PROCESAMIENTO PARALELO

2024

LIC. MARTHA SEMKEN - LIC. MARIANO VARGAS

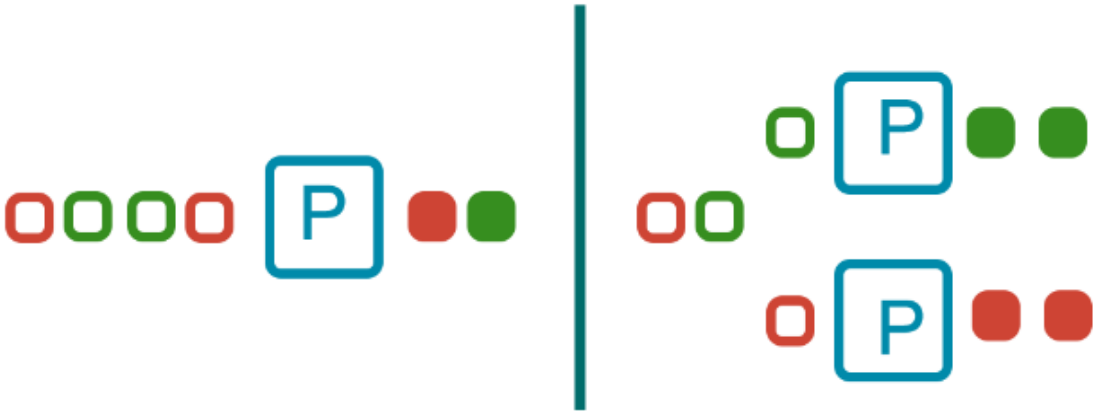


Tabla de contenido

Introducción

La computación paralela es un campo de estudio que se enfoca en la ejecución simultánea de múltiples tareas o procesos en un sistema de computadoras. A diferencia de la computación secuencial tradicional, donde las instrucciones se ejecutan una tras otra en una sola unidad de procesamiento, la computación paralela utiliza múltiples unidades de procesamiento para realizar tareas de forma simultánea, lo que puede mejorar significativamente el rendimiento y la eficiencia de los sistemas computacionales.

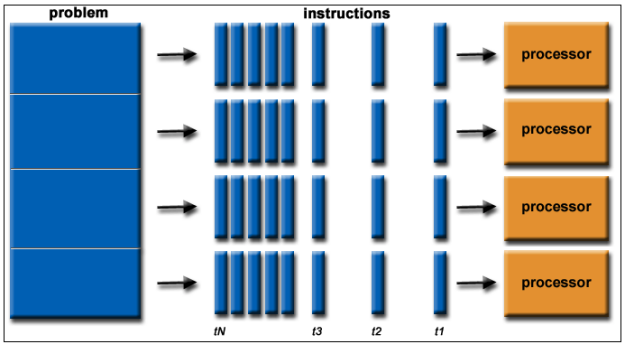
En un entorno de computación paralela, los problemas computacionales se dividen en tareas más pequeñas y se ejecutan en paralelo en diferentes procesadores o núcleos de procesamiento. Esto permite que múltiples tareas se realicen al mismo tiempo, lo que puede acelerar el tiempo de procesamiento y resolver problemas más grandes y complejos en un tiempo menor.

La computación paralela se utiliza en una variedad de aplicaciones, incluyendo la simulación científica, el procesamiento de grandes conjuntos de datos, el renderizado de gráficos en 3D, el análisis de datos en tiempo real y más. Con el crecimiento exponencial de la cantidad de datos y la complejidad de los problemas computacionales, la computación paralela se ha vuelto cada vez más relevante y es una herramienta fundamental en el campo de la informática moderna.

En este apunte, exploraremos los fundamentos de la computación paralela, incluyendo sus conceptos básicos, arquitecturas, modelos de programación, aplicaciones y desafíos. Además, analizaremos cómo aprovechar el potencial de la computación paralela para mejorar el rendimiento y la eficiencia en una variedad de escenarios computacionales.

¿Qué es el procesamiento paralelo y para qué se utiliza?

