<학술논문> DOI http://dx.doi.org/10.3795/KSME-B.2015.39.2.191 ISSN 1226-4881(Print)
2288-5324(Online)

분산 메모리 시스템에서 압력방정식의 해법을 위한 MPI와 Hybrid 병렬 기법의 비교

전 병 진* · 최 형 권**†
* 서울과학기술대학교 에너지환경대학원 에너지시스템공학과,
** 서울과학기술대학교 기계자동차공학과

Comparison of Message Passing Interface and Hybrid Programming Models to Solve Pressure Equation in Distributed Memory System

Byoung Jin Jeon* and Hyoung Gwon Choi**†
* Dept. of Energy System, Graduate School of Energy and Environment, Seoul Nat'l Univ. of Science and Technology
** Dept. of Mechanical/Automotive Engineering, Seoul Nat'l Univ. of Science and Technology.

(Received August 21, 2014 ; Revised November 1, 2014 ; Accepted November 3, 2014)

Key Words: Distributed Memory System(분산 메모리 시스템), Bi-Conjugate Gradient(이중공액구배법), Hybrid Parallel Model(하이브리드 병렬 모델), Message Passing Interface(MPI), OpenMP Directives(OpenMP 지시어)

초록: 본 연구에서는 분산 메모리시스템에서의 압력 방정식의 병렬해법을 위하여 MPI(Message Passing Interface)와 하이브리드 병렬기법을 사용하였다. 두 모델은 영역분할 기법을 활용하며, 하이브리드 기법은 성능이 양호한 두 가지 영역분할에 대해 수행하였다. 두 병렬기법의 성능을 비교하기 위해서 다양한 문제 크기에 대해 최대 96개의 쓰레드를 사용하여 속도향상을 측정하였다. 병렬 성능은 캐쉬 메모리에 따른 문제의 크기 및 MPI 통신, OpenMP 지시어의 부하에 대해 영향을 받음을 확인하였다. 문제의 크기가 작은 경우에는 쓰레드가 증가할수록 MPI 통신 및 OpenMP 지시어 부하에 대한 비율이 상대적으로 크기 때문에 병렬 성능이 좋지 않으며, MPI 통신 부하보다는 OpenMP 지시어 부하가 상대적으로 크므로 MPI 병렬 기법의 병렬 성능이 더 우수하다. 문제의 크기가 큰 경우에는 캐쉬 메모리의 활용도가 높고 MPI 통신 및 OpenMP 지시어 부하에 대한 비율이 낮아 병렬 성능이 좋으며, OpenMP 지시어보다 MPI 통신에 의한 부하가 더 지배적이어서 하이브리드 병렬 성능이 MPI 병렬 성능보다 더 양호하다.

Abstract: The message passing interface (MPI) and hybrid programming models for the parallel computation of a pressure equation were compared in a distributed memory system. Both models were based on domain decomposition, and two numbers of the sub-domain were selected by considering the efficiency of the hybrid model. The parallel performances for various problem sizes were measured using up to 96 threads. It was found that in addition to the cache-memory size, the overhead of the MPI communication/OpenMP directives affected the parallel performance. For small problems, the parallel performance was low because the percentage of the overhead of the MPI communication/OpenMP directives increased as the number of threads increased, and MPI was better than the hybrid model because it had a smaller communication overhead. For large problems, the parallel performance was high because, in addition to the cache effect, the percentage of the communication overhead was relatively low compared to that for small problems, and the hybrid model was better than MPI because the communication overhead of MPI was more dominant than that of the OpenMP directives in the hybrid model.

1. 서 론

최근 다양한 산업 분야에서는 수치해석을 이용

† Corresponding Author, hgchoi@seoultech.ac.kr

한 연구가 많이 진행되고 있다. 일반적으저로 산업 분야에서 풀고자하는 문제는 복잡한 형상들로 구성되어 있거나 보다 더 정확한 결과를 얻기 위해 많은 격자 수가 요구된다. 이러한 문제들을 푸는 경우에는 매우 많은 계산시간이 소요된다. 따라서, 빠른 계산 결과를 얻기 위해 사용하는 병렬 컴퓨팅에 대한 관심은 매우 높아지고 있다.

