네트워크상에서 프로세서 이중화 시스템의 구현

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

Implementation of Processor Duplication System in Network

Jung Du Goo
Division of Computer Games, Howon University

요 약

이동통신시스템에서 RNC의 MCP는 호 처리를 담당하는 부분으로, 신뢰도와 실시간성이 요구된다. MCP는 높은 견고성을 갖도록 구현되지만 다소간의 오류 율(Fault late)은 존재할 수밖에 없으므로 프로세서를 이중화하여 활성화된 프로세서가 장애를 일으키더라도 대기중인 프로세서가 연속적인 서비스를 제공할 수 있어야 한다. 따라서 본 연구에서는 이동통신에서 프로세서 이중화 시스템을 설계하고자 한다.

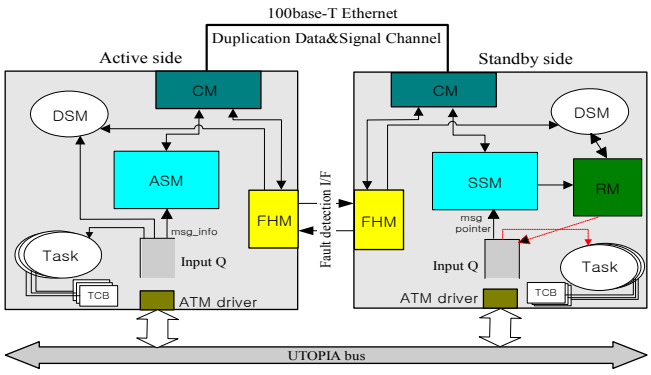
1. 서론

오늘날 고가용성을 지원하는 시스템들은 Warm 및 Hot standby 결함 감내 구조를 기본으로 채택하고 있다. Warm standby 구조는 시스템 구현이 Hot standby 구조에 비해 훨씬 용이하나 결함의 종류와 정도에 따라서 다소의 데이터 손실이 발생하며 현재 동작중인 모듈상에 감지되지 못한 오류 데이터(Fault data)가 궁극적으로 전체 시스템으로 확산되는 문제점을 지니고 있다. 그러나 현재 개발되어 사용되고 있는 AT&T사의 ESS 5 교환기, ETRI의 TDX 10 등의 대부분의 교환기들은 결함 감내 구조로서 Warm standby sharing 기법을 적용하고 있다. 그 이유는 결함 감내 구조로서 Hot standby Sharing 기법을 사용하는 교환 시스템은 데이터 무손실 등 많은 장점을 가지고 있으나, 동기화의 복잡성으로 인하여 실시간 교환 시스템으로의 구현 시에 이론적인 성능을 얻는 것이 어렵기 때문이다[7,10].

본 연구에서는 이중화 구조를 갖는 RNC에서 확장된 Warm standby sharing 결함 감내 구조를 구현함으로써 이동통신에서 프로세서 이중화 시스템을 설계하고자 한다.

2. 시스템 구조

이중화 시스템의 프로세서는 RNC의 핵심 Controller로 System의 안정화를 위하여 이중화 되어야 한다. 이중화를 위한 Active side와 Standby side간에 데이터 및 제어 신호를 신뢰성 있게 전송하기 위하여, 이중화 path는 100base T Ethernet으로 구축하며 통신 모듈에서 관리한다. 동기화를 위하여, 소프트웨어 간의 통신으로 D/B의 내용을 일치시킨다.



[그림 1] 이중화 시스템의 구조

통신 방식으로는 직렬 인터페이스(Serial Interface)로 구현하고, 100Mbps Fast Ethernet 상으로 전송

한다. 그림 1에 본 연구에서 제시한 이중화 시스템 구조를 보인다.

이중화 지원 모듈(DSM; Duplication Support Module)은 이중화 블럭의 전체적인 상태(state)를 가지고 있으며, 이를 이용하여 다른 module을 control하고, 상위 프로세서와 통신을 담당한다. 동기화 모듈(SM; Synchronization Module)은 Active side와 Standby side 사이의 상태 동기를 맞추기 위한 모듈로 Active 동기화 모듈(ASM; Active Synchronization Module)과 Standby 동기화 모듈(SSM; Standby Synchronization Module)로 구성되며 Active와 Standby에서 서로 다르게 동작한다. 동작 mode는 control module에서 지정하는 "system state"를 참조한다. ASM에서는 응용 프로그램(AP; Application Program)으로부터 Backup data와 Dump data를 받고, IPC로부터는 Message information을 받아 이들을 직렬화(serialize)하여, 통신 모듈(COM; Communication Module)로 전송한다. SSM은 Active side에서 전송된 Backup data/Dump data 및 Message info를 COM으로부터 받아 처리하고, IPC로부터 Standby side로 입력된 외부 Input message를 logging 및 이에 대한 Garbage collection을 수행한다. Input message information는 각 IPC(Inter process Communication)를 통해 받은 메시지의 순서를 결정하는데 사용하며, Backup data는 응용 프로그램들의 Memory update 및 Message garbage collection에 사용한다. Data dump는 Standby가 실장된 초기에 Active와 Standby의 상태를 일치시키기 위한 것으로, Dump data를 이용하여 응용 프로그램의 메모리를 갱신한다.

