

SC20를 통해 본 HPC 기술 동향

HPC Technology Through SC20

어익수 (I.S. Eo, iseo@etri.re.kr)

슈퍼컴퓨팅기술연구센터 책임연구원

모희숙 (H.S. Mo, hsmo@etri.re.kr)

슈퍼컴퓨팅기술연구센터 책임연구원

박유미 (Y.M. Park, parkym@etri.re.kr)

슈퍼컴퓨팅기술연구센터 책임연구원/센터장

한우중 (W.J. Han, woojong.han@etri.re.kr)

인공지능연구소 책임연구원/연구위원

ABSTRACT

High-performance computing (HPC) is the underpinning for many of today's most exciting new research areas, to name a few, from big science to new ways of fighting the disease, to artificial intelligence (AI), to big data analytics, to quantum computing. This report captures the summary of a 9-day program of presentations, keynotes, and workshops at the SC20 conference, one of the most prominent events on sharing ideas and results in HPC technology R&D. Because of the exceptional situation caused by COVID-19, the conference was held entirely online from 11/9 to 11/19 2020, and interestingly caught more attention on using HPC to make a breakthrough in the area of vaccine and cure for COVID-19. The program brought together 103 papers from 21 countries, along with 163 presentations in 24 workshop sessions. The event has covered several key areas in HPC technology, including new memory hierarchy and interconnects for different accelerators, evaluation of parallel programming models, as well as simulation and modeling in traditional science applications. Notably, there was increasing interest in AI and Big Data analytics as well. With this summary of the recent HPC trend readers may find useful information to guide the R&D directions for challenging new technologies and applications in the area of HPC.

KEYWORDS 고성능컴퓨팅, 슈퍼컴퓨팅, 병렬컴퓨팅, 병렬프로그래밍

1. 서론

슈퍼컴퓨터는 기존의 과학 연구와 기술 개발의 활용에서 사람과 집단에서 만들어지는 빅데이터

를 분석하거나 사람의 인지기능을 학습하여 새로운 인공지능을 구현하는 등 그 활용 범위를 넓혀가고 있다. 새로운 슈퍼컴퓨터의 고성능 요구는 엑사스케일급의 컴퓨터 시대를 이끌었으며, 연구와 개

* DOI: <https://doi.org/10.22648/ETRI.2021.J.360313>

* 이 논문은 대한민국 정부(과학기술정보통신부)의 재원으로 한국연구재단 슈퍼컴퓨터개발선도사업의 지원을 받아 수행된 연구임[과제번호: 2021M3H6A1017683].



본 저작물은 공공누리 제4유형

출처표시+상업적이용금지+변경금지 조건에 따라 이용할 수 있습니다.

©2021 한국전자통신연구원

발이 국가 과학기술의 수준을 보여주고 있어 국가 별로 개발에 매진하고 있다. SC 학회는 매년 미국에서 열리는 학회로 슈퍼컴퓨터의 여러 분야에 걸친 연구 논문 발표와 개발 결과물이 전시되는 종합 학술대회로 슈퍼컴퓨터의 연구 개발 동향을 한눈에 볼 수 있어 국내 주요 연구기관들도 참여하여 그 학술대회 동향을 분석, 보고하고 있다. 이번 코로나19 팬데믹 상황에서도 SC20은 2020년 11월 9일부터 19일까지 온라인으로 개최되었으며, 미국 동부시간 기준으로 오전 10시에 시작하였고 한국 표준시로 오전 10시에 한 번 더 재전송하였다. 참가 화면상에서 질문을 입력하면 진행자가 발표 후에 질문을 전달하는 방식으로 진행하였으며 동영상은 학회 종료 후 6개월 동안 참가자들에게 공개되고 있다.

본고는 SC20의 프로그램 구성을 설명하고, 고성능컴퓨팅 기술 선도 및 관심 국가와 기관을 파악하고자 발표 내용을 통계적으로 정리하고, 기술의 최신 동향을 파악하고자 튜토리얼, 워크숍, 논문 세션에서 발표된 주요 HW와 SW 기술 및 응용 기술을 폭넓게 정리하며 결론을 맺는다.

II. SC20 개요

SC20은 예년과 같이 튜토리얼, 워크숍, 프로그램, 전시회 등 네 종류의 모임이 열렸다.

튜토리얼은 12개 2파트 코스와 18개의 단일파트 코스로 개설되었으며, 132명의 발표자와 53개 기관이 참여하였다. 워크숍은 24세션에 168편의 논문이 발표되었으며, 369개 기관에서 922명의 저자가 참여하였다. 프로그램은 33개 세션에서 103편의 논문이 발표되었으며, 124개 기관 604명 저자가 참여하였다. 전시회에는 285개 업체가 참여하였으며 칩 제조사, 컴퓨터 시스템업체, 서비스업체, 연

구기관, 표준화 단체, 대학에서 참여하였다.

전체 일정의 첫째 주(11월 9일~13일)에 튜토리얼과 워크숍이 진행되었으며, 둘째 주(11월 16일~19일)에 keynote, plenary, 논문 발표, 포스터 발표, 패널 토의, 기업체 발표가 진행되었다. 이처럼 2주에 걸쳐 슈퍼컴퓨터 HW, 시스템 SW, 응용 SW, 개발 및 운용 분야의 다양하고 새로운 기술들이 발표되었다. HW 분야에는 Exascale 슈퍼컴퓨터와 CPU, GPU, FPGA 가속기를 포함하는 이종(Heterogeneous) 시스템, 메모리 운영과 데이터 저장기술, 양자컴퓨팅, 통신 네트워크 및 연결망 기술이 발표되었으며, 시스템/응용 SW 분야에서는 OpenMP, MPI, 이종 프로그래밍, 메모리 프로그래밍, 런타임 기술이 발표되었고, 응용 SW 분야에서는 대규모, 긴급의사결정, 실시간 상황 분석 및 가시화를 위한 기술들이 발표되었다. 그리고 슈퍼컴퓨터의 설계와 운영 분야인 성능 최적화, 모델링, 고장감내, HPC 컨테이너 런타임, 클라우드기술 활용 및 전력관리 기술들이 발표되었다.

