ET 영상복원에서 클러스터 컴퓨팅에 의한 자코비안 계산의 속도 향상 기법

이 정 훈
제주대학교 전산통계학과

손 수 방 ∪
제주대학교 전산통계학과

E-Mail :{jhlee, libra}@venus1.cheju.ac.kr

A performance enhancement scheme for Jacobian matrix via cluster computing on ET image reconstruction procedure

Junghoon Lee Subang Son ∪
Dept. of Computer Science and Statistics, Cheju National University

요 약

본 논문은 차원이 큰 행렬 연산 때문에 많은 계산 시간을 필요로 하는 ET 영상 복원 응용의 속도를 개선하기 위하여 3 대의 PC로 구성된 클러스터를 구축하고 복원 과정 중 가장 많은 시간을 차지하는 자코비언 행렬 계산에 대해 병렬 계산 기법을 제시한다. 각 노드는 리눅스 운영체제, MPI, 산술 계산 라이브러리 등을 탑재하여 C 언어로 응용이 작성될 수 있으며 자코비언 행렬은 각 계산 루프의 데이터 독립성이 강하므로 병렬 계산의 장점을 최대화할 수 있다. 구현된 클러스터 자코비언 프로그램은 주어진 인자를 분석하여 MPI 프리미티브에 의해 각각의 노드에 분배시키고 각 노드들로 하여금 자신의 계산 라이브러리를 이용하여 계산하게 한 다음 이 부분 결과를 모아 최종적인 자코비언 행렬을 생성한다. 이 프로그램을 클러스터에서 수행시키고 그 수행시간을 측정한 결과 기존의 자코비언 프로그램에 비해 최대 40 % 까지 수행시간을 단축시킬 수 있었으며 추후 행렬의 차원이 증가할 경우 클러스터 컴퓨팅에 의한 성능 개선을 기할 수 있다.

1. 서론

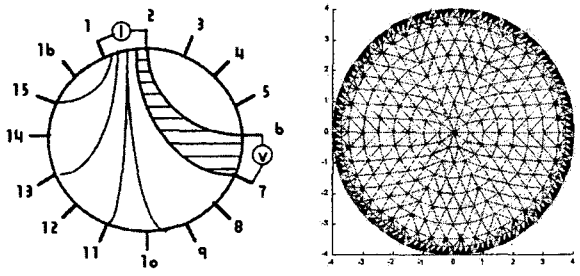
ET(Electric Tomography), 즉 전기적 단층촬영 기법은 대상이 되는 물체에 인위적인 전기신호를 주입한 후 그 물체에서 나오는 신호를 분석하여 물체의 내부를 영상화한다[1]. 이 기법은 CT(Computerized Tomography)나 NMR(Nuclear Magnetic Resonance) 등 비하여 가격이 저렴하고 데이터의 취득속도가 빠르기 때문에 최근 새로운 단층촬영 기법으로 주목을 받고 있다. 그러나 영상 복원 과정에 있어서 계산량이 막대하여 특별한 하드웨어의 도움없이 일반적인 소프트웨어 산술계산(numerical computing) 도구를 이용하여 복원 알고리즘을 수행하는 경우 상당한 시간이 소요되므로 병렬 혹은 분산 처리에 의한 계산 속도의 개선을 필요로 한다. 이러한 병렬 계산의 필요성을 충족시키기 위해 MPI(Message Passing Interface)나 PVM(Parallel Virtual Machine)과 같은 미들웨어들이 개발되어 있을 뿐 아니라 네트워크 기술의 발달에 따른 데이터 전송속도의 향상은 클러스터 컴퓨팅과 같은 병렬 분산 계산을 가능하게 한다[2].

본 논문은 ET 영상 복원의 속도를 향상시키기 위하여 네트워크로 연결된 PC들이 병렬 계산을 수행하는 계산 환경을 구축하고 이를 기반으로 영상 복원 과정 중 가장 많은 CPU 시간을 필요로 하는 자코비언 계산에 대해 병렬 계산 응용을 작성하고 이의 성능을 측정한다. 먼저 계산 환경 구축을 위해 리눅스 운영체제를 탑재한 PC들을 하나의 네트워크로 연결시키고 각 PC들에 효율적인 산술계산 라이브러리와 MPI를 설치한다. 자코비언 계산은 요소의 수만큼의 반복 루프로 구성되고 각 루프는 상당한 양의 행렬 계산을 수행하지만 각 루프의 데이터 의존성이 없으므로 효율적으로 여러 노드에 분할되어 수행될 수 있다. 이 과정에서 MPI 미들웨어는 협력계산에 필요한 자료구조들이 각 노드간에 효율적으로 교환될 수 있는 환경을 제공한다[3]. 각 노드에 작업이 분할되면 각 노드는 할당된 계산을 수행하여야 하는데 산술계산 라이브러리는 필요한 행렬 계산을 효율적으로 수행한다.

