Cómputo Paralelo en Estaciones de Trabajo no Dedicadas

Fernando G. Tinetti	Gerardo Sager	Dolores Rexachs	Emilio Luque
CeTAD[1] - LIDI[2] Universidad Nacional de La Plata	CeTAD[1] Universidad Nacional de La Plata	AOSO[3] Universidad Autónoma de Barcelona	AOSO[3] Universidad Autónoma de Barcelona

Resumen

Se presentan en este artículo una serie de consideraciones a tener en cuenta a la hora de utilizar una red de estaciones de trabajo ya instalada y en funcionamiento para procesamiento científico en paralelo. Estas consideraciones abarcan desde las ventajas de este tipo de procesamiento hasta los problemas de la paralelización, incluyendo también los problemas propios de la utilización de estaciones de trabajo que en general están dedicadas a otro tipo de tareas que no son cómputo paralelo. Se presentan las líneas generales de solución a los problemas de heterogeneidad de este tipo de redes de estaciones de trabajo, como así también algunas alternativas para el problema de adecuar las propias estaciones de trabajo para el cómputo paralelo ya que cada estación de trabajo será una *parte de* una máquina paralela virtual. La aplicación que se analiza en este artículo es la de multiplicación de matrices, dado que es muy sencilla en sí misma, y además posee características comunes a toda el área de cálculo numérico lineal, lo cual la hace apta para entender y analizar los problemas y soluciones en este tipo de aplicaciones.

Palabras Clave: Redes de Estaciones de Trabajo, Procesamiento Distribuido, Procesamiento Científico, Rendimiento.

1. Introducción

El cómputo paralelo en redes de estaciones de trabajo (Network of Workstations o NOW), siempre estuvo de alguna manera presente a partir de la posibilidad de conectar las computadoras en red/es, pero está propuesto más fuertemente a partir del aumento del rendimiento en relación con el costo de las mismas [1]. Además, si bien hay excepciones, normalmente estuvo ligado de alguna manera a aplicaciones con fuertes requerimientos de cómputo, como lo son los problemas *clásicos* provenientes del álgebra lineal, tales como la resolución de ecuaciones lineales, cálculo de autovalores y autovectores, diagonalización de matrices, etc.

Esta alternativa de procesamiento paralelo siempre se ha considerado como la más ventajosa en términos de costo. La razón principal está basada en la relación costo/rendimiento de las estaciones de trabajo. Mientras las supercomputadoras que siguieron la estructura tradicional de construcción y aplicaciones se mantienen siempre en el mismo orden de magnitud en cuanto a costo, las estaciones de trabajo han seguido el ritmo de crecimiento en cuanto a rendimiento y a su vez han bajado el costo aprovechando la ampliación del mercado y por lo tanto de las ventas. Quizás el extremo en

1 Centro de Técnicas Analógico-Digitales, Facultad de Ingeniería, Universidad Nacional de La Plata.
2 Laboratorio de Investigación y Desarrollo en Informática, Fac. de Informática, Universidad Nacional de La Plata.
3 Unidad de Arquitectura de Ordenadores y Sist. Operativos, Fac. de Ciencias, Universidad Autónoma de Barcelona.

este sentido lo han constituido las PCs, que al día de hoy están a la par de muchas estaciones de trabajo *clásicas* y a mucho menor costo.

A partir de estándares *de facto* como PVM [5] y estándares como MPI [7], para manejar la comunicación entre procesos y sincronización en una máquina paralela de pasaje de mensajes, muchos de los problemas relacionados con el cómputo paralelo en redes de estaciones de trabajo se asumieron como resueltos. Se puede decir que en el nivel más cercano al hardware esto es real. Por un lado, las comunicaciones entre computadoras con hardware ampliamente heterogéneo se resuelven utilizando estándares de comunicación previos tales como TCP-UDP/IP. Por otro lado, las diferencias tales como la representación de números, por ejemplo, se han resuelto utilizando métodos de codificación de datos que han resultado efectivos.

Si bien hay bastante discusión en términos de las distintas clasificaciones o caracterizaciones [12] que se pueden hacer sobre las redes de estaciones de trabajo para cómputo paralelo, en el contexto de este artículo se pueden identificar tres clases:

- Clusters: normalmente bajo esta clasificación se incluyen todos los conjuntos *homogéneos* de estaciones de trabajo. El término homogéneo en este contexto se utiliza en el sentido de: un mismo tipo de estaciones de trabajo, con características iguales o similares de hardware (cómputo y comunicación), y software. Lo más usual es encontrar estaciones de trabajo de un mismo modelo de un misma marca y con el mismo sistema operativo.
- Beowulf: esta clase es quizás un refinamiento de la clase anterior, pero con dos características muy distintivas [8]:
  1) la estación de trabajo básica es del tipo PC (con alguna versión del procesador Pentium), con sistema operativo Linux.
  2) el hardware de comunicaciones es bastante específico: suele encontrarse como mínimo placas Ethernet de 100Mb/s y switches de comunicación en el orden de los Gb/s.
- Heterogéneas: se podría decir que es la clase más general de NOWs, donde no hay uniformidad en el tipo de hardware ni software que se utiliza. Cualquier red local que se utilice para procesamiento paralelo se incluye dentro de esta clase.

