효율적인 유전자 서열 비교를 위한 데이타베이스 검색 모델

(A Database Retrieval Model for Efficient Gene Sequence Alignment)

김 민 준 † 임 성 화 †† 김 재 훈 ††† 이 원 태 †††† 정 진 원 †††††
(Min Jun Kim) (Sung-Hwa Lim) (Jai-Hoon Kim) (Weontae Lee) (Jin-Won Jung)

요 약 대부분의 생물정보학의 프로그램들은 데이타베이스로부터 유전자 등의 데이타를 검색하고 처리하여 생화학자와 생물학자에게 서비스를 제공한다. 이때 각각 클라이언트의 요청마다 데이타베이스의 검색을 수행한다면 많은 디스크 접근 시간이 소요 된다. 또한 서버에 과부하를 초래하여 응답시간이 길어질 수 있다. 본 논문에서는 생물정보학에서 서열 검색 프로그램의 데이타베이스 사용 패턴을 이용하여 많은 데이타베이스 요청에 대하여 데이타베이스의 검색을 위한 디스크 접근을 공유하는 그룹핑 기법을 제안한다. 또한, 사용자 요청을 대기 시간 없이 처리중인 작업과 동시에 데이타베이스의 검색을 위한 디스크 접근을 공유하여 시스템 처리율을 높이고 빠른 응답시간을 가지는 카플 방식을 제안한다. 제안된 기법은 수학적 분석과 시뮬레이션을 통하여 성능을 검증하였다.

키워드 : 생물정보학, 유전자, 단백질, 데이타베이스, 서열정렬

Abstract Most programs of bioinformatics provide biochemists and biologists retrieve and analysis services of gene and protein database. As these services retrieve database for each arrival of user's request, it takes a long time and increases server's load and response time. In this paper, by utilizing database retrieval patterns of sequence alignment programs in bioinformatics, grouping method is proposed to share database retrieval between many requests. Carpool method is also proposed to reduce response time as well as to increase system expandability by combining new arriving requests with the previous on going requests. The performance of our two proposed schemes is verified by mathematic analysis and simulation.

Key words : Bioinformatics, Gene, Protein, Database, Sequence alignment

1. 서 론

21세기 초에 인간 유전자 프로젝트가 성공적으로 수행되어 많은 생명과학 분야에 발전을 가속화 시켰다. 이러한 인간유전체 지도의 완성으로 전개되는 유전자이후 시대(post Genom)에는 인간의 모든 유전자와 유전자의 발현으로 생성되는 단백질들의 구조와 기능에 관한 연구가 활발히 수행될 것이다. 이런 연구는 막대한 양의 디지털 정보를 낳았다. 유전자 정보를 A, T, G, C라는 네 개의문자로 표현하여 30억개 이상의 정보를 축적하게 되었으며[1] 이렇게 저장된 정보는 데이타베이스로 구축되어 웹을 통해 공개되었다. 이런 예로, Swiss-Prot[5], GenBank[6], EMBL[7] 등이 있다.

데이타베이스를 사용자 요청에 따라 검색하여 비교하고 알맞은 유전자 정보를 찾아주는 많은 소프트웨어가 있다. 이러한 소프트웨어는 A, T, G, C로 이루어진 데이타를 비교 검색하여 서열비교를 수행하는 FastA, Blast, ClustalW등의 패턴 매치 프로그램과 데이타의 서열로부터 구조를 예측하는 J-NET이나 J-PRED와 같은 프로그램으로 나뉜다. FastA는 사용자가 비교를 의

· 본 연구는 정보통신부 정보통신선도기반기술개발사업의 지원에 의하여 이루어진 것임
† 비 회 원 : 미디어코리스(주) 부설정보통신연구소
xholic@korea.com
†† 비 회 원 : 현대디지탈테크㈜ 미디어 연구소
holyfire@ajou.ac.kr
††† 정 회 원 : 아주대학교 정보통신전문대학원 교수
jaikim@ajou.ac.kr
†††† 비 회 원 : 연세대학교 생화학과 교수
wlee@spin.yonsei.ac.kr
††††† 비 회 원 : 연세대학교 생화학과
solwind@spin.yonsei.ac.kr
논문접수 : 2002년 10월 5일
심사완료 : 2004년 2월 20일

뢰한 서열에 대해서 유사성을 갖는 서열을 데이타베이스로부터 검색하는 프로그램으로써 단백질 서열간의 비교 및 단백질 서열과 염기서열 데이타베이스와의 비교도 가능하다. Blast는 FastA와 마찬가지로 유전자 서열을 비교 검색하는 프로그램이다. Blast는 처리속도의 향상을 위하여 사용자 요청 서열로부터 3개의 단백질 또는 11개의 염기서열의 조합을 추출하여 데이타베이스의 서열과 비교한다. 데이타베이스에서 유사한 서열이 검색되면 이 조합을 확장시킨다. 확장을 마친 후, 데이타베이스 서열 중 일정값 이상의 HSP(High-scoring Segment Pair)를 갖는 서열을 추출하고 그 외 나머지 서열은 비교하지 않는다. 이렇게 높은 HSP를 갖는 서열만을 분석함으로써 FastA보다 빠른 검색 속도를 가지지만, 특정한 부분이 아닌 전체적인 유사성을 갖는 서열을 검색하지 못하는 단점이 있다. 이와 같은 서열 분석 프로그램을 이용해서 단백질 서열과 염기서열의 비교 뿐 아니라 구조예측을 통한 연구가 활발히 진행되고 있다. 앞으로의 생물학 연구는 직접적인 실험보다는 생물정보학의 소프트웨어 활용에 더 의존하게 될것이다. 이는 단순한 데이타베이스로부터 데이타를 제공하는 것 이외에 유전자 자체에 대한 완벽한 이해를 목적으로 함을 의미한다. 이로 인해 생물정보학의 소프트웨어가 더 강력한 기능과 컴퓨팅 파워를 가져야 할 것이다. 또한, 생물정보학에서 사용되는 데이타베이스는 생물정보 연구가 진행됨에 따라 데이타의 크기가 폭발적으로 커지고 있다. 이런 데이타베이스 크기의 증가는 생물정보학에서 데이타베이스의 효율적인 사용의 중요성을 증대시키고 있다.

