

Computación de Alto Rendimiento

Diego Alberto Rincón Yáñez MSc

drincony@poligran.edu.co

INSTITUCIÓN UNIVERSITARIA POLITÉCNICO
GRANCOLOMBIANO

Facultad de Ingeniería

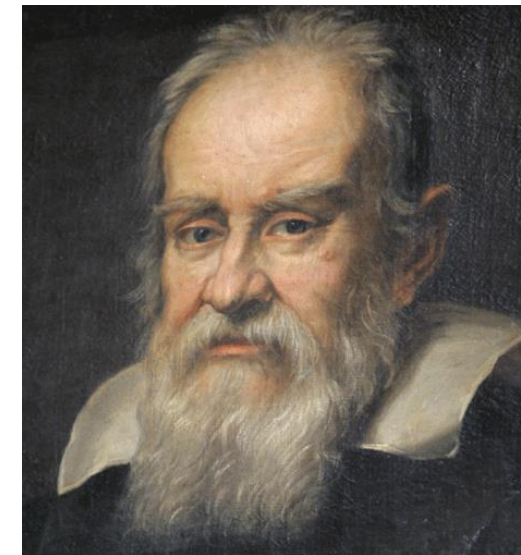
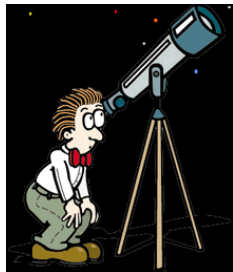
Introducción a la Computación Distribuida

E-Ciencia

"e-Science is about global collaboration in key areas of science and the next generation of infrastructure that will enable it." -

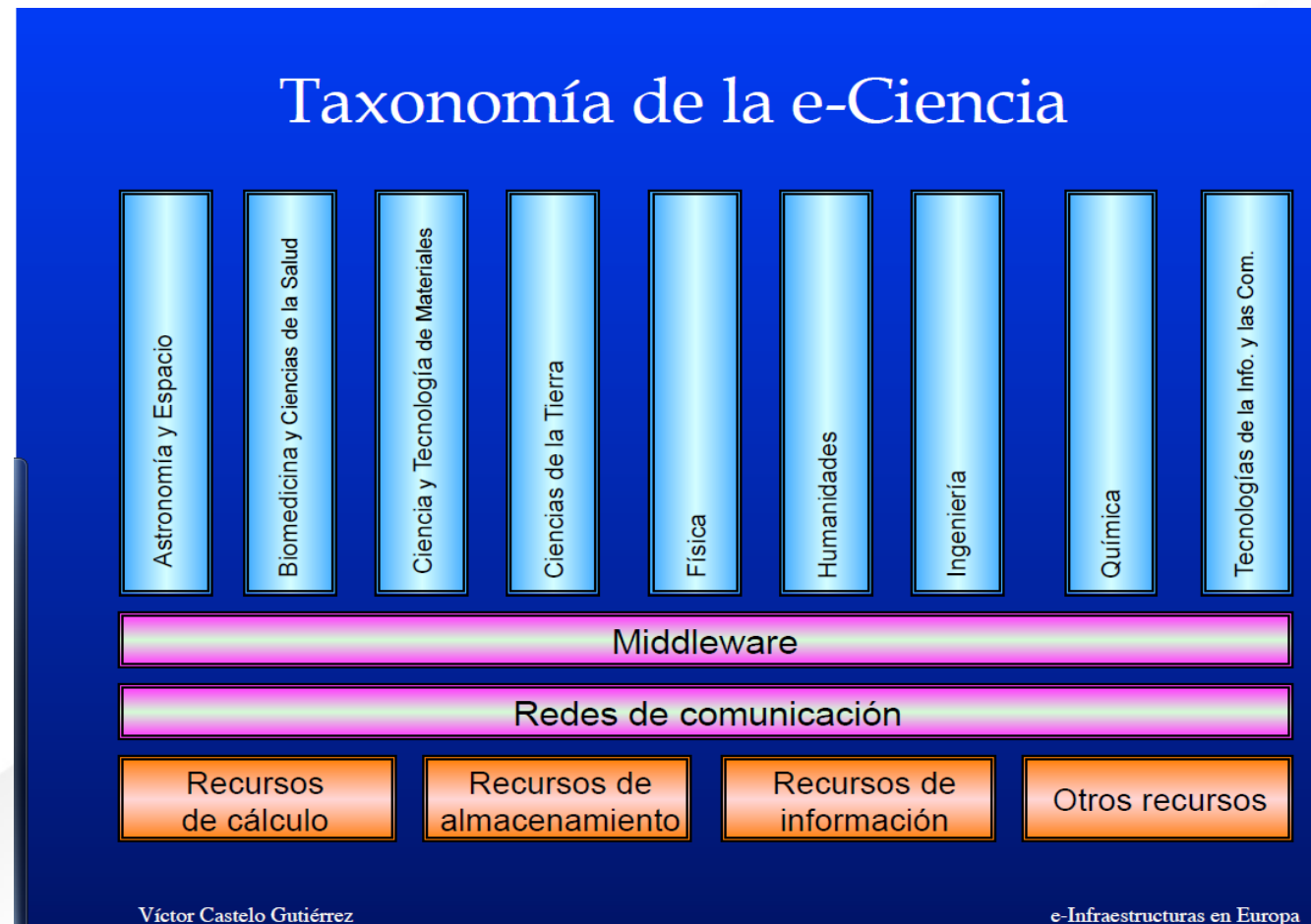
Dr John Taylor, Director General of Research Councils

