Computación Distribuida
Parte I: Computación en clusters

Juan Ángel Lorenzo del Castillo
Grupo de Arquitectura de Computadores
Departamento de Electrónica y Computación
Universidad de Santiago de Compostela

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

CESGA

Índice

- Introducción a las arquitecturas cluster
- Componentes básicos
- Aplicaciones de las arquitecturas cluster

Índice

- Introducción a las arquitecturas cluster
  - ¿Qué es un cluster?
  - Evolución histórica de las arquitecturas de computadores

¿Qué es un cluster?

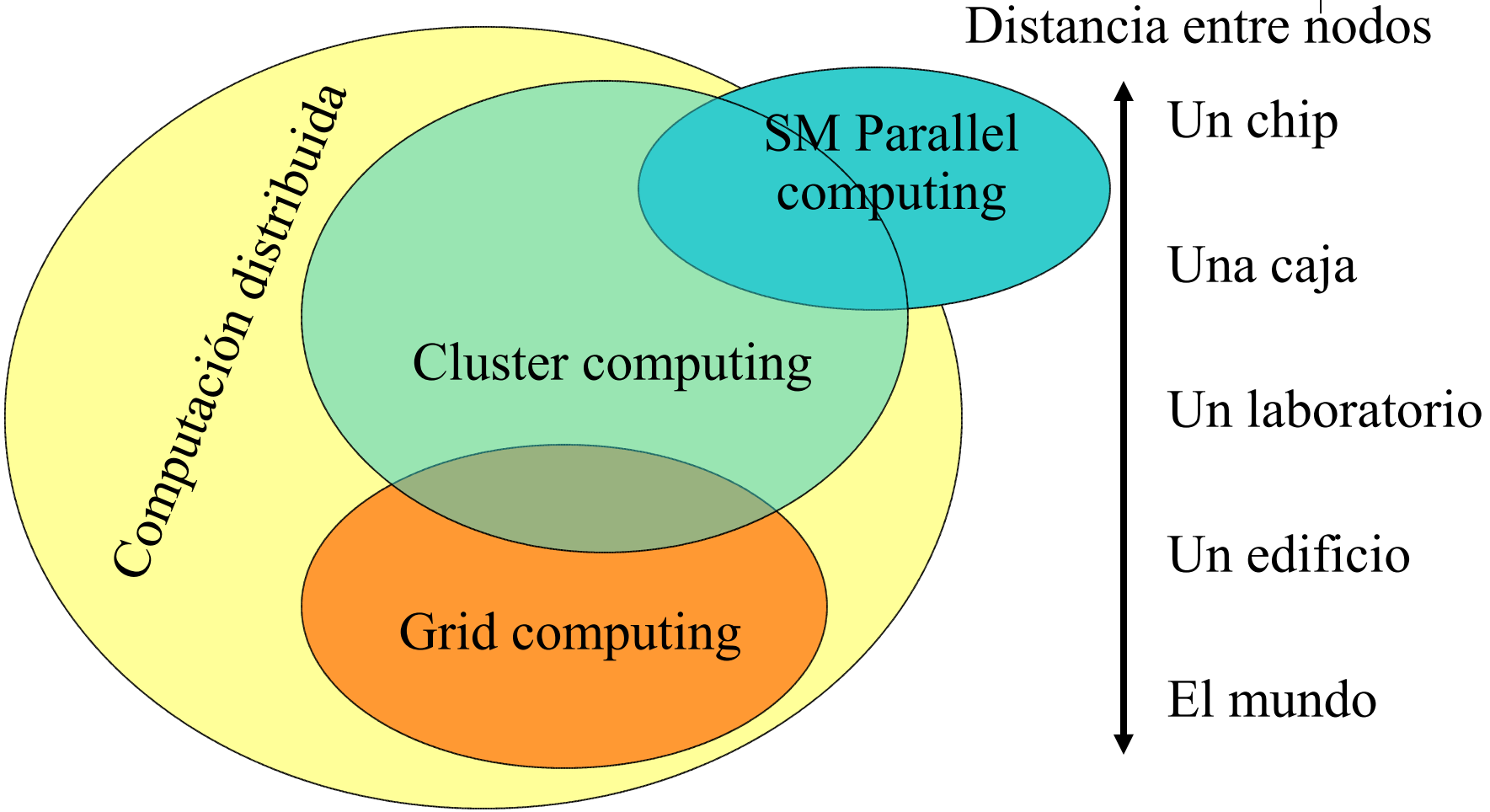
- Un cluster es ...
  - Un conjunto de nodos interconectados mediante una red de interconexión a los que un determinado software convierte en un sistema de mayores prestaciones

- Aunque esta definición peque de general o imprecisa explica en esencia lo que es un cluster

¿Qué es un cluster?

- Objetivos del Software de Sistema:
  - Unir el hardware
  - Proporcionar una visión unificada del sistema (SSI, Single-System Image)
  - Proporcionar alto rendimiento
  - Proporcionar escalabilidad
  - Proporcionar robustez
- Las redes de interconexión proporcionan:
  - Alto ancho de banda
  - Baja latencia
  - Fiabilidad
  - Escalabilidad
- La red de interconexión estará dedicada a la integración del cluster y estará separada del mundo exterior

¿Qué es un cluster?



¿Qué es un cluster?

- Debido a la proximidad entre nodos de un cluster no resulta muy compleja su identificación “de un vistazo” (e.g.)