병렬 컴퓨팅은 크게 분산 메모리와 공유 메모리 구조로 나누어진다. 분산 메모리 구조는 2개 이상의 노드가 인피니밴드 등의 네트워크로 연결되어 있는 시스템을 의미한다. 각 노드는 독립적으로 계산을 수행하지만 네트워크를 통해 데이터를 주고 받거나 마스터 노드로 모으는 등의 작업을 수행해야 하는 경우에는 MPI(Message Passing Interface) 라이브러리[1]가 필요하다. 이와 같은 병렬 방법은 계산에 참여하는 노드가 늘어나면서 병렬 성능이 증가한다. 강성우[2] 등은 자동차 주위의 유동을 풀기 위해 50개의 노드를 이용하여 50배 이상의 속도 향상을 얻을 수 있었다. 그러나, 문제 크기에 비해 계산 노드를 너무 많이 사용하면 데이터 전송에 따른 통신부하 또는 동기화에 대한 비율이 증가하여 병렬 성능이 저하되는 단점이 있다. 공유 메모리 구조는 하나의 메모리를 공유하는 멀티 코어 기반의 시스템을 의미한다. 이와 같은 시스템에서는 단순 반복 실행 영역에 OpenMP(Open Multi- Processing)[3]와 같은 지시어 기반의 라이브러리를 적용하여 병렬 계산을 수행한다. OpenMP를 이용한 알고리즘의 병렬화는 MPI 방식에 비해 매우 쉽고 통신부하가 없는 장점이 있으나 반복되는 fork-join 작업과 공유 메모리 구조에 의한 캐쉬 일관성 문제 등이 발생하게 된다.

앞에서 설명한 바와 같이 MPI와 OpenMP 병렬 방식은 서로 다른 장, 단점을 가지고 있다. 따라서, 많은 연구자들은 각각의 병렬 방식의 장점을 취한 하이브리드 병렬 방식에 대해서 관심을 갖고 공유메모리 또는 분산메모리 형태의 클러스터에서 성능 테스트를 수행하였다.

일반적으로 공유메모리 기반의 클러스터에서 하이브리드 병렬 방식을 사용하는 경우에는 하나의 메인보드에 2개 이상의 멀티코어 CPU가 장착되어있다. 이 때 CPU 간의 병렬화는 MPI 병렬 방식을 이용하고 하나의 CPU 내의 각 코어간의 병렬화는 OpenMP 병렬 방식을 사용한다. Rabenseifner, R.[4]는 공유메모리 기반의 클러스터에서 하이브리드 병렬 방식의 특성을 파악하기 위해 데이터 전송에 중요한 요인 중 하나인 대역폭을 측정하고 MPI와 하이브리드 병렬 방식에 대한 부하 시간을 비교하였다. 전병진 등[5]은 공유메모리 기반의 클러스터에서 MPI, OpenMP, 하이브리드 병렬 방식을 적용한 이중공액구배법(Bi-Conjugate Gradient) 알고리즘을 이용하여 다양한 문제 크기에 대해서 병렬 성능 테스트를 수행하였다. 그들은 특정 문제 크기에 대해서 하이브리드 병렬 방식이 다른 병렬 방식보다 병렬 성능이 높게 나타나는 것을 확인하였으며, 병렬 성능은 해석 문제의 크기뿐만 아니라 캐쉬 메모리에 의해서도 매우 큰 영향을 받는다고 보고하였다.

한편, 분산메모리 기반의 클러스터에서 하이브리드 병렬 방식을 사용하는 경우에는 멀티 코어로 구성되어 있는 2개 이상의 노드가 인피니밴드 등의 네트워크로 연결되어 있는 시스템에서 주로 사용된다. 병렬화에 의한 노드 간의 데이터 전송은 메시지 패싱 모델인 MPI 병렬 방식을 사용한다. 그리고, 각 노드 내에서는 공유메모리 기반인 OpenMP 병렬 방식을 적용한다. Robert 등[6]은 분산 메모리 형태의 클러스터에서 비대칭 고유값 문제를 풀기위한 QR 알고리즘의 속도 향상을 위해 하이브리드 병렬 방식을 적용하였다. 또한 특정 문제에 대해서는 널리 사용되는 ScaLAPACK 코드[7]와 비교했을 때 하이브리드 병렬 방식의 성능이 높음을 확인하였다. 이와 같이 연구자들은 다양한 문제에 대해서 계산속도 향상을 위해 하이브리드 병렬 방식을 적용하였다.

