

# Optimization of Parallel Code for Noise Prediction in an Axial Fan Using MPI One-Sided Communication

Oh-Kyoung Kwon<sup>†</sup> · Keuntae Park<sup>\*\*</sup> · Haecheon Choi<sup>\*\*\*</sup>

## ABSTRACT

Recently, noise reduction in an axial fan producing the small pressure rise and large flow rate, which is one type of turbomachine, is recognized as essential. This study describes the design and optimization techniques of MPI parallel program to simulate the flow-induced noise in the axial fan. In order to simulate the code using 100 million number of grids for flow and 70,000 points for noise sources, we parallelize it using the 2D domain decomposition. However, when it is involved many computing cores, it is getting slower because of MPI communication overhead among nodes, especially for the noise simulation. Thus, it is adopted the one-sided communication to reduce the overhead of MPI communication. Moreover, the allocated memory and communication between cores are optimized, thereby improving 2.97x compared to the original one. Finally, it is achieved 12x and 6x faster using 6,144 and 128 computing cores of KISTI Tachyon2 than using 256 and 16 computing cores for the flow and noise simulations, respectively.

**Keywords :** MPI, One-Sided Communication, Axial Fan, High Performance Computing, Supercomputing

## MPI 일방향통신을 이용한 축류 팬 주위 소음해석 병렬프로그램 최적화

권오경<sup>†</sup> · 박근태<sup>\*\*</sup> · 최해천<sup>\*\*\*</sup>

## 요약

축류 팬(axial fan)은 팬이 회전하면서 작은 압력 상승을 만들어 다량의 공기를 불어주는 유체 기계로서 최근 축류 팬의 소음 저감이 중요하게 인식되고 있다. 본 연구는 팬 주위의 유동 소음을 해석하는 MPI 병렬프로그램 방법 및 최적화 기법에 대해 다룬다. 이때 수억 개 이상의 격자에서 수만 포인트의 소음원을 해석하기 위해서 2차원 도메인 분할 방법을 사용해서 MPI 병렬화를 하였다. 이때 대규모 계산 시 MPI 프로세스 간의 통신이 많이 발생하여 성능이 심각하게 느려지는 현상이 발생한다. 이를 극복하기 위해 MPI 일방향 통신을 적용하였다. 뿐만 아니라 통신 및 메모리 최적화 방법을 통해 최대 2.97배 향상시켰다. 마지막으로 KISTI 타키온2 슈퍼컴퓨터를 활용하여 전체 시뮬레이션 실험에서 유동 계산 시 6,144코어에서 최대 12배, 소음 계산 시 128코어에서 최대 6배의 성능향상을 달성하였다.

**키워드 :** MPI, 일방향통신, 축류 팬, 고성능컴퓨팅, 슈퍼컴퓨팅

## 1. 서론

축류 팬(axial fan)은 팬이 회전하면서 작은 압력 상승을

만들어 다량의 공기를 축 방향으로 불어주는 유체 기계이다 (Fig. 1). 선풍기를 비롯한 일상 생활용품에서부터 대형 공조기와 같은 산업 시설에 이르기까지 냉각이나 환기를 목적으로 다양한 분야에서 널리 사용된다. 특히 가전제품에서 발생하는 열을 식히기 위한 용도로 많이 사용된다.

한편 축류 팬 설계에서 소음은 고려해야 할 중요한 인자 중에 하나이다. 가정이나 사무실에서 생활의 편의를 위해 도입된 많은 기계 장치들에 부착된 팬이 소음을 유발하여 오히려 쾌적한 환경을 해치고 있다. 이러한 이유로 우리나라에서는 2015년부터 가전제품에 소음표시제를 시행하고 있으며 기업들도 저소음 제품을 개발하고 있다. 하지만 실험으로는 축류 팬 주위 3차원 유동의 비정상 특성이나 소음의 전파 특성을 분석하기 어렵다. 실험의 한계를 극복하기 위

\* 이 논문은 (2017년도) 한국과학기술정보연구원 슈퍼컴퓨팅센터의 슈퍼컴퓨팅 자원과 기술지원(KSC-2016-C3-0040) 그리고 한국과학기술정보연구원(KISTI) 주요사업, 한국에너지기술평가원(No. 20152020105600)과 미래창조과학부(NRF-2014M3C1B1033980)의 지원을 받아 수행되었음.

\*\* 이 논문은 2017년도 한국정보처리학회 추계학술발표대회에서 "MPI 일방향 통신을 이용한 축류 팬 수치해석 병렬프로그램 최적화"의 제목으로 발표된 논문을 확장한 것임.

