



### **INTEGRANTES**

Yunisse Peña	1-18-2568
Félix Artiles	2-16-1692
Alberto García	2-17-1097
Diógenes Camacho	1-16-1383

### **Facilitador**

Prof. Iván Mendoza

### **ALGORITMOS PARALELOS PROYECTO FINAL**

26 de noviembre del año 2021  
Santiago de los Caballeros Rep. Dom.

## ÍNDICE

1. INTRODUCCIÓN .....	1
2. DESCRIPCIÓN DEL PROYECTO .....	2
3. OBJETIVOS.....	2
3.1. OBJETIVO GENERAL .....	2
3.2. OBJETIVOS ESPECÍFICOS.....	2
4. DEFINICIÓN DE ALGORITMOS PARALELOS.....	2
5. ETAPAS DE LOS ALGORITMOS PARALELOS .....	2
5.1. PARTICIÓN .....	2
5.2. COMUNICACIÓN .....	2
5.3. AGRUPAMIENTO.....	3
5.4. ASIGNACIÓN .....	3
6. TÉCNICAS ALGORÍTMICAS PARALELAS.....	3
6.1. DIVIDE Y VENCERÁS .....	3
6.2. SYMMETRY BREAKING .....	3
6.3. ACELERAMIENTO EN CASCADA .....	3
7. MODELOS DE ALGORITMOS PARALELOS.....	3
7.1. MODELO DE DATOS PARALELOS.....	3
7.2. MODELO DE GRAFO DE TAREAS .....	4
7.3. MODELO DE GRUPO DE TRABAJO .....	5
7.4. MODELO MAESTRO-ESCLAVO .....	6
7.5. MODELO PRODUCTOR-CONSUMIDOR .....	6
7.6. MODELO HÍBRIDO.....	7

## 1. INTRODUCCIÓN

La mayoría de los algoritmos de hoy en día se manejan de forma secuencial, por lo que siguen una secuencia de pasos, donde cada uno consiste en una acción u operación. Además, son los que están adaptados a las computadoras que utilizamos. Sin embargo, las mismas computadoras tuvieron una mejora la cual implicó operar en paralelo los procesos, usando esto para poder resolver problemas de forma más eficiente. Para lograrlo se necesita utilizar algoritmos que realicen varias operaciones en un solo paso. Por eso, en este documento se desarrollará toda la información con respecto a ellos y se implementarán para que se pueda solucionar un problema de forma más rápida y eficiente.

## **2. DESCRIPCIÓN DEL PROYECTO**

El proyecto se desarrolla con la finalidad de poder ejecutar todos los algoritmos de búsquedas al mismo tiempo y a la vez, calcular el tiempo que toma cada uno en ejecutarse. Con esto, se trata de obtener mejores soluciones más eficientes y rápidas para múltiples tareas que deben hacer con algoritmos de búsqueda, eliminando la forma secuencial que toman los algoritmos para resolver operaciones, donde en vez de varios pasos, solo se hará uno.

## **3. OBJETIVOS**

### **3.1. OBJETIVO GENERAL**

- Llevar a cabo el diseño, desarrollo y aplicación de cada uno de los algoritmos de búsqueda para lograr resolver muchas operaciones que necesitan de estos, de forma rápida y eficiente.

### **3.2. OBJETIVOS ESPECÍFICOS**

- Hacer un análisis de cada uno de los algoritmos de búsqueda, comprender cada uno de ellos para lograr que los algoritmos estén más optimizados al momento de enfrentarse a un problema.
- Hacer el diseño de un algoritmo de búsqueda para resolver aquellos problemas que necesiten ir en un solo paso.
- Implementar el algoritmo de búsqueda haciendo pruebas con los diferentes modelos de búsqueda que ayudan al trabajo en paralelo.

## **4. DEFINICIÓN DE ALGORITMOS PARALELOS**

Aquellos algoritmos que pueden ejecutarse de forma segmentada, ejecutándose cada segmento al mismo tiempo, pero al final se unen para dar un solo resultado.

## **5. ETAPAS DE LOS ALGORITMOS PARALELOS**

### **5.1. PARTICIÓN**

Se hace una segmentación de tareas, donde cada segmento trabaja con un cálculo específico, estas mismas segmentaciones son independientes del modelo de programación. Para que la segmentación sea correcta, se deben dividir los cálculos del problema y los datos de los cuales se fían.

