JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 23, No. 2, pp. 295-302, Feb. 2022

Check for updates

이기종 컴퓨팅 환경에서 시스템 성능 검증 기법 연구

윤 준 원[1] · 송 의 성[2*]
[1]한국과학기술정보연구원 슈퍼컴퓨팅인프라센터 책임연구원
[2*]부산교육대학교 컴퓨터교육과 교수

A Study on System Performance Verification Methods for Heterogeneous Computing Environments

Junweon Yoon[1] · Ui-Sung Song[2*]

[1]Principal Researcher, Department of Supercomputing Center, KISTI, Daejeon 34141, Korea
[2*]Professor, Department of Computer Education, Busan National University of Education, Busan 47503, Korea

[요 약]

4차 산업혁명을 이끄는 핵심 기술들이 사회 전반에 확산되고 있으며 관련된 데이터의 규모와 연산 처리 요구사항 또한 기하급수적으로 증가하고 있다. 효율적인 대규모의 데이터 처리를 위해 시스템이 학습을 통해 데이터 구조와 규칙성을 찾는 AI 기반의 연구가 접목되고 있으며, 이를 위한 고성능 계산자원의 요구사항도 동시에 높아지고 있다. 최근 주목되는 GPU 기반의 이기종 시스템은 고성능 병렬처리를 위한 대표적인 도구이다.

본 연구에서는 이기종 시스템 아키텍처의 특징을 알아보고 널리 활용되고 있는 GPU 기반 컴퓨팅 환경을 구축하여 기능, 성능, 안정성 등을 검증하고자 한다. 검증을 위해 오픈소스 기반의 벤치마크 도구를 적용하여 CPU, 메모리, 네트워크, 저장장치(파일시스템) 등의 요소성능부터 GPU 기반 연산 및 이미지 처리 성능을 측정하고 안정적인 서비스 수준을 확보하고자 한다.

[Abstract]

The core technologies that led to the 4th industrial revolution are spreading throughout society, and the scale of related data and computational processing requirements are also increasing exponentially. For efficient large-scale data processing, AI-based research that finds data structure and regularity through system learning is being grafted, and the requirements for high-performance computational resources are also increasing at the same time. The recently emerging GPU-based heterogeneous system is a representative tool for high-performance parallel processing.

In this paper, we examine the characteristics of heterogeneous system architectures and build a widely used GPU-based computing environment to verify functions, performance, and stability. For verification, we apply an open source-based benchmark tool to measure the performance of elements such as CPU, memory, network, and storage(file system), as well as GPU-based computation and image processing performance to identify characteristics and secure a stable service level.

색인어 : 이기종 컴퓨팅, 인공지능, 아키텍처, 벤치마크, GPU

Keyword : Heterogeneous Computing, AI, Architecture, Benchmark, GPU

http://dx.doi.org/10.9728/dcs.2022.23.2.295

Received 01 December 2021; **Revised** 05 January 2022
Accepted 05 January 2022

***Corresponding Author; Ui-Sung Song**

Tel: +82-51-500-7326
E-mail: ussong@bnue.ac.kr

Ⅰ. 서 론

인공지능(AI), 사물인터넷(IoT), 가상·증강현실(VR·AR)과 같은 4차 산업혁명의 주요 기술들이 다양한 분야에 적용되고 있다. 이런 기술들은 대량의 데이터를 생산하고 수많은 연산 처리를 요구한다. 따라서 대규모의 데이터들을 신속하게 분석하고 처리하기 위한 다양한 연구 방법들이 시도되고 있다. 일반적으로 빅데이터라는 용어는 데이터의 규모와 형태(정형, 비정형) 그리고 분석 방법까지를 포괄한다. 즉, 대규모 데이터의 수집·저장·분석·처리에 대한 기술까지가 범주에 해당한다[1]. 최근 머신러닝, 딥러닝과 같은 AI 연구가 접목되면서 빅데이터에 대한 가치와 효용성이 더욱 높아지고 있으며, 그에 따라 계산자원의 요구사항도 더욱 높아지고 있다.

1960년대 중반 인텔의 공동 창업자인 고든 무어(Gordon Moore)는 반도체의 집적회로 성능은 2년마다 2배로 증가한다는 무어의 법칙을 제시했으나 물리적인 직접도의 한계와 경제적 수익성에 대한 제고로 2010년 중반부터는 실현이 불가능하다고 판단하고 있다[2]. 전력소모량과 발열에 대한 단일 코어의 집적도 제한은 코어 수를 늘려 성능을 향상하기 위한 멀티코어, 매니코어 프로세서로 진화되었다. 국내 공공목적의 계산자원을 제공하는 KISTI 슈퍼컴퓨터 5호기 누리온(Nurion)은 인텔 매니코어(Many-core) 아키텍처인 KNL(68 cores) CPU가 탑재된 대표적인 시스템이다[3].

대규모 데이터 처리를 위하여 인공지능을 활용한 연구가 여러 분야에서 활성화되고 있다. 이에 따라 계산자원 성능에 대한 요구사항과 의존성이 높아지면서 이기종 시스템(Heterogeneous System)에 대한 수요가 증가하고 있다. 기존에 일반적으로 사용되던 방식은 동일한 성격의 프로세서를 사용하는 방식(Homogeneous System)이였다면 이기종 시스템은 하나의 소프트웨어가 서로 다른 프로세서 가령, CPU나 GPU를 모두 활용하여 연산을 수행하게 된다. 즉, CPU와 GPU를 하나의 계산자원으로 추상화하고 내부적으로는 CPU는 데이터 처리를 위해 GPU는 연산 가속을 위해 사용된다[4]. 수년간 NVIDIA GPU 제품군을 필두로 다양한 이기종 시스템들이 지속적으로 개발, 상용화되고 있으며 단일노드 계산 성능 집적도 또한 큰 증가폭으로 상승하고 있다.

