

그림 1. NCAR, KMA Met Office 협약 현황[7]
Fig. 1. Convention of NCAR, KMA Met Office

연구소(NCAR, National Center for Atmospheric Research)과 2004년에 기술협력을 체결[1]하였고 영국 기상청(Met Office)과 2012에 기상 관련 공동 운영 협약을 체결[2]하였으며, 더 나아가 2014년에 전 지구 기상모델(GCM, Global Climate Model)[3]인 Glosea5(Global Seasonal Forecast System)를 도입[4]하여 슈퍼컴퓨터 5호기에 이식하였으며, 한국 기상청은 Glosea5를 Glosea6로 버전 업그레이드를 진행하였다. 또한 2011년에 한국형 수치 예보 모델인 KIM(Korean Intergrated Model)[5] 개발사업에 착수하여 2020년에 한국형 수치 예보 모델을 현업으로 운영하고 있다.

그러므로 한국은 한국형 수치 예보 모델 KIM의 개발사업을 통해 전 세계에서 9번째 자체 전 지구 수치 예보 모델을 보유한 국가가 되었으며, 2022년 현재 전 지구 기상예보를 위해 Glosea5와 KIM를 병행하여 운영하고 있다.

CESM(Community Earth System Model)은 미국의 국립 대기과학 연구소의 기후 및 지구역학부(CGD, Global Dynamics Division)가 일차적으로 개발하고 국립과학부(NSF, National Science Foundation)가 상당한 자금을 지원하여 출시되었다[6]. CESM은 전 지구 기후모델(GCM)로써 대기, 해양, 해빙, 지면 4가지의 모델과 탄소 순환 및 기타 구성 요소로 구성되며, 그림 2와 같이 지구를 수평 방향과 수직 방향을 모두 고려하여 일정한 격자 크기로 나누는 격자점마다 대기의 상태와 운동에 대한 방정식을 계산하여 얻은 결과를 통해 기상을 시뮬레이션하는 전

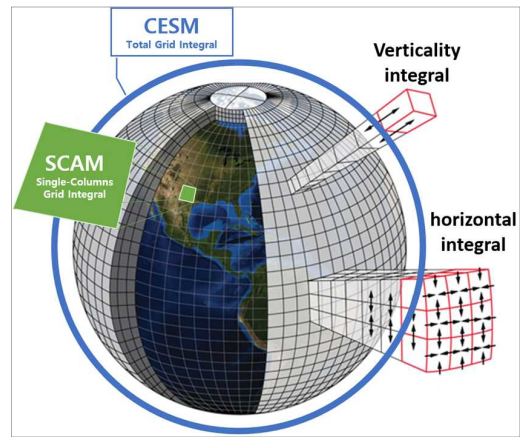


그림 2. SCAM 시스템 디렉터리 구조도[11]
Fig. 2. Directory Structure of SCAM System

지구 기후모델이다.

SCAM(Single-Columns Atmospheric Model)은 CESM의 간소화 버전으로 주로, 특정 기상 연구를 위해 사용된다[8]. SCAM은 Atmospheric(대기) 모델로 4가지를 모두 고려하는 CESM과 달리 오직 대기(Atmosphric)만 집중하여 모델을 수행하며, 지구에 대한 모든 격자가 아닌 오직 단 일 격자(Single-Columns)에 대한 수치 방정식을 계산하여 결과를 도출한다. 그리고 SCAM의 경우 step 이라는 단위로 모델이 순차적으로 수행되며 1step=1,200초이다. SCAM은 최대 2088번의 step을 수행할 수 있으며 이를 Day(일)로 환산할 시 대략 최대 29일까지 모델을 수행할 수 있다.

