KNOM Review '19-03 Vol.22 No.03
https://doi.org/10.22670/knom.2019.22.3.25

제약 조건에서의 예보를 위한 기상 응용의 실행 패턴 분석

오 지 선◆, 김 윤 희°

An Analysis of Execution Patterns of Weather Forecast Application in Constraints Conditions

Jisun Oh◆, Yoonhee Kim°

요 약

기상 응용의 경우 시간적, 자원적 한계 내에서도 의미 있는 결과를 도출해 제공해야 한다. 수많은 과거 데이터를 통한 예보는 시간적인 소요가 크며, 국지성 태풍 예보와 같은 재난 안전 관련 분석/예측의 경우에는 여전히 자원적 한계가 존재한다. 태풍 예보, 도로별 침수/홍수 지역 예측 서비스 등 시간 제약하에 결과를 도출해야 하는 경우와 제한적인 물리적 환경 조건으로 인해 발생하는 문제 없이 적합한 예보를 제공해야 한다.

본 논문에서는 시간적, 자원적 조건에서도 원활한 예보 서비스 제공을 위해 기상 및 기후 예측 응용을 분석한다. 격자 크게 따른 수행 시간 분석을 통해 격자 조절을 통해 시간적 제약 조건이 있는 경우에 대처할 수 있음을 확인하였다. 또한 메모리 자원 조절을 통해 수행 시간을 분석하여 성능에 영향을 미치지 않는 최소 자원 조건을 확인하였으며 swap, mlock 분석을 통해 응용의 자원 사용 패턴을 확인하였다.

Key Words : MPAS, time constraint, resource constraint, mesh sizes, available memory sizes

ABSTRACT

For meteorological applications, meaningful results must be derived and provided within time and resource limits. Forecasts through numerous historical data are time-consuming and still have resource limitations in the case of disaster safety-related analyses/predictions such as local typhoon forecasts. Suitable forecasts should be provided without any problems caused by limited physical environmental conditions and when results are to be drawn under time constraints, such as typhoon forecasts and forecast services for flooded areas by road.

In this paper, we analyze the application of weather and climate forecasting to provide a suitable forecasting service in both temporal and resource conditions. Through the analysis of execution time according to mesh sizes, it was confirmed that a mesh adjustment can cope with the case of the temporal constraint. In addition, by analyzing the execution time through memory resource control, we confirmed the minimum resource condition that does not affect the performance and the resource usage pattern of the application through the swap and mlock analysis.

※본 연구는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2017R1A2B4005681, 2015M3C4A7065646), 분석에 도움을 주신 진계신, 정혜린 학생에 감사를 표하고자 합니다.
◆ First Author : Sookmyung Women's University Department of Computer Science, jsoh8088@gmail.com
° Corresponding Author : Sookmyung Women's University Department of Computer Science, ICT Convergence Research Institute, yulan@sm.ac.kr
논문번호 : KNOM2019-03-06, Received November 1, 2019; Revised November 28, 2019; Accepted December 20, 2019

