

## LAN을 이용한 다중 서버를 가진 데이터베이스 시스템을 위한 질의 스케줄링 방법의 개발

正會員 施鍾翊\*, 朴宗熹\*

### A Query Scheduling Method for Multi-Server Database Systems in LAN

Jong Ik Si\*, Jong Hee Park\* Regular Members

#### 要 約

개인용 데이터베이스의 사용자 환경을 그대로 유지하면서 다중의 사용자가 성능을 저하시키지 않고 데이터를 공유할 수 있는 시스템을 개발한다. 개인용 데이터베이스를 사용하던 기업이 확장하면서 점차 빠른 처리속도를 요구하게 되고 다른 사용자들과의 데이터 공유가 긴급해짐에 따라 개인용 데이터베이스 관리시스템 (DBMS)에서 다중의 사용자 데이터베이스 시스템으로의 변경이 요구된다. 아무리 기능이 향상된 새로운 시스템으로의 이행이라도 기존의 사용자환경을 바꾸어야 한다면 순조롭지 못하다. 더구나 중소기업의 기업에서는 기존의 사용자환경의 유지는 필수적이라 할 수 있다. 이러한 필요성에 부응하여 사용중인 개인용 데이터베이스의 사용자 환경을 유지하면서, 다사용자환경을 제공함과 동시에 성능저하도 막을 수 있는 실용적인 분산 데이터베이스 시스템을 개발한다.

본 시스템의 개요를 보면, 근거리 통신망(LAN)을 통해 연결된 여러개의 개인용 DBMS들이 복수개의 서버로 작동하고, 통합 질의관리자는 이러한 서버들을 다수의 사용자가 효율적으로 사용할 수 있게 한다. 시스템의 핵심 요소인 통합 질의 처리방법은 LAN상의 다중 사용자의 질의들을 받아서 그것들의 예상 서비스 시간 (및 질의 유형)에 따라 다중의 서버를 위한 최적의 스케줄링을 행한다. 본 논문에서는, 다양한 큐잉 방법에 기초한 여러가지 질의 스케줄링 방법들을 해석적으로 비교하고, 이를 기초로 다중 사용자를 위한 질의처리 시스템을 구현하여 실제적인 성능을 평가한다.

연구결과로, 해석적 분석 결과와 구현된 시스템의 모의 실험 결과가 유사한 경향을 나타낸다는 사실을 알 수 있고 이러한 모의 실험 결과는 기존의 개인용 데이터베이스 시스템에서 다중의 사용자 데이터베이스 시스템으로 이행해 나가는 효과적인 방법을 제시해 준다. 또한 주어진 요구 사항을 충족하는 데 필요한 서버의 개수등의 시스템 사양을 모의 실험 결과로부터 산출해 낼 수 있고, 해석적 방법으로 분석이 용이하지 못한 본 시스템에 대한 수학적 모델의 이론적 성능을 추정할 수 있다.

\*경북대학교 전자공학과  
Department of Electrical Eng. Kyungpook  
National University  
論文番號 : 95045-0204  
接受日字 : 1995年 2月 4日

## ABSTRACT

A LAN-based and multi-user DBMS is developed which provides a painless transition to higher performances needed by initially-small enterprises as they grow. Such performance enhancement is achieved by using a cluster of single-user DBMS systems interconnected through LAN. If the current DB user environment cannot be maintained, migration to a new system would not be smooth however better performance a new DB system has over the current system. In this light, our system design objective focuses on immunity of the database user to the usual difficulties resulting from such an upgrade. As a consequence the database users can migrate to an enhanced database environment without losing their existing database practices.

The major components of the system are the global query processor controlling multiple database servers and multiple client systems connect- ed to the global query processor via network. Global query processor accepts queries from clients, schedules in an optimal fashion and distributes them to appropriate servers. We evaluate global query processing mechanisms based on various queuing methods to achieve an optimal performance. Based on the evaluations we implement these multi-user database system to verify our design.

Comparisons are conducted among various scheduling mechanisms in terms of waiting time and server utilization. We observed the analytical and experimental results of our mechanism agree with each other to a high degree. The experimental results confirms that this system offers an affordable way of upgrading existing database systems. Also, we can utilize this results to estimate the system specifications such as the number of servers to meet given performance requirements.

## 1. 서 론

개인용 데이터베이스 시스템은 소규모의 조직에서 경제적인 데이터 처리 기능을 제공해 주고 있지만 점차로 복잡한 정보 처리의 필요성이 생김에 따라 데이터를 조직 내의 다른 데이터베이스 사용자들과 공유해야할 필요성이 증대된다. 이러한 필요성에 대처할 때의 실질적 고려 사항은 첫째, 여러 사용자들이 데이터베이스를 공유할 때 발생하는 데이터베이스의 성능 저하를 막는 방안을 강구해야 하고 둘째, 새로운 데이터 처리 환경으로 이행할 때 기존의 익숙한 사용자 환경을 변화시키지 않아서 사용자가 새로운 환경에 적응하는데 드는 노력을 최소화해야 하는 것이다.

성능저하를 막는 동시에 다중의 사용자를 수용할 수 있는 시스템으로 mainframe용의 다사용자 DBMS는 비용과 응용프로그램들 간의 호환성문제가 단점이다 [SIN92]. 한편 PC를 파일 서버로 하는 다사용자 시스템은 PC의 성능에 의해 데이터처리가 제한되고, LAN상의 데이터 전송량이 과다한 단점이 있다 [SAL93].

분산데이터베이스 시스템은 이상적인 형태의 시스템이나 설계와 유지가 어려워 소규모의 환경에는 적합하지 않다. 더구나 기존의 환경을 확장해 나가는 경우는 이중 분산시스템이 필요하나 실용화는 아직 어려운 실정이다 [SAL93, CER85]. 기존의 client/server system은 실질적인 여러 장점을 가지고 있으나 사용자가 늘어나면 서버의 이용도(availability)가 떨어진다. 이와같은 상황에 비추어 본 연구는 데이터처리 용량이 한계에 도달한 수많은 중소규모의 데이터베이스환경에서 성능향상과 다사용자환경으로의 효과적인 이행이 가능하도록 해주는 시스템의 개발을 목표로 한다.

