

유전자 알고리즘을 이용한 분산 데이터베이스 할당 방법론

박성진* · 박화규* · 손주찬* · 박상봉* · 백두권**

An Allocation Methodology on Distributed Databases Using the Genetic Algorithmsplications

Seong-Jin Park · Hwa Gyoo Park · Joo Chan Sohn · Sang Bong Park · Doo-kwon Baik

〈요 약〉

분산 환경에서 데이터의 할당(allocation)는 중요한 설계 이슈이다. 데이터의 할당은 분산 데이터에 대한 비용(cost) 감소, 성능(performance) 및 가용성(availability) 향상 등의 이점을 극대화할 수 있도록 최적화되어야 한다. 기존 연구들의 대부분은 트랜잭션의 수행 비용을 최소화하는 방향으로만 최적화된 데이터 할당 결과를 제시하고 있다. 즉, 비용, 성능 및 가용성을 모두 함께 고려하는 연구는 아직까지 제시된 결과가 없으며 이는 복잡한 모델에 대한 적절한 최적화 기법이 없기 때문이다.

본 연구에서는 분산 데이터의 이점들인 비용, 성능 및 가용성 등의 다중측면을 동시에 고려함으로써 데이터 할당에 대한 파레토 최적해를 제공하는 DAMMA (Data Allocation Methodology considering Multiple Aspects) 방법론을 제안하였다. DAMMA 방법론은 데이터 분할 과정을 통하여 생성된 최적의 단편들을 분산 시스템의 운용 비용, 수행 성능, 가용성 등의 요소를 고려하여 각 물리적 사이트에 중복 할당하는 파레토 최적해들을 생성해낼 수 있는 설계 방법론이다.

* 한국전자통신연구원 컴퓨터, 소프트웨어기술연구소

** 고려대학교 컴퓨터학과

1. 서 론

데이터 분산(data distribution)은 접근 비용 및 처리 시간을 줄여 줄 수 있을 뿐만 아니라 데이터에 대한 가용성도 향상시킨다. 즉, 분산 데이터베이스의 구축과 변환은 비용(cost) 감소, 성능(performance) 향상, 데이터의 가용성(availability) 및 신뢰성(reliability) 증가 등의 이점을 갖는다[1,2,4].

그러나, 데이터 분산은 데이터 전송을 위한 통신 비용과 데이터 제어를 위한 오버헤드 등 성능저하 요인들이 수반될 수 있으므로 최적의 사이트에 적정 수준으로 데이터를 할당하는 데이터 할당(data allocation)에 관한 연구가 필요하다. 예를 들어, 데이터의 중복도를 높일 경우 읽기 트랜잭션은 성능이 좋아지지만, 쓰기 트랜잭션은 복사본을 가지고 있는 모든 노드에 접근해야 하므로 통신 오버헤드가 증가하여 성능이 저하된다. 따라서, 비용, 성능, 가용성 등의 데이터의 분산에 따른 이점들은 데이터 할당 결과에 따라 감소할 수도, 증가할 수도 있다[3,6,7].

데이터 할당에 관한 연구의 대부분은 질의 처리 비용(query processing cost), 통신 비용(communication cost) 및 저장 비용(storage cost)의 합으로 구성되는 전체 처리 비용을 최소화하는 쪽에 집중되어 있으며[3,4,5,6,7,8] 일부 연구가 시스템의 성능 즉, 응답 시간의 최소화 혹은 생산량(throughput)의 최대화를 고려하였다. 최근에는 신뢰성을 고려한 연구도 제시되고 있다[3]. 결국, 기존 연구들은 고려되어야 할 모든 측면들을 고려하지 못하고 특정 측면들에 대한 제한된 해결책만을 제시하고 있다.

데이터 할당을 위해서는 분산 데이터의 이점인 비용, 성능 및 가용성 요소를 함께 고려해야 하며 시스템의 특성에 따라 실시간(real-time) 시스템인 경우에는 비용에, 고장목인(fault-tolerant) 시스템인 경우에는 가용성에 중점을 두어서 데이터를 할당시켜야 한다. 그리고, 나머지 요소들도

함께 고려해야 한다.

따라서, 데이터 할당에 관한 기존 연구는 데이터 할당시에 고려해야 할 요소들을 충분히 고려하고 있지 않으며 데이터 할당이 한 측면이 아닌 다양한 측면에 대한 요구를 동시에 충족시켜야 하는 복잡한 문제임을 생각할 때 다중 측면에 대한 고려가 필요하다[22,23].

본 논문에서는 이와 같은 문제점을 해결할 수 있는 새로운 데이터 할당 방법론으로 비용, 성능 및 가용성 요소를 함께 고려하는 DAMMA(Data Allocation Methodology considering Multiple Aspects) 방법론을 제안하였다.

2. DAMMA 방법론

분산 시스템 설계에 있어 데이터 할당은 매우 중요한 과정이다. 데이터 할당 결과에 따라 분산 시스템의 성능 및 비용 등이 좌우되기 때문이다 [5,13]. 그러나, 데이터 할당은 시스템에 끼치는 영향이 큰 만큼 고려해야 하는 측면도 다양하고 복잡하다.

DAMMA 방법론에서는 데이터 할당 모델 구성에 있어 다양한 측면을 동시에 고려하였다. 이를 위해 TCOC(Transaction COmmit Cost), TCOT (Transaction COmmit Time), TCOP (Transaction COmmit Probability) 3가지 평가 척도를 제시하였다. 이를 통해 데이터 할당 결과에 따른 각 측면값들을 평가할 수 있을뿐만 아니라 각 상태들의 연관성을 분석할 수 있다.

한편, 다양한 측면값을 갖는 데이터 할당 모델에 대해서 새로운 데이터 할당 기법이 필요하다. 기존 데이터 할당 기법들은 단일 측면값만을 갖는 할당 결과를 최적화시키는 알고리즘이므로 이에 대한 확장 또는 새로운 알고리즘이 필요하다. 기존의 데이터 할당 기법을 확장할 경우 각 측면값들에 대해 가중치를 주거나, 특정 측면을 제외한 나머지 측면값들을 제한 조건으로 사용하는 방법 등이 있으나 가중치값 혹은 제한조건

등의 객관적인 결정이 현실적으로 어렵다. 따라서, 다양한 각 측면값들의 트레이드-오프를 파악할 수 있는 다수의 파레토 최적해를 제공함으로써 사용자들이 그 중에서 원하는 해를 선택할 수 있도록 하는 방법이 타당하다. 이를 위해 DAMMA에서는 GA(Genetic Algorithms)를 확장하여 다수의 해들이 파레토 최적해에 수렴하도록 하는 PAGA(Pareto Genetic Algorithms)를 데이터 할당 기법으로 사용하였다.

