

PI 121 - 08

PARALELIZACIÓN DE ALGORITMOS EN PLATAFORMAS DISTRIBUIDAS:
CASO DE ESTUDIO EN PROCESAMIENTO SÍSMICO EN LA INDUSTRIA
PETROLERA

Camilo Antonio Linares Moreno

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
Bogotá D.C.
Junio de 2014

PI 121 - 08

PARALELIZACIÓN DE ALGORITMOS EN PLATAFORMAS DISTRIBUIDAS:
CASO DE ESTUDIO EN PROCESAMIENTO SÍSMICO EN LA INDUSTRIA
PETROLERA

Camilo Antonio Linares Moreno

MEMORIA DE TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO
DE LOS REQUISITOS PARA OPTAR AL TÍTULO DE
MAGISTER EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Director

Ing. Juan Pablo Garzón Ruiz M.Sc.

Asesores

Ing. Enrique González Guerrero Ph.D.

Flor Alba Vivas Mejia D.Sc.

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Bogotá D.C.

Junio de 2014

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA DE SISTEMAS**

Rector Magnífico

P. Jorge Humberto Peláez Piedrahita S.J.

Decano Académico Facultad de Ingeniería

Ingeniero Jorge Luis Sánchez Tellez

Decano del Medio Universitario Facultad de Ingeniería

Padre Antonio José Sarmiento Leal S.J.

Director de Maestría en Ingeniería de Sistemas y Computación

Ingeniero Enrique González Guerrero Ph.D.

Director Departamento de Ingeniería de Sistemas

Ingeniero Rafael Andrés Gonzalés Rivera

Artículo 23 de la Resolución No. 1 de Junio de 1946

“ La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia ”

*A Dios, fuente de amor y
sabiduría.
A Carolina, por su comprensión
y apoyo incondicional.
A Samuel y Daniel por
motivarme con su existencia.*

Agradecimientos

A mis profesores en el programa MISYC, gracias por su motivación continua y ejemplo.

A mis orientadores Juan Pablo, Flor Alba y Enrique, por su guía, dedicación y paciencia en esta travesía.

A mi familia, gracias por su apoyo constante, comprensión y motivación.

A la Pontificia Universidad Javeriana, por darme la oportunidad de continuar aprendiendo cada día.

A ECOPETROL S.A. por su confianza y apoyo en el desarrollo de mis estudios.

Índice general

Índice de figuras	e
Índice de Tablas	h
1. Introducción	1
2. Marco conceptual	5
2.1. Oportunidades de la computación de alto desempeño en la industria de petróleo y gas	6
2.2. Marco teórico	7
2.2.1. Diseño de algoritmos paralelos	7
2.2.2. El procesamiento de datos sísmicos	11
3. Estrategias de paralelizacion	21
3.1. Estrategias de paralelización enmarcadas en PCAM	21
3.2. Características de plataformas Clúster y Grilla	26
3.3. Selección de estrategias de paralelización	27
3.4. Selección de estrategia de paralelización para caso de estudio	28
4. Caso de implementación	30
4.1. Caso de estudio	30
4.1.1. Migración sísmica preapilada en el dominio de disparo común por extrapolación del campo de onda en profundidad	31
4.1.2. Migración en dominio del disparo	33
4.1.3. Migración en dominio de ondas planas.	34
4.2. Implementación de los algoritmos de migración	35
4.2.1. Analisis PCAM del diseño de los algoritmos de migración	35
4.2.2. Implementación de algoritmo de migración por disparo	39
4.2.3. Implementación de algoritmo de migración por ondas planas	41
4.2.4. Características de los datos sísmicos	44
4.3. Experimentación	45
4.3.1. Plataforma de experimentación	45

4.3.2. Diseño del experimento	48
4.3.3. Experimentación	48
4.3.4. Resultados	49
5. Conclusiones	58
Bibliografía	60

Índice de figuras

1.1.	Plataformas líderes de alto desempeño de listado Top 500. a)Listado de las 10 plataformas líderes; nótese que el rango de desempeño va de 2.90 a 33.9 Pflops. b) Gráfica de desempeño proyectado al 2020 según análisis de Top500. [1]	1
1.2.	Representación del Método Sísmico y captura de un disparo. A la izquierda se representan los principios básicos del Método Sísmico. A la derecha, la gráfica de los datos correspondientes a un disparo en una captura sísmica. [2]	3
1.3.	Bloques de Procesamiento Sísmico: El esquema muestra en bloques las diferentes etapas del Procesamiento Sísmico. [3]	3
2.1.	Ejemplos Arquitecturas de Computadores. En cada una de las arquitecturas presentes en el cuadro se muestran ejemplos de implementaciones disponibles.	9
2.2.	Ejemplo Migración Sísmica pós-apilado [4]. a) muestra el campo antes de realizar el proceso de migración, b) muestra el campo despues del resultado de la migración	12
2.3.	Captura Datos Sísmicos [5]. Esta técnica muestra la perturbación realizada en el punto S, la cual se refleja y es capturada por los geofonos en los puntos H para muestrear el subsuelo	14
2.4.	Dominio en disparo común. Este dominio muestra una fuente que genera la perturbación y varios receptores que simultaneamente capturar la información de la señal incidente.	15
2.5.	Representacion de las Trazas de los Receptores en Disparo Común	15
2.6.	Traza en Disparo común despues de haber realizado la corrección de tiempos de propagación	16
2.7.	Dominio de Desplazamiento Común. Dominio en donde la fuente se va desplazando a una distancia constante y generando las perturbaciones en distintas posiciones	16
2.8.	Trazas de la Superficie de Estudio en Apilamiento por Offset Común	17

2.9. Imagen de dominio de punto medio común, en donde se captura la información de varios geofonos y se hace referencia como punto medio al R6[5].	18
2.10. Trazas del dominio de punto medio común[5]	18
2.11. Descripción de la obtención de los datos por el dominio de receptor común	19
2.12. Resultado de las trazas obtenidas por el geofono <i>G2L</i> de los datos en dominio de receptor común	20
3.1. Matriz de selección de estrategias de paralelización	28
4.1. Distribución grafica de la estructura del archivo de migración sísmica capturado	30
4.2. Modelo de velocidades Marmousi utilizado en la generación del dato sintético, para 240 disparos con 96 canales de captura por cada disparo [6]	31
4.3. Descripción en bloques del proceso de migración sísmica	32
4.4. Diagrama de bloques del esquema de migración sísmica por disparo.	33
4.5. Diagrama de bloques del esquema de migración sísmica por ondas planas.	34
4.6. Diagrama de la implementación de la metodología PCAM para los algoritmos de migración sísmica	36
4.7. Diagrama de Bloques Algoritmo de Migración Sísmica por Disparo	39
4.8. Diagrama de Bloques Algoritmo de Migración Sísmica por Ondas Planas	41
4.9. Campo de Velocidades correspondiente al dato sintético Marmousi [6]	48
4.10. Dato Marmoussi Migrado por Dominio del Disparo	49
4.11. Dato Marmoussi Migrado por Dominio de Ondas Planas con $pmin=700us$ y 81 Migraciones	50
4.12. Migración de 50 Shots	51
4.13. Migración de 100 Shots	51
4.14. Migración de 200 Shots	51
4.15. Migración de 5 Ondas Planas	52
4.16. Migración de 50 Ondas Planas	52
4.17. Migración de 101 Ondas Planas	53
4.18. Resultado Tiempos de Ejecución en el Cluster Zine1, en dominio del disparo y ondas planas	53
4.19. Resultado Tiempos de Ejecución en el Cluster Zine1, en dominio del disparo y ondas planas	54
4.20. Eficiencia del uso del Cluster con el Algoritmos de Migración en el Dominio del Disparo y Ondas Planas	55

4.21. Speedup del uso del Cluster con el Algoritmo de Migración por Disparo 56

4.22. Speedup del uso del Cluster con el Algoritmo de Migración por Ondas

 Planas 56

Índice de Tablas

3.1. Lista de chequeo de las consideraciones para el diseño de un algoritmo con la metodología PCAM	25
3.2. Escalas de valoración - Criterios de evaluación para alternativa a presentar	29
3.3. Evaluación de alternativas - selección de alternativa a implementar . .	29
4.2. Características Principales de los datos del Bloque Tairona	44
4.1. Características Principales del dato Marmoussi [6]	44
4.3. Características Principales del cluster Zine1 [7]	46
4.4. Parametros ejecutados para migración de Algoritmo de Ondas Planas	52

Capítulo 1

Introducción

La computación de alto desempeño ha tenido una gran evolución en las últimas décadas, llegando a niveles de desempeño del orden de los petaFlops, como se muestra en el listado TOP500 del 2013, el cual clasifica las supercomputadoras de mayor desempeño en el mundo. En efecto, la proyección de crecimiento en desempeño ha mantenido su tendencia creciente con una expectativa de alcanzar la escala de los ExaFlops en un estimado de 5 años como se ilustra en la Figura 1.1(b). [1]

#	Site	Manufacturer	Computer	Country	Cores	Peak Pfl/s	Share
1	National University of Defense Technology	NUDT	Tianhe-2 NUDT TH-HB-FEP, Xeon E5 2.7GHz, Intel Xeon Phi	China	3,120,000	33.9	17.8
2	Oak Ridge National Laboratory	Cray	Titan Cray XK7, Opteron 16C 2.2GHz, Gemini, NVIDIA K20x	USA	560,640	17.6	8.21
3	Lawrence Livermore National Laboratory	IBM	Sequoia BlueGene/Q, Power BQC 16C 1.6GHz, Custom	USA	1,572,864	17.2	7.89
4	RIKEN Advanced Institute for Computational Science	Fujitsu	K Computer SPARC64 VIIIx 2.0GHz, Tofu Interconnect	Japan	795,024	10.5	12.7
5	Argonne National Laboratory	IBM	Mira BlueGene/Q, Power BQC 16C 1.6GHz, Custom	USA	786,432	8.59	3.95
6	Swiss National Supercomputing Centre (CSCS)	Cray	Piz Daint Cray XC30, Xeon E5 2.6GHz, Aria, NVIDIA K20x	Switzerland	115,984	6.27	2.33
7	Texas Advanced Computing Center/UT	Dell	Stampede PowerEdge C8220, Xeon E5 2.7GHz, Intel Xeon Phi	USA	462,462	5.17	4.51
8	Forschungszentrum Juelich (FZJ)	IBM	JUQUEEN BlueGene/Q, Power BQC 16C 1.6GHz, Custom	Germany	458,752	5.01	2.30
9	Lawrence Livermore National Laboratory	IBM	Yucca BlueGene/Q, Power BQC 16C 1.6GHz, Custom	USA	393,216	4.29	1.97
10	Leibniz Rechenzentrum	IBM	SuperMUC iDataPlex DX360M4, Xeon E5 2.7GHz, Infiniband FDR	Germany	147,456	2.90	3.52

(a)



(b)

Figura 1.1: Plataformas líderes de alto desempeño de listado Top 500. a) Listado de las 10 plataformas líderes; nótese que el rango de desempeño va de 2.90 a 33.9 Pflps. b) Gráfica de desempeño proyectado al 2020 según análisis de Top500. [1]

Para lograr desempeños superiores a TeraFLOPS es indispensable el uso de arquitecturas y tecnologías que habilitan el procesamiento paralelo. El Top Ten del listado TOP500 incluye totalmente plataformas paralelas como clústeres, computadores vectoriales y supercomputadores. Ésta capacidad de cómputo es requerida para solucionar problemas científicos y comerciales complejos en diferentes campos. Nuevas arquitecturas que integran aceleradores basados en plataforma híbridas CPU-GPU están contribuyendo a lograr mejores niveles de desempeño, tema que presenta un campo importante de investigación [8].

En este contexto de arquitecturas paralelas emergentes y crecimiento en la capacidad de procesamiento disponible, el diseño de algoritmos paralelos continúa igualmente su evolución desde diferentes enfoques, que van desde modelos totalmente explícitos hasta aquellos en los que se abstraen por completo de la implementación del paralelismo. Sin embargo, no existe un enfoque único para que los programadores de software paralelo enfrenten el desarrollo de éste software garantizando el uso óptimo de los recursos disponibles en las plataformas de computación paralelas.

El objetivo del trabajo es el de identificar estrategias de paralelización y analizar sus características funcionales en plataformas distribuidas como los clúster o grillas, usando como caso de estudio el Procesamiento Sísmico en la industria petrolera.

Para esto se parte del análisis de una de las metodologías de desarrollo paralelo conocida como PCAM (Particionamiento, Comunicación, Aglomeración y Mapeo) [9]; mediante la identificación de puntos de decisión fundamentales de la metodología se van a plantear cuatro estrategias de paralelización que puedan implementarse en plataformas clúster y grilla. Una de esas estrategias será seleccionada con el fin de aplicarla en el caso de estudio en Procesamiento Sísmico.

El caso de estudio está enmarcado en la industria de Petróleo y Gas, específicamente en el campo de la Exploración Sísmica. La investigación en Exploración Sísmica es de interés en la actualidad del país por ser una actividad fuente de sostenibilidad económica y de desarrollo social para Colombia. Igualmente es un campo con posibilidades de financiación en investigaciones posteriores por parte de organizaciones académicas e industriales del sector privado y estatal soportados en el Plan de Desarrollo 2010 - 2014 [10].

Específicamente, el caso de estudio se implementa en el área de Procesamiento de Datos Sísmicos en dos dimensiones (2D); este proceso se fundamenta en el método sísmico de exploración, en el cual, a partir de la captura de las reflexiones de ondas sísmicas se obtiene una imagen del subsuelo que permite la identificación de las estructuras presentes en él. La imagen del subsuelo es interpretada con el apoyo de otras disciplinas como la geología con el fin de determinar la ubicación de posibles fuentes de hidrocarburos.

En la Figura 1.2 se ilustra la aplicación del método sísmico en la exploración en 2 dimensiones (izquierda) y un ejemplo del dato adquirido para un disparo (derecha).

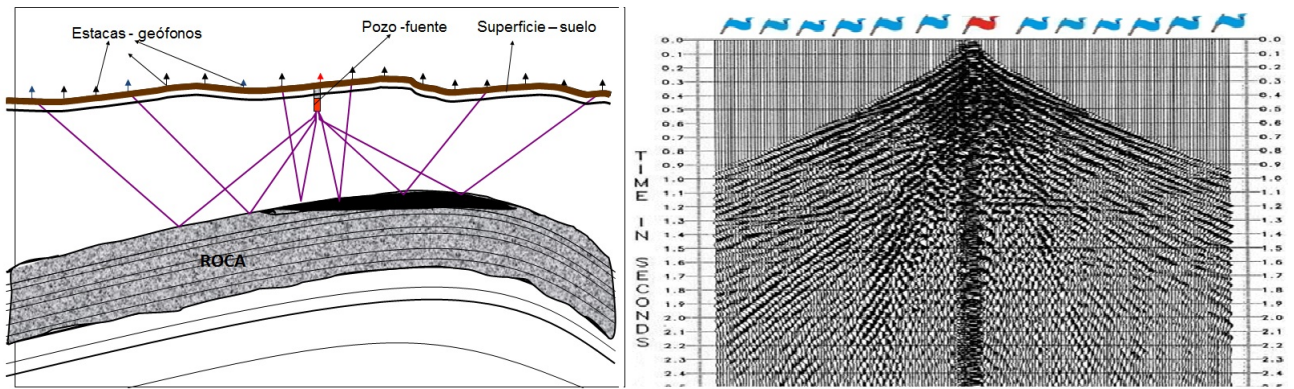


Figura 1.2: Representación del Método Sísmico y captura de un disparo. A la izquierda se representan los principios básicos del Método Sísmico. A la derecha, la gráfica de los datos correspondientes a un disparo en una captura sísmica. [2]

El Procesamiento de Datos Sísmicos es una disciplina que debido al tamaño de los datos obtenidos en la etapa de Adquisición y la complejidad de las operaciones numéricas que deben realizarse para su procesamiento, requiere de altas capacidades de cómputo para procesar la información en el menor tiempo posible, teniendo en cuenta la competencia en el sector y la constante necesidad de hallazgos de nuevas fuentes de hidrocarburos. En la Figura 1.3 se ilustran las etapas del procesamiento sísmico.

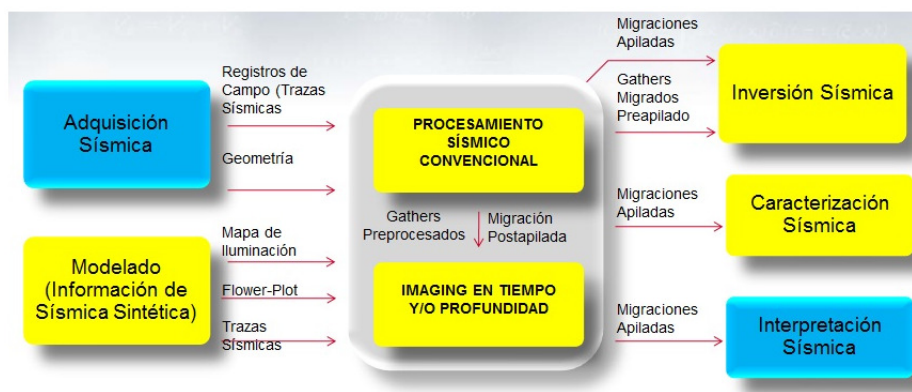


Figura 1.3: Bloques de Procesamiento Sísmico: El esquema muestra en bloques las diferentes etapas del Procesamiento Sísmico. [3]

La presentación del trabajo inicia con una revisión conceptual de los principios básicos de computación paralela, destacando los diferentes enfoques para el diseño de algoritmos paralelos, al igual que conceptos básicos de la aplicación de computación de alto desempeño en el área de la Exploración Sísmica en la industria de petróleo y gas.

En el capítulo 3 se analiza la metodología PCAM y se establece la propuesta de las estrategias de paralelización, soportada en la identificación de los puntos de decisión durante las etapas de análisis y las características fundamentales del diseño de software paralelo. Se realiza la selección de una estrategia de paralelización identificando la aplicabilidad al caso de estudio y la implementación que se describe en el capítulo 4 soportado en la plataforma de computación de alto desempeño del Centro de Alto Rendimiento Computacional ZINE- de la Pontificia Universidad Javeriana.

Capítulo 2

Marco conceptual

El diseño de software paralelo es en la actualidad, una necesidad para satisfacer diferentes requerimientos tales como:

- La creciente disponibilidad de plataformas de computación paralela, una nueva visión de las problemáticas y métodos de solución en ciencias naturales y sociales.
- El cambio del paradigma de pensamiento secuencial hacia pensamiento paralelo tanto para el planteamiento de problemas como de soluciones.
- Los costos de producción de plataformas de alto desempeño.

Estos requerimientos fundamentales han sido evidenciados por Skillicorn [11], Greenlaw [12] y Lin [13]. La Pontificia Universidad Javeriana, a través del grupo de investigación en Sistemas Distribuidos y Redes de Computadores (SIDRE) y de su Centro de Alto Desempeño Computacional ZINE, ha desarrollado investigaciones en sistemas distribuidos, paralelismo y computación de alto desempeño permitiendo resolver problemas científicos que demanden altos recursos computacionales y puedan soportarse en sistemas distribuidos de alto desempeño [7], [14].

Este trabajo presenta una propuesta de estrategias de paralelización que parten de la metodología de diseño presentada por Ian Foster denominada PCAM (Particionamiento, Comunicación, Aglomeración y Mapeo)[9], la cual orienta en la práctica el diseño y desarrollo de algoritmos paralelos.

