

KMTNet 자료처리 시스템 설계와 측광데이터베이스 구축
DESIGN OF DATA REDUCTION SYSTEM AND CONSTRUCTION
OF PHOTOMETRIC DATABASE FOR KMTNet

김동진, 이충욱, 김승리, 박병곤, 이재우

한국천문연구원

D.-J. KIM, C.-U. LEE, S.-L. KIM, B.-G. PARK, AND J. W. LEE

Korea Astronomy and Space Science Institute, Daejeon 305-348, Korea

E-mail: keaton03@kasi.re.kr

(Received October 27, 2009; Accepted November 23, 2009)

ABSTRACT

We have designed data processing server system to include data archiving, photometric processing and light curve analysis for KMTNet (Korea Microlensing Telescope Network). Outputs of each process are reported to the main photometric database, which manages the whole processing steps and archives the photometric results. The database is developed using ORACLE 11g Release 2 engine. It allows to select objects applying any set of criteria such as RA/DEC coordinate and Star ID, etc. We tested the performance of the database using the OGLE photometric data. The searching time for querying 70,000,000 records was under 1 second. The database is fully accessed using query forms via web page.

key words : 관측기기, 자료처리, KMTNet, 데이터베이스

1. 서론

1991년 처녀자리에 위치한 중성자별 PSR 1257에서 펄사의 타이밍 관측을 통해 최초의 외계행성이 발견 (Wolszczan and Frail 1992)된 이후 다양한 관측방법으로 발견된 외계행성의 수는 약 400여개에 이른다¹⁾. 발견된 외계행성의 대부분은 고분산 분광관측을 이용한 시선속도 변화를 측정하여 이루어졌으며, 미시중력렌즈 (micro gravitational lensing) 현상을 이용한 방법으로 9개가 발견되었다. 발견된 외계행성의 개수만을 비교하면 중력렌즈 현상을 이용한 외계행성 발견율은 시선속도방법에 비하여 그리 높지 않은 편이나, 중력렌즈방법은 시선속도방법과는 달리 외계행성이 모성으로부터 떨어진 거리 및 질량에 거의 무관하기 때문에 관측제약에 치우침 없이 완전한 자료를 제공할 수 있다. 따라서 중력렌즈방법을 이용한 외계행성 탐색연구는 단순한 외계행성의 발견을 넘어 외계행성계의 생성과 진화까지 그 연구의 폭을 넓힐 수 있는 장점을 가진다.

현재 중력렌즈 탐색은 OGLE (Optical Gravitational Lensing Experiment, Udalski et al. 1992)과 MOA (Microlensing Observations in Astrophysics, Bond et al. 2001) 그룹에 의해서 주로 이루어지고 있다. 두 연구구

룹은 은하중심 방향의 탐색 관측을 통해 매년 약 1000 개의 미시중력렌즈 현상을 발견하고, 매일 갱신된 각각의 광도곡선을 홈페이지를 통해 전 세계 중력렌즈 현상 연구자에게 대용량 자료를 제공한다. 특히 OGLE-II의 자료는 1996년부터 2000년까지 관측된 약 4천만 개의 별들에 대한 10억개의 측광결과를 웹을 통한 데이터베이스 검색으로 사용자에게 제공하고 있는데, 이때 MySQL 데이터베이스 엔진이 사용된다. 중력렌즈현상을 이용한 외계행성 탐색 연구자들은 이들이 제공하는 중력렌즈에 의해 변형된 광도곡선의 형태를 분석한 후, 그 변형이 외계행성에 기인한 효과인지의 여부를 판단한다. 외계행성에 의한 변형이 의심되는 후보 사건에 대해서는 국제 관측망을 이용하여 집중적인 후속관측을 수행한다. 중력렌즈 현상에 의한 광증폭 현상은 수 일에서 수십 일에 이른다. 이 중에서 외계행성을 포함한 중력렌즈 현상의 광도곡선 변형은 수 시간에서 수 일 정도에 걸쳐 주로 고증폭 현상에서 나타나기 때문에, OGLE과 MOA에서 제공하는 초기 광도곡선 정보는 추적관측의 필요성 여부를 판단할 수 있는 좋은 기준이 된다. 한국천문연구원은 2000년부터 중력렌즈방법을 이용한 외계행성 발견을 위해 관측, 자료처리 및 모델 분석을 통한 물리량 결정 등의 연구를 수행해 오고 있다.

