동기적 검사점 기법에서 불필요한 복귀를 회피하기 위한 쓰레기 처리 기법

(Lazy Garbage Collection of Coordinated Checkpointing Protocol for Avoiding Sympathetic Rollback)

정 광 식 † 유 헌 창 †† 이 원 규 ††† 이 성 훈 †††† 황 종 선 †††††

(Kwang Sik Chung)(Heon-Chang Yu)(Won-Gyu Lee)(Seong Hoon Lee)(Chong-Sun Hwang)

요 약 이 논문은 동기적 검사점 기법에서 결함 포용을 목적으로 불안전 저장 장치(volatile storage)에 저장되는 메시지 로그와 안전 저장 장치에 저장되는 검사점의 쓰레기 처리 기법을 제안한다. 기존의 동기적 검사점 기법을 기반으로 한 결함 포용 정보 쓰레기 처리 기법은 가장 최근의 검사점을 제외한 모든 결함 정보를 쓰레기 처리하였다. 하지만 TCP/IP와 같은 신뢰적 통신 기법을 기반으로 한 동기적 검사점 기법이 가장 최근의 검사점만을 복귀 회복 기법에서 사용한다면, 손실 메시지(lost message)로 인한 불필요한 복귀(sympathetic rollback)가 발생된다.

이 논문은 동기적 검사점 기법에서 손실 메시지로 인한 불필요한 복귀 문제를 해결하기 위해 각 프로세스가 동기화된 가장 최근의 검사점외에 검사점이나 메시지 로그를 유지해야 한다는 것을 보였다. 또한 손실 메시지로 인한 불필요한 복귀 문제의 해결을 위해 관리되어야 하는 검사점이나 메시지 로그가 쓰레기 처리되어지기 위해 필요한 조건을 새롭게 정의하며, 이 정의를 기반으로 한 검사점과 메시지 로그의 쓰레기 처리 알고리즘을 제안한다. 제시된 조건을 기반으로 한 검사점과 메시지 로그의 쓰레기 처리는 송수신 메시지에 부가된 손실 메시지 관련 프로세스 정보를 이용하므로 쓰레기 처리를 위한 부가적인 메시지를 발생시키지 않는다. 제안된 기법은 손실 메시지 관련 정보가 부가된 메시지가 송수신되기 전까지 쓰레기 처리가 지연되는 '지연 쓰레기 처리 현상(lazy garbage collection)'을 발생시킨다. 하지만 '지연 쓰레기 처리 현상'은 분산 시스템의 일관성을 위배하지 않는다.

키워드 : 동기화 검사점 기법, 메시지 로그, 쓰레기 처리, 불필요한 복귀

Abstract This paper presents a garbage collection protocol for checkpoints and message logs which are saved on the stable storage or volatile storage for fault tolerancy. The previous works of garbage collections in coordinated checkpointing protocol delete all the checkpoints except for the last checkpoints on each processes. But implemented on top of reliable communication protocol like as TCP/IP, rollback recovery protocol based on only last checkpoints makes sympathetic rollback.

We show that the old checkpoints or message logs except for the last checkpoints have to be preserved in order to replay the lost messages. And we define the conditions for garbage collection of checkpoints and message logs for lost messages and present the garbage collection algorithm for checkpoints and message logs in coordinated checkpointing protocol. Since the proposed algorithm uses process information for lost message piggybacked with messages, the additional messages for garbage collection is not required The proposed garbage collection algorithm makes 'the lazy garbage collectioneffect', because relying on the piggybacked checkpoint information in send/receive message.

· 본 연구는 한국과학재단 목적기초연구(R01-2001-00354)지원으로 수행되었음.

† 비 회 원 : 런던대학교 네트웍연구소 연구원
k.chung@cs.ucl.ac.uk
†† 정 회 원 : 고려대학교 컴퓨터교육과 교수
yuhc@comedu.korea.ac.kr
††† 종신회원 : 고려대학교 컴퓨터교육과 교수
lee@comedu.korea.ac.kr
†††† 비 회 원 : 천안대학교 정보통신학부 교수
shlee@mail.chonan.ac.kr
††††† 종신회원 : 고려대학교 컴퓨터학과 교수
hwang@disys.korea.ac.kr

논문접수 : 2001년 7월 12일
심사완료 : 2002년 3월 7일

But 'the lazy garbage collection effect' does not break the consistency of the whole systems.
Key words : coordinated checkpointing protocol, message log, garbage collection, sympathetic rollback

1. 서 론

분산 컴퓨팅 시스템의 발달로 인해 하나의 작업이 여러 프로세스에서 분산되어 수행되고 있다. 분산 작업의 수행은 결함 발생 확률을 높이며, 이에 따라 결함 발생을 포용해 줄 수 있는 여러 가지 결함 포용 기법이 연구되고 있다. 결함 포용을 위한 대표적인 기법으로 검사점 기법과 메시지 로깅 기법이 있으며, 결함 포용 기법들은 결함 포용을 위해 검사점과 메시지 로깅 정보를 저장한다. 이와 같은 결함 포용 정보의 유지는 어느 시점을 지나면 메모리의 사용량을 증가시키게 되고, 저장된 상태 정보 중 불필요한 정보의 처리, 즉 쓰레기 처리(garbage collection)를 필요로 한다. 메시지 로깅 기법은 비관적(pessimistic), 낙관적(optimistic), 인과적(causal) 메시지 로깅 기법으로 나뉘고, 결함 포용을 위한 결함 포용 정보인 메시지 로그를 쓰레기 처리 대상으로 하였다. 검사점 기법의 경우 동기적 검사점 기법과 비동기적 검사점 기법으로 나누고, 결함 포용을 위해 프로세스의 상태 정보를 저장하며, 쓰레기 처리의 대상은 검사점이다[1, 2, 3].

