Journal of Knowledge Information Technology and Systems

ISSN 1975-7700

http://www.kkits.or.kr

# A Study of GPU Based Computing Methods on Military Defense Systems

**Jung-Woong Lee[*1], Ye-Eun Hong[1], Won-Jae Jang[2], Sang-Jun Kim[3], Youn-Jeong Lee[3], Seung-Bae Jee[3] Ji-Hyeon Park[3,]**

[1]*Department of National Defense Business, WeUNUS. Co.*
[2]*Department of National Defense Business, Ssangyong Information & Communications Corp.*
[3]*2nd R&D Institute, Agency of Defense Development*

## ABSTRACT

By the mid of 2020, Korean military forces have planned to built the network centric warfare(NCW) environment by integrating advanced US/NATO standardized tactical data links and Korean unique data links. In this NCW environment, sensors, reconnaissance and weapon systems will be integrated with command and control systems utilizing tactical data links These inter-systems integrations bring out the dramatic growth of exchanging data and computing resources for real time processing. However, these issues could not be easily resolved with current CPU based computing architectures. Because, CPU is not dedicated parallel processing device, it is usually very expensive and multi-thread programming using CPU is very difficult even for expert programmers. By the way, the high-speed arithmetic dedicated GPUs(Graphic Processing Unit) with hundreds or thousands cores are being developed rapidly and applied commercially in the area of genetic analysis, hydrodynamics and cryptography in these days. It also provides grid-block-thread architectures for parallel data processing and software development kits(SDK) for developers to exchange data and functions between hosts systems and GPUs at ease. In a huge military system, there are lots of high computing power required areas such as receiving and transmitting tactical messages with specific rules, data fusions and management, threat evaluation, weapon assignment and drawing situational display. This study will identify those areas in detail and suggest GPU based parallel processing ideas which can be adaptable with several examples and experiments.

© 2017 KKITS All rights reserved

**KEYWORDS:** GPU vs CPU, Parallel processing, Tactical data links, Command and control systems, ROC

**ARTICLE INFO:** Received 24 August 2017, Revised 2 October 2017, Accepted 13 October 2017.

[*]Corresponding author is with the Department of National Defense Business, WeUNUS Co., Ltd., 220 Joongdae-ro, Songpa-gu, Seoul, 05818, KOREA.
*E-mail address*: jwlee@weunus.com

www.kci.go.kr

## 1. 서 론

현대전은 단일 무기체계의 성능에 의존하는 플랫폼 중심전(Platform Centric Warfare)에서 네트워크 중심전(NCW: Network Centric Warfare)으로 패러다임이 전환되고 있다. 이에 따라 감시체계(Sensor), 지휘통제체계(Command and Control), 정찰 및 타격체계(Reconnaissance and Shooter) 등 전장 환경의 모든 요소들이 전술데이터링크(TDL: Tactical Data Link)라고 하는 통신 네트워크를 통해 자기위치 및 식별정보, 감시정보, 전자전정보, 지휘통제 정보 및 이미지와 문자열과 같은 지원정보들을 근실시간으로 교환하게 함으로써 네트워크에 참여하는 모든 아군 부대들이 동일한 상황인식, 위협평가 및 무장할당, 지휘결심, 교전통제를 수행할 수 있도록 하여 작전 수행의 효율성을 극대화하고 있다.

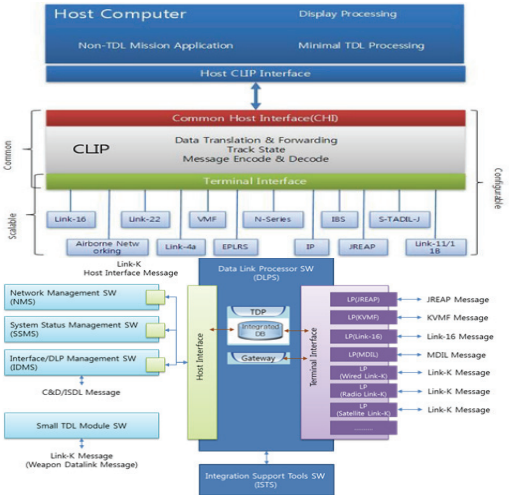



그림 1. CLIP과 한국형 다중 전술데이터링크 처리기 아키텍처
Figure 1. Architecture of CLIP and Korean Multi-Tactical Data Link Processor

<그림 1>의 상단에서 보듯이 미국은 1990년대부터 CLIP(Common Link Integration Processing)이라는 전술데이터링크 통합 모델을 제시하여 2010년까지 4단계로 나누어 점진적 개발을 이루었으며 탑재 하드웨어 환경의 영향을 최소화하고 전술 데이터링크 메시지 구현을 일반화함으로써 무기체계/지휘통제체계의 전술 데이터링크 통합의 문제를 해결하고 상호운용성을 극대화하였다.[2][13] <그림 1>의 하단은 한국형 합동전술데이터링크 사업의 일환으로 수행 중인 선도형 핵심기술 개발 사업의 다중 TDL 통합처리기술의 아키텍처이다. 2000년대 초반부터 한국형 전술데이터링크인 Link-K와 KVMF(Korean Variable Message Format)의 개발에서 시작하여 2017년 현재, JREAP(Joint Range Extension Application Protocol), Link-16, MDIL(M-SAM: Ditigial Information Link), ISDL(Inter-Site Data Link) 및 호스트 인터페이스를 통합하여 육.해.공군 무기체계에 공통으로 탑재되어 운용될 수 있는 수준으로 진화하였다.[15][17]

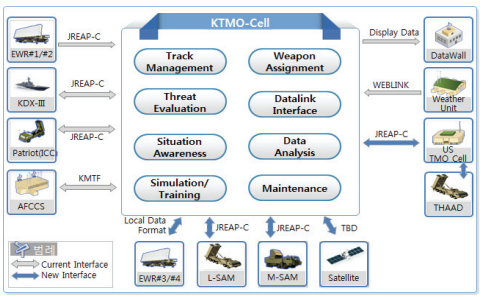



그림 2. 지휘 통제체계 일반 아키텍처
Figure 2. Architecture of Command and Control System

<그림 2>는 지휘통제체계의 하나인 탄도탄 통제소의 일반적인 아키텍처이다. 각종 센서(조기경보레이더(EWR: Early Warning Radar), 이지스함 레이더(KDX-III), 위성 등)들로부터 연동통제문서(ICD: Interface Control Document)와 전술데이터링크 표준에 따라 트랙정보를 수신 관리하고, 전장관리체



www.kci.go.kr

계(공군C4I체계, 전구합동화력운용체계(JFOS-K, Joint Fire Operating System Korea), 군사정보통합관리체계(MIMS: Military Information Management System))로부터 지휘통제에 필요한 적/아 무기체계 제원정보 등을 획득한다. 지휘통제체계는 수집된 정보들을 통합하여 적 표적에 대한 위협평가, 최적의 무장할당과 같은 지휘결심 절차를 수행하며 그 결과는 요격체계(KDX-III, 장거리 지대공 미사일(L-SAM: Long Range Surface to Air Missile), 중거리 지대공 미사일(M-SAM: Mid Range Surface to Air Missile) 등)로 전파되어 교전통제 및 교전상태 관리가 수행된다. 이러한 일련의 전술, 작전정보들은 실시간으로 다양한 형태의 지도 위에 전시되고, 대용량 저장장치에 저장되어 사후 분석 및 교육훈련용으로 활용되기도 한다.