본 논문에서는 위와 같이 기존의 다사용자 데이터베이스 시스템이 만족하지 못하는 요구 사항, 즉 기존의 사용중인 데이터베이스 환경을 변화시키지 않으면서 성능의 개선을 충족하는 다중 사용자 데이터베이스 시스템을 설계하고 성능을 분석한다. 구체적으로 시스템의 구조를 설계하고, 전체적인 관점에서의 최적의 질의 처리방법을 개발한다. 시스템의 기본 구조는 Client/Server system [SAL93, SIN92]에 기반하고, 다중의 서버를 갖

는다. 다중의 읽기전용 서버는 같은 내용의 데이터를 가지며 질의처리 성능의 저하를 막고 데이터의 이용도와 신뢰도(reliability)를 높여 준다. 중복 데이터간의 불일치 해소 문제(CER85)를 단순화하기 위해 쓰기질의(update queries)의 처리는 하나의 서버(primary copy)에서만 이루어지게 한다.

이를 위해서 본 논문에서는 II장에서 전체적인 시스템 구성을 제시하고 그 시스템의 개괄적인 동작 과정을 살펴 본다. III장에서 본 논문에서 제시된 시스템을 기초로 해석적 분석 방법과 실제 구현을 통해 효율적 질의 처리 방안을 개발한다. 구체적으로 큐잉 모델을 기본으로 해서 대표적인 몇 가지 모델의 성능을 해석적으로 비교 분석한다. IV장에서는 해석적 방법을 통해 분석이 용이하지 못한 모델을 포함해서 가능한 모든 모델에 대해 실제 시스템을 구현한다. 그리고 구현된 각 시스템의 실험을 통해 가장 우수한 질의 처리 방안을 개발하고 해석적 방법의 정당성을 검증한다. 또한 실험 결과를 통해 해석적 분석 방법으로 분석이 용이하지 못한 모델의 성능도 추정한다. V장에서 결론을 맺는다.

## II. 시스템 개요

### 1. 시스템의 구조

본 논문에서 제안하는 시스템의 각 구성 부분은 그림 1에서 보는 바와 같이 근거리 통신망(LAN)으로 연결되어 있다(BAK92). 주요 구성요소를 보면 첫째 여러 개의 데이터베이스 서버들, 둘째 여러 사용자(clients)들로부터 제기되는 질의를 받아 스케줄링을 하는 GQM(Global Query Manager), 셋째 사용자들과 각 사용자와 GQM과의 통신을 담당하는 LQM(Local Query Manager)이다. 본 시스템은 기본적으로 Client/Server system의 구조에 속하지만 분산처리시 성능의 저하를 막고 데이터의 이용도(availability)와 신뢰도(reliability)를 높이기 위해 중복의 데이터를 갖는 다수 개의 데이터베이스 서버를 두는 방식으로 한다. 그러나 빈번한 갱신(update)에 따른 데이터 불일치 문제와 성능 저하를 막기 위해 쓰기는 그중 하나의 데이터베이스에서만 가능하도록 한다(CER85). 이는 단순히 읽기만 원하는 질의가 쓰기를 원하는 질의보다 비교적 많다는 가정하에 기초를 두고 있다. 쓰기가 가능한 데이터베이스는 최신의(up to the minute) 정보를 저장하게 되며 주 데이터베이스(primary database) 역할을 한다. 그 외의 읽기 전용 데이터베이스들은 부 데이터베이스(secondary databases)로 지칭된다. 또한 본 논

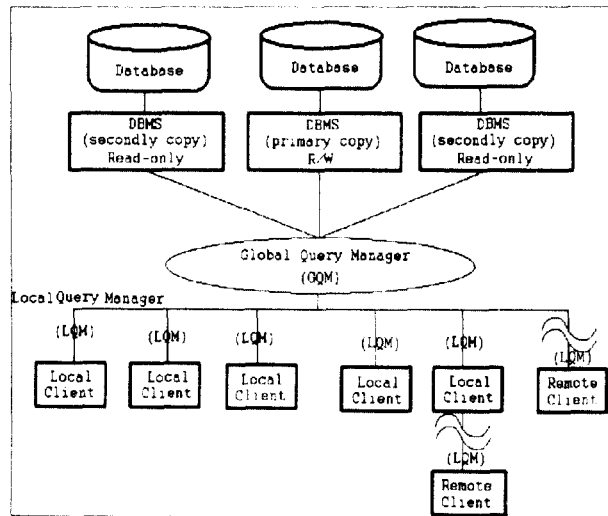


그림 1. 전체 시스템 구성  
Fig. 1. Overall System Structure

문에서는 읽기 전용 데이터베이스들에 대한 갱신의 전파(update propagation)는 질의가 적게 발생하는 시간에 주 데이터베이스 전체를 읽기 전용 데이터베이스들로 복사하는 방식을 취한다. 이러한 update propagation 방법과 달리 쓰기질의를 각 server에 동시에 적용시키는 방안도 가능하다. 한편 복수개의 읽기전용 데이터베이스를 돌으로써 데이터 불일치 해소 작업 도중에도 접근이 가능한 데이터베이스가 항상 한개 이상 존재하게 된다.

## 2. 질의 처리 방법

GQM은 LQM을 통해 보내진 사용자들의 질의들을 유형(type)과 우선 순위(priority)에 따라 다중의 데이터베이스 서버를 위한 최적의 스케줄링을 수행한다. 구체적으로 GQM은 두 가지 단계로 분산 질의들을 처리한다. 첫 단계는 질의들을 두 가지 유형으로 분류 [KLE76]해서 해당 유형의 우선 순위 큐에 보내는 것이고, 둘째 단계는 해당 큐 내의 우선 순위에 의거해서 가장 높은 우선 순위를 가진 질의를 우선적으로 처리하는 것이다. 첫째 유형은 쓰기가 필요한 질의이거나 現時性을 가진(up to the minute) 데이터를 요하는 질의이고, 둘째 유형은 그 외의 질의 다시 말해 현시성(recentness)을 요하지 않는 읽기만 요하는 질의이다. 질의의 타입은 질의발생자에 의해 결정되고, 질의에 표시된다. 따라서, 질의의 현시성의 결정은 사용자에게 전적으로 달려 있고 이의 파악도 간단하게 처리된다. 한편, 전체 질의개수에 대한 현시성을 가진 질의와 쓰기질의 비율이 데이터베이스의 갱신 주기에 비해 충분히 낮다고 가정한다. 그리고 읽기서버의 개수가 상대적으로 많아지면 현시성을 가진 질의의 영향은 매우 적다는 가정을 할 수 있다. 이와같은 조건 아래서 스케줄링방법들의 상호비교와 서버의 개수에 따른 상대적 경향의 파악을 주 목적으로 한다.

