

# 그리드 컴퓨팅

한국과학기술정보연구원 김동균 · 이필우 · 황일선

## 1. 서론

그리드(Grid)는 지리학적으로 분산되어 있는 고성능 컴퓨팅 자원을 네트워크로 상호 연동하여 조직과 지역에 관계없이 사용할 수 있는 환경을 말한다. 그리드라는 단어는 1990년 중반 미국의 슈퍼컴퓨팅센터를 중심으로 고성능의 분산 컴퓨팅 인프라를 구축하는데서 비롯되었고, 고성능 자원, 대용량 정보 및 혁신적인 애플리케이션에 초점이 맞추어진 것이 일반적인 분산 컴퓨팅과 구별된다[17].

그리드는 일반적으로 계산 그리드(computational grid), 데이터 그리드(data intensive grid), 액세스 그리드(access grid)로 구분할 수 있다. 계산 그리드는 분산되어 있는 고성능 컴퓨팅 자원을 연결하여 하나의 가상공간에서 사용함으로써 지금까지 불가능하였던 연구를 수행하기 위한 환경을 제공하고 high-throughput 계산환경을 구축하기 위한 고성능 컴퓨팅 자원을 통합하는 기술이다. 데이터 그리드는 한 곳에 집중된 대량 데이터를 효율적으로 공유하고, 여러 곳에 분산되어 있는 대량의 데이터 및 DB에 실시간으로 접근하기 위하여 단계적이고 체계적인 데이터 접근체계를 구축하는 기술을 말한다. 액세스 그리드는 동일 분야 연구자들의 공동 연구나 정책 결정을 위한 다양한 분야 연구자들이 원격지에서 접근하여 의견을 교환하고 협력할 수 있는 고성능 협업환경을 구축하는 기술이다.

이와 같이 그리드는 대량의 정보와 고성능 컴퓨팅 자원을 효율적으로 이용하기 위하여 차세대 인터넷(next generation internet)을 기반으로 구축되고 있으며, 새로운 응용연구의 수행에 큰 변화를 가져올 것으로 기대된다.

## 2. 그리드의 구조

본 절에서 언급하는 그리드 구조는 그리드 프로토콜과 서비스의 나열이 아니라 범용적인 구성 요소들을 위한 요구 사항을 나타낸다. 이러한 구성 요소들은 그림 1과 같이 각각의 계층으로 나누어져 분류된다. 그리드 구조의 다양한 계층을 나타냄에 있어서 흔히 “모래 시계” 모델[11]이 사용되는데 모래 시계의 좁은 목 부분은 핵심적인 기능과 프로토콜을 나타낸다. 모래 시계 모델의 상위 부분에는 서로 다른 고수준(high-level) 기능들이 매핑되고 이와 같은 기능들은 핵심 프로토콜을 통해서 모래 시계의 아래 부분에 위치하는 수많은 기반 기술과 대응된다. 본 절에서 제시되는 그리드 구조는 모래 시계의 목 부분은 자원과 연결 프로토콜(Resource and Connectivity protocol)로 구성되어 독립적인 자원들을 공유할 수 있는 기능을 제공한다. 그리고 이들 프로토콜을 통하여 상위에 있는 Collective 계층과 하위의 Fabric 계층이 각각 다양한 범위의 서비스, 애플리케이션 및 링크 자원을 서로 공유할 수 있게 된다.

