한국지능정보시스템학회
02 춘계학술대회 논문집, pp. 401~409

색인어 군집화를 이용한 효율적인 병렬정보검색시스템+

강재호* · 양재완** · 정성원*** · 류광렬*** · 권혁철*** · 정상화***
{jhkang, jwyang, swjung, krryu, hckwon, swchung}@pusan.ac.kr

Term Clustering and Interleaving for Parallel Information Retrieval

Jaeho Kang · Jae-Wan Yang · Sung-Won Jung
Kwang Ryel Ryu · Hyuk-Chol Kwon · Sang-Hwa Chung

요 약

인터넷과 같은 대량의 정보에 대응할 수 있는 고성능 정보검색시스템을 구축하기 위해서는 지금까지 고가의 중대형컴퓨터를 주로 활용하여 왔으나, 최근 가격대 성능비가 높은 PC 클러스터 시스템을 활용하는 방안이 경제적인 대안으로 떠오르고 있다. PC 클러스터 상에서의 병렬정보검색시스템을 효율적으로 운영하기 위해서는 사용자가 입력한 질의를 처리하는데 요구되는 개별 PC의 디스크 I/O 및 검색관련 연산을 모든 PC에 가능한 균등하게 분배할 필요가 있다. 본 논문에서는 같은 질의에 동시에 등장할 가능성이 높은 색인어들끼리 군집화하고 생성된 군집을 활용하여 색인어들을 각 PC에 분산저장함으로써 보다 높은 수준의 병렬화를 달성할 수 있는 방안을 제시한다. 대용량 말뭉치를 활용한 실험결과 본 논문에서 제시하는 분산저장기법이 충분한 효율성을 가지고 있음을 확인하였다.

Key words: 병렬정보검색, 색인어 클러스터링(군집화), PC 클러스터

1. 서론

최근 인터넷의 보급이 급격히 확대됨에 따라 정보검색 시스템이 처리해야 하는 정보의 양과 사용자의 검색요구는 폭발적으로 증대하고 있다. 이러한 수요에 대응하기 위하여 대부분의 정보검색 서비스 전문업체들은 고가의 중대형 서버 또는 슈퍼컴퓨터를 사용하여 서비스를 제공하고 있다. 대표적인 예로 AltaVista는 수십기가바이트의 주기억 용량을 가진 초대형 시스템을 사용하여 하루에 수백만 건의 검색 연산을 하고 있지만, 이러한 고가의 컴퓨터는 거액의 외화 부담을 요구할 뿐 아니라 대규모 사업자가 아닌 경우에는 거의 사용이 불가능하다. 반면에, 저가의 PC들을 고속 네트워크로 연결함으로써 고성능의 병렬 시스템을 실현하는 PC 클러스터 구조는 정보량이 폭증하는 검색 분야뿐 아니라 고성능과 함께 빠른 응답시간이 요구되거나 실시간 처리가 필요한 다양한 응용분야에서 저비용으로 시스템을 구축할 수 있는 대안으로 주목받고 있다(Lin and Zhou, 1993, Samanta *et al.*, 1999).

PC 클러스터 기반의 병렬 정보검색 시스템을 구현함에 있어서 병렬처리의 효율을 극대화하기 위해서는, 검색대상 자료를 각 PC의 하드디스크에 골고루 분산저장함으로써 I/O의 병목현상을 최소화하고, 각 PC에서의 검색계산 부하를 최대한 균등화하는 방안을 찾아야 한다. 정보검색의 병렬화를 위한 초기의 연구 중 대표적인 것으로는 Stanfill의 방법(Stanfill and Thau, 1991)을 들 수 있으나, 이는 기본적으로 Connection Machine을 대상으로 한 것으로서 대단히 고가의 하드웨어를 필요로 하는 방법이다. 이 방법은 문서가 무작위적으로 각 노드에 흩어지도록 색인어 역파일을 분할하는 것을 기본으로 하고 있는데, 어떤 입력 질의와 관련된 문서들이 일부 노드에 편중되어 나타날 가능성에 대해 적극적인 대비를 하지는 않는 방안이다. 여러 디스크 상의 색인어 역파일 분할을 성능의 측면에서 분석한 연구로는 (Jeong and Omieciniski, 1995)가 있다. 여기에서는 공유메모리 기반 병렬컴퓨터에서 고성능 디스크 I/O를 지원하기 위하여 디스크 어레이의 여러 디스크에 색인어 역파일을 분할하는 방법과 이에 따른 성능를 시뮬레이션을 통하여 평가하였다. 이 연구에서는 색인어별 또는 문서별로 색인어 역파일을 각 디스크에 분할하는 방안을 다양한 상황에서 시뮬레이션하였고, 특히 색인어 단위로 역파일 분할 시, 질의에 나타날

+ 정보통신부에서 지원하는 2001년도 대학기초연구지원사업(정보통신연구진흥원)으로 수행
* 동아대학교 지능형통합항만관리연구센터
** 온빛시스템 정보기술연구소
*** 부산대학교 전기전자정보컴퓨터공학부

색인어의 확률을 고려한다면 상당한 효과가 있음을 확인하였다. 그러나 질의어의 등장확률을 독립적으로 가정하여 현실적으로는 하나의 질의에 동시에 등장할 수 있는 색인어간의 연관관계를 보다 면밀하게 반영하지 못하였다는 한계를 가진다.

병렬 정보검색의 효율 향상을 목적으로 하는 저장방식을 제시한 또다른 연구로는 (강유경 *et al.*, 2001)이 있는데, 이 연구에서는 문서를 분류(classification)하는 방법을 사용하여 유사한 문서들을 군집으로 묶고, 생성된 군집에 대한 색인어 역파일 구조를 추가한 계층적인 검색방안을 제안하였다. 검색시에는 군집단위의 색인어 역파일을 먼저 활용하여 대상문서를 여과함으로써 병렬 정보검색의 효율을 향상시켰으나, 분류작업을 위해서는 수작업으로 학습데이터를 준비하여야 한다는 부담과 검색결과의 정확도와 재현율 측면에서 기존 정보검색시스템과는 차이가 있을 수 있으므로, 일반적인 정보검색시스템에 손쉽게 적용할 수 있는 방법이라고 하기는 어렵다.

