



FA  
UNIVER  
AD DE INFORMÁTICA  
D NACIONAL DE LA PLATA

# L oratorio de Procesamiento Paralelo MultiCluster accessible vía WEB

*Tesina de Licenciatura en Sistemas*

**Autor:**

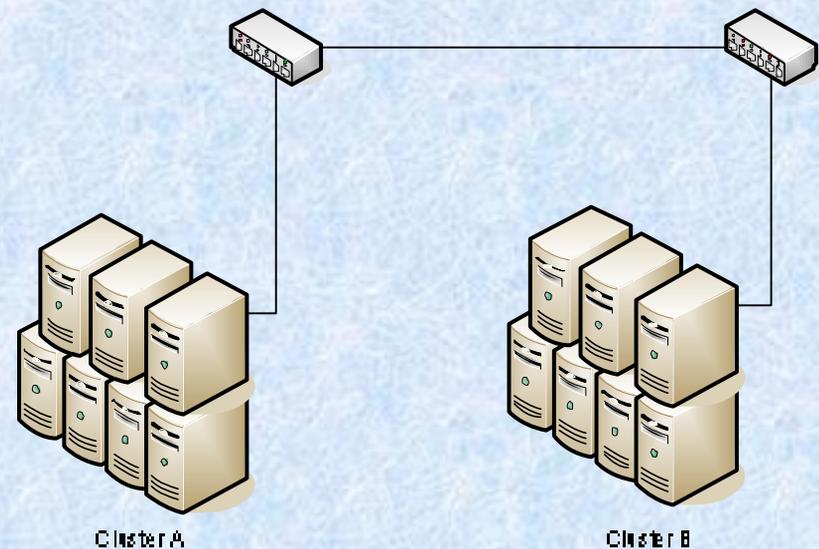
**Adrián Pousa**

**Director: Armando E. De Giusti**

**Co-Director: Marcelo Naiouf**

# Objetivo

Desarrollar una aplicación WEB que permita la ejecución de aplicaciones en una arquitectura multicluster heterogénea (en principio los clusters existentes en la Facultad) así como la administración de dicha arquitectura.



# Motivación

El uso de redes de computadoras como máquinas paralelas virtuales se ha generalizado. Clusters, multiclusters y grids es tema actual de investigación y desarrollo.

La facultad de informática cuenta con varios clusters utilizados por alumnos, docentes e investigadores.

Dar mayor disponibilidad de la arquitectura multicluster existente solucionando problemas de acceso.

Proyectos con otras universidades.

---

# Plan de tareas

## Tareas de Investigación

Investigar sobre arquitecturas de cluster y multicluster, ejecución de programas en este tipo de arquitecturas.

Investigar formas de acceso, problemas anteriores, ventajas del acceso remoto, herramientas de acceso remoto.

Investigar como ejecutar programas y comandos de sistemas desde un programa escrito en un lenguaje de programación adecuado, que permita ser accedido remotamente vía Web, para poder ser aplicado a una arquitectura multicluster.

## Tareas de Desarrollo

Desarrollar una aplicación Web, RIA (**R**ich **I**nternet **A**pplication), dar a cada usuario un espacio de trabajo (estructura de directorios y archivos, compilar, ejecutar y detener aplicaciones y para administradores la posibilidad de administrar la estructura).

Configuración de la arquitectura:

Montar y configurar un servidor para la aplicación Web.

Montar y configurar una arquitectura de cluster y/o multicluster para hacerla accesible a través del servidor en forma remota.

**Describir líneas de investigación y desarrollo abiertas.**

# Plan de tareas

## Tareas de Investigación

Investigar sobre arquitecturas de cluster y multicluster, ejecución de programas en este tipo de arquitecturas.

Investigar formas de acceso, problemas anteriores, ventajas del acceso remoto, herramientas de acceso remoto.

Investigar como ejecutar programas y comandos de sistemas desde un programa escrito en un lenguaje de programación adecuado, que permita ser accedido remotamente vía Web, para poder ser aplicado a una arquitectura multicluster.

## Tareas de Desarrollo

Desarrollar una aplicación Web, RIA (**R**ich **I**nternet **A**pplication), dar a cada usuario un espacio de trabajo (estructura de directorios y archivos, compilar, ejecutar y detener aplicaciones y para administradores la posibilidad de administrar la estructura).

Configuración de la arquitectura:

- Montar y configurar un servidor para la aplicación Web.

- Montar y configurar una arquitectura de cluster y/o multicluster para hacerla accesible a través del servidor en forma remota.

Describir líneas de investigación y desarrollo abiertas.

# Conceptos Cluster - Multicluster

Existencia de problemas complejos en distintas áreas:

- Modelos de sistemas biológicos.
- Reacciones químicas.
- Física de partículas.
- Predicción meteorológica.

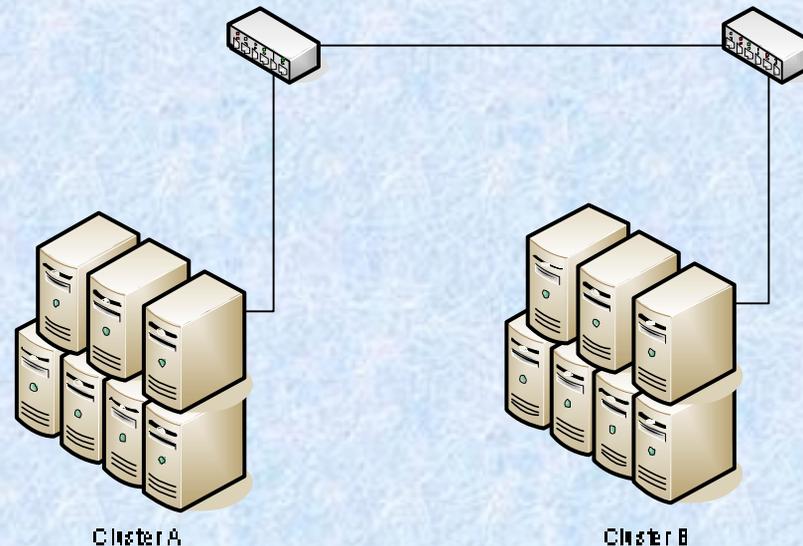
Necesidad de supercomputadoras con capacidad de procesamiento paralelo y alto poder de computo (TOP 500), Muy caras.

