



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0053910
(43) 공개일자 2024년04월25일

(51) 국제특허분류(Int. Cl.)

H04L 9/00 (2022.01)

(52) CPC특허분류

H04L 9/00 (2022.05)

(21) 출원번호 10-2022-0134022

(22) 출원일자 2022년10월18일

심사청구일자 2022년10월18일

(71) 출원인

국방과학연구소

대전광역시 유성구 북유성대로488번길 160 (수남동)

고려대학교 산학협력단

서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)

(72) 발명자

신영주

서울시 성북구 안암로 145

이석민

서울시 성북구 안암로 145

(74) 대리인

특허법인 광장리앤코

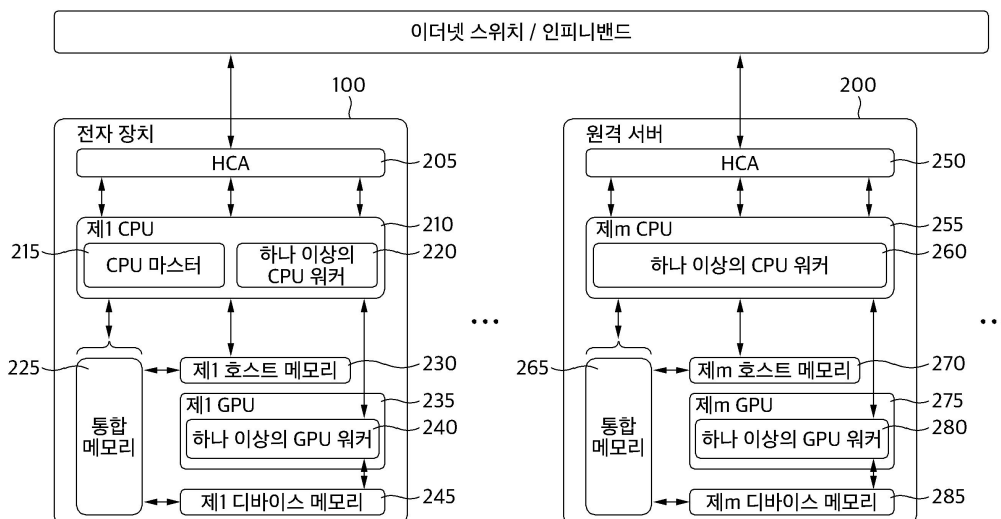
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 전자 장치 및 그의 암호 분석 방법

(57) 요약

전자 장치의 암호 분석 방법이 제공된다. 암호 분석 방법은 상기 전자 장치 및 하나 이상의 원격(remote) 서버 중 적어도 하나에 대응하는 하나 이상의 워커(worker) 노드를 확인하는 단계; 상기 하나 이상의 워커 노드의 수에 기초하여, 복수의 후보 평문(plain text)들을 분할하는 단계; 및 상기 분할된 복수의 후보 평문들 및 암호문(cypher text)을 상기 하나 이상의 워커 노드 각각으로 송신하는 단계를 포함하고, 상기 복수의 후보 평문들은 상기 하나 이상의 워커 노드에 의해 하나 이상의 암호화 알고리즘에 따라 암호화되고, 암호화된 결과 값이 상기 암호문과 일치하는지 여부가 확인될 수 있다.

대표도



명세서

청구범위

청구항 1

전자 장치의 암호 분석 방법에 있어서,

상기 전자 장치 및 하나 이상의 원격(remote) 서버 중 적어도 하나에 대응하는 하나 이상의 워커(worker) 노드를 확인하는 단계;

상기 하나 이상의 워커 노드의 수에 기초하여, 복수의 후보 평문(plain text)들을 분할하는 단계; 및

상기 분할된 복수의 후보 평문들 및 암호문(cypher text)을 상기 하나 이상의 워커 노드 각각으로 송신하는 단계를 포함하고,

상기 복수의 후보 평문들은 상기 하나 이상의 워커 노드에 의해 하나 이상의 암호화 알고리즘에 따라 암호화되고, 암호화된 결과 값이 상기 암호문과 일치하는지 여부가 확인되는, 암호 분석 방법.

청구항 2

제1 항에 있어서, 상기 분할된 복수의 후보 평문들 및 상기 암호문을 상기 하나 이상의 워커 노드 각각으로 송신하는 단계는,

상기 분할된 복수의 후보 평문들을 통합(unified) 메모리로 로드하는 단계; 및

메시지 전달 인터페이스(message passing interface, MPI)를 통해, 상기 분할된 복수의 후보 평문들이 로드된 상기 통합 메모리의 주소 값을 상기 하나 이상의 워커 노드 각각으로 송신하는 단계를 포함하는, 암호 분석 방법.

청구항 3

제2 항에 있어서,

상기 통합 메모리의 주소 값을 상기 하나 이상의 워커 노드 각각으로 송신하는 단계는, MPI 송신 애플리케이션 프로그래밍 인터페이스(application programming interface, API)를 통해 상기 통합 메모리의 주소 값을 송신하는 단계를 포함하고,

상기 통합 메모리의 주소 값은, MPI 수신 API를 통해 상기 하나 이상의 워커 노드에 의해 수신되는, 암호 분석 방법.

청구항 4

제1 항에 있어서, 상기 분할된 복수의 후보 평문들 및 상기 암호문을 상기 하나 이상의 워커 노드 각각으로 송신하는 단계는,

MPI 브로드캐스팅 API를 통해, 상기 하나 이상의 워커 노드에게 상기 암호문을 브로드캐스팅하는 단계를 포함하는, 암호 분석 방법.

청구항 5

제1 항에 있어서,

상기 하나 이상의 워커 노드 중 하나의 워커 노드가 상기 암호문과 일치하는 결과 값을 확인한 경우, 상기 하나

의 워커 노드로부터 나머지 워커 노드로 작업 중단 요청이 송신되는, 암호 분석 방법.

청구항 6

제5 항에 있어서,

상기 작업 중단 요청은 MPI 중단 API를 통해 송신되는, 암호 분석 방법.

청구항 7

제1 항에 있어서, 상기 하나 이상의 워커 노드를 확인하는 단계는,

상기 복수의 후보 평문들의 수, 하나 이상의 암호화 알고리즘의 수, 상기 하나 이상의 원격 서버의 수, 상기 전자 장치 내 GPU(graphics processing unit)의 수 및 상기 하나 이상의 원격 서버 내 GPU의 수 중 적어도 하나에 기초하여, 상기 하나 이상의 워커 노드의 수를 결정하는 단계를 포함하는, 암호 분석 방법.

청구항 8

제1 항에 있어서, 복수의 후보 평문들을 분할하는 단계는,

상기 하나 이상의 워커 노드에 대응하는 GPU의 수 및 성능 중 적어도 하나에 기초하여, 상기 하나의 워커 노드 각각에 상기 복수의 후보 평문들을 할당하는 단계를 포함하는, 암호 분석 방법.

청구항 9

전자 장치로서,

메모리; 및

제어부(controller)를 포함하고, 상기 제어부는,

상기 전자 장치 및 하나 이상의 원격(remote) 서버 중 적어도 하나에 대응하는 하나 이상의 워커(worker) 노드를 확인하고,

상기 하나 이상의 워커 노드의 수에 기초하여, 복수의 후보 평문(plain text)들을 분할하고,

상기 분할된 복수의 후보 평문들 및 암호문(cypher text)을 상기 하나 이상의 워커 노드 각각으로 송신하고,

상기 복수의 후보 평문들은 상기 하나 이상의 워커 노드에 의해 하나 이상의 암호화 알고리즘에 따라 암호화되고, 암호화된 결과 값이 상기 암호문과 일치하는지 여부가 확인되는, 전자 장치.

청구항 10

전자 장치의 암호 분석 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 비일시적 기록매체로서,

상기 암호 분석 방법은,

상기 전자 장치 및 하나 이상의 원격(remote) 서버 중 적어도 하나에 대응하는 하나 이상의 워커(worker) 노드를 확인하는 단계;

상기 하나 이상의 워커 노드의 수에 기초하여, 복수의 후보 평문(plain text)들을 분할하는 단계; 및

상기 분할된 복수의 후보 평문들 및 암호문(cypher text)을 상기 하나 이상의 워커 노드 각각으로 송신하는 단계를 포함하고,

상기 복수의 후보 평문들은 상기 하나 이상의 워커 노드에 의해 하나 이상의 암호화 알고리즘에 따라 암호화되고, 암호화된 결과 값이 상기 암호문과 일치하는지 여부가 확인되는, 비밀시적 기록매체.

발명의 설명

기술 분야

[0001] 본 개시는 암호 분석을 수행하기 위한 전자 장치 및 그 제어 방법에 관한 것이다.

배경 기술

[0002] 기존의 연구들에서는 전수 조사에 대한 시간을 단축하기 위해 CUDA 프로그래밍을 통한 전수 조사 암호 분석이 수행되었다. 또는, 해당 암호 분석 코드를 최적화 및 경량화하는 것에 집중하여 전수 조사의 전체 시간을 단축하기 위한 연구가 수행되었다. 더 나아가, HPC분야의 GPU 클러스터링 개념을 활용해 다량의 GPU를 병렬 연산 처리에 활용하여 전수 조사 암호 분석의 전체 시간을 단축하고자 하는 연구가 수행되었다.

[0003] 다만, 시스템에 따라 하나의 시스템에 장착될 수 있는 GPU의 수는 한정적일 뿐만 아니라, 최적화된 GPU 클러스터링 환경이 아닐 경우, 대용량의 데이터 전송 과정에서 전송 지연 문제를 발생시켜 하나의 GPU만을 활용해 암호 분석을 진행하는 것보다 더 긴 연산 시간이 소요되는 문제점이 존재한다.

발명의 내용

해결하려는 과제

[0004] 본 실시 예가 해결하고자 하는 과제는, 암호 분석을 수행하는 전자 장치 및 그 제어 방법을 제공하는데 있다.

[0005] 구체적으로, 본 실시 예는 다량의 GPU를 활용해 대용량의 데이터 전송을 최적화된 상태로 송수신할 수 있는 GPU 클러스터링 환경을 구축하고, 해당 구축된 환경에 암호 분석 시스템을 설계 및 구현하여 이전 연구들에서 제한되었던 다량의 GPU를 활용한 최적화된 암호 분석 시스템을 제공할 수 있다.

[0006] 본 실시 예가 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제들로 한정되지 않으며, 이하의 실시 예들로부터 또 다른 기술적 과제들이 유추될 수 있다.

과제의 해결 수단

[0007] 본 개시의 일 측면은 전자 장치의 암호 분석 방법에 있어서, 상기 전자 장치 및 하나 이상의 원격(remote) 서버 중 적어도 하나에 대응하는 하나 이상의 워커(worker) 노드를 확인하는 단계; 상기 하나 이상의 워커 노드의 수에 기초하여, 복수의 후보 평문(plain text)들을 분할하는 단계; 및 상기 분할된 복수의 후보 평문들 및 암호문(cypher text)을 상기 하나 이상의 워커 노드 각각으로 송신하는 단계를 포함하고, 상기 복수의 후보 평문들은 상기 하나 이상의 워커 노드에 의해 하나 이상의 암호화 알고리즘에 따라 암호화되고, 암호화된 결과 값이 상기 암호문과 일치하는지 여부가 확인되는, 방법을 제공할 수 있다.

[0008] 또한, 본 개시의 일 실시 예에서 상기 분할된 복수의 후보 평문들 및 상기 암호문을 상기 하나 이상의 워커 노드 각각으로 송신하는 단계는, 상기 분할된 복수의 후보 평문들을 통합(unified) 메모리로 로드하는 단계; 및 메시지 전달 인터페이스(message passing interface, MPI)를 통해, 상기 분할된 복수의 후보 평문들이 로드된 상기 통합 메모리의 주소 값을 상기 하나 이상의 워커 노드 각각으로 송신하는 단계를 포함하는, 암호 분석 방법을 제공할 수 있다.

