韓國水資源學會論文集
第47卷 第11號·2014年 11月
pp. 1051~1060

J. Korea Water Resour. Assoc.
Vol. 47, No. 11:1051-1060, November 2014
http://dx.doi.org/10.3741/JKWRA.2014.47.11.1051
pISSN 1226-6280 • eISSN 2287-6138

MPI 기법을 이용한 병렬 홍수침수해석

Parallel Flood Inundation Analysis using MPI Technique

박 재 홍*
Park, Jae Hong

Abstract

This study is attempted to realize an improved computation performance by combining the MPI (Message Passing Interface) Technique, a standard model of the parallel programming in the distributed memory environment, with the DHM (Diffusion Hydrodynamic Model), a inundation analysis model. With parallelizing inundation model, it compared with the existing calculation method about the results of applications to complicate and required long computing time problems. In addition, it attempted to prove the capability to estimate inundation extent, depth and speed-up computing time due to the flooding in protected lowlands and to validate the applicability of the parallel model to the actual flooding analysis by simulating based on various inundation scenarios. To verify the model developed in this study, it was applied to a hypothetical two-dimensional protected land and a real flooding case, and then actually verified the applicability of this model. As a result of this application, this model shows that the improvement effectiveness of calculation time is better up to the maximum of about 41% to 48% in using multi cores than a single core based on the same accuracy. The flood analysis model using the parallel technique in this study can be used for calculating flooding water depth, flooding areas, propagation speed of flooding waves, etc. with a shorter runtime with applying multi cores, and is expected to be actually used for promptly predicting real time flood forecasting and for drawing flood risk maps etc.

keywords : MPI, parallel technique, 2-Dimensional analysis, inundation model, DHM

요 지

본 연구에서는 분산 메모리환경 병렬프로그래밍 모델의 표준인 MPI (Message Passing Interface) 기법과 침수해석 모형인 DHM (Diffusion Hydrodynamic Model) 모형을 연계하여 침수모형을 병렬화하고 기존의 기법으로 복잡하고 장시간의 계산시간을 요구하였던 계산에 대해 향상된 계산 성능을 구현하고자 하였다. 개발된 모형을 다양한 침수 시나리오를 바탕으로 가상유역과 실제유역에 대하여 코어 개수별로 모의함으로써 제내지 침수에 따른 침수범위 및 침수위의 추정, 및 계산시간 단축 효과를 입증 하고 병렬기법에 대한 홍수해석 분야의 적용성을 입증하고자 하였다. 본 연구에서 개발된 모형의 검증을 위하여 2차원 가상 제내지 및 실제 침수 사례에 대하여 적용하였고, 적용결과 동일한 정확도를 기준으로 계산시간 면에서 단일 코어와 비교하여 멀티코어를 사용한 경우 약 41~48%의 개선효과가 나타나는 것을 확인하였다. 본 연구에서 개발된 병렬해석 기법을 이용한 침수해석 모형은 멀티코어를 적용하여 짧은 계산시간으로 침수심, 침수구역, 홍수파 전달속도 등이 계산 가능하여, 실제 홍수 발생 시 침수지역에서의 신속한 예측 및 대처, 홍수위험지도 구축 등에 유용하게 이용될 수 있을 것으로 기대된다.

핵심용어 : MPI, 병렬해석, 2차원 홍수, 침수해석, 확산파 모형

* 세명대학교 토목공학과, 교수 (e-mail: jhpark@semyung.ac.kr, Tel: 82-43-649-1332)
Semyung University Civil Engineering, Jaecheon, Korea

1. 서 론

세계 인구 중의 28만 이상이 하천변 15 km 이내에 거주하고 있다고 알려져 있으며 홍수재해가 가장 빈번하게 발생하고 매우 피해가 극심한 자연재해라는 것은 알려진 일이다. 매년 평균적으로 90개국 이상에서 1억 9천 6백만명 이상이 홍수로 인한 일정한 규모의 재해를 경험하고 있다. 1980년부터 2000년까지 전 세계적으로 170,000명 이상의 사망자가 홍수로 인해 발생하였으며 매해 평균 거의 9,000명의 사망자가 발생하고 있다(UNDP, 2004). 비록 국가와 자치단체에 의해 홍수 조절과 관리를 위한 노력들이 경주되고 있으나 하천변이나 해안가에 거주하는 인구의 증가와 기후패턴의 변화에 따른 홍수 발생과 이에 따른 위험 관리의 어려움은 가중되고 있다. 이와 같이 홍수위험 경감은 지방, 국가 및 전지구적인 재해 관리기관의 당면한 주요한 도전과제가 되었으며 홍수 위험 경감을 위해 보다 발전된 홍수터 관리 및 홍수위험 관리기법과 제도의 도입이 필요한 시점이다(Bedient et al., 2008). 우리나라에서도 홍수와 같은 자연재해의 관리를 위해 자연재해 대책법의 통과와 더불어 국토부와 소방방재청에 의해 여러 가지 홍수 관리 및 피해저감을 위한 제도 도입 및 수방시설의 건설을 통해 홍수조절을 위한 노력이 이루어지고 있으나 매년 홍수로 인한 막대한 인적, 물적 피해를 입고 있는 실정이다.

홍수 위험관리 기법과 제도는 위험에 관한 효율적 대중교육, 홍수 크기의 판단 및 추정에 기초한 세밀히 계획된 홍수 발생 긴급상황 동안 대중과의 빠른 정보교환이 요구된다(FEMA 2002). 홍수 크기의 판단을 위해 정교하고 연산능력이 뛰어난 홍수예측 모형에 의한 정확하고 빠른 홍수 침수해석을 바탕으로 홍수심, 유속과 홍수 침수범위의 빠른 예측기술은 관리자로 하여금 제내지 침수 여부 파악 및 침수 범위 파악 등을 이른 시간에 판단 가능하게 하여 홍수 위험을 조기에 대처할 할 수 있도록 하여 홍수관리 기법들의 대안들을 설계하는데 효율적인 결과들을 적용할 수 있게 한다(Bates et al., 2004). 그러나 홍수관리에 필요한 수리인자들의 산정은 복잡하고 장시간의 연산을 요구하여 실제 홍수 발생시 필요한 정보를 적절한 시간에 제공하지 못하고 있다.

