Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.32, NO.6, Dec. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.6.1127

GPUDirect RDMA 기반의 고성능 암호 분석 시스템 설계 및 구현*

이 석 민,[1†] 신 영 주[2‡]
[1,2]고려대학교 정보보호대학원 (대학원생, 교수)

Design and Implementation of High-Performance Cryptanalysis System Based on GPUDirect RDMA*

Seokmin Lee,[1†] Youngjoo Shin[2‡]
[1,2]School of Cybersecurity, Korea University (Graduate student, Professor)

요 약

GPU의 병렬 연산을 활용한 암호 분석 및 해독 기술은 암호 분석 시스템의 연산 시간을 단축하는 방향으로 연구되었다. 해당 연구들은 하나의 GPU에서 암호 분석 연산의 속도를 향상시키기 위해 코드를 최적화하거나 또는 단순히 GPU의 수를 늘려 병렬 연산을 강화하는 것에 집중되어 있다. 하지만 다량의 GPU를 데이터 전송에 대한 최적화 없이 사용하는 것은 하나의 GPU를 사용하는 것보다 더 긴 데이터 전송 지연 문제를 발생시키고, 암호 분석 시스템의 전체적인 연산 시간 증가를 야기한다. 이에, 본 논문은 딥러닝 또는 HPC 연구 분야의 GPU Clustering 환경에서 고성능 데이터 처리를 위해 활용되는 GPUDirect RDMA 및 관련 제반 기술들을 조사 및 분석한다. 그리고 해당 기술들을 활용한 고성능 암호 분석 시스템 설계 방법들을 제안한다. 더 나아가, 해당 설계를 기반으로 Password Cracking, GPU Reduction을 활용한 암호 분석 시스템 구현 방법에 대해 제시한다. 최종적으로, GPUDirect RDMA 기술 적용으로 구현된 암호 분석 시스템에 대해서 암호 분석 작업 성능 향상의 실증을 통해 제안한 시스템에 대한 기대효과를 제시한다.

ABSTRACT

Cryptographic analysis and decryption technology utilizing the parallel operation of GPU has been studied in the direction of shortening the computation time of the password analysis system. These studies focus on optimizing the code to improve the speed of cryptographic analysis operations on a single GPU or simply increasing the number of GPUs to enhance parallel operations. However, using a large number of GPUs without optimization for data transmission causes longer data transmission latency than using a single GPU and increases the overall computation time of the cryptographic analysis system. In this paper, we investigate GPUDirect RDMA and related technologies for high-performance data processing in deep learning or HPC research fields in GPU clustering environments. In addition, we present a method of designing a high-performance cryptanalysis system using the relevant technologies. Furthermore, based on the suggested system topology, we present a method of implementing a cryptanalysis system using password cracking and GPU reduction. Finally, the performance evaluation results are presented according to demonstration of high-performance technology is applied to the implemented cryptanalysis system, and the expected effects of the proposed system design are shown.

Keywords: Cryptanalysis system, GPUDirect RDMA, Remote MPI clustering

Received(10. 04. 2022), Modified(11. 24. 2022), Accepted(12. 02. 2022)

* 이 논문은 2021년 정부(방위사업청)의 재원으로 국방과학연구소의 지원을 받아 수행된 연구임(UD210027XD).

† 주저자, leeseokmin@korea.ac.kr
‡ 교신저자, syoungjoo@korea.ac.kr(Corresponding author)

I. 서 론

암호화된 데이터를 분석 및 해독하는 방법에 대한 다양한 연구들이 진행되었다. 대표적인 방법으로는 암호문을 만들 때 사용되는 조합 가능한 모든 평문 값을 암호화하고, 찾고자 하는 평문 값과 1 대 1 대응되는 암호문을 찾아 목표하는 비밀번호 값을 찾는 전수조사 방법이 있다.

이전 연구들에서 전수조사에 대한 시간을 단축하기 위해 GPU를 활용하여 CUDA 프로그래밍을 통한 전수조사 암호 분석을 진행하였다[1]. 또는, 해당 암호 분석 코드를 최적화 및 경량화하는 것에 집중하여 전수조사의 전체 연산 시간을 단축하고자 하였다[2]. 더 나아가, HPC 분야의 GPU Clustering 개념을 활용해 다량의 GPU를 하나의 서버에서 병렬 연산 처리에 활용하여 전수조사 암호 분석의 전체 연산 시간을 단축하고자 하였다 [3]. 그러나, 시스템에 따라 GPU 자원의 수가 한정적일 뿐만 아니라, 최적화된 GPU Clustering 환경이 아닐 경우, 대용량의 데이터 전송 과정에서 전송 지연 문제를 발생시켜 하나의 GPU만을 활용해 암호 분석을 진행하는 것보다 더 긴 연산 시간을 야기할 수 있다.

이에, 본 논문에서는 딥러닝 및 HPC 분야의 GPU Clustering 환경에서 활용되는 불필요한 데이터 복사 생략을 통한 데이터 전송 최적화 기술들을 소개 및 분석한다. 로컬 환경의 GPU 메모리와 리모트 환경의 GPU 메모리 사이의 데이터 전송 최적화를 의도하는 GPUDirect RDMA[4] 기술에 대해서 조사 및 분석한다. 그리고 MPI(Message Passing Interface)[5] 기술을 기반으로 로컬 서버와 리모트 서버 간 병렬 프로그래밍을 가능하게 하는 Remote MPI Clustering 기술을 조사한다. 또한 Unified 메모리의 개념을 활용하여 CPU 메모리에서 GPU 메모리로의 복사 과정 없이 하나의 MPI 명령어로 GPU 메모리 간 데이터를 송수신 가능하게 하는 CUDA-Aware MPI 기술에 대해서 분석한다.

그리고 해당 기술들을 활용해, GPU Clustering 환경에서의 고성능 데이터 처리가 가능한 마스터-워커 패턴의 암호 분석 시스템 설계를 제안한다. 더 나아가, 설계된 암호 분석 시스템에서 워커 노드 간 통신 가능성에 따른 시스템 토폴로지들을 설계하며, 설계한 토폴로지들을 기반으로 적용한 암호 분석 기술들의 사례를 제시한다. 그 예시로 Password Cracking, GPU Reduction 기술들을 소개하고 해당 기술들을 제안했던 시스템 토폴로지에 적용하여, 암호 분석 시스템 구현에 대한 방법을 제시한다.

