

열유체해석 모델의 병렬화

• 곽호상 | 금오공과대학교 기계공학부, 조교수 / e-mail:hskwak@kumoh.ac.kr

이 글에서는 유한차분(체적) 및 유한요소 열유체해석 모델에 적합한 병렬화 방법을 간략히 소개하고 지금까지의 병렬 컴퓨팅 동향과 앞으로의 전망을 조명하고자 한다.

확장성이 뛰어난 병렬컴퓨터의 출현으로 고성능 전산자원이 필수요건인 전산열유체공학은 새로운 전기를 맞이하고 있다. 실용도구로서의 위상이 확고해짐은 물론, 막대한 계산량과 기억용량 때문에 접근이 어려웠던 난류, 다상 유동과 같은 초대형 과제에 도전할 수 있는 기반이 조성되었다.

병렬 컴퓨팅의 목적은 빠른 계산, 큰 계산이다. 프로세서 하나로 처리하는데 100 시간 걸리던 문제가 있다면 프로세서 100 개를 동원하여 1 시간에, 또는 100 배 큰 문제를 해석하는 것이다. 필요조건은 복수의 프로세서를 효과적으로 활용할 수 있는 병렬 계산 모델이다.

병렬화 기법

여러 사람이 분업을 통해 하나의 프로젝트를 수행하는 것처럼 복수의 프로세서가 하나의 작업을 분할처리하는 것이 병렬 컴퓨

팅이다. 분업의 방법이 프로젝트의 성격에 따라 달라지듯이 병렬 처리 방법 또한 문제의 특성, 계산 알고리즘, 데이터 구조에 크게 의존한다. 여기서는 열유체 해석에 가장 많이 활용되는 유한차분(체적) 및 유한요소 모델의 병렬화에 적절한 방법을 알아보기로 하겠다.

프로그래밍 모델

병렬 계산에 필요한 작업의 분할과 프로세스의 제어, 데이터의 분할과 교환을 처리할 수 있도록 프로그램을 작성하는 것이 병렬 프로그래밍이다. 현재 활용 가능한 프로그래밍 모델은 크게 데이터 병렬형, 공유 메모리형, 메시지 전달형으로 분류할 수 있다.

데이터 병렬형은 HPF(High Performance Fortran)와 같은 자동병렬 컴파일러를 활용하는 가장 손쉬운 방안이지만 병렬계산효율이 좋지 않아 아직까지는 성공적인 기법이라 할 수 없다.

공유 메모리형은 다수의 프로세서가 대형 메모리를 공유하는 SMP(symetric multi Processors)를 겨냥한 프로그래밍 모델로 시스템이 지원하는 컴파일러 지시어(directive)를 이용하는 방법이다. 최근에는 SGI가 개발한 OPENMP가 공유메모리형 병렬화의 표준도로 자리잡고 있다. 메모리 주소체계를 공유하기 때문에 작업 분할과 제어만으로도 병렬처리가 가능하여 프로그래밍이 용이한 것이 장점이다. 반면에 프로세서 수가 어느 정도 이상 증가하면 프로세서와 메모리를 연결하는 버스에 병목이 생겨 성능이 저하되는 것이 약점이다. 16~32 개 미만의 프로세서를 사용하는 SMP 용 중규모 해석모델이라면 비교적 적은 노력으로 병렬화 할 수 있는 적절한 방안이다.