Un tipo de supercomputadora mas económica se puede lograr agrupando varias maquinas en un cluster.

# Conceptos Cluster - Multicluster

Un *cluster* consiste de un conjunto de computadoras conectadas en red que se comportan como si fuesen una única computadora.

Varios cluster conectados forman una arquitectura multicluster.



# Conceptos Cluster - Multicluster

Para desarrollo y ejecución sobre clusters se utilizan librerías que funcionan sobre distintos lenguajes. (C, C++, Fortran, Java entre otros).

La librerías mas comunes son las de **programación distribuida**:

- **MPI:** Protocolo estándar de comunicación entre nodos que ejecutan en memoria distribuida.  
Distintas distribuciones. LAM, MPICH, etc.  
Lenguajes (C, C++, Fortran y ADA).
- **PVM:** Protocolo de comunicación entre nodos.  
Lenguajes (C, C++, Fortran, JAVA).

***MPI es mas usado actualmente, definición estándar varias implementaciones, mayor eficiencia, mayor actualización, mejor implementación de comunicación colectiva.***

# Plan de tareas

## Tareas de Investigación

Investigar sobre arquitecturas de cluster y multicluster, ejecución de programas en este tipo de arquitecturas.

Investigar formas de acceso, problemas anteriores, ventajas del acceso remoto, herramientas de acceso remoto.

Investigar como ejecutar programas y comandos de sistemas desde un programa escrito en un lenguaje de programación adecuado, que permita ser accedido remotamente vía Web, para poder ser aplicado a una arquitectura multicluster.

## Tareas de Desarrollo

Desarrollar una aplicación Web, RIA (**R**ich **I**nternet **A**pplication), dar a cada usuario un espacio de trabajo (estructura de directorios y archivos, compilar, ejecutar y detener aplicaciones y para administradores la posibilidad de administrar la estructura).

Configuración de la arquitectura:

- Montar y configurar un servidor para la aplicación Web.

- Montar y configurar una arquitectura de cluster y/o multicluster para hacerla accesible a través del servidor en forma remota.

Describir líneas de investigación y desarrollo abiertas.

# Acceso al cluster – Problemas de acceso

¿Como era el uso de los clusters en la facultad?

## Acceso local al cluster.

Alumnos solo disponían de unas pocas horas por semana.  
Necesidad de personas disponibles para brindar acceso.



## Acceso remoto SSH.

Controles en firewalls.  
Limitado a unos pocos usuarios por seguridad.



En ambos casos no existe una interacción amigable, es poco transparente, sin abstracción y no se tiene un control del cluster como recurso único exclusivo además de permitir operaciones innecesarias.

**Es importante un acceso remoto para dar mayor disponibilidad, abstracción y transparencia, permitiendo el acceso también a usuarios ajenos a la facultad.**

# Acceso al cluster –Soluciones planteadas

Tecnologías de acceso remoto:

✦ Consola:  
telnet  
ssh

✦ Escritorio remoto:  
rdp  
xdmcp  
vnc  
nx

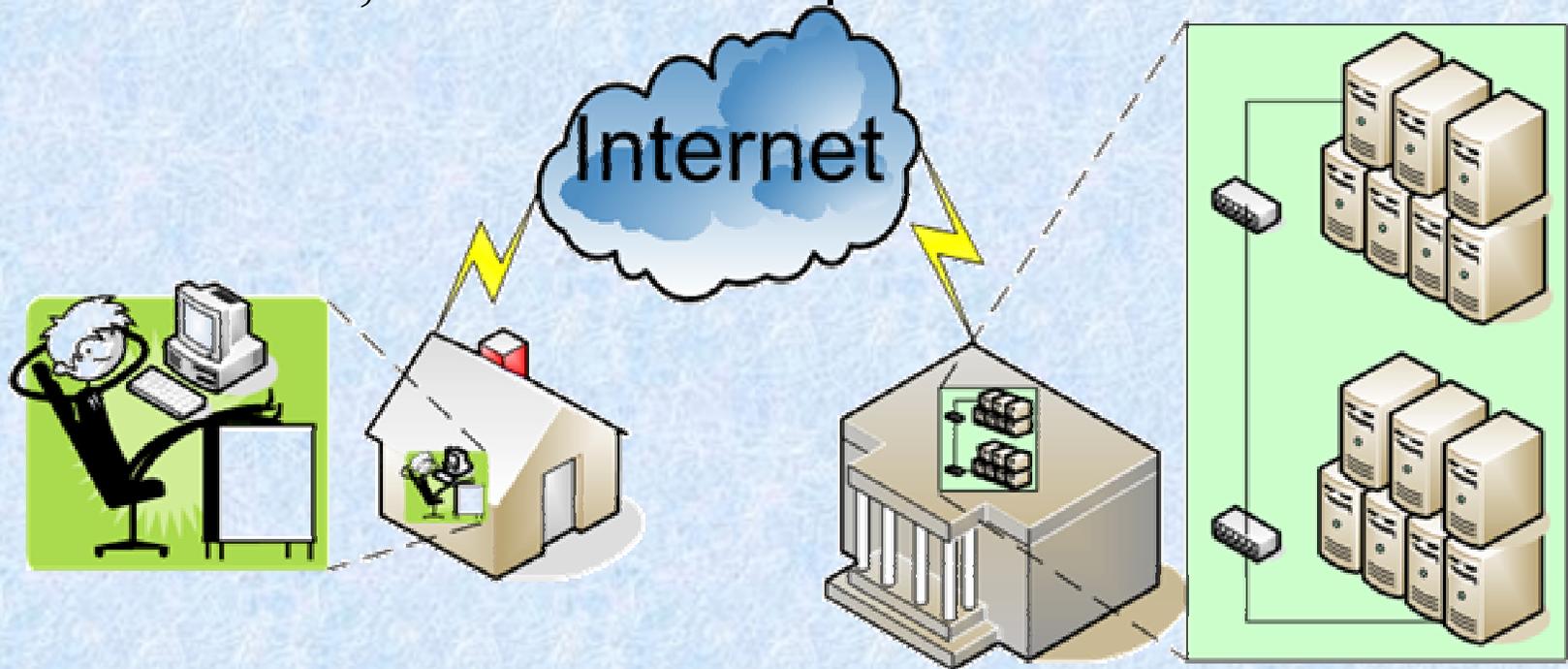
Desventajas:

Útiles pero dan información innecesaria al usuario (ver estructura de archivos, procesos etc.).

