

# 슈퍼컴퓨터의 이벤트 알림 서비스를 이용한 자동화된 후처리 시스템의 개발

이승민<sup>○</sup>, 김호윤, 김명일, 정희석, 김재성, 이상민  
한국과학기술정보연구원

{smlee76<sup>○</sup>, hoyoonkim, mikim, hsjeong, jaesungkim, smlee}@kisti.re.kr

## Development of an Automated Post Processing System Using Event Notification Service of Supercomputer

Seungmin Lee<sup>○</sup>, Hoyoon Kim, Myoungil Kim, Heeseok Jeong, Jaesung Kim, Sangmin Lee  
Korea Institute of Science and Technology Information

### 요 약

슈퍼컴퓨터는 스케줄러를 이용하여 사용자의 작업(task) 실행 및 자원할당 요청을 관리하고, 사용자는 배치 프로세스 형태로 스케줄러에 작업을 제출한다. 스케줄러에 배치 프로세스 형태로 작업을 제출하는 방식은 사용자의 작업이 비동기적으로 수행되어 작업의 상태가 변경될 경우 확인할 수 있는 서비스를 필요로 한다. 이를 위해 대부분의 슈퍼컴퓨터는 작업이 완료되거나 오류가 발생하여 중단되었을 때 전자메일을 발송하여 사용자에게 정보를 제공하는 이벤트 알림 서비스를 제공하지만, 장기간 수행되는 작업의 경우 주기적으로 전자메일을 확인해야 하는 불편함이 있고, 다수의 작업을 동시에 수행하는 경우 각각의 이벤트 정보를 포함한 전자메일을 관리하는데 어려움이 따른다. 본 논문에서는 슈퍼컴퓨터에서 제공하는 전자메일 기반 이벤트 알림 서비스를 이용하여 이벤트를 통지받을 때 지정된 방식으로 후처리 작업이 실행될 수 있는 자동화된 후처리 시스템의 설계 및 구현 방법을 제시한다.

### 1. 서 론

슈퍼컴퓨터를 이용하는 단계는 전처리(pre-processing), 해석(solving), 후처리(post-processing)로 나누어진다. 사용자는 전처리 단계에서 입력 데이터를 준비하고, 해석 단계에서 슈퍼컴퓨터를 이용하여 연산을 수행한 후 결과 데이터를 활용하여 후처리 작업을 수행한다.

인 스케줄러(scheduler)가 존재하고 사용자는 스케줄러에 자원할당 및 작업 실행을 요청한다. OSC(Ohio Supercomputing Center)의 TORQUE[1], NCSA의 TeraGrid[2], KISTI의 SGE, LoadLeveler[3]와 같이 거의 모든 슈퍼컴퓨터에는 스케줄러가 존재한다.

스케줄러에 배치 프로세스 형태로 작업을 제출하는 방식은 사용자의 작업이 비동기적으로 수행되어 작업의 상태가 변경될 경우 확인할 수 있는 서비스를 필요로 한다. 슈퍼컴퓨터의 스케줄러는 사용자의 작업에 대한 상태정보를 제공하지만, 이를 위해서는 매번 슈퍼컴퓨터에 접속을 해야하기 때문에 슈퍼컴퓨터는 작업의 완료 또는 오류 발생과 같은 이벤트에 대한 전자메일을 발송하여 사용자에게 작업의 변경된 상태 정보를 제공하는 이벤트 알림(event notification) 서비스를 제공한다. 전자메일을 확인한 사용자는 결과 데이터를 이용하여 후처리 작업을 수행하게 된다.

