

Recopilación Finales

Septiembre 2014

1.

Describa brevemente en qué consisten los mecanismos de RPC y Rendezvous. Para qué tipo de problemas son más adecuados?

2.

Hacer una sección crítica con variables compartidas

3.

¿Qué elementos de la forma general de rendezvous no se encuentran en el lenguaje ADA?

4.

Dados los siguientes dos segmentos de código, indicar para cada ítem si son equivalentes o no. Justificar en cada caso (dar ejemplos si es necesario).

Segmento 1	Segmento 2
<pre>... int cant=1000; DO (cant < -10); datos?(cant) → Sentencias1 □ (cant > 10); datos?(cant) → Sentencias2 □ (INCÓGNITA); datos?(cant) → Sentencias3 END DO ...</pre>	<pre>... int cant=1000; While (true) { IF (cant < -10); datos?(cant) → Sentencias1 □ (cant > 10); datos?(cant) → Sentencias2 □ (INCÓGNITA); datos?(cant) → Sentencias3 END IF } ...</pre>

- a) INCÓGNITA equivale a: (cant = 0),
- b) INCÓGNITA equivale a: (cant > -100)
- c) INCÓGNITA equivale a: ((cant > 0) or (cant < 0))
- d) INCÓGNITA equivale a: ((cant > -10) or (cant < 10))
- e) INCÓGNITA equivale a: ((cant >= -10) or (cant <= 10))

5.

Sea la siguiente solución al problema del producto de matrices de $n \times n$ con P procesos trabajando en paralelo.

```
process worker[w = 1 to P] { # strips en paralelo (p strips de n/P
  filas) }
int first = (w-1) * n/P; # Primera fila del strip
int last = first + n/P - 1; # Ultima fila del strip
for [i = first to last] {
  for [j = 0 to n-1] {
    c[i,j] = 0.0;
    for [k = 0 to n-1]
      c[i,j] = c[i,j] + a[i,k]*b[k,j];
  }
}
```

- a) Suponga que $n=128$ y cada procesador es capaz de ejecutar un proceso. Cuántas asignaciones, sumas y productos se hacen secuencialmente (caso en que $P=1$)? Cuántas se realizan en cada procesador en la solución paralela con $P=8$?
- b) Si los procesadores P_1 a P_7 son iguales, y sus tiempos de asignación son 1, de suma 2 y de producto 3, y si P_8 es 4 veces más lento, Cuánto tarda el proceso total concurrente? Cuál es el valor del speedup (Tiempo secuencial/Tiempo paralelo)?. Modifique el código para lograr un mejor speedup.

Noviembre 2014

1.

- a) ¿Qué significa el problema de "interferencia" en programación concurrente? ¿Cómo puede evitarse?
- b) ¿En qué consiste la propiedad de "A lo sumo una vez"? De ejemplos de sentencias que cumplan y de sentencias que no cumplan con ASV

- c) Dado el siguiente fragmento de programa concurrente con variables compartidas:

```
int s = 6; t = 4;  
co (s = s + t ) // (s = s * t)
```

- i) ¿Cuáles son los posibles valores finales de s y t ?
- ii) Si se eliminan los corchetes angulares y las asignaciones se implementan con 3 acciones atómicas (leer, sumar o multiplicar y escribir en una variable), ¿cuáles son los valores posibles de s y t ? (Nota: no es necesario que liste TODOS los resultados, pero sí todos los casos que resulten significativos)

2.

Resuelva el problema de acceso a sección crítica usando un proceso coordinador. En este caso, cuando un proceso $SC[i]$ quiere entrar a su sección crítica, le avisa al coordinador, y espera a que éste le dé permiso. Al terminar de ejecutar su sección crítica, el proceso $SC[i]$ le avisa al coordinador

Desarrolle una solución de grano fino usando **solo** variables compartidas (no semáforos ni monitores)

3.

- a) Defina qué son los paradigmas master/worker, token passing y prueba/eco. Marque claramente ventajas y desventajas para el caso de mensajes sincrónicos y asincrónicos
- b) Implemente una solución al problema de exclusión mutua distribuida entre N procesos utilizando un algoritmo de tipo token passing con mensajes asincrónicos

4.

- a) En qué consiste la comunicación guardada y cuál es su utilidad? Ejemplifique
- b) Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen comunicaciones guardadas

- c) Dado el siguiente bloque de código, indique para cada inciso que valor queda en *aux*, o si el código quede bloqueado. Justifique sus respuestas

```
aux = -2
if
(A == 0); P2?(aux) → aux = aux + 2;
  (A == 1); P3?(aux) → aux = aux + 5;
  (B == 0) P3?(aux) → aux = aux + 7;
end if;
```

- i) Si el valor de A = 1 y B = 2 antes del IF y solo P2 envía el valor 6
- ii) Si el valor de A = 0 y B = 2 antes del IF, y solo P2 envía el valor 8
- iii) Si el valor de A = 2 y B = 0 antes del IF y solo P3 envía el valor 8
- iv) Si el valor de A = 2 y B = 1 antes del IF y solo P3 envía el valor 9
- v) Si el valor de A = 1 y B = 0 antes del IF y solo P3 envía el valor 14
- vi) Si el valor de A = 0 y B = 0 antes del IF y P3 envía el valor 9 y P2 el valor 5

5.

