論文 90-27-2-15

분산 컴퓨터 시스템에 있어서의 화일 할당 알고리듬에 관한 연구

(A Study on File Allocation Algorithm in Distributed Computer Systems)

洪 進 杓*, 林 濟 鐸**

(Jin Pyo Hong and Chae Tak Lim)

要 約

분산 컴퓨터 시스템에서 정보 화일을 효율적으로 관리, 운용하기 위한 화일 할당 문제의 최적해를 구하는 화일 할당 알고리듬을 제안한다.

컴퓨터 노드(site)가 많은 경우 화일 할당에 많은 시간이 소요되므로 선 할당(preassignment)을 수행하여 문제의 규모를 축소하고 계산시간을 개선한다. 할당이 진행중인 상태의 표시치(representative value)를 정확하게 나타내기 위해 기대값을 이용한 평가치 계산기법을 제안하고 빠른 시간에 최적 할당이 이루어지도록 후보 노드(candidate node)에 대한 선택기준(selection criteria)을 설정한다. 선택기준에 의해 각 노드의 화일 할당 여부를 결정하고 최종 평가치가 할당 상태의 전체 비용을 나타낸다.

제안한 알고리듬은 다항식 시간(polynomial time)알고리듬이며 여러 예제에 실현한 결과 할당 소요시간이 개선되고 기존의 할당 알고리듬과 비교하여 최적해를 구하는데 있어 우수함을 나타냈다.

Abstract

An optimal file allocation algorithm which seeks optimal solution of file allocation problem for efficient management and operation of information files in distributed computer systems is proposed.

Since file allocation time in practical applications that have many computer sites is too long, the problem size has to be reduced and computation time is improved by using preassignment conditio. A new method which calculate appriasal value for accurrate representation of assigned state is proposed and the selection criteria to candidate nodes for rapid determination of allocation are given. By using selection criteria, file allocation is determined and final appraisal value represent total cost of assigned state.

*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 9月 12日

I. 서 론

정보 프로세서(information processor)를 네트워크 형태로 상호 연결하는 분산 컴퓨터 시스템에서 각노드(computer site)에 저장되어 있는 정보 화일을 효율적으로 운용하기 위해서는 화일의 최적 할당이 매

우 중요한 문제가 된다.

FAP(file allocation problem)의 해를 구하는 기법에 있어 최적의 기준에 따라 접근방법이 달라지는데 최소 비용을 목적으로 할때 비용이란 화일의 저장, 질의, 갱신과 통신비용의 구성으로 목적함수(objective function)를 표현한다. 일반적으로 최적화 문제는 비선형 정수 계획법(integer programming)의 문제이며 목적함수가 포괄적으로 구성되므로 구해진 해는 실제 응용이 가능하다. 발견적 탐색(heuristic search)에 의한 해는 최적해와의 차를 평가하기 힘들지만 경제적인 측면에 비중을 두고 시스템의 성능 평가와 컴퓨터 용량의 한계치를 계약조건(constraint)으로 포함한다.

최적 화일 할당의 또 하나의 평가는 성능이며 최소 반응시간(responsetime)과 최대 처리량(throughput)을 목적으로 하는 경우이다. 성능에 근거한 FAP의 기법은 다양한 화일 할당의 비용을 무시하거나 상수로 가정하고 총 비용의 한계치를 제약 조건으로 포함한다. 실제 응용에서 FAP의 해는 비용 뿐만아니라 성능면도 시스템 설계시와 운용에 있어 고려되어야 한다.

Chu[2]는 각 화일의 사본 수가 결정된 경우 완전연결(fully connected)네트워크에서 화일의 최적 할당을 연구하였고 Whitney[13]는 선형그래프의 노드에 화일 사본을 할당할때 그래프의 가장자리(edge)를 따라서 발생하는 트래픽량을 최소로 하는 할당기법을 제안하였으며 Casey[3]는 갱신, 질의 트래픽 트랜잭션을 구분하는 할당 모델을 제안하였으며 흐름행렬(flow matrix)이 주어졌을때 통신 링크에 적절한 용량을 할당하는 문제는 Mahmoud[4]에 의해서 제안되었다.

본 논문에서는 발견적 방법으로 $0(n^4)$의 복잡도(complexity)를 갖는 최적 화일 할당 알고리듬을 제안하고 할당 상태의 표시치를 나타냄에 있어 할당미결정 노드에 대한 기대값을 사용하여 평가치를 구하는 기법을 제안하였으며 C언어 프로그램으로 VAX 11/750에서 FAP의 여러 예제를 실현, 할당 소요시간을 개선하였으며 최적 해를 구하는데 있어 기존방법보다 우수함을 나타냈다. Ⅱ장에서는 수학적 모델을, Ⅲ장에서는 선 할당 조건을 나타냈으며 Ⅳ장에서는 화일 할당 알고리듬을 제안하였고 Ⅴ장에서는 성능을 평가하였다.

