

# OpenMP 디렉티브 병렬프로그램에서의 동적 자료경합 탐지를 위한 감시 필터링 기술의 실험적 비교

조아라<sup>0</sup>, 하옥균\*

<sup>0</sup>경상대학교 항공우주 및 소프트웨어공학전공

\*경상대학교 정보과학과

e-mail: realclout@naver.com<sup>0</sup>, jassmin@gnu.ac.kr\*

## An Empirical Comparison of Monitoring Filtering Techniques for Dynamic Data Race Detection in Parallel Programs with OpenMP Directives

Ahra Cho<sup>0</sup>, Ok-Kyoon Ha\*

<sup>0</sup>Department of Aerospace and Software Engineering, Gyeongsang National University

\*Department of Informatics, Gyeongsang National University

### ● 요약 ●

다중 스레드 기반 병렬 프로그램에서의 자료경합 탐지는 동시에 수행되는 스레드 간의 비결정적인 상호작용 때문에 탐지하기 어려운 것으로 잘 알려져 있다. 동적 분석기술을 사용하여 자료경합을 탐지할 경우 프로그램 수행의 감시와 충돌하는 모든 메모리 연산의 분석을 위해 추가적인 오버헤드가 발생한다는 단점이 있다. 이러한 동적 분석의 추가적인 오버헤드를 줄이는 방법으로 감시 필터링 기술이 소개되고 있으며, 본 논문에서는 동적 자료경합 탐지를 위한 감시 필터링 기술 중 OpenMP 디렉티브 병렬 프로그램에 적용 가능한 두 기술을 대상으로 실용성과 효율성을 실험적으로 비교한다.

**키워드:** OpenMP programs, dynamic analysis, data race detection, monitoring filtering techniques

### I. Introduction

OpenMP는 공유 메모리 환경에서의 병렬 프로그래밍 API로 컴파일러 디렉티브를 이용하여 순차 프로그램을 병렬 프로그램으로 쉽게 변환하는 방법을 제공한다[1]. 그러나 다중 스레드 기반 병렬 프로그램은 동시에 수행되는 스레드 간의 상호작용으로 인해 상호배제(deadlocks)나 자료경합(data races)[2]과 같은 동시성 오류(concurrency errors)가 발생할 수 있다. 대표적인 동시성 오류인 자료경합은 프로그램의 의도하지 않은 비결정적(non-deterministic) 수행 결과를 초래할 수 있기 때문에 소프트웨어의 테스트 및 디버깅을 위해서 반드시 탐지되어야 한다.

자료경합을 탐지[3]하기 위한 기술 중 동적 분석(dynamic analysis) 기법은 프로그램의 수행 동안 스레드의 발생과 동기화 및 메모리 접근 정보 등의 감시를 기반으로 자료경합을 분석한다. 동적 분석기반의 자료경합 탐지 기술은 프로그램의 추적, 재수행, 감시를 통해 수행 중에 발생한 실제 자료경합을 보고하기 때문에 높은 신뢰성을 제공하지만, 프로그램 수행의 감시와 충돌하는 모든 메모리 동작을 분석하기 때문에 추가적인 수행시간 오버헤드가 발생한다.

### II. Filtering Techniques

동적 자료경합 탐지의 추가적인 오버헤드를 줄이는 방법으로 감시 필터링 기술이 도입되고 있다. 감시 필터링 기술은 자료경합 탐지 시 분석 대상 중에서 불필요한 감시를 선택적으로 배제함으로써 동적 분석의 과도한 수행시간 오버헤드를 줄인다. 이러한 필터링 기술들 중 OpenMP와 같은 암시적(implicit) 병렬 스레드 수행 모델에 적용 가능한 대표적인 필터링 기술에는 중복 이벤트 필터링(REF: Redundant Event Filtering)[4]과 루프영역 필터링(LRF: Loop Region Filtering)[5]이 있다. 이들 필터링 기법을 적용함으로써 동적 자료경합 탐지 시 발생하는 시간적 오버헤드를 50% 이상 감소시킬 수 있다. 그러나 두 기법은 프로그램에서 공유변수의 개수, 록킹의 사용 유무, 병렬 루프의 규모 등과 같은 수행 특성과 방식에 따라 다른 효율성을 보인다.