- a) ¿Cuál es el objetivo de la programación paralela?
- b) Defina las métricas de speedup y eficiencia. ¿Cuál es el significado de cada una de ellas (que miden)? ¿Cual es el rango de valores posibles de cada uno? Ejemplifique
- c) Defina escalabilidad de un sistema paralelo
- 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 $S=p-2$ y en el otro por la función $S=p/2$ (para $p>2$) ¿Cual de las dos soluciones se comportará más eficientemente al crecer la cantidad de procesadores?
- e) Describa conceptualmente qué dice la ley de Amdah) (no es necesaria la formula)
- f) Suponga que el tiempo de ejecución de un algoritmo secuencial es de 10000 unidades de tiempo, de las cuales solo el 90% corresponde a código paralelizable. ¿Cuál es el límite en la mejora que puede obtenerse paralelizando el algoritmo?

Diciembre 2014

1.

Defina el problema de sección crítica. Compare los algoritmos para resolver este problema (spin locks, Tie Breaker, Ticket, Bakery). Marque ventajas y desventajas de cada uno.

2.

Resuelva el problema de acceso a Sección Crítica usando un proceso coordinador. En este caso cuando proceso SC[I] quiere entrar en su sección crítica le avisa al coordinador, y espera que este le de permiso. Al Terminar de ejecutar su sección crítica el proceso SC[I] le avisa al coordinador. Desarrolle una solución de grano fino usando solo variables compartidas (no semáforos, ni monitores).

3.

Suponga que una imagen se encuentra representada por una matriz a ($n \times n$), y que el valor de cada pixel es un numero entero que es mantenido por un proceso distinto(es decir, el valor del pixel I, J , esta en el proceso $P(I, J)$). Cada proceso puede comunicarse solo con su vecinos izquierdo, derecho, arriba y abajo.(los procesos de los esquinas tienen solo 2 vecinos, y los otros bordes de la grilla tienen 3 vecinos).

- a) Escriba un algoritmo Herbeat que calcule el maximo y el minimo valor de los pixels de la imagen. Al terminar el programa, cada proceso debe conocer ambos valores.
- b) Analice la solución de desde el punto de vista del numero de mensajes.
- c) Puede realizar alguna mejora para reducir el numero de mensajes.

4.

¿Cual es el objetivo de la programación paralela?

- a) Defina las métricas de speedup y eficiencia. Cual es el significado de cada una de ellas (que miden)? Cual es el rango de valores posibles de cada una? Ejemplifique.

- b) Defina escalabilidad de un sistema paralelo.
- c) Suponga que la solución a un problema es paralelizada sobre p procesadores de dos maneras diferentes. En un caso, el speedup(5) esta regido por la funcion $S = p - 4$ y el otro por la funcion $S = p/3$ para $p > 4$. Cual de las dos soluciones se comportara mas eficientemente al crecer la cantidad de procesadores.
- d) Describa conceptualmente que dice la ley de Amdahl (no es necesaria la formula).
- e) Suponga que el tiempo de ejecución de un algoritmo secuencial es de 10000 unidades de tiempo, de las cuales 95% corresponden a código paralelizable . Cual es el límite en la mejora que puede obtenerse paralelinzado el algoritmo.

5.

Sea la siguiente solucion al problema del producto de matrices de $n \times n$ con p procesos en paralelo con variables compartidas:

```

Process worker[w=1 to p] {#strips en paralelo (p strips de n/p
filas)}
Int first= (w-1) * n/p + 1
Int last= first + n/p - 1
For [i=1 to last]
{
    For [j= 1 to n]
    {
        C[i, j] = 0.0;
        For [k= 1 to N] {
            C[I,J] = C[I,J] + a [I,K] * b[K,j];
        }
    }
}

```

- a) Suponga $n = 128$ y cada procesador capaz de ejecutar un proceso. Cuantas asignaciones, sumas y productos se hacen secuencialmente (caso en que $p=1$)? Cuantos se realizan en cada procesador en la solución paralela con $p=4$?

- b) si $p_1=p_2=\dots=p_4$ y los tiempos de asignación son 1, de suma 2 y de producto 3, y si p_4 es 4 veces mas lento, cuanto tarda el proceso total concurrente? Cual es el valor del speedup? Modifique el código para lograr un mayor speedup

Julio 2015

1.

