

모듈 기반 적응형 그리드 환경 개발

이종만^o, 권성주, 최재영
송실대학교 컴퓨터학부

Module-based Adaptive Grid Environment

jmlee@ss.ssu.ac.kr^o lithmoon@ss.ssu.ac.kr choi@comp.ssu.ac.kr

Jongman Lee^o, Sungju Kwon, Jaeyoung Choi
School of Computing, Soongsil University

요 약

그리드는 다양한 기관들에 의해 소유되고 관리되는 컴퓨팅 자원들을 통합적이며 협력적으로 사용하기 위한 기반 구조이다. 이러한 기반 구조를 제공하기 위해서 Globus, Legion, Condor 등 많은 시스템이 개발되어 왔고 현재도 계속 개발 중에 있다. 하지만 이런 시스템은 시스템의 경직성 때문에 다양한 환경에 적용시키기 어렵다. 본 논문에서는 이러한 문제점을 보완하기 위해 기존의 시스템에 유연성을 제공하는 모듈 기반의 적응형 그리드 환경(Module-based Adaptive Grid Environment)을 개발하였다. 이에 개발자는 MAGE를 이용하여 유연하게 응용 프로그램을 개발할 수 있으며 사용자는 MAGE를 이용하여 자신이 사용하는 환경에 맞게 그리드를 사용할 수 있다. 더 나아가 MAGE를 사용하여 수시로 변화하는 환경에 적응하여 반응하는 자기 반영적인 미들웨어를 제공할 수 있다.

1. 서 론

그리드(Grid)[1]란 다양한 기관들에 의해 소유되고 관리되는 컴퓨터, 네트워크, 데이터베이스 등의 컴퓨팅 자원들을 통합적이며 협력적으로 사용하기 위한 기반 구조이다. 그리드는 1990년대 중반에 과학 및 공학용 분산 컴퓨팅 기반 구조를 설계하는 과정에서 등장하였다. 현재 이러한 그리드 환경을 제공하기 위해서 Globus[2], Legion[3], Condor[4] 등의 많은 프로젝트들이 진행 중이다. 이 중 미국의 ANL (Argonne National Laboratory)에서 개발되고 있는 Globus는 기존 시스템과의 연동과 각 서비스의 독립성이 뛰어나 가장 널리 사용되고 있다.

위의 그리드 미들웨어 각각은 여러 장점들을 가지고 있지만 시스템의 경직성 때문에 다양한 그리드 환경에서는 사용하기 어렵다. 이러한 시스템을 새로운 환경에서 사용하면 기존의 작업들을 재수행하거나 환경을 바꾸어야 하는 상황이 생길 수 있다. 그래서 다양한 그리드 환경 및 시스템에 대해 유연성을 가지는 시스템이 필요하다.

본 논문에서는 위의 요구를 만족할 수 있는 모듈 기반의 적응형 그리드 환경(Module-based Adaptive Grid Environment)을 개발하였다. MAGE는 기존의 시스템을 대체하는 것이 아니라 시스템이 유연성을 가지도록 도와주는 기능을 한다. 이러한 기능을 가지기 위해 MAGE는 다음의 두 가지 특징을 가진다. 첫 번째는 플러그블 모듈(pluggable module) 기반의 개발 방법이다. MAGE는 다양한 기능들을 모듈로써 제공한다. 개발자는 각각의 기능들을 분리하여 모듈 단위로 개발하기 때문에 환경에 대한 의존성이 줄어든다. 또 모듈들은 서로의 기능을 이

용할 수 있기 때문에 좀 더 쉽게 기능을 추가할 수 있다. 두 번째는 동적으로 모듈을 적재(load)하고 해제(unload)할 수 있는 기능이다. 이 기능을 이용하여 사용자 또는 관리자는 다양한 컴퓨팅 환경에 맞는 다른 기능을 가지는 MAGE들을 동적으로 구성할 수 있다. 또 수시로 변화하는 환경에 대응하여 모듈을 적재, 해제할 수 있는 자기 반영적인(self-reflective)[5] 그리드 미들웨어를 구축할 수 있다. 본 논문의 구성은 다음과 같다. 2장에서는 MAGE의 구조에 대해서 소개하고, 3장에서는 MAGE를 클러스터 모니터링 시스템에 적용한 사례에 대해서 설명한다. 마지막으로 4장에서는 본 논문의 결론을 맺는다.

