REVISTA COLOMBIANA DE COMPUTACIÓN
Volumen 9, número 1
Págs. 113 - 125

Algoritmo paralelo para la detección y caracterización de halos de materia oscura en simulaciones cosmológicas

Carlos A. Vera* Jorge I. Zuluaga* Juan C. Muñoz*

Fecha de Recibido: 08/01/2008; Fecha de Aprobación: 15/03/2008

Resumen

Es un hecho bien conocido que la materia oscura representa la mayor parte del contenido material del universo y por ello las simulaciones cosmológicas que estudian la dinámica en volumenes considerables de espacio no necesitan incluir la componente bariónica. En lugar de ello se propaga un escenario de materia oscura en el que después se identifican estructuras (halos) y se determinan sus propiedades. Los bariones se introducen al final cuando se ha construido dicho marco material. En este trabajo presentamos un algoritmo que usando computo paralelo con paso de mensajes permite la identificación y caracterización de halos de materia oscura en simulaciones cosmológicas. Se presentan los resultados de una serie de pruebas del algoritmo usando una implementación con el estándar MPI. Las pruebas demuestran su ventaja potencial frente a soluciones secuenciales normalmente utilizadas en el área.

Palabras clave: *Programación paralela usando paso de mensajes. Simulaciones cosmlógicas, Modelo ΛCDM, Recetas semianalíticas.*

Abstract

It is a well known fact that dark matter constitutes the largest part of the matter content in the Universe. Cosmological simulations where the dynamics of matter in large volumes is followed do not require to take into account the effect of the barionic component. Inspite of that a dark matter "scenario" is propagated and after that bound dark matter halos and their properties are detected and computed. Barions are introduced at the end using semianalytical recipes. In this work we present a novel algorithm to identify halos and compute their properties in cosmological simulations using parallel computing with a message passing approach. The results of a series of test performed on the algorithm using an implementation in MPI are also presented. The results points out to a potential advantaje of the algorithm respect to sequential solution in terms of an improvement in the total proccessing time for a given simulation size[‡].

Keywords*: Parallel programming using the message passing model. Cosmlogical simulations, ΛCDM model, Semianalytical recipies.*

* Grupo de Física y Astrofísica Computacional, *FACom*. Instituto de Física, Universidad de Antioquia, A.A. 1226, Medellín, Colombia. {cavera,jzuluaga,jcuartas}@udea.edu.co

1 Introducción

La dinámica del universo a gran escala está dominada por gravitación y por lo tanto, una vez fijada su geometría, solo son las componentes materiales las que influyen en la evolución (Longair, 1998). A su vez, el modelo cosmológico estándar se fundamenta en la existencia de dos cantidades que dominan el contenido material, a saber, los bariones y la materia oscura. Resultados observacionales muestran que en proporción por cada unidad de masa de materia bariónica en el universo hay ~5 unidades de masa de materia oscura (Ω_b=0.042, Ω_m=0.238) (Spergel et al. 2003). Esto tiene implicaciones importantes en la forma en la que se entiende y describe la evolución del Universo.