- a) explique el concepto de bag of tasks
- b) cuales son las ventajas y desventajas

2.

- a) En que consiste la comunicación guardada ejemplifique
- b) Explique como funciona el mecanismo de comunicación guardada para sentencias de decisión e iterativas.

Despues habia ejercicio de un if con comunicación guardada y te presentaba diferentes valores y habia que decir y justificar que valor iba a tener una variable.

3.

- a) Explique que es el paradigma Heart Beat para computación distribuida.
- b) Que tipo de pasaje de mensajes es mejor para implementarlo. Justifique.
- c) Escriba un algoritmo Heart Beat para construir la topografia de una red. // Este es con el diametro de la red no conocido

4.

N procesos deben realizar la suma de un conjunto de números y todos deben conocer el resultado, cada proceso tiene uno de los numeros. Compare las soluciones utilizando arquitecturas centralizada, anillo circular, full-conected y arbol. Compare el número de mensajes enviados, performance global(o algo asi).

Escribir el código para dos de las arquitecturas

5.

- a) ¿Cuál es el objetivo de la programación paralela?
- b) ¿Qué es speedup? ¿Qué es eficiencia? ¿De cada uno cuáles son los valores posibles? Ejemplifique
- c) Explique el concepto de escalabilidad
- d) Se tienen dos algoritmos para los que la función de speedup son respectivamente $S=p-4$ y $S=p/3$ (si $p>4$) donde p es el número de procesadores ¿qué algoritmo es más escalable? justificar.
- e) Explique el concepto de la Ley de Amdahl. No es necesaria la fórmula

Diciembre 2015

1.

Defina el problema de la sección crítica. Compare los algoritmos para resolver este problema (Spinlocks, Ticket, Bakery). Marque ventajas y desventajas de cada uno.

2.

- a) ¿En qué consiste la comunicación guardada y cuál es su utilidad? Ejemplifique.
- b) Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen comunicación guardadas.
- c) Dado el siguiente bloque de código, indique para cada inciso qué valor quedó en Aux, o si el código quedó bloqueado.

```
Aux = 1;
....
if (A==0); P2?(Aux) > Aux = Aux +2;
    (A==1); P3?(Aux) > Aux = Aux +5;
    (B==0); P3?(Aux) > Aux = Aux +7;
endif;
...
```

- i. Si el valor de $A = 1$ y $B = 2$ antes del if, y solo P2 envía el valor 6.

- ii. Si el valor de $A = 0$ y $B = 2$ antes del if, y solo P2 envia el valor 8.
- iii. Si el valor de $A = 2$ y $B = 0$ antes del if, y solo P3 envia el valor 6.
- iv. Si el valor de $A = 2$ y $B = 1$ antes del if, y solo P3 envia el valor 9.
- v. Si el valor de $A = 1$ y $B = 0$ antes del if, y solo P3 envia el valor 14.
- vi. Si el valor de $A = 0$ y $B = 0$ antes del if, P3 envia el valor 9 y P2 el valor 5.

3.

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.

- a) 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 y Arbol.
- b) Escriba las soluciones de al menos dos de las arquitecturas mencionadas.

4.

Sea el problema de ordenar de menor a mayor un arreglo de $A[1..n]$

- a) Escriba un programa donde dos procesos (cada uno con $n/2$ valores) realicen la operación en paralelo mediante una serie de intercambios.
- b) ¿Cuántos mensajes intercambian en el mejor caso? ¿Y en el peor caso?
- c) Utilizando la idea de a), extienda la solución a K procesos, con n/k valores c/u ("odd even exchange sort").
- d) ¿Cuántos mensajes intercambian en c) en el mejor caso? ¿Y en el peor caso?

Nota: Utilice un mecanismo de pasaje de mensajes, justificando la elección del mismo.

5.

Suponga que una imagen se encuentra representada por una matriz $A(n \times n)$, y que el valor de cada pixel es un número entero que es mantenido por un proceso distinto (es decir, el valor del pixel i,j está en el proceso $P(i,j)$). Cada proceso puede comunicarse sólo con sus vecinos izquierdo, derecho, arriba y abajo (los proceso de

las esquinas tienen solo 2 vecinos y los otros en los borde de la grilla tienen 3 vecinos).

- a) Escriba un algoritmo HeartBeat que calcule el máximo y el mínimo valor de los pixels de la imagen. Al terminar el programa, cada proceso debe conocer ambos valores.
- b) Analice la solución desde el punto de vista del número de mensajes.
- c) ¿Puede realizar alguna mejora para reducir el número de mensajes

Abril 2016

1.

Defina el problema de la sección crítica. Compare los algoritmos para resolver este problema (Spin locks, Tie Breaker, Ticket y Bakery). Marque ventajas y desventajas de cada uno.

