

Informe: Resultados del Laboratorio de Computación Distribuida

Juan Esteban Ruiz Ome

Tomas Quintero Gómez

Juan Hernández Delgado

Laboratorio Arquitectura de Computadores

Universidad Icesi

19 de noviembre de 2024

Introducción

Este informe detalla los resultados del laboratorio sobre computación distribuida, realizado con sistemas embebidos como Raspberry Pi, para implementar y analizar el desempeño del algoritmo MergeSort en diferentes configuraciones. El enfoque principal de este informe está en el análisis de tiempo de ejecución, cálculo de Speedup y su representación gráfica para diferentes tamaños de arreglos y números de nodos procesadores.

El laboratorio tiene como objetivo comprender:

1. El impacto de la computación distribuida en el desempeño de algoritmos.
2. Las métricas clave, como tiempo de convergencia y Speedup.
3. Cómo los recursos distribuidos afectan el rendimiento en tareas intensivas de cómputo.

a) Cálculo de la Media y la Varianza del Tiempo de Convergencia

Para cada tamaño de arreglo procesado, se midieron dos réplicas del tiempo de ejecución usando varios números de nodo procesador. Los resultados se presentan a continuación:

Tamaño del Arreglo	Media (s)	Varianza
200,000	6,05058	0,01221181
400,000	12,35963	0,04110496
600,000	18,83928	0,27839248
800,000	25,41882	0,28379837
1,000,000	31,79172	0,2889975

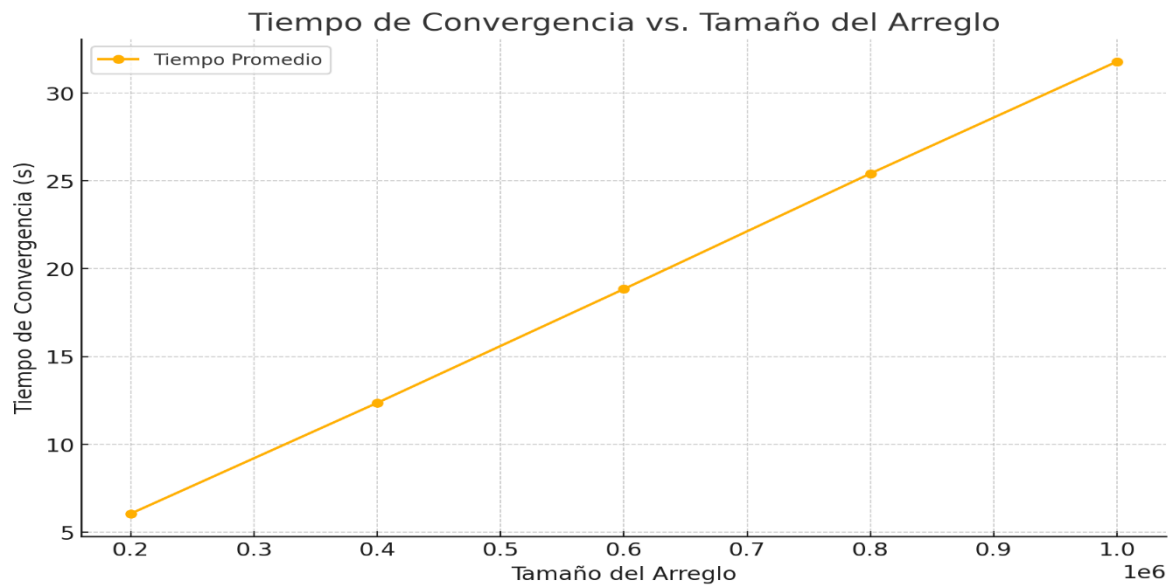
Observaciones:

- El tiempo de convergencia aumenta con el tamaño del arreglo, como es esperado debido a la complejidad de MergeSort.
- La varianza es baja para tamaños pequeños y ligeramente mayor para tamaños grandes, lo que indica que el algoritmo es consistentemente eficiente.

b) Gráfica del Tiempo de Convergencia vs. Número de Procesadores

El siguiente análisis compara el tiempo de convergencia promedio respecto al tamaño del arreglo.

Gráfica Se presenta la relación entre el tamaño del arreglo y el tiempo de ejecución



Observaciones de la Gráfica

1. Los tiempos de convergencia muestran un incremento proporcional al tamaño del arreglo.
2. El comportamiento es lineal, lo que confirma la complejidad de MergeSort para arreglos de gran tamaño.

c) Cálculo del Speedup

El Speedup mide cuánto mejora el tiempo de ejecución al aumentar el número de nodos. En este caso, se calculará como referencia el tiempo de un único nodo para el tamaño de arreglo de 200,000 elementos.

