계산 그리드 기반 의료영상 저장시스템
(Medical Image Storage System based on Computational Grid)

안 병 규 † 박 재 현 ††
(Byoungkyu Ahn) (Jae-Hyun Park)

요 약 대형병원에서 환자의 치료와 진단을 목적으로 하루에 생산되는 의료영상의 발생량은 보다 정확하고 정밀한 진단이 요구되는 촬영장비와 네트워크 인프라의 발달로 나날이 증가하고 있으며, 앞으로 그런 추세는 계속 될 것이다. 따라서 기존방식의 PACS보다 성능이 개선된 시스템이 요구된다. 본 연구에서는 영상압축속도를 개선하기 위해, 계산그리드 기술을 이용하여, PACS의 부분시스템으로써, 의료영상압축저장시스템을 설계 구현하였다. 제안된 시스템의 시작품을 사용한 실험을 통해, 처리기들이 추가됨에 따라 성능이 향상됨을 확인하였다.

키워드 : 계산 그리드 컴퓨팅, 의료영상저장시스템, PACS, DICOM

Abstract The use of medical imaging in hospitals is being gradually increased as it is of utmost importance in treatment and diagnosis of patients. With the drastic increase of the usage of medical imaging in hospitals per day necessitates more speedy and accurate systems for precise diagnosis and the treatment. Hence the modality and development of network infrastructure are also need to be improved day by day and this trend may be continued. Thus there is a great need improvement of PACS concerned. In this paper, by using the computational grid technology, we design a medical image storage system that improve the compression speed, and implement a prototype as a part of PACS. We also demonstrate the performance improvement from experimental results of the prototype.

Key words : Computational Grid Computing, Medical Image Storage System, PACS, DICOM

1. 서 론

최근 병원에서는 의료영상저장전송시스템(PACS: Picture Archiving Communication System)을 사용하여, 촬영장비(Modality)로부터 발생하는 디지털 의료영상을 압축, 저장, 보관하며, 의료진들이 이를 사용하여 진료업무를 수행한다[1].

PACS는 다이콤(Digital Imaging and Communication in Medicine: DICOM)[1]이라는 의료영상표준을 기반으로 구성된다[2]. PACS 구성하는 각 모듈들은 표준에 의거하여 개발이 되기 때문에 각 모듈별로 개발업체들이 달라도 상호 호환되며, 나아가서는 다른 병원정보시스템과도 연동이 가능하다.

현재 국내의 많은 병원들은 PACS를 도입하여 운영하고 있으며, 2000년부터 해마다 보급률이 급격히 상승하고 있다. 일본에서도 1982년부터 PACS가 보급되기 시작하여 1997년 이후 보급률은 해마다 계속해서 늘어나고 있다[3]. 비단 한국 일본 뿐만 아니라 대만, 미국, 독일, 중국, 스페인 등 많은 나라의 여러 병원에서 PACS를 도입하고 있다.

그런데, 병원에서 발생하는 디지털의료영상데이터의 양과 발생 속도는 해마다 급격히 증가하고 있다. 이는 보다 정밀하고 정확한 진단을 필요로 하는 촬영장비의 발달과 컴퓨터네트워크 인프라의 성능향상과 맞물려 더욱 가속화되고 있다. 이러한 추세에 대응하기 위해 업계에서는 하드웨어적인 측면과 소프트웨어적인 측면에서 다양한 솔루션을 찾고 있으며, 최근 그리드 컴퓨팅(Grid

† 정 회 원 : 이우테크놀로지 소프트웨어연구소 3D 영상연구부 대리
bkahn@wm.cau.ac.kr
†† 정 회 원 : 중앙대학교 공과대학 컴퓨터공학부 교수
hyunie@cau.ac.kr
논문접수 : 2009년 7월 27일
심사완료 : 2009년 8월 21일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제10호(2009.10)

Computing)[4] 기술을 활용하는 연구가 활발히 진행되고 있다[2,5-7].

그리드 컴퓨팅은 네트워크상의 분산된 여러 형태의 유휴자원을 묶어 하나의 가상화된 고성능 컴퓨터시스템을 형성하여 그동안 해결하기 힘들었던 많은 양의 연산을 병렬로 수행할 수 있다[4]. 지금까지의 그리드 컴퓨팅 기술을 이용한 PACS에 대한 연구는 분산된 저장장치들을 묶어 하나의 가상화된 시스템을 구성하여 데이터 저장공간을 늘리는 데이터그리드(Data Grid)[8] 중심으로 진행되어 왔으며 계산그리드(Computational Grid)[8]에 중심을 두고 진행된 연구는 찾아볼 수 없었다. 하지만, 향후 PACS의 처리속도의 개선을 위해서는 계산그리드가 활용될 수 있다[10].

효율적인 저장과 전송을 위한 압축의 수행속도는 환자에 대한 진단과 치료 속도의 개선에 중요한 영향을 미친다. 또한 이는 전체적인 PACS 성능에 중요한 영향을 미치는 요소이다. 더구나 그리드 기술은 병원에서 사용하는 기존의 컴퓨터들을 이용하여, 추가적인 하드웨어 비용 없이 시스템 성능을 개선시킬 수 있는 장점을 갖고 있다. 본 연구에서는 병원에서 운영하고 있는 기존의 PACS의 성능향상과 확장성의 개선을 위해 계산 그리드 기술 사용하여 그리드 PACS의 시작품을 구현하고, 실측을 사용하여 성능을 평가한다.

본 논문의 나머지는 다음과 같이 구성되어 있다. 2장에서는 데이터 그리드 관점에서 개발된 Grid-PACS에 대한 연구를 살펴보고 PACS에서 계산 그리드 연구의 필요성을 설명한다. 3장에서는 설계 구현된 계산그리드 컴퓨팅 기술을 적용한 의료영상압축저장 시스템을 제시한다. 4장에서는 본 논문에서 제안한 시스템의 성능을 측정 및 평가하고, 5장에서는 결론을 맺는다.

