

# 이동 에이전트 기반 결함 내성 서비스의 성능 평가

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

## Performance Evaluation of the Mobile Agent Based Fault-Tolerance Service

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

### 요 약

이동 컴퓨팅 시스템에 결함 내성 기능을 제공하기 위한 많은 방법들이 제안되어졌다. 그 중 하나가 이동 에이전트를 이용하여 이동 호스트의 검사점과 메시지 로그를 분산 관리하는 방법이다. 이동 에이전트 기반의 결함 내성 서비스는 기존 기지국 기반의 서비스에 비해, 비동기적인 복구 정보의 이동 및 복구 정보의 유연한 분산 관리 등의 장점이 있으나, 이동 에이전트 자체의 이동 비용 등이 문제가 될 수 있다. 따라서 본 논문에서는 기존의 기지국 기반 복구 정보 관리 기법과 이동 에이전트 기반 복구 정보 관리 기법의 성능을 비교 평가한다. 특히, 다양한 시스템 환경에서 이동 에이전트 자체 비용이 결함 내성 서비스의 성능에 미치는 영향을 분석한다.

### 1. 서론

이동 컴퓨팅 시스템의 신뢰도를 높이기 위해 사용되는 한 가지 방법은 검사점 및 메시지 로깅을 이용한 결함 내성 서비스를 제공하는 것이다[1]. 검사점이란 이동 호스트의 실행 상태를 주기적으로 안전한 저장 장소에 저장하는 것으로, 호스트의 결함 발생 시 최근 검사점이 설정되었던 상태로 이동 호스트의 상태를 되돌리게 된다. 메시지 로깅이란 이동 호스트에게 전달되는 모든 메시지들을 안전한 저장 장소에 저장하는 것으로, 호스트의 상태가 최근 검사점으로 되돌려진 후 로깅된 메시지를 이용하여 연산을 재실행함으로써, 결함 발생 후에도 호스트의 상태를 결함 발생 전과 동일하게 재구성 할 수 있다.

이동 컴퓨팅 시스템에서 검사점과 메시지 로그를 저장하기 위해 안전한 저장 장소를 제공하는 곳은 기지국이다. 따라서 이동 호스트가 여러 지역을 이동하며 연산을 수행하게 되면, 이 호스트를 위한 검사점 및 메시지 로그도 여러 기지국에 분산 저장되게 된다. 이렇게 분산된 복구 정보는 이동 호스트의 결함 발생 시, 그 위치가 추적되어 수집되게 되는데,

저장된 복구 정보의 위치가 이동 호스트의 위치와 멀수록 복구 지연이 발생하게 된다. 이러한 문제를 해결하기 위해 이동 호스트의 이동에 따라 해당 호스트의 복구 정보를 적절한 위치로 이동시키는 다양한 분산 복구 정보 관리 기법들이 제안되어졌다 [4,5].

그러나 이러한 기지국 기반의 복구 정보 관리 기법은 기지국에 과도한 부담을 주고, 복구 정보 이동이 이동 호스트를 위한 핸드오프 과정 중에 발생하므로, 핸드오프 지연 등의 문제가 발생하게 된다. 이러한 문제에 대한 대안으로 제안된 방법이 이동 에이전트 기반의 복구 정보 관리 기법이다[3]. 제안된 기법에서는 검사점과 메시지 로그를 관리하기 위해 두 개의 이동 에이전트가 생성되며, 각 에이전트들은 이동 호스트의 이동과는 비동기적으로 기지국과 교신하며 복구 정보를 이동, 관리한다. 이 기법의 가장 큰 장점은 비동기적인 복구 정보의 이동으로 핸드오프지연이 발생하지 않는다는 것이며, 또한 각 에이전트가 이동 호스트의 행동 특성에 맞는 적절한 복구정보 이동 정책을 선택할 수 있다는 점이다.

