

# 개인 통신 서비스(PCS)의 지능망을 위한 병렬 연역 데이터 베이스 시스템

## (Parallel Deductive Database Systems for Personal Communication Service)

이 원 석\*

### 요 약

개인 통신 서비스(PCS)는 미래의 고도 정보화 시대에서 정보 전달의 중요성에 대한 요구를 충족시키기 위한 총체적인 개인 정보 전달 서비스이다. 기존의 통신 전달 매체를 기본으로 한 통신 방법에서 개인의 이동성을 보장하는 개인 통신 서비스의 구현을 위해서는 상당한 지능을 갖춘 지능망의 사용이 필연적이며, 이러한 정보를 처리하기 위해서는 고속 대용량의 데이터 베이스 시스템의 개발이 중요한 과제이다. 연역 데이터 베이스는 데이터의 일반적인 법칙을 연역 법칙으로 표현할 수 있으므로 지능망의 데이터 관리에 적절한 시스템으로 고려된다. 지능망 데이터 베이스는 신속한 통신 전달을 위해 빠른 시간안에 주어진 질의의 처리가 가능해야 하므로, 이 논문에서는 앞으로 보편적으로 사용될 병렬 시스템에서 연역 데이터 베이스를 주기억 장치 데이터 베이스 시스템으로 구현하는 방안을 제시한다.

### Abstract

The future Personal Communication Service(PCS) is the total personal information communication service to satisfy the importance of information in a highly information-oriented society. Compared with the conventional communication method, it is necessary to provide considerable intelligence to the communication network in order to implement the PCS<sup>s</sup> which guarantees the mobility of each individual. Consequently, the PCS needs a fast large database system which stores the information of each individual. To achieve this goal, this paper presents the parallel implementation of deductive database systems as a main-memory database system.

### I. 서 론

미래의 정보화 사회에서 근간이 될 정보의 신속하고 정확한 전달은 고도로 정보화된 사회에서는 필요

불가결한 기본적인 요소이므로 세계 각국은 도래할 정보화 사회를 대비한 개인 휴대 통신의 실현을 위한 많은 연구를 진행하고 있다.

개인 통신 서비스(Personal Communication Service)는 다양하고 복잡한 고도의 정보화 시대를 위한 유선과 무선 시스템의 통합적인 차세대 정보 통신 기술로 미래의 통신 분야에 주축으로 각광을

\*삼성전자 컴퓨터 부문 선임연구원

받고 있다.<sup>12,9</sup> 개인 휴대 통신의 기본 개념은 시간, 장소, 휴대의 제약이 초월하여 각 개인에게 전 신속하게 전달하는 것으로 음성, 데이터 그리고 영상 등의 다양한 형태의 정보를 전달할 수 있는 ISDN(Integrated Service Data Network)과 호환성을 유지하는 총체적인 정보 통신 수단이다. 전화 또는 단말기를 중심으로한 기존의 통신 방법은 전화 통화의 70-80%가 상대 통화 대상의 부재로 통화가 연결되지 못하며 시간과 장소의 제약없이 전화를 사용하려면 전화기를 항상 휴대하여야 한다.

전화기 또는 단말기를 정보 전달의 마지막 매체로 인식한 기존의 통신망과 비교해 볼때, 개인 휴대 통신의 가장 중요한 특성은 개인을 중심으로 한 통신의 구현을 그 목적으로 하여 개인의 이동성을 보장한다. 즉, 개인에게 전달되는 정보를 시간, 장소 그리고 휴대의 제약없이 즉시 그 개인에게 전달하는 것이다.

이러한 통신 서비스의 개인화는 기존의 전화기에 할당되었던 전화 번호를 개인 휴대 통신망에서는 가입자 개인에게 할당되어 관리되며, 이동성의 보장을 위해 통신망의 지능화가 요구된다. 즉, 통신망은 항상 모든 가입자의 현재 위치를 파악하고 있어야만 개인에 대한 통신 요구가 있을 때 어디로 전달해야 하는지를 알 수 있다. 통신망의 지능화를 위해 개인 통신 서비스망에서는 HLR(Home Location Register)와 VLR(Visitor Location Register)을 유지 관리해야 한다. HLR은 개인 통신 서비스 가입자에 대한 제반 정보를 반영구적으로 기록 유지하는 데이터 베이스로 가입자 관련 정보, 사용 단말 정보, 구독 서비스의 종류 및 요구사항, 현 위치 정보 등을 갖는다. 반면에, VLR은 현재 해당 VLR 지역내에 위치하고 있는 가입자 및 단말에 대한 정보를 임시적으로 간직하는 데이터 베이스이다.

HLR과 VLR은 일반적인 데이터 베이스 시스템과 비교할 때 다음과 같은 특성을 갖는다. 첫째, 각 개인에게 전달되어야 할 정보는 신속히 전달되어야 하므로 실시간 데이터 베이스 시스템이어야 한다. 둘째, 대부분의 실시간 데이터 질의는 전달해야 할 가입자의 현 위치를 찾는 질문으로 복잡한 데이터 구조를 갖지 않는다. 셋째, 통신의 기본 방법은 정보를 보내는 사람과 받는 사람간의 point-to-point 통신이므로 동시에 많은 수의 통신이 발생한다. 따라서, 여러개의 다수 질의가 동시에 처리되어야 한다. 이러한 대용량의 고속 데이터 베이스 시스템의 구현은 개인 통신 서비스의 지능망 구조에 기본적으로 필요한 사항으

로 일반적으로 사용하는 데이터 베이스 관리 시스템으로는 그 요구 사항을 충족시킬 수 없다. 따라서, 이 논문에서는 HLR과 VLR을 위한 실시간 데이터 베이스 시스템을 병렬 컴퓨터에 질의 및 보전 규칙(Integrity Constraint), 연역 법칙(deductive law)의 처리를 빠른 시간에 실행할 수 있는 주기억 장치 연역 데이터 베이스 시스템의 구현을 제시한다.

