Informe Trabajo Paralelo - Computación Paralela y Distribuida

Trabajo Paralelo
Computación Paralela y Distribuida
Universidad Tecnológica Metropolitana
Profesor Sebastián Salazar Molina
Fecha de entrega: 2 de Julio de 2025

Integrantes:
Javier Villalobos Valle
Cristóbal Pérez Bustos

Índice

1. Introducción

En el presente informe se desarrolla un problema, el cuál se debe resolver utilizando paralelismo. El problema consiste en leer un fichero CSV, que contiene la siguiente información, donde cada campo está delimitado por comillas dobles (“”) y separado por punto y coma ‘;’:

1. **IDENTIFICADOR:** Representa el RUT del ciudadano.
2. **ESPECIE:** Tipo de especie dentro del reino, pueden ser:
   a) Humana.
   b) Elfica.
   c) Enana.
   d) Hombre Bestia.
3. **GÉNERO:** La separación sexual dentro de la especie, o como se identifican pueden ser:
   a) Macho.
   b) Hembra.
   c) Otro.
4. **NOMBRE:** Indica el primer nombre del ciudadano.
5. **APELLIDO:** Indica el apellido del ciudadano.
6. **FECHA NACIMIENTO:** Representa la fecha y hora del nacimiento en formato AÑO-MES-DÍATHORA:MINUTO:SEGUNDO, ejemplo: 1991-10-17T15:27:07
7. **CP ORIGEN:** Es el código postal asociado al lugar en que vive la persona.
8. **CP DESTINO:** Es el código postal asociado al lugar que más frecuentemente la persona debe viajar.

Para el desarrollo del problema se utilizó procesamiento en bloques, de $1{,}000{,}000$ líneas, donde en cada bloque se realizó de manera paralela la lectura, extracción y el procesamiento de cada dato necesario. Finalmente se muestra el total de datos y los resultados de las preguntas 1 a la 7 en consola, mientras que la respuesta correspondiente a la pregunta 8 se encontrará en un archivo TXT, llamado “Top 10000 poblados”.

En cuanto al procesamiento de los datos, se crearon funciones auxiliares, las cuales nos permitieron segmentar las tareas requeridas para responder a las preguntas propuestas.

2. Desarrollo

En el desarrollo de la solución se implementaron 2 técnicas relevantes para optimizar el procesamiento en cuanto tiempo de ejecución y memoria, estas técnicas son:

- **Procesamiento en bloques:** Consiste en segmentar datos en bloques de menor tamaño, para su procesamiento.
- **Paralelismo:** Consiste en aprovechar al máximo la cantidad de procesadores ó núcleos de un equipo, para realizar de manera paralela la misma tarea para distintos datos o realizar de manera simultanea diferentes tareas.

El procesamiento en bloques se utilizó para la leer y procesar los datos del fichero CSV, ya que este archivo al ser grande ($8{,}8$ GB), puede generar problemas en el uso de memoria del equipo, por lo que se asignó un tamaño de $1{,}000{,}000$ de lineas a cada bloque.
Dentro de cada bloque, de manera paralela se realizó la lectura y procesamiento de cada registro. Al comienzo del programa “main.cpp”, se explicitó que se utilizará el máximo de núcleos disponibles para el procesamiento, utilizando la siguiente sentencia:

```
omp_set_num_threads(omp_get_max_threads());
```

Para la paralelización, se utilizó la librería OpenMP.
En cuanto a la sección paralela del código, esta sección se implementa dentro del bloque de procesamiento, donde existen variables locales correspondientes a cada bloque; fuera del bloque de procesamiento existen las variables globales que contendrán la información final de la solución; dentro del bloque de procesamiento, se lee, procesa y guarda la información dentro de las variables locales correspondientes, todo lo anterior sucede registro por registro de manera paralela.
Una vez que se termina de procesar el bloque, las variables locales son almacenadas en las variables globales correspondientes, para su procesamiento final, mientras que las variables locales se restablecen.

En cuanto al procesamiento de los datos, se crearon funciones auxiliares, algunas de estas son:

```
int CalcularEdad(const string& birthDay);
float EdadPromedio(std::vector <int>& edades, int& count);
float EdadMediana(vector <int>& edades);
```

Todas las funciones creadas se encuentran comentadas, en el archivo “UtilsFunctions.h”. Cabe mencionar que una de las funciones que se creó utiliza paralelismo, esta función es ‘SegmentarEdad()’; se debe tener precaución al utilizar esta función dentro de un entorno paralelo, ya que puede generar conflictos, si es que no se encuentra correctamente configurado el entorno paralelo.

Por último, una vez que se finalizó la lectura y procesamiento de los datos desde el archivo CSV, se termina también el procesamiento en bloques.
Posterior a lo mencionado anteriormente, se sigue con el procesamiento de las variables globales, para finalizar con la impresión vía consola de las respuesta a las preguntas propuestas desde la $1 - 7$, ya que la respuesta a la pregunta 8 se encontrará en un archivo TXT, en la misma ubicación que todos los archivos.

3. Conclusión

En conclusión, se logró de manera exitosa el procesamiento de los datos, obteniendo un tiempo de ejecución de aproximadamente 4 minutos y 36 segundos en un equipo con 4 núcleos y 8 GB de RAM.