

<http://dx.doi.org/10.7236/JIIBC.2016.16.1.283>

JIIBC 2016-1-38

# 시스템 효율성 증대를 위한 그리드 컴퓨팅 기반의 위험 관리 시스템

## Risk Management System based on Grid Computing for the Improvement of System Efficiency

정재훈\*, 김신령\*\*, 김영곤\*\*\*

Jae-Hun Jung\*, Sin-Ryeong Kim\*\*, Young-Gon Kim\*\*\*

**요약** 최근 과학 기술이 발전함에 따라 복잡한 문제를 해결하기 위하여 고성능의 계산 자원이 필요하게 되었다. 이러한 요구를 충족시키기 위하여 지역적으로 분산되어 있는 이질적인 고성능 컴퓨팅 자원을 하나로 묶어 거대한 시스템을 구성하는 그리드 컴퓨팅에 대한 연구가 활발하게 이루어지고 있다. 하지만 제한된 자원 안에서 최대한의 결과를 얻어내기 위한 프로세스, 실시간 작업의 총 실행시간을 정확하게 예측할 수 있는 스케줄링 정책이 미흡한 실정이다. 이러한 문제점을 해결하기 위해, 본 논문에서는 시스템 효율성 증대를 위한 시스템 구조, 프로세스를 도출하고, 그리드 컴퓨팅 기반의 작업 방법론 도출, 자원(Agent)의 작업에 대한 문제점을 효율적으로 관리할 수 있는 위험 정책 모듈, 자원 할당 및 문제되는 자원들을 재할당 할 수 있는 스케줄링 기법 및 할당 기법, 자원(Agent) 모니터링을 효율적으로 관리할 수 있도록 그리드 컴퓨팅 기반의 위험 관리 시스템을 제안하였다.

**Abstract** As the development of recent science and technology, high-performance computing resources is needed to solve complex problems. To reach these requirements, it has been actively studied about grid computing that consist of a huge system which bind a heterogeneous high performance computing resources into on which are geographically dispersed. However, The current research situation which are the process to obtain the best results in the limited resources and the scheduling policy to accurately predict the total execution time of the real-time task are very poor. In this paper, in order to overcome these problems, we suggested a grid computing-based risk management system which derived from the system structure and the process for improving the efficiency of the system, grid computing-based working methodology, risk policy module which can manage efficiently the problem of the work of resources(Agent), scheduling technique and allocation method which can re-allocate the resource allocation and the resources in problem, and monitoring which can manage resources(Agent).

**Key Words** : Grid Computing, Risk management System, System Efficiency, Scheduling and resource(Agent) allocation techniques, Resource(Agent) Monitoring

\*준회원, 한국산업기술대학교 컴퓨터공학과

\*\*정회원, 동서대학교 정보통신과

\*\*\*정회원, 한국산업기술대학교 컴퓨터공학과

접수일자: 2015년 11월 18일, 수정완료: 2016년 1월 2일

게재확정일자: 2016년 2월 5일

Received: 18 November, 2015 / Revised: 2 January, 2016 /

Accepted: 5 February, 2016

\*\*\*Corresponding Author: ykkim@kpu.ac.kr

Dept. of Computer Engineering, Korea Polytechnic University, Korea

## I. 서론

최근 과학 기술이 발전함에 따라 복잡한 문제를 해결하기 위하여 고성능의 계산 자원이 필요하게 되었으며, 이러한 요구를 충족시키기 위하여 지역적으로 분산되어 있는 이질적인 고성능 컴퓨팅 자원을 하나로 묶어 거대한 시스템을 구성하는 그리드 컴퓨팅에 대한 연구가 활발하게 이루어지고 있다. 그리드 컴퓨팅은 지리적으로 분산되어 있는 고성능 컴퓨터, 대용량 저장장치, 첨단과학 설비 등 컴퓨팅 자원을 네트워크로 연동하여 병렬 분산처리 환경을 제공하는 방법이다[1].

그리드 컴퓨팅은 작업을 요청하게 되면 각 자원들의 구성에 따라 작업을 분할하게 되는데 자원들의 성능과 거리에 따라 처리 시간이 다르다.