본 논문의 구성은 다음과 같다. 1장에서 문제를 제기한 후 2장에서는 ET 문제의 특성과 연구 배경에 대해 소개한다. 3장에서는 클러스터 및 소프트웨어의 구축 내역과 프로그램 환경을 설명한다. 이와 아울러 자코비언 계산 과정을 제시한 후 클러스터 환경에서의 구현 내역을 자세히 설명한다. 4장에서는 수행시간을 측정한 결과를 보이고 마지막으로 5장에서는 결론을 도출하고 추후 연구 과제를 제시한다.

2. 연구 배경

ET는 그림 1(a)에서 보는 바와 같이 물체 외부에 배치된 전극을 통해 전류를 순차적으로 주입하고 각 주입된 전류에 대해 다른 전극에서 전압을 측정한다. 이 과정에서 그림 1(b) 에서처럼 물체 내부를 가상적인 요소(element)들로 분할한 후 측정된 전압을 기준으로 각 요소들의 저항값을 계산하여 물체 내부의 특성을 파악한다. 각 요소들이 미지의 변수가 되며 전극에서 측정된 데이터들은 변수의 값을 계산하는데 필요한 방정식을 제공한다. 전극의 수가 많아지면 방정식이 많아지고 이에 의해 변수를 많게 할 수 있다. 결국 요소의 개수는 계산 시간에 큰 영향을 주며 외부 전극의 개수와 해상도, 속도 요구사항에 의해 결정된다[4].



(a) 외부 전극 (b) 메쉬 요소

그림 1. ET의 원리

영상 복원 알고리즘은 반복적인 루프로 구성되며 각 루프에서는 하나의 전극에서 주입된 신호에 의해 다른 전극들에서 측정된 신호, 즉 한 프레임의 데이터에 각 요소의 값을 계산한다. 이 과정은 그림 2에서 보는 바와 같이 FEM(Finite Element Method), 자코비언, 차이보정 등의 과정을 포함하는 복원 루프의 반복으로 구성된다. FEM은 현재 불완정하게 추정된 내부 요소들의 저항값들에 의해 사전에 정의된 전류의 패턴이 주입되었을 때 예상되는 각 전극에서의 측정치를 계산하는 과정이며 실제 측정치와 예상치의 차이가 다음 복원 루프에 반영이 된다. 자코비언은 각 요소의 저항율을 순서대로 변화시켜 가면서 이에 대응하는 경계 전압의 변화를 계산하여 경계 전압의 변화와 저항율의 변화의 비를 계산함으로써 구할 수 있다. 이때 각각의 전극에 대해 모든 전류 주입 패턴, 즉 모든 FEM 요소를 대상으로 수행된다.

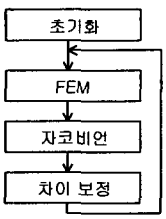


그림 2. 영상복원 알고리즘의 개요

한 반복 루프에서 처리되는 데이터들은 요소의 수에 의해 결정되는 다양한 행렬에 대한 연산으로 구성되어 있다. Pentium PC III, 128M 메모리, 윈도우용 Matlab 5.3에서 평균 50초 정도 소요되고 이중 자코비언 계산이 33초로서 전체 반복루프의 가장 큰 부분을 차지한다. 따라서 자코비언 계산의 속도를 개선하는 것이 가장 전체적인 복원 속도 향상에 필수적이다.