PC 클러스터 기반의 병렬 정보검색 시스템을 제안하면서 색인어 역파일의 효과적인 분산저장 방식을 소개한 최근의 연구로 (Chung *et al.*, 2000)이 있다. 이 연구에서는 디클러스터링(declustering) 기법을 이용하여 색인어간의 연관관계를 기반으로 색인어 역파일을 분산저장함으로써 무작위적 분산저장보다 성능향상을 이룰 수 있음을 보인 바 있다.

본 논문에서는 PC 클러스터 기반의 병렬 정보검색시스템의 효율향상을 위하여 색인어 역파일을 보다 효과적으로 분산저장할 수 있는 방안을 제시하고자 한다. 기존 연구에서는 면밀하게 고려하지 못하였던 질의내에 동시에 등장하는 색인어들간의 관계를 활용하기 위하여, 먼저 질의에 동시에 나타날 가능성이 높은 색인어들끼리 묶어 군집화(clustering)화 하였다. 그리고 생성된 군집정보를 활용하여 각 PC의 하드디스크에 색인어를 할당함으로써 검색시 PC들의 작업량을 보다 균등화함으로써 병렬효율을 향상시켰다. 정보검색 분야에서 이러한 군집화기법들(Schutze and Silverstein, 1997, Silverstein and Pederen, 1997)을 이용한 연구는 상당히 진척되어 있으나, 모두 검색 결과를 문서의 측면에서 정리하여 사용자가 원하는 문서를 보다 쉽게 찾을 수 있도록 하는데 그 초점이 맞추어져 왔고, 본 논문에서 제시하는 바와 같이 병렬검색의 효율 향상을 목표로 하지 않았다.

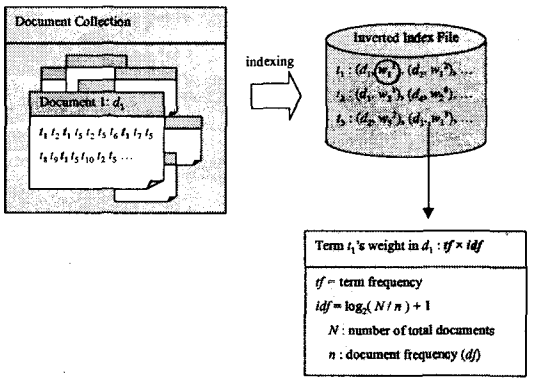
약 50만 건의 신문기사들로 구성된 말뭉치를 활용한 실험 결과 본 논문에서 제안하는 색인어 군집화 및 분산저장 기법을 적용함으로써 단순한 색인어 분산저장 방식보다 검색 성능을 더욱 향상시킬 수 있음을 확인하였다.

본 논문의 구성은 먼저 2장에서 정보검색에 대한 일반적인 설명을, 3장에서는 병렬효율을 향상시키기 위한 색인어 역파일의 분산저장방안의 개념을 소개한다. 이어지는 4장에서는 본 논문에서 제시하는 동시등장 기중치 기반의 색인어 군집화 방법과 생성된 군집을 이용한 분산저장 방안을 구체적으로 소개한다. 본 논문에서 제시하는 기법들을 활용한 실험결과를 5장에서 제시하여 분석하고, 마지막 6장에서는 결론 및 향후 연구과제를 제시한다.

2. 정보검색

정보검색(Frakes and Baeza-Yates, 1992)은 사용자가 빠른 시간내에 직접 파악하기가 어려운 방대한 양의 데이터에서 사용자가 요구하는 질문의 답으로 적절한 내용을 추출하고 가공하여 제시하는 컴퓨터 응용분야이다. 인터넷 정보검색과 신문기사 검색 등 현재 실용적으로 활용되고 있는 정보검색분야에서는 구축된 문서 및 이들 문서간의 연결 형태로 이루어진 데이터를 대상으로, 사용자가 질의어의 나열형태로 표현되는 질의를 입력하면, 검색시스템은 보유한 문서와 주어진 질의와의 적합정도를 계산하여 점수화하고 정렬하여 사용자에게 제시하는 형태로 구현되어 활용되고 있다. 본 논문에서도 이러한 일반적인 문서 정보검색모델을 바탕으로 한다.

2.1 문서 모델링과 색인어 역파일



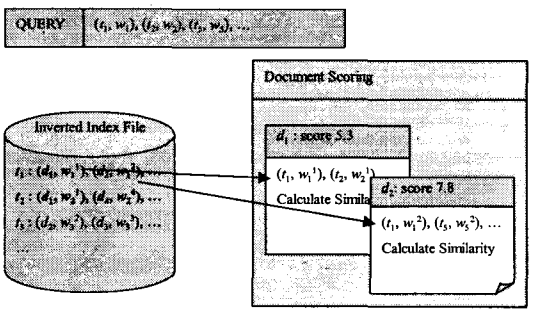
[그림 1] 색인어 역파일 생성

[그림 1]은 문서의 집합으로 이루어진 데이터를 검색에 효율적인 구조인 색인어 역파일(Inverted Index File) 형태로 가공하는 예를 보이고 있다. 하나의 문서는 등장 단어가 순서대로 나열된 데이터로 표현할 수 있다. 문서의 구조가 파악될 수 있는 경우에는 제목, 저자 또는 타 문서와의 연결관계와 같은 추가적인 정보도 포함하는 형태로 표현되어 보다 정확한 검색을 위한 자료로 활용되기도 한다. 문서에 등장하는 단어들은 동사 및 형용사등의 원형을 찾는 스테밍(stemming) 과정과 대부분의 문서에 등장하기

때문에 검색시 효용가치가 없는 in, the와 같은 단어를 제거하는 불용어(stop words) 제거 과정을 거쳐, 색인에 사용할 수 있는 색인어의 나열로 표현한다. 한국어 문서의 경우 여기에 형태소분석을 위한 단계가 추가로 필요하다.

