

Introducción a las Arquitecturas Distribuidas

Juan Mario Haut



QUÉ ES LA COMPUTACIÓN DISTRIBUIDA



COMPUTACIÓN DISTRIBUIDA

La computación distribuida es un modelo de procesamiento en el que múltiples computadores trabajan de manera conjunta para resolver un problema o procesar grandes volúmenes de datos.

- **División de tareas:** Un problema complejo se descompone en subtareas más pequeñas que se distribuyen entre diferentes máquinas.
- **Concurrencia y paralelismo:** Varias máquinas o nodos pueden ejecutar tareas simultáneamente, acelerando el procesamiento.
- **Coordinación y sincronización:** Las máquinas deben coordinarse para garantizar que las subtareas se completen correctamente y los resultados se combinen de manera eficiente.



COMPUTACIÓN DISTRIBUIDA

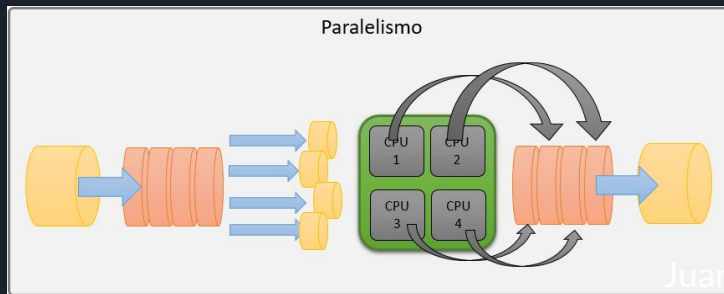
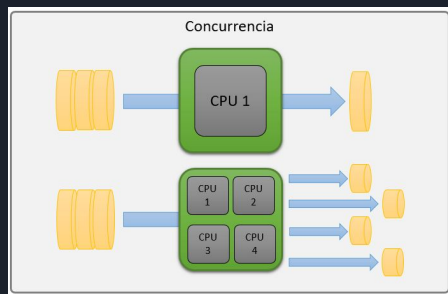
La concurrencia y el paralelismo son conceptos relacionados pero distintos en la programación, y a menudo se confunden.

- **Concurrencia:**
- **Paralelismo**

COMPUTACIÓN DISTRIBUIDA

La concurrencia y el paralelismo son conceptos relacionados pero distintos en la programación, y a menudo se confunden.

- **Concurrencia:** La concurrencia es la capacidad de un sistema para gestionar varias tareas que dan la impresión de ejecutarse simultáneamente, aunque no lo hagan de manera literal.
- **Paralelismo:** El paralelismo ocurre cuando múltiples tareas o procesos realmente se ejecutan simultáneamente, utilizando múltiples procesadores o núcleos en el sistema.





ASPECTOS FUNDAMENTALES

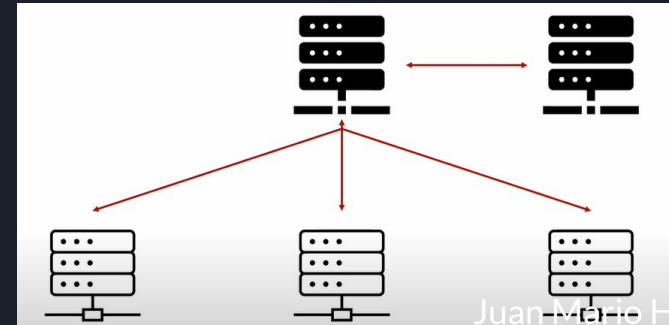
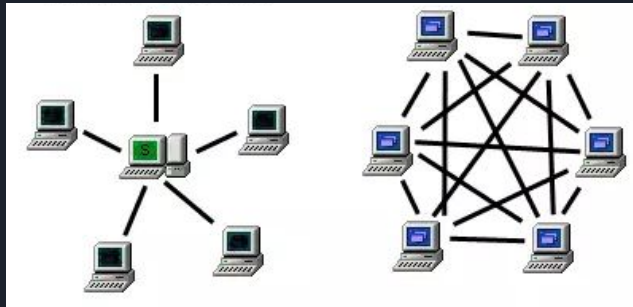
ARQUITECTURAS