이와 같이 다양한 전술데이터링크의 통합이 이루어지고 통합된 다중 전술데이터링크 네트워크 환경 위에서 운용되는 지휘통제체계는 점차 증가하는 대용량 트랙 처리, 대용량 송수신 필터, 실시간 자료 분배 및 이중화 등 최대 부하 상태에서 CPU 및 메모리 여유용량 확보와 같은 기본 작전요구성능(ROC: Required Operational Capability)을 만족시키기 위해 고용량 서버 구축, CPU 및 메모리 증설 등으로 대응하고 있다. 그러나, CPU를 포함한 하드웨어 발전이 요구성능의 증가에 못 미치고, 높은 서버가격으로 개발 비용 및 유지보수 비용을 증가시키는 이슈가 제기되고 있다. 이는 과거 수년간 체계사업 또는 성능개량 사업을 통해 구축된 체계사업의 기술시험 및 운용시험평가단계에서 개발자 및 운용자들을 괴롭혀온 사실이기도 하다. 또한, IP Over TDL(전술데이터링크로 IP 데이터 교환기술), Link-22, 합동교전능력(CEC: Cooperative Engagement Capability), WNW(Wideband Network Waveform), Cursor On Target과 같이 미래의 고속/대용량 데이터링크가 추가적으로 도입되어 통합될 경우 성능의 문제점은 보다 심화될 것이며 특히, 항공기나 소형 함정과 같이 대형 서버의 설치가 어려운 플랫폼의 경우 현재의 CPU중심의 컴퓨팅 환경만으로는 근실시간 전술자료 처리가 불가능해질 것이다.

본 논문은 이러한 문제점을 해결하기 위해 전술데이터링크와 지휘통제체계에서 CPU와 메모리 자원을 많이 사용하는 영역을 식별하여 이를 GPU 기반의 컴퓨팅으로 전환할 수 있는 방법을 찾아 제시함으로써 CPU와 GPU를 효과적으로 연계하여 활용할 수 있는 방안을 마련하고자 한다.

## 2. 병렬처리에 대한 이해

### 2.1 CPU vs GPU

GPU를 이용한 병렬처리 기법은 이미 오래전부터 그래픽 처리를 위한 알고리즘에 적용되어 Vertex 계산 및 Rendering을 수행하고 가속화되어 일반 PC나 태블릿에서도 고해상도의 동영상과 이미지를 볼 수 있게 되었다. 이러한 아키텍처는 그래픽카드가 아닌 병렬처리 전용 GPU 로 진화하여 수십억 개의 DNA 염기 서열로 이루어진 인간 게놈에서 의미 있는 유전자를 검색 추출하거나, 이미 알려진 유전자의 다형성(polymorphism)을 초고속 분석하는 등 과학 및 산업 전반에 적용되고 있다. 이를 가능하게 하는 요인은 크게 2가지인데 첫째는 다수의 점(Vertex)와 비트로 구성된 이미지를 동시에 처리하기 위해 GPU는 태생부터 수백~수천 개의 코어로 구성된 멀티코어 아키텍처를 가졌다는 것이다. 둘째는 GPU의 처리 속도의 발전 속도가 CPU보다 월등히 높아서 메인보드에 종속적인 CPU보다는 주변장치로써 언제든지 교체하고 업그레이드할 수 있는 GPU를 사용하여 계산을 맡기는 것이 비용대비 효과에 있어서 우수하다는 점이다.



www.kci.go.kr

<그림 3>은 many-core와 독립적인 메모리를 갖는 GPU의 일반적인 물리적인 구조를 보여주며 <그림 4>는 2015년 기준 GPU와 CPU의 연산속도의 차이를 보여준다.[4][5]

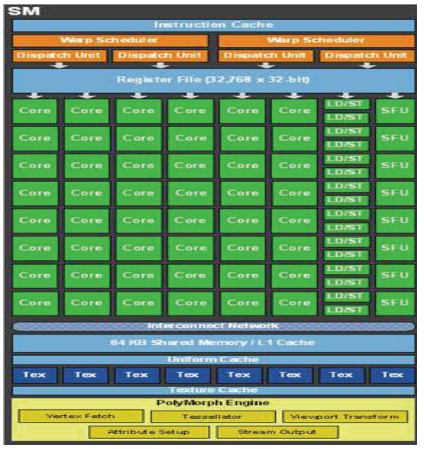



그림 3. 일반적인 GPU 코어 및 메모리 구조
Figure. 3. General Structure of GPU Cores and Memory

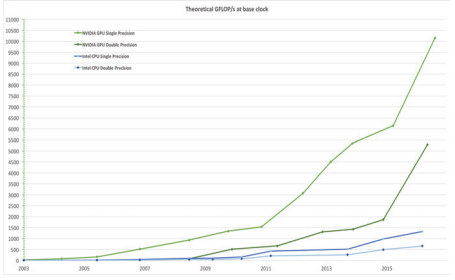



그림 4. CPU와 GPU간의 연산속도 차이 비교
Figure 4. Comparison of GFLOPs between CPUs and GPUs

## 2.2 병렬처리의 적용가능 분야 일반

GPU의 병렬처리 구조는 모든 데이터에 적용 가능한 것은 아니다. GPU적용의 선제 조건은 다음과 같다. 첫째는, 병렬처리 대상 데이터가 상호간에 종속적이지 않아야 한다는 점이다. 즉, 각각의 데이터는 상호 독립적이며 멀티코어에 로딩되어 있는 병렬처리 알고리즘 쓰레드에 개별적으로 전달될 수 있어야 한다는 것이다. <그림 5>는 MPEG-2 인코딩 알고리즘의 개략 다이어그램인데 점선 부분의 내용을 보면 이전 Frame의 이미지 정보가 현재 Frame에 대한 계산식에 영향을 준다. 이럴 경우 Frame별 병렬처리가 불가능하기 때문에 GPU를 이용하여 병렬처리 하는 것은 효율적이지 못하다.[6]

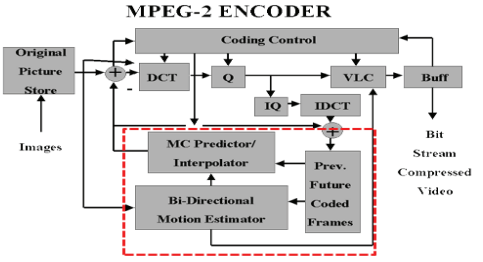



그림 5. MPEG-2 인코더 블록 다이어그램
Figure 5. MPEG-2 Encoder Block Diagram

비슷한 사례로 압축 알고리즘이 있다. <그림 6>은 LZ77 History Based 압축방식의 설명인데 동일한 패턴이 탐지되면 최초의 패턴과의 거리(Dist) 및 패턴 자체의 길이(Len) 정보로 치환하여 압축하는 방식이다. 이는 패턴 간 종속성이 발생하여 병렬처리 대상으로는 부적절하다.

