

ScienceDMZ 기반 인공지능 연구를 위한 분산 컴퓨팅 연계 방안 연구

김기현*, 문정훈°, 권우창*, 박병연*, 석우진*, 홍원택*, 이상권*, 조진용*,
윤태진**, 최재현**, 김대겸***, 황동아***, 최선웅***, 김종호****,
김준엽, 김기형, 정병훈, 이도섭, 유영근, 박은숙, 천재홍

Connecting Method Research of Distributed Computing for AI Research Based on ScienceDMZ

Ki-Hyeon Kim*, Junghoon Moon°, Woochang Kwon*, Byungyeon Park*, Woojin Seok*,
Won-taek Hong*, Sang-kwon Lee*, Jin-Yong Jo*, Taejin Yoon**, Jaehein Chio**,
DaeKyeom Kim***, Dongah Hwang***, SunWoong Choi***, Jongho Kim****,
Junyeop Kim, KiHyoun Kim, Byonghoon Chong, Dosub Lee,
Young-Geun Yu, Eun-Sook Park, Jae-Hong Cheon

요 약

최근 ICT 기술을 선도하는 핵심 분야는 단연 빅데이터와 AI이며, 과학기술 분야에서도 기존에는 시뮬레이션 중심의 연구 수행 형태에서 대용량 데이터 기반의 AI 분석 기법을 이용한 연구 환경으로 변화하고 있다. 이러한 변화는 데이터의 사이즈의 변화에 따른 데이터 전송 속도의 문제와 AI를 연구하기 위한 컴퓨팅 환경 부족의 문제와 직면하였다. 또한 기존의 클라우드 컴퓨팅 기술은 다수의 사용자가 사용할 경우 데이터 전송량의 급증으로 데이터 전송 지연, 장애대응의 문제가 발생하였다. 이를 위해 분산처리 환경의 컴퓨팅 기술을 사용하여 문제를 해결한다. 따라서 과학기술연구망에서는 연구 환경의 변화에 맞추어 빅데이터를 고속으로 전송할 수 있는 ScienceDMZ 환경 구축 및 부족한 컴퓨팅 인프라 환경을 개선하고자 한다. 본 논문에서는 출연연구소 8개의 기관과 협력하여 ScienceDMZ 기반의 빅데이터 전용 네트워크를 구축하고 구축된 인프라 위에 컨테이너 기반의 분산 처리가 가능한 AI 연구 환경을 제공하는 플랫폼을 개발하여 기존의 문제들을 해결하고자 한다. 국내의 ScienceDMZ 환경 구축 사례를 통해 대역폭 대비 80% 이상의 성능 향상의 결과를 보이며, AI 플랫폼의 CPU, GPU 컴퓨팅을 이용한 싱글, 병렬, 분산 컴퓨팅의 성능 비교를 통해 분산 컴퓨팅 성능의 우수성을 보이고자 한다.

Key Words : ScienceDMZ, Distributed Computing, DTN, Kubernetes, Big Data, AI

※ 본 연구는 2021년도 한국과학기술정보연구원(KISTI) 주요사업 과제로 수행한 것입니다.

◆ First Author : Korea Institute of Science and Technology Information(KISTI), kkh1258@kisti.re.kr, 정희원

° Corresponding Author : Korea Institute of Science and Technology Information(KISTI), jhmoon@kisti.re.kr, 정희원

* 한국과학기술정보연구원(KISTI)

** 한국과학기술원

*** 한국생명공학연구원

**** 한국환의학연구원

논문번호 : 202012-304-C-RN, Received December 7, 2020; Revised February 26, 2021; Accepted March 2, 2021

ABSTRACT

Recently, the key areas leading ICT technology are by far big data and AI. In the field of science and technology, it is changing from a simulation-oriented research practice to a research environment using AI analysis techniques based on large-scale data. This change faced the problem of data transmission speed according to the change of data size and the problem of lack of computing environment to study AI. In addition, when the existing cloud computing technology is used by a large number of users, the data transmission delay and failure response problems occurred due to the rapid increase in data transmission volume. To this end, we solve the problem using computing technology in distributed processing environment. To this end, the problems are solved by using computing technology in distributed processing environment. Therefore, KREONET aims to improve the fast transmission of big data and the insufficient AI computing environment, centered around the establishment of the ScienceDMZ environment. In this paper, we aim to solve existing problems by building big data dedicated network based on ScienceDMZ and DTNs capable of distributed computing in cooperation with the eight institutions of super facility and develop a platform that provides container-based AI research environment on the established ScienceDMZ infrastructure. Through the case of establishing the ScienceDMZ environment in Korea, the result of performance improvement of more than 80% compared to the bandwidth is shown, and the superiority of distributed computing performance is shown by comparing the performance of single, parallel, and distributed computing using the CPU and GPU computing of the AI platform.

I. 서 론

우리의 생활에서 사용하는 기기들 그리고 움직이는 모든 것들이 이제는 데이터가 되고, 데이터들이 수집되고 가공되어 빅데이터로 재생산된다. 빅데이터는 AI 기술과 융합되어 다양한 연구들이 진행되고 있고, ICT 기술을 선도하는 연구 분야들로 손꼽히고 있다. 실제로 국내에서 AI와 빅데이터를 이용한 연구들이 주를 이루고 있다. 한국 IDC에서 최근 발간한 보고서에서는 “국내 인공지능 2019-2023 시장 전망 연구 보고서에서 국내 인공지능(AI) 시장이 향후 5년 간 연평균 17.8% 성장해 2023년 6400억 원 이상의 규모를 형성할 것이라고 전망한다^[1]. 이와 같이 국내뿐만 아니라 국제적으로도 AI에 대한 시장 규모는 점점 증가

할 것이다. 또한 그림 1은 나라별 분산 컴퓨팅 시장 성장 비율을 보여준다^[2]. 이에 따라 과학기술 연구 분야에서도 기존의 시뮬레이션 중심의 연구 수행 형태에서 대용량의 데이터를 기반으로 AI 분석 및 계산을 수행하는 형태의 연구 환경으로 변화하고 있는 추세이다.

국내의 연구 기관에서는 빅데이터 전송 및 AI 연구를 수행하기 위해서는 방대한 양의 데이터를 확보, 저장, 처리, 전송 등을 처리해야 하는 일련의 과정을 수행할 수 있는 시스템이 필요하다. 하지만 국내 연구기관들의 경우 빅데이터 전송을 위한 고속의 네트워크 인프라와 AI 연구를 위한 시스템이 구축되어 있지 않다. 기존의 네트워크 환경을 이용하여 빅데이터를 전송할 경우 방화벽을 비롯한 복잡한 보안정책으로 전송 성능을 떨어뜨리며, 특히 연구전용 트래픽과 일반 트래픽의 혼재로 네트워크 성능은 더욱 떨어지고 있는 상황이다. 이러한 전송 환경의 결과로 빅데이터 기반의 연구를 위한 데이터의 전송 성능은 매우 열악한 상황이다. 따라서 빅데이터 전용의 네트워크 환경이 필요하다. 또한 기존의 컴퓨팅 인프라들의 경우 중앙 집중식 구조로 구성된 클라우드 컴퓨팅 기술을 활용한 인프라로 구축되어 있다. 클라우드 컴퓨팅의 경우 다수의 사용자들이 데이터를 전송할 때, 네트워크 전송 지연 문제가 발생할 수 있으며, 보안 및 접근성의

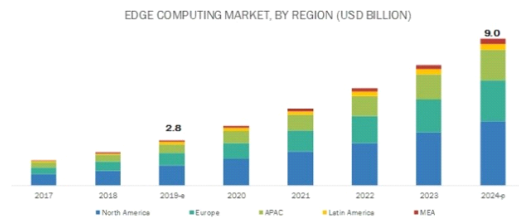


그림 1. 나라별 분산 컴퓨팅 시장 조사
Fig. 1. Distributed Computing Market Research by Country

문제^{3,4)}가 발생한다. 이러한 문제들을 해결하기 위해 출연연구소들의 빅데이터 전송 환경을 개선하고자 ScienceDMZ 기술을 이용하고, 컴퓨팅 인프라의 문제를 해결하기 위해 분산 컴퓨팅 기술을 이용하여 AI 연구 환경을 제공하는 플랫폼과 결합된 형태의 인프라를 제공하고자 한다. 본 논문에서의 분산 컴퓨팅 환경은 옛지 컴퓨팅 환경의 구성과 유사하며, 분산 컴퓨팅 환경으로 구성할 시 지역 적으로 분산되어 서버들이 구성되기 때문에 전국에 있는 많은 사용자들이 하나의 서버에 집중적으로 접속하지 않고 분산되어 접속하기 때문에 네트워크의 병목 문제가 크게 발생하지 않으며, 중앙 집중적이지 않은 구성이 데이터를 손상시킬 수 있는 공격으로 부터 보호할 수 있다. 또한 현재 나의 위치에서 가장 가까운 위치의 서버에 접속하여 컴퓨팅을 수행하기 때문에 높은 접근성을 갖을 수 있다.

ScienceDMZ⁵⁾는 중단 간 전송효율성을 극대화하기 위해 네트워크, 데이터 전송 노드 그리고 기관 내 네트워크 보안 정책 등 복합적인 구성요소들을 최적화하는 혁신적인 기술을 말한다. ScienceDMZ 기술은 기본적으로 인터넷을 위한 네트워크와 연구를 위한 네트워크를 분리하여 구성하여 기존에 TCP 성능 저하를 유발하는 패킷의 손실을 줄이거나 제거할 수 있다. 또한 데이터를 빠르게 전송할 수 있는 최적의 서버인 데이터 전송 노드(Data Transfer Node, DTN)를 이용하여 빅데이터 고속 전송을 수행한다. ScienceDMZ 및 DTN 서버 구성을 위한 방법은 3-1장에서 자세하게 설명한다. 이와 더불어 DTN 서버에 GPU를 설치하여 AI 컴퓨팅을 수행할 수 있도록 AI 연구 환경 제공 플랫폼을 DTN 위에 개발한다. 이를 통해 DTN 서버는 DTN 간 데이터를 빠르게 전송하기 위해 사용될 뿐만 아니라 AI 연구를 수행하는 GPU 컴퓨팅에 사용된다.