[0009] 또한, 본 개시의 일 실시 예에서 상기 통합 메모리의 주소 값을 상기 하나 이상의 워커 노드 각각으로 송신하는 단계는, MPI 송신 애플리케이션 프로그래밍 인터페이스(application programming interface, API)를 통해 상기 통합 메모리의 주소 값을 송신하는 단계를 포함하고, 상기 통합 메모리의 주소 값은, MPI 수신 API를 통해 상기 하나 이상의 워커 노드에 의해 수신되는, 암호 분석 방법을 제공할 수 있다.

[0010] 또한, 본 개시의 일 실시 예에서 상기 분할된 복수의 후보 평문들 및 상기 암호문을 상기 하나 이상의 워커 노드 각각으로 송신하는 단계는, MPI 브로드캐스팅 API를 통해, 상기 하나 이상의 워커 노드에게 상기 암호문을 브로드캐스팅하는 단계를 포함하는, 암호 분석 방법을 제공할 수 있다.

[0011] 또한, 본 개시의 일 실시 예에서 상기 하나 이상의 워커 노드 중 하나의 워커 노드가 상기 암호문과 일치하는

결과 값을 확인한 경우, 상기 하나의 워커 노드로부터 나머지 워커 노드로 작업 중단 요청이 송신되는, 암호 분석 방법을 제공할 수 있다.

[0012] 또한, 본 개시의 일 실시 예에서 상기 작업 중단 요청은 MPI 중단 API를 통해 송신되는, 암호 분석 방법을 제공할 수 있다.

[0013] 또한, 본 개시의 일 실시 예에서 상기 하나 이상의 워커 노드를 확인하는 단계는, 상기 복수의 후보 평문들의 수, 하나 이상의 암호화 알고리즘의 수, 상기 하나 이상의 원격 서버의 수, 상기 전자 장치 내 GPU(graphics processing unit)의 수 및 상기 하나 이상의 원격 서버 내 GPU의 수 중 적어도 하나에 기초하여, 상기 하나 이상의 워커 노드의 수를 결정하는 단계를 포함하는, 암호 분석 방법을 제공할 수 있다.

[0014] 또한, 본 개시의 일 실시 예에서 복수의 후보 평문들을 분할하는 단계는, 상기 하나 이상의 워커 노드에 대응하는 GPU의 수 및 성능 중 적어도 하나에 기초하여, 상기 하나의 워커 노드 각각에 상기 복수의 후보 평문들을 할당하는 단계를 포함하는, 암호 분석 방법을 제공할 수 있다.

[0015] 본 개시의 다른 측면은 메모리; 및 제어부(controller)를 포함하고, 상기 제어부는, 상기 전자 장치 및 하나 이상의 원격(remote) 서버 중 적어도 하나에 대응하는 하나 이상의 워커(worker) 노드를 확인하고, 상기 하나 이상의 워커 노드의 수에 기초하여, 복수의 후보 평문(plain text)들을 분할하고, 상기 분할된 복수의 후보 평문들 및 암호문(cypher text)을 상기 하나 이상의 워커 노드 각각으로 송신하고, 상기 복수의 후보 평문들은 상기 하나 이상의 워커 노드에 의해 하나 이상의 암호화 알고리즘에 따라 암호화되고, 암호화된 결과 값이 상기 암호문과 일치하는지 여부가 확인되는, 전자 장치를 제공할 수 있다.

[0016] 본 개시의 또 다른 측면은 전자 장치의 암호 분석 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 비일시적 기록매체로서, 상기 암호 분석 방법은, 상기 전자 장치 및 하나 이상의 원격(remote) 서버 중 적어도 하나에 대응하는 하나 이상의 워커(worker) 노드를 확인하는 단계; 상기 하나 이상의 워커 노드의 수에 기초하여, 복수의 후보 평문(plain text)들을 분할하는 단계; 및 상기 분할된 복수의 후보 평문들 및 암호문(cypher text)을 상기 하나 이상의 워커 노드 각각으로 송신하는 단계를 포함하고, 상기 복수의 후보 평문들은 상기 하나 이상의 워커 노드에 의해 하나 이상의 암호화 알고리즘에 따라 암호화되고, 암호화된 결과 값이 상기 암호문과 일치하는지 여부가 확인되는, 비일시적 기록 매체를 제공할 수 있다.

[0017] 기타 실시 예들의 구체적인 사항들은 상세한 설명 및 도면들에 포함되어 있다.

발명의 효과

[0018] 본 개시에 따르면, 전자 장치 및 그 제어 방법은, 다량의 GPU를 활용해 대용량의 데이터 전송을 최적화된 상태로 송수신할 수 있는 GPU 클러스터링 환경을 구축함으로써, 암호를 분석하는데 소요되는 시간을 줄일 수 있다.

[0019] 발명의 효과는 이상에서 언급한 효과로 제한되지 않으며, 언급되지 않은 또 다른 효과들은 청구범위의 기재로부터 당해 기술 분야의 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0020] 도 1은 일 실시 예에 따른 시스템을 나타낸다.

도 2는 일 실시 예에 따른 시스템의 동작 개요를 나타낸다.

도 3은 일 실시 예에 따른 마스터-워커 패턴 시스템의 작동 방식을 나타낸다.

도 4는 일 실시 예에 따른 마스터-워커 패턴 시스템의 작동 방식을 나타낸다.

도 5는 일 실시 예에 따른 마스터-워커 패턴 시스템의 작동 방식을 나타낸다.

도 6a 내지 도 6c는 일 실시 예에 따른 전자 장치가 암호를 분석하는 과정을 설명하기 위한 도면이다.

도 7a 내지 도 7c는 전자 장치가 일 실시 예에 따라 암호 분석을 수행한 결과를 설명하기 위한 도면이다.

도 8은 일 실시 예에 따른 전자 장치의 암호 분석 방법의 흐름도를 나타낸다.

도 9는 일 실시 예에 따른 전자 장치의 블록도를 나타낸다.

발명을 실시하기 위한 구체적인 내용

- [0021] 실시 예들에서 사용되는 용어는 본 개시에서의 기능을 고려하면서 가능한 현재 널리 사용되는 일반적인 용어들을 선택하였으나, 이는 당 분야에 종사하는 기술자의 의도 또는 관례, 새로운 기술의 출현 등에 따라 달라질 수 있다. 또한, 특정한 경우는 출원인이 임의로 선정한 용어도 있으며, 이 경우 해당되는 설명 부분에서 상세히 그 의미를 기재할 것이다. 따라서 본 개시에서 사용되는 용어는 단순한 용어의 명칭이 아닌, 그 용어가 가지는 의미와 본 개시의 전반에 걸친 내용을 토대로 정의되어야 한다.
- [0022] 명세서 전체에서 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있음을 의미한다.
- [0023] 명세서 전체에서 기재된 "a, b, 및 c 중 적어도 하나"의 표현은, 'a 단독', 'b 단독', 'c 단독', 'a 및 b', 'a 및 c', 'b 및 c', 또는 'a, b, 및 c 모두'를 포괄할 수 있다.
- [0024] 이하에서 언급되는 "단말"은 네트워크를 통해 서버나 타 단말에 접속할 수 있는 컴퓨터나 휴대용 단말로 구현될 수 있다. 여기서, 컴퓨터는 예를 들어, 웹 브라우저(WEB Browser)가 탑재된 노트북, 데스크톱(desktop), 랩톱(laptop) 등을 포함하고, 휴대용 단말은 예를 들어, 휴대성과 이동성이 보장되는 무선 통신 장치로서, IMT(International Mobile Telecommunication), CDMA(Code Division Multiple Access), W-CDMA(W-Code Division Multiple Access), LTE(Long Term Evolution) 등의 통신 기반 단말, 스마트폰, 태블릿 PC 등과 같은 모든 종류의 핸드헬드(Handheld) 기반의 무선 통신 장치를 포함할 수 있다.
- [0025] 아래에서는 첨부한 도면을 참고하여 본 개시의 실시 예에 대하여 본 개시가 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 개시는 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시 예에 한정되지 않는다.
- [0026] 이하에서는 도면을 참조하여 본 개시의 실시 예들을 상세히 설명한다.
- [0027] 도 1은 일 실시 예에 따른 시스템을 나타낸다.
- [0028] 도 1을 참조하면, 시스템은 전자 장치(100), 하나 이상의 원격 서버들(120, 140, 160) 및 네트워크(180) 중에서 적어도 하나를 포함할 수 있다. 한편, 도 1에 도시된 시스템은 본 실시 예와 관련된 구성요소들만이 도시되어 있다. 따라서, 도 1에 도시된 구성요소들 외에 다른 범용적인 구성요소들이 더 포함될 수 있음을 본 실시 예와 관련된 기술분야에서 통상의 지식을 가진 자라면 이해할 수 있다.
- [0029] 전자 장치(100)는 전자 장치(100) 또는 하나 이상의 원격 서버들(120, 140, 160)에 대응하는 하나 이상의 워커 노드에게 데이터 및 수행할 일을 분배하여 암호를 분석할 수 있다. 예를 들어, 전자 장치(100)에서 생성된 마스터 노드는 하나 이상의 워커 노드의 수에 기초하여, 복수의 후보 평문들을 분할하고, 분할된 복수의 후보 평문들을 하나 이상의 워커 노드에게 송신하여 암호문에 대응하는 평문을 찾으려 할 수 있다. 한편, 전자 장치(100)는 원격 서버(120, 140, 160)와 구별되는 개념으로서 로컬 서버 등으로 혼용될 수 있으나, 이에 한정되는 것은 아니다.
- [0030] 하나 이상의 원격(remote) 서버들(120, 140, 160)은 전자 장치(100)로부터 데이터 및 수행할 일을 할당 받아 암호를 분석할 수 있다. 예를 들어, 하나 이상의 원격 서버들(120, 140, 160)에서 생성된 워커 노드들은 복수의 후보 평문들 및 암호문을 수신할 수 있다. 이후, 워커 노드들은 각각 수신한 복수의 후보 평문들을 하나 이상의 암호화 알고리즘에 따라 암호화하고, 암호화된 결과 값들 중 암호문과 일치하는 결과 값을 확인할 수 있다.
- [0031] 하나 이상의 원격 서버들(120, 140, 160)과 전자 장치(100)는 네트워크(180) 내에서 서로 통신할 수 있다. 네트워크(180)는 근거리 통신망(Local Area Network; LAN), 광역 통신망(Wide Area Network; WAN), 부가가치 통신망(Value Added Network; VAN), 이동 통신망(mobile radio communication network), 위성 통신망 및 이들의 상호 조합을 포함하며, 도 1에 도시된 각 네트워크 구성 주체가 서로 원활하게 통신을 할 수 있도록 하는 포괄적인 의미의 데이터 통신망이며, 유선 인터넷, 무선 인터넷 및 모바일 무선 통신망을 포함할 수 있다. 무선 통신은 예를 들어, 무선 랜(Wi-Fi), 블루투스, 블루투스 저 에너지(Bluetooth low energy), 지그비, WFD(Wi-Fi Direct), UWB(ultra wideband), 적외선 통신(IrDA, infrared Data Association), NFC(Near Field Communication) 등이 있을 수 있다. 한편, 보다 높은 수율(throughput), 낮은 레이턴시(latency) 및 높은 안정성을 위하여, 인피니밴드(Infiniband) 또는 이더넷(Ethernet) 기술 등이 네트워크(180)로서 사용될 수 있으나, 이에 한정되는 것은 아니다.
- [0032] 도 2는 일 실시 예에 따른 시스템의 동작 개요를 나타낸다.
- [0033] 일 실시 예에 따르면, 시스템은 전자 장치(100) 및 원격 서버(200)를 비롯한 M개의 서버를 포함할 수 있고, 각

각의 서버는 GPU(graphics processing unit)(235, 275)를 포함할 수 있다. 즉, 시스템은 다량의 GPU가 병렬 연산 처리에 활용되어 암호를 분석할 수 있도록, GPU 클러스터링(clustering) 환경으로 설계될 수 있다.

[0034] 일 실시 예에 따르면, GPU 클러스터링 환경에서 고성능의 데이터 전송 처리가 가능하도록, GPUDirect RDMA, Remote-MPI 클러스터링, Cuda-Aware MPI 등의 기술들이 사용될 수 있다.