† 정 회 원 : 한국과학기술정보연구원 선임연구원

\*\* 비 회 원 : 서울대학교 기계항공공학부 박사과정

\*\*\* 비 회 원 : 서울대학교 기계항공공학부 교수

Manuscript Received : January 15, 2018

Accepted : January 21, 2018

\* Corresponding Author : Oh-Kyoung Kwon(okkwon@kisti.re.kr)

해 수치해석 프로그램을 통한 연구가 활발하게 이루어진다. 따라서 본 연구는 팬 주위의 유동 소음(flow-induced noise) 해석을 위한 고병렬 프로그램의 최적화 기법에 대해 다룬다.

프로그램은 비관성-원통형 좌표계에서 Navier-Stokes 방정식과 연속 방정식을 계산하여 유동을 해석한 후, 유동 정보를 바탕으로 소음을 별도로 해석한다. 축류 팬 해석을 위해 목표로 하는 격자 수준은 약 8억 개이며, 소음 해석을 위한 소음 발생 포인트의 개수는 약 7만 개이다. 이를 계산하기 위해서는 수백 기가바이트 이상의 메모리와 수천 CPU 코어를 가진 슈퍼컴퓨터가 필요하다. 해당 계산을 분산메모리 환경에서 계산하기 위해 MPI 병렬화를 구현하였다.

특히 본 연구에서는 팬 소음 해석 시 메모리 사용량 및 각 계산 노드 간 통신 횟수의 급격한 증가로 인해 통신이 멈추는 현상이 발생하여 MPI의 일방향 통신(one-sided communication) 방법을 적용하였다[1]. 그리고 노드 간 병렬 확장성이 급격하게 나빠져서 메모리 사용량을 최적화 하고, 통신 횟수를 줄여 성능을 향상시켰다.

본 논문의 내용은 다음과 같다. 먼저 축류 팬 해석 프로그램으로 큰 코어를 사용하는 경우에 발생하는 문제점을 자세히 살펴보고, MPI 일방향 통신을 사용하여 해결한 방법을 제시하고 실험 결과를 나타내기로 한다.



Fig. 1. Examples of Axial Fans

## 2. 축류 팬 해석 병렬 프로그램

본 절에서는 축류 팬 해석 프로그램을 살펴보고, 유동 소음 해석 시 발생하는 문제점을 살펴본다. 먼저 유동 해석 부분에 대해 소개한다. 축류 팬 주위 유동은 높은 레이놀즈 수를 갖는 난류 유동에 해당하여 큰 크기에서부터 매우 작은 크기에 이르기까지 다양한 크기의 유동 구조를 갖는다. 다양한 크기의 유동 구조를 모사하기 위해서는 조밀한 격자가 필요하며, 레이놀즈 수가 높을수록 요구 격자의 수준이 증가한다. 실제 축류 팬이 작동하는 레이놀즈 수는  $O(10^6)$ 로 해석을 위해 수억 개의 격자를 사용하고 있다[2, 3]. 이러한 거대규모의 축류 팬 해석 프로그램은 단일 계산 노드에서는 메모리 부족으로 계산이 불가능하다. 이에 분산메모리 환경의 수천 CPU 코어의 슈퍼컴퓨터에서 병렬확장성을 확보할 수 있게 [4]의 연구처럼 MPI를 이용해 Fig. 2처럼 2차원 영역 분할하였다. 각 프로세스로 분할된 메모리 영역간 통신에 MPI를 사용하여 병렬화 하였다.

축류 팬 해석을 위한 지배방정식은 연속 방정식과 Navier-Stokes 방정식이며[5], 팬과 함께 회전하는 비관성-원통형

좌표계를 사용하여 좌표계는  $x, r, \theta$ 이며 해석에 사용한 축류 팬의 형상과 계산 영역 그리고 경계 조건 등은 Fig. 3과 같다. 해석은 Akselvoll & Moin [6]이 제안한 준 내재적 분할 단계 방법(semi-implicit fractional step method)을 사용하였다. 전체 계산 영역을 반경 방향에 대해서 두 개의 영역(core and outer regions)으로 나누어 안쪽에서는 원주 방향 미분 항에 대해 TDMA(tridiagonal matrix algorithm)[7]를, 바깥쪽에서는 반경 방향 미분 항에 대해 TDMA를 수행하여 내재적으로 계산하게 된다. 따라서 MPI로 TDMA를 풀기 위해 안쪽에서는  $x, r$  방향으로, 바깥쪽에서는  $x, \theta$  방향으로 영역을 분할하였다. 반면에 연속 방정식을 만족시키기 위해 필요한 Poisson 방정식에서 FFT(fast Fourier transform)[8]는 안쪽과 바깥쪽 두 영역 모두에 대해서 원주 방향으로 수행하여  $x$ 와  $r$  방향으로 영역을 분할하였다.