DAMMA 방법론은 데이터 분할 과정을 통하여 생성된 최적의 단편들을 분산 데이터베이스 시스템의 운용 비용, 수행 성능, 가용성 등의 요소를 고려하여 각 물리적 사이트에 데이터를 중복 할당하는 파레토 최적해들을 생성하는 방법론이다. DAMMA 방법론에 관한 연구 내용은 크게 다음 2가지로 요약할 수 있다.

첫째, 데이터 할당시에 가용성 측면을 새로이 고려하기 위해 가용성 평가 척도로 TCOP(Transaction COmmit Probability) 척도를 제시하였다.

둘째, 다중 측면들에 대한 파레토 최적해들을 얻을 수 있도록 PAGA 최적화 기법을 사용하였다.

3. 데이터 할당 모델

DAMMA 방법론에서 데이터 할당을 위해 고려하고자 하는 측면은 운영비용, 수행성능, 가용성으로 각 측면들에 대한 정의와 평가 척도가 필요하다. 이에 따라 DAMMA에서는 3가지 측면 각각의 평가 척도를 TCOC (Transaction COmmit Cost), TCOT(Transaction COmmit Time), TCOP (Transaction COmmit Probability)로 정의하였다. TCOT는 성능 측면, TCOC는 비용 측면을 평가하기 위한 척도로서 기존 연구 결과를 참조하여 정의하였다. 한편, TCOP는 본 논문에서 새롭게 제시하는 가용성 평가 척도로서 트랜잭

션의 실행 완료를 위한 다중 경로의 가용성을 평가한 것이다.

3.1 가용성 측면(TCOP) 정의

3.1.1 TCOP : 트랜잭션 완료 가용성

기존 연구의 문제점 중 하나가 데이터의 중복을 결정하는데 있어 데이터 가용성에 대한 영향을 고려하지않고 있다는 점이다[19,21]. 가용성(availability)이란 회복될 수 있는 시스템이 특정 시점에서 동작 가능할 확률로 데이터 중복은 가용성에 큰 영향을 준다.

DAMMA에서는 데이터에 대한 가용성 평가 척도로 다중 경로를 통한 트랜잭션 완료 확률 TCOP(Transaction COmmit Probability)를 제시하였다. TCOP는 특정 트랜잭션들이 발생했을 때 필요로 하는 데이터에 대한 접근이 가능함으로써, 그에 따른 읽기 혹은 쓰기 작업이 정상적으로 수행된 뒤 완료될 확률이다. TCOP 척도에서는 트랜잭션 발생 사이트로부터 데이터 사이트까지 다수의 경로가 존재하는 경우, 트랜잭션의 성공적인 수행 완료 확률이 증가한다는 점을 고려한다. 즉, 동일 데이터에 대한 여러 경로중 최소한 하나의 경로만 가용하면 데이터에 대한 접근이 이루어지며 트랜잭션의 실행이 완료될 수 있다.

따라서, 특정 트랜잭션 t 에 대한 $TCOP_t$ 를 높이기 위해서는 트랜잭션 t 의 발생 노드에서 해당 트랜잭션 완료에 필요한 데이터 노드들까지의 경로가 되도록 많아야하고, 경로에 속한 사이트나 통신선들의 가용성이 높아야 한다. 여기서 경로가 가용하다는 것은 그 경로에 속한 처리기와 통신선, 데이터 저장 사이트의 저장 장치 모두가 가용하다는 것을 의미한다.

TCOP 정의에 사용되는 용어는 다음과 같다.

s: 트랜잭션 t의 발생 노드
d: 트랜잭션 t가 필요로하는 데이터의 저장 노드
N: 사이트 집합 ($s, d \in N, N_{Rt} \cup N_{Wt} = N$)
N_{Rt} : 읽기 트랜잭션 t의 데이터 노드 집합 ($d \in N_{Rt}, N_{Rt} \subset N$)
N_{Wt} : 쓰기 트랜잭션 t의 데이터 노드 집합 ($d \in N_{Wt}, N_{Wt} \subset N$)
F: 데이터(단편) 집합 ($f \in F$)
T: 트랜잭션 집합 ($t \in T, RT \cup WT = T$)
RT: 읽기 트랜잭션 집합 ($Rt \in RT, RT \subset T$)
WT: 쓰기 트랜잭션 집합 ($Wt \in WT, WT \subset T$)
$P_{s,d}$: 노드 s에서 노드 d로의 경로들의 집합 ($p_{s,d} \in P_{s,d}$)
$X_{P_{s,d}}$: 경로 $p_{s,d}$ 에 속하는 노드 x들의 집합 ($x \in X_{P_{s,d}}, s, d \notin X_{P_{s,d}}$)
$Y_{P_{s,d}}$: 경로 $p_{s,d}$ 에 속하는 링크 y들의 집합 ($y \in Y_{P_{s,d}}$)
Pr(x): 노드 x가 가용할 확률
Pr(y): 링크 y가 가용할 확률
Pr($p_{s,d}$): 경로 $p_{s,d}$ 가 가용할 확률
$COM_A(s,d)$: 노드 s와 노드 d사이의 단위통신 가용성
$CPU_A(s), CPU_A(d)$: 노드 s, 노드 d에서의 CPU 가용성
$IO_A(d)$: 노드 d에서의 IO 가용성
$TCOP_t(s,d)$: 트랜잭션 t의 실행이 완료될 확률

먼저, 임의의 경로 $p_{s,d}$ 가 가용할 확률 $Pr(p_{s,d})$ 는 경로 $p_{s,d}$ 를 구성하는 모든 노드 x와 링크 y들의 가용성의 곱으로 나타낼 수 있다. 경로 $p_{s,d}$ 가 중간 노드없이 단일 링크 y로만 구성되는 경우는 <식 1>, 하나 이상의 중간 노드 x들과 링크 y들로 구성된 경우는 <식 2>와 같이 정의된다.