[0035] 일 실시 예에 따르면, GPUDirect RDMA(remote direct memory access) 기술은 로컬 환경의 GPU 디바이스 메모리와 원격 환경의 GPU 디바이스 메모리 간의 직접적인 데이터 송수신을 가능케 하는 기술을 나타낼 수 있다. 보다 구체적으로, GPUDirect 기술은 GPU와 GPU 또는 GPU와 서드파티 디바이스들이 CPU의 호스트 메모리로의 복사 없이 직접적으로 통신할 수 있도록 하는 기술들의 집합을 나타내고, GPUDirect RDMA 기술은 GPUDirect 기술 중 네트워크 어댑터인 HCA(host channel adapter)(205, 250)를 활용해 원격 서버의 GPU 메모리에 직접적으로 접근할 수 있도록 하는 기술을 나타낼 수 있다.

[0036] 일 실시 예에 따르면, GPU에서 연산 처리가 수행될 때 CUDA API를 사용하여 GPU 디바이스 메모리가 활용될 수 있다. 이때, GPUDirect RDMA 기술이 사용되지 않는 시스템의 경우, 로컬 환경의 디바이스 메모리에서 호스트 메모리로 데이터가 복사되고, HCA를 통해 로컬 환경의 호스트 메모리에서 원격 환경의 호스트 메모리로 복사된 후, 호스트 메모리에서 디바이스 메모리로 데이터가 복사되는 과정이 수행되어야 한다. 그러나, GPUDirect RDMA 기술이 지원되는 시스템의 경우, 로컬 환경의 디바이스 메모리에서 원격 환경의 디바이스 메모리로 데이터가 직접적으로 전달될 수 있다. 이에 따라, 호스트 메모리와 디바이스 메모리 간의 불필요한 복사가 감소됨으로써 하나 이상의 GPU 메모리들이 향상된 전송 속도로 데이터를 송·수신하여 병렬 연산 성능이 향상되는 기술적 효과를 얻을 수 있다. 그리고 이를 통해, 전체적인 암호 분석 시스템의 연산 시간이 단축되는 기술적 효과 얻을 수 있다.

[0037] 일 실시 예에 따르면, Remote MPI(message passing interface) 클러스터링 기술은 서버에게 프로세스를 할당하여 로컬 서버와 원격 서버가 병렬적으로 동시에 프로그램을 실행 가능케 하는 기술을 나타낼 수 있다. 보다 구체적으로, Remote MPI 클러스터링 기술은 서로 다른 프로세서 간 공유 메모리를 사용하지 않고, 별도의 메시지라는 메모리를 선언하여 태그 값을 프로세스의 고유값으로 활용해 프로세스들을 구분함으로써, MPI API를 활용하여 프로세서 간 데이터를 송수신하고 다수의 프로세스에 작업을 할당하여 병렬적으로 작업을 수행 가능케 하는 기술을 나타낼 수 있다. 이에 따라, Remote MPI 클러스터링 기술이 사용됨으로써, 컴퓨터 자원이 보다 효율적으로 활용되는 기술적 효과를 얻을 수 있다.

[0038] 일 실시 예에 따르면, Cuda-Aware MPI 기술은 하나의 MPI 데이터 전송 명령어를 통해 GPU의 디바이스 메모리 간의 데이터 송수신을 가능케 하는 기술을 나타낼 수 있다. 보다 구체적으로, Cuda-Aware MPI 기술은 통합(unified) 메모리 개념을 도입하여, 호스트 메모리 영역과 디바이스 메모리 영역을 하나의 가상 주소로 접근 가능하게 함으로써, GPU의 디바이스 메모리에서 CPU의 호스트 메모리로의 불필요한 데이터 복사 과정을 생략 가능케 하는 기술을 나타낼 수 있다.

[0039] 도 2를 참조하면, 전자 장치(100)의 제1 디바이스 메모리(245)에서 원격 서버(200)의 제 m 디바이스 메모리(285)로 데이터가 전송될 때, 전송한 GPUDirect RDMA, Remote-MPI 클러스터링, Cuda-Aware MPI 등의 기술들이 사용될 수 있다.

[0040] 보다 구체적으로, Cuda-Aware MPI 기술에 기초하여, 통합 메모리(225, 265)를 통해 디바이스 메모리(245, 285)의 주소에 대한 접근이 진행될 수 있다. 또한, GPUDirect RDMA 기술에 기초하여, 데이터는 제1 CPU(210)의 제1 호스트 메모리(230)에 복사되지 않고 전자 장치(100)의 HCA(205)를 통해 원격 서버(200)의 HCA(250)에게 직접적으로 전달될 수 있다. 이때, 원격 서버(200)가 수신한 데이터는 제 m CPU(255)의 제 m 호스트 메모리(270)에 복사되지 않고, 원격 서버의 통합 메모리(265) 내에 할당된 제 m 디바이스 메모리(285)의 주소로 전달될 수 있다. 또한, Remote MPI Clustering 기술에 기초하여, 전자 장치(100)와 원격 서버(200)에서 MPI 컴파일러로 컴파일된 동일한 프로그램이 병렬적으로 실행됨으로써, 해당 데이터 전송 과정이 진행될 수 있다.

[0041] 일 실시 예에 따르면, 전자 장치(100)의 제1 CPU(210)는 CPU 마스터(215) 또는 하나 이상의 CPU 워커(220)를 포함할 수 있고, 제1 GPU(235)는 하나 이상의 GPU 워커(240)를 포함할 수 있다. 또한, 원격 서버(200)의 제 m CPU(255)는 하나 이상의 CPU 워커(260)를 포함할 수 있고, 제 m GPU(275)는 하나 이상의 GPU 워커(280)를 포함할 수 있다.

[0042] 이때, CPU 마스터(215)는 마스터 노드에 할당된 제1 CPU(210)의 부분 집합을 나타낼 수 있고, 하나 이상의 CPU 워커(220)는 워커 노드에 할당된 제1 CPU(210)의 부분 집합을 나타낼 수 있으며, 하나 이상의 GPU 워커(240)는

워커 노드에 할당된 제1 GPU(235)의 부분 집합을 나타낼 수 있다. 또한, 하나 이상의 CPU 워커(260)는 워커 노드에 할당된 제m CPU(255)의 부분 집합을 나타낼 수 있고, 하나 이상의 GPU 워커(280)는 워커 노드에 할당된 제m GPU(275)의 부분 집합을 나타낼 수 있다. 마스터 노드 및 워커 노드에 대한 보다 자세한 설명은 도 3 및 도 4를 참조하여 설명하기로 한다.

[0043] 도 3은 일 실시 예에 따른 마스터-워커 패턴 시스템의 작동 방식을 나타낸다.

[0044] 일 실시 예에 따르면, 전자 장치(100)에서 마스터 노드(320)가 생성될 수 있고, 원격 서버(300)에서 N개의 워커 노드들(340, 360, 380)이 생성될 수 있다. 이때, 노드는 논리적인 개념으로서, 노드는 각각에게 할당된 CPU 또는 GPU를 활용하여, 마스터 노드(320) 또는 워커 노드(340, 360, 380)의 역할을 수행할 수 있다.

[0045] 일 실시 예에 따르면, 전자 장치(100)는 데이터의 양, 수행될 역할의 양, 원격 서버(300) 내 GPU의 수 중 적어도 하나에 기초하여, 생성될 워커 노드의 수를 결정할 수 있다. 예를 들어, 데이터의 양이 많은 경우, 수행될 역할의 양이 많은 경우 또는 원격 서버(300) 내 GPU의 수가 많은 경우, 전자 장치(100)는 보다 많은 수의 워커 노드를 생성하는 것으로 결정할 수 있다. 또는, 데이터의 양이 적은 경우, 수행될 역할의 양이 적은 경우 또는 원격 서버(300) 내 GPU의 수가 적은 경우, 전자 장치(100)는 보다 적은 수의 워커 노드를 생성하는 것으로 결정할 수 있다. 다만, 이는 일 실시 예에 불과하며, 전자 장치(100)는 전술한 바와 상이한 방식으로 워커 노드의 수를 결정하거나, 다른 요인에 기초하여 워커 노드의 수를 결정할 수 있다.

[0046] 일 실시 예에 따르면, 마스터 노드(320)는 자신에게 할당된 CPU 및 호스트 메모리를 활용하여, 대용량 데이터와 수행할 역할을 분할하고, 분할된 데이터와 역할을 HCA를 통해 각각의 워커 노드들(340, 360, 380)에게 분배할 수 있다.

[0047] 이때, 마스터 노드(320)는 워커 노드들(340, 360, 380) 각각에 대응하는 GPU의 수 및 성능 중 적어도 하나에 기초하여, 워커 노드들(340, 360, 380)에게 할당할 대용량 데이터와 수행할 역할을 분할할 수 있다. 예를 들어, 마스터 노드(320)는 보다 많은 수의 GPU가 할당된 워커 노드 또는 보다 높은 성능의 GPU가 할당된 워커 노드에 많은 양의 데이터 및 역할을 분배할 수 있다. 또는, 마스터 노드(320)는 보다 적은 수의 GPU가 할당된 워커 노드 또는 보다 낮은 성능의 GPU가 할당된 워커 노드에 적은 양의 데이터 및 역할을 분배할 수 있다.

[0048] 일 실시 예에 따르면, 워커(worker) 노드들(340, 360, 380)은 HCA를 통해 마스터 노드(320)로부터 대용량 데이터 및 수행할 역할을 수신하고, 자신에게 할당된 CPU, GPU 및 디바이스 메모리를 활용하여 할당 받은 역할을 수행할 수 있다.

[0049] 일 실시 예에 따르면, MPI 병렬 프로그래밍에서 실행되는 프로세스들은 Rank 0부터 병렬 수행되는 프로세스의 수까지 각각의 Rank 번호가 부여될 수 있다. 이에 따라, 마스터 노드(320)는 Rank 0의 프로세스를 수행하고, 제1 워커 노드(340)는 Rank 1의 프로세스를 수행하며, 제2 워커 노드(360)는 Rank 2의 프로세스를 수행하고, 제N 워커 노드(380)는 Rank N의 프로세스를 수행할 수 있다.

[0050] 도 4는 일 실시 예에 따른 마스터-워커 패턴 시스템의 작동 방식을 나타낸다. 도 3과 관련하여 전술한 내용과 중복되는 내용은 생략하거나 간략히 설명하기로 한다.

[0051] 일 실시 예에 따르면, 전자 장치(100)에서 마스터 노드(400) 및 (N-1)개의 워커 노드들(420, 440, 460)이 생성될 수 있다. 즉, 전자 장치(100)에서 마스터 노드(320)만이 생성되었던 도 3의 시스템과는 달리, 도 4의 시스템에서는 마스터 노드(400)뿐만 아니라 워커 노드들(420, 440, 460)도 전자 장치(100)에서 생성될 수 있다. 이를 통해, 마스터 노드가 데이터 및 역할을 분배하기 위해 사용된 전자 장치(100)의 하드웨어 자원이 워커 노드가 분배된 역할을 수행하기 위해서도 사용될 수 있다.

[0052] 일 실시 예에 따르면, 전자 장치(100)는 데이터의 양, 수행될 역할의 양, 전자 장치(100) 내 GPU의 수 중 적어도 하나에 기초하여, 생성될 워커 노드의 수를 결정할 수 있다. 예를 들어, 데이터의 양이 많은 경우, 수행될 역할의 양이 많은 경우 또는 전자 장치(100) 내 GPU의 수가 많은 경우, 전자 장치(100)는 보다 많은 수의 워커 노드를 생성하는 것으로 결정할 수 있다. 또는, 데이터의 양이 적은 경우, 수행될 역할의 양이 적은 경우 또는 전자 장치(100) 내 GPU의 수가 적은 경우, 전자 장치(100)는 보다 적은 수의 워커 노드를 생성하는 것으로 결정할 수 있다. 다만 이는 일 실시 예에 불과하며, 전자 장치(100)는 전술한 바와 상이한 방식으로 워커 노드의 수를 결정하거나, 다른 요인에 기초하여 워커 노드의 수를 결정할 수 있다.