2. 관련연구

MammoGrid는 그리드 기술을 이용하여 유럽지역의 분산된 각 병원의 맘모그램(Mammogram) 데이터베이스를 상호 접근가능하게 하여, 공동작업할 수 있도록 지원하는 프로젝트 이다. 이 프로젝트는 영국의 Addenbrookes 병원, 이탈리아의 Udine 병원, 영국의 옥스포드대학교를 하나의 가상시스템으로 구성한 MammoGrid Virtual Organisation(MGVO)를 구현하였다. MGVO는 EGEE-gLite 미들웨어 컴포넌트인 ALICE Environment(AliEn)을 이용하여 구현하였으며, gLite는 EU에서 투자한 EGEE 프로젝트에서 개발된 그리드 미들웨어다. Addenbrookes 병원과 이탈리아의 Udine 병원은 수 천명분의 케이스에 대한 맘모그램 데이터를 보유하고 있으며, MGVO에 등록된 의사들은 각 지역에 분산된 맘모그램 데이터 및 진단결과 등을 한 곳에서 모두 열람할 수 있다[7].

데이터 그리드를 이용한 의료영상복구시스템은 지진, 불, 홍수 등의 재난에 대비하여 PACS에 저장된 의료영상 데이터의 손실을 방지하고자 지리적으로 분산된 각각의 PACS의 저장소를 데이터 그리드로 묶어 각 사이트끼리 서로 1차, 2차 백업데이터를 유지하는 것이다. 전통적인 PACS에서는 자체적으로 1차, 2차 백업서버를 만들고 관리하지만 만약에 해당 지역에 재난이 발생하였을 경우 모두 무용지물이 되고 만다. 이 시스템은 지리적으로 멀리 떨어져 있는 PACS 저장소끼리 서로 백업데이터를 복사하여 한 곳에 재난이 발생하여 문제가 생긴다 하더라도 다른 곳에 저장된 백업데이터를 이용하여 손실 데이터 복구가 가능하게하며 또한 영상조회도 가능하다[9]. 위의 연구들은 대용량 데이터를 처리하기 위해 데이터 그리드를 사용하여 저장 공간의 제약을 극복한 시스템들이다. 이들 시스템들이 활동적 저장장치(Active storage)[2]등의 계산 그리드의 일부를 사용하였으나, PACS에서의 실시간 영상압축 과정에서는 사용되지 않았다.

현재 사용되고 있는 PACS에서의 영상압축저장 시스템은 데이터베이스에 저장된 다이콤 파일 경로를 사용하여 저장장치의 다이콤 파일에 접근해서 단일처리기를 사용하여 압축을 수행한다. 압축을 수행하는 하드웨어는 현재 인텔 제온 마이크로프로세서를 장착한 서버급 컴퓨터가 사용되고 있다.

최근 의료영상 데이터의 발생량이 증가함에 따라서 PACS에서의 영상압축수행 시간도 점점 늘어나고 있다. 만약에 데이터 처리에 있어서 과도한 부하가 예상되는 경우는 압축을 담당하는 전용 서버를 추가하여 병렬처리 시스템을 구축 할 수도 있다. 그러나, 전용 압축 서버를 추가하기 위해 많은 비용이 소요된다. 따라서, 일부 대형 병원들을 제외하고는 비용의 절감을 위해서, 단일 처리기로 압축을 수행하고 있다. 더구나 압축을 위해서 별도의 하드웨어를 추가로 도입하는 것은 대부분의 중소규모 병원에 있어서는 공간 및 예산의 문제로 현실적이지 못한 상황이다.

따라서, 국내 의료산업계에서는 하드웨어 비용을 절감하며, 영상압축 수행 속도를 향상시킬 수 있는 기술이 요구된다. 본 논문에서는, 계산 그리드 기술을 사용하여, 사무용으로 사용 중인 개인용 컴퓨터들을 하드웨어로 활용하면서 저렴한 비용으로 의료영상을 압축하여 저장하는 시스템을 제시한다. 제안된 시스템은 추가적으로 활용 가능한 처리기들을 동적으로 관리하며, 부하에 대응하여 슬라이스 영상 단위로 작업을 분배한다. 의료영상 압축은 데이터들 간에 의존성이 없는 독립적인 작업이므로, 계산 그리드를 활용하여 병렬 처리의 효율을 충분히 얻을 수 있다.

본 연구에서는 PACS의 부분시스템으로서 계산그리드 기술을 이용하여 영상압축속도를 개선하기 위한 의료영상압축저장시스템의 시작품을 설계 구현하고, 결과물의 성능을 실측하여 제시한다.

3. 계산 그리드 컴퓨팅 기술을 적용한 의료영상압축저장 시스템의 설계와 구현

3.1 시스템 구조

PACS는 다이콤이란 의료영상표준을 기반으로 삼고 있다. 본 연구에서는 다이콤을 인터페이스로 하는 영상압축저장 부분시스템을 구현하였다.

그림 1에 도시한 바와 같이, 이러한 의료 영상 압축저장 시스템은 다이콤관문(DcmGate), 그리드관리자(Grid Manager), 계산노드(Computing Node), 저장장치의 4가지 요소들로 구성된다.

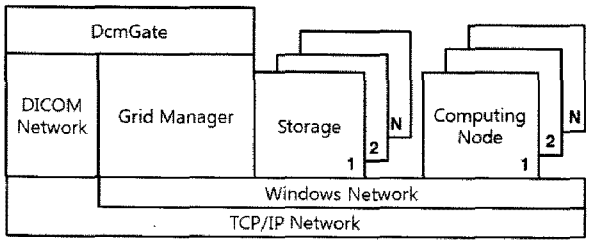

그림 1 의료영상압축저장 시스템 구조

다이콤관문은 다이콤 통신 인터페이스을 통해서 외부로부터 다이콤 파일을 받아들이고, 윈도우즈 네트워크(Windows Network)로 연결된 저장장치에 저장한다. 그리드관리자와는 윈도우즈 프로세스간 통신을 사용해 메시지를 전달하며, 그리드관리자에게 다이콤 파일이 저장된 결과를 알려준다.

그리드관리자는 다이콤관문으로부터 획득한 단면 단위로 저장된 다이콤 파일들의 압축 업무를 스케쥴링하며, 계산 자원들을 등록받아 관리한다. 그리드관리자는 계산노드와 TCP/IP 통신을 통해 메시지를 주고받는다. 또한 다중 계산노드와 통신을 하기위해 멀티쓰레드 방식의 네트워크 서버로 구현된다.