$$Pr(p_{s,d}) = \prod_{y \in Y_{p_{s,d}}} Pr(y) \quad (1)$$

$$Pr(p_{s,d}) = \prod_{x \in X_{p_{s,d}}} Pr(x) \times \prod_{y \in Y_{p_{s,d}}} Pr(y) \quad (2)$$

한편, 단위 통신 경로에 대한 가용성은 트랜잭션이 발생한 노드 s에서 데이터가 저장된 노

d사이의 경로 $p_{s,d}$ 들중 적어도 하나가 작동중일 확률로 정의할 수 있다. 따라서, 노드 s와 노드 d간의 단위 통신 가용성 $COM_A(s,d)$ 는 경로 $p_{s,d}$ 의 개수가 하나인 경우에는 <식 3>과 같이, 경로 $p_{s,d}$ 의 개수가 둘이상인 경우는 다음 <식 4>와 같이 정의한다.

$$COM_A(s,d) = Pr(p_{s,d}) \quad (3)$$

$$\begin{aligned} COM_A(s,d) &= [Pr(p_{s,d})_1 + \{(1 - Pr(p_{s,d})_1) \\ &\times Pr(p_{s,d})_2 + \{(1 - Pr(p_{s,d})_1 \times (1 - Pr(p_{s,d})_2) \\ &\times Pr(p_{s,d})_3 + \dots + \{(1 - Pr(p_{s,d})_1) \\ &\times (1 - Pr(p_{s,d})_2) \times \dots \times (1 - Pr(p_{s,d})_{m-1}) \\ &\times Pr(p_{s,d})_m\}] \\ &= Pr(p_{s,d})_1 + \sum_{i=2}^m \left\{ \prod_{j=1}^{i-1} (1 - Pr(p_{s,d})_j) \times Pr(p_{s,d})_i \right\} \end{aligned} \quad (4)$$

(단, m은 경로 집합 $P_{s,d}$ 의 개수)

트랜잭션 t는 여러 경로들중에서 접근 가능한 경로 $p_{s,d}$ 를 선택함으로써 필요로 하는 데이터 노드에 접근하게 되고 이때, 그 가능성이 $COM_A(s,d)$ 이며 접근 가능한 경로 $p_{s,d}$ 가 하나도 없을 때 완료되지 못하고 철회된다.

3.1.2 TCOP 정의

TCOP는 읽기 트랜잭션과 쓰기 트랜잭션에 따라 각각 다르게 정의된다. 읽기 트랜잭션 완료 확률 $R_TCOP_t(s,d)$ 는 <식 3>, <식 4>의 $COM_A(s,d)$ 를 이용하여 다음과 같이 나타낼 수 있다. 먼저, 트랜잭션 t가 읽고자하는 데이터가 중복없이 할당된 경우엔 <식 5>와 같이 정의한다.

$$\begin{aligned} R_TCOP_t(s,d) &= \prod_{\alpha} CPU_A(s) \times \prod_{\beta} COM_A(s,d) \\ &\times \prod_{\gamma} CPU_A(d) \times \prod_{\delta} IO_A(d) \end{aligned} \quad (5)$$

또한, 트랜잭션 t가 읽고자하는 데이터가 둘이상 중복되어 할당된 경우엔 <식 6>과 같이

$R_TCOP_t(s,d)$ 를 정의한다. <식 5>, <식 6>, <7>에서 α 는 트랜잭션 t 의 발생 노드에서의 CPU 접근 횟수, β 는 단위 통신 경로를 통한 통신 횟수, γ 는 데이터 저장 노드에서의 CPU 접근 횟수 그리고, δ 는 데이터 저장 노드에서의 디스크 입출력 횟수를 나타낸다. $\alpha, \beta, \gamma, \delta$ 는 적용한 분산 트랜잭션 처리 기법(질의 처리 기법 및 동시성 제어 기법 등)에 따라 그 값이 결정된다.

$$\begin{aligned}
 R_TCOP_t(s, d) &= (\prod_{\alpha} CPU_A(s) \times \prod_{\beta} COM_A(s, d) \times \prod_{\gamma} CPU_A(d) \times \prod_{\delta} IO_A(d))_1 \\
 &+ \{(1 - (\prod_{\alpha} CPU_A(s) \times \prod_{\beta} COM_A(s, d) \times \prod_{\gamma} CPU_A(d) \times \prod_{\delta} IO_A(d))_1) \times (\prod_{\alpha} CPU_A(s) \times \prod_{\beta} COM_A(s, d) \times \prod_{\gamma} CPU_A(d) \times \prod_{\delta} IO_A(d))_2\} + \dots \\
 &+ \{(1 - (\prod_{\alpha} CPU_A(s) \times \prod_{\beta} COM_A(s, d) \times \prod_{\gamma} CPU_A(d) \times \prod_{\delta} IO_A(d))_1) \times (1 - (\prod_{\alpha} CPU_A(s) \times \prod_{\beta} COM_A(s, d) \times \prod_{\gamma} CPU_A(d) \times \prod_{\delta} IO_A(d))_2) \times \dots \\
 &\times (1 - (\prod_{\alpha} CPU_A(s) \times \prod_{\beta} COM_A(s, d) \times \prod_{\gamma} CPU_A(d) \times \prod_{\delta} IO_A(d))_{n-1}) \times (\prod_{\alpha} CPU_A(s) \times \prod_{\beta} COM_A(s, d) \times \prod_{\gamma} CPU_A(d) \times \prod_{\delta} IO_A(d))_n\} \\
 &= (\prod_{\alpha} CPU_A(s) \times \prod_{\beta} COM_A(s, d) \times \prod_{\gamma} CPU_A(d) \times \prod_{\delta} IO_A(d))_1 \\
 &+ \sum_{i=2}^n \{ \prod_{j=1}^{i-1} (1 - (\prod_{\alpha} CPU_A(s) \times \prod_{\beta} COM_A(s, d) \times \prod_{\gamma} CPU_A(d) \times \prod_{\delta} IO_A(d))_j) \times (\prod_{\alpha} CPU_A(s) \times \prod_{\beta} COM_A(s, d) \times \prod_{\gamma} CPU_A(d) \times \prod_{\delta} IO_A(d))_i \} \\
 &\text{(단, } n \text{은 데이터 중복 노드의 개수)} \tag{6}
 \end{aligned}$$

이때, 읽기 트랜잭션 t 의 완료를 위해서는 n 개의 데이터 중복 노드 d 에 대한 경로들중 적어도 하나의 $p_{s,d}$ 경로와 트랜잭션 발생 노드 및 데이터 저장 노드, 그리고 저장장치가 모두 가용하여야 한다.

