

그리드 연구개발 동향

천안대학교 강경우
한국과학기술정보연구원 박형우

1. 서 론

1.1 그리드 기본정의

인터넷이 보편화되고 컴퓨터 및 네트워크의 성능이 향상됨에 따라 분야에 따라 컴퓨터를 이용하는 양상은 변하고 있다. 분산처리 분야에서 네트워크의 기술적인 변화는 통신망을 통해 서로 연결된 컴퓨터들을 하나의 문제를 해결하는데 사용할 수 있도록 해주고 있다. 지금까지는 동일한 컴퓨팅 자원들을 통합하는 NOW(Network Of Workstations), PC클러스터링 기술에 노력하여 왔다. 이제 또 다른 하나의 시도로서 자원통합에 있어서 동일 기종 컴퓨터들만이 아니라 이기종 컴퓨터 자원들과 대용량 저장장치, 다양한 고성능 연구 장비들이 포함되고 있는데, 이러한 통합된 환경을 "Grid"(그리드)라는 용어를 가지고 표현하고 있다[6].

1.2 그리드 역사

그리드와 유사한 용어로서 메타 컴퓨터 혹은 메타 컴퓨팅이라는 용어는 80년대 후반에 생겨났다. 네트워크 환경에서 연결된 워크스테이션들과 슈퍼컴퓨터들을 사용자가 단일 컴퓨터를 이용하듯이 사용할 수 있는 개념에 관련되었다. 초기의 메타 컴퓨터는 CM-2, Cray-2, 파일서버, 워크스테이션 정도를 기가비트 LAN으로 연결하고자 하는 것이 고작이었다. 사실 메타 컴퓨팅과 유사한 용어가 각 분야마다 독립적으로 사용되어 왔는데, 예를 들면, seamless 컴퓨팅, scalable 컴퓨팅, global 컴퓨팅 등이 그것이다. 그러다가 1990년대 후반에 그리드라는 용어로 통일이 되었다[6, 17].

그리드에 관한 연구개발은 1995년 슈퍼컴퓨팅 컨

퍼런스를 기점으로 확산되기 시작하였다. SC95에서 I-WAY의 시연은 OC-3 네트워크를 이용해서 연결된 슈퍼컴퓨터들과 Immersadesk상에서 60여 개의 어플리케이션을 수행함으로써 고성능 네트워크 기반 컴퓨팅의 가능성을 보여주었다. I-WAY의 성공적인 시연을 기점으로 미국 내에는 여러 개의 정부주도 그리드 개발 과제가 시작되었는데 대표적인 것이 다음과 같다[3].

- NSF PACI : 1998년 NCSA와 SDSC 중심으로 시작된 연구로써, 동역 연구기관들은 고성능 컴퓨팅 자원들이 연결된 "국가 그리드"의 발전을 위해 필요한 도구들과 알고리즘을 개발하고 테스트하는 일을 하고 있다[18].
- IPG : 1999년 NASA Ames, Glenn, Langley의 장비들을 초고속망으로 연결하여 NASA내 통합된 연구환경을 제공하고자 하는 연구이다 [23].
- ASCI DISCOM : 1999년 에너지성이 중심이 되어 무기관련 연구를 위한 테라플롭스급 계산기술을 확보하기 위해 시작하였다[24].

2. 그리드의 개념

2.1 그리드 응용분야

그리드 미들웨어를 개발하는 것은 지금까지 연구 및 개발되었던 전통적인 분산 시스템과 비교할 때 규모 면이나 기술적인 면에서 큰 차이가 있다. 왜냐하면 그리드에서 다루는 어플리케이션의 파라미터 연구나 데이터 분석은 수천 개의 프로세스와 수만의 데이터파일 그리고 페타바이트급의 데이터를 처리해야 하기 때문이다. 이에 대한 예로는 대량의 유전자 정보 처리, 비행체나 발사체 설계, 기후나 환경변화, 고

에너지 물리분야의 데이터분석에 관한 연구이다. 이상의 4개의 기술분야에 대한 그리드 어플리케이션의 예를 살펴보면 다음과 같다.

- 유전자 정보처리[14]

고속의 자동화된 DNA 염기서열 기계와 정밀한 단백질 구조결정 방법이 보편화되면서 생물학 분야에서 쏟아져 나오는 데이터의 양이 기하급수적으로 증가하고 있다. 최근 들어 인간을 포함하여 많은 종류의 생명체의 염기서열이 결정되었고, 이를 해석하기 위하여 점점 더 큰 데이터베이스가 필요해졌다. 생물학 관련 정보는 전 세계에 분포된 DB에 분산되어 있는 것이 특징이다. 현재 사용자, 자원이 지리적으로 분산되어 있고, 데이터베이스의 크기, 이용 가능한 네트워크의 제한된 대역폭 때문에 자원의 공유가 어렵고 분산된 DB들의 일관된 포맷의 유지, 데이터의 접근성과 보안성의 보장, 효율적인 해석을 위한 통일된 작업환경이 필요하다. 그리드의 구축은 연결된 자원들을 하나로 통합시켜주고 데이터의 포맷을 일관성 있게 유지해 주기 때문에 포스트게놈 연구에서 쏟아져 나올 엄청난 데이터까지 감당 할 수 있을 것으로 기대된다.

- 나노기술

마이크론(100만 분의 1)기술이 서서히 한계성을 드러내면서 물질의 기본 단위인 원자 및 분자 조작을 통한 기술적 한계 극복은 물론 지금까지 볼 수 없었던 신물질을 만들거나 기존 제품을 고성능화 할 수 있는 나노(10억 분의 1)기술이 크게 발전하고 있다. 나노기술 연구를 위해 필요한 실험 장비의 가격은 천문학적으로 증가하고 있어 문제 해결의 장애물로 인식되고 있다. 이를 해결하기 위한 대안 중의 하나가 고가의 대형 장비를 그리드 기반에서 공동 활용하거나, 실험으로 해석이 어려운 문제에 대해서는 초대형 컴퓨터를 활용하는 것이다. 따라서 그리드를 활용하면 단기간에 매우 많은 결과를 얻어낼 수 있을 것으로 기대된다.

- 기계설계 및 항공기설계[13]

NASA에서는 지난 2000년에 산하 3개 연구기관인 Langley, Glenn, Ames에 있는 SGI Origin 2000 시스템을 활용하여 비행체설계에서 가장 중요한 유체역학 어플리케이션이 그리드 컴퓨팅에 적합한 지를 테스트하였고 훌륭한 결과를 얻었다. 단일 시스템 내에서 수행 한 결과와 그리드 환경에서의 결과를 비교했을 때, 데이터 교환에 소요된 시간은 그리드 환경이

10배정도 느렸지만 전체 계산시간은 단일 시스템에 비해 비슷한 효과를 얻을 수 있었다. 이것은 그리드 환경이 초고속 네트워크의 힘을 입어 단일 시스템의 성능을 낼 수 있다는 것이다. 초고속 네트워크 상에 많은 사용 가능한 자원들이 연결될 때 기존에 수행하지 못하던 슈퍼컴퓨팅 어플리케이션을 수행 할 수 있게 된다. 초고속 네트워크와 많은 컴퓨팅 자원, 그리고 거대한 저장장치, 몰입형 가상화 시스템들을 기반으로 자동차설계, 차세대 엔진개발, 비행체설계 및 실험 등을 거대규모로 할 수 있을 것이다.