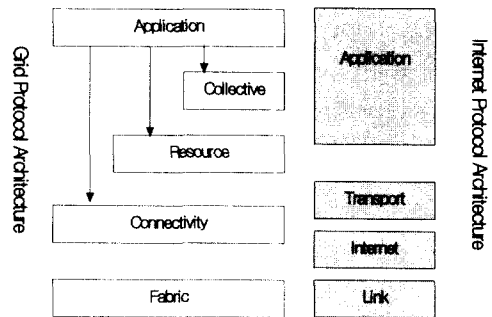


그림 1 그리드 구조 계층과 인터넷 프로토콜 계층의 관계

### 2.1 Fabric 계층

그리드 Fabric 계층은 그리드 프로토콜에 의해서 제어되는 공유 액세스에 자원을 제공하는 역할을 담당한다. "자원"은 분산 파일 시스템, 컴퓨터 클러스터, 분산 컴퓨터 풀 등의 논리적인 집합일 수도 있고 계산 자원, 저장 시스템, 카탈로그, 네트워크 자원과 센서 등일 수도 있다. 논리적인 자원의 경우에 자원 구현은 NFS 저장 장치 액세스 프로토콜이나 클러스터 자원관리 프로토콜과 같은 내부의 프로토콜과 관련되어 있으나 그리드 구조에서는 이런 것들이 크게 관계가 없다. Fabric 계층의 구성요소는 상위 레벨의 공유작업에 따른 결과로서 특정한 자원에서 발생하는 자원-제한적이며 지역적인 기능을 구현한다. 따라서 Fabric 레벨에서 구현되는 기능들과 공유 작업 사이에는 밀접한 상호 의존성이 존재하게 된다. 예를 들어 자원 레벨에서 자원 예약이 지원된다면 상위 레벨의 서비스들이 좀더 향상된 방법으로 자원들을 결합 혹은 공동 스케줄링할 수 있게 된다.

## 2.2 Connectivity 계층

Connectivity 계층은 그리드에 한정적인 네트워크 트랜잭션 처리를 위해서 요구되는 핵심적인 통신과 인증 프로토콜을 정의한다. 통신 프로토콜은 Fabric 계층 자원들 간의 데이터 교환을 활성화하고, 인증 프로토콜은 사용자와 자원의 식별을 검증하는 암호화 보안 메커니즘을 제공한다. 통신이 요구하는 것은 전송, 라우팅, 그리고 네이밍등이다. 현존하는 TCP/IP 프로토콜 스택을 가지고 이러한 기능들을 사용할 수 있는데, 예를 들면 인터넷 계층적 프로토콜 구조의 Internet(IP와 ICMP), Transport(TCP와 UDP), 그리고 애플리케이션(DNS, OSPF, RSVP등) 프로토콜등이 있다[2]. 물론 가까운 미래에 그리드 구조에서 동적인 네트워크를 고려하기 위하여 새로운 프로토콜을 요구할 수도 있다.

인증 기능이 가져야 하는 특징은 "Single sign on"과 "Delegation"기능, 다양한 지역 보안 솔루션들의 통합, 사용자 기반의 신용관계 등이다. 또한 그리드 보안 솔루션은 통신 보안을 위한 효율적인 지원을 해야 하며, 인가 결정에 대한 제어를 가능하게 해주어야 하고, 다양한 방법으로 대표성 권한을 제한할 수 있어야 한다. 통신 보안을 위한 효율적인 지원은 예를 들어 보안 등급의 제어, 안정적인지 못한 프로토콜의 독립적인 데이터 단위 보안, 그리고 TCP 이외의 안정적인 전송계층 프로토콜 지원 등이 있다.

## 2.3 Resource 계층(단일자원 공유)

Resource 계층은 개별적인 자원 상에서 이루어지는 공유 작업에 대한 보안 협상, 초기화, 모니터링, 제어 등을 위한 프로토콜을 정의한다. 이러한 Resource 계층의 프로토콜을 구현하기 위해서는 Fabric 계층의 함수들을 사용하여 지역 자원들을 제어하고 액세스한다. Resource 계층은 단지 개별적인 자원에만 한정되며, 따라서 전역적인 상태와 분산 집합체상의 개별적인 활동들은 무시하는데, 이러한 내용들은 Collective 계층에서 다루어지게 된다. Resource 계층의 두 가지 부분은 다음과 같다.

