

Laboratorio de Procesamiento Paralelo Multi-Cluster accesible vía WEB

Adrián Pousa, Armando E. De Giusti, Marcelo Naiouf

Instituto de Investigación en Informática LIDI
Facultad de Informática – Universidad Nacional de La Plata
{apousa , degiusti, mnaiouf}@lidi.info.unlp.edu.ar

Resumen

El laboratorio remoto paralelo multicluster es una aplicación Web que permite el acceso a una arquitectura multicluster tanto para ejecutar programas sobre ella como para administrarla.

La aplicación Web da la posibilidad a alumnos, docentes e investigadores de trabajar en forma remota sobre una arquitectura de este tipo de manera transparente, asignándoles un espacio de trabajo y permitiendo compilar y ejecutar programas con distintos niveles de interacción.

Una aplicación Web para trabajar sobre una arquitectura multicluster, cambia la forma en la que se utiliza el cluster: normalmente el usuario trabajaba localmente en el lugar físico donde se encuentran las máquinas o remotamente usando protocolos que requieren habilitar reglas de firewalls, lo que dificulta la transparencia y genera requerimientos de seguridad que producen un importante overhead.

En este trabajo se analizan los requerimientos del software desarrollado para el acceso remoto a los Clusters de la Facultad de Informática de la UNLP, utilizados por varias cátedras de grado y postgrado, en particular las relacionadas con procesamiento concurrente, distribuido y paralelo.

Palabras clave: *Clusters, Sistemas Distribuidos, Procesamiento Paralelo, Sistemas Web.*

1. Introducción

La utilización de redes de computadoras como maquinas paralelas virtuales se ha generalizado y el desarrollo de software de administración de recursos físicos, comunicaciones y software disponible en clusters, multiclusters y grids es un tema actual de investigación y desarrollo.

Sintéticamente, un *cluster* es un tipo de sistema paralelo que consiste de un conjunto de computadoras independientes, conectadas por alguna forma de red, y que configuran un recurso de procesamiento integrado.[1] [2] [3] Actualmente los clusters constituyen una clase de procesadores de alto rendimiento muy utilizados, al punto que más del 60% de las computadoras que figuran en el ranking de las “top-500” son clusters.[4]

Cuando se conectan dos o más clusters sobre una red tipo LAN o WAN, tenemos alguna forma de *multicluster*. Existen diferentes tipos de multicluster según las características de los procesadores, el sistema operativo que utiliza cada uno y la red que los conecta. Básicamente cada uno de estos aspectos define un grado de heterogeneidad en la configuración global, que será muy importante al estudiar performance.[5]

Un *Grid* es un tipo de sistema distribuido que permite compartir e integrar recursos distribuidos geográficamente, incluyendo procesadores, instrumentos, bases de datos e incluso recursos humanos. Esta clase de arquitectura permite definir múltiples configuraciones colaborativas entre comunidades de usuarios, donde cada usuario “ve” una arquitectura única.[6] [7] [8]

En la Facultad de Informática existen varios subsistemas en red que son utilizados por alumnos, docentes e investigadores como maquinas paralelas virtuales: Cluster de 16 maquinas (CL1), Cluster Sala IBM (CL2) y Cluster Sala 101 (CL3). Además se realizan investigaciones de procesamiento paralelo empleando todas las máquinas de CL1, CL2 y CL3.

Por otra parte (dentro de los proyectos de I/D del III-LIDI [9]) estos clusters trabajan “en GRID” con redes de procesadores de otras Universidades.

El acceso a la arquitectura multicluster disponible en la Facultad requería que los usuarios trabajaran en el lugar físico donde se encuentran los clusters o bien remotamente a través de protocolos que requieren habilitar reglas de *firewalls*, que no dan al usuario una vista transparente y que pueden tener problemas de seguridad que se pueden evitar.

Por estas razones se decidió implementar una aplicación Web que permita el acceso remoto, mayor transparencia e interacción y que de a los usuarios un espacio de trabajo sobre una arquitectura multicluster de manera que puedan compilar, ejecutar programas y obtener resultados.

Además la aplicación da la posibilidad de administración remota de la arquitectura multicluster.

