

OpenACC와 GPU를 이용한 3차원 파동 전파 모델링

김아름 · 이종우 · 하완수*

부경대학교 에너지자원공학과

Three-dimensional Wave Propagation Modeling using OpenACC and GPU

Ahreum Kim, Jongwoo Lee, and Wansoo Ha*

Department of Energy Resources Engineering, Pukyong National University

요약: 3차원 주파수 영역과 라플라스 영역 파동장을 얻기 위해 시간 영역에서 파동 전파 모델링을 하는 동시에 푸리에 변환과 라플라스 변환을 수행하였다. 이 과정에서 효율적인 계산을 위해 OpenACC와 GPU를 이용한 병렬 연산을 수행하였다. OpenACC를 이용하면 기존의 C, C++, Fortran 등 프로그래밍 언어에 간단한 지시어(directive)를 추가하여 GPU 연산 가속기를 사용할 수 있기 때문에 CUDA 또는 OpenCL과 같은 GPGPU 프로그래밍 언어를 배우지 않고도 GPU를 이용한 프로그래밍을 할 수 있다. OpenACC 프로그램은 GPU 메모리 공간 할당, 호스트와 디바이스 간의 데이터 복사 및 GPU 연산 과정을 자동으로 또는 사용자 정의에 따라 수행하게 된다. 수치 실험으로 OpenACC와 GPU를 사용한 3차원 파동 전파 모델링 프로그램과 단일 CPU 코어를 사용한 프로그램의 성능을 비교하였다. 상속도 모델과 SEG/EAGE 압연 속도 모델을 이용한 결과, OpenACC와 GPU를 사용한 경우 단일 CPU 코어를 사용하였을 때보다 계산 속도가 각각 53배와 30배 정도 향상되었다.

주요어: OpenACC, 그래픽 프로세서, 3차원 파동전파모델링

Abstract: We calculated 3D frequency- and Laplace-domain wavefields using time-domain modeling and Fourier transform or Laplace transform. We adopted OpenACC and GPU for an efficient parallel calculation. The OpenACC makes it easy to use GPU accelerators by adding directives in conventional C, C++, and Fortran programming languages. Accordingly, one doesn't have to learn new GPGPU programming languages such as CUDA or OpenCL to use GPU. An OpenACC program allocates GPU memory, transfers data between the host CPU and GPU devices and performs GPU operations automatically or following user-defined directives. We compared performance of 3D wave propagation modeling programs using OpenACC and GPU to that using single-core CPU through numerical tests. Results using a homogeneous model and the SEG/EAGE salt model show that the OpenACC programs are approximately 53 and 30 times faster than those using single-core CPU.

Keywords: OpenACC, Graphics Processing Units, 3D wave propagation modeling

서론

석유·가스 자원 개발 분야에서는 탄성파 탐사를 통해 취득한 방대한 양의 자료에 3차원 및 4차원 처리 기술을 적용하여 고해상도의 지하 지질 구조를 규명함으로써 자원 개발 성공률

을 높이고 있다. 최근 3차원 탐사의 경우 기존 2차원 탐사에 비해 취득 자료가 훨씬 많기 때문에 자료 처리를 위한 계산 시간과 비용 측면에서 효율성을 높이고자 병렬 처리 기술에 대한 관심이 높아지고 있다. 자료의 병렬 처리를 위한 기술로는 다양한 방법이 있는데, 그 중 과거 영상 정보 처리만 담당하던 그래픽 처리장치(Graphic Processing Unit, GPU)를 이용하여 일반 연산을 가속하는 방법이 주목 받고 있다. GPU는 다수의 프로세서들을 이용하여 대용량 영상 데이터를 병렬로 처리하는데 최근에는 이러한 GPU의 병렬 연산 능력을 확장하여 항공 우주, 기상관측, 생명과학 등 여러 분야에 적용하고 있다 (Lee *et al.*, 2013).