계산노드는 그리드관리자로부터 다이콤 파일경로를 전달받아 네트워크 드라이브(Network Drive)로 연결된 저장장치로 직접 접근하여 미리 정의된 전달 문법(Transfer Syntax)[1]에 맞게 영상압축을 수행한다.

저장장치는 윈도우즈 네트워크상에서 공유 저장소로 활용할 수 있는 시스템으로 윈도우즈 폴더를 공유하여 만들거나 또는 NAS(Network attached Storage)처럼 I/O전용프로세서를 탑재한 시스템을 이용한다.

3.2 시스템 흐름

그림 2에서 도시한 바와 같이, 다이콤관문은 다이콤 통신(DICOM Communication)[1]을 통해 의료영상데이터를 전달받을 수 있는 다이콤 저장 서비스 클래스 프로바이더(Dicom Storage Service Class Provider)[1]역할을 수행한다. 따라서 촬영장비로부터 다중 슬라이스 영상들로 구성된 대용량 영상을 획득하여, 다이콤 표준에 따라 전송받아 저장장치에 저장한다. 저장된 영상은 그림 3에서 보는 바와 같이 단면 영상들로 구성되며, 여러 장이 모여 하나의 볼륨(Volume)을 구성한다. 촬영장비가 환자를 일정한 간격으로 촬영한 이러한 MRI(Magmatic Resonance Imagining) 혹은 CT(Computed Tomography) 단면 영상들은 개별적으로 촬영되어 데이터 의존성이 없다.

그리드관리자의 작업 스케쥴러(Job Scheduler)는 다이콤관문을 통해서 실시간으로 전달받은 영상을 기본 영상 파일인 슬라이스 단위로 스케쥴링을 실시한다. 그리고 그리드 관리자의 자원관리자(Resource Manager)는 사용가능한 계산 자원을 모니터하여 여력이 있는 계산자원들에게 슬라이스 영상을 처리할 수 있도록 할당한다.

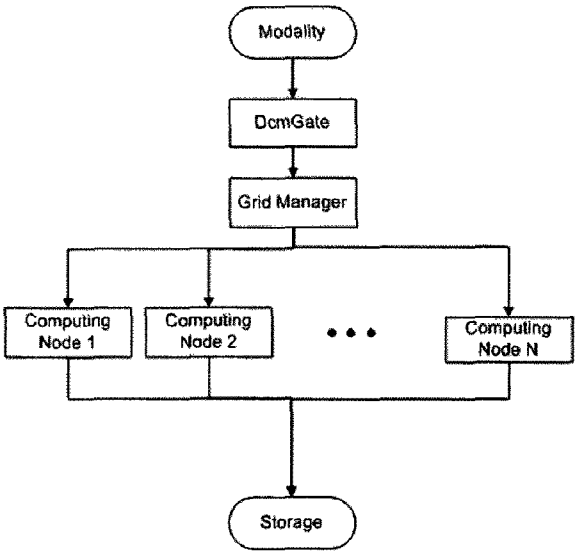

그림 2 제어 흐름도

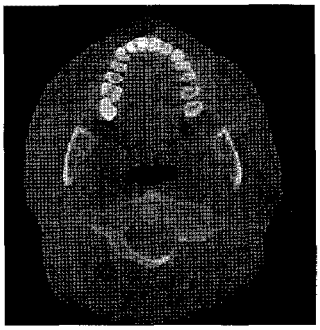

그림 3 슬라이스 영상의 예

계산노드는 실시간으로 자신의 상태를 자원관리자에게 업데이트하고 할당받은 슬라이스 영상 압축하여 다이콤에서 정의한 전달 문법(Transfer Syntax)[1]으로 변환한다. 이때 내부구성 모듈간 통신은 마이크로소프트 윈도우 네트워크 프로토콜과 TCP/IP 프로토콜을 사용하며, 다이콤관문과 촬영장비간 통신할 때만 의료영상표준인 다이콤을 따른다.

3.3 구성 요소들

3.3.1 다이콤관문

다이콤관문은 외부로부터 진단을 목적으로 하는 영상을 받기위해 의료영상표준인 다이콤 3.0 통신을 통해서 영상을 전달 받는다. C-STORE SCP(Service Class Provider)[1]는 다이콤관문에 구현되어 있으며, C-STORE SCU(Service Class User)는 촬영장비 에뮬레이터에 구현되어 있다.

이들 간의 메시지 순서도는 그림 4에 제시되어 있다. 그림 5는 다이콤 통신에서 전달되는 상세한 데이터 모델을 보여준다. 환자(Patient), 스터디(Study), 시리즈(Series), 영상(Image) 4가지 계층을 중심으로 개체관계를 보여주며[1], 구현한 다이콤관문 경우 시리이즈(Series)별로 영상을 저장장치에 저장한다.

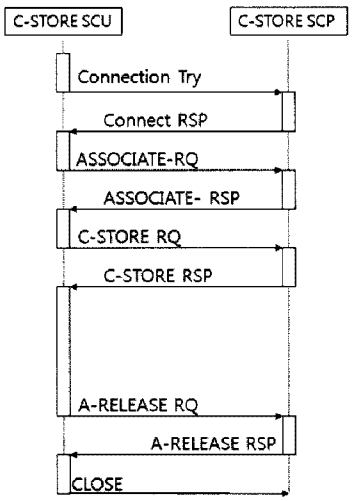

그림 4 DICOM Storage SCP와 SCU간 메시지 순서도[1]

다이콤관문은 표준에서 제시하는 대부분의 IOD(Information Object Definition)를 지원하고 다이콤 3.0을 완벽히 지원하도록 설계되었으며, 촬영장비(Modality)로부터 영상을 획득하기 시작하면, 콘솔을 통해 저장된 경로와 저장 날짜 그리고 시간을 표시한다.

3.3.2 그리드관리자

그리드관리자는 크게 작업 스케쥴러와 자원관리자로 구성되며, 작업 스케쥴러는 다이콤관문을 통해 전달받은 데이터를 슬라이스 단위로 작업큐(Job Queue)라는 자료구조에 담아놓는다. 작업큐는 FIFO(First In First Out) 구조를 갖는다. 자원관리자는 계산 자원으로 활용될 컴퓨터의 성능정보를 갖고 있으며 실시간으로 CPU 사용률도 점검한다. 또한 처리할 의료영상을 슬라이스 단위로 각 계산자원의 여력만큼 분배한다.