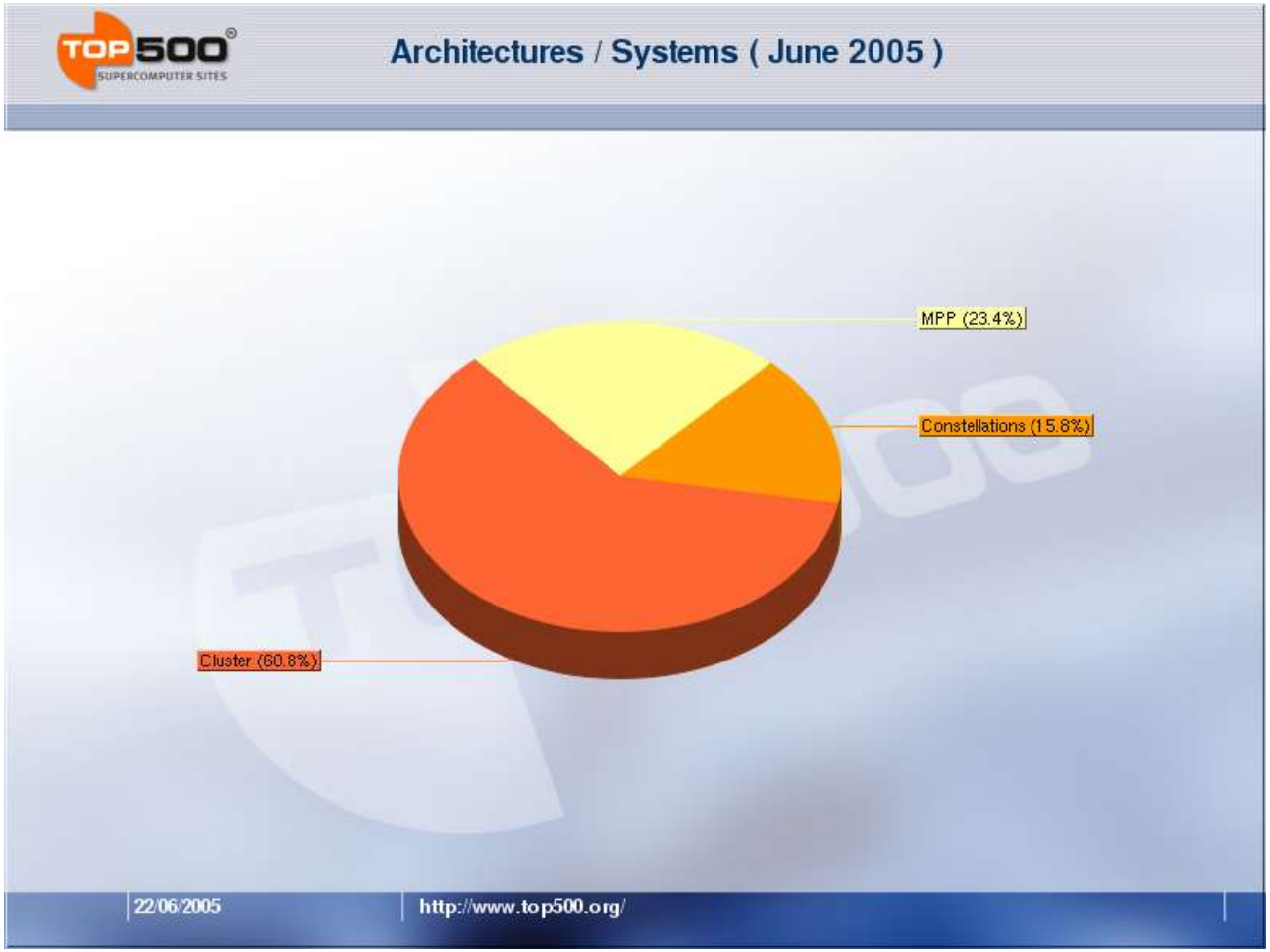
¿Qué es un cluster?

- Las arquitecturas cluster son una familia numerosa y heterogénea dentro de los MIMD (*Multiple Instruction, Multiple Data*) de memoria físicamente distribuida:
- Clasificación:
  - Aplicación objetivo:
    - Alto rendimiento
    - Alta productividad
    - Alta disponibilidad
  - Propiedad de los nodos:
    - Cluster dedicado
    - Cluster no dedicado

¿Qué es un cluster?

- Clasificación (cont.):
  - Tipo de nodos:
    - COWs (Cluster of Workstations)
    - PoPs (Pile of PCs)
    - CLUMPs (Cluster of SMPs)
  - Sistema operativo de los nodos:
    - Linux (Beowulf)
    - Solaris
    - Windows NT ....
  - Configuración de los nodos:
    - Homogéneos
    - Heterogéneos