[0053] 일 실시 예에 따르면, 마스터 노드(400)는 자신에게 할당된 CPU 및 호스트 메모리를 활용하여, 대용량 데이터와 수행할 역할을 분할하고, 분할된 데이터와 역할을 각각의 워커 노드들(420, 440, 460)에게 분배할 수 있다.

- [0054] 이때, 마스터 노드(400)는 워커 노드들(420, 440, 460) 각각에 대응하는 GPU의 수 및 성능 중 적어도 하나에 기초하여, 워커 노드들(420, 440, 460)에게 할당할 대용량 데이터와 수행할 역할을 분할할 수 있다. 예를 들어, 마스터 노드(400)는 보다 많은 수의 GPU가 할당된 워커 노드 또는 보다 높은 성능의 GPU가 할당된 워커 노드에 많은 양의 데이터 및 역할을 분배할 수 있다. 또는, 마스터 노드(400)는 보다 적은 수의 GPU가 할당된 워커 노드 또는 보다 낮은 성능의 GPU가 할당된 워커 노드에게 적은 양의 데이터 및 역할을 분배할 수 있다.
- [0055] 일 실시 예에 따르면, 워커 노드들(420, 440, 460)은 마스터 노드(400)로부터 대용량 데이터 및 수행할 역할을 수신하고, 자신에게 할당된 CPU, GPU 및 디바이스 메모리를 활용하여 할당 받은 역할을 수행할 수 있다.
- [0056] 일 실시 예에 따르면, 시스템 내에 전자 장치(100)를 제외한 (M-1)개의 원격 서버들이 존재하고, 각각의 서버가 최대 N개의 워커 노드들을 생성할 수 있는 경우, 전자 장치(100)에서 최대 (N-1)개의 워커 노드들이 생성될 수 있고, (M-1)개의 원격 서버에서 최대 N*(M-1)개의 워커 노드들이 생성될 수 있다. 이에 따라, 시스템 내에서 최대 (N*M-1)개의 워커 노드들이 생성될 수 있고, 최대 (N*M-1)개의 워커 노드들은 각각 Rank 1의 프로세스부터 Rank N*M-1의 프로세스를 수행할 수 있다.
- [0057] 일 실시 예에 따르면, 도 4와 같이 설계된 마스터-워커 패턴의 시스템에서 전수 조사(brute-force) 암호 분석이 수행될 수 있다. 보다 구체적으로, 마스터 노드(400)는 분석하고자 하는 대용량의 암호문을 최대 (N*M-1)개의 워커 노드들의 수로 분할하고 워커 노드들에게 분배할 수 있다. 각각의 워커 노드는 분배 받은 암호문을 병렬적으로 해독하여 목표하는 평문을 찾을 수 있다.
- [0058] 도 5는 일 실시 예에 따른 마스터-워커 패턴 시스템의 작동 방식을 나타낸다. 도 3 및 도 4와 관련하여 기술한 내용과 중복되는 내용은 생략하거나 간략히 설명하기로 한다.
- [0059] 일 실시 예에 따르면, 시스템 내에 전자 장치(100)를 제외한 (M-1)개의 원격 서버들이 존재하고, 각각의 서버가 최대 N개의 워커 노드를 생성할 수 있는 경우, 전자 장치(100)에서 1개의 마스터 노드와 최대 (N-1)개의 워커 노드들이 생성될 수 있고, (M-1)개의 원격 서버에서 최대 N*(M-1)개의 워커 노드들이 생성될 수 있다. 이에 따라, 시스템 내에서 1개의 마스터 노드와 최대 (N*M-1)개의 워커 노드들이 생성될 수 있고, 마스터 노드는 Rank 0의 프로세스를 수행하고, 최대 (N*M-1)개의 워커 노드들은 각각 Rank 1의 프로세스부터 Rank N*M-1의 프로세스를 수행할 수 있다.
- [0060] 일 실시 예에 따르면, 워커 노드들 간 상호 작용이 요구되는 경우, 시스템은 도 5에 도시된 바와 같이 설계될 수 있다. 예를 들어, 원격 서버(500)에서 생성된 N개의 워커 노드 각각에게 제m CPU의 일부가 할당될 수 있고, N개의 워커 노드 각각에게 할당된 제m CPU의 부분 집합 간에 또는 N개의 CPU 워커 간에 데이터가 송수신될 수 있다.
- [0061] 일 실시 예에 따르면, 전자 장치(100)에서 생성된 마스터 노드는 Rank 0의 프로세스를 수행하여, 최대 (N*M-1)개의 워커 노드들에게 대용량의 데이터와 일을 분배할 수 있다. 전자 장치(100) 및 원격 서버(500)를 포함한 최대 M개의 서버에서 생성된 최대 (N*M-1)개의 워커 노드들은 각각 Rank 1의 프로세스부터 Rank N*M-1의 프로세스를 수행하여, 특정 암호 분석 연산을 수행하고 해당 결과 값을 워커 노드들끼리 송·수신하여 최종적인 암호 분석을 수행할 수 있다. 이때, 다른 서버 내에 할당된 워커 노드들끼리 연산 결과 값을 송·수신하는 과정에서 HCA를 활용한 GPUDirect RDMA 기술이 활용되어, 보다 빠른 데이터 전송으로 고성능의 암호 분석이 수행될 수 있다.
- [0062] 일 실시 예에 따르면, 도 5와 같이 설계된 마스터-워커 패턴의 시스템에서, 워커 노드들의 독립적인 연산 수행과 더불어 워커 노드들끼리의 상호 작용이 요구되는 행렬 곱 연산 고속화를 활용한 암호 분석이 수행될 수 있다. 보다 구체적으로, 마스터 노드는 큰 스케일의 행렬 연산을 최대 (N*M-1)개의 워커 노드의 수로 분할하여 각각의 워커들에게 분배할 수 있다. 워커 노드들은 분배된 작은 행렬 연산 문제를 해결하고, 그 결과 값을 사용해 더 큰 행렬 연산 문제를 해결하는 워커 노드에게 해당 결과 값을 송신할 수 있다. 이와 같은 연산 과정이 반복되어 최종적으로 목표하는 큰 스케일의 행렬 문제가 해결될 수 있다.
- [0063] 도 6a 내지 도 6c는 일 실시 예에 따른 전자 장치(100)가 암호를 분석하는 과정을 설명하기 위한 도면이다.
- [0064] 일 실시 예에 따르면, 시스템 내에 전자 장치(100)를 제외한 (M-1)개의 원격 서버들이 존재하고, 각각의 서버가 최대 N개의 워커 노드를 생성할 수 있는 경우, 전자 장치(100)에서 1개의 마스터 노드(600)와 최대 (N-1)개의 워커 노드들이 생성될 수 있고, (M-1)개의 원격 서버에서 최대 N*(M-1)개의 워커 노드들이 생성될 수 있다. 이에 따라, 시스템 내에서 1개의 마스터 노드(600)와 최대 (N*M-1)개의 워커 노드들(610, 620, 630, 640)이 생성

될 수 있고, 마스터 노드(600)는 Rank 0의 프로세스를 수행하고, 최대 $(N*M-1)$ 개의 워커 노드들(610, 620, 630, 640)은 각각 Rank 1의 프로세스부터 Rank $N*M-1$ 의 프로세스를 수행할 수 있다.

- [0065] 일 실시 예에 따르면, 전술한 시스템에서 패스워드 크래킹(password cracking)의 암호 분석 기술이 수행될 수 있다. 패스워드 크래킹은 정답이 될 수 있는 모든 후보 평문에 대한 암호화를 진행하고, 순차적인 비교를 통해 목표하는 암호문과 동일한 암호 값을 탐색하여, 최종적으로 암호화가 되지 않는 정답 평문을 찾기 위한 암호 분석 기술을 나타낼 수 있다.
- [0066] 예를 들어, 전자 장치(100) 또는 마스터 노드(600)는 복수의 후보 평문들의 집합인 딕셔너리(dictionary) 파일, 해독하고자 하는 암호문 및 하나 이상의 암호화 알고리즘을 소유할 수 있다. 암호문을 복호화하여 얻어지는 평문을 획득하기 위하여, 전자 장치(100) 또는 마스터 노드(600)는 딕셔너리 파일을 워커 노드들(610, 620, 630, 640)의 수에 기초하여 분할하고, 분할된 딕셔너리 파일 및 암호문을 워커 노드들(610, 620, 630, 640)에게 분배할 수 있다. 워커 노드들(610, 620, 630, 640)은 수신한 딕셔너리 파일에 포함된 후보 평문들을 암호화 알고리즘에 따라 암호화하고, 암호화된 결과 값이 암호문과 동일한지 여부를 확인할 수 있다. 워커 노드들(610, 620, 630, 640)이 암호문과 동일한 결과 값을 확인하는 경우, 해당 결과 값에 대응하는 후보 평문이 출력됨으로써 암호 분석 작업이 중단될 수 있다.
- [0067] 전자 장치(100) 및 하나 이상의 원격 서버에 의해 패스워드 크래킹의 암호 분석 기술이 수행되는 보다 구체적인 과정은 다음과 같다.
- [0068] 일 실시 예에 따르면, 전자 장치(100) 또는 마스터 노드(600)는 전자 장치(100) 및 하나 이상의 원격 서버 중 적어도 하나에 대응하는 워커 노드들(610, 620, 630, 640)을 확인할 수 있다. 이때, 전자 장치(100) 또는 마스터 노드(600)는 복수의 후보 평문들의 수, 하나 이상의 암호화 알고리즘의 수, 하나 이상의 원격 서버의 수, 전자 장치 내 GPU의 수 및 하나 이상의 원격 서버 내 GPU의 수 중 적어도 하나에 기초하여, 하나 이상의 워커 노드의 수를 결정할 수 있다.
- [0069] 예를 들어, 후보 평문들의 수가 많은 경우, 암호화 알고리즘의 수가 많은 경우, 원격 서버의 수가 많은 경우, 전자 장치(100) 내 GPU의 수가 많은 경우 또는 원격 서버 내 GPU의 수가 많은 경우, 전자 장치(100) 또는 마스터 노드(600)는 보다 많은 수의 워커 노드를 생성하는 것으로 결정할 수 있다. 또는, 후보 평문들의 수가 적은 경우, 암호화 알고리즘의 수가 적은 경우, 원격 서버의 수가 적은 경우, 전자 장치(100) 내 GPU의 수가 적은 경우 또는 원격 서버 내 GPU의 수가 적은 경우, 전자 장치(100) 또는 마스터 노드(600)는 보다 적은 수의 워커 노드를 생성하는 것으로 결정할 수 있다. 다만, 이는 일 실시 예에 불과하며, 전자 장치(100) 또는 마스터 노드(600)는 전술한 바와 상이한 방식으로 워커 노드의 수를 결정하거나, 다른 요인에 기초하여 워커 노드의 수를 결정할 수 있다.
- [0070] 일 실시 예에 따르면, 전자 장치(100) 또는 마스터 노드(600)는 암호문을 워커 노드들(610, 620, 630, 640)에게 브로드캐스팅할 수 있다. 예를 들어, 도 6a를 참조하면, 전자 장치(100) 또는 마스터 노드(600)는 MPI 브로드캐스팅 API(MPI_Bcast())를 통해, 워커 노드들(610, 620, 630, 640)에게 암호문을 브로드캐스팅할 수 있다.
- [0071] 일 실시 예에 따르면, 전자 장치(100) 또는 마스터 노드(600)는 워커 노드들(610, 620, 630, 640)의 수에 기초하여, 복수의 후보 평문들을 분할할 수 있다. 예를 들어, $(N*M-1)$ 개의 워커 노드들이 존재하는 경우, 전자 장치(100) 또는 마스터 노드(600)는 전체 딕셔너리 파일을 통합 메모리로 로드한 후, 로드된 딕셔너리 파일을 $(N*M-1)$ 개로 분할할 수 있다.
- [0072] 일 실시 예에 따르면, 전자 장치(100) 또는 마스터 노드(600)는 워커 노드들(610, 620, 630, 640)에 대응하는 GPU의 수 및 성능 중 적어도 하나에 기초하여, 복수의 후보 평문들을 분할할 수 있다. 예를 들어, 전자 장치(100) 또는 마스터 노드(600)는 보다 많은 수의 GPU가 할당된 워커 노드 또는 보다 높은 성능의 GPU가 할당된 워커 노드에게 많은 양의 후보 평문을 할당할 수 있다. 또는, 전자 장치(100) 또는 마스터 노드(600)는 보다 적은 수의 GPU가 할당된 워커 노드 또는 보다 낮은 성능의 GPU가 할당된 워커 노드에게 적은 양의 후보 평문을 할당할 수 있다.
- [0073] 일 실시 예에 따르면, 전자 장치(100) 또는 마스터 노드(600)는 분할된 딕셔너리 파일을 워커 노드들(610, 620, 630, 640)에게 송신할 수 있다. 예를 들어, 도 6b를 참조하면, 전자 장치(100) 또는 마스터 노드(600)는, MPI 송신 API(MPI_Isend())를 통해 분할된 딕셔너리 파일이 로드된 통합 메모리의 주소 값을 워커 노드들(610, 620, 630, 640)에게 각각으로 송신할 수 있다.
- [0074] S650 단계에서, 워커 노드들(610, 620, 630, 640)은 일 실시 예에 따라 전자 장치(100) 또는 마스터 노드(600)

0)로부터 분할된 디렉터리 파일을 수신할 수 있다. 예를 들어, 도 6b를 참조하면, 워커 노드들(610, 620, 630, 640)은 MPI 수신 API(MPI_Recv())를 통해 분할된 디렉터리 파일이 로드된 통합 메모리의 주소 값을 수신할 수 있다.

