

병렬 연산을 이용한 방출 단층 영상의 재구성 속도향상 기초연구

서울대학교 의과대학 핵의학교실¹, 의생명과학과², 서울대학교 대학원 의용생체공학 협동과정³,
서울대학교 대학원 방사선응용생명과학 협동과정⁴
박민재^{1,3} · 이재성^{1,2,3} · 김수미^{1,4} · 감지연¹ · 이동수^{1,4} · 박광석³

Preliminary Study on the Enhancement of Reconstruction Speed for Emission Computed Tomography Using Parallel Processing

Min Jae Park, M.S.^{1,3}, Jae Sung Lee, Ph.D.^{1,2,3}, Soo Mee Kim, M.S.^{1,4}, Ji Yeon Kang, M.D.¹,
Dong Soo Lee, M.D.^{1,4}, and Kwang Suk Park, Ph.D.³

Departments of ¹Nuclear Medicine and ²Biomedical Sciences, Seoul National University College of Medicine, Seoul, Korea;
³Interdisciplinary Program, Biomedical Engineering Major, Graduate School of Seoul National University, Seoul, Korea;
⁴Interdisciplinary Program, Radiation Applied Life Science Major, Graduate School of Seoul National University, Seoul, Korea

Purpose: Conventional image reconstruction uses simplified physical models of projection. However, real physics, for example 3D reconstruction, takes too long time to process all the data in clinic and is unable in a common reconstruction machine because of the large memory for complex physical models. We suggest the realistic distributed memory model of fast-reconstruction using parallel processing on personal computers to enable large-scale technologies. **Materials and Methods:** The preliminary tests for the possibility on virtual machines and various performance test on commercial super computer, Tachyon were performed. Expectation maximization algorithm with common 2D projection and realistic 3D line of response were tested. Since the process time was getting slower (max 6 times) after a certain iteration, optimization for compiler was performed to maximize the efficiency of parallelization. **Results:** Parallel processing of a program on multiple computers was available on Linux with MPICH and NFS. We verified that differences between parallel processed image and single processed image at the same iterations were under the significant digits of floating point number, about 6 bit. Double processors showed good efficiency (1.96 times) of parallel computing. Delay phenomenon was solved by vectorization method using SSE. **Conclusion:** Through the study, realistic parallel computing system in clinic was established to be able to reconstruct by plenty of memory using the realistic physical models which was impossible to simplify. (Nucl Med Mol Imaging 2009;43(5):443-450)

Key Words: Parallel computing, image reconstruction, optimization

서 론

영상 재구성은 방사선의 투사영상을 이용하여 단층 영상

을 얻기 위한 필수과정으로, 영상의 질을 결정하는 가장 중요한 기본 과정이다. 현재 임상에서 보편적으로 사용하고 있는 영상 재구성은 여과후 역투사(FBP: filtered back-projection), 정렬 구획 기댓값 최대화(OSEM: ordered-subset expectation maximization)와 3D 행작용 기댓값 최대화 알고리즘(3D RAMLA: Row Action Maximum Likelihood Algorithm)등이다.¹⁻³⁾ FBP 영상 재구성 방법은 속도는 가장 빠르지만, 인공산물(artifact)이 많이 생겨 영상의 질이 떨어지고 품질이 좋지 못하다. OSEM 영상 재구성 방법은 속도는 FBP보다 느리지만 영상의 질이 좋아 현재 임상에서 널리 사용되고 있는 방법이다. 하지만 2D PET의 영상 재구성을 위해, 측정된 3D 데이터를 2D분포를 추정하는 간소화된 방법을 사용하기 때문에, 축 방향(axial

• Received: 2009. 6. 12. • Revised: 2009. 7. 29.

• Accepted: 2009. 9. 4.

• Address for reprints: Jae Sung Lee, Ph.D., Department of Nuclear Medicine, Seoul National University College of Medicine Seoul National University Hospital, 28 Yungun-dong, Jongno-gu, Seoul 110-744, Korea
Tel: 82-2-2072-2938, Fax: 82-2-745-2938
E-mail: jaes@snu.ac.kr

※ 이 연구는 과학재단 원자력기술개발사업 고유강점기술육성 과제 (2008-03852)에 지원을 받아 수행되었음. 병렬화에 도움을 주신 KISTI Super computing center에게 감사의 말씀을 전합니다.

