한국정보전자통신기술학회논문지(jkiiect)'22-4, Vol.15 No.2
논문 22-15-02-124 http://dx.doi.org/10.17661/jkiiect.2022.15.2.124

Unicon Optimization 기법을 이용한 적운모수화 코드 성능 향상

이창현, 김민규, 신대영, 조예린, 염기훈, 정성욱*

Performance Improvement of Cumulus Parameterization Code by Unicon Optimization Scheme

Chang-Hyun Lee, Min-gyu kim, Dae-Yeong Shin, Ye-Rin Cho, Gi-Hun Yeom,

Sung-Wook Chung*

요 약 하드웨어 기술이 발달하고 수치 모델 방식이 고도화됨에 따라 더욱 정밀한 기상예보를 진행할 수 있게 되었다. 본 논문에서는 CESM의 간소화 버전인 SCAM에 포함된 적운모수화 코드 (Unicon, Fortran)를 최적화하고 유지보수성을 증가시키기 위해 Loop Vectorization, Dependency Vectorization, Code Modernization 3가지가 결합한 Unicon Optimization 기법을 제안하고 이를 테스트 하기 위하여 SCAM 전체 실행 구조도를 제시하였다. 본 논문에서는 구축한 SCAM 실행 환경에서 논문에서 제안한 Unicon Optimization 기법을 테스트 하였고 기존 소스 코드 대비 Loop Vectorization은 3.086% Dependency Vectorization은 0.4572% 성능 향상을 이끌어 냈다. 그리고 이를 모두 적용한 Unicon Optimization의 경우 기존 소스 코드 대비 3.457%의 성능 향상을 이끌어 냈다. 이는 본 논문에서 제안한 Unicon Optimization 기법이 우수한 성능을 제공하고 있음을 입증한다.

Abstract With the development of hardware technology and the advancement of numerical model methods, more precise weather forecasts can be carried out. In this paper, we propose a Unicon Optimization scheme combining Loop Vectorization, Dependency Vectorization, and Code Modernization to optimize and increase Maintainability the Unicon source contained in SCAM, a simplified version of CESM, and present an overall SCAM structure. This paper tested the unicorn optimization scheme in the SCAM structure, and compared to the existing source code, the loop vectorization resulted in a performance improvement of 3.086% and the dependency vectorization of 0.4572%. And in the case of Unicorn Optimization, which applied all of these, the performance improvement was 3.457% compared to the existing source code. This proves that the Unicorn Optimization technique proposed in this paper provides excellent performance.

Key Words : Unicon Optimization, Performance Imporovement, SCAM, CESM, Global Climate Model

서론

하드웨어 기술이 발달하여 고성능의 슈퍼 컴퓨팅이 가능해지고 과학적으로 수치 모델 방식이 고도화됨에 따라 더욱 정밀한 기상예보를 진행할 수 있게 되었다. 그래서 각국의 기상청에서는 고성능의 슈퍼컴퓨터에 기상모델(Climate Model)을 이식하여 기상예보를 수행하고 있다.

그리고 한국 기상청 (KMA, Korea Meteorological Administration)은 그림 1과 같이 기상 예보를 위한 선진기술의 도입과 기상 전문인력 양상을 위해 미국의 국

This work was supported by Korea Instiute for Advancement of Technology(KIAT) grant funded by the Korea government(MOTIE) (P0017006, The Competency Development Program for industry Specialist)
*First & corresponding Author : Department of Computer Engineering, Changwon National University (swchung@changwon.ac.kr)
Received April 18, 2022 Revised April 20, 2022 Accepted April 23, 2022

립대기 과학 연구소(NCAR, National Center for Atmospheric Research)와 2004년에 기술협력(2004.03.22)을 체결[1]하였고 전 지구 기상예보를 위해 2012년 영국 기상청(met office)과 기상 관련 공동운영 협약 (2012.06.26.)을 맺어[2] 전 지구 기상모델(GCM, Global Climate Model)[3]에 관한 자료를 공유하기로 하였으며, 더 나아가 2014년에 영국 기상청으로부터 Glosea5(Global Seasonal Forecast System)이라고 하는 전 지구 기상모델을 슈퍼컴퓨터에 도입[4]하여 현재(2022)까지 기상예보를 수행하고 있다.

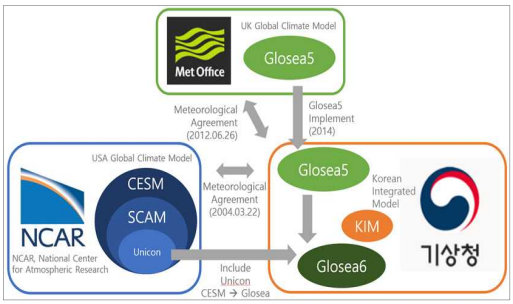

그림 1. NCAR, KMA, Met Office 협약 현황
Fig. 1. Convention of NCAR, KMA, Met Office

여기서, 한국 기상청은 실제 기상예보를 수행할 때, 한국이 자체적으로 개발한 기후 모델인 KIM(Korean Integrated Model)[5] 모델과 영국에서 도입한 Glosea5를 병행하여 사용하고 있으며, 전 지구 모델인 Glosea5는 대기 모델(UM), 해양 모델(NEMO), 해빙 모델(CICE) 그리고 지면 모델(JULES) 4가지로 구성된다[6]. 그런데, Glosea5가 대기 모델을 수행할 때, 적운모수화(Cumulus Parameterizations)라고 불리는 기상현상에 대한 시뮬레이션이 미흡하여 개선할 필요가 있다.

