

유전자 알고리즘을 이용한 최적의 분산 데이터베이스 시스템 설계

고 석 범[†] · 윤 성 대^{††}

요 약

최근에 정보네트워크 사용자의 급증에 따라 DDS(Distributed Database System)는 VAN(Value Added Network) 상에서 구현되었다. DDS는 지역적으로 분산된 작업환경에서 중앙집중식 데이터베이스 구축보다 여러 측면에서 장점이 있으나 불합리한 설계는 컴퓨터 및 네트워크 자원의 비효율적 사용에 의한 비용의 증가와 데이터 유지를 위한 복잡도의 증가를 야기한다. DDS 설계시 각 사이트에서 적절한 컴퓨터를 선택하는 문제와 단편화된 데이터를 적절한 사이트에 할당하는 문제가 중요하다. VAN 상에서 컴퓨터 선택과 데이터 파일의 할당은 응답대기시간(waited response time)과 투자비용(investment cost)의 상관관계를 반드시 고려하여 결정되어야 하므로, 본 논문에서는 각 컴퓨터와 파일의 할당의 영향에 따라 두 목적함수의 상관관계를 고려한다. 특히, 응답대기 시간에 대한 보다 실제적인 평가를 위해 M/M/1 큐잉 시스템을 기초로 하여 설계한다. 제안된 설계모델은 경험적 탐색법 중의 하나인 유전자 알고리즘(Genetic Algorithm)의 적용을 통해 효율적인 해의 탐색을 시도하고 제안된 수학적 모델과 알고리즘의 성능 검토를 위해 모의실험 및 결과분석을 한다.

The Optimal Distributed Database System Design Using the Genetic Algorithm

Sucbum KO[†] · Sungdae YOUN^{††}

ABSTRACT

Recently, DDSs(Distributed Database Systems) have been implemented on VAN(Value Added Network) as we know the amazing expansion of information network. DDS can yield significant cost and response time advantages over centralized systems for geographically distributed organizations. However, inappropriate design can result in high cost and poor response time to maintain the database at each site. In a DDS design, the main problem is how to select proper computer and how to allocate data fragment into a proper site. In this paper, we address DDS design problem of selecting the proper class of computers and the allocating data files on VAN. Also, the formulated model includes two objectives, the waited response time and the investment cost to include their relationship. Specially, the formulation of waited response time is based on M/M/1 queueing system to evaluate more precisely. GA(Genetic Algorithm), a kind of heuristic search method, is developed to search an optimal solution in the proposed design model and we show the simulation result to examine the algorithm performance.

1. 서 론

최근에 인터넷의 광역적인 연결과 기업 업무의 지역적 분산에 따라 DDS(Distributed Database System)

는 인터넷과 같은 VAN(Value Added Network)상에서 구현된다. DDS 설계에서 주요한 문제점은 각 사이트에서 적절한 컴퓨터를 선택하는 문제와 단편화된 데이터를 적절한 사이트에 할당하는 문제로 축약된다. DDS에서 적절한 컴퓨터와 단편화된 데이터의 할당은 네트워크 사용을 최소화함으로써 보다 빠른 질의의 처리가 가능함과 동시에 특정 사이트의 고장에 대해서도 시스템 전체에 영향을 미치지 않으므로써 신뢰성이 높

* 본 논문은 1999년도 부경대학교 학술연구조성비 지원에 의하여 연구되었음

† 준 회원 : 부경대학교 대학원 전자계산학과

†† 정 회원 : 부경대학교 전자계산학과 교수

논문접수 : 1999년 12월 23일, 심사완료 : 2000년 9월 1일

다. 그러나 불합리한 설계는 컴퓨터 및 네트워크 자원의 비효율적 사용에 의한 비용의 증가와 데이터 유지를 위한 복잡도의 증가를 야기한다. 그러므로 DDS 설계문제는 로컬 데이터베이스 설계문제(local database design), 네트워크 위상설계문제(network topology design), 자원할당문제(resource allocation problem)를 동시에 고려해야 하는 복잡한 문제이다[2, 3, 11].

Mitrani와 Sevic[8]은 프로세서들의 최적 속도를 선택하는 문제만을 고려하였고 Mahmound[10]와 Chu[9]는 최적 파일할당 문제로서 시스템 운용비용을 최소화 하는 방법만을 제안하여 파일할당 문제와 통신 네트워크의 위상설계 문제를 독립적으로 다루었다. Dutta와 Jain[12]은 프로세서 선택, 파일할당, 그리고 네트워크 위상설계를 동시에 고려한 조합적인 문제(combinatorial problem)에 대해 다루었으나, 제안된 모델에서의 목적함수들간의 상관관계에 대해 독립적으로 설계하였고 실세계 문제에서 고려해야 할 주요한 요소들이 누락되었다. March와 Rho[1]는 파일할당 문제뿐만 아니라 사용자 질의(query)를 서브질의로 분해해서 각 사이트로 적당하게 할당하여 분산수행 시키는 질의할당 문제를 다루었다. 따라서 기존의 연구들에서는 동급의 컴퓨터만을 고려함으로써 이질 기종 상에서의 트랜잭션 처리 및 컴퓨터 선택에 관해서는 언급되지 않았고 네트워크 운용비용과 질의 수행시간의 상관관계를 동시에 고려하지 않았다.