Evolución histórica

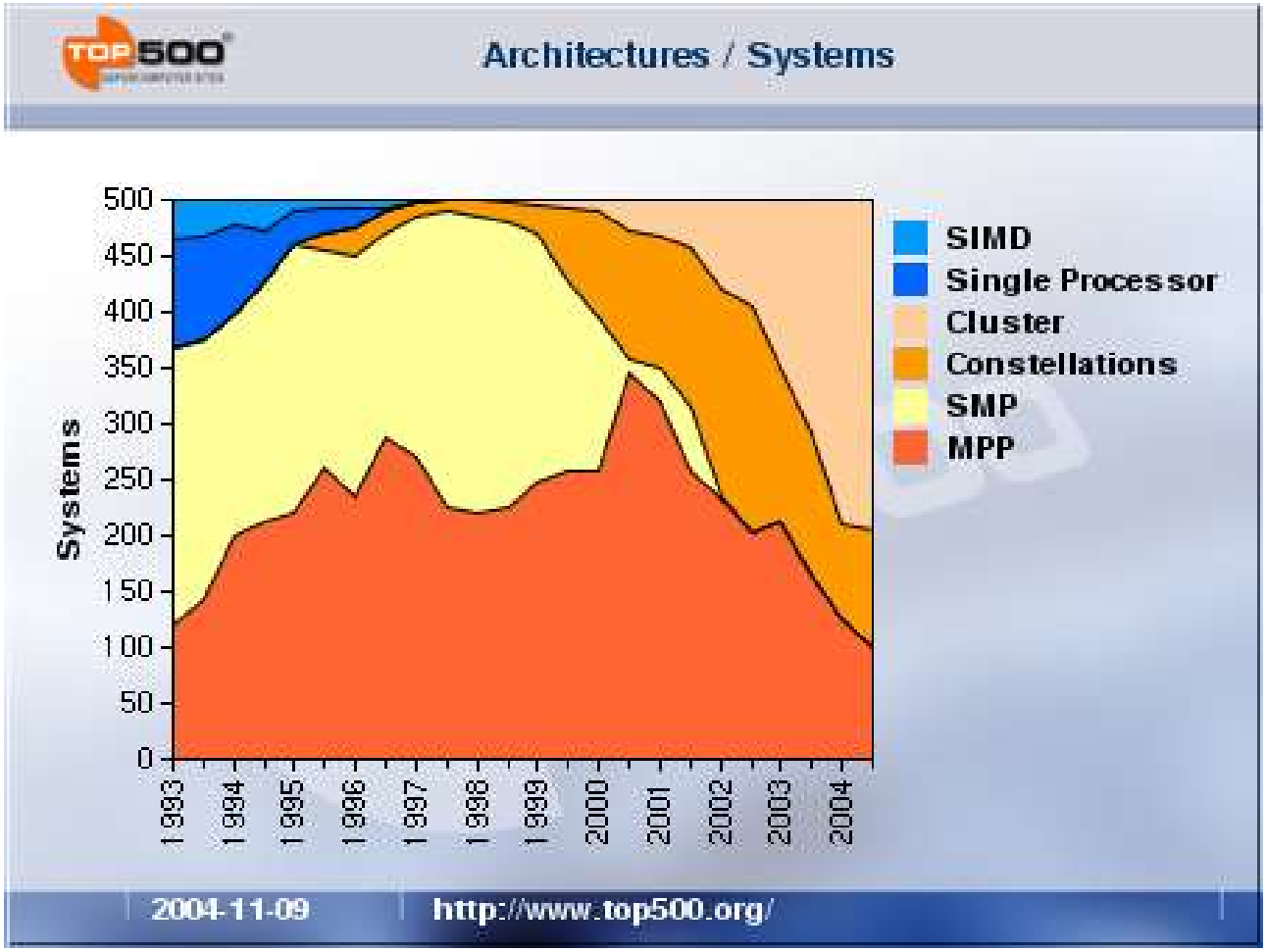


Evolución histórica

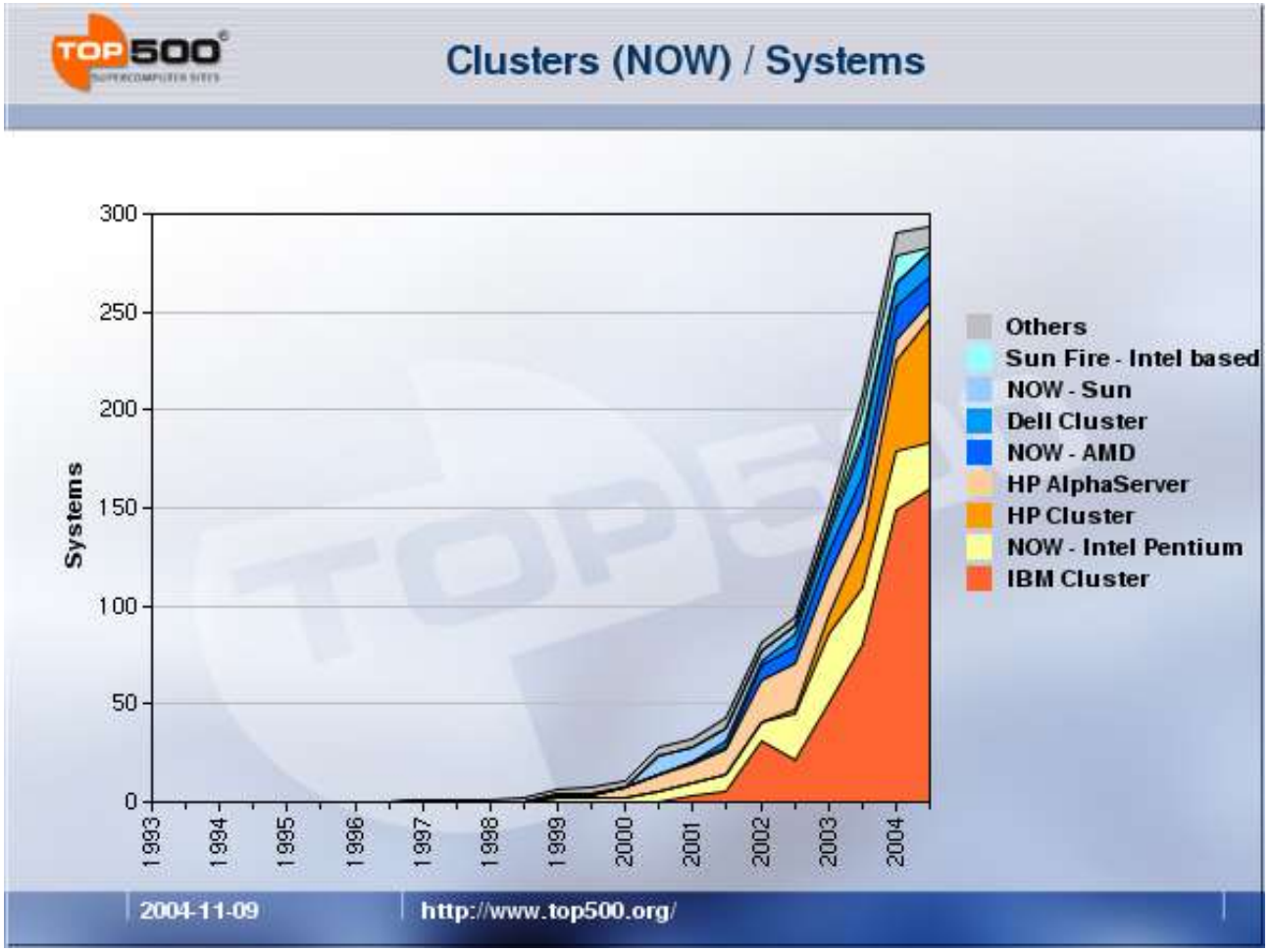
nombre	arquit.	fabric.	# PEs	rend.	año	país
BlueGene L	MPP	IBM	32768	70720	2004	EEUU
Columbia	MPP	SGI	10160	51870	2004	EEUU
Earth-simulator	MPP	NEC	5120	35860	2002	Japón
MareNostrum	cluster	IBM	3564	20530	2004	España
Thunder	cluster	CDC	4096	19940	2004	EEUU
ASCI Q	cluster	HP	8192	13880	2002	EEUU
System X	cluster	Self-made	2200	12250	2004	EEUU
BlueGene LDD1	MPP	IBM	8192	11680	2004	EEUU
eServer pSeries 655	MPP	IBM	2176	10310	2004	EEUU
Tungsten	cluster	Dell	2500	9819	2003	EEUU