기존의 연구 내용은 복수개의 대용량 작업을 요청하게 되면 작업의 크기나 우선순위에 따라 그리드 환경으로 구성된 자원들을 배치시키는데 연구가 진행되고 있다. 하지만 성능에 따라 먼저 완료된 유효자원은 아무런 작업을 하지 않으며 모든 작업이 끝났을 경우 다음 작업을 처리한다. 따라서 먼저 완료된 유효자원의 활용에 대한 연구는 현재까지 미비하다[2].

본 논문에서는 그리드 환경에서 자원의 재배치 및 제한된 자원 안에서 최대한의 결과를 얻어내기 위한 프로세스, 실시간 작업의 총 실행시간을 정확하게 예측할 수 있는 스케줄링 정책, 서비스 도입에 따른 위험 관리 프로세스를 정립할 수 있는 그리드 컴퓨팅 기반의 위험관리 시스템을 제안한다.

제안 방법론은 1) 시스템 효율성을 증대를 위한 시스템 프로세스 2) 그리드 컴퓨팅 기반의 작업 방법론 3) 자원(Agent) 스케줄링 및 할당 기법 4) 자원(Agent) 모니터링 5) 위험 정책 등록 및 자원(Agent) 재배치를 효율적으로 대응이 가능하게 하고자 한다.

본 논문은 다음과 같이 구성된다. 2장에서는 다중회귀 분석법, 위험관리체계, 의사결정 시스템의 동향을 기술하고, 3장에서는 본 연구의 제안방법론을 소개한다. 4장에서는 실험 및 평가를 설명하고 마지막 5장에서는 결론을 기술한다.

## II. 관련 연구

### 1. 그리드 컴퓨팅

그리드 컴퓨팅(Grid Computing)은 지리적으로 분산된 네트워크 환경에 수많은 이기종 컴퓨터와 대용량 저장장치, 데이터베이스 시스템, 인공위성 등과 같은 다양한 자원들을 고속 네트워크로 연결하여 그 자원들을 상호 공유할 수 있도록 하는 차세대 컴퓨팅 환경이다[3].

그리드 컴퓨팅에는 애플리케이션 특징에 따라 크게 4가지로 분류할 수 있다. 데이터 그리드는 대용량의 정보를 생산하는 장비, 대용량 DB 및 고성능 컴퓨터를 사용자 중심으로 연결하는 그리드이며, 액세스 그리드는 계산 그리드와 데이터 그리드에서 생성된 정보를 원거리의 연구자들이 공유할 수 있도록 협업 환경을 제공하는 그리드이다.

계산 그리드는 많은 자원들을 연결하여 계산을 해결할 수 있는 그리드이며, 모바일 그리드는 이동성 개념이 그리드 컴퓨팅에 접목된 그리드이다[4]. 본 논문에서는 CPU, 메모리를 이용하는 액세스 그리드 컴퓨팅을 말한다.

### 2. 위험기반 우선순위관리 시스템

요구사항에 대하여 리스크 노출 값을 파악하고, 이 값을 이용하여 테스트 케이스 우선순위를 결정하는 방법과 관리하는 자동화 시스템을 제안하였다[5]. 장점으로는 요구사항에 대한 리스크 노출 값을 파악할 수 있으며, 리스크 노출 값을 이용하여 테스트 케이스 우선순위를 결정할 수 있었다. 그러나 성능 특성에 대한 테스트 케이스 우선순위 및 자동화 연구와 제한된 시간, 자원, 한정된 비용 안에서 최대한의 결과를 얻어내기 위한 테스트 방법이 미흡하다는 단점이 있다.

### 3. 그리드 컴퓨팅 하이브리드 스케줄링 기법

그리드 컴퓨팅 시스템을 구성하는 전체 노드를 대상으로 작업을 분배하는 메타 스케줄링 정책과 특정 한 개의 노드 내에서 작업을 분배하는 작업 스케줄링을 동시에 고려하는 하이브리드 스케줄링 기법을 제안하였다[1]. 장점으로는 그리드 컴퓨팅 시스템 상의 임의의 노드로 제출된 작업을 실행 시키는데 필요한 프로세서 수와 예상 작업수행 시간에 따라 작업들을 구분하여, 우선순위가 높은 작업은 예약 후에 실행되는 작업 큐(Job Queue)로 분배하고, 우선순위가 낮은 작업과 다른 노드로부터 발송 되어진 원거리 작업은 백필 큐로 할당시킴으로써 시스템의 이용률을 높이면서, 동시에 작업 지연시간을

단축시킬 수 있다. 그러나 임의의 노드에 분배한 후 각 노드에서 작업 스케줄링 병행 처리하는 방법이 미흡하다는 단점이 있다.

