

EFICIENCIA DE LA PARALELIZACIÓN AUTOMÁTICA EN MULTIPROCESADORES UTILIZANDO MODELOS NUMÉRICOS DE CIRCULACIÓN DEL OCÉANO COSTERO

José Gómez-Valdés, Juan José Tapia y Hugo Sánchez Gijón
Centro de Investigación Científica y de Educación Superior de Ensenada,
Km 107, Carret. Tijuana-Ensenada, Ensenada, Baja California, México
E-mail: jgomez@cicese.mx

RESUMEN

Se estudia el rendimiento de la opción de paralelización automática de los multiprocesadores SGI Origin 2000 y UltraSPARC III utilizando dos modelos numéricos del océano, uno tridimensional y el otro bidimensional. Con dos procesadores, el modelo tridimensional alcanzó un incremento en la rapidez de procesamiento (IRP) de 1.2 en la SGI Origin 2000 y en la UltraSPARC III de 2.44. En ambos casos no hubo mejora en el IRP con la adición de más procesadores. El modelo bidimensional tuvo el mismo tiempo de ejecución en la SGI Origin 2000 para todo conjunto de procesadores accesados y alcanzó un IRP de 1.5 en la UltraSPARC III con dos o más procesadores. En todos los experimentos, se obtuvo un IRP más alto con el uso de la opción de optimización del compilador que con el uso de la opción de paralelización automática. En ambos modelos la parte paralela del código resultó menor que la parte no paralela; por ello, la opción de optimización funcionó más eficientemente que la opción de paralelización automática. Ambos modelos alcanzaron el más alto rendimiento usando la UltraSPARC III.

INTRODUCCIÓN

Ya que el usuario sólo necesita compilar y correr los modelos numéricos, la opción de paralelización automática es una elección atractiva en computación científica. Hay dos opciones para hacer procesamiento paralelo. Una de ellas es programación paralela y la otra paralelización automática. La primera consiste en diseñar códigos paralelos por el programador para luego implementarlos en computadoras paralelas, de tal forma que el usuario controla el grado de paralelización. Es decir, el rendimiento de la programación paralela depende del esfuerzo del programador y de la arquitectura de la computadora paralela (Gómez-Valdés y Wang, 1995; Gómez-Valdés y Soto-Amador, 1998). En cambio, paralelización automática, un caso particular de compilación, es una disyuntiva en la que el código fuente es secuencial y el código receptor producido por el compilador es paralelo. Un compilador para paralelizar programas analiza el código fuente con mucho más detalle que un compilador convencional. Si el compilador realiza una eficiente paralelización automática, el tiempo de ejecución del programa se reduce significativamente, señal de que se tomó ventaja del paralelismo de la aplicación. Como en un programa dado no se conoce *a priori* el rendimiento que se alcanzará con los compiladores de paralelización automática, el usuario necesita buscar la mejor opción que ofrece la computadora. Es decir, el rendimiento de la paralelización automática depende del código a paralelizar y del compilador. Este trabajo tiene como objetivo comparar el rendimiento de los paralelizadores automáticos de las computadoras SGI Origin 2000 y la UltraSPARC III, usando dos modelos del océano costero con diferente diseño numérico: el modelo de circulación general del océano costero de Wang (1993), de uso intensivo de operaciones de punto flotante, y un modelo lineal de mareas, de uso intensivo de memoria, para eva-

luar el estado actual de la opción de paralelización automática. También se compara el rendimiento de la opción de optimización de los compiladores de ambas computadoras.

MATERIALES Y MÉTODOS

El modelo de Wang (1993) es tridimensional; su marco teórico son las ecuaciones primitivas de la dinámica del océano. El modelo lineal de mareas es bidimensional; su marco teórico son las ecuaciones de marea de Laplace. Ambos modelos fueron discretizados con la técnica de diferencias finitas. El primero con un esquema implícito y el segundo con un esquema explícito (ver Haidvogel y Beckmann (1999) para una discusión sobre esquemas).