En el presente trabajo son presentadas cuatro estrategias, de las cuales se elige una para su implementación como caso de estudio, orientado al campo de procesamiento sísmico, usado en la exploración de hidrocarburos para la Industria de Petróleo y Gas, el cual es materia de interés a nivel nacional. Se estudia la problemática de la migración sísmica y se genera un aporte de paralelismo nuevo al existente desde la estrategia de paralelismo seleccionada.

2.1. Oportunidades de la computación de alto desempeño en la industria de petróleo y gas

La computación distribuida es el paradigma de computación que utiliza múltiples recursos de procesamiento o computadores conectados a la red, cada uno de los cuales tiene su propio procesador o procesadores y otros recursos. Algunos paradigmas de computación distribuida como los clúster, las grillas y la computación en la nube, los cuales serán definidos en el capítulo 3, habilitan la computación paralela la cual utiliza más de un procesador simultáneamente para la ejecución de un mismo programa.

Los clúster son una configuración de computadores, usualmente de bajo costo, interconectados a través de una red usualmente de alta velocidad, los cuales trabajan sincronizadamente como un solo recurso computacional. Este paradigma permite una solución costo-efectiva comparada con supercomputadores de características similares. Los elementos de procesamiento pueden ser PC'S, estaciones de trabajo ó sistemas multiprocesador, la interconexión a la red se realiza a través de interfaces dedicadas y requiere de un middleware que tiene la función de enmascarar la configuración para hacer ver el clúster como un solo sistema hacia el exterior [15].

Una exposición resumida de las bases de la computación en grilla y en la nube es presentada por Stanoevska y Wozniak [16]. Magoules y Wilkinson presentan detalladamente la computación en grilla, su arquitectura, técnicas, herramientas y ejemplos de aplicación [17], [18]. Una taxonomía para la computación en la nube enfocada al contexto científico es expuesto por Oliveira [19]. Se tiene referencia de investigación en geofísica utilizando estos paradigmas computacionales como una opción a evaluar [20].

Estos paradigmas de computación habilitan la computación de alto desempeño que requieren investigaciones en áreas, como la biología, la física, la aeronáutica y la predicción atmosférica entre otros campos [21]. En efecto, investigaciones muestran aplicaciones en diferentes campos de la ciencia y la industria de la computación paralela de alto desempeño: implementaciones en dinámica molecular [22], biología de tejidos, procesamiento de señales [23], planeación de tratamientos para radioterapia en cáncer [24], la industria agrícola [16], secuenciación genómica, física de altas energías y geociencias [20], problemas que incluyen el tratamiento de grandes cantidades de datos y procesamiento intensivo de los datos.

El petróleo, debido a su potencial energético, es actualmente uno de los recursos fundamentales para las sociedades del siglo XXI, siendo un imperativo actual la identificación de nuevas reservas de petróleo y gas a la par del descubrimiento de nuevas fuentes energéticas que permitan su reemplazo, al menos en forma parcial. Colombia es un país con 23 cuencas sedimentarias que la ubican con un potencial

en recursos petroleros; actualmente tiene una industria petrolera regulada por la Agencia Nacional de Hidrocarburos con participación estatal y privada y los ingresos provenientes de la extracción de petróleo y gas soportan parcialmente la economía nacional [25], [26].

Dentro de las iniciativas de investigación identificadas, es conveniente referenciar proyectos que adelanta Ecopetrol directamente y en convenio con Colciencias, como la convocatoria 531 Colciencias - Ecopetrol para "*financiar proyectos de investigación del sector de petróleo y gas, orientado a temas de exploración y producción*"; una de las líneas de investigación es el "*desarrollo de herramientas computacionales aplicadas a simular, en 2D y 3D, procesos geológicos que intervienen en la información, migración y acumulación del petróleo en cuencas sedimentarias*" [27].

2.2. Marco teórico

A continuación se presentan los conceptos generales del diseño de algoritmos paralelos y una introducción conceptual al procesamiento sísmico, caso de estudio sobre el cual se realiza la implementación en este trabajo.

2.2.1. Diseño de algoritmos paralelos

Se presenta una revisión rápida de la evolución de la disciplina del diseño de algoritmos paralelos, identificando conceptos esenciales y los diferentes enfoques desde el modelado de la máquina paralela y del paradigma de nivel de abstracción. Se analiza PCAM como una de las metodologías para desarrollo de algoritmos paralelos.

Antecedentes

Aunque la computación de alto desempeño inicia en la década de 1950 con la motivación de desarrollar cálculos científicos orientados a la seguridad nacional, como el cálculo de tablas de artillería (ENIAC), diseño de armas nucleares y/o criptografía, las primeras máquinas paralelas con múltiples procesadores aparecen hasta la década de 1970 y los primeros computadores paralelos comerciales hasta 1980 con aplicaciones comerciales como la exploración de hidrocarburos, la manufactura de automóviles y aplicaciones fundamentadas en la experimentación y simulación numérica en campos de la química, física, mecánica, cosmología, dinámica de fluidos, biología, medicina y modelado del clima [28], [9]; en la década de 1990 con la reducción de costos en la producción de PC'S y el incremento en el desempeño de las redes de comunicación se hizo viable la comercialización de clusters de PC'S a bajos costos de inversión, permitiendo que nuevos campos como el desarrollo farmacéutico,

la genética, el procesamiento de señales, modelamiento 2D y 3D y la animación por computador incentivarán el desarrollo de la computación paralela.[29].

Arquitecturas y modelamiento de computadores paralelos

En general, los computadores paralelos pueden modelarse con tres bloques principales: Procesadores, Memoria y Redes de Interconexión. Un resumen conceptual de diferentes arquitecturas de computadores paralelos son [28], [9], [30]:

- Unica instrucción, dato unico (*Single Instruction, Single Data Stream* (SISD)): Este tipo arquitectura resalta los computadores con un unico procesador en donde no es realizado ningun tipo de paralelismo. Una unidad de control basico realiza una pila de instrucciones desde la memoria, luego la unidad de control básico envia las señales necesarias para que se realice la tarea, una operación a la vez.
- Unicas instrucciones, multiples datos (*Single Instruction, Multiple Data Streams* (SIMD)): Son arquitecturas que aprovechan multiples datos en una sola instrucción, realizando operaciones que son facilmente paralelizables.
- Multiples Instrucciones, datos unicos (*Multiple Instruction, Single Data Stream* (MISD)): Es una arquitectura poco común que es generalmente utilizada para sistemas tolerantes a fallas. La arquitectura plantea que sistemas heterogeneos deben operar sobre los mismos datos y al finalizar deben concordar en el resultado.
- Multiples Instrucciones, multiples datos (*Multiple Instruction, Multiple Data streams* (MIMD)): Computadores con arquitecturas de multiprocesadores o multicomputadores hacen parte de este grupo, en donde varias CPU's pueden de forma simultanea ejecutar diferentes instrucciones y manipular multiples datos.

Existen computadores paralelos, los cuales pueden contener 100 o mas CPU's, este tipo de computadores van a requerir de algún tipo de memoria distribuida con el fin de brindar suficiente ancho de banda entre las CPU's. En un sistema de memorias distribuidas cada CPU contiene una memoria local, luego a medida que existan una mayor cantidad de memorias locales el ancho de banda va a ser mucho mayor, mejorando el desempeño de la computadora.

<p style="text-align: center;">SISD</p> <p style="text-align: center;">Computadores de Procesadores únicos</p> <p style="text-align: center;">Pcs, servidores de un único Procesador.</p>	<p style="text-align: center;">SIMD</p> <p style="text-align: center;">Arreglos de Procesadores</p> <p style="text-align: center;">IBM <u>Altivec</u>, Intel <u>MMX</u>.</p>
<p style="text-align: center;">MISD</p> <p style="text-align: center;">Lineas de procesamiento Independiente.</p> <p style="text-align: center;">Sistemas de navegación como los usados por el transbordador espacial.</p>	<p style="text-align: center;">MIMD</p> <p style="text-align: center;">Computadores de múltiples núcleos.</p> <p style="text-align: center;">Procesadores Intel <u>Larrabee</u>.</p>

Figura 2.1: Ejemplos Arquitecturas de Computadores. En cada una de las arquitecturas presentes en el cuadro se muestran ejemplos de implementaciones disponibles.

Modelos de programación paralelos

Un modelo de computación paralela es una abstracción de un computador paralelo que permite separar las consideraciones de diseño y desarrollo de software de las consideraciones propias de la implementación y la ejecución paralela. En los últimos 40 años, el diseño de algoritmos paralelos ha tenido diferentes enfoques, basados en modelos distintos de computación paralela, algunos con mayor abstracción y eficiencia que otros. Skillicorn desarrolló en 1998 una clasificación de los principales modelos de programación paralela bajo una propuesta de seis criterios que se enuncian a continuación [11]:

- Facilidad de programación.
- Metodología de desarrollo.
- Independencia de la arquitectura.
- Facilidad en su entendimiento.
- Garantías de desempeño.
- Información sobre el costo del desarrollo.

En su análisis, Skillicorn plantea 6 modelos presentados en orden decreciente de abstracción:

1. Sin notación explícita - paralelismo Implícito.

2. Paralelismo explícito, Descomposición implícita.
3. Descomposición Explícita, Mapeo implícito.
4. Mapeo Explícito, Comunicación implícita
5. Comunicación explícita, sincronización implícita
6. Todo explícito.

Por otro lado, Foster describe algunos modelos como el intercambio de mensajes (Message Passing Model), el modelo de paralelismo de datos (Data Parallelism Model) y el de Memoria Compartida (Shared Memory Model) [9]. Otros modelos los describe Chandy Mani: como el modelo UNITY, la programación funcional, la programación basada en reglas, la programación secuencial imperativa y la programación lógica [31]. El presente trabajo toma como metodología de desarrollo el propuesto por Ian Foster denominado PCAM por las iniciales de sus cuatro pasos principales [9]:

- Particionamiento
- Comunicaciones
- Aglomeración
- Mapeo.

Esta metodología de desarrollo se soporta en el modelo de programación tarea - canal, el cual se plantea para el modelo de arquitectura del multicomputador [9].

PCAM como metodolgia de diseño paralelo

Un modelo comúnmente aceptado para simplificar el diseño y análisis de algoritmos paralelos es el PRAM (Parallel Random Access Machine), modelo en el cual no se tiene en cuenta el problema de la comunicación entre procesadores y memoria ni entre diferentes conjuntos de procesadores [32], [15], [33]. Otros modelos de análisis incluyen consideraciones de la topología específica de conexión del sistema (por ejemplo: anillos, mallas, cubos, hipercubos) o la orientación de los datos, tareas, comunicación entre procesadores (maestro - esclavo) entre otros. Los modelos permiten estructurar el algoritmo paralelo seleccionando, las técnicas de diseño aplicando estrategias que permitan minimizar las interacciones entre los elementos de procesamiento.

Ian Foster desarrolló y propuso la metodología conocida como PCAM [9]. La metodología define cuatro pasos principales a saber: Particionamiento, Comunicación,

Aglomeración y Mapeo; cada uno de los pasos requiere el análisis de consideraciones especiales y puntos de decisión. Ejemplos de aplicación de la metodología se muestran en Quinn y Foster [28] [9] los cuales permiten identificar diferentes puntos de decisión al aplicar la metodología. Al aplicar esta metodología, un conjunto específico de decisiones de diseño producen una estrategia de paralelización, la cual generará una solución particular de paralelización.

2.2.2. El procesamiento de datos sísmicos

En la Industria de Petróleo y Gas, la exploración de hidrocarburos utiliza diferentes métodos geofísicos para la identificación de las zonas con potencial de recursos hidrocarburíferos en las cuales desarrollar pozos exploratorios. Entre estos métodos, el método sísmico es uno de los cuales ha tenido un mayor desarrollo y ha impactado fuertemente la industria, pues ha permitido reducir la incertidumbre exploratoria generando disminuciones de costos, riesgos y esfuerzos en la búsqueda de hidrocarburos. [5].

Este método permite la identificación de la geometría de los reflectores y la determinación de sus profundidades, lo cual se realiza utilizando las propiedades de reflexión de las ondas sísmicas, las cuales se reflejan entre los materiales que poseen propiedades elásticas diferentes.

Los datos sísmicos una vez obtenidos deben procesarse para habilitar su interpretación. Uno de los subprocesos en el procesamiento sísmico es la migración sísmica la cual colapsa la energía en el punto donde fue generada la reflexión.

Migración sísmica

La migración sísmica tiene como objetivo utilizar los campos de onda para determinar la posición de los reflectores sísmicos en el tiempo o en profundidad. Las secciones sísmicas migradas representarán los reflectores sísmicos con la inclinación y longitud verdadera, aumentando de esta forma la resolución lateral por el colapso de las difracciones [34].

Existen dos tipos de migraciones, las migraciones pre-apilado y las migraciones pos-apilado. La migración pre-apilado, realiza un ajuste de los datos sísmicos antes de su procesamiento, esta migración es generalmente utilizada cuando las capas del subsuelo tienen perfiles de velocidad complicados o cuando las estructuras son muy complejas como el caso de estructuras salinas las cuales son difíciles de estudiar con la técnica de pos-apilado. El proceso de migración pos-apilado realiza la migración sobre una sección apilada obtenida como la superposición de información sísmica que se presupone redundante. Actualmente este tipo de migración es realizada para tener una idea del modelo. En la industria, la migración actualmente utilizada es

la migración preapilado en tiempo (PSTM, de las siglas en ingles, Pre Stack Time Migration).

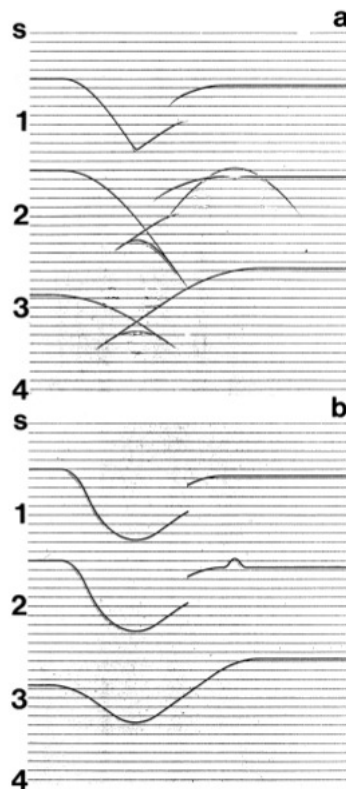


Figura 2.2: Ejemplo Migración Sísmica pós-apilado [4]. a) muestra el campo antes de realizar el proceso de migración, b) muestra el campo despues del resultado de la migración

En la Figura 2.2, se muestra un caso de migración. En la parte superior la imagen a muestra el esquema sin migrar y en la parte inferior la imagen b muestra el resultado despues de pasar por el proceso de migración. Sobre la region 2 en las imagenes se muestran dos características, en la imagen a se muestra una forma de cupula la cual despues de migrada se obtiene como resultado una curvatura sobre la superficie reflejada en la imagen b. Sobre esta misma area en la imagen a estan dos eventos cruzados, las cuales van a corresponder a una superfice concava en la imagen b. La Figura 2.2 muestra como la migración permite generar una mejor definición en la estructura describiendo mejor la superficie del campo muestreado. Para realizar este proceso de migración, existen distintos metodos los cuales se encuentran basados en la implementación de algoritmos matematicos para llegar a sus resultados.

Métodos de migración sísmica

Entre los primeros métodos de migración se utilizó el método de superposición por semicirculo previo al análisis computacional. Posteriormente se utilizó otro méto-

do llamado de sumatoria de difracción, el cual estaba basado en realizar una sumatoria de las amplitudes a lo largo de una hipérbola de difracción cuya curvatura depende de la velocidad del medio. Otro método llamado de sumatoria de Kirchhoff, utiliza también la sumatorias de difracción con la diferencia de realizar la suma en amplitud y correcciones de fase. Actualmente los métodos de migración pueden ser clasificados en tres grupos [35] [4]:

- Métodos basados en las ecuaciones de aproximación de altas frecuencias (Métodos Kirchhoff o de trazado de rayos)
- Métodos de extrapolación del campo de onda en el tiempo (utilizan ecuación de onda sin aproximaciones analíticas).
- Métodos de extrapolación del campo de onda en profundidad (utilizan las ecuaciones de onda unidireccionales las cuales incluyen operadores pseudo-diferenciales).

Los algoritmos que realizan el proceso de migración sísmica, se evalúan en términos de la variación vertical, lateral de la velocidad, apertura del operador de migración y eficiencia en la implementación [4].

Uno de los métodos utilizados es el de PSPI el cual es ideal para campos que se encuentran caracterizados por fuertes velocidades laterales, el principio de este método es introducir velocidades referencia para cada punto de la extrapolación con el fin de tomar en cuenta las variaciones de velocidades laterales, obteniendo así un campo de ondas de múltiples referencias en el campo de frecuencia-número de onda, basado en la siguiente ecuación [36].

$$P_o(x, y, z, \omega) = P_o(x, y, z, \omega) * e^{i * \frac{\omega}{v(x, y, z)}} \quad (2.1)$$

Organización de datos sísmicos - dominios

Al momento de realizar el proceso de migración, los datos deben tener una organización especial, la cual dependerá del experimento realizado o de las características que se deseen conocer del campo. Los datos sísmicos pueden organizarse teniendo en cuenta diferentes factores comunes que están reflejados en la información registrada en el archivo de captura sísmica.

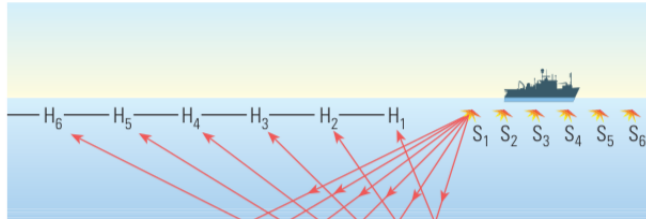


Figura 2.3: Captura Datos Sísmicos [5]. Esta técnica muestra la perturbación realizada en el punto S, la cual se refleja y es capturada por los geofonos en los puntos H para muestrear el subsuelo

Para generar un dato sísmico es necesario realizar un experimento como el que se muestra en la Figura 2.3, en este caso se tiene un barco el cual tiene varios receptores en la parte tracera los cuales van a capturar las señales reflejadas en las diferentes interfaces en el subsuelo, de la onda generada por la fuente S. En el archivo sísmico será indicado el disparo inicial y el delta de distancia entre el siguiente disparo, de esta forma para cada disparo existirá un grupo de receptores fijos que capturaran las señales, así a medida que la fuente se va desplazando sobre la superficie, se ira generando el dato sísmico correspondiente a un conjunto dado de disparos. Basandose en este experimento es posible organizar los datos sísmicos en diferentes dominios los cuales están clasificados de la siguiente forma:

- Dominio del Disparo Común (CGS: de las siglas en inglés Common Shot Gather);
- Dominio de distancia Común (COG: de las siglas en inglés para Common Offset Gather);
- Dominio del Punto Medio Común (CMP: de las siglas en inglés Common Midpoint);
- Dominio del Receptor Común.

Dominio del disparo común

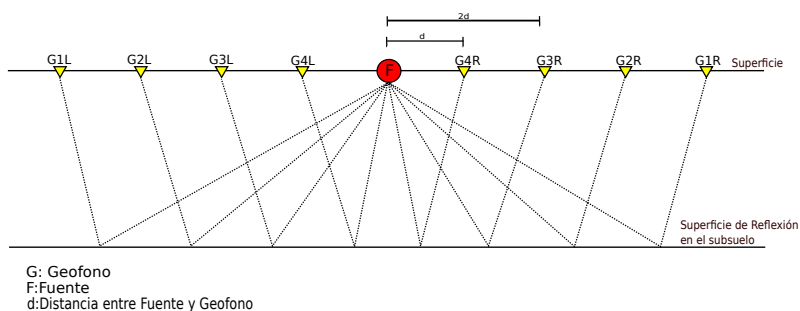


Figura 2.4: Dominio en disparo común. Este dominio muestra una fuente que genera la perturbación y varios receptores que simultáneamente capturan la información de la señal incidente.

