결함허용 실시간 시스템을 위한 이중화 기법과 체크포인팅 기법의 성능 비교

임 성 화[†] · 김 재 훈[††] · 김 성 수[††]

요 약

결함 허용(fault olerant) 방법에는 두 개의 시스템으로 같은 작업을 수행하게 하는 이중계(duplex) 시스템과, 체크포인트를 두어 결함 발생 시 rollback 하는 checkpoint & rollback 시스템이 있다. 기존 결함허용 시스템에서는 요구되는 신뢰성을 유지하며 태스크의 수행시간을 단축시키는데 주안점을 두었지만 시간제약을 갖는 실시간 응용 분야에서는 신뢰성 유지와 정해진 시간 이내에 태스크를 종료시키는 것이 더욱 중요하다. 본 논문에서는 이들 결함허용 시스템을 실시간 응용 측면에서 비교 분석하였다.

Performance Comparisons of Duplex Scheme and Checkpointing Scheme for Fault-Tolerant Real-Time Systems

Sung-Hwa Lim[†] · Sai-Hoo Kim[††] · Sung-Soo Kim[††]

ABSTRACT

Two schemes are widely used for fault-tolerant systems; one is the duplex system that has a physical redundancy, and the other one is the checkpointing scheme that rolls back to the last checkpoint at a failure. The average execution time and availability are important factors for measuring the performance of the fault-tolerant systems. However, in fault-tolerant real-time systems with a time constraint, meeting the time constraint instead of reducing the average execution time is the most important factor in the performance evaluation. We analyze and compare the performance of two fault-tolerant schemes (the duplex system and the checkpointing scheme) for real-time applications.

1. 서 론

결함허용 시스템이란 주어진 시스템에 일부 결함이 발생되어도 이것이 동작중지로 이어지지 않고, 계속적인 정상동작이 가능한 시스템을 말한다. 결함 허용성을 실현하기 위하여 구성요소의 중복(physical redundancy), 재수행(time redundancy) 등의 방법이 있다. 프로그램을 수행 시 결함 허용 기능을 갖기 위하여 두 가지 방법을 다음과 같이 적용할 수 있다.

1.1 이중계(duplex)시스템 (physical redundancy)

프로그램을 서로 다른 머신에서 동시에 수행하도록 하여 한 머신에 오류가 발생하여도 나머지 머신에서 프로그램이 계속 수행 할 수 있도록 하거나(예, mirrored process), 프로그램 수행 시 필요한 데이터를 적어도 두 머신 이상에 복제하여 한 머신에 오류가 발생하여도 다른 머신에 보존된 데이터를 이용하여 프로그

※ 본 연구는 아주대학교 1998년도 정착 연구비 지원으로 진행되었음
† 준 회 원 : 아주대학교 대학원 컴퓨터공학과
†† 정 회 원 : 아주대학교 정보및컴퓨터공학부 교수
논문접수 : 1999년 4월 1일, 심사완료 : 1999년 8월 20일

램이 계속 수행될 수 있도록 한다(예, mirrored disk).

1.2 체크포인트와 롤백 리커버리 스킴 (time redundancy)

프로그램 수행도중 일정 간격으로 프로세스 상태를 디스크에 저장하여, 시스템 오류 발생 시 가장 최근의 체크포인트(checkpoint) 상태로 되돌아 가서(rollback) 다시 수행한다.

이 외에도 세 개의 시스템으로 같은 작업을 수행시킨 후 결과 값을 내어서, 투표(vote)를 통하여 과반수의 시스템에서 산출된 값을 선택하는 삼중 모듈 중복(Triple Modular Redundancy)이 있다[2]. 오류 발생률을 변화시키며 각각의 수행시간을 측정해보면, 오류 발생률에 따라 서로 다른 결과를 보인다. 이러한 성질 때문에 상황에 따라 적절한 스킴을 사용해야 하는데, 그 동안 이에 대한 연구는 많이 이루어져 왔다[3, 5, 7]. 그러나 마감시간 준수와 예측성을 중요시하는 실시간 시스템에서는, 평균 수행시간을 줄이는 것만이 충분조건이라고 할 수는 없다. 그러므로 실시간 시스템에서의 결함허용을 위한 새로운 연구가 필요하였고, 이를 위하여 각 스킴의 성능을 실시간 시스템의 시간제약 준수율 관점에서 분석, 비교하였다.

