

## 제온 파이 x200 프로세서를 이용한 3차원 음향 파동 전파 모델링 병렬 연산 성능 비교

이종우 · 하완수\*

부경대학교 에너지자원공학과

### Comparison of Parallel Computation Performances for 3D Wave Propagation Modeling using a Xeon Phi x200 Processor

Jongwoo Lee and Wansoo Ha\*

Department of Energy Resources Engineering, Pukyong National University

**요약:** 본 연구에서는 제온 파이 x200 프로세서를 이용하여 3차원 파동 전파 모델링을 수행하고 기존의 제온 CPU를 사용한 경우와 병렬 연산 성능을 비교하였다. 제온 파이 1세대 프로세서인 제온 파이 나이즈 코너 보조프로세서와 달리 제온 파이 2세대 프로세서인 x200 프로세서는 직접 운영체제 실행이 가능하므로 내장 메모리와 주메모리 사이의 추가적인 통신이 필요 없다. 또한 제온 파이 x200 프로세서는 대용량 주메모리와 고대역폭 메모리를 이용하여 대규모 컴퓨팅을 독립적으로 실행할 수 있다. 병렬 연산 성능 비교를 위해 MPI (Message Passing Interface)와 OpenMP (Open Multi-Processing)를 이용해 모델링을 수행하였다. SEG/EAGE 암염동 모델을 이용한 수치 실험 결과 제온 파이에서 다량의 연산 코어와 고대역폭 메모리를 이용해 12 코어 CPU 대비 2.69 ~ 3.24배 우수한 모델링 성능을 얻을 수 있었다.

**주요어:** 3차원, 파동 전파 모델링, 제온 파이, OpenMP, MPI

**Abstract:** In this study, we simulated 3D wave propagation modeling using a Xeon Phi x200 processor and compared the parallel computation performance with that using a Xeon CPU. Unlike the 1st generation Xeon Phi coprocessor codenamed Knights Corner, the 2nd generation x200 Xeon Phi processor requires no additional communication between the internal memory and the main memory since it can run an operating system directly. The Xeon Phi x200 processor can run large-scale computation independently, with the large main memory and the high-bandwidth memory. For comparison of parallel computation, we performed the modeling using the MPI (Message Passing Interface) and OpenMP (Open Multi-Processing) libraries. Numerical examples using the SEG/EAGE salt model demonstrated that we can achieve 2.69 to 3.24 times faster modeling performance using the Xeon Phi with a large number of computational cores and high-bandwidth memory compared to that using the 12-core CPU.

**Keywords:** three-dimensional, wave propagation modeling, Xeon Phi, OpenMP, MPI

## 서 론

최근 석유·가스 자원 탐사 분야에서는 고해상도의 지하 지질 구조 영상을 얻기 위해 3차원 탄성과 탐사가 활발히 수행되고

있다. 3차원 탐사를 통해 얻어지는 탐사 자료는 자료 처리시 기존의 2차원 자료에 비해 훨씬 많은 시간과 비용이 필요하다. 이 때, 3차원 자료 처리의 시간 및 비용 측면의 효율성을 높이 고자 많은 연구에서 병렬 처리 기술을 이용하고 있다. 대량 연산 수행 시, 기존 중앙처리장치(CPU, Central Processing Unit)의 부족한 코어 수에 의한 성능의 한계를 극복하고자 개별 코어의 성능은 떨어지지만 대량의 코어를 이용해 병렬 연산을 수행하는 연산 가속기를 이용하는 사례가 많아지고 있다(Ryu *et al.*, 2017). 대표적인 연산 가속기로는 범용 그래픽 처리 장치(General Purpose Graphics Processing Unit, GPGPU)와 매니코어(Many Integrated Core, MIC) 구조의 제온 파이 가속기가 있다. 이러한 병렬 연산 가속기는 대규모 병렬화가 가능한 연산에 적합하며, 단일 코어의 성능은 CPU에 비해 떨어지기

Received: 31 July 2018; Revised: 6 September 2018;

Accepted: 11 September 2018

\*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)

©2018, 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.

때문에 하나의 코어만 사용하는 작업에는 적절하지 않다. GPU를 이용하면 수백 또는 수천 개에 이르는 그래픽 프로세서 코어를 이용해 3차원 구조보정이나 완전 파형 역산과 같은 대규모 병렬 연산을 효율적으로 수행할 수 있다(Abdelkhalek *et al.*, 2009; Lu *et al.*, 2015). 그러나 GPU를 이용한 병렬화를 위해서는 CUDA, OpenCL과 같은 별도의 병렬 프로그래밍 모델을 적용시키기 위한 노력이 필요할 뿐만 아니라 제한적인 GPU 메모리, 주메모리와 GPU 메모리간 데이터 전송 등으로 인해 성능 향상 정도에도 제약이 있다(Kim *et al.*, 2016). 제온 파이의 경우 CPU와 유사한 코어를 다수 장착하였기 때문에 CPU에서 사용하던 프로그램을 거의 그대로 사용하여 성능 향상을 얻을 수 있다는 장점이 있다. 그러나 기존의 제온 파이프 보조 프로세서 또한 별도의 주 프로세서와 연결된 경우에만 사용이 가능했기 때문에 데이터 이동이 필요했고 성능 향상 정도가 제한적이었다(Ryu *et al.*, 2017).

