

TECHNICAL NOTE

## 국가 대기질 예보 시스템의 모델링(기상 및 대기질) 계산속도 향상을 위한 전산환경 최적화 방안

명지수\* · 김태희 · 이용희 · 서인석 · 장임석  
국립환경과학원 대기질통합예보센터

### Optimization of the computing environment to improve the speed of the modeling (WRF and CMAQ) calculation of the National Air Quality Forecast System

Jisu Myoung\*, Taehee Kim, Yonghee Lee, Insuk Suh, Limsuk Jang

*Air Quality Forecasting Center, National Institute of Environmental Research, Incheon 22689, Korea*

#### Abstract

In this study, to investigate an optimal configuration method for the modeling system, we performed an optimization experiment by controlling the types of compilers and libraries, and the number of CPU cores because it was important to provide reliable model data very quickly for the national air quality forecast. We were made up the optimization experiment of twelve according to compilers (PGI and Intel), MPIs (mvapich-2.0, mvapich-2.2, and mpich-3.2) and NetCDF (NetCDF-3.6.3 and NetCDF-4.1.3) and performed wall clock time measurement for the WRF and CMAQ models based on the built computing resources. In the result of the experiment according to the compiler and library type, the performance of the WRF (30 min 30 s) and CMAQ (47 min 22 s) was best when the combination of Intel compiler, mvapich-2.0, and NetCDF-3.6.3 was applied. Additionally, in a result of optimization by the number of CPU cores, the WRF model was best performed with 140 cores (five calculation servers), and the CMAQ model with 120 cores (five calculation servers). While the WRF model demonstrated obvious differences depending on the number of CPU cores rather than the types of compilers and libraries, CMAQ model demonstrated the biggest differences on the combination of compilers and libraries.

**Key words** : Modeling system, Air quality forecasting, Optimization, Compiler, Message passing interface

#### 1. 서론

국가 대기질 예보제는 2013년 8월 31일 수도권 시범 예보를 시작으로 예보 권역이 점차 전국으로 확대되었으며 2014년 2월 6일부터 법정 예보를 시행하였다. 2014년 5월 22일 대기질통합예보센터가 신설된

이후 대기환경보전법 제 7조의 2(대기오염도 예측(점크기변경)발표)에 따라 현재 매일 4회(05시, 11시, 17시, 23시) 전국 19권역을 대상으로 3개의 항목( $PM_{10}$ ,  $PM_{2.5}$ ,  $O_3$ )에 대해 4단계(좋음, 보통, 나쁨, 매우 나쁨)로 예보 정보를 제공한다. 대기질 예보의 발표를 위해 예보관은 기상 및 대기질 모델의 예측 결과와

Received 7 February, 2018; Revised 4 May, 2018;

Accepted 8 June, 2018

\*Corresponding author: Jisu Myoung, Air Quality Forecasting Center, National Institute of Environmental Research, Incheon 22689, Korea  
Phone : +82-32-560-7720  
E-mail : rosie115@naver.com

The Korean Environmental Sciences Society. All rights reserved.

© This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

포함된 수치 모델의 결과를 근간으로 종합적이고 다양한 분석을 수행한다. 따라서 실시간으로 변화하는 대기의 흐름을 신속하게 파악해야하는 예보관에게 품질이 우수한 기상 및 대기질 모델 결과를 신속하게 제공하는 것은 예보 과정에 있어 매우 중요하다. 기상 및 대기질 예측 결과 생산은 광화학 반응에 의한 2차 생성·소멸 등 복합적인 대기오염현상을 과학적으로 예측할 수 있는 수치 모델 기반의 예측 시스템의 구축 및 구동을 통해 가능하다.

현재 대기질통합예보센터에서는 수치 모델을 기반으로 한 슈퍼컴퓨터 수준의 고성능 대기질 예보 모델링 시스템을 매일 4회 예보 횟수에 맞추어 운영하고 있다. 하지만 현 대기질 예보 모델링 시스템의 운영 시간은 총 약 4시간 30분으로 모델 결과의 생산·가공 및 제공까지 상당히 긴 시간이 소요된다. 대기질 예보 모델링 시스템은 크게 모델링 과정과 구동된 모델 자료의 처리 및 가공의 과정으로 나뉘지며 모델링 과정은 기상 모델(WRF, Weather Research and Forecasting), 배출량 모델(SMOKE, Sparse Matrix Operator Kernel Emissions), 대기질 모델(CMAQ, Community Multiscale Air Quality)순으로 세분화 되어 진행된다. 기상, 배출량, 대기질 모델들을 모두 구동하기 위해서는 약 3시간 40분이 소요되며 모델 구동 이후에 생성된 모델링 자료 처리 및 가공을 위해서 약 50분이 소요된다. Fig 1은 17시 예보 지원을 위한 예보 모델링 시스템 운영 일정을 간략하게 나타낸 그림이다. 모델 구동은 오전 9시에 시작되어 약 3시간 40분 동안 기상, 배출량, 대기질 모델링 과정이 진행된 후 오후 12시 40분경 완료된다. 이로부터 약 50분 뒤인 오후 13시 30분경이 되어야 예보 지원에 활용 가능한 모델링 자료의 처리 및 가공까지 완료되어 전체 대기질 예보 시스템 운영이 종료된다(Fig. 1). 따라서 신속한 모델링 결과를 제공하기 위해서는 모델들의 구동 시간 단축이 필수적이다. 특히, 대기질 예보 모델링 시스템 내에서 배출량 모델의 경우 구동 소요 시간이 20분 이내로 다소 짧은 반면, 기상 및 대기질 모델의 구동 소요 시간은 약 3시간 20분으로 전체 운영 시간의 약 70%를 차지한다. 따라서 기상 및 대기질 모델의 구동 소요 시간 단축이 전체 운영 시간 단축에 있어서 매우 중요한 인자로 판단할 수 있다.

모델의 계산속도 단축을 위해서는 각 모델의 특성을 반영한 최적화 된 모델의 구동 환경이 필요하다. 모델 구동에 영향을 미치는 요인으로는 컴파일러, 코어 수, 라이브러리, 네트워크 연결 방식 등이 언급되어져 왔다(Yang et al., 2004; Kordenbrock et al., 2006; Shainer et al., 2009; Delic, 2011; Langkamp et al., 2011; Krouse et al., 2013; Balle and Johnsen, 2016; Delic, 2017). 또한 클러스터 컴퓨팅 환경에서 스케줄러의 적용을 통해 자원을 효율적으로 사용할 수 있으며 작업 환경이 최적화 될 수 있음을 선행 논문들은 제시했다(Kim et al., 2014; Yoon et al., 2015).