많은 양의 데이터를 처리해야 하는 데이터 베이스의 질의 처리 속도를 빠르게 하기 위해 여러가지의 병렬화 방법이 연구되었다. 병렬화된 대부분의 데이터 베이스 작업을 수월하게 실행할 수 있게 구성한 하드웨어의 구조적인 특성을 이용하여 데이터 베이스 작업을 빠르게 구현한다. 이러한 예로서 백엔드 데이터 베이스 시스템<sup>10</sup> RAP<sup>7</sup>, 시스토크 어레이<sup>8</sup> 그리고 벡터 프로세서 등이 있다.<sup>9</sup> 특수한 하드웨어적인 구조를 갖지 않는 범용 다중 프로세서 시스템에서도 데이터 베이스 작업을 병렬화하는 여러개의 알고리즘이 연구되었다.<sup>10,11</sup> 이들은 데이터 베이스에서 많이 사용되는 작업들인 선택(select)과 조인(join), 프로젝트(project) 등의 데이터 베이스 작업에 대한 병렬화 알고리즘으로, 질의 처리 과정에 내재되어 있는 병렬성에 대한 연구가 없었다.

이러한 이유는 전통적으로 데이터 베이스에 있는 릴레이션(relation)의 크기가 시스템의 주기억 장치(main memory)의 크기보다 컸기 때문에 데이터 베이스 병렬화는 릴레이션을 여러개의 부분(fragment)으로 나누어, 동일한 작업을 여러 부분들에 병렬로 실행하는 방법이 그 주종을 이룬다. 하지만, 컴퓨터 기술의 놀라운 발전으로 주기억 장치 소자의 가격이 하락되면서 현재의 컴퓨터 시스템에 있는 주기억 장치의 크기는 예전의 시스템에서 사용하는 2차 기억 장치(secondary memory)의 크기를 넘어가고 있다. 이러한 경향에 맞추어 디스크 접근에 대한 효율을 높였던 과거의 접근 방법에서, 대부분의 릴레이션을 빠른 주기억 장치에 저장하여 질의 처리를 실행하는 주기억 장치 데이터 베이스 시스템의 연구가 활발해졌다.<sup>12,13</sup> 또한, 컴퓨터 시스템의 대용량화에 맞추어 예전의 한 질의에 대한 실행 순서의 최적화에서 여러개의 질의들을 동시에 효율적으로 처리하기 위한 다수 질의 최적화 연구가 활발히 진행되고 있다.<sup>14,15,16</sup>

2000년대에 개인 통신 서비스가 실현될때는 많은 용량의 주기억 장치를 갖는 병렬 시스템의 사용이 보편화될 것으로 예상되므로 이 논문에서는 범용 대용량의 병렬 시스템을 기본 하드웨어로 설정하여

지능망의 연역 데이터 베이스 구현을 연구한다.

가입자들간의 통신 요구를 처리하기 위한 다수 질의 처리와 HLR과 VLR 데이터 베이스에 있는 데이터의 일관성을 유지하기 위한 보전 규칙들의 처리는 개인 통신 서비스의 원활한 실행을 위해 가능한 빠른 시간에 처리해야 한다. 이러한 다수의 질의 및 보전 규칙, 연역 법칙의 처리 순서는 중복되는 데이터 베이스 작업을 한번만 실행하여, 각각의 질의, 연역 법칙과 보전 규칙은 그 실행 결과를 공유한다. 이렇게 최적화된 실행 순서는 RETE 알고리즘<sup>16)</sup>의 변형인 관계 그래프로 나타내어 서로 연관이 없는 작업들은 동시에 처리하여 작업 단위에서의 병렬화를 얻을 수 있다.

## II. 연역 데이터 베이스

특수한 데이터 베이스의 응용 분야에 대한 요구가 늘어나면서 연역 데이터 베이스 시스템의 연구가 활발히 진행되어 왔다.<sup>45)</sup>

기존의 관계 데이터 베이스 시스템과 비교해 볼 때, 연역 데이터 베이스는 다음과 같은 특성을 갖는다. 첫째, 다양한 주제에 대한 규칙을 갖으므로 사용자의 질의에 쉽게 대처할 수 있다. 예를들면, 가족 구성원에 대한 규칙, VLSI 데이터 규칙, 지형적인 정보에 대한 규칙 등의 데이터에 대한 일반적인 규칙을 인텐션얼(intentional) 데이터 베이스로 구현한다. 둘째, 이러한 규칙등의 데이터에 대한 일반적인 규칙을 인텐션얼(intentional) 데이터 베이스로 구현한다. 둘째, 이러한 규칙들이 연역 데이터 베이스의 데이터로 존재하여 많은 양의 인텐션얼(intentional) 데이터를 쉽게 처리할 수 있다. 셋째, 인텐션얼 데이터 베이스의 사용으로 관계 데이터를 갖는 익스텐션얼(extentional) 데이터 베이스의 크기를 줄일 수 있다. 응용 분야에 대한 일반적인 규칙들을 나타내고 처리할 수 있으므로 연역 데이터 베이스는 개인 통신 서비스망에서 요구되는 지능망의 구현에 적절한 시스템이다. 이러한 연역 데이터 베이스 시스템의 구현에 있어서 가장 중요한 과제는 인텐션얼 데이터 연역 법칙과 보전 규칙을 효율적으로 처리하는 방법의 연구로 이 논문에서는 연역 데이터 베이스의 질의 처리, 연역 법칙 처리, 그리고 보전 규칙의 효율적이고 빠른 처리를 위한 병렬화 방법을 제시한다.

