통신 패턴에 기반한 메시지 로깅#

김기범, 유헌창*, 안진호, 황종선
고려대학교 컴퓨터학과 분산시스템연구실, *고려대학교 컴퓨터교육과

Communication Pattern Based Message Logging

Kibom Kim, HeonChang Yu*, JinHo Ahn, Chong Sun Hwang
Dept. of Computer Science and Engineering, Korea Univ, *Dept of Com. Sci. Education

요 약

메시지로깅을 이용한 기존의 메시지 로깅기법에서는 모든 메시지에 대한 처리를 동일하게 수행하였다. 반면, 이 논문에서는 통신 패턴에 대한 연구를 바탕으로 통신시에 발생하는 중요한 성질인 통신 국부성을 발견하였다. 이를 기반으로 프로세스의 통신 집합을 유지하도록 함으로써 결함 발생 후 모든 프로세스에게 도움을 요청하는 메시지를 보는 것이 아니라 오직 자신과 통신을 수행했던 프로세스에게만 메시지를 보내는 기법을 제안한다

1. 서 론

메시지 로깅 프로토콜[1,3,4,5,7]은 분산시스템에서의 신뢰성과 가용성을 높여주기 위하여 많이 사용되어 지고 있는 기법이다 이 기법은 프로세스의 상태가 초기상태와 메시지의 수신 순서에 따라서 결정된다는 부분 결정성에 기반을 두고 있다. 부분 결정성에서의 중요한 성질로서, 재수행하더라도 같은 결과를 만들어낸다는 등멱성(idempotent)이 있다.

메시지 로깅 프로토콜은 언제 어디에 메시지 정보를 저장하는 가에 따라서 비관적, 낙관적, 인과적 메시지 로깅기법으로 나뉜다. 비관적 메시지 로깅 기법[4]에서는 메시지를 보낼 때 혹은 메시지를 받을 때 안전한 저장장치에 회복에 필요한 정보를 저장하고 나서 정상 수행을 계속하는 동기적인 방법으로서 정상 수행시 성능 저하를 초래한다 반면, 결함 발생시 정상인 프로세스들의 복귀가 필요없다는 장점을 가지고 있다 정상 수행시 성능 향상을 위해서 제안된 낙관적 메시지 로깅 기법[5,7]은 프로세스의 수행과 비동기적으로 복귀에 필요한 정보를 안전한 저장장치에 저장한다. 이로 인해 정상인 프로세스가 다른 프로세스의 결함으로 인해 복귀를 해야하는 문제점을 갖고 있다. 인과적 메시지 로깅기법[1,3]은 비관적 기법과 낙관적 기법의 장점을 결합한 방법으로서 복귀에 필요한 정보를 다른 프로세스들에 중복시킴으로써 정상 수행시 안전한 저장장치로의 접근 비용을 줄이고, 결함 발생 후 정상 프로세스의 복귀를 제거한 방법이다 인과적 메시지 로깅기법은 복귀에 필요한 정보가 많은 프로세스에 흩어져 있음으로 해서 결함 발생 후 회복시간이 크다는 단점과 어플리케이션 메시지를 보내면서 피기백해서 보내는 메시지 양이 많은 단점을 갖고 있다.

현존하는 인과적 메시지 로깅 프로토콜과 낙관적 메시지 로깅 프로토콜은 고아 프로세스의 식별을 결함이 발생한 프로세스가 직접할 수 없고 다른 정상인 모든 프로세스에게 정보를 획득하고 나서야 가능하다 이는 기본적으로 결함이 발생한 프로세스가 어떤 프로세스로 메시지를 보냈는가를 모르기 때문에 발생한다. 이러한 문제점을 해결하기 위해, 우리는 통신 패턴을 분석하는 중에 분산시스템에서의 통신의 중요한 성질인 통신 국부성(communication locality)을 발견하였다 통신 국부성이란 분산시스템에서의 프로세스간 통신시에 무작위로 모든 프로세스에게 메시지를 보내는 것이 아니라 관련된 서로 상호 작용하고 있는 프로세스에게만 메시지를 주고 받고 있다는 사실이다 제안하는 기법에서는 자신과 통신하는 프로세스들의 집합을 유지하여 기존 기법의 문제점을 해결하고자 한다. 즉, 프로세스들은 통신 국부성에 기반하여 자신과 통신하는 프로세스들의 집합인 CSet(Communication Set)을 유지하면 고아 프로

이 연구는 한국 과학재단 '98핵심전문연구(과제번호 981-0924-126-2)인 "분산시스템에서 비동기적 회복기법의 개발"의 지원을 받은 것임

세스는 항상 CSet의 원소가 된다. 따라서, 결함 발생 후 모든 프로세스에게 메시지를 보내는 것이 아니라 오직 CSet에 속한 프로세스에게만 메시지를 보내면 된다.