모델 수행을 위한 최적의 구성은 구축 환경에 따라 차이가 발생 할 수 있으며, 같은 모델이라 할지라도 최적의 조합 환경이 상이할 수 있는데 대기질 모델인 CMAQ을 대상으로 서버 성능 및 분해능에 따른 성능을 분석하여 최적의 조합을 구성한 바 있다(Delic, 2017). 기상 모델인 WRFv3의 성능과 관련된 실험의 경우 컴파일러를 가장 중요한 요소로 꼽았으며 컴파일러에 따른 성능 비교 결과 GCC (Gnu Compiler Collection) 및 PGI (Portland Group Inc)그룹보다는 Intel 그룹의 성능이 우수하였다. 특히, GCC 보다 Intel 적용시 약 26% 향상됨을 보였다(Langkamp et al, 2011). 이는 대기질 모델 중 하나인 CAMx (Comprehensive air quality model with extensio)의 경우에도 유사하게 나타났으며 ENVIRON(2015)은 CAMx 모델의 컴파일에 따른 실험을 통해 PGI 보다 Intel 적용시에 약 20% 더 향상됨을 보였다. 컴파일러와 더불어 복잡한 수치 모델링의 계산과정에 있어서 다수의 코어를 활용하는 것은 중요한 요소이기 때문에 코어수와 라이브러리와 관계에 대한 연구가 수행된 바 있다. WRFv3 모델을 활용한 Langkamp et al.(2011)의 실험에 의하면 도메인 사이즈에 따라 필요한 라이브러리가 달라질 수 있으며 같은 조합의 라이브러리를 적용하더라도 코어 수에 따라 성능이 달라질 수 있음을 제시하였다. Yang et al.(2004)은 WRF 모델 수행 시 NetCDF (Network common data form)와 Parallel HDF5를 사용하여 코어 수에 따른 성능분석을 실시한 결과 코어 수의 증가와 성능이 비례하지 않음을 밝혀낸바 있다. 따라서 성능 향상을 위한 최적의 조합을 결정하기 위해서는 최적의 라이브러리

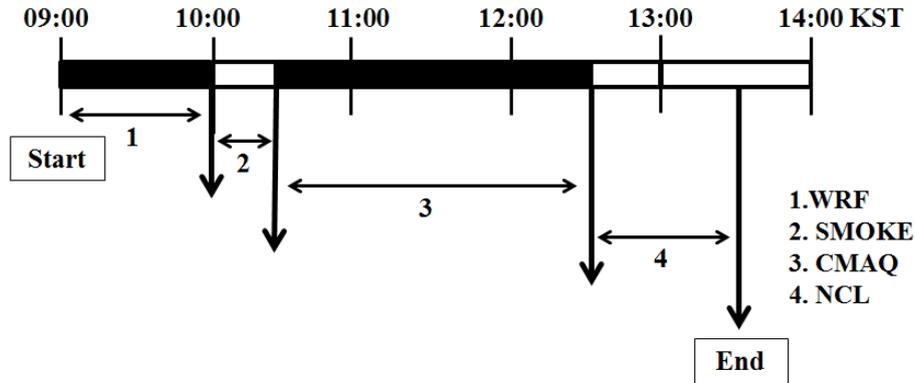


Fig. 1. Time schedule of the national air quality forecast system.

조합을 선택하기 위한 실험과 코어 수에 따른 실험 모 두가 고려되는 것이 바람직하다.

Shainer et al.(2009)은 WRF 모델 수행 시 MPI (Message Passing Interface) 종류(MVAPICH, Open MPI)별 성능 분석 결과 종류에 따른 성능의 차이는 없 음을 밝혔다. 하지만 서버 네트워크 연결 방식 (InfiniBand, GigE)에 따른 성능 분석을 실시한 결과 InfiniBand가 GigE보다 최소 2%(2노드), 최대 115%(23노드)의 성능향상이 있음을 밝혔다. 이는 Cho et al.(2013)의 결과에서도 동일하게 나타났으며 GigE와 InfiniBand에 따른 성능 분석을 실시한 결과 GigE 사용시에 비해 InfiniBand 사용시 25% 향상시 키는 것을 밝혔다.

이와 같은 선행 연구들은 모델 수행을 위한 최상의 구성은 구축 환경에 따라 차이가 발생 할 수 있으며, 같은 모델이라 할지라도 최적의 조합이 상이할 수 있 다는 것을 시사한다. 컴퓨팅 구축 환경에 의한 요인들 이 미치는 영향은 모델 고유의 특성에 따라 차이가 발 생할 수 있으므로 각각의 모델별 특성을 고려한 민감도 실험이 요구 된다. 따라서 현재 사용 중인 시스템에 기상 및 대기질 모델 수행을 위한 최적의 조합을 독자 적으로 구축해야할 필요가 있다. 본 연구에서는 기 구축된 시스템 환경을 기반으로 컴파일러, 라이브러리와 코어 수에 따른 최적화 테스트를 수행하여 사용 자 원 대비 수행 시간을 단축시킬 수 있는 최적의 조합을 도출하고자 한다. 이러한 최적화 된 대기질 예보 모델 링 환경 구축은 모델 자료 생산 가공 시간의 단축 및

효율적인 예보 자료 생산에 기여할 것으로 판단되며 예보관들은 모델결과를 분석하는데 충분한 시간을 확보하여 결과적으로 예보 정확도 향상에 크게 기여할 것으로 기대된다.

## 2. 재료 및 방법

### 2.1. 모델링 시스템 구성 및 환경

현업 예보 업무 지원을 위한 대기질 예보 모델링 시 스템은 크게 기상, 배출량, 대기질의 세부 모델로 구성 되어 있으며, 매일 4회 예보 시각에 맞춰 운영되고 있 다(Fig. 2). 매 1회 모델의 수행시 예보에 활용되기 위 해 요구되는 기상 및 대기질 모델의 구동 시간은 3시 간 차이가 존재하지만 총 구동 기간은 3.5일로 동일하 다(Fig. 3). 각 대기질 예보 모델의 버전 정보 및 상세 모의 환경은 Table(기존 소문자를 대문자로 변경하였 습니다. 문장내 사용시 대/소문자 사용에 대한 규정 확 인을 부탁드립니다.) 1과 같다. Fig. 4는 모델의 모의 영역을 나타낸 것으로 동아시아(D1, 27 km), 한반도 (D2, 9 km), 수도권(D3, 3 km)으로 구분되어 있으며, 각각의 도메인에 대한 격자 정보는 Table(기존 소문자 를 대문자로 변경하였습니다. 문장내 사용시 대/소문 자 사용에 대한 규정 확인을 부탁드립니다.) 2와 같다.

