

## PC 기반 그리드 환경에서 협업 지원을 위한 어플리케이션 서비스 시스템에 관한 연구

허혁<sup>o</sup> 문석재 엄영현 국윤규 정계동 최영근  
광운대학교 컴퓨터과학과

{korhaejuk<sup>o</sup>, msj1568, class76, ykkook, gdchung, ykchoi}@kw.ac.kr

### A Study on Application Service System for Collaboration in PC Grid Environment

Hyuk Heo<sup>o</sup> Seok-Jae Moon, Eum Young-Hyun, Kook Yoon-Gyu, Gae-Dong Jung, Young-Geun Choi  
Department of Computer Science Kwangwoon University

#### 요 약

그리드 환경에서 어플리케이션 콘텐츠는 특정한 문제 해결을 위한 실질적인 작업이며, 그리드 자원에 분산 배포 되고 서로 연관되어 실행되기 때문에 협업 환경을 구성하는 것은 매우 중요한 부분이다. 그리고 대부분의 협업은 워크플로우를 통하여 구체화 되고, 그리드 환경에서 협업 처리를 위한 미들웨어로는 Globus toolkit이 대표적이다. 하지만 이 미들웨어는 그리드 환경 구축을 위한 기본 서비스들만을 제공하고, 협업을 할 수 있는 워크플로우 생성, 작업 스케줄링, 어플리케이션 콘텐츠 관리 같은 부분은 적용 되지 않는다. 또한 Globus Toolkit은 대형화된 그리드 커뮤니티 구성에는 적합하나, 소규모 PC 기반에는 부적합하다. 따라서 본 논문에서는 PC 기반 그리드 환경에서 소규모 협업 처리에 효율적인 어플리케이션 서비스 시스템을 제안한다. 어플리케이션 콘텐츠 배포, PC기반으로 구성된 자원관리등의 역할을 통해 효율적인 협업 환경을 지원한다. 또한 워크플로우는 어플리케이션 콘텐츠 연관관계 사전을 만들어 협업에 필요한 어플리케이션 콘텐츠간의 정보 및 연관 관계를 표현한다.

#### 1. 서 론

그리드 컴퓨팅은 지리적으로 분산되어있는 컴퓨터, 저장 장치 그리고 실행 장비 등의 자원을 광역 네트워크로 연결하여 상호 운영을 하거나 PSE(Problem Solving Environment) 즉 특정한 문제해결을 위하여 물리적인 위치에 구애 받지 않고 자원 교류와 협업 환경을 구성하는 것이다.[1] 그리드 환경에서 수행되는 작업은 대부분이 여러의 작업으로 구성된 복잡한 작업들이다. 워크플로우를 이용하면 이러한 작업들을 효율적으로 처리할 수 있다. 최근에는 그리드 환경에서 워크플로우 기술을 적용하여 복잡한 작업들을 관리하기 위한 연구들이 활발하게 진행되고 있다. 대표적인 예로는 Pegasus[3], GridFlow[4] 등이 있다. GridFlow는 다양한 도메인을 통합하여 수시로 자원상태가 변동하는 동적인 그리드 환경에서 시간에 민감한 응용프로그램들을 스케줄링 하는데 초점을 두고 있다. 그리고 효율적인 스케줄링을 위해 GridFlow는 퍼지 타이밍 기술과 응용프로그램의 수행시간 예측기술을 사용한다. pegasus 등의 워크플로우 시스템은 그