2. MAGE의 구조

MAGE는 단순한 응용 프로그램이 아니라 그리드 상에서 사용할 수 있는 다양한 모듈들이 연계되어 동작하는 기반 환경이라고 할 수 있다. 개발자나 사용자는 MAGE를 사용함으로써 기존의 시스템이 유연성을 가지도록 도와주며 다양한 컴퓨팅 환경에 적용할 수 있는 시스템을 구축할 수 있다. 이러한 특성을 가지기 위해 MAGE는 여러 모듈들과 모듈들을 조율하는 모듈 관리자로 구성된다. 또 모듈들은 크게 기본적인 서비스를 제공하는 모듈과 개발자가 이를 이용하여 필요한 기능을 추가하기 위해 작성된 모듈로 나눌 수 있다. 그림 1은 이러한 MAGE의 구조를 보여준다. 그림 1에서 TCP, UDP, DLL Management 모듈들은 기본적인 서비스를 제공하는 모듈들이고, Monitoring 모듈은 개발자가 이를 이용하여 모니터링이라는 기능을 추가하기 위해 작성된 모듈이다.

MAGE는 여러 종류의 모듈들을 연계하여 사용할 수 있도록

모듈간의 통신이 필요하다. MAGE의 모듈간의 통신은 이벤트 기반 방식을 사용하며 모듈 관리자 (Module Manager)는 이런 모듈간의 통신에 필요한 기능들을 제공한다. MAGE가 동작하는 기본 원리는 이벤트가 발생하면 그 이벤트에 모듈 관리자에게 전달하고, 모듈 관리자는 이벤트를 처리하도록 등록된 모듈에 이벤트를 전달하고, 모듈은 이에 해당하는 기능을 수행한다. 예로 호스트 A에서 수집된 모니터링 정보를 호스트 B로 전송하기를 원한다면 모니터링 모듈이 이벤트를 생성하여 모듈 관리자에게 이벤트로 전달하고, 모듈 관리자는 이 이벤트를 처리하도록 등록된 TCP 모듈에게 전달하고, TCP 모듈은 수집된 모니터링 정보를 호스트 B에 TCP를 사용하여 전송한다. 만약 TCP 외에 다른 모듈을 사용하고 싶으면 모듈 관리자에게 그 이벤트에 대한 처리를 다른 모듈이 하도록 등록해두면 된다. MAGE에서 모듈간의 정보 교환을 이벤트 방식을 사용함으로써 쉽게 다른 모듈의 기능을 사용할 수 있고 모듈간의 독립성을 보장되어 모듈의 개발을 용이하게 해준다.

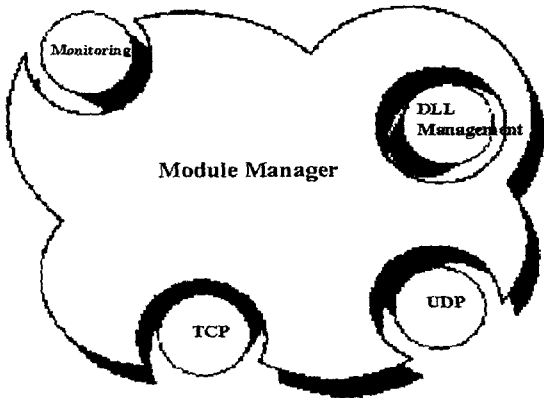


그림 1 MAGE의 구조

2.1 모듈 관리자 (Module Manager)

MAGE의 모듈 관리자는 다음과 같은 기능을 가진다.

- 1) 플러그를 모듈을 적재, 해제시켜서 모듈을 관리하는 기능을 수행한다.
- 2) 모듈 간의 자료를 교환하는 기능을 수행한다.
- 3) 모듈 간에 발생하는 이벤트를 처리하는 기능을 수행한다.

위의 기능에서 알 수 있듯이 모듈 관리자는 다양한 모듈들이 상호 운용될 수 있도록 조율해주는 역할을 한다. 또한 각각의 모듈은 동적으로 모듈 관리자에 의해 적재되고 해제되어서 필요한 모듈들만으로 구동할 수 있게 된다.

2.2 모듈 관리자의 구현

2.2.1 모듈 적재 방법

MAGE에서는 다시 컴파일하지 않아도 실행 중에 모듈을 적재, 해지할 수 있다. 이 기능을 가지기 위해서 모듈들은 DLL 형태로 되어 있으며 모듈의 적재는 모듈 관리자에 의해서 이루어진다. 모듈 관리자는 그림 2와 같은 XML 파일을 사용하여 모듈을 적재하는 데 필요한 정보를 나타낸다. XML 파일에는

모듈 이름, 모듈의 DLL 파일 이름과 모듈을 적재할 때, 모듈을 가동할 때, 모듈을 종료할 때 호출되는 루틴들로 이루어진다. 개발자는 위의 루틴을 반드시 작성하여 모듈을 개발해야 한다. 이는 모듈 관리자가 각각의 모듈들을 조율하기 위해 꼭 필요한 정보이기 때문이다.