Para lograr una mayor transparencia e interacción se utilizaron técnicas basadas en AJAX lo que hace de la aplicación Web una aplicación RIA (Rich Internet Application).[10]

2. Aplicación desarrollada

2.1 Perfiles de usuario

La aplicación Web permite dos perfiles de usuarios:

- Usuarios Standard (fundamentalmente alumnos): tienen un espacio de trabajo que es equivalente a un sistema de archivos, pueden editar, compilar y ejecutar programas en los tres modos

de ejecución disponibles además de detener estas ejecuciones.

- Usuarios Administradores (docentes): agregan a las posibilidades de los usuarios comunes las tareas de administración tales como realizar altas, bajas y modificaciones de usuarios, de equipos y clusters, cambiar modos de acceso al cluster, correr o detener ambientes de ejecución MPI y manejar registros de las ejecuciones hechas.

2.2 Espacio de trabajo

Cada usuario dentro de la aplicación Web tiene un espacio de trabajo semejante a un sistema de archivos de cualquier sistema operativo.

Este espacio de trabajo permite a los usuarios crear, modificar y eliminar archivos o directorios, permite subir archivos, comprimirlos o descomprimirlos utilizando zip, editar archivos, visualizarlos, compilar código fuente y ejecutar el código compilado en los tres modos de ejecución disponibles.

El espacio de trabajo es independiente del sistema de archivos del sistema operativo, lo que brinda un aceptable nivel de transparencia.

2.3 Compilación y modos de ejecución

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.[11] [12]

Una vez compilado un programa puede ser ejecutado utilizando tres modos de ejecución:

- 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.

- **Asincrónico:** Se envía el pedido de ejecución al servidor y no se espera una respuesta en lo inmediato. Es utilizado para aplicaciones que tienen una demora importante, por lo tanto la salida de la ejecución es retornada en un archivo dentro del espacio de trabajo del usuario, y el contenido de este archivo puede visualizarse posteriormente.
- **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 página Web.

Los modos sincrónico e interactivo son similares, la diferencia es que el modo sincrónico envía el pedido de ejecución y espera toda la salida de programa completa, y luego esta salida es mostrada en la página Web. Si el programa fallara por algún motivo el usuario no sabría la causa de la falla o el momento en que el programa se detuvo, recibiendo sólo una pantalla de salida en blanco o un mensaje genérico de error. En el caso del modo interactivo, por cada línea o conjunto de líneas que retorne el programa que se este ejecutando en el cluster, se irán mostrando una por una en la página Web, de esta forma si el programa se detuviera o fallara inesperadamente le dará una idea al usuario de dónde puede estar el error.

Las ejecuciones sobre una arquitectura de cluster, generalmente, suelen parecerse más al modo de ejecución asincrónico donde se envía a un programa ejecutar al cluster y se espera una salida en un archivo con una demora importante de tiempo (minutos, horas, unos pocos días). Notar que las aplicaciones de tratamiento masivo de datos, que

habitualmente se ejecutan en clusters suelen tener estos requerimientos.

Se supone que el programa fue anteriormente probado, asegurando la ausencia de errores, y luego ejecutado en el cluster, por lo tanto “no debería fallar”.

Los modos sincrónico e interactivo de la aplicación surgen por el uso práctico que se le da en algunas cátedras de la carrera, donde los programas no suelen tener demoras importantes, son más bien simples y el uso del cluster es a modo educativo, con fines académicos.

2.4 Modos de acceso

La arquitectura de cluster es un recurso en el que dos usuarios no deberían estar ejecutando al mismo tiempo para que puedan aprovecharse al máximo todos los recursos.

La aplicación Web permite establecer dos modos de acceso:

- **Exclusivo:** Se asigna a un usuario una franja horaria para que pueda trabajar impidiendo que otros usuarios puedan ingresar al sistema mientras se este haciendo uso de la arquitectura. No se les permite ningún tipo de acceso a otros usuarios (ni siquiera cuando el cluster esté ocioso), y no se les permite autenticación, salvo que sean administradores y que se deba solucionar algún problema para que el usuario que tiene asignado el uso exclusivo del cluster pueda usarlo sin inconvenientes.
- **Compartido:** A pesar del nombre, el modo compartido, no significa que se permita el uso del cluster a dos procesos de distintos usuarios al mismo tiempo, sino que se sigue manteniendo la exclusividad en la ejecución sobre la arquitectura de cluster. 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; de esta forma se aprovechan los tiempos ociosos de la arquitectura.

En caso de que el usuario desee ejecutar un programa sobre el cluster y el mismo esté ocupado, la aplicación reintentará la ejecución hasta que logre el acceso para ejecutar o bien hasta que el usuario cancele el pedido de ejecución.