"The term "e-Science" denotes the systematic development of research methods that exploit advanced computational thinking" - Professor Malcolm Atkinson, e-Science Envoy.



Introducción a la Computación Distribuida

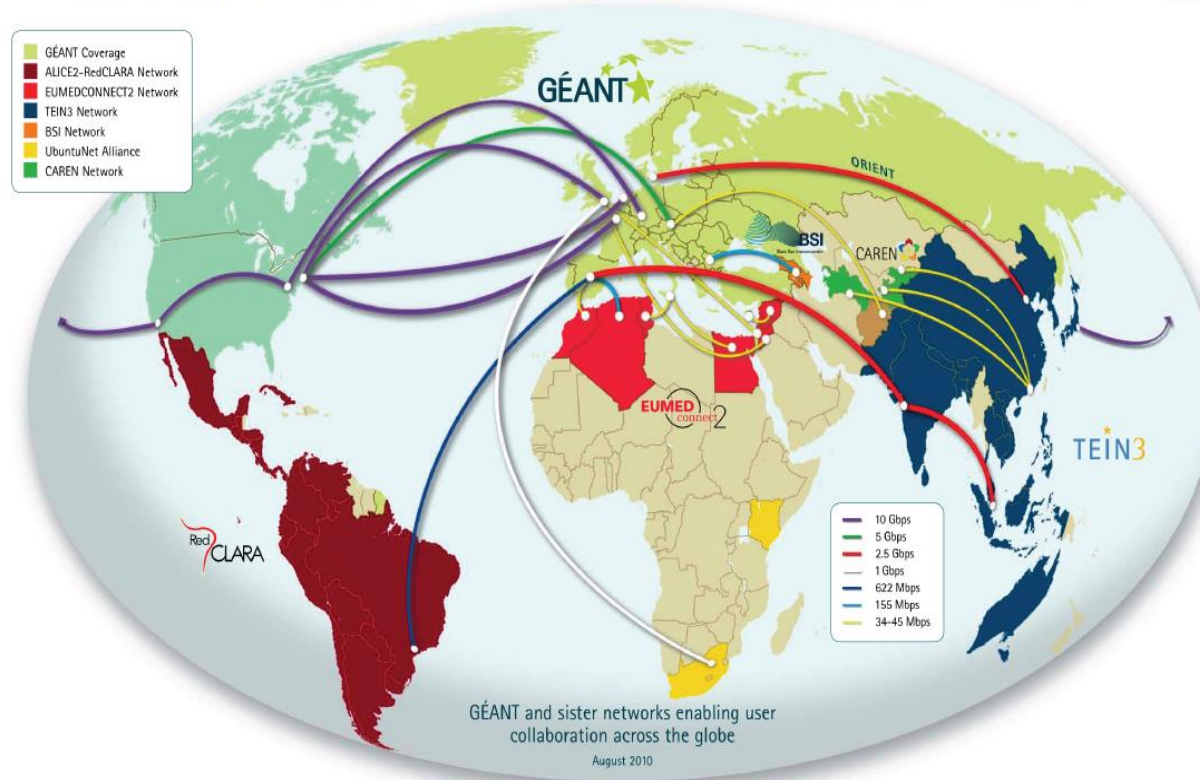
E-Ciencia



Introducción a la Computación Distribuida

E-Ciencia

GÉANT At the Heart of Global Research Networking



DANTE **GÉANT**
www.dante.net www.geant.net

connect • communicate • collaborate

GÉANT is co-funded by the European Commission within its 7th R&D Framework Programme.
This document has been produced with the financial assistance of the European Union. The contents of this document are the sole responsibility of DANTE and can under no circumstances be regarded as reflecting the position of the European Union.