마지막으로, Password Cracking 동작 방식으로 구현한 암호 분석 시스템에 대해서 GPUDirect RDMA 기술의 적용에 따른 성능 향상의 실증으로, 제안된 고성능 기술들을 활용한 암호 분석 시스템 설계에 대해서 기대 효과를 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 고성능 암호 분석 시스템을 구성하는 배경 기술들을 소개한다. 3장에서는 기술들을 활용한 암호 분석 시스템 설계 방법을 제시한다. 4장에서는 설계된 시스템에 암호 분석 기술을 적용하여, 암호 분석 시스템 구현 방법을 제시한다. 5장에서는 구현한 시스템에 대한 성능 평가를 진행한다. 마지막으로, 6장에서는 결론을 기술한다.

II. 배경 지식

2.1 GPUDirect RDMA

본 장에서는, GPUDirect RDMA 기술에 대한 소개와 해당 기술을 지원하는 시스템을 구성하는 방법에 대해 기술한다. 그리고 기술 지원 여부에 따른 성능 평가를 통해 GPUDirect RDMA 기술이 적용된 시스템에 대한 기대효과를 기술한다.

2.1.1 GPUDirect RDMA 개요

GPU에서 연산 처리가 수행될 때 CUDA API를 사용하여 GPU 디바이스 메모리가 활용된다. 이때 연산에 사용할 데이터를 CPU가 사용하는 호스트 메모리로부터 GPU 디바이스 메모리로 복사 후 연산이 수행된다. 다량의 GPU로 구성된 GPU Clustering 환경에서 GPU 간 연산 처리를 수행할 경우, 불필요한 데이터 복사를 줄이기 위해서 매번 호스트 메모리와 상호작용하지 않는다. GPU와 GPU 또는 GPU와 서드파티 디바이스들이 CPU의 호스트 메모리로의 복사 없이 직접적인 통신을 진행하며 이를 가능하게 하는 기술들의 집합을 GPUDirect라 명명한다. GPUDirect 기술 중 네트워크 어댑터인 HCA (Host Channel Adapter)를 활용해 리모트 서버의 GPU 메모리에 직접적으로 접근할 수 있도록 하는 기술을 GPUDirect RDMA라 한다.

2.1.2 GPUDirect RDMA 지원 시스템 구성

GPUDirect RDMA 기술을 지원하는 환경을 구성하기 위해서는 Kepler 아키텍쳐 이상의 GPU 디바이스와 ConnectX-4 이상의 HCA 네트워크 어댑터가 요구된다. 또한 GPU 인식을 위한 Nvidia Driver, 5.0 이상의 CUDA Driver가 요구된다. 그리고 운영체제와 HCA의 상호작용을 위해서 Mellanox에서 제공하는 HCA 전용 Driver를 운영체제의 버전에 맞게 설치가 필요하며, HCA Driver와 GPU 메모리 간에 직접적인 상호작용을 위해서 nv_peer_mem[6] 커널 모듈의 구성이 요구된다.

해당 GPUDirect RDMA 기술을 후술할 MPI의 병렬 프로그래밍 기술과 함께 활용하는 시스템을 구성하기 위해서 gdrcopy[7]와 ucx[8] 라이브러리를 사전에 설치해야 한다. 또한 gdrcopy와 ucx 라이브러리의 설치 경로를 MPI 컴파일러 설치 파일의 파라미터로 설정한 뒤, MPI 컴파일러 설치를 통해서 GPUDirect RDMA 기술과 MPI 병렬 프로그래밍을 함께 사용할 수 있다. 그 이후, mpirun (MPI 컴파일러)의 실행 파라미터로 "-x UCX_IB_GPU_DIRECT_RDMA"를 참으로 설정하고, "-x UCX_TLS" 파라미터 값을 "cuda_copy, gdr_copy" 값의 설정을 통해 GPUDirect RDMA 실행 가능 환경을 구성할 수 있다.

GPUDirect RDMA 기술이 지원되는 시스템 환경이라면 로컬 환경의 GPU 메모리에서 리모트 환경의 GPU 메모리로 직접적인 데이터 송•수신이 가능하다. 하지만 지원하지 않는 환경이라면, 로컬 환경의 GPU 메모리에서 CPU 메모리로 데이터 복사가 이뤄진 이후, HCA를 통해 리모트 환경의 CPU 메모리로 복사가 된다. 그리고 CPU 메모리에서 GPU 메모리로 데이터 복사가 최종적으로 진행된다. 즉, 불필요한 데이터 복사가 최소 두 번 이상 발생된다.

데이터 전송 벤치마킹 프로그램인 perftest를 활용해 GPUDirect RDMA 지원 여부에 따른 데이터 전송 성능 차이를 평가할 수 있다. Fig. 1.와 같이, 전송하는 데이터의 양이 커질수록, 데이터 전송 대역폭(bandwidth)의 크기 차이와 지연시간(latency)의 시간 차이가 약 100배 이상 발생함을 확인할 수 있다.

이에 따라, GPUDirect RDMA 기술을 활용하여 암호 분석 시스템을 구성할 경우, 하나 이상의 GPU 메모리들이 향상된 전송 속도로 데이터를 송•수신하여 향상된 병렬 연산 성능을 기대할 수 있다. 그리고 이를 통해, 전체적인 암호 분석 시스템의 연산 시간에 대한 단축된 시간을 기대할 수 있다.

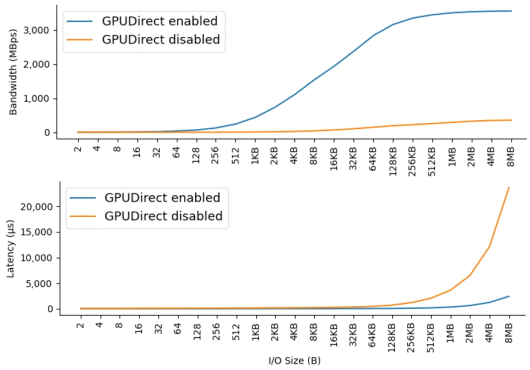

Fig. 1. GPUDirect RDMA performance comparison using perftest

2.2 Remote MPI Clustering

MPI란 병렬 프로그래밍의 대표적인 표준이다. 서로 다른 프로세서 간 공유 메모리를 사용하지 않고, 별도의 메시지라는 메모리를 선언하여 태그 값을 프로세스의 고유값으로 활용해 프로세스들을 구분한다. 그리고 MPI API를 활용하여 프로세서 간 데이터를 송•수신하고 다수의 프로세스에 작업을 할당하여 병렬적으로 작업을 수행해 효율적인 컴퓨터 자원 활용이 가능하다. Remote MPI Clustering 기술은 리모트 서버에게 프로세스를 할당하여 로컬과 리모트 서버가 병렬적으로 동시에 프로그램 실행이 가능하도록 지원한다.

