

Tarea 2. Aspectos básicos

SISTEMAS CONCURRENTES Y DISTRIBUIDOS

Ignacio David Vázquez Pérez

218292866

Tarea 2. Aspectos básicos

Objetivo

Conocer los aspectos básicos que involucran al sistema concurrente y distribuido.

Introducción:

En un mundo cada vez más interconectado y orientado a la tecnología, los sistemas concurrentes y distribuidos se han convertido en una parte fundamental de la informática moderna. Estos sistemas permiten la ejecución simultánea de múltiples tareas y la cooperación entre dispositivos y computadoras dispersas geográficamente, brindando una mayor flexibilidad y eficiencia en el procesamiento de datos y la gestión de recursos.

En esta tarea, nos adentraremos en el fascinante mundo de los sistemas concurrentes y distribuidos, explorando sus aspectos fundamentales y comprendiendo cómo se diseñan y operan. Descubriremos cómo estos sistemas abordan desafíos clave, como la sincronización de procesos, la comunicación entre sistemas, la escalabilidad y la tolerancia a fallos.

Actividades a realizar

A mitad de la década de 1980 los avances tecnológicos cambiaron y revolucionaron la era de las computadoras con dos grandes desarrollos.

- Desarrollo de microprocesadores
- Desarrollo de redes

Donde las máquinas son autónomas pero los usuarios perciben al sistema como una única computadora. Investiga los términos Sistema Centralizado y Sistema descentralizado y completa la siguiente tabla:

| | Definición | Características | Ventajas | Desventajas |
|--|------------|-----------------|----------|-------------|
|--|------------|-----------------|----------|-------------|

| | | | | |
|---------------------------------------|---|--|---|--|
| <p>Sistema Centralizado</p> | <p>Es aquel en el que todas las funciones y recursos de gestión están controlados por un único núcleo o kernel. Es un sistema en el que no hay distribución de tareas ni procesos autónomos para administrar hardware y software. Son adecuados para sistemas más simples y de menor escala, pero pueden ser menos apropiados para entornos complejos y de alta disponibilidad donde se requiere tolerancia a fallos y escalabilidad.</p> | <p>Monolítico: El sistema operativo centralizado se compone de un solo núcleo que gestiona todos los aspectos del sistema.</p> <p>Control Absoluto: El kernel tiene control total sobre recursos y ejecución de programas.</p> <p>Gestión Centralizada: Todas las decisiones y operaciones del sistema pasan por el núcleo central.</p> <p>Simplicidad: Suele ser más sencillo de diseñar y mantener debido a su estructura única.</p> | <p>Sencillez: Son más simples de implementar y comprender.</p> <p>Eficiencia: Pueden ser eficientes para sistemas de menor escala.</p> <p>Mayor Control: Proporcionan un control centralizado sobre recursos y políticas.</p> | <p>Fallos Críticos: Un fallo en el núcleo central puede llevar a la paralización de todo el sistema.</p> <p>Escalabilidad Limitada: Suelen tener dificultades para adaptarse a sistemas más grandes y complejos.</p> <p>Menos Tolerancia a Fallos: La falta de redundancia puede hacerlos menos resistentes a fallos.</p> |
| <p>Sistema Descentralizado</p> | <p>Es aquel en el que la gestión de recursos y tareas se distribuye entre múltiples núcleos o componentes, en lugar de depender de un único kernel central. Esto implica la existencia de múltiples nodos o puntos de control en el sistema. Son ideales para entornos en los que se necesita una mayor tolerancia a fallos, escalabilidad y rendimiento, pero pueden ser más complejos de implementar y gestionar.</p> | <p>Distribución de Tareas: Las funciones de gestión se dividen entre varios componentes o nodos.</p> <p>Independencia: Los nodos pueden funcionar de manera autónoma y tomar decisiones locales.</p> <p>Mayor Redundancia: Puede haber redundancia para mejorar la tolerancia a fallos.</p> <p>Escalabilidad: Suelen ser más escalables y adaptables a sistemas más grandes.</p> | <p>Tolerancia a Fallos: Mayor resistencia a fallos, ya que un fallo en un nodo no afecta necesariamente a todo el sistema.</p> <p>Escalabilidad: Son adecuados para sistemas grandes y complejos que requieren crecimiento.</p> <p>Rendimiento: La distribución de tareas puede mejorar el rendimiento al aprovechar múltiples recursos.</p> | <p>Mayor Complejidad: La gestión distribuida puede ser más compleja de diseñar y mantener.</p> <p>Coordinación Requerida: Se necesita una coordinación adecuada entre los nodos para evitar conflictos.</p> <p>Mayor Consumo de Recursos: La gestión distribuida puede requerir más recursos de hardware.</p> |

