

그리드 컴퓨팅 시스템에서 QoS 결합 탐지 방법

이화민* 정광식** 진성호⁰ 이대원* 정순영* 유현창*

* 고려대학교 대학원 컴퓨터교육과 ** 런던대학교 컴퓨터학과

{zelkova, wingtop⁰, ldw1996, jsy, yuhc}@comedu.korea.ac.kr, k.chung@ucl.ac.uk

A Design of QoS Fault Manager in Grid Computing System

Hwa-Min Lee Kwang Sik Chung Sung-Ho Jin⁰ Dae-Won Lee Soon-Young Jung Heon-Chang Yu

* Dept. of Computer Science Education, Korea University ** Dept. of Computer Science, University College London

요 약

그리드 컴퓨팅 시스템은 광범위한 지역에 분산된 고성능 자원, 대용량 정보 및 혁신적인 애플리케이션 등을 네트워크로 연결하여 마치 단일 컴퓨터를 사용하는 것처럼 자원을 상호공유하고 이용할 수 있도록 지원한다. 광범위 분산 컴퓨팅 시스템(wide area distributed computing systems)인 그리드 역시 분산 컴퓨팅 시스템과 같이 각 자원의 결합 발생 가능성이 존재한다. 그리드 컴퓨팅 시스템에서도 결합은 작업 수행에 있어서 치명적일 수 있기 때문에 결합 포용 기능은 필수적인 요소이다. 하지만 그리드 컴퓨팅 환경을 제공하는 글로버스(globus) 미들웨어에는 결합 포용 기능이 포함되어 있지 않다. 이에 본 연구에서는 그리드 컴퓨팅 시스템에서 QoS(quality of service)를 고려하여 프로세스, 프로세서, 네트워크 결합을 정의하고, 정의된 결합이 발생할 경우 결합을 탐지하고 해결하는 방법을 제안한다.

1. 서 론

기존의 분산 컴퓨팅 시스템에 대한 연구가 활발히 이루어지면서 1990년 중반 미국의 슈퍼 컴퓨팅 센터를 중심으로 고성능의 분산 컴퓨팅 인프라를 구축하면서 차세대 인터넷으로 연결하는 발단으로 그리드(Grid)라는 개념이 등장하였다. 진공관의 음극과 양극 중간에서 전류의 흐름을 제어하는 격자(Grid)에서 유래한 그리드는 한번에 한곳에만 연결할 수 있는 웹과 달리 신경조직처럼 작동하는 인터넷 네트워크 구조를 말한다[1, 2].

그리드는 지역적으로 분산되어 있는 고성능의 컴퓨팅 자원을 네트워크로 연결하여 마치 단일 컴퓨터를 사용하는 것처럼 자원을 상호공유하고 이용할 수 있도록 지원하는 컴퓨팅 시스템을 말한다. 즉, 그리드는 컴퓨터에 특정 프로그램을 설치하여 전 세계 컴퓨터, 데이터베이스, 첨단 장비 등을 인터넷으로 연결해 마치 슈퍼컴퓨터처럼 쓰자는 개념이다. 그리드 개념에서 실제적인 특징은 가상 조직(VO : virtual organization)이라 칭하는 자원 공유에 관한 것으로, 여기서 주목할 점은 단순한 파일 공유뿐만 아니라 컴퓨터, 데이터, 소프트웨어 등과 같은 다양한 자원들에 직접적으로 접근을 허용하는 것이다[1].

그리드 컴퓨팅 시스템에서 효율적인 자원 관리와 자원 공유를 위해서는 그리드에 존재하는 자원들에 대한 정보를 관리하고 자원을 할당하는 역할을 담당하는 자원 관리자가 필요하다[3]. 그리고 그리드 시스템에서 실행되는 애플리케이션들은 네트워크를 기반한 수행을 하기에 고성능의 수행을 보장을 받기를 요구한다. 이에 그리드 컴퓨팅 시스템에서 자원 관리자는 QoS를 고려한 자원 관리를 제공해야 한다. 그리고 그리드 역시 지역적으로 분산되어 있는 다양한 컴퓨팅 자원을 네트워크로 연결하여

사용할 수 있는 환경으로 각 자원의 결합 발생 가능성과 네트워크 결합 발생 가능성이 존재한다. 광범위 분산 컴퓨팅 시스템(wide area distributed computing systems)인 그리드에서도 결합은 작업 수행에 있어서 치명적일 수 있기 때문에 결합 포용 기능은 필수적인 요소이다. 즉, 그리드와 같은 광범위 분산 컴퓨팅 시스템에서의 결합 발생 잠재성은 애플리케이션 개발자들에게는 매우 중요하고 처리하기 어려운 문제가 되고 있다.

