이중화 프로세서에 대한 마코프 모델의 구현

구중두
호원대학교 컴퓨터게임학부
e-mail:ygslee@sunny.howon.ac.kr

Implementation of Markov Model for Duplication Processor

Jung Du Goo
Division of Computer Games, Howon University

요 약

이동통신시스템에서 Warm standby sharing에 비하여 Hot standby sharing은 데이터 손실이 없고 오류 데이터가 확산되지 않는 등의 다수의 장점을 갖지만 동기화 문제로 인하여 이를 시스템에 실제로 구현하는 것은 어렵다. 따라서 본 연구에서는 Hot standby sharing에 비하여 기존의 Warm standby sharing이 갖는 동기화의 장점에 데이터 손실 및 거짓 데이터의 확산 문제를 개선할 수 있는 이중화 프로세서에 대한 마코프 모델을 설계하고자 한다.

1. 서론

산업 및 사회 전반에 무선 및 유선 통신의 의존도가 높아 감에 따라 복잡하고 다양한 형태의 고성능 통신 시스템이 개발되고 있고, 한 부분으로서 통신 시스템의 안정성에 대한 관심이 고조되고 있다. 특히 개발 중에 있는 이동통신시스템에서 RNC(Radio network Controller)는 무선 자원을 제어하는 시스템으로 RNC를 구성하는 MCP(Main Control Processor)의 결함 포용 능력은 전체 시스템의 안정성을 좌우하는 중요한 변수가 된다. 시스템의 안정성 확보 및 서비스 지속성 같은 고신뢰성을 달성하려면, MCP를 이중화 혹은 다중화 함으로써 활성화된 프로세서가 장애를 일으키더라도 대기중인 프로세서가 연속적인 서비스를 제공할 수 있도록 해야 한다. 이는 교환 시스템으로의 호 처리 요청이 끊임없이 이루어지기 때문에 시스템의 일시적인 동작 중지는 사용자들에게 극도의 혼란을 야기시킬 수 있기 때문이다. 그러므로 기존의 교환시스템에서는 결함 감내 구조로서 Standby sharing 기법을 사용하고 있다.

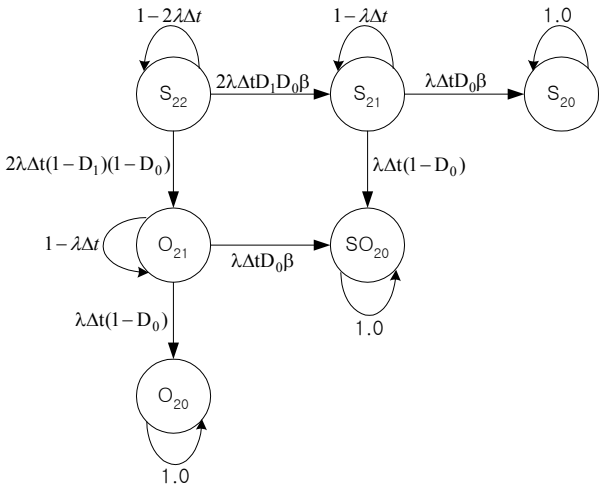
Standby sharing 기법은 크게 세가지로 구분되는데 Cold standby sharing, Warm standby sharing, Hot standby sharing이다[1,3]. Cold standby sharing 기법은 Standby side가 결함 발생으로 인해 Active 상태로 전환되기 전까지 전원 공급이 중단되어 있으므로 Active 기능을 수행하기까지에는 다소의 시간이 걸린다. 그러므로 고가용성을 요구하는 교환기의 결함 감내 시스템에는 이 기법이 적합하지 않다. Warm standby sharing 기법은 Active side가 시스템 정상 동작 시에 Concurrent write 방식을 사용하여 그 자신의 메모리 내용과 Standby side의 메모리 내용이 동일하도록 시스템을 동작시키기 때문에, 결함 발생 시에 Standby side가 Active 상태로 전환되고 본래의 정상 기능을 수행하는데 걸리는 시간은 Cold standby sharing 기법보다 매우 짧다.

