

# IoT-Cloud 융합 가상기계 시스템을 위한 Ahead-of-Time 컴파일 시스템의 설계

최찬휘\*, 손윤식\*\*, 이양선\*

\*서경대학교 컴퓨터공학과

\*\*동국대학교 컴퓨터공학과

e-mail:sftblw@skuniv.ac.kr

## Design of Ahead-of-Time Compilation System for IoT-Cloud Fusion Virtual Machine System

Chanwhi Choi\*, Yunsik Son\*\*, Yangsun Lee\*

\*Dept of Computer Engineering, SeoKyeong University

\*\*Dept of Computer Science and Engineering, Dongguk University

### 요 약

사물인터넷 기술의 사용이 증가하고 있으나 장치 및 플랫폼의 종류가 다양하여 한 번 구현한 응용 프로그램을 재사용하기 어렵다. 사물인터넷 장치에서 가상기계를 사용하여 이러한 문제점을 해결할 수 있지만 가상기계의 응용 프로그램 실행 속도는 네이티브 코드에 비해 속도가 느려 가상기계의 실행 속도를 개선할 필요가 있다.

AoT 컴파일은 바이트코드를 네이티브 코드로 사전에 컴파일하여 가상기계의 실행 속도를 향상시키는 기법이다. 본 논문에서는 IoT-Cloud 융합 가상기계 시스템을 위한 AoT 컴파일 시스템을 설계한다. 설계한 시스템은 사물인터넷과 클라우드의 융합 환경에 적합하며, 바이트코드 중 일부만을 네이티브 코드로 컴파일하므로 네이티브 코드 로드와 관련한 가상기계의 메모리 부담이 적다.

## 1. 서론

사물인터넷 기술의 사용이 증가하고 있다. 그에 비해 사물인터넷 장치와 플랫폼의 종류가 다양하여 한 번 개발한 응용 프로그램을 재사용하기 어렵다. 사물인터넷 장치에서 가상기계를 사용하여 이러한 문제를 해결할 수 있지만 가상기계의 응용 프로그램 실행 속도는 네이티브 코드에 비해 속도가 느리다. 따라서 가상기계의 응용 프로그램 실행 속도를 개선할 필요가 있다.

Ahead-of-Time(AoT) 컴파일[1]은 바이트코드를 사전에 네이티브 코드로 컴파일, 실행하여 가상기계의 실행 속도를 향상시키는 기법이다. 본 논문에서는 IoT-Cloud 융합 가상기계 시스템[3]을 위한 AoT 컴파일 시스템을 설계한다. 설계한 시스템은 클라우드에서 네이티브 코드를 다운로드할 수 있어 사물인터넷과 클라우드의 융합 환경에 적합하다. 또한 AoT 컴파일러가 바이트코드 중 일부만을 네이티브 코드로 컴파일하므로 네이티브 코드 로드와 관련한 가상기계의 메모리 부담이 적다.

## 2. 관련연구

### 2.1 Ahead-of-Time 컴파일러

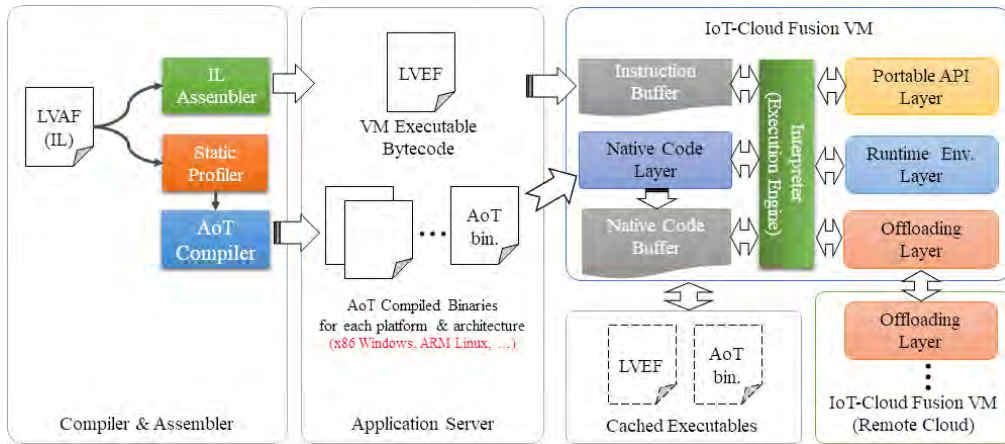
“이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No.2016R1A2B4008392)”

