

CFD 프로그램 수행을 위한 그리드 컴퓨팅 기반의 메타컴퓨팅 시스템

이건우^o, 우균*, 강경우**, 권오경***

*부산대학교 컴퓨터공학과

**백석대학교 정보통신학부

***한국과학기술정보연구원 그리드컴퓨팅 연구팀

{leoric99^o, woogyun}@pusan.ac.kr, kwkang@bu.ac.kr, okkwon@kisti.re.kr

Metacomputing System on Grid Computing for Executing CFD Programs

Gunwoo Lee^o, Gyun Woo*, Kyungwoo Kang**, Ohkyoung Kwon***

*Dept. of Computer Engineering, Pusan National University

**Dept. of Information and Communication, Baekseok University

***Gridcomputing Research Team, KISTI

요 약

CFD(Computational Fluid Dynamics)는 수치 기법(Numerical Methods)과 알고리즘을 사용하여 유체 유동 문제를 풀고 해석하는 것이다. 본 논문에서는 이러한 CFD 분석 프로그램의 효율적인 수행을 위해 분산 환경을 기반으로 하는 메타컴퓨팅(Metacomputing) 시스템에 대해서 기술한다. 실제 CFD 프로그램을 단일 클러스터 시스템에서 수행시켰을 때와 메타컴퓨팅 시스템을 이용하여 수행시켰을 때 소요되는 시간과 결과 파일을 실행을 통하여 비교한다. 그 결과 메타컴퓨팅 시스템을 이용하여 CFD 분석 프로그램을 분산 수행시킨 경우는 그렇지 않은 경우에 소요되는 시간보다 평균 15.3~38.5% 정도 빨랐고, 동일한 결과를 얻을 수 있음을 확인하였다.

1. 서 론

CFD(Computational Fluid Dynamics)[1] 프로그램은 유체역학 문제를 풀기 위한 것으로 컴퓨팅 기술의 발달로 열유체 유동역학 등과 같은 이론적으로 해석할 수 없었던 문제 등을 해석하는데 이용되고 있다. 이와 더불어 수치해석 방법의 진보가 더해져 미국의 FLUENT[2], GDT Software Group[3] 등과 같은 복잡한 유체유동현상 등을 해석할 수 있는 상용 CFD 프로그램들을 개발하여 시판하는 기업들도 많이 있다.

C, Fortran과 같은 컴퓨터 언어만으로 작성된 CFD 분석 프로그램은 매우 방대한 양의 계산을 수행하기 때문에 수행시간이 매우 오래 걸리게 된다. 이러한 오랜 수행시간을 줄이기 위해서 MPI(Message Passing Interface)나 PVM(Parallel Virtual Machine)과 같은 병렬처리 메커니즘을 함께 사용한다.

병렬처리를 하기 위해서는 우리가 일반적으로 사용하는 PC로는 한계가 있기 때문에 슈퍼컴퓨터를 이용하여 보

다 빠른 CFD 프로그램을 수행한다. 슈퍼컴퓨터의 경우는 개인이나 일반 연구소에서는 높은 가격 때문에 보유가 힘들기 때문에 국내 KISTI와 같이 유료 사용을 제공하는 곳을 이용할 수 있지만 직접 클러스터 시스템[4]을 구축하여 이용하는 경우도 많이 있다. 하지만 클러스터 시스템의 단점은 물리적으로 내부 네트워크에 직접 연결되어야 하고, 이기종간의 호환이 어렵다는 단점이 있다. 이러한 문제점들을 해결하고 보다 확장성을 높일 수 있는 방법이 바로 그리드 컴퓨팅(Grid Computing)이다.

그리드 컴퓨팅은 클러스터 시스템을 기반으로 하지만 물리적으로 직접 연결되지 않은 즉, 지리적으로 멀리 떨어져 있는 자원들을 인터넷으로 연결하고 이를 사용자에게는 마치 하나의 슈퍼컴퓨터처럼 사용할 수 있게 해준다. 그리고 클러스터 시스템의 단점이었던 이기종간의 연결도 가능하게 해준다.