2.

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

- a) 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 y Árbol.
- b) Implemente al menos dos de las soluciones mencionadas.

3.

Sea el problema de ordenar de menor a mayor un arreglo de $A[1..n]$

- a) Escriba un programa donde dos procesos (cada uno con $n/2$ valores) realicen la operación en paralelo mediante una serie de intercambios.
- b) ¿Cuántos mensajes intercambian en el mejor de los casos? ¿Y en el peor de los casos?

- c) Utilice la idea de a), extienda la solución a K procesos, con n/k valores c/u (“odd-even-exchange sort”).
- d) ¿Cuántos mensajes intercambian en 3) en el mejor caso? ¿Y en el peor de los casos?

Nota: Utilice un mecanismo de pasaje de mensajes, justifique la elección del mismo.

4.

Explicar la notación de primitivas múltiples

5.

Sea la siguiente solución al problema del producto de matrices de $n \times n$ con p procesos en paralelo con variables compartidas:

```

Process worker[w=1 to p] {#strips en paralelo (p strips de n/p
filas)}
Int first= (w-1) * n/p + 1
Int last= first + n/p - 1
For [i=1 to last]{
    For [j= 1 to n] {
        C[i, j] = 0.0;
        For [k= 1 to N] {
            C[I,J] = C[I,J] + a [I,K] * b[K,j];
        }
    }
}

```

- a) Suponga $n= 128$ y cada procesador capaz de ejecutar un proceso. Cuantas asignaciones, sumas y productos se hacen secuencialmente (caso en que $p=1$)? Cuantos se realizan en cada procesador en la solución paralela con $p= 4$?
- b) si $p_1=p_2=\dots=p_4$ y los tiempos de asignación son 1, de suma 2 y de producto 3, y si p_4 es 4 veces mas lento, cuanto tarda el proceso total concurrente? Cual es el valor del speedup? Modifique el código para lograr un mayor speedup.

Marzo 2016

1.

- a) Qué significa el problema de “interferencia” en programación concurrente?
¿Cómo puede evitarse?
- b) En qué consiste la propiedad de “A lo sumo una vez”. Dar ejemplos de sentencias que cumplan y no cumplan la propiedad asv
- c) Dado el siguiente programa concurrente con memoria compartida, tenga en cuenta que las instrucciones no son atómicas:

```
x:=4; y:=4; z:=2;  
Co x:=y*z // z:=z*3 //y:=y2x Oc
```

- i) ¿Cuáles de las asignaciones dentro de la sentencia co cumplen con la propiedad de a lo sumo una vez? Justifique.
- ii) Indique los resultados posibles de la ejecución (No es necesario listarlos todos). Justifique.

2.

Defina el problema de la sección crítica. Compare los algoritmos para resolver este problema (Spin locks, Tie Breaker, Ticket y Bakery). Marque ventajas y desventajas de cada uno.

3.

¿En qué consiste la comunicación guardada y cual es su utilidad? Ejemplifique. Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen comunicación guardadas.

- Dado el siguiente bloque de código, indique para cada inciso que valor quedó en Aux, o si el código quedó bloqueado.

```
Aux = 1;  
....  
if (A==0); P2?(Aux) > Aux = Aux +2;  
(A==1); P3?(Aux) > Aux = Aux +5;
```

```
(B==0); P3?(Aux) > Aux = Aux +7;  
endif;  
...
```

- a) Si el valor de A = 1 y B = 2 antes del if, y solo P2 envia el valor 6.
- b) Si el valor de A = 0 y B = 2 antes del if, y solo P2 envia el valor 8.
- c) Si el valor de A = 2 y B = 0 antes del if, y solo P3 envia el valor 6.
- d) Si el valor de A = 2 y B = 1 antes del if, y solo P3 envia el valor 9.
- e) Si el valor de A = 1 y B = 0 antes del if, y solo P3 envia el valor 14.
- f) Si el valor de A = 0 y B = 0 antes del if, P3 envia el valor 9 y P2 el valor 5.

4.

Describe el paradigma "Bag of Tasks". Describe ventajas y desventajas. Ejemplifique.

5.

Explique sintéticamente los 7 paradigmas de interacción entre procesos en programación distribuida.

6.

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 y Arbol.

7.

- a)Cuál es el objetivo de la programación paralela? Defina las métricas de speedup y eficiencia.
- b)Cuál es el significado de cada una de ellas (qué miden)?
- c) Suponga que la solución a un problema es paralelizada sobre p procesadores de dos maneras diferentes. En un caso, el speedup esta regido por la función

$S=p/2$ y el otro por la función $S=p^2$. ¿Cual de las dos soluciones se comportara más eficientemente al crecer la cantidad de procesadores?

d) En qué consiste la “ley de Amdahl”

Agosto 2016

1.