그림 6. LZ77 History 기반 압축 개요
Figure 6. LZ77 History based compression

반면, 블록 암호화(block cipher)는 GPU 병렬처리의 가장 적합한 예이다. 블록 암호화란 하나의



www.kci.go.kr

키 하에서 블록 암호를 반복적으로 안전하게 이용하게 하는 절차를 말한다. 블록 암호는 특정한 길이의 블록 단위로 동작하기 때문에, 가변 길이 데이터를 암호화하기 위해서는 먼저 이들을 단위 블록들로 나누어야 하며, 나뉘어진 블록들을 동일한 알고리즘에 적용하여 암호화 한 후 그를 다시 합쳐서 최종 결과물을 만들어 낸다.

<그림 7>은 전자코드북(Electronic Code Book)이라고 하는 블록 암호화에서 가장 간단한 알고리즘이다. 최초의 평문은 일정크기의 평문 블록으로 분할되고 동일한 마스터 키와 암호화 알고리즘을 거쳐 각각의 암호화 블록으로 변환된다.[3][7]

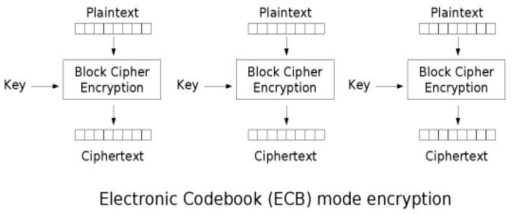



그림 7. 전자코드북 암호화 다이어그램
Figure 7. ECB Mode Encryption Diagram

현재 대한민국의 국방부를 포함한 전자정부의 각 체계에는 국산 암호화 알고리즘인 ARIA를 사용하고 있다. ARIA의 특징은 다음과 같다. [8]

- 블록크기: 128bit,
- 키크기: 128/192/256bit
- 전체구조: Involutiona Substitution Permutation Network
- 라운드 수: 12/14/16(키 크기에 따라 결정됨)

<그림 8>은 ARIA 암호화 알고리즘의 블록 다이어그램이다. 128bit로 나뉘어진 평문은 12~16번의 라운드를 통해 치환계층(Substitution Layer)와 확산계층(Diffusion Layer) 알고리즘을 통해 암호화된다. 만약, 1Mbyte 크기의 평문 파일이 있다고 가정하면 이는 총 625,000개의 128비트 평문 블록으로 분리된다. 256bit의 키를 사용하는 경우 16번 라운드를 통해 암호화 되므로 전체 파일이 암호화 되는데는 10,000,000의 라운드 기능을 수행하게 된다.

만약, GPU를 16개의 블록과 각 블록별 16개의 쓰레드를 구성하여 병렬처리를 하는 경우 1회에 최대 256개(16X16)의 평문 블록이 암호화되므로 순차적 처리보다 이론적으로는 256배 속도 개선을 기대할 수 있다. 그러나 호스트와 GPU 장치간 메모리 할당, 회수, 동기화, 장치 간 전송지연 시간을 고려할 경우 그 속도는 다소간 줄어들 것이다.

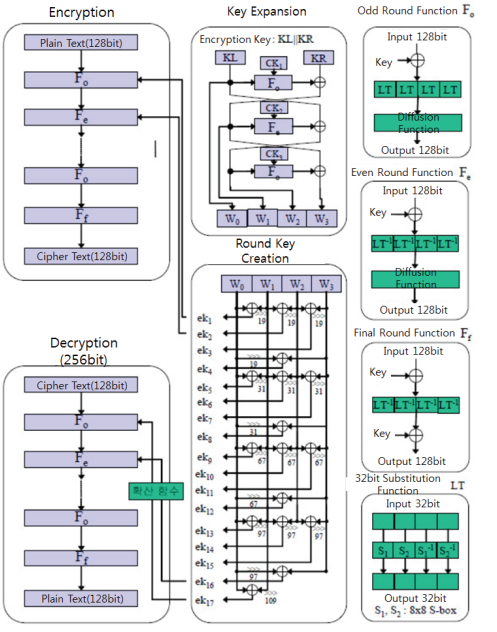



그림 8. ARIA 암호화 다이어그램
Figure 8. ARIA Encryption Diagram

이러한 병렬처리 암호화 기법은 체계간 송수신되는 메시지 또는 파일, 스토리지 등에 저장되는 대용량 파일 등에 적용이 가능한데 특히, 국방정보통합관리체계(MIMS)와 각군 및 합동C4I체계 사이의 이미지 교환, UAV와 지상통제소간 이미지 교환 등에 효과적으로 적용될 수 있다.



www.kci.go.kr

## 3. 전술데이터링크 영역에서의 병렬처리

### 3.1 전술데이터링크 메시지의 이해

전술데이터링크 메시지 프로토콜의 스키마는 크게 메시지구조, 필드 변환규칙, 송수신 규칙 세부분으로 구성된다. 호스트체계에서 관리되는 정보는 개발자 또는 운용자가 이해하기 쉬운 구조체 또는 데이터베이스에 정수형, 부동소수점, 문자열과 같이 개발언어 표준 타입(Type)의 필드로 구성되어지는데 이러한 정보가 전술데이터링크로 송수신되기 위해서는 아래에서 기술하는 세 가지 규칙을 준수하여 변환되고 검증되어야 한다.

첫째는 메시지 구조를 준수해야 한다. 전술데이터링크의 일반적인 메시지 구조는 <그림 9>와 같다. 하나의 메시지는 지정된 크기의 다수 개 워드로 구성되고 각 워드에는 정해진 필드들로 구성된다. 예를 들어, Link-16의 경우 공중트랙(Air Track) 메시지는 Initial Word, Extension Word, Continuous Word 1~5등 총 7개의 워드로 구성되며 각 워드에는 공중항적의 위치(위도/경도), 고도, 속도 등의 운동성 필드, 피아식별 필드, 각종 상태필드들이 표준에 따라 워드별로 분배되어 있다. [9][10]

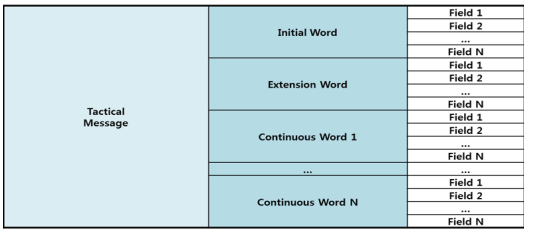



그림 9. Link-16과 Link-K 메시지의 일반적인 형식
Figure 9. General Message Format of Link-16 and Link-K

둘째는 메시지 송수신 규칙을 따라야 한다. 이렇게 구성되는 전술데이터링크 메시지의 모든 필드와 워드들이 항상 교환되는 것은 아니다. 전송하는 주체, 메시지의 타입, 타 정보와의 종속성 등을 고려하여 불필요한 정보교환을 최소화하기 위해 전송될 수 있는 워드와 그 워드내의 필드 값의 범위를 지정하고 있다. 이를 전송규칙(Transmit Rule)이라고 하며, 전술데이터링크 메시지를 구성할 때 반드시 따라야 하며 미 준수시 수신 측에서 해당 메시지는 거부되어 처리되지 않으며 상호운용성을 보장할 수 없다. 예를 들어 지휘통제노드(C2: Command and Control)가 공중항적을 최초 전송할 때에는 Initial Word, Extension Word 및 Continuous Word 1, 4, 5만이 전송되어야 한다.