Ⅱ. 수학적 모델

분산 정보 처리 시스템의 화일 할당에 대한 수학적 모델은 비용을 목적함수로 하는 경우 성능을 제약조건으로 성능을 목적함수로 하는 경우 비용을 제약조건으로 구성한다.

일반적인 시스템의 설계 운용시 컴퓨터 용량, 위치 통신회선, 프로그램, 데이타 화일의 할당에서 결정변수는

$Y_k^m = 1$ 용량 a_m 컴퓨터의 노드 k에 할당

$X_k^d = 1$ 화일 d의 사본이 노드k에 할당

$L_{ij}^c = 1$ 용량 Qc 통신 회선이 노드 i, j에 할당

$Z_j = 1$ 프로그래 p의 사본이 노드j에 저장

$X_{ijp}^d = 1$ 노드 i에서 화일d의 트랜잭션이 노드 j에 있는 프로그램p를 경유

$X_{jkpd} = 1$ 노드 j에 있는 프로그램p에 의해서 처리되는 트랜잭션이 노드k에 있는 화일 d를 액세스

2진 제약조건으로 인해 위의 결정 변수는 조건을 만족하지 않는 경우 0이다. 최소 비용을 목적으로할 때 목적함수는 전체 비용을 나타내며 결정변수의 값이 할당의 결과를 나타낸다. 컴퓨터 용량은 처리량(단위 시간당 처리할 수 있는 트랜잭션의 수)으로 표현한다.

$$\text{비용 } 1 = \sum_{m,k} Cm\ Y_k^m \quad k \in k, m \in M$$

Cm : 컴퓨터 m의 비용　　K : 노드의 집합

M : 컴퓨터의 집합

화일의 소프트웨어 비용은 화일 관리 시스템의 임대비용과 화일 구조및 데이타 항목의 설정 비용으로 표현한다.

$$\text{비용 } 2 = \sum_{d,k} D_d\ X_k^d \quad d \in N$$

D_d : 화일d의 설치 비용

N : 화일의 집합

노드를 연결하는 통신 회선은 서로 다른 용량(대역폭)을 가지며 설치 비용은 노드 간 거리와 회선용량의 함수로 표현된다.

$$\text{비용 } 3 = \sum_{i,j,c} [B_{1c} + (Bc\ D_{ij})] L_{ij}^c \quad i \in K, c \in C$$

B_{1c} : 통신회선 c의 고정 비용

Bc : 통신회선 설치비용

D_{ij} : 노드i, j의 거리 행렬

화일의 저장 비용과 프로그램의 저장 비용을 구분하여 나타낸다.

$$\text{비용 } 4 = \sum_{d,k} C_{kd}\ X_k^d$$

비용 5 $=\sum_{j,p} S_{jp} Z_j^p \quad p \in p$

C_{kd} : 노드k에 화일d의 저장비용
S_{jp} : 노드j에 프로그램p의 저장 비용
p : 프로그램의 집합

사용자에서 프로그램에 대한 질의 통신 비용

비용 6 $=\sum_{i,j,p,d} Q_{ip}^d Q_{ij} X_{ijp}^d (1+\gamma'_{ij})$

Q_{ip}^d : 노드i에서 p를 경유d에 대한 질의 트래픽 량
Q_{ij} : 노드i에서 j에 대한 질의 단위 비용
: 노드i에서 노드j에 있는 프로그램에 질의 요구에 대한 응답 추정비

사용자에서 프로그램에 대한 갱신 통신 비용

비용 7 $=\sum_{i,j,p,d} U_{ip}^d U_{ij} X_{ijp}^d$

U_{ip}^d : 노드i에서 p를 경유 d에 대한 갱신 트래픽 량
U_{ij} : 노드i에서 j에 대한 갱신 단위 비용

프로그램에서 화일에 대한 질의 통신비용

비용 8 $=\sum_{j,k,p,d} \alpha \lambda_{jpd} Q_{jk} X_{jkpd} (1+\gamma_{jk})$

α : 질의 메세지에 대한 확장인수(expansion factor)
λ_{jpd} : 노드j의 p가 수행될때 d에 대한 질의 트래픽 량
γ_{jk} : 노드j의 p에서 노드k에 있는 화일에 질의 요구에 대한 응답 추정비

프로그램에서 화일에 대한 갱신 통신비용

비용 9 $=\sum_{j,k,p,d} \beta \mu_{jpd} U_{jk} X_k^d$

β : 갱신 메세지에 대한 확장인수
μ_{jpd} : 노드j의 p가 수행될때 d에대한 갱신 트래픽량

전체 비용 $=\sum_{i=1}^{9}$ 비용i

전체 비용이 최소가 되도록 결정 변수의 해를 구하면 최적 할당이 되며 다음의 제약 조건을 만족하여야 사용 가능성(feasibility)을 갖는다.