본 논문에서는 이러한 그리드 컴퓨팅을 기반으로 하여 CFD 프로그램 수행 시 자동으로 가장 적절한 계산 자원 시스템을 선택하여 보다 빠르고, 효율적으로 CFD 프

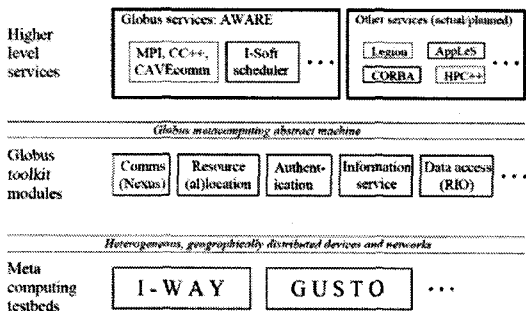
그럼을 수행할 수 있는 시스템 메타컴퓨팅(Metacomputing) 시스템을 설계하고 실험을 통하여 그 결과를 검증한다. 본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 설계한 메타컴퓨팅 시스템의 기반이 되는 배경 연구와 CFD 프로그램에 대해 설명하고, 3장에서는 메타컴퓨팅 시스템의 구조 및 동작에 대해서 기술한다. 4장에서는 실험을 통해서 단일 클러스터 시스템과 메타컴퓨팅 시스템을 이용하여 CFD 프로그램을 수행하고 프로그램 수행에 소요된 시간과 결과 파일을 비교한다. 마지막으로 5장에서는 결론 및 향후 연구를 기술한다.

2. 배경 연구

본 절에서는 논문에서 제안하는 메타컴퓨팅 시스템 구성을 위해 관련된 그리드 컴퓨팅, CFD 프로그램에 대해서 기술한다.

2.1 그리드 컴퓨팅

메타컴퓨팅 시스템은 다수의 클러스터 시스템을 사용자가 마치 하나의 시스템처럼 사용할 수 있게 해준다. 이는 기존의 클러스터 시스템의 단점을 그리드 컴퓨팅을 이용하여 해결할 수 있다. 그리드 컴퓨팅은 기본적으로 이기종간의 연결, 통신, 자원 할당, 등의 기능들을 제공한다.



[그림 1] Globus toolkit

글로벌스 프로젝트(Globus Project)[5]는 그리드 컴퓨팅을 위해 저수준 툴킷을 이용하여 고수준의 서비스를 제공하는 메타컴퓨팅 인프라스트럭처 툴킷을 개발하는 것이다. 그리드 컴퓨팅을 위해 Legion, MOL, Globus Toolkit과 같은 미들웨어들이 개발되었고 현재 가장 널리

이용되고 있는 글로벌스 미들웨어는 Globus Toolkit[6]으로 메타컴퓨팅 시스템은 이를 기반으로 프로그램 수행을 위한 작업 시스템 정보를 알 수 있다.

Globus Toolkit은 크게 다음과 같은 6가지 모듈로 구성된다.

- 자원 할당: 자원 할당, 초기화, 스케줄러 기능 제공
- 통신: 메시지 패싱, 원격 프로시저 호출 등의 기본적인 통신 메커니즘을 지원
- 자원 정보: 자원으로 사용되는 시스템의 구조와 상태를 정보를 감시
- 인증: RSH, SSH 등의 인증 시스템과 GGS(Generic Security System), CA(Certificated-based Authentication)을 지원
- 프로세스 관리: 프로세스 생성, 종료 등의 프로세스 관리
- 데이터: 데이터 액세스를 위한 고속 병렬 파일 시스템과 네트워크 접속이 가능한 I/O 장치 등을 지원

2.2 CFD 프로그램

Navier-Stokes 방정식을 이용하여 유체(기체나 액체)의 유동현상을 분석하는 연구를 흔히 CFD 또는 전산유체역학이라고 말한다. 현재 CFD는 다음과 같은 다양한 분야에서 적용되고 있다.

