Factibilidad y Rendimiento de las Comunicaciones para Cómputo Paralelo Intercluster

Fernando G. Tinetti*, Walter J. Aróztegui
III-LIDI, Facultad de Informática, UNLP
50 y 115, 1900, La Plata, Argentina
CeTAD, Facultad de Ingeniería, UNLP
48 y 116, 1900, La Plata, Argentina
fernando@info.unlp.edu.ar, waroz@graffiti.net

Abstract

This paper aims at solving the general communication problem among clusters used for parallel computing, taking as reference all what is known about parallel computing in clusters. In this sense, using more than one cluster for parallel computing can be considered as a natural extension of parallel processing in distributed computing platforms. We thus attempt to solve the problem of communicating processes in different computers that may belong to two or more clusters and, in addition, characterize their performance. Although communications among clusters may be considered as trivial due to the use of Internet, the situation becomes even more complex when the involved security characteristics are to be taken into account. In this sense, security problems are to be solved in a sustainable way so that these solutions can be applied into disjoint and collaborative management environments. From the performance point of view, the situation is rather more complex. Normally, communications among different clusters are shared with standard traffic of Internet of the involved institutions. Within this context, it is necessary to at least characterize the time intervals with higher congestion and / or the expected performance along the time the cluster interconnection is being used, with or without the use of some type of middleware which may solve the security problem among the involved networks.

Keywords: Process Communication, Performance Characterization, Communication Performance, Parallel and Distributed Systems, Parallelism in Clusters and Interclusters.

Resumen

Este trabajo se orienta a resolver el problema general de las comunicaciones entre clusters usados para cómputo paralelo, tomando como referencia todo lo conocido de cómputo paralelo en clusters. En este sentido, utilizar más de un cluster para cómputo paralelo se puede considerar como una extensión natural del procesamiento paralelo en plataformas de cómputo distribuidas. Se busca, por lo tanto, resolver el problema de comunicar procesos en diferentes computadoras que pueden pertenecer a dos o más clusters y, además, caracterizar el rendimiento de las mismas. Aunque las comunicaciones entre clusters pueden ser consideradas *triviales* con el uso de Internet, la situación se complica cuando se deben tener en cuenta las características de seguridad involucradas. En este sentido, se deben resolver los problemas de seguridad de manera sustentable en cuanto a que se puedan aplicar estas soluciones a ámbitos de administración disjuntos y colaborativos. Desde el punto de vista del rendimiento, la situación es bastante más complicada. Normalmente, las comunicaciones entre diferentes clusters son compartidas con tráfico estándar de Internet de las instituciones involucradas. En este contexto, es necesario por lo menos caracterizar los intervalos de tiempo de mayor congestión y/o el rendimiento esperable a lo largo del tiempo de uso de la interconexión entre los clusters, con y sin la utilización de algún tipo de *middleware* que resuelva la problemática de seguridad entre las redes intervinientes.

Palabras claves: Comunicación de Procesos, Caracterización de Rendimiento, Rendimiento de Comunicaciones, Sistemas Paralelos y Distribuidos, Paralelismo en Clusters e Intercluster.

* Investigador Asistente CICPBA

1 INTRODUCCION

Desde las primeras propuestas de uso de clusters para cómputo paralelo o al menos distribuido, ha quedado clara la importancia de las comunicaciones [5] [3]. Esta situación no es nueva, ya que es conocida en el ámbito de procesamiento paralelo clásico [1] [2]. Una de las primeras propuestas de uso libre para resolver las comunicaciones fue PVM (Parallel Virtual Machine) [5] y, de hecho, marcó muchas de las características de lo que luego se estandarizó como MPI (Message Passing Interface) [6]. Las primeras implementaciones de MPI de uso libre fueron LAM/MPI [7] [13] y MPICH [9] [10] que, por supuesto, implementan el estándar MPI y resuelven satisfactoriamente el problema de comunicar procesos que se ejecutan en diferentes computadoras de un cluster.

Una vez establecidas las bibliotecas precedentes, la tendencia ha sido y es caracterizar el rendimiento de las comunicaciones [4]. Más recientemente, con las múltiples propuestas de hardware de comunicaciones, la idea ha sido caracterizar el rendimiento de cada una de ellas para su comparación [12]. En el contexto de uso de más de un cluster para cómputo distribuido, todavía se está, de alguna manera, en la etapa de propuestas, como la de *grid* [8]. Sin embargo, al menos inicialmente, se descartan las herramientas de grid por varias razones. La razón más importante la constituyen los requerimientos actuales para el funcionamiento de hardware/software en grid, que son extremadamente altos en términos de interconexión de las computadoras que se utilizarán. Por otro lado, adoptar el contexto de grid todavía supone una restricción sobre la paralelización de aplicaciones, que ya cuenta con sus propias restricciones en clusters y con restricciones no totalmente definidas (ni cualitativa ni cuantitativamente) en el contexto de cómputo paralelo intercluster.

Sin llegar al ámbito más genérico de *grid computing*, la idea inicial es la de utilizar dos o más clusters para resolver un problema en paralelo [14]. En este sentido, se tienen dos líneas de acción que corresponden a dos ámbitos en principio diferentes pero necesarios para cómputo intercluster: seguridad (con la consecuente necesaria sustentabilidad de las soluciones técnicas propuestas/ adoptadas) y rendimiento caracterizable y, en el mejor de los casos, optimizado. De esta manera, para efectivizar cómputo paralelo intercluster se debe contar con:

- Una conexión viable entre los clusters, lo cual incluye una mínima sustentabilidad técnica de las comunicaciones TCP/IP, casi como en una red local. Esto implica analizar todo lo referente a la seguridad, al menos en términos de *firewalls* de protección entre las redes locales.
- Una caracterización mínima y confiable (al menos con cotas conocidas/determinadas) del rendimiento de las comunicaciones entre los clusters.