리드 환경에서 효율적인 워크플로우 관리 기능을 제공하지만 서비스 개념을 제공하지 않는다. 따라서 워크플로우 만으로는 작업을 실행하고 결과물을 얻을 수 없다. 또한 사용자가 작성한 추상화된 워크플로우를 바탕으로 물리적으로 저장되어있거나 사용자가 작성한 작업과 매핑하여 배포 할 수 있는 기반을 만들어야 한다. 이러한 작업을 어플리케이션 콘텐츠[5]라 하며, 그리드 환경에서 실행 하기 위하여 등록되어있는 어플리케이션 콘텐츠를 사용자가 보다 효율적으로 검색하여 워크플로우를 작성하기 위해서는 어플리케이션 콘텐츠의 정보 더불어 어플리케이션 콘텐츠간의 연관성이 함께 명시 되어있어야 한다. 어플리케이션 콘텐츠를 배포하여 실행 하기 위해서는 자원들을 하나의 VO(Virtual Organization)로 구성 할 수 있는 미들웨어가 필요하다. 현재 그리드를 구성하기 위한 미들웨어로 미국의 아르곤 연구소를 추축으로한 연구팀은 Globus Toolkit[2]를 제안하였다. 글로버서는 이 기종 자원의 단일한 접근 방법을 제공 하고 필요한 자원의 검색을 위해 디렉토리 서비스를 제공한다. 이 밖에도

원격지 데이터 전송 방법, 사용자의 인증과 보안 방법들을 제공한다. 글로버스는 그리드 자원들에 대한 단일화된 접근 방법을 제공하지만 상위 계층의 그리드 어플리케이션이 요구하는 다양한 응용 서비스를 모두 충족시킬 수는 없다. 또한 글로버스 미들웨어는 휘발성이 강한 PC를 기반으로 하는 그리드 환경을 구성하는 데에는 적합하지 않다. PC를 기반으로 하는 그리드 환경을 구성하기 위해서는 PC에 부담을 주어서는 안되며, 협업을 하기 위한 필요한 정보와 어플리케이션 콘텐츠를 실행 할 수 있는 메커니즘, 데이터를 전송 할 수 있는 등 작업처리를 하기위하여 최대한 단순화 해야하며, 장애에 대하여 유연하게 대처 할 수 있는 방법이 필요하다. 현재의 PC 기반의 그리드 시스템으로 가장대표적 이라 할 수 있는 SETI@Home[8], Korea@Home[9] 등의 @Home 프로젝트 들이 있다. 이들은 대규모 작업을 처리하며, 동일 프로세스에 상이한 데이터들을 분산시켜 실행 결과값을 취합 한다. 하지만 상이한 다수의 소규모 분산 작업을 처리하기 위해서는 이러한 방식은 적합 하지 않다. 본 논문에서는 PC 기반의 그리드 환경을 구성하고 다수의 분산 작업을 배포하고 협업 처리 할 수 있는 어플리케이션 서비스 시스템을 제안한다. 어플리케이션 서비스 시스템은 어플리케이션 콘텐츠 관리, PC기반 그리드 자원 관리, 자원이라 할 수 있는 노드(PC) 3가지로 구성 된다. 어플리케이션 관리 서비스는 사용자 또는 어플리케이션 등록자가 어플리케이션 콘텐츠를 콘텐츠 풀에 저장하고 이를 작업의 한 단위로 하여 배포하게 되며, 배포하기 위한 스케줄링은 노드로 부터 얻은 정보에 의하여 스케줄링 하게 된다. 어플리케이션 콘텐츠에 대한 정보, 연관관계, 저장위치의 정보 자원들을 상호 연관성에 따라 사전을 구성하고 이를 효율적으로 검색 관리 할 수 있도록 한다. 그리고 실질적인 작업을 실행하고 결과를 만들어 내는 자원제공자인 노드는 어플리케이션 콘텐츠 서버로부터 어플리케이션 콘텐츠와 실행에 필요한 데이터 그리고 워크플로우 명세를 받아 이를 분석하고 실행 하여 결과를 만들어내기 때문에 노드에 대한 부담을 줄이고 장애에 대하여 신속한 대응을 할 수 있으며, 노드간의 연결 방식을 Hybrid-P2P 방식을 사용하여 어플리케이션 배포 및 관리의 부담을 줄일 수 있다.

2. 관련연구

그리드 환경과 PC기반 그리드의 가장 큰 차이점은 자원의 가용성이라 할 수 있다. 대부분의 그리드 환경은 순수연구 목적을 목적으로 하기 때문에 계산 작업 이외에 기타 작업에서는 자원 사용이 거의 없으나 PC기반 그리드는 기본적으로 사용자의 업무처리를 위해 자원이 사용되다가 자원의 유휴 상태에 따라 제공된다는 것이

다. 또한 그리드 환경에서 그리드 어플리케이션이 표현되는 작업의 흐름도 차이가 있다. 크게 본다면 그림 1과 같이 3가지 형태로 나누어 볼 수 있다.

