

# 분산 컴퓨터 시스템에서 화일 할당에 관한 연구

## (A Study on the File Allocation in Distributed Computer Systems)

洪 進 杓\*, 林 濟 鐸\*\*

(Jin Pyo Hong and Chae Tak Lim)

### 要 約

분산 컴퓨터 시스템에서 비 결정성(non-deterministic) 프로세스 그래프에 대한 동적(dynamic) 재 할당 알고리즘을 제안한다.

프로세스 나무(tree)를 수행하는데 있어 최적 정책을 결정하는 기법을 제안한다. 일반적인 데이터베이스 질의 요구를 우선 순위 관계가 존재하는 서브프로세스의 집합을 나타내는 프로세스 나무로 표현한다. 프로세스 할당 모델은 서브 프로세스의 수행 위치(computer site) 선택, 데이터 감소(reduction) 함수와 화일 크기의 함수인 수행비용에 근거한다. 비 결정성 프로세스 나무에 대하여 파라미터의 기대값을 이용하여 수행비용을 최소로 하는 프로세스 그래프와 할당 정책을 결정한다. 결정된 프로세스 그래프에서 낮은 레벨의 프로세스를 수행한후 임계값과 파라미터의 새로운 정보에 따라 프로세스를 재 할당 하므로써 최적해에 근사한 할당상태를 얻는다.

제안한 알고리즘은 발견적 알고리즘이며 예제에 실현한 결과 최적해를 구하는데 우수함을 나타냈다.

### Abstract

A dynamic relocation algorithm for non-deterministic process graph in distributed computer systems is proposed.

A method is represented for determining the optimal policy for processing a process tree. A general database query request is modelled by a process tree which represent a set of subprocesses together with their precedence relationship. The process allocation model is based on operating cost which is a function of selection of site for processing operation, data reduction function and file size. By using expected values of parameters for non-deterministic process tree, the process graph and optimal policy that yield minimum operating cost are determined.

As process is relocated according to threshold value and new information of parameters after the execution of low level process for non-deterministic process graph, the assigned state that approximate to optimal solution is obtained.

The proposed algorithm is heuristic. By performing algorithm for sample problems, it is shown that the proposed algorithm is good in obtaining 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: 1990年 1月 16日

(※ 본 연구는 1989년도 한양대학교 교내 연구비의 지원으로 이루어진 것임.)

### I. 서 론

분산 컴퓨터 시스템에서 프로세스 그래프의 파라미터가 결정되지 않은 확률적 모델에 대하여 처리비용과 통신 비용으로 구성되는 목적함수를 최소로 하면서 프로세스를 수행하도록 서브프로세스를 컴퓨터 위치에 할당한다.

컴퓨터 시스템은 매우 동적인 시스템으로 프로세스를 서술하는 파라미터는 일정한 시간에는 유효하지만 시간이 경과된 후에는 부정확하다. 프로세스를 서브프로세스의 집합으로 나타낼 수 있으며 서브프로세스의 처리결과를 정확한 위치의 수신자에게 전송하므로써 프로세스 수행을 완료한다. 이러한 서브프로세스의 집합을 컴퓨터의 여러 위치중 최적의 위치에 할당한다. 프로세스 할당 문제에 대하여 2개의 프로세스에 max-flow min-cut 알고리즘을 이용하여 전체 비용이 최소가 되도록 프로세스를 할당하는 방법<sup>[1]</sup>과 k-프로세스 시스템에서 multi-terminal max-flow 방법으로 프로세스간의 통신비용만을 고려한 할당<sup>[2]</sup> 등 여러 종류의 할당 기법이 제시되었다. 위에서 제시한 방법은 프로세스의 파라미터가 미리 결정된 상황의 모델이며 우선 순위 관계를 고려하지 않은 프로세스 할당 기법이다.

본 논문에서는 데이터베이스 질의 요구를 표현하는 프로세스 나무의 서브프로세스 할당에 있어 발견적 방법으로 프로세스 재 할당 알고리즘을 제안하고 프로세스 나무의 수행 과정에서 다양한 프로세스 그래프를 생성하여 최적 정책을 결정하는 기법을 제안하였다. 서브프로세스의 처리 과정에서 파라미터에 대한 새로운 정보로 프로세스 재 할당을 행하며 파라미터의 기대값과 실제값의 편차가 일정한 임계값을 넘으면 프로세스를 재 할당한다.

제안한 최적 정책 결정 기법 및 재 할당 알고리즘을 C언어 프로그램으로 VAX 11/750에서 예제에 대하여 임계값을 변화 시키면서 50번씩 실행한 결과 최적해를 구하는데 있어 우수함을 입증하였다.

II 장에서는 프로세스 그래프의 생성을 III 장에서는 수학적 모델과 목적함수를 제시하였다. IV 장에서는 최적 정책 결정기법과 재 할당 알고리즘을 V 장에서는 예제 및 성능평가를 나타냈다.

## II. 프로세스 그래프의 생성

분산처리 시스템에서 통신 비용을 줄이기 위해 하나의 컴퓨터 위치에서 연산의 그룹을 수행하는 것이 바람직하다.

일종의 클러스터화로 다음의 처리를 위해 연산의 결과를 다른 위치에 전송한다. 프로세스 나무에 있는 연산(서브프로세스)의 집합을 그룹으로 분류하고 선택된 컴퓨터 위치에서 수행한다. 프로세스를 처리할 때 소요되는 통신 비용은 선택된 그룹들 사이의 통신 비용을 합하면 된다.