- 기상 예측 실험[12]

기상청 발표자료에 따르면 과거 10년 기간 동안 기상 재해로 인한 직접적인 재산 피해액은 연평균 3,800억원, 인명 피해는 280여명으로 보고되고 있으며, 경제규모 및 인구의 증가에 따라 피해는 지속적으로 증가할 것으로 내다보고 있다. 이러한 기상재해를 줄이기 위해 날씨예측의 정확성, 적시성, 가용성을 향상시키기 위해 전 세계 기상청들은 부단한 노력을 경주하고 있다. 사용자가 원하는 장소와 시간에 대해 즉각적인 날씨정보의 제공을 의미하는 국지기상 예측(Local Weather Forecast)은 초고속 네트워크, 고성능 컴퓨터, 예보모델, 모델 입력자료에 관한 DB 등이 결합된 기반 위에서 실현될 수 있다. 이러한 관점에서 초단기 국지 날씨 예측을 위한 수치모델링은 그리드를 활용하여 가시적인 효과를 거둘 수 있을 것이다.

- 고에너지 물리분야[26]

유럽연합은 고에너지 물리학과 나노기술 분야 등을 위하여 스위스 CERN에 Large Hadron Collider (LHC)를 2005년 가동 목표로 건설 중에 있다. LHC는 유럽에서 공동으로 사용할 가속기로써 양자들이 초당 4천만회 충돌하고, 그 결과 생성되는 데이터를 처리하기 위해서는 최신형PC 10만대를 동시에 활용해야 할 정도의 계산능력을 요구하며 일년동안 페타바이트급으로 쏟아져 나오는 데이터는 전 세계의 고에너지 물리학자들에게 필요로 하기 때문에 분산저장 및 관리해야 한다.

2.2 그리드의 분류

그리드의 분류는 대상 어플리케이션에 따른 수평적 분류와 사용자 계층에 따른 수직적인 분류가 있다. 먼저 수평적인 분류에서는 어플리케이션의 특징

을 고려해야 한다. 많은 양의 계산을 수행하는 어플리케이션인지, 데이터 전송이 빈번한 어플리케이션인지, 다루는 데이터의 양이 어떤지, 여러 다른 분야 전문가들이 협업을 해야 하는지 등에 따라 서로 다르다. 아래의 분류표는 목표에 따라 그리드 개발분야를 나누고 있다(6).

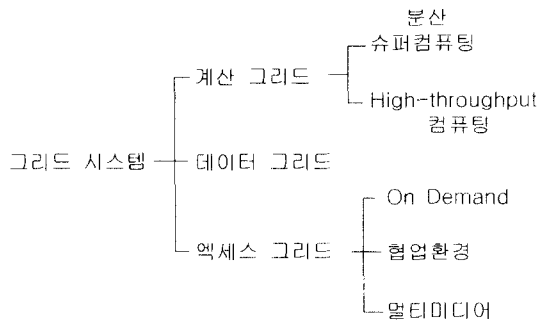


그림 1 그리드의 수평적 분류

계산 그리드에 속하는 시스템들은 컴퓨팅을 위해서 많은 자원들을 연결하여 해결할 수 있도록 하는 것을 말한다. 이들 중에서도 분산 슈퍼컴퓨팅 관련 그리드는 작업의 전체수행시간을 줄이기 위해서 많은 컴퓨터들을 동시에 이용하는 것을 말한다. 기계항공 분야나 물리 분야의 초대형 문제들을 주로 해결하기 위해서 사용된다. 반면에 high-throughput 컴퓨팅은 주어진 시간 안에 가능한 많은 작업을 처리하는 것이 목표인 그리드이다. 프로세스 설계와 같이 많은 다양성을 가진 작업을 처리하기에 적합한 그리드이다.

두 번째로 데이터 그리드는 원격지에 분산된 자료들을 통합하여 분석할 수 있게 해주기 위한 그리드이다. 대표적인 사용 분야는 자료가 분산된 여러 곳에서 생성되고 한 곳에서 처리하는 형태의 데이터 마이닝이나 분산된 대용량의 데이터를 여러 곳에서 처리해야 하는 고에너지 물리, 유전자 정보 처리분야 등이다.

액세스 그리드는 본질상 분산처리를 필요로 하는 어플리케이션을 위한 그리드이다. 이도 역시 사용분야에 따라 나눌 수 있는데, On Demand 그리드는 사용자 작업이 요구에 따라 동적으로 자원들을 연결하는 것이 가능한 그리드이다. 사용자는 작업 처리 중에 결과를 좀더 정밀하게 보기 위해 더 많은 자원들

요구할 수 있다. 협업환경 그리드는 사용자가 원격지에서 협력하며 일을 할 수 있게 하는 하나의 작업 공간을 제공한다. 멀티미디어 그리드는 실시간 멀티미디어 어플리케이션을 위한 인프라를 제공한다. 멀티미디어 그리드는 다양한 스트림이 동시에 교환되기 때문에 네트워크 QoS가 반드시 보장되어야 한다.

두 번째 분류는 어플리케이션 분야에 따른 그리드의 분류가 아니라 사용자에게 보이는 지식정보의 형태에 따라 그리드를 분류해 볼 수도 있다. 그리드의 가장 단순한 형태로는 고성능 컴퓨팅 자원들과 대용량 저장 장치들을 초고속 네트워크로 연결하여 사용자에게 제공하는 것이다. 반면에, 사용자에게 제공되는 지식정보가 고급화된 그리드는 사용자가 마치 집에서 가전제품을 사용할 때 전력을 사용하듯이 그리드를 사용할 수 있는 환경을 제공하는 것이다(6).

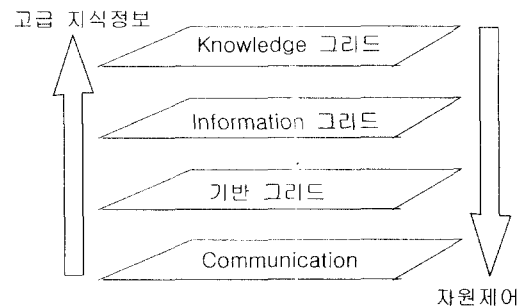


그림 2 그리드의 수직적 분류

- 기반 그리드 : 컴퓨팅 자원, 저장장치, 초고속 네트워크를 그대로 제공하는 그리드이다. 사용자는 그리드 관리자이거나 Information 그리드 개발자이다.
- Information 그리드 : 정보에 대한 위치, 분석을 위한 도구, 결과 가시화 등을 쉽게 할 수 있는 환경을 제공하는 그리드이다. 분산 어플리케이션 개발 및 시뮬레이션을 통해 연구를 수행하는 과학기술 분야 종사자가 사용한다.
- Knowledge 그리드 : 데이터 마이닝, 머신러닝과 같은 기술을 이용하여 사용자에게 지능적인 정보를 제공하는 그리드이다. 사용자 계층은 분산 처리에 관한 지식이 없지만 그리드에서 제공되는 정보를 활용하기 원하는 모든 사용자이다.

