

# Paralelismo en la predicción de lluvias en subregiones del Sahara: clusterización y comunicación.

Marco Antonio Franco Montoya [mfranc18@eafit.edu.co](mailto:mfranc18@eafit.edu.co)

Julián Ricaurte Hoyos [jricaur1@eafit.edu.co](mailto:jricaur1@eafit.edu.co)

[https://github.com/jricaur1/Proyecto4\\_HPC](https://github.com/jricaur1/Proyecto4_HPC)

## Contenido

|  |   |
|--|---|
| Paralelismo en la predicción de lluvias en subregiones del Sahara: clusterización y comunicación. .... | 1 |
| 1. Ideas Preliminares de Proyecto .....  | 2 |
| 2. Problema o Caso de estudio .....  | 2 |
| 3. Objetivos y alcance .....   | 2 |
| 4. Requerimientos técnicos .....   | 2 |
| 5. Plan de trabajo.....  | 2 |
| 6. Análisis del Problema .....   | 2 |
| 7. Algoritmo Secuencial.....   | 4 |
| 8. Algoritmo Paralelo (PCAM).....  | 5 |
| a. Particionamiento.....   | 5 |
| b. Comunicación.....   | 5 |
| c. Aglomeración .....  | 5 |
| d. Mapping .....   | 6 |
| 9. Eficiencia y Cálculos .....   | 6 |
| 10. Referencias .....  | 6 |

## **1. Ideas Preliminares de Proyecto**

- a. Optimizar la predicción de lluvias en el Sahara (IBM)

## **2. Problema o Caso de estudio**

Las poblaciones vulnerables en las regiones Contiguas al desierto del Sahara se encuentran con lluvias escasas durante el año. De estas depende la efectividad o fracaso de sus cosechas. Es por esto que requieren herramientas que permitan determinar con mayor precisión las temporadas de lluvia, para así garantizar la alimentación de sus pueblos.

## **3. Objetivos y alcance**

- a. Realizar predicciones sobre potenciales lluvias en regiones cercanas al Sahara en un rango no mayor a 72 horas
- b. Utilizar herramientas de cómputo paralelo como OpenMP para la implementación de hilos y MPI para la comunicación entre múltiples Clusters que procesen los datos.
- c. Optimizar código secuencial que ejecute la misma función con el objetivo de comparar sus diferencias.

## **4. Requerimientos técnicos**

- a. Clúster de Colfax.
- b. OpenMP Framework.
- c. MPI.
- d. Lenguaje de programación C++.

## **5. Plan de trabajo**

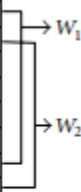
- a. Análisis del problema.
- b. Implementación secuencial.
- c. Diseño de la paralelización.
- d. Pruebas de optimización y speed up.
- e. Evaluación de resultados.

## **6. Análisis del Problema**

Existen múltiples acercamientos a la problemática del clima dentro de la computación, especialmente por las potenciales variaciones del clima y los diversos factores implicados, que pueden volver predicciones fuera de un rango de 10 días, Inutilizables. Por lo anterior, es común encontrar implementaciones que utilizan lógica difusa [5], Redes neuronales [4] con múltiples componentes que aumentan su complejidad, y modelos estadísticos [2] que aunque complejos, pueden omitir aspectos a considerar dependiendo del nivel de abstracción utilizado en el modelo.

En este caso, se implementará una solución que haga uso de un modelo estadístico que considere las condiciones climáticas de la semana anterior, además de las de las 2 semanas anteriores, a través de un algoritmo de ventana deslizante, que permitirá determinar las fechas que más se acerquen al comportamiento de los componentes del clima a analizar de la semana actual.

| S. No. | Max temp. | Min temp. | Humidity | Rainfall |
|--------|-----------|-----------|----------|----------|
| 1      |           |           |          |          |
| 2      |           |           |          |          |
| 3      |           |           |          |          |
| 4      |           |           |          |          |
| 5      |           |           |          |          |
| 6      |           |           |          |          |
| 7      |           |           |          |          |
| 8      |           |           |          |          |
| 9      |           |           |          |          |
| 10     |           |           |          |          |
| 11     |           |           |          |          |
| 12     |           |           |          |          |
| 13     |           |           |          |          |
| 14     |           |           |          |          |



Las fechas más cercanas pueden encontrarse a través de una diferencia euclidiana entre los componentes a considerar dentro del modelo determinado.

$$d_E(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}.$$

[https://wikimedia.org/api/rest\\_v1/media/math/render/svg/8f8d3e4756830488da4121a15b489cc481e4b58f](https://wikimedia.org/api/rest_v1/media/math/render/svg/8f8d3e4756830488da4121a15b489cc481e4b58f)

Al encontrar este valor, se puede determinar cuál matriz dentro de las ventanas, tiene el mayor acercamiento a la matriz actual, que denominaremos CD.