- [0075] S660 단계에서, 워커 노드들(610, 620, 630, 640)은 수신한 디렉터리 파일, 암호문 및 하나 이상의 암호화 알고리즘에 수에 기초하여, 암호 분석을 수행할 수 있다. 예를 들어, 워커 노드들(610, 620, 630, 640)은 수신한 디렉터리 파일에 포함된 후보 평문들을 하나 이상의 암호화 알고리즘에 따라 암호화하고, 암호화된 결과 값이 암호문과 일치하는지 여부를 확인할 수 있으며, 상기 과정은 일치하는 평문을 찾을 때까지 반복될 수 있다.
- [0076] S670 단계에서, 워커 노드들(610, 620, 630, 640) 중 하나의 워커 노드가 후보 평문들을 암호화한 결과 값들 중 암호문과 일치하는 결과 값을 확인한 경우, 하나의 워커 노드는 나머지 워커 노드로 작업 중단 요청을 송신하고, 모든 워커 노드의 작업이 중단될 수 있다. 예를 들어, 워커 노드들(610, 620, 630, 640) 중 하나의 워커 노드가 암호문을 복호화한 평문을 확인한 경우, 하나의 워커 노드는 MPI 중단 API(MPI_Abort())를 통해 나머지 워커 노드로 작업 중단 요청을 송신하고, 모든 워커 노드의 암호 분석 작업이 중단될 수 있다.
- [0077] S680 단계에서, 전자 장치(100) 또는 마스터 노드(600)는 암호문과 일치하는 결과 값에 대응하는 후보 평문을 출력할 수 있다. 예를 들어, 전자 장치(100) 또는 마스터 노드(600)는 정답 평문을 확인한 워커 노드로부터 정답 평문에 관한 정보를 수신하고, 정답 평문을 출력할 수 있다.
- [0078] 도 7a 내지 도 7c는 전자 장치(100)가 일 실시 예에 따라 암호 분석을 수행한 결과를 설명하기 위한 도면이다.
- [0079] 도 7a 내지 도 7c는 본 개시에 따라 설계된 시스템(700)을 이용하여 패스워드 크래킹의 암호 분석 기술이 수행되었을 때의 소요된 시간을 도시한다. 또한, 본 개시에 따라 설계된 시스템(700)의 암호 분석 결과의 성능을 검증하기 위하여, 기존의 시스템(750)을 이용하여 패스워드 크래킹의 암호 분석 기술이 수행되었을 때의 소요된 시간이 함께 도시된다.
- [0080] 도 7a를 참조하면, 마스터 노드가 워커 노드에게 분배하는 데이터의 크기에 따른, 암호 분석 시스템(700, 750)이 암호를 분석하는데 소요된 시간의 비교 결과가 도시된다. 보다 구체적으로, 시스템(700, 750)에서 사용되는 워커 노드의 수 및 작업해야 할 일의 수가 동일할 때, 마스터 노드가 워커 노드에게 분배하는 데이터의 크기에 따른 성능 비교 결과가 도시된다.
- [0081] 분배되는 데이터의 양이 증가할수록 전송할 데이터의 양이 증가하므로, 본 개시에 따라 설계된 시스템(700)과 기존의 시스템(750)을 이용한 경우 모두 암호 분석 소요 시간이 증가한다. 다만, 본 개시에 따라 설계된 시스템(700)을 이용하는 경우, 최적화된 데이터 전송에 대한 효과를 통해 기존의 시스템(750)에 비해 최대 약 2배 이상 암호 분석 소요 시간이 감소하는 기술적 효과를 얻을 수 있다.
- [0082] 도 7b를 참조하면, 시스템(700, 750)에서 사용되는 워커 노드의 수에 따른, 암호 분석 시스템(700, 750)이 암호를 분석하는데 소요된 시간의 비교 결과가 도시된다. 보다 구체적으로, 워커 노드 별 분배되는 데이터의 크기와 작업해야 할 일의 수가 동일할 때, 암호 분석 시스템(700, 750) 내에서 구성한 워커 노드의 수에 따른 성능 비교 결과가 도시된다.
- [0083] 시스템(700, 750)이 사용하는 워커 노드의 수가 증가할수록, 더 많은 수의 워커 노드들에게 대용량의 데이터를 분배하고, 그에 따른 프로세스를 처리해야 하므로, 암호 분석 소요 시간이 증가하는 양상을 확인할 수 있다. 다만, 본 개시에 따라 설계된 시스템(700)을 이용하는 경우, 약 9초에서 12초 내외의 암호 분석 시간이 소요되는 반면, 기존의 시스템(750)을 이용하는 경우 약 21초에서 27초 사이의 암호 분석 시간이 소요되는 것을 확인할 수 있다.
- [0084] 도 7c를 참조하면, 워커 노드가 작업해야 할 일에 따른, 암호 분석 시스템(700, 750)이 암호를 분석하는데 소요된 시간의 비교 결과가 도시된다. 보다 구체적으로, 마스터 노드가 워커 노드에게 분배하는 데이터의 크기와 시스템(700, 750)에서 사용되는 워커 노드의 수가 동일할 때, 워커 노드가 작업해야 할 일의 수에 따른 성능 비교 결과가 도시된다. 이때, 워커 노드가 작업해야 할 일의 수는 패스워드 크래킹의 암호 분석 기술이 수행될 때 필요한 후보 평문들의 수로 표현될 수 있다.
- [0085] 워커 노드 별 수행해야 할 작업의 수가 증가할수록 암호 분석 소요 시간이 더 많이 요구되므로, 전체적인 암호 분석 소요 시간은 워커 노드의 작업 수가 증가할수록 증가하는 양상을 확인할 수 있다. 다만, 본 개시에 따라 설계된 시스템(700)을 이용하는 경우, 최적화된 데이터 전송에 대한 효과를 통해 기존의 시스템(750)에 비해 최대 약 2배 이상 암호 분석 소요 시간이 감소하는 기술적 효과를 얻을 수 있다.

- [0086] 도 8은 일 실시 예에 따른 전자 장치의 암호 분석 방법의 흐름도를 나타낸다. 중복되는 내용에 대해서는 전술한 기재가 적용될 수 있다.
- [0087] S800 단계에서, 전자 장치는 전자 장치 및 하나 이상의 원격 서버 중 적어도 하나에 대응하는 하나 이상의 워커 노드를 확인할 수 있다.
- [0088] 일 실시 예에 따르면, 전자 장치는 하나 이상의 워커 노드를 확인할 때, 복수의 후보 평문들의 수, 하나 이상의 암호화 알고리즘의 수, 하나 이상의 원격 서버의 수, 전자 장치 내 GPU의 수 및 하나 이상의 원격 서버 내 GPU의 수 중 적어도 하나에 기초하여, 하나 이상의 워커 노드의 수를 결정할 수 있다.
- [0089] S820 단계에서, 전자 장치는 하나 이상의 워커 노드의 수에 기초하여, 복수의 후보 평문들을 분할할 수 있다.
- [0090] 일 실시 예에 따르면, 전자 장치는 복수의 후보 평문들을 분할할 때, 하나 이상의 워커 노드에 대응하는 GPU의 수 및 성능 중 적어도 하나에 기초하여, 하나의 워커 노드 각각에 복수의 후보 평문들을 할당할 수 있다.
- [0091] S840 단계에서, 전자 장치는 분할된 복수의 후보 평문들 및 암호문을 하나 이상의 워커 노드 각각으로 송신할 수 있다.
- [0092] 일 실시 예에 따르면, 복수의 후보 평문들은 하나 이상의 워커 노드에 의해 하나 이상의 암호화 알고리즘에 따라 암호화되어, 암호화된 결과 값이 암호문과 일치하는지 여부가 확인될 수 있다.
- [0093] 일 실시 예에 따르면, 전자 장치는 분할된 복수의 후보 평문들 및 암호문을 하나 이상의 워커 노드 각각으로 송신할 때, 분할된 복수의 후보 평문들을 통합 메모리로 로드하고, 메시지 전달 인터페이스를 통해, 분할된 복수의 후보 평문들이 로드된 통합 메모리의 주소 값을 하나 이상의 워커 노드 각각으로 송신할 수 있다.
- [0094] 일 실시 예에 따르면, 전자 장치는 통합 메모리의 주소 값을 하나 이상의 워커 노드 각각으로 송신할 때, MPI 송신 애플리케이션 프로그래밍 인터페이스를 통해 통합 메모리의 주소 값을 송신할 수 있다. 이때, 통합 메모리의 주소 값은, MPI 수신 API를 통해 하나 이상의 워커 노드에 의해 수신될 수 있다.
- [0095] 일 실시 예에 따르면, 전자 장치는 분할된 복수의 후보 평문들 및 암호문을 하나 이상의 워커 노드 각각으로 송신할 때, MPI 브로드캐스팅 API를 통해, 하나 이상의 워커 노드에게 암호문을 브로드캐스팅할 수 있다.
- [0096] 일 실시 예에 따르면, 하나 이상의 워커 노드 중 하나의 워커 노드가 암호문과 일치하는 결과 값을 확인한 경우, 하나의 워커 노드로부터 나머지 워커 노드로 작업 중단 요청이 송신될 수 있다. 이때, 작업 중단 요청은 MPI 중단 API를 통해 송신될 수 있다.
- [0097] 도 9는 일 실시 예에 따른 전자 장치(100)의 블록도를 나타낸다.
- [0098] 전자 장치(100)는 일 실시 예에 따라, 메모리(900) 및 제어부(controller)(950)를 포함할 수 있다. 도 9에 도시된 전자 장치(100)는 본 실시 예와 관련된 구성요소들만이 도시되어 있다. 따라서, 도 9에 도시된 구성요소들 외에 다른 범용적인 구성요소들이 더 포함될 수 있음을 본 실시 예와 관련된 기술분야에서 통상의 지식을 가진 자라면 이해할 수 있다. 예를 들어, 전자 장치(100)는 하나 이상의 트랜시버(transceiver)를 포함하는 통신부(communication device)(미도시)를 포함할 수 있다. 통신부는 유/무선 통신을 수행하기 위한 장치로서, 외부의 전자 장치와 통신할 수 있다. 외부의 전자 장치는 단말 또는 서버가 될 수 있다. 또한, 통신부가 이용하는 통신 기술에는 GSM(Global System for Mobile communication), CDMA(Code Division Multi Access), LTE(Long Term Evolution), 5G, WLAN(Wireless LAN), Wi-Fi(Wireless-Fidelity), 블루투스(Bluetooth?), RFID(Radio Frequency Identification), 적외선 통신(Infrared Data Association; IrDA), ZigBee, NFC(Near Field Communication) 등이 있을 수 있다.
- [0099] 제어부(950)는 전자 장치(100)의 전반의 동작을 제어하고 데이터 및 신호를 처리할 수 있다. 제어부(950)는 적어도 하나의 하드웨어 유닛으로 구성될 수 있다. 또한, 제어부(950)는 메모리(900)에 저장된 프로그램 코드를 실행하여 생성되는 하나 이상의 소프트웨어 모듈에 의해 동작할 수 있다. 제어부(950)는 프로세서 및 메모리를 포함할 수 있는 바, 프로세서는 메모리에 저장된 프로그램 코드를 실행하여 전자 장치(100)의 전반의 동작을 제어하고 데이터 및 신호를 처리할 수 있다. 또한 실시 예에서 제어부(950)는 적어도 하나의 프로세서를 포함할 수 있다.
- [0100] 제어부(950)는 전자 장치 및 하나 이상의 원격(remote) 서버 중 적어도 하나에 대응하는 하나 이상의 워커(worker) 노드를 확인하고, 하나 이상의 워커 노드의 수에 기초하여, 복수의 후보 평문들을 분할하고, 분할된 복수의 후보 평문들 및 암호문을 하나 이상의 워커 노드 각각으로 송신할 수 있고, 복수의 후보 평문들은 하나

이상의 워커 노트에 의해 하나 이상의 암호화 알고리즘에 따라 암호화되어, 암호화된 결과 값이 암호문과 일치하는지 여부가 확인될 수 있다.

