Journal of Korea Multimedia Society Vol. 20, No. 2, February 2017(pp. 363-370)
https://doi.org/10.9717/kmms.2017.20.2.363

대기질 예보의 성능 향상을 위한 커널 삼중대각 희소행렬을 이용한 고속 자료동화

배효식[†], 유숙현[††], 권희용[†††]

Fast Data Assimilation using Kernel Tridiagonal Sparse Matrix for Performance Improvement of Air Quality Forecasting

Hyo Sik Bae[†], Suk Hyun Yu[††], Hee Yong Kwon[†††]

ABSTRACT

Data assimilation is an initializing method for air quality forecasting such as PM10. It is very important to enhance the forecasting accuracy. Optimal interpolation is one of the data assimilation techniques. It is very effective and widely used in air quality forecasting fields. The technique, however, requires too much memory space and long execution time. It makes the PM10 air quality forecasting difficult in real time. We propose a fast optimal interpolation data assimilation method for PM10 air quality forecasting using a new kernel tridiagonal sparse matrix and CUDA massively parallel processing architecture. Experimental results show the proposed method is 5~56 times faster than conventional ones.

Key words: Data Assimilation, Optimal Interpolation, CUDA, Sparse Matrix

1. 서 론

중국 주요 공업도시지역에서 발생한 고농도 미세먼지에 의한 스모그가 편서풍 계열의 바람을 타고 국내로 유입되고, 국내에서 정체 또는 순환하면서 국내 배출원이 가중되어 고농도 연무현상을 유발하고 있다[1]. 이러한 연무를 동반한 고농도 미세먼지(PM_{10})는 국민 건강을 위협하는 직접적인 요소로서 예방을 위해서는 미세먼지를 포함하는 대기질 예보가 매우 중요하다[2]. 환경부에서는 2013년 8월 수도권 시범예보를 시작으로 2014년 2월 이래로 전국을 대상으로 정식예보를 시행하고 있으며, 미세먼지 예보 적중률은 71.8%로 분석되었다[3]. 하지만, '나쁨' 이상의 고농도 감지확률은 이보다 낮아서 실제 시민들이 느끼는 오염도와 많은 차이를 보이고 있는바, 기존 대기질 수치예보의 정확도를 향상시키기 위한 연구가 필요하다.

대기질 예보는 기본적으로 예보대상 오염물질에 직, 간접적으로 영향을 미칠 수 있는 인자들을 입력받아 모델링을 수행한다. 대기질 예보 입력 인자에는 기상 자료, 기상예보 자료, 대기질 자료, 대기오염물질 배출량 등이 있으며, 대기 중으로 배출된 오염물질들은 확산과 이류, 화학적 반응으로 인하여 생성과 소멸을 반복하기 때문에, 특정 시공간에 존재하는 오염물질의 농도를 예측하기 위해서는 화학수송모델링(CMAQ, CAMx)이 필요하다. 또한 체계적인 화학수송 모델링을 위해서는 기상 모델링(WRF), 배출량 모델링(SMOKE) 등이 선행되어야 한다. 이러한 대

※ Corresponding Author: Heeyong Kwon, Address: (430-714)22 Samdeokro 37 beon-gil, Manan-gu, Anyang-si, Gyeonggi-do, Korea , TEL : +82-31-467-0878, FAX : +82-31-467-0800, E-mail : hykwon@anyang.ac.kr
Receipt date : Jan. 13, 2016, Approval date : Jan. 26, 2017

[†] Dept. of Computer Eng., Anyang University (E-mail : glassant@dreamwiz.com)
[††] Dept. of Information & Communications Eng., Anyang University (E-mail : shy@anyang.ac.kr)
[†††] Dept. of Computer Eng., Anyang University

