**Información util:**

* Definición de computación distribuida y paralela.
* Ventajas y desventajas en comparación con el procesamiento secuencial.
* Aplicaciones en la industria y en la investigación

**Bibliografía:**

Introducción a la computacón paralela:  
<https://www.researchgate.net/profile/Jose-Aguilar-51/publication/267367623_Introduccion_a_la_Computacion_Paralela/links/544f7d6a0cf29473161ce888/Introduccion-a-la-Computacion-Paralela.pdf>

Computacion Distribuida ´ basada en Objetivos:

<https://dspace.itsjapon.edu.ec/jspui/bitstream/123456789/407/1/computacion-distribuida-basada-en-objetivos.pdf>

https://books.google.es/books?hl=es&lr=&id=n\_Q5EAAAQBAJ&oi=fnd&pg=PR3&dq=parallel+computation+and+distributed+computation+&ots=efDyKgyxp9&sig=5w52e8FNGgOrnhj6S7oCMmc\_kDE&redir\_esc=y#v=onepage&q=parallel%20computation%20and%20distributed%20computation&f=false

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<https://www.redalyc.org/journal/2570/257051186010/?utm_source=chatgpt.com>

<https://fastercapital.com/es/contenido/Computacion-distribuida-y-paralela--creacion-de-una-ventaja-competitiva--el-papel-de-la-computacion-distribuida-y-paralela-en-las-empresas.html?utm_source=chatgpt.com#Computaci-n-distribuida-en-la-era-del-Big-Data>

**Definición**

**Link de la info:** [**https://books.google.es/books?id=n\_Q5EAAAQBAJ&printsec=frontcover&hl=es#v=onepage&q&f=false**](https://books.google.es/books?id=n_Q5EAAAQBAJ&printsec=frontcover&hl=es#v=onepage&q&f=false)

**Obtenido del punto 1 del artículo:**

La computación paralela y distribuida ha crecido debido a la necesidad de resolver problemas complejos y los avances tecnológicos. Su desarrollo responde a aplicaciones como ecuaciones diferenciales parciales (PDEs), dinámica de fluidos, procesamiento de imágenes y optimización de sistemas interconectados. La computación distribuida también se emplea en redes de sensores y comunicación, donde la descentralización y la confiabilidad son clave.

Los sistemas paralelos agrupan varios procesadores en un mismo entorno con comunicación rápida y estable, trabajando juntos en una tarea específica. En cambio, los sistemas distribuidos conectan procesadores geográficamente separados mediante redes, donde la comunicación puede ser más lenta o menos confiable debido a fallos o variaciones en los enlaces.

Existen diferentes tipos de computaciones paralelas:

* **Masivamente paralelas:** Miles de procesadores para problemas complejos.
* **Control centralizado o descentralizado:** Desde instrucciones comunes hasta autonomía total.
* **Sincronización síncrona o asíncrona:** La primera facilita el control, mientras que la segunda es más flexible.
* **Interconexión de procesadores:** Puede ser por memoria compartida, paso de mensajes o modelos híbridos.

Estos avances han mejorado la capacidad de cómputo, permitiendo resolver problemas antes inabordables.

**Ventajas y Desventajas**

LINK:<https://fastercapital.com/es/contenido/Computacion-distribuida-y-paralela--creacion-de-una-ventaja-competitiva--el-papel-de-la-computacion-distribuida-y-paralela-en-las-empresas.html?utm_source=chatgpt.com#Introducci-n-a-la-computaci-n-distribuida-y-paralela>

La computación distribuida y paralela se utilizan para mejorar el rendimiento de las aplicaciones al dividir tareas entre múltiples procesadores o nodos.

**-Ventajas**

### **Mayor Velocidad y Eficiencia**

* **Distribución de carga:** Permite dividir el problema en varias tareas ejecutadas simultáneamente, reduciendo el tiempo total de ejecución en comparación con el procesamiento secuencial.
* **Mejor uso del hardware:** Se aprovechan múltiples núcleos o máquinas para ejecutar tareas más rápido.

### **Escalabilidad**

* En sistemas distribuidos, se pueden agregar más nodos para aumentar la capacidad de cómputo sin necesidad de reemplazar el hardware existente.
* La computación paralela permite optimizar el rendimiento al utilizar múltiples procesadores en una sola máquina.

### **Eficiencia en el Manejo de Grandes Volúmenes de Datos**

* Se pueden procesar grandes conjuntos de datos de manera más eficiente (ejemplo: Big Data con Hadoop y Spark).
* Útil en aprendizaje automático, simulaciones científicas y modelado de sistemas complejos.

### **Tolerancia a Fallos**

* Los sistemas distribuidos pueden continuar funcionando incluso si un nodo falla, mejorando la fiabilidad.
* Algunas arquitecturas paralelas implementan redundancia para evitar fallos críticos.

### **Ahorro Energético en Algunas Aplicaciones**

* En entornos distribuidos, las tareas pueden ser asignadas dinámicamente a nodos con menor consumo energético.
* En comparación con un solo procesador de alto rendimiento, distribuir la carga en varios núcleos pequeños puede reducir el consumo total de energía.

**-Desventajas**

### **Complejidad en la Implementación**

* Requiere algoritmos específicos para la distribución y sincronización de tareas.
* Es más difícil de depurar y mantener debido a la concurrencia y comunicación entre procesos.

### **Problemas de Comunicación y Latencia**

* En sistemas distribuidos, la comunicación entre nodos introduce latencia y sobrecarga en la red.
* La sincronización de tareas en la computación paralela puede generar cuellos de botella.

### **Mayor Consumo de Recursos Iniciales**

* Se necesita hardware más potente o una red de computadoras, lo que puede aumentar costos.
* Los algoritmos deben ser diseñados o adaptados para aprovechar el paralelismo, lo que requiere más tiempo de desarrollo.

### **Problemas de Consistencia y Sincronización**

* En sistemas distribuidos, garantizar la consistencia de los datos puede ser un desafío (ejemplo: bases de datos distribuidas).
* En paralelismo, el acceso concurrente a memoria compartida puede generar condiciones de carrera o bloqueos.

| **Criterio** | **Computación Secuencial** | **Computación Paralela** | **Computación Distribuida** |
| --- | --- | --- | --- |
| **Velocidad** | Lenta, ya que solo usa un núcleo. | Más rápida en hardware con múltiples núcleos. | Depende de la red y los nodos disponibles. |
| **Escalabilidad** | Limitada por la velocidad de un solo procesador. | Buena en hardware con múltiples núcleos. | Alta, se pueden agregar más nodos. |
| **Tolerancia a fallos** | Si el sistema falla, la ejecución se detiene. | Puede tener redundancia en algunos casos. | Alta, ya que otros nodos pueden continuar la ejecución. |
| **Complejidad de programación** | Simple, lógica secuencial. | Moderada, requiere manejo de hilos y concurrencia. | Alta, implica coordinación entre múltiples sistemas. |
| **Consumo de energía** | Puede ser alto si se usa un procesador potente. | Puede ser más eficiente en algunos casos. | Variable, depende de la distribución de carga. |
| **Latencia** | Baja, ya que no hay comunicación entre procesos. | Puede haber latencia si los hilos compiten por recursos. | Alta, debido a la comunicación entre nodos. |