

Capítulo 6

Estudio comparativo de las estrategias de asignación en un entorno cluster

En este capítulo se muestra la efectividad de las políticas de mapping basadas en el modelo TTIG para un conjunto de aplicaciones desarrolladas en C+PVM y ejecutadas en un cluster de PCs. La efectividad de realizar la asignación de tareas en una plataforma distribuida teniendo en cuenta el grado de paralelismo se mostró anteriormente, tanto para aplicaciones con comportamiento arbitrario [RRS⁺00a], como para aplicaciones con comportamiento homogéneo [RRS⁺01].

La experimentación se efectúa con un conjunto de aplicaciones con paralelismo funcional. Por un lado se han utilizado aplicaciones sintéticas que reproducen un determinado patrón de comportamiento, y por otro lado se experimenta con una aplicación real de tratamiento de imágenes. Una vez comprobada en el capítulo anterior la efectividad de las políticas basadas en el modelo TTIG respecto a la asignación óptima, en el presente capítulo se estudia la bondad de las políticas TASC y MATE comparando con dos políticas de mapping basadas en los modelos clásicos TPG y TIG, puesto que el número de tareas de las aplicaciones no hace viable el cálculo de la asignación óptima.

La plataforma sobre la cual se ha desarrollado este proceso de experimentación es un cluster formado por PCs Linux, que incorporan el entorno

PVM 3.4. Cada máquina contiene un procesador Pentium II a 350 MHz con 128 Mbytes de memoria RAM. La red de interconexión es una Fast Ethernet a 100 Mbps. La ejecución de las aplicaciones se ha realizado teniendo el cluster dedicado. Esto nos permite analizar de forma aislada la influencia de unas asignaciones de tareas respecto de otras.

Para verificar el comportamiento de las estrategias de mapping desarrolladas al aumentar el número de procesadores, se hace además un estudio de escalabilidad mediante un entorno de simulación.

6.1 Descripción de las aplicaciones utilizadas

Para el proceso de experimentación en el cluster de PC's se han utilizado aplicaciones formadas por un conjunto de tareas en C+PVM. Dichas aplicaciones corresponden por un lado a aplicaciones sintéticas y por otro lado se ha utilizado una aplicación real de tratamiento de imágenes denominada BASIZ. En los siguientes apartados se describen las características que corresponden a cada uno de estos tipos de aplicación.

Además de los resultados que se aportan en el presente capítulo, en el Apéndice E se muestra, como un resultado añadido, la utilidad del modelo TTIG para la asignación de grafos de tareas basados en el modelo TPG. Para los algoritmos de scheduling de TPG que se desarrollan en dos fases, una primera fase de agrupación de tareas en clusters, y una segunda fase de asignación de clusters de tareas a procesadores, se analizan los beneficios del modelo TTIG para realizar la etapa de asignación del grafo de clusters.

6.1.1 Aplicaciones sintéticas

Para poder estudiar los efectos de las políticas de mapping en aplicaciones con un patrón de comportamiento previamente definido, se han desarrollado un conjunto de aplicaciones sintéticas con distintas características de topología de interacción entre tareas y de comportamiento.

Cada tarea está formada por un conjunto de fases de cómputo (implementadas con bucles de sumas de enteros), y de primitivas de comunicación

pvm_send y *pvm_recv* para comunicarse y transferir datos con las tareas adyacentes.

A partir de la descripción del Grafo de Flujo Temporal de cada tarea, que refleja la disposición de las fases de cómputo y las comunicaciones entre tareas, se generan automáticamente los programas C+PVM que reproducen el comportamiento de las mismas [Piq01]. El número de sumas a realizar en cada fase de cómputo y el tamaño de los mensajes se fijó inicialmente de acuerdo a los resultados obtenidos en la evaluación del sistema teniéndolo dedicado. En este caso, la suma de 100 millones de enteros corresponde aproximadamente a un segundo de ejecución. Por lo que respecta a las comunicaciones, el volumen especificado en el TFG se traduce directamente a Mbytes, puesto que se comprobó que la transferencia de 1 Mbyte tarda aproximadamente un segundo. De esta forma el volumen de comunicación se traduce directamente a segundos al construir el modelo TTIG para el mapping, realizando la misma simplificación para las comunicaciones que se comentó en el Apartado 4.1.2.

La clasificación de los grafos que corresponden a las aplicaciones sintéticas puede realizarse en base a las mismas características que se especifican en el Apartado 5.2. El tiempo de cómputo de las tareas y los volúmenes de comunicación entre tareas son uniformes, y lo que diferencia básicamente a las aplicaciones estudiadas es su topología de interacción de tareas y los grados de paralelismo. Se han creado por un lado un conjunto de aplicaciones con comportamiento arbitrario con grados de paralelismo variando entre 0 y 1. Las características de topología, grado de paralelismo medio y ratio entre cómputo y comunicación medio de las tareas del grafo TTIG se reflejan en la Tabla 6.1

Por otro lado se han generado un conjunto de aplicaciones con comportamiento homogéneo en las que los grados de paralelismo entre las tareas tienen un valor uniforme. Las características de los grafos TTIG de dichas aplicaciones para cada uno de los grados de paralelismo se detallan en la Tabla 6.2.

