

## 위성 데이터 분산처리 시스템 최적화 및 안정화

최윤수\*, 이원구\*, 이민호\*, 김선태\*, 이상환\*

### Optimization and Stabilization of Satellite Data Distributed Processing System

Yun-Soo Choi\*, Won-Goo Lee\*, Min-Ho Lee\*, Sun-Tae Kim\*, Sang-Hwan Lee\*

#### 요약

본 논문은 다양한 분야에서 활용되고 있는 초기의 위성 데이터가 자체적으로 내포하고 있는 많은 왜곡 요소(구름, 광휘 등)에 대한 보정 작업을 클러스터에서 분산 처리함으로써 성능 향상 및 안정성을 제공해 주는 것이 목적이다. 이를 위해 로컬 스토리지와 상태 테이블을 이용한 SGE(Sun Grid Engine) 기반 분산 처리 방식을 제안하였고, 시스템으로 구현하였다. 이를 검증하기 위해 7개의 실행 노드를 이용한 실험 결과에서는 기존 시스템보다 138.81%의 성능 향상을 가져올 수 있었으며, 분산 처리 작업에 대한 안정성 또한 확보됨을 보여주었다. 이는 위성 데이터에 대한 분산 처리 작업이 I/O바운드 작업보다는 CPU 바운드 작업에 보다 더 적합하다는 것을 보여주며, 이러한 SGE 기반 분산 처리 방식은 위성영상 데이터를 처리해야 하는 다양한 분야에서의 데이터 처리 및 분석 속도 향상을 가져올 수 있고, 더 나아가 근실시간적 서비스를 가능하게 할 것으로 기대한다.

▶ Keywords : 위성 데이터, 분산처리, 최적화, 선그리드 엔진

#### Abstract

The goal of this paper is to provide performance improvement and stability for satellite data correction of some distortions due to cloud or radiance through distributed processing on cluster. To do this, we proposed and implemented SGE(Sun Grid Engine) based distributed processing methods using local storages and a status table. In the verification, the experiment result revealed that the proposed system on seven nodes improved the processing speed by 138.81% as compare to the existing system and provided good stability as well. This result showed that the proposed

•제1저자 : 최윤수 •교신저자 : 이원구

•투고일 : 2013. 9. 14, 심사일 : 2013. 9. 24, 게재확정일 : 2013. 10. 8.

\* 한국과학기술 정보연구원 (Korea Institute of Science & Technology Information)

distributed processing work is more appropriate to process CPU bound jobs than I/O bound jobs. We expect that the proposed system will give scientists improved analysis performance in various fields and near-real time analysis services.

▶ Keywords : Satellite Data, Distributed Processing, Optimization, Sun Grid Engine

## I. 서 론

인공위성에 의해 관측된 위성 데이터는 환경, 해양, 지질, 자원, 기상 기후 등 다양한 분야에서 활용되고 있다[1][2]. 인공위성으로부터 수신된 최초의 영상데이터는 많은 왜곡을 포함하고 있으므로, 이를 이용하여 신뢰할 수 있는 분석을 하기 위해서는 영상데이터에 대한 보정 작업이 필수적으로 선행되어야 한다.

위성 데이터에 대한 보정 작업은 이를 처리하는 다양한 연구 분야에서 필수적인 요소임에도 불구하고, 그 처리 방식이 대부분 단일노드/단일코어 상에서 이루어지고 있기 때문에 많은 시간을 소비하고 있다. MODIS 센서로부터 획득된 위성 데이터를 SeaDAS를 이용하여 처리할 때, 하나의 위성 데이터 당 평균 수십~수백초의 처리 시간이 소요된다. 일반적으로 위성 데이터를 이용한 분석에 사용되는 파일의 수는 수천 개 정도로 전처리 시간에만 수일을 소모하게 된다.

단일노드/단일코어 상에서 실행되던 위성 데이터의 처리 방식을 다중노드/다중코어를 이용하여 분산처리하기 위해서는 분산처리 시에 발생하는 병목현상을 찾아내어 해결해야 한다. 이러한 문제점을 해결하고, 대규모 위성 데이터에 대한 신속한 처리를 위해, 다중노드/다중코어를 이용하는 분산처리 방식이 일부 연구되고 있지만, 아직까지 초기 단계에 머물러 있어 이에 대한 보완이 필요하다.

본 논문에서는 NASA에서 개발된 위성 데이터 처리에 사용되는 SeaDAS 패키지의 영상 처리 흐름과, SGE를 이용한 다중노드/다중코어 상에서 위성데이터를 분산 처리하는 기존 시스템을 살펴보고, 처리 방법을 최적화하여 향상된 분산 처리 성능을 실험하고 평가 한다.