Estos dos modos surgen de las necesidades de los usuarios, los alumnos suelen correr programas de muy corta duración (algunos minutos) que no requieren un tiempo exclusivo y que bien puede compartirse con otros alumnos; en cambio, los investigadores, suelen ejecutar programas de mayor duración (horas) y requieren exclusividad de la arquitectura por un determinado periodo de tiempo.

3. Aspectos de seguridad

El riesgo más importante que se corre, en una aplicación Web que ejecute un programa en la arquitectura multicluster, es que no se conoce el tipo de código que se está ejecutando.

Fue importante tomar medidas de manera que los usuarios no puedan ejecutar tareas de administrador de sistema operativo.

4. Evolución de la aplicación Web

Existen algunos aspectos que se están estudiando para agregar mayor funcionalidad a la aplicación Web.

Un aspecto es el de ejecutar programas utilizando un manejador de recursos y un scheduler como pueden ser Torque y Maui respectivamente.[13] [14] Esta combinación permite que los programas se encolen y sean ejecutados de acuerdo a alguna política de scheduler además de una mejor utilización de los recursos.

Otro aspecto es el de fragmentar la arquitectura multicluster de manera de tener varios clusters para distintos propósitos y con diferentes políticas; por ejemplo, varios alumnos pueden estar utilizando parte de un

cluster de forma compartida mientras que al mismo momento un investigador puede tener parte del cluster de forma exclusiva.

Así como la arquitectura puede fragmentarse puede también expandirse incorporando clusters que tal vez no estén físicamente en el mismo lugar. Actualmente, se tiene la idea de conectar clusters de otras universidades.

Un último aspecto es el de adaptar la aplicación Web para que funcione en una infraestructura de grid, donde se consideran equipos como nodos de grid distribuidos ya no en una LAN (como pueden considerarse los clusters) sino en una WAN, y cada nodo puede tener a su vez clusters conectados. Esta infraestructura proporciona mayor disponibilidad de recursos y puede suponer una mayor potencia de cálculo.

5. Conclusiones

Se ha presentado un desarrollo de entorno y soporte de administración de recursos WEB, para la utilización de arquitecturas multiprocesador tipo cluster y multicluster en forma remota.

Este tipo de solución es particularmente importante en las carreras de Informática, donde un número importante de alumnos de grado y postgrado requieren trabajo experimental sobre arquitecturas multiprocesador disponibles en la Facultad.

6. Bibliografía

- [1] Grama A., Gupta A., Karypis G., Kumar V. "Introduction to Parallel Computing". Second Edition. Addison Wesley, 2003.
- [2] Jordan H, Alagband G. "Fundamentals of parallel computing". Prentice Hall, 2002.
- [3] Dongarra J, Foster I, Fox G, Gropp W, Kennedy K, Torczon L, White A. "The Sourcebook of Parallel Computing". Morgan Kauffman Publishers. Elsevier Science, 2003.

- [4] The TOP500 project <http://www.top500.org/>
- [5] Jiang, Yeung. “Scalable Inter-Cluster Communication System for Clustered Multiprocessors”. 1997.
- [6] Foster I., Kesselman C., Kaufmann M. “The Grid 2: Blueprint for a New Computing Infrastructure”. The Morgan Kaufmann Series in Computer Architecture and Design. 2 edition (November 18, 2003).
- [7] Juhasz Z. (Editor), Kacsuk P. (Editor), Kranzlmuller D. (Editor). “Distributed and Parallel Systems: Cluster and Grid Computing”. The International Series in Engineering and Computer Science. Springer; 1 edition (September 21, 2004)
- [8] Berman F.(Editor), Fox G.(Editor), Hey A.(Editor). “Grid Computing: Making The Global Infrastructure a Reality”. John Wiley & Sons (April 8, 2003).
- [9] III-LIDI <http://lidi.info.unlp.edu.ar>
- [10] AJAX ORG <http://www.ajax.org/>
- [11] LAM/MPI Parallel Computing <http://www.lam-mpi.org>
- [12] MPICH2/MPI Parallel Computing <http://www-unix.mcs.anl.gov/mpi/mpich2/>
- [13] Torque Resource Manager <http://www.clusterresources.com/pages/products/torque-resource-manager.php>
- [14] Maui Cluster Scheduler <http://www.clusterresources.com/pages/products/maui-cluster-scheduler.php>