그래서 Glosea5의 새로운 버전인 Glosea6에서 해당 부분을 개선했으며[7], 더 나아가 그림 1 과같이 미국의 전 지구 모델 CESM의 간소화 버전인 SCAM 모델에 포함된 적운모수화 코드(Unicon, Fortran)를 Glosea6에 적용할 필요가 있다. 따라서 이를 수행하기 전에 Unicon 소스 코드를 분석하고 그에 대한 테스트 및 검증할 필요가 있으며, 원활한 코드의 실행을 위해 소스 코드 최적화 기법을 통한 소스 코드 성능을 향상할 필요가 있다.

CESM(Community Earth System Mode)은 대기, 해양, 해빙 그리고 지면 4가지의 모델과 기타 구성 요소로 구성된 전 지구 기상모델 시스템으로 미국의 국립대기 연구 센터(US National Center for Atomospheric Reserch, NCAR)에서 개발되었다[8]. CESM은 전 지구 기상모델로, 과거, 현재 그리고 미래에 대한 기후 변화를 4개의 모델(대기, 해양, 해빙, 지면) 관점에서 그림 2와 같이 전 지구를 일정한 격자(수평, 수직) 크기로 구분하여, 수치 모델링을 통해 격자들을 미분 및 적분하여 얻은 결과를 통해 기상을 시뮬레이션하는 기후 모델이다.

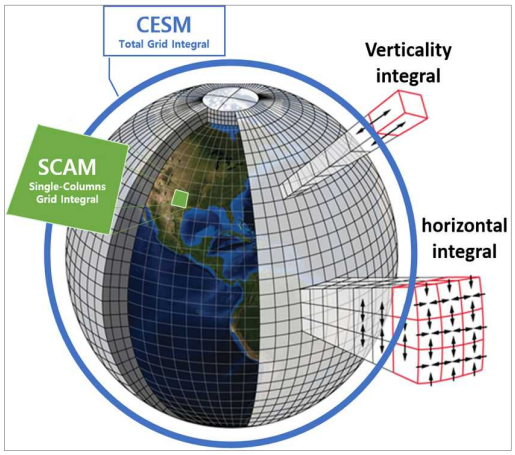

그림 2. CESM, SCAM 수치 모델링 적분 과정[9]
Fig. 2. Integration process of CESM, SCAM

SCAM(Single-Columns Atmospheric Model)은 CESM의 간소화 버전으로 주로, 특정 기상 연구를 위해 사용된다. SCAM은 4가지 모델 모두 고려하는 CESM과 달리 그림 2와 같이 오직 대기 모델(Atmosphric)에만 초점을 맞추어 수행되며, 오직 단일 격자(Single-Columns)의 적분을 통해 기상을 시뮬레이션하는 기후 모델이다[10].

Unicon 적운모수화 (Cumulus Parameterization)는 CESM, SCAM의 대기 모델을 구성하는 전체 코드의 일부이며, 대기의 기상 현상 중

하나인 적운모수화를 측정하기 위해 사용된다.

본 논문에서는 현재 CESM에서 사용하고 있는 Unicon 적운모수화 소스 코드를 한국 기상청에서 전 지구 기상예보를 위해 사용하고 있는 Glosea6에 적용하기 전에 Unicon 소스 코드의 이식성을 테스트하기 위해서 소스 코드를 실행할 수 있는 SCAM의 실행환경을 구축하고 이를 분석한다. 그 후, 본 논문에서 제안하는 최적화 기법을 통해 적운모수화 소스 코드를 최적화하여 이에 대한 이식성 검증을 진행하는 것이 주요 목적이다.

관련 연구(Related Work)

서론에서 기상예보에 대한에 대한 전반적인 현황[1][2]에 대해 설명하였고 기상모델 중 하나인 전 지구 기상모델에 관해 설명하였다[3]. 그리고 한국 기상청에서 사용하는 전 지구 기상모델 KIM[5], Glosea[6][7]에 대하여 소개하였고 미국의 전 지구 기상모델 CESM[8], SCAM[10]에 대하여 소개하였다. Randolph, et al.은 OOP(Object-Oriented Program), 이 가능한 Extended Fortran compiler에 대하여 OOP의 계산 속도에 대한 문제를 제시하고 이를 해결하기 위한 새로운 Optimization 방법들을 제시하였다[11]. Wayne, et al.은 Vector Architectures를 가지는 Fortran 코드들에 대하여 DO Loops를 변환하는 Fortran Optimization 방법을 제시하였다[12]. Vikram, et al.은 병렬 컴파일이 가능한 High Performance Fortran(HPF) Codes를 실제 NAS SW에 대한 벤치마크를 제공하였다[13]. Fernando, et al.은 Fortran Legacy Code에 관하여 순차실행(Sequential Processing)과 병렬실행 (Parallel Processing)을 구분하여 Optimization 방법을 제시하였고 이를 OpenMP를 이용하여 구현하였다[14].