메시지 수신 시에도 규칙이 존재하며 수신된 메시지의 사용방법(Message Use), 수신 노드가 지휘통제노드(C2: Command and Control)와 비(非)지휘통제노드(Non-C2)인지 여부 등에 따라 처리 및 전시 여부가 결정된다. 예를 들어 공중트랙이 수신된 경우 항적품질이 0인 비실시간 트랙의 경우 Non-C2 노드에서는 처리하지 않는다.

마지막으로는 필드 변환 규칙을 준수해야 한다. 전술메시지를 구성하는 필드들은 개발자 또는 시스템이 사용하는 일반적인 타입을 따르지 않는다. 예를 들어 공중트랙의 위도와 경도를 시스템에서 구현할 때 십진수 도(Decimal Degree)인 37.1234, 127.3456을 4byte의 float타입을 이용하여 관리하지만 전술데이터링크 상에서는 각각 21bit, 22bit의 2의 보수값(two's complementary)로 변환된다. 이러한 데이터의 경우 해상도(resolution)이라는 것이 정의되는데 위도의 경우 1bit는 90도/ "21bit가 표현할 수 있는 최대값" 의 해상도를 갖는다.

즉, "데이터링크 전송 값=시스템 자료 값 / 해상도" 가 되고 수신 시에는 그 역함수가 성립된다. 이러한 치환 대상은 위/경도 좌표, 고도 및 속도와 같은 운동성 정보 필드들이 해당된다.

반면, 모든 변환이 치환식으로 이루어지는 것은 아니다. 트랙번호가 대표적인데 공중트랙의 경우



www.kci.go.kr

19bit로 이루어진 필드로 전술데이터링크상에 교환되며 19bit는 각각 <그림 10>과 같은 구성을 가진다.[16]

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Expression Values | A-Z, 0~7 | | | | | A-Z, 0~7 | | | | | 0~7 | | | 0~7 | | | 0~7 | | |

그림 10. Link-16과 Link-K 트랙번호 자료구조(bit단위)
Figure 10. Track Number field's bit structure of Link-16 and Link-K

앞의 5bit 2개는 A~Z와 0~7까지의 Alphanumeric을 표현하도록 하였으며 3bit 3개는 0~7까지의 8진수를 표현하도록 되어 있다. 이러한 필드는 호스트 내부적으로는 정수값 또는 전시를 위한 스트링 값으로 변환되어진다. 이 때에는 비트단위의 분할과 조립이 필요하다. 트랙번호와 IFF/SIF Mode Code들이 여기에 해당된다.

## 3.2 전술데이터링크 영역에서의 병렬처리 적용 가능 사례

### 3.2.1 대량 메시지 인코딩/디코딩

Link-16과 Link-K 전술데이터링크의 경우 Data Field Identifier(DFI)와 Data Unit Identifier(DUI)로 구성되는 필드들의 종류는 10000여개에 달한다. 또한 각 필드들은 Data Item이라는 수많은 코드(Code)값, 해상도 값, 범위 값 등으로 속성으로 구성된다. 이러한 구성이 전체적인 Tactical Database(TDB) 스키마를 구성하게 되는데 메시지 송수신 시 각 필드의 값에 대해 TDB를 검색하고, 그 결과를 얻어진 필드의 속성을 이용하여 인코딩 및 디코딩을 수행하게 된다.

공중트랙(Air Track) 하나의 메시지는 최대 7개의 워드, 60여개의 필드로 구성된다. 만약 최대 10000개의 트랙을 동시에 생성할 경우 TDB를 60*10000번 조회하게 되고, 각 필드별 변환공식에 의해 인코딩/디코딩되기 때문에 실제로 내부 Context Change 및 연산식은 수억 번까지 발생하게 된다.

```
void CreateAutoMessage(int total)
{
    // 1 ~ 4 과정을 total 만큼 수행

    1. 레코드 명을 이용해 TDBRecord 생성

    2. TN Manager를 이용해 사용 가능한 TN 할당 받아 필드 값 설정

    3. 전송 룰에 맞게 모든 필드 값 설정
        1) 필드명을 이용해 필드 찾기
        2) 찾은 필드에 값 설정

    4. 메시지 목록에 레코드 추가
}
```

그림 11. 대량 메시지의 순차적 생성 절차
Figure 11. Sequential Generation Processing of Massive Messages

<그림 11>은 대량 메시지를 순차적으로 처리할 때의 절차이다. 입력받은 개수만큼 1~4번의 과정을 for나 while 문을 이용하여 반복하게 되며 3번에서 TDB 조회 및 필드의 변환이 수행되어 최대 부하가 걸리게 될 것이다. 이러한 변환은 단순히 인코딩/디코딩에만 국한되는 것이 아니라 MIL-STD-6020 및 한국군 이종 데이터링크 중계표준에 정의된 데이터링크간 메시지 변환에도 적용될 수 있다.[11][12]

```
void CreateAutoMessage(int total)
{
    1. map<int /*index*/, TDBRecord> records;

    2. total 만큼 records 채우기

    3. SetFields<<<blocksPerGrid, threadsPerBlock>>>(records);

    // 4 ~ 5 과정을 total 만큼 수행 (records[index])
    4. TN Manager를 이용해 사용 가능한 TN 할당 받아 필드 값 설정

    5. 메시지 목록에 레코드 추가
}

__global__ void SetFields(map<int, TDBRecord> records)
{
    1. index 값 및 생성할 레코드 구하기
        1) int index = blockDim.x * blockIdx.x + threadIdx.x;
        2) TDBRecord* record = &records[index];

    2. 레코드를 전송 룰에 맞게 모든 필드 값 설정
        1) 필드명을 이용해 필드 찾기
        2) 찾은 필드에 값 설정
        * Track Number 는 겹칠 수 있기 때문에 제외
}
```

그림 12. 대량 메시지의 병렬 생성 절차 및 함수
Figure 12. Parallel Generation Processing of Massive Messages and Kernel Function

<그림 12>는 대량 메시지를 병렬로 처리할 때의 절차와 병렬처리용 커널함수이다. 3번에서 보듯이



www.kci.go.kr

커널함수를 GPU의 특성에 맞도록 그리드/블록 및 쓰레드 인덱스로 병렬화하여 GPU에 전달함으로써 더 이상 for나 while문의 사용 없이 동시에 대량 메시지에 대한 필드 변환을 동시에 처리할 수 있도록 한다.

