울산대학교 경영학 연구논문집 제1권 제2호 pp 109~118, 1994
Journal of Management Vol 1, No. 2, pp. 109~118, 1994
University of Ulsan

분산처리시스템에서의 업무할당에 관한 연구

이영덕
경영학과

<요 약>

분산처리시스템은 정보통신시스템이나 생산시스템에서 많이 나타나고 있는 시스템으로, 시스템을 설계하고 업무를 할당하는 문제는 매우 중요하며 여러가지 기준에 의해 시스템이 설계되고 있다 본 연구에서는 평균대기길이와 평균대기시간의 기준으로 시스템을 설계하고 업무를 할당하는 것을 연구하였으며, 기존의 해법을 소개하고 이를 수정개선한 해법을 연구한다.

Task Allocation in a Distributed Processors System

Lee, Young-Deuk
Dept. of Management

<Abstract>

This paper considers a design problem for a distributed processors system. For the system, two decision levels are considered: which candidated processors to open with capacity option and how to allocate the task generated at each distributing point. Two performance measures with budget constraint are to minimize the maximum mean queue length and to minimize the maximum mean delay time. For solution, we use the parametric algorithm developed by Tcha and presented a new parametric algorithm.

본 연구는 1993년도 대학연구비 지원에 의한 것임.

I. 서 론

분산처리시스템은 처리시스템이 여러곳에 분산되어 설치되어 있고 각각의 수요지에서 발생한 업무를 이들 처리시스템에 보내어 처리하는 시스템을 말하는 것으로, 정보통신이나 이와 관련된 시스템, 그리고 생산설비가 분산되어 있는 생산시스템 등 산업의 여러 현장에서 볼 수 있는 시스템이다

분산처리시스템의 설계에 관한 연구는 주로 정보통신시스템을 대상으로 많은 학자들에 의해 연구되어졌다 이 시스템을 설계하는 성과 평가의 첫번째 기준은 전체시스템에서의 업무처리 비용과 업무를 처리시스템에 보내는 통신비(수송비)의 총합을 최소화 하는 것으로 Stone[6], Bokhari[1], Sofianopoulou[5] 등의 연구를 대표적으로 꼽을 수 있다. 둘째로 Efe[2]와 Shen과 Tsai[4]는 처리시스템의 업무균일화를 기준으로 분산처리시스템의 설계를 연구하였다 세째로 Tcha와 Lee 또다른 Lee[7]는 처리시스템의 평균대기 길이를 기준으로 분산처리 시스템의 설계를 연구하였다. 이밖에도 여러가지 기준에 의한 연구가 있는데, 이는 Gerla와 Kleinlock[3]의 조사 연구에 잘 제시되고 있다 본 연구에서는 Tcha등[7]의 연구와 같이 평균대기 길이 관점에서 분산처리시스템 설계에 관한 연구를 하였으며, 아울러 평균대기시간 기준에서의 시스템 설계 모형을 제시하고자 한다

여기서 다루는 분산처리시스템을 다음과 같은 상황을 가지고 있다. 여러곳의 후보입지에 처리시스템(예:컴퓨터)을 설치할 수 있으며, 이 처리시스템은 처리능력(capacity)을 기준으로 몇가지 형태가 있다 업무가 발생하는 각각의 수요지에서는 업무처리에 관한 수요가 임의로 발생하는데 이 업무는 처리시스템에 보내져 처리된다 처리시스템을 설치하는데는 설치비용이 발생하며, 업무를 처리시스템에 보내 처리하는데는 처리비용이 발생하는데 이는 수요지와 처리시스템에 따라 다르게 된다 본 연구에서는 어느곳에 어떤 처리 능력을 가진 처리시스템을 설치할 것인가 하는 시스템 설계 문제를 다루게 되는데, 특정 수요지에서 발생한 업무수요를 어느 처리시스템에서 처리하는가의 문제도 동시에 고려되어진다 이러한 시스템의 성과를 평가하는 기준으로 총비용을 최소화 하는 형태의 연구가 많이 있는데, 본 연구에서는 각 업무가 처리시스템에서 처리될때의 평균대기길이와 평균대기시간을 최소화하는 두가지 기준을 성과 평가의 기준으로 삼고자 하며, 비용요소는 총비용 성격을 가진 정해진 예산내에서 시스템을 설계하고자 하는데, 이 예산은 제약조건의 형태로 처리되게 된다 이 문제는 분수 정수 계획법과 정수 계획법의 형태로 모형화 된다.