Puertos de firewalls abiertos.

# Acceso al cluster - Aplicación WEB

Desarrollar una **Aplicación WEB**: Interacción mas amigable (RIA: uso de AJAX), mayor transparencia y abstracción, limitación en operaciones.



# Plan de tareas

## Tareas de Investigación

Investigar sobre arquitecturas de cluster y multicluster, ejecución de programas en este tipo de arquitecturas.

Investigar formas de acceso, problemas anteriores, ventajas del acceso remoto, herramientas de acceso remoto.

Investigar como ejecutar programas y comandos de sistemas desde un programa escrito en un lenguaje de programación adecuado, que permita ser accedido remotamente vía Web, para poder ser aplicado a una arquitectura multicluster.

## Tareas de Desarrollo

Desarrollar una aplicación Web, RIA (**R**ich **I**nternet **A**pplication), dar a cada usuario un espacio de trabajo (estructura de directorios y archivos, compilar, ejecutar y detener aplicaciones y para administradores la posibilidad de administrar la estructura).

Configuración de la arquitectura:

Montar y configurar un servidor para la aplicación Web.

Montar y configurar una arquitectura de cluster y/o multicluster para hacerla accesible a través del servidor en forma remota.

Describir líneas de investigación y desarrollo abiertas.

# Lenguaje de desarrollo

Elegir un lenguaje apropiado para desarrollar esta aplicación tal que permita :

- ◆ La ejecución de programas y comandos de sistemas.
- ◆ El acceso remoto vía Web.

Elección entre dos lenguajes:

PHP  
JAVA

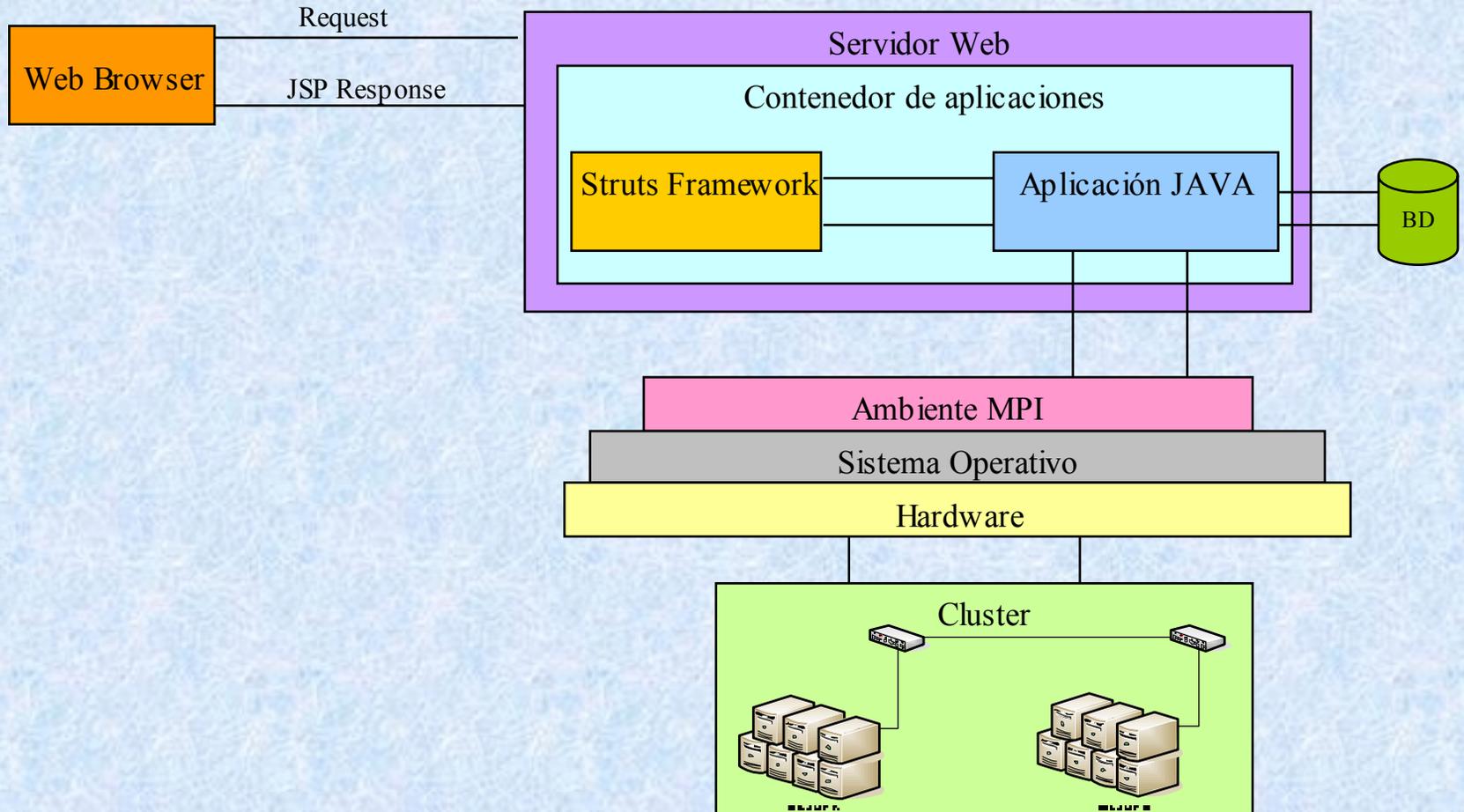
Ambos permiten las características anteriores pero JAVA se adapta mejor.