Intel OpenAPI는 Intel 에서 만든 개발자용 소프트웨어 개발 제품으로 크게 소스 코드를 컴파일할 수 있는 Compiler(Fortran = ifort, C = icc, C++ = icpc)와 소스 코드를 분석할 수 있는 Profiler를 포함하고 있으며, 다양한 분야에서 사용할 수 있는 Intel 관련 라이브러리를 제공한다[9]. 여기서 Intel Fortran Compiler의 경우 Intel OpenAPI에 포함되어 있다.

NVIDIA HPC SDK는 NVIDIA에서 만든 개발자용 소프트웨어 개발 제품으로 소스 코드를 컴파일할 수 있는 Compiler(Fortran = nvfortran = pgfortran, C = nvcc = pgcc, C++ = nvc++ = pgc++)와 소스

코드를 분석할 수 있는 Profiler를 포함하고 있으며, 다양한 분야에서 사용할 수 있는 NVIDIA 관련 라이브러리를 제공한다[10]. 여기서 PGI(The Portland Group, Inc) Fortran Compiler의 경우 NVIDIA HPC SDK에 포함되어 있으며, NVIDIA Compiler의 경우 과거 PGI Compiler(2022 현재 기준, NVIDIA Compiler = PGI Compiler)를 NVIDIA가 인수하여 개발한 컴파일러이다.

현재 한국 기상청은 앞서 이야기한 Glosea6에 적용된 모수화(Cumulus Parameterizations) 코드를 이식할 계획이 있으며, 해당 적용 모수화 코드를 미국의 전 지구 모델인 CESM의 간소화 버전인 SCAM 모델에서 도입을 시도하고 있다. 여기서, 해당 적용 모수화 코드의 이름은 Unicon이다. 따라서, 앞선 연구에서 이를 위해 SCAM 모델에 있는 Unicon 소스 코드를 분석하고 그에 대한 테스트 및 검증을 진행하였으며, 해당 소스 코드 분석 내용을 바탕으로 Unicon 코드의 최적화를 진행하였다[7].

본 논문은 앞선 연구에서 진행한 적용 모수화 코드 Unicon 최적화에서 더 나아가 SCAM 모델 전체에 대한 성능 향상을 끌어내는 것이 목적이며, 본 논문에서 제안하는 방법을 통해 SCAM 모델의 성능 향상을 끌어내고자 한다. 따라서, 기존에 Intel Fortran Compiler를 이용하여 수행된 SCAM 모델을 PGI Fortran Compiler로 변경하고 OpenACC API를 적용하는 방법으로 개발하고자 한다.

2. 관련 연구

기상 예측 모델인 SCAM에 대한 기본적인 개념에 대해서는 서론에서 설명하였다[7][8]. Michael Wolfe는 PGI 컴파일러를 이용하여 모델에 대한 컴파일러 설계에 대한 몇 가지 세부 사항을 제시하였다[12]. Douglas, et al은 PGI 컴파일러를 사용하여 Cray 시스템에서 단일 노드에 대하여 성능을 끌어내는 수단으로 컴파일 기능 몇 가지를 사용하는 예시를 제공하였다[13]. Chis J, et al.은 Intel Xeon 프로세스를 이용한 컴파일러 오프로드 소프트웨어 런타임 인프라를 소개하였다[14]. Chen, et al.은 Compiler 테스트 프로

