

이동 에이전트를 위한 K-결함 허용 검사점 기법의 성능 분석

윤재환, 변일수, 박대순
세종대학교 컴퓨터공학과
e-mail: tspark@sejong.ac.kr

Performance Evaluation of the K-Fault-Tolerant Checkpointing Scheme for Reliable Mobile Agent Systems

Jaehwan Youn, Ilsoo Byun, Taesoon Park
Dept. of Computer Engineering, Sejong University

요 약

신뢰할 만한 이동 에이전트 시스템을 구축하기 위해서는 이동 에이전트의 결함 내성 기능이 중요하다. 지금까지 여러 결함 내성 기법이 제안되었는데, 그 중의 하나가 검사점 기법이다. 에이전트의 중간 상태를 저장하는 검사점 기법은 에이전트 복제 기법에 비해 훨씬 적은 비용을 보장하는 반면, 검사점이 저장된 사이트의 결함 발생 시 에이전트 실행이 일시 또는 영구히 중지되는 문제가 발생한다. 따라서 본 논문에서는 k-결함 허용 검사점 기법을 제안한다. 이 기법에서는 에이전트 이동 경로에 저장된 검사점을 관리하는 관찰자들을 두어, 에이전트 실행 사이트의 결함 발생 시 관찰자간의 여론 수렴 과정을 통해 결함에 영향을 받지 않은 최근 검사점으로부터 에이전트의 실행을 재개시킨다.

1. 서론

이동 에이전트가 좀더 다양한 분야에 적용되기 위해서는 신뢰할만한 에이전트의 수행이 요구된다. 에이전트는 이동 중이나 수행 중에 시스템 오류로 손실될 수 있다. 반대로, 잘못된 오류 복구는 에이전트를 중복 실행시킬 수도 있다. 신뢰할 만한 에이전트 시스템은 에이전트가 아무 오류 없이 정확히 한번(exactly-once) 수행되는 것을 보장해야한다.

신뢰할 만한 에이전트 시스템을 위해 제안된 결함 내성 기법들은 크게 복제 기법[1]과 검사점 기법[2]으로 분류된다. 복제 기법은 $2k+1$ 개의 에이전트 복제 본을 사용하여 k 개의 오류를 견딜 수 있는 반면, 에이전트의 사이즈가 크거나 네트워크 비용이 비싼 환경에서는 비효율적이라는 점이다. 검사점 기법에서는 에이전트를 하나의 사이트에만 보내며 미래의 오류를 대비해 에이전트의 중간 상태를 방문한 사이트에 저장한다. 따라서 이 기법은 오류가 발생한 사이트가 복구될 때까지 에이전트 실행이 중단된

다는 단점을 가지고 있다. 하지만 오류가 없을 경우, 검사점 기법이 복제 기법에 비해 훨씬 나은 성능을 보인다.

k-결함 허용 검사점 기법[3]은 검사점 기법의 저비용을 보장하면서 기존 검사점 기법에서 발생하는 에이전트의 실행 봉쇄를 없애므로써 신뢰도를 향상시킬 수 있는 기법이다. 본 논문에서는 k-결함 허용 검사점 기법의 성능을 기존의 동기식 및 비동기식 복제 기법[4]들과 비교 평가하기 위해, 각 기법들을 이동 에이전트 시스템인 Aglets[5] 상에 구현하고, 성능 측정 결과를 통해 k-결함 허용 검사점 기법의 성능 우수성을 보인다.

2. k-결함 허용 검사점 기법

2.1 검사점 기법

검사점은 에이전트의 실행 상태를 주기적으로 안전장소에 저장함으로써, 시스템내의 결함 발생 시 에이전트를 저장된 상태에서부터 재개시키는 역할

을 한다. 검사점을 설정하는 한 방법은 각 스테이지의 끝 지점에서 검사점을 설정하는 것이다. 이 경우, 검사점은 해당 사이트의 결합이 아니라, 다음 스테이지가 실행되는 사이트의 결합 발생 시 이용된다. 이 경우, 에이전트는 결합이 발생한 사이트의 복구를 기다려 에이전트를 재전송 할 수도 있지만, 결합 사이트의 복구가 늦어지는 경우, 대체 사이트로 에이전트를 전송하여, 에이전트 실행이 지연되거나 손실되는 것을 막을 수 있다.