그러나, 슈퍼컴퓨터에서 제공하는 전자메일을 이용한 이벤트 알림 서비스는 사용자가 직접 메일을 확인해야 하기 때문에 작업이 종료되는 시점과 메일을 확인하는 시점이 일치하지 않게 되고 다수의 작업을 동시에 수행하는 경우 각각의 이벤트 정보를 포함한 메일을 관리하는데 어려움이 따른다. 특히, 슈퍼컴퓨터는 많은 사용자에게 디스크 자원을 효율적으로 제공하기 위해 사용자별로 디스크 쿼터(quota)를 할당하고, 데이터가 누적되는 것을 방지하기 위해 일정기간 사용하지 않는 데이터는

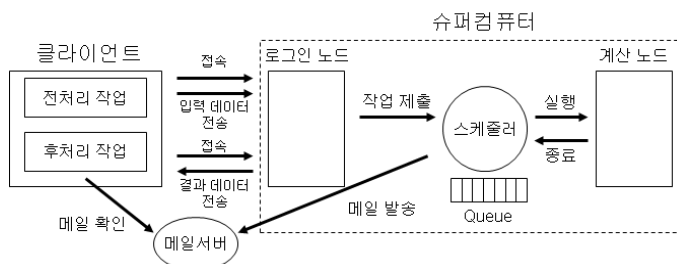


그림 1 슈퍼컴퓨터를 활용한 사용자의 작업 흐름

그림 1은 사용자가 슈퍼컴퓨터를 이용하여 작업을 수행하는 전과정을 나타낸 것이다. 전처리 단계에서 해석 단계로 진행하기 위해 사용자는 배치 프로세스(batch process) 방식으로 슈퍼컴퓨터에 작업(task)을 제출한다. 배치 프로세스 방식은 사용자와의 상호 작용을 최소화하고 사용자의 자원 요청을 효율적으로 처리할 수 있어 대부분의 슈퍼컴퓨터에서 작업제출 방식으로 이용하고 있다. 이를 위해 슈퍼컴퓨터에는 배치 프로세스 처리기

자동으로 삭제하는 정책을 적용하고 있어 장기간 수행되는 작업은 주기적으로 메일을 확인해야 한다.

이러한 불편을 해결하기 위해서는 슈퍼컴퓨터에서 제공하는 전자메일 기반 이벤트 서비스를 이용하여 이벤트를 통지받을 때 지정된 방식으로 후처리 작업이 진행될 수 있도록 하는 시스템을 필요로 한다. 본 논문에서는 현재 구축되어 있는 슈퍼컴퓨터의 전자메일을 이용한 이벤트 알림 서비스를 기존의 이벤트 알림 모델에 기초하여 분석하고, 이를 이용하여 슈퍼컴퓨터에 존재하는 결과 데이터를 사용자 컴퓨터에 자동으로 전송하는 자동화된 후처리 시스템의 설계 및 구현 방법을 제시한다.

본 논문의 구성은 다음과 같다. 2절에서는 전자메일 시스템의 구성요소를 살펴보고, 비동기 모드에서의 이벤트 알림을 위한 기존 연구를 통해 현재 구축되어 있는 전자메일 기반 이벤트 알림 서비스를 분석한다. 또한, 전자메일 서비스의 부수 효과(side effect)를 이용하여 확장된 시스템을 구축한 기존 연구를 살펴본다. 3절에서는 이벤트 알림 서비스를 이용한 자동 결과 데이터 전송 시스템을 제시한다. 4절에서는 제시한 자동 결과데이터 전송 시스템의 구현과 관련 이슈에 대하여 알아본다. 그리고, 마지막 5절에서는 결론 및 향후 연구 방향을 제시한다.

## 2. 관련 연구

본 절에서는 전자메일 시스템의 구성 요소를 알아보고, 전자메일 서비스를 이용하여 확장된 시스템을 구축한 기존 연구와 본 논문에서 제안하는 시스템과의 차이를 알아본다. 또한, 비동기 방식의 이벤트 통지에 대한 기존 연구들을 통해 슈퍼컴퓨터에서 제공하는 전자메일 기반 이벤트 통지 시스템에 대응하는 모델을 알아본다.