생물정보학의 서열 검색에서 사용되는 데이타베이스는 일반 데이타베이스와는 달리 전체 검색이 많으며 데이타를 액세스할 때 그 순서는 결과에 영향을 미치지 않는 특징이 있다. 이런 특징에 맞춰 생물정보 데이타를 이용하는 프로그램들의 성능을 증가시키기 위한 프로그램 모델을 제안한다. 우선 사용자 요청을 모아서 한번에 처리하는 사용자 요청 그룹핑 기법은 일정 주기 동안 사용자 요청을 모아서 데이타베이스를 한번만 검색하고 여러 번의 처리를 하게 된다. 따라서 데이타베이스의 검색 횟수를 줄여 응답시간과 시스템 비용을 줄일 수 있었다. 또한 카플방식은 그룹핑 방식에서 사용자 요청을 그룹핑 하기 위해서 지연되는 시간없이 사용자가 요청을 하면 이전에 처리하던 작업을 끝날 때까지 기다리지 않고 같이 처리 함으로써 데이타베이스를 한번만 검색하게 된다. 즉, 사용자 요청 하나당 데이타베이스의 검색을 위한 디스크 접근 횟수를 줄여서 시스템 비용과 응답시간에서 이익을 볼 수 있다.

본 논문에서 제안하는 그룹핑 모델과 카플 모델은 데이타베이스를 모두 검색하여 검색된 모든 데이타와 사용자의 서열들 간의 비교를 하는 경우만을 가정한다. 앞에서 예로 제시한 Blast와 같이 데이타베이스의 모든 서열을 검색하지 않고 일부 데이타만을 선별하여 비교하는 프로그램에서는 본 프로그램 모델을 그대로 사용할 수 없다. 본 논문에서 제안하는 데이타베이스 검색 모델은 정확한 결과를 얻기 위해서 모든 데이타베이스의 데이타를 검색하는 서열 정렬 프로그램의 응답시간과 평균 시스템 비용을 줄일 수 있다.

2. 기존 프로그래밍 모델

FastA나 Blast등의 프로그램들은 웹을 통해 서비스를 하며 사용자가 서버에 접속하여 비교할 단백질 서열을 서버에 보낸다. 서버는 데이타베이스에서 서열을 검색하여 사용자가 요청한 서열과 비교한다.

2.1 일반적인 구조

생물정보학에서 사용하는 대부분의 서열 검색 프로그램들은 데이타베이스 기반으로 작동한다. 즉, 매번 사용자의 요청 마다 데이타베이스를 검색한 후 사용자의 요구에 응답을 해야 한다. 예를 들어 FastA의 경우 사용자는 비교 분석하고 싶은 서열을 FastA 서버에 전송한다. 전송된 사용자 서열은 데이타베이스에 저장되어 있는 각각의 서열과 비교되어 유사도를 검사하게 일정치 이상의 유사도를 갖는 서열을 사용자에게 돌려주게 된다. 이때, 서버는 모든 사용자 요청 각각에 대해서 데이타베이스 검색을 위해서 디스크를 접근한다.

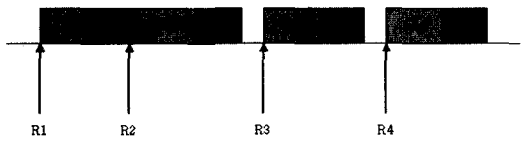

그림 1 기존의 프로그램 모델

그림 1에서 C_{DB}는 사용자 요청이 왔을 때 데이타베이스를 검색하기 위해서 디스크를 접근 하는 비용이다. 또한, C_{seq}는 데이타베이스로부터 검색된 모든 서열과 사용자가 요청한 서열을 비교 분석하는 비용이다. 즉, 하나의 사용자 요청 Rn(n = 1, 2, 3,)에 대해서 서버는 C_{DB} + C_{seq} 만큼의 비용이 드는 것이다. 일반적인 구조에서는 현재 처리되고 있는 요청이 없을 때 요청이 이루어지면 대기 시간 없이 바로 요청을 처리하게 되고 이미 다른 요청이 처리되고 있는 경우 새로 발생한 요청은 요청이 이루어진 순서대로 대기행렬(Queue)에 등록되게 된다. 그림 1에서 요청 R2는 R1이 처리되는 동안 발생하였기 때문에 R2는 큐에 등록되고 R1의 처리가 모두 끝나는 시점에 대기행렬(Queue)에서 제거되고 처리된다.

2.2 일반적인 구조의 비용

데이타베이스에서 한 블록을 검색할 때 소요되는 디스크 접근 시간을 C_{io}라 정의하고 서열 비교 대상이 되는 전체 데이타베이스내의 서열의 개수를 N_b라 정의한다. 사용자의 서열과 데이타베이스에서 검색 된 하나의 단백질 서열과 사용자가 요청한 단백질 서열간의 비교 시간 즉, 프로세싱 시간을 C_{cpu}로 정의한다.

데이타베이스는 하나의 단백질 서열을 받을 때 마다 데이타베이스의 모든 내용을 메모리로 가져와야 한다. 이때 걸리는 시간은 데이타베이스를 한번 검색하기 위해 디스크를 접근 하는 시간과 전체 데이타베이스 내 서열 개수의 곱과 같다. 한 블록을 읽어 들일 때의 시간은 모두 같다고 가정하면 디스크 접근 시간(C_{DB})를 아래와 같이 나타낼 수 있다.

$$C_{DB} = C_{io} \times N_b \quad (1)$$

이는 데이타베이스의 모든 서열을 검색하기 위해 디스크를 접근하는 시간이 된다. 식 (1)은 데이타베이스 검색을 위한 디스크 접근 시간이다. 그리고 각 서열간의 비교 시간은 사용자가 요청한 하나의 서열을 디스크에서 읽은 비교 대상이 되는 서열과 비교하는 시간(C_{seq})이 된다. 식 (2)는 모든 서열과 사용자 요청 서열을 비교하는 시간이다.

$$C_{seq} = C_{cpu} \times N_b \quad (2)$$

한 사용자가 서버에 접속하여 하나의 단백질 서열을 비교하는데 걸리는 평균 시간은 식 (1)과 식 (2)를 더한 시간이 된다.

$$\begin{aligned} C^o_{avg} &= C_{DB} + C_{seq} \\ &= C_{io} N_b + C_{cpu} N_b \\ &= (C_{io} + C_{cpu}) N_b \end{aligned} \quad (3)$$

2.3 일반적인 구조의 응답시간

사용자 요청은 발생률 λ의 포아송과정(Poisson process)이라 가정하였다. 서버가 다른 요청을 처리하고 있을 때, 다른 요청이 발생하면 새로운 요청은 큐에 등록된다. 요청들은 발생한 순서로 큐에 등록되고 등록된 순서로 순차적으로 서비스 된다. 모든 요청의 서비스 비용이 같다는 가정하며 M/G/1 큐잉 모델이 된다.