해당 기술을 지원하기 위해서는 ssh-keygen 명령어를 활용해 서버 간 공개키를 교환하여 상대 컴퓨터로 비밀번호 입력 없이 ssh 접속이 가능하도록 설정해야한다. 리모트 서버와 분배하여 실행할 프로세스의 개수를 설정하기 위해서 mpirun의 n 파라미터의 값으로 총 실행할 프로세스의 개수를 설정하고, host 파라미터 값 뒤 세미콜론을 통해 로컬 서버와 리모트 서버에 각각 할당할 프로세스 수를 설정한다. 또한, 리모트 환경의 서버와 HCA 디바이스를 활용한 통신을 수행하기 위해 mpirun 실행 파라미터로 "-x UCX_NET_DEVICES"의 값을 현재 구성된 HCA 디바이스 명으로 설정한다. 그리고 "-mca btl _openib_allow_ib" 파라미터를 참으로 설정하여, 최종적으로 Remote 환경의 MPI Clustering 환경을 구성한다.

2.3 Cuda-Aware MPI

GPU로 연산한 데이터를 MPI API를 활용해서 리모트 환경의 다른 프로세스에게 전송하기 위해서는 GPU의 디바이스 메모리에서 CPU의 호스트 메모리로의 데이터 복사 후 전송이 가능하였다. 하지만 Cuda-Aware MPI 기술을 통해 이러한 불필요한 데이터 복사 과정이 생략된다. CUDA 4.0 이후부터 Unified 메모리의 개념이 등장하게 되면서 호스트 메모리 영역과 디바이스 메모리 영역을 하나의 가상 주소로 접근할 수 있게 되었다. CUDA-Aware MPI에서는 해당 개념을 활용하여 하나의 MPI 데이터 전송 명령어를 통해 GPU의 디바이스 메모리 간 데이터 송•수신을 가능하게 한다.

해당 기술을 지원하기 위해서 OpenMPI 1.7.0 이상의 MPI 컴파일러를 설치해야만 하고, GPUDirect RDMA 기술의 구성 방법과 동일한 방법으로 gdrcopy와 ucx를 설치해야 한다. 또한 리모트 환경의 서버에게 데이터를 전송하는 MPI API의 인자 값으로 Unified 메모리의 주소를 반환하는 cudaMallocManged API의 반환 값을 삽입하여 시스템을 구성해야 한다.

III. 고성능 암호 분석 시스템 설계

3.1 시스템 환경 설정 및 동작 개요

본 연구에서는 고성능 데이터 처리 기술들을 적용한 시스템을 설계 및 구현한다. 해당 설계를 위해 Table 1.에 기술된 하드웨어 자원들을 통해 시스템을 구성하였다. 서버용 Intel CPU를 사용하여 안정적인 시스템 환경 구축이 가능하도록 구성하였고, GPUDirect 기술을 시스템에 구성하기 위해서 Amphere 버전의 A40과, A100 GPGPU를 이용하였다. 또한, GPUDirect RDMA를 활용해 CPU 호스트 메모리로의 복사 없이 데이터를 전송을 위해 ConnectX-5의 HCA를 시스템에 구성하였다.

해당 하드웨어 자원을 활용하여 고성능 기술들을 시스템에 적용하기 위해, 설치한 드라이브와 애플리케이션의 목록은 Table 2.와 같이 기술하였다. GPU와 HCA 디바이스들을 운영체제에서 인식하기 위해서 각각의 전용 드라이브를 설치한다. 그리고 CUDA 프로그래밍을 활용한 병렬 프로그래밍을 수행하기 위해서 GPU 디바이스와 호환이 되는 CUDA 드라이버를 설치한다. 더 나아가, MPI 프로그래밍을 활용하기 위해서 OpenMPI 애플리케이션 설치를 통해 MPI 컴파일러를 설치한다. GPUDirect RDMA를 활용하기 위해서 gdr_copy 어플리케이션과 nv_peer_mem 드라이브를 설치하고, GPUDirect RDMA와 MPI 프로그래밍을 연동하기 위해 UCX 드라이버를 설치한다. 마지막으로, 로컬 서버와 리모트 환경이 연동된 Remote MPI Clustering 환경을 구축하기 위해 로컬 서버에서는 nfs-kernel-server 애플리케이션을, 그리고 리모트

Table 1. Configured Hardware Resources

Type	Device	Detail
PC1	CPU	Intel Xeon Silver 4208
	GPGPU	Nvidia Amphere 40 A40(48GB), Nvidia Amphere 100 A100(80GB)
	HCA	ConnectX5-EDR
PC2	CPU	Intel Xeon Silver 4210
	GPGPU	Nvidia Amphere 40 A40(48GB), Nvidia Amphere 100 A100(80GB)
	HCA	ConnectX5-EDR

Table 2. Installed Driver and Application

Type	Supporting Device or Technology		Detail
Driver	Device	GPU	nvidia-driver 510.73.05
		HCA	mlnx_ofed_linux-5.5-1.0.3.2_ubuntu20.04
	Tech.	CUDA	cuda-driver-11.6
		GPUDirect	gdr_copy-1.4
		GPUDirect RDMA	nv_peer_mem_1.3
		High-level Connection of GPUDirect RDMA with MPI	UCX-1.2
Application	Tech.	MPI	openmpi-4.1.0
		Remote MPI-clustering (local part)	nfs-kernel-server-3.0
		Remote MPI-clustering (remote part)	nfs-common-3.0

