2.0 Fundamentos de diseño de algoritmos paralelos

Computación de Altas Prestaciones

Grado en Ingeniería de Computadores

Raúl Durán Díaz

Curso académico 2018–2019

Índice

Computación paralela y secuencial

- Dimensión adicional de la concurrencia.
- No se trata solo de traducir a lenguaje tradicional los pasos del algoritmo.
- Los extractores automáticos de paralelismo no funcionan demasiado bien.

Pasos del desarrollo paralelo

1. Descomposición.
2. Asignación.
3. Interacción.
4. Mapeo.

1. Descomposición

Conceptos: tarea

Definición 1. Una **tarea** es una porción del trabajo computacional según el juicio del programador. Una tarea posee instrucciones o *código* y *datos* sobre los que el código opera.

Comentario 2. Las tareas pueden ser

- de iguales o diferentes tamaños,
- dependientes unas de otras o independientes,
- normalmente indivisibles.

Conceptos: tarea

- Cada tarea se ejecuta en serie.
- La clave del aumento de rendimiento es conseguir que varias tareas se ejecuten simultáneamente.

Multiplicación matriz-vector

Ejemplo 3. Tenemos una matriz, A, de tamaño $n \times n$ y la multiplicamos por un vector b, de tamaño n, para generar el vector y, es decir, $y = A \times b$.

Sabemos que $y_i = \sum_{j=1}^{n} A_{ij} \cdot b_j$. Podemos considerar que el cómputo de cada y_i es una tarea. Tenemos, entonces, n tareas independientes y de tamaño similar.

Comentario 4. Cada tarea también podría calcular varios elementos de y, pero naturalmente lo haría en serie, uno detrás de otro.

Grafo de dependencia de tareas

- Puede haber tareas que no pueden comenzar mientras no finalicen otras.
- Para representar esas dependencias, utilizamos el *grafo de dependencia de tareas*, un grafo acíclico, dirigido, en donde cada nodo es una tarea y los arcos representan las dependencias.
- Cuando existen estas dependencias, se genera una *serialización* en la ejecución del programa.

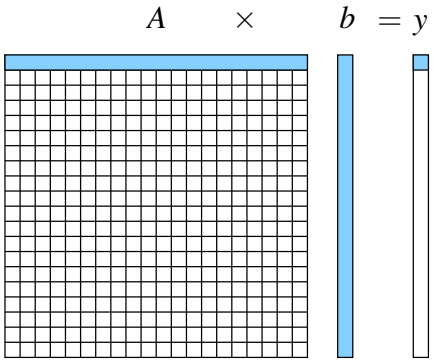

Figura 1: Generación del primer componente de y

Multiplicación matriz-vector

Multiplicación matriz-vector

- Podríamos también calcular cada producto $A_{ij} \times b_j$ como una tarea (para cierto i fijado).
- La componente y_i sería la suma de los n productos computados en tareas independientes.

Grafo de dependencia de tareas

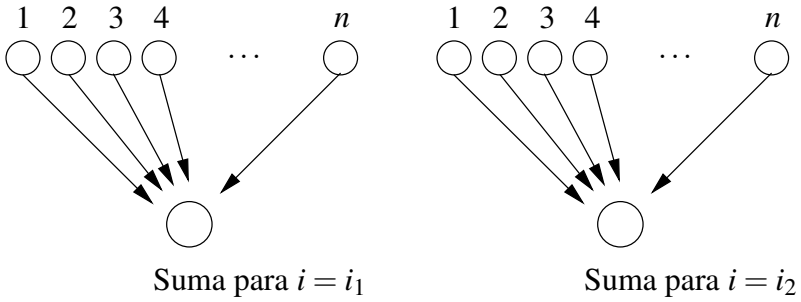

Figura 2: Cada producto $A_{ij} \times b_j$ es independiente

Multiplicación matriz dispersa-vector

- Cuando la mayor parte de las entradas de una matriz son cero, se llama *matriz dispersa*.
- Supongamos que repartimos la computación por filas: cada tarea computa un valor de

$$y_i = \sum_{j=1}^{n} A_{ij} \cdot b_j.$$

- Observemos que la tarea anterior se reduce bastante al haber un gran número de ceros en cada fila.
- Cada tarea i también es la «dueña» de una fila de la matriz A (es decir, es dueña de A_{i*}) y del elemento b_i del vector b.

