論文
7-4-4

이동컴퓨팅 환경에서 재배치 기법을 이용한 데이터 일관성 유지비용에 관한 연구 (Research about a Data Consistency hold Cost Using a Relocatio n Technique in Mobile Computing Environment)

최강희(Choi Kang Hee)[1)]

요 약

최근에 이동 컴퓨팅환경에서 사용자가 이동중에도 정보를 손실없이 빠르게 얻기 위해, 서버에 정보를 복제하여 사용하는 방법이 연구되고 있다. 지금까지는 이동 호스트가 복제 서버에 데이터를 복제하는 방법인 정적 복제 기법(Static Replica Allocation : SRA)기법을 사용하고 있다. 이 기법은 이동 호스트가 셀에 이동하고나서, 복제서버에 데이터를 복제하는 방법이다. <중략>그리고 정적 복제기법을 사용하였을 때와 본 논문에서 제안된 기법을 비교하여, 데이터 접근 비율을 이동성과 셀의 수에 따른 성능을 분석하였다. 그래서 그 결과, 이동 호스트들의 이동성이 낮을 때에는 접근 비용에서, 제시된 기법이 정적 복제 기법보다 5% 낮게 나타났다. 그리고 이동률에 따른 접근비용은 2%~2.5% 감소되었음을 알 수 있다. 마지막으로 셀의 수에 따라서, 선택복제기법의 확장성은 1.7%정도 떨어짐을 알 수 있다.

ABSTRACT

Recently, A method of using the replicated database on a server to get new data without mis sing any information has been being studied in mobile computing environment. So far we ha ve used the Static Replica Allocation(SRA) for the replication which is the method of the repl ication on the server. This method is to replicate the data on the replica server after a movin g host is transferred to a cell. Since the network of the SRA is very good, and if there are f ew moving users, no trouble will happen. But if there is no moving users in a cell, the data will not be shared.

Therefore, this paper is about the study of the method of relocation after replicating the data to the cells for the users(User Select Replica Allocation : USRA). We also analyze the access rate and the possibility which are closely related to the moving frequency of the mobile hosts and the numbers of the cells. As a result, We show that the 5% lower access cost and the 2%~2.5% gains are achieved from the low mobility of the mobile hosts. We also show that t he extension function of USRA reduces the cost by 1.7% and it is influenced by the numbers of the cells.

논문접수 : 2006. 7. 15.
심사완료 : 2006. 7. 28.

1)정회원 : 대원과학대학 컴퓨터정보계열 교수

1. 서 론

이동 컴퓨팅 환경에서 이동 데이터베이스 시스템은 데이터를 검색하고 저장하는 작업을 하는데, 이러한 일은 응용분야가 매우 넓고, 그 기능 또한 매우 중요하다. 이동 데이터베이스는 이동 호스트들이나 고정 호스트들에 분산된 데이터들을 다루고 있다. 그러나 분산된 환경에서와 이동 컴퓨팅 환경하에서 다른점은, 통신하는데 많은 제약이 있고, 이동 호스트가 존재하고 있다는 것이다.[3,4,6,8]

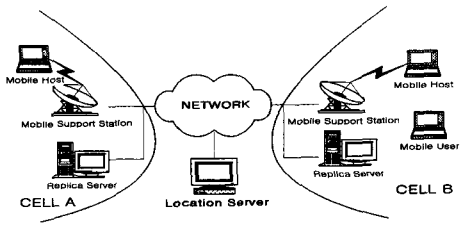
이동 사용자가 이동 중, 정보를 빠르게 얻게 위해, 캐쉬 방법을 쓰기도 하는데,[2] 요즈음은 데이터를 서버에 복제하여 셀에 사용중인 이동 사용자들에게 정보를 제공하는 방법이 연구되고 있다.[7,8] 기존 분산 데이터베이스에서는 데이터 복제 방법인 정적 복제 기법(Static Replica Allocation : SRA)으로 이동 컴퓨팅 환경에 적용되었는데, 정적 복제 기법은 이동 호스트가 셀에 이동하고, 이동된 셀의 복제 서버에 데이터를 복제하는 방법이다. 이것은 네트워크가 양호하고, 이동 사용자의 수가 적은 경우라면, 데이터를 사용하는데는 문제가 없지만, 셀에서 이동 사용자가 존재하고 있지 않다면, 그 데이터는 공유되지 못하는 데이터가 된다. 그리고 이 방법은 이동 컴퓨팅 환경에서의 제약조건, 서버와의 잦은 통신단절, 협소한 대역폭등으로 인해, 정적 복제 기법을 사용하는 것은 적합하지 않다.