그러므로 프로세스 나무를 연산의 집합으로 구성

된 프로세스 그래프로 변환하며 다양한 프로세스 그래프를 생성한다. 그림1에 데이터베이스 질의 요구를 표현하는 서브프로세스 m, 컴퓨터 위치 n과 정규 화계수 w로 구성되는 프로세스 나무와 컴퓨터 네트워크를 나타냈다.

프로세스 그래프에서는 초기 연산(4, 5, 6, 7, 8, 9에 해당), 중간연산(1, 2, 3에 해당)과 최종 연산(0에 해당)의 3형태로 구분하며 수행노드와 저장노드로 구성한다. 원으로 표시한 수행노드는 한 위치에서 연산 그룹의 수행을 나타내며 수행노드를 연결하는 화살표는 위치간의 데이터 통신을 나타낸다. 삼각형으로 표시한 저장노드는 특정한 위치에 화일이 저장되어 있음을 나타내고 동시에 화일이 저장되어 있는 노드에서 수행되는 초기연산 이다. 그러므로 초기연산은 화일의 원하는 부분집합을 선택한다. 수행노드와 연결된 화살표는 일시적으로 생성된 화일로 수행노드의 입력 데이터가 된다.

그러므로 컴퓨터 위치에 따른 단독 연산이나 그룹 연산에 관한 정보와 연산의 순서를 프로세스 그래프로 나타낼 수 있다.

프로세스 나무의 기저 레벨에서 상위 레벨로 가면 각 레벨에 대하여 이웃 연산과 결합하면서 연산 그룹의 집합을 생성한다. 저장노드를 제외한 중간연산과 최종연산에 해당하는 프로세스 나무에 대하여  $L_i$ 는 프로세스 나무(tree)의 레벨 i에 있는 연산(operation) 또는 프로세스를 포함하는 국지적(local 연산 그룹의 집합으로 정의하며 레벨 i 보다 높은 레벨에 있는 연산은 포함하지 않는다. 그러므로 집합  $L_i$ 의 원소는 레벨 i에 있는 국지적 연산 그룹이다.

$L_r$ 은 뿌리노드를 포함하는 연산 그룹의 집합이고 그림 1(a)에서 4개의 연산  $B_0, B_1, B_2, B_3$ 이 존재한다.  $B_0, B_1, B_2, B_3$ 는 각각 프로세스 나무의 프로세스 0, 1, 2, 3에 해당하며 즉 중간연산과 최종연산을 나타낸다. 레벨  $i=1$ 에는 연산  $B_3$ 만이 레벨  $i=2$ 에서는  $B_2, B_1$ 을 포함하는 연산 그룹의 집합이 마지막으로 뿌리노드를 포함하는 연산 그룹의 집합은  $L_r$ 이다.

$$L_1(B_3) = L_1 = \{B_3\}$$

$$L_2(B_2) = \{B_2\}$$

$$L_2(B_1) = \{B_1, B_3 B_1\}$$

$$L_2 = \bigcup L_2(B_i) \text{에서 } L_2 = \{B_2, B_1, B_3, B_1\}$$

$$L_3(B_0) = L_r = \{B_0, B_2, B_0, B_1, B_0, B_3, B_1, B_0\}$$

전체 프로세스 나무에 대한 연산 그룹의 집합은

$$L = \bigcup_{i=1}^3 L_i \cup L_r = \{B_3, B_2, B_1, B_0, B_3, B_1, B_2, B_0, B_1, B_0, B_3, B_1, B_0\}$$

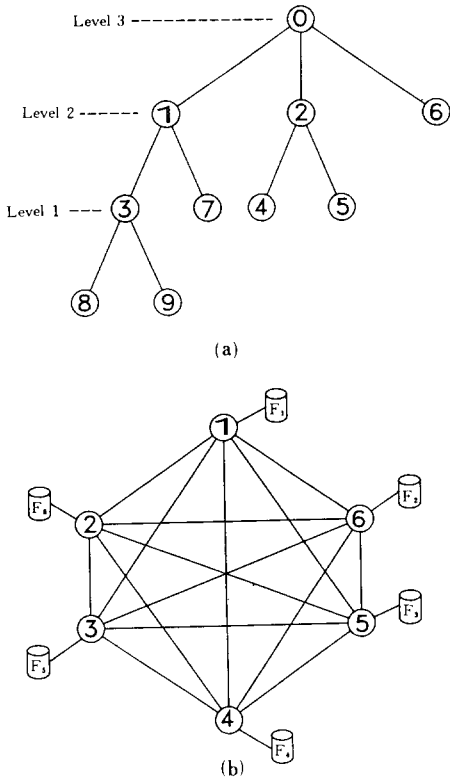


그림 1. 10노드 프로세스 나무와 6위치 컴퓨터 네트워크

- (a) 10副프로세스의 프로세스 트리
- (b) 通信網 構造와 화일 位置

Fig. 1. 10 node process tree and 6 site computer network.

- (a) Process tree with 10 subprocesses.
- (b) Network structure and file site.

프로세스 나무의 레벨 1에는 프로세스 3, 7, 4, 5가 존재하나 잎(leaf) 프로세스는 위에서 정의한 초기 연산 또는 저장 노드 이므로 연산 그룹의 집합에서 제외한다.