Las dos primeras han sido bastante estudiadas y analizadas en términos de paralelización de aplicaciones y rendimiento. En la primera clase también se han tratado de identificar los problemas a resolver cuando las estaciones de trabajo no son *dedicadas* al cómputo paralelo [1]. Esto significa que cada una de las estaciones de trabajo tiene un "dueño" que se puede identificar como el que corre una determinada clase de aplicaciones, como pueden ser las interactivas en la misma consola de la estación de trabajo.

Los sistemas del tipo Beowulf se construyen explícitamente para cómputo paralelo y por lo tanto cada estación de trabajo (PC) está dedicada a este fin. También cada instalación Beowulf tiene una fuerte relación con la aplicación a ejecutar y en áreas específicas y con algunas variaciones de hardware se han llegado a alcanzar muy buenos valores de rendimiento [11].

Dentro de las redes de estaciones de trabajo que se han denominado como *heterogéneas* en la clasificación anterior se encuentran las que tienden a tener lo que se podría llamar "costo cero" de instalación y mantenimiento. Estas son la mayoría de las redes locales que se pueden adaptar para hacer cómputo paralelo. Cada estación de trabajo ya está instalada y conectada en red, y además tiene de alguna manera un "dueño": uno o un conjunto reducido de usuarios que normalmente la utilizan en forma regular. Estos usuarios son los que a su vez se encargarán de hacer que cada

computadora funcione regularmente, con lo cual el costo (y el tiempo) de mantenimiento de la red completa ya está contemplado antes de su utilización para cómputo paralelo. La adecuación de esta red para cómputo paralelo se puede hacer en base a software que no tiene costo (como PVM o algunas implementaciones de MPI), con lo cual el costo total de esta forma de cómputo paralelo se reduce a la instalación y mantenimiento de este tipo de software, que es mínimo.

2. Configuración

Para que tenga justificación el uso de una red de estaciones de trabajo para cómputo paralelo debe haber una necesidad real procesamiento. En este sentido, el área de álgebra lineal ha sido siempre una fuente de problemas de este tipo, con una amplia aceptación en cuanto a la necesidad no solamente de resolver problemas sino de estandarizar las soluciones [4]. De todas maneras, cualquiera de las áreas tradicionales con grandes requerimientos de procesamiento son candidatas a ser resueltas en redes de estaciones de trabajo.

Los requisitos mínimos para que cada estación de trabajo pueda ser parte de una máquina paralela que permita la ejecución de aplicaciones en forma distribuida son dos:
1. que esté conectada físicamente a una red local, y
2. que posea herramientas de software para el desarrollo y ejecución de aplicaciones paralelas.

El primero de los requisitos se satisface por la pertenencia de la estación de trabajo a una red local. Es lo más probable en los ambientes académico-científicos tales como los laboratorios y/o unidades académicas de una universidad. También es cada vez más frecuente que todas las estaciones de trabajo de una institución (empresa, organización no gubernamental, etc.) estén conectadas a una red local y a su vez a Internet. Es interesante mencionar que, al tener las estaciones de trabajo ya instaladas y funcionando en red, se evitan los problemas de instalación y puesta en marcha de todo el subsistema de intercomunicaciones vía red. La instalación y puesta en marcha de este subsistema de comunicaciones es muy frecuente (y consume bastante tiempo), sobre todo en las PCs, donde el hardware es ampliamente heterogéneo y la instalación de hardware y software se realiza casi artesanalmente.

El segundo de los requisitos mencionados previamente normalmente no se encuentra satisfecho de por sí en todas las redes locales. Esto se refiere a que, además de tener una red de estaciones de trabajo, cada una de las computadoras se debe preparar para hacer cómputo paralelo. Las librerías y/o ambientes de software como PVM o MPI mencionados previamente han demostrado ser ampliamente útiles en esta tarea. Tanto PVM como muchas implementaciones de MPI son de instalación y utilización sin costo, al menos en el ámbito de investigación y desarrollo académico. En ambos casos se puede acceder al código fuente en Internet e instalarlo localmente en todas las estaciones de trabajo en que sea necesario [15] [14]. Además, tanto PVM como las implementaciones de MPI no tienen más requerimientos que conectividad vía TCP-UDP/IP, que se puede encontrar en la mayoría de las redes locales instaladas. A partir de este tipo de software, ya es posible desarrollar y ejecutar aplicaciones paralelas. Sin embargo, con respecto a cómo se llegan a tener instaladas las herramientas de desarrollo y ejecución de aplicaciones paralelas en las redes locales, se podrían identificar dos clases:
a) En estaciones de trabajo con sistema operativo UNIX.
b) En estaciones de trabajo con Windows.

Por un lado, en el caso de las estaciones de trabajo que ya tienen como sistema operativo alguna

versión de UNIX (AIX, IRIX, Solaris, Linux, etc.), la instalación es medianamente sencilla y bastante bien documentada, por lo tanto no presenta ningún problema adicional. El costo de instalación de las herramientas necesarias para procesamiento paralelo se reduce al tiempo de instalación, ya que, como se detalló previamente, el software se encuentra disponible sin costo en Internet.

