CAPÍTULO 11

SEISMIC IMAGING USING FPGA APPLIED FOR REVERSE TIME MIGRATION

Data de aceite: 02/06/2020
Data de submissão: 05/02/2020

Joaquim Ranyere Santana de Oliveira
SENAI/CIMATEC
Salvador – Bahia
http://lattes.cnpq.br/8072549971315902

João Carlos Nunes Bittencourt
Universidade Federal do Recôncavo da Bahia
Cruz das Almas – Bahia
http://lattes.cnpq.br/6741995694783822

Deusdete Miranda Matos Junior
SENAI/CIMATEC
Salvador – Bahia
http://lattes.cnpq.br/5317308859913857

Anderson Amorim do Nascimento
SENAI/CIMATEC
Salvador – Bahia
http://lattes.cnpq.br/8000911911840455

Laue Rami Souza Costa de Jesus
SENAI/CIMATEC
Salvador/BA
http://lattes.cnpq.br/3794131382143818

Georgina Gonzalez Rojas
SENAI/CIMATEC
Salvador – Bahia
http://lattes.cnpq.br/6526414323159768

Rodrigo Carvalho Tutu
SENAI/CIMATEC
Salvador – Bahia
http://lattes.cnpq.br/1203041686952391

Wagner Luiz Alves de Oliveira
Universidade Federal da Bahia
Salvador – Bahia
http://lattes.cnpq.br/7355315368234452

Silvano Moreira Junior
SENAI/CIMATEC
Salvador – Bahia
http://lattes.cnpq.br/3082139503007423

ABSTRACT: Reverse time migration (RTM) modeling is a computationally intensive component in the seismic processing workflow of oil and gas exploration. The RTM algorithm demands the manipulation of a large amount of data. Therefore, the computational kernels of the RTM algorithms need to access a large range of memory positions, leading to a large amount of cache misses, degrading the overall system performance. This work presents a FPGA accelerated platform targeting the computation of the RTM algorithm on a HPC environment. Experimental results highlight that speedups of 112x can be achieved, when compared to a sequential execution

on CPU. When compared to a GPU, the power consumption has been reduced up to 55%.
KEYWORDS: Supercomputing, RTM, FPGA, Green computing, Seismic Migration.
RESUMO: A modelagem de Migração Reversa no Tempo (RTM) é um componente de alto custo computacional dentro do fluxo de trabalho de processamento sísmico, na exploração de petróleo e gás. Os algoritmos RTM exigem a manipulação de uma grande quantidade de dados, levando a degradação do desempenho devido aos acessos a endereços não aninhados. O presente trabalho apresenta uma plataforma de aceleração em FPGA para o cálculo do algoritmo RTM em um ambiente HPC. Resultados experimentais destacam que acelerações de 112x podem ser alcançadas, quando comparadas à execução sequencial do algoritmo em CPU. Quando levado em consideração o consumo de energia, a plataforma proposta apresentou uma redução de 55% em relação à GPU.
PALAVRAS-CHAVE: Supercomputação, RTM, FPGA, Computação verde, Migração Sísmica.

IMAGEAMENTO SÍSMICO COM FPGA APLICADO A MIGRAÇÃO REVERSA NO TEMPO

1 I INTRODUCTION

Reconfigurable systems are becoming a key component in high-performance computing systems. The reconfiguration capability increases the computing flexibility, allowing for different solutions to be deployed and tested on-the-fly. This ability is useful for systems that require a combination of high processing power with remote reconfiguration at circuit level. Reconfigurable systems use highly flexible computing fabric, usually Field-Programmable Gate Arrays (FPGA). Nowadays, FPGA devices are adopted in a wide spectrum of applications, ranging from consumer electronics to critical systems such as vehicle automation or mobile network base station transceivers (ATITALLAH *et al.*, 2017; SEXTON *et al.*, 2017). FPGAs are currently also being embedded into HPC cluster nodes starting a new revolution in high performance and cloud computing. Toward this, recent works highlight not only how fast they are, but also its reduced cost and better energy efficiency when compared to CPU and GPU counterparts (KOBAYASHI *et al.*, 2012).

Seismic imaging algorithms for oil exploration, such as RTM (Reverse Time Migration), are a time consuming and memory bound class of application that usually demand large amounts of computational power. Although current computing capabilities meet the requirements for RTM algorithm on production environments, the cost of processing such large amount of data is high and must be taken into account. Efforts targeting the reduction of these costs by using different acceleration strategies are needed. Since the RTM algorithm uses Finite Differences (FD) method, any advancement in accelerating FD kernels may reduce design time of third-party solutions.