- Cliente-Servidor:
- Peer-to-Peer (P2P):
- Arquitectura Maestro-Esclavo:

ASPECTOS FUNDAMENTALES

ARQUITECTURAS

- **Cliente-Servidor:** Los nodos (clientes) solicitan servicios o recursos a servidores centralizados.
- **Peer-to-Peer (P2P):** No hay un servidor central. Todos los nodos actúan como iguales y pueden ser clientes y servidores a la vez.
- **Arquitectura Maestra-Esclava:** Un nodo maestro asigna tareas a nodos esclavos, como en el caso de Hadoop con su MapReduce.





ASPECTOS FUNDAMENTALES

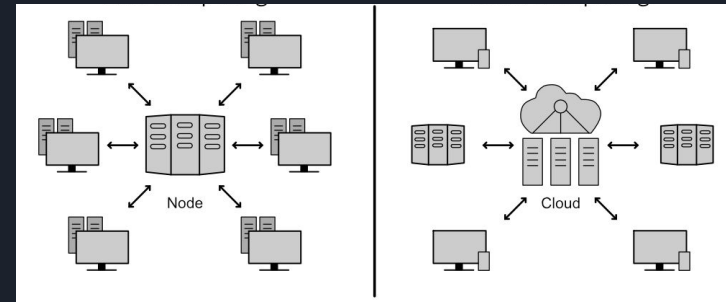
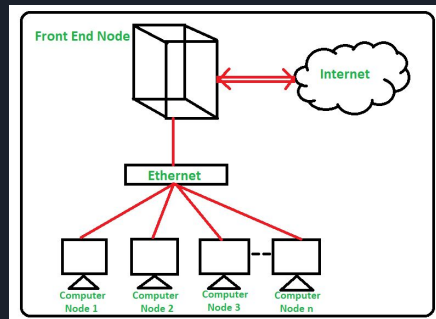
TIPOS DE COMPUTACIÓN DISTRIBUIDA

- Clústeres:
- Grid Computing:
- Cloud Computing:

ASPECTOS FUNDAMENTALES

TIPOS DE COMPUTACIÓN DISTRIBUIDA

- **Clústeres:** Grupos de computadores conectados para trabajar como una sola unidad. Ejemplo: sistemas de alto rendimiento (HPC).
- **Grid Computing:** Computadoras distribuidas geográficamente que cooperan para resolver un problema común.
- **Cloud Computing:** Recursos distribuidos a través de la nube (por ejemplo, AWS, Google Cloud).





ASPECTOS FUNDAMENTALES

TOLERANCIA A FALLOS:



ASPECTOS FUNDAMENTALES

TOLERANCIA A FALLOS: En los sistemas de computación distribuida, la tolerancia a fallos es un aspecto crucial, ya que estos sistemas están compuestos por múltiples nodos que pueden fallar de forma individual sin afectar el funcionamiento global. Para lograr esto, los sistemas emplean varias técnicas:

- **Réplicas de datos:** Los datos se copian en varios nodos, de modo que si uno falla, otros pueden asumir su función sin pérdida de información. Esto garantiza que el sistema continúe operando y los usuarios no experimenten interrupciones.
- **Algoritmos de consenso:** Para asegurar que los nodos en un sistema distribuido mantengan una vista coherente de los datos, incluso en caso de fallos, se utilizan algoritmos como Raft o Paxos.
- **Detección y recuperación:** Los sistemas también pueden detectar automáticamente fallos en los nodos y reasignar las tareas pendientes a otros nodos funcionales, minimizando el impacto del fallo.



ASPECTOS FUNDAMENTALES

ESCALABILIDAD: La escalabilidad se refiere a la capacidad de un sistema distribuido para aumentar su rendimiento al agregar más recursos, como nodos o máquinas, sin degradar el rendimiento global.