Con esto es posible comenzar a desarrollar/estudiar algoritmos y aplicaciones específicas, para aprovechar la capacidad de cómputo disponible entre los clusters. Aunque *a priori* pueda suponerse que serán necesarios nuevos algoritmos paralelos (y, de hecho, se pueden proponer y analizar), al llegar a este estudio en profundidad se necesita una caracterización cuantificada del rendimiento disponible para las transferencias de datos entre los clusters a utilizar. En este trabajo, la idea es proponer y llevar a cabo un conjunto mínimo de experimentos y con el mínimo de requerimientos tanto a nivel de hardware, software y ancho de banda a utilizar. A diferencia de la mayoría de los experimentos y estudios en el área de cómputo paralelo en clusters, no se puede asegurar la disponibilidad absoluta de la red de interconexión, dado que entre los clusters se utiliza la interconexión estándar y ya instalada para uso de Internet. Por lo tanto, no tendría sentido establecer como requerimiento que esta interconexión se utilice de manera excluyente, porque podría significar la exclusión (justamente) de posiblemente cientos de usuarios/computadoras que llevan a

cabo sus tareas usuales incluyendo tráfico en Internet. Más aún, es técnicamente bastante complicado establecer una determinada calidad de servicio entre los clusters, dado que las conexiones muy posiblemente están administradas por personal ajeno a cómputo paralelo y también ajeno a los problemas de rendimiento de los patrones de las comunicaciones para cómputo paralelo.

Por otro lado, dado que se tienen ámbitos de administración de las redes involucradas muy diferentes (posiblemente con muchos administradores de red involucrados implícitamente), también es importante tener una idea inicial de las características de conectividad entre los clusters. Se puede dar el caso de no tener disponible la conexión durante algunos períodos de tiempo y sería importante conocer cuáles, las razones, y si es posible evitar tales períodos de desconexión.

2 *MIDDLEWARES*: COMUNICACIÓN ENTRE PROCESOS DISTRIBUIDOS

En un entorno estándar de cómputo paralelo en un cluster, las herramientas también estándares son la utilización de alguna biblioteca de pasaje de mensajes tal como una implementación de MPI junto con los servicios básicos de disparo de procesos remotos: rsh o ssh. De hecho, cualquier instalación estándar de las implementaciones de MPI tales como MPICH y LAM/MPI utilizan el comando mpirun como herramienta para la ejecución de programas paralelos SPMD (Single program, Multiple Data) la cual, a su vez, utiliza el servicio rsh o ssh para el disparo de comandos en otras máquinas (o máquinas remotas).

En un cluster utilizado para cómputo paralelo se asume que no hay problemas internos de seguridad, sino que los problemas de seguridad (si existen) provienen del exterior, en el caso de que el cluster esté conectado a Internet, por ejemplo. Los sistemas de seguridad actuales consisten en la restricción casi sin discusión de todos o la mayoría de los servicios normalmente usados para interconexión de máquinas. También como parte del contexto actual de cómputo en clusters es cierto que los servicios llamados “seguros” son los que tienden a ser usados/permitidos, tales como los provistos por ssh y sus asociados scp y sftp. Aunque a priori este tipo de consideraciones (servicios, seguridad, etc.) no son parte del desarrollo software paralelo en clusters, aparecen como un problema agregado cuando se piensa en cómputo paralelo interclusters, considerando éste como el paso siguiente al cómputo paralelo en clusters tal como se viene desarrollando desde hace algunos años. Dado que es relativamente natural hoy pensar en distribuir el cómputo a realizar entre múltiples máquinas de un cluster o red local, también es posible pensar en distribuir el cómputo a realizar entre múltiples clusters disponibles. La Fig. 1 muestra el contexto más sencillo de aplicación, con la utilización de dos rede locales de computadoras para cómputo paralelo.

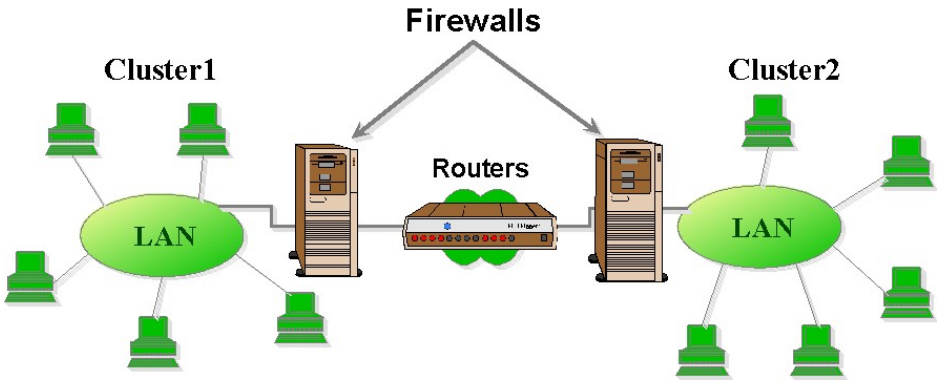

Figura 1: Interconexión de clusters.