분수 정수 계획법은 Tcha등[7]이 소개한 모수처리 해법을 이용하면서 이 방법을 수정한 또 다른 해법을 제시 하였으며, 이 두가지 방법의 성과를 비교하였다 모수처리해법에서는 일반적으로 정수해를 구 할 수 없는데, 분단탐색법을 이용하여 최종적으로 정수해를 구하게 된다

II. 모형 설정

1. 분산 처리시스템의 상황

본 연구에서 다루는 시스템은 다음과 같은 상황을 가지고 있다. 처리시스템의 후보지는 M개로 구성되어 있으며 $i \in \{1, 2, \ldots, M\}$의 색인(index)으로 표시된다. 업무가 발생하는 수요지는 N개로 구성되며 $j \in \{1, 2, \ldots, N\}$로 표시된다 이 업무는 크기만 다를 뿐 같은 성격의 업

무로 규정되어 평균 $1/\mu$의 길이를 가지게 되며 j번째 수요지의 업무는 λ_j의 비율을 가진 상호 독립적인 포아송 확률과정에 따라 발생하고, 발생즉시 처리시스템으로 보내져 처리된다. 처리시스템 i로 보내진 업무들은 대기장소(buffer)에서 대기되다 선입선출(FCFS) 원칙에 의해 처리된다. 처리시스템은 단위시간당 처리능력에 따라 구분되어지는데 색인 p로 표시된다. 이러한 상황에서 분산처리시스템을 두가지 평가기준에 따라 설계 될 수 있는데 첫째는 평균대기 길이를 줄일 수 있도록 시스템을 설계하는 경우이고, 둘째는 평균대기 시간을 줄이는 기준에 따른 설계이다.

2. 수리적 모형

1) 평균대기 길이 기준의 모형

각각의 수요지에서 발생한 업무는 처리시스템으로 보내어져 처리되는데, 처리시스템에 이미 수행중인 업무가 있을 때는 대기장소(buffer)에서 기다려지다 처리가 된다 이때 이 대기장소는 제한된 능력(capacity)를 갖게 되므로 평균대기 길이를 줄이는 것이 분산처리시스템 설계의 기준이 될 수 있다. 여기에서는 다음과 같이 각 처리시스템 중 평균대기 길이의 최대값을 최소화 하는 방향의 수리적 모형을 설정한다.

(P1)

$$\text{Min } Z = \max_{i\in I} \left\{ \frac{\sum_{j\in J} \lambda_j \chi_{ij}}{\sum_{p\in P(i)} \mu C_{ip} y_{ip} - \sum_{j\in J} \lambda_j \chi_{ij}} \right\} \tag{1}$$

s. t

$$\sum_{j\in J} \lambda_j \chi_{ij} \le \sum_{p\in P(i)} \mu C_{ip} y_{ip} \qquad \forall\ i\in I, \tag{2}$$

$$\sum_{j\in J} \chi_{ij} = 1, \qquad \forall\ j\in J, \tag{3}$$

$$\sum_{p\in P(i)} y_{ip} \le 1, \qquad \forall\ i\in I, \tag{4}$$

$$\sum_{i\in I} \sum_{j\in J} v_{ij} \chi_{ij} + \sum_{i\in I} \sum_{j\in J} F_{ip} y_{ip} \le b, \tag{5}$$