Fuente: Top 500 (www.top500.org). 2004

Evolución histórica



Evolución histórica



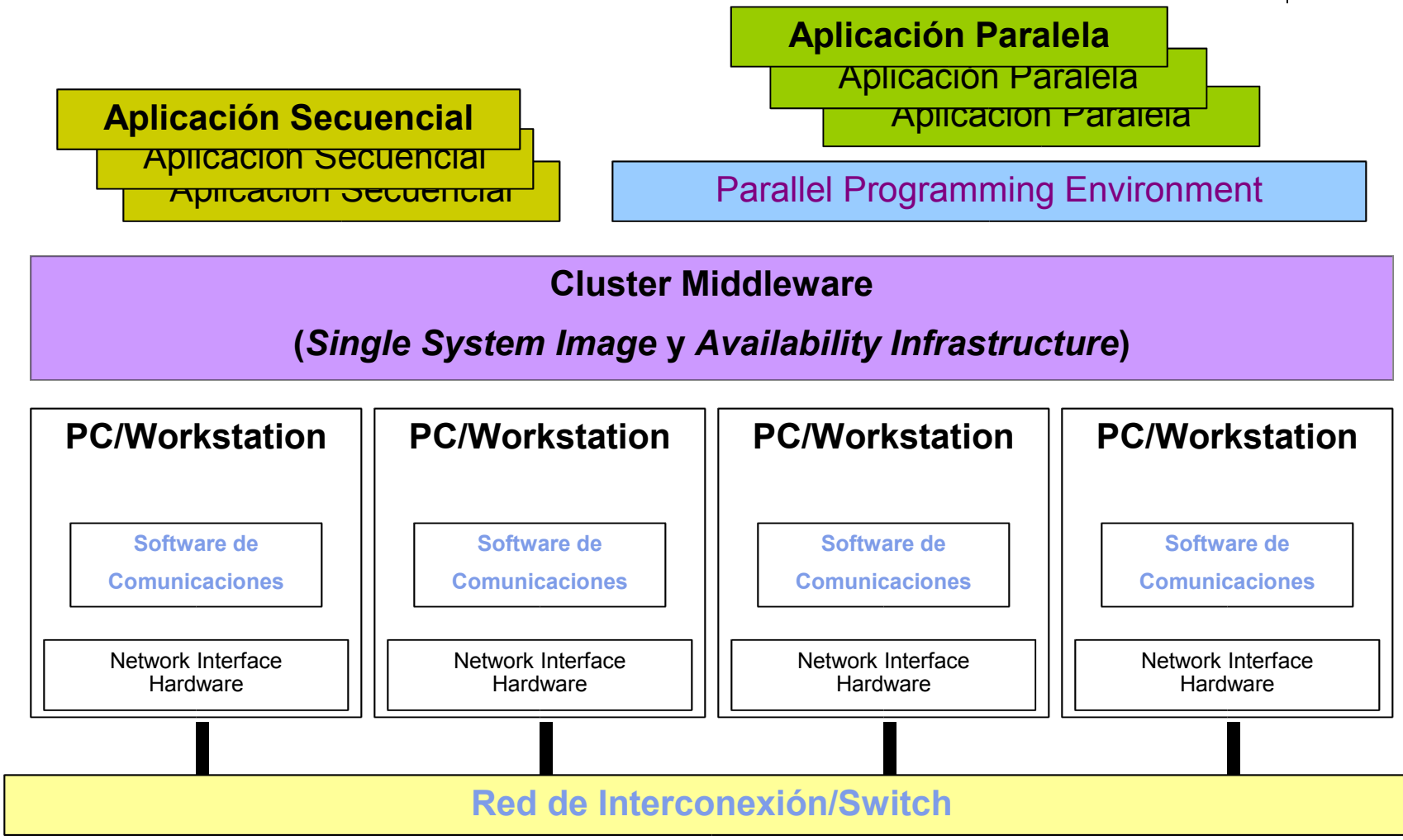
Evolución histórica

- Claves de su éxito:
  - Flexibilidad de configuración
  - Fácil incorporación de mejoras tecnológicas
  - Altamente escalable
- Inconvenientes:
  - Comunicaciones más costosas que en un supercomputador convencional
  - Administración del sistema más compleja
    $\Rightarrow$ software específico

Índice

- Componentes básicos
  - Arquitectura de un cluster
  - Nodos
  - Sistemas operativos
  - Redes de interconexión
  - Protocolos de comunicación
  - Middleware
    - Single System Image (SSI)
    - System Availability (SA)
  - Software
    - Software de gestión de recursos
    - Herramientas para la programación paralela

Arquitectura de un cluster



Nodos

- Múltiples computadores (PCs, WS o SMP)
- Basados en diferentes arquitecturas (CISC/RISC/VLIW/Superscalares)
  - Intel: Pentiums, Xeon, Itanium
  - Sun: SPARC, ULTRASPARC
  - IBM: RS6000/PowerPC
  - SGI: MIPS
  - Digital: Alphas