Sin embargo, los servicios de disparo remoto no representan los únicos problemas de seguridad,

más específicamente, no solamente se deben disparar tareas remotas sino que, además, los procesos se comunican en tiempo de ejecución. De acuerdo con las implementaciones de las bibliotecas de pasaje de mensajes, estas comunicaciones se implementan usando los protocolos TCP y/o UDP sobre IP desde la interfase socket. En términos de seguridad y relacionado con los posibles firewalls de cada cluster, esto implica la comunicación utilizando posiblemente múltiples ports, normalmente no privilegiados (unpriviledged). Teniendo en cuenta las consideraciones anteriores y enfocando específicamente los problemas de seguridad y los controles impuestos vía firewalls, se deben identificar los puertos que utilizan las bibliotecas de pasaje de mensajes y sus entornos de ejecución. Si bien es muy difícil identificar claramente todos los detalles de las comunicaciones entre las computadoras, dado que se depende de las implementaciones (versiones, por ejemplo, de PVM o de las implementaciones de MPI), al menos se considera útil la identificación de cantidades y tipos de puertos (al menos en términos de privilegiados o no) a utilizar desde los programas paralelos con pasaje de mensajes.

En este sentido, un análisis efectuado sobre el funcionamiento de la versión LAM/MPI en la ejecución de programas muy sencillos proporciona información detallada sobre su comportamiento en la apertura de conexiones. Aunque las extrapolaciones son relativamente complejas, aún se pueden identificar similitudes y diferencias con el resto de las bibliotecas de pasaje de mensajes de uso libre disponibles actualmente (PVM y MPICH). De tal análisis se puede, en resumen, contabilizar la apertura de puertos por parte de la máquina que lanza las aplicaciones, de la siguiente manera:

- Lamboot (para el inicio de la *máquina paralela virtual*):
  - 3 puertos TCP de negociación para ssh y uno más al lanzar los procesos lamd por cada máquina a comunicar (con 3 máquinas en total, se tienen 2 máquinas con las que comunicar).
  - 2 puertos UDP por cada máquina a comunicar, que permanecerán abiertos hasta la ejecución del comando lamhalt (para la finalización de la *máquina paralela virtual*).

  Total de puertos utilizados durante lamboot: 6 puertos por máquina.
- mpirun (c2c/lamd, para la ejecución de una aplicación): sin contabilizar los puertos que se mantuvieron abiertos desde el lamboot, se abre 1 puerto TCP por cada máquina a comunicar.
- lamhalt: no abre puertos adicionales, solamente cierra los que utilizaban los lamd.

Es también importante notar que los únicos puertos privilegiados utilizados son los que corresponden al servidor ssh. Esto demuestra que desde la perspectiva de la sustentabilidad técnica de cómputo intercluster con bibliotecas MPI, la situación es bastante complicada. A menos que las políticas de seguridad (normalmente implementadas sobre los firewalls) se relajen mucho o aún se eliminen, sería virtualmente imposible utilizar una de estas bibliotecas para cómputo paralelo intercluster. Como mínimo, se debería permitir:

- El disparo de tareas remotas con ssh.
- La interconexión de procesos en cualquiera de las máquinas usando los protocolos UDP y TCP donde tanto cliente como servidores (desde la perspectiva de los protocolos, no de las aplicaciones paralelas de usuario) pueden estar en cualquiera de las computadoras.
- La utilización, en principio no controlada/limitada, de puertos no privilegiados para las conexiones entre clientes y servidores.

El único de los problemas ya ampliamente resuelto es el primero, el disparo de tareas remotas con ssh. Para los dos puntos restantes, sería necesaria la habilitación de todos los puertos no privilegiados para clientes y servidores de los protocolos TCP y UDP. Por un lado, esto implica abrir al exterior del cluster todos los puertos no privilegiados y por el otro, se debe asumir que todos los routers involucrados permiten el tráfico peer-to-peer, algo que todavía no está completamente

garantizado. Una alternativa para resolver este problema consiste en la utilización de middleware ya desarrollado adaptándolo a las características de cómputo intercluster (VPN: Virtual Private Network, por ejemplo) o de herramientas que están en desarrollo (IMPI: Interoperable MPI, por ejemplo [11]).

3 RENDIMIENTO DEL COMANDO ping DE LINUX

Tal como se adelantara en parte en la sección anterior, el entorno de utilización de cómputo paralelo intercluster puede ser muy variable. Se pueden dar casos donde los clusters no están totalmente disponibles para cómputo paralelo, la conexión no se tiene permanentemente y/o el rendimiento de las comunicaciones intercluster varía en el tiempo por no estar dedicada exclusivamente a cómputo paralelo, o a comunicar los clusters involucrados. Por estas razones, se recurre a un entorno de experimentación que, aunque específico, (es uno de muchos posibles), presenta varias de las características mencionadas previamente. Se tienen computadoras en dos laboratorios involucrados: III-LIDI (Fac. de Informática, UNLP) y CeTAD (Fac. de Ingeniería, UNLP). Cada uno de estos laboratorios tiene acceso a una red local que es la que se utilizará para cómputo paralelo. En cada una de las redes locales, se utilizará para los experimentos una sola computadora, dado que solamente se tienen que identificar las características de la interconexión entre las redes. Ambas redes locales son subredes de una misma red Internet B. Esto significa que, en realidad, no se llega a utilizar la salida a Internet de la institución a la que pertenece la red B (la UNLP, en este caso), sino que se comparte todo o parte del tráfico Internet dentro de la red. En el caso de la red local que se utiliza en el III-LIDI no se tiene acceso exclusivo y, de hecho, es una sala de computadoras que se utiliza para dar clases a alumnos varias veces en la semana. Esto involucra no solamente carga en las computadoras y en la red de interconexión sino también la posibilidad de que al terminar la clase el docente directamente apague todas las computadoras, incluyendo la que se necesita para llevar a cabo los experimentos. Las características más importantes de la interconexión son:

- Cada una de las redes locales pertenecen a una subred diferente.
- Hay más de cinco routers intermedios involucrados en la transferencia de datos.
- No se tiene acceso a la mayoría de los routers intermedios, ni siquiera se conoce qué interfase de red tiene cada uno de ellos.
- No se conocen las políticas de seguridad de cada uno de estos routers intermedios. En particular, no se sabe qué ports están filtrados por firewalls, o aún si tienen firewalls en funcionamiento (con reglas activas), por ejemplo.

Aunque el principal motivo para la experimentación alrededor de las comunicaciones entre clusters es el rendimiento, se deben tener en cuenta las características dinámicas de la interconexión, tal como se ha mencionado antes. En el entorno de un único cluster, también es importante caracterizar el rendimiento y esto usualmente se lleva a cabo obteniendo los valores de latencia y ancho de banda que caracterizan a las comunicaciones entre dos computadoras. En el entorno intercluster quizás se tengan valores mínimos y/o máximos, pero además es importante conocer con cierto detalle todas las demás características de la interconexión entre los clusters. Entre estas características se pueden mencionar:

- Fallas en la interconexión.
- Si existen períodos de mayor o menor disponibilidad de ancho de banda.
- La dependencia (si existe) de la latencia respecto del tráfico existente en la red de interconexión de los clusters.
- La existencia de filtros de seguridad entre los clusters, permanentes o en intervalos.

Y en función de estos objetivos o, más específicamente, para cuantificar estas características, se definen los experimentos a realizar.

El experimento más sencillo pero muy significativo en cuanto a la importancia de la información suministrada sigue siendo el ping-pong de mensajes. Para llevar a cabo este experimento se necesitan solamente dos procesos y, en el contexto de cómputo intercluster, dos computadoras. Es por esta razón que se utiliza solamente una computadora de cada red local utilizada. Esta estrategia/definición es general, es decir que aunque se utilicen más de dos clusters, la cantidad de máquinas a usar de cada cluster será una sola. De hecho, para los experimentos habrá a lo sumo solamente dos computadoras funcionando en total, dado que no tiene sentido llevar a cabo más de un experimento ping-pong simultáneamente.

Una vez definido que el experimento básico será el ping-pong de mensajes y elegidas las computadoras de cada cluster, hay varias alternativas para implementar el ping-pong. Dado que inicialmente no se tiene ninguna clase de información, el mismo comando ping de Linux será suficiente para recoger los datos para obtener la información preliminar de las comunicaciones entre los clusters. Es importante resaltar que aunque se hagan experimentos extensivos e intensivos, con muchas recolección de resultados con su consiguiente estabilidad estadística, la información seguirá siendo preliminar, sencillamente por la diferencia existente entre el transporte de los datos con el protocolo ICMP (Internet Control Management Protocol) y los protocolos y/o técnicas usados para la comunicación confiable entre procesos de una aplicación paralela. En realidad, con ICMP se estaría en la mejor situación de rendimiento y en una de las peores de confiabilidad, dado que los datos obtenidos surgirán de los experimentos satisfactorios, lo cual implica que no hay problemas de confiabilidad.

En este punto se tienen definidos los detalles más significativos de los experimentos, pero aún restan los parámetros de ejecución de los mismos. Estos parámetros están relacionados inicialmente con lo que se necesita para cuantificar/aproximar los dos índices básicos de las comunicaciones punto a punto: latencia y ancho de banda. Estos índices normalmente se utilizan en el modelo de tiempo de las comunicaciones dado por

$$t(n) = \alpha + \beta n \qquad (1)$$

donde α es el tiempo de latencia (o startup) y β es la inversa del ancho de banda, es decir el tiempo por ítem de datos a transferir y n es la cantidad de ítems o datos a transferir. Siguiendo la idea de simplificación de los experimentos, la latencia se puede estimar con mensajes de tamaño mínimo o cero si fuera posible. En realidad, cualquier valor menor de 10 bytes puede ser útil, ya que lo que se intenta estimar específicamente es:

- La sobrecarga de los protocolos o pila de protocolos impuesta por la implementación del sistema operativo utilizado.
- El tiempo de transporte físico mínimo, que implica también la interfase con el subsistema de I/O de cada computadora.

Y se debe recordar que en el contexto de cómputo paralelo en clusters es muy poco probable tener mensajes de menos de 1 KB entre procesos. Se debe notar aquí también que se está asumiendo que la sobrecarga de las bibliotecas o rutinas de comunicación de procesos de un programa paralelo es nula (lo cual es muy poco probable). Por otro lado, para la estimación del ancho de banda se pueden utilizar mensajes relativamente grandes, para los cuales se puede asumir que la mayor parte del tiempo de comunicaciones se utiliza para la transferencia física de los datos de un proceso a otro. En todos los casos, se deben tener suficientes datos para que la información tenga validez desde el

punto de vista estadístico. Este último punto es particularmente complicado de sostener y/o justificar en el contexto de cómputo intercluster donde, justamente, se intentan capturar cambios relativamente importantes en el rendimiento de las comunicaciones. Sin embargo, la idea es aquí que si hay cambios importantes se puedan cuantificar, caracterizar su probabilidad o relacionar con algún factor externo e independiente del cómputo paralelo que, por lo tanto, se desconoce.