$$\chi_{ij} \ge 0, \qquad \forall\ i\in I,\ j\in J, \tag{6}$$

$$y_{ip} \in \{0,1\}, \qquad \forall\ i\in I,\ p\in P(i), \tag{7}$$

여기에서

$I = \{1,2,..,M\}$: 처리시스템 후보지의 집합,
$J = \{1,2,...,N\}$: 업무가 발생하는 수요지의 집합,

$P(i)$: i번째 처리시스템 후보지에 설치 가능한 시설 용량의 집합,

x_{ij} : 수요지 j에서 발생한 업무를 처리시스템 i로 보내진 비율,

y_{ip} : 0-1 변수로서 i번째 처리시스템이 $p(p \in P(i))$형의 시설용량으로 설치되면 1의 값, 아니면 0의 값을 갖는다

(1)식은 목적함수로 최대평균대기길이를 최소화 한다. (2)식은 각 처리시스템에 보내진 업무량의 합이 처리시스템의 처리능력을 초과 할 수 없음을 나타내는 식이다 (3)식은 모든 수요지의 업무는 전부 처리 되어야 함을 나타낸다 (4)식과 (7)식은 처리시스템이 i 후보지에 설치된다면 여러가지 시설 용량 중 하나만 선택되어야 함을 나타낸다. (5)식은 시스템을 설치하는데 드는 총비용을 예산 b 이내에서 사용되어야 함을 나타내는데, 비용 v_{ij}는 j번째 수요지에서 발생한 업무를 i번째 처리시스템으로 보내어 처리하는데 드는 비용을 나타내며, 비용 F_{ip}는 i번째 처리시스템 후보지에 시설용량 p형태의 처리시스템을 설치하는데 드는 비용을 나타낸다. (6)식은 수요지에서 발생한 업무는 나뉘어져서 각 처리시스템에서 처리 될 수 있음을 나타내는데, 업무가 하나의 처리시스템에서 처리되어야 하는 경우에는 이 식은 $x_{ij} \in \{0,1\}$의 형태로 바뀌게 된다.

2) 평균대기시간 기준의 모형

각 수요지에서 발생한 업무가 가능한 한 빨리 처리 되어야 한다는 기준에서 보면 평균대기시간을 줄 일 수 있도록 하는 것도 시스템을 설계하는 또 다른 기준이 될 수 있다 이러한 기준의 시스템을 설계하는 수리적 모형은 모형 P1 에서 목적 함수식 (1)만 바뀌어진 형태로 다음과 같은 모형을 갖게 된다

$$(\text{P2}) \quad \text{Min } Z' = \max_{i \in I} \left\{ \frac{1}{\sum_{p \in P(i)} \mu C_{ip} y_{ip} - \sum_{j \in J} \lambda_j x_{ij}} \right\} \qquad (1)'$$

s t (2), (3), (4), (5), (6) 그리고 (7)

여기에서 목적함수식 (1)′는 다음과 같이 바뀌어 질 수 있다.

$$\text{Min } Z' = \min_{i \in I} \left\{ \sum_{p \in P(i)} \mu C_{ip} y_{ip} - \sum_{j \in J} \lambda_j x_{ij} \right\} \qquad (1)''$$

따라서 이 문제는 단순한 혼합정수계획모형으로 나타낼 수 있다.

III. 해법 연구

본 연구에서는 분산처리시스템을 설계하는 모형으로 평균대기길이 기준의 모형과 평균대기시간 기준의 모형을 제시하였다 평균대기시간 기준의 모형은 단순한 혼합정수계획모형으로 표시되어 일반적인 정수계획법의 해법으로 풀이가 가능하다. 그러나 평균대기길이 기준의 모

형은 분수정수계획모형이므로 보다 복잡한 형태를 지니게 된다. 이러한 P1과 같은 문제의 해법은 Tcha등[7]의 연구에서 제시가 되었는데, 본 연구에서는 또다른 형태의 해법을 제시하고 Tcha등[7]의 해법과 비교를 하고자 한다.

1. 모수적 해법과정

P1은 0-1 분수정수계획법의 모형으로 이 모형을 풀기 위해 새로운 변수 u를 도입하여 다음과 같은 모형을 만든다.

(SP1) Min Z = u

$$\text{s.t} \quad \frac{\sum_{j\in J} \lambda_j x_{ij}}{\sum_{p\in P(i)} \mu C_{ip} y_{ip} - \sum_{j\in J} \lambda_j x_{ij}} \le u, \ \forall i \in I \qquad (8)$$

(2), (3), (4), (5), (6) 그리고 (7).

이때 (8)식과 (2)식은 합쳐져서 다음과 같은 식으로 바뀔 수 있다

$$(1 + u) \sum_{j\in J} \lambda_j x_{ij} - u \sum_{p\in P(i)} \mu C_{ip} y_{ip} \le 0. \quad \forall i \in I. \qquad (9)$$

따라서 SP1은 다음과 같이 바뀌어진다.