쓰기 트랜잭션 완료 확률 $W_TCOP_t(s,d)$ 는 <식 3>, <식 4>의 $COM_A(s,d)$ 를 이용하여 다음과 같이 정의한다.

$$\begin{aligned}
 W_TCOP_t(s, d) &= \prod_{d \in N_m} (\prod_{\alpha} CPU_A(s) + \prod_{\beta} COM_A(s, d) \times \prod_{\gamma} CPU_A(d) \times \prod_{\delta} IO_A(d)) \tag{7}
 \end{aligned}$$

이때, 쓰기 트랜잭션 t 의 실행 완료를 위해서는 모든 데이터 저장 노드 d 에 대한 단위 통신 경로 $COM_A(s,d)$ 와 트랜잭션 발생 노드 및 데이터 저장 노드, 그리고 저장장치가 모두 가용하여야 한다. 결과적으로, 가용성 척도 $R_TCOP_t(s,d)$ 와 $W_TCOP_t(s,d)$ 는 각 자원들에 대한 가용성들의 곱으로 표현되기때문에 트랜잭션 처리 기법의 적용에 따른 접근 횟수($\alpha, \beta, \gamma, \delta$)에 따라 그 값이 결정된다.

3.2 비용 측면(TCOC) 정의

3.2.1 TCOC : 트랜잭션 완료 비용

본 논문에서는 일반적으로 고려되는 통신 비용, CPU 비용, I/O 비용, 저장 비용 등을 비용 요소로서 고려하였다. 본 논문에서는 두 노드간의 최소 비용 경로를 포함하도록 트랜잭션 완료 비용 TCOC(Transaction COmmit Cost)를 제시하였다. 즉, TCOC는 트랜잭션 처리를 위한 노드간의 최소 통신 비용과 각 노드에서의 처리 비용에 대한 합으로 표현된다.

한편, TCOC 정의를 위해서 TCOP 정의에 사용된 용어외에 다음과 같은 용어들을 추가로 규정한다.

Co(x) : 노드 x의 CPU 비용
Co(y) : 링크 y의 전송 비용
Co(p _{s,d}) : 경로 p _{s,d} 의 통신 비용
COM _c (s,d) : 노드 s와 노드 d사이의 단위통신 비용
CPU _c (s), CPU _c (d) : 노드 s, 노드 d에서의 CPU 비용
IO _c (d) : 노드 d에서의 IO 비용
TCOC _t (s,d) : 트랜잭션 t의 실행이 완료되기까지의 처리 비용

먼저, 임의의 경로 $p_{s,d}$ 에 대한 전송 비용 $Co(p_{s,d})$ 는 경로 $p_{s,d}$ 를 구성하는 모든 노드 x 의 CPU 비용과 링크 y 들의 전송 비용의 합으로 나타낼 수 있다. 경로 $p_{s,d}$ 가 중간 노드없이 단일 링크 y 로만 구성되는 경우는 <식 8>, 하나 이상의 중간 노드 x 들과 링크 y 들로 구성된 경우는 <식 9>와 같이 정의된다.

$$Co(p_{s,d}) = \prod_{y \in Y_{p_{s,d}}} Co(y) \quad (8)$$

$$Co(p_{s,d}) = \prod_{x \in X_{p_{s,d}}} Co(x) + \prod_{y \in Y_{p_{s,d}}} Co(y) \quad (9)$$

단위 통신 경로에 대한 전송 비용은 트랜잭션이 발생한 노드 s 에서 데이터가 저장된 노드 d 사이의 경로 $p_{s,d}$ 들중의 최소 전송 비용이다. 따라서, 두 노드간의 단위 통신 비용 $COM_C(s,d)$ 는 다음과 같이 표현한다.

$$COM_C(s,d) = \min_{p_{s,d} \in P_{s,d}} [Co(p_{s,d})] \quad (10)$$

트랜잭션 t 는 여러 경로들중에서 전송 비용이 최소인 경로 $p_{s,d}$ 를 선택하여 필요로하는 데이터 노드에 접근하게 되며 이때 소요되는 비용이 $COM_C(s,d)$ 이다.

3.2.2 TCOC 정의

TCOC는 읽기 트랜잭션과 쓰기 트랜잭션에 따라 각각 다르게 정의된다.

읽기 트랜잭션 완료 비용 $R_TCOC_t(s,d)$ 는 <식 10>의 $COM_C(s,d)$ 를 이용하여 다음과 같이 정의한다.

$$\begin{aligned} R_TCOC_t(s,d) &= \min_{d \in N_{Rt}} (\sum_{\alpha} CPU_C(s) + \sum_{\beta} COM_C(s,d) \\ &+ \sum_{\gamma} CPU_C(d) + \sum_{\delta} IO_C(d)) \quad (11) \end{aligned}$$

읽기 트랜잭션 t 의 완료는 모든 데이터 중복

노드 d 에 대한 단위 통신 비용 $COM_C(s,d)$ 와 트랜잭션 발생 노드 및 데이터 저장 노드의 CPU 처리 비용, 그리고 데이터 저장 노드의 I/O 비용들의 합중 최소의 비용을 갖는다. <식 11>, <식 12>에서 α 는 트랜잭션 t 의 발생 노드에서의 CPU 접근 횟수, β 는 단위 통신 경로를 통한 통신 횟수, γ 는 데이터 저장 노드에서의 CPU 접근 횟수 그리고, δ 는 데이터 저장 노드에서의 디스크 입출력 횟수를 나타낸다.

