

분산 데이터 베이스 설계시의 자료 배정문제에 관한 연구

신 기태 * 박 진우*

A Study on Data Allocation Problems of Distributed Databases

This paper examines the problems of database partitioning and file allocation in a fixed topology distributed computer network.

The design objective is to make files as collections of attributes and to allocate these files to network nodes so that a minimum total transmission cost is achieved subject to storage capacity constraints.

A mathematical model for solving the problem is formulated and, the resulting optimization problem is shown to fall in a class of NP-complete problems. A new heuristic algorithm is developed which uses the idea of allocating attributes according to the transaction requirements at each computer node and then making files using the allocated attributes. Numerical results indicate that the heuristic algorithm yields practicable low cost solutions in comparison with the existing methods which deal with the file allocation problems and database partitioning problems independently.

I. 서 론

분산 데이터베이스(distributed database)란 논리적으로 같은 시스템에 속해있으나 실제적으로는 컴퓨터 네트워크상의 여러 컴퓨터에 흩어져 있는 자료들의 집합을 의미한다 (Ceri, 1985). 분산 데이터베이스 시스템은 자료를 처

리하는 컴퓨터와 컴퓨터들을 연결해주는 통신망, 자료화일 및 검색과 수정을 가능하게 해주는 응용 프로그램들로 구성되어 있다고 간주할 수 있는데 통신망에 연결된 사용자들은 전체 시스템내의 자료를 모두 사용할 수 있다 (Mahmoud, 1976).

분산 데이터베이스 시스템은 다음과 같은 장점을 지니고 있다. 첫째, 중앙 집중식 데이터

* 서울대학교 공과 대학 산업 공학과

베이스 시스템 (centralized database management system)에 비하여 신뢰도가 높다. 둘째, 분산 데이터베이스에서는 자료들이 필요한 곳에 적절히 배치되어 있기 때문에 자료 검색이 빠르고 통신 비용이 줄어든다. 셋째, 중앙 집중식 데이터베이스 시스템에 비하여 용량 증가가 용이하다.

분산 데이터베이스 시스템의 단점으로는 첫째, 전체 자료들의 일관성을 유지하는 방식이 필요하다. 둘째, 고장 발생시 복구가 복잡하다. 셋째, 수정작업이 복잡하다는 점들이 지적된다 (Coffmann, 1981).

이러한 분산 데이터베이스 시스템은 기업체나 조직에서 각 부문별, 지역별로 세분되어 있는 작업들을 통합하여 전체 시스템의 효율을 향상시키려는 노력에 유용한 도구가 되고 있다. 분산 데이터베이스 시스템의 효율에 관한 연구는 다섯 분야로 구별된다 (Benjamin, 1984).

첫째, 컴퓨터 시스템을 지역적으로 분산 설치하는 방법에 관한 연구.

둘째, 분산된 컴퓨터 시스템을 연결하는 통신망을 구성하는 방법에 관한 연구.

셋째, 자료들을 적절히 분할하는 방법 (database partitioning)에 관한 연구.

넷째, 분할된 자료화일을 분산된 컴퓨터 시스템에 할당하는 방법 (file Allocation)에 관한 연구.

다섯째, 사용자의 요구사항 (transaction)을 효율적으로 수행해내기 위한 시스템의 운용방안에 관한 연구.

그런데 이제까지의 관련된 연구 논문들을 살펴보면 위의 다섯가지 문제들이 크게 연관관계를 갖지 못하고 다루어져 왔음을 알 수 있다. 특히 데이터베이스 분할문제의 결과가 화일 배정문제의 입력으로 사용되어야 함에도 불구하고 이 두 문제를 동시에 고려한 것은 거의 전

무한 상태이다. 기존의 방법과 같이 두 문제를 서로 분리하여 접근하는 방법은 시스템의 입장에서 볼 때 부분의 최적해만을 구하는 것이라 할 수 있다. 그러므로 연결관계가 깊은 문제들을 동시에 고려한 연구가 필요하다.

본 연구에서는 데이터베이스 분할문제와 화일 배정문제가 동시에 고려될 경우, 보다 거시적 입장에서의 최적해가 구해질 수 있음에 착안하여 데이터 베이스 분할 문제와 화일 배정 문제를 동시에 고려한 분산 데이터베이스 시스템 설계문제를 수학적으로 모형화하고 이 모형에 대한 효율적인 해법을 제시하고자 한다.

II. 연구 현황

정보 통신망에서 화일 배정에 관한 연구는 Chu (1969)에 의하여 가장 먼저 시도되었다. Chu는 배정할 화일의 갯수가 일정하다고 가정하고 화일의 중복배정을 허용하지 않고 기억용량에 대한 제약식과 수행 작업의 기대 지연 시간을 만족시키면서 총 운용비용 (저장비용+통신비용)을 최소화하는 화일의 배치를 구하였다. Casey (1972)는 동일 화일의 중복 배정을 허용하며 검색과 수정작업을 구분하여 문제를 다루었다. 그런데 Casey의 모형은 NP-complete임이 증명되었다.