Por otro lado, en las estaciones de trabajo que tienen instalado únicamente alguna versión de Windows (95/98, NT) se debería, antes de instalar herramientas tales como PVM o MPI, instalar alguna de las versiones de UNIX que se mencionaron antes. Normalmente la alternativa elegida es Linux, dado que el costo se reduce al del tiempo de instalación y la flexibilidad que ofrece para su instalación en PCs. Si se puede dividir el disco de la estación de trabajo para dedicar una partición a Linux, primero se debería instalar Linux y luego las herramientas para procesamiento paralelo. Esta alternativa es sencilla desde el punto de vista técnico aunque consume algo de tiempo. Sin embargo, en algunas ocasiones suele ser muy costoso particionar el disco de la estación de trabajo, porque implica en la mayoría de los casos la reinstalación de Windows junto con todo el software instalado bajo Windows. Dado que la probabilidad de que esto suceda en estaciones de trabajo ya instaladas y en uso es muy alta, una alternativa válida es la instalación de WinLinux, que no es más que una distribución de Linux que puede ser instalada y utilizada con el sistema de archivos de Windows/ DOS. Siguiendo esta alternativa, el problema nuevamente se resuelve en dos etapas: instalación de WinLinux e instalación de las herramientas para procesamiento paralelo. Se debe destacar que ya sea que se instale Linux o WinLinux, el costo se reduce al tiempo de instalación ya que en ambos casos se tiene acceso al software de forma gratuita en Internet [13] [16].

Si bien desde el punto de vista técnico estarían resueltos la mayoría de los problemas para comenzar a utilizar las estaciones de trabajo de una red local para cómputo paralelo, se debe notar que la disponibilidad de estas computadoras no es *total*. Inicialmente, el dueño de cada estación de trabajo ya debió ceder parte de su almacenamiento en disco para instalar otro sistema operativo (en el caso en que solamente haya tenido Windows) y herramientas para cómputo paralelo. Pero además, para la ejecución de tareas paralelas, el dueño de cada computadora debe ceder tiempo de cómputo. Esto es real aunque las aplicaciones paralelas se desarrollen y depuren en máquinas de uso exclusivo. La mejor de las situaciones en este contexto se da cuando toda la ejecución en paralelo, es decir la utilización de todas las estaciones de trabajo, se da en horas de baja o nula utilización por parte de los dueños de las computadoras. Se podrían mencionar, en principio, las horas nocturnas o de fines de semana. Aún en este sentido, puede generar cierta incomodidad en el dueño de la estación de trabajo el hecho de llegar a su propia computadora y encontrar que está siendo utilizada y posiblemente con otro sistema operativo en marcha.

El problema de disponibilidad de las estaciones de trabajo no necesariamente puede pasarse por alto, y de hecho, por ejemplo, se ha propuesto la migración de las tareas paralelas como una posible solución al problema [1]. La restricción de ejecutar aplicaciones paralelas puede ser un poco incómoda cuando las respuestas a un problema se necesitan con cierta urgencia, pero además no siempre se puede, por ejemplo, depurar las aplicaciones paralelas con un subconjunto del total de las computadoras que finalmente se utilizarán en la ejecución *real*. Para el análisis de eficiencia de la ejecución, por ejemplo, necesariamente deben utilizarse todas las computadoras disponibles.

Sin embargo, el cómputo paralelo tiene varios puntos a favor en las redes de estaciones de trabajo no dedicadas:

- Cada usuario de una estación de trabajo puede tener acceso a una computadora de mayor rendimiento que la propia, aún en el caso de tener la mejor computadora de la red local.
- Como resultado de la instalación de software por más de un usuario, la estación de trabajo tiene

más control (normalmente cada usuario planifica mejor la instalación de software), y por lo tanto, mejor administración. Esto es más frecuente en los ambientes académicos, donde las estaciones de trabajo no son exclusivas de un usuario.
- Cada estación de trabajo tiene mayor utilización: se agregan los tiempos de cómputo de las tareas paralelas. Esto produce una mejor utilización de toda la red local.
- Se tiene mejor conocimiento de las características del hardware instalado: ejecutar aplicaciones paralelas implica conocer, por ejemplo, la capacidad de cómputo relativa de cada máquina; y por lo tanto se pueden planificar mejor las aplicaciones e incluso las actualizaciones de hardware.

3. Paralelización de Aplicaciones

Una vez que se han satisfecho los requisitos anteriores se tiene una máquina paralela capaz de realizar procesamiento distribuido formada por la conjunción de varias estaciones de trabajo que pueden ejecutar código paralelo. Desde un punto de vista de clasificación cercano al hardware, este tipo de máquina paralela pertenece a las MIMD (Multiple Instruction, Multiple Data Stream). Desde un punto de vista el punto de vista más cercano a las aplicaciones paralelas a desarrollar y ejecutar, pertenece a la clase de máquinas paralelas por pasaje de mensajes. En estas máquinas, el modelo general de las aplicaciones es el de CSP (Communicating Sequential Processes): procesos secuenciales que se comunican para cooperación y/o sincronización.