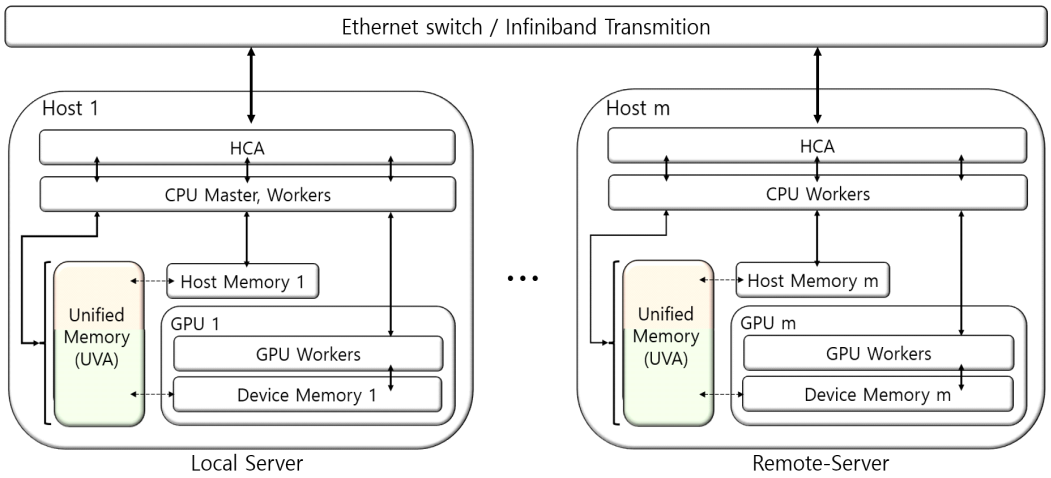

Fig. 2. GPUDirect RDMA and CUDA-Aware MPI technology applied system in Remote-MPI Clustering environment

서버에서는 nfs-common 애플리케이션을 설치한다.

해당 하드웨어 자원들에 기술한 드라이버 및 애플리케이션 설치를 통해 구성한 고성능 데이터 처리 기술 적용 시스템의 개요는 Fig. 2.와 같다. 로컬 서버의 GPU 디바이스 메모리에서 리모트 환경의 GPU 디바이스 메모리로의 MPI API를 활용한 데이터 전송이 진행될 때, CUDA-Aware MPI 기술이 활용된다. 해당 API의 인자로 Unified 메모리의 주소값을 삽입하여 GPU 메모리의 주소에 대한 접근이 진행된다. 그리고 GPUDirect RDMA 기술이 활용되어 전송되는 데이터는 CPU의 호스트 메모리에 복사되지 않고 로컬 서버의 HCA를 통해 리모트 서버의 HCA에게 직접적으로 전달된다. 리모트 서버에서 수신된 데이터는 CPU의 호스트 메모리에 복사되지 않고, 리모트 환경의 Unified 메모리 내에 할당된 GPU 디바이스 메모리 주소로 데이터가 전송된다. 또한, Remote MPI Clustering 기술을 통해 로컬 서버와 리모트 서버에서는 MPI 컴파일러로 컴파일된 동일한 프로그램을 병렬적으로 실행시켜 해당 데이터 전송 과정을 진행할 수 있다. MPI 병렬 프로그래밍에서 실행되는 프로세스들은 Rank 0부터 병렬 수행되는 프로세스의 수까지 각각의 Rank 번호가 부여된다. 이에 따라, Remote MPI Clustering 환경의 GPUDirect RDMA 기술과 CUDA-Aware MPI 기술이 적용된 고성능 데이터 처리 시스템 내에서 Rank 0의 경우 로컬 서버로 구성되고, Rank 0 이외의 Rank들은 리모트 서버로 구성된다.

3.2 시스템 설계

3.2.1 마스터-워커 패턴 시스템

제안한 고성능 기술이 적용된 시스템에 암호 분석 기술을 적용한다. 병렬적으로 암호 분석을 수행하기 위해서, 마스터로부터 분배되는 일을 다수의 워커가 병렬적으로 처리할 수 있는 마스터-워커 패턴의 시스템 토폴로지를 설계한다. Rank 별 역할이 분배된 마스터-워커 패턴 시스템에서 로컬 서버와 리모트 서버에게 두 가지의 방식으로 역할을 부여한다.

첫 번째 방식의 경우 Fig. 3.과 같이, 로컬 서버에서 Rank 0는 마스터 노드의 역할을 수행하고, Rank 0 이외의 Rank들은 리모트 서버에서 워커 노드의 역할을 수행한다. 이를 통해 마스터 노드는

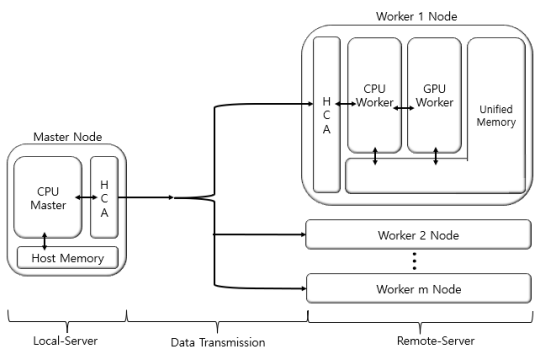

Fig. 3. Overview of the master-worker pattern system

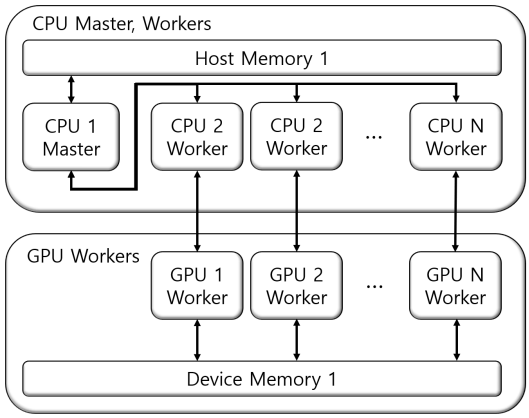

Fig. 4. Master and worker nodes assigned to local servers

GPUDirect RDMA 기술을 활용해 리모트 서버의 워커 노드들에게 대용량 데이터와 수행할 일을 분배하여 암호 분석을 진행한다. 두 번째 방식의 경우, 로컬 서버에 사용되는 하드웨어 자원이 마스터 노드의 데이터 분배 역할 이후에도 지속적인 사용을 위해 제안한다. 리모트 서버에게만 부여되었던 워커 노드의 역할을 Fig. 4.와 같이 로컬 서버에게도 부여하여 시스템을 설계한다. 이를 적용한 시스템에서 구성될 수 있는 총 워커 노드의 수는 다음과 같다. m 개의 서버가 있고, 로컬 서버를 제외한 m-1개의 서버가 있을 때, 각각의 서버에서 최대 n개의 워커 노드를 생성할 수 있다면 최대 n*(m-1)개의 워커 노드를 생성할 수 있다. 이에, 추가적으로 로컬 서버에서 생성할 수 있는 n-1개의 워커 노드를 추가하여 총 n*m-1개의 워커 노드를 구성할 수 있는 마스터-워커 패턴의 시스템을 설계한다.