본 논문에서는 최적화 문제로서 상호 밀접한 연계를 가지는 응답대기시간(Waited Response Time : WRT)과 투자비용(Investment Cost : IC)을 고려하였고 컴퓨터의 이질 기종 상에서의 분류된 트랜잭션 처리에 대한 영향을 고려하였다. 응답대기시간은 사용자로부터 질의가 발생된 후 그 결과를 사용자에게 제공하기까지의 시간으로 정의되고, 투자비용은 각 사이트에서의 컴퓨터 시스템을 설치하는 비용으로 고려하였다. DDS 설계 문제는 매우 어려운 작업(hard combinatorial problem)으로서 네트워크 문제와 파일의 할당(컴퓨터 선택)문제와의 밀접한 상관관계를 가짐으로 반드시 동시에 고려되어야 한다. 따라서 본 연구에서는 VAN상에서의 단편화된 데이터의 할당문제와 네트워크 문제를 동시에 고려하여 IC 및 WRT를 최소화하면서 각 사이트에 적합한 컴퓨터를 선택하고 파일 할당을 하는 효율적인 분산데이터베이스를 설계한다. 특히, WRT를 보다 정확하게 평가하기 위해 큐잉 시스템(queueing

system)을 도입하여 통신지연(communication delay), 디스크 입출력 지연(disk I/O delay), CPU 지연(CPU delay)으로 세분화하여 고려하였다.

본 논문에서는 제안된 모델에서 효율적인 해를 탐색하기 위해 경험적 탐색법(heuristic search method) 중의 하나로써 통계적 탐색(statistic search)[6]을 하는 유전자 알고리즘(Genetic Algorithm : GA)의 적용방법을 제안한다. 제안된 수학적 모델을 풀기 위해 GA를 사용하는 주요한 이유는 다음 두 가지 장점으로 요약할 수 있다. 첫째는 제안된 모델과 같이 두 개의 목적함수를 동시에 고려해야 하는 다목적함수의 처리와 해의 변화에 매우 민감한 어려운 문제를 푸는데 있어서 GA는 여러 가지 장점을 가진다. 둘째는 경험적 탐색법의 특성으로서 GA는 결정자가 어떤 해를 요구할 때마다 언제든지 해를 제공할 수 있다. 즉, GA는 수행시간이 길수록 좀 더 최소해에 가까운 해를 제공하고 일정시간이 지나면 최적해(optimal solution)를 제공한다.

본 논문의 구성은 2장에서는 VAN상에서의 DDS 구축을 위해 요구되는 각 사이트로의 데이터의 분산(data distribution)과 적당한 컴퓨터 계층을 선택하기 위한 설계모델을 제안하고, 3장에서는 설계모델에서 효율적인 해들을 탐색하기 위한 GA의 설계에 대해서 논한다. 그리고 4장에서는 GA수행을 통한 모의실험과 결과를 분석하고, 5장에서는 결론과 향후과제를 논의한다.

2. 분산데이터베이스 시스템 설계 모델

WRT와 IC를 최소화하면서 각 사이트에 대해 최적의 컴퓨터 선택 문제와 데이터 분산을 고려한 DDS를 설계하기 위해 WRT는 각 사이트에서의 통신지연시간, disk I/O 지연, CPU 지연으로 유도된다. 그리고 IC는 각 사이트에서 소요되는 컴퓨터 시스템의 설치비용으로서 CPU, 주기억장치, 보조기억장치로 구성된다. VAN 자체의 비용부담 특성에 따라 각 사이트를 연결하는 네트워크 구축비용은 제외된다.

설계모델을 기술하기 위해 다음의 집합들을 정의한다.

S : 네트워크 상의 모든 사이트들의 집합

K : 데이터 파일들의 집합

M : 컴퓨터 계층들의 집합

C : 작업 계층들의 집합으로 수행 가능한 컴퓨터의 최하위그룹

각 컴퓨터는 계층 1(개인용 컴퓨터(PC)), 계층 2(워크스테이션(WS)), 계층 3(메인프레임(MF)), 계층 4(대용량 메인프레임(LMF)), 그리고 계층 5(슈퍼컴퓨터(SC))로 분류한다. 또한, 작업들은 컴퓨터의 성능에 따라 분류되는데 예를 들어, 작업 계층 3은 계층 3 이상의 컴퓨터에서만 수행될 수 있음을 의미한다. <표 1>은 각 컴퓨터 계층에 따른 작업 계층의 실행 가능여부를 나타낸다.

수학적 모델을 정의하기 위한 인자들을 아래와 같이 정의한다.

- v_k : 파일 k 의 크기
- b_{sc} : 사이트 s 에 도착한 작업이 계층 c 일 확률
- q_{ck} : 작업 계층 c 가 파일 k 를 갱신할 확률
- d_{ck} : 작업 계층 c 가 질의 파일 k 일 확률
- o_{ck} : 작업 계층 c 에 의해서 파일 k 를 갱신하기 위하여 요구되는 데이터 전송의 평균길이
- t_{ck} : 작업 계층 c 에 의해 파일 k 로부터 질의를 만족하기 위해 요구되어지는 데이터 전송의 평균길이
- e_c : 작업 계층 c 에 대해 요구되는 평균기본메모리

<표 1> 컴퓨터 계층과 작업 계층의 분류

(계층번호) 컴퓨터 기종	작업 1	작업 2	작업 3	작업 4	작업 5
(1) 개인용컴퓨터(PC)	실행 가능				
(2) 워크스테이션(WS)	실행 가능	실행 가능			
(3) 메인프레임(MF)	실행 가능	실행 가능	실행 가능		
(4) 대용량 메인프레임(LMF)	실행 가능	실행 가능	실행 가능	실행 가능	
(5) 슈퍼 컴퓨터(SC)	실행 가능	실행 가능	실행 가능	실행 가능	실행 가능

제안된 모델에는 두 개의 결정 변수(decision variable)들이 존재한다. x_{ms} 는 각 사이트에 대해 분류된 컴퓨터들의 위치를 의미하고, y_{sk} 는 각 사이트에 대한 데이터 파일의 위치를 의미한다. 각 data는 수직적으로 관련 있는 필드들로 조합되어 미리 단편화(fragmentation)되고 단편화 된 각 데이터 파일의 중복 할당이 허용된다.

제안된 모델에서 각각의 결정 변수들은 아래와 같다.