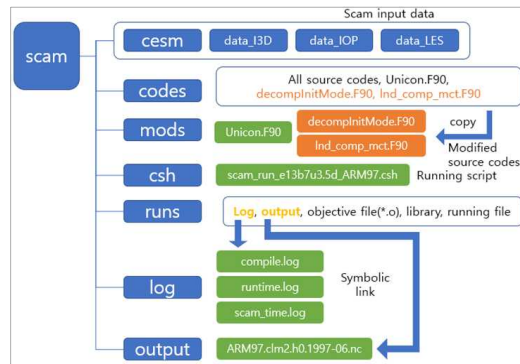


그림 3. SCAM 시스템 디렉터리 구조도

Fig. 3. Directory Structure of SCAM System

그램 구성 방법, 컴파일러 테스트를 효율적으로 하는 방법, 컴파일러 테스트에 발견된 버그를 조치하는 방법 등 컴파일러 최신 연구에 대한 포괄적인 요약은 제시하였다[15]. David, et al.은 C와 Fortran 같은 필수 언어에 대한 중요한 고급 프로그램 재구성기술에 대한 포괄적인 개요를 제시한다. 특히, 컴파일러가 수행할 수 있는 최적화에 대한 개요를 제시하고 있다[16]. Megan, et al.은 과학적 계산 모델에 대한 컴파일러 (GNU, PGI, Cray, Intel) 최적화 방법들을 제공하고 있다[17]. Dali, et al. 과학 코드를 더 잘 이해하기 위해 컴파일러 기반 기술을 사용하는 절차를 제시하고 있다[18].

3. SCAM PGI Compile With OpenACC

이 장에서는 본 논문에서 제안하는 SCAM PGI Compile with OpenACC 방법을 구체적으로 제안한다. SCAM PGI Compile with OpenACC 방법은 기존에 Intel Fortran Compiler로 수행되었던 SCAM 모델을 PGI Fortran Compiler로 변경한 후, OpenACC API 옵션을 이용하여 SCAM 모델의 러닝 속도를 향상하는 것이다.

3.1 SCAM 구조 및 실행 과정

SCAM의 디렉터리 구조는 Intel Fortran Compiler에서 수행되었던 모델의 구조와 그림 그리고 현재 개발 중인 Glosea6에 적용된 모수화

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

Name	Version
Linux(CentOS7)	7.6.1810
NVIDIA HPC SDK (pgfortran)	2022.2.0
GNU compiler (gcc, g++)	2019.01.5
cmake (GNU)	2.8.12.2
make (GNU)	3.8.2
gmake (GNU)	3.8.2
mpich (PGI)	3.1.4
hdf5 (PGI)	1.10.3
netcdf-c (PGI)	4.3.3.1
netcdf-fortran (PGI)	4.4.2
nco (PGI)	4.7.5
ncl-ncar (GNU)	6.6.2

(Cumulus Parameterizations) 코드 이식을 진행하고 있으며, 해당 적운 모수화 코드를 미국의 전 지구 모델인 CESM(Community Earth System Model)의 간소화 버전인 SCAM(Single-Columns Atmospheric Model) 모델에서 도입을 시도하고 있다. 여기서, 해당 적운 모수화 코드의 이름은 Unicon이다. 따라서, 이를 위해 SCAM 모델에 있는 Unicon 소스 코드를 분석하고 그에 대한 테스트 및 검증을 진행하였으며, 해당 소스 코드 분석 내용을 바탕으로 Unicon 코드의 최적화를 진행하였다

[6]. 하지만, 컴파일러가 Intel 에서 NVIDIA로 변경되었기 때문에 csh 디렉터리 내의 실행 스크립트와 mods 디렉터리 내의 파일이 일부 추가되었으며, 원활한 컴파일을 위해 소스 코드 수정을 진행하였다.

SCAM 전체 구조는 Intel Fortran Compiler 에서 수행되었던 모델의 구조와 같게 ‘Local PC Client <-> Gate <-> Cluster Server’로 구성된다. 여기서, Local PC Client는 사용자 단말을 의미하고 Gate는 사용자와 Cluster Server를 연결해주는 로그인 노드 역할을 하며, Cluster Server는 실제로 모델 수행을 위해 계산을 담당하는 계산 노드의 역할을 가진다. 그리고 해당 모델의 구조는 그림 4와 같다[7].

SCAM의 실행 과정은 Intel Fortran Compiler에서 수행되었던 모델의 실행 과정과 같으며, 각각의 구조(Gate USER <-> Gate SCAM <-> Cluster USER)들은 SSH(Secure Shell) 프로토콜을 통해 통신하며, RSYNC 파일 전송 프로그램을 이용하여 데이터를 주고받는다. 그리고 이에 대한 SCAM의 세부 실행 과정은 그림 5과 같다[7].

