유전자 알고리즘을 이용한 데이타 분산 기법

이순미°, 박혜숙
경인여자대학 컴퓨터정보기술학부
leesm@kic.ac.kr, edpsphs@kic.ac.kr

Data Distribution using Genetic Algorithm

Soon-mi Lee°, Hea-sook Park
School of Computer Information Technology, Kyungin Women's College

요 약

분산 데이타베이스 시스템은 통신망으로 연결되어 있는 컴퓨터 노드들의 집합으로 구성되어 있으며 각 노드들은 데이타, 프로그램, 처리능력 등의 자원을 공유한다. 데이타의 분산은 접근 시간 단축, 가용성과 신뢰성, 동시성의 증가와 같은 장점이 있으나 통신 비용과 시스템 부하와 같은 성능저하 요인이 될 수도 있으므로 데이타를 최적의 노드에 분산시키는 할당 문제가 중요한 이슈이다.

본 논문에서는 시스템 운영 비용을 최소화 시키는 최적의 할당 노드를 찾기 위한 목적 함수를 기술하였으며 유전자 알고리즘을 사용하여 할당 목적 함수의 해를 구현하였다.

1. 서 론

분산 데이타베이스 시스템은 통신망으로 연결되어 있는 컴퓨터 노드들의 집합으로 구성되어 있다. 분산 데이타베이스 시스템을 구축하는 가장 큰 이유는 각 노드가 가지고 있는 처리능력, 프로그램, 데이타 등과 같은 자원을 각 노드들이 공유할 수 있기 때문이다. 즉, 분산 데이타베이스를 구성하는 각 노드들은 필요한 데이타를 가지고 있지 않아도 필요한 경우에 다른 노드로부터 얻을 수 있다. 이러한 데이타베이스의 분산은 질의 처리 시간과 데이타 접근 비용을 줄일 수 있고 데이타에 대한 가용성 및 신뢰성, 동시성을 증가시켜 성능을 향상시킬 수 있다. 그러나, 이러한 데이타의 분산은 데이타와 메시지의 전송으로 인한 통신비용, 데이타 처리를 위한 CPU, I/O 부하 등의 성능 저하 요인들이 수반되기도 하므로 각 노드에 최적의 데이타를 할당하는 것이 분산 데이타베이스 시스템 설계에서 중요한 이슈가 된다.[1]

분산 데이타베이스 시스템의 설계는 릴레이션을 분산의 단위인 프래그먼트들로 분할하는 데이타 분할 단계와 프래그먼트를 최소의 비용과 응답시간을 가질 수 있도록 각 노드에 분산시키는 할당 단계로 나누어진다. 일반적으로 데이타 할당의 문제는 NP-Complete로 알려져 있으며 대부분의 연구는 처리 비용 또는 성능 면을 고려하여 최적의 해를 찾기 위한 여러 가지 휴리스틱(Heuristic)들을 제시하고 있다.

본 논문에서는 [2]에서 정의된 할당 모델에 대하여 유전자 알고리즘[3]을 적용하여 최소의 비용이 드는 최적의 할당 해를 찾아 구현하였다. 최적의 데이타 할당은 최소 비용의 측면과 최대 성능의 측면으로 평가될 수 있는데 본 논문에서는 비용 측면만을 고려하였다.

본 논문의 2장에서는 할당에 필요한 요구사항을 정량화하여 제시하였고 3장에서는 시스템 운영 비용을 최소화시키는 최적의 할당 노드를 찾기 위한 목적 함수를 기술하였으며 4장에서 할당 목적 함수의 해를 유전자 알고리즘을 사용하여 구현하였으며 5장에서 결론을 맺었다.

2. 할당 모델 구성요소

분할된 프래그먼트를 컴퓨터 통신망을 통해 흩어져 있는 여러 사이트 중 최적의 사이트에 할당하기 위해서는 데이타베이스와 사용자 질의에 관한 정보 및 통신망 정보, 각 사이트에 대한 저장 용량과 처리 성능에 대한 정보가 정량화되어 있어야 한다. 다음은 할당에 필요한 요구사항들을 정량화하여 정의하였다.