Defina el problema de la sección crítica. Compare los algoritmos para resolver este problema (Spin locks, Tie Breaker, Ticket y Bakery). Marque ventajas y desventajas de cada uno.

2.

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

- a) 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 y Árbol.
- b) Implemente al menos dos de las soluciones mencionadas.

3.

Sea el problema de ordenar de menor a mayor un arreglo de $A[1..n]$

- a) Escriba un programa donde dos procesos (cada uno con $n/2$ valores) realicen la operación en paralelo mediante una serie de intercambios.
- b) ¿Cuántos mensajes intercambian en el mejor caso? ¿Y en el peor caso?
- c) Utilizando la idea de a), extienda la solución a K procesos, con n/k valores c/u (“odd even exchange sort”).
- d) ¿Cuántos mensajes intercambian en c) en el mejor caso? ¿Y en el peor caso?

Nota: Utilice un mecanismo de pasaje de mensajes, justificando la elección del mismo.

4.

Defina la notación de primitivas múltiples.

5.

Sea la siguiente solución al problema del producto de matrices de $n \times n$ con p procesos en paralelo con variables compartidas:

```
Process worker[w=1 to p] { #strips en paralelo (p strips de n/p
    filas)}
int first= (w-1) * n/p + 1
int last= first + n/p - 1
for [i=1 to last]{
    for [j= 1 to n] {
        C[i, j] = 0.0;
        for [k= 1 to n] {
            C[i,j] = C[i,j] + a [i,k] * b[k,j];
        }
    }
}
```

- a) Suponga $n=128$ y cada procesador capaz de ejecutar un proceso. Cuántas asignaciones, sumas y productos se hacen secuencialmente (caso en que $p=1$)? Cuántos se realizan en cada procesador en la solución paralela con $p=8$?
- b) si $p_1=p_2=\dots=p_7$ y los tiempos de asignación son 1, de suma 2 y de producto 3, y si p_8 es 4 veces más lento, cuánto tarda el proceso total concurrente? Cuál es el valor del speedup?

Modifique el código para lograr un mayor speedup

Agosto 2022

1.

relación paralelismo recursivo y divide & conquer, cómo podría esto aplicarse al ordenar un arreglo.

2.

defina heartbeat y servidores replicados, ventajas y desventajas de cada uno con pma y pms

3.

comunicación guardada defina y explique cómo funciona if y do

4.

muestra un código y decir que hace, cuántos mensajes envía cada programa y que tipo de granularidad tiene cada programa (los programas eran, 2 for que recorrían una matriz y uno mandaba mensaje por cada casilla y otra función sumaba y el otro sumaba toda la fila y luego enviaba 1 mensaje)

5.

Escribir un programa para ordenar de menor a mayor un arreglo de $a[1..n]$, con 2 procesos donde cada uno recibe $n/2$.

Cuántos mensajes se hacen en el mejor caso? Y en el peor?

Haga el mismo ejercicio pero para n/k con c/u odd-even-exchange-sort

Octubre 2022

1.

Defina programación concurrente, programación paralela y programación distribuida

2.

¿En qué consiste la propiedad “a lo sumo una vez” y cuál es el efecto que tiene sobre las sentencias de un programa concurrente?

3.

Explique comunicación guardada. Explique sentencias if y do

4.

Qué es una barrera simétrica? Defina Butterfly Barrier. Instancie un ejemplo para $N=16$ (nota: había que hacer un dibujito como el que está en la teoría, de cada etapa con el $N=16$)

5.

- a) Describa brevemente en qué consisten los mecanismos de RPC y Rendezvous. ¿Para qué tipos de problemas son más adecuados?
- b) ¿Por qué es necesario proveer sincronización dentro de los módulos de RPC? ¿Cómo puede realizarse esta sincronización?
- c) ¿Qué elementos de la forma general de rendezvous no se encuentran en el lenguaje ADA?

6.

Suponga que N procesos poseen inicialmente cada uno un valor. Se debe calcular el promedio de todos los valores y al finalizar la computación todos deben conocer dicho promedio.

- a) Describa las soluciones posibles con memoria distribuida para arquitecturas en estrella (centralizada), anillo circular, totalmente conectada y árbol.
- b) Implemente al menos 2 de las soluciones.
- c) Calcule la cantidad de mensajes y el tiempo para todas las soluciones.
- d) Instancie c) para $N=4$, $N=8$, $N=16$, $N=32$ y $N=64$. Analice la performance para cada caso y compare las soluciones.

