

CLUSTERS

Antonio Antiñolo Navas

Antonio.Antinolo@uclm.es

Arquitectura e Ingeniería de Computadores

Profesor: Serafín Benito Santos

ESI-UCLM

Índice

1. Introducción.
2. Clasificación.
3. Ventajas y Desventajas.
4. Diseño.
5. Ejemplos.

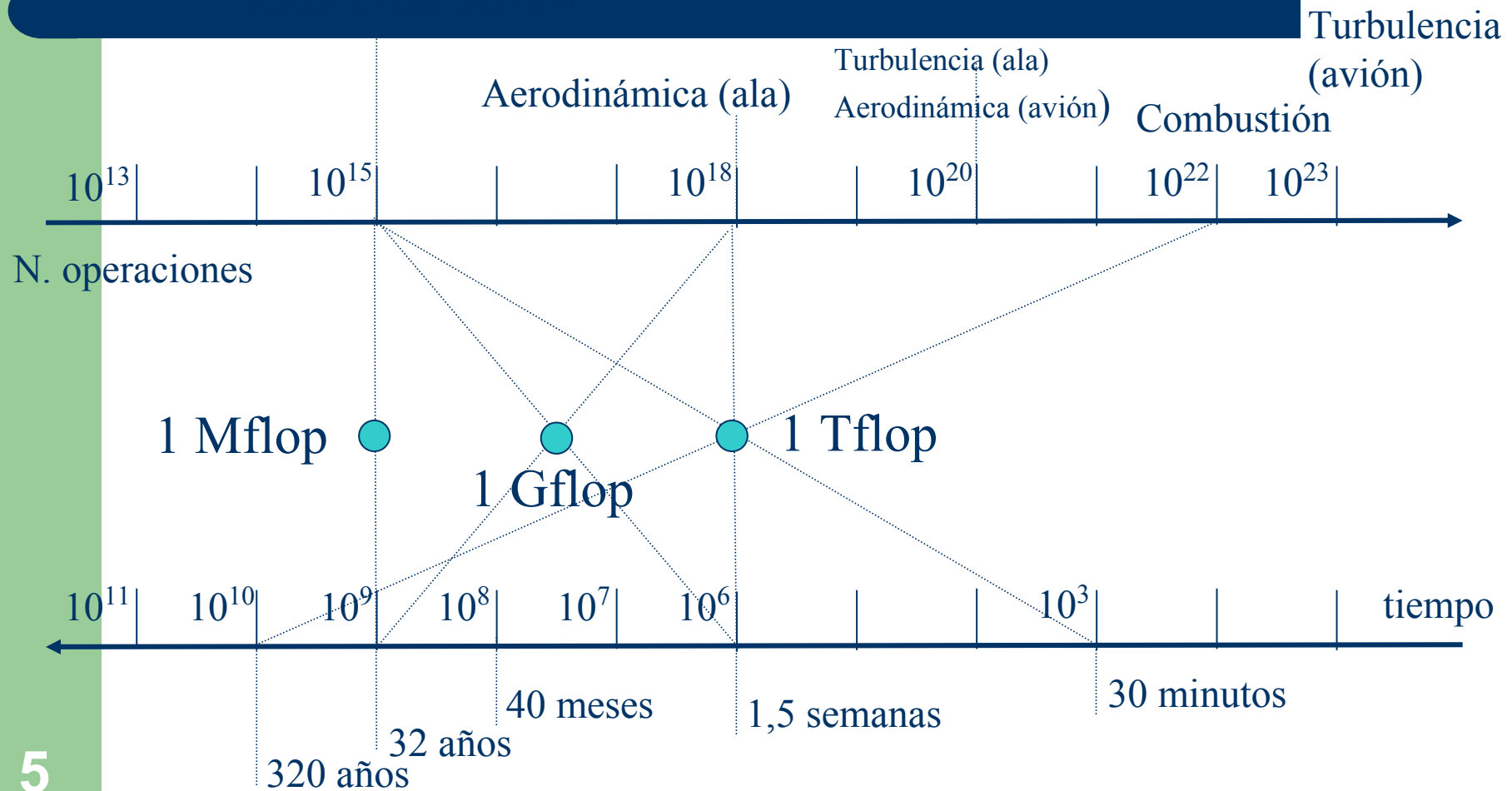
Índice

1. **Introducción.**
2. Clasificación.
3. Ventajas y Desventajas.
4. Diseño.
5. Ejemplos.

