효율적인 병렬정보검색을 위한 색인어 군집화 및 분산저장 기법

(Term Clustering and Duplicate Distribution for Efficient Parallel Information Retrieval)

강 재 호 † 양 재 완 †† 정 성 원 ††† 류 광 렬 †††† 권 혁 철 †††† 정 상 화 ††††

(Jaeho Kang) (Jae-Wan Yang) (Sung-Won Jung) (Kwang Ryel Ryu) (Hyuk-Chul 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 분배함으로써 보다 높은 수준의 병렬화를 달성할 수 있는 방안을 제시한다. 또한 일부 PC의 결함 또는 유지보수 등의 원인에 의한 서비스 중지상황에도 적극적으로 대처하기 위하여 색인어 역파일을 중복되게 분산저장하는 기법을 제안한다. 대용량 말뭉치를 활용한 실험결과 본 논문에서 제시하는 분산 및 중복저장기법이 충분한 효율성과 실용성이 있음을 확인하였다.

키워드 : 병렬정보검색, 색인어 클러스터링, PC 클러스터, 결함포용

Abstract The PC cluster architecture is considered as a cost-effective alternative to the existing supercomputers for realizing a high-performance information retrieval (IR) system. To implement an efficient IR system on a PC cluster, it is essential to achieve maximum parallelism by having the data appropriately distributed to the local hard disks of the PCs in such a way that the disk I/O and the subsequent computation are distributed as evenly as possible to all the PCs. If the terms in the inverted index file can be classified to closely related clusters, the parallelism can be maximized by distributing them to the PCs in an interleaved manner. One of the goals of this research is the development of methods for automatically clustering the terms based on the likelihood of the terms' co-occurrence in the same query. Also, in this paper, we propose a method for duplicate distribution of inverted index records among the PCs to achieve fault tolerance as well as dynamic load balancing. Experiments with a large corpus revealed the efficiency and effectiveness of our method.

Keyword : parallel information retrieval, term clustering, PC cluster, fault tolerance

· 정보통신부지원 대학기초연구지원사업(정보통신기초기술연구지원사업)으로 부산대학교 컴퓨터 및 정보통신연구소에서 수행하였음

† 비 회 원 : 동아대학교 지능형통합방관리연구센터
jhkang@pusan.ac.kr
†† 비 회 원 : 온빛시스템 정보기술연구원
jwyang@onbit.com
††† 비 회 원 : 부산대학교 전자계산학과
swjung@pusan.ac.kr
†††† 종신회원 : 부산대학교 정보컴퓨터 공학부 교수
krryu@pusan.ac.kr
hckwon@pusan.ac.kr
shchung@pusan.ac.kr

논문접수 : 2002년 4월 22일
심사완료 : 2002년 11월 18일

1. 서 론

최근 인터넷의 보급이 급격히 확대됨에 따라 정보검색 시스템이 처리해야 하는 정보의 양과 사용자의 검색 요구는 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, 이

러한 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[1, 2].

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[3]을 들 수 있으나, 이는 기본적으로 *Connection Machine*을 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 [4]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 역파일을 색인어 단위로 분할 시, 질의에 나타날 색인어의 확률을 고려하면 상당한 효과가 있음을 확인하였다. 이 후 문서별 분할과 색인어별 역파일 분할을 함께 고려하여 보다 개선한 연구로 [5]이 있으나, 이러한 방법들은 분할의 기본 요소인 색인어 또는 문서에 의해서 발생하는 작업부하를 독립적으로 가정하여, 현실적으로는 하나의 질의처리과정에서 특정 노드에 작업량이 편중될 수 있는 상황을 보다 면밀하게 반영하지 못한다는 한계를 가진다.

병렬 정보검색의 효율 향상을 목적으로 저장방식을 제시한 또 다른 연구로는 [6]가 있는데, 이 연구에서는 문서를 분류(classification)하는 방법을 사용하여 유사한 문서들을 군집으로 묶고, 생성된 군집에 대한 색인어 역파일 구조를 추가로 도입한 계층적인 검색방법을 제안하였다. 군집단위의 색인어 역파일을 활용하여 검색대상문서를 여과함으로써 병렬 정보검색의 효율을 향상시켰으나, 수작업으로 학습데이타를 준비하여야 한다는 부담과 정확도와 재현율 측면에서 기존 정보검색시스템과는 차이가 있을 수 있으므로, 일반적인 정보검색시스템에 손쉽게 적용될 수 있는 방법이라고 하기는 어렵다.