Tabla 6.1: Características de las aplicaciones sintéticas con comportamiento arbitrario.

Topología	Grafo	n. tareas	gr. medio	<i>rcc</i>
Irregular	AP1	16	0.62	4.76
	AP2	20	0.64	4.76
	AP3	25	0.69	4.76
	AP4	30	0.64	4.76
	AP5	32	0.64	5.42
Regular	pipe	32	0.43	4.83
	malla	36	0.51	4.6
	arbin	31	0.51	4.76

Tabla 6.2: Características de las aplicaciones sintéticas con comportamiento homogéneo.

		Grado de paralelismo			
Grafo	n. tareas	bajo	medio	alto	<i>rcc</i>
pipe	32	0.1	0.5	0.9	4.76
malla	36	0.1	0.5	0.9	4.76
arbin	31	0.1	0.5	0.9	4.76

6.1.2 Aplicación real

BASIZ (Brigth and Saturated Image Zones) es una aplicación con paralelismo funcional que lleva a cabo la detección de las zonas con más brillo e intensidad de color de una imagen de entrada. Esta aplicación se ha desarrollado a partir del soporte existente de tratamiento de imágenes modular conocido como URT (Utah Raster Toolkit) de la Universidad de Utah (<http://www.utah.com>), a partir del cual se ha paralelizado la aplicación para el entorno PVM [Bor01] [RRBL01].

Un uso importante de BASIZ en el entorno de visión por computador consiste en la identificación de las zonas de una imagen más sensitivas al ojo humano. Esto facilita en pasos posteriores un tratamiento más exhaustivo de estas zonas descartando las restantes. El proceso básico de identificación de las zonas más sensitivas se basa en aplicar un difuminado a la imagen original. Esto genera una imagen difuminada en la que las zonas más perceptibles al

ojo humano son las más brillantes, mientras que las más oscuras constituyen el fondo de la imagen. En la Figura 6.1 se muestra un ejemplo de imagen original, la que se obtiene después del proceso de difuminado y la imagen final con las zonas detectadas.

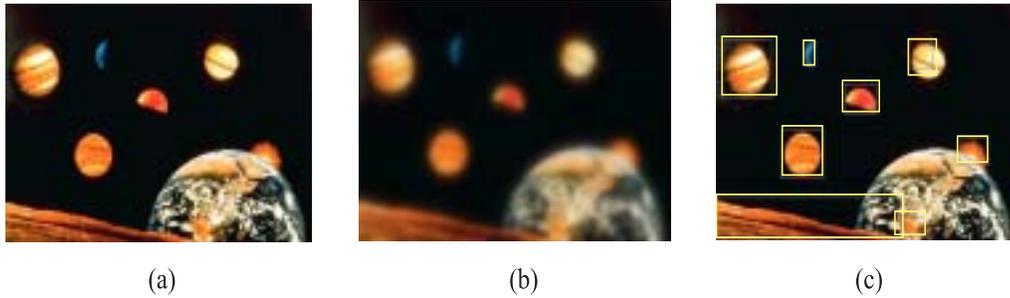


Figura 6.1: (a) Imagen original. (b) Imagen difuminada. (c) Zonas detectadas.

La aplicación BASIZ se compone de 26 tareas (de T0 a T25) implementadas en C+PVM, que se organizan en una estructura en pipeline de siete etapas. Las funciones que se realizan en cada etapa y las tareas que componen cada una son las siguientes:

1. *Separación de colores* (tareas T0,...,T5). Se separan los tres colores básicos rojo, verde y azul, de la imagen original y se generan tres imágenes, una con cada canal de color.
2. *Difuminado* (tareas T6,...,T14). Para cada canal de color se aplica un proceso de tres niveles distintos de difuminado, mediante la multiplicación matricial de la imagen original por una matriz gaussiana de 5×5 , aplicándola el número de veces especificado en el nivel de difuminado.
3. *Suma* (tareas T15, T17 y T19). Para cada canal de color se suman las tres imágenes difuminadas.
4. *Unión de color* (tareas T16, T18 y T20). Los tres canales de color difuminados se mezclan formando una única imagen. Este sería el resultado que corresponde a la imagen de la Figura 6.1(b).

5. *Conversión* (tarea T21). Cambio de formato de representación de la imagen de formato de colores RGB (Red, Green, Blue) rojo, verde y azul a formato HSI (Hue, Saturation, Intensity) de matiz, saturación y brillo.
6. *Umbral* (tareas T22 y T23). En función de los valores de matiz, saturación y brillo de la imagen global, se determina el valor umbral para detectar las zonas más sensitivas de la imagen. El resultado de esta etapa es una imagen binaria donde los pixels se codifican con los valores 255 o 0 según pertenezcan o no a una zona sensitiva respectivamente.
7. *Detección* (tareas T24 y T25). Utilizando la imagen binaria creada en el paso anterior, se detectan las zonas más sensitivas y se enmarcan en la imagen original, tal como muestra la imagen de la Figura 6.1(c).

