Análisis del Rendimiento de las Comunicaciones sobre NOWs

Fernando G. Tinetti*
Andrés Barbieri†
LIDI‡ CeTAD§

Resumen

Las redes de interconexión son un pilar fundamental en el campo de las arquitecturas paralelas. Dentro de las arquitecturas paralelas, las redes de estaciones de trabajo (NOW: Networks of Workstations) demuestran tener una relación costo/rendimiento entre tres y diez veces mejor que una supercomputadora tradicional.

Esto se puede observar, por ejemplo, en las instalaciones relacionadas con el proyecto *Beowulf*, donde se logran muy buenos resultados en términos de rendimiento para cálculos complejos y grandes volúmenes de datos.

A pesar de todas las publicaciones y reportes en esta área, aún quedan cuestiones sin resolver, tal como la utilización de ambientes *"Heterogeneos"* -redes de computadoras donde las estaciones de trabajo tienen diferentes cualidades técnicas- como arquitectura subyacente.

La idea principal de este artículo es estudiar las comunicaciones en estos entornos, detectar los problemas y proponer cómo solucionarlos para alcanzar el rendimiento ideal o lo más cercano al óptimo posible en el contexto de las aplicaciones paralelas.

Palabras Clave: Computación Paralela, Computación sobre Clusters, Rendimiento de Comunicaciones, Rendimiento, Ambiente Heterogéneo

1 Introducción

Las redes de estaciones de trabajo(NOWs) han demostrado ser la arquitectura paralela con más bajo costo y más alta escalabilidad [1]. En la actualidad se dispone de software gratuito tal como PVM [2] [5] y/o implementaciones de MPI [4] [6] para su utilización. Este tipo de beneficios no necesariamente tienen un "costo cero", ya que para alcanzar resultados satisfactorios y no perder de vista el objetivo principal del cómputo paralelo (rendimiento), se debe hacer una sintonización y balance teniendo en cuenta tanto el software como el hardware. Estos problemas parecen potenciarse más aún cuando no se dispone de máquinas con idénticas características.

Cuando se hace referencia a una máquina paralela, en general están implícitos dos aspectos que la constituyen:

*fernando@ada.info.unlp.edu.ar
†barbieri@lidi.info.unlp.edu.ar
‡Calle 50 y 115 - Universidad Nacional de La Plata - (1900) La Plata, Argentina
§Calle 48 y 116 - Universidad Nacional de La Plata - (1900) La Plata, Argentina

- comunicaciones
- elementos de cómputo (procesamiento)

A estos elementos está ligado el rendimiento que se obtiene cuando se ejecutan algoritmos adaptados a la arquitectura, y éstos son sobre los cuales se debe trabajar para lograr el mejor rendimiento.

Dentro de las redes de estaciones de trabajo, el punto más cuestionado es el de las comunicaciones debido a que las redes no fueron diseñadas para el procesamiento paralelo, y, salvo muy pocas excepciones, típicamente la latencia es muy alta y el ancho de banda relativamente bajo comparado con hardware concebido para cómputo paralelo. Es por esto que se torna muy importante evaluar su desenvolvimiento desde el punto de vista de los procesos de usuario que componen una aplicación paralela. El rendimiento de la red de interconexión además tiene relación directa con la granularidad de los programas que se pueden ejecutar sobre la máquina, toda operación que implique comunicarse con procesos residentes en otra computadora tiende a degradar el tiempo total, a menos que se aproveche al máximo la capacidad de solapar cálculo con comunicación/es.

Otro inconveniente que se debe enfrentar es el de la sobrecarga (*overhead*) de las librerías para cómputo paralelo sobre NOW. Por un lado son flexibles y robustas para ejecutar en sistemas muy dispares pero es muy posible que no aprovechen todos los recursos y generen resultados no esperados cuando son usadas sobre una NOW con máquinas heterogéneas.

2 Caracterización del Subsistema de Comunicaciones

Si bien en el ámbito de las NOWs el hardware de cómputo es bastante heterogéneo, por el lado de las comunicaciones pasa lo contrario, la norma más usada en las redes locales o LANs (Local Area Networks) es la *802.3*[8] estandarizada por *IEEE* y más conocida como *Ethernet* la cual abarca del 80 al 90 por ciento de las instalaciones. Las propiedades de estas redes son bien conocidas en términos de rendimiento y flexibilidad.