서비스 시간 $\frac{1}{\mu}$은 단일의 사용자 요청을 처리하는 시간과 같다. 즉, 서비스 시간 $\frac{1}{\mu}$은 하나의 요청이 서비스를 받는 평균 비용(C^o_{avg})이 된다. 서비스율 μ는 $\frac{1}{C^o_{avg}}$로 표시된다.

식 (4)는 사용자 요청의 발생률(λ)과 서비스율(μ)을 M/G/1 큐잉모델의 응답시간에 대입해본 결과이다.

$$W_n = \left(\frac{1}{C^o_{avg}}\right)^{-1} + \frac{\lambda \cdot \left(\frac{1}{C^o_{avg}}\right)^{-2}}{2\left(1 - \frac{\lambda}{\frac{1}{C^o_{avg}}}\right)} = C^o_{avg} + \frac{\lambda C^{o\ 2}_{avg}}{2(1 - \lambda \cdot C^o_{avg})} \quad (4)$$

3. 사용자 요청의 그룹핑

디스크 접근이 많은 VOD시스템에서 많은 사용자 요청을 효율적으로 처리하기 위한 방법으로 PDP 알고리즘을 사용한다. 이 방식의 VOD서버는 AT&T에서 현재 구현되어 있다[8]. PDP는 활동 그룹이라는 사용자 요청의 집합을 만들고 각 사용자 요청에 대해서 이미 플레이 된 영역을 캐시하고 다음 읽을 영역을 일정 시간 동안 저장하게 된다[8]. VOD에서 사용한 방식을 사용하여 데이타베이스의 검색 횟수를 줄이는 방법을 제안한다.

기존의 프로그램 모델의 문제점은 사용자의 요청마다 데이타베이스를 개별적으로 검색하여 같은 데이타베이스의 내용을 여러 번 읽게 된다는 것이다. 기존의 서열 검색 프로그램들은 서로 다른 사용자 요청을 처리하기 위해서 똑같은 데이타베이스에서 같은 데이타를 여러 번 접근하게 된다. 그룹핑은 같은 데이타가 여러 번 접근 되는 것을 줄여서 처리 비용과 응답시간을 향상시키는 것이다.

3.1 그룹핑의 구조

그룹핑은 매번 디스크 접근을 하지 않고 일정한 시간 동안 도착하는 사용자의 요청을 모아서 주기적으로 처리한다. 주기적으로 데이타베이스를 한번만 검색하고 데이타베이스로부터 검색된 서열들은 모아진 사용자 요청 서열과 각각 비교를 한다. 이럴 경우 데이타베이스를 모든 사용자 요청 때마다 검색할 필요가 없고 한 주기에 한번만 검색하므로 디스크 접근 횟수를 현저히 줄일 수 있다.

그림 2에서 R1, R2, R3, R4,..., Rn은 사용자 요청을 나타낸다. 사용자 요청은 큐에 저장되어 있다가 다음 주기에 처리된다. 사용자의 요청은 주기 D동안 그룹핑 되므로 다른 요청이 없어도 서비스를 받기까지 최대 D동안 지연되는 단점을 가진다. 그러나 주기 D동안 모아진 사용자 요청은 서비스 될 때 데이타베이스를 한번만 검색하고 이 데이타를 가지고 그룹핑 된 요청을 모두 처리하므로 사용자의 요청이 빈번한 경우 데이타베이스를 검색 하는 횟수를 현저히 절약할 수 있다.

그러나 그룹핑을 프로그램에 적용할 경우 모든 사용자 요청이 주기(D)가 끝난 후 처리되므로 사용자의 요청이 빈번하지 않을 경우에 기존의 방식(2장)에 비하여 응답시간이 느려질 수 있다. 또한, 이전 주기에서의 사

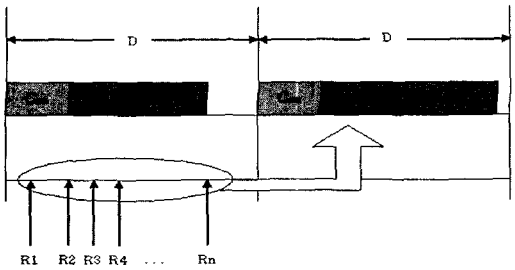

그림 2 그룹핑 모델

용자 요청이 많을 경우 처리시간이 이번 주기(D)보다 커질 경우 이번 주기(D)를 증가시켜야 한다. 그러나 본 논문에서는 주기(D)동안 저장되는 사용자 요청을 처리하는 전체 시간이 주기(D) 내에 이루어 질수 있도록 주기(D)를 충분히 길게 설정한다. 또한 주기(D)는 하나의 사용자 요청을 처리하는 시간보다는 길어야 한다. 이를 수식으로 표현하면 다음과 같다.

$$D \geq C^{o}_{avg} \tag{5}$$

3.2 그룹핑을 사용한 프로그램의 비용

일정 주기(D)동안 사용자의 요청을 한번에 처리할 때 소요되는 시스템 비용은 CPU시간과 디스크 접근 시간의 합으로 표시된다. 이를 수식으로 표현하면 아래와 같이 나타낼 수 있다. $D \cdot \lambda$는 주기 D동안 도착한 평균 사용자 요청의 수이다.

$$C^{g}_{total} = C_{DB} + D \cdot \lambda \times C_{seq} \tag{6}$$

즉, 그룹핑을 하는 프로그램 모델에서 한 사용자가 하나의 요청을 서비스하기 위한 시스템 비용은 다음과 같이 나타낼 수 있다.

$$C^{g}_{avg} = \frac{C_{DB}}{D \cdot \lambda} + C_{seq} \tag{7}$$

식 (7)은 주기(D)동안 사용자 요청이 있어야만 가능하다. 즉, 사용자 요청이 없을 경우 서버는 이를 검사하여 데이타베이스를 검색하지 않는다면 더 좋은 성능을 기대할 수 있다. 주기(D)동안 포아송 과정(poisson process)으로 발생하는 사용자 요청이 있을 확률은 $1 - e^{-\lambda D}$가 되는데 주기(D)동안 요청이 있을 경우만 데이타베이스를 검색하므로 그룹핑 방식에서 한 사용자가 하나의 요청을 서비스 받는데 드는 평균 비용은 식 (8)이 된다.

$$C^{g}_{avg} = \frac{C_{DB}}{D \cdot \lambda} \cdot (1 - e^{-\lambda D}) + C_{seq} \tag{8}$$

3.3 그룹핑을 사용한 프로그램의 응답시간

그룹핑을 사용하면 데이타베이스의 검색의 횟수를 줄일 수 있다는 장점이 있다. 사용자 요청이 주기(D)동안 $D \cdot \lambda$만큼 도착하게 된다. 기다리는 주기(D)동안 사용자 요청이 대기하는 시간의 평균은 D동안 발생하는 요청들이 지연되는 시간의 평균과 같다. 지연되는 값의 평균은 $\frac{D}{2}$가 된다.

