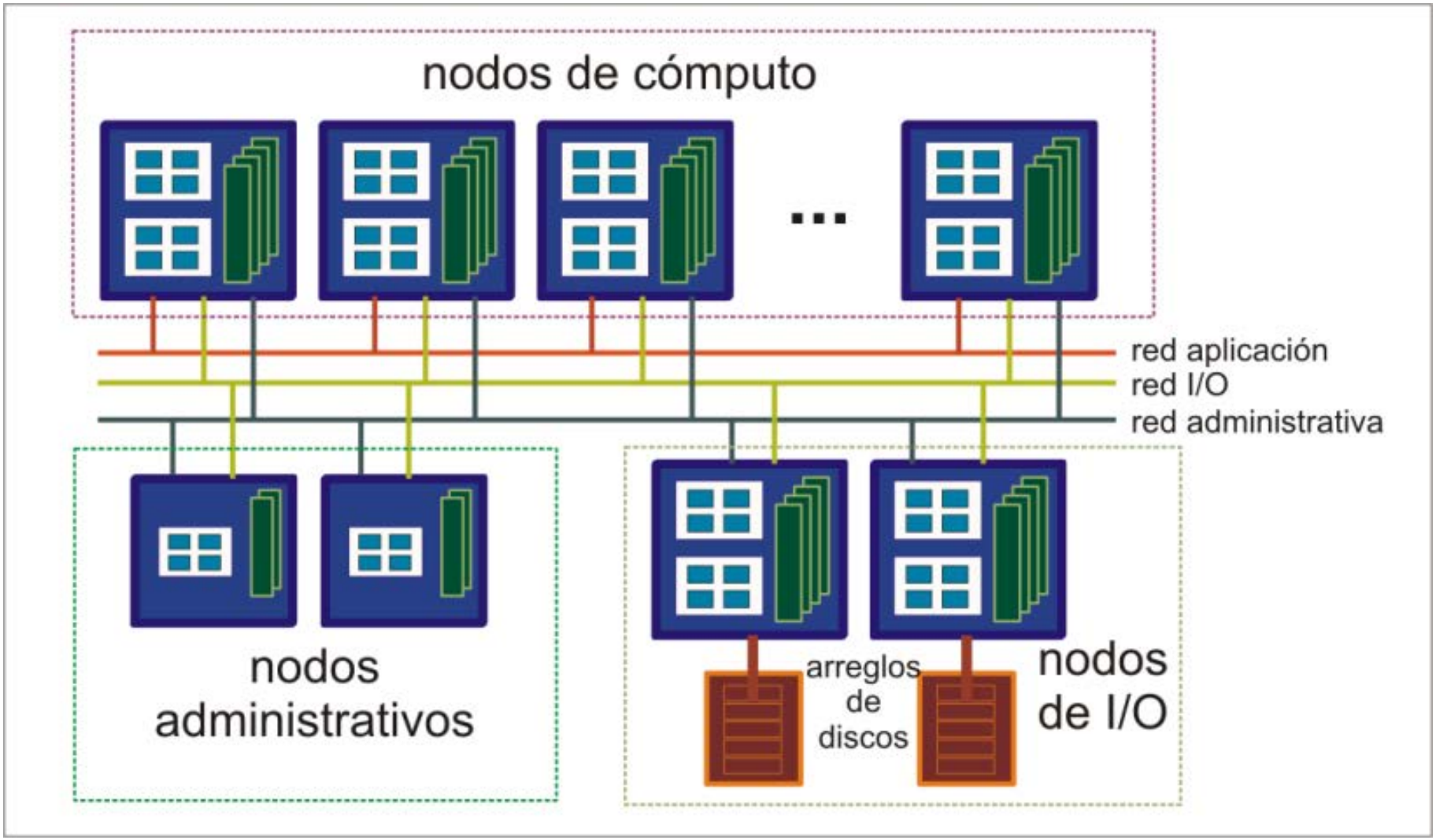
- La tecnología ha avanzado, permitiendo disponer de máquinas paralelas de bajo costo.
  - Clusters construidos por agregación de componentes de bajo costo.
- Internet surge como una fuente potencial de recursos de computación ilimitados.
  - Internet 2 amplía la banda y la potencia de comunicación entre equipos.
- Se han desarrollado las tecnologías grid y 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

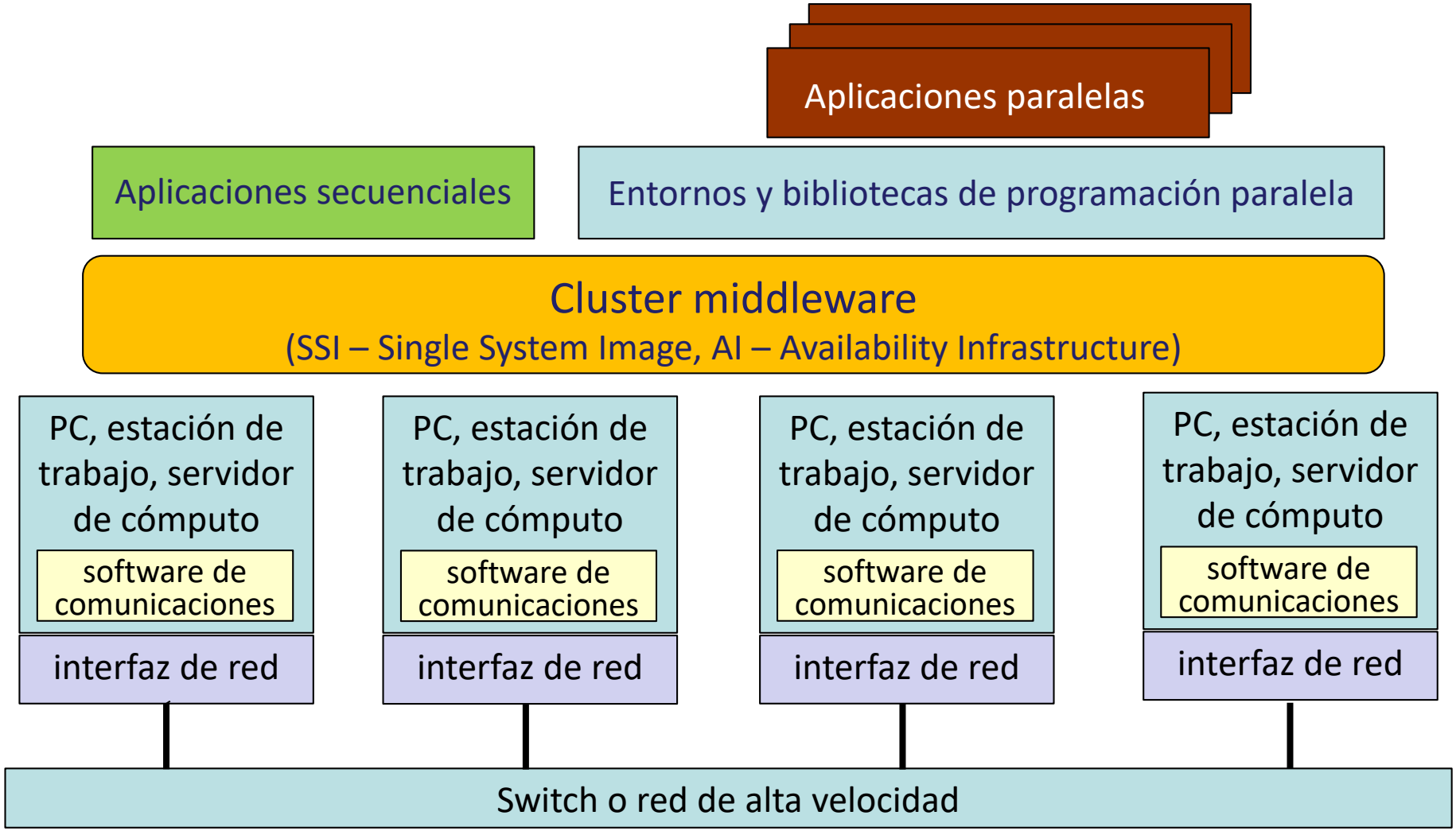
- Las alternativas mencionadas constituyen opciones realistas para tratar de lograr capacidad de cómputo competitivo.
  - Obviamente, sin llegar a los límites de los mejores supercomputadores del Top500

- Sin embargo, permiten resolver problemas interesantes en los entornos académicos, industriales y empresariales, con una infraestructura de bajo costo.

CLUSTERS


nodos de cómputo
...
red aplicación
red I/O
red administrativa
nodos
administrativos
arreglos
de
discos
nodos
de I/O

ARQUITECTURA DE UN CLUSTER



CLUSTER MIDDLEWARE

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Reside entre el SO y las aplicaciones, ofrece infraestructura para soportar:
  - Single System Image (SSI).
  - System Availability (SA).

- SSI permite que un cluster pueda ser visto como un único equipo ("globaliza" los recursos de un sistema).
  - Accceso a través de ssh cluster.myinstitute.edu.
  - Monitoreo centralizado, sistema de archivos global.