그래서, 본 논문에서는 이동 컴퓨팅 환경에 적합한 기법을 제안하였는데, 이 기법은 이동 사용자가 많은 셀에 데이터를 복제하여 재배치하는 (User Select Replica Allocation : USRA) 방법을 제안하였다. 그리고, 정적 복제 기법(SRA)을 사용하였을 때와 본 논문에서 제안된 기법 선택복제기법(USRA)을 비교하여, 데이터 접근 비율을 이동성과 셀(cell)의 수에 따른, 성능을 분석하였다.

2. 연구동향

2.1 이동 컴퓨팅 시스템 환경

(그림 1)에서와 같이 이동 컴퓨팅 시스템은 이동호스트(mobile host)와 무선 인터페이스를 갖춘 이동기지국(Mobile Support Station : MSS)과 복제서버(Replica Server)로 구성되어 있다. 또한 이동 기지국이 관리하는 지리적인 영역을 셀(cell)이라 하며, 하나의 셀은 여러개의 이동 호스트와 이동 사용자를 포함하고 있고, 이동 호스트는 이동 기지국과 무선으로 통신을 한다.[5,7,8,9]



(그림 1) 이동 컴퓨팅 시스템

(Fig.1) mobile computing system

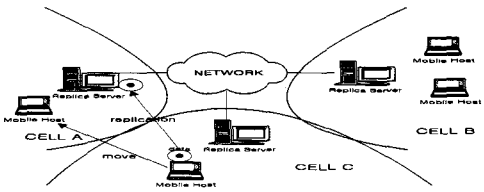
이동 호스트는 데이터 서버(Data Server)라 하는데, 이것은 기본적으로 트랜잭션 명령을 제공하고 있다. 이동 호스트들은 복제서버(Replica server)에 데이터를 복제할 수 있다. 그리고, 복제의 위치는 복제 기법에 의해 정해진다.

이동기지국의 조정자(Coordinator)는 다른 이동기지국과 이동 호스트들로 부터 트랜잭션된 데이터들을 관리하고 이들을 자신의 복제서버에 실행시킨다. 만약 이동기지국에 복제된 데이터가 존재하지 않으면, 조정자는 그들의 위치에 관한 정보를 위치서버(Location server)에 연결하여, 복제된 정보를 얻고, 그 위치에서 이동기지국의 조정자는 복제 정보를 트랜잭션하여 저장하게 된다. 이렇게 얻어진 정보는 위치서버에서 이동 호스트 위치 정보에 관한 데이터베이스를 갱신하게 된다.[10]

이렇게 되면, 이동 사용자(Mobile User)는 데이터 복제에 대한 정보위치가 항상 복제 서버에 저장되게 된다. 그러면 이동 사용자가 많은 데이터를 이용할 경우, 이동 사용자는 많은 복제에 대한 위치 정보를 가지게 된다.

2.2 정적 복제 기법

정적 복제 기법은 분산 데이터베이스에서 사용되던 기법을 이동 컴퓨팅 환경하에 적용시킨 기법이다 [2,3]. (그림 2)는 이동호스트가 셀C에서 셀A로 이동하고, 이동한 셀 A의 복제 서버에 데이터를 복제하는 상태를 나타내고 있다.



(그림 2) SRA기법의 데이터 복제 모델

(Fig.2) A model of data replica in SRA

SRA기법은 이와 같이 이동된 셀에 데이터를 복제하는 방법이다. 이 기법은 이동 호스트가 임의로 이동된 셀에 데이터를 복제하기 때문에 이동한 셀에 이동 사용자가 존재하지 않을 경우에는 공유되지 못하는 데이터가 된다. 정적복제 방법의 동작 방법은 다음과 같다.

첫 번째로, 서버는 이동 호스트에서 데이터의 복제가 이루어진다. 각각 쓰는 작업이 되어질 때, 이동 호스트에서는 복제서버에서 복사와 쓰는 작업이 필요하게 된다. 읽는 작업은 이동 호스트에서 복제 서버로부터 이루어진다. 여기서, 접근 비용은 데이터가 갱신되었을때, 고정 호스트에서는 이동 사용자의 수 보다 데이터 복제가 할당된 것이 크게 된다.

두 번째로, 복제된 데이터는 이동 호스트의 서버가 위치된 곳에 있게 된다. 그래서 이동 호스트는 자신이 위치한 서버로부터 읽는 작업을 수행하게 된다. 여기서 읽는 작업과 쓰는 작업은 정적인 복제 작업에 해당된다. 그러나 복제는 쓰는 작업을 하는 이동 호스트보다 이동 사용자의 읽는 작업이 먼저 끝이 나야한다.