PC 클러스터 기반의 병렬 정보검색 시스템을 제안하면서 색인어 역파일의 효과적인 분산저장 방식을 소개한 최근의 연구로 [7]이 있다. 이 연구에서는 색인어간의 연관관계를 기반으로 디클러스터링(declustering) 기법을 이용하여 색인어 역파일을 분산저장함으로써 무작위적 분산저장보다 성능향상을 이룰 수 있음을 보인 바 있다. 본 논문에서는 검색대상 자료인 색인어 역파일을 효과적으로 분산저장하기 위해서 색인어를 먼저 질의에 동시에 나타날 가능성이 높은 것들끼리 묶어 군집화(clustering)한 후, 각 PC의 하드디스크에 나누어 할당함으로써 병렬처리의 효율을 향상시키는 방안을 제시하고 있다. 정보검색 분야에서 유사한 군집으로 분류하는 군집화기법들을[8, 9] 이용한 연구가 상당히 진척되어 있으나, 모두 검색 결과를 문서의 측면에서 정리하여 사용자가 원하는 문서를 보다 쉽게 찾을 수 있도록 하는데 그 초점이 맞추어진 것이지, 본 논문에서 제시하는 바와 같이 병렬검색의 효율 향상을 목표로 한 것은 아니다.

PC 클러스터 기반 병렬정보검색 시스템을 실용적으로 활용하기 위해서는 성능측면에서의 최적화 이외에도, 일부 PC의 하드웨어 결함 또는 소프트웨어 유지보수 등의 원인으로 인한 서비스 중지가 요구되는 상황하에서도 지속적으로 운영할 수 있는 결함포용성이 일정 수준 이상 보장되어야 한다. 최근들어 통신 인프라 및 하드웨어의 결함률이 낮아지는 데 비해, 운영되고 있는 소프트웨어의 문제점 해결 또는 추가 서비스의 개발과 같은 시스템의 유지보수 요구가 더 빈번한 상황하에서는 이러한 결함포용성의 중요성은 더욱 커진다고 할 수 있다. 데이타베이스분야에서는 초기부터 이러한 연구가 활발히 진행하여 왔으나[10, 11], 병렬정보의 특성을 반영하여 이러한 문제에 접근한 연구는 아직 미흡한 실정이다. 본 논문에서는 실용적인 수준의 결함포용성을 제공하기 위하여 색인어 역파일을 여러 PC에 중복하여 저장하는 방안을 분산저장방안과 연계함으로써 실용적이면서 효율적인 병렬정보검색시스템을 위한 색인어 역파일 분산저장기법을 제안하고자 한다.

약 50만 건의 신문기사들로 구성된 말뭉치를 활용한 실험 결과 색인어 군집화 및 분산저장 기법에 의해 사용자 질의에 포함되어 있는 검색어들을 각 PC에서 최

대한 병렬로 처리될 수 있도록 함으로써 단순한 색인어 분산저장 방식보다 검색 성능을 더욱 향상시킬 수 있음을 확인하였으며, 중복저장기법과 병용하여 분산저장한 실험에서는 의도하였던 일부노드의 정지 상황하에서의 안정적인 성능제공과 더불어, 색인어의 중복저장에 의한 병렬검색작업분배의 유연성을 활용함으로써 부하균등화 효과도 충분함을 확인하였다.

본 논문의 구성은 먼저 2장에서는 병렬화를 위한 색인어 역파일의 분산저장방안에 대하여 설명한다. 3장에서는 병렬정보검색의 효율을 최적화하기 위하여 본 논문에서 제시하는 동시등장 기중치 기반의 색인어 군집화 방법과 생성된 군집을 이용한 분산저장방안을 소개한다. 이어지는 4장에서는 결함포용을 위한 색인어 역파일 중복저장 기법을 설명한다. 본 논문에서 제시하는 기법들을 활용한 실험결과를 5장에서 제시하여 분석하고, 마지막 6장에서는 결론 및 향후연구과제를 제시한다.

2. 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 피드백(relevance feedback)[12]에 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.

