다중 코어 환경에서의 코어 친화도에 따른 동적 스레드 할당에 관한 연구

이정환 O, 전승협, 차승준, 정성인

한국전자통신연구원

{jeonghwan.lee O, shjeon00, seungjunn, sijung}@etri.re.kr

A Study on the Dynamic Thread Allocation based on Core Affinity in the Multicore Processor

Jeong-Hwan Lee O, Seung-Hyub Jeon, Seung-Jun Cha and Sungin Jung

Electronics and Telecommunications Research Institute (ETRI)

요 약

다중 코어 환경에서 모든 코어를 효율적으로 활용 가능하도록 실현하기 위한 병렬화 프로그래밍의 중요한 요소 기술 중 하나가 코어 친화도(Core Affinity)이며, 이는 스레드를 동적으로 특정 코어에 할당하는 기술이다. 다중 코어 환경에서 스레드를 동적으로 코어에 할당하는 방법에는 코어들의 부하 상태, 동작중인 스레드의 개수, 스레드의 기능 등 다양하나, 본 논문에서는 다중 코어 환경에서 스레드를 특정 코어에 할당하기 위한 코어 친화도의 중요한 이슈를 살펴보고, 코어에 스레드를 할당하는 방법을 제시하고자 한다.

1. 서 론

프로세서의 패러다임은 과거 프로세서의 동작 주파수 속도를 높여 성능을 향상시키는 기술에서, 공정상의 안전성 발열 및 전력 소모 문제 등의 문제로 인해 현재는 코어 개수를 늘려서 성능을 향상시키는 멀티 프로세서 기술 방향으로 진화하고 있다. 이에 현재 4, 8코어가 널리 사용되고 있으며 서버급에서는 15, 18 코어까지 출시되고 있으며, 머지않아 수백 코어 또는 수천 코어로 구성된 매니코어 시스템이 출시될 것으로 예상하고 있다.

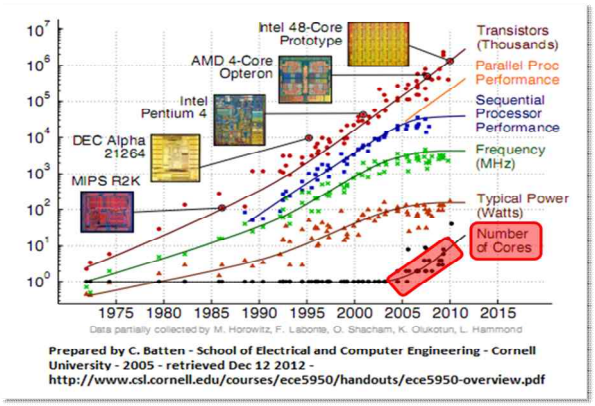

그림 1. 칩당 코어의 개수

이런 다중 프로세서의 흐름에서 모든 코어를 효율적으로 활용 가능하도록 스레드를 특정 코어에서 실행할 수 있도록 코어 친화도(Core Affinity) 기술을 지원해야 한다. 다중 코어 환경에서 스레드를 동적으로 코어에 할당하는 방법에는 코어들의 부하 상태, 동작중인 스레드의 개수, 스레드의 기능 등 다양하나, 본 논문에서는 코어에서 현재 동작중인 스레드의 소 개수로 판단하여 해당 코어에 할당하는 방법을 제시하고자 한다.

2. 연구 내용

본 연구는 각 코어에서 동작중인 스레드의 개수와 프로세스를 구성하는 스레드를 기반으로 하여 스레드를 동적으로 코어에 할당하고자 한다. 하나의 프로세스에서 동작하는 다수의 스레드는 동일한 코어에 할당한다. 이는 동일한 프로세스에서 동작하는 스레드가 각기 다른 코어에 할당되게 되면, IPC 같은 내부 통신으로 인해 성능이 저하될 수 있기 때문이다.

가. 개발 및 시험 환경

- OS : 리눅스 3.19.0-30-generic
- CPU : Intel® Core™ i7-5960X 3GHz (8코어)
- Emulator : QEMU version 2.1.2
- Compiler : gcc 4.9.2

나. 코어 친화도 알고리즘

스레드는 TCB(Thread Control Block) 라는 구조체로 관리되며, 이 구조체 내부에는 코어 친화도를 나타내는 코어 마스크(core mask) 필드가 존재하고 64비트로 구성되어 있다. 코어 마스크 필드는 해당 스레드가 할당될 코어의 위치를 지정하는 데 사용되며, 해당 필드가 0 인 경우, 스레드를 동적으로 내부 알고리즘에 의해 임의의 코어에 할당한다. 예를 들어 코어 마스크 값이 0xFFFFFFFFFFFFFFFE 인 경우,

해당 스레드는 코어 0번에 할당이 되며, 0xFFFFFFFFFFFFFFFD 인 경우, 해당 스레드는 코어 1번에 할당이 된다. 즉, 코어 마스크의 하위 비트부터 코어 번호와 매핑된다.