현재 전 세계적으로 그리드 개발을 위한 과제가 국가 슈퍼컴퓨팅 센터들을 중심으로 컨소시엄을 구

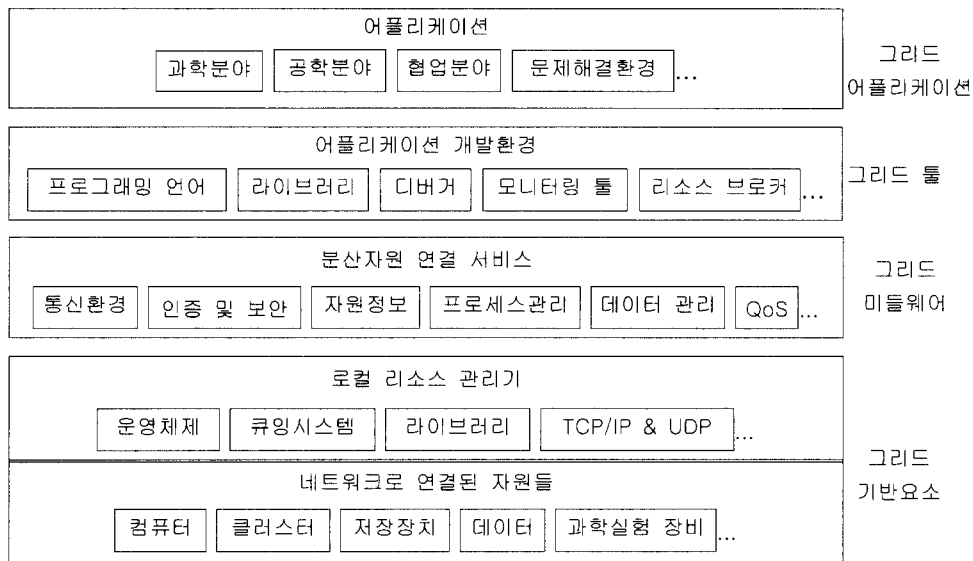


그림 3 그리드 구조

축하여 진행되고 있는데, 대부분 계산 그리드와 데이터 그리드를 결합한 하이브리드 형태를 시도하고 있으며 수직적인 분류에서 본다면 기반 그리드와 Information 그리드에 초점이 맞춰져 있다.

2.3 그리드 구조와 구축원칙

그리드를 만드는 것은 많은 서비스를 개발하여 제공하는 것이다. 이들 서비스들은 유휴 자원들을 찾아내는 자원 검색 서비스, 작업들의 분산 및 처리 순서를 결정해 주는 스케줄링 서비스, 시스템 안정을 위한 그리드 보안 서비스, 컴퓨팅 자원을 사용함으로써 발생하는 비용 처리를 위한 사용자 서비스 등이 있다. 그리드는 단지 초대형 문제를 해결하기 위해 계산 자원들을 제공하는 것만은 아니다. 이것은 고성능 실험장비, 관측장비, 슈퍼컴퓨터 뿐만 아니라 개인용 컴퓨터까지 다양한 자원들을 전 세계적으로 연결하는 인프라이다. 이와 같이 그리드가 확장되어 나간다면 이들 자원들을 필요로 하는 모든 사용자들에게 보편적인 서비스로 제공될 것이다. 계산 그리드를 구축하는 데는 세 가지 주요한 특징이 있다[15].

1. 다양성 : 여러 기관에 흩어져 있는 서로 다른 자원들은 다양한 형태로 존재한다. 컴퓨터 구조가 다르고 기종이 다르고 운영체제 등이 서로 다를 수 있다. 그리드는 이런 다양한 이질적인 면을 극복할 수 있어

야 한다.

2. 확장성 : 자원들이 그리드에 연결될 때 수백만 개로 늘어날 수 있다. 그리드의 사이즈가 커질수록 부하 균등화를 잘못하거나 네트워크 문제로 인하여 성능이 나빠질 수도 있는데 이를 해결할 수 있어야 한다. 물론, 그리드 환경에서 어플리케이션들은 지역적으로 떨어져 있는 자원들을 효율적으로 동시에 사용하기 위해 설계되어야 한다.

3. 고가용성 : 그리드 환경에서는 연결된 자원들이 아주 많기 때문에 자원들 중 일부는 사용 중에 고장으로 인해 사용불가 상태로 들어갈 수도 있다. 이를 위해 자원관리기나 어플리케이션은 사용 가능한 자원에서 최대의 성능을 얻기 위해 동적으로 환경변화에 대처하여야 한다.

이와 같은 특징들을 고려하여 전 세계적으로 그리드 구축을 위한 움직임들이 있는데 구축중인 그리드의 일반적인 구조는 계층적인 구조로써 다음 그림 3과 같이 나타낼 수 있다[15].

- 그리드 기반요소 : 개별적인 소프트웨어와 하드웨어들을 하나의 통합된 자원으로 연결시키는 작업이 필요하다. 이들 자원들은 PC들과 UNIX, NT와 같은 운영체제 기반의 컴퓨터들과 LSF, PBS 등으로 운용되는 클러스터, 대용량 저장장치, 데이터베이스, 전파망원경과 같은 고성능 연구장비 등을 포함한다.

- 그리드 미들웨어 : 사용 가능한 자원들을 사용자에게 하나의 시스템처럼 보이도록 하는 미들웨어가 구현되어야 한다. 미들웨어는 프로세스의 관리, 자원들의 동시 사용, 저장장치의 접근, 정보 보안, 사용자 인증, 네트워크 QoS, 자원예약 등과 같은 핵심 서비스들을 제공한다. 사실 미들웨어 구축이 가장 어렵고 핵심이라 할 수 있다.

- 그리드 개발환경 : 그리드에서 수행되는 어플리케이션과 인프라를 관리하기 위한 도구들을 개발 또는 기존에 있는 것들을 지원해야 한다. 프로그래머가 고성능 어플리케이션을 개발할 수 있게 해주는 서비스와 전체적인 자원 상에서 수행될 계산들을 관리하고 계획하는 사용자 agent들을 제공한다.

- 그리드 어플리케이션 : 분산된 자원들을 효율적으로 활용하는 어플리케이션 개발이 필요하다. 이들 어플리케이션들은 HPC와 같은 프로그래밍 언어나 MPI와 같은 메시지패싱 라이브러리를 이용하여 개발된다. 시뮬레이션이나 거대문제와 같은 어플리케이션들은 엄청난 계산능력이 필요하고 원격지에 있는 데이터를 쉽게 접근할 수 있어야 하고 고성능 실험장비들과 연동 할 수 있어야 한다.

이와 같은 계층 구조로 그리드를 구성하는 주된 이유들 중에 하나는 각 사이트에서 보유하고 있는 기존의 컴퓨팅 자원들을 현재 상태 그대로 그리드에 연동시키기 위해서이다. 이들 여러 기관들을 연결하는 그리드를 형성하기 위해서는 다음과 같은 여러 기본적인 원칙이 지켜져야 한다.

- 그리드 환경은 각 사이트의 자치권을 침해해서는 안 된다.
- 이미 사용중인 운영체제나 네트워크 프로토콜을 변경할 필요가 없어야 한다.
- 각 사이트는 자신의 선택에 따라 그리드에 가입과 탈퇴가 자유로워야 한다.
- 프로그래밍 패러다임을 제한한다거나 사용하는 언어나 도구들을 제한해서도 안 된다.
- 한군데에서 시스템 오류가 발생했다 하더라도 그리드 전체적으로는 별 문제 없이 운용되어야 한다.
- 이기종 자원들을 지원해야 한다.
- 표준과 기존의 기술들을 사용할 수 있어야 하고 애러가 없는 어플리케이션이라면 문제없이 수행되어야 한다.

