

SCAM 기상모델의 성능향상을 위한 LAPACK BLAS 라이브러리의 활용

신대영, 조예린, 정성욱*

Performance Improvements of SCAM Climate Model using LAPACK BLAS Library

Dae-Yeong Shin, Ye-Rin Cho, Sung-Wook Chung*

요약 슈퍼 컴퓨팅 기술 및 하드웨어 기술의 발달로 수치 연산 방식 또한 고도화되고 있다. 그에 따라 이전 대비 향상된 기상 예측 또한 가능해진다. 본 논문에서는 SCAM(Single-Columns Atmospheric Model, CISM(Community Earth System Model)을 간소화 한 버전)에 포함되어 있으며 대기 연산을 수행하는 적운 모수화 코드, Unicon(A Unified Convection Scheme)의 성능을 향상하기 위하여 소스 코드 내의 선형대수 수치적 연산 부분에 고밀도 선형대수 연산을 위한 라이브러리인 LAPACK(Linear Algebra PACKage) BLAS(Basic Linear Algebra Subprograms)의 level1 함수를 적용할 것을 제안한다. 이를 분석하기 위하여 SCAM의 전체적인 실행 구조도를 제시하고 해당 실행 환경에서 테스트를 진행하였다. 기존 소스 코드 대비 SCOPY 함수는 0.4053%, DSCAL 함수는 0.7812%, DDOT 함수는 0.0469%의 성능 향상을 이끌어 내었으며 이를 모두 적용한 결과 기존 소스 코드 대비 0.8537%의 성능 향상을 보였다. 이는 본 논문에서 제안한 고밀도 선형대수 연산을 위한 라이브러리인 LAPACK BLAS 적용 방법이 동일한 CPU 환경에서 추가적인 하드웨어의 개입 없이 성능을 향상시킬 수 있음을 의미한다.

Abstract With the development of supercomputing technology and hardware technology, numerical computation methods are also being advanced. Accordingly, improved weather prediction becomes possible. In this paper, we propose to apply the LAPACK(Linear Algebra PACKage) BLAS(Basic Linear Algebra Subprograms) library to the linear algebraic numerical computation part within the source code to improve the performance of the cumulative parametric code, Unicon(A Unified Convection Scheme), which is included in SCAM(Single-Columns Atmospheric Model, simplified version of CISM(Community Earth System Model)) and performs standby operations. In order to analyze this, an overall execution structure diagram of SCAM was presented and a test was conducted in the relevant execution environment. Compared to the existing source code, the SCOPY function achieved 0.4053% performance improvement, the DSCAL function 0.7812%, and the DDOT function 0.0469%, and all of them showed a 0.8537% performance improvement. This means that the LAPACK BLAS application method, a library for high-density linear algebra operations proposed in this paper, can improve performance without additional hardware intervention in the same CPU environment.

Key Words : BLAS, CISM, Global Climate Model, LAPACK, Performance Improvement, SCAM

1. 서론

슈퍼 컴퓨팅 기술의 발전에 따라 각국의 기후 예측

모델 또한 고도화되고 있다. 한국 기상청(KMA, Korea Meteorological Administration) 또한 기후 모델(Climate Mode)을 도입하여 기상 예보를 수행하

This research was funded by the Korea Meteorological Administration Research and Development Program under Grant KMI 2021-01310.

*Corresponding Author : Department of Computer Engineering, Changwon National University (swchung@changwon.ac.kr)
 Received February 02, 2023 Revised February 07, 2023 Accepted February 14, 2023

고 있다. 한국 기상청은 2004년 미국 국립대기 과학 연구소(NCAR, National Center for Atmospheric Research)와 기술 협력을 체결하였고 2014년에는 전 지구 기상모델(GCM, Global Climate Model)인 GloSea5(Global Seasonal Forecast System)를 도입하여 슈퍼컴퓨터 5호기에 이식하였다. 또한 해당 모델을 GloSea6로 버전 업그레이드를 진행한 바 있다.