El dominio del disparo común organiza las trazas sísmicas en familias o conjuntos de trazas correspondientes al mismo disparo como se muestra en la Figura 2.4. La fuente genera ondas que son propagadas en el subsuelo como muestran las líneas punteadas. Cada uno de los receptores va a recibir la onda generada por la fuente en tiempos diferentes debido a la distancia entre uno y otro, generando un gráfico de las trazas sísmicas como se muestran en la Figura 2.5.

Este dominio es conocido como el dominio natural de adquisición de los datos sísmicos. Actualmente este dominio es uno de los más utilizados en la migración por extrapolación de los campos de onda.

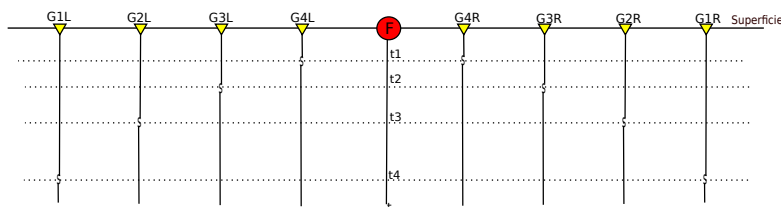


Figura 2.5: Representación de las Trazas de los Receptores en Disparo Común

Teniendo los datos de cada uno de los receptores y utilizando un modelo de velocidades el cual es obtenido según las características geofísicas del medio que se está estudiando, es posible llevar cada una de las trazas al tiempo inicial en el cual fue generado el disturbio, de tal forma que pueda tenerse una traza total de la superficie, como se muestra en la Figura 2.6.

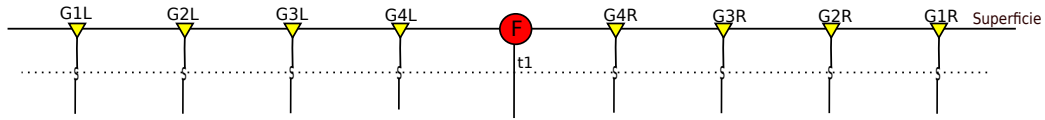


Figura 2.6: Traza en Disparo común despues de haber realizado la corrección de tiempos de propagación

Finalmente está traza describe la superficie sobre la cual la onda generada por la fuente fue reflejada y permitirá tener una imagen parcial del campo.

Domino de desplazamiento común

Otra forma de organización de los datos es el dominio de offset común, en este caso el concepto principal es realizar el modelo del campo generando movimientos en la fuente F , y los datos analizados son los de los receptores que se encuentran a una distancia constante común a todas las fuentes.

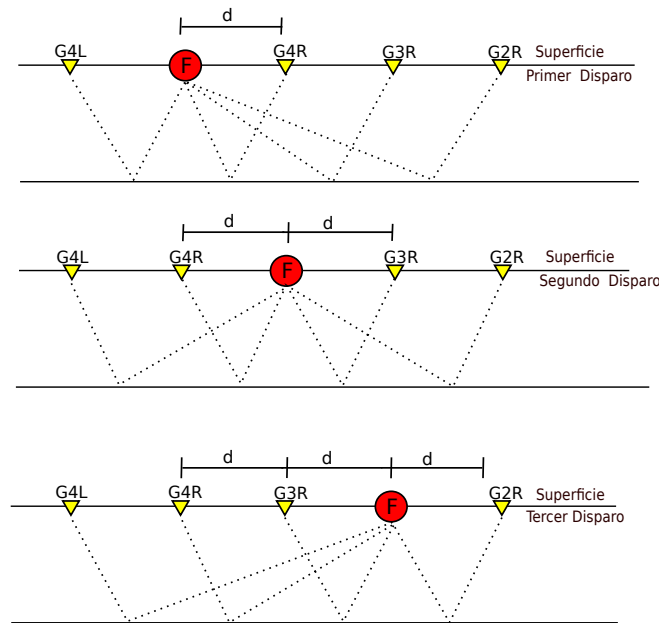


Figura 2.7: Dominio de Desplazamiento Común. Dominio en donde la fuente se va desplazando a una distancia constante y generando las perturbaciones en distintas posiciones

La Figura 2.7 ilustra el dominio de desplazamiento común. El procedimiento se va a describir para los receptores mas proximos a la fuente que genera el disturbio, este caso particular es llamado de desplazamiento minimo, pero es posible realizar esta organización con cualquier grupo de receptores siempre y cuando se respeten las condiciones que serán indicadas.

En el primer disparo, se registran los datos de los receptores $G4R$ y $G4L$, estos receptores se encuentran a una distancia d constante la cual es común con la fuente,

en el segundo paso, se cambia la posición de la fuente que genera el disturbio, en este caso se realizó un corrimiento a la derecha y son registrados los datos de los receptores $G4R$ y $G3R$, se repite el procedimiento de corrimiento a la derecha de la fuente para un tercer disparo y se capturan los datos de los receptores $G3R$ y $G2R$. Estos corrimientos generarán un grupo de receptores de offset común, que para este caso serán $G2R$, $G3R$, $G4R$, utilizando los datos de estos receptores se conforma el dominio de desplazamiento común.

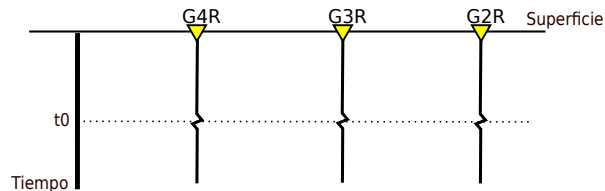


Figura 2.8: Trazas de la Superficie de Estudio en Apilamiento por Offset Común

En la Figura 2.8, se muestran las trazas resultantes de esta organización, estas trazas van a tener el mismo tiempo de propagación desde la fuente al receptor porque la distancia d , se mantiene constante. Los desplazamientos de la fuente pueden ser realizados en cualquiera de los dos lados, lo que aumentaría el conjunto de trazas en offset común. Este dominio es uno de los más utilizados, especialmente por los métodos de migración de Kirchhoff.

Dominio del punto medio común(Common Midpoint)

El dominio del Punto Medio común o CMP, tiene como objetivo agrupar las trazas resultantes del experimento con respecto a un punto de reflexión arbitrario, de esta forma se generará un grupo de receptores que van a dar como resultado una traza final resultante para ese punto de reflexión [4], esta técnica es una de las más utilizadas actualmente porque permite mejorar la calidad de la señal obtenida aumentando la relación señal a ruido.

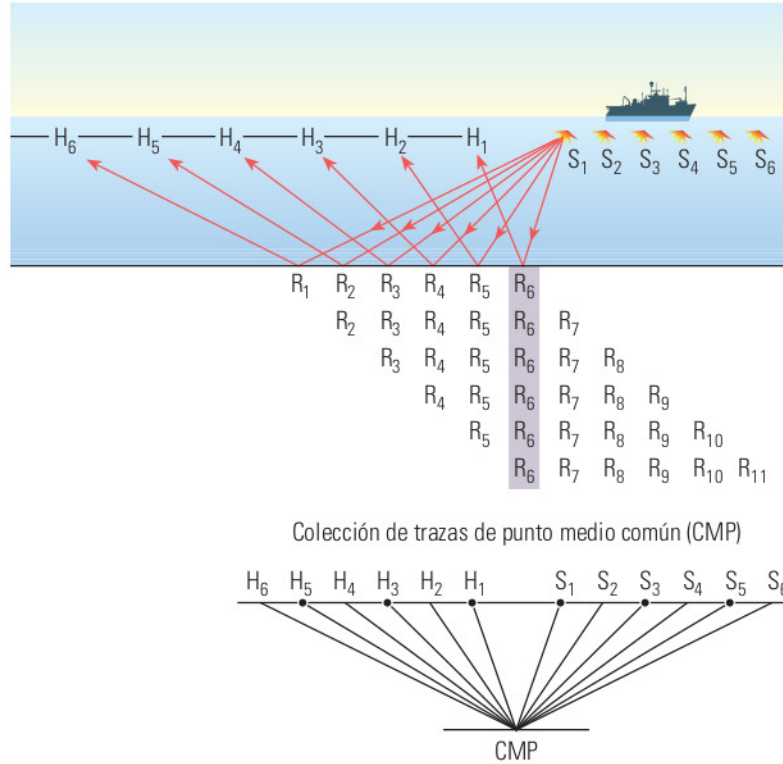


Figura 2.9: Imagen de dominio de punto medio común, en donde se captura la información de varios geofonos y se hace referencia como punto medio al R_6 [5].

En la Figura 2.9, en la parte superior se muestra como a medida que se van generando los disparos a través de las fuentes S , se van tomando los datos en los diferentes receptores. Para este caso fue escogido como punto medio común el receptor R_6 , como se muestra en la parte inferior. A medida que se muestrea el campo, varios puntos medios deben ser escogidos para obtener una traza resultante en ese punto.

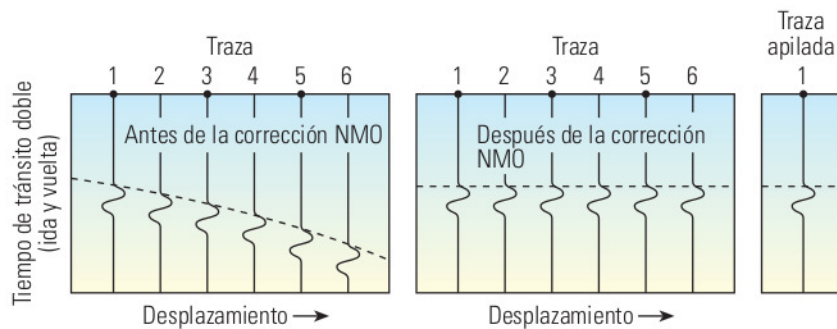


Figura 2.10: Trazas del dominio de punto medio común[5]

Una vez se tiene la información de cada uno de los sensores, las trazas resultantes de cada uno de los geofonos es graficada en el mismo plano. En la Figura 2.10 del lado izquierdo se muestran las trazas correspondientes a un CMP, las cuales describen

una forma de hipérbola; este fenómeno se da debido a que las trazas de los geofonos mas lejanos de la fuente, van a tener un retraso en tiempo debido a la velocidad de propagación de la onda en el medio. Para llevar las trazas a un mismo punto, es necesario aplicar una corrección denominada corrección de sobretiempo por distancia (*normal moveout, NMO*). El resultado una vez implementada la corrección se ve en el grafico derecho en donde todas las trazas se encuentran alineadas. Finalmente, estas trazas se deben sumar para obtener la traza equivalente para ese CMP determinado.

Este dominio es generalmente utilizado para el analisis de velocidades del campo muestreado, correcciones por NMO y stack o apilado.

Dominio de receptor común

El dominio de receptor común tiene como objetivo, escoger un receptor fijo en el cual van a ser capturadas las trazas a medida que la fuente se va desplazando. Este método es muy utilizado cuando se quiere realizar el filtrado de los datos muestreados.

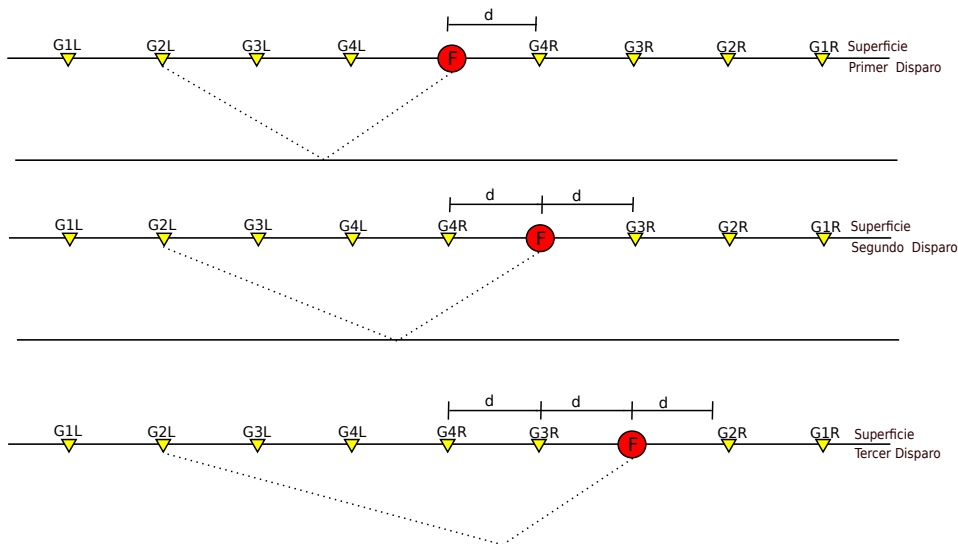


Figura 2.11: Descripción de la obtención de los datos por el dominio de receptor común

Como se ve en la Figura 2.11, la fuente F se va desplazando hacia el lado izquierdo, mientras se mantiene el receptor $G2L$ fijo en la misma posición. A medida que la fuente se mueve el punto de reflexión va cambiando y de igual forma existirá un tiempo de propagación mas demorado debido a la velocidad de propagación de la onda en el medio.

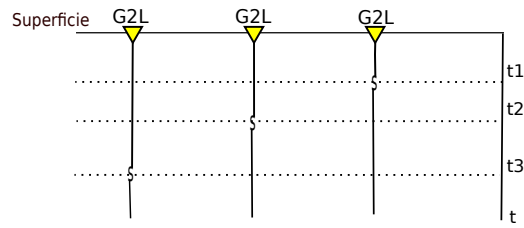


Figura 2.12: Resultado de las trazas obtenidas por el geofono $G2L$ de los datos en dominio de receptor común

En la Figura 2.12, se encuentran las trazas obtenidas por el geofono $G2L$ para cada uno de los disparos, cada una de ellas es generada en un tiempo t_i diferente, luego utilizando un modelo de velocidades del campo es posible llevar cada una de las trazas a un tiempo t_o y generar la imagen total del campo.

Capítulo 3

Estrategias de paralelización

Se desarrolla en este capítulo el análisis de la metodología de diseño PCAM con el objetivo de identificar los puntos de decisión y características de diseño fundamentales que permiten definir una estrategia de paralelización. Se seleccionan cuatro estrategias alineadas a características de las plataformas clúster y grilla y se plantea una estrategia la cual se implementa mediante el diseño de un caso de estudio de aplicación en la industria de Petróleo y Gas.

3.1. Estrategias de paralelización enmarcadas en PCAM

En el marco conceptual se identificaron diferentes enfoques desarrollados con el objetivo de diseñar algoritmos paralelos. El enfoque seleccionado para el presente trabajo es la metodología PCAM presentada por Ian Foster, el cual es un modelo adecuado para plataformas con memoria distribuida como es el caso de las plataformas Cluster y Grilla hacia las cuales se plantea esta propuesta [9], [28].

En el presente capítulo se presenta el análisis realizado a la metodología y se identifican puntos esenciales de decisión que definen una estrategia de paralelización.

Enmarcados en la metodología PCAM se propone la definición de estrategia de paralelización como el conjunto de decisiones de diseño específicas que direccionan el diseño del algoritmo paralelo. Estas estrategias dependerán de la combinación de las características fundamentales que se generan en el desarrollo de la metodología; la identificación de las características fundamentales y la caracterización de las estrategias resultantes se desarrolla a continuación.

Análisis PCAM

La metodología PCAM propone un enfoque progresivo en el diseño de algoritmos paralelos, en el cual cada etapa se enfoca en uno de los requerimientos fundamentales

identificados por Foster; a continuación se definen estos requisitos fundamentales en el contexto de PCAM [9]:

- **Concurrencia:** Es la capacidad de un algoritmo de realizar sus operaciones en forma simultánea en diferentes procesadores. Esta definición de concurrencia es diferente a la usada en el contexto de los sistemas operativos y bases de datos en el cual la concurrencia se usa para describir ejecuciones lógicamente simultáneas más que físicamente simultáneas. Esta diferencia contextual, sin embargo, debido a que enfrentan los mismos retos y utilizan soluciones similares permiten utilizar el mismo concepto de ejecución simultánea sea esta física o lógica, [13].
- **Escalabilidad:** Capacidad de un algoritmo de adaptarse a incrementos en el número de procesadores en forma eficiente
- **Localidad:** Capacidad de un algoritmo de mantener una alta proporción de accesos a memoria local respecto a los accesos a memoria remota
- **Modularidad:** La capacidad de descomponer entidades complejas en componentes más simples, es un concepto fundamental para reducción de costos en el desarrollo de software y su mantenimiento.

A continuación se resumen las etapas propuestas por la metodología, sus puntos fundamentales y

1. **Particionamiento:** El objetivo fundamental de esta etapa es la identificación de oportunidades de ejecución paralela. Esto se hace identificando tareas de procesamiento y datos relacionados a las tareas identificadas, partiendo inicialmente de la partición más pequeña posible, definiendo una característica que denomina la granularidad. En esta etapa se busca la granularidad más "fina", es decir, la partición de tareas más pequeña que permita aprovechar el mayor nivel de paralelismo posible para la especificación del problema. Se identifican dos técnicas de descomposición denominadas *descomposición por dominio* y *descomposición funcional*. La descomposición por dominio identifica inicialmente los datos asociados al problema, luego identifica un particionamiento de la computación sobre los datos y por último identifica cómo se asocia ese particionamiento sobre los datos. En el enfoque de "descomposición funcional" se identifican inicialmente las tareas computacionales a desarrollar y luego identifica los datos relacionados a las tareas identificadas.

El foco, desde el punto de vista de los requerimientos es la identificación de oportunidades de concurrencia, lo cual se realiza con la búsqueda de la granularidad más fina.

2. **Comunicación:**Esta etapa se enfoca en identificar cuales comunicaciones son requeridas para coordinar la ejecución del algoritmo, evaluar qué tipo de estructuras de comunicación son más apropiadas y qué algoritmos utilizar para la comunicación.

En esta etapa se identifican los patrones de comunicación que pueden ser locales o globales, estructurados o no estructurados, estática o dinámica y síncrona o asíncrona. [[28], p-67] - [[9], p-33].

La comunicación local es el patrón en el cual una tarea específica requiere datos de un pequeño número de otras tareas para lo cual se crean canales para generar el flujo de comunicación. El otro patrón es la comunicación global, en el cual una tarea específica necesita datos de un número significativo de otras de las tareas identificadas.

Un patrón estructurado es el que utiliza una estructura regular de comunicación entre las tareas, un patrón no estructurado utiliza estructuras no regulares, por ejemplo una estructura de grafos arbitrarios.

En el patrón estático, las estructuras no cambian en la ejecución, en el dinámico las estructuras pueden ajustarse y ser determinadas por condiciones de ejecución.

El patrón de comunicación síncrona coordina la comunicación entre productores y consumidores, en contraste, el patrón asíncrono no requiere esta coordinación. .

El foco en esta etapa es identificar a través de que patrones las particiones propuestas en la etapa anterior pueden comunicarse en forma más eficiente y refinar con éste análisis el diseño.

3. **Aglomeración:**En esta etapa el foco es evaluar el diseño obtenido en las etapas previas y evaluar sus características de desempeño y costo de implementación para poder ejecutarlo en una plataforma de computación paralela específico.

Esta evaluación puede generar la recombinación de tareas, el ajuste de los patrones de comunicación identificados o consideraciones de replicación de datos y/o el cómputo. La recombinación de tareas genera mayor granularidad y una compensación con el número de comunicaciones requeridas.

