병렬 및 분산 환경에서의 고장 감내 메시지 전달 인터페이스 [1]

송대기°, 김종훈, 강용호, 이철훈
충남대학교 컴퓨터공학과

Fault-Tolerant Message Passing Interface on Parallel and Distributed Systems

Dae-Ki Song°, Jong-Hoon Kim, Yong-Ho Kang, Cheol-Hoon Lee
Dept. of Computer Engineering, Chungnam National University

요 약

본 논문에서는 메세지 전달을 기반으로 하는 병렬 분산 시스템상에 고장 감내 기능을 추가하기 위한 고장 감내 기법과, 고장 복구에 따른 프로세스들간의 일관성 유지 방법을 제안하였다. 메시지 전달을 기반으로 하는 병렬 컴퓨터 시스템상에서 응용 프로그램들은 수많은 노드들에 분산 배치되어 수행이 되는데, 그 중 어느 한 노드 또는 작업 중인 프로세스가 고장을 일으킨다면 이로 인하여 전체 응용프로그램이 중단 될 것이다. 이러한 문제를 해결하기 위하여 고장 감내 기능 추가가 필요하며, 그 방법으로서 동일한 작업을 수행하는 프로세스를 서로 다른 노드 상에 이중화하여 하나의 프로세스에 고장이 발생하더라도 계속 작업중인 예비 프로세스를 이용함으로써 전체 응용프로그램이 아무런 영향을 받지 않도록하였다. 그리고 이를 MPI상에 서브 모듈로써 설계하고 구현하였다.

1. 서론

병렬 분산 환경 하에서 대부분의 응용 프로그램들이 수행되는 일반적인 형태는 다수의 프로세싱 노드상에서 많은 프로세스들이 서로 메세지 전달을 통해 계산 결과를 주고 받는 것이다. 이러한 메세지 전달을 기반으로 하는 병렬 분산 응용 프로그램들은 하나의 일을 처리하기 위해 수많은 프로세스들이 여러 프로세싱 노드상에서 수행되면서 그 결과를 메세지를 통해 전달하게 된다. 이때 프로세싱 노드의 이상이나 통신 채널, 또는 작업을 수행하는 프로세스 문제로 인하여 전체 프로그램이 중단될 수도 있다. 이러한 문제를 해결하기 위하여 고장 감내 기법 추가가 필요하며, 본 논문에서는 그 방법으로 Hot-Spare에 의한 프로세스 이중화 기법을 사용하였다[1].

메세지 전달을 기반으로 하는 병렬 프로그램들은 병렬 시스템 자체에서 제공하는 메세지 전달 방법을 이용하여 작성되는데 이렇게 작성된 응용 프로그램은 시스템에 의존적이기 때문에 프로그램 호환성에 많은 문제가 있다. 따라서, 메시지 전달 방법을 사용하는 시스템들 사이의 호환성을 높이기 위하여 표준을 만들기 위한 노력이 계속되어왔다. 그리고 이러한 결과로 MPI(Message Passing Interface)가 1994년에 표준 메세지 전달 라이브러리로 제정되었다[2]. MPI를 이용하여 작성된 병렬 프로그램은 MPI가 이식된 어느 시스템상에서든지 프로그램의 수정없이 바로 컴파일하여 실행시킬 수 있으므로, 병렬 시스템 자체 보다는 MPI에 고장 감내 기능을 추가함으로써 이식성을 높이고 선택적으로 고장 감내 기능을 사용할 수 있도록 하였다. 고장 감내 응용프로그램을 작성하기 위하여 단순히 고장 감내 기능이 추가 된 FT-MPI(Fault Tolerant-MPI) 라이브러리를 사용함으로써 해결할 수 있다.

본 논문에서는 MPI에 고장 감내 기능을 추가하기 위해 제안하고 설계한 내용을 기술하며, 또한 고장 복구과정에서 프로세스간의 메시지 손실을 해결하는 동기과정을 설명한다.

2. FT-MPI(Fault Tolerant-MPI)

FT-MPI는 병렬 시스템 상에 메시지 전달 방법으로 표준이 된 MPI에 고장 감내 기능을 추가한 것이다. FT-MPI

[1] 본 연구는 과학기술부-과학기술정책관리연구소 주관 미래 원천 국책과제의 연구비에 의하여 연구되었음.

를 사용함으로써 고장 감내를 위하여 별도의 하드웨어가 필요하지 않으며 최소한의 운영체제의 지원만으로 고장 감내가 가능하게 된다. 기존의 MPI를 기반으로 작성된 응용 프로그램을 최소의 수정으로 고장 감내 기능을 추가해 수행되도록한다 다음에서는 본 논문에서 제안한 병렬 분산 환경하의 고장 감내 방법에 대하여 설명한다. 그리고 고장 감내를 위해 제안된 FTM(Fault-Tolerant Manager)과 고장 감내 기법에 대하여 기술한다.