La capacidad de transmisión base es de 10Mb/s y existen en la actualidad normas similares y compatibles con la anterior que son de 100Mb/s. Está definida sobre diferentes medios como cable coaxil, par trenzado (blindado y no blindado), y fibra óptica. El mecanismo de conexión utilizado es un bus lógico por el que se transmiten los datos, el que es compartido por todos los nodos, lo que implica competir con el resto al momento de enviar información. Gradualmente, el cableado (wiring rule) tiende a ser UTP (unshielded twisted pair) con hubs o switches de 100 Mb/s.

En general sobre cualquier LAN, el subsistema de comunicaciones es inherentemente homogéneo debido a que todos deben usar el mismo medio para comunicarse. Existen algunas variantes en las cuales esto no sucede, pero son casos muy aislados. Aunque no son relevantes desde el punto de vista de la cantidad de redes locales instaladas, se pueden mencionar algunos ejemplos tales como:

- Un segmento conectado mediante un hub 10BaseT y un "uplink" hacia un switch 100BaseTX. En este caso, la conexión entre máquinas conectadas al switch con placas de red 100BaseTX será de 100Mb/s en Half-Duplex y de 200Mb/s en Full-Duplex, en cambio entre una máquina conectada al primer hub y cualquier otra será de 10Mb/s como máximo.

- Otro ejemplo puede ser el de dos segmentos de redes con protocolos diferentes, por ejemplo Ethernet y Token Ring, interconectados por un "bridge".

3 Factores que Afectan el Rendimiento

Los factores que determinan el tiempo de comunicación de las transmisiones entre dos computadoras o procesadores de una máquina paralela son básicamente dos:

Ancho de Banda: se define como la máxima cantidad de datos que pueden ser transmitidos por unidad de tiempo.

Latencia: es el mínimo tiempo requerido para transmitir cualquier información, incluyendo cualquier overhead de send/receive por software. Este determina a su vez la granularidad útil mínima.

En función de estos dos índices, el tiempo de una transmisión de datos entre dos computadoras o procesadores de una máquina paralela se puede calcular con la siguiente fórmula:

$$t(n) = \alpha + \beta n \tag{1}$$

donde α es la latencia o también conocida como *startup*, β el ancho de banda y n es la cantidad de unidades de datos a transmitir.

Por lo tanto, se podría calcular *a priori* el tiempo necesairio para enviar un mensaje: el ancho de banda sería de 10-100Mb/s y la latencia se puede estimar con el tiempo que toma transferir una unidad de información, o en el caso en que sea posible, usar un mensaje sin datos. Es necesario mencionar que tanto el ancho de banda como el tiempo de startup son válidos para una red sin tráfico, pues el medio es compartido y si existe otro tipo de información siendo transferida, los mensajes deben competir y los valores cambian. Básicamente, la información sobre la red local se puede clasificar en:

- Datos de *web browsers* (interactivos)
- Datos de servicios de intranet (internos a la red, ftp, telnet, rsh, etc.)
- Datos de servicios de Internet (http, mail, news, etc.)

El tráfico generado por los *web browsers* así como los de servicios de intranet prácticamente desaparecen cuando no hay usuarios en la red, por ejemplo en las noches. El tráfico generado por los servicios de Internet es muy difícil de anular, pero en general está sujeto a un grupo de estaciones de trabajo específicas (y muchas veces aisladas) por razones de seguridad y mantenimiento.

Otro factor que afecta el rendimiento es el alto startup que se tiene a nivel de hardware comparado con el de las computadoras paralelas tradicionales; el cual se hace cada vez menos importante a medida que se manejan mayores volúmenes de datos. Si bien la cantidad de datos a transmitir es dependiente de la aplicación y de la granularidad de la misma, es claro que para lograr buen rendimiento en un cluster la granularidad debe ser suficientemente grande.

Se ha experimentado que en general el usuario suele obtener valores peores que los *determinados* por el hardware. La latencia se ve muy afectada por las capas de comunicación necesarias para que un mensaje pase del espacio de usuario de una máquina al espacio de usuario en otra máquina de la red. Es también difícil poder estimar la sobrecarga de las librerías de comunicaciones que deben usar los procesos y la interfaces entre éstas y el sistema operativo. Es necesario recordar que en el contexto de hardware heterogéneo la sobrecarga se hace más importante al evaluar el rendimiento de la red. Por estas razones es que se imponen los métodos experimentales para obtener los verdaderos valores de α y β.

El modelo detallado está basado en comunicación punto a punto. El rendimiento de las comunicaciones colectivas es también un tema pendiente, ya que el uso de esta funcionalidad es necesario para un gran número de aplicaciones [7], y si no es correctamente implementada, los resultados obtenidos son bastante lejanos a lo esperado.