대기질 예보 모델링 시스템의 모델링 순서는 기상-배출량-대기질 모델 순이다. 기상 모델링을 통하여 생 산된 결과 자료는 배출량 모델의 입력 자료로 사용되 고, 배출량 모델링을 통하여 생산된 자료와 함께 대기질

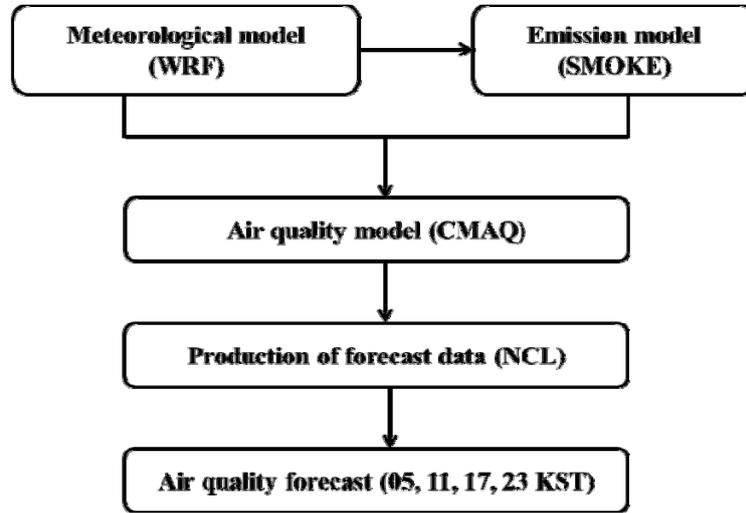


Fig. 2. Schematic diagram of the national air quality forecast system.

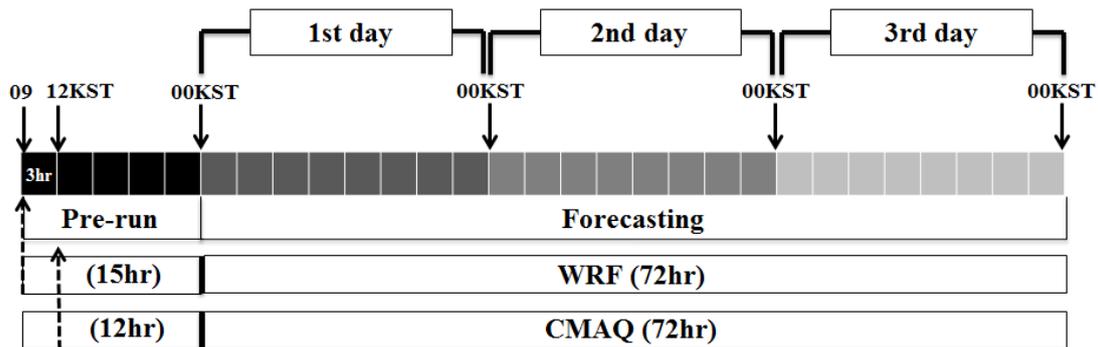


Fig. 3. Run time of WRF and CMAQ model.

모델의 입력 자료로 사용된다. 이렇듯 생산되는 모델 자료들은 긴밀하게 연결되어 있으며, 자료의 크기 또한 방대하기 때문에 대기질 예보 모델링 시스템은 안정적인 운영과 신속한 모델 자료 생산이 중요하다. 이와 같은 특수성을 고려하여 대기질 예보 모델링 시스템을 위한 전산 자원 및 서버 환경이 마련되었다. 안정적인 시스템 운영을 위해 스케줄러를 활용한 자동화 운영을 도입하였고 방대한 양의 자료를 효율적으로 처리하기 위하여 초고속 네트워크로 연결된 슈퍼컴퓨터에 상용하는 고성능 컴퓨팅 시스템을 구축하였다.

스케줄러는 작업을 위한 컴퓨팅 자원의 분배를 위해 작업 일정 또는 순서를 관리하는 프로그램으로 구

칙적인 작업을 보유한 자원에 효율적으로 분배가 가능하도록 설계되어 있기 때문에 이를 통한 자동화 운영은 매일 4회 수행되는 예보 모델링 작업에 필수적이다. 스케줄러의 종류로는 TORQUE (Terascale Open-source Resource & Queue manager), LSF (Load Sharing Facility), PBS (Portable Batch System), SGE (Sun Grid Engine) 등 다양한 솔루션들이 있다(Etsion and Tsafir, 2005; Yoon et al., 2017). 대기질통합예보센터에서는 계산 작업 환경을 안정적으로 구성하기 위해 본 작업 수행 환경에 적합하다고 판단이 되는 PBS Pro (Portable Batch System Professional) 스케줄러를 도입하였다. PBS Pro 스케줄러는 소규모

**Table 1.** Details of the WRF/CMAQ model configuration

Model		Configuration	
WRF	Version	v3.3	
	Scheme	PBL	YSU (Hong et al., 2006)
		Micro Physics	WSM3 (Hong et al., 2004)
		Cumulus	Kain-Fritsch (Kain et al., 2004)
		Long Wave Radiation	RRTM (Mlawer et al., 1997)
		Short Wave Radiation	Goddard (Chou et al., 1994)
SMOKE	Version	v3.1	
	Emission	Nature	Megan v2.04
		Antropogenic	MICS-Asia 2010
			CAPSS 2010
CMAQ	Version	v4.7.1	
	Option	Chemical mechanism	SAPRC99 (Carter et al., 1990)
		Aerosol Module	AERO5 (Binkowski et al., 2003)
		Horizontal Diffusion	Multiscale (Odman et al., 1997)
		Vertical Diffusion	Eddy (Sutton et al., 1932)
		Advection Schem	Yamo (Odman et al., 2000)

**Table 2.** Details of the grids and nested model domains

Domains	Horizontal grid	Horizontal resolution
D1	174 × 128	27 km
D2	67 × 82	9 km
D3	58 × 61	3 km

클러스터부터 복잡한 대규모의 시스템까지 검증된 솔루션으로 대규모의 컴퓨터 시스템 환경을 지원 가능한 작업 스케줄러이자 워크로드 관리 툴로 알려져 있다. 이러한 PBS Pro 스케줄러의 장점은 고성능 컴퓨팅 시스템에서 하루 4회 수행되는 예보 모델링 수행의 안정성 및 효율성을 증대 시킬 수 있다.

실제 스케줄러 적용시 이를 통한 작업의 모니터링이 가능해 지므로 각각의 모델 수행시작 및 정상적인 종료에 대한 성공률을 향상 시킬 수 있었으며, 어떤 계산 서버에서 어떠한 작업이 수행될지 지정하는 과정 없이 여러 작업을 제출할 수 있게 해주어 효율적인 서버 자원 관리를 가능하게 해주었다. 이와 같은 스케줄러를 통한 안정적인 모델 운영과 함께 모델 수행 시간의 단축을 위해서 모델의 구동 특성을 고려한 서버 환경을 구축하였다(Table 3).