본 논문의 구성은 다음과 같다. 2장에서 위성데이터 처리 패키지인 SeaDAS와 분산 처리를 위한 Sun Grid Engine에 관해 살펴보고, 3장에서 위성 데이터 분산 처리를 위한 이전 연구와 본 논문에서 제안하는 시스템을 설명한다. 4장에서 기존 시스템과 제안된 시스템에 대한 실험결과를 보여 주고, 5

장에서 결론 및 향후연구에 대해 논한다.

## II. 관련 연구

### 1. SeaDAS

SeaDAS는 1994년 SeaWiFS(Sea-viewing Wide Field of view Sensor)영상을 처리하기 위하여 시작된 프로젝트로서, 20여년 동안 지속적인 발전을 통해, OCTS(Ocean Color Temperature Scanner), CZCS(Coastal Zone Color Scanner), MODIS(Moderate-Resolution Imaging Spectroradiometer)등 다양한 인공위성 센서로부터 얻어진 영상에 대한 처리 방법을 제공하고 있다[1][2].

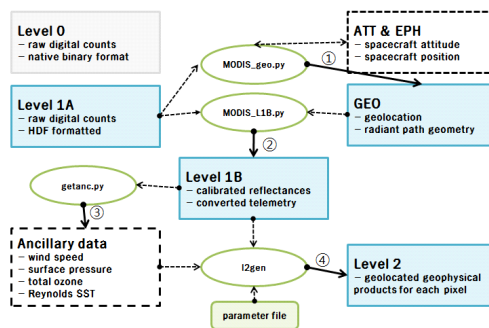


그림 1. SeaDAS를 이용한 위성데이터 처리 흐름도  
Fig. 1. Satellite data processing flow using SeaDAS

〈그림 1〉은 MODIS 센서로부터 얻어진 위성 데이터를 단일노드/단일코어 상에서 SeaDAS를 이용하여 처리하는 전체적인 흐름을 보여 준다[3][4][5][11].

단계 ①, ②는 Level-1A 파일로부터 Level-1B 파일을 생성하는 과정으로, 단계 ①에서 필요한 보조파일들을 지정된 웹사이트로부터 실시간으로 다운로드하는 작업을 수행한다. 단계 ②에서 지형정보 파일(GEO)을 생성하고, Level-1A 파일과 지형정보 파일을 이용하여 Level-1B 파

일을 생성한다.

단계 ③, ④는 Level-1B 파일로부터 Level-2 파일을 생성하는 과정으로, 단계 ③은 필요한 보조파일들을 다운로드하는 과정을 수행하고, 단계 ④는 보조파일들과 Level-1B 파일을 이용하여 Level-2 파일을 생성한다.

## 2. Sun Grid Engine

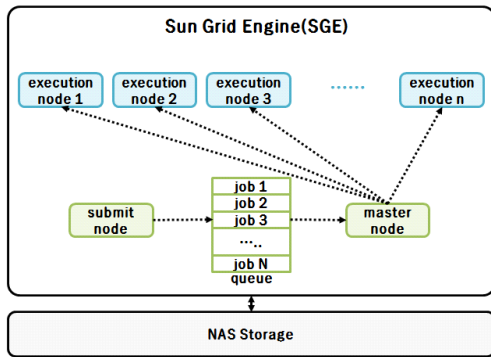


그림 2. Sun Grid Engine 구성도  
Fig. 2. The Structure of Sun Grid Engine

Sun Grid Engine(이하, SGE)은 이기종의 분산 환경에서 사용자의 작업을 분산해주는 배치 스케줄러이다. 클러스터 서버 상에서 사용되며, 다수의 분산 작업을 입력받고 작업을 스케줄링하고 관리한다. 또한 프로세서, 메모리, 디스크 같은 분산 자원의 할당을 관리하고 할당 한다[6][7].

〈그림 2〉는 SGE의 구성을 보여 준다. SGE는 크게 마스터 노드(master node), 작업 제출 노드(submit node), 작업 실행 노드(execution node)로 구성되어 있으며, 사용자가 자신의 작업을 작업 제출 노드의 작업큐에 제출하면, 마스터 노드가 자신이 관리하는 작업 실행 노드들 중에서 작업 부하가 가장 적은 노드를 선택하여 작업을 전달하는 방식으로 동작한다[8].

SGE이 동시에 처리할 수 있는 작업의 수는 작업 실행 노드들의 모든 코어수를 합한 수와 동일하며 이를 작업슬롯이라 부른다. 작업큐에 제출된 작업의 수가 전체 작업슬롯의 개수보다 많은 경우, 나머지 작업들은 작업큐에서 대기하다가 작업슬롯이 사용가능한 시점에 할당 된다[9][10][11].

