Documento de diseño de software para la segmentación de imágenes con K-Means con computación distribuida

Gabriel David Gonza Condori
Lesly Yaneth Mita Yagua
Flores Herrera, Jefferson
Pinto Medina, Brian
Supervised by: Milton Raul Condori Fernandez

27 de diciembre de 2021

1. Introducción

1.1. Objetivo

El presente documento de diseño de software describe el enfoque y la arquitectura para la segmentación de imágenes con K-means de manera distribuida para maximizar el rendimiento, minimizando así el tiempo de ejecución teniendo como hiperparámetros el número de procesadores (MPI) y el número de subprocesos considerados en la región paralela (OpenMP).

1.2. Alcance

El alcance del presente proyecto es desarrollar una solución distribuida para el problema de segmentación de imágenes utilizando K-means, para obtener resultados que superen en tiempo de ejecución obtenido de forma no distribuida, es decir, linealmente.

1.3. Visión general

El presente documento posee distintas secciones para el diseño de software que describe el enfoque y la arquitectura para la segmentación de imágenes con K-means de manera distribuida, entre las cuales son las siguientes:

- **Introducción:** Con el propósito de contextualizar el documento que se presenta de manera general.
- **Descripción general del sistema:** Con la función de proporcionar información a cerca de la funcionalidad, contexto y diseño.

- **Arquitectura del sistema:** En el cual se presentara el diseño de la arquitectura y justificación del diseño.
- **Diseño de datos:** En el cual se presentara la descripción y diccionario de datos.
- **Diseño de componentes:** Con el propósito de detallar lo que hace cada componente de una manera más sistemática.
- **Matriz de requisitos:** Con el propósito de identificar los componentes del sistema satisfacen cada uno de los requisitos funcionales.

1.4. Material de referencia

Se realizó una búsqueda de trabajos relacionados en la segmentación de imágenes con K-means de manera distribuida. Se encontró el artículo *The Study of Parallel K-Means Algorithm* [3] que propone una estrategia paralela al método de agrupamiento introduciendo el equilibrio de carga dinámico.

También se encontró el artículo *A Parallel K-Means Clustering Algorithm with MPI* [2] donde propone un algoritmo de agrupamiento de K-means paralelo con MPI llamado MKmeans, el cual permite aplicar el algoritmo de agrupación en grupoes de forma eficaz en el entorno paralelo.

Otro artículo encontrado es *Convex Hull Using K-Means Clustering in Hybrid (MPI/OpenMP) Environment* [1] donde posee un modelo híbrido que combina ambos enfoques de MPI y Open MP que utiliza para resolver el problema del casco convexo en un entorno paralelo. En este diseño los puntos 2D se agrupan en diferentes grupos y luego se calcula el casco convexo para cada uno de estos grupos; los puntos que definen estos cascos convexos se utilizan para construir el casco convexo final.

1.5. Definiciones y acrónimos

Se definirá los términos, acrónimos y abreviaturas que se encuentran para el presente documento de diseño de software mediante el Cuadro 1.

Termino	**Definición**
OpenMP	es una interfaz de programación de aplicaciones (API) para la programación multiproceso de memoria compartida en múltiples plataformas.
MPI	es una especificación para programación de paso de mensajes, que proporciona una librería de funciones para C, C++ o Fortran que son empleadas en los programas para comunicar datos entre procesos.

Cuadro 1: Tabla de definiciones y acrónimos.

2. Descripción general del sistema

La agrupación en grupoes es uno de los métodos más populares para el análisis de datos, que prevalece en muchas disciplinas como la segmentación de imágenes, la bioinformática, el reconocimiento de patrones y las estadísticas, etc. El algoritmo de agrupación más popular y simple es K-means debido a su fácil implementación, simplicidad, eficiencia y éxito empírico. Sin embargo, las aplicaciones del mundo real producen grandes volúmenes de datos, por lo tanto, cómo manejar de manera eficiente estos datos en una importante tarea de minería ha sido un problema importante y desafiante. A medida que la escala del conjunto de datos aumenta rápidamente, es difícil usar k-means para manejar una gran cantidad de datos.