검사점 기법의 경우, 결함 포용 정보의 쓰레기 처리를 위해 일관된 회복선을 찾아내고, 회복선 이전의 모든 회복 정보를 삭제하는 방법을 사용한다[3, 4]. 특히 동기적 검사점 기법은 결함이 발생하더라도 관련된 각각의 프로세스가 가장 최근의 검사점까지만 복귀하므로 이전의 모든 검사점은 삭제될 수 있었다[5]. 즉, 기존의 동기적 검사점 기법에서는 일관된 전역 검사점 정의만을 고려한 쓰레기 처리 기법이 고려되었다. 하지만 신뢰성을 갖는 통신 기법(TCP/IP 프로토콜)을 기반으로 한 동기적 검사점 기법에서, 결함 발생은 손실 메시지를 발생시키며, 손실 메시지는 불필요한 복귀(sympathetic rollback)를 발생시킨다. 따라서 복귀 회복 기법은 손실 메시지에 대한 처리를 고려해야 하며, 일관된 전역 검사점과 손실 메시지에 관련된 결함 정보가 함께 고려된 결함 포용 정보 쓰레기 처리 기법이 필요하다.

이 논문에서는 신뢰성을 갖는 통신 기법을 기반으로 한 회복 기법에서 손실 메시지의 처리를 위한 결함 포용 정보의 쓰레기 처리 조건을 정의하고, 쓰레기 처리 기법을 위해 필요한 손실 메시지 관련 프로세스 정보를 정의한다. 이를 기반으로 손실 메시지 처리를 위한 결함 포용 정보 쓰레기 처리 기법을 제안한다. 이 논문의 2장에서는 시스템 모델과 기존의 동기적 검사점 기법이 가지는 불필요한 복귀 문제를 해결하기 위한 쓰레기 처리 기법에 대해 설명한다. 3장에서는 손실 메시지의 처리를 위한 결함 정보의 쓰레기 처리 조건을 제안한다. 3장에서는 결함 포용 정보의 쓰레기 처리 기법을 위한 손실 메시지 관련 프로세스 정보를 정의하며, 이를 기반으로 결함 포용 정보 쓰레기 처리 알고리즘을 제안한다. 4장에서는 이 논문에서 제안되는 결함 포용 정보 쓰레기 처리 알고리즘의 정당성을 증명한다. 5장에서는 연구의 결론과 향후 연구 과제에 대해 언급한다.

2. 시스템 모델 및 연구 동기

2.1. 시스템 모델

전체 분산 시스템 N은 n개의 프로세스로 구성된다. N의 수행 단위를 ρ라 하며, ρ동안 전달된 메시지는 m으로 정의한다. 메시지 m의 결함 포용 정보는 송신자 프로세스의 식별자 $m.source$, 송신자 프로세스에 의해 메시지에 부여된 메시지 송신 식별자 $m.ssn$, 수신자 프로세스에 의해 메시지에 부여된 메시지 수신 식별자 $m.rsn$을 가진다. $deliver_{m.dest}(m)$는 메시지 수신 프로세스에 의해 메시지의 수신 사건이 프로세스의 상태에 반영되어 분산 시스템의 전체 상태가 전이하였음을 나타낸다[1].

메시지 전달을 기반으로 하는 분산 시스템의 전역 상태(global state)는 모든 관련 프로세스와 통신 채널의 상태 집합이다[6]. 직관적으로 일관된 전역 상태는 분산 작업의 옳은 정상 작업 수행동안 발생될 수 있다. 즉, 일관된 시스템 상태(consistent system state)는 하나의 프로세스 상태가 메시지의 수신을 반영하였다면, 상대 송신 프로세스의 상태가 그 메시지의 송신 사건을 반영한 상태이다. 동기적 검사점 기법에서 일관된 검사점 집합은 [정의 1]과 같이 정의한다[7].

[정의 1] 일관된 전역 검사점 : 각 프로세스로부터의 N개의 검사점의 집합을 $G_Ckpt = \{ C_1, C_2, \ldots, C_N \}$이라 하면, 임의의 메시지 M과 $1 \le i \le n$인 임의의 i에 대해 $\exists m,\ receive(M) \in G_Ckpt \Rightarrow send(M) \in C_i$이 성립할 경우, 전체 시스템의 상태 집합 G_Ckpt는 일관된 전역 검사점 집합이라 정의한다. ■

$Depend(m)$은 m이 수신 프로세스에 반영된 후 발생한 송신 메시지 m'이 반영된 프로세스들과 메시지 m이 반

영된 수신 프로세스의 합집합을 의미하며, *De pend*(m)에 포함되는 프로세스는 메시지 m에 의존한다[8].

$$Depend(m) \stackrel{\text{def}}{=} \left\{ j \in N \mid \begin{array}{l} ((j = m.dest) \\ \quad \wedge j \text{ has delivered } m) \\ \vee (\exists m' : (\text{deliver}_{m.dest}(m) \\ \quad \rightarrow \text{deliver}_j(m'))) \end{array} \right\}$$

복귀 회복 기법에서 결함 포용 정보 관리를 위해 필요한 메시지 m에 관련된 결함 포용 정보 $FTI(m)$는 [정의 2]와 같다.

[정의 2] $FTI(m)$는 메시지 m에 관련된 결함 포용 정보로서, 다음과 같이 정의한다.

i) $FTI(m) \stackrel{\text{def}}{=} \{content_m, m.dest, m.rsn, m.ssn\} \cup \{Depend(m)\}$

ii) $FTI(m)$는 검사점 혹은 메시지 로그에 저장된다 ■

이 논문에서 프로세스간의 관계는 두 사건 e_1, e_2를 Lamport의 전후 사건 관계(happen-before relation)를 이용하여 표현하고, $e_1 \rightarrow e_2$로 나타낸다[7]. 전후 사건 관계, $e_{P_i,1} \rightarrow e_{P_j,2}$는 프로세스 P_i의 사건 $e_{P_i,1}$와 프로세스 P_j의 사건 $e_{P_j,2}$에 대해 다음의 두 조건에 의해 정의된 이행 폐쇄 관계이다.

- $i=j$이면 사건 $e_{P_i,1}$는 사건 $e_{P_j,2}$이전에 수행되었다.
- $i \neq j$이면 $e_{P_i,1}$는 송신 사건이고 $e_{P_j,2}$는 그에 대한 수신 사건이다.