- Information 프로토콜 : 자원의 상태와 구조에 대한 정보, 즉 자원의 구성, 현재 부하, 그리고 사용 정책 등에 대한 정보를 얻기 위해 사용
- Management 프로토콜 : 공유 자원에 대한 액세스를 협상하는데 사용된다. 예를 들면 자원 예약과 품질보장 서비스등의 자원 요구사항을 정의하고 프로세스 생성 혹은 데이터 액세스와 같은 작업수행 등을 관리한다. 이외에 Management 프로토콜은 작업의 현황 모니터링, 작업 제어등을 담당한다. Resource와 Connectivity 계층은 모래시계 모델의 목 부분을 형성하며 작고 제한된 집합을 구성하게 된다. Resource 계층의 프로토콜들은 반드시 수많은 서로 다른 자원들의 근본적인 메커니즘을 충족시킬 수 있도록 선택되어야 한다[9].

## 2.4 Collective 계층

Resource 계층이 단일 자원과의 상호작용에 초점을 맞추었다면, 그리드 구조의 다음 계층은 자원의 집합을 통한 상호작용에 관련된 서비스(API(application programming interface)와 SDK(software development kits)) 및 프로토콜을 포함한다. 이러한 이유로 이 계층을 "Collective" 계층이라고 부른다. Collective 계층이 모래시계 모델의 좁은 부분을 구성하는 Resource와 Connectivity 계층 위에 구현되기 때문에 공유되고 있는 자원에 대해서 새로운 요구 없이 다양한 공유 기능을 구현할 수 있다. 이러한 기능들의 예는 다음과 같다.

- Directory Services
- Co-allocation, scheduling, and brokering services[4,5,8]

- Monitoring and diagnostics services
- Data replication services[9]
- Grid-enabled programming systems
- Workload management systems and collaboration frameworks
- Software discovery services
- Community authorization services
- Community accounting and payment services
- Collaboratory services

Collective 계층의 기능은 지속적인 서비스로써 구현되어야하며 애플리케이션과 연동될 수 있는 SDK나 관련된 프로토콜과 함께 만들어져야 한다. 이 두가지 경우 모두 Resource 계층 프로토콜과 SDK의 상위에서 구축된다. 예를 들어서 그림 2는 Collective 계층의 co-allocation API와 SDK를 보여주는데 이들은 미들웨어 계층으로서, 하위 기반이 되는 자원들을 제어하기 위하여 Resource 계층의 Management 프로토콜을 사용한다. co-reservation 서비스 프로토콜이 역시 정의되는데, co-allocation 작업을 구현하기 위하여 관련 API를 호출하고 Authorization, Fault Tolerance, 그리고 logging 등과 같은 추가적인 기능을 제공한다. 이와 같은 단계를 거쳐서 애플리케이션은 단대단 네트워크 자원 할당을 요구하기 위하여 co-reservation 서비스 프로토콜을 사용한다.

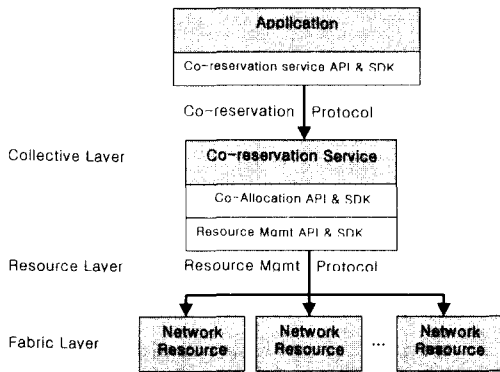


그림 2 Collective와 Resource 계층의 프로토콜, 서비스, API, SDK의 기능 구조

### 2.5 Application 계층

그리드 구조의 마지막 계층은 VO(virtual organi-

zation)내에서 동작하는 사용자 애플리케이션들로 이루어져 있다. 그림 3은 그리드 구조에 대한 애플리케이션 프로그래머의 관점을 보여준다. 애플리케이션은 모든 계층에서 정의되는 서비스를 위해서 구축된다. 각각의 계층에는 자원 관리, 데이터 액세스, 자원 발견 등과 같은 유용한 서비스에 접근하기 위해 정의된 프로토콜들이 있으며 API가 역시 제공된다.