이동 에이전트 기반 결함 내성 서비스는 여러 장

점에도 불구하고 이동 에이전트 도입에 따른 비용 부담이 그 문제점으로 지적될 수 있다. 따라서 본 논문에서는 기존의 기지국 기반 복구 정보 관리 기법과 이동 에이전트 기반 복구 정보 관리 기법의 성능을 비교 평가한다. 성능 평가를 위해 이동 컴퓨팅 환경을 시뮬레이트 할 수 있는 실험 시스템을 이동 에이전트 시스템인 Aglets 시스템 상에 구현하였고, 다양한 시스템 환경에서 기지국간에 복구 정보를 이동시키는 경우와 이동 에이전트를 이용하여 복구 정보를 이동시키는 경우의 비용을 분석하였다.

## 2. 연구 배경

### 2.1 기지국 기반 결합 내성 서비스

이동 컴퓨팅 시스템[2]은 빠른 유선 네트워크로 연결된 기지국(mobile support stations, MSS)들과 무선 네트워크로 MSS에 연결되어, 통신 및 이동을 위해 MSS의 지원을 받는 이동 호스트(mobile host, MH)들로 이루어진다. 각 MSS는 자신이 관리할 수 있는 물리적 영역이 한정되어 있으며, 한 MSS에 의해 관리되는 지역을 셀(cell)이라 부른다. MH가 한 셀을 떠나, 다른 셀로 진입 할 때 두 셀을 관리하는 MSS들 사이에서 MH의 정보 교환이 이루어지는데, 이를 핸드오프(hand-off)라 한다.

MH는 결합 발생 시 즉시 실행을 중단하고, 안전한 저장 장소에 저장된 자신의 이전 상태에서부터 실행을 재개한다. 이를 위해 MH는 주기적으로 자신의 실행 상태를 로컬 MSS의 안전한 저장 장소에 저장하며, 이를 검사점이라 한다. 서로 다른 MH에서 실행 중이던 두 프로세스가 메시지 교환에 의해 연산 의존성이 발생 할 경우, 한 MH의 결합이 다른 MH의 실행 상태에 영향을 미치게 된다. 이러한 영향을 최소화하기 위해 각 MSS는 MH에게 전달하는 메시지들을 안전한 저장 장소에 로깅 해 두었다가, 결합 발생 시 로깅 된 메시지들을 재연산하여 결합 전과 동일한 상태로 MH를 복구 한다.

MH의 빠른 복구를 위해서는 검사점과 메시지 로그를 자신의 저장 장치에 저장하는 것이 좋다. 그러나 대부분의 MH는 저장 공간의 제약 때문에 로컬 MSS에 검사점과 메시지 로그를 저장 한다. 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 이동시키는 것이 좋다. [4]에서 제안한 비관적 복구정보 이주 기법은 이동 호스트를 위한 MSS간의 핸드오프 시 검사점과 메시지로그를 다른 MH 관련 정보들과 함

께 새로운 MSS로 전달한다. 이 경우 많은 양의 정보를 이동 시키는데 따른 이동 비용과 MH의 핸드오프 지연 등의 문제가 발생 된다.

### 2.2 이동 에이전트 기반 결합 내성 서비스

이동 컴퓨팅 환경에 적합한 이동 에이전트 기반 결합 내성 기법의 설계 목표는 크게 두 가지이다. 첫째, 기존에 제안된 결합 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 결합 내성 서비스를 제공한다.

이러한 설계 목적을 달성하기 위해, 이동 에이전트 기반 결합 내성 시스템은 세 가지 종류의 에이전트를 사용한다. 먼저 복구 에이전트(recovery agent, RA)라고 불리는 고정 에이전트가 각 MSS 사이트마다 하나씩 존재하여, 각종 복구 정보들을 임시 보관하며 MSS의 부담을 덜어주는 역할을 한다. 즉, MH로 전달받은 검사점과 MH에게 전달된 메시지의 로그는 RA에 의해 일차로 관리된다. 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.