위의 두 논문은 단순 화일 배정 (simple file allocation) 문제의 대표적인 예로 분류되는데, 이 문제는 Ramamoorthy (1983)에 의하여 OR분야에서 많은 연구가 이루어진 단일상품 창고 위치선정 (single commodity warehouse location) 문제와 동일함이 증명되어서 이미 개발된 많은 기법들이 응용될 수 있었다.

단순 화일 배정문제를 확장하여 지연시간, 저

장용량, 병렬 처리기능, 가용성, 프로그램 화일과 데이터 화일의 구별등의 가정을 고려한 문제를 일반 화일 배정 (general file allocation) 문제라 부르는데 이와 관련하여 Benjamin (1985)은 Multi Access Broadcast Bus로 연결된 관계 데이터베이스에서 병행 수행 통제(Concurrency Control), 질의어 처리(query processing)를 고려한 화일 배정문제를 다루었다. Morgan (1977)은 프로그램 화일과 데이터 화일을 구별하여 배정하는 방법에 관한 연구를 수행하였다.

Coffman (1981)은 분산 데이터베이스 시스템에서 컴퓨터들의 읽기 처리 능력을 최대화하는 화일의 중복갯수를 결정하는 방법에 대하여 연구하였다.

이외의 연구로는 통신 라인의 용량결정, 컴퓨터 처리능력 선택등에 관한 연구들이 있었다 (Chen, 1980; Irani, 1982; Mahmoud, 1976).

화일 배정문제에서 배정의 대상이 되는 화일을 구성하는 방법에 관한 문제를 데이터베이스 분할(database partitioning) 문제라 한다.

데이터베이스 분할이란 데이터베이스 내의 모든 자료들을 하나의 화일로 보고 각 속성들을 수직분할(vertical partitioning) 혹은 수평분할(horizontal partitioning)하는 것을 말하는데 분할의 결과로 생성된 자료의 집합들이 화일을 구성하게 된다.

이 분야의 연구로는 Hoffer (1976)가 저장비용, 검색비용과 수정비용의 최소화를 목적함수로 하고 각 화일에 할당된 저장용량을 제약조건으로하는 데이터베이스 분할문제를 비선형 0-1 정수계획문제로 모형화하였다. 그 후 Babad (1977)는 속성(attribute)들의 길이가 다른 경우에 대하여 비선형 0-1 정수계획문제로 모형화하였다. Eisner (1976)는 주 기억장치와 보조 기억장치에서 데이터베이스 분할 문제를 다

루었는데 네트워크 이론에서 최소절단 최대유통(minicut-maxflow) 문제와 동일함을 증명하고 이의 해법을 사용하였다. Sacca (1985)는 여러 개의 처리기(processor)가 있는 경우의 데이터베이스 분할 문제를 다루었는데 중복 배정에 관한 고려는 없었다. Navathe (1985)는 데이터베이스의 수직분할에 관한 연구에서 이진 수직분할(hinary vertical partitioning) 방법을 통하여 Clustering기법을 사용한 해법을 제시하였다. Apers (1988)는 데이터베이스를 질의(query)에 따라 분할하여 화일을 구성하고 이들을 통신 비용이 최소화되도록 배정하는 문제를 다루었으나 문제크기가 너무 크기때문에 실제적이지 못한 측면이 있다. Navathe (1989)는 속성의 친밀도 행렬(attribute affinity matrix)을 완전 그래프(complete graph)로 보고 선형 연결 극대 나무(linearly connected spanning tree)를 구성하여 이때 생성되는 사이클을 화일로 구성하는 방법을 연구하였다.

이상의 연구들을 종합해 볼 때 화일의 배정문제를 화일의 구성문제와 별도로 취급한 것들이 대부분이며 화일 구성과 배정을 고려하였다 하더라도 해법상 다를 수 있는 문제의 크기가 극히 제한되어 있는 상태이다.

Ⅲ. 수리적 모형의 개발

1. 가정

1) 각 컴퓨터들은 LAN으로 연결되어 있다.

즉, 컴퓨터의 분산 배치 상태와 이들의 연결 방법은 이미 결정된 것으로 하는데 동일 자료의 통신비용은 발신 및 수신처에 무관하게 일정하다고 가정한다.

2) 발생 가능한 모든 트랜잭션(transaction)은 알려져있다.

즉, 검색과 수정을 요구하는 컴퓨터 위치가 알려져있다고 가정한다.

3) 트랜잭션들의 발생 빈도는 알려져있다.

4) 각 트랜잭션의 1회 수행시 필요한 자료의 양은 알려져있다.

