NEC 시스템에서의 컴파일 속도 향상을 위한 크로스 컴파일러 서버 구현

이영주, 성진우, 김성준, 이상동, 김중권
한국과학기술정보연구원
e-mail: yjlee@kisti.re.kr

The Implementation of Cross Compiler Server for Speed up in NEC System

Yuong-Joo Lee, Jin-Woo Sung, Sung Jun Kim,
Sang Dong Lee, Joong-Kwon Kim
Korea Institute of Science Technology Information

요 약

슈퍼컴퓨퍼 시스템의 종류에는 크게 벡터 시스템과 스칼라 시스템으로 두 가지 방식으로 나눌 수 있다. 이는 명령어 처리 형태로 구분한 것으로서, 현재 KISTI에서 보유하고 있는 슈퍼컴퓨터인 NEC 컴퓨터는 벡터 시스템으로서 명령어의 수행을 파이프라인 방식으로 처리한다. 벡터 시스템에서 컴파일을 수행할 때는 일반 명령 처리 장치에서 스칼라 형태로 수행되기 때문에 그 처리 속도가 저하되어 다른 스칼라 전용 컴퓨터보다 속도가 느리고 부하가 많이 발생하는 문제점을 안고 있다. 이러한 벡터 컴퓨터의 컴파일 속도를 향상시키기 위해서 크로스 컴파일 서버를 구축하고 이 서버를 NEC 시스템과 연계하여 빠르고 쉽게 크로스 컴파일을 수행할 수 있도록 하였다.

본 연구는 크로스 컴파일러 서버를 구축하고 이 서버에서 자동적으로 크로스 컴파일이 가능하게 함으로서 사용자가 기존의 컴파일 방식처럼 편리하게 이용할 수 있고, NEC 메인 시스템에 부하를 주지 않으면서 컴파일 속도의 성능을 향상할 수 있다.

1. 서론

현대는 컴퓨터 통신의 발달로 인하여 인터넷의 시대에 살고 있으며 업무에 있어서도 컴퓨터 처리에 대한 의존도가 점점 증가하고 있다. 그에 따라서 컴퓨터의 규모가 대형화 되고 보다 더 빠른 컴퓨터를 요구하게 된다. 현재 KISTI에서 보유한 NEC SX-5, 6 컴퓨터는 가장 빠른 슈퍼컴퓨터 중의 하나인 벡터 시스템이다. NEC 벡터 컴퓨터는 프로그램을 처리하는 방식이 파이프라인을 통한 벡터로 처리하기 때문에 그 처리 속도가 빠르지만 일반 명령어를 처리할 때는 스칼라 장치로 처리하기 때문에 벡터 처리에 비하여 상대적으로 느릴 뿐만 아니라 다른 스칼라 전용인 컴퓨터보다도 느리다.

벡터 컴퓨터에서 컴파일 할 때 일반 명령어인 스칼라 처리장치에서 처리되어 그 속도가 느리기 때문에 벡터 컴퓨터 사에서는 별도의 크로스 컴파일러를 제공한다. 크로스 컴파일러는 다른 이기종의 스칼라 컴퓨터에서 컴파일을 한 다음 그 목적 파일을 메인 컴퓨터에서 실행할 수 있는 목적 파일을 생성하여 준다. 그러나 사용자가 크로스 컴파일을 하기 위해서는 크로스 컴파일러 서버에 로그인하고 자신의 파일을 크로스 컴파일서버에 가져가서 컴파일을 한 다음 다시 그 컴파일된 파일과 작업 위치를 벡터 컴퓨터로 옮겨와서 실행하여야 한다. 이럴 경우 크로스 컴파일러의 성능이 좋더라도 사용자가 이를 사용하는 데 불편하면 시간이 다소 늦더라도 편리성을 추구하여 기존의 사용하던 방식대로 사용할 것이다. 이는 메인에서 수행하는 컴파일과 크로스 컴파일하는데 소요되는 전체적인 시간이 거의 비슷하게 소요되기 때문이다.

따라서 크로스 컴파일을 이용하여 속도 향상을 이루면서 기존의 컴파일러처럼 편리하게 사용할 수 있는 자동 크로스 컴파일을 할 수 있는 컴파일러 서버 시스템을 연구 개발하게 되었다.

2. 시스템 구성 환경

2.1 NEC 시스템 구성

NEC SX-5 시스템은 <표 1>에서 보는 바와 같이 벡터 시스템으로서 8개의 CPU가 하나의 메모리를 공유하는 SMP이고, NEC SX-6 시스템도 두 노드로 구성되어 있는 SMP 시스템으로서 노드와 노드 사이의 연결은 IXS(Internode Crossbar Switch)의 고속 채널로 접속되어 있어 SX-6의 두 노드 간의 MPI 작업이 가능하다.