2016년 인텔사에서 출시한 제온 파이 x200 프로세서는 고성능 병렬 연산을 위해 매니코어 구조로 설계된 독립 실행형 프로세서이다. 별도의 CPU 없이 직접 운영체제 실행이 가능하며 기존의 보조 프로세서와 달리 주메모리와 프로세서의 메모리 사이에 추가적인 통신이 필요 없다. 제온 파이 x200 프로세서는 대용량의 DDR (Double Data Rate) 메모리와 고대역폭의 MCDRAM (Multi-Channel Dynamic Random-Access Memory)을 탑재하여 대규모 컴퓨팅을 독립적으로 실행할 수 있다. 최근 탄성파 자료 처리시 제온 파이 x200 프로세서를 활용하는 사례가 많아지고 있다(Heinecke *et al.*, 2016; Rodriguez *et al.*, 2017; Sourouri and Birger Raknes, 2017; Tobin *et al.*, 2017). 본 연구에서는 제온 파이 x200 프로세서를 이용하여 고해상도 자료 처리의 기초가 되는 기술인 3차원 파동 전파 모델링을 수행하고 기존의 서버용 제온 CPU를 사용한 경우와 병렬 연산 성능을 비교하였다.

### 제온 파이 x200 프로세서

제온 파이는 인텔사에서 고성능 병렬 연산을 위해 개발한 MIC 구조의 프로세서이다. 제온 파이 1세대 프로세서인 나이즈 코너 모델은 별도의 호스트 프로세서에 연결되어 있는 경우에만 사용이 가능한 보조 프로세서였다(Ryu *et al.*, 2017). 반면에 2016년 출시된 제온 파이 2세대 프로세서인 x200 나이즈 랜딩 프로세서는 직접 리눅스 운영체제로 부팅할 수 있고 네트워크에 직접 연결이 가능한 독립형 프로세서이다. 따라서 제온 파이 x200 프로세서에서는 계산을 위해 별도의 데이터 이동 없이 연산하는 동안 데이터를 주메모리 또는 내장 고대역폭 메모리에 유지할 수 있다.

제온 파이 x200 모델의 내부는 최대 36개의 타일로 구성되어 있으며 각 타일은 두 개의 코어와 코어당 두 개의 VPU (Vector-Processing Unit) 그리고 두 코어 간에 공유된 1 MB의

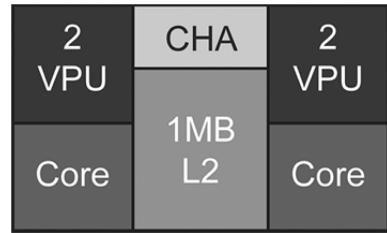


Fig. 1. Block diagram of a tile (Jeffers *et al.*, 2016, used with permission).

L2 캐시로 구성된다(Fig. 1). 각 VPU는 512 비트 폭으로 8개의 배정도 연산 또는 16개의 단정도 연산을 동시에 수행할 수 있다. 제온 파이 x200 모델은 하이퍼스레딩을 지원하여 코어당 최대 4개의 스레드 연산을 동시에 수행할 수 있다. 각 타일들은 캐시 일관성 있는 2차원 메시로 상호 연결된다(Fig. 2). 이를 통해 다른 타일의 L2 캐시와 메모리에 저장된 데이터에 접근하거나 입출력 할 수 있다. 제온 파이 x200 프로세서의 메시 상호 연결 구조는 나이즈 코너 모델의 1차원 링 상호 연결 구조에 비해 더 높은 대역폭과 낮은 지연 속도를 제공함으로써 보다 확장 가능한 타일 연결 방식을 제공한다(Jeffers *et al.*, 2016).