2.2. 연구 동기

그림 1에서 전역 검사점 상태의 관점에서 프로세스 P_1이 송신한 메시지 m_1을 프로세스 P_2는 아직 수신하지 않았다. 이러한 메시지를 미수신 메시지(in-transit message)라 한다. [정의 1]은 전체 시스템이 일관된 전역 검사점 집합을 가지는 경우, 결함이 발생하여도 일관된 상태의 복귀가 가능하다고 정의했다. 이것은 통신 채널의 상태와 미수신 메시지는 언제나 일관된 상태를 갖는다는 가정을 기반으로 하였기 때문이다. 미수신 메시지는 결함이 발생한다면, 손실 메시지(lost message)가 된다. 기존의 동기적 검사점 기법의 일관된 전역 검사점 집합에서 미수신 메시지는 전체 시스템의 전역 상태에서 제외하고, 통신 채널의 일부로 가정하였다. 따라서 미수신 메시지는 전역 검사점의 일관성을 위배하지 않는다. 그러나 시스템 모델에서 결함 포용 기법이 어떠한 통신 기법을 기반으로 하는가에 따라 복귀 회복 기법은 결함이 발생하였을 경우, 미수신 메시지의 전달(delivery)을 보장해 주어야 할 경우도 있다.

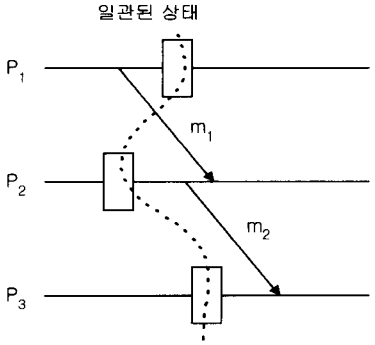


그림 1 손실 메시지

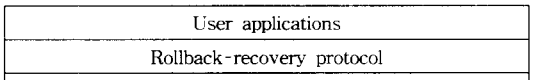


그림 2 신뢰적인 통신 기법을 기반으로 한 회복 기법

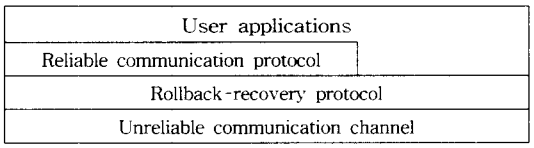


그림 3 비신뢰적인 통신 채널을 기반으로 한 회복 기법

그림 2와 그림 3은 통신 기법과 복귀 회복 기법의 구조를 나타낸다. 그림 2에서는 신뢰성을 가지는 통신 기법을 가정하였기 때문에, 복귀 회복 기법은 신뢰성을 가지는 통신 기법 위에서 구현된다. 반면 그림 3에서는 복귀 회복 기법이 신뢰성을 가지는 통신을 가정하지 않았다.

신뢰적인 통신 기법은 정상 수행동안의 메시지 전달을 보장한다. 하지만 신뢰적인 통신 기법은 프로세스의 결함이 있을 경우, 그들 스스로 메시지 전달의 신뢰성을 보장하지 못한다. 예를 들면, 수신 프로세스의 결함 때문에 미수신 메시지가 손실된다면, 통상적인 통신 기법은 타임아웃 기법을 통해 송신 메시지가 전달되지 못했다는 것을 송신 프로세스에게 알린다. 이때 복귀 회복 기법은 수신 프로세스가 다시 살아날 것을 보장해야 하며, 메시지 전달의 타임아웃을 처리해 주어야 한다. 그래서 복귀 회복 시스템은 결함이 발생한 수신 프로세스에게 메시지를 재전송해 주어야 한다. 만일 시스템이 그림 3과 같은 신뢰할 수 없는 통신 기법을 가정한다면, 복귀 회복 기법은 특별한 방법으로 손실 메시지를 처리해 주어야 할 필요는 없다. 그것은 프로세스의 결함으로

인해 손실된 손실 메시지와 신뢰할 수 없는 통신으로 인해 발생되는 손실 메시지와 구분될 수 없기 때문이다. 통신으로 인한 결함이나 프로세스로 인한 결함에 의해서 발생된 손실 메시지의 손실은 정상 수행에서 발생될 수 있는 사건이며, 옳은 시스템에서의 수행이다[7].

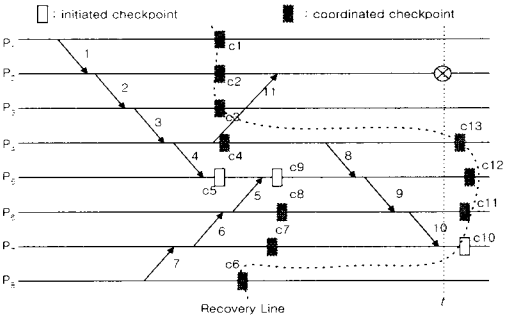


그림 4 전역 회복선

하지만 TCP/IP와 같은 신뢰성을 갖는 통신 기법을 기반으로 한 회복 기법은 손실 메시지를 고려한 쓰레기 처리 기법이 필요하다. 그림 4에서 전체 시스템의 각각의 프로세스들이 유지하고 있는 전역 검사점 집합{C1, C2, C3, C13, C12, C11, C10, C6}은 일관된 전역 검사점 집합이다. 따라서 프로세스 P_2에서 결함이 발생하더라도 전체 시스템은 그림 4에서의 회복선까지 복귀한다. 하지만 결함 포용 기법이 TCP/IP와 같은 신뢰성을 제공하는 통신을 기반으로 하는 경우, 메시지 11에 대한 처리가 불필요한 복귀 문제를 일으킬 수 있다. 즉, 결함이 발생한 프로세스 P_2는 검사점 C2까지 복귀할 것이고, C2로부터 재실행을 시작한다. 이때 메시지 11에 관련된 정보를 저장하지 못하고 있다면, P_2는 정상적인 작업 수행이 불가능해지며, P_4에게 메시지 재전송을 요구하게 된다. P_4는 메시지 11의 재전송을 위해 복귀를 해야 하며, 이러한 불필요한 복귀 문제점의 해결은 두가지 방법을 통해 해결될 수 있다. P_2가 메시지 11에 대한 메시지 로그를 관리하거나, 검사점 C4 이전의 검사점을 유지하는 것이다. 이 논문에서 제안하는 쓰레기 처리 기법은 동기화된 전역 검사점이 존재하더라도, 불필요한 복귀를 발생시킬 수 있는 손실 가능 메시지에 대한 쓰레기 처리를 보류시킨다. 즉, 신뢰적인 통신 기법을 가정한 동기적 검사점 기법의 회복 기법에서는 손실 메시지의 처리와 관련된 결함 포용 정보의 쓰레기 처리 기법을 제안한다.