## III. 분산 처리 기반 대용량 위성 데이터 처리

### 1. dSeaDAS-α

우리의 이전 연구에서 개발된 dSeaDAS-α는 단일노드/단일코어에서 수행하던 작업을 SGE를 이용하여 위성 데이터를 분산 처리하는 시스템이다. dSeaDAS-α는 SeaDAS의 처리 흐름을 분석하여 독립적으로 수행할 수 있는 부분과 병목현상이 발생하는 부분으로 구분하였고, 이를 기반으로 〈그림 3〉과 같이 단계별로 위성 데이터를 분산 처리하도록 구현하였다.

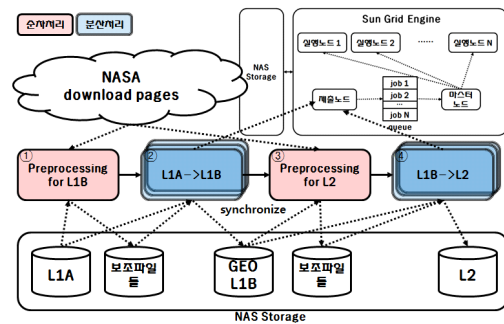


그림 3. dSeaDAS-α 구성도  
Fig 3. Architecture of dSeaDAS-α

이때, SeaDAS의 처리 흐름에서 병목현상이 발생하는 부분은 보조 파일들을 해당 웹사이트로부터 다운로드 하는 작업이다. 이러한 보조 파일은 Level-1A파일로부터 지형정보파일(GEO)과 Level-1B파일을 생성하기 위한 부분과, Level-1B 파일로부터 Level-2 파일을 생성하기 위한 부분에서 사용된다.

〈그림 3〉의 단계 ①과 단계 ③이 각각의 보조파일을 다운로드 하는 부분으로, 마스터 노드에서 순차처리 방식으로 수행된다.

〈표 1〉은 dSeaDAS-α의 단계별 작업 내용을 보여 준다. 단계 ①과 단계 ③은 순차작업이고, 단계 ②와 단계 ④는 SGE를 이용한 분산 작업이다. 단계 ①과 단계 ③은 각각 단계 ②와 단계 ④를 수행하기 위해 필요한 보조 파일들을 다운로드하는 작업을 수행하고, 단계 ②는 Level-1B파일을 생성하는 분산 작업, 단계 ④는 Level-2 파일을 생성하는 분산 작업이다.

단계 ③의 전처리 작업은 단계 ②에서 생성된 Level-1B 파일들을 입력으로 사용하기 때문에, 단계 ②와 단계 ③ 사이에서 동기화 작업이 수행되어야 한다. 단계 ②에서 모든 작업들이 중요한 후에 단계③과 단계④가 실행된다.

표 1. 단계별 작업 내용  
Table 1. Step-by-step Operations

단계	작업내용
① 순차작업	GEO, L1B 파일 생성을 위한 전처리 작업
	작업할 모든 L1A 파일들에 대해서, 웹 사이트로부터 필요한 보조 파일들을 다운로드 한다. SGE에서 처리할 작업에 대한 쉘 스크립트를 생성한다.
② 분산작업	GEO, L1B 생성 작업 (분산 작업)
	작성된 쉘 스크립트들을 SGE의 작업 제출 서버에 제출하고, SGE의 작업 실행 노드들에서 작업들이 분산되어 수행된다.
③ 순차작업	L2 파일 생성을 위한 전처리 작업
	②작업이 모두 종료할 때까지 기다린 후, 작업할 모든 L1B 파일들에 대해서, 웹 사이트로부터 필요한 보조 파일들을 다운로드 한다. SGE에서 처리할 작업에 대한 쉘 스크립트를 생성한다.
④ 분산작업	L2 생성 작업
	작성된 쉘 스크립트들을 SGE의 작업 제출 서버에 제출하고, SGE의 작업 실행 노드들에서 작업들이 분산되어 수행된다.

## 2. dSeaDAS-α의 개선방안

앞 절에서 살펴본 dSeaDAS-α는 처리 성능 및 안정성 측면에서 다음과 같은 문제점을 가지고 있다.

첫째, 분산 처리되는 위성 데이터가 NAS(Network Attached Storage) 스토리지에 위치하여 SGE의 실행 노드에서 수행되는 각각의 처리 프로세서가 NAS 스토리지에 대한 경쟁을 갖는 구조이므로, 분산 처리에 활용되는 노드 수 및 노드의 슬롯 수가 증가할수록 동시에 수행되는 프로세서의 수가 증가하여 NAS 스토리지에 대한 병목현상이 가중되는 문제점을 가지고 있다.

둘째, 단계 ③에서 Level-2를 위한 보조파일을 다운로드 하는 작업이 단계 ②의 Level-1B 생성 작업이 완전히 종료된 다음에 수행하도록 구현되어 있기 때문에, 단계 ②의 분산 작업을 통해 처리되는 위성 데이터 중 먼저 처리된 위성 데이터들도 마지막 위성데이터가 처리될 때까지 이후 작업을 진행하지 못하고 대기해야 한다. 이로 인해 전체 처리 시간에서 대기 시간 만큼의 처리 시간의 지연이 발생한다.