### 2.1. 전자메일 시스템의 구성 요소

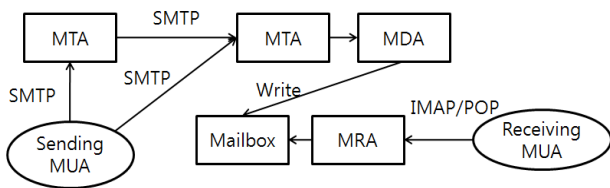


그림 2 전자메일 전송 메커니즘

그림 2는 전자메일 서비스를 이용하여 메일을 전송하고 사용자가 확인하는 메커니즘을 나타내고 있다. 그림 2에서 볼 수 있듯이 전자메일 시스템은 MTA, MDA, MRA, MUA로 구성되어 있다.

- MUA(Mail User Agent)  
이메일 클라이언트로 사용자가 메일을 읽고 쓰는 기본적인 기능과 메일을 관리하는 기능을 포함하는 프로그램을 지칭한다.
- MTA(Mail Transfer Agent)  
MUA와 다른 MTA로부터 메일을 받거나 다른 MTA로

메일을 전달하는 기능을 수행한다. MTA가 메일을 주고 받는 통신은 SMTP(Simple Mail Transfer Protocol)을 기반으로 한다.

- MDA(Mail Delivery Agent)  
MTA가 받아들인 메일을 사용자 환경에 맞도록 저장하는 역할을 수행한다.
- MRA(Mail Retrieval Agent)  
사용자가 요청하는 메일을 전달하는 기능을 수행한다. MRA와 MUA의 통신은 IMAP(Internet Message Access Protocol)/POP(Post Office Protocol)을 기반으로 한다.

### 2.2. 전자메일 서비스를 활용한 확장 시스템

전자메일 서비스를 정보 교환(communication) 수단인 아닌 공동작업과 알림(notification)의 기능으로 활용한 사례가 있다. P. Dewan[4]은 전자메일의 첨부파일 기능을 확장하여 버전관리 시스템을 연구하였고, T. Kwok[5]은 전자메일을 이용한 계약관리 시스템(Contract Management System)을 제시하였다. 두 경우 모두 잘 알려진 전자메일 서비스를 이용하여 확장성과 가용성이 높은 시스템을 구축하였다. 기존 연구는 전자메일 기능 중 첨부파일을 이용한 데이터 전송과 해당 데이터 관리를 위한 처리에 중점을 두고 있다. 즉, 버전관리 시스템은 사용자가 메일을 확인한 후 직접적인 선택을 통해 버전 관리를 결정하는 방식이고, 계약관리 시스템은 첨부파일에 워터마크(water mark)와 같은 디지털 권리 관리(DRM)를 추가하여 계약문서를 관리하는 시스템이다. 이외에도 웹기반 알림 모델을 분석한 연구가 있다[6]. 웹페이지가 변경되었을 때의 통지와 같이 웹 환경에서 발생하는 이벤트를 전자메일로 알려주는 방식이다.

본 논문에서 제안하는 시스템은 전자메일의 필터링 기능을 이용하여 일반적인 이메일과 슈퍼컴퓨터에서 생성하는 이벤트 메일을 구분하고, 비동기 방식의 이벤트 통지를 위하여 전자메일 서비스를 이용하는 점에서 기존 연구와 구별된다. 또한, 웹기반 알림 모델의 연구도 이벤트를 전자메일로 통지하지만, 이벤트가 발생하는 조건을 어떻게 지정할 것인가에 대한 연구로 전자메일 서비스를 이용한 후처리 시스템을 자동화하는 본 연구와는 차이가 있다.

