가상의 네트워크 프로세서 환경에서 프로그램 테스트를 위한 가상머신

홍순호*, 곽동규*, 고방원* 유재우*
*숭실대학교 컴퓨터학부
e-mail : ian@ssu.ac.kr

Virtual Machine for Program Testing on the Virtual Network Processor Environment

Soonho Hong*, Donggyu Kwak*, BangWon Ko*, Chae-Woo Yoo*
*Dept. of Computer Science & Engineering, SoongSil University

요 약

최근 인터넷 사용자 증가와 네트워크를 기반의 응용 프로그램이 다양하게 개발되고 있다. 또한 스마트 폰과 테블릿 PC 의 대중화로 누구나 쉽게 인터넷을 통해 정보검색 서비스를 이용할 수 있다. 따라서 갈수록 증가하는 패킷에 대한 제어와 이동, 삭제 등과 같은 처리를 빠르게 하기 위해 네트워크 프로세서(Network Processor)가 개발되었다. 네트워크 프로세서는 패킷 제어와 이동, 삭제를 수행하는데 최적화되어 있다. 하지만 네트워크 프로세서를 개발한 회사마다 교차개발환경 툴과 개발언어가 서로 다르기 때문에 소스코드 재사용 및 확장이 어렵다. 또한 네트워크 프로세서에서 동작하는 프로그램을 테스트 하기 위해 하드웨어 장비가 필요하고 네트워크 프로세서에 종속적인 개발환경과 언어를 배우는 것은 프로그래머에게 큰 부담을 준다. 본 논문에서는 네트워크 프로세서에 최적화된 기능을 언어 레벨에서 정의한 eFlowC 언어를 사용하고 범용 컴퓨터에서 테스트 및 실행을 할 수 있는 가상머신을 제안한다. 그리고 가상머신 중간언어를 사용하여 가상머신이 설치된 범용 컴퓨터에서 소스코드 재사용 및 확장을 가능하게 한다. 따라서 범용 컴퓨터에서 프로그램 테스트를 통해 신뢰성 높은 프로그램을 작성할 수 있다.

1. 서론

최근 인터넷 사용자가 계속해서 증가하고 네트워크 기반의 응용 프로그램이 다양하게 개발됨에 따라 네트워크 트래픽이 증가하고 있다. 게다가 스마트 폰과 테블릿 PC 의 대중화로 누구나 쉽게 무선인터넷을 통해 정보검색 서비스를 쉽게 이용할 수 있게 됐다. 게다가 무선인터넷은 시간과 공간의 제약을 받지 않기 때문에 인터넷을 통해 언제, 어디서나 무선인터넷을 통해 자료를 검색하거나 다운 받을 수 있게 되었다. 또한 정보통신 기기와 하드웨어 발전으로 대용량의 정보처리가 가능해져 사용자들은 동영상 멀티미디어 등으로 제작된 이해하기 쉬운 정보를 원하게 되었다[1]. 이러한 정보와 소통의 변화는 인터넷의 트래픽 양을 급속도로 증가 시켰다. 따라서 갈수록 증가되고 대용량화 되는 데이터에서 유해정보 차단과 해킹을 막기 위해 트래픽을 검사할 수 있는 빠른 방법이 요구되고 있다.

기존에는 네트워크 상에서 패킷을 조작, 삭제, 이동과 같은 기능을 수행하기 위해 일반적으로 주문형반도체(Application Specific Integrated Circuit)를 기반으로 프로그램이 개발되었다[2]. 하지만 이러한 개발 환경은 많은 비용을 요구하고, 빠른 시장의 요구 변화를 따라가지 못하고 있다. 따라서 이와 같은 문제를 해결하기 위해 Intel 과 Cisco 외 여러 회사에서 범용으로 사용할 수 있는 네트워크 프로세서(Network Processor)[3]를 개발했다. 네트워크 프로세서란 네트워크에서 패킷 처리를 위해 최적화된 마이크로 프로세서를 지칭하고 패킷 헤더 파싱, 패턴 매칭, 비트 단위 조작, 테이블 룩업, 패킷 순서 관리, 패킷 수정 그리고 데이터 이동과 같은 최적화된 기능을 제공한다[4].