2.1 색인어 군집화 및 분산저장

우리가 원하는 것은 한 질의 내에 포함된 색인어들이 최대한 골고루 분산처리 될 수 있도록 색인어 역파일을 색인어 기준으로 미리 분산저장해 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 1에서 보인 바와 같은 색인어 군집화를 이용한 분산저장 방안을 강구하게 되었다.

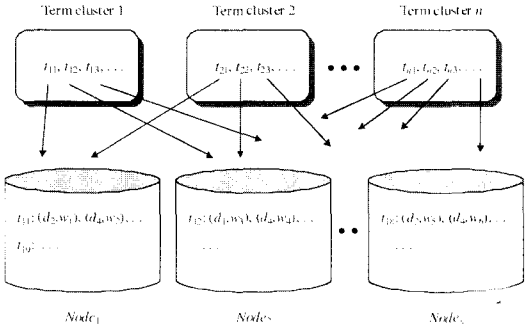

그림 1 색인어 클러스터링 및 분산저장

일단 색인어들을 동시등장 가능성이 높은 것들끼리의 군집으로(Term cluster 1~n) 묶고 나서, 그림 1에서와 같이 각 색인어 군집별로 그 군집 내의 색인어들을 전체 프로세싱 노드에 분산저장함으로써, 같은 군집에 속하는 색인어들이 서로 최대한 다른 노드에 저장되도록 할 수 있다. 이렇게 하면 어떤 질의가 입력될 때 그 질의에 포함된 색인어들은 그림 1의 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 색인어

들과 관련된 정보들만 원래의 색인어 역파일로부터 추출함으로써 구성된다. 그림 1에서 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.

2.2 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 2 색인어들의 동시 등장 빈도 행렬의 예

그림 2는 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 2의 행렬을 그래프로 표현할 수 있는데, 이 경우 각 노드는 색인어를 나타내고 두 노드간의 에지는 해당 두 색인어가 동시에 등장한 경우가 있음을 표시한다. 각 에지에는 동시 등장 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 $tf \times idf$ 값을 가공하여 새로이 동시등장 가중치 행렬을 만들어 사용하였다. $tf \times idf$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_i과 t_j가 동시에 등장하는 문서의 집합

w_{ki} : 문서 d_k에서의 색인어 t_i의 $tf \times idf$ 값

w_{kj} : 문서 d_k에서의 색인어 t_j의 $tf \times idf$ 값

3. 동시등장 가중치 기반의 색인어 군집화 및 분산저장 방법

3.1 동시등장 가중치 기반의 색인어 군집화 알고리즘

앞장에서 설명한 그림 1 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 강하

게 연결된 색인어들끼리 서로 묶이도록 해 보았다. 그러나, 초기 여러 차례 실험을 통해 관찰한 바 군집의 형성이 지극히 불균형한 것으로 나타났다. 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 추이적(transitive)이지 못하다는데 있다. 예를 들어 색인어 a와 b가 자주 동시에 등장하고 색인어 b와 c가 역시 자주 동시 등장하더라도 a와 c가 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 a와 c가 같이 묶여서는 안되는 것임에 주의할 필요가 있다.

```
입력: T: 색인어 집합
      W : n×n 크기의 색인어간 동시등장 가중치 행렬
      s: 연결강도 임계치
      c: 연결횟수 임계치
출력: 색인어로 클러스터 C1, C2, . . . , CN
      (하나의 색인어는 하나의 클러스터에만 포함)
      (클러스터 수 N은 미리 지정되지 않음)

CWC (T, W, s, c)
    i = 1;
    Ci = initCluster (T, W)
    while (T≠∅)
        tj ← 클러스터 Ci에 가장 최근에 포함된 색인어
        th ← t'∈T(W(id(tj), id(t'))값이 가장 큰 색인어)
        w = W(id(tj), id(th))
        R = {t | t∈Ci∧W(id(t), id(th))≥S·w0}
        if ( w≥s·w0∧|R|≥c·|Ci| )
            Ci = Ci∪{th}; T = T - {th}
    else
        클러스터 Ci 출력
        i = i + 1
        Ci = initCluster (T, W)

initCluster (T, W)
    ts ← T에 포함된 색인어 중 가장 중요도가 높은(Σ
    W(ts, tk) 값이 최대인) 색인어
        C = {ts}; T = T - {ts}
        th ← t'∈T(W(id(ts), id(t'))값이 가장 큰 색인어)
        w0 = W(id(ts), id(th))
        C = C∪{th}; T = T-{th}
        return C
```