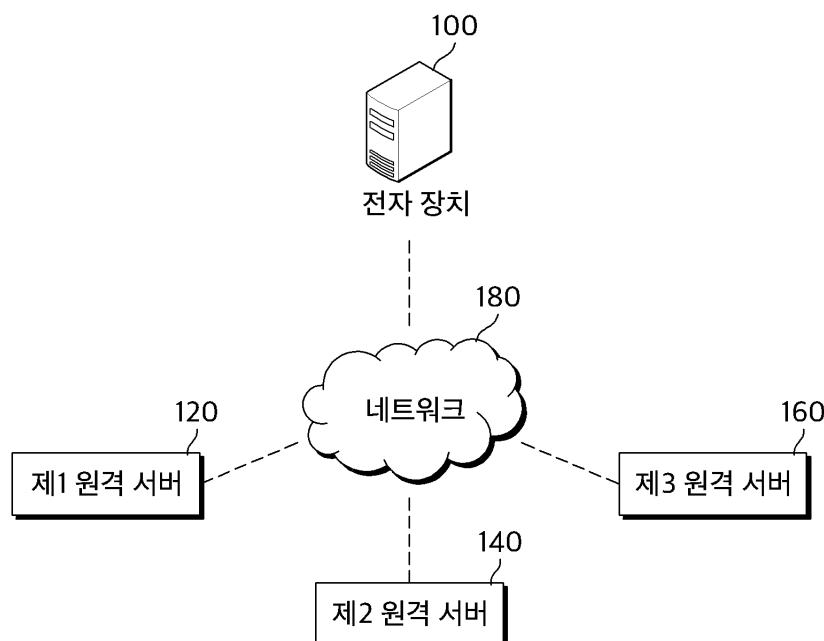
[0101] 전술한 실시 예들에 따른 전자 장치는 프로세서, 프로그램 데이터를 저장하고 실행하는 메모리, 디스크 드라이브와 같은 영구 저장부(permanent storage), 외부 장치와 통신하는 통신 포트, 터치 패널, 키(key), 버튼 등과 같은 사용자 인터페이스 장치 등을 포함할 수 있다. 소프트웨어 모듈 또는 알고리즘으로 구현되는 방법들은 상기 프로세서상에서 실행 가능한 컴퓨터가 읽을 수 있는 코드들 또는 프로그램 명령들로서 컴퓨터가 읽을 수 있는 기록 매체 상에 저장될 수 있다. 여기서 컴퓨터가 읽을 수 있는 기록 매체로 마그네틱 저장 매체(예컨대, ROM(read-only memory), RAM(random-access memory), 플로피 디스크, 하드 디스크 등) 및 광학적 판독 매체(예컨대, 시디롬(CD-ROM), 디브이디(DVD: Digital Versatile Disc)) 등이 있다. 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템들에 분산되어, 분산 방식으로 컴퓨터가 판독 가능한 코드가 저장되고 실행될 수 있다. 매체는 컴퓨터에 의해 판독가능하며, 메모리에 저장되고, 프로세서에서 실행될 수 있다.

[0102] 본 실시 예는 기능적인 블록 구성들 및 다양한 처리 단계들로 나타내어질 수 있다. 이러한 기능 블록들은 특정 기능들을 실행하는 다양한 개수의 하드웨어 또는/및 소프트웨어 구성들로 구현될 수 있다. 예를 들어, 실시 예는 하나 이상의 마이크로프로세서들의 제어 또는 다른 제어 장치들에 의해서 다양한 기능들을 실행할 수 있는, 메모리, 프로세싱, 로직(logic), 룩 업 테이블(look-up table) 등과 같은 직접 회로 구성들을 채용할 수 있다. 구성 요소들이 소프트웨어 프로그래밍 또는 소프트웨어 요소들로 실행될 수 있는 것과 유사하게, 본 실시 예는 데이터 구조, 프로세스들, 루틴들 또는 다른 프로그래밍 구성들의 조합으로 구현되는 다양한 알고리즘을 포함하여, C, C++, 자바(Java), 어셈블러(assembler) 등과 같은 프로그래밍 또는 스크립팅 언어로 구현될 수 있다. 기능적인 측면들은 하나 이상의 프로세서들에서 실행되는 알고리즘으로 구현될 수 있다. 또한, 본 실시 예는 전자적인 환경 설정, 신호 처리, 및/또는 데이터 처리 등을 위하여 종래 기술을 채용할 수 있다. "매커니즘", "요소", "수단", "구성"과 같은 용어는 넓게 사용될 수 있으며, 기계적이고 물리적인 구성들로서 한정되는 것은 아니다. 상기 용어는 프로세서 등과 연계하여 소프트웨어의 일련의 처리들(routines)의 의미를 포함할 수 있다.

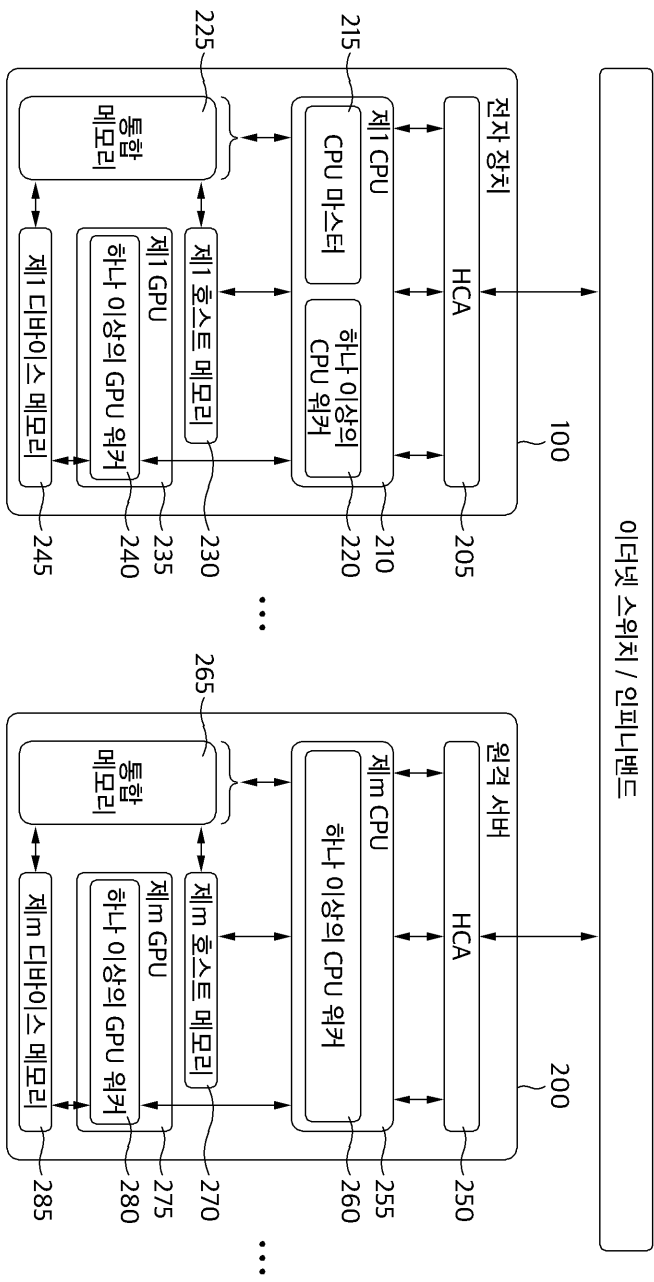
[0103] 전술한 실시 예들은 일 예시일 뿐 후술하는 청구항들의 범위 내에서 다른 실시 예들이 구현될 수 있다.

