論文 89-26-9-13

분산 컴퓨터 시스템에서 효율적 화일 할당에 관한 연구

(A Study on Efficient File Allocation for Distributed Computer Systems)

洪 進 杓*, 林 濟 鐸**

(Jin Pyo Hong and Chae Tak Lim)

要 約

분산 컴퓨터 시스템에서 화일 할당 문제(file allocation problem)의 해(solution)를 구하는 효율적인 화일할당 알고리듬과 후보 노드(candidate computer site)에 대한 평가치 계산기법을 제안한다.

할당 초기에 선할당(preassignment)을 수행하여 문제의 규모(problem size)를 축소하고 할당이 진행중인 노드의 제어변수에 의해 사용자 상태배열을 변화시키면서 후보 노드의 평가치를 구한다. 후보 노드의 값이 올바르게 평가된 후 선택기준(selection criteria)을 적용하여 가장 적합한 노드를 선택하고 할당상태를 결정한다.

제안한 알고리듬은 발견적 방법으로 다항식 시간(polynomial time) 알고리듬이며 여러 예제에 실현한 결과 최적해를 구하는데 기존 방법보다 우수함을 나타냈다.

Abstract

An efficient file allocation algorithm and a new method which calculate appraisal value of candidate computer site for distributed computer systems are proposed.

The file allocation problem size is reduced by using the preassignment condition. The appraisal value of candidate node is calculated as the user state array and node state array are varied according to control variables. As the selection criteria is applied to the candidates, the reasonable node is selected and assign state is determined.

The proposed algorithm is heuristic polynomial time algorithm. By performing algorithm for sample problems. It is shown that the proposed algorithm is superior to conventional method in terms of deviation from optimal solution.

*正會員, 金星電線研究所
(R & D Center GoldStar Cable Co., Ltd.)
**正會員, 漢陽大學校 電子工學科
(Dept. of Elec. Eng., Hanyang Univ.)
接受日字 : 1989年 6月 26日

Ⅰ. 서 론

분산컴퓨터 시스템에서 자원공유(resource sharing)를 위하여 시스템 설계시 효과적인 화일 액세스(access)가 가능하도록 프로그램과 데이타 화일을 각 노드(computer site)에 할당 한다.

FAP(file allocation problem)의 해를 구하는 방법으로 동적계획법(dynamic programming), 정수계획법(integer programming)과 발견적방법(heuristic method)등 많은 기법이 제안되었으며 화일할당의 목적함수(objective function)는 최소비용을 고려한 할당, 성능최적화를 고려한 할당으로 구분하여 표현된다.

저장(storage)비용, 질의(query)비용, 갱신(update)비용을 포함한 최소 동작비용(operating cost)을 구하는 모델[2]이 제안되었고 성능최적화를 목적으로 하는 경우는 각 노드의 작업 부하(workload)에 따라 대기행렬모델(queueing model)을 사용하여 화일할당을 행하며, 프로그램과 데이터를 분리한 할당[7] 통신회선의 용량(channel capacity)과 화일할당을 함께 고려한 모델[4] 질의-갱신 비의 변화에 따른 화일사본(copy)수의 결정[3] 등 여러종류의 할당기법이 제시되었다.

컴퓨터 시스템에서 단수(single)화일과 복수(multiple)화일인 경우와 신뢰도(reliability), 반응시간(response time)을 고려하여 제약조건(constraints)을 포함시킨 할당기법도 연구되었다.

본 논문에서는 단수화일에의 할당에 있어 발견적(heuristic)방법으로 $0(n^4)$의 복잡도(complexity)를 갖는 효율적 화일할당 알고리듬을 제안하고 할당상태의 표시치(representative value)를 나타내는데 있어 할당순서에 따른 제어변수를 결정하고 사용자와 노드상태 배열을 변화시켜 평가치를 구하는 기법을 제안하였다.

할당 상태의 비용을 고려하여 선할당조건을 실행하고 문제규모를 할당 미결정노드의 2배로 줄인다. 제안한 기법으로 평가한 노드에 대하여 2가지의 후보 선택기준을 적용하므로써 최적해에 근접한 노드를 선택한다.

제안한 평가치 계산기법 및 할당알고리듬을 C언어 프로그램으로 VAX11/750에서 여러 예제에 대하여 실현시킨 결과 최적해를 구하는데 있어 기존방법 보다 우수함을 입증하였다.

