고장을 허용하는 분산공유메모리 시스템의 성능 향상 기법*

김영재°, 박소연, 맹승렬
한국과학기술원 전자전산학과 전산학전공
{yjkim°, sypark, maeng}@camars.kaist.ac.kr

Enhancing the Performance of Fault-Tolerant Software Distributed Shared Memory

Youngjae Kim°, Soyeon Park, Seung Ryoul Maeng
Division of Computer Science, Department of Electrical Engineering & Computer Science, KAIST

요 약

클러스터 시스템의 규모가 커짐에 따라 소프트웨어 분산공유메모리 시스템의 고장 발생 확률은 점차 증가하고 있다. 그래서 최근에는 소프트웨어 분산공유메모리 시스템에 고장 허용 기능을 추가하는 연구가 활발히 이루어지고 있다. 연구의 초점은 소프트웨어 분산공유메모리 시스템이 정상 실행을 하는 동안 고장 허용 기능을 추가로 지원하면서 발생하는 오버헤드를 줄이는데 있다. 본 논문은 고장 허용 기능을 위해 원격 로깅 기법을 사용하는 소프트웨어 분산공유메모리 시스템에서 로그를 이용함으로써 전체적인 성능 향상을 기대한다.

1. 서론

소프트웨어 분산공유메모리 시스템은 클러스터 상에 분산되어 있는 메모리들을 하나의 가상의 이미지로 사용자에게 보여 줌으로써 병렬 프로그래밍을 용이하게 한다. 그리고 최근의 고성능 프로세서와 고속의 네트워크 환경은 소프트웨어 분산공유메모리 시스템의 성능을 높이는데 큰 역할을 하고 있다. 이와 같은 프로그래밍의 용이성과 하드웨어의 발전은 거대한 클러스터 시스템을 작업 집중적이고 오랜 수행 시간을 요구하는 응용프로그램을 위해서 사용될 수 있게 한다. 그러나 시스템 규모가 커짐에 따라 시스템에서 고장이 발생할 가능성은 점차 증가하고 있다. 그래서 높은 가용성을 요구하는 환경에서 클러스터 시스템의 정상 수행 시간을 저하시키지 않고 확장성 있는 고장 허용 기능의 지원이 반드시 필요하다.

최근 몇 년간 소프트웨어 분산공유메모리 시스템에 고장 허용 기능을 지원하기 위한 방법으로 각 노드는 독립적인 체크포인팅(checkpointing)을 하면서 노드 간의 교환된 메시지들을 로깅(logging)하는 기법을 주로 사용하여 왔다 [1,2]. 이러한 기법은 고장이 발생한 노드가 롤백(rollback)을 수행하면서 공유 데이터로 인한 노드 간의 의존 관계로 다른 노드가 연속적인 롤백을 수행해야 하는 도미노 현상(domino effect)을 제거한다.

최근에 주요하게 연구되고 있는 홈 기반 소프트웨어 분산공유메모리 시스템 [3]은 완화된 일관성 유지 프로토콜을 기반으로 하여 노드간의 통신량을 줄이고 홈 노드의 개념을 도입하여 확장성을 높인다. 이러한 홈 기반 소프트웨어 분산공유메모리 시스템은 각각의 공유 메모리 페이지에 홈 노드를 할당하고 쓰기를 수행한 모든 노드들로부터 수정된 영역들을 수집하여 페이지를 갱신 한다. 홈 개념을 사용하기 때문에 노드에서 페이지 폴트가 발생하였을 때 최신의 페이지를 홈 노드로부터 한번의 메시지 교환을 통해서 가져올 수 있다. 하지만 해당 페이지의 필요한 영역만이 아닌 페이지 전체를 가져옴으로써 노드간의 불필요한 데이터를 교환으로 인한 오버헤드가 존재한다.

본 연구는 원격 로깅 기법을 사용하여 고장을 허용하는 소프트웨어 분산공유메모리 시스템을 기반으로 한다. 원격 로깅 기법은 빈번한 디스크 접근을 요구하지 않으며 사용자 계층 통신 프로토콜에서 제공하는 원격 메모리 쓰기를 최대한 활용하므로 정상 실행 중 오버헤드를 최소화한다.

본 논문은 앞에서 언급한 홈 기반 소프트웨어 분산공유메모리 시스템 상에서 노드간의 불필요한 데이터 교환을 최소화하기 위해 로그를 이용한 성능 향상 기법을 제안한다. 이 기법은 메시지 오버헤드를 감소시킴으로써 전체적인 성능 향상을 기대한다.