### 3.2.2 대량 항적 전시

```
void DisplayObject()
{
    // 1 ~ 5 과정을 모든 전시 객체에 대해 수행
    1. 전시 필터링 여부 확인 (Environment, Identity 등)
    2. 위치 계산
    3. 심볼 label 설정
    4. 심볼 코드(파일명) 설정
    5. 심볼 히스토리, 심볼 이미지, 속도 선 그리기
}
```

그림 13. 대량 항적의 순차적 전시 절차
Figure 13. Sequential Display Processing of Massive Tracks

<그림 13>은 대부분의 시스템에서 가장 부하를 많이 주는 전시 영역에서의 순차적 처리 방법이다. 하나의 항적이 수신되면 전시 필터(영역, Environment, Identity 등)에 의해 전시 유무를 판단하게 되는데 특히 다각형, 원, 사각형 등의 영역 필터 계산 때 많은 컴퓨팅 자원과 시간이 소요된다. 이 뿐만 아니라, 수신시간과 현재의 시간을 고려한 위치보정 계산, 메시지로부터 심볼과 레이블 정보 추출 및 이미지 생성 등을 거쳐 최종적으로는 항적의 심볼, 레이블, 속도 선(Velocity Vector), 항적의 궤적(History)을 전시하게 되며 이 과정은 수신된 항적의 수만큼 반복된다.

```
void DisplayObject()
{
    1. map<int /*index*/, TacticalObject*>  objects;
    2. 전시 객체로 objects 채우기
    3. Display<<<blocksPerGrid,  threadsPerBlock>>>(objects);
}

__global__ void Display(map<int,  TacticalObject*>  objects)
{
    1. index 값 및 전시할 객체 구하기
        1) int index = blockDim.x  * blockIdx.x + threadIdx.x;
        2) TacticalObject*  object  = objects[index];
    2. 전시 필터링 여부 확인 (Environment, Identity 등)
    3. 위치 계산
    4. 심볼 label 설정
    5. 심볼 코드(파일명) 설정
    6. 심볼 히스토리, 심볼 이미지, 속도 선 그리기
}
```

그림 14. 대량 항적의 순차적 전시 절차
Figure 14. Parallel Display Processing of Massive Tracks and Kernel Function

<그림 14>는 순차적 처리 함수를 병렬화하여 GPU로 하여금 전시에 필요한 모든 계산을 항적개수만큼 동시에 처리할 수 있도록 재구성 한 것이다.

참고로, 병렬처리를 위한 데이터의 준비(Preparation) 및 병렬처리 결과에 대한 전시는 여전히 CPU를 이용한 호스트 체계가 수행해야 한다.

### 3.2.3 대량 항적 비교일치

다양한 방식으로 운용되고 있는 전술 데이터 링크를 통해 전술정보를 활용하는 무기체계에서는, 피아의 전술상황 정보를 운용자에게 제공하기 위해, 탐지체계가 보고하는 아군과 적군의 전술 데이터를 항적(track) 단위로 관리하고 있다.

전술 데이터를 관리하는 체계들은 Link-16, Link-11, ISDL, Link-K 등의 표준 규칙 또는 체계에서 정의한 고유 통신 규칙을 준수한다. 이러한 다양한 종류의 전술 데이터 링크 통신이 함께 운용되는 환경에서 전술 데이터를 통합 관리해야 하는 체계는 다수의 출처로부터 이종(異種) 전술 데이터를 보고받아 통합 관리할 필요가 있다.

다종의 전술 데이터 링크 운용환경에서 항적을 관리하는 체계는, 동일한 항적이 다수의 전술 데이터 링크 환경에 의해 탐지되어도 복수의 항적이 아닌 단일 항적으로 인식될 수 있도록 운용되어야 하며, 이를 위해 항적 관리 체계는 항적 비교일치(track correlation) 연산을 수행한다. 항적 비교일치 연산은 각기 보고된 두 항적이 특정 시점에서 유사한 위치, 진로, 속도 및 고도의 속성을 가지는지 분석하여, 탐지된 다수의 전술항적이 실제로 단일 항적인지를 판별하는 기술이다.

기존의 전술 데이터 링크 시스템에서는 이러한 항적 비교일치 연산을 순차적(sequential)인 방식으로 수행하기 때문에, 평균 수억 번의 비교일치 연



www.kci.go.kr

산 과정을 수반하여 시스템 부하를 가중시킬 수 있다.

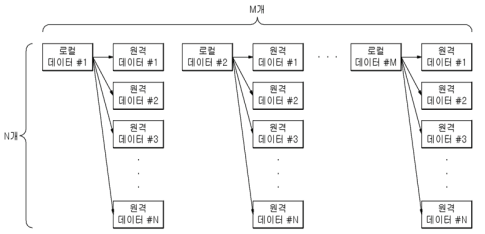



그림 15. 대량 항적의 순차적 비교일치 구조
Figure 15. Sequential Correlation Processing of Massive Tracks

<그림 15>는 로컬(Local) 항적의 수신시 원격(Remote) 항적과의 비교일치를 수행하는 구조이다. 만약, 외부로부터 N개의 원격 항적이 수신되어 관리되고 있고, M개의 로컬 항적이 수신될 경우 각각의 로컬은 N개의 원격 항적과 비교일치 테스트를 수행하여 가장 유사한 항적과 융합되며 이 때 최대 N*M개의 비교일치 테스트를 수행하게 된다.

이러한 각각의 비교일치 테스트는 다시 항적 전처리 및 세부비교일치 연산으로 구성된다. 항적 전처리는 측지오차 보정, 센서오차 보정, 데이터 처리 오차 보정, 원격 노드 위치 오차 보정 등의 계산식이 수행되며 세부비교일치 연산은 비교일치 파라미터 관리, 비교일치 제한조건 관리, 위치 시험, 속도 시험, 방위 시험, 고도 시험 및 때에 따라 IFF/SIF 코드 시험 등으로 구성된다.

이러한 복잡한 과정을 순차적으로 수행될 경우 네트워크 중심 운영 환경에서의 대용량 항적 처리는 필연적으로 시스템 성능 저하를 불러올 가능성이 높다.

<그림 16>은 이러한 문제점을 해결하기 위해 병렬처리 기법을 도입했을 경우의 비교일치 구성도이다. 비교일치를 위한 전처리 및 세부 연산 함수는 GPU에 커널함수로 쓰레드화하여 장입되고 로컬 항적은 그 개수에 따라 정해진 개수만큼 그룹화되어 GPU에 전달되어 동시에 병렬처리를 진행할 수 있도록 함으로써 처리 속도를 향상하고 심지어 병렬처리 간의 중간결과물(Temporary Data)를 GPU의 글로벌 메모리를 활용하여 관리하게 함으로써 호스트 시스템의 CPU 뿐만 아니라 메모리 자원의 절약도 기대해 볼 수 있다.

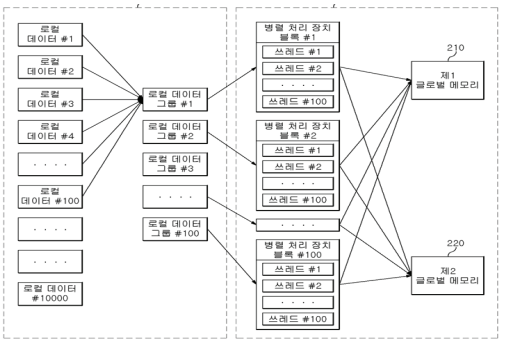



그림 16. 대량 항적의 병렬 비교일치 구조
Figure 16. Parallel Correlation Processing of Massive Tracks

## 4. 지휘통제 영역에서의 병렬처리 적용 가능 사례

지휘통제 분야의 각 기능들은 전술데이터링크와 같이 표준보다는 그 체계를 운용하는 사용군의 요구사항에 따라 개발된다. 그럼에도 지휘통제체계에서 공통적으로 사용되는 기능들이 존재하며 그 중 시스템 자원을 많이 사용하는 기능들을 중심으로 병렬처리 적용 방안에 대해 살펴보도록 한다.