ASPECTOS FUNDAMENTALES

ESCALABILIDAD VERTICAL

Consiste en aumentar los recursos de un solo servidor o nodo, como agregar más CPU, memoria RAM o almacenamiento. En lugar de agregar más nodos, se mejora la capacidad del hardware de un solo servidor para manejar una mayor carga de trabajo.

Ventajas: Es simple y no hay necesidad de dividir el dato en más nodos.

→ Equivalente a usar la misma torre cambiando el procesador y la gráfica



ASPECTOS FUNDAMENTALES

ESCALABILIDAD VERTICAL



Consiste en aumentar los recursos de un solo servidor o nodo, como agregar más CPU, memoria RAM o almacenamiento. En lugar de agregar más nodos, se mejora la capacidad del hardware de un solo servidor para manejar una mayor carga de trabajo.

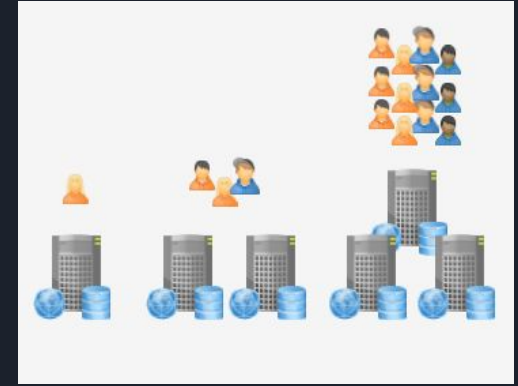
Ventajas: Es simple y no hay necesidad de dividir el dato en más nodos.

Inconvenientes: Caro y limitación física (tipo de socket)

→ Mejorar un servidor aumentando su memoria RAM y procesadores para manejar un mayor número de consultas sin tener que distribuirlas entre diferentes servidores.

ASPECTOS FUNDAMENTALES

ESCALABILIDAD HORIZONTAL



La escalabilidad se refiere a la capacidad de un sistema distribuido para aumentar su rendimiento al agregar más recursos, como nodos o máquinas, sin degradar el rendimiento global. Consiste en añadir más nodos o máquinas para distribuir la carga de trabajo de manera uniforme.

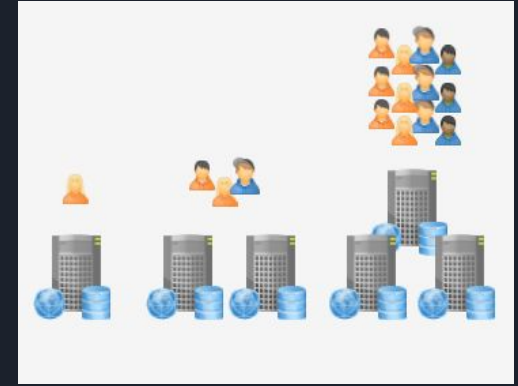
Ventajas: Mayor tolerancia a fallos, mejor distribución de carga y escalabilidad ilimitada.

Inconvenientes: Puede generar sobrecarga en la red, su gestión es más compleja

→ Un sistema como Hadoop, donde puedes agregar más nodos a un clúster de computación para procesar grandes volúmenes de datos en paralelo.

ASPECTOS FUNDAMENTALES

ESCALABILIDAD HORIZONTAL



La escalabilidad se refiere a la capacidad de un sistema distribuido para aumentar su rendimiento al agregar más recursos, como nodos o máquinas, sin degradar el rendimiento global. Consiste en añadir más nodos o máquinas para distribuir la carga de trabajo de manera uniforme.

Ventajas: Mayor tolerancia a fallos, mejor distribución de carga y escalabilidad ilimitada.