CPA는 해당 MH가 방문한 MSS 사이트들을 이동해 다니며, MH의 검사점 생성 시 RA로부터 최신 검사점을 전달받아 관리한다. LGA 역시 MH가 방문한 MSS 사이트들을 이동해 다니며, 해당 MH를 위해 저장된 로그 정보를 관리한다. 검사점과 로그 정보 관리를 위해 두 개의 에이전트를 사용하는 이유는 각 복구 정보의 우선순위에 따른 효율적인 복구 정보 이동 결정을 내릴 수 있게 하기위해서이다. 결합이 발생 한 MH의 복구를 위해서는 검사점의 재저장이 먼저 수행되고 이후 로깅 된 메시지를 이용하여 연산의 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 가지고 이동하게 된다.

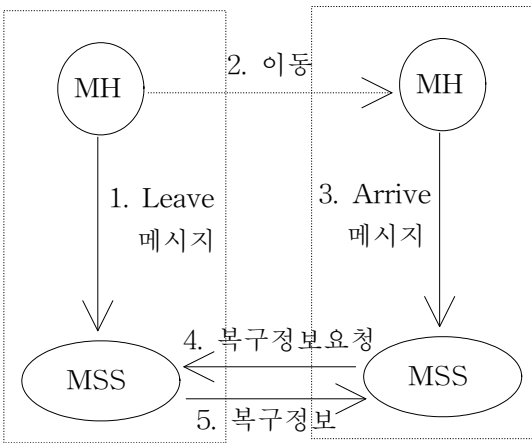
MH에 오류가 발생할 경우, MH는 MSS에게 복구 요청을 하게 된다. MSS는 이를 RA에게 전달하고, RA는 해당 MH의 CPA와 LGA에게 이를 전달하여 복구를 시작한다. CPA는 곧바로 자신이 가지고 있는 검사점을 RA에게 전달하여 신속한 복구가 가능하도록 하고, LGA는 로그를 가지고 있는 각

RA들에게 요청해 메시지를 수집, 현재 RA에게 전달한다. RA는 전달 받은 검사점과 메시지 로그를 이용해 MH를 복구하게 된다.

### 3. 실험 시스템 구현

#### 3.1 기지국 기반 결합 내성 서비스 구현

성능 실험을 위해, 이동 컴퓨팅 환경을 시뮬레이션 하기 위한 실험 시스템을 이동 에이전트 시스템인 Aglets 시스템 상에 구현하였다. 시스템내의 각 플레이스는 하나의 셀을 나타내며, MSS는 플레이스내의 고정 에이전트로 구현하였다. MSS는 해당 셀내의 MH들의 정보와 이웃 셀의 정보를 관리하며, MH간의 통신을 위한 위치 정보 관리를 담당한다. 또한, 방문한 MH들을 위한 검사점과 메시지 로그도 관리한다.



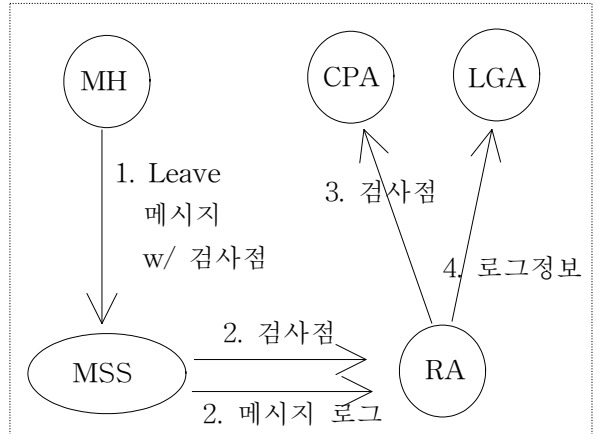
(그림 1) MSS간의 복구 정보 교환

MH는 이동 에이전트로 구현되었다. 각 MH는 시스템 상의 여러 플레이스들을 이동해 다니며, 연산과 다른 MH와의 통신을 수행한다. MH는 또한 주기적으로 검사점을 생성하여 로컬 MSS에게 전달한다. MSS는 MH가 해당 플레이스로 이동해 왔다는 메시지를 받게 되면, MH가 이전에 있었던 MSS에게 핸드오프를 요청하여, 검사점과 메시지 로그를 비롯한 복구 정보를 전달 받는다. 그림 1은 MSS간의 복구 정보 교환 절차를 도식화 한 것이다.