본 논문의 구성은 다음과 같다. 2절에서는 기반이 되는 시스템에 대해서 간략히 기술한다. 3절에서는 로그를 이용한 성능 향상 기법에 대해서 설명하고 4절에서는 구현된 시스템의 성능 결과 및 분석에 대해서 설명한다. 그리고 5절에서는 결론을 맺는다.

2. Fault Tolerant KDSM (FT-KDSM) 개요

FT-KDSM은 LINUX kernel version 2.2.15에서 구현되었으며 VIA(Virtual Interface Architecture) 표준에 의거하여 구현된 VI-GM [5]을 사용하여 통신한다. 캐

* 이 연구는 국가지정연구실 사업의 지원을 받는다.

쉬 일관성 유지를 위해 페이지 무효화 프로토콜 사용하고 HLRC(Home-based Lazy Release Consistency) 모델을 바탕으로 다중 쓰기(multiple writer)를 지원한다. 각 노드는 독립적인 체크포인팅을 수행을 가정하며 메모리 상태를 변화시키는 메시지를 받았을 때 원격 메모리에 로깅 하는 기법을 사용한다.

2.1 고장 발생 가정

FT-KDSM은 고장이 발생한 노드는 다른 노드의 상태에 영향을 미치지 않고 정지하는 고장-정지(fail- stop) 모델을 가정한다. 그리고 네트워크와 각 노드의 디스크는 안정적이라고 가정한다.

2.2 로깅 기법

FT-KDSM의 각 노드는 고장 시 복구를 위해 백업 노드를 하나씩 할당받는다. 메시지 송신 노드는 수신 노드의 백업노드에 동일한 메시지를 보내어 원격 메모리 로깅을 한다. 로깅을 위해 두 번의 메시지 전송이 필요하지만 FT-KDSM은 VI-GM에서 제공하는 원격 메모리 쓰기 기능을 최대한 활용함으로써 발생하는 오버헤드를 최소화 한다. 로깅되는 데이터는 페이지의 변경된 내용(diff)과 페이지 복사본을 무효화시키는 메시지(write notice)이다. 백업 노드에 저장된 로그는 상대 노드가 체크포인팅 될 때 비로소 메모리에서 삭제 가능하다.

고장이 발생한 노드는 고장 이전 상태와 동일하게 복구하기 위해서 독립적으로 체크포인팅 한 데이터와 주고받은 로그를 이용한다. 일단 고장이 발생한 노드는 마지막 체크포인팅 지점으로부터 수행을 다시 시작하며 이후에 받았던 메시지들은 백업 노드로부터 가져와서 적용함으로써 복구를 가능하게 한다.

FT-KDSM은 홈 노드에서 페이지 쓰기를 할 때 diff를 만들어 자신의 메모리에 로깅 해야 한다. 왜냐하면 노드에 고장이 생겨서 복구를 수행하는 동안 다른 노드가 페이지 홈 노드에게 이전의 특정 시점에 해당하는 페이지를 요구하면 적당한 페이지를 복원하여 서비스를 해 주어야 하기 때문이다.

3. 성능 향상 기법

홈 기반 소프트웨어 공유메모리 시스템에서 페이지 폴트가 발생한 노드는 홈 노드로부터 최신의 페이지를 얻는다. 이것은 메모리 일관성을 위해 실제로 갱신되어야 하는 영역이 페이지의 일부인 경우에도 언제나 페이지 전체를 전송 받아야 한다는 것을 의미한다. 하지만 원격 로깅 기법을 사용하는 FT-KDSM은 페이지의 홈 노드에서 로그를 이용하여 페이지 일관성을 저해하지 않으면서 페이지의 일부 영역만 서비스하는 것이 가능하다. 물론 페이지 요청 메시지가 도착할 때마다 적절한 메시지를 만들기 위해 로그를 검색하고 메시지를 만들어야 하는 프로세서 사이클의 오버헤드가 존재하긴 하지만 새롭게 제안하는 diff range 기법과 page blocking 기법은 이러한 오버헤드를 최소화한다.