본 연구에서는 전병진 등[5]의 선행연구를 확장하여 인피니밴드로 구성된 분산 메모리 형태의 클러스터에서 MPI 병렬 방식과 Hybrid 병렬 방식의 특성을 파악하기 위해, 압력방정식을 풀기위한 이중공액구배법의 병렬연산을 수행하고자 한다.

2. 병렬 컴퓨팅 시스템의 구성

본 연구에서는 병렬 기법과 문제 크기에 따른 압력방정식의 병렬 성능을 파악하기 위해서 CentOS 6.4를 운영체제로 사용하는 분산메모리 형태의 클러스터 시스템에서 계산을 수행하였다. 컴파일러는 Intel(R) Fortran Composer XE 2013 for linux[9]를 사용하였다. Fig. 1(a)는 본 연구에서 사용한 분산메모리 방식의 클러스터 시스템 구조를 나타

낸다. 컴파일 및 작업 지시는 1번 계산 노드에서 수행하며, 각 계산 노드는 병렬 계산을 위해 인피니밴드 기반의 네트워크 시스템으로 구성되어 있다. Fig. 1(b)는 하나의 노드에 대한 개략도를 나타낸다. 각 계산 노드의 사양은 Intel(R) Xeon CPU E5-2680 V2 제품 2개와 64GB의 메인 메모리로 구성되어 있다. Intel(R) Xeon CPU E5-2680 V2 제품은 클럭 수가 2.80GHz인 쓰레드가 10개로 구성되어 있으며, 각 CPU는 25MB의 캐쉬 메모리를 공유하여 사용한다.

3. 영역분할 및 병렬화 기법

본 연구에서는 이산화된 압력방정식의 해법을 위하여 예조건화된 이중공액구배법(Preconitioned Bi-Conjugate Gradient)을 사용하였으며, 예조건화 기법으로는 대각 예조건화(Diagonal Preconditioner)를 사용하였다.[2] 압력방정식의 해법을 위한 예조건화된(Preconditioned) 이중공액구배법(Bi-Conjugate Gradient) 알고리즘은 수렴할 때까지 다음 식들을 반복하게 된다. 여기서 하첨자 j는 반복 계산 횟수를 의미한다.

$$\alpha_j = \left(\{z_j\}^T\{r_j\}\right)/\left(\{d_j\}^T[A]\{d_j\}\right) \quad (1)$$

$$\{x_{j+1}\} = \{x_j\} + \alpha_j\{d_j\} \quad (2)$$

$$\{r_{j+1}\} = \{r_j\} - \alpha_j[A]\{d_j\} \quad (3)$$

$$\{z_{j+1}\} = [M]^{-1}\{r_{j+1}\} \quad (4)$$

$$\beta_j = \left(\{z_{j+1}\}^T\{r_{j+1}\}\right)/\left(\{z_j\}^T\{r_j\}\right) \quad (5)$$

$$\{d_{j+1}\} = \{z_{j+1}\} + \beta_j\{d_j\} \quad (6)$$

MPI 병렬화를 적용할 때 국소행렬 및 국소벡터는 분할된 영역에서 독립적으로 정의되므로 식 (2), (3), (6)은 각 코어에서 독립적으로 수행된다. 식 (1), (5)는 전체 계산 영역에 대한 값이므로 각 코어에서 독립적으로 계산을 수행한 후 그 값을 마스터 코어에 다시 모아서 최종적으로 구하게 된다. 값을 다시 모으는 과정에서 네트워크 부하가 발생하게 된다. 식 (4)에서 예조건화 벡터를