한정 연역 데이터 베이스(definite deductive database)<sup>46)</sup>는 데이터 상수 집합과 다음과 같은

일차절(first-order clause)의 집합으로 정의한다.

$$P_1 \wedge P_2 \wedge \dots \wedge P_k \rightarrow R_q.$$

술어  $P_i$ 들의 합동(conjunction)은 절의 왼편이라고 하며 술어  $R_q$ 들의 분리(disjunction)는 절의 오른편이라고 한다. 상수  $k$ 와  $q$ 의 값에 따라 사실(fact), 보전 규칙 그리고 연역 법칙으로 구분된다. 여기서는<sup>47)</sup>의 정의에 따라 절의 종류들을 간단히 설명한다.

(1) Type 1( $k=0, q=1$ ) :  $\rightarrow P(t_1, \dots, t_m)$

만약 모든 인수  $t_i$ 가 상수면,  $P$ 는 데이터 베이스의 사실(fact)을 나타내며 관계 데이터 베이스의 릴레이션을 구성하는 한 투풀에 해당된다.  $t_i$ 중 변수가 있으면  $P$ 는 보편적인 사실을 나타낸다. 예를 들면,

$$\rightarrow \text{ancestor}(\text{george}, X)$$

는 "george"가 모든 사람의 조상이라는 보편적인 사실을 나타낸다.

(2) Type 2( $k=1, q=0$ ) :  $P(t_1, \dots, t_m) \rightarrow$

만약 모든 인수  $t_i$ 가 상수면, 부정의 사실을 나타내며, 변수 인수를 갖으면, 보전 규칙 또는 존재하지 않는 값(null value)으로 간주한다.

(3) Type 3( $k>1, q=0$ ) :  $P_1 \wedge P_2 \wedge \dots \wedge P_k \rightarrow$

새로운 데이터가 첨가될 때 항상 만족해야 하는 보전 규칙이다. 예를들면, 어떤 사람도 어머니와 아버지가 동시에 될 수 없으므로 가족 관계를 나타내는 데이터 베이스에는 다음과 같은 보전 규칙을 갖는다.

$$\text{father}(X, Y) \wedge \text{mother}(X, Y) \rightarrow$$

(4) Type 4( $K \geq 1, q=1$ ) :  $P_1 \wedge P_2 \wedge \dots \wedge P_K \rightarrow R_1$

보전규칙 또는 술어  $P_1, \dots, P_K$ 를 사용한 술어  $R_1$ 을 정의하는 연역 법칙을 나타내는 절이다.

마지막으로  $k=0, q=0$ 인 절(empty clause)은 연역 데이터 베이스에 포함하지 않는다.

보전 규칙을 데이터 베이스에 새로운 사실을 첨가 또는 삭제가 데이터 베이스의 내용과 일치하는지를 검사하는 규칙으로 데이터 베이스가 변화될 때마다 검사해야 한다. 예를들면, 개인 통신 서비스에서 이동 가입자가 동시에 동일한 개인 전화번호로 한 장소 이상 등록되어 있으면 안된다. 반면에 데이터 베이스의 내용이 갖는 일반적인 사실을 표현하는 연역 법칙은 많은 양의 익스텐션얼 데이터 베이스를 사용하지 않고 사실을 유추해낼 수 있다. 이러한 연역 법칙에 대한 절의 변수와 상수를 인수로 갖는 술어들의

합동으로 표시된다. 보전 규칙, 연역 법칙 그리고 질의가 모두 술어들의 합동을 갖으므로 이러한 합동의 효율적이고 빠른 처리 방법의 개발은 연역 데이터 베이스에서 대단히 중요한 과제이다. 또한 연역 데이터 베이스 시스템에서는 동시에 여러개의 보전 규칙, 연역 법칙 그리고 질의들을 동시에 처리하므로 이들 합동들의 처리에는 상당한 부분이 중복된다. 따라서, 중복되는 작업 결과의 공유는 그들의 처리 속도를 빠르고 효율적으로 만든다.

### III. 릴레이션의 분할

데이터 베이스 작업을 병렬화하기 위해 전통적으로 많이 사용된 방법인 릴레이션을 여러개의 부분으로 나누어 각 부분을 병렬로 처리한다. 따라서, 질의 처리 과정은 두 단계의 병렬화를 통하여 처리 속도를 빠르게 할 수 있다. 제안된 방법은 이러한 병렬성을 살리기 위해 테이블형의 릴레이션을 수직과 수평으로 나누어 각 부분들을 주기억 장치에 저장한다. 또한, 메모리의 사용량과 데이터 처리량을 최소화하기 위해 각 데이터 베이스 작업의 결과 릴레이션을 두가지 벡터 표시법을 사용하여 저장하고 접근한다. 따라서, 결과 릴레이션의 생성과 유지가 필요없이 데이터 베이스에 있는 기본 릴레이션만을 사용하여, 질의 처리에 필요한 최소한의 데이터를 접근하여 데이터 베이스 작업을 수행한다.

릴레이션 R의 어트리뷰트가  $a_1, a_2, \dots, a_v$  라면, R은 어트리뷰트 단위로 수직 분할하여 각각의 부분을  $R_1, R_2, R_3, \dots, R_v$ 로 나타낸다. 또한, 수직으로 분할된 부분  $R_j$ 를 같은 크기로 수평 분할하여, 수직과 수평으로 분할된 부분을  $R(i, j)$ 라고 정의한다.

즉, 한 릴레이션은 행렬로 간주되어  $R(i, j)$ 는  $i$  번째 수평 분할된 부분에서  $j$  번째 어트리뷰트에 대한 데이터를 가리킨다. 각각의 수평 분할된 부분은 마지막 부분을 제외하고는 동일한 수의 데이터를 갖는다. 수평 분할이나 수직 분할이 릴레이션의 데이터를 복사하지 않으므로 릴레이션은 물리적으로 여러개의 부분으로 나누어 서로 다른 메모리 모듈에 저장되지만 논리적으로는 분할되기 전과 동일한 릴레이션 형태를 유지한다. 릴레이션을 수직 분할하는 이유는 오직 데이터의 접근시 메모리 충돌이나 데이터 전송량을 줄이기 위한 것이므로, 모든 데이터 작업은 논리적으로 수평 분할된 부분을 기준으로 실행된다. 이 논문에서 제안된 방법은 릴레이션의 수평 분할된 부분  $R-i=R(i, j)$ 에 대한 상태를 나타내기

위해 비트 벡터 표시법과 비교 벡터 표시법을 사용한다. 각 표시법은 헤더를 가지며 이 헤더에는 표시하는 릴레이션 R과 그 수평 분할 인덱스  $i$ , 그리고 분할된 부분의 투폴수를 저장한다.