### 2.3. 비동기식 이벤트 알림 모델

메시지 큐잉(Message Queuing) 모델은 메시지 교환에 큐를 이용하여 비동기식 통신이 가능하도록 정의하고 있다. 하나의 어플리케이션이 메시지 큐(queue)에 메시지를 보내고 나면, 받는 역할을 수행하는 어플리케이션은 큐에서 메시지를 가져와서 지정된 작업을 수행한다. 이러한 통신은 메시지를 받는 어플리케이션이 이미 결정되어 있기 때문에 Point-to-Point 통신이라고 한다. 이러한 Point-to-Point 방식의 대안으로 이용하는 모델이 Publish/Subscribe 모델이다. 이 모델에서 어플리케이션은 토픽(topic)이라는 특정 장소에 메시지를 퍼블리싱

(publishing)하면 메시지는 그 특정 토픽에 등록(subscribing)한 다른 어플리케이션에 의해 선택되어 이용이 가능하게 된다. 따라서 같은 메시지를 다중 클라이언트 어플리케이션이 받을 수 있다. 비동기식 통신에 적용할 수 있는 이벤트 통지 서비스는 많은 연구가 있었다. 그 중에서 P. T. Eugster[7]의 연구는 Publish/Subscribe 모델을 기반으로 한 통신과 Message Passing, RPC와 같은 다양한 통신 패러다임을 비교하며 동기 모드를 디커플링(decoupling)하기 위한 방법을 소개하고 있다.

메시지 큐잉과 Publish/Subscribe 모델을 적용한 MOM(Message-Oriented Middleware)은 제안하는 시스템과 이벤트 알림 모델 측면에서는 유사하지만, MOM은 Java 기반의 미들웨어에서 주로 구현되어 있고 본 논문에서 제안하는 방식은 확장성과 가용성이 높은 기존의 전자메일 서비스를 재사용한다는 측면에서 차이가 있다.

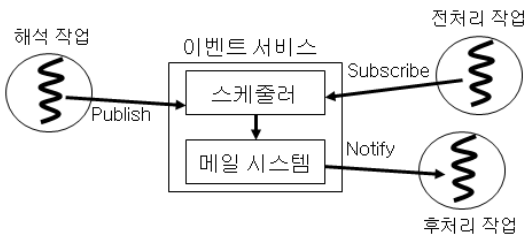


그림 3 사용자 작업 이벤트의 전달 모델

그림 3은 슈퍼컴퓨터의 이벤트 알림 서비스를 위에서 살펴 본 publish/subscribe 모델 관점에서 나타낸 것이다. 사용자가 슈퍼컴퓨터에 작업을 요청할 때 관심있는 이벤트를 등록하고, 스케줄러는 해석 작업이 완료된 후 메일 시스템을 이용하여 후처리 작업을 담당하는 프로세스에 통보한다.

### 3. 시스템 설계

본 절에서는 전자메일 시스템의 특징을 통하여 이벤트 알림 서비스를 구성하기 위한 모듈과 이를 활용한 자동화된 후처리 시스템의 설계를 제시한다.

#### 3.1. 전자메일 서비스를 이용한 이벤트 알림 시스템 설계

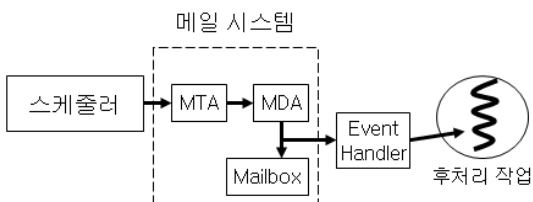


그림 4 Push 방식의 후처리 실행

그림 4는 2.1절에서 살펴 본 메일을 전달하는 과정과 메일 시스템으로부터 후처리 작업을 실행하는 메커니즘을 나타낸 것이다. 직접적으로 후처리 작업에 이벤트를 통지하고 응답(reply)을 요구하지 않는 방식으로 fire-and-forget에 해당한다.