제안한 마스터-워커 패턴 시스템 설계에 적용할 수 있는 암호 분석 기술의 예로 전수조사 암호 분석이 있다. 마스터 노드는 분석하고자 하는 대용량의 암호문을 최대 n*m-1개의 워커 노드들의 수로 분할하고 워커 노드들에게 분배한다. 그리고 각각의 워커 노드는 분배 받은 암호문을 병렬적으로 해독해 목표하는 평문을 찾는 과정을 진행한다.

3.2.2 워커 노드 간 통신 시스템

워커 노드들 간 상호 작용이 요구되는 마스터-워커 패턴 시스템을 설계할 경우, Fig. 5.와 같은 방식의 시스템 토폴로지 설계를 제안한다. 로컬 서버에서

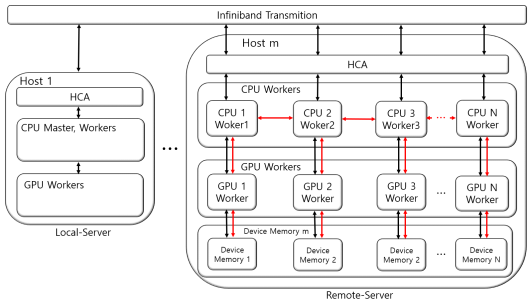

Fig. 5. Overview of worker to worker communication system

Rank 0의 프로세스는 마스터 노드 역할로 워커 노드들에게 대용량의 데이터와 일을 분배한다. 그리고 로컬 서버 또는 리모트 서버에 할당된 Rank 0 이외의 프로세스는 워커 노드 역할을 수행한다. 각각의 워커 노드들은 특정 암호 분석 연산을 수행하고, 해당 결과 값을 워커 노드들끼리 송•수신하여 최종적인 암호 분석을 수행한다. 이때, 다른 서버 내에 할당된 워커 노드들끼리 연산 결과 값을 송•수신하는 과정에서 HCA를 활용한 GPUDirect RDMA 기술이 활용되고 고속화된 데이터 전송으로 고성능의 암호 분석을 수행할 수 있다.

제안한 워커 노드 간 통신 가능한 시스템 설계에 적용할 수 있는 암호 분석 기술의 예로 행렬 곱 연산 고속화를 활용한 암호 분석이 있다. 대용량의 행렬 곱 연산의 경우, 워커 노드들의 독립적인 연산 수행과 더불어 워커 노드들끼리의 상호작용이 요구된다. 마스터 노드는 큰 스케일의 행렬 연산을 최대 n*m-1개의 워커 노드의 수로 분할하여 각각의 워커들에게 분배한다. 워커 노드들은 분배된 작은 행렬 연산 문제를 해결하고, 그 결과 값을 사용해 더 큰 행렬 연산 문제를 해결하는 워커 노드에게 해당 결과 값을 송신한다. 그리고 이와 같은 연산 과정이 반복되어 최종적으로 목표하는 큰 스케일의 행렬 문제를 해결하게 된다.

IV. 고성능 암호 분석 시스템 구현

4.1 Case Study 1: Password Cracking

Password Cracking이란 암호화된 정답 비밀번호를 갖고 있을 때, 정답이 될 수 있는 모든 후보 비밀번호 값에 대한 암호화를 진행한다. 그리고 순차적인 비교를 통해 목표하는 암호화된 정답 비밀번호 값

과 동일한 암호 값을 탐색하여, 최종적으로 암호화가 되지 않는 정답 비밀번호 값을 찾는 과정의 암호 분석 기술이다.

해당 Password Cracking 암호 분석 기술의 동작을 고성능 기술이 적용된 암호 분석 시스템으로 구현한다. 또한 마스터-워커 패턴의 시스템에 동작 과정들을 적용하여 마스터 노드와 워커 노드들에게 암호 분석 동작을 분배한다. 시스템 사용자는 공격자의 입장으로 후보 비밀번호들의 집합인 딕셔너리 파일, 해시 암호화된 비밀번호, 그리고 해시 암호화 모듈인 SHA-256을 소유하고 있다. 즉, 해시 연산된 정답 비밀번호의 정답 값을 알아내기 위해 전체 딕셔너리 파일을 해시 연산 모듈로 순차적인 암호화를 진행한다. 그리고 암호화된 후보 비밀번호 값과 기존에 보유한 암호화된 정답 비밀번호 값의 동일 여부를 판단한다. 최종적으로 동일한 암호화된 값을 발견하게 되면 해당 값과 1 대 1 대응되는 정답 비밀번호 값을 출력하여 암호 분석 동작을 수행한다.

마스터 노드는 정답 비밀번호에 대한 해시 값과 후보 비밀번호의 집합인 딕셔너리 파일을 워커 노드들에게 분배하는 역할을 수행한다. Fig. 6-①.과 같이 마스터 노드는 MPI_Bcast() API를 활용하여 전체 워커들에게 정답 비밀번호에 대한 해시 값을 브로드