Inconvenientes: Puede generar sobrecarga en la red, su gestión es más compleja

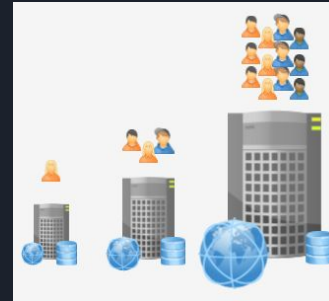
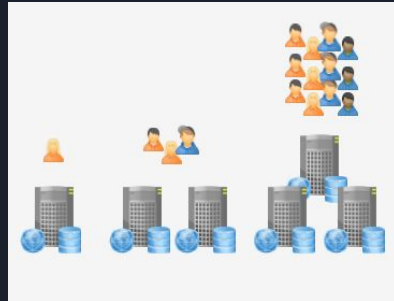
→ Un sistema como Hadoop, donde puedes agregar más nodos a un clúster de computación para procesar grandes volúmenes de datos en paralelo.

ASPECTOS FUNDAMENTALES

ESCALABILIDAD

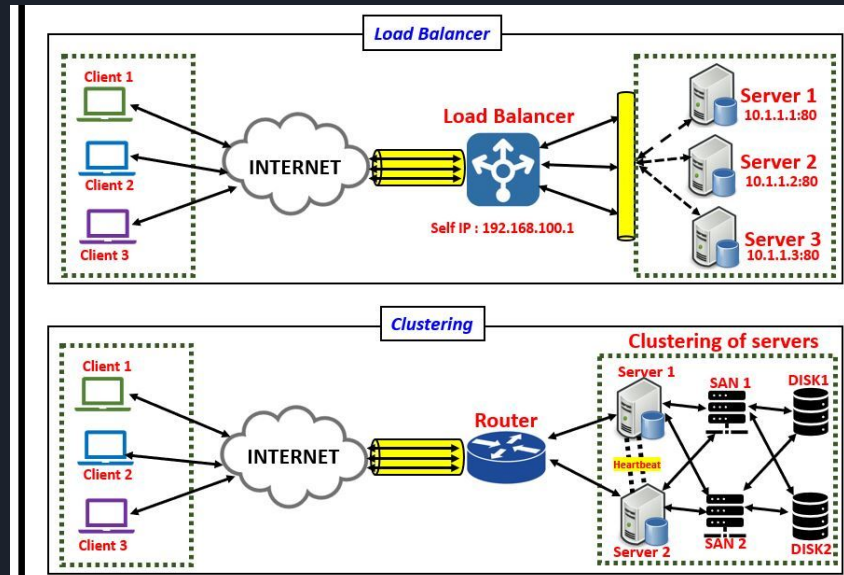
Escalabilidad horizontal es ideal cuando se busca crecer en capacidad agregando más nodos, lo que proporciona tolerancia a fallos y permite manejar cargas distribuidas. Es común en sistemas en la nube y en grandes infraestructuras distribuidas.

Escalabilidad vertical es útil cuando se desea maximizar los recursos de una máquina sin cambiar la arquitectura del sistema, pero tiene límites físicos y puede ser más costosa.