문서를 정보검색에서 보편적으로 활용되고있는 순서를 고려하지 않는 색인어의 집합형태로 모델링하는 경우, 하나의 문서는 색인어를 축으로 하는 다차원상의 벡터로 표현할 수 있다. 이때 특정 색인어 축에서의 문서벡터의 길이는 해당 색인어의 문서내에서의 중요도와 전체 문서군에서의 중요도를 함께 고려할 수 있는 *tf*×*idf*를 활용한 표현이 가장 널리 활용된다. *tf*×*idf*는 문서내에서의 해당 색인어의 상대적인 중요도인 *tf* (term frequency) 항목과 전체 문서군에서의 해당 색인어의 중요도를 표현하는 *idf* (inter-document frequency)를 각기 계산하여 곱한 값이다. *tf*는 색인어가 해당 문서에서 얼마나 중요한지를 표현하는데, 문서에 많이 등장할수록 그 색인어가 해당 문서에서 중요하다는 가정으로 색인어의 등장횟수로 계산한다. 보유한 문서들간의 크기 차이가 심한 경우에는 개별 문서에서 최다등장색인어의 등장횟수로 정규화하여 사용할 수도 있다. *idf*는 전체 문서군에서 색인어의 중요성을 추정하기 위하여 사용되는데, 많은 문서에 등장하는 색인어일수록 보다 보편적인 색인어로 간주되어 그 중요도가 낮도록 전체문서수를 해당 색인어가 등장하는 문서수로 나눈 값이다. 등장 문서수에 따라 *idf* 값이 급격하게 변하는 것을 방지하기 위하여 밑수가 2인 *log*를 취하는 경우가 일반적이며, 최소값을 보장하기 위하여 상수값을 더할 수도 있다. 색인어 역파일은 색인어별로 등장 문서 및 각각의 문서에서의 해당 색인어의 가중치를 파일에 기록한 형태로, 질의로 주어지는 색인어가 나타나는 문서에 대한 정보를 효율적으로 찾을 수 있도록 구성한 형태이다.

2.2 질의 모델링과 유사도 계산



[그림 2] 색인어 역파일을 이용한 검색

질의가 주어졌을 때 색인어 역파일을 이용하여 관련문서를 찾고 문서간의 우선순위를 매기는 방법에 대해서는 [그림 2]에 도시하고 있다. 먼저 질의는 문서와 동일한 형태로 표현한다. 즉 사용자가 제시한 질의를 각각의 질의어를 색인어로 하는 문서로 표현한다. 이렇게 함으로써 정보검색은 문서의 형태로 표현된 질의와 유사한 문서를 찾아내는 문제로 파악될 수 있다. 문서간의 유사도 비교방법으로는 그림에서의 수식과 같이 벡터 공간상의 두 벡터의 각도를 계산하는 코사인 유사도(cosine similarity)를 이용하는 방법이 널리 사용되고 있다. 이러한 질의모델링 및 유사도 계산 방법은 기본적으로 질의와 유사한 형태로 색인어들이 분포된 문서를 선호하게 된다.

질의어의 가중치로 음수값을 취할 수 있게 하여 해당 질의어가 가능한 등장하지 않는 문서를 선호하도록 표현할 수 있다. 보다 복잡하고 정교한 질의를 지원하기 위하여 AND, OR, NOT 등의 논리연산자(boolean operator)나 NEAR등의 인접연산자(proximity operator)를 허용하는 검색시스템도 상당수 있다. 이러한 질의에서의 논리연산자는 문서의 선택여부를 결정하는데 사용하고, NEAR 등의 인접연산자는 유사정도를 계산할 때 관여하게 된다.

2.3 적합성 피드백

일반적으로 사용자들이 질의를 만들어 필요로 하는 정보를 검색하는 경우, 습관적으로 자신이 찾고자 하는 정보와 관련된 소수의 단어만으로 질의를 표현하는 경우가 태반이다. 이에 비해 검색엔진에는 다양한 분야에 대하여 대규모의 문서집합을 보유하고 있음으로 해서, 해당 질의와 조금이라도 연관이 있는 문서 또한 상당한 분량이 된다. 이러한 상황에서는 검색결과문서 중 질의와 높은 유사도를 가졌다고 시스템이 평가한 문서라고 해서 사용자가 요구하는 정보에 근접한다고 말하기가 어렵다. 즉, 적은 개수의 질의어로 질의가 표현됨으로써, 상대적으로 사용자가 원하는 정보와 그렇지 못한 정보를 구분하는 변별력이 떨어지게 되는 것이다. 이는 곧 사용자에게는 검색이 정확하지 않은 것처럼 보여지게 되며, 다양한 방법을 통하여 이를 보완하는 연구가 꾸준히 진행되어 왔다(Salton and Buckely, 1990, Kleinberg, 1999).

적합성 피드백(relevance feedback)(Kleinberg, 1999)은 이러한 정확도를 개선하기 위한 방법중의 하나로, 검색된 문서 중에서 사용자가 문서의 제목이나 요약 또는 직접 살펴본 후 적절 또는 부적절하다고 판단되는 문서를 선정하면, 시스템이 이들 문서를 반영하여 재검색함으로써 사용자의 의도에 보다 충실한 검색결과에 얻고자 하는 방법이다. 초기 검색결과 중 높은 점수를 받은 문서들은 상대적으로 사용자 요구에 어

느 정도 부합하는 결과일 것이라는 가정하에 시스템이 자동적으로 이러한 재검색과정을 수행할 수도 있다. 이러한 적합성 피드백은 초기검색의 결과 문서들에 포함된 색인어들을 사용자 또는 시스템이 평가한 정도에 따라 가중치를 주어 추가함으로써 보다 복잡한 질의로 확장하고 새로이 생성된 질의로 검색을 다시 수행함으로써 구현된다. 적합성 피드백은 기존 검색시스템에 수정을 거의 요하지 않으면서 상당한 정확도를 추가로 얻을 수 있어 정보검색분야에서 널리 활용될 수 있는 방법이다.

본 논문에서 대상으로 하는 병렬정보검색시스템은 이상에서 설명한 문서와 질의를 벡터의 형태로 표현하고, 질의의 정확도를 향상시키기 위해 적합성 피드백 기능을 기본적으로 제공하는 시스템이다.