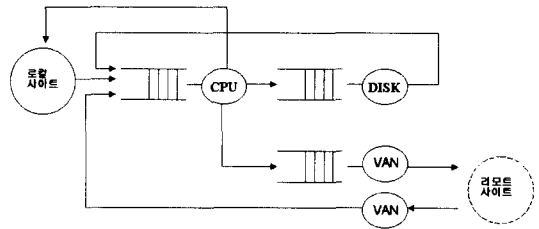
$$x_{ms} = \begin{cases} 1; & \text{if 컴퓨터계층 } m \text{이 사이트 } s \text{에 존재} \\ 0; & \text{otherwise} \end{cases}$$

$$y_{sk} = \begin{cases} 1; & \text{if } k \text{번째 데이터 파일이 사이트 } s \text{에 존재} \\ 0; & \text{otherwise} \end{cases}$$

2.1 목적 함수(objective functions)

2.1.1 응답 대기 시간

본 논문에서 고려하는 DDS 설계모델에서의 WRT는 각 사이트에서의 통신 네트워크의 사용에 의한 지연시간(communication delay), 디스크 I/O 사용에 의한 지연시간(disk I/O delay), 그리고 CPU 사용에 의한 지연시간(CPU delay)으로 유도된다. 이들 각각의 요소에 따른 지연시간에 대한 전체적인 구조는 그림1과 같다.



(그림 1) WRT에 대한 전체 구조

(그림 1)과 같이 설계모델에서의 WRT는 M/M/1 큐잉모델을 기반으로 하고 CPU와 disk I/O의 지연시간은 체제시간으로 유도되지만, 다른 사이트와 관련된 VAN을 통한 통신지연은 평균대기시간으로 유도된다.

(1) 통신지연시간

통신 지연시간은 현재 사이트에서 다른 사이트의 처리가 끝날 때까지 대기해야 하므로 평균대기 시간(Average Waiting Time : AWT)과 VAN상에서의 데이터를 송수신하기 위한 평균 전송시간(Average Transmission Time : ATT)의 합으로 유도된다.

각 사이트에서 발생하는 작업들은 네트워크를 사용하지 않고 로컬 사이트에서 처리되어질 수도 있고, 네트워크를 사용하여 원격 사이트에서 처리 될 수도 있다. 원격 사이트에서 처리되는 작업은 두 가지 경우를 생각할 수 있다. 첫째는 발생한 작업 계층이 현 사이트의 컴퓨터 계층 보다 큰 경우($c > m$) 즉, 컴퓨터의 성능이 발생한 작업을 처리 할 수 없는 경우로서 발생한 작업을 처리하기 위해 원격 사이트로 필요한 데이터 파일을 전송하고 처리하게 된다. 둘째는 발생한 질

의 처리에서 현재 사이트에 존재하지 않는 데이터 파일을 요구할 때 원격 사이트에서 데이터를 참조하고 처리하게 된다.

통신지연시간에 대한 수식을 유도하기 위한 인자들을 아래와 같이 정의한다.

UL_{tp} : 사이트 t 와 p 간의 통신링크용량
(bytes/unit time)

f_c : 단위시간당 작업 계층 c 의 수행 빈도수

H_{ctp} : 작업 계층 c 에 의한 사이트 t 와 p 간의 통신량

TL_{tp} : 사이트 t 와 p 간의 통신 링크상의 총 통신량

$$TL_{tp} = \sum_c f_c \cdot H_{ctp}$$

각 사이트에서 발생하는 작업은 크게 특정 정보를 검색하기 위한 질의처리(query processing)와 갱신된 데이터 파일들의 내용을 일치시키기 위한 갱신(update)으로 분류할 수 있다. 그러므로 질의에 대한 사이트 s 와 p 간의 통신 링크상의 평균 메시지 길이 (W_{sp}^R)와 갱신에 대한 평균 메시지길이(W_{sp}^U)는 다음과 같이 각각 유도된다.

$$W_{sp}^R = \frac{TL_{sp}}{\sum_k f_c \cdot N_{cs}} \cdot t_{ck} \cdot p_{ck}(1 - y_{sk}) \text{ for all } c \text{ and } k$$

$$W_{sp}^U = \max \left\{ \frac{TL_{sp}}{\sum_c f_c H_{csp}} o_{ck} q_{ck} y_{sk} \right\} \text{ for all } c \text{ and } k$$

여기에서 t_{ck} 는 작업 계층 c 가 파일 k 에 대해 질의 만족을 위해 요구되는 데이터 전송의 평균길이를 나타내고, p_{ck} 는 작업 계층 c 가 파일 k 를 참조할 확률이다. 그리고 o_{ck} 는 작업 계층 c 에 의한 파일 k 를 갱신하기 위해 요구되는 데이터 전송의 평균길이이고, q_{ck} 는 작업 c 가 파일 k 를 갱신할 확률을 나타낸다.

Little의 법칙[17]의 대기시간을 적용하기 위해 질의에 대해 발생하는 작업의 입력량 λ^R 과 통신 용량 μ^R 은 다음과 같이 유도된다.

$$\lambda^R = \frac{TL_{tp}}{W_{tp}^R}, \quad \mu^R = \frac{UL_{tp}}{W_{tp}^R}$$

갱신에 대해 발생하는 작업의 입력량 λ^U 와 통신 용량 μ^U 는 다음 수식으로 계산할 수 있다.

$$\lambda^U = \frac{TL_{tp}}{W_{tp}^U}, \quad \mu^U = \frac{UL_{tp}}{W_{tp}^U}$$

그러므로 평균대기시간 AWT(Average Waiting Time)는 다음과 같이 유도된다.

$$\therefore AWT = \frac{\lambda^R}{(\mu^R)^2 - \mu^R \lambda^R} + \frac{\lambda^U}{(\mu^U)^2 - \mu^U \lambda^U}$$

그리고 평균 전송 시간 ATT는 각 사이트에서의 통신 용량에 대한 총 통신량으로 다음과 같이 쉽게 유도될 수 있다.

$$ATT = \sum_t \sum_p \frac{H_{ctp}}{UL_{tp}}$$

AWT와 ATT의 합으로서 대한 전체 통신지연시간 R_{COM} 은 다음과 같다.

$$\therefore R_{COM} = AWT + ATT \quad (1)$$

(2) disk I/O 지연시간

각 사이트에는 적절한 컴퓨터들이 선택되고 임의의 사이트에 배정된 컴퓨터에서는 처리할 수 있는 작업 계층에 대해서만 disk I/O가 발생하므로 작업이 수행되는 사이트에 할당된 컴퓨터 내에서만 disk I/O가 발생한다.