튜토리얼은 최근 관심을 끄는 분야 또는 새롭게 부상하는 기술을 선택하고 기술을 소개하여 그 기술의 개발 동향을 가늠하게 한다. 48개의 기술 분야를 발표된 42개 트랙에 대응시켰다. 기술 분야별로 정리하면 Software Engineering 8개 트랙, Scalable Computing, Portability, Parallel Programming Language, Libraries and Models 각 7개 트랙, Heterogeneous System 6개 트랙, Performance/Productivity, Measurement and Evaluation 각 5개 트랙, Requirement and Performance, OpenMP, MPI, Bigdata, Benchmark 각 4개 트랙, Reproducibility and Transparency, Network, Cloud and Distributed System 각 3개 트랙으로 나뉜다.

워크숍에서는 성능 모델링, 재구성 컴퓨터, 메모리 중심 컴퓨터, 병렬 저장장치, 구조 및 알고리

즘, 프로그래밍 모델, MPI, 병렬 분산 계산, 다중 병렬 계산, 런타임, OS, ML, DL, 빅데이터, 긴급 의사결정, 성능/이식성/생산성, 고장감내, 응용 지원 툴, 데이터 관리, HPC 교육 등 기반기술, 하드웨어, 소프트웨어, 응용프로그램, 활용의 여러 분야 내용을 발표하였다.

워크숍에서 발표한 논문 수는 168편이며, 논문 저자가 속한 기관이 있는 국가를 기준으로 발표 국가 및 지역을 분류한 내용은 다음과 같다. 한 편에 다수의 저자가 있어서 발표 논문 수보다 많은 국가가 있음을 알 수 있다. 미국 235명이며, 독일 36명, 영국 24명, 일본 12명, 프랑스 11명, 스위스 8명, 스페인과 네덜란드 6명, 중국 5명, 대한민국과 덴마크 3명 순이다. 유럽 전체는 105명이며, 아시아는 24명, 기타 4명이다. 기관이 속한 국가별로 분류하면 미국 78개 기관, 유럽 62개 기관, 중국 5개 기관, 일본 8개 기관이 분포하고 있다.

프로그램에서 발표된 논문의 분야는 모델링과 시뮬레이션, 양자 시뮬레이션, 스토리지, I/O, 네트워크, 통신, 전력제어 기술, 커널, 컴파일러, 포팅, 구조와 결합된 알고리즘, 그래프, 전통적인 물리, 화학, 선형대수 분야와 최근 부상한 AI, ML, DL, 데이터 분석 등이 있다. 프로그램에서 발표한 논문 수는 103편이며, 저자가 속한 기관이 있는 국가별로 분류하면 미국 386명, 중국 84명, 일본 52명, 스위스 18명, 스페인 13명, 독일 8명, 대한민국과 네덜란드 각 6명, 호주와 홍콩 5명, 영국과 덴마크와 폴란드 각 3명이며, 이들 외에 1명 이상인 국가는 8개국이다. 기관이 속한 국가별로 분류하면 미국 71개 기관, 유럽 16개 기관, 중국 13개 기관, 일본 13개 기관이 분포하고 있다.

국가별로 고성능컴퓨팅 기술을 연구개발하는 기관은 주로 국가지원 연구기관과 대학들이었으며, 미국의 경우에는 시스템 구현기술, 인공지능

응용 프로그램 개발, 시스템 성능 평가 기술, 시스템 서비스 기술을 기업체에서도 활발히 연구하고 있다.

III. 슈퍼컴퓨터 주요 기술

1. 이종(Heterogeneous) 시스템

슈퍼컴퓨터 프로세서는 증가하는 AI 학습 및 연산 기능과 빅데이터를 처리하기 위하여 기존의 병렬 연산을 구현하던 CPU core 외에, GPU를 포함하여 FPGA와 같은 전용가속기를 사용하는 구조를 채택하고 있다. 전용가속기는 병렬성, 고속 및 저전력 특성이 있어 특정 분야의 연산에 효율적인 시스템 특성을 구현한다. 시스템 구현에 있어 프로세서는 레지스터, 캐시, 메인 메모리, 스토리지 메모리와 결합하여 고속 병렬 연산기능을 구현하고 다중 사용자를 지원한다. 다중 프로세서, 메모리들은 서로 고속 인터페이스를 통하여 연결되어 병렬 분산 연산을 처리하여 시스템의 요구 규격인 연산 속도를 달성하게 된다.

가. FPGA 가속기

FPGA(Field-Programmable Gate Arrays) 구조를 채택한 고속 연산 가속기로 사용하고자 하는 시도는 FPGA 제조사에서 제공하는 설계 환경인 HLS(High Level Synthesis) 툴을 이용하여 C/C++ 설계에서 2차원 병렬 연산, 행렬 위치 교환, 상수 연산 부하감소를 하드웨어로 빠르게 구현하는 방법을 제공하고[1], BLAS를 HLS 환경에서 FPGA에 구현한 예가 발표되었다[2].

연산 커널 알고리즘을 Intel Zeon, Nvidia V100 GPU와 Intel Arria10, Xilinx U280 FPGA에서 루프라인 모델을 이용하여 비교하여 최고 성능을 보이는 GPU보다 속도가 떨어지나 HBM을 사용한 FPGA

에서 전력소모가 적음을 보였다[3].

CPU, GPU, FPGA 각각을 이용하여 행렬 곱을 수행할 때 연산속도와 소모 전력을 비교하여 성능 면에서 GPU가 가장 좋으나 전력면에서 FPGA가 가장 좋음을 보였다[4].

나. FPGA 구동 소프트웨어

FPGA에서 이진검색 알고리즘을 벡터화하고, OpenCL을 이용하여 그리드 검색을 구현하였으며, 기존보다 단축된 검색시간을 달성하였다[5]. MPI를 FPGA 설계틀인 HLS를 이용하여 FPGA 하드웨어 설계를 함으로써 기존의 CPU만으로 수행했던 경우와 비교해서, 2배 이상 빠르면서 전력소모는 90% 줄였다[6]. OpenCL 커널을 개발하여 FPGA에 다양하게 적용하였으며 FPGA와 메모리의 데이터 교환이 매우 중요함을 보여주었다[7]. FPGA 가속기를 사용하여 구현한 kernel이 기존의 CPU와 CPU+GPU보다 행렬 연산에서 더 좋은 성능과 에너지 효율을 보여주고 있다[8]. FPGA service toolkit을 사용하여 구현한 FPGA 가속기에서 ML을 적용하는 고에너지 물리 문제를 푸는 대규모 연산을 제안하였다[9]. OpenCL을 이용하여 다수의 FPGA를 사용한 HPC는 기존의 GPU 성능을 뛰어넘는 결과를 얻었다[10].