La computación paralela es el uso de múltiples recursos de computación para resolver un problema computacional. Las computadoras paralelas se pueden clasificar aproximadamente como Multi-Core y Multi-Processor. En ambas clasificaciones, el hardware admite el paralelismo con un nodo de computadora que tiene múltiples elementos de procesamiento en una sola máquina. La programación paralela es la capacidad del programa para ejecutarse en esta infraestructura, lo que todavía es una tarea bastante difícil y compleja de lograr. Dos de los diferentes enfoques utilizados en el entorno paralelo son MPI y Open MP, cada uno de ellos tiene sus propios méritos y deméritos.

3. Arquitectura del sistema

3.1. Diseño de la arquitectura

El algoritmo de agrupación de K -Means es un algoritmo no supervisado y se utiliza para segmentar el área de interés del fondo. Agrupa o divide los datos dados en K-clusters o partes según los K-centroides. El algoritmo se utiliza cuando tiene datos sin etiquetar (es decir, datos sin categorías o grupos definidos). El objetivo es encontrar ciertos grupos basados en algún tipo de similitud en los datos con el número de grupos representados por K. Los pasos del algoritmo es el siguiente:

1. Elija el número de conglomerados K.
2. Seleccione al azar K puntos, los centroides (no necesariamente de su conjunto de datos).
3. Asigne cada punto de datos al centroide más cercano que forme K grupos.
4. Calcule y coloque el nuevo centroide de cada grupo.
5. Reasigne cada punto de datos al nuevo centroide más cercano. Si alguna reasignación. tuvo lugar, vaya al paso 4; de lo contrario, el modelo está listo.

Para una secuencia de pasos del nuevo algoritmo con paralelización tenemos en cuenta lo siguiente donde N puntos representados en el espacio con un vector $M - dim$, se tiene procesadores P definidos como (Nodo 0, Nodo 1,. . . , Nodo P-1), con K grupos y donde L número máximo de iteraciones a realizar.

Además, suponiendo que los puntos son independientes, un punto pertenece a uno y solo a un grupo K, el número de iteraciones $> L$.

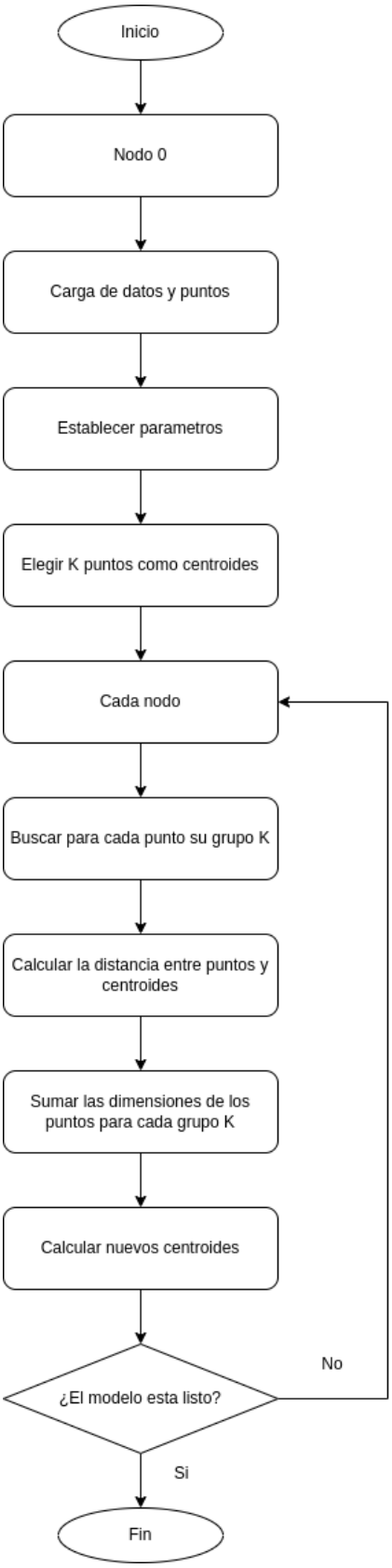

Figura 1: Diagrama de Flujo.