- 기상학: 기상예측
- 공기역학: 항공기 및 자동차의 양력 및 항력
- 수리학: 선박
- 환경공학: 공기 오염 분포 및 흐름
- 석유화학: 열전달 및 연소가 있는 층류 및 난류유동
- 해양학: 강 및 해수의 흐름,
- 전기전자공학: 반도체장치 내의 냉각

이러한 CFD를 위한 프로그램(편의상 CFD 프로그램으로 명명함.)은 일반적으로 크게 전처리기(Pre-processor), 솔버(Solver), 후처리기(Post-Processor)로 이루어진다. 전처리기는 CFD 프로그램의 실행을 위해 필요한 격자를 생성하고 사용자의 입력을 받아 분석하고자 하는 환경을 설정한다. 솔버는 CFD 프로그램의 엔진에 해당되는 부분으로 수치해석방법을 응용하여 문제를 분석하고, 그 결과를 도출한다. 이 결과는 후처리기에서 사용자에게 결과를 보여주는 부분으로 이는 간단한 텍스트 출력부터 그래픽 출력까지 보

여줄 수 있으며 이 후처리기에 대한 많은 개발이 이루어지고 있다.

```

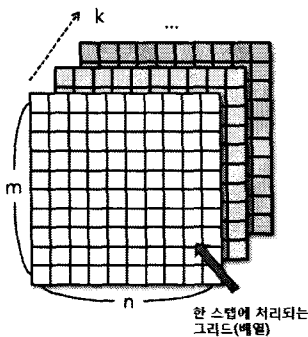
...
Int main(int argc, char** argv) {
    /* 변수 선언 */

    while(numstep >=0) {
        ...
        A(data);
        B(data);
        C(data);
        D(data);
    }

    return 0;
}
    
```

[그림 2] CFD 프로그램(C)의 기본구조

CFD 프로그램은 그림 2와 같이 핵심 루프(Kernel Loop)를 수백에서 수천, 수만 번 수행하는 구조로 되어 있다. 그림 3과 같이 핵심 루프에서는 입력으로 받은 그리드(grid)에 해당되는 2차원 배열을 생성하고 원 스텝 수행에 2차원 배열 하나를 계산한다. 그리고 이를 바탕으로 2차원 배열을 계속 이어나가 3차원 배열을 생성하고 유체 흐름을 분석하게 된다. 이 때문에 CFD 분석에 걸리는 시간은 짧게는 몇 분에서 길게는 수십 시간이 소요된다. 이 때문에 CFD 프로그램의 수행은 보통 병렬처리를 이용한다.



[그림 3] CFD 프로그램 그리드(배열)

CFD 프로그램은 C나 Fortran과 같은 언어를 이용하여 순차처리를 수행하는 형태로 작성할 수도 있지만, 난류 흐름 분석과 같은 아주 많은 계산을 수행하는 경우에는 앞서 말한 것과 같이 먼저 2차원 배열을 먼저 계산하고 이를 3차원으로 확장시켜가며 분석을 수행하기 때문

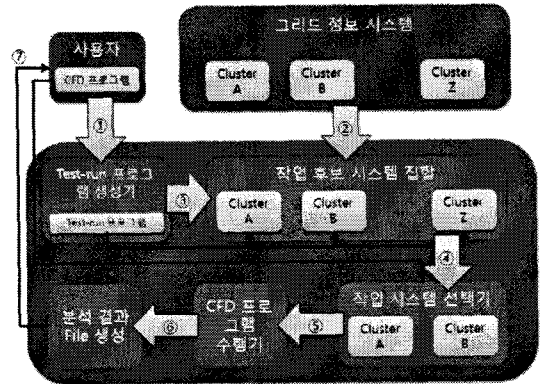
에 2차원 배열을 분석하는 한 스텝에 대해서 병렬처리를 하는 것이 가능하고 순차처리 보다 더 빠른 시간에 분석을 완료할 수 있다.