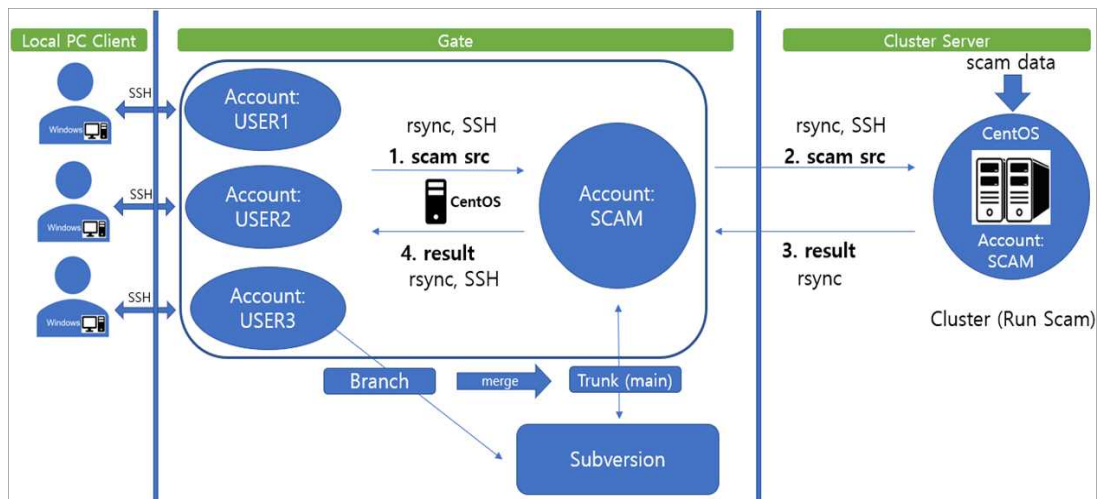


그림 4. SCAM(Single-Columns Atmospheric Mode) 전체 시스템 구조도[7]
Fig. 4. Overall Configuration of SCAM(Single-Columns Atmospheric Mode) System

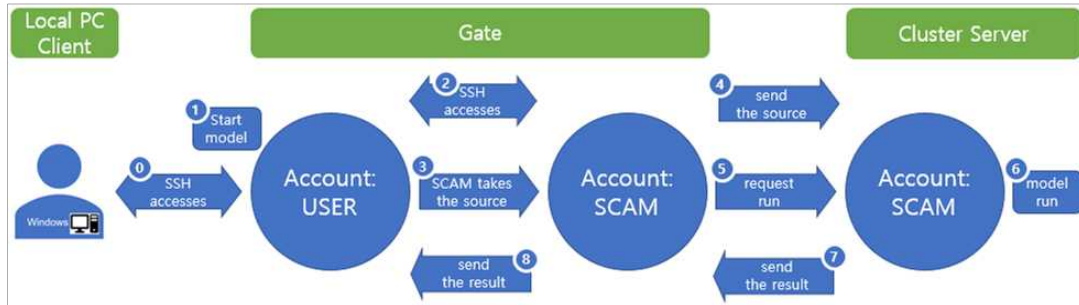


그림 5. SCAM(Single-Columns Atmospheric Mode) 실행 순서[7]
 Fig. 5. Running Process of SCAM(Single-Columns Atmospheric Mode) System

3.2 SCAM 컴파일 환경 구축

SCAM 컴파일 환경을 구축하기 위해서는 기존의 Intel Fortran Compiler 환경에서 수행되었던 모델이 PGI Fortran Compiler 환경에서 모델이 수행되어야 하므로, 시스템 환경 변수 재설정, 컴파일 옵션 조정, 관련 의존성 SW & 라이브러리 설치 그리고 소스 코드 수정 과정이 필요하다.