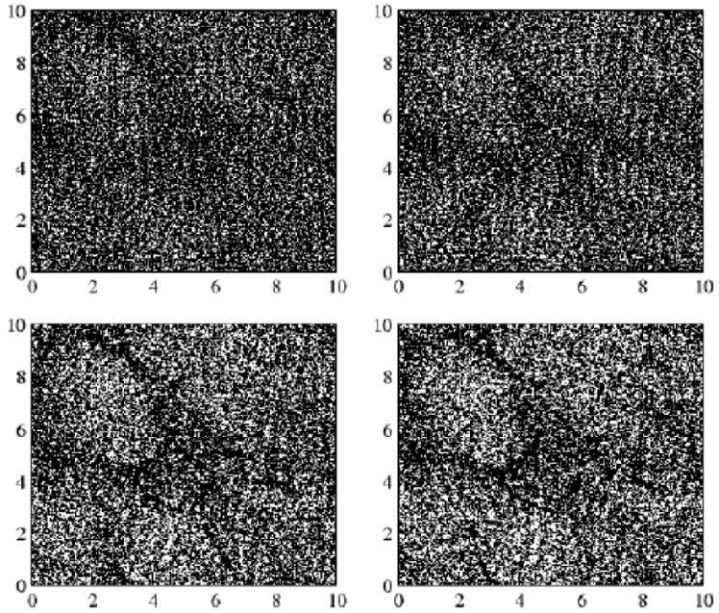

Figura 1: Snapshots de la simulación cosmológica utilizada para pruebas de este trabajo. La simulación tiene $N=10^5$ partículas. Los snapshots corresponden a redshifts z=2.47, 1.42, 0.68 y 0.0 (De izquierda a derecha y de arriba hacia abajo). Los parámetros cosmológicos de la simulaci\'on fueron $\Omega_\Lambda=0.75$, Ω_m =0.24, $h=0.73$ y $\sigma_8=0.77$.

Se puede pensar entonces que es la materia oscura la que principalmente gobierna la dinámica, mientras los bariones producen solo una pequeña perturbación en esa dinámica. Así pues, en una simulación cosmológica que estudie una gran porción del universo el método óptimo para describir el comportamiento de la materia no es propagando

numéricamente un enorme conjunto de partículas con las dos componentes materiales, lo cual es computacionalmente costoso aun para una pequeña fracción del mismo (Springel et al., 2005b). Por el contrario, una forma mucho más eficiente es integrar las ecuaciones de movimiento sólo para la componente no bariónica almacenando "periódicamente" información física relevante (posición y velocidad de cada partícula en la simulación) en "instantáneas de universo" (en lo sucesivo *snapshots*). Posteriormente se identifican en cada *snapshot* estructuras de materia oscura ligadas (en adelante *halos*) y en los pozos de potencial generados por estas configuraciones se introducen a posteriori y de manera "astuta" los bariones (Hatton et al., 2003).

Esta estrategia conduce a resultados coherentes con observaciones presentes del universo a gran escala (Hatton et al., 2003; Croton et al., 2005; Springel et al., 2006) y ofrece la oportunidad de estudiar volumenes prohibitivamente grandes dado que no es necesario introducir los efectos no gravitacionales que genera la presencia de materia bariónica sobre la dinámica (campos electromagnéticos, formación estelar, *feedback* por explosión de supernovas, etc.). Pero simular la dinámica de un número muy grande de particulas de materia oscura no es tampoco una tarea trivial. Una simulación razonable involucra del orden de 10^7 partículas distribuidas en un cubo de más de 100Mpc de lado. Propagar numéricamente este sistema desde un *redshift z* alto (una etapa temprana del universo) hasta un *redshift* pequeño (una época reciente) implica un tiempo de cómputo de 3000h-CPU en las más grandes facilidades de cómputo del mundo (Springel et al. 2005b).

Pero la construcción del "escenario" de materia oscura es apenas el primer paso de una serie compleja de tareas para reproducir computacionalmente la historia del universo. El siguiente paso requiere recorrer cada *snapshot* identificando posibles estructuras ligadas de partículas en las que pueda posteriormente introducirse materia bariónica para la formación de las galaxias. En una simulación con las características descritas en el parágrafo anterior el número total de estructuras formadas es del orden de 3×10^4 por snaphot y su detección es computacionalmente demandante. Se hace aún más exigente cuando se incluye también el cálculo de sus propiedades físicas (masa virial, momento angular, energía potencial,...).

En este trabajo se propone un algoritmo paralelo para el análisis de simulaciones cosmológicas y la detección en ellas de halos de materia oscura y el cálculo de sus propiedades. El algortimo utiliza la metodología de paso de mensajes para la comunicación entre procesos y ha sido implementado en *C* usando el estándar *MPI* para su puesta a punto y pruebas de procesamiento.

La organización del documento es la siguiente; en la sección 2 se hace una descripción de los algoritmos de fondo requeridos para la solución al problema y de las cantidades físicas que deben calcularse. En la

sección 3 se describen los detalles específicos del algoritmo y su implementación. En la sección 4 se presentan los resultados obtenidos para una serie de pruebas ejecutadas sobre el código. Finalmente en la sección 5 se presentan las conclusiones y perspectivas del trabajo.