셋째, 분산 처리 전체 작업에 대한 트랜잭션을 지원하지

않는다. 즉, 분산 처리 작업이 도중에 실패하였을 때, 실패한 지점부터 자동으로 복원될 수 없고, 모든 작업이 처음부터 다시 수행되어야 한다.

본 논문에서는 이러한 dSeaDAS-α에서의 세 가지 문제점을 해결함으로써 분산 처리 성능과 안정성을 동시에 향상시키고자 하였으며, 그 방안을 다음과 같이 제안 한다.

첫째, NAS 스토리지에 대한 경쟁 문제를 해결하기 위해 로컬 스토리지를 이용한다. 분산 처리를 위한 각각의 실행 노드에서 처리하고자 하는 위성 데이터를 자신의 노드로 복사하여 작업을 수행하고, 작업이 완료된 단계별 위성 데이터를 NAS 스토리지로 복사하는 방법을 채택한다.

둘째, 중간 단계에서의 동기화로 인한 대기 시간을 제거하기 위해, 위성 데이터 처리에 대한 진행 정보를 기록하는 '상태 테이블'을 도입한다. 마스터 노드의 마스터 프로세서와 실행노드의 위성 데이터 변환 프로세서가 이 '상태 테이블'을 이용하여 단계별 작업을 동기화 하도록 한다.

셋째, 전체 분산 작업에 대한 트랜잭션을 보장하기 위하여, '작업 테이블'과 '상태 테이블'을 활용한다. 두 테이블을 이용하여 분산 작업이 비정상적으로 종료하였을 경우, 전체 작업을 복원할 수 있도록 한다.

## 3. dSeaDAS-β

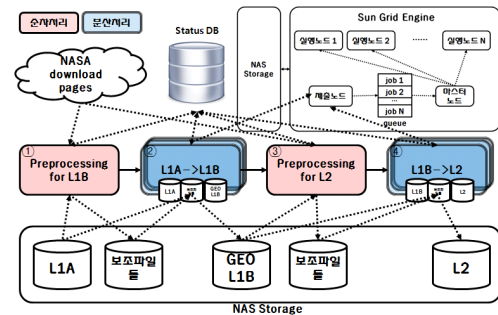


그림 4. dSeaDAS-β 구성도  
Fig 4. Architecture of dSeaDAS-β

(그림 4)는 본 논문에서 제안하는 위성데이터 분산처리 시스템(dSeaDAS-β)이다.

(그림 3)과 비교하여 볼 때, 상태 데이터베이스<sup>1)</sup>가 추가되어 있고, 마스터 노드와 실행 노드에서 상태 데이터베이스(status db)를 이용하고, 단계 ②, ④의 분산 처리 부분에서 로컬 스토리지를 이용함을 알 수 있다. 또한 기존 시스템은

1) 상태 데이터베이스 내에 작업 테이블과 상태 테이블이 생성된다.

단계 ②와 단계 ④를 위한 분산 처리 스크립트가 독립적으로 존재하지만, 본 논문의 시스템에서는 하나의 통합된 스크립트로 제공된다.

〈표 2〉는 본 논문에서 제안하는 ‘상태 테이블’을 보여준다. 상태 테이블은 작업 단위로 생성되고, 작업이 종료되면 상태 테이블은 더 이상 사용되지 않는다. 상태 테이블의 이름은 작업이 수행되는 시점에서 ‘Time Stamp(〈표 3〉 참조)’ 값을 사용하여 부여된다. 상태 테이블에 작업 내에서 처리하는 모든 위성데이터(Level-1A)에 대한 정보가 튜플로 생성되고, 각 튜플의 필드들이 초기화 된다.

표 2. 상태 테이블 스키마  
Table2. Status Table Schema

필드	형식	비고
id	STRING	기본키('0000'~'9999')
filename	STRING	L1A파일 이름
l1a_flag	BOOL	생성 시 TRUE
geo_flag	BOOL	GEO 파일 생성 시 TRUE
l1b_flag	BOOL	L1B파일 생성 시 TRUE
l2_flag	BOOL	L2 파일 생성 시 TRUE
l1b_anc_flag	BOOL	L1B용 보조 파일 다운로드 완료 시 TRUE
l2_anc_flag	BOOL	L2용 보조 파일 다운로드 완료 시 TRUE

접미사가 ‘flag’인 필드명은 l1a\_flag를 제외하고 모두 거짓(FALSE)값을 갖도록 초기화 된다. l1a\_flag, l1b\_anc\_flag, l2\_anc\_flag는 마스터 노드의 마스터 스크립트에 의해 설정되고, geo\_flag, l1b\_flag, l2\_flag는 실행노드의 영상 데이터 변환 스크립트에 의해 설정된다.

