

## Cell Broadband Engine을 위한 소스 대 소스 변환\*

전보성<sup>o</sup> 이준 이재진

서울대학교 컴퓨터공학부

[posung@aces.snu.ac.kr](mailto:posung@aces.snu.ac.kr), [jun@aces.snu.ac.kr](mailto:jun@aces.snu.ac.kr), [jlee@cse.snu.ac.kr](mailto:jlee@cse.snu.ac.kr)

### Source to Source Translation for Cell Broadband Engine

Posung Chun<sup>o</sup> Jun Lee Jaejin Lee

School of Computer Science and Engineering, Seoul National University

#### 1. 서 론

프로세서의 성능을 향상시키기 위한 단일 프로세서 방법과 기술들이 그 한계를 다함에 따라 다중 코어로의 피할 수 없는 변화에 직면하고 있다. 또한 프로세서와 메모리 간의 성능 차이로 발생하는 메모리 장벽 문제는 여전히 컴퓨터 분야에서 해결하기 어려운 문제로 남아있다. 이러한 문제를 배경으로 소프트웨어에 의해 명시적으로 메모리 계층구조를 관리하고 범용 코어와 계산 특화된 코어를 함께 내장하고 있는 프로세서인 Cell Broadband Engine[1, 2]이 출현하였다.

Cell BE는 하나의 범용으로 쓰이는 64bit PowerPC 코어와 부동소수점 연산에 특화된 8개의 SPE (synergistic processor element)를 가지고 있다. 각각의 SPE는 지역 저장소(Local Store)를 가지고 있으며, 외부 메인 메모리로부터 지역 저장소에 코드와 데이터를 복사하여 프로그램을 실행한다. 이러한 아키텍처를 사용하여 높은 성능을 달성하기 위해서는 프로그래머는 병렬을 고려하여 프로그래밍 할 뿐 아니라, 외부 메인 메모리와 지역 저장 장소의 데이터 이동 또한 직접적으로 프로그램에 명시 하여야 하는 어려움이 따른다. 이러한 어려움을 줄이기 위해서, 상대적으로 쉽게 사용할 수 있는 상위수준의 병렬 프로그래밍 모델을 사용하여 프로그래밍 하는 것이 필요하다.

본 연구에서는 OpenMP[3]로 작성된 C 소스 코드를 입력 받고 Cell BE를 위한 런타임 라이브러리를 타깃으로 하는 C 소스를 생성하는 컴파일러를 작성하였다. 작성된 컴파일러가 올바른 코드를 생성함을 확인하기 위해 OpenMP 프로그래밍 모델로 작성된 벤치마크 프로그램을 입력으로 코드를 자동 변환하여 정확성을 검증하였다.

#### 2. 본 론

OpenMP는 병렬 공유 메모리 프로세서 아키텍처에 적합한 프로그래밍 모델로서, 프로그래밍하기 쉬운 편이며, 현재 가장 많이 사용되는 프로그래밍 모델 중의 하나이다. OpenMP는 순차적으로 작성한 프로그램 코드에 컴파일러 지시어(Directives)를 추가함으로써 병렬 프로그래밍을 할 수 있다. 반면, 다른 병렬 프로그래밍 모델인 MPI (Message Passing Interface)의 경우 전역 공유 주소공간을 갖지 않고, 프로세서간의 데이터의 이동을 명시해야 하므로 프로그래밍이 어려운 편이다.

Open64[4] 컴파일러는 OpenMP 2.0으로 작성된 소스 코드를 언어에 독립적인 중간 표현단계로 변환 가능하다. 컴파일러 내부 중간 표현은 소스 수준의 표현력을 가지고 있다. 따라서 소스 수준의 변경을 가능할 뿐 아니라, 변경된 중간 표현을 다시 C 소스로 출력이 가능하다. 이러한 구조는 본 연구에서 목적으로 하는 런타임 시스템을 위한 소스 대 소스 변환에 적합하다. 본 컴파일러는 Open64 4.1의 구성요

\* 본 연구는 정보통신부 및 정보통신연구진흥원의 'IT신성장동력핵심기술개발사업[2006-S-040-01, Flash Memory 기반 임베디드 멀티미디어 소프트웨어 기술개발]'과 '한국학술진흥재단의 BK21 사업'의 일환으로 수행되었음.

소 중에서 전반부(Front-end)와 VHO(Very High Optimizer) 단계는 재사용하고, MP lowering과 Whirl2c 단계를 수정하여 작성하였다. MP lowering 단계는 OpenMP 지시어가 포함된 중간 표현 단계를 해석하여 런타임 시스템 호출로 변환하는 과정을 수행하며, Whirl2c 단계는 PPE와 SPE를 위한 C코드를 출력한다.