1) <http://exoplanet.eu>

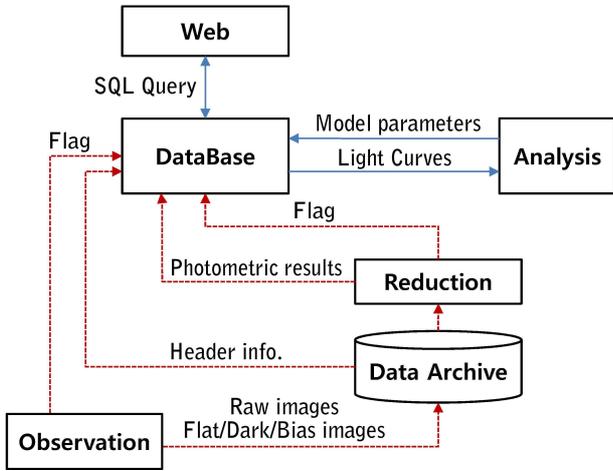


그림 1. KMTNet 자료처리 시스템 및 자료 흐름도

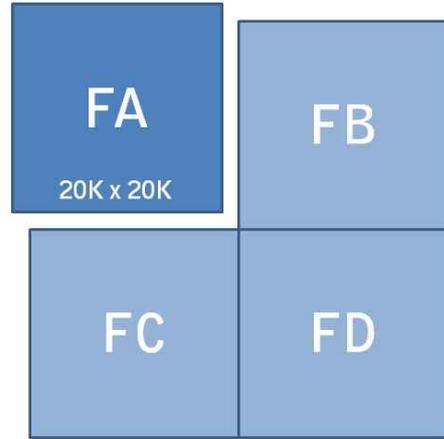


그림 2. KMTNet 관측영역

특히 2003년부터 국제공동관측망인 MicroFUN (Microlensing Follow-Up Network) 그룹에 참여하여 목성형 외계행성 (Udalski et al. 2005)과 해왕성급 외계행성 (Gould et al. 2006), 그리고 우리 태양계를 닮은 외계행성계를 발견 (Gaudi et al. 2008) 하였다.

앞서 기술한 바와 같이 외계행성을 포함한 중력렌즈에 의한 광도곡선의 변형은 수 시간에서 수 일 사이에 일어난다. 특히 지구와 같이 질량이 작은 행성의 경우에는 광도곡선의 변형 지속시간이 수 시간 정도로 짧기 때문에 현재의 관측전략, 즉 OGLE이나 MOA에 의한 중력렌즈 현상의 발견과 MicroFUN 등 후속 관측그룹의 추적관측으로 구성된 2단계 관측전략에서는 체계적이고도 효율적인 발견이 어려운 실정이다. 이에 한국천문연구원에서는 KMTNet (Korea Microlensing Telescope Network) 시스템을 구축하여 중력렌즈방법을 이용한 외계행성 발견의 체계적인 연구를 수행할 계획이다. 이 시스템은 은하중심 방향 $4^\circ \times 4^\circ$ 영역의 특정지역을 24시간 연속 관측하여 중력렌즈 현상의 발견과 추적관측이 동시에 이루어지는 시스템으로 기존의 2단계 관측을 하나로 합친 효과를 낸다. 남반구 천문대에 설치한 광시야 관측 장비로 얻은 자료는 파이프라인을 통하여 일괄적으로 처리할 계획이다. 자료처리를 마친 각각의 측광결과와 데이터센터의 측광데이터베이스를 통하여 효율적으로 관리된다.

이 연구에서는 관측자료의 수집부터 전처리, 측광과정을 거쳐 모델 분석에 이르는 KMTNet 자료처리 시스템을 구성하는 각 서버의 하드웨어 설계와 각 자료처리 단계 및 측광결과를 관리하는 측광데이터베이스의 구축, 그리고 시험결과에 대해 기술한다. 특히 2장에서는 자료처리 시스템을 구성하는 Observation, Reduction,

Data Archive, Analysis 서버에 대하여 기술하고, 3장에서는 DataBase와 Web에 대하여 기술한다 (그림 1).

2. 자료처리 시스템 설계

이 장에서는 3개 관측지에서 수집한 관측자료를 체계적으로 분리하고 전처리와 측광과정을 수행하는 각각의 자료처리 시스템 설계에 대하여 기술한다. 처리를 마친 측광결과는 측광데이터베이스에 저장되어 웹문서를 통해 사용자에게 제공된다. 그림 1에 자료처리 시스템에서 각 자료의 흐름과 관계를 제시하였다.

2.1 관측자료의 수집 (Observation)

KMTNet의 $20K \times 20K$ CCD 카메라를 탑재한 광시야 관측시스템이 한 번에 관측할 수 있는 최대시야는 $2^\circ \times 2^\circ$ 이므로 $4^\circ \times 4^\circ$ 의 영역을 관측하기 위해서 미리 정의된 4 곳으로 망원경을 순차적으로 이동시키며 관측한다(그림 2). 특히 이 관측시스템은 지구와 같은 질량이 작은 외계행성의 검출을 목표로 하고 있기 때문에 짧은 변형지속시간동안 관측자료를 획득하기 위하여 적어도 10분에 한번 씩 관측이 이루어 져야 한다. 따라서 각각의 관측영상은 120초의 노출시간을 가지며, 30초 동안 CCD의 읽기와 저장이 이루어진다.

$20K \times 20K$ CCD 카메라로 촬영한 $2^\circ \times 2^\circ$ 영역의 관측영상은 픽셀 당 16bit의 정보를 가지며 관측영상 1장의 크기는 이미지 헤더를 포함하여 약 840MB정도가 된다. 각각의 관측소에서 2.5분 간격으로 1장씩 하루 10시간을 촬영한다고 가정하면 한 관측소에서 생산되는 영상은 약 240장이며, 전처리에 필요한 Flat, Bias, Dark 영상들을 포함할 경우 하루 동안 생산된 자료가 차지하는 공간은 약 210GB가 된다. 여기에 측광과정에서 생성되

표 1. LTO-4 tape specifications (from <http://www.imation.com>)

General Specifications		Lifetime Tape Durability		Approximate years of life assuming	
Tape Type	Native Capacity	Total end-to-end passes	Entire tape reads/writes	one entire tape written per month	one entire tape writer per week
LTO-4	800GB	11200	200	17	4

는 중간처리 영상들을 포함하면 자료처리에 매년 360TB 이상의 저장 공간이 필요하다.

