Hoja ejercicios 1

1. Visita la página del Top 500 Supercomputers (http://www.top500.org/) y analiza los 5 primeros supercomputadores en relación al número de FLOPS.

2. Listar los principales problemas que requieren el uso de computación de alto rendimiento en diferentes campos de la ciencia

3. Considerar el siguiente segmento de código para sumar 2 vectores:

```
1  for (i = 0; i < 1000; i++)
2          c[i] = a[i] + b[i];
```

   Dar un modelo eficiente para la ejecución de este código.

Consider the following code segment that adds two vectors:

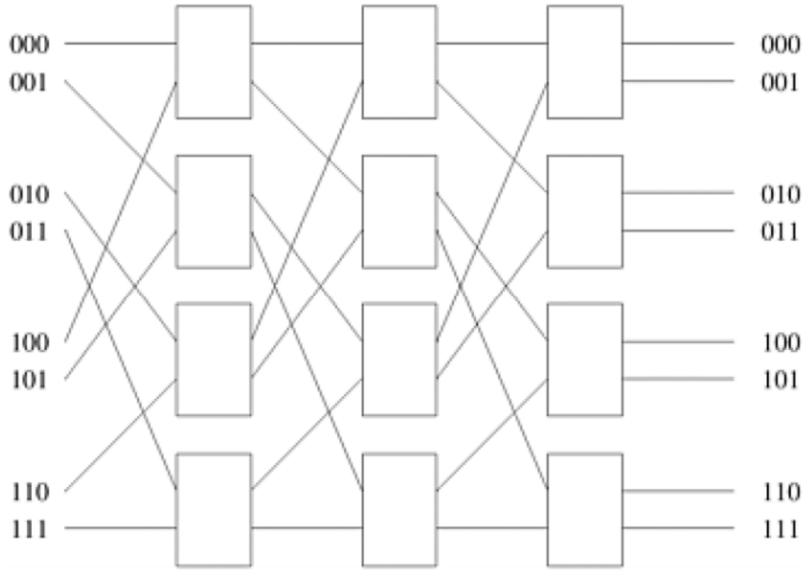
```
1  for (i = 0; i < 1000; i++)
2          c[i] = a[i] + b[i];
```

En este ejemplo, varias iteraciones del bucle son independientes entre sí; es decir, c[0] = a[0] + b[0]; c[1] = a[1] + b[1];, etc., por lo que pueden ejecutarse independientemente unos de otros. En consecuencia, si existe un mecanismo para ejecutar la misma instrucción, en este caso "sumar" en todos los procesadores con los datos apropiados, podríamos ejecutar este bucle mucho más rápido.

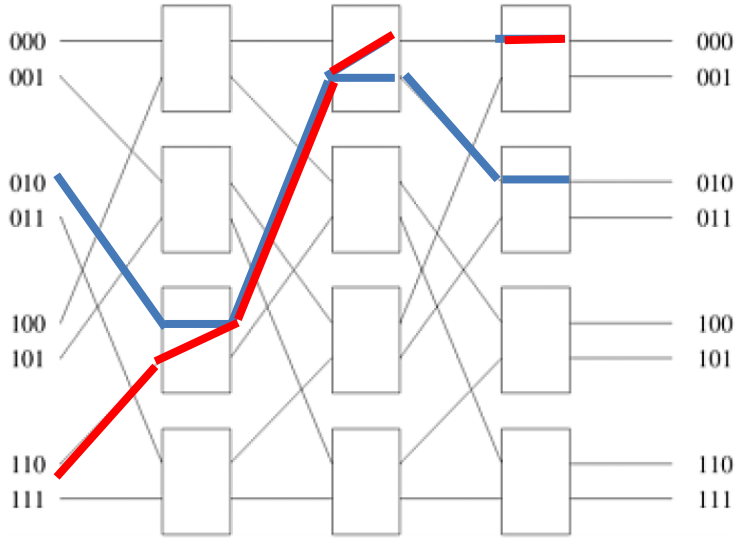
4. Cómo modelar un sistema de memoria compartida en un sistema de paso de mensajes.