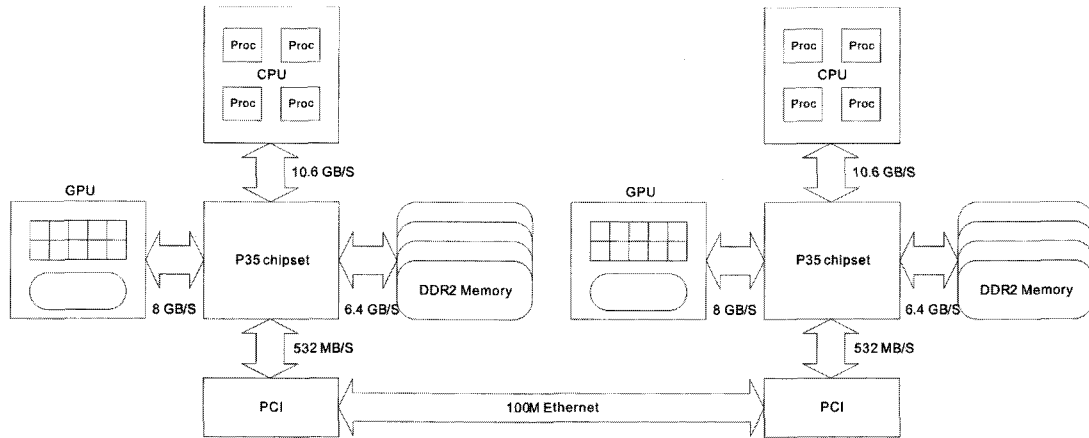


Figure 1. Data are shared between CPUs, GPUs and RAM on parallel computing. All speeds were listed on Intel P35 chipset.

direction, z축 방향)으로 불연속성이 생기는 문제가 있어 중심에서부터 멀어질수록 축방향 해상도의 저하를 가져 올 수 있다. 3D RAMLA는 완화법(relaxation)의 적용으로 좋은 영상 질을 빠르게 재구성하지만, 3D 재구성이기 때문에 아직 임상에서 모든 데이터에 적용하기에는 충분히 빠르지 않다. 또한 상용화되어 사용중인 위의 3가지 방법 외에도 많은 영상 재구성 방법들이 연구되고 있다. 감마선의 검출 시간 차이를 이용하여 양성자의 방출 위치 추정의 정확성을 높일 수 있는 TOF(Time of Flight) 리스트 모드(list-mode) 영상 재구성 방법은, 영상 대조도와 신호대 잡음비는 좋아지지만 시간이 많이 걸려서 임상적으로 적용하는데 있어 불편함이 많다. PET의 가공되지 않은 기본 저장 방식인 응답 라인(LOR: Line of response)을 이용하는 영상 재구성 방법이나 재구성 영상의 중앙이 아닌 곳에서 공간 분해능(spatial resolution)을 개선하고 부분체적효과(partial volume effect)에 의한 계수를 저하를 보정(count recovery)하는데 좋은 결과를 보여주는 점 퍼짐 함수(Point spread function, PSF)를 투사기(projector)와 역투사기(back-projector)로 이용하는 영상 재구성 방법들이 우수한 성능을 보이고 있다. 그러나 역시 속도와 메모리 문제로 인해 모든 위치에서의 PSF를 저장하여 이용하는 것이 현대의 일반적인 재구성 머신으로는 불가능하기 때문에, PSF함수를 모델링하여 어느 정도의 오차를 포함하고 있는 간소화 방법을 사용해 왔다.⁴⁻⁶⁾ 이렇듯 고해상도 영상을 위해서는 많은 영상 재구성 시간이 걸리기 때문에 알고리즘 향상뿐만 아니라 하드웨어적으로도 병렬화 및 최적화하여야 한다.

본 논문에서는 메모리 사용이 자유롭고 연산장치 추가가

용이한 분산 메모리 방식의 병렬연산을 이용하여, 방출 단층 촬영 영상 재구성의 속도 향상에 대한 연구를 하였다.

대상 및 방법

1. 병렬 연산의 종류

여러 개의 연산장치를 동시에 사용하기 위한 방법으로는 공유메모리 방식, 분산 메모리 방식, 그리고 그래픽 처리 카드(GPU:Graphic Processing Unit)를 이용한 GPGPU(General Purpose GPU)방식이 대표적이다. Fig. 1은 각종 병렬 연산에서 사용하는 데이터의 흐름을 개인 컴퓨터를 예를 들어 나타낸 것이다.⁹⁾ 공유메모리 방식은 한 컴퓨터 내에서 여러개의 연산장치가 하나의 메모리를 공유하여 각각 독립적으로 연산하는 것으로 Open Source의 OpenMP 패키지가 대표적이다. 현재 4개의 연산장치(Quad-core CPU)까지는 쉽게 사용이 가능하지만, 그 이상으로 확장을 하려면 가격이 기하급수적으로 상승하여 대규모 연산에는 적합하지 않고, 상대적으로 적은 수의 연산장치와 메모리 공유로 인한 소규모의 빠른 연산 작업에 적합하다. 분산 메모리 방식은 프로세스마다 별도의 메모리를 가지고 있는 방식으로 메시지 패싱 인터페이스(MPI) 패키지가 대표적이다. 분산 메모리 방식은 네트워크를 통해 쉽게 확장이 가능하여 연산장치와 메모리를 충분히 사용할 수 있는 장점이 있지만, 연산장치 사이에 통신이 필요하기 때문에 빠른 네트워크 속도가 필수적이다. GPGPU 방식은 느리지만 많은 수의 연산장치로 이루어진 별도의 장비를 이용하는 것으로 NVidia사의 CUDA(Compute Unified Device Architecture) 패키지가 대표적이다. GPGPU 방식은 시스템 버스를 이용하기 때문에, 네트워크 카드를 이용한 분산 메모리 방식보다