2 기호 설명

I : 속성(Attribute)들의 집합, $|I| = r$.

J : 속성 집합들의 모임(속성들의 가능한 모든 조합), $|J| = m = 2^r - 1$.

K : 컴퓨터 위치들의 집합, $|K| = n$.

A : $a_j = \begin{cases} 1, & \text{속성 } j \text{가 화일 } j \text{에 포함되는 경우.} \\ 0, & \text{그외의 경우.} \end{cases}$

SL_k : 컴퓨터 위치 k의 저장 용량.

$S_{i,k}$: 속성 i를 컴퓨터 위치 k에 저장하는 비용.

L_i : 속성 i의 길이.

$CS_{i,k}$: 속성 i의 단위길이를 컴퓨터 위치 k에 저장하는 비용.

따라서, $S_{i,k} = CS_{i,k} * L_i$

$CR_{i,k}$: 컴퓨터 위치 k에서 화일 j에 포함된 속성 i를 검색하는 비용.

$CU_{i,k}$: 컴퓨터 위치 k에서 화일 j에 포함된 속성 i를 수정하는 비용.

CC : 단위당 통신 비용.

$FR_{i,k}$: 속성 i를 컴퓨터 위치 k에서 검색을 요구하는 빈도.

$FU_{i,k}$: 속성 i를 컴퓨터 위치 k에서 수정을 요구하는 빈도.

$LR_{i,k}$: 속성 i를 컴퓨터 위치 k에서 검색을 요구하는 양.

$LU_{i,k}$: 속성 i를 컴퓨터 위치 k에서 수정을

요구하는 양.

따라서,

$$CR_{i,k} = CC \sum_{i \in j} FR_{i,k} LR_{i,k}$$

$$CU_{i,k} = CC \sum_{i \in j} \sum_{k \neq k1} FU_{i,k1} LU_{i,k1}$$

AC_j : 속성 i를 화일 j에 포함시키는 경우 발생하는 비용.

e : 모든 요소가 1인 벡터.

$$x_j = \begin{cases} 1, & \text{화일을 선택하는 경우, } j=1,2,\dots, m \\ 0, & \text{그 외의 경우.} \end{cases}$$

X = x_j 를 Element로 하는 Column Vector.

$$y_k = \begin{cases} 1, & \text{화일 } j \text{를 컴퓨터 위치 (computer site) } \\ & \text{k에 배정하는 경우.} \\ 0, & \text{그 외의 경우.} \end{cases}$$

3 모형화

$$\text{Min} \sum_{j=1}^m \sum_{i \in j} AC_{i, x_j} + \sum_{j=1}^m \sum_{i=1}^n \left(\sum_{i \in j} CR_{i,k} \right) + \left(\sum_{i \in j} CU_{i,k} \right) + \left(\sum_{i \in j} S_{i,k} \right) y_k$$

st

$$AX \geq e \quad \text{----- (1)}$$

$$\sum_{k=1}^n y_k \leq 1, \quad \forall j=1,2,\dots,m \quad \text{----- (2)}$$

$$\sum_j \left(\sum_{i \in j} L_i \right) y_k \leq SL_k, \quad \forall k=1,2,\dots,n \quad \text{----- (3)}$$

$$x_j, y_k = 0 \text{ or } 1$$

목적함수의 첫번째항은 속성들을 화일로 구성하는 비용이고, 두번째항은 화일을 분산된 컴퓨터들에 배정할때 발생하는 통신 비용과 저장 비용의 합이다.

제약식 (1)은 속성들의 중복 배정을 허용하는 것이다. 제약식 (2)은 구성된 화일이 한 장소의 컴퓨터에 배정되는 것을 보장한다. 제약식 (3)은 각 컴퓨터의 저장 용량제약을 의미한다.

$|k|=1$ 이고 $SL_1 = \infty$ 라 하면 위의 모형은 Set-Covering문제의 형태가 되어서 NP-Complete가 됨을 알 수 있다.

IV. 휴리스틱(heuristic) 해법의 개발

3.절에서 제시한 연구 모형은 조합 최적화문제 (combinatorial optimization problem)로 NP-complete임을 알 수 있기 때문에 현실 문제에 적합한 다항시간 알고리즘 (polynomial time algorithm)을 개발하는것은 어려운 일이다.

그러므로 본 연구에서는 속성들을 분산된 컴퓨터에 중복을 허용하여 배정하고 배정된 속성들을 사용하여 화일을 구성하는 이 단계 접근법을 시도하였다.

1. 입력 자료

Heuristic해법을 사용하기 위한 입력 자료로는 각 컴퓨터의 저장 용량(SLK)과 속성 - 트랜잭션 - 위치 발생 행렬 (attribute-transaction-site incidence matrix)이 있다. 컴퓨터의 저장 용량은 화일의 배정이 저장 용량을 초과하지 못한다는 제약식을 만족시키기 위한것이다.