Little의 법칙[17]의 체재시간을 적용하여 각 사이트에서의 disk I/O 지연시간을 유도하기 위해 우선 사이트 s 에서의 disk I/O 용량 UIO_s 는 다음과 같이 유도된다.

$$UIO_s = \sum_m IO_m x_{ms} \text{ for all } s$$

여기에서 IO_m 은 단위시간당 수행되는 disk I/O 횟수로 컴퓨터 계층 m 의 IO 용량(capacity)이다.

사이트 s 에서 발생하는 disk I/O의 총수 TIO_s 는 다음과 같다.

$$TIO_s = \sum_i \sum_{i=1}^m f_i \cdot O_{is}$$

이 수식에서 O_{is} 는 사이트 s 에서 작업 계층 i 에 의해 발생하는 disk I/O 수이고, TIO_s 는 x_{ms} 와 관련하여 구해진다. 즉, $m=5$ 인 슈퍼컴퓨터는 모든 작업 계층을 수행할 수 있으므로 PC보다 상대적으로 TIO_s 가

크게 된다.

그러므로 Little의 법칙에 따라 총 disk I/O 지연시간은 다음과 같이 유도된다.

$$R_{IO}(c) = \sum_s O_{cs} \cdot \frac{1}{UIO_s - TIO_s}$$

(3) CPU 지연시간

CPU에서의 체제시간을 기술하기 위한 인자들을 다음과 같이 정의한다.

- p_m : 컴퓨터 계층 m 의 처리속도
(명령어 처리수 / unit time)
- $UCPU_s$: 사이트 s 에서의 CPU 용량
(명령어 처리수 / unit time)
- $TCPU_s$: 사이트 s 에서의 수행되는 총 명령어수
- U_{cs} : 사이트 s 에서 작업 계층 c 에 의한 CPU 처리수

disk I/O 지연시간을 유도한 절차와 유사하게 적용하여 CPU 지연시간 $R_{CPU}(c)$ 는 다음과 같이 유도된다.

$$UCPU_s = \sum_c \sum_m p_m x_{ms}$$

$$TCPU_s = \sum_c f_c \sum_s U_{cs}$$

$$\therefore R_{CPU}(c) = \sum_s U_{cs} \cdot \frac{1}{UCPU_s - TCPU_s}$$

이상의 통신 지연시간, disk I/O 지연시간, 그리고 CPU 지연시간을 통해, 전체 WRT를 다음과 같이 유도한다.

$$z_1(x, y) = \frac{\sum_c f_c (R_{COM}(c) + R_B(c) + R_{CPU}(c))}{\sum_c \mathcal{A}(c)} \quad (2)$$

2.1.2 투자비용(investment cost)

본 문제에서 IC는 각 사이트 안에 설치된 컴퓨터의 비용만으로 계산됨을 가정하면, IC는 다음과 같이 유도된다.

$$z_2(x, y) = \sum_{s \in S} \sum_{m \in M} (pp_m + pfm_m \cdot pm) \cdot x_{ms} + \sum_{s \in S} \sum_{k \in K} (psm_m \cdot sm) \cdot y_{sk} \quad (3)$$

위 식에서 pp_m 은 컴퓨터 계층 m 에 대한 프로세서

의 비용이고, pm 과 sm 은 각각 주기억장치와 보조기억 장치의 메가 당 평균비용을 나타낸다. 그리고 pfm_m 과 psm_m 은 컴퓨터 계층 m 에 대한 주기억장치와 보조기억 장치의 용량으로 각각 정의된다.

2.2 제약조건(constraints)

제안된 수학적 모델에서의 전체 해공간에서 가능해(feasible solution)를 분류하기 위한 5개의 제약조건들을 다음과 같이 기술한다.

1) 한 사이트에 한 개의 컴퓨터만 설치될 수 있다.

$$\sum_{m \in M} x_{ms} \leq 1 \quad \text{for } s \in S \quad (4)$$

2) 각 사이트의 주기억장치양은 도착하는 작업을 처리하는데 충분해야 한다.

$$\sum_{m \in M} pfm_m x_{ms} \geq \sum_{c \in C} b_{sc} e_c \quad (5)$$

3) 각 사이트의 보조기억장치양은 그 사이트에 할당된 데이터 파일의 모든 사본을 저장 할 수 있도록 충분해야 한다.

$$\sum_{m \in M} psm_m x_{ms} > \sum_{k \in K} y_{sk} v_k \quad \text{for } s \in S \quad (6)$$

4) 각 데이터 파일은 적어도 하나의 사이트에는 반드시 할당되어야 하고, 하나의 데이터 파일에 대해서는 여러 사이트로의 중복 할당이 허용된다.

$$\sum_{s \in S} y_{sk} \geq 1 \quad \text{for } k \in K \quad (7)$$

5) 각 사이트에 할당된 데이터파일이 존재하기 위해서는 기억장치를 보유한 컴퓨터가 설치되어져 있어야만 한다.