수치모델의 계산을 위한 병렬 처리 작업시 하나 또는 다수의 파일을 분할하고 다시 합치는 과정이 필요한데 이때, 일반 파일 전송량보다 많은 통신 정보들이 만들어진다. 이러한 문제점을 해결해 줄 수 있는 방안으로 여러 개의 파일을 동시에 처리하는데 효율적인 병렬 파일 시스템이 사용되는데 이는 여러 서버들로부터 동시다발적인 읽기와 쓰기를 가능하게 해준다. 병렬 파일 시스템의 종류에는 Lustre, GPFS (General Parallel File System) 등이 있다(Schmuck et al., 2002; Yoon et al., 2016). GPFS는 고성능 공유 디스크 파일 시스템으로 모든 클러스터내의 노드들은 단일 파일에 대한 동시적 읽기 쓰기가 가능하며 파일들이 여러 디스크에 걸쳐 다수의 시스템을 통하여 작동되기 때문에 고성능 시스템인 예보 모델링 시스템에 적합하다. 따라서 본 시스템에서는 생산되는 자료에 대한 신속한

I/O를 제공하고 안정적이고 효율적인 관리가 가능한 IBM의 GPFS를 구축하였다.

네트워크 연결방식은 병렬처리를 통한 대용량 자료를 빠르게 전송시키기에 적합한 Infiniband 네트워크로 구성하였다. Cho et al.(2013)에 의하면 사용되는 네트워크 디바이스에 따라서 요구되는 프로세서 자원이 달라질 수 있음을 언급한바 있는데 최적의 코어에 대해서 10GigE는 약 60%인 반면, InfiniBand 는 약 75%였다. 기 구축된 이더넷방식 네트워크 보다 우수한 성능을 보이는 InfiniBand 방식은 적은 노드의 사용시에는 기존 이더넷 방식과 큰 차이가 없지만 노드의 수가 증가됨에 따라 성능향상이 기대되어진다. 특히 WRF 및 MM5 (Mesoscale Model 5) 모델의 경우, DDR (Double Data Rate) Infiniband는 4개의 노드 사용시 10GigE와 성능에 큰 차이가 없었지만 24개의 노드 사용시 각각 113%, 30%의 성능 향상이 있었다 (HPC Advisory Council, 2009). Infiniband 시스템으로 구성된 분산 메모리 형태의 클러스터 구조에서 병렬 계산을 수행하기 위해서는 Infiniband 시스템을 대상으로 개발된 MPI 라이브러리가 필요한데 MPI 병렬 계산을 위한 라이브러리의 종류로는 MVAPICH, Open MPI 등이 존재한다(Jeon et al., 2014). 따라서 본 시스템에서는 모델링 수행을 위해 GPFS 파일 시스템과 더불어 InfiniBand 네트워크를 구축하였고 InfiniBand 기반의 MPI 라이브러리들을 사용하였다.

CPU (Central Processing Unit)의 생산 세대에 따른 성능은 벤치마킹 자료를 기반으로 WRF 모델의 경우 14 세대 적용시 이전 세대에 비해 대략 30~50% 성능 향상이 있다고 한다(Dell EMC, 2017). 하지만 이러한

성능향상은 단일 CPU간의 단순 비교로 의미를 가질 뿐 실제 모델의 수행 성능을 판단하기에는 어려움이 있다. 이는 모델 수행 성능이 CPU, 메모리, 네트워크, 파일시스템 및 컴파일러, 라이브러리 종류 등에 복합적으로 영향을 받기 때문으로 코어가 많을수록 동일한 계산량에서 더 많은 통신이 필요하므로 네트워크 제한으로 인해 잠재적인 성능 저하가 있을 수 있다 (Langkamp et al., 2011). 따라서 CPU가 향상되더라도 그에 비례하는 성능향상을 보이지 못할 경우가 있을 수 있다.

## 2.2. 실험 설계

전체 모델링 서버의 자원은 약 760개의 CPU (Central Processing Unit) 코어와 2.7 PiB (Pebibyte)의 저장용량, 20 Tflops (Tera floating-point operations) 성능의 계산 서버 30대로 이루어져 있지만 본 연구에서는 기존 계산 서버 대비 더 향상된 사양의 CPU와 메모리, 상위 버전의 OS (Operating System)로 구성된 신규 계산 서버 10대(CPU 코어 수 : 280개)를 활용하여 최적의 라이브러리를 도출하기 위한 민감도 실험을 설계 하였다. InfiniBand는 FDR (Fourteen Data Rate) 타입으로 56 Gbit/s의 데이터 전송률을 가지며 계산 서버와 병렬파일시스템의 I/O로 사용하고 있다 (Table 3). 민감도 실험은 기상 모델(WRF)과 대기질 모델(CMAQ)에 대해서만 진행하였으며, 모델의 연산 범위는 현재 예보를 위해 필요한 모델의 모의 영역 (Fig. 4)에 대해 수행하였다. 수치모델의 최적화를 통한 모의 시간 단축 방안으로 서버 구축 환경을 고려하여 2가지 실험을 수행하였다. 첫 번째 실험은 최적화에 중요한 요소인 컴파일러와 라이브러리들의 선택이다. Table 4에 제시된 컴파일러별 MPI와 NetCDF 라이브러리의 종류에 따라 12가지 조합에 대한 최적화 실험을 구성하였다. 12가지의 실험 조합 구성은 모델 계산 및 속도에 가장 중요한 컴파일러를 구분하여 설계하였고, 두 컴파일러에 따른 라이브러리의 버전별 세부 구성을 통해 세부 조합을 완성하였다. 컴파일러 종류는 2가지로 PGI (ver.14.9)와 Intel (ver.15.0) 컴파일러를 사용하였다. 라이브러리는 모델의 병렬 처리에 중요한 MPI와 파일의 입출력 부분에 중요한 NetCDF의 구성에 따라 조합을 세분화하였다. MPI는

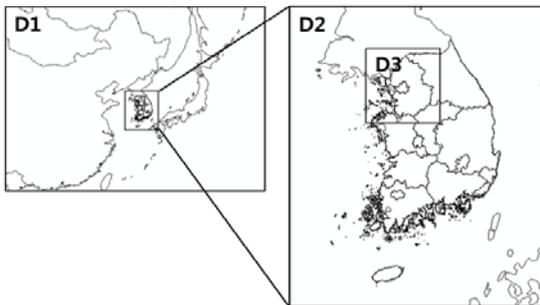


Fig. 4. The nested model domains for 27(D1), 9(D2), 3(D3) km.