이에 본 연구에서는 홍수로 인한 제내지 범람이 발생할 경우 제내지 침수, 침수심, 도달 시간 등의 홍수 위험관리를 위한 핵심적 수리학적 인자 도출을 위해 빠른 연산시간으로 필요한 요소들을 산정할 수 있는 침수해석 프로그램을 개발하고자 하였다.

제내지 범람 해석에 관한 연구는 2차원 동수역학 방정식을 바탕으로 하나의 연산 장치를 이용하는 순차적 프로그램(Sequential Program)이 주로 연구되어 왔다. 그러나 순차적 프로그램을 이용한 2차원 해석기법을 이용할 경우 제내지 침수해석을 위한 수치해석 기법으로 많이 이용되는 양해적 기법은 모형을 실행하는데 여전히 오랜 연산 시간을 소요하고 있다. 특히 반복기법을 기반으로 한 불확실성 해석이나 시나리오를 바탕으로 하는 홍수 침수도 작성의 경우 계산적인 강도를 엄청나게 증가시킨다(Sayers et al., 2000). 2000 년대에 들어오면서 컴퓨터의 하드웨어와 소프트웨어의 발전과 더불어 2차원 모형에서 순차적 모형과 동일한 정확도를 유지하면서 전산해석의 연산부하를 줄이기 위해 병렬 프로그램(Parallel Program)기법을 도입한 2차원 해석모형의 개발이 선진국을 중심으로 이루지고 있다.

2차원 홍수해석을 위해 Rao (2005)의 RMA, Pau and Sanders (2006)에 의한 CalTWiMS, Villanueva and Wright (2006)에 의한 TRENT, Yu (2010)에 의한 FlooMap-Parallel 그리고 Neal et al. (2009)에 의한 LISFLOOD-FP, Hluchy 에 의한 Flo2DH를 포함한 고성능 병렬 연산 모형에서 도입되어 왔다. 최근에는 Monte-Carlo 기법에 기초한 불확실성 해석 및 시나리오에 기반한 홍수침수 지도 제작 등에 병렬기법을 이용하는 침수해석 모형도 개발되고 있다 (Di Baldassarre et al., 2010).

국내에서의 범람 홍수파에 대한 연구는 주로 댐 및 제방붕괴를 중심으로 1차원 해석이 수행되어 왔으며(Lee et al., 1995) 2차원 침수해석으로는 Han et al. (2006)이 제방붕괴로 인한 제내지 침수해석을 수행하였고 Park (2012)은 제방붕괴를 인한 제내지의 위험도 산정을 위한 기법을 개발하였으나 연산시간 개선 등을 위한 병렬기법을 도입한 2차원 수치해석 모형의 본격적인 개발은 이루어지지 못하고 있는 실정이다.

따라서 병렬프로그래밍 기법을 이용하여 2차원 침수해석 모형과 연계한 병렬화 모형을 개발해 기존의 기법으로 복잡하고 장시간의 계산시간을 요구하였던 2차원 제내지 침수해석을 본 모형에서는 기존의 순차적 모형보다 빠른 연산 능력을 가진 모형을 구현하고자 하였다. 개발된 병렬화 모형을 다양한 침수 시나리오를 바탕으로 가상유역과 실제유역에 대하여 코어 개수별로 모의함으로써 제내지 침수에 따른 침수범위 및 침수위의 추정, 예상 피해규모 및 계산시간 단축 효과를 입증 하고 병렬기법에 대한 홍수해석 분야의 적용성을 입증하고자 하였다.

2. 병렬해석 기법

최근의 컴퓨터 시스템들은 마이크로프로세서 발전 추세에 영향을 받아 왔으며 그에 따라 최근의 PC들은 거의 대부분 멀티코어(Multi-Core) CPU를 장착하고 있다. 이러한 멀티코어 CPU들을 장착한 PC들은 메인 메모리를 여러 CPU 코어들이 공유하여 계산을 한다. 그러므로 이러한 범용 PC들은 PC 자체가 하나의 작은 규모의 클러스터 시스템이라고 볼 수 있다. 이러한 멀티코어 CPU는 하드웨어적으로는 작은 규모의 클러스터가 구축되어 있어 높은 성능을 낼 수 있도록 구성되어 있지만 소프트웨어 역시 병렬화되지 않으면 기대되는 성능을 낼 수 없다. 이러한 클러스터 컴퓨터의 설계 성능을 달성하기 위해 MPI, OpenMP, CUDA 등 몇 가지 병렬 프로그래밍 기법이 주로 이용된다.

2.1 MPI

MPI (Message Passing Interface)는 분산 메모리 시스템을 사용하며 언어 독립적인 통신 프로토콜을 사용하는 병렬 프로그래밍 라이브러리이다. MPI는 각 계산이 할당된 코어(Rank) 사이에 점대 점(Point to Point) 또는 집합적 통신을 사용하여 정보를 교환하여 각 코어에서 자신에게 할당된 프로세스를 처리하여 병렬 컴퓨팅 환경을 구현한다. 현재 MPI는 Fortran, C, C++ 언어에 맞게 만들어진 MPI 서브루틴들을 이용하여 병렬 프로그래밍을 적용할 수 있도록 되어있으며 제한적으로 공유 메모리(Distributed Shared Memory) 개념을 포함하고 있다(Gropp et al., 1999).