### 4.1. 가시거리 분석(LOS: Line of Sight)

지휘통제체계에서의 가시거리 분석은 공중감시 레이더의 경우 음영지역을 판단하거나, 요격체계의 레이더인 경우 적성(Hostile) 항적의 요격이 가능한지를 판단하는데 중요한 지원요소가 된다. <그림 17>은 200NM의 탐지 거리를 갖는 3개의 레이더의 가시거리분석을 각각 6개의 고도별(3000ft, 5000ft,



www.kci.go.kr

10000ft, 15000ft, 20000ft, 30000ft)로 분석한 결과 이미지이다. 사용된 지형자료는 공개된 GTOPO30 데이터를 활용하였다.

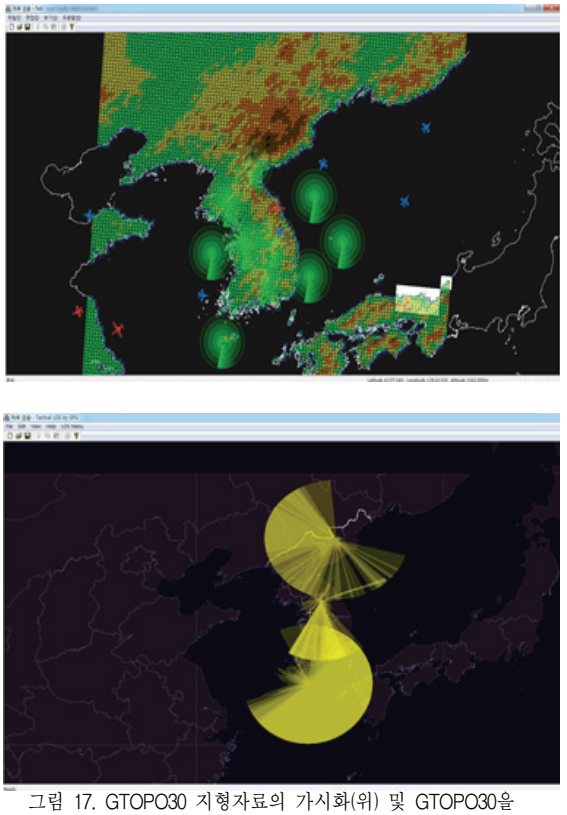


그림 17. GTOPO30 지형자료의 가시화(위) 및 GTOPO30을 이용한 가시거리 분석 결과
Figure 17. Visualization of GTOPO30 Terrain Data and LOS Analysis Results using GTOPO30

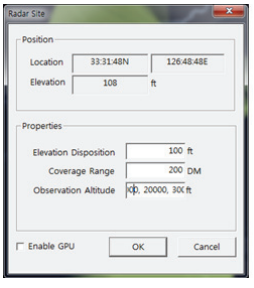
이러한 가시거리 분석은 주로 GTOPO 또는 DEM(수치고도자료: Digital Terrain Map)이라고 하는 지형 자료, 레이더의 해발고도가 입력 값이 되는데 지형 자료의 경우 특정 지역을 동일 간격(수m~수백m)의 그리드로 나누고 각 그리드 영역의 평균 고도 값으로 구성된 데이터를 갖는다.

200NM(370km)의 탐지 반경을 갖는 레이더를 100m 해상도의 DEM을 이용하여 가시거리 분석을 할 경우 6개의 고도레벨에서 82,140,000(3700 x 3700 x 6)번의 탐지 여부 계산이 발생한다. 한반도에는 수십 개의 공중감시, 추적 및 요격 용 레이더가 존재하는데 이에 대한 가시거리 분석을 동시에 수행 할 경우 일반적인 작전 콘솔은 십 수초의 계산이 수행하여 그 동안의 작전자료 실시간 처리에 영향을 줄 수 있게 된다.

이러한 문제점을 극복하기 위해 GPU 병렬처리를 실제 수행해 본 결과 다음과 같이 일반적인 노트북/데스크톱 환경에서도 <그림 18>에서 보듯이 4배 가까운 성능향상을 확인할 수 있었다.

- CPU: Intel(R) Core(TM) i7-3630QM, Dual Core
- GPU: NVIDIA GeForce GT 650M
- Mem: 2Gbyte
- HDD: 128GByte SSD
- GPU 구성: 8block X 256쓰레드

- 사용된 DEM의 그리드 크기: GTOPO30 기준
- 모의 레이더 입력 값
- 위치: 33:31:48N, 126:48:48E(제주도 동쪽)
- 탐지거리: 200NM
- 가시거리 분석 대상 고도: 6개(3000ft, 5000ft, 10000ft, 15000ft, 20000ft, 30000ft)




| GPU Processing Time(sec) | CPU Processing Time(sec) |
|---|---|
| 1.3 | 5.02 |

그림 18. 가시거리분석을 위한 사용자 UI와 GPU 처리 결과
Figure 18. Input UI and GPU Processing Result of Line of Sight



www.kci.go.kr

## 4.2. 기타 적용가능 분야 식별

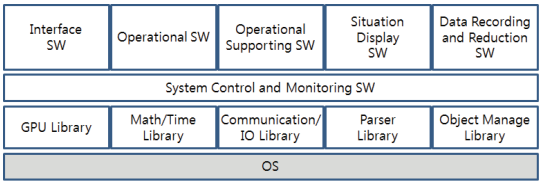



그림 19. 지휘통제체계의 개념적인 SW 아키텍처 다이어그램
Figure 19. Conceptual SW Architecture Diagram of Command and Control System

<그림 19>는 지휘통제체계의 일반적인 SW 아키텍처이다. 센서 및 전술데이터링크와의 자료교환을 위한 연동 SW, 감시, 식별, 위협평가, 무기할당, 교전통제 및 사용자 행위 입력을 처리하는 작전 SW, 가시거리분석(LOS), Terrain Screening Map을 지원하는 작전지원 SW, 지도 및 전술객체 및 전술자료 처리결과 전시, 사용자 행위입력 UI를 제공하는 상황전시 SW, 전술자료의 주기적인 저장 및 저장된 데이터의 추출/분석을 위한 저장 및 분석 SW, 시스템을 구성하는 서버, 콘솔, 프로세스, 네트워크의 상태를 감시하고 통제할 수 있는 시스템 제어 SW 등으로 구성된다.

<표 1>은 이러한 SW 내에서 병렬처리가 적용가능한 영역과 소요되는 알고리즘을 제시하고 있으며, 향후 연구개발 또는 탐색개발을 통해 기존의 알고리즘을 병렬처리용으로 개선하고, 데이터 병렬화 방안이 검토되어야 한다.