Es fácil emular una arquitectura de paso de mensajes que contiene nodos p en un equipo de espacio de direcciones compartido con un número idéntico de nodos. Asumiendo nodos de un solo procesador, esto se puede hacer particionando el espacio de direcciones compartidas en p partes disjuntas y asignando una partición de este tipo exclusivamente a cada procesador. Un procesador puede entonces "enviar" o "recibir" mensajes escribiendo o leyendo desde la partición de otro procesador mientras utiliza primitivas de sincronización apropiadas para informar a otros nodos de comunicación cuando ha terminado de leer o escribir los datos. Sin embargo, emular una arquitectura de espacio de direcciones compartido en un equipo que pasa mensajes es costoso, ya que acceder a la memoria de otro nodo requiere enviar y recibir mensajes. En otras palabras habría que implementar toda una arquitectura de mensajes para acceder a las zonas de memoria de cada nodo, no es fácil.

5. Construir una red OMEGA con 8 procesadores conectados a 8 bancos de memoria.
   Una vez construida la red queremos enrutar datos desde 010 a 010 y de la 110 a 000.
   ¿Existe la posibilidad de bloqueo?
   Calcular ancho de bisección, diámetro y coste de la topología.



De 010 a 010 y de 110 a 000 hay bloqueo



Network	Diameter	Bisection Width	Arc Connectivity	Cost (No. of links)
Crossbar	1	p	1	p^2
Omega Network	log p	$p/2$	2	$p/2$
Dynamic Tree	2 log p	1	2	p - 1

6.- Sobre una topología completamente conectada con p=8:

a. Calcular el costes de la red
b. Calcular el ancho de bisección

Mirar la tabla

Network	Diameter	Bisection Width	Arc Connectivity	Cost (No. of links)
Completely-connected	1	$p^2/4$	p - 1	$p(p - 1)/2$

7.- Considerar una red 2D con p procesadores, donde el patrón de comunicación que sigue el algoritmo implementado en cada nodo se comunica con sus nodos más cercanos (asumir link bidireccionales).

c. Calcular el tiempo de comunicación para esta red asumiendo que cada mensaje contiene m palabras.
d. Calcular el tiempo de comunicación de este sistema ahora considerando otro escenario distinto donde cada nodo se comunica con otro elegido de forma aleatoria.

La red malla tendrá dimensiones $\sqrt{p} \times \sqrt{p}$ en la que cada nodo se comunica con su vecino, como los enlaces se utilizan sólo para una comunicación, el tiempo invertido en ello es $t_s + t_w m$ siendo m el número de palabras.

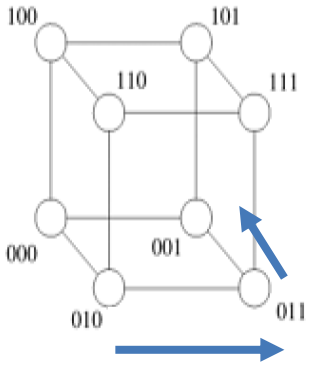
En el segundo caso la aleatoriedad implica que hay p/2 comunicaciones (o p/4 comunicaciones bidireccionales) que se producen en cualquier partición equivalente de la máquina (ya que el nodo con el que se comunica otro podría estar en cualquier mitad con la misma probabilidad). Sabemos que una malla 2-D tiene un ancho de bisección $\sqrt{p}$. De estas dos consideraciones podemos inferir que algunos enlaces tendrán que transportar $\frac{p/4}{\sqrt{p}} = \sqrt{p}/4$ mensajes asumiendo canales de comunicación bidireccional. Los mensajes, evidentemente van uno luego de otro, es decir se serializan sobre el enlace, si cada mensaje es de tamaño m el tiempo para esta operación será de al menos $t_s + t_w m \times \sqrt{p}/4$. Este tiempo evidentemente no es el mismo que el modelo simplificado. Mirar en las transparencias el modelo de comunicación para redes congectionadas.

$t_{com} = t_s + t_w \cdot m \cdot (\text{número de mensajes}/b)$ b: ancho bisección

6. Sobre una estructura de hipercubo de dimensión 3, muestra la ruta que seguirían los siguientes mensajes:

010 – 001 000-111 011-101

Resuelto en clase 010 OREX 001 → 011 (2 saltos) primer salto al 011, segundo salto al 001
Aplicar el algoritmo E cube.