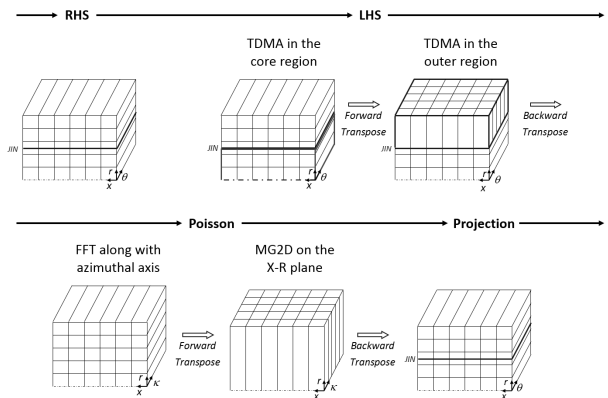


Fig. 2. Schematic Diagram of 2D Domain Decomposition of Parallel Program

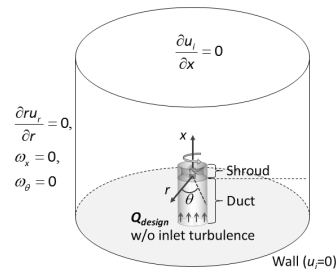


Fig. 3. Schematic Diagram of Coordinates, Computational Domain, and Boundary Conditions

큰 도메인 문제를 2차원 영역 분할하여 각 계산 노드에서는 작은 도메인만을 담당하게 되어 병렬성이 확보된다. 하지만 도메인을 분할하였기 때문에 전체 도메인에 대한 계산이 필요한 TDMA와 FFT에서는 행렬 전치(matrix transpose)를 반드시 해야 한다. 행렬 전치 시 전체 노드 간의 데이터 교환이 필요하므로 MPI 집합통신인 MPI\_Alltoall 함수를 사용하는 데 많은 통신 부하가 발생한다. 따라서 MPI 프로세스 간의 인접 통신만 필요한 다른 수치계산 방법에 비해 병렬확장성이 떨어진다. 이를 위해  $x$ 와  $r$  방향에 대해 새로운 통신 커뮤니케이터(communicator)인 MPI\_COMM\_X와

MPI\_COMM\_WORLD를 정의해 분할된 방향에 대해서만 통신이 이루어지게 설계하였다. 행렬 전치 시에는 MPI\_COMM\_WORLD의  $\Theta$ 방향에 대한 통신 커뮤니케이터 역할을 하게 된다.

유동 해석이 끝나면 해당 유동 정보를 바탕으로 소음을 예측하는 음향학적 상사이론(acoustic analogy)을 이용하였다. 축류 팬과 같이 움직이는 음원을 고려한 Ffowcs Williams-Hawkings 방정식[9]을 이용하여 각 소음 발생 포인트들에 대해서 계산하였다. 본 연구에서는 해당 방정식을 이용하여 축류 팬 표면에서 발생하는 소음만을 고려하여 표면을 구성하는 메쉬(mesh)들이 소음 발생 포인트에 해당한다. 이때 한 포인트에서 발생하는 소음을 공간에 대해서 2차 정확도로 계산하기 위해서는 해당 위치의 주위 속도 36개와 압력 8개 포인트 정보가 필요하다[10]. 계산 시 필요한 주위의 정보는 도메인이 2차원으로 분할되어 있기 때문에 MPI 통신을 통해 전달 받아야 한다. 이를 위해 Fig. 4와 같이 소음 해석을 위한 별도의 통신 커뮤니케이터인 MPI\_COMM\_SURF를 정의하였다. 각 포인트에서 발생하는 통신은 MPI\_COMM\_SURF내에서만 발생하여 다른 포인트와의 통신들과 분리가 가능하다.