메세지 전달형은 모든 프로세서가 별도의 메모리를 가지는 분산 메모리를 전제로 한다. 각 프

로세서는 독립적인 주소체계와 프로세스 제어를 가지게 되어 작업분할과 함께 데이터 배분과 관리가 병렬처리의 필수조건이다. 이때 다른 프로세서의 메모리에 저장된 데이터가 필요한 경우가 발생하는데 이 문제는 프로세서 간 통신에 의한 자료교환으로 해결한다. 이러한 기능을 제공하는 모듈을 모은 것이 PVM(parallel virtual machine), MPI(message passing interface)와 같은 메시지 전달 라이브러리이다. 순차형 원본 코드에 병렬처리에 필요한 메시지 전달 모듈을 불러 사용하는 것이 프로그래밍의 기본틀이다. 상대적으로 프로그래밍이 어렵지만 병렬 확장성이 우수하여 소형에서 초대형 계산까지 적용범위의 제약이 없는 것이 장점이다. 특히 MPI는 SMP를 포함한 모든 종류의 컴퓨터가 지원하는 비공식 표준으로 완벽한 이식성을 보장한다. 이러한 이유로

MPI를 이용한 메시지 전달형 프로그램이 병렬형 열유체해석 모델의 주류를 형성하고 있다.

병렬화 알고리즘 - 영역분할
 격자를 사용하는 열유체해석 모델에 가장 적합한 병렬처리 알고리즘은 영역분할(domain decomposition)이다. 영역분할은 전체 계산영역을 다수의 소영역으로 나누고 소영역마다 하나의 프로세서를 배정하여 해를 구하는 방법이다. 오일러 관점에서 나비에-스토크스(Navier-Stokes) 방정식의 해를 구하는 유한차분(체적)법과 유한요소법은 물리적 공간을 계산격자로 일대일 투영하여 해를 구하는 공통점을 가지고 있다. 따라서 계산영역의 분할은 직접적이며 용이하다. 유사한 것이 복잡한 계산영역을 여러 개의 단순 블록으로 나누어 모사하는 다중블록 해법이다. 이때 하나의 블록을 소영역으로

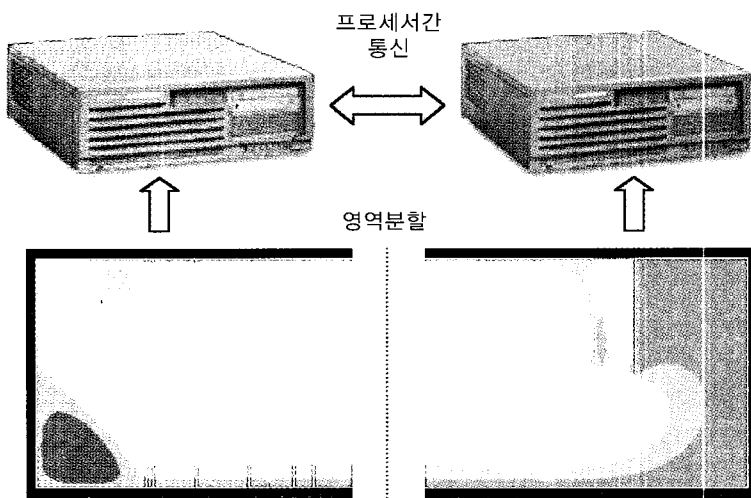
간주하고 하나의 프로세서에 배당하면 이것이 바로 영역분할이다.

병렬모델 설계에 있어 일차적인 고려사항은 부하균형이다. 프로세서에 할당된 작업량의 차이, 즉 부하불균형은 일부 프로세서의 공전을 의미하기 때문에 병렬 계산효율에 치명적이다. 열유체해석 모델의 중요한 성질은 특별한 국소생성향이 없는 경우 격자점 당 계산량이 거의 일정하다는 것이다. 따라서 계산영역을 균등 분할하여 프로세서 당 격자수를 동일하게 배정하면 부하균형 문제가 자연스럽게 해결된다. 일반적으로 열유체해석 모델이 화학 계산이나 기상예보 모델보다 뛰어난 병렬성능을 자랑하는 것은 바로 이러한 특성 때문이다.