#### 3.2 이동 에이전트 기반 결합 내성 서비스 구현

기지국 기반 결합 내성 서비스와 마찬가지로 각 플레이스는 셀을 나타내며, MSS는 고정 에이전트로 MH는 이동 에이전트로 구현되었다. 이외에 결합 내성 서비스를 제공하기 위해, RA는 고정 에이전트로, CPA와 LGA는 이동 에이전트로 구현되었으며, 각 에이전트의 기능은 2.2절의 설명을 따른다. CPA는 MH의 검사점을 가지고 MH의 이동 경로를 따라 이

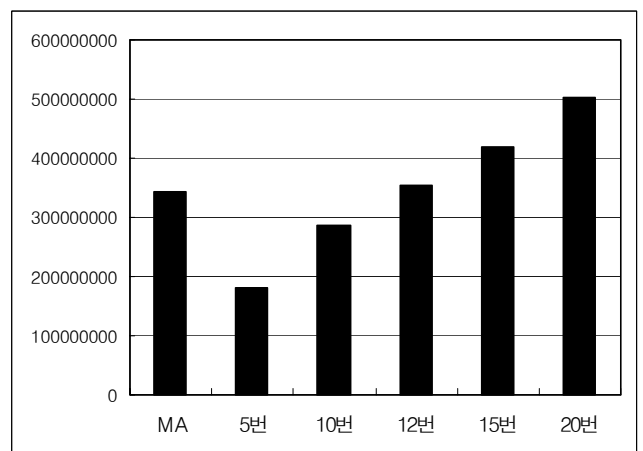
동하며, LGA는 MH의 로그 정보만을 가진 채 MH의 이동 경로를 따라 이동한다. 여기서, 주목할 점은 CPA와 LGA의 이동은 MH의 이동과 비동기적으로 이루어지며, 따라서 MH의 핸드오프를 지연시키지는 않는다는 점이다. 그림 2는 이동 에이전트 기반 결합 내성 기법에서의 복구 정보 교환 절차를 도식화한 것이다.



(그림 2) 이동 에이전트 간의 복구 정보 교환

### 4. 실험 결과

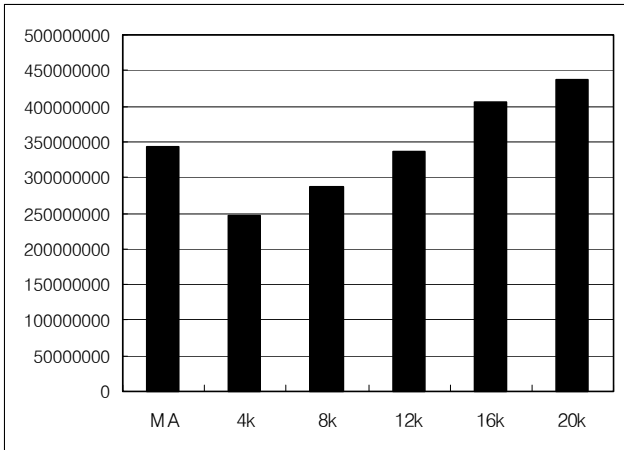
결합 내성 서비스를 구현하는 두 가지 방법의 성능을 평가하기 위해, Aglets 플랫폼을 이용해 총 16개의 셀을 구현하였고, MH의 크기는 검사점 크기가 128Kbytes가 되도록 조정하였다. MSS 간에 검사점 및 메시지 로그를 전달하는 경우, 복구 정보 전달에 소요되는 시간과 CPA 및 LGA가 복구 정보를 가지고 이동하는 경우의 이동 시간을 측정하여 비교하였다. 실험은 검사점 생성 주기, 메시지의 크기와 메시지 전송 횟수 등을 변경해 가면서 진행되었다.