2.2 k-결합 허용 검사점 기법

각 스테이지의 끝 지점에서 검사점을 설정하는 경우, 다음 사이트의 결합 발생을 관찰하고, 저장된 검사점으로부터 에이전트를 재개시키는 작업을 수행할 관찰자(observer) 에이전트가 필요하다. 그러나 에이전트와 관찰자 에이전트 양쪽 모두 오류가 발생한 경우에는 에이전트의 손실이 불가피해진다. 따라서 k개의 결합 극복을 보장하기 위해서는 적어도 k개의 관찰자가 필요하게 된다. 본 논문에서는 관찰자 에이전트의 생성에 따른 비용을 최소화하기 위해, 에이전트의 이동 경로에 관찰자를 두기로 한다.

대부분의 이동 에이전트 시스템에서는 에이전트의 이동 시, 먼저 에이전트를 복사하여 원격 메소드 호출(RMI)등의 방법으로 복제 에이전트를 이동 시키고, 이동 성공 시 남아 있는 에이전트를 제거한다. 이때 남아있는 에이전트를 제거하지 말고, 관찰자 에이전트로 활용하게 되면, 관찰자 에이전트 생성에 따른 비용을 줄이 수 있다. 관찰자 에이전트는 에이전트가 이동 직전 생성한 검사점을 관리하며, 다음 스테이지에서의 에이전트의 결합이 발견되면 검사점으로부터 에이전트를 재개시키는 역할을 한다.

에이전트는 각 사이트에서의 실행이 끝나고 검사점을 설정 한 후 다음 사이트로 이동하면, 현재의 관찰자 노드들에게 M_{exec_end} 메시지를 보낸다. 본 논문에서는 에이전트 실행 사이트의 결합 발견을 위해 타임아웃을 사용한다. 즉, 각 관찰자 에이전트는 에이전트가 j번째 스테이지로의 이동 후 일정한 시간이 지나도록 M_{exec_end} 메시지가 오지 않으면, 에이전트의 현재 실행 사이트에 결합이 발생하였음을 가정하고, 각 관찰자가 관리하는 검사점으로부터의 복구를 수행한다.

이때, 관찰자들의 복구 시점이 적절히 조정되지 못하면, 다중 실행(duplicate execution)의 문제가 발생 할 수 있다. 이 문제를 해결하기 위해 널리 사용되는 방안으로는 여론 수렴(consensus) 기법이 있

다. 여론 수렴 기법은 에이전트 복제를 이용한 결합 내성 시스템의 설계 시 복제 에이전트간의 실행 조정을 위해 사용되는 기법으로, 에이전트 복제기법을 위한 여론 수렴은 여러 방법이 제안되었지만, 본 논문은 에이전트 간 우선순위를 활용한 여론 수렴 방법[1]을 따른다.

즉, 관찰자 에이전트들은 생성 순서에 따른 우선순위를 가지고 있어, 실행 중인 에이전트의 오류 발생 시 가장 우선순위가 높은 관찰자가 오류 발견 사실을 다른 관찰자들에게 알리고, 응답 메시지를 받은 후 오류가 발생한 에이전트의 실행을 떠맡게 된다. 우선순위가 j번째인 관찰자는 j-1의 우선순위를 갖는 관찰자 에이전트의 결합 발생시에만 복구를 시도한다.

이때 여러 관찰자들의 다중 결합 상황에서도 에이전트의 단일 실행(exactly-once execution)을 보장하기 위해, 에이전트는 각 스테이지 실행 후 투표(voting)에 의한 여론 수렴 과정을 거친다. 즉, 에이전트는 한 스테이지에서의 실행이 끝나면 검사점 설정 후 모든 관찰자들에게 M_{vote} 메시지를 보내고, 이 메시지를 받은 관찰자들은 M_{vote_reply} 메시지를 보낸다. 과반수이상의 M_{vote_reply} 메시지를 받은 에이전트만이 성공적으로 다음 스테이지에 진입하게 되고, 중복 실행된 다른 에이전트들의 실행 결과는 해당 사이트에서 적절히 무효화(undo)가 된다.