| | | | | |
|----------------------------|--|--|---|--|
| Sistema distribuido | <p>Es un tipo de sistema en el que múltiples computadoras o nodos independientes trabajan juntos como una única entidad coherente. Estos sistemas permiten la ejecución de tareas y la gestión de recursos distribuidos geográficamente a través de una red de comunicación.</p> | <p>Distribución Geográfica: Los recursos y procesos se encuentran en ubicaciones físicas separadas.</p> <p>Comunicación en Red: Los nodos se comunican a través de redes de datos.</p> <p>Transparencia: Los usuarios pueden acceder a recursos de manera transparente sin conocer su ubicación física.</p> <p>Alta Confiabilidad: Diseñados para tolerancia a fallos y redundancia.</p> | <p>Escalabilidad: Pueden crecer fácilmente para manejar grandes volúmenes de datos o usuarios.</p> <p>Alta Disponibilidad: La redundancia y tolerancia a fallos mejoran la disponibilidad.</p> <p>Eficiencia de Recursos: La distribución de tareas puede aprovechar mejor los recursos.</p> <p>Acceso Remoto: Permite el acceso y la colaboración desde ubicaciones remotas.</p> | <p>Complejidad: Son más complejos de diseñar, implementar y mantener.</p> <p>Problemas de Coordinación: La sincronización y coordinación pueden ser desafiantes.</p> <p>Seguridad: Requieren medidas de seguridad sólidas debido a la comunicación en red.</p> <p>Latencia de Red: La latencia de la red puede afectar el rendimiento.</p> |
|----------------------------|--|--|---|--|

Taxonomía de Flynn

La taxonomía de Flynn es un modelo de clasificación de las arquitecturas de computadoras y sistemas de procesamiento basado en el número de instrucciones concurrentes (control) y flujos de datos. Fue propuesta por Michael J. Flynn en 1966 y se divide en cuatro categorías principales, que se definen a continuación:

1. SISD (Single Instruction, Single Data):

- **Definición:** En esta categoría, solo se ejecuta una instrucción a la vez y opera en un solo flujo de datos.
- **Características:** Es el modelo más simple y se encuentra en sistemas uniprosesores convencionales.
- **Ejemplo:** Una computadora personal típica, donde se ejecuta una sola instrucción a la vez en una única unidad de procesamiento.

2. SIMD (Single Instruction, Multiple Data):

- **Definición:** En esta categoría, una sola instrucción se ejecuta en múltiples conjuntos de datos simultáneamente.
- **Características:** Adecuado para aplicaciones de procesamiento en paralelo, como procesamiento de imágenes y gráficos.
- **Ejemplo:** Un procesador gráfico (GPU) que realiza la misma operación en múltiples píxeles simultáneamente.

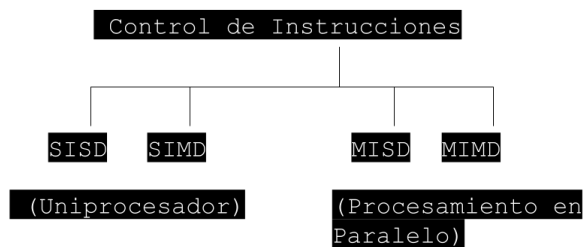
3. MISD (Multiple Instruction, Single Data):

- **Definición:** En esta categoría, varios flujos de instrucciones actúan sobre un solo flujo de datos.
- **Características:** Es menos común y se utiliza en aplicaciones específicas, como sistemas de detección de errores.
- **Ejemplo:** En sistemas de redundancia de hardware, donde se utilizan múltiples instrucciones para verificar un flujo de datos único.