SCAM 구조도

SCAM 디렉터리 구조

SCAM 디렉터리 구조는 그림 3과 같이 SCAM의 입력 데이터와 관련된 cesm, data_I3D, data_IOP, data_LES 디렉터리, 실제 SCAM의 전체 소스 코드가 들어있는 codes 디렉터리, codes 디렉터리에서 일부 수정이 필요한 코드를 가져와 자신의 상황에 맞게 변경한 mods 디렉터리(적운모수화 Unicon이 포함되어 있음), SCAM의 실행 스크립트가 들어있는 csh(c-shell) 디렉터리, 모델이 수행된 모든 결과(목적코드, 실행파일, 실행로그, 결과파일)가 저장되는 runs 디렉터리가 있으며 runs 디렉터리의 로그들(compile, running, both time)을 링크시켜 각각의 로그들에 대한 참조를 쉽게 만든 log 디렉터리와 SCAM 실행에 대한 결과물을 참조하는 output 디렉터리로 구성된다.

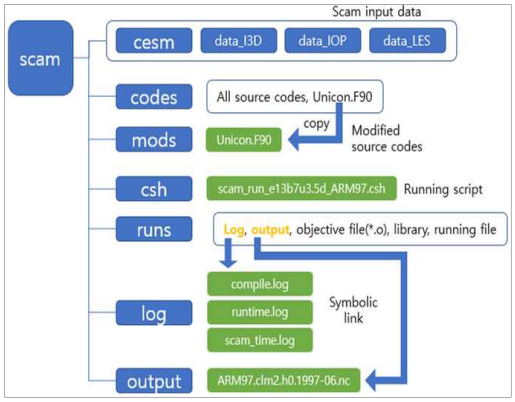

그림 3. SCAM 시스템 디렉터리 구조도
Fig. 3. Directory Structure of SCAM System

SCAM 전체 구조도

적운모수화 코드인 Unicon을 테스트하기 위해 2가지 방법으로 SCAM 디렉터리 구조를 재구축하였다. 첫 번째는 소스 코드를 원활히 관리하기 위해 소프트웨어 형상 관리 도구 중 하나인 SVN (Subversion)을 활용하여 구축하였고 두 번째는 실제로 모델이 수행되는 Cluster Server와 모델 수행을 요청하는 User를 rsync 와 SSH를 이용해 분리하였다. 따라서 위의 2가지 방법을 이용하여 구축된 SCAM 테스트 환경은 그림 4와 같이 Local PC Client, Gate, Cluster Server의 형태로 구성된다.

Local PC Client

Local PC Client는 그림 4의 SCAM 전체 시스템 구성에서 사용자 단말에 속하는 부분이다. 그리고 Local PC Client의 주요 기능은 SSH 프로토콜을 사용하는 원격 접속 프로그램(putty, xshell, MobaXterm 등)을 이용해 사용자가 Gate에 있는 자신의 USER 계정에 접속하기 위한 인터페이스 역할을 한다. 이를 통해 사용자는 SCAM의 소스 코드를 분석 및 수정하여 SCAM의 컴파일 및 실행을 요청하고 그에 대한 모니터링을 수행한다. 그리고 지정된 디렉터리를 동기화시키는 linux의 rysnc라는 명령어를 통해 Cluster Server로부터 결과 데이터를 받아 이에 대한 테스트와 검증을 할 수 있다.

Local PC Client를 구축하기 위해서는 SSH를 지원하는 원격 접속 프로그램을 다운로드(본 논문에서는 MobaXterm을 사용하여 진행하였음)하여 Gate의 USER & Host 정보를 가져와 SSH 설정을 수행하기만 하면 된다.

Gate

Gate는 그림 4의 SCAM 전체 시스템 구성에서 사용자와 Cluster Server 사이를 연결해주는 부분이다. 그림 4를 보면, Gate의 계정은 USER 계정과 SCAM 계정으로 구분되며, Gate의 기능은 계정별로 별도의 다른 기능을 제공한다. Gate의 USER 계정의 경우 각각의 Local PC Client의 사용자가 자신의 USER 계정에 맞게 접속하여 앞서 Local PC Client에서 언급한 다양한 활동들을 할 수 있는 Home 환경을 제공하며, 각자 자신의 SVN branch와 연동되어 있다. Gate의 SCAM 계정의 경우 Gate의 USER 계정과 Cluster Server의 SCAM 계정 사이에서 데이터와 명령어를 전달하는 기능을 수행하며, SVN의 trunk(main stream)와 연동되어 있다. 이를 통해 각각의 사용자는 각 소스 코드의 분기점마다 branch를 만들어 분기점에 대해서만 소스 코드를 테스트할 수 있으며, 테스트가 완료되고 나면 SCAM 계정의 trunk와 병합할 수 있다.

Gate를 구축하기 위해서는 Gate의 USER 계정과 SCAM 계정을 양방향으로 SSH와 rsync를 연결하고 Gate의 SCAM 계정과 Cluster Server의 SCAM 계정과 단방향으로 SSH를 연결하고 양방향으로 rsync를 연결한다.