La redundancia de estructuras de comunicación, datos o computación se evalúan también en esta etapa con relación a los requerimientos de desempeño y costos, en ocasiones para obtener menores tiempos de ejecución, se pueden implementar estructuras redundantes o consolidar tareas computacionales repetitivas.

Esta etapa se enfoca en aumentar la localidad del algoritmo con el fin de cumplir con los requerimientos de desempeño y costo requeridos por el problema.

4. **Mapeo:** En esta etapa se realizan las definiciones acerca de la forma en que las tareas son asignadas a los procesadores que van a ejecutarlas. Se tienen en cuenta tareas que pueden ejecutarse concurrentemente para ubicarlas en diferentes procesadores y aprovechar los recursos paralelos; igualmente, se identifican las tareas que se comunican frecuentemente para ubicarlas en el mismo procesador incrementando la localidad.

Esta etapa incluye el análisis de creación estática o dinámica de tareas, los patrones de comunicación, la interacción de comunicaciones entre tareas y la intensidad de las operaciones de cómputo de las diferentes tareas. Quinn [28] plantea un árbol de decisión para tomar decisiones sobre la estrategia de mapeo con base en los factores enunciados.

Al final de cada etapa Foster propone la evaluación del diseño o diseños alternativos resultantes contra una "lista de chequeo" de cada etapa que se presenta en el Cuadro 3.1. Esta lista de chequeo genera preguntas para validar el diseño y advertencias relacionadas con las implicaciones de la decisión de diseño tomada para el análisis del diseñador.

Particionamiento	<p>1 ¿Su partición esta definida con un orden de magnitud mayor a la cantidad de procesadores disponibles en el equipo de procesamiento?. En caso contrario, se tiene una pequeña flexibilidad en los estados de diseño siguientes.</p> <p>2 ¿ Su partición evita computación redundante y requerimientos de almacenamiento?. En caso contrario, el algoritmo resultante no va a ser escalable para problemas de gran escala.</p> <p>3 ¿Hay tareas de tamaño comparable?, si no, será complicado asignar a cada procesador la misma cantidad de trabajo.</p> <p>4 ¿El número de tareas se escalan con el problema?, idealmente el aumento en el tamaño del problema generará un aumento en las tareas. Si este no es el caso, el algoritmo en paralelo, no tendra la capacidad de resolver problemas mayores cuando más procesadores esten disponibles.</p> <p>5 ¿Ha identificado varias estrategias de particionamiento?, Pueden maximizar la flexibilidad en estados siguientes de diseño, considerando alternativas ahora. Se debe considerar la descomposición por dominios y descomposición funcional.</p>
Comunicación	<p>1¿Todas las tareas se desempeñan sobre la mismas operaciones de comunicación? requerimientos de comunicaciones desbalanceadas sugiere un escaleamiento no constructivo, es necesario re plantear el diseño para distribuir las comunicaciones de forma mas equilibrada.</p> <p>2 ¿Cada tarea se comunica con un pequeño número de vecinos? Si cada, tarea debe comunicarse con varias tareas, es necesario evaluar la posibilidad de evaluar una comunicación local en vez de una global.</p> <p>3 ¿Las operaciones de comunicación estan habilitadas para trabajar de forma concurrente?. En caso contrario, el algoritmo posiblemente será ineficiente, y no escalable, es bueno evaluar técnicas de división para mejorar la concurrencia.</p> <p>4 ¿Esta la computación asociada a diferentes tareas habilitadas a funcionar concurrentemente?. En caso contrario, el algoritmo tiene el riesgo de ser ineficiente y no escalable. Es necesario considerar una reorganización de las comunicaciones y las operaciones computacionales.</p>
Aglomeración	<p>1 ¿La aglomeración redujo los costos de comunicación incrementando las comunicaciones locales?.</p> <p>2 ¿Si la aglomeración presenta computaciones repetidas, se verifica si los beneficios de esta replicación compensan los costos, para un determinado tamaño de problema y procesadores?</p> <p>3 Si la aglomeración replica los datos, ¿Verifique que no comprometa la escalabilidad de su algoritmo, restringiendo el tamaño del problema o conteos del procesador que pueda soportar?</p> <p>4 ¿La aglomeración maneja tareas con costos similares de computación y comunicación?, Si la tarea creada por la aglomeración, es mas importante que tengan costos similares.</p> <p>5 ¿El número de tareas se escalan con el problema?, En caso contrario, el algoritmo no tiene la capacidad de resolver problemas en grandes computadores paralelos.</p> <p>6 ¿Pueden el número de tareas reducirse aun mas, sin introducir desbalanceamiento, incrementando costos de ingeniería o reduciendo escalabilidad?</p>
Mapeo	<p>1 ¿Ha considerado un algoritmo basados en creaciones de tareas dinamicas? esto puede llevar a en algoritmos mas simples</p> <p>2 ¿Ha verificado que el administrador no se convierta en un cuello de botella?. Es posible reducir comunicaciones pasando apuntadores a las tareas.</p> <p>3 Utilizando un balanceamiento dinamico, ¿Ha evaluado el costo relativo de diferentes estrategias?</p>

Cuadro 3.1: Lista de chequeo de las consideraciones para el diseño de un algoritmo con la metodología PCAM

Si bien, la metodología fundamenta el diseño en los análisis y definiciones realizados en la etapa anterior y se sugiere evitar reprocesos, se aclara que las evaluaciones

parciales o completas de un diseño específico en alguna etapa pueden generar cambios en decisiones de etapas anteriores.

3.2. Características de plataformas Clúster y Grilla

En la actualidad, entre las plataformas paralelas más accesibles para implementación de software paralelo, es necesario considerar las plataformas clúster y grilla. A continuación se presentan sus características fundamentales:

- **Cluster:** Puede estar compuesto por uno o varios computadores independientes conectados a través de una red dedicada de bajo costo, baja latencia y gran ancho de banda. Son implementados a bajo costo, cada nodo mantiene sus características principales como CPU, capacidad de almacenamiento y memoria propia. Posee administrador de recursos que permite distribuir y programar cada uno de los trabajos. Los clusters pueden ser clasificados dependiendo de las necesidades del problema en tres grupos, [37]
 - **Requerimientos de uso:** Son utilizados para problemas de alto desempeño y alto rendimiento, en donde aplicaciones necesitan de un alto procesamiento computacional. Clusters de alta disponibilidad, en donde se requiere que el cluster este disponible el mayor tiempo posible, siendo tolerante a fallas ya sean de software o hardware.
 - **Tipos de Nodos:** Se encuentran clasificados en dos tipos, los nodos homogéneos, en donde todos los nodos presentan especificaciones de memoria e interconexiones similares. Los nodos heterogéneos, en contraste, presentan especificaciones diferentes en sus recursos, con énfasis en la memoria y el desempeño de conexiones.
 - **Jerarquía:** Son clasificados también en dos tipos, clusters de primer nivel, en donde no hay una jerarquía definida entre los nodos del cluster y cada nodo puede ser utilizado para cualquier propósito. Multi-nivel, en donde si existe una entre los nodos, cada nivel tiene una función especializada.
- **Grillas:** Presentan una amplia escalabilidad, teniendo la capacidad de trabajar desde unas cuantas estaciones hasta millones. Distribución geográfica, en donde las estaciones de la grilla pueden estar posicionadas en lugares distantes. Heterogeneidad, en donde los recursos de software y hardware pueden variar desde datos, archivos, componentes de software o programas, sensores o equipos científicos. Intercambio de recursos: Los recursos en una grilla pueden pertenecer a varios usuarios, esto permite que los recursos puedan ser

usados por aplicaciones, promoviendo eficiencia y bajo costo. Múltiples administraciones: Cada organización que hace parte de la grilla puede establecer diferentes políticas administrativas y de seguridad. Coordinación de recursos: Los recursos deben ser coordinados para proveer capacidades computacionales compartidas. Acceso transparente: La grilla debe ser vista como un único computador virtual [38].

La definición de una plataforma específica para la implementación depende de los recursos disponibles en cada caso específico de desarrollo; esta definición genera decisiones importantes respecto a las herramientas utilizadas para el desarrollo. La portabilidad de los desarrollos es una característica que debe evaluarse teniendo en cuenta la plataforma objetivo en la cual se plantea la ejecución del algoritmo.

3.3. Selección de estrategias de paralelización

Del análisis de la metodología PCAM ya expuesta, se identifica que el cumplimiento de los requisitos fundamentales de concurrencia, escalabilidad, localidad y modularidad pueden utilizarse directamente como criterios de selección de las estrategias de paralelización.

Del mismo modo se identifica que las decisiones fundamentales implican compensaciones entre tres características fundamentales del diseño:

- La granularidad: El número de tareas que puedan habilitarse concurrentemente es uno de las características fundamentales de diseño. Una granularidad fina habilita muchas tareas con impacto probable en incremento de comunicaciones entre las mismas. Una granularidad gruesa permitirá aumentar la localidad en compensación a perder concurrencia.
- Las comunicaciones: Los patrones de comunicación se identifican como la segunda característica fundamental de diseño. Las decisiones que se tomen en estos patrones tienen efectos directos en el tiempo de coordinación y ejecución de los algoritmos. Esta etapa exige una profunda comprensión de los impactos en escalabilidad que generan las decisiones del tipo de patrones que se van a implementar.
- La replicación: Las decisiones de replicación son alternativas enfocadas a obtener mayor es tiempos de ejecución en compensación a mayores costos en procesamiento y/o comunicaciones.

Una estrategia de diseño, debe contemplar un balance adecuado para el problema específico entre estas tres características. La metodología, en efecto, enfoca el análisis

del diseño en cada etapa a una de estas características teniendo claro que siempre existe una compensación entre las tres.

El análisis de las estrategias bajo los criterios de habilitación de la concurrencia, escalabilidad y localidad cruzado con las características funcionales de las plataformas clúster y grilla, genera la selección de las siguientes estrategias alineadas a la plataforma de alto desempeño en la cual se plantea la implementación del algoritmo.

Característica	Granularidad		Comunicación entre procesos		Redundancia tareas /datos		Plataforma más adecuada	Análisis
	Gruesa	Fina	Global	Local	Alto	Bajo		
Estrategia 1	X			X	X		Grilla	Se tienen muchos cores disponibles lo que permite un procesamiento paralelo de muchas tareas. Debido a la dispersión geográfica se deben minimizar las comunicaciones entre procesos. Debido a la dispersión geográfica se debe garantizar una disponibilidad alta que permita evitar reprocesos por caídas de nodos en la ejecución.
Estrategia 2	X			X		X	Grilla	La redundancia puede sacrificarse dependiendo de las condiciones de disponibilidad de la grilla específica
Estrategia 3	X		X			X	Cluster	Los cluster pueden contar con número limitado de core. Una ventaja de la plataforma cluster es su posibilidad de altas velocidades de acceso a datos y entre procesadores del mismo cluster lo que habilita altos requerimientos de comunicación entre procesos.
Estrategia 4		X	X		X		Cluster	Una opción de bajo costo tipo cluster puede tener restricciones en comunicaciones de alta velocidad - debe considerarse en el diseño.

Figura 3.1: Matriz de selección de estrategias de paralelización

La Figura 3.1 ilustra la definición de cuatro estrategias resultantes de la combinación de características específicas de granularidad, comunicación y redundancia, y su relación con características de las plataformas clúster y grilla respectivamente.

3.4. Selección de estrategia de paralelización para caso de estudio

Partiendo de las estrategias identificadas en el numeral anterior (Figura 3.1), se selecciona en esta etapa una estrategia para desarrollar la metodología PCAM en el área de interés del procesamiento sísmico en el área de Petróleo y Gas.

Los criterios de selección para seleccionar una de las estrategias identificadas se relacionan a continuación:

- 1. Plataforma de implementación: Al realizar la revisión de problemas computacionales en el área de procesamiento sísmico se identifica que es un problema que implica grandes conjuntos de datos y altos requerimientos de procesamiento. Las plataformas que se utilizan en la industria normalmente son cluster de computadores con múltiples procesadores comerciales estándar y/o clusters mixtos con procesadores y GPU'S. Una plataforma cluster está disponible en la Universidad en el Centro de Procesamiento de alto desempeño ZINE.

- 2. Tiempo de Implementación: Se dispone de un tiempo de 3 meses para la implementación y pruebas con un solo programador para lo cual se debe partir de una solución algorítmica probada para identificar sus posibilidades de paralelización con la estrategia seleccionada.
- 3. Innovación: El grado de innovación posible al realizar un diseño de un algoritmo secuencial estable y probado.

Definidos los criterios anteriores, y con una asignación homogénea para el peso cuantitativo se definen escalas para evaluación de cada alternativa así:

Escalas de Valoración - Selección de Alternativa a Implementar					
Plataforma		Tiempo de Implementación		Grado de Innovación	
Puntaje	Tipo	Puntaje	Tipo	Puntaje	Tipo
1	Grilla	1	Mayor a 4 meses	1	no identificado
5	Cluster	2	Mayor a 3 meses	5	identificado
		3	Menor a 3 meses		

Cuadro 3.2: Escalas de valoración - Criterios de evaluación para alternativa a presentar

Evaluación de Alternativas - Selección Alternativa a Presentar				
	Plataforma	Tiempo Implementación	Grado de Innovación	Puntaje Total
Estrategia 1: GG-CL-RA	1	1	5	7
Estrategia 2: GG-CL-RB	1	1	5	7
Estrategia 3: GG-CG-RB	5	3	5	13
Estrategia 4: GF-CG-RA	5	3	1	9

Cuadro 3.3: Evaluación de alternativas - selección de alternativa a implementar

Con base en la valoración de los criterios establecidos bajo la escala del cuadro 3.4, la estrategia seleccionada para implementación del caso de estudio es la que presenta granularidad gruesa, comunicación global, estática y baja replicación.

Se aclara que para efectos de análisis, en el caso de estudio además de la implementación de la estrategia seleccionada se va a analizar un algoritmo existente e identificar la estrategia tomada para su desarrollo.

Capítulo 4

Caso de implementación

Como caso de implementación se analiza la problemática de la migración sísmica expuesta en el marco teórico y su solución mediante el paralelismo de algoritmos de migración sísmica.

Se analizarán dos estrategias diferentes de paralelización siguiendo la metodología PCAM, las cuales luego serán implementadas y se comparará el resultado de cada una de las dos verificando los tiempos de ejecución de los algoritmos y los resultados obtenidos por cada uno.

4.1. Caso de estudio

El dato sísmico, comprende todas las trazas correspondientes a los geófonos (canales de grabación) y a cada uno de los disparos realizados en la captura de la información sísmica. Una traza es el registro de la amplitud de la reflexión de la onda sísmica en el punto específico relativo a un geófono para el tiempo definido de registro de captura sísmica.

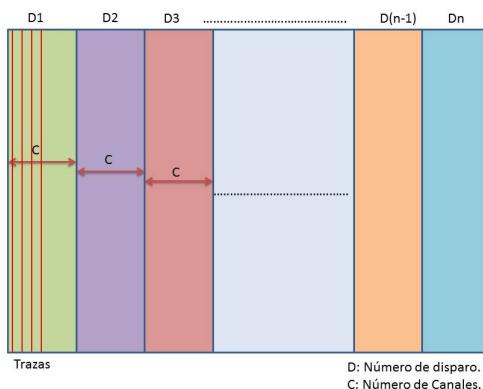


Figura 4.1: Distribución grafica de la estructura del archivo de migración sísmica capturado

La Figura 4.1, muestra graficamente como se encuentra organizado el archivo que contiene las trazas. En este caso D , indica el número de disparos realizados para el experimento, por otro lado, para cada uno de los disparos existen una cantidad de canales C , los cuales van a capturar las señales de cada disparo.

Para el caso se estudio se utilizará el campo sintético marmoussi, este campo tiene un campo de velocidades como el que se muestra en la Figura 4.2, este campo es fundamental en el proceso de migración sísmica porque indica la velocidad del medio en cada punto del plano, teniendo el espacio vs profundidad del área en la cual se desarrolló el experimento sísmico.

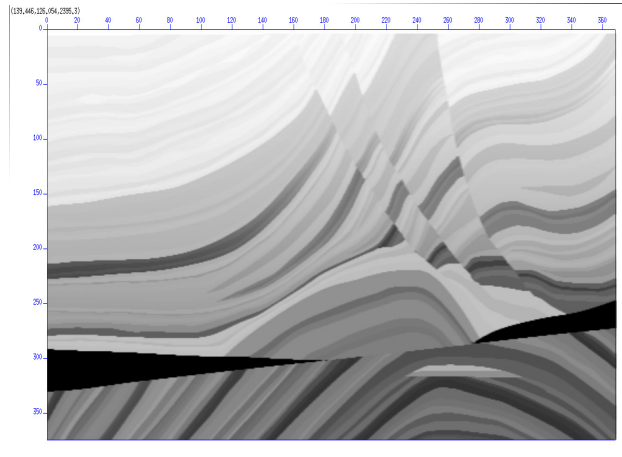


Figura 4.2: Modelo de velocidades Marmoussi utilizado en la generación del dato sintético, para 240 disparos con 96 canales de captura por cada disparo [6]

4.1.1. Migración sísmica preapilada en el dominio de disparo común por extrapolación del campo de onda en profundidad

Dado un campo de ondas incidentes S y un campo de ondas reflejadas R , la migración debe realizar la propagación de estas ondas en un medio que tiene las velocidades indicadas por el campo de velocidades V . La propagación se realiza en profundidad calculando la extrapolación de los dos campos en cada punto, tomando como base las soluciones a las ecuaciones numéricas definidas en el método de migración.

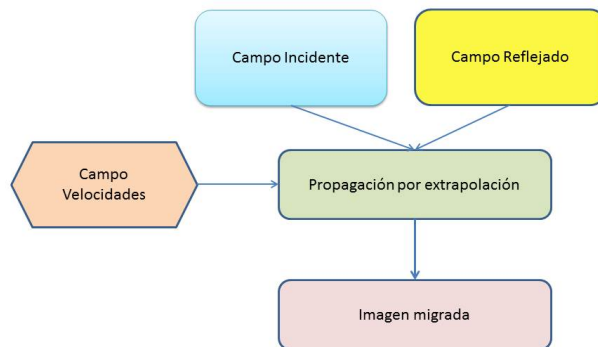


Figura 4.3: Descripción en bloques del proceso de migración sísmica

Existen diferentes métodos de migración sísmica los cuales dependen del tipo de ecuaciones numéricas que utilizan para la propagación de los campos y los dominios de datos sobre los cuales trabajan. Evaluando las posibilidades de migración se identificó uno que realiza la transformación de los datos al dominio de la frecuencia para realizar la extrapolación de los campos incidente y reflejado en profundidad, el cual se denomina migración por corrimiento de fase más extrapolación PSPI por sus siglas en inglés (Phase Shift Plus Interpolation).

Al analizar el problema de paralelismo en la migración PSPI, se identifica una estrategia de granularidad fina, que consiste en dividir la computación por cada uno de los disparos del experimento sísmico, trabajando en el dominio del disparo común.

Por otro lado, con el fin de evaluar una estrategia de granularidad gruesa se realizó una búsqueda conceptual de opciones alternativas de paralelismo y se desarrolló el planteamiento de la migración paralela de ondas planas que representen una síntesis de los disparos del experimento organizados en un dominio de receptor común, esto es, organizando los datos en el dominio de ondas planas. Este concepto será explicado posteriormente para dar más claridad a esta estrategia.