3. 그리드 관련 툴킷

3.1 Globus Toolkit 소개

Globus toolkit은 그리드 서비스를 제공하는 미들웨어로서 전세계적인 그리드 개발 과제에서 가장 많이 사용되고 있다(National Technology Grid, European DataGrid, NASA Information Power Grid, Grid Physics Network, ASCI Distributed Resource Management (DRM) Testbed, GUSTO,... 등). 이렇게 Globus toolkit이 널리 사용되게 된 이유는 globus toolkit이 분리될 수 없는 단일 시스템이 아니다. 그리드에서 필요로 하는 다양한 서비스들을 독립적인 요소로써 제안하고 있기 때문이다. globus의 또 다른 장점으로는 기존에 각 시스템 및 네트워크의 관리 정책이나 운영 도구들을 무시하지 않고 각 요소들과 협력하여 그리드를 이루어 나간다는 점이다. 따라서, globus toolkit의 요소들도 하드웨어 측면이나 소프트웨어 측면에서 상이한 시스템들간에 성능 저하를 줄이면서 통합하기 위한 기능을 위한 것이 대부분이다. Globus toolkit은 크게 그리드 보안, 정보 서비스, 자원 관리, 데이터 관리 등으로 나뉘어진다(2, 4, 6).

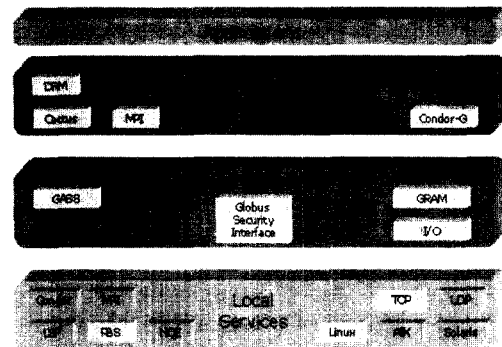


그림 4 Globus Toolkit의 구조

Globus toolkit에서는 보안을 담당하는 부분을 GSI라고 부르며, 그리드 보안은 분산 자원들을 공유함에 따라 발생하는 자연스러운 문제이다. 사용자의 입장에서는 안전하면서도 사용의 편리성을 요구할 것이고, 각 자원을 소유하고 관리하는 관리자의 입장에서는 자원이 그리드 환경에 노출되는 것이기 때문에 사용의 편리성보다는 더 안전한 보안을 원할 것이다. 이를 위해 GSI는 single-sign-on을 제공하고 globus proxy를 이용한다. 사용자는 그리드환경에

한번의 인증과정을 거침으로 사용이 허용된 자원들을 사용할 수 있고 분산된 각 자원에 대한 사용자 인증은 proxy가 대신 수행한다. 그렇지만 각 자원 내에서 자원에 대한 사용에 대한 허용범위는 각 자원이 제시하는 보안체계를 따른다[5].

두 번째로, globus toolkit에서 정보 서비스를 수행하는 요소를 MDS라고 부른다. MDS는 그리드 내에 존재하는 자원들의 상태정보를 공유하고 사용자들에게 제공하기 위한 요소로써 인터넷의 DNS와 비슷한 것이다. 정보를 저장하고 사용자들에게 제공하기 위해 MDS는 LDAP를 이용한다. 정보 서비스를 위해 Globus에서는 두 개의 서버를 제공하는데, 각 자원의 정보를 수집하는 GRIS와 수집된 정보를 통합하는 GIS이다. 이들이 수집하여 제공하는 정보는 각 자원의 구조, 노드 수, 부하 정보, 배치작업 스케줄러, 네트워크 상태,... 등이다. 이들 정보는 어플리케이션 개발자나, resource broker,..등에 제공된다[7].

세 번째로, globus toolkit에서 자원 관리를 담당하는 부분을 GRAM이라 부른다. GRAM은 globus toolkit의 가장 중심이 되는 요소로써 원격지 자원을 사용할 수 있게 하고 분산 자원들을 동시에 사용하게 하며 자원들의 관리의 상이함을 처리한다. 사용자는 자신의 작업을 그리드 환경에서 처리할 때 원하는 요구사항을 RSL이라는 스크립트를 이용하여 표현한다. GRAM은 resource broker를 이용하여 RSL 스크립트를 저급의 스크립트로 변환하고 각 자원에 있는 스케줄러가 처리할 수 있는 형태의 스크립트로 변환한다. 이 변환과정에서 resource broker는 현재 또는 미래에 사용 가능한 자원을 검색하기 위해 MDS를 이용하게 된다. 작업이 분산환경에 할당되면 각 작업의 협업을 위해 GRAM은 DUROC이라는 요소를 이용한다.

마지막으로 globus에서는 데이터 관리를 위해 GASS, GridFTP, Replica catalog를 제공한다. GASS는 GRAM과 밀접한 관련이 있는 요소로써 원격지에 있는 파일을 사용하여 작업을 처리하기 원하거나 원격지에서 처리한 작업의 결과를 또 다른 저장장치에 저장하고 싶을 때 사용한다. GridFTP는 그리드 내의 데이터가 대규모 대용량이란 점을 고려하여 고속으로 파일 전송과 파일의 이어받기를 가능케 하는 요소이다. Replica catalog는 데이터 그리드를 위해 개발된 것으로 데이터들을 분산 저장 및 관리함으로써 필요할 때에 신속하게 데이터를 사용할 수 있게 하는 기술이다[8].

3.2 Legion 소개

Legion은 다양한 컴퓨팅 자원과 네트워크를 연결하여 단일 시스템과 같은 모양을 내는 객체지향 메타 시스템이다. Legion은 객체지향 방법의 미들웨어라는 면에서 CORBA와 비슷하지만 CORBA와는 달리 고성능컴퓨팅을 필요로 하는 사용자들을 위해 개발되었다[1, 21].

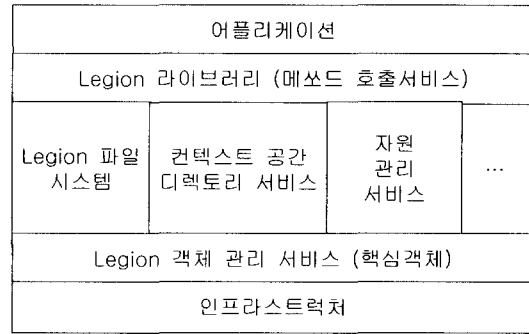


그림 5 Legion 아키텍처의 개요

Legion은 사용자 OS 상에 위치하며 컴퓨팅 자원들과 어플리케이션들 사이를 중재한다. Legion은 자원의 스케줄링을 처리하고 보안 문제를 다루고 있으며, 컨텍스트 공간이라는 네이밍 시스템이 사용되고 있다. 컨텍스트 공간은 사용자가 쉽게 객체들을 모니터링하고 사용할 수 있게 해 준다. Legion에서의 컴퓨팅 자원들과 스토리지 자원을 모두 객체로 표현하는데 호스트 객체, 볼트 객체 등으로 표현한다. Legion의 IDL은 Java, C, C++, Fortran, CORBA IDL로 변환 가능하다.