Si bien es bastante sencillo definir tamaños de mensajes pequeños(el límite inferior es, claramente, cero) no es el caso para la definición de los mensajes grandes. Se debe tener en cuenta que las aplicaciones paralelas son muy variadas y de muy variados patrones de cómputo también. En este caso, se deben tener en cuenta dos puntos importantes que necesariamente restringen el tamaño de los datos a transferir en los experimentos:

- El comando ping normalmente establece un máximo en la cantidad de datos que se envían/reciben. Este tamaño está estrechamente relacionado con la simplicidad del comando y el protocolo utilizado(ICMP).
- Se utilizará una red de interconexión no exclusiva para cómputo paralelo y, por lo tanto, es deseable no inundar esta red con tráfico diferente del que la originó y mantiene su razón de existir: Internet. En este sentido, se tiene un problema relativamente importante por cuanto se debe usar la red para llevar a cabo cómputo paralelo pero se debe también dejar ancho de banda para las aplicaciones que normalmente corren usando estas interconexiones. Si bien es importante notar que el tráfico ICMP se descarta en caso de sobrecarga extrema de los routers, también es importante recordar que toda sobrecarga termina afectando de una manera u otra a las aplicaciones que usan la red.

Por lo tanto, los tamaños elegidos inicialmente son 8 bytes para los mensajes pequeños(con los cuales se tiene una idea de latencia o startup time) y 20000 bytes (aproximadamente 20 KB) para los mensajes grandes (con los cuales se tiene una idea del ancho de banda). Aunque estén definidos los tamaños de los mensajes, es necesario definir también la frecuencia con las que se los utilizará, para tener una mejor idea del ancho de banda que efectivamente podrán llegar a utilizar los experimentos. El mismo comando ping, por su propio funcionamiento, *a priori* define que se envía/recibe un mensaje por segundo. Esto, en principio, ya impone un límite máximo de tráfico. Sin embargo, se necesitan datos de los mensajes de las dos longitudes, es decir de 8 bytes y de 20KB. No tiene sentido usar dos comandos ping concurrentes porque puede llevar a errores de interferencia de las comunicaciones entre ellos dado que normalmente existe una sola placa de interfase de comunicaciones en cada una de las computadoras. Es así que la idea será usar un comando ping para mensajes de 8 bytes y luego, en secuencia, un comando ping para mensajes de 20 KB. Además, se utiliza la opción de *timeout* del comando ping porque se ha comprobado que cuando hay problemas de conectividad puede darse que el comando ping no termina. En este punto se tienen no solamente las definiciones más importantes de los experimentos sino también los detalles más específicos para utilizar el comando ping. En resumen, asumiendo que la red de interconexión es de 10 Mb/s (lo cual es bastante probable, por el uso de placas Ethernet de 10 Mb/s en los routers), se utiliza un poco más del 0.17% del ancho de banda disponible entre los clusters.

La Fig.2 muestra los resultados obtenidos de tiempos de ida y vuelta (rtt: round trip time) en milisegundos durante una semana de clases: Lunes 21 Nov. 2005 - Sábado 26 de Nov. 2005. Durante esta semana se tenía actividad normal tanto en los laboratorios involucrados (III-LIDI y CeTAD) como en la sala de computadoras que se utiliza desde el III-LIDI, como en la red de interconexión de los clusters. Estos tiempos corresponden a mensajes pequeños (de 8 bytes), con lo cual se utiliza para aproximar la latencia de la interconexión. La sala de computadoras que se utiliza desde el III-LIDI se comparte con horarios de clases, la computadora del CeTAD está completamente dedicada a los experimentos y las redes involucradas (tanto dentro de cada cluster

como entre los clusters) es compartida con todo el tráfico estándar de la red Internet B de la cual los clusters son subredes. En la misma figura se muestran algunos datos resumidos: en promedio, cuando la conexión no falló, el tiempo en una sola dirección (one way message time) es de 0.66 ms (Promedio), y la conexión estuvo disponible el 63% del tiempo durante el cual se tomaron las muestras (Disponible). Los espacios en los cuales no se muestran resultados son los correspondientes a los períodos en los cuales los paquetes ICMP no tuvieron respuesta. Estos períodos de pérdida de conexión, por otro lado, se corresponden directamente con el apagado de la computadora no disponible del lado del III-LIDI, no con la caída de enlaces/routers.

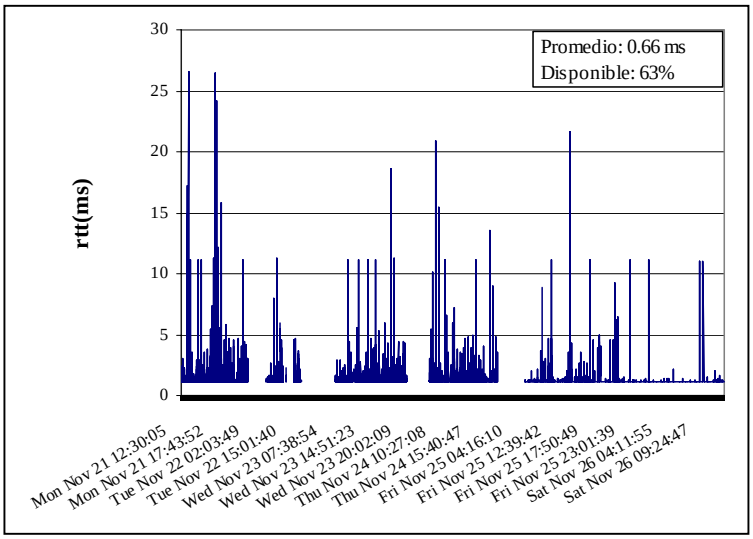

Figura 2: Latencia en una Semana de Actividad Normal.