위성 데이터 변환 작업에서 첫 번째 작업인 Level-1B 생성을 위한 보조파일 다운로드 작업이 마스터 노드에서 종료하면, ‘l1b\_anc\_flag’가 참(TRUE)으로 설정되고, 분산 처리 작업을 위한 스크립트들이 생성되어 SGE를 통하여 각각의 실행노드에 전달되고, 마스터 노드는 이후 작업을 위해 상태 테이블을 주기적으로 조사한다.

실행노드에서 각각의 프로세서는 자신이 처리해야할 위성 데이터를 NAS 스토리지로부터 로컬스토리지로 복사한 후, 지형정보 파일(GEO)과 Level-1B 파일을 생성한다. 생성이 완료된 이 두 파일을 NAS 스토리지로 다시 복사한 후, 해당되는 튜플의 geo\_flag와 l1b\_flag를 참(TRUE)으로 설정하고, 상태 테이블의 l2\_anc\_flag가 참(TRUE)으로 설정될 때까지 대기한다.

l1b\_flag가 참(TRUE)으로 설정된 튜플들은 마스터 노드에 의해 확인되며, 마스터 노드는 다음 작업인 Level-2 파일을 위한 보조 파일을 다운로드하고, 해당되는 튜플의 l2\_anc\_flag를 참(TRUE)으로 설정하고, 더 이상 다운로드할 작업이 존재하지 않으면, Level-2 파일에 대한 처리가 모두 종료할 때까지 대기 한다.

실행노드에서 대기하고 있던 위성 데이터 변환 프로세서가 l2\_anc\_flag가 참(TRUE)이 되었음을 확인하면, 관련된 보조 파일들을 NAS 스토리지로부터 로컬 스토리지로 복사 후, Level-2 생성 작업을 수행하고, 생성된 Level-2 파일을 NAS 스토리지로 복사한 후, l2\_flag를 참(TRUE)으로 설정 후 작업을 종료한다.

〈표 3〉은 본 논문에서 사용하는 ‘작업 테이블’이다. 새로운 작업이 수행될 때마다, 작업테이블에 작업에 대한 신규 튜플이 생성된다. 〈그림 5〉와 같이 작업 테이블과 상태 테이블의 관계는 1:N의 관계이다. 상태 테이블은 작업 단위로 생성되며, 이와 연동하기 위하여 작업 테이블에 신규 튜플이 생성된다.

작업 테이블의 신규 튜플은 신규 작업의 ‘타임스탬프’를 기본키(job\_id)로 할당하고, 연관된 정보를 초기화 한다. 작업 테이블은 전체 작업 시간에 대한 정보를 제공하고, job\_status 필드를 이용하여 현재 작업이 진행 중인지 종료하였는지에 대한 정보를 제공한다.

표 3. 작업 테이블 스키마  
Table 3. Work Table Schema

필드	형식	비고
job_id	STRING	Primary Key (= Time Stamp)
table_name	STRING	상태 테이블 이름(Status table name)
job_count	INT	상태 테이블의 작업 수(레코드 수)
stime	STRING	시작시간 (YYYYMMDD_hhmmss)
etime	STRING	종료시간 (YYYYMMDD_hhmmss)
job_status	BOOL	상태 테이블의 작업 정보 (TURE : 작업 중)

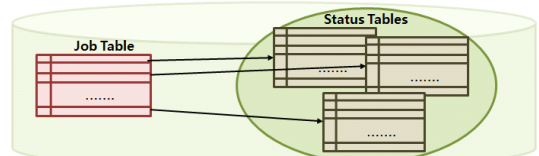


그림 5. 작업 테이블과 상태 테이블 관계  
Fig 5. Relationship of Work Table and Status Table

2) 한 번에 분산 처리하는 위성 데이터 집합 전체에 대한 작업을 의미한다. 개별 위성 데이터 파일에 대한 작업은 ‘태스크’로 명명한다.

기존 시스템(dSeaDAS-α)에서 문제로 대두되었던 작업 실패에 대한 처리를 작업 테이블과 상태 테이블을 이용하여 해

결하였다.

작업 수행 시, 실행 노드에서 생성된 위성 데이터는 로컬 스토리지에서 NAS 스토리지로 완전히 복사가 종료된 이후에, 상태 테이블의 연관 튜플의 필드값을 참(TRUE)으로 설정한다. 즉, 어떤 시점에서 전체 작업이 실패로 종료하더라도, 필드값이 참(TRUE)이면 이 부분적인 작업은 성공하였음이 보장된다.