제온 파이 x200 프로세서를 사용할 경우 DDR과 MCDRAM 두 가지 유형의 메모리를 사용할 수 있다. 이 두 유형의 메모리는 대규모 컴퓨팅 기능과 독립 실행형 프로세서 상태를 지원하도록 설계되었다. DDR 메모리는 일반 서버의 주메모리에 해당하는 것으로, 제온 파이 x200 프로세서 패키지 외부의 대용량 메모리를 제공한다. 칩의 양쪽 편에 각각 3개의 채널을

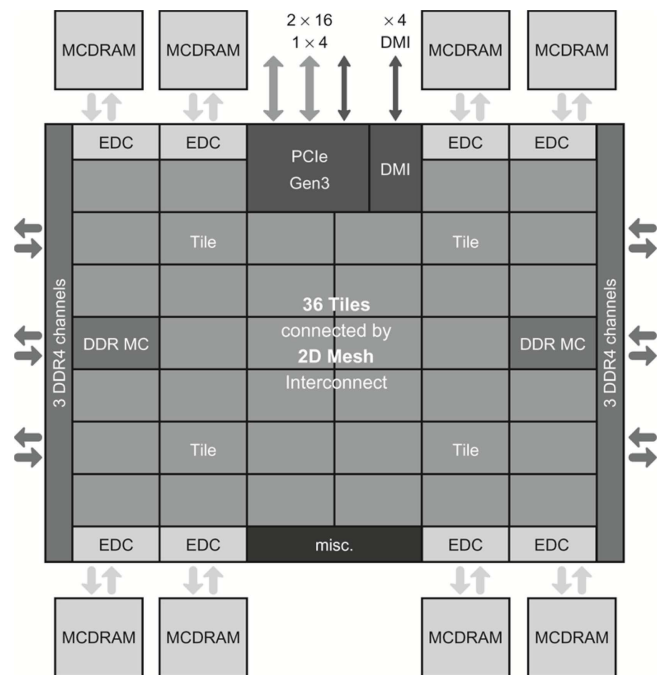


Fig. 2. Block diagram showing overview of Xeon Phi x200 Architecture (Jeffers *et al.*, 2016, used with permission).

제어하는 2개의 DDR4 메모리 컨트롤러가 있다. 각 채널당 최대 64 GB 용량을 지원하며 최대 메모리 용량은 384 GB이다. DDR4 메모리는 제온 파이 x200 프로세서의 독립 실행이 가능하도록 대용량 메모리를 지원하지만 최대 대역폭은 약 90 GB/s 정도로, 대규모 컴퓨팅에 필요한 고대역폭 데이터 전송 기능은 없다. 이를 위해 제온 파이 x200 프로세서 패키지에는 고대역폭의 MCDRAM이 통합되어 있다. 총 8개의 MCDRAM 장치가 통합되어 있으며 각 장치의 용량은 2 GB로 총 용량은 16 GB이다. 최대 대역폭은 450 GB/s 이상이다(Jeffers *et al.*, 2016).

### 3차원 시간 영역 음향파 파동 전파 모델링

본 연구에서는 3차원 시간 영역 음향파 파동 전파 모델링을 이용해 제온 파이와 서버 용 제온 CPU의 성능을 비교하였다.

$$\frac{1}{v(\mathbf{x})^2} \frac{\partial^2 u(\mathbf{x}, t)}{\partial t^2} = \nabla^2 u(\mathbf{x}, t) + f(\mathbf{x}, t). \quad (1)$$

위 식에서  $u$ 는 압력 파동장,  $v$ 는 P파 속도,  $f$ 는 송신원이다. 위 식에 다음과 같이 유한 차분식 양함수법을 적용하여 파동 전파 모델링을 수행하였다(Min *et al.*, 2016).

$$u(\mathbf{x}, t + \Delta t) = v(\mathbf{x})^2 \Delta t^2 [L\{u(\mathbf{x}, t)\} + f(\mathbf{x}, t)] + 2u(\mathbf{x}, t) - u(\mathbf{x}, t - \Delta t). \quad (2)$$

위 식에서  $\Delta t$ 는 샘플링 간격,  $L$ 은 유한 차분 라플라시안 연산자로, 수식은 유한 차분식 차수에 따라 달라진다(Jo and Ha, 2018). 위 식을 이용한 모델링 알고리즘은 Fig. 3과 같다.

### 실험 설계

제온 파이와 CPU의 성능 비교를 위해 병렬 3차원 파동 전파 모델링 프로그램을 이용하였다. 프로그램은 MPI (Message Passing Interface)와 OpenMP (Open Multi-Processing)를 이용해 병렬 연산을 수행하도록 만들었다. MPI는 분산 메모리 병렬 연산 기술이므로 데이터 공유가 거의 필요 없는 송신원별 병렬화에 사용하였고, OpenMP는 공유 메모리 병렬 연산 기술이므로 파동 방정식의 유한 차분식 및 경계조건 연산에 사용하였다(Fig. 3). 이에 따라 각각의 MPI 프로세스들은 서로 다른 송신원을 이용해 파동 전파를 모델링하는 태스크 병렬화 (Task parallelization)에 사용하고, 각각의 OpenMP 스레드들은 하나의 송신원 모델링에서 서로 다른 격자점의 파동장을 계산하는 데이터 병렬화(Data parallelization)에 사용한다(Fig. 4). 모든 실험에서 컴파일러 옵션을 이용해 CPU에서는 AVX2 (Advanced Vector Extensions 2), 제온 파이에서는 AVX512 벡터 연산을 기본적으로 사용하였다.