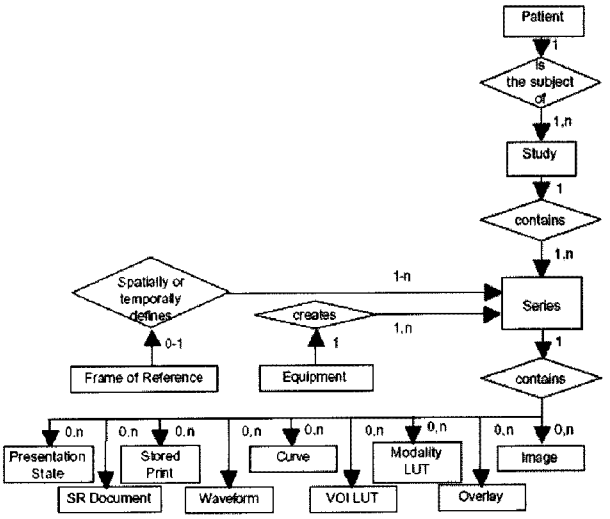

그림 5 DICOM 복합 IOD 정보 모델[1]

작업을 분할하여 계산자원의 처리능력과 상황에 따라서 분할 된 작업을 분배하기위해 우선순위 값을 사용한다. 우선순위 값은 동일한 시간 내에 컴퓨터가 압축의 주연산인 ADD 연산을 얼마나 빨리 수행할 수 있는지 계산하여 그 결과를 수치화한 값이다. 표 1에서 보는 것과 같이 계산자원 각각이 우선순위 값을 가지고 있다면, 이 값들을 우선순위 값의 합으로 정규화하여, 해당 계산자원의 우선순위 값에 비례하게 작업을 할당 받는다. 또한 CPU 사용률을 실시간으로 체크 하여 임계값을 설정한 다음 그 임계치가 넘지 않으면 작업을 할당받도록 설계한다.

계산 자원과 통신방법은 그림 6과 같은 형식의 패킷을 처리할 수 있도록 구현한다. 명령(Command) 필드는 1 바이트 크기이며, 두 가지 동작을 명시한다. 하나는 다이콤관문으로부터 저장장치에 있는 다이콤 파일경로를 받아 작업큐에 쌓는다. 여기서, 파일경로는 키(Key) 필드에 넣어 보낸다. 또 하나는 계산노드로부터 컴퓨터 상태정보를 업데이트 받을 수 있는 동작이며 이때 키(Key) 필드에 CPU사용률 및 컴퓨터의 처리속도 정보를

표 1 우선순위 값에 따른 작업 할당표 예

계산 자원	우선순위값	작업할당비율
PC 1	400,000	40%
PC 2	300,000	30%
PC 3	200,000	20%
PC 4	100,000	10%

Command(1)	NULL(3)	Reserve(1)	Reserve(1)	File Name(64)	File Size(4)	Not Use(182)
Key 256						

데이터크기 : BYTE

그림 6 그리드관리자에서 사용하는 TCP 패킷 구조

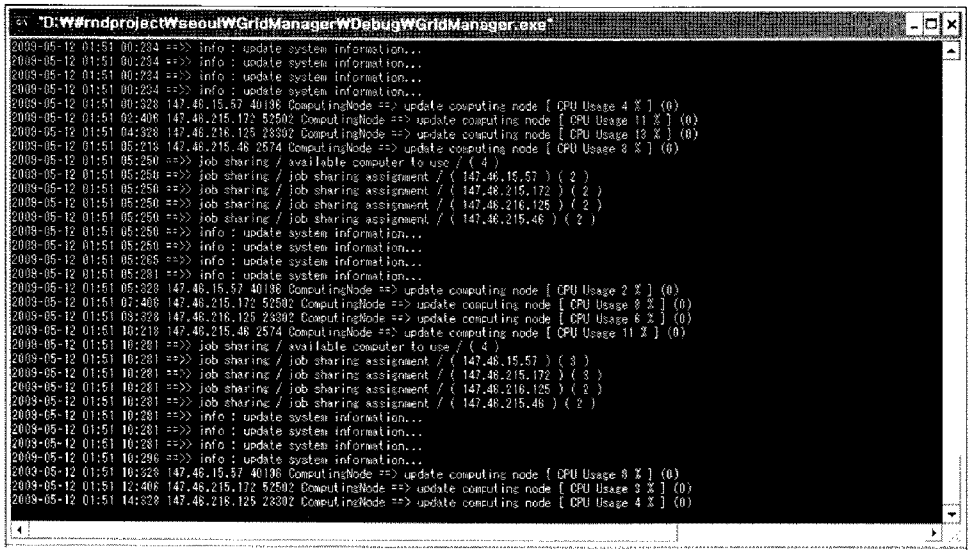

그림 7 그리드 관리자 구현화면

담아 보낸다.

그림 7은 다이콤관문과 계산노드 4대가 연동되어 동작하는 화면이다. 이 화면에서 CPU 점유율과 각 노드에 다이콤 파일 개수를 확인할 수 있다.

3.3.3 계산노드

계산노드는 기존에 다른 용도로 사용하고 있는 컴퓨터의 유휴자원을 제어하기 위해 해당 컴퓨터에 설치되며, 컴퓨터의 성능에 미치는 영향을 최소화 시킬 수 있도록 작고 가벼운 32비트 윈도우즈 콘솔(win32 console) 형태의 실행파일로 설계한다. 지원하는 압축방식은 다이콤 JPEG방식이며 Kuratorium OFFIS에서 개발한 DCMTK 라이브러리[11]를 이용하여 압축을 수행한다. 그리드관리자로부터 작업을 많이 할당받기 위해서는 타 계산노드보다 우선순위가 높고 CPU의 사용률 또한 사용자가 설정한 기준 이하이어야 한다.