는 빠르고 메모리와 CPU사이의 통신만 필요한 공유 메모리 방식보다는 느리다. GPGPU 방식은 연산 중심적인 작업에는 많은 수의 프로세서를 이용할 수 있어 유리하지만, 메모리 사용량이 많은 연산에는 작은 메모리 양으로 인해 잦은 데이터 전송이 필요하다. 분산 메모리 방식의 병렬화가 다른 방식에 비해 가장 큰 이점은 메모리를 충분히 사용할 수 있어, 반복 사용되는 큰 사이즈의 system matrix를 미리 메모리에 저장하여 계산시간을 줄이는 방법이 사용가능하다. 투사 확률을 연산장치마다 메모리(LUT : Look-up-table)에 나누어 저장을 한 다음, 투사 확률을 계산하는 시간 보다 빠른 메모리 접근 시간으로 반복 연산시의 속도를 향상시켰다. 하지만 3D PET에 대한 LOR을 이용한 방법은 수십 기가 바이트정도의 메모리를 필요로 하기 때문에 일반 컴퓨터로는 실행 불가능하였다.

2. 연산 시스템 구성

테스트 컴퓨터는 3.2GHz의 속도를 가지는 연산장치(processor) 4개와 2GB의 메모리를 가지고 있는 개인 컴퓨터이다. 512MB의 메모리를 가지는 두 개의 듀얼 코어 시스템을 가상(Emulation)하였고, 가상 시스템에서도 속도 저하가 없게 하는 Intel사의 Virtualization Technology(VT)로 인해 성능저하가 없는 것을 확인하였다. 두 개의 시스템에 Debian Linux etch를 바탕으로 GCC(GNU C Compiler)와 MPICH2를 설치하였고, 100M 이더넷 방식의 네트워크에서 NFS(Network File System)를 연결하였다. MPICH2는 MPI를 이용한 패키지 중 가장 속도가 빠른 패키지로, 가상으로 공유메모리 모델을 구현하는 MPI2를 구현한다. Daemon을 이용하여 MPI프로그램을 동시에 실행하도록 구성하였다. NFS를 이용하여 프로그램과 데이터의 동질성을 유지시켰다.

다른 cluster들과의 객관적인 비교를 위해, 한국과학기술정보연구원(KISTI) 슈퍼 컴퓨팅 센터에 있는 Tachyon슈퍼컴퓨터를 이용하였다. AMD opteron의 쿼드코어가 4개, 총 16개의 연산장치가 한 개의 컴퓨팅 노드에 있고, 총 188개의 컴퓨팅 노드로 이루어져 총 3008개의 연산장치로 이루어져 있다. Redhat linux 중의 하나인 CentOS 4.4을 기본으로, GNU, PGI, Intel의 compiler와 MVAPICH, OpenMPI의 MPI 패키지가 설치되어 있다. PGI compiler는 Portland Group에서 나오는 상용 컴파일러이고, MVAPICH와 OpenMPI는 MPI 패키지 중 하나이다. 노드 사이는 InfiniBand라는 전용 직렬 통신망을 통해 연결되어 있다.

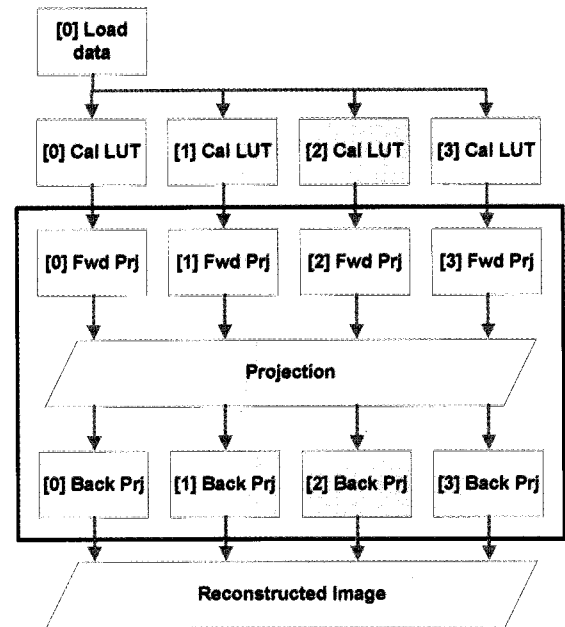


Figure 2. LUT and projection of EM reconstruction were parallelized. The image was sliced for Y-axis.