Introducción a la Computación Distribuida

Supercomputación

- Es lo mas grande y rápido en computación en este preciso momento.
- Es también la computadora mas grande y rápida del mundo es este instante.
- Hay una regla?, un supercomputador es mínimo 100 veces mas rápido que el PC mas rápido del mundo

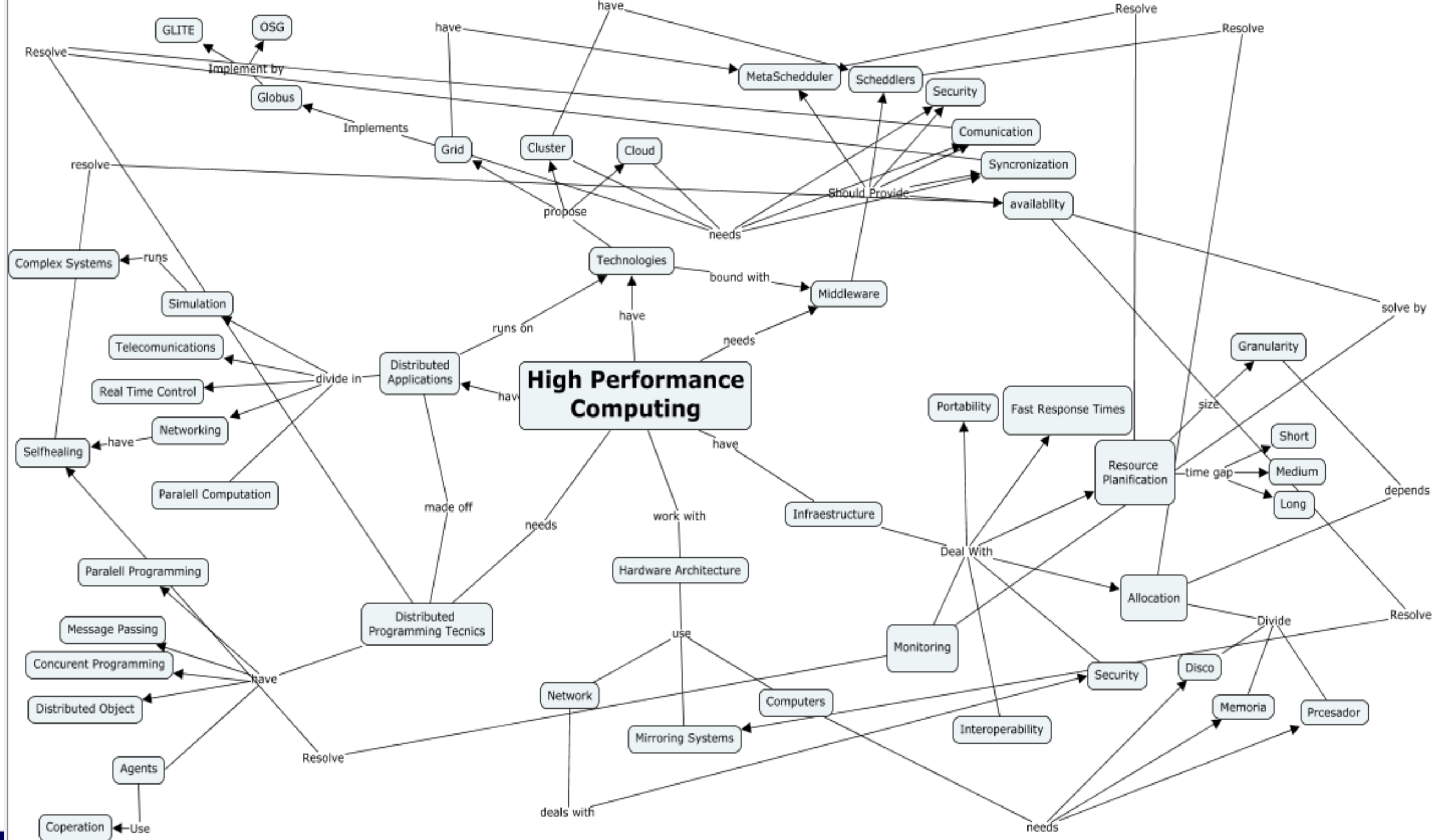


Introducción a la Computación Distribuida

Supercomputación

- La supercomputación también se conoce como:
- Computación de Alto Rendimiento - High Performance Computing (HPC).
- High End Computing (HEC).
- Ciberinfraestructura - Cyberinfrastructure (CI).