환경 구축에 대한 전체적인 방법을 설명하자면, 기존에 Intel Compiler로 설치되었던 Software Dependency의 경우 NVIDIA Compiler와 호환이 안 되므로 NVIDIA Compiler를 이용하여 다시 설치를 진행하였고 해당 SW 목록은 표 1과 같다.

그리고 Software Dependency가 변경되었기 때문에 그에 맞춰 시스템 환경 변수를 다시 설정할 필요가 있으며, 컴파일 옵션 또한 수정을 진행해야 한다. 컴파일 옵션이 수정되었기에 일부 코드의 경우 컴파일이 원활히 진행되지 않을 경우가 발생하였고 해당 부분에 대하여 소스 코드를 수정하였다.

3.3 PGI OpenACC를 이용한 SCAM 컴파일

컴파일을 NVIDIA로 변경하여 모델이 성공적으로 수행되는 것을 확인한 후, OpenACC 라이브러리를 기반으로 컴파일 옵션을 조정하여 SCAM 소스 코드에 대한 성능 향상을 진행하였다. 표 2는 SCAM에 적용된 컴파일러 옵션이다.

표 2. SCAM NVIDIA 포트란 컴파일 옵션
 Table 2. SCAM NVIDIA Fortran Compiler Option

Option	contents
-acc	Enable OpenACC
-c	Stops after the assembly phase
-Mfree	Source are 90/95 freeform in fortran
-debug	Enable code debugging for profiling
-l	Include SCAM library

4. 실험 결과 및 분석

이 장에서는 기존의 Intel Fortran Compiler의 성능과 비교하여 PGI Fortran Compiler로 변경한 후, 어떠한 조치도 취하지 않고 수행한 모델의 성능과 3장에서 제안한 SCAM PGI Compiler with OpenACC 방법을 적용한 PGI Fortran Compiler(OpenACC)로 수행한 모델의 성능을 비교 및 분석한다.

성능 측정 방법은 컴파일 시간(Compile Time)과 러닝 시간(Running Time)을 측정하여 각각 따로 비교 분석한다. 여기서 SCAM 모델 또한 소프트웨어이기 때문에 처음 1번 모델이 컴파일이 진행되어 실행 파일(Executable File = Binnray File)이 생성되어 모델이 수행(Running)되고 나면, 그다음부터는 모델의 소스 코드를 변경하거나 컴파일 옵션을 변경하지 않는 이상 SCAM 모델을 다시 컴파일할 필요가 없이 모델을 수행할 수 있다. 즉, 이렇게 생성된 실행 파일은 예측일에 따라 반복 수행되어야 하므로 러닝 시간이 더욱 중요하다. 본 실험에서는 앞서 언급한 방법을 적용한 SCAM 모델을 수행하여 컴파일 시간, 러닝 시간, 예측일에 따른 러닝 시간을 비교한다.

표 3. 실험에 수행된 시스템 스펙

Table 3. 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 SSD: 100GB OS: CentOS 7

여기서, SCAM모델이 수행되어 예측하는 기후에 대한 날짜의 기본 단위를 step이라고 하며, 1step의 경우 1200초(=20분)를 가진다. 그리고 1Day를 1step으로 환산할 경우, 72step(20분*3 = 1시간, 1Day = 24시간이므로 3 * 24 = 72)이 1Day이다. 여기서, 본 논문에서는 SCAM 모델 예측 기간을 29일(1회 모델 수행 시 최대 수행시간)을 기준으로 컴파일 시간과 러닝 시간을 측정할 것임으로 SCAM의 예측 기간은 총 2088step(29*72 =2088)로 구성된다.

본 연구에서 성능을 측정하기 위해 사용한 하드웨어 시스템 스펙은 실제로 모델 수행을 위한 Cluster로 총 2개의 노드를 사용하였으며, 1번 노드가 메인 노드이다. 그리고 Cluster의 Batch System은 PBS를 사용하였고 Job Scheduler는 torque를 사용하였다. 3.1장에서 제시한 시스템 전체에 대한 세부 내용은 표 3에 자세히 설명하였다.