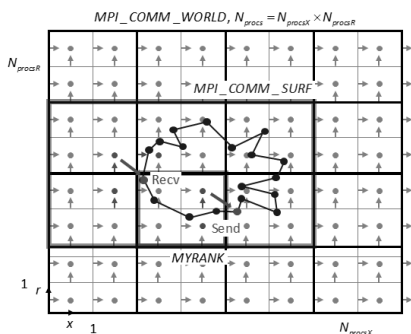


Fig. 4. New MPI Communicator, MPI\_COMM\_SURF, to Calculate the Noise Points on the Surface. Blue Line with Symbols Denote the Surface Mesh of the Body

Table 1. Size and Transfer Number of Noise Data per MPI Process (Number of Grid Points: 46,000,000, Noise Source Points: 67,000, Number of Cores: 64)

MPI rank of sending process	Number of receiving MPI processes	Buffer size of transfer	Number of MPI communication
2	1	1056 Byte	10
4	1	1056 Byte	6
10	6	1056 Byte	2666
11	7	1056 Byte	1524
12	5	1056 Byte	936
18	4	1056 Byte	2699
19	7	1056 Byte	1647
20	5	1056 Byte	500
26	6	1056 Byte	4592
27	8	1056 Byte	2931
28	5	1056 Byte	1327
34	4	1056 Byte	948
35	7	1056 Byte	887
36	3	1056 Byte	641
42	2	1056 Byte	28
43	2	1056 Byte	7

MPI 프로세스별로 필요한 주위 데이터 개수 및 통신 횟수는 Table 1에 있다. 하나의 포인트에서 발생하는 MPI 통신 1회시 전달되는 통신 데이터 버퍼 크기는 1056바이트로 크지 않다. 하지만 MPI 프로세스별로 여러 포인트가 포함되어 있고, 주위의 여러 MPI 프로세스에서 포인트 정보를 가져와야 하므로 통신 횟수가 Table 1에 보듯이 최대 4,000번 이상일 수 있다. 특히 한 번에 수행하는 MPI 프로세스 수가 수천 개 이상일 경우에 한 번에 발생하는 통신 횟수가 백만 번 이상으로 발생하여 계산 속도가 심각하게 느려진다. KISTI 슈퍼컴퓨터인 타키온2에서 실험 시 통신 멈춤 현상이 발생하였다.

### 3. MPI 일방향통신을 이용한 최적화

본 절에서는 소음 해석 시 발생한 통신 멈춤 현상을 해결하기 위한 방법 및 추가 최적화 방법을 서술한다.

먼저 통신 멈춤 현상을 해결하기 위해 다음과 같은 방법을 사용했다. 기존 통신에 사용된 방법은 Fig. 5A와 같이 MPI에서 일반적으로 사용하는 양방향 통신(two-sided communication) 함수인 MPI\_Isend와 MPI\_Irecv를 사용하였다. 해당 통신 함수는 양방향 쌍(pair)에 대한 동기화가 이루어진 다음에 통신을 시작하므로 통신 횟수가 많은 경우에 큰 부하가 발생한다. 그래서 본 연구에서는 MPI-2 표준[11]에서 제시한 일방향 통신(one-sided communication) 방법을 도입하였다. 일방향 통신은 수신처와 송신처에 대한 동기화가 이루어지지 않은 상황에서 한 방향으로 메시지 전달이 가능하다. Fig. 5B와 같이 왼쪽에 윈도우(window)라고 불리는 메모리 영역을 할당하고, 왼쪽 메모리에 MPI\_Put 함수를 사용하여 전달이 가능하다. 해당 경우에는 점대점 통신의 MPI\_Irecv 함수와 같은 수신역할이 필요 없고, 그에 따라 통신 초기화 오버헤드가 줄어든다.

본 논문에서는 원격 윈도우 메모리 영역에 대한 동기화의 경우 수동형 방법(passive target synchronization)인 MPI\_Win\_lock과 MPI\_Win\_unlock 함수를 사용하였다. 원격지 메모리를 원격지(target) 프로세스에서는 제어할 필요가 없으므로 능동형(active target synchronization) 방법을 사용하지 않았다.

일방향 통신 적용과 더불어 추가적으로 많은 CPU 코어에서의 병렬확장성을 확보하기 위해 노드 간의 통신 횟수를 줄이고 메모리 사용량을 줄였다. 먼저 통신 횟수를 줄이기 위해 송신처에서 임시 버퍼를 할당하고 이 버퍼에 개별 MPI 프로세스로 전송하는 데이터를 모아서 한 번에 전송하는 방식을 사용하였다. 다만 해당 수신처의 MPI 프로세스 임시버퍼에서 메모리 복사를 해야 하는 부하는 발생한다. 그리고 메모리 사용량을 줄이기 위해 소음 발생 포인트에 해당하는 축류 팬 표면을 구성하고 있는 삼각형 메쉬의 꼭지점에 작용하는 소음원(noise source)을 각 프로세스에서 미리 계산하고 결과 값만을 보내는 방식을 사용하였다. 꼭지점에 작용하는 소음원을 계산하기 위해서는 주위 속도 36개와 압력 8개 정보를 통신해야 하는데, 개선된 방식을 이용하면 꼭지점에 정의된 소음원에 대한 9개의 정보만 통신하게 된다.