해외에서는 GPU의 병렬 연산 기술을 석유·가스 자원 탐사 분야에서도 응용하여 2차원 파동 전파 모델링(Wang *et al.*, 2014), 크기가 큰 행렬 풀이 또는 많은 계산이 필요한 역시간

Received: 20 February 2017; Revised: 14 April 2017;

Accepted: 17 April 2017

*Corresponding author

E-mail: wansooaha@pknu.ac.kr

Address: Department of Energy Resources Engineering, Pukyong National University, 45 Yongso-Ro, Nam-Gu, Busan, Republic of Korea (48513)

©2017, Korean Society of Earth and Exploration Geophysicists

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

구조보정, 완전파형역산 등의 연구에 이용하고 있다 (Abdelkhalek *et al.*, 2009; Lu *et al.*, 2012; Mehra *et al.*, 2012). 국내에서도 GPU를 이용하여 3차원 라플라스 영역 완전 파형 역산 시 반복법을 이용한 행렬 풀이(Shin *et al.*, 2014), GPU를 이용한 이방성 탄성매질에서의 파 분리(Choi *et al.*, 2015), GPU를 이용한 이방성 매질에서의 역시간 구조보정의 가속화(Kim *et al.*, 2013) 등의 연구가 진행되었다.

본 연구에서는 GPU 병렬 프로그래밍에 쉽게 접근할 수 있도록 개발된 OpenACC (Open Accelerators) 기술을 이용하여 3차원 시간 영역 파동 전파 모델링을 수행하는 동시에 푸리에 변환 및 라플라스 변환을 통해 주파수 영역과 라플라스 영역 파동장을 생성하였다. 상속도 모델과 압연도 모델을 이용하는 수치 예제들을 통해 계산의 효율성과 정확도를 비교하였다.

OpenACC 기술

OpenACC는 가속기를 이용한 병렬 연산을 위해 Cray, CAPS, NVIDIA, PGI 사에서 개발한 기술로, CPU (Central Processing Unit)와 GPU를 같이 사용하는 이기종 시스템에 적용 가능하다. Fig. 1은 OpenACC의 개념을 간략하게 도식화한 것이다. CPU와 GPU에 각각 메모리를 할당하여 CPU의 데이터를 GPU로 전송하고 GPU에서 연산을 실행한다. 연산 결과는 다시 GPU에서 CPU로 전송하고 메모리 할당을 해제한다. CPU는 이러한 과정을 통제하는 역할을 담당한다. CUDA (Compute Unified Device Architecture)나 OpenCL (Open Computing Language)은 공유 메모리 사용, 데이터 복사 등의 저수준 제어가 가능하여 높은 성능을 얻을 수 있지만, 이를 사용하기 위해서는 새로운 프로그래밍 언어를 배워야 한다는 단점이 있다. 반면, OpenACC를 이용하면 프로그래밍 언어를 따로 학습하지 않고 기존의 프로그래밍 언어인 C, C++, Fortran 등으로 작성된 프로그램 코드에 지시어만 추가하여 쉽게 병렬화를 할 수 있다. 또한, 고수준의 플랫폼 독립 언어로 OpenACC를 사용하면 다양한 장비에서 상대적으로 쉽게 병렬 연산을 수행할 수 있다. 이러한 특징으로 개발 시간을 단축할 수 있고, 폭 넓은 활용이 가능하지만 CUDA와 OpenCL과 같

은 저수준 프로그래밍 모델에서만 가능한 최적화가 제한된다는 한계가 있다(OpenACC Programming and Best Practices Guide, 2015).

3차원 파동 전파 모델링

본 연구에서는 OpenACC를 이용하여 3차원 주파수 영역과 라플라스 영역 파동장을 생성하였다. 이 때 효율적인 연산을 위해 시간 영역 모델링을 수행하는 동시에 각각 푸리에 변환과 라플라스 변환을 통해 주파수 영역 및 라플라스 영역 파동장을 생성한다. 다음은 3차원 시간 영역 파동 방정식이다.

$$\begin{aligned} \frac{1}{v(x, y, z)^2} \frac{\partial^2 u(x, y, z; t)}{\partial t^2} \\ = \frac{\partial^2 u(x, y, z; t)}{\partial x^2} + \frac{\partial^2 u(x, y, z; t)}{\partial y^2} + \frac{\partial^2 u(x, y, z; t)}{\partial z^2} \\ + f(x, y, z; t) \end{aligned} \quad (1)$$

위 식 (1)에서 v 는 P 파 속도, u 는 3차원 시간 영역 파동장, f 는 송신원을 나타낸다. 시간 영역 모델링은 유한 차분법과 시간 전진 기법을 이용하여 수행할 수 있다.