Marzo 2023

- explicar ASV y dar ejemplo de sentencias que cumplan y no cumplan
 - que son y para que sirven las barreras, explicar butterfly barrier y mostrar el grafico para 16 procesos
 - como se calculan y que miden el speedup y la eficiencia
 - te dan un codigo con dos programas parecidos de PMA y te piden explicar qué hacen, analizarlos segun la cantidad de mensajes y la granularidad y decir cual funciona mejor con una arquitectura de grano grueso y por qué
 - hacer un algoritmo con PMA para resolver el problema de exclusion mutua distribuida
 - te dan un problema de un arreglo $n \times n$ que representa una ciudad y cada elemento es una esquina que tiene la cantidad de motos y la cantidad de autos que pasaron en el día y se comunica con sus 5 vecinos (3 en el caso de las esquinas del arreglo)
- y te piden hacer un algoritmo heartbeat con PM (elegis y justificas vos si usas sincronico o asincronico) para lograr que todos los elementos del arreglo conozcan qué esquina tuvo el minimo de motos y qué esquina tuvo el máximo de autos
- despues te preguntan cómo podrías mejorar no me acuerdo qué cosa, qué pasaría si no se conocieran las esquinas y algo más que tampoco me acuerdo

Abril 2023

1.

Defina programa concurrente, programa paralelo y programa distribuido

2.

Definir el problema general de asignación de recursos y su resolución mediante una política SJN. ¿Minimiza el tiempo promedio de espera? ¿Es fair? Si no lo es, plantee una alternativa que lo sea

3.

Defina el concepto de sincronización barrier.

Defina Butterfly Barrier. Instancie un ejemplo para N=16

4.

a) ¿En qué consiste la comunicación guardada (introducida por CSP) y cuál es su utilidad?

b) Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen comunicaciones guardadas

c)

```
Aux = 1;
....
if (A==0); P2?(Aux) > Aux = Aux +2;
(A==1); P3?(Aux) > Aux = Aux +5;
(B==0); P3?(Aux) > Aux = Aux +7;
endif;
...
```

- i) Si el valor de A = 1 y B = 2 antes del if, y solo P2 envía el valor 6
- ii) Si el valor de A = 0 y B = 2 antes del if, y solo P2 envía el valor 8
- iii) Si el valor de A = 2 y B = 0 antes del if, y solo P3 envía el valor 6
- iv) Si el valor de A = 2 y B = 1 antes del if, y solo P3 envía el valor 9

- v) Si el valor de $A = 1$ y $B = 0$ antes del if, y solo P3 envía el valor 14
- vi) Si el valor de $A = 0$ y $B = 0$ antes del if, P3 envía el valor 9 y P2 el valor 5

5.

Defina las métricas de speedup y eficiencia. Cual es el significado de cada una de ellas (que miden)? Cual es el rango de valores posibles de cada una?

6.

Sea la siguiente solución al problema del producto de matrices de $n \times n$ con P procesos trabajando en paralelo

```
process worker[w = 1 to P] { # strips en paralelo (p strips de n/P
  filas) }
int first = (w-1) * n/P; # Primera fila del strip
int last = first + n/P - 1; # Última fila del strip
for [i = first to last] {
  for [j = 0 to n-1] {
    c[i,j] = 0.0;
    for [k = 0 to n-1]
      c[i,j] = c[i,j] + a[i,k]*b[k,j];
  }
}
```

Suponga $n=512$ y cada procesador capaz de ejecutar un proceso

- a) Calcular cuántas asignaciones, sumas y productos se hacen secuencialmente (caso en que $P=1$), y cuántas se realizan en cada procesador en la solución paralela con $P=8$
- b) Si los procesadores P1 a P7 son idénticos, con tiempos de asignación 1, de suma y de producto 3, y si el procesador P8 es 3 veces más lento, calcule cuánto tarda el proceso total concurrente
- c) ¿Cuál es el valor del speedup?
- d) ¿Cómo modificaría el código para lograr un mejor speedup?

NOTA: para realizar los cálculos no tenga en cuenta las operaciones de asignación e incremento correspondientes a las sentencias for

Junio 2023

1.

Explicar propiedad asv(A lo sumo una vez) y que efecto tiene sobre el programa.
Dar ejemplos de sentencias que cumplan y no cumplan

2.

Suponga los siguientes programas concurrentes. Asuma que EOS es un valor especial que indica el fin de la secuencia de mensajes, y que los procesos son iniciados desde el programa principal.

P1	chan canal (double) process Genera { int fila, col; double sum; for [fila= 1 to 10000] for [col = 1 to 10000] send canal (a(fila,col)); send canal (EOS) }	process Acumula { double valor, sumT; sumT=0; receive canal (valor); while valor<>EOS { sumT = sumT + valor receive canal (valor); } printf (sumT); }	P2	chan canal (double) process Genera { int fila, col; double sum; for [fila= 1 to 10000] { sum=0; for [col = 1 to 10000] sum=sum+a(fila,col); send canal (sum); } send canal (EOS) }	process Acumula { double valor, sumT; sumT=0; receive canal (valor); while valor<>EOS { sumT = sumT + valor receive canal (valor); } printf (sumT); }
----	--	---	----	--	---

- ¿Qué hacen los programas?
- Analice desde el punto de vista del número de mensajes
- Analice desde el punto de vista de la granularidad de los procesos

3.