Ⅰ. 서 론

기상 응용의 경우 시간적인 한계를 가지고 의미 있는 결과를 도출해야 한다. 실제 기상 예보를 위해서는 실제 테라, 엑사바이트 이상 규모의 데이터들을 처리해야 하므로 고성능 컴퓨팅 환경에서도 많은 시간이 소요된다. 실제 국내에서도 이러한 고성능 응용의 실행, 전송[9]을 위한 연구가 진행 중이다. 그러나 일례로 국지성 태풍 예보와 같은 재난 안전 관련 분석/예측의 경우, 자원적 한계로 인하여 여전히 실시간 대규모 데이터 분석이 필요한 과학 실험이 불가능하다. 기상청은 정기적인 예보를 위해 과거 데이터를 바탕으로 다년간 개발된 기상모델을 이용하여 슈퍼컴퓨터 등 고성능 자원을 활용한 단기/장기/중기 예보를 실시하고 있다. 그러나 급작스러운 기상 변화에 대한 위성 및 레이더로 측정한 데이터로 지형 및 광역 & 국지 구름 모델을 분석하여 강우 예상 데이터를 전처리하고, 기상모델에 적용하여 지역별 실시간 침수 예측 서비스 등은 아직 미흡하다. 이를 위한 응용 분석을 통하여 대용량 데이터의 이동, 고성능 자원의 상태에 기반을 둔 기상 분석이 필요하다. 국지적 태풍 예보, 도로별 침수/홍수 지역 예측 서비스 등 시간 제약 아래 결과를 도출해야 하는 경우에 맞는 모델을 선택하여 제공해야 한다. 또한 모델의 자원 사용량 분석을 통해 OOM(Out of Memory)가 발생하는 경우를 회피하며, 실제 요구량에 맞게 모델을 실행함으로써 더 많은 지역에 기상 예보의 기회를 제공할 수 있다. 따라서 기상 응용 프로그램의 이해를 기반으로 하여, 더욱 효율적인 예보를 진행할 수 있다[1][3].

본 논문은 다음과 같이 진행된다. 2장에서는 기상 예측 응용들에 대한 설명과 선택한 기후, 기상 예측 모델의 전반적인 실행 구조를 설명한다. 3장에서는 기상 예측 모델의 시간적 제약 상황을 위한 격자 크기에 따른 수행 시간 비교와 자원적 제약 상황을 위한 메모리 조절 분석을 진행한다. 4장에서는 이러한 분석을 통한 예측 모델 제공의 필요성을 설명하며 결론을 맺는다.

Ⅱ. 관련 연구

1 WRF

대기 연구 및 운영 예측 응용인 WRF(Weather Research and Forecast)는 전처리 단계인 WPS(WRF Pre-processing System)과 실제 예보 단계인 WRF로 구성되어서 실제 기상에 대한 예보와 실험을 실시한다[7]. WRF 응용은 전구에서 실제 예측을 원하는 도메인 지역의 정보와 동, 서 격자의 수 등 변수 값 조절을 통해 원하는 해상도 크기를 설정하고 예보 결과를 도출할 수 있다[8].

그러나 실제 예보를 원하는 지역의 위치와 해상도 변수들에 대한 정보를 사전에 알고 있어야 하며, 예보마다 전처리 과정을 수행해야 한다.

2 MPAS

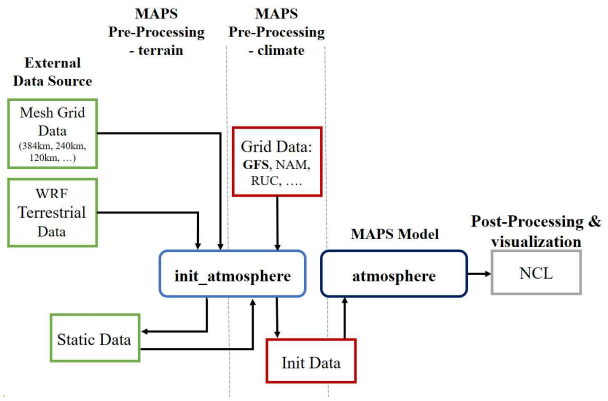


그림 1. MPAS 실행 구조도

Fig. 1. Execution structure of MPAS

기후 및 기상 예측 연구에 사용하는 대기, 해양 및 기타 지구 시스템 시뮬레이션 응용인 MPAS(Model for Prediction Across Scales)[2]는 기후, 지형 등 전처리 과정 단계(init_atmosphere)와 해석 단계(atmosphere)로 나뉜다. 그림 1은 MPAS 응용 실행 구조도이다. 전처리 과정에서는 사용될 전구 도메인에 지형자료(WRF Terrestrial Data)를 생성하고, 격자 크기(Mesh Grid Data)에 따라 기상, 기후 데이터를 내삽한다. 지형 자료는 원하는 해상도의 격자 파일과 전구 데이터인 GEOG를 입력받아 생성한다. 기상 자료는 WPS의 ungrib 과정에서 얻은 GFS 데이터와 지형자료 생성으로 나온 결과 파일(Static Data)을 입력으로 하여 사용하여 생성한다. 이러한 전처리 과정이 완료되면 해석 모델을 수행하여 모델 적분을 수행한다. 전처리 과정에서 생성된 기상 초기 데이터 init.nc 파일과 N개의 멀티 프로세싱이 가능한 격자 분할 자료 데이터 graph.info.part.N 파일을 사용한다. 그 후 나온 결과 파일을 사용하여 NCL 도구를 통해 시각화하여