일반적인 주파수 영역이나 라플라스 영역 모델링을 위해서는 행렬을 풀게 되는데 3차원의 경우 많은 계산량과 메모리가 필요하다. 이러한 문제를 해결하기 위해 시간 영역 양함수법에 푸리에 변환(식 (2))을 결합하여 주파수 영역 파동장을 생성하는 방법이 제안되었다(Furse, 2000; Nihei and Li, 2007; Sircue *et al.*, 2008).

$$\tilde{u}(\omega) = \int_{-\infty}^{\infty} u(t)e^{i\omega t} dt \quad (2)$$

위 식 (2)에서 ω 는 각주파수를 의미한다. 이 때 시간 영역 파동장을 저장한 후 푸리에 변환을 실행하는 것이 아니라 시간 전진의 매 시간 격자마다 위의 적분을 차례로 계산하여 메모리 사용량을 줄일 수 있다. 라플라스 영역 파동장 또한 식 (3)과 같이 라플라스 변환을 이용하여 주파수 영역에서와 같은 방식으로 생성할 수 있다(Ryu *et al.*, 2015).

$$\tilde{u}(s) = \int_{-\infty}^{\infty} u(t)e^{-st} dt \quad (3)$$

위 식 (3)에서 s 는 양의 감쇠 계수이다.

실험 구성

GPU와 OpenACC를 사용한 프로그램의 성능과 단일 CPU 코어를 사용한 프로그램의 성능을 비교하기 위해 포트란 프로그래밍 언어로 작성한 3차원 파동 전파 모델링 프로그램이 이

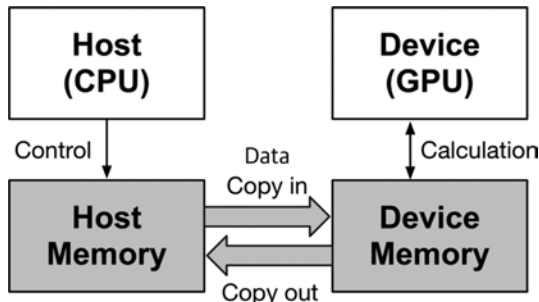


Fig. 1. OpenACC's abstract operation.

용하였다. 실험에 사용한 CPU는 Intel Xeon E5-2680 v3 모델 이고, GPU는 NVIDIA K40c를 사용하였다. 일반적으로 GPU는 단정도 부동소수점 연산 성능과 배정도 부동소수점 연산 성능 차이가 CPU의 경우보다 크다. 본 연구에 사용된 GPU의 단정도 연산 성능은 4.29 TFLOPS, 배정도 연산 성능은 1.43 TFLOPS이다. 이러한 특징으로 인해 단정도 연산을 이용하는 주파수 영역과 배정도 연산을 이용하는 라플라스 영역 파동 전파 모델링의 연산 성능이 달라지게 된다.

CPU와 GPU의 연산 성능과 정확도를 비교하기 위해 속도가 3 km/s인 3차원 반무한 균질 매질에서 모델 크기를 변화시켜 가며 모델링을 수행하였고, SEG/EAGE 3차원 암염돔 모델을 이용하여 파동장을 얻어 성능과 정확도를 비교하였다.

3차원 주파수 영역에서는 단정도 복소수의 부동 소수점 연산을 사용하였고, 라플라스 영역에서는 감쇠 계수가 클 경우 파동장의 크기가 단정도의 한계를 넘어가므로 배정도 실수 부동 소수점 연산을 사용하였다. CPU와 GPU용 실행 파일을 생성하기 위해 OpenACC를 지원하는 PGI사의 컴파일러를 사용하였고, 최적화를 위해서 -O2 옵션을 사용하였다. 모든 연산은 CPU와 GPU에서 세 번씩 반복한 후 평균 계산 시간으로 결과를 비교 분석하였다.