#### 4. 그리드 환경에서 워크플로우 서비스를 제공하기 위한 메타 스케줄링 프레임워크

그리드에서 어플리케이션의 사용을 확대시킬 수 있는 메타 스케줄링 프레임 워크를 제안하였다[6]. 장점으로서는 그래픽 기반의 인터페이스를 제공하여 직관적인 해석과 쉽게 이해될 수 있는 그래픽 방식의 표현법을 제공하여 작업에 대한 내용과 처리 절차 등을 자세하게 표현할 수 있는 사용자 작업 환경을 제공하였다. 그러나 OGSA 기반의 그리드 웹 서비스를 지원하기 위한 웹 표준을 위한 시스템의 기능 확장과 새로운 어플리케이션에서 요구되는 서비스에 대응하여 다양한 기능을 제공할 수 있는 서비스가 없는 단점이 있다.

### III. 시스템 효율성 증대를 위한 시스템 구조 및 프로세스

이번 절에서는 시스템 효율성 증대를 위한 그리드 컴퓨팅 기반의 위험 관리 시스템의 프로세스, 시스템 효율성 증대를 위한 그리드 컴퓨팅 기반의 작업 방법론, Agent 스케줄링 및 할당 기법, 위험 정책 등록 및 자원 (Agent) 재배치의 방법 및 수식에 대하여 설명한다.

#### 1. 시스템 효율성 증대를 위한 시스템 프로세스

본 논문에서 제안하는 시스템의 효율성 증대를 위한 그리드 컴퓨팅 기반의 위험관리 시스템의 구성도는 그림 1과 같다.

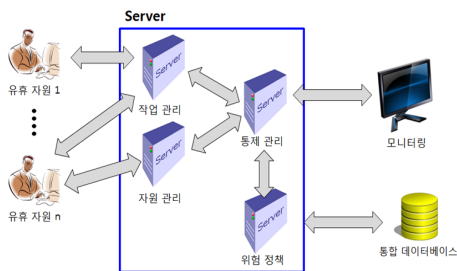


그림 1. 시스템 구성도  
 Fig. 1. System Configuration

그리드 컴퓨팅에서 작업을 등록/수정 할 수 있는 작업 관리 모듈, 유휴자원들을 관리할 수 있는 자원관리 모듈, 작업의 용량 및 작업의 완료시간에 따라 유휴자원의 필요 수, 실행시간 및 스케줄링의 정책을 관리하는 통계관리 모듈, 그리드 컴퓨팅에서 네트워크, 자원의 문제점을 등록할 수 있는 위험정책 관리 모듈로 이루어져있다.

본 논문에서 제안한 시스템 효율성 증대를 위한 그리드 컴퓨팅 기반의 위험 관리 시스템의 순서도는 그림 2와 같다.

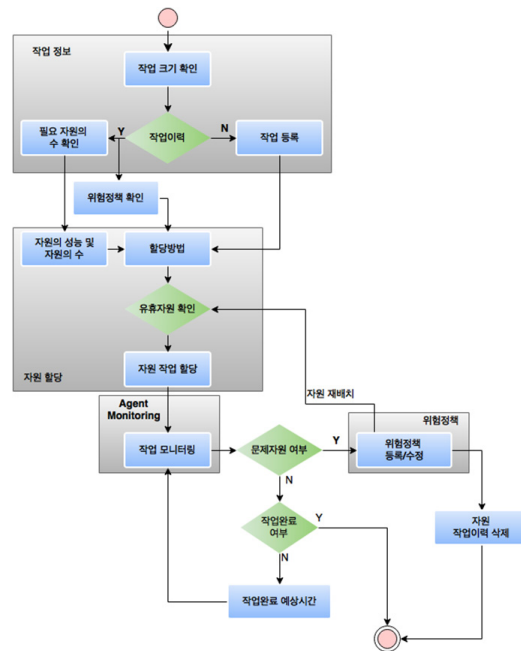


그림 2. 시스템 순서도  
 Fig. 2. System Flowchart

시스템의 진행 순서는 작업을 등록하면 작업의 크기를 확인한 후 기존에 비슷한 작업의 이력을 확인하여 작업을 진행하였던 자원(Agent)의 수, 자원(Agent)의 성능을 확인하고, 위험 정책이 등록되어 있는지 확인한다.