본 논문에서 제안하는 메타컴퓨팅 시스템은 Test-run 프로그램 생성기를 통해 CFD 프로그램의 핵심 루프를 찾고 이에 해당되는 Test-run 프로그램을 생성한다. 핵심 루프의 수행시간은 CFD 특성상 수치해석 계산을 계속적으로 하는 것이기 때문에 Test-run 프로그램 수행을 통한 결과를 바탕으로 본래 CFD 프로그램의 수행을 위해 적절한 자원을 선택할 수 있다.

3. 메타컴퓨팅 시스템

메타컴퓨팅 시스템은 기본적으로 CFD 병렬 프로그램의 분산 수행을 지원하기 위해 MPI와 PVM이 필요하고, 작업 후보 시스템 선택을 위해서는 이기종 시스템 연결, 자원 정보, 데이터 전송 등의 기능이 필요하기 때문에 Globus Toolkit을 필요로 한다[7].

메타컴퓨팅 시스템은 크게 Test-run 프로그램 생성기, 작업 시스템 선택기, CFD 프로그램 수행기 3가지의 모듈로 이루어진다. 메타컴퓨팅 시스템의 구성과 동작 과정은 그림 4와 같다.



[그림 4] 메타컴퓨팅 시스템 구성 및 동작 과정

사용자는 분석할 CFD 프로그램을 메타컴퓨팅 시스템에 제출하면(①), Test-run 프로그램 생성기는 Test-run CFD 프로그램을 생성하고, 그리드 정보 시스템에 등록되어 있는 시스템들 중 사용 가능한 작업 후보 시스템 집합을 받아온다(②). 작업 시스템 선택기는 그리드 정보 시스템으로부터 받은 작업 후보 시스템 집합에 생성한 Test-run 프로그램을 모두 전송하여 프로그램을 수행(③) 후 그 중 가장 최적의 작업 시스템을 선택하여(④)

CFD 프로그램을 수행한다(⑤).

작업 시스템 선택기가 가장 적절한 작업 시스템을 선택하는 방법은 바로 네트워크 전송 지연시간을 고려하는 것이다. 해당 시스템 자체에서 아무리 빨리 수행 된다고 하더라도 전송 지연시간이 다른 시스템들에 비해 더 오버헤드가 크다면 해당 시스템은 결국 전송 지연 시간은 크고 보다 낮은 성능을 가진 시스템보다 프로그램 수행을 완료하는데 있어 더 늦을 수 있기 때문에 사전에 Test-run 프로그램을 이용하여 자원 정보에 등록된 시스템들의 성능을 평가하고 선택 유·무를 결정한다[8]. 네트워크 토폴로지가 어떻게 구성되어 있는냐에 따라서 시스템의 전송 지연 시간이 결정되기 때문에 이 또한 자원 선택에 있어 중요한 결정 요소가 된다.

CFD 프로그램의 분산 수행을 위해서는 한 스텝에 계산할 그리드를 나누어 선택된 작업 시스템으로 보내야 한다. 이를 위해서 Globus Toolkit을 이용하고, 프로그램을 수행할 클러스터 시스템 노드의 자원 할당 및 관리는 설치된 PBS(Portable Batch System)에서 담당하게 된다. 한 스텝이 끝나면 다음 그리드 계산을 위해 마스터 노드가 분산 수행을 끝마친 그리드에 대한 결과 파일을 한 곳으로 취합하여 다음 계산을 계속 진행한다.

4. CFD 프로그램 분산 수행 실험

메타컴퓨팅 시스템은 MPI나 PVM을 사용한 병렬처리 프로그램을 한 스텝에 분석할 값들을 선택된 시스템을 사용하여 다시 한 번 더 병렬로 분석하게 된다. 본 절에서는 MPI와 C로 작성된 CFD 프로그램을 단일 클러스터 시스템에서 수행시킨 결과와 분산 수행 결과가 동일한지 비교하고, 각 수행 방법에 소요된 시간을 확인하였다.