Si nos transportamos un par de décadas atrás, una computadora tradicional contaba con una unidad de procesamiento individual para ejecutar las instrucciones especificadas en un programa. Los problemas eran divididos en series discretas de instrucciones, donde estas eran ejecutadas de a una por vez, unas tras otras. Una manera de incrementar el poder de cómputo es usar más de una unidad de procesamiento dentro de una única computadora o bien utilizar varias computadoras, trabajando todas juntos sobre el mismo problema. El problema es dividido en partes discretas, donde cada una de ellas es resuelta por una unidad de procesamiento individual de manera paralela. Se puede definir al procesamiento paralelo como el uso simultáneo de múltiples recursos computacionales para resolver un problema.[i]



La necesidad de la computación paralela se origina por las limitaciones de los computadores secuenciales: integrando varios procesadores para llevar a cabo la computación es posible resolver problemas que requieren de más memoria o de mayor

velocidad de cómputo. El objetivo principal de la computación paralela es reducir el tiempo de resolución de problemas computacionales, o bien para resolver problemas más grandes que no podrían ser resueltos por un computador convencional. Para llevar a cabo esto, es necesario emplear sistemas de cómputo de altas prestaciones y algoritmos paralelos que utilicen estos sistemas eficientemente.

Los problemas habituales en los cuales se aplica la programación paralela son: problemas con alta demanda de cómputo, problemas que requieren procesar una gran cantidad de datos, o problemas de tiempo real, en los que se necesita la respuesta en un tiempo máximo. De esta forma, la comunidad científica usa la computación paralela para resolver problemas que sin el paralelismo serían intratables, o con tiempos de respuesta inaceptables. Algunos campos que se favorecen de la programación paralela son: predicciones y estudios meteorológicos, estudio del genoma humano, modelado de la biosfera, predicciones sísmicas, simulación de moléculas, modelización y simulación financiera, computación gráfica, realidad virtual, motores de búsqueda web, exploración de hidrocarburos, diseño de fármacos, entre otros.

Aunque tradicionalmente el objetivo primario de HPC fue reducir el tiempo de ejecución de las aplicaciones, en las últimas décadas la eficiencia energética ha cobrado un valor semejante. Esto se debe al elevado consumo energético y generación de calor de las arquitecturas paralelas que afectan al funcionamiento y repercuten en el costo, debido a la adquisición de equipos para refrigeración y al consumo energético que generan los mismos.

Dificultades

Existen diversas dificultades que se pueden encontrar a la hora de escribir un programa paralelo. Por ejemplo, no siempre es posible paralelizar un programa; la paralelización requiere de tareas que no están presentes en la programación secuencial (descomposición del problema, comunicación y sincronización, mapeo, entre otras); mayor propensión a cometer errores por aumento de la complejidad; mayor dificultad a la hora de probar o depurar un programa; y por último, la fuerte dependencia entre el programa paralelo y la arquitectura de soporte para obtener alto rendimiento.

Una plataforma paralela consiste de dos o más unidades de procesamiento vinculadas a partir de algún tipo de red de interconexión. Es posible clasificarlas de dos formas, según su organización lógica o según su organización física. Se entiende como organización lógica, a la manera en que el programador visualiza la plataforma paralela.

Por otro lado, la organización física se refiere al hardware real de la plataforma. A continuación se describen los diferentes tipos de plataformas paralelas.

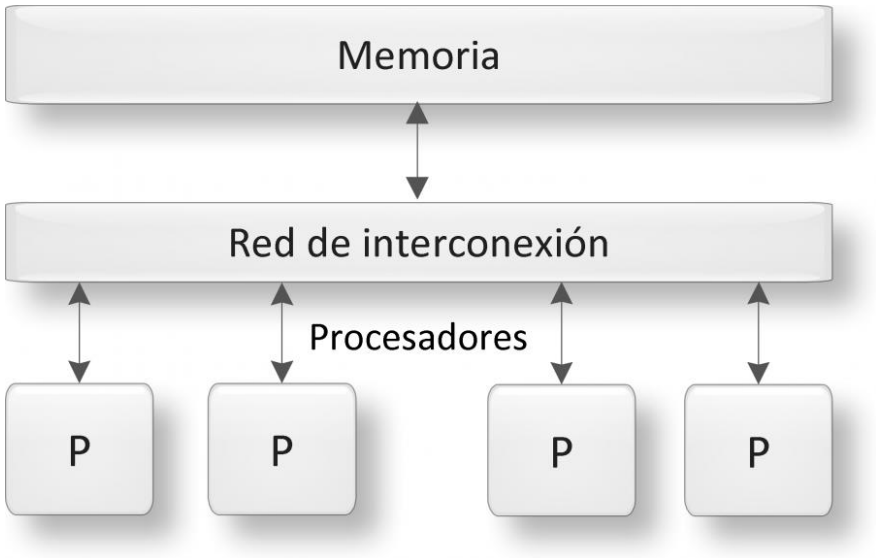
Plataformas de memoria compartida

Un multiprocesador de memoria compartida consiste en dos o más unidades de procesamiento conectadas a múltiples módulos de memoria. La forma de conexión de estos elementos es a través de algún tipo de red de interconexión como puede ser un switch crossbar o un bus. En estos sistemas, existe un único espacio de direcciones, por lo que todas las unidades de procesamiento pueden acceder a la misma dirección de memoria, y modificar los datos almacenados en este espacio compartido. Si todas las unidades de procesamiento tienen el mismo tiempo para acceder a cualquier dirección de memoria,