# Lenguaje de desarrollo - ¿Por qué JAVA?

- Mayor control en la ejecución de comandos y programas externos.
    - Clase **Runtime** (algunos problemas de portabilidad).
    - Control de la ejecución mediante el objeto **Process**.
    - Ejecución sincrónica (waitFor) y asincrónica.
    - Captura de los resultados de ejecución y código de salida.
  - Herramientas para uso de **threads** y **memoria compartida**
    - Métodos synchronized.
    - Monitores con variables de condición, locks (JDK > 1.5).
- Importante para **excluir mutuamente el cluster**.
- MVC (servlets, jsp, **struts**).
  - Adaptación a Grid (se trabaja sobre C, C++, JAVA, WS).

# Estructura general

## MVC y acceso Web a una aplicación JAVA.



# Plan de tareas

## Tareas de Investigación

Investigar sobre arquitecturas de cluster y multicluster, ejecución de programas en este tipo de arquitecturas.

Investigar formas de acceso, problemas anteriores, ventajas del acceso remoto, herramientas de acceso remoto.

Investigar como ejecutar programas y comandos de sistemas desde un programa escrito en un lenguaje de programación adecuado, que permita ser accedido remotamente vía Web, para poder ser aplicado a una arquitectura multicluster.

## Tareas de Desarrollo

Desarrollar una aplicación Web, RIA (**R**ich **I**nternet **A**pplication), dar a cada usuario un espacio de trabajo (estructura de directorios y archivos, compilar, ejecutar y detener aplicaciones y para administradores la posibilidad de administrar la estructura).

Configuración de la arquitectura:

Montar y configurar un servidor para la aplicación Web.

Montar y configurar una arquitectura de cluster y/o multicluster para hacerla accesible a través del servidor en forma remota.

Describir líneas de investigación y desarrollo abiertas.

# Aplicación desarrollada

Algunas características:

- Desarrollada en JAVA.
- Acceso Web: Servidor Tomcat – Framework Struts.
- Ventajas de interacción usando de AJAX, aplicación RIA.
- Consideraciones sobre **usuarios** (de sistema, de aplicación, perfiles, espacio de trabajo).
- **Modos de acceso** al cluster (compartido, exclusivo).
- **Modos de ejecución.**
- Distintas **áreas** según el perfil de usuario.

# Aplicación desarrollada – Niveles de usuarios

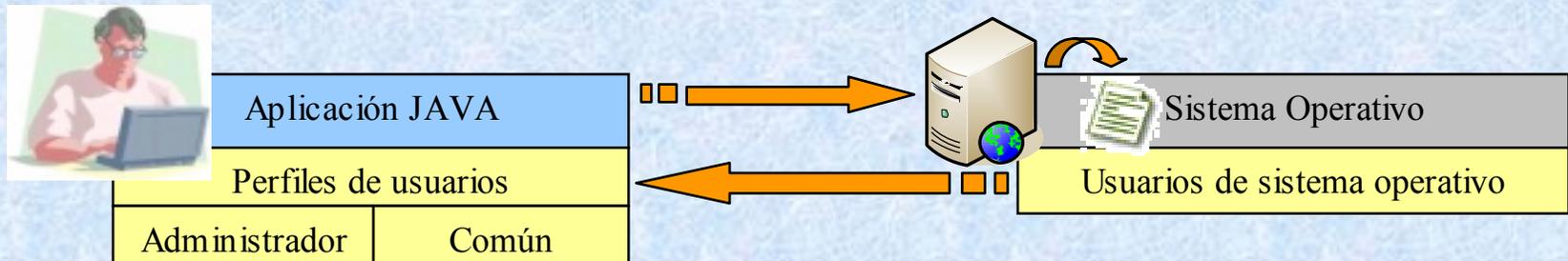
Usuarios en dos niveles:

➤ De aplicación:

Son los usuarios definidos en la aplicación.

➤ De sistema:

Un usuario de aplicación necesita ejecutar programas sobre un SO y escribir o leer archivos y directorios, para esto necesita una correspondencia con un usuario de sistema operativo con permisos para realizar esta tarea.



# Aplicación desarrollada – Usuarios de aplicación

Dos perfiles de usuarios:

## ➤ Usuarios Standard:

- Sistema de archivos.
- Editar.
- Compilar.
- Ejecutar y detener programas.

## ➤ Usuarios Administradores:

- Iguales posibilidades a usuarios Standard.
- Tareas de administración:
  - ABM de usuarios, equipos y clusters.
  - Modos de acceso al cluster.
  - Ambientes de ejecución MPI.
  - Registros de las ejecuciones hechas.

# Aplicación desarrollada – Usuarios de sistema

Correspondencia entre usuarios de aplicación y de sistema:

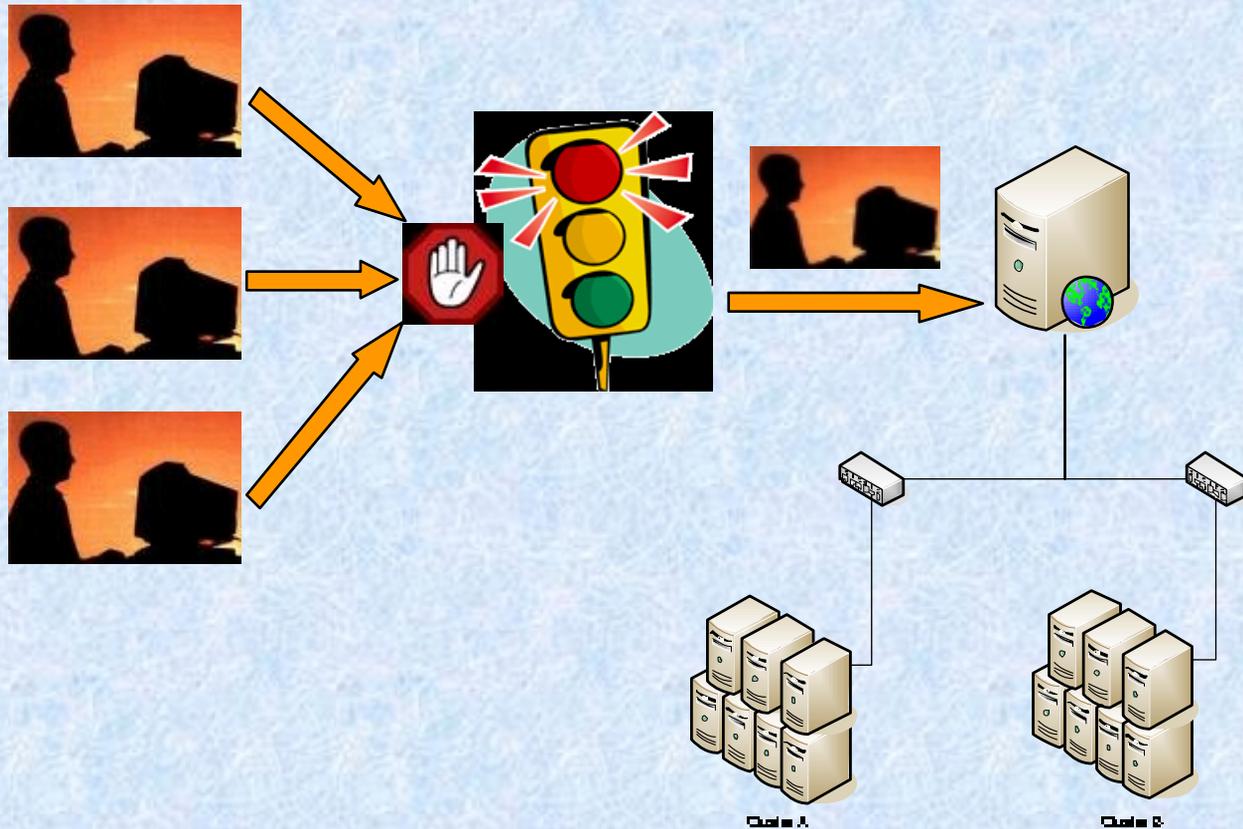
- **1 a 1:** Crear un usuario de sistema por cada usuario de aplicación:
  - Muchas configuraciones por cada usuario nuevo, dificultoso en caso de altas masivas de usuarios.
  - Centralización de usuarios.
  - Cambio de usuario al ejecutar un comando o programa externo.
  - Mayor seguridad.
  
- **1 a muchos:** Crear un usuario de sistema para todos los usuarios de aplicación.
  - Facilita el alta masiva de usuarios.
  - Mas fácil de automatizar.
  - Problemas de seguridad.

Se eligió la segunda alternativa:

- Simple de implementar.
- Necesidades de las cátedras.
- Posibilidad de alta masiva de usuarios en cátedras por año.
- Seguridad con control de ejecuciones y transparencia en espacio de trabajo.

# Aplicación Web – Modos de acceso

La arquitectura de cluster es un recurso en el que dos o mas usuarios no deberían estar ejecutando al mismo tiempo.



# Aplicación Web – Modos de acceso

Para definir los modos de acceso se tuvieron en cuenta quienes utilizan el cluster y los fines con los que se utilizan para definir los modos de acceso.

- Docentes e investigadores: cluster en forma exclusiva por un periodo de tiempo considerable.
- Alumnos: de acuerdo a las cátedras que la utilizan.
  - Programación paralela: ídem docentes e investigadores pero con menor tiempo.
  - Programación concurrente: no requieren exclusividad ni tiempos grandes, introducción a la noción de cluster.

# Aplicación Web – Modos de acceso

De acuerdo a lo anterior los modos resultantes son:

- **Exclusivo:** Se asigna una franja horaria a un usuario, se impide que otros usuarios puedan ingresar al sistema mientras se este haciendo uso de la arquitectura. (Excepción administradores). Docentes e investigadores. Luego del tiempo transcurrido la sesión caduca.
- **Compartido:** Permite la autenticación de usuarios dejándolos operar sobre su área de trabajo, pero sólo les deja ejecutar programas si el cluster no esta ejecutando programas de otro usuario. Alumnos.

# Aplicación Web – Areas

## Areas de acuerdo al perfil:

- Usuarios comunes:
  - Area de ejecución.
  - Area de programas en ejecución (propios).



# Aplicación Web – Areas

Usuarios administradores:

- Area de ejecución (ídem usuarios comunes).
- Area de programas en ejecución (propios y ajenos).
- Area de manejo de usuarios.
- Area de clusters.
- Area de ambientes MPI.
- Area de asignaciones exclusivas.
- Area de registro de ejecuciones.



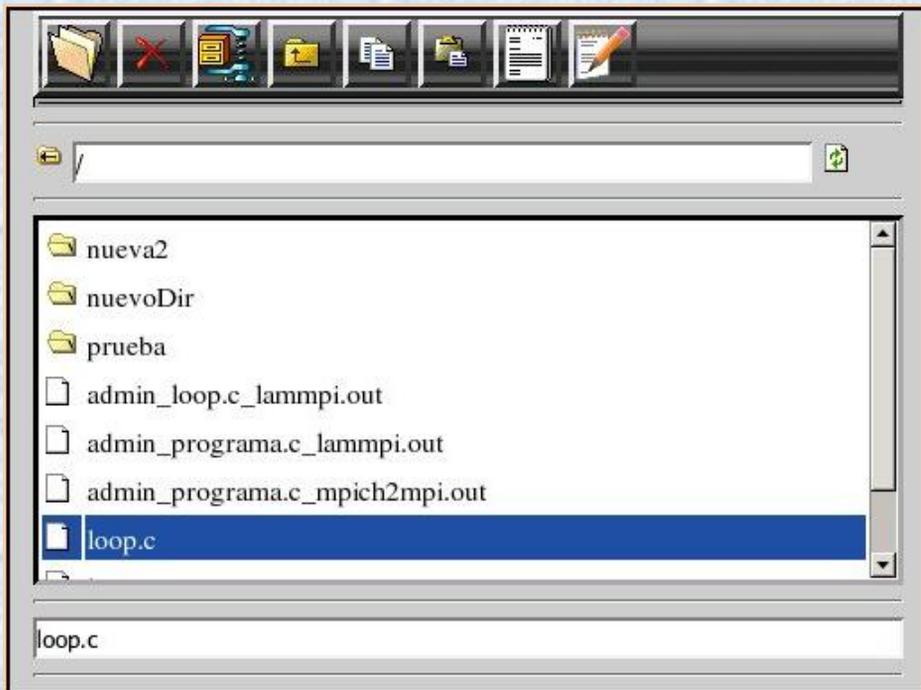
# Aplicación Web – Area de ejecución

- Espacio de trabajo (sistema de archivos, estructura transparente, operaciones sobre archivos y directorios).
- Compilación.
- Ejecución.
- Visualización de resultados.
- Edición de archivos fuentes.

