

## **Capítulo 7: CONCLUSIONES Y LÍNEAS ABIERTAS**

*En este capítulo, se resume el contenido de este trabajo, haciendo hincapié en las aportaciones realizadas, y se presentan las líneas abiertas y futuros trabajos a realizar.*

## **7.1 CONCLUSIONES**

Este trabajo se enmarca en el estudio de técnicas de paralelización automática de programas secuenciales. El trabajo se centra en la paralelización de bucles de un sólo nivel (el bucle más interno en caso de bucles anidados) con recurrencias que dificultan su paralelización utilizando técnicas clásicas. Las técnicas aquí presentadas permiten paralelizar y vectorizar estas recurrencias. Se obtiene además una medida de la eficiencia del proceso de paralelización con respecto a las características del bucle.

La primera contribución a destacar es la evaluación del paralelismo y longitud vectorial de las sentencias de un bucle. Como características del bucle permiten valorar a priori el interés de esforzarse en su paralelización. Por otro lado sirven de referencia absoluta para comparar la eficiencia o calidad de las distintas técnicas de paralelización.

Se ha estudiado el paralelismo de bucles sin y con sentencias condicionales. En el primer caso se obtiene directamente este valor a partir del grafo de dependencias del bucle. En el segundo caso, el paralelismo depende de la secuencia concreta de ejecución de las posibles ramas condicionales. Hemos presentado cotas entre las que se encontrará el paralelismo, sea cual sea la secuencia de ejecución. Para una estimación más fina de éste valor se ha evaluado la variación del paralelismo en función de la probabilidad de ejecución de las distintas ramas, considerando un modelo probabilístico en el que dicha probabilidad es independiente de los cálculos realizados en el bucle.

Como segunda contribución importante se ha presentado un método que hemos denominado GTS y que permite la generación de código paralelo alcanzando el paralelismo máximo disponible. Este método realiza una asignación estática de operaciones a tareas paralelas mediante recorridos en el grafo de dependencias, de forma que la ejecución secuencial garantice el mayor número posible de dependencias. Las tareas resultantes son independientes si el grafo tiene una recurrencia hamiltoniana. Si tiene más recurrencias, GTS introduce las sincronizaciones necesarias entre tareas.

Se ha presentado la condición que debe cumplir una dependencia para no necesitar su sincronización explícita por el hecho de quedar cubierta por otras.

También se ha descrito cómo añadir falsas dependencias al grafo para poder aplicar GTS en grafos no hamiltonianos o para mejorar la eficiencia de la paralelización o reducir el número de sincronizaciones necesarias.

EL código de las distintas tareas puede escribirse como un único DOACRQSS, si bien es necesario separar algunas iteraciones en un prólogo y epílogo. El tamaño del código generado aumenta con respecto al del programa original pero de forma comparable al obtenido por otras técnicas de paralelización.

El tercer gran tema considerado ha sido la generación<sup>de código</sup> vectorial\*. Dada la menor flexibilidad de este, tipo de máquinas resulta complicado generar código que intente aprovechar toda la longitud vectorial de las distintas sentencias. Puede además ser contraproducente por los prólogos y epílogos internos que se necesitan. Se han descrito dos técnicas de generación de código que no necesitan estos prólogos y epílogos. De ellas la consistente en forzar a que todas las longitudes vectoriales sean múltiplo de la menor permite alcanzar tiempos de ejecución muy próximos al óptimo teórico.

Finalmente se ha usado la medida de eficiencia presentada para evaluar las técnicas descritas en la bibliografía y compararlas con las aquí propuestas. Se observa cómo el paralelismo de recurrencias relativamente cerradas puede ser importante, GTS se muestra siempre superior a las otras técnicas teniendo además la gran flexibilidad que ofrecen las técnicas descritas de introducción de arcos adicionales.

Una última consideración de interés es la comparación entre máquinas MIMD y SIMD habiendo cuantificado, aunque sea de forma bastante genérica, la diferencia de eficiencia que permite obtener la mayor flexibilidad en las MIMD.

## **7.2 LÍNEAS ABIERTAS**

Este trabajo deja un gran número de líneas abiertas que merecen un esfuerzo de investigación. De entre ellas cabe destacar:

- (a) Desarrollo de heurísticas que permitan obtener de forma rápida las recurrencias de distribución y grafos transformados óptimos para los distintos criterios;
- (b) Vectonzación de bucles condicionales;
- (c) Extensión de GTS a la paralelización para computadores VLIW y sistemas multiprocesadores con memoria distribuida;
- (d) Aplicación conjunta de GTS para paralelización y vectorización en multiprocesadores vectoriales;
- (e) Extensión a bucles anidados, incluyendo la evaluación del paralelismo para este tipo de bucles y la aplicación de GTS a los mismos;
- (f) Desarrollo de técnicas de paralelización para programas no numéricos.
- (g) Realización de medidas sobre programas reales;