#### (1) 비트 벡터 표시법(bit-vector notation)

수평 분할된 부분 R-i에 있는 투폴들을  $\langle t_1, t_2, \dots, t_k \rangle$ 라고 하면, 이들 투폴에 해당하는 비트 벡터는

$$[R-i \mid x_1, x_2, \dots, x_k], x_k = \begin{cases} 0 & \text{if present} \\ 1 & \text{otherwise} \end{cases}$$

로 나타낸다. 예를들면, 릴레이션 R에 대한 select  $r_1 = 'a'$  작업을 부분 R-i에 실행하면,  $R(i, 1)$  부분을 접근하여 선택된 결과를 다음과 같이 비트 벡터로 나타낼 수 있다.

부분 R(i, 1)	Select R1= $a$
a	true
b	false
c	false
d	true
e	false

이 부분에 대한 선택 작업의 결과는 성공(true)일때, '1'을, 실패일때 '0'을 사용하여 다음과 같이 비트 벡터로 나타낸다.

$$[R-i \mid 1, 0, 0, 1, 0]$$

#### (2) 비교 벡터 표시법

비트 벡터로 표시된 부분에서 '1'을 갖는 투폴을 그 부분의 첫 투폴로부터의 오셋(offset)으로 나타내며, 이를 투폴 번호라고 정의한다. 위의 비트 벡터  $[R-i \mid 1, 0, 0, 1, 0]$ 에서 투폴 번호 1과 4가 '1'을 가지므로 이 비트 벡터를 비교 벡터로 표시하면 다음과 같다.

$$[R-i \mid 1, 4]$$

비교 벡터는 여러개의 비교 벡터를 연결하여 한 비교 벡터를 만들 수 있다. 이렇게 연결된 비교 벡터는 조인 작업 결과를 나타내는 중간 릴레이션(intermediate relation)을 나타낸다. 비교 벡터

$$[R-i \mid f_1, f_2, \dots, f_m] [Q-j \mid h_1, h_2, \dots, h_m]$$

은 부분 R-i와 부분 Q-j를 조인한 결과를 나타내며, fk는 부분 R-i의 튜플 번호를, hx는 부분 Q-j의 튜플 번호를 각각 나타낸다. 이 조인 결과는 m개의 튜플을 가지며, 그 첫째 튜플은 R-i 부분은 f1 번째 튜플과 Q-j 부분의 h1번째 튜플을 결합한 튜플이다. 다음은 R(i, 1)와 Q(j, 2) 부분들을 조인 조건 join R1=Q2로 조인 작업한 결과를 비교 벡터로 나타낸다.

R(i, 1)	Q(j, 2)
b	a
a	c
	a

비교 벡터 : [R-i | 2, 2] [Q-j | 1, 3]

이 비교 벡터는 조인 작업의 결과 릴레이션을 나타내며, 그 첫 튜플은 R-i 부분의 두번째 튜플과 Q-j 부분의 첫번째 튜플이 연결된 튜플을 나타낸다. 또한, 결과 릴레이션의 두번째 튜플은 R-i 부분의 두번째 튜플과 Q-j 부분의 세번째 튜플이 연결된 튜플이다.

데이터 베이스 작업을 실행하여 구한 결과 릴레이션은 항상 비교 벡터로 표시하지만, 작업을 실행할 때는 비트 벡터로 변환되어 실행된 후, 다시 비교 벡터로 재변환된다. 이런 과정을 거치는 이유는 다음절에 기술한다. 비교 벡터로 결과 릴레이션을 나타내는 방법은 주기억 장치 데이터 베이스 시스템<sup>(213)</sup>에서 데이터 베이스의 모든 릴레이션에 있는 튜플에 유일한 번호를 붙여서 결과 릴레이션을 나타내는 방법과 유사하지만, 모든 튜플에 대해 유일한 번호를 붙여 관리할 필요가 없다.

#### IV. 술어 합동들에 대한 관계 그래프의 병렬 처리

이절에서는 보전 규칙과 연역 법칙, 질의에 있는 술어 합동들의 병렬처리 방법을 기술한다. 다수의 합동을 동시에 처리하기 위해 포함하는 작업을 고려한 실행 순서를 나타내는 관계 그래프를 만들어 서로 연관이 없는 작업을 병렬로 처리하게 된다. 관계 그래프는 프로덕션(production) 시스템에서 프로덕션 법칙들을 동시에 처리하기 위해 제안된 레테(RETE) 알고리즘의 변형된 형태로 이에 대한 연구는<sup>(214)</sup>에 제시되었다. 관계 그래프는 다수의 합동들에서 중복된 선택(select)과 조인(join) 작업을

찾아내어, 이들을 오직 한번만 실행하면서 다수의 합동들을 동시에 처리하는 실행 그래프를 만든다. 레테 알고리즘은 프로덕션 법칙들에 있는 동일한 작업만을 고려한 반면에, 관계 그래프는 동일한 작업 뿐만 아니라 포함되는 작업도 고려한다. 예를들면, 술어 p(X, Y, a)는 P(X, b, a)를 포함한다. 따라서, 술어 p(X, Y, a)에서 구한 결과 튜플들에서 Y='b'의 선택 작업을 하면 술어 p(X, b, a)의 변수 X의 값을 구할 수 있다. 만약, 릴레이션 p의 튜플수가 n이고 p(X, Y, a)의 튜플수가 m이라면 (m<n), 두 술어가 포함된 관계를 고려하지 않고 릴레이션 p에 각각의 선택 작업을 실행했을 때의 처리는 0(2n)이지만, 포함된 관계를 고려했을 때는 0(n+m)이 된다. 따라서, 포함된 관계를 고려했을 때, 0(n-m)만큼의 처리를 줄일 수 있다.