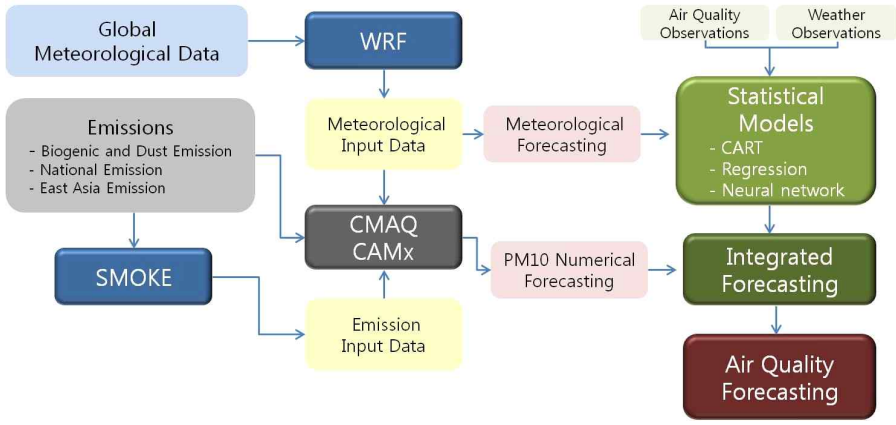


Fig. 1. A schematic diagram integrated forecasting system.

기질 예보 모델의 개념을 Fig. 1에서 보이고 있다.

대기질 예보는 Fig. 1에서 나타낸 바와 같이 화학 수송모델에 의해 계산된 기상자료, 배출량 자료와 관측된 대기질 자료, 날씨 자료를 통계모델에 입력하여 실행하고, 출력으로 PM_{10} 예보 값을 생성하는 과정을 의미한다.

대기질 예보에 있어 모델에 입력되는 초기장은 예보의 정확성에 영향을 미치는 매우 중요한 요소이다. 하지만, 기존에 운영되고 있는 대기질 예보 모델은 측정된 자료를 이용하지 않고, 모델링의 결과로 나온 예보 값을 초기장으로 사용하기 때문에, 예보의 오차가 커지는 한계를 지닌다. 이것은 예보의 특성상, 예보를 하는 현시점에는 미래의 측정치가 존재하지 않음으로, 불가피하게 예보 값을 사용할 수밖에 없는 이유에서 기인된다.

이러한 불가피성은 개선될 수 있다. 예보를 한 시점에는 존재하지 않던 측정값들이 시간이 지남에 따라 지속적으로 생성되기 때문에, 이 값들을 이용하여 예보 값을 보정할 수 있기 때문이다. 이러한 개념을 자료동화(Data Assimilation)라고 하며, 자료동화 과정을 통해 좀 더 신뢰도 높은 초기장을 생성할 수 있고, 예보의 오차 또한 감소하는 효과를 볼 수 있다 [4,5].

따라서, 예보의 성능을 향상시키기 위해서는 초기장을 개선할 수 있는 자료동화 과정이 필요하다. 자료동화를 수행하기 위해 여러 가지 기법이 연구되어 왔으며, 최근 최적내삽법(Optimal Interpolation, 이하 OI)에 의한 자료동화 기법이 많이 사용되고 있다. 하지만, OI를 구현 시, 기존의 방법대로 순차 처리를 하면, 자료 벡터의 크기가 커짐에 따라 기하급수적으로 메모리 공간과 수행 시간이 증대되므로 그대로 현업에 적용하기에는 어려움이 있다.

이러한 문제점을 해결하기 위해서 본 논문에서는 두 가지 해결책을 제안하였다.

첫째, OI를 구현 시 커널 삼중대각 희소 행렬(KTSM: Kernel Tridiagonal Sparse Matrix, 이하 KTSM)을 제안하여 메모리 공간을 줄이고, 행렬 곱셈의 전통적인 문제점인 계산 복잡도를 낮추었다.

둘째, 수행시간 단축을 위하여 CUDA(Compute Unified Device Architecture, 이하 CUDA)를 이용하여 병렬로 구현하였다.

제안한 방법의 우수성을 평가하기 위해서 OI를 기존의 순차처리 기법과 제안한 KTSM 및 병렬처리 기법으로 구현하여 수행시간을 측정하고, 상호 비교하였다. 그 결과, 제안한 방법이 기존의 순차처리 방식에 비하여 약 15~56배의 속도를 향상시키는 것을 확인하였다.