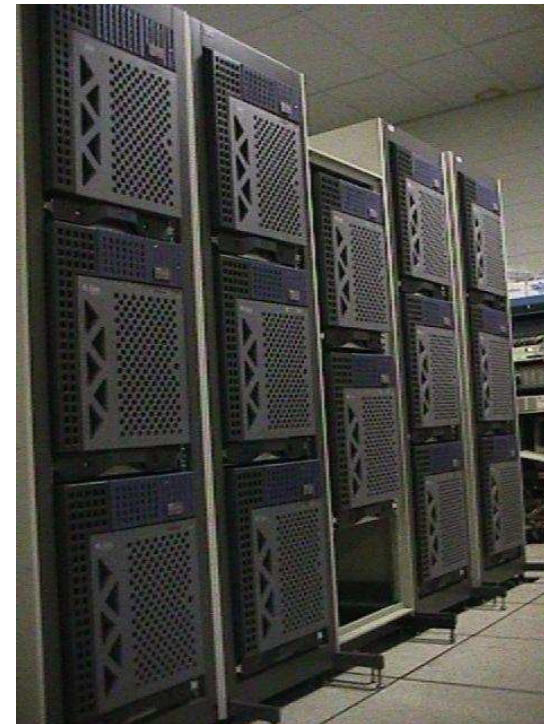
1. Introducción

- Cluster:” Tipo de computación paralela o de procesamiento distribuido, formado por una colección de computadores individuales interconectados entre si, trabajando conjuntamente en un objetivo unificado de cómputo”

¿Es necesario el paralelismo?