예측 결과를 제공한다[4].

MPAS 응용은 초기에 정의된 격자 크기 별로 사전에 도메인에 맞춰 기후, 지역 내삽 등 전처리 과정을 수행하면, 예보 시 해석 모델만 수행하여 예보 결과를 도출할 수 있다. 따라서 다양한 제약 조건에서 기상 예보를 수행하는 데 적합한 응용이다.

Ⅲ. 기후 및 기상 예측 모델 분석

여러 제약 사항이 존재할 경우에 적합한 예측을 위해 MPAS 응용의 분석이 필요하다. 이를 위해 격자별, 메모리 조절 실험을 통해 MPAS를 분석하여 시간적, 자원적 제약에 대비할 수 있다. 본 실험에서는 지형 자료는 120km 격자 크기와 전지구 지형 데이터 13GB를 사용하였다. 기후 자료는 수십 개의 대기 및 토양 변수 데이터 세트인 GFS를 사용하였으며 총 24시간 적분 기간 동안 2시간 단위로 예보를 진행하였다. 실험 환경은 Intel(R) Xeon(R) CPU E5-1650 v4 @ 3.60GHz, 12 cores, RAM 32GB이며, MPAS version은 6.2를 사용하였다.

1. 실행 단계별 수행 시간 비교

본 실험에서는 자원 상태 및 실행 조건에 따라 적시적으로 예보를 수행하기 위해 응용의 사전 분석을 진행하였다. 격자에 크기에 따른 실행 단계별 수행 시간을 분석한다.

실험을 위해 격자 크기 240km, 120km 두 격자를 사용하여 수행 시간을 측정하였다(표 1). 지형 전처리 단계에서는 240km 격자에서는 약 23분 16초, 120km 격자에서는 약 25분 42초 정도 소요된다. 기후 전처리 단계에서는 240km 격자에서는 약 3초, 120km 격자에서는 16초 정도 수행되었다. 실제 적분 계산을 수행하는 atmosphere 해석 단계에서는 240km 격자에서는 약 18분 33초, 120km에서는 약 25분 42초 정도 소요되는 것을 확인할 수 있다. 수행 시간은 약 24.73% 정도 증가하였으며, 조밀한 격자 크기일수록 해석 단계의 수행 시간에 차이가 나는 것을 확인할 수 있다.

표 1. 격자 크기에 따른 실행 단계별 수행 시간
Table 1. Execution time for each steps per mesh sizes

steps	mesh size	
	240km	120km
init_atmosphere(terrain)	23min 16s	25min 42s
init_atmosphere(climate)	3s	16s
atmosphere	18min 33s	25min 42s

본 실험의 결과에 따라 다양한 격자 크기를 통해 시간적 제약 조건이 존재하는 경우 적합한 격자를 선택하여 제공함으로써 적합한 예보를 제공할 수 있다. 예를 들어 빠른 예보를 제공해야 할 경우, MPAS에서 제공하는 가장 큰 격자인 480KM를 통해 예측하고 상세한 분석이 필요한 경우에는 가장 작은 격자(3KM)를 사용함으로써 상황에 맞게 제공할 수 있다.

