Universidad Carlos III de Madrid	**Departamento de Informática** **Ingeniería Informática** **Arquitectura de computadores II** **Examen final** **16 de junio de 2011**	ARCOS

Nombre:-……

- Dispone de dos horas y quince minutos para realizar la prueba.
- No se podrán utilizar libros ni apuntes, ni calculadoras de ningún tipo.
- Los teléfonos móviles deberán permanecer desconectados durante la prueba

Pregunta 1 (1 punto).

¿Qué componentes software son necesarios para desplegar una arquitectura cluster? Cita ejemplos de funcionalidades de este software.

SOLUCIÓN

Capas de middleware:
SSI (Single System Image)
SA (System Availability):

El Single System Image (SSI) es la ilusión que presenta un conjunto de recursos como uno solo y más potente.

SSI hace aparecer al cluster como una máquina única para el usuario y sus aplicaciones.

Ejemplos de funcionalidades del SSI: un único….

punto de entrada
jerarquía de archivos
punto de control
espacio de memoria
gestor de trabajos
interfaz de usuario
espacio de E/S [SIO]
espacio de procesos [SPP]

Pregunta 2 (1.5 puntos).

El siguiente código describe la implementación de un *lock* y *unlock* empleando la instrucción atómica *Test and Set.*

```
lock:     t&s   .R1, /dir_cerrojo
          bnz   $lock
          ret

unlock:   st    #0, /dir_cerrojo
          ret
```

Se pide:

- Asumiendo una sección crítica establecida mediante este lock y unlock, describe cómo sería la ejecución de la misma cuando se ejecutan dos hilos que alcanzan esta región (el lock) de forma simultánea.
- Explica qué consiste el *Test and Set con Backoff*

Pregunta 3 (1 punto).

Sobre un determinado sistema se han hecho una serie de pruebas con un diferente tamaño de problema (variable n) y de procesos (variable p) que han arrojado los siguientes valores de aceleración:

n	p = 1	p = 4	p = 8	p = 16	p = 32
64	1.0	.80	.57	.33	.17
192	1.0	.92	.80	.60	.38
320	1.0	.95	.87	.71	.50
512	1.0	.97	.91	.80	.62

Justificar si el sistema es ó no escalable según los resultados anteriores.

SOLUCIÓN

El sistema es escalable ya que la eficiencia se mantiene al aumentar el número de procesadores y tamaño del problema.

Pregunta 4 (2.5 puntos).

Dado el siguiente código paralelo que se ejecuta en hilos distintos en una arquitectura de memoria compartida con coherencia caché basada en el protocolo MSI y consistencia secuencial.

Hilo 1 (procesador 1)	Hilo 2 (procesador 2)
`(a) x = 5`	`(c) x = 1`
`(b) x = 2`	`(d) print x`

Se asume que la variables `x` tiene un valor inicial `x=0,` y que la instrucción `print x` equivale a una instrucción de lectura. Inicialmente las memorias caché están vacías.

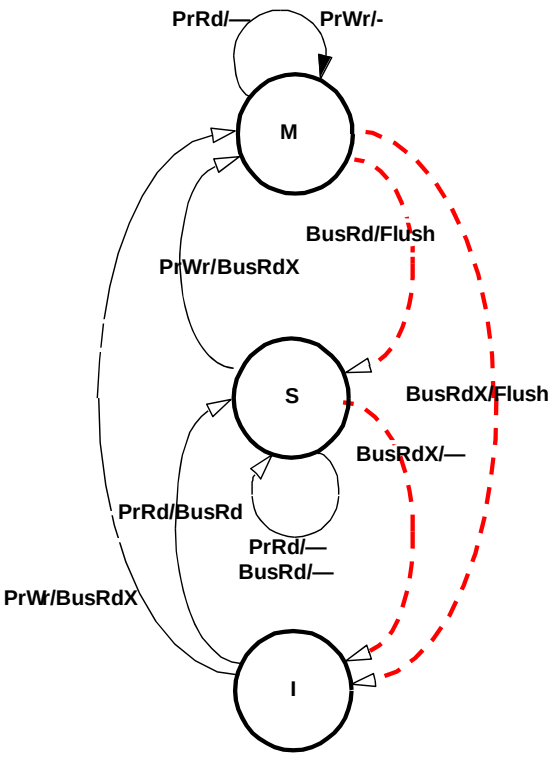
Se pide:

1. Indica qué valores de `x` se pueden imprimir bajo un modelo de consistencia secuencial. Justifica tu respuesta con ejemplos de secuencias válidas bajo este modelo.
2. Únicamente para la siguiente secuencia de ejecución de instrucciones:

   (c) → (a) → (d) → (b)

   Indicar en la siguiente tabla los estados, transiciones y transacciones de bus del sistema.