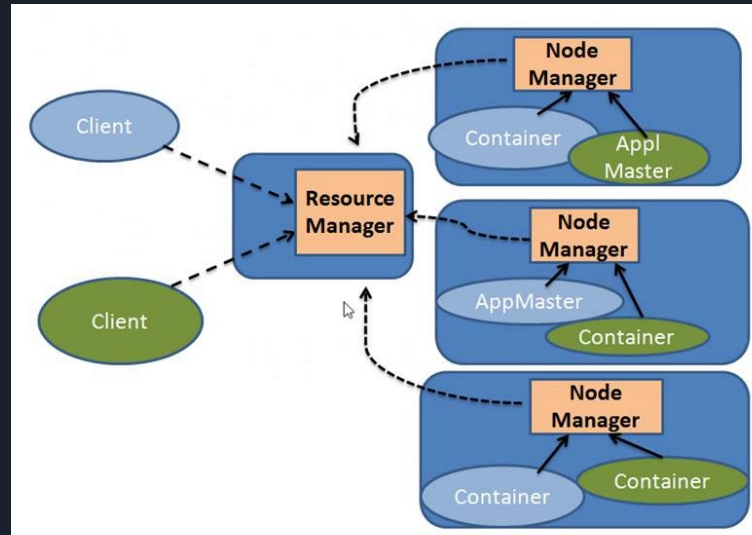
ASPECTOS FUNDAMENTALES

Balanceo de carga: Para mantener un rendimiento eficiente a medida que se agregan nodos, se utilizan técnicas de balanceo de carga, distribuyendo las tareas entre los nodos de manera equitativa para evitar sobrecargar a uno solo.



ASPECTOS FUNDAMENTALES

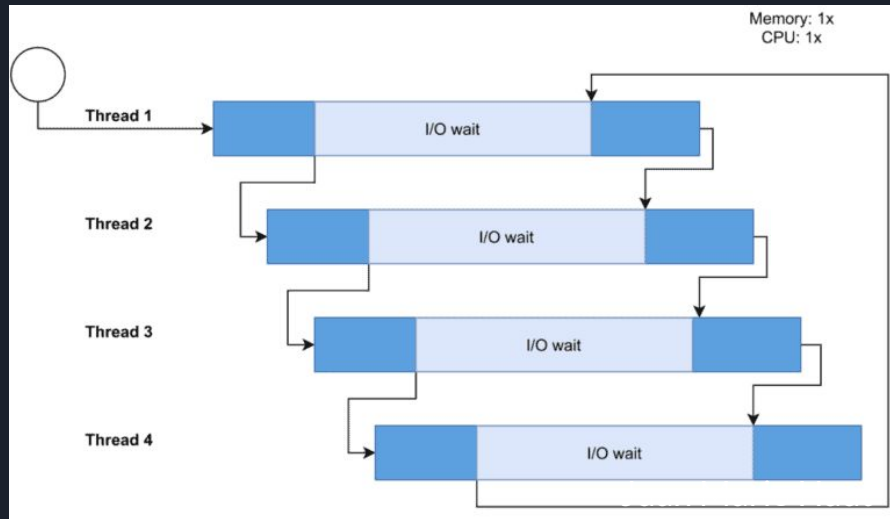
Elasticidad: Algunos sistemas distribuidos, como los basados en la nube, permiten ajustar dinámicamente la cantidad de recursos disponibles (añadir o eliminar nodos) según las necesidades del momento, asegurando que el sistema pueda manejar picos de carga sin perder rendimiento.



CONCURRENCIA

La concurrencia es la capacidad de un sistema para manejar múltiples tareas a la vez, avanzando de forma intercalada o solapada, a diferencia del paralelismo, que implica la ejecución simultánea en distintos procesadores. El objetivo principal de la concurrencia es maximizar el uso de la CPU minimizando su tiempo de inactividad.

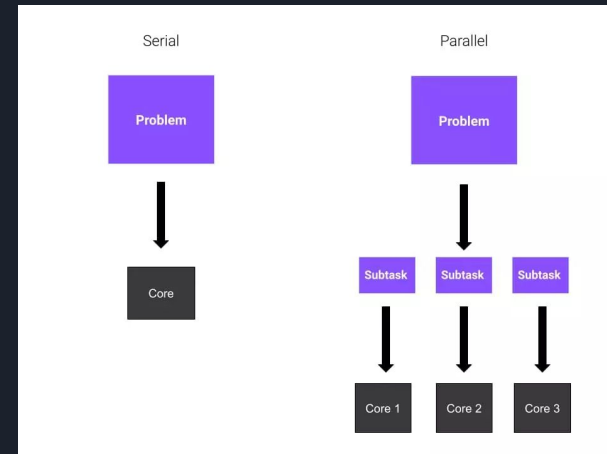
Mientras el hilo o proceso actual está esperando operaciones de entrada/salida, transacciones de base de datos o el lanzamiento de un programa externo, otro proceso o hilo recibe la asignación de la CPU. Por parte del kernel, el sistema operativo envía una interrupción a la tarea activa para detenerla-



