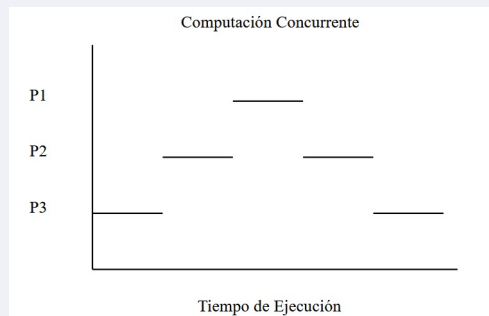




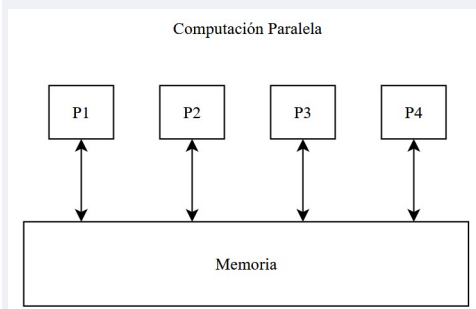
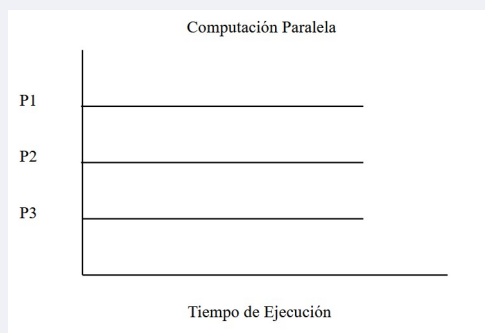
DEFINICIÓN 1: Computación Concurrente

La concurrencia es una técnica utilizada para disminuir el tiempo de respuesta del sistema que utiliza una sola unidad de procesamiento o procesamiento secuencial. Una tarea se divide en varias partes, y su parte se procesa simultáneamente pero no en el mismo instante.



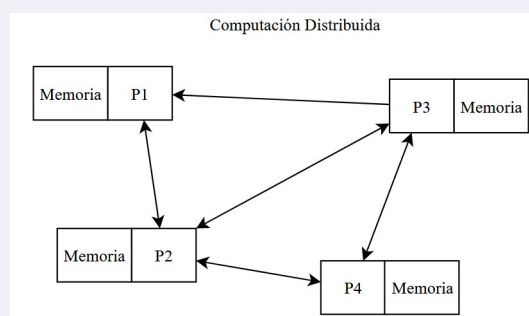
DEFINICIÓN 2: Computación paralela

El paralelismo se concibe con el fin de aumentar la velocidad de cálculo mediante el uso de múltiples procesos. Es una técnica de ejecución simultánea de diferentes tareas en el mismo instante. Implica varias unidades de procesamiento independientes que operan en paralelo y realizan tareas con el fin de aumentar la velocidad de cálculo y mejorar el rendimiento.



DEFINICIÓN 3: Computación distribuida

Computadoras conectadas que trabajan para resolver un problema en común mediante el intercambio de mensajes.



DEFINICIÓN 4: Sistema distribuido

Un sistema distribuido es una colección de dispositivos de cómputo individuales que se pueden comunicar entre sí.

DEFINICIÓN 5: Sistemas asíncronos

Un sistema es asíncrono si no hay un límite superior fijo sobre cuanto tiempo tarda en entregar un mensaje o cuanto tiempo transcurre entre los pasos consecutivos de un procesador.

DEFINICIÓN 6: Sistemas síncronos

En el modelo síncrono, los procesadores se ejecutan paso a paso: la ejecución se divide en rondas, y en cada ronda cada procesador puede enviar un mensaje a cada vecino, los mensajes se entregan, y cada procesador calcula basándose en los mensajes que acaba de recibir.

Este modelo, es muy conveniente para el diseño de algoritmos porque un algoritmo no necesita enfrentarse a mucha incertidumbre.