영역분할에 의해 각 프로세서에 할당된 소영역은 해를 구해야 하는 해석영역과 이를 둘러싼 경계로 구성된다. 경계는 다시 실제 유동 경계조건이 적용되는 물리적 경계와 분할에 의해 생긴 인접 소영역과의 가상경계로 구분된다. 가상경계는 인접 소영역의 해석영역과 중첩되기 때문에 인접 소영역을 담당하는 프로세서와 통신에 의해 경계값을 교환하여야 한다. 이처럼 프로세서간 통신 때문에 소요되는 시간을 통신비용(communication overhead)이라 하며, 이의 최소화가 효율적인 병렬 컴퓨팅의 또 하나의 관건이다.

병렬형 행렬해법

열유체해석 모델 병렬화에서



영역분할에 의한 분산 메모리형 병렬계산 개념

반드시 고려해야 할 요소가 행렬 해법이다. 모든 행렬해법은 고유의 특성과 배경을 가지고 있어 단순하게 병렬화하면 본래의 수학적 본질을 잃어 수렴력이 달라지거나 경우에 따라서는 오답을 주기도 한다. 그러나 순차모델 행렬해법을 알고리즘의 훼손없이 병렬화를 하는 것이 대단히 어렵거나 가능하더라도 상당한 병렬처리 비용을 요구하여 실익이 없는 경우가 많다. 이 때문에 영역분할을 기조로 하는 병렬해법에서는 분할된 소영역을(병렬계산을 통한 비행체 주위 유동 해석, 색상은 비정렬격자의 영역분할을 나타낸다.

독립영역처럼 간주하여 원본의 순차 행렬해법으로 해를 구하고 통신에 의해 경계치를 갱신하는 과정을 되풀이하는 반복법을 채용하는 것이 일반적이다. 프로그램이 쉽고 확장성이 보장되기 때문이다. 그러나 이 해법은 순차해법과는 근본적으로 다른 것이기 때문에 충분한 신뢰성 검증이 이루어져야 한다.

효율적인 병렬모델 작성방법

이상의 논의를 바탕으로 효율적인 병렬모델 작성순서를 정리하면 다음과 같다. 작업은 병렬화할 순차형 코드를 준비하는 것으로부터 시작한다.

(1) 병렬 가능 알고리즘의 결정 : 프로그램의 일부 또는 전체 흐름이 병렬화에 적합한지 분석한다. 만약 전체 흐름이 병렬화에 부적절하면 알고리즘의 수정이나 다른 병렬화 알고리즘을 채용을

고려한다.

(2) 병렬화 전략 수립 : 프로그램 래밍 모델을 선택하고, 부하균형 및 통신비용 최소화 방안을 수립한다. 유한차분 또는 유한요소 모델의 경우, 영역분할과 MPI를 이용한 메시지 전달형이 표준적인 방법이다.

(3) 프로세서간 통신 설정 : 데이터 의존성을 분석하여 동기화 지점과 교환할 정보, 통신방법을 결정한다. 통신을 요구하는 주된 작업은 초기 입력자료와 같은 전역 정보 공유, 해의 수렴판정, 가상경계에서의 해의 갱신 등이다. 가능하면 동기화 지점과 통신횟수를 줄여 통신비용을 최소화하는 것이 좋다.

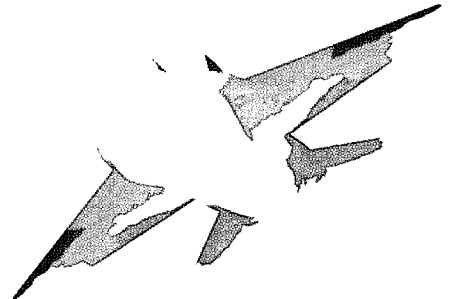
(4) 병렬화 코딩 : 계산영역의 분할, 프로세서 토폴로지, 계산 인덱스 정의, 전체적 작업과 국소적 작업의 분류 등 병렬 처리에 필요한 부분을 추가하고 메시지 전달 모듈을 이용하여 프로그램을 완성한다. 이때 통신처리 등 추가되는 기능을 모듈화하는 것이 원본코드에 주는 영향 최소화 와 디버깅의 편이를 위해 바람직하다.