CPU를 이용한 3차원 주파수 영역과 라플라스 영역에서의 파동 전파 모델링 알고리즘은 Table 1에 제시하였다. 처음 변수를 초기화 한 후(Line 1), 시간에 따른 반복 루프를 시작한다(Line 2). 루프 안에서 공간에 대한 유한 차분식을 계산하고

Table 1. Three dimensional frequency-domain and Laplace-domain wave propagation modeling algorithm using CPU.

| |
|---|
| (1) Initialize |
| (2) For each time |
| (3) Solve finite-difference equation |
| (4) Inject source |
| (5) Boundary condition |
| (6) Time marching |
| (7) Fourier transform / Laplace transform |
| (8) End |
| (9) Write output |

Table 2. Three dimensional frequency and Laplace-domain wave propagation modeling algorithm using GPU.

| |
|--|
| (1) Initialize |
| (2) CPU to GPU : Copy velocity, source wavelet |
| (3) For each time |
| (4) Solve finite-difference equation |
| (5) Inject source |
| (6) Boundary condition |
| (7) Time marching |
| (8) Fourier transform / Laplace transform |
| (9) End |
| (10) GPU to CPU : Copy frequency/Laplace-domain wavefields |
| (11) Write output |

(Line 3), 송신원과(Line 4), 경계 조건을 추가한 후(Line 5), 시간 전진 기법을 적용 한다(Line 6). 시간 영역 파동장에는 푸리에 변환 또는 라플라스 변환(Line 7)을 적용하여 주파수 영역 또는 라플라스 영역 파동장을 생성한다. GPU 알고리즘에서는 OpenACC 지시어를 통해 CPU와 GPU 사이의 데이터 복사가 이루어진다(Table 2). 초기화 이후(Line 1), 병렬 계산을 위해 '\$!acc data copyin' 명령어를 통해 GPU 메모리를 할당하여 속도 모델을 복사한다(Line 2). 시간 반복문 내의 모든 연산은 GPU에서 수행한다(Line 3 ~ Line 9). 결과로 생성된 주파수 영역과 라플라스 영역 파동장은 '\$!acc data copyout' 명령어를 이용하여 GPU 연산 종료와 함께 디바이스(GPU)에서 호스트(CPU)로 전송한다(Line 10). CPU와 GPU 사이의 데이터 복사는 모델링 시작과 끝 부분에서만 발생하기 때문에 위의 알고리즘을 이용하면 효율적인 GPU 연산이 가능하다(Kim *et al.*, 2016).

모델 크기에 따른 실행 시간 비교

앞서 실험 구성에서 설명한 바와 같이 주파수 영역과 라플라스 영역 파동 전파 모델링 알고리즘을 이용하여 3차원 반무한 균질 매질에서 모델 크기를 바꾸어가며 모델링을 실행하였다. 매질에서의 P파 속도는 3 km/s, 주파수 영역 파동장의 주파수는 20 Hz, 라플라스 영역에서의 감쇠 계수는 10 s^{-1} , 시간 샘플링 간격은 1 ms로 10초간 모델링을 수행하였다.

주파수 영역에서는 계산 시간 비교를 위해 모델 크기를 $200 \times 200 \times 200$ 부터 각 방향으로 100씩 증가시켜가며 $700 \times 700 \times 700$ 까지 계산하였고, 라플라스 영역에서는 연구에 사용하는 장비의 메모리 용량을 고려하여 $600 \times 600 \times 600$ 까지 계산하였다.

Fig. 2와 Fig. 3은 3차원 주파수 영역과 라플라스 영역 파동 전파 모델링을 모델 크기에 따라 단일 CPU 코어와 OpenACC (GPU)를 이용하여 각각 세 번의 반복 계산 후 얻은 평균 연산 시간이다. 그래프에서 실선으로 나타낸 속도 향상 정도는 단일

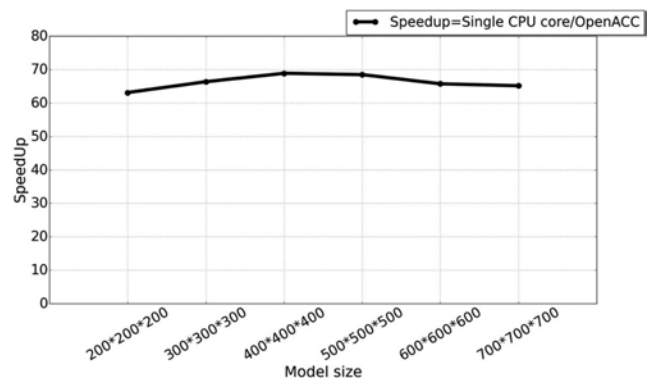


Fig. 2. Speedups of calculation times depending on the model size in the frequency domain using single CPU core and a GPU card.