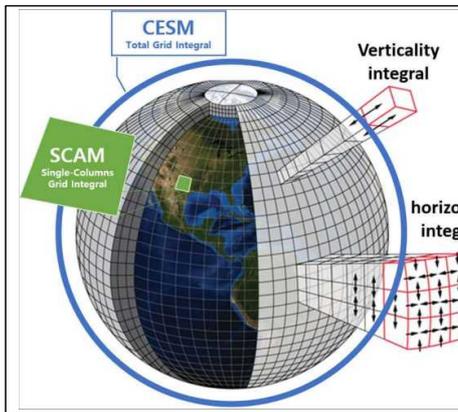


그림 1. CISM, SCAM의 격자 모델링 형태
Fig. 1. Grid modeling of CISM, SCAM

CISM(Community Earth System Model)은 주로 NSF(National Science Foundation)에서 지원하는 대기, 해양, 얼음, 지표면, 탄소 순환 및 기타 구성 요소로 구성된 지구의 과거, 현재 및 미래 기후 상태에 대한 최첨단 컴퓨터 시뮬레이션을 제공하는 지구 기후 모델이다. 이는 그림 1과 같이 지구 모델을 일정한 수평, 수직의 격자 크기로 나눈 후 각 격자에 대한 대기의 상태와 운동에 대한 방정식을 계산하여 얻은 결과를 통해 시뮬레이션 한다[1].

SCAM(Single-Columns Atmospheric Model)[1]은 특정 기상 연구를 위해 CISM을 간소화 한 버전이다. 전 지구가 아닌 단일 격자(Single-Columns)에 대하여 CISM이 수행할 수 있는 4가지 요소 중 오직 대기(Atmospheric) 모델에만 집중하여 모델을 수행한다. SCAM은 2088번의 step (1step=1,200초)을 수행할 수 있으며 이를 Day(일)로 환산할 시 최대 29일까지 모델을 수행할 수 있다[2].

LAPACK(Linear Algebra PACKage)은 선형대수학의 수치적 계산을 수행하는 라이브러리로, 일차 연립방정식, 행렬 분해, 선형 최소제곱법 등을 수행하는 함수를 제공한다. LAPACK은 1992년에 Fortran77로 작성되었으며 2008년에 Fortran90언어로 변경되었다[3]. BLAS(Basic Linear Algebra Subroutines)는 기본 선형 대수 하위 프로그램에서 벡터 덧셈, 스칼라 곱셈, 내적, 선형 조합 및 행렬 곱셈과 같은 일반적인 선형 대수 연산을 수행하는 선형 대수 라이브러리를 위한 표준 저 수준 루틴이다[4]. 그림 2와 같이 LAPACK은 BLAS 라이브러리의 호출을 통해 가능한 한 많은 계산이 수행되도록 작성되었다[5]. 따라서 고성능 컴퓨터에서 LAPACK BLAS를 사용할 경우 성능 효율성 증가를 기대할 수 있다[6].

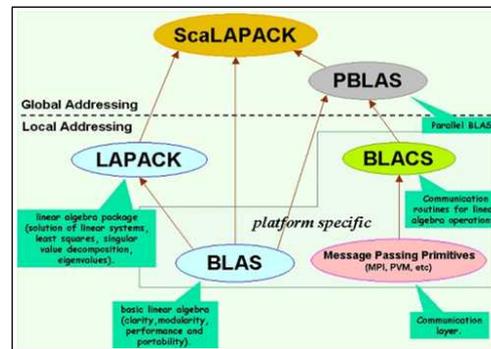


그림 2. LAPACK의 개요
Fig. 2. Overview of LAPACK

SCAM 모델에서 적은 모수화 연산을 담당하는 코드는 Unicon(A Unified Convection Scheme)[7]이며 해당 코드에 LAPACK BLAS 라이브러리를 적용함으로써 Unicon 코드를 포함한 SCAM 모델 전체의 성능 향상을 끌어내어 관련 기술을 축적하고 이를 한국형 기상 모델(KIM)에 적용하여 한국의 계절 예측 시스템의 고도화 방안을 연구하고자 한다.