Al obtener estos componentes, se puede ejecutar una diferencia simple para determinar la relación entre la semana anterior y la actual. Al tener valores que pueden ser cero, ejecutar una diferencia porcentual no es posible. Después de esto se requiere encontrar una media para cada componente de las semanas y entre ellas para así encontrar la asociación entre la semana anterior y la actual. Este dato podrá asociarse al día anterior al solicitado, para generar así la predicción. El algoritmo puede definirse de la siguiente manera:

1. Se obtiene la matriz de los 7 días anteriores a la fecha a predecir (CD), retornando una matriz de 7x4
2. Se obtiene la matriz de los 14 días del año anterior a la fecha a predecir (PD) obteniendo una matriz de 14x4
3. Se generan 8 ventanas deslizantes usando la matriz PD, (W1, W2 ... W7, W8)
4. Se obtiene la distancia euclidiana entre las matrices de las ventanas y la matriz CD (ED1, ED2 ... ED7, ED8)

5. Se selecciona la matriz ventana con la menor distancia euclidiana
6. Para cada componente del clima:
  - a. Se encuentra su vector de diferencia en cd  $ABS(CD_{i+1j} - CD_{ij})$ , Obteniendo una matriz de 6x1 denominada VC
  - b. Se encuentra su vector de diferencia en pd  $ABS(PD_{i+1j} - PD_{ij})$ , Obteniendo una matriz de 6x1 denominada VP.
  - c. Se encuentra la media del vector VC, denominada MVC
  - d. Se encuentra la media del vector VP, denominada MVP
  - e. Se encuentra la media general  $Mean = (MVC + MVP)/2$
  - f. Se suma este componente al componente de la fecha anterior.

A través de este proceso se encuentra una aproximación de los componentes a analizar en este caso. Por motivos de facilidad, se tomaron los datos de National Oceanic and Atmospheric Administration (NOAA), para la ciudad de Washington en su estación GHCND:USC00450008, utilizando un rango de fechas desde el 12/05/2013 hasta 12/05/2020, fecha en la que se cierra dicha estación.

## 7. Algoritmo Secuencial

Es importante resaltar que el algoritmo secuencial expresado anteriormente fue tomado del paper [Weather Forecasting Using Sliding Window Algorithm](#), referenciado en la parte inferior del algoritmo. Sin embargo, su implementación fue desarrollada por los estudiantes que presentan este trabajo.

Para la implementación secuencial de este algoritmo se utilizó el lenguaje C++, creando 2 clases: Main y Prediction. En la clase Main se encuentra la estructura general del algoritmo, con la lectura de datos y su almacenamiento en vectores. A través del método main, se llaman los métodos de la clase Prediction, que ejecutan funciones como la distancia euclidiana, entre otras.

Cada función hace uso de vectores para obtener los datos requeridos para este proceso, y se retornan resultados de la misma manera. Pensando en el futuro de la implementación paralela, se reemplazaron las instancias en las cuales una función requería valores de la iteración anterior, utilizando un equivalente a través de la semana siguiente. Por ejemplo, para encontrar la diferencia entre un componente actual y el anterior, se encuentra la diferencia entre uno actual y el componente futuro, evitando así la incertidumbre que podría causar errores al paralelizar el código.

Para probar este algoritmo se utilizan 3 meses de fechas, desde el primero de enero de 2020 hasta el final del mes de marzo, almacenadas en el csv llamado sample. Una vez se obtienen estos resultados, se toma el tiempo de ejecución de este algoritmo, obteniendo como resultado:

```
jricauri@DESKTOP-FCHV13A:/mnt/c/Users/ASUS/Documents/Ubuntu/ST0263_ProyectoHPC/Proyecto4_HPC/src$ ./main
finished computation at Thu Jun  4 23:33:25 2020
elapsed time: 7.93672s
```

Cabe resaltar que este tiempo puede variar dependiendo de la cantidad de tareas ejecutándose en la máquina actualmente. Si es un equipo personal, se pueden encontrar variaciones al hacer streaming de la pantalla, entre otras tareas que demanden recursos.

Para más información sobre el código implementado, puede referirse al [repositorio](#) creado para este proyecto.