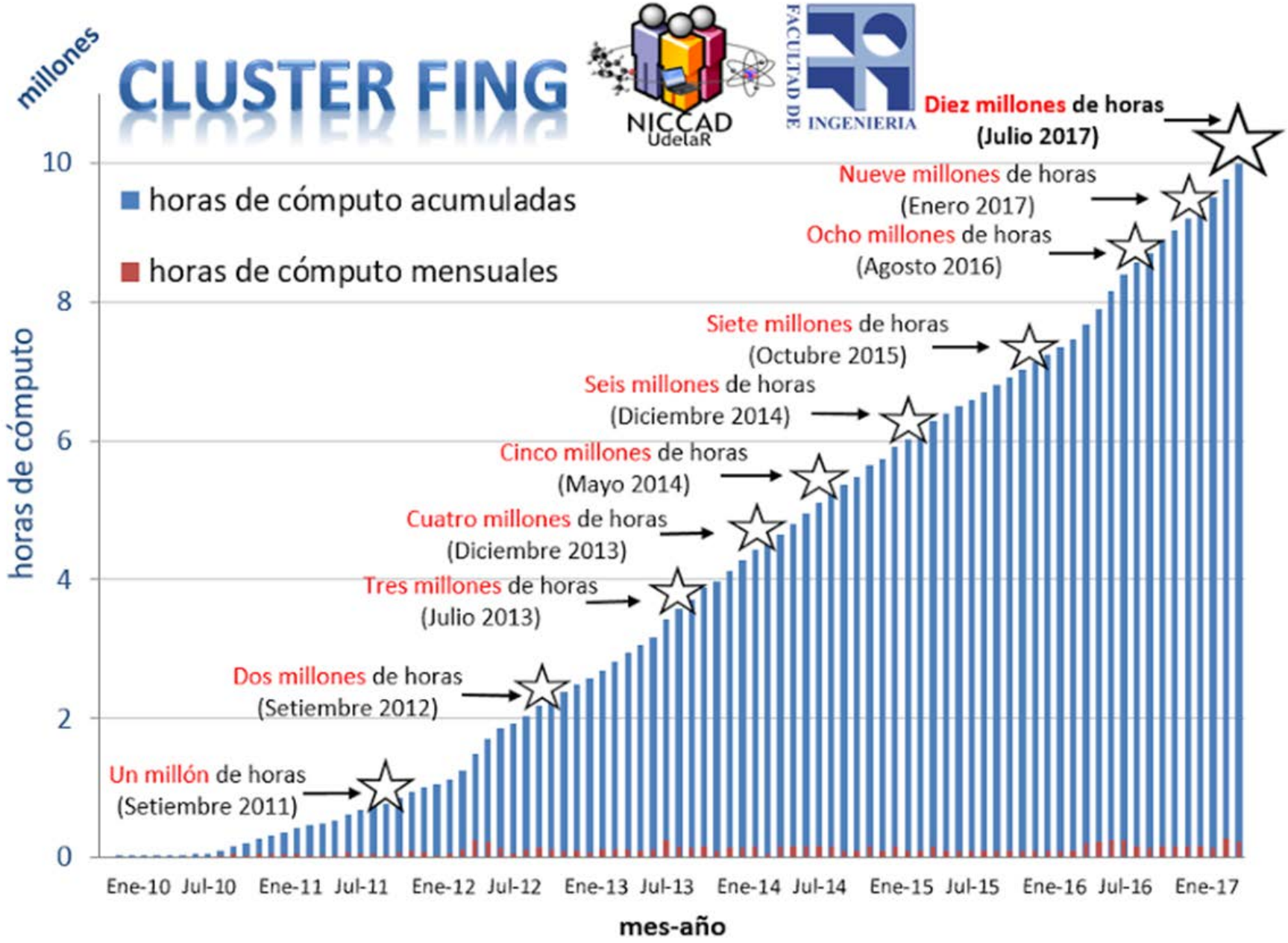
- SA provee disponibilidad.
  - Check pointing, migración de procesos, replicación de datos, etc.

EL CLUSTER FING

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Infraestructura para computación científica de alto desempeño, UdelaR.
- Operacional desde marzo de 2009:
  - Autofinanciada y autogestionada.
  - Más de 350 usuarios de 12 países.
- 1740 cores (540 de CPU, 960 de GPU, 240 Xeon Phi)
  - >1 TB de memoria RAM, >250 TB RAID storage, 34 kVA batería
  - Pico de performance: 6000 GFLOPS (6×10^{12} operaciones de punto flotante por segundo), *el mayor poder de cómputo disponible en el país*

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



2009–2017: 10 millones de horas de cómputo utilizadas y 14 millones de horas aportadas

EL CENTRO NACIONAL DE SUPERCOMPUTACIÓN (CLUSTER-UY)

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Proyecto de grandes equipos (ANII-UdelaR, 2017–2018)
  - Autofinanciado y autogestionado.
- 30 nodos interconectados: más de 1500 núcleos de cómputo de CPU y 125.000 núcleos de cómputo de GPU de última generación.
- Capacidad de procesamiento: 120 TeraFlops (120×10^{12} operaciones por segundo)
- >20x Cluster FING.

El mayor poder de cómputo disponible en el país

http://cluster.uy/

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

avanzamos juntos

SITEMAS DISTRIBUIDOS

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- 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 gran aplicabilidad que unen 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 computadores, 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.

CLOUD COMPUTING

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Un cloud de recursos computacionales combina los paradigmas de computación paralela y de computación distribuida.
- 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 basada en servicios, donde los recursos computacionales (procesamiento y almacenamiento) es suministrado a demanda.
- El desarrollo del paradigma de cloud computing fue impulsado por:
  - los avances en la tecnologías de virtualización.
  - 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.

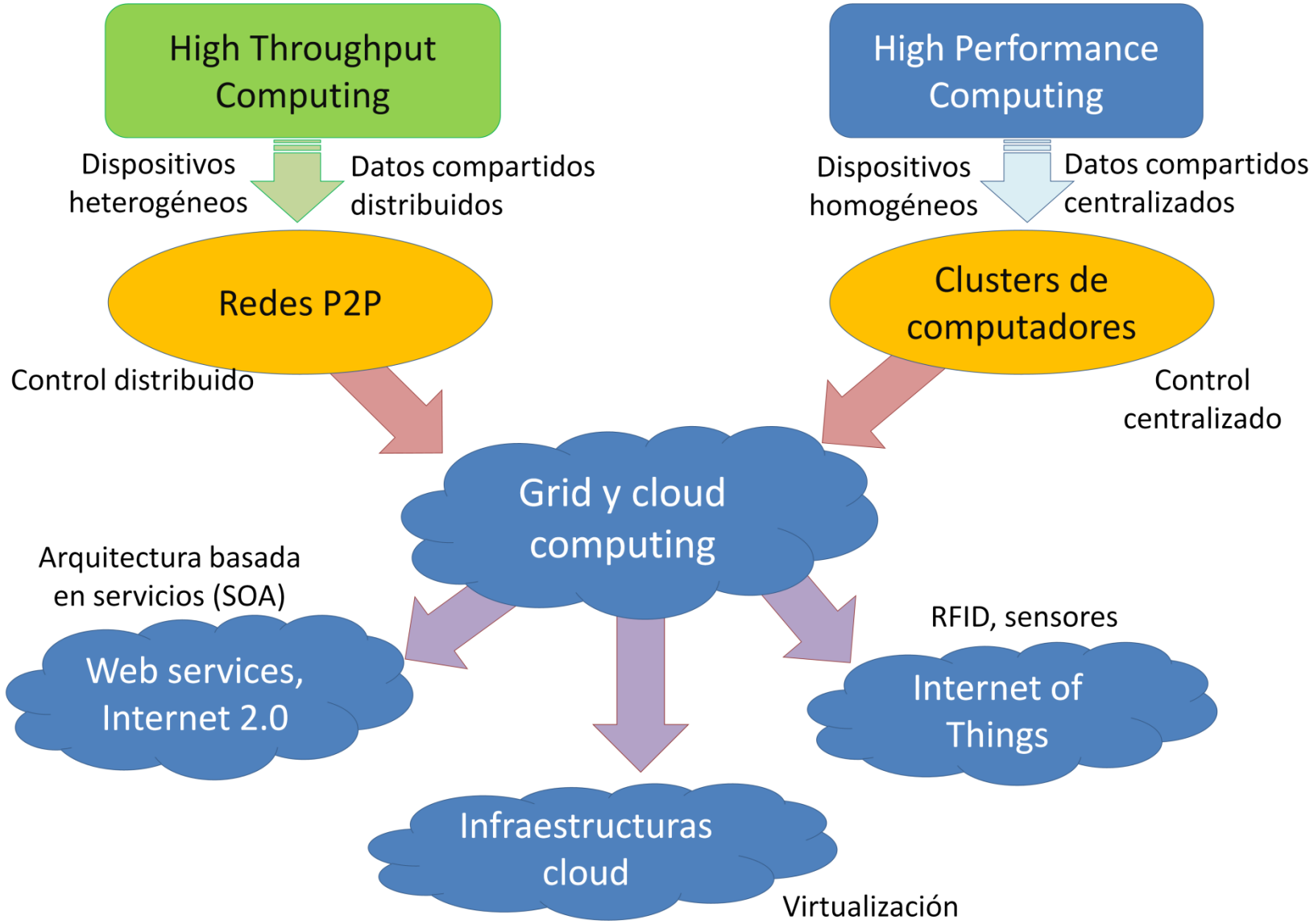
COMPUTACIÓN PARALELA y DISTRIBUIDA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