(SP2) Min Z = u
s.t (9), (3), (4), (5), (6) 그리고 (7)

SP2를 풀기 위해서는 다음과 같은 개념이 적용된다. u가 확정된 상태에서의 SP2의 실현가능 영역을 S(u)라하면, S(u)는 u가 점차 축소됨에 따라 그 영역도 같이 축소된다. 즉 $u^{k+1} \le u^k$이면 $S(u^{k+1}) \le S(u^k)$가 된다. 따라서 SP2의 최적해 u^*를 찾는 것은 $S(u^k)$를 점차 축소하여 $S(u^k)$가 공집합이기 직전의 u^k값을 찾는 것이 된다 이러한 개념에 의해 오차의 한계를 가진 최적해 u^*는 다음과 같이 구해진다.

$$u^* = \min \{ u \mid S(u) \ne \emptyset \}$$
$$\fallingdotseq \{ u \mid S(u) \ne \emptyset, \ S(u - \varepsilon) = \emptyset \}$$

SP2를 푸는 해법은 다음의 두 단계로 나뉘어 진다. 첫번째 단계는 주어진 u값을 가지고 실현가능해 (x,y)의 존재여부를 확인하는 것이고 두번째 단계는 다음번 u값을 구하는 것이다. P((x,y)|u)를 주어진 u값을 가지고 (x,y)를 찾는 문제라 하고, P(u|(x,y))를 다음번 u값을 구하는 문제라 할 때 해법의 절차는 다음과 같다.

해법절차 SP

0단계 t = 1로 놓는다 u의 값을 $P((x,y)|u^t)$가 실현가능하도록 충분히 큰 값을 준다
1단계 . $P((x,y)|u^t)$를 푼다 만약 실현 불가능이면 과정을 끝낸다.
아니면 실현 가능해를 $(x,y)^t$라 한다
2단계 . $P((x,y)|u^t)$를 풀고 여기서 얻은 해를 $u^t(x,y)$라 한다
$u^{t+1} = u^t(x,y) - \varepsilon$로 하고 $t = t + 1$로 한 뒤 1 단계로 돌아간다.

최초로 u^t값을 $P((x,y)|u^t)$이 실현가능하도록 충분히 큰 값을 보장하기 위해서는 식(9) 대신에 식(2)를 사용하는 것도 하나의 내안이 될 수 있다

$P((x,y)|u^t)$는 인위변수 w^t를 도입하여 다음 모형과 같이 설정된다

$P((x,y)|u^t)$ Max $Z = w^t$

$$\text{s.t } (1+u^t)\sum_{j\in J}\lambda_i x_{ij} - u^t \sum_{p\in P(i)} \mu C_{ip} y_{ip} + w^t \le 0, \quad \forall i\in I,$$

$$\sum_{i\in I} x_{ij} = 1, \quad \forall j\in J,$$

$$\sum_{p\in P(i)} y_{ip} \le 1, \quad \forall i\in I,$$

$$\sum_{i\in I}\sum_{j\in J} V_{ij}x_{ij} + \sum_{i\in I}\sum_{p\in P(i)} F_{ip}\, y_{ip} \le b,$$

$$w^t \ge 0, \text{ (6) 그리고 (7)}$$

$P(u|(x,y)^t)$로는 Tcha등[7]은 다음과 같이 간단한 방법으로 $u^t(x,y)$를 구하였다

$$u^t(x,y) = \max_i \left\{ \frac{\sum_{i\in j}\lambda_i x_{ij}^t}{\sum_{p\in P(i)}\mu C_{ip} y_{ip}^t - \sum_{i\in j}\lambda_i x_{ij}^t} \;\middle|\; \sum_{p\in P(i)}\mu C_{ip} y_{ip}^t > 0 \right\}, \quad (10)$$

본 연구에서는 $P(u|(x,y)^t)$를 다른 모형으로 구하는데 이는 다음과 같다 우선 다음의 세 가지 목적함수는 같은 최적해 $(x,y)^*$를 얻음을 쉽게 알 수 있다.