다이콤관문으로부터 영상이 획득되기 시작하면 계산노드역시 실시간으로 동작하기 시작하며 그림 8에서 보는 바와 같이 2개의 실행큐(Execute Queue)를 통해서 각각의 작업쓰레드(Worker Thread)로 슬라이스 단위 영상들이 분배되어 압축작업이 이뤄진다. 이때 그리드관리자로부터는 슬라이스 단위로 물리적 파일경로를 전달받아 실행 큐에 저장한다. 계산 노드의 작업쓰레드는 실행 큐의 경로명을 사용하여, 공유된 윈도우즈 네트워크 저장장치(Windows Network Drive)에 직접접근하여 압축을 수행한다. 각 실행큐로 슬라이스 단위 영상들이

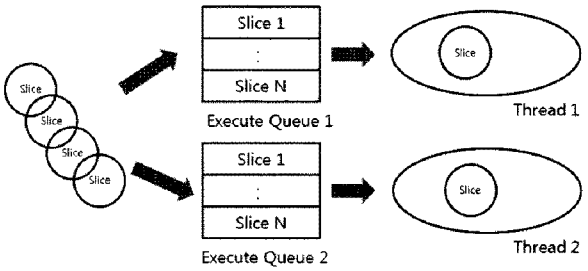

그림 8 계산노드 작업처리 구조

분배되는 기준은 라운드로빈 방식으로 이루어진다. 즉, 서로 번갈아가며 할당이 이뤄진다.

그림 9에서 보는 바와 같이 그리드관리자로부터 처리될 작업을 할당받고 영상압축을 완료하면 해당 경로와 처리시간이 화면에 표시된다. 주기적으로 CPU 사용률을 체크하여 그리드관리자에게 전달한 결과도 화면에 표시한다.

3.4 구현방법 논의

각 모듈의 유지보수를 위해, 각 모듈의 각각의 쓰레드들은 처리되는 상황을 모듈별로 하나의 로그에 기록한다. 각 쓰레드는 그림 10에서 도시하는 바와 같이 크리티컬섹션 함수를 사용하여 동시에 하나의 파일에 접근하게 될 경우의 데이터 동기화 문제를 해결한다.

계산노드의 소프트웨어 모듈은 소켓을 통해 전달받은 작업을 실행 큐에 넣는 쓰레드와 이들 큐에 접근하여 파일경로를 획득하여 압축을 수행하는 작업쓰레드들로

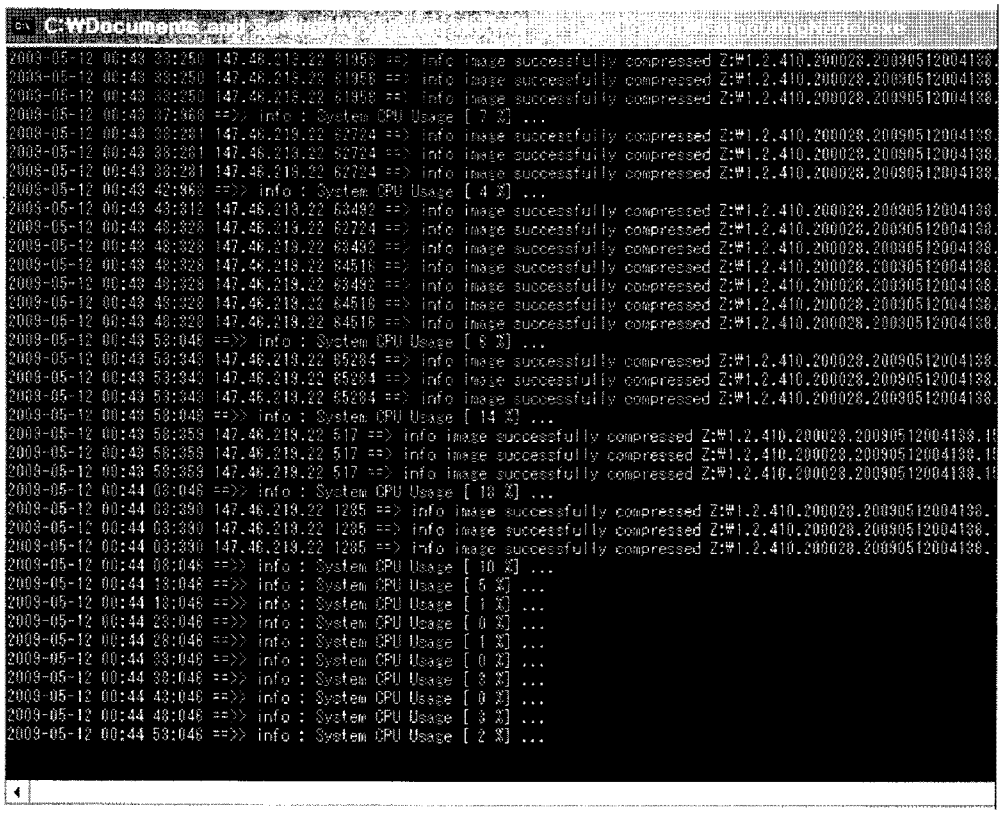

그림 9 계산노드 구현 화면

```
void WriteLog(char *text)
{
    EnterCriticalSection(&CriticalSection);
        :
    write log
        :
    LeaveCriticalSection(&CriticalSection);
return;
}
```

그림 10 쓰레드의 로그파일 동기화

```
DWORD WINAPI AcceptHandler(void* Socket)
{
                :
  while(bLoop){
    nBytesRecv = recv(RemoteSocket, szBuffer, 512, 0 );
    char szLogBuffer[1024] = { 0 };
                :
    ServerCommand(pThreadData, sCommand, sData,
sSaveType, sReturnType, &TransfileInfo);
                :
  }
  return 0;
}
```

그림 11 그리드 관리자의 주요 코드

구성되어 있다. 이러한 설계는 통신 쓰레드와 작업쓰레드를 분리하여, 통신의 동기화로 인한 지연이 작업 쓰레드로 전이되지 않도록 한다. 따라서 작업 쓰레드들이 모듈 외부와의 통신에 따른 동기화에 영향을 받지 않고, 실행 큐에 단면 영상이 존재하는 한 압축을 수행하도록, 다시 말해 최대한 병렬적으로 수행되게 설계되었다.