DEFINICIÓN 7: Proceso

Un sistema distribuido está formado por un conjunto de unidades de procesamiento, cada una de las cuales se abstrae mediante la noción de proceso. Se supone que los procesos cooperan en un objetivo común, lo que significa que intercambian información de un modo u otro.

El conjunto de procesos es estático. Se compone de n procesos y se denota $\Pi = \{p_1, \dots, p_n\}$, donde cada p_i , $1 \leq i \leq n$, representa un proceso distinto. Cada proceso p_i es secuencial, es decir, ejecuta un paso cada vez.

El entero i denota el índice del proceso p_i , es decir, la forma en que un observador externo puede distinguir los procesos. Casi siempre se asume que cada proceso p_i tiene su propia identidad, denotada id_i ; entonces p_i conoce id_i (en muchos casos (pero no siempre) $id_i = i$). Por lo general, suponemos que cada identificador consta de sólo $\log(n)$ bits si el sistema tiene n nodos.

DEFINICIÓN 8: Medio de comunicación

Los procesos se comunican enviando y recibiendo mensajes a través de canales. Se supone que cada canal es fiable (no crea, modifica ni duplica mensajes).

En algunos casos, suponemos que los canales son first in first out (FIFO), lo que significa que los mensajes se reciben en el orden en que se han enviado. Se supone que cada canal es bi-direccional (puede transportar mensajes en ambas direcciones) y tiene una capacidad infinita (puede contener cualquier número de mensajes, cada uno de cualquier tamaño). En algunos casos particulares, consideraremos canales que son unidireccionales (tales canales transportan mensajes en una sola dirección).

Cada proceso p_i tiene un conjunto de vecinos, denominado $neighbors_i$. Según el contexto, este conjunto contiene o bien las identidades locales de los canales que conectan p_i con sus procesos vecinos, o bien las identidades de estos procesos.

DEFINICIÓN 9: Algoritmo Distribuido

Un algoritmo distribuido es una colección de n autómatas, uno por proceso. Un autómata describe la secuencia de pasos ejecutados por el proceso correspondiente.

Además de la potencia de una máquina de Turing, un autómata se enriquece con dos operaciones de comunicación que le permiten enviar un mensaje por un canal o recibir un mensaje por cualquier canal. Las operaciones son `send()` y `receive()`.

DEFINICIÓN 10: Algoritmo síncrono distribuido

Un algoritmo síncrono distribuido es un algoritmo diseñado para ser ejecutado en un sistema síncrono distribuido. El progreso de dicho sistema se rige por un reloj global externo, y los procesos ejecutan colectivamente una secuencia de rondas, correspondiendo cada ronda a un valor del reloj global.

En un algoritmo síncrono, los nodos operan en rondas síncronas. En cada ronda, cada procesador ejecuta los siguientes pasos:

- Realiza algún cálculo local (de complejidad razonable).
- Enviar mensajes a los vecinos de la gráfica (de tamaño razonable).
- Recibir mensajes (enviados por los vecinos en el paso 2 de la misma ronda).

OBSERVACIÓN. A considerar ...

- Cualquier otro orden en los pasos es válido.
- ¿Qué significa "razonable" en este contexto? En general, trataremos con algoritmos que sólo realizan cálculos muy sencillos (una comparación, una suma, etc.). El cálculo en tiempo exponencial suele considerarse tramposo en este contexto. Del mismo modo, enviar un mensaje con un ID de nodo o un valor se considera correcto, mientras que enviar mensajes realmente largos se considera trampa. Tendremos definiciones más exactas más adelante, cuando las necesitemos.
- Durante una ronda, un proceso envía como máximo un mensaje a cada uno de sus vecinos. La propiedad fundamental de un sistema síncrono es que un mensaje enviado por un proceso durante una ronda r es recibido por su proceso de destino durante la misma ronda r . Por lo tanto, cuando un proceso pasa a la ronda $r + 1$, ha recibido (y procesado) todos los mensajes que se le han enviado durante la ronda r , y sabe que lo mismo ocurre con cualquier proceso.

