

병렬 프로그램의 동적 분석을 위한 효율적인 감시 필터링 기술

박세원[○], 하옥균^{*}

[○]경상대학교 정보과학과

^{*}경상대학교 공학연구원

e-mail: swpark3179@nate.com[○], jassmin@gnu.ac.kr^{*}

An Efficient Filtered Monitoring Technique for Dynamic Analysis of Parallel Programs

Se-Won Park[○] and Ok-Kyoon Ha^{*}

[○]Department of Informatics, Gyeongsang National University

^{*}Engineering Research Institute, Gyeongsang National University

● 요약 ●

본 논문에서는 멀티 스레드 기반 병렬 프로그램의 동적 분석을 통한 자료경합 탐지를 위해 효율적인 감시 필터링 기술을 제시한다. 제시하는 감시 필터링 기술은 자료경합과 같은 동시성 오류를 탐지하기 위해 프로그램의 수행 중에 발생하는 공유 메모리에 대한 스레드들의 접근 분석 시에 동일한 코드영역을 단순 반복하는 스레드들을 감시대상에서 제외시킴으로써 동적 분석을 위한 추가적인 오버헤드를 최소화한다. 제시하는 필터링 기술을 수행 중 자료경합 탐지 도구에 적용한 실험적 평가를 통해 개선된 수행시간 오버헤드의 제공이 가능함을 보인다.

키워드: Multithread programs, dynamic analysis, monitoring, data race detection, runtime overheads, filtering

I. Introduction

멀티코어 및 멀티프로세서 시스템의 사용이 보편화되면서 멀티스레드 프로그램과 같은 병렬 프로그램의 필요성이 강조되고 있다. 그러나 멀티스레드 기반 병렬 프로그램의 신뢰성은 동시성 스레드 간 미묘한 상호작용과 잠재적으로 초래할 수 있는 자료경합[1]과 같은 동시성 버그 때문에 여전히 어려운 일이다.

자료경합을 탐지[2]하기 위한 동적 분석기술은 추적된 정보를 분석 및 재 수행시켜 탐지하거나, 프로그램 수행 중에 자료경합을 탐지한다. 그러나 프로그램 수행의 감시와 충돌하는 모든 메모리 동작의 분석을 위해 발생하는 10~100배의 추가적인 오버헤드의 발생은 동적 자료경합 탐지기술의 가장 큰 단점이다.

II. Background

효율적인 자료경합 탐지를 위해 본 연구팀에서는 다수의 필터링 기술을 제시하였다. 먼저, 동일한 스레드 세그먼트 내의 하나의 쓰기 접근은 병행하여 수행되는 스레드 세그먼트의 쓰기읽기 접근과 자료경합을 발생 시키는 점에 착안하여, 동일한 공유 메모리 영역으로의 반복적인 접근에 대해 각 스레드 세그먼트들의 각 접근 유형별로 첫 번째 이벤트만을 감시하고 나머지는 무시하는 필터링 방법[3]을

개발하였다.

다음으로, 감시를 수행할 프로그램의 실행 바이너리의 이미지 수준, 섹션 수준, 명령어 수준에서 자료경합 분석과 상관없는 각각의 감시제외 대상을 제외하는 계층별 필터링 기법[4]을 개발하였고, 이를 통해 동적 자료경합 탐지의 평균 실행 오버헤드를 약 45% 이상 줄인다. 그러나 규모가 큰 병렬 프로그램의 경우 동적 자료경합 탐지를 위한 추가적인 오버헤드가 프로그램의 병렬 구간 반복횟수에 비례하여 증가하기 때문에 효율성 측면에서 여전히 개선의 여지가 있다.

III. The Proposed Filtering Technique

프로그램에서 동일한 코드의 반복적 수행을 위해 일반적으로 반복문을 사용하며, 이러한 반복 영역의 병렬 수행은 자료경합 탐지를 위해서는 불필요한 감시를 동반한다. 따라서 본 논문은 병렬 루프 수행의 동적 감시를 위해 오직 두 개의 병렬 스레드만을 고려하는 감시 필터링 기술을 제시한다.

자료경합 탐지를 위한 동적 분석기술은 프로그램의 수행동안 공유 메모리에 대한 접근을 수집하기 위한 명령어들의 감시를 기반으로 한다. 제시하는 감시 필터링 기법은 명령어를 읽어 들인 후 감시 코드를 삽입하기 위한 프로세스를 재배치한다.