여러 자원을 사용할 때 발생하는 문제 중 가장 중요한 것은 인증과 권한부여에 관한 문제이다. 일반적으로 각 사이트는 인증에 관한 자신만의 정책을 가지고 있다. 그런 정책은 SSII나 Kerberos와 같은 보안 서비스를 이용하거나 단순한 패스워드를 이용한다. Legion 객체들은 자신만의 보안정책을 구현할 수 있어야 하고 다음 특징을 가진다

- 인증 : Kerberos v5, X.509인증서를 이용하는 PKI기반 보안
- MayI : Legion 객체에 접근할 수 있는 자신의 메소드를 정의하기 위한 접근 제어 모듈
- RSAREF 2.0 라이브러리에 기반을 둔 3개의 메

시지 계층 보안(private, protected, no security) 사용자가 그리드 상에서 자원들에 관한 사용 권한을 부여받았다고 가정하면 공유되고 있는 하드웨어/소프트웨어 자원들을 사용하기 위한 것이다. Legion은 자원 상에서 사용자 코드를 실행시킬 때 다음 3단계를 거친다.

- (1) Legion 라이브러리와 헤더파일을 가지고 코드를 컴파일
- (2) legion_register_program을 이용해서 코드를 등록(MPI코드는 legion_mpi_register)
- (3) legion_run을 이용해서 프로그램 실행(MPI코드는 legion_mpi_run)

Legion은 LoadLeveler, Codine, PBS, NQS와 같은 로컬 스케줄링 시스템을 지원한다. Legion 객체를 다루기 위해서 legion_cp, legion_mv 등 유닉스와 비슷한 명령어를 이용해서 파일들을 복사 또는 이동할 수 있다. Legion하에서 객체들 사이의 통신은 네트워크 연결메소드 호출에 의해 수행된다. 메소드 호출의 허락은 스케줄링과 보안 정책에 달려있고 네이밍은 컨텍스트 네임, LOID(Legion Object Identifiers), LOA(Legion Object Address) 등 3단계 네이밍 시스템을 이용한다. Legion 객체는 하나의 공유 컨텍스트 공간에서 유지되고 컨텍스트 네임에 의해 불리어 진다. Context 이름들은 RSA Security로부터 받은 공개키를 포함한 LOID로 매핑 된다. LOID들은 위치와 상관없는 정보이기 때문에 통신을 위해 LOA에 매핑 된다. 하나의 LOA는 객체들과 통신하도록 IP주소와 포트번호와 같은 정보를 포함하는 물리적인 주소이다. 따라서, 병렬처리에서 일반적인 MPI 프로세스들과 PVM 태스크들은 Legion 객체로 표현된다.

Legion은 SPMD 스타일 어플리케이션을 위한 체크포인팅 라이브러리를 제공하고 있다. SPMD 어플리케이션에서 각 태스크는 데이터의 일정 부분에 대한 책임이 있고 이웃하는 태스크들과 데이터를 주고 받는다. 라이브러리 루틴들은 문제가 발생해서 재시작 하게 된다면 필요한 데이터를 주기적으로 저장하는 체크포인트 파일을 만들기 위해 사용될 수 있다. legion_mpi_run을 이용해서 프로그램을 수행시킬 때 사용되는 특별한 옵션들은 체크포인팅을 활성화시키고 문제 발생 후 어플리케이션은 언제 재시작 되어 하는지를 기술할 수 있다.

3.3 Condor 소개

많은 과학자들에게 있어서, 그들의 연구의 수준은 컴퓨팅 능력에 따라 많이 좌우된다. 보통 몇 주씩 걸리는 몇 달씩 걸리는 계산을 해야 하는 분야들은 컴퓨팅 파워들을 배분하는 High Throughput Computing 환경이 필요하다. 따라서, 컴퓨팅 성능을 표시하는 FLOPS도 초당 연산횟수보다 달 또는 년 당 수행하는 연산횟수 개념으로 확대되고 있다. 이와 같은 성능을 얻기 위해서는 사용 가능한 컴퓨팅 자원들을 어떻게 효율적으로 활용하는가에 달려있다. 몇 년전만 해도 계산작업은 메인프레임 컴퓨터에 의존했다. 요즘 많은 연구기관 및 연구실에는 사용 가능한 개인용 고성능 컴퓨팅 자원들이 많이 존재한다. 이들 자원들은 많은 시간동안 사용하지 않은 채 존재한다. Condor는 이들 버려지는 컴퓨팅 능력을 모아서 사용하게 하는 것이다[5].

Condor는 분산된 워크스테이션들을 통합하여 사용할 수 있게 하는 High Throughput Computing 환경이다. Condor의 개발은 과학기술분야의 계산과학 분야 전문가들이 유휴 컴퓨팅 자원들을 통합하여 사용하고자 하는 필요에 의해 시작되었다. 현재 대부분의 유닉스 시스템 상에 설치가 가능하고, NT 시스템을 위해서는 개발 중에 있다. Condor 환경에서 High Throughput Computing의 성공은 각 컴퓨팅 자원을 소유한 소유주에 달려있다. 각 소유주는 자신의 컴퓨팅 자원을 어떤 상황일 때 외부에 할당할 지를 결정할 수 있다. 예를 들어, 소유주가 조금이라도 사용하면 외부사용자가 사용할 수 없는 엄격한 조건을 달 수도 있고 어느 정도의 부하까지는 허용하는 조건을 달 수도 있다.

Condor의 중요한 특징은 ClassAd 기법에 있다. 각 사용자는 자신의 작업을 수행하기 위해 적합한 컴퓨팅 자원을 찾기 원한다. ClassAd는 각 컴퓨팅 자원에 대해 프로세서 타입, 메모리 양, 운영체제 타입 등 모든 가능한 특징에 관한 정보이다. 또한, 사용자는 자신의 작업이 부동소수점 연산에 있어 나은 컴퓨터에서 수행되길 원하거나 특정한 컴퓨터에서 수행되기를 원할 수 있는데 이와 같은 모든 것도 ClassAd에서 기술할 수 있다. 사용자는 자신의 작업을 위한 ClassAd를 기술한다. 그러면, Condor는 사용자 작업에 관한 ClassAd와 각 컴퓨팅 자원의 ClassAd를 일치시킴으로 적당한 자원들을 찾는다. 각 컴퓨팅 자원 관리자는 자신의 자원에 대해 다른 사용자의 접근을 명시함으로써 제한할 수 있다. 특정 시간에만 사용 가

능하게 한다거나 특정한 사용자 그룹에만 사용을 허용한다거나, 어떤 작업을 더 선호하는지도 기술 할 수 있다.

3.4 Cactus 소개

Cactus는 그리드 환경에서 과학기술분야 연구를 위한 문제해결환경으로 특징은 구조가 모듈화 되어 있다는 것이다. Cactus라는 이름은 구조가 마치 선인장의 모양과 유사하다는데서 따온 것이다. 선인장이 가운데 줄기가 있고 줄기에 많은 가시가 박혀있듯이 Cactus는 가운데 줄기의 역할을 하고 어플리케이션 프로그램이나, 다른 도구들은 가지처럼 Cactus에 연결되어 사용될 수 있다는 것이다. 예를 들어, 가지처럼 붙을 수 있는 것들은 사용자가 작성한 유체역학해석 프로그램, 체크포인팅을 위한 도구, 병렬 입출력을 위한 도구, 가시화 도구... 등이다[16].

