Ambientes de Computación Paralela/Distribuida para NOW

Barbieri, Andres
Tinetti, Fernando
Bertone, Rodolfo
De Giusti, Armando

{barbieri,fernando,rbertone,degiusti}@lidi.info.unlp.edu.ar

Laboratorio de Investigación y
Desarrollo en Informática (LIDI)
Facultad de Informática
Universidad Nacional de La Plata

Calle 50 y 115, 1900 La Plata Tel/Fax. 0221-4227707

1 Introducción

Las computadoras y los sistemas informáticos día a día evolucionan y en ocasiones toman direcciones radicalmente diferentes a las tradicionales. Desde la época de los grandes sistemas centralizados tipo mainframes y los procesos por lote hasta hoy, el sentido de la tecnología en el área ha sufrido muchos cambios. A partir de la década de los ochenta los avances han sido cada vez más relevantes. El área de microprocesadores mantiene un crecimiento sostenido, lo que lleva a que hoy, cualquier computadora de escritorio tenga capacidad en el orden de GFlops (mil millones de operaciones de punto flotante por segundo). Por otro lado, con respecto a las comunicaciones, se obtuvieron capacidades que permitieron la explosión de "Internet", lo que significa millones de computadores y personas en el mundo intercambiando información durante todo el tiempo. El desarrollo de las redes locales, LANs, permiten conectar edificios enteros con velocidades de 100 y hasta 1000 Mb/s (millones de bits por segundo). En efecto, todos estos resultados permitieron que se reflote una idea que hoy es una realidad: "Los Sistemas Distribuidos": Colecciones de computadoras independientes que pueden ser utilizadas en conjunto como una única computadora para ser explotada con diferentes propósitos.
Uno de los principales usos de "los Sistemas Distribuidos" es el de ejecutar aplicaciones paralelas, debido a la inmensa potencia de cómputo y al bajo costo de la misma. La necesidad de reducir los tiempos de procesamiento de grandes volúmenes de datos parece tener una respuesta adecuada mediante el uso de redes de computadoras tradicionales. Uno de los problemas que se presentan al hacer el "matching" entre el problema y la solución es que las NOW (redes de estaciones de trabajo conformadas por equipos heterogéneos u homogéneos) y en particular los clusters (redes de equipos homogéneos) no ofrecen una visión única de sistema a comparación de una máquina paralela tradicional, sino que cada equipo o nodo posee un conjunto propio e independiente de recursos que sólo puede ser utilizado desde el mismo nodo. Si no se agregan mecanismos y/o medios para compartir y agrupar recursos, se limita el poder de las NOW como máquinas paralelas. Una de las formas de aprovechar esta tecnología es tener una visión

global de algunos o todos los recursos, dando una vista única del sistema. La forma de lograrlo se conoce como SSI (Single System Image). Así es posible disponer de los recursos desde cualquier equipo en la red de forma confiable y transparente. Además, nunca se debe perder de vista el rendimiento, que es uno de los principales objetivos del cómputo paralelo. Los servicios tradicionales que puede brindar un SSI son:

- Memoria distribuida compartida
- Espacio unificado de procesos
- Migración transparente de procesos
- Scheduling global y co-scheduling, sistema de colas
- Imágen de file system única
- Administración centralizada

La forma resolverlos consiste en implementar capas o estratos de software y/o hardware que haga que cada sistema o nodo independiente se transforme en parte de un sistema mayor mucho más potente.
Los beneficios extra que se tienen a partir de crear una SSI además de aprovechar los equipos para cómputo paralelo, son las facilidades de administración y desarrollo de aplicaciones. A éstos se agregan los pros inherentes de los sistemas distribuidos: la tolerancia a fallas o confiabilidad, los beneficios económicos (mejor relación precio/rendimiento) y la escalabilidad.

2 Implementación de SSI Existentes

Las formas prácticas de crear SSI se pueden dar a diferentes grados y en diferentes niveles. Se puede hacer de forma total o completa ofreciendo todos los servicios (Memoria compartida distribuida, Single File System, identificación de procesos en todo el sistema distribuido, etc.) integrados en un solo módulo o parcial, dando algún servicio particular, por ejemplo solamente memoria compartida físicamente distribuida. Los niveles en los cuales podemos enmarcar a los SSI son:

- Hardware
- Sistema Operativo (Underware)
- Middleware (Runtime Subsystem)
- Aplicación

Otra forma de clasificar los proyectos sobre SSI que no sea de acuerdo a un nivel o a la cobertura es según el objetivo:

- SSI para administración del sistema.
- SSI a nivel de system-calls para las aplicaciones.