- 프래그먼트들의 집합 : $F = \{ F_1, F_2, F_3 \cdots Fn \}$
- 사이트들의 집합 : $S = \{ S_1, S_2, S_3 \cdots Sm \}$
- 질의들의 집합 : $F = \{ q_1, q_2, q_3 \cdots qn \}$
- 프래그먼트 F_i의 크기: $size(F_i)=card(F_i)\times length(F_i)$ ($length(F_i)$는 프래그먼트 F_i에서 튜플의 바이트 수)
- CR_{ij} : 질의 q_i가 프래그먼트 F_j에 대하여 읽기 연산을 수행한 접근의 수
- CU_{ij} : 질의 q_i가 프래그먼 F_j에 대하여 갱신 연산을 수행한 접근의 수
- $O(i)$: 질의 q_i가 발생한 사이트
- USC_k : 사이트 S_k에 데이타를 저장하는 데에 필요한 단위비용
- UPC_k : 사이트 S_k에서 한 단위의 일을 처리하는 데에 필요한 비용
- g_{ij} : 사이트 S_i와 S_j사이에 프래임 당 통신 비용
- $fsize$: 한 프래임의 크기(바이트 수)

3. 할당 최적화 모델

3.1 결정 변수

할당 수식을 정의하기 위한 결정 변수를 다음과 같이 정의한다.

$r_{ij} = \begin{cases} 1 & \text{질의 } qi\text{가 프래그먼트 } Fj\text{를 검색한 경우} \\ 0 & \text{그렇지 않은 경우} \end{cases}$

$u_{ij} = \begin{cases} 1 & \text{질의 } qi\text{가 프래그먼트 } Fj\text{를 갱신한 경우} \\ 0 & \text{그렇지 않은 경우} \end{cases}$

$x_{ij} = \begin{cases} 1 & \text{프래그먼트 } Fi\text{가 사이트 } Sk\text{에 저장된 경우} \\ 0 & \text{그렇지 않은 경우} \end{cases}$

3.2 할당 함수

할당 함수를 정의하는 목적은 저장비용과 질의처리비용 및 전송비용을 포함한 시스템 운영 비용을 최소화하기 위함이다. 할당함수(AC)는 다음과 같다.

$$AC = \sum_{\forall Sk \in S} \sum_{\forall Fj \in F} STC_{jk} + \sum_{\forall qi \in Q} QPC_i + \sum_{\forall qi \in Q} TC_i$$

여기서, STC_{jk}는 프래그먼트 F_j를 사이트 S_k에 저장하는 비용이며 QPC_i는 질의q_i를 처리하기 위한 접근 비용을 의미하며 TC_i는 통신 비용을 의미한다. 할당 함수 AC를 최소로 하는 사이트 할당이 최적의 할당 사이트가 된다.

3.2.1 저장 비용

저장비용 STC_{jk}는 다음과 같이 산출된다.

$$STC_{jk} = USC_k \times size(F_j) \times x_{jk}$$

여기서, USC_k는 사이트 S_k에 데이타를 저장하는 데에 필요한 단위비용이며 $size(F_j)$는 프래그먼트 F_j의 크기를 의미한다.

3.2.2 질의처리 비용

질의 처리 비용 QPC_i는 프래그먼트를 검색 및 갱신하는 데에 필요한 접근 비용(CPU 비용)으로 본 논문에서는 같은 사이트 내에서 검색과 갱신을 처리하는 비용은 동일한 것으로 가정하였으며 무결성 유지 비용과 동시성 제어 비용은 고려하지 않았다.

$$QPC_i = \sum_{\forall Sk \in S} \sum_{\forall Fj \in F} (r_{ij} \times CR_{ij} + u_{ij} \times CU_{ij}) \times x_{jk} \times UPC_k$$

여기서, $(r_{ij} \times CR_{ij} + u_{ij} \times CU_{ij})$는 질의 q_i가 프래그먼트 F_j에 접근하는 수를 계산하는 식이며 이 때에 CR_{ij}는 읽기 연산을 수행한 접근의 수를 나타내며 CU_{ij}는 갱신 연산을 수행한 접근의 수를 나타낸다. UPC_k는 사이트 S_k에서 한 단위의 일을 처리하는 데에 필요한 비용이다.