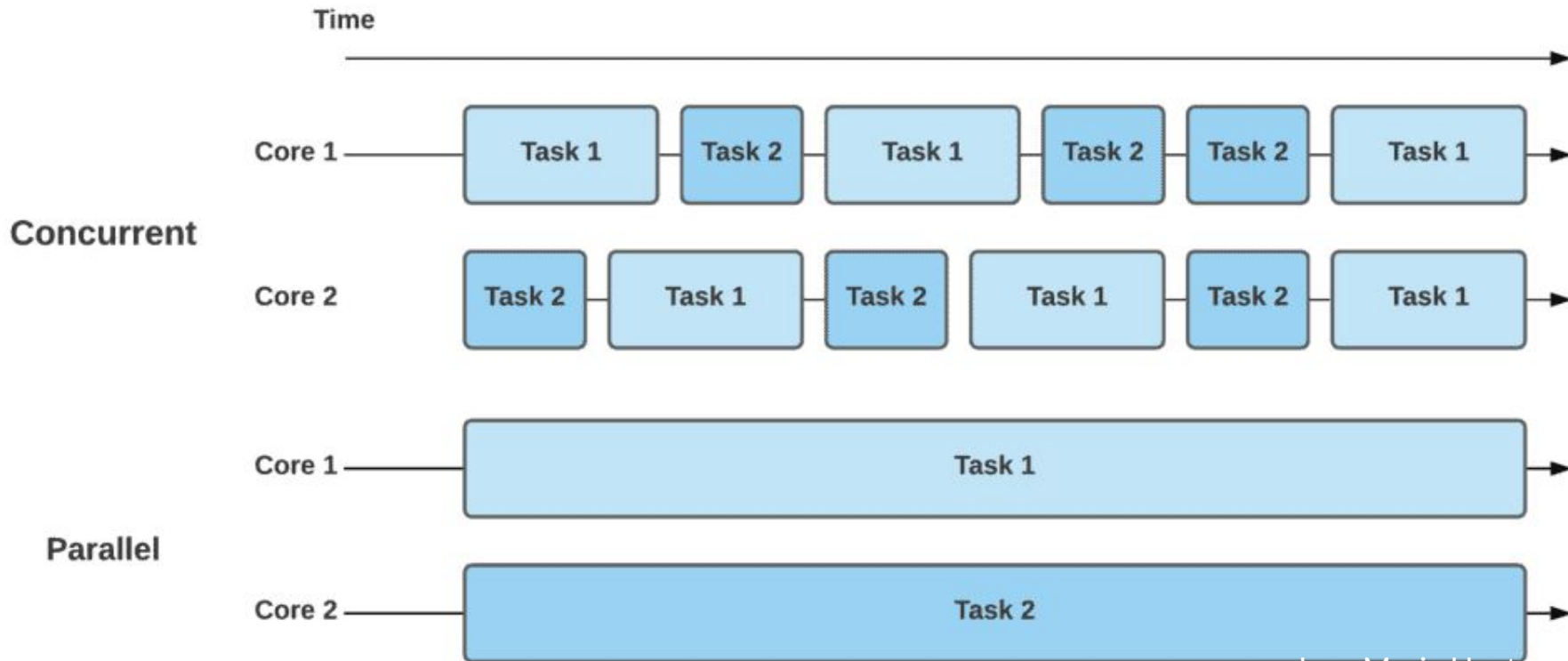
PARALELISMO

El paralelismo es la ejecución simultánea de tareas en varios núcleos o procesadores, mejorando la eficiencia y reduciendo el tiempo de ejecución. Es común en aplicaciones de alto cómputo, como simulaciones y entrenamiento de modelos de machine learning.