Nodos

- Los procesadores más usuales en clusters *self-made* son:
  - Procesadores Intel.- Itanium*, Itanium 2*, Xeon, Xeon MP, Xeon HT, P4, P4 Prescott, Celeron
  - Procesadores AMD.- Opteron*, Athlon FX*, Athlon, AMD XP, Duron, Athlon 64*
  - Procesadores Apple.- G4, G5*

*Procesadores de 64 bits

Nodos

- Medidas de rendimiento:
  - MIPS (millones de instrucciones por segundo)
  - MFLOPS (millones de operaciones en punto flotante por segundo)
- Programas de evaluación (benchmarks)
  - Los más populares son los SPEC (*Standard Performance Evaluation Corporation) http://www.spec.org*

Sistemas operativos

- Linux, Solaris, Windows NT, ...
- El SO por excelencia en los clusters es Linux:
  - Es gratuito. Disminuye el *Total Cost of Ownership* (TCO)
  - Existe una gran cantidad de software y una gran comunidad de desarrolladores y usuarios
  - Es tan seguro y fiable como un Unix y presenta numerosas facilidades de administración

Sistemas operativos

- El SO Linux suele ser instalado con distribuciones, que son colecciones de software con el sistema base, programa de instalación y numerosas aplicaciones. Destacan:
  - **Red Hat:** Distribución muy popular. No tiene nuevas versiones desde RH 9
  - **Fedora:** Es la sustituta de RH especializada para desarrollo
  - **Linux Enterprise Server:** Sustituye a RH especializándose en sistemas empresariales y en clusters
  - **Slackware:** Distribución basada en RH, para computadores de escritorio
  - **SUSE:** Distribución alemana con una importante cantidad de software
  - **Debian:** Distribución para usuarios avanzados y administradores. Es la más segura. Existe una gran comunidad de desarrolladores Debian en Galicia.
  - **Mandriva (Mandrake + Conectiva):** Distribución brasileña orientada a la empresa.
  - **Linex:** Distribución de la Junta de Extremadura basada en Debian
  - **Ubuntu:** Distribución basada en Debian, más sencilla de utilizar y con actualizaciones más frecuentes.

Redes de interconexión

- Se han llevado a cabo una enorme cantidad de esfuerzos para mejorar sus características, especialmente en cuanto a:
  - Nivel físico
  - Nivel de enlace
  - Enrutado
  - Conmutación de envíos
  - Detección y corrección de errores
  - Operaciones colectivas

Redes de interconexión

- Medidas de rendimiento:
  - **Latencia:** intervalo de tiempo entre que se inicia el envío y los datos empiezan a estar disponibles en el destino (tiempo de envío de un mensaje vacío)
  - **Ancho de Banda efectivo (P):** También llamado Productividad. Mide el volumen de tráfico que puede ser transferido entre dos nodos por unidad de tiempo
  - **Ancho de Banda asintótico (Bw):** velocidad a la que se transmiten los datos, una vez iniciada la transmisión para un número de datos muy elevado. Idealmente, infinito. Equivale a la Productividad máxima. Se mide en bits por segundo

Redes de interconexión

- Redes más usuales:
  - Ethernet
  - Fast-Ethernet
  - Gigabit-Ethernet
  - Myrinet
  - SCI
  - Infiniband

Redes de interconexión

- Ethernet
  - Son las redes más utilizadas en la actualidad, debido a su relativo bajo coste.
  - Su tecnología limita el tamaño de paquete, realizan excesivas comprobaciones de error y sus protocolos no son eficientes.
  - Para aplicaciones con un patrón de comunicaciones casi inexistente pueden suponer una solución acertada

Redes de interconexión

- Myrinet y Myrinet 2000
  - Es una red de baja latencia utilizada en la actualidad tanto en clusters como en MPPs
  - Tiene dos bibliotecas de comunicación a bajo nivel (GM y MX) disponibles de forma gratuíta
  - Sobre esas bibliotecas está implementado de forma eficiente otras interfaces software como MPI, Sockets o TCP/IP

Redes de interconexión

- SCI (Scalable Coherent Interface)
  - Es una red de extremadamente baja latencia, que presenta ventajas frente a Myrinet en clusters de pequeño tamaño al tener una topología punto a punto y no ser necesaria la adquisición de un conmutador
  - El software sobre SCI está menos desarrollado que sobre Myrinet, pero los rendimientos obtenidos son superiores, destacando SCI Sockets (que obtiene startups de 2 microsegundos) y ScaMPI, una biblioteca MPI de elevadas prestaciones

Redes de interconexión

- Infiniband
  - Es una red reciente surgida de un estándar desarrollado específicamente para realizar la comunicación en clusters
  - En vez de enviar datos en paralelo los envía en serie y puede manejar múltiples canales de datos a la vez en una señal multiplexada
  - Mediante la agregación de canales permite obtener anchos de banda muy elevados (del orden de los Gigabytes/s)
  - Los productos para IB están apareciendo en la actualidad en el mercado.

Redes de interconexión

- Comparativa:

	Ancho de Banda	Latencia	Tarjeta	Switch
Fast-Ethernet	100Mbits/s	50us	10 €	10 €
Gb-Ethernet	1Gbit/s	70us	50 €	30 €
10Gb-Ethernet	10Gbit/s	100us		
SCI	1.33Gbit/s (full duplex)	2us	1.000 €	0 €
Myrinet	2Gbit/s (full duplex)	7us	800 €	1.000 €
IB	2Gbit/s / canal	10us	1.500 €	2.000 €

Protocolos de comunicación

- Tradicionales (pesados):
  - TCP y UDP
- Diseño específico (ligeros):
  - Active Messages
  - Fast Messages
  - VMMC
  - BIP
  - VIA

Middleware

- Single System Image (SSI)
  - Hace ver al cluster como una única máquina
  - Se puede construir a nivel hardware, a nivel operativo/middleware o a nivel de aplicación
- System Availability (SA) Infrastructure
  - Servicios de tolerancia a fallos