쓰기 트랜잭션 완료 비용 $W_TCOC_t(s,d)$ 는 <식 10>의 $COM_C(s,d)$ 를 이용하여 다음과 같이 정의한다.

$$\begin{aligned} W_TCOC_t(s,d) &= \sum_{d \in N_w} (\sum_{\alpha} CPU_C(s) + \sum_{\beta} COM_C(s,d) \\ &+ \sum_{\gamma} CPU_C(d) + \sum_{\delta} IO_C(d)) \quad (12) \end{aligned}$$

<식 12>에서 쓰기 트랜잭션 t 는 모든 데이터 저장 노드 d 에 대한 단위 통신 비용 $COM_C(s,d)$ 와 트랜잭션 노드 및 모든 데이터 저장 노드의 CPU 처리 비용, 그리고 모든 데이터 저장 노드의 I/O 비용들의 합만큼 비용을 소요한 후 종료한다.

3.3 성능 측면(TCOT) 정의

3.3.1 TCOT : 트랜잭션 완료 시간

기존 데이터 할당 기법들 중 성능을 평가 척도로 하는 경우 대부분이 응답 시간의 최소화를 목표로 하고 있다. 본 논문에서도 기존 연구와 동일하게 트랜잭션에 대한 응답 시간을 성능 척도로 고려하며 여기에는 앞서 제시한 비용 요소들에 대한 통신 시간, CPU 처리 시간, I/O 시간 등을 성능 요소로서 고려하였다. 본 논문에서는 두 노드간의 최소 시간 경로를 포함하도록 트랜잭션 완료 시간 TCOT(Transaction Commit Time)를 제시하였다. TCOT는 트랜잭션 처리를

위한 노드간의 최소 통신 시간과 각 노드에서의 처리 시간에 대한 합으로 표현된다.

한편, TCOT 정의를 위해서 TCOP 정의에 사용된 용어외에 다음과 같은 용어를 추가로 가정한다.

$Tm(x)$: 노드 x 의 CPU 처리 시간
 $Tm(y)$: 링크 y 의 전송 시간
 $Tm(p_{s,d})$: 경로 $p_{s,d}$ 의 통신 시간
 $COM_T(s,d)$: 노드 s 와 노드 d 사이의 단위통신 시간
 $CPU_T(s), CPU_T(d)$: 노드 s , 노드 d 에서의 CPU 처리 시간
 $IO_T(d)$: 노드 d 에서의 IO 시간
 $TCOT_t(s,d)$: 트랜잭션 t 의 실행이 완료되기까지의 처리 시간

먼저, 임의의 경로 $p_{s,d}$ 에 대한 전송 시간 $Tm(p_{s,d})$ 는 경로 $p_{s,d}$ 를 구성하는 모든 노드 x 의 CPU 시간과 링크 y 들의 전송 시간의 합으로 나타낼 수 있다. 경로 $p_{s,d}$ 가 중간 노드없이 단일 링크 y 로만 구성되는 경우는 <식 13>, 하나 이상의 중간 노드 x 들과 링크 y 들로 구성된 경우는 <식 14>와 같이 정의된다.

$$Tm(p_{s,d}) = \prod_{y \in Y_{p_{s,d}}} Tm(y) \quad (13)$$

$$Tm(p_{s,d}) = \prod_{x \in X_{p_{s,d}}} Tm(x) + \prod_{y \in Y_{p_{s,d}}} Tm(y) \quad (14)$$

단위 통신 경로에 대한 전송 시간은 트랜잭션이 발생한 노드 s 에서 데이터가 저장된 노드 d 사이의 경로 $p_{s,d}$ 들중의 최소 전송 시간이다. 따라서, 두 노드간의 단위 통신 시간 $COM_T(s,d)$ 는 다음과 같이 표현된다.

$$COM_T(s,d) = \min_{p_{s,d} \in P_{s,d}} (Tm(p_{s,d})) \quad (15)$$

트랜잭션 t 는 여러 경로들중에서 전송 시간이 최소인 경로 $p_{s,d}$ 를 선택하여 필요로하는 데이터

노드에 접근하게 되며 이때 소요되는 시간이 $COM_T(s,d)$ 이다.

3.3.2 TCOT 정의

TCOT는 읽기 트랜잭션과 쓰기 트랜잭션에 따라 각각 다르게 정의된다.

먼저, 읽기 트랜잭션 완료 시간 $R_TCOT_t(s,d)$ 는 <식 15>의 $COM_T(s,d)$ 를 이용하여 다음과 같이 정의한다.

$$\begin{aligned} R_TCOT_t(s,d) &= \min_{d \in N_R} (\sum_{\alpha} CPU_T(s) + \sum_{\beta} COM_T(s,d) \\ &\quad + \sum_{\gamma} CPU_T(d) + \sum_{\delta} IO_T(d)) \quad (16) \end{aligned}$$

읽기 트랜잭션 t 의 완료는 모든 데이터 중복 노드 d 에 대한 단위 통신 시간 $COM_T(s,d)$ 와 트랜잭션 노드 및 데이터 저장 노드의 CPU 처리 시간, 그리고 데이터 저장 노드의 I/O 시간들의 합중 최소의 시간을 갖는다. <식 16>, <식 17>에서 α 는 트랜잭션 t 의 발생 노드에서의 CPU 접근 횟수, β 는 단위 통신 경로를 통한 통신 횟수, γ 는 데이터 저장 노드에서의 CPU 접근 횟수 그리고, δ 는 데이터 저장 노드에서의 디스크 입출력 횟수를 나타낸다.

쓰기 트랜잭션 완료 시간 W_TCOT_t 는 <식 15>의 $COM_T(s,d)$ 를 이용하여 다음과 같이 정의한다.

$$\begin{aligned} W_TCOT_t(s,d) &= \sum_{d \in N_W} (\sum_{\alpha} CPU_T(s) + \sum_{\beta} COM_T(s,d) \\ &\quad + \sum_{\gamma} CPU_T(d) + \sum_{\delta} IO_T(d)) \quad (17) \end{aligned}$$

<식 17>에서 쓰기 트랜잭션 t 는 모든 데이터 저장 노드에 d 에 대한 단위 통신 시간 $COM_T(s,d)$ 와 트랜잭션 노드 및 모든 데이터 저장 노드의 CPU 처리 시간, 그리고 데이터 저장 노드의 I/O 시간들의 합만큼 경과된 후 종료된다.