#### 4.1 관계 그래프

다음의 조건이 만족됐을 때, 술어 p가 술어 q보다 더 평범한 술어(more general predicate : MGP)라고 정의한다.

- (1) 술어 P와 q가 동일한 릴레이션의 이름이고
- (2) 술어 p에 있는 인수들이 술어 q의 인수들보다 더 평범하다.

위에서 평범하다는 뜻은 변수 인수가 상수 인수보다 더 평범하다는 뜻으로, 술어 p와 q가 동일화(unification) 되었을 때, 술어 p의 모든 상수 인수에 대해 술어 q에 대응되는 인수는 같은 상수여야 하며, 술어 p의 적어도 한 변수 인수는 술어 q의 상수 인수와 동일화 되어야 한다. 즉, 술어 p(X, Y, a)와 술어 p(Z, b, a)는 X/Z와 Y/b의 치환으로 동일화가 성공하며, 이때 술어 p(X, Y, a)는 술어 p(Z, b, a)의 MGP이다.

마찬가지로, 두개의 술어 합동(conjunction) A와 B에서, 다음의 조건이 만족되면 합동 A는 B보다 더 평범한 합동(more general formula : MGF)이라고 정의한다.

- (1) 합동 A와 합동 B의 술어들간에는 1-대-1 대응 관계 B=f(A)가 존재한다.
- (2) 합동 A의 조인 조건과 합동 B의 조건이 동일하다.
- (3) 합동 A의 모든 술어 p에 대해, p는 f(p)와 동일하거나 더 평범하다.

예를들면, 합동 A=p(X, Y, Z) ∧ q(a, Y, Z)와 합동 B=p(b, X, Y) ∧ q(a, X, Y)에서 합동 A가 합동 B의 MGF가 된다. 그 이유는 합동 A의 술어와

합동 B의 술어들이 각각 같은 릴레이션을 나타내며, 합동 A의 술어들이 합동 B의 술어 보다 더 평범하다. 또한, 합동 A와 B의 조인 조건은 각각  $p2=q2$ 와  $p3=q3$ 로 동일하다. 결과적으로, MGF는 MGF 집합에 있는 술어 합동의 처리를 위한 조인 작업을 선택 작업으로 대체한다.

다음의 합동들에서 MGP와 MGF들을 구해보자.

합동들 :

- Q1 :  $p(X, Y, Z) \wedge q(a, Y, Z)$
- Q2 :  $p(a, X, Y) \wedge q(a, X, Y)$
- Q3 :  $p(X, Y, c) \wedge q(a, Y, c) \wedge r(X, Y)$
- Q4 :  $q(X, Y, Z) \wedge r(d, Y)$

동일한 술어 :  $q(a, Y, Z)=q(a, X, Y)$

MGP	MGP 집합
$p(X, Y, Z)$	$\{ p(a, Y, Z), p(X, Y, c) \}$
$q(X, Y, Z)$	$\{ q(a, Y, Z), q(a, X, Y, c) \}$
$q(a, X, Y)$	$\{ q(a, X, c) \}$
$r(X, Y)$	$\{ r(d, Y) \}$

MGF  
 $p(X, Y, Z) \wedge q(a, Y, Z)$     MGF 집합  
 $\{ p(a, Y, Z) \wedge q(a, Y, Z) \}$

술어에 있는 변수는 한 합동에서만 그 의미가 있으므로, MGP와 MGF 집합에 있는 술어에는 각각의 MGP나 MGF의 변수 이름으로 동일화하여 사용하였다.

관계 그래프는 선택 작업을 나타내는 선택 노드와 조인 작업을 나타내는 조인 노드, 각 합동에 대한 합동 노드로 구성된다. 데이터 베이스의 작업 결과를 여러개의 합동이 공유하기 위해 프로젝트(project) 작업은 작업 노드로 고려하지 않고, 합동 노드에서 각 합동에서 필요한 어트리뷰트에 대해 프로젝트 작업을 실행하여 최종적인 답을 구한다. 이 그래프는 가장 윗 단계에 데이터 베이스의 기본 릴레이션에서 부터 가장 평범한 MGP를 갖는 선택 노드를 만들고, 이들의 MGP 집합에 있는 술어들을 평범한 순으로 각각 선택 노드를 만든다. 이때, MGF 집합에 있는 원소 합동의 술어에 대한 선택 노드를 만들지

