

Estudio del rendimiento de la técnica DPD utilizando estrategias en arquitecturas paralelas

E. Escalante^a, C. Echeverría^{a,b}, P. Guillen^c, S. Rojas^c y P.J. Colmenares^{a,d}

^aPostgrado de Física Fundamental, Facultad de Ciencias, Universidad de Los Andes, Mérida, Venezuela,

^bUniversidad Nacional Experimental del Táchira, San Cristobal, Venezuela,

^cCentro Nacional de Cálculo Científico, Universidad de los Andes,

^dDepartamento de Química, Facultad de Ciencias, Universidad de Los Andes, e-mail: colmenar@ula.ve.

Recibido el 24 de noviembre de 2003; aceptado el 12 de octubre de 2004

Hemos estudiado la implementación de un código paralelo que simula la dinámica de un fluido mediante la técnica denominada Dinámica Disipativa de Partículas. Específicamente calculamos la función de distribución radial que describe la probabilidad de encontrar un par de partículas separadas una distancia r . Analizamos la eficiencia del código paralelizado en una computadora de cuatro procesadores con memoria compartida, comparando los resultados con aquellos provenientes de Coveney y Novik [1].

Descriptores: Simulation; dissipative particle dynamics; stochastic processes.

We have studied the implementation of a parallel code that simulates the dynamics of a complex fluid in the framework of the dissipative particle dynamics technique (DPD). In particular, we calculate the radial distribution function that describes the likelihood of finding two particles separated a distance r in the fluid. We look for the efficiency of the parallelized code in a shared memory machine with four processors by taking as a reference the results from Coveney and Novik [1] in order to assess the effectiveness of our code.

Keywords: Simulation; dissipative particle dynamics; stochastic processes.

PACS: 02.50.Ey; 82.20.Wt; 83.10.Pp; 83.10.Rs

1. Introducción

Uno de los problemas que confronta la dinámica molecular clásica (DM) es el manejo y control del excesivo número de grados de libertad de las partículas componentes del sistema, así como, el de la correcta definición de los potenciales de interacción. Si a lo anterior se le añade el excesivo costo computacional que demanda esta técnica, entonces su rango de aplicación se restringe especialmente cuando aumenta la complejidad del sistema bajo estudio. En estos casos se disminuye el número de variables, tal que la descripción aún mantenga el carácter estadístico propio de la colección de partículas. Ésta es la esencia del método DPD. En él se agrupa a las partículas como un conglomerado de paquetes de moléculas, llamadas meso-partículas (MP), a las cuales, al igual que en DM les es impuesta una regla para la propagación de su momento y posición [2]. A diferencia del trabajo pionero de Einstein [3] sobre el movimiento browniano, la dinámica de la MP se realiza a través de una ecuación lineal de Langevin [4].

En este trabajo se simula un fluido 3D con la técnica DPD. Se investiga el rendimiento y eficacia de un código en FORTRAN77 (F77) y FORTRAN90 (F90) cuando éste es paralelizado de manera automática en una estación de cuatro procesadores con memoria compartida. A diferencia de trabajos previos [1], en el que se utilizan hasta 32 procesadores, observamos que no se requiere de una gran cantidad de ellos, pues el rendimiento del cálculo es afectado por el protocolo

de comunicación. Hemos encontrado que el número óptimo de procesadores es tres y discutimos las razones de este comportamiento, asunto no tratado por Coveney y Novik [1].

2. El método de simulación DPD

La fuerza total ejercida sobre la MP es dividida en una contribución conservativa (C), en una de carácter disipativo (D) y una fuerza estocástica (A) que compila el efecto aleatorio colisional del solvente [7]. Esta es dada por

$$\frac{d\vec{P}_i}{dt} = \sum_{j \neq i} [\vec{F}_{ij}^C + \vec{F}_{ij}^D + \vec{F}_{ij}^A], \quad (1)$$

$$\frac{d\vec{r}_i}{dt} = \frac{\vec{P}_i}{M}, \quad (2)$$

donde $\vec{P}_i$, $\vec{r}_i$ y M son el momento, posición y masa de la MP, respectivamente.

Tal como lo sugirieron Español y Warren [1] las formas funcionales de las fuerzas D y A en las distintas aplicaciones de este método son

$$\vec{F}_{ij}^D = -\gamma\omega_D(\vec{r}_{ij})(\hat{e}_{ij} \cdot \vec{v}_{ij})\hat{e}_{ij}, \quad (3)$$

$$\vec{F}_{ij}^A = \sigma\omega_A(\vec{r}_{ij})\zeta_{ij}\hat{e}_{ij}, \quad (4)$$

donde $\vec{v}_{ij} = \vec{v}_i - \vec{v}_j$ es la velocidad relativa de aproximación de las MP, $\hat{e}_{ij}$ es un vector unitario que apunta desde la partícula i a la j , ζ_{ij} es una función aleatoria gaussiana delta