AI 연구 환경 제공 시스템은 컨테이너 기반의 AI 연구 환경을 제공 및 공유 할 수 있는 플랫폼을 말하며, 분산 컴퓨팅 환경으로 구성하여 기존의 클라우드 컴퓨팅 환경에서 발생하는 문제점들을 해결할 수 있다. 분산 컴퓨팅 기술의 경우 모든 서버들이 지역적으로 분산되어 구성되기 때문에 각 기관에서 가장 가까운 서버에서 컴퓨팅을 수행할 수 있어 접근성이 좋으며, 여러 사용자들이 데이터를 전송하더라도 분산되어 데이터를 전송하기 때문에 네트워크 병목구간이 발생하지 않는다는 장점⁶⁾을 갖기 때문에 본 연구에서는 분산 컴퓨팅 환경을 이용하여 인프라를 구성한다.

국내의 경우 25개 출연연구소들(소재과학, 입자물

리학, 천체물리학, 바이오메디컬, 지구과학, 고해상도 가지화 및 가상현실 등의 분야)은 인공지능과 빅데이터가 융합된 다양한 연구들이 진행되고 있다. 출연연구소에서 필요로 하는 컴퓨팅 인프라를 구성하고, 이를 토대로 AI 연구를 돕기 위한 플랫폼을 제공함으로써, 기존에 사용되어오던 CLI(Command Line Interface) 기반의 시스템이 아닌 Web UI 기반의 시스템을 제공하여 사용자들이 손쉽게 AI 연구를 수행할 수 있도록 구성한다.

국가과학기술연구망(KREONET)⁷⁾은 이러한 출연연구소들의 과학 연구 수행을 돕기 위해 8개의 출연연구소들(한국과학기술정보연구원, 한국항공우주연구원, 한국화학연구원, 한국한의학연구원, 한국핵융합에너지연구원, 한국과학기술원, 한국에너지기술연구원, 한국생명공학연구원)과 함께 ScienceDMZ 구축 및 AI 기반 빅데이터 분석 기술을 제공하는 프로젝트 R&E Together를 진행하고 있으며, 이 프로젝트를 통해 AI 연구자가 다른 AI 연구자로 부터 빅데이터를 전송 받아 즉시 시스템에 적용하여 연구 결과를 도출하는 미래형 과학 협업 인프라 서비스를 제공하는 것을 목표로 하고 있다.

국내에서 ScienceDMZ 기술을 적용한 사례에 대해 자세하게 알아보고 빅데이터 전송을 수행하였을 때, 기존의 네트워크 구조와 ScienceDMZ를 구축하였을 때의 Disk to Disk 데이터 전송 성능을 비교하며, AI 연구 환경 제공 플랫폼을 통해 CPU, GPU를 이용한 싱글 컴퓨팅, 병렬 컴퓨팅 및 분산 컴퓨팅을 통한 AI 처리 성능을 비교하여 시스템의 성능 결과를 보이고자 한다.

II. 연구 동향 및 관련 논문

2.1 ScienceDMZ 연구 동향

2010년 초반 미국의 ESnet에서 시작된 ScienceDMZ 개념의 정립과 실제 슈퍼컴퓨팅센터, 데이터센터 등에 적용을 통해 연구 전용 데이터의 전송 성능을 획기적으로 개선하였으며, 이를 토대로 2015년부터 미국의 과학재단인 NSF의 지원을 통해 PRP(Pacific Research Platform)⁸⁾ 프로젝트는 ScienceDMZ 기술을 고도화하고 확대 적용하여 미국내 빅데이터 기반 주요 6개 연구분야를 중점 지원하는 리서치 플랫폼으로 고도화를 진행하고 있다. PRP는 미국의 서부 지역의 저명한 연구소와 대학교를 중심으로 기존의 백본 업그レード 사업을 통해 구축된 100G 백본을 이용하여 주요 거점 지역의 ScienceDMZ를 고속으로 연동함으

로써 빅데이터 전달 체계와 컴퓨팅 체계를 연계하는 하나의 거대한 리서치 플랫폼으로 구축하여 현재는 NRP 프로젝트로 확대 되었다. NRP(National Research Platform)^[9] 프로젝트는 미국 서부 지역으로 한정 지었던 한계를 벗어나 미국 전역으로 확대되어 ScienceDMZ을 구축하고 있으며, ScienceDMZ 기반의 빅데이터 고속도로 체계를 구축하고 CPU, GPU 기반의 DTN(Data Transfer Node)^[10]을 이용하여 AI 플랫폼을 연계하여 연구 기관 및 교육기관에 대한 빅데이터 기반의 AI 연구와 교육을 지원하고 있다.

이러한 국제 동향을 통해 최근 아시아 국가들도 ScienceDMZ 구축을 진행하고 있으며, 이를 토대로 APRP(Asia Pacific Research Platform)^[11] 프로젝트를 진행하고 있다. APRP에 속한 국가들 중 ScienceDMZ을 구축하여 운영하고 있는 나라들은 한국, 호주, 싱가포르, 뉴질랜드, 일본, 파키스탄, 말레이시아, 필리핀 등이 있으며, 점차적으로 아시아 지역에서 늘어나고 있는 추세이다. 위의 나라들 중 ScienceDMZ을 구축하고 CPU, GPU 기반의 DTN을 이용하여 AI 플랫폼을 연계하는 작업을 진행하고 있는 나라는 한국이 유일하며, 미국, 유럽, 아시아 지역을 빅데이터 고속도로 체계를 구축하는 프로젝트인 GRP(Global Research Platform)^[12] 프로젝트가 현재 진행 중이다. GRP 프로젝트를 통해 한국은 선진국의 빅데이터 거점으로서 아시아 지역 허브 역할을 하고 있으며, 이러한 빅데이터를 아시아 지역에 공유하고, 유관 기술을 선도하고 있다.

PRP 프로젝트의 경우 주요 빅데이터 연구 분야를 6개의 그룹(입자물리, 천체관측 및 물리, 화학, 바이오 메디컬, 지구과학, 인공지능, 가상현실 등)으로 지정하여 연구 기관의 네트워크 구조 및 연구에 특화된 서비스를 제공하며, 응용연구를 위한 고성능의 컴퓨팅 자원, 빅데이터 처리를 위한 데이터 센터와 빅데이터 전송을 위한 빅데이터 고속도로 체계를 자유롭게 연동할 수 있는 리서치 플랫폼을 구축하고 고도화 하고 있다.

미국에서 ScienceDMZ을 적용한 사례 중 의료 ScienceDMZ^[13]을 구축하기 위한 노력을 하고 있으며, 기존의 ScienceDMZ의 경우 네트워크를 구성할 때 VLAN 또는 라이트패스를 이용하여 구성하였다면, 의료 ScienceDMZ의 경우 보안적으로 민감한 데이터들을 전송하기 때문에 기존의 네트워크 구성과는 다르게 구성되고 있다. 의료 ScienceDMZ를 구성하기 위한 네트워크 구조를 보면 ScienceDMZ 구간 앞단에 Openflow를 지원하는 스위치와 Firewall, IDS 등의

| 연구분야 | 기관 | 연구내용/연구망 역할 | 연구망 속도 |
|------------------------------------|----------------------------|---|----------|
| 고에너지물리 (LHC) | 서울대학교 적자계이지 이론연구단 외 14개 기관 | - 입자가속기 중심의 실험데이터를 국내연구단의 클라우드 전송/활용 | 1G-10G |
| 천문우주 (Astronomy) | 천문연구원, 국토지리정보연구원(천파망원경센터) | - 울산, 세종, 제주, 서울에 구축된 천파망원경 관측 데이터의 상관센터 중심의 데이터 공유 | 100G-40G |
| | 천문연구원(SDO) | - NASA 태양위성 관측 이미지를 글로벌 전송 및 국내 우주기상센터간 전송 | 10G |
| | 천문연구원(SKA, LSST) | - 칠레, 남아프리카공화국 역 위치한 망시와 망원경 관측 데이터의 공유 | 10G |
| 게놈/바이오 (Genome/Bio) | 서울대병원 CMI센터의 8개 기관 | - ICGC 유전체 데이터 중심 RDC센터 구축 | 10G |
| 기상기후 (Weather/Climato) | 기상청의 5개 기관 | - 기상청 슈퍼컴퓨터 및 기상데이터 공유 및 연구 활용 | 1G-40G |
| KSTAR (Fusion Research) | NFRRI의 3개 기관 | - KSTAR를 활용한 데이터 저장 및 연구 활용 | 100G |
| 기초연구 관측장비 (Basic Science & Survey) | 기초연구원 | - 대형현미경으로 관찰된 고해상도이미지 및 데이터 전송 | 10G |
| 연구/교육 (Research & Education) | 부산대학교의 5개기관 | - 데이터, 계산자원, 저장자원의 연계를 통한 대규모 인력교육(원격) | 1G-10G |
| 건설 건축 | KOCEC-CI(1,2차 실험센터) | - 지진실험 등 대규모 건설/건축 실험장비를 활용한 연구 및 교육 | 1G |
| 국내 가속기 | 포항가속기센터(4세대) | - 포항가속기센터에서 연간 실험/발생되는 데이터의 공유 | 10G |

그림. 2. 국내 출연연의 빅데이터 기반 연구 그룹과 네트워크 환경^[14]

Fig. 2. Domestic big data research groups and network environment

보안 장비들이 같이 존재하면 전체적인 Flow를 확인하기 위한 Controller가 존재한다. Inbound/Outbound 플로우들 중 알려지지 않은 데이터에 대해서는 컨트롤러와 보안장비를 거쳐야만 지나갈 수 있도록 구성되어 있다.