Cell BE 아키텍처에서는 프로그램의 컨트롤 집중적인 부분은 PPE로, 계산에 집중적이며 병렬적으로 수행해야 할 부분은 SPE로 나누어 프로그래밍 하는 것이 유리하다. 본 컴파일러는 PPE와 SPE를 구분하지 않은 하나의 OpenMP로 작성된 소스 코드를 입력 받아 지시어에 의해 표시된 계산 집중적인 병렬 구역은 SPE에서 실행되고, 나머지 컨트롤에 집중적인 부분은 PPE에서 실행 될 수 있도록 코드를 자동으로 분할할 수 있게 하였다. 또한 병렬 프로그램을 어렵게 하는 요소 중에 하나인 데이터 일관성 문제는 전역 공유 변수에 대한 접근(Read/Write)을 데이터 일관성을 보장하는 런타임 시스템 라이브러리 인터페이스로 자동 변경한다. 또한, 동기화(Synchronization)을 위해 그에 상응하는 라이브러리 인터페이스를 추가한다. 마지막으로, 병렬 루프는 여러 SPE에서 병렬로 나누어 수행할 수 있도록 루프를 스케줄링한다.

본 컴파일러에 의해 생성된 C 소스코드로 Cell BE를 위한 실행파일을 생성하기 위해서는 다음과 같은 과정이 필요하다. 컴파일러가 생성하는 PPE와 SPE용 C 코드는 PPE 컴파일러와 SPE 컴파일러에 의해 각각 컴파일한다. 먼저 생성된 SPE 목적 파일은 SPE 런타임 라이브러리와 함께 SPE 링커에 의해 링크되어 SPE 실행파일을 생성한다. PPE 목적 파일은 PPE 런타임 라이브러리와 SPE 임베더(Embedder)에 의해 생성된 목적파일과 링크되어 런타임 시스템을 위한 실행 가능한 파일을 생성한다. 이렇게 생성된 실행 바이너리는 Cell BE가 탑재되어 있는 Sony PlayStation3 이나, Cell Blade 서버에서 실행 가능하며 각각의 환경에는 Fedora Core Linux 또는 Redhat Enterprise Linux, 그리고 Cell SDK가 필요하다.

본 컴파일러가 일관성을 보장하는 런타임 시스템을 대상으로 올바른 코드를 생성하는지 확인하기 위하여, C와 OpenMP를 통해 작성된 벤치마크 프로그램을 본 컴파일러로 C 소스 코드를 생성하였다. 검증에는 jacobi, md, STREAM, SWIM의 4가지 벤치마크 프로그램을 사용하였으며, 생성된 소스 코드를 SPE와 PPE 컴파일러로 바이너리를 생성하여, Sony Playstation3에서 올바른 결과를 출력함을 확인할 수 있었다.

### 3. 결 론

이 논문에서 OpenMP 프로그래밍 모델을 입력 받아 런타임 시스템을 타겟으로 하는 컴파일러의 구현에 대하여 설명하고 있다. 또한, 실험을 통해 컴파일러에 의해 생성된 코드의 정확성을 보여주었다.

이 컴파일러는 가장 널리 사용되고, 비교적 쉽게 병렬화 할 수 있는 OpenMP 프로그래밍 모델을 대상으로 함으로써, Cell SDK를 사용한 저 수준의 프로그래밍 방법에 비해 프로그래머에게 보다 많은 편의를 제공할 수 있었다. 또한, 본 컴파일러는 런타임 시스템을 타겟으로 하고 있으므로, 플랫폼에 독립적인 C 코드를 생성할 수 있다. 런타임 시스템이 Cell과 비슷한 구조의 이기종의 프로세서에 적용 될 수 있다면 컴파일러가 생성한 C 코드는 해당 플랫폼에 맞게 컴파일되어 실행 바이너리를 생성할 수 있기 때문이다.

### 참고문헌

- [1] B. Flachs et. al. A Streaming Processing Unit for a CELL Processor. *IEEE International Solid-State Circuits Conference(ISSCC)*, February 2005
- [2] Michael Gschwind. Chip Multiprocessing and the cell broadband engine. *Proceeding of the 3rd Conference on Computing Frontiers(CF)*, pages 1-8, May 2006.
- [3] The OpenMP specification for parallel programming. <http://www.openmp.org>
- [4] Open64, The Open Research Compiler. <http://www.open64.net>