하지만 그리드 컴퓨팅 환경을 제공하는 글로버스(globus) 미들웨어에는 결합 포용 기능이 포함되어 있지 않다. 현재까지의 결합에 대한 연구는 HBM 툴킷(toolkit)을 이용하여 결합의 탐지만을 제공하는 것으로 결합에 대해 수동적으로 대처하고 있을 뿐이다. 이에 본 논문에서는 그리드 컴퓨팅 시스템에서 QoS를 고려한 자원 관리와 결합 포용 기능을 제공하고자 한다. 이를 위해 QoS 측면을 고려하여 프로세스, 프로세서, 네트워크의 결합을 새롭게 정의한다. 그리고 정의된 결합을 바탕으로 QoS 탐지 에이전트, 결합 탐지 에이전트, 결합 관리 에이전트를 이용하여 결합 발생을 탐지하고 탐지된 결합을 해결하는 방법을 제안한다.

2. 관련 연구

그리드라는 개념은 등장한지 불과 몇 년이 되지 않는 최신의 이론으로 그리드 환경에서 직접적으로 결합 포용에 관련된 연구는 [4] 뿐으로 그리드에서 결합 포용은 미개척 분야이다. [4]에서는 결합 탐지 서비스 위해 HBM이라는 툴킷을 제안하였다. HBM은 로컬 모니터(local monitor)와 데이터 컬렉터(collector)로 구성된다. 각 호스트마다 로컬 모니터가 위치하고 로컬 모니터는 주기적으로 i-am-alive 메시지를 통해 해당 호스트의 프로세스들의 상태를 모니터링하여 결합 발생 여부를 찾아낸다. 각

* 본 과제(결과물)는 정보통신부의 정보통신기술기초연구지원사업(정보통신연구진흥원)으로 수행한 연구결과입니다.

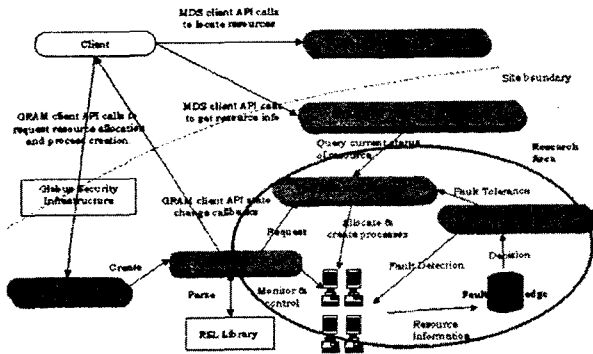
호스트 외부에 위치한 데이터 컬렉터는 자신의 호스트의 로컬 모니터가 모니터링한 결과를 정리하여 생성한 심장 박동(heart-beat) 메시지를 통해 결함이 발생한 요소를 실제적으로 식별해낸다.

[4]는 프로세스와 컴퓨터의 파손(crash)에 대한 결함만을 다루고 있어 결함의 범위가 너무 제한적이라는 한계가 있고 또한 프로세스와 컴퓨터의 결함에 대한 탐지 방법만 제시되고 그 이후 해결 방안은 제시되지 않았다는 문제점이 있다.

3. QoS 결함 관리자 설계

3.1 시스템 모델

본 논문에서는 글로벌스 미들웨어 환경을 전제로 한다. <그림 1>은 기존의 글로벌스에 본 논문에서 제안하는 QoS 결함 관리자가 도입되는 경우의 구성도이다.



<그림 1> 글로벌스에 QoS 결함 관리자가 도입된 모습

일반적인 그리드 컴퓨팅 환경에서의 시스템 요소들은 전체 사이트, 컴퓨터, 프로세서, 프로세스, 쓰레드, 네트워크 인터페이스, 네트워크 연결(connection), 저장 장치, 데이터베이스, 각종 어플리케이션, 다양한 하드웨어 관련 서비스 등이 될 수 있다. 하지만 이같은 시스템 요소 모두를 결함 대상으로 고려할 경우 결함 관리의 복잡성과 오버헤드가 커지는 관계로 본 논문에서는 프로세스, 프로세서, 네트워크 서비스만을 결함의 대상으로 가정한다. 그리고 글로벌스가 지원하는 운영체제를 기반으로 본 논문의 QoS 결함 관리자가 동작하는 운영체제를 리눅스(유닉스)와 솔라리스 환경으로 제한한다.

