

이동 컴퓨팅 환경에 적합한 에이전트 기반 결함 내성 기법 설계 및 성능평가

윤재환, 김동령, 박태순
세종대학교 컴퓨터공학과
e-mail: tspark@sejong.ac.kr

Design and Performance of Mobile Agent based Fault-Tolerance Scheme for Mobile Computing Environment

Jaehwan Youn, Dongryung Kim, Taesoon Park
Dept. of Computer Engineering, Sejong University

요 약

이동 컴퓨팅 시스템의 결함 내성을 지원하기 위해 다양한 검사점 조정 기법과 메시지 로깅 기법들이 제시되어왔다. 그러나 이동 호스트의 이동성 때문에 이러한 기법들은 복잡하고 비효율적이 될 수 있다. 본 논문에서는 이동 에이전트 기반의 결함 내성 기법을 제안한다. 제안된 기법에서는 이동 호스트가 자신의 복구 정보를 일정한 거리 내에 두기 위해 두 개의 이동 에이전트들과 고정 에이전트를 사용하며, 에이전트에 의한 복구 정보의 이동이 이동 호스트의 이동과 비동기적으로 이루어지므로, 이동 호스트의 핸드오프를 지연시키지 않는다.

1. 서론

결함 내성은 신뢰성 높은 컴퓨팅 시스템을 만들기 위한 가장 중요한 설계 요소 중 하나이다. 특히 이동 컴퓨팅 시스템에서는 이동 호스트의 이동성 때문에 결함 내성 기능이 주의 깊게 디자인되지 않으면 시스템 복구가 매우 복잡하고 비효율적이 될 수 있다. 이동 호스트의 이동성 문제를 해결하고 복구 비용을 줄이기 위해 다양한 기법들이 제시되어 왔는데, 이들은 크게 검사점 기법과 메시지 로깅 기법으로 분류된다[1].

검사점 기법은 주로 네트워크 부담을 줄이기 위해 검사점 조정을 위한 동기화나 메시지 전송 횟수를 낮추는데 초점을 맞추었지만, 롤백 복구 시의 동기화 문제를 해결 하지는 못하였다. 반면, 검사점과 함께 메시지를 기록하는 메시지 로깅 기법은 검사점 조정이 필요 없다는 장점이 있으나, 이동 호스트가 메시지를 로깅한 곳에서 멀리 떨어지게 되면 복구 시 검사점과 메시지 로그를 가져오는 데 시간이 소비되어 복구 시작 지연이 발생하게 된다.

이러한 문제점을 해결하기 위해 분산 복구 정보 관리를 위한 비관적, 낙관적, 흠 기반의 복구 정보 관리 기법들이 제안되었다[3,4]. 그러나 이 기법들도 이동 호스트의 핸드오프 중 복구 정보의 이동이 요구 되어 핸드오프 지연이 발생한다는 문제점이 있다. 또한, 검사점과 메시지 로그의 이동 시 각 정보의 중요도를 고려하지 않아 효과적인 정보 이동이 이루어지지 않고 있다.

본 논문에서는 기존 분산 복구 정보 관리 기법의 문제점을 해결하기 위해 이동 에이전트 기반의 관리 기법을 제안한다. 먼저, 각 MSS (Mobile Support Station)마다 복구 정보 관리를 위한 고정 에이전트를 하나씩 두어, MSS가 이동 호스트의 위치 관리 및 질의 처리 등 자신의 원래 작업에 집중할 수 있도록 한다. 또한, 검사점 이동과 메시지 로그 이동을 관리하는 두 개의 이동 에이전트를 두어, 두 가지 정보의 중요도에 따라 서로 다른 이동 방식을 선택할 수 있도록 한다. 이동 에이전트에 의한 복구 정보의 이동은 이동 호스트의 핸드오프와 비동기적으

로 진행 되어 핸드오프 지연을 방지하는 효과도 있다.