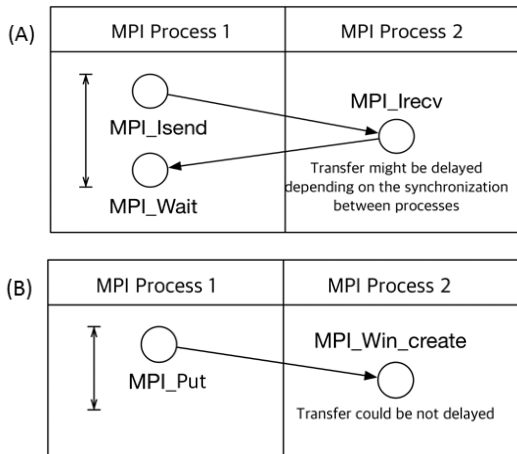


Fig. 5. Schematic Diagram of (A) Two-Sided and (B) One-Sided Communications between MPI Processes

4. 실험 결과

본 절에서는 소음 해석에 적용한 최적화 방법을 구현 및 실험하여 성능향상도를 분석하고 슈퍼컴퓨터에서 병렬확장성을 살펴본다.

일방향 통신을 적용하여 통신 멈춤 현상이 더 이상 발견되지 않았으며, 최적화를 통해 성능을 향상시켰음을 확인하였다. Fig. 6에서 최적화 방법에 따른 성능 향상도를 표시하였다. Type A는 MPI 일방향 통신만 적용하고 최적화를 적용하지 않은 버전, Type B는 통신 횟수를 줄이는 방법만 적용한 버전, Type C는 통신 횟수 및 메모리 사용량을 줄이는 방법을 모두 적용한 버전이다. 통신 횟수 감소를 통해서 1.74배, 메모리 사용량 최적화까지 적용하여 최대 2.97배 계산 속도가 향상되었다.

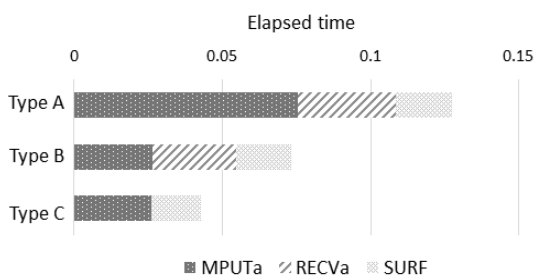


Fig. 6. Performance Comparison among Three Optimization Techniques (MPUTa: MPI Communication, RECVa: Memory Transfer, SURF: Noise Computation on Surface Meshes, Number of Grid Points : 46,000,000, Noise Generation Points: 67,000, x-axis: Elapsed Time, y-axis: Optimization Technique)

MPUTa는 MPI 통신시간을 나타낸 것이다. Type A는 통신에 참여하는 소음원 포인트의 개수만큼 통신하지만, Type B와 C는 특정 MPI 프로세스에 보내야 하는 정보를 한 번에 모아서 보내기 때문에 다른 MPI 프로세스로 보내야 하는 소음원 포인트의 수만큼 통신 횟수가 감소하여 계산 시

간도 줄어든다.

RECVa는 수신 시 발생하는 메모리 복사 오버헤드를 나타낸 것이다. Type A와 B는 소음원을 계산하기 위한 속도와 압력 정보를 직접 주고받아 수신한 데이터를 원본 행렬에 넣어주기 위한 과정이 복잡한 반면에 Type C는 소음이 발생하는 표면에서 계산된 결과 값을 보내기 때문에 메모리 전송 과정이 단순하다. 따라서 Type C의 경우에는 부하가 무시할만하여 그래프에서 표시되지 않았다.

SURF는 통신 이후에 측류 팬 표면에서 발생하는 소음을 계산하는 과정이다. 이는 최적화 방법에 상관없이 모두 동일하여, 계산 시간 역시 동일함을 알 수 있다.

