

자가치유 무선 센서 네트워크의 효율적인 메모리 관리 메커니즘

정동원^o 최창열 김성수
아주대학교
{windsage^o, clchoi, sskim}@ajou.ac.kr

An Efficient Memory Management Mechanism for Self-healing Wireless Sensor Network

Dongwon Jung^o, Changyeol Choi, Sungsoo Kim
Ajou University

요 약

무선 센서 네트워크에서는 센서로부터 유입되는 데이터의 정확성을 위하여, 결함이 발생한 센서 노드의 자가 치유는 필수적이다. 이를 위해, 롤백 기술을 사용하기 위한 글로벌 체크포인트 프로토콜들이 제시되었지만, 각각의 프로토콜은 메모리 효율과 성능에 있어서 상충 관계가 존재한다. 따라서, 본 논문은 성능 위주 글로벌 체크포인트 프로토콜의 메모리 효율을 개선하기 위해서, 플랫폼 기반의 주소 로그 기반 구조(Address Log Based Architecture, ALBA)를 제시한다. 이 구조는 롤백 동작에 필요한 메모리의 양을 줄이기 위해 간접 메모리 접근 방식과 원격 접근 방식을 지원한다.

1. 서 론

유비쿼터스 환경에서는 사용자의 인식 없이 현실 세계의 데이터를 얻기 위해서, 무선 센서 네트워크 개념이 도입되었다. 이 때, 결함이 발생한 센서 노드에 대해서 롤백 기술을 통한 자가 치유는 정확한 데이터 센싱에 커다란 도움을 줄 수 있다. 하지만, 로컬 시스템만을 고려한 롤백 기술은 무선 센서 네트워크에서 도미노 현상(domino effect)이라 불리는 시스템 차원의 결함을 발생시킨다 [1]. 이런 이유로, 네트워크 차원으로 이 기술을 활용하기 위해서 글로벌 체크포인트 프로토콜들이 제시되었지만, 무선 센서 네트워크가 가지는 제약조건들 때문에 기존의 프로토콜들은 성능과 메모리 효율 사이에 상충 관계가 있다. 센서의 짧은 송신 사거리로 인한 간접 라우팅을 사용하는 환경에서, 동기 노드(coordinate node)에서 동기화를 위해 동기 신호(beacon)를 전송함에도 불구하고 전파 지연(propagation delay)과 전송 지연(transmission delay)으로 외곽 노드의 동기 시간이 심한 오차를 갖게 된다. 이 오차를 고려하여 모든 노드의 행동을 일정 시간 동안 멈추고, 동시에 모든 노드들의 상태를 저장하는 동기 기반 프로토콜은 자칫 오랫동안 시간 동안의 유희한 상태로 인해 성능의 하락을 초래한다[2]. 이 문제를 피하기 위해서, 비동기식 프로토콜들이 제시되는데, 특징은 로컬 기반의 롤백 기술을 사용하되, 결함이 발생한 프로세스가 다른 프로세스들의 실행환경과 일치 상태로 만드는 기능

을 제공한다는 것이다. 하지만, 이 프로토콜들은 센서에서 동작 중인 프로세스 자체의 체크포인트 상태 외에도 다른 노드와 통신하기 위해 사용되는 모든 메시지까지 저장해야 하는 부담을 지닌다. 이러한 문제는, 센서의 제한적인 메모리 양을 고려해 볼 때 합당치 않다. 특히, 선 메시지 전송 후 저장 방식을 채택함으로써 마감시간을 만족시키기 위해 설계된 인과 로그 기반 프로토콜(causal log based protocol)의 경우, 메시지 송신 프로세스에서 모든 송신 메시지를 저장해야 한다는 단점을 지니며, 이와 같은 문제는 싱크 노드와 같이 여러 개의 하위 노드들과 통신하는 경우 더욱 심각해진다[3].