실험에 사용한 CPU는 Intel Xeon E5-2680 v3 2.5 GHz, 12

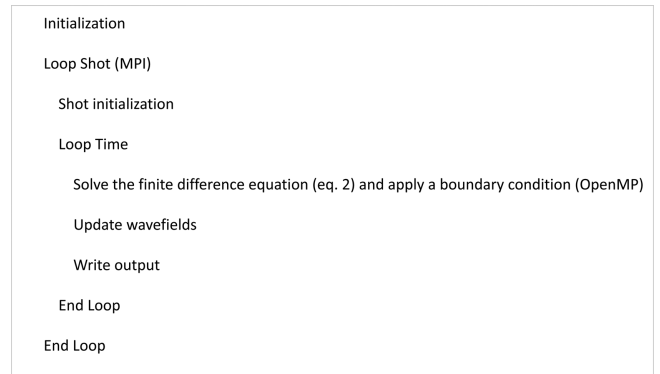


Fig. 3. A time-domain modeling algorithm.

코어 모델이고 CPU 서버에는 DDR4 16 GB 2133 MHz 메모리가 8장 설치되어 있다. 제온 파이는 Intel Xeon Phi 7210 1.3 GHz, 64 코어 모델이고, 제온 파이 서버에는 고대역폭 메모리로 MCDRAM 2 GB 6400 MHz 메모리 8장과 대용량 주메모리로 DDR4 16 GB 2133 MHz 메모리 6장이 설치되어 있다.

수치 실험에서는 SEG/EAGE 3차원 암염돔 모델을 이용하여 12개 송신원 모델링을 수행하고 연산 시간을 비교하였다. 모델링 결과 탄성과 기록은 정밀도 범위 내에서 동일하기 때문에 따로 제시하지 않았다. 속도 모델의 크기는  $676 \times 676 \times 201$ 이고, 격자 크기는 20 m이다. 공간에 대해 2차와 8차 유한 차분식을 사용하였고, 시간에 대해 2차 유한 차분식을 사용하였다. 2 ms 간격으로 총 6초간 모델링을 수행하였다. 공간에 대한 2차와 8차 유한 차분식의 경우 각각 단정도와 배정도 연산 성능을 비교하여 병렬 연산 종류별로 총 네 가지의 경우를 모두 비교하였다.

공간에 대한 2차 유한 차분식을 사용할 경우 연산을 위해 메모리에서 현재 계산 격자점과 주변 6개의 격자점 정보를 읽어오는 반면, 8차 유한 차분식을 사용할 경우 현재 계산 격자점과 주변 24개의 격자점 정보를 읽어오게 된다. 부동소수점 연산 횟수는 8차 유한 차분식이 많지만 CPU 연산 속도는 메모리 속도보다 월등히 빠르기 때문에 8차 유한 차분식의 경우

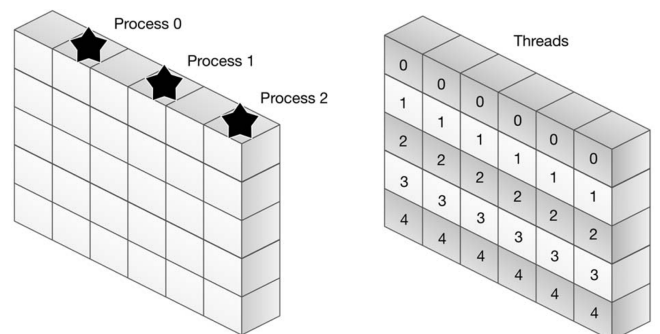


Fig. 4. Parallelization using MPI processes (left) and OpenMP threads (right). Each number of star shows the rank of a process who performs a shot simulation. Each number on a grid shows the ID of a thread who calculates the wavefield on each grid block.

2차 유한 차분식에 비해 메모리 성능이 실행 시간에 크게 영향을 미치게 된다. 배정도 연산의 경우도 단정도 연산에 비해 두 배의 메모리 정보를 읽고 쓰게 되므로 단정도에 비해 상대적으로 메모리 성능에 크게 영향을 받게 된다.

### OpenMP 병렬화 성능 분석

우선 CPU와 제온 파이에서 OpenMP를 이용한 병렬 연산 성능을 비교하며 제온 파이의 하이퍼스레딩 및 고대역폭 메모리 성능 또한 분석하였다. OpenMP를 이용하면 병렬 연산이 필요 없는 부분에서 하나의 스레드만 이용해 연산을 수행하다가 병렬 연산 지시자가 시작되는 부분에서 여러 개의 스레드를 생성하고 병렬로 연산을 수행한 후 병렬 연산 지시자가 끝나는 위치에서 다시 하나의 스레드만 남기게 된다. 이 때 모든 스레드는 메모리 공간을 공유하기 때문에 OpenMP를 공유 메모리 병렬 연산이라 한다(Pacheco, 2011).

