

Qué se Puede Lograr Utilizando Programación Paralela

Jorge Luis Ortega Arjona
IIMAS-UNAM

Las arquitecturas paralelas representan una solución para aquellos problemas de cómputo en los que la velocidad de respuesta es preponderante. Sin embargo, actualmente presentan la desventaja de contar con un software limitado que los respalde. Este artículo se plantea como una introducción práctica al procesamiento paralelo, y muestra los resultados de algunas métricas utilizando la arquitectura Occam-Transputer.

1. Introducción

"Si uno es bueno, entonces 10 ó 1,000 debe ser mejor" [1]. Esta afirmación básica que en general se considera cierta representa la atracción primaria hacia los sistemas de procesamiento paralelo.

Sin embargo, a pesar de las posibilidades que ofrece un sistema de procesamiento paralelo, actualmente son pocos quienes lo consideran como una solución viable a los problemas de cómputo. Esto se debe, tal vez, a que aunque generalmente se acepta que una arquitectura paralela proporciona un desempeño mayor que un sistema secuencial tradicional, la mayoría de quienes trabajan con sistemas de cómputo no conocen, siquiera de una manera aproximada, los alcances de un sistema de procesamiento paralelo.

La utilización de una arquitectura paralela para la solución de problemas complejos es una realidad hoy día debido al aumento en el desempeño y disminución de costos del hardware, y al desarrollo en los últimos años (mediante la aportación de un grupo numeroso de investigadores) de una teoría alrededor del procesamiento paralelo, procurando soluciones a los problemas que surgen de la complejidad de un sistema de múltiple procesamiento. Tales esfuerzos han contribuido al diseño de varias arquitecturas paralelas, entre las que se encuentra la arquitectura *Occam-Transputer*.

Desarrollado a partir del lenguaje de especificación formal C.S.P., propuesto por C.A.R. Hoare en 1978 [3], Occam representa el lenguaje de programación que realiza de una manera natural la expresión abstracta del paralelismo, mientras que la arquitectura Transputer provee de un soporte físico específico a Occam. Esta relación da como resultado que la arquitectura Occam-Transputer represente una solución poderosa a problemas computacionales complejos, desde un enfoque paralelo.

Este artículo no pretende ser un tratamiento exhaustivo del tema del paralelismo utilizando la arquitectura Occam-Transputer. Los principales elementos de Occam y Transputers han sido tratados con anterioridad en artículos de esta misma publicación [7,9], y de una manera más profunda en manuales y libros de varios autores [4,5,8]. En lugar de eso, este trabajo se plantea

Jorge Luis Ortega Arjona está por obtener el grado de Maestro en Ciencias de la Computación, del IIMAS-UNAM. Actualmente trabaja en el Laboratorio de Procesamiento Paralelo, del Departamento de Electrónica y Automatización, IIMAS-UNAM. Sus áreas de interés son: Procesamiento Paralelo, Programación OO, Procesamiento de Imágenes.

E-mail: jloa@uxdea1.iimas.unam.mx

como una introducción práctica al procesamiento paralelo.

Para lograr esto último, el trabajo se ha organizado en cuatro partes: la primera como una semblanza de algunas métricas de desempeño utilizadas comúnmente para evaluar la operación de un sistema paralelo. La segunda parte consiste propiamente en la descripción y ejecución del ejemplo de programación Occam, obteniendo sus medidas de desempeño. La tercera parte es un breve análisis de las métricas obtenidas en el ejemplo. Finalmente, se expone una conclusión general sobre los conceptos tratados.