3.2.3 전송 비용

분산 데이타베이스에서 갱신 요청을 수행하기 위한 전송 방식과 검색 요청을 수행하기 위한 데이타 전송 방식은 매우 다르다.[4] 본 논문에서는 검색 질의 수행과 갱신 질의 수행의 차이점을 반영하기 위하여 전송 비용 수식을 검색 전송 비용(RTC_i)와 갱신 전송 비용(UTC_i)으로 나누어서 정의하였다.

■ **검색 연산 전송 비용**

검색에 필요한 전송 비용은 다음과 같이 정의된다.

$$RTC_i = \sum_{\forall Fj \in F} \{ \min_{Sk \in S} (ATC + BTC) \} \times CR_{ij\,O(i)}$$

단, $ATC = r_{ij} \times x_{jk} \times g_{o(i),k}$, $BTC = r_{ij} \times x_{jk} \times \frac{seli(Fi)}{fsize} \times g_{k,o(i)}$

여기서, ATC는 검색요청을 전송하는 비용이며 BTC는 질의 발생 사이트로 결과를 전송하는 비용이다. $CR_{ij\,O(i)}$는 질의q_i가 발생한 사이트 $o(i)$에서 프래그먼트 F_j에 대하여 수행되는 검색질의 q_i의 발생 빈도수를 나타낸다. $g_{o(i),k}$는 질의 발생 사이트인 $o(i)$와 사이트 K 사이의 통신 비용을 의미한다.

■ **갱신 연산 전송 비용**

갱신 연산에 필요한 전송 비용은 다음과 같이 정의된다.

$$UTC = \sum_{\forall Sk \in S} \sum_{\forall Fj \in F} 2(STC + CTC) \times CU_{ij\,O(i)}$$

단, $STC = u_{ij} \times x_{jk} \times g_{o(i),k}$, $CTC = u_{ij} \times x_{jk} \times g_{k,o(i)}$

여기서, STC는 질의가 발생한 사이트에서 다른 사이트 k로 메시지를 보내는 비용이며 CTC는 사이트 k에서 질의가 발생한 사이트로 확인 메시지를 반환하는 비용이다. 갱신 수행 단계에서 STC와 CTC는 각각 2번씩 발생하게 된다. $CU_{ij\,O(i)}$는 질의q_i가 발생한 사이트 $o(i)$에서 프래그먼트 F_j에 대하여 수행되는 갱신질의 q_i의 발생 빈도수를 나타낸다.

4. 할당 해의 생성

본 논문에서는 위에서 제시한 할당 목적 함수의 해를 찾기 위하여 유전자 알고리즘[3]을 사용하였으며 윈도우 XP PC에서 C언어로 구현하였다.

표 1 릴레이션의 프래그먼트

릴레이션	프래그먼트		
Student(19K)	S1(9K)	S2(6K)	S3(4K)
Course(250K)	C1(130K)	C2(70K)	C3(50K)
Professor(15K)	P1(7K)	P2(5K)	P3(3K)