네트워크 프로세서는 네트워크 상에서 패킷을 이동, 조작, 삭제를 수행하는데 최적화 되어 있다. 그러나 네트워크 프로세서를 개발한 Intel 이나 Cisco 에서 제공하는 교차개발환경과 개발언어, 라이브러리는 서로 다르다. 그러므로 서로 다른 개발환경에서 네트워크 프로세서에서 동작하는 프로그램을 개발할 경우 소스코드 재사용 및 확장이 어렵다. 소스코드 재사용 및 확장이 쉽고 공통적으로 사용할 수 있는 개발언어 및 교차개발환경이 요구된다.

본 논문은 네트워크 프로세서에 최적화된 기능을 언어 레벨에서 정의하고 프로그램을 작성하기 위한 수단으로 eFlowC 언어를 사용한다. 그리고 하드웨어 기반으로된 네트워크 프로세서에서 동작하는 응용 프로그램 테스트 및 개발을 할 수 있는 가상머신(ETRI Vritual Machine)을 제안한다. 가상머신을 사용함으로써 네트워크 프로세서에서 동작하는 프로그램을 범용 컴퓨터에서 테스트 및 실행을

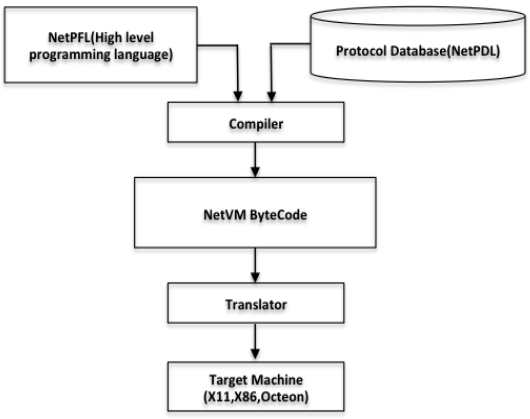
할 수 환경을 제공한다. 또한 서로 다른 개발환경에서 프로그램 소스코드 재사용 및 확장을 가능하게 하기 위해 가상머신 중간언어(ETRI Virtual Machine Intermediate Language)를 제시한다. 가상머신 중간언어는 eFlowC 언어로 작성된 프로그램을 eFlowC 컴파일러를 통해 생성하며 가상머신이 설치된 범용 컴퓨터에서 소스코드 재사용이 가능하다. 결과적으로 네트워크 프로세서 구조와 같은 환경에서 프로그램 테스트 및 실행을 적절하게 수행함으로써 신뢰성 높은 프로그램을 작성할 수 있다.

본 논문의 구성은 2 장에서 관련연구 NetVM 과 인텔 네트워크 프로세서인 IXP1200 알아본다. 그리고 eFlowC 언어에 대해 알아보고 3 장에서 가상머신시스템의 구조를 설명한다. 4 장에서는 가상머신 테스트 및 실행 결과를 보여주고, 마지막으로 5 장에서는 결론 및 향후 연구과제에 대해서 기술한다.

2. 관련연구

2-1 NetVM

NetVM(NetWork Virtual Machine)은 패킷 핸들링 응용 프로그램 실행에 최적화된 가상 네트워크 프로세서이다[5]. NetVM 은 패킷 필터링, 패킷 카운팅, 문자열 매칭 수행을 위한 통합된 추상 레이어를 제공하고 다양한 네트워크 응용 프로그램에서 수행된다[6].