2. 시스템 모델

분산시스템은 N개의 실패-정지(fail-stop) 프로세스들로 이루어져 있다 프로세스들의 통신은 오직 메시지를 교환함으로써 가능하다 또한 분산시스템은 비동기적으로 수행된다. 즉, 프로세스의 상대적인 수행속도에 대한 제한이 없고, 메시지 전송지연에 대한 한계치가 없고 전역 시계 자원이 존재하지 않는 시스템이다

프로세스들은 실패-정지 결함 모델에 따라서 독립적으로 실패가 발생한다. 이 모델에서 프로세스는 실패가 발생하면 바로 정지하게 되고 정지했다는 사실을 궁극적으로는 정상인 모든 프로세스들이 탐지해 낼 수 있다

통신 패턴	패턴에 대한 설명
R	Requests and replies Pattern으로서, 한 프로세스가 요구 메시지를 보내면 다른 프로세스는 그에 대한 응답 메시지를 보내는 형태의 통신 유형
B	Broadcast Pattern으로서, 임의의 한 프로세스에서 모든 프로세스에게 메시지를 전송하는 통신 유형
O	One-Way Pattern으로서, 단방향으로 메시지가 전송되는 형태의 통신 유형(임의의 i에 대하여 P_i에서 P_{i+1}로 메시지를 보내고 P_{i+1}이 P_{i+2}로 메시지를 보내는 형태)
C	Circular Pattern으로서, 프로세스들을 논리적인 링(ring)으로 구성한 형태로 메시지가 전송되는 통신 유형
H	Hierarchical Pattern으로서, 프로세스들이 트리 형태로 부모-자식간에 메시지를 송수신하는 통신 유형
P	Partitioned Pattern으로서, 분산시스템의 위의 패턴을 시스템에서 부분적으로 갖고 있는 형태의 통신 유형

<그림 1> 분산시스템의 통신 패턴

3. 통신 패턴에 기반한 메시지 로깅 기법

기존의 메시지 로깅 기법에서는 메시지에 대한 처리를 모든 메시지에 대하여 동일하게 수행하고 있다. 이는 획일적인 방법으로서 시스템에서 추출해 낼 수 있는 정보를 이용하고 있지 않다. 이를 개선하기 위하여 우리는 분산시스템에서 갖고 있는 통신의 유형을 <그림 1>과 같이 구분하였다 [2,6].

<그림 1>의 통신 패턴을 분석해 보면, 임의의 프로세스가 메시지를 주고 받는 프로세스가 제한적임을 알 수 있다 우리는 이러한 사실로부터 통신에 있어서도 국부성(locality)이 존재한다는 사실을 발견할 수 있었다 즉, 프로세스의 임의의 수행시점을 살펴보았을 때 통신하고 있는 프로세스는 매우 제한적이라는 사실이다

통신의 국부성을 이용하면, 지금까지 제안된 기법에서 결함 발생 프로세스가 모든 프로세스에게 도움을 요청하는 브로드캐스트 메시지를 이용할 필요가 없어진다. 이는 송수신을 수행하고 있는 해당 프로세스에 대한 목록을 유지해 줌으로서 가능하다 즉, 목록내에 있는 프로세스에게만 도움을 요청하면 된다 이러한 통신의 국부성은 낙관적 메시지 로깅기법에서 회복 속도를 느리게 하는 고아 프로세스의 식별에도 이용되어 질 수 있다.

통신의 국부성을 지원해 주기 위하여 기존 메시지 로깅 알고리즘에서 추가되는, 임의의 프로세스 P_i가 유지하는 자료구조는 다음과 같다.

$CSet_i$

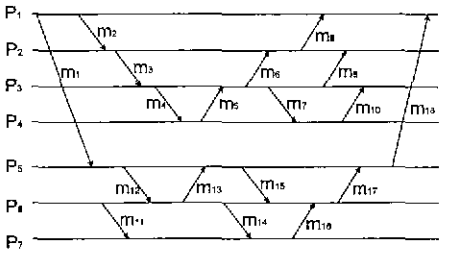
자신과 메시지를 송수신한 프로세스들 집합으로서, 초기값은 공집합이다. 시스템의 성능향상을 위하여 프로세스의 수행을 시작하는 시점에서 자신과 통신하게 되는 프로세스의 집합을 정의해 줄 수도 있다

