Capítulo 6: Conclusiones y Trabajo Futuro

En este capítulo se presentan las principales conclusiones de esta tesis a partir de los capítulos anteriores con un énfasis especial en la dirección de identificar problemas y soluciones para la obtención de máximo rendimiento con cómputo paralelo en redes locales de computadoras. El contexto inicial de hardware de procesamiento paralelo lo dan las redes locales de computadoras que están instaladas y que se pueden aprovechar para resolver problemas en paralelo. También se presenta un breve resumen de los aportes de esta tesis y su relación con las publicaciones que se han hecho a lo largo del desarrollo de la misma.

También en este capítulo se presentan algunas consideraciones respecto de la continuación de la investigación en esta área. Si bien es muy difícil una estimación precisa de las extensiones sí se pueden identificar con bastante claridad algunos problemas inmediatos que se pueden resolver en este contexto de cómputo paralelo y también algunas alternativas de utilización de hardware más allá de una red local de computadoras.

6.1 Conclusiones

La evolución de las computadoras paralelas ha sido clara en varias direcciones, una de las cuales es la utilización de hardware de cómputo estándar. En este sentido, los microprocesadores de uso masivo en las computadoras de bajo costo como las estaciones de trabajo y PCs son utilizados en las computadoras paralelas que se reportan entre las de mayor capacidad de cálculo absoluta [86]. En el escalafón más bajo en cuanto a costo de cómputo paralelo se pueden ubicar a las redes locales de computadoras, que tienen el mismo tipo de procesadores de base pero que en principio no han sido orientadas a cómputo paralelo. De hecho, la mejor relación costo/rendimiento de hardware paralelo es el de las redes locales ya instaladas porque no tienen ningún tipo de costo de instalación ni de mantenimiento dado que su existencia es independiente del cómputo paralelo. Sin embargo, no se pueden dejar de lado otros costos adicionales que tienen estas redes, tales como el de instalación y mantenimiento del software específico para cómputo paralelo y el de la baja disponibilidad de las computadoras que, como se ha mencionado, no tienen como prioridad la ejecución de programas paralelos.

Tanto las redes de computadoras ya instaladas que se pueden utilizar para cómputo paralelo como las instalaciones del tipo Beowulf y/o clusters homogéneos que evolucionan en el sentido de reposición y/o agregado de computadoras suelen tener hardware de cómputo heterogéneo y hardware de comunicaciones homogéneo. La heterogeneidad de las computadoras de las redes locales instaladas es más o menos “natural” teniendo en cuenta tanto el tiempo de instalación y consiguiente evolución, como las distintas funciones o tipo de problemas hacia los cuales están orientadas cada una de las computadoras de la red local. La heterogeneidad de las computadoras de las instalaciones del tipo Beowulf y/o clusters homogéneos es una consecuencia no buscada de la evolución del hardware de bajo costo utilizado como base: las PCs. El tiempo de disponibilidad en el mercado de los componentes básicos tales como un tipo de procesador, memoria y disco rígido de PCs, por ejemplo, es muy corto. Por lo tanto, siempre que se necesita mayor potencia de cálculo (más PCs) o reemplazar una computadora que deja de funcionar, la probabilidad de tener heterogeneidad en el hardware es relativamente grande comparada con la que tiene una computadora paralela “tradicional” bajo las mismas circunstancias. Y la probabilidad crece a medida que transcurre el tiempo y los componentes no están disponibles de manera inmediata en el mercado.

La homogeneidad del hardware de comunicaciones está dada también por el bajo costo, en este caso de las placas de interfase (NIC: Network Interface Cards) de comunicaciones Ethernet. El estándar definido como Ethernet en sus varias versiones se ha instalado como el de más bajo costo y aparentemente va a seguir con esta tendencia en todas sus versiones definidas hasta el momento: 10 Mb/s, 10/100 Mb/s, 1 Gb/s y 10 Gb/s. De hecho, las redes locales mayoritariamente utilizan Ethernet de 10 y de 10/100 Mb/s dado que ha probado ser muy útil para la mayoría de las aplicaciones de oficina. Además, las instalaciones Beowulf se recomiendan con hardware de comunicaciones de 100 Mb/s y con cableado basado en switches de comunicaciones Ethernet de 100 Mb/s y/o 1 Gb/s. El bajo costo de estas redes incluye no solamente el mismo hardware de interconexión de las computadoras (NICs), sino también todo el personal técnico ya capacitado y con experiencia, con el que los demás tipos de redes utilizadas no cuentan.

Tanto la heterogeneidad de cómputo como las redes Ethernet de interconexión de procesadores (desde el punto de vista de una máquina paralela) tienen características muy bien definidas y no necesariamente apropiadas para cómputo paralelo. La heterogeneidad de cómputo plantea un problema que raramente (sino *nunca*) se debe enfrentar en las computadoras paralelas tradicionales, que es el desbalance dado por las capacidades de cálculo de los procesadores que son diferentes. El hardware de interconexión Ethernet plantea problemas quizás más serios:

- Ethernet no está orientado *a priori* a cómputo paralelo y por lo tanto índices de rendimiento tales como latencia y ancho de banda son bastante mayores que los de las redes de interconexión de las computadoras paralelas. Expresado de otra manera, el rendimiento de las redes de comunicación Ethernet no está *balanceado* de acuerdo con la capacidad de procesamiento de las computadoras.
- El método de acceso al "único" medio de comunicaciones definido por el estándar, CSMA/CD (Carrier Sense-Multiple Access/Collision Detect), hace que el rendimiento de la red de interconexión sea altamente dependiente del tráfico y del cableado (con la utilización de switches, por ejemplo).

Por lo tanto, es necesaria una revisión bastante exhaustiva de los algoritmos paralelos para identificar problemas y soluciones en el contexto de este *nuevo* hardware paralelo proporcionado por las redes de computadoras heterogéneas. En ningún caso se debe perder de vista que la razón de ser del procesamiento paralelo es el aumento del rendimiento con respecto al proporcionado por el procesamiento secuencial. La revisión de los algoritmos paralelos tiende a ser caso por caso al menos en términos de áreas de aplicaciones o de problemas a ser resueltos utilizando procesamiento paralelo.