국내의 경우 그림 2와 같이 ScienceDMZ 확장 구축을 통해 거대과학연구 및 빅데이터 기반의 연구 분야인 고에너지물리, 천문우주, 게놈/바이오, 기상기후, KSTAR, 기초연구/관측장비, 연구/교육 등과 KREONET과의 협력으로써, 각 기관들은 연구자의 수요에 따라 1GE ~ 100GE까지의 다양한 연동 환경으로 빅데이터 고속도로를 구축하였다^[14].

2.2 AI 컴퓨팅 연계 플랫폼 관련 논문

최근 데이터센터에서는 GPU나 FPGA같은 가속기들을 이용한 계산 작업이 활발히 진행되고 있으며 이에 따라 이러한 자원들을 활용하는 방법들에 대한 연구를 많이 진행하고 있다. 컨테이너 기반의 환경에서 GPU 컴퓨팅에 대한 연구로 Kube-Knots이 있다^[15]. Kube-Knots는 현재의 데이터센터에서 사용하고 있는 최신의 스케줄러가 지연속도에 민감한 작업들에 자원을 세밀하게 분배해주지 못한다는 문제를 발견하여 이를 해결하기 위해 개발되었다. Kube-Knots은 GPU 자원 인식 오케스트레이션이고 쿠버네티스의 컨테이너를 통해 빌드 할 수 있다. Kube-Knots에서는 2가지의 스케줄링 기법을 설계하고 평가한다. 상관 기반 예측과 피크 예측을 통해 님러닝 작업에 필요한 시간을 데이터센터의 최신 스케줄러 대비 36% 감소시켰다.

다른 연구로는 컨테이너 기반 클러스터 환경에서

GPU 자원을 메모리 자원의 부족 없이 컨테이너 간 공유해서 사용할 수 있도록 하는 방법을 제안해 전체적인 성능이 향상되도록 하는 연구가 있다¹⁶⁾. 논문에서 제안한 Adaptive Fair-share 방법을 기본 방법과 Fair-share 방법을 동일 작업 실행과 다른 작업 실행을 통해 비교하였고 그 결과 작업에 필요한 시간이 크게 줄고 메모리의 활용률이 크게 증가되었음을 확인할 수 있다.

컨테이너 기반의 딥러닝 수행과 관련된 연구로는 Optimus가 있다. Optimus는 딥러닝 클러스터를 위한 효율적인 동적 자원 스케줄러이다¹⁷⁾. 현재의 클러스터 스케줄러는 딥러닝 작업에 맞도록 되어 있지 않고 구체적으로 고정되어 있는 양 만큼의 자원을 사용할 수 있기 때문에 자원의 효율과 작업의 성능을 향상시키기 어렵다는 문제가 있다. Optimus에서는 학습에 예측모델을 사용하고 성능 모델을 만들어 정확하게 학습시간 측정하여 필요한 자원을 할당해주도록 하였다. Optimus와 기존의 스케줄러의 성능을 비교한 결과 전체적인 작업 완료시간이 62% 정도로 단축되었다.

컨테이너 기반의 클라우드를 통해 자원을 효율적으로 사용할 수 있고 소프트웨어의 유지 보수가 간단해졌지만, 컨테이너 환경에서 딥러닝 작업을 수행 시 성능에 어떠한 악영향을 줄 수 있는지에 대한 연구가 부족하기 때문에 이러한 테스트를 진행한 연구도 있다¹⁸⁾. 이 연구에서는 서버와 PC를 가지고 컨테이너 환경에서 실행했을 때와 베어메탈 상태에서 실행했을 때를 각각 비교하였다. 결과 값으로 매우 근소한 차이만을 보이고 있어 도커 컨테이너가 딥러닝 실행에 있어 큰 문제가 없음이 확인되었다.

III. R&E Together 인프라 구축 및 개발

R&E Together 인프라는 빅데이터 전송 시스템과 AI 오케스트레이션 시스템으로 구성된다. R&E Together 인프라 구성 개념도는 그림 3과 같다.

빅데이터 전송 시스템은 ScienceDMZ 인프라 구축과 최적의 데이터 전송 노드인 DTN을 이용하여 빅데이터의 손실이 없는 고속의 데이터 전송을 위한 네트워크를 구축하는 것을 말하며, 실제로 데이터 전송 시스템을 구축하기 위해 각 출연연구소(한국과학기술정보연구원, 한국항공우주연구원, 한국화학연구원, 한국한의학연구원, 한국핵융합에너지연구원, 한국과학기술원, 한국에너지기술연구원, 한국생명공학연구원)의 전산 담당자들과 협력하여 각 기관의 DTN 서버를 구

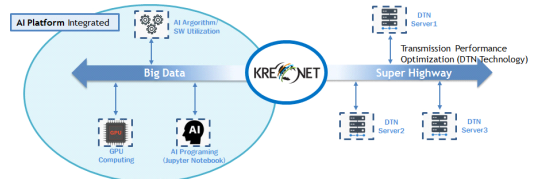


그림 3. R&E Together 인프라 개념도
Fig. 3. R&E Together infrastructure conceptual diagram

축하고 ScienceDMZ을 위한 네트워크 구축을 수행하였다. 그림 4는 실제로 각 기관의 전산 담당자들과 ScienceDMZ 구축 후 찍은 사진들이며, 사진에 없는 기관들은 추후 ScienceDMZ 구축을 진행할 예정이다.

AI 오케스트레이션 시스템은 각 기관에 설치된 DTN 서버들에 GPU를 장착하고 이 서버들을 하나의 거대한 클러스터로 구성 후 컨테이너 기반의 컴퓨팅, 데이터 저장소, AI 프로그래밍을 제공하는 AI를 연구하는 환경을 제공하기 위한 시스템을 개발하고 있다. 본 절에서는 R&E Together 인프라를 구성하는 구성요소들과 AI 오케스트레이션 시스템의 개발 과정에 대해 자세하게 설명한다.



그림 4. ScienceDMZ 구축 기관
Fig. 4. R&E Together infrastructure conceptual diagram

3.1 빅데이터 전송 시스템 구축

기존의 대학교 및 연구소들의 네트워크 구조를 보면 그림 5와 유사하게 구성되어 있다. 그림 5를 보면 데이터를 전송하는 서버들이 내부의 Border Router를 거쳐 다양한 보안 장비들이 연결되어 있는 인터넷 망으로 구성되어 있다. 이와 같이 구성되어 있는 구조에서 데이터를 전송할 경우 TCP의 성능을 저하시키는 패킷 로스 현상이 빈번하게 일어난다. 인터넷 망을 사용하기 때문에 패킷 로스가 발생하지만 다양한 전송장비들 또한 패킷 리다이렉션 현상이 발생하여 TCP 성능의 저하를 발생시킨다. 전송 장비들의 경우 1Gbps 이상의 네트워크를 지원하는 경우 가격이 비싸며, 구매를 하더라도 1Gbps의 여러 링크들을 하나로 묶어 구성하기 때문에 네트워크의 성능 저하를 가져

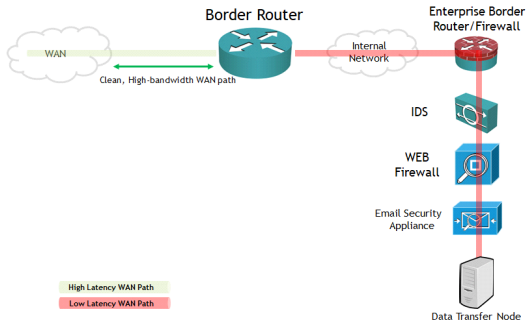


그림 5. 기존 연구소들의 네트워크 구성도
Fig. 5. Existing Institutions Network Structure

을 수 있다. 따라서 데이터 전송 서버의 경우 보안장비와 같은 네트워크에 구성할 경우 전송 성능의 저하를 가져온다. 이러한 문제들을 해결하기 위해 ScienceDMZ 네트워크 구조를 구성하고 데이터 전송 최적의 서버인 DTN(DA TA Transfer Node)을 구축한다. 그림 6은 ScienceDMZ 네트워크를 구축한 구성도이다. ScienceDMZ의 네트워크 구조를 구성하기 위해서는 기존에 사용해오던 네트워크 구조에서 일반 인터넷 망과 연구를 위한 연구망 망을 분리하여 구성해야 한다. 이를 위해 일반 망에 존재하는 기존 보안장비를 구성했던 내부 네트워크에 설치되어 있던 데이터 전송 서버를 그림 6과 같이 Border Router에서 하나의 라인을 따로 구성하여 DTN과 Perfsonar 서버를 연결해야 한다. 이와 같이 DTN 서버를 구성하였을 때 보안장비들을 우회하여 성능이 향상되며, 일반 인터넷 망의 트래픽 혼재를 막을 수 있기 때문에 대역폭 대비 높은 성능을 제공할 수 있다.

KISTI에서는 각 기관의 네트워크를 ScienceDMZ으로 구축하기 위한 네트워크 스키마는 그림 7과 같다. 그림 7을 보면 왼쪽이 KISTI의 ScienceDMZ 구성 네트워크이며 오른쪽이 각 기관의 ScienceDMZ 구성 네트워크이다. KISTI와 각 기관 간에 네트워크 연결을 수행할 때, 기관의 내부 망으로 들어가기 이전 라우터에서 하나의 새로운 네트워크 라인을 이용하여 DTN 서버들을 설치하고 데이터를 전송할 때 보안장비를 구성한 네트워크를 거치지 않고 데이터를 전송하기 때문에 TCP 성능 저하 문제가 발생하지 않는다. 또한 인터넷 망과 분리되어 인터넷 망에서 존재하는 작은 단위의 네트워크 커넥션을 피할 수 있다. ScienceDMZ을 구성하기 위해서는 이와 같은 네트워크의 구조가 뒷받침 되어야 하며, 데이터를 전송하기 위한 시스템이 필요하다.