본 논문에서는 제안한 방법을 다음과 같은 순서로 설명한다. 2장에서는 제안한 방법의 이론적 배경인 자료동화와 OI에 대하여 기술하고, 3장에서는 제안한 방법에 대해 설명하고, 4장에서는 실험 및 분석 결과를 보이고, 5장에서 결론 및 향후 연구를 제시하는 것으로 끝맺는다.

2. OI에 의한 자료동화

2.1 자료동화

미세먼지와 같은 대기질 예보에 있어 초기장은 예

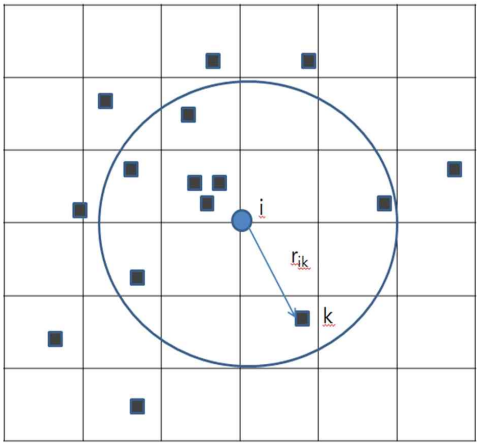


Fig. 2. Concept of Data Assimilation.

보의 정확성에 영향을 미치는 중요한 요소이다. 하지만, 기존의 대기질 예보 모델에서는 이전 시간에 예보된 모델 결과를 초기장으로 사용하고, 변경하지 않아 예보의 정확성이 떨어지는 문제점을 가지고 있다. 예보의 정확성을 높이기 위해서는 모델링의 결과로 생성된 예보 값을 측정망에서 측정한 실측값으로 보정하여 초기장을 갱신하는 계산과정이 필요하다. 이러한 기법을 자료동화라 하며, 그 개념을 Fig. 2에 나타내었다.

자료동화란 Fig. 2에 나타낸 바와 같이 예보 값(격자 중앙의 파란색 점) i를 영향반경(파란색 큰 원)내에 속하는 불규칙하게 분포하는 측정값들(검은색 사각형)을 이용하여 보정하는 방법이다. Fig. 3의 빨간 점들은 자료동화에 사용하는 중국과 국내 측정망 위치를 나타낸 것으로[6] 이 지점에서 측정된 값들이 Fig. 2에서는 검은색 사각형으로 표현되었다.

Fig. 4에는 동아시아 지역에서 발생한 미세먼지에 대해서 자료동화를 적용하기 전과 후의 농도 분포를 나타내었다[6]. Fig. 4의 지도상에 표현된 색상들은 미세먼지의 농도가 낮은 파란색부터 농도가 높은 붉은색까지 표현되었고, 상세한 수치에 따른 색상은 각 그림의 오른쪽에 표기하였다. Fig. 4(a)는 자료동화

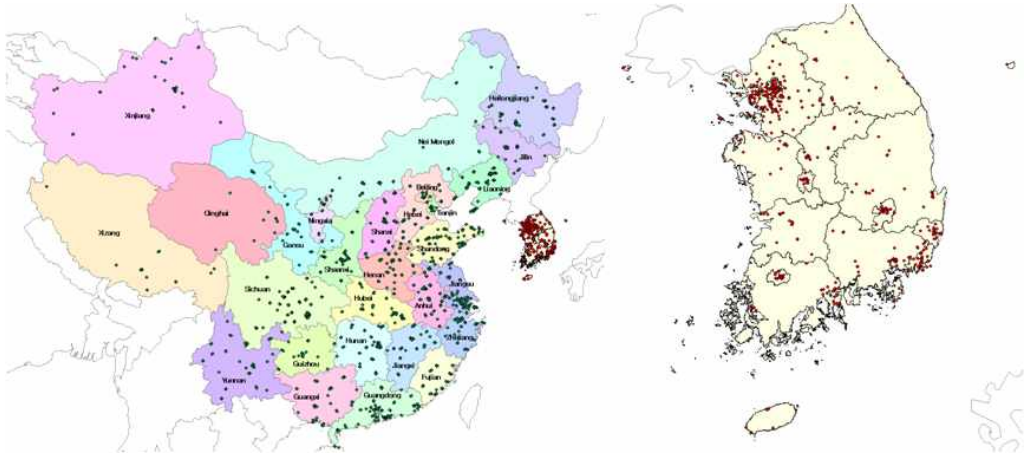
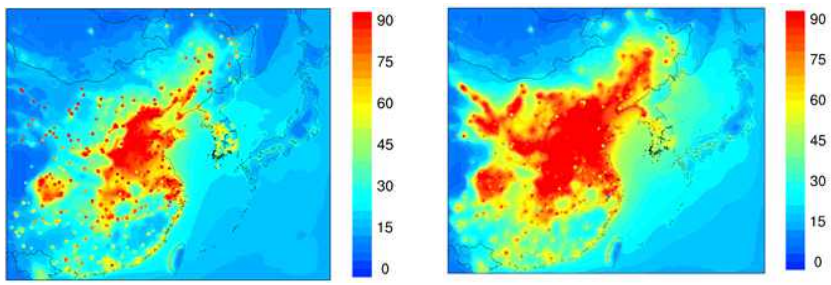