OpenACC를 사용하여 GPU와 함께 하는 연산 환경에서 CPU만 사용할 때와 비교하여 수십 배의 성능 향상을 얻을 수 있었다[11].

2. 메모리와 스토리지

컴퓨터 시스템에서 메모리는 프로세서와 가속기와 연결되고 접근 속도 및 용량에 따라 다양한 계층을 이룬다. SRAM, DRAM, HBM, NVM, SSD, HDD로 구성된 메모리와 스토리지는 대역

폭, 접근속도, 용량에 따라 캐시, 메인 메모리, 데이터 및 파일 저장에 사용된다. 프로세서의 연산 능력과 메모리의 저장 데이터의 읽고 쓰기 속도는 전체 컴퓨팅시스템에 성능을 결정하게 된다. 프로세서와 메모리 속도 불일치를 최소화하는 방식으로 메모리를 배치하며 대개 메모리 용량이 커짐에 따라 메모리 속도가 느려지므로 메모리 크기를 변화시키며 계층적으로 프로세서와 결합한다. HPC의 경우 노드 사이의 데이터 교환이 발생하므로 원격 메모리와 데이터 이동 역시 중요한 성능 요소이다.

가. 메모리 중심 컴퓨팅 시스템

전력 소모 증가와 동작 속도의 한계 때문에 나노 수준의 반도체 제조는 성능 향상지표인 무어의 법칙을 만족하지 못하지만, 비 폰노이만 방식 구조 및 프로그래밍과 런타임은 이러한 성능 제약을 뛰어넘게 한다. 데이터 전송량 증가, 메모리 대역폭 증가, 메모리 지연 감소를 가진 메모리 중심 HPC 가속기는 성능 향상을 가져온다[12].

데이터 집중 응용분야의 성능 향상에 메모리 접근과 데이터 이동이 매우 중요한 요소이며, 데이터 스트리밍을 처리하는 새로운 구조 혁신이 필요하며 쓰레드 이동 개념을 제안하고 있다[13].

많은 고성능 시스템은 여러 가지 형태의 메모리 장치를 가지고 있으며, 고사양의 저용량 메모리와 저사양의 고용량 메모리를 사용한다. 여러 메모리를 효율적으로 사용하고자 운영체제는 응용 프로그램에 따라 HW 관리 메모리, SW 관리 메모리를 혼합하여 데이터를 관리한다. HBM을 사용한 혼합 데이터 관리모드는 단일 메모리 관리를 사용한 경우보다 향상된 성능을 보인다[14].

나. 이종 메모리

매니코어 프로세서와 가속기를 고성능 패키징

기술과 혁신적 메모리들로 만든 컴퓨터 노드는 효율적인 메모리 데이터 관리를 통하여 시스템 하드웨어 성능을 높여야 한다. 메모리 접근 모니터링과 메모리인지 데이터 구조(Memory-aware Data Structure)를 사용하여 성능을 향상할 수 있음을 보여준다[15].

이중 시스템에서 캐시와 메모리 사이의 메모리 접근이 프로세서별로 서로 달라서 메모리 접근 패턴을 기반으로 캐시 프리페칭(Prefetching)을 예측하여 성능 향상을 가져올 수 있다[16].

RDMA를 사용하는 분산메모리 시스템에서 응용 프로그램의 생산성을 향상하고자 개선된 분산공유메모리 라이브러리(MENPS)를 사용하여 OpenMP 프로그램을 수행하였다[17].

이중 메모리 디바이스를 사용하여 데이터 중심 응용을 처리하는 시스템을 구성할 때 메모리 데이터 배치, 이동, 접근, 변환에 적합한 시스템 구성과 구현이 필요하다[18].

FPGA 가속기가 분산 메모리에 접근할 때 호스트를 거치거나 FPGA에 직접 연결하면 성능이 떨어지거나 구성의 유연성이 떨어진다. Streaming Message Interface를 FPGA에 구현하고 API로 제어하여 FPGA와 메모리 사이의 데이터 전달을 효율적으로 하는 구조를 제안하였다[19].

3. 네트워크 및 인터커넥션

슈퍼컴퓨터는 하나의 기판 위에 여러 CPU, GPU, 가속기와 이들이 사용하는 메인 메모리, 스토리지를 탑재한 노드를 구성하고, 이 노드를 다수 연결한 클러스터로 구성된 시스템이다. 한 노드 위에 있는 CPU, GPU, 가속기는 많은 데이터를 교환하기 때문에 고속의 연결망을 이용하여 서로 연결한다. 노드와 노드의 연결 역시 분산 작업 및 여러

작업을 동시에 처리하기 때문에 고속의 데이터 전달이 가능한 네트워크 연결이 필요하다.

가. HPC 인터커넥션 기술

HPC TOP500 시스템에 사용되는 고속 네트워크 전송기술로 2007년 이후 고속 Ethernet(HSE), InfiniBand(IB) 기술이 80% 정도 사용되고 있으며, 2010년 이후 자체 전송 프로토콜이 10% 가까이 점유하고 있다[20,21].

시스템 영역의 네트워크는 저지연(Low Latency), 광대역(High Bandwidth) 낮은 CPU 사용률(Low CPU Utilization) 조건을 만족하고 스토리지 영역은 고성능 I/O를 만족해야 한다. 데이터 전송 QoS를 보장하고 계산 클러스터, 저장 클러스터에 LAN/SAN 연결이 대규모 기업형 데이터 센터 연결에 WAN/LAN/SAN이 사용된다. 기존의 I/O 인터페이스 기술인 PCIe, HT(HyperTransport)와 새롭게 등장하는 기술인 CCIX(Cache Coherent Interconnect for Accelerators), NVLink, CAPI/OpenCAPI, Gen-Z 등이 있다[20]. 2020년 6월 발표된 Top500 자료에 따르면 HSE 채용시스템이 51%로 가장 많으며, IB가 30%로 그 뒤를 따르고 있으며, Omni-Path가 10%를 점유하고 있다. 성능을 고려한 점유율에서 IB가 37%로 가장 많으며, HSE가 21%, 표준을 변형한 네트워크(Proprietary Network)가 17%이다[20,21]. 전력효율 시스템 순위를 정한 Green 500의 상위 10개 시스템 중에서 NVIDIA GPU와 네트워크 기술은 IB가 70%를 점하고 있다[22].