속성 - 트랜잭션 - 위치 발생 행렬은 Heuristic 알고리즘 개발의 가장 기초가 되는 자료로서 (그림-1)과 같은 형태를 갖고 있는데 각 기호의 정의는 다음과 같다.

- R : 검색 요구.
- U : 수정 요구.

$e_{qik} = 1$, 트랜잭션 q가 컴퓨터 위치 k에서 속성 i를 요구하는 경우.

0, 그외의 경우.

a_{qik} : 트랜잭션 q가 컴퓨터 위치 k에서 속성 i를 요구하는 양.

f_{qik} : 트랜잭션 q가 컴퓨터 위치 k에서 속성 i를 요구하는 빈도.

속성-트랜잭션-위치 발생 행렬을 기초로하여

| Type | Transaction | A ₁ | ... | A _i | ... | A _r | Computer site | Am't of trans. | Frequency |
|------|----------------|------------------|-----|------------------|-----|----------------|----------------|------------------|------------------|
| R | T ₁ | e ₁₁₁ | | | | | S ₁ | a ₁₁₁ | f ₁₁₁ |
| | | | | | | | S ₂ | | |
| | | | | | | | S _n | | |
| | T _q | | | e _{qik} | | | | a _{qik} | f _{qik} |
| U | T _Q | | | | | | S ₁ | | |
| | | | | | | | S ₂ | | |
| | | | | | | | S _n | | |

그림 1. 속성-트랜잭션-위치 발생 행렬

속성들의 배정에 사용될 입력 자료인 속성-위치 사용 행렬 (attribute - site usage matrix)과 배정된 속성들을 화일로 구성하는데 사용되는 입력자료인 속성 친밀도 행렬 (attribute - affinity matrix)을 구성하게된다.

속성-위치 사용 행렬은 속성들이 각 컴퓨터에서 요구되는 검색과 수정의 양을 나타낸다. 그 형태는 (그림-2)와 같고 기호의 정의는 다음과 같다.

$$SR_{ik} = \sum_{q|Type=R} a_{qik} * f_{qik}$$

$$SU_{ik} = \sum_{q|Type=U} a_{qik} * f_{qik}$$

$$Si = \sum_{k=1}^n (SR_{ik} + SU_{ik})$$

$$SU_i = \sum_{k=1}^n SU_{ik}$$

| | A1 | | Ai | | Ar |
|-----|--------------|--|--------------|--|----|
| S1 | SR11 SU11 | | | | |
| ⋮ | | | | | |
| Sk | | | SRki SUki | | |
| ⋮ | | | | | |
| Sn | | | | | |
| Si | | | | | |
| SUi | | | | | |

그림 2. 속성-위치 사용 행렬

속성 친밀도 행렬은 속성 i와 j가 동시에 요구되는 양을 표시하는데 이는 속성 i와 j가 동시에 한 파일에 들어가도 좋은 정도를 나타낸다. 그 형태는 (그림-3)과 같고 기호의 정의는 다음과 같다.

$$b_{ij} = \sum_{p:k^e_{pik}=1 \wedge e_{pik}=1, i \neq j} (a_{pik} * f_{pik} + a_{pjk} * f_{pjk})$$

| | A1 | A2 | | Ai | | Ar |
|----|-----|-----|-------|-----|-------|-----|
| A1 | b11 | b12 | | b1i | | b1r |
| A2 | b21 | b22 | | b2i | | b2r |
| | | | | | | |
| Ai | bi1 | bi2 | | bii | | bir |
| | | | | | | |
| Ar | br1 | br2 | | bri | | brr |

그림 3. 속성 친밀도 행렬

2. 휴리스틱(Heuristic)

해법의 기본 개념

2.1. 초기배정

속성을 가장 많이 요구하는, 즉, 검색과 수정 요구가 가장 많은 컴퓨터 위치에 배정

2.2. 중복배정

속성 i를 초기배정한 곳 이외에 다른 곳에 배정하는 경우로 총 통신비용의 변화를 고려한다. 속성 i를 다른 곳에 배정하여 얻는 비용의 감소는 $CR * SR_{ik}$ (검색 비용)이며, 증가되는 비용은 수정 비용으로써 $CU * (S_{ui} - S_{uk})$ 로 주어진다. 그러므로, 총 통신비용의 변화(TTC)는 $CU * S_{ui} - CU * S_{uk} - CR * SR_{ik}$ 가 된다. 따라서 $TTC < 0$ 이고, 저장 용량이 허용하면 최소의 TTC를 갖는 컴퓨터 위치에 배정한다.

