고장 탐지 방법의 성능 분석을 위한 분산 실시간 시뮬레이션

노진홍[0] 홍영식
동국대학교 컴퓨터공학과
{jhno, hongys}@dongguk.edu

Distributed Real-time Simulation for the performance analysis of Fault Detector

Jin-Hong No[0] , Young-Sik Hong
Dept of Computer Engineering, Dongguk Univ.

요 약

신뢰성이 높은 분산 시스템은 고장 발생 시 고장을 탐지하고, 다른 관련된 노드들에게 고장을 알려주어서 적절한 처리를 할 수 있어야 한다. 기존의 ack 와 time-out 을 사용한 고장 탐지방법은 수신되지 않는 ack 에 대한 부하가 높은 단점을 가지고 있다. 높은 신뢰성을 요구하는 분산 실시간 시스템에서는 데드라인을 준수하기 위해 고장처리이 대한 time-bound 를 결정할 수 있어야 하므로, 기존의 ack 와 time-out 에 기반한 고장방법을 사용하기에 부적절하다. 따라서 본 논문에서는 신뢰성 있는 분산 실시간 멀티캐스트 프로토콜에 결합된 고장 탐지 기법으로서 사용될 기존의 고장 탐지 방법을 대상으로 고장 탐지 방법을 실험하고, 그 결과를 분석하여 고장탐지 및 고장처리 지연시간이 데드라인을 보장할 수 있는지 검사한다.

1. 서 론

신뢰성 있는 분산 시스템은 메시지의 전달이 모든 관련 노드들에게 전송할 수 있음을 보장한다. 이때 메시지가 전송되었는지 확인하기 위해 ack 와 time-out 방법을 사용하는 고장 탐지 방법을 주로 사용하지만, ack가 수신되지 않는 경우 재전송과 관련한 문제들이 발생한다. 분산 실시간 시스템에서의 통신도 높은 신뢰성을 요구하므로 고장 탐지 방법을 사용해야 한다. 하지만 시간적 제약성을 가지고 있는 분산 실시간 시스템에서 기존의 일반적인 고장 탐지 방법은 고장처리 시간을 고려한 시간 적시성(timeliness)을 보장하기 힘들다.

따라서 본 논문에서는 분산 실시간 시스템에서 사용 가능한 기존의 고장 탐지 기법을 대상으로 실험하였다. 본 연구에서 실시한 실험은 분산 환경을 지원하는 시스템 설계모델인 TMO (Time-triggered Message-triggered Object)[6]을 기반으로 하였다.

2. 관련연구

PSTR(Primary-Shadow TMO Replication)[5]은 DRB/PSP (primary-shadow active replication principle)의 방법[4]과 TMO 구조[2]를 통합시킨 방법이다. PSTR은 서로 다른 주 TMO (primary TMO)와 보조 TMO(shadow TMO)가 네트워크 상에 존재하여 PSTR 본부(station)를 구성한다. 주 TMO와 보조 TMO는 클라이언트의 요청을 함께 받아 다음과 같은 방법으로 처리한다. 주 TMO는 요청을 받았는지를 보조 TMO에게 통보하고, 정상적으로 요청이 처리가 되었는지를 보조 TMO에게 통보한다. 마지막으로 주 TMO가 처리 결과를 클라이언트에게 응답하는데 성공했는지를 통보한다. 만약 일정시간 안에 주 TMO가 요청을 못 받거나 처리를 하지 못하는 경우, 혹은 처리 실패를 보조 TMO에게 통보한 경우나 응답 통보를 하지 못한다면 보조 TMO가 처리한 작업을 대신 클라이언트에게 응답하게 된다. 이 방법은 실시간 환경에 적합하게 시간 적시성을 제공할 수 있다는 장점이 있다.