그리드 관리자는 다수의 계산노드와 다이콤관문과 동시에 통신을 하며 동작하는 다중 쓰레드로 구성된다. 그림 11에서 도시하는 바와 같이 ServerCommand라는 다중 쓰레드를 생성하는 함수를 만들어 각 메시지에 대응하도록 설계하였다.

4. 평가 및 논의

4.1 시스템 사양

하드웨어 시스템의 구성은 영상을 획득하고 분할하는 개인용 컴퓨터 한 대와 계산 자원으로 활용될 다수의 개인용 컴퓨터들로 구성하며, 운영체제로는 윈도우즈 운영체제가 사용되었다. 제안된 시스템 구성요소들은 윈도우즈 콘솔과 마이크로소프트 파운데이션 클래스(MFC) 환경에서 C++언어를 이용하여 구현하였으며, 다이콤파일[1] 핸들링과 압축 모듈은 기존에 공개된 라이브러리를 이용하여 구현한다.

다이콤관문과 그리드관리자 그리고 계산노드가 설치된 듀얼코어를 가진 개인용 컴퓨터의 사양은 표 2와 같으며, 한 대 이상의 계산노드로 구성할 수 있다. 저장장치는 PC에 장착된 하드디스크를 공유하여 윈도우즈 네트워크 저장장치를 사용하였다.

표 2 그리드 관리자, 계산 노드, 다이콤관문 사양

CPU	RAM	HDD	Network	OS
Intel Core 2 Duo E8200 2.66GHz	2GB	250GB	Broadcom NetXtreme Gigabit Ethernet Card 100Mbps	Microsoft Windows XP

4.2 측정방법

본 논문에서 구현한 시스템은 기술적 특성상 의료영상을 슬라이스 단위로 작업을 분할하기 때문에 다중슬라이스로 구성된 의료영상을 이용한다. 측정에 사용된 샘플데이터는 그림 3과 같은 환자의 구강구조를 3차원 의료영상으로 재구성하기 위해 촬영된 CT(Computed Tomography)영상이다. 총 255장의 이미지로 구성되며 약60MB(Mega Byte)의 볼륨을 갖는다. 즉 하나의 촬영장비에서 생성하는 데이터는 일괄적으로 1명의 환자에 대한 60MB용량의 검사결과 영상을 생산하는 것으로 가정한다.

성능측정 방법은 촬영장비로부터 의료영상을 전달받기 시작하여 압축이 완료되는 시간을 측정한다.

계산 그리드 적용의 효과를 입증 받기위해서는 계산노드의 수의 증가에 따른 영상압축속도를 측정한다. 우리는 1대의 계산노드를 구성 했을 때와 2대의 계산노드를 구성 했을 경우 등, 그 수를 점차 늘려가며 최종 4대의 계산노드까지 구성하여 측정하였다. 또한 촬영장비로부터 전달 받는 데이터크기 변화 따른 측정을 실시한다. 1대의 촬영장비는 60MB 볼륨의 영상을 생산하여 전달해주며, 점차 촬영장비의 수를 늘려가며 최종 4대까지 영상데이터를 전송하여 그 측정된 데이터를 분석한다.

4.3 평가 및 논의

그림 12와 그림 13 그리고 그림 14와 그림 15는 4.2절에서 언급한 계산노드의 수와 촬영장비로부터 전달받는 영상의 크기에 따른 총 16가지 다른 구성에 따른 측정결과를 보여준다.

그림 12에서 보인 바와 같이, 1대의 촬영장비로부터 약 60MB의 영상을 전송받을 때는 계산노드 1대로 구성했을 때 25초의 처리시간이 소요되었으며, 2대로 구성했을 경우에는 1대로 구성했을 때보다 20%의 처리속도가 개선되었다. 하지만 그 이상의 계산노드를 구성하였을 때 처리속도 개선은 없었다.

그림 13에서 보인 바와 같이, 2대의 촬영장비로부터 총 120MB의 영상을 전송받을 때는 계산노드의 수를 2대로 구성했을 때 1대로 구성했을 때보다 22%의 속도가 개선되었고, 2대 구성 대비 3대를 사용하는 경우에는 7%의 처리속도가 개선되었다. 계산노드 4대로 구성했을 경우에는 3대로 구성했을 때보다 4%의 처리속도가 개선되었다. 계산노드의 수가 늘어남에 따라 3대 이상의 구성으로는 미미하게 압축에 소요되는 시간이 단축되었다.