Cluster Server

Cluster Server는 그림 4의 SCAM 전체 시스템 구성에서 실제로 SCAM을 실행하는 부분이다. Cluster

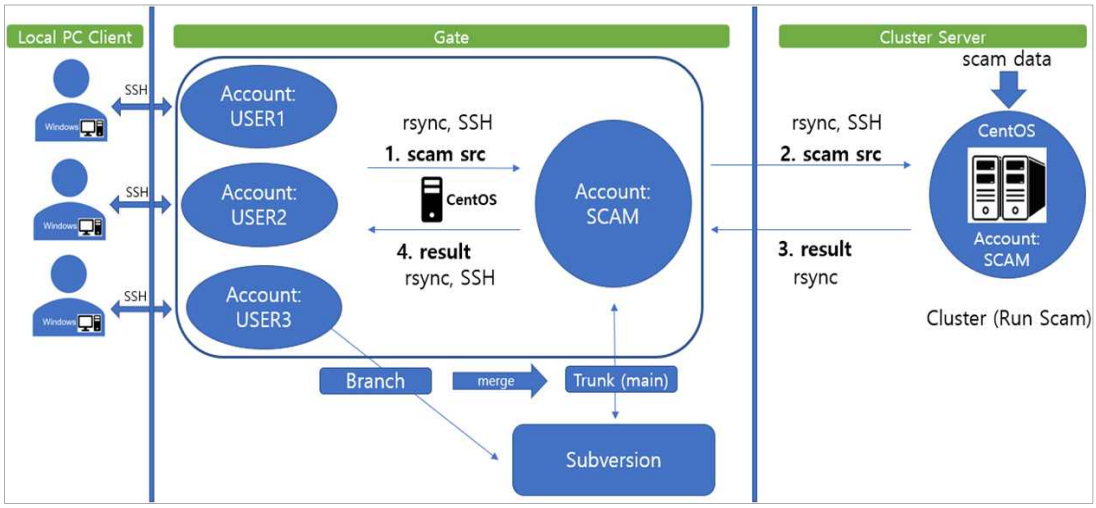

그림 4. SCAM(Single-Columns Atmospheric Mode) 전체 시스템 구조도
Fig. 4. Overall Configuration of SCAM(Single-Columns Atmospheric Mode) System

Server는 여러 노드로 구성되어 있으며, Gate로부터 명령을 지시받아 Server의 batch system에 해당 명령을 전달하여, SCAM 모델을 구동하고 그에 관한 결과를 반환하여 Gate의 USER 계정에 넘겨주는 역할을 가진다.

Cluster Server를 구축하기 위해서는 실제로 모델을 구동하기 위한 SCAM의 다양한 Dependency Software를 설치해야 하며, 해당하는 Software는 표 1과 같다. 그리고 그림 3의 디렉터리 구조에서 Gate의 USER 및 SCAM 계정에는 소스 코드 정보를 저장하는 codes 디렉터리와 적운모수화 Unicon 소스 코드가 있는 mod 디렉터리만이 존재하며, Cluster Server의 SCAM 계정에는 SCAM의 Inpudata에 해당 해당하는 cesm, data_I3D, data_IOP, data_LES 디렉터리가 존재한다.

표 1. SCAM 실행을 위한 의존성 소프트웨어
Table 1. Dependency Software for SCAM

Name	Version
Linux(CentOS7)	7.6.1810
Intell oneAPI HPC (ifort)	2021.2.0
GNU compiler (gcc, g++)	2019.01.5
cmake	2.8.12.2
make	3.8.2
gmake	3.8.2
mpich	3.1.4
hdf5	1.10.3
netcdf-c	4.3.3.1
netcdf-fortran	4.4.2
nco	4.7.5
ncl-ncar	6.6.2

SCAM 전체 실행 과정

SCAM의 모델 전체 수행 과정은 그림 5와 같이 먼저 Local PC Clinet에서 SSH 원격 접속 프로그램을 이용하여 Gate의 USER 계정에 접속한다. 그리고 사용자가 SCAM 모델을 실행하게 되면 Gate의 USER 계정으로부터 SCAM으로 전환하게 된다. 그 후, Gate의 SCAM 계정은 Gate의 USER 계정에 있는 SCAM 소스 코드를 가져가 Cluster Server의 SCAM 계정으로 소스 코드를 전달하게 된다. 소스 코드 전달이 완료되고 나면, Cluster Server의 SCAM 계정에 SCAM 모델 수행을 요청하게 되고 Cluster Server의 SCAM 계정은 서버의 Batch System(pbs)에 SCAM Running Job을 전달하게 된다. 그 후, Batch System의 Job scheduler(torque)를 거쳐 모델이 실행된다. 여기서, SCAM 모델의 실행은 전달받은 SCAM의 소스 코드를 컴파일 후(모델을 실행하면 새로 컴파일이 진행한다) 모델이 run 하게 되면 각각 해당하는 compile log와 running log를 남기게 된다. 그리고 모델이 모두 정상적으로 종료가 되고 나면 nc 확장자를 가진 모델의 결과가 저장되며 rynch를 이용하여 Gate의 SCAM 계정을 거쳐 USER 계정으로 전달하게 된다. 또한, 이러한 과정들은 Local PC Client에서 원격 접속 프로그램을 통해 실시간으로 모니터링을 할 수 있다.