4. MIMD (Multiple Instruction, Multiple Data):

- **Definición:** En esta categoría, múltiples instrucciones se ejecutan en múltiples flujos de datos de manera independiente.
- **Características:** Es el modelo más flexible y se encuentra en sistemas multiprosesores o clusters de computadoras.

- **Ejemplo:** Un clúster de servidores web en el que cada servidor ejecuta instrucciones de manera independiente en sus propios conjuntos de datos.



Nivel de acoplamiento del hardware

Se refiere a la forma en que los componentes hardware de un sistema están conectados entre sí y cómo se comunican.

1. Sistema Débilmente Acoplado (Loosely Coupled System):

- **Definición:** En un sistema débilmente acoplado, los componentes hardware o subsistemas operan de manera bastante independiente y tienen una comunicación limitada entre ellos. Cada componente puede funcionar por sí mismo y no depende en gran medida de los otros.
- **Características:** Este tipo de sistema se basa en la independencia entre los componentes y se utiliza en situaciones donde la comunicación entre ellos es mínima o no crítica.
- **Ejemplo:** Un clúster de servidores web donde cada servidor opera de forma independiente y atiende solicitudes de manera independiente. Los servidores no necesitan una comunicación constante y directa entre sí para funcionar.

2. Sistema Fuertemente Acoplado (Tightly Coupled System):

- **Definición:** En un sistema fuertemente acoplado, los componentes hardware o subsistemas están altamente interconectados y tienen una comunicación constante y directa entre ellos. Estos componentes dependen estrechamente unos de otros para funcionar de manera eficiente.
- **Características:** Este tipo de sistema se basa en la comunicación constante y la cooperación estrecha entre los componentes y se utiliza en situaciones donde la sincronización y la cooperación son esenciales.
- **Ejemplo:** Un supercomputador donde múltiples procesadores trabajan en conjunto para resolver problemas complejos. En este caso, los procesadores están altamente acoplados y se comunican intensivamente para dividir y resolver tareas.

Un sistema débilmente acoplado tiene componentes que funcionan de manera independiente con una comunicación limitada, mientras que un sistema fuertemente acoplado tiene componentes altamente interconectados que cooperan estrechamente y se comunican constantemente. La elección entre estos enfoques depende de los requisitos específicos de la aplicación y las necesidades de rendimiento del sistema.

Estos niveles de acoplamiento dan pie a obtener diferentes formas de organizar los microprocesadores y las multicomputadoras;

| | | | Dibujo | Definición y Características | Formula |
|--|--|--|--------|------------------------------|---------|

| | | | | | |
|-------------------|-------------------------------|--------------------------|---|---|---|
| | | | | | |
| Multiprocesadores | Multiprocesadores en base bus | Con base a bus sin cache |  | <p>Un microprocesador sin caché es una unidad de procesamiento central que carece de memoria caché incorporada de nivel 1 (L1). La caché suele ser una memoria pequeña pero rápida que almacena datos y/o instrucciones comúnmente utilizados para acelerar el acceso a la memoria principal.</p> <p>1. Sin Caché L1: La característica distintiva es que no tiene memoria caché L1 incorporada, lo que significa que todas las operaciones de lectura/escritura se realizan directamente en la memoria principal.</p> <p>2. Acceso a Memoria Principal: Cada vez que se necesita un dato o una instrucción, el microprocesador accede a la memoria principal, lo que puede ser más lento en comparación con la caché.</p> <p>3. Menos Costoso: Estos microprocesadores suelen ser más económicos de fabricar, ya que no incluyen la caché L1, lo que reduce el costo de producción.</p> | <p>Rendimiento = (Instrucciones por ciclo) x (Ciclos por segundo) x (Número de núcleos)</p> |