2. 시스템 모델

2.1 이동 컴퓨팅 시스템 모델

본 논문은 [1,2]에서 제시된 이동 컴퓨팅 시스템 모델을 따른다. 즉, 빠른 유선 네트워크로 연결된 MSS들과 MSS와는 무선 네트워크로 연결되어 MSS의 지원을 받는 이동 호스트(MH)들로 이루어진 이동 컴퓨팅 시스템을 가정한다. 각 MSS는 자신이 관리할 수 있는 물리적 영역이 한정되어 있으며, 한 MSS에 의해 관리되는 지역을 셀(cell)이라고 부른다. MH가 한 셀을 떠나, 다른 셀로 진입 할 때 두 셀을 관리하는 MSS들 사이에 MH의 정보에 관한 교환이 이루어지는데, 이를 핸드오프(hand-off)라 한다. 또한, MH의 위치 정보 관리를 위해서는 HLR과 VLR로 이루어진 2단계 위치 정보 관리 기법을 가정한다.

2.2 검사점, 메시지 로깅, 롤백 복구

본 논문에서는 MH의 결함에 관해 다음과 같은 가정을 한다. MH는 오류 발생 시 즉시 실행을 중단하고, 안전한 저장 장소에 저장된 자신의 이전 상태에서부터 실행을 재개한다. 이를 위해 MH는 주기적으로 자신의 상태를 로컬 MSS의 안전한 장소에 저장하며, 이를 검사점이라 한다. 또한, MH의 결함 발생 시 이전 검사점에 저장된 상태에서부터 실행이 재개되는 과정을 롤백 복구라 부른다.

만약, 서로 다른 MH에서 실행 중이던 두 프로세스가 메시지 교환에 의해 연산 상태에 의존성이 발생 하였다면, 한 MH의 결함은 다른 MH의 상태에도 영향을 미치게 된다. 이러한 영향을 최소화 하는 방법이 메시지 로깅 기법으로, 각 MH는 자신에게 전달된 메시지들을 안전한 저장 장소에 로깅 해 두었다가, 결함 발생 시 MH에서 수행 중이던 프로세스들이 로깅된 메시지를 이용하여 결함 전과 동일한 상태로 복구가 가능하도록 한다.

2.3 분산 복구 정보 관리

MH의 빠른 복구를 위해서는 자신의 검사점과 메시지 로그를 자신의 저장 장치에 저장하는 것이 좋다. 그러나 대부분의 MH의 저장 공간의 제약으로 MH의 로컬 MSS에 검사점과 메시지 로그가 저장

된다[4]. 이 경우 검사점과 메시지 로그는 이동 호스트의 이동에 따라, 여러 MSS들에 분산 저장이 되고, MH의 복구 시 필요한 복구 정보를 찾는 데 드는 시간과 비용이 커지게 된다. 빠른 복구를 위해서는 검사점과 로그 정보를 MH의 이동 시 최근 MSS로 함께 이동 시키는 것이 좋으나, 이 경우 많은 양의 정보를 이동 시키는데 따른 이동 비용과 MH의 핸드오프 지연 등의 문제가 발생 된다.

3. 이동 에이전트 기반의 결함 내성 기법

3.1 개요

본 논문에서 제안하는 이동 컴퓨팅 환경에 적합한 이동 에이전트 기반의 결함 내성 기법의 설계 목표는 크게 두 가지이다. 첫째, 기존에 제안된 결함 내성 기법들에서 복구 정보 관리를 담당하던 MSS의 부담을 최대한 줄여 MSS가 자신의 본래 역할에 집중할 수 있도록 한다. 둘째, 각 MH의 연산 특성에 적합한 효과적인 결함 내성 서비스를 제공한다.