2.3. 배정된 속성들의 분할

이 부분은 데이터 베이스 분할문제중에서 가장 최근에 발표된 Navathe [1989]의 알고리즘을 응용하였는데 이는 데이터 베이스의 수직 분할에 관한 효율적인 해법으로 알려져있다. 그 기본개념은 속성 친밀도 행렬을 완전 그래프로 보고 친밀도를 호의 가중치로 삼아, 이들을 선형 연결 극대 나무로 만들면 이 때 생성되는 사이클들은 의미있는 분할이된다는 것이다.

1), 2) 단계를 통하여 수리적 모형의 제약적인 저장용량제약과 속성들이 한곳이상의 컴퓨터 위치에 배정되어야하는 내용을 충족시키고

3) 단계를 통하여한 컴퓨터위치내에 효율적인

검색을 달성할 수 있도록 화일을 구성하도록 해 준다. 위의 배경과 분할단계에서 통신비용의 감소를 추구하여 수행해나간다.

3. 휴리스틱(Heuristic) 알고리즘

STEP 1 : 속성-트랜잭션-위치 발생 행렬을 입력으로 하여 속성 친밀도 행렬과 속성-위치 사용 행렬을 작성한다.

{ 속성의 배정 }

STEP 2 : 초기배정

각 속성들에 대하여 통신 비용이 가장 큰 위치에 배정, 즉,

$\text{Max}[CR \cdot SR_{ik} + CU \cdot SU_{ik}]$ 인 컴퓨터 위치 k에 속성 i를 배정 ; $\sum_k X_{jk} = 1$.

{ 속성의 중복배정 }

STEP 3 : 각 속성들에 대하여 총 통신 비용의 변화(TTC)를 구한다.

$$TTC = \min_{k|X_{ik}=0} (CU \cdot SU_{ik} - CU \cdot SU_{ik} - CR \cdot SR_{ik})$$

STEP 4 : TTC 0 이고 저장 용량을 넘지 않으면 속성 i를 컴퓨터 위치 k에 배정 ; $X_{jk} = 1$.

STEP 5 : STEP 4의 조건을 만족하는 속성이 없을 때까지 STEP 3과 STEP 4를 반복

{ 위치에 배정된 속성들의 분할 }

STEP 6 : 속성 친밀도 행렬을 완전 그래프로 간주하여 임의의 마디(속성)를 선택.

STEP 7 : 다음 조건을 만족하는 호(edge)를 선택한다.

(1) 기존의 구성된 나무에 선형 연결된 것.

(2) 나무 끝의 선택가능한 호중가

장 큰값을 갖는 것.

모든 Node가 사용될 때까지 계속.

STEP 8 : 선택된 호가 원시 사이클(primitive cycle)을 형성하는 경우.

(1) 만약 사이클 마디가 없다면 사이클의 가능성을 확인하여 친밀도 사이클(affinity cycle)로 지목.

---> Goto STEP 7

(2) 사이클 마디가 존재하면 선택된 호는 삭제. ---> Goto STEP 7

STEP 9 : 사이클을 형성하지않고 후보 분할(candidate partition)이 존재하는 경우

(1) Former Edge가 없으면 확장 가능성을 확인하고 선택된 호를 절단하여 만들어진 사이클을 분리한다.

---> Goto STEP 7

(2) Former Edge가 있으면 Former Edge에 의한 확장 가능성을 확인하고 확장 가능성이 없으면 Former Edge를 절단하고 분할한다. ---> Goto STEP 7

STEP 10 : STEP 6 부터 STEP 9 까지를 모든 컴퓨터 위치에 대하여 수행

4. 알고리즘의 복잡도(complexity)

4.1. 배정 문제의 복잡도는

$O(n \cdot r \cdot a)$ 로 구해진다.

단, 여기서 a는 중복 배정을 수행하는 최대 횟수를 의미한다.

$$a = \frac{\sum SL_k}{\sum L_i}$$

4.2 데이터 베이스 분할문제의 복잡도는

$O(r^2 \cdot n)$ 로 구해진다.

4.3 전체 문제의 복잡도는

$O(n * r^{r+1})$ 로 구해진다.

단, 여기서 n 은 컴퓨터 위치 의 갯수, r 은 속성의 갯수를 의미한다.

5. 예제와 비교

8개의 트랜잭션(검색요구 4개, 수정요구 4개)과 컴퓨터 위치 3곳 그리고 속성이 10개인 경우 입력 자료는 (그림-4)와 같다.

3. 절에서 제안된 방법으로 속성-위치 사용 행렬을 작성하면 (그림-5)와 같다.

3. 절에서 제안된 방법으로 속성들을 컴퓨터에 배정하면 (그림-6)과 같은 결과를 얻게된다.

Heuristic 알고리즘의 비교를 위하여 본 연구에서는 제안된 방법을 IBM - AT 컴퓨터에서 프로그램하였다. 그리고 비교대상이되는 알고리즘은 데이터 베이스의 분할을 통하여 화일을 구성하고 구성된 화일의 배정을 다루는 즉, 두 개의 문제를 별도로 다루는 것으로 선택하였다.