위험 정책을 확인하여 문제가 생겼던 자원(Agent)과 비슷한 성능을 가진 자원(Agent)은 작업 할당 시 할당 방법론에서 제외한 후 다른 자원에게 자원을 할당한다.

작업 할당은 유휴 자원(Agent)들에게 작업을 할당을 하면 자원들은 작업을 진행하면서 작업 진행 데이터베이스에 진행 정보를 실시간으로 저장을 하며, 작업 진행 정보를 작업 모니터링 시스템을 통하여 자원들의 진행 현

황을 확인할 수 있다.

작업 진행시 문제가 생긴 자원(Agent)가 발생한 경우, 위험 정책에서 정보를 추가/수정한 후 문제가 생긴 자원(Agent)에서 작업을 하던 정보는 삭제하며, 유휴 자원들을 확인하여 작업을 재할당하여 작업을 진행할 수 있도록 하였다.

유즈케이스 다이어그램은 사용자 시각에서 소프트웨어 시스템의 범위와 기능을 설명하고 정의한 다이어그램이다. 제안한 시스템의 액터는 유휴 자원(Agent), 관리자(Admin), 모니터링 시스템으로 설정하였으며, 작업관리, 자원관리, 통계관리, 위험정책 모듈을 서브시스템으로 설정하였다. 본 논문에서 제안한 유즈케이스 다이어그램은 그림 3과 같다.

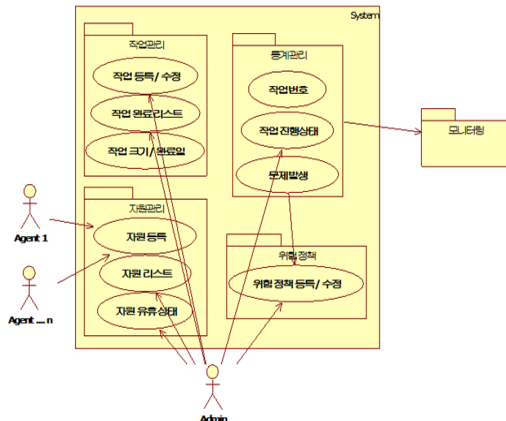


그림 3. 유즈 케이스 다이어그램  
Fig. 3. Usecase Diagram

순차 다이어그램은 다른 객체들 사이에 일어나는 상호작용을 시간 순으로 파악하기 위하여 사용하며, 교류를 주도하는 객체를 왼쪽에 배치하여 메시지들을 시간의 흐름에 따라 위에서 아래로 세로축에 따라 배치하는 다이어그램이다. 사용자가 자원(Agent)을 등록을 하고, 작업관리에서 유휴자원에 작업을 할당하기 전에 유휴자원의 정보를 확인한 후 자원에게 작업을 할당한다. 작업 도중 자원의 시간이 늦어지거나 문제가 발생하였을 시 위험 정책에 문제점 등록, 문제가 되는 사용자(Agent)는 작업의 진행 상태를 파악하여 계속 진행 또는 다른 유휴자원으로 대체를 시킬 수 있다. 본 논문에서 제안한 순차 다이어그램은 그림 4와 같다.