다음으로 유체 및 소음 해석관련해서 슈퍼컴퓨터 타키온2 (Table 2)에서 병렬확장성의 실험결과를 살펴보겠다. 먼저 16억 개의 격자를 이용한 유동 해석에 대한 병렬확장성, 다음으로 4,600만 개의 격자와 67,000개의 소음원을 사용한 소음 실험에 대한 병렬확장성을 살펴보았다.

Table 2. Specification of KISTI Supercomputer Tachyon2

Section	Contents	
Model	Sun Blade 6275	
CPU	Intel Xeon X5570 2.93GHz(Nehalem)	
Number of nodes	3,200	
Total cores	25,408	8cores/node
Rpeak	300 Tflops (3,200 nodes)	
Memory	DDR3/1333MHz 76.8TB	24GB/node, 3GB/core
Interconnect Network	Infiniband 40G 4x QDR	

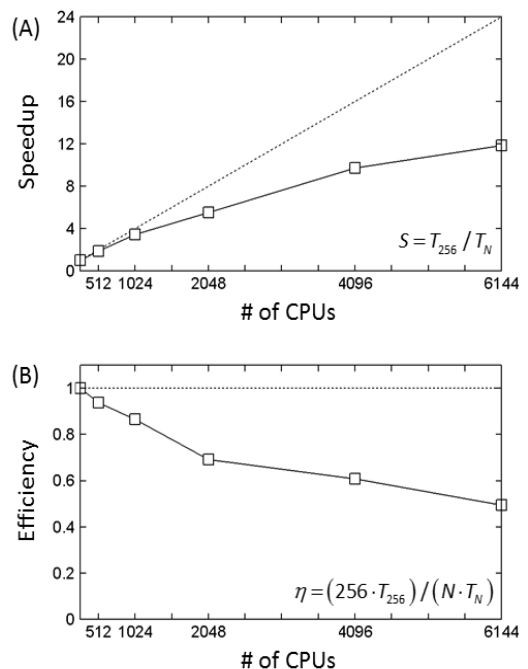


Fig. 7. Speedup (A) and Parallel Efficiency (B) of Code up to 6144 Cores at Tachyon2 (TN: Number of N Nodes, Number of grids: 16×108)

Fig. 7에 16억 개 격자를 이용한 유동 계산의 병렬확장성을 나타내었다. Fig. 7A는 6,144코어까지의 성능증가를 나타내며 256코어 대비 약 12배까지 성능이 향상됨을 알 수 있다. Fig. 7B는 병렬효율을 나타내며 최대 약 50%까지 보여준다. 이는 유동 계산 시 행렬 전치를 위해 집합통신이 사용되어 코어가 증가함에 따라 병목현상이 발생하기 때문이다.

Fig. 8에 4,600만 개의 격자와 67,000개의 소음원 포인트를 이용한 유동 및 소음 계산의 병렬확장성을 나타내었다. Fig. 8A는 128코어까지의 성능증가를 나타내며 16코어대비 약 6배까지 성능이 향상됨을 알 수 있다. Fig. 8B는 병렬효율을 나타내며 128코어에서 약 80%의 효율을 보여준다. Fig. 7의 실험과 달리 격자수가 적어 128코어에서 병렬효율이 좋게 나오지만 격자수를 8억 개 이상으로 증가시키면 비슷한 병렬효율성을 보여줄 것으로 예상된다.

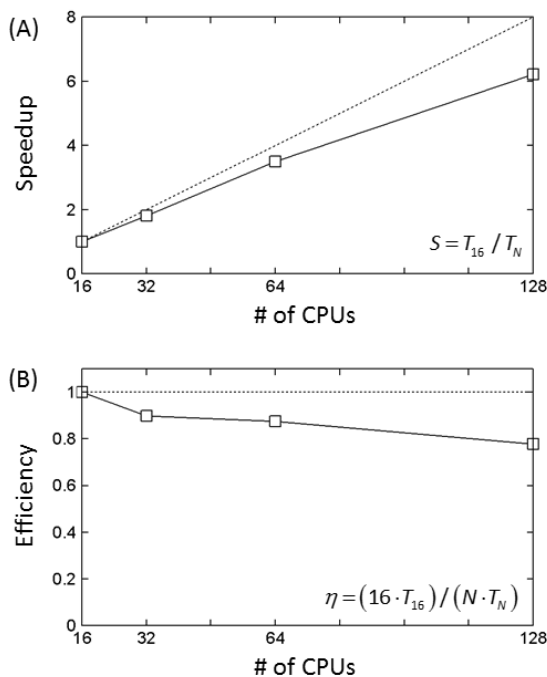


Fig. 8. Speedup (A) and Parallel Efficiency (B) of Code up to 128 Cores at Tachyon2 (TN: Number of N Nodes, Number of Grids:  $46 \times 10^6$ ,  $67 \times 10^4$  Noise Source Points)

## 5. 논의 사항 및 향후 연구

본 절에서는 앞서 논의하지 않은 네 가지 사항에 대해 기술하겠다. 앞의 세 가지 논의 사항은 병렬효율성을 향상시키기 위한 방법이고, 마지막은 거대규모에서의 소음 계산 시 병렬확장성에 관련된 사항이다.