3.2 QoS를 고려한 결함의 확장

분산 컴퓨팅에서는 프로세스를 공유하여 컴퓨팅 파워를 증가시키는 것에 자원 공유의 초점이 맞추어졌다. 하지만 그리드 컴퓨팅에서는 프로세스뿐만 아니라 CPU, 저장 장치 서비스, 각종 어플리케이션 서비스, 다양한 네트워크 서비스 등 다양한 자원을 공유 대상으로 보고 있다. 이에 그리드 컴퓨팅에서는 결함의 대상을 프로세스에 한정짓기 보다 각종 VO에서 발생하는 결함들을 대상으로 결함에 대한 정의를 확장하는 것이 의미가 있다. 이에 본 논문에서는 QoS를 고려하여 프로세스, 프로세서, 네트워크 결함의 범위를 확장하여 다음과 같이 결함을 새롭게 정의한다.

[정의 1] 결함(failure)

다음의 조건을 하나라도 만족하는 경우 결함이라 한다.

1. VO의 파손(crash)으로 인한 서비스 정지(stop)
2. 각 VO에서 요구하는 QoS 조건을 만족하지 못하는 경우

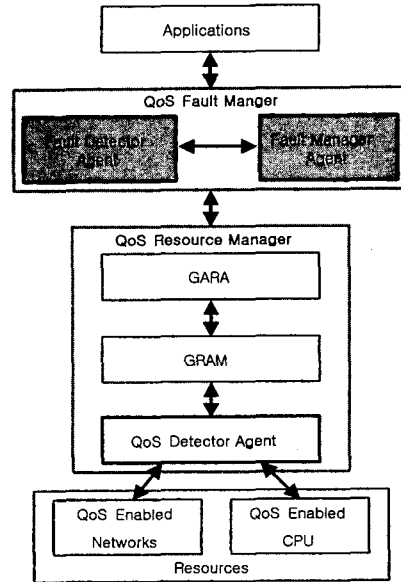
<표 1>은 [정의 1]에 따라 본 논문에서 결함의 대상으로 보는 각 VO의 세부적인 결함을 정리한 것이다.

<표 1> 각 자원에 대한 결함의 종류

| 프로세스의 결함 |
|--|
| ① 프로세스의 작업 수행 정지(fail-stop) |
| 프로세서의 결함 |
| ① 프로세서의 동작 정지(crash) |
| ② 작업량 폭주(burst job)로 인한 프로세서의 처리율(throughput) 감소(QoS 결함) |
| 네트워크의 결함 |
| ① 통신 단절(communication disconnection) |
| ② 통신 트래픽(communication traffic) 증가로 인한 대역폭(bandwidth) 감소(QoS 결함) |

3.3 QoS 결함 관리자 구조

본 논문에서 제안하는 결함 관리자의 구조는 <그림 2>와 같다.



<그림 2> 결함 관리자 구조

① 결함 탐지 에이전트(Fault Detector Agent) : 하위의 QoS 자원 관리 시스템을 기반으로 애플리케이션에서 사용하는 프로세스, 프로세서, 네트워크 자원의 결함 발생 여부를 탐지한다.

② 결함 관리 에이전트(Fault Manager Agent) : 결함 탐지 에이전트의 수행 결과를 리턴받아 결함이 발생한 자원을 대체할 자원을 할당해 주도록 하위의 QoS 자원 관리자에게 요청한다. 그리고 필요한 경우 애플리케이션에 결함이 발생한 자원을 통보한다.

③ GARA : 프로세서와 라우터와 같은 물리적인 장치들의 조정과 기본적인 대역폭 예약을 위해 사용된다. 즉, QoS 지원을 위해 자원의 예약과 할당을 관리한다.

④ GRAM : 글로버스에서 자원들의 할당을 관리한다.

⑤ QoS 탐지 에이전트(QoS Detector Agent) : 프로세서와 네트워크의 QoS 지원을 위해 프로세서의 사용도(utilization)와 네트워크 대역폭 정보를 수집한다.

⑥ 물리적 장치(Network & CPU) : QoS 지원을 위해 DS(Differentiated Service)와 DSRT를 통해 조정된다.

4. QoS 결함 관리자 서비스

4.1 결함 탐지 서비스(Fault Detection Service)

결함 탐지 서비스는 프로세스 결함 탐지, 프로세서의 결함 탐지, 네트워크 결함 탐지로 세 가지로 이루어진다.

프로세스 결함 탐지의 경우 결함 탐지 에이전트가 주기적으로 결함 탐지 메시지를 프로세스에게 전송하여 일정한 시간(timeout)동안 그에 대한 응답이 없을 경우 프로세스에 결함이 발생하였다고 판단한다.