<표 1> NEC 시스템 사양

구분		SX-5	SX-6
모델명		SX-5/8B	SX-6/16M x 2
O.S명		SUPER-UX R13.1	SUPER-UX R13.1
CPU(개)		8	16
Clock(MHz)		312.5	625
메모리(GB)		128	128
성능(GFLOPS)		80	160
디스크	내장(GB)	70	140
	외장(TB)	2.85	2.85

2.2 벡터 시스템 특성

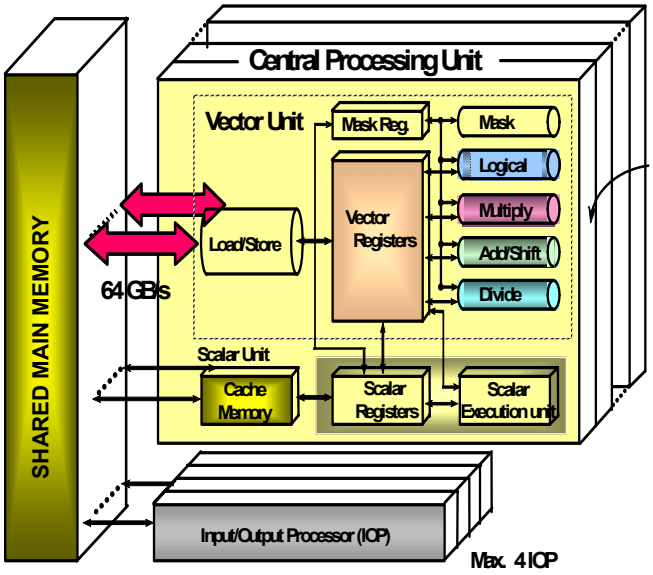
NEC 벡터 시스템은 내부에 벡터 처리 장치와 스칼라 처리 장치를 각각 가지고 있으며 처리하고자 하는 프로그램의 코드에 따라서 각각의 장치에서 처리된다. 따라서 스칼라 명령을 실행할 때에 그 속도가 상대적으로 느린데 그것은 NEC 시스템이 벡터 코드에 더 적합하게 설계되었기 때문이다. NEC SX 시스템은 <그림 1>에서 보는 바와 같이 벡터명령과 스칼라명령을 동시에 실행할 수 있으며, 이를 위해 각각 16개의 Add-Shift Pipelines과 Multiply-Shift Pipelines이 있다.

각각의 Pipeline Unit이 모두 동작중일 때 1 Machine cycle(312.5Mhz)당 1회의 실수 연산이 가능함으로 모든 Pipeline이 동시에 동작할 때 1 Machine cycle 당 32회의 실수 연산이 가능하게 된다.

(Peak Performance = 32/(3.2ns) = 10 x 10^9/sec = 10 GFLOPS

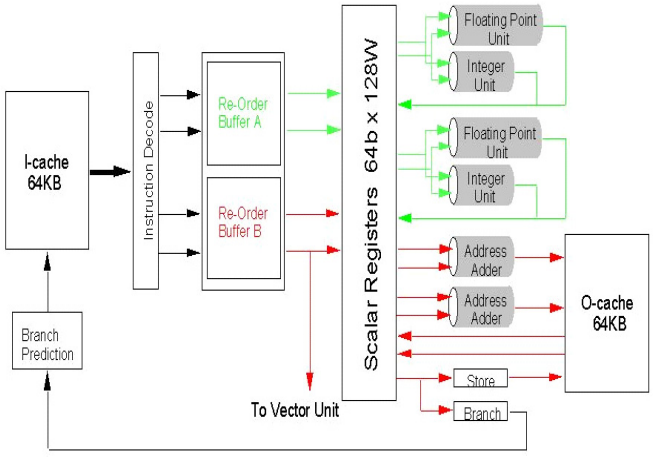
□ Pipeline

실수 연산을 위해 필요한 4단계의 operation unit을 동시에 작동하여 N+3 cycle동안에 N개의 실수연산이 완료하게 된다.(1 machine cycle 당 약 1회의 실수연산 가능)



<그림 1> SX-5 Processor Architecture

아래 <그림 2>는 NEC SX의 Scalar unit을 나타내고 있다. 그림에서 보는 바와 같이 2개의 Scalar Load 와 2개의 Floating point/Integer ALUs가 있으며 스칼라 처리 기능을 높이기 위하여 CPU는 scalar 연산용 및 address 계산용의 register 8Byte로 구성된 128개의 scalar register가 있다.