나. RDMA와 인터커넥션

IB, HSE를 사용하여 RDMA(Remote Direct Memory Access)를 수행하며 DMA 전송 전에 메모리 등록 및 메모리 보호 기능을 수행하여 DMA 데이터를 보호한다. HSE에서는 iWRAP(internet

Wide-area RDMA Protocol)를 사용하여 DMA를 수행하고, HSE와 IB 사이의 DMA는 RoCE(RDMA over Convergence Enhanced Ethernet)를 사용한다[20].

다. CXL

CXL(Compute Express Link)와 Gen-Z는 프로토콜의 짧은 지연과 프로세서, 가속기의 캐시, 메인 메모리들과 직접 연결을 가능하게 한다. CXL은 노드 수준에서 프로세서와 가속기가 캐시와 메모리 데이터를 동기화하여 읽고 쓸 수 있게 한다. 반면 Gen-Z는 패브릭 기반의 구조로 랙(Rack) 스케일 연결을 제공한다. CXL과 Gen-Z는 두 프로토콜을 상호 연결할 수 있는 기술을 공동 개발하기 위한 협의회를 구성하였다[23]. HPC 노드에 있는 여러 개의 CPU 칩들이 작업을 분할하여 병렬 수행할 때 CXL을 활용하여 칩의 병렬 작업 속도를 높일 수 있어 HPC에 활용할 것을 제안하고 있다[24].

4. 인공지능/머신러닝/딥러닝

인공지능(AI)은 인간의 인지능력을 지원하거나 강화해 주고 인간을 대신하여 자동으로 일을 수행, 제어하기 위하여 개발, 사용되는 컴퓨팅(Computational) 기술이다[25]. 머신러닝(ML)/딥러닝(DL)은 데이터를 기반으로 신경망 훈련을 통해서 인간의 지식을 자동 학습하게 되며, 데이터의 학습량이 증가하고 심층 신경망의 레벨이 깊어질수록 예측 정확도가 증가하게 되는 특징이 있다.

머신러닝/딥러닝은 초기 거대 규모의 복잡한 시뮬레이션이 필요한 기초 과학 분야에서 주로 적용되었고, 그 가능성의 가시화로 거의 모든 분야에서 도입이 증가하고 있다. 최근 COVID-19와 같은 자연재해는 디지털 기술혁명을 가속화하고, 이로 인한 인공지능, 고성능컴퓨팅의 활용 분야도 폭

발적으로 넓어지고 확대되고 있다. SC20에서도 HPC+AI 기술이 활발히 논의되었으며, 본 고에서는 AI 등장으로 드러난 HPC 기술 이슈 및 융합 사례 등을 정리한다.

가. HPC for AI

머신러닝과 딥러닝의 워크로드가 엄청난 속도로 증가하면서 컴퓨팅 자원 요구량도 급격하게 늘어나고 있다. 이를 수용하기 위하여 최근 고성능컴퓨팅 분야에서는 다양한 전용 가속기(GPU, IPU[26], NPU, DPU, FPGA)나 AI 전용 반도체를 사용해서 컴퓨팅 처리 속도를 가속화하는 경향이 강해지고 있다. 다양한 종류와 다양한 구성의 하드웨어/응용들이 단기간에 폭발적으로 증가하면서, 다양한 종류의 기술적 이슈들도 같이 등장하고 있다[27]. 첫째, 가속기 연산의 고속화에 상대적으로 증가되는 통신비용의 문제가 있으며, 노트르담 대학에서는 확률적 경사 하강법(SGD)에서 이벤트 트리거 기반의 통신 기법(EventGraD)을 통해 분산 학습의 동일 수준의 정확도를 유지하면서 통신비용을 70% 감소했다는 연구 성과를 발표했다[28]. 둘째, 대규모 데이터와 복잡한 워크로드 등장과 이종 가속기들의 자원 한계로 인해 딥러닝 분산 학습의 효율성 이슈에 관하여, 암스테르담 대학에서는 다양한 모델, 데이터 세트 및 하드웨어 시스템에서 대규모 분산 병렬 처리 모델 성능을 심층 비교 분석할 수 있는 벤치마크(DDL Bench)를 제안했다[29].

나. HPC+AI 융합

SC20 Plenary에서 HPC 커뮤니티는 COVID-19 대응을 위한 국제 공조 활동을 공유했다. 코로나 바이러스의 확산, 전파 경로의 시뮬레이션, 백신 및 치료제 개발을 위한 데이터의 수집, 처리, 분석,

설계, 모델링 등 업무의 모든 단계에서 인공지능 기술이 적용되었고, 기존 HPC에서 몇 년 걸릴 연산을 고성능컴퓨팅과 AI의 융합을 통해 처리시간을 단축했다고 설명했다. HPC와 AI의 융합은 선택이 아니라 필수가 되었고, 그 활용과 확산이 가속화될 것이라는 전망을 갖게 한다. 동시에 해결되어야 할 기술적 이슈들도 남겼다. 보건/의료와 같은 중요 의사 결정을 목표로 하는 알고리즘의 데이터 투명성[30]과 모델의 신뢰성이 더욱 중요하게 대두될 것이며, 인프라로 활용되는 HPC 시스템의 무결성, 즉 결과의 재현 가능성, 결과를 설명할 수 있는 검증 기술, 초대규모 모델 등장으로 글로벌 협력과 자료공유를 위한 기술, 즉 타인에 의해 검증된 모델이나 데이터를 이어받아서 재시작할 수 있는 기술[31], 표준화된 딥러닝 데이터의 표현 기술, 궁극적으로, 실제 인간의 두뇌와 같이 적은 데이터로 학습, 추론할 수 있는 새로운 혁명적 모델 제시를 위한 연구가 진행되어야 할 것이라고 토의했다[32].