(5) 검증 시험 및 조율 : 계산 결과의 신뢰성을 검증한 후 프로세서의 수를 늘여가면서 확장성을 시험하여 프로그램을 조율하고 최적화한다.

병렬화 동향

초기 병렬화 사례 : NPB

열유체 해석은 대표적인 고성능 컴퓨팅 응용분야로 병렬컴퓨



Mirrored image, 16 domains

병렬계산을 통한 비행체 주위 유동 해석, 색상은 비정렬격자의 영역분할을 나타낸다

터 등장 초기부터 적극적인 병렬화가 시도되어 선도적인 성과를 보여주었다. 대표적인 예가 병렬 컴퓨터 성능분석용 벤치마크 시험예제로 자주 활용되는 NPB (NAS parallel benchmark)이다. NPB는 유동계산에 필요한 기능을 대표하는 다섯 개의 단순 커널 예제 EP(embarassingly parallel), MG(multi-grid), CG-(conjugate gradient), FT(Fourier transform), IS(integer sort)와 세 개의 대형예제, LU(lower & upper triangular), SP(scalar pentadiagonal), BT(block tridiagonal)로 구성된다. LU, SP, BT는 NASA 에임즈연구소에서 개발한 INS3D, ARC3D 코드에 기반한 실제 유동해석 프로그램이다.

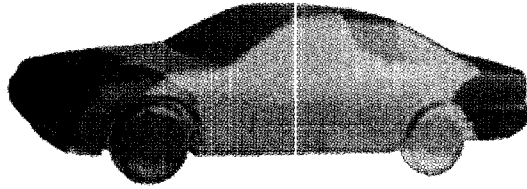
NPB는 공개 소프트웨어로 열유체해석 모델 병렬화의 훌륭한 참고자료이다. MPI를 이용한 NPB 2.3이 최신의 공식버전이며, OPENMP나 HPF를 이용한 병렬화 모델도 찾아볼 수 있다. NPB 홈페이지(www.nas.nasa.gov/NAS/NPB)에는 원본 코드

와 함께 주요 컴퓨터 기종에서의 병렬 시험결과가 정리되어 있다. 대부분의 병렬 컴퓨터에서 우수한 병렬계산성을 얻을 수 있음을 확인할 수 있는데 열유체 해석이 병렬 컴퓨팅에 적절함을 보여주는 결과로 이해해도 좋을 것이다.

상용 코드의 병렬화

1980년대에 설립된 주요 열유체 해석용 소프트웨어 회사들이 1990년대에 들어 집중적인 노력을 기울인 것이 전후처리가능 보강과 코드의 병렬화이다. 전자가 사용편이성을 위한 것이라면 후자는 코드를 장착할 주요 플랫폼이 병렬형으로 전환되는 현실의 반영이었다. 초기 병렬화는 연산 부하가 많이 걸리는 모듈을 중심으로 제한적으로 이루어졌으나 최근에는 신형 모델을 기획할 때 병렬처리를 기본 설계개념으로 채택하고 있다. 하드웨어 측면에서 보면 SMP 노드에서만 쓸 수 있는 공유메모리형에서 모든 종류의 병렬컴퓨터에 장착가능한 분산 메모리형으로 발전하고 있다. 이러한 노력에 힘입어 초창기 제한적이던 확장성이 상당히 개선되어 대형문제인 경우 64개의 프로세서까지도 우수한 확장성이 유지되는 성과들이 발표되고 있다. 현재 FLUENT, CFX, STAR-CD 등 주요 상용 패키지들이 모두 병렬처리를 지원하고 있으며 홈페이지를 방문하면 관련 자료를 찾아볼 수 있다.