4.1 컴파일 시간(Compile Time) 비교

그림 6은 기존의 Intel Fortran Compiler, PGI Fortran Compiler 그리고 PGI Fortran

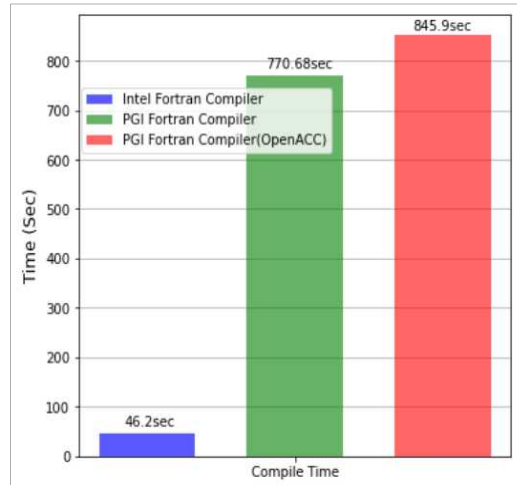


그림 6. SCAM 컴파일 시간 비교

Fig. 6. Compare SCAM Compile Time

Compiler(OpenACC)를 수행한 후, 컴파일 시간을 비교한 그림이다.

컴파일 시간을 비교해 보면, Intel Fortran Compiler의 경우 46.2초, PGI Fortran Compiler의 경우 770.68초, PGI Fortran Compiler (OpenACC)의 경우 845.9초가 소요되었다. 즉, 기존의 Intel 에서 PGI로 컴파일러가 변경되면 컴파일 시간이 724.48초가 더 소요되었고 OpenACC를 추가로 적용할 경우, PGI only보다 75.22초가 더 소요되었다. 이것은 서버 하드웨어가 Intel CPU인 것과 컴파일러 차이에 의한 것으로 보이며 특히 PGI OpenACC의 경우 Running Time의 수행시간을 줄이기 위한 컴파일 오버헤드로 볼 수 있다. 특히, SCAM 기상 예측 모델의 경우, 1회 컴파일이 되고 나면, 그 후 예측 일자 및 예측지역에 따른 반복적 추가 실행시간이 중요하며 이는 다음 결과에서 비교분석 한다.

4.2 러닝 시간(Running Time) 비교

그림 7은 기존의 Intel Fortran Compiler, PGI Fortran Compiler 그리고 PGI Fortran Compiler(OpenACC)를 수행한 후, 러닝 시간을

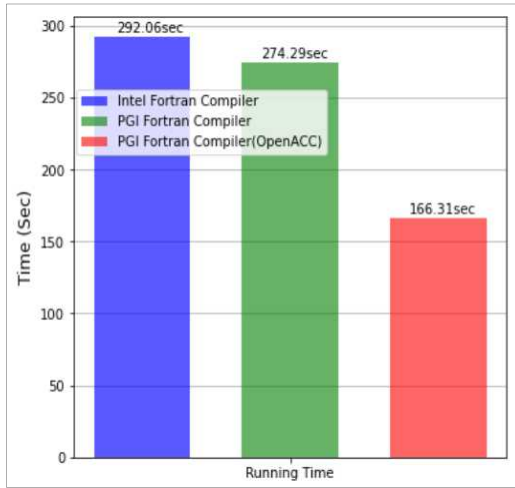


그림 7. SCAM 러닝 시간 비교
Fig. 7. Compare SCAM Running Time

비교한 그림이다.

러닝 시간을 비교해 보면, Intel Fortran Compiler의 경우 292.06초, PGI Fortran Compiler의 경우 274.29초, PGI Fortran Compiler (OpenACC)의 경우 166.31초가 소요되었다. 이는 기존 Intel 에서 PGI로 컴파일러가 변경되면, 러닝 시간의 경우 6.08%의 성능 향상이

있으며, OpenACC를 추가로 적용할 경우, 기존 대비 43.05%의 성능 향상이 이루어졌다는 것을 의미한다. 특히, 기상예측 모델에서는 1회성의 컴파일시간 보다 예측기간 및 지역에 따른 반복적인 러닝시간 향상이 중요한데 43.05%의 성능향상은 우수한성능으로 분석된다.