2 El modelo físico

En una simulación cosmológica es clara entonces la necesidad de seguir la formación de estructuras no bariónicas a lo largo de la evolución del sistema; lo cual consta esencialmente de tres tareas, a saber, identificación de halos, determinación de sus propiedades y la construcción de una estructura jerárquica (árbol de fusión) donde se hacen explícitas las relaciones de parentesco entre los halos identificados en cada *snapshot* (Hatton et al., 2003). Este último paso permite a su vez obtener información de los procesos de fusión y desmembración de los halos de materia oscura a lo largo de la simulación. Nos concentraremos aquí en las tareas específicas de detección de halos y el cálculo de sus propiedades.

2.1 Identificación de Halos

Existen varios métodos para la identificación de halos de materia oscura que han sido ampliamente estudiados en la literatura (Hunchra et al., 1982; Davis et al., 1985; Bertshinger et al., 1991; Gelb et al., 1994; Weinberg et al., 1997). En el fondo se basan en la misma idea: la búsqueda de regiones del espacio donde la densidad media de partículas sea mayor al valor medio en un cierto factor (típicamente 200 veces mayor que la densidad crítica). Sin embargo el más sencillo y computacionalmente más barato de todos es el llamado *Friend-of-Friend* (en adelante FOF) (Davis et al., 1985). Este método utiliza el criterio de considerar dos partículas como pertenecientes a un mismo halo si están a una distancia menor que una cierta fracción, l, de la distancia media entre todas las partículas de la simulación, $\langle d \rangle$. Esa fracción representa el primer parámetro libre del método y se la conoce como la *linking lenght*.

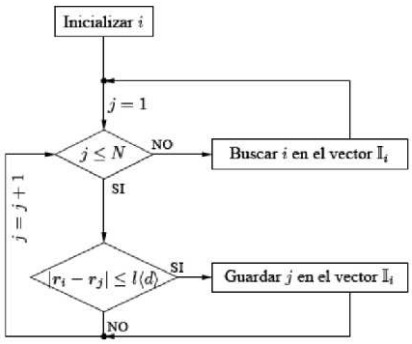

Figura 2: Esquema algorítmico del método FOF. Los índices latinos se utilizan para rotular las partículas *i,j=1,...,N*. Para cada partícula *i* se construye una estructura $\mathbb{I}_i$ donde se almacenan los rótulos de las partículas vinculadas a *i* por el método FOF.

Existe una probabilidad no nula de que por efectos de la dinámica un conjunto reducido de partículas se encuentre en una región común del espacio sin necesidad de que exista ningún vínculo definitivo entre ellas. Este efecto conduce a la identificación de un conjunto significativo de halos *no virializados* que generan un sesgo no físico en los estadísticos que caracterizan los mismos halos en la simulación. Para evitar este tipo de conteos no físicos se introduce entonces otro pa4ámetro libre $n_{\min}$ que se define así: si un halo identificado con FOF tiene un número de partículas *n* menor a $n_{\min}$ entonces es descartado.

En síntesis el proceso de detección de halos usando el algoritmo FOF implica la clasificación de las partículas en la simulación en halos determinados si cumplen el criterio de vecindad definido. De allí el número de halos que resultan puede ser muy grande e incluir halos artificiales que son excluidos si contienen un número de partículas menor que un cierto umbral. El algoritmo se representa diagramáticamente en la figura 2.