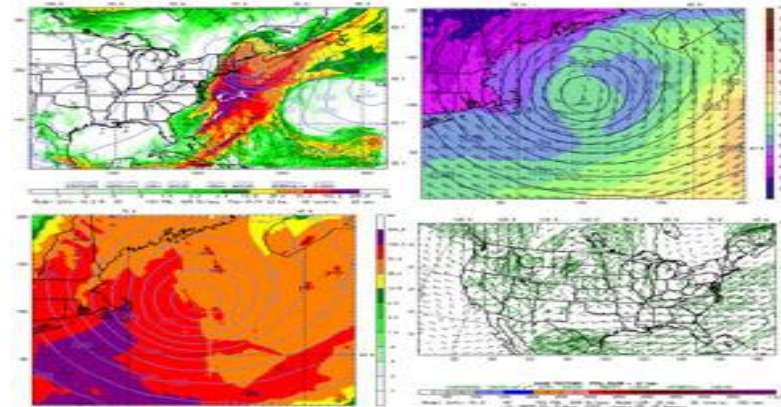
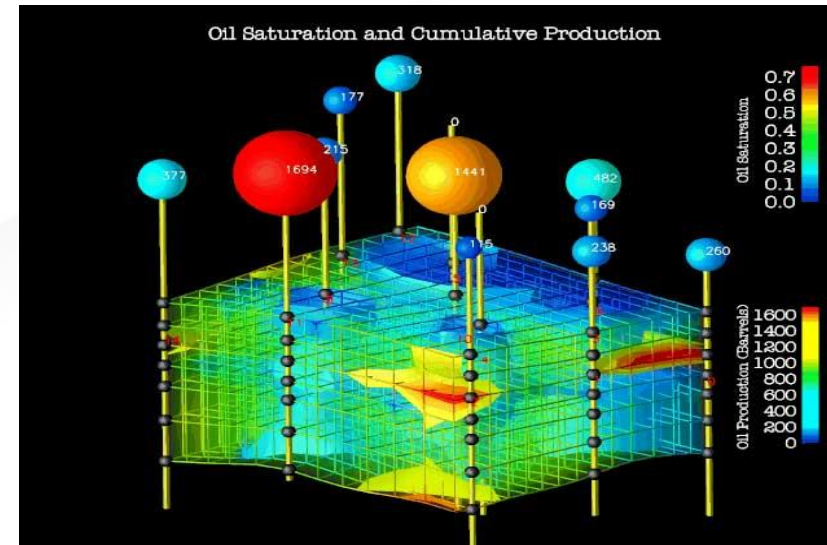




Introducción a la Computación Distribuida

Estos procesos podrían ser:

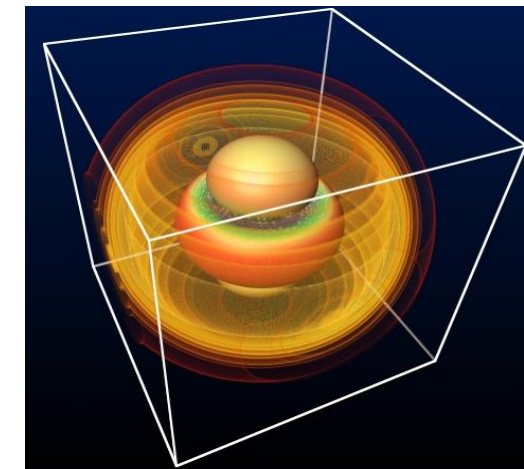
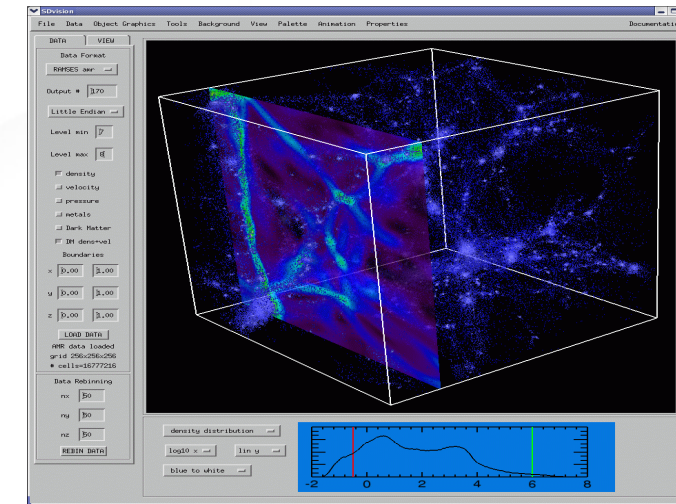
- Simulación de fenómenos físicos, tales como:
- Pronóstico del tiempo
- Formación de Galaxias
- Administración de las reservas de petróleo



Introducción a la Computación Distribuida

Estos procesos podrían ser:

- Minería de datos:
- Secuenciación de genes
- Procesamiento de señales
- Detección de las tormentas que pueden producir tornados

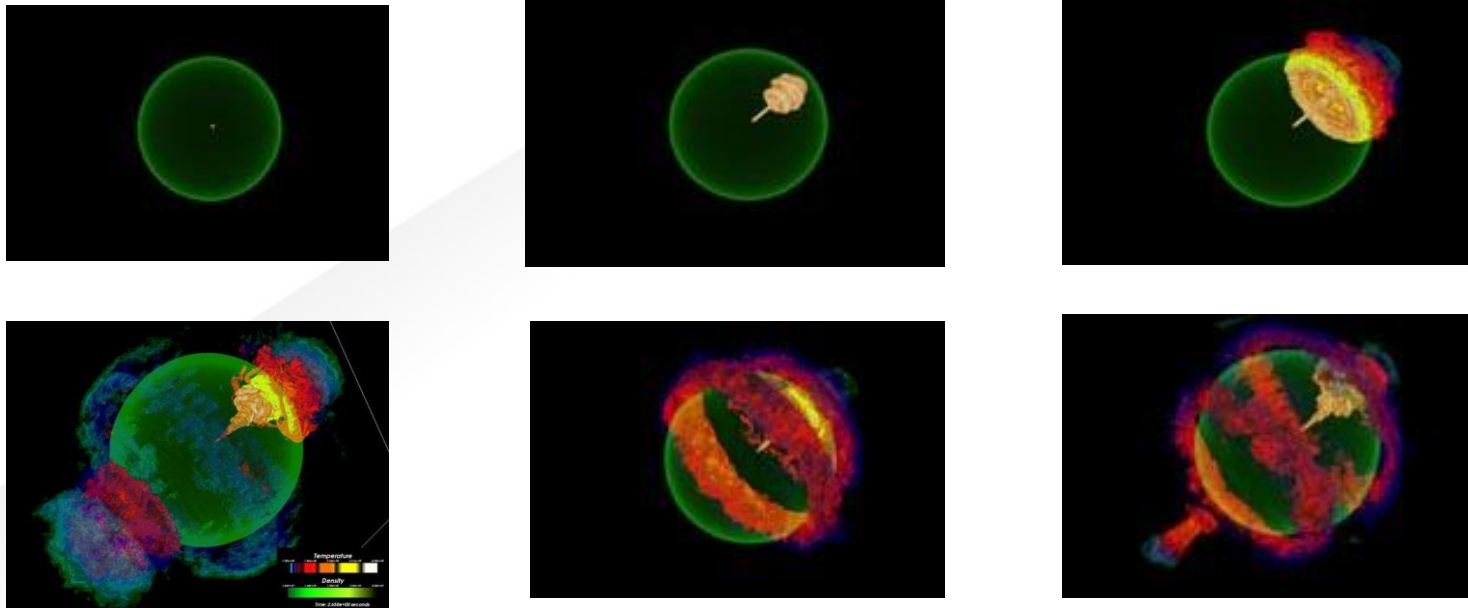


Introducción a la Computación Distribuida

Ejemplo

En el laboratorio Argonne el supercomputador Blue Gene/P permite ver la simulación de la explosión de una estrella supernova, este proceso en un PC tardaría 22 millones de horas el evento tarda solo 5 segundos.

- Este supercomputador tiene mas de 160.000 procesadores



Introducción a la Computación Distribuida

Computadora Paralela: Máquina con dos o más procesadores que pueden trabajar simultánea y/o coordinadamente.

Memoria Compartida: En una máquina paralela existe una sola memoria que puede ser accedida por todos los procesadores.

Memoria Distribuida: Cada uno de los procesadores de un multiprocesador tiene asociado a él una unidad de memoria.

Conceptos básicos

Computación paralela

Máquina con dos o más procesadores que ejecutan trabajos simultáneamente compartiendo los recursos de hardware de la propia máquina.

Computación distribuida

Múltiples máquinas separadas físicamente conectadas a través de la red de datos que permite ejecutar trabajos dividiendo los problemas grandes en tareas mas pequeñas distribuidas entre las máquinas conectadas.

Qué es un CLUSTER ?

- Es un conjunto de máquinas conectadas a través de la red.
- Permite que el usuario final perciba al CLUSTER como una sola máquina de grandes prestaciones computacionales.
- Permite la escalabilidad del sistema mediante la adición de nuevo hardware.
- Requiere de software que coordine el trabajo entre las máquinas.
- Esta compuesto por un *nodo principal* que coordina la ejecución de trabajos entre los *nodos trabajadores* y *nodos cliente* para el envío de trabajos.