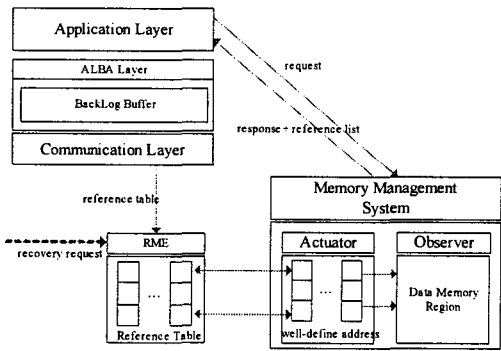
따라서 본 논문에서는, 무선 센서 네트워크에서 마감시간을 만족시키기 위한 인과 로그 기반 프로토콜의 송신측 메시지 로그(sender-based message log) 방식을 기반으로, 이 프로토콜이 가진 비효율적인 메모리 관리를 개선하기 위해서 주소 로그 기반 구조(Address Log Based Architecture, ALBA)를 제시한다. 이 구조는 롤백 동작을 위해 필요했던 메시지 전체 부분에 대한 보존을 메시지의 작은 부분으로 국한시킬 수 있으며, 이를 위해 간접 메모리 접근(indirect memory access)을 지원하도록 설계되었다. 더 나아가, 결함이 발생한 프로세스를 위해 복구에 필요한 데이터 전송으로 인한 불필요한 성능 저하를 막기 위해서 원격 관리 요소(remote management element, RME)를 센서에 탑재하였다. 이것은 센서의 메모리에 접근할 수 있는 임베디드 시스템으로, 자신의 CPU와 메모리를 이용하여 센서의 자원을 소모하지 않은 채 필요한 데이터를 결함이 발생한 프로세스에 전송할 수 있다[4]. 2절에서는 구체적인 ALBA의 구조

본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅 및 네트워크 원천기반기술개발 사업의 지원에 의한 것이다.

를 설명하며, 3절에서는 ALBA를 사용한 인과 로그 기반 기존의 프로토콜을 메모리 측면에서 성능을 측정한다. 마지막으로, 4절에서는 결론 및 향후 연구 과제를 제시한다.

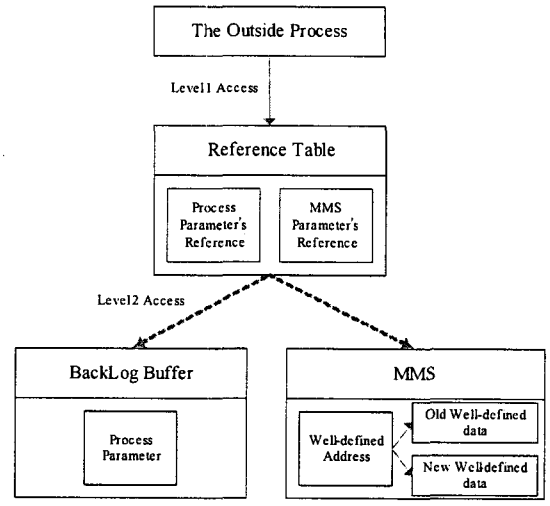
2. 주소 로그 기반 구조

ALBA는 그림 1처럼 크게 메모리 관리 시스템(Memory Management System, MMS), RME, 그리고 ALBA 계층으로 이루어져 있다. MMS는 프로세스가 사용하는 데이터 중 지속적인 관리(persistent management)가 필요한 데이터를 자신의 메모리에 저장한다. 그리고, 이 메모리와 해당 데이터를 얻기 위한 송신자 프로세스 사이의 요청 처리자(request handler)이면서 동시에 자신의 메모리의 데이터를 얻기 위한 요청에 대해서 응답(response)은 물론 이것의 주소에 대한 참조 목록(reference list)을 같이 넘겨주는 역할을 담당한다. ALBP 구조에서는 응답 메시지에 참여한 실제 데이터를 명시 데이터(well-defined data) 그리고 이 데이터에 대한 주소를 명시 주소(well-defined address)라고 정의한다. RME는 센서의 자원을 사용하지 않고, 자신만의 CPU와 메모리를 사용하여 센서의 메모리에 접근할 수 있는 임베디드 시스템이다. ALBA 계층은 MMS로부터 생성된 참조 목록을 통하여 외부의 프로세스가 RME를 이용해 접근 관리할 수 있는 일련의 기능을 제공한다.