2. 관련 연구

Merchant, et al는 BLAS 및 LAPACK이 여러 고

성능 컴퓨팅(HPC) 애플리케이션의 기본 구성 요소를 형성하므로 성능 결정과 BLAS/LAPACK 루틴의 성능 튜닝을 위한 FPU(Floating Point Unit) 마이크로 아키텍처의 영향을 조사하였다[8]. Kim, et al.는 행렬의 대규모 그룹에 적용된 BLAS/LAPACK 루틴에서 SIMD 벡터 길이에 따라 행렬을 블록으로 인터리브하는 새로운 소형 데이터 레이아웃을 제안하고 이를 계층적 병렬 애플리케이션 내에서 사용할 수 있는 새로운 인터페이스와 결합하여 아키텍처의 속도를 향상시켰다[9]. Phillips는 공유 메모리 멀티 프로세서를 기반으로 각 처리 요소가 스칼라 특성만 갖는 기본 선형 대수 하위 프로그램(BLAS)의 성능을 고려하여 동일한 환경에서 실행되었을 때 특정 LAPACK 루틴의 성능에 대한 블록 크기의 영향을 조사하였다[10]. Bjørstad는 BLAS 루틴이 컴파일러 제한에 의해 고성능이 억제된다는 것을 조사하였으며 그에 따라 최적의 데이터 이동을 위해 단일 블록에 적용되는 명시적 LU 인수 분해의 필요성을 제기하고 블록 크기의 확장을 통해 LAPACK 루틴의 블록 구조를 수정하지 않고도 고성능에 달성할 수 있음을 보였다[11].

3. SCAM 실행 구조

SCAM 모델을 수행한 방법은 그림 3과 같이 Local PC Client에서 Gate로 SSH 원격 접속을 이용하여 실제로 모델을 수행하는 Cluster Server에 연산을 요청하는 형태로 진행하였다. 해당 방식으로 사용자가 SCAM 모델을 수행시키게 되면 Gate가 USER 계정의 SCAM 소스 코드를 Cluster Server로 전달하게 된다. 전달받은 소스 코드는 Batch System(pbs)에 SCAM Running Job으로 전달되어 Job scheduler(torque)를 거쳐 실행된다. 모델을 실행할 때마다 컴파일은 새로 진행되며 모델 수행이 종료되면 최종 출력값인 nc 확장자 파일 4개와 컴파일 로그 및 러닝 로그가 생성되고 이를 rynch를 이용하여 User가 전달받을 수 있다[12].



그림 3. SCAM 실행 구조도

Fig. 3. Running Process of SCAM System

4. 성능 향상을 위한 LAPACK BLAS 적용 방법

4.1 LAPACK BLAS

LAPACK BLAS는 고밀도 선형대수 연산을 위해 만들어진 고성능 라이브러리이다. 그중 Level1 함수는 벡터 - 벡터의 일반화된 연산 $y = ax + y$ (여기서 x, y 는 벡터, a 는 스칼라를 의미함)을 수행한다. 따라서 본 논문에서는 대기 모델의 연산에 사용되는 방정식 풀이를 LAPACK BLAS의 Level1 함수를 통해 연산하도록 하여 성능을 향상시킬 것을 제시한다.

BLAS function의 명명 규칙은 그림 4와 같이 첫 번째 글자는 Number type을 의미하며 각 함수의 기능에 따라 명명된다[13]. 본 논문에서는 Unicon 소스 코드 연산에 사용되는 변수들의 Data type을 고려하여 SCOPY, DSCAL, DDOT의 함수를 적용하였으며 3가지의 함수가 모두 적용되어 연산을 수행하도록 하였다.

