Universisdad de Los Andes
Facultad de Ingeniería
Escuela de Sistemas

Diseño de Algoritmos Paralelos

Prof. Gilberto Díaz
gilberto@ula.ve

Departamento de Computación, Escuela de Sistemas, Facultad de Ingeniería
Universidad de Los Andes, Mérida 5101 Venezuela

Diseño de Algoritmos Paralelos

Antes de mostrar los detalles de MPI debemos realizar algunas consideraciones sobre el diseño de algoritmos paralelos.

Diseñar algoritmos paralelos no es tarea fácil y es un proceso altamente creativo.

Inicialmente se deben explorar los aspectos independientes de la máquina

Los aspectos específicos a la máquina deben ser dejados para más tarde.

Diseño de Algoritmos Paralelos

El diseño involucra cuatro etapas las cuales se presentan como secuenciales pero que en la práctica no lo son.

- Particionamiento
- Comunicación
- Agrupamiento
- Asignación

Particionamiento: El cómputo y los datos sobre los cuales se opera se descomponen en tareas. Se ignoran aspectos como el número de procesadores de la máquina a usar y se concentra la atención en explotar oportunidades de paralelismo.

Comunicación: Se determina la comunicación requerida para coordinar las tareas.

Se definen estructuras y algoritmos de comunicación.

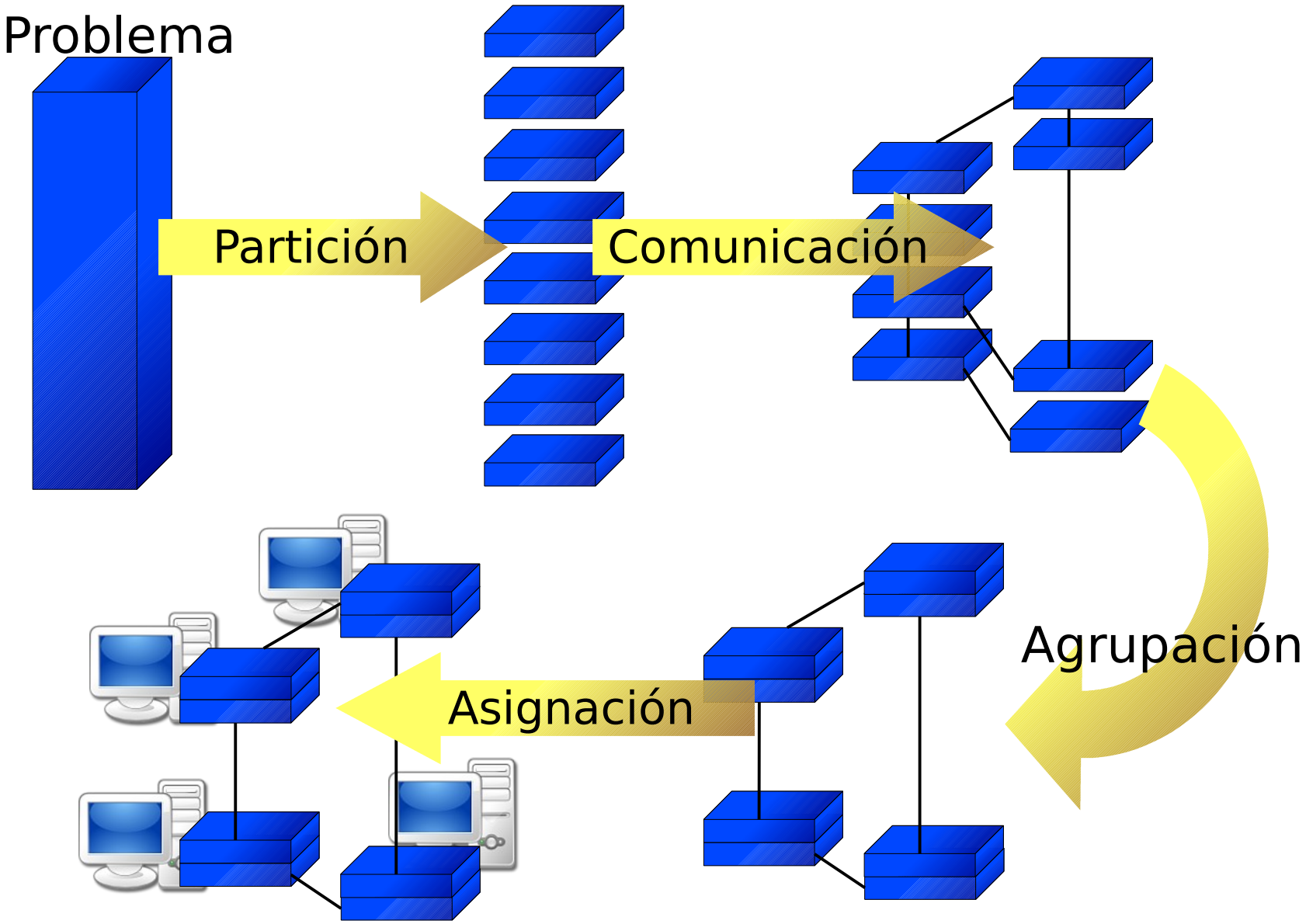
Agrupamiento: El resultado de las dos etapas anteriores es evaluado en términos de eficiencia y costos de implementación.

De ser necesario, se agrupan tareas pequeñas en tareas más grandes.

Asignación: Cada tarea es asignada a un procesador tratando de maximizar la utilización de los procesadores y de reducir el costo de comunicación.

