

TMR 시스템 기반의 Checkpointing 기법에 관한 연구

김태욱, 강명석, 김학배
연세대학교 전기전자공학과
e-mail: united, mskang, hbkim@yonsei.ac.kr

A Study for Checkpointing Schemes based on a TMR System

Taewook Kim, Myungseok Kang, Hagbae Kim
Dept of Electrical and Electronic Engineering, Yonsei University

요 약

TMR(Triple Modular Redundancy)은 공간여분(H/W 및 S/W)을 정적으로 활용하는 가장 간단한 구조를 지닌 대표적인 고장포용 기법중의 하나이다. TMR 구조 고장시 TMR 시스템 고장복구를 위해 잘못된 결과를 가지고 있는 프로그램의 일부분을 재실행 또는 프로그래밍 전체를 재시작하는 기법을 적용하는 것은 일반적으로 상당한 시간을 필요로 한다. 이러한 단점을 극복하기 위해 본 논문에서는 TMR 고장을 효과적으로 복구하기 위해 또 다른 형태의 시간여분 기법인 rollback과 roll-forward 기법에 checkpoint들을 적용하여 처리하는 시간 및 공간여분을 혼용하는 기법을 제안하였다.

1. 서론

TMR 시스템은 공간여분(H/W 및 S/W)을 사용하여 신뢰도를 향상시키는 목적에 널리 활용되는 대표적인 고장포용기법중의 하나이다. 일반적으로 safety-critical 시스템과 같은 엄격한 실시간 제어 시스템은 상당히 높은 신뢰도를 요구하기 때문에, 이러한 고 신뢰도를 만족시키기 위해서 제어기의 구조는 점점 복잡해져가고 있는 상황이다. 한편 retry, restart, rollback[1] 및 roll-forward[2] 기법은 모두 신뢰도를 향상시키기 위해 시간 여분을 사용하며 널리 활용되는 고장포용 기법들이다. 이러한 기법들을 정적여분(static redundancy) 활용방법인 TMR 시스템에 적용하여 시간 및 H/W 여분을 혼용[3][4][5]하면 일시적인 결함(fault)으로 인해 일어나는 TMR 고장(failure)이나 차폐오류(masked error)를 효과적으로 복구시킬 수 있다. 본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 H/W, S/W 및 시간 여분을 바탕으로 하는 여분 시스템에 관해 설명하고, 3장에서는 TMR 시스템의 공간여분과 시간여분 기법에 checkpoint들을 사용한 check pointing 기법에

대하여 제안하였다. 마지막으로 결론 및 추후 연구 방향에 대하여 기술하였다.

2. 관련연구

앞에서 언급된 TMR 시스템은 고장포용을 위한 시스템에서 H/W 여분을 활용한 대표적인 시스템이다. 보다 구체적인 고장포용 시스템과 H/W, S/W 및 시간 여분을 바탕으로 하는 여분 시스템에 관해 설명하겠다.

2.1 고장포용 시스템

고장포용이란 시스템의 일부 요소에 고장이 일어나더라도 전체 시스템이 잘못된 행동을 일으키기 전에 고장을 회복시키거나 고장의 효력을 제거하여 안전한 작업수행을 보장할 수 있게 하는 기법을 말한다.

고장포용기법의 설계시의 기본개념은 시스템에 여분(redundancy)의 자원(resource)을 제공하는 것이다. 여분은 종류에 따라 H/W, S/W 및 시간 여분이 있다.

2.1 H/W 여분 시스템

H/W 여분 시스템은 동일한 작업을 할 N개의 프로세서나 모듈을 구성하여 고장이 발생하더라도 고장차폐를 통해서 고장과 관계없이 작업수행을 완결할 수 있도록 하거나 또는 스페어를 추가설치해서 고장이 발생할 경우 적절한 스위칭을 통해서 정상적인 작업수행을 도모한다.

2.2 S/W 여분 시스템

S/W 여분을 응용한 시스템으로 N-version programming 기법과 recovery block approach 기법이 있다. N-version programming 기법은 동일한 작업을 수행하기 위해 여러 개의 S/W를 구현해서 병렬로 작업을 수행한 후 그 출력 값을 voting하는 기법이다. Recovery block approach 기법은 적절한 acceptance test를 통해서 작업수행 결과의 타당성을 검토한 뒤 적절하지 않은 경우에 또 다른 version으로 작업을 수행하는 기법이다.