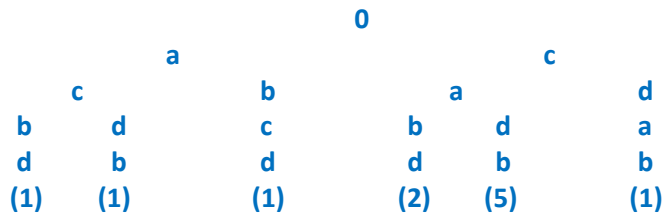
	Transición P1	Transición P2	Acciones del bus
P2: x=1			
P1: x=5			
P2: print x			
P1: x=2			



SOLUCIÓN

Apartado 1

Árbol de estados que evalúa todas las combinaciones posibles:



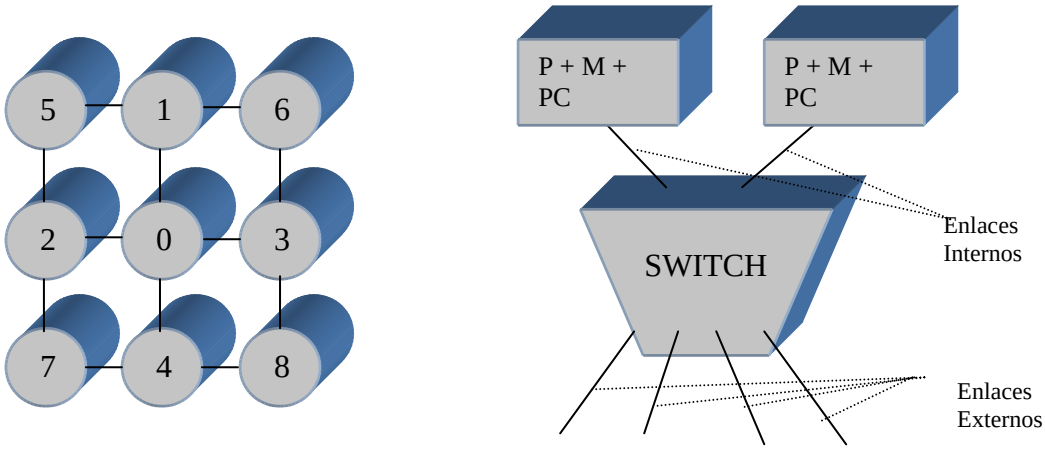
La única que no es posible es imprimir el valor 0

Apartado 2:

	Transición P1	Transición P2	Acciones del bus
P2: x=1		I -> M	BusRdX
P1: x=5	I -> M	M -> I	BusRdX/Flush
P2: print x	M -> S	I -> S	BusRd /Flush
P1: x=2	S -> M	S -> I	BusRdX

Pregunta 5 (2 puntos).

La siguiente figura muestra una topología con 9 nodos y las siguientes características:



Esquema nodos 5, 6, 7 y 8

- Cada nodo tiene un switch de comunicaciones con dos enlaces internos conectados a dos módulos de cómputo (en total 18 módulos de cómputo) y dos (nodos 5, 6, 7, 8), tres (nodos 1, 2, 3, 4) o cuatro (nodo 0) enlaces externos.
- Los enlaces internos no tienen *routing delay* y tienen un ancho de banda infinito.
- Los enlaces externos tienen un ancho de banda de 1Mbit/seg y un *routing delay* (retardo de encaminamiento del switch) de 2ms.
- El retardo de envío y recepción del procesador de comunicaciones es de 1ms.
- Los enlaces internos permiten el tráfico simultáneo de múltiples paquetes.
- Los enlaces externos permiten el tráfico simultáneo de un único paquete.
- En la red de comunicaciones no existe restricción en el tamaño del paquete (cualquier conjunto de datos puede ser enviado en un único paquete).
- El protocolo de encaminamiento es de conmutación de paquetes (*store and forward*)