# Aplicación Web – Espacio de trabajo

Sistema de archivos independiente del sistema de archivos del SO.

- ABM archivos y directorios.
- Visualización y edición de archivos.
- Uploads de archivos.
- Unzip.
- Portapapeles.



# Aplicación Web – Compilación - Librerías

La aplicación Web permite el uso de las librerías para programación distribuida MPI utilizando el lenguaje de programación C.

Es posible compilar y ejecutar programas utilizando estas librerías en dos distribuciones LAM y MPICH2.



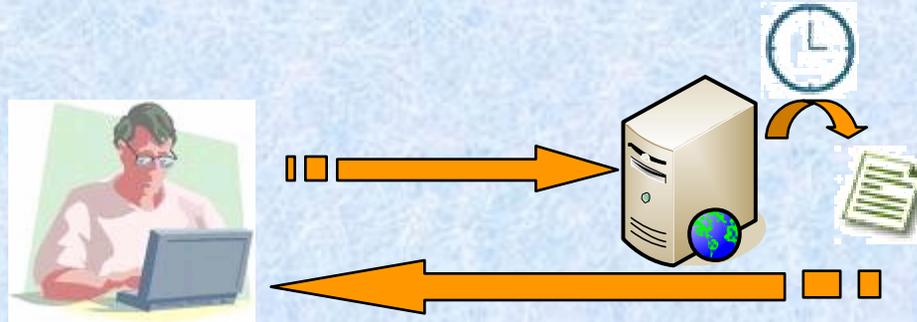
La compilación deja un archivo:

**usuario\_programa\_ambiente.out**

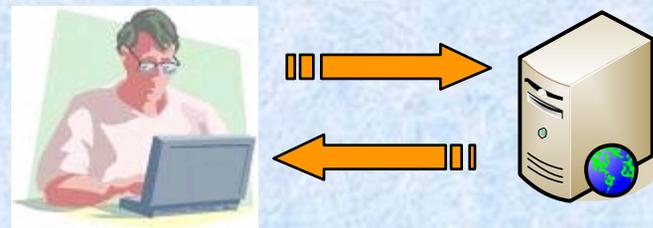
# Aplicación Web - Modos de ejecución

La ejecución se puede hacer en tres modos:

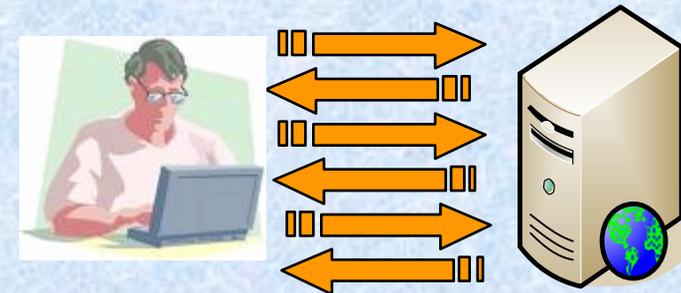
Asincrónico



Sincrónico



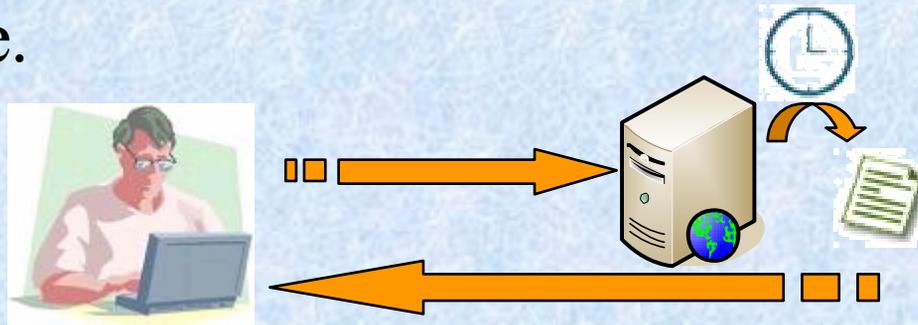
Interactivo



# Aplicación Web - Modo asincrónico

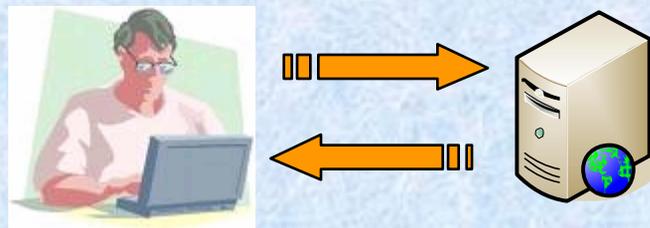
Se envía el pedido de ejecución al servidor, el resultado no es inmediato y la salida de la ejecución es retornada en un archivo dentro del espacio de trabajo del usuario, el contenido de este archivo puede visualizarse posteriormente.

Es utilizado para aplicaciones que tienen una demora importante.



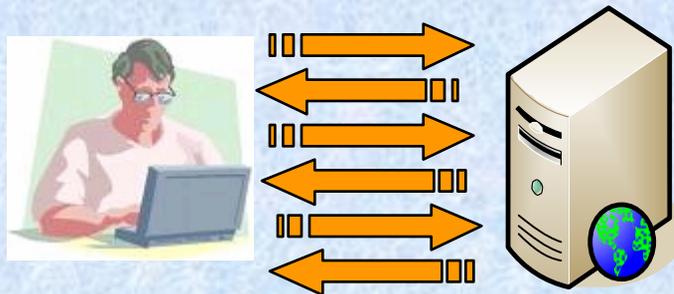
# Aplicación Web - Modo sincrónico

Se envía el pedido de ejecución al servidor y se espera una respuesta en lo inmediato, por lo tanto la salida de la ejecución es retornada en la misma página WEB.



# Aplicación Web - Modo interactivo

Da al usuario la misma sensación que tiene cuando ejecuta una aplicación de este tipo en una consola, donde la salida de la ejecución se va mostrando línea a línea. El usuario envía a ejecutar el programa y el servidor va entregando de a línea o conjunto de líneas a medida que se desarrolla la ejecución, las cuales va mostrando en la pagina Web.