본 연구에 사용한 CPU는 12코어 CPU로, 병렬화를 위해 1개부터 12개까지의 코어를 사용하여 연산을 수행하였다. 제온 파이의 경우 하이퍼스레딩 기술을 통해 코어당 4개의 스레드까지 할당할 수 있기 때문에 64코어 장비임에도 불구하고 최대 256 스레드까지 비교하였다. 최적의 코어당 스레드 수는 연산의 종류에 따라 달라지게 된다(Jeffers *et al.*, 2016). 소량의 데이터를 이용해 대량의 연산을 수행하는 경우 더 많은 스레드를 사용하면 연산 속도가 빨라질 수 있다. 그러나 격자를 이용한 파동 방정식 모델링의 경우 연산을 위해 메모리에서 많은 데이터를 읽어 들여야 하기 때문에 스레드가 많아지면 메모리 접근에서 병목현상이 발생하여 속도가 느려질 수도 있다. 개별 코어 속도가 느린 제온 파이에서 병렬 연산을 사용하지 않고 단일 코어를 이용해 연산하는 것은 실용적이지 않기 때문에 제온 파이에서는 8개부터 256개까지의 스레드를 사용하였다.

연산 결과는 Table 1에 제시하였다. OpenMP 스레드 수 증가에 따른 성능 변화를 보면, 제온 파이와 CPU에서 모두 스레드 수가 증가할수록 선형 속도 증가에서 멀어지는 양상을 볼 수 있다. 이는 메모리에서 데이터를 읽어오는 속도가 CPU의 연산 속도를 따라가지 못하기 때문에 발생하는 현상으로, 스레드 수가 증가할수록 데이터를 기다리느라 대기하는 스레드가 늘어나 연산 속도 향상 정도가 작아지게 된다. 네 가지 실험에서 제온 파이는 64코어를 사용한 경우 CPU 1 코어에 비해 14.88 ~ 23.14배, CPU 12 코어에 비해 5.25 ~ 8.43배 빨랐다(Fig. 5).

제온 파이의 하이퍼스레딩과 관련하여, 코어당 1개 또는 2개의 스레드를 사용하는 경우의 성능이 가장 좋았으나 그 차이는 크지 않았다. 코어당 4개의 스레드를 사용하면 연산 속도가 오히려 느려졌는데, 이는 메모리에 동시에 접근하는 스레드 수가 늘어서 병목 현상이 발생하였기 때문이다. 이러한 병목 현상은 제한된 메모리 접근 속도로 인해 발생하는 현상으로, 100 × 100 × 100 크기의 작은 모델에서도 동일하게 발생하였다. 따라서 이후의 실험과 분석에서는 코어당 1개의 스레

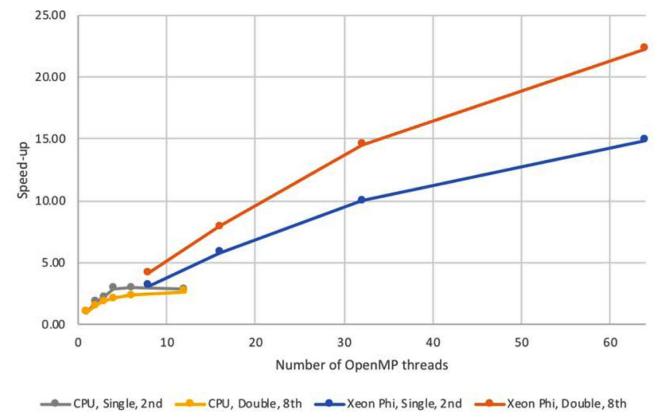


Fig. 5. Speed-ups using OpenMP with respect to the calculation times using one CPU core.

Table 1. Comparison of calculation times depending on the number of OpenMP threads, precision and order of FDM (s).

Number of OpenMP threads	Single precision		Double precision		
	2 <sup>nd</sup> order FDM (s)	8 <sup>th</sup> order FDM (s)	2 <sup>nd</sup> order FDM (s)	8 <sup>th</sup> order FDM (s)	
Xeon Phi	8	2286.5	4004.2	3367.4	6549.3
	16	1248.0	2125.5	1820.4	3436.0
	32	730.8	1176.0	1061.0	1877.3
	64	488.8	751.6	735.1	1226.4
	128	472.6	743.2	730.9	1269.3
	256	517.2	847.2	837.3	1785.4
CPU	1	7272.7	17393.5	12193.3	27261.4
	2	3930.4	9012.1	7591.2	17869.3
	3	3280.2	8070.6	7129.9	14704.5
	4	2523.3	6301.4	6275.1	12673.2
	6	2435.3	5656.8	5961.1	11380.5
	12	2566.3	4927.4	5511.1	10333.9

