

# Cuestionario teoría 7 a 11

lunes, 15 de febrero de 2021 20:20

1- Defina y diferencie programa concurrente, programa distribuido y programa paralelo.

Un programa concurrente especifica dos o más “programas secuenciales” que pueden ejecutarse concurrentemente en el tiempo como tareas o procesos.

Un programa paralelo conciste en la ejecución concurrente en múltiples procesadores con el objetivo principal de reducir el tiempo de ejecución.

Programa distribuido: programa concurrente comunicado por mensajes. Supone la ejecución sobre una arquitectura de memoria distribuida, aunque puedan ejecutarse sobre una de memoria compartida (o híbrida).

2- Marque al menos 2 similitudes y 2 diferencias entre los pasajes de mensajes sincrónicos y asincrónicos.

Similitudes:

Receive es una primitiva bloqueante en ambos casos. Otra similitud es que ambos son mecanismos para el Procesamiento Distribuido

Diferencia: con la instrucción send, en PMS, el sender se queda esperando la recepción del mensaje, en cambio en PMA no. Otra diferencia es que en PMS los canales son de tipo link o punto a punto, en cambio en PMA puede haber otros tipos de canales.

3- Analice qué tipo de mecanismos de pasaje de mensajes son más adecuados para resolver problemas de tipo Cliente/Servidor, Pares que interactúan, Filtros, y Productores y Consumidores. Justifique claramente su respuesta.

Cliente/Servidor: RPC o Rendezvous, ya que tiene canales bidireccionales.

Pares que interactúan: PMA, ya que cada nodo va a estar conectado con otro nodo al que le va a enviar datos de manera asincrónica.

Filtros: PMA, ya que hay un camino de canales unidireccionales por los que va a pasar el dato que enviamos.

Consumidores: PMA, ya que la comunicación entre productores y consumidores suele ser unidireccional.

4- Defina el concepto de “continuidad conversacional” entre procesos.

Cuando un proceso requiere comunicarle varios mensajes a otro proceso de manera ininterrumpida, primero establece una comunicación por un canal, para especificarle al proceso receptor que va a enviar varios mensajes, y luego envía los mensajes por otro canal, hasta enviar uno con una marca especial que define el fin de la comunicación.

5- Indique por qué puede considerarse que existe una dualidad entre los mecanismos de monitores y pasaje de mensajes. Ejemplifique

Se puede identificar una dualidad cuando a un monitor se le hace una Call, se podría considerar una similitud la llamada a un proceso del monitor con un mensaje enviado de un proceso a otro.

6- ¿En qué consiste la comunicación guardada (introducida por CSP) y cuál es su utilidad? Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen comunicaciones guardadas

La comunicación guardada consiste en sentencias del estilo  $B;C \rightarrow S$ , en donde B es una condición booleana, C una comunicación (B y C forman la guarda) y S un conjunto de sentencias que se ejecutará cuando B sea verdadera y la comunicación se puede efectuar.

La utilidad de este tipo de comunicación es el soporte del no determinismo.

7- Marque similitudes y diferencias entre los mecanismos RPC y Rendezvous.

Diferencias:

La ejecución de un procedimiento en RPC se hace en un servidor creado en ese momento con el

único propósito de resolver el procedimiento, en cambio, con Rendezvous, la ejecución de un procedimiento es realizada por un servidor que está siempre activo a la espera de que los procesos le indiquen que procedimientos ejecutar.  
RPC por sí solo es solo un método de comunicación, no posee sincronización.

Similitudes:

Ambos usan canales bidireccionales por los que proveen comunicación.

Ambos combinan una interfaz “tipo monitor” con operaciones exportadas a través de llamadas externas (CALL) con mensajes sincrónicos.

8- Describa el paradigma “bag of tasks”. ¿Cuáles son las principales ventajas del mismo?

El concepto de bag of tasks supone que un conjunto de workers comparten una “bolsa” con tareas independientes. Los workers sacan una tarea de la bolsa, la ejecutan, y posiblemente crean nuevas tareas que ponen en la bolsa

La mayor virtud de este enfoque es la escalabilidad y la facilidad para equilibrar la carga de trabajo de los workers.

9- Describa sintéticamente las características de sincronización y comunicación de MPI

MPI define una librería estándar que puede ser empleada desde C o Fortran (y potencialmente desde otros lenguajes).

El estándar MPI define la sintaxis y la semántica de más de 125 rutinas.

Hay implementaciones de MPI de la mayoría de los proveedores de hardware.

Usa un modelo SPMD.

Todas las rutinas, tipos de datos y constantes en MPI tienen el prefijo “MPI\_”. El código de retorno para operaciones terminadas exitosamente es MPI\_SUCCESS.

Básicamente con 6 rutinas podemos escribir programas paralelos basados en pasaje de mensajes:

MPI\_Init, MPI\_Finalize, MPI\_Comm\_size, MPI\_Comm\_rank, MPI\_Send y MPI\_Recv

10- Describa sintéticamente las características de sincronización y comunicación de Java y ADA.

jajaxd

Java usa un enfoque el cual es declarar un procedure para cada operación y crear un nuevo proceso (al menos conceptualmente) para manejar cada llamado (RPC porque el llamador y el cuerpo del procedure pueden estar en distintas máquinas). Para el cliente, durante la ejecución del servicio, es como si tuviera en su sitio el proceso remoto que lo sirve.