A continuación se desarrolla el análisis de las dos estrategias:

4.1.2. Migración en dominio del disparo

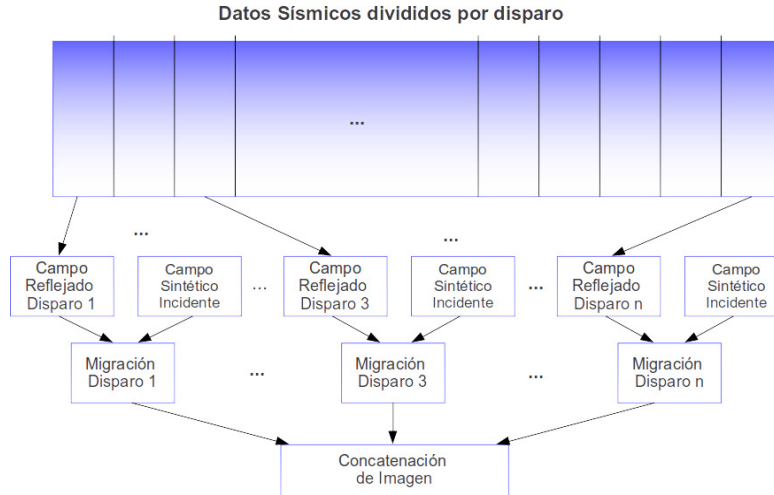


Figura 4.4: Diagrama de bloques del esquema de migración sísmica por disparo.

La Figura 4.4 muestra la estrategia de migración por disparo la cual realiza una descomposición del dato sísmico, seleccionando el grupo de trazas correspondientes a un disparo específico.

$$S(\omega, X, z) = L^* [S(\omega, X, z - \Delta z)] \quad (4.1)$$

$$R(\omega, X, z) = L [R(\omega, X, z - \Delta z)] \quad (4.2)$$

$$I(X, z) = \sum_{\omega} S^*(\omega, X, z) R(\omega, X, z) \quad (4.3)$$

Las ecuaciones 4.1, 4.2 corresponden al comportamiento de las señales de la fuente y los receptores respectivamente, y la ecuación 4.3 corresponde a la condición de imagen.

Una vez se ha seleccionado la información correspondiente al grupo de receptores dependiendo de ese disparo, se realiza la propagación de los campos incidente y reflejado utilizando la ecuación 2.1 definidas por el método PSPI y se genera la imagen resultante basado en la condición de imagen determinada como lo muestra la ecuación 4.3.

Este proceso puede ser realizado en paralelo para cada uno de los disparos que se encuentran en el campo, teniendo en cuenta que no hay concurrencia en los datos de cada disparo pues cada uno tiene sus propias trazas registradas para un disparo específico identificado.

Finalmente para obtener la imagen total, se superponen todas las trazas resul-

tantes de la migración de cada uno de los disparos procesados para obtener la imagen final.

4.1.3. Migración en dominio de ondas planas.

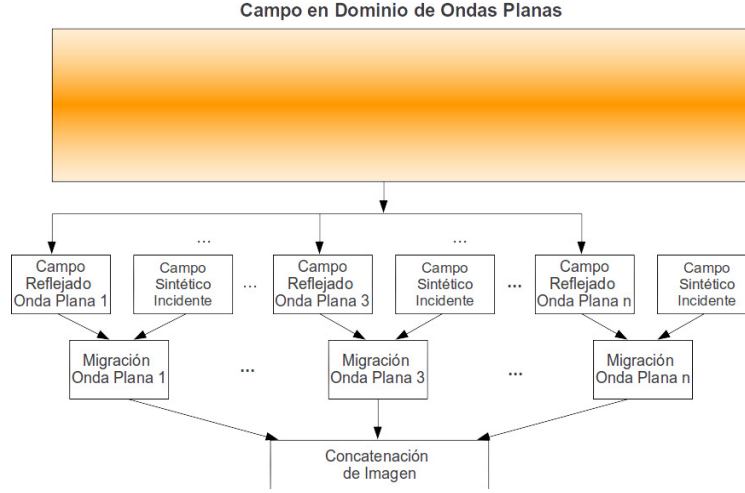


Figura 4.5: Diagrama de bloques del esquema de migración sísmica por ondas planas.

Esta estrategia se fundamenta en la teoría de ondas planas, a través de la cual una onda plana puede sintetizar todas las trazas de un dato sísmico. El proceso de ondas planas inicia llevando el campo distribuido en el dominio del disparo común a un nuevo dominio llamado τ -p, en donde su representación va a estar dado por dos parámetros τ que indica una medida temporal y p indica el parámetro de rayo, el cual estará dado por la velocidad del medio y el ángulo de incidencia. Esta nueva representación permite sintetizar toda la distribución del campo a un solo campo completo para ser procesado.

$$S_s(x, \omega) = \sum_n a_n(\omega) S_n(x, \omega) \quad (4.4)$$

$$R_s(x, \omega) = \sum_n a_n R_n(x, \omega) \quad (4.5)$$

$$I_n(x) = \sum_{k=1}^{N_\omega} W(\omega_k) S_n^*(x, \omega_k) R_n(x, \omega_k) \quad (4.6)$$

Las ecuaciones 4.4, 4.5, corresponden a las señales de las fuentes y los receptores sintetizados respectivamente. La ecuación 4.6 corresponde a la condición de imagen necesaria para generar la imagen sísmica final.

Una de las propiedades de las ondas planas es que pueden propagarse mediante un algoritmo de migración y posteriormente regresar la información al dominio del

espacio permitiendo obtener una imagen equivalente a una migración en dominio del disparo común. [39] [21]

El enfoque utilizado para paralelismo, es realizar la transformación de los datos a ser migrados, los cuales pueden ser migrados de forma simultanea debido a la independencia que genera el considerar diferentes parametros de rayo para cada problema. Este enfoque permite tener una granularidad gruesa requiriendo de menos migraciones de ondas planas para obtener una imagen equivalente comparado al número de disparos en donde en comparación con el esquema anterior es necesario realizar una migración por cada uno de los disparos.

Al final se realiza la sumatoria de la imagen de cada uno de las ondas planas migradas para proceder con la generación de la imagen final. En la Figura 4.5, se describe a través de un diagrama de bloques este proceso, utilizando este diagrama es posible observar como en este caso se analiza el campo completo a diferencia del metodo anterior en el cual el campo es fraccionado dependiendo de la cantidad de disparos.

4.2. Implementación de los algoritmos de migración

Para el presente caso de estudio fueron implementados dos algoritmos de migración. El primero realiza el analisis de los datos sísmicos en el dominio del disparo común, realizando la migración disparo a diparo como será explicado en detalle posteriormente. El segundo algoritmo es basados en la teoría de ondas planas, este algoritmo tiene como principio analizar los datos sísmicos como si estos conformaran un frente de onda continua con una determina orientación y sentido.

A pesar de que los dos algoritmos realizan el analisis sísmico en dos dominios diferentes, los dos utilizan la teoria de desplazamiento de fase mas interpolación (*PSPI*) para realizar la migración de la imagen.

4.2.1. Analisis PCAM del diseño de los algoritmos de migración

El problema del diseño de los algoritmos fue abordado de dos formas. El algoritmo por disparo común era un algoritmo ya existente, luego fue realizado un analisis detallado para abordarlo desde la metodologia PCAM, por otro lado se analizo el problema desde la perspectiva de ondas planas en donde si fue necesario el diseño e implementación completa de este algoritmo siguiendo estrictamente los lineamientos de la metodologia PCAM.

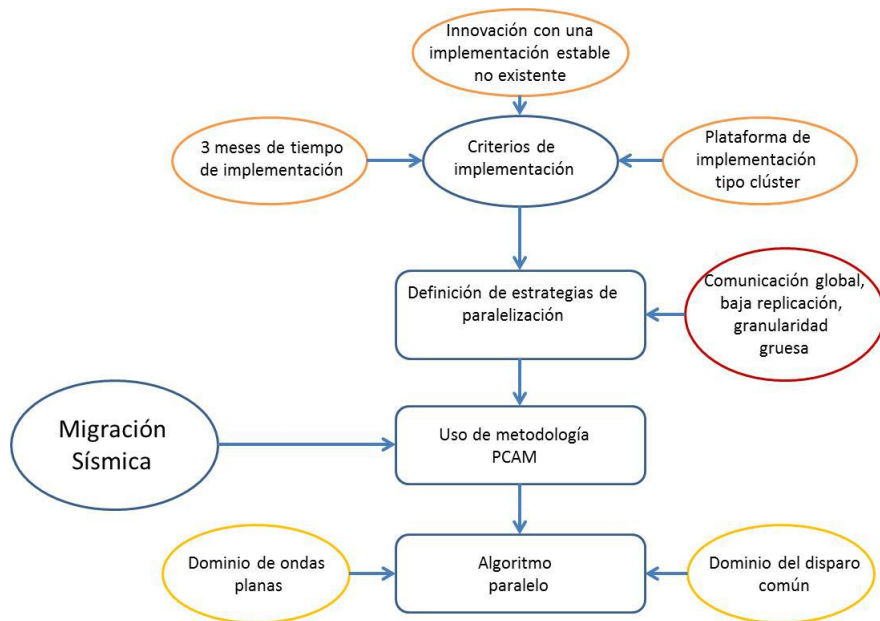


Figura 4.6: Diagrama de la implementación de la metodología PCAM para los algoritmos de migración sísmica

En la Figura 4.2.1 se ilustra la aplicación de la metodología PCAM tomando las decisiones de diseño establecidas en la estrategia seleccionada.

Algoritmo en el dominio del disparo

El análisis de los datos sísmicos utilizando el dominio de disparo común, presenta características importantes en el agrupamiento de las trazas sísmicas contenidas en el problema. En el archivo de entrada del problema esta contenida la información sobre la cantidad de receptores para cada uno de los disparos, así como el tiempo de muestreo y características específicas del muestreo del campo. Teniendo claras estas características es posible abordar la teoría de PCAM.

Como primer paso es necesario identificar la estrategia de *partición* que fue utilizada para el problema. Como ya fue dicho los datos de entrada se encuentran divididos por disparos, cada disparo contiene una cantidad de trazas las cuales estarán dadas al inicio del problema, para cada uno de estos disparos es posible realizar un proceso de migración independiente, lo cual permitiría particionar el problema en donde cada núcleo del cluster tiene la capacidad de procesar los datos de cada uno de los disparos, dándole una característica de granularidad fina al problema. Es posible resolver el problema de esta forma debido a que cada disparo equivale a un experimento diferente, lo cual generará independencia entre cada uno de los procesos.

El paso siguiente será la *comunicación*. Al conocer las características del experimento como la cantidad de disparos y canales en los receptores, es posible de-

terminar la cantidad de datos con los cuales se va a trabajar y evaluar que patrones de comunicación utilizar. Debido al tamaño del problema se observó un patrón de comunicación global, estructurada, estática y sincrónica controlada por las funciones específicas del modelo de comunicaciones de mensajes. Al mantener una comunicación estructurada es posible diseñar un esquema de árbol en donde la información de cada disparo va pasando de nodo a nodo para su procesamiento.

Como tercer paso se define la **aglomeración**. Para el problema de migración sísmica se hace un énfasis importante en la replicación de la información. Para realizar la migración es necesario conocer el modelo de velocidades del campo, de tal forma que sea posible conocer el comportamiento de la propagación de la onda en el experimento. La información sobre las velocidades será necesaria para cada uno de los disparos, luego debe estar disponible en cada uno de los núcleos que estén procesando la información, luego para este problema se manejó una replicación intensa.

Finalmente se analiza el **mapeo** del problema. En este caso serán definidas las tareas a llevar a cabo por cada uno de los núcleos. En la distribución principal se observó un esquema en donde se tiene un nodo maestro y varios nodos esclavos. El nodo maestro tendrá a su cargo las tareas de asignación y clasificación de la información en cada uno de los núcleos, respetando el esquema de árbol estático observado en la fase de particionamiento. Por otro lado los nodos esclavos tendrán a su cargo las tareas de extrapolación y migración de los datos de cada disparo. Bajo este esquema de mapeo no se tiene comunicación entre los núcleos esclavos para este caso, evitando la dependencia de información entre cada uno de los procesos.

Algoritmo en el dominio de las ondas planas

Se realiza ahora un enfoque diferente al problema de migración abordando el problema bajo el dominio de ondas planas. Utilizando este dominio se busca reducir el número de migraciones en comparación con un algoritmo como el de disparo, en consecuencia se tendrá una mayor influencia de artefactos o ruidos en la imagen los cuales podrían llegar a afectar el resultado [40]. Para este caso, fueron seguidos los lineamientos de la metodología PCAM para el diseño del algoritmo.

El estudio por ondas planas inicia introduciendo los datos al dominio de receptor común para luego transformarlos al dominio de ondas planas. Como primer paso se realiza el análisis de **particionamiento** reorganizando los datos, formando grupos de ondas planas los cuales dependerán de los parámetros de rayo definidos para el problema. Este particionamiento por ondas planas plantea un esquema de granularidad gruesa debido a que se realiza un procesamiento de todo el campo por cada onda plana. Al momento de determinar el particionamiento del problema fueron evaluados dos esquemas además de la división por ondas planas. Ya que el campo es

estudiado en profundidad la primera propuesta tenia por objetivo analizar el campo por fracciones de profundidad en donde cada uno de los procesadores podria estar encargado de una fracción del campo, a pesar de las ventajas de este particionamiento se considero que no era posible realizarlo debido a que el campo es analizado en el dominio de la frecuencia, luego es necesario el uso del campo completo por cada proceso. Debido al problema anterior, se aborda el problema desde el dominio de la frecuencia. Teniendo los datos clasificados en el dominio de ondas planas se busca que el particionamiento este enfocado a las componentes de frecuencia presentes en el campo, así cada uno de los procesos estará encargado de una componente en frecuencia para generar un resultado, este abordaje llevo a ver que en el momento en que se este realizando la conformación de la imagen el resultado de cada uno de los procesos va a ser dependiente del resultado de los otros procesos, debido a que para conocer el comportamiento de la onda sobre el campo es necesario tener todas las componentes en frecuencia.

En la fase de **comunicaciones** fueron evaluados varios esquemas de comunicación entre los cuales estan arboles binarios y estructuras en mariposa, pero fueron descartados debido al tamaño de los datos a ser procesados, luego fue escogido un esquema de comunicación global, utilizando una distribución estructurada, estatica y sincrona la cual es controlada por las funciones del modelo de comunicaciones por mensaje. A diferencia del caso de disparo, al depender de la cantidad de parametros de rayo, las comunicaciones entre los nodos puede que sean mas intensas o por el contrario que no exista una gran cantidad de comunicaciones lo cual puede llevar a un mejor desempeño temporal del algoritmo.

El estudio de **aglomeración**, se mantiene con la replicación de la información. Debido a que las características del modelo de velocidades es fundamental para la migración, es necesario que cada uno de los nucleos que estarán encargados del procesamiento de la información mantengan una copia de este archivo ya que será utilizado para cada fase de la interpolación.

La fase de **mapeo** se diseñará basado en dos enfoques. Dentro del problema de ondas planas existen dos fases, la primera busca la organización de los datos sísmicos en diferentes grupos, por otro lado la segunda fase busca realizar la migración del campo. Basandose en estas dos fases las tareas serán distribuidas, dependiendo del proceso que se realice. El nodo maestro estará encargado de dos tareas, la primera es la clasificación de los datos en el dominio de ondas planas y la segunda en la distribución de la información a cada uno de los nodos. Los nodos esclavos tendran como tareas principales realizar la extrapolación del campo en la profundidad y el procesamiento del algoritmo de migración sísmica que generará la imagen final del campo.

4.2.2. Implementación de algoritmo de migración por disparo

La implementación del algoritmo de migración utilizando la teoría de disparo común, fue basado en la organización de los datos obtenidos en el momento de realizar el muestreo. Para los archivos de prueba utilizados como es el caso del Marmoussi, se obtienen la cantidad de disparos y los canales activos correspondientes a ese disparo. Para este caso particular el archivo contiene 240 disparos y 96 canales por disparo, lo que da un total de 23040 trazas.

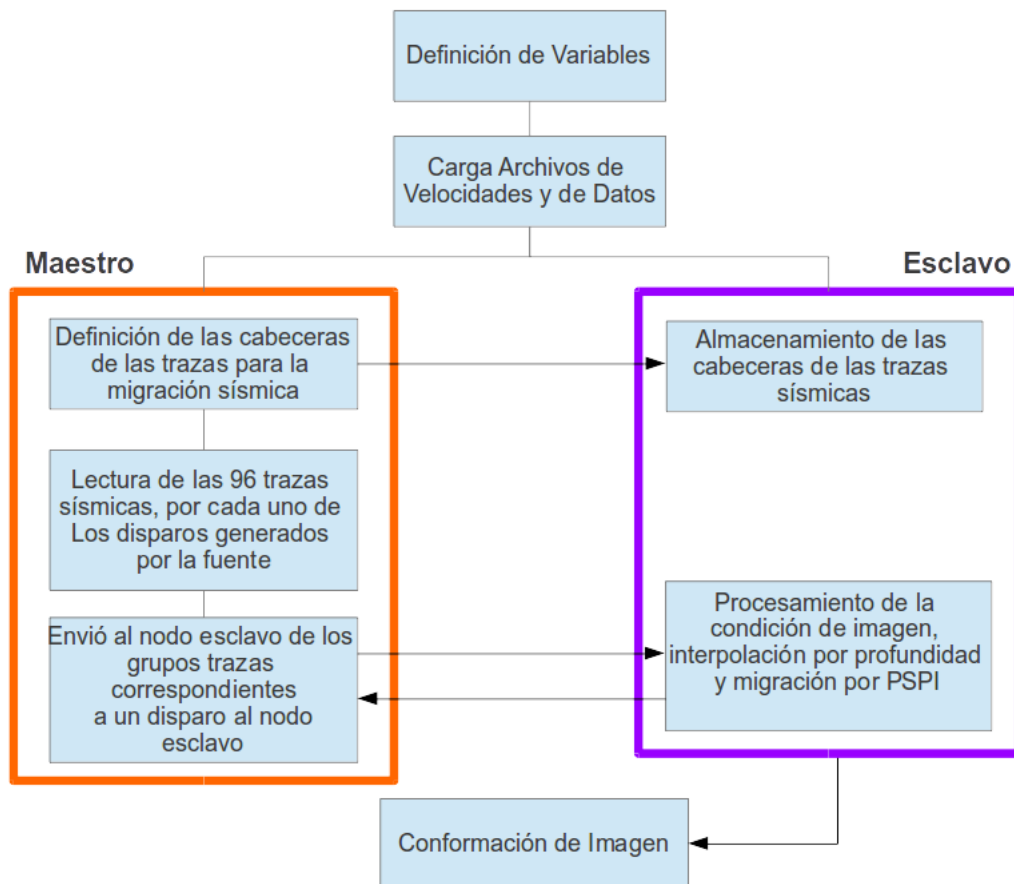


Figura 4.7: Diagrama de Bloques Algoritmo de Migración Sísmica por Disparo

La Figura 4.7 muestra la organización del algoritmo. El algoritmo está dividido en dos partes, por un lado se encuentran las tareas realizadas por el nodo maestro y de forma paralela el procesamiento realizado por cada uno de los nodos esclavos. Las tareas de cada uno de los nodos se desarrollaron de la siguiente forma:

■ Nodo Maestro

Una vez las variables han sido definidas y los archivos de datos de las trazas sísmicas como el modelo de velocidades se encuentran almacenados en va-

riables del algoritmo se inicia el procesamiento de los datos por parte del nodo maestro. Primero se almacenan las cabeceras de las trazas las cuales daran la información sobre el tamaño del campo, cantidad de canales, tiempo de muestreo, muestreo espacial y cantidad de receptores.