관측시스템을 통해 매일 생산되는 대량의 관측자료를 우리나라에 위치한 데이터센터로 전송하는 것은 현실적으로 어려움이 많다. 각 관측소에서 관측된 10시간 분량의 관측자료 (약 210GB)를 12시간 이내에 전송하기 위해서는 최소 50Mbps 정도의 네트워크 속도가 요구되는데 관측소와 데이터센터 사이에서 이 정도의 속도를 유지하기란 쉽지 않다. 특히 각 관측소와 우리나라 사이의 네트워크 연결 상황과 네트워크가 장시간 끊어져 자료 전송이 불가능한 상황 등을 고려하면 네트워크를 통한 실시간 자료전송은 거의 불가능하다. 따라서 우리는 관측자료의 전송을 위하여 2.5/3.5인치 하드디스크 또는 LTO (Linear Tape Open) 등의 저장매체에 관측자료를 담아 항공우편으로 전송하는 방법을 사용할 계획이다.

개인용 PC에서 널리 사용 중인 3.5인치 하드디스크는 현존하는 여러 저장매체 중 구입이 가장 쉽고 용량대비가 가격이 저렴하나 디스크, 스피들 모터, 헤더 등 고속으로 움직이는 정밀부품이 내장되어 있어 이동 중 외부 충격으로 손상을 받아 데이터가 소실될 가능성이 크다. 노트북용으로 개발된 2.5인치 하드디스크는 이동성을 고려하여 생산하였기 때문에 3.5인치 하드디스크에 비하면 안정성이 높지만 아직까지는 충분한 저장 공간을 확보하기 어렵다. 현재 시판되고 있는 3.5인치 하드디스크는 2TB까지 개발되었고 2.5인치 하드디스크는 640GB까지 개발되었다.

LTO는 자기 테이프의 한 종류로 가격대비 저장 공간은 하드디스크와 비슷하다. 접촉에 의한 마그네틱 필름의 열화 현상으로 테이프 자체의 수명이 존재하며 자력과 열에는 취약한 편이나, 정밀부품이 내장되어있지 않기 때문에 이동 중에 발생할 수 있는 충격에는 매우 강한 장점을 가지고 있어 자료의 이동이 필요한 곳에 적합하다. 현재 가장 널리 사용되고 있는 규격은 LTO-4이고 비 압축 시 800GB, 하드웨어 압축 시 1.6TB까지 저장 가능하다. 표 1에 LTO-4의 특징을 나타내었다.

LTO는 데이터를 읽고 쓰기 위한 별도의 전용 드라이브 장치가 필요하기 때문에 하드디스크와 달리 추가 비용이 소요되지만, 대형 테이프 라이브러리를 이용할 경우 자료가 들어있는 테이프에 바코드 또는 RFID(Radio-frequency identification)를 적용하여 관측자료를 체계적으로 관리할 수 있고, 데이터 아카이브 구축비용을 절감할 수 있는 장점도 있다.

2.2 자료처리 서버 (Reduction)

매일 관측되는 대용량의 영상을 전처리하고 축광과정을 거쳐 축광결과를 얻기 위해서는 고성능, 대용량의 컴퓨터가 필수적이다. 관측영상은 전처리 작업과 Alard and Lupton (1998)에 의하여 소개된 영상차감법²⁾ (Difference Image Analysis)을 이용하여 축광과정을 거치게 되는데, 이미지 전체를 메모리에 상주시키는 단일 처리방법 대신 이미지를 분할하여 처리하는 분산 처리방법을 이용한다.

일반적으로 20K×20K의 거대한 영상을 단일 CPU로 처리할 경우 CPU의 연산 시간도 길고 메모리 사용량 또한 증가하게 된다. 더욱이 메모리 저장 공간이 부족하면 서버는 하드디스크를 가상메모리로 이용하는데, 연산에 필요한 기본 입출력 외에도 가상메모리의 입출력이 더해지기 때문에 이로 인한 서버의 전반적인 연산 속도는 더욱 느려지게 된다. 이때 가상메모리의 가용공간마저 부족해지면 서버가 정지하거나 스스로 리부팅되기도 하는데 이러한 문제는 자동화된 데이터처리 파이프라인 구축 시 오작동, 처리 지연 또는 중단 등의 원인이 될 수 있다. 단일 처리방법시 서버의 부하를 줄이고 연산시간을 감소시키기 위하여 고성능의 CPU, 대용량의 RAM 그리고 빠른 하드디스크에 의존할 수밖에 없어 시스템의 구성에 많은 비용이 요구 된다. 우리는 이러한 요인을 감안하여 한 장의 거대한 영상을 6400개의 조각영상으로 나누어 처리하는 분산 처리방법을 도입하였다. 관측영상을 가상메모리를 사용하지 않는 법