Fig. 3. China and domestic metrology sites for forecasting model initial condition data assimilation.



(a) PM_{10} before data assimilation (b) PM_{10} after data assimilation

Fig. 4. Distribution of PM_{10} before and after data assimilation in East Asia.

를 수행하기 전의 미세먼지 농도 분포를 나타낸 것으로 이는 예보 값이기 때문에 거리에 따른 농도의 확산 정도가 전혀 적용되지 않았다. 하지만, 자료동화를 수행한 후에는 Fig. 4(b)에서 보이는 바와 같이 예보 값이 측정값들로 보정되어 현시점의 미세먼지 농도 분포를 더 잘 반영하고 있음을 알 수 있다.

이러한 자료동화 기법에는 연속보정법(SCM, Successive Corrections Method)[7], 최적내삽법(OI, Optimal Interpolation)[8], 3차원 변분법(3D-VAR, Three Demensional Variational Analysis)[9], 칼만 필터링(KF, Kalman Filterling)[10], 4차원 변분법(4D-VAR, Four Demensional Variational Analysis)[11] 등이 있다.

본 논문에서는 현업 대기질 예보모델에 적용할 경우, 제안한 KTSM과 CUDA 병렬 처리 방식에 의해 가장 극적인 성능 향상 효과를 볼 수 있는 OI를 이용하여 자료동화를 수행하였다.

2.2 OI

OI의 식은 Eliassen(1954)에 의해 처음으로 유도되었으며, Gandim(1963)이 다변수 최적내삽식을 독자적으로 유도했고, 이 식들을 구소련의 객관 분석에 적용하였다[12]. OI와 이전 방법들과의 가장 큰 차이는 경험적으로 결정된 가중함수를 사용하지 않고, 배경장과 관측의 차이를 통계적 분석과 관측이 가지고 있는 오차를 고려한 가중함수를 사용한다는 것이다.

OI는 통계적 내삽으로 결정된 최적 가중치로 가중된 관측 증분을 배경장에 더해줌으로써 다음과 같이 결정된다.

$$x_a = x_b + W[y_o - H(x_b)] = x_b + Wd \quad (1)$$

$$W = BH^T(R + HBH^T)^{-1} \quad (2)$$

식(1)에서 관측값(y_o)은 길이가 p인 벡터, 분석장(x_a)과 배경장(x_b)은 길이가 n인 벡터이며, 가중치 W는 n × p 행렬이고, 벡터 d는 길이가 p인 '개선값' 혹은 '관측 증분' 벡터이다. 식(2)에서 B 는 배경장 오차 공분산으로 n × n 행렬이고 R 은 관측 오차 공분산으로 p × p 행렬이고 H 는 모형 변수를 관측 변수로 변환시키는 비선형 관측 연산자로서 p × n 행렬이다.

이 방법을 기존의 순차처리 방식으로 구현할 경우, OI의 주요 파라메터인 배경장(x_b)과 관측치(y_o) 벡터의 크기가 증가함에 따라 행렬 곱셈의 계산복잡도가 기하급수적으로 증가하여 수행시간이 매우 오래 걸리므로 그대로 현업에 적용하기에 어려움이 있다.