4 Experimentación

Toda la experimentación se orienta a la obtención de los valores *reales* de α y β, donde el conjunto de estaciones de trabajo utilizadas es heterogéneo:

Nombre	Descripción	Sistema Operativo	Reloj	Memoria
purmamarca	PC-PII	Linux 2.2.13	400MHz	64 MB
Josrap	PC-K6 2	Linux 2.2.13	450MHz	62 MB
tilcara	PC-Pentium	Linux 2.2.5-15	133MHz	16 MB
sofia	IBM RS6k PPC604e	AIX 4.3	200MHz	64 MB
paris	Sun SPARC Station 4	Solaris 2	110MHz	96 MB
cetad	Sun SPARC Station 5	SunOS 4.1.4	85MHz	96 MB
prited	Sun SPARC Station 2	SunOS 4.1.3	40MHz	32 MB
cf1	PC-Celeron	Linux 2.2.5-15	300MHz	32 MB
ileana	PC-K6 2	Linux 2.2.13	266MHz	64 MB

La LAN usada es de 10Mb/s y el cableado es UTP con 4 hubs. Los tiempos de las comunicaciones punto a punto fueron evaluados con las siguientes características:

- El método de evaluación usado fue el envío de los mensajes *ping-pong*: una tarea envía datos a otra y ésta le contesta con el mismo mensaje. El tiempo se calcula en base al *Round Trip Time*, dividiendo por dos el tiempo total.
- Las longitudes de los mensajes fueron seleccionadas entre 8B y 10MB.
- También se utlizó el comando *ping*, a nivel de sistema operativo enviando paquetes ICMP (Internet Control Message Protocol) para medir el rendimiento de la red con *overhead* mínimo.
- La estación de trabajo purmamarca fue seleccionada para calcular y almacenar los tiempos debido a que tiene memoria suficiente para todos los tamaños de mensajes y es la más rápida. Por lo tanto, esta PC envía el *ping* inicial a otra máquina de la cual recibe el correspondiente *pong*.
- Las mediciones se realizaron con la red libre de interferencia (sin tráfico extra).
- La librería de comunicaciones utilizada fue PVM, que está basada en mensajes no bloqueantes, y que además tiene dos niveles de fexibilidad: codificación y ruteo.

4.1 Experimentación con PVM

En un ambiente heterogéneo, la codificación de los datos es fundamental para conservar la consistencia de los datos en cuanto a las distintas formas de representación que tiene cada estación de trabajo. Las variantes con respecto a la codificación de mensajes de PVM son:

- *PvmDataDefault*: los datos son codificados utilizando el formato *XDR* [9].
- *PvmDataRaw*: los datos no son codificados y sólo es útil cuando la arquitectura de la máquina paralela es homogénea.
- *PvmDataInPlace*: es similar a *PvmDataRaw* pero sin copiar los datos del área de usuario a los *buffers* de PVM.

En la Fig. 1 se muestran en forma esquemática cada una de las tres alternativas.

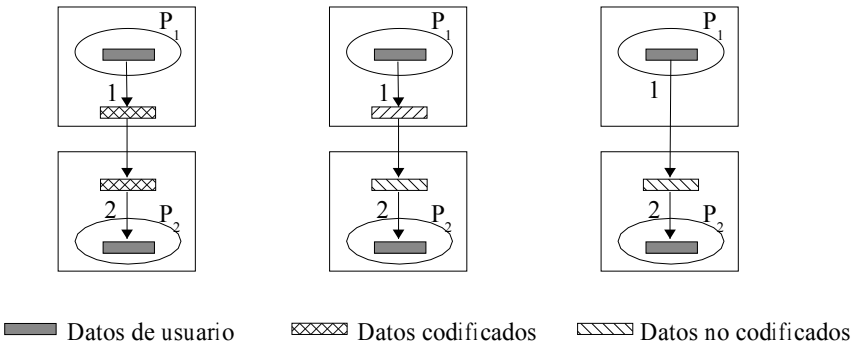

Figura 1: Alternativas de Codificación en PVM.

Debido a que en las estaciones de trabajo se encuentra una sorprendente adhesión al estándar *IEEE 754*[3] para la representación de números en punto flotante, por ejemplo, esto se puede aprovechar para reducir los tiempos dedicados a mantener la consistencia de datos. Por lo tanto, se implementa también el método de *Traducción* que implica realizar una transformación de los datos en la computadora destino de un mensaje siempre y cuando sea necesaria. De hecho, el único problema de codificación que se tiene entre las diferentes plataformas utilizadas es el de la representación *little endian* y *big endian*. En este caso particular, los mensajes se pueden enviar sin codificar y luego en el receptor se determina si se debe pasar de una representación a otra (*big* a *little* o *little* a *big*) de acuerdo a la representación origen. Este mismo análisis realizado con los números en punto flotante puede extenderse a otros tipos como enteros, enteros dobles, etc.