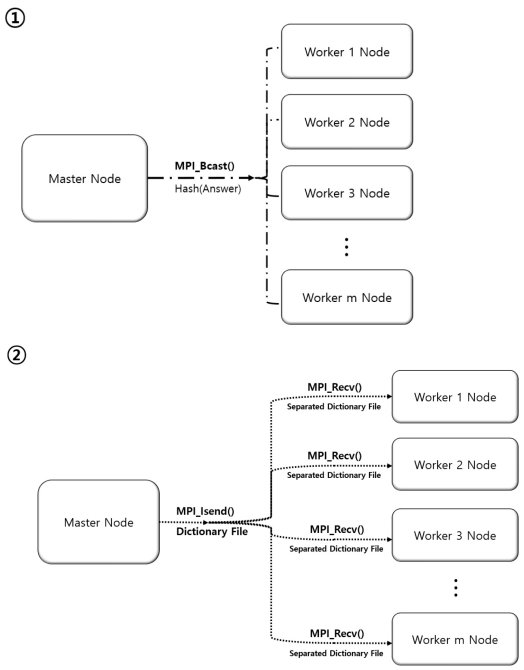

Fig. 6. Role of master node

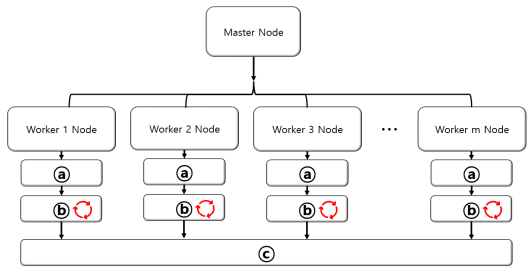

Fig. 7. Role of worker nodes

캐스팅한다. 또한, Unified 메모리로 전체 딕셔너리 파일을 로드 후, 암호 분석을 수행할 워커의 수만큼 해당 딕셔너리 파일을 분할한다. 이후 Fig. 6-②.와 같이 MPI_Isend() API를 활용하여 각각의 워커들에게 분할된 딕셔너리 파일을 송신한다. 이때 MPI_Isend() API를 통해 비동기로 워커 노드들에게 데이터 송신이 수행되어, 각각의 워커들은 데이터를 받음과 동시에 워커 노드들의 일을 처리하게 된다. 워커 노드는 마스터 노드로부터 수신한 데이터를 통해서 해시 값 연산 및 일치 여부를 판단하는 역할을 진행한다. 그리고 최종적으로 정답 비밀번호를 찾을 경우, 모든 워커의 동작을 종료시키고 최종적으로 시스템의 동작을 종료시키는 역할을 수행한다. Fig. 6-②.와 Fig. 7-(a).같이 워커 노드들은 MPI_Recv() API를 활용해 분할된 딕셔너리 파일을 수신한다. 데이터를 수신한 워커들은 Fig. 7-(b).와 같이 후보 비밀번호들에 대해서 SHA-256 해시 모듈을 활용해 해시 값 연산을 진행하고, 해시 연산된 정답 비밀번호의 값과 일치하는지 여부를 판단한다. 해당 작업은 일치하는 값을 찾을 때까지 반복되고, 특정 워커에서 일치하는 값을 발견했을 경우, Fig. 7-(c).와 같이 MPI_Abort() API를 통해서 모든 워커들의 작업을 중단시키고 최종적으로 시스템을 종료한다.

4.2 Case Study 2: GPU Reduction

Reduction이란 주어진 값들을 활용해 특정 하나의 값을 생성하는 과정을 의미한다. 예를 들어, 벡터 원소의 전체 합을 구하거나, 원소 값 중 평균값 얻는 과정 등을 의미한다. 그리고 CUDA 프로그래밍을 통해 GPU 하드웨어 자원을 활용한 병렬 프로그래밍으로 Reduction 문제를 해결하는 과정을 GPU Reduction이라 명명한다. GPU Reduction 과정은 벡터 원소들에 대한 동일한 연산을 어떤 알고리즘

의 Kernel Code로 구성했는지 그리고 해당 Kernel Code가 GPU의 어떤 하드웨어 자원을 사용하느냐에 따라 연산 속도의 성능 차이가 발생한다[9]. 이에 따라, GPU Reduction 기술을 활용해 다양한 연산 고속화 방법들이 제안되었다. 특히 암호 분석을 위하여 최적화된 벡터 연산 방법으로 행렬 덧셈 및 곱 연산을 고속화시키는 연구들이 진행되었다. 하나의 GPU에서 행렬 연산을 수행하는 Kernel Code가 CUDA Core를 최대한으로 활용하도록 하여 병렬성의 극대화를 의도하거나, 또는 행렬 연산 수행 알고리즘을 다양한 시각으로 최적화시켜 고속 연산을 의도한다. 하지만 이전 연구들에서 GPU Reduction 기술을 활용한 암호 분석 방법은 하나의 GPU만을 사용하여 분석을 진행하기에, 고속화된 연산 속도의 최대값은 하나의 GPU의 최대 속도라는 한계점이 존재한다. 해당 한계점을 해결하기 위해서 단순히 다량의 GPU를 활용하는 것은 대용량의 데이터 전송 과정에서의 큰 지연시간을 발생시켜 하나의 GPU에서 연산을 수행하는 것보다 더 하향된 성능을 나타낼 수 있다.

이에, 고성능 데이터 처리 기술이 적용된 GPU Clustering 환경에서 GPUDirect RDMA 기반의 고성능 암호 분석 시스템을 통해 해당 한계점을 해결할 수 있다. 제안한 워커 간 통신 가능한 마스터-워커 패턴의 시스템 설계에 행렬 곱 연산 고속화를 수행하는 GPU Reduction 기술을 적용한다. 대용량의 정방행렬 M에 대하여 서로 다른 값으로 초기화되어있는 M_1부터 M_{2N}까지의 행렬에 대한 곱 연산을 통해 암호 분석을 수행하는 문제가 있다고 가정하자. 해당 문제를 해결하기 위해서 Fig. 8.과 같이 마스터 노드와 N개의 워커 노드로 구성되어 있는 암호 분석 시스템을 구성 통해 문제를 해결한다.

마스터 노드는 Unified 메모리로 M_1부터 M_{2N}까지의 대용량의 행렬 데이터들을 모두 로드한다. 그리고 Fig. 8.의 Level 0와 같이 각각의 워커 노드들에게 GPUDirect RDMA 기술을 활용하여 MPI_Isend() API로 행렬 곱 연산을 수행할 두 개의 대용량의 행렬 데이터를 비동기 송신한다. 이후, Fig. 8.의 Level 1과 같이 모든 워커 노드들은 MPI_Recv() API를 통해 연산할 대용량의 행렬 데이터를 수신하고, GPU Reduction 기술을 활용하여 CUDA Core를 최대한으로 사용할 수 있는 행렬 곱 연산 알고리즘으로 대용량 행렬 곱 연산을 수행한다. 그리고 연산된 결과 값들을 인자로 활용해 행렬 연산 수행할 워커 노드들에게 MPI_Isend() API를 활용하여 비동기로 송신한다. 이후, Fig. 8.의 Level 2와 같이 행렬 연산을 진행할 데이터들을 워커 노드들이 MPI_Recv() API를 활용해 모두 수신하게 되면 Level 1에서 수행했던 것과 동일한 행렬 곱 연산을 수행한다. 해당 과정은 목표하는 행렬 곱 연산을 모두 수행하기까지 Level을 증가시키면서 반복 수행된다. 그리고 최종적으로 모든 행렬 연산이 완료되었을 때 연산된 행렬 값을 출력하고 시스템을 종료한다.