$$\text{(i) Min } Z = \max_{i\in I} \left\{ \frac{\sum_{i\in j}\lambda_i x_{ij}}{\sum_{p\in P(i)}\mu C_{ip} y_{ip} - \sum_{i\in j}\lambda_i x_{ij}} \right\}$$

$$\text{(ii) Min } Z = \max_{i \in I} \left\{ \frac{(\sum_{i \in j} \lambda_i x_{ij}) / (\sum_{p \in P(i)} \mu C_{ip} y_{ip})}{1 - (\sum_{i \in j} \lambda_i x_{ij}) / (\sum_{p \in P(i)} \mu C_{ip} y_{ip})} \right\}$$

$$\text{(iii) Min } Z = \max_{i \in I} \left\{ \frac{\sum_{i \in j} \lambda_i x_{ij}}{\sum_{p \in P(i)} \mu C_{ip} y_{ip}} \right\}$$

주어진 y^t값을 가지고 (iii)의 목적함수를 이용하여 다음의 선형계획문제를 유도한다.

$$(\ LP(y^t)\ )\quad \text{Min } Z = v^t$$

$$\text{s.t}\quad \frac{\sum_{j \in J} \lambda_j x_{ij}}{\sum_{p \in P(i)} \mu C_{ip} y_{ipt}} - v^t \le 0 \qquad \forall\ i \in I,$$

$$\sum_{j \in J} x_{ij} = 1, \qquad \forall\ j \in J,$$

$$\sum_{i \in I} \sum_{j \in J} v_{ij} x_{ij} \le b - \underset{i \in I}{S}\ \underset{j \in J}{S}\ F_{ip}\ y_{ip}^t,$$

$$\chi_{ij} \ge 0, \qquad \forall\ i \in I,\ j \in J,$$

위 모형에서 y_{ip}^t 값은 주어져 있으므로 위의 문제는 선형계획문제가 된다. v^*를 위에서 구한 최적해라 할때 $u^t(x,y)$값을 다음과 같이 구할 수 있다.

$$u^t(x,y) = \frac{v^*}{1 - v^*} \qquad (11)$$

위와 같이 (11)식에 의해 $u^t(x,y)$값을 구하는 것은 (10)식에 의해 구하는 것과 비교하여 다음과 같은 장단점이 있다. 우선, 장점으로는 (11)식에 의해 구한 u^t값이 (10)식에 의해 구한 값보다 작거나 같게 되므로 u^t를 구하는 반복연산(iteration)수를 줄일 수 있다. 단점으로는 (10)식은 아주 간단히 구할 수 있으나 (11)식은 선형계획문제를 하나 묶어야 하므로 시간이 소요되게 된다.

$u^t(x,y)$값이 구해지면 u^{t+1}값을 다음과 같이 구한다.

$$u^{t+1} = u^t(x,y) - \varepsilon$$

이 때 $u^{t+1} < u^t(x,y) \le u^t$의 관계가 성립되며 해법과정 SP를 통하여 최적해 u^*를 구할 수 있다.

2. 분단탐색과정

위와 같은 내용을 통하여 해법과정 SP로 문제를 푸는데, 대부분의 경우 곧바로 P1의 혼합 정수해를 구할 수 없다. 이 때는 분단탐색법(branch and bound method)을 이용하여 혼합정수해를 구하게 되는데 분단탐색마디에서 후입선출법(last-in first-out)에 의해 처리를 한다 물론 이 때는 y_{ip}의 값이 정수해가 나와야 하므로 y_{ip}의 기준으로 분단탐색과정이 진행된다

IV. 결 론

분산처리 시스템의 설계는 여러각도에서 연구가 되는데, 본 연구에서는 처리시스템에서의 대기행렬길이 기준의 모형을 인용 소개하였으며, 대기행렬시간 기준의 모형을 제시하였다 대기행렬시간 기준의 모형은 단순한 혼합정수계획모형을 가지고 있으므로 일반적인 해법으로 풀 수가 있으나, 대기행렬길이 기준의 모형은 복잡한 과정을 거쳐 풀이가 가능한데, 기존의 해법을 소개함과 동시에 이를 수정한 새로운 해법을 연구하였다.

