

PARIS : 피어-투-피어 기반의 그리드 정보 서비스 구조

임민열[○], 홍원택, 박형우, 이상산
한국과학기술정보연구원 슈퍼컴퓨팅센터
{philipus, wthong, hwpark, sslee}@kisti.re.kr

PARIS : Grid Information Service based on Peer-to-Peer architecture

Minyeol Lim[○], Wontaek Hong, Hyungwoo Park, Sangsan Lee
Supercomputing center, Korea Institute of Science and Technology Information(KISTI)

요 약

초고속 네트워크에 연동된 고성능 컴퓨팅 자원들을 효율적으로 활용하려는 그리드 컴퓨팅(Grid computing) 기술은 네트워크를 통한 기존의 단순한 정보 공유(Information sharing)뿐만 아니라 자원의 공유(Resource sharing)를 통해 분산된 자원들의 통합을 가능하게 한다. 이상적인 그리드 컴퓨팅 환경을 구축하기 위해서는 무엇보다도 그리드 환경에 속한 자원들의 정보를 통합적으로 관리 및 서비스하는 것이 무엇보다도 중요하다. 그리드 정보서비스의 역할은 그리드 사용자가 필요한 자원 및 그에 대한 정보를 쉽게 찾을 수 있도록 다양한 검색 기능을 제공하는 것 이외에도, 전세계에 분산된 자원들의 최신 정보를 빠르게 서비스하도록 하여야 한다. 이를 위해서는 그리드 환경의 특성을 고려하여 효율적인 그리드 정보 서비스 아키텍처를 구성하는 것이 필요하다. 본 논문에서는 기존에 널리 사용되는 그리드 정보 서비스(MDS) 모듈 구조인 클라이언트/서버 구조가 아닌 피어-투-피어 구조의 정보 서비스 구조(PARIS)를 제안한다. 이는 그리드 자원 활용의 지역성과 사용자 접근 패턴에 따라 정보 서비스의 정확성을 높이면서 지연 시간을 획기적으로 감소시킬 수 있다.

1. 서론

최근의 인터넷 기반의 컴퓨팅 환경은 정보의 공유(Information sharing)를 가능하게 한다. 이를 이용하여 전자상거래와 같은 다양한 서비스가 생성되었다. 하지만, 단순히 정보의 공유를 통해서만 실제로 우리가 할 수 있는 다양한 일들을 실현할 수 없다. 그리드 컴퓨팅[1]은 이 한계를 뛰어 넘는 것을 가능하게 한다. 즉, 앞에서 말했던 정보의 공유뿐만 아니라, 다양한 종류의 자원의 공유(Resource sharing)를 가능하게 하기 때문이다. 예를 들어, 어떤 사용자가 엄청난 컴퓨팅 자원을 필요로 한다면, 그리드 컴퓨팅 환경은 전세계에 존재하는 컴퓨팅 자원을 그리드 컴퓨팅 환경을 통해 마치 자신이 슈퍼컴퓨터를 보유한 것처럼 이용할 수 있다.

위와 같은 그리드 컴퓨팅 환경을 구축하기 위해서는 기본적으로 분산 컴퓨팅 환경에 존재하는 자원들에 대한 정보를 효율적으로 통합하여 접근할 수 있도록 지원하는 것이 필수적이다. 이것을 일반적으로 그리드 정보서비스라고 말한다[2]. 그리드 정보 서비스는 이기종의 다양하고 분산되어진 자원들에 대한 정보(Metadata)를 효율적으로 통합 관리하여, 그리드 사용자가 원하는 자원에 대한 정보를 효율적으로 서비스해야 한다. 그리드 사용자는 이를 통해 이질적이고 분산된 자원들을 자신이 직접 보유한 로컬 자원이고 또 하나의 자원인 것처럼 볼 수 있게 된다.

그리드 컴퓨팅의 전체 성능을 향상 시키기 위해서는 그리드 정보 서비스의 성능 향상이 필요하다. 왜냐하면, 그리드 사용자가 실제 필요한 자원을 사용하기 전에 그리드 정보 서비스를 통해 가용한 자원의 다양한 정보(예를 들어, 자원의 설정 및 성능)를 얻으려고 시도하기 때문이다. 따라서, 사용자에게 빠르고 항상 최상의 자원 정보를 제공하는 것이 그리드 정보 서비스의 주된 목표가 된다.