장애 처리 모듈(FHM; Fault Handling Module)의 역할은 오류를 감지하여 적절한 조치를 취하는 것으로 이때 감지해야 할 오류는 상대 B'd(Board)의 치명적인 오류와 자기 B'd에서의 자원의 상태 변화이다. 상대 B'd의 오류 발생을 탐지하기 위한 모듈은 ISR(Interrupt Servive routine)로 구현한다. FD 인터럽트가 발생하면, 이에 해당하는 ISR이 FD register를 읽고, 메시지에 상대 B'd의 오류 내용을 담아 이중화 지원 모듈로 전송하여, 적절한 동작을 요구한다.

현재 구축중인 이중화 방식은 Active side과 Standby side에 적재되는 각 AP는 초기화 시에 이중화 모듈로 자신의 기억장소 영역을 등록한다. 이 과정은 Active side와 Standby side가 동일한 절차로 수행한다. 이후에 Active side 및 Standby side는 동시에 Call message를 받는다. Active side에 있는 IPC단은 이 메시지를 해당 AP로 전송하여 처리되도록 하고 동시에 Standby side에 메시지에 대한 정보를 전송한다. 그리고 메시지 처리 결과로 인해 기억장소의 내용이 변경되었다면 Backup message(변경된 내역 및 그 메시지에 대한 정보)를 이중화 모듈로 전송한다. 이중화 모듈은 이를 AP별로 관리하다가 그 AP가 checkpoint를 호출하면 이를 Standby side의 이중화 모듈로 전송한다. Checkpoint를 두는 이유는 AP가 Backup message를 관리할 수 있도록 융통성을 부여하는 것이다.

Standby side에서는 IPC단에서 외부 Tx를 막고, 외부로부터 입력된 메시지에 대해서도 해당 AP로 전송하지 않고 큐 형태로 저장한다. 그리고, 효율적인 메시지 관리 차원에서 각 메시지들에 대해서 AP ID를 키로 하는 해쉬 테이블을 작성하여 관리한다. Active side에서 메시지에 대한 정보를 보내오면 이를 AP별로 관리하는 해쉬 테이블에 삽입하고, Data backup message를 받으면 자신의 기억장소에 이를 반영한다. 그리고, Backup message와 함께 전송되어온 메시지 정보를 이용해서 해쉬 테이블에서 일치하는 해당 AP를 조사해서 일치하는 메시지를 찾아 그 메시지까지 이전의 모든 메시지를 삭제한다.

복구 모듈(RM; Recovery Module)은 Active side의 오류를 감지한 후, Standby side였던 B'd에서 수행되는 모듈로, 이전 Active side가 수행되었던 곳까지를 follow up한다. 이전의 Active side에서 Data backup이 완전이 수행되지 않은 경우, 새로이 Active가 된 side는 현재 백업되어 있는 데이터를 바탕으로, 절체된 순간 이전까지 Input queue에 저장된 Input message를 처리하여 절체되기 전의 Active가 수행했던 상태를 복구하며, IPC message의 유실을 방지한다.

3. 결 론

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 해결 함으로서 성능의 향상을 도모하였다. Warm standby는 Hot standby에 비해 구현이 용이하나 결함 발생 시의 데이터 손실 문제 그리고

오류 데이터가 시스템으로 확산되는 문제점을 가지고 있었다. 반면에 Hot standby 구조는 데이터 무손실 등의 다수의 장점이 있지만 이중화 모듈간에 빈번한 동기로 인한 전체 시스템 성능 저하 문제로 인하여 구현 시 이론적인 성능을 얻기 어렵다.

본 연구에서는 Warm standby의 동기화의 단순함에 따른 구현의 용이성이라는 측면에 기존 Warm standby의 단점을 개선함으로서 효율적인 이중화 시스템을 설계하였다.

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.