El primero en general se obtiene a nivel de middlewares o aplicaciones ejecutando procesos y servicios que facilitan las tareas de manejo del sistema como un todo. La otra tiene que ver con la transparencia que debe tener la ejecución de programas en el cluster con SSI o sobre una computadora en particular. La clasificación que usaremos para mostrar el estado actual y antecedentes en SSI para "Cluster Computing y NOW" es la primera.
A nivel de hardware se encuentran sistemas como el "Memory Channel", llevado a cabo por Digital (DEC) y Compaq que implementa DSM (Distributed Shared Memory) y ofrece a los usuarios ver todas las memorias distribuidas a lo largo de un cluster como una única y compartida, accesible desde cualquier punto. Para su uso se requiere un módulo de software encargado de la reserva y mapping de páginas, y en un nivel más alto ofrece una interfaz de usuario basada en pasaje de mensajes. Es usado por middlewares como implementaciones de MPI (Message Passing Interface), la librería PVM (Parallel Virtual Machine) o extensiones de lenguajes como HPF (High Performance Fortran). Al ser un desarrollo en hardware tiene la ventaja de tener una latencia muy baja y un alto ancho de banda con un pico de 88 MB/s. El nivel de hardware no fue muy desarrollado debido a que las tecnologías de interconexión no ofrecían buenas prestaciones. Hoy en día debido a nuevas innovaciones como SCI (Scalable Coherent Interface) o Myrinet que tienen bajas latencias y alto "data rate" se ha permitido que se desarrollen mejores soluciones. De cualquier forma, es el área más costosa, lo que produce que en ocasiones se pierda la ventaja de los clusters con respecto al precio.
En el área de Sistemas Operativos los representantes de SSI que se encuentran son más variados: En el sector comercial se tiene a SCO UnixWare NonStop Clusters y Sun Solaris-MC. El primero es una extensión del Unix UnixWare. Se ofrecen servicios completos de un sistema operativo sobre todo el cluster, file system distribuido, acceso a dispositivos remotos o locales de forma transparente, IPC (Inter Process Communication), migración activa de procesos, balance de carga de CPU y tolerancia a fallas. Otro competidor con caracterísicas similares Solaris-MC que está desarrollado con tecnologías de Objetos usando C++ y el estándar CORBA (Common Object Request Broker) definido por el OMG (Object Management Group).
En el sector de Open Source encontramos a GLUnix desarrollado dentro del proyecto NOW de la Universidad de Berkeley, California. Tiene la ventaja de ser implementado totalmente en espacio de usuario sin necesidad de modificar el kernel permitiendo una "fácil" portabilidad, aunque las versiones actuales ejecutan solo sobre Solaris. Una de las principales características que ofrece es el co-scheduling. Otro producto Open Source es MOSIX de la Universidad de Hebrew en Jerusalén. Este ofrece un conjunto de servicios como el anterior pero sobre un kernel Linux modificado. Encontramos como una idea un tanto diferente a lo visto hasta ahora a PUMA Operating System, del Sandia National Labs. y la Universidad de Nuevo México que corre sobre un Intel Paragon y también es usado en los proyectos ASCI (Accelerated Strategy Computing Initiative). Este toma una posición completamente minimalista: no tiene nada compartido entre los nodos, debido que ejecuta sólo sobre la "partición de cómputo" ofreciendo así alto rendimiento en comunicaciones y cómputo. Si bien esta idea no se ha aplicado directamente a clusters tradicionales es un buen antecedente para futuros trabajos.
Con respecto a los middlewares (capa que reside entre el sistema operativo y las aplicaciones de usuario) encontramos numerosas variantes. Tenemos a PVM que basado en el paradigma de pasaje de mensajes ofrece hacer de una red heterogénea una máquina virtual paralela. Existen numerosos programas que hacen manejo o administración de colas batch, poniendo a disponibilidad de los usuarios una gran cantidad de ciclos de CPU ociosos. CONDOR, PSB y CODINE son ejemplos significativos. CONDOR es un proyecto Open Source desarrollado por la Universidad de Madison en Wisconsin que permite hacer scheduling global y migración de procesos a lo largo de extensas redes. CODINE es un administrador de recursos (básicamente de ciclos de CPU) para redes heterogéneas. Surgió en la Universidad de Florida y se compone de varias