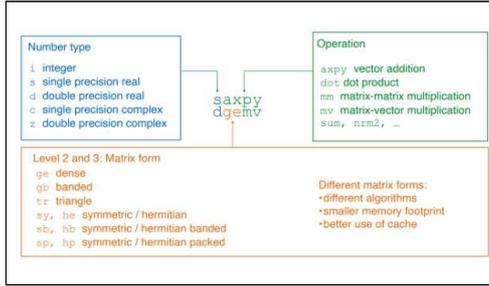


그림 4. BLAS routines 명명 규칙
Fig. 4. BLAS routines: naming conventions

4.2 SCOPY 함수 적용

SCOPY 함수는 벡터 x 를 벡터 y 로 unrolled loops를 사용하여 복사하는 함수이다. SCOPY 함수에 사용되는 각 파라미터가 의미하는 바는 아래 표 1과 같다[14]. Unicon.F90에서 행렬을 복사하는 수식이 loop 안에 존재할 경우 수동으로 loop 문을 분리하고 SCOPY 함수를 call 하여 연산하도록 하였다. 그림 5는 SCOPY가 적용된 Unicon.F90 소스코드의 일부이다.

표 1. Parameters of SCOPY
Table 1. Parameters of SCOPY

name	parameter
n	number of elements in vectors x and y
x	vector x of length n
incx	stride for vector x
y	vector y of length n
incy	stride for vector y

```

Origin Source Code
enddo
  slten_dis(k) = sten_dis(k)
Modified Source Code
enddo
  call scopy(max( ktop_mxen(ixi), ktop_mxen(ixf) ),
            sten_dis, 1, slten_dis, 1)
do k = 1, max( ktop_mxen(ixi), ktop_mxen(ixf) )

```

그림 5. SCOPY 함수 적용
Fig. 5. Apply of SCOPY function

4.3 DSCAL 함수 적용

표 2. Parameters of DSCAL
Table 2. Parameters of DSCAL

name	parameter
n	number of elements in vector x
alpha	scalar α
x	vector x of length n
incx	stride for vector x

DSCAL 함수는 벡터 x 를 스칼라 α 만큼의 곱 연산을 unrolled loops를 사용하여 곱셈값을 연산하고 벡터 x 를 반환하는 함수이다. DSCAL에 사용되는 파라미터가 의미하는 바는 표 2와 같다 [15]. 해당 함수 또한 마찬가지로 unrolled loops를 사용하여 연산하므로 적용하려는 연산식이 loop문 안에 존재할 경우 수동으로 loop문에서 분리한 후 function call하여야 한다. 그림 6은 DSCAL이 적용된 Unicon.F90 소스코드의 일부이다.

```

Origin Source Code
Pau(m) = Pau(m) * ( cmfu_base / rnorm_m )
Modified Source Code
enddo
  if( inorm .eq. 1 ) then
    Pau(m) = Pau(m) * ( au_base / rnorm_a )
  elseif( inorm .eq. 2 ) then
    call dscal(nseg, cmfu_base / rnorm_m, Pau, 1)
  endif
do m = 1, nseg

```

그림 6. DSCAL 함수 적용
Fig. 6. Apply of DSCAL function

4.4 DDOT 함수 적용

DDOT 함수는 벡터 x 및 벡터 y 의 내적, 즉 $x \cdot y = x_1y_1 + x_2y_2 + \dots + x_ny_n$ 를 연산하는 함수이다. DDOT에 사용되는 파라미터가 의미하는 바는 표 3과 같다[16]. 앞선 두 함수의 경우 parameter에 out값 또한 선언되어 있었으나 DDOT 함수는 return할 값의 타입이 parameter에 정해져있지 않아 그림 7과 같이 fortran에서 제공하

는 External이라는 사용자 정의 함수를 사용하여 수동으로 정의한 후 적용하였다. 그림 8은 DDOT 이 적용된 Unicon.F90 소스코드의 일부이다.

```
external ddot
integer, intent(in)
integer, intent(in)

real(r8) alpha
real(r8) ddot
```