실패된 작업을 재수행할 때, 이 상태 테이블의 정보를 이용하여 수행된 작업은 더 이상 수행하지 않고, 수행되지 않은 작업들만 수행하도록 분산 처리 프로세서를 구성하여, 안정적인 작업 수행을 하도록 구현하였다.

#### IV. 실험 및 성능평가

본 논문에서 사용한 클러스터 사양은 <표 4>와 같다. 클러스터는 1개의 마스터 노드와 7개의 실행 노드로 구성되어 있다. 각 실행노드는 8개의 슬롯(8Core)을 보유하고 있으므로, 전체 56개의 슬롯을 동시에 활용할 수 있다.

표 4. 클러스터 서버 사양  
Table 4. Cluster Server Spec

항목	CPU	MM	Storage	Nodes
내용	3.50GHz 8Core	32GB	NAS 3TB	Master-1 Execution-7

위성 데이터 Level-1A 70개 파일(평균 384.7M)이 실험을 위한 데이터로 사용되었다<sup>3)</sup>.

표 5. 입력데이터 및 단계별 생성데이터 정보  
Table 5. input & output data Information

단계	파일 유형	파일 크기(byte)
입력데이터	Level-1A	384,714,284.5
단계 ①,② L1A->L1B	GEO	60,685,415.29
	Level-1B	243,114,603.1
	L1B_QKM	143,090,927.2
	L1B_HKM	137,593,034.1
	TOTAL	523,798,564.4
단계 ③,④ L1B->L2	Level-2	148,610,133.3

<표 5>는 70개의 입력 데이터와 단계별 처리 후 생성된 데이터에 대한 통계를 보여 준다. Level-1B 파일 생성 시 생성되는 파일의 평균 크기(약 523.8M)는 Level-2 파일 생성 시 생성되는 파일 크기(약 148.6M)의 약 3.525배이다.

단일코어/단일노드에서 Level-1A파일로부터 Level-1B 파일을 생성하는 시간은 Level-1B파일로부터 Level-2파일을 생성하는 시간의 약 0.4배이다.

생성되는 Level-1B 파일이 Level-2 파일의 크기보다 약 3.525배인 반면, 처리되는 시간은 0.4배 이므로, Level-1B 파일을 생성하는 단계 ②는 I/O 바운드 작업으로, Level-2를 생성하는 단계 ④는 CPU 바운드 작업으로 분류할 수 있다.

70개의 입력 데이터가 SGE의 작업큐에 할당되면, 이 중 56개의 위성 데이터가 56개의 슬롯에 할당되어 처리가 진행되고, 나머지 14개의 태스크는 작업큐에서 대기한다.

위성 데이터에 대한 처리가 완료되어 유휴 슬롯이 발생하면 대기하고 있던 태스크들이 유휴 슬롯에 할당되어 나머지 작업을 수행하게 된다.

기존 시스템(dSeaDAS- $\alpha$ , Baseline)의 처리 성능을 향상시키기 위하여 로컬 스토리지를 사용하는 방법과 상태테이블을 사용하는 방법을 도입하였기 때문에, 이 두 가지 측면에서의 성능 향상을 측정하기 위해 로컬 스토리지만 사용하는 방법(dSeaDAS- $\beta$ L, Local)과 로컬 스토리지와 상태 테이블을 동시에 사용하는 방법(dSeaDAS- $\beta$ LS, Local & Status table)을 독립적으로 실험하여 그 결과를 비교하였다.

기존 시스템(dSeaDAS- $\alpha$ )을 기준으로 선정하고, 본 논문에서 제안하는 dSeaDAS- $\beta$ 의 분산 처리 성능에 대해 실험한 결과는 다음과 같다.

표 6. 실험결과  
Table 6. Experiment Result

	dSeaDAS- $\alpha$	dSeaDAS- $\beta$ L	dSeaDAS- $\beta$ LS
L1A->L1B ①,②단계	540	489.2 (110.38%)	482.4 (111.94%)
L1B->L2 ③,④단계	518	410.6 (126.16%)	279.8 (185.13%)
TOTAL	1,058	899.8 (117.58%)	762.2 (138.81%)

<표 6>과 <그림 6>은 단계별 실험결과를 보여준다. 각각 10회의 실험을 수행하여 평균값을 활용하였다. 단계 ①, ②와 단계 ③, ④에서 수행된 각각 수행된 시간과 전체시간을 측정하였다.

3) SGE 슬롯개수(56개)보다 많은 70(실행노드 당 10개)개를 선정하여 일반적인 처리 상황에서의 성능을 살펴보았다. 증가하는 실험데이터에 대한 성능 평가는 추후 확장성 실험에서 다룰 예정이다.