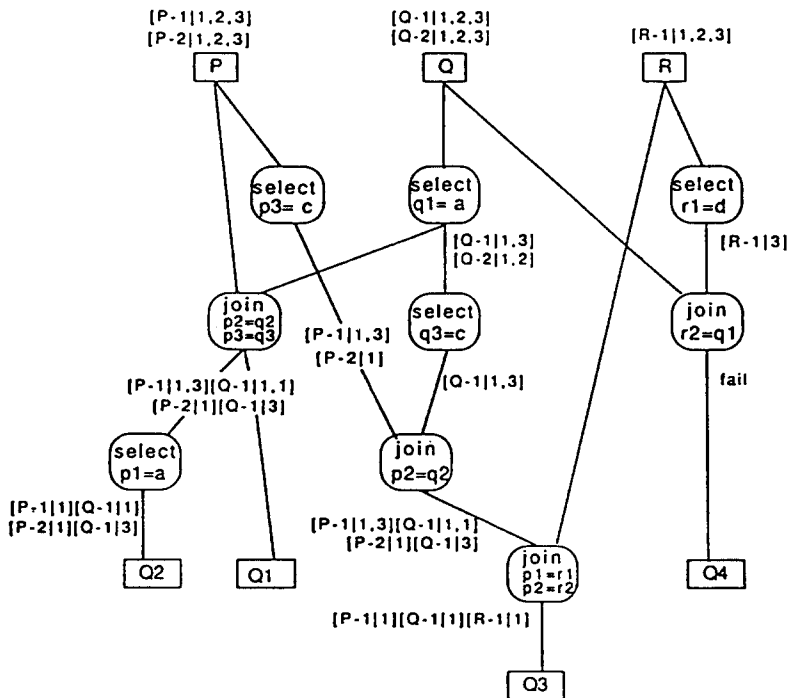


그림 2. 예제 데이터 베이스

않는다. 이렇게 모든 MGP 집합에 있는 술어들에 대한 선택 노드를 만든 후, MGF에 대한 조인 노드를 만든다. 계속해서, MGF 집합에 있는 원소들에 대한 선택 노드를 만든다. 이 과정이 끝나면, 모든 중복되는 작업 노드를 만들었으므로 MGP와 MGF 집합에 없는 나머지 술어들에 대한 작업 노드를 만든다. 마지막으로, 각각의 합동 노드를 만들면, 여러개의 합동들을 처리하는 실행 순서를 갖는 관계 그래프가 완성된다.

그림1은 위에서 구한 MGP와 MGF를 고려한 관계 그래프를 그림2에 있는 예제 데이터 베이스에서 3 절에서 제시한 벡터 표시법을 사용하여 처리한 결과를 보여준다. 각 작업 노드에서의 작업 조건은 술어의 인수 위치를 사용하여 표시한다. 예를들면,  $p(a, X, Y)$ 에 해당하는 선택 노드에서 작업 조건은  $select\ p1='a'$ 로 표시되었다. 또한, 조인 조건  $p(a, Y, Z) \wedge q(a, Y, Z)$ 에 대한 조인 노드는  $join\ p2=q2, p3=q3$ 로 표시되었다.

관계 그래프의 실행은 가장 윗 단계에 있는 작업 노드들을 병렬로 실행하고, 나머지 단계의 노드는 그 노드의 윗 단계에 연결된 모든 노드의 실행이 끝나는데로 실행이 가능하다. 따라서, 노드가 위치한 단계에 관계없이 실행이 가능한 노드는 모두 병렬로 처리한다. 이러한 방법으로 모든 작업 노드의 실행이 끝나면, 각각의 합동 노드에서 그 결과를 구하게 된다.

릴레이션 P	릴레이션 Q	릴레이션 R
<P-1 부분>	<Q-1 부분>	<R-1 부분>
$p(a, b, c)$	$q(a, b, c)$	$r(a, b)$
$p(e, d, f)$	$q(e, d, c)$	$r(a, b)$
$p(f, b, c)$	$q(a, d, c)$	$r(d, c)$
<P-2 부분>	<Q-2 부분>	
$p(a, d, c)$	$q(a, c, f)$	
$p(d, e, f)$	$q(a, b, d)$	
$p(g, e, f)$	$q(e, d, c)$	

기존의 데이터 베이스 시스템에서는 데이터 베이스 작업의 성능을 높이기 위해 기본 릴레이션을 여러 가지 방법을 적용하여 일정한 형태로 선행 처리한 후, 그 결과를 사용하여 데이터 작업을 효율적으로 실행한다. 예를들면, 선택 작업은 선택 작업이 많이

실행되는 열(column)에 있는 데이터에 여러가지 인덱싱 기법<sup>(12)</sup>을 적용하여 데이터의 인덱스 정보를 저장한다. 이렇게 저장된 정보는 선택 작업이 실행될 때, 모든 투표를 검색하지 않고도 그 결과를 구할 수 있다.

조인 작업은 머지소트(merge-sort) 또는 해싱(hashing) 하여 만든 변형된 릴레이션을 사용하여 조인 작업의 속도를 향상시킨다<sup>(7)</sup>. 이러한 방법은 한개의 프로세서를 갖는 시스템에서 제한된 프로세싱 능력의 효율적인 사용을 위해 유용한 방법이지만, 여러개의 프로세서를 갖는 병렬 시스템에서는 처리 능력의 한계보다는 여러개의 프로세서가 동시에 처리할 수 있는 환경을 만들어 주는 것이 더욱 효과적이다.

이러한 관점에서 볼때, 위의 선행 처리 방법은 주기억 장치 데이터 베이스 시스템에서 다음과 같은 단점을 갖는다. 첫째, 변형된 형태의 데이터(인덱스 또는 릴레이션)를 주기억 장치에 저장해야 하므로 메모리 공간을 사용한다. 둘째, 기본 릴레이션의 모든 투표를 검색해야 하므로, 변형 작업이 데이터 베이스 작업 실행 바로전에 동적으로 수행된다면, 각 데이터 베이스 작업은 모든 인수 릴레이션 부분에 대한 선행 작업이 끝날때까지 기다려야 하는 동기화가 필요하다. 이러한 이유로 제안된 방법은 기본 릴레이션을 그대로 사용하는 선택 작업과 조인 작업(nested-loop) 을 사용한다.