<그림 2> SX-5 Scalar Unit

3. 크로스 컴파일러 서버 구현 방법

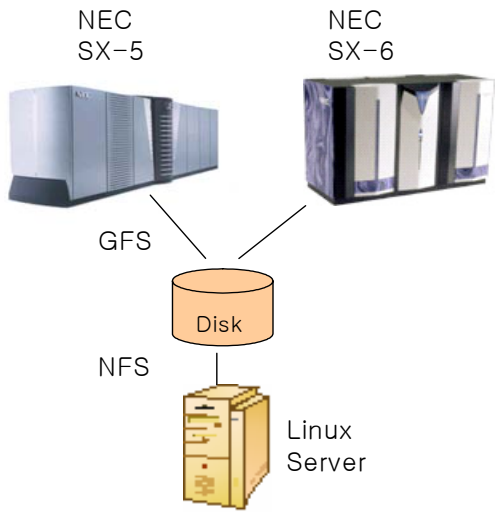
3.1 크로스 컴파일러 서버 사양

크로스 컴파일러 서버 시스템 사양은 <표 2>에서 보는 바와 같이 일반 PC로서 O.S는 Linux 8.0이고 CPU 성능은 2.7Gflops, 메모리는 1GB로 구성되어 있으며 NEC 벡터 컴퓨터하고는 Ethernet을 통하여 연결되었다.

<표 2> Cross Compiler Server 사양

구분	Desktop PC
O.S	Linux 8.0
CPU 성능	2.7Gflops
CPU(개)	1
메모리(GB)	1

전체 시스템 구성은 <그림 3>에서 보는 바와 같이 NEC SX-5와 SX-6 시스템 두 대 그리고 Linux 서버 시스템으로 구성되었다. NEC SX-5와 SX-6는 하나의 홈디렉토리를 GFS(Global File System)로 공유하고 있으며 Login 노드는 SX-6a 이고 Linux 서버 시스템의 홈디렉토리는 NFS로서 SX-6a에 연결하였다. SX-5, SX-6, Linux 시스템은 각각의 노드에 로컬 디스크가 설치되어 있으며 이 로컬 디스크는 각각의 컴파일에 필요한 library 파일과 사용자 작업을 위한 스크래치 공간이 있다.



<그림 3> NEC 전체 시스템 구성도

3.2 크로스 컴파일러 서버 구성 방법

크로스 컴파일러의 구성방법은 여러 가지가 있을 수 있는데 크게 다음과 같이 두 가지로 나눌 수 있고 각각의 방법에는 장단점이 있다.

1안) 크로스 컴파일러 서버를 단독 시스템으로 구성하고 다른 시스템과의 연결은 네트워크로 연결한다. 다른 시스템에서 크로스 컴파일을 하고자 할 때는 크로스 컴파일러 서버 시스템에 login 하여 사용한다.

□ 장점

단독 시스템으로 구축하여 네트워크에 연결만 하면 됨으로 구축하기가 쉽다.

□ 문제점

크로스 컴파일 할 때마다 크로스 컴파일러 서버에 로그인 하고 컴파일 할 파일을 가져다가 컴파일이 끝난 실행 파일을 다시 보낸 다음 작업 하고자하는 Login 노드에서 컴파일 된 실행파일을 가지고 실행한다. 크로스 컴파일을 할 때마다 크로스 컴파일 서버에 로그인 하고 로그오프 해야 하고 해당 파일을 그 때마다 매번 옮겨야 하는 불편이 있다.

2안) Linux 서버를 구축하여 이 서버를 크로스 컴파일러와 NEC 시스템의 로그인 노드를 겸한다.

□ 장점

크로스 컴파일러 시스템이 로그인 노드이기 때문에 별도의 크로스 컴파일러 서버를 구축할 필요가 없으며 로그인 노드에서 직접 크로스 컴파일을 수행할 수가 있다.

□ 문제점

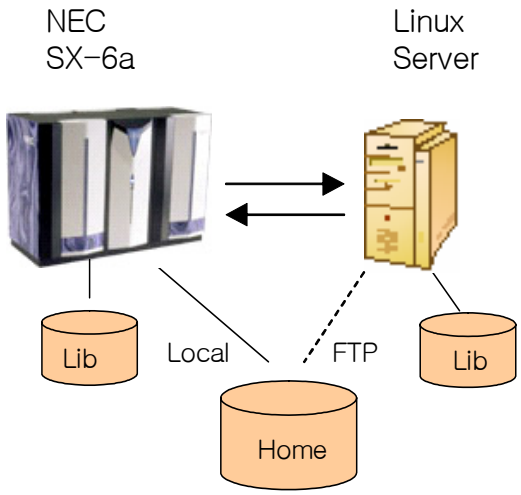
Linux 서버를 NEC 시스템의 로그인 노드로 사용하기 위해서는 먼저 Linux 서버의 시스템이 NEC 시스템과의 하드웨어 및 소프트웨어적으로 완벽한 호환성을 가져야 하지만, 서로의 O/S와 시스템의 특성이 다르기 때문에 Linux 시스템을 NEC의 로그인 시스템으로 사용하기가 어렵울뿐만 아니라 Linux 시스템 자체가 지니고 있는 안정성도 낮다. 이는 사용 중에 로그인 노드에 장애가 있을 경우 시스템 전체에 심각한 영향을 줄 수 있기 때문이다.