entonces se dice que el multiprocesador tiene acceso uniforme a memoria (UMA). Por el contrario, si el tiempo de acceso a algunas direcciones de memoria es mayor que a otras, entonces se dice que el multiprocesador es de acceso no uniforme a memoria (NUMA).

Idealmente, se desea que el sistema sea UMA. No obstante, los grandes sistemas de memoria compartida suelen ser de tipo NUMA dada la dificultad de implementar hardware que provea un acceso rápido a toda la memoria compartida. Por lo que cuentan con alguna estructura de memoria jerárquica o distribuida.

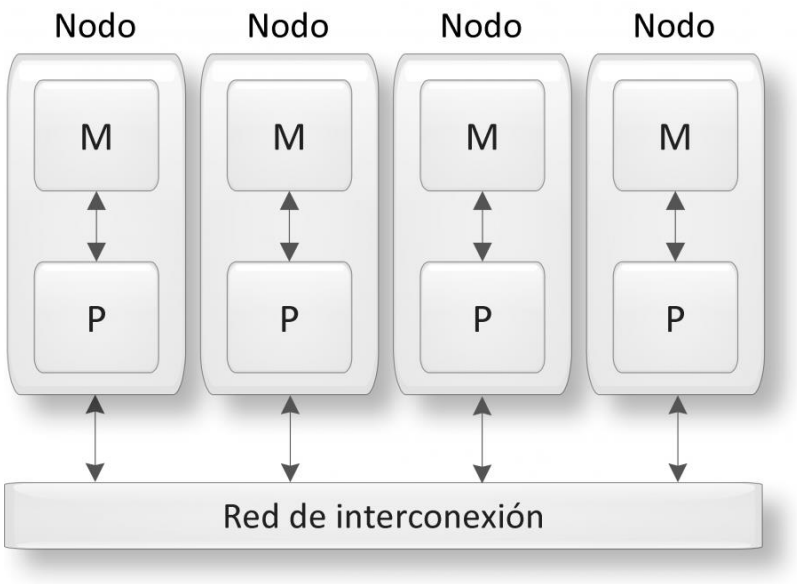
Tanto los sistemas UMA como los NUMA, cuentan con una memoria caché de alta velocidad para mantener los contenidos recientemente referenciados de las direcciones de memoria principal. A pesar de obtener mejoras con la presencia de cachés en las unidades de procesamiento, este tipo de memoria también acarrea la problemática de tener múltiples copias de una única palabra de memoria siendo manipulada por más de una unidad de procesamiento al mismo tiempo. Un ejemplo de este tipo de arquitecturas son los multicores.



Plataformas de memoria distribuida

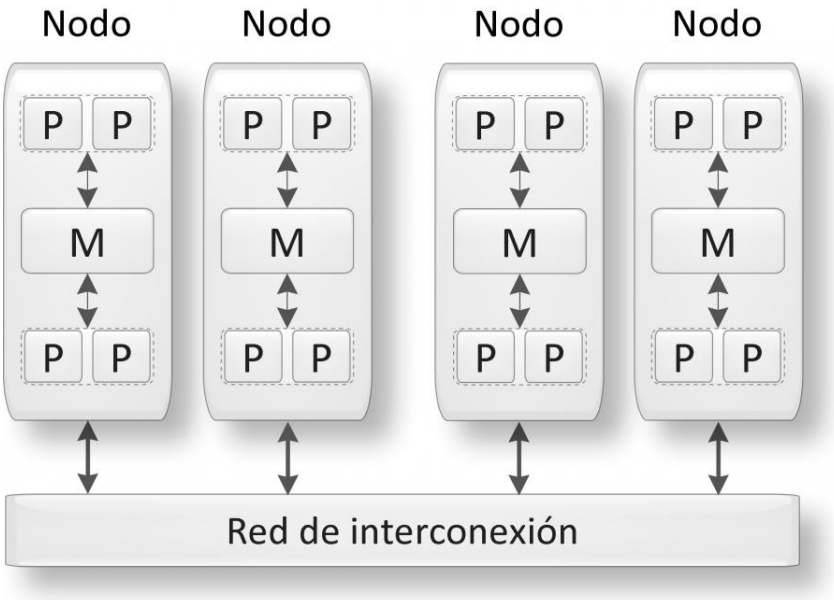
Una plataforma de memoria distribuida consiste de varios nodos de procesamiento independientes con módulos de memoria locales, esto significa que cada uno cuenta con su propio espacio de direcciones. Además, estos nodos están conectados por una red de interconexión. Cada nodo puede ser una computadora individual o un multiprocesador de memoria compartida. Al contar con su propio espacio de memoria, el mismo no es accesible por el resto y los nodos deben comunicarse entre sí enviándose mensajes. Éste intercambio de mensajes es utilizado para transferir datos, trabajo y sincronizar acciones entre nodos