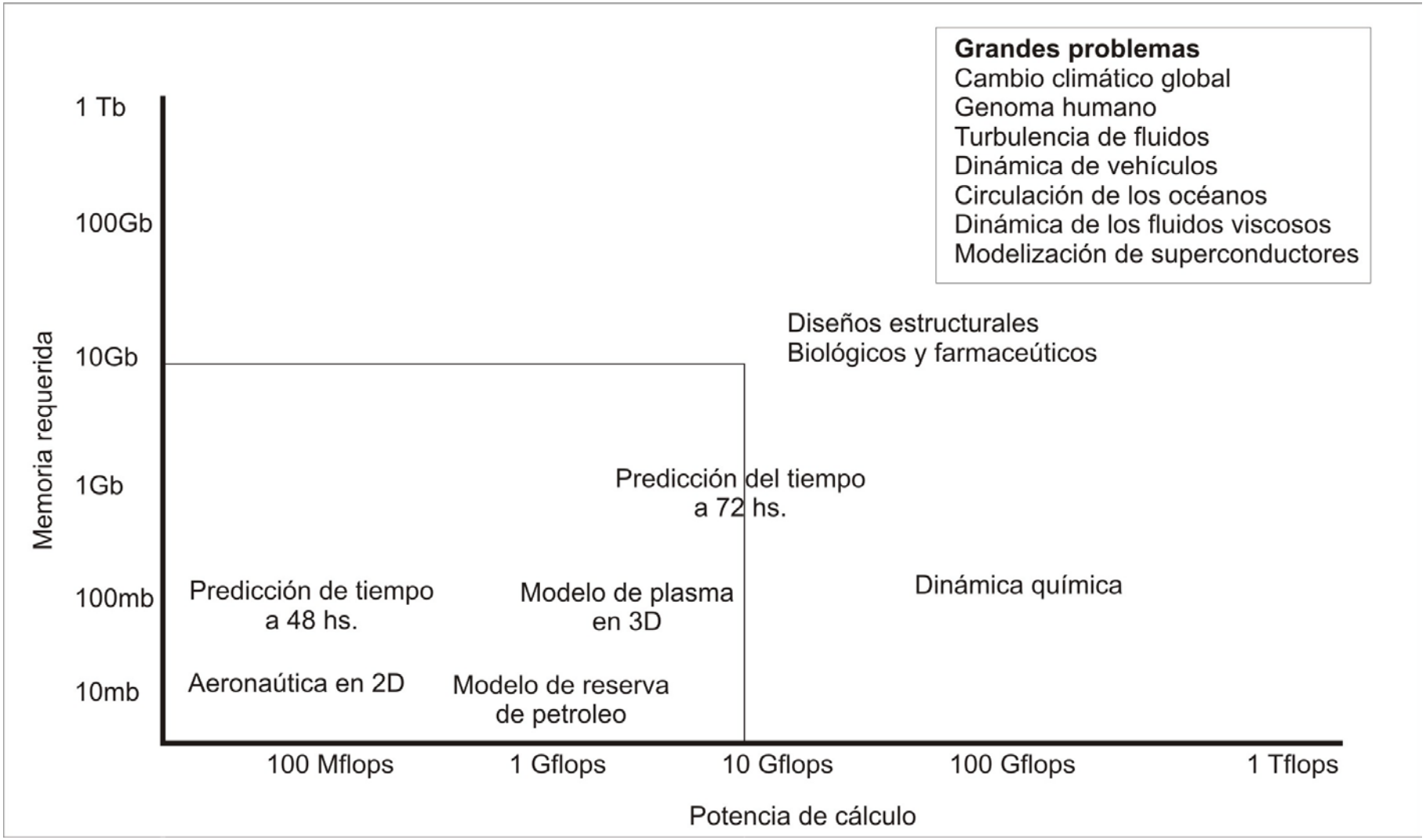
- Los sistemas de HPC se enfocan en aspectos relevantes de rendimiento y disponibilidad, y deben ser capaces de satisfacer as crecientes demandas de poder de procesamiento.
- Para lograr estos objetivos, deben contemplarse los siguientes principios de diseño:
  - La eficiencia, que en sistemas de HPC 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 y su almacenamiento.
  - 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 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).

EVOLUCIÓN TECNOLÓGICA



LOS PROBLEMAS TAMBIÉN CRECEN

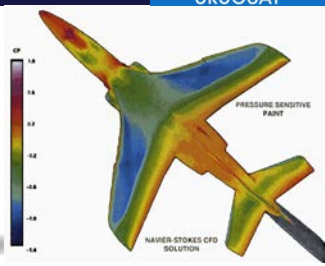


Requerimientos computacionales de problemas complejos

APLICACIONES

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Amplia aplicabilidad en problemas científicos e industriales
  - Química y bioingeniería, fluidodinámica, mecánica industrial, medicina, electromagnetismo, etc.



- Gran utilidad en aplicaciones comerciales
  - Telecomunicaciones, comercio electrónico, servicios web, sistemas de tiempo real, bases de datos paralelas, análisis de datos masivos, etc.

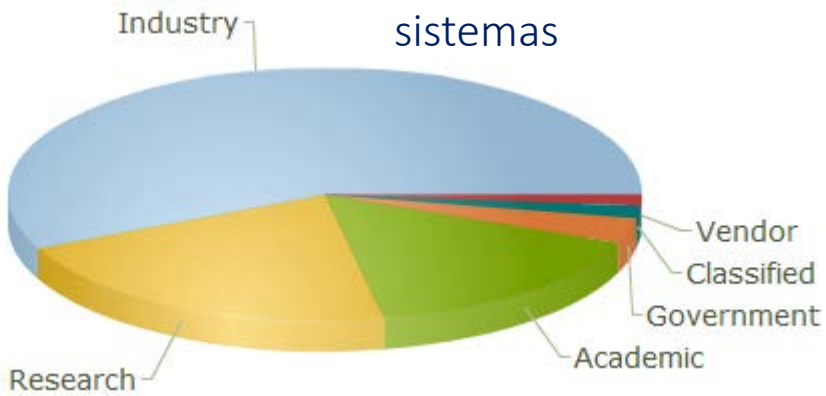
- Recreación
  - Simulaciones tridimensionales y realidad virtual, actores virtuales, procesamiento multimedia (voz e imágenes), computación gráfica y videojuegos

APLICACIONES

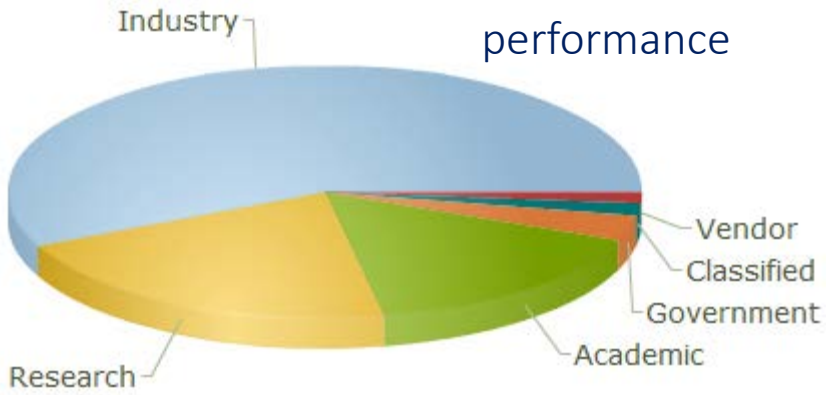
- Sectores de aplicación (Top500)

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**

sistemas



performance



APLICACIONES

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Utilizar herramientas de desarrollo, simulación, procesamiento de datos y optimización que utilicen computación paralela y distribuida permite:
  - Reducir el tiempo necesario para desarrollar, analizar y optimizar diversas alternativas de diseño.
  - Obtener información valiosa para mejorar la calidad de los servicios.
  - 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 computación de alto desempeño posibilitan obtener resultados más precisos y mejorar la calidad de la información de un modo eficiente en la resolución de instancias difíciles de problemas complejos.

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

MODELOS de COMPUTACIÓN PARALELA/DISTRIBUIDA

MODELO DE PROGRAMACIÓN PARALELA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Programación en máquina de Von Neumann
  - Secuencia de operaciones (aritméticas, read/write de memoria, avance de program counter).
  - Abstracciones de datos e instrucciones.
  - Técnicas de programación modular.

- Programación paralela/distribuida
  - Incluye complicaciones adicionales:
    - Multiejecución simultánea.
    - Comunicaciones y sincronización.
  - La modularidad pasa a ser fundamental, para manejar la (potencialmente) enorme cantidad de procesos en ejecución simultánea.

MODELO CONCEPTUAL DE PARALELISMO

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- El paradigma de diseño y programación difiere del utilizado para diseñar y programar aplicaciones secuenciales.
  - Una buena estrategia de división del problema puede determinar la eficiencia de un algoritmo paralelo.
  - Es importante considerar el tipo y la disponibilidad de hardware.

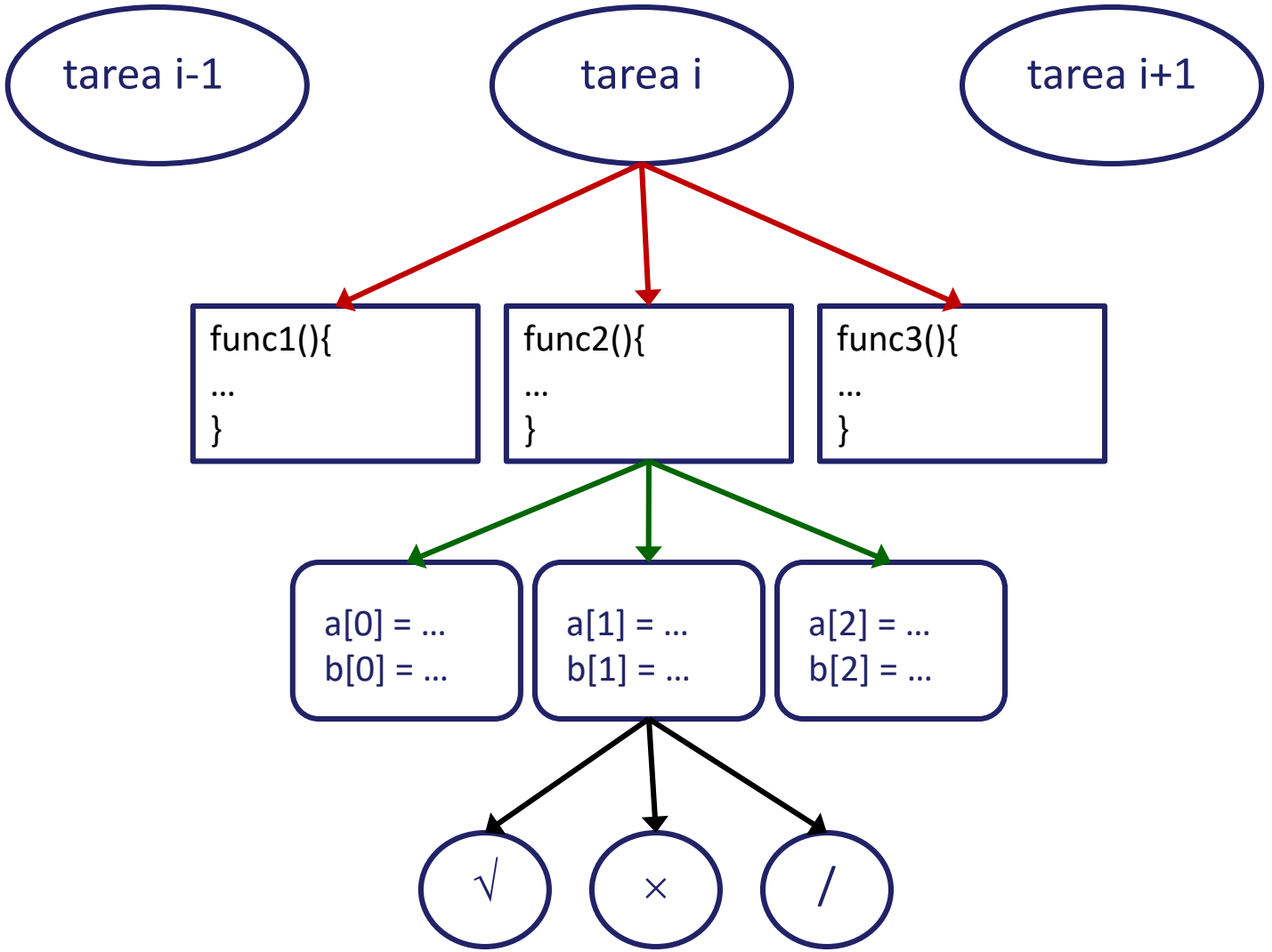
- Respecto a los mecanismos de comunicación entre procesos, existen dos paradigmas de computación paralela
  - MEMORIA COMPARTIDA
  - PASAJE DE MENSAJES
  - Existen también MODELOS HÍBRIDOS, que combinan ambas técnicas.