2.2 Cálculo de Propiedades

La introducción de la componente bariónica en cada uno de los halos identificados se hace de manera consistente con las propiedades físicas de estos últimos. Así por ejemplo dos halos de materia oscura con masas viriales diferentes hospedarán también diferentes cantidades de masa bariónica. Otros criterios se utilizan para asignar bariones y a su vez caracterizar sus propiedades, en el seno de halos de materia oscura de acuerdo a las propiedades físicas de estos últimos. Es necesario por tanto, antes de proceder con la "barionización" de los halos, calcular el conjunto de sus propiedades relevantes. Se consideran en este trabajo y como es regular en el contexto de este tipo de estudios (Hatton et al., 2003) el cálculo del siguiente conjunto de propiedades:

- Masa virial. Esta propiedad se calcula suponiendo que cada estructura identificada está virializada. La masa virial *M* es simplemente la suma de las masas de las partículas contenidas en el halo.Usando argumentos cosmológicos se puede encontrar la forma de la distribución que debe tener este estadístico. El modelo que permite hacer esto se conoce como el formalismo de *Press-Schechter* (Longair, 1998). Esta descripción permite entonces

verificar la validez de los resultados de una simulación numérica para la cual se puede calcular fácilmente la distribución estadística de la masa virial una vez se han identificado las estructuras de materia oscura (ver figura 4).

- La posición $\boldsymbol{R}$ y la velocidad $\boldsymbol{V}$ del centro de masa calculadas como el promedios pesado de la posición y velocidad de las partículas en el halo respectivamente.

- Las longitudes de los ejes principales de la distribución de masa a, b y c, que se consiguen diagonalizando el tensor de inercia $\boldsymbol{I}$ del sistema,

$$\mathrm{I}_{ij} = \sum_k m_k \left(r_k^2 \delta_{ij} - r_{ki} r_{kj} \right)$$

donde $i,j=x,y,z$ y $\boldsymbol{r}_k$ es la posición de la partícula del halo medida a partir del centro de masa.

- La densidad ρ de masa de la distribución

$$\rho = \frac{3M}{4\pi abc}$$

- La energía térmica del halo, calculada como la suma de los cuadrados de las velocidades relativas a la velocidad del centro de masa, multiplicada por la masa virial.
- La energía cinética T y potencial V del halo que se calculan sumando directamente las energías cinéticas individuales de las partículas y la contribución por pares a la energía potencial respectivamente.
- El parámetro de espín λ que se calcula a partir de la magnitud del momento angular $\boldsymbol{J}$ de la distribución (ver fig. 4)

$$\lambda = \frac{J\left|T+V\right|^{1/2}}{GM^{5/2}}$$

Cantidades tales como la energía potencial son normalmente costosas computacionalmente, $\mathcal{O}(n^2)$, y normalmente pueden sobrecargar de forma importante a los programas utilizados para estas tareas. En este trabajo se describe una forma eficiente para calcular todas estas cantidades basada en el esquema de detección mismo. Los detalles se discuten en la sección 3.

2.3 Árbol de Fusión

Cada halo detectado en un *snapshot* tiene un historial de parentesco con halos en otros *z*. Incluso en el mismo *snapshot* dos halos pueden provenir de la desintegración de un halo en el *snapshot* inmediatamente anterior lo que los vincula de manera directa. Dado que las propiedades de la componente bariónica se deben directamente a las características de los halos de materia oscura y su evolución es necesario entonces construir una estructura de datos que tenga en cuenta los parentescos de cada halo detectado en la simulación. A esta estructura se la conoce como *árbol de fusión* y se construye usando distintos criterios.
La construcción del árbol de fusión y las etapas posteriores del proceso no son el objeto de este trabajo y son tareas ampliamente discutidas en la literatura (Hatton et al., 2003).

El espacio entre secciones debe ser de 2 líneas en blanco. En caso de referencias bibliográficas, se debe seguir el formato del paquete Bibtex, como se presenta al final de este archivo, donde se referencia a [1] y [2]. Como usted puede ver, cada citación corresponde a un numero encerrado en paréntesis cuadrados.