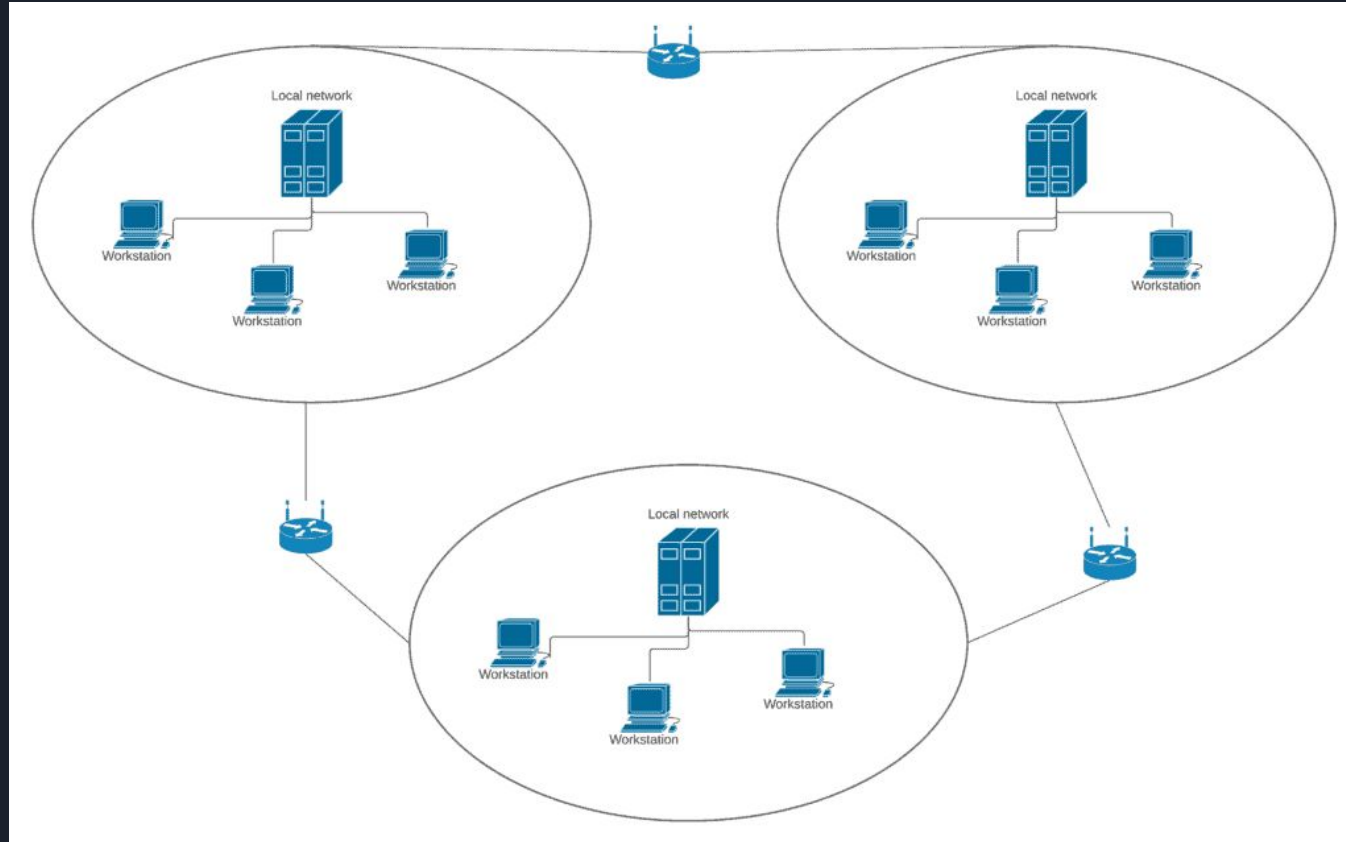
En lugar de procesar las tareas de forma secuencial, se dividen y ejecutan al mismo tiempo en diferentes unidades de procesamiento, lo que mejora la eficiencia y reduce el tiempo de ejecución.



