CPU와 GPU의 혼합 병렬 계산에 대한 성능 분석

황근창, 김영태
강릉원주대학교 컴퓨터공학과
e-mail:ka1325@cs.gwnu.ac.kr, ykim@gwnu.ac.kr

Performance Analysis on Parallel Processing of a Hybrid of a CPU and a GPU

Keunchang Hwang, Youngtae Kim
Dept. of Computer Science, Gangneung-Wonju National University

요 약

본 논문에서는 고성능 병렬 계산 장치로 주목받고 있는 GPU를 CPU와 동시에 병렬로 사용한 계산 성능을 분석하였다. 성능 분석을 위하여 원주율(π)을 적분으로 계산하는 CUDA 프로그램을 사용하였으며, 전체 계산을 GPU 대비 CPU 계산 부분으로 할당하여 성능을 분석하였다.

1. 서론

그래픽 장치인 GPU는 격자 구조의 프로세서를 병렬로 계산하는 장치로서 가격 대비 성능이 뛰어나다. 따라서 일반 계산에서 컴퓨터의 계산 성능을 향상시키는 프로그램 기술은 초고속의 계산을 요구하는 많은 분야에서 활용하고 있다[1, 2].

GPU는 컴퓨터에서 CPU와는 별개로 주변 장치를 연결하기 위하여 PCI-E(Peripherical Component Interconnect Express)를 통하여 장착되기 때문에 CPU와 동시에 계산이 가능하다[3]. 하지만 일반적인 프로그램에서는 CPU가 GPU의 계산을 마칠 때까지 기다리게 되어 CPU의 효율성이 낭비된다.

본 논문에서는 GPU와 CPU를 동시에 계산하여 컴퓨터 자원을 최대한 활용한 계산의 성능을 분석하였다. 성능 분석을 위하여 원주율(π)을 구분구적법으로 계산하는 CUDA 프로그램을 사용하였으며 계산의 일부분을 CPU에 할당하여 성능을 분석하였다.

본 논문의 2장에서는 GPU와 CPU의 계산 프로그램 구조를 설명하고 3장에서는 실험 결과를 분석하였다. 마지막으로 4장에서는 결론을 작성하였다.

2. GPU와 CPU의 혼합 병렬 계산 프로그램

본 논문에서는 성능을 계산하기 위하여 원주율(π)을 구분구적법으로 계산하는 CUDA 프로그램을 구현하였다. 이 징에서는 프로그램의 구현 방식에 대해 알아본다.

2.1 원주율 계산 방식

본 논문에서의 원주율을 계산하는 프로그램은 구분구적법[4]을 사용하여 원의 구간[0, 1]을 N 등분했을 때의 각각의 사각형 넓이의 합을 이용하였다.

구분구적법에 이용되는 식은 단위원(중심이 (0, 0)이고, 반지름이 1인 원) 넓이식인 $x^2+y^2=1$ 식을 y에 대한 식으로 변경한 $y=\sqrt{1-x^2}$ 식을 사용하였다. 각 사각형의 가로 길이는 1/N으로 고정하고, 각 사각형의 오른쪽 x좌표를 위의 식에 대입하여 나온 값을 y값으로 사용하여 사각형의 넓이를 구한다. 각 사각형을 모두 합한 넓이는 구하고자 하는 원의 넓이보다 크지만 구분적분법은 N의 크기가 늘어날수록 각 사각형이 차지하는 넓이는 단위원의 넓이에 가까워진다. 최종적으로 사각형 넓이들의 합은 단위원 넓이의 1/4이기 때문에 4를 곱하여 단위원 전체의 넓이를 계산하였고 이를 통하여 원의 넓이인 π값을 계산하였다.

2.2 병렬 프로그램의 구조

GPU와 CPU의 병렬 계산을 위하여 GPU 프로그램은 CUDA를 사용하였다[5]. CUDA는 Nvidia에서 개발한 GPU를 사용할 수 있는 프로그램 모듈로서 C 언어를 사용할 수 있다[6].

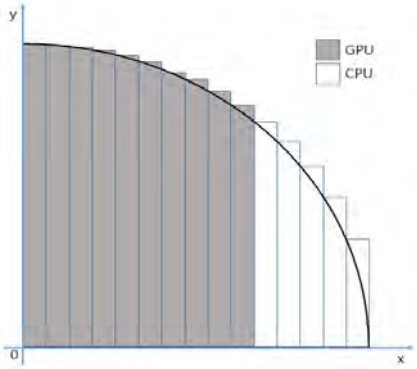
일반적인 CUDA 프로그램에서는 데이터 의존성에 따라 GPU와 CPU가 순차적으로 계산을 수행한다. 이 경우에는 GPU의 계산을 마칠 때까지 CPU가 기다려야 한다. 하지만 본 논문에서의 프로그램에서는 대기 함수를 CPU 함수 이후에 진행함으로써 동시에 진행할 수 있도록 하였다. 이 경우에는 GPU 함수와 CPU의 수의 데이터 의존성이 없어야 한다. 프로그램의 기본 구조[7]는 다음과 같다. 프로그램에서 ①은 GPU 함수, ②는 CPU 함수 그리고 ③은 GPU의 계산을 기다리는 함수이다.

```
GPU_PI<<<...>>>(...);-①
CPU_PI(...);-②
cudaThreadSynchronize();-③
```

위 프로그램에서 GPU와 CPU의 효율적인 작업의 비율

을 찾기 위해서 CPU만으로 값을 계산하는 방식에서 GPU의 비율을 일정하게 늘려가며 프로그램을 실행하였다.