DTN은 높은 스펙의 하드웨어로 구성된 서버이며,

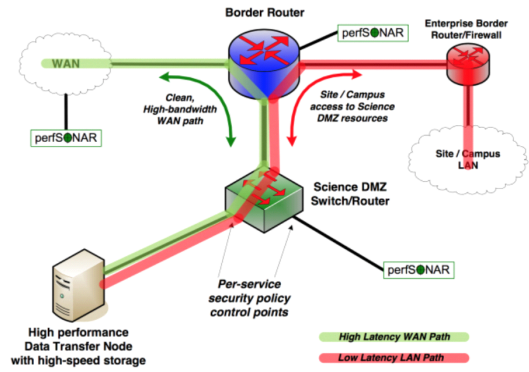


그림 6. ScienceDMZ 네트워크 구성도[5]
Fig. 6. ScienceDMZ Network Structure

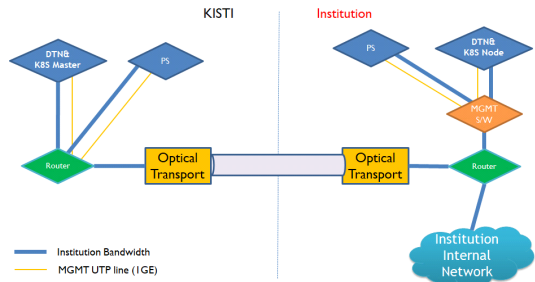


그림 7. 출연연구소 ScienceDMZ 네트워크 스키마
Fig. 7. government-contributed research institute ScienceDMZ Network Schema

데이터 전송에 최적화 시켜 전송 성능을 극대화할 수 있다. DTN 서버를 구성할 때, 하드웨어적으로 필요한 부분들이 있다. CPU의 경우 Core 수 보다는 클럭 속도가 매우 중요하며, 디스크의 경우 SSD 또는 NVMe SSD를 사용하는 것을 권장한다. 또한 네트워크 인터페이스 카드의 경우 40G NIC 또는 100G NIC을 사용하는 것을 권장한다. 네트워크 인터페이스 카드의 경우 DTN에서 높은 대역폭을 요구하는 이유는 대용량의 데이터를 전송하기 위해서는 1G 이상의 고 대역폭 환경에서 DTN을 사용하였을 때 더 효율적으로 사용할 수 있기 때문이다.

DTN을 구성하기 위해서는 시스템 튜닝이 필요하다. 기본적으로 네트워크 튜닝과 하드웨어 튜닝으로 나누어진다. 네트워크 튜닝의 경우 MTU(maximum transmission unit) 사이즈 튜닝과 txqueuelen 사이즈 튜닝을 이용하여 높은 대역폭에서 사용되는 환경으로 설정할 수 있다. MTU 사이즈를 설정하기 위한 설정 명령어는 'ifconfig <Network Interface> mtu 9000'으로 설정한다. DTN의 네트워크 속도 측면에서 가장 큰 역할을 수행하는 부분이 바로 점보프레임 크기가

다. 짐보프레임은 서버에서 데이터를 전송할 때 전송하는 패킷의 사이즈를 늘려 고속 전송을 가능하게 해주는 기술로써, MTU 사이즈를 통해 변경할 수 있다. 서버에서 처음 설정되어 있는 값은 1500이지만 위의 명령어를 통해 9000으로 설정을 변경할 수 있다. MTU 사이즈의 경우 서버에서만 설정을 변경해 주는 것이 아니며, 스위치에서도 MTU 사이즈 변경을 수행하여 서버에서 다른 서버로 통하는 모든 네트워크의 MTU 사이즈를 맞추어야만 정확한 네트워크 성능을 보장받을 수 있다.

다음 옵션으로는 송신 큐 사이즈의 설정이다. 송신 큐 사이즈의 경우 데이터를 전송할 때, 데이터의 사이즈가 커지게 되면 많은 큐를 사용하게 된다. 그에 따라 송신 큐를 모두 사용하게 되면 패킷의 손실이 발생하기 때문에 큐의 사이즈를 늘려주어야 한다. 이때 사용할 수 있는 명령어는 'ifconfig <Network Interface> txqueuelen 10000' 으로 설정할 수 있다. 서버에서 처음 설정되어 있는 기본 값은 1000으로 대용량의 데이터를 전송하기 위한 환경으로 변경하기 위해서는 10000으로 변경해줘야 한다.

DTN의 하드웨어 튜닝은 OS의 전력 효율화를 위해 CPU 사용이 낮은 경우 IDLE 상태가 되도록 구성되어 있다. CPU의 높은 성능이 필요한 작업을 수행하게 되면 CPU가 유휴 상태에서 최고 성능 상태로 진입하는데 오버헤드가 존재하기 때문에 전송 성능이 낮아진다. 이와 같은 문제를 해결하기 위해 CPU의 성능을 항상 최고 성능으로 동작하도록 'cpupower frequency -set -g performance' 명령어를 통해 설정을 변경할 수 있다.

데이터의 전송을 수행할 때 실제로 네트워크의 성능은 높지만, Disk to Disk의 성능을 측정해보면 네트워크 성능의 50% 조차도 나오지 않는 경우가 많다. 이와 같은 경우 네트워크를 통해 전송된 데이터들을 Disk로 옮기는 과정에서 읽고 쓰는 속도를 높여야 한다. 스토리지의 속도는 스토리지의 종류와 스토리지의 구성 방법에 따라 속도를 높일 수 있다. 스토리지의 종류에는 HDD, Sata, SSD, NVMe SSD가 있다. 스토리지의 종류에서 속도를 비교해 보면 HDD < Sata < SSD < NVMe SSD 순서로 성능이 높아진다. HDD와 NVMe SSD의 쓰기 속도는 약 22배 정도의 차이를 보인다.

DTN 서버와 더불어 DTN 서버들 간에 데이터를 전송하기 위해 필수적으로 필요한 어플리케이션이 존재한다. KISTI에서는 Globus Online^[19]이라는 어플리케이션을 이용하여 데이터를 전송하고 있다. Globus

Online은 전송에 참여하는 종단 호스트들 간에 대용량 데이터의 고속 전송 기능을 SaaS(Software as a Service) 방식으로 제공하는 소프트웨어 기반 클라우드 플랫폼 서비스로서 3자간 파일 전송 기능을 통해 전송 요청, 종단간 파일 전송 및 전송완료 공지의 단계로 전송 과정의 단순화 및 사용자 편의성을 제공한다. 또한, 이러한 전송 기능은 GridFTP를 기반으로 동작하도록 구현되어 있고, GridFTP는 일반적인 파일 전송 프로토콜들에 비해 병행성(Concurrency), 병렬성(Parallelism)의 수준을 조절하여 다수의 TCP 스트림들을 동시에 활용하여 전송할 수 있도록 함으로써 고대역의 ScienceDMZ 네트워크 자원의 활용도를 극대화시킬 수 있고 궁극적으로 높은 전송 성능을 가져올 수 있다.

3.2 AI 오케스트레이션 시스템 개발

AI 오케스트레이션 시스템은 기본적으로 3.1 장에서 데이터 최적화 전송을 위한 시스템인 DTN 서버에 설치되어 있는 GPU 리소스를 이용하여 사용자들이 분산된 컴퓨팅 환경에서 AI 연구를 손쉽게 수행할 수 있도록 개발한 시스템이다. AI 오케스트레이션 시스템은 분산된 DTN 서버들을 활용하여 거대한 하나의 서버와 같이 구성하기 위해 Kubernetes^[20]를 사용한다. Kubernetes는 기존에 컨테이너 기반의 가상화 엔진 툴인 Docker를 관리해주는 툴로써, Google에서 개발하여 시스템 가상화 분야에서 많이 활용되고 있다. 컨테이너의 경우 기존에 사용되던 가상화 시스템인 VM(Virtual Machine) 보다 빠르고 가볍기 때문에 서버 가상화를 사용하는 기업에서 높은 점유율을 보이고 있다. 하지만 컨테이너 특성 상 Desktop 모드를 지원하지 않기 때문에 Windows Server, Desktop 모드의 리눅스 등 Desktop을 필요로 하는 OS에서는 사용할 수 없다는 단점이 존재한다. 본 시스템을 구성하기 위해 Kubernetes를 사용하는 이유는 두 가지이다. 첫 번째는 R&E Together 시스템의 특성상 분산된 서버들을 마치 하나의 거대한 클러스터와 같이 구성하기 위해서이며, 두 번째는 Kubernetes의 경우 사용자가 필요한 리소스를 요청할 시, 그 자원을 수용할 수 있는 서버를 자동적으로 찾아 컨테이너를 할당하는 기능을 사용하기 위함이다. Kubernetes를 모든 DTN 서버에 구축하여 마스터 노드와 워커노드로 나누어 구성하고, Kubernetes에서 제공되는 Kubelet을 이용하여 마스터와 노드 간에 컨테이너를 생성, 수정, 삭제 수행한다.

AI 오케스트레이션 시스템의 구성도는 그림 8과

같다. 구성도를 보면 Kubernetes에서 제공되는 REST API를 이용하여 사용자들의 요청을 처리하도록 Web UI를 구성한다. 그림 8을 보면 AI 오케스트레이션 시스템은 총 4가지의 모듈로 Web UI와 연계되어 구성된다. 첫 번째 모듈은 로그인 모듈로써, 본 시스템에서는 기존에 KREONET에서 개발한 KAFE(Korea Access Federation)^[21]를 이용하여 편하게 로그인을 수행할 수 있도록 구성하였다. KAFE는 ID연합에 포함되어 있는 ID 제공자 기관의 구성원들이 별도의 사용자 계정을 생성하지 않아도 소속 기관의 포털계정을 이용하여 KAFE에 연합되어 있는 서비스에 로그인할 수 있는 연합 인증 체계이다. 연합인증 체계의 경우 SAML(Security Assertion Markup Language)을 이용하여 식별정보제공자와 서비스제공자 간의 인증을 수행하고 인가정보를 교환한다. 연구협업 서비스 및 연구자원을 제공하는 기관인 서비스제공자는 웹응용서버에 SAML 기능을 구현해야하며, 서비스 제공자는 로그인 기능을 제공하는 기관인 아이디 제공자로부터 사용자 인증과 인가 절차를 거쳐 로그인을 수행할 수 있다^[22]. 그림 9는 KAFE의 사용자 인증 및 인가 절차를 보여준다.