본 논문에서 발견한 통신의 국부성을 기존의 메시지 로깅 알고리즘에 적용시키기 위해서는 메시지를 송수신하기 전에 대응하는 프로세스가 CSet

```
When process P_i send or receive a message to(or from) P_j,
if ( P_j ∉ CSet_i ) {
        CSet_i = CSet_i ∪ {P_j};
        logging CSet_i to stable storage;
}
```

<그림 2> 메시지 송수신시 알고리즘

에 존재하는가 여부를 검사하고 만약 존재하지 않는다면 해당 프로세스를 CSet에 추가해주고 CSet값을 안전한 저장장소에 저장해 주면 된다. 이때 안전한 저장장소로 접근하게 되는 경우는 통신의 국부성을 벗어나 지금까지 통신을 수행하지 않던 프로세스와 통신이 발생한 것이기 때문에 자주 발생하지 않는 사건이다. 이를 위한 알고리즘은 <그림 2>와 같다.



<그림 3> 분산시스템의 수행 예제

<그림 3>의 분산시스템의 수행예를 살펴보자. 프로세스 P_3와 P_4는 통신 패턴 중 R 패턴을 이용하여 통신을 수행하고 있다 이때 P_4와 통신을 수행하고 있는 통신집합 $CSet_4$는 $\{P_3\}$가 된다. 또한 프로세스 P_2의 통신집합 $CSet_2$는 $\{P_1, P_3\}$이다. 프로세스 P_2인 경우 수행을 시작하는 시점에서는 통신 집합이 공집합이다. 메시지 m_2를 수신하면서 프로세스 P_1이, m_3를 송신하면서 P_3가 $CSet_2$에 추가되고 안전한 저장장치에 자신의 통신집합이 저장된다. 이후 프로세스 P_2의 수행에서 모든 메시지에 대하여 추가적인 안전한 저장장치로의 접근이 없다 만약 P_4에서 결함이 발생하면 모든 프로세스에게 자신의 결함 발생사실을 알리는 것이 아니라 P_3에게만 알려주면 된다. 즉, 낙관적 메시지 로깅 기법인 경우에는 P_1의 결함으로 인하여 고아 프로세스가 될 가능성이 있는 프로세스는 P_3 이고, P_3가 만약 복귀(rollback)를 수행해야 한다면 P_3의 통신 집합으로 다시 복귀 메시지를 보내는 형태로 회복을 수행할 수 있다 그리고, 인과적 메시지 로깅기법을 사용하는 경우에 자신의 상태를 회복시켜줄 정보를 갖고 있는 프로세스는 통신 집합에 있는 프로세스가 된다.

4. 결론

이 논문에서는 분산시스템에서 결함 포용을 지원하는 기법을 제시하였다. 기존 논문에서는 모든 메시지에 대한 처리를 동일하게 수행한 반면, 이 논문에서는 통신 패턴을 조사한 결과 통신의 국부성을 찾을 수 있었다 이를 이용하여 메시지 송수신 연산이 발생할 때마다 메시지가 자신의 통신하는 프로세스의 집합에 속하는지를 검사한다. 이는 만약 결함이 발생하였을 때 모든 프로세스에게 도움을 요청하는 것이 아니라, 자신과 통신한 프로세스에게만 요청함으로써 회복 속도를 개선할 수 있는 새로운 기법이다. 현재 통신의 국부성을 기존의 기법에 접목시키는 연구가 진행중이다.

[참고문헌]

[1] Lorenzo Alvisi and Keith Marzullo, "Message Logging: Pessimistic, Optimistic, Causal and Optimal," IEEE Transactions on Software Engineering, pp 149-159 24(2), Feb 1998

[2] G.R. Andrews, "Paradigms for process interaction in distributed programs," ACM Computing Surveys, pp.49-90, 23(1), 1991.

[3] E.N Elnozahy and W. Zwaenepoel. "Manetho: Transparent rollback-recovery with low overhead, limited rollback, and fast output commit" IEEE Transactions on Computers, pp.526-531, 41(5), May, 1992,

[4] David B Johnson, Willy Zwaenepoel, "Sender-Based Message Logging," *In Proc. Conf. on Fault-Tolerant Computing Systems*, pp. 14-19, 1987.

[5] D.B. Johnson and W Zwaenepol, "Recoevery in distributed systems using optimistic message logging and checkpointing," Journal of Algorithms, pp.462-491, 1990

[6] Taesoon Park and Heon Y. Yeom, "Communication Pattern Based Checkpointing Coordination for Fault-Tolerant Distributed Computing Systems," Proceedings of the International Conference on Information Networking, 1998.

[7] S Venkatesan, Tony T Y. Juang, Sridhar Alagar "Optimistic Crash Recovery without Changing Application Messages," IEEE Transaction on Parallel and Distributed Systems, pp. 263-271, Vol. 8, No. 3, March 1997