Multiplicación matriz dispersa-vector

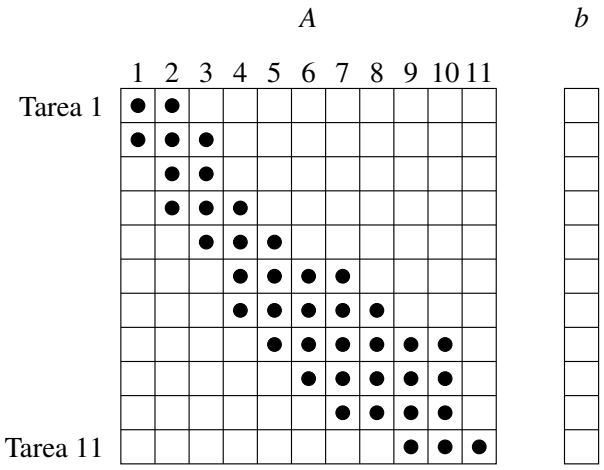

Figura 3: La tarea i computa $\sum_{1 \leq j \leq 11, A_{ij} \neq 0} A_{ij} \times b_j$

Granularidad

Definición 5. El número y tamaño de las tareas determinan la **granularidad**.

- Muchas tareas, tamaño pequeño ⇒ granularidad fina.
- Pocas tareas, tamaño grande ⇒ granularidad gruesa.

Grado de concurrencia

Definición 6. El **grado (máximo) de concurrencia** es el número (máximo) de tareas que se pueden ejecutar simultáneamente.

Comentario 7. En un grafo de dependencia de tipo arborescente, es igual al número de hojas en el árbol.

► Análogamente se define el *grado de concurrencia promedio*.

Camino crítico

Definición 8. El **camino crítico** es la longitud más larga entre un nodo-inicio y un nodo-fin cualesquiera.

Comentario 9. La relación entre la computación total y la longitud del camino crítico viene a ser, *grosso modo*, el grado de concurrencia.

2. Asignación

Conceptos: proceso

Definición 10. Un **proceso** es un agente abstracto capaz de ejecutar una tarea. Para ello utiliza el código y los datos que pertenecen a esa tarea y realiza la computación en un tiempo finito.

Comentario 11.
- Además de la ejecución de una tarea, un proceso puede también entrar en *interacción* con otros procesos, para *comunicar* o *sincronizar*.
- Para obtener ganancia con respecto a una ejecución puramente secuencial, es imprescindible que varios procesos estén en ejecución simultáneamente.

Conceptos: asignación

Definición 12. La **asignación** es el mecanismo por el cual las tareas se asignan a procesos para que puedan ejecutarse.

Conceptos: asignación

Comentario 13. Una buena asignación debería:

- maximizar la concurrencia, asignando tareas independientes sobre distintos procesos;
- minimizar el tiempo de ejecución, disponiendo de procesos libres para ejecutar las tareas del camino crítico en cuanto estas estén disponibles;
- minimizar la interacción entre procesos, asignando al mismo proceso las tareas más relacionadas.

► ¡Estos objetivos suelen ser antagónicos!