그림 1은 SC21('21.11) 기준 이기종 아키텍처의 TOP500 점유율을 나타내고 있으며 NVIDIA 제품군의 대부분을 확보하고 있다[5].

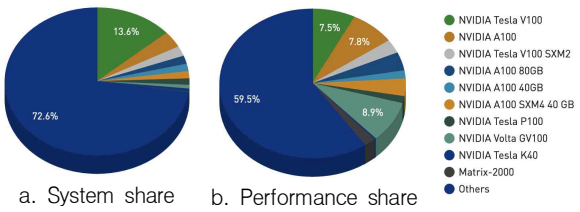

그림 1. TOP500 가속기/코프로세서 시장 점유율
Fig. 1. TOP500 share of Accelerator/Co-Processor

컴퓨터의 수치 계산은 대부분 부동소수점 연산(floating point operation)으로 구성되며, 이를 수치화한 (theoretical peak performance)을 통해 시스템 성능을 나타낸다. 이론성능은 시스템에 물리적인 특성을 반영하여 아래와 같이 계산된다.

• Theoretical Performance (Rpeak) =
Number of cores × Floating point instructions per Cycle × Clock Frequency

앞서 언급한 동일한 CPU 기반 아키텍처만을 사용하는 누리온 슈퍼컴퓨터의 단일 계산노드 이론성능을 계산하면 FP64(배정밀도 부동소수점, Double-precision floating-point format) 기준으로 약 3TF(Rpeak)를 갖는다.

한편, 슈퍼컴퓨터 5호기에는 GPU 기반의 연구를 지원하기 위한 뉴론(Neuron) 시스템을 동시에 구축하여 서비스 중이다. 이 시스템은 NVIDIA GPU 제품군이 장착되어 있으며 이기종 시스템에 해당된다. 뉴론 시스템은 다양한 계산노드들로 구성되어 있으며, 그 중 NVIDIA V100 GPU 8개를 탑재한 계산노드의 경우 동일하게 FP64 기준으로 약 65TF(Rpeak)의 이론성능을 갖는다[3]. 두 시스템의 단일 계산노드 성능이 약 20배 이상 차이가 난다. 이기종 아키텍처는 병렬처리 성능을 집중시킴으로써 단일 노드의 집적도를 높이고 높은 성능 향상을 얻을 수 있다. 계산 목적에 따라 GPU, FPGA, Xeon Phi 등의 다양한 이기종 컴퓨팅 아키텍처가 중대규모의 클러스터 시스템 환경 구축이 점차 확대되고 있다.

최근 HPC 시장은 엑사(1018)플롭스 급의 시스템이 등재되면서 본격적인 성능의 단위가 도약하는 시점이다. 이기종 시스템은 단일노드의 성능 밀도를 높여 상대적으로 기존 단일 아키텍처 시스템에 비해 적은 계산노드로 고성능의 시스템 구성이 가능하다. 또한, NVLink, GPUDirect 등의 기술을 통해 이기종 시스템의 연산 수행처리 성능을 더욱 가속화시키고 있다. AMD, Intel와 같은 CPU 개발 업체 또한 이 대열에 합류하여 제품군을 출시하고 있어 시장의 확대와 발전 전망이 뚜렷하다.

본 연구에서는 이기종 시스템의 특징과 대표적인 기술들을 조사하고 성능을 평가하여 실제 수행하고자 하는 연구자에게 지표를 제시할 수 있다.

Ⅱ. 관련 연구

이기종 컴퓨팅은 일반적으로 CPU 외의 다른 프로세서를 동시에 사용하여 연산을 가속화하는 환경이다. 기존의 동일한 성격의 CPU만을 활용해서 수행하는 환경(Homogeneous System)은 싱글코어의 집적도 한계로 멀티코어(Multi-core) 또는 매니코어(Many-core) 환경으로 발전해 왔다. 하지만 단일 노드의 공유 메모리 환경은 결국 메모리

대역폭의 제한으로 성능 향상에 한계가 발생한다. 이기종 컴퓨팅은 이런 성능의 한계를 극복할 수 있는 대표적인 기술로 다른 종류의 연산처리 장치를 이용한다. 보조 프로세서(Coprocessor) 또는 가속기(Accelerator)라고도 불리기도 했으며, 이기종 컴퓨팅 환경은 이런 서로 다른 처리 장치를 모두 활용하여 고성능의 연산처리 환경을 제공한다. 나아가 이기종 시스템 아키텍처(Heterogeneous System Architecture, HSA)는 최근 다양하게 적용되는 GPU 기반의 시스템에서 보면 CPU와 GPU를 하나의 연산장치로 추상화하여 제공한다. 여기에는 하드웨어 뿐만 아니라 커널 및 드라이버, 라이브러리, 개발도구 등의 소프트웨어 개념까지 포함된다. 그림 2은 HSA의 개념을 나타낸다[6].