partes, un daemon master que es el encargado de mantener una base de datos sobre la cual el scheduler tomará las decisiones para hacer el balance de carga. La información recopilada en la base de datos es provista por los daemons de ejecución, que están en todas las máquinas del cluster. Para la comunicación entre los diferentes daemons existen otros procesos encargados de las comunicaciones, que las resuelven vía TCP. Al nivel de las aplicaciones encontramos el LVS (Linux Virtual Server). Es un servidor altamente escalable de servicios tradicionales de red TCP/IP. Sobre un cluster rutea requerimientos a diferentes equipos de acuerdo a un algoritmo de scheduling. Con respecto a la administración del sistema como un todo tenemos varias iniciativas. Por ejemplo OSCAR, que es software para instalar de forma rápida un cluster homogéneo proveyendo una serie de herramientas post-instalación para facilitar la administración como los PUCs (Parallel Unix Commands). Dentro del Proyecto Beowulf también se ofrecen una serie de herramientas (scripts y programas) que facilitan las tareas.

3 Tareas de Investigación

Inicialmente, se debe hacer un análisis exhaustivo de las variantes existentes de SSI en cada uno de los niveles. Básicamente, se trata de identificar en la bibliografía disponible las características y requerimientos de cada una de estas variantes. Es posible que sea necesaria la instalación y utilización de alguna(s) de ellas (al menos entre las que son de uso libre, disponibles en Internet) para una mejor comprensión de los conceptos involucrados y valoración de las capacidades, requerimientos y limitaciones. De la etapa inicial de análisis se buscará la identificación de: Uno o un conjunto de servicios de SSI sobre los que se profundizará la investigación El conocimiento necesario para avanzar en la caracterización, el aprovechamiento y las variantes posibles de implementación de dicho(s) servicio(s).
La segunda etapa consiste en avanzar con los conceptos de rendimiento, desarrollo de aplicaciones, transparencia, tolerancia a fallas, etc. de uno o un conjunto de servicios considerados representativos y/o esenciales de SSI. Este análisis será exhaustivo y por lo tanto puede involucrar no solamente la instalación de uno o varios paquetes de software sino su utilización en aplicaciones específicas (aunque no muy complejas, para evitar el desvío de la atención en la investigación).
Una tercera etapa puede considerarse la propuesta y la implementación efectiva de mejoras en rendimiento y/o en capacidades de alguno o un conjunto de servicios de SSI. En este sentido, teniendo la visión global del análisis de la primera etapa y el conocimiento más detallado de la segunda etapa (que puede incluir desarrollo de aplicaciones), será posible proponer con el suficiente criterio de aporte significativo e implementar con éxito la propuesta que se haga.

Expresandolo de forma más específica Las tareas de investigación a desarrollarse comienzan por definir objetivos y funciones que se requieren para tener SSI, esto comenzando por:

- Describir cuáles son los requerimientos para llevar a cabo un sistema distribuido de imagen única (SSI: Single System Image), y cuáles son los servicios esenciales que debe proveer para desarrollo y ejecución de programas paralelos y/o distribuidos. Asignar prioridad de acuerdo al tipo de aplicación y uso del sistema a los servicios necesarios.

- Clasificar las diferentes arquitecturas de las redes de computadoras como sistemas paralelos, en cuanto a caracterización del hardware y la arquitectura de las redes desde el punto de vista del desarrollo de aplicaciones paralelas y/o distribuidas teniendo una imagen única del sistema.

Luego es necesario evaluar y probar los desarrollos existentes:

- Analizar los diferentes modelos de programación paralela, básicamente los de memoria compartida y los de pasaje de mensajes y comparar sus características al implementarlos sobre un sistema multi-computador débilmente acoplado como lo son las redes de estaciones de trabajo.

- Investigar los ambientes de software, aplicaciones, middlewares y sistemas operativos que ofrezcan SSI parcial o total sobre una red de estaciones de trabajo. Comparar sistemas operativos tradicionales con los que ofrecen características de SSI. Estudiar posibles extensiones de los sistemas tradicionales.

- Comparar características de rendimiento, escalabilidad, transparencia, tolerancia a fallas, inter-operabilidad y seguridad en las implementaciones disponibles de SSIs anteriormente analizadas.