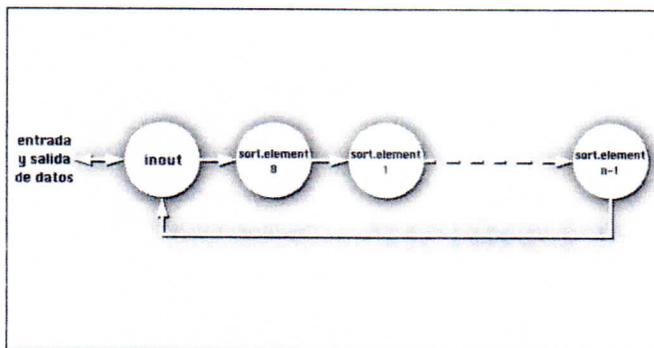


Figura 1. Comunicación entre procesos inout y sort.element

2. Métricas de Desempeño

A fin de proporcionar parámetros de comparación entre las diferentes pruebas del ejemplo a desarrollar, es necesario definir de antemano algunos conceptos relativos a las métricas utilizadas para determinar el desempeño de un sistema paralelo. Primeramente, es necesario establecer qué se considera como *desempeño de un sistema paralelo*.

Una definición apropiada considera al desempeño de un sistema físico como *una medida global de las cualidades de tal sistema, que lo hacen atractivo para su uso*. Aplicando esto a la computación, el desempeño resultaría de observar diversas cualidades deseables de una arquitectura, como el tiempo de ejecución, la velocidad, la eficiencia, el costo, etc. El análisis aquí descrito considera las tres primeras métricas mencionadas. Un tratamiento más detallado de éstas se describe en [2].

2.1 Tiempo de ejecución

En el campo del procesamiento en paralelo se hace especial énfasis en las características de velocidad del sistema, es decir, el tiempo que le lleva al sistema realizar un cierto proceso. Aún cuando se continúan considerando las demás propiedades (como costo y simplicidad), en general consideraremos el desempeño de un sistema en paralelo en base a la cantidad de tiempo que requiera para realizar alguna tarea específica. De esta manera, consideraremos al *tiempo de ejecución* de un programa como una métrica del desempeño del sistema paralelo. Tal

valor, en función del número de procesadores, se especifica como $T(n)$.

2.2 'Speedup'

A partir de la medida de tiempos de ejecución, se obtiene una relación del tiempo de ejecución de un programa paralelo en un solo procesador entre el tiempo de ejecución del mismo programa en n procesadores. A tal relación se le conoce como *speedup*, o $S(n)$. La relación matemática es como sigue:

$$S(n) = \frac{T(1)}{T(n)}$$

Es preciso hacer notar que el *speedup* es una medida del desempeño que únicamente considera como variable la cantidad de procesadores utilizados en cada ejecución. Por lo tanto, otros factores que influyen sobre el desempeño de un programa paralelo, como son la topología, el número de comunicaciones, el número de procesos, etc., se consideran como constantes en un análisis de desempeño por *speedup*.

2.3 Eficiencia

Otra medida del desempeño de la programación paralela, íntimamente ligada a los conceptos de tiempo de ejecución y *speedup*, es la eficiencia. Esta medida se asocia a la creencia que *"N trabajadores deben hacer el trabajo en una fracción 1/N del tiempo que le lleva a un solo trabajador"* [6]. En el caso del procesamiento paralelo, la eficiencia se refiere al hecho que un procesador se encuentre activo la mayor parte del tiempo, es decir, la manera en que se utiliza cada elemento del hardware. Una eficiencia que tiende a un valor de 1, significa que los procesadores son usados efectivamente. Un valor bajo de eficiencia significa que se desperdician recursos. De esta manera, la relación matemática que define a la eficiencia de un sistema paralelo es:

$$E(n) = \frac{S(n)}{n} = \frac{T(1)}{nT(n)}$$

3. Caso de Estudio

Para ilustrar los conceptos descritos anteriormente se considera un programa: El ejemplo consiste en ordenar una serie de caracteres. Una muestra del problema original se encuentra en el manual del usuario de Occam 2 [4]. La solución propuesta se modela como una colección de procesos concurrentes de dos tipos: *inout* y *sort.element*. Estos se comunican entre sí como se muestra en la figura 1.