La forma en que se rutean los datos de los mensajes de PVM se puede determinar desde las aplicaciones de usuario, y las alternativas disponibles son:

- *PvmRouteDefault*: los datos son transferidos entre *pvmds* sobre la red utilizando el protocolo *UDP/IP*. Toda transferencia de datos entre distintas estaciones de trabajo se realiza entre los procesos *pvmd*. Cada proceso *pvmd* administra *todas* las comunicaciones de los procesos PVM locales (en la misma estación de trabajo).
- *PvmRouteDirect*: los datos son transferidos directamente entre procesos de usuarios sobre la red y se utiliza el protocolo *TCP/IP*. De esta manera, las rutinas de PVM manejan directamente las comunicaciones sin intervención de los procesos *pvmd*. En este caso, tanto las transferencias locales como con otras estaciones de trabajo se resuelven utilizando *solamente* las rutinas de PVM.

En la Fig. 2 se muestran esquemáticamente ambas formas de ruteo de los datos.

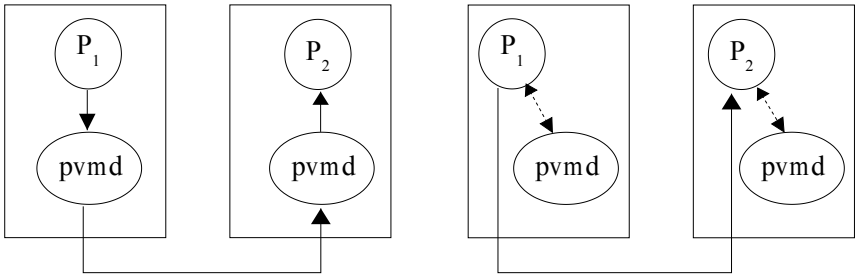

Figura 2: Alternativas de ruteo en PVM.

4.2 Resultados Obtenidos con PVM

El rendimiento obtenido comunicando tareas de PVM se midió como resultado de las siguientes combinaciones de alternativas de codificación (o traducción) y ruteo mencionadas y explicadas previamente.

- *PvmRouteDefault + PvmDataDefault*
- *PvmRouteDefault + Traducción*
- *PvmRouteDirect + PvmDataDefault*
- *PvmRouteDirect + Traducción*

Los gráficos de las subsecciones siguientes se muestran el rendimiento obtenido en términos de tiempo de transferencia de datos y de cantidad de datos por unidad de tiempo: MB/s.

4.2.1 Rendimiento con Ruteo entre *pvmds* y con Codificación

Esta alternativa es la menos conveniente en cuanto al rendimiento de la red. El tiempo de latencia varía considerablemente, con valores que van entre 1 y 10 ms dependiendo de la computadora, de acuerdo a la Fig. 3. Para una misma longitud de mensajes, cada estación de trabajo tiene un tiempo de comunicación diferente del resto, aunque la escala logarítmica de la figura tiende a enmascarar estas diferencias.
En la Fig. 4 se muestran los mismos resultados pero en MB/s, donde se pueden visualizar mejor las diferencias relativas entre computadoras. Además, como era de esperar, a mayor longitud de mensaje se obtiene mejor rendimiento.

En algunos casos se puede observar un comportamiento anómalo para los mensajes de 10MB, pero es explicable en función de los tamaños de memoria principal y la memoria intermedia (*buffers*) que utiliza PVM para las comunicaciones. Otras conclusiones inmediatas pueden ser:

- Se tienen claras diferencias de rendimiento de las distintas estaciones de trabajo, los valores están comprendidos entre un poco más de 0.4 y 0.8 MB/s.
- El mejor rendimiento obtenido es de 0.8 MB/s para tres estaciones de trabajo.
- El mejor rendimiento se logra con mensajes del orden de 10^4 bytes.