Respecto del tiempo de latencia que se quiere aproximar/cuantificar, se debe recordar que una interconexión TCP en una red local Ethernet de 100 Mb/s tiene una latencia de aproximadamente 0.052 ms. Aunque a priori se podría suponer que la conexión no es confiable, en realidad los mayores problemas de conectividad surgen de la falta de control sobre las máquinas involucradas. De hecho los mayores períodos en los cuales los comandos ping no se completan satisfactoriamente (con un ICMP echo reply) se dan por las noches y/o a partir de la finalización de una clase. Siguiendo con este punto, a partir de experimentos que se llevaron a cabo durante todo casi el mes de Noviembre de 2005 solamente hubo una desconexión que no se debió a este tipo de problemas de falta de control sobre las máquinas involucradas: durante un fin de semana no hubo energía eléctrica y/o falló intermitentemente durante la mayor parte del fin de semana. Todos los demás inconvenientes para las respuestas a los comandos ping se debieron al apagado de la computadora que se accede del lado del III-LIDI.

La Fig.3 muestra los resultados obtenidos de ancho de banda (para paquetes ICMP de 20 KB) en bytes/segundo durante la misma semana de la figura anterior: una semana de clases (Lunes 21 Nov. 2005 - Sábado 26 de Nov. 2005). Está claro que estos experimentos con mensajes grandes tuvieron los mismos problemas de conectividad que los experimentos con mensajes pequeños. Por lo tanto, el porcentaje de disponibilidad de la conexión es el mismo: 63% del tiempo de los experimentos. Aunque la impresión visual de la Fig.3 pueda indicar otra cosa, el promedio de ancho de banda es muy bueno teniendo en cuenta que la red física tiene la capacidad de 10 Mb/s. Sin embargo, en términos relativos con lo que sucede en cada uno de los clusters, lo mínimo que se suele tener disponible es 100 Mb/s, lo cual en teoría podría transferir datos a 12.5 MB/s y, más realista, 10 MB/s. En este sentido, el ancho de banda es de menos del 10% de lo que se tiene en cada uno de los

clusters. Por otro lado, se debe recordar que la aproximación del ancho de banda con paquetes ICMP de 20 KB deja de lado muchos detalles propios de las conexiones entre procesos de una aplicación paralela que normalmente involucran sobrecarga que no se tiene en cuenta o no se puede cuantificar de esta manera.

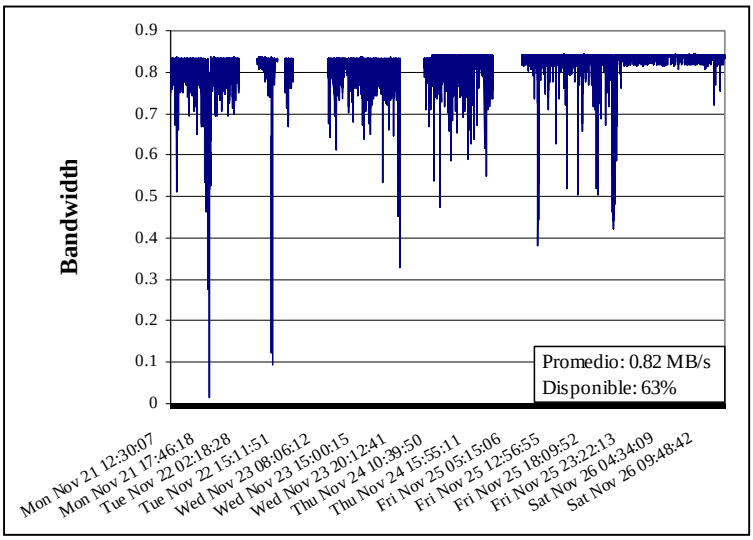

Figura 3: Ancho de Banda de Actividad Normal.

Quizás a modo de resumen/conclusión de los resultados que se han mostrado hasta aquí, tanto la latencia como el ancho de banda de la conexión intercluster tienen más de un orden de magnitud de penalización sobre los valores que se pueden obtener en una red local con TCP. Volviendo al contexto de cómputo paralelo, esto también implica serias restricciones en términos de rendimiento. Aunque la idea de escalar las aplicaciones siempre ha sido muy apropiada para el rendimiento de cómputo paralelo (en el área de cómputo numérico intensivo, al menos), esta idea no parecería ser aplicable de manera inmediata en el contexto de cómputo intercluster. Es claro que tanto los cambios a nivel de latencia como de ancho de banda con las conexiones intercluster indican que las aplicaciones paralelas tendrán que, de alguna manera, adaptarse al rendimiento disponible de la red de interconexión. Lo que parece ser otro problema a resolver es el de la disponibilidad de las computadoras, más que el de la disponibilidad de la interconexión. Claramente, las interconexiones dedicadas al tráfico Internet son compartidas pero, también, tienden a ser estables en cuanto a disponibilidad. No es lo que sucede en términos de las computadoras, donde *compartidas*, tal como lo muestran los experimentos, suele ser sinónimo de *no disponibles*.

