

HPC를 활용한 지휘무장통제체계 SW 성능향상 연구

백치선^o

^o한화시스템 해양연구소 SW팀(해양)

e-mail: cs214.baek@hanwha.com^o

A Study of Performance Improvement of CFCS SW Using HPC

Chi-Sun Baek^o

^oSWTeam(Naval), Naval R&D Center, Hanwha Systems Co.,LTD.

● 요약 ●

본 논문에서는 지휘무장통제체계(이하 CFCS) 소프트웨어의 성능 향상 기법으로 고성능 컴퓨팅(이하 HPC) 시스템 활용 기법을 제안한다. 이 기법으로 본 논문에서는 HPC 분야인 멀티코어 프로세서를 활용하는 방법을 제안한다. 복잡한 반복연산을 하는 작업이 많은 CFCS의 특정 SW모듈에 대해 멀티코어 프로세싱 아키텍처를 이용한 병렬처리를 적용하여 기존 순차처리 대비 작업 실행시간을 단축함으로써 작업 응답시간을 상당히 줄일 수 있다. 본 논문에서는 CFCS 시험 환경의 일부 특정 SW모듈 상에서 기존의 순차처리 방식으로 수행한 연산 결과와 다중 처리 프로그래밍 API인 OpenMP를 적용하여 수행한 연산 결과를 비교하여 CFCS에서의 멀티코어 프로세싱이 체계 전반의 성능 향상 면에서 효율적으로 사용될 수 있음을 보인다.

키워드: 지휘무장통제체계(CFCS), 고성능컴퓨팅(HPC), 멀티코어프로세싱(multi-core processing), 병렬처리(parallel processing), OpenMP

I. Introduction

지난 10년간 컴퓨터 세계에 역동적인 변화가 조용히 일어났다. 단일 코어 프로세서를 기반으로 하던 시스템은 해를 거듭하여 어느덧 멀티코어(multi-core), 매니코어(many-core) 프로세서 시스템으로 진보했다. 그 덕분에 오늘날 다양한 분야의 과학자, 연구원, 엔지니어, 개발자 등은 효율 좋은 매니코어 프로세서와 병렬 시스템을 활용해 더 빠르게 결과를 받아볼 수 있게 되었다. 반면, 제품의 교체주기가 매우 길고 성능보다는 안전성, 신뢰성이 우선시되는 방산분야에서는 아직까지도 십수 년 혹은 그 이상 오래된 하드웨어(이하 HW)들과 그곳에 탑재된 소프트웨어(이하 SW)를 운용하고 있다. 구식 HW와 SW의 낮은 효율성과 제품 단종 등의 비싼 유지비용은 높은 국방비 지출을 초래하는 여러 원인 중 하나이다.

본 논문에서는 멀티코어 프로세싱 아키텍처 기반의 병렬처리 API 중 하나인 OpenMP를 CFCS 시험 환경에 적용하여 기존 시스템 대비 성능 향상 척도를 구함으로써 HPC가 CFCS에 성능과 비용측면에서 활용가치가 높음을 보인다.

II. Preliminaries

1. Related works

1.1 국내 동향

대부분의 국내 무기체계 SW는 수십 년 전 개발된 오래된 외산 시스템을 기반으로 개발되었기 때문에 구조적, 기술적으로 선진국에 비해 뒤쳐져 있다고 볼 수 있다[1].

1.2 국외 동향

미국, 독일 등의 선진국들은 십수 년 전부터 방위산업 분야에 HPC 도입을 연구해온 것을 여러 학술지료를 통해 알 수 있고, 공개하지 않지만 현재 이미 상당 부분 도입되어 있을 것으로 예측 된다[2][3][4].

2. Background

2.1 OpenMP

OpenMP(Open Multi-Processing)는 공유 메모리 다중 처리 프로

그래픽 API로, C, C++, 포트란 언어와, 유닉스 및 마이크로소프트 윈도 플랫폼을 비롯한 여러 플랫폼을 지원한다.



Fig. 1. OpenMP

III. Performance Test of CFCS SW with OpenMP based on HPC

Legacy 코드와 OpenMP를 적용한 병렬화 코드의 성능 측정을 위해 구성된 시험 환경은 다음 Table 1과 같다.

Table 1. Test System Environment

Item	Value
CPU Clock	3.3 GHz
Thread Core	4
Memory Size	4 GB

시험을 위해 CFCS의 SW모듈 중 하나인 합대함유도무기체계 정보처리 모듈의 배경표적 처리 루틴의 Legacy 코드와 해당 코드에 OpenMP를 적용하여 병렬화한 코드를 준비하고 각각의 실행시간을 측정하였다.

IV. Evaluation

Legacy 코드와 OpenMP를 적용한 병렬화 코드의 성능 측정 시험의 결과는 아래 Fig. 2와 같다.

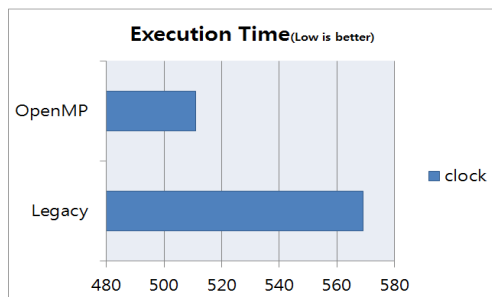


Fig. 2. Execution-Time Comparison

최대 4개 Thread까지 병렬로 처리할 수 있는 본 시험 환경에서는 OpenMP를 통한 병렬화 적용 시 약 10%의 성능 향상을 보였다.

V. Conclusions

본 논문에서는 CFCS SW의 성능 향상 기법으로 HPC 환경 상에서 OpenMP를 통한 SW 병렬화 기법을 제안하였다. 또한 Legacy 코드와

OpenMP를 적용한 병렬화 코드의 성능 평가를 통해 병렬화를 통한 CFCS SW의 성능 향상을 입증 하였다. 더 나아가 SW 구조 설계 단계부터 병렬화를 고려하여 병렬화 구간을 증가 시킨다면 성능향상 척도를 더욱 높일 수 있을 것으로 기대할 수 있다.

추후 연구 과제로 CFCS 상에서 OpenCL, TBB 등 또 다른 다중 처리 프로그래밍 API의 성능 평가를 통하여 CFCS에 가장 효율적인 방법을 찾는 연구와, 매니코어(many-core) 프로세서 상에서의 성능 향상 척도를 구하는 연구를 진행할 예정이다.

REFERENCES

- [1] S.N. Lee, and T.W. Kim, "Weapon System SW Status and Development Direction," Communications of the Korean Institute of Information Scientists and Engineers, Vol. 29, No. 10, pp. 16-22, 2011.
- [2] J.J. Dongarra, et al., eds. "High performance computing: technology, methods and applications. Vol. 10," Elsevier, 1995.
- [3] S.J. Schraml, K.D. Kimsey, and J.A. Clarke, "High-performance computing applications for survivability-lethality technologies," Computing in Science & Engineering, Vol. 4, No. 2, pp. 16-21, 2002.
- [4] J. Kepner, "HPC productivity: An overarching view," International Journal of High Performance Computing Applications, Vol 18, No. 4, pp. 393-397, 2004.