2. 가용 메모리 조절에 따른 비교 및 분석

앞서 진행한 실험을 통해 격자 크기에 따라 수행

그림 2. 가용 메모리별 함수 수행 시간
Fig. 2. Function execution time per available memory sizes

시간이 많이 증가하는 단계를 확인하였다. 본 실험에서는 제한적 자원 환경이 존재하는 경우 예보를 위해 필요한 메모리를 확인한다. 또한 가용 메모리 조절을 통해 해석 모델인 atmosphere 단계의 코드를 분석함으로써 수행 시간에 영향을 미치는 함수와 swap이 발생하는 함수를 파악한다.

표 2. 가용 메모리 조절에 따른 수행 시간, swap 크기 비교
Table 2. Comparison of execution time and swap size according to available memory sizes

Memory size(GB)	Swap size(GB)	execution time
7.29(default)	0.247	25min 42s
6	0.813	25min 18s
3	4.67	81min 45s
2	4.79	85min 25s
1.5	4.98	100min 31s

가용 메모리를 6GB, 3GB, 2GB, 1.5GB, 1GB로 조절하여 수행 시간과 swap 크기를 측정하였다(표 2). swap 크기는 swap 공간을 사용한 크기이며 glances[5] 툴로 관측하였다. 가용 메모리 6GB에서 수행 시간 약 25분, 3GB에서 수행 시 약 81분으로 급격하게 증가하는 것을 확인할 수 있다. 또한 3GB에서 수행하였을 경우, swap 크기 역시 4.67GB로 증가하는 것을 확인하였다. 120KM 격자의 경우, 가용 메모리를 6GB로 하여 실행할 때 수행 시간에 영향을 미치지 않으며 불필요한 자원 사용을 방지와 성능을 유지할 수 있다. 이를 통해 자원적으로 제약 사항이 존재할 경우 예보 수행을 위해 시간에 영향을 미치지 않는 자원 조건을 분석하여 수행함으로써 해결할 수 있다.

수행 시간이 급격하게 증가하는 위의 실험 결과를 통해 수행 시간과 swap 크기 증가의 원인을 파악하기 위해 내부 함수를 분석하였다. atmosphere의 모든 호출 함수의 수행 시간을 확인하고, 수행 시간의 대부분을 차지하는 함수의 swap 발생 코드를 확인하였다. 그림 2는 가용 메모리에 따른 내부 함수의 수행 시간을 나타낸다. atmosphere의 주요 함수 initialzie, running, finalize 3단계가 있으며 수행 시간 증가는 initialize 내부 함수인 time integration과 running 내부 함수인 atm_timestep에서 발생하는 것을 확인할 수 있다(atm_srk3는 atm_timestep 내부 함수). initialize 함수는 메모리 할당을 수행하므로 가용 메모리가 작으면 수행 시간이 증가한다. 다시 말해 할당할 내용의 사이즈가 가용 메모리보다 큰 경우 swap이 발생하는 것을 확인하였다. 그러나 대부분의 swap 발생은 running 단계에서 발생하며 이는 수행 시간이 증가한 atm_timestep 함수에 해당한다. 이는 실제 atm_srk3 과정이 각 시간 간격에 대한 가중치를 확인 후 가중치를 획득하는 과정을 수행함에 따라 swap이 발생하는 것으로 분석된다.

2.1 swap 발생 비교 분석

swap 이 발생하는 구간 및 함수를 확인함에 따라 수행 시간과의 상관관계를 확인하기 위해 swap 발생 가상 메모리 주소, swap 발생 횟수를 확인하였다. 빠른 분석을 위해 4개의 멀티 프로세서를 사용하여 가용 메모리의 범위를 2GB, 1.2GB, 1GB로 조절하여 수행하였다. swap 트레이서[6]를 사용하여 atm_timestep 구간을 트레이싱 하였다. 그림 3은 각 메모리당 swap이 발생한 가상 메모리 영역을 나타낸다. x축은 처음 swap이 발생한 시점부터 마지막 swap 발생까지의 수행 시간을 나타내며, y축은 가상 메모리 주소를 나타낸다. 2GB, 1.2GB, 1GB로 수행했을 때 총 수행 시간은 각각 약 211초, 약 256초, 약 674초 소요되었으며, swap이 발생하는 시간은 각각 약 145(0~144, 288)초, 124초, 162초이다. Swap이 발생하는 가상 메모리 주소는 일정하며, swap 발생 횟수는 1GB, 1.2GB의 경우