본 논문에서는 기존 문제점의 개선 및 고속의 자료동화를 위하여 KTSM을 제안하여 계산 복잡도를 개선하였으며, CUDA[13,14]를 이용한 병렬처리 방식으로 수행시간을 단축시켰다.

3. 고속 최적내삽 자료동화

3.1 KTSM을 이용한 행렬 곱셈의 고속화

OI를 구현하는 과정 중 가장 수행 시간이 많이 걸리는 부분 중 하나는 식 (2)의 B와 H^T의 행렬 곱셈을 구하는 과정이다. B는 배경장 오차 공분산으로 배경장(x_b)이 길이 n인 벡터라 할 때, $O(n^2)$의 공간 복잡도를 갖고, H^T는 관측값(y_o)이 길이 p인 벡터일 때, $O(p \times n)$의 공간 복잡도를 갖는다. 또한, 이들 행렬의 곱셈을 위해 $O(p \times n^2)$의 계산 복잡도를 갖는다. 이는 각각 배경장(x_b)과 관측값(y_o)이 커질수록 공간 복잡도는 더욱 증가하고, 수행시간도 길어짐을 의미한다.

하지만, B의 각 요소는 주어진 커널 함수에 의해 계산된 값을 가지며, 정해진 영향 반경 밖의 요소는 모두 0이 된다는 특징을 고려한다면, 이를 희소행렬로 만들 수 있다. 일반적으로 $N \times N$ 크기의 행렬을 저장하기 위해 메모리 공간이 N^2이 필요하다면, 희소행렬로 표현하면 $2N$ 또는 $3N$의 메모리 공간이 요구되므로 현저하게 공간 복잡도는 감소하게 된다. 그러나 이 같은 표현 형식은 공간복잡도는 낮출 수 있지만 계산 복잡도는 증가하게 된다.

희소행렬의 표준 형태는 Fig. 5에서 보이듯이 여러 가지가 있다[15]. 본 논문에서는 대상 행렬의 특징과 부합되는 형태인 Fig. 5의 (c)와 같은 블록 삼중대각 희소행렬을 기본으로 하되, 모든 블럭의 모양이 동일하고, 주어진 커널 함수에 따라 블록이 정의되는 KTSM을 제안하였다. 제안한 방법에서 적용한 커널 함수는 가우시안 함수이다.

KTSM에 의해 행렬 곱셈을 수행하게 되면, 해당 행렬을 위한 메모리는 필요로 하지 않고, 단지 커널 함수와 커널함수를 적용할 중심 위치만이 필요하고, 일반적인 행렬 곱셈의 시간복잡도가 $O(n^3)$임에 비해 $O(k \times n)$으로 크게 낮아진다. 또한 이러한 중심 위치

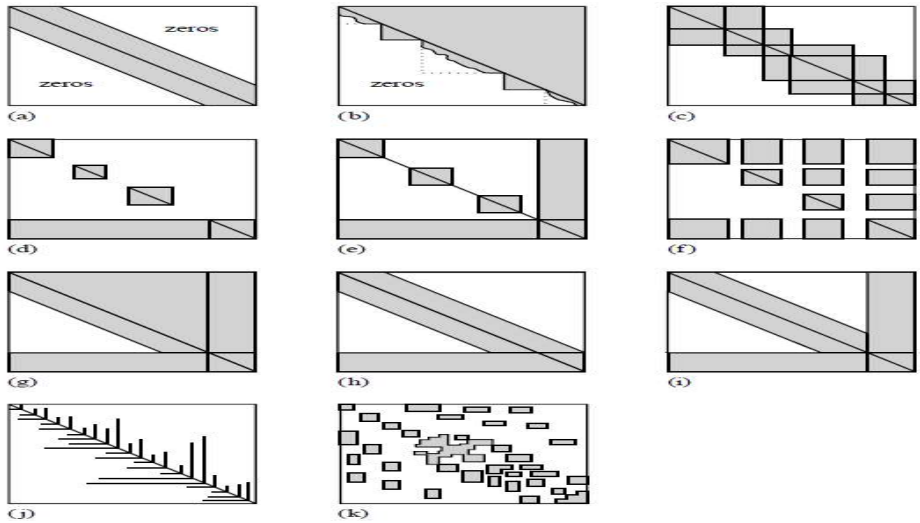


Fig. 5. Some standard forms for sparse matrices. (a) Band diagonal; (b) block triangular; (c) block tridiagonal; (d) singly bordered block diagonal; (e) doubly bordered block diagonal; (f)singly bordered block triangular; (g) bordered band–triangular; (h) and (i) singly and doubly bordered band diagonal; (j) and (k) other.