그러므로 레벨 1에서의 연산 그룹의 집합  $L_1$ 의 원소는 프로세스 3에 해당하는 연산  $B_3$  뿐이다. 레벨 2에서는 프로세스 1, 2, 6이 존재하며 잎 프로세스 6을 제외한 프로세스 1, 2에 해당하는 연산  $B_1$ 과  $B_2$ 에 대하여  $B_2$ 를 기준으로 한 집합  $L_2$ 의 원소는  $B_2$  뿐이며  $B_1$ 을 기준으로 한 집합  $L_2$ 의 원소는  $B_1, B_3, B_1$  이다.

그러므로 레벨 2에서의 국지적 연산 그룹의 집합  $L_2$ 의 원소는  $B_2, B_1, B_3, B_1$  임을 나타내며 프로세스 실행에 있어  $B_2, B_1$ 은 각각 다른 노드에서 처리하며  $B_3, B_1$ 은 동일한 노드에서 처리할 수 있음을 나타낸다. 프로세스 나무의 상위레벨에서도 위의 과정을

반복하면 전체 연산 그룹의 집합  $L$ 의 원소는  $B_3, B_2, B_1, B_0, B_3, B_1, B_2, B_0, B_1, B_0, B_3, B_1, B_0$ 의 8개이다.

각 원소는 노드에서 처리될 수 있는 연산 또는 연산의 그룹을 나타내며 예를 들어 원소  $B_3, B_1, B_0$ 는 연산  $B_3, B_1, B_0$ 가 동일한 노드에서 처리됨을 나타낸다. 위에서 생성된 연산 그룹의 집합으로부터 생성된 프로세스 그래프를 그림 2에 나타냈다.

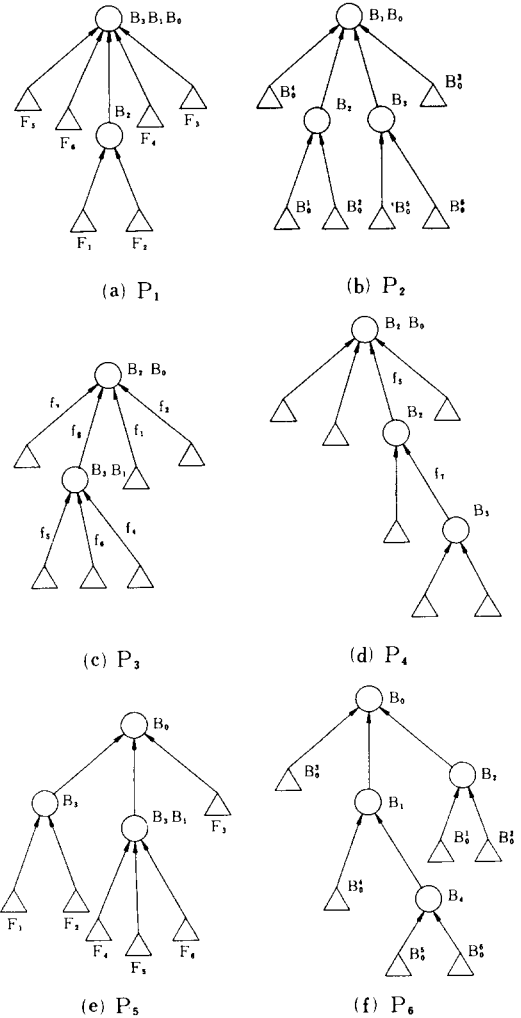


그림 2. 生成된 프로세스 그래프  
Fig. 2. Process graph generated.

### III. 수학적 모델과 목적함수

프로세스 나무에 대하여 최적 할당이란 목적함수를 최소화 하는 서브프로세스의 할당이며 목적함수는 처리비용과 통신 비용으로 구성되는 비용 함수이다.

처리 비용이란 다양한 컴퓨터 위치에서 서브프로세스를 처리하는데 소요되는 비용이며 통신 비용은 컴퓨터 위치에서 수행되는 서브프로세스 사이의 정보교환으로 트래픽량과 거리의 곱으로 나타낸다.

최적화 모델에 필요한 구성 요소는 프로세스 그래프에 대한 서술, 서브프로세스 사이의 전송량, 컴퓨터 위치간의 거리와 각 위치에서의 서브프로세스 처리 비용이다. 처리 비용과 통신 비용은 데이터 감소 함수와 화일 크기에 따라 달라지며 데이터 감소 함수  $R(B_j, F, C)$ 는 연산  $B_j$ 의 입력 데이터량, 입력화일  $F$ 와 연산 파라미터  $C$ 에 대한 출력 데이터량의 비율이다. 입력화일  $F$ 와 연산 파라미터  $C$ 를 갖는 연산  $B_j$  다음의 연산  $B_i$ 의 데이터 감소 함수는  $R(B_i | B_j, F, C)$ 로 표기한다.

$Q_{ij}$ 는 연산  $B_i$ 의  $Q_j$ 는 연산  $B_j$ 의 단위 입력 화일에 대한 처리 비용이라 하고  $C_{ij}$ 는 화일  $f_i$ 를 출력하는 위치에서 연산  $B_j$ 의 위치까지 단위 데이터에 대한 통신 비용이라 한다.