Single System Image

- Servicios ofrecidos por un cluster SSI:
  - punto de entrada único
  - jerarquía de ficheros única
  - espacio de E/S único
  - punto de control y gestión único
  - espacio de memoria único
  - sistema de gestión de trabajos único
  - interfaz de usuario única
  - espacio de procesos único
- Pueden ocurrir que algunos estén disponibles y otros no

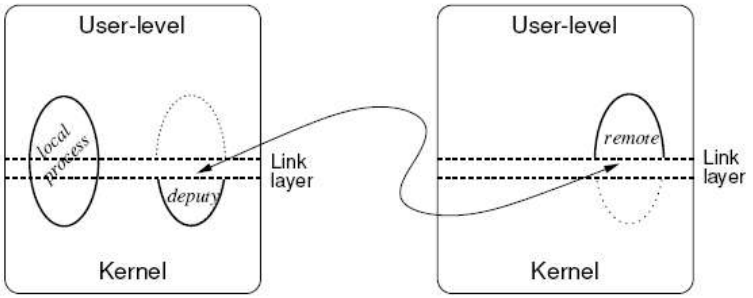
Single System Image

- Beneficios:
  - Simplifica la ejecución de aplicaciones por parte del usuario
  - Simplifica la gestión del sistema
- Se puede ofrecer por
  - hardware:
    - Ofrece una visión unificada al S.O.
  - a nivel S.O./middleware
    - Ofrece una visión unificada a las aplicaciones
    - SCO UnixWare, Sun Solaris MC, GLUnix, MOSIX
  - a nivel de aplicación:
    - Ofrece una visión unificada al usuario

Single System Image

- MOSIX (Multicomputer Operating System for UnIX)
  - Paquete software gratuito que extiende el kernel de Linux
  - Hace que un cluster linux de PCs se vea como un único computador paralelo de alto rendimiento. Monitoriza el sistema y, si es necesario, balancea la carga entre los nodos a través de la migración de procesos de forma transparente al usuario

  - El proceso migrado es dividido en dos contextos:
    - Contexto del sistema (deputy) el cual se queda en el nodo origen
    - Contexto del proceso (remote) el cual se migra



  - Se establece un enlace de comunicación entre ambos de tal forma que el proceso siempre puede acceder a su entorno local via el deputy

System Availability

- Objetivo: un fallo en el sistema debería ser recuperable sin afectar a la aplicaciones de los usuarios
- Mecanismos para conseguirlo:
  - Checkpointing
  - Tecnologías de tolerancia a fallos:
    - Mirroring
    - Hot standby
    - Failover
    - Failback

Software

- Software de gestión de recursos
  - Administración/monitorización del sistema
  - Gestión/planificación de trabajos
- Herramientas para la programación paralela
  - librerías
  - debuggers
  - herramientas de análisis de rendimiento

Administración/ monitorización del sistema

- Software de administración/monitorización del sistema:
  - Herramientas de monitorización
  - Herramientas de alarma
  - Comandos del sistema paralelos
- Algunos ejemplos:
  - SCMS
  - PARMON
  - Ganglia

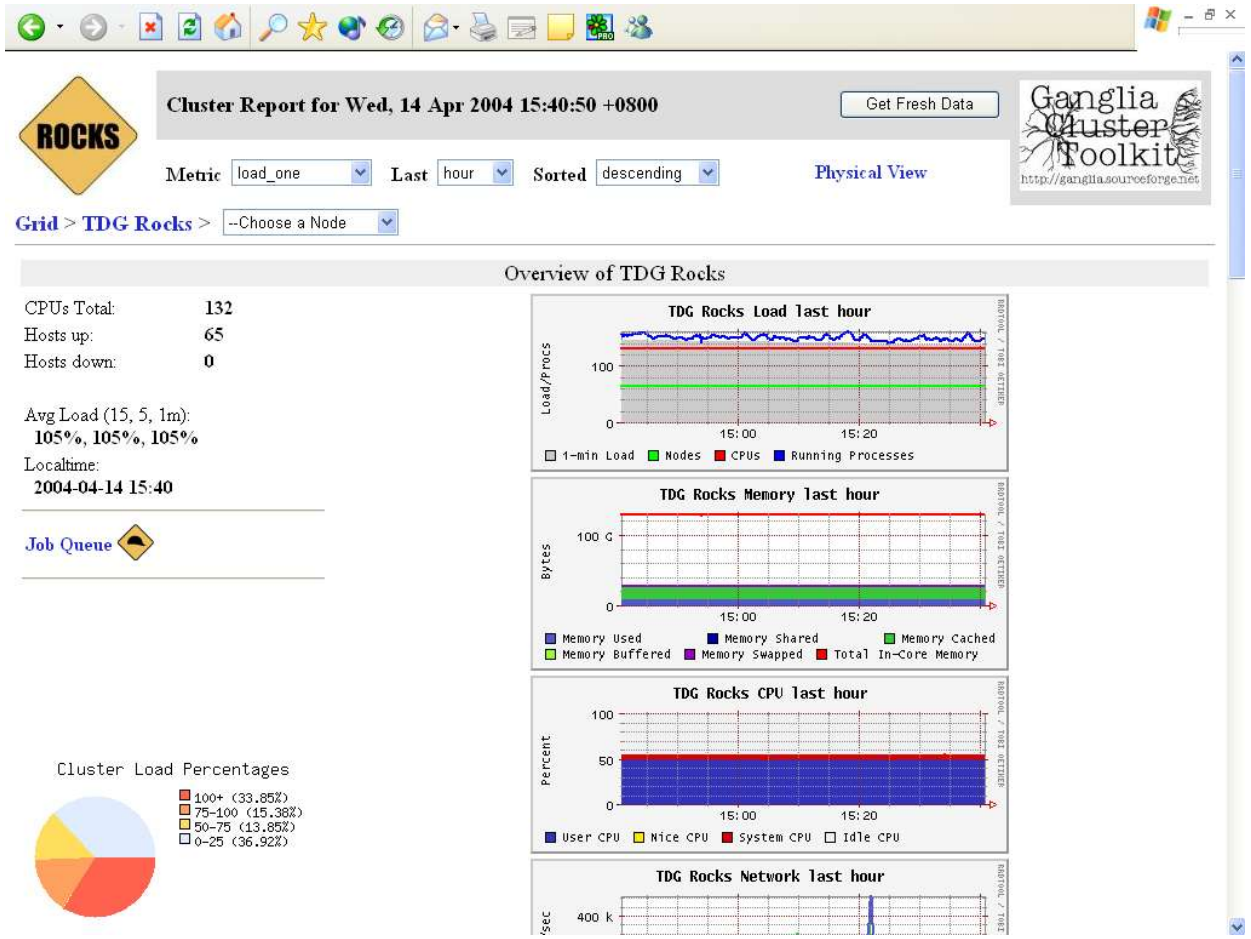
Administración/ monitorización del sistema