Teniendo la información de las cabeceras, se inicia el proceso de captura de las trazas sísmicas correspondientes a cada uno de los disparos. Cada uno de estos disparos fue registrado por un número determinado de receptores en un rango espacial en el campo definido en la cabecera, estos datos de los receptores son agrupados y enviados al nodo que se encuentre disponible para la migración de las trazas. Cada nodo entonces estará ocupado con el procesamiento de un disparo, de tal forma que se va migrando el campo disparo a disparo.

Una vez todos los disparos han sido leídos y los datos han sido enviados a cada uno de los nodos, se espera a que estos terminen de generar la imagen migrada para realizar la conformación final de la imagen. Esta conformación es realizada obteniendo la información de cada uno de los nodos y realizando una sumatoria entre cada uno de los resultados de los nodos, de tal forma que se genere la imagen final de todo el campo, al mismo tiempo se realiza el proceso de cierre de todas las comunicaciones entre los nodos para no dejar procesos abiertos que puedan generar lentitud en el algoritmo.

■ **Nodo Esclavo**

Las tareas del nodo esclavo estan divididas en dos partes. La primera parte captura la información correspondiente a las cabeceras de las trazas, de tal forma que pueda conocer las características principales del campo analizado.

En la segunda parte, el nodo esclavo recibe la información enviada por parte del nodo maestro sobre las trazas correspondientes al disparo que se esta analizando, estas trazas son procesadas en la función de migración, la cual fue denominada para este caso como "*migrate_shot*". En esta función se realiza la extrapolación por profundidad, haciendo variaciones por deltas de z en profundidad y se va realizando el proceso de migración utilizando el algoritmo de *PSPI*, el cual llevará la posición de los reflectores a su posición real dentro del campo.

Una vez terminado el proceso de migración, se genera una imagen final correspondiente a ese disparo, la cual será enviado de regreso al nodo maestro para que este realice la concatenación de cada una de las imagenes y asi se logre obtener una imagen de todo el campo.

4.2.3. Implementación de algoritmo de migración por ondas planas

La implementación del algoritmo de migración sísmica por ondas planas fue realizado en una nueva organización de los datos analizados, transformando desde el dominio temporal hacia el dominio de la frecuencia y asumiendo que cada uno de las señales fuentes se comportaban como un frente de onda que se desplazaba sobre la superficie. Toda la programación del algoritmo fue desarrollada utilizando lenguaje C y compiladores MPICH.

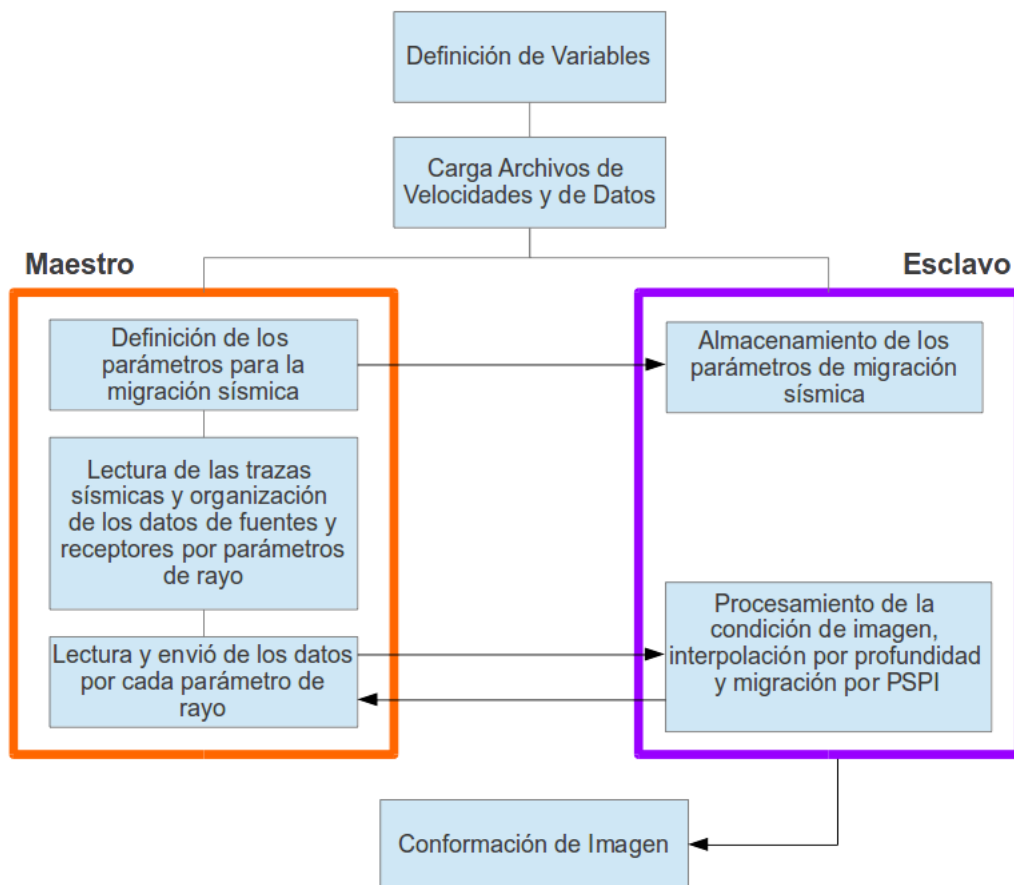


Figura 4.8: Diagrama de Bloques Algoritmo de Migración Sísmica por Ondas Planas

En la Figura 4.8, se muestra el proceso realizado en la implementación. Primero se realiza la definición y carga de archivos en dos pasos:

- Primero se realiza una definición global de todas las variables que van a ser utilizadas dentro del algoritmo de migración.
- Se cargan de los archivos de velocidades y los datos sísmicos capturados para poder ser almacenados en una variable.

Una vez definidas las variables se inicia el proceso de paralelización, utilizando las funciones del modelo de comunicaciones por mensajes, se inicia el proceso con `MPI_Initialize` y se determinan la cantidad de nodos disponibles para el procesamiento de los datos y se inicia el procesamiento de los datos, siguiendo el siguiente procedimiento:

- **Nodo Maestro**

Primero el nodo maestro realizará una lectura del archivo de velocidades previamente cargado y procesará la información de las velocidades maximas y minimas en el archivo, de igual forma enviará la información sobre las características del campo como lo son, distancia entre los receptores, tiempo de muestreo, número de canales de procesamiento, cantidad de trazas leidas a cada uno de los nodos disponibles para el procesamiento de los datos.

Una vez cargados los parámetros, se inicia el proceso de lectura de las trazas. Las trazas son procesadas de dos formas, primero se genera una arreglo en donde estarán las trazas generadas por la fuente, las cuales se asumen de forma ideal y son generadas utilizando la función Ricker, por otro lado las trazas de los receptores son las leidas en el archivo de entrada, estas trazas primero son filtradas utilizando una ventana de Hamming, la cual va a suavizar la respuesta y mejorar el procesamiento de la información. Durante el proceso de clasificación, tanto las trazas de los receptores como las trazas de las fuentes son transformadas al dominio de la frecuencia para su procesamiento. Como resultado final se obtiene dos arreglos de tres dimensiones los cuales dependeran del parámetro de rayo, la frecuencia de trabajo y los desplazamientos espaciales en x .

Teniendo los datos organizados, se incia el proceso de comunicación con cada uno de los nodos. Primero se guardan en una variable los datos correspondientes a el parámetro de rayo que se va a analizar, debido a que la información se encuentra en el dominio de los complejos, es necesario clasificar los datos de forma independiente en su parte real y parte imaginaria debido a que el modelo de comunicaciones por mensajes no soporta variables complejas para comunicación entre los nodos utilizando lenguaje C. Se verifica entonces cual de los nodos se encuentra disponible y se envian los datos del parametro de rayo a ese nodo para su procesamiento. Los procesos de comunicación son controlados directamente por las funciones `MPI_Send` y `MPI_Recv` las cuales se encargan del empaquetamiento y verificación de la comunicación entre los nodos.

Una vez el procesamiento ha terminado en cada uno de los nodos esclavos, se realiza la captura de la información de cada una de las imagenes para poder

realizar la conformación de onda. Utilizando la función `MPI_Recv` se realiza un barrido en cada uno de los nodos y se va realizando la conformación de imagen con la migración realizadá por cada uno de ellos, de igual forma se cancelan las comunicaciones establecidas entre los nodos de tal forma que no quede ningun proceso en espera que pueda impedir generar el resultado final de la migración.

■ **Nodos Esclavos**

Cada uno de los nodos esclavos van a cumplir las funciones de procesamiento e interpolación dentro del algoritmo. Una vez el nodo maestro envia la información de los parámetros de migración estos son capturados por cada uno de los nodos y almacenados en las variables definidas.

Al momento en que inicia el procesamiento de la información de las trazas, cada uno de los nodos esclavos capturará la información proveniente por el nodo maestro, la cual corresponderá a un determinado parámetro de rayo. Esta información tendra que ser nuevamente organizada en una variable del tipo complejo, para luego iniciar el proceso de interpolación en profundidad utilizando un ciclo. El ciclo de interpolación esta dividido en dos partes, primero se realiza un condicionamiento de la imagen en donde es realizada la conformación de la imagen para cada uno de los deltas de profundidad, luego es realizado el proceso de migración utilizando el algoritmo de PSPI, el cual utilizando un corrimiento de fase va a generar que la posición de los receptores se ubique posición real como fue explicado en capitulos anteriores. Para cada paso dentro de la función de **PSPI** esta va a realizar una actualización de las trazas del campo de tal forma que se vaya generando el resultado de la imagen real y poder computarla en la conformación de la imagen.

Una vez el proceso de migración ha terminado, el nodo queda disponible para enviar la información resultado al nodo maestro, con el fin de que este pueda generar la superposición de cada uno de los resultados obtenidos por los otros nodos, y asi tener como resultado la imagen total.

Durante el diseño de este algoritmo fueron tomadas en cuenta las características de repititividad de la información con el fin de evitar el uso excesivo de recursos de memoria en cada uno de los nodos. Esta característica es posible verla en la fase en que el nodo maestro realiza todo el procesamiento de los parámetros que van a ser migrados y luego los envia a cada uno de los nodos, mejorando asi el desempeño del algoritmo.

Disparos	3437
Receptores por Disparo	24
Offset	200m
Espacio entre Receptores	25m
Espacio entre Disparos	25m
Tiempo de grabación	7s
Tiempo de muestreo	2ms

Cuadro 4.2: Características Principales de los datos del Bloque Tairona

4.2.4. Características de los datos sísmicos

Para el análisis de los algoritmos implementados fueron realizadas pruebas con datos sintéticos, estos datos son generados computacionalmente con características especiales del campo de tal forma que los algoritmos puedan ser ajustados para responder como se desea. Los datos del campo Marmoussi, presenta las características que se muestran en el cuadro 4.1

Disparos	240
Receptores por Disparo	96
Offset	200m
Espacio entre Receptores	25m
Espacio entre Disparos	25m
Posición del Primer Disparo	3000m
Tiempo de grabación	2.9s
Tiempo de muestreo	4ms

Cuadro 4.1: Características Principales del dato Marmoussi [6]

Con estos datos es posible calcular el tamaño del archivo que va a ser analizado de la siguiente forma:

$$\frac{2,9 \text{ s}}{0,004 \text{ s}} = 725 \text{ muestras} \quad (4.7)$$

$$725 \text{ muestras} * 4 \text{ Bytes} = 2900 \text{ Bytes} \quad (4.8)$$

$$2900 \text{ Bytes} * 23040 \text{ trazas} = 66816000 \text{ Bytes} \approx 67 \text{ MBytes} \quad (4.9)$$

De estas ecuaciones se obtiene que el peso del dato es de 67MB, este resultado puede ser comparado con los datos de otro campo como es el campo del bloque Tairona, el cual presenta las características presentes en el cuadro 4.2.

Calculando el tamaño del dato final se obtiene lo siguiente:

$$\frac{7 \text{ s}}{0,002 \text{ s}} = 3500 \text{ muestras} \quad (4.10)$$

$$3500 \text{ muestras} * 4 \text{ Bytes} = 14000 \text{ Bytes} \quad (4.11)$$

$$14000 \text{ Bytes} * 82488 \text{ trazas} = 1154832000 \text{ Bytes} \approx 1,2 \text{ GBytes} \quad (4.12)$$

Teniendo un peso total de 1.2GB, el cual es varias veces mayor que el dato sintético, lo cual generará mayor tiempo de procesamiento para los algoritmos analizados. Estas características fueron consideradas en el diseño planteado con la teoría PCAM, ya que debido al tamaño del dato fases como la de aglomeración en donde se considera la repetitividad del dato van a tener una gran relevancia, toda vez que el realizar comunicaciones con datos de gran tamaño va a requerir más tiempo y por lo tanto generará lentitud en el procesamiento.

4.3. Experimentación

El experimento se diseña con el objetivo de verificar las diferentes respuestas de las dos estrategias de paralelización y el cumplimiento del objetivo de generar la imagen migrada del dato sísmico.

A continuación se describe la plataforma definida para la implementación del caso de estudio y los resultados obtenidos.

4.3.1. Plataforma de experimentación

La implementación del caso de estudio se utiliza soportado en el software Seismic Unix (*SU*), el cual es una plataforma de software libre que permite la experimentación y desarrollo de algoritmos de procesamiento sísmico y/o de tratamiento de señales en general; fue desarrollada y es soportada por Colorado School of Mines.

Características de la plataforma de experimentación

Para las pruebas experimentales la plataforma SU fue probada en dos equipos, primero en equipos PC utilizando sistema operativo Linux Ubuntu para la fase de desarrollo y verificación de los algoritmos implementados. Segundo, para la fase de pruebas fue utilizado el cluster Zine1 de la Pontificia Universidad Javeriana, que presenta las siguientes características

Modelo	HP ProLiant SL390s G7
Procesadores	Intel(R) Xeon(R) CPU L5640 @ 2.27GHz
Número de núcleos	384
Tamaño en Cache	12288kB

Cuadro 4.3: Características Principales del cluster Zine1 [7]

Para implementar el modelo de comunicaciones por mensajes se seleccionó MPI (Message Passing Interface) como herramienta de paralelismo, debido a que es un protocolo de comunicaciones utilizado para programar computadores paralelos, con soporte en comunicaciones punto a punto o comunicaciones colectivas. El MPI utilizado fue basado en el desarrollo MPICH y ejecutado bajo la plataforma Linux.

Características de los datos sísmicos

Los datos sísmicos utilizados para la experimentación corresponden con los datos sintéticos del modelo Marmoussi los cuales son datos patrón utilizados en la disciplina para verificación de procesamiento sísmico. Estos datos generados por el Instituto Frances de Petroleo (IFP), son datos sintéticos en 2-D, donde su geometría fue basada en los perfiles desde el Norte de Quenguela hasta cuenca de Cuanza. La geometría y el modelo de velocidades fue creado con la finalidad de generar datos sísmicos complejos que requirieran de técnicas de procesamiento avanzadas para obtener una imagen correcta de la tierra. [41] [6].

En la industria de procesamiento sísmico los datos adquiridos se transfieren en un formato estándar llamados SEG-Y. El formato SEG-Y esta compuesto por tres partes, la primera parte esta compuesta por 3200 bytes EBCDIC cabecera de tarjeta de imágenes, el cual contiene 40 tarjetas (40 líneas de texto con 80 caracteres por línea), la segunda parte es una cabecera binaria de 400 bytes, con información sobre el contenido de la cinta o archivo sísmico que ese esta leyendo y la tercera parte consiste en los trazos sísmicos adquiridos [42].

En el momento de realizar la lectura del archivo es importante tener en cuenta sobre el orden de los bytes leídos, los cuales están divididos en dos formatos "little-endian." "big-endian". Los archivos del tipo "little-endian." archivos de "low-byte", corresponden a sistemas que su arquitectura esta basada en procesadores de Intel o Dec. Los archivos del tipo "big-endian." "high-byte" con formato IEEE, son encontrados en sistemas basados en procesadores del tipo SGI, SUN, IBM RS6000 y todos los de Motorola [42]. Para el caso de este experimento el archivo estaba definido en el formato "big-endian", como la idea inicial era realizar pruebas en un PC, fue necesario realizar la conversión para "little-endian", utilizando la función `SUSWAPBYTES` exclusiva de *SU*.

Tienedo como resultado un archivo transformado que cumpla con las características necesarias para su procesamiento en un PC. Luego de realizar la transformación se verifican las características del archivo para conocer las características de la trama, utilizando la función **SURANGE** para obtener el siguiente resultado,

```
23041 traces:
tracl 1 23041 (1 - 23041)
tracr 1 23041 (1 - 23041)
fldr 3000 8975 (3000 - 8975)
tracf 1 96 (1 - 96)
cdp 1712 8875 (1712 - 8875)
cdpt 1 96 (1 - 96)
trid 1
duse 1
offset -2575 -200 (-2575 - -200)
sx 3000 8975 (3000 - 8975)
gx 425 8775 (425 - 8775)
ns 725
dt 4000
```

La información indica los valores característicos del dato, en este caso tiene 23041 trazas, con un rango de 1 hasta 23401, 96 canales de recepción, posiciones de fuentes desde 3000 a 8975 m y de geófonos desde 425 hasta 8775 m; el número de muestras en tiempo es de 725.

El archivo del campo de velocidades para Seismic Unix se ingresa como un archivo binario el cual es uno de los parámetros de entrada de los algoritmos de migración. Para ver su distribución se realizó un gráfico del campo de las velocidades utilizando la función **XIMAGE** como se muestra en la Figura 4.9.

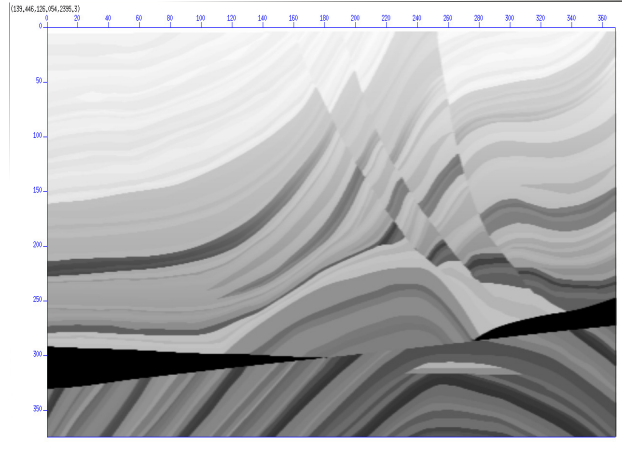


Figura 4.9: Campo de Velocidades correspondiente al dato sintético Marmousi [6]

4.3.2. Diseño del experimento

El experimento se planteo para evaluar dos resultados, en la primera parte utilizando el cluster Zine1 se evaluan los algoritmos secuenciales en el dominio del disparo y de ondas planas, luego en el mismo cluster se realiza la ejecución de las implementaciones de estos mismos algoritmos pero en paralelo.

El tamaño de los datos a procesar se mantiene constante, al igual que la arquitectura de procesamiento y conectividad del cluster. Se realiza la variación en el número de procesos paralelos que se envían al cluster para su procesamiento y se toman las mediciones de los tiempos que dura cada uno de los algoritmos en producir la respectiva salida de información, que es el archivo en formato Seismic Unix.

4.3.3. Experimentación

La etapa experimental se realizo en dos etapas:

Prueba funcional de los algoritmos de migración. Se implementaron cuatro algoritmos de migración en la plataforma Seismic Unix, ajustando las salidas a los mismos rangos de amplitud para las trazas sísmicas.