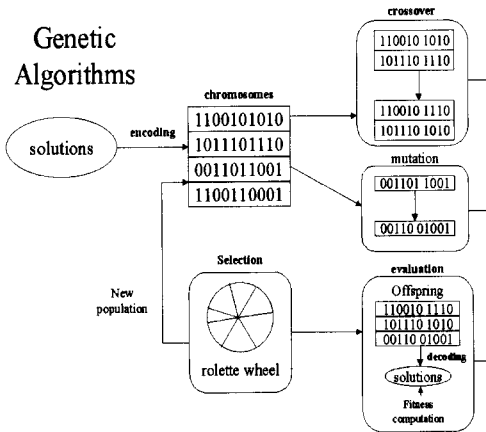
$$\sum_{m \in M} x_{ms} \geq \min\{1, \sum_{k \in K} y_{sk}\} \quad \text{for } s \in S \quad (8)$$

3. 유전자알고리즘 설계

본 장에서는 제안된 설계모델에서 각 사이트에서 최적의 컴퓨터 계층과 단편화된 데이터의 할당을 의미하는 최적해를 탐색하기 위해 GA 설계에 관한 논의를 한다. 2.2 절에서 기술한 제약조건을 검사하고 2.1 절

에서 기술한 두 목적함수로부터 유도되는 평가함수를 통해 해의 질(quality)을 평가한다.

유전자 알고리즘은 유전자의 진화를 모방한 경험적 탐색법 중의 하나로서 최근 여러 공학분야에서 많이 활용되고 있다[5,6]. 이에 대한 전체적인 구조를 (그림 2)에 나타내었고 각각의 요소들에 대해서는 다음절부터 설계방법을 제안하면서 자세히 언급한다.



(그림 2) 유전자 알고리즘의 전체적인 구조

3.1 염색체 표현(chromosome representation)

문제에 대한 해를 염색체로 표현하는 것은 유전자 알고리즘의 설계에 있어서 가장 기초적인 단계이고 이를 인코딩(encoding)이라 하고 역으로 염색체를 실제 해들로 바꾸는 조작을 디코딩(decoding)이라 한다. 그러므로 염색체 표현의 구조는 특히 메모리 사용의 효율성과 제약조건의 위반 및 처리의 복잡도면에서 전체 GA의 성능에 큰 영향을 준다. 그러므로 본 연구에서는 염색체 표현에 있어서 메모리 공간의 효율적 사용과 제약조건을 최소로 위반하도록 설계에 중점을 두었고 제안한 염색체 구조를 <표 2>에 나타내었다.

<표 2>에서처럼 선택된 컴퓨터를 배정하고, 연이어 할당된 데이터베이스 파일들을 표시하고, 0은 다음 사이트와의 구분을 위해 사용된다. 그러므로 최대 요구 기억공간은 $\theta((\text{데이터 파일 총수} + 2) * (\text{사이트의 총수}))$ 이다. 이는 행렬을 기반으로 한 표현 $\theta(\text{사이트 총수} * (\text{컴퓨터 계층 수} + \text{데이터 파일 총수}))$ 보다 훨씬 적은 메모리 공간이 요구됨을 이론적으로 증명할 수 있다.

3.2 초기화(initialization)

<표 2>의 구조에 따라 초기 염색체를 생성하여 염색체들의 집합으로 구성되는 초기 세대를 구성하는 과정을 초기화라고 한다. 이 단계에서는 3.1절에서 제안한 염색체 구조는 기본적인 제약조건이 반드시 보장됨을 알 수 있다. 즉, 제약조건식 (4)와 (7)이 염색체 구조 자체로써 제약조건을 위반하는 비가능해(in-feasible solution)를 생성하지 않는다. 단, 제약조건 (7)을 만족시키기 위해 같은 사이트 내에 동일한 데이터파일이 중복할당 되었는지 검사하는 절차가 추가로 요구된다.

초기화 단계를 통해 구성된 초기 세대는 다음절에서 기술된 교배(crossover)와 돌연변이(mutation) 과정을 통해 해를 변화시키고, 평가함수(evaluation function)를 통해 해의 질을 평가하고, 선택(selection) 단계를 통해 양질의 해로서 다음 세대를 구성하게 된다.

<표 2> 제안된 염색체 표현

사이트 1			사이트 2			
컴퓨터 계층	데이터 파일들	0	컴퓨터 계층	데이터 파일들	0	
사이트 n-1			사이트 n			
...	컴퓨터 계층	데이터 파일들	0	컴퓨터 계층	데이터 파일들	0

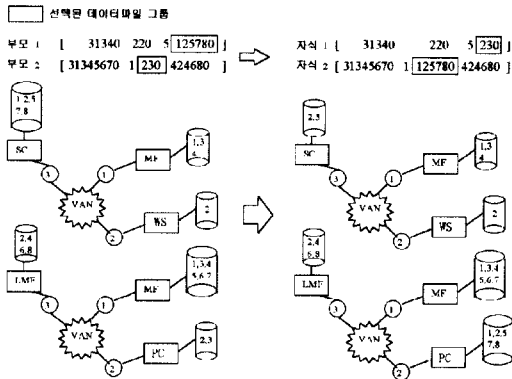
3.3 유전자 연산자들(genetic operators)

3.3.1 교배(crossover)

임의로 선택된 두 개의 해들(parent)과 관련하여 두 개의 새로운 해(offspring)를 생성하기 때문에 교배는 실제 해공간에서 지역적인 탐색을 한다. 제안된 교배에서도 기본적인 제약조건을 위반하지 않는 염색체를 생성하도록 설계에 중점을 두었고 선택된 부모의 양질의 유전자를 각 자식(offspring)들에게 유전되도록 하였다. (그림 3)에서처럼 교배는 다음의 과정을 통해 이루어진다.

- Step 1 : 교배할 두 부모를 임의로 선택한다.
- Step 2 : 각 부모에서 유전자를 교환할 사이트를 임의로 선택한다.
- Step 3 : 각 선택된 사이트 내에서 컴퓨터 계층은 무시하고 데이터베이스 파일들만 교환하여 자식들을 구성한다.

Step 4 : Step 1~3 과정을 교배확률(p_c)*세대수(population size)만큼 반복 수행한다.