2.3 시간 여분 시스템

앞에서 제시한 방법들이 전방에러회복(forward error recovery) 방법이라면, 시간 여분 시스템은 후방에러회복(backward error recovery) 방법이다. 후방에러 회복방법에서 고장이 발생했을 때 수행되어진 작업을 무시하고 다시 수행하는 retry, 모든 작업을 다시 수행하는 restarting, 회복기점으로 회귀하는 rollback 방법이 있다.

2.4 TMR 시스템

TMR 시스템은 가장 대표적이면서도 간단한 H/W 여분 시스템으로 세 개의 동일한 replication의 실행 결과를 받아 이들의 값을 서로 비교하여 세 개 중 두개 이상의 결과가 같으면 이를 출력으로 발생시키는 보팅(voting) 작동에 기초한다. 이러한 보팅 기법을 사용할 때 임의의 한 모듈의 값이 잘못되더라도 나머지 두개의 값이 옳은 값이라면 잘못된 결과를 가지는 모듈의 영향은 무시될 수 있다. 이러한 에러를 차폐오류(masked error)라 한다. 그림 1은 대표적인 TMR system 모델이다.

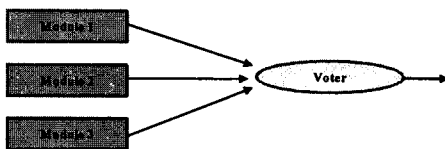


그림 1 TMR 시스템

3. TMR 시스템 기반의 Checkpointing 기법

TMR 시스템에서 두 개 이상의 모듈에서 잘못된 출력 값이 나와 보팅(voting)을 할 수 없는 상황을 TMR 고장이라 한다. TMR 시스템에서 TMR 고장 복구를 위해 잘못된 모듈의 일부분을 재실행(retry)하거나 전체를 재시작(restart)하는 방법은 상당한 시간이 필요로 한다. 이러한 단점을 해결하기 위해 재실행이나 재시작과는 다른 시간여분 기법인 rollback과 roll-forward를 적용한 checkpointing 기법을 소개한다.

TMR 시스템에서 시간여분 기법을 적용하기 위해서 checkpointing을 위한 추가적인 저장 공간이 필요하다. 저장 공간에 고장이 발생할 확률은 무시할 만큼 작다고 가정한다.

3.1 TMR-Checkpointing-1

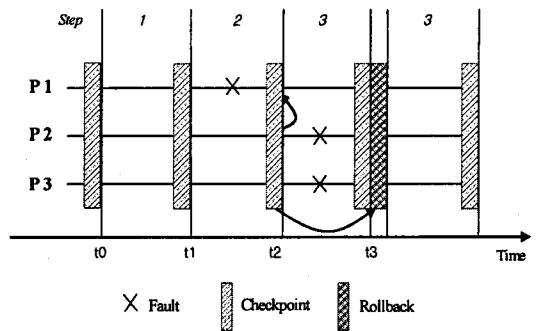


그림 2 TMR-Checkpointing-1

그림 2에서 TMR 고장이 발생했을 때 checkpointing 기법을 적용하는 가장 기본적인 예이다. 각각 실행 간격을 Step으로 나눴다. Step 2에서 하나의 고장 발생하였을 경우 일반적 TMR시스템처럼 무시할 수 있음을 보였다. Step 3에서 TMR 고장이 발생하는 경우 그 이전 checkpoint에서 rollback을 함을 보이고 있다.

3.2 TMR-Checkpointing-2

Checkpointing을 하는 TMR 시스템에서 모든 모듈에 저장 공간을 두고 같은 vote를 행하는 것은 낭비가 될 수 있다. 다음에 제안하는 방법은 이 같은 낭비를 줄이고자 하였다.

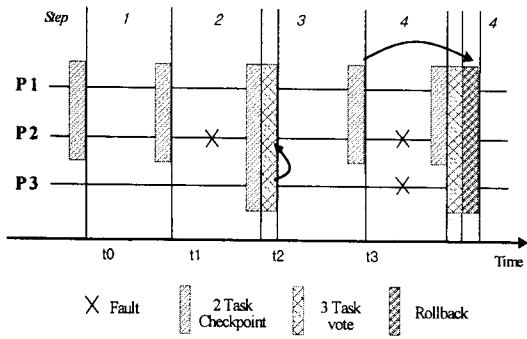


그림 3 TMR-Checkpointing-2

그림 3에서 3개의 모듈 중 3개의 모듈만 저장 공간을 두어 checkpoint를 하고, 고장이 발생하였을 때 세 번째 모듈과 비교하여 고장 모듈을 찾도록 하였다. step 4에서는 TMR 고장이 발생하였을 경우 그 이전에 checkpoint에서 rollback을 함을 보이고 있다.