3. 손실 메시지 관련 프로세스 정보 및 쓰레기 처리

3.1. 결함 포용 정보의 쓰레기 처리

그림 4에서 메시지 11은 결함이 발생되기 전까지 회복선의 시점에서 미수신 메시지이다. 그리고 프로세스 P_2의 시점 t에서 결함이 발생했을 때, 결함 포용 시스템은 유한 시간내에 결함 프로세스의 회복을 보장하므로, 메시지 11은 손실 메시지가 된다[4]. P_2의 회복 기법 수행중, 메시지 11과 같은 손실 메시지는 회복 프로세스를 위해 재전송되어야 한다. 따라서 송신 프로세스는 손실 메시지의 재전송을 위해 메시지 로그를 유지하거나 이전의 검사점을 유지해야 한다. 메시지 로그를 관리하는 기법에는 송신자 기반의 메시지 로깅 기법과 수신자 기반의 로깅 기법이 있다[4]. 가장 최근의 검사점 이후로 복귀하지 않으므로 송신자 기반의 메시지 로깅 기법은 메시지 로그를 불안전 저장 장치에 저장할 수 있다. 이 논문에서는 손실 메시지를 위한 결함 포용 정보를 송신자 기반의 메시지 로그를 통해 관리하는 경우에 대한 쓰레기 처리 조건을 다음과 같이 제안한다.

[조건 1] 결함 포용 정보의 쓰레기 처리 조건은 동기적 검사점과 신뢰적인 통신 기법이 보장되고, 프로세스 p_i에서 프로세스 p_j로 메시지 m이 송신되었을 때, $FTI(m)$의 쓰레기 처리 조건은

i) $receive_j(m) \in G_Chpt \Rightarrow send_i(m) \in chpt_i^k$이고,

ii) $\exists\, chpt_j^{p-1}$, $chpt_j^{p-1} \rightarrow receive_j(m)$이고,

iii) $\exists\, chpt_j^p$, $receive_j(m) \rightarrow chpt_j^p$이다. ■

[조건 1]의 세 개의 조건이 만족할 때, $garbage_i(FTI(m))$의 사건이 유효하다. 즉, 송신 프로세스에서 불안전 저장 장치에 있는 메시지 로그를 쓰레기 처리할 수 있다. $chpt_j^p$는 프로세스 p_j가 메시지 m의 수신 사건 이후에 취한 검사점이며, $garbage_i(FTI(m))$는 프로세스 p_i가 취할 수 있는 $FTI(m)$의 쓰레기 처리 사건이다. [정의 2]에서 $FTI(m)$이 메시지 로그의 형태와 검사점의 형태로 저장된다고 정의했다. $FTI(m)$의 쓰레기 처리 조건에 의해 메시지 m에 관련된 결함 포용 정보의 쓰레기 처리는 수신 프로세스가 수신 메시지를 포함하는 검사점이 취해진 것을 아는 시점에서 이루어질 수 있다. 즉, $Depend(m)$에 포함되는 프로세스에서 결함이 발생하더라도 손실 메시지가 발생하지 않는 시점에서 쓰레기 처리가 이루어질 수 있다. 이를 위해 이 논문에서는 손실 메시지 관련 프로세스 정보를 정의한다.

그림 5에서 시점 t_i에서 프로세스 P_3에게 메시지를 송

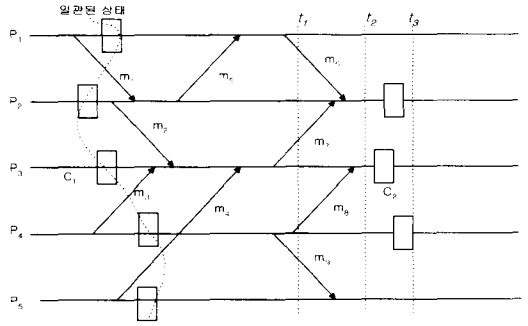


그림 5 손실 메시지 관련 프로세스 정보

신한 프로세스는 P_2, P_4, P_5이다. 시점 t_1에서 P_2, P_4, P_5는 메시지 m_2, m_3, m_4의 메시지 로그를 불안전 저장 장치에 유지하고 있다. 따라서 시점 t_1에서 시점 t_2까지의 시간동안 P_3에서 결함이 발생된다하더라도 각 메시지에 대한 송신자 기반의 메시지 로그가 관리되고 있기 때문에 P_2, P_4, P_5는 불필요한 복귀를 하지 않으며, 손실 메시지는 처리되어 질 수 있다. 또한 시점 t_3에서 P_3에서 결함이 발생하더라도 P_3는 검사점 C_1로 복귀하지 않는다. 따라서 m_2, m_3, m_4의 메시지 로그는 불안전 저장 장치에 유지될 필요가 없으며, 쓰레기 처리될 수 있다. 즉, 각 프로세스가 불안전 저장 장치에서 관리하는 메시지 로그는 [조건 1]을 만족할 경우, 쓰레기 처리된다. 이 조건을 표현하기 위해 손실 메시지 관련 프로세스 정보의 자료 구조를 정의한다.

[정의 3] 손실 메시지 관련 프로세스 정보는 다음과 같이 정의한다.