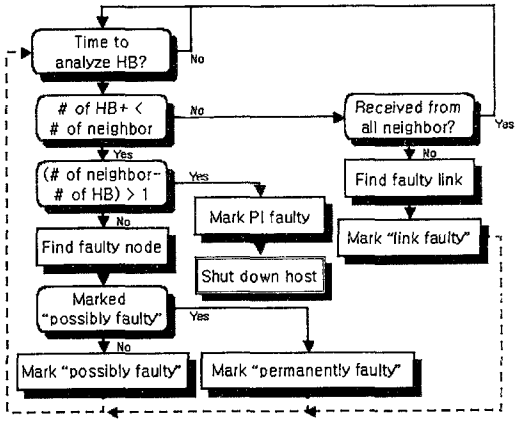
DCMS(Dynamic Configuration Management Subsystems)[3]은 NS(network surveillance), NR(network reconfiguration), OD(object distribution)의 세 가지 계층으로 구성된 고장 탐지 방법이다. NS 계층은 다른 노드들의 상태 및 연결상태를 검사하고 발생한 고장을 감지한다. NR 계층은 감독자(supervisor)와 일반 노드(worker)로 구성되어 고장 발생 시 감독자가 일반 노드에게 고장을 통보하고 시스템 설정을 바꾼다. OD 계층은 NR 계층이 설정을 바꾸거나 새로운 TMO가 생성되어야 할 때 건강한 노드들간에 새로운 TMO들을 분배하는 역할을 한다. NS 계층은 DCMS의 핵심 계층으로 그룹 멤버쉽 유지에 관련된 여러 연구가 있었다. 그 중 SNS(Supervisor-based Network Surveillance)[3]는 중앙 집중식 감독자(supervisor)를 사용하며 실시간 환경에 적용할 수 있도록 고장을 탐지하는 방법이다. SNS는 일대일 네트워 구조를 가지고 있으며 주기적으로 이웃 노드에게 heartbeat 메시지를 전송하면서 자기자신도 이웃 노드로부터 메시지를 수신하는 방법이다. 이웃 노드로부터 heartbeat 메시지가 없을 경우 프로세서, 입력 통신 처리기, 출력 통신 처리기, 통신 네트워크 중 어떤 고장이 발생했는지를 판별할 수 있다.

3. 고장 탐지 기법

실시간 시스템에 사용 가능하려면 최악의 경우(worst-case)의 고장 탐지 시간을 결정할 수 있어야 하고, 실시간 시스템에 사용 가능한 방법은 많지 않다. 그러므로 본 논문에서는 분산 실시간 환경에서 중앙 집중식 감독자를 사용하는 SNS 형태의 고장 탐지 방법의 성능 평가를 위한 분산 실시간 시뮬레이션을 하고자 한다.

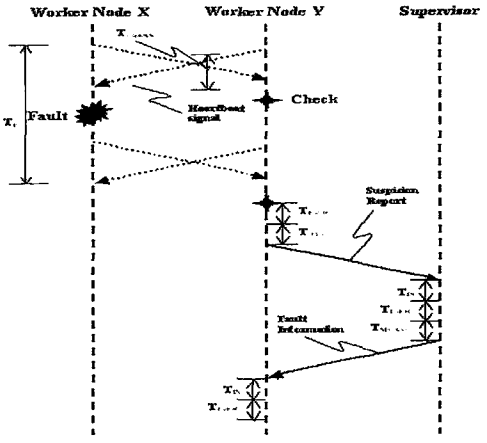
중앙 집중식 감독자를 사용하는 고장 탐지에서 모든 노드들은 자신의 존재를 알리기 위해 이웃 노드에게 주기적으로 heartbeat 시그널을 직접 연결된 링크를 통해 보내고, 추가적으

로 직접 연결되지 않은 다른 링크를 통해 한번 더 시그널을 전송할 수 있다. 또한 송신자나 수신자의 순간적인 에러(transient error)를 방지하기 위해 각각의 heartbeat 시그널은 두 번씩 중복 전송한다. 반대로 모든 노드들은 이웃 노드로부터 주기적으로 heartbeat 시그널을 수신하고 일정 기간동안 heartbeat 시그널이 도착되지 않는다면 [그림 1]과 같이 이웃 노드의 고장 종류를 감지하여 감독자에게 suspicion report를 전송한다. Suspicion report를 수신한 감독자는 해당 노드의 고장을 인식하고 이러한 고장 사실을 정상 노드들에게 전달한다.



[그림 1] 고장 분석 예

Heartbeat 시그널 주기를 T_P, 메시지 전송 시간을 T_{TRANS}, 입력정보 처리 시간을 T_{IN}, 출력정보 처리시간을 T_{OUT}, 고장 분석 시간을 T_{PROC}, 멀티캐스트 시간을 T_{MCAST}라고 가정하자. 본 논문에서 실험한 방법은 기존의 중앙 집중식 감독자 기반 고장 탐지 방법과 달리 T_P마다 heartbeat 메시지를 전송하고, T_P마다 heartbeat 메시지의 수신을 체크하고, 전송과 체크의 시간 차이는 $0.5\times T_P$이다. 실험에서 두 노드와 감독자 사이의 메시지 송/수신에 따른 시간 도표는 [그림 2]와 같다.