노드간에 적어도 하나의 통신 회선이 존재

$\sum_{j,c} L_{ij}^c \geq 1 \quad i \in K$

원하는 화일에 대한 트랜잭션은 관계된 프로그램을 경유 하나의 정의된 루우트를 갖는다.

$\sum_j X_{jpd}^d = 1 \quad {}_j p \in p, \ d \in N$

$\sum_k X_{jkpd} = Z_j^p$

트랜잭션의 처리를 위해 특정한 화일과 프로그램에 대하여 루우트가 정의 되었다면 화일의 프로그램이 적당한 노드에 존재해야 한다.

$\sum_i X_{ijp}^d \leq n \cdot Z_j^p, \ \sum_j X_{jkpd} \leq n \cdot X_k^d$

모든 트랜잭션의 처리 요구가 각 노드에 있는 컴퓨터의 처리 능력을 초과해서는 안된다.

화일과 프로그램은 반드시 컴퓨터가 있는 노드에 존재해야 한다.

$X_k^d \leq \sum_m Y_k^m, \quad Z_j^p \leq \sum_m Y_j^m$

전 이중방식과 포인트-포인트 통신회선을 가정할 때 어느 방향이든지 전송량이 회선용량을 초과해서는 안된다.

사용가능한 해를 구하기 위해 화일과 프로그램의 사본이 적어도 하나 존재해야 한다.

$\sum_k X_k^d \geq 1, \ \sum_j Z_j^p \geq 1$

Ⅲ. FAP의 간편화 및 선 할당

앞장에서 서술한 목적함수와 제약 조건에서 입력 파라미터를 단일화 하고 제약조건을 완화하여 단수 화일 할당 문제의 해를 구하고자 한다.

컴퓨터의 용량이 일정하고 모든 노드에 컴퓨터가 존재한다고 가정하면 고정비용이 동일하다. 데이타 베이스 관리 시스템을 사용하지 않는 기존 화일체제에 대한 할당 이므로 임대, 설정 비용은 고려하지 않는다.

통신 회선은 완전 연결 구조로 설치 되어 있으며 회선 용량은 동일 하다고 가정하고 단 노드 간 거리 차이에 따라 통신비용이 달라진다.

프로그램과 데이타 화일은 동일한 노드에 존재하고 사용자 노드에서 화일 노드를 직접 액세스 하며 응답 트래픽(return flow)은 질의 요구와 동일하다고 가정한다. 컴퓨터의 처리능력 및 저장용량은 제약을 받지 않고 완전 결합된 네트워크이며 국부성을 갖기 때문에 회선용량은 충분하다고 가정한다.

모든 결정변수는 2진이며 제약조건이 아니라 최소 비용을 목적으로 하는 경우 정수치(integrality)를 갖는다.

이때 목적 함수는

$$Z = \sum_j \sum_k \lambda_j d_{jk} X_{jk} + \sum_k F Y_k \qquad (1)$$

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

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

$$F_k = \sigma_k + \sum_k \lambda_j d_{jk} \qquad (4)$$

$$0 \leqq \sum_{j \in P_k} X_{jk} \leqq n_k Y_k \qquad (5)$$

단수 화일 할당 문제에서 K0, K1은 화일의 사본을 가지고 있지 않는 노드, 사본을 가지고 있는 노드, K2는 할당 미 결정 노드의 집합이라 할 때 조건에 따라 할당할 노드를 탐색하여 할당 가능한 노드의 집합에서 화일 사본의 할당 여부를 결정한다.

FAP의 선 할당 조건은 창고(warehouse) 위치 선정 문제에서의 조건을 이용하거나 보강하여 최적해에 근사한 노드가 할당 되도록 한다.