Cactus는 단일 시스템뿐만 아니라 클러스터나, 슈퍼컴퓨터 등 다양한 플랫폼에 구축될 수 있으며 현재 어플리케이션 사용자들이 사용하는 도구들을 쉽게 연결하여 사용할 수 있는 환경을 제공한다. 예를 들면, Globus toolkit, HDF5 병렬 파일I/O, PETSc 과학 라이브러리, 가시화 도구들 등이다.

4. 그리드 관련 연구

4.1 표준화 작업

1995년 I-WAY의 시연을 전후로 많은 그리드 미들웨어 개발과제가 진행되고 있는데, 독자적으로 이들을 개발하다가 서로 통일된 인터페이스의 필요성을 느끼게 되었다. 양쪽 도메인에서 서로 다른 툴킷을 이용하여 그리드를 구축하면 각 도메인 내에서는 사용이 가능하겠지만 두 도메인간 상호운용은 불가능하게 된다. 이를 위해 표준화 작업을 위한 포럼이 국내외적으로 만들어지기 시작했다[19].

- Global Grid Forum : 미국을 중심으로 분산 컴퓨팅과 관련된 일을 하는 사람들과 연구를 수행하는 사람들이 모여서 결성된 포럼으로 그리드 기술의 개발과 그리드 기술의 촉진을 위해 힘쓰고 있다. 현재 가장 크고 영향력 있는 포럼이다.
- Computing Portals : 슈퍼컴퓨터, NOW, 실험장비들, 데이터 자원들과 같은 자원들을 자신의 컴퓨터를 통하여 사용하게 하고 여러 전산관련

프로젝트들 사이에 협력을 하고자 하는 모임이다.

- European Grid Forum : 유럽을 중심으로 구성된 EGRID라고 불리는 이 모임은 WAN환경에서 사용 가능한 분산 컴퓨팅 자원들을 문제해결을 위해 사용하고자 하고 있다. 이 공동체는 유럽의 연구 기관들과 대학 및 분산 컴퓨팅과 관련된 회사들을 포함하는 열린 모임이다.
- Grid Forum Korea : 국내 그리드 기술개발을 위한 모임으로 2001년 10월에 발족되었다. 슈퍼컴퓨터, 고성능 클러스터, 데이터 자원들을 연결하여 국내 그리드 인프라를 구축하기 위해 결성하였고 국내의 대학들과 연구소 업체가 중심이 되어 15개의 워킹그룹 중심으로 활동하는 열린 모임이다[20].

4.2 그리드 구축 과제

2001년 10월 미국 보스턴에서 Global Grid Forum 기간 중, 미국, 유럽, 아시아 지역을 중심으로 현재 추진중인 주요 프로젝트들이 소개되었는데, 이는 실제로 개발된 Grid 기술이나 응용을 위한 프로젝트로서 볼 수 있다. 현재 전 세계적으로 그리드 개발을 위한 과제가 국가 슈퍼컴퓨팅센터들을 중심으로 컨소시엄을 구축하여 진행되고 있는데, 대부분 계산 그리드와 데이터 그리드를 결합한 하이브리드 형태이다. 진행되고 있는 그리드 개발 사업을 보면 다음과 같다[19].

- European Data Grid : 스위스의 CERN과 고에너지 물리학자들이 중심이 되어서 구축중인 그리드로써 일본의 TACC에서 공동연구로 참여중이고 참여 국가는 스위스, 이태리, 프랑스, 영국 등 약 10개 국가가 참여하고 있다. 특히 국제적인 테스트베드 인프라가 개발될 예정이다. 프로젝트 시작은 2001년부터이고 3년 동안 개발할 예정이다. Globus를 11개 사이트에 35개 시스템에 설치할 예정이며, 이를 개발하기 위하여 GRAM, RSL 등을 시험중이다.
- 독일의 CACTUS Grid Computing : Cactus Code는 그리드 어플리케이션을 위한 미들웨어로써 웹 기반의 실시간 가시화가 가능한 기술이다. 독일은 이를 발전시켜 과학자들이 인터넷에서 협업 연구가 가능한 새로운 코드를 개발하여 그리드를 구축할 예정이며 DFN Gigabit 프로

젝트에 적용할 예정이다.

NASA IPG : NASA의 Multi-component simulation을 위한 프로젝트로 Grid common 서비스를 제공할 예정이며, 미국 내 항공기 개발과 관련된 사이트가 Globus기반으로 Grid Computing 환경으로 구축될 예정이며, 현재 ARC, CML, laRC가 IPG/NREN QoS Testbed로 구축되어있다. Grid 기반의 Large Scale의 분산 어플리케이션 응용이 가능한 기술을 개발할 예정이다.

SDSC의 NPACI Grids/Metasystem : 샌디애고 슈퍼컴퓨터센터 중심으로 NPACI 동맹기관이 참여하는 프로젝트로 데이터 처리 중심 컴퓨팅, Interaction Environments, Programming Tools Environments 분야에서 개발 중이며, SDSC, Texas, Michigan, UVA가 중심이 되어 4개 시스템을 중심으로 Globus기반의 메타 컴퓨팅 시스템을 구축하고 있다.

ASCI Grid 프로젝트 : SNL, LANL, LLNL 중심으로 ASCI 프로젝트를 성공적으로 수행하기 위하여 최근에 구축된 프로젝트로 3개 연구실에서 이음새 없이 분산된 슈퍼컴퓨팅 자원을 이용할 수 있도록 메타컴퓨팅 시스템을 구축하는 것으로 2010년까지 개발할 예정이다.

ApGrid 프로젝트 : 일본이 중심이 되어 APAN 프로젝트를 중심으로 Globus 기반의 분산컴퓨팅, 디지털 라이브러리, 원격교육 등을 구축하기 위하여 추진중인 프로젝트이다.

DisCom2 : 원거리에 있는 고성능 컴퓨팅 자원들을 효과적으로 연동하기 위한 기술들을 개발하기 위한 과제이다. 이 과제의 목표는 몇 가지로 요약해 볼 수 있는데, 첫 번째는 자원들을 효율적으로 사용할 수 있게 하는 인프라 스트럭처를 개발하는 것이다. 두 번째 목표는 공학분야의 계산 모델링과 시뮬레이션을 지원하는 것이다. 이들 목표를 위해서 여러 개의 작은 과제들로 나누어서 수행하고 있다.

Alliance의 National Technology Grid : 인프라 스트럭처 중심의 계산 그리드와 사용자 중심의 액세스 그리드로 구성된 분산 및 협업환경을 구축하는 것이 본 연구의 목적이다.

N*Grid: 계산 그리드와 데이터 그리드를 위한 미들웨어 개발과 BT, NT, ST분야의 그리드 어

플리케이션을 개발하는 과제로써 과학기술정보 연구원이 주축이 되어 2002년부터 진행할 계획을 세우고 있다.