두 번째 모듈은 계산 모듈이다. 계산 모듈은 오픈소스 프로젝트인 Jupyter^[23]를 이용하여 구성되며, Jupyter 프로젝트는 다양한 프로그래밍 언어를 이용하여 인터랙티브하게 데이터 과학과 사이언티픽 컴퓨팅을 지원하는 오픈소스 어플리케이션으로 주로 데이터 변형, 수치 시뮬레이션, 통계 모델링, AI 등에 사용된다. Jupyter 프로젝트는 크게 Jupyter Notebook과 Jupyter Hub로 구성된다. Jupyter Notebook은 40개 이상의 다양한 언어를 지원하며, 실시간 대화형 위젯을 제공하고, 싱글 유저를 위한 시스템이며, Jupyter Hub 또한 기능은 유사하나 인증 시스템과의 연계를 통한 멀티유저를 위한 시스템 구성을 위해 사용된다.

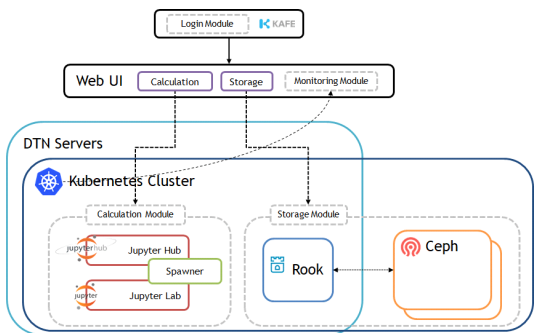


그림 8. AI 오케스트레이션 시스템 구성도
Fig. 8. AI Orchestration System Structure Diagram

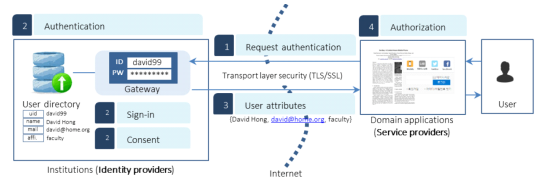


그림 9. KAFE 사용자 인증 및 인가 절차^[22]
Fig. 9. KAFE user authentication and authorization process

Jupyter 어플리케이션을 사용하기 위해서는 Jupyter Notebook과 Hub를 Docker 이미지로 구성하여 Kubernetes로 부터 컨테이너를 요청할 시 필요한 리소스(CPU, GPU, 스토리지)와 필요한 Jupyter 이미지를 YAML 파일로 생성하여 요청 시 AI 연구 환경을 구성할 수 있다.

Jupyter Hub를 사용하는 그룹 사용자의 경우 KAFE를 통해 인증 받은 사용자가 그룹의 장이 되어 Jupyter Hub 그룹 사용자를 위한 AI 연구 환경을 구성할 수 있다. 그룹의 장은 KAFE를 통해 인증된 사용자가 Jupyter Hub에 접속할 그룹 사용자들의 아이디와 패스워드를 만들 수 있으며, 만약 대량의 사용자 등록이 필요한 경우 이메일, 사용자 이름, 패스워드를 적은 엑셀 파일을 이용하여 사용자를 등록하면, 대량의 사용자들은 생성된 Jupyter Hub 아이디, 패스워드를 이용하여 그룹에서 제공하는 동일한 AI 연구 환경을 이용할 수 있다. 그룹장이 생성한 AI 연구 환경은 모든 그룹 멤버들이 같은 환경으로 구성되어 동일한 빅데이터 및 AI 알고리즘을 사용할 수 있는 환경으로 사용할 수 있으며, 활용 사례로는 그룹으로 사용해야하는 경우 또는 AI 수업을 진행할 경우 사용될 수 있다.

계산 모듈에서 제공되는 기능 중 분산되어 있는 서버들의 컴퓨팅 리소스들을 통합하여 사용하기 위한 분산 컴퓨팅 서비스를 지원한다. 분산 컴퓨팅 서비스는 Kubeflow^[24]라는 Kubernetes 기반의 오픈소스를 이용하여 수행할 수 있다. Kubeflow에는 분산 컴퓨팅을 수행하는 TF-Operator라는 기능이 존재하며, TF-Operator의 동작 과정은 그림 10과 같다.

분산 컴퓨팅을 수행하기 위해서는 여러 노드에 존재하는 GPU를 여러 컨테이너에서 할당 받아야하기 때문에 Kubernetes의 Job을 이용하여 수행된다. Kubernetes의 Job은 하나 이상의 컨테이너를 생성할 수 있고, 한번 실행하고 종료되는 성격을 갖는 작업에 사용될 수 있다. Job을 사용하면 여러 작업을 병렬로 실행할 수 있기 때문에 여러 노드에 걸쳐 있는 GPU 들을 컨테이너에 할당하고, 할당된 컨테이너들은 미리

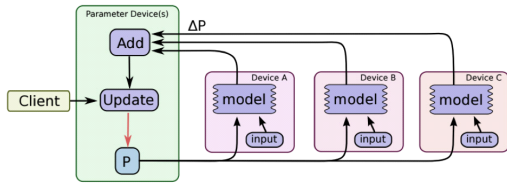


그림 10. 분산 컴퓨팅 동작 과정
Fig. 10. Distributed Computing Operation process

생성하여 배포된 인공지능 수행을 위한 소스코드를 이용하여 입력되는 데이터들을 처리하는 동작과정을 반복적으로 수행한다. Kubeflow에서 실제로 TF-Operator 시스템을 수행하게 되면 PS(Parameter server) 컨테이너와 Worker 컨테이너로 구성되어 있다. PS는 AI 모델이 수행되는 Worker로부터 수행된 기울기 값들을 전달 받아 Update를 진행한 뒤 다시 Worker로 데이터를 분배해주는 시스템이며, 하나의 AI 프로그래밍 작업을 관리해주는 역할을 수행한다. Worker는 데이터를 받아서 AI 모델을 실제로 트레이닝하여 결과를 도출하는 역할을 수행한다.

세 번째 모듈은 저장 모듈이다. 저장 모듈의 경우 300TB의 스토리지 서버를 이용하여 구성하였으며, Ceph^[25]과 Rook^[26]을 이용하여 스토리지를 구성한다. Ceph 스토리지는 Block 스토리지, Object 스토리지, 파일 시스템을 제공하는 소프트웨어 정의 스토리지로써, 오픈소스로 제공되며, 하드웨어 종속성 없이 어떤 저장장치에 사용할 수 있으며, 일관된 성능과 대규모 확장성을 보장하는 기술이다. 또한 다른 스토리지 기술들과도 연계가 가능하여 최근 많이 사용되는 스토리지 솔루션이다. Rook은 Ceph 스토리지와 Kubernetes를 연결하기 위한 오케스트레이터 역할을 수행한다. 그림 11은 Rook과 Ceph 그리고 Kubernetes의 시스템 연계도를 보여준다. 실제로 KAFE 인증을 통해 들어온 사용자가 필요한 만큼의 스토리지를 Kubernetes로 요청하면, PV(Persistent Volume)과 PVC(Persistent Volume Claim)을 생성하고 Rook을 통해 Ceph에 실제 스토리지가 생성된다. 생성된 스토리지는 Kubernetes의 PVC를 통해 컨테이너에 연계되며, 연계된 스토리지는 하나의 리눅스 서버와 같이 사용할 수 있다. Ceph을 통해 생성되는 스토리지는 File System과 Object 스토리지 두 가지 형태로 구성되고 있으며, File System의 경우 사용자의 인증을 통해 스토리지를 다른 사용자의 컨테이너에 연결할 수 있으며, Object Storage를 사용하는 경우에는 Bucket을 이용하여 사용자 간 인증을 통해 데이터

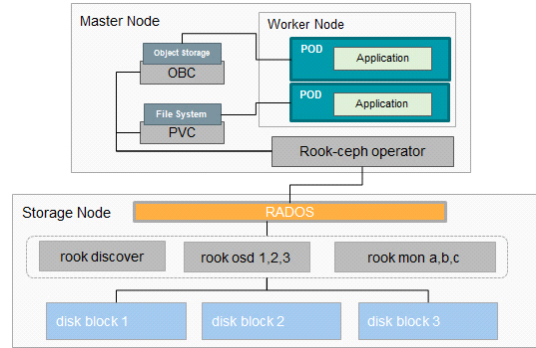


그림 11. Rook Ceph 시스템 구성도
Fig. 11. Rook Ceph System Diagram

를 전송할 수 있는 기능을 가지고 있다.

네 번째 모듈은 모니터링 모듈이다. 모니터링 모듈은 사용자들이 현재 사용하고 있는 시스템들의 리소스 사용량을 확인하기 위해 만들었으며, 모니터링 시스템을 구성하기 위해 Grafana^[27]와 Prometheus^[28]를 이용하여 구성하였다. Kubernetes에서 제공되는 리소스들의 사용 정보를 Prometheus로 보내지고, 보내진 정보들은 인덱스 형태로 저장되어 Grafana를 통해 그래프로 볼 수 있다. 그래프를 통해 볼 수 있는 정보들은 노드들의 GPU, CPU, Memory, Disk, Network 등의 사용량을 확인할 수 있다. 그림 12은 Prometheus와 Grafana를 통해 생성된 모니터링 그래프들이다.