Ⅱ장에서는 FAP의 일반적 모델을 Ⅲ장에서는 문제규모(problem size)를 줄이기 위한 선할당(preassignment)조건을 제시하였다. Ⅳ장에서는 평가치 계산기법과 화일할당 알고리듬을, Ⅴ장에서는 예제및 성능 평가를 나타냈다.

Ⅱ. FAP의 일반적인 모델

일반적으로 화일할당의 비용 최소화 문제의 최적해를 정수선형 계획법과 열거법(enumerative method)으로 구하는데 목적함수의 최적해는 노드수가 증가함에 따라 계산시간이 지수함수적으로 증가하므로 (2^n, n : 컴퓨터 시스템의 노드 수) 비록 최적해를 구할 수 있지만 시스템 규모가 큰 경우에는 매우 많은 비용이 소요된다.

발견적 방법에 의한 해는 모든 문제에 대하여 전역적 최적해(global solution)를 구하는 것은 불가능 하지만 국부적(local)최적해, 즉 합리적인 해를 규모에 따른 다항시간 복잡도(polynomial time complexity)와 적은 비용으로 구할 수 있어 응용분야에 따라서 경제적인 방법이 된다.

복수 화일 할당 문제는 각각 단수 화일 문제로 분해하여[7] 나타낼 수 있으므로 SFAP(single file allocation problem)의 전체 비용에 대한 표현은 다음과 같다.

식(1)에서 목적함수 Z를 최소화하는 변수 X_{jk}와 Y_k를 구한다.

$$Z=\sum_{j=1}^{n}\sum_{k=1}^{n}\lambda_j d_{jk} X_{jk}+\sum_{k=1}^{n} F_k Y_k \qquad (1)$$

$$\text{단, } \sum_{k\in N_j} X_{jk}=1 \ (j=1, 2\cdots\cdots n) \qquad (2)$$

$$Y_k=0,\ 1 \qquad (k=1, 2\cdots n) \qquad (3)$$

$$F_k=\sigma_k+\sum_{k=1}^{n}\lambda_{jk} d_{jk} \qquad (4)$$

$$0\leqq\sum_{j\in P_k} X_{jk}\leqq n_k Y_k \quad (k=1, 2\cdots n) \qquad (5)$$

n = 시스템의 노드 수
λ_j = 노드 j에서의 질의 트래픽량(volume)
λ_j = 노드 j에서의 갱신 트래픽량.
d_{jk} = 노드 j, k사이의 질의 통신 단위비용
d_{jk} = 노드 j, k사이의 갱신 통신 단위비용
σ_k = 노드 k의 화일 저장비용
X_{jk} = λ_j의 노드 k를 액세스하는 일부분(fraction)
N_j = 사용자 j가 액세스 가능한 노드의 색인(index) 집합
P_k = 노드 k를 액세스 가능한 사용자의 색인집합
n_k = P_k의 원소 수

Ⅲ. SFAP의 간소화(simplification)

SFAP에서 K1, K0는 화일의 사본을 가지고 있는 노드, 사본을 가지고 있지 않은 노드, K2는 할당 혹은 비할당이 결정되지 않은 노드의 집합이라고 할때,

K0 = {i : Yi = 0} ; 비할당 노드

K1={i : Yi=1} : 할당 노드
K2={i : Yi=U} ; 할당 미결정 노드
간소화 기법은 선택기준(selection criteria) 에 의해 평가치를 비교 조사하는 과정에서 후보 할당상태의 수를 줄이고 반복회수를 줄일 수 있다.

1. 할당조건

화일할당을 위한 최소경계(bound) Δi를 결정하고 만일 최소경계가 양(positive)이면 해당 노드를 할당으로 결정한다. 수학적으로 표현하면

i∈K2, j∈Pi에 대하여

$$\Delta i=\sum_j \lambda_j \min_{k\in K1\cup K2} (d_{jk}-d_{ji})_+ - F_i \qquad (6)$$

Δi>0 이면 $Y_i=1$.

사용자가 비할당을 제외한 노드에 대하여 최소 비용 절감(minimum cost saving)의 합의 노드 i의 고정비용(저장비용과 갱신비용의 합)을 초과할때 화일사본을 노드 i에 할당한다.

2. 비할당 조건(Ⅰ)

화일 할당을 위한 최대 경계(bound) Ωi를 결정하고 만일 최대 경계가 음(negative)이면 해당노드를 비할당으로 결정한다. 수학적으로 표현하면,

i∈K2, j∈Pi에 대하여

$$\Omega i=\sum_j \lambda j \min_{k\in K1} (d_{jk}-d_{ji})_+ - F_i \qquad (7)$$

Ωi<0 이면 $Y_i=0$

모든 사용자가 할당노드에 대하여 최소비용 절감의 합이 노드 i의 고정비용을 초과하지 않을 때 화일사본을 노드 i에 비할당한다.