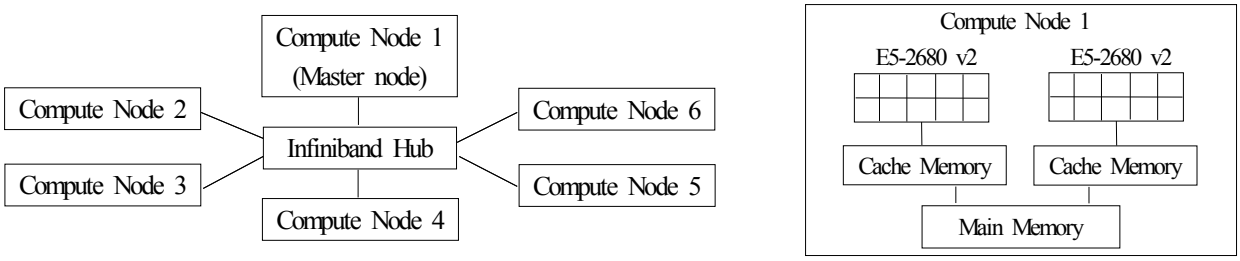


Fig. 1 Structure of parallel computing system based on distributed memory

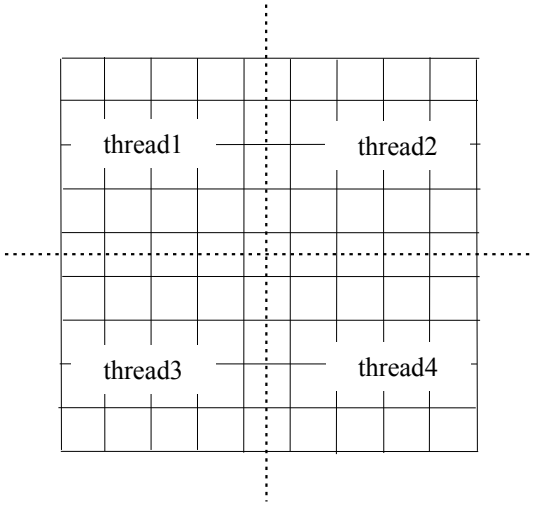


Fig. 2 Decomposition of computational domain[2,5]

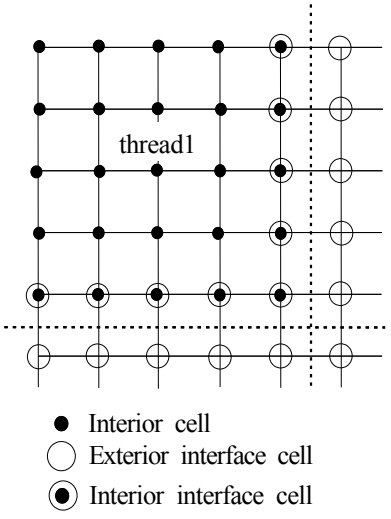
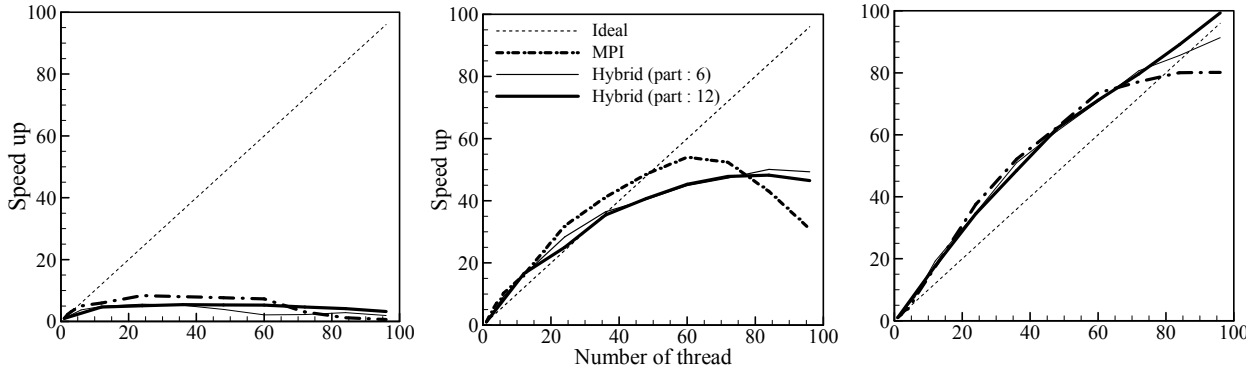


Fig. 3 Classification of cells in a sub-domain[2,5]



(a) N=23,000 (b) N=288,305 (c) N=1,472,000

Fig. 4 Comparison of parallel performances of Bi-CG algorithm for the three meshes

구하는 방법은 대각 예조건화 기법을 이용하였다. $[M_n]=Diag[A_n]$이고 $[M_n]^{-1}$은 각 국소영역 내부의 값으로 구성된 대각 성분들의 역수로 각 코어에서 독립적으로 수행할 수 있다.