이러한 설계 목적을 달성하기 위해, 제안된 시스템은 세 가지 에이전트를 사용한다. 먼저 복구 에이전트(Recovery Agent, RA)라고 불리는 고정 에이전트가 각 MSS 마다 하나씩 존재하여, 각종 복구 정보들을 임시 보관하며 MSS의 부담을 덜어주는 역할을 한다. 다음은 검사점 에이전트(Checkpointing agent, CPA)와 로그 에이전트(Log agent, LGA)라 불리는 두 개의 이동 에이전트가 존재하여, RA가 임시 관리하고 있는 검사점과 메시지 로그의 관리를 담당한다. 이렇게 두 개의 에이전트를 사용하여 검사점과 메시지 로그를 따로 관리하도록 함으로써, 각 복구 정보의 우선순위에 따른 효율적인 복구 정보 이동 결정을 내릴 수 있게 된다.

3.2 복구 에이전트(RA)

MSS가 존재하는 사이트마다 하나씩의 고정 에이전트인 복구 에이전트가 존재하여, 해당 셀에 방문 중인 MH들의 복구 정보를 관리한다. RA가 관리하는 MH의 복구 관련 정보는 세 가지이다. 첫째, 각 MH가 현재 셀에 들어올 때와 나갈 때 제공하는 MH의 위치 정보, 즉, MH의 이전 위치와 다음 위치, 둘째, 현재 셀에서 MH가 새로운 검사점 설정을 한 경우, MH의 새 검사점, 그리고 MH가 현재 셀에 머무르는 동안 다른 MH들로부터 받은 메시지들의 로그가 그에 해당된다.

이러한 정보들을 관리하기 위해 RA가 수행하는 주요 작업들은 다음과 같다.

- 검사점 관리 : MH가 해당 셀에서 새로운 검사점을 설정하게 되면, 검사점 순서번호와 함께 이를 MSS에 전달한다. MSS는 이를 RA에게 전달하고, RA는 안전한 저장 장소에 MH의 식별자와 검사점 순서 번호와 함께 검사점을 저장한다.

- 메시지 로그 관리 : MH간의 메시지 통신은 MH가 현재 방문 중인 셀을 관리하는 MSS를 통해 이루어진다. 따라서 MH를 위한 메시지 로깅은 MSS의 주도하에 이루어진다. 즉, MSS가 특정 MH에게 메시지를 전달할 때, 그 메시지의 복사본을 RA에게 전달한다. RA는 각 MH의 식별자와 메시지 순서 번호를 메시지 내용과 함께 메시지 로그에 저장한다.

- 위치 정보 관리 : MH가 한 셀에서 다른 셀로 이동을 하게 되면, 두 셀을 관리하는 MSS 사이에 핸드오프 절차가 진행 된다. 이 과정에서 이동전과 이동후의 셀을 담당하는 MSS에 관한 정보가 교환이 되는데, 이 정보들이 RA로 전달되고 관리된다.

3.3 검사점 에이전트(CPA)

검사점 에이전트는 이동 에이전트로 각 MH마다 하나씩 생성되며, 주요 역할은 MH의 최근 검사점을 관리하는 것이다. MH가 새로운 검사점을 설정하면, 일단 현재 위치의 RA가 검사점을 임시 관리하게 된다. 이 검사점은 MH가 현재 셀을 떠날 때 RA로부터 CPA에게로 전달되고, 이 때 CPA는 자신의 이전 검사점 정보를 최신 것으로 교체한다. MH는 일정한 간격을 두고 검사점을 재설정하기 때문에 매 이동 시마다 이 작업을 반복할 필요는 없다. 또한 CPA의 이동과 MH의 핸드오프는 비동기적으로 이루어지기 때문에, MH의 이동 후 CPA가 해당 사이트에 도착할 경우도 있다. 이런 경우, RA는 CPA의 도착 시까지 검사점을 보관하고, CPA가 도착 신고를 할 때 해당 검사점 정보를 제공한다.

검사점을 관리하는데 있어 CPA의 또다른 주요 역할은 검사점의 이동시기를 결정하는 것이다. 결함 발생 시 MH의 복구는 검사점 재저장부터 시작되므로, 검사점은 MH와 가까운 곳에 위치하는 것이 좋다. 이를 위해 CPA는 MH의 이동이 통보 될 때마다, MH를 관리하는 다음 MSS 사이트로 따라 이동하게 된다. 그러나 MH의 이동 시 CPA가 검사점과 함께 이동할 경우, 이동 비용이 너무 커지게 되므로,

어느 정도 거리를 유지한 채 일정 시간 단위 마다 이동하는 것이 효과적일 수 있다. 따라서 복구비용을 최소화하기 위해 MH의 이동 시 CPA도 따라 이동하는 비관적 이동 방식과 일정한 시간 단위로 CPA가 이동하여 이동 비용을 줄이는 시간 기반 이동 방식이 사용될 수 있다.