DEFINICIÓN 11: Algoritmo asíncrono distribuido

Un algoritmo asíncrono distribuido es un algoritmo diseñado para ser ejecutado en un sistema distribuido asíncrono. En un sistema de este tipo, no existe la noción de tiempo externo. Por ello, los sistemas asíncronos se denominan a veces sin sistema sin tiempo.

En el modelo asíncrono, los algoritmos se rigen por eventos ("al recibir el mensaje . . . , haz . . . "). Los procesadores no pueden acceder a un reloj global. Un mensaje enviado de un procesador a otro llegará en un tiempo finito pero ilimitado.

OBSERVACIÓN. A considerar ...

- En un algoritmo asíncrono, el progreso de un proceso está garantizado por su propio cálculo y los mensajes que recibe. Cuando un proceso recibe un mensaje, lo procesa y según su algoritmo local, puede enviar mensajes a sus vecinos.
- Tenga en cuenta que en el modelo asíncrono, los mensajes que toman un camino más largo pueden llegar antes.
- Un proceso procesa un mensaje a la vez. Esto significa que el procesamiento de un men-

saje no puede ser interrumpido por la llegada de otro mensaje. Cuando llega un mensaje, se añade al buffer de entrada del proceso receptor. Se procesará después de que se hayan procesado todos los mensajes que le preceden en este buffer.

OBSERVACIÓN. El modelo asíncrono y el modelo síncrono son los modelos fundamentales de la computación distribuida. Como no reflejan necesariamente la realidad, existen varios modelos intermedios entre el síncrono y el asíncrono. Sin embargo, desde un punto de vista teórico, el modelo síncrono y el asíncrono son los más interesantes (porque todos los demás modelos se encuentran entre estos extremos).

DEFINICIÓN 12: Complejidad en tiempo para algoritmos síncronos

Para los algoritmos síncronos la complejidad temporal es el número de rondas hasta que el algoritmo termina.

OBSERVACIÓN. El algoritmo termina cuando el último procesador ha decidido terminar.

DEFINICIÓN 13: Complejidad en tiempo para algoritmos asíncronos

Para los algoritmos asíncronos, la complejidad temporal es el número de unidades de tiempo desde el inicio de la ejecución hasta su finalización en el peor de los casos (cada entrada, cada escenario de ejecución), suponiendo que cada mensaje tiene un retraso de una unidad de tiempo como máximo.

OBSERVACIÓN. No se puede utilizar el retardo máximo en el diseño del algoritmo. En otras palabras, el algoritmo tiene que ser correcto aunque no exista ese límite superior de retardo.

DEFINICIÓN 14: Complejidad de mensajes

La complejidad de mensajes de un algoritmo síncrono o asíncrono viene determinada por el número de mensajes intercambiados (cada entrada, cada escenario de ejecución).

DEFINICIÓN 15: Grado

El número de vecinos de un vértice v , denotado por $\delta(v)$, se denomina grado de v . El vértice de grado máximo en una gráfica G define el grado de la gráfica $\Delta(G) = \Delta$.

DEFINICIÓN 16: Sistemas anónimos

Un sistema es anónimo si los nodos no tienen identificadores únicos.

DEFINICIÓN 17: Algoritmo uniforme

Un algoritmo se denomina uniforme si el número de nodos n es desconocido para el algoritmo (para los nodos, si se desea). Si se conoce n , el algoritmo se denomina no uniforme.

DEFINICIÓN 18: Ejecución

Una ejecución de un algoritmo distribuido es una lista de eventos, ordenados por tiempo. Un evento es un registro (hora, nodo, tipo, mensaje), donde tipo es "envia" o "recibir".