첫 번째 논의 사항은 통신과 계산을 중첩시켜 병렬효율성을 향상시키는 방법이다. 타키온2의 6144코어에서 유동 해석 프로그램의 병렬효율성이 약 50%정도인 이유는 앞서 밝혔듯이 MPI 집합통신으로 인한 성능 저하 때문이다. 같은 계산을 코어 수를 증가하여 계산하면 성능 저하가 더 발생

할 것으로 예상된다. 이를 극복하기 위해 통신과 계산을 중첩하여 통신 병목현상을 극복하면 성능 향상이 가능할 것으로 예상된다. 통신과 계산을 중첩시키기 위한 방법은 비동기식 집합통신(non-blocking communication) 함수를 이용하는 연구([12, 13])와, MPI 일방향통신 함수를 이용하는 방법이 있다[14].

두 번째 향상 방법은 하이브리드 병렬화를 사용하는 것이다. 현재 계산노드 내 도메인 분할을 MPI만을 이용해서 구현하였는데, 노드 내에서는 OpenMP 등의 공유 메모리 분할 방법을 노드 간에는 MPI를 이용한 하이브리드 병렬화 방법을 사용하면 전체 성능 향상에 도움이 될 것이다. 뿐만 아니라 현재 슈퍼컴퓨터의 개발이 CPU당 코어 수가 많아지는 방향으로 이루어지므로, 하이브리드 병렬화 방법을 사용하는 것이 반드시 필요하다.

세 번째 사항은 MPI 일방향통신 사용시 최적의 동기화 방법을 탐구하는 것이다. 현재는 윈도우 메모리 영역에 대한 동기화를 수동형 방법을 사용하고 있는데, 다른 두 가지 동기화 방법과 비교를 통해 병렬확장성을 향상시킬 수 있는 방법을 고안할 예정이다. 첫 번째는 MPI\_Win\_fence 함수를 사용하여 통신하는 MPI 함수들이 모두 동기화에 참여하는 능동형 방법(active target synchronization)이다. 두 번째는 MPI\_Win\_post, MPI\_Win\_start, MPI\_Win\_complte, MPI\_Win\_wait 함수를 사용하여 능동형 방법보다 동기화 시점을 좀 더 세밀하게 제어할 수 있는 일반적 능동형 방법(general active target synchronization)이다. 세 가지 동기화 방법은 [15]의 연구와 같이 MPI 구현물과 하드웨어에 따라 성능차이가 발생할 수 있다.

마지막으로 목표로 하는 큰 격자수와 소음원 개수를 가진 소음 계산의 병렬확장성을 실험으로 확인하지 못하였다. 하지만, 유동 해석만 실험하였을 경우 목표로 하는 격자수에서 최대 50%까지 병렬효율성을 확인할 수 있었다. 소음이 추가되어 계산되더라도 병렬효율성이 크게 다르지 않을 것으로 앞선 실험들에서 살펴볼 수 있었다. 추후 해당 내용에 대해서 실험할 예정이다

## 6. 결론

축류 팬(axial fan)은 선풍기와 같은 팬이 회전하면서 다량의 공기를 불러주는 유체 기계이다. 본 연구는 팬 주위의 소음 해석을 위한 MPI 병렬프로그램을 설계하였으며 최적화 기법에 대해 다뤘다.

목표로 하는 문제인 수억 개의 격자와 수만 포인트의 소음원 문제는 단일 계산 노드에서 수행하기 불가능하여 이차원 도메인 분할 방법을 사용해 MPI 병렬화를 하였다. 이때 소음 문제 해석 시 각 MPI 프로세스 간의 통신이 많이 발생하여 통신 멈춤 현상이 발생한다. 본 연구는 이를 극복하기 위해 MPI 일방향 통신을 적용하였다. 뿐만 아니라 통신 최적화를 통해 1.74배, 메모리 최적화 방법까지 적용하여 최대 2.97배 향상을 시켰다. 마지막으로 KISTI 타키온2 슈퍼

컴퓨터를 활용하여 유동 계산 시 6,144코어에서 최대 12배, 소음 계산 시 128코어에서 최대 6배의 성능향상을 달성하였다.