세 번째로, 서버는 자신이 위치한 복제서버의 데이터를 복사한다. 그리고, 이동 사용자나 이동 호스트는 그 서버로부터 읽는 작업을 한다. 여기서, 읽는 작업과 쓰는 작업은 정적인 원격 복사에 의해 이루어진다.

이 정적 복제 기법은 네트워크 상태가 양호하고, 셀에서 이동 사용자의 수가 기지국 수보다 적은 경우라면, 이 기법을 적용시키는 것이 좋다. 그러나 한 셀에서 이동 사용자가 존재하지 않을 경우에는 데이터가 공유되지 못하기 때문에 좋지 않다. 다음은 정적 복제 기법에서 비용계산 방법을 나타내었다.

2.3 비용 계산식

정적 복제 기법의 비용계산에서, 어느 한 셀에 복제하는 이동 사용자의 읽는 작업과 쓰는 작업, 그리고 이동 호스트의 쓰는 작업을 함께 나타내었다. 각각 $R_{\mu ser}$ 와 $W_{\mu ser}$ 그리고 $W_{\ddot{U}st}$ 로 표기하여, 다음과 같이 분석되어졌다. 이 기법에서 각각의 사용자 데이터에서 평균 접근 비용은, AVG_{SRA}로 나타내어 보면, 다음 <식2-1>과 같다.

$$AVG_{SRA} = R_{\mu ser} + W_{\mu ser} + W_{\ddot{U}st} \quad <식2-1>$$

여기서, 이동 사용자의 데이터 접근 요구는 λ로 포아송 분포에 의해 되어지고, 이동 호스트로부터의 접근은 α의 비로 나타내었다 ($0 \le \alpha \le 1$). 복제된 셀에 존재하는 k명의 이동 사용자와 이동 호스트가 존재할 확률 P(k)를 계산하여 보면 다음 <식2-2>와 같다.

$$P(k) = \binom{n-1}{k}\left(1-\frac{1}{N}\right)^{n-1}(N-1)^{-k}$$

(N: 셀의수, n: 이동사용자의수), <식2-2>

그래서 이동사용자와 이동호스트에 대한 접근 비용은 다음 <식2-3,4,5>와 같이 계산된다.[2]

$$R_{\mu ser} = \sum_{k=0}^{n-1} P(k)\frac{n-1-k}{n-1}(1-\alpha)\lambda \quad <식2-3>$$

$$W_{\mu ser} = \sum_{k=0}^{n-1} P(k)\gamma_{\mu ser}(1-\alpha)\lambda \quad <식2-4>$$

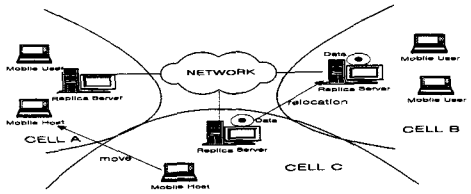
$$\left\{\frac{k-N-1}{n-1-N}+\frac{n-1-k}{n-1}\left\{\frac{2}{N}+\frac{3(N-1)}{N}\right\}\right\}$$

$$W_{Ust} = \frac{N-1}{N}\gamma_{Ust}\alpha\lambda \qquad <식2-5>$$

3. 개선된 데이터 재배치 기법

본 논문에서 제안된 기법은, 복제의 위치와 이동 호스트의 위치가 위치 서버에 각각, 분리된 테이블에 저장되게 한다. 그리고, 분리된 테이블을 연결해주는 매핑 테이블은 부분적으로 복제를 이동 사용자가 알고 있도록 복제 순서를 유지하고 있다. 이 테이블은 이동호스트의 위치와 복제의 위치를 둘다 사용할 수 있도록 해야 한다. 데이터의 트랜잭션에서 갱신 작업의 경우, 각각의 테이블은 다음과 같은 2가지 상태가 발생할 경우라면, 즉시 갱신된다.