그림 3 CWC 알고리즘

따라서, 본 연구에서는 기존의 군집화 알고리즘들과는 달리 추이성(transitivity)이 성립하지 않는 관계를 기반으로도 휴리스틱하게 군집을 형성할 수 있는 CWC(Co-occurrence Weight-based Clustering) 알고리즘을 새로이 제안하게 되었다. CWC 알고리즘의 특징은 색인어간 연관관계에 추이성이 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 3에 기술되어 있으며, 그림 4에는 군집을 생성하고 있을 당시의 예를 보이고 있다.

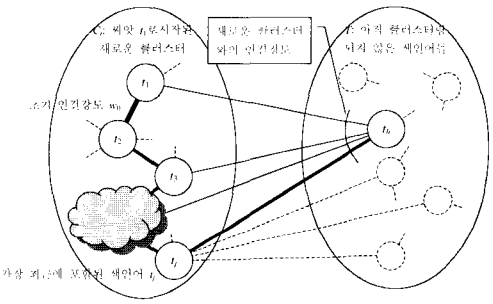

그림 4 CWC 알고리즘의 동작

CWC 알고리즘은 가장 중요한 (타 색인어와의 연결강도의 합이 가장 높은) 색인어 t_1를 씨앗(seed)로 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 추이성이 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 씨앗을 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 씨앗으로 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 임계치의 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.

3.2 군집기반의 색인어 분산저장

색인어의 분산은 종래의 그리디(greedy) 디클러스터링 방법[7]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.

4. 결함포용 및 부하균등화를 위한 색인어의 중복저장

이상과 같은 병렬 정보 검색 시스템을 실용적으로 활용하기 위해서는 이러한 효율성 증대를 위한 노력과 더불어 서비스 제공 중에 발생할 수 있는 노드의 결함이나 유지관리를 위해 일부 노드를 정지하여야 하는 상황에도 적극적으로 대비할 필요가 있다. 특히 본 시스템과 같이 빠른 응답시간과 고성능을 달성하기 위하여 클러스터에 포함된 모든 노드가 전체 작업에 참여하는 병렬 시스템의 경우, 노드 하나의 결함이 전체 시스템의 운영 중지로 이어지므로 이러한 결함포용성에 대한 중요성이 더욱 커진다.

일부 노드의 사용이 불가능한 경우에도 전체 시스템의 운영이 가능하게 하는 가장 적극적인 대처방법은 전체 시스템을 복제하여 운영하는 방법이나, 이는 그 효율에 비하여 추가 비용이 과다하게 발생하므로, 본 논문에서는 작업분배의 직접 대상이 되는 색인어 역파일을 중복저장함으로써 시스템의 효율성과 결함포용성을 동시에 달성하고자 하였다. 병렬 정보검색 시스템에서 개별 색인어 관련 역파일 엔트리 정보를 2개 이상의 서로 다른 노드에 중복하여 저장한다면 특정 노드 결함시에도 운영이 가능하므로, 본 논문에서는 저장 공간상의 요구가 가장 낮은 2 노드 색인어 역파일 엔트리 중복저장방안을 연구하였다.

4.1 색인어 중복저장 방안

하나의 색인어 관련 역파일 정보를 2개의 서로 다른 노드에 중복저장할 때, 첫번째 노드를 1차노드, 두번째 노드를 2차노드라고 부르기로 한다. 1차노드는 시스템의 효율성을 위하여 앞 장에서의 색인어 분산저장방법을 이용하였으며, 2차노드는 효율성과 결함포용성 측면에서 여러 방안을 고안하여 실험하였다. 구체적인 2차노드의 선정방식은 1차노드와 무관한 임의의 노드에 중복저장하는 경우와 1차노드의 단방향 이웃노드에 중복저장하는 두가지 방안을 연구하였다. 임의노드 중복저장 방안은 1차노드의 색인어 분배와는 완전히 독립된 방식으로 색인어를 중복저장하므로, 하나의 노드 결함시에 파급되는 부하의 불균형이 전반적으로 분산된다는 장점이 있으며, 단방향 이웃노드 중복저장 방안은 이웃하지 않은 2개 이상의 노드가 동시에 문제가 발생하더라도 전체

시스템 운영이 가능하다는 장점을 가지고 있다. 계속해서 임의노드 중복저장 방안 및 단방향 이웃노드 중복저장 방안의 구체적인 방법과 노드 결함 시 대응책을 설명한다.