그림 3에서 API는 SDK에 의해서 구현되는데 SDK는 단말 사용자에게 여러 가지 기능을 제공하기 위한 네트워크 서비스와 상호 동작하기 위해 그리드 프로토콜을 사용한다. 상위 레벨의 SDK는 특정 프로토콜에 직접 맵핑하지 않지만 지역적인 기능 구현 및 부가적인 API 호출과 프로토콜 동작을 연관시킬 수 있는 기능을 제공한다.

그림에서 실선은 직접적인 호출을 나타내고 점선은 프로토콜간의 상호 동작을 나타낸다.

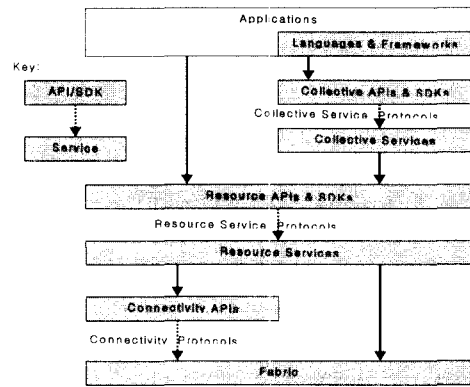


그림 3 애플리케이션 관점의 그리드 구조

## 3. 그리드 컴퓨팅 환경

### 3.1 그리드 협업환경

원거리에 있는 연구자들이 협력하여 하나의 과제를 공동으로 수행하기 위해서는 가상현실을 위한 시스템, 고해상도 장비 등 고성능 협업 장비들이 필요하다. 이들 장비가 위치하는 많은 사이트들 사이에는 네트워크로 연결되어 있어야 함은 물론이고, 오디오, 비디오, 제어 명령, 데이터 등을 포함하는 다양한 자료들이 협업이 진행되는 동안 대화식으로 빈번하게 주고받게 된다. 이런 응용분야들은 주고받아야 할 다양한 데이터들이 QoS적인 특성들을 가지고 있으며,

이들 흐름의 특징이 체계적으로 관리되는 QoS기능이 필요하다. 한편, 고성능 애플리케이션들을 위한 QoS 서비스의 개발은 전체 시스템의 아키텍처 수준과 네트워크 수준에서 처리해야 할 문제가 있다. 아키텍처 수준에서는 여러 자원들 사이의 흐름들을 처리하기 위하여 새로운 개념과 구조들이 요구된다. 예를 든다면 가용성 보장을 위한 예약 방법, 네트워크, 계산자원, 저장 장치 등 자원들을 동시에 사용하기 위한 예약방법 등이다. 네트워크 수준에서는 많이 사용되는 DS(Differentiated Services)방법이 여러 문제들을 제시하고 있다. DS는 확장성에서 장점을 가지고 있지만 전문화된 고성능 흐름들을 어떻게 처리할 것인가는 앞으로 풀어야 할 숙제이다.

### 3.2 대용량 데이터 전송

어떤 영역에서는 많은 데이터들이 원격으로 가시화되지 않고 일부만 또는 전체적으로 원격지 저장장치에 전달된다. 저장 장치에 전달된 후 데이터에 대한 작업이 이루어 질 것이기 때문이다. 이때 관계되는 여러 자원들과 네트워크는 여러 사람이 공유하는 것이기 때문에 여러 자원들의 사용과 기가급 또는 테라급 전송 사이에는 시간적인 일치를 보아야 한다. 일치를 위해 필요한 것은 충분한 대역폭이 아니라 정해진 시간 내에 전달 되어야 한다는 QoS의 보장이다. 이와 같은 목표를 얻기 위해서는 네트워크 뿐만 아니라 저장장치와 CPU등 여러 자원들간의 스케줄링이 필요하다.