2) <http://users.camk.edu.pl/pych/DIAPL/>

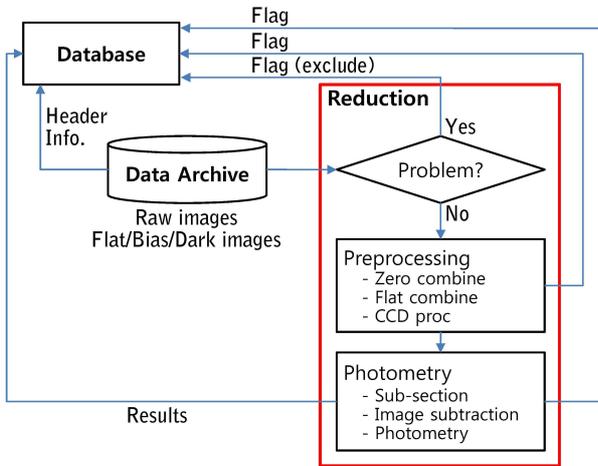


그림 3. 관측자료의 측광과정

위 내에서 분할하고, 분할된 영상을 여러 개의 서버를 구성하는 각각의 CPU에 네트워크로 분산하여 처리한다. 이때 각각의 분할된 작업을 각 CPU로 분산하는데 걸리는 시간이 추가되지만 단일 처리에 비해 계산시간을 크게 단축시킬 수 있으므로 심각하게 고려할 사항은 아니다. 그림 3에 관측자료의 처리 과정을 흐름도로 나타내었다.

우리는 구체적인 하드웨어 설계를 위하여 가상영상을 만들고 이를 측광파이프라인에 적용하여 자료처리 속도를 측정해 보았다. 실험에 사용한 자료처리 서버는 Intel Xeon 3GHz, 8GB RAM, 7200 RPM SATA 하드디스크로 구성된 시스템을 사용했다. 전처리가 완료된 20K×20K의 영상을 6400장으로 분할하여 영상차감법을 수행할 경우 각각의 영상처리에 0.8초가 소요되었고 6400장의 이미지 처리에 총 85분이 소요되었다. KMTNet은 2.5분마다 20K×20K의 영상을 획득하기 때문에 실시간 자료처리 시스템의 구성을 위해서는 실험에 사용한 동일한 성능의 서버가 총 34대 필요하게 된다. 이때 공유된 저장 공간을 사용한다면 파일 분배에 걸리는 시간이 소요되지 않지만, 수십개의 CPU가 동시에 입출력 명령을 내리기 때문에 디스크의 과부하로 입력 속도의 저하가 발생하고 이로 인해 실제 걸리는 시간은 조금 더 소요된다. 그러나 관측자료 처리 서버의 구축 시 일반 하드디스크 형태 대신 메모리 형태의 디스크인 SSD (Solid State Drive)를 RAID0 형태로 묶어서 사용할 경우 자료 분배 시간 및 디스크 과부하를 줄여 최적의 속도를 낼 수 있다.

2.3 데이터 아카이브 (Data Archive)

KMTNet의 관측자료 측광과정에서 발생하는 처리영상

및 측광결과를 저장하고 관리하기 위해서는 연간 360TB의 저장 공간이 필요하다. 이러한 대규모의 공간을 여러 대의 서버에 독립적으로 장착할 경우 많은 자원의 낭비와 함께 비용과 관리 측면에서 모두 비효율적이다. 따라서 우리는 모든 서버들이 공동으로 사용할 수 있는 통합 저장 공간을 구성하고 이를 네트워크를 통해 공유할 수 있도록 설계하였다. 대용량 저장장치의 구성을 위하여 고려할 점은 대용량 기능의 지원 여부를 확인하는 것이다. 운영체제에서는 대용량 기능을 지원하지만 하드웨어가 이를 지원하지 못한다면 저장 공간을 효율적으로 사용할 수 없게 된다. 그 반대의 경우도 마찬가지이다. 따라서 이러한 통합 저장 공간을 운영하기 위한 파일서버 구축 시 하드웨어나 운영체제가 대용량 공간을 지원하는지에 대해 반드시 확인하여야 한다.

KMTNet 자료처리 시스템을 구성하는 각 서버의 운영체제는 Enterprise Linux 기반의 64bit CentOS를 사용할 계획이다. 일반적으로 Redhat 계열 Linux는 EXT3 (Extended file system 3) 파일 시스템을 기본으로 사용하고 있다. EXT3는 2001년 리눅스 커널 2.4.15에서 처음 사용된 저널링 파일 시스템으로 기존의 EXT2 파일 시스템과는 완벽히 호환되는 장점을 가진다. 그러나 EXT3로 파일 시스템을 구성할 경우 생성할 수 있는 최대 볼륨의 크기는 16TiB³⁾ (기본 블록사이즈 4096KB 사용) 이고 최대 파일사이즈는 2TiB (커널 버전에 따라 차이가 나며 최대 1TiB를 지원하는 경우도 있다)를 넘지 못하여 KMTNet에서 요구하는 용량과 크게 차이가 난다. 게다가 다른 저널링 파일 시스템에 비해 속도가 느리고 EXT2와의 호환성 향상 때문에 타 저널링 파일 시스템에 비해 성능이 떨어지는 편이다. 또한 조각 모음을 수행하는 기능이 없어 장기간 사용 시 단편화된 파일들로 인한 데이터의 입출력 에러가 발생하고 이런 경우 데이터의 손실로 이어진다. 이러한 이유로 우리는 EXT3 파일시스템 대신 XFS 파일시스템을 채택하였다. 지금까지 배포된 CentOS의 최신 버전은 5.3로 공식버전의 커널에서는 XFS를 지원하지 않는다. 하지만 함께 제공하는 커널 패치를 통하여 XFS 파일시스템을 적용할 수 있고, 차기 버전인 CentOS 5.4 부터는 커널의 패치없이 XFS를 공식 지원한다.