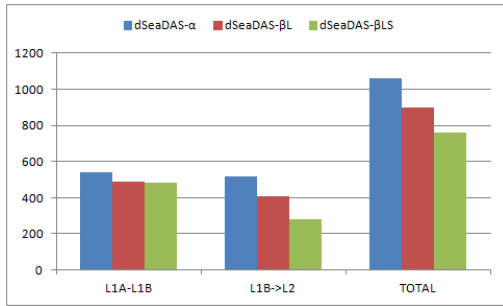


그림 6. 성능비교  
Fig 6. Performance Comparison

기준시스템(dSeaDAS- $\alpha$ )과 dSeaDAS- $\beta$ LS를 비교한 결과, 단계 ①, ②에서 111.94%의 성능 향상을, 단계 ③, ④에서 185.13%를, 전체적으로 138.81%의 성능이 향상되었음을 보여준다.

이와 같은 성능 향상 요인을 두 가지 측면에서 분석해 보면 다음과 같다.

첫째, 기존 시스템(dSeaDAS- $\alpha$ )에서 NAS 스토리지에서 직접 작업을 수행하여 문제로 지적되었던 병목현상을 제거하기 위하여 로컬 스토리지를 사용한 효과이다. 로컬 스토리지만 적용한 dSeaDAS- $\beta$ L의 실험결과를 살펴보면, 단계 ①, ②와 단계 ③, ④ 모두 성능이 향상되었음을 알 수 있다. 일반적으로 I/O바운드 작업보다 CPU 바운드 작업이 분산 처리 효과가 더 좋으므로, 110.38%의 성능 향상을 보여주는 단계 ①, ②의 처리 보다, 126.16%의 성능을 보여주는 단계 ③, ④의 처리가 더 효과적임을 알 수 있다.

둘째, 단계 ②와 단계 ③ 사이의 동기화 수행에 필요한 대기 시간에 대한 문제점을 해결하기 위해 상태 테이블을 도입한 효과이다. 단계 ①, ②는 로컬 스토리지를 사용하는 점을 제외하면, 기존 시스템(dSeaDAS- $\alpha$ )과 dSeaDAS- $\beta$ L)가 동일한 방식으로 동작하기 때문에 성능향상이 기존 시스템과 비교하여 110.38%(dSeaDAS- $\beta$ L)과 111.94%(dSeaDAS- $\beta$ LS)에 불과하지만, 단계 ③, ④는 기존 시스템에 비해 185.13% 향상되었고, 로컬 스토리지만 사용(dSeaDAS- $\beta$ L)한 경우와 비교하여, 상태 테이블을 동시에 사용(dSeaDAS- $\beta$ LS)한 경우에 146.75% 성능이 향상되었음을 보여 준다. 이는 상태 테이블을 이용하여 동기화를 위한 대기시간을 제거하는 방법이 전체 성능 향상에 상당히 기여하였음을 보여준다.

## V. 결론

전 세계적으로 환경, 해양, 지질자원, 기상/기후 등 다양한 분야에서 인공위성으로부터 제공되는 영상데이터에 기초한 많은 연구를 진행하였으며, 영상데이터의 활용성을 고려해 볼 때 앞으로 더 많은 연구가 진행될 것이다. 하지만, 연구현장에서는 인공위성 영상데이터 기반의 분석 작업을 위한 전처리 작업에 대부분의 시간을 할애할 정도로 많은 시간을 소비하고 있다. 이를 해결하기 위한 근본적인 방안은 NASA에서 제공하고 있는 SeaDAS를 포함한 영상데이터 분석시스템들을 병렬 프로그래밍화하는 것이 가장 바람직하다고 판단된다.

본 연구에서는 이러한 영상데이터 분석시스템을 분산/병렬 처리구조로 변경하기에 앞서, 전처리 작업에서의 처리속도 지연을 해결함으로써, 전체적인 분석 작업의 속도를 향상시키고자 하였으며, 더욱이 대용량 영상데이터의 분석에 소비되는 시간을 전처리 작업에 소모하고 있다는 점에 주목하여 본 연구를 수행하였다.

이를 위해, 위성 데이터 처리 프로세서인 SeaDAS를 SGE와 접목하여 대용량 위성 데이터에 대한 분산처리를 수행하였던 이전 연구의 문제점을 분석하고, 이를 해결하기 위한 방안을 연구하였다.

본 논문에서 제안된 위성 데이터 분산 처리 시스템은 데이터 처리에 대한 지역화와, 상태 테이블을 사용하여 기존 시스템의 문제점인 안정성을 해결하였고, 전체 처리 속도 또한 138.81% 향상시켰다.

제안된 시스템은 Level-1B를 생성하는 단계 ①, ②보다 Level-2를 생성하는 단계 ③, ④에서 훨씬 더 나아진 성능을 보여 주었다. 결론적으로, 분산 처리 작업이 I/O바운드 작업보다 CPU 바운드 작업에 보다 더 적합하다는 것을 알 수 있었다.