4 RENDIMIENTO PRELIMINAR CON MPI EN UNA VPN

Desde una perspectiva general y por su propia definición, VPN (Virtual Private Network) provee una forma de comunicar las computadoras en dos o más redes locales como si estuvieran en una única red local. Esto en sí mismo puede ser considerado útil para cómputo paralelo intercluster y sin embargo también provee una forma de multiplexado de todas las comunicaciones interclusters a través de una única interconexión física entre los clusters. Es decir: aunque haya varias máquinas de uno de los clusters transfiriendo datos con varias máquinas de otro cluster, todas las transferencias se llevarán a cabo utilizando esa única interconexión, que además es conocida *a priori* y por lo tanto puede ser mantenida a nivel de seguridad de los firewalls. Esta relativa solución de los problemas de seguridad en cuanto a la apertura de puertos y la restricción sobre éstos que pudieran

tener los firewalls intermedios entre las distintas redes que conforman el intercluster y que se mencionaron en una sección anterior, también acarrea una disminución en el rendimiento, tanto con el aumento de la latencia como con la disminución del ancho de banda. Tal situación define una nueva serie de experimentos en la caracterización de las comunicaciones entre los clusters, de la manera en que se hizo antes. Con la inclusión de VPN, la apertura de puertos por parte de las librerías MPI y el posible filtrado que efectúan los firewalls, pueden "manejarse" de manera más simple pues ahora sólo se tiene una conexión entre servidor y clientes dentro de la VPN y sólo existe la necesidad de habilitación del puerto correspondiente al servidor, previamente definido en la configuración de éste. Se puede ahora, para caracterizar las comunicaciones entre los clusters (intercluster), experimentar con ping-pong dentro de un entorno MPI, encausado a su vez, por medio de una VPN.

La Fig. 4 muestra los resultados de latencia (con mensajes de 8 bytes) obtenidos durante un período relativamente corto de tiempo en un día de actividad normal con:

- La VPN se puso en funcionamiento con OpenVPN 2.0 entre las dos computadoras, cada una en un cluster distinto.
- El período de tiempo es de los de uso más alto para tráfico de Internet, entre las 9:30 y las 10:00 de la mañana. No se tiene ningún tipo de control ni de información sobre este tráfico.
- Se usó LAM/MPI sobre la VPN, de manera tal que en este caso sí se tiene en cuenta la sobrecarga de una biblioteca para cómputo paralelo.
- No hay más tráfico intercluster que el generado por el ping-pong de MPI con longitud de mensajes de 8 bytes.

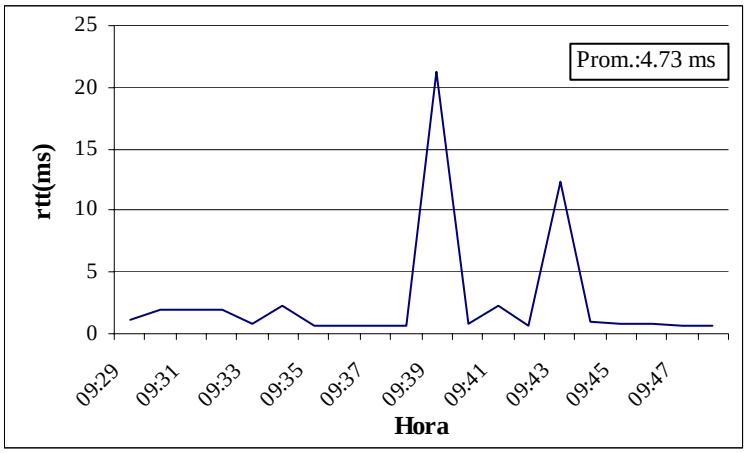

Figura 4: Latencia con MPI sobre VPN en un Período Corto de Actividad Normal.

Si bien La Fig. 4 no es directamente comparable con la Fig. 2, sí se puede estimar que el mayor promedio de tiempo de latencia se debe en parte a la sobrecarga impuesta por el software de VPN más el de la biblioteca de cómputo paralelo. Por otro lado, se debe recordar que MPI sobre VPN sí es seguro en el sentido de que se recupera de errores mientras que en ICMP simplemente se *descartan* los errores. Con respecto a los errores o a la falta de interconexión se han detectado inconvenientes relativamente serios en relación con el ruteo de los paquetes IP entre las dos redes locales. Según experimentos preliminares, se han detectado diferentes rutas entre ambas redes, donde una de estas rutas no funciona adecuadamente y esto produce que la VPN (o el software que la maneja, en realidad) deje de funcionar también, produciéndose un error del que no se puede recuperar (a pesar de usar TCP como protocolo de transporte). Se está trabajando actualmente para resolver este problema.

A pesar de que en el contexto de MPI sobre una VPN no hay límite para la longitud máxima de los mensajes (al menos en teoría no se lo tiene para las implementaciones de MPI tales como LAM/MPI), de todas maneras se mantiene en 20 KB la longitud de los mensajes con los cuales se aproxima el ancho de banda. La Fig. 5 muestra los resultados para el mismo período de tiempo que se da en la Fig. 4, pero ahora en términos de ancho de banda estimado a partir de mensajes de 20 KB. Es muy interesante que el ancho de banda obtenido es relativamente alto a pesar de que ahora se tiene toda la sobrecarga y, además, se comparte la interconexión con tráfico relativo a Internet de otras computadoras de la red.