다음은 2가지의 병렬 연산 기법에 대하여 간략히 서술하고자 한다.

3.1 영역분할과 MPI를 이용한 병렬화

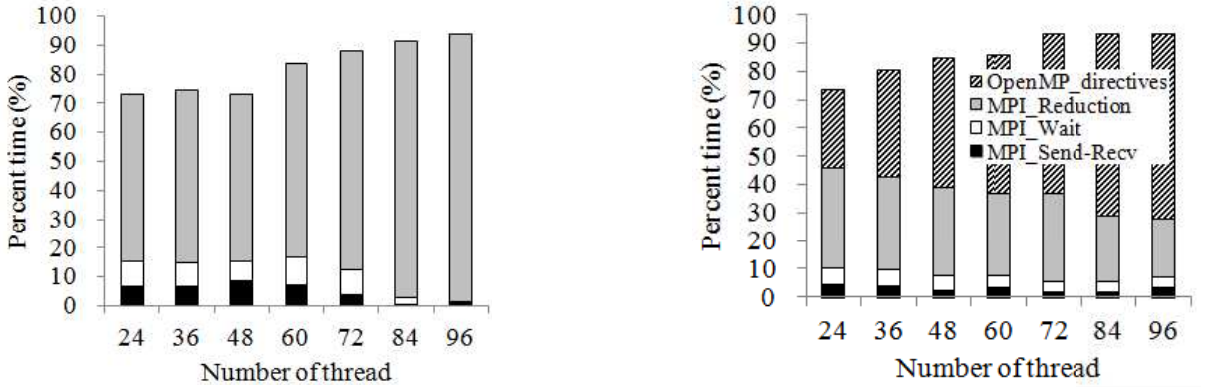
정렬 격자계 또는 비정렬 격자계 기반의 문제에 대해서 효율적인 병렬화를 수행하기 위해서는 영역분할 알고리즘인 METIS[10] 라이브러리가 필요하다. METIS 라이브러리를 이용하면 Fig. 2와 같이 1개의 영역을 독립적인 4개의 영역으로 나눌 수 있다. 분할되어진 각 영역은 Fig. 3과 같이 내부 셀, 내부 경계 셀, 외부 경계 셀로 구성된다. 내부 셀은 각 쓰레드가 독립적으로 계산하며, 경계 셀들은 계산하기 전에 주위 쓰레드와 값의 교환을 필요로 한다. 값의 교환을 위한 각 쓰레드간의 통신은 MPI 라이브러리를 이용하며, 하나의 노드로 구성되어 있지 않은 쓰레드들은 인피니밴드 네트워크를 통해 이루어진다. 분할된 각 영역에서는 내부 셀과 내부 경계 셀에 대해서만 계산하며, 각 영역에서 국소행렬이 구성될 때 외부 경계 셀에 대한 행은 무시된다.[2,5]

3.2 하이브리드 병렬 기법을 이용한 병렬화

서론에서 설명한 대로 하이브리드 병렬 기법은 MPI와 OpenMP 병렬 기법을 혼합하여 사용하는 것을 의미한다. 본 연구에서는 6개와 12개로 영역 분할된 격자를 이용하여 하이브리드 병렬 기법에 대한 성능 테스트를 수행하였다. 격자를 6개의 영역으로 분할한 이유는 계산 노드의 개수가 6개인 것을 기준으로 두었기 때문이다. 12개의 영역으로 분할한 경우는 CPU의 개수가 12개인 것을 기준으로 두었기 때문이다. 하이브리드 병렬 기법이 적용된 코드는 각 영역의 경계에 위치하는 셀의 정보가 MPI에 의해서 교환이 이루어지며, 독립적으로 계산을 수행하는 내부 셀은 OpenMP 병렬 기법이 적용되었다.