Las aplicaciones de álgebra lineal constituyen una de las grandes áreas de problemas que tradicionalmente han sido resueltos aprovechando el rendimiento que proporcionan las arquitecturas de cómputo paralelo disponibles. Dentro de las aplicaciones del álgebra lineal se han identificado un conjunto de operaciones o directamente rutinas de cómputo que se han considerado básicas y de utilización extensiva en la mayoría de los problemas incluidos dentro de esta área. Tales rutinas se han denominado BLAS (Basic Linear Algebra Subroutines) y tanto para su clasificación como para la identificación de requerimientos de cómputo y de memoria de cada una de ellas se las divide en tres niveles: nivel 1, nivel 2 y nivel 3 (Level 1 o L1 BLAS, Level 2 o L2 BLAS y Level 3 o L3 BLAS). Desde el punto de vista del rendimiento, las rutinas de nivel 3 (L3 BLAS) son las que se deben optimizar para obtener rendimiento cercano al óptimo de cada máquina y de hecho, muchas empresas de microprocesadores estándares proveen bibliotecas BLAS con marcado énfasis en la optimización y el consiguiente rendimiento de las rutinas incluidas en BLAS de nivel 3.

La multiplicación de matrices puede considerarse el pilar o la rutina a partir de la cual todas las demás incluidas en BLAS de nivel 3 se pueden definir. Quizás por esta razón y/o por su simplicidad la mayoría de los reportes de investigación en esta área de procesamiento paralelo comienza por el "problema" de la multiplicación de matrices en paralelo. Expresado de otra manera, al optimizar la multiplicación de matrices de alguna manera se optimiza todo el nivel 3 de BLAS y por lo tanto se tendrían optimizadas la mayoría de las aplicaciones basadas en álgebra lineal y que dependen de la optimización de

las rutinas que llevan acabo las operaciones provenientes del álgebra lineal. Aunque esta optimización no sea necesariamente directa, sí se puede afirmar que el tipo procesamiento que se debe aplicar para resolver la multiplicación de matrices es muy similar al del resto de las rutinas definidas como BLAS de nivel 3 e incluso muy similar también a los problemas más específicos que se resuelven recurriendo a operaciones del álgebra lineal. En este sentido, es muy probable que lo que se haga para optimizar la multiplicación de matrices (en paralelo o no) sea utilizable y/o aprovechable en otras operaciones. Es por esta razón que la orientación de toda esta tesis ha sido a la multiplicación de matrices en paralelo con algunos comentarios hacia la generalización.

Enfocando específicamente la multiplicación de matrices en paralelo, al analizar los algoritmos propuestos hasta ahora se llega a la conclusión de que son bastante orientados hacia las computadoras paralelas tradicionales. De hecho, cada algoritmo paralelo de multiplicación de matrices se puede identificar como especialmente apropiado para las computadoras paralelas de memoria compartida (denominados multiprocesadores) o para las computadoras paralelas con memoria distribuida o de pasaje de mensajes (denominados multicomputadoras).

Los algoritmos paralelos orientados a multiprocesadores no son apropiados para los sistemas de cómputo paralelo de memoria distribuida. Más aún, las redes de computadoras en general (instalaciones Beowulf, sistemas heterogéneos, etc.) son especialmente inapropiadas dado el bajo nivel de acoplamiento o más específicamente la distribución y separación del hardware disponible para cómputo paralelo.

Los algoritmos paralelos orientados a multicomputadoras siguen teniendo una base muy cercana al hardware de las computadoras paralelas tradicionales. Más específicamente, al proponer estos algoritmos se han asumido varias características subyacentes del hardware paralelo tales como:

- Interconexión de los procesadores en forma de malla o toro bidimensional, arreglos de árboles o hipercubos. Esto significa tener la posibilidad de múltiples conexiones punto a punto y múltiples caminos opcionales de la red de interconexión para la transferencia de datos entre dos procesadores.
- Elementos de procesamiento homogéneos. Esto implica que el balance de carga es trivial y directamente dado por la distribución de la misma cantidad de datos de las matrices involucradas a todos los procesadores.

Ninguna de las dos características anteriores es posible de mantener en las redes locales de computadoras heterogéneas. Por lo tanto, es necesario desarrollar algoritmos que hagan uso eficiente de las características de estas *nuevas* arquitecturas paralelas. Por un lado, estos algoritmos deben estar preparados para las diferencias de capacidad de cómputo de las máquinas interconectadas por las redes locales y por el otro deben aprovechar al máximo el rendimiento y las características de las redes de interconexión Ethernet. Y estas son las dos bases sobre las que se apoyan los algoritmos paralelos de multiplicación de matrices propuestos en esta tesis (de hecho, se pueden considerar como dos variantes de un mismo algoritmo paralelo):

- Balance de carga dado por la distribución de datos que a su vez se hace de acuerdo con la capacidad de cálculo relativa de cada computadora.
- Comunicaciones de tipo broadcast únicamente, de tal manera que se aprovecha al

máximo la capacidad de las redes Ethernet.
- Distribución de los datos de manera unidimensional, siguiendo casi de manera unívoca el propio hardware de interconexión física definido por el estándar Ethernet.

Tal como se muestra en el capítulo de experimentación, el sólo hecho de proponer un algoritmo "apropiado" no garantiza obtener rendimiento aceptable ni escalable. De hecho, tal cual lo muestran los resultados de los experimentos, al utilizar la biblioteca de comunicaciones PVM se tiene que agregando máquinas para llevar a cabo cómputo paralelo (aumentar la capacidad de cálculo) y resolver el mismo problema, el rendimiento se reduce. Más aún, dependiendo de las computadoras esta reducción del rendimiento puede ser drástica a punto tal que se tiene peor rendimiento que con una única computadora. En este caso, el tiempo de cómputo paralelo termina dominado por el tiempo necesario para los mensajes broadcast.