## 8. Algoritmo Paralelo (PCAM)

Para el algoritmo paralelo se hace uso exclusivo de OpenMP para las funciones de la clase Prediction, en este caso a través de estos métodos, se encuentra el ciclo óptimo a paralelizar y se ejecuta el proceso, aclarando en los puntos necesarios las tareas sincrónicas que se deben ejecutar a través de Mutex. En el [repositorio](#) se puede evidenciar el proceso de este código.

Una vez se genera el ejecutable optimizado y sus respectivos reportes para cada clase, debe admitirse como un trabajo dentro de cloudfax. Esto puede presentar errores si solamente se provee el ejecutable generado. Para facilitar el proceso de ejecución se creó un archivo appjob que permite añadir con facilidad a la cola la tarea a ejecutar.

Cuando se ejecuta el código se evidencia una mejora en tiempo de ejecución comparado con los 3 segundos expresados anteriormente. esto puede observarse en el archivo appjob.o<id del job>

### a. Particionamiento

Los datos a analizar se segmentan en cada una de las funciones a través de los procesos como el Sliding window ejecutado. Para este proceso cada uno de los métodos que usa vectores fue vectorizado y hace uso de OpenMP en cada for ejecutado. Esto puede observarse especialmente en el archivo prediction.cc

### b. Comunicación

La comunicación entre procesos ejecutados la realiza OpenMP de forma transparente haciendo uso de hilos. Dado que todos los datos se encuentran en memoria compartida, se puede acceder con facilidad a los datos, a menos que se requiera un proceso de escritura pues en estos casos se ejecuta un Mutex para garantizar una gestión sincrónica de los datos. En este caso no fue posible finalizar la implementación haciendo uso de MPI para el paso de mensajes.

### c. Aglomeración

La aglomeración de datos es realizada a través de matrices después del proceso de Sliding window. En este, gracias a los vectores, contamos con varias matrices resultantes de las Sliding windows y tenemos que empezar a computar las

predicciones. Este proceso se hace cuando encontramos la distancia euclideana, la varianza, la media varianza y la aproximación.

#### d. Mapping

Dado que solo hacemos uso de OpenMP y no usamos OpenMPI, el mapping lo maneja OpenMP y esto se puede ver evidenciado en el reporte de salida ipo\_out.optrpt.

## 9. Eficiencia y Cálculos

|    |          |  |    |          |
|----|----------|--|----|----------|
| TP | 0.215282 |  | tt | 0.01328  |
| T1 | 0.849942 |  | sp | 0.061688 |
| P  | 64       |  | ep | 0.000964 |

## 10. Referencias

- <https://www.worldcommunitygrid.org/discover.action>
- <https://www.worldscientific.com/doi/abs/10.1142/S0129053393000049>
- <https://doi.org/10.1155/2020/814837>
- <https://www.mpi-forum.org/>
- <https://www.openmp.org/>
- <https://scijinks.gov/forecast-reliability/#:~:text=A%20seven%2Dday%20forecast%20can,90%20percent%20of%20the%20time.&text=Since%20we%20can't%20collect,assumptions%20to%20predict%20future%20weather.>
- <https://www.ncdc.noaa.gov/>
- <https://www.hindawi.com/journals/isrn/2013/156540/>
- <https://ieeexplore.ieee.org/abstract/document/1384579>
- [https://www.researchgate.net/profile/Zuraidi\\_Saad/publication/221258399\\_Weather\\_Forecasting\\_Using\\_Photovoltaic\\_System\\_and\\_Neural\\_Network/links/57037ea608aea09bb1a3d96d/Weather-Forecasting-Using-Photovoltaic-System-and-Neural-Network.pdf](https://www.researchgate.net/profile/Zuraidi_Saad/publication/221258399_Weather_Forecasting_Using_Photovoltaic_System_and_Neural_Network/links/57037ea608aea09bb1a3d96d/Weather-Forecasting-Using-Photovoltaic-System-and-Neural-Network.pdf)
- <https://www.sciencedirect.com/science/article/abs/pii/S0165011488901236>
- <https://ark.intel.com/content/www/es/es/ark/products/94033/intel-xeon-phi-processor-7210-16gb-1-30-ghz-64-core.html>

### Código de Honor:

Cada uno de los autores, debe declarar explícitamente que el trabajo es original y cuál fue su aporte en el desarrollo del proyecto 4, o si es copiado de algún sitio en internet (porque hay muchas

implementaciones de este problema) deberá referenciar o citar el sitio de donde tomó el trabajo y declarar entonces cuál fue su aporte con esta copia en el proyecto4.

Esta declaración debe ser colocada en README.md del github del proyecto4 por cada autor.