2.4 정보검색시스템의 작업부하

앞에서 살펴본 정보검색시스템에 질의가 주어졌을 때 발생하는 작업부하는 크게 색인어 역파일에 접근하여 필요한 정보를 읽어내는데 소요되는 디스크 I/O관련 작업과 질의와 문서간의 유사정도를 계산하는 작업이 주된 요소를 이룬다. 이들은 기본적으로 주어진 질의에 의해 검색결과로 나타나는 문서의 수에 비례하여 작업량이 늘어난다. 즉 많은 문서에 나타나는 질의어가 질의에 포함되는 경우 디스크 I/O 및 유사도 계산을 위한 작업량이 그만큼 늘어나게 되며, 정보검색시스템이 보유한 총문서의 수가 많을수록 각 색인어가 등장하는 문서수 역시 이에 비례하여 늘어난다.

대략적인 정보검색시스템의 작업부하를 주요 항목으로 모델링하면 작업량은 $|D| \times |q| \times r$ 에 비례한다고 말할 수 있다. $|D|$는 정보의 양 즉 보유한 문서의 갯수이며, $|q|$는 질의에 나타나는 평균적인 질의어의 갯수이며, r은 사용자의 질의 요청 횟수이다. 서론에서 밝힌 바와 같이 검색분야 특히 인터넷 검색의 경우에는 검색의 대상이 되는 문서 수가 급증하고 있으며, 사용자의 질의 요청도 더욱 빈번해지고 있다. 또한 보다 정확한 검색결과의 도출을 위한 방법들은 기본적으로 검색시스템의 계산 부하를 상당히 가중시킨다. 본 연구에서 채택하고 있는 적합성 피드백과 같은 질의 확장기법의 경우 처음 사용자가 제시한 질의보다 수배 내지 수십배 많은 질의어를 가진 새로운 질의를 처리해야 하므로, 검색시스템의 부하는 매우 높아지게 된다. 비록 컴퓨터 하드웨어의 가격은 지속적으로 하락하고 이에 비해 성능은 계속 높아지지만, 대단위의 문서를 대상으로 하는 검색 분야에서는 하나의 소형 시스템으로 감당할 수 있는 그 한계가 있음은 분명하다. PC 클러스터를 활용한 병렬정보검색시스템은 이러한 지속적인 문서의 추가 유입과 사용자의 검색요구의 증가, 그리고 보다 정확도가 높은 검색결과를 제공하기 위하여 늘어날 수 밖에 없는 작업량을 기존 클러스터에 추가의 PC를 편입하는 방법으로, 질의응답시간을 적절한 수준으로 계속 유지할 수 있는 현실적인 해결방법의 하나임은 틀림이 없다.

계속해서 3장에서는 이러한 PC 클러스터 상의 병렬정보검색시스템의 효율을 최대한 활용하기위한 색인어 역파일 분산저장기법에 대하여 설명한다.

3. 색인어 역파일의 분산저장

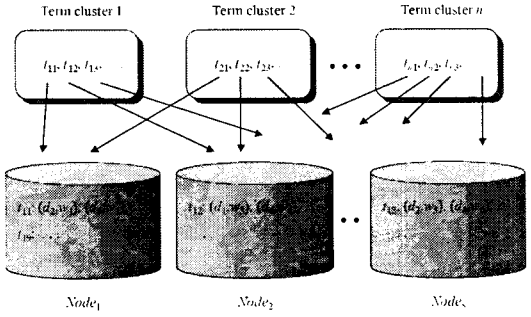
정보검색의 병렬처리가 효율적으로 이루어지려면 색인어 역파일이 각 프로세싱 노드에 적절히 잘 분산저장되어 있어야 한다. 그래야 임의의 입력 질의에 포함되어 있는 색인어들의 처리가 최대한 병렬로 이루어지게 된다. 질의가 입력되면 그 질의에 포함된 색인어들이 등장하는 문서 및 각 등장 문서 내에서의 가중치 정보를 가져오기 위해 색인어 역파일을 디스크로부터 읽게 되는데, 만약 색인어 역파일의 분산저장이 잘못되어 한 노드에 관련 정보가 편중되어 있다면 병렬처리의 효과를 보지 못하게 된다. 특히, 검색의 정확도를 향상시키기 위하여 적합성 피드백에 의한 질의 확장 기법에서처럼 1차로 질의 처리 후 찾아 낸 문서들 중 관련성이 높은 문서를 선택하여 그 문서에 등장하는 주요 색인어들을 질의에 포함시켜 재차 검색을 시도할 경우 색인어의 수가 많아져서 병렬처리의 필요성이 매우 높아지며 이 때 병렬도의 개선 문제는 더욱 중요한 과제가 된다.

따라서, 본 연구에서는 개별 질의 처리 시 부하가 최대한 균등화될 수 있도록 색인어 역파일을 적절히 분산시키는 방안을 강구한다. PC 클러스터는 각 노드마다 하나씩의 프로세서와 하드디스크가 있는 구조로서 각 프로세서는 자신의 하드디스크로부터 읽은 정보의 처리를 우선적으로 담당하게 된다. 그러므로, 각 하드디스크에 색인어 역파일을 적절히 분산저장하는 것은 곧 디스크 I/O의 분산 뿐 아니라 프로세서에 대한 초기 부하 할당의 균등화까지 이루게 됨을 의미한다. 물론, 색인어 역파일의 군집화 및 그에 따른 분산저장은 특정 질의만이 아닌 임의의 질의에 대해서도 질의 처리 시 부하 균등화가 최대한 이루어질 수 있는 방향으로 최적화되어야 한다.