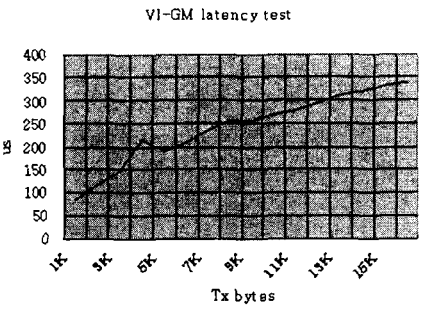

그림 1: VI-GM 한 방향 전송 지연시간 측정

3.1 메시지 크기에 따른 전송지연 시간

그림 1은 Pentium III 850MHz으로 구성된 두 개의 컴퓨터를 LANai 9.1 프로세서를 장착한 통신망으로 연결하여 메시지 크기에 따른 한 방향 전송 지연시간을 측정한 실험 결과를 보여 준다. 그래프는 메시지 크기가 증가함에 따라 전송 지연시간이 증가함을 보인다. 4KB 전송 시 평균 전송 지연시간은 215μs인 반면 2KB를 전송 시 118μs이다. 4KB를 전송할 때보다 2KB를 전송할 때 전송 지연시간은 약 두 배 빠르다.

그림에서 보듯이 홈 노드가 페이지 요청 메시지를 서비스하는데 필요한 페이지 영역의 크기가 4KB에 미치지 않는 경우 오히려 전체 페이지 전송은 시스템 성능에 효과적이지 못할 수 있다. 하지만 FT-KDSM에서는 페이지 요청에 대한 필요한 페이지 영역의 위치와 크기를 로그의 검색하여 알 수 있기 때문에 페이지 전체가 아닌 필요한 페이지 영역만 전송함으로써 메시지 오버헤드를 줄일 수 있다.

3.2 diff range 기법

페이지 요청에 대한 빠른 서비스를 하기 위해서 diff range 기법을 제안한다. 페이지 폴트가 발생하면 현재 가지고 있는 페이지의 복사본의 버전과 갱신되어야 하는 버전 정보를 홈 노드에게 전송한다. 홈 노드는 이 두 버전을 비교하여 갱신에 필요한 diff 메시지들을 로그에서 찾을 수 있다. diff 메시지는 갱신된 데이터의 위치 및 크기 정보가 포함하고 있으므로 홈 노드는 자신의 diff 로그를 검색하면서 필요한 diff 크기 및 위치 정보를 얻는다. 그리고 적절한 데이터 영역만 복사하여 요청 노드에게 전송한다. 이와 같이 홈이 받았던 diff 메시지를 그대로 전송하지 않고 필요한 영역을 계산함으로써 같은 페이지 영역의 중복 복사를 피할 수 있다.

3.3 서비스 기준

페이지 전송하는 것 보다 적절한 diff 영역을 전송하는 방법이 효과적이기 위해서는 로그로부터 필요한 diff 메

표 1: 벤치마크 프로그램과 입력 데이터 크기

ocean	258 X 258 grid
water	1728 mols, 5 steps
raytrace	cars
tsp	20 cities

시지를 찾고 필요한 페이지의 영역을 계산하는데 소비되는 오버헤드를 최소화시켜야 한다. 그래서 본 논문에서는 서비스 가능한 페이지 영역의 크기를 2KB로 설정하여 갱신에 필요한 페이지의 영역이 2KB을 넘게 되면 로그 검색을 그만 두고 페이지를 전송하도록 구현한다. 또한 필요한 페이지 영역의 크기가 작다 하더라도 복사본의 버전과 요청된 버전의 차이가 클 경우 해당 필요한 diff를 찾아서 영역을 계산하는 오버헤드가 크기 때문에 적절한 버전 차이를 두어 기준에 따라 적당한 페이지의 영역 또는 페이지를 서비스하도록 한다.

3.4 page blocking 기법

갱신에 필요한 페이지의 diff 영역이 지나치게 잘게 분할되어 있는 경우 적당한 메시지를 만들기 위해 메모리 복사가 빈번하게 요구된다. 이러한 메모리 복사 오버헤드를 피하기 위해 4KB의 페이지를 8B 크기의 블록으로 나누어 블록 단위로 갱신의 유무를 기록하고 복사하는 방식을 제안한다. 이 기법은 필요한 페이지의 diff를 하나의 메시지로 만드는데 소비되는 프로세서 사이클의 비용이 줄일 뿐 아니라 구현상 diff range 기법을 쉽게 적용할 수 있다는 장점을 가진다.