4.3 예측 기간 대비 모델 수행시간 분석

그림 8은 기존의 Intel Fortran Compiler, PGI Fortran Compiler 그리고 PGI Fortran Compiler(OpenACC)를 수행한 모델에 대하여, 예측 기간(Day 기준) 대비 러닝 시간의 변화량을 나타낸 그래프이며, 해당 그래프는 모델 예측 기간이 길수록 선형으로 증가하고 있으며 모델의 예측 기간이 길수록 Intel Fortran Compiler와 PGI Fortran Compiler(OpenACC)의 러닝 시간의 차이는 증가하고 있다. 예측 기간 29일을 기준으로 Intel Fortran Compiler와 PGI Fortran Compiler의 러닝 시간을 비교하면 43.05%의 성능 향상을 보여준다.

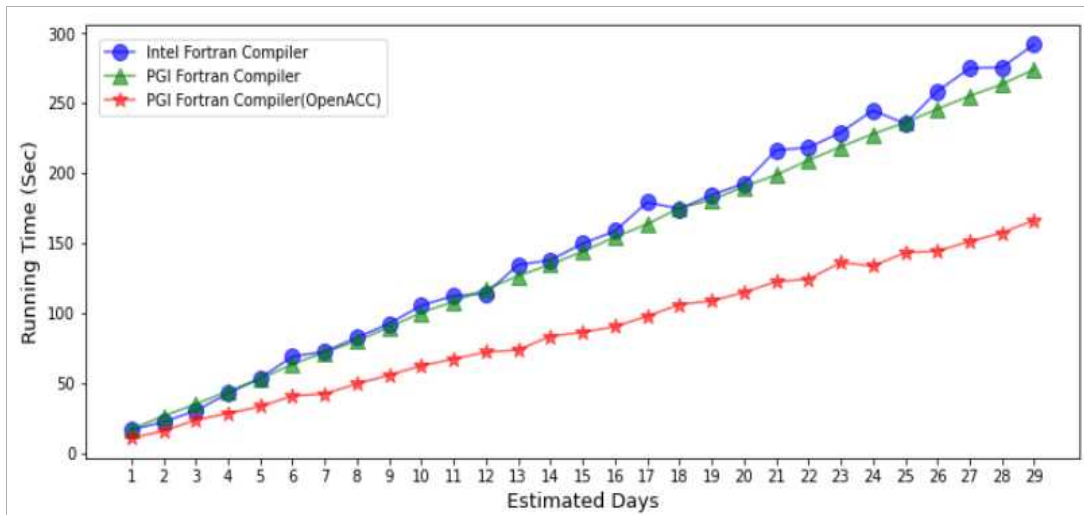


그림 8. 예측 기간에 따른 SCAM 모델 러닝 시간
Fig. 8. Running Time of SCAM Model for Estimated Days

5. 결론

본 논문에서는 SCAM의 러닝시간 향상을 끌어 내어 기상 예측 시간을 줄이기 위하여 PGI OpenACC를 이용한 SCAM 컴파일 방법을 제안 하였으며, 이를 위하여 기존에 Intel Fortran Compiler 환경에서 수행된 모델을 PGI Fortran Compiler로 모델이 수행할 수 있도록 3.2장에서 설명한 SCAM 컴파일 환경을 구축하였다. 그리고 3.3장에서 제안한 성능 향상 방법을 적용하였으며 구축된 환경에서 테스트한 결과, 기존의 Intel Fortran Compiler와 비교해서, PGI Fortran Compiler의 경우 6.08%의 성능 향상을 보였으며, PGI Fortran Compiler(OpenACC)의 경우 43.05% 성능 향상을 끌어냈다. 이는 본 논문에서 제안한 SCAM PGI Compiler with OpenACC 기법의 성능 우수성을 보여준다.