1. sumigreprespi-secuencial-normalizado.su: Migración secuencial PSPI en dominio del disparo.
2. sumigpresspsi-mpi-normalizado.su: Migración paralela PSPI en dominio del disparo.
3. supmigprespwpsi-secuencial-normalizado: Migración PSPI en dominio de ondas planas

4. `sumigpresspsi-mpi-normalizado`: Migración paralela PSPI en dominio de ondas planas.

Una vez verificado que el archivo de salida refleja adecuadamente el modelo del dato sísmico Marmoussi, se procede al montaje en la plataforma computacional Zine1. Se realizaron tres corridas de datos para los diferentes algoritmos.

El desempeño de los algoritmos, se evaluará utilizando como variables de medición el tiempo de ejecución, la eficiencia y el speedup. El tiempo de ejecución es una medición directa que tiene como variable independiente el número de núcleos utilizados en cada corrida, en contraste la eficiencia y el speedup son relaciones que dependerán de los tiempos de ejecución y los núcleos como se mostrará posteriormente.

4.3.4. Resultados

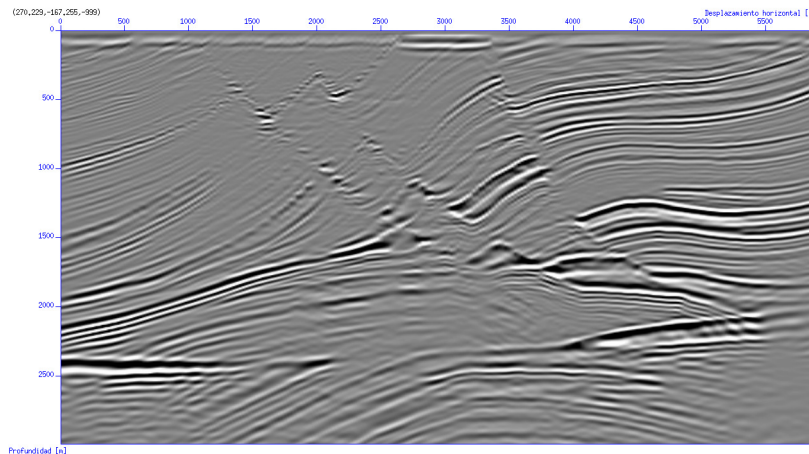


Figura 4.10: Dato Marmoussi Migrado por Dominio del Disparo

La Figura 4.10, muestra el resultado de una migración utilizando el algoritmo paralelo `sumigpresspsi-mpi-normalizado.su`. Los datos sísmicos analizados para este problema, son datos sintéticos con características especiales para realizar pruebas de desempeño de los algoritmos de migración. El presente dato tiene 240 disparos y grupos de 96 receptores, con una amplitud del campo de aproximadamente 6km. El algoritmo evaluado realiza la migración en el dominio del disparo para los 240 disparos, utilizando el algoritmo de PSPI para la migración. El campo es migrado disparo a disparo asignando a cada uno de los nodos los datos de un disparo. Como resultado el algoritmo muestra el campo migrado, destacando cada una de las estructuras, especialmente aquellas con ángulos inclinados como se ve en la parte central de la imagen. Estas estructuras debido a sus características generan un alto

grado de complejidad en el procesamiento del algoritmo ya que este va calculando su posición a medida que se va desplazando en profundidad.

Para la realización de este experimento fue utilizado el cluster Zine1, disponiendo de 340 nucleos para el procesamiento del problema. Siguiendo con la metodología PCAM planteada en la implementación del algoritmo se realizo la partición dandole a cada nucleo los datos de cada disparo.

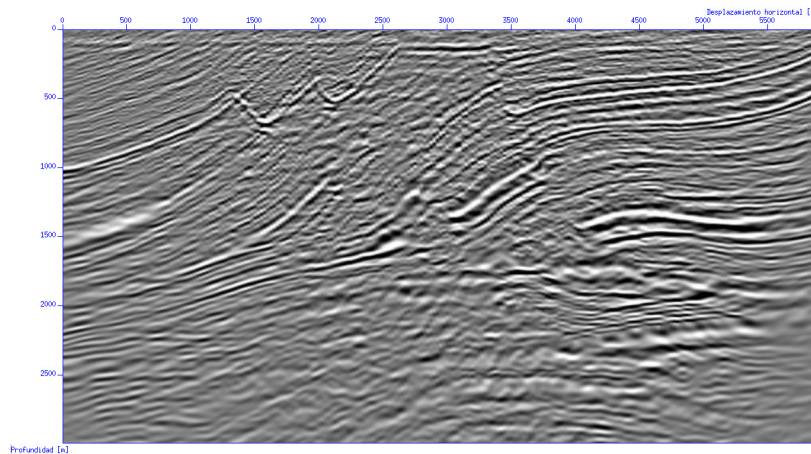


Figura 4.11: Dato Marmoussi Migrado por Dominio de Ondas Planas con $p_{min}=700us$ y 81 Migraciones

Con la finalidad de evaluar el segundo metodo de paralelismo, se realizan pruebas al mismo campo de los datos marmoussi, pero en este caso se evalua la migración por ondas planas. La Figura 4.11, muestra el resultado de realizar la migración del campo con 81 parametros ondas planas con incrementos de 700us. Para realizar estas pruebas fueron evaluados varios criterios, primero era necesario determinar los valores de los parametros de rayo que irian a iluminar el campo, para esto fue necesario definir los valores maximos, minimos y el valor del incremento de la variación del angulo con el que la onda plana iba iluminando el campo.

El resultado de la Figura 4.11, muestra la estructura del campo de una forma adecuada, realizando comparaciones entre los dos algoritmos evaluados, este presenta menos migraciones pero en consecuencia la imagen presenta algunos artefactos que pueden dificultar el analisis de la imagen final.

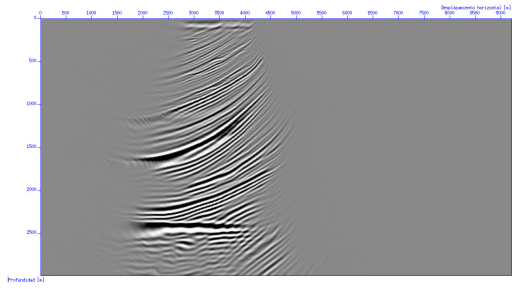


Figura 4.12: Migración de 50 Shots

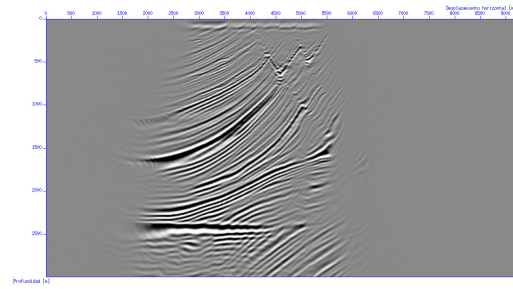


Figura 4.13: Migración de 100 Shots

Con la finalidad de observar el procesamiento del algoritmo a medida que se van migrando cada uno de los disparos se presentan las Figuras 4.12, 4.13 y 4.14. En la Figura 4.12, se realiza la migración de únicamente 50 disparos, en donde es posible ver como la extrapolación del campo en profundidad empieza a identificar la curvatura de las estructuras del campo muestreado. La Figura 4.13 son migrados 100 disparos, comienza a identificar además de las curvaturas, también picos presentes en la estructura del campo mejorando así el resultado. Finalmente la Figura 4.14 muestra la migración de 200 disparos, con un resultado parcial del campo en donde se aprecian las estructuras horizontales, verticales y picos con ángulos pronunciados.

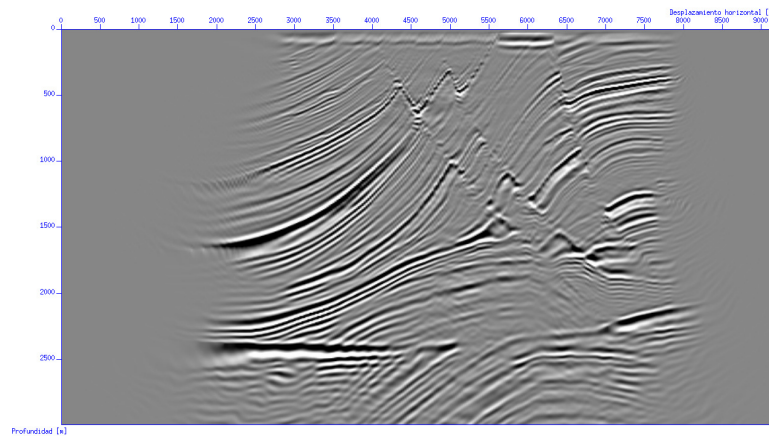


Figura 4.14: Migración de 200 Shots

Para comparar los resultados de los dos algoritmos las Figuras 4.15, 4.16 y 4.17, muestran el resultado del mismo campo migrado pero en este caso utilizando el dominio de ondas planas. A diferencia del algoritmo en el dominio del disparo, este algoritmo para cada uno de las ondas planas migradas requiere el uso de todo el campo y no de una fracción. La Figura 4.15, muestra el campo ya destacando algunas de las estructuras, especialmente líneas horizontales. En las Figuras 4.16 se muestran las estructuras del campo mucho más definidas después de haber migrado el campo 50 veces. Finalmente se realiza la migración de 101 ondas planas y se

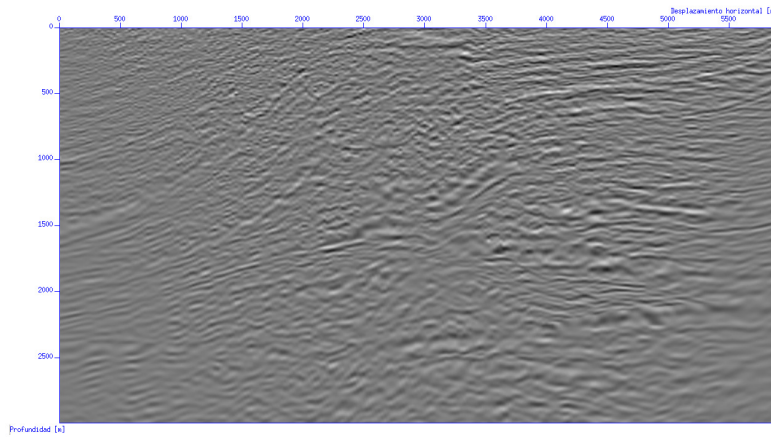


Figura 4.15: Migración de 5 Ondas Planas

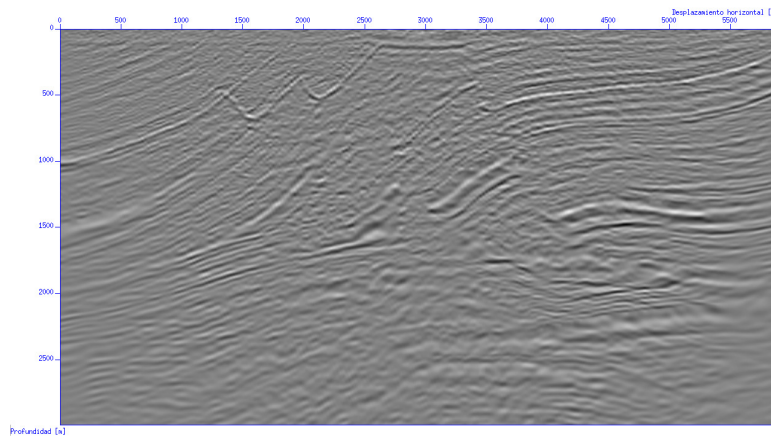


Figura 4.16: Migración de 50 Ondas Planas

obtiene como resultado la Figura 4.17, ya es posible ver el campo mas definido, ya que a medida que las ondas incidentes iluminan el campo en diferentes angulos, las distintas superficies comienzan a resaltarse de una forma mas intensa.

Durante el desarrollo de las pruebas de cada uno de los algoritmos era necesario determinar una equivalencia en migraciones para el dominio del disparo y para el dominio de ondas planas, luego con la ayuda de la información presente en [21], [39], se logro determinar que existía una equivalencia en los resultados obtenidos cuando se realiza la migración por ondas planas presentaba las siguientes características:

Parametros de rayo	101
Posición minima	-35350
Incremento	700m

Cuadro 4.4: Parametros ejecutados para migración de Algoritmo de Ondas Planas

Realizando el procesamiento con estos valores utilizando los algoritmos de ondas planas, es posible realizar una comparación entre los resultados en el dominio del

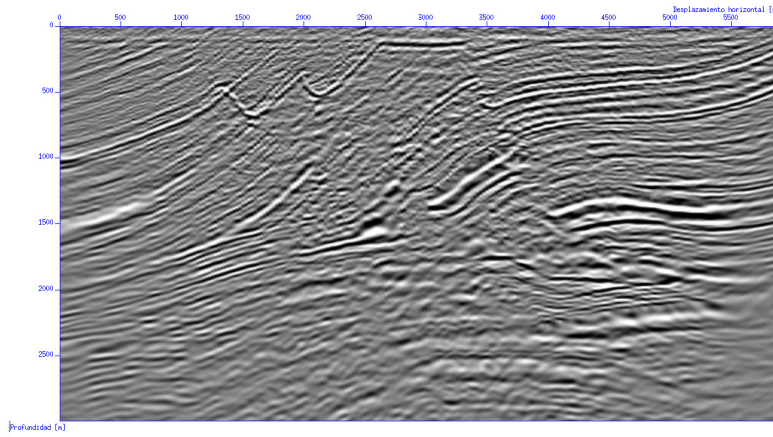


Figura 4.17: Migración de 101 Ondas Planas

disparo.

Una vez evaluados los resultados de los algoritmos de migración se realiza el análisis del desempeño de estos en el procesamiento en el cluster. El experimento fue realizado ejecutando cada uno de los algoritmos de migración primero en un núcleo y luego aumentando progresivamente la cantidad de núcleos hasta llegar a 340, en cada ejecución era medido el tiempo de procesamiento y este luego era almacenado.

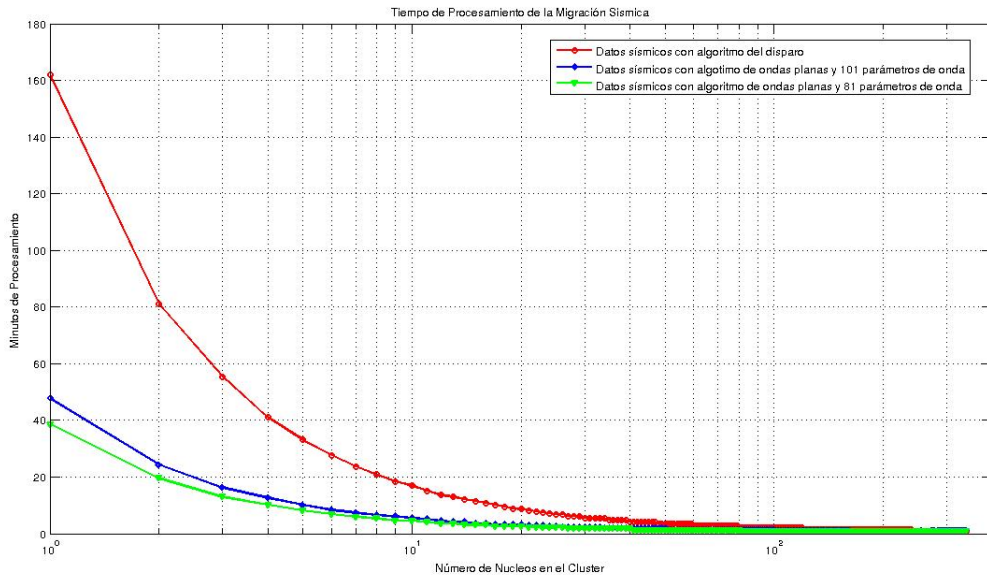


Figura 4.18: Resultado Tiempos de Ejecución en el Cluster Zine1, en dominio del disparo y ondas planas

la Figura 4.19, muestra el resultado de los tiempos de ejecución del algoritmo por disparo y ondas planas ejecutados en el cluster. Fueron colocadas tres curvas, la roja sobre la migración por disparo, la curva gris muestra la migración por ondas

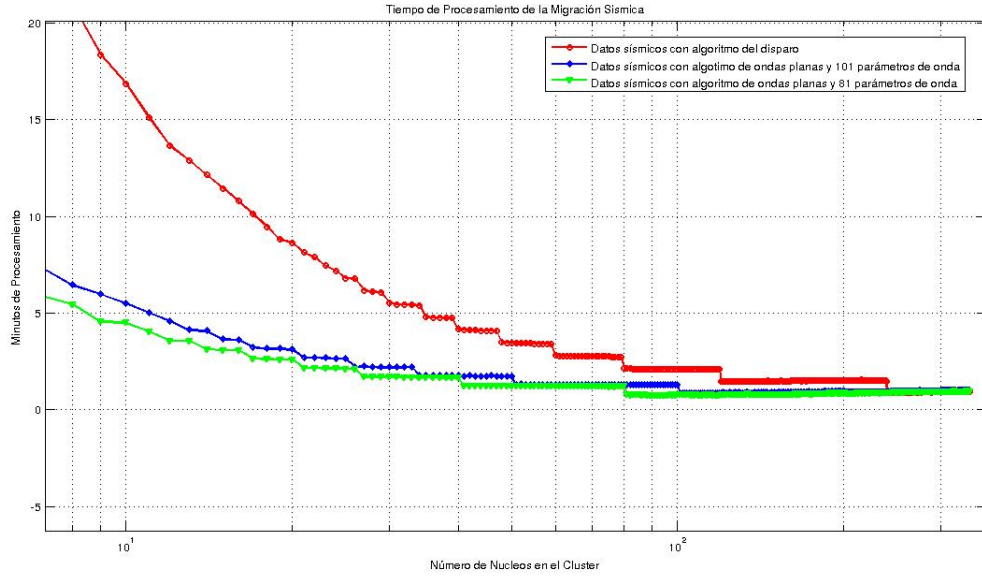


Figura 4.19: Resultado Tiempos de Ejecución en el Cluster Zine1, en dominio del disparo y ondas planas

planas para 101 migraciones, y la curva azul muestra la migración por ondas planas para 81 migraciones.

Es posible ver como a medida que se van aumentando el número de núcleos disponibles en el cluster para el procesamiento, el tiempo de ejecución va disminuyendo. Al comparar las tres graficas resultado de los algoritmos, es posible ver como el tiempo de procesamiento del algoritmo de ondas planas es menor generando un resultado adecuado para una menor cantidad de núcleos.

Otra medida fundamental para conocer el desempeño de los algoritmos en el cluster es la eficiencia. Para este experimento fueron tomados los resultados de los tiempos de ejecución presentes en la Figura 4.19 y fue aplicada la formula de la eficiencia como:

$$Eficiencia = \frac{T_s}{T_p * p} \tag{4.13}$$

Donde,

T_s : Tiempo de ejecución del algoritmo secuencial

T_p : Tiempo de ejecución del algoritmo en paralelo

p: Número de procesadores.