표 1. 지휘통제시스템에서의 GPU 적용 가능 영역
Table 1. GPU Applicable Areas in the Command and Control Systems

| Category | GPU Applicable Area | Descriptions |
|---|---|---|
| Interface SW | Tactical Datalink Data Encoding/De coding | ▪ 송수신 메시지를 구성하는 각 Field를 메시지/워드/필드 프로토콜 DB 스키마에 따라 동시에 변환 |
| | Transmit/Receive Filtering | ▪ 송수신 필터가 적용되는 경우 영역 필터에 있어서의 내외부 판단 알고리즘 |
| | Radar Error Correction | ▪ 알려진 레이더의 기계적인 오차(방위, 거리)에 대해 탐지자료에 일괄 적용 |
| | Collimation | ▪ 특정 레이더에 수신된 동일 항적에 대한 감시 자료와 IFF 자료 간의 융합 |
| | Registration | ▪ 동일 항적에 대해 기준 레이더의 자료와 타 레이더 자료의 방위/거리 편차에 대한 보정(자동/수동) 알고리즘 |
| | Mass Tx/Rx Data Block Encryption | ▪ ARIA, SEED 또는 AES 등 블록 암호화 알고리즘 |
| Operational SW | Tracking | ▪ 체계에서 관리되는 항적 자료와 수신된 자료의 연관을 통한 Kalman Filter 적용, 위치예측(Prediction), 내삽법을 이용한 Smoothing 등 항적 추적 알고리즘 |
| | Track Identification | ▪ 자동 항적 식별 구역 설정시 구역 내부 및 외부 판단 알고리즘 |
| | Track Correlation | ▪ 다수 로컬 센서 자료들의 융합에 의한 공통 로컬 자료 생성 및 공통 로컬 자료와 원격항적자료간의 비교일치 |
| | Flight Plan | ▪ ICAO 비행계획서, |



www.kci.go.kr

| | | |
|---|---|---|
| | Correlation | 임무계획서 등 계획된 자료와 탐지 항적간의 비교일치 기능 |
| | Threat Evaluation | ▪ 방어구역, 제한구역, 금지구역 설정시 해당 구역에 진입한 항적의 식별정보, 운동성 정보 등 가중치를 고려한 위협 평가 및 우선순위 결정 알고리즘. |
| | Weapon Assignment/ Trial | ▪ 위협대상 적성기류 항적에 대해 우군 자산들의 대기상태, 무장상태, 연료상태, 운동성 정보를 고려한 최적의 대응 전력 선택 |
| Operational Support SW | Terrain Screening Map(TSM) | ▪ 지형정보를 고려하여 요격체계의 레이더 탐지 거리 분석 및 요격가능 여부 판단 |
| | Radar Screening Map(RCM) | ▪ 감시 레이더의 전파 특성을 고려한 가시거리 분석 및 음영 지역 판단 |
| | Mission Planning | ▪ 작전 경로 설정시 지형정보와 적 자산의 RCM등을 고려하여 적 센서로부터 가장 낮은 포착률이 보장되는 경로 계산 |
| Situation Display SW | Map data Conversion | ▪ 다수의 위/경도 지점으로 구성된 지도 데이터(해안선, 등고선 등)의 전시화면 좌표로의 병렬처리 변환 알고리즘 |
| | DEM Rendering | ▪ 지형자료로로부터 고도 구간별 색상표 병렬 매핑 |
| | Display Filtering | ▪ 전시 필터가 적용되는 경우 영역 필터에 있어서의 내외부 판단 알고리즘 |
| Data Recording/ Reduction SW | Data Reduction Control | ▪ 데이터 분석의 경우 높은 배속 설정시 저장된 데이터의 동시 변환 및 전시용 데이터 생성 |

## 5. 결 론

최근, 전술데이터링크 및 지휘통제체계에는 NCW환경을 반영하기 위해 다중 TDL 통합 기술이 적용되고 있으며, 이로 인해 최대처리 용량에 대한 요구사항이 비약적으로 증가하고 있다. IT 기술의 발전에 따라 통신 속도와 CPU 컴퓨팅 파워 또한 증가하고 있지만 '황의 법칙'이나 '무어의 법칙'이 깨진 것처럼 반도체 집적도가 한계점에 다다르기 때문에 기존 아키텍처로 성능 만족을 위해서는 서버를 다중 병렬화하거나, 고성능/고비용의 장치를 비싼 비용으로 이용해야 한다. 다행히, 최근의 GPU는 일반적인 그래픽처리 뿐만 아니라 NVIDIA의 Tesla제품군과 같이 연산처리 전용 장치와 개발자들을 위한 SDK(CUDA)를 제공하여 호스트 시스템과 GPU 장치 간의 병렬처리를 용이하게 하였으며, 최근에 각광받고 있는 알파고를 비롯한 인공지능, 빅데이터 처리, 신호, 영상, 이미지 처리 등 과학 및 산업의 거의 모든 영역에 진입하고 있다. 국방영역에 이러한 민수용 신기술을 보다 적극적으로 연구, 도입한다면 다음과 같은 3가지 효과를 기대할 수 있을 것이다.

첫째, 작전 운용 측면에서 실시간 처리 보장에 따른 전투 승수 향상이 보장된다. 최근 급증하고 있는 북한의 핵위협은 탄도탄 방어 기술을 요구하고 있으며 이러한 탄도탄 방어 기술은 수 초 내에



www.kci.go.kr

탄도탄의 운동성을 추적하고 관리하여야 하며 큐잉 및 요격명령까지 이루어져야하기 때문에 기존 시스템 보다 훨씬 높은 실시간 처리를 요구하게 된다. 병렬처리를 이용하여 다수개의 탄도탄 추적, 위협평가, 무장할당 알고리즘을 동시에 수행하도록 함으로써 방어 능력 향상이 기대된다.

둘째, 신규 기술 도입에 따른 전문가 양성 및 인력 채용 증대가 예상된다. 데이터의 병렬화, 알고리즘의 GPU 장치 커널화는 일반적인 응용프로그래밍보다 더 높은 기술수준 및 장치에 대한 이해도를 요구한다. 그러므로 단순한 코더(coder)가 아닌 시스템의 성능 요구사항과 데이터 구조에 대한 이해도 높은 개발자의 양성이 가능하다.

마지막으로 GPU의 자원 활용에 따른 CPU, Memory 여유용량 확보 및 시스템 확장성 보장을 보장할 수 있다. 대부분의 국방시스템은 변화하는 환경에 대응하기 위해 30% 이상의 여유용량과 확장성을 요구한다. GPU는 외부장치이기 때문에 CPU보다 확장성이 뛰어나고, CPU보다 높은 처리속도, 자체 메모리를 가지고 있어 여유용량 확보에 유리하다 . 뿐만 아니라, 동일한 컴퓨팅 파워를 달성하기 위해 필요한 공간사용량, 전력사용량 등도 CPU보다 우수해서 무기체계 수명주기 동안 필요한 유지보수 비용 절감에도 기여할 수 있다.

## References


[1] J. S. Kim, S. J. Kim, and M. Y. Lim, *Overview of tactical data link technology*, Journal of The Korean Institute of Information, Scientists and Engineers. Vol. 74, No. 9. pp. 18-28, 2007.

[2] H. Baek, S. Jeong, and J. Lim, *Trends of tactical data link technologies for network centric operations*, Journal of The Korean Institute of Information, Scientists and Engineers. Vol. 28, No. 7. pp. 59-69, 2010.

[3] Y. J. Yeom, and Y. K. Cho, *High-speed implementations of block ciphers on graphics processing units using CUDA library*, Korea Institute of Information Security and Cryptology(KIISC), Vol. 18, No. 3, pp. 23-32, Jun. 2008.