1. 할당 조건

$$\lambda_i \min_{\substack{k \in k1 \cup k2 \\ k \neq i}} (d_{ik} - d_{ii}) > F_i \text{ 이면 } Y_i = 1 \qquad (6)$$

$$i \in K2,\ (f)_+ = \begin{cases} f & \text{if } f \geqq 0 \\ 0 & \text{if } f < 0 \end{cases}$$

$$\sum_j \lambda_j \min_{\substack{k \in k1 \cup k2 \\ k \neq i}} (d_{jk} - d_{ji}) > F_i \text{ 이면 } Y_i = 1 \qquad (7)$$

식(7)이 식(6)에 비해 할당 조건이 보강된 것임을 보이기 위해

$$\sum_{j=1}^{n} \lambda_j \min_{\substack{k \in k1 \cup k2 \\ k \neq i}} (d_{jk} - d_{ji})_+ \geqq \lambda_i \min_{\substack{k \in k1 \cup k2 \\ k \neq i}} (d_{ik} - d_{ii})_+$$

$$\text{좌변} = \lambda_i \min_{\substack{k \in k1 \cup k2 \\ k \neq i}} (d_{ik} - d_{ii})_+$$

$$+ \sum_{j=1} \lambda_j \min_{\substack{k \in k1 \cup k2 \\ k \neq i}} (d_{jk} - d_{ji})_+$$

$$\geqq \lambda_i \min_{\substack{k \in k1 \cup k2 \\ k \neq i}} (d_{ik} - d_{ii})_+ \quad = \text{우변}$$

2. 비할당 조건(I)

$$\sum_j \lambda_j \max_{k \in k1 \cup k2} (d_{jk} - d_{ji}) < F_i \text{ 이면 } Y_i = 0 \qquad (8)$$

$$\sum_j \lambda_j \min_{k \in k1} (d_{jk} - d_{ji}) < F_i \text{ 이면 } Y_i = 0 \qquad (9)$$

식(9)가 식(8)에 비해 보강된 비 할당 조건임을 보이기 위해

$$\sum_j \lambda_j \min_{k \in k1} (d_{jk} - d_{ji})^+ \text{와} \sum_j \lambda_j \min_{k \in k1 \cup k2} (d_{jk} - d_{ji}) \text{의}$$

각 항을 비교하여 다음식을 만족하면 된다.

$$\min_{k \in k1} (d_{jk} - d_{ji})_+ \leqq \max_{k \in k1 \cup k2} (d_{jk} - d_{ji})$$

3가지 경우로 나누어서

1) $d_{ji} \leqq \min_{k \in k1} d_{jk}$ 이면 $\min_{k \in k1} (d_{jk} - d_{ji})_+$

$$= \min_{k \in k1} (d_{jk} - d_{ji}) \leqq \max_{k \in k1 \cup k2} (d_{jk} - d_{ji})$$

2) $\min_{k \in k1} d_{jk} < d_{ji} < \max_{k \in k1} d_{jk}$ 이면

$$\min_{k \in k1} (d_{jk} - d_{ji}) = 0,\ \max_{k \in k1 \cup k2} (d_{jk} - d_{ji}) \geqq 0$$

그러므로 항상 좌변≦ 우변이 성립한다.

3) $\max_{k \in k1} d_{jk} < d_{ji}$ 이면 $\max_{k \in k1} (d_{jk} - d_{ji}) < 0$

$$\text{좌변} = \min_{k \in k1} (d_{jk} - d_{ji})_+ = 0$$

$$\text{우변} = \max_{k \in k1 \cup k2} (d_{jk} - d_{ji}) =$$

$$\max[\max_{k \in k1} (d_{jk} - d_{ji}),\ \max_{k \in k2} (d_{jk} - d_{ji})]$$

$\max_{k \in k1} (d_{jk} - d_{ji}) > \max_{k \in k2} (d_{jk} - d_{ji})$ 이면

우변은 항상 음이므로 좌변>우변

$\max_{k \in k1} (d_{jk} - d_{ji}) < \max_{k \in k2} (d_{jk} - d_{ji})$ 이면

우변이 음과 양이 될 수 있으므로 식(9)를 비 할당 조건으로 사용하려면

$d_{ji} \leqq \max_{k \in k1} d_{jk}$ 이 성립할 때이다.

3. ni의 감소기법(I)

$i \in K2,\ j \in Pi$에 대하여

$$\alpha_i = \min_{k \in k1} (d_{jk} - d_{ji})$$

$\alpha i < 0$이면 $Pi = Pi - \{j\},\ ni = ni - 1$

사용자 j가 기존할당 노드k에 액세스하는 것이 노드i에 대한 액세스보다 경제적이므로 집합 Pi에서 j를 삭제하고 ni를 1감소 시킨다.