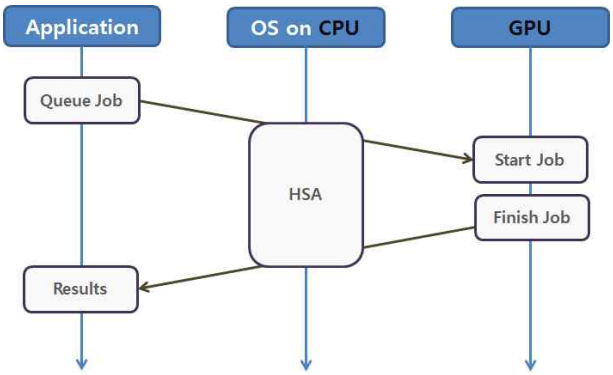

그림 2. GPU 환경에서 HSA 수행 방식
Fig. 2. HSA execution method in GPU environment

최근 몇 년간 인공지능 발전을 주도한 딥러닝(Deep Learning) 기술은 인간 뇌의 신경망(Neural network) 구조를 묘사한 것으로 뉴런(Neuron)이라 불리는 작은 세포 단위가 연속적으로 연결된 구조이다.

뉴런은 시냅스 (Synapse)라는 연결 세포를 통해 이전에 연결된 뉴런으로부터 다수의 전기 자극을 입력으로 받아 저장하거나 또는 새로운 자극을 만들어 다음 뉴런으로 넘겨준다. 인간의 뇌는 뉴런이라 불리는 뇌세포가 1,000억 개나 있고 뉴런들의 연결인 시냅스 연결이 100조 개나 있다. 수천, 수만 개의 큰 규모의 문제를 풀기 위해서는 이런 연산을 반복해야 한다. 결국 딥러닝은 컴퓨터의 성능에 따라 결정되며 특히나 병렬처리(Parallel processing) 능력이 이를 결정하게 된다[7].

일반적인 CPU는 대규모 병렬연산에는 적합하지 못했다. GPU는 초기 병렬처리를 통해 수많은 픽셀(Pixel)을 처리하는 기술로 CPU보다 상대적으로 낮은 클럭 수를 갖고 있지만 수 천개의 코어를 배치하여 병렬처리의 성능을 최대한 끌어올릴 수 있다. NVIDIA의 GPU는 범용(GPGPU) 및 병렬 플랫폼(CUDA) 기술을 앞세워 수년 동안 범용 GPU 계산자원의 높은 시장 점유율을 확보하고 있다.

GPU의 경우 수많은 산술연산장치(ALU)를 배치하고 단일 명령어로 복수의 데이터를 쓰레드 처리(SIMT, Single Instruction, Multiple Threads)하는 방식이다. 가장 기본적인 단위는 SP(Scala 또는 Stream Processor)로 이것들을 모아 SM(Streaming Multiprocessor)로 구성된다. SM은 한 번에 처리할 수 있는 쓰레드로 그룹으로 워프(warp)라고 부른다. SM 내의 SP는 레지스터와 메모리를 공유하며 GPU의 세대마다 SM, SP의 개수 차이가 난다[8][9].

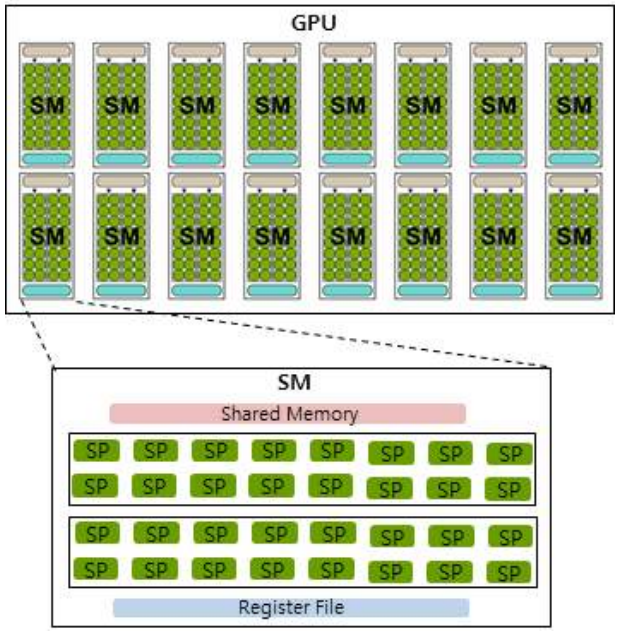

그림 3. NVIDIA GPU 아키텍처 구조
Fig. 3. Schematic of NVIDIA Architecture

Ⅲ. 시스템 환경

본 연구를 위해 이기종 시스템으로 널리 사용되는 범용 GPU 기반의 시스템을 구축하여 성능을 검증하고자 한다. Intel Xeon 기반의 CPU와 전용 계산 가속을 위한 NVIDIA의 V100 GPU가 장착된 시스템으로 구성은 표 1과 같다. InfiniBand 기반의 인터커넥트로 모든 계산노드와 스토리지가 밀결합 되어있는 클러스터 환경으로 8대의 이기종 아키텍처가 탑재된 계산노드를 사용하여 성능을 검증하게 된다.