**Table 3.** Air Quality forecasting model system experimental environment

Classification	Contents
File system	GPFS 3.5.0
OS	CentOS 6.7 (64bit)
CPU	Intel Xeon E5-2690 v4 14C 2.6 GHz 35 MB Cache * 2 ea / node
Infiniband Driver	MLNX_OFED_LINUX-2.2-1.0.1.1
Disk	Usable : 2.7 PiB
Memory	128 GB / node

**Table 4.** Designs of compiler, MPI, NetCDF for optimization experiment

No.	Compiler	MPI	NetCDF
1	PGI-v14.9	mvapich-2.0	NetCDF-3.6.3
2		mvapich-2.0	NetCDF-4.1.3
3		mvapich-2.2	NetCDF-3.6.3
4		mvapich-2.2	NetCDF-4.1.3
5		mpich-3.2	NetCDF-3.6.3
6		mpich-3.2	NetCDF-4.1.3
7	Intel-v15.0	mvapich-2.0	NetCDF-3.6.3
8		mvapich-2.0	NetCDF-4.1.3
9		mvapich-2.2	NetCDF-3.6.3
10		mvapich-2.2	NetCDF-4.1.3
11		mpich-3.2	NetCDF-3.6.3
12		mpich-3.2	NetCDF-4.1.3

**Table 5.** Designs of core(node) for WRF model optimization experiment

Model	Experiment	Core	Node
WRF	WC1	56	2
	WC2	84	3
	WC3	112	4
	WC4	140	5
	WC5	168	6
	WC6	196	7
	WC7	224	8
	WC8	252	9

현재 사용 중인 PBS와 호환이 가능한 버전 3가지 (mvapich-2.0, mvapich-2.2, mpich-3.2)를 사용하였다. 3가지 MPI 종류 중 mpich-3.2는 mvapich-2.0, mvapich-2.2와 달리 InfiniBand 네트워크 방식을 반영하지 못한 라이브러이지만 본 연구에서는

InfiniBand 네트워크 방식의 효과를 검증하기 위하여 mpich-3.2를 실험의 조합으로 추가하였다. 또한 NetCDF의 경우 모델 및 MPI 버전에 사용 가능한 2가지(NetCDF-3.6.3, NetCDF-4.1.3) 버전을 사용하였다. 12가지의 실험 수행에 대한 결과 평가는 서버에서

**Table 6.** Designs of core(node) for CMAQ model optimization experiment

Model	Experiment	Nx	Ny	Core	Node
CMAQ	CC1		8	48	2
	CC2	6	12	72	3
	CC3		16	96	4
	CC4		8	64	3
	CC5	8	12	96	4
	CC6		16	128	5
	CC7		12	120	5
	CC8	10	16	160	6
	CC9		12	144	6
	CC10	12	16	192	7

측정된 작업수행시간(wall clock time)으로 평가하였다. 작업수행시간은 해당 된 작업이 실제로 수행하는데 소요되는 시간이다. 모델의 모의 영역 및 예보 기간에 대하여 WRF 모델의 경우 real.exe와 wrf.exe 과정에 대해, CMAQ 모델은 BCON 및 CCTM 과정에 대해 작업수행시간을 평가 하였다.

두 번째 실험은 첫 번째 실험 결과를 통해 결정된 최적화된 컴파일러 및 라이브러리의 조합을 이용하여 WRF, CMAQ 모델의 CPU 코어 수 증가에 따른 최적화를 수행하였다. 1대의 계산 서버가 보유하고 있는 CPU 코어 수는 28개 이므로 WRF 모델은 2~9대의 계산 서버를 활용하여 56~252개의 코어 수로 실험을 진행하였고(Table 5), CMAQ은 분해능에 따라 CPU 코어 수를 결정한 뒤 48~192개까지 CPU 코어를 증가시켜 2~7대 사이의 계산 서버를 활용하여 테스트를 진행하였다(Table 6). 최적화 여부는 컴파일러 및 라이브러리 조합 후 모델이 구동하는데 소요된 시간이 가장 짧은 구성을 최적화 구성으로 판단하였다.

### 3. 결과 및 고찰

#### 3.1. 컴파일러 및 라이브러리 종류에 따른 실험 결과

컴파일러 및 두 종류의 라이브러리들에 따라 12개의 실험을 진행하였다. 모델 구동에 있어 가장 중요한 컴파일러 선택을 위해 PGI (ver.14.9)와 Intel (ver.15.0) 두 컴파일러에 대하여 WRF와 CMAQ 모델을 각각 적용한 실험을 우선적으로 진행하였다.

WRF 모델의 컴파일러에 따른 민감도 실험 결과 Intel 그룹군이 PGI 그룹군보다 더 좋은 성능을 보였다. Intel 그룹군은 평균 약 32분이 소요되었고, PGI의 경우는 약 36분으로 Intel 컴파일러 사용시 약 4분 정도의 시간이 단축되었다. 라이브러리는 MPI의 경우 mpich-3.2 버전보다 mvapich-2.0의 조합이 더 좋은 성능을 나타냈고, 이는 Intel과 PGI 컴파일러 모두 동일하게 나타났다. 컴파일러와 MPI 라이브러리의 조합이 고정되어 있다는 조건하에서는 NetCDF-4.1.3 버전보다 NetCDF-3.6.3이 더 좋은 성능을 나타냈다. 이를 종합해 볼 때 컴파일러 및 라이브러리 종류에 따른 WRF 모델의 실험결과, WL7번 실험의 조합(Intel compiler/mvapich-2.0/NetCDF-3.6.3)의 소요 시간이 약 30분으로 전체 실험 중 가장 좋은 결과를 나타냈다 (Fig.5, Table 7).

CMAQ 모델의 실험결과 컴파일러의 경우 WRF 모델의 결과와 동일하게 Intel 그룹군이 PGI 그룹군보다 더 좋은 성능을 보였다. Intel 그룹군은 평균 약 97분, PGI의 경우는 약 105분으로 Intel 컴파일러를 사용했을 때 PGI 컴파일러보다 약 8분 정도의 시간이 단축되는 효과를 보였다. 따라서 PGI보다 Intel 컴파일러를 사용했을 경우에 더 향상된 성능을 기대할 수 있다. 동일한 컴파일러 군에서 NetCDF 버전에 따른 속도 차이가 상당히 나타났는데 특히 NetCDF-4.1.3버전보다 NetCDF-3.6.3은 약 2배 더 좋은 성능을 보였다. 컴파일러와 NetCDF 라이브러리의 조합이 동일하게 구성 되어있다는 조건하에서는 작업수행시간이 약 150

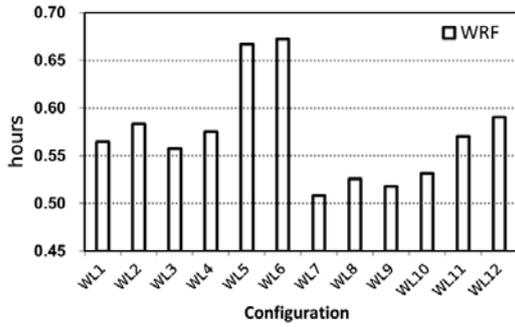


Fig. 5. Results of according to WRF model configuration (WL1~WL12).

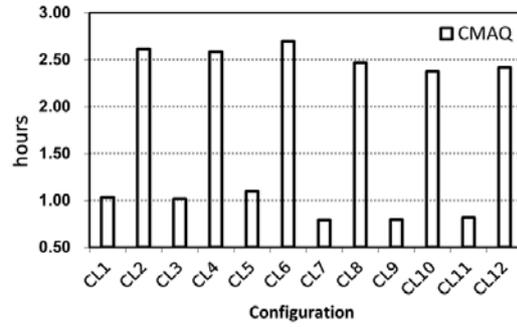


Fig. 6. Results of according to CMAQ model configuration (CL1~CL12).