(그림 3) 제안된 교배연산의 구조

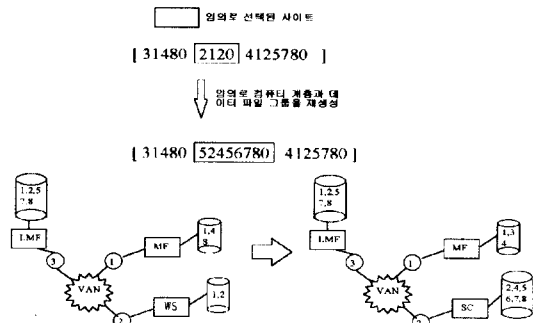
제안된 교배만으로는 협소한 해공간을 탐색함으로써 효율적인 해의 탐색시간이 오래 걸리고 극소해에 수렴할 가능성이 있으므로 해공간의 탐색 범위를 보안하기 위한 돌연변이 연산이 필요하다.

3.3.2 돌연변이(mutation)

돌연변이 연산은 전체 염색체에 대해 임의로 변환되는 특성에 의해 해공간에서 전역적인 탐색을 가능하게 함으로써 대개 교배연산의 협소한 탐색을 보안하기 위해 설계된다. 본 논문에서 제안한 돌연변이 연산은 각 선택된 사이트 내에서 컴퓨터 계층도 재생성 되는 것을 제외하고는 앞 절에서 기술한 교배와 거의 유사하고 (그림 4)에서처럼 다음의 과정에 따라 수행된다.

- Step 1 : 돌연변이 할 사이트를 임의로 선택한다.
- Step 2 : 선택된 사이트내의 컴퓨터계층과 데이터베이스 파일을 임의로 재생성 한다.
- Step 3 : Step 1, 2 과정을 돌연변이확률(p_m) * 세대수 만큼 반복 수행한다.

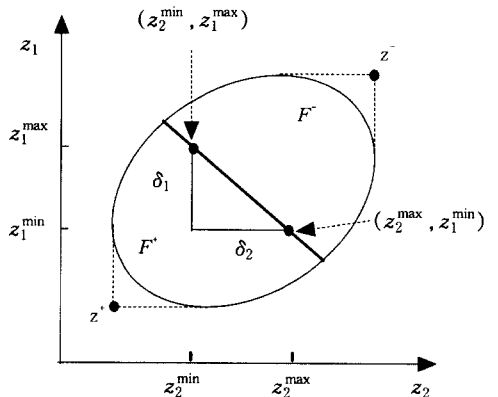
제안된 두 유전자 연산자 교배와 돌연변이는 제약조건식을 위반하는 해의 변환은 비가능해를 생성함으로써 이를 검출하고 조정하기 위한 수행 복잡도를 가중시키기 때문에 기본 제약조건식 (4), (7)을 반드시 보장하도록 설계하였다.



(그림 4) 돌연변이 연산의 구조

3.4 평가함수와 선택(evaluation function and selection)

제안된 모델에서는 두 개의 다목적 함수를 고려함으로써 하나의 해에 대해 두 목적 함수를 동시에 공정하게 평가해야 한다. 대개 임의의 해에 대해서 WRT가 우수한 경우에 대해서 IC층면에서는 비효율적인 상관관계를 가지기 때문에 두 목적함수의 공정한 평가를 위해 Gen[6]이 제안한 가중치합법(weighted sum method)를 이용하고 두 목적의 상관관계에 따라 결정자(decision maker)가 해를 선택하기 용이하도록 파레토 해(Pareto solution)로 표현한다.



(그림 5) 가중치합법의 개념

(그림 5)는 가중치합법에 대한 개념을 나타낸다. 좌표 (z_2^{min}, z_1^{max}) 와 (z_2^{max}, z_1^{min}) 으로 형성된 직선은 제약조건을 만족하는 타원 내부에서 $F+$ 영역과 $F-$ 영역으로 분할된다. 본 문제에서 제안한 두 목적함수는 최소화 문제이기 때문에 $F+$ 영역에 파레토 해들이 존재함을 알 수 있다. 파레토 해집합 F 와 두

좌표는 각 세대별로 갱신되고 세대의 진화에 따라 두 좌표에 의한 직선은 효율적인 해의 탐색을 하도록 한다. 가중치합법을 기초로 한 평가함수를 다음과 같이 제안한다.

Step 1 : t 번째 세대에서 2장에서 유도한 각각의 목적 함수에 대해 최적의 해와 최악의 해를 추출한다.

$$z_q^{\min(t)} = \min_k \{ z_q^{\min(t-1)}, z_q^{(t)}(T_k) | k=1, 2, \dots, pop_size \}$$

$$z_q^{\max(t)} = \max_k \{ z_q^{\max(t-1)}, z_q^{(t)}(T_k) | k=1, 2, \dots, pop_size \} \quad q=1, 2$$

여기에서 $z_q^{\max(t)}$ 와 $z_q^{\min(t)}$ 은 세대 t에서 함수 q에 대한 최대, 최소값을 나타낸다.

Step 2 : 각 함수에 대한 최소치와 최대치의 차이를 구한다.

$$\delta_q = | z_q^{\max(t)} - z_q^{\min(t)} |, \quad q=1, 2$$

Step 3 : 모든 제약조건식을 만족하면, 아래와 같이 각 해에 대한 평가 함수치를 구한다. 그리고 제약조건을 만족하지 않으면 상당히 큰 평가치를 주어 다음 세대를 구성하는 선택 단계에서 제외되게 한다.

$$eval(T_k) = \begin{cases} \frac{\delta_2 z_1(T_k) + \delta_1 z_2(T_k)}{\delta_1 + \delta_2}, & \text{if 모든 제약조건식만족} \\ k=1, 2, \dots, pop_size, \\ +\infty, & \text{otherwise} \end{cases} \quad (9)$$

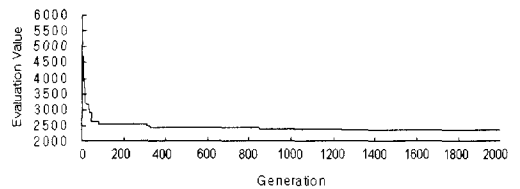
위에서 기술한 평가함수를 통해 효율적인 해들로 다음 세대를 구성하기 위한 선택 단계를 수행한다. 본 연구에서는 평가함수치가 우수한 해부터 정렬하고 효율적인 해부터 우선순위(priority)를 가지고 다음세대에 포함되도록 차별화를 주는 방법인 엘리트 선택법(elitist selection method)을 응용한다.