(그림 1) NetVM 구조

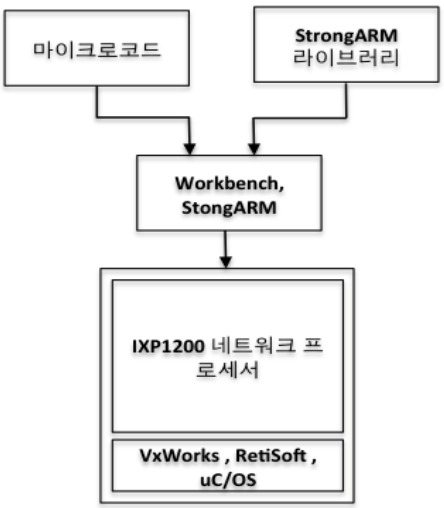
NetVM 은 [그림 1]과 같이 NetPFL(Network Packet Filtering Language)와 NetPDL(Network Protocol Description Language)를 사용하여 프로그램을 작성한다. 그리고 작성된 프로그램을 컴파일하여 NetVM 바이트코드를 생성한다. NetVM 바이트코드는 자바가상머신(Java Virtual Machine)과 같은 구조로 구성되었기 때문에 코드 재사용 및 확장에 유용하고 운영체제에 독립적인 성질을 가진다. 그러나 소스코드 재사용과 확장을 목적으로 설계된 NetVM 바이트코드는 NetVM 번역기에(NetVM Translator) 의해 타겟머신인 X86, Octeon, X11[7]의 네이티브코드로 변환된다. NetVM 에서 동작하는 프로그램을 테스트 및 실행하기 위해서 NetVM 의 타겟머신이 필요하고 NetVM 바이트코드를 네이티브코드로 변환시키기 위해서는 타겟머신에 종속적인 변환기(NetVM Translator) 가 필요하다[8].

eVM(ETRI Virtual Machine)과 NetVM 의 차이는 타겟머신인 하드웨어 장비 없이 프로그램을 테스트 할 수 있는지 여부이다. eFlowC 컴파일러가 설치된 범용 컴퓨터에서 NetVM 이 제공하는 네트워크 프로세서에서 특화된 기능을 범용 컴퓨터에서 실행이 가능하도록 eVM 이 설계되었다. NetVM 은 추상레이어 레벨에서 프로그램을 실행시킬 수 있는 인터프리터가 없기 때문에 타겟머신인 하드웨어 장비가 있어야만 프로그램을 실행할 수 있다.

NetVM 단점은 NetVM 을 이용하여 개발할 때 프로그램의 신뢰성 및 안전성을 위해 적절하게 테스트할 수 있는 영역이 존재하지 않는다. 프로그램의 신뢰성 향상시키기 위한 일환으로 프로그램 테스트를 수행할 수 있는 개발환경이 요구된다.

2-2 Intel Network Processor IXP1200

네트워크 프로세서(Network Processor)는 패킷 헤더 파싱, 패턴 매칭, 비트 단위 조작, 테이블 룩업, 순서 관리, 수정 그리고 데이터 이동과 같은 기능을 수행한다. 또한 빠른 속도로 특정 패킷 또는 전체 패킷을 검사, 조작, 추출 또는 삭제 기능을 제공한다.



(그림 2) IPX1200 구조

[그림 2]는 Intel 에서 개발한 IXP1200[9] 네트워크 프로세서이다. IXP1200 네트워크 프로세서에서 동작하는 프로그램을 테스트 및 실행하기 위해 [그림 2]와 같이 Workbench 나 StorngARM 과 같은 교차개발환경을 사용한다.

마이크로코드와 StrongARM 라이브러리를 이용하여 프로그램을 작성하고 네트워크 프로세서인 IXP1200 에서 동작하는 프로그램을 작성할 수 있다. 그러나 Intel 에서 제공하는 IXP1200 네트워크 프로세서에서 응용 프로그램을 작성하고 테스트 및 실행을 하려면 세 가지 조건이 요구된다. 첫 번째는 네트워크 프로세서를 이용하여 개발을 하기 위해서는 WindRiver 에서 제공하는 VxWorks, RetiSoft 또는 uC/OS 운영체체가 필요하다. 그러나 IPX1200 은 범용으로 사용하는 운영체제(Windows, Mac OS, Linux)가 아니기 때문에 범용 컴퓨터에서 실행하기 어려운 구조로 되어 있다. 두 번째는 네트워크 프로세서를 개발한 회사마다 제공되는 교차개발환경 툴과 개발언어가 다르다는 점이다. 따라서 늘 새로운 개발환경과 언어를 배워야 하는 번거로움이 있다.

IXP1200 네트워크 프로세서는 교차개발환경 툴인 Workbench 와 StrongARM 를 사용 해야 하고 Strong ARM 언어와 마이크로코드에 대한 이해가 필요하다. 세 번째는 프로그램 테스트 및 실행을 위해서 IXP1200 장비와 운영체제가 필요하다. 따라서 프로그램 테스트 및 실행을 위해 반드시 하드웨어 자원이 필요하다. 컴퓨팅 자원 부족으로 인해 테스트를 적절하게 수행하지 못하면 프로그램의 신뢰도가 저하될 수 있다.