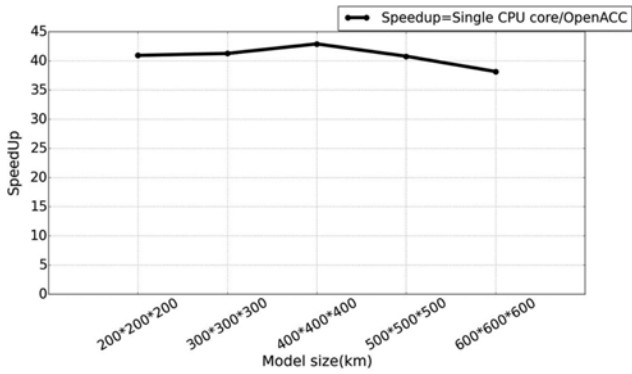


Fig. 3. Speedups of calculation times depending on the model size in the Laplace domain using single CPU core and a GPU card.

CPU 코어에서의 연산 시간을 GPU 연산 시간으로 나눈 값이다. GPU 연산 시간에는 CPU와 GPU 사이의 데이터 복사 시간도 포함된다.

3차원 주파수 영역 파동 전파 모델링을 수행하였을 때의 계산 속도 향상 결과인 Fig. 2로부터 OpenACC와 GPU를 사용한 경우 단일 CPU 코어를 사용하여 계산 했을 때 보다 60배 이상 계산 속도가 향상 됨을 알 수 있었다. Fig. 3을 보면 OpenACC와 GPU를 이용하여 3차원 라플라스 영역 파동 전파 모델링을 수행할 경우 계산 속도가 단일 코어 CPU 대비 최대 40배 이상 향상 됨을 확인할 수 있다.

암염돔 모델 적용 결과 비교

3차원 반무한 균질 매질에서 모델 크기에 따라 성능을 비교한 후, 실제 지형을 묘사한 SEG/EAGE 암염돔 속도 모델을 이용하여 계산 성능과 정확도를 비교하였다(Fig. 4). 모델링에 적용된 SEG/EAGE 암염돔 모델의 크기는 $676 \times 676 \times 201$, 격자 크기는 20 m로, 송신원은 모델 지표면 가운데 격자에 두고 최대 20 Hz 주파수의 송신원을 사용하여 1 ms 간격으로 총 10초간 모델링을 수행하였다.

주파수 영역과 라플라스 영역 파동 전파 모델링에서 CPU와 GPU 계산 시간을 비교한 결과를 Table 3에 제시하였다. 3차원 주파수 영역에서 모델링 결과, CPU에서의 실행 시간은 7시간 3분 2초, GPU에서의 실행 시간은 7분 55초로 계산 시간

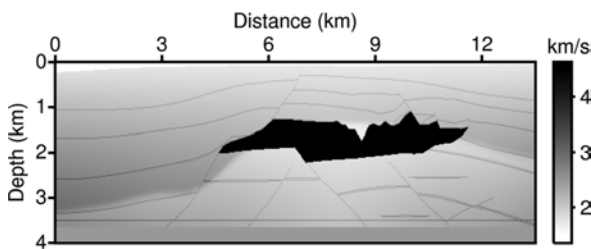


Fig. 4. A section of the SEG/EAGE 3D salt velocity model.