A la hora de escalar físicamente alguna de estas plataformas, resulta más fácil hacerlo en los sistemas de memoria distribuida que en los sistemas de memoria compartida.



Plataformas híbridas

Con la incorporación de los procesadores multicore a las arquitecturas de clusters tradicionales, surgió una nueva plataforma híbrida conocida como cluster de multicores. Con la aparición de esta plataforma, surgen nuevos tipos de comunicación debido a la heterogeneidad de la arquitectura, que se clasifican en: inter-nodo e intranodo. El primero se da entre los núcleos de distintos nodos y se lleva a cabo mediante envío de mensajes a través de la red de interconexión. El segundo se da entre los distintos núcleos que tiene un nodo y se lleva a cabo a través la jerarquía de memoria que estos núcleos comparten.



Clusters

Un cluster es un tipo de sistema de procesamiento paralelo compuesto por un conjunto de componentes de hardware estándares interconectadas vía algún tipo de red, las cuales cooperan configurando un recurso que se ve como único e integrado, más allá

de la distribución física de sus componentes. Cada uno de los componentes que conforman un cluster se denomina nodo y son los encargados de llevar adelante el procesamiento. La construcción de los nodos de un clúster es relativamente fácil y económica debido a su flexibilidad: pueden tener toda la misma configuración de hardware y sistema operativo (clúster homogéneo), o tener diferente hardware y/o sistema operativo (clúster heterogéneo). Esta característica constituye un elemento importante en el análisis del rendimiento que se puede obtener de un clúster como máquina paralela.

Para que los nodos se comuniquen entre sí, es necesario proveerlos de un medio de interconexión mediante algún tipo de red de alta velocidad, por ejemplo, una red LAN. Sin embargo, no basta sólo con interconectar nodos para que un cluster funcione como tal, sino que es necesario proveer al mismo de un sistema de administración de cluster, el cual se encargue de interactuar con el usuario y las aplicaciones que se ejecuten en él.

Éste tipo de sistemas ofrece una manera rentable de mejorar el rendimiento (velocidad, disponibilidad, rendimiento, etc.) comparado con supercomputadoras de similares características. A continuación se detallan los tipos de cluster según sus características.

- Cluster de alto rendimiento (High Performance Clusters): se caracterizan por ejecutar tareas que requieren de gran capacidad computacional, grandes cantidades de memoria, o ambos a la vez. El realizar estas tareas puede comprometer los recursos del cluster por periodos de tiempo indeterminados.
- Cluster de alta disponibilidad (High Availability): el objetivo principal de estos cluster se centra en la disponibilidad y la conabilidad. Los clusters que pertenecen a estar categoría intentan brindar la máxima disponibilidad de los servicios que ofrecen. La conabilidad es provista mediante software que detecta fallos y permite recuperarse frente a los mismos, mientras que por medio de hardware se previene tener un único punto de fallos.
- Cluster de alta eficiencia (High Throughout): son clusters cuyo objetivo de diseño se centra en ejecutar la mayor cantidad de tareas en el menor tiempo posible. Para poder llevar adelante esto debe existir independencia de datos entre las tareas individuales. El retardo entre los nodos del cluster no es considerado un gran problema.

Si bien cada tipo de cluster visto tiene sus características representativas, todos tienen un objetivo común: obtener un alto rendimiento a un bajo costo. Si a esto se le suma que pueden escalarse fácilmente, esto explica por qué el uso de clusters es hoy una de las posibilidades de cómputo paralelo/distribuido más elegidas.

[i] Análisis del Uso de un Cluster de Raspberry Pi para Cómputo de Altas Prestaciones- Autor: Lic. Pablo Sebastián Rodríguez Eguren. UNLP