병렬적 워크플로우의 경우 각 작업의 결과에 영향을 받지 않기 때문에 작업 완료시간은 문제가 되지 않는다. 그러나 순차적 워크프로우나 네트워크적 워크플로우는 다른 작업의 결과를 바탕으로 실행 되어야 하기 때문에 이전 작업이 완료되지 않으면 작업을 진행 할 수 없다. 따라서 PC 기반의 그리드의 경우 병렬적 워크플로우 형태의 작업을 대상으로 한다.

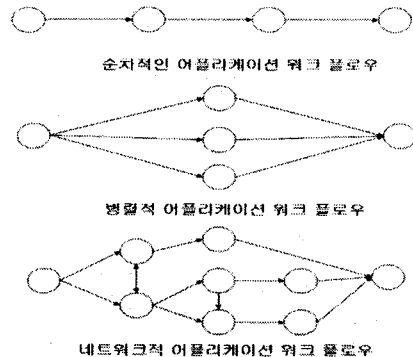


그림 1 작업 흐름

3. 협업 지원을 위한 어플리케이션 서비스 시스템

3.1 시스템 개요

본 논문에서 제안하는 어플리케이션 서비스 시스템 모델은 그림 2와 같다. 구성은 크게 4가지로 나누어 볼 수 있다. 작업을 선택하고, 선택된 작업의 순서를 정하며, 작업에 대한 결과를 전달 받을 클라이언트가 있다. 그리고 어플리케이션 콘텐츠의 등록, 삭제, 배포, 스케줄링 및 워크플로우 생성을 할 수 있는 어플리케이션 파트와 PC로 구성된 그리드 자원을 관리, 감독 그리고 정보 수집을 하는 그리드 자원관리 파트 그리고 실질적인 어플리케이션 콘텐츠를 실행하는 그리드 자원 파트가 있다.

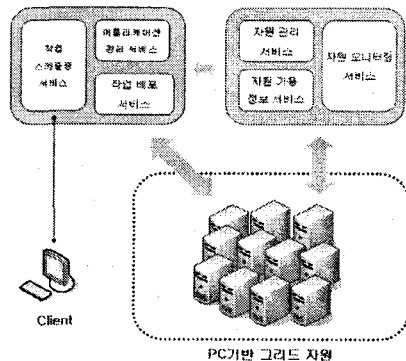


그림 2 제안하는 PC 기반 그리드



작업이 필요하다. 따라서 노드 정보 제공자로부터 노드의 유휴 자원 정보, 노드의 위치 정보 그리고 자원 정보를 제공 받아 작업 스케줄링을 만들게 된다. 또한 워크플로우 생성기에서 어플리케이션 콘텐츠 간의 정보 흐름이 만들어 지면 이것이 워크플로우 사양이 담긴 문서가 된다. 이것은 협업을 하기 위한 모든 노드들에게 어플리케이션 콘텐츠와 함께 배포되어 협업 환경을 구성하게 된다.

### 3.2.4 그리드 구성 계층

그리드 구성 계층은 그리드를 구성하고 있는 자원들에 대한 상태 및 정보를 관리하는 역할을 한다. 그리드를 관리 하는데 가장 필요한 요소는 그리드를 이루고 있는 자원 즉 표2 과 같은 성능에 대한 정보를 수집 하고 관리하여, 이러한 정보가 필요한 곳에 정보를 제공 한다. 이러한 기능을 그리드 자원관리 서비스와 그리드 가용정보 서비스에서 나누어 처리한다. 이러한 정보는 어플리케이션 서비스 서버에서 요청하면 자원제공자는 자신의 노드 정보를 수집하여 전달한다.