3 El algoritmo

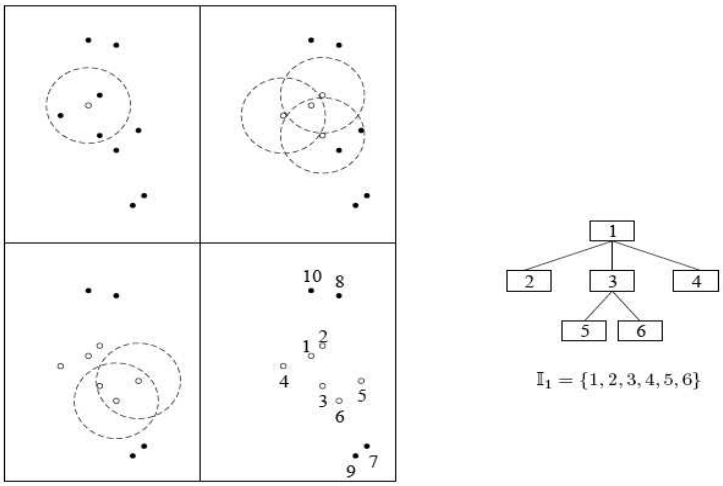

Figura 3: Panel izquierdo: en cada paso (de izquierda a derecha y de arriba hacia abajo) de la construcción se identifican las partículas dentro de la longitud de vinculación de la partícula nodo y se asignan como ramas de dicho nodo; las partículas incluidas en el árbol se simbolizan

por "○" y las que quedan por fuera de la longitud de vinculación se simbolizan por "●". Panel derecho: representación esquemática de la construcción del árbol.

La identificación básica de los halos en el marco de este trabajo se hace recurriendo a un algoritmo del tipo FOF explicado en la sección 2.1. Sin embargo y a diferencia de como se implementa normalmente el FOF, una característica novedosa de la aproximación utilizada aquí es la construcción de una estructura de árbol que relaciona las partículas que satisfacen los criterios definidos en el FOF. En esa estructura cada partícula en un halo es un nodo del árbol y las partículas que estén dentro de una esfera centrada en $\boldsymbol{r}_i$ con radio $<d>$ que no pertenecen ya al árbol se agregan como ramas de ese mismo nodo.

En la figura 3 se representa gráficamente el algoritmo descrito para la construcción de un árbol partiendo de una partícula semilla como nodo raíz. El producto de esta construcción es la lista $\mathbb{I}_i$ de las partículas pertenecientes al halo detectado usando como semilla la partícula *i*.
Este algoritmo tiene dos características fundamentales que contribuyen con el desarrollo de una estrategia de paralelización: primero se reconoce que para identificar los halos es necesario escoger partículas *semilla* que sirvan como nodos raíz para la construcción de los respectivos árboles. En segundo lugar la detección de un halo para una partícula semilla dada es independiente de la detección del halo asociado a otra partícula. Naturalmente puede darse el caso de que ambos halos coincidan, si ambos procesos son concurrentes.

Con base en las anteriores ideas se diseñó el siguiente algoritmo paralelo que dado los resultados de una simulación cosmológica de materia oscura entrega el conjunto de listas de partículas que pertenecen al mismo halo y las propiedades físicas relevantes de cada una de esas estructuras. El algoritmo usa el mecanismo de paso de mensajes para la comunicación entre los procesos.

Instancia raiz

- Empieza
- Haga (1)
    - o Genere *n* semillas que no estén incluidas dentro de ningún halo detectado hasta este punto.
    - •o Para $i=1,...,n$ haga (2)
        - Envíe las partículas a la instancia *i*
        - Envíe la semilla *i* a la instancia *i*
        - Reciba el árbol correspondiente al halo y sus propiedades de la instancia *i*
        - Si el halo recibido tiene un número de partículas menor que n_{min} descártelo
    - o Repita (2)
    - o Descarte los halos repetidos entre las *n* instancias ejecutadas

- Repita (1) mientras que no se haya agotado el número de partículas
- Calcule las propiedades de los halos seleccionados
- Termina

\item Empieza

Instancia *i*

- Reciba las partículas de la instancia raíz
- Reciba la semilla de la instancia raíz
- Construya el halo
- Envíe el halo a la instancia raíz

Esta estrategia de paralelización implica por tanto una descomposición tanto de dominio y funcional en los términos definidos en (Pressman 1988).

Para poner a prueba el anterior algoritmo se diseñó un paquete de módulos y programa en *ANSI C* usando la librería *MPICH* para el paso de mensajes. Se utilizaron también módulos de la *GNU* Scientific Library para la diagonalización del tensor de inercia. Para la manipulación de los datos se uso la librería *HDF5* muy popular en esta área.