### **5.2. COMUNICACIÓN**

Las tareas no pueden ejecutarse de forma independiente, pero si pueden generarselas particiones que fueron propuestas de forma concurrente, dicho esto, los datos deben de transferirse entre las diferentes tareas para que los cálculos procedan, ya que mientras las tareas se van ejecutando, muchas veces se necesitan datos que están asociados a otras tareas.

### **5.3. AGRUPAMIENTO**

A partir de tener las 2 primeras etapas, entonces son evaluados los requisitos de costos y ejecución de implementación, para así poder verificar si es necesario, agrupar las tareas en tareas más grandes, para que así la ejecución pueda ser mejory el costo de sincronización y comunicación sea menor.

### **5.4. ASIGNACIÓN**

Se asigna cada tarea a un procesador con el fin de que trate de satisfacer las metas al maximizar el uso del procesador y al disminuir los costos de comunicación.

## **6. TÉCNICAS ALGORÍTMICAS PARALELAS**

### **6.1. DIVIDE Y VENCERÁS**

Es un algoritmo planteado en una situación real, en el cual se tiene un problema, este se subdivide o se parte en pedazitos para poder solucionarse de forma más rápida, uniendo la solución de todas las partesitas. Lo que pasa con esta técnica es que cadaparte del algoritmo se soluciona varias veces, hasta que se encuentre la forma más sencilla de hacerlo.

### **6.2. SYMMETRY BREAKING**

Es la técnica que permite romper la simetría de una estructura o de un gráfico también, y que estos mismos puedan ser vistos de forma local y de la misma manera en todos los vértices se implementa con un aleatorizador. Dentro de esta técnica existen 2 tipos, el primero es el de la ruptura explícita que le indica las situaciones que se presentan en el grupo y también está la ruptura espontánea que se ve cuando hay soluciones que no varían.

### **6.3. ACELERAMIENTO EN CASCADA**

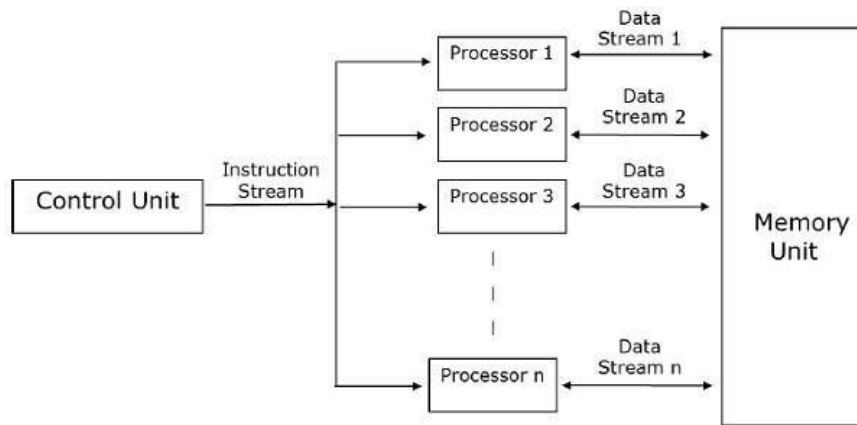
Esta técnica hace una combinación en un algoritmo, que mezcla lo rápido e ineficaz con un trabajo que es óptimo, por ello los problemas son divididos en muchos problemitas, y estos son solucionados por esos algoritmos óptimos y sus resultados se mezclan con la parte rápida del algoritmo.

## **7. MODELOS DE ALGORITMOS PARALELOS**

El modelo de un algoritmo paralelo se desarrolla considerando una estrategia para dividir los datos y el procesamiento, y aplicar una estrategia adecuada para reducir las iteraciones. A continuación, se van a mostrar los diferentes modelos de algoritmos paralelos que existen.

### **7.1. MODELO DE DATOS PARALELOS**

Las tareas en los modelos paralelos de datos se asignan a procesos y de cada una de las tareas, se hacen algunas operaciones similares en diferentes datos. Agregando a esto, el paralelismo de datos es una consecuencia de operaciones únicas que se aplican a varios elementos de datos.