Table 7. Results of WRF model according to compiler and library type

No.	Configuration	Wall clock time
		Hour (hh: mm: ss)
WL1	PGI compiler / mvapich-2.0 / NetCDF-3.6.3	0.565 (00:33:54)
WL2	PGI compiler / mvapich-2.0 / NetCDF-4.1.3	0.583 (00:35:00)
WL3	PGI compiler / mvapich-2.2 / NetCDF-3.6.3	0.558 (00:33:27)
WL4	PGI compiler / mvapich-2.2 / NetCDF-4.1.3	0.575 (00:34:31)
WL5	PGI compiler / mpich-3.2 / NetCDF-3.6.3	0.667 (00:40:01)
WL6	PGI compiler / mpich-3.2 / NetCDF-4.1.3	0.673 (00:40:21)
WL7	Intel compiler / mvapich-2.0 / NetCDF-3.6.3	0.508 (00:30:30)
WL8	Intel compiler / mvapich-2.0 / NetCDF-4.1.3	0.526 (00:31:34)
WL9	Intel compiler / mvapich-2.2 / NetCDF-3.6.3	0.518 (00:31:05)
WL10	Intel compiler / mvapich-2.2 / NetCDF-4.1.3	0.531 (00:31:53)
WL11	Intel compiler / mpich-3.2 / NetCDF-3.6.3	0.571 (00:34:14)
WL12	Intel compiler / mpich-3.2 / NetCDF-4.1.3	0.591 (00:35:26)

Table 8. Results of CMAQ model according to compiler and library type

No.	Configuration	Wall clock time
		Hour (hh: mm: ss)
CL1	PGI compiler / mvapich-2.0 / NetCDF-3.6.3	1.033 (01:02:00)
CL2	PGI compiler / mvapich-2.0 / NetCDF-4.1.3	2.610 (02:36:36)
CL3	PGI compiler / mvapich-2.2 / NetCDF-3.6.3	1.019 (01:01:09)
CL4	PGI compiler / mvapich-2.2 / NetCDF-4.1.3	2.584 (02:35:03)
CL5	PGI compiler / mpich-3.2 / NetCDF-3.6.3	1.096 (01:05:44)
CL6	PGI compiler / mpich-3.2 / NetCDF-4.1.3	2.693 (02:41:34)
CL7	Intel compiler / mvapich-2.0 / NetCDF-3.6.3	0.789 (00:47:22)
CL8	Intel compiler / mvapich-2.0 / NetCDF-4.1.3	2.466 (02:27:59)
CL9	Intel compiler / mvapich-2.2 / NetCDF-3.6.3	0.796 (00:47:45)
CL10	Intel compiler / mvapich-2.2 / NetCDF-4.1.3	2.372 (02:22:18)
CL11	Intel compiler / mpich-3.2 / NetCDF-3.6.3	0.818 (00:49:04)
CL12	Intel compiler / mpich-3.2 / NetCDF-4.1.3	2.418 (02:25:06)

분으로 MPI 라이브러리 버전에 상관없이 유사하게 나타났으며 이는 Intel과 PGI 컴파일러 모두 동일하게 나타났다. 따라서 CMAQ 실험결과 컴파일러와 NetCDF의 선택이 중요함을 알 수 있다. 총 12개의 조합 중에서 CL7번 실험(Intel complier/mvapich-2.0/NetCDF-3.6.3)이 약 47분의 소요 시간으로 가장 좋은 결과를 보였다(Fig. 6, Table 8).

### 3.2. 코어수에 따른 실험결과

WRF 모델에 대하여 컴파일러 및 라이브러리 종류에 따른 실험결과 가장 우수한 성능을 보인 WL7 (Intel complier, mvapich-2.0, NetCDF-3.6.3) 조합을 바탕으로 CPU 코어 수 테스트를 위해 WRF 모델의 CPU 코어 수를 56~252개 까지 증가시켜 총 8개의 실험(WC1~WC8)을 수행하였다. 1대의 계산 서버는 28개의 CPU 코어로 구성되어 있어 코어 할당시 서버 전체 코어를 모두 사용하여 실험을 진행하였다. 전체 실험 중에서 56개의 CPU 코어를 사용했을 때 가장 긴 작업수행시간을 보였는데 CPU 코어 수가 증가함에 따라 성능이 향상되었지만 168개부터 하락하였다. 이는 코어 수가 많아짐에 따라 계산을 위해 자료를 분배하고 취합하는데 더 많은 시간이 더 소요되었기 때문으로 판단된다. CPU 코어수가 224개를 넘어서자 작업수행시간이 더 이상 증감을 보이지 않는 상태가 되었다. 전체 실험의 평균 작업수행시간은 25분 48초이며 평균 이상의 성능을 보인 실험은 WC2, WC3, WC4로 나타났다(Fig. 7, Table 9). 이중 가장 좋은 성능을 보인 실험은 WC4로 계산 서버 5대, 140개의 코어를 사용하였을 때 가장 뛰어난 성능을 보임을 알 수 있다. 단순히 모델의 구동 소요 시간 측면에서는 WC4번 실험이 가장 우수하지만, CPU 코어 수 사용대비 소요 시간 측면을 함께 고려하면 WC2~WC4 경우 모두 우수하다고 볼 수 있다. 따라서 구동 시간 외에 CPU 코어 수 까지 고려하면 현업 운영상 가장 효율적인 최적 구성은 84개 코어를 사용할 때로 판단 할 수 있다. WRF 실험을 종합한 결과, Intel complier, mvapich-2.0, NetCDF-3.6.3의 라이브러리 조합 및 3대의 계산 서버를 사용한 모델링 구동이 현업 운영 조건 하에서 최적의 조합임을 알 수 있다.

WRF 모델에 대한 CPU 코어 수 실험에 이어 CMAQ

모델에 대해서도 실험을 별도로 진행하였다. 컴파일러 및 라이브러리 종류에 따른 실험결과 WRF 모델과 마찬가지로 가장 우수한 성능을 보인 CL7 (Intel complier, mvapich-2.0, NetCDF-3.6.3) 조합을 바탕으로 CPU 코어 수에 따른 테스트를 코어 수를 48~192개까지 증가시켜 총 10개의 실험(CC1~CC10)을 수행하였다. CMAQ 실험의 경우 WRF와 달리 CPU 코어 수 구성시 x와 y방향의 분해능의 조합을 고려하여 실험을 설계하였다. 각각의 실험군은 Nx와 Ny의 조합에 의해 전체 코어수가 Nx와 Ny의 곱이 될 수 있도록 설계하였는데 이때 Nx를 기준으로 Ny를 증가시키며 실험을 수행하였다. 실험 결과 전체 실험 중에서 가장 많은 192개 코어를 사용한 실험이 47분 46초로 가장 긴 작업수행시간을 보였다. 고정된 Nx에서는 코어수가 증가함에 따라 성능 하락을 보였지만 Nx가 6으로 고정된 CC1~CC3의 실험의 경우 코어수가 증가함에 따라 향상된 성능을 보였다. 전체 실험의 평균 작업수행시간은 46분 19초이며 평균이상의 성능을 보인 실험은 CC3~CC8로 이중 가장 좋은 성능을 보인 실험은 CC7실험이다. CC7 실험은 Nx가 10이고 Ny가 12인 분해능일 때 약 45분 29초의 작업수행시간이 소요 되었다(Fig. 8, Table 10). CPU 코어 수에 따른 CMAQ 모델의 결과, WRF 모델 실험결과 비교해 보았을 때 CPU 코어 수에 따른 작업수행시간의 민감도 결과는 거의 유사하였다. 따라서 자원 사용 대비 효율성을 고려해 볼 때, CC3~CC8의 실험에서 소요된 구동 시간이 큰 차이가 없으므로 Intel complier, mvapich-2.0, NetCDF-3.6.3 조합 및 CPU 코어 수를 64개(Nx, Ny 각각 8개)로 사용할 때 가장 효율적임을 알 수 있다.