가상기계는 기본적으로 인터프리트 방식으로 실행되므로 네이티브 코드에 비해 실행 속도가 느리다. 바이트코드를 네이티브 코드로 컴파일, 실행하여 가상기계의 실행 속도를 향상시킬 수 있다. 이러한 가상기계의 실행 속도 향상 기법으로 Just-In-Time(JIT) 컴파일[1,2], AoT 컴파일 이 있다.

AoT 컴파일은 바이트코드를 가상기계 실행 전에 미리 네이티브 코드로 컴파일하여 가상기계의 응용 프로그램 실행 속도를 향상시키는 기법이다. AoT 컴파일은 실행시간 컴파일 비용이 없고, 가상기계가 컴파일러를 탑재하지 않으므로 성능이 낮고 가용 자원이 적은 사물인터넷 환경의 가상기계에 적합하다.

### 2.2 IoT-Cloud 융합 가상기계 시스템

IoT-Cloud 융합 가상기계 시스템은 저성능 사물인터넷 장치에 클라우드의 높은 연산력을 부여하는 시스템이다. 본 시스템은 응용 프로그램 중 복잡한 연산을 수행하는 부분의 실행을 클라우드 서버에 위임하여 단말장치의 자원 소모를 줄이고 높은 성능을 부여할 수 있는 오프로딩 기법을 기반으로 한다. 따라서 본 시스템에서는 사물인터넷 장치의 가상기계와 클라우드 시스템과의 융합 및 연계를 가정한다.



(그림 1) Ahead-of-Time 컴파일 시스템의 전체 구성도

### 3. Ahead-of-Time 컴파일 시스템의 설계

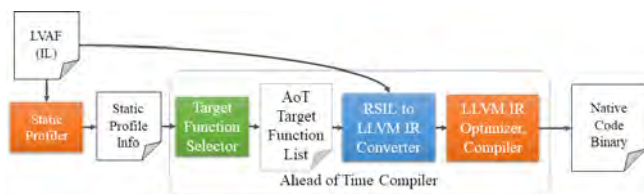
본 AoT 컴파일 시스템은 클라우드에서 응용 프로그램을 다운로드할 수 있도록 구성하여 IoT-Cloud 융합 가상기계 시스템의 환경인 사물인터넷과 클라우드의 융합 환경에 적합하다. 또한 AoT 컴파일러는 바이트코드 중 일부분을 네이티브 코드로 컴파일하므로 가상기계의 메모리 부담이 적다.

그림 1은 AoT 컴파일 시스템의 전체 구성도이다. AoT 컴파일 시스템은 AoT 컴파일러, 애플리케이션 서버, 가상기계의 네이티브 코드 레이어로 구성된다. 어셈블러는 중간언어 코드를 바이트코드로 변환하고, AoT 컴파일러는 중간언어 코드를 네이티브 코드로 컴파일한다. 컴파일된 바이트코드와 네이티브 코드는 애플리케이션 서버에 보관된다. 가상기계는 애플리케이션 서버 혹은 로컬 파일 시스템으로부터 바이트코드와 네이티브 코드를 로드하여 실행한다.

#### 3.2 AoT 컴파일러

바이트코드는 대체로 네이티브 코드에 비해 메모리 사용량이 적다. 바이트코드 전체를 네이티브 코드로 컴파일하면 가상기계의 메모리 사용량이 증가하여 사물인터넷 장치용 가상기계에는 부담이 될 수 있다.