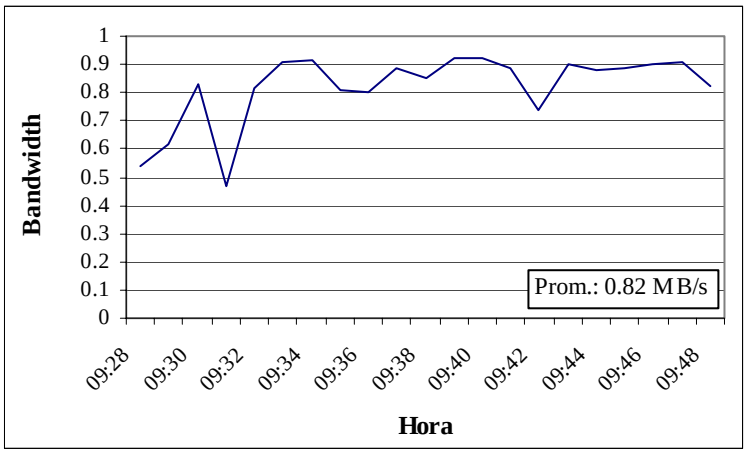

Figura 4: Ancho de Banda con MPI sobre VPN en un Período Corto de Actividad Normal.

5 CONCLUSIONES

Una de las primeras conclusiones es la necesidad de *algo más* que una biblioteca de cómputo paralelo para cómputo intercluster. Una de las más sencillas consiste en la construcción de una VPN, que ha sido probada en este trabajo. Sin embargo, no está exenta de errores y se está trabajando actualmente en la resolución de los mismos. Aún sin tener en cuenta la sobrecarga de un *middleware* para cómputo paralelo intercluster, queda claro a partir de los experimentos usando uno de los protocolos más sencillos (ICMP, con el comando ping), que la diferencia entre el rendimiento intracluster e intercluster es al menos de un orden de magnitud. Esto necesariamente atenta contra el uso de algoritmos paralelos que no estén específicamente preparados para cómputo paralelo en estas arquitecturas distribuidas.

Se está trabajando para resolver los problemas de conectividad entre las dos redes locales en las que se hicieron los experimentos y, además, en experimentos más extensos y representativos para la caracterización de la interconexión. Esto permitirá avanzar tanto en la confiabilidad de la/s herramienta/s como en la propuesta de estudio de nuevos algoritmos paralelos específicos u optimizaciones de los existentes.

REFERENCIAS

[1] Akl S., The Design and Analysis of Parallel Algorithms, Prentice-Hall, Inc., 1989.

[2] Akl S., Parallel Computation: Models and Methods, Prentice-Hall, Upple Saddle River, 1997.

[3] Anderson T., D. Culler, D. Patterson, and the NOW Team, “A Case for Networks of Workstations: NOW”, IEEE Micro, Feb. 1995.

[4] S. Araki, A. Bilas, C. Dubnicki, J. Edler, K. Konishi, and J. Philbin, “User-space communication: A quantitative study”, In *SC98: High Performance Net-working and Computing*, November 1998.

[5] Dongarra J., A. Geist, R. Manchek, V. Sunderam, Integrated pvm framework sup-ports heterogeneous network computing, Computers in Physics, (7) 2, pp. 166-175, April 1993.

[6] MPI Forum, “MPI: a message-passing interface standard”, International Journal of Supercomputer Applications, 8 (3/4), pp. 165-416, 1994.

[7] Burns G., R. Daoud, J. Vaigl, “LAM: An Open Cluster Environment for MPI", Proceedings of Supercomputing Symposium, pp. 379-386, 1994. Available at http://www.lammpi.org/download/files/lam-papers.tar.gz

[8] Foster I., The Grid: Blueprint for a New Computing Infrastructure, 2nd Edition, Morgan Kaufmann, 2004. ISBN: 1-55860-933-4.

[9] Gropp W., E. Lusk, “Sowing MPICH: A Case Study in the Dissemination of a Portable Environment for Parallel Scientific Computing", The International Journal of Supercomputer Applications and High Performance Computing, Vol. 11, No. 2, pp. 103-114, Summer 1997,

[10] Gropp W., E. Lusk, N. Doss, A. Skjellum, “A high-performance, portable implementation of the MPI message pas-sing interface standard", Parallel Com-puting, Vol. 22, No. 6, pp. 789-828, Sep. 1996.

[11] IMPI Steering Committee, IMPI - Interoperable Message-Passing Interface DRAFT March 22, (NIST) National Ins-titute of Standards and Technology, 1999. Disponible en: http://impi.nist.gov/.

[12] Liu J., B. Chandrasekaran, W. Yu, J. Wu, D. Buntinas, S. Kini, P. Wyckoff, and D. K. Panda, “Micro- Benchmark Performance Comparison of High-Speed Cluster Interconnects”, IEEE Micro, January/ February, 2004.

[13] Squyres J. M., A. Lumsdaine, “A Component Architecture for LAM/MPI", Proceedings, 10th European PVM/MPI Users' Group Meeting, pp. 379-387, 2003, Venice, Italy, Springer-Verlag Lecture Notes in Computer Science 2840, September/ October 2003.

[14] Tinetti F. G., Aróztegui W., “Bibliotecas de Pasaje de Mensajes y Cómputo Intercluster”, Reporte Técnico PLA-003-2005, Septiembre 2005. Disponible en https://lidi.info.unlp.edu.ar/~fernando/publis/portsrep.pdf