그림 7. External DDOT 선언
Fig. 7. External DDOT declaration

표 3. Parameters of DDOT
Table 3. Parameters of DDOT

name	parameter
n	number of elements in vectors x and y
x	vector x of length n
incx	stride for vector x.
y	vector y of length n
incy	is the stride for vector y

```
Origin Source Code
do m = 1, nseg
  tmp1 = tmp1 + Pmu(m)
  cushavg_mxen(iter) = cushavg_mxen(iter) + ztop_msfc(m) * Pmu(m)
enddo

Modified Source Code
do m = 1, nseg
  tmp1 = tmp1 + Pmu(m)
  ! cushavg_mxen(iter) = cushavg_mxen(iter) + ztop_msfc(m) * Pmu(m)
enddo
cushavg_mxen(iter) = ddot(nseg, ztop_msfc, 1, Pmu, 1) + cushavg_mxen(
```

그림 8. DDOT 함수 적용
Fig. 8. Apply of DDOT function

5. 실험 결과 및 성능 평가

성능 분석을 위하여 3장에서 제시한 SCAM 실행 방법에 따라 모델 수행을 30회 실행하고 그에 대한 평균 시간을 기준으로 하였다. SCAM의 step은 2088,

즉 29일로 수행하였다. 또한 1회당 수행시간은 컴파일 시간(compile time)과 모델 수행시간(running time)을 합한 것으로 한다. 함수를 개별적으로 적용한 경우와 모두 적용한 경우의 평균값을 각각 도출하였다. 실험에 수행된 시스템 스펙은 표4와 같다.

표 4. 실험에 수행된 시스템 스펙
Table 4. System specs used for SCAM

Name	Hardware Specification
Node 1	CPU: Intel® Xeon Silver 4208 (8 cores 16 threads) * 2 EA GPU: NVIDIA GeForce GTX 3090 * 2 EA RAM: 128GB(16GB * 8EA) SSD: 960GB OS: CentOS 7
Node 2	CPU: Intel® Xeon Silver 4208 (8 cores 16 threads) * 2EA GPU: NVIDIA GeForce GTX 3090 * 2 EA RAM: 128GB(16GB * 8EA)
Local PC	CPU: Intel® i7-10700K CPU (8 cores 16 threads) GPU: NVIDIA GeForce GTX 3080 12GB RAM: 32GB SSD: 1TBGB OS: Window 10
Gate	CPU: Intel® i7-7700 CPU (4 cores 8 threads) GPU: GeForce GTX 1060 3GB RAM: 8GB SDD: 100GB OS: CentOS 7

5.1 SCOPY 성능평가

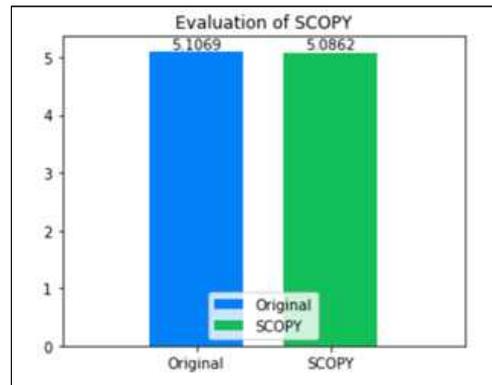


그림 9. SCOPY 성능평가
Fig. 9. Evaluation of SCOPY

적은 모수화 코드의 기존 연산 대신 SCOPY 함수를 적용하여 연산한 결과는 그림 9와 같다. 기존 소스

코드는 평균 5.1069초, SCOPY가 적용된 코드는 5.0862초의 성능을 보이며 이는 기존 대비 0.4053% 성능 향상을 보인다.

5.2 DSCAL 성능평가

적운 모수화 코드의 기존 연산 대신 DSCAL 함수를 적용하여 연산한 결과는 그림 10과 같다. 기존 소스코드는 평균 5.1069초, DSCAL이 적용된 코드는 5.067초의 성능을 보이며 이는 기존 대비 0.7812% 성능 향상을 보인다.