스케줄러가 메일을 발송하면 수신측의 MTA가 전자메일을 확인하고 MDA에 전달한다. MDA가 전자메일을 메일함(Mailbox)에 저장하기 위해 관여할 때 이벤트 처리기(handler)가 이벤트 정보를 구분하고, 후처리 작업을 수행할 응용 프로그램에 해당 내용을 전달하게 된다. 이 모델은 MDA에서 모니터링 및 필터링이 이루어지게 되어 메일 시스템이 사용자의 컴퓨터에 설치되어 있는 경우에는 시스템 호출을 통하여 실행할 수 있으나, 원격에 존재하는 경우 원격 절차 호출(RPC: Remote Procedure Call)과 같은 통신이 가능한 경우에 한해 구성이 가능하다.

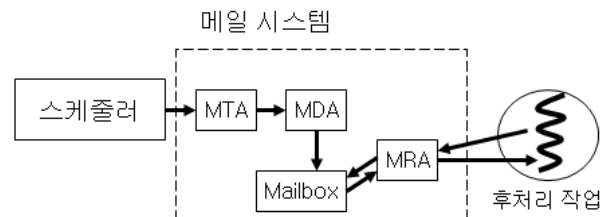


그림 5 Pull 방식의 후처리 실행

그림 5는 메일 시스템을 이용하여 후처리 작업을 실행하는 방법을 나타낸 것으로 메일을 전달하는 과정은 동일하지만, 사용자 측에서 이벤트 메일을 확인한 후 후처리 작업을 수행한다. 이러한 pull 방식의 이벤트 확인 및 후처리 수행은 메일 서버의 위치에 대한 제약이 해결되고 여러개의 전자메일을 한번에 처리할 수 있어 유용하다.

#### 3.2. 자동화된 후처리 시스템 설계

자동화된 후처리 시스템은 슈퍼컴퓨터에 데이터를 전송하고 작업을 제출하는 기능을 포함해야 한다. 개별 작업 뿐 아니라 동시에 여러 개의 작업을 실행하는 경우 작업을 구분하여 관리하기 위해서는 작업에 대한 ID를 필요로 하고 이를 데이터베이스에 관리해야만 사용자별 또는 작업 그룹별 관리가 가능해진다. 또한, 슈퍼컴퓨터에 작업을 제출할 때 스케줄러를 위해 매번 유사한 작업 스크립트(job script) 파일을 작성해야 하는 반복적인 일을 해결할 수 있다.

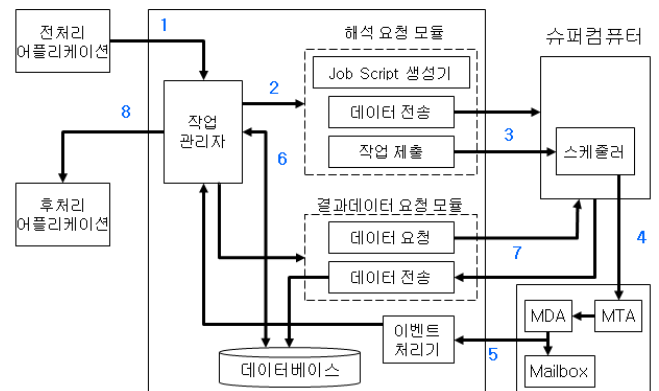


그림 6 자동화된 후처리 시스템 구조 및 실행 흐름  
그림 6은 자동화된 후처리 시스템에 필요한 모듈과 각

모듈이 수행되는 실행 순서를 나타낸 것이다. 전처리 어플리케이션이 작업 관리자에게 슈퍼컴퓨터를 활용한 해석을 요청하면 작업 관리자는 템플릿(template)을 이용하여 스케줄러에 제출해야 하는 작업 스크립트를 생성한다. 준비된 작업 스크립트와 해석에 필요한 입력 데이터를 슈퍼컴퓨터에 전송한 후 스케줄러에 작업을 제출한다. 스케줄러는 작업이 모두 종료된 후 전자메일 서비스를 이용하여 메일 서버에 이벤트 정보가 포함된 메일을 발송하고 MDA가 메일함에 메일을 저장하는 시점에서 이벤트 처리기가 메일을 확인하여 이벤트 정보를 포함한 경우 작업관리자에게 후처리 작업을 진행하도록 요청한다. 작업 관리자는 메일의 내용을 확인하여 결과 데이터를 다운로드하고 최종적으로 후처리 어플리케이션을 호출한다.