La ecuación 4.20, fue evaluada teniendo el tiempo de ejecución secuencial constante, y variando el tiempo de ejecución en paralelo y multiplicandolo por cada uno de los núcleos. Los resultados presentes en la Figura 4.19, muestran en el grafico rojo la

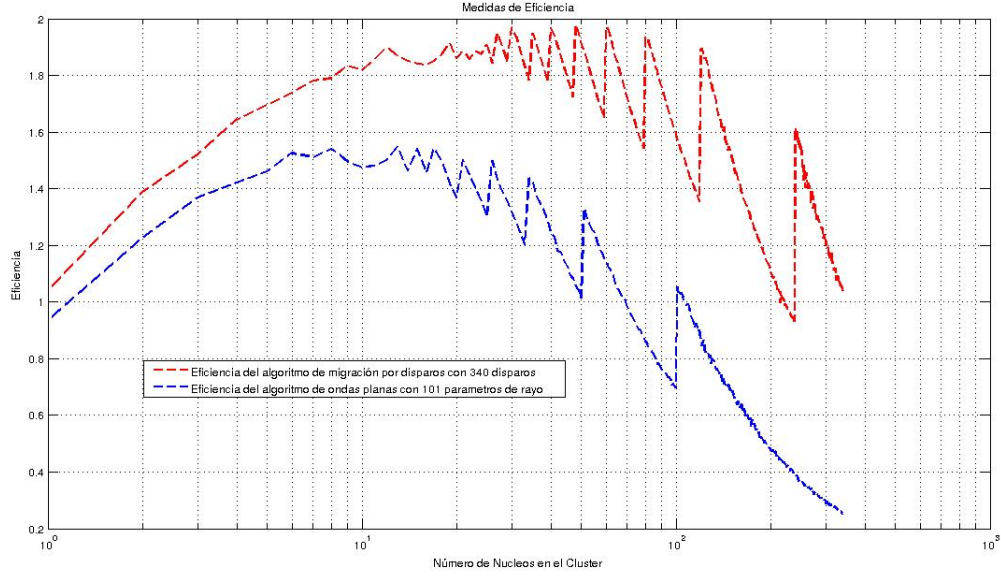


Figura 4.20: Eficiencia del uso del Cluster con el Algoritmos de Migración en el Dominio del Disparo y Ondas Planas

eficiencia para el algoritmo de migración en el dominio del disparo, este presenta un incremento a medida que aumentan la cantidad de nucleos teniendo una eficiencia maxima en 48 nucleos. En el grafico es posible ver otros picos en donde el algoritmo presenta una buena eficiencia pero estos son considerados como maximo locales que no dan un punto de eficiencia a estable y que con pequeñas variaciones lleven a tener una caída abrupta en la eficiencia.

Por otro lado el grafico azul corresponde a la eficiencia del algoritmo de ondas planas evaluado para 101 migraciones. Fue utilizada la ecuación 4.20 para el calculo de la eficiencia. En este caso este algoritmo presenta una maxima eficiencia al momento de llegar a 13 nucleos, igual que en el caso del disparo, existen otros puntos maximos, pero estos fueron considerados como maximos locales.

Entre los analisis de datos fueron evaluadas tambien las características de SpeedUp. Esta medida relacionará el tiempo del algoritmo secuencial y el tiempo del algoritmo implementado en el cluster como se muestra en la ecuación 4.14.

$$Speedup = \frac{T_s}{T_p} \quad (4.14)$$

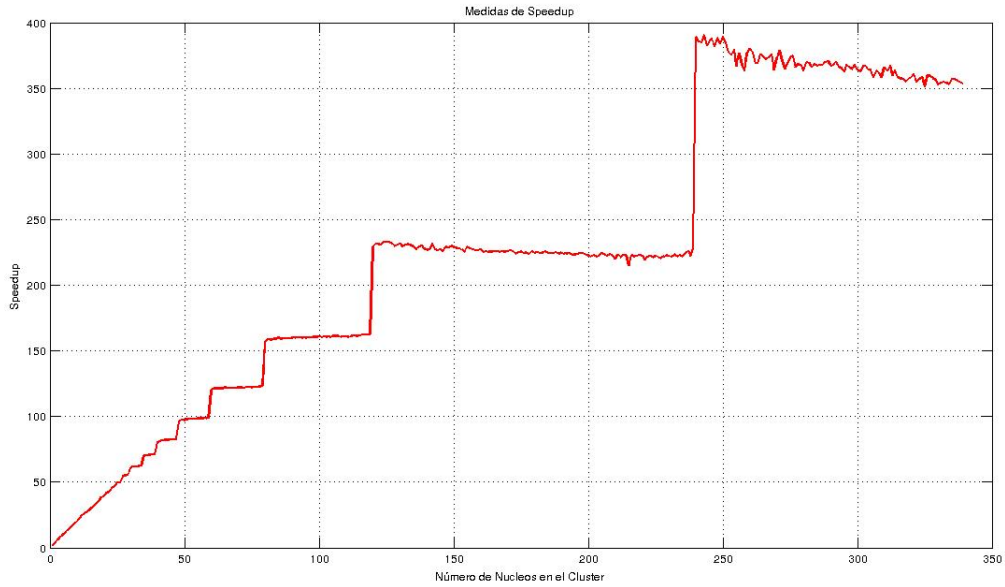


Figura 4.21: Speedup del uso del Cluster con el Algoritmo de Migración por Disparo

La Figura 4.21, muestra el comportamiento del algoritmo en paralelo con respecto al algoritmo secuencial en el dominio del disparo, el grafico presenta dos características, primero muestra escalones al aumentar la cantidad de procesadores que hacen ver como el algoritmo en paralelo se acelera varias veces con respecto al secuencial, pero de la misma forma presenta secciones planas en las que a pesar de aumentar la cantidad de procesadores, no se presenta ningun tipo de cambio o por el contrario tiende a disminuir la medida como es el caso entre 120 y 245 procesadores aproximadamente.

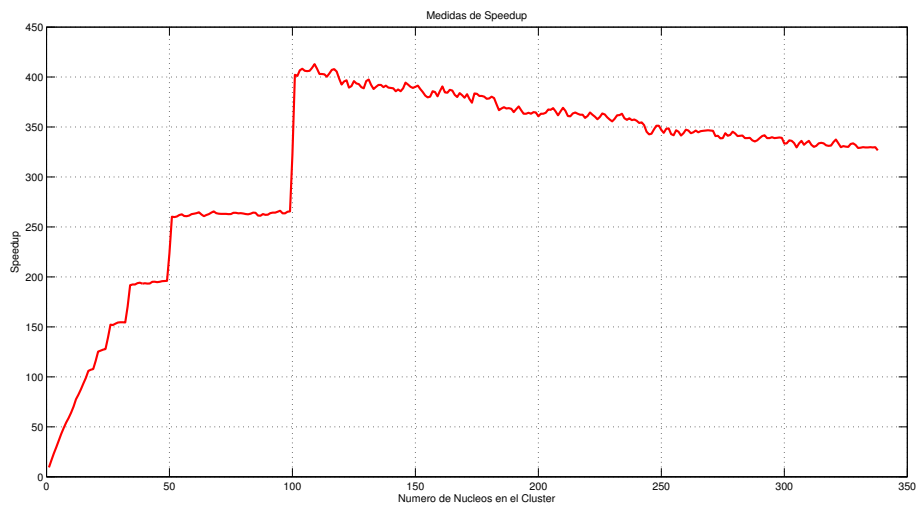


Figura 4.22: Speedup del uso del Cluster con el Algoritmo de Migración por Ondas Planas

La Figura 4.22, muestra el comportamiento del algoritmo en paralelo del algoritmo en el dominio de las ondas planas. El grafico presenta una alta tasa de aceleración en el proceso en los primeros 40 nodos aproximadamente, luego inicia un comportamiento de escalones en los cuales se mantiene estable o en ocasiones reduce la variación.

Cada uno de los resultados aqui presentados para la evaluación del desempeño de los algoritmos en el cluster fue normalizado con la finalidad de poder comparar los resultados entre cada uno de los algoritmos.

Capítulo 5

Conclusiones

El presente trabajo realizó el análisis de la metodología de desarrollo paralelo PCAM e identificó tres características de diseño fundamentales al analizar los puntos fundamentales de decisión: La granularidad, la comunicación y la redundancia. Con estas características se identificaron cuatro estrategias propuestas de desarrollo de algoritmos paralelos enfocados a las plataformas distribuidas grilla y clúster.

De las estrategias propuestas, se realiza la selección de la estrategia número tres 3.4 la cual se caracteriza por una granularidad gruesa, comunicación global y estructurada y baja replicación de cómputo y datos. Esta selección se realizó teniendo en cuenta los criterios establecidos de plataforma y tiempo de implementación y el grado de innovación posible.

Se presenta como caso de experimentación el análisis de un algoritmo existente en el campo de la Exploración Sísmica en la industria de Petróleo y Gas, específicamente en el procesamiento de datos sísmicos para el proceso de migración sísmica: el algoritmo de migración en dominio del disparo.

Se evalúa bajo la estrategia identificada, un nuevo enfoque geofísico con el objetivo de generar un diseño alternativo al existente y realizar su implementación utilizando la plataforma experimental Seismic Unix en el cluster Zine 1 del Centro de Alto Rendimiento Computacional - ZINE - de la Pontificia Universidad Javeriana.

El desarrollo realizado se basa en un particionamiento diferente a la implementación encontrada en el dominio de las ondas planas. Para el caso de estudio se realizó el diseño soportado en la metodología PCAM orientado con la estrategia seleccionada; igualmente se realizó la implementación y se verificó la viabilidad del diseño y su funcionalidad con un conjunto de datos sintéticos de dominio público, evaluando sus características de desempeño en tiempo de ejecución, eficiencia y speedup.

Como trabajo futuro se identifican las siguientes oportunidades:

1. Posibilidad de ampliar el estudio de las estrategias identificadas con el fin de identificar patrones de diseño paralelo que permitan agilizar el desarrollo de software paralelo orientado a otras arquitecturas paralelas existentes hoy en día como arquitecturas híbridas.
2. Verificación de la escalabilidad del algoritmo implementado en el dominio de Ondas Planas en conjuntos de datos sintéticos y reales con una dimensión similar a la productiva y desarrollar un análisis de desempeño del mismo con respecto a otras implementaciones.
3. Viabilidad de continuar con la investigación en el campo de la paralelización de algoritmos de procesamiento sísmico con el uso de nuevas estrategias que permitan mejores rendimientos y la incorporación de éstos a plataformas como Seismic Unix u otras herramientas comerciales.
4. Extender las implementaciones a migraciones de datos sísmicos 3D las cuales representan un gran reto en relación al manejo de datos el cual es del orden de Terabytes.

Bibliografía

- [1] TOP500. *Top500 November 2013*. Technology, TOP500 BOF, 2013.
- [2] ECOPETROL. *Generalidades del Proceso de Exploración Sísmica*. Documento técnico, Ecopetrol, 2012.
- [3] ECOPETROL. *Memorias Curso de Geofísica para no Geofísicos*. Documento técnico, Ecopetrol, 2013.
- [4] YILMAZ, Ö., DOHERTY, S. *Seismic Data Analysis: Processing, Inversion, and Interpretation of Seismic Data*. N. v. 1, Investigations in geophysics. Society of Exploration Geophysicists, 2001. ISBN: 9781560800989. Disponible en: <http://books.google.com.br/books?id=ceu1x3JqYGUC>.
- [5] FRAZER BARCLAY, ANDERS BRUUN, K. B. R. J. C. A. *Inversión Sísmica: Lectura entre Líneas*. Documento técnico, Schlumberger, Verano de 2008.
- [6] IFP. *Marmousi Synthetic Dataset*. Documento técnico, institut Français du Pétrole, 1990. Presentado en el Encuentro No. 52 de la EAEG en 1990.
- [7] “Proyectos | ZINE - Centro de Alto Rendimiento Computacional Javeriano”. Disponible en: <http://zine.javeriana.edu.co/?pageid=369>.
- [8] LASTOVETSKY, A. “Heterogeneity in parallel and distributed computing”, *Journal of Parallel and Distributed Computing*, v. 73, n. 12, pp. 1523–1524, dec 2013. ISSN: 07437315. doi: 10.1016/j.jpdc.2013.08.010.
- [9] FOSTER, I. *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. ADDISON WESLEY Publishing Company Incorporated, ene. 1995. ISBN: 9780201575941. Disponible en: <http://www.mcs.anl.gov/itf/dbpp/>.
- [10] DE PLANEACIÓN, D. N. “Plan Nacional de Desarrollo 2010-2014- Retos y Metas para el Sector Minero - Energético” . .

- [11] SKILLICORN, D. B. “Strategies for parallel data mining”, *IEEE Concurrency*, 1999.
- [12] Atallah, M. J., Fox, S. (Eds.). *Algorithms and Theory of Computation Handbook*. 1st ed. Boca Raton, FL, USA, CRC Press, Inc., 1998. ISBN: 0849326494.
- [13] LIN, Y., SNYDER, L. *Principles of parallel programming*. Pearson/Addison Wesley, 2008. ISBN: 9780321487902. Disponible en: <http://books.google.com.co/books?id=iYghAQAIAAJ>.
- [14] “Grupo de Investigación SIDRe (Sistemas Distribuidos y Redes de Computadores)”. .
- [15] AKL, S. G., NAGY, M. “Introduction to Parallel Computation”. In: Trobec, P. R., Vajteric, D. M., Zinterhof, P. D. P. (Eds.), *Parallel Computing*, Springer London, pp. 43–80, jan 2009. ISBN: 978-1-84882-408-9, 978-1-84882-409-6. Disponible en: <http://link.springer.com/chapter/10.1007/978-1-84882-409-62>.
- [16] STANOEVSKA-SLABEVA, K., WOZNIK, T. “Cloud Basics - An Introduction to Cloud Computing”. In: Stanoevska-Slabeva, K., Wozniak, T., Ristol, S. (Eds.), *Grid and Cloud Computing*, Springer Berlin Heidelberg, pp. 47–61, 2010. ISBN: 978-3-642-05193-7. Disponible en: <http://www.springerlink.com/content/g461223v1283g848/abstract/>.
- [17] MAGOULES, F., PAN, J., TAN, K.-A., et al. *Introduction to Grid Computing*. CRC Press, ene. 2009. ISBN: 9781420074079.
- [18] WILKINSON, B. *Grid computing: techniques and applications*. Boca Raton, CRC Press, 2010. ISBN: 9781420069532 1420069535.
- [19] ANTONOPOULOS, N., GILLAM, L. *Cloud Computing: Principles, Systems and Applications*. Springer, jul. 2010. ISBN: 9781849962414.
- [20] MUDGE, J. C., CHANDRASEKHAR, P., HEINSON, G. S., et al. “Evolving Inversion Methods in Geophysics with Cloud Computing-A Case Study of an eScience Collaboration”. .
- [21] F. LIU, D. HANSON, N. W. R. D. R. S. “Toward a unified analysis for source plane-wave migration”. In: *GEOPHYSICS, VOL. 71, NO. 4*, Julio-Agosto 2006.

- [22] JANEI, D., BORTNIK, U., PRAPROTNIK, M. “Parallel Approaches in Molecular Dynamics Simulations”. In: Trobec, P. R., Vajteric, D. M., Zinterhof, P. D. P. (Eds.), *Parallel Computing*, Springer London, pp. 281–305, ene. 2009. ISBN: 978-1-84882-408-9, 978-1-84882-409-6. Disponible en: <http://link.springer.com/chapter/10.1007/978-1-84882-409-610>.
- [23] KUTIL, R. “Short-Vector SIMD Parallelization in Signal Processing”. In: Trobec, P. R., Vajteric, D. M., Zinterhof, P. D. P. (Eds.), *Parallel Computing*, Springer London, pp. 397–433, ene. 2009. ISBN: 978-1-84882-408-9, 978-1-84882-409-6. Disponible en: <http://link.springer.com/chapter/10.1007/978-1-84882-409-613>.
- [24] “Remote Computational Tools for Radiotherapy Cancer Treatment Planning”. In: Stanoevska-Slabeva, K., Wozniak, T., Ristol, S. (Eds.), *Grid and Cloud Computing*, Springer Berlin Heidelberg, pp. 147–158, ene. 2010. ISBN: 978-3-642-05192-0, 978-3-642-05193-7. Disponible en: <http://link.springer.com/chapter/10.1007/978-3-642-05193-79>.
- [25] VARGAS, C. A. “Evaluating total Yet-to-Find hydrocarbon volume in Colombia”, *Earth Sciences Research Journal*, v. 16, n. Special Issue, pp. 1–246, 2012. ISSN: 1794-6190.
- [26] HAGER, GEORG, W. G. *Introduction to high performance computing for scientists and engineers*. Boca Raton [u.a., CRC Press, 2011. ISBN: 9781439811924 143981192X 9781439811931 1439811938 9786612902772 6612902779.
- [27] “Convocatoria para Conformar un Banco de Programas Estratégicos y Proyectos Elegibles de Investigación con Enfoque de Mercado, Desarrollo Tecnológico e Innovación en la modalidad de cofinanciación”. Disponible en: <http://www.colciencias.gov.co/convocatoria/>.
- [28] QUINN, M. J. *Parallel Programming in C with MPI and OpenMP*. McGraw-Hill Higher Education, 2004. ISBN: 9780072822564.
- [29] MATTSON, T., SANDERS, B., MASSINGILL, B. *Patterns for Parallel Programming*. Software Patterns Series. Pearson Education, 2004. ISBN: 9780321630032. Disponible en: <http://books.google.com.co/books?id=LNcFvN5Z4RMC>.
- [30] AKL, S. *Diseño y análisis de algoritmos paralelos*. Colección Informática Fundamentos Teóricos: serie Paradig-

- ma. Ra-Ma, 1992. ISBN: 9788478970476. Disponible en: <http://books.google.com.co/books?id=uRzbAAAACAAJ>.
- [31] FOWLER, P. J. *SEISMIC VELOCITY ESTIMATION USING PRESTACK TINIE MIGRATION*. Tesis de Doctorado, Stanford University, 1988. Disponible en: <http://sep.stanford.edu/data/media/public/docs/sep58/5800.pdf>.
- [32] STEIN, E. M., SHAKARCHI, R. *Princeton Lectures in Analysis*. Princeton University Press, 2003. Disponible en: <http://www.ams.org/notices/201205/rtx120500641p.pdf>.
- [33] MATTSON, T. G., SANDERS, B. A., MASSINGILL, B. *Patterns for parallel programming*. Boston, Addison-Wesley, 2005. ISBN: 0321228111 9780321228116 0321940784 9780321940780.
- [34] DO ROSÁRIO, D. A. N. *Escalabilidade Paralela de um Algoritmo de Migração Reversa no Tempo (RTM) Pré-Empilhamento*. Tesis de Maestria, Universidade Federal do Rio Grande do Norte, 2012.
- [35] MEJIA, F. A. V. *Migração pré-empilhamento através das equações de ondas unidireccionais com amplitude verdadeira*. Tesis de Doctorado, Universidade Federal da Bahia, Curso de Pos-graduação em Geofísica, Agosto de 2008.
- [36] ALKHALIFAH, T. “Prestack phase-shift migration of separate offsets in laterally inhomogeneous media”, *Geophysics*, May - June 2010.
- [37] DEMREL, D. “HIGH PERFORMANCE COMPUTING CLUSTER ARCHITECTURES”. In: *CMPE 511 COMPUTER ARCHITECTURE*.
- [38] MIGUEL L. BOTE-LORENZO, Y. A. D., GÓMEZ-SÁNCHEZ, E. “Grid Characteristics and Uses: a Grid Definition”. In: *Postproc. of the First European Across Grids Conference*.
- [39] LOUIS A. ROMERO, DENNIS C. GHIGLIA, C. C. O., MORTON, S. A. “Phase encoding of shot records in prestack migration”. In: *GEOPHYSICS, VOL. 65, NO. 2*, March-April 2000.
- [40] SHAN, G., BIONDI, B. “Plane-wave migration in tilted coordinates”. In: *Geophysics Vol 73*, 2008.
- [41] JOHN W. STOCKWELL, J. “A Course in geophysical Image Processing with Seismic Unix, GPGN 461/661 Lab”. Septiembre de 2013. Disponible en: <http://www.cwp.mines.edu/john/GPGN461.561/ch1-3.pdf>.

[42] JHON W. STOCKWELL, J. . J. K. C. “The New SU User’s Manual”. Enero 2008.