Para resolver aplicaciones en paralelo, y en particular sobre una red de estaciones de trabajo, se deben tener programas paralelos. La paralelización de aplicaciones no es trivial en muchos casos, aunque hay excepciones. Dentro del área de los problemas relacionados con álgebra lineal, un gran porcentaje se puede resolver bajo el modelo SPMD (Single Program, Multiple Data). En este modelo de paralelización y ejecución, un mismo programa se ejecuta en cada procesador disponible, procesando un conjunto distinto de datos. En el caso particular de las NOW, un mismo proceso se ejecuta en cada estación de trabajo, y se comunican normalmente por pasaje de mensajes entre las computadoras.

El modelo SPMD puede ser sencillo de codificar, con un mismo código básicamente secuencial más partes de código para comunicación y/o sincronización. Sin embargo, siempre están presentes las soluciones a problemas numéricos que tienen una amplia gama de soluciones ya desarrolladas para máquinas secuenciales o máquinas paralelas vectoriales. Esto hace difícil que vuelvan a ser consideradas para darles otro tipo de soluciones, como con el modelo SPMD, teniendo en cuenta la cantidad de análisis que se ha hecho, por ejemplo, para conocer la estabilidad numérica de las soluciones computacionales. Sin embargo, este tipo de inconvenientes han sido descartados y siempre que se construyó una computadora suficientemente *buena* (hasta ahora este término ha estado asociado directamente con la velocidad de procesamiento), se han desarrollado o codificado soluciones apropiadas para esta computadora. Sin embargo, se deben tener algunas consideraciones específicas en el contexto de las NOWs:

1. El código a ejecutar en cada estación de trabajo puede no ser el mismo. Como mínimo, cada estación de trabajo tiene un formato de instrucción y de datos que es propio y muy probablemente no uniforme con respecto a la mayoría de las estaciones de trabajo. El código fuente del programa, por ejemplo, puede ser el mismo, pero necesariamente el programa ejecutable tendrá que ser generado en (o compilado para) cada estación de trabajo particular.
2. Las soluciones paralelas desarrolladas para máquinas paralelas tradicionales asumen que el hardware es homogéneo. El balance de carga para los programas paralelos que tienen el modelo de cómputo SPMD se obtiene más o menos automáticamente por la asignación de la misma

cantidad de datos a procesar en cada procesador. Las estaciones de trabajo de una red local son normalmente muy heterogéneas y por lo tanto con distintas capacidades de cálculo, por ejemplo. Esto elimina la simplificación de asignar la misma cantidad de datos para tener balance de carga computacional.

3. Las soluciones paralelas asumen normalmente alguna forma de red física de intercomunicación de procesadores que es estática y del tipo malla o toro. En general, estos tipos de redes de interconexión no son los de las redes locales, donde la gran mayoría son un único bus de comunicaciones, donde una sola computadora a la vez puede emitir un mensaje y todas las demás al mismo tiempo pueden recibirlo.
4. Las soluciones paralelas normalmente asumen una cantidad determinada de procesadores. En el caso de las máquinas paralelas estándares, la cantidad de procesadores es estática. En el caso de las máquinas paralelas construidas a partir de estaciones de trabajo, no siempre se puede asumir una cantidad determinada de procesadores. Esto se debe a la disponibilidad también heterogénea de las estaciones de trabajo, que depende a su vez de las necesidades de cada usuario "dueño" de la misma.

Las últimas tres consideraciones muestran el fuerte impacto que tienen las máquinas paralelas tradicionales sobre la paralelización de código. Por un lado esto es lógico dada la especificidad de la mayoría de algunas máquinas paralelas, y la necesidad de aprovechar al máximo las características específicas de estas máquinas. Pero por otro lado también demuestra que hay una gran necesidad de establecer soluciones en el ámbito de la paralelización de aplicaciones a ser resueltas en redes de estaciones de trabajo.

4. Una aplicación: Multiplicación de Matrices

Uno de los problemas que ha recibido mayor atención en cuanto a su solución paralela dentro del área del álgebra lineal es el de multiplicación de matrices: C = AxB, donde A es una matriz de *mxk*, B es una matriz de *kxn* y C es una matriz de *mxn*. Entre las características más importantes que hacen atractivo el estudio y el análisis de la multiplicación de matrices se pueden mencionar:

- Es ampliamente útil. En el contexto de la librería de software LAPACK (Linear Algebra PACKage), por ejemplo, pertenece al subconjunto denominado BLAS (Basic Linear Algebra Software) Level 3. Este subconjunto de operaciones se caracteriza por ser operaciones entre matrices, y diferenciadas de las operaciones entre escales, escalares con vectores, y vectores con matrices. De hecho se ha demostrado que todo BLAS Level 3 se puede llevar a cabo en función de la multiplicación de matrices [6].
- Tiene grandes requerimientos de procesamiento (crece de forma cúbica respecto al crecimiento cuadrático de los datos), lo cual la hace comparable con la mayoría (sino todas) de las aplicaciones del ámbito numérico.
- Es sencilla de codificar, aún las soluciones paralelas, en relación con las demás aplicaciones de su área, lo cual facilita la experimentación y el análisis de alternativas.