4.3 NASA IPG 소개

4.2절에서 전 세계적으로 추진되고 있는 그리드 구축 과제를 간단하게 살펴보았다. 각 응용분야에 맞게 툴킷을 선정하고 필요한 부분을 정의하며 새로운 툴킷들을 만들어 나가고 있다. 본 절에서는 그리드 구축과제들 중 NASA에서 추진중인 IPG의 구축전략을 소개한다[23].

IPG를 구축하기 위한 전략은 일단 첫 번째는 계속되는 요구분석이다. 요구분석 작업은 어플리케이션 사용자들이 요구하는 요구사항을 그리드의 서비스로 일반화시키기 위한 것이다. 다음으로, 그리드 서비스들 중 이미 구현되어 있는 것들과 개발해야 할 것들을 구분한다. 그리고, 그리드 서비스의 타당성을 위해 대표적인 어플리케이션 작업을 구성한다. 마지막으로, 요구되는 서비스와 인프라를 정의 및 구현해야 한다. 다음 그림 6은 어플리케이션 분야의 요구사항을 분석하고 기술의 방향을 바탕으로 IPG의 서비스 구조를 나타낸 것이다. IPG에서는 미들웨어 서비스를 위한 시스템으로 globus를 선정하였다. 그림에서 색칠 된 부분이 globus에서 이미 구현된 부분이다. 그러나, 아직도 IPG가 필요로 하는 기능을 globus가 모두가지고 있는 것이 아니기 때문에 체크포인트/리스타트 기능, 접근 제어기능, 글로벌 파일 시스템, 원격 입출력 기능을 갖춘 그리드 통신 라이브러리 등을 포함해야 한다.

그리드 서비스를 이용할 사용자들은 여러 계층으로 나뉘어 볼 수 있고 이들 그룹들은 그리드에 대해서도 다른 요구사항들이 있다.

- 일반 사용자들은 가장 익숙한 웹브라우저와 같은 형태의 인터페이스를 이용하여 그리드에 접근하기를 원한다.
- 어플리케이션 영역의 연구인력들은 문제해결을 위한 특정 응용분야에 맞는 도구들을 이용하기를 원한다.
- 어플리케이션 영역 중에서 전산학 분야의 사용자들은 분산 컴퓨팅 및 데이터들을 통합하기 위해 다양한 프로그래밍 패러다임을 위한 프로그램 개발환경을 직접 사용하기를 원한다.

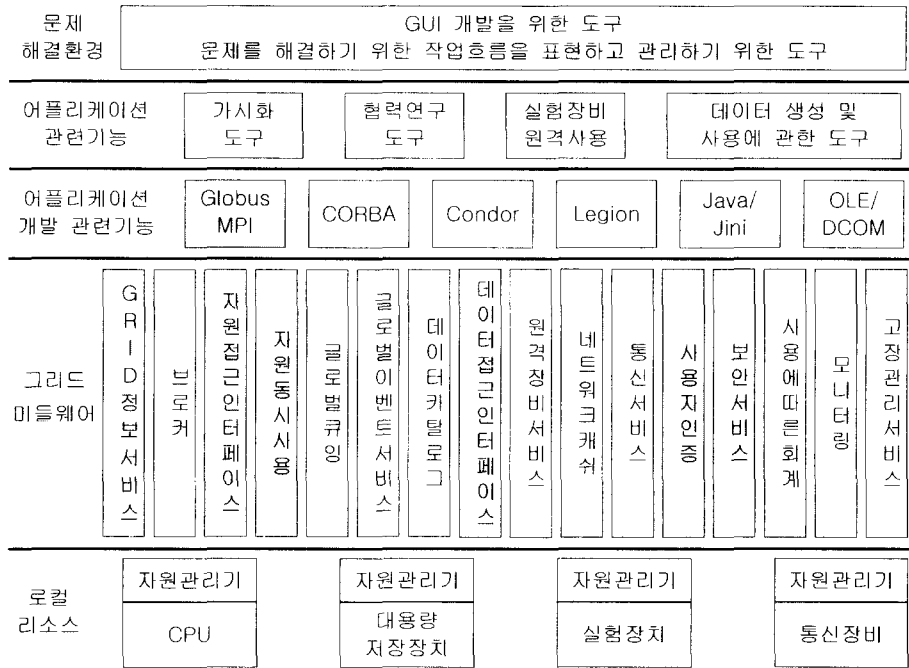


그림 6 IPG의 서비스 구조

- 분산 시스템 개발자들은 작업관리, 접근제어, 일 반화된 통신서비스, 자원 검색 등을 이용하기 원 한다.
 - 그리드의 미들웨어 개발자들은 각 자원에 대한 자원관리기를 이용한다.
- 어플리케이션 분야의 요구사항을 분석하고 현재 기술을 바탕으로 IPG가 제공할 기능을 살펴보면 다음과 같다.
- (1) 문제해결환경
 - 그리드의 최상위 계층에 속하며 각 어플리케이션 분야에 밀접한 부분이다.
 - 어플리케이션 문제를 해결하기 위한 계산, 데이터 처리 및 가시화 도구들을 활용할 수 있어야 한다.
 - 문제를 해결하기 위해 쉽게 도구들을 통합하고 관리할 수 있어야 한다.
 - 어플리케이션 분야에 친숙한 표현으로 문제해 결 방법을 기술 할 수 있어야 한다.
 - 수행제어 및 데이터의 가시화가 가능해야 한다.
 - 문제해결 환경의 예로는 LabView, AVS 등이 있다. AVS는 그래픽을 이용한 프로그래밍 환경으로 아이콘들을 이용한다.
 - (2) 프로그래밍 환경

- 그리드 환경에서 수행되는 어플리케이션을 작성하기 위한 도구와 기술들은 다양한 프로그래밍 패 러다임을 포함해야 한다. 그리드 환경에서 Globus MPI 뿐만 아니라 Java, CORBA, Condor, Legion, DCOM등을 지원하게 한다.
- 컴파일 환경 관리, 분산 디버깅, 성능 분석 등은 해결해야 할 중요한 분야이다.
- 다른 환경에 수행을 위해 코드를 추가하는 도구 들도 필요하다. 예를 들면 보통 포트란 코드가 CORBA 환경에서 수행되게 하기 위해 코드들이 추 가되어야 하는데 이를 위한 자동화된 도구들이 필요 하다
- (3) 그리드 공통 서비스
 - 사용자가 필요로 하는 자원을 자동으로 찾아주는 기능 : 이와 같은 기능을 수행하기 위해서 사용자는 필요한 자원의 특징을 명시해야 하고 이를 기반으로 사용할 자원을 결정해야 한다.
 - 글로벌 큐잉 기능 : 큐에 자신의 작업을 제출하면 큐는 필요한 자원이 사용 가능하게 될 때 할당해 준다.
 - 작업의 흐름 관리 기능 : 자원들 사이에 어떤 흐름으로 수행되고 있는지 알려주고 흐름을 제어할 수 있게 한다.

- 자원에 발생하는 오류를 처리하는 기능 : 수행 중 예기치 않은 오류가 자원 상에 발생했을 때 자동으로 처리할 수 있는 기능이 필요하다.

- 그리드 내에 있는 자원들의 현재 상태와 같은 환경을 관리하기 위해 사용되는 중요한 서비스는 "공통정보베이스"라고 부르는 그리드 자원에 대한 정보 서비스이다. 이 서비스는 현재 Globus MDS에 의해 제공하고 있는데 자원들에 대한 세부적인 사항과 상태를 제공해야 한다. 더 나아가 동적으로 변하는 성능에 관한 정보를 제공하고 프로세스들의 상태, 사용자에 대한 정보 등을 제공한다.