XFS (eXtended File System)는 1994년 Silicon Graphics에서 안정성과 고성능을 필요로 하는 중대형 서버를 위해 개발된 파일 시스템으로 1990년 후반에 리눅스로 포팅되었다. XFS는 리눅스 커널 2.4.25 이후 버전부터 커

3) 전통적으로 사용하는 1GB (기가바이트) = 1000⁴, 1TB (테라바이트) = 1,000⁵, 1EB (엑사바이트) = 1000⁶ 와 구분하여 정확한 표현을 위해 2진 배수표기법으로 나타냄. 1GiB (기비바이트) = 2⁴⁰, 1TiB (테비바이트) = 2⁵⁰, 1EiB (엑스비바이트) = 2⁶⁰

표 2. 외계행성 모델분석 서버의 사양

외계행성 모델분석 서버	
CPU	Intel Quad-Core Xeon E5420 2.5GHz × 24 (96Core)
RAM	DDR3 768GB (8GB/Core)
HDD	500GB SATA2
OS	CentOS 5.3 (EXT3 filesystem)

널 소스에 포함되었으나, 각각의 OS 배포판에 따라 지원 유무가 다르다. 64bit 주소를 지원하여 생성 가능한 파일시스템의 최대 볼륨 크기는 16EiB이고, 최대 파일 크기는 8EiB로 KMTNet에서 필요한 용량을 충분히 만족 시킨다. 128bit 주소를 지원하는 Solaris ZFS 파일시스템도 있지만 아직 리눅스에서 입출력 부분이 제 성능을 내지 못하여 고려하지 않았다.

KMTNet을 통해 생성되는 관측자료와 처리결과의 저장에는 Enterprise급의 Disk Array 장비를 사용한다. Disk Array는 수백개의 하드디스크를 장착할 수 있는 Enclosure 모듈과 이들을 직렬로 연결할 수 있는 병렬 컨트롤러를 Rack에 장착하여 Array를 구성한다. Disk Array는 네트워크 형태로 자료처리 시스템의 각 서버들에 연결되는데 병목현상을 줄이기 위하여 자료전송에 Fiber Channel로 연결하여 사용한다. 일반 UTP 케이블을 사용할 경우 최대 1Gbit의 전송속도 내에서 자료의 송신과 수신이 이루어지는데 반해 Fiber Channel은 2Gbit 이상의 전송속도로 자료의 송신과 수신이 가능해 효율적인 전송이 가능하다. Disk Array의 대용량 파일 시스템 구성은 RAID5나 RAID6를 사용하므로 몇 개의 디스크에서 고장이 발생하여도 데이터 손실이 발생하지 않는다. 또한 HDD의 고장시 컨트롤러가 관리자에게 경고메시지를 보내주도록 설정할 수 있어 효율적인 관리가 가능하다.

2.4 외계행성 모델분석 서버 (Analysis)

외계행성 모델분석 서버를 2009년 3월에 구성하였다. 이 서버는 외계행성에 의해 변형된 광도곡선을 분석모델과 최적화시켜 행성 파라미터를 구하는 작업을 수행한다. 여러 개의 변수에 대해 다양한 값을 대입하여 반복 계산하기 때문에 매우 긴 계산 시간이 요구된다. 반복계산을 효율적으로 처리하기 위하여 리눅스 클러스터를 이용하였고, 사용한 리눅스 클러스터의 사양을 표 2에 나타내었다.

분석모델은 Intel Fortran Compiler와 셸스크립트를 이용한 분산처리 기법을 사용한다. 측광 데이터베이스에



그림 4. 외계행성 모델분석 서버

저장된 측광결과는 모델분석 서버로 입력된 후, 모델 최적화를 통해 결정된 행성 파라미터가 측광데이터베이스에 다시 기록된다. 그림 4는 우리가 구성한 리눅스 클러스터의 모습이다.

3. 측광 데이터베이스의 구축과 시험

KMTNet을 이용하여 변광을 검출하고 각각의 측광결과를 저장 및 관리하기 위해서는 대규모 고성능의 데이터베이스를 구축하여야 한다. 현재 여러 분야에서 일반적으로 사용하는 데이터베이스 시스템은 MySQL Enterprise, ORACLE 등이 있다. 그러나 MySQL은 ORACLE 데이터베이스와는 달리 테이블의 크기에 제약이 있어 대용량의 데이터베이스를 구축하는데 어려움을 가지고 있다. 따라서 우리는 ORACLE 11g Release 2를 이용하여 데이터베이스를 구성하였다.