3. 영상 재구성 프로그램

실험에 사용된 영상 재구성 프로그램은 임상에서 널리 사용되는 EM 알고리즘으로, 영상품질은 좋지만 재구성 시간이 오래 걸려 병렬화가 필수적이다. EM 알고리즘은 다음과 같다.

$$A_i^{n+1} = \frac{A_i^n \sum_j M_{ij} S_j}{\sum_j M_{ij} \sum_i M_{ij} A_i^k} \quad (1)$$

A는 추정영상내의 방사능량(activity in estimating image), M은 i번째 픽셀에서 j번째 bin으로 투사될 확률(system matrix), S는 투사 영상내의 카운트(count in sinogram), i는 추정영상의 인덱스, j는 투사영상의 인덱스이다. 병렬화가 가능하려면 서로 독립적인 연산이어야 하기 때문에, 위의 식에서는 연산장치 별로 영상을 여러 구획으로 나누어 투사와 역투사를 하는 것으로 Fig. 2와 같이 병렬화하였다.

투사와 역투사에 사용된 투사 확률은 보간법을 이용한 화소 구동 방법 (PDM: Pixel driven method)과 Siddon의 광선 길이를 계산하는 방법(RTM: Ray-tracing method)으로 계산하였다.⁷⁾ 화소 구동 방법은 화소의 무게 중심이 각도에 따라 검출기에서 흡수되는 위치를 추정하는 방법이다. 선추적법 (Ray driven method)은 각 화소가 차지하는 공간을 지나는 방사선의 길이를 추정하는 방법이다.

Sinogram과 LOR (Line of response) 형식의 투사 데이터를 이용하였다. Sinogram은 SPECT의 원 데이터의 형식으로, 영상재구성에서 사용되는 가장 기본적인 형식이다. 각 픽셀에서 sinogram으로 투사되는 비율을 LUT에 저장하기 위해서는 다음과 같은 크기의 메모리가 필요하다.

$$X \times Y \times ang \times kernel \times type \times [pos, weight] \quad (2)$$

X, Y는 재구성된 영상의 크기, ang는 투사 각도의 개수, kernel은 보간의 범위, type은 데이터 형, pos는 검출 위치, weight는 검출 확률이다. LOR형식의 투사 데이터는 PET의 원 데이터 형식으로, sinogram형식보다 많은 정보를 가지고 있지만 큰 사이즈의 메모리를 필요로 한다. 각 픽셀에서 3차원 LOR으로 투사되는 비율을 LUT에 저장하기 위해서는 다음과 같은 크기의 메모리가 필요하다.

$$LOR \times [x, y, z, weight] \times type \times intersect_{max} \quad (3)$$

LOR은 crystal-pair(LOR) 개수, [x,y,z]는 화소의 위치, $intersect_{max}$ 는 광선이 화소를 지나가는 지점의 최대 개수이다.

4. 영상 재구성의 평가 시스템

다음의 두 가지의 시스템에 대한 재구성 알고리즘을 대상으로, mathematical phantom을 사용하여 PDM과 RTM으로 각각 sinogram과 LOR 데이터를 얻어 테스트 하였다.

1. SPECT 시스템 상에서 2D PDM으로 미리 계산한 투사확률을 LUT에 저장하였다. 120개의 각도와 128개의 bin을 가지는 sinogram형태의 투사 데이터를 이용하여 128×128크기의 mathematical 2D Hoffman phantom영상을 재구성 하였다.

2. SNU animal PET 시스템 상에서 1273만개의 LOR 투사 데이터를 이용해서, 128×128×57크기의 mathematical cylinder phantom영상을 재구성하였다.⁸⁾ 3D RTM으로 계산한 투사확률의 개수가 컴퓨터 메모리의 양보다 많아서, LUT(look up table)에 저장하지 못하고 반복연산할때마다 계산하였다.

5. 병렬화 속도 향상 평가

전체 걸린 시간, 1회 반복당 걸린 시간, 계산 과정마다 걸린 시간을 기록하여 비교하였다. 전체 걸린 시간은 EM 반

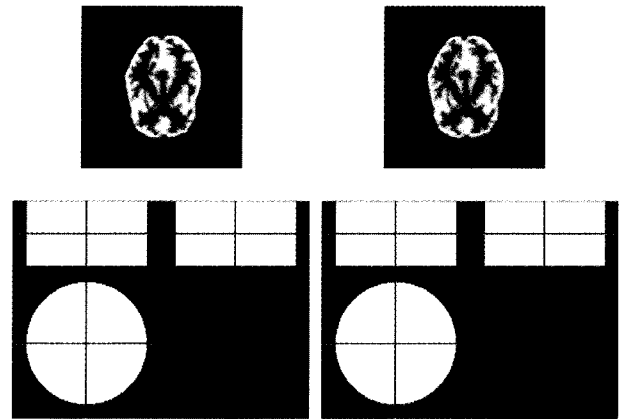


Figure 3. Reconstructed images using parallel computing(left) and single computing(right). Total absolute-difference is below floating-point error.

복연산의 시작과 끝 사이의 시간으로 실제 재구성할 때 걸리는 시간이라고 생각할 수 있다.