$Lost\_MSG\_Info_i^j[CI_j]$: P_i에서 P_j로 메시지가 송신되었을 경우, P_i 자신이 송신했던 메시지를 수신하는 각각의 프로세스에 관한 검사점 간격(checkpoint interval)을 저장한다. P_j로부터 수신된 메시지에 부가되어 온 검사점 간격보다 작은 메시지 로그들은 모두 쓰레기 처리되어 질 수 있다. ■

3.2. 손실 메시지 관련 프로세스 정보를 이용한 쓰레기 처리 알고리즘

이 절에서는 [정의 3]에서 정의한 $Lost_MSG_Info_i^j[CI_j]$를 이용하여 메시지 로그를 쓰레기 처리하는 알고리즘을 제안한다.

메시지 m이 프로세스 P_i에서 P_j로 송신되었을 경우, P_i는 m의 메시지 로그를 불안전 저장 장치에 저장한다. m의 메시지 로그는 전역 검사점을 동기화되거나 검사점 간격이 부가된 메시지가 수신된 경우, 검사점 간격값을 검사하여 쓰레기 처리할 수 있다. 전역 검사점의 동기화 기법에서 검사점 동기화 시작 프로세스(coordinated checkpoint initiator process)가 P_i에서 관리되는 메시지 로그의 상대 프로세스라면, 검사점 동기화를 위해 보내온 검사점 간격값을 비교하여 메시지 로그의 쓰레기 처리를 할 수 있다.

P_i의 손실 메시지 관련 프로세스 정보에 저장된 검사점 간격값이 전역 검사점을 동기화 단계나 메시지 수신 사건을 통해 알게 된 상대 프로세스의 검사점 간격값보다 작다면, 메시지 로그 쓰레기 처리 조건을 만족한다. 또한, 그러한 메시지 로그에 대한 쓰레기 처리는 손실 메시지로 인한 불필요한 복귀를 발생시키지 않는다.

그림 6은 손실 메시지 관련 프로세스 정보를 이용한 메시지 로그에 대한 쓰레기 처리 알고리즘이다.

```
process P_i sends process P_j a message
process P_i maintaining message log in volatile storage
on coordinating global checkpointing or receiving
 message from P_k
pivot_CI = Checkpoint_Interval from P_k
while(P_k ∈ P_i's related process list)
{
      if(pivot_CI > Lost_MSG_Info_i^j[CI_j])
            garbage_i(FTI(m))
}
```

그림 6 메시지 로그에 대한 쓰레기 처리 알고리즘

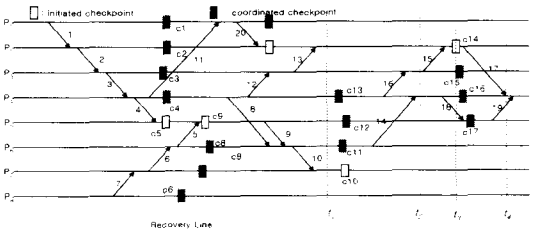


그림 7 손실 메시지 관련 프로세스 정보를 이용한 쓰레기 처리 기법

3.3. 쓰레기 처리 알고리즘의 적용

그림 7의 시점 t_1에서 프로세스 P_4가 관리해야 하는 메시지 로그는 메시지 4, 11, 8, 12이다. 시점 t_2에서 새롭게 관리되어야 하는 메시지 로그는 메시지 16이다. 시점 t_3에서 새롭게 관리되어야 하는 메시지 로그는 메시지 18이다. 하지만 동기적 검사점을 취하는 단계이거나, 수신된 메시지에 송신 프로세스의 검사점 간격이 부가

되어 온다면, 각 시점에서 관리되어야 하는 메시지의 메시지 로그는 감소될 수 있다.

먼저 시점 t_1에서 관리되어야 할 메시지 4, 11, 8, 12의 메시지 로그 중, 메시지 4의 메시지 로그는 불안전 저장 장치로부터 제거되어질 수 있다. 이것은 프로세스 P_4가 P_5와 동기화된 검사점을 취했기 때문이다. 시점 t_2에서 관리되어야 할 메시지 11, 8, 12, 16의 메시지 로그 중, 메시지 8의 메시지 로그는 불안전 저장 장치로부터 제거되어질 수 있다. 이것은 P_4가 메시지 14의 수신으로 P_6이 새로운 검사점을 취했다는 것을 알게 되었기 때문이다. 시점 t_3에서 관리되어야 할 메시지 11, 12, 16, 18의 메시지 로그 중, 메시지 12, 16, 18의 메시지 로그는 불안전 저장 장치로부터 제거되어질 수 있다. 이것은 메시지 12, 16을 수신한 P_3와 메시지 18을 수신한 P_5가 동기화된 검사점을 취했기 때문이다. 시점 t_4에서 관리되어야 할 메시지 로그는 메시지 11이다. 메시지 11은 [정리 2]을 만족하지만 P_3가 결정을 내릴 수 없으므로 나중에 쓰레기 처리가 이루어질 것이다. 만일 불안전 저장 장치의 부족으로 쓰레기 처리를 해야 한다면, 메시지 11을 수신한 P_1에게 검사점 간격값을 요구하여 쓰레기 처리 결정이 내려진다.

4. 정당성 증명

결함 포용 정보의 쓰레기 처리 조건은 결함 발생시 고아 메시지(orphan message)를 만들어서는 안된다. 쓰레기 처리 조건이 유효하다는 것은 쓰레기 처리 시점에서 결함이 발생한다면, 회복 복귀 기법이 복귀와 재수행을 통해 결함 이전의 시스템 상태와 같은 일관된 상태를 유지할 수 있다는 것을 의미한다. 즉, 결함 포용 정보의 쓰레기 처리는 불필요한 정보를 찾아내고, 기억 장치로부터 제거하는 조건이 언제나 유효해야 한다. [정리 1]과 [정리 2]의 증명을 통해 제안된 쓰레기 처리 조건이 결함 발생으로 인한 시스템의 일관된 상태의 재생성이 언제나 가능함을 보인다.

[정리 1] 신뢰적인 통신 기법과 동기적 검사점이 보장된다면, 미수신 메시지로 인한 불필요한 복귀(sympathetic rollback)를 피하기 위해 가장 최근의 검사점 외에 결함 포용 정보 $FTI(m)$를 유지하여야 한다.

