그룹핑과 카풀 방식을 이용한 효율적인 생물정보 DB접근*

김민준°, 김재훈
아주대학교 정보통신전문대학원
xwind@dmc.ajou.ac.kr,jaikim@ajou.ac.kr

Effective Bioinformatics Database Access using Grouping and Carpool

Min Jun Kim°, Jai-Hoon Kim
Graduate School of Information and Communication Ajou University

요 약

생물정보학 관련 프로그램들은 데이터베이스로부터 유전자 등의 데이터를 검색하고 처리한다. 이때 각각 클라이언트의 요청마다 매번 데이터베이스의 검색을 수행한다면 많은 시간이 걸리게 된다. 또한 서버에 과부하를 초래하여 응답시간이 길어 질 수 있다. 본 논문에서는 사용자 요청을 그룹핑 하여 데이터베이스 액세스 시 사용도를 높이는 방법과, 사용자 요청을 대기 시간 없이 바로 처리하여 데이터베이스 액세스를 공유하여 시스템 사용도를 높이고 빠른 응답시간을 가지는 카풀 방식을 제안한다.

1. 서론

최근 인간 유전자 프로젝트의 성공적인 수행은 모든 생명과학 분야의 급속한 발전을 야기시켰으며, 이러한 인간유전체 지도의 완성으로 전개되는 유전자이후시대(post Genom)에는 인간의 모든 유전자와 유전자의 발현으로 생성되는 단백질들의 구조와 기능에 관한 연구가 활발히 수행될 것이다. 이런 연구가 진행됨은 궁극적으로 막대한 디지털 정보의 축적을 의미한다. 인간은 A, T, G, C라는 네 개의 문자로 표현된 30억 개의 정보를 저장하고 있다. 웹을 통해 공개된 생물학 관련 데이터베이스는 SwissProt, GenBank, EMBL등 헤아릴 수 없을 정도로 많다.

본 논문에서는 생물정보학에서 사용되는 데이터베이스는 일반 데이터베이스와는 달리 전체 검색이 많으며 데이터를 액세스할 때 그 순서는 결과에 영향을 미치지 않는 특징이 있다. 이러한 특징에 맞춰 이를 이용하는 프로그램들의 성능을 증가시키기 위한 프로그램 모델을 제안한다. 우선 사용자 요청을 모아서 한번에 처리하는 사용자 요청 *그룹핑 기법*은 일정 주기 동안 사용자 요청을 모아서 데이터베이스를 한번만 액세스하고 여러 번의 처리를 하게 되어 데이터베이스의 액세스 횟수를 줄여 응답시간과 시스템 비용을 줄일 수 있었다. 또한 *카풀방식*은 그룹핑 방식에서 사용자 요청을 그룹핑하기 위해서 지연되는 시간 없이 사용자 가 요청을 하면 전에 처리하던 요청을 끝날 때까지 기다리지 않고 같이 처리함으로써 데이터베이스를 한번만 액세스하게 된다. 즉, 사용자 요청 하나당 데이터베이스의 액세스 횟수를 줄여서 시스템 비용과 응답시간에서 이익을 볼 수 있는 것이다.

2. 기존 프로그래밍 모델

FastA나 Blast등의 프로그램들은 웹을 통해 서비스를 하며 사용자가 서버에 접속하여 비교할 단백질 서열을 서버에 보낸다. 서버는 데이터베이스에서 서열을 읽어 들여 사용자가 요청한 서열과 비교한다.

2.1 일반적인 구조

생물정보학에서 사용하는 대부분의 프로그램들은 데이터베이스 기반으로 작동한다. 즉, 매번 사용자의 요청 마다 데이터베이스에 접근하여 데이터를 읽은 후 사용자의 요구에 응답을 해야 한다.

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

2.2 일반적인 구조의 비용과 응답시간

사용자 서열과 데이터베이스에서 읽어 들인 모든 단백질 서열과 사용자가 요청한 단백질 서열간의 비교 시간 즉, 프로세싱 시간을 C_{seq} 로 정의한다. 또 사용자 요청의 서열과 데이터베이스 서열을 비교하기 위해서 데이터베이스를 액세스 하는 시간을 C_{DB} 라 한다.