1. Introducción.



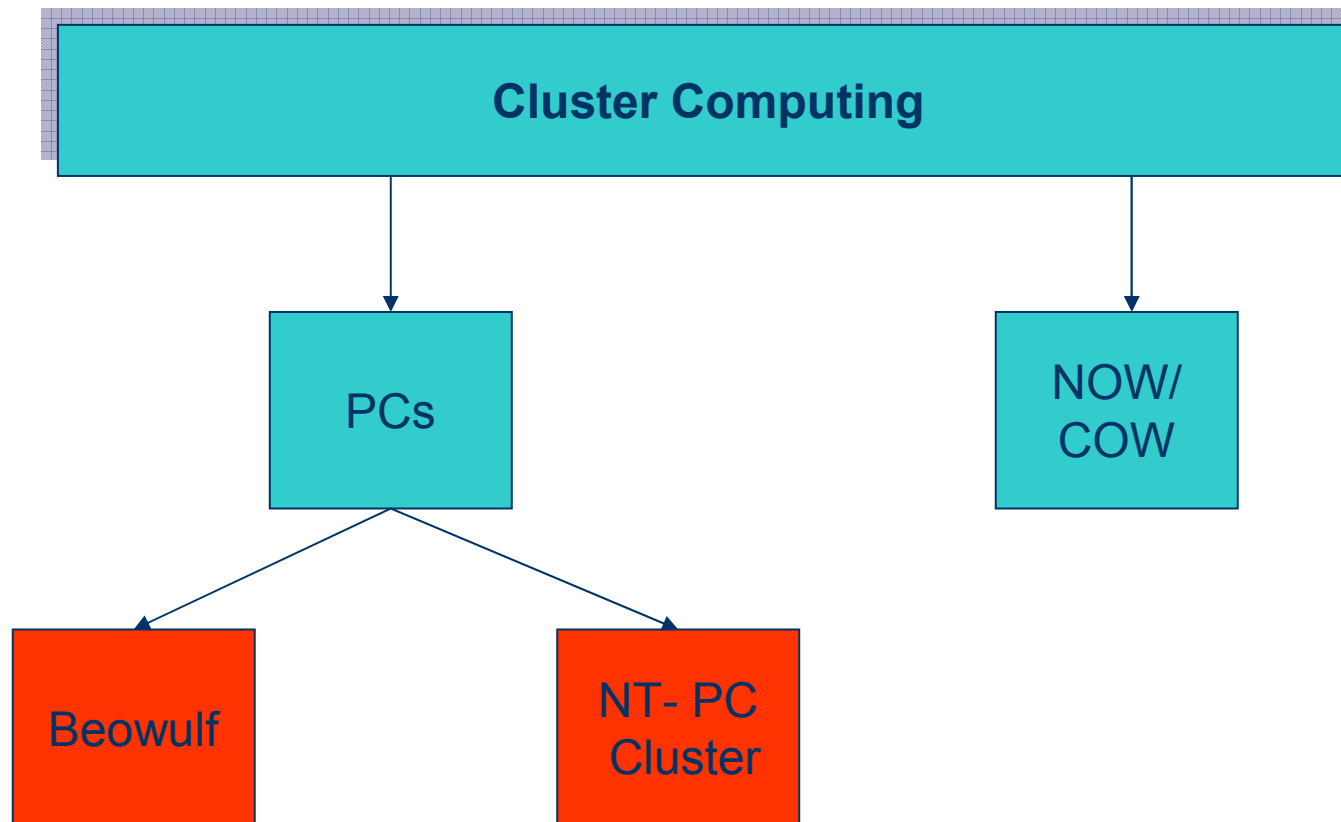
Índice

1. Introducción.
2. **Clasificación.**
3. Ventajas y Desventajas.
4. Diseño.
5. Ejemplos.

2. Clasificación

- Hay muchas formas de clasificar un cluster:
 - Según el objetivo de su aplicación:
 - Alto Rendimiento y Alta Disponibilidad.
 - Según el uso de los nodos:
 - Dedicados y no dedicados.
 - Según el SO de los nodos.
 - Según la configuración de los nodos:
 - Homogéneos y Heterogéneos.

2. Clasificación



2.Clasificación: Beowulf (I)

- Ideados en 1994 por Thomas Sterling y Don Becker.
- Compuesto por 16 Procesadores Intel DX4.
- Conectados por Ethernet.

2.Clasificación: Beowulf (II)

- Cluster en el que cada nodo (PC) no tiene teclado, ratón, tarjeta de video o monitor.
- Buena relación Precio/Rendimiento.
- Hardware de bajo precio.
- Soporte Linux.
- Usan los estándares (PCI, Ethernet,SCSI,...)

2. Clasificación: Beowulf (y III)

- Arquitectura cliente(nodo)/servidor(frontend).
- Los clientes se conectan al servidor.
- Nodos
 - Red local.
- Frontend:
 - Red local & Internet.
 - Separa el trabajo y lo distribuye a los nodos.

2. Clasificación: NOW/COW (I)

- NOW: Networks of Workstations.
- COW: Cluster of Workstations.
- PCs completos conectados por una red.
- Ejemplo de Computación Distribuida.

2. Clasificación: NOW/COW (y II)

- COW, el PC se toma “prestado” cuando el dueño no lo usa.
- NOW, se “cogen” ciclos de los PCs siempre.
- Algoritmos poco tolerantes a problemas de balanceo, es decir todos los ordenadores no tienen la misma carga de trabajo.

Índice

1. Introducción.
2. Clasificación.
3. **Ventajas y Desventajas.**
4. Diseño.
5. Ejemplos.

3. Ventajas.

- Alto rendimiento.
- Alta disponibilidad, fracción o porcentaje de tiempo de vida durante el cual no está inhabilitado a causa de fallos
- Gran capacidad de Expansión y de Escalabilidad, se puede ampliar un cluster con poco dinero.
- Tecnología, los clusters pueden llegar a los 70,78 TFLOPS (BlueGene/L)

3. Inconvenientes.

- El software disponible es limitado.
- La Red puede hacer perder la fiabilidad. (Pérdida de mensajes,...)
- Problemas de seguridad. (Hay muchos puntos que vigilar, en un computador centralizado solo hay UN punto)

Índice

1. Introducción.
2. Clasificación.
3. Ventajas e Inconvenientes.
4. **Diseño.**
5. Ejemplos.

4. Diseño

1. Sistemas Operativos.
2. Middle Ware.
3. Coste.

4.Diseño: Sistemas Operativos.