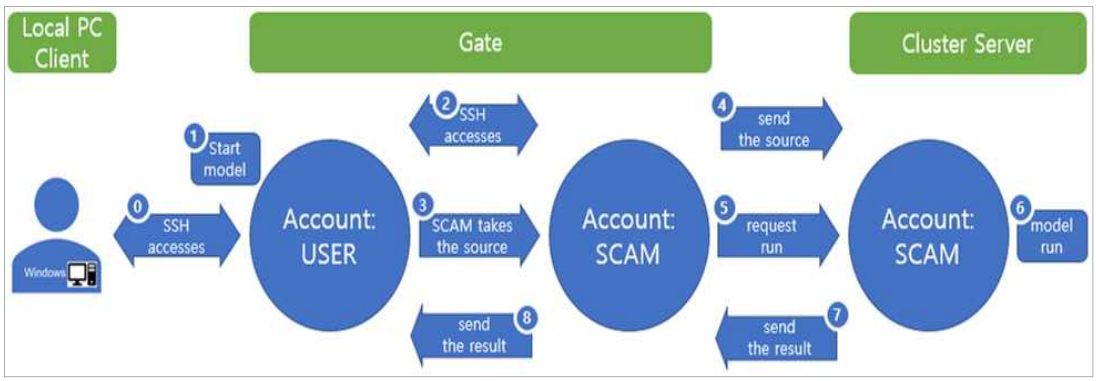

그림 5. SCAM(Single-Columns Atmospheric Mode) 실행 순서
Fig. 5. Running Process of SCAM(Single-Columns Atmospheric Mode) System

Unicon Optimization 기법

이 장에서는 본 논문에서 제안하는 Vectorization Optimization 기법을 구체적으로 제안하고 실제로 앞서 소개한 SCAM의 대기 모델 중 하나인 적운모수화 Unicon 소스 코드에 적용한다. 본 논문에서 제안하는 Unicon Optimization 기법은 총 3가지의 방법이 적용되며 소스 코드의 성능 향상을 위해 Scalar 연산을 Vectorization 하여 병렬성을 증진 시키고 유지보수성을 향상하기 위하여 작명법과 표기법을 최신화한다. 본 논문에서는 Unicon 적운모수화코드를 Optimization 할 때, 성능 향상을 위하여 Loop Vectorization과 Dependency Vectorization 방법을 제안하며, 소스 코드의 원활한 유지 보수를 위해 Code Modernization 방법을 제안한다. 그리고 본 논문에서 제안한 Unicon Optimization 기법은 3가지의 방법이 모두 함께 적용되어 동작하게 된다.

Loop Vectorization

Loop Vectorization 방법은 Loop 연산에 있어서 프로세서 수준의 기술인 벡터화(Vectorization)를 사용하여 소스 코드의 성능 향상을 끌어내는 기법이다. 그림 6과 같이 루프의 벡터화는 반드시 가장 안에 감싸여져 있는 루프만을 벡터화할 수 있으므로 루프가 중첩될 경우, 두 루프 사이의 식을 재구성하여 중첩된 루프를 제거한다. 그림 7은 실제 적용된 Unicon 소스 코드의 일부이다.

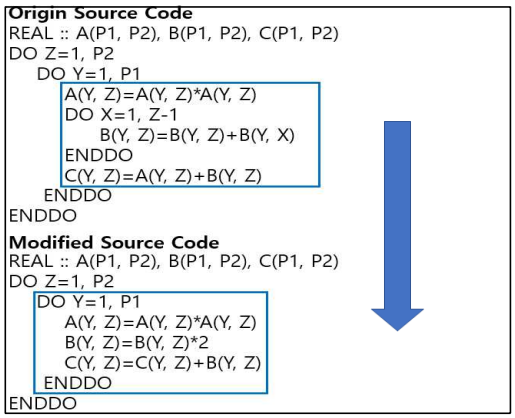

그림 6. Loop Vectorization 의사코드

Fig. 6. Loop Vectorization presudo-code

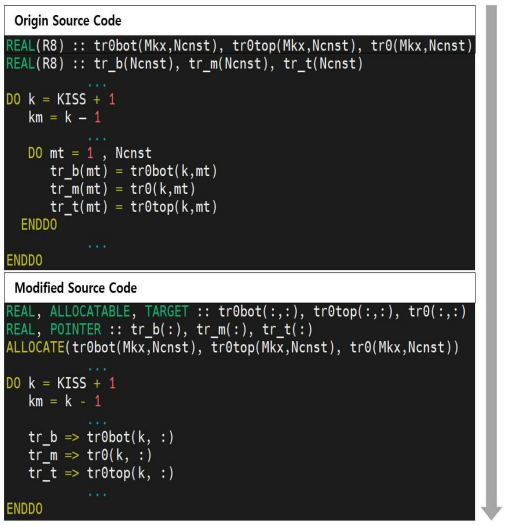

그림 7. Loop Vectorization 적용

Fig. 7. Apply of Loop Vectorization

Dependency Vectorization

Dependency Vectorization의 방법은 그림 8과 같이 하나의 반복문 안에 여러 개의 종속되지 않는 식이 존재하면 각각의 식을 서로 다른 루프로 분리화하