- 동시할당 및 예약을 위한 자원 관리 : 실험장비들과 같은 자원을 스케줄링하고 여러 자원을 동시에 할당하는 기능. CPU 예약 스케줄링과 네트워크 대역폭을 예약하는 것은 동시할당 스케줄링에서 아주 중요한 문제이다.

(4) 그리드의 보안

- 사용자 인증과 보안을 위한 첫 번째 요구사항은 그리드 내의 자원들에 대해 단 한번의 인증과정만이 존재해야 한다는 것이다.

- 서로 데이터를 주고받을 때 데이터의 무결성을 위해 암호화된 통신채널이 필요하다.

(5) 운영을 위한 서비스

- 그리드를 운영하기 위해서 가장 먼저 필요한 것은 그리드의 상태를 나타내는 그리드 정보베이스이다.

- 그리드 관리자들이 원격지에서 발생한 문제를 조사하기 위한 진단 도구들이 필요하다.

- 다른 요구되는 서비스들은 사용자 계정 관리, 사용자료 계산, 작업에 대한 조사 등을 위한 도구들이다.

- 그리드의 성능과 신뢰성 등을 검증하기 위한 벤치마크와 회귀분석 도구들이 필요하다.

5. 결론

그리드는 지역적으로 분산되어 있는 이기종 컴퓨팅 자원을 통합하여 마치 단일 컴퓨터를 사용하는 것처럼 사용자 환경을 지원하는 첨단 컴퓨터 통합시스템이다. 또한 글로벌 환경에서 고성능 컴퓨팅 자원과 첨단 IT 기술들을 효율적으로 통합 연계하여 고성능 협업환경을 지원함으로써 이기종 컴퓨터간의 원활한 데이터 통신 및 상호공유를 지원한다. 이상에서 살펴본 바와 같이 여러 선진국에서는 국가 주도로

그리드 인프라 구축을 위해 많은 노력을 하고있다. 사실 정보기술분야에 있어서 미래를 예측하는 것은 아주 어려운 일이다. 하지만 현재 프로세서 기술의 발전과 네트워크 속도의 발전을 기반으로 하여 정보 기술 분야를 예측해 본다면 그리드 기술이 대세를 이룰 것이라 예측할 수 있다. Larry Smarr는 그리드 컴퓨팅은 19세기에 철도가 미국 중서부에 큰 영향을 미쳤던 것처럼 21세기 고성능 컴퓨팅 사회에 커다란 변화를 줄 것이라고 예측하고 있다.

향후 그리드는 많은 분야에 적용되어 전세계를 대상으로 글로벌 그리드 인프라로 발전 할 것으로 예상된다. 우리나라도 정부 차원이나 사업단 차원에서 장기적으로 투자하여 국가 그리드 인프라를 구축, 선별적으로 주요 어플리케이션 분야를 확장하여 나가는 것이 바람직하다고 생각된다. 아울러 국제 협력 프로젝트를 추진하여 현재 정부에서 구축해 놓은 APII Test-bed나 향후 구축될 유라시아 네트워크를 중심으로 국내 그리드 인프라 구축을 활성화시켜야 한다.

참고문헌

[1] A.S. Grimshaw and W.A. Wulf "Legion -- a view from 20,000 feet", Proc. 5th IEEE International Symposium on HPDC, IEEE Computer Society Press 1996.

[2] I. Foster, C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit" Intl. J. Supercomputer Applications, 11(2):115-128, 1997.

[3] I. Foster, J. Geisler, W. Nickless, W. Smith, S. Tuecke, "Software Infrastructure for the I-WAY High Performance Distributed Computing Experiment.", Proc. 5th IEEE Symposium on HPDC. pg. 562-571, 1997.

[4] I. Foster, C. Kesselman, "The Globus Project: A Status Report" Intl. J. Supercomputer Applications, 11(2): 115-128, 1997.

[5] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke, "A Security Architecture for Computational Grids." Proc. 5th ACM Conference on Computer and Communications Security Conference, pg. 83-92, 1998.

[6] I. Foster and C. Kesselman (eds.) "The Grid:

Blueprint for a new Computing Infrastructure” Morgan Kaufmann Publishers, 1998.

[7] 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.

[8] W. Allcock, A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke, “The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets.” Journal of Network and Computer Applications, 23:187-200, 2001.

[9] I. Foster, C. Kesselman, S. Tuecke, “The Anatomy of the Grid: Enabling Scable Virtual Organizations” Intl. J. Supercomputer Applications, 15(3), 2001.

[10] G. Allen, T. Dramlitsch, I. Foster, N.T. Karonis, M. Ripeanu, E. Seidel, and B. Toonen “Supporting Efficient Execution in Heterogeneous Distributed Computing Environments with Catus and Globus” Proc. SC01 (SC2001), Denver, CO, November 10-16, 2001.

[11] 정일웅, “기상/기후 예측 그리드”, 슈퍼컴퓨팅소식 5권 33-35페이지 2001년 6월.

[12] 조금원, “그리드 응용과 전산 유체역학”, 슈퍼컴퓨팅소식 5권 36-40페이지 2001년 6월.

[13] 이석, “인터넷 컴퓨팅의 생물학 응용 및 BiodGrid의 동향”, 슈퍼컴퓨팅소식 5권 41-45페이지 2001년 6월.

[14] Mark Baker, Rajkumar Buyya, and Domenico Laforenza, “Grids and Grid Technologies for Wide-Area Distributed Computing”, Submitted to the Software: Practice and Experience (SPE) Journal.

[15] <http://www.cs.wisc.edu/condor>

[16] <http://www.cactuscode.org>

[17] <http://archive.ncsa.uiuc.edu/Cyberia/MetaComp/MetaHome.html>

[18] <http://www.ncsa.uiuc.edu/About/PACI>

[19] <http://www.gridforum.org>

[20] <http://www.gridforumkorea.org>

[21] <http://www.cs.virginia.edu/~legion>

[22] <http://www.globus.org/>

[23] <http://www.ipg.nasa.gov>

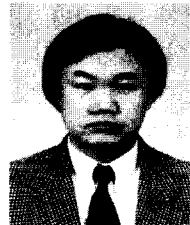
[24] <http://www.cs.sandia.gov/discom/main.html>

강 경 우



1990 경성대학교 전산통계학과 학사
 1992 한국과학기술원 전산학과 석사
 1998 한국과학기술원 전산학과 박사
 1998-1999 KISTI 선임연구원
 2000- 현재 천안대학교 정보통신학부
 관심분야: 그리드 정보서비스, 분산처리,
 컴파일러
 E-mail: kwkang@infocom.cheonan.ac.kr

박 형 우



1985 서울시립대학교 전자공학과 학사
 1996 성균관대학교 정보공학과 석사
 2001 성균관대학교 전기전자컴퓨터공학과 박사
 현재 KISTI 슈퍼컴퓨팅센터 슈퍼컴퓨팅
 팀 연구실장
 관심분야: 그리드 컴퓨팅, 차세대 TCP,
 인터넷 관리(QoS, 보안)
 E-mail: hwpark@hpcnet.neri.kr