

GRAM 작업 관리용 확장형 실행환경 정보 서비스 시스템 설계

정미숙*, 장경일*, 김완규**, 박규석*
*경남대학교 컴퓨터공학과
**진주산업대학교 컴퓨터공학과

A Design of Extensible Execution Environment Information Service System for GRAM Job Managing

Mi-Suk Jeong*, Kyung-Il Jang*, Wan-Kyu Kim**, Kyoo Seok Park*
*Dept. of Computer Engineering, KyungNam University
**Dept. of Computer Engineering, Chinju National University

요 약

그리드는 컴퓨터류만 연결된 기존의 인터넷에 모든 종류의 비컴퓨터류의 기계장치들을 인터넷 상에 연결시켜 컴퓨터와 더불어 활용하는 기술을 지칭한다. 즉, 지역적으로 분산되어 있는 컴퓨팅 파워를 공유하여 마치 한 대의 고성능 컴퓨터처럼 사용할 수 있게 해 주는 것을 말한다. 이러한 그리드환경에서 client가 요구하는 작업을 명시하는 표준 RSL(Resource Specification Language)을 이용해 GRAM (Globus Resource Allocation and Management)에 작업을 제출한다. 하지만 기존의 GRAM에서는 특별한 수행 환경을 요구하는 작업을 수행시키는 방법을 알지 못한다.

이에 본 논문에서는 브로커와 상호작용가능한 실행환경정보(Excution Environment Information)를 제안함으로써, GRAM의 특수 수행 환경을 지원 할 수 있게 하였다.

1. 서론

그리드란 지역적으로 분산된 컴퓨터, 대용량 스토리지, 거대 실험장치 등과 같은 고성능 자원들을 고성능 네트워크로 연동해 사용자가 단일 시스템처럼 그리드에 연결된 모든 자원들을 쉽게 사용할 수 있도록 하는 정보통신 인프라이며 첨단 컴퓨터 통합시스템이다.[1] 그리드는 클라이언트 측면에서 보면 유틸리티 컴퓨팅이고, 서버 측면에서 살펴보면 자원 할당, 정보 공유 그리고 고가용성과 관련된 개념이다. 자원 할당은 자원을 요청하고 필요로 하는, 누구나 원하는 것을 얻을 수 있도록 하는 것이다.

그리드 컴퓨팅 시스템은 작업의 명세화, 관리 그리고 원격 수행을 위한 메카니즘이다. 이러한 서비스를 위해 현재 글로버스 툴킷[1]이 업계의 표준으로 자리 잡고 있다. 현재 글로버스는 사용자가 요구한 작업을 MDS(Meta Directory Service)를 이용하여 현재 이용할 수 있는 자원을 파악하고, 이 MDS를 이용하여 요청된 RSL(Resource Specification Language)로 쓰여진 자원을 DUROC(Dynamically Updated Request

Online Co-allocator)을 통해 개별 컴퓨팅 리소스에 존재하는 GRAM(Globus Resource Allocation and Management)에 세부적인 RSL로 요구하게 되면, GRAM은 허용되는 수준에서 할당된 자원을 제공하게 된다. 그러나 현재 GRAM은 Java 가상 머신 내에 실행되는 Java 응용프로그램과 같은 특별한 수행 환경이 요구되는 응용프로그램에 대한 지원이 결여되어 있다[8].

본 논문은 위에서 말한 GRAM의 문제점을 해결하기 위해 실행 환경에 대한 정보를 가지는 EEI(Excution Environment Information)를 제안한다. 2장에서는 본 논문에 필요한 관련 연구를 살펴봄, 3장에서 현재 그리드 미들웨어의 표준으로 자리잡고 있는 Globus Toolkit의 구성요소 및 제안하는 EEI(Excution Environment Information)를 이용한 실행 환경 지원 메카니즘을 보여준다. 마지막으로 4장에서는 결론 및 향후연구에 대해 기술한다.