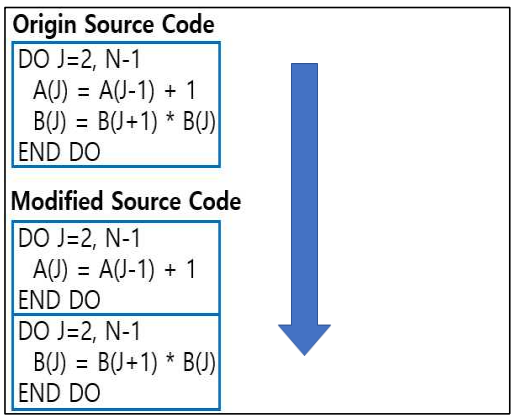

그림 8. Dependency Vectorization 의사코드

Fig. 8. Dependency Vectorization presudo-code

여 벡터화를 한다. 그림 9는 실제 적용된 Unicon 소스 코드의 일부이다.

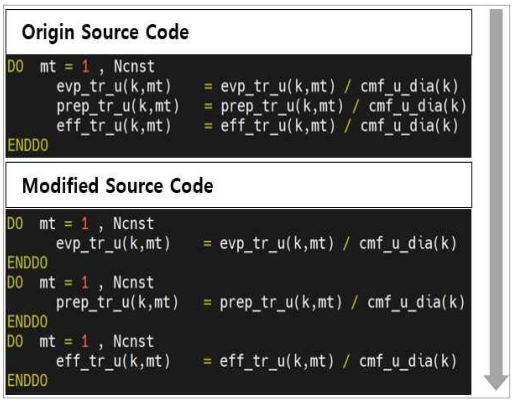

그림 9. Dependency Vectorization 적용

Fig. 9. Apply of Dependency Vectorization

Code Modernization

Code Modernization은 앞서 성능 향상을 목적으로 소개한 2가지의 방법과는 달리 원활한 유지 보수를 위하여 Fortran 소스 코드의 작성방식을 균일화시켜 유지보수성을 증가시키는 방법이다. 소스 코드의 원활한 유지 보수를 위해 제안하는 작성방식은 두 변수가 서로 같음을 표시하는 방법을 '.eq.'에서 '=='로 변경하고 예약된 약어(IF, ELSE, DO) 등은 모두 대문자로 변경하며, 한 라인에 너무 길게 작성된 코드는 여러 라인으로 분리한다. 그리고 변수 및 프로시저의 변수명일 때 가능한 소문자 알파벳으로 시작하고 여러 의미가 중첩되었으면 '_'로 구분시킨다. 그림 10은 실제 적용된 Unicon 소스 코드의 일부이다.

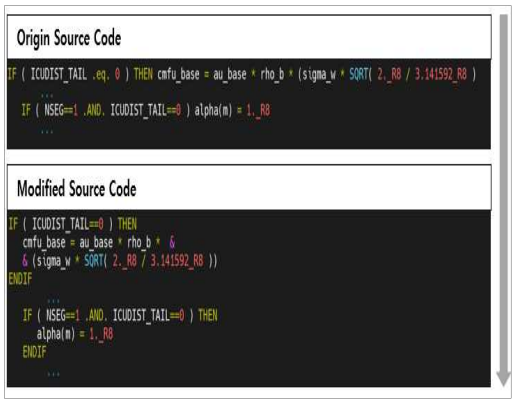

그림 10. Code Modernization 적용

Fig. 10. Apply Code Modernization

실험 결과 및 성능평가

제안한 기법의 효과적인 성능분석을 위하여 앞서 3장에서 구체적인 SCAM 실행환경을 제시하였다. 그리

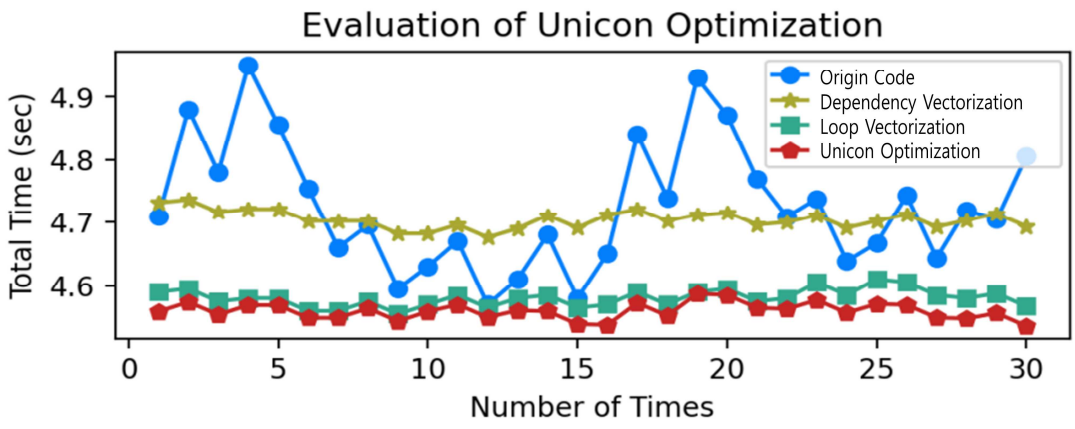

그림 11. Unicon Optimization 성능 평가

Fig. 11. Evaluation of Unicon Optimization

고 그림 4와 같은 SCAM의 전체 구조도에 대한 구체적인 시스템 스펙은 표 2와 같다. 여기서, Cluster Server의 노드는 총 2개이며, 표 2에서 노드 1이 메인 노드이고 노드 2는 서브 노드이며, 각각의 노드에 대한 시스템 스펙을 보여준다. 그리고 Cluster Server의 batch System은 PBS를 사용하고 Job Scheduler는 torque가 사용된다. 여기서 실험은 기법당 30회를 수행하고 그에 대한 평균을 도출하였고 1회에 대한 모델 실행 시간(Total time)은 컴파일 시간(compile time)과 모델 수행 시간(running time)의 합으로 계산한다.