SCMS (SMILE Cluster Management Systems)

- Desarrollado por el High Performance Computing and Networking Center de Thailandia
- Conjunto de herramientas de gestión que incluye:
  - Monitorización portable en tiempo real
  - Comandos Unix paralelos
  - Sistema de alarmas
  - Conjunto de herramientas GUI para usuarios y administradores
  - Interfaz Web
- Forma parte de un conjunto de herramientas cluster lla mado openSCE (Scalable Cluster Environment)
- Más información en http://www.opensce.org

Administración/ monitorización del sistema

- Ganglia:



Gestión/planificación de trabajos

- Componente software que determina cómo, cuando y donde ejecutar los trabajos de los usuarios
- En los supercomputadores este servicio es proporcionado por el SO
- En los clusters tenemos 2 opciones:
  - Utilizar un sistema operativo distribuido
    - Mosix, Glunix,...
  - Utilizar facilidades software construidas sobre el SO
    - PBS, Condor, SGE, …

Gestión/planificación de trabajos

- Funciones:
  - Manejar los múltiples recursos disponibles como si se tratase de una única máquina
  - Controlar el buen uso de los recursos:
    - Evita que la memoria sea sobre-comprometida
    - Marca límites en
      - el número de trabajos ejecutados por nodo
      - el número de trabajos ejecutados concurrentemente por el mismo usuario
    - ....

Gestión/planificación de trabajos

- Un sistema de gestión de trabajos ofrecerá todas o un subconjunto de las siguientes características:
  - Soporte para entornos heterogéneos
  - Suporte para trabajos interactivos
  - Soporte para trabajos batch
  - Soporte para trabajos paralelos
  - Checkpointing
  - Migración
  - Balanceo de la carga
  - Límites en el tiempo de ejecución
  - GUI

Gestión/planificación de trabajos

- Componentes y arquitectura:
  - Interfaz de usuario
  - Entorno de administración
  - Objetos gestionados

- Interfaz de usuario
  - Envío de trabajos para ejecución
  - Muestra del estado de los trabajos
  - Borrado de trabajos

Gestión/planificación de trabajos

- Entorno de administración que permita:
  - Especificar las características de las máquinas que componen el entorno
  - Definir clases de trabajos permitidos y las máquinas apropiadas para la ejecución de cada clase
  - Definir los permisos de los usuarios
  - Especificar las limitaciones en el uso de los recursos para usuarios y trabajos
  - Especificar políticas para la asignación de trabajos
  - Controlar y asegurar el funcionamiento correcto del sistema
  - Analizar los datos sobre su uso que permitan reconfigurar de forma óptima el sistema

Gestión/planificación de trabajos

- Objetos gestionados:
  - Colas
  - Nodos
  - Trabajos
    - Batch
    - Interactivos
    - Paralelos
    - Trabajos con checkpointing
  - Recursos
  - Políticas
    - de utilización de recursos
    - de planificación

Gestión/planificación de trabajos

PBS (Portable Batch System)

- Desarrollado por la NASA. OpenPBS es la versión gratuita para entornos UNIX (www.openpbs.org)
- Permite el enrutamiento de trabajos desde diferentes ordenadores
- Permite definir diferentes políticas sobre la utilización de los recursos distribuidos
- Pensado para entornos dedicados

Gestión/planificación de trabajos

- Otros:
  - Condor (www.cs.wisc.edu/condor):
    - Soporta checkpointing y permite la migración de procesos
    - Indicado para entornos no dedicados
  - SGE (Sun Grid Engine):
    - El usuario expresa a través de un script sus requerimientos y SGE busca el mejor recurso disponible
    - Pensado para entornos dedicados
  - LoadLeveler:
    - Versión de Condor, por IBM (de pago). El usuario especifica los requerimientos y LoadLeveler busca el mejor recurso disponible
    - Soporta checkpoiting y migración de procesos
    - Soporte para entornos no dedicados

Herramientas para la programación paralela

- Las herramientas para la programación paralela convierten a las plataformas cluster en buenas alternativas a los supercomputadores
- Alternativas para la programación paralela:
  - Paso de mensajes
    - Librerías MPI, PVM
  - Basada en memoria compartida:
    - Distributed Shared Memory (DSM) Systems
      - Hardware
      - Software:
        - TreadMarks (http://www.cs.rice.edu/~willy/TreadMarks/overview.html)

Herramientas para la programación paralela

- Herramientas de análisis de rendimiento
  - Objetivo: identificar *cuellos de botella* en las aplicaciones paralelas
  - La mayoría de las herramientas incluyen alguno o todos de los siguientes componentes:
    - Un medio para insertar dentro de la aplicación del usuario llamadas a las rutinas de monitorización de rendimiento
    - Una librería consistente en un conjunto de rutinas de monitorización que miden diferentes aspectos del rendimiento del programa
    - Un conjunto de herramientas para procesar y mostrar la información obtenida