재구성 알고리즘의 차이가 아닌 컴파일러와 최적화 기법에 따른 연산 속도의 차이도 측정하였다. GCC, PGI와 Intel 컴파일러에서의 속도와 unrolling, vectorization과 같은 최적화 기법의 유무에 따른 속도 차이를 비교하였다. Unrolling 기법은 루프 반복에 따른 추가 자원을 줄이는 방법으로, 분기와 분기사이에 더 많은 명령어를 넣어 분기 횟수를 줄인다. Vectorization 기법은 루프를 재정렬하여 블록단위로 계산하는 방법으로, 멀티 미디어 데이터 처리에 주로 사용되는 SSE(Streaming SIMD Extensions)와 같은 CPU의 기능과 같이 사용된다.⁹⁾ 네트워크, CPU와 같은 시스템에 의한 속도차이는 복잡적이어서 비교하지 못하였다.

결 과

Linux 운영체제에서 MPICH 라이브러리와 NFS를 이용하여, 여러 컴퓨터에서 하나의 프로그램으로 분산 연산이 가능하였다. 한 컴퓨터 내에서 가상으로 구성한 cluster의 평균 네트워크 응답속도는 0.278±0.025 msec, 두 개의 쿼드 코어 컴퓨터들 사이에는 0.234±0.022 msec으로 속도차이가 나지 않았다. Tachyon에서는 평균 0.149±0.012 msec으로 두배 정도 빨리 전송할 수 있어 네트워크로 인한 부하가 적게 걸리는 것을 알 수 있다.

여러 연산장치로 나눠서 연산을 하더라도 EM연산의 반복횟수가 동일할 때는 재구성된 영상간의 차이가 없음을 우선 확인하여야 한다. Fig. 3의 아래는 값이 1로 이루어진

Table 1. Total Time of 32 Iterations as the Increments of Process using Default Compiler Option.

CPUs ^{a)}	1	2	4	8	16	32
GCC ^{b)}	3.84	1.92	0.99	0.50	0.27	0.14
PGI ^{c)}	1.89	0.95	0.50	0.26	0.14	0.07
Intel	1.01	0.52	0.36	0.19	0.11	0.05

^{a)}Number of CPU(central processing unit), ^{b)}GNU C compiler, ^{c)}C Compiler of Portland Group, Incorporated

Table 2. Total Time of 32 Iterations as the Increments of Process Using Optimization of the Compiler

CPUs ^{a)}	1	2	4	8	16	32
GCC ^{b)}	0.87	0.45	0.34	0.19	0.11	0.05
PGI ^{c)}	1.34	0.69	0.37	0.2	0.12	0.06
Intel	1.61	0.81	0.48	0.25	0.14	0.07

^{a)}Number of CPU(central processing unit), ^{b)}GNU C compiler, ^{c)}C Compiler of Portland Group, Incorporated

백만개 정도의 화소수를 가지는 3차원 cylindrical phantom의 LOR 재구성한 영상으로, 병렬 재구성한 영상(좌)과 하지 않은 영상(우)이다. 이 두 영상의 픽셀마다의 절대 차이를 합하면 0.5정도이다. 이를 화소수로 나눈 "픽셀당 평균 오차"는 실수(floating-point number)의 유효숫자인 6자리보다 작은 것으로, 모든 컴퓨터에서 발생하는 실수연산 오차이기 때문에 무시해도 되는 수준이다.

병렬화의 효율을 계산하기 위해, GCC를 이용하여 최적화를 하지 않고 32번 반복 계산하여 재구성 하였다. 1번 시스템(2D sinogram)에서 병렬화하지 않았을 때의 연산 계산 시간은 4.12~4.25초, 2개의 연산장치를 사용하였을 때는 2.38~2.44초가 소요되었고, 4개의 연산장치를 사용했을 때는 1.62~1.68초가 소요되었다. 2번 시스템(3D LOR)에서는 병렬화하지 않았을 때의 연산 계산 시간은 4912초, 2개의 연산장치를 사용하였을 때는 2611초가 소요되었고, 4개의 연산장치를 사용했을 때는 1375초가 소요되었다. 듀얼 코어나 쿼드코어를 가진 일반 개인 컴퓨터로도 병렬화 하지 않았을 때에 비해 57%와 27%정도로 시간을 줄일 수 있었다.