3. ni의 감소기법 (Ⅰ)

i∈K2, j∈Pi에 대하여

$\alpha i=\min_{K\in K1}(d_{jk}-d_{ji})$

$\alpha_i=0$ 이면 Pi=Pi−{j}, ni=ni−1

사용자 j가 기존할당 노드 k에 엑세스하는 것이 노드 i에 대한 액세스보다 경제적이므로 집합 Pi에서 j를 삭제하고 ni를 1감소시킨다.

4. ni의 감소기법(Ⅱ)

i∈K2, j∈K0∪K2, k∈K1∪K2에 대하여

$$\lambda_j d_{jk}+f_k/n_k=\min_{\ell\in K1\cup K2}(\lambda_j d_{j\ell}+f_\ell/n_\ell) \qquad (8)$$

식(8)을 만족하면 $X_{jk}=1$이고 아닌 경우 $X_{jk}=0$

단, $f_k=\begin{cases}F_k & k\in K2\\ 0 & k\in K1\end{cases}$

$X_{jk}=1$인 경우,

k∈K2이면 Pi=Pi−{j}, ni=ni−1

사용자 j가 노드 k의 화일을 액세스 할때 k가 K2의 원소가 아니면 집합 Pi에서 j를 삭제하고 ni를 1감소시킨다.

5. 비할당 조건(Ⅱ)

할당의 초기화 과정에서 즉 모든 노드가 할당 미결정일때 노드 i, k에 대하여 통신비용의 차의 합이 저장비용의 차보다 작을 때 노드 i에는 화일 사본을 비할당한다.

$$Fi-Fk>\sum_j \lambda_j (d_{jk}-d_{ji})_+ \text{ 이면 } Y_i=0$$

위 조건은 집합 K0, K1, K2와 독립적이므로 항상 흐름의 초기에만 사용된다.

Ⅵ. 할당 알고리듬 및 평가치계산

1. 할당 알고리듬

집합 K0, K1, K2, K3=K1∪K2, K4=K0∪K2, Pi는 알고리듬의 프로그램 수행 과정에서 배열 구조(array structure)로 표현하는 것이 편리하다. 할당 상태[0∪∪1101]에 대하여 노드 상태, 사용자 상태 배열을 그림 1에 나타냈다. 알고리듬은 제어변수에 대하여 n^2, 사용자의 집합에 대하여 n, 각 노드의 할당상태에 대한 오퍼레이션으로 $0(n^4)$의 복잡도를 갖는다.

```
K0[  ]=[1000010] /*할당상태가 0인 비트를 1로세트*/
K1[  ]=[0001101] /*할당상태가 1인 비트를 1로세트*/
K2[  ]=[0110000] /*할당상태가 ∪인 비트를 1로세트*/
K3[  ]=[0111101] /*K1[ ]와 K2[ ]의 비트 2진 연산*/
K4[  ]=[1110010] /*K0[ ]와 K2[ ]의 비트 2진연산*/
```

(a) 노드 상태

```
P[k][j] = | 1 0 0 1 1 1 1 1 |   /*노드1을 액세스 할수
          |         .       |     있는 사용자 상태를
          |         .       |     나타내며1, 4, 5, 6, 7, 8
          | 0 1 1 1 1 0 0 0 |     사용자가 가능*/
          | . . . . . . . 1 |
```

(b) 사용자 상태

그림 1. 노드상태와 사용자상태 배열
Fig. 1. Node state and user state array.

할당 알고리듬의 순서를 단계별로 나타내면 다음과 같다.

〔단계 1〕 모든 노드의 상태를 할당 미결정상태로 초기화 한다.

K 0 =[0, 0, ····0]

K 1 =[0, 0, ····0]

K 2 =[1, 1, ····1]

〔단계 2〕 비할당조건 (Ⅱ)의 수행

조건을 만족하는 노드 i가 존재하면 Yi=0, i=3

〔단계 2〕 할당조건의 수행

조건을 만족하는 노드 j가 존재하면 Yj=1, j=n−1

K 1 =[0, 0, 0, ····1, 0] j번째 비트를 1로 세트

K 2 =[1, 1, 0, ····0, 1] j번째 비트를 0로 세트

〔단계 4〕 비할당 조건 (Ⅰ)을 수행하고 조건을 만족하는 노드 k가 존재하면

Yk=0, k=2

K0=[1, 0, 1, ····0, 0]

K2=[0, 1, 0····0, 1]

〔단계 5〕 할당 미결정노드 U에 대하여 U−cut=0로 세트하고 U비트에 대하여 순차적으로 0,1로 세트한다.