Ada usa rendezvous con un proceso existente. Un rendezvous es servido por una sentencia de Entrada (o accept) que espera una invocación, la procesa y devuelve los resultados.

13- Suponga que N procesos poseen inicialmente cada uno un valor. Se debe calcular la suma de todos los valores y al finalizar la computación todos deben conocer dicha suma.

Analice (desde el punto de vista del número de mensajes y la performance global) las soluciones posibles con memoria distribuida para arquitecturas en estrella (centralizada), anillo circular, totalmente conectada, árbol y grilla bidimensional.

jajaaaaaxd

14- Explique brevemente los 7 paradigmas de interacción entre procesos en programación distribuida vistos en teoría. En cada caso ejemplifique, indique qué tipo de comunicación por mensajes es más conveniente y qué arquitectura de hardware se ajusta mejor. Justifique sus respuestas

jaaaaaaaaaaaaaaaaaaaaaaaaa ja ja xd

16- Investigue el problema de detección de terminación en programación distribuida. Plantee soluciones para el caso de procesos conectados en anillo y en grafos.

La solución para procesos conectados en grafos son los semáforos distribuidos usando broadcast

La solución para procesos conectados en anillo puede ser token ring.

17- ¿En qué consiste la utilización de relojes lógicos para resolver problemas de sincronización distribuida? Ejemplifique para el caso de implementación de semáforos distribuidos.

jajaxd

Se puede usar broadcast para diseminar información o para resolver problemas de sincronización distribuida. Ejemplo: semáforos distribuidos, la base es un ordenamiento total de eventos de comunicación mediante el uso de relojes lógicos.

19- ¿Qué relación encuentra entre el paralelismo recursivo y la estrategia de “dividir y conquistar”? ¿Cómo aplicaría este concepto a un problema de ordenación de un arreglo?

La relación es que ambas técnicas dividen los problemas y los datos en porciones más pequeñas para sub-dividir las ejecuciones e incrementar la eficiencia.

Para la ordenación de un arreglo se pueden hacer subdivisiones del mismo, repartiendo cada segmento de datos en cada proceso, y al terminar el ordenamiento de 2 segmentos se unen y se los vuelve a ordenar, hasta obtener el vector completo ordenado.

20)

a) Cómo puede influir la topología de conexión de los procesadores en el diseño de aplicaciones concurrentes/paralelas/distribuidas? Ejemplifique.

La topología de la conexión tiene una influencia directa en el diseño de las aplicaciones, si tenemos una topología en anillo por ejemplo, la comunicación podría ser por PMA, en cambio, si tenemos una topología de forma descentralizada, lo más probable es que el nodo central tenga una comunicación bidireccional con cada nodo, por lo que podría utilizar Rendezvous o RPC.

b) Qué relación existe entre la granularidad de la arquitectura y la de las aplicaciones?

La granularidad de una aplicación está dada por la relación entre el cómputo y la comunicación, la misma se adapta a la arquitectura que utiliza, hay arquitecturas que requieren una mayor comunicación.

21)

a) ¿Cuál es el objetivo de la programación paralela?

Dividir las tareas de una aplicación de la manera más eficiente posible para distribuirlas en los procesadores y lograr una mayor eficiencia.

b) ¿Cuál es el significado de las métricas de speedup y eficiencia? ¿Cuáles son los rangos de valores en cada caso? Speedup: es el cociente entre el tiempo de ejecución del algoritmo serial conocido más rápido y el tiempo de ejecución paralelo del algoritmo elegido. El rango en general va entre 0 y el speedup óptimo

Eficiencia: cociente entre speedup y speedup óptimo. Mide la fracción de tiempo en que los procesadores son útiles para el cómputo.

El valor es entre 0 y 1.

c) ¿En qué consiste la “ley de Amdahl”?

La ley de Amdahl mide el porcentaje de código secuencial (ponele) y es utilizada para averiguar la mejora máxima de un sistema de información cuando solo una parte de éste es mejorado.

d) Suponga que la solución a un problema es paralelizada sobre p procesadores de dos maneras diferentes. En un caso, el speedup (S) está regido por la función  $S=p-1$  y en el otro por la función  $S=p/2$ . ¿Cuál de las dos soluciones se comportará más eficientemente al crecer la cantidad de procesadores? Justifique claramente.

jaja son re jodones para tomar parciales :)

e) Suponga que el tiempo de ejecución de un algoritmo secuencial es de 10000 unidades de tiempo, de las cuales sólo el 90% corresponde a código paralelizable. ¿Cuál es el límite en la mejora que puede obtenerse paralelizando el algoritmo? Justifique.

El límite de mejora es 1000, por qué te preguntaras. Bueno te explico. Ya sabemos que 1000 unidades no son paralelizables, por lo que esas 1000 no las vés a poder mejorar, como mucho, si tenés 9000 procesadores más, podés dividir las 9000 unidades de tiempo en 1 para cada procesador, siendo el tiempo de ejecución total 1000, siendo el procesador 9001 el que ejecuta las 1000.

jaaaaaaaaaajaxd asxdjas