앞으로 SCAM 모델의 경우 추가적인 분석을 통해 병렬 가능한 부분을 찾아 GPGPU (OpenACC) 활용한 SCAM 병렬화를 진행할 계획이며, 이를 통해 추가적인 성능 향상을 끌어낼 계획이다. 더 나아가 해당 부분들에 대한 고도의 병렬성을 증진 시키기 위하여 OpenACC보다 더욱 세부적으로 코드를 수정할 수 있는 NVIDIA CUDA를 이용하여 병렬 최적화 작업을 진행할 계획이다. 또한 SCAM에서 진행하였던 병렬화 작업을 Glosea6 모델에도 적용할 계획에 있다.

REFERENCES

- [1] 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.sp>
- [2] 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
- [3] 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
- [4] H. Ham, et al., "Performance Assessment of Weekly Ensemble Prediction Data at Seasonal Forecast System with High Resolution," Atmosphere pp.261-276, 2017
- [5] Y. Kwon, et al., "Overview and main specifications of the Korean numerical forecast model," Proceedings of Korea Meteorological Society Conference p.177, 2019
- [6] 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, 1333-1349, p.2015
- [7] "Lee, et al., ""Performance Improvement of Cumulus Parameterization Code by Unicon Optimization Scheme"", Korea Institute of Information, Electronics, and Communication Technology, vol.15, pp.124-133, 2022
- [8] JNCAR - CESM(Community Earth System Model), Single Column Atmospheric Model (SCAM), [online] <https://www.cesm.ucar.edu/models/simpler-models/scam/index.html>
- [9] Intel oneAPI Toolkits, [online] <https://www.intel.com/content/www/us/en/developer/tools/oneapi/toolkits.html>
- [10] NVIDIA HPC SDK, [online] <https://developer.nvidia.com/hpc-sdk>
- [11] Jeonbuk National University, [online] <https://wz3.jbnu.ac.kr/cml/11846/subview.do>
- [12] Michael Wolfe, "Implementing the PGI Accelerator model," Association for Computing Machinery pp.43-50, 2010
- [13] Miles, D, et al., "Optimizing Application Performance on Cray Systems with PGI Compilers and Tools," Cray User Group pp.1-7, 2006

[14] Chris J, et al., "Offload Compiler Runtime for the Intel® Xeon Phi Coprocessor," IEEE 27th International Symposium on Parallel & Distributed Processing Workshops pp.1213-1225, 2013

[15] Junjie Chen, et al., "A Survey of Compiler Testing", ACM Computing Surveys, vol.53, pp.1-36, 2021

[16] David F, et al., "Compiler transformations for high-performance computing," ACM Computing Surveys pp.345-420, 1994

[17] Megan, et al., "The Effects of Compiler Optimizations on Materials Science and Chemistry Applications at NERSC," Optimization pp.1-8, 2017

[18] Dali, et al., "Compiler technologies for understanding legacy scientific code: A case study on an ACME land module", Procedia Computer Science, vol.108, pp.2418-2422, 2017

정 성 욱 (Sung-Wook Chung)

[중신회원]



<관심분야>

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

IoT, 스마트모빌리티, HPC, 실시간 분산 멀티미디어 시스템

저자약력

이 창 현 (Chang-Hyun Lee)

[학생회원]



<관심분야>

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

IoT, 클라우드 컴퓨팅, HPC, 실시간 분산 멀티미디어시스템

강 보 경 (Bol-Kyung Kang)

[정회원]



<관심분야>

- 2009년 8월 : 부산대학교 무역학부 경제학박사
- 2012년 3월 ~ 현재 : 창원대학교 글로벌비즈니스학부 부교수

IoT, 국제경제, 국제금융