GPU 병렬성을 이용한 정보 검색 시스템의 성능 개선

박일남○, 배병걸, 임은진, 강승식
국민대학교 컴퓨터공학과
pin0156@naver.com bazel1984@naver.com ejim@kookmin.ac.kr sskang@kookmin.ac.kr

Improving the Performance of Information Retrieval System by using GPU Parallelism

Il-Nam Park○, Byunggurl Bae, Eun-Jin Im, Seung-Shik Kang
School of Computer Science, Kookmin University

요 약

정보 검색 시스템에서 사용되고 있는 벡터 공간 모델은 벡터 유사도 계산 속도에 따라 전체 시스템의 성능에 많은 영향을 미친다. 본 논문에서는 문서 유사도 계산 성능을 향상시키기 위하여 GPU(Graphic Processing Unit)를 이용하는 CUDA프레임워크에서 병렬처리 연산을 구현하였으며, CPU(Central Processing Unit) 환경에서의 연산 속도와 비교했을 때 최대 15배의 성능 향상 효과가 있음을 확인하였다.

주제어: 벡터 공간 모델, 코사인 유사도, GPU

1. 서론

정보 검색 시스템에서 벡터 공간 모델[1]의 코사인 유사도 계산은 문서와 문서 간에 있어 각각의 벡터들에 대하여 곱셈과 덧셈 연산을 수행한다. 이러한 코사인 유사도 계산은 입력 문서에서 출현하는 단어 벡터들이 많을수록 연산량이 증가하여 연산 속도가 늦어지게 된다.

본 논문은 문서내 특징 벡터들이 늘어남에 따라 증가하는 벡터 공간 모델의 코사인 유사도 연산량에 대하여 연산 속도를 줄이기 위해 싱글 코어인 CPU 대신에 멀티 코어인 GPU를 사용하여 벡터 공간 모델의 코사인 유사도 연산을 병렬로 처리 하였다. GPU 프로그래밍을 위한 도구로는 NVIDIA사에서 제공하는 CUDA프로그래밍 [2][3] 환경을 사용하였다.

2. 코사인 유사도

문서 d_j와 d간의 유사도 계산을 위한 벡터 공간 모델의 코사인 유사도(cosine similarity) 값은 문서 벡터 d_j와 d간의 내적을 각각의 벡터 크기의 곱으로 나누면 구할 수 있으며 (식1)처럼 표현한다.

$$sim(d_j,d) = \frac{\vec{d_j} \bullet \vec{d}}{|\vec{d_j}| \times |\vec{d}|} = \frac{\sum_{i=1}^{t} w_{ij} \times w_{id}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \times \sqrt{\sum_{i=1}^{t} w_{id}^2}}$$

(식 1) 코사인 유사도 계산식

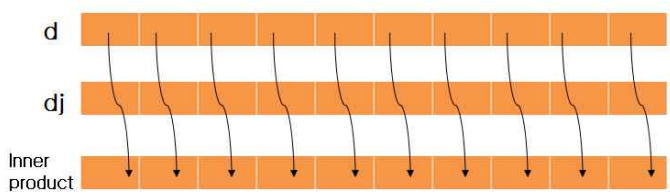
문서 벡터는 문서의 수가 증가할수록, 문서 내 단어들이 증가할수록 벡터 차원수가 증가하기 때문에 문서 수가 증가할수록 코사인 유사도 연산량은 비례적으로 증가하므로 연산 수행 시간이 증가하는 문제가 있다.

3. 코사인 유사도 연산을 위한 CUDA 설계

벡터 차원수가 증가할수록 벡터의 곱셈과 덧셈에 대한 연산 처리량은 증가하게 된다. CPU는 코어가 하나이기 때문에 연산속도는 일정할 수밖에 없으므로 많은 연산 처리량에 비례하여 연산 속도가 좌우된다. 본 논문에서는 많은 처리량에 대하여 연산 속도를 줄이기 위해 NVIDIA사에서 제공하는 CUDA프로그래밍 환경을 사용하여 멀티 코어인 GPU로 코사인 유사도 연산을 병렬 처리 하도록 설계 하였다.

3.1. 벡터 유사도 곱셈연산

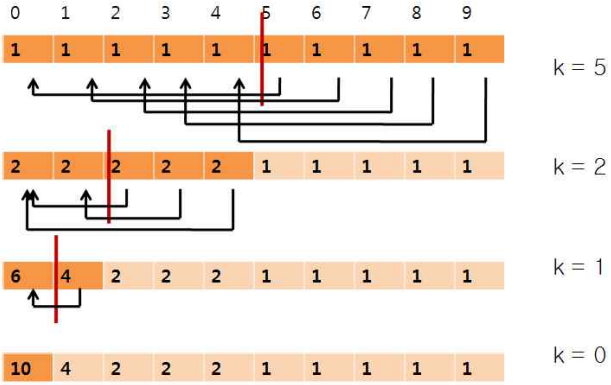
코사인 유사도 계산에서 곱셈연산은 벡터들의 곱셈부분과 벡터 크기를 구하기 위한 제곱연산에 사용된다. 입력된 문서벡터 d_j와 d에 대하여 곱셈을 그림 1과 같이 병렬로 수행하도록 CUDA프로그래밍을 하였다. 한 번의 곱셈연산을 수행할 때 n개 쓰레드들을 사용하여 n개의 벡터들에 대하여 처리할 수 있도록 하였으며, 여기서 n은 NVIDIA의 GPU가 가질 수 있는 최대 블록 내 쓰레드 개수인 512개로 하였다.



