

Trabajo práctico II

Comunicación de procesos en computación paralela

Introducción

Los niveles de integración electrónica han permitido la generación de procesadores de arquitecturas *multiprocesos*, *multicore* y ahora *many integrated core* (MIC). Este avance hace necesario que los programadores cuenten con un profundo conocimiento del hardware sobre el que se ejecutan sus programas, y que dichos programas ya no pueden ser monocore.

Entre las técnicas y estándares más utilizados para sistemas de memoria compartida y memoria distribuida, se encuentra OpenMP y MPI respectivamente.

Objetivo

El objetivo del presente trabajo práctico es que el estudiante sea capaz diseñar una solución que utilice el paradigma de memoria distribuida, utilizando OpenMP.

Desarrollo

Se dispone de un radar meteorológico que opera en Banda C (5,625 Ghz), al cual se le desea introducir un procesador Doppler que permita calcular la componente radial del campo velocidad. La palabra radar, es un acrónimo de **radio detection and ranging**.

El funcionamiento de un radar, de forma muy resumida y simplificada, consiste en la emisión de una onda electromagnética, que recorre el espacio hasta encontrarse con un blanco. Este blanco, al ser inducido por la onda, se comporta como un espejo, que hace retornar la señal

a la antena que emitió la onda. Esta señal se procesa y es permite extraer información del blanco. El funcionamiento del mismo se observa en la Ilustración 1.

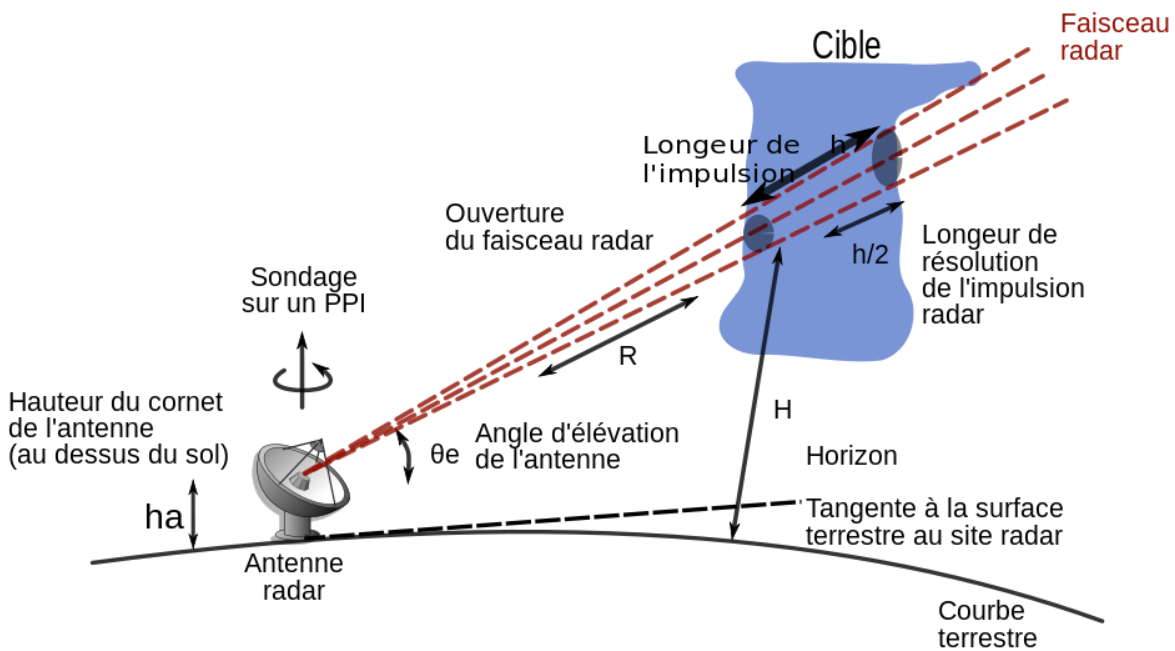


Ilustración 1 - Radar Meteorológico

La antena gira sobre su eje horizontal en 360° (una vuelta completa) y la posición de la antena sobre este eje se denomina *acimut*. A su vez, puede moverse en su eje vertical, más comúnmente entre 0 y 90° , que se denomina *elevación*.