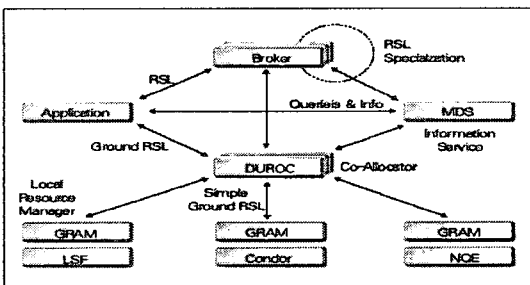
2. 관련 연구

그리드는 지리적으로 그리고 조직 면에서 분산되어 있는 이종의 자원들을 서로 연결시켜 하나의 가상 컴퓨터처럼 동작하는 체제이다. 고성능 슈퍼컴퓨터에서 개인용 컴퓨터까지 광범위하게 이용할 수 있는 컴퓨터 자원과 수십기에서 수백테라까지 이용할 수 있는 저장장치, 세계 각지의 연구인력 및 다양한 통신 시스템등 현재 분산된 자원들을 하나의 동일한 인터페이스를 통해 접근하도록 하여 체계적으로 협업시스템을 만들고 고성능의 계산 능력과 데이터 처리를 가능하게 할 수 있는 기술이 그리드이다.

2.1 그리드 미들웨어

현재 주로 사용되는 그리드 미들웨어 중에는 인터넷 상에서 큰 업무를 작게 나누어 처리할 경우에 필요한 사용자 서비스 미들웨어, 작업을 스케줄링하거나 자원을 선정하는 등의 응용 미들웨어, 보안 및 자원 할당을 수행하는 인프라들을 중심으로 구성되어 있는 글로벌스가 있고[1], 글로벌스의 상위에 위치하며 자원 사이의 동적 부하분산을 포함한 스케줄러로서, 사용자 요구와 자원의 매칭을 시도하는 Condor-G와 과금정보와 작업의 종료기한 등을 조건으로 하여 자원검색과 스케줄링을 행하는 Nimrod/G 등이 있다[3].

독자적인 과제별로 진행되던 이들 미들웨어 역시 공통의 인터페이스를 통해 각 개별 그리드 환경을 포괄하는 한단계 높은 새로운 그리드 환경에서 서로 연동하기 위한 표준화 작업이 현재 GGF, 컴퓨터 포럼, 유럽 그리드 포럼, 한국 그리드 포럼 등의 기관을 중심으로 진행중이다.



(그림 1) 글로벌스 자원관리 아키텍처

그림1은 글로벌스 자원 관리 아키텍처를 설명하고 있다. 글로벌스 자원 관리 아키텍처는 크게 GRAM, DUROC이라는 모듈을 가지며, 이들 사이에 RSL이라는 명세 언어가 존재하며, 이것은 자원에 대한 표현이나 자원 요구를 표현한다.

2.2 RSL

RSL(Resource Specification Language)은 글로벌스상

에서 전달되는 자원 명세와 작업 환경을 표현하는 언어로써 글로벌스 구성 요소간의 작업 요구를 전달한다.

자원 명세에는 자원의 타입, 필요한 노드 개수, 메모리등이 포함되고 작업 환경에는 실행 프로그램의 이름과 위치, 인수, 환경 변수 등이 포함된다.

```
Specification := request
Request       := multirequest | conjunction |
                disjunction | parameter
Multirequest  := +request-list
Conjunction   := &request-list
Disjunction   := |request-list
Request-list  := (request) request-list |(request)
Parameter     := parameter-name op value
              := = | > | < | >= | <= | !=
Value         := ({a...Z}[0...9] [_])+
```

(그림 2)RSL 문법

그림 2는 RSL 문법이다. RSL은 <에트리뷰트, 값> 쌍의 리스트로써 정보를 표현한다. [7]

RSL의 간단한 예를 들어보면 다음과 같다.

```
& (count>=5) (count<=10)
(max_time=240) (memory>=64)
(executable=myprog)
```

이 문장은 myprog라는 프로그램을 실행하는데 5개에서 10개의 동시 작업이 이루어 져야하고, 각각의 노드에서는 최소 64MB 이상의 메모리와 4시간 이상의 CPU 시간이 필요하다는 의미이다.

2.3 Broker

브로커는 그리드 미들웨어의 가장 중요한 컴포넌트의 하나이다. 브로커는 응용프로그램으로부터 받은 추상적인 요청(RSL)을 구체적인 요청(RSL)으로 변환하여 일을 한다. 그림 3에서 그 예를 보여 주고있다.

```
사용자 : &(count=5){executable=myprog}
↓
Broker : + (& resourceManagerContact=RM1){count=3}
          (executable=myprog))
          (& (resourceManagerContact=RM2)
           (count=2){executable=myprog})
```

(그림 3) 사용자 작성 RSL에 대한 브로커의 변환 즉, 브로커는 자원에 대한 정보를 요약하는 작업과 자원을 선택하는 작업 등을 수행한다. 중개자는 이러한 역할을 수행하기 위해 정보 서비스의 한 형태인

MDS에게 자원에 대한 정보를 문의한다[4]. 다중자원을 요구할 경우에는 그 요청을 동시 할당자에게 전달하고, 단일 요청일 경우에는 그 요청을 직접 GRAM에게 전달한다.