MPI가 가지는 장점은 분산메모리 시스템에서 사용이 가능하다. 즉, 컴퓨터 네트워크를 이용한 클러스터에서 사용이 가능하다는 장점이 있다. 클러스터가 가지는 최대의 장점은 슈퍼컴퓨터에 비해서 성능향상을 꾀하기가 쉽다는 점이다. 장비의 교체를 통해서 성능향상을 해야 하는 슈퍼컴퓨터에 비해서, 클러스터는 내부에 노드(일반적으로 계산을 위해 사용되는 PC)를 추가함으로써 슈퍼컴퓨터에 비해서 손쉽게 클러스터의 성능을 향상시킬 수 있다는 장점이 있다.

2.2 OpenMP

OpenMP는 공유메모리(Shared Memory) 환경에서 프로그램을 병렬화하는 표준도구이다. OpenMP의 경우 OpenMP의 지시자 등을 통하여, 사용자가 지정한 구간에 대해서 병렬 실행을 한다. OpenMP의 실행은 병렬 실행을 할 구간을 만나게 되면, 사용가능한 스레드(Thread)들이 병렬적으로 해당 구간의 작업을 처리하게 된다. OpenMP는 MPI와 다르게 MPI는 각 프로세서를 하나의 작업을 처리하는 개체로 보지만, OpenMP의 경우 각 스레드들이 작업을 처리하는 개체로 본다. 현재 지원되는 언어는 MPI와 마찬가지로, C, C++, Fortran에서 지원된다(Chapman et al., 2008).

OpenMP는 사용자가 소스 코드 안에 삽입한 컴파일러 지시자를 이용하여 실행파일이 병렬 실행이 가능한 파일로 생성해준다. 또한 OpenMP는 직접 스레드를 생성하고, 관리하여 병렬프로그래밍을 더 쉽게 할 수 있도록 도와준다. OpenMP가 가지는 장점은 이와 같이 사용자가 손쉽게 병렬프로그래밍을 할 수 있도록 한다. 또한 공유 메모리 환경에서 실행되어 서로 데이터를 교환하는 과정을 통하여 데이터 동기화를 시킬 필요가 없다. MPI에 비해 요구되는 메모리의 양이 더 적다는 점과, MPI처럼 네트워크 통신을 이용하여 데이터 교환을 하지 않게 된다는 점에서 병렬화 구간에 따른 간접적 손실이 작다는 점이 장점이다.

2.3 CUDA

CUDA (Compute Unified Device Architecture)는 그래픽 처리 장치(GPU)에서 수행하는 병렬처리 알고리즘을 C 프로그래밍 언어를 비롯한 산업 표준 언어를 사용하여 작성할 수 있도록 하는 범용 GPU (General-Purpose computing on Graphics Processing Units, GPGPU)기술이다. CUDA는 엔비디아(nVidia)사에서 개발해오고 있으며 이 구성장치를 사용하려면 엔비디아 GPU와 특별한 스트림 처리 드라이버가 필요하다. CUDA는 일정규모 이상의 그래픽 장치에서 동작하며 CUDA GPU 안의 명령셋과 대용량 병렬 처리 메모리에 접근할 수 있도록 해 준다(Kirk et al., 2010).

그러나 CPU와는 달리 GPU는 병렬 다수 코어 구조를 가지고 있고, 각 코어는 수천 스레드를 동시에 실행시킬 수 있다. 응용 프로그램이 수행하는 작업(계산)이 이러한 병렬처리연산에 적합할 경우, GPU를 이용함으로써 커다란 성능 향상을 기대할 수 있으나 아직까지 범용화가 이루어지지 않고 있으며 수치해석 분야의 프로그램 작성이 용이하지 않은 실정이다.

본 연구에서는 클러스터 구축에 좀 더 강점이 있으며 현재 범용적으로 이용될 수 있고 향후 클러스터를 이용한 대규모 연산에 적합한 MPI를 이용하여 홍수 해석모형을 구축하고자 하였다.

Table 1. Characteristics and Types of Parallel Programming Techniques

	MPI	OpenMP	MPI+OpenMP	CUDA
Development Year	1994	1997	1997	2007
Development	MPI Forum	Architecture Review Board	-	nVidia
Feature	Distributed Memory	Shared Memory	Shared+Distributed	GPU
Parallel Method	Library	Compiler	Compiler+Library	Compiler
Portability	Excellent	Good	Good	Poor
Difficulty	High	Low	High	Very High

3. 홍수해석 모형

제방 붕괴로 인한 홍수파가 제내지로 전달되는 경우에 홍수터의 저류, 홍수의 감쇠, 건물주위에서의 흐름 등으로 인해 홍수파의 전달과정의 물리적인 현상을 수식으로 정확하게 나타내기에는 큰 어려움이 있다. 따라서 본 연구에서는 제방의 월류 및 붕괴에 따른 홍수파가 제내지로 전파되는 경우에 관성력의 항이 압력, 마찰력, 중력의 항과 비교하여 그 중요도가 작게 나타나는 물리적인 특성을 고려하여 2차원 천수방정식을 기본 식으로 하였으며, 이를 확산형 방정식으로 근사화한 홍수확산 모형을 가상 및 실제유역에 적용하였다(Hromadka, 1987).

2차원 천수방정식은 Eqs. (1)~(3)과 같은 연속방정식과 운동량방정식으로 구성된다.

$$\frac{\partial h}{\partial t}+\frac{\partial q_x}{\partial x}+\frac{\partial q_y}{\partial y}=0 \quad (1)$$

$$\frac{\partial q_x}{\partial t}+\frac{\partial}{\partial x}\left(\frac{q_x^2}{H}\right)+\frac{\partial}{\partial y}\left(\frac{q_x q_y}{H}\right)+\mathrm{g}\,H\left(S_{fx}+\frac{\partial h}{\partial x}\right)=0 \quad (2)$$

$$\frac{\partial q_y}{\partial t}+\frac{\partial}{\partial y}\left(\frac{q_y^2}{H}\right)+\frac{\partial}{\partial x}\left(\frac{q_x q_y}{H}\right)+\mathrm{g}\,H\left(S_{fy}+\frac{\partial h}{\partial y}\right)=0 \quad (3)$$

여기서, S_{fx}, S_{fy}는 x, y 방향에 대한 마찰경사, H는 수심, h는 수위, ∂는 중력가속도, q_x, q_y는 x, y에 대한 단위폭당 유량을 나타내고 있다.