Esta aplicación, además permite en el caso particular de las soluciones paralelas sobre estaciones de trabajo, mostrar cómo se aplican y resuelven las consideraciones mencionadas en la sección anterior.

El balance de carga para obtener la matriz resultado (C), de una multiplicación de matrices se logra muy fácilmente en las máquinas paralelas homogéneas asignando la misma cantidad de datos de A, y B (y por lo tanto de C) a cada procesador. De hecho, todas las soluciones paralelas a este problema

son similares en este sentido. En una red de estaciones de trabajo la cantidad de datos de A y B que se asignen a cada computadora deberá ser proporcional a la velocidad de esta computadora con respecto a las demás. Es decir que si la estación de trabajo ws_i es dos veces más veloz que la estación de trabajo ws_j, ws_i debería recibir el doble de datos a calcular de la matriz C que ws_j [9].

Las soluciones paralelas a este problema suelen diferir en cuanto al patrón de comunicaciones que establecen, es decir en la frecuencia y cantidad de datos transferidos punto a punto entre procesadores [3] [10] [2]. Sin embargo, en general todas las implementaciones asumen una red estática de interconexión del tipo malla o toro. Dado que en general tiene que llevar a cabo algún tipo de comunicación multicast o broadcast, se han diseñado distintas formas de implementarlo sobre este tipo de redes intentando aprovechar la capacidad de los procesadores para hacer cómputo solapado en el tiempo con comunicaciones. En el caso de la paralelización sobre redes de estaciones de trabajo, estas soluciones no son viables en términos de rendimiento ya que la red de interconexión (en general) es del tipo bus, y por lo tanto:

- a lo sumo un procesador puede hacer un envío a datos a la vez
- las comunicaciones multicast y broadcast podrían aprovechar el tipo de interconexión y llevar a cabo las transferencias de datos más rápidamente. Esto depende fuertemente de la implementación de las herramientas de software que se instalen.
- no necesariamente las estaciones de trabajo pueden realizar cómputo y comunicación solapados en el tiempo.

También relacionado con las comunicaciones en las redes de estaciones de trabajo se debe notar que tanto la latencia (definida como el tiempo mínimo para enviar un dato de una computadora a otra) como el ancho de banda (definido como cantidad de datos por unidad de tiempo que se pueden transferir) están bastante lejos de los que se pueden encontrar en una computadora paralela tradicional. Esto tiene un impacto directo sobre la granularidad de la aplicación, ya que al ser elevado el tiempo mínimo de comunicación entre las estaciones de trabajo existe un límite estricto a la cantidad de comunicaciones con respecto al procesamiento realizado por datos transferidos que se debe respetar para que los índices de rendimiento no caigan a valores inaceptables. En el caso de la multiplicación de matrices, mientras mayor sea cada mensaje se estará en mejores condiciones para llegar a tener valores de rendimiento aceptables.

La cantidad de procesadores de una máquina paralela sin lugar a dudas afectará el rendimiento. Sin embargo, si a esto se le agrega una cantidad "variable" de procesadores, el programa paralelo que resuelva el problema deberá adaptarse a esta situación. Dejando de lado la posibilidad de fallas, en las redes de estaciones de trabajo aún es posible que no todas las computadoras estén disponibles al mismo tiempo y por lo tanto la multiplicación de matrices debe llevarse a cabo con las que sí se pueden utilizar.

Si bien existen muchas alternativas para la paralelización de esta aplicación, se probaron las dos con mayor granularidad posible:

- Totalmente Paralela: cada estación de trabajo calcula su parte de la matriz resultado C sin comunicarse con las demás. Para esto debe calcular/generar previamente todos los datos de las matrices A y B que sean necesarios.
- Por Filas y Columnas: cada estación de trabajo tiene una determinada cantidad de filas correspondiente a su velocidad relativa, y una parte de las columnas de la matriz B. Debe recibir de las demás computadoras las columnas de B que no tiene para completar el cálculo de C.

Asumiendo cuatro estaciones de trabajo homogéneas, el primer tipo de paralelización podría describirse esquemáticamente como:

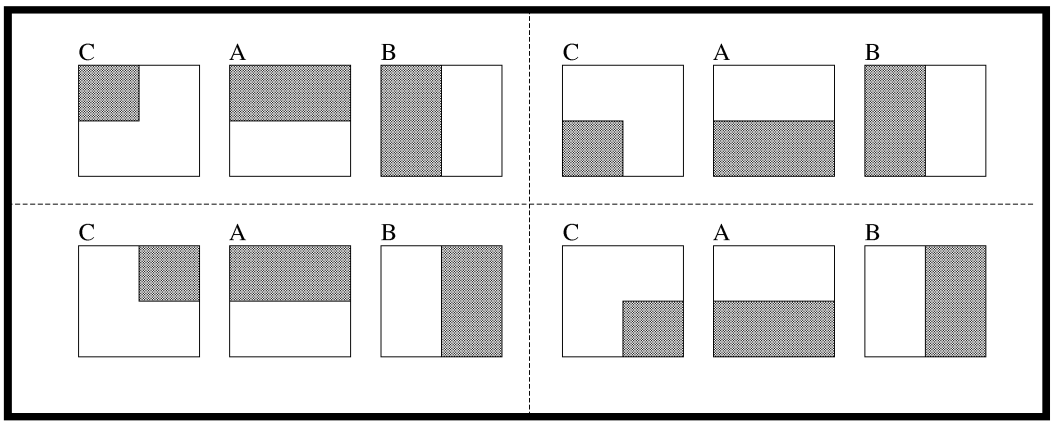

Figura 1: Paralelización Totalmente Paralela.