2.4 자원 동시 할당자

DUROC(Dynamically-Updated Request Online Co-Allocator)은 세분화된 RSL(Multi-request RSL)을 파싱해 각 세부 작업단위로 나눠 분배하고 작업이 동시에 실행되도록 하는 기능과 각 세부작업을 모니터링하고 그 결과를 취합하는 기능을 제공하는 에이전트이다[7].

3. 제안 시스템

3.1 글로벌스 툴킷

글로벌스 툴킷은 그리드 서비스를 제공하는 미들웨어로서 계산 그리드를 구축하는데 필요한 기본적인 기술이다. 이것은 그리드와 그리드 애플리케이션을 지원하는 커뮤니티 기반, 공개 구조, 공개 소스 집합의 서비스 및 소프트웨어 라이브러리로서 보안, 정보 발견, 자원 관리, 자료 관리, 통신 오류감지, 이식성 등 그리드에서 필요한 다양한 서비스들을 독립적인 요소로 제공한다.

글로벌스 툴킷은 크게 GSI(Globus Security Infrastructure), GIS(Globus Information Service), 자원 관리, 데이터 관리 등으로 이루어진다.

GSI는 보안을 담당하는 부분으로서, 이것은 단일 인증 방식(Single Sign-On), 통신 보호, 제한된 위임(Delegation)등을 제공한다.

MDS는 정보서비스를 수행하는 요소다. 그리드내에 존재하는 자원들의 상태 정보를 공유하고 사용자들에게 제공하기 위한 요소로서 인터넷의 DNS와 유사하다. 각 자원의 정보를 수집하여 제공하는 정보는 각 자원의 구조, 노드 수, 부하 정보, 배치작업 스케줄러, 네트워크 상태 등이다. 이러한 정보는 어플리케이션이나 개발자나, 자원 브로커 등에게 제공된다.

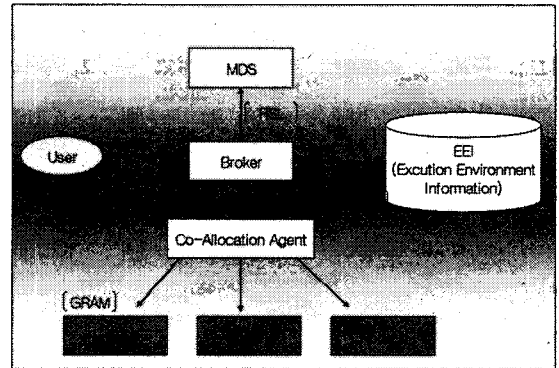
GRAM은 자원 관리를 담당하며, GRAM 프로토콜은 자원에 대해 신뢰성있고 안전한 원격 생성 및 관리를 수행한다.

GASS(Globus Access to Secondary Storage)는 데이터 관리를 수행한다. 이 GASS는 수행 노드 상에 있는 파일의 처리나 접근, 데이터의 분산 저장을 담당한다.

3.2 실행환경지원 시스템

본 논문에서는 사용자나 어플리케이션이 보낸 추상적인 RSL을 브로커가 명확한 RSL로 변환하는 과정에

서 실행환경정보의 정보를 통해 GRAM이 실행 환경을 이해할 수 있도록 하는 실행 환경 지원 메카니즘을 제안한다.



(그림 4) 실행 환경 지원 메카니즘

❖ 실행 환경 정보 메카니즘

제안한 실행 환경 정보 메카니즘은 사용자나 어플리케이션이 제출한 RSL을 브로커가 명확한 RSL로 변환할 때 제안한 EEI의 정보를 RSL의 environment 속성에 추가 함으로서 실행 환경을 인식할 수 있게 한다. 그림4는 다음과 같은 제안 실행 환경 메카니즘을 보여주고 있다.