4. 파레토 최적화 기법

유전자 알고리즘(GA; Genetic Algorithms)은 다양한 문제에 적용가능하면서도 전역적인 최적해를 찾는 효과적인 최적화 알고리즘으로 최근 많은 분야에 적용되고 있다[10,14,15]. DAMMA 방법론에서는 GA를 다중 측면 최적화에 적합하도록 파레토-박스-선택 기법[22]과 파레토-crowding 기법[22]을 통하여 확장한 PAGA (Pareto Genetic Algorithms)를 사용하였다[22].

GA를 최적화 기법에 관한 기존 연구의 대부분은 주로 단일 측면 함수를 갖는 문제들을 대상으로 적용한 것으로 이를 다중 측면 최적화에 적용하기 위해서는 다수의 목적 함수들이 하나의 적합도 함수로 결합되어야 한다. 이것은 다중 측면들을 벌칙 함수로 사용하여 제한 조건으로 전환하거나 각 측면에서 평가한 값들에 특정 가중치를 주는 방법 등을 사용함으로써 가능하다. 하지만, 이 경우 각 측면들에 대한 가중치 설정이 어렵고 주된 평가 척도가 결정되지 않는 경우 벌칙 함수의 대상 선정 역시 불가능하다. 특히, 다양한 측면을 동시에 고려해야 하는 경우, GA의 탐색이 한 방향으로만 고정됨으로써 다양한 최적해를 찾기 어려운 문제점이 있다[14].

특히, 다중 측면 최적화에서는 각 측면간의 트레이드-오프가 존재하는 경우, 모든 측면에서 다른 어떤 해보다도 우수한 단일 해를 찾는다는 것이 불가능하다. 따라서, 각 측면을 다양하게 고려하면서 복수 해들을 생성하는 것이 바람직하다[8,20]. 이 경우 파레토 최적 집합(Pareto-optimal set)으로 알려진 해들은 다중 측면 최적화의 적절한 목표가 될 수 있으며 생성된 파레토 최적해들은 자연스럽게 각 측면간의 트레이드-오프 경계를 따라 생성된다. 이때, GA는 복수개의 해들로 한 세대를 구성하여 여러 해들에 대한 탐색을 동시에 수행하므로 다중 측면 최적화 문제에 적합한 특성을 갖는다. 즉, GA는 탐

색 과정 자체가 병렬적으로 수행됨으로써 다양한 해를 생성하고 유지한다는 점에서 이점이 있다[8,16,18].

제시한 DAMMA 방법론을 다음과 같은 DAP에 적용하였다. 가정한 DAP(Data Allocation Problem)는 <그림 1>과 같은 4개의 물리적인 사이트에서 발생하는 읽기 및 쓰기 트랜잭션에 대한 처리 비용, 응답시간, 가용성을 최적화하도록 적정 사이트에 할당하는 문제로 가정하였다.

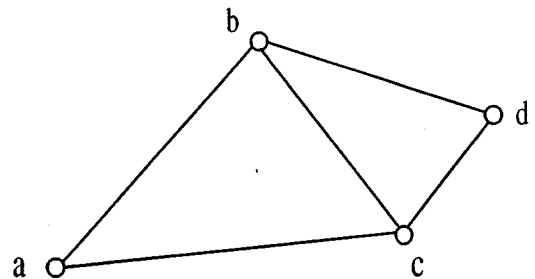


그림 1: 분산 환경

먼저, <그림 1>과 같은 분산 환경에서의 DAP에 대한 최적화를 위해서 다음과 같은 측면들을 평가하였다.

- $TCOC_t$: 비용 측면
- $TCOT_t$: 응답시간 측면
- $TCOP_t$: 가용성 측면

트랜잭션 t 에 대한 $TCOC_t$, $TCOT_t$, $TCOP_t$ 측면값은 읽기/쓰기 트랜잭션 평가 모델에 의해 평가될 수 있다. MAPN(Multiple Aspects Petri Net) 그래프는 distributed 2-phase locking 기법[2]과 centralized 2-phase commit 규칙[2]을 가정하고 모델링한 것이다. 또한, ROWA (Read-One-Write-All) 규칙을 사용함을 가정하였다.

이때, 초기 세대(population) 구성은 임의로 랜덤하게 선택되며 파레토 최적화를 위한 적합도

함수(fitness function)들은 트랜잭션이 발생하는 경우를 가정하여 다음과 같이 전체 트랜잭션의 비용합(C_T), 시간합(T_T), 가용성합(A_T)으로 정의된다. 이때, F_t 는 트랜잭션 t 의 발생 빈도를 의미한다.

$$\begin{aligned}
 C_T(\text{전체 트랜잭션 완료 비용의 합}) &= \text{읽기 트랜잭션들의 TCOC}_t \text{ 합} \\
 &+ \text{쓰기 트랜잭션들의 TCOC}_t \text{ 합} + \text{저장비용} \\
 &= \sum_{t \in RT} (F_t \times R_TCOC_t(s,d)) \\
 &+ \sum_{t \in WT} (F_t \times W_TCOC_t(s,d)) \\
 &+ \sum_{i,j} \sum_{v_j} (X_{ij} \times L(D_i) \times Media_co(N_j))
 \end{aligned}$$

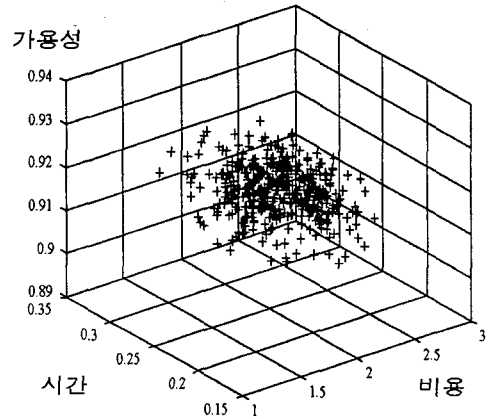
$$\begin{aligned}
 T_T(\text{전체 트랜잭션 완료 시간의 합}) &= \text{읽기 트랜잭션들의 TCOT}_t \text{ 합} \\
 &+ \text{쓰기 트랜잭션들의 TCOT}_t \text{ 합} \\
 &= \sum_{t \in RT} (F_t \times R_TCOT_t(s,d)) \\
 &+ \sum_{t \in WT} (F_t \times W_TCOT_t(s,d))
 \end{aligned}$$

$$\begin{aligned}
 A_T(\text{전체 트랜잭션 완료 확률의 합}) &= \text{읽기 트랜잭션들의 TCOP}_t \text{ 합} \\
 &+ \text{쓰기 트랜잭션들의 TCOP}_t \text{ 합} \\
 &= \sum_{t \in RT} (F_t \times R_TCOP_t(s,d)) \\
 &+ \sum_{t \in WT} (F_t \times W_TCOP_t(s,d))
 \end{aligned}$$