데이터 베이스의 분할법은 본 연구에서와 동일한 것을 사용하였고 화일의 배정 방법은 정수계획법모형에의한 최적해를 구하는 방법을 선택하였다(이하 기존의 분리 배정 방법이라 부른다).

문제 크기에 따른 비교 결과는 (표-1)에 요약되어 있다. 비교 결과를 분석해보면 본 연구의 Heuristic 알고리즘이 배정 결과에 별 차이가 없고 총 통신 비용이 적다는 사실을 확인할 수 있다.

그 이유는 본 연구의 Heuristic 알고리즘이

총 통신 비용을 줄이도록 속성들을 배정한 후 화일을 구성하기 때문에 컴퓨터별 속성의 검색과 수정의 요구량에 대한 고려없이 화일을 구성하고 그 화일들을 배정하는 방법에 비하여 분산 데이터 베이스 시스템내의 총 통신 비용을 줄여주기 때문인것으로 풀이된다.

V. 결론 및 추후 연구 방향

본 연구에서는 각 컴퓨터들이 LAN으로 연결되어 있고 트랜잭션들의 내용과 발생빈도 그리고 1 회 수행시 필요한 자료의 양이 알려져 있는 경우에 대하여 컴퓨터간의 통신 비용을 최소화하는 자료의 배정 문제를 다루었다.

LAN으로 연결되어 있는 시스템의 경우, 통신 네트워크의 용량이 크지 않고 속도가 빠르지 않으므로 가능한 한 통신량을 줄이는 것이 바람직 하다고 여겨진다. 특히 자동화된 생산 시스템의 경우 정확한 제어를 위하여 빠른 반응 시간을 요구하므로 각각의 자료들이 적절히 배치되는 것이 바람직하다.

그러나 앞에서 살펴본 바와 같이 자료의 배정 문제는 NP-Complete 문제이므로 최적해를 구하는 것은 어려운 일이다. 그래서 본 연구에서는 자료의 배정과 화일의 구성 문제를 연결하여 해결하는 Heuristic 알고리즘을 개발하였는데 이 알고리즘의 복잡도는 $O(n * r^{r+1})$ 로 상당히 효율적인 것으로 판단되며 화일 구성과 화일의 배정을 별도로 취급하는 것보다 우수한 결과를 제공함을 확인할 수 있었다.

본 연구에서는 분산 데이터 베이스의 설계에서 자료 배정에 관한 분야로 한정하여 다루었는데 본 연구에서 고려하지 않은 컴퓨터 하드웨어의 구성 방법에 관한 부분, 즉, 컴퓨터의

설치 장소, 통신망의 용량 결정, 통신망 구성 방법등의 고려를 추가하여 연구를 확장하는 것이 요구된다.

| TYPE | TRAN | SITE | ATTR | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | AMT | FREQ |
|-----------|------|------|------|---|---|---|---|---|---|---|---|---|----|-----|------|
| RETRIEVAL | 1 | 1 | | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 28 | 12 |
| | 1 | 2 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1 | 3 | | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 28 | 7 |
| RETRIEVAL | 2 | 1 | | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 26 | 11 |
| | 2 | 2 | | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 26 | 26 |
| | 2 | 3 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| RETRIEVAL | 3 | 1 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 3 | 2 | | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 23 | 12 |
| | 3 | 3 | | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 23 | 21 |
| RETRIEVAL | 4 | 1 | | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 16 | 15 |
| | 4 | 2 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 4 | 3 | | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 16 | 7 |
| UPDATE | 5 | 1 | | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 54 | 8 |
| | 5 | 2 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 5 | 3 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| UPDATE | 6 | 1 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 6 | 2 | | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 25 | 9 |
| | 6 | 3 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| UPDATE | 7 | 1 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 7 | 2 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 13 | 19 |
| | 7 | 3 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| UPDATE | 8 | 1 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 8 | 2 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 8 | 3 | | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 46 | 6 |

그림 4. 속성-트랜잭션-위치 발생행렬

| ATTR | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| SITE1 | 336 | 526 | 286 | 0 | 336 | 0 | 576 | 526 | 286 | 0 |
| | 432 | 432 | 432 | 0 | 432 | 0 | 432 | 432 | 432 | 0 |
| SITE2 | 0 | 676 | 676 | 276 | 0 | 276 | 0 | 676 | 676 | 276 |
| | 225 | 0 | 247 | 0 | 225 | 0 | 0 | 0 | 247 | 0 |
| SITE3 | 196 | 112 | 0 | 693 | 196 | 693 | 308 | 112 | 0 | 693 |
| | 0 | 0 | 276 | 276 | 0 | 276 | 0 | 0 | 276 | 276 |

그림 5. 속성-위치 사용 행렬

| ATTR | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|--------|---|---|---|---|---|---|---|---|---|----|
| SITE 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| SITE 2 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| SITE 3 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