표 3. 관측자료 테이블

Entity	Attribute
Raw DB	Site ID(Number(1)), HJD(Number(9.5), Field ID(Number(1)), Raw(BFILE), Image(BLOB)
Site DB	Site ID(Number(1)), Site Name(VARCHAR2(10))
Field DB	Field ID(Number(1)), Field Name(VARCHAR2(20))

표 4. KMTNet 측광 데이터베이스 테이블

Entity	Attribute
FLAG DB	HJD(Number(9.5), Field ID(VARCHAR2(10), FLAG(VARCHAR2(40))
Site DB	Site ID(Number(1)), Site Name(VARCHAR2(10))
Field DB	Field ID(VARCHAR2(10)), Field Name(VARCHAR2(20))
Catalog DB	Star ID(Number(9)), Field ID(VARCHAR2(10)), RA(Number(8,5)), DEC(Number(8,5)), X(Number(5)), Y(Number(5))
Observe DB	Site ID(Number(1)), Field ID(VARCHAR2(10)), Star ID(Number(9)), HJD(Number(10,5)), Mag(Number(7,5)), Mag_err(Number(7,5)), Filter(VARCHAR(1)), Obs_Flag(VARCHAR2(20))
Variable DB	Star ID(Number(9)), Mag_max(Number(7,5)), Mmag_min(Number(7,5)), Filter(VARCHAR(1)), Type(VARCHAR2(5))

3.1 측광 데이터베이스 구축

각 관측소에서 보내온 자료는 데이터베이스에 저장되어 관리되도록 설계하였다. 관측자료 테이블에 저장되는 요소는 관측소 (Site ID), 관측시각(HJD), 관측영역 (Sector ID), 관측영상 (Raw), 관측자료의 확인을 위한 JPG나 GIF 영상 (Image)이다. 표 3는 관측자료 저장을 위한 테이블의 구조를 나타낸다.

전처리를 마친 관측자료의 측광은 영상차감법을 이용하여 이루어지고 측광결과는 ObserveDB에 저장된다. 이 테이블이 검색 기준으로 사용되며, 다른 5개의 테이블을 참조하여 자료의 저장을 효율적으로 수행한다. 저장되는 자료의 양을 줄이기 위하여 각 레코드에서 관측소, 관측영역의 정보는 ID를 부여하여 관리하고, 이를 참조하여 별의 기본정보를 찾도록 설정하였다. 관측자료에서 관측소와 관측영역을 검색하여 해당 값을 출력하는 SQL 검색문의 한 예는 아래와 같다. 다른 참조값이 필요한 경우 해당 테이블의 값을 SQL 검색 질의에 추가할 수 있다.

```
SQL> select SiteName, FieldName, StarId, Imag
from SiteDB, FieldDB, ObserveDB
where SiteDB.starID = ObserveDB.StarID
and FieldDB.FieldID = ObserveDB.FieldID;
```

SiteName	FieldName	StarID	I_mag
칠레	FA	12345	17.123

KMTNet 데이터베이스 구성을 위하여 표 4과 같은 속성 및 자료형으로 각각의 테이블을 설계하였다. Catalog DB는 별의 기본정보를 담고 있는 테이블로 관측영역 내의 별들의 RA/DEC값, 픽셀의 좌표값 등이 저장된다. ObserveDB에는 영상차감법으로 측광된 결과가 입력되고 검색의 기준이 된다. Field DB와 Site DB는 참조 테이블이다. Variable DB는 Observe DB에서 입력된 자료 중 최대값과 최소값 및 변광형태를 저장한다. 그림 5는 각각의 테이블 간 관계를 나타내었다.

테이블의 생성시 숫자는 NUMBER 형을, 문자는 VARCHAR2 형을 선언하였다. 만약 10Byte의 자료를

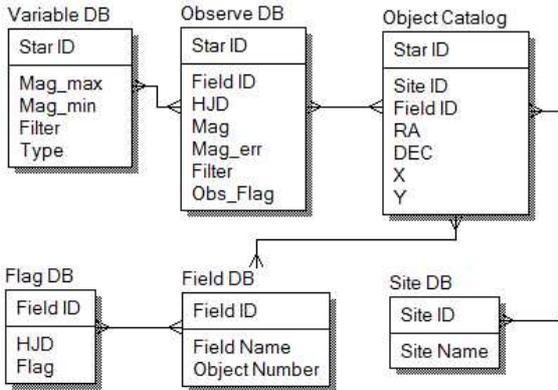


그림 5. 데이터베이스 테이블 관계도

입력할 경우 레코드의 형식이 VARCHAR(20)로 선언되면 입력 자료의 크기에 상관없이 20Byte가 사용되지만, VARCHAR2(20)로 선언할 경우에는 입력자료의 크기인 10Byte만 사용한다. 따라서 입력되는 자료의 편차가 심한 데이터를 저장할 경우 VARCHAR2형으로 선언하면 저장 공간의 낭비를 막을 수 있다. 관측자료 테이블의 BLOB (Binary Large Object) 형은 바이너리 자료를 테이블 내에 직접 저장하고, BFILE (Binary File) 형은 바이너리 자료를 외부 파일 형태로 저장한다. 두 형태 모두 4GB까지 저장이 가능하다.