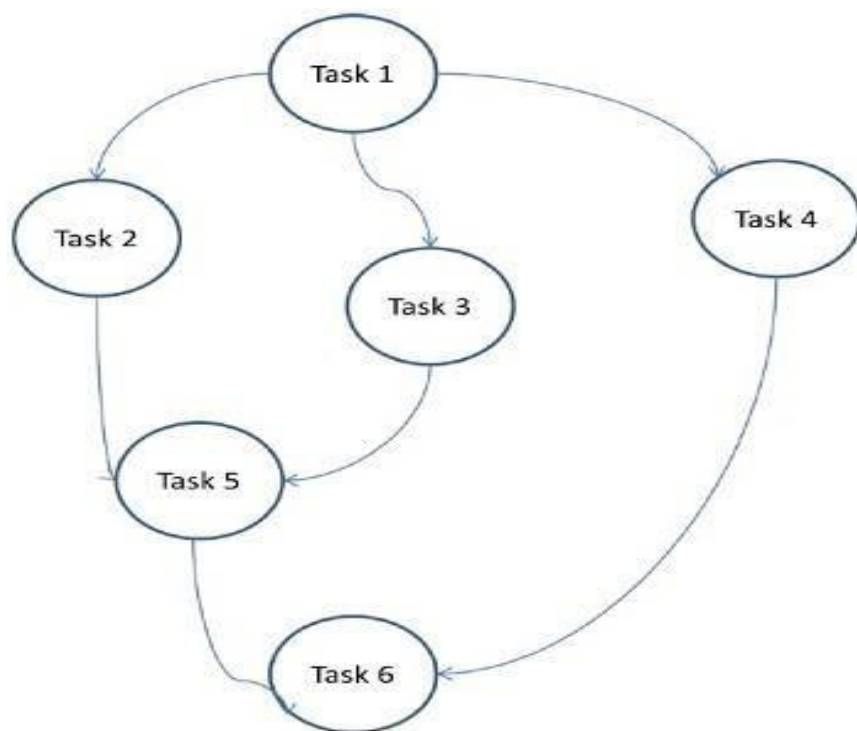


El modelo paralelo de datos es aplicado en espacios compartidos y en paradigmas de paso de mensajes. Además, en este tipo de modelo, los gastos de interacción pueden reducirse al seleccionar una localidad que cuide y mantenga la descomposición, usando rutinas de interacción colectiva optimizadas o superponiendo el cálculo y la interacción.

La característica principal de los problemas de modelos de datos paralelos es que la intensidad del paralelismo de datos aumenta con el tamaño del problema, lo que a su vez permite utilizar más procesos para resolver problemas de mayor tamaño.