Los experimentos de análisis de rendimiento de los compiladores automáticos se llevaron a cabo en los multiprocesadores SGI Origin 2000, de los que se puede encontrar una descripción completa en: <http://telematica.cicese.mx/computo/super/cicese2000/hard2k.html> y UltraSPARC III, de los que se puede encontrar una descripción completa en: <http://telematica.cicese.mx/computo/super/calafia/caracteristicas.php> a continuación se dan los comandos que se utilizaron.

En la SGI Origin 2000 se usó el Power Fortran Analyzer versión 7.1, que es un preprocesador que asiste en la paralelización de lazos del programa. Primero se ejecutó el programa en un sólo procesador tanto para evaluar su tiempo de ejecución como para obtener los resultados del modelo. Se usan las instrucciones:

```
f77 -o modelo modelo.f
```

```
timex modelo
```

```
cp fort.21 salida
```

donde `modelo.f` es el nombre del modelo numérico escrito en Fortran, `timex` es el comando para medir el tiempo de ejecución, `cp` es el comando para hacer una copia de `fort.21`, un archivo que arroja el modelo, copia que se utilizará para comparar los resultados de las subsecuentes corridas. Con el comando `timex` se obtienen tres cantidades: el tiempo de ejecución, el tiempo del usuario y el tiempo del sistema. Aquí se usó la primera de estas cantidades en el experimento de análisis de rendimiento. En segundo lugar se aplicaron los comandos:

```
f77 -pfa -o model model.f
setenv MP_SET_NUMTHREADS 2
setenv MP_SCHEDTYPE simple
timex model
diff fort.21 salida
```

donde con `-pfa` se invoca el compilador de paralelización Fortran; en la segunda instrucción se accesa el número de procesadores, en este caso 2 de ellos; en la tercera se selecciona el método de calendarización, en este caso se usa `simple`; también se puede usar `interleave`, `dynamic` o `gss`. Una vez seleccionado el número de procesadores y el tipo de calendarización se ejecuta el modelo precedido del comando `timex`. Finalmente se compara la salida del modelo con dos procesadores y la salida del modelo con un procesador. En el tercer paso ya no se tiene que compilar el modelo, ni cambiar el método de calendarización, a menos que se quiera estudiar la eficiencia de las diferentes opciones, pero sí se tiene que ejecutar la segunda instrucción para seleccionar el número de procesadores, así como las últimas dos instrucciones.

En la UltraSPARC III se usó el mismo procedimiento que en la SGI Origin 2000. En el primer paso se siguen las mismas instrucciones en ambas computadoras. En el segundo paso se usaron las instrucciones:

```
f77 -autopar -o model model.f
setenv PARALLEL 2
timex model
diff fort.21 salida
```

En esta computadora, el método de calendarización se establece a nivel de código. Aquí el equivalente al método `simple` de la SGI Origin 2000 es el método `static`. `-autopar` es equivalente a `-pfa`, `PARALLEL` es equivalente a `MP_SET_NUMTHREADS`. Enseguida, para obtener el tiempo de ejecución con otro conjunto de procesadores, se sigue al procedimiento arriba descrito.

En las dos computadoras se puede optimizar el cómputo con la opción `-On`, en donde $1 < n \leq 3$, del compilador Fortran respectivo. En la SGI Origin 2000 se usó el comando

```
f77 -On -OPT:roundoff=n -o model model.f
```

para cada nivel de optimización, en donde se da la opción `-OPT:roundoff=0` para evitar errores de punto flotante en los niveles 0, 1, y 2, y `-OPT:roundoff=2` para el nivel 3. En la UltraSPARC III se usó el comando

```
f77 -On -fsimple=0 -o model model.f
```

donde `-fsimple=0` tiene los mismos efectos que `-OPT:roundoff=n` de la otra computadora.