그러나 결함의 종류와 정도에 따라서 다소의 데이터 손실(Data loss)이 발생하며 Active side와 Standby side간에 주기적인 결함 점검이 요구된다. Hot standby sharing 기법에서는 두 side가 active하게 동작하므로 정상 동작 시 모든 시스템 모듈의 상태와 내용을 시스템 동기화를 통해서 동일하게 유지해야 한다. 이 기법에서는 외부 시스템과의 데이터 교환 시 한 모듈만이 Active로서 동작하게 된다. 임의의 한 모듈에서 결함이 발생하면, 결함 모듈을 시스템으로부터 제거하고 수행 중인 일을 계속 진행시킬 수 있으므로 결함 감지로부터 시스템 정상기능 재가동까지 걸리는 시간이 극히 짧다. 또한 단일 시

스템 결함으로부터의 데이터 손실이 발생하지 않는 결함 감내 구조의 설계가 가능하다. 반면에 Warm Standby에 비하여 정상 동작 중에 결함 감내 시스템 모듈간의 동기화 유지가 어렵고 결함에서 복구된 시스템 모듈의 재정상 가동 등의 구현의 어려움이 해결해야 할 큰 문제점으로 지적되고 있다[4,6]. 그러므로 Hot standby sharing에 비하여 기존의 Warm standby sharing이 갖는 동기화의 장점에 데이터 손실 및 거짓 데이터의 확산 문제를 개선할 수 있는 이중화 프로세서에 대한 마코프 모델을 설계하고자 한다.

2. 이중화 프로세서에 대한 마코프(Markov) 모델

B'd에서 오류가 발생할 확률 λ는 지수분포에 의한 hazard 함수 Z(t)=λ이다. 상대 보드 결함 탐지 확률(Mate Fault Detection Probability) D_1이며, 자기 보드 상태 탐지 확률(My Fault Detection Probability)은 D_0, 시간 증가 인자는 Δt 그리고 결함 회복 확률은 β, 여기서 결함 회복 확률은 오류를 탐지한 후에 손실 없이 시스템을 정상 상태로 회복시킬 수 있는 확률을 말한다. 제안된 시스템의 마코프 모델은 그림 1과 같다.



[그림 1] 확장된 Warm standby의 Markov 모델

본 연구에서 제시한 시스템은 Markov 모델을 적용할 경우 6개의 시스템 상태를 나타낸다. 시간 $t+\Delta t$에서 시스템이 임의의 상태 S일 확률은 시스템이 임의의 상태에서 상태 S로 전이할 수 있는 확률과 자신의 상태를 계속 유지할 확률로 정의할 수 있다. 이 경우, 제안된 모델을 수식화하면 다음과 같이 정의할 수 있다.

상태 $P_{S_{22}}(t+\Delta t)$는 두 B'd가 하드웨어적으로 완전한 상태로서 호 처리 및 백업이 완전하게 수행중인 상태를 나타낸다. 이는 Markov 모델에 의하여 다음과 같이 나타낼 수 있다.

$$P_{S_{22}}(t+\Delta t) = (1-2\lambda\Delta t)P_{s_{22}}(t) \quad (1)$$

상태 $P_{S_{21}}(t+\Delta t)$는 Active side에서 오류가 발생하고 Standby side가 성공적으로 Active로 실행된 경우, Standby side에 오류가 발생했지만 오류가 검출된 상태를 나타낸다.

$$P_{S_{21}}(t+\Delta t) = 2\lambda\Delta t D_1 D_0 \beta P_{s_{22}}(t) + (1-\lambda\Delta t)P_{s_{21}}(t) \quad (2)$$

상태 $P_{S_{20}}(t+\Delta t)$는 Active side와 Standby side에서 오류가 발생하고 오류가 검출된 상태를 나타낸다.