4. 시스템 구현

본 절에서는 3절에서 제시한 시스템의 구성 요소 중 가장 중요한 해석 요청, 데이터 요청 모듈, 이벤트 처리기를 구현한 예제와 두 가지 응용 분야에 적용한 내용 및 구현 관련 이슈를 살펴본다.

```
#!/bin/sh
.....
reg1=s/input_file_name/$1/g
reg2=s/num_of_cpus/$2/g
sed -e $reg1 -e $reg2 template.cmd > \
$workdir/$jobcmd
perl -pi -e 's/r//' $jobcmd
.....
```

그림 7 템플릿을 이용한 작업 스크립트 생성

그림 7은 스케줄러에 작업을 제출하기 위한 작업 스크립트를 생성하는 쉘스크립트 예제이다. 슈퍼컴퓨터에 자원을 요청할 때는 필요한 입력 데이터와 사용할 자원의 수를 명시해야 한다. 슈퍼컴퓨터에서 가장 중요한 자원은 CPU로 예제에서는 사용할 CPU의 수를 입력받아 작업 스크립트를 완성한다.

```
#!/bin/sh
.....
TRANSFER_BATCH=$JOB_ID.batch
.....
for i in $(echo $5 | sed -e "s/,/\ /g")
do
echo "$REQ_CMD $i" >> $TRANSFER_BATCH
done
psftp $USER_ID@gaia.ksc.re.kr -b \
$TRANSFER_BATCH -be
.....
```

그림 8 데이터 전송 프로그램 예제

그림 8은 스케줄러에 작업을 제출하기 전에 슈퍼컴퓨

터에 데이터를 전송할 때와 결과 데이터를 슈퍼컴퓨터로부터 받을 때 사용하는 쉘스크립트의 일부분이다. REQ\_CMD는 파일 전송 시 이용할 명령어로 슈퍼컴퓨터에 데이터를 전송할 때는 ftp의 put 명령어로 이용하고, 해석 결과 데이터를 다운로드할 때는 ftp의 get 명령어로 바꾸어 사용한다.

```
$ cat .forward
# Exim filter
if $h_subject: contains "event_notification" then
  pipe "/usr/local/bin/req_post.sh $h_subject: &"
  seen finish
endif
```

그림 9 스크립트를 이용한 Filtering Agent 구현

그림 9는 exim 프로그램에서 필터링 기능을 이용하여 후처리 작업을 호출하는 스크립트의 예제이다. 즉, exim에서 스팸 메일을 처리하는 모듈을 이용하여 메일 헤더에 "event\_notification"이라는 문자열을 체크한 후 패턴이 존재할 경우 이미 정의되어 있는 후처리 실행을 요청한다. 메일을 받을 때 자동으로 실행되어 필터링 기능을 수행할 수 있도록 설정하기 위해서는 홈디렉토리에 .forward 파일에 저장하고, '# Exim filter' 문구로 시작해야 한다.

4.1. 축류팬 유동해석 자동화 시스템

축류팬 유동해석을 자동화하는 시스템은 팬 설계자가 전처리 단계로 사용자의 컴퓨터에서 모델 및 격자를 생성하고, 그 후 슈퍼컴퓨터를 이용하여 해석을 수행한다. 후처리 작업은 해석된 결과를 사용자가 확인하기 쉽도록 Excel 및 html 형태로 보고서를 작성하고, 수치데이터를 이미지화하는 작업을 수행한다.