5. 병렬 프로그래밍 모델

병렬 프로그래밍 모델이란 여러 개의 프로그램 또는 한 프로그램의 분할된 부분들이 다수의 프로세서들에서 동시에 처리되게 구현하는 프로그래밍 방식을 의미한다. HPC의 대표적인 병렬 프로그래밍 모델로는 MPI, OpenMP, PGAS 등이 있다. MPI(Message Passing Interface)는 프로세스 간 메시지를 주고받는 방식의 병렬 프로그래밍 모델로서 단일 노드 내와 노드 간 병렬 처리를 지원하기 때문에 HPC 분야에서 가장 널리 이용되고 있다. OpenMP(Open Multi-Processing)는 공유 메모리 기반의 병렬 프로그래밍 모델로서 프로그래머의 directive에 따라 병렬화를 수행하는 방식으로 메모리

를 공유하는 노드 내의 병렬 처리만을 대상으로 한다. PGAS(Partitioned Global Address Space)는 분산 공유 메모리 기반의 병렬 프로그래밍 모델로서 다수의 노드 간 병렬 처리를 지원한다. SC20에서도 다양한 병렬프로그래밍 모델의 발전방향을 파악할 수 있었으며 본 고에서는 MPI에 집중하여 정리하였다.

가. MPI + X 기술

대규모 컴퓨팅 노드에서 병렬 처리를 위해서 노드 간 병렬 처리용으로 MPI, 그리고 노드 내에 이중 가속기용 병렬 프로그래밍 모델인 OpenCL, OpenACC, OpenMP 5.0 등이 함께 이용되는 하이브리드 모델(MPI+X)이 대세이다.

미국 샌디아 국립연구소에서는 MPI와의 통합 및 상호 운용성을 제공하는 비동기 다중 작업 런타임 시스템인 가상 전송(VT) 실행 모델을 제시하였다. 가상 전송(VT) 실행 모델을 통해 동기 MPI 작업들과 비동기 작업들을 조화롭게 실행할 수 있음을 보임으로써 하이브리드 병렬화의 MPI+X 모델이 MPI+VT+X로 발전하는 것을 목표로 한다고 설명했다[33].

미국 오하이오 주립대(DK Panda 교수 연구팀)에서는 GPU 기반의 기계학습 라이브러리인 cuML을 가속하는 MPI 최적화 라이브러리(MVAPICH2-GDR)를 발표하며, MPI 통신 최적화를 위한 Python API를 제안하고 K-Means, Nearest Neighbors, Random Forest 등에 적용하여 속도가 향상된 결과를 발표하였다[34].

PAW-ATM(Alternatives To MPI+X) 워크숍은 MPI+X의 대안을 찾고자 구체적인 사례를 제시하고 설명하는 워크숍이었다[35]. 현재 대부분의 HPC 응용들은 MPI+X와 같은 하이브리드 프로그래밍 모델을 사용하고 있으나 이들이 점차 복잡

해지고 있어 발생하는 문제들을 해결하고자 새로운 프로그래밍 언어(예: Chapel, Regent, XcalableMP)나 대규모 데이터 과학을 위한 프레임워크(예: Arkouda, Dask) 또는 기존 언어 확장(예: Charm++, COMPSs, Fortran, Legion, UPC++) 등의 대안을 제시하였다.

미국 샌디아 국립연구소에서는 쓰레딩 모델과 MPI와의 상호작용에 초점을 맞춘 OpenMPI에서 다중 쓰레드를 구현 기술과 인터페이스를 소개하였다. 이 아키텍처는 쓰레딩 기본요소의 정적 초기화와 쓰레딩 개체의 동적 인스턴스화를 모두 제공한다[36].

스위스 ETH Zurich(Torsten Hoefler 교수 연구팀)에서는 키노트 연설을 통해, 최근 데이터 센터 및 HPC 클러스터에서 널리 가속기로 이용되고 있는 FPGA(Field-Programmable Gate Arrays) 간의 분산 처리를 위한 통신 모델인 스트리밍 메시지를 소개하였다. 특히 스트리밍 메시지의 의미를 파악할 수 있도록 설계된 통신 인터페이스 사양인 SMI(Streaming Message Interface)를 이용하여 Intel FPGA에서 OpenCL과 통합되는 사례를 보여주었다. 이로써 프로그래머는 SMI를 사용하여 FPGA에 분산되고 확장 가능한 HPC 프로그램을 구현할 수 있음을 보여주었다[19].

나. 엑사스케일을 향한 MPI 기술

Exascale MPI 워크숍은 MPI 프로그래밍 모델에서 혁신적인 알고리즘과 개념을 발표하고 엑사스케일 시대 MPI의 미래에 대해 논의하였다[37]. MPI 집합 명령어를 위한 혁신적인 알고리즘, 데이터 중심 모델을 포함한 MPI 확장, 네트워크 혼잡을 방지하기 위한 스케줄링/라우팅, 내결함성 통신, MPI 및 PGAS 모델의 상호 운용성, MPI에서 작업 병렬 모델 통합 및 대규모 시뮬레이션 시 MPI

사용법 등에 관한 것이다.

로렌스 리버모어 국립연구소에서는, HPC 응용 처리 성능을 높이기 위해 채택된 다수의 GPU들로 인해 발생하는 통신 오버헤드를 MPI기술들이 어떻게 다루고 있는지 성능을 측정하기 위해 기존 통신 벤치마크 프로그램을 수정하여 OpenMPI, Spectrum MPI, MVAPICH의 성능을 측정하고 시스템을 튜닝한 사례를 발표했다[38].

미국 오하이오 주립대(DK Panda 교수 연구팀)에서는 MPI 집합 통신 처리 시, 통신을 네트워크 단으로 오프로드하여 CPU의 개입을 줄인 InfiniBand 스위치 SHARP(Scalable Hierarchical Aggregation and Reduction Protocol)를 이용하여 MPI 라이브러리인 MVAPICH2-X를 최적화한 사례를 발표하였다. 최적화 라이브러리를 Frontera 슈퍼컴퓨터에 적용하여 실험한 결과, 7,861개 노드에서 Reduce의 경우 대기 시간이 최대 5.4배, Allreduce의 경우 5.1배, Barrier의 경우 7.1배 감소한 성과를 제시하였다[39].

6. HPC 클라우드

전통적으로 HPC 분야에서는 고성능의 컴퓨팅 자원을 극한 계산 응용 프로그램에 전적으로 할당하여 다른 사용자나 다른 프로그램과 공유하지 않은 채로 고속으로 처리할 수 있게 해 왔다. 이때 이용자에게는 고성능 컴퓨터 HW에 최소한의 SW 개발 환경(운영체제, 네트워크)만을 제공하는 베어메탈 프로비저닝 방식으로 자원을 제공했으며, 그 외 필요한 SW 환경은 이용자(즉, SW개발자)가 직접 꾸려야 했다. 최근 들어 빅데이터, 인공지능 등 여러 분야에서 HPC의 수요가 늘자 효율적 자원공유 및 관리, 이용의 편의성, 워크플로우 자동화 등에 대한 요구가 강력히 대두되며 클라우드 기술이 HPC 커뮤니티 내에서 동력을 얻게 되었다.