NIVELES DE PARALELISMO

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- El paralelismo puede aplicarse:
  1. A nivel intrainstrucción (hardware, pipelines).
  2. A nivel interinstrucción (SO, optimización de código, compilador).
  3. A nivel de procedimientos o funciones
     - Algorítmico funcional
  4. A nivel de programas o trabajos
     - Algorítmico tareas
- Los niveles 1 y 2 se estudian en cursos de arquitectura de sistemas y sistemas operativos.
- En este curso se considerarán los niveles 3 y 4 de aplicación de las técnicas de paralelismo (enfocado al diseño y programación de algoritmos paralelos/distribuidos).

NIVELES DE PARALELISMO



Algorítmico tareas

Algorítmico funcional

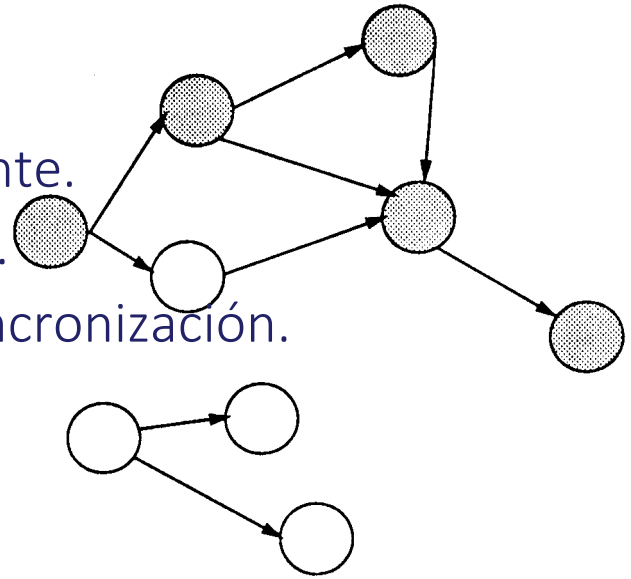
Inter-instrucción

Intra-instrucción

MODELO CONCEPTUAL DE PARALELISMO

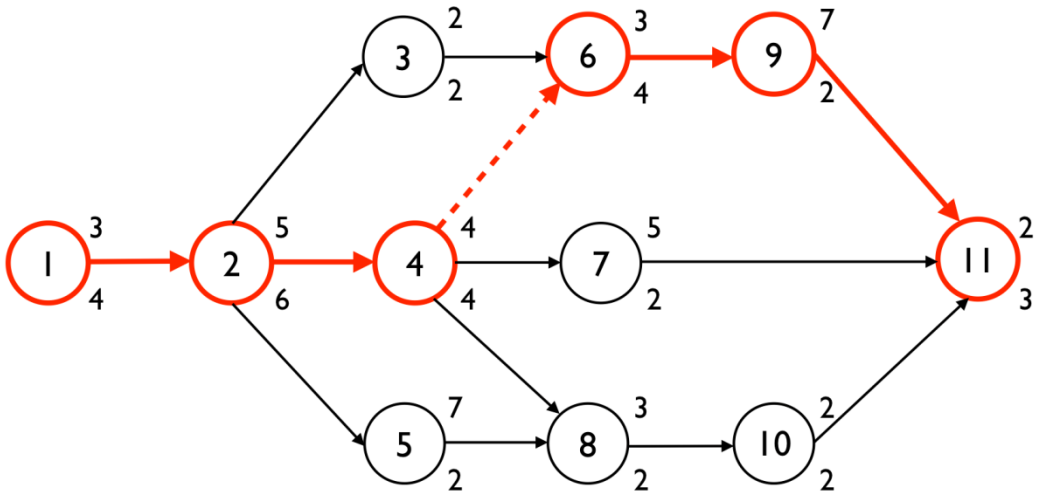
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- GRAFOS DIRIGIDOS ACICLICOS (ADGs)
  - Los nodos representan tareas o procesos (partes de código que ejecutan secuencialmente).
  - Las aristas representan dependencias (algunas tareas preceden a otras).
- El problema a resolver se divide en tareas a ejecutar cooperativamente en múltiples procesadores.
- El diseño e implementación de la aplicación debe:
  - Definir tareas que pueden ejecutar concurrentemente.
  - Lanzar la ejecución y detener la ejecución de tareas.
  - Implementar los mecanismos de comunicación y sincronización.



MODELO CONCEPTUAL DE PARALELISMO

- Las dependencias entre tareas imponen limitaciones al nivel de paralelismo:
  - Dependencia de datos.
  - Sincronizaciones.
- El camino crítico en el grafo de tareas permite determinar el mínimo tiempo posible de ejecución.
  - Técnicas de investigación operativa permiten determinar los niveles de paralelismo (Gantt, planificación).



ESTRATEGIAS DE PROGRAMACIÓN PARALELA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

1. Paralelismo implícito o automático
   - Automático: el usuario no controla el control ni el procesamiento.
   - Se parte de un código existente.
   - Cambios muy menores en el código.
   - Optimizadores de código y compiladores.
2. Paralelismo explícito utilizando bibliotecas
   - Se parte de un código existente.
   - Se implementa la lógica utilizando rutinas de bibliotecas ya paralelizadas.
   - Permite explotar ciertas características de cada subproblema.
3. Paralelismo explícito con rediseño de código
   - No se utiliza código existente.
   - Se trabaja rediseñando la aplicación, para tomar la mayor ventaja de las tareas a realizar concurrentemente.
   - Se implementa la lógica utilizando bibliotecas de programación paralela.

DISEÑO DE ALGORITMOS PARALELOS

- No siempre es una buena decisión partir de un algoritmo secuencial (“paralelizar” una aplicación).
- En ocasiones será necesario diseñar un nuevo algoritmo, que puede ser muy diferente al secuencial.
- Resumiendo las etapas de diseño de aplicaciones paralelas:
  - Identificar el trabajo que puede hacerse en paralelo.
  - Dividir el trabajo y los datos entre los procesadores.
  - Resolver los accesos a los datos, las comunicaciones entre procesos y las sincronizaciones.
  - Asignar recursos de cómputo a los procesos que ejecutarán en paralelo.

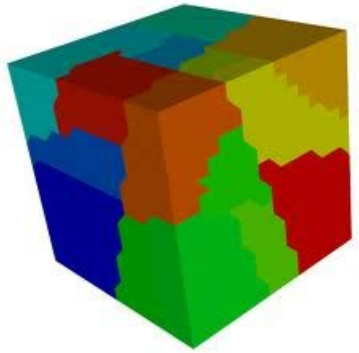
DESCOMPOSICIÓN – ASIGNACIÓN – ORQUESTACIÓN – MAPEO

DISEÑO DE ALGORITMOS PARALELOS

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- DESCOMPOSICIÓN
  - Identifica los procesos concurrentes y decide el nivel y la forma en la cual se explotará la concurrencia.
  - Decisiones:
    - Cantidad de tareas.
    - Tareas estáticas o dinámicas.
    - Criterios de división y utilización de recursos.

- ASIGNACIÓN
  - Asigna los datos a los procesos concurrentes.
  - Criterios de asignación:
    - Estáticos o dinámicos.
    - Balance de cargas.
    - Reducción de costos de comunicación y sincronización.



DISEÑO DE ALGORITMOS PARALELOS

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- ORQUESTACIÓN
  - Se toman decisiones sobre los parámetros de arquitectura, el modelo de programación, los lenguajes o bibliotecas a utilizar
  - Resuelve:
    - Las estructuras de datos, la localidad de referencias.
    - La optimización de comunicaciones y sincronizaciones.

- MAPEO (SCHEDULING)
  - Asigna los procesos a los recursos (procesadores).
  - Criterios de asignación:
    - Desempeño
    - Utilización
    - Reducción de costos de comunicación y sincronización
  - El mapeo puede ser estático, dinámico o adaptativo.

DISEÑO DE ALGORITMOS PARALELOS

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Los mecanismos para definir, controlar y sincronizar tareas deben formar parte del lenguaje a utilizar o ser intercalados por el compilador.
- Estos mecanismos deben permitir especificar:
  - El control de flujo de ejecución de tareas.
  - El particionamiento de los datos.
- Algunos ejemplos:
  - Definición y ejecución de tareas: parbegin-parend (Pascal concurrente), task (Ada), spawn (PVM), fork & join (C).
  - Comunicación y sincronización: pipes, semáforos, barreras (en memoria compartida), mensajes sincrónicos (rendezvouz Ada) y mensajes asincrónicos (C/C++, PVM, MPI) (en memoria distribuida).
  - El particionamiento de los datos es una tarea usualmente asignada al diseñador del algoritmo paralelo.
- El modelo se complica por características peculiares de la computación paralela – distribuida.

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

PROBLEMAS DE LA COMPUTACIÓN PARALELA-DISTRIBUIDA

PROBLEMAS
DE LA COMPUTACIÓN PARALELA-DISTRIBUIDA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- NO DETERMINISMO EN LA EJECUCIÓN
  - Los programas tienen muchos estados posibles.
  - Impide utilizar herramientas de diseño del estilo de los "diagramas de flujo (en el tiempo)" para especificar secuencias de control.