한 사용자가 서버에 접속하여 하나의 단백질 서열을 비교하는데 걸리는 평균 시간은 단백질 서열간의 비교시간과 전체 데이터베이스를 액세스하는 시간의 함과 같다.

$$C^{o}_{avg} = C_{DB} + C_{seq} \quad \text{식(1)}$$

기존 방식의 응답시간은 사용자 요청이 λ 의 포아송과정(Poisson process)이라 가정할 때, 요청이 발생한 순서로 대기행렬에 등록되고 순서대로 처리될 때 M/G/1 큐잉 모델이 된다. 이때 서비스 율은 하나의 사용자 요청을 처리하는 비용($\frac{1}{C^{o}_{arg}}$)으로 나타낼 수 있다. 사용자 요청율과 처리율을 M/G/1 큐잉모델의 응답시간 수식을 이용하여 식 (2)를 구할

본 연구는 정보통신부 정보통신선도기반기술개발사업의 지원에 의하여 이루어진것임.

할 수 있다.

$$w_o = C^o_{avg} + \frac{\lambda C^{o\ 2}_{avg}}{2(1-\lambda \cdot C^o_{avg})} \quad \text{식(2)}$$

3. 사용자 그룹핑 요청[4]

기존의 프로그램 모델의 문제점은 사용자의 요청마다 데이터베이스를 개별적으로 액세스하여 같은 데이터베이스의 내용을 여러 번 읽게 되다는 것이다. 생물정보학의 프로그램들은 서로 다른 사용자 요청을 처리하기 위해서 똑 같은 데이터베이스에서 같은 데이터를 여러 번 액세스하게 된다.

3.1 그룹핑의 구조

그룹핑 방식은 주기(D)동안 사용자 요청을 모으고 다음 주기에 모아진 사용자 요청을 한번에 처리한다. 이때, 데이터베이스는 한번만 액세스하게 된다. 사용자의 요청은 주기 D동안 그룹핑 되므로 다른 요청이 없어도 서비스를 받기까지 최대 D동안 지연되는 단점을 가진다. 하지만 주기 동안 모아진 사용자 요청을 한번에 처리할 때 데이터베이스를 한번만 액세스 하므로 시스템 비용을 절약할 수 있다.

3.2 그룹핑을 사용한 프로그램의 비용 및 응답 시간

사용자 요청이 λ의 포아송과정 일 때 주기(D)동안 도착한 평균 사용자 요청의 수는 $D \cdot \lambda$이다. 이때 주기(D)동안 시스템 비용은 다음과 같다.

$$C^g_{total} = C_{DB} + D \cdot \lambda \times C_{seq} \quad \text{식(3)}$$

이때, 하나의 사용자를 처리할 때 비용은 식(4)와 같다.

$$C^g_{avg} = \frac{C_{DB}}{D \cdot \lambda} + C_{seq} \quad \text{식(4)}$$

주기(D)동안 사용자 요청이 없을 확률은 $1-e^{-\lambda D}$ 이므로 이를 식 (3)에 적용하면 다음과 같다.

$$C^g_{avg} = \frac{C_{DB}}{D \cdot \lambda} \cdot \left(1-e^{-\lambda D}\right) + C_{seq} \quad \text{식(5)}$$

그룹핑 방식에서 응답시간의 평균은 주기(D)동안 모아진 사 용자 요청을 처리하는 시간($C_{DB} + D \cdot \lambda \cdot C_{seq}$)과 사용자 요청을 그룹핑 하는 동안 지연되는 시간의 평균 ($\frac{D}{2}$)의 합이 된다.

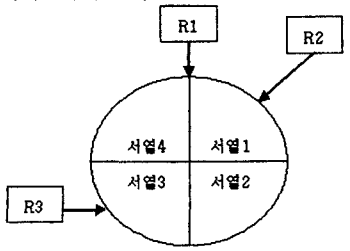
$$w_g = C_{DB} + D \cdot \lambda \cdot C_{seq} + \frac{D}{2} \quad \text{식(6)}$$

4. 카플 방식

그룹핑방식은 사용자 요청을 모아서 다음 주기에 처리하므로 그룹핑 하는데 지연되는 시간이 응답시간을 길어지게 한다. 생물정보학의 서열정렬에서는 사용자의 요청 마다 서열 데이터베이스의 모든 데이터를 전부 탐색하는 것이 일반적이다. 또한 이렇게 액세스 되는 데이터는 순서와 상관이 없이 모든 데이터를 액세스하면 되는 것이다. 그래서 그룹핑을 수행하지 않고 데이터베이스의 한 블록을 액세스할 때마다 도착된 요청을 즉시 수용하여 검색된 데이터를 즉시 사용할 수 있는 방법을 제안한다.