Eqs. (1)~(3)에 대해 확산파 이론을 적용하여 근사화된 공식은 Eq. (4)와 같다.

$$\frac{\partial}{\partial x}F_x\frac{\partial h}{x}+\frac{\partial}{\partial y}F_y\frac{\partial h}{\partial y}=\frac{\partial h}{\partial t} \quad (4)$$

확산형 방정식을 해석하기 위해 Fig. 1에서와 같이 양해석법의 중앙차분법을 변형한 격자망 계산방법을 사용하며, 직사각형 격자로 구분하였고 각 격자에 대한 표고, 조도계수, 위치 등을 분포시켰다. 흐름해석은 유량을 계산하기 위하여 흐름의 수심치를 해석하여 격자 시스템 내에서 계산될 수 있다(Han et al., 2006).

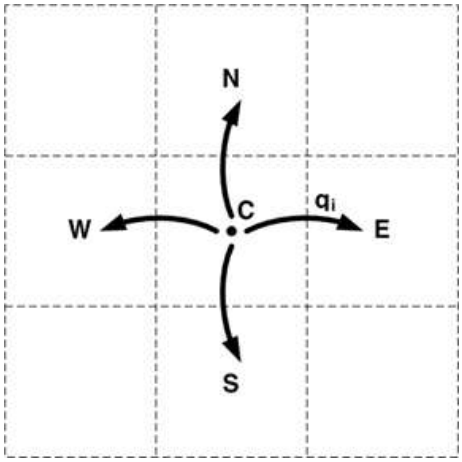

Fig. 1. The Finite Difference Network in the Diffusion Wave Model

도입된 확산파 2차원해석 프로그램을 MPI 기법을 이용하여 병렬화하였다. 병렬화 개념은 MPI를 이용하여 병렬화할 때 일반적으로 이용되는 Master-Slave 기법을 이용하였다. 이 기법은 Master라고 불리는 하나의 프로세스가 다른 나머지의 작업들을 조정하는 책임을 가진다. 이러한 메카니즘은 특히 나머지 Slave 프로세스가 서로 통신을 할 수 없을 때와 각 Slave들이 수행해야 할 작업의 양을 예측하기 힘들 때 적당하다.

본 프로그램에서는 Master에서는 각 Slave에게 작업영역을 할당하고 계산시간 단계에서 Slave에게 할당된 영역의 요소(Cell)의 수위를 넘겨주어 각 Slave에서 넘겨받은 수위를 이용하여 할당된 요소별 유량을 계산하였고 계산된 유량 및 개선된 수위는 다시 Master로 넘겨주어 전체 영역에 대해 개선된 수위는 Master에서 수집되어 전체 영역에 대해 수위에 대한 허용오차를 검증하였다.이에 대한 알고리즘은 Fig. 2와 같다. 이와 같이 확산파 해석모형은 MPI와 결합되어 병렬 프로그램이 구성되었고 이를 이용하여 가상유역 및 실제유역에 대해 개발된 모형의 적

용성을 검증하였다.

4. 가상 제내지 침수에 대한 개발 모형의 적용

본 연구에서 개발한 모형의 제내지 침수해석 모의 적용 가능성을 검토하기 위하여 가상의 제내지를 격자망으로 구성하였으며, 계산시간 개선 효과를 입증하기 위하여 단일코어(Single -Core)와 멀티코어(Multi-Core)를 각각 적용하였다. 본 연구에서는 병렬 프로그램을 작성하기 위해 Master-Slaver 기법을 이용하였기 때문에 Master 코어는 작업지시 및 분배의 역할을 하고 실제 연산에는 참여치 않아 2-Core의 사용은 생략되었고 병렬해석으로 3, 4-Core만 이용되었다.

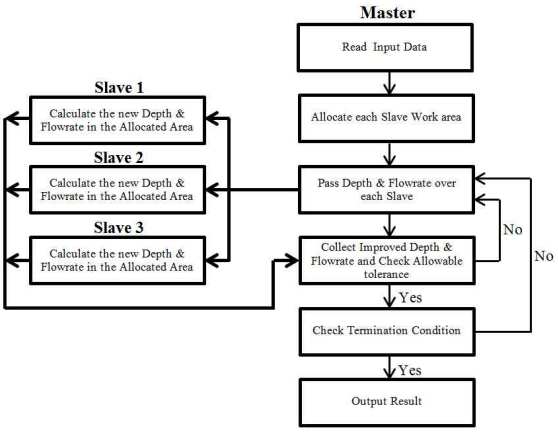

Fig. 2. Flowchart about Master-Slave Algorithm in this Model

격자망의 구성은 Fig. 3과 같이 Element 450, 4,500 × 2,250 m로 이루어진 2차원 제내지 구간을 고려하였다. 붕괴지점은 Element 45지점에서 붕괴가 발생되어 제내지로 붕괴유량이 유입되도록 가정하였으며, 유입량은 최대 15,000 m^3/sec 적용하였다.

계산결과를 침수심도, 유속 벡터도를 통하여 나타내었으며 Figs. 4~9와 같다.