일시적 화일  $f_i$ 가 연산  $B_j$ 에 대한 입력 인지를 표시하는  $Y_{ij}$ 를 사용하고

$$Y_{ij} = \begin{cases} 1 & \text{화일 } f_i \text{가 연산 } B_j \text{의 입력이면} \\ 0 & \text{그렇지 않은 경우} \end{cases}$$

$f_i$ 가  $B_j$ 를 수행하는 위치에서 이용 가능한가를 표시하는  $X_{ij}$ 를 사용한다.

$$X_{ij} = \begin{cases} 1 & f_i \text{가 필요하나 } B_j \text{를 수행하는 위치에서} \\ & \text{이용 불가능 하면} \\ 0 & \text{그렇지 않은 경우} \end{cases}$$

이와 같이  $X_{ij}=1$ 이면 반드시 화일  $f_i$ 는  $B_j$ 가 수행되는 위치에 전송되어야 한다.

할당 정책  $P$ 에 대한 프로세스 그래프의 비용함수는

$$Z(P) = \sum_{i=1}^s Q_i^0 l(F_i) + \sum_{j=1}^k Q_j \sum_{i=1}^q l(f_i) Y_{ij} + \sum_{j=1}^k C_{ij} \sum_{i=1}^q l(f_i) X_{ij} \tag{1}$$

$$C_{ij} = w \cdot D_{ij} \tag{2}$$

- $l(F_i)$  = 화일  $F_i$ 의 크기
- $l(f_i)$  = 일시적 화일  $f_i$ 의 크기
- $q$  = 일시적 화일의 수
- $k$  = 연산  $B$ 의 수
- $w$  = 정규화 계수
- $D_{ij}$  = 위치  $i, j$  사이의 거리

식(1)의 첫째항은 초기연산의 단위 처리 비용과 입

력 화일로 나타낸 처리비용이고 둘째항은 중간 연산  $B_j(j=1, 2, \dots, k)$ 의 처리 비용이며 셋째항은 프로세스 그래프의 화일표에 해당하는 데이터의 교환에 기인한 단위 통신비용과 입력 데이터로 나타낸 통신비용이다. 즉 목적함수  $Z(P)$ 는 처리 비용과 통신 비용으로 구성되는 프로세스 처리의 비용 함수이다.

식(2)는 노드  $i, j$  사이의 단위 통신 비용으로 노드 사이의 거리  $D_{ij}$ 에 비례하며 통신 비용과 처리비용과 동일한 유니트로 구성하기 위하여 정규화(normalization) 계수를 곱하여 사용한다.

식(1)과 같이 비용함수를 정의 하였을때 최적 할당이란 다음식을 만족하는 정책을 결정하는 것이다.

$$Z(P') < Z(P), \forall P \tag{3}$$

즉 주어진 프로세스 나무에 대하여 국지적 연산그룹 집합의 원소들로 구성된 프로세스 그래프에 대하여 목적함수  $Z(P)$ 를 최소로 하는 프로세스 그래프  $P'$ 를 선택한다.

비 결정성 프로세스 그래프 즉 확률적 모델에 대한 최적 할당이란 실행의 기대값을 최소로 하는 정책과 할당 상태를 결정하는 것이다.

$$E [Z(P')] < E [Z(P)], \forall P \tag{4}$$

$E [X_1 + X_2 + \dots + X_i] = E[X_1] + E[X_2] + \dots + E[X_i]$  특성과 처리비용  $Q$ 와 화일 크기  $l(f)$ 가 독립적 확률변수로 가정하면

$$E[Z(P)] = \sum_{i=1}^k [E[Q_i^0] \sum_{i=1}^q E[l'(f_i)] Y_{ij}] + C_{ij} \sum_{i=1}^q E[l'(f_i)] X_{ij} + \sum_{i=1}^s E[Q_i^0] E[l'(F_i)] \tag{5}$$

식(5)에서 악센트 부호는 확률 변수를 표시한다. 위에서 식(4)와 식(5)는 확률적 모델 즉 프로세스 그래프를 구성하는 파라미터의 값이 결정되지 않은 상황이므로 파라미터의 초기 기대값을 이용하여 목적함수를 구성하고 프로세스 실행의 기대값을 최소로 하는 프로세스 그래프를 결정한다.  $E[ ]$ 는 프로세스 실행의 기대값을 나타내며 식(5)의 첫째항은 중간 연산의 처리 비용을 둘째항은 연산이 실행되는 노드  $i, j$  사이의 통신 비용을 셋째항은 초기 연산의 처리 비용을 나타낸다.

제 할당을 수행하기 위한 조건은 에러 편차가 임계값을 초과하는 경우이다.

$$\sigma^2 = \sum_{i=1}^s (Q_i' l'(F_i) - Q_i l(F_i) - Q_i l(F_i))^2 + \sum_{j=1}^k \sum_{i=1}^s (Q_j' l'(f_i) - Q_j l(f_i))^2 Y_{ij} \quad (6)$$

파라미터의 초기 기대값으로 구한 할당 상태를 프로세스 그래프를 구성하는 파라미터에 대하여 재할당 한다. 파라미터의 값이 변할때 마다 최적 할당상태는 변화한다. 그러므로 파라미터의 변화량이 클수록 앞에서 구한 할당상태는 최적 할당이라 할 수 없다. 이에 근거하여 재할당 결정 규칙을, 파라미터의 이전의 값에 대한 편차로 정한다.

에러 편차(error deviation)를 식(6)과 같이 정의할 때 비용 함수와 관련된 파라미터의 변화량이 rms (root mean squared) 값으로 나타낸다.

주 (prime) 기호는 현재 기대값 즉 실행된 연산과 관련된 파라미터 이면 실제값을 나타내고 미실행 연산의 값이면 파라미터의 새로운 기대값이다.

