



UNIVERSIDAD
TECNOLÓGICA
METROPOLITANA
del Estado de Chile

Informe Examen.

Computación Paralela y Distribuida

Profesor: Sebastián Salazar Molina

Sección 411

Integrantes:

- Cristian Montecinos Fuenzalida

Fecha de entrega: Martes 20 de diciembre de 2022

CONTENIDO

Introducción	3
Marco Teórico	4
Openmp	4
Regresión lineal	4
Planteamiento del problema	6
Desarrollo del problema	8
Conclusión	10

Introducción

La paralelización es una técnica utilizada en programación para mejorar el rendimiento de una aplicación al dividir un proceso en múltiples partes que pueden ser ejecutadas de forma concurrente en diferentes hilos de ejecución. Esto permite que la aplicación aproveche al máximo el potencial de procesamiento de la máquina en la que se ejecuta.

En C++, existen diferentes formas de implementar la paralelización, una de ellas es utilizando la directiva `#pragma omp parallel for`. Esta directiva indica que el bucle `for` que le sigue debe ejecutarse de forma paralela utilizando múltiples hilos de ejecución. Es importante tener en cuenta que la paralelización no es adecuada para todos los casos y puede tener un costo de overhead que puede reducir el rendimiento en lugar de mejorarlo.

La regresión lineal es una técnica utilizada en estadística para modelar la relación entre dos variables. Se trata de una línea recta que se ajusta lo mejor posible a un conjunto de puntos en un diagrama de dispersión. La regresión lineal se puede utilizar para hacer predicciones sobre una variable a partir de un valor conocido de otra variable. Por ejemplo, se puede utilizar para predecir el número de accidentes de tráfico en una ciudad en una fecha futura, utilizando como variable independiente el número de días transcurridos desde una fecha inicial y como variable dependiente el número de accidentes.

El código a presentar en este informe define dos funciones: `fecha_a_segundos` y `regresionLineal`. La primera función toma como argumentos tres punteros a caracteres, que corresponden a un año, mes y día, respectivamente, y devuelve la fecha dada en segundos utilizando la librería `ctime`. La segunda función realiza una regresión lineal utilizando dos vectores de enteros `x` e `y`, y un contador que indica el tamaño de ambos vectores. También recibe un flotante `segundos`, que corresponde a la fecha en segundos ingresada por el usuario para hacer la predicción. La función utiliza un bucle `for` con una directiva `#pragma omp parallel for` que indica que el bucle debe ejecutarse de forma paralela utilizando múltiples hilos de ejecución. Finalmente, la función devuelve el resultado de la regresión lineal.

Marco Teórico

Openmp

OpenMP es una especificación de la plataforma para el desarrollo de programas paralelos en C, C++ y Fortran. Se trata de una interfaz de programación de aplicaciones (API) que permite a los programadores escribir código que pueda ser ejecutado de forma concurrente en múltiples procesadores o núcleos de procesadores. OpenMP utiliza el modelo de programación "shared memory" (memoria compartida), lo que significa que los procesadores comparten un espacio de memoria común y pueden acceder a él simultáneamente.

OpenMP proporciona una serie de directivas y funciones que se pueden utilizar para paralelizar código de forma sencilla. Una de estas directivas es la que se utiliza en el código que has proporcionado, `#pragma omp parallel`, que indica que el bloque de código siguiente debe ejecutarse de forma paralela utilizando múltiples hilos de ejecución. OpenMP es ampliamente utilizado para acelerar la ejecución de código en aplicaciones de ciencia de datos, ingeniería y otros campos donde se requiere un alto rendimiento computacional.

Regresión lineal

La regresión lineal es una técnica utilizada en estadística para modelar la relación entre dos variables. Se trata de una línea recta que se ajusta lo mejor posible a un conjunto de puntos en un diagrama de dispersión. La regresión lineal se puede utilizar para hacer predicciones sobre una variable a partir de un valor conocido de otra variable.

La regresión lineal se basa en la relación matemática entre dos variables X e Y:

$$Y = a + bX$$

En esta ecuación, a es la ordenada al origen (también conocida como el intercepto) y b es la pendiente de la recta. Estos dos parámetros se determinan ajustando la recta de forma que minimice la suma de los cuadrados de los errores entre los valores observados y los valores predichos por la recta.

Una vez que se ha ajustado la recta de regresión, se puede utilizar para hacer predicciones sobre Y dado un valor conocido de X . Por ejemplo, si se quiere predecir el número de accidentes de tráfico en una ciudad en una fecha futura, se puede utilizar la regresión lineal para encontrar la relación entre el número de días transcurridos desde una fecha inicial y el número de accidentes, y luego utilizar esta relación para hacer una predicción sobre el número de accidentes en una fecha futura dada.

Planteamiento del problema

Una pequeña empresa (25 trabajadores), dedicada al rubro de la minería, trabaja manipulando materiales peligrosos, a pesar de las medidas de seguridad, siempre hay accidentes, que aunque no sean graves, disminuyen la fuerza laboral y afectan a la formación de los equipos.

Para su análisis, se adjunta un csv con los datos históricos, desde que se iniciaron las operaciones (06/10/2017) hasta la fecha de corte (10/12/2022):

- Fecha. Esta es la fecha en la que sucedió el accidente.
- Cantidad de accidentes. Es el número de accidentes ocurridos.

Se solicita desarrollar una aplicación que, dada una fecha futura, pueda predecir la cantidad de accidentes esperados para el día de entrada. Esta aplicación debe implementar un modelo matemático, que use los datos históricos, esté codificada en C o C++ y use OpenMP o MPI para maximizar el rendimiento.

Entrada.

La aplicación debe leer la fecha como el primer argumento de la línea de comandos, el formato de fecha corresponde al ISO 8601.