Se calculó el incremento de la rapidez de procesamiento (IRP) de acuerdo a la definición de rendimiento de Hennesy and Patterson (1995)

$$IRP(n) = \frac{\text{Tiempo de ejecución sin la mejora}}{\text{Tiempo de ejecución con la mejora}},$$

en donde para el caso de la comparación entre el rendimiento del ejecutable paralelo y del ejecutable secuencial se sustituye el numerador por el tiempo de ejecución del secuencial y el denominador por el tiempo de ejecución del paralelo. En el caso de la comparación del código optimizado y el no optimizado, se sustituye el numerador por el tiempo de ejecución del código no optimizado y el denominador por el tiempo de ejecución del código optimizado.

RESULTADOS

Se hicieron experimentos de análisis de rendimiento del modelo de Wang (1993) y de un modelo lineal de mareas en dos computadoras, con la finalidad de encontrar la mejor opción de paralelización

La Tabla 1 muestra los tiempos de ejecución en segundos de los dos modelos en las dos computadoras con la opción de paralelización automática. El modelo de Wang (1993) se denota como `modelo_1` y el modelo lineal de mareas como el `modelo_2`. El `modelo_1` corrió casi cinco veces más rápido en la UltraSPARC que en la SGI Origin 2000 con el uso de un sólo procesador; el mismo comportamiento tuvo el `modelo_2` pero con un menor IRP. El `modelo_1` corrió casi diez veces más rápido en la UltraSPARC que en la SGI Origin 2000 con más de dos procesadores. El `modelo_2` corrió seis veces más rápido en la UltraSPARC que en la SGI Origin 2000 con más de dos procesadores. Con el `modelo_1` se obtuvo el mejor tiempo de ejecución con dos procesadores en ambas computadoras. Con el `modelo_2` el tiempo de ejecución no varió con el uso de más de dos procesadores.

La Figura 1 muestra el comportamiento de la IRP del `modelo_1` usando el compilador de paralelización automática. La computadora UltraSPARC III alcanzó más alto IRP que la Origin 2000 para todos los procesadores. La computadora UltraSPARC III alcanzó un IRP mayor a 2 con dos procesadores. Sin embargo, con más de tres procesadores la IRP es menor que el número de procesadores.

Tabla 1. Tiempo de ejecución con la opción de paralelización automática

Numero de procesadores	SGI Origin 2000		Ultra SPARC II	
	Modelo_1	Modelo_2	Modelo_1	Modelo_2
Tiempo de ejecución (seg.)		Tiempo de ejecución (seg.)		
1	2845	3605	581	926
2	2353	3607	238	601
3	2361	3609	240	600
4	2351	3607	239	601
5	2365	3604	242	601
6	2359	3610	247	600
7	2373	3605	253	600
8	2361	3600	281	600

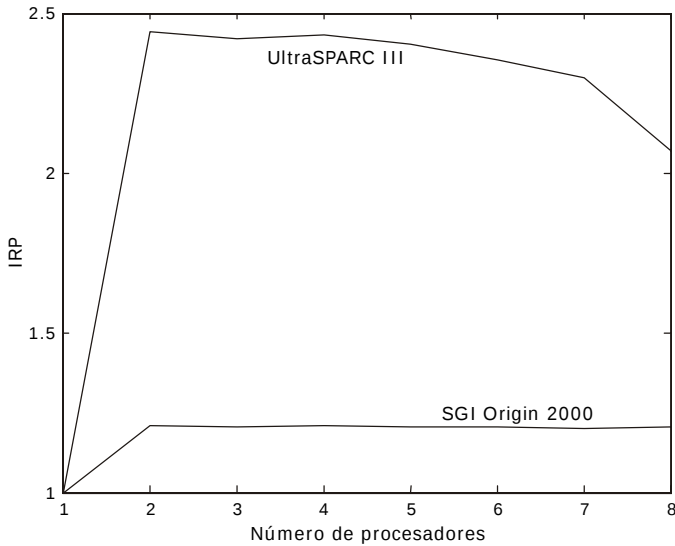


Figura 1. IRP vs número de procesadores para el modelo de Wang con la opción de paralelización automática.