다중 코어 환경에서의 코어 친화도 기반 스레드 동적 할당을 위한 간략한 알고리즘은 아래와 같다.

```
if( 코어마스크 값 != 0 && 코어마스크 값이 한개의 코어에만 할당되도록 설정되어 있는 경우 )
    return    해당 코어 값

for( 전체 코어 개수만큼 수행 )
{
    if( 동작중이 아닌 코어 or 코어 마스크가 해당 코어 설정이 안되어 있는 경우)
        continue;

    코어에 동작중인 스레드 개수 저장;
    if( 코어에 동작중인 스레드 개수 < 최소로 동작중인 스레드 개수 )
    {
        최소로 동작중인 스레드 개수 = 코어에 동작중인 스레드 개수;
        선택된 코어 번호 = 해당 코어 번호 저장;
        if( 최소로 동작중인 스레드 개수 == 0 )
            return    해당 코어 번호;
    }
}
return    선택된 코어 번호;
```

다. 연구 결과

아래의 두 그림은 코어 친화도 알고리즘 적용 여부에 따라 15개의 스레드를 코어에 동적으로 할당한 결과이다. [그림 2]에서 보는 바와 같이 코어 마스크가 정의된 스레드 9~16 번은 해당 코어에 할당되었으나, 코어 마스크가 미정의된(0x0값) 나머지 스레드는 특정 5번 코어에 집중적으로 할당된 형태를 나타낸다. 이는 다수의 코어를 효율적으로 사용할 수 없게 만든다.

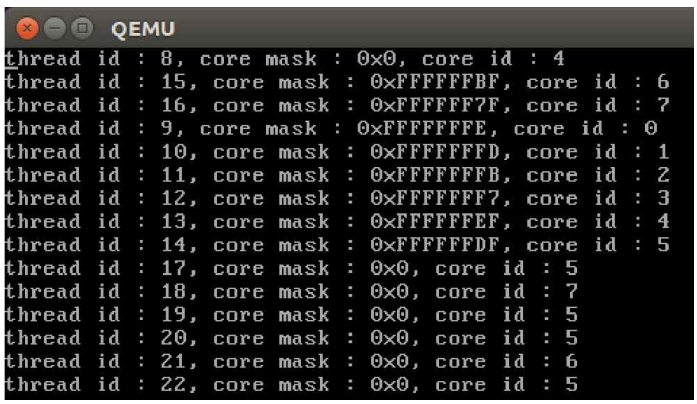

그림 2. 코어 친화도 알고리즘이 미적용된 결과 화면

[그림 3]은 코어 친화도 알고리즘이 적용된 결과 화면으로 보는 바와 같이 코어 마스크가 정의된 스레드 9~16 번은 해당 코어에 할당되었으며, 코어 마스크가 미정의된(0x0값) 나머지 스레드는 여러 코어에 분산되어 할당된 형태를 나타낸다. 이는 다수의 코어를 효율적으로 사용할 수 있게 되었음을 의미한다.

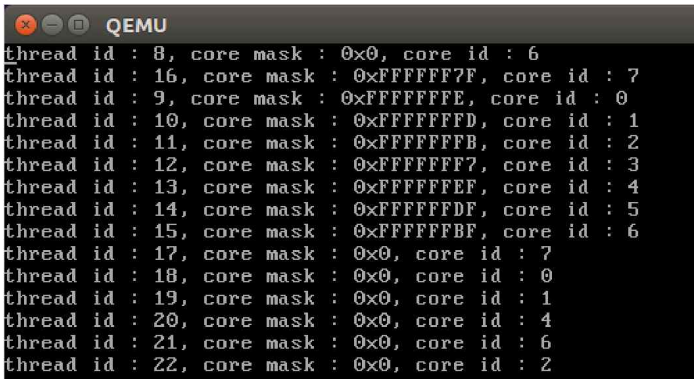

그림 3. 코어 친화도 알고리즘에 의한 결과 화면

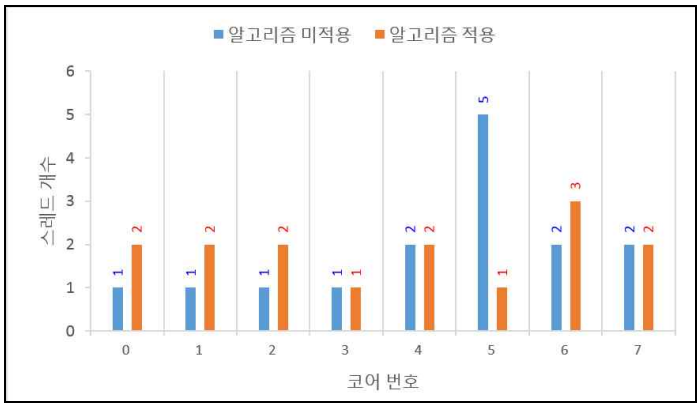

그림 4. 코어에 할당된 스레드 개수 비교 그래프

3. 결론 및 향후 연구

본 논문에서 제시한 코어 친화도와 각 코어별 동작중인 스레드 개수를 기반으로 한 알고리즘을 통하여 멀티 코어 환경에서의 효율적인 스레드 분산 방법을 제시하였다. 향후, 연구는 각 코어의 현재 부하 상태를 고려하여 부하가 적은 코어에게 스레드를 동적으로 할당하는 방법을 제시할 것이다.

ACKNOWLEDGMENT

이 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.B0101-15-0644, 매니코어 기반 초고성능 스케일러블 OS 기초연구(차세대 OS 기초연구센터))

참 고 문 헌

[1] Robert Love, “CPU Affinity”, Linux Journal #111, 2003.07.

[2] Felipe Cerqueira, Arpan Gujarati, Bj¨orn B. Brandenburg, “Linux’s Processor Affinity API”, RTSS, 2014.