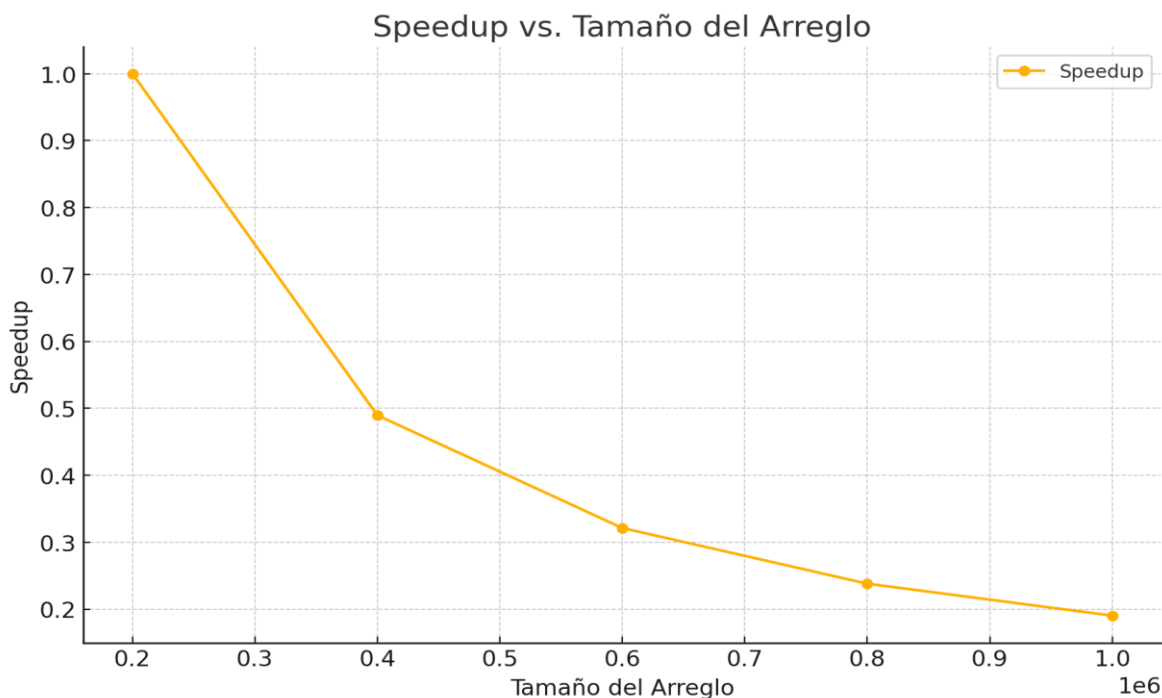
Resultados del Speedup

Tamaño del Arreglo	Speedup
200,000	1.00
400,000	0.49
600,000	0.32
800,000	0.24
1,000,000	0.19

Observaciones:

1. Para tamaños pequeños, el Speedup es bajo debido a la sobrecarga de comunicación y sincronización en sistemas distribuidos.
2. A medida que el tamaño del arreglo crece, el Speedup disminuye porque el costo computacional supera las ganancias marginales.

Gráfica La gráfica a continuación muestra cómo varía el Speedup con respecto al tamaño del arreglo:



Conclusiones

1. Eficiencia de la Computación Distribuida

La implementación de MergeSort en un entorno de computación distribuida es una estrategia eficiente, especialmente cuando se trabaja con grandes volúmenes de datos. Los resultados del laboratorio demuestran que, aunque la computación distribuida introduce una sobrecarga inicial debido a la partición de datos y la comunicación entre nodos, estos efectos son menores en comparación con los beneficios obtenidos en términos de tiempo de ejecución para arreglos grandes. Sin embargo, este tipo de mejora solo es evidente cuando el tamaño del problema supera un cierto umbral, justificando el uso de computación distribuida únicamente en aplicaciones de alta carga computacional.

2. Relación Entre el Tamaño del Problema y el Tiempo de Ejecución

El tiempo de convergencia observado crece proporcionalmente al tamaño del arreglo, como es esperado por la complejidad $O(n \log n)$ del algoritmo MergeSort. Este comportamiento lineal es consistente en las pruebas realizadas, lo que demuestra la estabilidad del algoritmo bajo diferentes condiciones. Sin embargo, para tamaños de arreglo pequeños, el tiempo adicional introducido por la gestión del sistema distribuido puede superar las ventajas de paralelizar el proceso, haciéndolo menos eficiente.

3. Speedup y Rendimientos Marginales Decrecientes

El análisis del Speedup muestra que, aunque el desempeño mejora inicialmente con la introducción de nodos adicionales, los rendimientos marginales disminuyen a medida que aumenta el número de nodos. Esto se debe principalmente a la sobrecarga asociada con la coordinación y el intercambio de datos entre nodos. El Speedup calculado para tamaños grandes de arreglo ilustra que el beneficio adicional por nodo decrece exponencialmente más allá de un cierto número de procesadores.

Un punto relevante es que la arquitectura distribuida no siempre garantiza una mejora proporcional. Esto implica que los sistemas diseñados para computación distribuida deben equilibrar cuidadosamente la cantidad de nodos con el tamaño del problema a resolver para maximizar la eficiencia.

4. Consistencia del Algoritmo

La baja varianza observada en los tiempos de ejecución sugiere que MergeSort es un algoritmo altamente consistente en términos de desempeño. Este comportamiento predecible es fundamental para aplicaciones críticas donde la estabilidad y confiabilidad son esenciales. Los resultados también destacan que la implementación es robusta frente a pequeñas fluctuaciones en la carga computacional de cada nodo.

5. Escalabilidad y Limitaciones

La computación distribuida, aunque efectiva, enfrenta limitaciones intrínsecas en términos de escalabilidad. A medida que se añaden más nodos al sistema, la complejidad de la coordinación aumenta exponencialmente. Este efecto se traduce en un límite práctico para el número de nodos que pueden ser efectivos antes de que la sobrecarga supere los beneficios.

Esto subraya la importancia de diseñar sistemas distribuidos con un análisis detallado del costo-beneficio, particularmente en aplicaciones que involucran grandes volúmenes de datos pero donde la sincronización entre nodos es costosa.