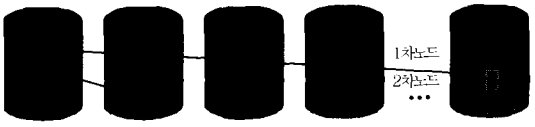

그림 5 임의노드 중복저장 방안의 저장 방법

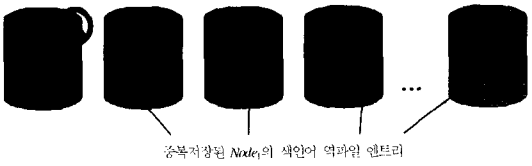

그림 6 임의노드 중복저장 방안의 노드 결함 시 대처 방법

그림 5는 임의노드 중복저장 방안의 저장방법을 보여주고 있다. 각각의 노드에 1차노드로 저장된 색인어들은 해당 1차노드를 제외한 임의의 노드에 중복되어 저장되고 있다. 예를 들어 색인어 t_1의 경우 1차노드로 $Node_1$, 2차노드로는 $Node_2$가 선정되었으며, 색인어 t_2의 경우에는 1차노드로 $Node_2$, 2차노드로는 $Node_8$이 사용되었다. 그림 6에서는 임의노드 중복저장 방안의 노드 결함 시 대처방법을 보여주고 있는데 $Node_1$에 문제가 발생한 경우를 가정하였다. $Node_1$에 있던 색인어 t_1, t_9, t_{17}, t_{25}는 각각 $Node_2$, $Node_8$, $Node_3$, $Node_4$에 중복저장된 색인어를 활용함으로써 전체시스템의 지속적인 운영이 가능하다. 임의노드 중복저장 방안은 노드 하나의 결함 시 해당 노드에 할당되어야 하는 작업량이 타 노드로 고르게 분배될 수 있는 반면에, 2개 이상의 노드 결함 시에는 시스템의 정상운영을 보장하지 못한다는 단점이 있다.

그림 7은 단방향 이웃노드 중복저장방안의 저장방식을 보여주고 있다. $Node_1$이 1차노드로 선정된 t_1, t_9, t_{17}, t_{25}는 모두 이웃하는 $Node_2$를 2차노드로 하여 중복저장되며, $Node_3$의 t_3, t_{11}, t_{19}, t_{27}은 $Node_4$에 중복저장된다. 그림 8에는 단방향 이웃노드 중복저장방안 적용시 노드에 문제가 발생한 경우의 대처방법을 설명한다. 이 예에서는 $Node_1$과 $Node_3$이 동시에 결함이 발생한 경우를 가정하였다. 서로 이웃하지 않은 노드들에 문제가 발생하였으므로 각각 $Node_2$, $Node_4$를 활용하여 전체시스템의 지속적인 운영이 가능하다.

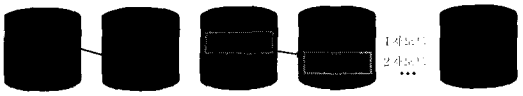

그림 7 단방향 이웃노드 중복저장 방안의 저장방법

그림 8 단방향 이웃노드 중복저장 방안의 2개 노드 동시 결함 시 대처 방법

단방향 이웃노드 중복저장 기법을 활용한 결함포용시스템의 커다란 잇점 중의 하나는 최대 절반의 노드가 멈춘 상황에서도 지속적인 서비스를 제공할 수 있으므로, 시스템 하드웨어 및 소프트웨어 업그레이드 등 부득이한 상황에서도 서비스를 중지하지 않고 빠른 시간내에 본래 시스템의 성능으로의 환원이 가능하다는 점이다. 절반의 노드를 정지시킨 경우에도 정상 시스템의 50%의 성능을 보장하므로 서비스 요구가 낮은 시간대를 선택하여 시스템의 유지보수를 원활히 수행할 수 있다. 최근들어 통신 인프라 및 하드웨어의 문제로 인한 운영중지 상황보다는 하드웨어의 확장, 소프트웨어의 수정 또는 추가 서비스의 제공에 의한 서비스 일시 중지 요구가 보다 빈번하게 발생하는 추세이므로 이러한 기능은 더욱 절실히 요구된다고 할 수 있다.