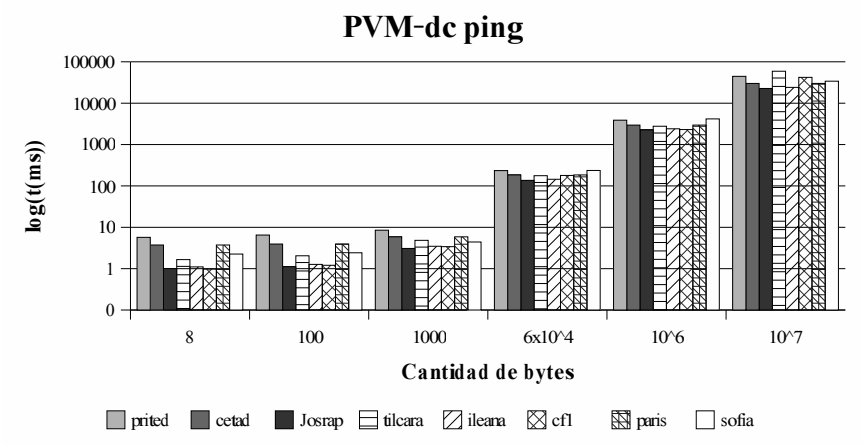

Figura 3: Tiempos con ruteo y codificación *PvmDefault*.

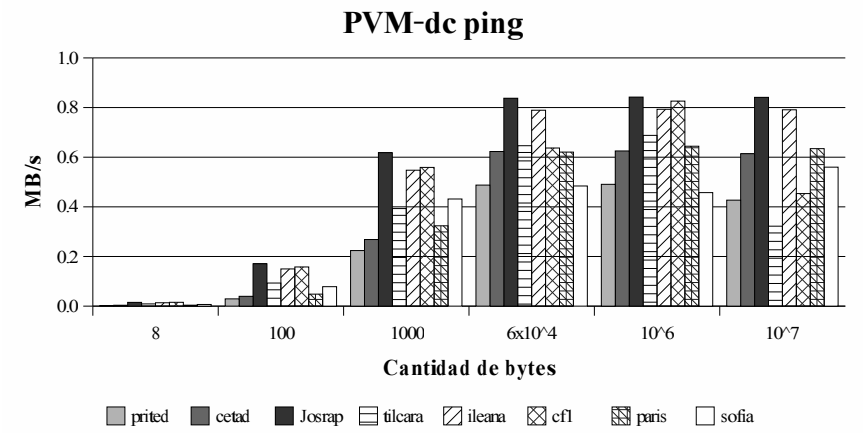

Figura 4: MB/s con ruteo y codificación *PvmDefault*.

4.2.2 Rendimiento con Ruteo entre *pvmds* y con Traducción

Esta alternativa para el manejo de los mensajes es significativamente similar a la anterior, dando una idea clara de la importancia de la transmissión sobre la codificación o manejo de la consistencia de los datos. La Fig. 5 muestra los resultados obtenidos en términos de tiempo transcurrido y la Fig. 6 en términos de MB/s.

Comparando los resultados de la Fig. 4 con la Fig. 6 se puede notar que se mejora el rendimiento en un pequeño porcentaje, y esto se debe a:

- Las máquinas con igual representación de datos que la que envía no deben utilizar tiempo de ejecución para mantener la consistencia de los mismos. La estación de trabajo que envía es purmamarca y las que tienen igual representación de datos son: Josrap, tilcara, ileana y cf1.

- Al utilizar menos memoria intermedia (*buffers* de PVM), las computadoras con menos memoria mejoran su rendimiento. Las estaciones de trabajo con menos memoria son, en este caso: prited (32 MB), tilcara (16 MB) y cf1 (32 MB).

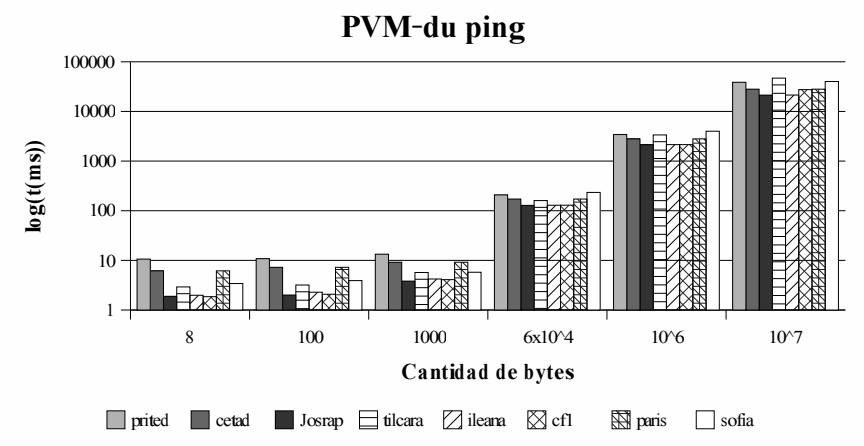

Figura 5: Tiempos con ruteo *PvmDefault* y traducción.