본 논문에서는 그리드 미들웨어의 시장 표준인 글로버스 툴킷(Globus Toolkit™)[3]내에 포함된 그리드 정보 서비스 컴포넌트인 MDS(Monitoring and Discovery Services)와 호환성을 유지하면서 피어-투-피어 형태의 서비스 구조를 보여준다. 이는 클라이언트/서버 구조를 가지는 MDS의 문제점을 해결하고 확장성과 응답시간을 빠르게 할 수 있는 서비스 구조이다.

본 논문의 구성은 우선 2장에서 글로버스 툴킷의 MDS에 의한 그리드 정보 서비스 구조를 살펴보고, 3장에서는 이를 개선한 구조를 살펴본다. 4장에서는 3장에 기술된 내용에 대한 프로토타입 구현에 대해 기술한다. 그리고 마지막 장에서 정리 및 향후 발전 방향을 기술한다.

2. 글로버스 툴킷내 그리드 정보 서비스

현재 글로버스 툴킷 2.2버전에 포함된 MDS2.2버전은 그림1과 같은 서비스 구조를 가지고 있다. MDS는 기본적으로

로 두 가지 서비스 즉, GRIS(Grid Resource Information Services)와 GIIS(Grid Index Information Services)로 나누어 진다.

- GRIS는 특정 로컬 자원에 대한 메타데이터를 유지한다. 이는 다양한 정보 제공자(Information Provider)로부터 특정한 주기마다 메타데이터를 받아 유지한다.

- GIIS는 GRIS를 통해 제공되는 메타데이터를 계층적 구조로 모아 가상의 통합된 정보 서비스를 제공한다. 이를 통해 그리드 컴퓨팅의 가상의 조직(VO : Virtual Organization) 단위로 통합된 그리드 정보 서비스를 제공할 수 있다.

GRIS는 자신이 속한 자원에 대한 정보를 여러 개의 정보 제공자(Information Provider)들로 세분화하여 수집하게 된다. 예를 들어, CPU, Memory, Disk, Network, Job manager와 같은 정보 제공자가 존재한다. 이 정보 제공자들은 미리 정해진 서로 다른 주기를 가지고 GRIS에 의해 호출되며 담당하는 메타데이터를 전달하게 된다. GRIS는 이 메타데이터를 LDAP(Lightweight Directory Access Protocol) 서버에 저장한다[4][5]. GIIS는 분산된 자원들을 계층적 구조로 통합시킨다. 그림 1에 보이는 것과 같이, GRIS로부터 독립된 자원들의 메타데이터를 받아서 모으게 되며, 이들을 다시 상위의 GIIS에 보내 통합하게 된다. 이를 이용하여 그리드 컴퓨팅을 위해 생성된 VO 단위로 자원들을 통합할 수 있다. GIIS는 GRIS와 마찬가지로 LDAP 서버를 이용하여 메타데이터를 유지하고 질의 서비스를 제공한다.

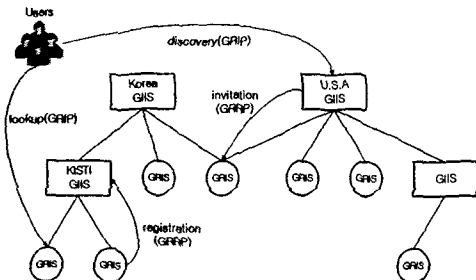


그림 1. MDS기반 그리드 정보 서비스 구조

글로벌스의 MDS를 통해 제공되는 그리드 정보 서비스는 기본적으로 클라이언트/서버 구조로 이루어진다. 이는 프로그램이 용이하다는 장점을 가지고 있지만, VO 마다 별도의 서버(GIIS 서버)를 두어야 하고, VO의 확장성 및 정보의 가용성 측면에서 한계를 가진다. 또한, MDS 아키텍처상의 문제로서 서비스 되는 정보의 정확성이 떨어지고 서비스의 응답시간이 상당히 큰 문제를 안고 있다.