2 관련된 연구

실시간 개념에서의 결함허용 시스템에 관해 그 동안 진행되어 온 연구들은 다음과 같다.

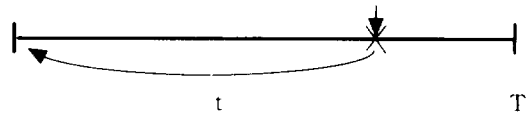
- **결함 허용 실시간 시스템에서의 중복레벨(redundancy level)결정**[6]
  많은 실시간 시스템은 성능 요구와 신뢰성 요구를 갖는다. 성능은 보통 한번에 실행 할 수 있는 태스크의 수나 양으로 측정한다. 신뢰성은 하드웨어나 소프트웨어의 결함모델에 의해 검증된다. 여러 상황에서 태스크에 대한 성능이나 신뢰성이 상충되는 경우가 많다. 그러므로, 실시간 결함허용 시스템의 성능 검증을 위해서는 성능-신뢰성의 상충에 대한 수학적 평가가 필요하다. 만약 신뢰성 향상을 위하여 태스크를 수행시킬 때 똑같은 카피를 두는 방법을 사용한다면, 그 카피의 정도에 대한 결정이 필요하고 이에 대한 방법을 연구하였다.

- **실시간 시스템을 위한 적응성 있는 결함허용 기법**[1]
  이 연구는 실시간 상황에서의 적응성 있는 결함 허용에 관한 토대를 제공하였다. 과거의 연구를 다음과 같이 두 가지로 확장하였다. 첫째는 실시간 시스템의 제약들을 포함하는 방향, 둘째는 응용 소프트웨어 모듈에서의 중복 관리를 위한 유연하고 적응성 있는 제어방법이다. 어플리케이션 디자이너들은 각각의 응용 모듈에 대해서 가능한 자원과 마감시간, 관측된 오류 등을 고려한 적응성 있는 방책을 갖는 결함허용 방법을 설정 할 수 있게 하였다.

- **분산 결함 허용 실시간 시스템 : Mars approach**[4]
  Mars 프로젝트는 1984년 유지 가능한 분산 실시간 결함 허용 시스템을 개발하기 위해 시작했다. 이것의 주요 관점은 실시간 자료의 제한된 시간 허용성, 최대 사용 점에서의 예측 가능한 성능, 결함 허용성, 그리고 유지 용이성 및 확장성 등을 고려하였다.

3. 각 스킴에 대한 비교

- **단일 시스템** (non fault-tolerant system)
  결함허용을 고려하지 않은 시스템이며, 하나의 시스템으로 태스크를 실행하다가 오류가 발생하면 처음부터 다시 실행한다.

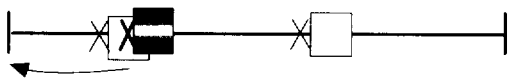


(그림 1) 단일 시스템

(그림 1)은 단일 시스템에서 오류발생시 처리과정을 나타낸다. 작업의 전체 수행시간이 T이고 오류 발생시간을 t 라고 한다면, 작업의 실제 수행시간의 기대값(Γ)을 구하면, $\Gamma = T + t$ 이다. 위의 경우는 오류가 단 한번 일어난 경우이고, 실제 수행시간은 오류 발생 없이 한번에 끝까지 수행될 때까지 누적된다.

- **이중계(duplex) 시스템** (single fault-tolerant system)
  두 개의 시스템에서 같은 작업이 수행되며 만약 한 시스템에 오류가 발생하면 나머지 시스템으로 작업을 계속 수행한다. (그림 2)에서와 같이 X 지점에서 오류가 발생하면, 남은 시스템으로 작업을 계속하게

그 동안 오류가 발생한 시스템은 복구를 하게 된다 (①의 경우). 만약 첫 번째 오류가 발생한 시스템이 복구하기 전에 두 번째 시스템마저 오류가 발생하면, 시작점으로 되돌아가 작업을 다시 시작하게 된다.