4 Resultados

Para estudiar el comportamiento del algoritmo en una situación específica se utilizó una simulación cosmológica con las características descritas en la tabla 1. La simulación escogida es relativamente pequeña una elección que se hizo para simplificar algunos aspectos prácticos de las corridas de prueba (manejo de memoria, tiempo de ejecución de todas las pruebas en un recurso compartido, entre otras).

Tabla 1: Datos de la simulación usada para pruebas

N	6×10^4
Ω_Λ	0.75
Ω_b	0.24
Ω_m	0.04
σ_8	0.77

Se fijaron los valores de los parámetros libres del FOF en $l=0.2$ y $n_{\min}=32$ y se hicieron corridas de detección de halos y determinación de sus propiedades variando el número de instancias de ejecución por vez. Los resultados obtenidos para las propiedades física de los halos detectados se representan gráficamente en la figura 4. La distribución observada de las propiedades de los halos es consistente con los modelos físicos de evolución de materia oscura en un contexto cosmológico.

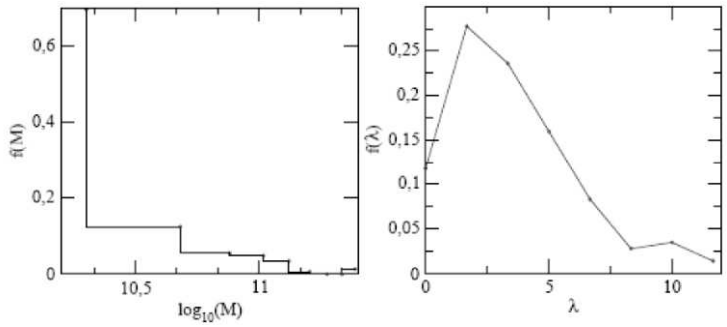

Figura 4: Distribución de masa (en unidades de 1M) y *spin* para una simulación con $N=6\times10^4$ partículas tomando en $l=0.2$ y $n_{min}=32$ en $z=0$.

Para estudiar el desempeño del algoritmo se midió el tiempo de ejecución (incluyendo el tiempo de inicio) variando el número de instancias en cada corrida. Los valores de esos tiempos como funci\'on del número de procesadores (número de instancias) se representan en la figura 5

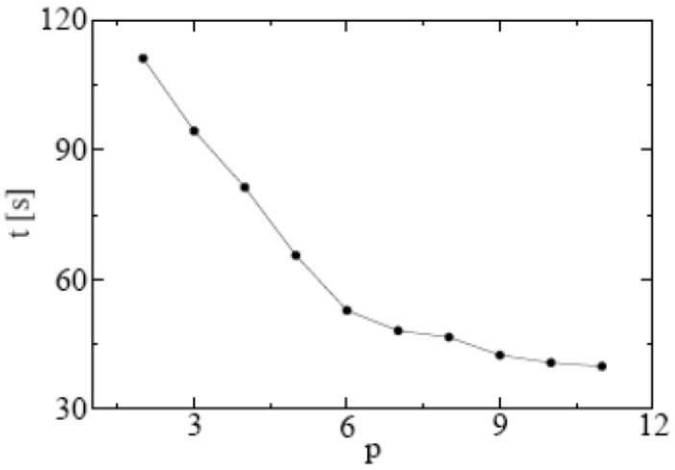

Figura 5: Tiempo utilizado t por el programa para la detección y el cálculo de las propiedades de los halos de materia oscura como función del número de procesadores p utilizados.

5 Conclusiones y discusión

En la figura 5 se puede apreciar el impacto importante que la paralelización tiene en la reducción del tiempo de detección y cálculo de las propiedades de los halos en la simulación; el tiempo de procesamiento se aproxima a un mínimo cuando $n=11$. A partir de este valor el tiempo de procesamiento no se reduce más. Este cambio se puede atribuir a distintos factores.