Se pide:

- Diseñar el modo de orquestación de una operación de *Gather* iniciada por el nodo 0 sobre todos los nodos.
- Indicar el flujo de comunicaciones y el tiempo total de la siguiente operación (cada dato MPI_INT son 4 bytes):

MPI_Gather(x, 900, MPI_INT, y, 900, MPI_INT, 0, MPI_COMM_WORLD)

SOLUCIÓN

a) La manera más efectiva de orquestar una operación de Gather sería la siguiente:

Los nodos más alejados (5, 6, 7, 8) inician el envío hacia el nodo 0 por uno de sus dos enlaces (el 5 a través del 1, el 6 a través del 3, el 8 a través del 4, el 7 a través del 2). En paralelo los nodos intermedios (1, 2, 3, 4) inician el envío hacia el nodo 0.

b) Teniendo en cuenta el apartado anterior, el tiempo total de la operación será el del envío entre los nodos más alejados (5, 6, 7, 8) y el nodo 0:

Hay 18 módulos de cómputo y 900 datos a recolectar en la operación de Gather, por lo que hay 50 datos por módulo de cómputo.

Tamaño del paquete: 4Bytes/dato * 50 datos * 8bits/byte = 1600 bits
Tiempo de transferencia: 1600 bits/paquete / 1000000 bits/seg = 1,6mseg

Tiempo Total: Retardo envio + Tiempo Transferencia + RoutingDelay + Retardo en la recepción = 1ms + (1,6ms + 2ms)*2enlaces + 1ms = 9,2

Pregunta 6 (2.5 puntos).

Dada una arquitectura CC-NUMA, se pretende ejecutar de la forma más eficiente el siguiente código, donde la función $\sqrt[4]{x}$ tiene una complejidad (tiempo de ejecución) creciente conforme aumenta el índice i.

```
float x,tmp
DO ALL i = 1,1000
tmp = tmp + ∜x
END DO
```

1. Explicar cuál es la principal característica de las arquitecturas CC-NUMA con 16 procesadores.
2. Si pretendemos paralelizar este código, ¿qué cambios tendríamos que aplicar? Indicar los cambios a realizar y tener en cuenta cada una de las etapas del proceso de paralelización para una ejecución de 8 hilos.
3. Reescribir el código para expresar que cambios habría en el hipotético algoritmo paralelo. ¿Qué problemas hay desde el punto de vista del balanceo de carga? ¿Y sobre el valor numérico del resultado, es siempre el mismo?

SOLUCIÓN

1.-

Las arquitecturas CC-NUMA se caracterizan por ser arquitecturas de coherencia caché de acceso no uniforme a memoria.

2.-

a.- Descomposición: Dividir el problema en tareas más pequeñas. En este caso el lazo lo paralelizaremos asignado cada iteración a un procesador.

b.- Asignación: Se asigna cada iteración ó conjunto de iteraciones a un proceso. Hay que tener en cuenta que el coste de cada operación no es simétrico (por el cálculo que implica cada iteración). Este es un problema al que debe aplicarse un balanceo de carga. Por tanto sería conveniente añadir un planificador dentro del lazo (teniendo en cuenta el overead que conlleva) para conseguir un equilibrado en el balanceo de carga.

c.- Orquestación: Todos los procesos depende uno de los otros por el hecho de usar una variable compartida TMP. Solución: Privatizar la variable tmp y después aplicar una reducción.

d.-Mapeo: Se asignará cada hilo a un procesador.

3.-

Como se ha indicado en el apartado anterior existe un problema debido a que cada iteración tiene un coste computacional diferente. El hecho de manejar operaciones en punto flotante y al ser operaciones donde se aplica asociatividad en

el orden de ejecución de las operaciones de acumulación y reducción puede ocurrir que el resultado final varíe.

```
DOALL i=1,1000
#invocar a planificador dinámico#
            tmp2[Rank]   =   tmp2[Rank]   +
sqrt(x,i)
END DO
DO i=1,Nprocess
            tmp += tmp2[rank]
END
```