〔단계 6〕 할당 상태에 대한 평가치를 계산하고 U−cnt를 1증가시킨다.

〔단계 7〕 할당 미결정노드의 순차적 세트에 대한 평가치 완료검색

U−cnt<2x(U의 갯수)이면 단계 5로

〔단계 8〕 선택 기준에 의해 노드 j를 선택하고 j의 할당상태를 결정한다.

〔단계 9〕 K2[ ]의 모든 비트=0이면 최종평가치와 할당상태를 결정하고 아니면 단계 3으로

위의 단계에서 K[ ]=0 이면 항상 단계 9로 종료한다.

2. 평가치 계산기법

후보노드 할당 상태에 대하여

〔단계 1〕 노드상태 배열을 구성한다.

K0[ ], K1[ ], K2[ ]

〔단계 2〕 ni의 감소기법 (Ⅰ)을 수행하고 조건을 만족하면 사용자 상태 배열에서 j를 제거한다.

P[i] [j]=0 ni=ni− 1

〔단계 3〕 ni의 감소기법 (Ⅱ)를 수행하고 제어변수가 1인 경우 노드 k가 K2[ ]의 원소가 아니면 사용자상태 배열에서 j를 제거한다.

〔단계 4〕 j∈K4에 대하여

평가치 $1=\sum_{j}(\lambda[j]\ d[j]\ [k]\ X[j]\ [k]+X\ X[j]\ [j]\ F[j]/nj)$

j∈K1에 대하여

평가치 $2=\sum_{j}F[j]$

평가치=평가치 1+평가치 2

3. 선택기준

평가치 최소선택(MINAPP) ;

할당 알고리듬의 〔단계 9〕에서 2×[U−cnt]의 후보중 최소평가치를 갖는 할당상태를 선택한다.

평가치 최대차 최소선택(MAXDAPP) ;

할당 알고리듬의 〔단계 7〕에서

차(diff)=| 할당상태[U−cnt]의 평가치−할당상태[U−cnt+1]의 평가치 | 〔단계 9〕에서 평가치의 차가 가장 큰 후보의 짝(pair)중 최소평가치를 갖는 할당상태를 선택한다.

V. 결과 및 성능평가

CASEY의 5노드 예제에 대하여 제안한 평가치 계산기법과 할당알고리듬을 이용하여 할당한 예를 나타낸다.

$$d_{jk}=d'_{jk}=\begin{bmatrix} 0, & 6, & 12, & 9, & 6 \\ 6, & 0, & 6, & 12, & 9 \\ 12, & 6, & 0, & 6, & 12 \\ 9, & 12, & 6, & 0, & 6 \\ 6, & 9, & 12, & 6, & 0 \end{bmatrix}$$

$\lambda_j=[24,\ 24,\ 24,\ 24,\ 24,\ 24]$

$\lambda_j=[2,\ 3,\ 4,\ 6,\ 8]$

$\sigma_k=[0,\ 0,\ 0,\ 0,\ 0]$

그림 2. 예제의 입력파라미터

Fig. 2. Input parameter of example.

열거법에 의해 2^n-1개의 할당상태를 전부 계산하여 최적해는 노드 1, 4, 5에 화일사본을 할당하고 총 비용은 705이다.

그림 3 에서 MAXDAPP기법에 의해서 최적해를 나타내고 MINAPP는 국부적 최적해를 나타낸다.