OBSERVACIÓN. A considerar ...

- Suponemos que no hay dos sucesos que ocurran exactamente al mismo tiempo (o que se pueden desempatar arbitrariamente).
- La ejecución de un algoritmo asíncrono no suele estar determinada sólo por el algoritmo, sino también por un planificador "divino". Si hay más de un mensaje en tránsito, el

programador puede elegir cuál llega primero.

- Si dos mensajes se transmiten por la misma arista dirigida, a veces se exige que el mensaje transmitido primero también se reciba primero ("FIFO").

Para el límite inferior, suponemos lo siguiente:

- Se nos da un anillo asíncrono, en el que los nodos pueden despertarse en momentos arbitrarios (pero a más tardar al recibir el primer mensaje).
- Sólo aceptamos algoritmos uniformes en los que el nodo con el máximo identificador puede ser el líder. Además, cada nodo que no sea el líder debe conocer la identidad del líder.

DEFINICIÓN 19: Modelo LOCAL

El tiempo de ejecución es igual al número de rondas síncronas hasta que todos los nodos se detienen y anuncian sus salidas locales; en cada ronda cada nodo puede enviar un mensaje a cada uno de sus vecinos; el tamaño del mensaje es ilimitado; el cálculo local es libre.

DEFINICIÓN 20: Modelo CONGEST

Parecido al modelo anterior, pero los mensajes están limitados a $O(\log(n))$ bits.

DEFINICIÓN 21: Complejidad en tiempo de algoritmos distribuidos

Los modelos estándar de computación distribuida suelen asumir que la computación local es libre. Se deduce que en el modelo LOCAL de computación distribuida, se puede resolver cualquier problema de gráficas en tiempo $O(n)$, y en el modelo CONGEST de computación distribuida, se puede resolver cualquier problema de gráficas en tiempo $O(m)$ por fuerza bruta; aquí n es el número de nodos y m es el número de aristas.

Sin embargo, en estos modelos no nos interesan estos tiempos de ejecución. Por ejemplo, para el modelo LOCAL, la pregunta clave es qué puede resolverse, por ejemplo, en tiempo polilogarítmico, o en tiempo $O(\log(n))$, o incluso en tiempo $O(1)$. Ahora bien, se trata de cuestiones muy poco triviales, incluso si se supone que el cálculo local es libre.

LOCAL y CONGEST suelen ser los modelos equivocados si se está interesado en estudiar, por ejemplo, problemas NP-duros. Sin embargo, si consideras problemas "fáciles" (por ejemplo, algo que puedas resolver trivialmente en tiempo lineal con un algoritmo centralizado), entonces estos modelos se vuelven mucho más interesantes. Sí, claro que se puede encontrar un emparejamiento máximo o un conjunto independiente máximo en tiempo lineal, pero ¿se puede encontrar en tiempo sublineal?

DEFINICIÓN 22: Ordenamiento

Elegimos una gráfica con n nodos v_1, \dots, v_n . Inicialmente, cada nodo almacena un valor. Tras aplicar un algoritmo de ordenamiento, el nodo v_k almacena el k -ésimo valor más pequeño.

OBSERVACIÓN. ¿Y si dirigimos todos los valores al mismo nodo central v , dejamos que v ordene los valores localmente y luego los dirija a los destinos correctos? Según el modelo de paso de mensajes estudiado en los primeros capítulos, esto es perfectamente legal. Con una topología en estrella, la clasificación finaliza en tiempo $O(1)$.

DEFINICIÓN 23: Contención de nodos

En cada paso de un algoritmo síncrono, cada nodo sólo puede enviar y recibir $O(1)$ mensajes que contengan $O(1)$ valores, independientemente del número de vecinos que tenga el nodo.

- M. Raynal, Distributed Algorithms for Message-Passing Systems, Sec. 1.1.1
- H. Attiya, Distributed Computing, Sec. 2.1