Dado que el rendimiento del cómputo local es satisfactorio, es necesario mejorar el rendimiento de los mensajes broadcast para tener rendimiento aceptable para este problema. Tal como se ha discutido, es muy difícil asegurar *a priori* que la implementación de los mensajes broadcast propuestos por las bibliotecas de "propósito general" tales como PVM y las implementaciones de MPI se hagan específicamente para aprovechar la capacidad de broadcast de las redes Ethernet. Por lo tanto, se propone una *nueva* rutina de mensajes broadcast entre procesos basada directamente en el protocolo UDP que es tanto o más utilizado que las propias redes Ethernet. Aunque esta rutina de mensajes broadcast sea extendida a toda una biblioteca de comunicaciones colectivas, en principio el propósito no es definir una *nueva* biblioteca ni reemplazar a las bibliotecas existentes. Sí es necesario el aprovechamiento máximo de las redes Ethernet y por lo tanto la implementación de las rutinas más utilizadas y/o de las que depende el rendimiento se deberían adecuar a las características y capacidades de estas redes de interconexión de computadoras.

El rendimiento de las multiplicaciones de matrices en paralelo es aceptable en las redes locales heterogéneas cuando el algoritmo propuesto específicamente para éstas se implementa utilizando una rutina de broadcast que aprovecha las capacidades de las redes Ethernet. Por lo tanto, y como era de esperar, al menos dos aspectos se combinan desde el punto de vista del rendimiento para obtener el máximo de las redes locales instaladas que se pueden utilizar para cómputo paralelo: algoritmo e implementación en general, y en particular la implementación de los mensajes broadcast. Expresado de otra manera, sin un algoritmo apropiado no se puede obtener buen rendimiento, y aún con un algoritmo apropiado el rendimiento no es satisfactorio si la implementación es inadecuada. En este caso, la parte más problemática de la implementación ha sido la de los mensajes broadcast. Dado que ninguna biblioteca de pasaje de mensajes implementa de manera apropiada los mensajes broadcast o por lo menos no se puede asegurar *a priori* que lo haga, se ha desarrollado una rutina específica que resuelve el problema de rendimiento. Una vez más, se debe recordar que el rendimiento es la razón de ser del procesamiento paralelo en general o como mínimo del procesamiento paralelo que se utiliza para resolver los problemas numéricos en general y las operaciones de álgebra lineal en particular.

La red del LIDI en particular muestra que los algoritmos propuestos también son apropiados para las instalaciones del tipo Beowulf, y/o con hardware de procesamiento homogéneo y red de interconexión con mejor rendimiento que el de las redes locales

instaladas. En el caso de las computadoras paralelas *tradicionales* la utilización no es tan inmediata o incondicional. En los multiprocesadores, *a priori* parece innecesaria la utilización de un "nuevo" algoritmo paralelo dado que los propuestos son muy adecuados o por lo menos más adecuados que cualquiera propuesto para multicomputadoras. En el caso de las multicomputadoras se debe ser muy cuidadoso sobre todo en la implementación y rendimiento de los mensajes broadcast. En este sentido, las redes de interconexión estáticas (con enlaces punto a punto limitados y predefinidos) suelen imponer ciertos límites a la escalabilidad y consecuente rendimiento de los mensajes broadcast. En todo caso, los múltiples esfuerzos de investigación en la dirección de mejorar el rendimiento de las comunicaciones colectivas y de los mensajes broadcast en particular en estas computadoras es aprovechable por los algoritmos propuestos para multiplicar matrices en paralelo.

La experimentación que se llevó a cabo para comparar los algoritmos propuestos para la multiplicación y factorización LU de matrices con respecto a los que implementa ScaLAPACK fue muy satisfactoria. De hecho, la ganancia mínima de los algoritmos propuestos es mayor al 20% del rendimiento obtenido con ScaLAPACK, es decir que con el mismo hardware y el mismo código de cómputo local totalmente optimizado, los algoritmos propuestos obtienen en todos los casos más del 20% mejor rendimiento que ScaLAPACK. Se debe recordar que se considera a ScaLAPACK como una de las bibliotecas que implementa los mejores algoritmos paralelos para el área de álgebra lineal, tanto en rendimiento paralelo como en escalabilidad.

Desde el punto de vista del problema de multiplicación de matrices resuelto (y aún considerando el problema de factorización LU de matrices) se deben tener en cuenta dos consideraciones muy importantes:

- El problema no es significativo en sí mismo, dado que es muy poco frecuente que todo lo que se necesita resolver sea una multiplicación de matrices. Normalmente la multiplicación de matrices es parte de o se utiliza entre otras operaciones para resolver un problema en general.
- Tal como se ha explicado al principio, la multiplicación de matrices es representativa en cuanto a procesamiento de todo el nivel 3 de BLAS y por lo tanto lo que se obtiene con la multiplicación de matrices se puede utilizar en todas las rutinas incluidas en el nivel 3 de BLAS. Dado que en general lo más importante en cuanto a rendimiento se relaciona con estas rutinas (L3 BLAS), la optimización de la multiplicación de matrices se transforma en un aporte significativo para todas o la mayoría de las aplicaciones de álgebra lineal. Esta ha sido la tendencia en general, sea en procesadores secuenciales, computadoras paralelas en general y multiprocesadores, multicomputadoras y/o redes locales de computadoras en particular.

Dado que

- la comunicación entre procesos se resuelve siempre con mensajes broadcast
- la implementación de estos mensajes se hizo aprovechando las capacidades de las redes Ethernet

el rendimiento es escalable al menos hasta el límite dado por la granularidad mínima. De todas maneras, no se debe olvidar que esta granularidad mínima es bastante grande en el caso de las redes locales y dependiente del hardware de comunicaciones (10 Mb/s, 100 Mb/s, 1 Gb/s, etc.).