| | | | | | |
|--|--|--|--|---|---|
| | | | | <p>Un microprocesador con caché es una unidad de procesamiento central que incluye una memoria caché incorporada de nivel 1 (L1). La caché L1 es una memoria de alta velocidad que almacena datos e instrucciones comúnmente utilizados para acelerar el acceso a la memoria principal.</p> <p>1. Caché L1: La característica distintiva es la presencia de una memoria caché L1 incorporada en el microprocesador. Esta caché almacena datos e instrucciones para un acceso rápido y reducir la latencia de memoria.</p> <p>2. Mejora del Rendimiento: La caché L1 mejora significativamente el rendimiento al permitir que el microprocesador acceda a datos e instrucciones comunes sin tener que acceder a la memoria principal más lenta.</p> <p>3. Mayor Costo de Producción: Debido a la inclusión de la caché, estos microprocesadores tienden a ser más costosos de fabricar en comparación con los que carecen de caché.</p> | <p>Rendimiento = (Instrucciones por ciclo) x (Ciclos por segundo) x (Número de núcleos)</p> |
|--|--|--|--|---|---|

| | | | | | |
|--|--|--|--|---|---|
| | | | | <p>Los multiprocesadores con conmutador de punto de cruce son sistemas de computación paralela en los que varios microprocesadores están interconectados mediante un conmutador central que facilita la comunicación y el intercambio de datos entre los microprocesadores. Este tipo de arquitectura se utiliza para mejorar el rendimiento y la escalabilidad de sistemas computacionales.</p> <p>Interconexión Central: La característica distintiva es la presencia de un conmutador de punto de cruce que actúa como el centro de comunicación entre los microprocesadores. Esto permite un alto grado de flexibilidad en la comunicación entre los nodos.</p> <p>Paralelismo: Estos sistemas están diseñados para ejecutar múltiples tareas en paralelo, aprovechando la capacidad de varios microprocesadores para realizar cálculos simultáneos.</p> <p>Escalabilidad: Pueden escalarse fácilmente agregando más microprocesadores al sistema, lo que permite adaptarse a las necesidades cambiantes de rendimiento.</p> | <p>Rendimiento = (Instrucciones por ciclo) x (Ciclos por segundo) x (Número de núcleos)</p> <p>Esta fórmula es similar a la de un microprocesador individual, pero en un multiprocesador, el número de núcleos se refiere al total de microprocesadores interconectados. El conmutador de punto de cruce mejora la comunicación entre los núcleos, lo que puede llevar a un rendimiento global aún mayor.</p> |
|--|--|--|--|---|---|

| | | | | | |
|--|--|--|--|--|--|
| | | | | <p>Los multiprocesadores con red omega son sistemas de computación paralela en los que varios microprocesadores están organizados en una topología que se asemeja a la letra omega griega (Ω). Esta estructura de red permite una comunicación eficiente entre los microprocesadores, lo que mejora el rendimiento y la capacidad de procesamiento en paralelo.</p> <p>Topología Omega (Ω): La característica distintiva es la disposición de los microprocesadores en una topología que se asemeja a la letra omega griega (Ω). Esto crea una estructura de interconexión específica que facilita la comunicación entre los nodos.</p> <p>Paralelismo Eficiente: La estructura de la red omega está diseñada para facilitar la comunicación eficiente entre los microprocesadores, lo que permite una ejecución más rápida y eficaz de tareas en paralelo.</p> <p>Escalabilidad: Al igual que otros sistemas multiprocesadores, estos sistemas pueden escalarse agregando más microprocesadores a la red omega, lo que proporciona mayor capacidad de procesamiento.</p> | <p>Rendimiento = (Instrucciones por ciclo) x (Ciclos por segundo) x (Número de núcleos)</p> <p>En un multiprocesador con red omega, el número de núcleos se refiere al total de microprocesadores en la estructura de la red. La eficiente comunicación entre estos núcleos, facilitada por la topología omega, contribuye a un mayor rendimiento en aplicaciones paralelas.</p> |
|--|--|--|--|--|--|

| | | | | | |
|-------------------|-------------------------------|----------------|--|---|--|
| | | | | | |
| Multicomputadoras | Multicomputadoras en base bus | Con base a bus | | <p>Las multicomputadoras en base bus son sistemas de cómputo paralelo en los que varios microprocesadores están conectados a través de un bus de datos compartido. Cada microprocesador tiene acceso al bus y puede comunicarse con otros microprocesadores mediante la transferencia de datos a través del bus común.</p> <p>Bus Compartido: La característica principal es la presencia de un bus de datos compartido al que están conectados todos los microprocesadores. Este bus actúa como un medio de comunicación y transferencia de datos entre los nodos de procesadores.</p> <p>Interconexión Simple: La interconexión se simplifica al utilizar un único bus de datos compartido, lo que facilita la comunicación entre los nodos.</p> <p>Facilita la Programación Paralela: Estos sistemas son adecuados para aplicaciones de programación paralela donde los microprocesadores pueden trabajar juntos en tareas específicas.</p> | <p>Rendimiento = (Instrucciones por ciclo) x (Ciclos por segundo) x (Número de núcleos)</p> <p>En este caso, el número de núcleos se refiere al total de microprocesadores conectados al bus compartido. La comunicación eficiente a través del bus influye en el rendimiento general del sistema.</p> |