donde las áreas sombreadas de cada matriz corresponden a los datos residentes en cada computadora. Cada una de las estaciones de trabajo calcula un cuarto de la matriz resultado C y debe, para ello, tener una mitad de cada una de las matrices A y B. Esta paralelización tiene al menos dos inconvenientes: a) replicación de datos en varias estaciones de trabajo, y b) cantidad de memoria local necesaria para realizar los cálculos. Por otro lado, si una o las dos matrices A y B son calculadas por métodos complejos, habría que paralelizar a su vez la generación de estas matrices y redistribuir los datos una vez que se hayan generado.

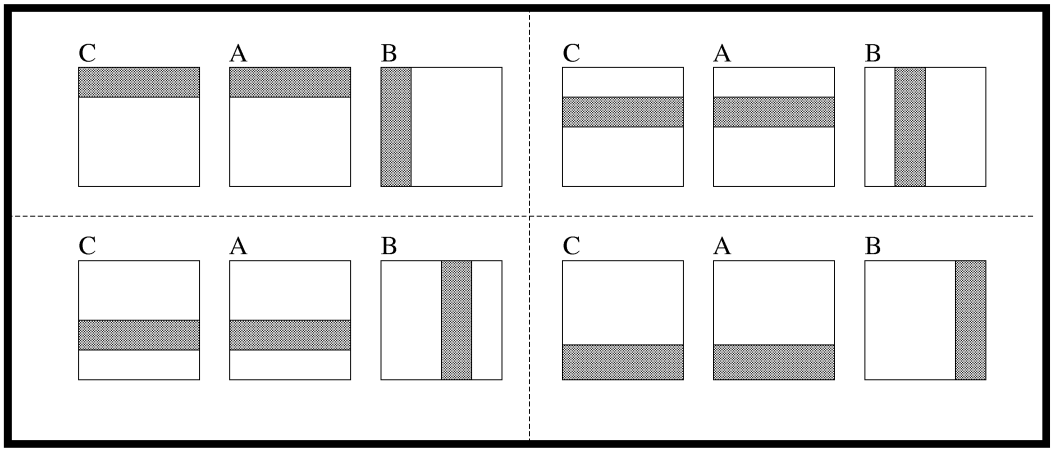

Figura 2: Paralelización por Filas y Columnas.

donde nuevamente las áreas sombreadas de cada matriz corresponden a los datos residentes en cada computadora. Cada una de las estaciones de trabajo calcula un cuarto de la matriz resultado C y debe, para ello, recibir los tres cuartos de las columnas de la matriz B que no tiene localmente. A diferencia de la paralelización anterior, no requiere la replicación de datos y la cantidad de memoria local necesaria también se reduce. El inconveniente en este caso es la necesidad de comunicación de cada estación de trabajo con todas las demás para computar el sector de la matriz C que le corresponde.

Ambos tipos de paralelización de multiplicación de matrices se pueden llevar a cabo en el caso de estaciones de trabajo heterogéneas asignando la cantidad de filas de A de forma proporcional a la velocidad relativa de cada computadora.

5. Experimentación

Inicialmente, en toda paralelización sobre estaciones de trabajo heterogéneas se debe conocer la velocidad relativa de cada computadora. Estas capacidades de cálculo, de por sí establecen una relación de velocidad entre las distintas computadoras siempre que se utilice un índice uniforme para caracterizar a todas las estaciones de trabajo. Si bien el cálculo de las velocidades puede generar discusión en cuanto a la forma en que se calcula, en este caso se realizó con las siguientes tres características:

1. Se ejecutó la misma aplicación (multiplicación de matrices) en todas las computadoras, y además es la misma aplicación que se paraleliza.
2. La multiplicación de matrices permite expresar la capacidad de cálculo de cada computadora en Mflop/s independientemente de las características de cada computadora, incluyendo la posibilidad de que la computadora sea paralela o no.
3. El tamaño de la multiplicación de matrices en cada estación de trabajo es el máximo tal que no excede la capacidad de la memoria principal. Esto implica que no es necesario recurrir al *swapping* de páginas o de procesos para ejecutar la aplicación.

En este caso se utilizarán cinco estaciones de trabajo con la siguiente capacidad de cálculo expresada en Mflop/s, junto con una breve descripción del hardware:

Nombre	CPU / Mem	Mflop/s
purmamarca	Pentium II 400 MHz / 64 MB	316
cetadfomec1	Celeron 300 MHz / 32 MB	243
cetadfomec2	Celeron 300 MHz / 32 MB	243
sofia	PPC604e 200 MHz / 64 MB	225
Josrap	AMD K6-2 450 MHz / 62 MB	99

Tabla 1: Mflop/s de Cinco Estaciones de Trabajo.