Índice

- Aplicaciones de las arquitecturas cluster
  - Alta productividad
  - Alto rendimiento
  - Alta disponibilidad

Alta productividad

- Objetivo: ejecutar un mayor número de aplicaciones por unidad de tiempo
- Para ello se necesita un gestor de recursos
- **Condor** es un sistema de gestión de tareas cuyo objetivo es conseguir alta productividad

Alto rendimiento

- El objetivo es reducir el tiempo de ejecución de las aplicaciones
- Para ello se utiliza la computación paralela. En la actualidad, MPI es el estándar de facto
- Tradicionalmente, los clusters para alto rendimiento se denominan Beowulf.
  - Nodos dedicados.
  - La red o redes están dedicada exclusivamente al beowulf
  - Los ordenadores y la red son M^2COTS (*Mass Market Commodity-Off-The-Shelf)*
  - Todos los nodos utilizan software *open source.*

Alto rendimiento

- Ejemplos de clusters Beowulf:
  - Scyld Cluster O.S. (http://www.scyld.com)
  - ROCKS (http://www.rocksclusters.org)
  - OSCAR (http://oscar.sourceforge.net)
  - OpenSCE (http://www.opensce.org)
  - DCC: Debian Cluster Components (http://dcc.irb.hr/)

Alto rendimiento

- Librerías Numéricas
  - Todas las librerías disponibles para arquitecturas paralelas serían adecuadas para los clusters
  - ScaLAPACK y PLAPACK (paralelización de LAPACK): resolutores de sistemas de ecs por métodos directos
  - Aztec, Blocksolve, PSPARSLIB: resolutores de sistemas de ecs por métodos iterativos
  - PARPACK y PeIGS: resolutores de problemas de autovalores
  - ...

Alta disponibilidad (HA)

- La disponibilidad de un sistema se define como:

$$disponibilidad = \frac{MTBF}{MTBF + MTTR}$$

MTBF = Mean Time Between Failure (tiempo promedio entre fallos)

MTTR = Maximum Time To Repair (máximo tiempo de reparación)

- Se busca que el cluster esté disponible la mayor cantidad de tiempo posible.
  - Aumentar MTBF es incrementar la fiabilidad (difícil)
  - Reducir MTTR es más habitual, mediante redundancia de software y componentes

Alta disponibilidad (HA)

- La redundancia es la base de la alta disponibilidad
- Se denomina Punto Único de Fallo (*Single Point of Failure*, SPOF) a cualquier elemento no replicado que pueda estar sujeto a fallo
- Para conseguir HA no deben existir SPOFs
- La técnica básica utilizada en clusters para conseguir HA es *failover*
  - *Failover:* Si una parte del sistema falla otra parte del sistema debe retomar el trabajo

Alta disponibilidad (HA)

- Soluciones de alta disponibilidad:
  - Linux-HA Project (http://www.linux-ha.org)
  - VERITAS Cluster Server (http://www.veritas.com)
  - HP's MC/Service Guard
  - Microsoft's Cluster Server (Wolfpack)
  - RedHat HA cluster (http://ha.redhat.com)
  - Turbolinux Cluster Server
    (http://www.turbolinux.com/products/tcs)
  - Linux Virtual Server Project
    (http://www.linuxvirtualserver.org/)

Ejemplos de sistemas cluster

- El cluster Google (sistema de alta disponibilidad):
  - Requerimientos:
    - Dar respuesta a miles de búsquedas por segundo
    - Cada búsqueda
      - lee cientos de MB de datos
      - consume decenas de billones de ciclos de CPU
    - Aplicación con alto grado de paralelismo
      - Diferentes búsquedas pueden realizarse en paralelo
      - Cada búsqueda admite paralelismo particionando el espacio de búsqueda

Ejemplos de sistemas cluster

- Solución hardware:
  - Varios clusters de PCs geográficamente distribuidos
  - Cada cluster de PCs se compone de:
    - Unos pocos miles de nodos
    - Nodos: CPUs de distintas generaciones, desde Intel-Celeron a 533 MHz a Intel-Pentium III dual a 1.4Ghz
    - Cada nodo con gran capacidad de almacenamiento
    - Organizados en racks de 40-80 servidores unidos mediante Fast-ethernet
    - Los racks se unen mediante Gigabit-Ethernet
- Para saber más:
  - Web Search for a Planet: The Google Cluster Architecture. L.A. Barroso, J. Dean and U. Hölzle. IEEE Micro, 23(2): 22-28, 2003.

Ejemplos de sistemas cluster

- PAPIA (PArallel Protein Information Analysis) Cluster (1998) (sistema de alto rendimiento)
  - Requerimientos:
    - Construir una infraestructura computacional para el análisis de moléculas de proteinas y secuencias de ADN
      - Manejo de enormes bases de datos
      - Aplicaciones con alto grado de paralelismo

Ejemplos de sistemas cluster

- Solución hardware:
  - Cluster compuesto de 64 nodos
  - Cada nodo consta de:
    - Procesador dual Intel Pentium Pro 200Mz
    - 256 MB de RAM
    - 4.1 GB de disco
  - Red de interconexión: Myrinet + fast ethernet
  - SO: NetBSD
  - MPICH-PM para ejecución MPI de alto rendimiento

Más Información

- R.Brown, "*Engineering a Beowulf-style computer cluster*", http://www.phy.duke.edu/brahma/beowulf_online_book/beowulf_book.html, 2002.
- R.Buyya, *Cluster Computing Info Centre*, http://www.buyya.com/cluster, última visita, octubre de 2005.
- CESGA, www.cesga.es, última visita, octubre de 2005.
- Lista Top 500, http://www.top500.org, última visita, octubre de 2005.
- M.J.Martín Santamaría, *Apuntes de Arquitecturas Distribuidas*. Ingeniería Informática, Universidade da Coruña, 2004.
- Hispacluster, http://www.hispacluster.org, última visita, octubre de 2005.