| | | | | | |
|--|--|--|--|--|---|
| | | | | <p>Las multicomputadoras con conmutador de retícula son sistemas de cómputo paralelo en los que varios microprocesadores están organizados en una retícula o matriz bidimensional. Estos sistemas permiten una comunicación eficiente entre los microprocesadores al conectarlos en una estructura de red específica.</p> <p>1. Topología de Retícula: La característica principal es la disposición de los microprocesadores en una retícula bidimensional. Esto facilita la comunicación entre los nodos adyacentes en filas y columnas.</p> <p>2. Comunicación Estructurada: La topología de retícula permite una comunicación estructurada y eficiente entre los microprocesadores, lo que es beneficioso para aplicaciones paralelas.</p> <p>3. Escalabilidad: Pueden escalarse agregando más microprocesadores a la retícula, lo que proporciona mayor capacidad de procesamiento y comunicación.</p> | <p>Rendimiento = (Instrucciones por ciclo) x (Ciclos por segundo) x (Número de núcleos)</p> <p>En este caso, el número de núcleos se refiere al total de microprocesadores organizados en la retícula bidimensional. La comunicación eficiente entre los nodos adyacentes en la retícula contribuye al rendimiento general del sistema.</p> |
|--|--|--|--|--|---|

- Son adecuados para aplicaciones que requieren comunicación simple entre los núcleos.
- La eficiencia puede verse afectada por cuellos de botella en el bus compartido.

2. Multiprocesadores con Conmutador:

- Los sistemas multiprocesadores con conmutador permiten una comunicación más eficiente y flexible entre los microprocesadores.
- Son ideales para aplicaciones que requieren alta velocidad de comunicación y escalabilidad.
- Pueden tener un costo más alto debido a la complejidad del conmutador.

3. Multicomputadoras en Base a Bus:

- Estas configuraciones utilizan un bus de datos compartido para conectar múltiples sistemas independientes.
- Son adecuadas para aplicaciones que requieren sistemas separados pero con capacidad de comunicación ocasional.
- La comunicación entre sistemas puede ser menos eficiente que en los multiprocesadores.

4. Multicomputadoras con Conmutador:

- Las multicomputadoras con conmutador permiten la comunicación eficiente entre sistemas independientes a través de un conmutador central.
- Son ideales para aplicaciones que requieren una comunicación rápida y constante entre sistemas.
- Pueden ser altamente escalables y ofrecer un rendimiento excepcional.

En general, la elección de la arquitectura depende de las necesidades específicas de la aplicación y los recursos disponibles. Cada una de estas configuraciones tiene ventajas y desventajas, y la selección adecuada se basa en consideraciones como el rendimiento, la escalabilidad, la complejidad y el presupuesto.

En un mundo cada vez más orientado hacia la computación paralela y distribuida, comprender estas arquitecturas es esencial para diseñar sistemas de alto rendimiento que puedan abordar una variedad de aplicaciones y desafíos computacionales.

Bibliografía

- Tanenbaum Andrew. (1995). Sistemas Operativos Distribuidos. España. Prentice-Hall Hisp.
- McIver Ann. (2011). Sistemas Operativos. México. Cengage Learning.
- Tanenbaum, A., & Van Steen M. (2008). Sistemas Distribuidos, Principios y Paradigmas. (Segunda ed.). Prentice Hall.
- Tanenbaum, A. (2011). Redes de Computadoras. (Quinta ed.). Prentice Hall.
- Elmasri, R., Gil Carrick, A., & Levine, D. (2010). Sistemas Operativos, Un enfoque en espiral. McGraw-Hill.