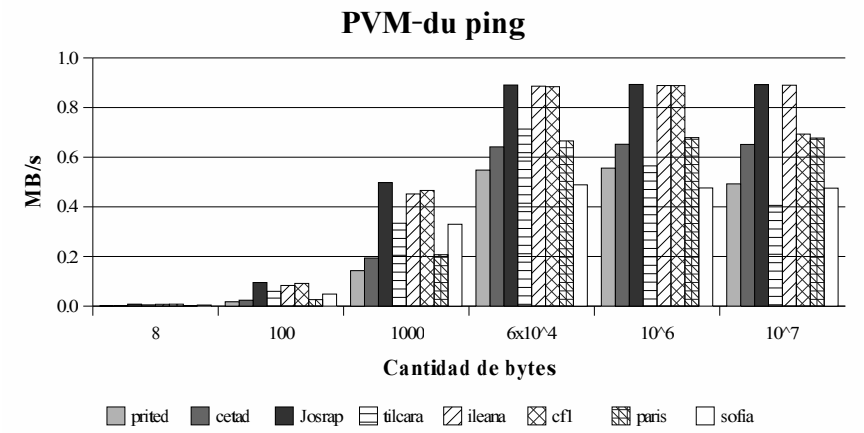

Figura 6: MB/s con ruteo *PvmDefault* y traducción.

4.2.3 Rendimiento con Ruteo entre Tareas

Tal como sucede con las alternativas de ruteo entre *pvmd*s (con codificación *PvmDataDefault* y con traducción), las alternativas con ruteo entre tareas (*PvmRouteDirect*), son similares en cuanto a rendimiento. Es decir que son bastante independientes de la forma en que se mantiene la consistencia de los datos en un ambiente heterogéneo, sea con codificación *PvmDataDefault* o con Traducción. En la Fig. 7 y en la Fig. 8 se muestra el rendimiento de las comunicaciones punto a punto con las alternativcas *PvmRouteDirect* y con Traducción para mantener la consistencia de los datos. Las principales características de rendimiento son:

- Con la mayoría de las estaciones de trabajo se superan los 0.8 MB/s, que era lo máximo al transmitir los mensajes con *PvmRouteDefault*. Esta tasa de transferencia se logra para mensajes del orden de los 10^6 bytes.
- Con todas las PCs (en las que no hace falta la traducción de datos) se supera el MB/s.
- Es significativamente bajo el rendimiento con las estaciones de trabajo prited y sofia. Pruebas preliminares posteriores demostraron que todas las comunicaciones con TCP son excesiva-

mente lentas y por lo tanto lo que falla es la configuración de las transferencias de datos utilizando el protocolo TCP en tales computadoras.

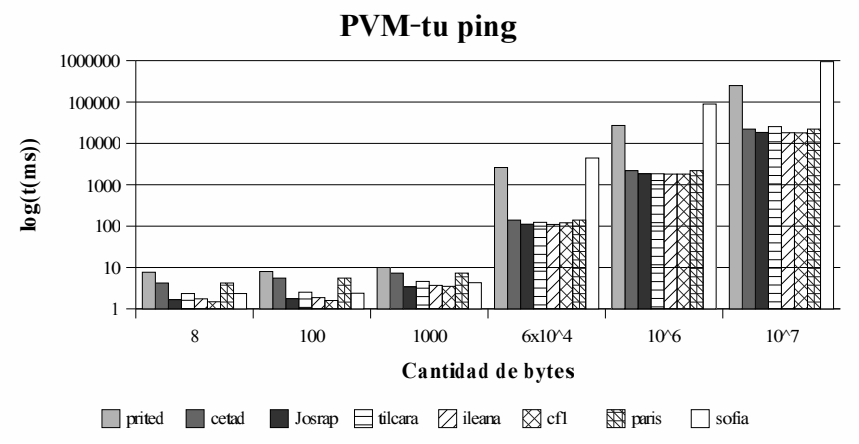

Figura 7: Tiempos con ruteo entre tareas y traducción.