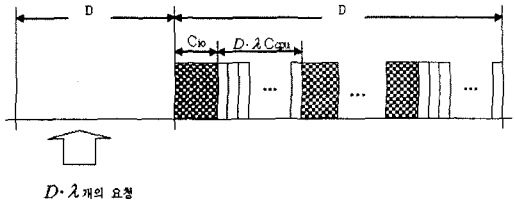

그림 3 그룹핑 프로그램 모델

그림 3에서 체크 보드 무늬는 데이타베이스에서 한 개의 서열 데이타를 검색하기 위해서 디스크를 접근 하는데 드는 시간이다. 그리고 데이타베이스로부터 얻은 하나의 서열을 모든 요청에 대해서 처리하는 시간은 $D \cdot \lambda C_{cpu}$로 나타낼 수 있다. 사용자 요청에 대해서 응답을 하는 시간은 D동안 발생한 $D \cdot \lambda$개의 요청이 모두 처리된 시간이 된다. 즉 평균 응답시간은 전 주기(D)동안 모인 사용자 요청이 대기하는 시간과 사용자 요청의 모든 처리가 끝난 시간의 합이 된다.

$$W_g = C_{DB} + D \cdot \lambda \cdot C_{seq} + \frac{D}{2} \tag{9}$$

4. 카풀방식

3장에서 설명한 그룹핑 방식은 사용자 요청을 그룹핑하여 여러 요청을 한번에 처리함으로써 데이타베이스를 검색하는 횟수를 줄일 수 있었다. 그러나 그룹핑을 함으로써 다음 주기까지 지연되는 시간이 응답시간이 길어질 수 있다. 생물정보학의 서열정렬에서는 사용자의 요청 마다 서열 데이타베이스의 모든 데이타를 전부 탐색하는 것이 일반적이다. 또한 이렇게 탐색 되는 데이타는 순서와 상관이 없이 모든 데이타를 처리하면 된다. 이는 일반 데이타베이스 검색과 같이 사용자마다 서로 다른 특정 부분의 데이타를 검색하는 응용과는 다르다. 그래서 그룹핑을 수행하지 않고 데이타베이스의 한 블록을 검색할 때마다 도착된 요청을 즉시 수용하여 검색된 데이타를 즉시 사용할 수 있는 방법을 제안한다.

4.1 카풀방식의 구조

기존의 서열 검색 프로그램이 하나의 요청을 처리할 때, 하나의 서열을 데이타베이스에서 메모리로 로딩한 후 이를 사용자 요청 서열과 비교 분석하고 분석이 끝나면 다시 하나의 서열을 데이타베이스로부터 검색 한다. 요청에 대한 서비스 중 다른 요청이 오면, 2장에서 설명한 기존의 방식은 이를 큐에 등록하고 현재의 서비스가 종료된 후 새로운 요청을 처리하였다. 3장에서 설명한

그룹핑 방식에서는 사용자의 요청마다 독립적인 데이타 검색을 위한 디스크 접근에 따르는 비용을 줄이기 위하여 주기적으로 사용자의 요청을 모아 동시에 처리하였다. 그러나 그룹핑 방식은 막 도착된 사용자의 요청을 바로 처리하지 않고, 한 주기가 끝난 후 모아 처리하기 때문에 대기 시간이 발생한다. 본 장에서 설명하는 카플 방식에서는 사용자의 요청을 큐에 등록하지 않으며 현재 처리되고 있는 요청과 함께 처리한다. 즉, 한번 데이타베이스를 검색하여 하나의 서열을 로딩하면 원래 서비스되고 있었던 요청과 함께 비교 분석하고 새로 들어온 요청 역시 함께 비교 분석을 하게 된다. 그리고 다음 서열을 데이타베이스로부터 검색하게 된다. 생물정보학의 데이타베이스는 각각의 데이타들 간에 의존성이 없으므로 데이타베이스의 데이타들이 처리되는 순서와 상관없이 모든 데이타베이스의 데이타를 처리할 수 있다.

그림 4는 서열이 4개 들어 있는 데이타베이스를 사용자 요청 R1에 의해서 처리되고 있을 때 다른 요청 R2와 R3가 들어온 것을 나타낸다.

그림 5는 사용자 요청(Rn, n=1, 2, 3, 4)이 왔을 때 모든 요청에 대한 처리가 끝날 때까지 카플방식에서 서비스 하는 방식을 보여준다. 여기서 D(i)는 i번째 데이타베이스의 서열을 처리하는데 소비되는 비용이며 데이타베이스는 그림 4에서와 마찬가지로 4개의 서열만을 갖는다고 가정한다. 그리고 R(I,j)는 i번째 사용자 요청

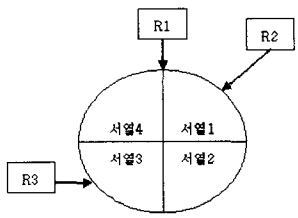

그림 4 데이타 처리와 사용자 요청

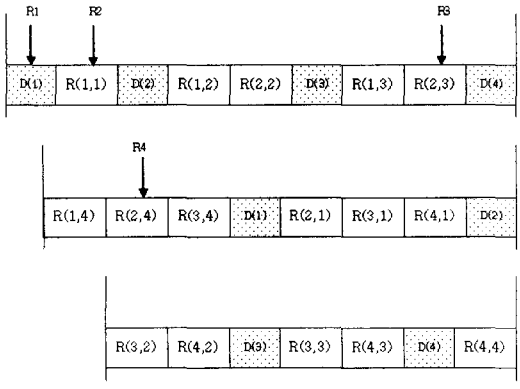

그림 5 카플의 서비스 방식

을 처리하기 위해서 j번째 데이타베이스 서열과 비교하는데 소비되는 비용을 나타낸다.