Tipos de CLUSTER

1. **Alto rendimiento (HPC)** ejecutan tareas que requieren de alta capacidad computacional (procesamiento, memoria, almacenamiento). No garantiza la disponibilidad.
2. **Alta disponibilidad (HAC)** garantiza la prestación continua del servicio y la recuperación ante posibles fallos.
3. **Alta eficiencia (HTC)** su objetivo es el de ejecutar la mayor cantidad de trabajos en el menor tiempo posible.

Clasificación de los CLUSTER

De acuerdo a su configuración de hardware.

Homogéneos: iguales tipos de tecnologías y plataformas.

Heterogéneos: distintos tipos de tecnologías y plataformas.

De acuerdo al nivel de dedicación del sitio.

Dedicado: su funcionamiento es exclusivo para prestar servicios al CLUSTER.

No dedicado: su destinación al CLUSTER es parcial de acuerdo a la disponibilidad de la máquina (oportunista).

Para que sirve un CLUSTER

Para ejecutar trabajos complejos que requieren de altos volúmenes de recursos de cómputo (procesamiento, memoria y almacenamiento) que no pueden ser ubicados en máquinas individuales.

Estos trabajos pueden dejarse ejecutando durante varios días gracias a la *disponibilidad* del CLUSTER mientras se realizan los cálculos y procesamientos solicitados. Posteriormente el usuario final puede recuperar los resultados obtenidos.

Para que no sirve un CLUSTER

No es posible ejecutar cualquier tipo de aplicaciones, estas deben estar diseñadas para aprovechar las características del CLUSTER.

No está diseñado para obtener resultados en “tiempo real”, todo depende del flujo de trabajo al cual esté sometido el CLUSTER HTC.

Los trabajos ejecutados no pueden interactuar directamente con el usuario a través de un GUI.

Ambientes de programación

- La programación es dependiente de la arquitectura (32/64bits, otros) y sistema operativo.
- Para que un problema sea distribuido se debe hacer uso de un sistema de colas y/o librerías especiales:
 - PVM (*parallel virtual machine*) usada especialmente en cluster con nodos heterogéneos (arquitectura del procesador, sistemas operativo, entre otras), y pertenecientes a diferentes dominios de red.
 - MPI (*Message passage interface*), librería usada para cluster homogéneos.

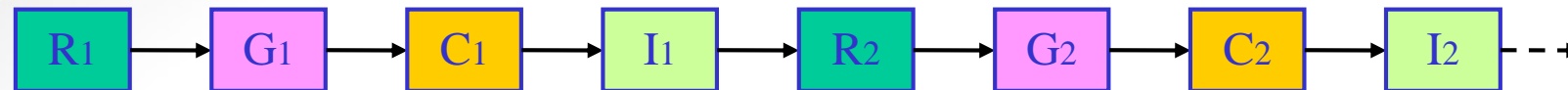


Alto Desempeño vs Paralelizar

Solución No-Paralela

Programa Secuencial

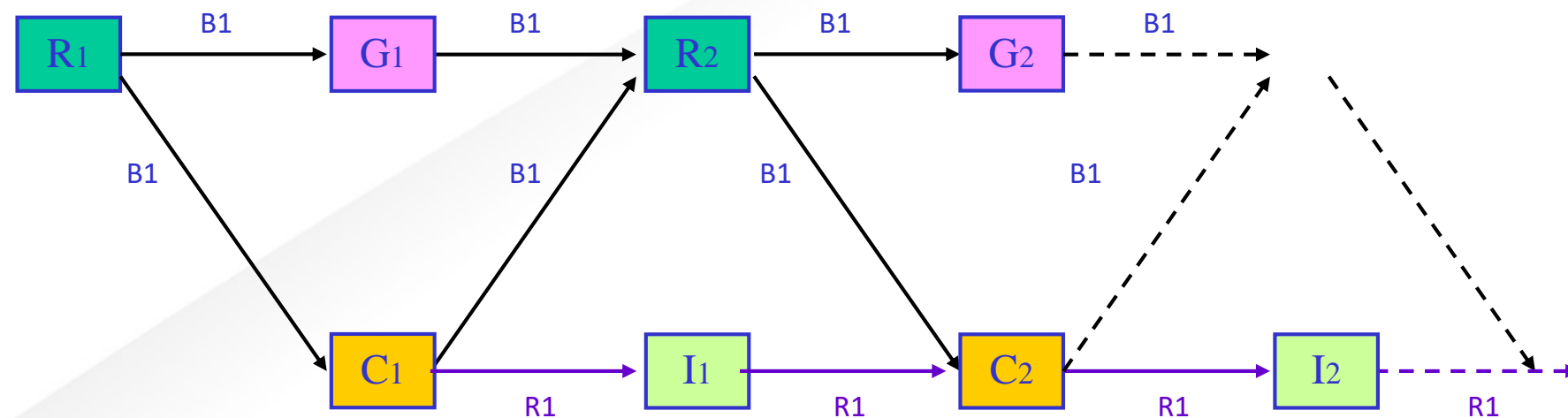
```
while (!fin) {  
    bufer = recoger();  
    guardar(bufer);  
    result = calcular(bufer);  
    informar(result);  
}
```