El proceso *inout* es único, y tiene la función de aceptar los caracteres a ordenar, pasarlos al primer proceso *sort.element*, y terminado el ordenamiento, se encarga de recibir los caracteres ordenados del último proceso *sort.element*, enviándolos a la salida.

Por otra parte, se utilizan n procesos *sort.element*. Cada proceso realiza el ordenamiento de la siguiente manera: primero, al-

macena un primer caracter recibido; segundo, recibe un siguiente caracter, comparándolo con el valor de su caracter almacenado; tercero, envía el caracter con un mayor valor al siguiente proceso. Al terminar el ordenamiento, cada proceso pasa su caracter almacenado y(o) los demás caracteres ordenados de los procesos anteriores al proceso *inout*.

3.1 Implementación

La implementación del caso de estudio se desarrolló en una plataforma de procesamiento paralelo con 16 transputers IMS T805-30, que pertenece al Laboratorio de Procesamiento Paralelo del Departamento de Electrónica y Automatización, IIMAS-UNAM.

Con el fin de observar más claramente el comportamiento del programa del ejemplo, en el proceso de implementación es necesario considerar una serie de lineamientos generales: primero, el

¿Necesita Información y Conectarse a Otras Partes del Mundo? Internet

La Red mas Grande del Mundo
La Red de Redes

SEI_{net}

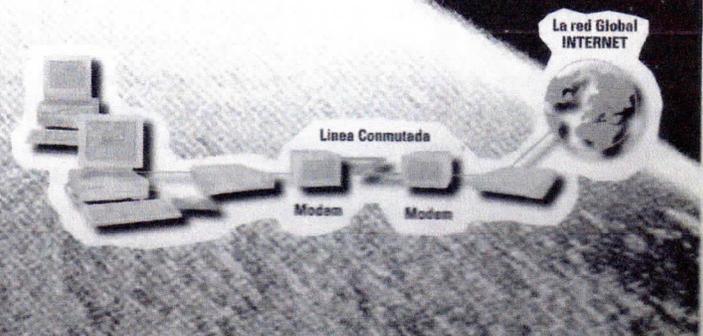
REPCOM DE MEXICO S.A. DE C.V.
Av. Nueva Loma 67 Piso 6
Hipodromo Condasa, Mexico 06140, D.F.
Tel/Fax: (525) 211-2282, 211-2391
Email: internet.staff@sei.net.mx

SEInet da como valor agregado las vías de acceso a medios especializados de investigación a bibliotecas electrónicas, bases de datos y otras fuentes de información.

Los accesos SEInet son dirigidos hacia la red normal o por la red digital de TELMEX en México.

SEInet proporciona direcciones de acceso a la red INTERNET para nuestros clientes, al estar reconocidos como un Commercial Internet Service Provider por el organismo Commercial Internet Exchange (CIX)

Usted podrá estar enlazado a la red SEInet con toda la información disponible por INTERNET, **SIN COSTO ADICIONAL.**



programa se ejecutará manteniendo una topología de anillo, variando únicamente la cantidad de procesadores que lo forman. La topología se refiere a la manera de interconexión entre los procesadores. De ella depende también la respuesta del sistema. Para cada caso, mostraremos un esquema de tal interconexión;

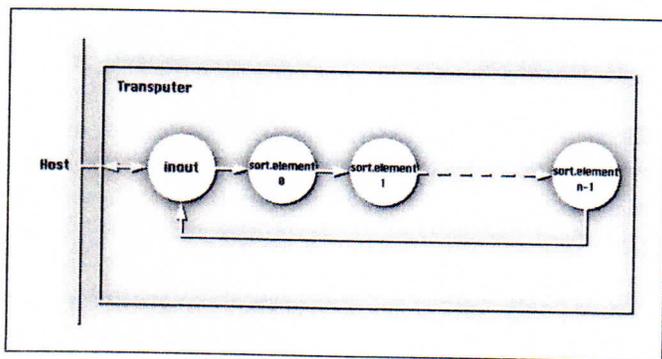


Figura 2. Topología de anillo de los procesos.