**Table 2.** Calculation times without high bandwidth memory on the Xeon Phi processor.

Number of OpenMP threads	Single precision		Double precision	
	2 <sup>nd</sup> order FDM (s)	8 <sup>th</sup> order FDM (s)	2 <sup>nd</sup> order FDM (s)	FDM (s)
16	1565.5	2628.5	2655.5	4783.1
64	1520.5	2500.2	2605.3	4835.8

**Table 3.** Calculation times using both MPI and OpenMP.

	Number of OpenMP threads	Number of MPI processes	Single precision		Double precision	
			2 <sup>nd</sup> order FDM (s)	8 <sup>th</sup> order FDM (s)	2 <sup>nd</sup> order FDM (s)	8 <sup>th</sup> order FDM (s)
Xeon Phi	64	1	488.8	751.6	735.1	1226.4
	32	2	397.2	658.8	628.7	1115.8
	16	4	364.2	622.2	582.5	1074.0
CPU	12	1	2566.3	4927.4	5511.1	10333.9
	6	2	1017.0	2090.2	1853.4	3501.3
	4	3	1248.4	2608.1	2715.6	5188.4
	3	4	994.0	2030.5	1845.0	3466.1
	2	6	986.5	2014.2	1846.1	3463.9
	1	12	979.9	1997.7	1846.1	3484.4

드를 사용하는 경우를 기준으로 하였다.

제온 파이와 비교하여 상대적으로 CPU에서 단정도 연산과 배정도 연산 속도 차이가 크고, 2차와 8차 유한 차분식 연산 속도 차이도 컸다. 단정도보다는 배정도 연산이, 2차 유한 차분식 연산보다는 8차 유한 차분식 연산이 메모리 속도의 영향을 더 많이 받기 때문에 위의 결과는 메모리 속도 차이에 영향을 받은 결과임을 알 수 있다. 제온 파이에서는 고대역폭 메모리를 사용하였기 때문에 상대적으로 배정도 연산과 8차 유한 차분식 연산에서 속도 저하가 적었다.

제온 파이에서 고대역폭 메모리를 사용하지 않고 모델링을 수행한 경우의 결과는 고대역폭 메모리를 사용한 경우에 비해 16개의 OpenMP 스레드를 사용한 경우 1.24 ~ 1.46배까지 느려졌다(Table 2). 64개의 OpenMP 스레드를 사용한 경우에는 3.11 ~ 3.94배까지 느려졌다. 따라서 더 많은 OpenMP 스레드를 사용할수록 메모리 병목 현상으로 인해 메모리 대역폭에 민감해지는 것을 알 수 있고, 메모리 성능이 전체 성능에 큰 영향을 끼침을 확인할 수 있다.

### OpenMP + MPI 병렬화 성능 분석

앞에서의 성능 비교는 메모리를 공유하는 OpenMP를 이용해 병렬화 한 경우의 성능 비교였다. 여러 개의 프로세스들이 메모리를 독립적으로 사용하는 MPI 병렬화를 적용하게 되면 메모리를 충분히 사용할 수 있는 CPU에 좀 더 유리해진다. 메모리가 충분할 경우 CPU 작업 시 OpenMP를 사용할 필요 없이 MPI만으로 병렬화가 가능하다. 이 경우 프로세스들이 메모리를 공유하지 않기 때문에 OpenMP의 메모리 병목 현상을 피할 수 있다. MPI는 OpenMP에 비해 공유 메모리 동시 접근

문제가 적은 대신 프로세스간 통신 문제가 존재한다. 통신 성능은 네트워크 장비의 영향을 크게 받는 문제로, 본 연구에서는 다루지 않는다. 또한 모든 MPI 프로세스에서 대량의 디스크 입출력이 발생할 경우 입출력 과정에서 병목 현상이 발생할 수 있다. 본 연구에서는 지표면의 한 축선에서만 파동장을 기록하여 디스크 입출력 병목 현상이 발생하지 않도록 하였다.

제온 파이에서도 용량이 큰 주메모리에 접근할 수 있지만 앞의 실험에서 살펴본 것처럼 고대역폭 메모리에 비하면 접근 속도가 느리고 고대역폭 메모리는 16 GB로 제한되어 있다. 따라서 메모리가 많이 필요한 연산에서 고대역폭 메모리만 사용해 최적의 성능을 얻기 위해서는 MPI의 분산형 메모리와 OpenMP의 공유 메모리 병렬 연산을 함께 사용해야 한다. 제온 파이의 64 코어를 MPI 프로세스와 OpenMP 스레드에 어떻게 할당할지는 모델링에 필요한 메모리의 양에 따라 달라진다. 예를 들어, 모델링에 4 GB의 메모리가 필요할 경우 하나의 제온 파이에서 최대 4개의 MPI 프로세스(각각 16개의 OpenMP 스레드 이용)를 할당할 수 있는 반면에 모델링에 8 GB의 메모리가 필요할 경우 최대 2개의 MPI 프로세스(각각 32개의 OpenMP 스레드 이용)를 실행할 수 있게 된다. 따라서 이번 실험에서는 제온 파이와 CPU의 모든 코어를 활용하되, OpenMP와 MPI를 동시에 사용하는 경우의 성능을 확인하였다.