4. ni의 감소기법(II)

$i \in K2,\ j \in K0 \cup K2,\ K \in K1 \cup K2$에 대하여

$$\lambda_j d_{jk} + f_k / n_k = \min_{1 \in k1 \cup k2} (\lambda_j d_{j1} + f_1 / n_1) \qquad (10)$$

식(10)을 만족하면 $X_{jk} = 1$이고 아닌 경우 $X_{jk} = 0$

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

$X_{jk} = 1$인 경우,

$k \notin K2$이면 $Pi = Pi - \{j\},\ ni = ni - 1$

사용자 j가 노드k의 화일을 액세스 할때 k가 K2

의 원소가 아니면 집합 Pi에서 j를 삭제하고 ni를 1감소 시킨다.

5. 비할당 조건(Ⅱ)

할당의 초기화 과정에서 즉 모든 노드가 할당 미결정 일 때 노드i, k에 대하여 통신비용의 차의 합이 저장비용의 차보다 작을때 노드i에는 화일사본을 비할당한다.

$Fi-FK>\sum_j \lambda_j(d_{jk}-d_{ji})_+$ 이면 $Yi=0$

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

Ⅳ. 할당 알고리듬

1. 할당 알고리듬

집합 K0, K1, K2, K3=K1∪K2, K4=K0∪K2, K5=K0∪K1를 알고리듬의 프로그램 수행 과정에는 배열구조(array structure)표현이 편리하며 할당 알고리듬의 순서를 단계별로 나타내면 다음과 같다.

[단계 1] 모든 노드에 화일이 할당 미결정 상태로 초기화 하여 노드의 할당상태가 "U"가 된다.

[단계 2] 할당의 첫 단계에서 문제의 규모를 축소하기 위하여 입력 파라미터의 초기치에 대하여 비 할당 가능한 노드가 존재하는지 검색하고 조건을 만족하는 노드는 항상 할당상태 "0"을 유지한다.
조건을 만족하는 노드가 많을수록 문제 규모가 대폭 줄어든다.

[단계 3] 초기에는 단계2]할당이 진행중인 경우에는 단계9]에서의 할당 상태에 대하여 선 할당을 수행하여 할당 가능한 노드의 존재를 결정하고 조건을 만족하는 노드는 할당상태 "1"을 유지한다.

[단계 4] 단계3)의 할당 상태에 대하여 비 할당조건(Ⅰ)을 수행하여 비 할당 가능한 노드의 존재를 검색한다. 조건을 만족하는 노드는 할당상태 "0"을 유지한다.

[단계 5] 할당 미 결정노드, 할당 상태 "U"비트 즉 K2[ ]의 "1"비트에 대하여 순차적으로 0, 1로 세트 시킨다.
K0[ ], K1[ ]의 상태도 따라서 변화한다.

[단계 6] 단계5)에서의 순차적으로 세트시킨 각각의 할당 상태에 대하여 평가치를 구하고 u-cnt를 1씩 증가 시킨다. 최대 u-cnt는 단계4]의 할당 상태 "U"의 갯수의 2배가 된다.

[단계 7] 현재 u-cnt가 최대 u-cnt 인지 비교하여 작으면 평가치 계산이 미 완료 이므로 단계5]를 수행하고 동일하면 단계8]을 수행한다.

[단계 8] 평가치와 할당 상태에 대하여 원하는 선택기준에 의해 노드의 할당 상태를 "0" 혹은 "1"로 결정한다.
각각의 선택기준에 따라 MIN, MAXD인 경우 반드시 노드1개의 할당 상태가 결정되고 CMIN인 경우는 동시에 2개의 노드의 할당 상태가 결정된다.

[단계 9] 단계8]에서 할당 상태의 "U"비트 존재 여부를 검색하고 계속 존재하면 단계3]을 수행하고 K2[ ]의 모든 비트가 "0"이면 할당이 종료이므로 평가치가 전체비용이 되고 할당 상태가 결정된다.

할당 조건, 비 할당 조건의 수행중 단계2], 3], 4]에서 할당 상태의 "U"비트가 존재하지 않으면 할당 완료이므로 평가치를 구하고 할당 상태를 결정한다.

2. 평가치 계산기법

1) APP(appraisal value)

[단계 1] 할당 알고리듬의 단계5]에서 0, 1로 세트된 후보 노드의 할당 상태 배열을 구성한다.

[단계 2] n_i의 감소기법(Ⅰ)을 수행하여 조건을 만족하는 노드j를 사용자 상태 배열에서 제거하고 n_i를 1감소한다.

[단계 3] n_i의 감소기법(Ⅱ)를 수행하여 제어변수가 1인 경우 노드k가 K2[ ]의 원소가 아니면 사용자 상태 배열에서 k를 제거하고 n_i를 1감소한다.