Simplicidad vs. **Ineficiencia**

Solución Paralela Base

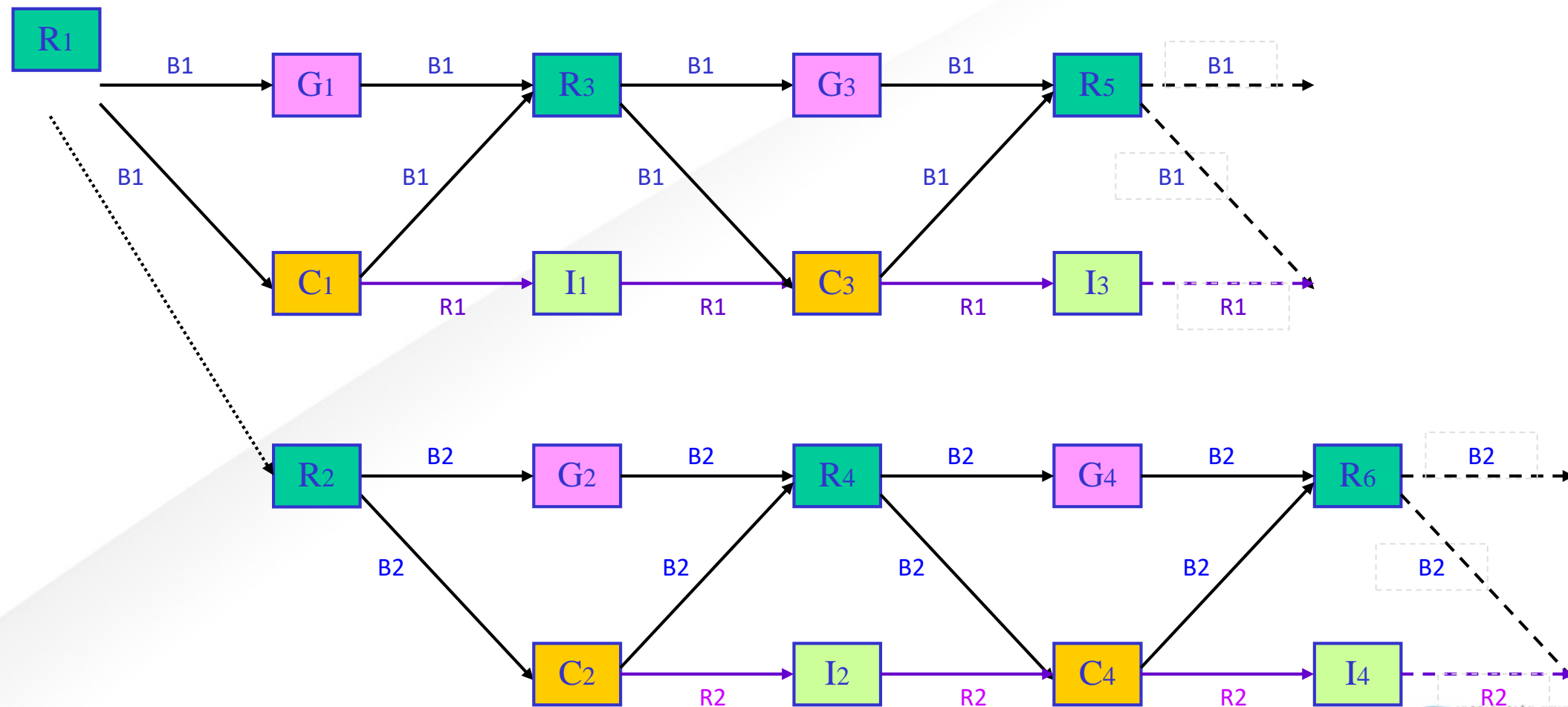
Procesos Paralelos y Precedencia Relativa