3. 구현

3.1. 클러스터의 구축

자코비언의 병렬 계산을 위한 클러스터는 3 대의 PC와 1 대의 스위칭 허브로 구성했다. 각 노드의 사양은 **표 1**에서 보이는 바와 같으며 각 노드는 10 Mbps 이더네트 인터페이스를 통해 DAVID SYSTEM의 DSI 허브와 연결되어 있다. 지역 클러스터 구축을 위해 노드 1은 두 개의 네트워크 인터페이스를 갖고 있는데 하나는 외부로 접근할 수 있는 인터페이스인 반면 다른 하나는 클러스터 내의 노드들을 연결하고 있어서 클러스터를 외부의 트래픽과 차단시킨다. 또한 각 노드는 Redhat 리눅스 운영체제 상에 MPI lam-6.5.6와 산술계산 라이브러리를 탑재하고 있다. 이를 기반으로 C 언어로 자코비언 응용을 작성하여 수행시킬 수 있다.

표 1. 노드의 구성표

HOST	CPU(Intel)	대역폭(Mbps)	메모리
노드1	Pentium-III 933	100(2)	256 M
노드2	Pentium-II MMX 333	100	256 M
노드3	Pentium I 200	10	96 M

3.2 자료구조의 송수신

CLAPACK과 같은 산술계산 라이브러리는 임의 크기의 행렬을 저장하기 위하여 그림 3에서 보는 바와 같이 행의 수, 열의 수 및 데이터에 대한 포인터 등을 갖고 있다. 데이터 영역에는 행렬의 각 원소들이 열 우선(column major) 방식으로 저장되어 있다. 이를 MPI를 통해 전달하려면 MPI_send 함수에 데이터 영역에 대한 포인터를 전달하여야 하며 행의 수와 열의 수를 기반으로 전달될 데이터의 크기를 계산하여야 한다. 이때 데이터 영역의 각 요소들은 C 언어의 double 형을 가지며 수신자는 수신 바이트 수를 결정하기 위해 수신될 행렬의 차원을 미리 알고 있어야 한다[5].

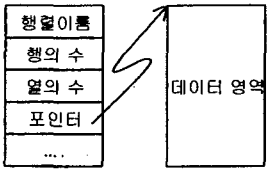


그림 3. 행렬에 대한 자료구조

각각의 노드들은 자신의 rank에 따라 자코비언 루프의

부분을 수행한다. 이 과정에서 각 노드들은 자신의 산술 계산 라이브러리를 호출하게 되며 수행의 결과로 자코비언 행렬의 일부분이 생성된다. 각 부분 수행이 종료하면 마스터 노드에 부분 결과가 보고되어 결합되는데 마스터는 사전에 자코비언 행렬의 크기에 해당하는 기억장소를 할당받은 후 각 노드의 rank 번호에 해당하는 데이터 포인터를 MPI_Receive 함수에 주어 수신한다. 각 노드에 대해 MPI_Receive 함수의 수행이 종료하면 자코비언 계산이 종료하게 된다.

3.3 자코비언의 병렬 계산

그림 4는 ET 영상복원에서 자코비언 함수가 Matlab 코드로 작성된 프로그램을 보이고 있는데 Agrad는 요소의 저항값에 영향을 줄 수 있는 주변 요소들의 영향도를 갖고 있는 희소행렬이며 rho는 현재까지 루프에서 계산된 각 요소들의 저항값이다[6]. 또 NNode는 노드의 개수를 저장하는 변수이다. U0는 주입된 신호를 각 전극에서 측정한 값인 반면 U는 각 전극에서 주입된 신호에 따른 예상 측정치로서 이의 차이가 내부 영상을 복원하는데 가장 중요한 정보가 된다. 본 논문의 영상 복원 모델은 속도계산을 목적으로 하므로 각 행렬의 차원이 중요한데 J는 992 * 776, rho는 776 * 1, U0와 U는 각각 1681 * 31, 1681 * 32이다. 그림에서 각 루프는 자코비언 행렬의 부분을 계산하는데 각 루프의 데이터 의존성이 없기 때문에 독립적으로 분산되어 수행되어도 무방하다.

```
J = zeros(992, 776);
for i=1:1:776
    J(:,i) = U0.'*1/rho(i)^2 *
          reshape(Agrad(:,i), NNode, NNode) * U;
end
```