- CONFIABILIDAD
  - Varios componentes del sistema pueden fallar:
    - nodos, interfaces, tarjetas, caches, bridges, routers, repeaters, gateways, medios físicos.
  - Problemas de utilizar equipamiento no dedicado:
    - problemas de uso y tráfico (propio y ajeno), etc.
    - no repetibilidad de condiciones de ejecución.

PROBLEMAS
DE LA COMPUTACIÓN PARALELA-DISTRIBUIDA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- SEGURIDAD
  - Conexión y acceso a equipos remotos.
  - Accesos a datos distribuidos.
- DIFICULTAD DE ESTIMAR LA PERFORMANCE DE UNA APLICACIÓN
  - Como consecuencia del no determinismo en la ejecución.
  - Deben utilizarse criterios estadísticos.
- DIFICULTAD DE ANALIZAR Y VERIFICAR PROGRAMAS
  - Es difícil reproducir escenarios (múltiples estados, asincronismo).
  - Los datos y procesos no están centralizados .
  - Implica la necesidad de un “debugger” en cada nodo utilizado.

PROBLEMAS
DE LA COMPUTACIÓN PARALELA-DISTRIBUIDA

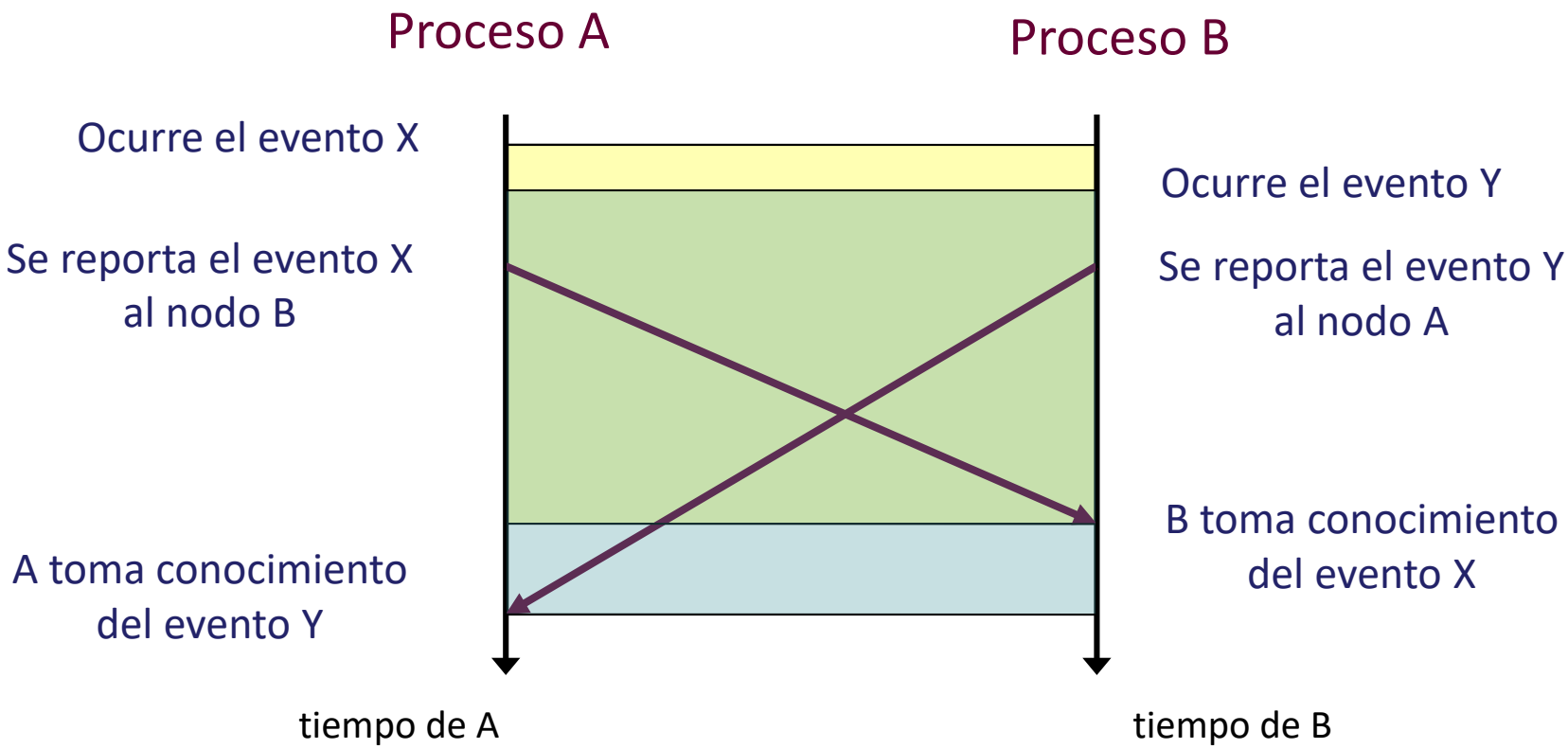
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- INCOMPATIBILIDAD POTENCIAL ENTRE PRODUCTOS Y PLATAFORMAS (para sistemas heterogéneos)
  - Sistemas Operativos: Linux y otros (variantes personalizadas de Unix, Windows, etc.)
  - Protocolos de comunicaciones: TCP/IP, protocolos propietarios
  - Herramientas de software y lenguajes: PVM/MPI, C/Java, bibliotecas de threads, etc....

- USO RACIONAL DE RECURSOS DE CÓMPUTO
  - En sistemas no dedicados

ASINCRONISMO

DEJA DE EXISTIR EL CONCEPTO DE “TIEMPO UNIVERSAL”



“el evento X sucedió antes que el evento Y”

“el evento Y sucedió antes que el evento X”

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

TÉCNICAS de COMPUTACIÓN PARALELA-DISTRIBUIDA

TÉCNICAS de COMPUTACIÓN PARALELA/DISTRIBUIDA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Las técnicas de programación paralela/distribuida aplican estrategias de DESCOMPOSICIÓN o PARTICIONAMIENTO de los datos y del cómputo, que permiten dividir un problema en subproblemas de menor complejidad, a resolver en paralelo.

- El objetivo primario de la descomposición será dividir en forma equitativa tanto los cálculos asociados con el problema como los datos sobre los cuales opera el algoritmo.

TÉCNICAS de COMPUTACIÓN PARALELA/DISTRIBUIDA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- ¿ Cómo lograr el objetivo de la descomposición ?
  - Definir al menos un orden de magnitud más de tareas que de procesadores disponibles (utilización de recursos).
  - Evitar cálculos y almacenamientos redundantes.
  - Generar tareas de tamaño comparable.
  - Generar tareas escalables con el tamaño del problema.
  - Considerar varias alternativas de descomposición, en caso de ser posible.

- Según se enfoque principalmente en la descomposición de datos o de tareas, resultará una técnica diferente de programación paralela.

- Las técnicas más difundidas son las de descomposición de dominio y descomposición funcional.

DESCOMPOSICIÓN de DOMINIO

- Se concentra en el particionamiento de los datos del problema (*paralelismo de datos - data parallel*).
- Se trata de dividir los datos en piezas pequeñas, de (aproximadamente) el mismo tamaño.
- Luego se dividen los cálculos a realizar, asociando a cada operación con los datos sobre los cuales opera.
- Los datos a dividir pueden ser:
  - La entrada del programa
  - La salida calculada por el programa
  - Datos intermedios calculados por el programa