(그림 1). 주소 로그 기반 구조

이 구조를 기본으로 하여 ALBA에서는 비동기 프로토콜들의 비효율적인 메모리 관리를 해결하기 위해 프로세스 사이의 통신에서 생성된 메시지 전체를 저장하는 것이 아니라 주소 기록 기반으로 하여 메시지의 작은 부분과 명시 주소만을 저장한다. 이 때, 센서의 프로세스나 ALBA 계층을 통한 직접 메모리 접근(direct memory access) 방식이 아닌, RME를 통한 간접 메모리 접근 채택함으로써 센서 메모리 안에 보존되고 있는 복수 메시지를 얻기 위한 일련의 행동들은 센서의 자원을 소모하지 않게 되고, 이와 같은 관리로 인한 전체 시스템 성능 저하는 발생



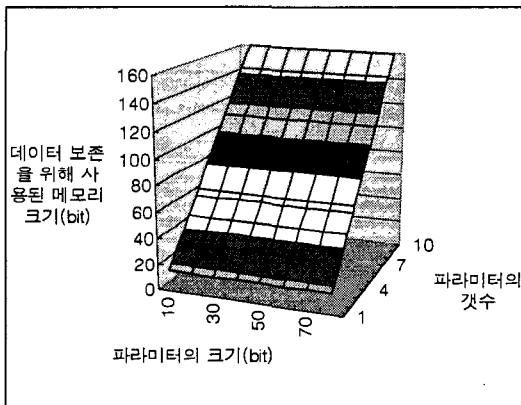
(그림 2). 2 단계 주소 기반 메모리 접근

하지 않는다. 이러한 간접 메모리 접근은 그림 2처럼 2 단계 주소 기반 메모리 접근 방식을 사용한다. 레벨 1 접근은 외부의 프로세스에서 참조 테이블(reference table)로 의 접근이다. 이 테이블은 하나의 메시지에 대한 모든 참조 목록들의 집합이다. 이것을 사용하는 이유는, 하나의 메시지가 MMS 메모리 또는 프로세스 메모리에서 추출된 파라미터들로 구성되어 있는 이상, ALBA 계층에서 프로세스에서부터 받은 정보들로는 모든 파라미터의 참조 목록을 보존시킬 수 없기 때문이다. 더군다나, 프로세스는 요청에 의해서 자신의 메모리 상태가 변할 수 있음에도 불구하고, 본래의 명시 데이터를 백업하기 위한 관리 도구를 제공하기 어렵다. 따라서, ALBA 계층에서는 프로세스 메모리로부터 얻어진 파라미터들을 백로그 버퍼에 저장하고 참조 목록을 생성함으로써 참조 테이블을 완성시킬 수 있고, 특별한 조치 없이도 이 백업 데이터들은 프로세스 메모리와 독립적으로 저장되기 때문에 지속성(persistent)을 유지할 수 있다. 한편, 명시 데이터나 이것을 참조하고 있는 명시 주소의 경우, MMS 메모리 상태 자체도 요청에 의해서 변할 수가 있기 때문에 잘못된 메모리 참조를 막기 위해 명시 데이터를 백업한 뒤 명시 주소 자체를 수정해야만 한다. 반면에, 명시 주소와 고정적으로 연결된 참조 목록은 수정될 필요가 없으므로 외부의 프로세스들은 이것을 이용하여 레벨 2 접근인 명시 주소로의 접근을 통해 적절한 명시 데이터를 얻을 수 있다.

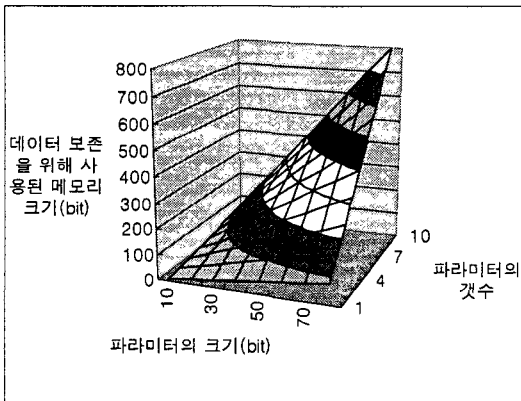
이 밖에도, MMS의 메모리를 수정할 우려가 있는 요청을 처리하기 위해 관찰자(observer) 컴포넌트와 작용자(actuator) 컴포넌트를 MMS 안에 배치했다. 관찰자는 감시(monitoring) 컴포넌트로, MMS가 요청을 받을 때마다 이것을 실행하기 전에 수정 요청(modification request) 인지를 판단하는 역할을 담당한다. 반면, 작용자는 이러한 수정 요청이 메모리에 영향을 주기 전에, 바뀌어질 데이터를 다른 곳으로 저장한 뒤 명시 주소의 값을 바꾸는 역할을 수행한다.