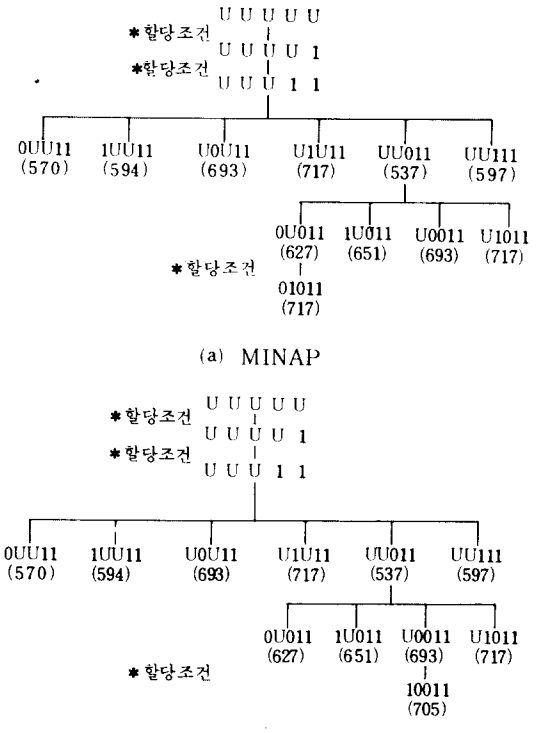

그림 3. 5노드 예제에 대한 평가
Fig. 3. Evaluation of 5node example.

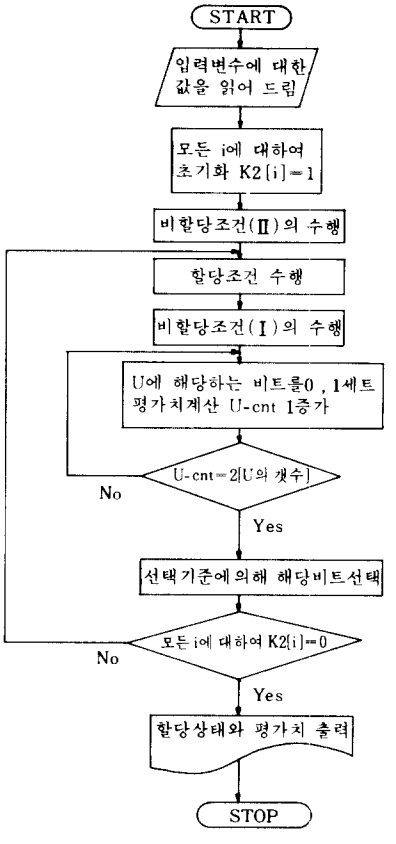

그림 4. 할당알고리듬의 흐름도
Fig. 4. Flowchart of allocation algorithm.

제안한 알고리듬은 발견적 방법으로 다항시간 알고리듬이며 $O(n^4)$의 복잡도를 갖는다. 문헌의 예제에 대하여 실현한 결과를 표 1에 나타냈다. 예제 1은 앞에서 보인 5노드 문제이고 예제 2-6은 19노드의 ARPANET[3]에 대한 예로 질의 트래픽량에 대한 갱신 트래픽량을 0.1, 0.2, 0.3, 0.4, 1.0으로 변화시키면서 할당을 실행한다. 예제 7-8은 Mabmoud[4]의 9노드 예제이다.

최적해를 구하는데 있어 MAXDAPP가 가장 좋은 결과를 보였고 8개의 예제중 6개의 최적해 즉,75%의 최적해를 구할 수 있었다. 문제 규모가 클수록 제안 알고리듬이 유리하며 노드수가 많은 경우 MIAPP와 MAXDAPP가 동일한 결과를 얻었다. 예제 7, 8의 최적해는 국부적 최적해의 가정이며 MAXDLB가 가장 편차가 크게 나타났다. 제안한 방법이 기존 방법보다 최적해에 대한 편차면에서 우수하게 나타났다.

표 1. 할당 알고리듬의 최적해에 대한 퍼센트 편차
Table 1. Percent deviation of allocation algorithm from optimal solution.

예제	최적해	본 논 문 제 안 방 법				기 존 방 법			
		MINAPP		MAXDAPP		MINLB		MAXDLB	
		비 용	편 차	비 용	편 차	비 용	편 차	비 용	편 차
1	705	717	1.7	705	0	717	1.7	717	1.7
2	117896	117896	0	117896	0	117896	0	117896	0
3	188738	189929	0.63	189929	0.63	189929	0.63	189929	0.63
4	243480	243480	0	243480	0	243480	0	272208	11.8
5	291790	297032	1.8	297032	1.8	306404	5.0	297032	1.8
6	437720	437720	0	437720	0	465860	6.4	437720	0
7	27698	27698	0	27698	0	27698	0	29788	7.5
8	203585	203585	0	203585	0	203585	0	206066	1.2
평 균			0.51		0.30		1.71		3.0
표준 편차			0.62		0.54		2.06		3.70

Ⅵ. 결　　론

분산 컴퓨터 시스템의 화일 할당 문제에 관해 발

견적 탐색에 근거를 두어 효과적인 화일 할당 알고리듬을 제시하였고 화일 할당 상태의 평가치 구하는 방법을 제안하였다.