표 2. 실험에 수행된 시스템 스펙
Table 2. 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 HDD: 4TB 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® 7-10700K CPU (8 cores 16 threads) GPU: NVIDIA GeForce GTX 3080 12GB RAM: 32GB HDD: 4TB SSD: 1TBGB OS: Window 10
Gate	CPU: Intel® i7-7700 CPU (4 cores 8 threads) GPU: GeForce GTX 1060 3GB RAM: 8GB HDD: 1TB SDD: 100GB OS: CentOS 7

Loop Vectorization 성능평가

4.1에서 제안된 기법을 3장에서 제시된 SCAM 실행환경에서 수행하였으며, Loop Vectorization을 적용한 소스 코드는 그림 11에서 붉은색 네모에 해당하며, 원본 소스 코드는 하늘색 동그라미에 해당한다. 그리고 그림 12와 같이 Loop Vectorization 기법은 평균 4.5805초, 기존 소스 코드는 4.7263초의 성능을 보여주며, 이는 기존 대비 3.086% 성능 향상을 보였다.

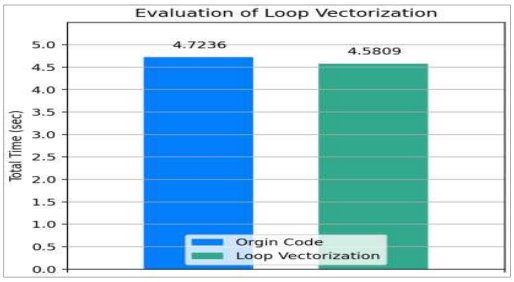

그림 12. Loop Vectorization 성능평가
Fig. 12. Loop Vectorization Evaluation

Dependency Vectorization 성능평가

4.2에서 제안된 기법을 3장에서 제시된 SCAM 실행환경에서 수행하였으며, Dependency Vectorization 을 적용한 소스 코드는 그림 11에서 노란색 별 모양에 해당하며, 원본 소스 코드는 하늘색 동그라미에 해당한다. 그리고 그림 13과 같이 Dependency Vectorization 기법은 평균 4.7047초, 기존 소스 코드는 4.7263초의 성능을 보여주며, 이는 기존 대비 0.457% 성능 향상을 보였다.

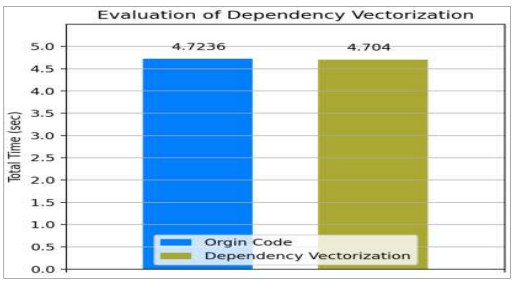

그림 13. Dependency Vectorization 성능평가
Fig. 13. Dependency Vectorization Evaluation

Unicon Optimization 성능평가

Unicon Optimization 기법은 앞서 제안한 4.1, 4.2, 4.3의 방법을 모두 적용한 기법으로 성능 향상을 위해 4.1(Loop Vectorization)과 4.2(Dependency Vectorization)의 기법을 제안하였고 오래된 Fortran 소스 코드를 유지 보수를 위하여 4.3(Code Modernization) 방법을 제안하였다. 그리고 3장에서 제안한 SCAM의 실행환경에서 SCAM 모델이 수행되

었다. Unicon Optimization을 적용한 소스 코드는 그림 11에서 초록색 오각형에 해당한다. 그리고 그림 14와 같이 Unicon Optimization 기법은 평균 4.5599초, 기존 소스 코드는 4.7263초의 성능을 보여준다. 이는 기존 대비 3.521%의 성능 향상을 보이며, Unicon Optimization 기법의 성능 우수성을 의미한다.

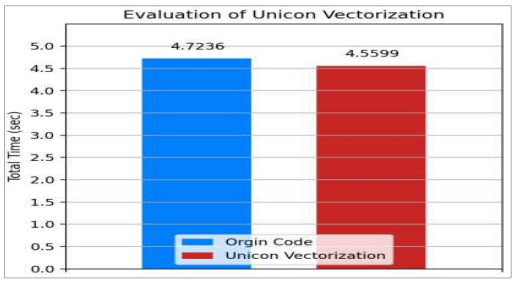

그림 14. Unicon Vectorization 성능평가
Fig. 14. Unicon Vectorization Evaluation

6. 결론

본 논문에서는 SCAM의 일부인 적운모수화 코드 Unicon을 Glosea6에 이식하기 전에 모델에 대한 이식성을 검증하고 그에 대한 성능 향상 및 유지보수성을 증가시키기 위하여 본 논문에서 제안한 3가지의 기법 Loop Vectorization, Dependency Vectorization, Code Modernization으로 구성된 Unicon Optimization 기법을 제시하였다. 그리고 이를 구축된 환경에서 테스트한 결과 Loop Vectorization은 3.086% 성능 향상을 Dependency Vectorization은 0.457% 의 성능 향상을 보여주었으며, 최종적으로 적용되는 Unicon Optimization의 경우 3.457%의 성능 향상을 끌어냈다. 이는 본 논문에서 제안한 Unicon Optimization 기법의 성능 우수성을 보여준다.