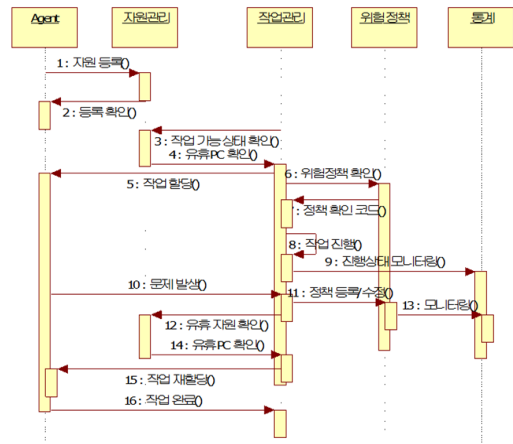


그림 4. 시퀀스 다이어그램  
Fig. 4. Sequence Diagram

## 2. 그리드 컴퓨팅 기반의 작업 방법론

표 1. 그리드 컴퓨팅 기반의 작업 방법론  
Table 1. Grid computing based work methodology

No.	In	내용	Out
1	작업내용	기존 비슷한 작업 여부	이력 여부
2	이력여부(Y)	작업 자원의 수, 작업 시간 확인	필요 자원/총 작업시간
3	이력여부(Y)	위험 정책 확인	등록 여부
4	자원	필요 자원과 비슷한 유휴 자원 확인	자원의 수
5	자원의 수	작업 할당 여부	작업할당
6	자원(Agent)	작업 모니터링	문제 여부
7	문제여부(Y), 위험정책(Y)	위험 정책 추가 및 수정	위험 정책 추가 및 수정
8	문제여부(Y), 위험정책(N)	위험 정책 등록	위험 정책 등록
9	문제여부(Y)	작업 삭제 및 유휴 자원 재배치	삭제 여부, 자원 재배치
10	재배치	자원 재배치	자원 재배치, 완료예상시간
11	완료여부	작업 완료	작업 시간, 사용한 자원 수

본 논문에서 제안한 시스템 효율성 증대를 위한 그리드 컴퓨팅 기반의 위험 관리 시스템에서는 작업 정보를 입력받으면 기존에 비슷한 작업 여부를 확인한 후 작업 이력이 있으면 작업에 참여 자원의 수, 작업 완료 시간, 위험 정책이 등록여부를 확인한다.

위험 정책의 내용을 확인하여 자원에게 작업을 할당할 때 문제가 생겼던 자원의 정보로 유휴 자원의 정보를 확인한다. 작업 시 문제가 생겼던 유휴 자원과 비슷한 정

보를 가지고 있는 자원에게는 작업 할당을 제한하여 작업 중 문제 발생을 최소화하며, 작업 할당된 자원은 실시간 정보를 작업 관리 모니터링을 통하여 진행 여부를 확인할 수 있다. 작업 중 문제가 발생하면 위험 정책에 내용을 등록/수정하고, 문제가 발생한 자원(Agent)에 대하여 작업의 내용을 삭제하고 작업을 하지 않는 유휴 자원에 작업을 재배치한다. 시스템 효율성 증대를 위한 그리드 컴퓨팅 기반의 작업 방법론은 표 1과 같다.

### 3. 자원(Agent) 스케줄링 및 할당 기법

자원(Agent)들의 자원은 CPU, 메모리 등의 하드웨어 장비와 네트워크의 상태에 따라서 작업 시간에 많은 영향을 미친다. 자원(Agent)에 작업 할당하기 전 자원(Agent)의 정보와 작업을 자원(Agent)에 할당 방법, 작업 진행시 문제 발생 시 위험 정책 등록 및 자원(Agent) 재배치 방법과 수식에 대하여 설명한다.

#### 1) 자원 제공 시간

자원 제공 시간은 자원(Agent)이 제공하기로 한 시간을 자원 총 제공 시간(AT<sub>T</sub>), 작업에 참여하는 시간(AT<sub>S</sub>), 작업에 참여하지 않는 시간(AT<sub>NS</sub>)으로 변수를 정의하였으며, 자원 제공 시간을 측정하기 위해서 수식 (1)과 같이 함수를 정의하였다.

$$AT_t = AT_s + AT_{ns} \quad (1)$$

AT<sub>T</sub>(Agent Time Total) : 자원 총 제공 시간  
 AT<sub>S</sub>(Agent Time Spend) : 자원 할애 시간  
 AT<sub>NS</sub>(Agent Time Not Spend) : 자원 미제공 시간

#### 2) 자원의 가용성

자원의 가용성은 자원(Agent)이 작업을 정상적으로 진행한 시간(AT<sub>S</sub>)과 자원의 할애를 하지 못하는 자원 고장의 시간(AT<sub>F</sub>)으로 정의를 하였으며, 자원의 고장 시간 으로서는 자원의 CPU, 메모리를 작업에 할당하지 못하는 시간을 의미한다. 자원의 가용성을 측정하기 위해서 수식 (2)와 같이 함수를 정의하였다.