표 1. GPU 기반 시스템 환경
Table 1. GPU-based System Environment

Type	Specifications
Node	[8 nodes with Intel CPU and NVIDIA GPU] · **CPU:** Intel Xeon Gold 6258R(Cascade Lake) 2.70GHz 28c 총 56(28x2)cores · **Memory:** 384GB 6CH (32GB X 6slots X 2sockets) · **GPU:** Nvidia V100 PCIe(32GB) 장착
Storage	DDN SFA12K(2.3PB)/Theoretical Aggregation BW 38GB/s
Interconnect	Storage(FDR10,40Gbps), Node(EDR,100Gbps)

Ⅳ. 성능 분석

4-1 시스템 성능 검증

성능 검증은 자체 서버의 성능(Stream, HPL), 파일 I/O 성능(IOR, MDTEST), GPU 성능(tf_cnn_benchmark, HPL-NVIDIA), 실제 GPU 기반의 이기종 시스템에서 수행되는 현업 어플리케이션(GROMACS) 테스트를 수행한다. BMT 검증은 이기종 시스템의 다양한 요소 성능을 표 2와 같이 측정한다.

표 2. BMT 성능 및 기능 시험 측정 항목
Table 2. BMT performance and functional lists

S/W	Unit	Features and evaluation points
STREAM	GB/s	Measuring bandwidth between processor and memory on a single node
IOR	GB/s	Measurement of I/O performance in parallel filesystem
MDTEST	IOPS	Measuring the performance of metadata processing in parallel filesystem
HPL	TFLOPS	Measuring Computational Performance in a Distributed Memory Environment
tf_cnn_ benchmark	images/ sec	Measuring the performance of training in deep learning
HPL-NVIDIA	TFLOPS	Measuring the computational performance of GPU powered by NVIDIA
GROMACS	ExeTime	Molecular dynamics analysis program

4-2 벤치마크 테스트 (BMT)

1) STREAM

STREAM은 단일노드에서 프로세서와 메모리 간의 대역폭을 측정하는 도구로 빌드를 위해서 C 컴파일러가 필요하다. 공유메모리 환경에서 멀티쓰레드(Multi-thread) 방식으로 수행되며 쓰레드 개수(OMP_NUM_THREADS) 설정을 변경하여 최적의 성능을 측정한다[10]. 시스템의 구성 내역은 표 3과 같으며 성능은 이론성능 대역폭(256GB/s) 대비 약 88%가 측정되었다. 표 4는 최대 성능과 그때의 쓰레드 수를 나타낸다. 그림 4는 쓰레드 수 증가에 따른 STREAM 성능을 보여준다.

표 3. 시스템 구성 및 성능
Table 3. System configuration and performance

[Main memory configurations] ·6 channels of DDR4, up to 2666 MT/s(RDIMM and LRDIMM) ·Bandwidth of 21.33 GB/s ·Aggregated bandwidth 128 GB/s per CPU → 128GB/s X 2CPUs= 256 GB/s

표 4. STREAM 테스트 결과
Table 4. STREAM test results

Type (Processor&Memory)	# of Optimal threads	Triad rate (MB/s)
Intel Xeon Gold 6258R 2.70GHz X2 56(28x2)cores/ DDR4(2666 MT/s) 384GB(6CH)	27	225,348.8

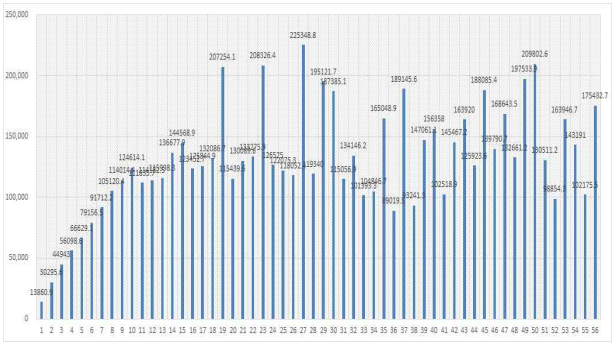

그림 4. Thread 수에 따른 STREAM 성능
Fig. 4. STREAM performance by the number of threads

2) HPL

분산 메모리 환경에서 연립방정식 해석을 통해 실수 연산 성능을 측정하기 위한 벤치마크 테스트로 HPL 코드를 수행한다. 컴파일러(C, Fortran 등), Message Passing Interface(MPI) 병렬 라이브러리, 수치 라이브러리(BLAS, CBLAS, ATLAS, MKL 등) 설치가 사전에 설치되어 있어야 한다[11]. 본 시스템 환경의 물리적인 이론성능(Rpeak)은 배정도 연산(FP64) 기준으로 약 4.8TFlops로 표 5와 같이 계산된다.

표 5. CPU 사양 및 이론성능
Table 5. CPU specification and theoretical performance

·Intel(R) Xeon(R) Gold 6258R CPU @ 2.70GHz (28Cores)X2 AVX-512 FMA units per Core ·(8 x 2) x 2 x 2.7G x 28cores x 2cpu = 4,838.4GF

성능 벤치마크는 인텔에서 컴파일러에서 배포하는 Math Kernel Library(MKL) HPL binary(xhpl_intel64)를 사용하여 수행하였다. 이론성능과 비교하여 단일노드에서는 64.85%, 8노드에서는 59.78%가 측정되었다. 그림 5는 노드수의 확장에 따른 HPL 성능을 보여준다.