La asignación puede ser estática (se establece antes de la ejecución del programa) o en tiempo de ejecución mediante algoritmos de balanceo de carga.

Gráficamente tenemos



En la etapa de partición se buscan oportunidades de paralelismo y se trata de subdividir el problema lo más finamente posible, es decir; que la granuralidad sea fina.

Un grano es una medidad del trabajo computacional a realizar.

Evaluaciones futuras podrán llevar a aglomerar tareas y descartar ciertas posibilidades de paralelismo. Una buen particionamiento divide tanto los cómputos como los datos.

- Descomposición funcional
- Descomposición de dominio

Al particionar se deben tener en cuenta los siguientes aspectos:

- El número de tareas debe ser por lo menos un orden de magnitud superior al número de procesadores disponibles para tener flexibilidad en las etapas siguientes.
- Hay que evitar cómputos y almacenamientos redundantes; de lo contrario el algoritmo puede ser no extensible a problemas más grandes.

Al particionar se deben tener en cuenta los siguientes aspectos:

- Hay que tratar de que las tareas sean de tamaños equivalentes ya que facilita el balanceo de la carga de los procesadores.
- Considere alternativas de paralelismo en esta etapa ya que pueden flexibilizar etapas subsecuentes.

Al particionar se deben tener en cuenta los siguientes aspectos:

- El número de tareas debe ser proporcional al tamaño del problema. Así, se podrá resolver problemas más grandes cuando se tenga más procesadores. Es decir, que el algoritmo sea escalable.

La comunicación requerida por un algoritmo puede ser definida en dos fases.

- Primero se definen los canales que conectan las tareas que requieren datos con las que los poseen.
- Segundo se especifica la información o mensajes que deben ser enviado y recibidos en estos canales.

En ambientes de memoria distribuida las tareas interactuan enviando y recibiendo mensajes.

Pero en ambientes de memoria compartida se debe utilizar mecanismos de sincronizacion para controlar el acceso a la memoria como: Semáforos, semáforos binarios, barreras, etc.

En la etapa de comunicación hay que tener en cuenta los siguientes aspectos:

- Todas las tareas deben efectuar aproximadamente el mismo número de operaciones de comunicación. Si esto no se da, es muy probable que el algoritmo no sea extensible a problemas mayores ya que habrán cuellos de botella.

En la etapa de comunicación hay que tener en cuenta los siguientes aspectos:

- La comunicación entre tareas debe ser tan pequeña como sea posible.
- Las operaciones de comunicación deben poder proceder concurrentemente.
- Los cómputos de diferentes tareas deben poder proceder concurrentemente.

Hasta ahora tenemos un algoritmo abstracto en el sentido de que no se tomó en cuenta la máquina sobre el cual correrá.

En la etapa de agrupación se va de lo abstracto a lo concreto y se revisa el algoritmo obtenido tratando de considerar si es útil agrupar tareas y si vale la pena replicar datos y/o cómputos de acuerdo a la plataforma seleccionada.

En la fase de partición se trata de establecer el mayor número posible de tareas con la intensión de explorar al máximo las oportunidades de paralelismo.

Esto no necesariamente produce un algoritmo eficiente ya que el costo de comunicación puede ser significativo.

Mediante la agrupación de tareas se puede reducir la cantidad de datos a enviar y así reducir el número de mensajes a transmitir y por ende el costo de comunicación

Reduciendo el número de mensajes, a pesar de que se envíe la misma cantidad de información, se reduce el costo de crear un nuevo canal de comunicación.

Así mismo se puede intentar replicar cómputos y/o datos para reducir los requerimientos de comunicación.

Por otro lado, también se debe considerar el costo de creación de tareas y el costo de cambio de contexto (context switch) en caso de que se asignen varias tareas a un mismo procesador.

En caso de tener distintas tareas corriendo en plataformas de memoria distribuida, se debe tratar de que la obtener una granularidad gruesa.

Es decir, que exista una cantidad de computo significativa antes de tener necesidades de comunicación.

Se puede tener granularidad media si la aplicación correrá sobre una máquina de memoria compartida.

En estas máquinas el costo de comunicación es menor siempre y cuando el número de tareas y procesadores se mantenga dentro de cierto rango.

En la etapa de agrupación se debe tener en cuenta los siguientes aspectos:

- Chequear si la agrupación redujo los costos de comunicación.
- Si se han replicado cómputos y/o datos, se debe verificar que los beneficios son superiores a los costos.
- Se debe verificar que las tareas resultantes tengan costos de computo y comunicación similares.

En la etapa de agrupación se debe tener en cuenta los siguientes aspectos:

- Hay que revisar si el número de tareas es extensible con el tamaño del problema.
- Si el agrupamiento ha reducido las oportunidades de ejecución concurrente, se debe verificar que aun hay suficiente concurrencia y posiblemente considerar diseños alternativos.

En la etapa de agrupación se debe tener en cuenta los siguientes aspectos:

- Analizar si es posible reducir aun más el número de tareas sin introducir desbalances de cargas o reducir la extensibilidad.

En la etapa de asignación se determina en que procesador se ejecutará cada tarea.

Este problema no se presenta en máquinas de memoria compartida UMA.

La asignación de tareas puede ser realizada de las siguientes maneras:

- Estática: una tarea es asignada a un procesador desde su inicio hasta su fin
- Dinámica: una tarea puede ser migrada durante su ejecución. Esto puede agregar un costo adicional