재할당 기준을 정하는데 있어 파라미터의 에러 편차는 재할당 소모비용과 직접적인 상호관련이 없다. 그러므로 실제 응용으로 임계값(threshold value)을 정의한다. 임계값  $\sigma_{th}$ 는 최대 에러 편차 크기에 따라 달라지며

$$\sigma_{th} = \rho \cdot \sigma_{max}, \quad 0 < \rho < 1$$

여기에서  $\sigma_{max}$ 는 기대값과 실제값의 차가 가장 큰 경우의 에러 편차이다.

IV. 최적 정책 결정과 할당 알고리즘

1. 최적 정책 결정

컴퓨터 위치  $n=6$ 이고 연산의 수  $m=10$ 인 프로세스 그래프에 대하여 화일 F는 위치 i에 할당되어 있고 화일의 크기는  $l(F_i)$ 일때 연산의 최적 위치는 표 1과 같다.

비 결정성 모델에서 확률 변수  $F_i, Q_i'$ 와  $Q_i$ 의 평균은  $\bar{F}_i, \bar{Q}_i', \bar{Q}_i$ 이고  $(0, 2 \cdot \text{평균})$ 사이에서 균일 분포를 갖는다고 가정할 때 확률 변수는

$$F_i = 2 \cdot \bar{F}_i \cdot \text{frandom}(\quad) \\ Q_i' = 2 \cdot \bar{Q}_i' \cdot \text{frandom}(\quad) \\ Q_i = 2 \cdot \bar{Q}_i \cdot \text{frandom}(\quad)$$

데이터 감소 함수와 일시적 화일로 구성된 파라미터의 기대값은

$$E[l(f_1)] = \bar{F}_1 R(B_0, \bar{F}_1, C), i=1, \dots, 6 \\ E[l(f_7)] = R(B_3, f_6, C) \quad E[l(f_8)] \\ E[l(f_9)] = R(B_1, \bar{B}, f_6, C) \quad E[l(f_{10})] \\ E[l(f_{10})] = R(B_2, f_1, C) \quad E[l(f_1)]와 같다.$$

표 1. 연산의 최적위치  
Table 1. Optimal sites for operation.

$P_i \backslash B_j$	$B_0$	$B_1$	$B_2$	$B_3$
$P_1$	a	a	b	a
$P_2$	a	a	b	c
$P_3$	a	b	a	d
$P_4$	a	e	a	c
$P_5$	a	d	b	d
$P_6$	a	e	b	c

a=질의 요구를 발생한 기원  
b= $E[l(f_1)], E[l(f_2)]$ 에 의해서 결정  
c= $E[l(f_3)], E[l(f_6)]$ 에 의해서 결정  
d= $E[l(f_4)], E[l(f_5)], E[l(f_6)]$ 에 의해서 결정  
e= $E[l(f_4)], E[l(f_7)]$ 에 의해서 결정  
집합  $\{P_1, P_2, P_3, P_4, P_5, P_6\}$ 에 있는 정책중에서 최선의 정책을 구하려면 각 정책에 대한 통신 비용을 구하여

$$COMM(P') < COMM(P), \quad \forall P \quad (7)$$

식(7)을 만족하는  $P'$ 가 최적 정책이다.

표 1의 연산의 최적 위치는 각각의 연산의 처리비용이 모든 노드에서 동일할때 성립하며 식(7)은 각각의 프로세스 그래프에 대하여 연산의 처리비용은 동일하므로 목적함수를 구성하는 통신비용을 최소화 하는 프로세스 그래프를 선택한다.

2. 할당 알고리즘

최적 정책 결정에서 정해진 정책의 프로세스 그래프에 대하여 할당 알고리즘의 순서를 단계별로 나타내면 다음과 같다.

- [단계 1] 프로세스 그래프의 기저 레벨에서 상위 레벨로 가면서 j 개의 서브그룹으로 분할하고  $k=1$ 로 세트
- [단계 2] 현재의 파라미터 기대값으로 서브프로세스의 할당 상태를 결정한다.
- [단계 3] 단계 2에서 결정된 할당 상태의 위치에서 k서브그룹의 서브프로세스를 수행한다.
- [단계 4] 모든 프로프로세스를 수행하였는지 검색  $k=j$  이면 종료
- [단계 5] 서브그룹 k의 서브프로세스와 관련된 파라미터의 기대값과 실제값 사이의 최대 에러 편차를 구하고 임계값을 결정한다.
- [단계 6] 서브그룹 이하의 서브프로세스 수행 후 발생한 파라미터의 실제값과 기대값의 에러 편차를 구한다.

[단계 7] 단계 5에서 구한 에러편차가 임계값과 비교하여 에러편차 > 임계값이면 단계 2로 아니면 단계 3으로, k를 1증가 시킨다.

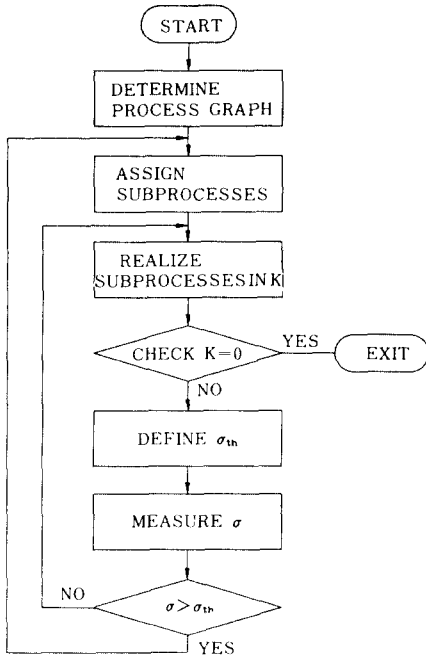


그림 3. 할당 알고리즘의 흐름도  
Fig. 3. Flowchart of relocation algorithm.