첫 번째 서열(D(1))을 읽고 요청 R1을 위한 서비스(R(1,1))를 하게 된다. 처리 동안 요청 R2가 발생하여 두 번째 서열(D(2))를 읽고 R1을 위한 서비스(R(1,2))와 R2를 위한 서비스(R2(2))를 하게 된다. 즉, 데이타베이스는 한번 검색하여 여러 요청에 대해서 처리를 하므로 그룹핑과 같이 데이타베이스 검색을 위한 디스크 접근 횟수를 줄일 수 있다. 요청 R1은 4번째 서열(D(4))까지 읽은 후 서비스를 마치게 된다. 4번째 서열 (D4)까지 읽은 후 요청 R2는 첫 번째 서열(D1)을 위한 처리를 하지 않았으므로 요청 R3와 함께 첫 번째 서열(D1)을 읽고 처리하는 루틴을 실행하게 된다. 즉, 카플방식의 사용자 요청은 모든 데이타베이스를 검색할 때까지 처리하게 되지만 전에 처리되고 있던 사용자 요청에 대한 처리가 끝날 때까지 새로운 요청이 지연되지 않는다는 장점이 있다. 이는 생물정보학의 데이타베이스의 데이타를 읽는 순서가 결과에 미치는 영향이 없기 때문에 가능하다.

카플 방식은 그룹핑 기법과 같이 데이타베이스의 접근 횟수를 줄여 비용 면에서 이득을 얻을 수 있다. 또한 사용자 요청을 지연시킬 필요가 없으므로 더 좋은 응답 시간을 갖게 될 것이다.

4.2 카플방식을 사용한 프로그램의 비용

카플 방식의 전체 비용은 데이타베이스의 시작점부터 검색을 시작하여 다시 시작점으로 돌아오기 전까지의 처리시간을 기준으로 구할 수 있다. 사용자 요청의 발생율을 λ라 가정하고 데이타베이스를 모두 검색 하는 동안 소요되는 디스크 접근 시간과 도착한 사용자 요청과 비교하는 시간의 합을 C_{total}^{cp}라 하자. 이주기(C_{total}^{cp})동안 발생하는 평균 요청의 수는 $\lambda \cdot C_{total}^{cp}$가 된다. 주기(C_{total}^{cp})동안 데이타베이스는 한번만 검색한다. 주기동안의 총 비용을 구해보면 다음과 같다.

$$C_{total}^{cp} = C_{DB} + \lambda \cdot C_{total}^{cp} \cdot C_{seq} \quad (10)$$

식 (10)을 C_{total}^{cp}에 대해서 정리하면 식 (11)과 같다.

$$C_{total}^{cp} = \frac{C_{DB}}{1 - \lambda \cdot C_{seq}} \quad (11)$$

하나의 사용자 요청을 처리할 때 드는 비용은 다음과 같이 나타낼 수 있다. 그런데 하나의 요청도 발생하지 않는 경우를 고려해서 하나의 사용자 요청을 처리하는데 드는 시스템 비용을 구해보자.

$$C_{avg}^{cp} = \frac{C_{DB}}{C_{total}^{cp} \cdot \lambda} \cdot \left(1 - e^{-\lambda \cdot \frac{C_{DB}}{1 - \lambda \cdot C_{seq}}}\right) + C_{seq} \quad (12)$$

식 (12)는 식 (10)을 사용자 요청의 수로 나눈 값, 즉 하나의 사용자 요청이 처리되는 비용을 구한 것이고 이

때 한 주기에 사용자 요청이 발생할 확률 $1-e^{-\lambda C_{total}^{cp}}$를 고려한 식이다. 식 (12)를 정리하면 다음과 같다.

$$C_{avg}^{cp} = (\frac{1}{\lambda} - C_{seq})(1 - e^{-\lambda \cdot \frac{C_{DB}}{1-\lambda C_{seq}}}) + C_{seq} \quad (13)$$

4.3 카플방식을 사용한 프로그램의 응답시간

카플 방식에서 하나의 요청이 서비스 완료될 시각은 모든 데이타베이스를 검색하고 이에 대해 처리가 모두 끝난 시간이다. 각각 읽혀진 데이타베이스의 서열은 동시에 처리되고 있는 다른 사용자 요청을 위해서도 사용되며 다른 사용자 요청을 처리하는 동안 다른 요청들은 기다리게 된다. 하나의 사용자 요청이 모든 데이타베이스를 검색 하는 동안 등록되어 있는 모든 사용자 요청이 동시에 처리된다. 즉, 카플 방식의 응답시간은 하나의 사용자 요청을 처리하는 시간과 이를 처리하는 시간(C_{DB} + C_{seq}) 동안 같이 처리되는 사용자 요청의 처리 시간($\lambda \cdot W_{cp} \cdot C_{seq}$)의 합으로 나타낼 수 있다.

$$W_{cp} = C_{DB} + C_{seq} + \lambda \cdot W_{cp} \cdot C_{seq} \quad (14)$$

식 (14)를 정리하면 다음과 같다.

$$W_{cp} = \frac{C_{DB} + C_{seq}}{1 - \lambda \cdot C_{seq}} \quad (15)$$

5. 성능 비교

5.1 임계 값

각 방식은 사용자 요청의 도착률 λ에 대해서 임계 값을 갖는다. 도착률 λ의 임계 값은 서버의 사용률이 1보다 적은 조건을 만족하는 최대의 도착률 λ를 나타낸다. 서버의 사용률은 사용자 요청의 도착률을 서버가 각 방식을 사용해서 하나의 사용자 요청을 처리하는 평균 비용의 곱으로 나타낼 수 있다. 이값이 1보다 작을 경우 서비스가 가능한 것이다. 물론 데이타베이스 사용과 CPU사용을 동시에 할 수 있는 기법을 이용하면 서버의 사용율을 1보다 높일 수 있다. 서열비교를 순차적으로 수행하는 방식을 가정하였다. 각 방식에서 임계값을 구해보자.