첫 번째, 이동사용자가 데이터에 접근하여, 이동 사용자가 오류를 발생하여 시작하지 못하는 경우이다. 여기서는, 복제의 위치에 대해 위치 서버가 질의를 해야 하기 때문에, 이 질의에 대해서는, 위치 서버가 매핑테이블에 이동 사용자의 등록을 추가해야되고, 복제가 되기 위해, 이동 사용자의 위치가 저장되어야만 한다. 두 번째, 이동 사용자가 셀간의 이동이 수행되어질 때이다, 그것은 새로운 셀에 들어가는 즉시, 이 이동을 알려야 한다. 위치 서버는 이동 호스트의 위치 정보에 대해 테이블을 갱신해야 만 한다. 결국, 위치 서버는 모든 테이블을 포함하여 계산을 수행해야 하고, 그 결과로 복제가 재배치 된다. 복제가 재배치되었다는 것은, 복제에 대한 위치 테이블이 갱신되었다는 의미이다. 그런 후에, 복제 서버의 내용만 참조하면 된다. 사용자가 같은 접근 평균률로 데이터에 접근한다면, 제안된 선택 복제 기법(USRA)은 접근비용이 아주 좋게 나타난다. 또한, 복제 재배치는 이동 사용자의 수에 의해 수행되어진다. 이동 호스트가 데이터를 갱신하는 경우에는, 제안된 기법으로는 복제의 위치가 전이되지는 않는다, 왜냐하면, 이동 호스트는 복제가 같은 셀에 항상 있지는 않기 때문이다. (그림 3)에 복제 재배치 기법의 모델이 제시되어 있다. 여기서는 이동 호스트는 셀A로 이동하고 있는데, 이동 사용자가 많은 셀 C를 선택하여 복제서버에 데이터를 복제하고 있다. 이동 호스트가 복제 서버에 데이터를 복제하는데 있어, 이동 사용자의 수가 많은 셀에, 재배치를 해야 하는 작업이 추가 된다. 그래서, 비용 계산에 있어, 복제 재배치에 대한 비용을 함께 계산하여야 한다.



(그림 3)USRA기법의 복제재배치 되는 모델
(Fig.3) A model of replica relocation in USRA

3.1 알고리즘

3.1.1 선택복제기법의 읽기, 쓰기

선택복제를 하기 위해서는 이동사용자와 이동호스트는 읽기, 쓰기, 데이터 복제작업과 셀을 선택하는 작업, 재배치작업을 한다. 선택복제기법에 대한 메인 알고리즘은 [알고리즘 3-1]과 같다.

```
Void USRA(Req, data)
 {switch (Req)
  { case 'r' : USRA_r(data) /* Read Data */     break;
  case 'w' : USRA_w(data) /* Write Data */ break;
  case 're' : USRA_re(data) /*Replicate Data*/ break;
  case 'rel' : USRA_rel(data) /* Relocate Data
*/ break;
  case 's' : USRA_s(data) /* select cell */     break;
  case 'x' : USRA_x(data) /* exit request */ break;
  default : printf("parameter err₩n");     break; }}
```

[알고리즘 3-1] 선택복제기법 메인 알고리즘

[알고리즘 3-2]와 [알고리즘 3-3]은 복제재배치기법에 대한 읽기작업과 쓰기작업을 나타냈다. Void USRA_r은 읽는 작업이 이동된 셀에서 요구되었을 때 수행되고, 다른 셀에 존재하는 이동사용자가 읽는작업을 요구시 수행되는 알고리즘이다. 그리고 Void USRA_w는 현재 셀이나 아니면 다른 셀을 선택하여 데이터를 쓰는 작업을 하는 알고리즘이다.

```
Void USRA_r(data)
 { if (replica data existence)
{ retrieve the data from the local databases;
send data to processor data; cell→SumR +=1;
  if (data != cell → data) othercell → Muser +=1
   (for othercell submitted the read data); }
 else /* Not existence local replica data */
{ find a othercell, such that othercell → Drt == 1;
 submit read request othercell, and wait for reply;
 after get the data, send it to current cell data; }}
```

[알고리즘 3-2] 선택복제기법의 읽기 알고리즘

[알고리즘 3-3]은 선택복제기법의 쓰기 알고리즘으로 현재 복제서버에 복제된 데이터가 존재하지 않으면, 해당 셀의 복제서버에 데이터를 쓰고, 그렇지 않으면, 다른 셀을 선택하여 다른 셀의 복제서버에 쓰기가 적용되는 알고리즘이다.

```
   Void USRA_w (data)
 { if (existence replica data)
    {update the local replica; cell → sumw +=1;
  if (data = cell → data)/* 현재 셀에서 데이터를 읽음 */
 { for each othercell, do
   {if(othercell → Drt==1)propagate write othercell;}}
   else   /* 다른 셀에 데이터를 읽어옴 */
 { find the othercell such that othercell has data;
    othercell → Now +=1;
    for each othercell, do
{if(othercell→Drt==1) and (othercell != existence
data) propagate the write to othercell;} } }
else /* Not existence replica data */
{ find a othercell so that othercell → Drt == 1;
  submit the write request to othercell; }}
```

[알고리즘 3-3] 선택복제기법의 쓰기 알고리즘

3.1.2 선택복제기법의 재배치

[알고리즘 3-4]는 복제 알고리즘으로 Void USRA_re는 이동호스트가 지정한 셀로 데이터를 복제하거나, 다른 셀로 재배치하라는 메시지를 받았을 경우, 복제하는 알고리즘이다.