3.2 측광 데이터베이스의 시험 (DataBase)

구성한 측광 데이터베이스를 이용하여 데이터베이스에 자료의 입력과 검색을 시험했다. 아직까지 생산된 관측자료가 없기 때문에 OGLE 데이터베이스에서 자료를 추출하여 사용하였다 (Szymański 2005). OGLE 관측결과는 13개 필드와 70,000,000 레코드를 가지고 있으며 자료의 입력을 위하여 별도의 임시 테이블을 생성하였다. 우리는 이번 시험을 위해 각기 다른 구성을 갖는 두 개의 서버를 사용하였다. 시험에 사용한 두 서버의 성능을 표 5에 나타내었다.

OGLE 측광자료를 측광 데이터베이스의 테이블에 입력하기 위하여 테이블이나 각종 매체로부터 입력이 가능한 SQL*Loader를 이용하였다. OGLE 자료의 입력에 소요되는 시간은 각각의 서버에서 55분과 260분이었으며, 최적화 과정을 거치지 않은 입력 자료의 검색시간은 각각 2분과 7분이었다. 데이터베이스의 최적화 후에 속도 개선을 알아보기 위하여 오라클에서 가장 간단한 최적화방법인 Index를 적용한 후 재시험하였다. Index는 SQL 명령문의 처리 속도를 향상시키기 위해 컬럼에 생성하는 ORACLE 객체로 B*트리 구조를 가지고 있다 (성윤정과 이은정 2006). Index 최적화 수행 시간은 19

표 5. 시험에 사용한 서버의 사양

	서버 1	서버 2
CPU	Intel Core i7 920 2.66GHz	Intel Xeon 5130 2GHz
RAM	6GB (DDR3)	2GB (DDR2)
HDD	500GB SATA	500GB SATA
OS	CentOS 5.3 (XFS filesystem)	CentOS 5.2 (EXT3 filesystem)
Database	ORACLE 11g Release 2	ORACLE 11g Release 1

표 6. 자료 입력 및 검색 속도

	서버 1	서버 2
OGLE 자료 입력시간	55min	260min
검색시간 (최적화 전)	2min	7min
검색시간 (최적화 후)	> 0.1sec	> 1sec

분이었고, Index 수행 후 검색시간은 1초 미만으로 줄어들었다. 표 6에 OGLE 자료의 입력과 검색에 소요된 시간을 나타냈다. 서버 1과 서버 2의 검색시간 차이는 RAM 용량 때문이다. RAM의 메모리 점유는 운영체제가 기본적으로 차지하는 공간을 제외한 나머지 공간을 데이터베이스 엔진 프로그램이 사용하는데, 서버 2의 경우 2GB의 공간에서 운영체제가 차지하는 공간을 제외하면 가용공간이 적어 디스크에서 한 번에 읽어올 수 있는 양이 제한되고, 이로 인해 입출력이 잦아져 속도가 느려진다.

3.3 웹페이지를 통한 검색 (Web)

최근 수행된 여러 탐색관측 프로젝트의 결과는 모두 웹 페이지를 통하여 사용자에게 제공된다. 우리도 KMTNet의 측광결과를 웹에서 검색할 수 있게 측광데이터베이스와 웹서버가 서로 연동하도록 설계하였다. Linux 배포판을 설치할 경우 Apache Web Server와 PHP 프로그램이 MySQL을 기본 데이터베이스 엔진으로 설정한다. 따라서 이들을 제거하고 설정을 변경하여 PHP가 ORACLE을 지원할 수 있도록 재설치하였다. 실험에 사용한 데몬은 Apache Web server 2.2.14 버전과 PHP

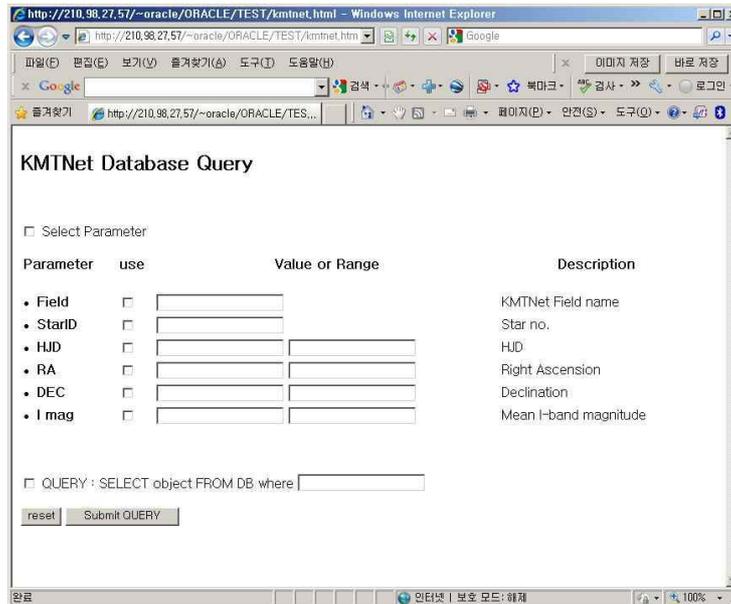


그림 6. 시험중인 KMTNet 검색 페이지

표 7. 측광자료처리 서버와 측광데이터베이스 서버의 최소 요구 사양

	측광데이터베이스 서버	측광자료처리 서버
CPU	AMD 6-Core Opteron 8431 2.4GHz × 2 (12 Core)	AMD 6-Core Opteron 8431 2.4GHz × 6 (36Core)
RAM	48GB (4GB/core)	72GB (2GB/Core)
HDD	SAS 146GB × 4 (RAID 0+1)	SSD 128GB × 4 (SLC, RAID 0)
OS	CentOS	CentOS