본 연구에서는 IXP1200 네트워크 프로세스에서 프로그램을 테스트할 수 없는 단점을 보완한 가상머신을 제안하고 범용 컴퓨터에서 가상머신을 이용하여 테스트할 수 있는 환경을 제공함으로써 보다 높은 신뢰성 있는 프로그램을 개발할 수 있다.

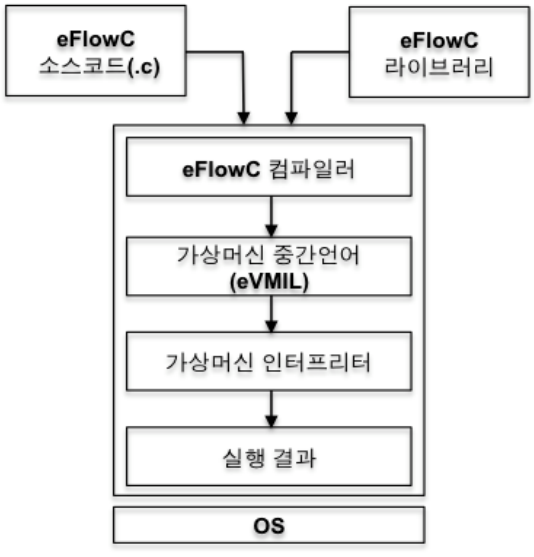
2-3 eFlowC

eFlowC 언어는 네트워크 프로그래밍을 쉽게 하기 위해 한국전자통신연구원(ETRI)에서 개발한 고급 언어이다.

eFlowC 언어의 장점은 네트워크에 필요한 최소한 기능를 유지하기 위해 불필요한 연산을 제거 하였다. 예를 들면 연산시간을 많이 요구하는 소수점 연산과 프로그래머가 실수할 수 있는 포인터 연산을 제거하였다. 또한 eFlowC 언어는 기존의 C 언어의 문법과 유사하게 설계하여 C 언어 개발 경험이 있는 개발자라면 쉽게 문법을 익힐 수 있도록 설계된 언어이다. 또한 eFlowC 언어는 네트워크 프로그램을 쉽게 할 수 있도록 고안된 고급 언어이기 때문에 패킷 헤더 파싱, 패턴 매칭, 비트 단위 조작, 테이블 룩업, 순서관리, 수정 그리고 데이터 이동을 수행하는데 최적화된 언어이다.

3 장. 가상머신 구조

고급 언어인 eFlowC 를 사용하여 응용 프로그램 테스트 및 실행을 위한 가상머신(ETRI Virtual Machine) 시스템을 구축하였다. 그리고 가상머신의 중간언어(ETRI Virtual Machine Intermediate Language)를 사용하여 코드 재사용 및 확장이 가능하도록 한다.



(그림 3) 가상머신 구조

[그림 3]과 같이 네트워크 프로세서의 특화된 기능은 패킷 헤더 파싱, 패턴 매칭, 비트 단위 조작 등이 있으며 고급 언어인 eFlowC 로 정의 된다. 네트워크 프로세서의 최적화된 기능을 언어 레벨에서 정의하고 프로그램을 작성을 위해 eFlowC 언어를 사용한다. eFlowC 언어는 C 언어와 유사하기 때문에 C 언어 관련 개발 경험이 있는 개발자라면 쉽게 배울 수 있다. 또한 기능과 성격이 비슷한 문법을 구조체로 정의함으로써 문법을 쉽게 이해하고 배울 수 있는 장점이 있다. eFlowC 언어로 작성된 응용 프로그램을 eFlowC 컴파일러로 컴파일하고, 그 결과로 가상머신 중간코드(ETRI Virtual Machine Intermediate Language)를 생성한다.