- ¿En qué consiste la comunicación guardada y cuál es su utilidad?
Ejemplifique.
- Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen las comunicaciones guardadas.

4.

Dado el siguiente bloque de código, indique para cada inciso que valor queda en aux o si el código queda bloqueado. Justifique sus respuestas.

```
aux = -1;  
...  
if (A == 0); P2?(aux) -> aux = aux + 2;  
□ (A == 1); P3?(aux) -> aux = aux + 5;  
□ (B == 0); P3?(aux) -> aux = aux + 7;  
end if;  
...
```

- Si el valor de A=1 y B=2 antes del IF, y sólo P2 envía el valor 6.
- Si el valor de A=0 y B=2 antes del IF, y sólo P2 envía el valor 8.
- Si el valor de A=2 y B=0 antes del IF, y sólo P3 envía el valor 6.
- Si el valor de A=2 y B=1 antes del IF, y sólo P3 envía el valor 9.
- Si el valor de A=1 y B=0 antes del IF, y sólo P3 envía el valor 14.
- Si el valor de A=0 y B=0 antes del IF, y P3 envía el valor 9 y P2 el valor 5.

5.

- a) Defina las métricas de speedup y eficiencia. ¿Cuál es el significado de cada una de ellas (que miden)? ¿Cuál es el rango de valores para cada una?
- b) Suponga que el tiempo de ejecución de un algoritmo secuencial es de 5000 unidades de tiempo, de las cuales el 90% corresponden a código paralelizable. ¿Cuál es el límite en la mejora que puede obtenerse paralelizando el algoritmo?
- c) Suponga que la solución a un problema se paraleliza sobre p procesadores de dos maneras distintas. En un caso, el speedup (S) está dado por la función $S=p-5$ y en el otro por $S=p/2$ para $p>5$. ¿Cuál de las dos soluciones se comportara más eficientemente al crecer la cantidad de procesadores? Justifique.

6.

Suponga que N procesos poseen inicialmente cada uno un valor. Se debe calcular el promedio de todos los valores y al finalizar la computación todos deben conocer dicho promedio.

- a) Describa conceptualmente las soluciones posibles con memoria distribuida para arquitectura en estrella (centralizada), anillo circular, totalmente conectada y árbol.
- b) Implemente al menos 2 de las soluciones.
- c) Para cada una de las soluciones (todas), calcule la cantidad de mensajes y el tiempo. Instancie c) para $N=4$, $N=8$, $N=16$, $N=32$ y $N=64$. Analice la performance en cada caso y compare las soluciones.

NOTA: Puede suponer que cada una de las operaciones tarda una unidad de tiempo

Septiembre 2023

1.

Explicar propiedad asv(A lo sumo una vez) y q efecto tiene sobre el programa. Dar un ejemplo que cumpla y otro que no. Justificar.

2.

Suponga los siguientes programas concurrentes. Asuma que EOS es un valor especial que indica el fin de la secuencia de mensajes, y que los procesos son iniciados desde el programa principal.

P1	<pre>chan canal (double) process Genera { int fila, col; double sum; for [fila= 1 to 10000] for [col = 1 to 10000] send canal (a(fila,col)); send canal (EOS) }</pre>	<pre>process Acumula { double valor, sumT; sumT=0; receive canal (valor); while valor<>EOS { sumT = sumT + valor receive canal (valor); } printf (sumT); }</pre>	P2	<pre>chan canal (double) process Genera { int fila, col; double sum; for [fila= 1 to 10000] { sum=0; for [col = 1 to 10000] sum=sum+a(fila,col); send canal (sum); } send canal (EOS) }</pre>	<pre>process Acumula { double valor, sumT; sumT=0; receive canal (valor); while valor<>EOS { sumT = sumT + valor receive canal (valor); } printf (sumT); }</pre>
----	---	--	----	---	--

- ¿Qué hacen los programas?
- Analice desde el punto de vista del número de mensajes
- Analice desde el punto de vista de la granularidad de los procesos

3.

- ¿En qué consiste la comunicación guardada y cuál es su utilidad? Ejemplifique.
- Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen las comunicaciones guardadas.

4.

Explicar heartbeat, token passing y servidores replicados. Indicar un ejemplo donde sean útiles y marcar sus ventajas y desventajas.

Después pedía algo de indicar si conviene usarlos con PMA o PMS pero no me acuerdo bien

5.

Defina las métricas de speedup y eficiencia. ¿Cuál es el significado de cada una de ellas (que miden)? ¿Cuál es el rango de valores para cada una?