4. 모의실험 및 결과분석

본 논문에서 제안한 설계모델과 GA의 성능을 평가하기 위해 5개의 사이트와, 5개의 컴퓨터와 작업 계층

그리고 8개의 데이터 파일들을 포함하는 분산 데이터 베이스를 VAN상에 구현하는 문제를 다룬다. GA의 진화환경 인자로서 최대 세대수 $max_gen=2000$, 교집합률 $p_c=0.3$, 돌연변이 확률 $p_m=0.2$, 집단크기 $pop_size=20$ 으로 두고 수행하였다.

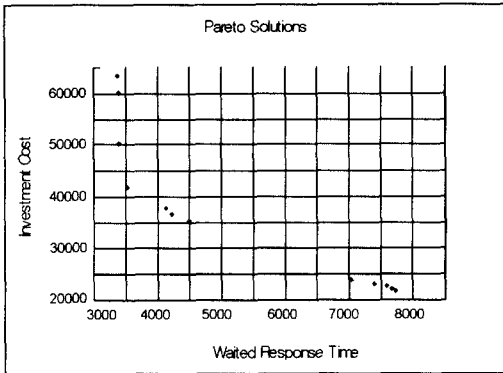
(그림 6)은 GA 진화환경 인자들을 통해 2000 세대에 대한 각 세대별 최적해의 평가함수치의 변화를 나타내었고 1241세대에서 효율적인 해가 구해졌음을 알 수 있다. 그리고 2000 세대 수행 후 추출된 파레토 해들을 (그림 7)에 나타내었고 이를 통해 WRT와 IC간의 상관관계를 알 수 있다. (그림 7)에 나타낸 파레토 해들은 GA 수행을 통해 구해진 해들에서 IC는 같고 WRT만 긴 의미 없는 해들을 제거한 것이다. 이러한 의미 없는 해들은 동급의 컴퓨터의 사용에 대해 로컬 사이트에서 요구하는 데이터 파일을 할당하지 않기 때문에, 리모트 사이트에서 트랜잭션 수행을 위해 필요한 데이터파일을 참조하기 위한 네트워크 사용에 의한 차이 때문에 발생한다.



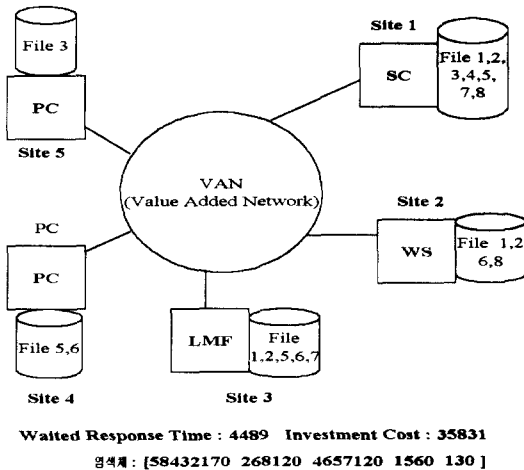
(그림 6) 평가함수치의 변화

대체로 슈퍼컴퓨터와 같은 상위 계층의 컴퓨터를 배정하면 IC의 증가를 보이지만 상대적으로 우수한 프로세서의 속도와 I/O 처리능력에 의해 WRT는 우수하다. 그리고 동일 데이터의 각 사이트로의 중복할 당은 질의가 발생한 사이트에서의 로컬처리를 가능하게 함으로써 원격 사이트로의 데이터 참조를 위한 추가적인 데이터 전송시간이 요구되지 않기 때문에 WRT에 대해 우수하지만 추가적인 갱신시간(data updating time)이 가중된다는 점을 알 수 있다.

(그림 8)은 파레토 해들에서 선택된 가장 효율적인 해에 대한 탐색체를 해로 표현한 결과로서 전체 DDS 구조를 그림으로 나타내고 있다. 추출된 파레토 해집합에서 두 목적 함수의 상관관계를 고려하여 가장 타협적인 해(the best compromised solution)를 구하기 위해 Yoon과 Hwang[13]이 제안한 TOPSIS 방법을 응용하였다.



(그림 7) GA 수행 후 추출된 파레토 해



(그림 8) 타협해에 대한 전체 DDS 구조

5. 결 론

본 연구에서 WRT와 IC의 상관관계를 고려하면서 단편화된 데이터 파일을 적합한 사이트로 할당하고 각 사이트에 대한 적합한 컴퓨터 계층을 선택하는 문제를 다루었다.

제안된 수학적 모델에서 WRT를 계산하기 위해 M/M/1 큐잉이론을 기초로 하여 각각의 통신지연, disk I/O 지연, CPU 지연시간을 고려하였고 무엇보다도 컴퓨터의 이질 기종 상에서 수행되는 작업의 크기(granularity)를 계층화함으로써 보다 실제적인 DDS의 설계모델을 제안하였다. 즉, 각 계층화된 작업이 수행되는 컴퓨터 플랫폼과의 상관관계를 고려하고 밀접한 관련성을 가지는 WRT와 IC를 DDS 설계를 위한 평가함수로써 동

시에 고려하였다. 무엇보다도 설계모델에서는 이질기종의 컴퓨터 상에서 발생하는 트랜잭션의 집합인 작업을 5개의 계층으로 세분화하여 각 작업이 수행될 때의 IC와 WRT의 상관관계의 고려에 역점을 두었다.