3. PARIS 구조

PARIS는 기존의 MDS와 호환성을 유지하면서 그리드 컴퓨팅 환경의 특성에 맞는 서비스를 제공하기 위한 그리드 자원 정보 서비스 구조이다. 이는 피어-투-피어[6] 기반의 서비스 구조를 기반으로 한다. 왜냐하면, 그리드 컴퓨팅은 쉽게 말해 자원의 효율적인 공유를 통해 컴퓨팅 자원의 활용도를 높이고 보다 높은 컴퓨팅 성능을 이끌어내기 위한 환경이다. 이를 통해 자원들끼리 새로운 조직, 즉 VO에 대한 생성과 소멸이 빈번히 이루어 지고, 그 VO에 대한 컴퓨팅 자원의 가입 및 탈퇴가 자유롭다. 따라서, 자원 정보 서비스에 대한 확장성 및 정확성에 대한 요구가 매우 높다. 또한, 사용자는 VO에 속한 모든 자원을 한꺼번에 동시에 이용할 가능성은 상대적으로 적은 반면, 실제적으로 자원의 활용은 지역성(locality)을 가지게 된다. 따라서, MDS와 같이 특정 한 서버가 분산된 자원들의 정보를 유지하고 서비스하는 것보다 자원들끼리 자신의 정보를 공유할 수 있도록 하는 것이 그리드 정보 서비스측면에서 효율적이게 된다.

그림 2는 PARIS 서비스 형태의 한 예를 보여준다. 여기서 특정컴퓨팅 자원은 Peer 개념으로 매핑된다. Peer 1이 자원 정보 요청을 받으면 그 VO에 속한 Peer들에게 요청을 전달한다. Peer 2,3는 Peer 1의 요청에 대해 검색 조건을 만족하는지 체크하고 만족하면 응답을 하고 Peer 4 또는 5에게 그 요청을 포워딩한다. Peer 4,5는 Peer 2,3과 마찬가지로 Peer 1에게 결과를 리턴한다. 따라서, PARIS를 통해 제공되는 서비스는 비동기식(Asynchronous) 형태의 서비스를 제공하게 된다. 여기서, Peer 3는 Peer 1이 다른 자원들에 대한 정보를 얻을 수 있도록 게이트웨이 역할을 수행하는 란데뷰 피어(Rendezvous Peer)가 된다.

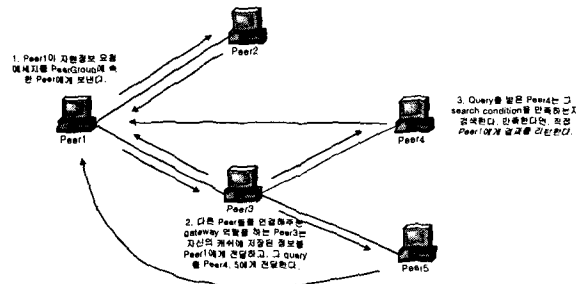


그림 2. PARIS 서비스 형태

위의 그림 에서 각 자원들의 정보는 기본적으로 요청을 받으면 그 결과를 전달하도록 되어 있다. 하지만, 자원 정보 서비스의 지연 시간을 줄이기 위해서 어느 정보의 캐싱 메커니즘을 필요로 한다. 예를 들어, Peer 3가 Peer 4 또는 5에 대한 자원 정보를 캐싱하고 있다면 Peer 1의 자원 정보 요청에 대한 응답 시간은 더 줄어들 수 있다.

그림 3은 위의 요구 사항을 고려하여 각 자원에서 제공

될 그리드 정보 서비스의 내부 구조를 보여준다. PARIS는 사용자의 요청을 처리하는 요청처리기 이외에 크게 로컬 자원정보관리기, 캐쉬관리기, 그리고 PeerGroup관리기 3부분으로 나누어진다.

을 검색하고 그 자원에 대한 상세 정보를 얻을 수 있다. 자원에 대한 상세 정보는 그림과 같이 계층적(Hierarchical) 구조의 트리 형태로 보여진다.