도면

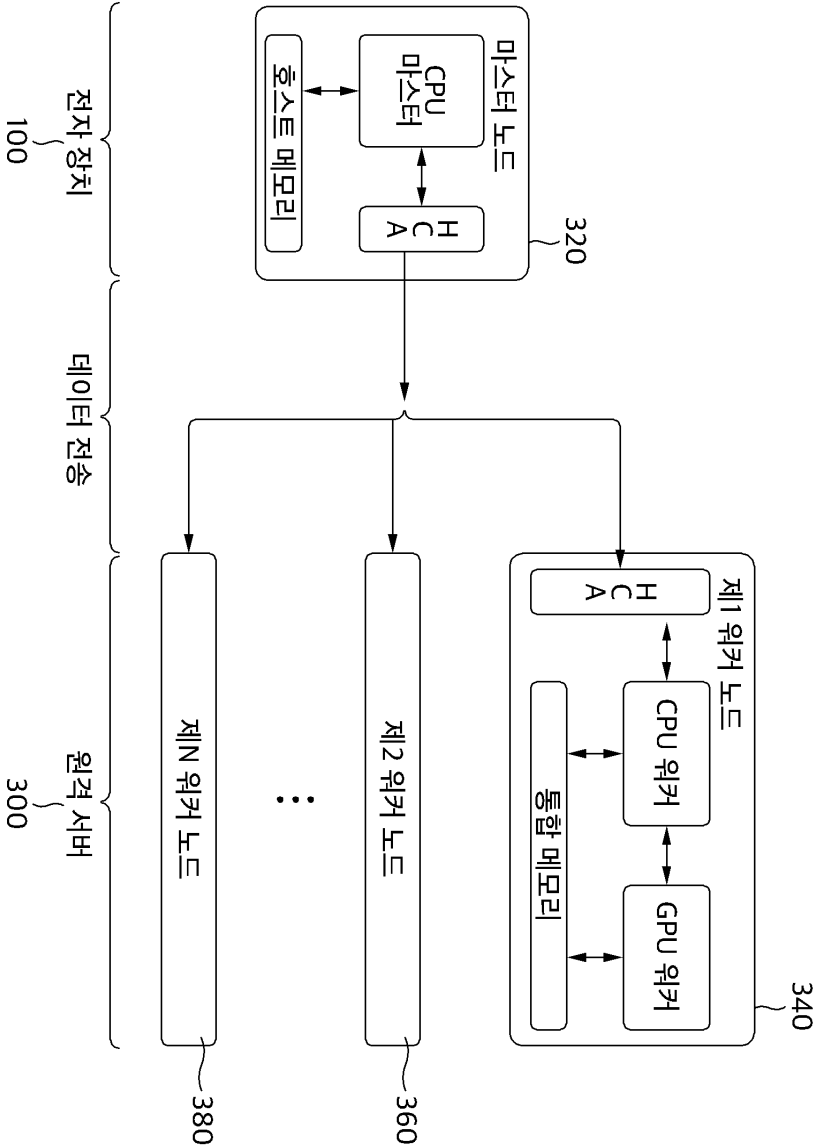
도면1

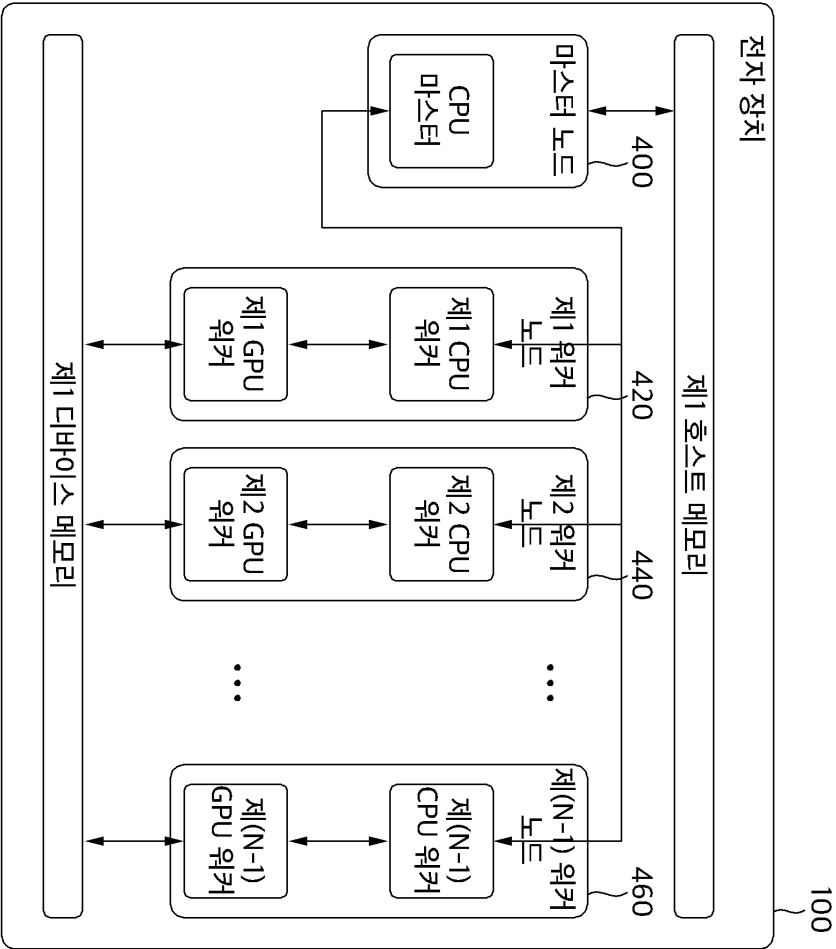


도면2



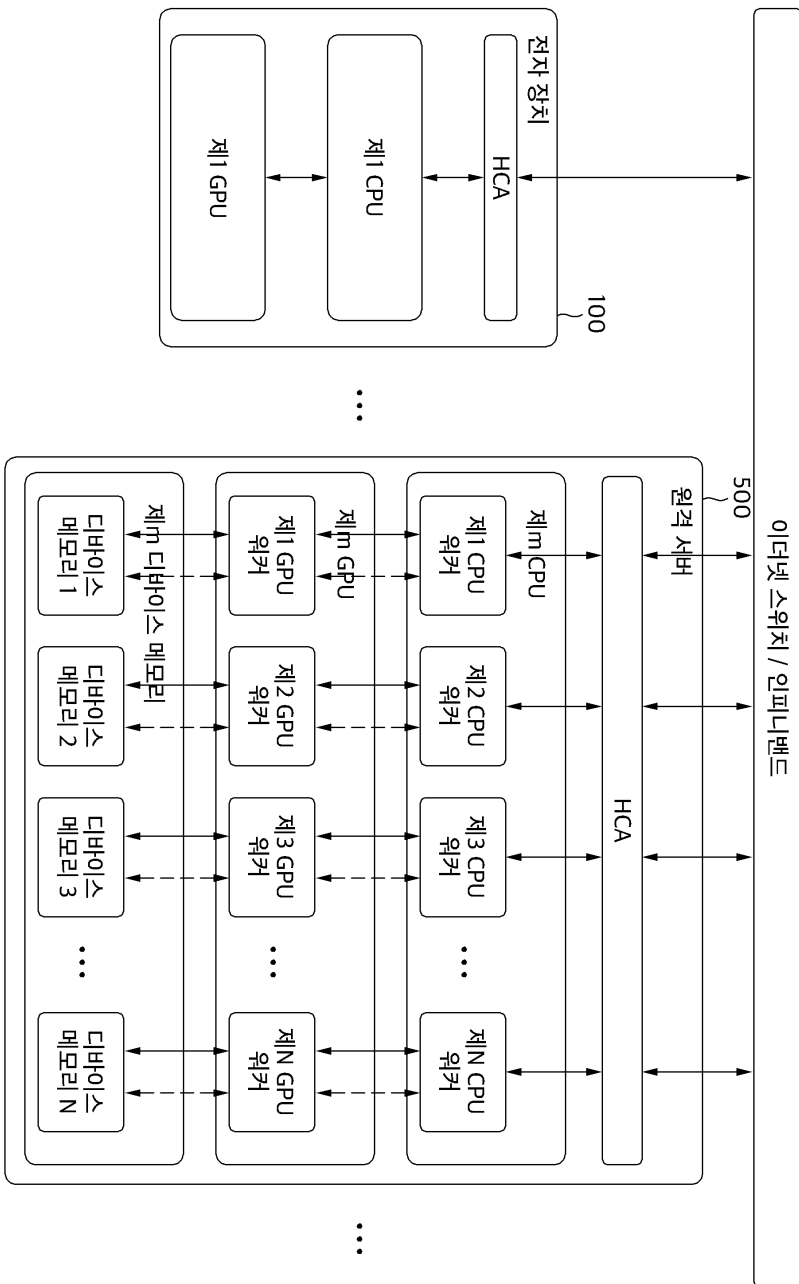
도면3



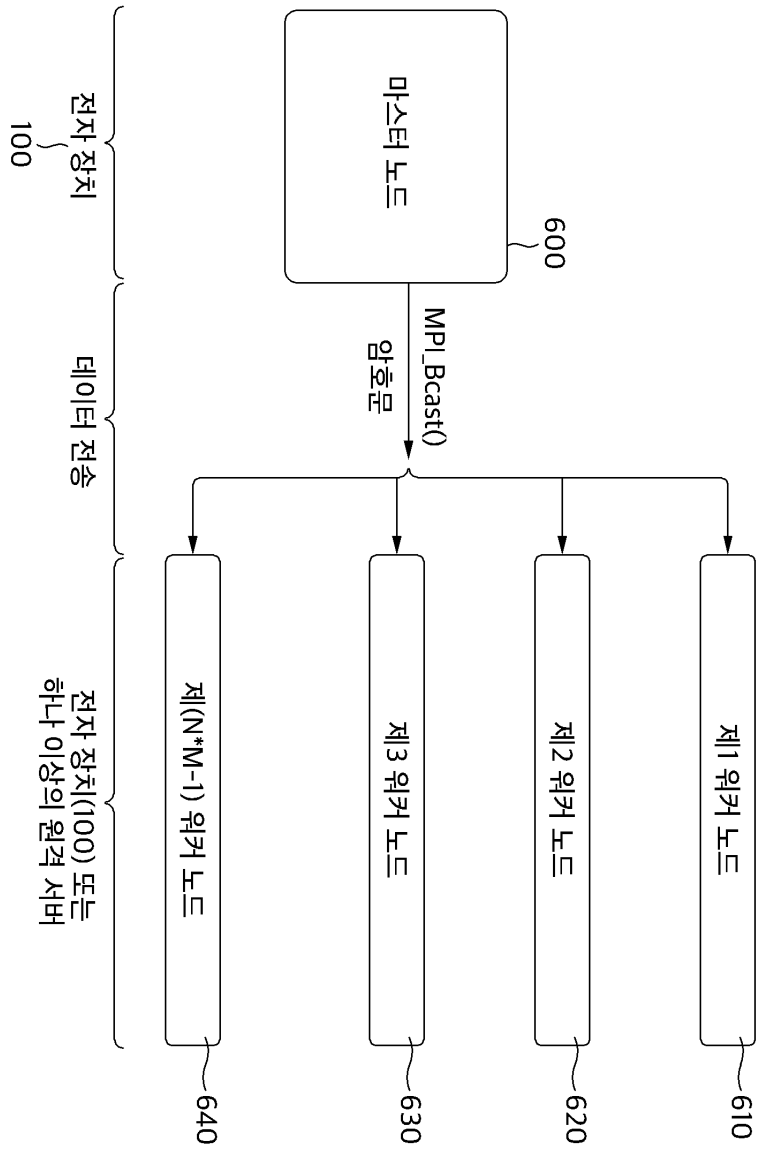


도면4

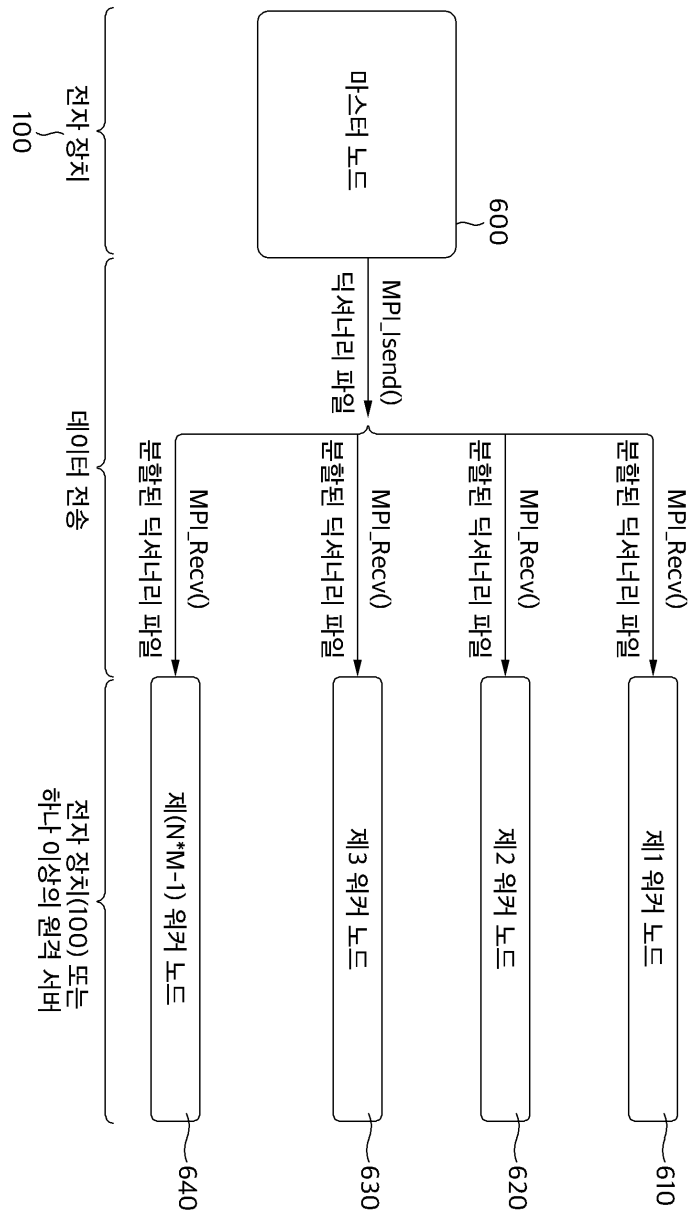
도면5