우리나라의 병렬화 동향



FLUENT를 이용한 병렬계산 사례 : 360만 개의 격자로 자동차 주위의 유동 해석

우리나라에서 병렬 컴퓨팅이 본격적으로 시작된 것은 1997년부터라고 볼 수 있다. 당시 SERI (현 KISTI) 슈퍼컴퓨팅센터에 128 프로세서를 장착한 초병렬컴퓨터 CRAY T3E가 도입되면서 국내 최초로 체계적인 병렬 프로그래밍 교육이 이루어졌는데 이에 힘입어 슈퍼컴퓨터 사용자를 중심으로 초기 병렬화 작업이 이루어졌다. 특히 열유체 해석 분야의 병렬화는 정렬격자를 사용하는 유한체적 모델에서 비정렬 격자를 사용하는 유한요소 모델에 이르기까지 다양하게 이루어져 타 분야에 모범적인 성공사례로 제시되었다.

이후 다소 소강상태에 있던 병렬화 열기를 다시 고조시킨 것은 PC 클러스터였다. 전 세계적인 Linux 열풍과 함께 국내의 고성능 컴퓨팅 그룹들도 Linux 기반 PC 클러스터를 직접 제작하여 사용하기 시작하였고 이러한 추세는 확산되면서 자연스럽게 병렬 컴퓨팅 저변이 확대되었다. 병렬화 대상도 격자모델 중심에서 전역함수 전개에 기초한 스펙트럴 기법 등 다양한 해석방법론으로 확장되었다. 여러 가지 병렬 프로그래밍 모델이 시험되는가 하면 새로운 병렬화 알고리즘 개

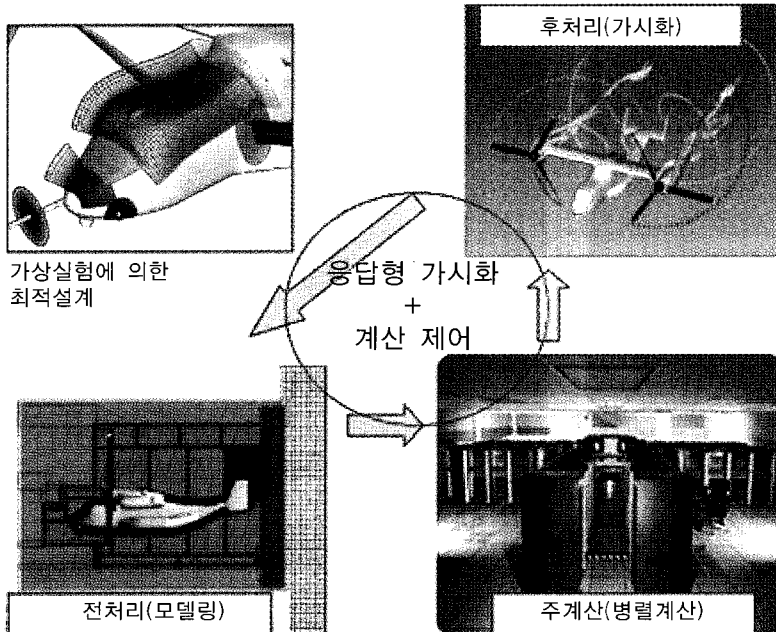
발과 병렬 최적화도 이루어지고 있다. 그 성과로 열유체 해석은 이제 100만 개를 넘어 1,000만 개에 육박하는 격자를 사용하는 초대형 문

제에 도전하고 있다. 공력해석의 경우, 날개 중심 해석을 탈피하여 전체 비행체를 완전 모사하는 것이 현실로 다가오고 있다.