그림 6. 배정 결과

3. 절에서 제안된 방법으로 속성 친밀도 행렬을 구하면 (그림-7)과 같다.

| ATTR | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------|------|------|------|------|------|------|------|------|------|------|
| 1 | 1189 | 432 | 432 | 0 | 1189 | 0 | 964 | 432 | 432 | 0 |
| 2 | 432 | 1746 | 1394 | 0 | 432 | 0 | 784 | 1746 | 1394 | 0 |
| 3 | 432 | 1394 | 1917 | 276 | 432 | 276 | 432 | 1394 | 1917 | 276 |
| 4 | 0 | 0 | 276 | 1245 | 0 | 1245 | 0 | 0 | 276 | 1245 |
| 5 | 1189 | 432 | 432 | 0 | 1189 | 0 | 964 | 432 | 432 | 0 |
| 6 | 0 | 0 | 276 | 1245 | 0 | 1245 | 0 | 0 | 276 | 1245 |
| 7 | 964 | 784 | 432 | 0 | 964 | 0 | 1316 | 784 | 4 | 32 |
| 8 | 432 | 1746 | 1394 | 0 | 432 | 0 | 784 | 1746 | 1394 | 0 |
| 9 | 432 | 1394 | 1917 | 276 | 432 | 276 | 432 | 1394 | 1917 | 276 |
| 10 | 0 | 0 | 276 | 1245 | 0 | 1245 | 0 | 0 | 276 | 1245 |

그림 7. 속성 친밀도 행렬

3. 절에서 제안된 방법으로 화일을 구성하면 (그림8)과 같다.

| | |
|--------|--------------------------------|
| FILE 1 | ATTR 1, ATTR 5, ATTR 7 |
| FILE 2 | ATTR 2, ATTR 3, ATTR 8, ATTR 9 |
| FILE 3 | ATTR 4, ATTR 6, ATTR 10 |

그림 8. 화일 구성 결과

| | | |
|--------|--------|--------------------------------|
| SITE 1 | FILE 1 | ATTR 1, ATTR 5, ATTR 7 |
| | FILE 2 | ATTR 2, ATTR 8 |
| | FILE 3 | |
| SITE 2 | FILE 1 | |
| | FILE 2 | ATTR 2, ATTR 3, ATTR 8, ATTR 9 |
| | FILE 3 | |

그림 9. 최종 결과

[표-1] 알고리즘의 비교 결과

| 문제 | ATTR | TRAN | COM | Heuristic Algorithm | | 기존의 분리배정 방법 | |
|----------------------|----------------------|------|-----|---------------------|-------|--------------------------|-------|
| | | | | 배정 결과 | TCC | 배정 결과 | TCC |
| 1 | 10 | 8 | 3 | { 1, 5, 7 } | 5054 | { 1, 5, 7 } | 5528 |
| | | | | { 2, 8 } | | { 2, 3, 8, 9 } | |
| | | | | { 4, 6, 10 } | | { 4, 6, 10 } | |
| 2 | 15 | 10 | 3 | { 1, 2, 7 } | 8333 | { 1, 2, 7 } | 10084 |
| | | | | { 6, 9, 11 } | | { 5, 8, 10, 12, 13, 15 } | |
| | | | | { 3, 4, 14 } | | { 6, 9, 11 } | |
| | | | | { 2, 7 } | | { 5, 8, 10, 12, 13, 15 } | |
| | | | | { 10, 13, 15 } | | { 6, 9, 11 } | |
| | | | | { 6, 9, 11 } | | { 5, 8, 10, 12, 13, 15 } | |
| | | | | { 5, 8, 12, 15 } | | | |
| 3 | 20 | 10 | 4 | { 1, 3, 6 } | 15777 | { 1, 3, 6 } | 17356 |
| | | | | { 18, 19 } | | { 18, 19 } | |
| | | | | { 7, 10 } | | { 7, 10 } | |
| | | | | { 5, 13, 16 } | | { 5, 13, 16 } | |
| | | | | { 2, 14, 15, 17 } | | { 2, 4, 14, 15, 17 } | |
| | | | | { 1, 3, 6 } | | { 1, 3, 6 } | |
| { 7, 10 } | { 18, 19 } | | | | | | |
| { 18, 19 } | { 7, 10 } | | | | | | |
| { 5, 13, 16 } | { 5, 13, 16 } | | | | | | |
| { 2, 4, 14, 15, 17 } | { 2, 4, 14, 15, 17 } | | | | | | |
| { 8, 9, 11, 12, 20 } | { 8, 9, 11, 12, 20 } | | | | | | |

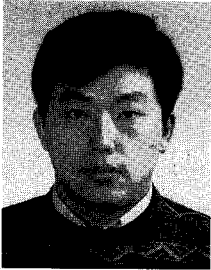
|| 안의 숫자는 속성의 번호를 의미하고, TCC는 총 통신 비용을 뜻한다.