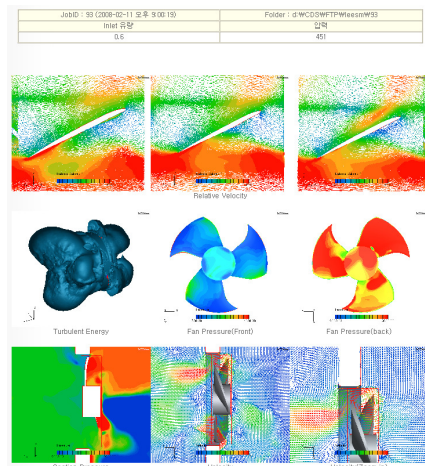


그림 10 후처리 작업 결과 이미지

그림 10은 후처리 작업이 완료되어 보고서 형태로 생성된 최종 결과물로 후처리를 위한 프로그램은 상용 S/W인 SC/Tetra, CADThru, SC/Post를 이용하였고, 결과 처리의 자동화를 위해 제안한 후처리 자동화 시스템을 적

용하였다. 유동해석의 경우 많은 자원을 이용하여 장기간 작업이 수행되기 때문에 Push 방식의 후처리 자동화 시스템을 구현하였다.[8]

#### 4.2. 자동화된 최적영상획득 시스템

최적영상획득 시스템은 차량 번호판의 인식율을 높이기 위해 시간대에 따른 태양의 고도와 주변 밝기 정보를 이용하여 최적의 차량 영상을 획득하는 시스템으로 하나의 번호판 인식을 위해 여러 장의 이미지를 비교해야 하고, 하루에도 수만대의 차량에 대한 데이터가 생성되기 때문에 데이터를 일괄적으로 슈퍼컴퓨터에 전송하고 해석이 완료되면 자동으로 후처리 작업이 수행되도록 처리해야 한다.

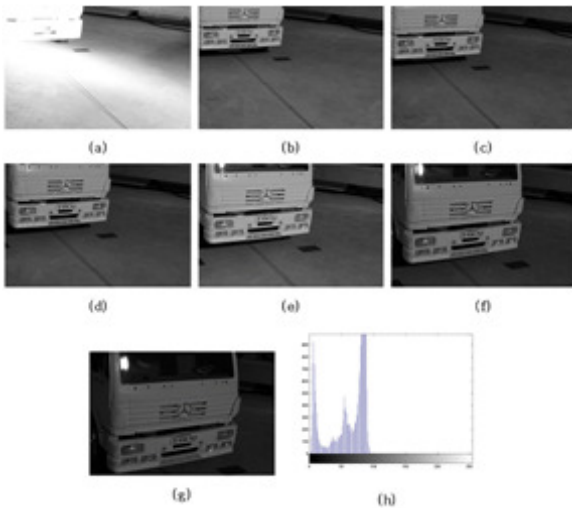


그림 11 주변 밝기에 따른 최적의 영상 추출

이 시스템은 카메라 장비에서 생성되는 이미지 데이터를 전처리 단계에서 준비하고, 해석 프로그램은 IJG(Independent JPEG Group)에서 제공하는 JPEG 라이브러리를 이용하여 구현하였다. 해석 프로그램에서는 여러 장의 이미지를 이용하여 번호판으로 인식되는 위치의 정보와 해당 영역으로부터 번호판을 인식하기 위한 최적의 영상을 추출한다. 후처리 작업은 해당 영역과 이미지를 이용하여 번호판을 인식하게 된다. 이 시스템의 경우 해석 시간이 비교적 짧지만 많은 작업을 동시에 수행한다는 특징이 있다. 따라서, push 방식으로 후처리 작업을 호출할 경우 수만개의 개별 작업이 이루어지기 때문에 pull 방식으로 후처리 자동화 시스템을 구현하였다.[9] 메일 서버가 존재하는 서버 시스템은 리눅스 운영체제로 CentOS 5.4 버전을 이용하였고, 메일 서버와 데이터베이스 서버로는 각각 Sendmail과 mysqld를 설치하였다. 다량의 이벤트 메일이 발생하는 경우 전자메일의 필터링 기능을 이용하는 방식은 동일하게 이용하지만, 이벤트 정보를 데이터베이스에 저장한다. 이후에 주기적으로 database에 질의를 통하여 새로운 이벤트를 확인하고 수집된 이벤트에 대해 일괄적으로 슈퍼컴퓨터에 결과 데이터를 요청한다.