(그림 1) 문서 d_j와 d간의 병렬 곱셈 수행

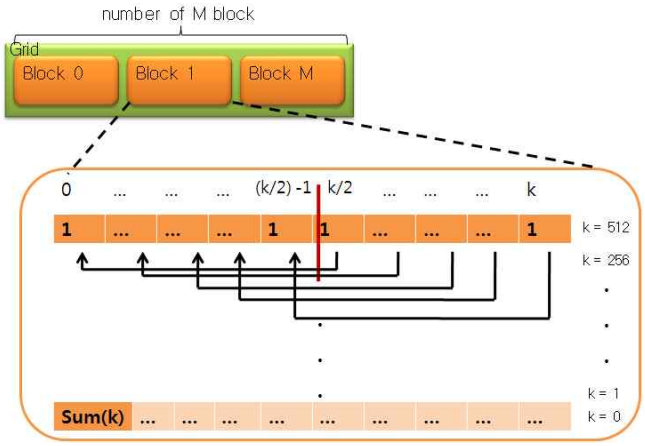
3.2. 벡터 유사도 덧셈연산

문서 d_j와 d의 단어 벡터들에 대하여 각각 곱셈연산을 수행한 뒤 벡터의 합 부분을 계산하기 위하여 수행되는 덧셈연산은 단어 벡터 차원 수만큼 연산을 수행하여야 한다. 단어 벡터 개수가 m개라면 코어가 하나인 CPU는 m번의 연산 수행을 차례대로 하여야한다. 이러한 CPU 연산은 m이 커지면 커질수록 연산 수행 시간이 비례적으로 증가한다. m에 대하여 비례적으로 증가하는 연산 수행 시간을 줄이고자 하나의 블록당 그림 2와 같이 $\log_2 k$만큼 수행할 수 있도록 하였으며, k는 NVIDIA의 GPU가 가질 수 있는 최대 블록 내 쓰레드 개수인 512개로 하였다.



(그림 2) 덧셈 병렬 처리 방법

하나의 블록당 $\log_2 k$만큼 덧셈 연산을 수행할 수 있으므로 m개의 벡터 차원에 대하여 $\lceil m/k \rceil + \log_2 k$ 로 덧셈 연산을 병렬로 처리할 수 있도록 그림 3과 같이 CUDA프로그래밍을 하였다.



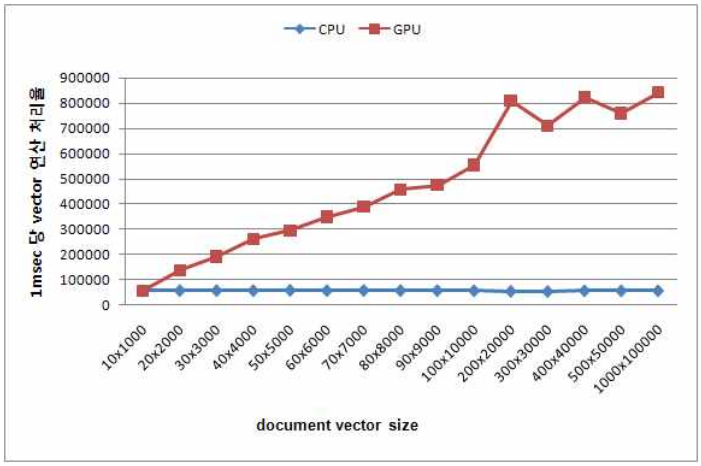
(그림 3) 블록당 512개 쓰레드를 사용한 덧셈 방법

4. 실험 및 비교 평가

코사인 유사도 연산을 GPU를 사용하여 병렬로 처리하였을 때 성능 효율성에 대한 평가를 하기위하여 CPU와 비교하는 실험을 하였다. 표 1은 실험 PC사양을 나타내며, 그림 4는 입력 벡터 크기에 따른 1msec당 CPU와 GPU에서의 연산 처리율을 나타내는 그래프이다. 그래프의 x축은 문서 벡터 사이즈를 나타내며 문서 벡터 크기는 n x m으로 이루어져있으며, n은 문서개수, m은 특징 벡터 차원을 나타낸다. y축은 1msec당 처리량을 나타내고 CPU는 입력 벡터 크기가 클수록 1msec당 처리량이 일정하다. 본 논문에서 제시한 GPU 방법은 벡터 크기와 비례하여 처리할 수 있는 처리량이 증가함을 볼 수 있다. 입력 벡터 최대 크기일 때 GPU방법이 CPU방법보다 15배나 처리율이 좋은 것을 볼 수 있다.

<표 1> 실험 PC 사양

구분	성능
CPU	인텔 코어i3-530 : 듀얼코어, 2.93GHz
GPU	Device : Tesla C2050 Number of multiprocessors : 14 Number of cores : 448



(그림 4) 벡터크기에 따른 1msec 당 연산 처리율

5. 결론

본 논문에서는 정보검색 분야에서 사용되고 있는 벡터 공간 모델의 유사도 연산 속도를 향상시키기 위하여 VNIDIA사의 GPU프로그램 개발 환경인 CUDA를 이용하여 병렬로 처리하였으며, 실험 결과 CPU환경에서의 연산 처리 방법보다 최대 15배의 성능의 개선 효과가 있음을 확인할 수 있었다.

참고문헌

[1] G. Salton, A. Wong and C. S. Yang, "A vector Space Model for Automatic Indexing", ACM Article, 1975.

[2] Greg Ruetsch, Brent Oster "Getting Started with CUDA, "http://developer.nvidia.com"

[3] Jason Sanders, "CUDA by Example: An Introduction to General-Purpose GPU Programming", Addison-Wesley, 2010.