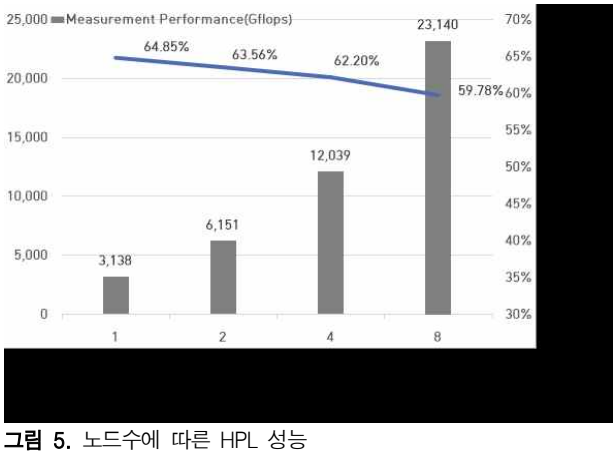

그림 5. 노드수에 따른 HPL 성능
Fig. 5. HPL results by the number of nodes

인텔의 Cascade Lake는 Skylake 마이크로 아키텍처의 후속 모델로 내부 코어는 메쉬 인터커넥트(Mesh interconnect) 구조를 동일하게 사용하였다[12]. 상대적으로 코어의 개수가 많은 모델일수록 캐시 일관성(cache coherence issue)의 부하로 성능 저하가 발생한다.

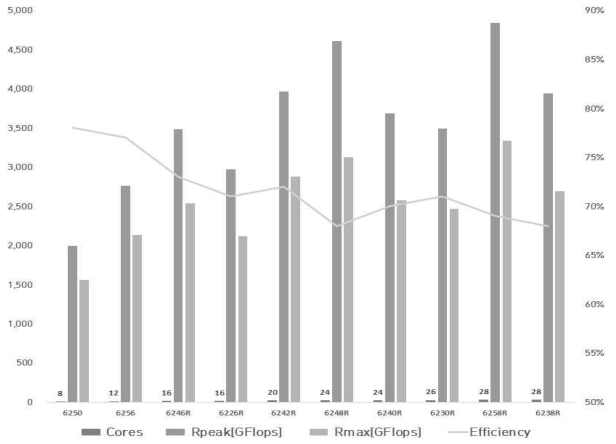

그림 6. CPU 코어수에 따른 HPL 실측 성능 비율
Fig. 6. HPL Rmax performance ratio by the number of CPU cores

그림 6은 Intel Xeon Cascade Lake 계열 CPU의 코어수에 따른 성능 비율을 나타내고 있으며 코어 수가 많은 모델일수록 실측성능이 조금씩 낮아짐을 볼 수 있다[13].

3) IOR

IOR은 클러스터 환경에 구성된 공유 병렬파일시스템의 I/O 대역폭을 측정하는 도구로 POSIX, MPIIO, HDF5 등과 같은 인터페이스와 파일 접근 패턴을 이용하여 테스트한다[14]. IOR 또한 MPI 병렬 라이브러리를 사용하여 노드의 수를 증가하며 성능을 측정하는데 최대 대역폭은 스토리지에 구성된 물리적 대역폭에 수렴하게 된다. 테스트에 수행되는 노드는 병렬파일시스템을 마운트하여 각 프로세스마다 파일 읽기와 쓰기를 처리한다. 벤치마크 테스트 결과는 Max Write(MB/s)와 Max Read(MB/s)의 값 중 최대치를 적용하며 테스트 수행시 병렬파일시스템의 정확한 성능 측정을 위해 노드에 구성된 메모리의 캐시 효과를 배제하도록 수행한다. 파일 접근 방식에 따른 명령어와 옵션 (POSIX, MPIIO, Single-shared-file, File-per-process)을 설정하고 노드당 20개의 프로세스를 사용하여 표 6과 같이 수행한다.

표 6. IOR 벤치마크 수행 옵션
Table 6. IOR benchmark options and execution

```
#IOR.posix -a POSIX -b 10g -o iorData -t 10m -s 2 -d 10 -C
-Q 25 -e -w -r -k (-F)
#IOR.mpiio -a MPIIO -b 10g -o iorData -t 10m -s 2 -d 10 -C
-Q 25 -e -w -r -k (-F)
```

실험환경에 구축된 병렬 파일시스템의 대역폭은 다음과 같이 구할 수 있다.

• (IB FDR-10) 40Gb/s X (64/66) X 8 Ports ≒ 38 GB/s

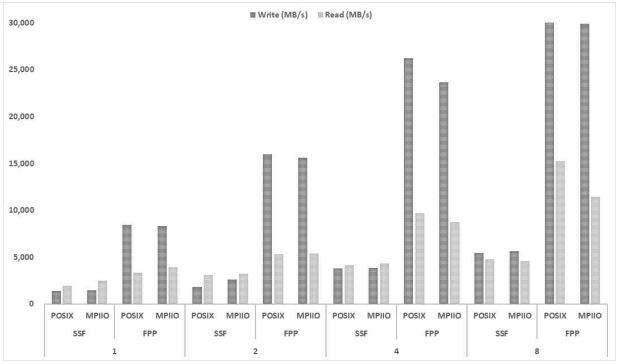

그림 7. 노드수에 따른 IOR 성능
Fig. 7. IOR results by the number of nodes

그림 7에서 IOR 쓰기 수행시 커밋 발생은 실제 디스크가 아닌 메모리에 캐싱되는 시점(write-back)으로 디스크에서 실제 접근하는 읽기 성능에 비해 상대적으로 높은 성능을 보인다.