$$A_a = \sum_{i=S}^E AT_s - \sum_{i=S}^E AT_f \quad (2)$$

A<sub>a</sub>(Agent Availability) : 자원 가용 시간  
 AT<sub>S</sub>(Agent Time Spend) : 자원 할애 시간  
 AT<sub>F</sub>(Agent Time Fail) : 자원 고장 시간

#### 3) 자원 할애 예상 시간

자원 할애 예상 시간(A<sub>et</sub>)은 자원의 자원을 할애하면서 작업을 완료하기까지 실제로 작업에 참여하는 예상 시간을 의미하며, 자원 할애 예상 시간을 측정하기 위해서 수식 (3)과 같이 함수를 정의하였다.

$$A_{et} = AT_t \times A_a \quad (3)$$

A<sub>et</sub>(Agent Expectation Time) : 자원할애 예상시간

#### 4) 자원(Agent) 할당 기법

자원(Agent) 할당 기법은 작업을 자원들에게 할당하기 전에 작업을 안전하게 완료될 수 있도록 하기 위하여 자원들의 정보를 확인 후 작업을 자원에게 할당하는 기법으로 총 작업 예상시간(TJ<sub>et</sub>)과 자원 할애 예상시간(A<sub>et</sub>)을 확인하여 1보다 큰 값이 나오는 자원들에게 우선 할당하는 하도록 정의하였으며, 자원(Agent) 할당 수식은 (4)와 같다.

$$IR_{A} = \left( \frac{A_{et}}{TJ_{et}} \times 100 \right) > 1 \quad (4)$$

IR<sub>A</sub>(Idle Resource Agent) : 유휴 자원 에이전트

TJ<sub>et</sub>(Total Job Estimated Time) : 총 작업 예상 시간

A<sub>et</sub>(Agent Expectation Time) : 자원 할애 예상 시간

### 4. 자원(Agent) 모니터링

자원(Agent)들의 작업 중 실시간으로 작업 진행 현황 정보를 저장하며, 자원들의 사용되는 CPU, 메모리 등 하드웨어의 정보를 저장하여 실시간으로 모니터링을 할 수 있으며, 모니터링을 하면서 문제가 발생하는 자원(Agent)을 확인할 수 있으며, 일시적인 문제인지 작업 진행이 불가능한지 파악할 수 있다.

### 5. 위험 정책 등록 및 자원(Agent) 재배치

자원(Agent)들의 작업 중 문제가 발생하는 자원들의 이벤트 발생내용, CPU, 메모리의 사용 정보를 위험 관리에 등록하며, 자원(Agent)의 진행여부를 확인하여 자원 재할당 여부를 확인하고, 자원의 작업 불능상태(대기시간 증가)로 판단되면 자원(Agent)에서 작업 중이던 내용은 삭제 후 유휴 자원(Agent)에 재할당하도록 정의하였다.

### IV. 실험 및 결과

본 논문에서 제안한 시스템 효율성 증대를 위한 그리드 컴퓨팅 기반의 위험 관리 시스템의 검증에 위하여 작업의 크기는 3GB로 제한하였으며, 네트워크의 속도에 따라 작업의 시간이 비례하므로 자원(Agent)의 설치 범위를 사내 네트워크로 제한하였으며, 사용가능한 자원(Agent)은 50대로 한정하여 동일한 작업을 20회에 걸쳐 실험 하였다.

첫 번째 실험은 자원의 수를 최대 40대로 지정하였으며, 문제가 생길 자원(Agent)을 유휴 자원으로 재할당 가능하게 하였다. 자원(Agent)의 수를 점차 증가시키고, 기존 그리드와 문제가 되는 Agent를 제외하고 다른 유휴자원으로 재할당 후 소요되는 시간을 측정하였다. 기존 그리드에서 작업을 할 경우 자원(Agent)의 수가 증가할수록 작업의 시간이 증가하였으며, 소요되는 시간의 증가 원인은 각 자원들에게 작업을 할당을 하였으나 자원(Agent)의 CPU, 메모리가 다른 작업으로 사용을 하지 못하여 소요되는 시간이 증가되었다.