## 4. 2 선택 작업(Selection)

선택 작업은 한 릴레이션을 인수로 하는 일항 작업이므로, 관계 그래프에서는 위로 오직 한 노드와 연결되어 있다. 선택 노드는 윗노드에서부터 내려오는 비교 벡터에 대해 자신의 선택 조건에 따라 선택 작업을 실행한다. 작업을 실행하기 전에 비교 벡터로 표시된 인수를 항상 비트 벡터로 변환한다. 변환된 비트 벡터에서 오직 '1'을 갖는 투표들만을 접근하여 선택 작업을 실행하여, 그 결과가 실패하면 해당되는 원소를 0으로 만든다. 예를들면, 그림2의 선택 노드  $select\ p1=a$ 에서 첫 부분에 대한 실행은 선택 조건의 릴레이션 p에 관한 비교벡터를 다음과 같이 비트 벡터로 변환한다.

$$[P-1 \parallel 1, 3] [Q-1 \parallel 1, 1] \Rightarrow P \text{의 비트 벡터} : [P-1 \parallel 1, 0, 1]$$

이 변환은 비교 벡터의 헤더에 그 부분의 총 투표

수가 저장되어 있으므로 비교 벡터에 존재하지 않는 투폴 번호를 모두 0으로 만든다. 따라서, 인수 P-1의 수평 부분인 1과 선택 작업  $p1=a$ 의 어트리뷰트 1을 사용하여 릴레이션 부분  $p(1,1)$ 의 첫번째와 세번째 데이터만을 접근하여 선택 조건과 비교하면 된다. 그 결과는 세번째 투폴이 'a'가 아니므로  $[P-1 | 1, 0, 0]$ 이 된다. 선택 작업이 완료되면, 결과 비트 벡터를 다시 비교 벡터로 변환하여  $[P-1 | 1]$ 이 되므로, 인수의 P-1 부분의 세번째 투폴과 하나의 투폴을 이루는 Q-1 부분의 비교 벡터에 있는 두번째 1을 제거하면, 이 선택 노드의 결과는  $[P-1 | 1] [Q-1 | 1]$ 이 된다. 비교 벡터에는 같은 투폴 번호가 여러번 사용될 수 있으므로, 비트 벡터의 변환으로 선택 작업을 위해 접근해야 하는 데이터의 양을 줄일 수 있다. 예를들면, 만약 위의 인수 비교 벡터에  $select\ q1='a'$ 를 실행한다면, Q-1 부분의 첫번째 투폴을 두번 접근하여 비교해야 한다. 기존의 중간 릴레이션을 만드는 방법은 중복된 투폴이 생성되므로 중복된 수만큼의 데이터 접근과 비교가 필요하다.

이렇게 실행된 선택 작업의 결과가 아무런 투폴을 갖지 않으면, 그에 대한 결과를 아래에 있는 노드로 전달하지 않는다. 그림2에서 선택 노드  $select\ q3='c'$ 는 Q-2 부분에서 선택 조건을 만족하는 투폴이 없으므로, 그 아래에 연결된 작업 노드로 Q-2 부분에 대한 결과를 전달하지 않았다.

### 4.3 조인 작업(join)

조인 작업은 두개의 인수를 갖는 작업으로, 관계 그래프에서 두개의 윗 노드로부터 각각의 인수를 받는다. 만약, 위에 있는 두 노드에서 내려오는 결과들이 각각 n개와 m개라고 하면,  $n \times m$  번의 조인 작업을 위한 프로세스를 생성하여 네스티드 루프(nested loop) 조인 알고리즘<sup>7)</sup>으로 실행한다. 즉, 한 윗노드에서 내려오는 각각의 결과를 다른 윗 노드에서 내려오는 모든 결과와 조인 작업을 한다. 조인 작업이 프로세스에 할당되면, 각각의 인수 비교 벡터에서 조인 조건에 해당하는 비교 벡터를 비트 벡터로 변환한다. 그림2에서 조인 노드  $join\ p2=q2$ 에서의 조인 작업 실행시, 각 인수 비교 벡터들을 아래와 같이 비트 벡터로 변환한다.

인수 비교 벡터    →    인수 비트 벡터

$[P-1 | 1, 3]$          $[P-1 | 1, 0, 1]$   
 $[Q-1 | 1, 3]$          $[Q-1 | 1, 0, 1]$

이러한 비트 벡터는 다음과 같이 각각의 투폴 번호를 인덱스로한 조인 행렬을 정의한다.

$1\ 0\ 1 \leftarrow Q-1\ 부분의\ 인수비트\ 벡터$

P-1부분	1	2	3
	1	1	
	0	2	
	1	3	

이제 각각의 행의 비트 벡터가 1인 행과 열의 비트 벡터가 1인 원소들만을 접근하여, 그들이 조인 조건을 만족하는지를 검사한다. 위의 행렬에서는 (1, 1), (1, 3), (3, 1)과 (3, 3)에 해당하는 P-1과 Q-1 부분의 데이터를 비교한다. 따라서, 조인의 결과는 다음과 같이 행렬의 원소로 나타낼 수 있다.

$1\ 0\ 1 \leftarrow Q-1\ 부분의\ 인수비트\ 벡터$

P-1부분	1	2	3		
	1	1	0	0	
	0	2	1	0	0
	1	3	1	0	0

이 행렬은 다음과 같이 다시 두개의 인수 비교 벡터를 연결한 비교 벡터로 변환한다.

$[P-1 | 1, 3] [Q-1 | 1, 1]$

만약, 인수 비교 벡터에 조인 조건과 관계없는 다른 비교 벡터가 연결되었으면, 이들도 조인 결과에 따라 변환된다. 예를들면, 만약 Q-1에 대한 인수 비교 벡터가  $[Q-1 | 1, 3] [R-2 | 1, 2]$ 였다면, 이 조인의 결과는 다음과 같다.

$[P-1 | 1, 3] [Q-1 | 1, 1] [R-2 | 1, 1]$

위의 비교 벡터에서 Q-1 부분의 3번째 투폴이 제외되었고 첫번째 투폴이 두번 반복되므로, Q-1에서 제거된 원소에 해당하는 R-2의 두번째 원소 2를 제거하고, 새로 생성된 원소에 해당하는 R-2의 첫번째 원소를 한번 더 생성하였다.