• 기존 방식

기존의 방식에서 하나의 사용자 요청을 처리하는 평균 비용은 식 (3)에서와 같이 나타낼 수 있다. 이를 식으로 나타내면 $\lambda \cdot C_{avg}^{o} < 1$와 같다. 즉, 기존의 방식에서 λ이 임계 값은 다음과 같이 나타낼 수 있다.

$$\lambda \le \frac{1}{C_{seq} + C_{DB}} \quad (16)$$

• 그룹핑 방식

그룹핑 방식에서 평균비용은 식 (7)과 같다. 같은 방법으로 λ의 임계 값을 얻을 수 있다.

$$\lambda \le \frac{1 - C_{DB}/D}{C_{seq}} \quad (17)$$

• 카플 방식

카플 방식에서 서버의 사용율은 $\lambda \cdot \frac{1}{\lambda}$이 되므로 항상 조건을 만족한다. 또한 식 (11)에서 C_{total}^{cb}는 양수이고 C_{DB} 또한 양수이다. 즉, $1 - \lambda \cdot C_{seq} > 0$을 만족해야 한다. 이를 풀면 아래와 같이 λ의 임계 값을 얻을 수 있다.

$$\lambda < \frac{1}{C_{seq}} \quad (18)$$

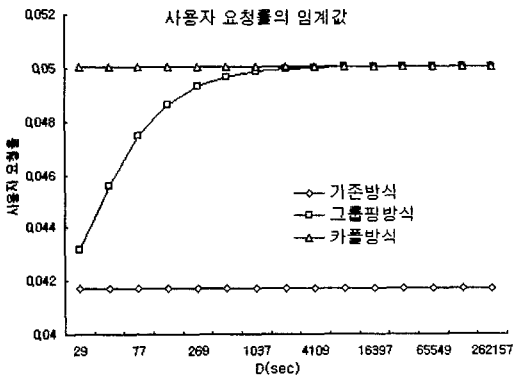

그림 6 사용자 요청률의 임계값

그림 6은 각 방식의 임계 값을 나타내는 식 (16), (17), (18)으로부터 나온 결과이다. 각 파라미터는 생물정보학의 서열 정렬 프로그램에서 실제적으로 사용되고 있는 데이타베이스 Genbank(Protein Sequence Database of Rip International Release 72.02)를 사용하여 측정한 결과를 이용하였다. GenBank는1981년 미국립보건원으로부터 지원을 받아 로스 알라모스 연구소가 이를 관리하다가 1992년 미국립보건원의 국립의학도서관산하 미국립생물공학정보센터(NCBI)로 이전되서 관리되는 서열정보 데이타베이스이다. 본 논문에서 사용한 GENBANK는 전체 283,177개의 서열을 가지고 있으며 전체가 96,101,346라인으로 이루어져 있는 텍스트 형 데이타베이스를 사용하였다. 사용자가 요청하는 서열은 인간(human)단백질 중 세포의 산화 환원에 작용하는 색소 단백질(cytochrome)을 사용하였다. 그 결과 C_{DB}는 3.99 sec.가 되고, 데이타베이스의 모든 서열과 사용자 요청 서열을 모두 비교하는 비용, C_{seq}는 19.98 sec.가 되었다. 그룹핑을 위한 주기를 증가 시키고 각 방식의 임계값의 변화를 살펴 보았다. D가 증가할수록 그룹핑의 사용자 요청률 임계값이 증가하여 카플 방식의 임계값으로 수렴하는 것을 볼 수 있다. 같은 성능의 하드웨

어에서 가장 많은 사용자를 받을 수 있는 방식은 카플 방식임을 알 수 있다.

5.2 비용 비교

그림 7은 기존의 방식과 그룹핑 방식 그리고 카플 방식의 평균비용의 비교를 위해서 식 (3), 식 (8) 그리고 식 (13)의 결과를 이용하였다. 그룹핑 방식에서 주기는 50 sec.와 70 sec.으로 설정하였다.

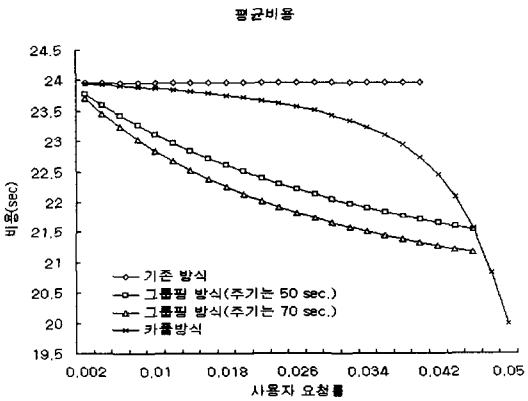

그림 7 시스템 비용 비교

그림 7은 각 방식의 사용자당 시스템 비용을 비교한 그래프이다. 각 방식은 임계값까지 그래프에 표시하였다. 그림 7에서 x축은 사용자 요청률 λ이고 y축은 사용자당 시스템 비용을 나타낸다. 기존의 방식을 제외한 그룹핑 방식과 카플 방식은 λ값이 증가할수록 시스템 비용이 줄어드는 것을 볼 수 있다. 또한 그룹핑 방식은 D에 따라 시스템 비용을 많이 감소 시킬 수 있는 것을 볼 수 있다. 주기(D)를 70sec로 설정한 경우 시스템 비용이 더 낮아진 것을 볼 수 있다. 그리고 주기(D)의 값에 따라서 그룹핑 방식이 받을 수 있는 사용자 요청의 임계 값이 결정된다는 것을 볼 수 있다. 카플 방식 역시 λ값이 증가할수록 시스템 비용이 줄어드는 것을 알 수 있다. 그런데 카플방식이 그룹핑 방식보다 사용자당 시스템 비용이 큰 이유는 사용자 요청률 λ가 작을 때 상대적인 데이타베이스 검색 비용이 크기 때문이다. 사용자 요청률 λ가 커질수록 상대적인 데이타베이스 검색 비용이 줄어들기 때문에 사용자당 비용이 급격히 감소하게 된다.