- La elección del SO es fundamental a la hora de construir un Cluster.
- Es necesario que su Kernel soporte:
 - Programación paralela, el compilador determina en tiempo de compilación que partes
 - Paralelización Inherente;
 - Equilibrio de carga, todos los computadores tienen que tener el mismo trabajo.
- Además tiene que ser tolerante a fallos y que provea soluciones a estos.

4.Diseño: Sistemas Operativos.

- Los clusters se suelen montar principalmente con estos Sistemas Operativos o Partes de ellos.
 - OSCAR, añade a un SO linux un software específico para clusters sobre todo en el paso de mensajes.
 - OPENMOSIX, parchea el núcleo.
 - Rocks, empezó con RH7.2 y actualmente usa RH Enterprise. Tiene software específico para el mantenimiento y la actualización de nodos.
 - Windows 2000 Clustering.

4.Diseño: Middleware

- Es la diferencia entre una red estándar y una solución para el procesamiento paralelo.
- Interfaz entre las aplicaciones y el Hardware del cluster y el SO.
- Es una capa de Software que se añade en lo alto del SO para proveerlo del SSI (Single System Image)
- Proporciona un acceso uniforme a todos los nodos sin tener en cuenta el SO.
- La función principal del Middleware para el paso de mensajes es asegurar la integridad de estos.

4.Diseño: Middleware

- Sus objetivos son:
 1. Único Punto de Ingreso.
 2. Único árbol de directorios.
 3. Único punto de control.
 4. Única red Virtual.
 5. Único espacio de memoria.
 6. Único administrador de tareas.

4. Diseño: Middleware

- 7. Única Interfaz de Usuario.
- 8. Espacio Único de E/S.
- 9. Espacio Único de procesamiento.
- 10. Checkpointing.
- 11. Migración de Procesos

4. Diseño: Middleware

- Las funciones más importantes del Middleware son:
 - Planificador (Sheduler) -> MAUI
 - Gestor de Colas ->OPENPBS
- Ayudan a ver la disponibilidad de nodos y la gestión de tareas.

4. Diseño: Coste.

- Es más caro construir un supercomputador que un cluster.
- La memoria es uno de los componentes que aumentan mucho de precio cuando se aumenta la cantidad.

Índice

1. Multiprocesamiento.
2. Introducción.
3. Clasificación.
4. Diseño.
5. Ejemplos.

5. Ejemplos

1. Google.
2. Seti.
3. QCyCAR.
4. MareNostrum

5. Ejemplos: Google™

- Es el navegador que más ha crecido en estos últimos 10 años.
- Se basa en un *Rank* para mostrar los resultados.
- Atiende más de 5000 millones de búsquedas al mes. (2000 por segundo)

5. Ejemplos: Google (II)

- Formador por un clúster de 20.000 servidores repartidos en 7 *data-centers*.
- Servidor:
 - 1 Procesador INTEL
 - 256-1GB RAM
 - 2 HD: IBM de 40-75GB
 - Linux RedHat.

5. Ejemplos: Google (III)

- Cada data-center se conecta a Internet a través de redes OC48 (2488 Mbps).
- 40-80 PC se conectan a 2 Switch por GigabitEthernet. (Para compactar el espacio se usa un rack) y cada rack se conecta a uno o varios switches centrales.