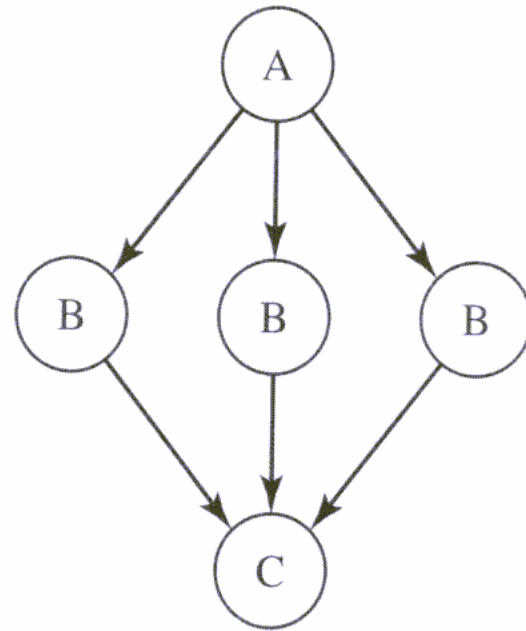
Paralelismo vs. **Sincronización**

Solución Más Paralela

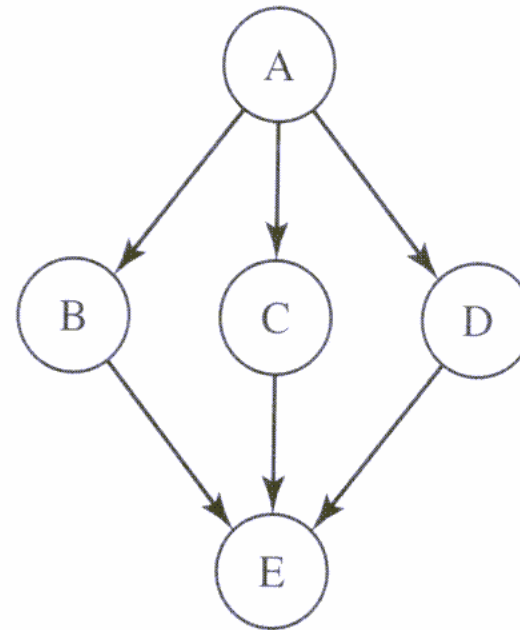
Procesos Paralelos y Memoria



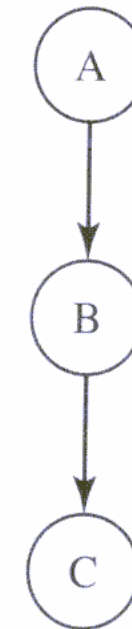
Tipos Base de Concurrency



Paralelismo de
Datos

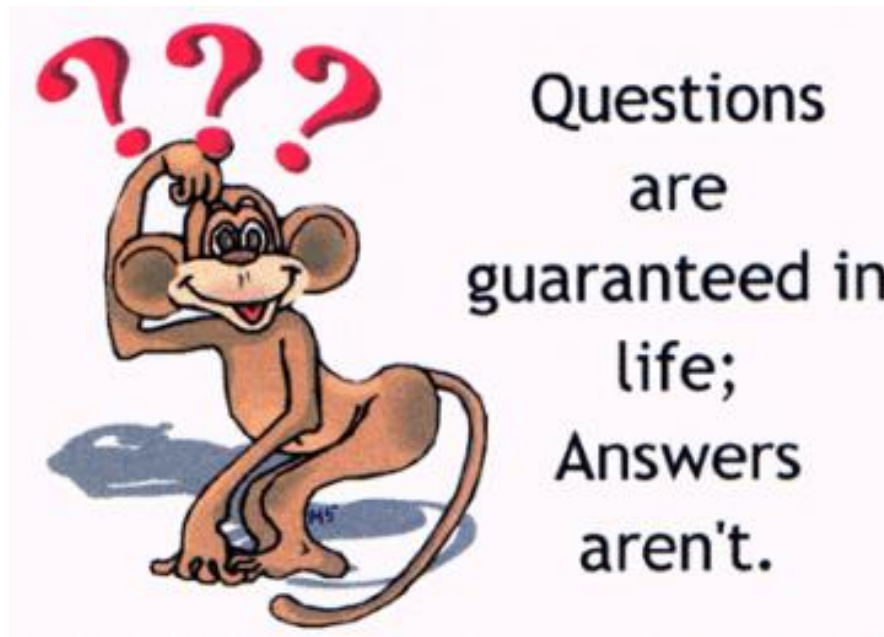


Paralelismo
Funcional



Pipeline

[QUI2004]



¿Preguntas?

Diego Alberto Rincón Yáñez MCSc.
Twitter: @d1egoprog.



INSTITUCIÓN UNIVERSITARIA
**POLITÉCNICO
GRANCOLOMBIANO**

¡GRACIAS!

POLI.EDU.CO |   Poligran | MIEMBRO DE LA RED
ILUMNO