표 2 노드 정보 항목

|        |   |
|--------|---|
| CPU    | 해당노드의 CPU정보를 가져온다.<br>Ex) intel p4 1.8G, AMD Athlon 64 3500+  |
| 메모리    | 해당 노드의 물리적으로 설치된 메모리의 용량을 가져온다.<br>Ex) 256MB, 512MB, 1024MB   |
| 디스크 용량 | 노드 이름에 따라 설치되어 어플리케이션이 실행되는 하드디스크의 용량을 가져온다.  |
| 작업ID   | 현재 실행 중이거나 또는 JOB 디렉토리에 있는 작업들의 ID정보를 가져온다.   |
| 가용수치   | CPU 사용률과 3분간의 프로세스 평균값과 메모리 사용률을 얻어 표현 한다.<br>가용값 = CPU% * ((TotalMemory - useMemory) / TotalMemory)<br>Ex) 27% * ((512-32) / 512) * 100 = 63.814 |
| OS     | 해당 노드에서 사용중인 노드의 OS정보를 가져온다.<br>Ex) Windows XP Pro Service Pack 4<br>Linux Fedora Core  |

## 4. 노드의 구조 및 동작 과정

### 4.1 노드의 구조

노드의 구조는 아래 그림 6과 같은 소프트웨어 구조를 가지고 있다. 어플리케이션 콘텐츠를 실행하고 결과를 정해진 곳으로 전달하며, 자신의 상태 정보와 가용정보 등을 제공하는 것이 노드이다.

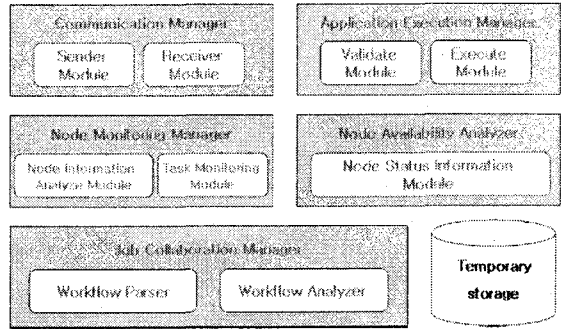


그림 6 노드의 소프트웨어 구조

노드가 하는 가장 큰 역할은 워크프로우중에 하나의 구성요소로 동작하는 것이며 해당 어플리케이션 콘텐츠를 실행 하는 것이다. 이러한 실행 환경이 구성 되려면 어플리케이션 콘텐츠를 받는 부분과 결과를 보내는 부분이 필요하다. 또 받은 어플리케이션에 대하여 검증하는 부분도 필요하다. 그리고 그리드 자원으로써 자기 자신의 가용성 정보, 노드가 가지고 있는 성능 정보를 그리드 관리서비스에게 전달해야 한다. 따라서 그림 6 과 같은 협업 자원을 하기 위한 노드의 소프트웨어 구조를 가지게 된다. 어플리케이션 콘텐츠의 무결성은 부여된 ID 정보와 파일 크기로 검증한다. 자원 제공자는 어플리케이션 서비스 서버에서 작업 스케줄링 을 할 수 있는 기본 자료를 제공해야 한다. 따라서 자원제공자인 노드는 현재 자기 자신의 CPU, 메모리, 스토리지 등의 정적 자원 정보와 현재 작업 중인 프로세스, 타스크, 현재 자원 가용정보의 동적 자원 정보를 제공한다. 이러한 두 가지 자원 정보를 제공함으로써 동적인 상황에 대처하여 자원 스케줄링과 작업 스케줄링을 할 수 있다.

### 4.2 노드의 동작 메커니즘

본 시스템의 기본적인 흐름은 그림7과 같으며, Hybrid-P2P의 메커니즘을 사용 한다. 워크플로우 명세와 어플리케이션 콘텐츠는 어플리케이션 서비스 서버에서 할당 받고 그 이후 작업은 노드들 간의 협업에 의하여 이루어진다. 노드간 에는 어플리케이션 콘텐츠와 작업 결과를 주고받는다.