동일 유형의 질의들의 우선 순위 설정 방법은 질의들의 상대적 예상 서비스 시간에 따라서 우선 순위를 부여하는 것이다. 첫째 유형은 주 데이터베이스에 기초하여 처리되고 둘째 유형은 나머지 데이터베이스들을 포함한 어느 데이터베이스에서나 실행될 수 있다.

## Ⅲ. 질의 처리 방법들의 해석적 분석

제한된 다중 사용자 데이터베이스 시스템의 해석적 분석을 위해 질의 스케줄링방법을 크게 두 가지 방법으로 모델링할 수 있다. 첫째 방법은 서버 각각이 개별적인 큐를 이용한 스케줄링 시스템(local queueing system)이고, 둘째 방법으로는 하나의 글로벌 큐를 이용한 스케줄링 시스템(global queueing system)이다. 이 두 방법은 큐잉 방법, 즉 First Come First Service (FCFS)나 Shortest Job First (SJF)에 따라 다시 두 가지 방법으로 각각 나눌 수 있다. 두가지의 방법을 택한 이유는 SJF는 이론적으로 queue waiting time 면에서 최적의 스케줄링 방법으로 알려져 있다[SIL94]. 한편, FCFS는 일반적으로는 효율이 떨어지나 단순한 구조이기 때문에 실제 구현된 경우에는 효율적일 수가 있고, 또한 성능상의 비교를 위한 기준(reference)이 될 수 있기 때문이다.

글로벌 큐잉 시스템에서는 SJF 큐잉 방법을 갖는 모델은 수학적 분석이 용이하지 못하다[GRO85]. 따라서 본 논문에서는 수학적 성능의 비교 분석을 이 모델을 제외한 나머지 세 가지 모델에 관해 수행[KLE75]하고 분석하지 못한 모델의 성능은 4장의 구현된 시스템의 모의실험을 통해 분석하기로 한다.

### 1. 로컬 큐를 갖는 모델의 성능 분석

이 모델의 분석의 전제로서 몇 가지 가정이 필요하다.

가정 1) 읽기 전용 서버의 수를 늘림으로써 상대적으로 읽기/쓰기 서버 (Primary Database)로 배분되는 읽기 질의들(read queries)의 수를 무시할 수 있기 때문에 읽기/쓰기 서버는 쓰기 전용 서버로만 가정한다.

가정 2) 사용자의 수가 충분히 크다고 가정하여 질의들의 입력 스트림(stream)은 Poisson (or random) arrival process로 가정한다.

가정 3) 질의들의 서비스 시간은 Poisson (or random) service process로 가정한다.

그림 2에서 보는 바와 같이 주어진 모델의 특징은 아래와 같다. 첫째, 읽기 질의(read query)들의 평균 도착율(average arrival rate)  $\lambda_r$ 과 쓰기 질의(write query)들의 평균 도착율  $\lambda_w$ 는 Poisson 스트림(Poisson stream)으로 가정한다. 그래서 전체 평균 도착율도 Poisson 스트림이 된다[CON67]. 둘째,  $\lambda_w$ 는 전부 Server-1(쓰기 전용)의 큐로 전달된다. 그래서 Server-1의 평균 도착율 1도 포아송 스트림이다.

셋째,  $\lambda_1$ 에서  $\lambda_2, \lambda_3$ 로의 전송 경로가 두 서버에 관계없이 독립적으로 선택된다면  $\lambda_2, \lambda_3$ 도 역시 포아송 스트림이다(CON67). 즉  $\lambda_1$ 에서  $\lambda_2, \lambda_3$ 로의 경로 선택은 각각 동일 확률을 가진다. 그래서  $\lambda_2 = \lambda_3 = \lambda_1/2$ 이고 위의 모델은 세계의 독립적인 SJF 방법을 가진 M/M/1 모델로 단순화할 수 있다. 넷째, 각 서버의 평균 서비스율(average service rate)은  $\mu$ 로 동일하다고 가정한다. 다섯째, 질의가 세 개의 큐로 배분되는 지점의 큐잉 시간은 무시한다. 그것의 근거는 그 지점의 기능이 단순히 입력 질의가 읽기 질의인지 쓰기 질의 인지의 유형만 구분하면 되기 때문이다.

본 모델에서는 성능의 척도로서 질의당 평균 큐 대기 시간(average queue waiting time)  $\bar{W}$ 를 고려한다. 성능 분석은 먼저 SJF 큐잉 방법을 가진 M/M/1 모델에 대하여 행해지고 이를 기초로 시스템 전체에 대한 통합 평가가 행해진다. 이에 대한 분석은 다음과 같다. 서비스 시간이  $x$ 인 질의에 대한 SJF 큐잉방법을 갖는 단일 서버에 대한 평균 큐 대기 시간은 다음과 같이 유도된다(KLE76).

$$\bar{W}_{SJF, one-server}(x) = \frac{W_0}{[1 - \lambda \int_0^x y dB(y)]^2}$$

Y : 서비스시간을 나타내는 확률변수

y : 임의의 서비스 시간

$$W_0 = \frac{\lambda}{2} \int_0^c y^2 dB(y) = \frac{\lambda y^2}{2}$$

$$= \lambda/2 E(Y^2)$$

$$\begin{aligned} \therefore \bar{W}_{SJF, one-server} &= \int_0^c W_{SJF, one-server}(x) dB(x) \\ &= W_0 \int_0^c \frac{dB(x)}{[1 - \lambda \int_0^x y dB(y)]^2} \end{aligned}$$

서비스 시간의 cdf는  $B(y) = 1 - e^{-\mu y} \quad y \geq 0, \mu > 0$ 이므로 SJF 방법을 갖는 M/M/1 모델의 평균 큐 대기 시간은 다음과 같이 계산할 수 있다.

$$\bar{W}_{SJF, one-server}(x) =$$

$$\frac{\frac{\lambda}{2}}{[1 - \rho(1 - e^{-\mu x}) + \lambda x e^{-\mu x}]^2}$$

$$\bar{W}_{SJF, one-server} =$$

$$\rho \int_0^c \frac{e^{-\mu x}}{[1 - \rho(1 - e^{-\mu x}) + \lambda x e^{-\mu x}]^2} dx$$