# Aplicación Web – Areas

## Area de programas en ejecución.

El cluster en si necesita de exclusión mutua.

**¿Qué ocurre si un programa no se detiene?**

Los usuarios comunes solo pueden detener sus programas.

Los usuarios administradores pueden detener programas propios y ajenos.

# Aplicación Web – Areas

## Area de Manejo de usuarios.

- Altas, bajas y modificaciones de usuarios a nivel de aplicación.
- Cambio de perfiles.

## Area de Manejo de clusters.

- Altas, bajas y modificaciones de clusters.
- Altas, bajas y modificaciones de equipos (previamente configurados) en clusters .
- Cambio en modos de acceso (compartido, exclusivo).
- Estado de clusters y equipos.

# Aplicación Web – Areas

## Area de ambientes de ejecución.

Los ambientes MPI requieren arrancar o detener “motores” o “daemons”.

Sin ellos corriendo, los programas MPI, no podrán ser ejecutados.

El área muestra el estado de los ambientes y permite arrancarlos o detenerlos.

# Aplicación Web – Areas

## Area de asignaciones exclusivas.

- La arquitectura de cluster tiene dos modos de acceso, compartido y exclusivo, para esta ultima se pueden asignar franjas horarias a los usuarios.
- Si el cluster esta en modo exclusivo solo podrá ingresar el usuario que tenga la asignación correspondiente al tiempo actual del cluster.
- Luego del tiempo transcurrido la sesión caduca.

## Area de registro de ejecuciones.

Control de las ejecuciones realizadas:

- Usuario.
- Fecha y hora.
- Programa
- Modo (sincrónico, asincrónico o interactivo).
- Ambiente.

# Plan de tareas

## Tareas de Investigación

Investigar sobre arquitecturas de cluster y multicluster, ejecución de programas en este tipo de arquitecturas.

Investigar formas de acceso, problemas anteriores, ventajas del acceso remoto, herramientas de acceso remoto.

Investigar como ejecutar programas y comandos de sistemas desde un programa escrito en un lenguaje de programación adecuado, que permita ser accedido remotamente vía Web, para poder ser aplicado a una arquitectura multicluster.

## Tareas de Desarrollo

Desarrollar una aplicación Web, RIA (**R**ich **I**nternet **A**pplication), dar a cada usuario un espacio de trabajo (estructura de directorios y archivos, compilar, ejecutar y detener aplicaciones y para administradores la posibilidad de administrar la estructura).

Configuración de la arquitectura:

- Montar y configurar un servidor para la aplicación Web.

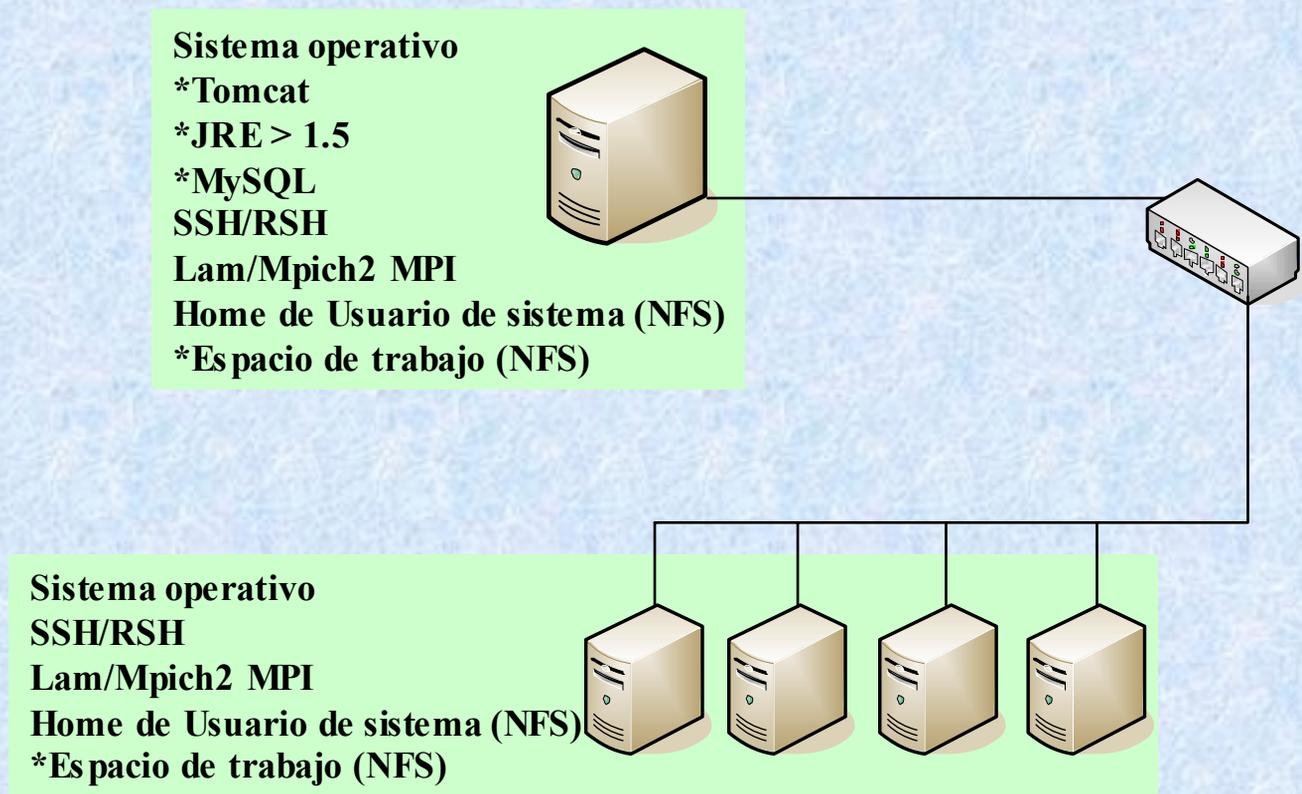
- Montar y configurar una arquitectura de cluster y/o multicluster para hacerla accesible a través del servidor en forma remota.

Describir líneas de investigación y desarrollo abiertas.