Desde otro punto de vista, la misma rutina de mensajes broadcast basada en UDP que se ha implementado muestra que no necesariamente se debe trasladar la heterogeneidad del hardware de cómputo de las redes locales al rendimiento de las comunicaciones. Más específicamente:

- El ancho de banda asintótico y/o el tiempo de transferencia de mensajes relativamente grandes es independiente de las computadoras involucradas y depende de la capacidad de las redes de comunicaciones.
- El tiempo de latencia de las comunicaciones es dependiente de la capacidad de cómputo de las máquinas involucradas en una transferencia de datos.

Tal como se ha mostrado tanto en la experimentación con la multiplicación de matrices misma, como en el Apéndice C específicamente para los mensajes punto a punto; las bibliotecas de comunicaciones de propósito general tales como PVM y las implementaciones de uso libre de MPI tienen la tendencia a hacer el rendimiento de las comunicaciones dependiente de la heterogeneidad de las computadoras. Esto se debe a las capas de software que deben agregar para resolver las múltiples rutinas de comunicaciones entre procesos que normalmente implementan y que implican una sobrecarga (overhead) considerable de procesamiento.

Más aún, la misma rutina de mensajes broadcast muestra que es posible tener un mensaje broadcast entre procesos de usuario que cumpla con:

- Rendimiento cercano al óptimo absoluto proporcionado por el hardware de comunicaciones. Aunque la rutina está destinada a los mensajes broadcast el rendimiento de las comunicaciones punto a punto (entre dos computadoras) también es altamente satisfactorio.
- Rendimiento escalable, es decir que el tiempo de broadcast es *casi* independiente de la cantidad de máquinas involucradas. Evidentemente la sincronización y la forma utilizada para confirmación de la llegada de los mensajes a cada computadora implican la existencia de un costo por computadora que interviene en un mensaje broadcast, pero este costo es mucho menor en tiempo de ejecución que la replicación completa de todo el mensaje a cada proceso (máquina) receptor.
- Portabilidad, dado que los únicos requisitos para que esta rutina se pueda utilizar son la conectividad IP (TCP y UDP) y un compilador del lenguaje C. De hecho, al utilizar protocolos estándares y utilizados extensivamente hasta se logra independencia del hardware de comunicaciones. Aunque inicialmente orientada al aprovechamiento de las redes Ethernet, la rutina para llevar a cabo mensajes broadcast es portable a cualquier ambiente con conectividad IP. Aunque no se han hecho pruebas específicas, es bastante probable que en redes de interconexión de computadoras que no tienen la posibilidad de broadcast de datos en el hardware (tales como las redes ATM), el rendimiento de esta rutina de todas maneras sea satisfactorio.
- Ningún requisito adicional desde el punto de vista de un usuario de las redes locales de computadoras que se utilizan para cómputo paralelo. En particular, no son necesarias alteraciones del sistema operativo ni prioridades o procesos con prioridades más allá de las disponibles para los procesos de usuario.
- Sencillez de utilización. De hecho, en los programas con los cuales se realizó la experimentación los cambios a nivel de código fuente no fueron mucho más allá de el reemplazo de la rutina de PVM utilizada para los mensajes broadcast. Todo el resto de las comunicaciones (que no tienen influencia sobre el rendimiento o su influencia es mínima) se continuaron haciendo con rutinas provistas por PVM.

- Manejo de la heterogeneidad en la representación de los datos de las computadoras que normalmente están interconectadas en una red local. En la misma experimentación se utilizaron diferentes tipos de máquinas con diferentes procesadores y sus propias representaciones de los tipos de datos numéricos.
- Interfase de utilización común a la de las demás bibliotecas de pasaje de mensajes de propósito general. De hecho, la implementación de esta rutina hace suponer que las demás rutinas que comúnmente se incluyen dentro de las comunicaciones colectivas es relativamente simple como para tener una biblioteca completa de comunicaciones colectivas.

El algoritmo que resuelve las multiplicaciones de matrices en paralelo con los períodos de procesamiento local y de mensajes broadcast de manera secuencial es simple y confiable en cuanto a estimación de rendimiento. En este sentido, se tiene un modelo de máquina paralela que es capaz de

- Ejecutar simultáneamente en cada procesador tal como cualquier otra perteneciente a la clase MIMD de memoria distribuida. No hay ningún tipo de interferencia entre distintos procesadores (máquinas) para resolver cómputo local.
- Llevar a cabo mensajes broadcast de manera relativamente independiente de la cantidad de máquinas involucradas.
- La interferencia de los mensajes sobre el rendimiento de cómputo local es poco significativa.
- No hay interferencia del cómputo local sobre el rendimiento de las comunicaciones.

Y por otro lado, se tiene un algoritmo de cómputo paralelo que además de aprovechar todas estas características involucra un tipo de procesamiento que es altamente regular. Si bien no se puede asegurar que todos los problemas numéricos tienen procesamiento tan regular, sí se puede afirmar que es una característica similar de una gran parte de las rutinas y aplicaciones provenientes del álgebra lineal. La combinación de este modelo de máquina con este tipo de algoritmos paralelos hace muy sencilla y relativamente confiable la estimación de rendimiento que se puede obtener. Quizás como una consecuencia de esto, también es posible identificar con bastante claridad cuándo comienzan los problemas de rendimiento debido a la granularidad de los problemas que se resuelven. Por lo tanto, el propio programa de multiplicación de matrices en paralelo con los períodos de cómputo y comunicaciones ejecutados u organizados de manera secuencial puede ser utilizado como un benchmark para la identificación de la granularidad mínima de un conjunto de computadoras interconectadas en una red local.