5.2.11 버전이며 ORACLE 연동을 위해 PHP 설치시 `--with-oci8` 옵션을 사용하였다. 검색 페이지는 라디오버튼을 통하여 Field, Star, RA, DEC 등 사용자가 필요로 하는 검색 요소를 선택하는 방법과 SQL 질의문을 직접 입력하는 방법을 제공한다. 그림 6은 웹에서 제공하는 검색 화면이다.

4. 결론 및 토의

우리는 이 연구를 통하여 KMTNet의 자료처리 시스템을 구성하는 각각의 하드웨어를 설계하였다. 또한 측광

결과를 저장하고 관리하는 측광 데이터베이스를 구축하고 OGLE 관측자료를 이용하여 시험하였다. 데이터 아카이브 및 파일서버는 관측자료의 저장과 관리를 위해 연간 약 360TB 정도의 저장 공간을 필요로 한다. 매년 증가되는 저장 공간과 관측자료의 이동을 고려할 때 외부 충격으로부터 자료를 보호할 수 있는 LTO를 사용한다. 또한 LTO Library를 데이터 아카이브에 추가할 경우 Disk Array에서 관측자료가 저장되는 공간을 줄일 수 있기 때문에 데이터 아카이브의 구축비용을 절감하는 장점을 가진다. 입출력으로 발생하는 병목현상의 해결을 위해 4Gb Fiber Channel로 각각의 서버와 연결하

도록 설계하였다.

측광자료처리 서버는 20K×20K 이미지를 전처리 하고 여러 개의 조각 영상으로 분리한 후 분산처리가 이루어 지는 클러스터 시스템을 설계하였다. 디스크 접근 시간의 손실을 낮추기 위하여 대용량 RAM과 SSD (Solid State Disk)를 장착하도록 설계하였다. 표 7은 구축할 측광자료처리 서버와 측광데이터베이스 서버에 요구되는 최소 사양이다.

측광 데이터베이스는 측광자료 테이블과 5개의 참조 테이블로 구성하였다. 테이블의 구성 시 일정한 크기를 갖지 않아 자료 입력시 발생하는 저장공간의 낭비를 막기 위하여 VARCHAR2를 사용하였다. 측광 데이터베이스는 웹서버와 연동하여 웹페이지를 통한 검색화면 제공이 가능토록 하였다. 측광 데이터베이스의 시험을 위해 OGLE 측광자료를 이용하여 검색속도와 안정성을 시험한 결과 검색속도는 1초 미만으로 KMTNet의 사용에 만족할 만한 수준이었고, 여러 조건에서의 시험 결과 안정적으로 작동하였다.

현재 영상차감법을 이용한 측광자료처리 파이프라인은 독립적으로 시험과 수정과정을 거쳐 개발 중이며, 각각의 단계별 처리상태 및 결과를 측광데이터베이스에서 관리하는 과정이 필요하다. 또한 우리가 측광 데이터베이스의 시험을 위해 사용한 OGLE 결과에 비해 KMTNet의 측광결과는 더욱 방대할 것으로 예상된다. 따라서 자료의 크기가 계속 증가할 경우 인덱스를 통한 최적화에는 한계가 있어 데이터베이스의 클러스터링 방법 등의 다양한 최적화 방안 마련과 시험이 뒤따라야 할 것이다.

참고 문헌

- 성윤정, 이은정, 2006, 오라클 10g 입문+PL/SQL, 대림출판사
- Alard, C., & Lupton, Robert H., 1998, A Method for Optimal Image Subtraction, *ApJ*, 503, 325
- Bond, I. A., et al., 2001, Real-time difference imaging analysis of MOA Galactic bulge observations during 2000, *MNRAS*, 327, 868
- Gaudi, B. S., et al., 2008, Discovery of a Jupiter/Saturn Analog with Gravitational Microlensing, *Science*, 319, 927
- Gould, A., et al., 2006, Microlens OGLE-2005-BLG-169 Implies That Cool Neptune-like Planets Are Common, *ApJ*, 644, 37
- Szymański, M. K., 2005, The Optical Gravitational Lensing Experiment. Internet Access to the OGLE Photometry Data Set: OGLE-II BVI maps and I-band data, *Acta Astronomica*, 55, 43
- Udalski, A., Szymanski, M., Kaluzny, J., Kubiak, M., & Mateo, M., 1992, The Optical Gravitational Lensing Experiment, *Acta Astronomica*, 42, 253
- Udalski, A., et al., 2005, A Jovian-Mass Planet in Microlensing Event OGLE-2005-BLG-071, *ApJ*, 628, 109
- Wolszczan, A. & Frail, D. A., 1992, A planetary system around the millisecond pulsar PSR1257 + 12, *Nature*, 355, 145