

Investigación Semana 2

Procesamiento paralelo

El procesamiento paralelo implica la división de un problema en subproblemas que se resuelven simultáneamente en múltiples núcleos o procesadores. Se basa en la idea de que muchos procesos pueden ejecutarse al mismo tiempo para reducir el tiempo total de ejecución. Los sistemas multiprocesador, multiprocesadores, GPUs y clústeres son ejemplos de arquitecturas que permiten el procesamiento paralelo.

Modelos de cómputo paralelos

Existen varios modelos de cómputo paralelos como:

- Memoria compartida: donde múltiples hilos acceden a una memoria común, por ejemplo usando OpenMP.
- Memoria distribuida: donde cada nodo posee su propia memoria local y se comunican mediante mensajes, ejemplo MPI.
- Modelo híbrido: combina memoria compartida y distribuida, para aprovechar jerarquías de memoria.

Desempeño computacional de algoritmos paralelos

El desempeño de algoritmos paralelos se mide en términos de aceleración (speedup) y eficiencia.

Aceleración (S) = $T_{\text{serial}} / T_{\text{paralelo}}$. La eficiencia (E) = S / P , donde P es el número de procesadores.

El objetivo es acercarse a una aceleración lineal, pero se ven limitaciones por sobrecarga de gestión de hilos, sincronización y comunicación entre procesos.

Complejidad de la comunicación

La complejidad de la comunicación se refiere al costo asociado con el intercambio de datos entre procesadores en un sistema paralelo. Se modela mediante parámetros como latencia (α) y ancho de banda (β). El tiempo de comunicación para enviar un mensaje de tamaño n se estima como $T_{\text{comm}} = \alpha + \beta * n$. A medida que aumenta la cantidad de procesadores, la comunicación puede convertirse en un cuello de botella.

Optimización

La optimización en sistemas paralelos incluye técnicas para minimizar la sobrecarga de comunicación, equilibrar la carga entre procesadores y reducir la latencia. Ejemplos:

- Descomposición de dominio: particionar datos para reducir dependencias.

- Alineación de datos: ubicar datos cercanos físicamente para reducir tiempos de acceso.
- Fusión de comunicaciones: agrupar múltiples mensajes en uno solo.

Notification Push

Las notificaciones push son alertas enviadas desde un servidor hacia los dispositivos de los usuarios sin necesidad de que estos soliciten la información activamente. Tecnologías como Firebase Cloud Messaging (FCM) en Android o Apple Push Notification Service (APNS) en iOS permiten implementar este servicio. Se centran en una arquitectura cliente-servidor donde el servidor envía mensajes a través de brokers hacia los dispositivos registrados.

Docker

Docker es una plataforma que utiliza contenedores para empaquetar aplicaciones y sus dependencias en un entorno aislado. Esto garantiza que la aplicación se ejecute de manera consistente en diferentes entornos. Un contenedor comparte el kernel del sistema operativo pero tiene su propio espacio de usuario, sistema de archivos y procesos.

Docker-Compose

Docker-Compose es una herramienta que permite definir y ejecutar aplicaciones multi-contenedor. Se utiliza un archivo docker-compose.yml para configurar servicios, redes y volúmenes. Con un solo comando (docker-compose up), todos los servicios definidos se inician y conectan según la configuración.

Nginx

Nginx es un servidor web de alto rendimiento, que también puede actuar como proxy inverso, balanceador de carga y servidor de caché. Está optimizado para manejar un gran número de conexiones concurrentes mediante un modelo de eventos asíncronos. Se utiliza comúnmente para servir contenido estático, como proxy inverso para aplicaciones backend o balancear cargas entre múltiples servidores.

Amazon Web Services Principales Servicios

AWS ofrece numerosos servicios en la nube. Algunos de los principales:

- EC2 (Elastic Compute Cloud): instancias de servidor en la nube.
- S3 (Simple Storage Service): almacenamiento de objetos escalable.
- RDS (Relational Database Service): bases de datos relacionales gestionadas.
- Lambda: ejecución de funciones sin servidor.
- VPC (Virtual Private Cloud): redes virtuales aisladas para recursos.

- CloudFront: CDN para distribución de contenido.
- ECS/EKS: orquestación de contenedores con Docker y Kubernetes.