Más allá de obtener mejor rendimiento, el algoritmo de multiplicación de matrices en paralelo que está diseñado para solapar cómputo con comunicaciones es particularmente útil para identificar problemas de rendimiento. Específicamente, con la implementación de este algoritmo es posible identificar con claridad las computadoras que efectivamente pueden solapar cómputo con comunicaciones y hasta cuál es la penalización en términos de rendimiento que esto produce. En este sentido, en los ambientes heterogéneos se pueden tener distintas penalizaciones en diferentes máquinas y con la cuantificación de esta penalización se puede mejorar el balance de carga para compensar las diferencias. Una herramienta de este tipo se torna valiosa cuando funciona en múltiples computadoras y proporciona información que es muy difícil de obtener por otros medios.

Las computadoras involucradas en una red local quizás son las que tienen menor capacidad

tanto en procesamiento como en memoria principal instalada entre todas las disponibles en el mercado. En este sentido, la ganancia obtenida por el uso de una red local procesando en paralelo puede ser muy grande. La experimentación que se hizo involucrando problemas que iban más allá de la capacidad de almacenamiento de la mejor computadora de cada red local intenta cuantificar esta ganancia. La idea básica en esta dirección es: aún utilizando la mejor computadora de una red local pueden haber problemas de rendimiento dado que no es suficiente para resolver un problema dado, principalmente por la cantidad de memoria disponible en esa computadora. Si bien gracias al manejo de memoria swap es posible almacenar una gran cantidad de datos más allá de la memoria instalada, el rendimiento puede sufrir una gran penalización. Por lo tanto, la utilización de las demás computadoras de la red local no solamente proveen memoria para almacenar datos sino que también permiten que *todas* las computadoras lleven a cabo su procesamiento a la máxima velocidad. Es decir que en todas las computadoras se pueden aprovechar los recursos disponibles de manera óptima u optimizada.

Específicamente en términos del valor de speedup como métrica de rendimiento se ha mostrado algo que es relativamente sencillo pero muy poco frecuente en cuanto a los reportes de investigación: el rendimiento en los ambientes heterogéneos no está directamente relacionado con la *cantidad* de computadoras (o procesadores) que se utilizan. En este sentido, las máquinas paralelas tradicionales, con su hardware de procesamiento homogéneo ha establecido que el máximo valor de speedup posible de obtener es igual a la cantidad de procesadores que se utilizan. En los ambientes heterogéneos esto queda sin sustento dado que los procesadores no necesariamente tienen la misma capacidad de cálculo. De hecho, la recta $y = x$ con la que se ha relacionado tradicionalmente el máximo valor de speedup ha permitido la interpolación de valores intermedios y esta interpolación de valores intermedios también queda sin sustento en los ambientes de procesamiento paralelo con procesadores heterogéneos.

Una de las bases para obtener rendimiento satisfactorio y *predecible* con procesamiento paralelo es la utilización del mejor código secuencial para cómputo local. Además, si se utiliza código de cómputo local no optimizado se llega a que la estimación de rendimiento dada por el factor de speedup pierde casi todo su significado, dado que el rendimiento paralelo se obtiene como una combinación de

- Rendimiento local de cada computadora.
- Cantidad de operaciones que se pueden realizar simultáneamente.
- Rendimiento de las comunicaciones.

Se muestra con bastante detalle en el Apéndice B que cuando se utiliza código no optimizado el rendimiento de cada computadora es altamente dependiente del tamaño del problema, básicamente por la relación que existe entre la cantidad de datos a procesar y la capacidad de la memoria cache de los procesadores (específicamente de la memoria cache de primer nivel). En general, en todas las arquitecturas paralelas de memoria distribuida y en el caso particular de las redes locales de computadoras que se utilizan para cómputo paralelo, aumentar la cantidad de procesadores implica que cada procesador tiene problemas cada vez menores en cuanto a cantidad de datos a procesar. Esta menor cantidad de datos tiene una mayor probabilidad de aprovechar mejor el espacio de memoria cache y por lo tanto el rendimiento de cómputo se mejora notablemente. Así se llega a que cuando las rutinas de de cómputo local no son optimizadas el rendimiento paralelo no

necesariamente mejora por utilizar más computadoras sino porque cada computadora resuelve un problema con menor cantidad de datos y por lo tanto el rendimiento de cómputo local es significativamente mayor. Por otro lado, el código de cómputo totalmente optimizado hace que el rendimiento sea relativamente independiente del tamaño de problema que se resuelve y por lo tanto toda ganancia obtenida por cómputo paralelo es

- "Real", dado que no hay otra forma de obtener mejor rendimiento de las computadoras secuenciales que se utilizan porque el rendimiento secuencial con el que se compara es el óptimo.
- Debida únicamente a la utilización de mayor cantidad de computadoras o procesadores, ya que el tamaño del problema no influye significativamente en el rendimiento local de cada máquina.

6.2 Resumen de Aportes y Publicaciones Relacionadas con Esta Tesis

Se pueden enumerar de manera resumida los aportes de esta tesis también relacionados con las publicaciones que se han hecho al respecto. Inicialmente, se debe identificar con cierta precisión el problema básico de rendimiento paralelo en las redes locales de computadoras y clusters y proponer algún tipo de solución. Estos dos aportes iniciales puede ser resumidos como:

1. **Análisis de los algoritmos de multiplicación de matrices en paralelo para su utilización en redes locales de computadoras que se pueden aprovechar para cómputo paralelo.**
2. **Propuesta de los principios de paralelización que se utilizaron para diseñar los algoritmos propuestos en esta tesis.**

Y estos aportes están directamente relacionados con las publicaciones:

- [135] Tinetti F., A. Quijano, A. De Giusti, "Heterogeneous Networks of Workstations and SPMD Scientific Computing", 1999 International Conference on Parallel Processing, The University of Aizu, Aizu-Wakamatsu, Fukushima, Japan, September 21 - 24, 1999, pp. 338-342.
- [137] Tinetti F., Sager G., Rexachs D., Luque E., "Cómputo Paralelo en Estaciones de Trabajo no Dedicadas", VI Congreso Argentino de Ciencias de la Computación, Ushuaia, Argentina, Octubre de 2000, Tomo II, pp. 1121-1132.