2.2.2 모듈 적재 및 이벤트 처리

모듈 관리자는 그림 2의 XML 파일을 이용하여 필요한 모듈들을 적재하고 모듈들의 정보를 얻어온다. 모듈이 적재된 후 그림 3와 같은 순서로 MAGE는 동작하게 된다. 모듈이 적재되면 모듈은 모듈 관리자가 유지하는 정보인 모듈 테이블 (Module Table)에 등록된다. 이때 모듈안의 init 루틴이 호출되는데 모듈은 모듈 관리자가 유지하는 정보인 이벤트 테이블 (Event Table)에 모듈이 처리해야 할 이벤트들을 등록하게 된다. 모듈이 실행 중에 이벤트를 발생시켜 이를 모듈 관리자에게 알리면, 모듈 관리자는 발생한 이벤트를 처리해야 할 다른 모듈들에게 이를 통보한다. 이벤트를 통보받은 모듈들은 이벤트를 처리하고 필요하다면 다른 모듈에게 이벤트를 통보한다. 마지막으로 모듈의 사용이 필요 없게 되면 모듈이 해지된다. 이 때 이벤트 테이블에서 모듈에 의해 등록된 이벤트를 삭제한다.

```
<?xml version="1.0" ?>
<module_manager>
  <dll name="TCP Communication">
    <fileName>./lib/TCP/libTCP.so</fileName>
    <init>initTCP</init>
    <startup>startupTCP</startup>
    <finalize>finalizeTCP</finalize>
  </dll>
  <dll name="UDP Communication">
    <fileName>./lib/UDP/libUDP.so</fileName>
    <init>initUDP</init>
    <startup>startupUDP</startup>
    <finalize>finalizeUDP</finalize>
  </dll>
</module_manager>
```

그림 2. 모듈적재를 위한 XML 파일

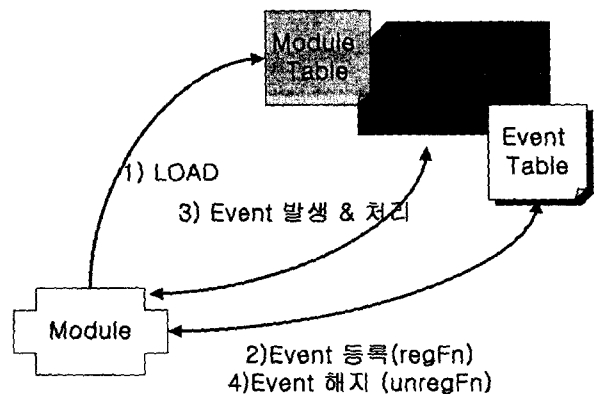


그림 3 모듈 적재 및 이벤트 처리

2.3 네트워크 모듈 (Network Module)

MAGE는 그리드 기반 환경이기 때문에 새로운 응용프로그램을 작성하기 위해서는 통신 수단이 필수적으로 제공되어야 한다. 또한 그리드 환경을 고려하여 다양한 통신 수단에 대한 지원도 제공해야 한다. MAGE는 개발자가 용이하게 모듈을 개발할 수 있도록 통신 수단을 모듈로써 제공한다. 이에 개발자는 하위의 통신 수단을 상관하지 않고 응용프로그램을 개발할 수 있으며 응용프로그램은 여러 통신 수단을 사용할 수 있다. 또한 MAGE에서는 동적으로 모듈을 적재할 수 있는 특징을 가지기 때문에, 이를 이용하여 그림 4과 같이 통신에 필요한 기본적인 모듈들을 개발하여 환경 또는 상황에 맞추어 필요한 모듈을 사용할 수 있게 해준다.