① : 한 시스템의 오류 발생 및 복구

② : 한 시스템의 오류 발생 및 복구 이전에 다른 시스템의 오류 발생 및 복구

(그림 2) 이중계 시스템

전체 수행시간이 T이고, 복구 시간이 R, 중복 오버헤드(redundancy overhead)를 α라하고 오류가 한번 발생했을 때 수행시간의 기대값(Γ)을 구하면 $\Gamma = T*\alpha+R$ 이다. 그러나 만약 오류가 R 보다 작은 간격으로 발생한다면(즉 오류가 발생한 시스템이 복구되기 이전에 다른 시스템에 오류가 또 발생하면), 처음으로 되돌아가 다시 수행해야 한다. 오류 발생률을 λ라 할 때 이중계 시스템의 오버헤드(γ)는 다음과 같다[3].

$$\gamma = \frac{A}{B}\, e^{BaT} - \frac{A}{B}$$

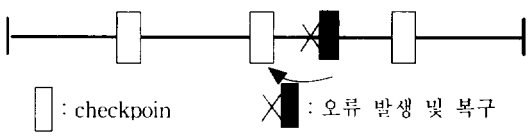
$$E(R) = \int_0^R t \frac{\lambda e^{-\lambda t}}{1-e^{-\lambda R}}\, dt = \lambda^{-1} - \frac{R e^{-\lambda R}}{1-e^{-\lambda R}},$$

$$A = 1 + \lambda e^{-\lambda R} R + \lambda (1-e^{-\lambda R}) E(R),$$

$B = \lambda(1-e^{-\lambda R})$ 일때,

- **Checkpoint and rollback system**

한 개의 시스템으로 작업을 하되, 수행도중 일정시간마다 체크포인트를 두고 현재의 상태를 안전한 저장장치(disk)에 저장한다. 오류가 발생하면 가장 최근에 저장된 체크포인트로 귀환(roll back) 하여 그 시점부터 다시 시작한다. (그림 3)과 같이 X 지점에서 오류가 발생하면 가장 최근에 저장된 체크포인트로 되돌아가서, 그 지점에서부터 다시 수행하게 된다.



(그림 3) 체크포인트를 사용하는 시스템

(그림 3)은 체크포인트를 사용하는 시스템에서 오류가 발생할 때의 처리과정을 나타낸다. 전체 수행 시간을 T, 귀환(roll back) 시간을 R, 체크포인팅 시간을 C, 체크포인트 간격을 T_c라 할 때, 최적의 체크포인트 간격(T_{opt})은 다음과 같다 (λ: 오류 발생률).

$$T_{opt} = \sqrt{\frac{2C}{\lambda}} \quad [5,7].$$

체크포인트를 사용한 시스템의 한 체크포인트 간격에서 수행시간의 기대값은 다음과 같다.

$$\Gamma = \frac{1}{\lambda} \times e^{\lambda R} \times (e^{\lambda(T_c + C)} - 1) \quad [5,7]$$

4 성능 비교 및 분석

4.1 비실시간 시스템에서의 성능비교

실시간 시스템에서의 성능비교에 앞서 우선 두 가지 결함 허용 시스템(duplex 와 checkpointing)과 결함을 허용하지 않는 일반 단일(single) 시스템을 사용했을 때 태스크의 평균 수행시간을 시뮬레이션을 통하여 비교하였다. 오류 발생률(λ)은 poisson분포라 가정하였고, 시간 [0, t]에서 작업이 오류 없이 수행될 확률은 $e^{-\lambda t}$ 이며, 오류가 발생될 때까지의 평균시간 (MTTF : Mean Time To Failure)은 $1/\lambda$이다. 시뮬레이션은 100,000번 반복수행 하였다. 각 스킴에서의 파라메터 와 그 값들을 다음과 같이 정의하였다.

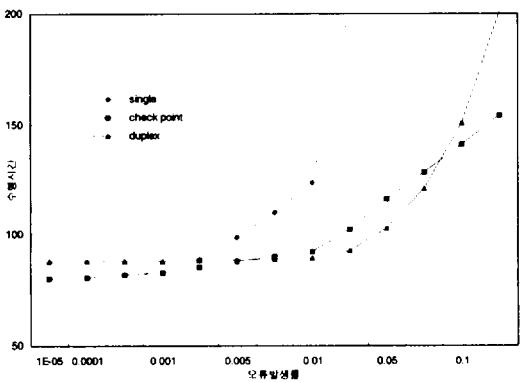
- 단일 시스템 (non fault-tolerant system)
  - T = 80 (태스크의 유효 수행시간 : 오류 없을 때 태스크의 순수 수행시간)
- 이중계 시스템 (duplex system)
  - T = 80 (태스크의 유효 수행시간)
  - R = 1 (recovery time)
  - α = 0.1 (redundancy overhead)
- checkpoint and rollback system
  - T = 80 (태스크의 유효 수행시간)
  - R = 1 (rolling back time)
  - C = 1 (checkpoint time)
  - $T_c = T_{opt} = \sqrt{\frac{2C}{\lambda}}$ (checkpoint interval time)

체크포인트 간격은 시뮬레이션을 통하여 구했는데, T의 약수중 T_{opt} 와 근접한 값 중의 하나가 된다.