그림 4. 자코비언 계산을 위한 Matlab 코드

분산 계산은 하나의 마스터 노드가 작업을 분리하여 몇 개의 슬레이브 노드에 나누어주고 각각 할당된 작업을 수행한 후 부분 결과를 결합하는 과정으로 구성되는데 네트워크를 통한 자료구조의 전달이 포함된다[7]. 자코비언 계산은 복원 루프의 한 부분으로 구성되는데 복원 루프가 반복됨에 따라 변경되는 것은 rho이다. 따라서 분산 계산을 함에 있어서 분산된 노드는 rho를 제외한 다른 행렬은 오프라인 시에 미리 값을 알고 있을 수 있으며 슬레이브 노드들은 rho만 새로이 수신하여 자코비언의 부분 계산을 수행한다. 부분 계산을 수행한 후 결과를 마스터에게 전달하는데 J를 결합하는 시간이 가장 많은 오버헤드를 초래할 수 있다.

4. 성능 측정

3장에서 구현된 분산 계산 방식을 클러스터에서 수행한 결과는 그림 5와 같다. 수행시간 측정은 일반 코드로 작성하는 경우, MPI에 의해 두 개의 프로세스로 나누어 모두 노드 1에서 수행시키는 경우, 노드 2에서 수행시키는 경우, 그리고 두 개의 MPI 응용을 노드에 나누어 수행시키는 경우 등 4 가지에 대해 분할 열수를 150에서 700까지 변화시켜 가며 측정하였다. 각 실험은 10회씩 반복하여 수행시간의 평균을 취하였다. 하나의 노드에서 두 MPI 응용을 수행시키는 경우는 CPU의 차이 때문에 수행시간의 편차가 발생하며 MPI 오버헤드 때문에 일반적인 응용으로 작성하는 경우보다 하나의 노드에서 자코비언을 구현하여 수행시키면 20초 정도 소요되는데 협력계산에 의해 12초까지 단축시킬 수 있다.

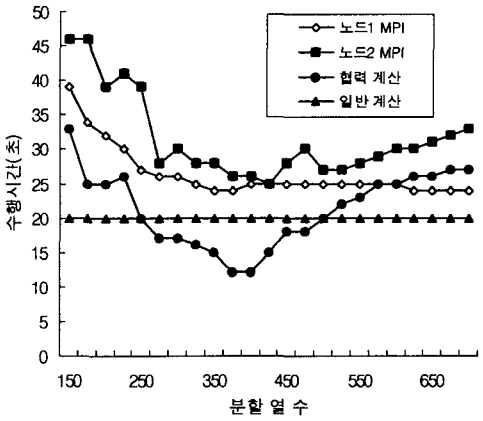


그림 5. 수행시간 측정 결과

5. 결론

본 논문에서는 ET 영상복원에서 가장 수행시간을 많이 필요로 하는 자코비언 계산을 클러스터 컴퓨팅에 의해 속도를 향상시켰다. 해상도 증대를 위해 요소의 수를 증가시키더라도 클러스터 계산에 의해 영상 복원의 속도의 개선을 기대할 수 있다. 추후 연구 과제는 자코비언 이외의 코드들을 C 언어로 구현하여 전체 루프의 수행 속도를 클러스터에 기반한 병렬 계산에 의해 개선할 예정이다. 또 노드의 속도차이를 반영할 수 있는 부하 균배 방식에 대한 연구와 도입이 수행될 예정이다.

참고문헌

[1] M. Cheney, et. al., "Electrical impedance tomography," *SIAM Review*, No, 41, pp.85-101, 1999.

[2] R. Buyya, *High Performance Cluster Computing: Programming and Applications*, Vol. 2, Prentice Hall, 1999.

[3] P. S. Pacheco, *Parallel Programming with MPI*, Morgan-Kaufmann, 1997.

[4] 과학기술부, 「이상유동장 가시화를 위한 ET 기법 개발에 관한 연구」, 2001년 7월

[5] 김철민, 이정훈, "이중 CPU PC에서 병렬 계산을 위한 Matlab 행렬 연산 라이브러리의 구현 및 성능 측정," 「정보과학회 추계학술대회(III)」, pp.871-873, 2001년 10월.

[6] M. Vauhkonen, at. al, "A Kalman filter approach to track fast impedance changes in eletrical impedance tomography," *IEEE Trans. on Biomedical Engineering*, pp.486-493, 1998.

[7] K. Wadleigh, I. Crawford, *Software Optimization for High Performance Computing*, Prentice Hall, 2000.