가상머신 중간코드는 어셈블리어 언어와 유사한 기계어 코드를 생성한다. 가상머신 기계어코드와 유사한 것으로 현재 범용 컴퓨터에서 사용되고 있는 자바 가상 머신(Java Virtual Machine)[10]이 있다. 가상머신 중간언어는 자바 바이트코드 구조와 같은 성격을 지니고 있다. 따라서 가상머신 중간코드는 소스코드 재사용성 및 확장이 가능함을 보여준다. 또한 OS 독립적인 성실을 가진다. 그리고 가상머신 중간코드는 프로그래머에 의해 재구성 및 확장이 가능하여 응용 프로그램의 유연성을 제공한다.

가상머신 중간코드는 가상머신에 의해서 구문 검사가 수행된 후 가상머신의 인터프리터에 의해 실행된다. 따라서 가상머신을 이용하면 컴퓨팅자원 부족 문제를 극복할 수 있다. 또한 네트워크 프로세서를 이용하여 개발할 때와 반대로 복잡한 구조를 사용하지 않고 쉽게 접근함으로써 시간과 노력을 줄여주는 효과를 기대할 수 있다. 결과적으로 가상머신을 통해 프로그램 테스트 및 실행을 함으로써 응용 프로그램의 신뢰도를 향상시킬 수 있다.

4 장. 가상머신 중간언어

가상머신 중간언어는 고급언어인 eFlowC 와 가상머신인 (ETRI Virtual Machine) 사이에 위치하며 소스코드 재사용과 확장이 가능하다.

<표 1> eVMIL 명령어 종류

목록	목록 설명	명령어	명령어 설명
초기화	실행 시 메모리 초기화 할때 사용	INIT	사용할 메모리 초기화한다.
데이터 전송	메모리 간에 데이터 전송	DP COPY	메모리 버퍼를 복사하여 다른 메모리로복사한다.
패턴매칭	메모리 버퍼와 스택탑의 값을 비교할 때 사용	EQL	스택탑의 두 값을 비교한다.
흐름제어	실행 흐름 제어 시 사용	JMP	조건없이 명시된 주소로 이동한다.
스택관리	스택을 관리할 때 사용	SWAP	스택탑의 두 값을 변경한다.
산술로직	연산을 할 때 사용	SHL	스택탑의 값을 왼쪽으로 SHIFT 연산을 한다.

가상머신의 중간언어(ETRI Virtual Machine Intermediate Language)는 고급 언어인 eFlowC 로 작성된 프로그램을

컴파일 하여 생성된다. 가상머신 중간언어인 <표 1>과 같이 가상머신 중간언어에 사용되는 대표적인 사용 예제이다. 가상머신 중간언어는 초기화, 데이터 전송, 패턴 매칭, 흐름 제어, 스택 관리, 산술 로직으로 구성된다.

5 장. 가상머신의 테스트 및 실행

가상머신 중간언어를 이용하여 IP 주소 및 포트를 검사할 수 있다. 또한 IP 주소를 확인하여 패킷을 변형하거나 버리는 기능을 수행할 수 있다. [그림 4]는 고급언어 eFlowC 컴파일러를 통해 얻은 결과의 일부이다.

```
   INT 0, 9264              // 메모리 할당
   CAL 0 , main             // 메인 호출
   HALT 0, 0
.global 0
.global 9215
main :
   LDA 1, 20                // IP 주소 로드
   LDIB 0, 4
   LDA 1, 12                // 비교수행할 IP 주소
   LDIB 0, 4
   EQL 0, 0                 // 비교 연산 수행
   JPC 0, forward           // 불법 아이피 경우 forward
                            // 되고 아니면 계속 진행
   LDA 1, 28
   LIT 0, 0
      .
      .
      .
forward :
   JMP 0, main              // 메인으로 이동
```