## 4. 결 론

본 연구에서는 국가 대기질 예보에 필수적인 모델 자료의 신속하고 안정적인 제공을 위하여 현업 운영 중인 모델링 시스템의 최적화 구성 방안으로 컴파일러, 라이브러리, CPU 코어 수에 따른 최적화 실험을 수행하였다. 전산 시스템 환경은 모델의 수치 계산을 위한 병렬 처리에 유용한 GPFS 시스템 및 Infiniband 네트워크를 사용하였고, PBS Pro 스케줄러를 도입

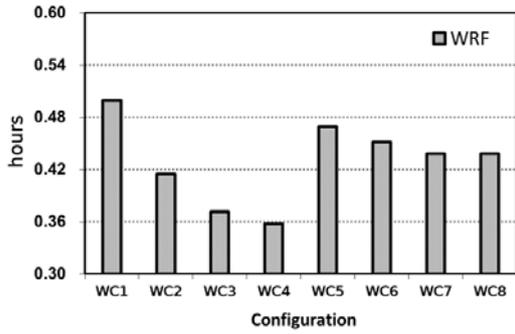


Fig. 7. Results of according to WRF model configuration (WC1~WC8).

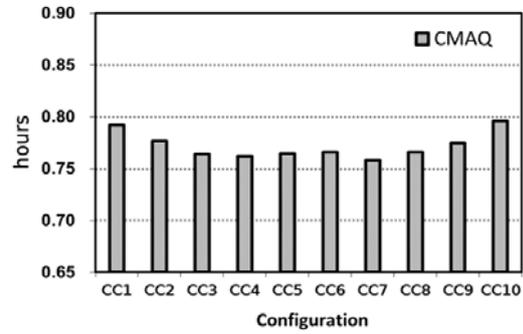


Fig. 8. Results of according to CMAQ model configuration (CC1~CC10).

Table 9. Results of WRF model according to core(node) (based by WL7)

Ex.	Node	Core	Wall clock time
			Hour (hh: mm: ss)
WC1	2	56	0.499 (00:29:57)
WC2	3	84	0.415 (00:24:53)
WC3	4	112	0.371 (00:22:17)
WC4	5	140	0.358 (00:21:28)
WC5	6	168	0.469 (00:28:09)
WC6	7	196	0.452 (00:27:06)
WC7	8	224	0.438 (00:26:17)
WC8	9	252	0.438 (00:26:18)

Table 10. Results of CMAQ model according to core(node) (based by WL7)

Ex.	Nx	Ny	Node	Core	Wall clock time
					Hour (hh: mm: ss)
CC1		8	2	48	0.792 (00:47:31)
CC2	6	12	3	72	0.777 (00:46:36)
CC3		16	4	96	0.764 (00:45:50)
CC4		8	3	64	0.762 (00:45:43)
CC5	8	12	4	96	0.764 (00:45:52)
CC6		16	5	128	0.766 (00:45:57)
CC7		12	5	120	0.758 (00:45:29)
CC8	10	16	6	160	0.766 (00:45:58)
CC9		12	6	144	0.775 (00:46:29)
CC10	12	16	7	192	0.796 (00:47:46)

하였다. 스케줄러 사용시 작업의 모니터링이 가능해 지므로 각각의 모델 수행시작 및 정상적인 종료에 대

한 성공률을 향상 시킬 수 있었으며, 어떤 계산 서버에서 어떠한 작업이 수행될지 지정하는 과정 없이 여러

작업을 제출할 수 있게 해주어 효율적인 서버 자원 관리를 가능하게 해주었다. 구축된 전산을 기반으로 WRF와 CMAQ 모델에 대해 컴파일러(PGI, Intel), MPI (mvapich-2.0, mvapich-2.2, mpich-3.2), NetCDF (NetCDF-3.6.3, NetCDF-4.1.3) 종류에 따른 12가지 조합을 구성하여 작업수행시간 측정을 통한 최적화 실험을 수행하였다. 컴파일러 및 라이브러리 종류에 따른 실험 결과, WRF 및 CMAQ 모델은 12개의 조합 중에서 Intel compiler, mpich-2.0, NetCDF-3.6.3 조합을 적용한 경우 WRF는 30분 30초, CMAQ은 47분 22초로 가장 우수한 성능을 보였다. 또한 CPU 코어 수에 따른 최적화 결과, WRF 모델은 140코어(계산 서버 5대)에서, CMAQ 모델은 120코어(계산 서버 5대)에서 가장 좋은 결과를 보였다. 그러나 사용된 CPU 코어 개수와 구동 시간을 종합적으로 고려할 때, WRF 모델의 WC2~WC4 실험에서 소요된 각 구동 시간은 큰 차이가 없지만 WC2의 실험에 사용된 CPU 코어 수가 84개로 가장 적었다. 또한 CMAQ 모델은 CC3~CC8 실험에서도 소요된 구동 시간은 큰 차이가 없었지만 CPU 코어 수를  $N_x$ ,  $N_y$  각각 8개로 구성하여 64개를 사용한 CC4의 경우가 코어수가 가장 적었다. 따라서 자원을 최소한으로 사용한 경우 모델 수행을 위한 최적의 CPU 코어 및 서버 수는 WRF 84개(계산 서버 3대), CMAQ 64개(계산 서버 3대)로 판단할 수 있다. 위의 다양한 민감도 실험 연구를 종합해 보면 예보 모델의 구동 성능은 각 모델의 특성에 따라 구성 환경에 영향을 받을 수 있다. WRF 모델의 경우 컴파일러 및 라이브러리의 조합 실험에 의한 차이보다는 CPU 코어 수에 따른 차이에 의한 변화가 컸던 반면, CMAQ 모델의 경우에는 컴파일러 및 라이브러리의 조합에 의한 변화가 가장 크게 나타났다.