5.3 응답시간 비교

그림 8은 각 방식의 응답시간을 비교한 그래프이다. 그림 8에서 x축은 사용자 요청률 λ를 나타내고 y축은 임의의 요청에 대한 프로그램의 평균 응답시간을 나타낸다. 기존의 방식은 임계 점에 가까워 질수록 응답시간이 급격히 증가하였다. 카플방식이 가장 짧은 응답시간

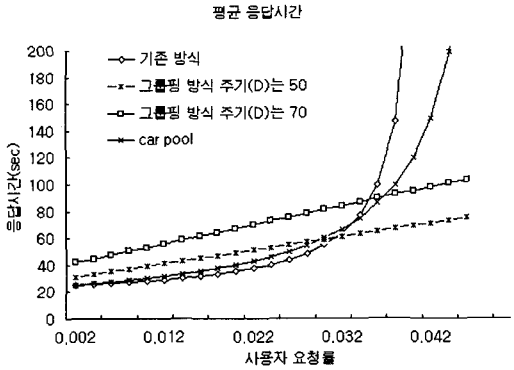

그림 8 응답 시간 비교

을 보였다. 이는 사용자가 적을 때는 데이타베이스를 즉시 읽기 때문에 응답이 빠를 수 있는 것이다. 또한, 기존 방식보다 데이타베이스를 검색 하는 횟수가 적기 때문에 응답시간이 기존방식보다 더 좋은 것이다. 반면에 그룹핑 방식은 사용자 요청을 그 다음 주기에 모아서 처리는 지연 시간이 있기 때문에 응답시간이 느리게 된다. 하지만 기존방식은 사용자 요청률 λ가 커질수록 대기행렬(Queue)에서 대기하는 사용자요청의 수가 많아지므로 응답시간이 급격히 증가한다. 그림 8에서 그룹핑 방식의 주기가 70일 때 사용자 요청률이 0.058보다 클 때와 주기 50일 때 사용자 요청률이 0.052보다 클 때 카플방식보다 응답시간이 좋은 것은 그룹핑 방식의 한 주기 동안에 처리할 수 없는 수의 사용자 요청이 들어올 경우를 가정하지 않았기 때문이다.

6. 시뮬레이션 결과

파라매터 실측을 위해서 fasta34t11버전을 펜티엄4 1.6G Hz, 256Mbyte Ram(pc2700) 사양에서 인간(human)단백질 중 세포의 산화 환원에 작용하는 색소 단백질(cytochrome)을 데이타베이스로Genbank(Release 72.02)를 사용하여 비교 분석하는데 소요되는 비용, 즉 데이타베이스를 검색하기 위해 디스크를 접근하는 비용(C_{DB})는 3.99 sec.가 되고, 데이타베이스의 모든 서열과 사용자 요청 서열을 모두 비교하는 비용, C_{seq}는 19.98 sec.이 되었다. 데이타베이스로는 Genbank (Release 72.02)를 사용하였다.

표 1은 각 방식에 따라서 fasta프로그램을 시뮬레이션 했을 때 나타나는 결과이다. 사용자 요청은 포아송 분포로 이루어진다고 가정하였다. 기존방식은 각 요청률(λ)에 따라 500개의 사용자 요청을 처리하는 평균시간을 측정하였으며 그룹핑 방식은 그룹핑 주기(D)를 50, 70, 100으로 하여 각 요청률(λ)마다 2000개의 사용자 요청을 처리하는데 걸리는 응답시간의 평균을 구한 것이다.

표 1 처리 비용

람다값	기존	그룹핑 방식 주기(D)는 50	그룹핑 방식 주기(D)는 70	카플방식
0.02	36.4699	61.6255	81.0800	39.9890
0.025	44.7755	64.5305	85.0592	48.0905
0.03	66.0756	68.9833	90.9914	60.3532
0.035	150.6701	72.7457	96.9161	80.9072
0.04	416.5828	76.8546	106.7105	121.3321
0.045	882.2224	83.6668	122.5694	244.8262
0.05	1303.3760	91.1893	143.8982	7656.7200
0.055	1704.3970	103.7128	190.3468	130092.2000

한 주기의 처리할 수 있는 수 이상의 사용자 요청이 발생했을 경우 다음 주기에 처리하게 했다. 그리고 카플 방식은 각 요청률(λ)마다 60000개의 사용자 요청을 처리하는데 걸리는 평균시간을 구한 값이다.

표 1은 각 방식에 따라서 fasta프로그램을 시뮬레이션 했을 때 나타나는 결과이다. 사용자 요청은 포아송 분포로 이루어진다고 가정하였다. 기존방식은 각 요청률(λ)에 따라 500개의 사용자 요청을 처리하는 평균시간을 측정하였으며 그룹핑 방식은 그룹핑 주기(D)를 50, 70, 100으로 하여 각 요청률(λ)마다 2000개의 사용자 요청을 처리하는데 걸리는 응답시간의 평균을 구한 것이다. 한 주기의 처리할 수 있는 수 이상의 사용자 요청이 발생했을 경우 다음 주기에 처리하게 했다. 그리고 카플 방식은 각 요청률(λ)마다 60000개의 사용자 요청을 처리하는데 걸리는 평균시간을 구한 값이다.

그림 9는 표 1을 바탕으로 그래프를 그린 것이다. 기존의 방식은 많은 응답시간이 사용자 요청률이 증가할수록 급격히 증가하는 것을 알 수 있다. 또한 그룹핑 방식은 요청률(λ)이 증가하여도 가장 안정적인 응답시간

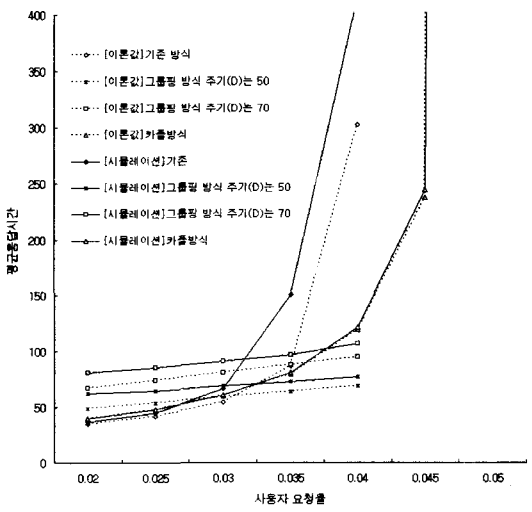

그림 9 응답시간 시뮬레이션

을 보였다. 또한, 그룹핑 방식은 사용자 요청을 그룹핑하는 시간이 응답시간에 큰 영향을 미치는 것을 볼 수 있다. 하지만 그룹핑 방식은 람다값이 증가함에 따라 해당 주기에 처리하지 못하는 사용자 요청이 많아졌다. 이럴 경우 해당 사용자 요청에 대한 응답시간이 길어지게 되는 단점이 있지만 가장 안정적인 서비스를 제공할 수 있다. 카플방식은 사용자 요청률이 작을 때는 응답시간이 짧아지고 사용자 요청이 많을 때는 데이타베이스의 데이타를 사용자 요청이 공유하여 사용할 수 있어서 기존 방식보다 좋은 응답시간을 가졌다. 또한 그룹핑 방식에서는 늘어나는 사용자 요청에 의해서 응답시간이 한 주기를 넘는 사용자 요청이 생기게 되어 사용자 요청에 대한 응답시간이 고루지 않을 수 있지만 카플방식은 사용자 요청에 대한 응답시간이 고루게 분포하여 공평한 서비스를 제공할 수 있다.