<표 1>은 오류 발생률(λ)을 0.00001~0.1 까지 변화시키면서 시뮬레이션을 100,000회 반복 수행하여 평균값을 구한 결과이다 - 수학적으로도 구할 수 있다. (그림 4)에서 이 결과를 그래프로 나타내었다. 상대적으로 두 결함 허용 시스템이 단일 시스템 보다 우수한 성능을 보였다. 그러나 λ값이 0.0025 이하에서는 이중계 시스템에서 중복 오버헤드(redundancy overhead - α) 때문에 오히려 단일 시스템에서 보다 더 많은 시간이 소요됨을 알 수 있다.

〈표 1〉 오류발생률(λ)의 변화에 따른 실제수행시간

오류 발생률 / 시스템	0.00001	0.0001	0.001	0.01	0.1
single	80.0351	80.3265	83.3131	123.9819	32790
check point	80.0351	80.3265	82.7284	92.385	140.9972
duplex	88.0009	88.0089	88.0889	89.2846	150.6991



(그림 4) 오류발생률 (λ) 의 변화에 따른 실제 수행시간의 변화 그래프

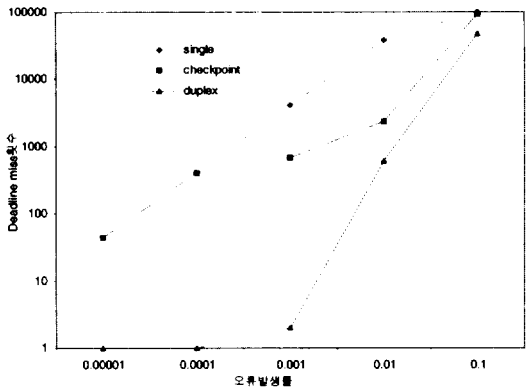
4.2 실시간 시스템에서의 성능비교

4.1절 에서의 결과는 시간제약을 고려하지 않고 태스크의 평균 수행시간만을 고려하여 두 가지 결함 허용 시스템과 비 결함 허용 시스템의 성능을 비교하였다. 그러나 시간제약을 갖는 실시간 시스템에서는 마감시간의 준수가 성능비교의 주요 척도가 되기 때문에 이의 관점에서 성능측정을 하였다. 마감시간의 준수를 성능비교의 척도로 삼아 4.1절과 같은 시스템 변수를 사용하여 시뮬레이션 한 결과가 <표 2>에 나와 있고 (그림 5)에서 그래프로 나타내었다. 이때 마감시간은 태스크 유효 수행시간의 1.5배라고 가정하였다 - 다음 실험에서 마감시간의 변화를 주었다.

오류발생률이 0.00001과 0.0001에서 체크포인트 시스템과 단일 시스템에서의 마감시간 초과 횟수가 같은 이유는 체크포인트 시스템의 최적 체크포인트 간격 (T_{opt}) 이 전체수행시간 (T) 보다 커서, 전체 수행동안 체크포인트가 없었기 때문이다.

〈표 2〉 오류발생률(λ) 변화에 따른 deadline miss 횟수

오류 발생률 / 시스템	0.00001	0.0001	0.001	0.01	0.1
single	44	406	3998	37920	99842
checkpoint	44	406	681	2352	92316
duplex	0	0	2	603	47863



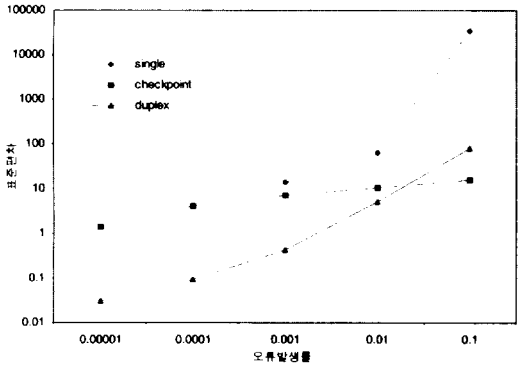
(그림 5) 오류발생률(λ)변화에 따른 각 스킴의 deadline miss 횟수