4.2 색인어 중복저장을 이용한 부하균등화

색인어를 중복저장함으로써 얻을 수 있는 결함포용성이라는 장점이외의 또 다른 중요한 이점은, 주어진 질의에 포함된 색인어별로 1차노드와 2차노드 중 작업수행 노드를 선택할 수 있으므로 개별 질의에 대해 적극적인 동적 부하균등화를 수행할 수 있다는 점이다. 앞에서의 색인어간 동시 등장 가중치를 이용한 분산저장 기법은 질의내에 동시에 등장할 색인어들을 확률적인 측면에서 추정하여 노드들간의 전반적인 불균형을 해소시키는 것을 목표로 하므로, 실제로 처리하여야 하는 개별 질의 하나하나 별로 발생할 수 있는 일부 노드에 대한 부하의 집중 현상에 대해서 직접적으로 대처하는 방안은 되지 못한다. 이에 비해, 색인어를 중복하여 분산저장하는 경우에는 하나의 색인어 관련 정보가 여러 노드에 중복저장되어 있으므로, 질의 처리시 작업량이 평균보다 높을 것으로 예상되는 노드에 할당된 작업의 일부를 2차노드로 선정된 다른 노드에 이전할 수 있으므로, 개별

질의에 대한 작업 부하의 평준화를 보다 손쉽게 달성할 수 있다. 그림 9는 단방향 이웃노드 중복저장기법을 이용한 시스템에서 중복저장된 색인어를 동적 부하균등화에 적용하는 예를 들어 보이고 있다. 그림에서 별모양 표시는 해당 노드의 작업부하를 나타낸다. 이 예에서 색인어를 중복저장한 경우 상대적으로 높은 수준의 부하의 평준화가 가능함을 보여주고 있다. 색인어 할당은 그리디하게 순차적으로 부하가 낮은 노드를 선정하는 방법을 사용하였다.

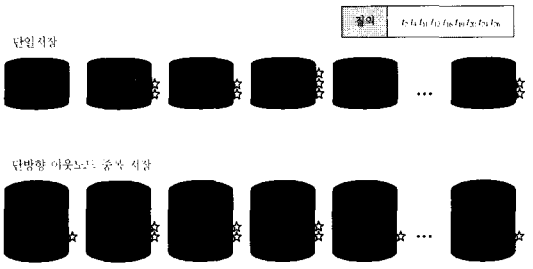

그림 9 단방향 이웃노드 중복저장 시 동적부하균등화의 예

5. 실험 결과

이상에서 설명한 색인어 군집화기법을 활용한 분산 및 중복저장 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 5년간의 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 *tfidf* 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 해 볼 수 있었다. 아래의 50만건 말뭉치를 이용한 실험은 이 5만건 문서를 대상으로 한 실험에서 가장 성능이 좋았던 4가지 파라메터값을 이용하여 실험하였다.

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	5.1	7.9	7.7	7.1	8.4

50만건 문서를 대상으로 성능을 측정한 표 1의 실험결과에서 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 3.2절에서 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 3.1절에서도 지적하였듯이 연결강도 및 연결횟수 임계치의 설정에 따라 성능의 차이가 상당하다는 문제점도 실험적으로 확인되었다.

표 2 병렬정보검색 시스템의 성능 향상정도

2차노드	Random (임의노드 중복저장)			Shift (단방향 이웃노드 중복저장)		
1차노드	Random	Greedy	CWC 50/80	Random	Greedy	CWC 50/80
향상도(%)	13.3	13.0	13.5	11.0	11.8	13.2

표 2는 다양한 색인어 역파일 중복저장 기법을 앞의 실험에서 중복저장하지 않는 무작위 분산저장 방식에 비해 그 성능이 향상된 정도를 표시하고 있다. 1차노드 선정방법으로는 무작위, 그리디 및 분산저장기법실험에서 가장 좋은 성능을 보인 CWC 50/80을 사용하였다. 2차노드 선정 방법으로는 4장에서 설명한 임의노드 중복저장방안과 단방향이웃노드 중복저장방안을 실험하였다. 중복저장을 이용한 기법들은 전반적으로 개별 질의에 대해서도 부하균등화 효과가 높게 발휘되므로 중복저장을 하지 않은 경우에 비해서 상당한 추가의 성능향