3.1 색인어 군집화 및 분산저장

우리가 원하는 것은 한 질의 내에 포함된 색인어들이 최대한 골고루 분산처리 될 수 있도록 색인어 역파일을 색인어 기준으로 미리 분산저

장해 두는 것이다. 그러려면 결국 같은 질의 내에 등장할 가능성이 높은 색인어들이 최대한 서로 다른 노드에 저장되어 있도록 하면 된다. 그러나, 임의의 색인어들에 대해 같은 질의 내에 동시에 나타날 확률이 과연 얼마일지 직접적으로 알아내기는 어렵다. 다만, 간접적으로 같은 문서에 자주 동시에 나타나는 색인어들이 같은 질의에 동시 등장할 가능성도 높다는 추정은 가능하다. 특히 앞에서 설명한 적합성 피드백에 의한 질의 확장 시 이러한 상관관계의 존재는 거의 의심의 여지가 없다고 하겠다. 따라서, 본 연구에서는 같은 문서에 동시에 등장하는 빈도수가 높은 색인어들이 최대한 서로 다른 노드에 저장되도록 아래의 [그림 3]에서 보인 바와 같은 색인어 군집화를 이용한 분산저장 방안을 강구하게 되었다.



[그림 3] 색인어 클러스터링 및 분산저장

일단 색인어들을 동시등장 가능성이 높은 것들끼리의 군집으로(Term cluster 1 ~ n) 묶고 나서, [그림 3]에서와 같이 각 색인어 군집별로 그 군집 내의 색인어들을 전체 프로세싱 노드에 분산 저장함으로써, 같은 군집에 속하는 색인어들이 서로 최대한 다른 노드에 저장되도록 할 수 있다. 이렇게 하면 어떤 질의가 입력될 때 그 질의에 포함된 색인어들은 [그림 3]의 색인어 군집 중 하나에 속하는 것들일 가능성이 크고, 따라서 그 색인어들이 여러 노드에 분산되어 있을 가능성 또한 커지게 된다.

각 노드별 색인어 역파일은 그 노드에 할당된 색인어들과 관련된 정보들만 원래의 색인어 역파일로부터 추출함으로써 구성된다. [그림 3]에서 각 노드에 표시된 내용은 이상의 과정을 거쳐 만들어진 노드별 색인어 역파일을 보여주고 있다. 그림의 예에서 $Node_1$의 저장 내용 중 t_{11}: (d_2, w_1), (d_4, w_2), . . . 은 색인어 역파일의 색인어 t_{11} 항목을 보인 것으로서 색인어 t_{11}이 문서 d_2와 d_4에 각각 w_1과 w_2의 가중치로 등장함을 기록하고 있는 것이다.

3.2 동시등장 가중치 행렬

이상 설명한 방안이 얼마나 부하균등화에 효과가 있을 것인가 하는 것은 색인어들을 과연 어떤 방법으로 적절한 군집으로 묶을 수 있는가에 달려 있다. 색인어 군집화의 기준으로서 임의의 두 색인어가 한 질의에 동시에 등장할 확률이 얼마인지를 추정할 수 있으려면, 상당수의 과거 검색 요청 질의들을 모아 둔 기록이 필요하지만 대개 이러한 기록을 찾기란 쉬운 일이 아니다. 그러나, 한 질의 내에 어떤 두 색인어가 동시에 등장할 가능성을 간접적으로 추정할 수 있는 대안으로서 우리는 대용량 말뭉치를 분석하여 두 색인어가 하나의 문서 혹은 한 문장 내에 동시에 등장하는 빈도수가 얼마인지를 조사해 볼 수는 있다. 사용자의 검색요청 질의에 여러 색인어가 제시되는 것은 그것들이 동시에 포함되어 있는 문서를 선호함을 의미하는 것으로 간주되기 때문이다.

	t_1	t_2	t_3	t_4	⋯	t_n
t_1	f_1	2/0	0	0	⋯	
t_2	2/0	f_2	3/1	4/2		
t_3	0	3/1	f_3	0		
t_4	0	4/2	0	f_4		
⋮	⋮					
t_n						f_n

[그림 4] 색인어들의 동시 등장 빈도 행렬의 예

[그림 4]는 말뭉치의 분석 결과 두 색인어가 한 문서 혹은 한 문장 내에 동시에 등장한 빈도수를 행렬로 보인 예이다. 이 행렬에서 i번째 행의 j번째 열에 있는 내용은 색인어 t_i와 t_j가 한 문서 및 한 문장 내에 동시 등장한 빈도수이다. 예를 들어 3번째 행의 2번째 열의 내용인 3 / 1은 t_3와 t_2가 같은 문서 내에 3번 그리고 같은 문장 내에 1번 나타났음을 의미한다. 실제로 이들 빈도수는 문서의 크기나 문장의 길이를 감안하여 정규화된 값으로 대치하여 사용할 수도 있다. 이 행렬은 대칭행렬이고 대각원소(diagonal element) f_i의 값은 t_i가 말뭉치 내에 등장한 총 횟수가 된다. 주의할 것은 t_i와 t_j가 동시 등장하고 t_j와 t_k가 동시 등장한다고 하더라도 반드시 t_i와 t_k가 동시 등장하는 것은 아니라는 점이다. 이는 동시에 등장한 문서 혹은 문장이 서로 다를 수 있기 때문이다.

[그림 4]의 행렬을 그래프로 표현할 수 있는데, 이 경우 각 노드는 색인어를 나타내고 두 노드간의 에지는 해당 두 색인어가 동시에 등장한 경우가 있음을 표시한다. 각 에지에는 동시등장 빈도수에 비례하는 가중치(w_{ij})가 부여되는데, 이는 문서 내의 동시 등장 빈도수와 문장 내의 동시 등장 빈도수를 모두 반영함으로써 계산된다. 다만, 문장 내 동시 등장의 경우는 문서 내 동시 등장에 비해 색인어들의 상호관련성이 훨씬 높음을 의미하므로 그 반영 비율을 보다 높게 조정한다. 이렇게 계산된 에지의 가중치는 에지에 연결된 두 색인어의 상호관련성에 비례하는 연결 강도로 간주할 수 있다.