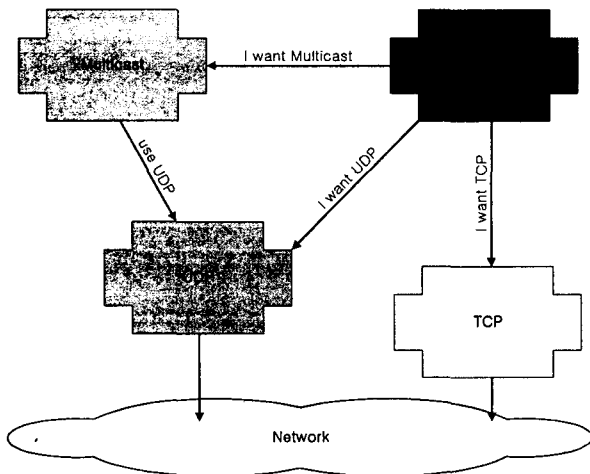


그림 4. 네트워크 모듈

3. MAGE 응용사례

MAGE의 기능을 실험해 보기 위해서 클러스터 모니터링 시스템에 적용하여 구현하였다. 클러스터 모니터링 시스템을 개발하기 위해서는 MAGE에서 제공하는 기본적인 서비스를 제공하는 모듈뿐만 아니라, 모니터링에 필요한 기능에 필요한 기능을 제공하는 모듈들이 필요하다. 그림 5는 MAGE를 이용하여 구현된 모듈들을 바탕으로 설계된 모니터링 구조를 나타내는 그림이다. 이 구조는 다음과 같은 시나리오로 동작한다. 각각의 노드에서 cpu 사용량, 메모리 사용량, 디스크 사용량 등의 모니터링 정보를 모은 다음, 이를 멀티캐스트 방식으로 전송하여 그룹 내의 모든 노드들이 그룹 내의 노드들에 대한 정보를 가지게 된다. 모아진 모니터링 정보는 모니터링 클라이언트로 보내진다. 또 원하는 컴퓨터들을 그룹화하여 많은 수의 컴퓨터를 관리할 수 있는 기능을 제공한다. 위의 기능들은 자원 수집 모듈, 멀티캐스트 모듈, 정보 서비스 모듈, 그룹 관리자 모듈이라는 MAGE의 모듈로 구현되었다.

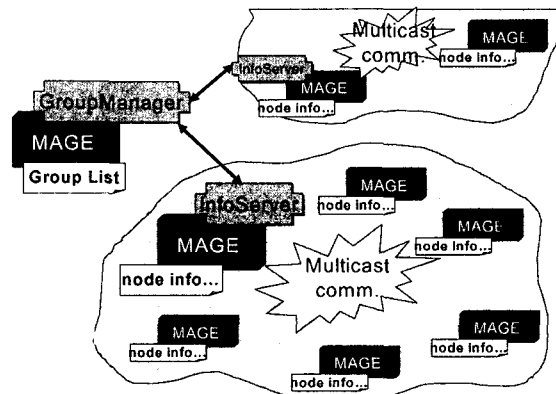


그림 5. MAGE를 이용한 모니터링 구조

4. 결론 및 향후과제

본 연구에서는 기존의 그리드 시스템의 경직성을 보완하여 유연성을 제공하기 위해 MAGE를 제안하고 구현하였다. MAGE는 플러그인 모듈 개발 방법과 동적으로 모듈을 적재, 해제하는 기능을 제공함으로써 다음과 같은 효과가 기대된다. 개발자 각각의 기능을 분리하여 모듈로 개발하기 때문에 환경에 대한 의존성을 줄이게 되며, 모듈들은 서로의 기능을 사용할 수 있기 때문에 쉽게 원하는 기능을 개발하도록 도와준다. 또 사용자와 관리자에게는 수시로 변화는 상황에도 능동적으로 대응하는 환경을 구축할 수 있도록 도와준다.

앞으로는 다른 개발자가 쉽게 모듈을 개발하기 위해 많은 기본 모듈들과 편리한 API를 제공할 예정이다. 또한 클러스터 관리자가 사용하는 시스템이기 때문에 안전한 시스템 관리를 위해서 보안 모듈과 모니터링 정보를 바탕으로 클러스터들을 관리할 수 있는 도구를 개발할 예정이다.

참고문헌

- [1] I. Foster and C. Kesselman, "The Grid:Blueprint for a New Computing Infrastructure",Morgan Kaufmann,1998.
- [2] The Globus Project, <http://www.globus.org>
- [3] Andrew S. Grimshaw, William A. Wulf, James C. French, Alfred C. Weaver, Paul F. Reynolds - Legion: The Next Logical Step Toward a Nationwide Virtual Computer,1994.
- [4] M. Litzkow, M. Livny, & M. Mutka, Condor - A hunter of idle workstations. In Proceedings of the Eighth Conference on Distributed Computing Systems, San Jose, California, June 1988.
- [5] G.S. Blair, G. Coulson, P. Robin, M. Papatomas Distributed Multimedia Research Group Computing Department, Lancaster University - An Architecture for Next Generation Middleware ,1998.