상이 이루어 짐을 알 수 있다. 2차노드 선정방법으로 임의노드 중복저장기법을 사용하는 경우가 단방향 이웃노드로 중복저장한 경우보다 전반적으로 더 나은 성능을 보여주고 있는데, 이는 특정 질의에 의해 몇몇 노드에 부하가 집중되는 경우 임의노드 중복저장 기법이 상대적으로 부하균형화를 위해 질의어를 재분배할 수 있는 유연성이 높기 때문이다. 하지만 1차노드 선정방법으로 부하균등화 효과가 뛰어난 CWC 알고리즘을 이용한 경우에는 그 차이가 미미하다.

표 3 노드 결함 시 병렬정보검색 시스템의 성능

방법	노드 하나가 결함인 경우						노드 두개가 결함인 경우		
2차 노드	Random			Shift			Shift		
1차 노드	Random	Greedy	CWC 50/80	Random	Greedy	CWC 50/80	Random	Greedy	CWC 50/80
성능	6.6	6.6	7.3	2.8	1.9	5.9	-15.1	-12.3	-8.5

표 3에는 병렬 시스템의 일부 노드가 운영중지된 상황을 가정하여 검색시스템의 성능을 측정한 결과를 나타내고 있다. 2차노드 선정 방법을 어떠한 것을 쓰더라도 색인어를 중복하여 분산저장하는 방법은 하나의 노드가 결함인 경우에도 기준이 되는 무작위 분산 기법에 비해 더 우수한 성능을 나타내고 있는데, 이는 노드 하나의 결함에 의해서 발생하는 평균 3개의 추가 색인어 처리 부담이 타 노드로 적절히 잘 분배되고 있음을 나타낸다. 단방향 이웃노드 중복저장의 경우에도 문제가 발생한 노드에서 이전된 작업부하는 2차노드가 직접 처리해야 하지만, 자신이 1차노드인 작업 부하는 다시 이웃노드로 전달할 수 있기 때문에 우려할 수준으로 부하의 집중이 발생하지 않았다.

단방향 이웃노드 중복저장기법을 사용한 경우에는 2개 이상의 노드가 운영중지 상황인 경우에도 서비스가 지속될 수 있는데, 본 실험에서는 가장 부하균등화가 어려운 1, 3번 노드가 동시에 사용불가능하게 된 상황을 가정하였다. 특정 노드 주위의 양쪽노드에 동시에 문제가 발생한 경우는, 해당 노드가 왼쪽 노드에서 유입되는 부하와 자신의 부하를 모두 처리해야 하므로 부하균등화가 가장 어려운 상황이라 할 수 있다. 이러한 2개 노드의 동시결함이라는 극한 상황에서도 1차노드 선정방법으로 CWC 알고리즘을 사용하는 경우 본래 성능의 90%이상이라는 지속적인 운영이 가능한 수준으로 서비스가 제공될 수 있음을 확인할 수 있다.

표 2, 표 3의 단방향 이웃노드 중복 방안의 실험 결과를 비교하여 보면, 보다 많은 노드가 정지된 상황일수록 1차노드 선정방식에 따른 그 성능의 차이가 심해지는 경향을 발견할 수 있다. 이는 1차노드들간의 색인어 분배의 불공정성이 2차노드로의 색인어 재분배를 통하여 상쇄될 수 있는 그 유연성이 점차 줄어들기 때문이다. 극단적으로 절반의 노드가 정지되는 상황에서는 표 1의 중복하지 않는 해당 색인어 분산저장방식과 동일한 성능을 나타내게 된다.

6. 결론 및 향후과제

본 논문에서는 PC 클러스터 기반의 병렬 정보검색 시스템의 효율을 향상시키기 위하여 색인어 역파일을 PC 클러스터의 각 노드에 분산 및 중복저장하는 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 시도한 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. 또한 실용적인 PC 클러스터 기반 병렬정보검색시스템을 운영하기 위하여 필수적으로 요구되는 결함포용성과 함께 추가의 동적 부하균형화를 달성할 수 있는 색인어 중복저장기법을 분산저장기법과 연계하여 제안하였다. 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, 지속적인 추가 문서의 유입과 변화하는 질의에 대응하여 이미 생성되어 사용되고 있는 분산 및 중복저장 구조를 저비용으로 변경하는 방안에 대한 연구가 추가적으로 요구된다.