V. 예제 및 성능 평가

컴퓨터 위치 n=6이고 질의 요구에 의한 프로세스 나무가 그림 1과 같을때 화일의 수 s=6이고 질의 요구 기원은 위치 4일때 화일과 일시적 화일의 특성은 표 2에 나타냈다.

표 2. 화일과 일시적 화일의 특성  
Table 2. File and temporary file characteristics.

화일 위치	평균크기	$f_i$	$R(B_i, F_i, C)$	$E[1(f_i)]$	
$F_1$	1	$10^5$	$f_1$	0.3	$0.3 \times 10^5$
$F_2$	2	$10^5$	$f_2$	0.28	$0.28 \times 10^5$
$F_3$	3	$10^5$	$f_3$	0.04	$0.4 \times 10^4$
$F_4$	4	$10^5$	$f_4$	0.01	$0.1 \times 10^4$
$F_5$	5	$10^5$	$f_5$	0.15	$0.15 \times 10^5$
$F_6$	6	$10^5$	$f_6$	0.2	$0.2 \times 10^5$

연산의 집합으로 부터 그림 2에 나타낸 6개의 프로세스 그래프를 생성할 수 있다. 최적 정책을 얻기 위해서 단독(unary) 연산은 기저 레벨에 위치해야 한다. 왜냐하면 단독 연산의 출력량은 항상 입력량과 같거나 적기 때문이다. 그러므로 초기연산  $B_0^*$ 는 화일이 위치한 곳에서 수행한다. 정책 1의 프로세스 그래프는 초기연산  $B_0^*, B_1^*$ 로 부터 생성된  $f_1, f_2$ 를 연산  $B_2$ 의 위치에 1혹은 2에 전송하고  $f_3, f_4, f_5, f_6$ 은 위치 4에 전송되어 연산  $B_3, B_1, B_0$ 를 수행한다.

데이터 감소함수  $R(B_i, F_i, C), i=1, \dots, 6$ 은 표 2에 주어졌고  $R(B_3, f_6, C)=0.02, R(B_2, f_1, C)=0.1, R(B_1, f_5, C)=0.05$ 이다. 단위 통신 비용  $C_{ij}=w \cdot D_{ij}=0.2/\text{바이트}$ 이고 단위 처리비용  $Q_i^j=10^{-4}/\text{바이트}, i=1, \dots, 6, Q_j=0.1/\text{바이트}, j=0, \dots, 3$ 이다.

프로세스 그래프 중에서 최선의 정책을 찾기 위해서 각 정책에 대한 통신비용  $C(P)$ 를 계산하면 다음과 같으며 정책 결정시의 파라미터는 기대값으로 나타낸다.

$$\begin{aligned}
 C(P_1) &= C[1(f_1)+1(f_2)+1(f_3)+1(f_5)+1(f_6) \\
 &\quad +R(B_2, f_1, C)1(f_1)] = 14000 \\
 C(P_2) &= C[1(f_2)+1(f_3)+1(f_5)+R(B_3, f_6, C)1(f_6) \\
 &\quad +R(B_2, f_1, C)1(f_1)] = 10800 \\
 C(P_3) &= C[1(f_1)+1(f_2)+1(f_3)+1(f_4)+1(f_5) \\
 &\quad +R(B_1, \bar{B}, f_6, C)1(f_6)] = 15800 \\
 C(P_4) &= C[1(f_1)+1(f_2)+1(f_3)+1(f_4)+1(f_5)+1(f_7) \\
 &\quad +R(B_1, \bar{B}, f_6, C)1(f_6)] = 15800 \\
 C(P_5) &= C[1(f_2)+1(f_3)+1(f_4)+1(f_5)+R(B_1, \bar{B}, f_6, C)1(f_6) \\
 &\quad +R(B_2, f_1, C)1(f_1)] = 10400 \\
 C(P_6) &= C[1(f_2)+1(f_3)+1(f_4)+1(f_5)+1(f_7) \\
 &\quad +R(B_1, \bar{B}, f_6, C)1(f_6)+R(B_2, f_1, C)1(f_1)] \\
 &= 10400
 \end{aligned}$$

연산  $B_1$ 과  $B_2$ 가 동일 위치에서 수행되므로 일시적 화일  $f_7$ 에 대한 통신 비용이 0이 되어 정책 3, 4와 5, 6의 통신 비용이 같게되며 임의로 정책을 결정한다.

$$C(P_6) < C(P_2) < C(P_1) < C(P_4) \text{ 이므로}$$

$P_6$ 를 최적 정책으로 결정하고 식(1)에 의해 비용 합수를 계산하면

$$\begin{aligned}
 Y(P_6) &= \begin{vmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} & X(P_6) &= \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix}
 \end{aligned}$$

$$\begin{aligned} \text{비용 함수 } Z(P_0) &= \sum_{i=1}^6 Q_i l(F_i) \\ &+ Q_0 [ (l(f_3) + l(f_8) + l(f_9)) \\ &+ Q_1 [ (l(f_4) + l(f_7)) \\ &+ Q_2 [ (l(f_1) + l(f_2)) \\ &+ Q_3 [ (l(f_5) + l(f_6)) ] + C(P_0) = 21060 \end{aligned}$$

최적 정책  $P_0$ 에 따라 그림 3의 할당 알고리즘에 의하여 수행되면 그룹 k에서 프로세스 그래프의 실현 기대값은  $E[Z(S_k)]$ 이며  $S_k$ 는 할당 상태를 나타낸다.  $S_k$ 에 대한 실제값은  $Z(S_k)$ 이며 그룹 k에 있는 연산을 수행한다. 연산후의 새로운 정보로 프로세스 그래프를 수행한 새로운 기대값은  $E'[Z(S_k)]$ 이다.