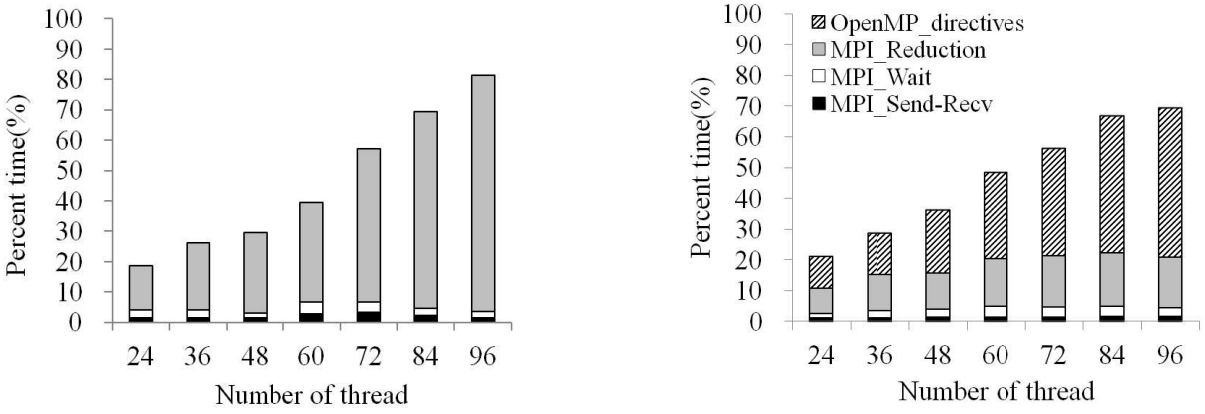
3. 해석 결과 및 토의

본 연구에서는 6개의 노드로 구성된 분산 메모리 기반 클러스터에서 다양한 셀의 개수를 가지는 격자계들에 대해 최대 96개의 쓰레드를 사용하여 MPI, 하이브리드 방식의 병렬 성능을 측정하였다. Fig. 4는 3가지 격자계에 대해서 2가지 병렬기법을 적용한 Bi-CG 알고리즘의 병렬 성능을 측정한 결과를 보여준다. 그림에서 표기한 Part는 하이브리드 병렬 기법의 경우에 사용한 영역 분할 개수를 나타낸다. 예를 들어서, 쓰레드를 96개 사용한 경우 영역 분할 개수가 12개이면, 각 영역에서 적용된 OpenMP의 수는 8개이다. Fig. 5, 6, 7은 셀의 수가 23,000개, 288,305개, 1,472,000개인 격자계에 대해서 병렬 계산을 수행할 때 발생하는 부하를 전체시간에 대한 비율로 나타낸 것이다. Fig. 4(a)는 셀의 개수가 23,000개일 때 MPI, 하이브리드 병렬 기법에 대해서 쓰레드 수에 따른 속도 향상 결과를 나타낸 그림이다. MPI 병렬 기법만을 적용하여 병렬 계산을 수행한 경우에는 쓰레드의 개수가 24개일 때 최대 10배 정도의 속도 향상을 보여준다. 그러나, 쓰레드의 개수를 24개 이상으로 증가시키면 병렬 성능이 저하되는 것을 볼 수 있다. 셀의 개수가



(a) MPI (b) Hybrid(Part : 12)

Fig. 5 Overhead of the MPI communications and OpenMP directives for coarse grid (cell : 23,000)

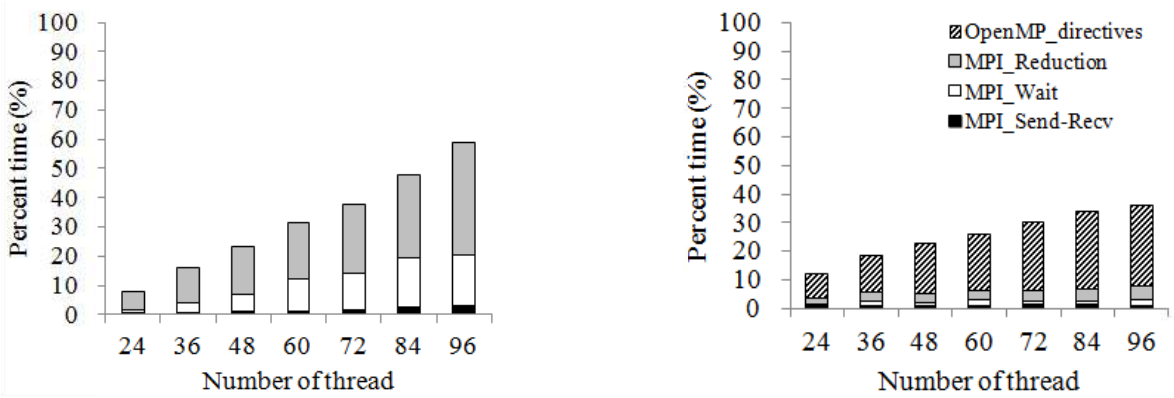


(a) MPI (b) Hybrid(Part : 12)