결과에서 알 수 있듯이, λ 값이 0.001이하 일 때 이중계 시스템의 평균 수행시간이 타 스킴보다 높지만 (그림 4), 마감시간을 어긴 횟수는 현저히 적다(그림 5). 단일 시스템과 체크포인트 시스템(특히 단일 시스템)의 평균 수행시간은 작으나 마감시간을 어긴 횟수가 더 많다는 것은, 매 실행마다의 결과의 편차가 심해서 평균값은 작더라도 마감시간을 넘는 결과가 더 많이 발생하기 때문이다. 실제로 λ값이 0.001인 상황에서 수행시간의 표준편차를 구해보면<표 3>,(그림 6), 이중계(duplex) 시스템이 0.4405로 가장 적음을 알 수 있다. 결과에서 알 수 있듯이 수행시간의 크기보다는 표준편차의 크기(분포)가 마감시간을 어기는 횟수에 더 큰 영향을 주었다. 태스크의 수행도중 오류가 발생하지 않으면 태스크가 빠른 시간이내에 종료되지만, 오류가 발생하면 처음부터(단일시스템의 경우) 또는 가장 최

근의 체크포인트시점부터(checkpoint 시스템의 경우) 재수행 해야 하기 때문에 마감시간을 어길 가능성이 이중계 시스템 보다 높다. 반면 이중계 시스템은 중복 구성으로 인하여, 오류가 발생하지 않는 상황에서는 수행시간은 길지만 오류 발생시 재수행 시간이 없기 때문에 예측 가능한 수행시간을 기대 할 수 있기 때문에 마감시간을 어기는 횟수도 적다. 이중계 시스템은 ① 이중화의 오버헤드가 작을 때, ② 오류 발생률이 그다지 크지 않기 때문에 두 노드 이상에서 오류가 발생되어 처음부터 다시 시작할 가능성이 희박할 때, 약간의 오버헤드로 예측 가능한 수행시간을 기대할 수 있다. 그러나 이중계 시스템은 오류발생률이 거의 없을 경우에도 오류발생률과 무관한 오버헤드(fault-free overhead)로 인하여 성능이 저하될 수 있으며, 특히 마감시간이 촉박할 경우 마감시간 준수에 치명적인 영향을 줄 수 있는 반면, 체크포인트 기법은 오류발생률이 거의 없을 경우 체크포인팅 최적 주기가 길어지므로 오버헤드(failure-free overhead)를 최소화시킬 수 있다.

〈표 3〉 오류발생률(λ)에 따른 각 스킴의 표준편차

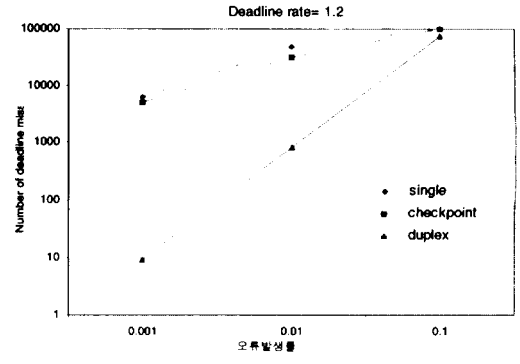
시스템 \ 오류발생률	0.00001	0.0001	0.001	0.01	0.1
single	1.4143	4.1679	13.7129	64.3970	33466
check point	1.4143	4.1679	7.0496	10.6158	15.9121
duplex	0.0300	0.0940	0.4405	5.0149	79.2562