(그림 4) 가상기계 중간언어을 이용한 IP 필터링

[그림 4] 프로그램은 네트워크를 통해 들어오는 패킷의 IP 주소가 지정된 주소인지를 검사하는 프로그램이다. IP 주소를 검사하여 미리 정의해 놓은 IP 주소가 아니면 [그림 4]에서 흐름 제어를 통해 forward 로 이동시킨다. 그리고 정의해 놓은 IP 주소라면 비교문 수행 후 다음 명령어로 이동한다. 프로그램 카운터가 forward 로 이동하면 아무일도 하지 않고 main 의 다음 위치로 프로그램 카운터를 이동시킨다. 다시 말하면 유효한 IP 주소가 들어올 경우 패킷을 조작하거나 이동하는 기능을 수행하지만 그렇지 않은 경우에는 현재 패킷을 무시하고 미리 정의한 IP 주소가 들어 올때까지 프로그램을 반복 시킨다. [그림 4]의 프로그램과 같이 eFlowC 언어를 이용해 언어 레벨에서 네트워크 프로세서에 최적화된 기능을 정의된 eFlowC 언어로 프로그램을 작성한다. 그리고 eFlowC 컴파일러로 컴파일 하면 [그림 4] 결과를 얻을 수 있다. [그림 4]는 eFlowC 컴파일러가 설치된 범용 컴퓨터에서 가상머신을 통해 프로그램을 실행 할 수 있다.

6. 결론 및 향후 연구과제

본 논문에서 제안한 가상머신은 네트워크 프로세서에서 직접 프로그램 테스트를 하기에 어려움이 있다. 또한 네트워크 프로세서를 개발한 회사가 제공하는 교차개발환경 툴과 개발언어가 서로 다르기 때문에 소스코드 재사용 및 확장이 어려운 점에 대해 문제를 제기했다.

위의 문제를 해결하기 위해 네트워크 프로세서에 최적화된 기능을 언어 레벨에서 정의할 수 있는 eFlowC 언어를 소개했다. 그리고 네트워크 프로세서를 개발한 회사마다 제공하는 교차개발환경과 개발언어가 서로 다른 점에 대한 개선 방안으로 가상머신 중간언어를 제시했다. 그 결과 가상머신 중간언어를 이용하여 소스코드 재사용 및 확장을 가능하게 하였다. 결론은 eFlowC 컴파일러가 설치된 범용 컴퓨터에서 네트워크 프로세서 구조와 같은 환경에서 프로그램 테스트 및 실행을 적절하게 수행할 수 있도록 하였다. 그 결과 범용 컴퓨터에서 테스트를 적절하게 수행할 수 있으며 신뢰성 높은 프로그램을 작성할 수 있을 것이다.

향후 계획으로는 가상머신 중간코드를 변형하여 여러 타겟머신으로 연결시킬 수 있는 언어번역기(Translator)에 대한 연구가 요구된다.

참고문헌

[1] 홍순화, 김재영, 홍원기 " 로드 분산 방법을 이용한 대용량 네트워크 트래픽 모니터링 및 분석", 포항공과대학교 대학원 : 컴퓨터공학과(네트워크), 2002.

[2] 유상경, 안윤영, 김봉태, " Network Processor 동향분석",정보통신산업진흥원,학술정보 주간기술동향 971 호, pp. 16-30, 2000.11.

[3] Intel Home page, http://www.intel.com/intelpress/sum_ixp1200.htm?wapkw=ixp1200

[4] 강구홍, 김익균, 장종수, " 고속 망에 적합한 네트워크 프로세서 기반 인-라인 모드 침입탐지 시스템", 한국정보과학회, pp. 363-374, 2004.

[5] Morandi, Risso, Rolando P, Hagsand, Ekdahl, " Mapping packet processing applications on a systolic array network processor", IEEE CONFERENCES, pp.213-220, 15-17 May 2008.

[6] NetVM Home Page, http://nbee.org/doku.php?id=netvm:index#the_netvm_virtual_machine.

[7] Loris Degioanni, Mario Baldi, Diego Buffa, Fulvio Risso, Federico Stirano, Gianluca Varenni, " Network Virtual Machine(NetVM) : A New Archicture for Efficient and Portable Packet Processing Applications", IEEE CONFERENCES, pp.163-168, June 15- 17, 2005.

[8] Mario Baldi, Fulvio Risso, " A Framework for Rapid Development and Portable Execution of Packet-Handling Applications", IEEE CONFERENCES, pp.233-238, 21 Dec 2005.

[9] 조혜영, 김대영, " 네트워크 프로세서 기반 고성능 네트워크 침입 탐지 엔진에 관한 연구" 정보과학회논문지, pp.113-130, 2006.4.

[10] JAVA Home page, http://www.oracle.com/technetwork/java/javase/downloads/index.html.