향후 연구에서는 단계 ①, ②에서 사용하는 위성 데이터에 대한 압축 등을 통하여 I/O 바운드 작업을 CPU 바운드 작업으로 전환하는 연구와, 대규모 작업 실행 노드를 갖는 환경에서 본 논문에서 제안한 분산 처리 방식에 대한 확장성 검증을 수행할 예정이다.

4) 단계 ①, ②에서 dSeaDAS- $\beta$ L과 dSeaDAS- $\beta$ LS는 동일하다

### 참고문헌

[1] M. A. Mustapha, S. Sei-Ichi, T. Lihan, "Satellite-measured seasonal variations in primary production in the scallop-farming region of the Okhotsk Sea," ICES Journal of Marine Science, Vol. 66, No. 7, pp. 1557-1569, April 2009.

[2] Adam Keith, Steve Boehinger, Euroconsult, "Satellite-based Earth Observation: Market Prospects to 2018" Euroconsult, 2009.

[3] OBPG, "SeaDAS Training Manual," [http://seadas.gsfc.nasa.gov/SeaDAS\\_Training](http://seadas.gsfc.nasa.gov/SeaDAS_Training)

[4] OBPG, "Ancillary Files for Level 1B and Level 2," <http://seadas.gsfc.nasa.gov/doc/toplevel/ancinfo.html>

[5] Sakharin Suwannathatsa, Prungchan Wongwises, "Chlorophyll distribution by oceanic model and satellite data in the Bay of Bengal and Andaman Sea," Oceanological and Hydrobiological Studies, Vol. 42, No. 2, pp. 132-138, June 2013.

[6] "Grid Scheduler/Grid Engine HOWTOs," <http://gridscheduler.sourceforge.net/howto/GridEngineHowto.html>

[7] Amit Sheth, "A new landscape for distributed and parallel data management," Distributed and Parallel Databases, Vol. 30, No. 2, pp. 101-103, April 2012.

[8] W. Gentzsch, "Sun Grid Engine: towards creating a computing power grid," Proceedings 1st IEEE/ACM Intl. Symp. on CC&G, pp. 35-36, May 2001.

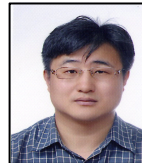
[9] Akihiro Nakamura, Jong Geol Park, Kotaro Matsushita, Kenneth J. Mackin, Eiji Nunohiro, "Development and evaluation of satellite image data analysis infrastructure," Artificial Life and Robotics, Vol. 16, No. 4, pp. 511-513, Feb. 2012.

[10] El-Sayed M. T. El-kenawy, Ali Ibraheem El-Desoky, Mohamed F. Al-rahmawy,

"Distributing Graphic Rendering using Grid Computing with Load Balancing," International Journal of Computer Applications, Vol. 47, No. 9, pp. 1-6, June 2012.

[11] Yunsoo Choi, Minho Lee, SangHwan Lee, "Evaluating the Scalability of Distributed Satellite Data Processing System," Proceeding of KSCI, Vol. 21, No. 2, pp. 395-397, July 2013.

### 저자 소개



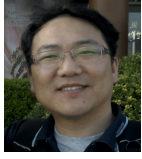
**최 윤 수**  
 1993: 충남대학교  
 컴퓨터공학과 공학사.  
 1995: 충남대학교  
 컴퓨터공학과 공학석사.  
 현 재: 한국과학기술정보연구원  
 선임연구원  
 관심분야: 빅데이터, 분산처리,  
 정보검색  
 Email : armian@kisti.re.kr



**이 원 구**  
 2000: 한남대학교  
 컴퓨터공학과 공학석사  
 2005: 한남대학교  
 컴퓨터공학과 공학박사.  
 현 재: 한국과학기술정보연구원  
 선임연구원  
 관심분야: 빅데이터, 과학데이터  
 Email : wglee@kisti.re.kr



**이 민 호**  
 2000: 충남대학교  
 컴퓨터공학과 공학석사  
 2006: 충남대학교  
 컴퓨터공학과 박사수료  
 2000: 데이콤 중앙연구소 연구원  
 현 재: 한국과학기술정보연구원  
 선임연구원  
 관심분야: 빅데이터, 정보보호  
 Email : cokeman@kisti.re.kr



김 선 태

2003: 전북대학교  
문헌정보학과 문학석사  
2012: 전북대학교  
문헌정보학과 문학박사  
현 재: 한국과학기술정보연구원  
선임연구원  
관심분야: 빅데이터, 과학데이터  
Email : stkim@kisti.re.kr



이 상 환

2004: 고려대학교  
S/W 공학과 공학석사  
2006~: 서울시립대  
컴퓨터과학부 박사수료  
현 재: 한국과학기술정보연구원  
책임연구원  
관심분야: 빅데이터, 과학데이터  
Email : sanglee@kisti.re.kr