그리고, Void USRA_rel은 재배치 알고리즘으로써, 데이터를 복제하여, 이동사용자의 수가 많은 셀에 있는 복제서버에 재배치되는 알고리즘이다.

```
Void USRA_re(data)
 /* 이동 호스트가 지정한 셀로 데이터를 복제하는 프로
시져 */
 { if (existence singleton replication schema)
 /* 복제된 데이터가 현재 셀에 존재하고 이동 호스트가
지정한 셀의 값이 0인 경우 */
    { if (Expansion( ) == 0)  switch( ); }
    else if(existence replication othercell)
/* 다른 셀에 데이터가 존재하고 데이터위치를 알고 있는 경우 */
Contraction( ); }
 else if (Replica_Othercell processor)  Contraction( );
  cell → SumR = cell → sumw=0;
 for each othercell, do  othercell → NoR = othercell
→ NoW = 0; }
```

[알고리즘 3-4] 복제 알고리즘

[알고리즘 3-5]은 셀 선택 함수 알고리즘으로 복제데이터를 가지고 있는 셀로 패스가 이루어졌을 경우, 1을 리턴하고, 그렇지 않으면 0을 리턴하게 하는 알고리즘이다.

```
 int Expansion( )
{ int succeed = 0;
 for each othercell
 if (othercell → Drt == 0)
  {if (othercell → NoR > cell → sumw - othercell →
NoW)
  {send a "replica" message to othercell, along with
a replica of the object;
  /* 메시지는 USRA_rel 프로시져에서 처리되어진다 */
  othercell → Drt=1 ; succeed +=1; }}
  return(succeed);  }
```

[알고리즘 3-5] 셀 선택 함수 알고리즘

[알고리즘 3-6]는 복제판단 함수의 알고리즘으로 다른 셀에 있는 복제구조가 있는 경우, 이를 테스트하여 성공하면 1을 리턴하고, 그렇지 않으면 0을 리턴하는 알고리즘이다.

```
 int switch( )
 { int succeed = 0;
 while ((succeed==0) && (exists an ummarked othercell))
{ if (2*(othercell → NoR + othercell → NoW) >
     cell → sumR + cell → sumW)
/* 메시지는 USRA_s 프로시져에서 처리된다 */
 othercell → Drt = 1;delete the local replica;
 deallocate counters; succeed = 1;  }
   else  mark othercell;  }
   return (succeed);  }
```

[알고리즘 3-6] 복제판단 함수 알고리즘

```
int Contraction( )
  /* 동시에 복제 데이터를 처리하는 함수 */
{ find a othercell such that (othercell → Drt ==1);
 if (othercell → NoW > cell → sumR)
{ send the "exit" message to othercell, and wait
      for a response;}
    if (the response is "exit")
 delete the local replica; deallocate the counters;
    else if (an "exit" request from othercell is
received)
    { if (cell → data < othercell → data)
 { delete the local replica; deallocate the counters;
    send a message to othercell "not exit"; }
      else{ send a message to othercell "exit";
          othercell → Drt=0; } } }
```

[알고리즘 3-7] 복제처리 함수 알고리즘

```
Void USRA_rel(data)
  /* 다른셀로 재배치하라는 메시지를 받았을 경우, 처
리하는 프로시져 */
  { save the data in the local database;
    allocate replica server and initialize to 0; }
```

[알고리즘 3-8] 재배치 알고리즘

```
Void USRA_s
/* 다른 셀로부터 재배치 메시지를 받았을 경우 처리하
는 프로시져 */
{ save the data in the local database;
  data → Drt = 0;
  allocate the counters and initialize to 0; }
```

[알고리즘 3-9] 재배치 후 셀처리 알고리즘

[알고리즘 3-10]은 재배치 후 셀처리 알고리즘으로 다른 셀에서 exit 명령을 받았을 경우, 또는 이동호스트가 2명 이상 복제명령을 실행할 경우, 이를 처리하지 않고, 복제재배치 수행을 종료하는 알고리즘이다.

```
Void USRA_x(data)
{ send a message to the othecell
  data "exit";
  data → Drt = 0; }
```