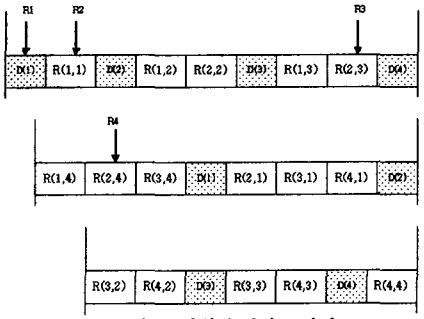
4.1 카플 방식의 구조

카플 방식에서는 사용자의 요청을 큐에 등록하지 않으며 현재 처리되고 있는 요청과 함께 처리한다. 즉, 한번 데이터베이스를 액세스하여 하나의 서열을 로딩하면 원래 서비스되고 있었던 요청과 함께 비교 분석하고 새로 들어온 요청 역시 함께 비교 분석을 하게 된다. 그리고 다음 서열을 데이터베이스로부터 액세스하게 된다.



[그림 1] 데이터처리와 사용자 요청

그림 1은 서열이 4개 들어 있는 데이터베이스를 사용자 요청 R1에 의해서 처리되고 있을 때 다른 요청 R2와 R3가 들어온 것을 나타낸다.



[그림 2] 카플의 서비스 방식

그림 2는 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 데이터베이스의 서열을 액세스하는데 소비되는 비용이며 데이터베이스는 그림 1에서와 마찬가지로 4개의 서열만을 갖는다고 가정한다. 그리고 R(i,j)는 i번째 사용자 요청을 처리하기 위해서 j번째 데이터베이스 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, 데이터베이스는 한번 액세스하여 여러 요청에 대해서 처리를 하므로 그룹핑과 같이 데이터베이스 액세스 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 데이터베이스를 액세스할 때까지 처리하게 되지만 전에 처리되고 있던 사용자 요청에 대한 처리가 끝날 때까지 새로운 요청이 지연되지 않는다는 장점이 있다.

4.2 카플방식을 사용한 프로그램의 비용과 응답시간

카플 방식의 전체 비용은 데이터베이스의 시작점부터 액세스를 시작하여 다시 시작점으로 돌아오기 전까지의 처리시간을 기준으로 구할 수 있다. 사용자 요청의 발생율을 λ라 가정하고 데이터베이스를 모두 액세스 하는 동안 소요되는 데이터베이스 액세스 시간과 도착한 사용자 요청과 비교하는 시간의 합을 C^{cp}_{total}라 하자. 이 주기(C^{cp}_{total})동안 발생하는

평균 요청의 수는 $\lambda \cdot C_{total}^{cp}$ 가 된다. 주기(C_{total}^{cp})동안 데이터베이스는 한번만 액세스 한다. 주기동안의 총 비용을 구해보면 다음과 같다.

$$C_{total}^{cp} = C_{DB} + \lambda \cdot C_{total}^{cp} \cdot C_{seq} \quad \text{식(7)}$$

식 (7)을 C_{total}^{cp} 에 대해서 정리하면

$$C_{total}^{cp} = \frac{C_{DB}}{1-\lambda \cdot C_{seq}} \quad \text{식(8)}$$

식(7)을 사용자 요청이 없을 확률을 고려하면 식(9)과 같다.

$$C_{avg}^{cp} = (\frac{1}{\lambda} - C_{seq})(1 - e^{-\lambda \cdot \frac{C_{DB}}{1-\lambda C_{seq}}}) + C_{seq} \quad \text{식(9)}$$

카플 방식의 응답시간(W_{cp})은 사용자 요청이 처리되는 시간($C_{DB} + C_{seq}$)과 이를 처리하는 시간 동안 같이 처리되는 사용자 요청의 처리 시간($\lambda \cdot W_{cp} \cdot C_{seq}$)의 합과 같다.