# Configuración de la arquitectura

Pasos para configurar un cluster.

Como se adapta a la aplicación Web.



(\* Tareas para adaptar la aplicación Web)

# Configuración de la arquitectura

En todos los equipos:

- Sistema operativo (Mayoría Fedora, Ubuntu).
- Configuración de paquetes y servicios (GCC, G++, NFS, SSH/RSH).
- Usuario de sistema (Idénticas características, *centralización a futuro*).
- Ambientes de ejecución MPI (Lam y Mpich2).
- Configuración SSH, RSH (acceso sin contraseña).

En el servidor:

- **\*Servidor Web.**
- **\*Motor de base de datos MySQL.**
- **\*JAVA (> 1.5).**
- **\*Crear directorio para el espacio de trabajo de los usuarios.**
- **\*Compartir el espacio de trabajo vía NFS.**
- Compartir el Home Directory del usuario de sistema vía NFS.

En las maquinas del cluster:

- **\*Crear directorio para el espacio de trabajo de los usuarios.**
- **\*Corresponder el directorio de espacio de trabajo con el compartido en el servidor.**
- Corresponder el Home Directory de usuario de sistema con el compartido en el servidor.

# Herramientas utilizadas

Software de distribución **gratuita**:

- Sistema operativo: **Fedora, Ubuntu.**
- Lenguaje de desarrollo: **JAVA.**
- Servidor Web con contenedor de aplicaciones: **Tomcat.**
- Ambiente de desarrollo: **Eclipse Lombok (+Amateras), Struts.**
- Ambientes MPI: **Lam, Mpich2.**
- Lenguajes de programación: **GCC, G++.**
- Motor de base de datos: **MySql.**

# Conclusiones

Una aplicación accesible remotamente vía Web, para administración de una arquitectura multi-cluster y para uso por parte de alumnos, docentes e investigadores.

Mayor disponibilidad y transparencia.

Mayor control en el acceso al cluster como recurso compartido.

Facilidad en la administración.

Posibilidades de acceso a personas ajenas a la facultad de informática, que trabajen sobre clusters y no puedan contar o configurar una arquitectura de este tipo.

# Plan de tareas

## Tareas de Investigación

Investigar sobre arquitecturas de cluster y multicluster, ejecución de programas en este tipo de arquitecturas.

Investigar formas de acceso, problemas anteriores, ventajas del acceso remoto, herramientas de acceso remoto.

Investigar como ejecutar programas y comandos de sistemas desde un programa escrito en un lenguaje de programación adecuado, que permita ser accedido remotamente vía Web, para poder ser aplicado a una arquitectura multicluster.

## Tareas de Desarrollo

Desarrollar una aplicación Web, RIA (**R**ich **I**nternet **A**pplication), dar a cada usuario un espacio de trabajo (estructura de directorios y archivos, compilar, ejecutar y detener aplicaciones y para administradores la posibilidad de administrar la estructura).

Configuración de la arquitectura:

- Montar y configurar un servidor para la aplicación Web.

- Montar y configurar una arquitectura de cluster y/o multicluster para hacerla accesible a través del servidor en forma remota.

**Describir líneas de investigación y desarrollo abiertas.**

# Líneas de I/D futuras

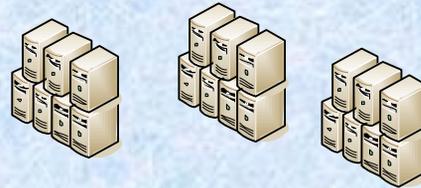
Aspectos de seguridad.



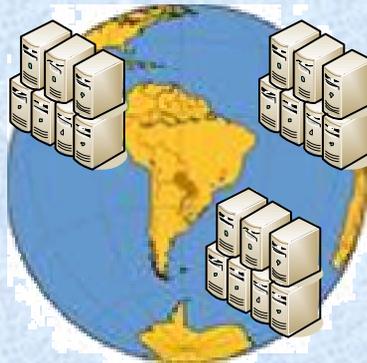
Scheduler y administrador de recursos.



Fragmentación y escalabilidad.



Extensiones a grid.



# Líneas de I/D futuras

## Aspectos de seguridad

Problema:

**Desconocimiento del contenido del código a ejecutar en el cluster.**

Solución actual:

- Un usuario de sistema creado sin privilegios.
- Todos los servicios para ejecución corren sin privilegios de sistema.
- Problema: No impide que se pueda leer algún área de sistema.
- Importancia de la transparencia y control de ejecuciones.

Estudiar otras políticas:

- Usuarios de aplicación equivalentes a usuarios de sistema.
- ACL y/o LDAP.
- Analizar los procesos de configuración para altas masivas de usuarios.

# Líneas de I/D futuras

## Schedulers y administradores de recursos.

Se utilizarían para dar la posibilidad al usuario, en caso de que el cluster este ocupado, de encolar el pedido de ejecución.

Los **schedulers** deciden las políticas con las que se ejecutan determinados trabajos, determinan prioridades etc. (Ejemplo **Maui**).

Los **administradores de recursos** permiten administrar los recursos de una arquitectura multicluster, proveen control sobre trabajos encolados y permiten una mejor distribución de los mismos sobre los recursos del cluster. (Ejemplo **Torque**).

Ambos están probándose en entornos de Grid.

# Líneas de I/D futuras

## Fragmentación y escalabilidad.

**Separar o fragmentar** el cluster en “subclusters” aprovechando áreas del cluster que no estén en uso:

- Estática (que usuario puede usar que parte del cluster).
- Dinámica utilizando un manejador de recursos.

Problemas con MPI (dominio de maquinas, mapeo).

**Escalar** para poder utilizar cluster en una red mas amplia.

Problemas con MPI (pensados para LAN).

# Líneas de I/D futuras

## Extensiones a grid

Un *Grid* es un tipo de sistema distribuido que permite compartir e integrar recursos distribuidos geográficamente.

Características:

- Mayor complejidad.
- Lenguajes multiplataforma (heterogeneidad, recompilado en C).
- Web services.
- Mpich-g2(por problemas de MPI en WANs).

**Adaptar la aplicación a Clusters de nodos de Grid.**

Proyectos:

- Actualmente Cyted (Globus Toolkit 4).
- EELA2.