The present work introduces a high performance hardware/software co-design that provides a scalable model for seismic imaging. The main goal is to provide an analysis of such an implementation performance and power efficiency against other accelerators. For that it has been used a custom method for 2-D RTM, namely Hybrid Boundary Condition

(HBC), which optimizes memory bandwidth and allows to explore a pipelined architecture (LIU *et al.*, 2013). This design provides a high performance and powerful efficient alternative for accelerating seismic imaging processes through a combination of state of the art RTM techniques and hardware-oriented optimizations.

2 I REVERSE TIME MIGRATION

Reverse Time Migration (RTM) is a seismic imaging technique to map the subsurface reflectivity using recorded seismic waveforms. The practice in exploration seismology has established a two-fold approach of seismic imaging: using velocity modeling building to establish the long-wavelength reference velocity models, and using seismic migration to map the short-wavelength reflectivity structures (ZHOU *et al.*, 2018). Especially in complex geological settings, RTM produces better images than other methods. In addition, RTM is the only method that is capable to use all seismic wave types that can be computed numerically (ZHOU *et al.*, 2018).

The migration step can be done before or after stacking and in time or in depth. Basically, in every migration there are two common steps: (1) wavefield extrapolation; and (2) application of an image condition. The extrapolation of the wavefield is a mathematical technique which allows to delay or advance the wavefield. Extrapolations in time are called reverse time extrapolation when the prediction is made for an earlier time; otherwise, when the prediction is made at a later time, the extrapolation is called forward time propagation. Such extrapolations are performed from the solution of the wave equation, which can be seen as a way to make explicit the reverse/forward propagation of the field from measurements made in a given position. Although wavefield modeling can also be performed in the frequency domain, its application to large scale problems is prohibitive due to the implied high computational cost. The image condition is the criterion applied to the extrapolated field to obtain a subsurface image by showing the reflecting points of the seismic energy through the correct positioning of the amplitudes in the migrated section, related to the image of the form.

The RTM method is based on the solution of the full wave equation and uses the condition of cross-correlation image (ZHOU *et al.*, 2018). This image condition was proposed by Clarebout (1971) and establishes that reflectors are located at points in the subsurface where the wavefield of the source coincides in time and space with the receiver wavefield. In this case, the wavefield of the source is propagated, the receiver field is reverse-time extrapolated, and the cross-correlation between both is calculated. At the points where the fields coincide, the image condition will be nonzero indicating the presence of a reflector in that position. For a more in-depth study on RTM refer to (ZHOU *et al.*, 2018), which summarizes the key contributions of representative publications about this method.

The most significant effort toward hardware acceleration of RTM algorithm optimization is reducing the storage requirements. The RTM models proposed in (LIU *et al.*, 2013; ZHOU *et al.*, 2018) indicate the realization of both reverse propagation using the final snapshot and random boundary as the most suited method in order to reduce such storage requirements.

The random boundary fundamentals are based on one copy of the source and receiver wavefields. Since only one copy of such wavefields need to be stored into a memory system, such a technique becomes suitable for encapsulating the entire RTM computation into a single FPGA accelerator instance.

The random boundary technique was presented in Clapp (2009), where the authors replaced the conventional damped region with an increasingly random velocity region. Rather than eliminate the reflections, the method distorts them in order to reduce coherent correlations with the receiver wavefield. Since the wavefield of the boundaries is distorted, it does not coherently correlate with the receiver wavefield. The main disadvantage of random boundary lies in the need for the Free Surface Boundary Condition (FSBC), since the Random Boundary Condition (RBC) induces severe noise along the imaging profile at the surface boundary.

The method presented in Liu (2013) proposes a boundary condition to reconstruct the shot wavefield called Hybrid Boundary Condition (HBC), which adds an absorbing boundary condition to the upper boundary. The HBC scheme aims to solve the free surface boundary problem without increasing memory storage requirements. This methodology requires saving the upper boundary of the source wavefield. The saving boundary technique stores only $d/2$ layers of data at every surrounding boundary, where d is the order of the finite-difference scheme. Additionally, by saving layers of the upper side, it is possible to decrease the computational cost. Experimental results also show the effectiveness of the HBC to obtain good images of structures at subsurface, as well as the decrease of random noise present in the final results when compared against the RBC (LIU *et al.*, 2013). Therefore, the HBC with random and saving boundary was defined as our system methodology implementation model.