k개의 결합 발생 상황에서도, 여론 수렴 과정이 성공적으로 수행되기 위해서는 시스템 내에 적어도 2k개의 관찰자가 필요하다. 이 가운데, 우선순위가 높은 k개의 관찰자는 검사점을 관리하며 에이전트와 자신 보다 높은 우선순위 관찰자의 결합 발생 시 실행 재개를 위한 것이며, 나머지 k개의 관찰자는 여론 수렴 과정에 참여하기 위한 것이다. 이와 같이 2k+1개의 에이전트 및 관찰자를 이용하면, k개의 결합 발생시에도, 적어도 한 관찰자는 과반수의 투표를 얻어 실행을 계속할 수 있게 된다.

3. 실험 환경 및 결과

3.1 실험 환경

우리는 실험을 위해 기존의 동기적인 복제 기법과 검사점 기법을 Aglet SDK 1.1b2를 이용하여 구현하였다. 사용된 컴퓨터는 펜티엄IV 1.8GHz 머신 5대이며 에이전트의 크기를 조정하기 위해 NxN 배열을 사용하였고 실험 결과 값은 총 10번 실험한 후 최고 값과 최저 값을 제외한 8개의 평균값을 사용하

였다.

3.2 실험 시스템 구현

3.2.1 에이전트 클래스 구조

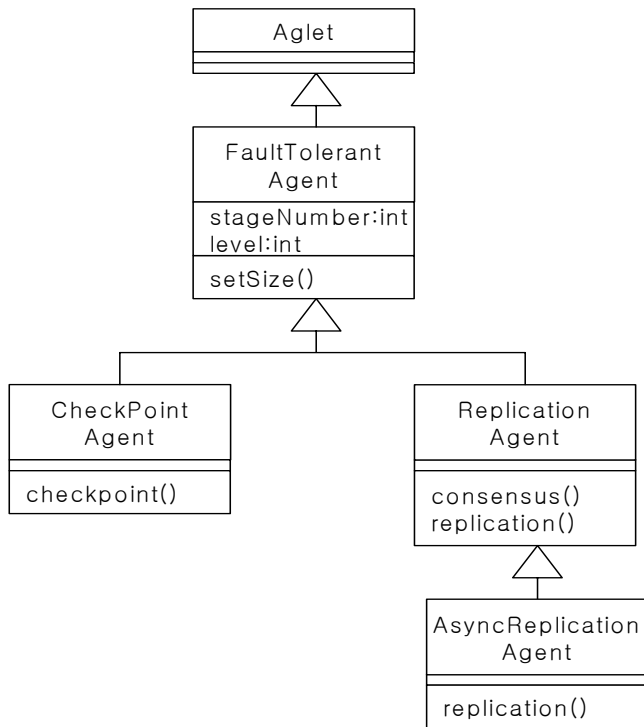


그림 1. 클래스 계층 구조

Aglet은 Aglets 플랫폼에서 이동 가능한 Agent를 뜻하며 이동이나 메시지 송수신과 관련된 모든 기능이 구현되어 있다. 이를 상속한 FaultTolerantAgent에서는 모든 결함 내성 기능 에이전트에서 필요로 하는 기본 적인 기능들을 구현하고 있다. 에이전트의 현재 스테이지를 뜻하는 stageNumber 필드를 가지고 있으며 이동시 자동으로 이 값은 증가되도록 되어 있다. level은 이를 상속한 결함 내성 에이전트들이 어느 정도의 결함내성정도를 지원하게 될 것인지를 뜻한다. setSize() 메소드에서는 에이전트의 크기를 조정하는 코드가 담겨 있다.

검사점 기법을 이용한 에이전트(CheckpointAgent)나 복제 기법을 이용한 에이전트(ReplicationAgent)는 모두 FaultTolerantAgent로부터 상속된다.

3.2.2 복제 기법의 구현

복제 기법 에이전트의 이동시 다음과 같은 일들이 이루어진다.

1. 이동전에 테이블에 등록된 다른 복제 본들에게 동의를 구한다.
2. 과반수이상의 승인 메시지가 도착한다.
3. 자신의 복제 본들을 level에 따라 생성한다.