도 대각성분에만 위치하기 때문에 계산의 효율성이 매우 뛰어나다.

Fig. 6에는 제안한 KTSM을 이용하여 B와 H^T의 행렬 곱셈을 구하는 과정을 나타내었다. Fig. 6에서 m과 n은 배경장(x_b) 벡터의 길이이고, col_size는 배경장(x_b)를 2차원 격자로 표현했을 때 칼럼의 크기이며, p는 관측값(y_o)의 개수이다. 또한, base_point는 2차원 격자 내에서 기준점이고, target_point는 base_point와의 비교를 위한 비교점이다. L은 영향반경이고, BKernel은 커널 함수이며, B는 H^T를 의미하고, C는 두 행렬 B와 H^T의 곱셈 결과를 의미한다.

```
For i = 0 to m
    base_point ← (i / col_size, i % col_size)
    For j = 0 to p
        sum = 0
        For k = 0 to n
            target_point ← (k / col_size, k % col_size)
            If distance between base_point and
target_point < L Then
                sum = sum + Bkernel[L] * B[k, j]
            End If
        End For
        C[i, j] = sum
    End For
End For
```

Fig. 6. Matrix multiplication using KTSM.

3.2 CUDA를 이용한 병렬처리

자료동화를 수행 시 제안한 KTSM을 이용하여 행렬 곱셈을 위한 메모리 공간을 줄이고, 계산 시간도 고속화 시켰으나, 기존의 순차처리 방식으로는 빠른 속도를 요구하는 현업에 바로 적용하기에 무리가 있다. 이러한 문제를 해결하기 위해서 CUDA를 이용한 병렬처리로 더욱 수행시간을 단축시켰다. CUDA는 그래픽 처리 장치(Graphic Processing Unit, GPU) 전문 제조업체인 NVIDIA에서 GPU의 대용량 병렬처리 능력을 범용화 시킨 것으로 기존의 CPU가 2~16개의 Core로 구성되는 것에 비해 보통 500~1000개 이상의 Core로 구성되고, Pipelining 처리에 의해 고도의 병렬 처리를 수행한다. 이와 같은 GPU 환경에서 CUDA가 제공하는 grid의 block 수와 차원, block당 thread의 수 등을 고려하여 알고리즘을 작성하는 것이 중요하며, 이 과정을 Fig. 7에 나타내었다.

4. 실험 및 분석

본 논문에서는 고속의 자료동화를 위해서 제안한 KTSM과 CUDA를 이용하여 OI 수행하였고, 그 결과를 각각 Table 1과 Table 2에 제시하였다. 실험에

```
i = blockDim.x * blockIdx.x + threadIdx.x
j = blockDim.y * blockIdx.y + threadIdx.y

If i < m And j < p Then
    base_point ← (i / col_size, i % col_size)
    sum = 0
    For k = 0 to n
        target_point ← (k / col_size, k % col_size)
        If distance between base_point and
            target_point < L Then
            sum = sum + Bkernel[L] * B[k, j]
        End If
    End For
    C[i, j] = sum
End If
```

Fig. 7. Parallel Processing using CUDA.

사용된 데이터는 측정망별로 수집된 미세먼지 측정값과 예보 값이고, 데이터 수집 기간은 2015년 2월이다. 또한, 측정 지역은 동아시아(27km), 한국(9km), 수도권(3km)으로 각 지역 안에 표기된 거리는 격자의 간격 의미한다. 실험에 사용한 하드웨어는 Intel사의 CPU i5-2500K(3.3Ghz), Nvidia사의 GPU Geforce GTX960이다.