증명: 동기적 검사점을 기반으로 하는 회복 복귀 기법이 TCP/IP와 같은 신뢰적인 통신 프로토콜을 기반으로 한다면, 각 프로세스가 가장 최근에 취한 검사점 이후로 복귀하는 경우는 없다. 각 프로세스가 취한 검사점은 관련 프로세스들과 부분적으로 일관된 전역 상태를 갖도록 동기적으로 취해졌기 때문이다. 하지만 결함 발생 시점에서의 미수신 메시지는 전체 시스템의 일관성을 무효화시키지 않음에도 불구하고 회복 복귀 기법이 재생성해 주어야 한다. 일관된 전역 검사점 시점에서의 미수신 메시지는 복귀 회복 수행중에 손실 메시지가 되고, 손실 메시지의 처리는 결함 프로세스의 작업 수행을 불가능하게 하기 때문이다.

손실 메시지로 인해 발생되는 복귀를 불필요한 복귀라 할 때, 불필요한 복귀를 방지할 수 있는 결함 포용 정보가 필요하다.

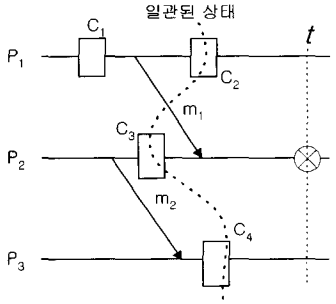


그림 8 손실 메시지에 대한 $FTI(m)$

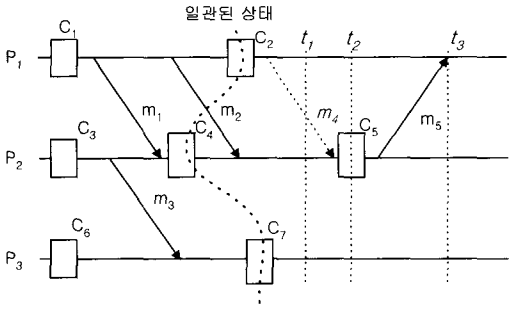


그림 9 손실 메시지에 대한 $FTI(m)$의 쓰레기 처리

그림 8의 시점 t에서 발생한 결함으로 인해 P_2는 검사점 C_3으로 복귀한다. C_3으로 복귀한 P_2는 메시지 m_1의 수신을 필요로 한다. 따라서 P_1은 m_1을 재송신해주어야 한다. 만일 P_1이 m_1의 메시지 로그를 유지하고 있지 않다면, P_1은 검사점 C_1로 복귀해야 한다. 복귀한 P_1은 m_1의 재송신을 위해 C_1로부터 작업을 재수행해야 한다. 따라서 메시지 송신 프로세스는 손실 메시지를 처리하기 위해 가장 최근의 검사점외에 결함 포용 정보 $FTI(m)$인 손실 메시지 관련 프로세스 정보를 유지해야 한다. ■

[정리 2] 동기적 검사점과 신뢰적인 통신 기법이 보

장되고, 프로세스 p_i에서 프로세스 p_j로 메시지 m이 송신되었을 때,

i) $receive_j(m) \in G_Chpt \Rightarrow send_i(m) \in chpt_i^k$이고,

ii) $\exists\, chpt_j^{p-1}$, $chpt_j^{p-1} \rightarrow receive_j(m)$이고,

iii) $\exists\, chpt_j^p$, $receive_j(m) \rightarrow chpt_j^p$ 이라면,

$garbage_i(FTI(m))$을 만족한다.

증명 : [정리 2]의 조건 i) $receive_j(m) \in G_Chpt \Rightarrow send_i(m) \in chpt_i^k$에서 $send_i(m) \in chpt_i^k$를 만족하는 메시지는 그림 8에서 m_1과 m_2이다. 메시지 m_1과 m_2에 대해 조건 ii)과 iii)을 적용시킨 경우는 다음과 같은 두 가지가 존재한다.

가) 먼저 메시지 m_1의 경우, 검사점 C_4의 시점에서 프로세스 P_2의 이전 검사점을 C_3이라 할 때, ii) $\exists\, chpt_j^{p-1}$, $chpt_j^{p-1} \rightarrow receive_j(m)$을 만족하는 검사점 $chpt_j^{p-1}$인 검사점 C_3이 존재한다. 그리고 조건 iii)를 만족시키는 검사점 C_4가 존재한다. 따라서 메시지 m_1는 손실 메시지가 되지 않으므로 쓰레기 처리되어 질 수 있다.

이제 시점 t_1에서 결함이 발생한다고 가정해 보자. 시점 t_1에서 메시지 m_1의 손실 메시지 관련 정보는 이미 삭제되었다. 하지만 결함 발생으로 인해 P_2가 복귀하는 지점은 C_4이므로 m_1은 손실 메시지가 되지 않는다. 따라서 [정리 2]에 의한 메시지 m_1의 손실 메시지 관련 정보는 쓰레기 처리 가능하다.

나) 메시지 m_2의 경우, 프로세스 P_2의 검사점 C_4이전의 검사점을 C_3이라 하고, 조건 ii) $\exists\, chpt_j^{p-1}$, $chpt_j^{p-1} \rightarrow receive_j(m)$을 만족하는 검사점 $chpt_j^{p-1}$은 검사점 C_4가 존재한다. 그리고 검사점 C_5가 취해진다면, 조건 iii)를 만족하게 된다. 따라서 메시지 m_2에 대한 손실 메시지 관련 정보는 검사점 C_5가 취해지는 시점부터 쓰레기 처리되어질 수 있다.

이제 메시지 m_2가 쓰레기 처리되어 질 수 있는 시점에 대해 논의하자. 시점 t_1에서 결함이 발생한다고 가정해 보자. P_2는 검사점 C_3로 복귀할 것이다. 복귀한 P_2는 재수행을 할 것이고, 메시지 m_2는 손실 메시지가 되므로, 메시지 m_2의 손실 메시지 관련 정보가 필요하다. 따라서 P_1은 $FTI(m)$을 쓰레기 처리할 수 없다.