Entre los más importantes está el hecho de que al aumentar el número de nodos la probabilidad de detección de halos repetidos se incrementa y la eficiencia del algoritmo se deteriora considerablemente. Otros factores incluyen también las perdidas por comunicaciones (preparación de mensajes, latencia) que son importantes cuando el tiempo de procesamiento por instancia se reduce mucho como es el caso aquí por el reducido tamañoo de la simulación considerada.

Una de las más importantes ventajas ofrecidas por el algoritmo descrito en este trabajo es el uso del árbol para el cálculo de la energía potencial. Como se mencionó al principio esta es normalmente una tarea muy costosa computacionalmente ya que el número de operaciones crece muy rápido con el número de partículas $\sim n^2$. Recurriendo al árbol este tiempo se reduce significativamente si se identifica el nodo que contiene la partícula más cercana al centro de masas de la distribución y se suman las contribuciones al potencial descendiendo y ascendiendo por el árbol a partir de ese nodo. Se espera que si el halo tiene una alta simetríaa esta cantidad converja rápidamente y no se haga necesario considerar todas las parículas contenidas en él. En este último caso el tiempo de cálculo del potencial varía como $n \log n$ (Hernquist, 1988), mucho más pausadamente que en el caso de la suma directa.

Para comparar la calidad del algoritmo se hicieron algunas pruebas con herramientas secuenciales: una desarrollada y utilizada activamente por uno de los autores (J.C. Muñoz) y una del *N-body shop*. Para simulaciones de tamaño similares los tiempos de cálculo son superiores en un factor de 2-3 que los obtenidos con el algoritmo. Vale la pena mencionar que los programas usados para la comparación no hacen el cálculo de las propiedades de los halos, una tarea que de incluirse incrementaría aún más el tiempo de cálculo.

El algoritmo presentada aquí es por tanto una alternativa viable para reducir el tiempo que toma la ejecución de estas fases en los estudios de formación de galaxias en tiempos cosmológicos. Si bien esta no es la más complicada, ni la más larga de las tareas computacionales

involucradas en el contexto de estos estudios, la reducción de tiempo obtenida podría siempre facilitar que los recursos disponibles para esos trabajos se concentraran en la solución a los problemas más complicados.

El código desarrollado en el contexto de este trabajo esta siendo integrado a una plataforma desarrollada por el Grupo de Física y Astrofísica Computacional de la Universidad de Antioquia para la aplicación de recetas semianalíticas en el estudio de la formaciónn de galaxias en un contexto cosmológico.

6 Agradecimientos

Los autores agradecen al Instituto de Física de la Universidad de Antioquia por proveer tiempo de CPU en el cluster Hercules para la ejecución de las pruebas utilizadas en este trabajo. Se agradece también a Jaime Forero del Centre de Rechearche Astrophysique de Lyon por la productiva discusión y comentarios a las ideas contenidas en este trabajo

Referencias

[1] Bertschinger E. and Gelb J.M. 1991, Computers in Physics 5, 164.

[2] Croton D.J., Springel V., White S.D., et al., 2005.

[3] Davis M., Efstathiou G., Frenk C. S., White S. D. M., 1985, ApJ, 292, 371.

[4] Gelb J.M. and Bertschinger E. 1994, Astroph. J., 436, 467.

[5] Hatton S., Devriendt J.E.G., Ninin S., et al., 2003, MNRAS, 343, 75.

[6] Hernquist, L. 1988. Comp. Phys. Comm., 48, 107.

[7] Hunchra J.P. and Geller M.J., 1982, Astroph. J. 257, 423.

[8] Longair, M.S, Galaxy Formation, 1998, Astronomy and Astrophysics Library.

[9] Pressman R. 1988, Ingeniería de Software, MacGraw Hill.

[10] Spergel D.N., Verde L., Peiris H.V., et al., 2003, ApJS, 148, 175.

[11] Springel V., White S.D., Jenkins A., et al., 2005b, Nature, 435, 629.

[12] Springel V., Frenk C.S., White S.D., 2006, Nature, 440, 1137.

[13] Weinberg D.H. Hernquist L. and Katz, A. 1997, Astroph. J., 477, 8.