La Figura 2 muestra el comportamiento de la IRP del modelo_2 usando el compilador de paralelización automática. La computadora UltraSPARC III alcanzó más alto IRP que la Origin 2000 para todos los procesadores. La computadora UltraSPARC III alcanzó un IRP de 1.54 con un número de procesadores mayor o igual a dos. En la computadora SGI Origin 2000 no hubo IRP, para todo conjunto de procesadores.

La Tabla 2 muestra los tiempos de ejecución en segundos de los dos modelos en las dos computadoras con la opción de optimización. El tiempo de ejecución mínimo se obtuvo con el nivel 3 de optimización en la UltraSPARC para los dos modelos. La Figura 3 muestra la IRP del modelo_1 y del modelo_2 para las dos computadoras. En la SGI Origin 2000 el IRP con el nivel dos fue mayor que la IRP con el nivel tres, debido a que se activó la opción `-OPT:roundoff=2`, si se quita esta opción, el IRP del nivel tres mejora, pero los resultados del modelo son diferentes.

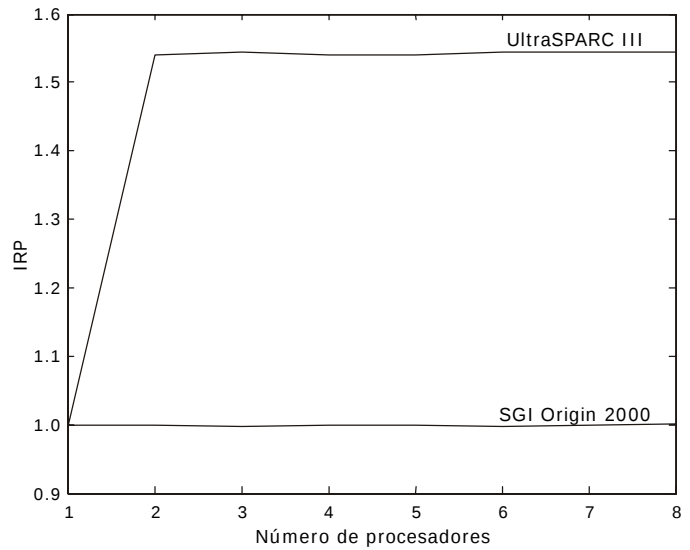


Figura 2. IRP vs número de procesadores para el modelo lineal de mareas con la opción de paralelización automática.

Tabla 2. Tiempo de ejecución con la opción de optimización

Nivel	SGI Origin 2000		Ultra SPARC II	
	Modelo_1	Modelo_2	Modelo_1	Modelo_2
Tiempo (seg.)		Tiempo (seg.)		
0	2845	3601	936	1333
1	2603	2748	587	988
2	1136	2415	342	661
3	977	2898	232	606

DISCUSIÓN Y CONCLUSIONES

Se corrieron dos modelos numéricos del océano costero en los multiprocesadores SGI Origin 2000 y UltraSPARC III con la opción de paralelización automática. En la búsqueda del más alto rendimiento para cada uno de los modelos y con la finalidad de comparar el rendimiento de las opciones que ofrece el compilador de cada computadora, también se usó la opción de optimización en la serie de experimentos. Para lograr que los resultados no fueran afectados por una carga excesiva en las computadoras, los experimentos se corrieron con un sólo usuario o con un número reducido de usuarios.

El par de modelos seleccionados representan dos tendencias muy populares en la modelación del océano costero. Por un lado, el modelo de Wang (1993) es tridimensional y el modelo lineal de mareas es bidimensional. Por otro lado, aunque los dos usan la técnica de diferencias finitas, el primero está estructurado en un esquema implícito y el segundo en un esquema explícito. De tal forma que es amplio el universo de aplicación de los resultados obtenidos en el presente trabajo, aunque el IRP puede ser afectado por otros factores, e.g. uso intensivo de los canales de entrada / salida.