Figs. 4~6은 계산시간 5, 10, 15 hr에서 단일코어와 멀티코어를 이용하여 해석하였을 경우 제내지에서의 침수심을 나타내었다. 그림들에서 나타난 바와 같이 단일코어와 멀티코어에서의 침수심 및 침수양상이 거의 동일하게 나타나고 있다. Figs. 7~9는 침수심과 동일한 시간에서 침수 유속벡터도를 나타내었다. 유속벡터 또한 침수심도와 마찬가지로 단일코어와 멀티코어에서 동일한 결과를

Breach Point

450	435	420	405	390	375	360	345	330	315	300	285	270	255	240	225	210	195	180	165	150	135	120	105	90	75	60	45	30	15
449	434	419	404	389	374	359	344	329	314	299	284	269	254	239	244	209	194	179	184	149	134	119	104	89	74	59	44	29	14
448	433	418	403	388	373	358	343	328	313	298	283	268	253	238	263	208	193	178	203	148	133	118	103	88	73	58	43	28	13
447	432	417	402	387	372	357	342	327	312	297	282	267	252	237	282	207	192	177	222	147	132	117	102	87	72	57	42	27	12
446	431	416	401	386	371	356	341	326	311	296	281	266	251	236	301	206	191	176	241	146	131	116	101	86	71	56	41	26	11
445	430	415	400	385	370	355	340	325	310	295	280	265	250	235	320	205	190	175	260	145	130	115	100	85	70	55	40	25	10
444	429	414	399	384	369	354	339	324	309	294	279	264	249	234	339	204	189	174	279	144	129	114	99	84	69	54	39	24	9
443	428	413	398	383	368	353	338	323	308	293	278	263	248	233	358	203	188	173	298	143	128	113	98	83	68	53	38	23	8
442	427	412	397	382	367	352	337	322	307	292	277	262	247	232	377	202	187	172	317	142	127	112	97	82	67	52	37	22	7
441	426	411	396	381	366	351	336	321	306	291	276	261	246	231	396	201	186	171	336	141	126	111	96	81	66	51	36	21	6
440	425	410	395	380	365	350	335	320	305	290	275	260	245	230	415	200	185	170	355	140	125	110	95	80	65	50	35	20	5
439	424	409	394	379	364	349	334	319	304	289	274	259	244	229	434	199	184	169	374	139	124	109	94	79	64	49	34	19	4
438	423	408	393	378	363	348	333	318	303	288	273	258	243	228	453	198	183	168	393	138	123	108	93	78	63	48	33	18	3
437	422	407	392	377	362	347	332	317	302	287	272	257	242	227	472	197	182	167	412	137	122	107	92	77	62	47	32	17	2
436	421	406	391	376	361	346	331	316	301	286	271	256	241	226	491	196	181	166	431	136	121	106	91	76	61	46	31	16	1

Fig. 3. The Protected Lowland Composed of the Element 450

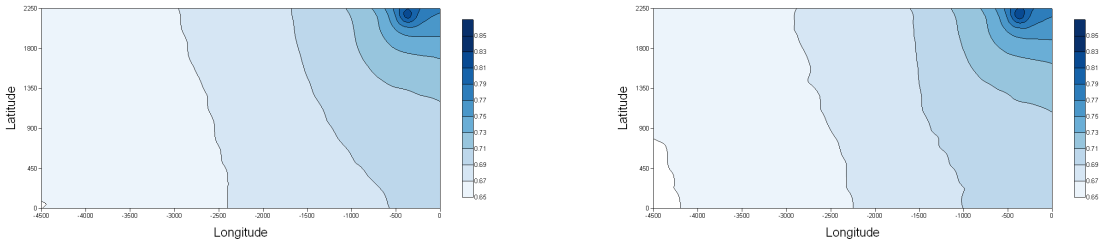

Fig. 4. The Flooding Depth for the Calculation Time, t=5 hr, using 1-Core and 4-Core

나타냄을 알 수 있다.

Tables 2 and 3은 가상 제내지 침수에 대한 프로세서의 코어 개수 별 수심차이를 나타내었다. Table 2에서 보이듯이 각 지점에서의 침수심은 오차범위 내에서 침수심을 계산하였고 단일코어와 비교하여 요소의 개수에 상관없이 모든 적용에서 유사한 침수심을 얻을 수 있었다. Table 3은 단일코어를 기준으로 침수심의 차이에 대한 비를 산정하였다. 해석 과정내의 반복 작업에서 단일코어 또한 오차를 포함하고 있지만 멀티코어의 상대적 차이를 파악코자 하였다. 임의의 지점인 요소 100에서의 오차율

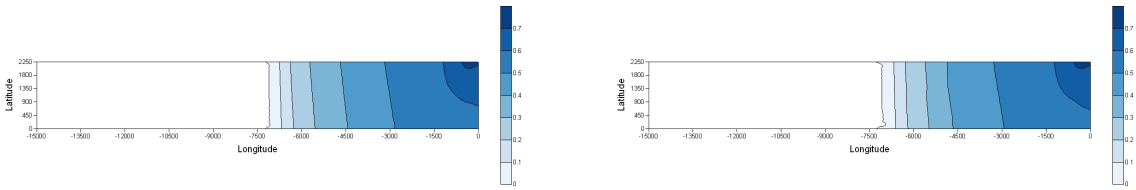

Fig. 5. The Flooding Depth for the Calculation Time, t=10 hr, using 1-Core and 4-Core

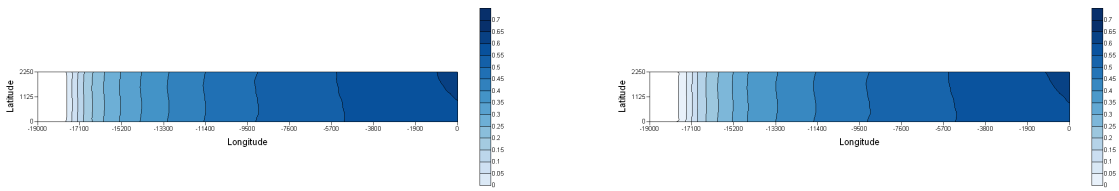

Fig. 6. The Flooding Depth for the Calculation Time, t=15 hr, using 1-Core and 4-Core