Donde se presenta experimentación específicamente orientada a mostrar que los algoritmos tradicionales no necesariamente son útiles en las redes locales de computadoras. Por otro lado, las publicaciones (en orden cronológico):

- [116] Tinetti F., "Aplicaciones Paralelas de Cómputo Intensivo en NOW Heterogéneas", Workshop de Investigadores en Ciencias de la Computación (WICC 99), San Juan, Argentina, 27 y 28 de Mayo de 1999, pp. 17-20.
- [117] Tinetti F., "Performance of Scientific Processing in Networks of Workstations", Workshop de Investigadores en Ciencias de la Computación (WICC 2000), La Plata, Argentina, 22 y 23 de Mayo de 2000, pp. 10-12.
- [124] Tinetti F., Barbieri A., Denham M., "Algoritmos Paralelos para Aprovechar Redes

Locales Instaladas", Workshop de Investigadores en Ciencias de la Computación (WICC 2002), Bahía Blanca, Argentina, 17-18 de Mayo de 2002, pp. 399-401.
- [128] Tinetti F., Denham M., "Algebra Lineal en Clusters Basados en Redes Ethernet", Workshop de Investigadores en Ciencias de la Computación (WICC 2003), Tandil, Argentina, 22-23 de Mayo de 2003, pp. 575-579.
- [134] Tinetti F., Quijano A., "Costos del Cómputo Paralelo en Clusters Heterogéneos", Workshop de Investigadores en Ciencias de la Computación (WICC 2003), Tandil, Argentina, 22-23 de Mayo de 2003, pp. 580-584.

Están más orientadas a presentar las ideas como líneas de investigación abiertas y/o en desarrollo. Se debe notar que cada uno de los años en que se ha participado en este congreso se han reportado los avances de la línea de investigación con respecto al año anterior.

Una vez identificados los inconvenientes y alguna propuesta de solución en general, es necesario probar la propuesta. La alternativa elegida ha sido hacerlo de forma específica en el área de las aplicaciones de álgebra lineal y de las operaciones básicas. En este contexto se aporta en esta tesis:

3. **Propuesta de algoritmos específicos de multiplicación de matrices en clusters, diseñados siguiendo los principios de paralelización mencionados antes.**
4. **Utilización del algoritmo de multiplicación de matrices en paralelo que está diseñado para solapar cómputo con comunicaciones para identificar problemas de rendimiento (como *benchmark*, en cierta forma).**

Estos algoritmos han sido presentados junto con experimentación que avala su validez en las publicaciones:
- [118] Tinetti F., "Performance of Scientific Processing in NOW: Matrix Multiplication Example", JCS&T, Journal of Computer Science & Technology, Special Issue on Computer Science Research, Vol. 1 No. 4, March 2001, pp. 78-87.
- [131] Tinetti F., Luque E., "Parallel Matrix Multiplication on Heterogeneous Networks of Workstations", Proceedings VIII Congreso Argentino de Ciencias de la Computación (CACIC), Fac. de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Buenos Aires, Argentina, 15 al 18 de Octubre de 2002, p. 122.
- [132] Tinetti F., Luque E., "Efficient Broadcasts and Simple Algorithms for Parallel Linear Algebra Computing in Clusters", Workshop on Communication Architecture for Clusters, International Parallel and Distributed Processing Symposium (IPDPS '03), Nice Acropolis Convention Center, Nice, France April 22-26, 2003.
- [136] Tinetti F., A. Quijano, A. De Giusti, E. Luque, "Heterogeneous Networks of Workstations and the Parallel Matrix Multiplication", Recent Advances in Parallel Virtual Machine and Message Passing Interface, 8th European PVM/MPI Users' Group Meeting, Santorini/Thera, Greece, September 23-26, 2001, Proceedings, Yannis Cotronis, Jack Dongarra (Eds.), Lecture Notes in Computer Science 2131 Springer 2001, ISBN 3-540-42609-4, pp. 296-303.

En muchas de las publicaciones anteriores también se presenta otro de los aportes de esta tesis, específicamente orientado al aprovechamiento de las redes Ethernet, aporte que se puede resumir en:

5. Propuesta de una rutina de mensajes broadcast basada en el protocolo UDP para optimizar la utilización de las redes Ethernet.

En este caso, las publicaciones más específicamente relacionadas son:

- [120] Tinetti F., Barbieri A., "Collective Communications for Parallel Processing in Networks of Workstations", Proceedings SCI 2001, Volume XIV, Computer Science and Engineering: Part II, Nagib Callaos, Fernando G. Tinetti, Jean Marc Champarnaud, Jong Kun Lee, Editors, International Institute of Informatics and Systemics, Orlando, Florida, USA, ISBN 980-07-7554-4, July 2001, pp. 285-289.
- [123] Tinetti F., Barbieri A., "An Efficient Implementation for Broadcasting Data in Parallel Applications over Ethernet Clusters", Proceedings of the 17th International Conference on Advanced Information Networking and Applications (AINA 2003), IEEE Press, ISBN 0-7695-1906-7, March 2003.

También en esta tesis se tratan aspectos de rendimiento de cómputo paralelo en clusters homogéneos que se pueden resumir como:

6. Propuesta de algoritmos específicos de multiplicación de matrices y factorización LU de matrices en clusters homogéneos, diseñados siguiendo los principios de paralelización mencionados antes. En realidad, el algoritmo de multiplicación de matrices es el mismo que el presentado para los clusters heterogéneos, mostrando de esta manera su utilización directa en clusters homogéneos.