4. 성능 평가

본 논문에서는 8대의 Pentium III 850MHz 컴퓨터로 구성된 클러스터 상에서 성능 향상을 위해 제안한 기법을 적용한 FT-KDSM의 성능을 측정한다. 각 노드는 LANai 9.1 프로세서를 장착한 Myrinet 통신망으로 연결되고 벤치마크 프로그램으로 SPLASH2의 ocean, water, raytrace와 Rice 대학의 tsp를 상용한다. 표1은 각 응용프로그램에서의 입력 데이터 크기를 보인다.

그림 2는 성능 향상 기법을 적용한 FT-KDSM와 기본적인 FT-KDSM의 정규화한 실행시간을 비교한다. 각 응용 프로그램에서 왼쪽 그래프는 성능 향상 기법이 적용되기 전의 결과를 나타내며 오른쪽 그래프는 성능 향상이 적용된 후의 결과를 나타낸다. ocean을 제외한 나머지 프로그램에서 페이지 대신 페이지의 필요한 영역만 전송하였을 때 좋은 성능을 보여준다. raytrace와 tsp의 경우 약 8-12%의 성능 향상을 보인다. 이것은 페이지를 서비스 받는데 소비되는 페이지 폴트 시간이 감소했기 때문이다. 반면 ocean의 경우 성능 향상 기법을 적용한 FT-KDSM의 성능이 오히려 더 나빠지는데 이것은 응용 프로그램 특성 때문이다. ocean은 페이지 홈 노드에서 페이지 요청에 대한 서비스를 하기 위해 적절한 diff 영

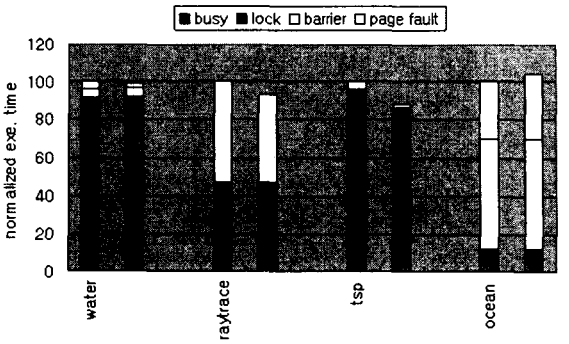

그림 2: 성능 향상 기법 적용 여부에 따른 성능 비교

역을 로그에서 검색하지만 페이지 일관성을 위해 필요한 영역이 페이지의 대부분이며 복사본의 버전과 요청한 버전의 차이가 대부분 너무 커서 실제로 서비스 기준 내에서 서비스 할 수 있는 경우의 수가 상당히 작기 때문이다.

5. 결론

본 논문에서는 VIA 통신 계층상에서 동작하는 분산공유메모리 시스템에 고장 허용 기능을 추가한 FT-KDSM의 성능 향상 기법을 제안한다. 구현의 최적화를 위해서 diff range기법과 page blocking기법을 제안하였다. 본 시스템은 고장 허용을 위해 각 노드가 유지하는 로그를 최대한 활용하여 몇몇 응용 벤치마크에서 성능향상을 보였다. 로그를 검색하고 diff 영역을 계산함에 따른 오버헤드는 전송되는 메시지의 크기의 감소로 인한 성능 향상을 크게 저해하지 않았다.

참고문헌

[1] A. Kongmunvattana and N.F.Tzeng. "Coherence-Centric Logging and Recovery for Home-Based Software Distributed Shared Memory", In *Proc. of the 1999 Int'l Conf. on Parallel Processing (ICPP'99)*, pages 274-281, September 1999

[2] F. Sultan, T. Nguyen, and L.Iftode. "Scalable Fault-Tolerant Distributed Shared Memory", *Proc. SC 2000 High Performance Networking ana Computing Conf.*, November 2000

[3] Y. Zou, L. Iftode, and K. Li. "Performance Evaluation of Two Home-Based Lazy Release Consistency Protocols for Shared Virtual Memory Systems", In *Proc. of the 2nd USENIX Symp. on Operating Systems Design and Implementation (OSDI)*, pages 75-88, October 1996

[4] 박소연, 김영재, 이상권, 맹승렬. "VIA (Virtual Interface Architecture)를 기반으로 하는 소프트웨어 분산공유메모리 시스템의 설계 및 구현", 제 29 회 한국정보과학회 춘계 학술발표논문집(A), April 2002

[5] http://www.myricom.com/scs/index.html