시점 t_2에서 P_2는 검사점 C_5를 취했으므로 조건 iii)를 만족하고, 시점 t_2와 시점 t_3사이에서 결함이 발생하여도 메시지 m_2은 손실 메시지가 되지 않으므로, m_2의 손실 메시지 관련 정보는 쓰레기 처리되어 질 수 있다. 하지만 P_1은 검사점 C_3가 취해졌다는 것을 알 수 없으므로, 메시지 m_2의 손실 메시지 관련 정보는 부가적인 쓰레기 처리 메시지가 존재하지 않는다면, 쓰레기 처리되어질 수 없다.

시점 t_3에서 P_1은 메시지 m_5에 부가되어 온 정보를 통해 검사점 C_5가 취해졌다는 것을 알 수 있다. 따라서 메시지 m_2의 손실 메시지 관련 정보는 쓰레기 처리되어 질 수 있다.

각 시점 t_1, t_2, t_3에서 손실 메시지 관련 정보의 쓰레기 처리 시점은 시점 t_2, t_3에서 가능하며, 결함이 발생하여도 불필요한 복귀를 필요로 하지 않는다. 하지만 P_1에서의 쓰레기 처리 가능 시점은 P_2로부터의 메시지 수신 사건이 발생한 시점인 t_3이다.

가)와 나)에서의 증명에 의해 [정리 2]의 조건을 만족시키는 메시지 m_2의 손실 메시지 관련 정보는 쓰레기 처리가 가능하다. ■

[정리 2]의 증명에서 나)의 경우, 손실 메시지 관련 정보의 쓰레기 처리는 메시지 m_5의 수신 사건의 발생 시점 t_3에서 가능하였다. 하지만 P_2에서의 검사점 C_5가 P_1의 동기화된 검사점을 요구한다면, P_1은 P_2와 동기화된 검사점을 취하게 되고, P_2의 검사점 간격값이 증가된다는 것을 알게된다. 시점 t_2에서 조건 i), ii), iii)을 만족하므로 메시지 m_2의 손실 메시지 관련 정보는 쓰레기 처리되어질 수 있다. 또한 시점 t_2에서 손실 메시지 관련 정보의 쓰레기 처리가 가능하다는 것을 알게된다.

5. 기존 연구와의 비교

동기적 검사점 기법 기반의 기존 연구들[5, 9, 10, 11, 12]은 결함 포용 정보의 쓰레기 처리를 위해 일관된 회복선을 찾아내고, 회복선 이전의 모든 회복 정보를 삭제하는 방법을 사용했다. 이것은 동기적 검사점 기법에서의 결함 발생은 관련된 각각의 프로세스가 가장 최근의 검사점까지만 복귀하므로 이전의 모든 검사점은 삭제될 수 있었기 때문이었다. 이것은 [정의 1]을 기반으로 한 일관된 전역 검사점 정의를 기반으로 하였다. 하지만 동기적 검사점 기법이 신뢰성을 갖는 통신 기법(TCP/IP 프로토콜)을 기반으로 구현된다면, 결함 발생은 손실 메시지를 발생시키며, 손실 메시지는 전체 시스템의 불필요한 복귀를 발생시킨다. 본 논문에서 제안된 동기적 검사점 기법에서의 쓰레기 처리 기법은 가장 최근의 검사점이 취해지더라도 손실 메시지를 발생시킬 수 있는 검사점의 삭제를 지연시킴으로써 불필요한 복귀를 발생시키지 않았다. 이러한 사실은 본 논문의 [정리 1]과 [정리 2]에서 증명되었다. 불필요한 복귀의 회피는 회복 시간을 감소시킬 수 있다. 기존 연구에서는 검사점과 검사

점 사이를 하나의 회복 단위로 처리하여, [정의 1]을 기반으로 검사점 단위의 쓰레기 처리가 되어졌다. 하지만 본 논문에서는 [정의 1]보다 더 제한적인 쓰레기 처리 조건 [조건 1]을 제안하였으며, 이를 기반으로 검사점의 지연된 쓰레기 처리 조건인 [조건 1]을 기반으로 한 쓰레기 처리 기법이 신뢰적인 통신 기법을 기반으로 한 결함 포용 시스템에서 불필요한 복귀를 발생시키지 않는다는 것을 [정리 1]과 [정리 2]의 증명을 통해 보였다.

또한 [정의 2]와 [정의 3]을 통해 동기적 검사점 기법에서 필요한 결함 포용 정보의 성질을 보였으며, 본 논문에서는 검사점을 이용함으로써 메시지 로그와 같은 부가적인 결함 포용 정보없이 기본적인 검사점만을 사용하여 불필요한 복귀를 회피하는 쓰레기 처리 기법을 제안하였다.

6. 결론 및 향후 연구

이 논문은 동기적 검사점 기법에서 결함 포용을 목적으로 불안전 저장 장치에 저장되는 메시지 로그과 안전 저장 장치에 저장되는 검사점의 쓰레기 처리 기법을 제안하였다. 기존의 동기적 검사점 기법을 기반으로 한 결함 포용 정보 쓰레기 처리 기법은 가장 최근의 검사점을 제외한 모든 결함 포용 정보를 쓰레기 처리하였다. 하지만 TCP/IP와 같은 신뢰적인 통신 기법을 기반으로 한 동기적 검사점 기법이 가장 최근의 검사점만을 복귀 회복 기법에서 사용한다면, 손실 메시지로 인한 불필요한 복귀(sympathetic rollback)가 발생된다. 이러한 불필요한 복귀를 없애고, 손실 메시지를 해결하기 위해 각 프로세스가 동기화된 가장 최근의 검사점외에 검사점이나 메시지 로그를 유지해야 한다는 것을 입증했다. 또한 손실 메시지 해결을 위해 관리되어야 하는 검사점이나 메시지 로그가 쓰레기 처리되어지기 위해 필요한 조건을 새롭게 정의하며, 이 정의를 기반으로 한 검사점과 메시지 로그의 쓰레기 처리 알고리즘을 제안하였다.