컴파일러의 종류와 최적화 기법에 따른 속도차이는 Tachyon에서 비교하여 결과는 Table 1과 같다. GCC, PGI, Intel 컴파일러에 따라 1개의 연산장치로 3.84, 1.89, 1.01초가 각각 소요 되었고, 연산장치 두배씩 증가함에 따라 평균 0.53±0.05배 정도로 줄일 수 있었다. 컴파일러마다 같은 최적화 기법을 사용한 결과는 Table 2와 같다. GCC, PGI, Intel 컴파일러에서 0.87, 1.34, 1.61초가 각각 소요 되었고, 연산 장치의 두배씩 증가함에 따라 평균 0.55±0.07배 정도로 줄었다. 최적화로 인한 속도 증가가 GCC를 사용했을 때

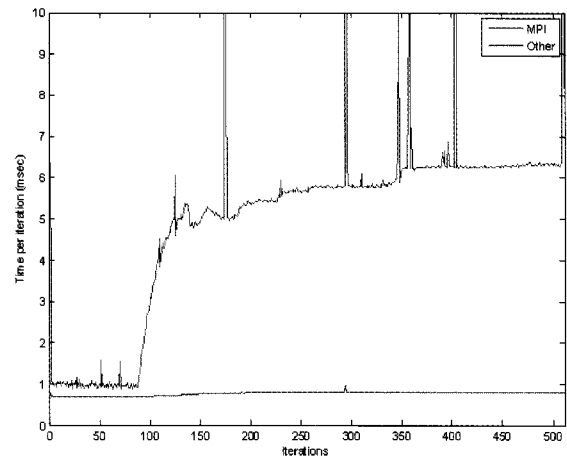


Figure 4. MPI transfer was getting slower after an iteration. 15360 bytes were transferred via MPI.

3.25±0.86배였고, PGI 컴파일러는 1.30±0.11배였다. 반면, Intel 컴파일러는 0.71±0.07배로 오히려 느려지는 결과를 보여주었다.

반복 연산 횟수가 증가함에 따라 MPI 통신 속도가 최대 6배 정도까지 느려지는 현상이 테스트한 모든 시스템에서 여러번 반복 실험에 모두 나타났었다. Fig. 4은 반복 연산마다 MPI 통신하는 부분에서 소요된 시간을 그린 것으로, 반복 수(iteration number)가 증가함에 따라 매회 반복 할 때마다 걸린 시간이 전체적으로 점점 늦어지고, 연산장치 사이에도 시간차이가 많은 것을 알 수 있다. 위의 그래프에서 x축은 각 연산장치를 뜻하고 y축은 매회 반복 할 때마다 걸린 시간을 평균한 시간이다. 아래 그래프에서 x축은 반복횟수, y축은 매회 반복 할 때마다 걸린 시간을 뜻한다. 그래

Table 3. Speed-up of Each Optimization Technique

	MPI ^{a)}	Other	Total
No Optimization	2.34	1.99	5.42
Unrolling	2.51	2.06	5.68
Vectorization using SSE ^{b)}	0.81	0.38	2.34

^{a)}Time for Message Passin Interface transfer, ^{b)}Streaming SIMD(Single Instruction Multiple Data) Extensions

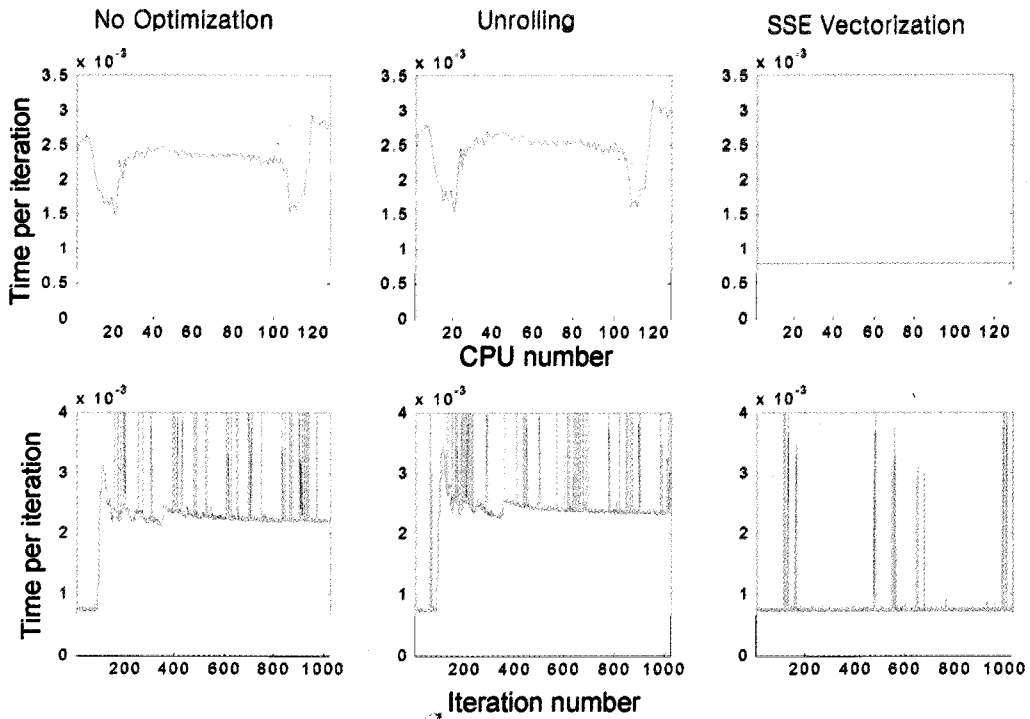


Figure 5. Proper optimization of compiler reduced the transfer time via MPI. Upper graphs were MPI time of each process. Lower graphs were MPI time of each iterations. SSE v timezation was effective to the delay phenomenon.