[단계 4] 평가치 $1=\sum_j(\sum_K \lambda_j d_{jk} X_{jk}+X_{jj}F_j/n_j),\ j\in K4$

평가치 $2=\sum_j F_j,\ j\in K1$

최종 평가치는 평가치 1, 2의 합으로 나타낸다.

2) EE(estimated expected value)

후보 노드의 할당 미 결정 노드에 대하여 저장 비용과 통신 비용을 기대값으로 표현한다. 기대값이 정규분포를 갖는다고 가정하고 평균 및 표준편차를 평가치를 구하는데 이용된다.

[단계 1] 할당 알고리듬의 단계5]에서 0, 1로 세트된 할당 상태에 대하여 노드 상태 배열을 구성한다.

[단계 2] 비 할당 노드, 할당 미 결정 노드에서의 통신비용에 대한 기대값을 구한다.

기대값 $1=\sum_{j\in K_0\cup K_2} \lambda_j E(\min_{k\in I} d_{jk})$

[단계 3] 할당 미결정 노드에 대한 저장 비용의 기대값을 구한다.

$$\text{기대값 } 2 = \sum_{k \in k2} F_k \, E(Y_k)$$

[단계 4] "U"의 갯수의 2배, u-cnt에 해당하는 통신 비용의 기대값에 대한 평균을 구한다.

$$\text{평균치} = \sum_{u\text{-}cnt} \sum_{j \in k0 \cup k2} \lambda_j E(\min_{k \in 1} d_{jk})/K4[\ \ ]\text{의 "1"갯수}$$

[단계 5] 평가치 $1 = \sum_j \lambda_j$, 평균치/C(K), $j \in K4$

평가치 $2 = \sum_j F_j$, $j \in K1$

$$C(K) = \begin{cases} 2^{n-|\overline{K2}|-1} & \text{if } |K1| = 0 \\ 2^{n-|\overline{K2}|} & \text{if } |K1| > 0 \end{cases}$$

$|\overline{K2}| = K0[\ \ ] \cup K1[\ \ ]$의 "1"의 갯수

EE=평균치1+평균치2+기대값2로 표현한다.

3. 선택 기준

1) 최소치 선택(MIN)

할당 알고리듬의 단계7]에서 u-cnt의 후보 노드중 최소치를 갖는 할당 상태를 선택한다.

2) 최대차 최소 선택(MAXD)

할당 알고리듬의 단계7]에서 0, 1로 세트된 할당 상태의 평가치 차가 가장 큰 후보의 짝을 선택하고 최소치를 갖는 할당 상태를 선택한다.

3) 최소치의 결합(CMIN)

할당 알고리듬의 단계7]에서 0으로 세트된 할당 상태중 최소치를 갖는 상태와 1로 세트된 후보 상태중 최소치를 갖는 상태를 2진 연산으로 처리하여 할당 상태를 결정한다. 즉 u-cnt가 홀수인 상태중 최소치와 짝수인 상태중 최소치를 갖는 상태를 결합한다.

V. 결과및 성능 평가

화일 할당 알고리듬의 성능평가는 문헌에 나타난 할당 문제의 여러 예제에 대하여 최적 해와의 편차, 수행시간을 결과로 나타냈다.

표 1. 액세스 요구와 비용 파라미터
Table 1. Access requests and cost parameters.

S \ D	통신비용/Mbyte 1	2	3	4	5	질의 트래픽	갱신 트래픽
1	0	6	12	9	6	24	2
2	6	0	6	12	9	24	3
3	12	6	0	6	12	24	4
4	9	12	6	0	6	24	6
5	6	9	12	6	0	24	8

단수 화일 할당 문제의 가장 기본적인 CASEY의 5노드 예제에 대한 입력 파라미터를 표1에 나타냈고 S, D는 출발지와 목적지를 표시한다.

제안한 평가치 계산기법과 다양한 선택기준을 이용하여 할당 알고리듬을 수행 하였으며 열거법에 의한 최적 해는 할당 상태가 [1, 4, 5]이고 총비용 705를 나타낸다. CMINAPP 기법과 MINEE를 이용한 할당 과정을 그림1에 나타냈다. 할당 초기에 할당조건을 만족하는 4, 5는 할당이 결정 되었으며 후보 노드 1, 2, 3에 대하여 선택 기준에 따라 할당 한다.

그림a)의 노드2는 비할당 조건에 의하여 그림b)의 노드1은 할당조건에 의해 결정된다.

최적 해와의 편차를 가지고 알고리듬의 성능을 평가하고 수행결과를 표2에 나타냈으며 제안한 알고리듬은 다항식 시간 알고리듬으로 VAX 11/750에서 컴파일 시간을 제외한 프로그램 수행시간을 표3에 나타냈다.