$$P_{S_{20}}(t+\Delta t) = \lambda\Delta t D_0 \beta P_{s_{21}}(t) + P_{s_{20}}(t) \quad (3)$$

상태 $P_{O_{21}}(t+\Delta t)$는 Active side 또는 Standby side에서 오류가 발생했지만 검출하지 못한 상태를 나타낸다.

$$P_{O_{21}}(t+\Delta t) = 2\lambda\Delta t(1-D_1)(1-D_0)P_{s_{22}}(t) + (1-\lambda\Delta t)P_{0_{21}}(t) \quad (4)$$

상태 $P_{O_{20}}(t+\Delta t)$는 Active side와 Standby side에서 오류가 발생했지만 검출하지 못한 상태, 그리고 Active side에서 오류가 발생했지만 Standby side의 사용이 실패한 상태를 나타낸다.

$$P_{O_{20}}(t+\Delta t) = \lambda\Delta t(1-D_0)P_{O_{21}}(t) + P_{O_{20}}(t) \quad (5)$$

상태 $P_{SO_{20}}(t+\Delta t)$는 Active side의 오류 검출 실패, Standby side의 오류 검출 실패, Active side에서 오류가 발생했지만 Standby side의 사용이 실패(Standby side의 오류 검출 실패)한 상태를 나타낸다.

$$P_{SO_{20}}(t+\Delta t) = \lambda\Delta t(1-D_0)P_{S_{21}}(t) + \lambda\Delta t D_0 \beta P_{O_{21}}(t) + P_{SO_{20}}(t) \quad (6)$$

시스템의 신뢰도(Reliability Degree)는 시스템이 안전하게 동작 가능한 상태로서 식 (7)과 같은 확률로 나타낼 수 있으며, 안전도(Safety Degree)는 B'd에서 오류가 발생하여 기능을 중단하는 경우에도, 효과적인 방법으로 오류를 취급하여, B'd가 안정 상태를 유지하는 것을 말한다. 이는 식 (8)과 같다.

$$R_w(t) = P_{S_{22}}(t) + P_{S_{21}}(t) + P_{O_{21}}(t) \quad (7)$$

$$S_w(t) = P_{S_{22}}(t) + P_{S_{21}}(t) + P_{S_{20}}(t) + P_{O_{21}}(t) + P_{O_{20}}(t) \quad (8)$$

3. 결론

본 연구에서는, 각 프로세스가 메시지 단위로 동작하면서 갱신된 데이터를 Standby side에 전달하는 Task기반 이중화방식을 기본으로, 기존의 Warm standby 구조가 갖는 데이터 손실 및 거짓 데이터의 확산 문제를 해결함으로서 성능의 향상을 도모하였다.

참고 문헌

1) D. P. Siewiorek , "Architecture of fault tolerant computers : an historical perspective," Proc. IEEE, Vol. 79, No. 12, 1710~1734, Dec. 1991.
2) J. C, Laprie, et al., "Definition and Analysis of Hardware and Software Fault - Tolerant Architectures," Computer, pp.39~51, July 1990.
3) A.M. Tyrrell and G.F. Carpenter, "CSP Methods for Identifying Atomic Actions in the Design of Fault Tolerant Concurrent Systems," IEEE Trans.on SE, Vol.21,No.7, pp.59 68, Jul. 1995
4) L. Young, "Software Fault Tolerance at the Operating System Level," Proc.of COMPSAC '95, pp.393, 1995
5) M. Russinovich, and Z. Segall, "Fault Tolerance for Off The Shelf Applications and Hardware," Proc.of FTCS 25, pp.67 71, Jun. 1995
6) B.J. Min, S.S. Shin and K. W. Rim, "Design and Analysis for a Multiprocessor System wit h Extended Fault Tolerance," IEEE FTDCS., pp. 301 307, Aug. 1995.