자원(Agent)의 문제가 생기는 부분에 대하여 작업을 재할당하기 위해서 수식 (5)를 통해 산출하였다.

$$MJ_A = A_\alpha < \frac{(TJ_{et} - A_{et})}{TJ_{ac}} \quad (5)$$

- MJ<sub>A</sub>(Move Job Agent) : 유휴자원 에이전트 이동
- A<sub>α</sub>(Agent Availability) : 자원 가용 시간
- TJ<sub>et</sub>(Total Job Estimated Time) : 총 작업 예상 시간
- A<sub>et</sub>(Agent Expectation Time) : 자원 할애 예상 시간
- TJ<sub>ac</sub>(Total Job Agent Count) : 투입된 에이전트의 수

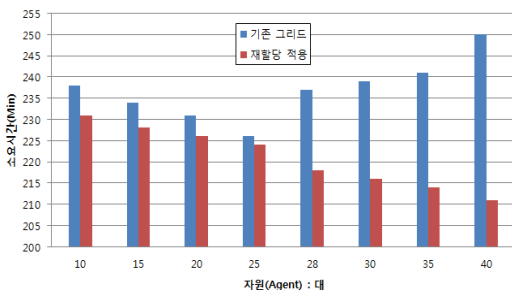


그림 5. 자원(Agent) 수에 따른 비교  
Fig. 5. Comparison in accordance with the number resource(Agent)

자원(Agent) 재할당 알고리즘을 적용하였을 경우, 문제의 자원(Agent)의 작업을 제외하고, 유휴 자원에 재배치함으로써 소요되는 시간이 줄어드는 것을 확인할 수 있었다. 자원(Agent) 수에 따른 비교는 그림 5와 같다.

두 번째 실험은 작업을 진행함에 있어 위험 정책이 등록되어 있는지 확인 후 위험 정책이 등록되어져 있을 때와 위험 정책이 등록되지 않았을 때 소요되는 시간을 비교하였다.

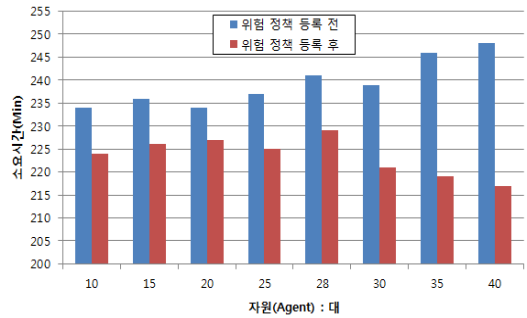


그림 6. 위험 정책 등록 전/후 비교  
Fig. 6. Risk policy registration before/after comparison

위험 정책이 등록이 되어져 있을 경우 문제가 생길 수 있는 자원(Agent)을 제외한 후 유휴 자원(Agent)에 작업을 할당하여 작업의 소요되는 시간이 줄어드는 것을 확인할 수 있었다. 위험 정책 등록 전/후의 비교는 그림 6과 같다.

이 결과는 시스템 효율성 증대를 위한 자원(Agent) 재할당 기법과 위험 관리 정책 등록 후 그리드 기반의 작업을 진행하였을 경우 빠르게 작업을 처리할 수 있음을 보여준다.

### V. 결론

본 연구에서는 시스템 효율성 증대를 위한 그리드 컴퓨팅 기반의 위험 관리 시스템을 제안하였다. 기존의 그리드 컴퓨팅은 제한된 자원 안에서 최대한의 결과를 얻어 내기 위한 프로세스, 실시간 작업의 총 실행시간을 정확하게 예측할 수 없는 문제점을 가지고 있다. 본 논문에서는 시스템 효율성 증대를 위한 그리드 컴퓨팅 기반의

위험관리 시스템을 제안하였다.

기존의 시스템과 제안한 시스템을 비교하기 위하여 위험 정책 모듈, 자원(Agent) 제한당 모듈을 추가하여 자원(Agent)의 수에 따라 작업에 소요되는 시간을 비교하였다. 기존의 그리드 컴퓨팅보다 본 논문에서 제안한 위험 정책, 문제의 자원(Agent)을 제외하고 유휴 자원에 할당하여 소요시간을 감소시킴을 확인하였다.

향후 연구과제로는 네트워크의 속도로 인하여 사내로 제한하였지만 네트워크의 제한을 하지 않고 기존의 그리드 컴퓨팅과 제안한 시스템과 성능을 비교할 예정이다.