5.3 DDOT 성능평가

적운 모수화 코드의 기존 연산 대신 DDOT함수를 적용하여 연산한 결과는 그림 11과 같다. 기존 소스코드는 평균 5.1069초, DDOT이 적용된 코드는 5.1045초의 성능을 보이며 이는 기존 대비 0.0469% 성능 향상을 보인다.

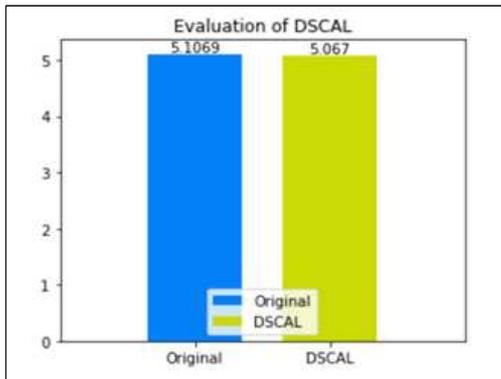


그림 10. DSCAL 성능평가
Fig. 10. Evaluation of DSCAL

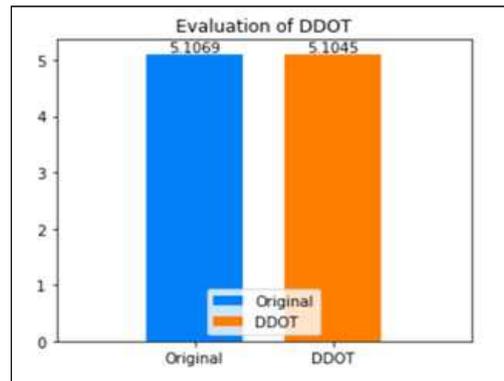


그림 11. DDOT 성능평가
Fig. 11. Evaluation of DDOT

5.4 SCOPY+DSCAL+DDOT 성능평가

적운 모수화 코드의 기존 연산 대신 앞서 제시한 세 함수를 모두 적용하여 연산한 결과는 그림 12와 같다. 기존 소스코드는 평균 5.1069초, 세 함수가 모두 적용된 코드는 5.0633초의 성능을 보이며 이는 기존 대비 0.8537% 성능이 향상되었음을 보인다.

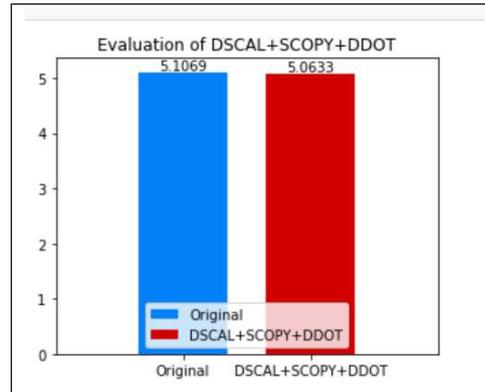


그림 12. DSCAL+SCOPY+DDOT 성능평가
Fig. 12. Evaluation of DSCAL+SCOPY+DDOT

6. 결론

본 논문에서는 SCAM에 포함되어 대기 연산을 담당하고 있는 적운 모수화 코드 Unicon의 성능 향상을 위하여 LAPACK BLAS 라이브러리 중 level1의 function 세 가지를 적용하는 방안을 제시하였다. 그리고 이를 구축된 환경에서 SCAM step 2088 (즉, 29일)을

기준으로 실행 및 분석한 결과 SCOPY 함수 적용은 0.4053% 성능 향상을, DSCAL 함수 적용은 0.7812%의 성능 향상을, DDOT 함수 적용은 0.0469%의 성능 향상을 보여주었으며 모든 함수를 종합하여 적용한 결과 0.8537%의 성능 향상을 보였다. 이는 동일한 CPU 환경에서 추가적인 하드웨어의 개입 없이 고밀도 선형 대수 연산을 위한 고성능 라이브러리인 BLAS를 사용하여 성능을 향상시킬 수 있음을 의미한다. 또한, 본 논문에서 제안한 LAPACK BLAS 라이브러리 적용 방법이 SCAM의 성능 향상을 이끌어내었음을 보여준다.