Table 1은 OI를 순차처리 방식과 제안한 KTSM 및 CUDA 병렬처리를 적용한 경우에 대해 이론적으로 비교한 결과이다. 메모리 공간과 연산 횟수 측면에서 봤을 때, 기존의 방법은 행렬의 크기가 커짐에 따라 메모리 공간과 수행 횟수(시간)가 매우 증가되지만(각각 $O(n^2)$과 $O(n^3)$), 제안한 방법은 큰 차이가 없고, 단지 GPU 하드웨어 용량에 따라 선형적인 증가(각각 $O(1)$과 $O(n)$)만이 있을 뿐임을 알 수 있다.

Table 2에서는 각각의 방법으로 OI를 수행한 시간을 비교하여 나타내었다. Table 2의 결과를 보면 제안한 방법이 기존의 순차 처리 방식에 비해 수행속도가 15~56배 극적으로 개선되었음을 알 수 있다. 특히, 이 중 동아시아의 경우를 예를 들어 설명하자면, 기존의 방식에 의하면 B와 H^T 행렬이 각각 90 Giga Bytes와 500 Giga Bytes의 배정도 부동소수 변수 공간을 필요로 하며, B와 H^T의 행렬 곱셈을 위해 $(317 \times 10^6) \times 1720 \times (317 \times 10^6) \approx (173 \times 10^{18})$번의 배정도 부동소수 곱셈 연산을 필요로 한다. 이는 동아시아의 경우, 특정 시점의 예보를 위해서는 2시간에 이르는 수행 시간을 필요로 한다는 의미이다. 이 때 제안한 방법으로 같은 연산을 수행했을 때 걸리는 시간은 2분 정도이다.

5. 결론 및 향후 연구

본 논문에서는 미세먼지로 대표되는 대기질 예보의 성능을 향상시키기 위해서 고속 자료동화 기법을 제안하였다. 자료동화의 한 방법인 OI를 기존 방법대로 순차처리를 하면, 자료 벡터의 크기가 커질수록

Table 1. Memory Spaces and Operation Number Comparison between Serial and Parallel Processing with KTSM for a Matrix Multiplication(k is a kernel operation memory, 5×5, and P is a pipelining steps in a GPU, 640 is a core number in GTX960)

Matrix Size	Serial Processing	Parallel Processing (CUDA)
$2^{10} \times 2^{10}$	2^{20}, 2^{30}	k, $\frac{k \times 2^{10}}{640 \times P}$
$2^{14} \times 2^{14}$	2^{28}, 2^{42}	k, $\frac{k \times 2^{14}}{640 \times P}$
$2^{20} \times 2^{20}$	2^{40}, 2^{60}	k, $\frac{k \times 2^{20}}{640 \times P}$

Table 2. Execution Time Comparison between Serial OI and a Parallel OI with KTSM

Forcasting Area	Grid Interval	Grid Dimension	B matrix	Serial Processing	Parallel Processing
Metropolitan	3km	3K × 3K	9M × 9M	15sec	less then 1sec
Korean Peninsula	9km	4K × 4K	16M × 16M	56sec	1sec
East Asia	27km	17K × 17K	317M × 317M	111min 14sec	2min 22sec

메모리와 수행시간이 기하급수적으로 증가하는 문제가 있다. 이를 해결하기 위해 KTSM을 제안하여 메모리 공간과 행렬 곱셈의 연산 횟수를 획기적으로 줄이고, CUDA를 이용한 병렬처리로 수행속도를 15~56배 향상시켰다. 따라서, 제안한 방법을 빠른 처리를 요구하는 현업에 적용한다면, 예보 시간을 크게 단축할 수 있다.

향후 연구계획으로는 OI를 CUDA로 처리하는 과정에서 대량으로 발생하는 CPU와 GPU간의 데이터 전송시간을 단축시켜 병목 현상을 해결하고, 병렬화가 제외된 과정에 대해서도 병렬 처리를 수행하여 더욱 고속화된 자료동화 기법을 개발하고자 한다.

REFERENCE

[1] National Institute of Environmental Research, *A Study of Accuracy Improvement of Numerical Air Quality Forecasting Model (I)*, NIER-SP2015-064, 11-1480523-002327-01, 2015.