앞으로의 전망

‘병렬계산’이 인구에 회자된 것은 불과 5년 남짓이지만 이미 병렬컴퓨터는 대세가 되었다. 주요 계산과학용 컴퓨터는 예외 없이 병렬형이고, 다중 CPU를 장착한 PC를 발견하는 것도 어렵지 않다. KISTI 슈퍼컴퓨터센터에서는 조만간 4TFLOPS 규모의 초고성능 병렬컴퓨터를 도입하고 1 TFLOPS급 클러스터 시스템을 구축한다고 한다. 이전에 꿈꾸지 못했던 초대형 계산이 현실로 다가오고 있다. 고성능 대용량 컴퓨팅이 필요한 전산열유체 연구자에게 병렬 컴퓨팅은 더 이상 선택이 아니라 필수가 되었다.

일반적 의미에서 병렬화는 복잡한 성격의 작업이다. 그러나 열유체 해석 모델은 병렬처리에 유리한 여러가지 특성을 가지고 있어 병렬화가 비교적 용이하고 병렬계산 성능 또한 뛰어남이 이미 증명되었다. 초기 병렬화에서는 축적된 경험과 기술의 부족이 어려움이었지만 이제는 참고할 만



분산형 가상풍동 시뮬레이터의 개념도

한 성공사례가 적지 않고 도움을 받을만한 전문가 그룹도 존재한다. 오늘의 전산자원을 향유하고 내일의 고성능 계산을 대비하기 위한 해석모델의 병렬화, 아직 시작하지 않았다면 바로 지금이 시작할 시점이다.

병렬화의 뒤를 이어 떠오르는 화두는 무엇일까? 분산컴퓨팅이다. 전산해석의 지향은 실제 실험

처럼 계산결과를 직접 관찰하면서 시뮬레이션을 실시간 조작하는 완전한 가상실험이다. 관심을 모으고 있는 가상풍동이 그 예가 될 것이다. 열유체 해석은 형상 모델링과 격자생성과 같은 전처리, 수치해를 구하는 주계산, 계산결과를 가시화하고 분석하는 후처리로 구성되는데 서로 성격이 상이하여 독립적으로 수행되

거나 하드웨어를 달리하는 경우도 많다. 이처럼 이질성이 강한 구성요소의 통합, 즉 분산 처리가 가상실험 환경구축을 위한 핵심적인 기술과제로 떠오르고 있다. 다른 한편으로 전국에 분산된 고성능 전산자원을 초고속망으로 연계활용하는 국가 그리드 계획이 추진되면서 원격분산 컴퓨팅(메타컴퓨팅)에 대한 관심이 더욱 고조되고 있다. 분할된 작업간의 연관성이 강한 병렬 컴퓨팅에 비해 분산 컴퓨팅은 망을 통해 느슨하게 연결된 분산자원을 활용하는 방안이다. 내용면에서 병렬 컴퓨팅이 동일한 과업을 분할처리하는 작업병렬(task parallel)형 분업이라면 분산 컴퓨팅은 역할을 분담하는 임무병렬(mission parallel)형 협업의 성격이 강하다. 이러한 차이에도 불구하고 분산 컴퓨팅은 기술적인 면에서 분산 메모리형 병렬처리의 확장으로 볼 수 있다. 열유체해석 모델의 병렬화에서 축적된 기술과 경험이 분산형 모델 개발의 밑거름이 될 것이다.

기계유어 해설

▶ 공기 주축부(Air Spindle)

공기주축부는 컴퓨터용 하드 디스크 등의 정밀 부품 및 반도체용 웨이퍼 가공을 위한 정밀 동작기계의 주축으로 사용되는 공기베어링 지지 주축부를 의미한다. 공기 주축부는 일반적으로 회전하는 주축과 주축을 지지하기 위한 공기베어링 그리고 주축을 회전시키기 위한 구동부로 구성된다. 공기 주

축부는 공기를 작동유체로 사용하기 때문에 오일을 작동유체로 사용하는 저널베어링 지지 주축부에 비하여 지지하중은 작지만, 회전시 발생하는 마찰이 작아 고속회전에 적합하며, 오일저널베어링에서 발생하는 오일 휩(oil whip)과 같은 과도한 구축불안정 현상이 발생하지 않는다.