본 연구에서는 동시등장 빈도 행렬을 그대로 사용하여 색인어들을 군집화하는 대신 각 색인어가 어떤 문서에 나타날 때 그 색인어가 그 문서에서 차지하는 중요도를 반영하는 *tf×idf* 값을 가공하여 새로이 동시등장 가중치 행렬을 만들어 사용하였다. *tf×idf*는 특히 적합성 피드백과 같은 질의확장기법에서 질의에 추가될 색인어의 선정 또는 가중치 결정에 주요한 역할을 담당하므로, 동시등장 빈도에 비해 색인어가 질의에 함께 나타날 확률을 보다 정확하게 반영할 수 있다. 동시등장 가중치 행렬의 각 엔트리 C_{ij}(색인어 t_i와 t_j의 동시등장 가중치)는 다음의 식에 의해 구해진다.

$$C_{ij} = \sum_{k \in D_{ij}} w_{ki} \cdot w_{kj}$$

여기서 각 기호의 의미는 다음과 같다.
D_{ij}: 색인어 t_i, t_j가 동시등장하는 문서의 집합
w_{ki}: 문서 d_k에서의 색인어 t_i의 *tf×idf* 값
w_{kj}: 문서 d_k에서의 색인어 t_j의 *tf×idf* 값

4. 동시등장 가중치 기반의 색인어 군집화 및 분산저장 방법

4.1 동시등장 가중치 기반의 색인어 군집화 알고리즘

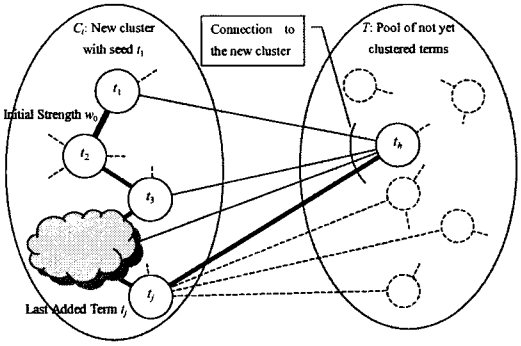
앞장에서 설명한 [그림 3] 방식의 성공의 관건은 결국 색인어 군집화가 얼마나 잘되느냐 하는데 있다. 본 연구에서는 일단 임의의 두 색인어에 대한 동시등장 빈도수를 그 두 색인어를 연결하는 연결강도로 간주하여 강하게 연결된 색인어들끼리 서로 묶이도록 해 보았다. 그러나, 초기 여러 차례 실험을 통해 관찰한 바 군집의 형성이 지극히 불균형한 것으로 관찰되었다. 하나의 매우 큰 규모의(총 색인어의 10%에 해당하는) 군집이 형성됨과 동시에 단 하나의 색인어만으로 이루어진 군집 또한 매우 많이(거의 90%에 육박하게) 생성되었다. 주된 원인은 매우 많은 문서에 등장하는 색인어가 다른 색인어들을 너무 많이 자신의 군집으로 끌어들이는데 있었다. 등장 문서 수가 매우 많은 색인어는 다른 색인어들과 동시에 등장하는 빈도수 또한 높아지는 경향이 있기 때문이다. 그러나 문제는 동시등장 관계(relation)가 transitive하지 못하다는데 있다. 예를 들어 색인어 *a*와 *b*가 자주 동시에 등장하고 색인어 *b*와 *c*가 역시 자주 동시 등장하더라도 *a*와 *c*가 전혀 동시에 등장하지 않을 수 있는 것이다. 서로 동시에 등장하는 문서가 다를 경우 그렇게 된다. 이 경우 군집화 알고리즘이 이들 모두를 서로 하나의 군집으로 묶을 가능성이 높지만 사실 *a*와 *c*가 같이 묶여서는 안되는 것임에 주의할 필요가 있다.

```
Input:
  set of index terms T with term ids form 1 to n
  co-occurrence weight matrix W(n, n)
  strength threshold s
  connectivity threshold c
Output:
  N disjoint clusters C_1, ..., C_N of index terms.
  (N is not predetermined.)

CWC (T, W, s, c)
  i = 1;
  C_i = initCluster (T, W)
  while (T ≠ ∅)
    t_j ← the term most recently added to C_i
    t_h ← t' ∈ T such that w = W(id(t_j), id(t')) is
      maximum among all t' ∈ T
    R = {t | t ∈ C_i and W(id(t), id(t_h)) ≥ s·w_0}
    if ( w ≥ s·w_0 and |R| ≥ c·|C_i| )
      C_i = C_i ∪ {t_h}; T = T - {t_h}
    else
      Output C_i
      i = i + 1
      C_i = initCluster (T, W)

InitCluster (T, W)
  t_s ← the term in T which has the highest
    importance
  C = {t_s}; T = T - {t_s}
  t_h ← t' ∈ T such that w_0 = W(id(t), id(t')) is
    maximum among all t' ∈ T
  C = C ∪ {t_h}; T = T - {t_h}
  return C
```

[그림 5] CWC 알고리즘

따라서, 본 연구에서는 기존의 군집화 알고리즘들과는 달리 transitivity가 성립하지 않는 관계를 기반으로도 휴리스틱하게 군집을 형성할 수 있는 CWC (Co-occurrence Weight-based Clustering) 알고리즘을 새로이 제안하게 되었다. CWC 알고리즘의 특징은 색인어간 연관관계에 transitivity

가 성립하지 않더라도 한 색인어를 어떤 군집에 편입시킬 때 그 색인어가 군집 내에 이미 존재하는 다른 색인어들과 충분한 관련성이 있는지 여부를 조사한다는데 있다. CWC 알고리즘의 상세한 내용은 [그림 5]에 기술되어 있으며, [그림 6]에는 군집을 생성하고 있을 당시의 예를 보이고 있다.