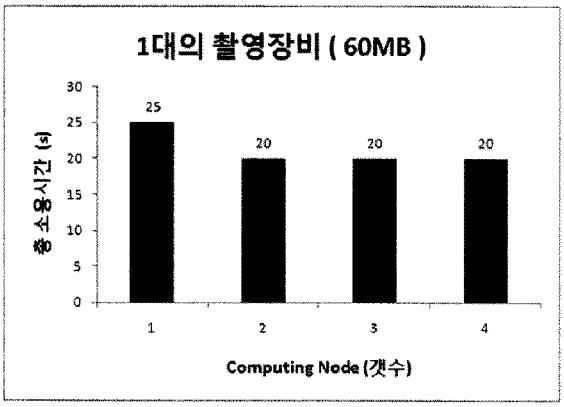

그림 12 1대의 촬영장비로부터 영상을 전달받을 경우의 다양한 계산 노드 수에 따른 처리시간

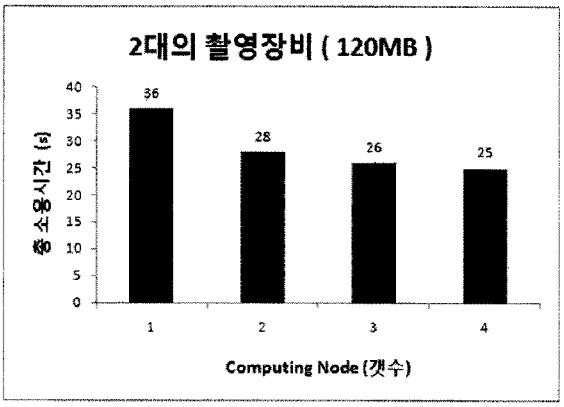

그림 13 2대의 촬영장비로부터 영상을 전달받을 경우의 다양한 계산 노드 수에 따른 처리시간

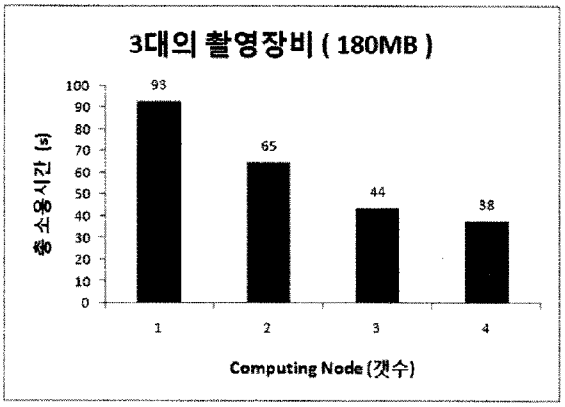

그림 14 3대의 촬영장비로부터 영상을 전달받을 경우의 다양한 계산 노드 수에 따른 처리시간

그림 14에서 보인 바와 같이, 3대의 촬영장비로부터 총 180MB의 영상을 전송받을 때는 계산노드의 수가 1대에서 3대까지 수를 증가할 경우에 각 단계별로 30%, 32% 처리속도가 개선되었다. 하지만, 4대의 계산노드를 사용

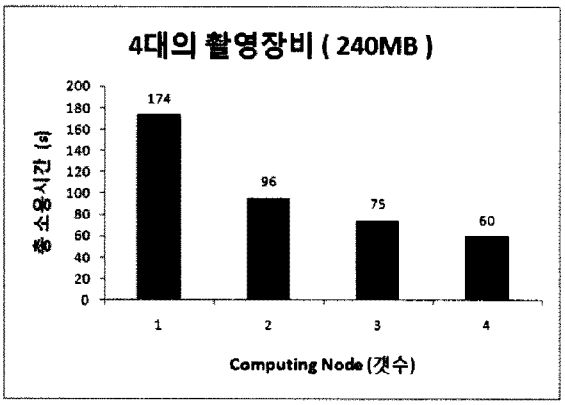

그림 15 4대의 촬영장비로부터 영상을 전달받을 경우의 다양한 계산 노드 수에 따른 처리시간

하였을 경우에는 3대 대비 14%의 시간단축에 그쳤다.

그림 15에서 보인 바와 같이, 4대의 촬영장비로부터 총 240MB의 영상을 전송받을 경우에 계산노드의 수가 한 대 씩 증가할 때마다 각 단계별로 45%, 22%, 20%씩 처리속도가 개선되었으며, 계산노드의 수가 증가할 때마다 각 단계별로 최소 20% 이상 처리속도가 개선되었다.

1대의 촬영 장비로 발생하는 데이터 60MByte는 총 256개의 단면 영상으로 구성된다. 이 영상들은 네트워크를 통해서 전달되며 디스크 I/O를 통해서 저장장치에 저장되며, 디렉토리 시스템에 파일명을 저장해야 된다.

이러한 파일 저장은 다이콤관문이 받아서 네트워크 저장장치로 저장할 때 한 번, 그리고, 계산 노드가 압축해서 저장할 때 한 번하여 총 2번 일어난다. 실제 실험 환경에서 512개(즉, 256 × 2개)의 단면 영상들을 복사하여 저장한 실험에서 약 20초가 소요되었다. CPU 처리속도보다 더 느린 이러한 I/O 장치의 지연으로 인해 총 소요시간이 20초 이하로 줄일 수 없다고 생각된다.

하지만, 촬영 장비의 증가로 압축저장해야 할 데이터의 양이 늘어나면, 압축저장에 소요되는 총 소요시간이 중에 CPU가 소요하는 시간의 비중이 늘어나서, 계산노드의 개수를 늘림에 따라서 총 소요시간을 감소하는 효과가 증대된다.

그림 16은 다양한 데이터 크기를 처리할 경우에, 처리기 수 변화에 대한 처리에 필요한 총 소요시간을 보여주고 있다. 이는 입력된 데이터의 양이 많은 경우에 다중 계산노드들이 충분히 활용되어 처리속도를 더욱 개선할 수 있음을 보여준다.

따라서, 그림 16에서 1개의 계산노드를 구성하여 180 MByte의 영상을 압축하는 경우와 같이, 만약 현재 병원에서 사용되고 있는 단일 처리기 기반의 PACS의 처리 소요시간이 처리해야 하는 데이터의 양이 많아짐에 따라 지수적으로 증대된다면, 제안된 압축저장 시스템을 사용하여 계산노드들의 수를 늘려 총 소요시간을 획기

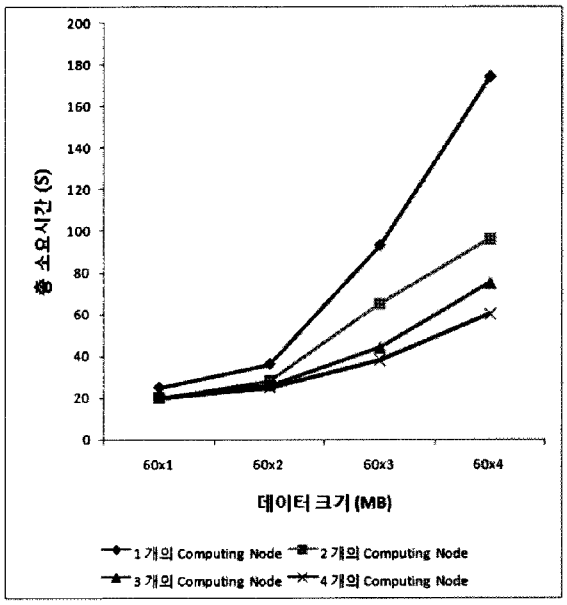

그림 16 데이터크기 변화에 따른 계산노드의 수가 처리시간에 미치는 영향

적으로 줄일 수 있음을 알 수 있다. 이는 계산노드의 소프트웨어 모듈이 다중 쓰레드들을 사용하여 다중 데이터에 대해 병렬화됨으로써, 처리될 데이터의 양이 늘어나면서 그 병렬화 효과를 충분히 발휘한 결과로 볼 수 있다.