위의 적분은 상수인  $\lambda$ 와  $\mu$ 가 주어질 때 컴퓨터를 통해서 근사적으로 계산할 수가 있다. 그 결과 그림 2에 주어진 모델에 관한 전체 평균 큐 대기 시간은 다음과 같이 유도할 수 있다.

$$\bar{W}_{SJF, LQ} = \sum_{i=1}^c \frac{\lambda_i}{\lambda} W_{i, SJF}$$

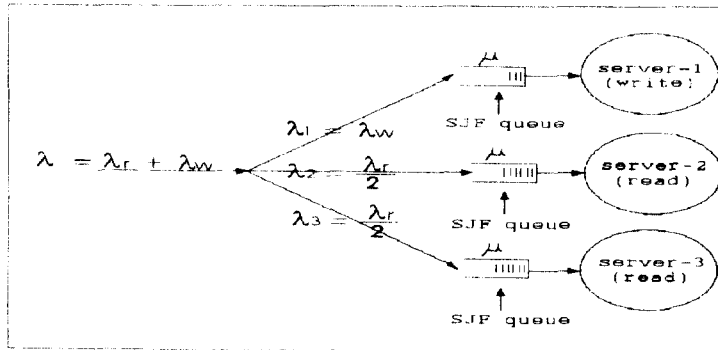


그림 2. 로컬 SJF 큐를 갖는 모델  
Fig. 2. Model with local SJF queues

위 식에서  $W_{i,SJF}$ 는 SJF 방법을 갖는 서버  $i$ 의 평균 큐 대기 시간이며  $W_{2,SJF} = W_{3,SJF}$ 이다.  $c$ 는 서버의 개수를 나타낸다.

한편 질의들이 큐에 입력되는 순서대로 서비스 받는 FCFS M/M/1 모델의 평균 큐 대기 시간은 아래와 같이 유도된다[ALL90].

$$\bar{W}_{FCFS} = \frac{\rho}{\mu(1-\rho)} = \frac{\lambda}{\mu(\mu-\lambda)}$$

따라서 그림 2의 모델에서 각 큐의 큐잉방법이 FCFS 일 경우 전체 평균 큐 대기 시간 또한 다음과 같이 유도할 수 있다.

$$\bar{W}_{FCFS\ TQ} = \sum_{i=1}^c \frac{\lambda_i}{\lambda} W_{i,FCFS}$$

2. 글로벌 FCFS 큐를 갖는 모델의 성능 분석

글로벌 FCFS 큐잉 모델은 그림 3과 같이 나타낼 수 있다. 제안된 시스템의 특성상(즉 각 서버가 동일한 종류의 서버가 아니다.) 질의가 입력되는 부분을 두 부분으로 나누어서 고려하고 있다. 즉 쓰기 질의를 받는 부분의 주 데이터베이스 서버(Server-1)는 항상 하나이며 큐잉 방법은 SJF로 한다. 그 이유는 앞 절의 분석에서와 같이 단일 서버를 갖는 SJF 시스템은 단일 서버를 갖는 FCFS 시스템에 비해 큐 대기시간 면에서 우수하기 때문에 단일 서버로 구성되는 주 데이터베이스는

SJF큐로 한다. 한편 읽기 질의가 도달하는 부분은 읽기 전용 데이터베이스 서버 (Server-n, n>2)로서 두 개 이상으로 숫자를 증가시킬 수 있다. 이는 동일 종류의 다중 서버로서 글로벌 FCFS 큐잉 방법으로 모델링한다. 따라서 성능 분석은 읽기 질의가 도달하는 부분에 대해서만 행한다.

질의들의 입력이 포아송 스트림이고 서비스 시간 역시 포아송 분포를 이룬다고 가정할 때 그림 3에서 읽기 질의가 입력되는 부분은 M/M/c 큐잉 모델로 단순화할 수 있다. 따라서 평균 큐 대기 시간은 다음과 같이 유도된다[ALL90].

$$\begin{aligned} \bar{W}_{FCFS\ CQ} &= \frac{P[N \geq c] \bar{x}}{c(1-\rho)} \\ P[N \geq c] &= C[c, a] \\ &= \frac{\frac{a^c}{c!}}{(1-\rho) \sum_{n=0}^{c-1} \frac{a^n}{n!} + \frac{a^c}{c!}} \end{aligned}$$

위 식에서  $\rho = \frac{\lambda}{c\mu}$ ,  $\bar{x}$ 는 평균 서비스 시간을 나타내고  $C[c, a]$ 는 Erlang's C formula로서 시스템이 busy 할 확률이다. 또한  $a = \frac{\lambda}{\mu}$ 이고  $c$ 는 서버의 개수이다.  $N$ 은 시스템 (큐와 서버) 상의 질의의 개수를 나타낸다.

3. 성능 분석 결과 및 검토

3장 1절과 2절의 모델들에 대해 컴퓨터를 이용한 수

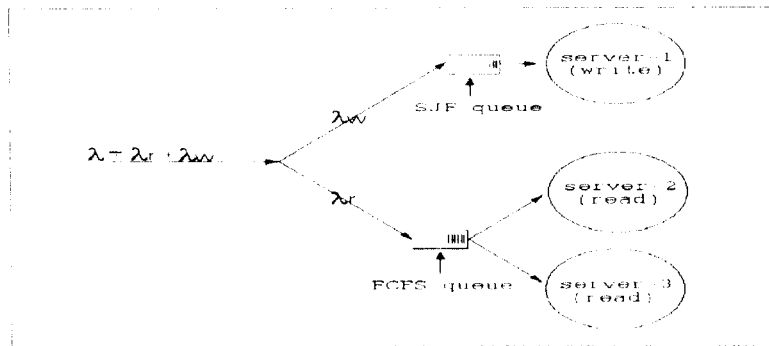


그림 3. 글로벌 FCFS 큐를 갖는 모델  
Fig 3. Model with global FCFS queue

치 해석을 통해 분석된 결과를 비교 검토하면 아래와 같다.