중간 릴레이션에는 조인 작업의 결과로 기본 릴레이션에 있는 투폴이 여러번 복사(duplicates)되어 사용된다.<sup>12)</sup>에서는 복사된 투폴을 제거하기 위해 해싱(hashing) 기법을 사용하였지만, 제안된 조인 방법은 선택 작업에서와 같이 조인 작업을 실행하기 전에 인수 비교 벡터를 비트 벡터로 변환하므로 복사된 투폴들을 모두 접근하여 비교하지 않고, 기본 릴레이션에 존재하는 투폴만을 검색한다. 따라서,



기존의 방법에 비해<sup>(213)</sup> 실제로 접근하여 비교하는 데이터의 양을 줄일 수 있다. 예를들면, 다음의 두 인수 벡터를 조인 조건  $p2=q2$ 로 실행하면,

$$[P-1 \mid 1, 3] \rightarrow [P-1 \mid 1, 0]$$

$$[Q-1 \mid 1, 1, 1, 1, 2] [R-2 \mid 1, 2, 3, 5, 1] \rightarrow [Q-1, 1, 0]$$

결과적으로 비교 벡터만을 사용했을 때 릴레이션 Q의 첫 투풀이 4번 복사되었으므로 동일한 투풀을 4번 접근하여 비교하지만, 비트 벡터로의 변환은 첫번째 투풀을 오직 한번만 접근하여 처리하면 되므로 불필요한 데이터 접근과 처리를 줄일 수 있다.

## V. 결 론

개인 통신 서비스의 통신망은 복잡한 접근 방법 및 여러 형태의 통신 방법에 대한 제어 지식, 다양한 통신 요금 계산 및 그 정산 방법, 그리고 망 관리에 관한 지식 등 여러가지 지식을 관리, 처리할 수 있는 원활하고 신속한 망 관리가 요구된다. 특히 개인 이동성의 보장을 위해 개인의 위치를 빠른 시간안에 추적해야 하며, 변화된 개인의 위치를 신속히 수정하여야 한다.

미래의 개인 통신 서비스가 될 PCN은 모든 개개인 이 가입자가 된다고 가정할 수 있으며 많은 수의 가입자들간에 동시에 발생하는 많은 수의 통신 요청을 신속히 처리하기 위해서는 대용량의 병렬 시스템에서 접근 속도가 빠른 주기억 장치 데이터 베이스로 구현하여야 한다.

이 논문에서는 개인 통신 서비스(PCN)에서의 지능망 구현을 위한 주기억 장치 연역 데이터 베이스의 병렬 처리 방법을 기술하였다. 제안된 방법은 개인 통신 서비스에서 동시 다발적으로 생기는 이동 가입자에 대한 정보 검색과 변화를 빠른 시간안에 처리하기 위해 데이터 베이스 작업들중에 중복되는 작업을 찾아내어, 이들의 실행을 최적화한 관계 그래프의 실행 순서에 따라 합동들을 동시에 처리한다.

이때, 동일한 데이터 작업을 한번 실행하여 그 결과를 공유하고, 내포하는 작업은 인수 릴레이션의 크기를 줄이는 순으로 실행 순서를 정하게 된다. 이렇게 만들어진 관계 그래프는 정보화 시대에 보편적으로 사용될 병렬 컴퓨터의 주기억 장치에서 서로 연관이 없는 작업 노드들을 병렬로 실행된다. 작업 노드는 수평으로 분할된 릴레이션 부분들에 병렬로

실행되므로 슬어들의 합동 처리 과정에서 2단계의 병렬화를 얻을 수 있다. 또한, 제안된 두가지 벡터 표시법은 프로세서간의 데이터 전송량과 검색 데이터의 양을 최소화 할 수 있다.

## References

1. Koshiyama, K., Hattori, T., Sekibichi, H. and Kawasaki, R., "Advanced personal communication system," Proc. VTC '90, pp 161-166.
2. 최두환, "Personal Communication Service, 무엇이 며 어떻게 할 것인가?," Telecommunication Review, Vol. 2, No 5, 1992.
3. Russel, J. E., "The U.S. evolution Towards personal communications in the 90's", Proc. DCRC '90.
4. Gallaire, H., Minkser, J. and Nicolas, J. M., "Logic and database, a deductive approach," Computing Surveys Vol. 16, 1984.
5. Kellogy, C., "From data management to knowledge management," Computer, Jan. 1986.
6. Masryanski, F. J., "Backend Database Systems," ACM Computing Survey, March, 1980.
7. Stewart, A. and at al., "RAP. 2- An Associative Processor for Databases and Its Applications," IEEE Trans. on Computers, June. 1979.
8. Kung, H. T. and Lehman, P. L., "Systolic(VLSI) Arrays of Relational Database operations," Proc. ACM SIGMOD, 1980.
9. Shun' ichi, T and et al, "A Relational Database System Architecture Based on a Vector Processing Method," Conf. on Data Engineering, Feb. 1987.
10. Bitton, D., and et al, "Parallel Algorithms for the Execution of Relational Database Operations," ACM Trans. on Database Systems, Vol 8, Sept, 1983.
11. Qadah, G. Z. and Irani, K. B., "The Join Algorithms on a Shared-Memory Multiprocessor Database Machine," IEEE Trans. on Software Engineering, Vol. 14, Nove. 1988.
12. Lehman, T. J. and Carey, M. J., "Query Processing in Main Memory Database Management Systems," Conf. on SIGMOD, 1986.
13. Bitton, D., Hanrahan, M. B., and Turbyfill, C.,

- “Performance of Complex Queries in Main Memory Database Systems,” 3rd Int. Conf. on Data Engineering, Feb. 1987.
14. Park, J. and Segev, A., “Using Common Sub-expressions to Optimize Multiple Queries,” Int. Conf. on Data Engineering, Feb. 1988.
  15. Sellis, T., “Multiple-Query Optimization,” ACM Trans, on Database Systems, Vol. 13, March, 1988.
  16. Forgy, C., “RETE : A Fast Algorithm for the Many Pattern / Many Object Pattern Matching Problem,” Artificial Instelligence 19, Sept. 1982.
  17. Sheu, P. C. and Lee, W. S., “Efficient Processing of Integrity Constraints in Deductive Databases,” Future Generation Computer System, Vol 3, 1988.
-