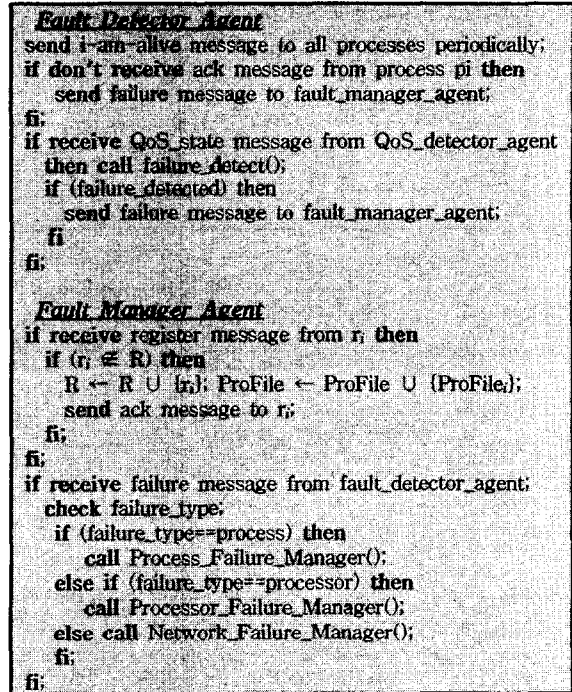
프로세서의 정지 결함은 주기적으로 결함 탐지 에이전트가 프로세서를 모니터링하여 탐지한다. 프로세서의 작업 처리 성능 저하 결함 탐지를 위해서는 QoS 탐지 에이전트가 리눅스와 솔라리스에서 제공하는 CPU의 사용률을 측정하는 명령어를 활용한다. 기본적으로 리눅스 혹은 유닉스에서는 CPU나 기타 여러 자원에 대해서 사용량을 측정하게 해주는 API 나 함수 혹은 라이브러리는 지원하고 있지 않기에 쉘 프로그램을 이용하여 프로그램을 작성하여 사용한다. 본 연구에서는 'top' 명령어를 사용하여 'top' 명령은 프로그램을 시간 간격을 두고 백그라운드로 실행하여 결과물을 로그로 남길 수 있기에, 남겨진 로그 정보를 파싱하여 CPU 사용도를 측정하는 프로그램을 작성한다. 솔라리스의 경우 'vmstat'와 'sar' 명령어를 사용하여 CPU 사용도를 측정한다.

네트워크 결함의 경우 QoS 탐지 에이전트가 글로버스의 미들웨어에서 네트워킹 성능 측정을 위해 제공되는 툴킷을 이용하여 네트워크의 지연, 선로 용량, 손실율, 지터 등의 정보를 얻어 통신 단절과 대역폭이 감소하는 QoS 결함 발생 유무를 탐지한다. 네트워크 QoS를 지원하기 위해 본 논문에서는 DS 모델을 사용한다.

4.2 결함 관리 서비스(Fault Management Service)

결함 관리 에이전트는 탐지된 결함을 해결한다. 결함 관리 에이전트는 결함 탐지 에이전트로부터 결함 메시지를 받게 되면 결함의 종류를 체크하여 식별하고 각 결함들을 해결한다. 이를 위해 결함 관리 에이전트는 그리드를 구성하는 호스트로부터 호스트가 제공할 수 있는 각종 자원들(VO)을 등록받아 자원 profile을 작성하여 유지한다. 발생한 결함이 식별되면 profile의 정보를 이용하여 결함이 발생한 자원을 대체할 수 있는 백업(backup) 자원을 찾아 결함을 해결해준다. 대체 대상을 찾는 방법은 기존의 리소스 관리 기법을 이용한다.

결함 탐지와 해결을 위해 결함 탐지 에이전트와 결함 관리 에이전트가 동작하는 알고리즘은 <그림 3>과 같다.



<그림 3> 결함 탐지 및 관리 알고리즘

5. 결론 및 향후 연구과제

본 논문에서는 그리드 컴퓨팅 시스템에서 QoS를 고려하여 프로세스, 프로세서, 네트워크 결함을 새롭게 정의하고 QoS 탐지 에이전트, 결함 탐지 에이전트, 결함 관리 에이전트를 이용하여 정의된 각 결함을 탐지하고 해결하는 방법을 제안하였다.

향후 연구과제로 본 논문에서 제안한 QoS 탐지 에이전트, 결함 탐지 에이전트, 결함 관리 에이전트를 설계하여 글로버스에서 동작하는 QoS 자원 관리자 툴킷(toolkit)과 QoS 결함 관리자 툴킷을 구현하고자 한다.

[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] Ian Foster, Carl Kesselman, The Grid : Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers, 1998.
- [3] I. Foster, A. Roy, V. Sander, A Quality of Service Architecture that Combines Resource Reservation and Application Adaptation, 8th International Workshop on Quality of Service, 2000.
- [4] P. Stelling, I. Foster, C. Kesselman, C.Lee, G. von Laszewski, A Fault Detection Service for Wide Area Distributed Computations, Proc. 7th IEEE Symp. on High Performance Distributed Computing, pp. 268-278, 1998.