그림 12. Prometheus와 Grafana를 이용한 모니터링 시스템
Fig. 12. Monitoring System using Prometheus and Grafana

IV. 적용 사례

본 장에서는 국내 연구 기관들 중 ScienceDMZ을 구축한 사례에 대해 설명하며, ScienceDMZ의 구축 전과 후에 대한 성능을 비교한다. 국내에서 ScienceDMZ을 구축한 사례들 중 가장 대표적인 기관은 APEC기후센터이다. APEC기후센터는 국가기상수퍼컴퓨터센터의 슈퍼컴퓨터를 활용하여 기후의 변화 모델링을 연구하기 위해 데이터를 업로드 및 다운로드하고 있다. 두 기관 간에 데이터 전송 성능을 측정

표 1. CPU와 GPU 싱글 컴퓨팅 실험 환경 비교 표
Table 1. CPU and GPU Single Computing experiment environment comparison table

| 실험환경 | 싱글 CPU 컴퓨팅 | 싱글 GPU 컴퓨팅 |
|-----------|------------|------------|
| 사용 개수 | 1개 | 1개 |
| 데이터셋 | CIFAR-10 | CIFAR-10 |
| 최적화 알고리즘 | Adam | Adam |
| 딥러닝 모델 | CNN | CNN |
| Epoch 수 | 50 | 50 |
| Batchsize | 600 | 600 |

을 수행하여 ScienceDMZ 구축 전과 후의 성능을 비교한다²⁹⁾. 전송 프로토콜로 기존에 사용하던 FTP와 비교를 위해 사용한 SCP로 테스트를 수행하였으며, 실질적으로 사용하고 있는 데이터를 활용하여 201개의 기후 자료(약 101.62MB)를 사용하여 성능을 측정하였다. ScienceDMZ 구축 전 FTP를 사용하여 데이터 전송 시 평균 13.5MB/s의 성능을 보였으며, SCP를 사용하여 데이터 전송 시 평균 24MB/s의 성능을 보였다. 이 성능은 네트워크의 불안정한 네트워크 품질을 보인다고 볼 수 있다. 이는 APEC기후센터 내부의 보안장비의 접근 제한 또는 방화벽 정책으로 인해 패킷이 Redirecting 되는 이유라고 볼 수 있다. ScienceDMZ 구축 후 성능 측정을 수행하였을 때 SCP를 사용하여 데이터 전송 시 속도는 48~96MB/s로 나타났으며, Globus Online을 사용하여 전송을 측정하였을 때 성능은 109MB/s의 성능을 보였다. 사용하고 있는 대역폭 대비 88% 이상의 성능을 보이고 있으며, ScienceDMZ 구축 전 후의 성능 대비 4배 이상의 성능 향상을 보였다.

사례들 중 또 다른 대표적인 기관은 한국핵융합에너지연구원이다. 한국핵융합에너지연구원은 초진도 핵융합연구장치인 KSTAR을 이용하여 국내 핵융합에너지 개발을 선도하고 있다. KSTAR로 부터 생산되는 데이터는 국내 및 국제적으로 공동 연구를 진행하고 있으며, 국내 공동 연구기관으로는 KAIST, 포항공대, 서울대 등이 있으며, 국제적으로는 미국 동부의 ORNL, PPPL, GA 등의 기관들과 공동연구를 진행 중에 있다. 기존의 네트워크 구조는 많은 보안장비들(Firewall, ISP, DDos 방어 등)로 구성되어 있던 네트워크 구조를 ScienceDMZ 구조로 변경하여 데이터를 전송하였다. 기존의 네트워크 구성에서 데이터를 전송하였을 때 데이터 전송 성능은 120Mbit/s이며, ScienceDMZ 네트워크 구성 후 DTN 서버를 통해 데

이터를 전송할 경우 데이터의 전송 성능은 9000Mbit/sec의 성능을 보였다³⁰⁾.

두 기관 모두 ScienceDMZ을 구축 후 큰 성능 차이를 보이고 있다. 실제로 ScienceDMZ를 구축하고 있는 회원 기관들은 늘어나고 있지만, 실질적으로 해외 또는 기관과 기관 간에 데이터를 전송하는 사례가 현재는 많지 않다. 하지만 추후 인공지능 연구가 더 활성화된다면, 점점 그 사례는 늘어날 것이라고 생각한다.

V. 실험

본 장에서는 R&E Together 플랫폼에서 두 가지의 실험을 통해 본 플랫폼의 성능을 보이고자 한다. 현재 구축된 AI 오케스트레이션 시스템에서 DTN 서버의 GPU와 CPU를 이용한 AI 컴퓨팅 속도를 비교하여 시스템의 성능을 보이고자한다. 또한 싱글 컴퓨팅, 병렬 컴퓨팅, 분산 컴퓨팅의 성능을 비교하여 개발된 시스템을 사용하였을 때 나타는 성능을 확인하고자 한다.

AI 오케스트레이션 시스템에서 싱글 CPU, GPU의 컴퓨팅 성능을 첫 번째로 비교하여 성능적인 차이를 보고자 한다. 실험 방법은 데이터 셋의 경우 CIFAR-10을 사용한다. CIFAR-10 데이터 셋은 10개의 레이블 airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck으로 구성된 32x32 크기의 이미지로 구성된 데이터 셋이다. 전체 데이터 셋의 수는 6만개로 트레이닝 셋은 5만개이며 테스트 셋은 1만개로 구성되어 있다. CIFAR-10 데이터 셋을 CNN과 Fully Connected 조합의 모델을 이용하여 성능을 평가할 때, 트레이닝 시키는데 걸리는 시간을 비교한다. 실험을 위해 epoch라는 용어를 사용하는데, 전체 데이터 셋에 대해 한 번의 학습 과정을 완료한 경우를

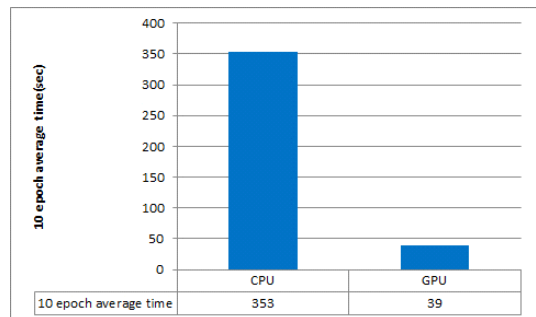


그림 13. CPU와 GPU의 싱글 컴퓨팅 성능 분석
Fig. 13. Single computing performance analysis of CPU and GPU

1epoch라고 말한다. 본 실험에서는 10 번의 epoch를 10번 수행한 결과의 평균을 이용하여 성능을 비교한다. 싱글 CPU, GPU 실험 환경을 비교한 표는 표 1과 같다. CPU와 GPU의 싱글 컴퓨팅을 10번의 epoch를 수행하였을 때 나타나는 그래프는 그림 13와 같다. 10epoch를 10번의 실험을 수행한 값의 평균을 구하면 CPU의 경우 평균 353초가 걸리고, GPU의 경우 평균 39초가 걸린다. CPU보다 GPU를 사용하여 컴퓨팅을 수행하였을 때 약 10배의 성능을 보인다는 것을 확인할 수 있다.

다음으로 AI 오케스트레이션 시스템에서 CPU와 GPU의 병렬 컴퓨팅을 수행하였을 경우를 비교한다. 실험 환경을 비교한 표는 표 2과 같다. 또한 분산 컴퓨팅을 수행하였을 경우의 성능을 GPU 병렬 컴퓨팅과 비교한다. 싱글 컴퓨팅의 실험과 똑같이 CIFAR-10의 데이터 셋을 이용하였으며, 50개의 epoch 전체를 10번 수행하였을 때 걸리는 시간의 평균을 비교한다. CPU 병렬 컴퓨팅을 수행한 결과는 그림 14과 같다. 실험은 1개부터 6개까지의 CPU 병렬 컴퓨팅을 수행하였으며, 3개의 CPU 부터 결과를 보이는 이유는 1개와 2개의 CPU를 사용한 경우 성능의 결과를 나타낼 수 없었기 때문이다. 6개의 CPU를 사용하여 결과를 도출하였을 때 3503초가 걸리는데 대략적으로 CPU 컴퓨팅의 경우 병렬로 사용하더라도 빠른 결과를 내기는 쉽지 않다는 것을 확인할 수 있다.

GPU 병렬 컴퓨팅의 경우 하나의 서버에서 가지고 있는 GPU의 수가 4개까지 가능하기 때문에 1개부터 4개까지 사용하여 병렬 컴퓨팅 성능을 비교한다. GPU의 평균 학습 시간을 비교한 그래프는 그림 15와 같다. 1개의 GPU를 이용하여 평균 학습 시간은 180초이고, 2개의 GPU를 이용할 경우 평균 학습 시간은 119초이고, 3개의 GPU를 이용할 경우 평균 학습 시간은 107초이고, 4개의 GPU를 이용할 경우 평균 학습

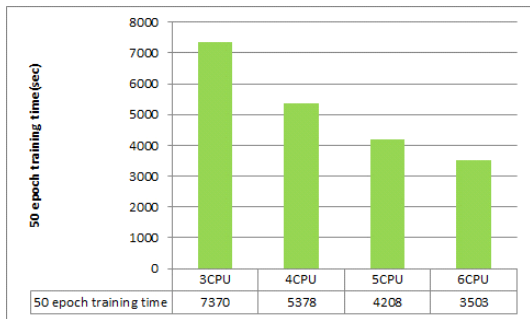


그림 14. CPU의 병렬 컴퓨팅 성능 분석
Fig. 14. Multi-Computing performance analysis of CPU

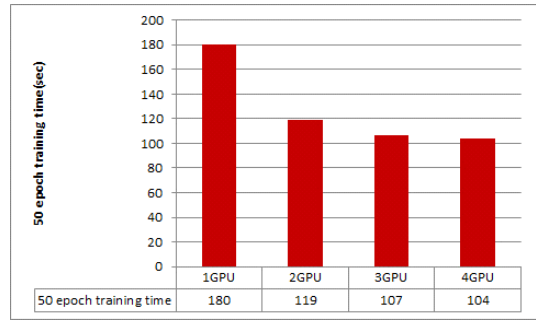


그림 15. GPU의 병렬 컴퓨팅 성능 분석
Fig. 15. Multi-Computing performance analysis of GPU

시간은 104초이다. 병렬컴퓨팅을 수행할 경우 CPU의 성능과 비교해보면 확연히 빠르다는 것을 확인할 수 있었으며, GPU의 수가 증가 할수록 높은 성능을 보인다는 것을 확인할 수 있다. 하지만 GPU의 수가 증가할수록 시간의 축소율이 줄어드는 것 또한 확인할 수 있었다.