3 I DESIGN ARCHITECTURE

The reference application for the RTM method was designed using C language for both migration and modeling steps. The algorithm was designed from the HBC method, since this strategy better fits the hardware needs without losing image quality. The workload and the data structure management were built to be adapted to other computing architectures, such as FPGA and GPU.

The prototype is composed of a computing node with two Intel Gold 6148 processors and an Intel Arria 10 GX FPGA Development Kit. The hardware component set is composed by the RTM IP-core, an Avalon-MM Interconnection, a DDR4 Memory Interface, and a PCIe controller. The RTM core is the proposed hardware implementation of the HBC RTM functionality.

The Figure 1 presents an overview of the RTM IP-core components designed. The RTM core corresponds to the implementation of the RTM processing element on the target FPGA device. This core is composed of three functional components: (1) a Main Control System; (2) the Scalable Streaming Array (SSA) of Pipelined Stage Modules (PSM); and (3) a Cross-correlation mechanism.

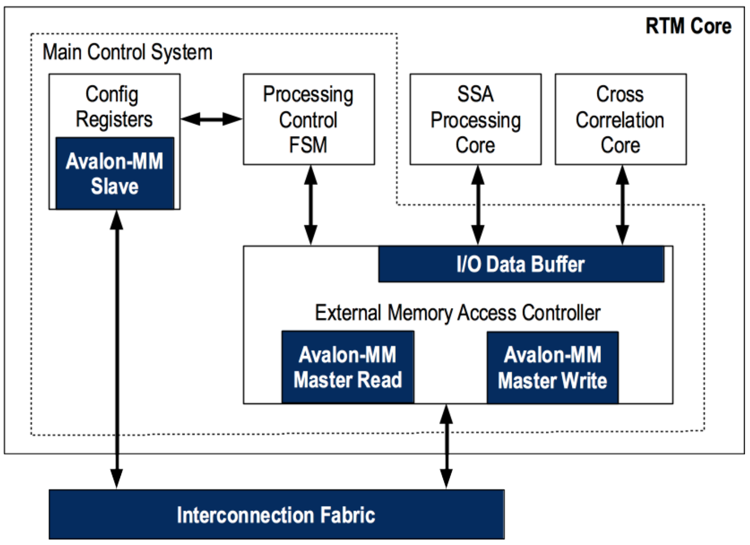


Figure 1. RTM Core main components and interface logical structure.

The Main Control System controls the internal data flow and external memory access through a control interface, a set of configuration registers and the External Memory Access Controller. The Processing Control FSM handles the internal data flow and the external memory access. Such control system mediates the data exchange between the external memory and both the SSA and cross-correlation components. The operation begins when the host software sends the initialization signal, causing specific submodules of the main control system to start their operation by reading the memory data. This data is later stored into several internal I/O data buffers and sent to the PSM units inside the SSA and the cross-correlation. At the same time, when a PSM or cross-correlation module needs to write its data into the device memory, the I/O data buffers provide a mechanism for data to be written and to be sent to memory after that.

The SSA processing core is the main processing unit of the RTM Core, responsible for performing the time steps of the stencil computation. Such a processing is accomplished through several consecutive PSM, which compute the forward and reverse propagation as well as the image reconstruction of the RTM algorithm. The PSM is a multiple stencil operator and multiple time step hardware implementation strategy (SANO; HATSUDA; YAMAMOTO, 2014). When operating in forward propagation mode, the SSA module can process up to N time steps, where N is the number of PSM components, starting with the time step $T = t + 2$. By processing the time steps from $t + 2$ to $t + 2 + N - 1$ in a pipeline approach, the PSM module calculates $t + 2 + N$. After the computation of the source wavefield time steps, the data values for $T = t + 2 + N$ and $T = t + 2 + N + 1$ are stored into the external memory, in order to calculate the following time steps starting from $T = t + 2 + N + 2$ until the SSA finishes the computation of all time steps. In the reverse propagation, the PSM are divided into two groups of $N/2$. The PSM units ranging from 0 to $(\frac{N}{2} - 1)$ are responsible for calculating the reverse propagation and the PSM units ranging from $\frac{N}{2}$ to $N - 1$ are responsible for the reconstruction of the propagated signal. The results of such calculations are forwarded to the correlation module for the image condition evaluation process. The advantage of such a strategy relies on both spatial and temporal scalability working together. Internally, the PSM is composed of several Processing Elements (PEs) responsible for the current RTM computation. Such a structure can be replicated in order to compute more points in parallel at the same iteration as well as streaming out points to next PSM module without needing additional memory access operations.