앞으로 적운모수화 Unicon 소스코드에 MPICH를 이용한 분산 처리와 GPGPU(CUDA)를 이용한 병렬처리 연구가 추가로 연구되어 더욱 많은 코드를 병렬화하여 성능 향상을 끌어낼 필요가 있을 것으로 예상된다.

REFERENCES

KMA - Exchange of Memorandum of Understanding between the Exchange of Memorandum of Understanding between the National Center for Atmospheric Research and Korea Meteorological Administration, [online]https://www.kma.go.kr/kma/news/press.jsp

KMA - Production of the world's best numerical forecast model signed a memorandum of understanding (MoU) with the UK Meteorological Administration, [online]https://web.kma.go.kr/notify/press/kma_list.jsp?bid=press&mode=view&num=1192374&page=161&field=subject&text

Seiji YUKIMOTO, et al., "A New Global Climate Model of the Meteorological Research Institute: MRI-CGCM3," Meteorological Society of Japan, vol. 90, pp.23-64, 2012

H. Ham, et al., "Performance Assessment of Weekly Ensemble Prediction Data at Seasonal Forecast System with High Resolution," Atmosphere pp.261-276, 2017

Y. Kwon, et al., "Overview and main specifications of the Korean numerical forecast model," Proceedings of Korea Meteorological Society Conference p.177, 2019

C. MacLachla, et al., "Global Seasonal forecast system version 5 (GloSea5):a high-resolution seasonal forecast system," Quarterly Journal of the Royal Meteorological Society pp.1072-1084, 2015

Hyeri Kim, et al., "The KMA Global Seasonal Forecasting System (GloSea6) - Part 1: Operational System and Improvements," Atmosphere. Korean Meteorological Society, Vol. 31, No. 3, pp.341-359, 2021

Kay, Jennifer E, et al., "The Community Earth System Model (CESM) large ensemble project: A community resource for studying climate change in the presence of internal climate variability," Bulletin of the American Meteorological Society, p.2015, 1333-1349

Jeonbuk National University, [online] https://wz3.jbnu.ac.kr/cml/11846/subview.do

NCAR - CESM(Community Earth System Model),

Single Column Atmospheric Model (SCAM), [online] https://www.cesm.ucar.edu/models/simpler-models/scam/index.html

Scarborough, Randolph G, et al., "Improved optimization of FORTRAN object programs" IBM Journal of Research and Development, pp.660-676, 1980

Cowell, Wayne R, et al., "Transforming FORTRAN DO loops to improve performance on vector architectures," ACM Transactions on Mathematical Software (TOMS), pp.324-353, 1986

Adve, Vikram, et al., et al., "High Performance Fortran compilation techniques for parallelizing scientific codes," IEEE Proceedings of the 1998 ACM/IEEE Conference on Supercomputing, pp.11-11, 1998

Tinetti, Fernando G, et al., "Fortran Legacy Code Performance Optimization: Sequential and Parallel Processing with OpenMP," IEEE 2009 WRI World Congress on Computer Science and Information Engineering, Vol. 2, pp.471-475, 20

저자약력

이 창 현 (Chang-Hyun Lee) **[학생회원]**

- 2017년 3월 ~ 2018년 2월: 창신대학교
- 2019년 3월 ~ 2021년 2월: 창원대학교 졸업
- 2021년 3월 ~ 현재: 창원대학교 대학원 석사 과정

〈관심분야〉 IoT, 클라우드 컴퓨팅, 실시간 분산 멀티미디어시스템

김 민 규 (Min-Gyu Kim) **[학생회원]**

- 2020년 3월 ~ 현재 : 창원대학교 학부 과정

〈관심분야〉 IoT, 실시간 분산 멀티미디어

신 대 영 (Dae-Yeong Shin) **[학생회원]**

- 2020년 3월 ~ 현재: 창원대학교 학부 과정

〈관심분야〉 IoT, 실시간 분산 멀티미디어

조 예 린 (Ye-Rin cho) **[학생회원]**

- 2020년 3월 ~ 현재 : 창원대학교 학부 과정

〈관심분야〉 IoT, 실시간 분산 멀티미디어

염 기 훈 (Gi-Hun Yeom) **[학생회원]**

- 2016년 3월 ~ 현재: 창원대학교 학부 과정

〈관심분야〉 실시간 분산 멀티미디어시스템

정 성 욱 (Sung-Wook Chung) **[종신회원]**

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

〈관심분야〉 IoT, 스마트모빌리티, HPC, 실시간 분산 멀티미디어 시스템