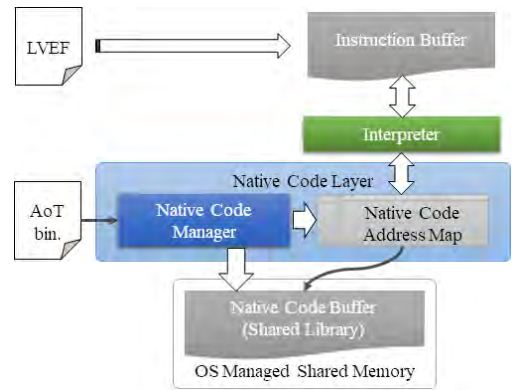
응용 프로그램은 대부분의 실행 시간을 전체 코드 중 일부분을 실행하는데 할애한다[4]. 따라서 AoT 컴파일러가 바이트코드 프로그램 중 오랜 시간 실행되는 부분만을 네이티브 코드로 컴파일하면 가상기계의 메모리 부담을 감소하면서도 가상기계의 실행 속도를 향상시킬 수 있다.



(그림 2) Ahead-of-Time 컴파일러의 구조도

본 시스템의 AoT 컴파일러는 정적 프로파일 정보를 기준으로 응용 프로그램의 함수 중 일부분을 컴파일하여 가상기계의 네이티브 코드 로드 때 따른 메모리 부담을 경감한다. AoT 컴파일러는 정적 프로파일러[5]를 통해 계산된 함수별 복잡도를 기준으로 AoT 컴파일 대상 함수를 선정한다. 선정된 함수들을 대상으로 RSIL to LLVM IR 변환기[6]는 중간언어 프로그램을 LLVM IR로 변환한다. LLVM IR로 변환된 프로그램은 LLVM IR 최적화기, 컴파일러에 의해 네이티브 코드로 컴파일된다.

#### 3.2 가상기계의 네이티브 코드 로드



(그림 3) 네이티브 코드 레이어의 구조도

컴파일된 네이티브 코드는 가상기계의 실행 시 네이티브 레이어에 의해 로드 및 관리된다. 네이티브 코드 매니저는 네이티브 코드 버퍼에 네이티브 코드를 로드하고, 메타 정보 및 함수별 네이티브 코드 로드 여부를 네이티브 코드 주소 맵에 로드한다. 인터프리터는 함수 호출 명령어의 실행 시 네이티브 코드 주소 맵을 확인하여 네이티브 코드가 로드되어 있는 경우 바이트코드 대신 로드된 네이티브 코드 함수를 호출한다.

### 4. 결론 및 향후 연구

본 논문에서는 IoT-Cloud 융합 가상기계 시스템을 위

한 AoT 컴파일 시스템을 설계하였다. 설계한 AoT 컴파일 시스템은 클라우드에서 응용 프로그램을 다운로드할 수 있어 사물인터넷과 클라우드의 융합 환경에 적합하다. 또한 바이트코드의 함수 중 복잡도가 높은 일부만 네이티브 코드로 컴파일하므로 네이티브 코드 로드에 의한 메모리 부담이 적다.

본 설계를 기반으로 AoT 컴파일 시스템을 구현할 것이다. 이를 위해 AoT 컴파일러의 각 구성요소, 최적화 방법, 가상기계의 로드된 네이티브 코드 메모리 관리 등을 위한 추가 연구가 필요하다.

### 참고문헌

- [1] Bucciarelli Thiemo, "Just-In-Time Compilation.", Institute for Software Engineering and Programming Languages, 2016.
- [2] Aycock John, "A brief history of just-in-time.", ACM Computing Surveys (CSUR) 35.2, pp.97-113, 2003.
- [3] Yunsik Son, Yangsun Lee, "Offloading Method for Efficient Use of Local Computational Resources in Mobile Location-Based Services Using Clouds," Mobile Information Systems, vol.2017, 9 pages, 2017. Netherlands Hindawi Publishing Corp.
- [4] Knuth Donald E., "An empirical study of FORTRAN programs.", Software: Practice and experience 1.2, 105-133, 1971.
- [5] 서동현, 저성능 IoT 디바이스의 효율적인 오프로딩을 위한 정적 프로파일러에 대한 연구, 서경대학교 공학석사 학위논문, 2017.
- [6] 조재현, 최찬휘, 손윤식, 이양선, "IoT 가상기계의 중간 코드 검증을 위한 RSIL to LLVM IR 변환기의 설계 및 구현", 한국정보처리학회 논문지, Vol.24, No.1, pp.1135-1137, 2017.