분산 컴퓨팅을 수행하는 경우 각 노드별로 1개의 GPU가 남아 있다는 가정 하에 실험을 수행하며, GPU가 구축되어 있는 전체 노드가 5개이기 때문에 1개부터 5개 까지 분산 컴퓨팅을 수행하여 성능을 비교하고자 한다. 본 시스템의 경우 실험을 수행하기 위한 GPU의 모델이 각 노드마다 다르기 때문에 성능의 결과가 모델마다 조금씩 차이가 있다. 따라서 10번의 실험을 수행한 평균 시간을 계산하여 결과를 도출한다. 분산컴퓨팅은 병렬컴퓨팅과 같은 알고리즘과 같은 실험 방법과 같은 데이터를 이용하여 실험을 수행하였으며, 알고리즘을 수행하는데 있어 차이점은 Batch 사이즈의 차이만 있을 뿐이다. 분산 컴퓨팅과 병렬 컴퓨팅을 구성한 실험 환경을 비교해 보면 표 3와 같으며, 분산컴퓨팅의 성능 그래프는 그림 16와 같다. 1개의 GPU를 수행하였을 때, 평균 학습 시간은 460초이며, 2개의 GPU를 이용할 경우 평균 학습 시간은 168초이며, 3개의 GPU를 이용할 경우 평균 학습 시간은 87초이며, 4개의 GPU를 이용할 경우 평균 학습 시간은 76초이며, 5개의 GPU를 이용할 경우 평균 학습 시간은 68초로 확인된다.

GPU의 수를 적게 사용할 경우 분산 컴퓨팅의 성능이 더 낮게 나오지만, GPU의 수가 증가할수록 분산 컴퓨팅 성능이 더 높게 나온다는 것을 확인할 수 있었다.

본 실험에서 빅데이터를 전송하기 위해서는 ScienceDMZ의 구성이 필수적이라는 것을 확인할 수 있었다. 또한 본 논문에서 개발한 AI 오케스트레이션

표 2. CPU와 GPU 병렬 컴퓨팅 실험 환경 비교 표
Table 2. CPU and GPU Parallel Computing experiment environment comparison table

| 실험환경 | 병렬 CPU 컴퓨팅 | 병렬 GPU 컴퓨팅 |
|-----------|------------|------------|
| 사용 개수 | 1~6 개 | 1~4 개 |
| 데이터셋 | CIFAR-10 | CIFAR-10 |
| 최적화 알고리즘 | Adam | Adam |
| 딥러닝 모델 | CNN | CNN |
| Epoch 수 | 50 | 50 |
| Batchsize | 600 | 1200 |

표 3. 분산 컴퓨팅과 병렬 컴퓨팅 실험 환경 비교 표
Table 3. Distributed Computing and Parallel Computing experiment environment comparison table

| 실험환경 | 병렬 컴퓨팅 | 분산 컴퓨팅 |
|-----------|----------|----------------|
| GPU 사용 개수 | 1~4 개 | 1~5 개 |
| GPU 사용모델 | TAITAN | TAITAN, 2080Ti |
| 데이터셋 | CIFAR-10 | CIFAR-10 |
| 최적화 알고리즘 | Adam | Adam |
| 딥러닝 모델 | CNN | CNN |
| Epoch 수 | 50 | 50 |
| Batchsize | 1200 | 600 |

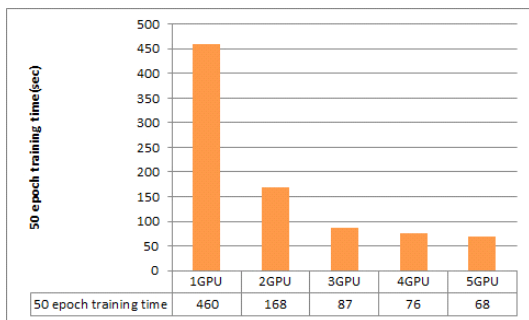


그림 16. GPU의 분산 컴퓨팅 성능 도출
Fig. 16. Distributed Computing performance analysis of GPU

시스템을 통해 CPU와 GPU의 성능을 비교하였으며, GPU의 성능이 CPU 보다 높다는 것을 확인할 수 있었다. GPU의 경우 병렬컴퓨팅과 분산컴퓨팅을 비교해 보면 GPU의 수가 낮을 경우 병렬컴퓨팅 성능이 더 높다는 것을 확인할 수 있었고, GPU의 수가 증가할수록 분산컴퓨팅의 성능이 더 높다는 것을 확인하

였다.

VI. 결 론

국내·외 AI를 이용한 연구들이 활발히 진행되고 있고, AI를 수행하기 위해 거대한 데이터를 고속으로 전송하기 위한 노력들이 이루어지고 있다. 이에 따라 KREONET에서는 R&E Together라는 프로젝트를 통해 출연연구소의 전산 담당자들과 빅데이터를 데이터의 손실 없이 빠르게 전송할 수 있는 기술인 ScienceDMZ를 함께 구축하고 ScienceDMZ 인프라 위에 AI 오케스트레이션 시스템을 개발하여 하나의 거대한 AI 연구를 위한 서비스를 구축하는 프로젝트를 진행 중에 있다.

R&E Together 프로젝트는 ScienceDMZ 네트워크 구축을 통해 빅데이터 고속도로를 구축하고, DTN 서버를 이용하여 데이터를 고속으로 전송할 뿐만 아니라 컴퓨팅이 가능한 DTN 서버를 설치하여 전송을 담당할 때는 전송을 위한 CPU를 사용하고 컴퓨팅을 사용하기 위해서는 GPU를 사용하여 AI 작업을 수행할 수 있도록 구성한다. 이러한 DTN 서버 위에 AI 오케스트레이션 플랫폼을 개발하여 AI 연구 환경을 제공하여 자유롭게 사용자들이 AI 연구를 수행할 수 있도록 돕는다.

국내의 ScienceDMZ 구축 기관들의 구축 전후의 성능 비교를 통해 ScienceDMZ의 성능을 확인한다. ScienceDMZ 인프라를 이용하여 데이터를 전송하면, 기존 네트워크 구성의 대역폭 성능 대비 80% 이상의 성능이 증가 하였다는 것을 실험을 통해 확인할 수 있으며, AI 오케스트레이션 시스템을 통해 본 시스템의 컴퓨팅 능력에 대해 비교를 해본 결과 CPU와 GPU의 성능 비교를 통해 GPU를 사용할 때 대략 9배의 시간 차이를 보였으며, GPU의 분산 컴퓨팅과 병렬 컴퓨팅의 경우 1개와 2개의 GPU를 사용하였을 때는 병렬 컴퓨팅의 성능이 높게 나왔으며 3개 이상의 GPU를 사용하였을 때 분산 컴퓨팅의 성능이 높게 나온 것을 확인할 수 있었다.

향후 본 시스템은 미국의 PRP에서 개발한 AI 연구 환경 제공 시스템인 Nautilus 시스템과 Admiralty^[31]를 이용하여 Federation을 수행하여 R&E Together의 컴퓨팅 리소스와 NRP의 컴퓨팅 리소스를 같이 사용할 수 있도록 구성할 것이다. 또한 KISTI의 슈퍼컴퓨터 5호기와 연계하여 사용자들이 슈퍼컴 컴퓨팅 자원을 이용하여 컴퓨팅 리소스의 부족함을 해소할 수 있도록 구성하고자 한다.