[알고리즘 3-10] 수행종료 알고리즘

3.2 비용 계산식

다음은 복제 재배치에 대한 비용계산을 하는 알고리즘이다. (그림 3)에서와 같이, N개의 셀이 있고, 각각의 셀에는 복제 서버가 존재한다. 그리고 n명의 이동 사용자들이 있고, 그들은 각각 다른 소유의 데이터를 가지고 접근을 한다. 이동사용자가 존재할 확률 $P(k)$는 다음 <식3-1>과 같이 된다.

$$p(\bar{k},j)=\begin{cases} if\, k<(\frac{n}{N}),\ 0, & \text{갱신} \\ otherwise \binom{N}{j} & \text{갱신} \\ \text{곱} p(k_1=k,\cdots,k_j=k), & \end{cases} \quad <식3-1>$$

여기서, $p(\bar{k},j)$는 j개의 셀에 존재하는 k명의 이동 사용자가 있는지를 나타내고, 복제된 셀에 존재하는 것임을 나타낸다.

각각, 이동 사용자들은 독립적으로 움직이고 있고, 이동 사용자들은 불규칙적으로 다른 셀들을 이동할 수가 있는 것이다.

여기서, 이동 사용자의 데이터에 접근 요구는 λ로 포아송 분포에 의해 되어진다. 그들의 범위내에서, 이동 호스트로부터의 접근들은 α의 비로 나타내었다($0\le\alpha\le1$). 갱신의 비율은 이동 호스트와 이동사용자에 대해서, 각각 γ_{mhost}는($0\le\gamma_{mhost}\le1$) 그리고 γ_{muser} 는 ($0\le\gamma_{muser}\le1$)로 나타내었다. 여기서 접근 비용은, 셀간에 접근 요구가 있다면, 그것의 비용은 1이고 그렇지 않을 경우는 0이 된다. 여기서 복제 재배치는 셀 간의 복제서버를 복사함으로 수행되어질 수 있다. 이럴 경우, 복제 재배치의 비용은 1이다.

또한 이동 사용자의 위치를 복제 서버에서는 이동 경로를 알고 있어야 한다. 이것은 M(k,α)라고 표기하였다. 여기서 비용은 셀의 수에 따른 이동 사용자의 수에 의해서 계산되어지기 때문에 복제된 데이터에 대한 접근 ($\frac{n}{N}$)보다 적은 셀에서는 존재하지 않게 된다. 여기서 N은 셀의 수이고, n은 이동 사용자의 수이다. 그러므로, 실제 복제 된 셀에서 존재하는 이동사용자의 수가 k명이라면, $k<\frac{n}{N}$일 경우에는 0이되어 수행이 되지 않고, 그렇지 않으면, 1이 리턴되어 그 셀에 대한 경로를 가지게 된다. 여기서, 복제 셀에서의 이동 사용자의 수가 ($\frac{n}{2}$)보다 작을 때에는 결과가 나오지 않는다. 왜냐하면 이동 사용자의 수가 1이라는 의미이기 때문이다. 그래서 이런 경우를 제외하고는

3가지 상태가 나올수 있는데, 그것은 $k>\frac{n}{2} or\ k<\frac{n}{2}$,일 때와 $\frac{n}{N}\le k<\frac{n}{2}$, $k=\frac{n}{2}$,일 때이다. 그래서 셀에 수에 따른 이동 사용자에 대한 비용 계산 알고리즘은 다음과 같다. 여기서 변수 δ(α)는 셀의 수에 따라 이동 사용자의 수가 많으면 1이 리턴되고, 그렇지 않으면 0을 리턴한다.

Void USRA_cost(k) { I **if**($k>\frac{n}{2}$ or $k<\frac{n}{2}$,) **then**

$M(k,\alpha)=0$} **else** { **if**($\frac{n}{N}\le k<\frac{n}{2}$,)

$$M(k,\alpha)=k\mu\overbrace{\sum_{j=0}^{\frac{n}{k}}p(\overline{k},j)}^{\text{전송}}+\sum_{l=0}^{\binom{n-k}{k-1}}p(\overline{k},1,\overline{k-1},l)$$

$$+\left\{\frac{l}{N-1}+\delta(\alpha)\frac{l(N-1-l)(1-\beta(k))}{(N-1^2)}\right\}$$