6.

Este dijo que era obligatorio tenerlo bien para aprobar Suponga que N procesos poseen inicialmente cada uno un valor. Se debe calcular el promedio de todos los valores y al finalizar la computación todos deben conocer dicho promedio.

- a) Describa conceptualmente las soluciones posibles con memoria distribuida para arquitectura en estrella (centralizada), anillo circular, totalmente conectada y árbol.
- b) Implemente al menos 2 de las soluciones.
- c) Para cada una de las soluciones (todas), calcule la cantidad de mensajes y el tiempo. Instancie c) para $N=4$, $N=8$, $N=16$, $n=32$ y $N=64$.

Analice la performance en cada caso y compare las soluciones.

NOTA: Puede suponer que cada una de las operaciones tarda una unidad de tiempo

Octubre 2023

1.

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

2.

- a) Defina las métricas de speedup y eficiencia. ¿Cuál es el significado de cada una de ellas (que miden)? ¿Cuál es el rango de valores para cada una?.
- b) Suponga que la solución a un problema se paraleliza sobre p procesadores de dos maneras distintas. En un caso, el speedup (S) está dado por la función $S=p-5$ y en el otro por $S=p/2$ para $p>5$. ¿Cuál de las dos soluciones se comportará más eficientemente al crecer la cantidad de procesadores? Justifique.

3.

Defina los paradigmas de interacción entre procesos distribuidos Token Passing, Servidores replicados y Prueba Eco. Marque ventajas y desventajas de cada uno de ellos cuando se utiliza comunicación por mensajes sincrónicos o asincrónicos.

4.

Implemente una solución al problema de exclusión mutua distribuida entre n procesos utilizando un algoritmo Token Passing con mensajes asincrónicos.

5.

Suponga una ciudad representada por una matriz $A(n \times n)$. De cada esquina x, y se conocen dos valores enteros que representan la cantidad de autos y motos que cruzaron en la última hora. Los valores de cada esquina son mantenidos por un proceso distinto $P(x,y)$.

Cada proceso puede comunicarse con sus vecinos izquierdo, derecho, arriba y abajo, y también con los de las 4 diagonales (los procesos de las esquinas tienen solo 3 vecinos y los otros en los bordes de la grilla tienen 5 vecinos).

Febrero 2024

1.

Describa el concepto de sincronización barrier y cuál es su utilidad. ¿Qué es y cómo funciona una "butterfly barrier"? Ejemplifique gráficamente el funcionamiento de una "butterfly barrier" para 16 procesos.

2.

¿En qué consiste la comunicación guardada y cuál es su utilidad? Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen comunicaciones guardadas.

3.

Dado el siguiente bloque de código, indique para cada uno de los items si son equivalentes o no. Justificar cada caso (de ser necesario dar ejemplos).

Segmento 1	Segmento 2
<pre>... int cant=1000; While (true) { IF (cant > 15); datos?(cant) → Sentencias1 □ (cant < 5); datos?(cant) → Sentencias2 □ (INCOGNITA); datos?(cant) → Sentencias3 END IF }</pre>	<pre>... int cant=1000; DO (cant > 15); datos?(cant) → Sentencias1 □ (cant < 5); datos?(cant) → Sentencias2 □ (INCOGNITA); datos?(cant) → Sentencias3 END DO</pre>

- a) **INCOGNITA** equivale a: **(cant = 5) or (cant = 15)**
- b) **INCOGNITA** equivale a: **(cant > 0)**
- c) **INCOGNITA** equivale a: **((cant >= 2) and (cant <= 20))**
- d) **INCOGNITA** equivale a: **((cant > 5) and (cant <= 15))**
- e) **INCOGNITA** equivale a: **((cant > 5) and (cant <15))**

4.

Defina los paradigmas de interacción entre procesos distribuidos heartbeat, servidores replicados y token passing. Marque ventajas y desventajas en cada uno de ellos cuando se utiliza comunicación por mensajes sincrónicos o asincrónicos

5.

Suponga que N procesos poseen inicialmente cada uno un valor. Se debe calcular el promedio de todos los valores y al finalizar computación todos deben conocer dicho promedio.

- a) Describe conceptualmente las soluciones posibles con memoria distribuida para arquitecturas en estrella (centralizada), anillo circular, totalmente conectada y árbol.
- b) Implemente al menos 2 de las soluciones.
- c) Para cada una de las soluciones (todas), calcule la cantidad de mensajes y el tiempo (considerando que eventualmente hay operaciones que pueden concurrentemente)

Instancie c) para $N=4$, $N=8$, $N=16$, $N=32$ y $N=64$. Analice la performance para cada caso y compare las soluciones.

Nota: puede suponer que cada una de las operaciones tarda una unidad de tiempo.