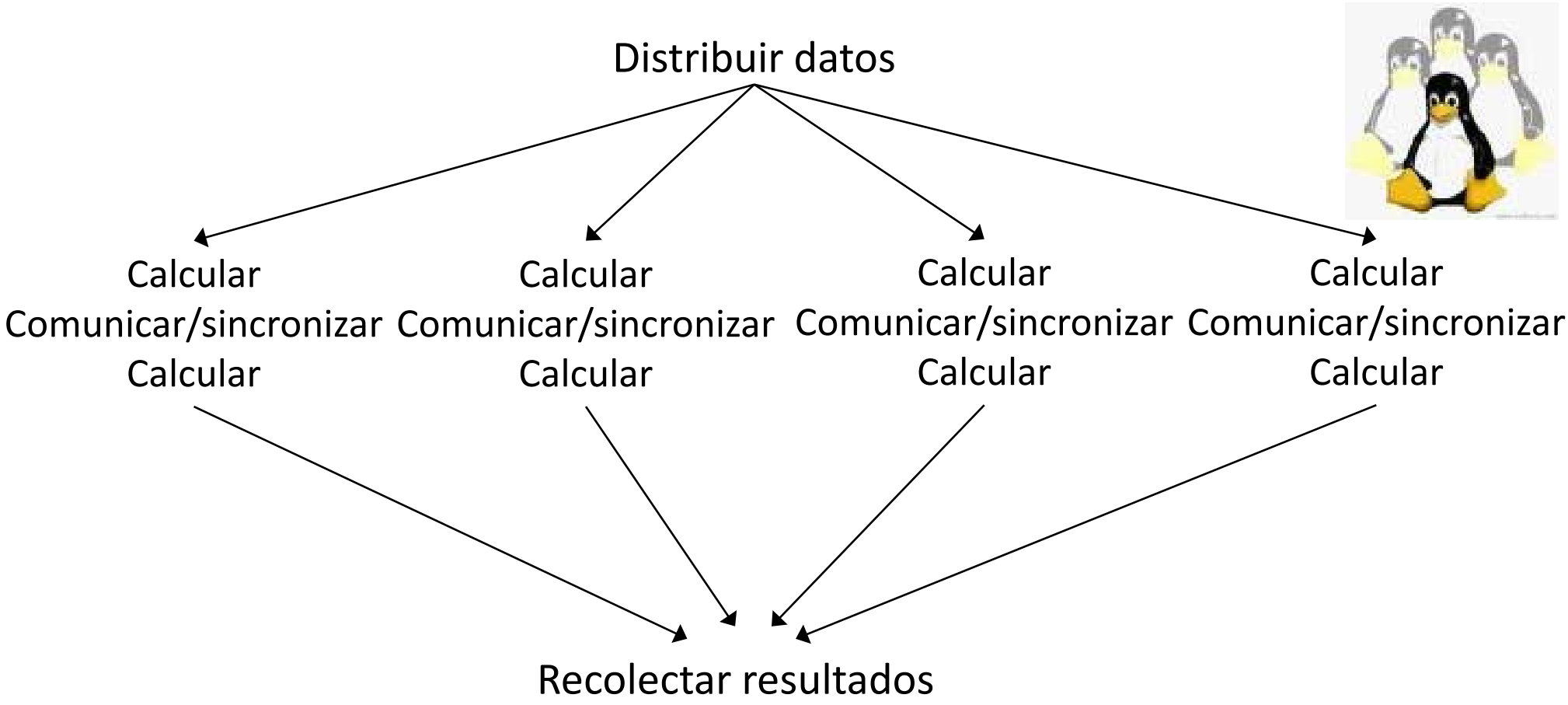
DESCOMPOSICIÓN de DOMINIO

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Si bien no existe una regla general para determinar como realizar la división de datos, existen algunas sugerencias obvias dadas por:
  - La estructura o “geometría” del problema.
  - La idea de concentrarse primero en las estructuras de datos más grandes o las accedidas con mayor frecuencia.
- Ejemplo 1: paralelizar un producto cartesiano entre dos repositorios o un join entre tablas en una base de datos.
  - Debe particionarse la tabla más grande.
- Ejemplo 2 : Procesamiento de imágenes
  - División de dominios de cálculo.
  - Mismo programa en cada dominio.
  - Comunicación necesaria para cálculos en los bordes.
- La técnica de descomposición de dominio se asocia comúnmente con la estrategia de Divide&Conquer.

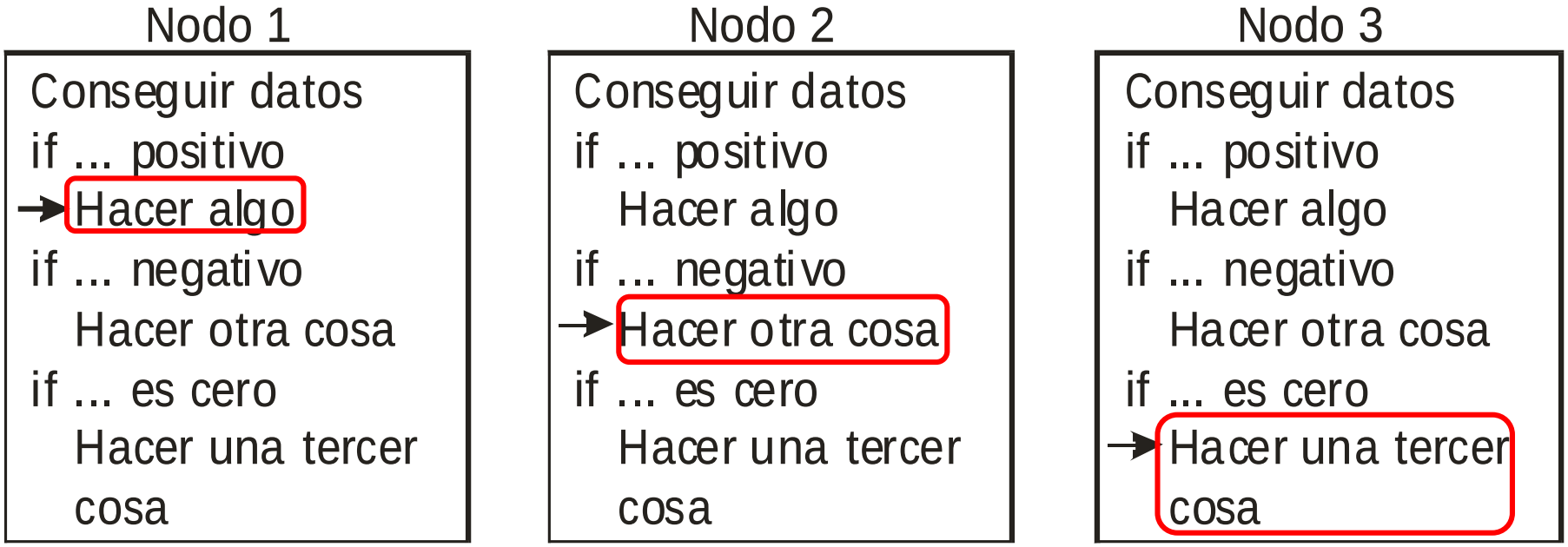
DESCOMPOSICIÓN de DOMINIO

- Modelo de programación SPMD (Single Program, Multiple Data)



DESCOMPOSICIÓN de DOMINIO

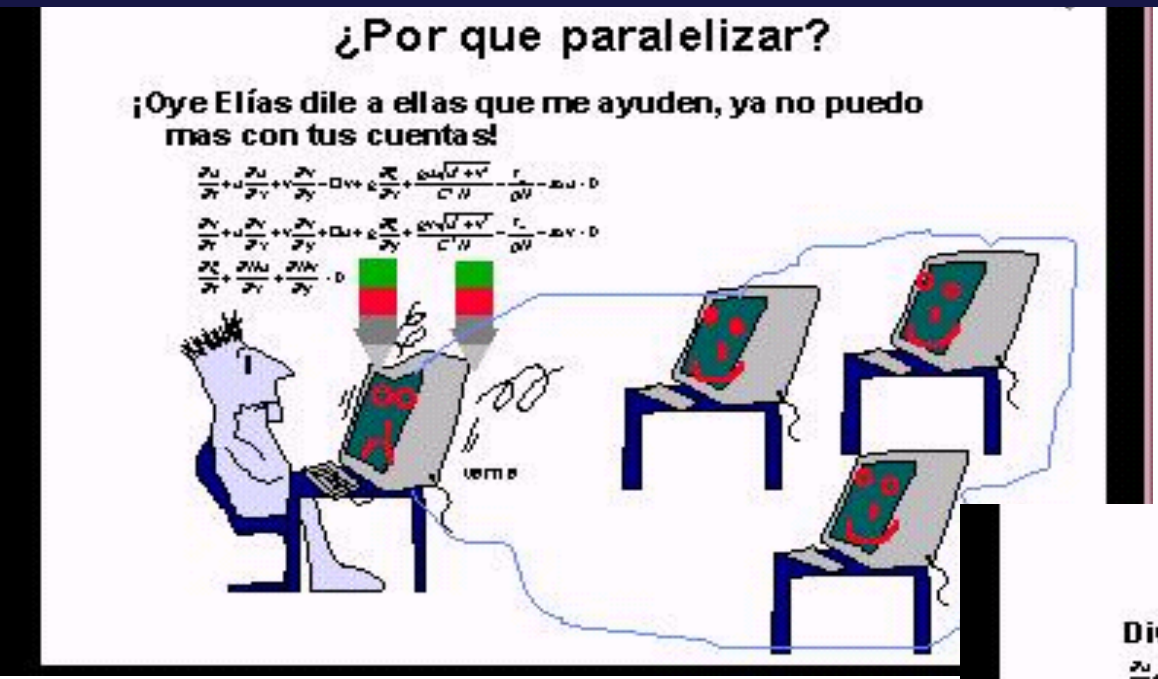
- Modelo de programación SPMD (Single Program, Multiple Data).
- Un mismo programa se ejecuta sobre diferentes conjuntos de datos.

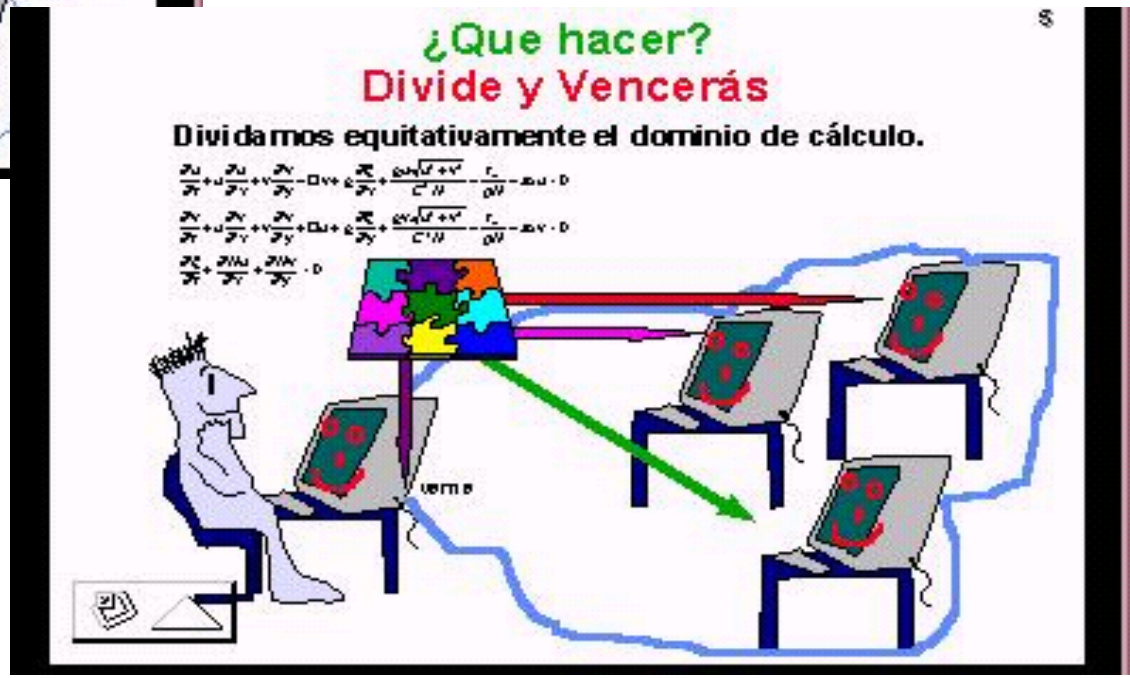


Todos los nodos ejecutan el mismo programa,
pero no necesariamente las mismas instrucciones

- Es un modelo muy utilizado en clusters y sistemas grid/cloud.