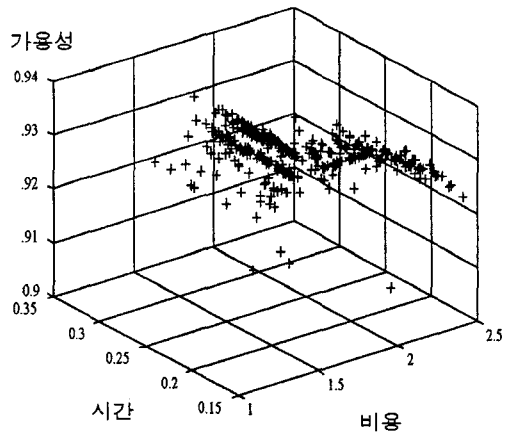
<그림 2>는 위의 적합도 함수들을 적용하여 DAMMA 방법론에 의해 얻어진 DAP 파레토 최적해들을 그래프로 나타낸 것이다. 다음 실험 결과는 세대크기(population size) 400, 세대 생성횟수(generation number) 10000, 교차연산 발생확률(crossover rate) 0.9, 돌연변이 발생확률(mutation rate) 0.1로 설정했을 경우의 얻어진 값들이다.

<그림 2>(a)와 <그림 2>(b)를 통해서 초기해(gen=0)와 최종해(gen=10000)를 비교함으로써 DAMMA 방법론에 의한 결과의 진화성 혹은 수

렴성을 분석할 수 있다. 먼저, <그림 2>(a)는 초기해와 최종해를 3차원 공간에 나타낸 그래프



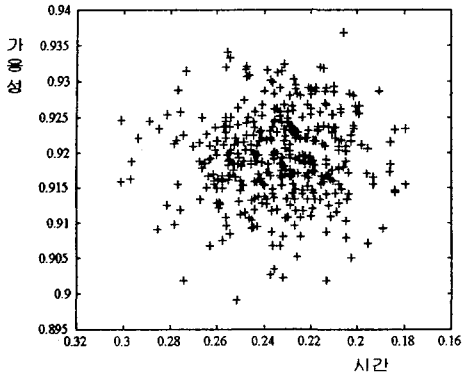
(a) 비용/시간/가용성 비교(gen=0)



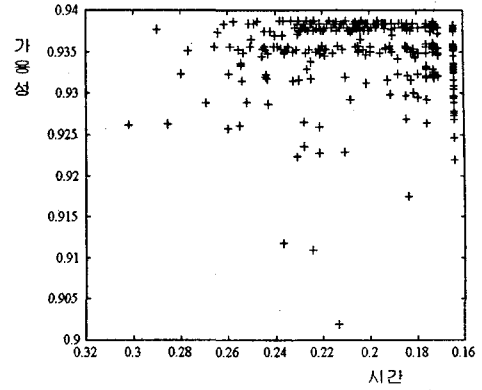
(b) 비용/시간/가용성 비교(gen=10000)

그림 2: DAMMA 방법론에 의한 최종해 분석

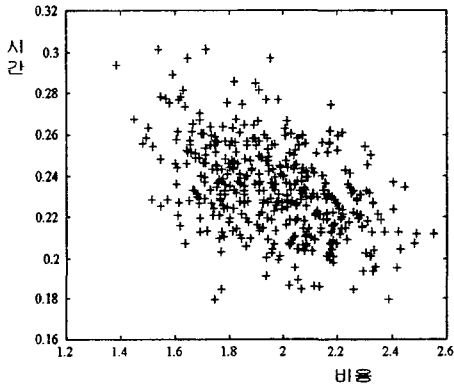
이며 <그림 2>(b)를 통해서 전체적으로 각 측면에서 우성인 방향으로 해들이 진화되었으며 일정한 경계안에 수렴되어 있음을 확인할 수 있다.



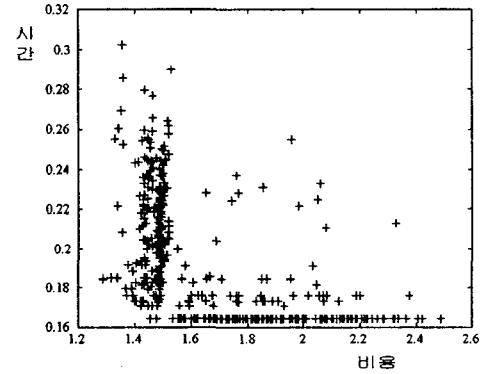
(a) 시간/가용성 비교(gen=0)



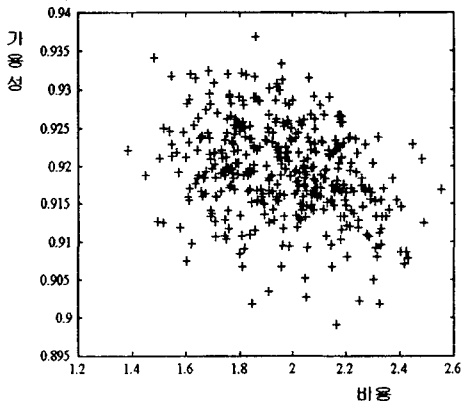
(d) 시간/가용성 비교(gen=10000)



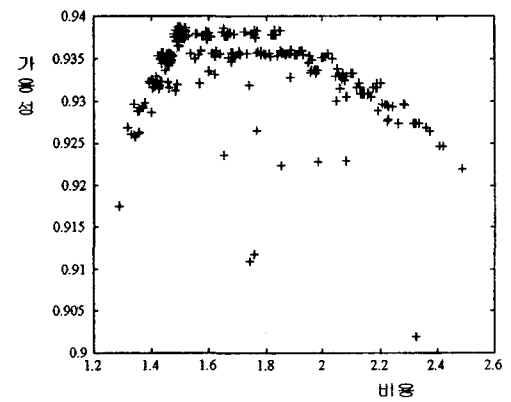
(b) 비용/시간 비교(gen=0)