Tareas y asignaciones

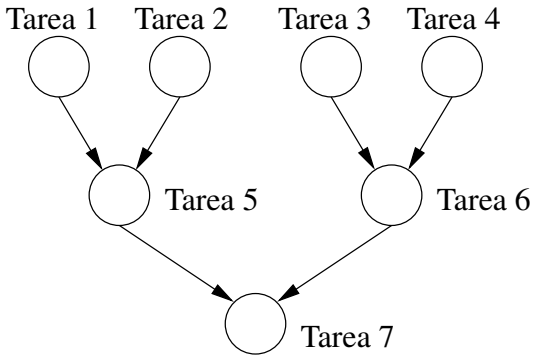

Figura 4: Tareas, procesos, dependencias, interacciones

Tareas y asignaciones

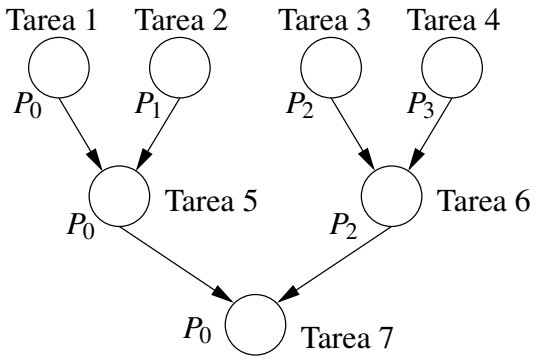

Figura 5: Tareas, procesos, dependencias, interacciones

Tareas y asignaciones

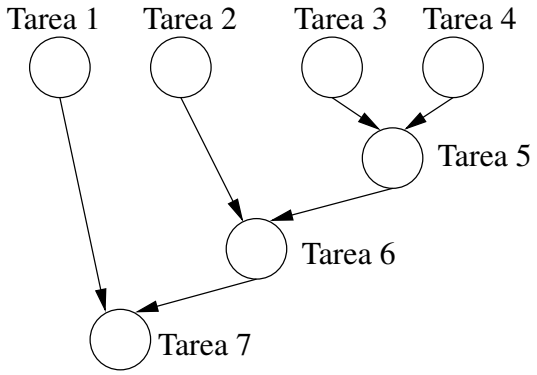

Figura 6: Tareas, procesos, dependencias, interacciones

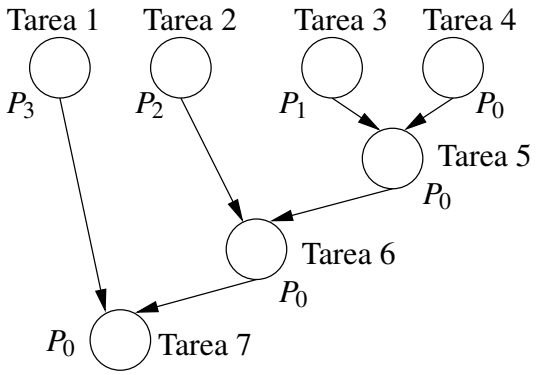

Figura 7: Tareas, procesos, dependencias, interacciones

Tareas y asignaciones

3. Interacción

Conceptos: interacción

Definición 14. La **interacción** es cualquier relación que se necesita establecer entre dos tareas para intercomunicar datos o sincronizar acciones.

Comentario 15. Puesto que las tareas con frecuencia comparten datos, se pueden llegar a establecer relaciones que no estaban inicialmente previstas en el grafo de dependencias. Las interacciones pueden también capturarse en forma de *grafo de interacciones entre tareas*.

Conceptos: interacción

Ejemplo 16. Observemos que, en el caso de la multiplicación matriz-vector, el vector b ha de ser accesible a todas las tareas, lo que puede implicar un intercambio (distribución) inicial, si el vector residía en la memoria de solo uno de los procesos.

Multiplicación matriz dispersa-vector: grafo de interacción

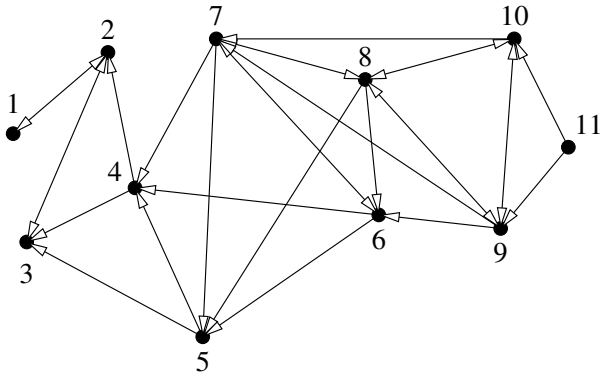

Figura 8: La flecha de i a j indica que la tarea i necesita un dato de j

4. Mapeo

Conceptos: procesador, mapeo

Definición 17. Un **procesador** es un elemento físico que permite la ejecución del trabajo computacional.

Definición 18. Entendemos por *mapeo* la asociación de la entidad proceso con un elemento de proceso físico concreto (podría tratarse de un *core* o núcleo).

Procesos versus procesadores

Comentario 19.

- Separar procesos de procesadores nos permite un nivel de abstracción superior, independizándonos del hardware y del sistema operativo.
- En general, podemos suponer una correspondencia uno a uno entre procesos y procesadores.
- Típicamente, será tarea del planificador del sistema operativo (con la posible colaboración de las bibliotecas de programación paralela) mapear los procesos a procesadores.