- Analizar arquitectura de software y modelos a partir del cual se desarrollan, (Modelos de Objetos, Modelos en Capas, Kernels Monolíticos, Microkernels, Nanokernels, etc.). Estudiar la factibilidad de mejorar software que ofrezca SSI en términos de nuevas prestaciones o prestaciones optimizadas en rendimiento.

- Evaluar y comparar los lenguajes y bibliotecas de software de uso gratuito más utilizadas para generar aplicaciones paralelas sobre redes tradicionales e identificar cuál es su aplicación sobre los sistemas de SSIs analizados.

Por último, luego de tener un panorama de lo que existe, analizar que desarrollos nuevos serían necesarios:

- Identificar la necesidad de herramientas para el desarrollo directo o para apoyo del desarrollo de aplicaciones paralelas sobre casos concretos (Cluster de 16 PCs del Laboratorio de Procesamiento Paralelo de la Facultad de Informática, o la Red Heterogénea de la Sala de PCs de la misma Facultad).

- Estudiar la posibilidad de extender el concepto de SSI al dominio de redes de área ancha (WANs) identificadas dentro de Grid Computing. Analizar cuáles son los inconvenientes que se agregan en este contexto y qué nuevos servicios se requieren.

- Investigar la posibilidad de extensión de algunos de los servicios de SSI de cobertura local a cobertura metropolitana o global.

Referencias

[MPI] MPI: The Complete Reference - Marc Snier, Steve Otto, Steven Huss-Lederman, David Walker and Jack Dongarra - The MIT Press, Cambridge, Massachusetts - 1996.

[PVM] PVM: Parallel Virtual Machine A Users'Guide and Tutorial for Networked Parallel Computing , Al Geist , Adam Beguelin , Jack Dongarra , Weicheng Jiang , Robert Manchek and Vaidy Sunderam - The MIT Press, Cambridge, Massachusetts - 1994.

[Tanen96] Sistemas Operativos Distribuidos, 1era. Edición. Anderew S. Tanenbaum. Prentice Hall Inc., 1996.

[Coul94] Distributed System Concepts and Design. 2da. Edición. George Couloris, Jean Dollimore y Tim Kindberg. Reading, MA: Addison-Wesley, 1994.

[Litz98] Condor: A Hunter of Idle Workstations. Litzkow, M, Livny, M. Y Mutka, M. Publicado en Proceedings of the 8th. International Conference on Distributed Computing Systems. Disponible on-lne en; http://www.cs.wisc.edu/condor/.

[Buyy01] Single System Image (SSI) . Rajkumar Buyya, Toni Cortes, Hai Jin. Publicado en The International Journal of High Performance Computing Applications, Vol 15, Nro. 2. Verano del 2001, pp. 124-135. Sage Publications, Inc.

[Barak98] The MOSIX Multicomputer Operating System for High Performance Cluster Computing. Barak, A. Y Laádan, O. Publicado en The Journal of Future Generation Computer Systems, 1998. Disponible on-line en http://www.mosix.cs.huji.ac.il.

[Ghorm98] GLUnix a Global Layer Unix for a Network of Workstations. Ghormley, D., Petrou, D., Rodrigues, S., Vahdat, A. y Anderson, T. Publicado en Journal of Software Practice and Experience, 1998. Disponible on-lne en http://now.cs.berkeley.edu/Glunix/glunix.html.

[Walker99b] Implementing a Full Single System Image UnixWare cluster: Middleware vs. Underware. Walker, B. y Steel, D. .Publicado en Proceedings of the International Conference on Parallel and Distributed Porcessing Techniques and Applications, 1999. Disponible on-lne en: http://www.sco.com/products/clustering/nscwhtpapers/.

[Zhang00] Linux Virtual Servers for Scalable Networks Services. Zhang, W. Ottawa Linux Symposium, 2000. Disponible on-line en http://www.LinuxVirtualServer.org.

[Beowulf] www.beowulf.org.

[IEEE-1003.1] IEEE Std. 1003.1: Information Technology-Portable Operating System Interface (POSIX)-Part 1: System Application: Program Interface (API). Institute of Electrical and Electronics Engineers , Inc. 1990.

[Kotz] Parallel I/O Archive, Dartmouth College http://www.cs.dartmouth.edu/pario/.

[Solaris-MC] Solaris-MC http://www.sunlabs.com/resarch/solaris-mc/.

[DSM MC] Distributed Shared Memory, Memory Channel http://www.digital.com/info/hpc/systems/symc.html.