Resultado.

La salida del programa, debe mostrar en pantalla:

- La fecha (en formato ISO 8601).
- La cantidad de accidentes esperados.
- Un salto de línea.

Una vez entregue el resultado, el programa debe terminar.

Código.

La fecha límite de entrega es el martes 20 de diciembre de 2022 a las 17:59:59 hora continental de Chile. Se debe trabajar en un repositorio Github y debe entregar los enlaces del proyecto.

EVALUACIÓN

Documentación.

Parte de la evaluación consiste en la documentación de las funciones. Que debe ser clara, concisa y descriptiva de lo que el código realiza.

Código

El código debe ser claro, fácil de leer, ordenado y cumplir con buenas prácticas de programación, se inspeccionará el código fuente.

Resultados.

Un criterio de evaluación que se tomará en consideración: el tiempo de ejecución de la tarea. Menos es mejor.

Desarrollo del problema

El código para el desarrollo de este problema se incluye las siguientes librerías:

- `iostream`: permite la entrada y salida de datos por consola.
- `fstream`: permite la lectura y escritura de archivos.
- `sstream`: permite la manipulación de cadenas de caracteres como si fueran flujos de datos.
- `vector`: permite el uso de vectores (arrays dinámicos).
- `ctime`: permite el uso de funciones relacionadas con el tiempo y las fechas.
- `cstring`: permite el uso de funciones relacionadas con cadenas de caracteres.
- `cmath`: permite el uso de funciones matemáticas.
- `omp.h`: permite el uso de directivas de OpenMP, una API de paralelización para C++.

El código define dos funciones: `fecha_a_segundos` y `regresionLineal`. La primera función toma como argumentos tres punteros a caracteres, que corresponden a un año, mes y día, respectivamente, y devuelve la fecha dada en segundos. Para ello, primero crea un objeto `tm` con la fecha dada. Luego convierte este objeto a un objeto `time_t` utilizando la función `mktime`. Finalmente, obtiene el número de segundos transcurridos desde la medianoche del 1 de enero de 1970 utilizando la función `time`.

La segunda función realiza una regresión lineal utilizando dos vectores de enteros `x` e `y`, y un contador que indica el tamaño de ambos vectores. También recibe un flotante segundos, que corresponde a la fecha en segundos ingresada por el usuario para hacer la predicción. La función utiliza un bucle `for` con una directiva `#pragma omp parallel for` que indica que el bucle debe ejecutarse de forma paralela utilizando múltiples hilos de ejecución.

Dentro del bucle, se realizan las siguientes operaciones:

- Se suma el valor de la variable `y[i]` a la variable `sumaAccidente`.
- Se suma el producto de las variables `x[i]` e `y[i]` a la variable `sumatoriaXY`.
- Se suma el cuadrado de la variable `y[i]` a la variable `sumatoriaXCuadrado`.

- Se suma el cuadrado de la variable $x[i]$ a la variable sumaYPromAccidente.
- se suma la variable $x[i]$ a la variable sumaFecha.

Una vez que se ha terminado de recorrer el bucle, se calculan los siguientes valores:

promedioAccidente: es el promedio de los valores de y . Se calcula dividiendo la variable sumaAccidente entre el contador.

promFecha: es el promedio de los valores de x . Se calcula dividiendo la variable sumaFecha entre el contador.

pendiente: es la pendiente de la recta de regresión. Se calcula utilizando la fórmula $(\text{sumatoriaXY} - (\text{contador} * \text{promedioAccidente} * \text{promFecha})) / (\text{sumaYPromAccidente} - (\text{contador} * \text{promFecha} * \text{promFecha}))$.

El código principal del programa comienza con la apertura de un archivo llamado datos_examen.csv utilizando un objeto ifstream llamado miArchivo.

Se declaran dos vectores de enteros llamados x e y , un entero contador y un flotante segundos. Se utiliza un bucle while para leer cada línea del archivo hasta que se alcance el final del archivo. En los vectores x e y , se almacenan las fechas en segundos utilizando la función fecha_a_segundos, y los accidentes en el parametro x .

Una vez que se han leído todas las líneas del archivo, se cierra el archivo y se le pide al usuario que ingrese una fecha en formato iso 8601 se valida un formato correcto.

luego se utiliza la función regresionLineal donde se le pasan los paramatros x , y , contador y segundos que es la fecha transformada a segundos para poder realizar la regresion lineal, luego se imprime por pantalla la cantidad de accidentes predecidos y la fecha en la priso 8601.

Conclusión

En conclusión, el código es un programa en C++ que realiza una regresión lineal utilizando dos vectores de datos y un valor de tiempo. El programa comienza leyendo un archivo .csv que contiene una serie de filas con tres campos: una fecha en formato dd/mm/yyyy, un entero y otro entero. Cada fila se procesa convirtiendo la fecha a segundos, almacenando los dos enteros en vectores y incrementando un contador.

La función `regresionLineal` incluye un bucle `for` que recorre los elementos de los vectores de datos y calcula varios valores necesarios para realizar la regresión lineal. Además, incluye una directiva `#pragma omp parallel for` que indica que el bucle debe ejecutarse de forma paralela utilizando múltiples hilos de ejecución. Esto permite que el cálculo de los valores necesarios para la regresión lineal se realice de forma más rápida utilizando varios núcleos de procesador.

Una vez que se han leído todas las filas del archivo y se han calculado los valores necesarios para la regresión lineal, se llama a la función `regresionLineal` pasándole como argumentos los vectores de datos, el contador y el valor de tiempo en segundos. La función realiza la regresión lineal y devuelve el resultado, que es impreso en pantalla. En resumen, el programa proporcionado realiza una regresión lineal utilizando paralelismo para acelerar el cálculo de los valores necesarios.