2.1 프로세스 이중화

고장 감내를 위하여 프로세스를 이중화 하여 수행되도록 하였는데, 프로세스 이중화 기법 중 Hot-Spare 기법을 사용하였다. Hot-Spare는 동일한 두 프로세스가 각각 primary와 backup이 되어 동시에 같은 일을 수행하는 기법이다. 이 두 프로세스는 같은 메시지를 주고 받으면서 같은 결과를 출력할 것이다. Hot-Spare 방법을 사용하면 짝을 이루고 있는 두 프로세스 중 어느 하나가 고장이 발생한다 하더라도 나머지 한 개의 프로세스에 의해서 상태가 보존되므로 이를 사용하여 복구가 이루어 질 수 있다. 두 프로세스가 같은 상태에서 수행 되므로 고장 복구를 위한 응용프로그램이 수행 중에 Checkpoint 같은 추가작업을 하지 않아도 되며, 복구 후 전역상태에서의 일관성 유지를 위한 Roll-Back을 하지 않아도 된다.

2.2 FTM(Fault-Tolerant Manager)

FTM은 MPI의 서브 모듈로 추가 되며 고장 감내 기능을 담당한다. 기존의 MPI에 고장 감지 및 진단과 복구기능을 제공해주며, 프로세스 이중화에 따른 메시지 전달의 이중화와 메시지 비교 기능을 한다.

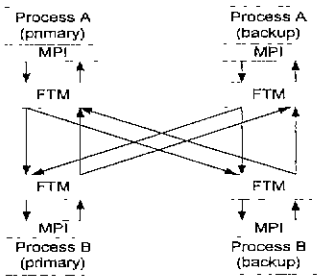


그림.1 FT-MPI를 이용한 고장 감내 응용 프로그램

그림1은 FT-MPI상의 응용프로그램을 나타낸다. MPI에 추가된 FTM은 프로세스 이중화와 고장 감지,진단 및 고장 복구 과정을 책임진다.

2.2 고장 감지

고장 감지는 메시지를 받는 쪽에서 한다. 모든 메시지는 primary 프로세스와 backup 프로세스 양쪽으로부터 받아야 한다. 만약 일정 시간 안에 양쪽으로부터 메시지가 도착하지 않으면 고장 여부를 확인하기 위한 고장 진단 과정에 들어가게 된다

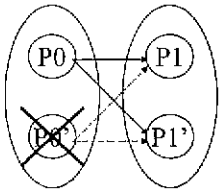
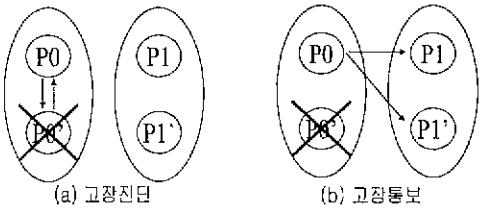


그림2 고장 탐지 과정

그림2는 고장을 탐지하는 과정으로 P0' 프로세스에서 고장이 발생한 경우이다. 이때 메시지를 수신하는 P1,P1' 프로세스는 P0'의 고장을 발견하게 된다.

2.3 고장 진단

메시지 수신시 고장을 발견한 프로세스는 해당 프로세스의 짝 프로세스(pair process)에게 고장 확인을 요청하게 된다. 고장 확인 요청을 받은 프로세스는 고장 진단 메시지를 자신의 짝 프로세스에게 전달하며 이에 대한 응답이 없으면 현재 작업을 중단하고 고장 통보를 하고 고장 복구 모드로 들어가게 한다.



(a) 고장진단 (b) 고장통보

그림3. 고장 진단 및 고장 통보과정

2.4 고장 통보

진단 후 복구 모드로 들어가기 위해서 고장 발생을 응용프로그램을 구성하는 모든 프로세스들에게 전달하여야 한다. 고장 통보 메시지에는 고장이 발생한 프로세스의 ID를 포함하며, 고장 통보 메시지를 받은 프로세스들은 고장 복구 완료 메시지를 기다리게 된다.

2.5 고장 복구

고장 복구는 고장이 발생한 프로세스의 짝 프로세스가 수행하게 되며, 일단 새로운 backup 프로세스를 생성할 프로세싱 노드를 선택하게 된다. 복구를 수행하는 프로세

스는 선택 된 노드에 자신의 프로세스를 그대로 복사하여 backup을 만든다. 생성된 backup 프로세스는 자신의 통신 포트 번호를 다른 프로세스들에게 알려 정보를 업데이트 하도록 한다.