The cross-correlation core performs the computation needed to assess the resulting image quality, which is carried out by the cross-correlation of the wavefields based in reflectors located on the subsurface. The RTM is based on the computation of a full wave equation and uses the condition of cross-correlation image. Such an image condition establishes that reflectors are located at points in the subsurface where the wavefield of the source coincides in time and space with the receiver wavefield. The correlation in each shot is defined by Equation 5, where P_s is the source wavefield in forward propagation, and P_r the receiver wavefield in reverse propagation. The final image is given by the sum of the images of all single shots.

$$I_s(z,x) = \sum_{t=0}^{T} P_s(z,x,t) P_r(z,x,t) \tag{5}$$

4 I RESULTS

In this section, experimental results for the conceived RTM algorithm hardware acceleration platform are presented. The results were obtained using the Intel Quartus Prime Pro Edition (v18.0), while the design itself was described using SystemVerilog-HDL language. The hardware platform is an Intel Arria 10 GX FPGA Development Kit, with a single 4 GB DDR4 external memory. The experiment analysis considered the Pluto velocity model. Speed up and power measurements considered migrations of four seismic shots evenly spaced over the model surface.

For an optimized memory bandwidth, the RTM IP-core uses fixed-point numerical representation. An optimal word size for the hardware was determined by comparing the outputs of a fixed-point RTM software with its 32-bit floating-point equivalent, used as reference. Satisfactory migration images were obtained when SNR values ranged between -10 and -20 dB and with IQI above 70%. Thus, 24-bit was the smallest length that met the quality requirements. The total power measure was generated using the Watts Up Pro Portable Power Meter during the runtime of both GPU and FPGA accelerators. The measured data were obtained by multiplying the measured mean power with the total runtime in seconds.

The Table 2 presents the design performance estimated through synthesis results and system runtime on both FPGA and CUDA GPGPU kernel. The values for the Arria 10 were obtained after place & route processing and static timing analysis with parameters for timing optimization, namely Performance (high effort).

When compared to the serial implementation, a speedup of about 112x can be achieved by using the proposed FPGA accelerated platform. Regarding the other implementation strategies the GPU is only 9% faster, while the runtime of Multi-thread MPI/OMP is about 26% lower. Although FPGA presents such performance decrease, it runs on around eight times slower frequency.

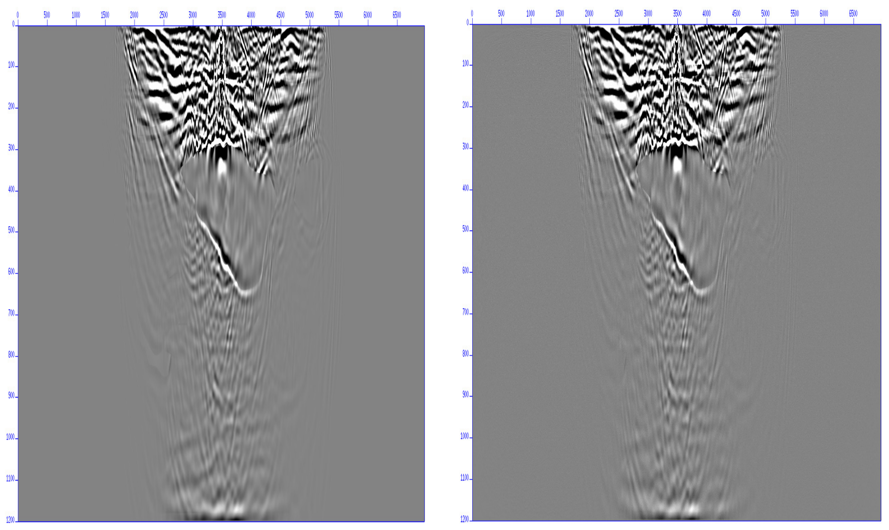
Configuration	Platform	Runtime (s)	Speedup	Energy (Wh)	Efficiency (Speedup/W)
Sequential CPU	Intel Xeon Gold 6148	21,873	1	N/A	N/A
Multi-thread with 16 nodes and 40 threads	Intel Xeon Gold 6148	145.82	150	N/A	N/A
CUDA	NVIDIA Titan XP	706	123.9	36	3.4
FPGA HDL code (180MHz)	Intel Arria 10 GX	781	112.0	**20**	**5.6**

Table 2. RTM Accelerated Platform performance analysis and comparison.