$$+\sum_{l=0}^{\frac{n-k}{k-1}}\sum_{m=0}^{\frac{n-k-l(k-1)}{k-2}}p(\overline{k},1,\overline{k-1},l,\overline{k-2},m)$$

$$+\delta(\alpha)\left\{\frac{m\beta(k)}{k(N-1)}+\frac{m(1-\beta(k))}{(N-1^2)}\right\}\overbrace{+\mu\sum_{j=2}^{\binom{n}{k}}p(\overline{k},j)(n-2k)\frac{j-1}{N-1}}^{\text{전송}}$$

$$+\delta(\alpha)(1-\beta(k))\sum_{j=1}^{\binom{n}{k}}\sum_{l=0}^{\binom{n-jk}{k-1}}p(\overline{k},j,\overline{k-1},l)$$

$$+(n-2k+1)\left\{\frac{l}{(N-j)(N-1)}+\frac{l(N-j-l)}{(n-jk-l(k-1))(N-1)(N-j)}\right\}\overbrace{\ }^{\text{전송}}\}$$

else { if($k=\frac{n}{2}$) **then**

$$M(k,\alpha)=k\mu\overbrace{p(\overline{k},1,\overline{k-1},1)}^{\text{전송}}\left\{\frac{1}{N-1}+\delta(\alpha)\frac{(1-\beta(k))(N-2)}{(N-1)}\right\}$$

$$+p(\overline{k},1,\overline{k-2},1)\delta(\alpha)\left\{\frac{\beta(k)}{k(N-1)}+\frac{1-\beta(k)}{2(N-1)}\right\}\overbrace{\ }^{\text{전송}}\}\}$$

[알고리즘 3-11] 비용계산 알고리즘

그래서, 각각 이동 호스트와 이동 사용자 그리고 복제 재배치에 대한 비용 계산식은 다음 <식 3-2,3,4,5>와 같이 나온다.

$$R_{user}=\sum_{k=0}^{n}p(k)(1-\alpha)\lambda\left\{\beta(k)\frac{n-k}{n-1}+(1-\beta(k))\frac{n-1-k}{n-1}\right\}$$

<식3-2>

$$W_{user}=\sum_{k=0}^{n}p(k)\gamma_{user}(1-\alpha)\lambda\left\{2\beta(k)\frac{n-k}{n-1}+(1-\beta(k))\left\{\frac{k}{n-1}+\frac{3(n-1-k)}{n-1}\right\}\right\}$$

<식3-3>

$$W_{Usr}=\sum_{k=0}^{n}p(k)(1-\beta(k))\gamma_{Usr}\alpha\lambda$$ <식3-4>

$$M_{replica}=\sum_{k=0}^{n}M(k,\alpha)$$ <식3-5>

4. 성능 평가

4.1 이동사용자 접근비용

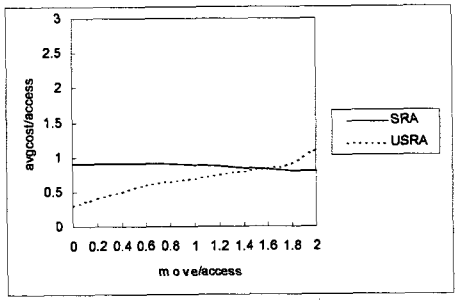
(그림 4와 5)는 0부터 2까지의 이동 매개변수들의 값의 증가로서, SRA기법과 USRA기법의 접근 비용을 보여주고 있다. 여기서 이동비율은 0부터 2λ까지 발생한다. (그림 4)에서 이동 호스트 접근 비율은 모든 접근을 5%로 제한 시켰다, 이것의 의미는 이동 사용자가 데이터에 드물게 접근했다는 의미이다. (그림 5)에서는 이동 호스트의 접근 비율이 모두 40%로 제한시켰다, 이것의 의미는 이동 사용자가 데이터에 자주 접근 했다는 의미이다. 이동 비율이 증가함으로서, 이 두가지 경우를 보면, SRA기법의 평균 비용은 USRA기법보다 3배정도 증가했음을 알 수 있다. 또한, SRA기법의 평균비용은 이동 매개변수에 독립적이었음을 알 수 있다.