#### 4.3. 구현 관련 이슈

사용자가 슈퍼컴퓨터에 작업을 제출하거나 결과 데이터를 얻기 위해서는 계정 정보(아이디, 비밀번호)를 이용해야 한다. 그러나, 후처리 시스템을 자동화하기 위해서는 해석을 요청하는 단계에서 제공한 계정 정보가 저장되어 있어야 하는 문제가 발생한다. 따라서, 이러한 문제점을 해결하기 위해 ssh-keygen 프로그램을 이용하여 비밀번호와 계정 정보 없이도 연결이 가능한 시스템으로 구성하거나 계정 정보를 자동화된 후처리 시스템에 전달하는 방법이 필요하다. 또 다른 이슈는 전자메일 내용 작성의 제한이다. 슈퍼컴퓨터에서 통지하는 전자메일의 본문은 스케줄러가 작성을 하게 되어 사용자는 메일의 제목에 대해서만 수정이 가능하게 된다. 그러나, 메일 제목의 크기에 제한이 있어 이벤트 구분을 위한 키워드와 작업의 구분을 위한 작업 ID 및 사용자 정보만이 포함된다.

#### 5. 결론

슈퍼컴퓨터에서 제공하는 전자메일 기반 이벤트 알림 서비스를 비동기 알림 모델에 적용하여 파악하였고, 이를 통해 후처리 단계를 자동으로 실행하기 위한 두가지 방식을 제안하였다. 제안한 시스템의 구성 요소를 정의하고, 슈퍼컴퓨터를 활용하는 실제 프로젝트에 적용하여 자동화된 후처리 시스템을 구현하였다. 시스템 설계와 구현 이슈를 통해 유사한 시스템을 활용하여 서비스를 제공하거나 슈퍼컴퓨터를 활용하기 원하는 사용자에게 필요한 모듈을 제공할 수 있을 것으로 기대한다. 향후에는 이기종의 슈퍼컴퓨터에서 작업이 수행되는 비즈니스 프로세스를 자동화하는 방법과 보안 관련 이슈를 연구할 계획이다.

#### 참고문헌

- [1] 오하이오 슈퍼컴퓨팅센터 홈페이지, <http://www.osc.edu>
- [2] NCSA 홈페이지, <http://www.ncsa.illinois.edu>
- [3] KISTI 홈페이지, <http://www.ksc.re.kr>
- [4] Prasun Dewan and Henry McEuen, "Active Notifications", CollaborateCom. 2007
- [5] Thomas Kwok, Thao Nguyen, Linh Lam and Trieu Chieu, "A Web-based and Email Driven Electronic Contract Management System", ICEBE 2007
- [6] Jameela Al-Jaroodi and Nader Mohamed, "Analysis of Web Alert Models", NBIS 2009
- [7] Patric Th. Eugster, Pascal A. Felber, Rachid Guerraoui and Anne Marie Kermarrec, "The many faces of publish/subscribe", ACM Computing Surveys, Volume 35, Issue 2, Jun 2003
- [8] 김명일, 이승민, 이상민, "슈퍼컴퓨터 활용 축류팬 유동해석 자동화 시스템 개발", 한국전산유체공학회 2008, pp. 288-291
- [9] 강현인, 곽한섭, 방경주, 김명일, 이승민, "환경 적응형, 관심영역 최적화, 자발적 학습 알고리즘이 적용된 영상 획득 유닛 개발", 중소기업선도형기술혁신전략과제 지원사업 최종보고서 2008