[2] S. Yu, Y. Koo, and H. Kwon, "Inverse Model Parameter Estimation Based on Sensitivity Analysis for Improvement of PM_{10} Forecasting," *Journal of Korea Multimedia Society*, Vol. 18, No. 7, pp. 886-894, 2015.

[3] K. Lee, S. Lee, and E. Kim, "Assessment of Global Air Quality Reanalysis and Its Impact as Chemical Boundary Conditions for a Local PM Modeling System," *Journal of Environmental Science International*, Vol. 25, No.7, pp. 1029-1042, 2016.

[4] R. Park, K. Han, C. Song, M. Park, S. Lee, and S. Hong, et al., "Current Status and Development of Modeling Techniques for Forecasting and Monitoring of Air Quality over East Asia," *Korea Society for Atmospheric Environment,* Vol. 29, No. 4, pp. 407-438, 2013.

[5] F. Bouttier and P. Courtier, *Data Assimilation Concepts and Methods, Meteorological Training Course Lecture Series*, European Centre for Medium-Range Weather Forecasts, Bershire, 2002.

[6] National Institute of Environmental Research, *A Study of Accuracy Improvement of Numerical Air Quality Forecasting Model(II)*, NIER-SP2016-064, 11-1480523-002720-01, 2016.

[7] G.P. Cressman, "An Operational Objective Analysis System," *Monthly Weather Review*, Vol. 87, No. 10, pp. 367-374, 1959.

[8] K. Ide, P. Courtier, M. Ghil, and A.C. Lorenc, "Unifined Notation for Data Assimilation: Operational, Sequential and Variational," *Journal of the Meteorological Society of Japan*, Vol. 75, No. 1, pp. 181-189, 1997.

[9] A.C. Lorenc, "Analysis Methods for Numerical Weather Prediction," *Quarterly Journal of the Royal Meteorological Society*, Vol. 112, Issue 474, pp. 1177-1194, 1986.

[10] R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, Vol. 82, Issue 1, pp. 35-45, 1960.

[11] P. Courtier, J.N. Thepaut and A. Hollingsworth, "A Strategy for Operational Implementation of 4D-Var, Using An Incremental Approach," *Quarterly Journal of the Royal Meteorological Society*, Vol. 120, Issue 519, pp. 1367-1387, 1994.

[12] E. Kalnay, *Atmospheric Modeling, Data Assimilation and Predictability,* Sigma Press, Seoul, 2012.

[13] S. Jason and K. Edword, *Compute Unified Device Architecture by Example : An Introduction to General-Purpose Graphics Processing unit Programming*, Addison-Wesley Professional, Boston, Massachusetts, 2011.

[14] D.B. Kirk and W.W. Hwu, *Programming Massively Parallel Processors : A Hands-on Approach,* Morgan Kaufmann Publishers, San Francisco, 2010.

[15] W.H. Press, *Numerical Recipes in C++*, Cambridge University Press, Cambridge, Cambridgeshire, 1992.

배 효 식

2003년 안양대학교 컴퓨터공학과 학사
2013년 안양대학교 컴퓨터공학과 석사
2016년 안양대학교 컴퓨터공학과 박사과정 수료

2003년~현재 ㈜넥스트소프트 SI사업부
관심분야 : 패턴인식, 영상처리, 병렬처리응용

유 숙 현

1999년 안양대학교 컴퓨터공학과 학사
2002년 안양대학교 컴퓨터공학과 석사
2011년 안양대학교 컴퓨터공학과 박사

2012년~현재 안양대학교 정보통신공학과 조교수
관심분야 : 패턴인식, 신경망, 영상처리, 병렬처리응용, 딥러닝

권 희 용

1983년 서울대학교 전자계산기공학과 학사
1985년 서울대학교 전자계산기공학과 석사
1993년 서울대학교 컴퓨터공학과 박사

1986년~1995년 한국통신 연구개발단 선임연구원
1995년~현재 안양대학교 컴퓨터공학과 교수
관심분야 : 패턴인식, 신경망, 영상처리, 병렬처리응용