correlacionada con promedio igual a cero, y $\omega_{(D,A)}(\vec{r}_{ij})$ es una función peso que depende de la distancia de separación del par; $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$. Las constantes γ y σ son escogidas de tal forma que reflejen la importancia relativa de las fuerzas identificadas. El efecto colisional es descrito por un proceso de Wiener dW_{ij} [5]. Así, la Ec. (1) se reescribe como:

$$d\vec{P}_i = \left[\vec{F}_{ij}^C(\vec{r}_{ij}) - \sum_{j \neq i} \gamma \omega_D(\vec{r}_{ij})(\hat{e}_{ij} \cdot \vec{v}_{ij})\hat{e}_{ij} \right] dt + \sum_{j \neq i} \sigma \omega_A(\vec{r}_{ij})\hat{e}_{ij} dW_{ij} \quad (5)$$

El balance entre las fuerzas disipativa y aleatoria conducen el sistema al equilibrio, que de acuerdo al teorema de disipación – fluctuación (TDF), [6,7], son dada por:

$$\omega_A(r) = \omega_D^{1/2}(r), \quad (6)$$

$$\sigma = (2kT\gamma)^{1/2}. \quad (7)$$

Este resultado acota los valores de las intensidades del ruido en función de la energía disipada en la fricción, así como la relación entre las constantes ω para estas dos contribuciones.

Inicialmente se asignan posiciones y velocidades iniciales aleatorias, y de acuerdo a una modificación del algoritmo de Verlet propuesto por Groot y Warren [8] se reproducen los resultados de Novik y Coveney [1] de la función de distribución radial (FDR).

3. Resultados y discusión

Los parámetros que caracterizan el fluido son descritos en Ref. 1. El código fue implementado en una plataforma SUN Ultra SPARC II con cuatro procesadores y compiladores Sun Whorkshop 6 update 2 para F77 y F90, respectivamente. Los resultados de los tiempos de ejecución (TE) secuencial en la simulación para los códigos F77 y F90 se muestran en la Fig. 1. Se observa que para un mismo número de partículas, F90 presenta un mayor rendimiento de ejecución al compararlo con el de F77. En la Tabla I se muestran los TE para las diferentes opciones de optimización, así como también el número de procesadores utilizados en cada ejecución. En esta tabla la eficiencia en paralelo es definida como la razón entre el TE en forma secuencial al producto del número de nodos y el tiempo de ejecución en paralelo. Puede observarse que al hacer uso de tres procesadores se obtiene una reducción del 62 % en el TE, cuando se compara con el del código secuencial. Es importante indicar que la verificación de los resultados se realizó comparando las FDR funciones, obtenidas después de cada ejecución. En la Tabla II se varía el número de partículas de la simulación y se mide el TE para F77 y F90 con opciones de compilación, especificando el número de procesadores para el multiprocesamiento. En el análisis de la data de la Tabla II se observa que aumentando el número de partículas de 2048 a 2916, la disminución del tiempo es proporcional a la cantidad de procesadores, pero al aumentar la cantidad a

TABLA I. Perfiles para las diferentes compilaciones en la máquina Sun Ultra SPARC II con 4 procesadores y compiladores Sun Whorkshop 6.2 para FORTRAN (77 y 90).

Código	Opc. de compilación	Variable de ambiente	Tiempo Efic. (seg)	en paralelo	Reducción en %
F77	-p	secuencial	5633.17	—	—
F90	-p	secuencial	5334.19	—	—
F90	-fast -p	secuencial	1978.23	1.00	—
F90	-autopar -fast -xtarget = ultra 2	PARALLEL 2	957.30	1.03	51.60
F90	—	PARALLEL 3	747.04	0.88	62.23
F90	—	PARALLEL 4	2331.18	0.21	117.84
F90	-autopar -fast -xarch=v9	PARALLEL 2	2130.65	0.75	33.74
F90	—	PARALLEL 3	3130.66	0.50	33.74
F90	—	PARALLEL 4	4130.60	0.37	33.74
F90	-autopar -O4 -dalign -xarch=v9	PARALLEL 2	2130.63	0.75	33.74
F90	—	PARALLEL 3	3130.66	0.50	33.74
F90	—	PARALLEL 4	4130.65	0.32	33.74

TABLA II. Tiempos de ejecución del programa para diferentes cantidades de partículas.