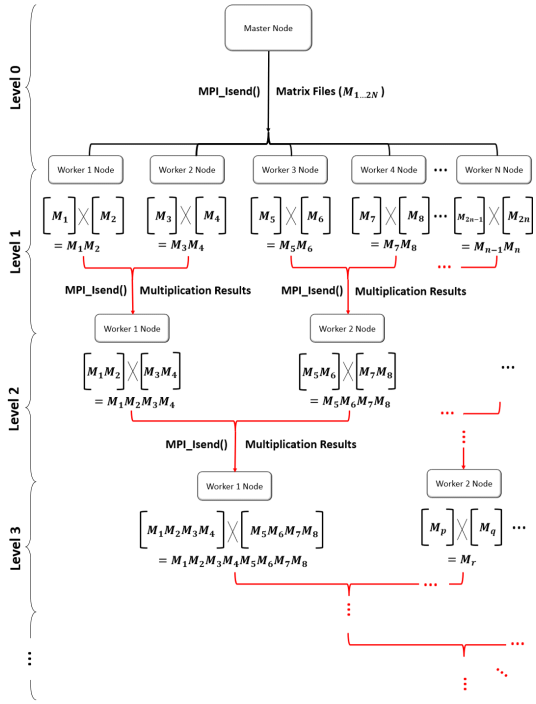

Fig. 8. Matrix multiplication accelerated of high-performance cryptanalysis system applying GPU Reduction technology

V. 성능 평가 및 기대효과

제안한 고성능 암호 분석 시스템 설계 및 구현 방법으로 Password Cracking 기술을 활용한 GPUDirect RDMA 기반의 고성능 암호 분석 시스템을 구축하였다. 그리고 GPUDirect RDMA 기술 적용 유무에 따른 암호 분석 시스템의 전체 연산 시간에 대한 비교를 통해 GPUDirect RDMA 기술이 적용된 시스템에 대한 감소된 연산 시간을 확인하였

다. 총 3가지의 방법으로 GPUDirect RDMA 적용 유무에 따른 성능 평가를 진행하였다. 첫 번째는 마스터 노드가 워커 노드에게 분배하는 데이터의 크기에 따른 성능 평가 비교이다. 두 번째는 워커 노드 별 분배되는 데이터의 크기와 작업해야 할 일의 수가 동일할 때, 암호 분석 시스템 내에서 구성한 워커 노드의 수에 따른 성능 평가이다. 세 번째는 워커 노드 별 분배되는 데이터의 크기와 시스템에서 사용하는 워커 노드의 수가 동일할 때 워커 노드가 작업해야 할 일의 수에 따른 성능 평가이다.

마스터 노드가 워커 노드에게 분배하는 데이터의 양에 따른 암호 분석 시스템 전체 실행 시간의 비교 결과는 다음 Fig. 9-(a).와 같다. 분배되는 데이터의 양이 증가할수록 전송할 데이터의 양이 증가하기에 GPUDirect RDMA 기술이 적용된 시스템과 적용되지 않은 시스템 모두 전체 연산 시간이 증가하였다. 하지만 GPUDirect RDMA 기술이 적용된 시스템의 경우, 최적화된 데이터 전송에 대한 효과를

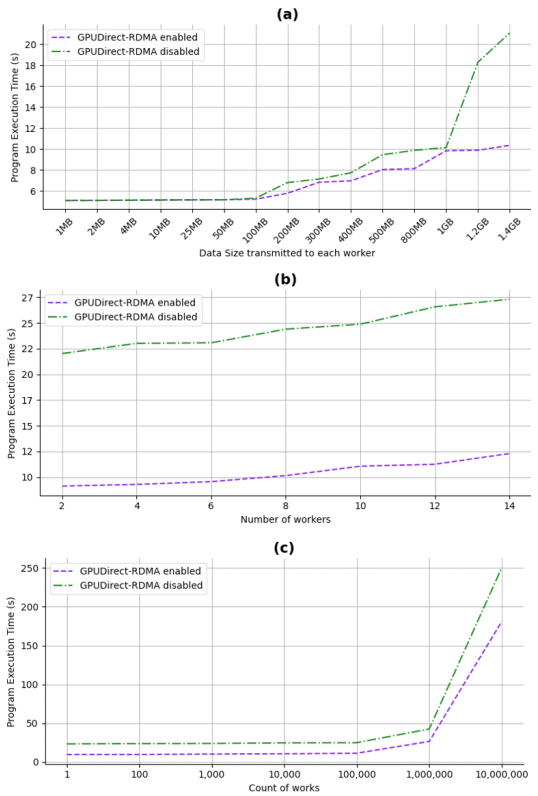

Fig. 9. Performance evaluation of high-performance cryptanalysis system based on GPUDirect RDMA

통해서 GPUDirect RDMA 기술이 적용되지 않은 시스템에 비해 최대 약 2배 이상의 암호 분석 연산 시간에 대한 감소를 확인할 수 있었다. 또한 워커 노드 별 분배되는 데이터의 크기가 동일할 때 구성한 워커 노드 수에 따른 성능 비교는 Fig. 9-(b).에서 확인할 수 있다. 시스템이 사용하는 워커 노드의 수가 증가할수록, 더 많은 수의 워커 노드들에게 대용량의 데이터를 분할하고 분배하는 프로세스를 처리해야 한다. 이 때문에 암호 분석 전체 연산에 대한 약간의 시간 증가 양상을 확인할 수 있다. 하지만 증가하는 워커 노드의 수에 대해서 GPUDirect RDMA가 적용되는 시스템의 경우 약 9초에서 12초 내외의 전체 연산 시간이 소요되는 반면 GPUDirect RDMA가 적용되지 않는 시스템의 경우 약 21초에서 27초 사이의 전체 연산 시간이 소요되는 것을 확인할 수 있다. 마지막으로, 워커 노드가 작업할 일의 수에 따른 성능 평가는 Fig. 9-(c).에서 확인할 수 있다. 워커 노드가 작업할 일의 수에 대한 지표는 Password Cracking의 정답을 찾는 동작 과정에서 워커 노드가 해시 연산 수행하는 후보 비밀번호의 수를 수치화하여 설정하였다. 그 결과, 워커 노드 별 수행해야 할 작업의 수가 증가할수록 암호 분석 연산 시간이 더 많이 요구되기에 전체적인 암호 분석 연산 시간은 워커 노드의 작업 수가 증가할수록 증가하는 양상임을 확인할 수 있다. 하지만 GPUDirect RDMA 기술이 적용된 시스템의 경우 적용되지 않은 시스템에 비해 전체 암호 분석 시간에 대해 최대 약 두 배 이상 감소한 것을 확인할 수 있다.