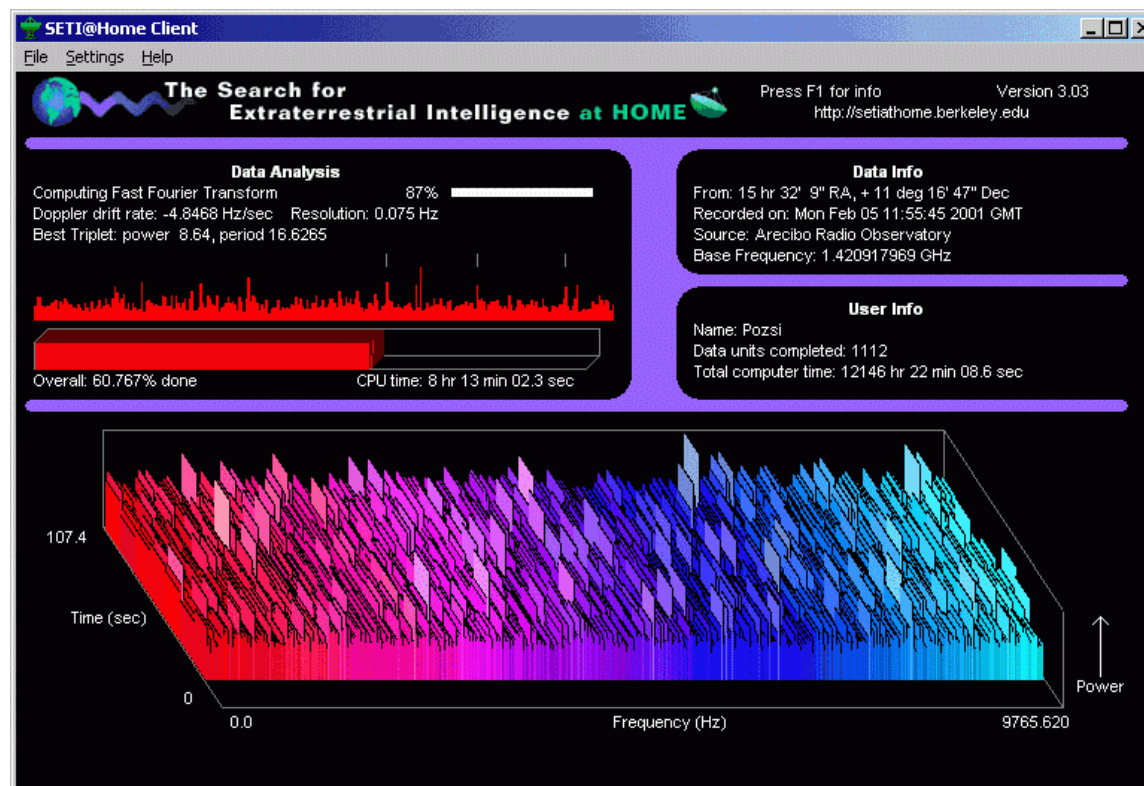
5. Ejemplos: Google (y IV)



5. Ejemplos: Seti@home

- Desarrollado por la Universidad de Berkeley.
- Analiza señales de radio extraterrestres.
- Eso sencillo mediante un Salvapantallas.
- Tecnología Grid.

5. Ejemplos: Seti@home



5. Ejemplos: QCyCAR

- Grupo de Investigación de la ESI.
- Realiza cálculos de hipersuperficies de energía potencial molecular en moléculas de interés astrofísico.