4. 복제 본들의 프락시를 테이블에 등록한다.

5. 생성된 복제 본들을 다른 사이트로 이동시킨다.

6. 자신도 다른 사이트로 이동한다.

3.2.3 검사점 기법의 구현

검사점 기법에서는 checkpoint()메소드를 구현해야 하는데 여기에는 두 가지 방법이 있다. 에이전트의 중간 상태를 디스크에 저장하는 방법과 메모리에 중간 상태 정보를 남겨두는 방법이다. 우리는 후자의 방법을 택했으며 전자의 방법을 사용하기 위해서는 checkpoint() 메소드만을 재정의 하면 된다.

3.3 실험 결과

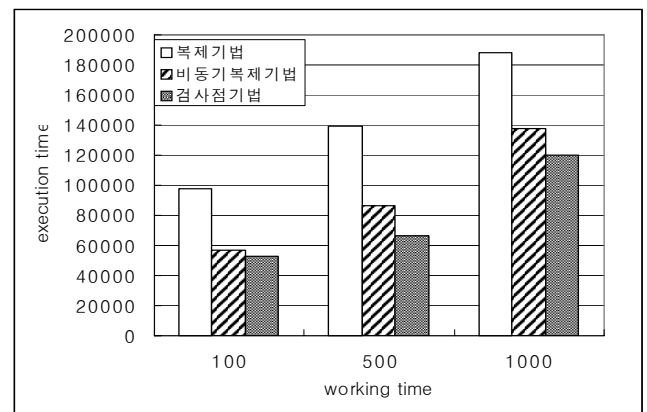


그림 2. 워킹타임 변화에 따른 성능

그림 2에서는 에이전트의 크기를 1KBytes로 하였으며 워킹 타임 변화에 따른 성능 변화를 보여준다. 비동기 복제 기법이 복제 기법보다 훨씬 나은 성능을 보이지만 검사점 기법보다는 못한 성능을 보이고 있다. 이러한 현상이 발생하는 것은 비동기 복제 기법이 검사점 기법보다 더 긴 동의 시간을 갖기 때문이다. 비동기 복제의 경우 동의하기 위해서는 자신의 복제 본들이 어떤 사이트로 이동했는지에 관한 정보를 전단계의 에이전트로부터 받아야 하는데 워킹 타임이 끝난 후에도 이러한 정보들이 도달하지 못했다면 동의 과정이 시작되기 전에 이 정보들이 갱신되기를 기다려야 하기 때문이다. 반면 검사점 기법의 에이전트는 자신이 지나온 사이트에 관찰자를 남겨놓으며 그 에이전트들과 동의 과정을 가지면 되기 때문에 이동 시간 외에 메시지를 주고받는 시간만이 성능에 반영되게 됨으로 더 적은 동의 시간을 가는 것은 당연하다.

그림 3을 보면 복제 기법은 에이전트의 크기에 민감함을 볼 수 있다. 반면 비동기 복제 기법이나 검사점 기법의 경우에는 에이전트 크기에 상대적으로

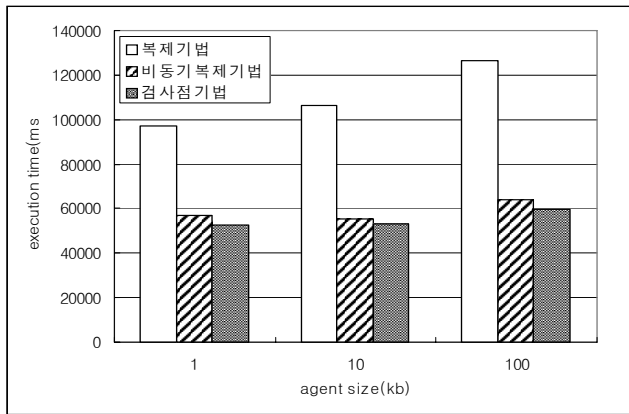


그림 3. 에이전트 크기에 따른 변화

로 덜 민감한 것을 볼 수 있다. 에이전트의 크기가 증가하면 전송시간도 같이 증가한다. 복제 기법을 사용하면 증가된 전송 시간이 에이전트 개수의 배수로 전체 성능에 반영될 것이다. 비동기 복제 기법이나 검사점 기법은 주 에이전트의 전송 시간 증가만이 성능에 영향을 미친다.