SC20에서도 HPC를 위한 클라우드 기술이 활발히 논의되었으며, 본 고에서는 HPC 클라우드의 핵심 기술인 HPC 컨테이너 런타임 기술, HPC 클라우드 구축 사례, HPC 클라우드 구축과 서비스 사례 등을 정리한다.

가. HPC 컨테이너 런타임

HPC 컨테이너 런타임 기술은 고성능 컴퓨터의 성능과 보안성을 유지하며 HPC 워크로드를 처리할 수 있게 개발된 컨테이너 기반 가상화 기술이다. SC20 튜토리얼을 통해 오픈소스 HPC 컨테이너 런타임들인 Charliecloud, Singularity, Shifter, Podman들이 소개되었다[40].

Charliecloud는 HPC 환경에서 사용자 정의 소프트웨어 스택(UDSS)을 기반으로 하는 오픈소스 프로젝트로 Linux 사용자 네임 스페이스를 사용하여 권한이 없는 컨테이너를 실행하며 한다. 특정 사용 사례와 충돌하지 않도록 최소화하고 경량으로 설계되었다.

Shifter는 미국 국립에너지연구과학컴퓨팅센터(National Energy Research Scientific Computing Center)에서 750개 프로젝트에 6천여 명의 이용자를 대상으로 HPC상에서 대규모 데이터를 처리할 때 겪는 어려움을 조사한 결과로 컨테이너의 필요성을 입증하고, 도커보다 보안성과 성능을 강화하여 HPC의 특성에 맞도록 개발한 컨테이너 런타임이다. 클러스터 관리와 작업 스케줄링에 Slurm을 사용하고, MPICH 중심으로 MPI를 지원한다.

Singularity는 Shifter 공동 작업자가 분기하여 시작된 오픈소스이며, 병렬 프로그래밍 모델인 MPI, 고속 네트워크 기술인 InfiniBand, 병렬 분산 파일 시스템인 Lustre를 지원하고 작업관리를 위해 Slurm 플러그인을 지원한다. TACC, 샌디에이고 슈퍼컴퓨터 센터와 오크리지 국립연구소를 포함

하여 약 25,000개 이상 사이트에서 이용하고 있으며, HPC 컨테이너 런타임 중 개발자와 커뮤니티 활동이 가장 활발한 편이다.

Podman은 2017년부터 도커에서 파생된 코드로 Red Hat에서 개발한 컨테이너 런타임으로서 HPC 사용을 고려하여 설계되지 않아 HPC 지원에는 부족한 면이 있지만, 도커보다 경량으로 컨테이너를 실행하는 경량 래퍼(Wrapper)가 될 것으로 기대된다.

이외에 캘리포니아 대학에서는 컨테이너 네이티브 워크플로우 엔진인 Popper를 발표하며 컨테이너 엔진 및 리소스 관리자의 복잡성을 추상화하여 사용자가 워크플로우 작성에만 집중할 수 있도록 하는 방법으로 다양한 HPC 컴퓨팅 환경에서 쉽게 구축하고 검증할 수 있다고 설명했다[41].

나. HPC 클라우드 구축 및 서비스 사례

애리조나 대학에서는 OpenStack 클라우드 인프라에서 컴퓨팅 워크로드를 관리하기 위한 Exosphere를 소개하며 거대과학 연구에 클라우드 환경을 쉽게 이용할 수 있는 환경 제공 사례를 소개했다[42].

BSC(스페인), CEA(프랑스), CINECA(이탈리아), CSCS(스위스) 및 JSC(독일)의 5개 유럽 슈퍼컴퓨팅 센터는 컴퓨팅, 스토리지 서비스를 상호 연합하여 휴먼 브레인 프로젝트를 지원하는 Fenix라는 인프라를 구축하기로 했다는 소식을 발표했다[43].

스위스 국립 슈퍼컴퓨팅센터에서는 HPC컴퓨팅 센터에서도 과학 분야를 위한 디지털 인터페이스를 제공하고자 웹기반의 인터페이스인 FirecREST를 제공한 사례를 소개하였다. FirecREST는 HPC 컴퓨팅 자원을 보다 손쉽게 이용하기 위한 방법으로 워크플로우 자동화와 통합 리소스 및 서비스 접근 기능을 제공하는 RESTful API이다[44].

다. HPC 클라우드를 통한 대규모 응용 사례

텍사스 어드밴스트 컴퓨팅센터에서는 페타급 Frontera 시스템에서 세 종류의 컨테이너 런타임 (Singularity, Charliecloud, Podman) 기술을 실행한 경험을 발표하였다. 컨테이너 런타임 환경에서 양자 색역학 코드, 과학응용 MILC(MIMD Lattice Computation)를 1,296개 노드에서 70,000개의 프로세스로 실행하여 네이티브에 가까운 성능과 최소한의 메모리 오버헤드를 보였다고 설명했다[45].

오크리지 국립 연구소에서는 Summit 슈퍼컴퓨터에서 초거대과학계산 응용사례인 SKA(Square Kilometer Array) 데이터를 6시간 관찰하고 시뮬레이션한 다음 이미징 파이프라인으로 해당 데이터 세트를 처리한 사례를 설명했다. SKA 응용 프로그램은 4,560개의 컴퓨팅 노드에서 배포 및 실행되었으며, 27,360개의 GPU를 사용하여 2.6PB의 데이터를 생성했다. 전파 천문 데이터가 이 규모로 처리된 것은 이번이 처음이라고 하며, 향후 차세대 전파 망원경과 이에 필요한 전용 처리 시설의 설계 요소를 파악하는 데 기여할 것으로 기대했다[46].