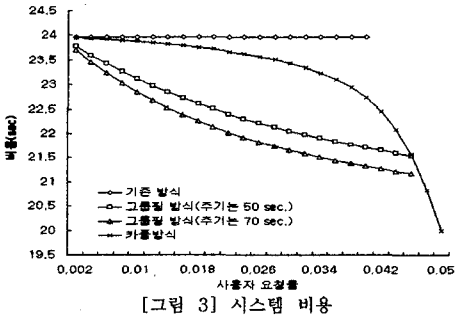
$$W_{cp} = C_{DB} + C_{seq} + \lambda \cdot W_{cp} \cdot C_{seq} \quad \text{식(10)}$$

식 (10)를 정리하면 다음과 같다.

$$W_{cp} = \frac{C_{DB} + C_{seq}}{1-\lambda \cdot C_{seq}} \quad \text{식(11)}$$

5. 성능비교

성능 비교를 위한 C_{DB} 와 C_{seq} 는 fasta34t11버전을 펜티엄4 1.6G Hz, 256Mbyte Ram(pc2700) 사양에서 인간(human) 단백질 중 세포의 산화 환원에 작용하는 색소 단백질(cytochrome)을 데이터베이스로 Genbank(Release 72.02)를 사용하여 비교 분석하는데 드는 시간으로 C_{DB} 는 3.99 sec., C_{seq} 는 19.98 sec.이다.



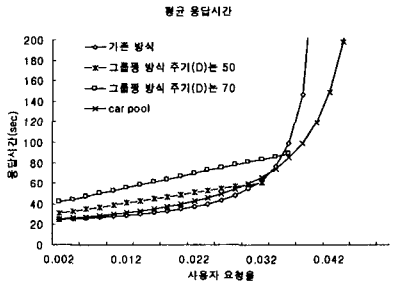
[그림 3] 시스템 비용

5.1 비용비교

그림 3은 기존의 방식과 그룹핑 방식 그리고 카플 방식의 평균비용의 비교를 위해서 식(1), 식(5) 그리고 식(9)의 결과를 이용하였다. 그룹핑 방식에서 주기는 50 sec.와 70 sec.으로 잡아보았다.

그림 3은 각 방식의 사용자당 시스템 비용을 비교한 그래프이다. 각 방식은 임계값까지 그래프에 표시하였다. 그림 3에서 x축은 사용자 요청율 λ 이고 y축은 사용자당 시스템 비용을 나타낸다. 기존의 방식을 제외한 그룹핑 방식과 카플 방식은 λ 값이 증가할수록 시스템 비용이 줄어드는 것을 볼 수 있다. 또한 그룹핑 방식은 D에 따라 시스템 비용을 많이 감소 시킬 수 있는 것을 볼 수 있다. 카플 방식 역시 λ 값이 증가할수록 시스템 비용이 줄어드는 것을 알 수 있다.

5.2 응답시간 비교



[그림 4] 응답 시간 비교

그림 4은 각 방식의 응답시간을 비교한 그래프이다. 그림 4에서 x축은 사용자 요청율 λ 를 나타내고 y축은 임의의 요청에 대한 프로그램의 평균 응답시간을 나타낸다. 기존의 방식은 임계 점에 가까워 질수록 응답시간이 급격히 증가하였다. 카플방식이 가장 짧은 응답시간을 보였다. 이는 사용자가 적을 때는 데이터베이스를 즉시 읽기 때문에 응답이 빠를 수 있는 것이다. 또한, 기존 방식보다 데이터베이스를 액세스 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 요청율 λ 가 커질수록 대기행렬(Queue)에서 대기하는 사용자요청의 수가 많아지므로 응답시간이 급격히 증가한다.

6. 결론

단백질 데이터베이스는 매우 빠르게 증가 하고 있다. 단백질 데이타의 폭발적인 증가는 컴퓨터의 발전 속도를 능가할 정도이며 또한 이를 분석하기 위해서도 데이터베이스의 빈번한 액세스는 생물 정보학 관련 문제들을 처리함에 있어 과부하로 작용할 것이다.

그룹핑 방식과 카플 방식은 사용자 요청당 데이터베이스의 액세스를 줄여 더 많은 사용자들에게 서비스를 제공할 수 있다.

참고문헌

[1] Hyun Seok Park, PhD. " 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.(1990), " 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] 김민준, 김재훈, " 생물정보학에서 사용자 그룹핑 기법을 이용한 서열 정렬 방법" , 정보과학회 2002 춘계학술발표 29권 1호 pp. 13~15, 2002년 4월.