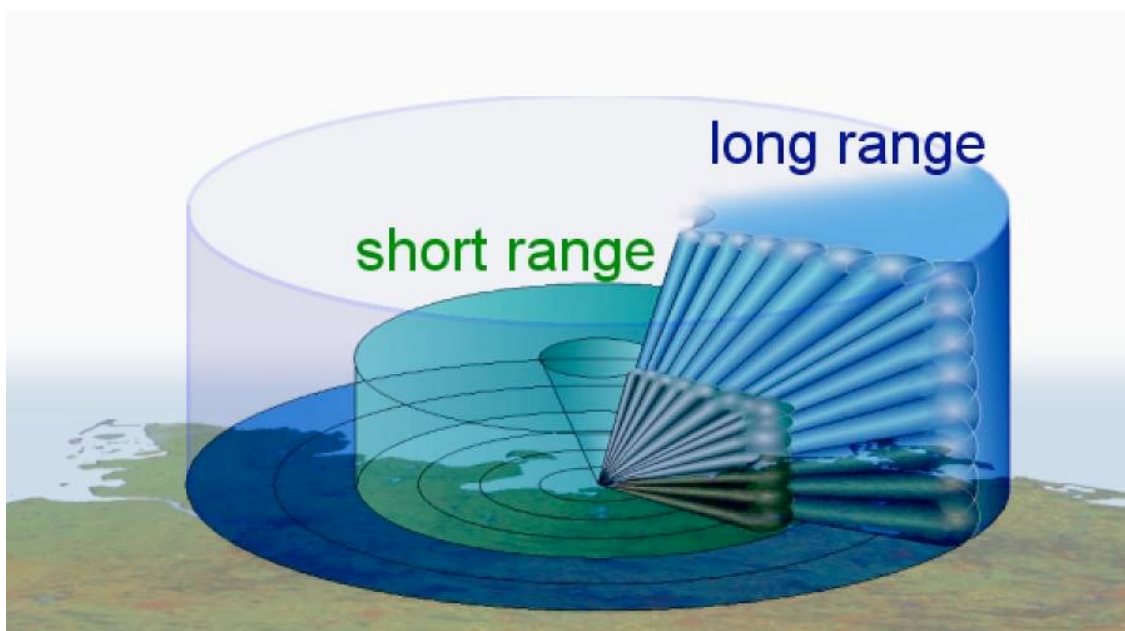


Ilustración 2 - Movimientos en elevación y acimut

El radar con que se desea trabajar del tipo pulsado, posee un transmisor que genera pulsos electromagnéticos en doble polarización, es decir, transmite un pulso con polarización horizontal (0°) y otro con polarización vertical (90°), en forma conjunta. Cada canal se denomina H y V respectivamente. A su vez, recordemos que una señal electromagnética es compleja, y esta se puede descomponer en sus componentes en fase y en cuadratura ($I + jQ$).

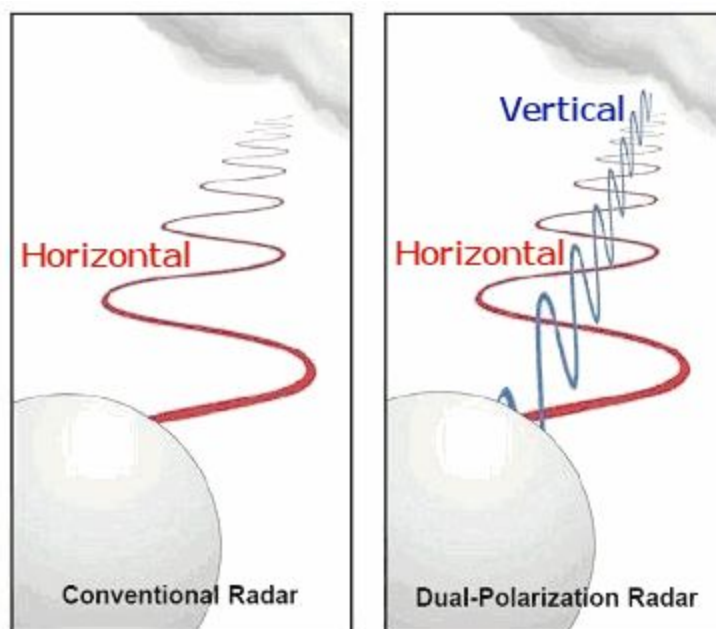


Ilustración 3: Pulso de doble polarización

El radar posee una resolución de 0,5 km en rango, y un rango máximo de cobertura de 250km de radio. A cada componente de rango se le denomina *gate*.