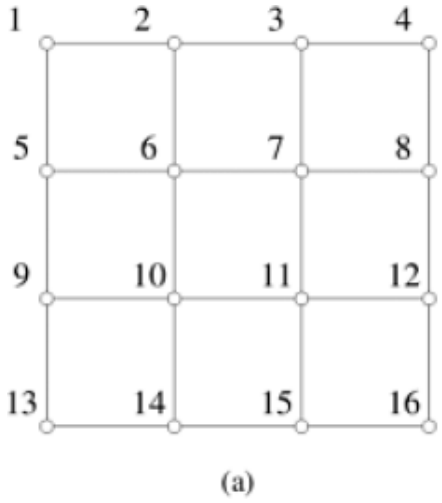


7. Cuál es la diferencia entre un algoritmo de enrutado determinista y otro que no lo es (adaptativo).

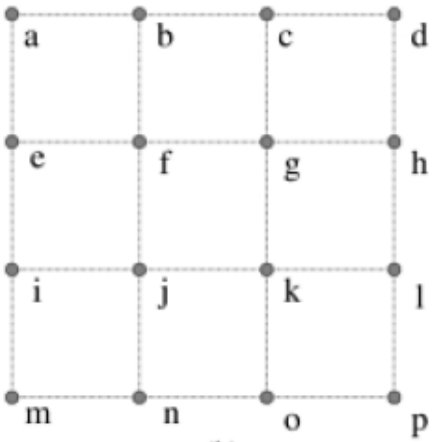
Está en las transparencias

8. Considera el escenario mostrado en la figura. Sobre una red de 16 nodos de procesamiento se quiere mapear un algoritmo que utiliza 16 procesos (a-p).
    a. Mapea los procesos sobre la red evitando congestión
    b. Si el mapeo se hace de forma aleatoria, calcula la congestión máxima que podría darse sobre la red.

k	H	M	I
J	P	O	B
D	E	A	N
C	L	G	F

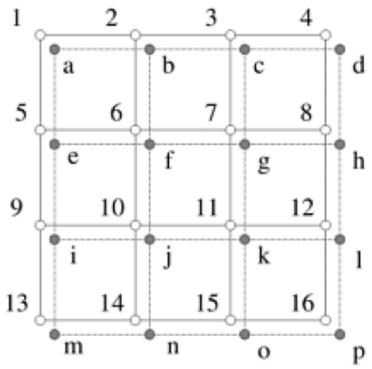


(a)

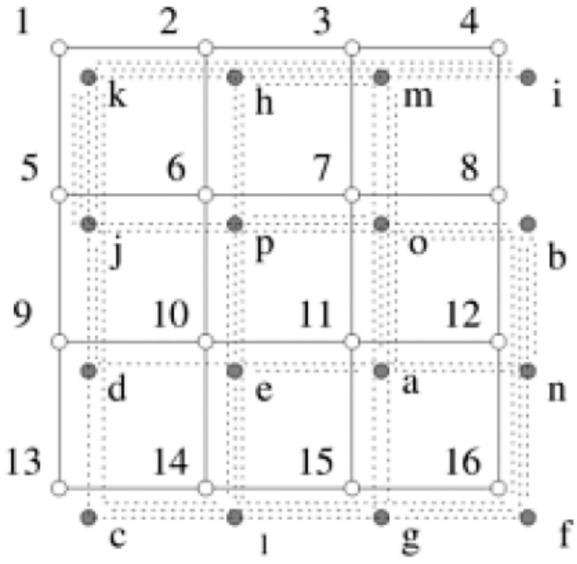


(b)

En el primer caso sería evidentemente



Para el segundo se tendría un mapeo



Es fácil ver que en este escenario un enlace podría transportar hasta 6 canales de comunicación por lo que esa sería la máxima congestión.

9. Embeber una malla 4x4 en un hipercubo.
    a. Congestión
    b. Dilatación

Está en las transparencias

Consider a $p \times q \times r$ mesh being embedded into a 2^d processor hypercube. Assume that $p = 2^x$, $q = 2^y$, and $r = 2^z$. Furthermore, since $p \times q \times r = 2^d$, $x + y + z = d$.
The embedding of the mesh can be performed as follows: Map processor (i, j, k) in the mesh to processor

10. Embeber una malla 2x4 en un hipercubo.
    a. Congestión
    b. Dilatación

Está en las transparencias