Partículas	F90 (seg)	P.A. 2 (seg)	P.A. 3 (seg)	P.A. 4 (seg)
2048	479.48	552.83	755.18	206.63
2916	985.10	929.34	699.33	425.10
4000	1842.27	958.41	750.53	797.28
5324	3259.73	943.75	723.33	1393.87
6912	5334.19	957.30	747.04	2331.18

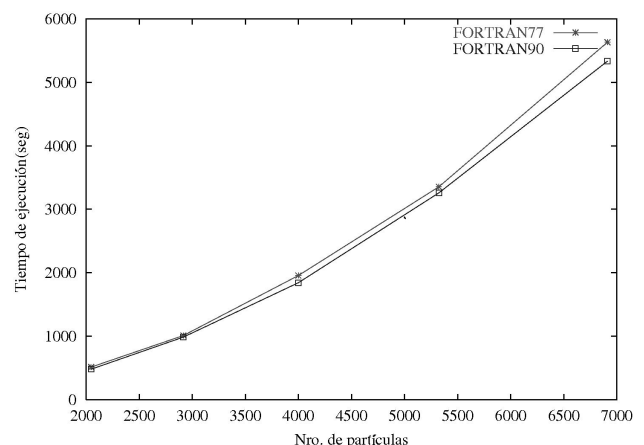


FIGURA 1. Comparando los códigos compilados en FORTRAN77 y FORTRAN90.

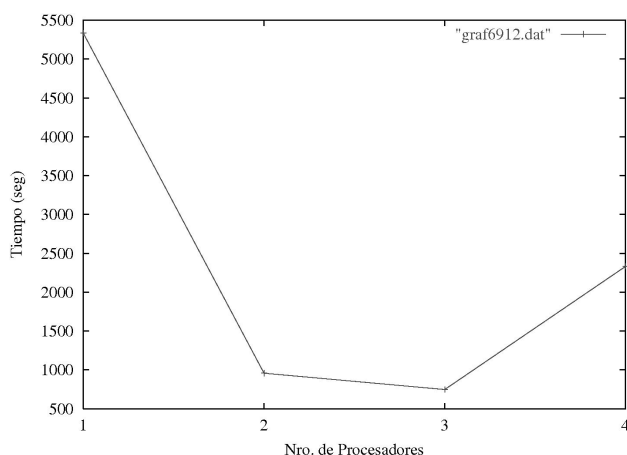


FIGURA 2. Rendimiento en función del número de procesadores, para 6912 partículas utilizadas en la simulación

4000, hasta el máximo reportado en la Ref. 1, esto es 6912 partículas, aumenta la cantidad de trabajo a realizar por el código y se acentúa una drástica disminución del tiempo para la configuración de tres procesadores, obteniéndose de esta manera una mayor eficiencia computacional mediante el paralelismo automático. Esta tabla permite graficar el tiempo de ejecución en cada una de las simulaciones en función del número de procesadores, conservando el número de partículas fijo, tal como lo describe la Fig. 2.

No es posible reportar el número máximo de hebras que pueden crearse exitosamente durante la ejecución del programa. Este número depende de la carga del sistema, la cantidad de memoria utilizada por el programa y la cantidad del espacio de memoria dependiente de la implementación disponible para cada hebra. Si el mecanismo dinámico para determinar este número está desactivado, como es nuestro caso, observa-

remos el comportamiento particular mostrado en la Fig. 2. Por lo tanto, el comportamiento del programa es dependiente de la implementación. Cuando más hebras son requeridas que puedan ejecutar exitosamente el programa, el tiempo del mismo se reduce, pero posteriormente aumenta debido a la granularidad del programa para ser paralelizado y las condiciones de comunicación entre los procesadores. Si el mecanismo dinámico para determinar el número de hebras está activado, el requerimiento de más hebras que una implementación puede soportar, se satisface mediante el número más pequeño y conveniente de ellas.

4. Conclusiones

Hemos reportado la simulación de un fluido infinito utilizando opciones de compilación y de paralelización automática. Es importante resaltar que los resultados para la FDR coinciden totalmente con los reportados en la Ref. 1 en plataforma de 16 y 32 procesadores. A pesar de que este trabajo se llevó a cabo en una estación de tan solo cuatro procesadores, hemos podido estudiar el comportamiento de la particularidad señalada en [1], es decir, la disminución del TE del programa trabajando con 6912 partículas. El estudio del comportamiento con esta cantidad de partículas nos indica que la eficiencia del código es aceptable con un número modesto de procesadores y no con un arreglo masivo, lo que propiciará en un futuro cercano obtener simulaciones más ambiciosas.

Agradecimientos

Los autores agradecen al FONACIT por el financiamiento a través de la subvención S1-2001000910.

1. P. Coveney y K. Novik, *Physical Review E* **61** (2000) 435.
2. P. Español y P. Warren, *Europhys. Letts.* **30** (1995) 191.
3. A. Einstein, *Investigations on the Theory of the Brownian Movement*, (Dover, New York, 1956).
4. J. Keizer, *Thermodynamics of Non Equilibrium Processes*, (Springer Verlag, New York, 1987).
5. C.W. Gardiner, *Handbook of Stochastic Processes for Physics,*

Chemistry and the Natural Sciences, (Springer Verlag, Berlin, 1985)

6. P. Español, *Phys. Rev. E*, **52** (1995) 1734.
7. McQuarrie D., *Statistical Mechanics*, (Harper Row, New York, 1978).
8. R. Groot y P. Warren, *Journ. Chem. Phys.* **107** (1997) 11.