5. 결 론

대형병원에서 환자의 치료와 진단을 목적으로 하루에 생산되는 의료영상의 발생양은 보다 정확하고 정밀한 진단의 요구하는 촬영장비와 네트워크 인프라의 발달로 나날이 증가하고 있으며 앞으로 그런 추세는 계속 될 것이다. 그러나 별도의 하드웨어를 추가로 도입하는 것은 대부분의 중소규모 병원에 있어서는 공간 및 예산의 문제로 현실적이지 못한 상황이다. 따라서 기존방식의 PACS보다 처리속도가 개선되며 추가적인 하드웨어 비용이 절감된 시스템이 요구된다. 본 연구는 계산그리드 기술을 이용하여 영상의 획득과 처리단계에서 많은 데이터 발생량으로 인해 발생할 수 있는 시스템 병목현상을 개선하며 병원운영의 비용측면에서 효과적인 의료영상압축저장 시스템을 제안하였다.

본 논문에서 구현한 시스템으로 실험한 결과 4대의 촬영장비로부터 총 240MB의 영상을 전송받아 처리속도를 측정한 결과 계산노드의 수를 한 대로 구성하여 측정 후 노드를 한 대씩 증가시키며 소요시간을 실제 측정한 결과, 처리노드가 한 대씩 증가할 때마다 각각 45%, 22%, 20%씩 처리속도가 개선됨을 알 수 있었다. 따라서 계산자원을 다중으로 구성한 효과를 검증하였으며, 촬영장비들로부터 발생하는 의료영상 데이터가 늘어

날수록 단일 처리노드로 처리할 때보다 처리속도가 크게 개선됨을 확인하였다. 본 논문에서 구현한 시스템은 일반PC를 계산자원으로 활용할 수 있으므로 대형병원의 다양한 용도로 사용하는 수 많은 데스크톱컴퓨터를 활용할 수 있어 비용 측면에서 효과적이며, 추가적인 비용을 발생시키지 않고 병원에서 운영하는 PACS의 성능을 개선하고, 일시적인 부하 증대에도 대비할 수 있는 장점을 가지고 있다. 향후 연구로는 실시간 네트워크 트래픽이 발생하는 대형병원 및 메디컬 센터에서 성능측정이 필요하다.

참 고 문 헌

[1] National Electrical Manufacturer Association (NEMA), "Digital Imaging and Communication in Medicine (DICOM)," PS 3 2007. 2007: Parts 1-18, Virginia, U.S.A. (Available: ftp.nema.org/medical/ dicom/2007/)

[2] S. Hastings, S. Oster, S. Langella, T.M. Kurc, T. Pan, U.V. Catalyurek, and Saltz J.H. "A Grid-based image archival and analysis system," *Journal of the American Medical Informatics Association*, vol.12, pp.286-295, January 2005.

[3] K. Inamura, S. Kousaka, Y. Yamamoto, Y. Sukenobu, Y. Okura and Y. Matsumura et al., "PACS development in Asia," *Comp Med Imaging Graph*, vol.27, pp.121-128, 2003.

[4] V. Berstis, *Fundamentals of Grid Computing*, Reading Mass, IBM Redbook, October 2002.

[5] J. Montagnat, F. Bellet, H. Benoit-Cattin, V. Breton, L. Brunie, H. Duque,Y. Legre, I. Magnin, L. Maigne, S. Miguet, J.-M. Pierson, L. Seitz, and T. Tweed, "Medical images simulation, storage, and processing on the european datagrid testbed," *Journal of Grid Computing*, vol.2, no.4, pp.387-400, December 2004.

[6] H. K. Huang, Aifeng Zhang, Brent Liu, Zheng Zhou, Jorge Documet, Nelson King, L. W. C. Chan, "Data grid for large-scale medical image archive and analysis," *Proceedings of the 13th annual ACM international conference on Multimedia*, Singapore, November 06-11, 2005.

[7] Florida Estrella, Richard Mcclatchey, and Dmitry Rogulin, "The Mammogrid Virtual Organization - Federating Distributed Mammograms," *Studies in Health Technology and Informatics*, vol.116, pp.935-940, 2005.

[8] I. Foster and C. Kesselman and S. Tuecke, "The Anatomy of the Grid," *The International Journal of High Performance Computing Applications*, vol.15, no.3, pp.200-222, 2001.

[9] B.J. Liu, M.Z. Zhou and J.E. Documet, "Utilizing data grid architecture for the backup and recovery of clinical image data," *Computerized Medical Imaging and Graphics*, pp.95-102, 2005.

[10] Wolfgang Leister et al, "Implications of Introducing Grid in Medical Applications," *Proceedings of the 24th International EuroPACS Conference*, pp.15-17, Trondheim, Norway, 2006.

[11] Kuratorium OFFIS, "DCMTK: DICOM-Toolkit, Version 3.5.4," 2005. (Available: http://dicom.offis.de / dcmtk)

안 병 규

2004년 2월 건양대학교 정보관리학과 졸업. 2009년 8월 중앙대학교 정보대학원 컴퓨터소프트웨어학과 석사학위 취득예정. 2004년 9월~2005년 2월 비트교육센터Software Engineer 263기 전문가과정 수료. 2005년 5월~2008년 3월 인피니트테크놀로지 연구개발부 PACS/CDIS 개발담당. 2008년 3월~현재 이우테크놀로지 소프트웨어연구소 3D영상연구부 Dental 3D Viewer 및 DICOM 개발담당. 관심분야는 PACS, CIS/CDIS, DICOM, HL7, ECG XML, Grid Computing

박 재 현

1988년 2월 중앙대학교 전자계산학과 졸업. 1991년 2월 한국과학기술원 전산학과 석사. 1995년 8월 한국과학기술원 전산학과 박사. 1995년 8월~2000년 2월 삼성전자 정보통신본부 데이터네트워크개발팀 MPLS/ATM 개발담당. 2000년 3월~2002년 8월영남대학교 전자정보공학부 정보통신전공 교수. 2002년 9월~현재 중앙대학교 컴퓨터공학부 교수, 중앙대학교 정보대학원 컴퓨터소프트웨어학과 교수. 관심분야는 ATM Switch Arch., Multiprotocol Label Switching System, Routing Protocols, Ad Hoc Networking, Peer-to-Peer Networking, Grid Computing