본 연구에서 수행된 최적화 결과를 도입한 결과 기존 시스템의 모델링 운영 시간 대비 최적화된 모델링 과정의 소요 시간은 약 30%가 단축되었으며, 구동된 모델 자료의 처리 및 가공의 과정까지 포함한 전체 시스템의 소요 시간은 기존 대비 약 45%가 감소하였다. 본 연구가 현재 구축되어진 전산 환경 및 가용 가능한 컴파일러 및 라이브러리의 버전에 특화된 최적의 조합이지만 이를 통해 모델의 실행 환경 구성으로 얻어

진 모델 수행 시간 단축은 예보관들에게 충분한 모델 결과 분석 시간을 제공하여 예보의 정확도 향상에 기여할 것으로 사료 된다. 향후 본 연구의 구성과 다른 전산 환경에서 예보 모델 구동시 운영 모델별 최적화 실험이 추가로 요구되지만 본 연구 결과를 활용하면 적은 시간과 노력으로 좋은 결과를 빠르게 얻을 것으로 사료된다.

### 감사의 글

본 논문은 국립환경과학원 대기질통합예보센터의 "대기질 현업예보 모델링 시스템 증설 사업"(1946-303-430)의 지원으로 수행되었습니다.

### REFERENCES

- Balle, T., Johnsen, P., 2016, Improving I/O performance of the Weather Research and Forecast (WRF) model, Cray user group, [https://cug.org/proceedings/cug2016\\_proceedings/includes/files/pap123s2-file2.pdf](https://cug.org/proceedings/cug2016_proceedings/includes/files/pap123s2-file2.pdf).
- Binkowski, F. S., Roselle, S. J., 2003, Models-3 Community Multiscale Air Quality (CMAQ) model aerosol component, 1, Model description, *J. Geophys. Res.*, 108, 4183.
- Carter, W. P. L., 1990, A Detailed mechanism for the gas-phases atmospheric reactions of organic compounds. *Atmos. Environ.*, 24, 481-518.
- Cho, J. Y., Jin, H. W., 2013, An Optimization tool for determining processor affinity of networking processes, KIPS tran softw data eng, <http://dx.doi.org/10.3745/KTSDE.2013.2.2.131>.
- Chou, M. D., Suarez, M. J., 1994, An Efficient thermal infrared radiation parameterization for use in general circulation models, NASA tech, Memo, 104606, 3, 85.
- Delic, G., 2011, CMAQ performance with three compilers on two platforms, 10<sup>th</sup> Annual CMAS conference, Chapel Hill, NC, 24-26.
- Delic, G., 2017, CMAQ 5.1 performance with portland and intel compilers, 16<sup>th</sup> Annual CMAS conference, 23-25.
- Dell EMC, 2017, 14G with skylake - how much better for HPC?, Ready solutions Engineering test results, Dell EMC HPC innovation lab, Sep, 2017.
- ENVIRON, 2015, Final report : CAMx speed

- improvements, Prepared for the texas commission on environmental quality, Austin, TX, Prepared by ENVIRON international corp., Work order no. 582-15-52944-05, Novato, CA.
- Etsion, Y., Tsafirir, D., 2005, A Short survey of commercial cluster batch schedulers, Technical report 2005-13, School of computer science and engineering, the Hebrew university, Jerusalem, Israel.
- Hong, D. Y., Dudhia, Y. J., 2006, A New vertical diffusion package with an explicit treatment of entrainment processes, *Mon. Weather rev.*, 134, 2318-2341.
- Hong, S. Y., Dudhia, J., Chen, S. H., 2004, A Revised approach to ice microphysical processes for the bulk parameterization of clouds and precipitation, *Mon. Weather rev.*, 132, 103-120.
- HPC Advisory Council, 2009, Interconnect analysis : 10GigE and InfiniBand in high performance computing, <http://www.hpcadvisorycouncil.com>.
- Jeon, B. J., Choi, H. G., 2014, Performance analysis of the parallel cupid code for various parallel programming models in a cluster system based on infiniband, *Korean society of computational fluids engineering*, 325-326.
- Kain, J. S., 2004, The kain-fritsch convective parameterization: an update, *J. applied meteorology*, 43, 170-181.
- Kim, J. S., Kim, S. K., Kim, S., Rho, S., Hwang, S., 2014, An Analysis of multi-level scheduling mechanism for large-scale scientific computing, *Journal of KIISE: Computing Practices and Letters*, 20(7), 379-385.
- Kordensbrock, T. H., Oidfield, R. A., Young, J., 2006, Parallel I/O advancements in air quality modeling system, *CMAS conference*, 16-18.
- Krouse, C. G., Vento, D. D., Montuoro, R., Lubin, M., McMillan, S., 2013, Evaluation of WRF scaling to several thousand cores on the yellowstone supercomputer, *Front range consortium for research computing 2013*, <http://n2t.net/ark:/85065/d7cn736p>.
- Langkamp, T., Böhner, J., 2011, Influence of the compiler on multi-cpu performance of WRFv3, *Geosci. Model Dev.*, 4, 611-623.
- Mlawer, E. J., Taubman, S. J., Brown, P. D., Iacono, M. J., Clough, S. A., 1997, Radiative transfer for inhomogeneous atmosphere: RRTM, a validated correlated-k model for the long-wave, *J. Geophys. Res.*, 102, 16663-16682.
- Odman, M. T., 1997, A Quantitative analysis of numerical diffusion introduced by advection algorithms in air quality models, *Atmos. Environ.*, 31, 1933-1940.
- Odman, M. T., Russel, A., 2000, Mass conservative coupling of non-hydrostatic meteorological models with air quality models, *Air pollution modeling and its application*, dordrecht, Kluwer, 651-660.
- Schmuck, F., Haskin, R., 2002, GPFS: A Shared-disk file system for large computing clusters, *proceedings of the conference on file and storage technologies*, Monterey, CA, 231-244.
- Shainer, G., Liu, T., Michalakes, J., Liberman, J., Layton, J., Celebioglu, O., Schultz, S. A., Mora, J., Cownie, D., 2009, Weather Research and Forecast (WRF) model performance and profiling analysis on advanced multi-core HPC clusters, 10th LCI conference, Boulder, Colorado, USA, <http://www.linuxclustersinstitute.org>.
- Yang, M., McGrath, R. E., Folk, M., 2004, Performance study of HDF5-WRF IO modules, In: *Fifth WRF user workshop*, 22-25.
- Yoon, J. W., Hong, T. Y., Kong, K. S., Park, C. Y., 2015, Improving the job success rate through analysis of user logs in HPC, *Journal of Digital Contents Society*, 16(5), 691-697.
- Yoon, J. W., Song, U. S., 2016, Study of optimization through performance analysis of parallel distributed filesystem, *Journal of Digital Contents Society*, 17(5), 409-416.
- Yoon, J. W., Song, U. S., 2017, Study of scheduling optimization through the bath job logs analysis, *Journal of Digital Contents Society*, 18(7), 1411-1418.