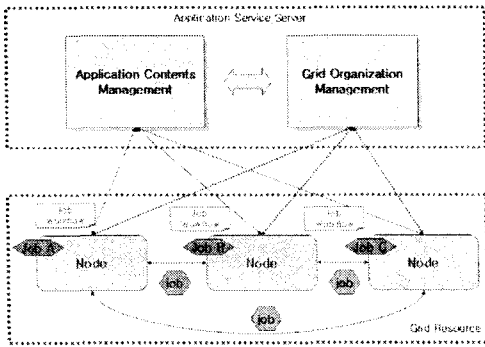


그림 7 노드 협업 흐름도

노드는 할당 받은 어플리케이션 콘텐츠를 사용 후 바로 폐기 하는 것이 아니라 추후 재사용을 하기 위하여 일정량을 보관하게 된다. 일정량을 초과 하게 되면 최초 할당 받은 순서대로 폐기 한다. 때문에 최초 할당하는 어플리케이션 콘텐츠는 서비스 서버에서 할당 시키지만 한번 이상 배포된 어플리케이션 콘텐츠는 다른 노드에서의 어플리케이션 콘텐츠를 복제 받는다. 어플리케이션 콘텐츠에 대한 정보는 서비스 서버의 지식베이스에서 정보를 제공받으며, 어플리케이션 콘텐츠의 식별은 ID 정보를 가지고 한다. 협업이 시작되면 2개 이상의 어플리케이션 콘텐츠가 연관 되어 실행 되며, 워크플로우의 순서에따라 실행 된다. 이때 순차와 병렬적 흐름이 혼합된 작업 처리중에 장애가 발생되면 전달하려는 노드의 어플리케이션 콘텐츠와 같은 노드를 서비스 서버로부터 검색하여 워크플로우 명세서의 해당 부분을 수정하여 작업을 계속 한다. 만약 그런 어플리케이션 콘텐츠를 가진 노드가 없다면 가용성이 가장 좋은 노드를 찾아 어플리케이션 콘텐츠를 할당하고 워크플로우 명세서를 수정 후 계속 진행 한다.

### 5. 결론

본 논문에서는 PC 기반 그리드 환경에서 소규모 협업 처리를 위한 효율적인 어플리케이션 서비스 시스템을 제시하였다. 이러한 방법으로 인하여 사용자는 추상적인 간단한 방법으로 그리드 자원을 사용하여 처리하고자 하는 작업을 수행 할 수 있다. 또한 본 시스템을 사용함으로써 어플리케이션에 대한 관리능력이 향상 되었고 자동적으로 어플리케이션 콘텐츠가 배포가능 하게 되었다. 이 시스템을 기반으로 하여 소규모 협업 작업을 용이하게 처리 할 수 있을 것으로 기대 된다.

향후 연구로는 지능적으로 어플리케이션 콘텐츠를 검색할 수 있는 검색 시스템이 필요하며, 보안적인 측면에서 어플리케이션 콘텐츠의 검증부분이 강화 되어야 할 것이다.

### 6. 참고문헌

- [1] I. Foster, C. Kesselman, S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations." International J. Supercomputer Application. 15(3). 2001.
- [2] Globus Project <http://www.globus.org>
- [3] Y.Gill, C Kesselman, G.Mehta, S.Patil, M. Su, K. Cahi M. Livny, E. Deelman and J Blythe, "Pagasus Mapping Scientific Workflows onto the Grid." Across Grids Conference 2004
- [4]J. Cao, S. A Jarvis, S. Saini and G. R. Nadd, "GridFlow: WorkFlow Management for Grid Computing." 3rd International Symposium on Cluster Computing and the Grid,pp 12-15, 2003
- [5]Application Contents Service WG (ACS-WG) <http://forge.gridforum.org/projects/acs-wg>
- [6] Michel Biezunski, Martin Bryan, Steve Newcomb, ISO/IEC 13250 TopicMap
- [7]This specification describes version 1.0 of XML Topic Maps (XTM) 1.0 <http://www.topicmaps.org/xtm/1.0/>
- [8] SETI@home. <http://setiathome.ssl.berkeley.edu>
- [9] Korea@home <http://www.koreaathome.org>