4) Tf_cnn_benchmarks

이기종 아키텍처 서버에 장착된 GPU 성능 측정을 위해 딥러닝에 많이 사용되는 Tensorflow 코드를 이용한 벤치마크인 Tf_cnn_benchmarks를 수행하였다[15]. 테스트는 GPU 개수를 증가하며 이미지 처리 확장성을 테스트 할 수 있다. 해당 코드는 Github를 통해 받을 수 있으며 ImageNet ILSVRC2012 데이터 셋은 보유 자료를 활용하였다. 성능 측정을 위해 CUDA, MPI 병렬 라이브러리, 수학 라이브러리(cuBLAS, cuDNN 등), 프레임워크 구동환경(conda, docker 등) 설치가 필요하며 단일노드와 전체 시스템을 사용하여 낼 수 있는 최대 이미지 처리 성능(images/sec)을 측정한다. ResNet-50 모델 학습 과정 로컬 배치 크기 256, TF32 데이터타입, ILSVRC2012 데이터셋에 대하여 1 epoch 테스트를 수행하고 결과는 Total images/sec 값의 평균치를 수용한다. 수행 명령 및 옵션은 표 7과 같다.

표 7. Tf_cnn_benchmarks 벤치마크 수행 옵션
Table 7. Tf_cnn_benchmark benchmark options and execution

```
$ mpirun -np [#ps] [#MPIOPT] python tf_cnn_benchmarks.py
--data_format=NCHW
--batch_size [#batch] --model [#model]
--optimizer=momentum
--variable_update=horovod —nodistortions --num_gpus=1
--num_epochs=[#epoch]
--weight_decay=1e-4—data_dir=<DATA>
--horovod_device=gpu
※ #ps: the number of processes, #MPIOPT: MPI options, #batch:
local batch size, #model: CNN model, #epoch: total learning
number <DATA>: dataset directory
```

단일노드에는 NVIDIA의 V100 GPU가 1개씩 장착되어 있으며, 8개 노드(V100 8 GPU)까지 확장하며 성능 실험을 수행하였다. 표 8과 같이 단일노드에서는 380.86 image/s, 8개의 노드 전체 사용시 2,841.46 images/s의 처리 속도를 얻었다. 그림 8은 노드를 증가시키며 수행한 확장성 테스트 결과이다. 참고로 NVIDIA에서 제공하는 Docker Resnet-50 image를 이용한 DGX 1(8xV100 16G) 시스템에서 성능은 1 Node(412img/s), 8 Node(3,170img/s)를 보인다[16]. DGX는 NVIDIA의 전용 플랫폼으로 네트워크, 메모리, NVLink 등의 특화된 H/W가 장착되어 실험 결과보다 상위의 성능을 나타낸다.

표 8. TensorFlow ResNet-50 테스트 결과
Table 8. TensorFlow ResNet-50 test results

System	Performance (images/sec)
Single Node	380.86
Full Nodes(8)	2,841.46

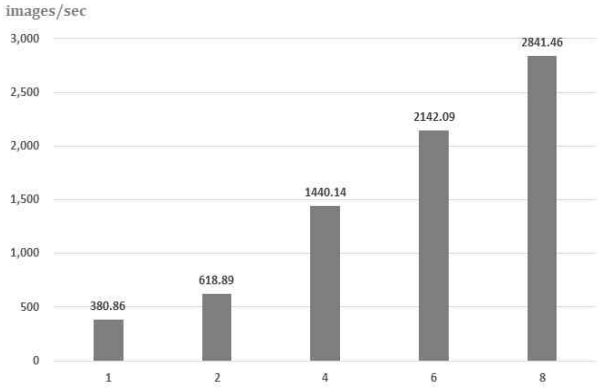

그림 8. TensorFlow ResNet-50 확장성 테스트 결과
Fig. 8. TensorFlow ResNet-50 Scalability Test Results

5) HPL-NVIDIA

NVIDIA는 NGC(NVIDIA GPU Cloud)를 통해 GPU의 연산 성능을 측정할 수 있는 NVIDIA HPC-Benchmarks docker 이미지를 제공하고 있다[16]. 단일노드에 장착된 PCIe 기반 V100의 성능(Double-precision, FP64)은 7 TFlops로 벤치마크 테스트 결과 약 80% 수준의 성능을 얻을 수 있다[17]. 수행 명령은 표 9와 같으며 시스템에 장착된 CPU와 GPU의 구조에 따라 cpu-affinity와 gpu-affinity 옵션을 설정해야 한다. 그림 9는 GPU가 1개씩 장착된 4대의 노드를 사용하여 테스트를 수행한 결과이다. HPL의 행렬 연산 크기는 GPU 내부에 장착된 고속메모리(HBM) 맞게 수행되어 상대적으로 높은 성능을 얻게 된다. 표 8은 벤치마크를 통해 얻은 성능 결과 로그이다.