segundo, se procura mantener en cada caso un número igual de procesos *sort.element* por procesador, a fin de estandarizar la respuesta. Esto debido a que la distribución del trabajo entre el total de procesadores también tiene injerencia sobre el desempeño de la aplicación; tercero, el proceso *inout* se ejecuta únicamente en el primer procesador del anillo, manteniendo la comunicación con el sistema *host*. Por otra parte, es este proceso quien se encarga además de realizar el cálculo del tiempo de procesamiento, desde que se introducen los caracteres, hasta que salen ordenados. Esto último constituye una variación respecto al ejemplo del manual del usuario de Occam 2 [4]; cuarto, en cada caso se introducen al programa como datos exactamente las mismas cadenas de caracteres; quinto, se realiza varias veces la ejecución del programa, variando en cada caso únicamente el número de procesadores, obteniendo la media estadística del tiempo de ejecución.

Como se menciona, se utiliza una configuración de transputers en anillo. Para el caso de un solo transputer, la configuración resulta muy simple, como se ve en la figura 2.

El *host* se refiere al sistema al que la red de transputers se encuentra conectado, a fin de tener una comunicación al exterior. Es necesario siempre contar con un transputer "raíz", que permita la entrada y salida de los datos.

En el caso de utilizar 16 transputers, la configuración del anillo es como se muestra en la figura 3.

Las configuraciones para 2, 4 y 8 procesadores son variaciones de esta última, es decir, se mantiene un anillo de procesadores para cada caso.

Dado que se trata de un ejemplo ilustrativo sobre el desem-

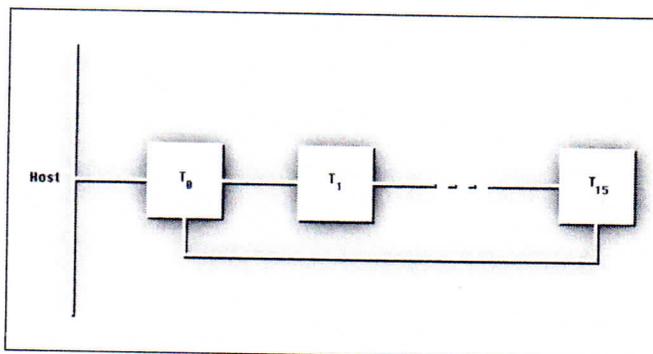


Figura 3. Transputers en anillo

peño de la programación paralela, los códigos en Occam de los procesos involucrados (*inout*, *sort.element* y el programa principal) no se incluyen, debido principalmente a su extensión.

4. Análisis de Desempeño

Los resultados de las mediciones de los tiempos de ejecución para el ordenador de caracteres utilizando 1, 2, 4, 8 y 16 procesadores, se presentan en la tabla 1.

Número de Procesadores	Tiempo de Ejecución Promedio	Speedup	Eficiencia
1	0.084432	1.0	1.0
2	0.067054	1.2591	0.62955
4	0.038344	2.2019	0.5504
8	0.013948	6.0533	0.7566
16	0.007759	10.8814	0.68009

Tabla 1. Tiempos de ejecución

A partir de esta información, es posible generar las gráficas correspondientes. Así, se tiene la gráfica del tiempo de ejecución promedio respecto al número de procesadores, que se muestra en la figura 4.

De manera semejante, se obtienen las gráficas para el *speedup* y la eficiencia, como se muestra en las figuras 5 y 6, respectivamente.