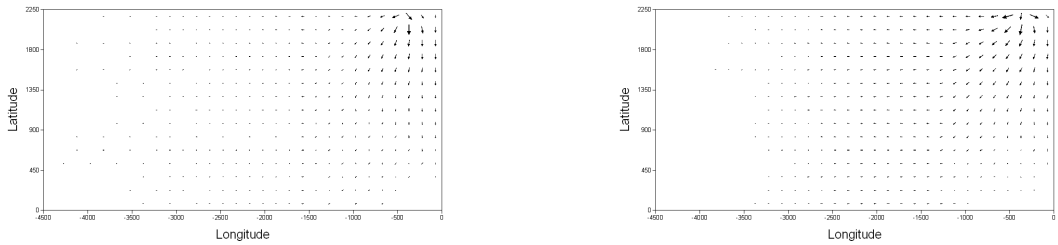

Fig. 7. The Degree of Flux Vector for the Calculation time, t=5 hr, using 1-Core and 4-Core

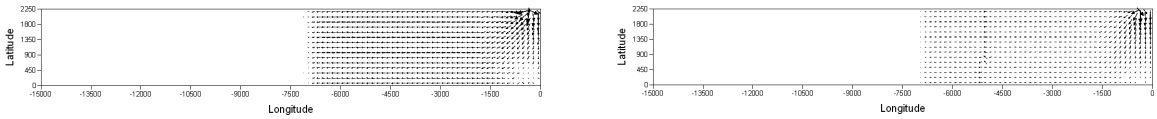

Fig. 8. The Degree of Flux Vector for the Calculation time, t=10 hr, using 1-Core and 4-Core

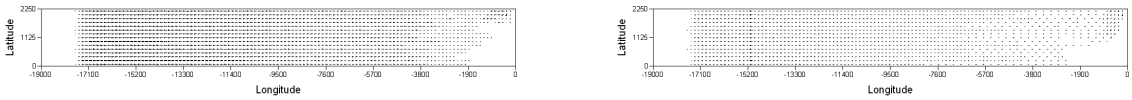

Fig. 9. The Degree of Flux Vector for the Calculation time, t=15 hr, using 1-Core and 4-Core

Table 2. Comparison of Flooding Depths according to the Number of Cores (Time-5.00)

Depth (m)	Element 450	Element 1500	Element 3000	Element 4500
1-Core	0.702	0.602	0.602	0.602
3-Core	0.704	0.602	0.602	0.602
4-Core	0.705	0.603	0.602	0.602

Fig. 11. The studied Basin in this Model, Gamcheon in Gimcheon City

은 단일 코어와 비교하여 4-Core에서 최대 0.4% 수심차를 확인할 수 있었으나 이는 허용범위내의 오차로 반복과정에서 발생하는 차이로 판단되었다.

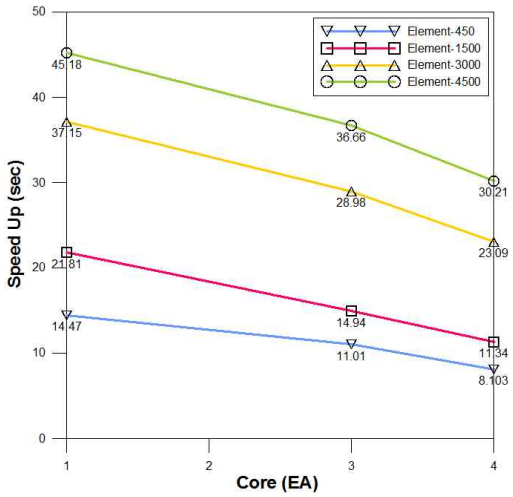

Fig. 10. Speed-up Trend with the Number of Cores

Table 4는 멀티코어를 이용하였을 경우 병렬화로 인한 연산속도의 개선효과를 확인코자 하였다. 요소의 개수별 멀티코어의 사용으로 연산시간을 확인하였다. 표에서 보이듯이 단일코어 및 멀티코어를 각각 적용한 결과, 단일코어만을 사용한 경우보다 멀티코어를 사용한 경우, 4-Core를 사용할 경우 요소의 수가 450, 1,500, 3,000, 45,000일 경우 연산시간이 8.1, 11.3, 23.1, 30.2초로 나타나 단일코어를 사용할 때보다 계산시간의 개선 효과가 약 41% 정도 개선되는 것을 알 수 있다.

Fig. 10 코어 사용에 따른 연산시간 개선효과를 나타내기 위해 그래프로 코어별 연산시간을 나타내었다. 그림에서 보여지듯이 요소의 수가 많을수록 속도 감소율이 증가하고 있으며 코어의 수 역시 증가할수록 연산속도가 증가하여 시간 감소율이 커지는 것을 알 수 있다. 이와 같은 결론은 코어를 좀 더 증가할수록 연산 속도의 효과는 더욱 더 커질 수 있다는 것과 또한 좀 더 큰 제내지 유역에 적용할 경우 병렬화의 효과가 크게 나타날 수 있다는 것을 알 수 있었다.

Table 3. Depth Difference Rate according to the Number of Cores

Depth Difference (%)	Element 450	Element 1500	Element 3000	Element 4500
1-Core	0.0	0.0	0.0	0.0
3-Core	0.285	0.0	0.0	0.0
4-Core	0.427	0.166	0.0	0.0

Table 4. Execution Time according to the Number of Cores

Time (sec) / Element #	1-Core	3-Core	4-Core
450	14.471	11.006	8.103
1500	21.806	14.943	11.341
3000	37.154	28.982	23.087
4500	45.177	36.665	30.208

가상 제내지 적용을 통해 본 연구에서 개발된 모형은 실제 유역에 대한 적용성이 클 것으로 판단되었고 제내지가 클수록 코어의 수가 많을수록 좀 더 확실한 연산 속도 증가 효과가 기대되어 졌다.