Estos algoritmos en el contexto de los clusters homogéneos se presentaron junto con experimentación específica y/o de comparación con ScaLAPACK en algunas de las publicaciones anteriores y en:

- [127] Tinetti F., Denham M., "Paralelización de la Factorización de Matrices en Clusters", Proceedings VIII Congreso Argentino de Ciencias de la Computación (CACIC), Fac. de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Buenos Aires, Argentina, 15 al 18 de Octubre de 2002, p. 121.
- [130] Tinetti F., Denham M., De Giusti A., "Parallel Matrix Multiplication and LU Factorization on Ethernet-based Clusters", High Performance Computing. 5th International Symposium, ISHPC 2003, Tokyo-Odaiba, Japan, October 20-22, 2003, Proceedings. Series: Lecture Notes in Computer Science, Vol. 2858. Veidenbaum, A.; Joe, K.; Amano, H.; Aiso, H. (Eds.), 2003, XV, 566 p. ISBN: 3-540-20359-1
- [129] Tinetti F., Denham M., "Paralelización de la Factorización LU de Matrices en Clusters Heterogéneos", Proceedings IX Congreso Argentino de Ciencias de la Computación (CACIC), Fac. de Informática, Universidad Nacional de La Plata, La Plata, Argentina, 6 al 10 de Octubre de 2003, p. 385-396.

Donde la última publicación muestra los primeros resultados obtenidos al utilizar los principios de paralelización para la factorización LU de matrices en clusters heterogéneos.

Aunque la evaluación de las comunicaciones es bastante conocida, en esta tesis tiene especial relevancia dado que se ha mostrado la penalización excesiva que puede llegar a

imponer sobre algoritmos paralelos específicamente diseñados para la obtención de rendimiento optimizado. También se presenta en el Apéndice C toda la metodología y los resultados obtenidos en términos de

7. **Evaluación de rendimiento de las comunicaciones desde la perspectiva de cómputo paralelo en clusters heterogéneos (operaciones punto a punto y colectivas).**

Que se ha reflejado en las publicaciones:

- [119] Tinetti F., Barbieri A., "Cómputo y Comunicación: Definición y Rendimiento en Redes de Estaciones de Trabajo", Workshop de Investigadores en Ciencias de la Computación (WICC 2001), San Luis, Argentina, 22-24 de Mayo de 2001, pp. 45-48.
- [121] Tinetti F., Barbieri A., "Análisis del Rendimiento de las Comunicaciones sobre NOWs", Proceedings VII Congreso Argentino de Ciencias de la Computación (CACIC), El Calafate, Santa Cruz, Argentina, 16 al 20 de Octubre de 2001, pp. 654-656.
- [122] Tinetti F., Barbieri A., "Cómputo Paralelo en Clusters: Herramienta de Evaluación de Rendimiento de las Comunicaciones", Proceedings VIII Congreso Argentino de Ciencias de la Computación (CACIC), Fac. de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Buenos Aires, Argentina, 15 al 18 de Octubre de 2002, p. 123.
- [125] Tinetti F., D' Alessandro A., Quijano A., "Communication Performance of Installed Networks of Workstations for Parallel Processing", Proceedings SCI 2001, Volume XIV, Computer Science and Engineering: Part II, Nagib Callaos, Fernando G. Tinetti, Jean Marc Champarnaud, Jong Kun Lee, Editors, International Institute of Informatics and Systemics, Orlando, Florida, USA, ISBN 980-07-7554-4 July 2001, pp. 290-294.
- [133] Tinetti F., Quijano A., "Capacidad de Comunicaciones Disponible para Cómputo Paralelo en Redes Locales Instaladas", Proceedings VIII Congreso Argentino de Ciencias de la Computación (CACIC), Fac. de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Buenos Aires, Argentina, 15 al 18 de Octubre de 2002, p. 125.

Aunque no directamente relacionado con el contexto de las redes locales instaladas se han llevado a cabo algunos estudios relacionados con el rendimiento de la multiplicación de matrices en supercomputadoras o al menos computadoras paralelas tradicionales, que se ha publicado en

- [126] Tinetti F., Denham M., "Paralelización y Speedup Superlineal en Supercomputadoras. Ejemplo con Multiplicación de Matrices", Proceedings VII Congreso Argentino de Ciencias de la Computación (CACIC), El Calafate, Santa Cruz, Argentina, 16 al 20 de Octubre de 2001, pp. 765-774.

Muchas de las conclusiones a las que se llega en esta publicación tiene relación directa con el Apéndice B, dedicado a mostrar el rendimiento secuencial de las computadoras utilizadas. En particular, la noción distorsionada de rendimiento que se puede llegar a obtener cuando el código de los programas utilizados no son optimizados específicamente para la aplicación resuelta y la arquitectura de cómputo utilizadas.

6.3 Trabajo Futuro

Como se explica antes, el problema de la multiplicación de matrices no es significativo en sí mismo sino representativo de un conjunto de problemas de procesamiento numérico de datos. En el contexto de las rutinas BLAS de nivel 3, la extensión inmediata es:

- Utilizar directamente la multiplicación de matrices para la implementación de todas las rutinas incluidas en BLAS de nivel 3.
- Utilizar los principios de paralelización utilizados en la multiplicación de matrices para resolver las demás rutinas de BLAS nivel 3.

La primera opción tiene la ventaja de no tener mucho más costo que codificar las rutinas en términos de multiplicaciones de matrices. Aunque la segunda opción no tiene la ventaja anterior sino que involucra el costo asociado de paralelización caso por caso, tiene la ventaja de permitir un rango más amplio de posible ganancia debida al cómputo paralelo. Tal cual están definidas, las rutinas BLAS de nivel 3 son una cantidad bastante reducida y en la paralelización caso por caso se pueden aprovechar mejor las características propias de procesamiento para obtener mejor rendimiento. Cualquiera sea la alternativa elegida, o incluso en la experimentación con ambas, el aprovechamiento de las ideas o principios de paralelización de esta tesis es bastante directo.