REFERENCES

- [1] Jeonbuk National University, [online] <https://wz3.jbnu.ac.kr/cml/11846/subview.do>
- [2] NCAR - CESM(Community Earth System Model), Single Column Atmospheric Model (SCAM), [online] <https://www.cesm.ucar.edu/models/simpler-models/scam/index.html>
- [3] LAPACK — Linear Algebra PACKage, [online] <https://netlib.org/lapack/>
- [4] BLAS (Basic Linear Algebra Subprograms), [online] <https://netlib.org/blas/>
- [5] LAPACK BLAS, [online] https://www.hpc.kau.st.edu.sa/sites/default/files/files/public/Shaheen_training/Numerical_libraries_HPC101_KSL_2018.pdf
- [6] Dongarra, Jack J., James W. Demmel, and Susan Ostrouchov. "LAPACK: a linear algebra library for high-performance computers." *Computational Statistics*, vol. 1, pp. 23-28, 1992.
- [7] Park, Sungsu. "A unified convection scheme (UNICON). Part I: Formulation." *Journal of the Atmospheric Sciences*, vol. 71, no. 11, p. 3902-3930, 2014.
- [8] Merchant Farhad, "Accelerating BLAS and LAPACK via efficient floating point architecture design.", *Parallel Processing Letters*, vol. 27, no. 03n04, pp. 1750006, 2017.
- [9] Kim, Kyungjoo, et al. "Designing vector-friendly compact BLAS and LAPACK kernels." *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1-12, 2017.
- [10] Phillips, Chris. "The performance of the BLAS and LAPACK on a shared memory scalar multiprocessor." *Parallel computing* vol. 17, no. 6-7, pp. 751-761, 1991.
- [11] Bjørstad, P. E., and T. Sørveik. "Data-parallel BLAS as a basis for LAPACK on massively parallel computers." *Linear Algebra for Large Scale and Real-Time Applications*, pp. 13-20, 1993.
- [12] "Lee, et al., "Performance Improvement of Cumulus Parameterization Code by Unicorn Optimization Scheme", Korea Institute of Information, Electronics, and Communication Technology, vol. 15, pp. 124-133, 2022.
- [13] BLAS routines: naming conventions, [online] <https://www.cse-lab.ethz.ch/wp-content/uploads/2020/10/Linear-Algebra-BLAS-ISA-Pipelining.pdf>
- [14] LAPACK: SCOPY, [online] https://netlib.org/lapack/explore-html/df/d28/group__single__blas__level1_ga24785e467bd921df5a2b7300da57c469.html#ga24785e467bd921df5a2b7300da57c469
- [15] LAPACK: DSCAL, [online] https://netlib.org/lapack/explore-html/de/da4/group__double__blas__level1_ga793bdd0739bbd0e0ec8655a0df08981a.html
- [16] LAPACK: DDOT, [online] https://netlib.org/lapack/explore-html/de/da4/group__double__blas__level1_ga75066c4825cb6ff1c8ec4403ef8c843a.html

저자약력

신 대 영 (Dae-Yeon Shin)

[학생회원]



<관심분야>

- 2020년 3월 ~ 현재: 창원대학교 학부 과정

IoT, 실시간 분산 멀티미디어

조 예 린 (Ye-Rin Cho)

[학생회원]



- 2020년 3월 ~ 현재: 창원대학교 학부 과정

〈관심분야〉

IoT, 실시간 분산 멀티미디어

정 성 욱 (Sung-Wook Chung)

[중신회원]



- 2010년 8월: CISE dept. Univ. of Florida, USA, (Ph.D)
- 2010년 10월 ~ 2012년 2월: KT 종합기술원 중앙연구소 선임연구원
- 2012년 3월 ~ 현재: 창원대학교 컴퓨터공학과 정교수

〈관심분야〉

IoT, 스마트모빌리티, HPC, 실시간 분산 멀티미디어 시스템