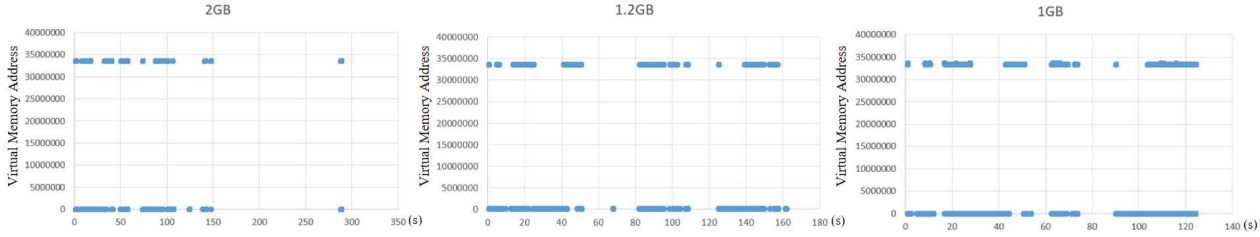


그림 3. 가용 메모리별 swap 분석
Fig. 3. Analysis of swap per available memory sizes

1,048,576번, 2GB의 경우 635,871번 발생하였다. 실제 응용의 수행 시간은 2GB에 비해 1GB, 1.2GB가 약 3배 정도 늘어났으나, 전체 수행 시간 대비 swap 발생 구간에 대한 시간은 약 20% 이하로 작다. 따라서 swap 발생 횟수는 늘어났으나, 수행 시간 증가에 swap이 영향을 주지 않음을 알 수 있다.

2.2 mlock 유무에 따른 수행 시간 비교 분석

수행 시간이 증가하는 원인을 찾기 위해 수행 시간 차이가 나는 메모리 크기(1.2GB, 1GB)에 따라 mlock을 걸어 수행하였다. 빠른 분석을 위해 4개의 멀티 프로세서를 사용하였으며, atm_timestep 구간에 mlock 코드를 삽입하였다.

mlock 유무에 따른 각 메모리별 수행 시간은 표 3과 같다. 수행 시간이 증가하는 1GB의 경우 atm_timestep 구간에서는 약 565초 정도 소요된다. 이 구간에 mlock을 걸고 수행했을 경우 mlock 적용한 데이터 크기는 약 544MB이며, 수행시간은 약 568초로 차이가 거의 없다. 이를 통해 mlock으로 인한 성능 향상은 어려움을 확인하였으며, swap out 발생이 적고 데이터 로컬리티가 좋은 것으로 예상된다.

표 3. mlock 유무에 따른 수행 시간 비교
Table 3. Execution time comparison with or without mlock

function	1.2GB (w/o mlock)	1.2GB (w/ mlock)	1GB (w/o mlock)	1GB (w/ mlock)
initialize	23.49	24.28	27.77	29.51
running	207.5	205.45	638.75	642.1
atim_timestep	140.4	138.43	564.59	567.67
total	257.1	258.17	695.26	700.66

본 논문에서는 효율적인 예측을 위해 기상 및 기후 예보 모델인 MPAS 응용을 분석하였다. 앞서 진행한 실험을 통해 태풍, 침수 등 실시간 예보가 필요한 경우 정확한 기후 예상보다는 빠른 시간 내에 예보하는 것이 중요하기 때문에 큰 격자 크기로 수행해야 한다. 반대로 시간 제약 없이 정확한 예보가 필요한 경우에는 응용 모델의 자원 사용에 대한 분석을 통하여 적합한 자원 환경에서 실행할 수 있다. 또한 실제 대규모의 예보의 경우에는 수많은 데이터 처리가 필요하기 때문에 메모리 요구량이 클 수 있다. 따라서 가용 메모리 조절을 통해 수행시간 변화가 없는 실제 예보에 사용하는 메모리양을 파악함으로써 자원적으로 한계가 있는 상황에서도 예보가 실행 가능 하도록 하는데 도움이 될 것이다.