어떠한 이동시점 결정 방식을 선택하든, 이동 에이전트에 의한 검사점 이동은 MH의 이동과 비동기적으로 이루어지기 때문에, MH의 핸드오프 과정에 시간 지연을 발생시키지는 않는다.

3.4 로그 에이전트(LGA)

로그 에이전트는 검사점 에이전트와 마찬가지로 각 MH마다 하나씩 생성되어, 여러 RA에 분산 저장된 MH의 메시지 로그 정보를 관리하는 이동 에이전트이다. 결함이 발생 한 MH의 복구를 위해서는 검사점의 재저장이 먼저 수행되고 MH가 로그된 메시지를 이용해 연산 재실행을 진행 하게 된다. 따라서 분산된 메시지 로그의 수집은 검사점의 이동에 비해 상대적으로 중요도가 덜하므로, 메시지 로그의 높은 이동 비용을 고려해 볼 때, 전체 메시지 로그의 이동은 바람직하지 않다고 볼 수 있다.

따라서 LGA는 단지 메시지의 분산 정보, 즉 로그를 관리하고 있는 RA의 식별자와 로그 번호만을 관리하게 된다. LGA는 CPA와 마찬가지로 MH의 이동 경로를 따라 MH와는 비동기적인 이동을 한다. LGA가 새로운 사이트에 도착하면, 해당 사이트의 RA는 자신의 식별자와 해당 MH의 로그된 메시지 번호를 LGA에게 전달하고, LGA는 그 정보를 이용 결함 복구 시 어느 RA들에 어떤 순서로 메시지 로그가 저장되어 있는지 알 수 있게 된다.

LGA의 또 다른 역할은 쓰레기 수집으로, 새 검사점이 설정되어 이전 메시지 로그가 필요하지 않게 되었을 때 해당 RA에게 삭제 요청을 보내는 일을 한다.

3.5 오류 복구

MH에 오류가 발생할 경우, MH는 MSS에게 복구 요청을 하게 된다. MSS는 이를 RA에게 전달하고, RA는 해당 MH의 CPA와 LGA에게 이를 전달하여 복구를 시작한다. CPA는 곧바로 자신이 가지고 있는 검사점을 RA에게 전달하여 신속한 복구가 가능하도록 하고, LGA는 로그를 가지고 있는 각 RA들에게 요청해 메시지를 수집, 현재 RA에게 전

달한다. RA는 전달 받은 검사점과 메시지 로그를 이용해 MH를 복구하게 된다.

4. 성능

이동 에이전트 기반 결함 내성 기법의 성능을 평가하기 위해 간단한 시뮬레이션 실험이 이루어졌다. 아래 두 실험 결과는 제안된 기법(MA-based)의 성능을 [4]에서 제안된 비관적 기법(pessimistic), 낙관적 기법(lazy)과 비교한 것이다.