DESCOMPOSICIÓN de DOMINIO



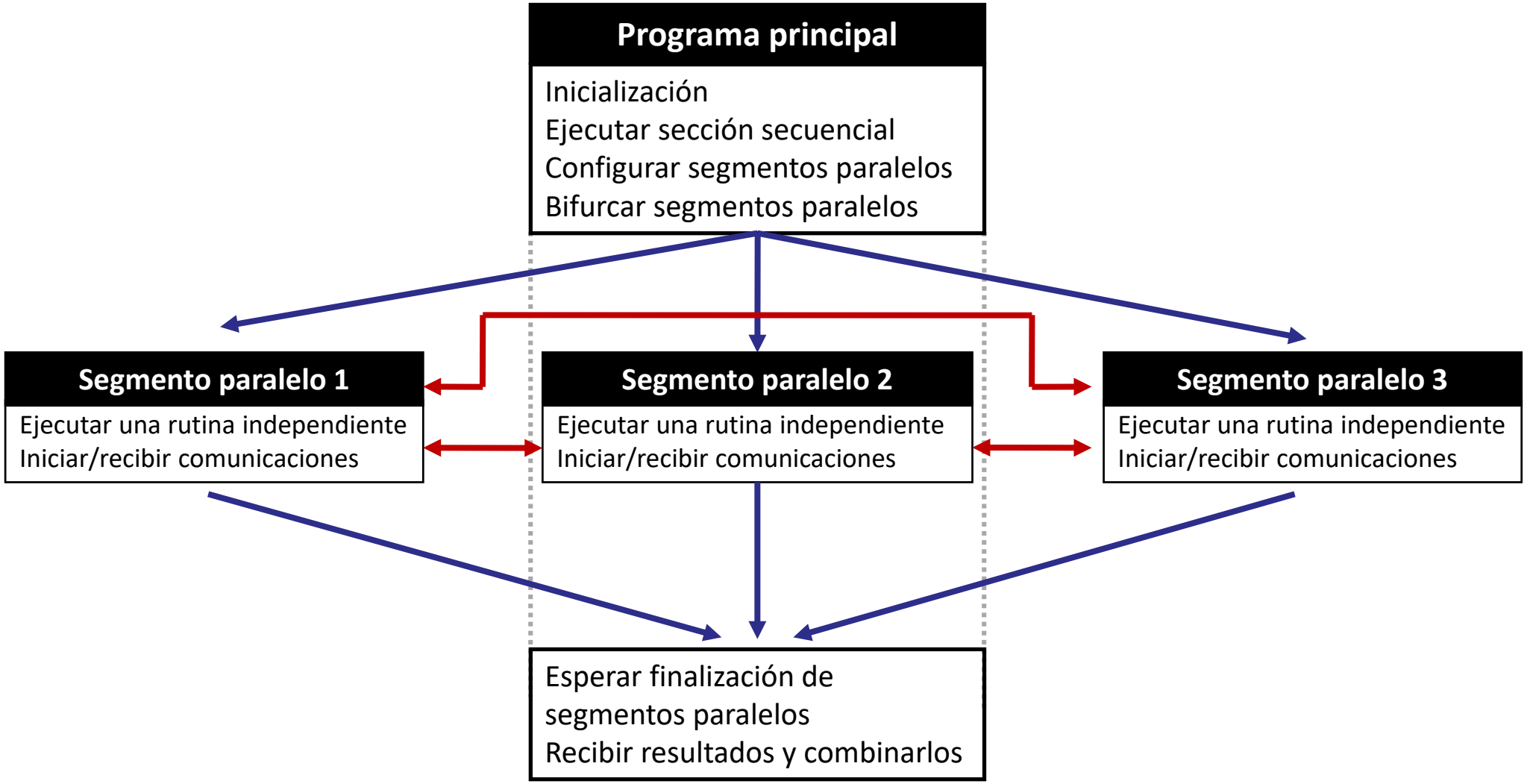


DESCOMPOSICIÓN FUNCIONAL

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

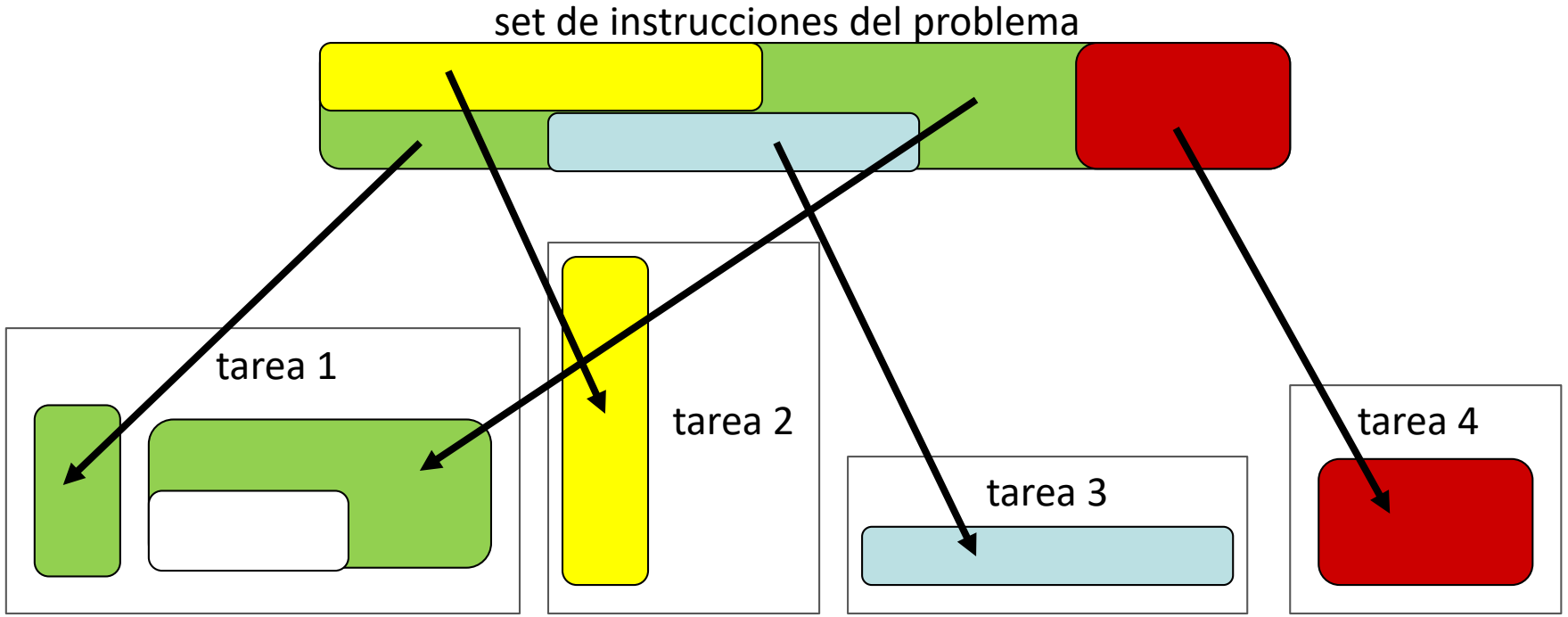
- Se concentra en el particionamiento de las operaciones del problema (*paralelismo de control - control parallel*).
- Se trata de dividir el procesamiento en tareas disjuntas.
- Luego se examinan los datos que serán utilizados por las tareas definidas.
- Si los datos son disjuntos, resulta un PARTICIONAMIENTO COMPLETO.
- Si los datos NO son disjuntos, resulta un PARTICIONAMIENTO INCOMPLETO (el caso más usual). Se requiere replicar los datos o comunicar datos entre los procesos asociados a las diferentes tareas.
- Modelos de computación basada en agentes, computación basada en servicios, etc.

DESCOMPOSICIÓN FUNCIONAL



DESCOMPOSICIÓN FUNCIONAL

- Casos típicos:
  - Distribuir código para asociar requerimientos de procesos a recursos locales.
    - Cada tarea trabaja temporalmente con sus datos locales, pero se requiere comunicación para lograr la cooperación.
  - Procesamiento de datos en bases de datos relacionales o distribuidas.
    - Diferentes tareas realizan diferentes operaciones.



PARALELISMO OPTIMISTA

- Realizar operaciones adicionales previendo que deban ser ejecutadas en el futuro (*look-ahead execution*).
  - Bajo la hipótesis de que hay recursos disponibles para ejecutarlas.

```
if condicion then
    Funcion1()
else
    Funcion2()
end if
```

  - Evaluar Función1() y Funcion2() de antemano, independientemente del valor de la condición.
- Típico en aplicaciones de tiempo real.
  - Ejemplo : sistemas que requieren conexión/autenticación.
- Típico mecanismo para proveer tolerancia a fallos.
  - En aplicaciones distribuidas en Internet.

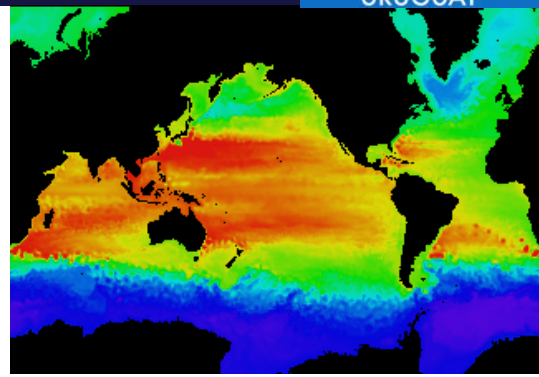
MODELOS HÍBRIDOS

- Incluyen dos o más tipos de programación paralela para la resolución de una misma aplicación.

- Comúnmente utilizados en programas paralelos distribuidos en Internet (donde casi siempre existe la posibilidad de conseguir recursos ociosos adicionales).
  - Se combina una estrategia "tradicional" de descomposición con una estrategia de paralelismo optimista, para mejorar la eficiencia o proveer mecanismos para tolerancia a fallos.
  - Las principales ventajas son eficiencia y robustez, aunque el manejo de los distintos modelos de paralelismo puede incluir complejidades adicionales.