### III. Design of Experimentation

본 논문에서는 REF와 LRF 기술을 OpenMP 벤치마크를 적용하여 프로그램 특성 별로 동적 자료경합 탐지 효율성을 실험적으로 비교한다

다. 실험은 각 벤치마크 프로그램을 대상으로 필터링을 적용하지 않은 경우, REF를 적용한 경우, LRF를 적용한 경우, REF와 LRF 모두 적용한 경우 총 네 가지 경우에서의 수행시간 오버헤드를 비교한다. 표 1은 비교 실험을 위해 사용한 벤치마크 프로그램들과 각 응용 프로그램의 특성을 나타낸다.

Applications	Lines	Accesses		Locks	Loop Count
		Read	Write		
FFT6	542	2285K	15399K	1	38K
MD	266	23584K	9451K	0	5632
Mandelbrot	144	144537K	8	0	1024
Pi	83	20000K	8	0	5000K

Table 1. The features of OpenMP benchmarks

REF와 LRF의 구현과 실험을 위한 환경은 2.6 커널 리눅스 운영체제의 64bit Intel Xeon Quad-core CPU 2개와 48GB 메모리를 사용하는 시스템을 사용하였다. 또한 OpenMP를 지원하는 gcc 4.1.2 컴파일러와 경상대 연구팀이 개발한 동적 자료경합 탐지 도구[4]를 설치하였다. 구현된 REF와 LRF기법은 PIN 소프트웨어 프레임워크를 기반으로 탐지도구에 동적으로 적용하였고, 프로그램의 원(original) 수행에 비해 발생한 수행시간(runtime) 오버헤드의 비율을 측정하였다.

#### IV. Results

실험을 통해 측정된 수행시간 오버헤드는 그림 1과 같다. 그림 1의 결과에서 보는 것과 같이 REF를 적용한 자료경합 탐지는 기존 수행시간 오버헤드에 비해 평균 92.2%의 오버헤드로 자료경합을 탐지하고, LRF를 적용한 자료경합 탐지는 기존 수행시간 오버헤드의 평균 14.2%의 오버헤드만으로 자료경합을 탐지한다. REF와 LRF 모두 적용한 자료경합 탐지는 기존 수행시간 오버헤드의 평균 8.7% 오버헤드만으로 자료경합을 탐지하는 결과를 보여 실용적이다.

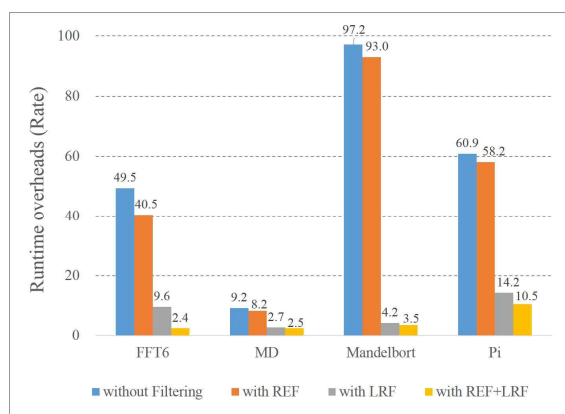


Fig. 1. The measured results of runtime overheads

#### V. Conclusion

본 논문에서는 동적 자료경합 탐지 시 발생하는 추가적인 수행시간 오버헤드를 불필요한 감시 대상의 배제를 통해 줄이는 대표적인 두 가지 감시 필터링 기술의 효율성을 OpenMP 벤치마크를 사용하여 실험적으로 비교하였다. 실험 결과 두 가지 필터링 기술을 모두 결합한 자료경합 탐지에서 수행시간 오버헤드가 평균 10% 이내로 감소하여 가장 실용적임을 보였다.

#### References

- [1] Chapman, B., Jost, G., and Pas, R. V., "Using OpenMP: Portable Shared Memory Parallel Programming," The MIT Press, 2007.
- [2] Ha, Ok-Kyoon, "Case Study of Dynamic Detectors for Data Races," In proc. of International Conference on Elect. Eng. and Comp. Science (EECS 2013), IERI Procedia, Beijing, China, Vol. 4, pp. 174-180, 2013.
- [3] 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.
- [4] Ha, O.-K., Kuh, I.-B., G. M. Tchamgoue, Jun, Y.-K., "On-the-fly Detection of Data Races in OpenMP Programs," In Proceeding of the 2012 Workshop on Parallel and Distributed Systems: Testing, Analysis, and Debugging (PADTAD'2012), pp. 1-10, ACM, Minneapolis, USA, July 2012.
- [5] Park, S.-W., Ha, O.-K., "An Efficient Filtered Monitoring Technique for Dynamic Analysis of Parallel Programs," In Proc. of the Korean Society of Computer Information Winter Conference, 24(1): 9-10, KSCI, Kwangju, Korea, January, 2016.