[4] NVIDIA corporation, Whitepaper, *NVIDA's Next Generation CUDA Compute Architecture: Kepler GK110/210, v1.1(2014)*, http://images.nvidia.com/content/pdf/tesla/NVIDIA-Kepler-GK110-GK210-Architecture-Whitepaper.pdf, Jul. 2017

[5] Chris Ueland, *Utilizing GPUs in the Cloud(2014)*, https://www.scalescale.com/utilizing-gpus-cloud, Jul. 2017

[6] Architectures for Video Processing, http://book.huihoo.com/design-of-vlsi-systems/ch13/ArchiMultimed.htm, Jul. 2017

[7] Korea Wikipedia, block cipher modes of operation, https://www.tutorialspoint.com/cryptography/block_cipher_modes_of_operation.htm, Jun. 2017.

[8] Marcof, ARIA(Academy, Research Institute, Agency) Cipher Algorithm, http://marcof.tistory.com/109, Jun. 2017

[9] US Department of Defense(DoD), *Tactical data link(TDL) 16 message standard*, MIL-STD-6016E, Jul. 2012.

[10] US Department of Defense(DoD), *Variable message format(VMF), MIL-STD-6017C*, May 2012.

[11] US Department of Defense(DoD), *Data forwarding between tactical data links(TDL)*, MIL-STD-6020, Sep. 2011.




www.kci.go.kr

[12] H-J. Park, Y-K Park, K-M. Park, H-J. Park, S-W. Kim, J-W. Lee, *Application plan of Korean TDL Forwarding Standards for ROK Military Future Weapon System*, Korean Institute of Information, Scientists and Engineers, pp. 1104-1106, Vol. 2015, No. 12, 2015.

[13] W-P. Kang, J-H. Song, K-H. Lee, D-H. Lee, S-J. Jung, and H-J. Choi, *Analysis of common data link technology trends for the next generation Korean common data link development*, Korea Information and Communications Society, 14-39C-03-01, Vol. 39C, No. 03, Mar. 2014.

[14] E. S. Kim, *Concept of tactical data link employment and next C4ISR system*, Quart. J. Defense Policy Stud., Vol. 74, pp. 49-83, 2007.

[15] Lockheed Martin UK, Integrated Systems and Solutions, Other tactical data link protocols, http://archive.is/fTtvf, Jul. 2017.

[16] Y. J. Lee, S. J. Kim, and M. Y. Lim, *Methodology of interoperating Link-K track number in multi TDLs*, J. KICS, Vol. 38C, No. 12, pp. 1186-1195, 2013.

[17] IDLSoc, Quick Guide: International Data Link Society's guide to: TACTICAL DATA LINKS (TDLs), http://idlsoc.com/Documents/IDLS_tactical_DL_leaflet_WEB1.pdf, Jun. 2017.

## 국방시스템 내에서의 GPU 기반 병렬처리 방법에 대한 연구

**이정웅[1], 홍예은[1], 장원재[2], 김상준[3], 이윤정[3], 지승배[3], 박지현[3]**

[1]*주식회사 위우너스*
[2]*쌍용정보통신 주식회사*
[3]*국방과학연구소*

**요 약**

2020년 중반까지 한국군은 미군 및 나토 표준 선진 전술데이터링크와 한국군 고유의 전술데이터링크들을 통합시켜 네트워크 중심전(NCW: Network Centric Warfare) 환경을 구축할 계획이다. NCW환경에서 감시 및 정찰체계, 요격체계는 전술데이터링크를 통신 인프라로 활용하여 지휘통제체계에 통합된다. 이렇게 시스템 간 통합이 이루어지면 교환되어야 할 자료의 양과 실시간 처리를 위해 필요한 컴퓨터 자원은 극적으로 증가하게 된다. 그러나, 이러한 문제점을 현재의 CPU 기반의 컴퓨팅 아키텍처로는 쉽게 해결하지 못한다. 왜냐하면, CPU는 태생적으로 병렬처리 전용 장치가 아니며 일반적으로 비용이 많이 들고 전문 프로그래머에게도 CPU를 이용한 병렬처리는 매우 어려운 문제이기 때문이다. 다행히, 최근 수백/수천 개의 코어를 갖는 고속 수학연산 전용 GPU가 급격하게 발전하였고, 그리드-블록-쓰레드라고 하는 병렬처리 전용 아키텍처와 개발자를 위한 툴킷을 제공함으로 민수시장에서 유전자분석, 유체역학, 암호학 등에 많이 사용되고 있다. 본 논문은 거대한 국방시스템에서 높은 컴퓨팅 능력을 요구하는 영역을 보다 상세하게 식별하고 GPU의 특징을 적용할 수 있는 방법을 예제와 실험을 통해 제시함으로써 갈수록 높아지는 대용량 고속 처리 요구사항에 대응하고자 한다.

**Jung-Woong Lee** received the bachelor's degree in the Department of Molecular Biology from the Seoul National University in 1994. From 1998 to 2003, he was a software engineer at SICC. He was a senior system engineer at Huneed Technologies from 2003 to 2010. He has been a CEO at WeUNUS Co. Ltd., since 2011. His current research interests include tactical data links, command and control system's architecture, parallel processing with GPU and image processing. He



www.kci.go.kr

is a full-fledged member of the KKITS.
*E-mail address*: jwlee@weunus.com

**Ye-Eun Hong** received the bachelor's degree in the Department of Computer Science from Seojung University in 2013. She has been a junior software engineer since 2015. Her current research interests include tactical data links, tactical situation display and analysis tools. She is a full-fledged member of the KKITS.
*E-mail address*: yenny@weunus.com

**Won-Jae Jang** received the bachelor's degree in the Department of Industrial Engineering from Konkuk University in 1998. He has been a senior system engineer at SsangYong Information & Communication Corp since 2007. His current research interests include tactical data links, system integration method. He is full-fledged member of the KKITS.
*E-mail address*: jwjang@sicc.co.kr

**Sang-Jun Kim** received the bachelor's degree and the M.S. degree in the Department of Electronic Communication Engineering from Hanyang University in 1986 and 1988, respectively. He has been a principal researcher at Agency of Defense Development since 1988. His current research interests include NCW, Tactical Data Link, Link-K. He is a full-fledged member of the KKITS.
*E-mail address*: sangjun@add.re.kr

**Youn-Jeong Lee** received the bachelor's degree (2001), the master degree (2003), the doctor degree(2006) in the school of electronic engineering from the Soongsil University. She has been a researcher at the USC in California US from 2015 July to 2016 June. She has been in the ADD since 2006. Her interesting field are tactical data links, speech signal processing and deep learning. She is a full-fledged member of the KKITS.
*E-mail address*: youn@add.re.kr

**Seung-Bae Ji** received the bachelor's degree in the Department of Electronic Engineering from the Sogang University in 2001. He received the M.S. degree in the Department of Electronic Engineering from KAIST in 2003. From 2003 to 2006, he was a researcher at SAMSUNG SDS. He has been a senior researcher at Agency of Defense Development since 2006. His current research interests include NCW, Tactical Data Link Systems and Protocols, Link-K.
*E-mail address*: seungbae@add.re.kr

**Ji-Hyeon Park** received the bachelor's degree and M.S. degree in the Department of Computer Engineering from Pusan National University in 2003 and 2005. She has been a senior researcher at Agency of Defense Development since 2005. Her current research interests include Tactical Data Link, Link-K, Link-16. She is a full-fledged member of the KKITS.
*E-mail address*: jhpark@add.re.kr



www.kci.go.kr