De las gráficas se pueden realizar algunas observaciones respecto al comportamiento del sistema considerando el desempeño en función exclusivamente del número de procesadores. Así, en la gráfica de tiempo de ejecución se puede observar que como era de esperarse, al incrementar el número de procesadores

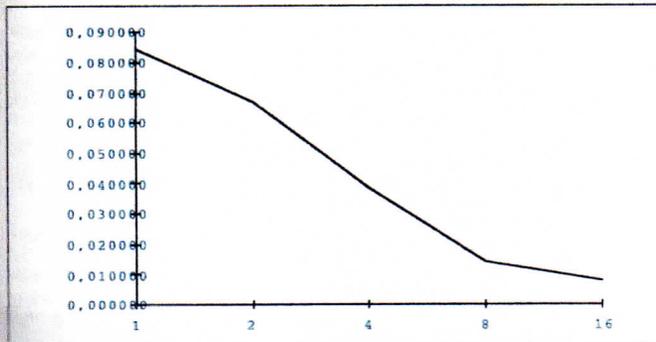


Figura 4. Tiempo de ejecución promedio respecto al número de procesadores.

en una potencia de 2 los tiempos de respuesta del sistema paralelo disminuyen respecto al desempeño concurrente en un solo procesador. Utilizando 16 procesadores en lugar de uno, se obtiene una respuesta mucho más rápida. La comparación entre los tiempos de ejecución se puede observar directamente de la gráfica de *speedup*. Es necesario considerar la relación del *speedup* como una medida de la ganancia que proporciona el sistema paralelo en el desarrollo del proceso. La gráfica resultante muestra que el desempeño del sistema paralelo aumenta respecto al número de procesadores. Por otra parte, la gráfica de la eficiencia provee de una información sobre el aprovechamiento de cada procesador durante la ejecución. En este ejemplo, es curioso que se obtenga una mayor eficiencia para 8 procesadores que para 16.

Tomando en cuenta la razón de cambio de cada gráfica, se observa una aceleración en el comportamiento del sistema al aumentar el número de procesadores, es decir, la variación entre los casos con pocos procesadores (1 y 2) no presentan un gran cambio entre sí. Sin embargo, para los casos con varios procesadores (4, 8 y 16), la razón de cambio aumenta dramáticamente. Una explicación de este comportamiento puede ser que debido a que el primer procesador debe ejecutar también el proceso *inout*, el tiempo de ejecución total se ve más afectado en los casos en los que el número de procesadores es menor, y mejora conforme el número de procesadores aumenta, ya que los procesos *sort.element* se distribuyen uniformemente entre los procesadores.

5. Conclusiones

Las arquitecturas paralelas representan una solución para aquellos problemas de cómputo en los que la velocidad de respuesta es preponderante. Sin embargo, actualmente presentan la

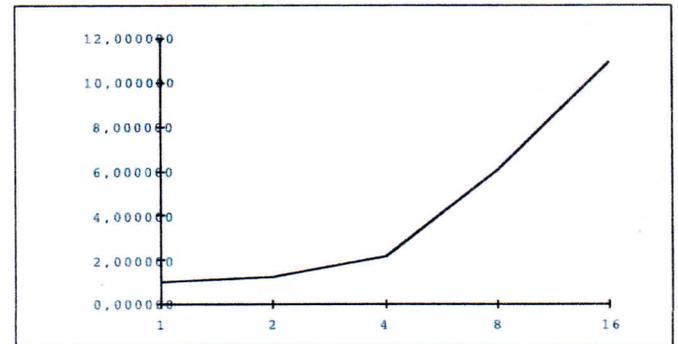


Figura 5. Speedup

desventaja de contar con un software limitado que los respalde. Diversas instituciones realizan esfuerzos a este respecto, para generar resultados a un mediano plazo.

Por otro lado, respecto al ejemplo es posible hacer dos observaciones:

- Los resultados se obtienen en función exclusivamente del número de procesadores, lo que no excluye que existen otros puntos a considerar dentro del desempeño de una arquitectura paralela, como son la topología de interconexión de procesadores, la distribución de los procesos entre los procesadores, la cantidad de comunicaciones entre los procesos, entre otros.
- El concepto de desempeño de una arquitectura paralela debe extenderse no únicamente considerando las características de velocidad del sistema, sino mejorarse con cualidades deseables en la programación, como claridad, simplicidad, eficiencia, etc., lo que daría como resultado una programación más atractiva para quienes requieren sistemas de procesamiento de alto rendimiento.