Another FPGA advantage is the energy efficiency. The Titan XP GPU running at 1,417 MHz have presented a total energy of 36 Wh, while the Arria 10 GX FPGA at 180 MHz resulted in 20 Wh. The FPGA implementation is more energy-efficient, by consuming about 55% less power when compared to the GPU. Considering the speedup per Watt as an efficiency metric, an efficiency of 5.6 Speed-up/Wh can be achieved by using our acceleration solution, against 3.4 Speedup/Wh obtained by GPU. Thus, the FPGA implementation leads to a design 60% more efficient.

The prototype tests have aimed at validating functions of the communication and image stacking. Initially, memory transfers were made between the host computer and the FPGA in order to guarantee the correctness of the device driver. The bench tests resulted in the images presented in Figure 2.

Figure 2a represents the final image obtained by the software reference model using a 32-bit floating-point representation. Figure 2b is the resulting image obtained after processing the Pluto velocity model on the FPGA accelerated platform.



(a) Reference 32-bit Float (b) FPGA

Figure 2. Comparison between reference and accelerated FPGA designs output.

5 I CONCLUSION

This paper presented an FPGA accelerated platform targeting the computation of the RTM algorithm on an HPC environment. Hardware implementation aspects were correlated with RTM computation optimizations present in the literature, resulting in the proposed co-design implementation. The main techniques used were the multiple stencil operator and multiple time step hardware implementation strategy as well as the Hybrid Boundary Condition method. The application runs on top of a GNU/Linux operating system, which provides built-in API for memory management and peripheral device access. The FPGA IP-core is a hardware accelerator that receives input data, processes seismic migration, and then sends back the resulting output data. The analysis of performance and power consumption, compared against GPU and CPU, highlights that speedups of 112x can be achieved, when compared to a Sequential CPU implementation. Although the design present lower speedup compared to GPU and CPU multi-threaded, our FPGA accelerator achieved better energy efficiency. The power consumption when compared to a GPU has been reduced up to 55% with an efficiency 60% greater.

Acknowledgment

The research described in this chapter was funded by the Brazilian Company for Industrial Research and Innovation (EMBRAPII), Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB - 5903/2018) and Shell/BG E&P Brasil LTDA. The work was also supported by the SENAI / CIMATEC Supercomputing Center.

REFERENCES

ATITALLAH, Rabie Ben; ALI, Karim M. A. **FPGA-Centric High Performance Embedded Computing: Challenges and Trends.** *In*: 2017 Euromicro Conference on Digital System Design (DSD). [s.l.]: IEEE, 2017,

p. 390–395.

SEXTON, Conor; KAMINSKI, Nicholas J.; MARQUEZ-BARJA, Johann M.; 5G: **Adaptable Networks Enabled by Versatile Radio Access Technologies**. IEEE Communications Surveys & Tutorials, v. 19, n. 2, p. 688–720, 2017.

KOBAYASHI, R.; TAKAMAEDA-YAMAZAKI, S.; KISE, K. **Towards a Low-Power Accelerator of Many FPGAs for Stencil Computations**. *In*: 2012 Third International Conference on Networking and Computing. [s.l.]: IEEE, 2012, p. 343–349.

LIU, Hongwei; DING, Renwei; LIU, Lu; **Wavefield reconstruction methods for reverse time migration**. Journal of Geophysics and Engineering, v. 10, n. 1, 2013.

ZHOU, Hua-Wei; HU, Hao; ZOU, Zhihui; **Reverse time migration: A prospect of seismic imaging methodology**. Earth-science reviews, v. 179, p. 207–227, 2018.

CLAPP, Robert G. **Reverse time migration with random boundaries**. *In*: SEG Technical Program Expanded Abstracts 2009. Huston, USA: Society of Exploration Geophysicists, 2009, p. 2809–2813.

CLAERBOUT, Jon F. **Toward a unified theory of reflector mapping**. Geophysics, v. 36, n. 3, p. 467–481, 1971.

SANO, K.; HATSUDA, Y.; YAMAMOTO, S. **Multi-FPGA Accelerator for Scalable Stencil Computation with Constant Memory Bandwidt**h. IEEE Transactions on Parallel and Distributed Systems, v. 25, n. 3, p. 695–705, 2014.