이에 따라, Password Cracking 기술을 활용하여 GPUDirect RDMA 기술이 적용된 고성능 암호 분석 시스템에 대한 전수조사 암호 분석의 성능 향상 실증할 수 있었다. 더 나아가, 제안한 고성능 암호 분석 시스템 설계 방식은 다양한 암호 분석 기술을 적용하는 것에 범용적으로 활용할 수 있을 것이라 기대한다. 마스터 노드에서 워커 노드로 데이터 전송 관련 MPI API를 사용할 때, 수신 받는 워커 노드의 위치가 리모트 서버일 경우, GPUDirect RDMA 기술이 사용되어 HCA를 통한 고성능 데이터 송신이 수행된다. 그렇기에 하나 이상의 GPU에게 대용량의 데이터와 암호 분석 작업을 병렬적으로 분배하여 암호 분석을 의도한다면, 제안한 마스터-워커 패턴의 시스템 구성을 통해 대용량의 데이터를 분배하는 과정에서 GPUDirect RDMA 기술이 활용되고, 전체적인 암호 분석 연산의 성능 향상을 기대할 수 있다.

그리고 암호 분석 과정 중 워커 노드 간의 상호작용이 요구된다면 워커 노드 간 대용량의 데이터를 송신하는 과정에서 GPUDirect RDMA 기술이 활용되어 전체적인 암호 분석 연산의 성능 향상을 기대할 수 있다.

VI. 결 론

본 논문은 딥러닝 및 HPC 분야의 GPU Clustering 환경에서 최적화된 데이터 전송을 위해 활용되는 GPUDirect RDMA, Remote MPI Clustering, CUDA-Aware MPI 기술에 대한 정의와 구성 방법을 기술한다. 그리고 GPUDirect RDMA 기술에 대한 벤치마킹을 통해 기술의 성능 평가를 수행하여 GPUDirect RDMA 기술이 적용되는 시스템에 대한 기대효과를 제시한다. 더 나아가, 해당 고성능 데이터 처리 기술들을 활용한 고성능 암호 분석 시스템 설계 방법 제시하여 GPUDirect RDMA 기반의 고성능 암호 분석 시스템을 구현한다. 일반적인 마스터-워커 패턴의 시스템 설계의 경우, 전수조사 암호 분석과 같이 처리할 대용량의 데이터를 워커 노드들에게 분배하는 형태의 암호 분석 기술에 대해 적용하는 것이 용이하다. 그리고 워커 간 통신 가능한 시스템 설계의 경우, 행렬 곱 연산 고속화와 같이 워커 노드들 간 상호작용이 요구되는 암호 분석 기술에 대해 적용하는 것이 용이하다. 그 사례로, Password Cracking, GPU Reduction 기술을 제안한 암호 분석 시스템 설계에 적용하여 고성능 암호 분석 시스템 구현 방법에 대해 제시한다. 더 나아가, Password Cracking 기술로 구현한 암호 분석 시스템에 대해서 고성능 기술의 적용 유무에 따른 성능 평가를 진행한다. 그 결과, 암호 분석 처리할 데이터의 전송되는 양이 증가할수록 GPUDirect RDMA 기술의 최적화된 데이터 전송에 대한 효과를 확인할 수 있었고, 전체적인 암호 분석 연산의 수행 시간이 더 빠른 것을 확인할 수 있었다. 결론적으로, 제안된 암호 분석 시스템 설계를 통해 다양한 암호 분석 기술을 범용적으로 적용할 수 있고, 제안된 설계를 활용하여 GPUDirect RDMA 기반의 고성능 암호 분석 시스템을 구현할 경우, GPUDirect RDMA 기술의 데이터 전송 최적화를 통해 더 빠른 전체 암호 분석 연산 시간을 기대할 수 있다.

References

[1] Cihangir Tezcan, “Key lengths revisited: gpu-based brute force cryptoanalysis of des, 3des, and present,” Journal of Systems Architecture, Vol. 124, pp. 102402, Mar. 2022.

[2] Alex Biryukov and Johann Großschädl, “Cryptanalysis of the full aes using gpu-like special-purpose hardware,” Fundamenta Informaticae, Vol. 114, pp. 221-237, Apr. 2012.

[3] Marco Cianfriglia, and Stefano Guarino, “Cryptanalysison gpus with the cube attack: design, optimization and performances gains,” 2017 International Conference on High Performance Computing & Simulation (HPCS), pp. 753-760, Jul. 2017.

[4] NVIDIA GPUDirect - NVIDIA Developer, “GPUDirect RDMA NVIDIA” https://developer.nvidia.com/gpudirect, Oct. 2022.

[5] Open MPI: Open Source High Performance Computing, “Open MPI(Message Passing Interface)” https://www.open-mpi.org/, Oct. 2022.

[6] Mellanox/nv_peer_memory - GPUDirect RDMA - Github, “Mellanox nv_peer_mem with GPUDirect RDMA” https://github.com/Mellanox/nv_peer_memory, Oct. 2022.

[7] Nathan Hanford, Ramesh Pankajakshan, Edgar A León and Ian Karlin. “Challenges of gpu-aware communication in mpi,” In 2020 Workshop on Exascale MPI (ExaMPI) IEEE, pp. 1-10, Nov. 2020.

[8] Jaemin Choi, Zane Fink, Sam White, Nitin Bhat, David F. Richards and Laxmikant V. Kale, “Gpu-aware communication with ucx in parallel programming models: charm++, mpi,

and python," 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 479-488, Jun. 2021.
[9] Optimizing Parallel Reduction in CUDA - NVIDIA, "GPU Reduction" https://developer.download.nvidia.com/assets/cuda/files/reduction.pdf, Oct. 2022.

〈저자소개〉

이 석 민 (Seokmin Lee) 학생회원
2022년 2월: 광운대학교 컴퓨터정보공학부 컴퓨터공학전공 학사
2022년 3월~현재: 고려대학교 정보보호대학원 융합보안학과 Samsung Advanced Security 전공 석사과정
〈관심분야〉 시스템 보안, 클라우드 보안, 딥러닝, GPU Clustering 시스템 설계

신 영 주 (Youngjoo Shin) 종신회원
2006년 2월: 고려대학교 컴퓨터학과 학사
2008년 2월: KAIST 전산학과 석사
2014년 2월: KAIST 전산학과 박사
2008년 4월~2017년 2월: 국가보안기술연구소 선임연구원
2017년 3월~2020년 8월: 광운대학교 컴퓨터정보공학부 조교수
2020년 9월~2022년 2월: 고려대학교 정보보호대학원 정보보호학과 조교수
2022년 3월~현재: 고려대학교 정보보호대학원 정보보호학과 부교수
〈관심분야〉 시스템 보안, CPU 마이크로아키텍처 취약점 분석, 클라우드 보안