Fig. 6 Overhead of the MPI communications and OpenMP directives for mid grid (cell : 288,305)

23,000개인 경우에 6개의 계산 노드에서 나누어 계산을 수행하면 한 노드에 할당되는 계산량이 매우 적어지면서 Fig. 5(a)에서 보여주는 바와 같이 전체 계산 시간에 대한 병렬 부하 비율이 매우 크고 캐쉬 메모리의 성능을 최대로 사용하지 못하여 병렬 성능이 떨어지게 된다. 72개 이상의 쓰레드를 사용하는 경우에는 MPI_Reduction 명령어에 의한 부하가 급격히 증가하면서 병렬 성능도 급격히 낮아지는 것을 볼 수 있다. 하이브리드 병렬 기법을 이용한 병렬 계산은 MPI 병렬 기법만을 적용한 경우보다 최대 성능이 낮게 나오는 것을 볼 수 있다. 그 이유는 Fig. 5(b)의 결과와 같이 MPI 병렬 기법에 의한 부하에 OpenMP 지시어에 의한 부하가 더해지면서 계산 시간에 대한 부하율이 MPI 병렬 기법만을 적용한 경우보다 상대적으로 크기 때문이다.

Fig. 4(b)는 셀의 개수가 288,305개일 때 MPI, 하이브리드 병렬 기법에 대해서 쓰레드 수에 따른 속도 향상 결과를 나타낸 그림이다. MPI 병렬 기법만을 적용하여 병렬 계산을 수행한 경우에는 최대 48개 쓰레드까지 이상적인 병렬 성능보다 더 높은 성능을 보여주다가 그 이후에는 병렬 성능이 급격히 떨어지는 것을 볼 수 있다. 셀의 개수가 288,305개인 경우에 적은 쓰레드 사용 시 높은 병렬 성능이 나오는 이유는 각 쓰레드에 주어지는 계산량이 충분하고 캐쉬 메모리를 효율적으로 사용하기 때문이다. 그러나, 계산에 참여하는 쓰레드 수가 70개 이상이 되면 Fig. 6(a)의 결과와 같이 각 쓰레드에 주어지는 계산량보다 MPI 통신 부하에 대한 비중이 상대적으로 커져서 병렬 성능이 떨어지게 된다. 하이브리드 병렬 기법을 이용한 계산은 영역 분할 개수에 대한 차이가 거의 없으며, 최대 36개 쓰레드까지 이상적인 병렬 성능보다 더 높다가 쓰레드를 더 늘리게 되면 병렬 성능이 조금씩 증가하는 것을 알 수 있다. 70개 이상의 쓰레드를 사용하는 경우에는



(a) MPI (b) Hybrid(Part : 12)

Fig. 7 Overhead of the MPI communications and OpenMP directives for fine grid (cell : 1,472,000)

Fig. 6(b)의 하이브리드 경우의 부하보다 Fig. 6(a)의 MPI 통신 부하가 상대적으로 크기 때문에 하이브리드 병렬 기법이 MPI 병렬 기법보다 병렬 성능이 더 높게 나타난다.