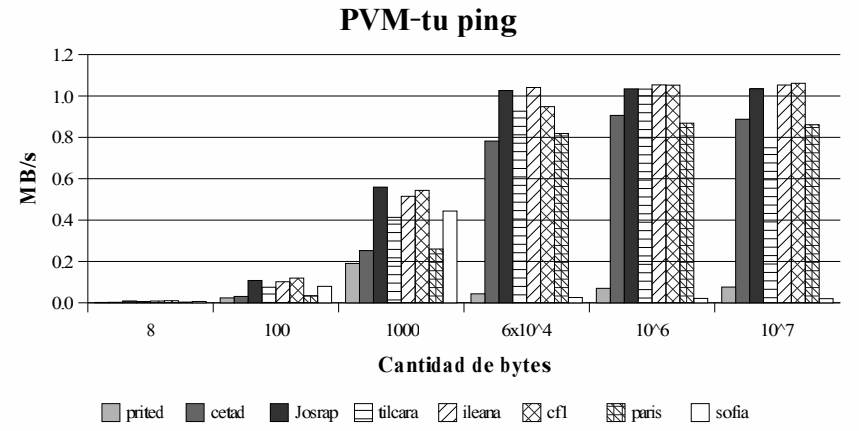

Figura 8: MB/s con ruteo entre tareas y traducción.

4.2.4 Conclusiones de la Experimentación con PVM

Desde el punto de vista del rendimiento, las comunicaciones punto a punto entre procesos de usuario utilizando PVM tienen las siguientes características:

- Conservar la consistencia de los datos no necesareiamente implica pérdida de rendimiento aunque sí tiene mayores requerimientos de memoria si se utiliza la alternativa *PvmDataDefault*.

- La comunicación utilizando la alternativa de PVM *PvmRouteDirect* tiende a dar mejores resultados que *PvmRouteDefault*, pero es mucho más dependiente de la configuración de las comunicaciones TCP tal como lo muestran los resultados de la Fig. 8. Tal sensibilidad no se encontró con la alternativa *PvmRouteDirect*.

- Aún para mensajes del orden de 10^6 bytes o mayores, el redimiento obtenido en las comunicaciones es dependediente de las máquinas que intercambian datos. Las diferencias de rendimiento se hacen mayores cuando es menor el tamaño de los mensajes.

- Los valores de α y β son dependientes de las máquinas, algo que era de esperar *solamente* para la latencia α. En el caso de α, los valores varían entre 1 y 10 ms (Fig. 7). En el caso de β, fuera de los casos de las estaciones de trabajo prited y sofia, los valores varían entre 0.85 y 1.1 MB/s aproximadamente (Fig. 8).

El hecho de que β sea dependiente de las (heterogeneidad) de las computadoras es significativamente negativo, ya que implica trasladar la heterogeneidad del hardware de cómputo de las comuputadoras al rendimiento de las comunicaciones que, a nivel de *hardware* de comunicaciones es homogéneo: *Ethernet*.

4.3 Resultados Obtenidos con ICMP

Tal como se explicó anteriormente, también se llevaron a cabo pruebas sencillas para determinar de alguna manera el mínimo *overhead* en cuanto a comunicaciones que tienen los procesos de usuario. Para esto se utilizó directamente el comando *ping* de Linux que permite variar la cantidad de datos de los paquetes ICMP que utiliza. La Fig. 9 muestra los resultados en términos de tiempo y la Fig. 10 muestra los resultados en términos de MB/s.

De acuerdo con la Fig. 9 se puede notar que, aunque no es uniforme, la latencia de los mensasjes no supera 1 ms en la comunicación con cualquier computadora. Comparando estos resultados con los de la Fig. 7 se puede ver que la sobrecarga de tiempo cuando se utiliza la librería PVM es varias veces mayor que la que tienen los procesos de usuario con el protocolo ICMP.

De acuerdo con la Fig. 10 se puede notar que la tasa de transferencia supera 1 MB/s para *todas* las computadoras, con lo cual se verifica que, aunque el valor de β no es uniforme para todas las computadoras tampoco tiene grandes variaciones. Comparándola con la Fig. 8 se puede notar que en algunos casos el rendimiento se mejora notablemente (aún sin considerar las estaciones de trabajo prited y sofia) y que la baja latencia hace que se llegue a mejor ancho de banda para mensajes con menor longitud.