표 9. HPL-NVIDIA 벤치마크 수행 옵션
Table 9. HPL-NVIDIA benchmark options and execution

```
$ mpirun -np [#gpu] singularity run --nv -B $PWD:/input ./hpc-benchmarks₩:20.10-hpl.sif hpl.sh --cpu-affinity 0:0:0:0:1:1:1:1 —gpu-affinity 0:1:2:3:4:5:6:7 —cpu-cores-per-rank 4 --dat HPL.dat
※ #gpu: the number of GPU for BMT
```

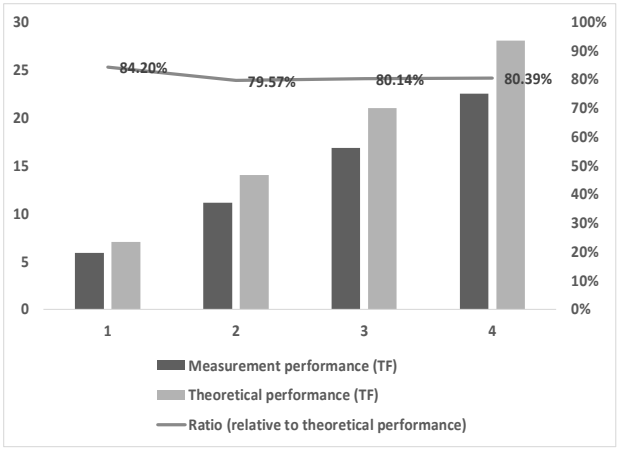

그림 9. NVIDIA HPC 벤치마크 성능 및 확장성 테스트 결과
Fig. 9 NVIDIA HPC-Benchmarks performance and test Results

표 10. NVIDIA HPC-BMT 수행 결과 로그
Table 10. NVIDIA HPC-BMT performance results logs

```
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR03L2L2       60000   512     1     1              24.43             5.894e+03
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0047216 ..... PASSED
================================================================================
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR03L2L2       60000   512     2     1              12.73             1.132e+04
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0044848 ..... PASSED
================================================================================
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR03L2L2      102400   512     2     2              31.57             2.267e+04
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0039033 ..... PASSED
================================================================================
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR03L2L2      128000   512     3     2              40.44             3.457e+04
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0040469 ..... PASSED
================================================================================
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR03L2L2      204800   512     4     2             123.60             4.633e+04
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0000368 ..... PASSED
================================================================================
```

6) Application 테스트(GROMACS)

실제 GPU 기반에서 수행 가능한 어플리케이션을 선정하여 수행 성능 결과를 측정하였다. 이를 위해 분자동력학 시뮬레이션으로 오픈소스 프로그램 패키지인 Groningen MAchine for Chemical Simulation(GROMACS) 설치하고

NVIDIA GPU를 사용하여 테스트를 수행하였다[18]. 그림 10은 GROMACS의 성능 확장성 테스트 결과로 노드와 GPU의 수가 증가할수록 수행시간이 단축됨을 볼 수 있으나 확장성에 따른 시간 단축은 크지 않았다.

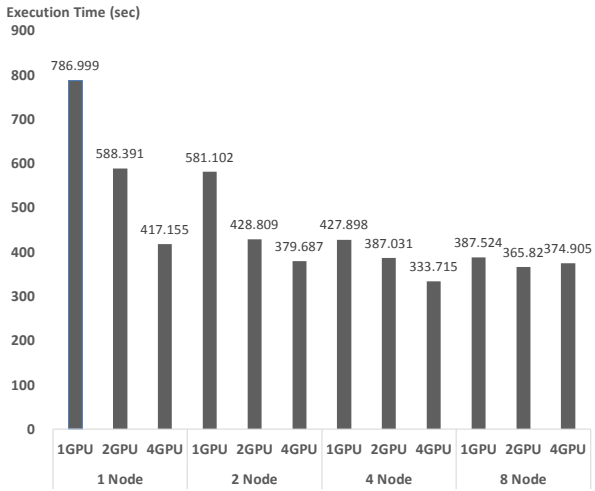

그림 10. GROMACS 성능 및 확장성 테스트 결과
Fig. 10. GROMACS performance and test Results

Ⅴ. 결 론

이기종 시스템은 단일노드의 성능 집적도를 향상시켜 적은 계산노드로 고성능의 시스템 구현이 가능하다. 순차 연산 처리 기반의 CPU 환경은 최근 빅데이터, AI 기반의 연구들이 대규모의 병렬 연산을 요구함으로써 이기종 아키텍처 기반의 시스템으로 시장 점유율이 확장되고 있다.

본 연구는 이기종 시스템을 사전에 검증하거나 도입 시점에 객관적인 성능 지표를 측정하기 위한 방법을 제시한다. 최근 범용 이기종 시스템 아키텍처로 각광받고 있는 NVIDIA의 GPU 기반 시스템 환경을 구축하였다. 이기종 시스템의 특징을 반영한 요소성능, 이기종 아키텍처인 GPU 성능 그리고 인터커넥트를 통해 연결된 스토리지의 I/O 성능을 측정하였다. 나아가 실제 GPU 기반의 어플리케이션을 수행하여 효용성 테스트 또한 검증한다. 성능 실험은 단일노드부터 구축된 전체 클러스터까지 확장성 실험을 수행하여 병렬연산 수행의 안정성을 명확히 확보할 수 있다.

서론에서 언급하였듯이 최근 이기종 아키텍처 시스템은 NVIDIA의 GPU가 대부분의 점유율을 확보하고 있다. 본 논문에서 제시한 벤치마크 수행 내역은 도입 타당성 및 안정적인 시스템 확보 방안으로 제시될 수 있다.

최근 AI 연산의 요구가 급증함에 따라 NVIDIA GPU 뿐만 아니라 AMD GPU(MI 시리즈), Intel GPU(Ponte Vacchio, Graphcore Intelligence Processing Unit(IPU) 등 다양한 제품들이 시장에 출시되고 있다. 나아가 CPU, GPU, 메모리를 단일 칩에 집적(System On Chip, SoC)하여 대역폭을 높이고 연산 지연을 축소하기 위한 아키텍처를 제시하고 있다. 이런 다양한 이기종 아키텍처를 연구하고 소규모 시스템을 구축하여 성능 검증을 위한 연구를 지속적으로 수행할 예정이다.