3.3 TMR-Checkpointing-3

TMR 고장이 일어날 확률이 매우 작다면 세 모듈을 동시에 실행하는 것은 비효율적일 수 있다. 이 경우 voter의 기능과 checkpoint 프로세서의 기능을 세 번째 모듈에 두고, 고장 발생시 직접 자신이 roll-forward 될 수 있도록 하였다.

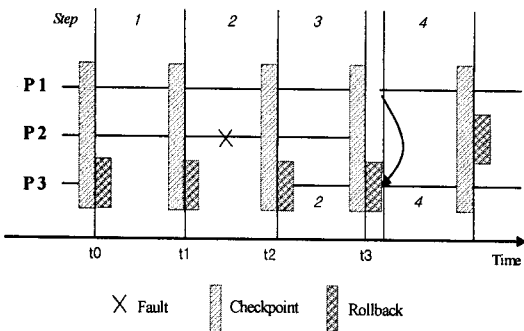


그림 4 TMR-Checkpointing-3

그림 4의 system에서 세 번째 모듈은 평소에는 vote의 기능을 수행한다. Step 2에서 고장이 발생하였을 때 세 번째 모듈은 step 2를 실행하여 다음 step에서 고장을 찾아 roll-forward 기법을 적용 하게 된다. TMR 고장이 발생하였을 경우 다음 스텝에서 rollback 기법을 적용 하게 된다.

4. 결론

본 논문에서는 고장 포용 시스템과 H/W, S/W, 시간 여분 시스템의 정의에 대하여 살펴보고, TMR 시스템에 대하여 알아보았다.

TMR 시스템은 간단한 구조를 지닌 대표적인 고장 포용 기법중의 하나이다. 본 논문에서는 TMR 시스템에서 TMR 고장복구를 위해 rollback과 roll-forward를 이용한 세 가지 checkpointing 기법을 제안하였다. 첫 번째 방법은 TMR 시스템에 checkpoint를 위한 저장 공간을 두고, TMR 고장 발생시 rollback 기법으로 고장 복구를 하는 시스템이다. 두 번째 방법은 모듈의 낭비를 줄이고 voting을 효과적으로 하기 위하여 제안된 방법이다. 마지막 방법은 세 번째 모듈에 비교기능을 주고 TMR 고장 발생시 roll-forward 기법을 사용하여 고장 복구를 하는 시스템이다. 향후 시뮬레이션을 통해 TMR 시스템 checkpointing 기법의 성능을 검증할 것이다.

참고문헌

- [1] Avi Ziv, Jehoshua Bruck, "Performance Optimization of Checkpointing Schemes with Task Duplication", IEEE Transaction on Computers, December 1997, vol. 46, no. 12, pp1381-1386.
- [2] Dhiraj K. Pradhan and Nitin H. Vaidya, "Roll-Forward and Rollback Recovery Performance-Reliability Trade-Off", IEEE Transaction on Computers, March 1997, Vol. 46, No. 3, pp. 372-378
- [3] K.G. Shin, H. Kim, "A Time Redundancy Approach to TMR Failures Using Fault-State Likelihoods", IEEE Transaction on Computer, October 1994, vol. 43, no. 10, pp1151-1162.
- [4] H. Kim, K.G. Shin, "Design and analysis of an optimal instruction retry policy for TMR controller computers", IEEE Transaction on Computers, 1996, 45, (11), pp. 1217-1226
- [5] J. Yoon, H. Kim, "Time redundant recovery policy of TMR failures using rollback and roll-forward methods", IEE Proc.-Comput. Digit. Tech, March 2000, Vol. 147, No. 2, pp. 124-132
- [6] P.K. Chande, A.K. Ramani, and P.C. Sharma, "Modular TMR Multiprocessor System", IEEE trans. on Computers, 1989, Vol.C-21, No. 6, pp. 546-556

- [7] Kam Hong Shum, "Fault Tolerance for Parallel Applications through Replication", In ICICS'97, September 1997, pp1462-1465.
- [8] Kai Li, Jeffrey F. Naughton, James S. Plank, "Low-Latency, Concurrent Checkpointing for Parallel Programs", IEEE Transaction on Parallel and Distributed System, August 1994, vol. 5, no. 8, pp874-879.
- [9] Hagbae Kim, Kang G. Shin, "Sequencing Tasks to Minimize the Effects of Near-Coincident Faults in TMR Controller Computers", IEEE Transaction on Computers, November 1996, vol. 45, no. 11, pp1331-1337.
- [10] Shu-Yi Yu and Edward J. McCluskey, "On-line Testing and Recovery in TMR Systems for Real-Time Applications", ITC International Test Conference, 2001, pp 240-248