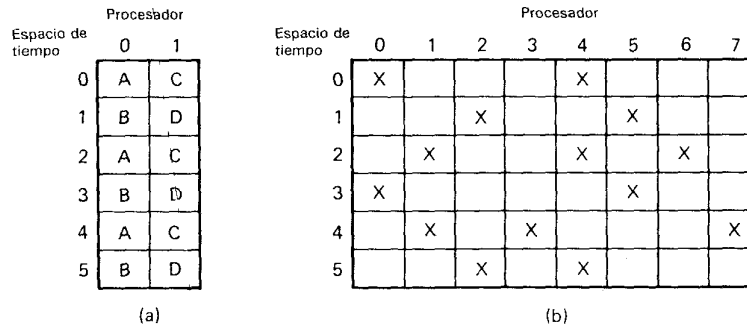


#### 4.4. PLANIFICACIÓN EN SISTEMAS DISTRIBUIDOS

No hay mucho que decir de la planificación en los sistemas distribuidos. Por lo general, cada procesador hace su planificación local (si tiene varios procesos en ejecución), sin preocuparse por lo que hacen los demás procesadores. Lo normal es que este método funcione. Sin embargo, si un grupo de procesos relacionados entre sí y con gran interacción se ejecutan en distintos procesadores, la planificación independiente no es el camino más eficiente.



**Figura 4-20.** (a) Dos procesos que se ejecutan en forma desfasada entre ellos. (b) Matriz de planificación para ocho procesadores, cada uno con seis espacios de tiempo. Las X indican los espacios asignados.

La dificultad básica se puede mostrar mediante un ejemplo, en el cual los procesos A y B se ejecutan en un procesador y los procesos C y D en otro. El tiempo de cada procesador se comparte en pedazos de 100 milisegundos, donde A y C se ejecutan en los pedazos pares y B y D en los nones, como se muestra en la figura 4-20(a). Supongamos que A envía muchos mensajes o lleva a cabo muchas llamadas a procedimientos remotos de D. Durante el tiempo 0, A inicia y llama de inmediato a D, que por desgracia no se ejecuta en ese momento, puesto que es el turno de C. Después de 100 milisegundos, se alternan los procesos, D obtiene el mensaje de A, lleva a cabo el trabajo y responde con rapidez. Puesto que B está ejecutándose, pasarán otros 100 milisegundos antes de que A obtenga la respuesta y pueda proseguir. El resultado neto es un intercambio de mensajes cada 200 milisegundos.

Lo que se necesita es una forma de garantizar que los procesos con comunicación frecuente se ejecuten de manera simultánea.

Aunque es difícil determinar en forma dinámica los patrones de común entre los procesos, en muchos casos, un grupo de procesos relacionados entre sí iniciarán juntos. Por ejemplo, en general está bien suponer que los filtros de un entubamiento en UNIX se comunicarán entre sí más de lo que lo harán con otros procesos ya iniciados. Supongamos que los procesos se crean en grupos y que la comunicación dentro de los grupos prevalece sobre la comunicación entre los grupos. Supongamos además que se dispone de un número de procesadores lo bastante grande como para manejar al grupo de mayor tamaño y que cada procesador se multiprograma con N espacios para los procesos multiprogramación de nivel N).

Ousterhout (1982) propuso varios algoritmos con base en un concepto llamado coplanificación, el cual toma en cuenta los patrones de comunicación entre los procesos durante la planificación para garantizar que todos los miembros de un grupo se ejecuten al mismo tiempo. El primer algoritmo utiliza una matriz conceptual, en la que cada

columna es la tabla de procesos de un procesador, como se muestra en la figura 4-20(b). Así, la columna 4 consta de todos los procesos que se ejecutan en el procesador 4. El renglón 3 es la colección de todos los procesos que se encuentran en el espacio 3 de algún procesador, a partir del proceso en el espacio 3 del procesador 0, el proceso en el espacio 3 del procesador 1, etc. La esencia de esta idea es que cada procesador utilice un algoritmo de planificación roundrobin, donde todos los procesadores ejecuten el proceso en el espacio 0 durante un cierto período fijo, para que después todos los procesadores ejecuten el proceso del espacio 1 durante un cierto período fijo, etc. Se puede utilizar un mensaje para indicarle a cada procesador el momento en que debe hacer un intercambio de procesos, para mantener sincronizados los intervalos de tiempo.

Si todos los miembros de un grupo se colocan en el mismo número de espacio, pero en procesadores distintos, se tiene la ventaja del paralelismo de nivel  $N$ , con la garantía de que todos los procesos se ejecutarán al mismo tiempo, lo cual maximiza el desempeño de la comunicación. Así, en la figura 4-20(b), los cuatro procesos que se deben comunicar entre sí tendrían que colocarse en el espacio 3 de los procesadores 1, 2, 3 y 4, para obtener un desempeño óptimo. Esta técnica de planificación se puede combinar con el modelo jerárquico de administración de procesos utilizado en MICROS al hacer que cada jefe de departamento mantenga la matriz de sus trabajadores, asignando procesos a los espacios en la matriz y transmitiendo las señales de tiempo.

Ousterhout también describió algunas variantes de este método básico para mejorar el desempeño. Una de estas variantes separa la matriz por renglones y concatena los renglones para formar un gran renglón. Con  $k$  procesadores, cualesquiera  $k$  entradas consecutivas pertenecen a distintos procesadores. Para asignar un nuevo grupo de procesos a las entradas, se deja una ventana de  $k$  entradas de ancho en el renglón de gran tamaño, de modo que la entrada del extremo izquierdo esté vacía pero que la entrada justo a la izquierda de la ventana esté ocupada. Si existe el número suficiente de entradas en dicha ventana, los procesos se asignan a las entradas vacías; o bien, la ventana se desliza a la derecha y se repite el algoritmo.

La planificación se lleva a cabo al iniciar la ventana en la orilla izquierda y moviéndola a la derecha una ventana por cada intervalo de tiempo, teniendo cuidado de no dividir los grupos en las ventanas. El artículo de Ousterhout analiza éstos y otros métodos con más detalle y da algunos resultados relativos al desempeño.