Se ejecutó la aplicación para distintos tamaños de matrices cuadradas, variando entre matrices de 500x500 elementos hasta matrices de 3500x3500 elementos. El rendimiento máximo y mínimo obtenido según el tipo de paralelización, expresado en Mflop/s, se muestra en la Tabla 2.

Tamaño	Totalmente Paralela (Mflop/s)		Filas y Columnas (Mflop/s)	
	Máx.	Mín.	Máx.	Mín.
500	735	694	42	14
1000	905	905	68	59
1500	986	979	120	99
2000	1025	1014	133	126
2500	1053	1042	205	194
3000	1063	1047	205	163
3500	769	447	181	162

Tabla 2: Rendimiento con Cada Tipo de Paralelización.

En el caso de la ejecución Totalmente Paralela, los valores de rendimiento expresan las siguientes situaciones:

- El costo de generación de las columnas de la matriz B es relativamente menos importante a medida que el tamaño de las matrices crece. Esto explica por qué aumenta el rendimiento junto con el aumento de la dimensión de las matrices.
- Para todos los tamaños, excepto el último, hay una gran estabilidad entre el máximo y el mínimo de rendimiento.
- Para el tamaño de matrices de 3500x3500 elementos los valores de rendimiento no son los esperados en magnitud ni en variación. Un análisis un poco más exhaustivo de la ejecución muestra que las dos computadoras con menor cantidad de memoria comenzaron a hacer *swapping* de páginas en este tamaño de matrices.
- Para las dimensiones de matrices entre 2000 y 3000 los valores de rendimiento son muy cercanos al óptimo de 1126 Mflop/s (más del 90% de la suma de los valores individuales mostrados en la Tabla 1).

En el caso de la paralelización por filas y columnas, los resultados de rendimiento están muy por debajo de lo esperado. En el mejor de los casos se llega a tener un rendimiento sostenido cercano al 20% del óptimo, lo cual es inaceptable. En este caso es necesario un análisis un poco más exhaustivo de la ejecución paralela para conocer los detalles del problema, que claramente provienen de la comunicación entre las estaciones de trabajo. Un esquema gráfico de lo que sucede en tiempo de ejecución (tomado en segundos) para matrices de 1000x1000 elementos se da en la Fig. 3.

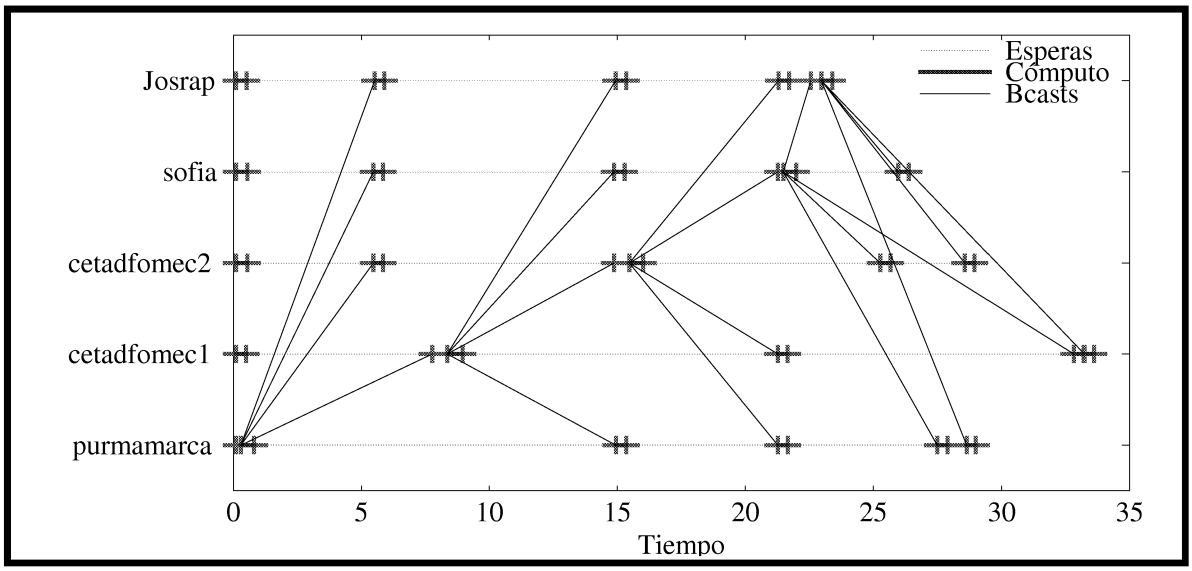

Figura 3: Ejecución de la Paralelización por Filas y Columnas.

Analizando los tiempos relativos de cada tarea se puede notar que los broadcasts de las columnas de B ocupan casi todo el tiempo de ejecución. Las causas pueden ser variadas, aunque se debe notar también que cada broadcast transfiere un total de 800000 bytes, lo cual, en teoría se podría transferir en el orden del segundo de comunicación. sobre una red Ethernet de 10 Mb/s. Esto implica que las herramientas de comunicación (en este caso las que aporta PVM), tienen una sobrecarga excesiva sobre el tiempo total de la ejecución de las aplicaciones paralelas, o como mínimo sobre este tipo de aplicaciones paralelas.