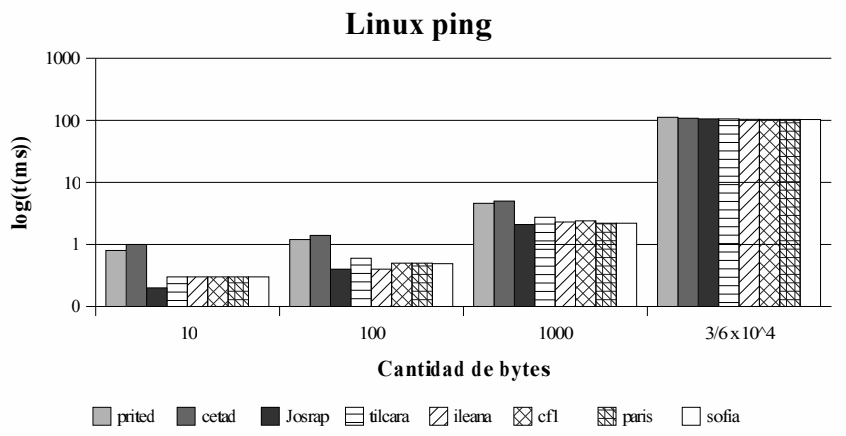

Figura 9: Tiempos con ICMP - *ping*.

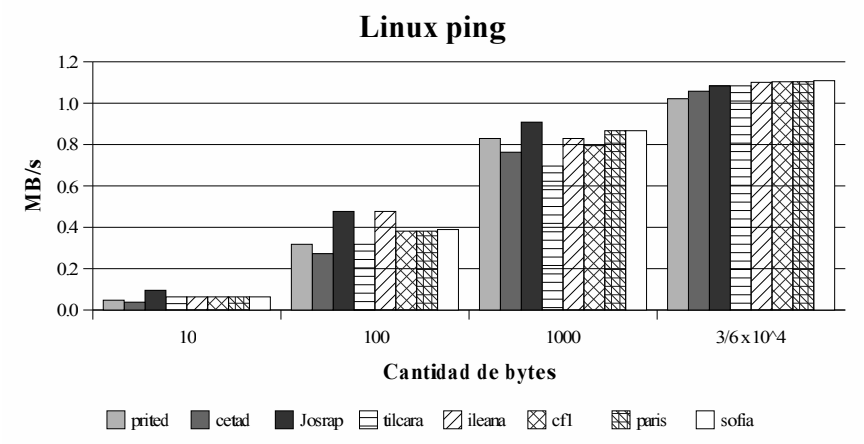

Figura 10: MB/s con ICMP - *ping*.

5 Conclusiones y Trabajo Futuro

Las principales conclusiones a partir de la experimentación con PVM y con el comando *ping* de Linux utilizando el protocolo ICMP son:

- Como era de esperar, tanto la latencia como el ancho de banda de las comunicaciones entre los procesos son dependientes de la librería/protocolo de comunicaciones entre procesos que se utiliza.
- Los tiempos de comunicación entre procesos utilizando PVM pueden ser mejorados, o al menos la sobrecarga de las rutinas/procesos de PVM es mayor que las de un protocolo de comunicaciones de más bajo nivel.
- Quizás la peor característica en cuanto a rendimiento de la librería PVM es la de trasladar la heterogeneidad del hardware de cómputo a las comunicaciones.

Por lo tanto, inmediatamente se puede pensar en mejorar las rutinas de comunicaciones más frecuentemente utilizadas con dos objetivos:

- Aumentar el rendimiento en términos de α y β de las comunicaciones (latencia y ancho de banda respectivamente.
- Mantener el rendimiento similar entre las computadoras, lo más independiente posible de la heterogeneidad, tal como se muestra en el caso de utilizar el protocolo ICMP (Fig. 9).

Por otro lado, y a más largo plazo, se pueden verificar también otro tipo de rutinas de comunicación, tales como las que resuelven las comunicaciones colectivas. En este caso, también se podrían evaluar en cuanto a rendimiento y proponer e implementar mejoras para aprovechar las características y rendimiento del *hardware* de comunicaciones.

Referencias

[1] Beowulf Homepage http://www.beowulf.org

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

[3] Institute of Electrical and Electronics Engineers, IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Std 754-1984, 1984.

[4] Pacheco P., Parallel Programming with MPI, Morgan Kaufmann, San Francisco, California, 1997.

[5] PVM (Parallel Virtual Machine) Homepage http://www.emm.ornl.gov/pvm/pvm_home.html

[6] LAM/MPI (Local Area Computing / Message Passing Interface) http://www.mpi.nd.edu/lam

[7] Wilkinson B., Allen M., Parallel Programming: Techniques and Applications Using Networking Workstations, Prentice-Hall, Inc., 1999.

[8] Institute of Electrical and Electronics Engineers, Local Area Network - *CSMA/CD Access Method and Physical Layer Specifications* ANSI/IEEE 802.3 - 1985, IEEE Computer Society.

[9] Sun Microsystems, Inc. XDR: External Data Representation Standard. RFC 1014, Sun Microsystems, Inc., June 1987.