프에서 피크는 다른 유저와 같이 사용하는 공용 컴퓨터의 특성상 어쩔 수 없이 발생하는 발생할 수 있는 일시적 속도 저하현상이다. Fig. 5와 Table 3는 4개 CPU를 사용하여 128번 반복 연산할 때, 1회 반복마다 MPI 통신과 그 외의 걸린 시간을 최적화 방법에 따라 측정된 것이다. 최적화를 하지 않을 때는 5.42±0.01초가 걸렸지만, Unrolling방법만을 사용했을 때는 5.68±0.04초가 걸려서 유의미하게 느려졌다. 반면 SSE를 이용한 Vectorization방법만을 사용했을 때는 속도 저하 현상이 없어져서 2.34±0.04초가 걸려 유의미하게 빨라졌다. 빨라진 부분을 살펴보면, MPI 통신과 그 외 부분이 다 빨라졌다. 이것으로 단순한 전송속도가 문제가 아니라, MPI 통신을 하면서 수행하는 더하기 연산에서 느려졌기 때문이라고 결론 내릴 수 있다.

고 찰

최근에는 컴퓨터 산업의 눈부신 발전에 힘입어, 저렴한 가격의 개인용 컴퓨터도 한 컴퓨터 내에 여러 개의 연산장치(processor)를 가지고 있는 것이 보편적이다. 하지만 여러 개의 연산장치를 제대로 활용할 수 없는 프로그램을 사용하면, 좋은 성능의 컴퓨터라 할지라도 단일 연산장치 이상의 성능향상은 얻을 수 없다. 그리고 소프트웨어 방식의 속도 향상 방법은 하나의 하드웨어 위에서 돌아가기 때문에 향상에 한계가 있다. 소프트웨어 방식으로 속도 향상을 이루었을 경우라도 하드웨어 방식을 추가적으로 적용하면 그 이상의 속도 향상을 이룰 수 있다. 이에 컴퓨터의 다중 연산장치에 따른 성능향상을 효율적으로 구현하기 위한 프로그램의 중요성도 커지고 있다. 효율적인 병렬화 뿐만 아니

라 컴파일러의 최적화 옵션에 따른 재구성 속도차이도 큰 것으로 나타났다. 이는 intel compiler의 기본 옵션에 이미 최적화가 포함되어 있는 것으로 추정되고, 부가적인 최적화 작업이 더 안좋은 결과를 가져올 수 있기 때문에 알고리즘에 따른 효율적인 최적화 기법을 테스트해봐야 한다.

단일 컴퓨터로는 불가능한 복잡한 영상 재구성을 가능하게 하는 것이 분산 메모리 모델을 이용한 병렬 컴퓨팅의 가장 큰 장점이다. 보다 정확한 원영상을 재구성하기 위해, 감쇠(Attenuation), 산란(Scatter), 양전자 포획거리(Positron annihilation range), 환자의 움직임등과 같은 측정할 때 일어나는 물리적인 요소들을 영상 재구성에 포함되어야 한다. 이러한 물리량을 영상 재구성시에 반복 계산하는 것보다, 미리 계산하여 반복 연산할 때 메모리에서 불러서 사용하는 것이 훨씬 빠르다. 하지만 이러한 물리적인 요소들을 저장해서 사용하기에는 작은 메모리를 사용하는 일반 컴퓨터로는 불가능하고, 큰 규모의 컴퓨터가 필요하다. 하지만 현실적으로는 영상 재구성 하는데 슈퍼컴퓨터를 독점적으로 사용할 수 없고, 연동하기가 힘들기 때문에, 소규모 cluster를 구성하여 사용하는 것이 현실적인 대안이다.

개인용 컴퓨터에서 사용가능한 Linux에서 NFS, GCC와 MPICH2를 이용하여 cluster를 구성하였다. 슈퍼컴퓨터에 비해 상대적으로 낮은 성능의 개인용 컴퓨터들을 이용한 병렬 컴퓨팅이 가능해져서, 컴퓨터의 빠른 발전으로 인한 교체 비용도 가장 현실성이 있다. 따라서 각 병원에서 사용가능한 병렬 컴퓨팅을 이용한 영상 재구성 방법으로는 분산 메모리 모델을 이용할 것을 권장한다.

영상 재구성은 병렬 컴퓨팅에 적합하다는 것을, 반복연산시 프로세스당 걸린 시간이 연산장치 개수가 증가함에 따라 줄어드는 것으로 알 수 있었다. 연산장치 개수가 두배로 증가함에 따른 연산 속도 향상 비율은 1.91 ± 0.16 배 정도로 병렬화 효율이 좋았다. 이것은 병렬화된 연산의 독립성이 있었기 때문이며, 각 연산장치마다 연산의 적절한 분배가 이루어졌음을 알 수 있다.