감사의 글

본 연구는 2021년도 부산교육대학교 학술연구과제로 지원을 받아 수행되었음.

참고문헌

[1] G. Mouzhi, H. Bangui, and B. Buhnova, “Big data for internet of things: a survey,” *Future generation computer systems 87,*pp. 601-614, October 2018.

[2] Khan, Hassan N., David A. Hounshell, and Erica RH Fuchs, “Science and research policy at the end of Moore’s law,” *Nature Electronics* 1.1, pp. 14-21, January 2018. https://doi.org/10.1038/s41928-017-0005-9

[3] KISTI Supercomputing Center (KSC). Available: http://www.ksc.re.kr/.

[4] MA. Guzmán, Dávila, R. Nozal, RG. Tejero, M. Villarroya -Gaudó, Gracia, DS., Bosque, J. L., “Cooperative CPU, GPU and FPGA heterogeneous execution with EngineCL,” *The Journal of Supercomputing* 75.3, pp.1732-1746, March 2019. https://doi.org/10.1007/s11227-019-02768-y

[5] Top500, List-November, “Top500 supercomputer sites,” November 2021. Available: http://top500.org/.

[6] Rogers, Phil, and A. Fellow, “Heterogeneous system architecture overview,” *Hot Chips Symposium*, pp. 1-41, August 2013.

[7] Roh, Yuji, G Heo and S. E. Whang, “A survey on data collection for machine learning: A big data-AI integration perspective,” *IEEE Transactions on Knowledge and Data Engineering*, pp. 1328-1347, Vol. 33, No 4, April 2021. https://doi.org/10.1109/TKDE.2019.2946162

[8] Owens, J., “GPU architecture overview,” *In ACM SIGGRAPH 2007 courses*, pp. 2-es, August 2007. https://doi.org/10.1145/1281500.1281643

[9] M. Gebremedhin, A. H. Moghadam, P. Fritzson, and K. Stavåker, “A data-parallel algorithmic modelica extension for efficient execution on multi-core platforms,” *In Proceedings of the 9th International MODELICA Conference,* Munich Germany, Linköping University Electronic Press., No. 76, pp. 393-404, September 3-5, 2012.

[10] JD. McCalpin, “STREAM: Sustainable memory bandwidth

in high performance computers," Technical Report, University of Virginia, Charlottesvile, VA 1991-2007. Available: http://www.cs.virginia.edu/stream/.

[11] Luszczek, P., Dongarra, J. J., Koester, D., Rabenseifner, R., Lucas, B., Kepner, J., and Takahashi, D, "Introduction to the HPC challenge benchmark suite," Ernest Orlando Lawrence Berkeley NationalLaboratory, Berkeley, CA US, No. LBNL-57493, November 2005.

[12] M. Arafa, B. Fahim, S. Kottapalli, A. Kumar, LP. Looi, S. Mandava and S. Vora, "Cascade lake: Next generation intel xeon scalable processor," IEEE Micro, Vol. 39, No. 2, pp. 29-36, February 2019. https://doi.org/10.1109/MM.2019.2899330

[13] Technical data, Measurements with 2nd Generation Intel Xeon Processor Scalable Family. Performance Report PRIMERGY RX2540 M5 Version 1.0, SPECcpu, April 30, 2019.

[14] H Shan, J. Shalf, "Using IOR to analyze the I/O performance for HPC platforms," Ernest Orlando Lawrence Berkeley NationalLaboratory, Berkeley, CA US, No. LBNL-62647, June 2007.

[15] TensorFlow benchmarks(tf_cnn_benchmarks). Available: https://github.com/tensorflow/benchmarks/.

[16] NVIDIA NGC's HPC-Benchmarks. Available: https://catalog.ngc.nvidia.com/orgs/nvidia/containers/hpc-benchmarks/.

[17] S. Markidis, SW. Der Chien, E. Laure, IB. Peng, and JS. Vetter, "Nvidia tensor core programmability, performance and precision," *In 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, IEEE, pp. 522-531, May 2018.

[18] D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, AE. Mark, and HJ. Berendsen, "GROMACS: fast, flexible, and free. Journal of computational chemistry," Vol. 26, No. 16, pp. 1701-1718, October 2005. https://doi.org/10.1002/jcc.20291

윤준원(Junweon Yoon)

2002년~2004년 : 고려대학교 대학원 컴퓨터학과 (이학석사)
2010년~2018년 : 고려대학교 대학원 컴퓨터학과 (공학박사)
2005년~현 재 : KISTI 국가슈퍼컴퓨팅본부 책임연구원

※관심분야: 분산컴퓨팅, 결함포용시스템, 슈퍼컴퓨팅, 병렬파일시스템, 배치스케줄링, 벤치마크(BMT)

송의성(Ui-Sung Song)

1997년~1999년: 고려대학교 대학원 컴퓨터학과 (이학석사)
1999년~2005년: 고려대학교 대학원 컴퓨터학과 (이학박사)
2006년~현 재: 부산교육대학교 컴퓨터교육과 교수

※관심분야: 컴퓨터교육, 교육용로봇교육, 컴퓨터네트워크, 스마트러닝