V. 결 론

본 논문에서는 기상 및 기후 응용의 다양한 조건에 맞도록 수행하기 위해 MPAS 응용을 분석하였다. 격자 크기 조절을 통해 수행 시간을 비교하여 재난 상황 같은 경우에는 큰 격자로 모델을 수행하여 빠른 시간 내에 예보를 진행할 수 있음을 확인하였다. 또한 가용 메모리에 따른 수행 시간, swap, mlock 분석을 통해 MPAS 모델의 자원 사용 패턴을 파악하였다. 이를 통해 실제 예보와 같이 더 많은 데이터를 통해 더 많은 자원을 사용하는 경우가 존재 할 때 자원을 조절하여 성능에 영향을 미치지 않는 범위 내에서 문제 없이 실행이 가능함을 보였다.

향후 연구로는 컨테이너 클러스터 환경 내에서 실제 격자별, 메모리별 이미지를 구성하여 시간적, 자원적 제약 조건에 따라 예보를 위한 스케줄링 기법을 진행할 것이다.

References

[1] Park, Sang-Hun, Joseph B. Klemp, and William C. Skamarock. "A comparison of mesh refinement in the global MPAS-A and WRF models using an idealized normal-mode baroclinic wave simulation." Monthly Weather Review 142.10 (2014): 3614-3634.

[2] MPAS, https://mpas-dev.github.io/

[3] Hagos, Samson, et al. "Error characteristics of two grid refinement approaches in aquaplanet simulations: MPAS-A and WRF." Monthly Weather Review 141.9 (2013): 3022-3036.

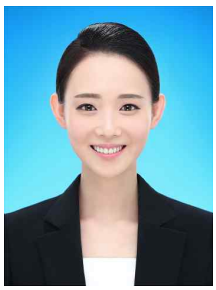
[4] MPAS arhcitecture, http://www2.mmm.ucar.edu/projects/mpas/mpas_atmosphere_users_guide_6.0.pdf

[5] glances, https://nicolargo.github.io/glances/

[6] swap 트레이서, https://github.com/lynring24/swptracer

[7] WRF, https://www2.mmm.ucar.edu/wrf/users/
[8] 오지선, 이효정, 김윤희, "HPC 응용의 런타임 실행 패턴에 따른 소프트웨어 정의 계산 구조 분석", KNOM Review, Vol. 20, No. 1, pp. 17-23
[9] 문정훈, 곽재승, 홍원택, 김기현, 이상권, 김동균, 김용환, 유기성, "과학빅데이터 고속전송을 위한 ScienceDMZ 구축 방안 연구", KNNOM Review, Vol. 22, No. 2, Oct. 2019, pp. 12-21

오 지 선 (Jisun Oh)



2017년 2월 : 숙명여자대학교 컴퓨터과학과 졸업
2017년 9월 ~ 현재 : 숙명여자대학교 컴퓨터과학과 석사 과정

<관심분야> 클라우드 컴퓨팅, 분산컴퓨팅, GPU 가상화, 작업 스케줄링

김 윤 희 (Yoonhee Kim)

1991년 : 숙명여자대학교 전산학과 졸업(학사).
1996년 : Syracuse University 전산학과 졸업(석사).
2000년 : Syracuse University 전산학과 졸업(박사).
1991년~ 1994년 : 한국전자통신연구원 연구원.
2000년~2001년 : Rochester Institute of Technology 컴퓨터공학과 조교수.
2001년 ~ 2016년 : 숙명여자대학교 컴퓨터과학부 교수.
2017년 ~ 현재 : 숙명여자대학교 소프트웨어학부 교수.
<관심분야> 클라우드 컴퓨팅, 워크플로우 제어, 그리드/클라우드 관리