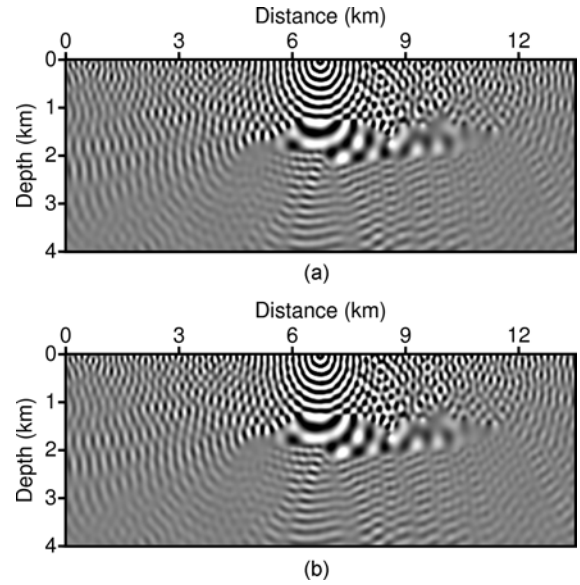


Fig. 5. Real parts of XZ sections of 3D frequency-domain wavefields obtained using (a) CPU and (b) GPU at the frequency of 10 Hz.

Table 3. Calculation times of CPU and GPU programs in the frequency and Laplace domains.

| Calculation time | Frequency domain (s) | Laplace domain (s) |
|------------------|----------------------|--------------------|
| CPU | 25,382 | 22,575 |
| GPU | 475 | 756 |

이 약 53배 향상됨을 알 수 있었다. Fig. 5는 CPU와 GPU에서 얻은 주파수 영역 파동장의 XZ 단면이다. 육안으로 거의 일치함을 볼 수 있으나, 더욱 정확한 비교를 위해 뒤에서 트레이이스를 추출하여 정확도를 분석하였다.

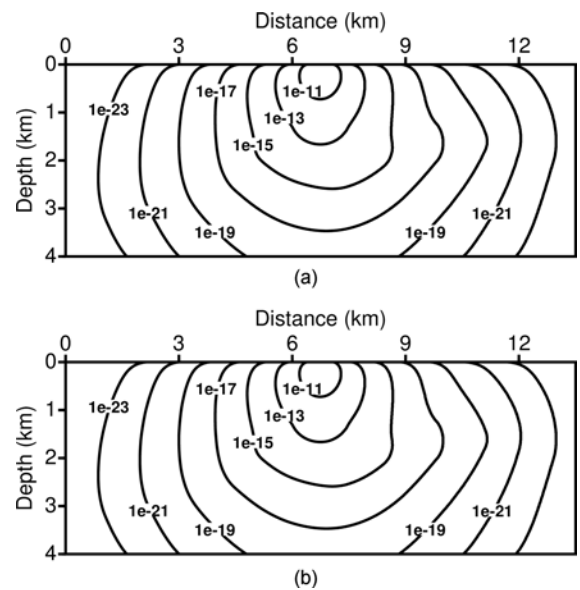


Fig. 6. XZ sections of 3D Laplace-domain wavefields obtained using (a) CPU and (b) GPU at the damping constant of 10 s^{-1} .

3차원 라플라스 영역에서의 계산 시간은 CPU에서 6시간 16분 15초가 걸린 반면, GPU에서는 12분 36초가 걸려 CPU 대비 약 30배 정도 계산 시간이 향상 되었음을 알 수 있었다. Fig. 6은 CPU와 GPU에서 얻은 라플라스 영역 파동장의 XZ 단면이다.

정확도 비교

CPU에서 얻은 결과와 GPU에서 얻은 결과의 정확도를 비교하였다. 순차적으로 계산하게 되는 CPU 프로그램과 달리 GPU에서는 병렬 연산을 이용하기 때문에 실수 및 복소수의 계산 순서가 달라지게 된다. 컴퓨터에서 실수 및 복소수 연산은 부동소수점을 사용하는데, 부동 소수점 연산은 엄밀하게 말해 교환법칙이 성립하지 않기 때문에 CPU와 GPU의 계산 결과가 달라질 수 있다. 따라서 비록 그 차이가 크지 않더라도 CPU와 GPU에서 얻은 결과가 동일한지 비교해 볼 필요가 있다. SEG/EAGE 3차원 암염동 속도 모델을 이용해 얻은 주파수 영역과 라플라스 영역 파동장의 정확성을 비교하기 위해 송신원에서 4 km 떨어진 지점에서 트레이스를 추출하여 비교