References

- [1] IDC, "Korea Artificial Intelligence Forecast," 2019 - 2023," 2020.
- [2] CIO, "Edge Computing Market to 2024 is growing steady at 26.5%," 2019.
- [3] H.M. Park, T.H. Hwang, "Changes and Trends in Edge Computing Technology," *KICS Inf. and Commun. Mag.*, vol. 36, no. 2, pp. 41-47, 2019.
- [4] E. Lee, "Edge computing for improving the security of national defense cloud computing," M.S. Thesis, Graduate School of Engineering, Hanyang Univ., 2020.
- [5] E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski, "The ScienceDMZ: A network design pattern for data-intensive science," *Scientific Programming*, vol. 22, no. 2, pp. 173-185, 2014.
- [6] J.-H. Lee, "Edge Computing Resource Management Technology Trends," in *Proc. KIEES*, vol. 30, no. 3, pp. 34-40, 2019.
- [7] *KREONET*, <https://www.kreonet.net/>, 11, 2020.
- [8] *PRP(Pacific Research Platform)*, <https://pacificresearchplatform.org/>, 10, 2020.
- [9] *NRP (National Research Platform)*, <https://pacificresearchplatform.org/nrp/>, 10, 2020.
- [10] *DTN (Data Transfer Node)*, <https://pacificresearchplatform.org/nrp/>, 10, 2020.
- [11] *APRP (Acia Pacific Research Platform)*, <http://aprnet.org/>, 11, 2020.
- [12] *GRP (Global Research Platform)*, <http://www.theglobalresearchplatform.net>, 11, 2020.
- [13] E. Balas and A. Ragusa, "SciPass: a 100Gbps capable secure science DMZ using OpenFlow and Bro," in *Supercomputing 2014 Conf.*, 2014.
- [14] J. Moon, J. Kwak, W. Hong, K. Kim, S. Lee, D. Kim, Y. Kim, and K. Yu, "A study on ScienceDMZ construction for high speed transfer of science big data," *KNOM Rev.*, vol. 22, no. 2, pp. 12-21, 2019.
- [15] P. Thinakaran, J. R. Gunasekaran, B. Sharma, M. T. Kandemir, and C. R. Das, "Kube-knots: Resource harvesting through dynamic container orchestration in gpu-based datacenters," in *2019 IEEE Int. Conf. Cluster Computing*, pp. 1-13, 2019:
- [16] J. Oh, S. Kim, and Y. Kim, "Toward an adaptive fair GPU sharing scheme in container-based clusters," in *2018 IEEE 3rd Int. Wkshps. Foundations and Applications of Self* Systems (FAS* W)*, pp. 79-85, 2018.
- [17] Y. Peng, Y. Bao, Y. Chen, C. Wu, and C. Guo, "Optimus: An efficient dynamic resource scheduler for deep learning clusters," in *Proc. Thirteenth EuroSys Conf.*, pp. 1-14, 2018.
- [18] P. Xu, S. Shi, and X. Chu, "Performance evaluation of deep learning tools in docker containers," in *IEEE BIGCOM*, pp. 395-403, 2017.
- [19] *Globus Online*, <https://www.globus.org/>, 09, 2020.
- [20] *kubernetes*, <https://kubernetes.io/>, 09, 2020.
- [21] J. Jo, H. Jang, J. Kong, and Y. H. Chae, "Federated IAM service of KAFE identity federation," *J. KICS*, vol. 43, no. 12, pp. 2200-2214, 2018.
- [22] *KAFE*, <https://www.kafe.or.kr>, 11, 2020.
- [23] *Jupyter*, <https://jupyter.org/>, 09, 2020.
- [24] *Kubeflow*, <https://www.kubeflow.org/>, 09, 2020.
- [25] *Ceph*, <https://ceph.io/>, 09, 2020.
- [26] *Rook*, <https://rook.io/>, 09, 2020.
- [27] *Grafana*, <https://grafana.com/>, 09, 2020.
- [28] *Prometheus*, <https://prometheus.io/>, 09, 2020.
- [29] J. Park, S. Kim, and M. Noh, "*KREONET ScienceDMZ Construction Report*," KISTI, 2018.
- [30] J. S. Park, Y. K. Oh, U. Y. Lee, T. H. Lee, S. H. Son, and D. J. LEE, "*In real time international collaborative environment configuration for large scaled experimental data of KSTAR*," 2014 Major Business Report, pp. 10-13, Mar. 2015.
- [31] *Admiralty*, <https://admiralty.io/>, 09, 2020.

김 기 현 (Ki-Hyeon Kim)



2016년 2월 : 강원대학교 컴퓨터
과학과 석사
2017년 2월~현재 : 강원대학교
컴퓨터과학과 박사과정
2016년 2월~현재 : 한국과학기술
술정보연구원(KISTI) 연구원

<관심분야> ScienceDMZ, SDN/NFV, Network
Virtualisation

박 병 연 (Byung-yeon Park)



2004년 : 국립공주대학교 교육정
보학 석사
2010년 : 국립공주대학교 바이오
정보학(정보보호전공) 석사
1990년~현재 : 한국과학기술정
보연구원 책임연구원

<관심분야> ScienceDMZ, 네트워크 QoS, 클라우드
컴퓨팅, TCP 성능향상

문 정 훈 (Jeong-hoon Moon)



1997년 : 경일대학교 컴퓨터공학
과 학사
1999년 : 경북대학교 컴퓨터공학
과 석사
2000년~현재 : 한국과학기술정
보연구원 책임연구원

<관심분야> ScienceDMZ, 네트워크 QoS, 가상화,
SDN, 클라우드 컴퓨팅, TCP 성능향상

석 우 진 (Woojin Seok)



1996년 2월 : 경북대학교 컴퓨터
공학과 학사
2001년 1월 : UNC Chapel Hill
Computer Science M.S.
2008년 8월 : 충남대학교 컴퓨터
공학과 박사
1998년~현재 : KISTI 근무, (현)
과학기술연구망센터장

2021년~현재 : 한국통신학회 KNOM 연구회 위원장
2020년~현재 : SDN/NFV 포럼 PoC 분과장
2012년~2020년 : UST 겸임교원

<관심분야> TCP 프로토콜, 양자암호통신, 비면허대역
무선통신

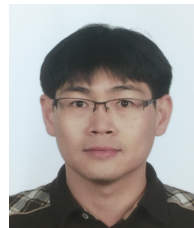
권 우 창 (Woo-Chang Kwon)



2011년 2월 : 안동대학교 컴퓨터
공학과 석사
2016년 9월~현재 : 한남대학교
컴퓨터공학과 박사과정
2015년 9월~현재 : 한국과학기술
술정보연구원(KISTI) 연구원

<관심분야> ScienceDMZ,
Network Automation

홍 원 택 (Wontaek Hong)



1998년 : 성균관대학교 정보공학
과 학사
2000년 : 성균관대학교 전기전자
및 컴퓨터공학과 석사
2019년 : 성균관대학교 전기전자
컴퓨터공학과 박사
2000년~2002년 : (주)콤텍시스템
기술연구소 연구원

2002년~현재 : 한국과학기술정보연구원 책임연구원

<관심분야> 망 성능 분석, 대용량 데이터 전송 프로토
콜, 고성능 네트워크 구조, 클라우드 네트워킹

이 상 권 (Sang-kwon Lee)



2015년 : 충북대학교 컴퓨터공학과 학사
2017년 : 충북대학교 컴퓨터공학과 석사
2017년~현재 : 과학기술연합대학원대학교 과학기술정보과학학과 박사과정

<관심분야> 네트워크 성능 테스트, 네트워크 튜닝, 클러스터 컴퓨팅, IoT 네트워크

최 재 현 (Jae-heon Choi)



2018년 : 충남대학교 정보통신공학 학사
2018년~현재 : 한국과학기술원 기술원
<관심분야> SDN, 네트워크 보안, 무선통신

조 진 용 (Jinyong Jo)



2003년 : 광주과학기술원 정보통신공학과 석사
2013년 : 광주과학기술원 정보통신공학과 박사
2003년~현재 : 한국과학기술정보연구원 책임연구원
2016년~현재 : eduGAIN 운영그룹 위원

<관심분야> Trust & Identity

김 대 겸 (Dae-Kyeom Kim)



1996년 : 한밭대학교 전자계산학과 학사
2002년 : 배재대학교 정보통신학과 석사
1996년~현재 : 한국생명공학연구원 책임기술원

<관심분야> Hypervisor, SDN, Zero Trust, SOAR

윤 태 진 (Taejin Yoon)



1987년 2월 : 우송정보대학교 전산학과 학사
1989년~현재 : 한국과학기술원 선임기술원
<관심분야> ScienceDMZ, 네트워크매니지먼트, 네트워크 보안

황 등 아 (Dong-Ah Hwang)



1992년 : 대전대학교 전자계산학과 학사
1996년 : 충남대학교 전자계산교육학과 석사
1996년~현재 : 한국생명공학연구원 책임기술원
<관심분야> 경영정보, 데이터베이스, EA

최 선 웅 (Sun-Woong Choi)



2002년 : 전남대학교 정보통신공
학 학사
2010년~현재 : 한국생명공학연
구원 선임기술
<관심분야> 서버보안, 네트워크
보안, 가상화, 클라우드 컴퓨
팅

김 기 형 (KiHyoung Kim)



2006년 2월 : 충북대학교 컴퓨터
공학 학사
2006년~2012년 : Continental
Automotive Korea Ltd. IT
dept. 근무
2012년 5월~현재 : 한국항공우
주연구원 선임기술원
<관심분야> 클라우드 컴퓨팅, 네트워크 가상화, 네트워
크 보안

김 종 호 (Jongho Kim)



1998년 : 군산대학교 정보통신공
과 학사
2000년 : 광주과학기술원 정보통
신공학과 석사
2003년~2007년 : 한국항공우주
연구원 선임연구원
2016년 정보통신기술사
2007년~현재 : 한국한의학연구원 책임기술원
<관심분야> 네트워크, 5G, Private 5G, 네트워크 보안

정 병 훈 (Byonghoon Chong)



2001년 : 부산대학교 컴퓨터공학
학사
2003년 : 부산대학교 컴퓨터공학
석사
2003년~2009년 : LG전자 MC연
구소 선임연구원
2009년~현재 : 한국항공우주연
구원 선임기술원
<관심분야> 분산환경 컴퓨팅, 초고속 유무선 네트워크

김 준 엽 (Jun-Yeop Kim)



1996년 : 중앙대학교 전자계산학
과 학사
1998년 : 중앙대학교 컴퓨터공학
과 석사
1998년~현재 : 한국항공우주연
구원 책임연구원
<관심분야> SDN, 클라우드 컴
퓨팅, 정보화전략계획, 정보화수준평가, 정보보안

이 도 섭 (Do-Sub Lee)



2005년 : 배재대학교 정보통신
공학 학사
2001년~현재 : 한국핵융합에너
지연구원 책임기술원
<관심분야> 802.1x, SDN, HCI,
ScienceDMZ, 클라우드 컴퓨
팅, 네트워크 보안

유 영 근 (Young-Geun Yu)



2009년 : 한밭대학교 컴퓨터공학과 석사
2019년 : 공주대학교 컴퓨터공학과 박사
1996년~현재 : 한국화학연구원 선임기술원
<관심분야> 정보보안, 가상화, SDN

천 재 흥 (Jae-Hong Cheon)



2007년 : 숭실대학교 정보과학대학원 정보보안학과 석사
2020년 : 호서대학교 벤처대학원 융합과학기술학과 박사
1997년~현재 : 한국환경정책·평가연구원 책임전문원
<관심분야> 정보보안, 가상화, 클라우드, VDI, ScienceDMZ, 네트워크

박 은 숙 (Eun-Sook Park)



1988년 : 한국방송통신대학교 전자계산학과 학사
1986년~1995년 : (주)케이티브이 연구소
1996년~현재 : 한국화학연구원 책임기술원
<관심분야> 정보시스템 개발 및 기획