3.3 크로스 컴파일러 서버 구현

□ 시스템 구성

크로스 컴파일러 서버의 구성은 <그림 4>와 같이 구성하였다. 이 구성은 위에서 언급한바 있는 1안)과 2안)의 장점을 모두 수용하면서 문제점을 최소화한 구성으로서 NEC SX-6a 시스템이 로그인인 환경을 그대로 두고 별도의 크로스 컴파일러 서버로 구성하였으며, 크로스 컴파일러 서버의 홈디렉토리는 FTP를 이용하여 NEC SX-6a 시스템과 공유함

으로서 하나의 파일을 서로 다른 시스템에서 사용할 수 있도록 하였다.



<그림 4> Cross Compiler Server 구성도

3.4 remote 프로그램으로 자동화 구현

로그인 노드에서 직접 remote shell 등을 사용하여 크로스 컴파일러 서버를 이용하여 일반적인 방법으로 소스 파일을 컴파일 하고자 할 때 두 가지의 문제점이 발생하였다. 하나는 로그인 노드인 로컬 시스템과 remote인 크로스 컴파일러 서버의 shell 환경을 맞추어 주는 것이고, 다른 하나는 rsh 등으로 로컬 시스템에서 remote 시스템으로 매개변수 전달하는 문제점이 있다.

이 두 가지를 해결하기 위하여 C 프로그램을 이용하여 shell 명령어 형식으로 구현하여 문제점을 해결하였다. C 프로그램을 이용하여 로컬 시스템에서 remote인 크로스 컴파일러에 명령어를 전달 실행하면 실행되는 환경 실행 shell에 영향을 받지 않고 C 프로그램을 통하여 매개 변수가 그대로 전달 할 수가 있다.

3.5 실행결과

아래의 <표 3>과 같이 여러분야의 프로그램을 각각의 시스템에서 컴파일하여 그 수행시간을 비교 분석하였다. 각각의 프로그램 특성에 따라서 시간의 차이는 있지만 크로스 컴파일을 이용한 컴파일 시간이 대체로 10배 이상 빠르다는 것을 알 수 있었다. 시스템 자체의 성능은 SX-5가 SX-6에 비하여 더 좋지만 컴파일은 SX-6 시스템이 더 빠르게 나왔다. 이것은 SX-6 시스템의 클럭이 더 빠르기 때문에 클럭의 의존도가 높은 스칼라 명령의 컴파일 시간이 더 빠르다는 것을 알 수 있었다.

<표 3> 벡터 컴퓨터에서의 컴파일과 크로스 컴파일러의 수행 시간 비교

Program	Language	Compile Time(sec)			Ratio
		SX-5	SX-6	Cross	
Couple	Fortran	81	40	7	11.6
YONU AGCM	Fortran	1455	906	116	12.6
QCD_vector	Fortran	108	42	7	14.7
FEM_LSA	C++	975	884	60	16.3

4. 결론

NEC 시스템 자체에서 컴파일 할 때 처리 속도가 지연되는 현상을 Linux 시스템으로 크로스 컴파일러 서버를 구축하여 그 성능을 향상하였다. 이 구현된 시스템을 이용하면 성능 시험에서 보여준 바와 같이 10배 이상의 성능 향상을 나타냈다.

이러한 빠른 크로스 컴파일을 벡터 시스템 내에서 기존의 방법과 동일하게 자동으로 처리할 수 있게 함으로서 성능도 향상되면서 쉽게 사용할 수 있다.

참고문헌

[1] Advanced Computer Architecture by Kai Hwang
[2] NEC SUPER-UX System Design Guide
[3] NEC FORTRAN90/SX Programmer's Guide
[4] NEC C++/SX Programmer's Guide
[5] NEC Network Programmer's Guide
[6] NEC System Administrator's Reference Manual
[7] 이영주 Linux를 이용한 Cross Compiler 구성 한국과학기술정보인프라워크샵, 2003