이와 같이 고성능 컴퓨팅 애플리케이션 응용 분야 중 실시간 가시화, 대용량의 정보처리 및 대화형 협업연구 환경이 강한 애플리케이션을 그리드 컴퓨팅 환경에서 개발한다면 보다 쉽게 개발할 수 있다. 특히 유체역학, 물리학, 인공심장의 개발 및 설계, 기타 제조설계 분야 등은 이러한 고성능 애플리케이션이 요구하는 협동공학의 필요 충분조건을 갖추고 있어서 차세대 인터넷 서비스의 선도 애플리케이션으로 적합하다.

## 4. 결론

그리드 컴퓨팅에서는 차세대 인터넷 기술을 기반으로 연결된 고성능 컴퓨팅 자원을 이용하여 대형의 문제들을 분산 처리할 수 있고, 처리 후 원격지에 있는 연구자들이 고성능 가시화 장비들을 이용하여 협

업을 할 수 있는 환경을 제공한다. 이와 같은 기술은 차세대 인터넷의 급속한 발전과 Globus와 같은 미들웨어의 기술 개발로 성능 저하 현상의 주요 원인이었던 저속의 네트워크 문제와 통합의 어려움이 해결되고 있다. 이에 따라 미국과 유럽은 국가 슈퍼컴퓨터 자원들과 고성능 실험장비들을 결합하여 네트워크로 연결된 그리드 환경을 구축하고 있으며, 국내에서도 정보통신부가 주관하여 국내 고성능 컴퓨터 자원을 연결하는 국가 그리드 구축 사업이 추진 중에 있다.

## 참고문헌

- [1] Realizing the Information Future: The Internet and Beyond. National Academy Press, 1994. <http://www.nap.edu/readingroom/books/rtif/>
- [2] Butler, R., Engert, D., Foster, I., Kesselman, C., Tuecke, S., Volmer, J. and Welch, V. Design and Deployment of a National-Scale Authentication Infrastructure. *IEEE Computer*, 33(12):60-66. 2000.
- [3] Czajkowski, K., Fitzgerald, S., Foster, I. and Kesselman, C. Grid Information Services for Distributed Resource Sharing, 2001.
- [4] Berman, F. High-Performance Schedulers. In Foster, I. and Kesselman, C. eds. *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999, 279-309.
- [5] Berman, F., Wolski, R., Figueira, S., Schopf, J. and Shao, G. Application-Level Scheduling on Distributed Heterogeneous Networks. In *Proc. Supercomputing '96*, 1996.
- [6] Frey, J. Foster, I., Livny, M., Tannenbaum, T. and Tuecke, S. Condor-G: A Computation Management Agent for Multi-Institutional Grids, University of Wisconsin Madison, 2001.
- [7] Abramson, D., Sosic, R., Giddy, J. and Hall, B. Nimrod: A Tool for Performing Parameterized Simulations Using Distributed Workstations. In *Proc. 4th IEEE Symp. on High Performance Distributed Computing*, 1995.
- [8] Beiriger, J., Johnson, W., Bivens, H., Humphreys, S. and Rhea, R., Constructing the ASCI Grid. In *Proc. 9th IEEE Symposium on*

*High Performance Distributed Computing*, 2000, IEEE Press.

[9] Allock, B., Bester, J., Bresnahan, J., Chervenak, A.L., Foster, I., Kesselman, C., Meder, S., Nefedova, V., Quesnel, D. and Tuecke, S., Secure, Efficient Data Transport and Replica Management for High-Performance Data-Intensive Computing. In *Mass Storage Conference*, 2001.

[10] Casanova, H., Obertelli, G., Berman, F. and Wolski, R., The AppLeS Parameter Sweep Template: User-Level Middleware for the Grid. In *Proc. SC'2000*, 2000.

[11] Armstrong, R., Gannon, D., Geist, A., Keahy, K., Kohn, S., McInnes, L. and Parker, S., Toward a Common Component Architecture for High Performance Scientific Computing. In *Proc. 8th IEEE Symp. on High Performance Distributed Computing*, 1999.