2.6 재시작

새로운 프로세스가 생성되고 난 후 backup 프로세스는 응용프로그램의 모든 프로세스들에게 복구완료 메시지를 보내 복구모드를 마치도록 한다. 재시작 메시지를 받은 프로세스들은 메시지 동기를 맞춘 후 재수행 하게 된다. 메시지 동기 과정은 3절에서 설명한다.

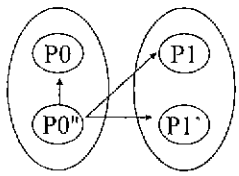


그림4. 고장복구 완료 메시지 전송과정

그림4.는 고장 복구 완료 후 재수행 메시지를 보내는 과정을 나타낸다

3. 고장 복구 후 처리과정

FTM에서의 고장 복구 후 일관성 있는 상태로 재수행 하기 위한 과정을 기술한다. 이 과정을 아래의 예로 설명한다.

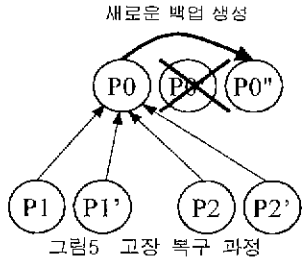


그림5 고장 복구 과정

그림5는 고장 복구 과정을 나타내는데, P0' 프로세스에서 고장이 발생한 후 그의 짝인 P0에 의해 새로운 backup 프로세스인 P0'' 프로세스가 생성된다.이 때 복구 후의 P0''와 다른 프로세스(P1.P1'.P2.P2')사이에 missing 메시지가 발생하는데, 이는 새로 생성된 P0''프로세스가 P0의 상태를 그대로 이어 받았기 때문이다.

Missing 메시지가 발생하는 또 다른경우는, 통신포트의 변경이 이루어지지 않고 메시지가 보내어진 때이다. 고장이 복구되고 나서 메시지를 받게되는 경우인데, 새로 생성된 backup 프로세스는 통신포트가 변경되었기 때문에 상대방 프로세스에게서 메시지를 받지 못한다.

이러한 문제를 해결하지 않으면 재수행시 일관성을 잃게 되고,응용프로그램이 정상동작할 수 없게 된다. 첫번째 복구 후 발생하는 missing 메시지는 고장 발생한 프로세스의 짝에게서 온 메시지를 이용하는데,이는 Hot-Spare로 프로세스가 이중화 되어 있기 때문에 가능하다. 두번째 고장 복구 후 통신 포트의 변경으로 인하여 missing 메시지가 발생하는 문제는 메시지 전송시 ack메시지를 사용해서 ack에 대한 응답이 없을 때 메시지를 재전송하게 함으로써 처리할 수 있다

4. 결론 및 향후 과제

본 논문에서 제안된 고장 감내 기법은 메시지 전달을 기반으로 하는 병렬 분산 시스템에 고장 감내 기능 추가를 목적으로 한다. 대부분 병렬 시스템의 메시지 전달을 위해 사용 되는 MPI에 고장 감내 기능을 추가함으로써 별도의 하드웨어가 필요하지 않고 기존의 MPI 응용프로그램도 최소한의 수정으로 고장감내 기능을 가질 수 있도록 하였다. 고장 감내를 위하여 MPI에 추가되는 모듈로서 FTM(Fault-Tolerant Manager)를 설계하고 구현하였으며, 고장 복구 후 재수행시 프로세스들 사이의 일관성을 유지할 수 있는 방법을 제안하였다.

본 논문의 향후 과제로는 고장 감내 기능을 더 원활히 수행하도록 개선해 나가는 것이며, 또한 Hot-Spare에서 프로세스 이중화에 의해 발생하는 I/O작업의 문제점들을 해결해 주는 것이다.

참고문헌

[1] Flavin Cristian, "Understanding Fault-Tolerance Distributed Systems", *CACM*, vol.34, no.2, Feb. 1994.

[2] Marc Snir, et al, "MPI: The Complete Reference", The MIT Press, 1996.

[3]D. Manivannan, Robert H. B. Netzer, and Mukesh Singhal, "Finding Consistent Global Checkpoints in a Distributed Computation." *IEEE Trans. on Parallel and Distributed Systems*, vol.8, no.6, June 1997.

[4]D.K. Pradhan and N.H. Vaidya, "Roll-forward checkpointing scheme: A novel Fault-Tolerant Architecture," *IEEE Trans. Comput.*, vol.43,no.10,pp 1163-1174,Oct.1994.