6. Conclusiones y Trabajo Futuro

La alternativa de cómputo paralelo en estaciones de trabajo no dedicadas es quizás la forma más accesible en costo y disponibilidad para resolver aplicaciones con grandes requerimientos de cómputo. Sin embargo, no necesariamente es inmediato el aprovechamiento de esta forma de cómputo paralelo por varias razones, entre las que se pueden mencionar:

- Administración - disponibilidad de cada estación de trabajo.
- Paralelización de aplicaciones, incluyendo el estudio de la granularidad y distribución de los datos.
- El rendimiento y la arquitectura de la red de comunicaciones de las redes locales no siempre es apropiada para el cómputo paralelo.
- Los algoritmos paralelos ya desarrollados para las computadoras paralelas tradicionales no siempre son apropiados para este tipo de arquitecturas y por lo tanto no se puede aprovechar todo lo que se ha desarrollado y probado en ese ámbito.

Se deben llevar a cabo en muchos casos estudios profundos para aprovechar al máximo cada estación de trabajo así como también las comunicaciones en las redes locales. De la misma manera, no se puede pasar por alto que la base de muchos problemas de rendimiento pasa por la granularidad de las aplicaciones, que no siempre es muy fácil de decidir *a priori* o manejar gran cantidad de alternativas.

Las herramientas de desarrollo y ejecución de aplicaciones paralelas sobre estaciones de trabajo no siempre están o pueden estar optimizadas para aprovechar al máximo el hardware existente (aunque sea estándar y muy conocido), y por lo tanto en muchas oportunidades se debe reemplazar (al menos en parte) para lograr índices de rendimiento aceptables. Esto implica recodificar, por ejemplo, subprogramas de comunicación, que no siempre es fácil de realizar.

Agradecimientos

Tanto el Ing. A. Quijano como el Ing. A. De Giusti han proporcionado múltiples ideas y la infraestructura necesaria para llevar a cabo toda la experimentación. Dadas las características de este artículo y el proyecto dentro del cual se enmarca, muchos integrantes del CeTAD han cedido parte de sus estaciones de trabajo para realizar la implementación, y se debe destacar su amplio espíritu de colaboración.

Bibliografía

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

[2] Choi J., "A New Parallel Matrix Multiplication Algorithm on Distributed-Memory Concurrent Computers", Proceedings of the High-Performance Computing on the Information Superhighway, IEEE, HPC-Asia '97.

[3] Choi J., J. Dongarra, D. Walker, "PUMMA: Parallel Universal Matrix Multiplication Algorithms on Distributed Memory Concurrent Computers", Concurrency: Practice and Experience, 1994.

[4] Demmel J., J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, D. Sorensen, Prospectus for the Development of a Linear Algebra Library for High-Performance Computers, ANL, MCS-TM-97, Sep. 1987, disponible en http://www.netlib.org/lapack/lawns.

[5] 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.

[6] Kågström B., P. Ling, C. Van Loan, "Portable High-Performance GEMM-based Level 3 BLAS", R. F. Sincovec et al., Editor, Parallel Processing for Scientific Computing, Philadelphia, 1993, SIAM, pp. 339-346.

[7] MPI Forum, "MPI: a message-passing interface standard", International Journal of Supercomputer Application, 8 (3/4), pp. 165-416, 1994.

[8] Ridge D., D. Becker, P. Merkey, T. Sterling, "Beowulf: Harnessing the Power of Parallelism in a Pile-of-PCs, Proceedings IEEE Aerospace, 1997, disponible también en http://www.beowulf.org/papers/papers.html

[9] Tinetti F., A. Quijano, A. De Giusti, "Heterogeneous Networks of Workstations and SPMD Scientific Computing", Proceedings of the 1999 International Workshops on Parallel Processing, IEEE, Inc., 1999 International Conference on Parallel Processing, University of Aizu, Aizu-Wakamatsu, Fukushima, Japan, September 21 - 24, 1999.

[10] Van de Geijn R., J. Watts, "SUMMA Scalable Universal Matrix Multiplication Algorithm", LAPACK Working Note 99, Technical Report CS-95-286, University of Tennessee, 1995, disponible en http://www.netlib.org/lapack/lawns.

[11] Warren M. S., T. C. Germann, P. S. Lomdahl, D. M. Beazley, J. K. Salmon, "Avalon: An Alpha/Linux Cluster Achieves 10 Gflops for $150k". Disponible en http://cnls.lanl.gov/avalon/avalon_bell98/ Submitted for the 1998 Gordon Bell Price/Performance Prize.

[12] Zhang X., Y. Yan, "Modeling and characterizing parallel computing performance on heterogeneous NOW", Proceedings of the Seventh IEEE Symposium on Parallel and Distributed Processing, (SPDP'95), IEEE Computer Society Press, San Antonio, Texas, October 1995, pp. 25-34.

[13] Linux RedHat Homepage, http://www.redhat.com

[14] MPICH Homepage, http://www-unix.mcs.anl.gov/mpi/mpich

[15] PVM Homepage, http://www.epm.ornl.gov/pvm

[16] Winlinux Homepage, http://www.winlinux.net