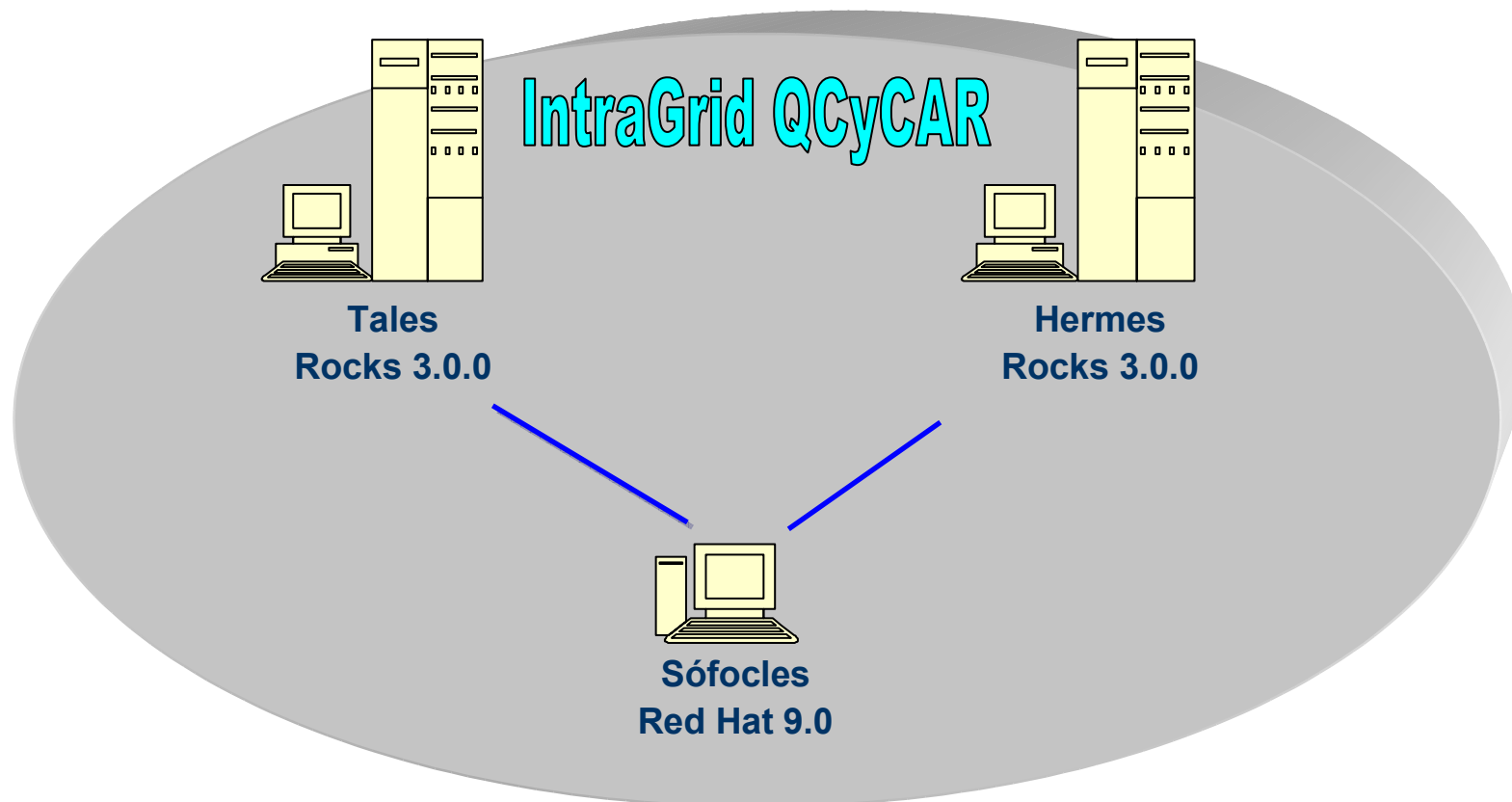
www.rockclusters.org

- Clusters QCyCAR
 - Explotación
 - Hermes (12 nodos)
 - Tales (12 nodos)
 - Desarrollo
 - Diógenes (3 nodos)