표 2 노드별 검색 및 갱신 질의 발생빈도

질의	타입	부질의	참조 프래그먼트	발생빈도	노드별 발생빈도			
					노드1	노드2	노드3	노드4
R1	검색	R1.1	P1, S1, C1	840	0	800	0	40
		R1.2	P2, S2, C2	800	800	0	0	0
		R1.3	P2, S2, C2	800	0	0	800	0
R2	검색	R2.1	P1, S1	35500	20000	8000	0	7500
		R2.2	P2, S2	46000	40000	2000	0	4000
		R2.3	P3, S3	41000	0	0	40000	1000
R3	검색	R3.1	S1	800	0	800	0	0
		R3.2	S2	800	800	0	0	0
		R3.3	S3	800	0	0	800	0
U1	갱신	U1.1	P1	120	20	80	0	20
		U1.2	P2	72	40	20	0	12
		U1.3	P3	48	0	0	40	8
U2	갱신	U2.1	S1	120000	20000	80000	0	20000
		U2.2	S2	72000	40000	20000	0	12000
		U2.3	S3	48000	0	0	40000	8000
U3	갱신	U3.1	C1	120000	20000	80000	0	20000
		U3.2	C2	72000	40000	20000	0	12000
		U3.3	C3	48000	0	0	40000	8000

구현을 위하여 <표 1>과 같은 3개의 릴레이션을 사용하였으며 릴레이션은 각각 3개의 프래그먼트로 구성되어 있고, 노드는 4개가 있는 것으로 가정하였다. <표 2>는 노드별로 검색 및 갱신 질의에 대한 발생빈도를 나타내고 있다. <표 2>에서 볼 수 있듯이 검색 질의 3개와 갱신 질의 3개가 있는 것으로 가정하였으며 각 질의는 다시 부질의로 나뉘어져 처리된다.

<표 1>및 <표 2>와 같은 구현 환경 하에서 유전자 알고리즘을 실행시킨 할당의 결과는 <표 3>과 같다. 결과에서 알 수 있듯이 프래그먼트 S1과 P1은 노드1, 노드2, 및 노드 4에 중복하여 할당되어 있다. <표 2>의 R2.1, R2.2, R2.3 질의는 발생빈도가 높으며 다른 프래그먼트 2개를 각각 참조하고 있다. 따라서 <표 3>의 결과를 보면 R2.1, R2.2, R2.3에 의하여 함께 참조되고 있는 프래그먼트 P1과 S1, P2와 S2 그리고 P3와 S3는 각각 같은 노드에 할당되어 있어 할당 결과가 합리적으로 나왔음을 알 수 있다.

5. 결 론

본 논문에서는 시스템 운영 비용을 최소화 시키는 최적의 할당 노드를 찾기 위한 목적 함수를 기술하였으며 제시된 할당 목적 함수의 해를 유전자 알고리즘을 사용하여 구현하였다. 본 논문에서 제시한 할당 모델에서는 저장, 질의처리 및 전송 비용 요소를 고려하였으며 검색 연산과 갱신 연산의 수행 시에 데이타 전송 방식이 서로 다른점을 전송 비용 측정에 반영하였다. 또한 기술된 할당 목적 함수의 해를 찾기 위하여 유전자 알고리즘을 이용하여 구현을 하였다.

표 3 프래그먼트의 할당 결과

프래그먼트	노드1	노드2	노드3	노드4
S1	X	X		X
S2	X			
S3			X	
C1		X		
C2	X			
C3			X	
P1	X	X		X
P2	X			
P3			X	

참고문헌

[1] S. Ram and R.E. Marsten "A model for database allocation incorporating a concurrency control mechanism", IEEE Transactions on Knowledge & Data Engineering, Vol.3 No.3, 1991.

[2] 이순미, "분산 데이타베이스에서 할당 알고리즘의 설계", 한국정보처리학회학술발표논문집, 10권1호, 2003.

[3] David E. Goldberg, "Genetic Algorithms : in Search, Optimization & Machine Learning", Addison-Wesley, 1989.

[4] Salvatore T. March, Sangkyu Rho, "Allocating data and operations to nodes in distributed database design", IEEE Transactions on Knowledge and Data Engineering, 1995.

[5] M.T. Ozsu and P. Valduriez, "Principles of Distributed Database Systems", Prentice Hall, 1991.