Como un paso un poco mayor en cuanto a la extensión de esta tesis es directamente atacar un problema completo proveniente del álgebra lineal. Como un ejemplo se puede mencionar el mismo método de factorización LU presentado en el Capítulo 5, que se puede utilizar para la resolución del problema de sistemas de ecuaciones lineales. En un contexto un poco más general se podría avanzar en la dirección de los problemas que resuelve la biblioteca LAPACK como para experimentar con una gama relativamente amplia de problemas provenientes del álgebra lineal. La ventaja asociada a la experimentación con LAPACK es que la biblioteca misma ha sido utilizada hasta el momento y por lo tanto existe una cantidad relativamente grande de usuarios potenciales. Se puede afirmar que hasta este punto, es decir manteniéndose en operaciones y aplicaciones de álgebra lineal el tipo de procesamiento, es bastante similar respecto del procesamiento de la multiplicación de matrices. Si bien existen muchas particularidades, la gran mayoría de las operaciones son:

- Bastante sencillas en cuanto a codificación.
- Muy conocidas en cuanto a métodos de solución.
- Con un alcance muy bien definido de dependencia de datos y también con subconjuntos de datos que se pueden calcular de manera independiente.

Esta extensión del trabajo está avalada por el hecho de haber encontrado que la paralelización de operaciones como la multiplicación y la factorización de matrices con los principios relativamente sencillos de esta tesis proporciona código optimizado para las redes locales interconectadas por Ethernet. De hecho, la experimentación que se llevó a cabo con el objetivo de comparar los algoritmos propuestos con los implementados por ScaLAPACK avalan esta línea de investigación futura.

El siguiente nivel de extensiones, un poco más complejo, lo representan las aplicaciones numéricas en general y en particular todo lo que involucra procesamiento no lineal. Es más

complejo desde dos puntos de vista:

- Codificación de los métodos de solución de problemas específicos.
- Relaciones de dependencia de cálculos, que no son tan estructuradas como en la mayoría de las operaciones de álgebra lineal.

Un área específica es la de procesamiento de señales, que tiene múltiples aplicaciones y donde los métodos de solución a los problemas específicos son numerosos y muchas veces muy dispares entre sí. El cálculo conocido y relativamente simple en este contexto de una FFT (Fast Fourier Transform) involucra por ejemplo un patrón de acceso a datos que es en cierta forma regular pero tan específico que ha dado lugar directamente a modos de direccionamiento de datos *ad hoc* en los procesadores diseñados para procesamiento de señales digitales o DSP (Digital Signal Processor). Si bien los principios de paralelización en esta área son los mismos, dado que están pensados para el aprovechamiento optimizado de los recursos de cómputo de las redes locales y no para un área de procesamiento en particular, la aplicación de estos principios no es tan sencilla como en el caso de la multiplicación de matrices o las demás operaciones o rutinas relacionadas con álgebra lineal.

En otro eje de investigación, siempre es posible pensar extensiones o al menos experimentar con la posibilidad de utilización de más de una red local. En este sentido, y para las aplicaciones de álgebra lineal con sus características de procesamiento fuertemente acoplado, es muy importante hasta qué punto es posible la ganancia de rendimiento con la utilización de más de una red local. Más específicamente, la cuantificación de la penalización (por ejemplo en cuanto a granularidad mínima) por la distribución de los datos en múltiples redes locales es útil para caracterizar *a priori* la utilidad de uso de más de una red local para solucionar un problema en paralelo.

En el caso de múltiples redes locales también puede ser significativo el aporte de otros métodos simples pero efectivos de procesamiento tales como el *pipelining* (similar a una línea de ensamblaje tradicional) o el establecimiento de “servidores” específicos para tareas especialmente penalizadas por la distribución física de las computadoras que se utilizan. Se debe tener especial cuidado en este contexto con todas las comunicaciones que sean *remotas* en el sentido de transferir datos entre dos o más computadoras que pertenecen a distintas redes locales. El caso específico de los mensajes broadcast, por ejemplo, sigue siendo sumamente útil y sencillo en una red local y en todas las redes locales que se utilizan, pero la implementación de estos mensajes cuando están involucradas varias computadoras de distintas redes locales se debe pensar con mucho cuidado en cuanto a tráfico, congestión (competencias) de los enlaces intermedios de transporte entre redes, tiempo de latencia, etc. La estrategia a seguir ya no es tan inmediata, aunque el rendimiento tan satisfactorio que se puede obtener en cada red local en cierta forma favorece esta línea de investigación.

La utilización de las tres redes locales sobre las que se llevó a cabo toda la experimentación sigue siendo posible y podrían diseñarse un conjunto de experimentos para analizar los resultados y a partir de allí proponer alternativas de aprovechamiento de cada una de las computadoras. De alguna manera, la posibilidad de utilizar más de una red local aumenta considerablemente el rango de tamaños de problemas que se pueden resolver (independientemente de que el problema sea multiplicar matrices o cualquier otro) pero también agrega problemas bastante “desconocidos” al menos en este contexto de las

aplicaciones de álgebra lineal tales como el impacto sobre la granularidad mínima y la escalabilidad ahora a nivel de redes locales. Otro de los problemas en este contexto es el de rendimiento vs. capacidad de almacenamiento en memoria principal: ¿Es preferible una red local con mayor capacidad de almacenamiento en memoria principal o con mayor capacidad de procesamiento? Es bastante probable que las redes locales que tienen mayor capacidad de almacenamiento (sumando las capacidades de cada una de las computadoras interconectadas) sean también las de mayor capacidad de procesamiento, pero esto no se puede asegurar dado que las redes locales no necesariamente están diseñadas para cómputo paralelo y aún más para cómputo paralelo con otras redes.

En una extensión de esta tesis que se podría denominar “a gran escala” se pueden combinar los dos tipos de extensiones que se han mencionado hasta este punto:

- Extensión en cuanto a otros problemas a resolver
- Extensión en cuanto a la utilización de mayor cantidad de máquinas a utilizar involucrando más de una red local.

Quizás en ambos casos los problemas serán mucho mayores con respecto a procesamiento necesario como a cantidad de datos a procesar, pero se pueden seguir utilizando al menos inicialmente los principios básicos de paralelización de matrices. En todo caso, a partir de los problemas que se identifican vía experimentación se pueden proponer otros más específicos y apropiados para obtener el máximo rendimiento posible a partir de los recursos disponibles.