[그림 2] 시간 도표

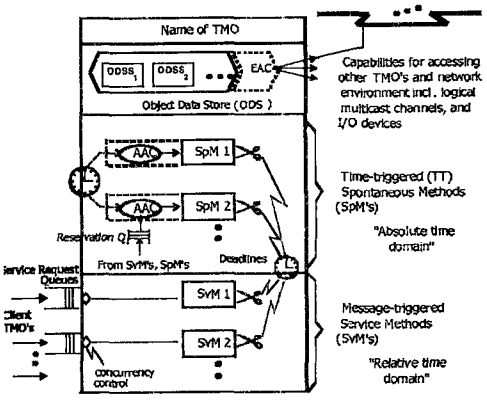
이 방법에서 출력 통신 처리기나 프로세서의 고장 탐지 시간이 가장 크게 되는 최악의 경우(worst-case)는 heartbeat 시그널을 보낸 직후에 고장이 발생하는 경우이다. 고장 발생 후 이웃 노드들이 고장을 감지하는 시간외에도 감독자에게 suspicion report를 전송하고, 감독자가 다른 노드들에게 고장정보를 전송하는 시간이 추가적으로 소요된다. 이 경우의 고장 탐지 시간(T_{F1})은 다음과 같다.

$$T_{F1} = 1.5\times T_P + 3\times T_{PROC} + 2\times(T_{TRANS} + T_{IN}) + T_{OUT} + T_{MCAST}$$

이와 비슷하게 입력처리기나 통신네트워크 고장의 고장 탐지 시간이 가장 크게 되는 최악의 경우는 heartbeat 시그널을 받은 직후 노드나 네트워크에 고장이 발생하는 경우이다. 이 경우의 고장 탐지 시간(T_{F2})은 다음과 같다.

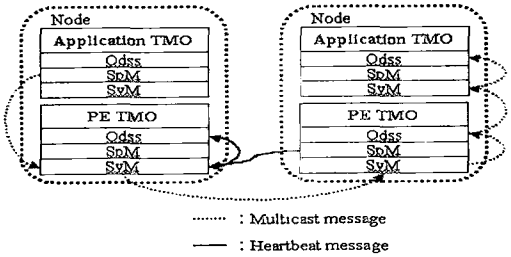
$$T_{F2} = 1.5\times T_P + 3\times T_{PROC} + 2\times T_{IN} + T_{TRANS} + T_{OUT} + T_{MCAST}$$

고장 탐지 방법의 시뮬레이션을 위해서 본 연구에서는 TMO(time- triggered message-triggered object) 모델을 기본 모델로 채택하였다. [그림 3]과 같이 TMO 모델은 세 가지 객체로 구성되어 있다. SpM(spont aneous method)은 주어진 시간 조건 AAC(automatic activation condition)가 만족되면 자동으로 활성화되는 객체이다. SvM(service method)은 이벤트에 반응하는 객체로서 외부로부터의 메시지 수신을 처리한다. 또한 ODSS(object data store)는 SpM과 SvM사이의 데이터 공유 및 동기화의 역할을 수행하고 있다.



[그림 3] TMO 시뮬레이션 모델 [6]

고장 검출을 위한 시뮬레이션 모델은 TMO 모델을 사용하여 작성되고, [그림 4]와 같이 각 노드는 Application TMO와 Protocol Engine TMO(PE TMO)로 구성된다.



[그림 4] 노드 구조

Application TMO SpM은 멀티캐스트 메시지를 생성하고 전

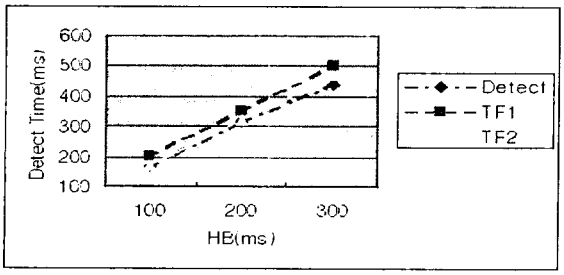
송하며, TMO SvM은 멀티캐스트 메시지를 수신하여 ODSS에 저장한다. 그리고 Application TMO에 독립적으로 PE TMO는 고장 처리를 담당한다. 주어진 시간마다 실행되는 PE TMO의 SpM이 주기적으로 heartbeat 시그널을 이웃 노드에게 전송하고, 이웃 노드의 PE TMO SvM이 heartbeat 시그널을 수신하고 이를 PE TMO ODSS에 저장한다. 그리고 주기적으로 PE TMO SpM이 PE TMO ODSS를 조사하여 이웃 노드의 heartbeat 시그널이 수신되었는지 체크한다.

4. 실험 및 성능분석

고장 탐지 기법을 실험하기 위해서 Pentium III PC 2대 및 윈도우즈 2000 및 TMO 미들웨어 상에서 실험하였다. 기타 실험에 사용된 매개변수는 다음과 같다.