5. 실제 제내지 침수에 대한 개발 모형의 적용

본 연구에서 개발한 연구모형을 2002년 08월 30일~09월 01일 기간 동안 태풍 "루사"로 인하여 침수 사례가 있는 김천시 개령면 감천에 적용하였다(Fig. 11). 감천 유역의 제내지를 격자망으로 구성하였으며, 계산시간 개선 효과를 입증하기 위하여 단일코어와 멀티코어를 각각 적용하였다(Park, 2012).

격자망의 구성은 50 m 간격으로 이루어진 2차원 격자로 구성하였고, 요소의 수는 1,823개이다. 제방 붕괴로 인한 제내지로의 유입유량은 감천 유역의 1차원 동수역학 해석을 통해 추정된 유량을 감안하여 최고 유량 1,500 m^3/s이 유입되도록 적용하였다.

계산결과를 계산시간 15 hr에 대해 침수심도, 유속벡터도를 나타내었으며 Figs. 12~13과 같다. 가상제내지 적용의 경우와 마찬가지로 단일코어와 멀티코어의 결과에서도 침수심, 침수양상 및 유속벡터 등이 동일하게 나타났다.

Tables 5 and 6은 실제 제내지 침수에 대한 프로세서의 코어 개수 별 수심과 오차율을 나타내었다. 50 m 격자망의 경우 요소 900지점을 기준으로 하였으며, 요소 900지점의 수심에 대해 단일코어와 멀티코어를 적용한 결과에 대하여 비교하였으며, 결과의 일치성과 오차율을 확인하였다.

비교결과 침수심은 비교지점에서 0.005 m의 차이를 나타내어 0.00008 %에 해당하는 차이로 거의 같은 값을 나타내었고 이러한 차이는 모형의 반복과정중의 허용범위 내의 오차로 인해 발생된 것으로 판단된다.

Fig. 12. The Flooding Depth for the Calculation Time, t=15 hr, using 1-Core and 3-Core

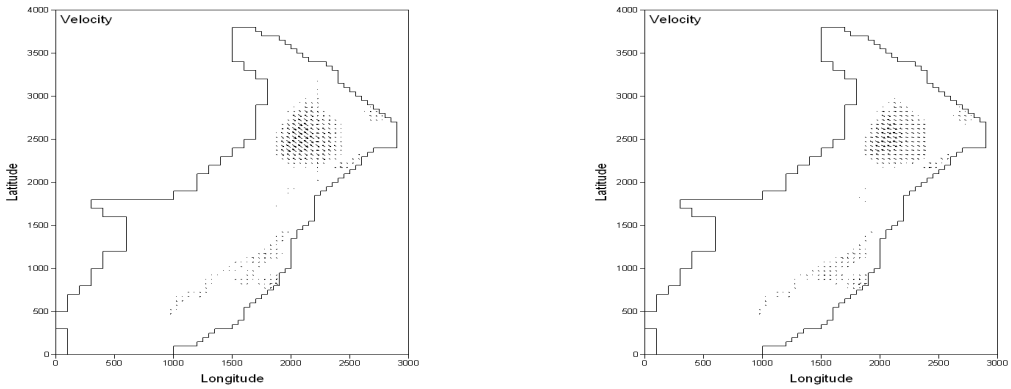

Fig. 13. The Degree of Flux Vector for the Calculation Time, t=15 hr, using 1-Core and 3-Core

Table 6은 실제 제내지에 대한 격자망에 따른 프로세서의 코어 개수 별 계산시간의 개선효과를 나타내었다. 본 모형의 적용 시 단일코어 및 멀티코어를 각각 3차례 적용하여 보다 분명한 결과를 얻고자 하였으며 그 결과, 단일코어만을 사용한 경우보다 멀티코어를 사용한 경우 본 적용의 경우 계산시간의 개선효과가 1-Core의 경우 623.3초, 4-Core일 경우 328.1초로 나타나 최고 48% 정도 개선되는 것을 알 수 있다.

Table 5. Comparison of Flooding Depth by the Number of Cores (Time-5.00)

Element 100	Depth (m)	Depth Difference Rate (%)
1-Core	0.590	0.0
3-Core	0.590	0.0
4-Core	0.595	0.00008

Table 6. Execution Time according to the Number of Cores in Gamcheon

Runtime (sec)	1-Core	3-Core	4-Core
Simulation 1	623.342	409.766	328.097
Simulation 2	623.033	407.958	328.811
Simulation 3	623.740	409.303	328.553

가상 제내지의 경우 연산 시간의 개선효과는 41% 정도 되는 것으로 나타났으나 실제 유역의 경우 48% 정도 개선되어 연산속도 증가 효과가 뚜렷하게 나타났다. 이는 실제 유역의 경우 바닥이 평평한 가상 제내지보다 각 요소에서 수렴속도가 빨라 가상 제내지보다 자료의 전송량이 적은 것으로 판단된다. 이는 각 연산단계에서 각 요소의 계산된 결과의 수렴여부를 판단하기 위해 Slave 코어에서 Master 코어로 계산된 결과의 전송이 필요한데 이때 자료 전송량과 비교적 작게 발생하는 실제유역에서 계산시간 단축효과가 크게 나타나는 것으로 보인다.

각 코어 개수별 적용 시 가상 제내지 및 실제 유역에서도 침수심, 유속 등이 단일코어와 동일한 정확도를 유지하면서 코어 개수가 증가할수록 연산속도의 증가가 나타나 다 차원 해석 및 연산속도가 오래 걸리는 제내지의 경우 병렬화된 프로그램을 이용하여 침수해석을 실시할 경우 연산속도의 뚜렷한 개선이 기대할 수 있을 것으로 판단되었다.