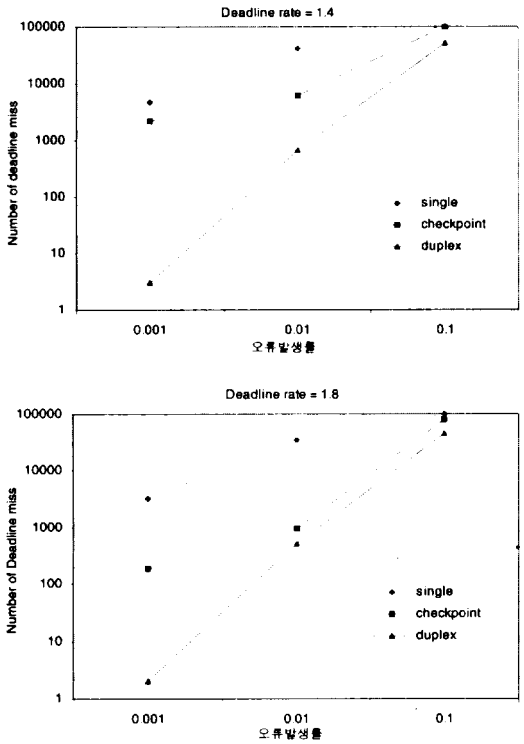


(그림 6) 오류발생률(λ)에 따른 각 스킴의 표준편차

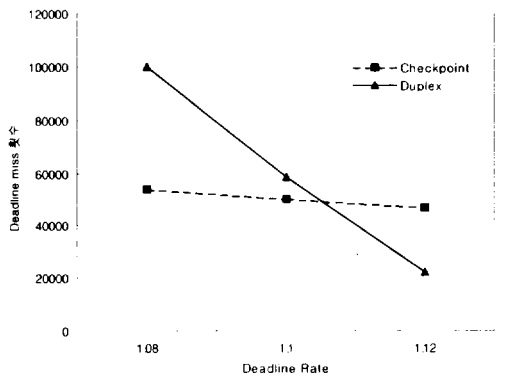
다음은 여러 가지 다른 마감시간에서 오류 발생률이 변화할 때의 마감시간 초과 횟수를 비교해 보았다. 마감시간의 비율 (deadline rate : deadline/오류가 없을 때의 수행시간)이 1.2, 1.4, 1.8 일 때, 오류발생률(λ)을 0.001, 0.01, 0.1 로 변화시켜 가며 마감시간을 어긴 횟수를 측정했다. 마감시간이 촉박한 경우(1.2배의 경우) 체크포인트 시스템의 마감시간 초과횟수는 단일 시스템의 그것과 유사하고, 마감시간이 느슨해지면(1.8) 단일시스템보다 적어지며 점차 이중계 시스템을 따라감을 알 수 있다. 오류발생률이 아주 작고(0.001) 마감시간이 클 경우(1.8), 단일 시스템과 결함허용 시소템과의 차이가 크지 않다. 그러나 오류발생률이 크거나, 마감시간이 작으면 차이가 많은 것을 알 수 있다.

일반 시스템(non fault-tolerant)에서 이중계 시스템은 수행시간에 있어서 타 스킴의 시스템보다 일반적으로 적은 수행시간이 소비되었지만, 중복에 의한 오버헤드 때문에 오류발생률이 일정수준 이하에서는 오히려 더 큰 수행시간을 소비하였다. 하지만 마감시간을 중요시 여기는 실시간 시스템에서는 이중계 시스템의 수행시간이 더 큰 지점((그림 1)에서 λ = 0.001 이하) 에서도, 마감시간을 어기는 횟수가 타 시스템보다 월등히 낮게 나타났다. 이것은 이중 시스템의 수행시간의 분포가 타 스킴의 분포보다 작기 때문이다[그림 6]. 즉, 약간의 이중화의 오버헤드를 부담해야 하지만 대부분의 결함(단일노드 결함)으로부터 영향을 받지 않기 때문에 수행시간이 대부분의 경우 예측 가능하다. 만일, 이중화 오버헤드가 상대적으로 크거나 결함 발생률이 높아 두 노드 이상에서 동시에 결함이 발생될 가능성이 크다면 이중계 시스템 보다는 체크포인팅이 효과적인 방법이 될 수 있다. (그림 8)은 오류발생률이 0.01일 때, 이중계 스킴과 체크포인트 스킴에 대해 마감시간의 임박정도에 따라 마감시간을 어기는 횟수를 비교한 그래프 이다. 그림에서 볼 수 있듯이 마감시간이 촉박한 경우(deadline rate이 1.15 이하)에는 이중화 오버헤드 때문에 이중계 스킴의 성능이 오히려 떨어진다.