3. 시뮬레이션

이 절에서는 ALBA를 사용함으로써 절약할 수 있는 메모리의 양을 기존의 비동기식 프로토콜들과 비교 분석한다. 기본적으로 하나의 메시지는 n개의 파라미터들로 이루어져 있다고 가정한다. 그리고, 각각의 파라미터들을 MMS의 힙 메모리(heap memory)에 저장되는 레퍼런스 타입(reference type)이라고 가정하였을 때, 16 비트 임베디드 시스템을 탑재한 센서가 각각의 데이터들을 포인팅하기 위해서는 16 비트를 필요로 하게 된다. 따라서, ALBA를 사용할 경우 한 파라미터를 백업시키기 위해 실제 데이터의 크기가 아닌 16 비트를 필요로 한다. 그림 3과 4는 ALBA를 사용한 프로토콜과 기존의 전체 메시지 보존 방식의 프로토콜을, 평균 파라미터의 크기와 메시지를 구성하는 파라미터의 갯수 n 값에 변화를 주면서 데이터 보존을 위한 메모리의 양을 측정 한 것이다.



(그림 3). 주소 로그 기반 구조를 사용하였을 때의 메모리 사용량



(그림 4). 기존 프로토콜의 메모리 사용량

그림 3은 ALBA를 사용할 경우, 개개의 파라미터의 크기와 관계 없이, 파라미터의 갯수에 따라 메시지 보존을 위해 사용되는 메모리의 사용량이 점진적으로 증가하는 것을 보여주고 있다. 이와는 반대로, 그림 4는 ALBA를 사용하지 않을 경우, 개개의 파라미터의 크기가 작을 때는 ALBA를 사용했을 때보다 효율적이었지만, 이것이 파라미터의 갯수와 같이 증가할 때마다 메모리의 사용량이 급증하는 것을 보여준다. 결론적으로, 두 개의 그림을 비교할 때, ALBA를 사용한 프로토콜은 메시지의 크기가 클수록, 파라미터의 수가 작을수록 기존의 프로토콜보다 메모리 관리 측면에서 효율적인 것으로 나타났다. 더 나아가, 센서노드에서 센싱된 정보를 송신하기 전에 모든 파라미터를 하나의 객체로 만드는 병합과정(aggregation operation)을 수행한다면, 이 ALBA의 메모리 관리 효율은 보다 증가할 것이다.

4. 결론

본 논문에서는, 비동기식 방식의 글로벌 체크포인트 프로토콜이 가진 비효율적인 메모리 관리를 개선하기 위해 ALBA구조를 제시하였다. 이 구조는 인과 로그 기반 프로토콜을 기반으로 하고 있기 때문에, 무선 센서 네트워크에서 중요한 고려 사항인 마감시간을 맞추기 위한 빠른 응답시간 또한 만족할 수 있다.

향후에는 좀 더 정확한 성능 분석을 위하여 NS2로 임베디드 DB를 탑재한 싱크노드와 이 노드와 통신하는 하위 노드들로 구성된 센서 네트워크를 구축하는 한편, ALBA를 탑재한 인과 로그 기반 프로토콜과 기존 프로토콜들과 응답시간, 메모리 사용량, 그리고 CPU 소모량 측면에서 성능을 비교 측정할 것이다.

4. 참고 문헌

- [1] M. Elnozahy and L. Alvisi, "A Survey of Rollback-recovery Protocols in Message-passing System," ACM Computing Surveys, Vol. 34, pp. 375-409, Sep. 2002.
- [2] J. Cao, Y. Li, and M. Guo, "Process Migration MPI Applications based on Coordinated Checkpoint," Proceedings of International Conference on Parallel Distributed Systems, pp. 306-312, July 2005.
- [3] A. Bouteiller, et al., "Coordinated Checkpoint versus Message Log for Fault Tolerant MPI," Proceedings of IEEE International Conference on Cluster Computing, pp. 242-250, Dec. 2003.
- [4] F. Sultan, et al., "Nonintrusive Remote Healing Using Backdoors," Proceedings of Algorithms and Architectures for Self-Managing Systems, pp. 69-74, June 2003.