1) 로컬 큐를 갖는 모델

본 절에서는 그림 2의 모델을 분석한 결과를 기술한다. 그림 4에서는  $\lambda_r$ 과  $\lambda_w$ 의 비가 4 : 1로 주어지고  $\lambda$ 와  $\mu$ 가 입력 파라미터로 주어질 때 읽기 전용 서버의 개

수 증가에 따른 평균 큐 대기 시간의 변화를 나타내었다. 그림 4에서는 각각의 질의 도달율에 해당하는 큐 대기 시간( $W_{SIF-LQ}$ ,  $W_{FCFS-LQ}$ )이 나타나 있다.

그림 5에서는 쓰기과 읽기 질의의 비가 " $\lambda_w : \lambda_r = 1 : 1$  : 읽기 서버의 개수"와 같을 때  $\lambda$  변화에 따른 평균 큐 대기 시간의 변화를 나타내었다. 그런데 특별히 쓰기과 읽기 질의의 비가 " $\lambda_w : \lambda_r = 1 : 1$  : 읽기 서버의 개수"와

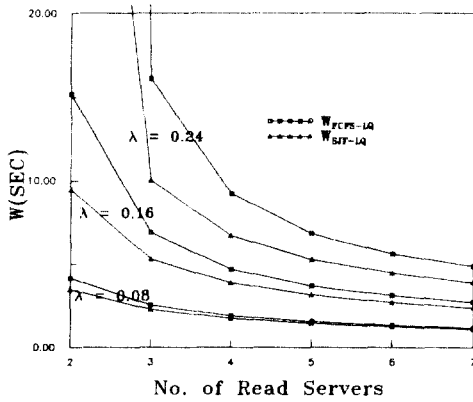


그림 4. 읽기 서버 개수에 따른 평균 큐 대기 시간  
Fig. 4. Avg. queue waiting time to # of read servers

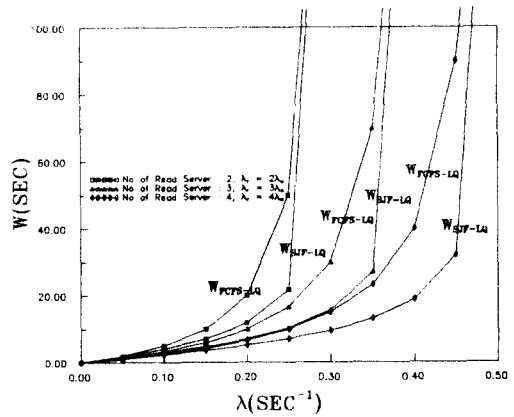


그림 5.  $\lambda$ 에 따른 평균 큐 대기 시간  
Fig. 5. Avg. queue waiting time to  $\lambda$

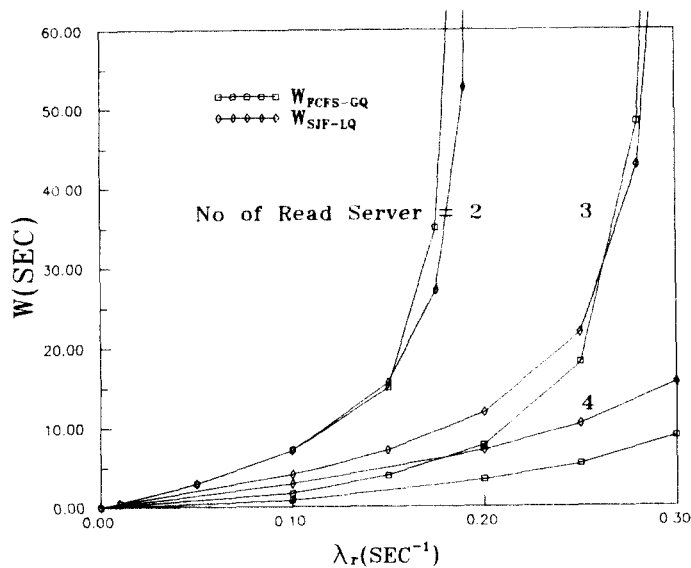


그림 6.  $\lambda_r$ 에 따른 평균 큐 대기 시간  
Fig. 6. Avg. queue waiting time to  $\lambda_r$

같은 경우를 고려한 이유는 쓰기과 읽기 질의의 비가 그러한 때에 그림 2의 모델에서 전체 평균 큐 대기 시간의 최소로 되기 때문이다. 이것은 직관적으로 볼 때 타당한 결과이다. 즉 쓰기과 읽기 질의의 비가 그러한 때 질의들이 세개의 서버 각각으로 균등하게 분포되기 때문이다.

## 2) 글로벌 FCFS 큐를 갖는 모델

그림 6에서는 읽기 질의가 입력되는 부분이 글로벌 FCFS 큐를 갖는 그림 3의 모델과 그 부분이 로컬 SJF 큐를 갖는 모델의 큐 대기 시간을 읽기 서버의 개수가 변할 경우에 대해 비교해서 나타내었다.

이상의 분석으로부터 알 수 있는 사실은 위에서 분석한 세 가지 모델 중 두 가지 모델(로컬 SJF 큐를 갖는 모델과 읽기 질의가 도달되는 부분이 글로벌 FCFS 큐를 갖는 모델)보다도 확실한 성능 우위라는 것을 알 수 있다. 그런데 이 두 가지 모델(로컬 SJF 큐를 갖는 모델과 글로벌 FCFS 큐를 갖는 모델)은 질의의 도달율에 따라서 성능 우위가 좌우된다. 즉 질의의 도달율이 비교적 작을 때는 글로벌 FCFS 큐를 갖는 모델이 로컬 SJF 큐를 갖는 모델 보다도 성능이 우월하고 질의의 도달율이 상당히 증가할 경우에는 반대의 현상을 발견할 수 있다.

## IV. 구현 및 검토

3장에서 분석된 질의 스케줄링 모델들과 해석적 분석이 용이하지 않았던 글로벌 SJF 모델에 대해서 실제로 질의처리 시스템을 구현하여 분석한다. 이러한 구현을 통해 해석적인 분석 결과를 검증하고, 또한 분석이 불가능한 글로벌 SJF 모델의 성능을 추정한다. 이와함께 그림 2의 모델에서 질의 분배 지점에서 질의 분배를 최적 분배 방법(즉 큐에 있는 질의들 뿐만 아니라 서버에서 수행중인 질의의 잔류 서비스 시간까지 고려하여 질의를 분배하는 알고리즘)을 사용할 경우에 대해서도 성능을 실험적으로 알아본다.