실험을 위해 사용된 단일 클러스터 시스템은 Intel Pentium4-3.0GHz CPU를 사용하는 마스터 PC 1대와 CPU Intel Pentium4-2.8GHz CPU를 사용하는 계산 PC 2대로 구성하였고, 메타컴퓨팅 시스템은 AMD Athlon XP 2800+ CPU를 사용하는 PC 2개로 구성된 nodeB와 Intel Pentium4-2.8GHz CPU를 사용하는 PC 2개로 구성도니 nodeC로 구성하였다. 단일 클러스터 시스템의 경우는 마스터 PC도 작업에 참여하지만 메타컴퓨팅 시스템의 경우는 작업에 참여하지 않기 때문에 계산 PC만을 실험 결과 요인에 고려하였다. 두 시스템의 자세한 구성 정보는 다음과 같다.

[표 1] 단일 클러스터 시스템

클러스터 구성		노드 정보
HW	nodeA (3개)	Intel Pentium4-3.0GHz(M)/Intel Pentium4-2.8GHz(C)-2, 1G, 80G, Fedora Core 4 - Kernel 2.6.11-1
	MPICH	MPICH-1.2.7pl
SW	PBS	Torque-2.1.6

* (M) = 마스터 노드, (C) = 계산 노드

[표 2] 메타컴퓨팅 시스템

클러스터 구성		노드 정보
HW	nodeB (2개)	AMD Athlon XP 2800+, 256M, 80G(M)/160G(C), RedHat9-Kernel 2.4.32
	nodeC (2개)	Intel Pentium4-2.8GHz, 256M, 80G(M)/160G(C), RedHat9-Kernel 2.4.32
SW	MPICH	MPICH-1.2.6
	PBS	Torque-2.3.16
	Toolkit	Globus Toolkit-3.2.1

* (M) = 마스터 노드, (C) = 계산 노드

실험에 사용된 CFD 병렬 프로그램은 인터넷에 공개된 버전으로 난류의 흐름을 분석하는 프로그램(mpi_dust)으로 C언어로 작성된 프로그램에 MPI 코드가 더해진 병렬 프로그램이다. 실험은 mpi_duct 프로그램의 핵심 루프 횟수를 1에서부터 점차 증가시키면서 단일 클러스터 시스템과 메타컴퓨팅 시스템에서 수행시켜 분석 완료까지 소요된 시간을 측정하여 비교하였다. 그리고 두 시스템에서 생성된 결과 파일을 gnuplot을 이용하여 동일한 분석 값이 나왔는지 그래프를 확인해 보았다.

실험에 앞서서 먼저 초당 수행할 수 있는 부동소수점(floating-point) 연산 회수인 Flops를 계산하여 두 시스템의 성능을 비교해 보았다.

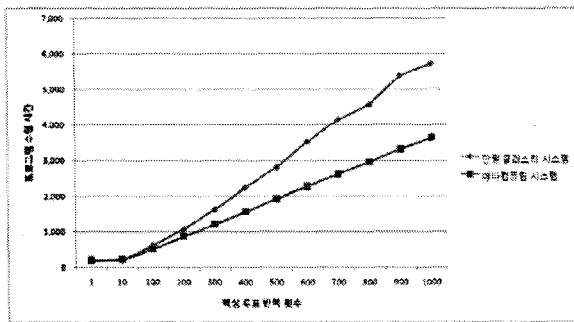
[표 3] 시스템 성능

시스템		성능(GFlops)	
단일 클러스터 시스템	nodeA	17.2	
	nodeB	6.6	17.8
메타컴퓨팅 시스템	nodeC	11.2	

Flops 수치 값 이외에도 네트워크 지연 시간, 메모리 캐시 등의 성능에 영향을 미치는 다른 요인이 있지만

nodeB를 구성하는 PC의 CPU가 nodeA, nodeC를 구성하는 PC의 CPU보다 성능이 낮은 점을 제외하고는 크게 성능에 영향을 미칠 요소가 없기 때문에 Flops 값만으로도 성능 비교가 가능하다. 표 3과 같이 17.2GFlops의 단일 클러스터 시스템(nodeA)에 비해 메타컴퓨팅 시스템(nodeB + nodeC)은 총 17.8GFlops로 0.6GFlops(≈ 3.5%) 더 높은 성능이 높지만 큰 차이가 없다는 것을 알 수 있다.