매우 난해한 문제로 유도되는 제안된 다목적 함수 모델(multiobjective function model)에서 효율적인 해를 탐색하기 위해 설계된 GA에서는 염색체 표현에서 행렬을 기반으로 한 유전자 표현에 비해 메모리 자원의 절약과 기본적인 제약조건식을 위반하지 않는 장점을 가지도록 설계에 중점을 두었다. 모의실험을 통한 성능 검토에서 GA 수행 후 파레토 해를 결정자에게 제공함으로써 각 목적의 중요도에 따라 DDS 구조를 선택할 수 있는 방안을 제시하였다.

향후과제로서 두 가지 방향을 고려할 수 있다. 첫째는 제안된 수학적 모델은 보다 실제적인 모델로의 확장을 들 수 있다. 즉, 분산데이터베이스의 중앙 집중식 데이터베이스에 비해 가장 큰 장점중의 하나가 어떤 사이트에서의 시스템오류가 발생하더라도 정상적인 동작이 가능하다는 점이다. 그러므로 모델의 설계에 있어서 중요한 요인(factor)으로서 시스템 오류에 대한 신뢰성을 극대화시키는 고려가 필요하다. 둘째는 최근 급증하는 인터넷의 구축에 따라 LAN환경에서 DDS를 구축하는 문제를 들 수 있고 평균 지연시간을 최소화하는 네트워크 라우팅(network routing) 문제와 데이터의 형태가 다양해짐에 따라 분산 환경에서의 이종 데이터베이스간의 상관관계를 고려한 멀티미디어 DDS 구현에 대한 연구가 필요하다.

참 고 문 헌

[1] S. T. March and S.K. Rho, "Allocating Data and Operations to Nodes in Distributed Database Design," IEEE Trans. on Knowledge and Data Engg., Vol.7, No.2, pp.305-317, Apr. 1995.
 [2] M. Ozsu and P. Valduriez, 'Principles of Distributed Database Systems', Prentice-Hall Inc., Englewood Cliffs; N.J. 1991.
 [3] S. Ram and R. E. Marsten, "A Model for Database Allocation Incorporating a Concurrency Control Mechanism," IEEE Trans. on Knowledge Data Eng., Vol.3, pp.389-395, Sep. 1991.
 [4] H. K. Jain, "A Comprehensive Model for the Design

of Distributed Computer Systems," IEEE Trans. on Software Eng., Vol.SE-13, pp.1092-1104, Oct. 1987.

[5] M. Gen and R. W. Cheng, 'Genetic Algorithms and Engineering Design,' John Wiley and Sons, New York, 1997.

[6] Z. Michalewicz, 'Genetic Algorithms+Data Structures = Evolution Programs,' second edition, Springer - Verlag, New York, 1994.

[7] R. J. A. Buhr and C. M. "Woodwide, Microscopic Economic Planning Models for Distributed Information Systems," INFOR, Vol.15, No.2, 1977.

[8] I. Mitrani and Sevcik, "Evaluating the Trade-off between Centralized and Distributed Computing," Proc. 1st Int. Conf. Distributed Computing Systems, pp.520-528, Oct. 1979.

[9] W. W. Chu, "Optimal File Allocation in Multiple Computer Systems," IEEE Trans. Comput., Vol. C-18, pp.885-889, 1969.

[10] S. Mahmoud and J. S. Riordan, "Optimal Allocation of Resources in Distributed Information Networks," ACM Trans. on Database System, Vol.1, No.1, pp. 66-78, 1976.

[11] J. P. Ignizio, D. F. Palmer, and C. Murphy, "A Multicriteria Approach to Super System Architecture Definition," IEEE Trans. on Comput., Vol.C-31, pp. 410-418, May 1982.

[12] A. Dutta and H. Jain, "A DSS for Distributed Computer System Design in the Presence of Multiple Conflicting Objectives," Decision Support System, Vol.1, No.3, pp.233-246, Sept. 1985.

[13] C. Hwang and K. Yoon, 'Multiple Attribute Decision Making Methods and Applications,' Springer-Verlag, Berlin, 1981.

[14] M. Gen, Y. Tsujimura and S.B. Ko, "Allocation Strategy for Distributed Database System with Fuzzy Data Using Genetic Algorithm," 5th European Congress on Intelligent Techniques and Soft Com-

puting (EUFIT'97), Vol.1, pp.737-742, Sep. 1997.

[15] S. Ko, Y. Tujumura and M. Gen, "Data Distribution Considered Communication Flow in Network Using Genetic Algorithm," 2nd Inter. Conf. on Knowledge-Based Intelligent Electronic Systems, Vol.2, pp. 264-271, Apr. 1998.

[16] J. Jo, S. Ko, S. Yoon, Y. Tsujimura, and M. Gen, "Processor Selectio and Data Allocation Problem in Distributed Database System Using GA," Inter. Conf. on APIEMS, pp.357-360, Oct. 1999.

[17] D. C. Little, "A Proof of the Queueing Formula $L = \lambda W$," Operation Res., Vol.9, pp.383-387, 1961.



고 석 범

e-mail : sbko@unicom.pknu.ac.kr
 1996년 동서대학교 컴퓨터공학과
 졸업(공학사)
 1998년 일본 아시카가공대 대학원
 경영정보공학과(공학석사)
 1999년~현재 부경대학교 대학원
 전자계산학과 박사과정

관심분야 : 분산데이터베이스, 병렬처리시스템, 유전자 알고리즘



윤 성 대

e-mail : sdyoun@mail.pknu.ac.kr
 1980년 경북대학교 컴퓨터공학과
 (공학사)
 1984년 영남대학교 전자계산학과
 (이학석사)
 1997년 부산대학교 전자계산학과
 (이학박사)

1992년~1995년 부산 공업대학교 전자계산소 소장
 1986년~현재 부경대학교 전자계산학과 교수
 2000년~현재 BK 사업단 컴퓨터 멀티미디어 분야 인
 력양성 사업팀장
 관심분야 : 병렬처리시스템, 병렬 운영체제