La opción de paralelización automática fue más eficiente en la computadora UltraSPARC III, para ambos modelos. Si definimos rendimiento como el tiempo de ejecución entre el número de procesadores accedidos en cada experimento, los resultados indican

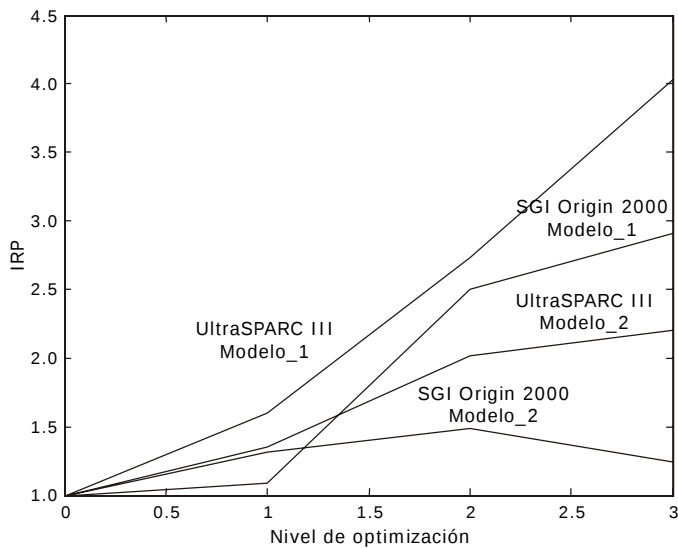


Figura 3. IRP vs nivel de optimización para los dos modelos y las dos computadoras.

que la mejor opción en ambos modelos es el uso de la UltraSPARC III con dos procesadores. Esto es un indicación de que el grado de paralelismo de los códigos no es alto.

La opción de optimización del compilador resultó más eficiente en la computadora UltraSPARC III, para ambos modelos. El tiempo de ejecución mínimo se alcanzó con el nivel de optimización 3. Como la parte paralela de los códigos resultó menor que la parte no paralela, la opción de optimización funcionó con más eficiencia que la opción de paralelización.

Con la opción de paralelización automática el compilador reorganiza los códigos para hacer un uso efectivo del paralelismo. Según Bauer (1992) el grado de eficacia de un paralelizador automático radica en la habilidad que tenga de romper con las estructuras de dependencia de datos. Algunas situaciones de dependencia de datos resultan no paralelizables por el compilador, e.g. recurrencia y reducción. Los modelos empleados en el presente trabajo resultaron no paralelizables y no escalables con la opción de paralelización automática. Con el uso de las herramientas de programación paralela se puede lograr un más alto rendimiento. Gómez-Valdés y Wang (1995) probaron que con el paradigma de paso de mensajes el modelo de Wang (1993) es escalable, es decir, la IRP se incrementa linealmente con el número de procesadores. Otra manera, relativamente más simple, es con el uso de un conjunto reducido de directivas del Power Fortran para paralelización manual disponibles en la UltraSPARC III.

AGRADECIMIENTOS

Los autores desean agradecer a Julián Javier Delgado Jiménez su ayuda en el proceso de correr los modelos en las computadoras SGI Origin 2000 y UltraSPARC III de CICESE y a un árbitro anónimo. Este trabajo fue financiado por CICESE.

REFERENCIAS

- Bauer, B.B., 1992. Practical Parallel Programming, Academic Press, San Diego, CA.
- Gómez-Valdés, J. and Wang D-P., 1995. Massively parallel processing in coastal ocean circulation model. *Journal of Scientific Computing*, 10, 305-323.
- Gómez-Valdés, J. y Soto-Amador, R., 1998. Programación paralela en estaciones de trabajo de un modelo numérico de un acuífero confinado resuelto por elemento finito. *GEOS*, Vol.18, No. 3, p. 206-212.
- Haidvogel, D.B., Beckmann, A., 1999. Numerical Ocean Circulation Model, Imperial College Press, London (320 pp).
- Wang, D-P., 1993. Model of frontogenesis: subduction and upwelling. *Journal of Marine Research* 51 (3), 497-513.