(e) 비용/시간 비교(gen=10000)



(c) 비용/가용성 비교(gen=0)



(f) 비용/가용성 비교(gen=10000)

그림 3: DAMMA 방법론에 의한 초기해와 최종해 분석

<그림 3>은 최종해들이 후보해들에 비해서 비용, 시간 측면에서는 감소 방향으로, 가용성 측면에서는 증가 방향으로 진화 및 수렴함을 분명하게 보여준다. <그림 3>의 (f)의 경우, 비용 측면에서 저장 비용의 비중이 크기 때문에 비용이 증가함에 따라 가용성은 처음에는 증가하다가 점차 둔화되며 결국에는 다시 감소함을 볼 수 있다. 이것은 데이터 가용성이 일정 수준의 데이터 중복을 통해 증가되기 때문에 그 수준까지는 자연스럽게 비용 증가와 함께 증가되지만 일정 수준을 넘을 경우에는 더 이상 증가되지 않고 결국에는 불필요한 데이터 중복에 따른 가용성의 감소를 가져오기 때문이다.

5. 결 론

본 연구에서는 DAP에 대해 분산 데이터베이

스의 이점들인 비용, 성능 및 가용성 요소를 함께 고려하는 새로운 데이터 할당 방법론으로 DAMMA(Data Allocation Methodology considering Multiple Aspects)을 제안하였다.

제안한 DAMMA 방법론은 데이터 분할 과정을 통하여 생성된 최적의 단편들을 분산 데이터베이스 시스템의 운용 비용, 수행 성능, 가용성 등의 요소를 고려하여 각 물리적 사이트에 중복 할당함으로써 파레토 최적해들을 생성하는 설계 방법론이다. 또한, DAMMA 방법론은 실제 분산 데이터베이스 설계시에 유용하게 적용할 수 있을 뿐만 아니라 다중 측면을 고려해야하는 모든 최적화 문제에도 적용할 수 있다.

추후 연구로는 GA의 병렬화를 통한 최적화 기법의 속도 향상과 초기해 생성시의 오류 제거 등을 통한 파레토 해들의 다양성 향상 등이 필요하다.

<참고 문헌>

- [1] P.M.G. Apers, "Data allocation in distributed database systems," *ACM Trans. on Database Systems*, Vol. 13, No. 3, pp. 263-304, 1988.
- [2] S. Ceri and G. Pelagatti, *Distributed Database: Principles and Systems*, New York, N. Y. : McGraw-Hill Book Company, 1984.
- [3] A. Kumar, R.M. Pathak, and Y.P. Gupta, "Genetic algorithm based approach for file allocation on distributed systems," *Computers Operations Research*, Vol. 22, pp. 41-54, 1995.
- [4] S.T. March, and S. Rho, "Allocating data and operations to nodes in distributed database design," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 7, pp. 305-317, 1995.
- [5] M. Ozsu and P. Valduriez, *Principles of Distributed Database Systems*, Englewood Cliffs, N.J. : Prentice-Hall Inc., 1991.
- [6] S. Ram, and R.E. Marsten, "A model for database allocation incorporating a concurrency control mechanism," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 3, No. 3, pp. 389-395, 1991.
- [7] S. Rho, and S.T. March, "A nested genetic algorithm for distributed database design", *Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences*, pp. 33-42, 1994.
- [8] B. W. Wah, "File placement on distributed computer systems," *IEEE Trans. on Computer*, pp. 23-32, 1984.

- [9] R. Blankinship, A.R. Hevner, and S.B. Yao, "An iterative method for distributed database design," *Proceedings of the 17th International Conference on Very Large Database Design*, Barcelona, Spain, pp. 389-400, 1991.
- [10] L. Chamber, *Practical Handbook of Genetic Algorithms (I) (II)*, CRC Press, 1995.
- [11] W.W. Chu, Optimal file allocation in a computer network, *Computer Communications Systems*, Prentice Hall, Englewood Cliffs, N.J., 1973.
- [12] D.W. Cornell and P.S. Yu, "On optimal site assignment for relations in the distributed database environment," *IEEE Trans. on Software Engineering*, Vol. 15, No. 8, pp. 1004-1009, 1989.
- [13] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, Benjamin/Cummings Publishing, 1989.
- [14] C.M. Fonseca and P.J. Fleming, "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization", *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufman, pp. 416-423, 1993.
- [15] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing, 1989.
- [16] J. Horn, N. Nafpliotis, and D.E. Goldberg, "A niched pareto genetic algorithm for multiobjective optimization", *Proceedings of the First IEEE Conference on Evolutionary Computation*, IEEE World Congress on Computational Intelligence, Vol. 1, pp. 82-87, 1994.
- [17] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Second extended Edition, Springer-Verlag, 1992.
- [18] T. Murata and H. Ishibuchi, "MOGA: Multi-Objective Genetic Algorithms", *IEEE International Conference on Evolutionary Computation*, Vol. 1, pp. 289-294, 1995.
- [19] Seong-Jin Park, Doo-Kown Baik, "A Data Allocation Considering Data Availability in Distributed Database Systems", *1997 International Conference on Parallel and Distributed Systems*, 1997. 12. to be published.
- [20] J.D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms", *Proceedings of an International Conference on Genetic Algorithms and their Application*, pp. 93-100, 1985.
- [21] 박성진, 백두권, "분산데이터베이스에서의 데이터 중복을 위한 TCOP 가용성 모델" *봄학술발표논문집*, 한국정보과학회, 24권 1호, pp. 155-158, 1997.
- [22] 박성진, 백두권, "다중 측면 데이터 할당 방법론에서의 파레토 최적화 기법", *가을학술발표논문집*, 한국정보과학회, 24권 2호, pp. 477-480, 1997.
- [23] 박성진, 백두권, "다중 측면 페트리 넷 모델링을 이용한 데이터 할당 방법론", *가을학술발표논문집*, 한국정보처리학회, pp. 416-421, 1997.