참 고 문 헌

[1] 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

[2] 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

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

[4] 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

[5] Sornil, O. and Fox, E. A., *"Hybrid partitioned inverted indices for large-scale digital libraries"*, Proceedings of The 4th International Conference of Asian Digital Library, Bangalore, India, Dec. 10-12, 2001

[6] 강 유경, 류 광렬, 정상화, 문서 클러스터링에 의한 효율적인 병렬 정보검색 시스템, 정보과학회논문지 : 소프트웨어 및 응용, 제28권 제2호, pp. 157-167, 2001.

[7] 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.

[8] 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

[9] 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.

[10] Wolfson, O., Jajodia, S., and Huang, Y., *"An Adaptive Data Replication Algorithm,"*, ACM Transactions on Database Systems, vol. 22, no. 2, pp. 255-314, 1997

[11] Gray, J., Helland, P., O'Neil, P., and Shasha, D., *"The dangers of replication and a solution."* Proceedings of ACM SIGMOD '96. pp. 173-182, 1996

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

강 재 호
1995년 부산대학교 컴퓨터공학과 학사. 1997년 부산대학교 컴퓨터공학과 석사. 1997년 3월~현재 부산대학교 컴퓨터공학과 박사과정. 1999년 3월~1999년 12월 해동EMC 연구원
2000년 2월~현재 동아대학교 지능형통합항만관리연구센터 연구원. 관심분야는 인공지능, 기계학습, 정보검색, 데이타 마이닝

양 재 완
2000년 부산대학교 전자컴퓨터공학부 컴퓨터공학과 학사. 2002년 부산대학교 컴퓨터공학과 석사. 2002년 2월~현재 온빛시스템 정보기술연구소 연구원. 관심분야는 데이타 마이닝, 기계학습, 정보검색

정 성 원
2000년 부산대학교 전자계산학과 학사. 2000년~현재 부산대학교 전자계산학과 석사과정. 관심 분야는 한국어 정보처리, 정보검색, 데이타 마이닝

류 광 렬
1979년 서울대학교 전자공학과 학사. 1981년 서울대학교 전자공학과 석사. 1983년 3월~1984년 8월 충북대학교 컴퓨터공학과 전임강사. 1992년 University of Michigan 전기 및 컴퓨터공학과박사. 1992년 3월~1993년 2월 Scientific Research Lab., Ford Motor Company 선임연구원. 1993년 3월~현재 부산대학교 정보컴퓨터 공학부 부교수. 관심분야는 기계학습, 데이타 마이닝, 정보검색, 최적화

권 혁 철
1982년 서울대학교 공과대학 전산학 학사. 1984년 서울대학교 공과대학 전산학 석사. 1987년 서울대학교 공과대학 전산학 박사. 1988년~현재 부산대학교 정보컴퓨터 공학부 교수. 1992년~현재 한국정보과학회, 한국어 정보처리 연구회, 프로그래밍 언어 연구회, 운영위원. 1992년~1993년 미국 Stanford 대학 CSLI연구소 연구원. 1992년~1993년 Xerox Palo Alto Research Center 자문. 1997년~1999년 문체부 국어심의회 정보분과 위원. 2000년~현재 부산대학교 정보통신 창업지원 센터장. 2002년~현재 부산대학교 정보컴퓨

터 공학부 학부장. 관심 분야는 한국어 정보처리, 정보검색, 프로그래밍언어, 인공지능

정 상 화

1985년 서울대학교 전기공학과 학사. 1988년 Iowa State University 전기 및 컴퓨터공학과 석사. 1993년 University of Southern California 전기 및 컴퓨터 공학과 박사. 1993년~1994년 University of Central Florida 전기 및 컴퓨터공학과 조교수. 1994년~현재 부산대학교 정보컴퓨터공학부 부교수 및 컴퓨터및 정보통신연구소 연구원. 관심분야는 클러스터시스템, 병렬처리, 정보검색, VOD, Infiniband