3.2. Justificación del diseño

El diseño de arquitectura anteriormente presentado presenta un flujo de trabajo que se muestra en la Figura 1 el cual se detallara a continuación:

1. El nodo 0 posee los datos y asigna puntos N/P a cada nodo. Los puntos restantes se asignan uno por uno.
2. El nodo 0 lee y establece los parámetros de configuración inicial
3. El nodo 0 elige K puntos como centroides iniciales y los transmite a los otros nodos
4. Cada nodo:
   - Para cada punto local, busque la pertenencia al grupo entre los K grupos
   - El cálculo de la distancia entre puntos y centroides se realiza en paralelo con OpenMP.
   - Para cada grupo, suma los valores de las dimensiones de los puntos.
5. Después de una operación MPI_Allreduce, cada nodo conoce el número de puntos y la suma de sus valores dentro de cada grupo. Calcular nuevos centroides.
6. Se continuará la iteración desde el punto 4 hasta que finalice.

4. Diseño de datos

4.1. Descripción de datos

La segmentación divide una imagen en regiones con propiedades internas coherentes. Se puede segmentar una imagen utilizando el color. Donde el objetivo con el que se agrupan los píxeles es para separar los elementos significativos de una imagen y así poder extraer cierta información de alguno de ellos.

4.2. Diccionario de datos

La estructura de datos que se utiliza en la segmentación debe contener cada píxel de la imagen asociado a su valor numérico en su escala.

5. Diseño de componentes

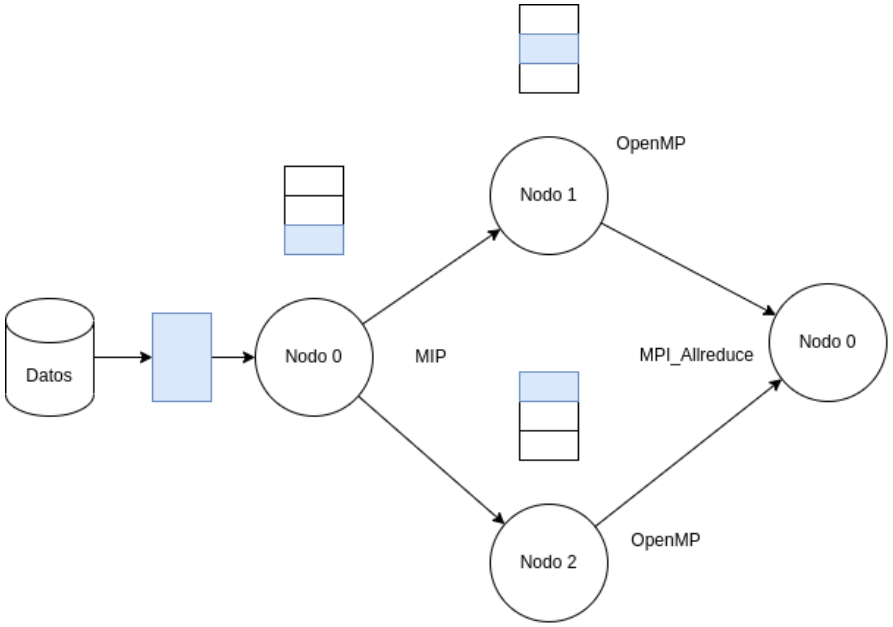

Figura 2: Diagrama estructural.

6. Matriz de requisitos

Los componentes del sistema que satisfacen cada uno de los requisitos funcionales son los siguientes:

- Poseer 3 Nodos para la distribución de datos y procesos que se ejecutaran dentro de ellos.
- Características detalladas de cada computador.
- Una Red LAN entre las computadoras para la comunicación MIP entre ellas.
- Cada computador deberá tener instalado y funcionando correctamente OpenMP.

Referencias

[1] V. N. Waghmare and D. B. Kulkarni, "Convex hull using k-means clustering in hybrid (mpi/openmp) environment," in *2010 International Conference on Computational Intelligence and Communication Networks*, 2010, pp. 150–153.

[2] J. Zhang, G. Wu, X. Hu, S. Li, and S. Hao, "A parallel k-means clustering algorithm with mpi," in *2011 Fourth International Symposium on Parallel Architectures, Algorithms and Programming*, 2011, pp. 60–64.

[3] Y. Zhang, Z. Xiong, J. Mao, and L. Ou, "The study of parallel k-means algorithm," in *2006 6th World Congress on Intelligent Control and Automation*, vol. 2, 2006, pp. 5868–5871.