(그림 7) deadline rate = 1.2, 1.4, 1.8 일 때 오류 발생률의 변화에 따른 deadline miss 횟수



(그림 8) λ = 0.01일 때 마감시간에 따른 두 기법의 deadline miss 횟수

5 결 론

시뮬레이션 결과에 의하면 비 실시간 시스템에서 이중계(duplex) 시스템이 대체적으로 우수함을 알 수 있었다. 그렇지만 이중계 시스템은 물리적 중복(physical redundancy)으로 인한 오버헤드로, 오류발생률이 일정 수준 이하이면 체크포인트 시스템이나 결함허용을 하지 않는 시스템보다 오히려 평균수행시간이 클 수 있p었다. 그러나 실시간 시스템에서는 단순한 평균 수행시간 보다는 수행시간의 분포(표준편차)가 성능에 더 큰 영향을 끼침을 알 수 있었다. 즉 수행시간의 편차가 적어 수행시간을 예측할 수 있는 시스템이 실시간 시스템에 적합하다는 일반적 사실이 결함 허용 시스템에도 동일하게 적용되며, 실시간 결함시스템의 성능을 결정하는 데는 평균 수행시간 보다는 실시간 시스템의 성능평가 기준인 마감시간 준수율이나 예측성 등에 영향을 주는 수행시간의 편차가 중요한 요소가 됨을 알 수 있다.

참 고 문 헌

[1] A. Bondavalli, J. Stankovic, L. Strigini, "Adaptable Fault-Tolerance for Real-Time Systems," CNUCE-CNR, Pisa, Italy, 1994.

[2] Barry W. Johnson, *Design and Analysis of Fault-Tolerant Digital Systems*, Addison-Wesley Publishing Company, Reading, Mass., pp.52, 1989.

[3] Jai-Hoon Kim and Nitin H. Vaidya, "Analysis of one-level and two-level failure recovery schemes for distributed shared memory systems," IEE Proceedings-Computers and Digital Techniques, vol. 146, Issue 3, pp.125-130, May 1999.

[4] H. Kopetz, A. Damn, C. Koza, M. Mulazzani, W. Schwabl, C. Senift, R. Zainlinger, "Distriduted Fault-Tolerant Real-Time Systems : The Mars Approach," IEEE Micro, pp.25, Feb, 1989.

[5] N. H. Vaidya, "On checkpoint latency," in *Proc. of the 1995 pacific Rim International Symposium on Fault-Tolerant Systems*, pp.60-65, Dec. 1995.

[6] F. Wang, K. Ramamritham, J. Stankovic, "Determining Redundancy Levels for Fault Tolerant Real-Time Systems." Dept. of Computer Science, University of Massachusetts, 1995.

[7] J. Young, "A first order approximation to the optimal checkpoint interval," Communication of the ACM, Vol.17, pp.530-531, Sept, 1974.

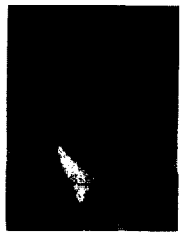
임 성 화

e-mail : holyfire@madang.ajou.ac.kr

1999년 아주대학교 정보 및 컴퓨터 공학부(학사)

1999년 현재 아주대학교 컴퓨터공학과 석사과정

관심분야 : 실시간 시스템, 결함허용 시스템, 이동컴퓨팅 등



김 재 훈

e-mail : jaikim@madang.ajou.ac.kr

1984년 서울대학교 제어계측공학과(학사)

1993년 Indiana University, Computer Science(석사)

1997년 Texas A&M University, Computer Science(공학박사)

1984년~1991년 대우통신㈜ 컴퓨터연구실 대리

1995년~1997년 Texas A&M University, Graduate Research Assistant

1997년~1998년 삼성전자㈜ 컴퓨터시스템팀 수석연구원

1998년~현재 아주대학교 정보및컴퓨터공학부 조교수

관심분야 : 분산시스템, 실시간시스템

김 성 수

e-mail : sskim@madang.ajou.ac.kr

1982년 서강대학교 전자공학과 (공학사)

1984년 서강대학교 전자공학과 (공학석사)

1995년 Texas A&M University, 전산학과(공학박사)

1983년~1986년 삼성전자(주) 종합연구소 컴퓨터연구실(주임연구원)

1986년~1996년 삼성종합기술원 수석연구원

1991년~1992년 Texas Transportation Institute 연구원

1993년~1995년 Texas A&M University, 전산학과, T.A.

1997년~1998년 한국정보처리학회, 한국정보과학회 논문지 편집위원

1996년~현재 아주대학교 정보통신대학 정보및컴퓨터공학부 교수

관심분야 : 멀티미디어, 결함 허용, 이동 컴퓨팅, 성능평가, 시뮬레이션