[그림 6] CWC 알고리즘의 동작

CWC 알고리즘은 가장 중요한 (타 색인어와의 연결강도의 합이 가장 높은) 색인어 t_1를 seed로 한 초기 군집에 그 t_1와 가장 동시등장 가중치가 높은 색인어 t_2를 추가하는 것으로 시작한다. 이 때의 동시등장 가중치를 w_0라 한다. 이 후에는 가장 최근에 군집에 편입된 색인어 t_j를 기준으로 하여 아직 군집에 미 편입된 나머지 색인어들 중에서 t_j와의 동시등장 가중치가 제일 높은 색인어 t_h를 찾아서 추가로 편입시키는 방식으로 군집을 확장해 가되, t_j와 t_h의 동시등장 가중치가 적어도 $s \cdot w_0$ 이상이어야 할 뿐 아니라, 또한 t_h는 이미 군집에 소속이 되어 있는 색인어들 중 일정 비율 c 이상의 색인어들과 역시 $s \cdot w_0$ 이상의 동시등장 가중치를 가지는 것을 조건으로 한다. 이런 조건을 부여함으로써 transitivity가 성립하지 않아 생기는 문제를 상당부분 극복할 수 있게 되는 것이다. 여기서, s와 c를 각각 연결강도 및 연결횟수 임계치라 부르며 모두 0과 1 사이의 실수로 그 값을 정해 주어야 의미가 있게 된다. 만약 t_h가 연결강도와 연결횟수 임계치에 관한 조건을 만족시키지 못하게 될 경우에는 현재의 군집은 그 상태에서 더 이상의 확장을 멈추게 되고, 그 대신 새로운 군집을 생성시키기 위해 새로운 seed를 찾은 뒤 다시 마찬가지의 방법을 반복하게 된다.

CWC 알고리즘에 의해 형성되는 첫 번째 군집은 다른 색인어와 동시등장가중치의 합이 가장 높은, 즉 가장 중요하다고 추정되는 색인어를 중심으로 서로간의 동시등장 가중치가 매우 높은 색인어들을 포함하게 된다. 그 다음에 형성되는 군집은 남은 색인어들을 대상으로 다시 그 중 타 색인어와 가장 동시등장가중치의 합이 높은 색인어를 seed로 고른 뒤 이를 중심으로 역시 서로 동시등장 가중치가 높은 색인어들을 찾아 군집에 편입시키게 된다. 이러한 알고리즘의 성격상 군집의 형성이 진행될수록 나중에 생기는 군집에 포함되는 색인어들의 연관관계는 느슨해지게 되지만, 남은 색인어들 중에서는 상대적으로 연관관계 즉 동시등장 가중치가 가장 높은 것들의 모임이 되는 것은 분명하다.

CWC 알고리즘은 실험결과 종래의 방법들처럼 군집의 크기가 편중되지 않고 비교적 고른 형태로 형성됨을 확인할 수 있었다. 다만, 연결강도와 연결횟수 임계치를 적절한 값을 실험적으로 결정해 주어야 하는 부담이 있는 것은 결점이라 할 수 있겠다. 연결강도와 연결횟수 임계치값을 너무 작게 할 경우 개별 군집의 크기가 너무 커져서 사실상 크게 관련성이 없는 색인어들이 한데 묶이게 되고, 반대의 경우에는 개별 군집의 크기는 작아지면서 전체적으로 군집의 개수가 너무 많아지는 문제가 있다.

4.2 군집기반의 색인어 분산저장

색인어의 분산은 종래의 그리디(greedy) 디클러스터링 방법(Chung et al., 2000)을 거의 그대로 따른다. 그리디 디클러스터링은 동시등장 가중치를 기준으로 색인어들을 각 노드에 뿌리는 방법으로서 그 구체적 내용은 다음과 같다. 먼저 색인어들을 등장 문서수의 내림차순으로 정렬한다. 다음에 정렬 결과의 첫 m개(PC 클러스터의 노드 수) 색인어를 각 노드에 차례로 배정한다. 그 다음 m개의 색인어에 대해서는 차례로 이미 배정된 색인어와의 동시등장 가중치가 가장 낮은 노드로 배정한다. 이 후 같은 방법으로 m개씩 모든 색인어들의 배정이 완료될 때까지 계속한다. 여기서, 색인어들을 등장 문서수의 내림차순으로 정렬부터 하는 이유는 등장 문서수가 많을수록 색인어 역파일 내용이 길고 이들의 디스크 I/O 부담 또한 크므로, 디스크 I/O의 관점에서 비중이 엇비슷한 것들 사이에 서로 동시등장 가중치를 고려하여 분산저장이 이루어져야 좋은 결과를 얻을 수 있기 때문이다.

CWC 기반의 색인어의 분산저장기법은 그리디 디클러스터링과 기본적으로 동일하나 색인어의 배정 순서를 등장 문서수의 내림차순으로 하는 대신 CWC의 결과로 형성된 군집의 순서대로 하는 것이 다르다. 각 군집 내에서의 색인어 처리 순서는 군집에 편입된 순서를 따른다. 이렇게 함으로써 동시등장 가중치가 높은 색인어들이 서로 다른 노드에 배정될 수 있게 되는 것이다.

<표 1> 50만 건 문서에 대한 색인어 분산저장 방법의 비교 실험 결과

	Random	Greedy	CWC 기반 분산저장 (strength/connectivity)			
			40/50	40/60	50/60	50/80
검색시간 (초)	3081	2923	2839	2844	2862	2824
향상도 (%)	0.0	5.1	7.9	7.7	7.1	8.4

Random (무작위 분산저장 방안), Greedy (그리디 분산저장 방안)

5. 실험 결과

이상에서 설명한 색인어 군집화기법을 활용한 분산저장 방안의 효과를 검증하기 위해 일련의 실험을 수행한 결과를 이 장에서 정리하였다. 실험을 위한 병렬 컴퓨팅 환경으로는 8대의 PC를 80MBps(mega bytes per second)의 SCI(Scalable Coherent Interface) 기반의 고속 네트워크로 연결한 PC 클러스터 시스템을 사용하였다. 실험 대상의 대용량 말뭉치로는 5년 간의 신문기사 약 50만 건의 모음을 사용하였다. 실험에는 각각 24개의 색인어를 가진 5,000개의 질의가 사용되었고 이들을 이용하여 병렬 검색을 실시한 결과 검색에 소요된 누적 총 시간을 측정하여 비교하였다. 하나의 질의를 생성하는데는 적합성 피드백을 가정하여 임의로 96개 이상의 색인어를 가진 문서를 선정한 뒤 그 문서에서 *tf×idf* 값의 상위 순으로 24개의 색인어를 선택하는 방법을 사용하였다.