Nuestro Grid

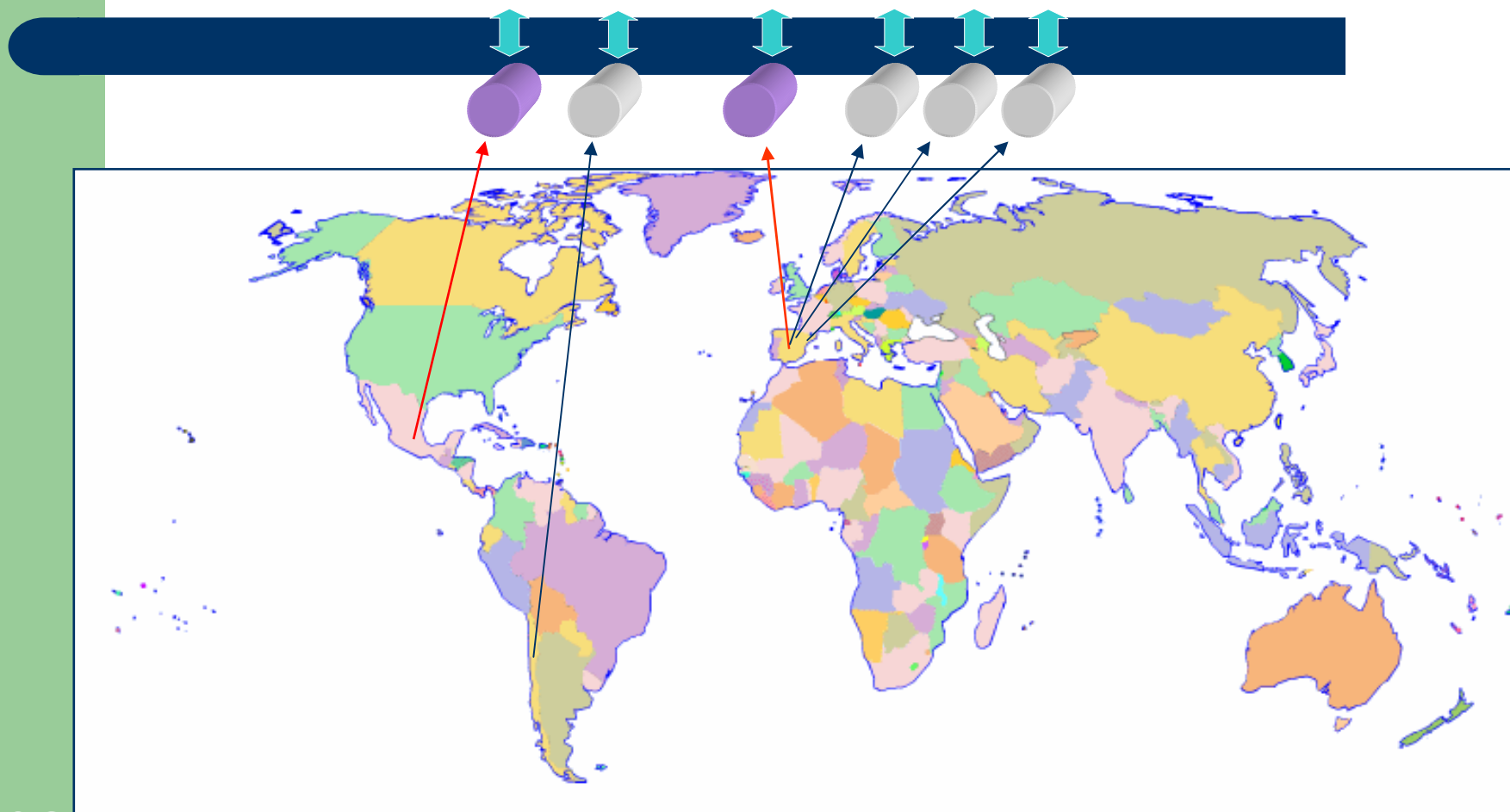
- Topología actual del grid del Grupo QCyCAR