Por último, observando los resultados se puede suponer que aumentado constantemente el número de procesadores es posible obtener mejores resultados. En realidad, esto no sucede, ya que el comportamiento del desempeño de las arquitecturas paralelas en algunas otras aplicaciones, resulta menor que si se utiliza una arquitectura secuencial. Esto obedece a que para tales apli-

caciones, existe un límite real para el uso del paralelismo. Sin embargo, para la mayoría de las aplicaciones computacionales hoy en día, y aún considerando un límite para el paralelismo, la gama de posibilidades que las arquitecturas paralelas ofrecen en la solución de problemas de cómputo es enorme en comparación con los sistemas secuenciales tradicionales. ●

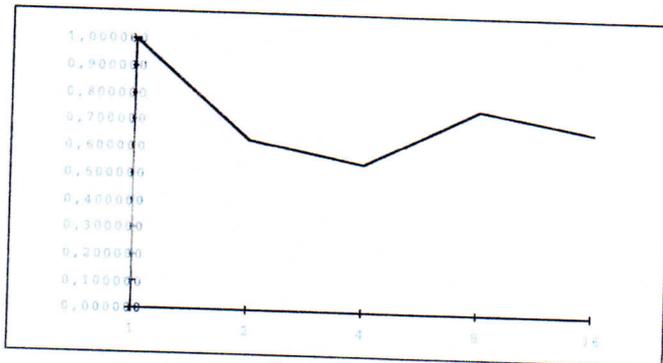


Figura 6. Eficiencia

Referencias:

[1] Bond, J., "Parallel-processing concepts finally come together in real systems", *Computer Design*, PennWell Publications, June 1, 1987.
 [2] H.P. Flatt, H.P., and Kennedy, K., "Performance of parallel processors", Elsevier Science Publishers

B.V.(North Holland), 1989.
 [3] Hoare, C.A.R., "Communicating Sequential Processes", *Communications of the ACM*, Vol. 21, No 8, August 1978.
 [4] INMOS Ltd., "Occam 2 Toolset User Manual - Part 1", (User guide and tools), March, 1991.
 [5] INMOS Ltd., "Occam 2 Reference Manual", Prentice-Hall International Series in Computer Science, 1988.

[6] Lewis, T.G., and El-Rewini, H., "Introduction to parallel computing" Prentice-Hall, 1992.
 [7] Monroy, R., y Mejía, M., "Paralelismo en Computación", *Soluciones Avanzadas*, No 5: Paralelismo Paradigmas, Septiembre-Octubre 1993.
 [8] Pountain, D., and May, D., "A tutorial introduction to Occam programming", BSP Professional Books, March 1991.
 [9] Romero, M., "Occam y Transputer: un lenguaje y una arquitectura para la programación y el procesamiento paralelo y distribuido", *Soluciones Avanzadas*, No 5: Paralelismo Paradigmas, Septiembre-Octubre 1993.



INTERNET DE MEXICO

Ofrece el servicio de acceso a la red mundial **Internet** tanto a empresas como a personas en lo individual.

¡ La comunicación mundial a su alcance !

- Correo electrónico
- Consulta a bancos de información y acceso a depósitos de software, con conexión total y sin restricciones a Internet (WWW, Mosaic, Gopher, Veronica, FTP, Archie, MUDs, Finger, WAIS, etc.)
- Tarifas y servicios atractivos para cualquier necesidad de empresa o persona
- Soporte técnico por la red
- Asesoría e instalación del software adecuado para lograr los mejores beneficios utilizando Internet
- Capacitación

Llámenos y uno de nuestros ejecutivos le explicará la manera de conectarse con el resto del mundo y aprovechar el cúmulo de información disponible en Internet.

Internet de México, SA de CV

Paseo de Echegaray No. 3 - 206, Naucalpan, Edo. de México 53310
 Tel: 360-2931 Fax: 373-1493 e-mail: info@mail.internet.com.mx