### 1. 구현 환경 및 시스템 개요

구현환경은 통신망으로 연결된 UNIX workstation 들로 구성되어 있다. 통신기능의 구현에는 UNIX에서

제공되는 Socket과 UDP(User Datagram Protocol)를 사용한다(RAG93). 그림 7에 본 질의처리 시스템의 전체적 구조가 주어져 있다.

본 논문에서 구현한 시스템에서는 도달 질의들의 예상 서비스 시간은 미리 추정할 수 있다고 가정한다(KOR91). 그리고 본 시스템에서 각각의 서버는 독립된 프로세스(RAG93)로서 구현되고 질의를 할당받은 서버는 질의 처리를 위해 필요로 하는 서비스 시간만큼 경과후에 질의 처리 완료 신호(reply the system message)를 그 질의를 전송한 해당 클라이언트로 보내도록 되어 있다. 그러나 이것을 제외한 나머지 부분은 실질적으로 통신망을 이용한 분산 질의 처리 시스템으로 동작한다.

## 2. 시스템 구조

### 1) 글로벌 큐를 갖는 다중 서버 시스템

이 시스템은 기능에 따라 세 개의 계층으로 구성되고 각 계층은 상호 유기적으로 동작하는 프로세스들로 구성된다. 그림 8에 그 구조가 주어져 있다. 각 계층에 대한 설명은 다음과 같다.

#### ○ 통신모듈 (communicator: parent process)

질의를 전송 받아서 글로벌 큐(SJF 혹은 FCFS)에 해당 스케줄링 방법에 따라 저장한다.

#### ○ 질의분배모듈(query distributor: communicator의 child process)

첫째, 먼저 서버의 상태(busy 혹은 idle)를 체크한다. 둘째, 서버의 상태가 idle 하다면(즉, 다중 서버중 idle한 서버가 있으면) 글로벌 큐에 저장된 질의를 해당 서버로 넘겨준다. 셋째, 서버의 상태가 busy 하다면 세개의 서버중에서 idle한 서버가 나올 때까지 계속 기다린다.

#### ○ 서버(query distributor의 child process)

실질적 질의 처리를 행하는 부분으로서, 첫째, 해당 질의의 서비스 시간만큼 그 질의를 잡아준다(hold). 둘째, 서비스 완료후 서비스를 완료했다는 정보를 그 질의를 전송한 해당 client로 전송한다.

### 2) 로컬 큐를 갖는 다중 서버 시스템

이 시스템은 기능에 따라 네 개의 계층으로 구성되고 이 네개의 계층은 상호 유기적으로 동작하는 프로세스들로 구성된다. 그 구조가 그림 9에 주어져 있다. 각 계층



의 기능은 다음에 설명되어 있다.

- communicator(parent process)

질의들을 전송 받아서 일단 버퍼(FCFS 구조)에 저장한다.

- query distributor(communicator의 child process)

버퍼(FCFS 구조)에서 가져온 질의를 최적의 분배 알고리즘과 로컬 큐로의 동일한 분배 확률을 가지는 무작위 분배 알고리즘을 각각 사용하여 해당로컬 큐 (SJF 혹은 FCFS 구조)로 분배한다.

- query extractor(query distributor의 child process)

첫째, 서버의 상태(busy 혹은 idle)를 체크한다. 둘째, 서버의 상태가 idle하다면 각각의 로컬 큐에 저장된 질의를 해당 서버로 넘겨준다. 셋째, 서버의 상태가 busy하면 각 서버가 idle할 때까지 계속 기다린다.

- 서버(query extractor의 child process)

앞 질의의 글로벌 큐를 갖는 다중 서버 시스템에서의 서버 기능과 동일하다.

#### 4. 실험 결과 및 검토

본 실험에서는 실제와 다르게 몇가지 제한 요소를 둔다. 그것은 첫째, LAN을 통해 연결된 서버의 상태와는 독립적으로 질의들은 별도의 workstation상에서

exponential 간격으로 발생한다. 둘째, 발생 질의 자체의 서비스 시간도 exponential 분포를 가진다.

질의의 서비스 시간의 예측은 개별적인 추산을 할 수 있을 뿐이다 [KOR91]. 본 연구를 위해, 서버 데이터베이스를 관계데이터베이스에 국한하고 그것이 제공하는 통계적 특성(DAT91)과 연산의 종류에 따라 구체적인 추산 알고리즘을 개발하였다 [박종94].

다음은 본 시스템에 대한 몇 가지 실험 결과이고 거기에 사용된 성능(큐 대기 시간) 척도에 대한 표기는 : ①  $W_{xxx-GQ}$  - 글로벌 큐잉 시스템의 큐 대기 시간( $xxx$  : SJF 혹은 FCFS) ②  $W_{xxx-LQ-yyy}$  - 로컬 큐잉 시스템의 큐 대기 시간( $xxx$  : SJF 혹은 FCFS,  $yyy$  : ran 혹은 opt 즉 random 혹은 optimal distribution) 그림 10의 실험 결과로부터 여섯 가지 시스템중 글로벌 SJF 큐잉 시스템의 성능이 가장 우수하다는 사실을 알 수 있고 또한 로컬 큐잉 시스템에서는 최소 큐 길이를 만족하는 질의 분배 알고리즘(optimal distribution algorithm)을 적용한 시스템이 무작위 질의 분배 알고리즘(random distribution algorithm)을 사용한 시스템보다 최고 5배 이상의 성능 향상을 보인다는 사실도 알 수 있다. 그림 11의 결과로부터 해석적 방법으로 분석한 세 가지 모델의 분석 결과와 구현 시스템의 실험 결과는 유사한 경향을 나타낸다는 사실을 알 수 있다.