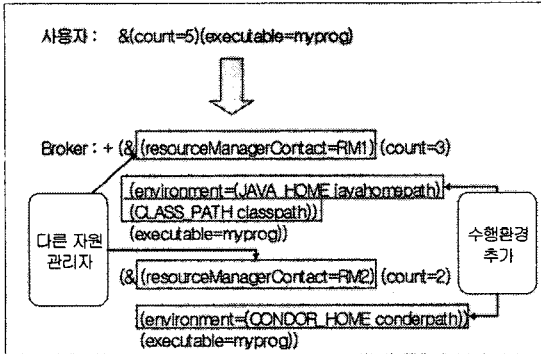
- ① 사용자는 그리드 포털을 통해 자신의 작업을 수행한다. 그리드 포털은 사용자가 손쉽게 그리드를 사용할 수 있도록 구현된 인터페이스이다.
- ② 그리드 포털로 넘겨진 작업은 간단하고 추상적인 작업단위의 RSL로 브로커에게 넘겨진다.
- ③ 브로커는 이 RSL 스크립트를 받아 MDS를 통해 발견한 자원 정보를 토대로 여러 개의 세부 작업으로 분할한 다음 표1의 EEI Table에서 실행 환경 정보를 가져와 보다 복잡하고 구체적 자원이 명시된 RSL 작업 요구를 생성한다(그림 5).

(표 1) 수행환경정보 테이블

노드명	타입	타입 패스	파라미터
Node 1	Java	Java Home path /opt/java	Class path
Node 1	Condor	Condor path	Class path
Node 2	Java	Java Home path /usr/local/java2	Class path
.	.	.	.

④ 이 작업 요구는 Co-Allocation agent에 전달되고

세부 작업들을 나눠 각각의 자원들로 할당해 동시에 실행되도록 한다.



(그림 5)수행환경이 추가된 RSL

㉔ 각각의 세부 작업들이 각 지역의 GRAM을 통해 할당되고 실행 환경을 이해한 프로세스를 생성하여 실행한다.

4. 결론 및 향후 연구과제

그리드는 지리적으로 분산된 자원을 네트워크로 연결된 사용자가 단일 시스템처럼 자원과 데이터를 공유하여 사용할 수 있도록 구성되어진 환경이다. 이러한 그리드 환경에서 애플리케이션을 수행하기 위해서는 그리드 환경에 적합한 자원 관리 시스템이 필수적이다. 기존의 글로버스의 GRAM은 Java 어플리케이션과 같은 특별한 수행 환경을 요구하는 어플리케이션에 대한 지원이 결여 되어있다.

본 논문에서는 EEI(Execution Environment Information)를 두어 RSL의 environment 속성에 수행 환경에 대한 정보를 추가하여 GRAM이 수행 환경을 지원할 수 있도록 하는 메카니즘을 제안하였다.

향후 연구로는 각 지역 자원의 기능을 적용해 사용자의 필요에 따라 지리적으로 흩어져 있는 자원들을 이용하여 동시에 병렬 작업을 수행할 수 있도록 하며, 동시 할당 기능, 사용자의 작업 요구에 맞는 자원을 그리드에서 찾아 내고 여러 자원들 사이에 스케줄링을 수행해 주는 브로커링 기능과 원격에서 수행중인 사용자의 작업에 대한 모니터링 기능을 제공 하는 자원 관리 시스템에 대한 연구와 구현이 요구된다.

[참고문헌]

[1] <http://www.50001.com> , “그리드 미들웨어 기술 동향”

[2] <http://globus.org>, “Globus Toolkit”
 [3] 김완석, 김정국, “그리드 기술 연구 동향”
 [4] 서울시립대 김학두, 김진석, “그리드 미들웨어: 자원 관리 및 원격 데이터접근 기술동향”,
 [5] Ian Foster, Carl Kesselman, Jeffrey M. Nick Steven Tuecke, “The Physiology of the Grid”, 2002. 6
 [6] 이춘희, “그리드 컴퓨팅(Grid Computing)”, 『정보처리 학회지』, 제10권 제1호, 2003. pp.109.
 [7] “월간 마이크로 소프트웨어”, 2002. 7
 [8] Paul D. Coddington, Lici Lu, Darren Webb, Andrew L. Wendelborn, “Extensible Job Managers for Grid Computing”, Twenty-Sixth Australasian Computer Science Conference, 2003
 [9] <http://www.globalgridforum.org/>