Fig. 4(c)는 셀의 개수가 1,472,000개일 때 MPI, 하이브리드 병렬 기법에 대해서 쓰레드 수에 따른 속도 향상 결과를 나타낸 그림이다. MPI 병렬 기법만을 적용하여 병렬 계산을 수행한 경우에는 최대 70개 쓰레드까지 이상적인 병렬 성능보다 더 높은 성능을 보여주다가 그 이후에는 병렬 성능 증가 폭이 작아지는 것을 볼 수 있다. 그 이유는 Fig. 7(a)의 결과에서 보여주듯이 쓰레드를 60개 이상 사용한 경우에 MPI_Redcution과 MPI_Wait 명령어에 의한 부하가 커지면서 전체 계산시간에 대한 부하 비율이 상승하기 때문이다. 반면, 하이브리드 병렬 기법을 적용한 경우의 부하 비율을 나타낸 Fig. 7(b)를 보면 계산에 참여하는 쓰레드 수가 증가할수록 OpenMP 지시어에 의한 부하는 증가하지만 전체 수행시간에 대한 부하 비율은 MPI 병렬 기법만을 사용한 경우보다 적은 것을 볼 수 있다. 따라서, 하이브리드 병렬기법을 적용했을 때 더 높은 병렬 성능을 보여줬다. 추가적으로 영역을 12개로 분할한 경우에는 6개보다 OpenMP 지시어에 대한 부하가 상대적으로 적다. 영역을 12개로 분할했을 때 하이브리드 병렬 기법을 적용한 경우에는 최대 100배, 영역을 6개로 분할한 경우에는 92배 상승하였다.

4. 결 론

본 연구에서는 압력방정식의 대표적인 해법인 이중공액구배법의 병렬화를 MPI와 하이브리드 기법을 적용하여 수행하고, 다양한 문제 크기에 대하여 두 기법의 성능을 비교하였다.

(1) 병렬 성능은 캐쉬 메모리에 따른 문제의 크기 및 MPI 통신, OpenMP 지시어의 부하에 대해 영향을 받음을 확인하였다.

(2) 문제의 크기가 작은 경우에는 큰 경우보다 계산에 참여하는 쓰레드가 증가할수록 전체 계산 수행에 대한 MPI 통신 및 OpenMP 지시어 부하에 대한 비율이 상대적으로 크기 때문에 병렬 성능이 좋지 않다. 그리고, MPI 통신 부하보다 OpenMP 지시어 부하가 상대적으로 크기 때문에 하이브리드보다는 MPI 병렬 기법만을 사용한 경우가 더 나은 병렬 성능을 보인다.

(3) 문제의 크기가 큰 경우에는 캐쉬 메모리의 활용도가 높으며 전체 수행 시간에 대한 MPI 통신 및 OpenMP 지시어 부하에 대한 비율이 낮으므로 이상적인 병렬 성능보다 더 좋은 성능을 보여준다. 그리고, OpenMP 지시어보다 MPI_Reduction, MPI_Wait 명령어에 의한 부하가 지배적이므로 하이브리드 병렬 기법이 MPI 병렬 기법보다 더 좋은 병렬 성능을 보여준다.

후 기

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구 개발사업의 일환으로 수행하였음.[10044910, 다중의료영상을 활용한 3차원 초정밀 시뮬레이션 기반 심·혈관 질환 진단-치료지원 통합소프트웨어 시스템 개발]

참고문헌
(References)

(1) http://mvapich.cse.ohio-state.edu

(2) Kang, S. W., Choi, H. G. and Yoo, J. Y., 2002, "Parallelized Dynamic Large Eddy Simulation of Turbulent Flow Around a Vehicle Model," *Proceedings of the KSME 2002 Spring Annual Meeting*, pp. 1562~1567.

(3) http://www.open-mpi.org

(4) Rabenseifner, R., 2003, "Hybrid Parallel Programming: Performance Problems and Chances," *Proceedings of the 45th CUG Conference 2003,* pp. 1~11.

(5) Jeon, B. J., Lee, J. R., Yoon, H. Y. and Choi, H. G., 2014, "Performance Analysis of the Parallel CUPID Code for Various Parallel Programming Models in Symmetric Multi-Processing System," *Trans. Korean Soc. Mech. Eng. B*., Vol. 38, pp. 71~79.

(6) Robert, G., Bo, K. and Daniel, K., 2010, "A Novel Parallel QR Algorithm for Hybrid Distributed Memory HPC Systems", *SIAM J. Sci. Comput,* Vol. 32, pp. 2345~2378.

(7) http://www.netlib.org/scalapack/slug/

(8) KAERI, 2013, CUPID code 1.7 manual, Vol. 1, pp. 46~53.

(9) http://software.intel.com/en-us/fortran-compilers/

(10) http://www-users.cs.umm.edu/~karypis/metis