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 1000 Flops por punto por minuto
- Predicción del tiempo (7 días, cada 24 horas): 560 GFLOPS
- Predicción climática (50 años, cada 30 días): 48 TFLOPS
- Perspectiva:
  - En un computador tradicional con procesador de 3GHz (≈10 GFLOPS) la predicción climática demandaría más de 1000 años de tiempo de cómputo.

Es necesario disponer de estrategias más potentes para el análisis

CASO DE ESTUDIO 2
ANÁLISIS DE DATOS

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- 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 intentar fraudes a los servicios públicos?
  - ¿Cuándo conviene poner en oferta la cerveza?
  - ¿Qué tipo de publicidad enviar a un cliente?
- Recolección de datos:
  - Sensores remotos y dispositivos IoT.
  - Microarrays generando datos genéticos.
  - Información empresarial y/o simulaciones generando terabytes de datos
  - Información de uso de software y visitas a sitios web.
  - 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, técnicas de inteligencia computacional para realizar comparaciones y relaciones que permitan 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.

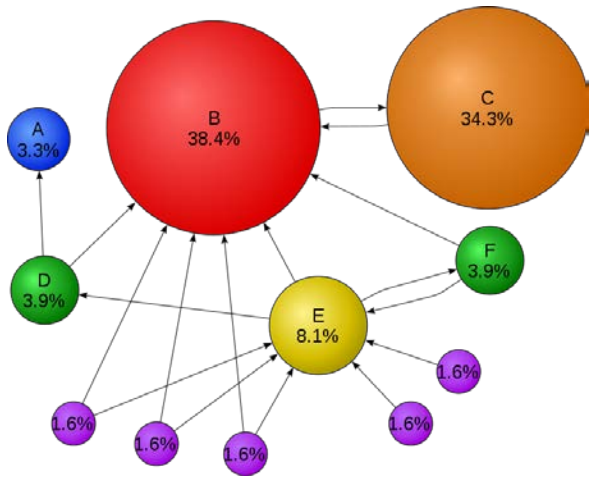
Es necesario disponer de métodos más potentes para el análisis

CASO DE ESTUDIO 3
ANÁLISIS DE DATOS: PAGERANK DE GOOGLE

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- 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".

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



CASO DE ESTUDIO 3
ANÁLISIS DE DATOS: PAGERANK DE GOOGLE

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

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

valores de PageRank que tienen las páginas que enlazan a la página i

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

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

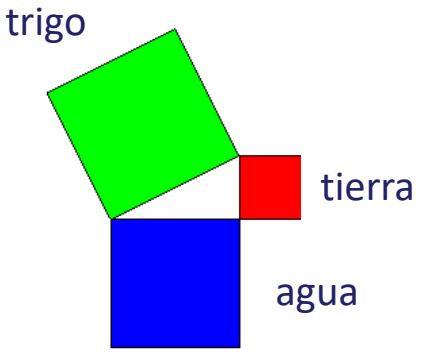
- El PageRank no se actualiza instantáneamente, ni siquiera diariamente [tarda muchos días en completarse].
- Datos:
  - 30.000 millones de páginas en 2005 (Yahoo)
  - 90.000 millones en 2007 (estimado Google)
  - Google dejó de reportar luego de indicar que indexaba 8.000 millones
  - Más de 50.000 millones de páginas indexadas (estimado 2014)

CASO DE ESTUDIO 4
SETI @HOME

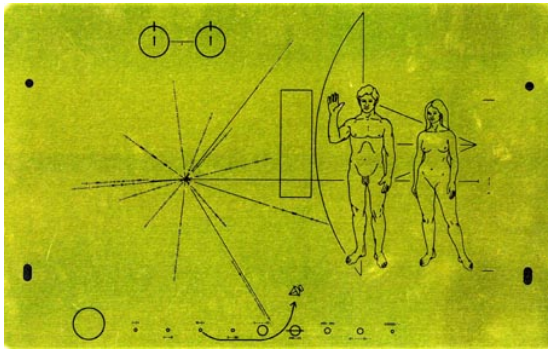
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- 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

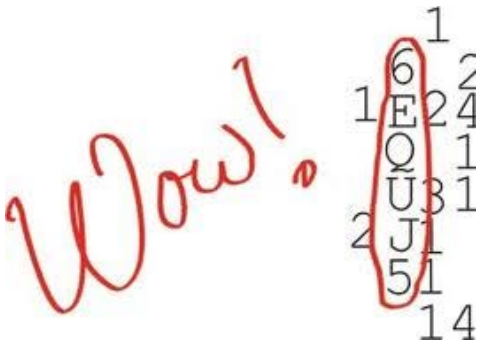


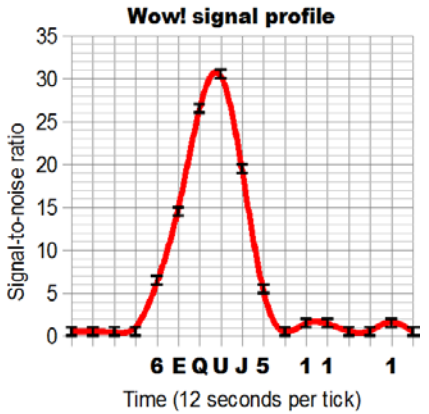
CASO DE ESTUDIO 4
SETI @HOME

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Nikola Tesla (1899) anunció “señales coherentes desde Marte”
- Guglielmo Marconi (1920) detectó “señales extrañas desde el espacio”
- Frank Drake (1960): proyecto Ozma, buscó en canal de 1420-1420.4 MHz

- SETI (Search for ExtraTerrestrial Intelligence)
  - Universidad de California (desde 1971)
  - Utiliza métodos científicos para la búsqueda de emisiones electromagnéticas por parte de civilizaciones en planetas lejanos





CASO DE ESTUDIO 4
SETI @HOME

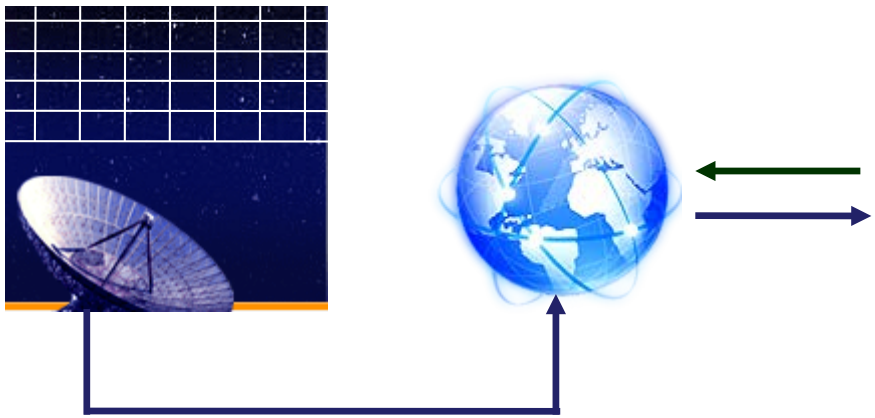
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

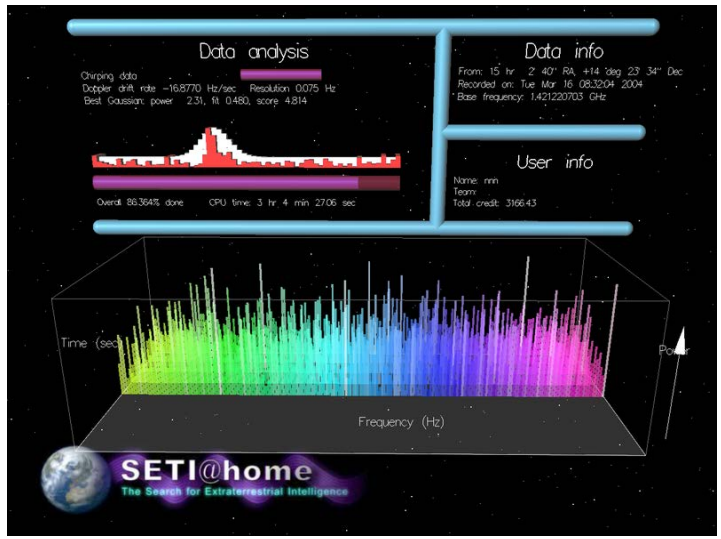
- Avances en SETI
  - Corrección del efecto Doppler coherente
    - Ancho de canal más fino, incrementa la sensitividad
  - Resolución variable de ancho de banda y tiempo
  - Búsqueda de múltiples tipos de señales
    - Análisis de distribución Gaussiana
    - Búsqueda de pulsos repetidos
- Problema: 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



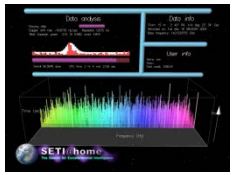


DESCOMPOSICIÓN DE DOMINIO

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

SETI@HOME

Distribuir datos

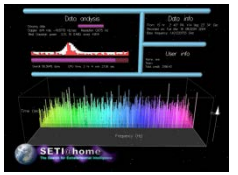


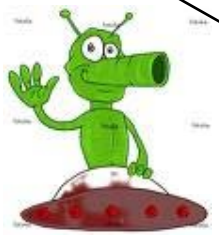
Calcular
Enviar resultados

Calcular
Enviar resultados

Calcular
Enviar resultados

Calcular
Enviar resultados





Recolectar resultados

CASO DE ESTUDIO 4
SETI @HOME

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Estadísticas
  - Más de 5 millones de usuarios (mayor número para un proyecto de computación distribuida)
  - Más de 3 millones de computadores en 253 países
  - Más de 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).
  - Capacidad de cómputo mayor a 1 PFLOPS
  - Más de 1000 años de tiempo de cómputo por día
  - Más de tres 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.

CONSIDERACIONES IMPORTANTES

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- HARDWARE y UTILITARIOS
  - Tecnología, poder y cantidad de los elementos de procesamiento.
  - Conectividad entre elementos.
  - Middleware para gestión de recursos computacionales.
- TÉCNICAS de PROGRAMACIÓN
  - Abstracciones y primitivas para cooperación.
  - Mecanismos para particionar el problema e implementar comunicaciones.
  - Bibliotecas de desarrollo.

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