6. 결 론

1) 본 연구에서는 기존에 개발되었던 2차원 침수해석 모형의 구동에 있어 제약이 되어온 방대한 계산시간을 개선하고자, 분산 메모리환경 병렬프로그래밍 모델의 표준인 MPI 기법과 2차원 천수방정식을 기본식으로 하여 이를 확산형 방정식으로 근사화한 침수해석 모형 DHM을 연계하여, 병렬화된 침수해석 모형을 개발하였고 이를 가상제내지 및 실제 제내지에 적용하여 개발된 모형의 적용성을 검토하였다.
2) 본 연구에서 개발된 모형을 검증을 위하여 2차원 가상 제내지에 대하여 적용하였으며, 본 모형의 적용결과 단일코어와 동일한 정확도를 유지하면서 뚜렷한 연산속도의 증가가 나타나 실제 유역의 적용시 짧은 연산시간으로 만족할 만한 연산 결과를 얻을 수 있음을 입증하였다.
3) 병렬화된 침수모형을 태풍 "루사"로 인하여 침수된 김천시 개령면 감천에 적용하였으며, 침수심도 및 유속 벡터도 등에서 정확한 결과를 나타내었고 특히 계산시간은 48%의 개선효과를 얻을 수 있었다.
4) 본 연구에서 개발된 병렬해석 기법을 이용한 침수해석 모형은 멀티코어를 적용하여 단시간에 효율적, 안정적으로 계산함으로써 제내지 침수에 따른 침수범위를 예측하고 실시간 제내지 범람 예·경보 분야에 적용하여 침수지역에서의 신속한 사고현황 파악 및 대처, 사회적, 경제적 비용 절감에 도움이 될 수 있을 것으로 기대된다.

References

Bates, P.D., Horritt, M.S., Aronica, G., and Beven, K. (2004.) "Bayesian updating of flood inundation likelihoods conditioned on flood extent data." *Hydrological Processes*, Vol. 18, pp. 3347-3370.

Chapman, B.M., Jost, G., and R. van der Pas. (2008). *Using OpenMP, Portable shared Memory Parallel Programming,* The MIT Press Cambridge, Massachusetts London, England.

David, B.K., and Wen-mei, W.H. (2010). Programming Massively Parallel Processors, A Hands-on Approach.

Di Baldassarri, G., Schumann, G., Bates, P.D., Freer, J.E., and Beven, K. (2010). "Flood plain mapping, a critical discussion of deterministic and probabilistic approaches." *Hydrological Sciences Journal*, Vol. 55, No. 3, pp. 364-376.

Federal Emergency Management Agency FEMA (2002). National Flood Insurance Program, Program Description. Federal Emergency Management Agency/Federal Insurance Mitigation Administration, Available at <http://www.fema.gov/library/viewRecord.do?id=1480>.

Gropp Lusk and Skjellum (1999). Using MPI, Second edition, MPI Press.

Han, K.Y., and Park, J.H. (2006). Development of Urban Inundation Analysis Model, Urban Flood Disaster Management Research Center.

Han, K.Y., Lee, J.T., and Lee, W.H. (1985). "An Analysis of Outflow Hydrograph Resulting from an Earth Dam-Break." *Journal of* KSCE, Vol. 5. No. 2, pp. 41-50.

Hluchy, L., Froehlich, D., Tran, V.D., Astalos, J., Dobrucky, M., and Nguyen, G.T. (2002). *Parallel numerical solution for flood modeling systems. in Sloot P.M.A, Hoekstra A.G., Tan C.J.K., and Dongarra, J.J. (Eds.)*, Computational Science ICCS 2002, International Conference, Krakow, Poland, Proceedings, Part 1, Lecture Notes in Computer Science, Vol. 2329, Springer-Verlag, Berlin Heidelberg New York, pp. 543-551.

Hromadka, T.V., and Yen, C.-C. (1987). *A Diffusion Hydrodynamic Model,* Water Resources Investigations Report 87-4137, US Geological Survey.

Lee, J.T., Han, K.Y., and Park, J.H. (1995). Floodwave Modeling in Inundated Area Resulting from Levee-Break, KOSEF-931-1200-024-2.

Neal, J.C., Fewtrell, T.J., and Trigg, M.A. (2009). *Parallelisation of storage cell flood models using OpenMP.* Environmental Modeling & Software, Vol. 24, pp. 872-877.

Park, J.H. (2012). "Development of Technique to Estimate Inundation Hazard Level Caused by River Levee Failure." *Journal of KOSHAM*, Vol. 11. No. 5, pp. 259-268.

Pau, J.C., and Sanders, B.F. (2006). "Performance of parallel implementations of an explicit finite volume shallows-water model." *Journal of Computing in Civil Engineering,* Vol. 20, No. 2, pp. 99-110.

Rao, P. (2005). "A parallel RMA2 model for simulating large-scale free surface flows." *Environmental Modeling & Software*, Vol. 20, No. 1, pp. 53-57.

Sayers, P.B., Hall, J.W., and Meadowcroft, I.C. (2000). "Towards risk-based flood hazard management in the UK, Floods-a new approach." *Special Issue One, Proceedings of the Institute of Civil Engineers*, Vol. 150, pp. 36-42.

Villanueva, I., and Wright, N.G. (2006). "Linking Riemann and storage cell models for flood prediction. Proceedings of the Institution of Civil Engineers." *Water Management*, Vol. 159, No. 1, pp. 27-33.

Yu, D. (2010). "Parallelization of a two-dimensional flood inundation model based on domain decomposition." *Environmental Modelling & Software*, Vol. 25, pp. 935-945.

paper number : 14-048
Received : 17 June 2014
Revised : 7 September 2014 / 13 October 2014
Accepted : 13 October 2014