그 결과 제온 파이와 CPU 모두 MPI를 최대로 활용하는 것이 성능 향상에 유리함을 확인할 수 있었다(Table 3). CPU의 경우 OpenMP 없이 12개의 MPI 프로세스를 사용하는 경우, 제온 파이의 경우 4개의 MPI 프로세스에서 각각 16개의 OpenMP 스레드를 사용하는 경우의 성능이 가장 좋았다. 모델링에 필요한 메모리 양이 적을 경우 제온 파이에서도 MPI 프

로세스를 늘리고 OpenMP 스레드를 줄이면 더 좋은 성능을 얻을 수 있을 것이라 예상할 수 있다. 실험 결과 중 가장 빠른 연산의 경우 제온 파이에서의 연산 속도는 CPU에서의 연산 속도에 비해 2.69에서 3.24배 빨랐다.

## 토 의

과동 방정식 모델링에 사용한 유한 차분 격자법은 편미분 방정식 풀이 외에도 야코비(Jacobi) 방법이나 가우스-자이델(Gauss-Seidel) 방법과 같은 행렬 풀이, 영상 처리 등에 널리 사용되는 방법으로, 격자법 풀이 속도는 일반적으로 프로세서보다 메모리 성능에 더 크게 영향을 받는다(Wikipedia, 2018). 따라서 제온 파이에서는 최적의 성능을 위해 고대역폭 메모리를 적극적으로 사용할 필요가 있다. 본 연구에서는 리눅스 numactl 명령을 이용해 제온 파이에서의 모든 연산을 고대역폭 메모리에서 실행하였지만 필요할 경우 memkind 라이브러리를 이용해 자주 사용하는 배열은 고대역폭 메모리에 할당하고 덜 사용하는 배열은 주메모리에 할당하는 방식으로 프로그램을 작성하는 것도 가능하다.

수치 실험에서 제온 파이와 CPU 모두 OpenMP 보다 MPI의 확장성이 우수했기 때문에 메모리 용량 문제가 없다면 OpenMP보다는 MPI를 이용해 병렬 연산을 수행하는 것이 성능 향상에 도움이 된다는 것을 알 수 있다. 제온 파이에서는 고대역폭 메모리가 16 GB로 제한되어 있으므로 필요 메모리 용량에 따라 OpenMP와 MPI를 동시에 이용하는 것이 좋은 성능을 얻기 위한 전략이라 할 수 있다.

본 연구에 사용한 프로그램이 가능한 최고의 성능에 가까운 성능을 보였다고 할 수는 없다. GPGPU 가속기 대비 제온 파이 가속기의 가장 큰 장점 중 하나는 기존의 CPU 프로그램을 거의 그대로 사용할 수 있다는 점인데, 최고의 성능을 위해 캐시 블로킹, 메모리 주소 조정, 내장 벡터 연산 명령, 곱셈과 덧셈 동시 연산 등의 최적화 기술(Reinders and Jeffers, 2015)을 적용하다 보면 위의 장점이 사라지게 된다. 따라서 제온 파이 x200 프로세서를 소개하기 위한 본 연구에서는 성능 최적화보다는 최적화되지 않은 기본적인 모델링 프로그램에 MPI와 OpenMP 병렬 연산 기능만 추가하여 제온 파이와 CPU에서의 상대적인 성능을 비교하는데 중점을 두었다.

본 연구에 사용한 Xeon E5-2680 v3 프로세서의 소비 전력은 120 W, 공식 가격은 \$ 1745.0이며 Xeon Phi 7210 프로세서의 소비 전력은 215 W, 공식 가격은 \$ 1881.0이다(Intel, 2018). 제온 파이를 이용해 CPU 대비 두 배 이상의 성능 향상을 얻었으므로 3차원 파동 방정식 모델링을 위해서는 제온 파이가 CPU보다 경제적임을 알 수 있다.

2017년 말에 출시된 제온 파이 나이즈 밀 프로세서는 x200 나이즈 랜딩 프로세서와 구조가 거의 유사하나 인공지능과 기계학습 분야 적용을 위해 단정도 및 반정도 연산 성능을 높이

고 배정도 연산 성능을 낮춘 모델이다. 따라서 나이즈 밀 프로세서를 이용하면 앞선 실험 결과에서 단정도 연산 성능은 향상되고 배정도 연산 성능은 저하될 것이라 예상할 수 있다. 단정도 연산만 사용하는 자료 처리에서는 새로운 나이즈 밀 프로세서를 이용하는 것이 유리하겠지만, 배정도 연산이 필수인 분야에서는 나이즈 랜딩 프로세서가 더 낫다고 할 수 있다.