Grupo virtual de Química Computacional



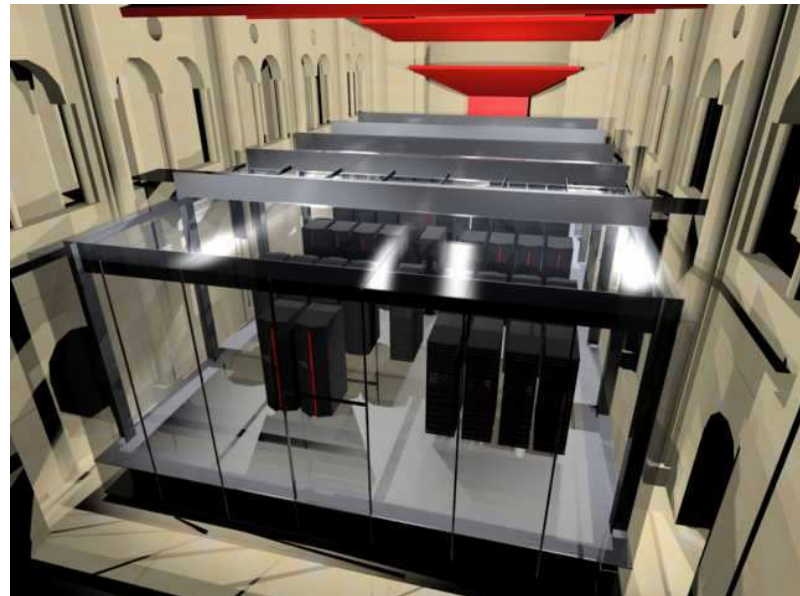
Infraestructura grid



5. Ejemplos: MareNostrum

- Linux Cluster de JS20 blade centers:

- ✓ 40.1 TFlops pico
- ✓ 9 TBytes RAM
- ✓ 130 TBytes disco en Servidores
- ✓ 90 TBytes disco duro.



5. Ejemplos: MareNostrum



- 1 rack (42U) tiene:
 - 6 chassis
 - 168 procesadores
 - 336 GBytes RAM

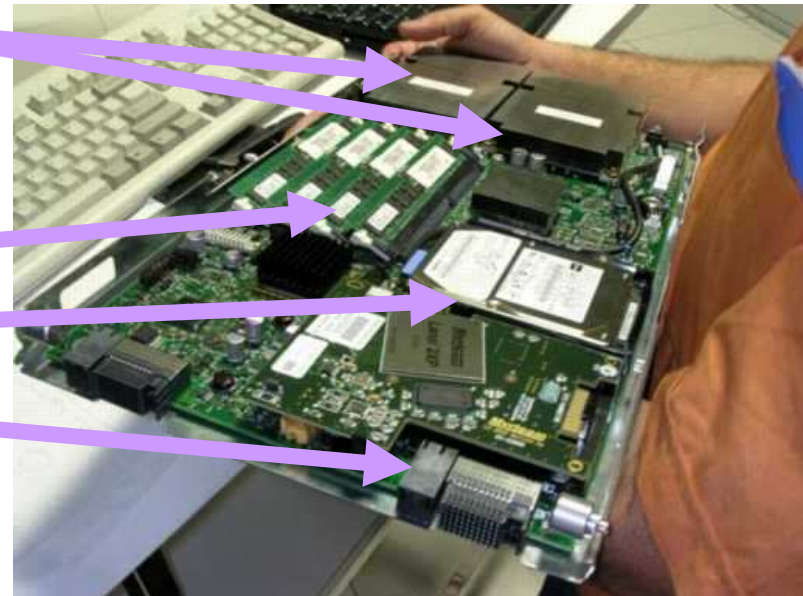
5. Ejemplos: MareNostrum



- 1 Blade center tiene:
 - 14 JS20 blades:
 - 28 procesadores
 - 56 GBytes de memoria
 - 1 switch Gigabit ethernet

5. Ejemplos: MareNostrum

- 1 JS20 blade tiene:
 - 2 PowerPC 970 FX de memoria compartida a 2.2 GHz
 - 4GB ECC 333 DDR
 - 40 Gbytes IDE
 - 3 redes
 - Myrinet
 - Gigabit Ethernet
 - 10/00 Ethernet



BIBLIOGRAFÍA RECOMENDADA

- Morrison, R. *Cluster Computing*. 2003 GNU
- Hennessy & Patterson. *Computer Architecture. A Quantitative Approach*. 2002 Morgan Kaufmann. (Apartados 8.10, 8.11 y 8.12)
- Bookman, C. *Clustering con Linux*. 2003 Prentice Hall
- www.top500.org