(그림 3) 검사점 생성 주기의 영향

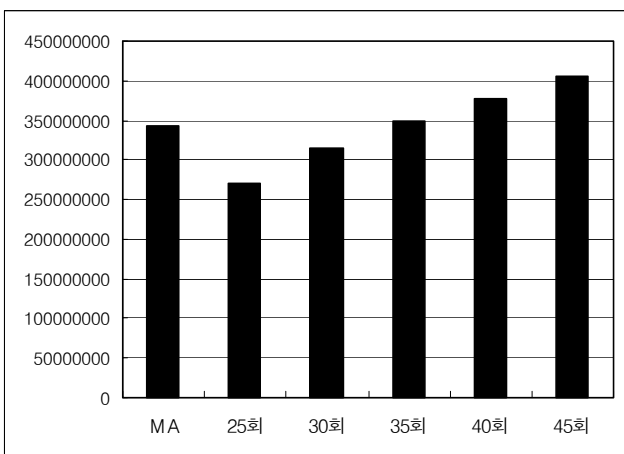
그림 3은 검사점 생성주기를 변경하면서 실험한

결과이다. 이 실험을 위해 MH는 8Kbytes 크기의 메시지를 매 셀마다 10회씩 주고받았다. MH의 복구에 사용되는 메시지는 최근 검사점 설정 이후에 저장된 메시지들이다. 따라서 검사점의 생성 주기가 길어지면, 메시지 로그의 양도 많아지고 이에 따라 MSS 기반 기법의 복구 정보 전달 시간이 길어진다. MA로 표시된 이동 에이전트 기반 기법의 경우에는 메시지 자체를 가지고 이동하지 않기 때문에 로그의 증가가 성능에 큰 영향을 미치지 않는다.



(그림 4) 메시지 크기의 영향

그림 4는 메시지의 크기를 변화시키면서 실험한 결과이다. 검사점은 10번의 핸드오프 시마다 생성되도록 하고 메시지 전송 횟수는 매 셀마다 10번이 되도록 고정하여 실험하였다. 메시지의 크기가 커질수록 MSS간에 전달되는 복구 정보의 크기가 커져 성능에 영향을 미치게 되며, 에이전트 기반 기법은 역시 메시지 크기 변화에 별다른 영향을 받지 않는다.



(그림 5) 메시지 전송횟수의 영향

그림 5는 메시지 전송 횟수를 변경하며 실험한

결과이다. 이 실험에서는 메시지 크기를 8Kbytes로 고정하고, 검사점은 3번의 핸드오프시마다 생성되도록 하였다. 매 셀에서의 메시지 전송 횟수가 증가하면, 메시지 로그의 크기가 커져 MSS간 복구 정보 전달 기법의 경우 성능에 영향을 주게 되며, 전송 횟수가 약 35회 정도보다 커지면, MSS간 복구 정보 교환이 에이전트 기반 기법 보다 낮은 성능을 보이는 것을 볼 수 있다.

## 5. 결론

본 논문에서는 기지국 기반 결합 내성 기법과 이동 에이전트 기반 결합 내성 기법을 이동 에이전트 시스템 상에 구현하여 복구 정보 이동 시간을 비교 실험하였다. 특히, 다양한 시스템 환경에서 이동 에이전트 자체 비용이 결합 내성 서비스의 성능에 미치는 영향을 분석한다.

## 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 the 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] T. Park, "Mobile Agent Based Fault-Tolerance Support for the Reliable Mobile Computing Systems," Proc. of the 7th Int'l Conf. on Coordination Models and Languages, pp. 173-187, 2005.
- [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.
- [5] 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.