라. HPC 클라우드 사업자 사례

HPC 클라우드 서비스를 제공하는 기업으로 Amazon, Oracle, Suse에서 자사의 서비스를 소개하며 기존의 HPC 컴퓨팅 환경을 클라우드 환경으로 이전하기 위한 문제점, 고려사항, 전략 등을 제시했다. 이중 Amazon은 AWS에서 제공하고 있는 HPC Cloud 사례를 발표하고, HPC 워크로드를 roofline model로 분석하여 클라우드에서 응용을 최적화시킬 수 있는 자원 요구량을 분석함으로써 오케스트레이션, 오토 스케일링, 서버리스 컴퓨팅 활용 사례를 소개하였다[47]. Red Hat은 부

스 발표를 통해, DOE(Department of Energy)의 엑사 스케일 컴퓨팅 프로젝트의 일부인 Supercomputing Containers 프로젝트를 통해 DOE 연구소에서 사용 중인 컨테이너 기술을 비교 평가한 결과, 오픈소스를 기반으로 표준을 지원하며 여러 하드웨어 아키텍처에서 실행할 수 있는 보다 강력한 품질의 컨테이너 솔루션의 필요성을 강조했다[48].

IV. 결론

본 고에서는 2020년 11월 개최된 SC20의 튜토리얼, 워크숍 등에서 발표된 내용 중 고성능컴퓨팅 HW 기술로 이중 프로세서, 메모리와 스토리지, 연결망 기술과 SW 기술로 병렬 프로그래밍 모델, 응용기술로 AI와 클라우드 기술이 접목된 사례들을 정리하였다. 고성능컴퓨팅 HW 기술은 프로세서, 메모리, 스토리지 등 이중 시스템들이 더욱 분화되고 계층화되는 추세이며, 병렬 프로그래밍 모델은 MPI+X가 대세이나 이의 대안들이 논의되고 있었고, 응용기술로는 특히 대규모 데이터에 대한 집중적 계산이 필요한 AI, 빅데이터 분야에서 확산 추세였으며, 클라우드 기술이 고성능 컴퓨터 응용 개발자들 사이에서 필수 편의 서비스로 자리 잡고 가고 있었다.

SC20의 발표를 전체적으로 살펴봄으로써 고성능컴퓨팅 기술의 최신 동향과 엑사스케일 규모로 발전하는 과정에서 대학, 연구소들의 연구 동향과 고성능컴퓨팅 기술을 다양한 응용/기술 분야에 적용하는 일선 기업들의 노력을 확인할 수 있었다. 마지막으로, 고성능컴퓨팅 기술 전문가뿐만 아니라 입문자들에게도 관심 분야를 찾는 데 도움이 되기를 바라며 국내의 지속적인 관심이 이어지기를 기대한다.

용어해설

High Performance Computing 고사양 HW를 최대한 활용하여 대규모 계산을 빠르게 수행하는 컴퓨팅 기술

Heterogeneous System 프로세서, 그래픽 프로세서, 가속기 등의 서로 다른 연산기 구조로 구성된 이종 시스템

Heterogeneous Memory Register, SRAM, DRAM, HBM, Persistent Memory, Flash memory 등 구조와 구성을 다르게 한 다양한 제작 기술로 만든 여러 형태의 메모리 또는 스토리지

약어 정리

AI	Artificial Intelligence
CPU	Central Processing Unit
DL	Deep Learning
DPU	Data Processing Unit
DRAM	Dynamic Random Access Memory
FPGA	Field Programmable Gate Array
GPU	Graphic Processor Unit
HBM	High Bandwidth Memory
HDD	Hard Disk Drive
HW	Hardware
IPU	Intelligence Processing Unit
ML	Machine Learning
MPI	Message Passing Interface
NPU	Neural Processing Unit
NVM	Non-Volatile Memory
OS	Operating System
SGD	Stochastic Gradient Descent
SRAM	Static Random Access Memory
SSD	Solid-State Drive
SW	Software

참고문헌

- [1] Johannes de Fine Licht and Torsten Hoefler, "Productive Parallel Programming for FPGA with High-Level Synthesis," Tutorial, SC20, Nov. 2020.
- [2] Tiziano De Matteis et al., "FBLAS: Streaming Linear Algebra on

FPGA," SC20, Nov. 2020.

- [3] Tan Nguyen et al., "The Performance and Energy Efficiency Potential of FPGAs in Scientific Computing," 11th International Workshop on Performance Modelling, Benchmarking and Simulation of High Performance Computer Systems(PMB), Nov. 2020.
- [4] Nick Brown, "Exploring the Acceleration of Nekbone on Reconfigurable Architectures," 6th International Workshop on Heterogeneous High-Performance Reconfigurable Computing(H2RC), Nov. 2020.
- [5] Zheming Jin and Hal Finkel, "Performance Evaluation of the Vectorizable Binary Search Algorithms on an FPGA Platform," 10th Workshop on Irregular Applications: Architectures and Algorithms(IAAA), Nov. 2020.
- [6] Burkhard Ringlein et al., "Programming Reconfigurable Heterogeneous Computing Clusters Using MPI with Transpilation," 6th International Workshop on Heterogeneous High-Performance Reconfigurable Computing(H2RC), Nov. 2020.
- [7] Marius Meyer et al., "Evaluating FPGA Accelerator Performance with a Parameterized OpenCL Adaptation of Selected Benchmarks of the HPC Challenge Benchmark Suite," 6th International Workshop on Heterogeneous High-Performance Reconfigurable Computing(H2RC), Nov. 2020.
- [8] Ryan L. Blanchard et al., "FPGA Acceleration of Fluid-Flow Kernels," 6th International Workshop on Heterogeneous High-Performance Reconfigurable Computing(H2RC), Nov. 2020.
- [9] Dylan Rankin et al., "FPGAs-as-a-Service Toolkit(FaaST)," 6th International Workshop on Heterogeneous High-Performance Reconfigurable Computing(H2RC), Nov. 2020.
- [10] Narihisa Fujita et al., "OpenCL-enabled Parallel Raytracing for Astrophysical Application on Multiple FPGAs with Optical Links," 6th International Workshop on Heterogeneous High-Performance Reconfigurable Computing(H2RC), Nov. 2020.
- [11] Xiaomeng Zhai et al., "GPU acceleration of the FINE/FR CFD solver in a heterogeneous environment with OpenACC directives," 7th Workshop on Accelerator Programming Using Directives(WACCPD), Nov. 2020.
- [12] Thomas Sterling, "Invited Talk: A Parallel Execution Model for Exascale non von Neumann Memory-Centric Architectures," 5th International Workshop on Extreme Scale Programming Models and Middleware(ESPM2), Nov. 2020.
- [13] Peter Kogge, "Keynote: 3rd Wall and the Need for Innovation in Architectures," Workshop on Memory Centric High-Performance Computing(MCHPC), Nov. 2020.
- [14] T. Chad Effler et al., "Performance Potential of Mixed Data Management Modes for Heterogeneous Memory Systems," Workshop on Memory Centric High-Performance Computing(MCHPC), Nov. 2020.
- [15] Nuwan Jayasena, "Keynote: Memory Performance Optimization," 10th Workshop on Irregular Applications:

- Architectures and Algorithms(IAAA), Nov. 2020.
- [16] Mohammad Alaul Haque Monil et al., "Understanding the Impact of Memory Access Patterns in Intel Processors," Workshop on Memory Centric High-Performance Computing(MCHPC), Nov. 2020.
- [17] Wataru Endo, "MENPS: A Decentralized Distributed Shared Memory Exploiting RDMA," 4th International Workshop on Emerging Parallel and Distributed Runtime Systems and Middleware, Nov. 2020.
- [18] Ada Gavrilovska, "Data-Centric Resource Management for Complex Memory Fabrics," 10th International Workshop on Runtime and Operating Systems for Supercomputers, Nov. 2020.
- [19] Tiziano De Matteis, "ExaMPI-Keynote: Streaming Messages-A Distributed Memory Programming Model for Reconfigurable Hardware" 1Workshop on Exascale MPI, Nov. 2020.
- [20] Dhableswar K. Panda, "InfiniBand, High-speed Ethernet, RoCE, Omni-Path, EFA and Slingshot for Beginners," Tutorial, SC20, Nov. 2020.
- [21] sc20 top500 topics
- [22] Wu Feng et al., "The Green 500: Trends in Energy-Efficient Supercomputing," SC20, Nov. 2020.
- [23] Thomas Coughlin et al., "Compute Express Link™(CXL™) and Gen-Z Protocols: Bridging the Resource Gap Between CPU, Memory and Accelerators," Tutorial, SC20, Nov. 2020.
- [24] <https://www.computeexpresslink.org/about-cxl>
- [25] Steve Lee, "Keynote: Scientific Machine Learning and Artificial Intelligence, DoE, Office of Science," Workshop on Artificial Intelligence and Machine Learning for Scientific Applications(AI4S), Nov. 2020.
- [26] Simon Knowles, "Invited Talk: The Re-invention of Accelerated Computing for AI," SC20, Nov. 2020.
- [27] Cliff Young, "Invited Talk: Can We Build a Virtuous Cycle Between Machine Learning and HPC?," SC20, Nov. 2020.
- [28] Soumyadip Ghosh and Vijay Gupta, "EventGraD: Event-Triggered Communication in Parallel Stochastic Gradient Descent," Workshop on Machine Learning in High Performance Computing Environments(MLHPC), Nov. 2020.
- [29] Matthijs Jansen et al., "DDL Bench: Towards a Scalable Benchmarking Infrastructure for Distributed Deep Learning" 5th Deep Learning on Supercomputers Workshop(DLS), Nov. 2020.
- [30] Yuxin Yang et al., "How Good is Your Scientific Data Generative Model?," Workshop on Artificial Intelligence and Machine Learning for Scientific Applications(AI4S), Nov. 2020.
- [31] Romain Lion and Samuel Thibault, "From Tasks Graphs to Asynchronous Distributed Checkpointing with Local Restart," Workshop on Fault-Tolerance for HPC at Extreme Scale(FTXS), Nov. 2020.
- [32] Lorena A. Barba, "Invited Talk: Trustworthy Computational Evidence Through Transparency and Reproducibility," Responsible Application of HPC, SC20, Nov. 2020.
- [33] Jonathan Lifflander et al., "Design and Implementation Techniques for an MPI-Oriented AMT Runtime," Workshop on Exascale MPI(ExaMPI), Nov. 2020.
- [34] Seyedeh Mahdiah Ghazimirsaeed et al., "Accelerating GPU-based Machine Learning in Python using MPI Library: A Case Study with MVAICH2-GDR," 6th Workshop on Machine Learning in HPC Environments, Nov. 2020.
- [35] <https://sourceryinstitute.github.io/PAW/>
- [36] Noah Evans et al., "Implementing Flexible Threading Support in Open MPI," Workshop on Exascale MPI(ExaMPI), Nov. 2020.
- [37] <https://sc20.supercomputing.org/session/?sess=sess220>
- [38] Nathan Hanford et al., "Challenges of GPU-aware Communication in MPI," Workshop on Exascale MPI(ExaMPI), Nov. 2020.
- [39] Bharath Ramesh et al., "Scalable MPI Collectives using SHARP: Large Scale Performance Evaluation on the TACC Frontera System," Workshop on Exascale MPI(ExaMPI), Nov. 2020.
- [40] Richard Shane Canon et al., "Using Containers to Accelerate HPC," Tutorial, SC20, Nov. 2020.
- [41] Jayjeet Chakraborty et al., "Enabling Seamless Execution of Computational and Data Science Workflows on HPC and Cloud with the Popper Container-Native Automation Engine," 2nd International Workshop on Containers and New Orchestration Paradigms for Isolated Environments in HPC(CANOPIE-HPC), Nov. 2020.
- [42] Julian L. Pistorius et al., "Exosphere-Bringing The Cloud Closer," 3rd International Workshop on Interoperability of Supercomputing and Cloud Technologies(SuperCompCloud), Nov. 2020.
- [43] Sadaf R. Alam et al., "Archival Data Repository Services to Enable HPC and Cloud Workflows in a Federated Research e-Infrastructure," 3rd International Workshop on Interoperability of Supercomputing and Cloud Technologies(SuperCompCloud), Nov. 2020.
- [44] Felipe A. Cruz et al., "FirecREST: a RESTful API to HPC Systems," 3rd International Workshop on Interoperability of Supercomputing and Cloud Technologies(SuperCompCloud), Nov. 2020.
- [45] Amit Ruhela et al., "Containerization on Petascale HPC Clusters," SC20, Nov. 2020.
- [46] Ruonan Wang et al., "Processing Full-Scale Square Kilometre Array Data on the Summit Supercomputer," SC20, Nov. 2020.
- [47] Pierre-Yves Aquilant et al., "Best Practices for HPC in the Cloud: Part 2(Amazon)," Tutorial, SC20, Nov. 2020.
- [48] Joshua Hursey, "Design Considerations for Building and Running Containerized MPI Applications," 2nd International Workshop on Containers and New Orchestration Paradigms for Isolated Environments in HPC(CANOPIE-HPC), Nov. 2020.