먼저 CWC 알고리즘에서 적절한 연결강도 및 연결횟수 임계치를 선정하기 위하여 50만 건 말뭉치로부터 무작위로 추출한 약 5만 건의 문서를 대상으로 색인어 역파일을 만들어 실험하였다. 문서의 규모를 줄임으로써 개별 실험에 소요되는 시간이 단축되어 보다 다양한 실험을 해볼 수 있었다. 아래의 50만건 말뭉치를 이용한 실험은 이 5만건 문서를 대상으로 한 실험에서 가장 성능이 좋았던 4가지 파라메터를 대상으로 실험하였다.

50만건 문서를 대상으로 성능을 측정한 <표 1>의 실험 결과에서 향상도는 비교의 기준이 되는 무작위 분산저장 방식에 대해 총 질의 수행시간이 단축된 정도를 백분율로 표시한 것이다. 임계치 설정에 관한 수치값 40, 50 등의 단위도 역시 %이다. 색인어의 군집화는 말뭉치 내에 등장하는 색인어들 중 일부만을 대상으로 하였다. 본 실험의 경우 50만 건 문서에 등장하는 총 색인어의 수는 약 470만 개이지만 이들 중 5개 이상의 문서에 등장하는 색인어 약 56만 개만을 대상으로 동시등장 가중치 행렬을 구성하고 CWC 알고리즘으로 군집화한 후 앞의 4.2절에서 제시한 색인어 분산저장 방법으로 8개의 노드에 분배하였다. 실제로 등장빈도가 지나치게 낮은 나머지 색인어들은 비록 그 수는 많으나 질의에 등장할 가능성이 매우 낮으므로 모두 무작위로 각 노드에 배정하였다. 실험결과 종래의 그리디 디클러스터링 방식에 비해 CWC 기반의 저장방식이 뚜렷한 성능의 향상을 보였다. 그러나, 앞의 4장에서도 지적하였듯이 연결강도 및 연결횟수 임계치의 설정에 따라 성능의 차이가 상당하다는 문제점도 실험적으로 확인되었다

6. 결론 및 향후과제

본 논문에서는 PC 클러스터 기반의 병렬 정보검색 시스템의 효율을 향상시키기 위하여 색인어 역파일을 PC 클러스터의 각 노드에 분산저장하는 기법을 제시하였다. 부하 균등화를 통한 병렬도의 향상을 위해서는 한 질의 내에 동시에 등장할 가능성이 높은 색인어들이 가능한 서로 다른 노드에 저장될 필요가 있다. 특히 적합성 피드백에 의한 질의 확장 시 질의와 관련성이 높은 문서를 선택하여 그 문서에 등장하는 주요 색인어들을 질의에 포함시켜서 재차 검색을 시도할 경우, 색인어의 수가 많아져서 병렬처리의 필요성이 매우 높아지며 이 때 병렬도의 개선 문제는 더욱 중요한 과제가 된다.

본 연구에서는 색인어들이 어떤 문서에서 얼마만큼의 중요도를 가지고 얼마나 동시에 등장하는지를 대량의 말뭉치를 분석하여 작성한 색인어 동시등장 가중치 행렬을 기반으로, 관련성이 높은 색인어들을 군집화하는 CWC 알고리즘을 제시하고, 이를 이용하여 색인어들을 기존의 그리디 디클러스팅과 유사한 방식으로 각 노드에 분산저장하는 방안을 소개하였다. 실험적인 시스템으로는 비교적 대규모라 할 수 있는 50만 건 말뭉치를 대상으로 한 실험결과 본 연구에서 제안한 방식이 기존의 방식보다 좋은 성능을 보여 충분한 실용성이 있음을 확인하였다.

향후 병렬정보검색시스템을 실용적으로 활용하기 위해서 색인어분산저장의 측변에서 고장포용성을 제공하는 방안에 대한 연구와, 지속적인 추가 문서의 유입 및 변화하는 질의에 대응하여 기존 분산저장구조를 저비용으로 유지관리할 수 있는 방법에 대한 연구가 추가적으로 요구된다.

참고문헌

강유경, 류광렬, 정상화, "문서 클러스터링에 의한 효율적인 병렬 정보검색 시스템," 정보과학회논문지 : 소프트웨어 및 응용, 제28권 제2호, pp. 157-167, 2001.

Chung, S-H., Kwon, H-C., Ryu, K. R., Jang, H-K., Kim, J-H. and Choi, C-A., "Parallel Information Retrieval on an SCI-Based PC-NOW," Lecture Notes in Computer Science Vol. 1800, (IPDPS-2000 Workshops, Cancun, Mexico) pp. 81-90, 2000.

Frakes, W. B. and Baeza-Yates, R., Information Retrieval: Data Structures and Algorithms, Prentice-Hall, North Virginia, 1992

Jeong, B. and Omiecinski, E., "Inverted File Partitioning Schemes in Multiple Disk Systems," IEEE Transactions on Parallel and Distributed Systems, 6(2):142-153, 1995

Kleinberg, J. M., "Authoritative Sources in a Hyperlinked Environment", Journal of the ACM, 46(5):604-632, 1999

Lin, Z., and Zhou, S. "Parallelizing I/O intensive applications for a workstation cluster: a case study,". Computer Architecture News 21, 5, pp. 15-22., 1993

Salton, G. and Buckely, C., "Improving retrieval performance by relevance feedback", Journal of the American Society for Information Science 41:288-297, 1990

Samanta, R., Zheng, J., Funkhouser, T., Li, K. and Singh, J. P., "Load Balancing for Multi-Projector Rendering Systems," SIGGRAPH/Eurographics Workshop on Graphics Hardware, August, 1999

Schutze, H. and Silverstein, C., "Projections for Efficient Document Clustering," Proceedings of The 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.74-81, 1997

Silverstein, C. and Pedersen, J. O., "Almost-Constant-Time Clustering of Arbitrary Corpus Subsets," Proceedings of The 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 60-66, Philadelphia, Pennsylvania, 1997.

Stanfill, C. and Thau, R., "Information Retrieval on the Connection Machine : 1 to 8192 Gigabytes," Information Processing & Management, pp. 285-310, 1991