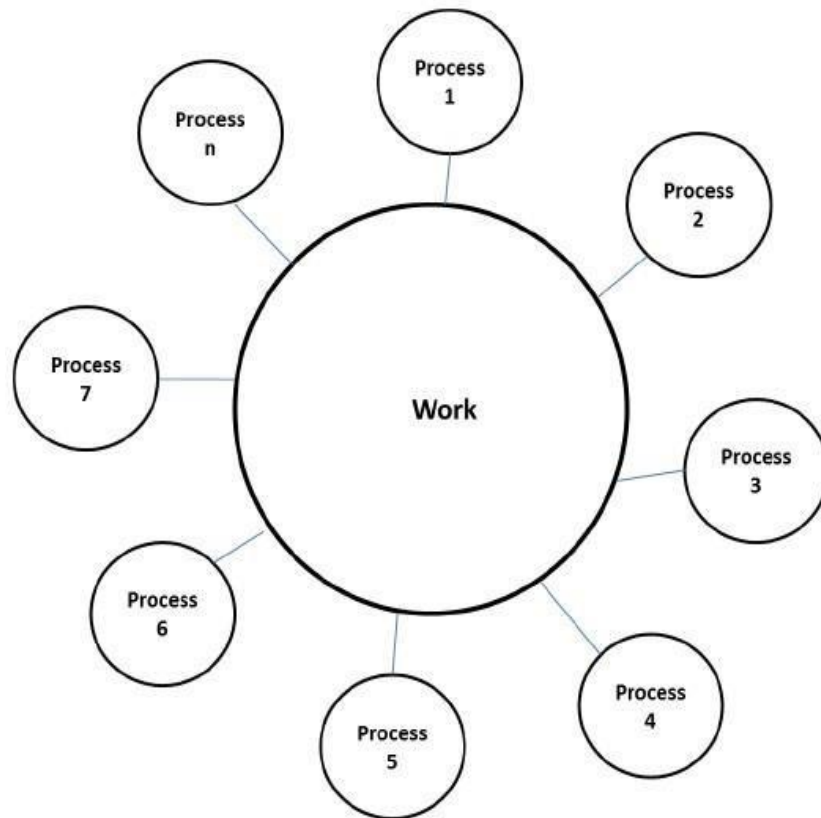
## 7.2. MODELO DE GRAFO DE TAREAS

En el modelo de gráfico de tareas, el paralelismo se expresa mediante un gráfico de tareas, dicho gráfico puede expresarse de forma trivial o no trivial. Además, la correlación entre las tareas que se manejan en este modelo, se utiliza para elevar la localidad o disminuir los costos de la interacción. Por otro lado, el modelo se utiliza para poder solucionar problemas en los que el número de datos que van de la mano con las tareas, es mucho mayor comparado con el número de cálculos que están relacionados a ellas.



Los problemas pueden ser divididos en tareas atómicas que se implementan como un gráfico. Cada tarea es una unidad de trabajo independiente que depende de una o más tareas anteriores. Luego de haber completado una tarea, el resultado de una tarea anterior se hace la tarea dependiente. Cuando una tarea tiene otra tarea antecedente comienza a ejecutarse solo cuando se completa toda su tarea antecedente y el resultado final del gráfico se recibe cuando se completa la última tarea dependiente.

### 7.3. MODELO DE GRUPO DE TRABAJO

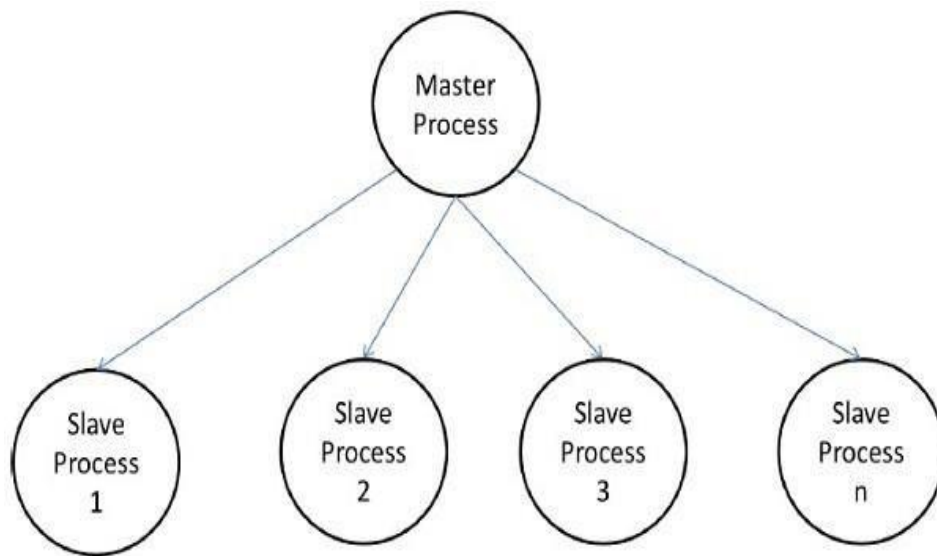


En este tipo de modelo, las tareas son asignadas de forma dinámica a los procesos para lograr equilibrar la carga. Por esta razón, cualquier proceso puede ejecutar potencialmente cualquier tarea. Este modelo es usado cuando el número de datos asociados con las tareas se compara menor que el cálculo asociado con las tareas.

No existe una asignación deseada de forma previa de tareas a los procesos. Esta asignación de tareas se encuentra descentralizada. Los punteros a las tareas se guardan en una lista compartida físicamente, en una cola de prioridad o en una tabla o árbol hash, o podrían guardarse en una estructura de datos distribuida físicamente.