그림 12의 결과로부터 특정 질의 도달율이 주어질 경

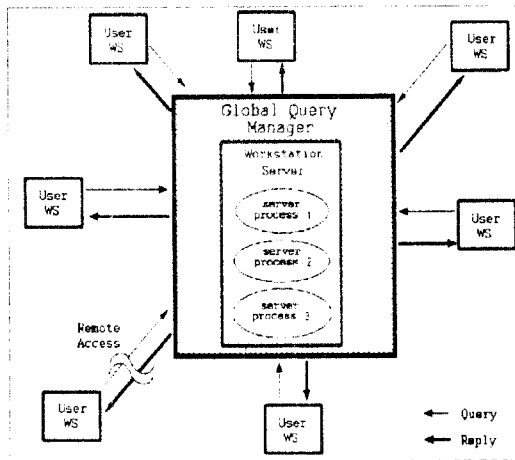


그림 7. 시스템의 전체적 동작 과정  
Fig. 7. Overall functional diagram

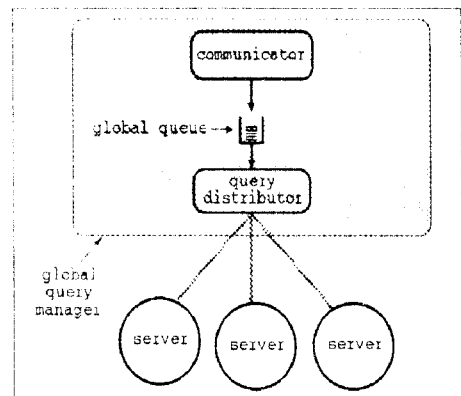


그림 8. 글로벌큐를 갖는 다중 서버 시스템  
Fig. 8. Multi-server system with global queues

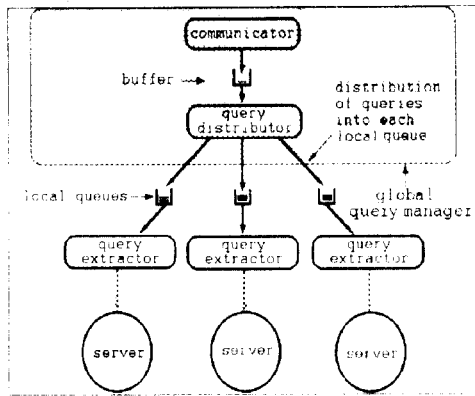


그림 9. 로컬큐를 갖는 다중 서버 시스템  
Fig. 9 Multi-server system with local queue

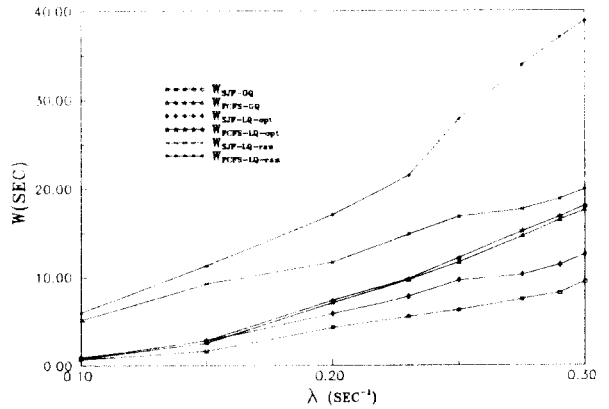


그림 10. λ에 따른 큐 대기 시간의 실험 결과(#=3)  
Fig. 10. Experimental results for queue waiting time to λ(#=3)

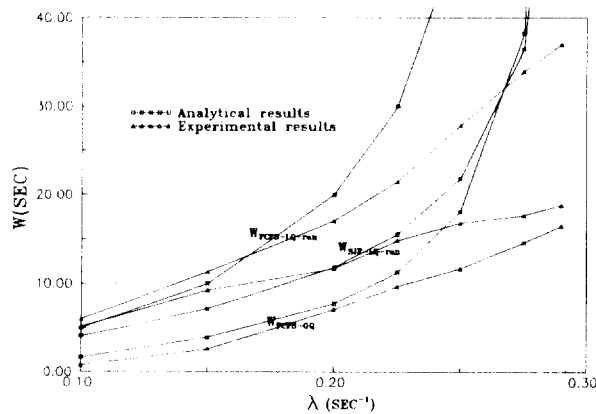


그림 11. 실험 결과대 해석적 분석 결과의 비교 (#=3)  
Fig. 11. Comparison between analytic and experimental results (#=3)

우 필요로하는 대기 시간을 만족하는 서버의 개수를 산출할 수 있다. 한편 해석적 방법으로 분석이 용이하지 못한 다중 서버를 갖는 글로벌 SJF 큐잉 시스템의 큐 대기 시간을 실험 결과를 통해 근사적으로 추정할 수 있다는 것이다. 그것의 근거는 글로벌 FCFS 큐잉 시스템 (이 시스템은 질의 분배 지점의 큐잉 지연을 고려할 필요가 없기 때문에 이론과 실험 결과를 비교하여 가장 적합하다.)의 실험 결과와 해석적 분석 결과가 상당히 유

사하게 나타난다는 사실에 기인한다. 구체적으로 실험 결과를 통해 글로벌 FCFS와 글로벌 SJF 시스템의 성능 차이를 구하고 그것을 이용하여 다중 서버를 갖는 글로벌 SJF 큐잉 모델에 대한 성능을 추정할 수 있다. 추정의 구체적인 방법으로 선형회귀 [WIL92] 등을 이용할 수 있다 [SI94]. 그리고 추정 가능 범위는 큐 대기 시간이 극단적으로 증가하지 않는 범위에서만 가능하다. 그림 13은 그 결과를 보여준다.

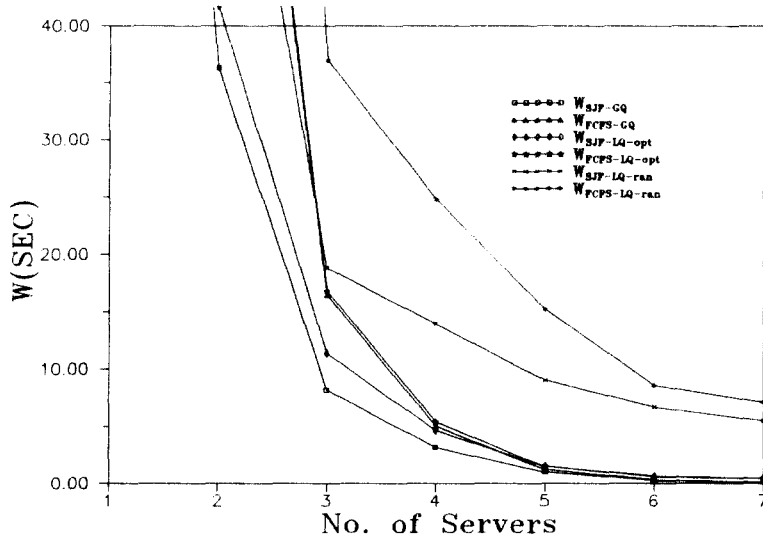


그림 12. 서버 개수(＃)에 따른 큐 대기 시간의 실험 결과 ( $\lambda=0.29$ )  
 Fig. 12. Experimental results for queue waiting time to # ( $\lambda=0.29$ )

