Luis Miguel Benitez Ruiz

994361

Tarea 1: Desarrollo de Algoritmos Paralelos

Para construir un algoritmo paralelo, aunque no existe una receta, se pueden seguir generalmente tres pasos. La descomposición, que divide la tarea global en muchas tareas pequeñas, la obtención de dependencias, que permite ver que tareas si se pueden paralelizar y que tareas no, y finalmente la elección de un modelo de concurrencia. A continuación se describen a más detalle.

Descomposición

La descomposición es el proceso de dividir un proceso computacional en procesos más pequeños. A estos procesos se les llaman Tareas, y su tamaño está definida completamente por los programadores que hagan la descomposición del proceso completo. Del tamaño de las tareas dependerá la granularidad de la descomposición. Una granularidad fina se obtiene descomponiendo un proceso en muchas tareas pequeñas, mientras que granularidad gruesa se obtiene descomponiéndolo en tareas grandes.

La mejor opción de descomposición dependerá de la tarea y del modelo del algoritmo paralelo que se quiera seguir.

Establecimiento de dependencias.

Una vez que se ha descompuesto el proceso en tareas, se procede a establecer la dependencia entre tareas. Una de las formas de establecer esto es con grafos de dependencia de tareas o dependencia de datos. Ambos son grafos dirigidos, donde cada nodo ‘N’ tiene una arista que dirige a cada uno de los nodos que dependan, para su ejecución, de que N termine. El peso de cada nodo establece el costo computacional de dicha tarea.

Una forma de medir el grado promedio de concurrencia que se tiene para un conjunto determinado de tareas, requiere establecer el grafo de dependencia de las mismas. Se suma el peso total de las tareas a ejecutar, y después se obtiene el peso del camino crítico. El camino crítico es el camino más largo que inicie en un nodo raíz y termine en un nodo hoja (en el final del grafo). La división del peso total del grafo entre el peso del camino critico estima el grado promedio de concurrencia. Esto indica el número promedio de tareas que pueden ejecutarse de forma concurrente durante la ejecución de todo el algoritmo. Diferentes descomposiciones resultaran en diferentes grafos y en diferentes grados promedio de concurrencia.

Modelos de Algoritmos Paralelos

Hay diferentes modelos que pueden seguirse para definir el algoritmo, una vez que se ha realizado la descomposición del proceso y se ha establecido la dependencia entre tareas.

Datos-Paralelos

Este es el modelo más sencillo, cuando los procesos aplican la misma tarea sobre diferentes datos. Un ejemplo seria la multiplicación de matrices, pues la misma operación (multiplicación) se aplica sobre muchos datos diferentes de forma concurrente.

Grafo de Datos

Consiste en modelar las tareas a realizarse en base a un grafo de dependencia de tareas. Las tareas que resulten independientes entre sí son las que podrían paralelizarse. En este modelo no suele haber un mapeo de proceso a tareas, si no que de forma predeterminada ciertos procesos se asignan a las tareas.

Pool de Trabajo

En este modelo cualquier tarea puede ser asignada a cualquier proceso. Se mantiene una estructura de datos que sirva como repositorio de trabajos, y puede ser distribuida y centralizada. Los procesos pueden generar trabajos y agregarlos al directorio, o elegir un trabajo disponible del directorio y trabajar en el mismo. Se necesita una condición que determine cuando se ha terminado el trabajo en su totalidad.

Amo-Esclavo o Gerente-Trabajador

Bajo este modelo hay uno o más procesos *gerentes* que generan trabajos y los asignan a los *trabajadores* que tengan bajo su mando. El modelo preferido es donde cada trabajador comienza su trabajo de forma independiente a los otros, para maximizar el aprovechamiento de los recursos disponibles.

En este modelo es importante que las unidades de trabajo sean de tamaño suficiente para que la repartición del mismo a diferentes unidades valga la pena. En caso contrario solo se hará más lento el proceso, pues al trabajo que se quiere hacer habrá que añadirle el tiempo del proceso *gerente*.

Productor-Consumidor o Pipeline

Este modelo es como una línea de producción. Al inicio está un proceso productor de una serie de datos que necesitan ser procesados a través de varios pasos. Una vez que el pipeline este lleno, cada tarea a lo largo de el mismo puede ejecutarse en paralelo a las otras, sobre los datos que tengan disponibles en ese momento. Todos estos componentes del pipeline pueden verse como consumidores de los datos que se encuentren en su etapa del pipeline.

Modelos Híbridos

No siempre las soluciones existentes son la mejor opción para un problema. También pueden combinarse cualquiera de los modelos anteriores en modelos nuevos.

Referencias

Ananth Grama, A. G. (2003). *Introduction to Parallel Computing, Second Edition.* Addison Wesley.