Se dispone de un archivo binario (pulsos.iq), que contiene la salida del conversor analógico digital (ADC), que está ubicado en la entrada del Receptor Digital. El mismo digitaliza el canal H y el canal V, datos que se denominan N1 (Nivel 1). El archivo está conformado por N muestras de pulsos, emitidos a elevación 0,5° y desde los 83° hasta los 90° en acimut, con el RMA 1, ubicado en la UNC. Por grado en acimut se tienen 100 pulsos emitidos.

La estructura de cada pulso, se muestra en la siguiente tabla.

Variable	Tipo	Descripción
validSamples	uint16_t	Cantidad de muestras válidas por canal polarimétrico
V_I[1]	float	Muestra 1 polaridad vertical en fase
V_Q[1]	float	Muestra 1 polaridad vertical en cuadratura

V_I[2]	float	Muestra 2 polaridad vertical en fase
V_Q[2]	float	Muestra 2 polaridad vertical en cuadratura
V_I[..]	float	
V_Q[..]	float	
V_I[validSamples]	float	Muestra <i>validSamples</i> polaridad vertical en fase
V_Q[validSamples]	float	Muestra <i>validSamples</i> polaridad vertical en cuadratura
H_I[1]	float	Muestra 1 polaridad horizontal en fase
H_Q[1]	float	Muestra 1 polaridad horizontal en cuadratura
H_I[2]	float	Muestra 2 polaridad horizontal en fase
H_Q[2]	float	Muestra 2 polaridad horizontal en cuadratura
H_I[..]	float	
H_Q[..]	float	
H_I[validSamples]	float	Muestra <i>validSamples</i> polaridad horizontal en fase
H_Q[validSamples]	float	Muestra <i>validSamples</i> polaridad horizontal en cuadratura

Tabla 1 - Estructura lógica de un pulso

A los fines de implementar el procesador Doppler se pide que:

- Obtenga el valor de cada *gate*, a partir del cálculo de la media aritmética de las n muestras que caen en el rango de resolución. Es decir, si se tienen 5000 *validSamples*, cada *gate* se obtendrá a partir de 10 muestras (0,5km * 5000 muestras / 250 km). Esto se debe realizar para el canal H y V. Como resultado, obtendremos una estructura de datos que se puede interpretar como una matriz (gate,pulso), por canal.
- Luego, se debe aplicar la formula de auto correlación en tiempo discreto, sobre todos los valores de de todos los pulsos para un determinado gate, es decir, por columna. La ecuación de autocorrelación es la siguiente:

$$\hat{R}(T_s) = \frac{1}{M} \sum_{m=1}^{M-1} V^*(m)V(m+1),$$

Donde M es la cantidad de pulsos y V es el valor absoluto de cada *gate* compleja.

El resultado de este procesamiento, se debe guardar en un archivo binario, indicando en la documentación, la estructura del mismo.

Como metodología para resolver este problema, se solicita que, primero, se realice un diseño que sea solución al problema sin explotar el paralelismo (procedural). Luego, a partir de este, realizar una nueva implementación que realice el proceso mediante el uso de la librería OpenMP, explotando el paralelismo del problema. Para ello, se requiere reconocer qué tipo de paralelismo exhibe el problema en cuestión y luego, diseñar la solución del mismo determinando cuáles son los datos/operaciones paralelizables. Se tendrá en cuenta, el nivel de paralelismo alcanzado.