PARALELISMO vs CONCURRENCIA



COMPUTACIÓN DISTRIBUIDA





PARALELISMO DE LOS DATOS

El modelo es replicado en cada dispositivo o nodo del sistema, y los datos se dividen en diferentes subconjuntos que son procesados en paralelo. Es particularmente útil cuando se tiene un modelo que cabe completamente en la memoria de cada dispositivo y se quiere procesar grandes conjuntos de datos de manera eficiente.

- **Ejemplo:** Supón que tienes un dataset grande y 4 GPUs. Divides el dataset en 4 partes iguales, y cada GPU entrena el modelo completo con su parte de los datos. Al final de cada paso, se combinan las actualizaciones de los pesos del modelo desde todas las GPUs.

Ventajas:

- Aumenta la velocidad de procesamiento de los datos.

- Escala bien con grandes conjuntos de datos.

Desventajas:

- No es efectivo si el modelo es demasiado grande para caber en una sola GPU.



PARALELISMO DEL MODELO

Este enfoque se utiliza cuando el **modelo es demasiado grande** para caber en la memoria de un solo dispositivo. En lugar de replicar el modelo completo en cada dispositivo, el modelo se divide entre múltiples dispositivos, y cada uno procesa una parte diferente del modelo.

- **Ejemplo:** Si tienes un modelo de red neuronal profundo muy grande, puedes dividir las capas del modelo entre varias GPUs, de modo que las capas inferiores se procesen en una GPU y las superiores en otras.

Ventajas:

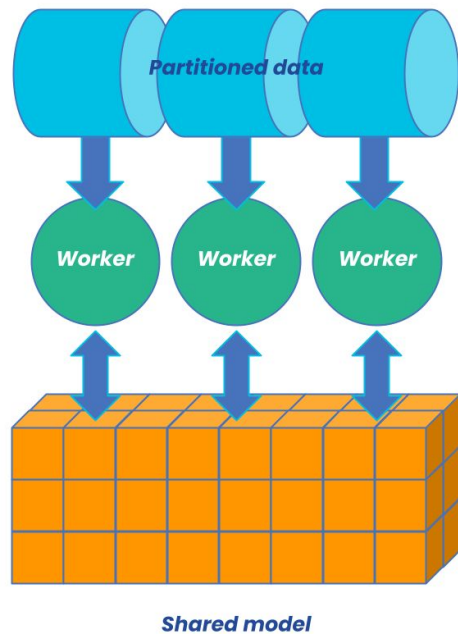
Permite entrenar modelos que son más grandes que la capacidad de memoria de una sola GPU.

Desventajas:

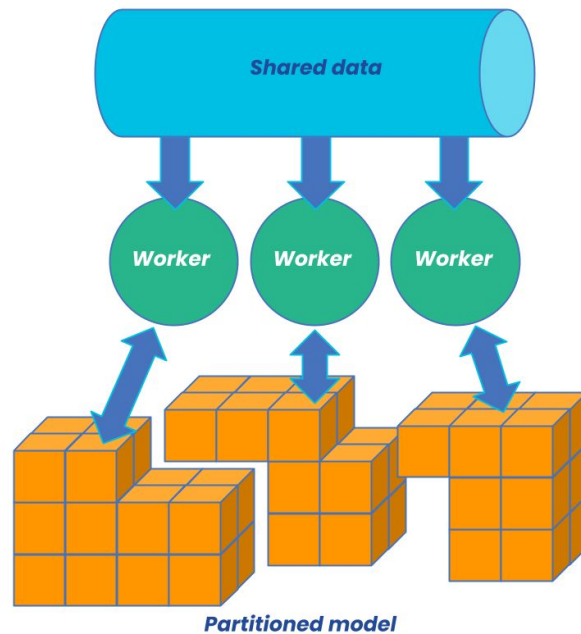
La comunicación entre dispositivos puede ser costosa y reducir la eficiencia.


Más complicado de implementar que el paralelismo de datos.

Data parallelism



Model parallelism





Introducción a las Arquitecturas Distribuidas

Juan Mario Haut