직접 CFD 프로그램 수행에 소요된 시간을 측정한 결과는 그림 5와 같으며 앞서 예상한 것과 같은 결과를 확인하였다. 하지만 반복 횟수가 증가하면 할수록 프로그램 수행에 소요된 시간 차이가 점점 커지는 것을 확인하였다. 편의상 측정 시간을 소수점 첫 번째 자리에서 반올림하였다.

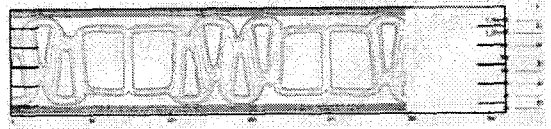


[그림 5] 반복 횟수에 따른 CFD 프로그램 수행 시간

실험 결과 단일 클러스터 시스템을 이용한 경우에 비해서 메타컴퓨팅 시스템을 이용하여 CFD 프로그램을 분산 수행시킨 경우가 최소 15.3%에서 최대 38.5%까지 시간이 단축되었다. 그리고 반복 횟수가 1,000일 때 각 결과 파일을 gnuplot을 사용하여 그래프로 보았을 때 분석 결과가 아래 그림 6, 그림 7과 같이 동일한 결과 값을 확인할 수 있었다.



[그림 6] CFD 프로그램 수행 결과 - 단일 클러스터 시스템



[그림 7] CFD 프로그램 수행 결과 - 메타컴퓨팅 시스템

5. 향후 연구 및 결론

급속도의 컴퓨팅 기술 발달에 따라 현재 일반 컴퓨터가 과거 슈퍼컴퓨터만큼의 성능을 내고 있으며, 하나의 CPU에 2~4개 코어가 들어가는 듀얼 코어나 쿼드 코어 CPU도 일반 사용자들이 쉽게 사용할 수 있다. CPU의 성능뿐만 아니라 네트워크 기술도 발달하여 현재는 일반 가정집에서도 100Mbps 속도의 네트워크망을 사용할 수 있으며, 대기업이나 대학교의 경우는 기가급 망을 사용하고 있다. 이런 기술의 발달로 어렵지 않게 슈퍼컴퓨터 성능을 낼 수 있는 클러스터 시스템을 구축하여 과거에 해결할 수 없었던 복잡한 문제들을 해결할 수 있다.

CFD 프로그램은 유체의 유동흐름을 분석하기 위해 Navier-Stokes 방정식을 이용하는 수치해석 계산 프로그램이다. 일반적으로 CFD 분석은 매우 방대한 계산량으로 인해 슈퍼컴퓨터나 클러스터 시스템을 이용하여 프로그램을 수행시킨다.

본 논문에서는 CFD 병렬 프로그램 분산 수행을 위한 그리드 컴퓨팅 기반의 메타컴퓨팅 시스템을 설계하여 CFD 프로그램 수행을 보다 빠르고 효율적으로 할 수 있는 방법을 제시하였다. 메타컴퓨팅 시스템은 그리드 컴퓨팅을 기반으로 하기 때문에 사용자가 연결되어 있는 다수의 시스템들을 마치 하나의 슈퍼컴퓨터처럼 사용할 수 있게 해주며, 사용 가능한 작업 자원들에 대해 Test-run 프로그램과 네트워크 지연 시간을 고려하여 가장 적절한 자원을 자동으로 선택하여 CFD 프로그램을 분산 수행한다. 실험을 통하여 메타컴퓨팅 시스템 이용 시 단일 클러스터 시스템을 이용하는 것보다 (얼마나 더 빠른지) 확인하였다.

향후 연구로는 메타컴퓨팅 시스템을 Globus Toolkit의 웹 서비스로 제공하는 것을 생각해 볼 수 있다.

참고 문헌

[1] Wikipedia, CFD, "http://en.wikipedia.org/wiki/Computational_fluid_dynamics"