참 고 문 헌

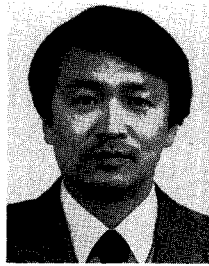
- Apers, P. M. G., "Data allocation in distributed database systems," *ACM Trans. Database Sys.*, 13(3), 1988.
- Babad, J. M., "A record and file partitioning model," *Comm. ACM*, 20(1), 1977.
- Benjamin, W. W., "File placement on distributed computer systems," *IEEE Trans. Comp.*, 1984.
- Benjamin, W. W., "Design of distributed databases on local computer systems with a multiaccess network," *IEEE Trans. Soft. Eng.*, 11(7), 1985.
- Brassard, G. and P. Bratley, *Algorithmics: Theory & Practice*, Prentice hall, 1988.
- Casey, R. G., "Allocation of copies of a file in an information network," *AFIPS Conf. Proc.*, 40, 1972.
- Ceri, S., S. Navathe and G. Wiederhold, "Distribution design of logical database schemas," *IEEE Trans. Soft. Eng.*, 9(4), 1983.
- Ceri, S. and G. Peragatti, *Distributed databases*, McGRAW-HILL, 1985.
- Chen, P. P. and J. Akoka, "Optimal design of distributed information systems," *IEEE Tran. Comp.*, 29(12), 1980.
- Christofides, N., *Combinatorial Optimization*, John Wiley, 1979.
- Chu, W. W., "File allocation in a multiple computer system," *IEEE Trans. Comp.*, 18, 1969.
- Coffman, E. G., E. Gelenbe and B. Plateau, "Optimization of the number of copies in a distributed database," *IEEE Trans. Soft. Eng.*, 7(1), 1981.
- Dowdy, L. W. and D. V. Foster, "Comparative models of the file assignment problem," *ACM. Computing Surveys*, 14(2), 1982.
- Eisner, M. L. and D. G. Severance, "Mathematical technique for efficient record segmentation in large shared databases," *J. ACM*, 23(4), 1976.
- Grapa, E. and G. G. Belford, "Some Theorems to aid in solving the file allocation problem," *Comm. ACM*, 20(11), 1977.
- Hoffer, J. A., "An integer programming

- formulation of computer database design problems," *Inf. Sci.*, 11, 1976.
- Irani, K. B. and G. K. Nicholas, "A methodology for the design of communication networks and the distribution of data in distributed supercomputer systems," *IEEE Tran. Comp.*, 31(5), 1982.
- Karp, R. M., "Reducibility among Combinatorial problems," in *Complexity of computer computations*, Miller, R. E. and J. W. Thatcher (eds), Plenum Press, 1972.
- Mahmoud, S. and J. S. Riordon, "Optimal allocation of resources in distributed information networks," *ACM Tran. Database Sys.*, 1(1), 1976.
- Morgan, H. L. and K. L. Levin, "Optimal program and data locations in computer networks," *Comm. ACM*, 20(5), 1977.
- Navathe, S., S. Ceri, G. Wiederhold and J. Dou, "Vertical partitioning algorithms for database design," *ACM Trans. Database Sys.*, 9(4), 1984.
- Navathe, S. and M. Ra, "Vertical partitioning for database design : A graphical algorithm," *Proc. ACM SIGMOD*, 1989.
- Pirkul, H., "An integer programming model for the allocation of databases in a distributed computer system", *Eur. J. Oper. Res.*, 26, 1986.
- Ramamoorthy, C. V. and B. W. Wah, "the isomorphism of simple file allocation," *IEEE Trans. Comp.*, 32(3), 1983.
- Rothnie, J. B. and N. Goodman, "A survey of research and developement in distributed database management," *Proc. 3rd Conf. VLDB*, 1977.
- Sacca, D. and G. Wiederhold, "Database partitioning in a cluster of processors," *ACM Trans. Database Sys.*, 10(1), 1985.

◆ 저 자 소개 ◆



공동 저자 신기태는 현재 서울대학교 산업공학과 박사과정에 있다. '87년 서울대학교 산업공학과에서 학사과정을 이수하고, '90년 동 대학원에서 석사과정을 마쳤다. 그의 주 관심분야는 CIM, 데이터베이스 등이다.



공동 저자 박진우는 현재 서울대학교 산업공학과 교수로 재직중이다. '74년 서울대학교 산업공학과를 졸업하고, '76년 한국과학기술원에서 석사과정을 마친 후 한국중공업에서 근무한 바 있으며, '85년 캘리포니아 버클리 대학에서 공학 박사학위를 취득했다. 주요 연구분야는 FMS, CIM, 데이터베이스, 시뮬레이션 등이다.