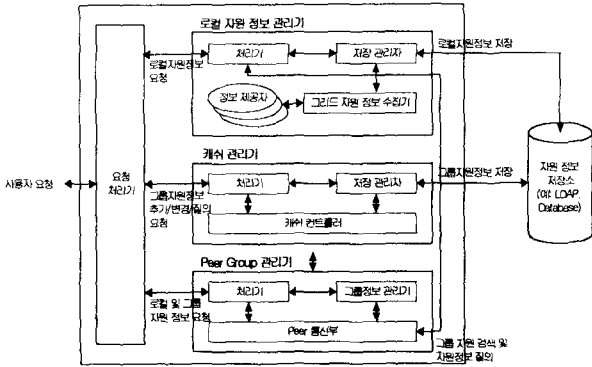


그림 3. PARIS 구조

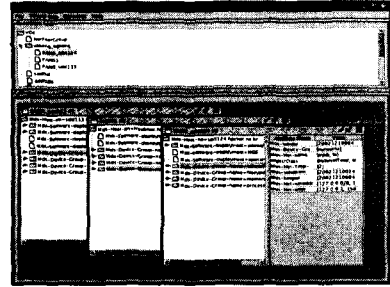


그림 4. PARIS 프로토타입 사용자 인터페이스

로컬자원정보관리기는 기존 MDS의 GRIS와 같은 역할을 수행하는 부분으로 각 로컬 자원에 대한 정보를 수집, 저장하고 서비스하도록 되어 있다. 로컬 자원 정보는 그리드 자원정보수집기가 각 정보제공자로부터 자원의 특성에 따라 다양한 특정한 주기를 가지고 정보를 모으게 된다.

PeerGroup관리기는 각 피어들간에 자원정보를 공유할 수 있도록 서로 통신하는 모듈을 포함한다. 이를 통해, 다른 피어에 대한 정보를 얻게 되며, 필요에 따라 랑대부 피어와 같은 역할을 수행하게 된다.

캐쉬관리기는 PARIS의 서비스 지연 시간을 단축하기 위한 메커니즘이다. 캐쉬관리기를 통해 각 피어는 자주 사용되는 자원에 대한 정보를 캐싱하게 된다. 특히, 랑대부 피어는 다른 랑대부 피어들이 유지하는 정보를 캐쉬관리기를 통해 공유함으로써 사용자의 자원 정보 요청을 보다 빠르게 리턴하고 효율적으로 포워딩하게 된다.

4. 구현

PARIS에 대한 프로토타입 구현은 기존의 피어-투-피어 라이브러리인 JXTA[7]를 이용하였다. JXTA는 썬마이크로 시스템에 의해 개발되고 있는 자바 플랫폼기반의 피어-투-피어 라이브러리이다.

그림 4는 PARIS의 프로토타입에 대한 사용자 인터페이스를 보여준다. PARIS 프로토타입은 기존 MDS와의 호환성을 위해 LDAP 프로토콜기반의 전송 데이터 구조를 이용한다. 따라서, 사용자의 자원 정보 요청에 대한 결과는 LDIF(LDAP Data Interchange Format)[8] 형태를 가진다. 사용자는 PARIS 프로토타입을 통해 동적으로 새로운 VO 생성하고 가입 및 탈퇴를 할 수 있고, 특정 VO에 속한 자원

5. 결론 및 향후 계획

PARIS는 사실상의 표준으로 자리잡고 있는 글로벌스 툴킷내 그리드 정보 서비스 요소인 MDS의 문제들을 해결 또는 보완할 수 있는 서비스 구조이다. 이는 그리드 컴퓨팅의 특성인 VO 및 자원 정보의 동적성과 분산 환경을 통한 고성능에 대한 요구를 만족시킬 수 있는 구조를 가진다. 이를 기존의 라이브러리를 이용한 프로토타입 구현을 통해 입증하였다. 하지만, 이를 실질적인 서비스 모듈로 완성시키기 위해서는 JXTA를 이용한 어플리케이션 형태가 아니라 JXTA 내부의 Peer Information Protocol(PIP)를 통한 PARIS 모듈을 구현해야 할 것이다. 또한, 기존 글로벌스 툴킷과의 완벽한 호환성을 위해서는 Grid Security Infrastructure(GSI)를 통한 PARIS 서비스를 접근할 수 있도록 보안 기능을 추가해야 할 것이다.

참고문헌

- [1] I. Foster, C. Kesselman, S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International J. Supercomputer Applications, 15(3), 2001.
- [2] K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman, "Grid Information Services for Distributed Resource Sharing.", Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.
- [3] I. Foster, C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", Intl J. Supercomputer Applications, 11(2):115-128, 1997.
- [4] I. Foster, G. von Laszewski, "Usage of LDAP in Globus", <http://www.globus.org>, 1998
- [5] RFC2251, "Lightweight Directory Access Protocol(v3)"
- [6] Andy Oram, "Peer-To-Peer", O'REILLY, 2001
- [7] Brendon J. Wilson, "JXTA", New Riders, 2002
- [8] RFC2252, "Lightweight Directory Access Protocol(v3) : Attribute Syntax Definitions"