그룹k의 연산 수행후 계산한 에러 편차가 정해진 임계값을 초과하면 재 할당을 수행한다. 표3에 수행 결과를 나타냈으며  $Z(S_k)$ 행의 최상위 원소가 재 할당 하지않은 경우의 실제 비용 함수이며 기저 원소가 재 할당을 행하였을때 실제 비용 함수이다.

우선순위 관계가 존재하는 연산의 집합으로 그림1의 (a)와 같은 프로세스 트리가 주어졌을때 파라미터에 대한 초기 기대값은 표 2에 나타낸 경우에 대하여 초기 파라미터의 기대값으로 프로세스 그래프  $P_0$  선택하고 기저레벨의 연산으로 부터 상위 레벨로 진행하며 각각의 연산을 레벨에 따라 그룹으로 분류한다. 그룹 4에는 연산 4, 5, 6, 7, 8, 9가 그룹 3에는 연산 3, 2가 그룹 2에는 연산1이 그룹1에는 연산 0가 해당된다.

먼저 파라미터의 기대값으로 할당을 수행하면 표 1의 산출 방식에 따라 할당 상태와 전체 프로세스의 실현 기대 비용을 구한다. 할당 상태  $S_0 = [4 \ 2 \ 1 \ 6 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6]$  기대 비용은 21060이다. 연산 4, 5, 6, 7, 8, 9를 각각 노드 1, 2, 3, 4, 5, 6에서 수행하면 파라미터  $Q_i (i=1, 2, \dots, 6)$ 과  $l(f_i) (i=1, 2, \dots, 6)$ 의 실제값을 알 수 있다.

실제값과  $Q_j (j=0, 1, 2, 3)$ 와  $l(f_i) (i=7, 8, 9)$ 의 기대값으로 전체 연산을 실현한 기대 비용은 14044이다. 이때 각 파라미터에 대한 실제값을 미리 안다는 가정하에 계산된 연산 실현 비용이 17128이다.

그룹 4의 연산 수행후 에러 편차의 임계값은 640이며 에러 편차는 4341로 재할당 조건을 만족하므로 재할당을 행하면 할당 상태는  $[4 \ 2 \ 1 \ 5 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6]$  이고 연산 실현 기대 비용은 11875이다. 그림3의 연산 2, 3을 각각 노드 1, 5에서 수행하면 파라미터  $Q_j (j=2, 3)$ 과  $l(f_i) (i=7, 8)$ 의 실제값을 알 수 있다.  $Q_j (j=0, 1)$ 과  $l(f_i) (i=9)$ 의 기대값으로 전체연산을 실

행한 기대값은 15412이다. 위의 할당 상태에서 파라미터의 실제값을 안다는 가정하에 수행한 연산 실현 비용이 14996이다. 위의 과정을 그룹 1까지 반복하면 재할당으로 인한 비용 감소는  $17128 - 14996 = 2132$ 가 된다. 그룹 3에서 그룹 1로 연산을 실행하면서 재할당은 일어나지 않았음을 보인다.

표 3. 예제의 재 할당 결과  
Table 3. Relocation result for the example.

그룹k	$E[Z(S_k)]$	$E'[Z(S_k)]$	$Z(S_k)$	에러편차
4	21060	14044	17128	4341
3	11875	15412	14996	3428
2	15412	15373	14996	620
1	15373	14996	14996	140

임계값=최대 편차의 50%  
최소 비용 함수=14612  
재 할당으로 비용 감소=2132  
재 할당 회수=2

제한한 알고리즘은 발견적 방법으로 예제에 대하여 재 할당 수행 조건인 임계값이 최대 에러 편차의 50%, 60%인 경우에 대하여 50번씩 수행한 결과를 표4(경우1)와 표 5(경우 2)에 나타냈다. 단위 수행당 평균 재 할당회수는 1.66(경우 1)과 1.3(경우 2)이며 모든 경우에 대하여 재 할당으로 비용함수가 감소하는 것은 아니며 증가하는 경우는 발생하지 않았다. 전체적으로 재 할당으로 인한 비용 감소는 11.97%, 11.07% 이고 최적해에 대한 편차는 0.12%, 1.14%로 나타났다.