Puesto que en este caso se dispone del código fuente de la aplicación, la obtención del comportamiento temporal de la aplicación BASIZ se ha llevado a cabo mediante la técnica de instrumentación del código. En la Figura 6.2 se muestra el grafo TFG que corresponde al comportamiento de esta aplicación. Este puede considerarse que es el comportamiento estable de la aplicación puesto que el tiempo de ejecución no cambia, independientemente del contenido de la imagen de entrada

A partir del grafo TFG se ha obtenido el grafo TTIG de la Figura 6.3, en el cual pueden observarse las distintas situaciones de paralelismo que se dan entre las tareas de la aplicación.

6.2 Descripción de las políticas de mapping

Para realizar la asignación de las tareas a los procesadores del cluster en las aplicaciones descritas, se han utilizado las siguientes estrategias de mapping basadas en el modelo TTIG y los modelos clásicos TIG y TPG.

1. *TTIG mapping*. Se han utilizado los dos algoritmos de mapping TASC y MATE desarrollados para el modelo TTIG, cuyo funcionamiento se detalla en el Capítulo 4 del presente trabajo.

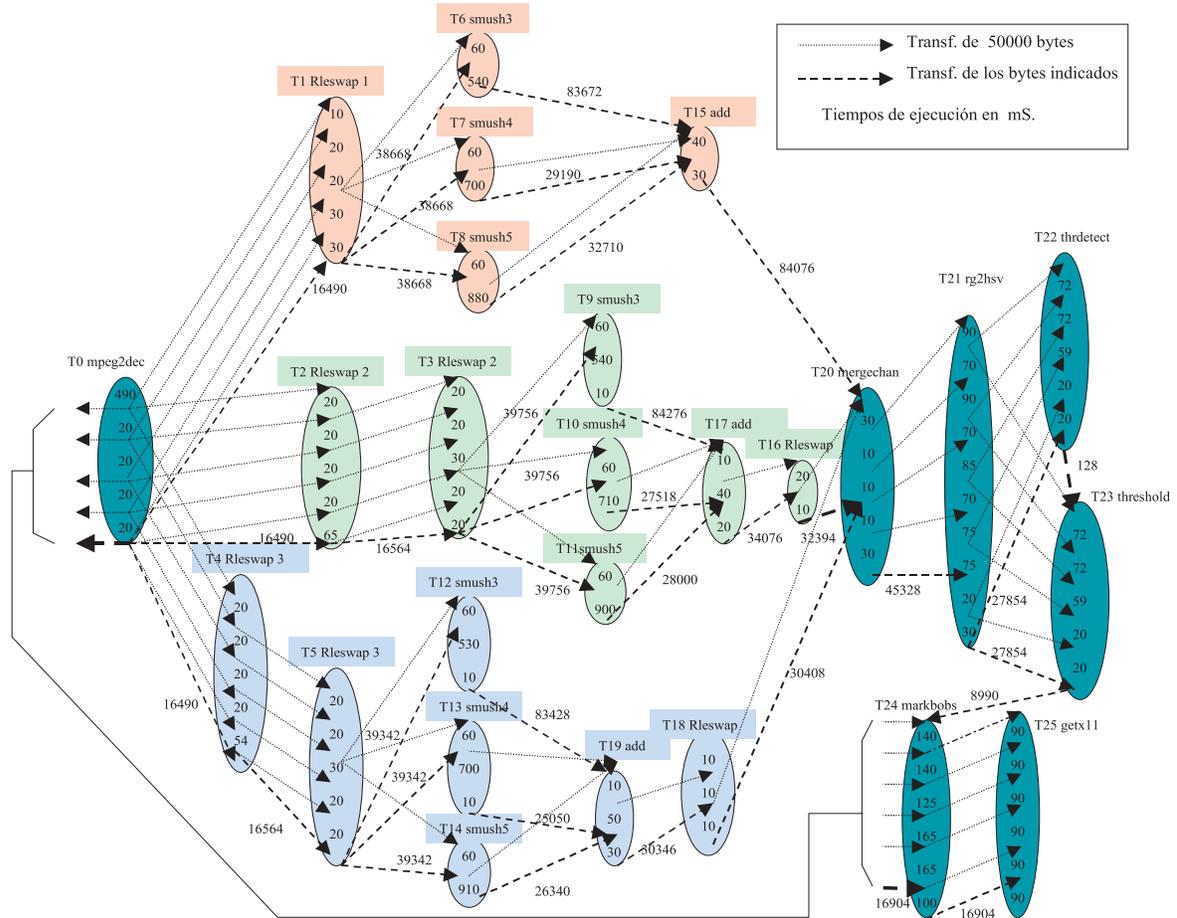


Figura 6.2: Grafo TFG de la aplicación BASIZ.