## 결 론

수치 실험을 통해 제온 파이와 CPU에서 OpenMP 공유 메모리 병렬 연산과 MPI 분산형 메모리 병렬 연산을 이용한 3차원 파동 방정식 모델링 성능을 살펴보았다. 단정도와 배정도, 공간에 대해 2차와 8차 유한 차분식을 이용한 실험 결과 제온 파이에서 다량의 연산 코어와 고대역폭 메모리를 이용해 CPU 대비 2.69 ~ 3.24배 우수한 모델링 성능을 얻을 수 있음을 확인하였다. 이는 고대역폭 메모리만 사용했기 때문에 가능했던 결과로, 고대역폭 메모리 용량을 넘어서는 다량의 메모리에 자주 접근하는 연산이라면 제온 파이에서의 성능 향상 정도가 미미할 수 있다. 반면에, 고대역폭 메모리를 효율적으로 사용하도록 알고리즘을 개선한다면 더 큰 성능 향상을 얻을 수도 있을 것이다. 병렬 연산의 확장성은 MPI가 OpenMP보다 우수하지만 제온 파이에서 제한된 고대역폭 메모리를 최대로 활용하여 좋은 성능을 얻기 위해서는 MPI와 OpenMP를 함께 사용해야 함을 알 수 있었다. 최신 제온 파이와 GPGPU 장비를 구축할 수 있다면 동일한 3차원 모델을 이용해 두 연산 가속기의 모델링 성능을 비교해보는 것도 좋을 것이다.

## 감사의 글

본 연구는 산업통상자원부(MOTIE)와 한국에너지기술연구원(KETEP)의 지원(No. 20182510102470) 및 한국 지질자원연구원 주요사업인 ‘유가스 부존특성 및 해저천부구조 정밀 규명을 위한 융복합 탐사기술 개발’ 사업의 지원을 받아 수행한 연구과제입니다.

## References

- Abdelkhalek, R., Calandra, H., Coulaud, O., Roman, J., and Latu, G., 2009, Fast seismic modeling and reverse time migration on a GPU cluster, *2009 International Conference on High Performance Computing & Simulation*, 36-43.
- Heinecke, A., Breuer, A., Bader, M., and Dubey, P., 2016, High order seismic simulations on the intel Xeon Phi processor (Knights Landing). *2016 International Conference on High Performance Computing & Simulation*, 343-362.
- Intel, 2018, <https://ark.intel.com/compare/81908,94033> (August 20, 2018 Accessed)
- Jeffers, J., Reinders, J., and Sodani, A., 2016, *Intel Xeon Phi*

- processor high performance programming: Knights Landing edition*, Morgan Kaufmann, 3-145.
- Jo, S.-H., and Ha, W., 2018, 3D time-domain wave propagation modeling using high-performance Python libraries, *J. Korea Inst. Mineral Mining Eng.*, **55(3)**, 213-218 (in Korean with English abstract).
- 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(3)**, 145-152 (in Korean with English abstract).
- Lu, L., Renwei, D., Hongwei, L., and Hong, L., 2015, 3D hybrid-domain full waveform inversion on GPU, *Comput. Geosci.*, **83**, 27-36.
- Min, D.-J., Pyun, S., Ha, W., Kwak, S., Chung, W., and Shin, C., 2016, *Numerical Analysis for Geophysics*, CIR, Seoul, Korea, 37-52.
- Pacheco, P., 2011, *An introduction to parallel programming*, Morgan Kaufmann, 15-82.
- Reinders, J., and Jeffers, J., 2015, *High performance parallelism pearls: Multicore and Many-core Programming Approaches*, Morgan Kaufmann, 377-396.
- Rodriguez, S., Farré, P., Rosas, C., and Hanzich, M., 2017, Evaluating directive-based programming models on Wave Propagation Kernels, *79th EAGE Conference and Exhibition 2017-Workshops*, Paris, France.
- Ryu, D., Jo, S. H., and Ha, W., 2017, Parallelizing 3D frequency-domain acoustic wave propagation modeling using a Xeon Phi coprocessor, *Geophys. and Geophys. Explor.*, **20(3)**, 129-136 (in Korean with English abstract).
- Sourouri, M., and Birger Raknes, E., 2017, Accelerating 3D Elastic Wave Equations on Knights Landing based Intel Xeon Phi processors, *19th EGU General Assembly Conference Abstracts*, **19**, 7790p.
- Tobin, J., Breuer, A., Heinecke, A., Yount, C., and Cui, Y., 2017, Accelerating seismic simulations using the Intel Xeon Phi knights landing processor, *2017 International Supercomputing Conference, High Performance Computing*, 139-157.
- Wikipedia, 2018, [https://en.wikipedia.org/wiki/Stencil\\_code](https://en.wikipedia.org/wiki/Stencil_code) (July 30, 2018 Accessed)