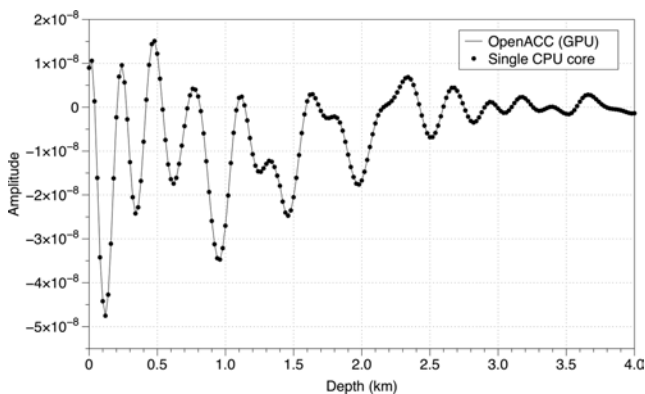


Fig. 7. Traces extracted from the XZ section (Fig. 5), at a distance of 4 km from the source.

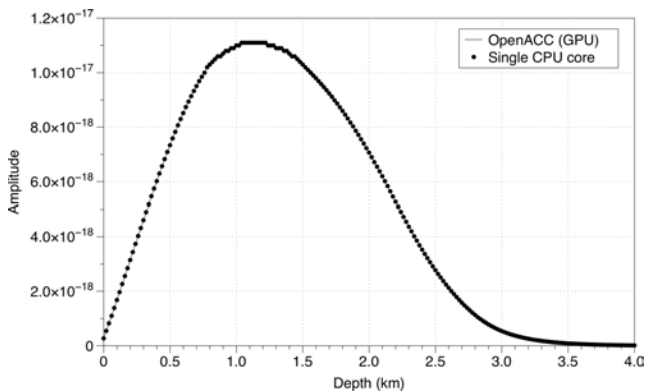


Fig. 8. Traces extracted from the XZ section (Fig. 6), at a distance of 4 km from the source.

하였다(Fig. 7, 8). 그 결과 두 영역에서 모두 오차율이 0.01% 미만으로 결과가 거의 일치함을 알 수 있었다.

토의 및 결론

본 연구에서는 OpenACC와 GPU를 이용한 3차원 주파수 영역과 라플라스 영역 파동 전파 모델링을 수행하고 성능을 비교하였다. 3차원의 경우 양함수법을 이용하는 시간 영역 파동 전파 모델링과 달리 행렬 풀이를 이용하는 주파수 영역과 라플라스 영역에서의 파동 전파 모델링을 위해서는 많은 계산량과 메모리가 필요하다. 이 문제를 해결하기 위해 시간 영역 모델링에 푸리에 변환과 라플라스 변환을 결합하여 주파수 영역과 라플라스 영역 파동장을 빠르게 계산하는 방법을 사용하였다.

CPU와 GPU의 계산 성능 비교를 위해 3차원 반무한 균질 매질과 3차원 SEG/EAGE 암염동 속도 모델을 이용하였다. 그 결과, 모델 크기가 증가함에 따라 CPU와 GPU에서 각각 계산 시간은 증가하였지만, 단일 CPU 코어를 사용하는 경우에 비해 GPU를 이용하면 계산 속도가 주파수 영역에서는 약 60배 이상, 라플라스 영역에서는 약 37 ~ 40배 정도 향상됨을 알 수 있었다. 또한, SEG/EAGE 암염동 속도 모델을 이용하여 모델링을 수행하였을 때에도 각각 53배와 30배 정도의 속도 향상이 있었다. GPU에서의 연산 결과는 CPU와 GPU 사이의 데이터 복사를 포함한 연산 시간으로 전체 실행 시간에서 데이터 복사가 수행되는 시간의 비율은 크지 않았다.

본 연구에서는 상대적으로 주파수 영역에서의 계산보다 라플라스 영역에서의 계산 효율성이 낮음을 알 수 있었다. 일반적으로 GPU에는 단정도 부동 소수점 연산을 위한 프로세싱 유닛을 배정도 부동소수점 연산을 위한 프로세싱 유닛보다 많이 가지고 있어 단정도 연산이 더 빠르게 수행된다. 본 연구에서는 단정도 연산을 이용하여 주파수 영역, 배정도 연산을 이용하여 라플라스 영역 연산을 하였다. 따라서, Table 3에서와 같이 GPU를 이용한 모델링은 주파수 영역 연산이 라플라스 영역 연산보다 계산 효율성이 더 높다는 것을 알 수 있다. 반면, CPU를 사용하는 경우에는 복소수를 사용하는 주파수 영역 연산이 배정도 실수를 사용하는 라플라스 영역 연산보다 오래 걸렸는데, 이는 상대적으로 CPU의 배정도 연산 능력이 우수하기 때문이다.