이 논문에서 제시된 결함 포용 정보의 쓰레기 처리 조건을 기반으로 한 검사점과 메시지 로그의 쓰레기 처리 기법은 송수신 메시지에 부가된 손실 메시지 관련 프로세스 정보를 이용하므로 쓰레기 처리를 위한 부가적인 메시지를 발생시키지 않는다. 손실 메시지 관련 프로세스 정보는 각 메시지에 부가되었으며, 메시지 송신 프로세스의 검사점 간격값만을 포함하였다. 이런 이유에서 수신 메시지가 없거나 동기화 검사점의 그룹에 포함되지 않는다면, 메시지 로그의 쓰레기 처리가 이루어지지 못했다. 하지만 '지연 쓰레기 처리 현상'은 분산 시스템의 일관성을 위배하지 않는다. 이러한 단점을 해결하기 위해 인과적 관계에 있는 모든 프로세스의 검사점 간격값을 송신 메시지에 부가하는 기법을 연구중이며, 인과적 관계의 손실 메시지 관련 프로세스 정보의 쓰레기 처리 기법도 함께 연구중이다.

참 고 문 헌

[1] Yunlong Liu, Junliang Chen, "On Thorough Garbage Collection in Distributed Systems," *Proceedings of Third IEEE Symposium on Computers and Communications*, pp. 576-581, 1998.

[2] Jian Xu, Robert H. B. Netzer, Milon Mackey, "Sender-based Message Logging for Reducing Rollback Propagation," *Seventh IEEE Symposium on Parallel and Distributed Processing*, pp. 602-609, 1995.

[3] D.B. Johnson, W. Zwaenpoel, "Sender-based message logging," *Proceedings of the seventeenth International Symposium on Fault-Tolerant Computing*, pp.14-19, Jun. 1987.

[4] M. V. Sreenivas, Subhash Bhalla, "Garbage Collection in Message Passing Distributed Systems," *First Aizu International Symposium on Parallel Algorithms/Architecture Synthesis*, pp. 213-218, 1995.

[5] R. Koo, S. Toueg, "Checkpoint and Rollback-Recovery for Distributed Systems," *IEEE Trans. on Software Engineering*, Vol. 13, pp.23-31, Jan. 1987.

[6] D. Manivannan, Mukesh Singhal, "A Low-Overhead Recovery Technique Using Quasi-Synchronous Checkpointing," *Proceedings of the 16th ICDCS*, pp100-107. 1996.

[7] K. M. Chandy and L. Lamport, "Distributed Snapshots: Determining Global States of Distributed Systems," *ACM Symp. Principles of Database Syst.*, pp. 63-75, Vol. 3, No. 1, Feb. 1985.

[8] Mootaz Elnozahy, Lorenzo Alvisi, Yi-Min Wang, David B. Johnson, "A Survey of Rollback-Recovery Protocols in Message-Passing Systems," *Technical Report CMU-CS-96-181, Department of Computer Science, Carnegie Mellon University*, Sept. 1996.

[9] E. N. Elnozahy, D. B. Johnson, W. Zwaenepoel, "The Performance of Consistent Checkpointing," *In Proc. IEEE Symp. Reliable Distributed Systems*, pp. 39-47, Oct. 1992.

[10] P. Ramanathan, K. G. Shin, "Use of Common Time base for Checkpointing and Rollback Recovery in a Distributed System," *IEEE Trans. on Software Engineering*, Vol. 9(6), pp. 571-583,

June 1993.
[11] Z. Tong, R. Y. Kim, W. T. Tsai, "Rollback Recovery in Distributed Systems using Loosely Synchnonized Clocks," *IEEE Trans. on Parallel and Distributed Systems*, Vol. 3(2) pp. 246-251, March 1992.
[12] L. M. Silva, J. G. Silva, "Global Checkpointing for Distributed Program," *In Proc. IEEE Symp. Reliable Distributed Systems*, pp. 155-162, Oct. 1992.

정 광 식

1993년 고려대학교 컴퓨터학과 학사. 1995년 고려대학교 컴퓨터학과 석사. 1995년 ~ 현재 고려대학교 컴퓨터학과 박사과정. 관심분야는 분산시스템, 이동 컴퓨팅 시스템, 결함포용시스템

유 헌 창

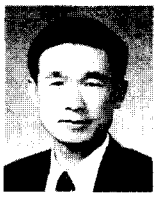
1989년 고려대학교 이과대학 컴퓨터학과 졸업. 1991년 고려대학교 대학원 컴퓨터학과 졸업(이학석사). 1994년 고려대학교 대학원 컴퓨터학과 졸업(이학박사). 1995년 ~ 1997년 서경대학교 이공대학 컴퓨터공학과 조교수. 1998년 ~ 현재 고려대학교 사범대학 컴퓨터교육과 조교수. 관심분야는 분산 시스템, 이동 컴퓨팅 시스템, 결함 포용 시스템, 웹기반교육

이 원 규

1985년 고려대학교 문과대학 영어영문학과 졸업(문학사). 1989년 츠쿠바대학 대학원 전산학과 졸업(공학석사). 1993 츠쿠바대학 대학원 전자정보공학과 졸업(공학박사). 1993년 ~1995년 한국문화예술진흥원 문화정보본부 책임연구원. 1996년 ~ 현재 고려대학교 사범대학 컴퓨터교육과 부교수. 관심분야는 데이터베이스, 정보검색, 분산시스템 등

이 성 훈

한남대학교 컴퓨터공학과 졸업. 고려대학교 컴퓨터학과 졸업(이학석사). 고려대학교 컴퓨터학과 졸업(이학박사). 1998년 ~ 현재, 천안대학교 정보통신학부 교수. 관심분야는 분산시스템, 유전알고리즘, Bioinformatics 등



황 종 선

1978년 Univ. of Georgia, Statistics and Computer Science 박사. 1978년 South Carolina Lander 주립대학교 조교수. 1981년 한국표준연구소 전자계산실 실장. 1995년 한국정보과학회 회장. 1982년 ~ 현재 고려대학교 컴퓨터학과 교수. 1996년 ~ 현재 고려대학교 컴퓨터과학기술대학원 원장. 관심분야는 알고리즘, 분산시스템, 데이타베이스 등