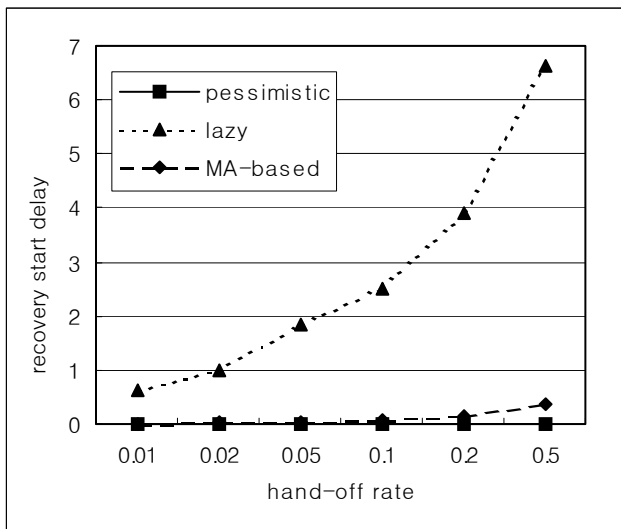


그림 1: 핸드오프 비율에 따른 복구 시작 지연

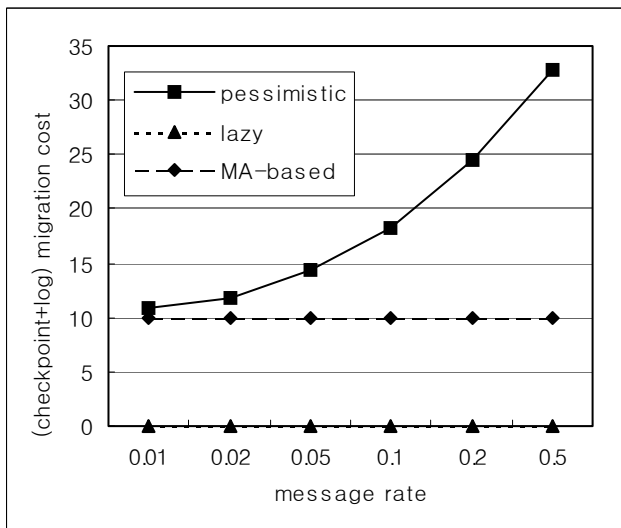


그림 2: 검사점과 로그 이동 비용

첫 번째 성능 기준이 되는 복구 시작 지연은 결함 발생 시 검사점의 위치와 MH의 현재 위치와의 거리를 나타내는 것으로 낙관적기법의 경우 MH의 핸드오프가 많아질수록 검사점과 MH와의 거리가

멀어짐을 볼 수 있다. 이에 비해 이동 에이전트는 MH와 비동기적으로 이동해 MH의 이동이 빨라지면, CPA의 이동이 약간씩 지연이 되기는 하지만, 항상 가까운 거리를 유지함을 알 수 있다.

두 번째 성능 기준은 검사점과 메시지 로그의 이동 비용으로 단위 거리 당 한 메시지 로그의 이동 비용은 1, 검사점 이동 비용은 10으로 계산 되었다. 비관적 기법은 메시지 통신 비용이 높아짐에 따라 고 비용이 요구되지만, 제안된 기법은 일정한 검사점 비용만이 요구되는 것을 볼 수 있다.

5. 결론

본 논문은 이동 컴퓨팅 환경에 적합한 이동 에이전트 기반의 결함 내성 기법의 설계 방법을 논의하고, 제안된 기법의 성능을 알아보았다. 실험 결과에서 보듯 제안된 기법은 상대적으로 적은 검사점 이동 비용으로 이동 호스트의 빠른 복구를 가능하게 한다.

Acknowledgments

본 연구는 정보통신 기초기술연구 지원사업 지원으로 수행되었음 (과제번호: 04-기초-051).

참고문헌

- [1] A. Acharya and B.R. Badrinath, "Checkpointing Distributed Applications on Mobile Computers," Proc. of 3rd Int'l Conf. on Parallel and Distributed Information Systems, pp. 73-80, 1994.
- [2] I.F. Ayildiz and J.S.M. Ho, "On Location Management for Personal Communications Networks," IEEE Communications Magazine, pp. 138-145, 1996.
- [3] B. Yao, K. Ssu, and W.K. Fuchs, "Message Logging in Mobile Computing," Proc. of the 29th Symp. on Fault Tolerant Computing Systems, pp. 294-301, 1999.
- [4] D.K. Pradhan, P. Krishna, and N.H. Vaiday, "Recoverable Mobile Environment Design and Trade-Off Analysis," Proc. of the 26th Int'l Symp. on Fault Tolerant Computing Systems, pp. 16-25, 1996.