[12] Casanova, H., Dongarra, J., Johnson, C. and Miller, M., Application-Specific Tools. In Foster, I. and Kesselman, C. eds. *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999, 159-180.

[13] Casanova, H., Obertelli, G., Berman, F. and Wolski, R., The AppLeS Parameter Sweep Template: User-Level Middleware for the Grid. In *Proc. SC'2000*, 2000.

[14] Chervenak, A., Foster, I., Kesselman, C., Salisbury, C. and Tuecke, S., The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Data Sets. *J. Network and Computer Applications*, 2001.

[15] Childers, L., Disz, T., Olson, R., Papka, M.E., Stevens, R. and Udeshi, T., Access Grid: Immersive Group-to-Group Collaborative Visualization. In *Proc. 4th International Immersive Projection Technology Workshop*, 2000.

[16] Clarke, I., Sandberg, O., Wiley, B. and Hong, T.W., Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *ICSI Workshop on Design Issues in Anonymity and Unobservability*, 1999.

[17] Czajkowski, K., Fitzgerald, S., Foster, I. and Kesselman, C., Grid Information Services for Distributed Resource Sharing, 2001.

[18] Czajkowski, K., Foster, I., Karonis, N., Kesselman, C., Martin, S., Smith, W. and Tuecke, S., A Resource Management Architecture for Metacomputing Systems. In *The 4th Workshop on Job Scheduling Strategies for Parallel Processing*, 1998, 62-82.

[19] Czajkowski, K., Foster, I. and Kesselman, C., Co-allocation Services for Computational Grids. In *Proc. 8th IEEE Symposium on High Performance Distributed Computing*, 1999, IEEE Press.

[20] DeFanti, T. and Stevens, R., Teleimmersion. In Foster, I. and Kesselman, C. eds. *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999.

[21] Frey, J., Foster, I., Livny, M., Tannenbaum, T. and Tuecke, S., Condor-G: A Computation Management Agent for Multi-Institutional Grids, University of Wisconsin Madison, 2001.

---

김 동 균



1996. 2 한남대학교 전자계산공학과  
공학사  
2000. 2 중남대학교 컴퓨터학과와 컴퓨  
터 네트워크 전공 이학석사  
2000. 6-현재 한국과학기술정보연구원  
슈퍼컴퓨팅센터 연구원  
관심분야 Access Grid 응용 및 Grid 네트  
워킹, IPv4/IPv6 유니캐스트 & 멀티  
타캐스트 라우팅  
E-mail: mirm@kreonnet2.net

---

이 필 우



1988. 2 동국대학교 전기공학과 공학사  
1991. 3 쓰쿠바대학 이공학연구과 컴퓨터공학전공, 공학석사  
1999. 3 쓰쿠바대학 공학연구과 컴퓨터공학전공, 공학박사  
2000. 4~현재 한국과학기술정보연구원(KISTI) 슈퍼컴퓨팅센터 선임연구원  
관심분야 컴퓨터구조, 프로그래밍환경, 분산컴퓨팅.

E-mail: pwlee@hpcnet.nerl

황 일 선



1993. 5 시스템공학연구소(SERI) 실장  
1998. 9 한국전자통신연구원(ETRI) 초고속정보망 실장  
현재 한국과학기술정보연구원(KISTI) 슈퍼컴퓨팅센터 초고속연구망 실장  
관심분야 Grid 네트워킹, 멀티캐스트 & IPv6 네트워킹  
E-mail: his@hpcnet.nerl

● 제4회 한국소프트웨어공학 학술대회 ●

- 일 자 : 2002년 3월 26 ~ 27일
- 장 소 : 서울대 호암교수회관
- 주 최 : 소프트웨어공학연구회
- 문 의 처 : 비트컴퓨터 기술연구소 전진옥 소장  
Tel. 02-3486-1045