Algoritmo bucketsort

<table>
<tr><td colspan="2">UNIDAD DE APRENDIZAJE: Aplicaciones para comunicaciones en red
UNIDAD TEMÁTICA IV: Hilos</td></tr>
<tr><td>No. y título de la práctica:

Algoritmo bucketsort</td><td rowspan="2">Tiempo de realización: 3 horas</td></tr>
<tr><td>Objetivo de la práctica: El estudiante implementará el algoritmo de ordenamiento bucket sort en un entorno distribuido haciendo uso de hilos.</td></tr>
<tr><td colspan="2">Situación problemática: En muchos tipos de aplicaciones se requiere aplicar técnicas de ordenamiento de datos, por ejemplo, en procesamiento de imágenes existen muchos tipos de filtros que hacen uso de estas técnicas, sin embargo, cuando la cantidad de datos a ordenar se vuelve demasiado grande, algoritmos como el quicksort se vuelven poco eficientes, simplemente porque no es posible almacenar tal cantidad de datos en un arreglo. En estos casos el algoritmo bucketsort es una buena alternativa, ya que puede ser paralelizado y la carga de procesamiento puede ser distribuida en varios servidores para acelerar el ordenamiento. La única condición es que la distribución de los números sea uniforme.</td></tr>
<tr><td colspan="2">Competencia específica: Desarrolla aplicaciones con hilos para el envío de datos para la paralelización de los procesos.</td></tr>
<tr><td>Competencias genéricas:
Aplica los conocimientos en la práctica
Demuestra capacidad de investigación
Desarrolla aplicaciones con base en la tecnología de paralelización</td><td>Elementos de competencia:
Programa aplicaciones utilizando hilos de ejecución para distribuir tareas a distintos servidores</td></tr>
</table>

Introducción

El ordenamiento por cubetas (bucket sort en inglés) es un algoritmo de ordenamiento que distribuye todos los elementos a ordenar entre un número finito de cubetas. Cada cubeta sólo puede contener los elementos que cumplan unas determinadas condiciones. Para esta práctica esas condiciones son intervalos de números. Las condiciones deben ser excluyentes entre sí, para evitar que un elemento pueda ser clasificado en dos cubetas distintas. Después, cada una de esas cubetas se ordena individualmente con otro algoritmo de ordenación (que podría ser distinto según la cubeta), o se aplica recursivamente este algoritmo para obtener cubetas con menos elementos. Se trata de una generalización del algoritmo Pigeonhole sort. Cuando los elementos a ordenar están uniformemente distribuidos la complejidad computacional de este algoritmo es de O(n).

El algoritmo contiene los siguientes pasos:

1. Crear una colección de cubetas vacías
2. Colocar cada elemento a ordenar en una única cubeta
3. Ordenar individualmente cada cubeta
4. Devolver los elementos de cada cubeta concatenados por orden

Recursos y/o materiales

- Manual de prácticas de laboratorio de Aplicaciones para Comunicaciones en Red
- Plumones
- Bibliografía
- Internet
- Computadora
- IDE de desarrollo
- Apuntes

Instrucciones

En esta práctica debes implementar el algoritmo de ordenamiento bucketsort en un modelo de hilos, haciendo uso de alguna técnica para el ordenamiento de cada cubeta.

Desarrollo de la práctica

A partir de los programas de hilos en C que te fueron proporcionados por el profesor deberás de implementar los siguientes programas:

- Implementar un programa en C que se le pase un número n que corresponderán a la misma cantidad de hilos que deberá de levantarse.
- Se generará una lista (o arreglo) con 3500 números enteros de forma aleatoria en el rango de 0 y 999, los cuales deberán ser ordenados usando el algoritmo bucketsort.
- Para esto, se crearán igual número de rangos de números desde 0 hasta 999. Es decir, si hay 5 servidores activos, genera 5 rangos que serían 0 a 199, 200 a 399, 400 a 599, 600 a 799 y 800 a 999.
- Agrupar los números creados aleatoriamente en cada uno de esos rangos.
- Lanzar igual número de hilos que se conectaran a cada uno de ellos mediante alguna técnica de comunicación entre hilos. No usar una memoria común.
- Enviara cada conjunto de números aleatorios a cada uno de ellos mediante la técnica elegida.
- Esperará que le regrese cada conjunto de números ordenados del menor al mayor, una vez que terminen todos los hilos, agrupara los números de forma ordenada y los guardará en un archivo para su revisión.