## References

- [1] Chang-Hoom Kang, Chang-Yeol Choi, Kie-Jin Park, Sung-Soo Kim, "Multi-queue Hybrid Job Scheduling Mechanism in Grid Computing", Korea Information Science Society, Journal of KIISE : Computer Systems and Theory, Vol. 34, No. 7/8, pp.304-318, 2007.
- [2] Jae-Kwon Kim, Jong-Sik Lee, "Dynamic Available-Resource Reallocation based Job Scheduling Model in Grid Computing", The Korea Society For Simulation, Journal of the Korea Society for Simulation, Vol. 21, No. 2, pp.59-67, 2012.
- [3] J. Y. Choe, W. J. Lee, and C. H. Jeon, "Real-Time Job Scheduling Strategy for Grid Computing," The Korea Society of Computer and Information, Vol. 15, No. 2, pp. 1-8, 2010.
- [4] Su-Geun Oh, Si-Gwan Kim, "Design and Implementation of Multi Type Grid Computing System Using Dynamic Module Loading Technique", Korean Institute of Information Technology, Journal of Korea Institute of Information Technology, Vol. 9, No. 5, pp.111-118, 2011.
- [5] Eun-Young Lee, Mi-So Yoon, Byoung-Ju Choi, "A Risk-based Test Case Priority Management System", Korea Information Science Society, Journal of KIISE : Computing Practices and Letters, Vol. 18, No. 6, pp.439-449, 2012.
- [6] Seog-Chan Hwang, Jae-Young Choi, "A Meta Scheduling Framework for Workflow Service on the Grid", Korea Information Science Society, Journal of KIISE : Computing Practices and Letters, Vol. 10, No. 5, pp.375-384, 2004.
- [7] Eun-Ser Lee, "A Study for Process Improvement by State Analysis of Risk Items", Korea Information Processing Society, Journal of Information Processing Systems, Vol. 15-D, No. 4, pp.523-530, 2008.
- [8] Sol-Ji Kim, Tae-Ho Kim, Hong-Chul Lee, "Multi-Objective Job Scheduling Model Based on NSGA-II for Grid Computing", The Korea Society of Computer and Information, Journal of the Korea Society of Computer and Information, Vol. 16, No. 7, pp.13-23, 2011.
- [9] Jung-In Koh, Hye-Jeong Kang, Yoon-Hee Kim, "Adaptive Policy-based Task Scheduling for Scientific Applications in Hybrid Cloud", Korea Information Science Society, Journal of KIISE : Computing Practices and Letters, Vol. 19, No. 11, pp.572-579, 2013.
- [10] Yean-Woo Jung, Seong-Soo Cho, Jong-Young Lee, Kye-Dong Jeong, "A Design of P2P Cloud System Using The Super P2P", The Institute of Internet, Broadcasting and Communication, International Journal of Internet, Broadcasting and Communication, Vol. 7, No. 1, pp.42-48, 2015.

## 저자 소개

### 정 재 훈(준회원)



- 2010년 2월 : 한국산업기술대학교 컴퓨터공학과(공학사)
- 2012년 2월 : 한국산업기술대학교 IT 융합학과(공학석사)
- 2012년 3월 ~ : 한국산업기술대학교 컴퓨터공학과 박사재학

<주관심분야 : 소프트웨어공학, 정보통신시스템, 객체지향 분석 및 설계>

**김 신 령(정회원)**



- 1983년 2월 : 경북대학교 전자공학과 (공학사)
- 1985년 2월 : 연세대학교 본대학원 전자공학과(공학석사)
- 1990년 2월 : 연세대학교 본대학원 전자공학과(공학박사)
- 1992년 2월 ~ : 동서울대학교 정보통신과 부교수

<주관심분야 : 정보통신시스템, 부호화 방식>

**김 영 곤(정회원)**



- 1983년 2월 : 경북대학교 전자공학과 (공학사)
- 1985년 2월 : 연세대학교 본대학원 전자공학과(공학석사)
- 2000년 2월 : 한국과학기술원 전산학과(공학박사)
- 1985년 ~ 2007년 : KT 수석연구원

• 2007년 ~ : 한국산업기술대학교 컴퓨터공학과 부교수  
<주관심분야 : 소프트웨어공학, 정보통신시스템, 객체지향 분석 및 설계>