예제1은 앞에서 보인 5노드 문제이고 예제 2-6은 19노드의 ARPANET[3]에 대한 예로 질의 트래픽

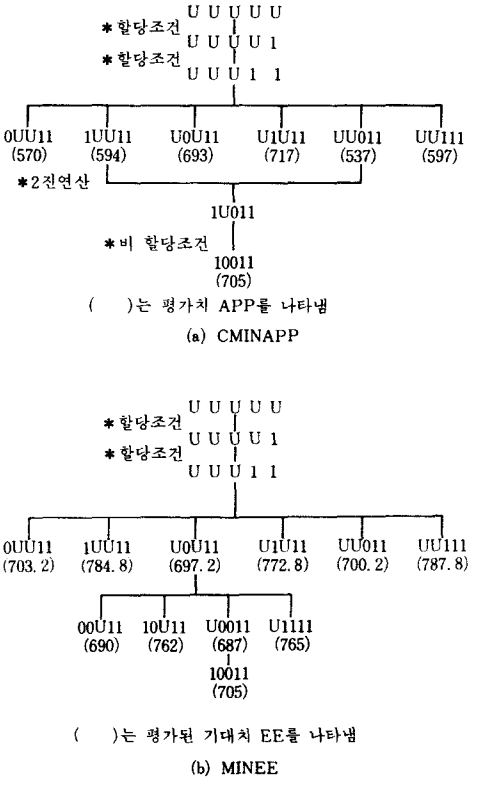

그림 1. 5노드 예제에 대한 평가
Fig. 1. Evaluation of 5 node example.

량에 대한 갱신 트래픽량을 0.1, 0.2, 0.3, 0.4, 1.0 으로 변화시키면서 할당을 실행하고 예제7-8은 Mahmoud[4]의 9노드 예제이며 예제9-12는 예제 2-6의 갱신 트래픽량을 전체적으로 0.1로 감소시킨 경우이다.

최적 해를 구하는데 있어 $0(n^4)$의 복잡도를 갖는 MAXDEE 기법이 가장 좋은 결과를 보였고 80%의 최적 해를 구할 수 있었다.

예제9-12의 최적해는 국부적 최적해의 가정이며 CMINAPP 기법은 갱신 트래픽량과 저장 비용이 질의 트래픽에 비해 작은 경우 할당수행중 빠른 시간에 최적 해에 근사한 결과를 얻을 수 있다.

갱신, 저장, 질의 비용의 상태적인 크기를 비교하여 화일 사본의수가 적은 경우는 MINE 기법이나 MAXDEE를 사용하고 화일 사본의수가 전체 노드의 0.5인 경우 CMINAPP 알고리듬이 우수함을 나타냈다.

표 2. 할당 알고리듬의 최적 해에 대한 퍼센트 편차

Table 2. Percent deviation of allocation algorithm from optimal solution.

	최적해	MAXDEE		M AXDLB		MINE	
		비 용	편차	비 용	편차	비 용	편차
1	705	717	1.7	717	1.7	705	0
2	117896	117896	0	117896	0	127805	8.4
3	188738	189929	0.63	189929	0.63	197160	4.4
4	243480	243480	0	272208	11.7	243480	0
5	292990	292990	0	297032	1.8	292990	0
6	437720	437720	0	437720	0	437720	0
7	27695	27695	0	29788	7.5	27698	0.01
8	203585	206066	1.2	206066	1.2	206066	1.2
9	22886	22886	0	23322	1.9	23036	0.65
10	39323	39323	0	39323	0	39323	0
11	51012	51012	0	51012	0	51012	0
12	61982	61982	0	61982	0	61982	0
평 균			0.29		2.2		1.22
표준 편차			0.55		3.19		2.29

표 3. 할당 알고리듬의 VAX 11/750에서 수행시간

Table 3. Execution time of allocation algorithm on the VAX 11/750.

[min : sec : csec]

	MAXDEE	MAXDLB	MINE
1	00 : 00 : 2	00 : 00 : 2	00 : 00 : 3
2	02 : 29 : 6	02 : 59 : 7	02 : 16 : 2
3	01 : 57 : 8	02 : 40 : 3	02 : 02 : 4
4	01 : 34 : 4	02 : 05 : 3	01 : 25 : 1
5	01 : 07 : 2	01 : 43 : 7	01 : 10 : 0
6	00 : 03 : 3	00 : 04 : 6	00 : 02 : 9
7	00 : 01 : 3	00 : 02 : 1	00 : 01 : 1
8	00 : 00 : 5	00 : 00 : 5	00 : 00 : 4
9	01 : 29 : 1	02 : 21 : 3	01 : 45 : 3
10	01 : 31 : 7	02 : 33 : 3	01 : 41 : 8
11	01 : 42 : 4	02 : 37 : 3	01 : 45 : 3
12	02 : 01 : 8	02 : 50 : 8	02 : 07 : 6
평 균	01 : 09 : 1	01 : 39 : 9	01 : 11 : 5
표준편차	00 : 51 : 8	01 : 11 : 8	00 : 52 : 5