빠른 영상 재구성을 위해서는 프로그램의 병렬화 이외에도 고려해야 할 것도 있었다. 컴파일러에서 최적화 기법을 적절히 사용하지 않으면 오히려 속도가 더 느려지기도 하였다. SSE를 이용한 vectorization 기법은 MPI 통신 속도가 느려지는 현상을 해결해주었다.

이번 연구를 통해 현실적인 병렬 컴퓨터 시스템을 구성

하여, 불가능 하였던 영상 재구성 방법을 실사용 가능하게 되었다. 슈퍼 컴퓨터와 같은 특수한 장비가 아니더라도 사용가능하게 되어, 실제 임상에서도 업체에서 제공받은 방법이 아닌 최신 기술의 혜택을 받을 수 있으리라 생각된다.

요 약

목적: 기존의 영상 재구성은 간소화된 투사 물리 모델을 사용하고 있다. 하지만 3D 재구성과 같은 실제적인 물리 모델은 시간이 많이 걸려서 임상에서 모든 데이터에 적용하기 힘들고, 복잡한 물리모델을 설명하기 위해 큰 메모리를 사용하면 한대의 일반적인 재구성 머신으로는 불가능하다. 개인 컴퓨터들에서도 큰 규모의 기술을 가능하게 하기 위해, 병렬 연산을 이용한 빠른 재구성의 현실적인 분산메모리 모델을 제시한다. **대상 및 방법:** 실제로 구현하는 가능성을 보기 위해 가상 컴퓨터들을 이용하여 선행 연구를 진행하였고, 다양한 가능성을 테스트하기 위해 상용서비스를 하고 있는 슈퍼컴퓨터(Tachyon)에서 성능 테스트를 하였다. 가장 많이 사용되는 2D 투사 영상과 실제적인 물리 모델인 3D 응답라인을 이용한 기댓값 최대화 알고리즘을 테스트하였다. 스터디 중 특정 반복횟수 이후에 속도가 최대 6배까지 느려지는 현상이 발견되어 컴파일러 최적화를 통해 병렬 효율의 극대화를 꾀하였다. **결과:** Linux에서 MPICH와 NFS를 이용하여, 여러 컴퓨터에서 하나의 프로그램으로 분산 연산이 가능하였다. 병렬 연산을 했을 때 동일한 반복 연산에서 재구성된 영상간의 차이가 실수의 유효숫자(6bit) 정도임을 확인하였다. 2배의 연산장치를 사용했을 때 1.96 배의 좋은 병렬화 효율을 보여주었다. 반복 연산 횟수가 증가함에 따라 느려지는 현상은 SSE를 이용한 Vectorization 방법을 사용했을 때 해결할 수 있었다. **결론:** 이번 연구를 통해 일반 컴퓨터들을 이용한 현실적인 병렬 컴퓨터 시스템을 구성하여, 작은 메모리의 단일 일반 컴퓨터로는 불가능한 간단화 할 수 없는 복잡한 물리 과정도 영상 재구성 방법에 사용 가능하게 되었다.

References

1. Lange K, Carson R. EM reconstruction algorithms for emission and transmission tomography. *J Comput Assist Tomog* 1984;8: 306-16.
2. Shepp LA, Vardi Y. Maximum Likelihood Reconstruction for Emission Tomography. *Medical Imaging, IEEE Transactions on* 1982;1:113-22.
3. Browne J, de Pierro AB. A row-action alternative to the EM algorithm for maximizing likelihood in emission tomography.

- Medical Imaging, IEEE Transactions on* 1996;15:687-99.
4. Snyder DL, Thomas LJ, Ter-Pogossian MM. A Mathematical Model for Positron-Emission Tomography Systems Having Time-of-Flight Measurements. *Nuclear Science, IEEE Transactions on* 1981;28:3575-83.
 5. Snyder DL, Polite DG. Image Reconstruction from List-Mode Data in an Emission Tomography System Having Time-of-Flight Measurements. *Nuclear Science, IEEE Transactions on* 1983;30:1843-9.
 6. Conti M, Bendriem B, Casey M, Chen M, Kehren F, Michel C, et al. First experimental results of time-of-flight reconstruction on an LSO PET scanner. *Physics in Medicine and Biology* 2005; 19:4507.
 7. Siddon RL. Fast calculation of the exact radiological path for a three-dimensional CT array. *Med Phys* 1985;12:252-5.
 8. Hong SJ, Kwon SI, Ito M, Lee GS, Sim KS, Park KS, et al. Concept Verification of Three-Layer DOI Detectors for Small Animal PET. *Nuclear Science, IEEE Transactions on* 2008;55: 912-7.
 9. PortlandGroup. PGI User's Guide <<http://www.pgroup.com/doc/pgiug.pdf>>.
 10. Intel. Intel® P35 Express Chipset <<http://www.intel.com/Products/Desktop/Chipsets/P35/P35-technicaldocuments.htm>>.