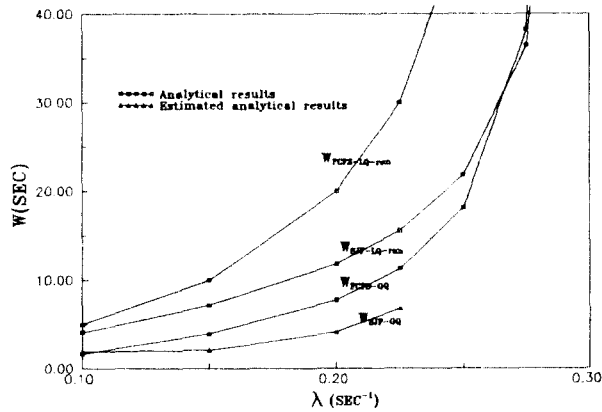


그림 13. 글로벌 SJF 큐잉 모델에 대한 성능의 추정 결과(＃=3)  
 Fig. 13. Estimated performance for model with global SJF queue

### V. 결 론

기존의 개인용 데이터베이스 사용자들이 사용하던 환경을 그대로 유지하면서 데이터의 이용도와 신뢰도를 향

상시킬 수 있는 기능을 제공하는 다사용자를 위한 데이터베이스 시스템 모델을 개발하였다. 데이터 처리의 성능을 높이는 방안으로 다중의 서버를 사용하고 이를 위한 방안으로 최적의 질의 스케줄링 방법을 개발하였다.

결과적으로 데이터의 처리 수요나 공유의 필요에 따라 시스템의 성능을 융통성 있게 증감시킬 수 있다.

시스템 기본 구조를 보면 다수의 사용자들이 통신망을 통하여 글로벌 질의처리 관리자(global query manager)에 질의 처리를 의뢰하고, 그 관리자는 질의들의 상대적인 처리 시간에 근거하여 최적의 스케줄링을 행한다. 이러한 스케줄에 따라 질의들은 다중의 서버를 통해 처리되는 기본 구조를 가진다. 설계 방법으로는, 먼저 다수의 서버를 가진 큐잉 모델에 기초하여 해석적인 방법으로 분석하였다. 그리고 해석적 방법으로 분석이 용이하지 못한 모델을 포함해서 제안된 여러 모델들에 대하여 실제 시스템의 구현을 통해 효율적 스케줄링 방법을 찾아내었다. 이러한 구현을 통해 해석적 분석 방법의 정당성을 검증함과 동시에 해석적 방법으로 분석이 용이하지 못한 수학적 모델의 성능을 추정하였다.

### 참고문헌

- [ALL90] Allen, Arnold O., "Probability, Statistics and Queueing Theory with Computer Science Application," 2nd Ed., Academic Press, 1990, pp.247-336.
- [BAK92] Baker, Richard H., "Downsizing : How to get big gains from smaller computer systems," McGraw-Hill, Inc., 1992, pp.65-174.
- [CER85] Ceri S. and Pelagatti G., "Distributed Databases," McGraw-Hill, International Ed., 1985, pp.266-271.
- [CON67] Conway, Richard W., et. al, "Theory of Scheduling," Addison-Wesley Publishing Company, 1967, pp.141-190.
- [DAT91] Date C., "An Introduction to Database Systems," Vol.1, 5th ed., Addison-Wesley Co., pp.3-21
- [GRO85] Gross, D., Harris Carl M., "Fundamentals of Queueing Theory," 2nd Ed., John Wiley & Sons, 1985, p.210.
- [KLE75] Kleinrock, L., "Queueing Systems, VOL I : Theory," John Wiley & Sons, Inc., 1975, pp.87-240
- [KLE76] Kleinrock, L., "Queueing Systems, VOL II : Computer Applications," John Wiley & Sons, Inc., 1976, pp.106-269.
- [KOR91] Korth, Henry F. and Silberschatz A., "Database System Concepts," 2nd Ed., McGraw-Hill International Editions, 1991, pp.281-311.
- [RAG93] Rago, Stephen A., "UNIX System V Networking Programming," Addison-Wesley Professional Computing Series, 1993, pp.291-353.
- [SAL93] Salemi, Joe, "Guide to Client/Server Databases," Ziff-Davis Press, 1993, pp.1-59.
- [SILN94] Silberschatz, A. and Galvin P., "Operating System Concepts, Addison-Wesley Co., 4th ed., 1994, pp.131-145.
- [SIN92] Sinha, Alok, "Client/Server Computing," Communications of the ACM, July 1992, Vol. 35, No. 7, pp.77-98.
- [SI94] Si J., "LAN에 기초한 다중의 서버를 이용한 분산 데이터베이스 질의처리를 위한 스케줄링 방법의 개발," 경북대 석사학위 논문, 1994, pp. 38-41.
- [ULL93] Ullman E., "Client/Server Frees Data," Byte, June 1993, pp.96-108.
- [WIL92] William, H. Press, et. al, "Numerical Recipes in C," Cambridge University Press 2nd Ed., 1992, pp.656 - 699.
- [박종94] 박종희, "PC 네트워크에 기초한 Remote Access 기능을 가진 보급형 Multi-User 데이터베이스 관리 시스템의 개발", 통신학술 연구보고서, 1994.



施鍾嫻(HJong Ik Si) 정희원

1968년 11월 24일생.  
1993년 2월 : 경북대학교 전자공학과 졸업(공학사).  
1995년 : 경북대학교 대학원 졸업(공학 석사).

현재 : LG 전자기술원 연구원.

\*주관심 분야 : 멀티미디어, 데이터베이스.



朴宗熹(Jong Hee, Park) 정희원

1979年 : 서울대학교 졸업(공학사).  
1981年 : 한국과학원 졸업(공학석사).  
1990年 : Univ. of Florida 졸업(공학박사).

현재 : 경북대학교 전자공학과 조교수.

\*주관심 분야 : 멀티미디어 응용, Computer Aided Education, CAD/CAM, 지능형 정보 시스템, 분산 데이터 처리 시스템.