La tarea se puede ver disponible al inicio o se puede generar de forma dinámica. Si la tarea es de forma dinámica y se realiza una asignación descentralizada de la tarea, entonces se requiere un algoritmo de detección de terminación para que todos los procesos detecten cuando finaliza todo el programa y se deja de buscar más tareas.

#### 7.4. MODELO MAESTRO-ESCLAVO



En este tipo de modelo, los procesos maestros pueden generar tareas y estas mismas asignarlas a procesos esclavos. Las tareas se pueden asignar si el maestro puede estimar el volumen de las tareas, o una asignación aleatoria puede hacer un trabajo satisfactorio de equilibrio de carga, o se les asignan los esclavos a tareas más pequeñas en diferentes momentos.

Este modelo es adecuado para el uso de paradigmas de espacio de direcciones compartidas o de paso de mensajes, ya que la interacción es de dos formas. Aunque en algunas situaciones, es posible que una tarea deba completarse en fases, y la tarea en cada fase debe completarse antes de que se pueda generar la tarea en las siguientes fases. Además, este modelo, se puede generalizar a un modelo maestro- esclavo jerárquico o multinivel en el que el maestro de nivel superior alimenta la gran parte de las tareas al maestro de segundo nivel, quien subdivide aún más las tareas entre sus propios esclavos y puede realizar una parte de la tarea en sí.

Se debe tener precaución al usar este modelo, dentro de las vertientes que debe tener en cuenta, están el tener cuidado para asegurar que el maestro no se convierta en un punto de congestión. Puede suceder si las tareas son demasiado pequeñas o los trabajadores son comparativamente rápidos.

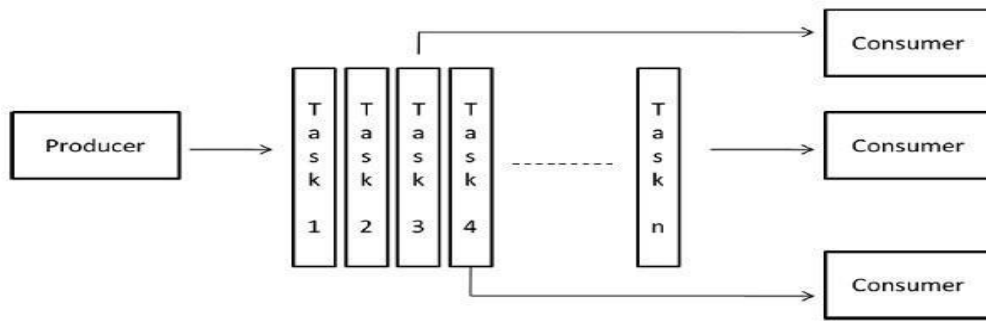
Las tareas deben seleccionarse de manera que el costo de realizar una tarea domine el costo de la comunicación y el costo de la sincronización.

La interacción asincrónica puede ayudar a superponer la interacción y el cálculo asociado con la generación de trabajo por parte del maestro.

#### 7.5. MODELO PRODUCTOR-CONSUMIDOR

También se lo conoce como **modelo productor-consumidor**. Aquí, un conjunto de datos se transmite a través de una serie de procesos, cada uno de los cuales realiza alguna tarea en él. Aquí, la llegada de nuevos datos genera la ejecución de una nueva tarea por un proceso en la cola. Los procesos podrían formar una cola en forma de matrices lineales o multidimensionales, árboles o gráficos generales con o sin ciclos.





El modelo es como una cadena de productores y consumidores, donde cada proceso de la cola se puede ver como consumidor de una secuencia de elementos de datos para el proceso que lo precede en la cola y como productor de datos para el proceso que lo sigue en la cola. La cola no necesita ser una cadena lineal, sino que puede ser un gráfico dirigido. La técnica de minimización de interacciones más común aplicable a este modelo es la interacción superpuesta con la computación.

## 7.6. MODELO HÍBRIDO

Se requiere un modelo de algoritmo híbrido cuando se puede necesitar más de un modelo para resolver un problema.

Este tipo de modelo, puede componerse por varios modelos aplicados de forma jerárquica o de forma secuencial a diferentes fases de un algoritmo paralelo.