3차원 주파수 영역과 라플라스 영역 모델링 프로그램의 단순하게 작성한 CPU 코드 줄 수는 핵심 부분이 150여 줄 정도였다. 여기에 10줄 이내의 OpenACC 지시어를 추가함으로써 GPU 병렬 연산에 쉽게 접근 가능함을 알 수 있었다.

추후 석유·가스 자원 탐사 분야의 자료 처리 외에도 대량의 연산을 필요로 하는 여러 분야에서 OpenACC를 통해 GPU 병렬 프로그래밍에 쉽게 접근하여 효율적인 결과를 얻을 수 있을 것이라 예상된다.

감사의 글

본 연구는 한국지질자원연구원 주요사업인 ‘유가스 부존특성 및 해저천부구조 정밀 규명을 위한 융복합 탐사기술 개발’, 부산과학기술기획평가원의 신진연구자 지원사업 및 산업통상자원부 자원개발특성화대학사업의 지원을 받아 수행되었습니다.

References

- Abdelkhalek, R., Calandra, H., Coulaud, O., Roman, J., and Latu, G., 2009, Fast seismic modeling and reverse time migration on a GPU cluster, *International Conference on High Performance Computing & Simulation*, 36-43.
- Choi, H., Seol, S., and Byun, J., 2015, Acceleration of anisotropic elastic reverse-time migration with GPUs, *Geophys. and Geophys. Explor.*, **18**, 74-84 (in Korean with English abstract).
- Furse, C. M., 2000, Faster than Fourier: ultra-efficient time-to-frequency-domain conversions for FDTD simulations, *IEEE Antennas and Propagation*, **42**, 24-34.
- Kim, Y., Cho, Y., Jang, U., and Shin, C., 2013, Acceleration of stable TTI P-wave reverse-time migration with GPUs, *Computers & Geosciences*, **52**, 204-217.
- Kim, A., Ryu, D., and Ha, W., 2016, Time-domain 3D wave propagation modeling and memory management using graphics processing units, *Geophys. and Geophys. Explor.*, **19**, 145-152 (in Korean with English abstract).
- Lee, H., Jeong, Y., and Lee, S., 2013, Survey on GPGPU programming models, *Korea Information Processing Society*, **20**, 389-391.
- Lu, L., Renwei, D., Hongwei, L., and Hong, L., 2015, 3D hybrid-domain full waveform inversion on GPU, *Computers & Geosciences*, **83**, 27-36.
- Mehra, R., Raghuvanshi, N., Savioja, L., Lin, M. C., and Manocha, D., 2012, An efficient GPU-based time domain solver for the acoustic wave equation, *Applied Acoustics*, **73**, 83-94.
- Nihei, K. T., and Li, X., 2007, Frequency response modelling of seismic waves using finite difference time domain with phase sensitive detection (TD-PSD), *Geophysical Journal International*, **169**, 1069-1078.
- OpenACC Programming and Best Practices Guide, 2015, <http://www.openacc.org/content/openacc-programming-best-practices-guide> (April 14, 2017 Accessed)
- Ryu, D., Kim, A., and Ha, W., 2015, Expanding domain method for 3D time-Laplace-domain hybrid modeling, *Geosystem Engineering*, **18**, 259-265.
- Shin, J., Ha, W., Jun, H., Min, D., and Shin, C., 2014, 3D Laplace-domain full waveform inversion using a single GPU card, *Computers & Geosciences*, **67**, 1-13.
- Sirgue, L., Etgen, J., and Albertin, U., 2008, 3D frequency domain waveform inversion using time domain finite difference methods, *70th EAGE Conference and Exhibition Incorporating SPE EUROPEC*.
- Wang, W., Xu, L., Cavazos, J., Huang, H. H., and Kay, M., 2014, Fast acceleration of 2D wave propagation simulations using modern computational accelerators, *PLOS ONE*, **9**, e86484.