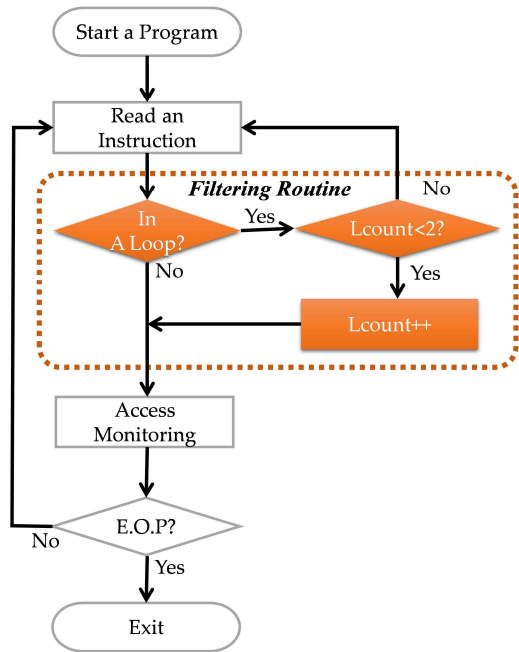


Fig. 1. The process of monitoring programs with proposed filtering technique

그림 1은 제시하는 감시 필터링 기법을 사용하는 프로그램 수행 감시 프로세스를 보인다. 그림에서와 같이 공유 메모리에 대한 접근 명령어를 병렬 반복구간에 포함되는지를 분석하고, 동일한 명령에 대한 다중 스레드의 단순 반복일 경우 Lcount 식별자를 통해 최대 2개의 스레드만을 감시한다. 이를 통해 불필요한 감시를 제거함으로써 자료경합 탐지를 위한 동적 분석의 추가적인 오버헤드를 획기적으로 줄일 수 있다.

IV. Evaluation

제시하는 감시 필터링 기술을 동적 자료경합 탐지도구에 적용하기 위하여 대표적인 동적 이진코드 삽입 프레임워크인 PIN을 사용하여 구현하였다. 구현된 도구는 병렬 반복문의 분석을 위해 ‘for 문’, ‘while 문’, ‘do while’을 대상으로 식별하였다. 구현된 감시 필터링 기술의 효율성을 합성 프로그램을 이용하여 자료경합 탐지 시 발생하는 시간적 오버헤드를 실험적으로 비교하였다. 표 1은 실험된 결과의 요약을 제시한다. 실험결과 제시된 감시 필터링 기법을 적용한 자료경합 탐지를 통해 평균 64%의 시간적인 성능향상을 얻을 수 있었다.

Table 1. The measured results of runtime overheads

| Programs | without Filtering | with Filtering | Rate |
|-----------|-------------------|----------------|------|
| Single | 80x | 30x | 62% |
| Serialize | 79x | 26x | 66% |

V. Conclusion

본 논문에서는 동시에 수행되는 반복영역에 대해 처음 수행되는 두 개의 스레드만 감시하는 효과적인 감시 필터링을 제시하였다. 제시된 감시 필터링은 프로그램 내의 반복문 반복횟수에 상관없이 일정한 오버헤드를 가지며, 병행성 루프가 있는 프로그램에서의 자료경합 탐지 오버헤드를 60%이상 감소시킬 수 있음을 보였다.

References

- [1] Ha, O.-K., "Case Study of Dynamic Detectors for Data Races," In proc. of International Conference on Electronic Engineering and Computer Science (EECS 2013), IERI Procedia, Beijing, China, Vol. 4, pp. 174-180, 2013.
- [2] Ha, O.-K., Jun, Y.-K., "An Efficient Algorithm for On-the-Fly Data Race Detection Using an Epoch-Based Technique", International Journal of Scientific Programming, Article No. 13, 2015.
- [3] Ha, O.-K., Kuh, I.-B., G. M. Tchamgoue, Jun, Y.-K., "On-the-fly Detection of Data Races in OpenMP Programs," In Proc. of the 2012 Workshop on Parallel and Distributed Systems: Testing, Analysis, and Debugging (PADTAD'2012), pp. 1-10, ACM, Minneapolis, USA, July 2012.
- [4] Ha, O.-K., Jun, Y.-K., "Effective Monitoring Memory Operations for Dynamic Race Detection through Hierarchical Filtering Method," International Journal of Multimedia and Ubiquitous Engineering, pp. 199-208, 2014.