컴퓨터 시스템에서 노드가 많은 경우 열거법에 의한 최적해를 구하는데 실행 시간이 지수 함수적으로 증가하여 실제적 응용이 불가능하므로 빠른 시간에 근사적 최적해를 구하는 방법을 이용한다.

먼저 문제의 규모를 축소하기 위하여 주어진 할당 상태의 비용을 고려하여 선할당을 수행하고 선할당 조건을 만족하는 노드는 후보 노드에서 제외되므로써 가능한 해의 수가 대폭 줄어듬을 알 수 있다.

제안한 할당 알고리듬은 할당 과정에서 제어 변수에 동적(dynamic)으로 대응하여 질의, 갱신, 저장비용으로 후보 노드의 정확한 평가치를 얻었으며 75%의 최적해를 구하였다. 분산 처리 시스템의 일반적인 모델로 많은 응용분야에 적용이 용이하며 선할당을 조절하므로써 전체적인 성능을 높일 수 있고 노드수가 많을수록 경제적이다.

단수 화일 할당 문제의 여러 예제에 대하여 제안한 알고리듬과 평가치 계산기법을 C언어로 프로그램하여 VAX 11/750에서 실현, 최적해에 근사한 결과를 얻으므로써 알고리듬의 우수성을 입증하였다.

앞으로 일반적인 복수화일 할당 문제와 여러가지 제약조건이 부가된 문제에 대하여 적용 가능한 할당 알고리듬의 개발이 기대된다.

參 考 文 獻

[1] A.A. Keuhn and M.J. Hamburger, "A heuristic program for locating warehouses," *Manage. Sci.*, vol. 9, pp. 643-666, July 1963.

[2] W.W. Chu, "Multiple file allocation in a multiple computer system," *IEEE Trans. Comput.*, vol. C-18, pp. 885-889, Oct. 1969.

[3] R.G. Casey, "Allocation of copies of a file in an information network," *AFIPS, SJCC*, pp. 617-625, 1972.

[4] S. Mahmoud and J.S. Riordon, "Optimal allocation of resources in distributed information networks," *ACM Trans. Data Base Syst.*, vol. 1, pp. 66-78, Mar. 1976.

[5] U. Akinc and B. Khumawala, "An efficient branch and bound algorithm for the capacitated warehouse location problem," *Manage. Sci.*, vol. 23, pp. 585-594, Feb. 1977.

[6] E. Grapa and G.G. Belford, "Some theorems to aid in solving the file allocation problem," *Commun. Ass. Comput. Mach.*, vol. 20, pp. 878-882, Nov. 1977.

[7] H.L. Morgan and K.D. Levin, "Optimal program and data locations in computer networks," *Commun. Ass. Comput. Mach.*, vol. 20, pp. 315-322, May, 1977.

[8] C.V. Ramamoorthy and B.W. Wah, "File placements of relations in a distributed relational data base," in Proc. 1st Int. Conf. Distrib. *Comput. Syst., Huntsvill, AL*, Oct. 1979.

[9] P.P.S. Chen and J. Akoka, "Optimal design of distributed information system," *IEEE Trans. Comput.*, vol. C-29, pp. 1068-1080, Dec. 1980.

[10] K.B. Irani and N.G. Khabbaz, "A methodology for the design of communication networks and distribution of data in distributed supercomputer systems," *IEEE Trans. Comput.*, vol. C-31, pp. 419-434, May 1982.

[11] C.V. Ramamoorthy and B.W. Wah, "The isomarphism of simple file allocation," *IEEE Trans. Compt.*, vol. C-32, no. 3, pp. 221-231, Mar. 1983.

[12] B. Gavish and H. Pirkul, "Computer and Database location in distributed computer systems," *IEEE Trans. Compt.*, vol C-35, no. 7, pp. 583-590 July 1986.

著 者 紹 介

洪 進 杓(正會員) 1958年 3月 20日生. 1980年 2月 한양대학교 전자공학과 졸업. 1982年 2月 한양대학교 대학원 전자공학과 졸업. 1982年 3月～현재 한양대학교 대학원 전자공학과 박사과정 재학중. 1983年 9月～현재 금성전선 연구소 재직중. 주관심분야는 컴퓨터 아키텍처, 컴퓨터 네트워크 시스템등임.

林 濟 鐸 (正會員) 第25卷 第8號 參照. 현재 한양대학교 전자공학과 교수, 대한전자공학회 부회장.