### References

[1] W. Jiang, J. Liu, H. W. Jin, D. K. Panda, W. Gropp, and R. Thakur, "High performance MPI-2 one-sided communication over InfiniBand," In *2004 IEEE International Symposium on Cluster Computing and the Grid, CCGrid*, pp.531-538, 2004.

[2] A. Pogorelov, M. Meinke, and W. Schröder, "Cut-cell method based large-eddy simulation of tip-leakage flow," *Physics of Fluids*, Vol.27, No.7, 075106, 2015.

[3] K. Park, H. Choi, S. Choi, Y. Sa, and O.-K. Kwon, "Unsteady characteristics of tip-leakage flow in an axial flow fan," in *Proceedings of the 10th International Symposium on Turbulence and Shear Flow Phenomena (TSFP10)*, Chicago, USA, 2017.

[4] S. Laizet, E. Lamballais, and J. C. Vassilicos, "A 2D domain decomposition, a customized immersed boundary method and a zest of numerical dissipation: a successful cocktail to tackle turbulence on HPC systems," *APS Division of Fluid Dynamics (Fall)*, 2015.

[5] F. M. White, "The differential equation of linear momentum," in *Fluid Mechanics, 6th ed.*, McGraw-Hill Boston, p.238, 2008.

[6] K. Akselvoll and P. Moin, "An efficient method for temporal integration of the Navier-Stokes equations in confined axisymmetric geometries," *Journal of Computational Physics*, Vol.125, No.2, pp.454-463, 1996.

[7] L. H. Thomas, "Elliptic Problems in Linear Differential Equations over a Network," Watson Sci. Comput. Lab Report, Columbia University, New York, 1949.

[8] P. Moin, "Discrete Transform Methods," in *Fundamentals of Engineering Numerical Analysis*, Cambridge University Press, p.159, 2001.

[9] J. E. Ffowcs Williams and D. L. Hawkins, "Sound Generation by Turbulence and Surfaces in Arbitrary Motion," *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, Vol.264, No.1151, pp. 321-342, 1969.

[10] Woojin Kim, Injae Lee, and Haecheon Choi, "A weak-coupling immersed boundary method for fluid-structure interaction with low density ratio of solid to fluid," *Journal of Computational Physics*, Vol.359, pp.296-311, 2018.

[11] MPI: A Message-Passing Interface Standard Version 2.2, <http://mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf>, 2009.

[12] T. Hoefler, P. Gottschling, and A. Lumsdaine. Brief announcement: Leveraging non-blocking collective communication in24.

[13] S. Song and J. K. Hollingsworth, "Scaling parallel 3-D FFT with non-blocking MPI collectives," *Proceedings of the 5th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems*, IEEE Press, pp.1-8, November 16-21, 2014.

[14] V. Cardellini, A. Fanfarillo, and S. Filippone, "Overlapping communication with computation in MPI applications," Feb. 2016, [online] Available: <http://hdl.handle.net/2108/140530>.

[15] William D. Gropp and Rajeev Thakur, "Revealing the Performance of MPI RMA Implementations," *EuroPVM/MPI 2007*.



### 권오경

<http://orcid.org/0000-0001-9734-9257>

e-mail : okkwon@kisti.re.kr

2000년 고려대학교 컴퓨터학과(학사)

2002년 KAIST 전산학과(석사)

2002년~현 재 한국과학기술정보연구원  
선임연구원

관심분야 : Parallel Computing, High Performance Computing, MPI



### 박근태

<http://orcid.org/0000-0002-5081-0621>

e-mail : parkkeuntae@gmail.com

2008년 포항공과대학교 기계공학과(학사)

2010년 서울대학교 기계항공공학부(석사)

2013년~현 재 서울대학교

기계항공공학부 박사과정

관심분야 : Axial Fan, CFD, Parallel Computing



### 최해천

<http://orcid.org/0000-0003-0696-847X>

e-mail : choi@snu.ac.kr

1985년 서울대학교 기계공학과(학사)

1987년 서울대학교 기계공학과(석사)

2011년 Stanford University Mechanical  
Engineering(박사)

1993년~현 재 서울대학교 기계항공공학부 교수

관심분야 : Turbulence, Flow Control, CFD, Bio-Mimetic Engineering