7. 결 론

단백질 데이타베이스는 매우 빠르게 증가 하고 있다. 단백질 데이타의 폭발적인 증가는 컴퓨터의 발전 속도를 능가할 정도이며 또한 이를 분석하기 위해서도 데이타베이스의 빈번한 검색은 생물정보학 관련 문제들을 처리함에 있어 과부하로 작용할 것이다. 본 논문에서는 생물정보학의 데이타베이스를 효율적으로 사용할 수 있는 서열 검색 프로그램 모델을 제시하였다. 그룹핑 방식은 각 주기 마다 사용자 요청을 처리 하는 방식이다. 한 주기 동안 모아진 사용자 요청은 다음 주기에 한번에 처리되는데 이때 모아진 사용자 요청을 위해서 단 한번의 데이타베이스 검색을 하므로 시스템 비용을 줄일 수 있다. 또한 적은 시스템 비용을 가지므로 기존 방식에 비해서 많은 사용자 요청을 처리할 수 있다. 그리고 카플 방식은 그룹핑 방식의 단점인 사용자 요청을 모으는 대기 시간을 없애고, 데이타베이스를 한번만 검색하여 현재 처리되는 모든 사용자 요청과 함께 처리한다. 이 방식은 사용자 요청을 모으는 시간이 없고, 또한 현재 처리되는 모든 사용자 요청에 대해서 한번의 데이타베이스 검색을 하므로 평균 시스템 비용이 적어져 짧은

응답시간을 갖게 된다. 또한, 사용자 수가 적을 때는 많은 데이타베이스의 검색을 하게 되어 그룹핑 방식보다 시스템 비용이 많이 들지만 기존의 방식에 비해서는 여전히 좋은 성능을 갖는다. 그룹핑 기법은 가장 적은 시스템 비용이 소비되지만 사용자 요청이 많은 경우 응답시간이 고르지 못하게 된다. 카플 방식은 시스템 비용은 그룹핑 방식보다 많지만 안정적이고 공평한 응답시간을 갖는다. 또한, 카플 방식은 다른 방식에 비해서 같은 하드웨어 상에서 받아들일 수 있는 사용자 도착율의 임계값이 가장 높아서 가장 많은 사용자에게 서비스를 제공할 수 있다.

그룹핑 방식과 카플 방식은 데이타베이스의 검색이 많고, 데이타의 갱신이 거의 없는 데이타베이스를 사용하는 응용프로그램에서 사용될 수 있다. 특히, 검색의 결과가 데이타베이스의 검색 시작위치와 관련 없는 독립적인 데이타를 검색하는 저작권 검색, 특허권 검색, 논문 검색, 웹 데이타베이스 검색, 등 많은 응용에서 그룹핑 방식과 카플 방식을 사용하여 서버의 부하를 줄이고, 빠른 응답시간으로 서비스 할 수 있다. 생물정보학에서 사용되는 데이타베이스와 비슷한 특징을 갖는 많은 응용에서 그룹핑 방식과 카플 방식을 사용함으로써 가능한 많은 사용자에게 빠른 응답시간을 갖는 서비스를 제공할 수 있을 것이라 기대된다.

참 고 문 헌

[1] Hyun Seok Park, "포스트지놈 시대 생물정보학(Bioinformatics)의 역할", 대학내분비학회지, 제16권, 제1호, pp. 1-8, 2001.

[2] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman, "Basic Local Alignment Search Tool," J. Mol. Biol., pp. 403-410, 1990.

[3] Tieng K. Yap, Ophir Frieder, and Robert L. Martino, "Parallel Homologous Sequence Searching in Large Databases," IEEE Proceedings of the Fifth Symposium on the Frontiers of Massively Parallel Computation(Frontiers'95), 1995.

[4] Jerry Banks, John S. Carson, and Barry L. Nelson "Discrete-Event System Simulation," Prentice Hall, 2000.

[5] SWISSPROT: http://www.expasy.org/

[6] GENBANK : http://www.ncbi.nlm.nih.gov/Genbank/

[7] EMBL : http://www.ncbi.nlm.nih.gov/Genbank/

[8] Ozden, B., R. Rastogi, A. Silberschatz, and C. Martin "Demand Paging for Video on Demand Servers," IEEE International Conference on Multimedia Computing and Systems, May 1995.

김 민 준

2002년 아주대학교 학사. 2004년 아주대학교 석사. 2004년~현재 미디어코러스(주) 부설정보통신연구소

임 성 화

1999년 아주대학교 정보 및 컴퓨터 공학부 졸업(학사). 2001년 아주대학교 정보통신전문대학원 졸업(석사). 2003년 아주대학교 정보통신전문대학원 박사과정 수료. 2003년~현재 현대디지탈테크㈜ 미디어 연구소

김 재 훈

1984년 서울대학교 학사. 1984년~1991년 대우통신 종합연구소. 1988년 Tolerant System 연구소. 1993년 인디애나대학교(미국) 석사. 1997년 텍사스 A&M대학교(미국) 박사. 1998년 삼성전자 컴퓨터시스템개발팀. 1998년~현재 아주대학교 정보통신전문대학원 부교수

이 원 태

1982년 서울대학교 학사. 1984년 서울대학교 석사. 1992년 Univ. of Alabama at B'ham, Ph.D (biophysics). 1992년~1994년 Univ. of Toronto, Canada, Dep. of Structural Biology, Post doc./Research Associate. 1994년~1997년 LG 바이오텍 연구소 선임연구원. 1997년~1998년 연세대학교 생화학과 조교수. 1998년~현재 연세대학교 생화학과 부교수

정 진 원

1998년 연세대학교 학사. 2000년 연세대학교 석사. 2000년~현재 연세대학교 박사과정(생화학)