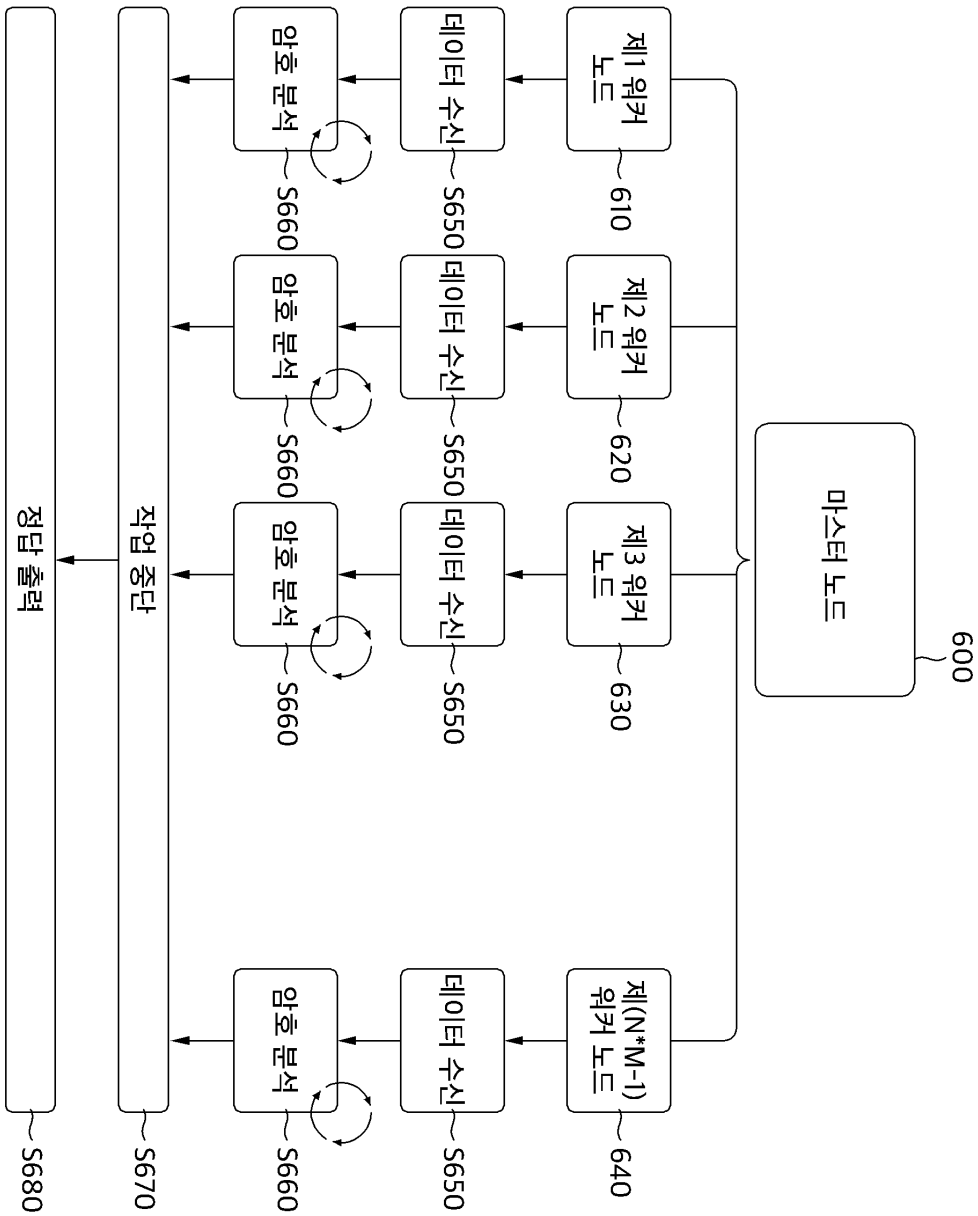
도면6a



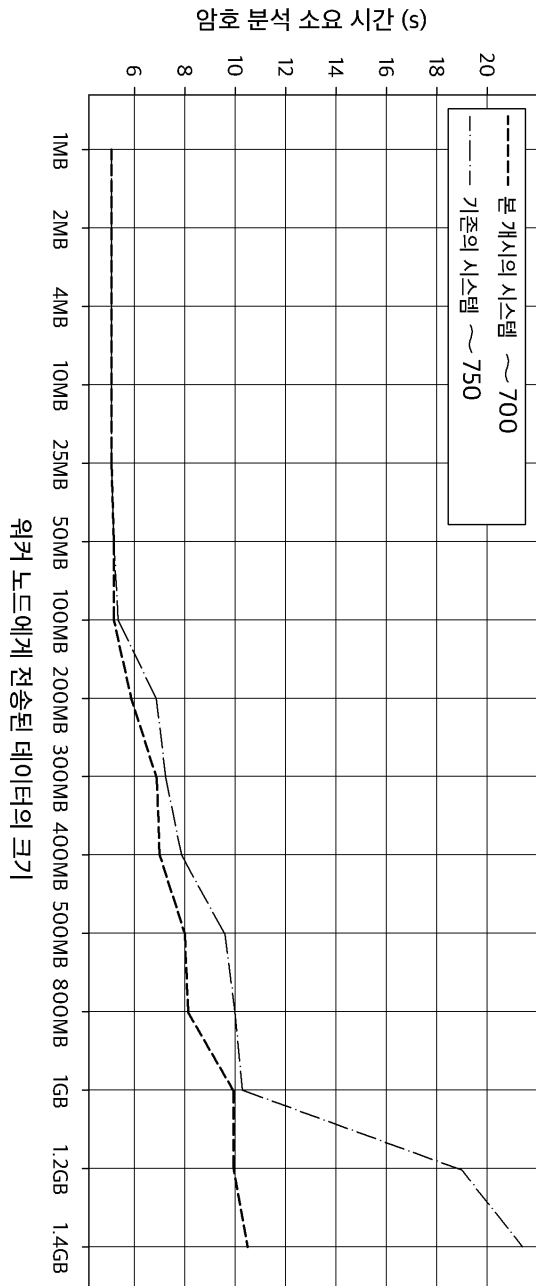
도면6b



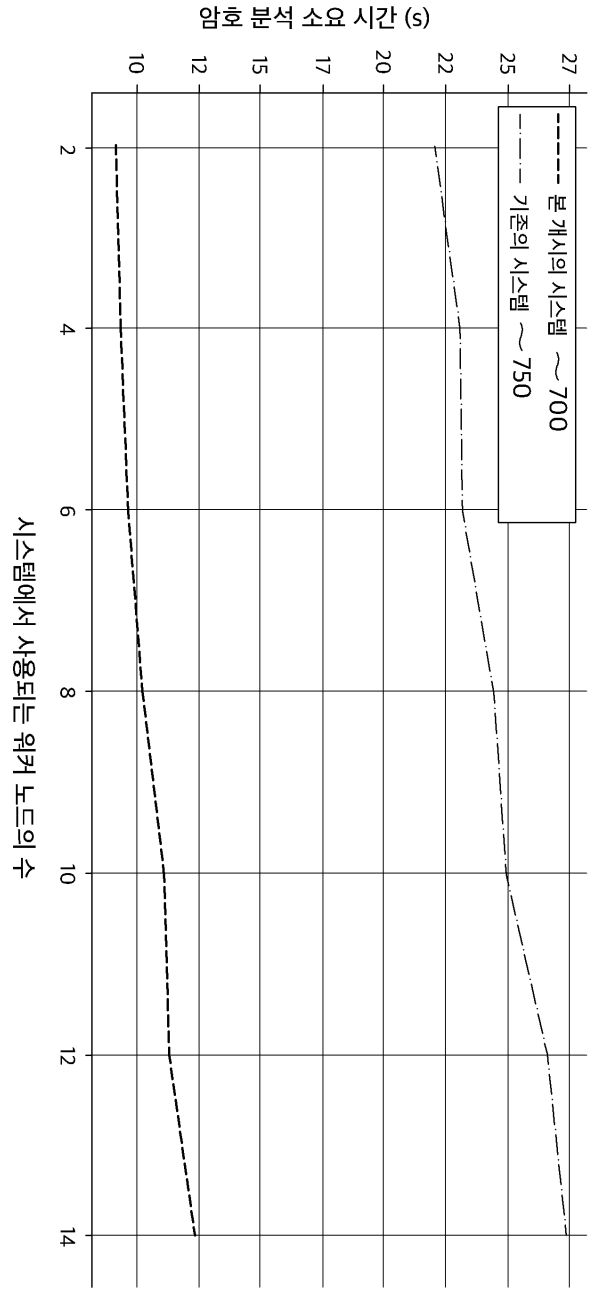
도면6c



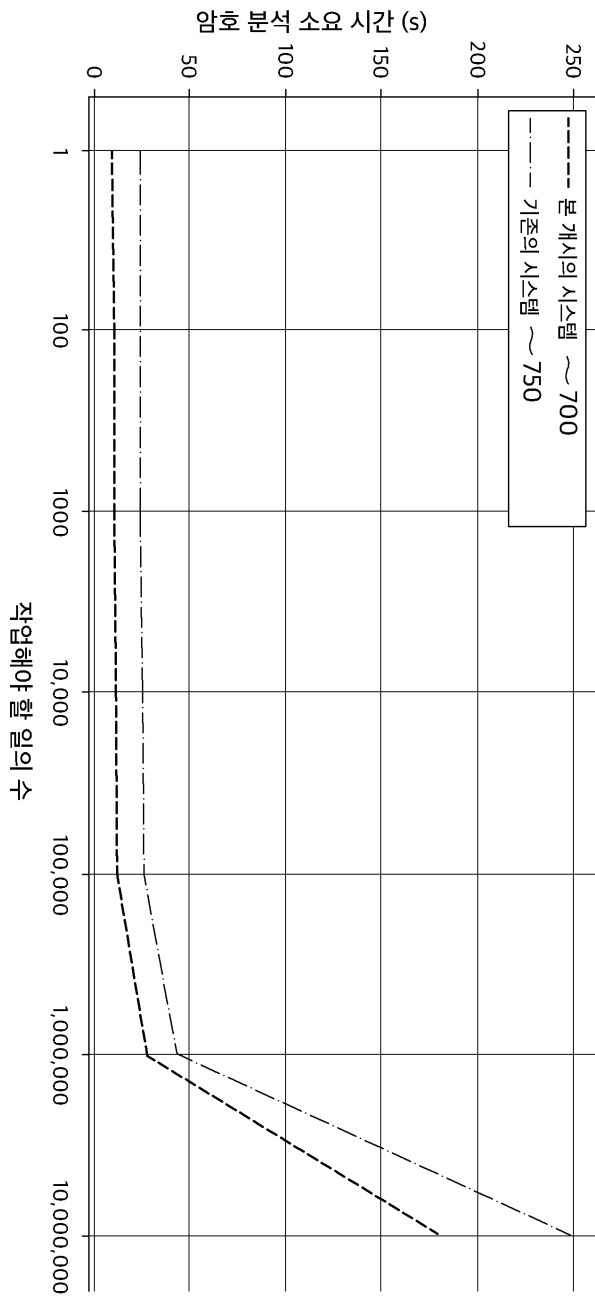
도면7a



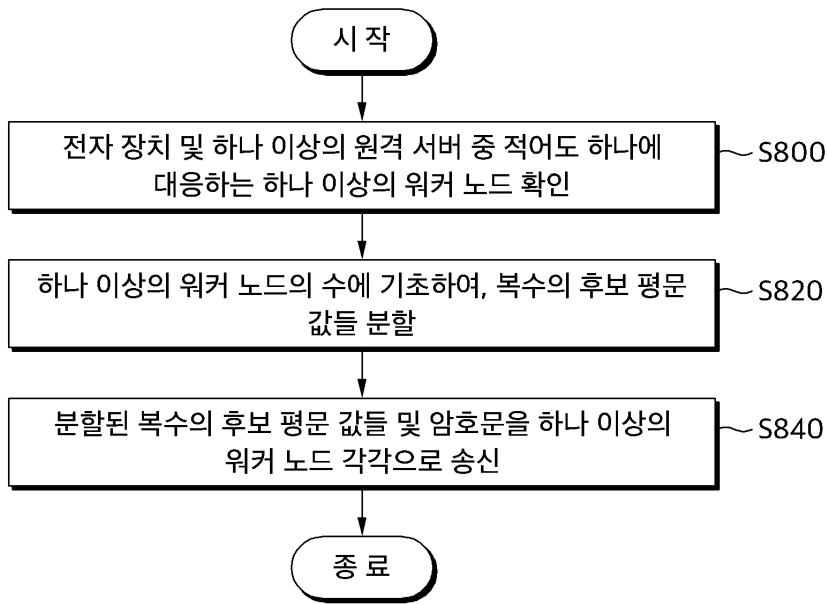
도면7b



도면7c



도면8



도면9