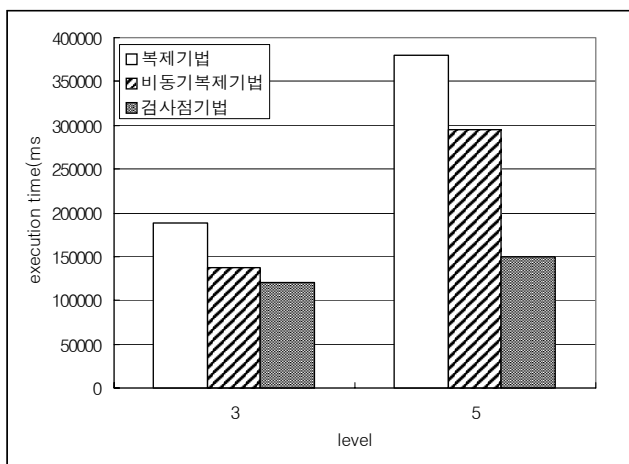


그림 4. 레벨에 따른 변화

그림 4에서 에이전트의 워킹 타임은 1000ms이다. 여기에서 레벨이란 복제기법의 경우 이동시 생성된 에이전트의 총 개수(복제본+주 에이전트)를 말하며 검사점 기법의 경우에는 관찰자 에이전트와 주에이전트를 합친 개수를 말한다. 복제 기법은 레벨이 증가할수록 성능 시간도 비례해서 증가한다. 그럴 수밖에 없는 것이 복제본의 생성 시간과 전송 시간이 그대로 성능에 반영되기 때문이다. 비동기 복제 기법에서는 레벨이 증가하면 메시지를 주고받는 시간 뿐만이 아니라 전 단계에서 복제 본들에 대한 정보가 넘어오는 시간이 늦어지기 때문에 성능이 저하

된다. 하지만 검사점 기법에서는 전단계로부터 관찰자의 위치에 대한 정보를 받아야 할 필요가 없기 때문에 더 나은 성능을 보이고 있다.

4. 결론

본 논문에서는 검사점을 기반으로 하는 결함 내성 기능을 갖춘 이동 에이전트 시스템을 구현하였다. 검사점 기법의 단점인 신뢰도를 개선키 위해, 실행 중인 에이전트 외에 여러 관찰자 에이전트를 두어, 결함 발생 시 이전에 설정된 검사점에서 실행 재개가 가능하도록 하였다. 이 기법은 복제 기법과 비슷한 신뢰성을 지니면서도 복제 기법의 단점이었던 오버헤드를 줄임으로 타 기법에 비해 탁월한 성능을 지니게 된다.

Acknowledgements

본 연구는 한국과학재단 목적기초연구 지원으로 수행되었음 (과제번호: R04-2002-000-20102-02003).

5. 참고문헌

- [1] S. Pleisch and A. Schiper, "FATOMAS-A Fault Tolerant Mobile Agent System Based on the Agent Dependent Approach," *Proc. of the Int'l Conf. on Dependable Systems and Networks*, pp. 215-224, 2001.
- [2] M. Strasser and K. Rothermel, "System Mechanism for Partial Rollback of Mobile Agent Execution," *Proc. of the 20th Int'l Conf. on Distributed Computing Systems*, 2000.
- [3] 강수석, 변일수, 박태순, "k-결함 허용 검사점 기법을 이용한 이동 에이전트 복구 기법에 관한 연구," 한국정보처리학회 제19회 춘계학술발표대회 발표논문집, 2003년 5월.
- [4] 변일수, 김수석, 박태순, "비동기 에이전트 복제를 이용한 효율적인 이동 에이전트 복구 기법," 한국정보과학회 제30회 춘계학술발표회 발표논문집, 2003년 4월.
- [5] G. Karjoth, D.B. Lange and M. Oshima, "A Security Model for Aglets," *IEEE Internet Computing*, 1997.