노드 개수	4
SpM 활성화 주기	50 ms
SpM 종료 시간	1000 sec
노드의 고장	SpM 활성화 횟수의 1%
멀티캐스트	SpM 활성화 횟수의 1%

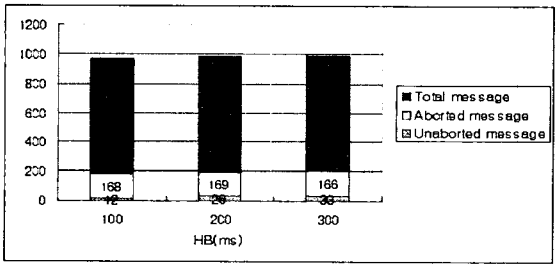
[그림 5]는 각 노드에서 송신하는 heartbeat 시그널의 주기에 따라 실제 고장이 탐지되는 평균 시간을 나타낸다.



[그림 5] 고장 탐지 시간

실험 결과에서는 T_{TRANS}=20ms, T_{IN}=T_{OUT}=1ms, T_{PROC}=T_{MCAST}=2ms의 값을 가졌다. 그러므로 heartbeat 시그널의 주기가 100ms, 200ms, 300ms일 때 T_{F1}은 201ms, 351ms, 501ms이고, T_{F2}는 171ms, 321ms, 471ms이다. [그림 5]를 보면 평균 탐지 시간이 이보다 작은 값인 166ms, 314ms, 443ms인것을 알 수 있다.

[그림 6]은 총 멀티캐스트 개수와 이 중 고장이 발생하여 멀티캐스트를 취소한 개수, 그리고 취소하지 못한 멀티캐스트의 개수를 나타내고 있다.



[그림 6] 멀티캐스트 메시지

Heartbeat 시그널의 수신을 체크하여 고장이 없다는 것을 확인한 후에 멀티캐스트를 시작하고 바로 고장이 발생한 경우에는 이미 전송한 멀티캐스트를 취소할 수 없다. 그러므로 heartbeat 시그널의 송신 주기나 수신체크 주기가 길수록 취소하지 못한 멀티캐스트의 개수가 증가하게 된다. 그러나 잦은 heartbeat 시그널의 송/수신은 각 노드에서의 부하를 발생시키게 되는 단점을 생각할 수 있다.

5. 결론

본 논문에서는 분산 실시간 시스템을 위한 고장 탐지 기법 중 중앙 집중식 감독자를 사용하는 고장 탐지 방법을 TMO 미들웨어를 적용하여 실험하고 그 결과를 분석하였다. 실험 결과를 통해 사용된 고장 탐지 방법이 실시간 환경에서 사용 가능하도록 처리시간을 보장하는지 알아보고, 고장 처리 시간이 얼마나 소요되는지 확인하였다. 실험에서 고장 탐지 시간과 멀티캐스트 취소 메시지의 수를 측정하였다.

향후 과제로는 실험한 고장 탐지 방법을 구현하여 그룹 통신 시스템에 통합할 것이다. 그리고 구현된 고장 탐지 방법의 실제 성능을 평가해 보아야 할 것이다.

참고문헌

[1] M. Grossglauser, "Optimal Deterministic Timeouts for Reliable Scalable Multicast", IEEE INFOCOM '96, San Francisco, CA, March 1996

[2] K. H. Kim, "Object Structures for Real-Time Systems and Simulators", IEEE Computer, August 1997, pp.62-70

[3] K. H. Kim and C. Subbaraman, "Dynamic Configuration Management in Reliable Distributed Real-Time Information Systems", IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 1, 1999

[4] K. H. Kim and C. Subbaraman, "An Integration of the Primary-Shadow TMO Replication Scheme with a Supervisor-based Network Surveillance Scheme and its Recovery Time Bound Analysis", Proc. IEEE CS 17th Symp. on Reliavle Distributed Systems (SRDS '98), West Lafayette, IN, 1998

[5] K.H. Kim and C. Subbaraman, "The PSTR/SNS Scheme for Real-Time Fault Tolerance via Active Object Replication and Network Surveillance", *IEEE Trans. on Knowledge and Data Engr.*, Vol.12, No.2, Mar./April 2000, pp.145-159.

[6] K.H. Kim, Chittur Subbaraman, Masaki Ishida, Jaqiang Liu, "TMO Support Library(TMOSL): Facilities for C++ TMO Programming", Univ. of California, Irvine, 2000

[7] A.C. Liang, S. Bhattacharya, and W.T. Tsai, "Fault-Tolerant Multicasting on Hypercubes," J. Parallel and Distributed Systems, pp. 405-412, France, Sept. 1996