일반적인 결정성 프로세스에서 프로세스 또는 연산의 최적 할당은 항상 프로세스 수행전에 미리 결

표 4. 재 할당 결과(경우 1)  
Table 4. Relocation result. (case 1)

	재 할당 않은 경우	재 할당 경우	최소치
평균 실현 비용	21401.64	18840.67	18816.88
최소치에 대한 증가율	13.73%	0.12%	0%
초기 할당치에 대한 감소율	0%	11.97%	12.08%

임계값=최대 에러 편차의 50%  
평균 재 할당/수행=1.66  
평균 비용 감소/재 할당=1542.12  
비용 감소 건수=28  
전체 비용 감소=128048.10  
평균 비용 감소/수행=2560.96

표 5. 재 할당 결과  
Table 5. Relocation result. (case 2)

	재 할당 않은 경우	재 할당 경우	최소치
평균 실행비용	21401.64	19032.74	18816.88
최소치에 대한 증가율	13.73%	1.14%	0%
초기 할당치에 대한 감소율	0%	11.07%	12.08%

임계값=최대 에러 편차의 60%  
 평균 재 할당/수행=1.3  
 평균 비용 감소/재 할당=1821.53  
 비용 감소 건수=25  
 전체 비용 감소=118444.75  
 평균 비용 감소/수행=2368.88

정된다. 한번 결정된 할당 상태는 프로세스의 수행이 완료 될때까지 변하지 않는다. 그러나 확률적 모델에 있어 프로세스 그래프의 연산을 처리중에 동적 재할당을 행하는 것은 경제적인 프로세스 실행이 될 수 있다. 본 논문에서는 확률적 모델에서의 연산을 재할당하는 방법을 제안하였고 성능 평가 방법으로 예제에 대하여 제안한 방법으로 프로세스를 실행 하므로써 재할당으로 인하여 프로세스 실행 비용이 감소함을 보였고 최소치에 대한 편차가 적음을 나타냈다.

VI. 결 론

분산 컴퓨터 시스템에서 프로세스 사이의 우선 순위 관계가 존재하는 그래프의 프로세스 할당 문제에 관해 최적 할당 정책을 결정하는 기법을 제시하였고 재 할당 알고리즘을 제안하였다.

제안한 할당 알고리즘은 통신과 처리 비용을 최소로 하는 할당 정책 결정 부분과 프로세스를 수행 하면서 임계값에 따라 재 할당을 행하는 부분으로 구성하였다. 할당 초기에 연산 그룹의 집합을 나타내는 프로세스 그래프를 생성하여 할당 상태의 후보를 제한 하므로써 문제 규모를 축소하고 프로세스의 수행에 따라 파라미터값을 갱신하여 할당 상태를 결정 하므로써 최적해에 근접한 결과를 얻었다.

제안한 알고리즘은 발견적 알고리즘으로 Chu 의 질의 처리 예제에 대하여 임계값을 최대 에러 편차의 50%, 60%인 경우로 나누고 최적 정책 결정및 재 할당 알고리즘을 C 언어로 프로그래밍하여 VAX 11/750에서 각각 50번씩 수행한결과 비용함수가 평균 11.97%, 11.07% 감소하였으며 최적해에 대한 편차가 0.12%, 1.14%로 최적해에 근사한 결과를

얻으므로써 알고리즘의 우수성을 입증하였다. 앞으로 프로세스 그래프에 대한 분석알고리즘과 병렬처리및 부하 균형을 고려한 프로세스 할당에 대한 연구가 필요하다고 생각한다.

參 考 文 獻

- [1] S. Mahmoud and J.S. Riordon, "Optimal allocation of resources in distributed information networks," *ACM Trans. Database Syst.*, vol. 1, pp. 66-78, Mar. 1976.
- [2] 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.
- [3] H.S. Stone, "Multiprocessor Scheduling with Aid of Network Flow Algorithms," *IEEE Trans. Software Eng.*, vol. SE-3, no. 1, pp. 85-93, Jan. 1977.
- [4] G.S. Rao, H.S. Stone and T.C. Hu, "Assignment of tasks in a distributed processor system with limited memory," *IEEE Trans. Computers*, vol. C-28, no. 4, pp. 291-299, Apr. 1979.
- [5] Shyue B. Wu and Ming T. Liu, "Assignment of tasks and resource for distributed processing," *COMPCON Fall 80*, pp. 655-662, Sept. 1980.
- [6] W.W. Chu, "Task allocation in distributed data processing," *IEEE Computer*, vol. 13, no. 11, pp. 57-69 Nov. 1980.
- [7] W.W. Chu and P. Hurley, "Optimal query processing for distributed database systems," *IEEE Trans. Compt.*, vol. C-31, no. 9, pp. 135-150 Sept. 1982.
- [8] C.V. Ramamoorthy and B.W. Wah, "The isomorphism of simple file allocation," *IEEE Trans. Compt.* vol. C-32, no. 3, pp. 221-231, Mar. 1983.
- [9] H. Kasahara and S. Narita, "Practical multiprocessor scheduling algorithms for efficient parallel processing," *IEEE Ans. Compt.* vol. C-33, no. 11, pp. 1823-1829, Nov. 1984.
- [10] G.C. Pathak and D.P. Agrawal, "Task division and multicomputer systems," *The 5th International Conference on Distributed Computing Systems*, ACM pp. 273-280, 1985.



[11] B. Gavish and H. Dirkul, "Computer and data-base location in distributed computer systems," *IEEE Trans. Compt.*, vol. C-35, no. 7, pp. 583-598, July 1986.

[12] 홍진표, 임제탁, "분산 컴퓨터 시스템에서 효율적 화일 할당에 관한 연구," 대한전자공학회 논문지 제26권, 제 9 호, pp. 97-103, 1989.

---

著 者 紹 介

---

洪 進 杓 (正會員) 第26卷 第9號 參照  
현재 금성전선연구소 근무

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