Ⅵ. 결 론

분산 컴퓨터 시스템의 최적 화일 할당을 위한 발

```
# include "minapp.h"
main( )
{
  int n, i ;
 first-initial( ) ; /*모든 노드를 K2의 원소로 초기화*/
 computer-efcost ( ) ; /* 저장 비용과 갱신 비용*/
 pre-assign-d ( ) ; /*비 할당 조건(Ⅱ)*/
 while (1)
{
 pre-assign-a ( ) ; /*할당 조건*/
 n=check-u-cnt ( ) ;
 if (n=0) /*할당 상태에서 "U"가 없으면 종료*/
{
 u-cnt=0 ;
 compute-user-number ( ) ; /*사용자 상태 배열*/
 deter-fraction ( ) ; /*제어 변수 결정*/
 for (i=0 i<19 i++)
   ass-st[a-ind][i]=temp-b[i]=disp-a[i]
                    /*할당 상태 구성*/
   compute-cost ( ) ;  /*전체 비용*/
   most-min=repre--value[u-cnt] ; /*최소치 선택*/
   break ;
}}
  pre-assign-b ( ) ; /*비 할당 조건*/
  cand-list ( ) ; /*후보 노드 선정*/
  select-rule ( ) ; /*선택 기준*/
  n=check-u-cnt ( ) ;
  if (n==0)
    breck ;
 }
  print f ("n Most Minimum Value=% f : Assign-state
  =[%s] n," most-min, ass-st[a-ind]) ;
}
```

그림 2. 할당 알고리듬의 주 프로그램

Fig. 2. Main program of allocation algorithm.

견적 알고리듬을 제안하였고 모델링 상황을 실제 응용에 근사하도록 하였으며 후보 노드의 할당 상태에 대한 평가치및 기대값을 구하는 방법을 제안하였다.

제안한 할당 알고리듬은 선택기준의 설정과 평가치 계산의 2부분으로 구성했으며 할당 초기에 선할당을 수행하여 문제규모를 축소하고 제안한 기법으로 평가한 할당 상태에 대하여 최대차를 갖는 할당 상태에 대하여 최소치를 갖는 후보 선택 기준을 적용하므로써 할당 소요시간을 하한치를 선택하는 방법에 비해 30% 개선하였다. 선택기준에 있어 CMIN방법은 빠른시간에 할당이 결정되며 평가치의 EE기법은 후보 노드에 대하여 정확한 표시치를 얻을 수 있다. 컴퓨터 시스템의 노드가 많은 경우 19노드에 대하여 2분 이내의 소요 시간으로 최적해를 구하므로써 열거법에 비해 화일 할당 문제에의 실제적용이 용이하다.

화일 할당 문제에 있어 CASEY의 5노드 예제 등 여러 예제에 대하여 할당상태의 기대치를 이용하는 방법의 제안 알고리듬과 제어변수의 결정에 따른 평가치 계산 기법을 C 언어로 프로그램하여 VAX 11/750에서 할당 상태, 총 비용 및 사용자 시간을 실현한 결과 할당에 소요되는 시간을 개선하였으며 높은 확률의 최적해를 구하므로써 알고리듬의 우수성을 입증하였다. 앞으로 네트워크의 사용자들이 편리하도록 동적 모델의 할당 알고리듬과 적응 라우팅기법을 이용하는 컴퓨터 시스템의 화일 할당 문제에 대한 연구가 필요하다고 생각한다.

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 informationnetworks," *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 warenhouse 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 systems," *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. Comput.*, vol. C-32, no. 3, pp.2221-231, Mar. 1983.

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

[13] Whiteney, V.K.M. "A study of optimal file assignment and communication network configuration," Ph. D. Dissertation, Dept. Electrical Eng., Univ. of Michigan, Sept. 1970.

[14] 홍진표, 임제탁,"분산 컴퓨터 시스템에서 효율적 화일 할당에 관한 연구,"대한전자 공학회 논문지 제26권, 제 9 호, 1989.

著 者 紹 介

洪 進 杓 (正會員) 第26卷 第9號 參照
현재 한양대학교 전자공학과 박사과정

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