해법과정은 FORTRAN-IV로 코딩하고 PC-486/50을 이용하여 돌려보았다 이 과정에서 u(x,y)를 구하는 방법을 두가지로 하였는데 (10)식을 이용한 방법을 Tcha로 하고 (11)식을 이용한 방법을 Lee로 이름하여 구분하였다. 예제로 15문제를 다루어 비교하였는데 이는 표1에 나타나 있다 Tcha 방법과 다른 분수계획법과의 직접비교 혹은 간접비교는 Tcha등[7]에서 이미 실시하였으므로 여기에서는 Tcha와 Lee만을 비교하였다

표1에서 첫번째 열은 문제의 크기를 나타내며 세번째 열은 분단탐색마디수를 나타내는데 Tcha와 Lee가 비슷함을 알 수 있다 네번째는 분단탐색과정 동안 SP과정을 몇번 거쳤는가를 나타내는 표로 Lee가 Tcha보다 SP과정을 적게 수행하는 것을 알 수 있다. 마지막열을 CPU 계산시간을 나타내는데 Tcha와 Lee가 비슷함을 알 수 있다 이는 Lee가 Tcha가 보다 SP과정을 적게 수행하지만 u'(x,y)를 구하는데 시간이 더 소요되기 때문에 오는 결과라 할 수 있다

표1에서 나타난 예제문제들은 Tcha와 Lee가 비슷한 소요시간을 가진다 볼 수 있다 그러나 이 비교들은 예제문제들의 형태에 따라 달라 질 수 있다. 예를 들면 SP과정을 수행하는데 시간이 좀더 많이 걸리고 상대적으로 (11)식에 의하여 u'(x,y)를 구하는데 소요되는 시간이 적을 때에는 Lee의 계산시간이 적게 될 것이며, 반대의 경우도 성립될 수 있다 따라서 Tcha와 Lee는 경우에 따라 보완적으로 사용될 수 있다.

이상과 같이 대기행렬의 길이와 시간의 관점에서 분산처리시스템의 설계에 관한 모형을 연구하고 구 해법을 개발하였는데 분산처리시스템은 상황에 따라 여러가지 기준이 생길 수 있고 이에 관한 연구도 지속적으로 수행되어야 할 것이다

표 1 계산결과

문제크기[a] $\lvert I\rvert \times \lvert J\rvert \times \lvert P\rvert$	문제번호	분단탐색마디수		분단탐색과정의 반복연산 수 (SP의 수)		계산시간(초)[b]	
		Tcha	Lee	Tcha	Lee	Tcha	Lee
5×8×2	1	5	5	15	12	0.94	0 92
	2	15	14	38	30	1.87	1.74
	3	35	34	77	61	3.18	3.21
	4	51	52	126	101	4.37	4.41
	5	95	94	167	130	5 21	5.19
5×8×3	6	63	61	137	110	4.36	4.21
	7	61	59	159	127	5.03	4.98
	8	57	60	146	115	5.97	6.06
	9	135	133	274	219	7.88	7.74
	10	149	150	305	232	11.36	11.61
5×10×4	11	137	140	284	226	13.76	14.03
	12	207	205	511	401	20.64	19.87
	13	219	210	627	511	28 28	29.14
	14	275	260	738	573	28.85	28.61
	15	331	337	979	782	43.69	43.34

a P(i)는 모든 i에 대하여 같음.

b 계산시간은 CPU시간을 의미하며 I/O시간 제외.

참고문헌

1 S. H Bokhari, "A shortest tree algorithm for optimal assignments across space and time in a distributed processor system" IEEE Trans. on Soft Eng. Se-7, (1981) 583-589.
2 K Efe, "Heuristic models of task assignment scheduling in distributed systems" IEEE Trans. on Computers, (1982) 50-56
3. M Gerla and L. Kleinrock, "On the topological design of distributed computer networks", IEEE Trans. on Commun. COM-25 (1977) 48-60.
4 C. C. Shen and W H. Tsai, "A graph matching approach to optimal task assignment in distributed problem-solver" IEEE Trans. on Computers C-34, (1985) 197-203
5. S. Sofianopoulou, "The process allocation problem A survey of the application of graph-theoretic and integer programming approaches" J Opl Res. Soc 43, (1992) 407-413.
6 H. Stone, "Multiprocessor scheduling with the aid of network flow algorithm" IEEE Trans. on Soft. Eng SE-3 1, (1977) 87-93.
7 D W. Tcha, B I Lee and Y. D. Lee, "Processor selection and traffic spilitting in a parallel processors system" Acta Informatica 28, (1992) 415-423.