(그림 1)은 전체 조각의 수가 15개이고, GPU 비율이 2/3(함수 ①에서 계산), CPU 비율이 1/3(함수 ②에서 계산)인 예를 그림으로 나타낸 것이다. 구분 구적법의 계산을 위하여 전체 15개의 사각형 중 10개의 사각형의 넓이는 GPU가 계산하고, 나머지 5개의 사각형은 CPU가 계산하도록 하였다.



(그림 1) GPU와 CPU 계산의 할당 예

3. 실험 결과

3.1 CPU와 GPU의 계산 환경

본 논문에서 사용한 CPU와 GPU의 계산 환경은 <표 1>과 같다. CPU의 성능은 3.4GHz이고 GPU의 성능은 1.3GHz이지만, GPU의 스레드를 계산을 위한 최대 수인 256개를 사용할 경우에는 1.3×256=332.8GHz이기 때문에 이론적인 성능으로는 GPU가 97.9배가 더 빠르다.

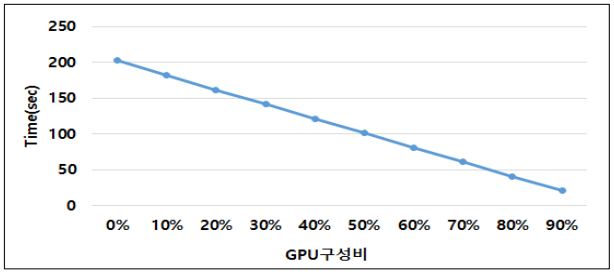
<표 1> CPU와 GPU의 계산 환경

CPU	Intel Zeon Nehalem I7-2600 3.4GHz
GPU	Nvidia Tesla C1060 1.3GHz 512 스레드

3.2 성능 분석

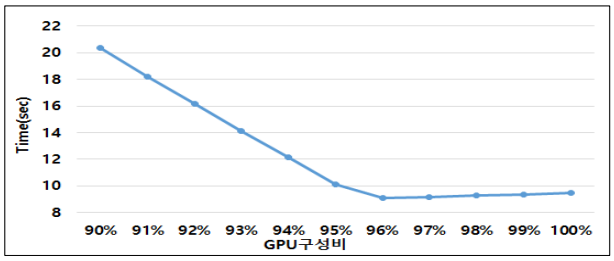
성능 분석 프로그램에서 구분구적법의 계산을 위하여 조각의 수를 10^{10}으로 하였으며 GPU 함수는 256개의 스레드를 사용하였다.

(그림 2)는 GPU의 계산 비율을 90%까지 10%씩 증가하면서 측정한 계산 시간을 초(sec)로 보여 준다. CPU 대비 GPU의 사용이 증가할수록 성능이 선형으로 증가함을 볼 수 있다.



(그림 2) GPU 계산 비율(-90%)에 따른 실행시간

(그림 3)에서는 GPU 작업 비율을 90% 이상에서 1%씩 증가하면서 계산 시간을 비교하였다. 그림에서 볼 수 있듯이 계산 시간이 96%까지는 감소하다가 다시 97%부터 증가하는 것을 볼 수 있다. 따라서 CPU에게 작업을 3-4% 정도 할당하는 것이 계산 성능을 가장 극대화할 수 있는 것으로 보인다. 이 결과는 GPU의 이론적인 성능이 CPU보다 90배 이상 빠르기 때문에 이론적인 성능 비교 결과와 거의 일치함을 볼 수 있다.



(그림 3) GPU 계산 비율(90%-100%)에 따른 실행시간 비교

4. 결론

본 논문에서는 GPU와 CPU를 동시에 계산하여 병렬 프로그램의 성능을 극대화하기 위한 성능 분석을 하였다. 성능 분석을 위해서 원주율(π)을 계산하는 프로그램을 구현하여 GPU 계산 대비 CPU의 계산 비율을 변화시키면서 성능을 분석하였다. 성능 분석 결과, CPU에 3-4%정도의 작업을 할당하는 것이 성능에 가장 효율적으로 나타났다. 이는 GPU 대비 CPU의 이론 성능을 비교한 결과와 거의 일치함을 볼 수 있다.

한편, 일반적인 계산에서는 데이터 의존성이 존재하는 경우가 많고, CPU를 동시에 활용한 성능의 향상이 크지 않기 때문에 GPU를 사용한 계산만으로도 충분한 성능을 낼 수 있다고 보인다.

사사

본 논문은 2016년 강릉원주대학교 서울어코드활성화사업단의 지원을 받아 발표합니다.

참고문헌

[1] 정수진, “효율적인 GPU 컴퓨팅을 위한 CUDA 기술 및 사례 소개”, CAD&Graphics p.74, 2008.4.

[2] 김영태, 황규현, “NVIDIA GPU 상에서의 난수 생성을 위한 CUDA 병렬프로그램”, 정보과학회논문지, 42권 12호 2015년 12월.

[3] Patterson D., J. Hennessy, “컴퓨터 구조 및 설계 4판 ARM버전”, 비제이퍼블릭, 2010.

[4] 신보미, 구분구적법과 정적분의 개념 분석, 한국학교수학회논문집, 2008.

[5] NVIDIA, CUDA Architecture, Introduction & Overview Version 1.1, 2009.

[6] D. Kirk, “NVIDIA CUDA software and GPU parallel computing architecture”, ISMM, Vol.7, 2007

[7] NVIDA CUDA C Programming Guide v7.5, Sep. 1, 2015.