Además, el informe debe contener gráficos/tablas de los datos recopilados de varias ejecuciones del programa (30, estadístico suficiente), tanto en el clúster de la facultad como localmente en una sola PC, indicando qué ganancia en *performance* existe al distribuir este proceso en comparación a la ejecución local. Comparar la ejecución procedural y de memoria compartida, explicando la diferencias observadas. También se deberá investigar acerca de qué utilidades de *profiling* gratuitas existen y que brinda cada una (un capítulo del informe), optando por una para realizar las mediciones de tiempo de ejecución de la aplicación diseñada

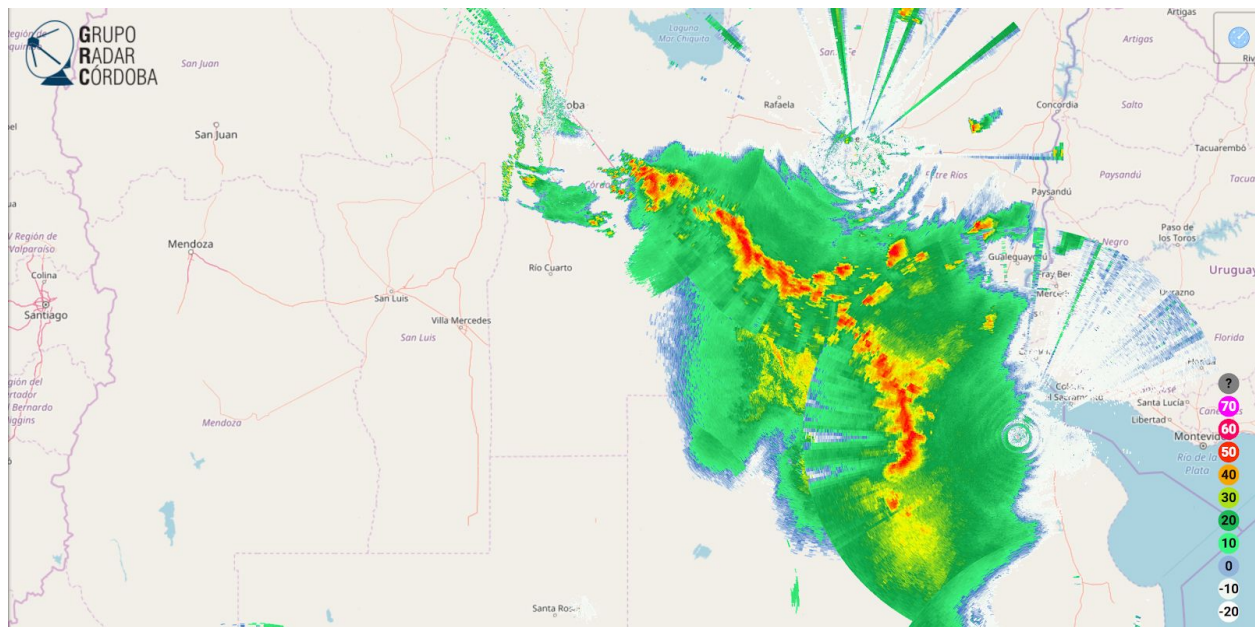


Ilustración 4: Mosaico de radares meteorológicos en Argentina

Entrega

Se deberá proveer los archivos fuente, así como cualquier otro archivo asociado a la compilación, archivos de proyecto "Makefile" y el código correctamente documentado. También se debe entregar un informe, con el formato adjunto. Gráficos y análisis de comparación entre la ejecución procedural y la distribuida. El informe además debe contener el diseño de la solución y la comparativa de profilers. Se debe asumir que las pruebas de compilación se realizarán en un equipo que cuenta con las herramientas típicas de consola para el desarrollo de programas (Ej: gcc, make), y **NO** se cuenta con herramientas "GUI" para la compilación de los mismos (Ej: eclipse).

NOTA: todavía no se encuentra habilitada la cuenta de acceso al clúster de la facultad.

Cuando esté disponible la misma, se informará por LEV los datos necesarios.

Evaluación

El presente trabajo práctico es individual deberá entregarse antes de las **23:55 (UTC -3) del día 13 de mayo de 2018** mediante el LEV. Será corregido y luego deberá coordinar una fecha para la defensa oral del mismo.

Referencias

[1] Doviak, R. J., ZRNIC, D. S., "*Doppler Radar and Weather Observations*", National Severe Storms Laboratory – NOAA, Academic Press, 1984

[2] Richardson, M. A., "*Fundamentals of Radar Signal Processing*", McGraw Hill, 2005