이동 호스트의 이동성이 낮았을 때, 제안된 기법의 비용을 보면, USRA기법은 SRA기법보다는 낮게 측정되었다. (그림 4)에서, USRA기법의 접근 비용은 이동 매개변수의 값이 1.6이상일 때부터 SRA기법보다 크게 되었음을 알 수 있다. (그림 5)에서는 매개변수 값이 1.4이상일 때부터 이런 현상이 나타나고 있다. 이런 현상은 다음과 같이 설명되어질 수 있다. 이동 매개변수의 값이 1.6보다 작을 경우에는 USRA기법에서 복제 재배치로서의 비용은 이동 호스트로부터 접근되는 비용과 복제 셀에서 존재하는 이동 사용자들이 아주 적기 때문에 이런 현상이 나타난다. 그러나 이동성이 증가됨으로서, 빠르게 복제 재배치가 이루어짐으로 비용이 증가되는데, (그림 4)에서 보듯 이동 매개변수가 1.6에서 같게 되고 , 그 이후에는 빠르게 증가하고 있음을 알 수 있다. (그림 5)에서는, 1.4부터 이런 현상이 나타난다. USRA기법에서, 이동 사용자가 접근하는 비용은 이동 매개변수 값이 1.6이상이었을 경우를 제외하고는 복제 재배치로서 비용은 그다지 높게 나타나지는 않았다.

(그림 4와 5)를 비교하여, 이동적인 관점에서

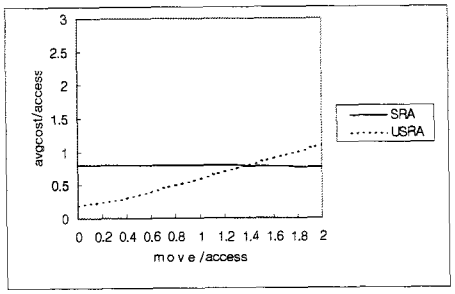
보았을 경우, 이동 호스트가 증가함에도 접근 비율이 적을 수록 USRA기법이 SRA기법보다 더 좋다. 이런 현상은, 이동 호스트가 같은 셀에 계속해서 존재할수록 접근 비용이 낮아진다는 의미이고, 한 셀에 이동 호스트가 복제 서버에 트랜잭션을 일으킨 후, 셀을 이동한다면, 접근 비용은 높게 측정된다.

그러므로 한 셀에 있는 이동 사용자의 수에 관계 없이 낮은 접근 비용을 얻을 수 있다.



(그림 4) 접근비율 5%일때의 이동성과 접근비용

(Fig.4) mobility and access cost when access rate 5%.

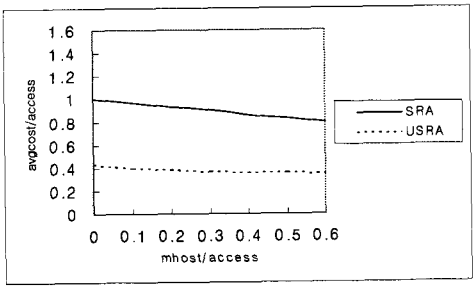


(그림 5) 접근비율 40%일때의 이동성과 접근비용

(Fig. 5) mobility and access cost when access rate 40%

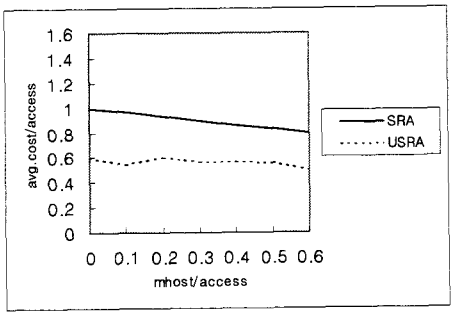
4.2 이동 호스트 접근률에 따른 접근비용

(그림 6과 7)은 이동호스트의 접근비율이 0에서 0.6까지 변화되는 것으로부터, 접근 비율을 2가지 기법에 적용하여, 접근 비용을 보여주고 있다. (그림 6)에서는 이동 매개변수의 값이 0.25로 나타냈는데, 이것의 의미는 이동 비율이 접근 비율과 비교하여 상대적으로 낮음을 의미한다. (그림 7)에서는 이동 매개변수 값이 0.75로 나타냈는데, 이것의 의미는 이동 비율이 접근 비율보다 높음을 의미한다. 이 두상황에서, SRA와 USRA 기법은 같은 경향이 있는데, 그것은 평균단가가 이동 호스트의 접근 비율이 증가할수록 점점 낮아지고 있음을 알 수 있다. 여기서, USRA기법은 소유자 접근 비율에 대한 모든 값들에 대해 최소의 비용이 나타남을 알 수 있다.



(그림 6) 이동률(0.25)이 낮을 때의 이동 호스트 접근률과 접근비용

(Fig.6) mobile host access rate and access cost when low of mobiliy(0.25)



(그림 7) 이동률(0.75)이 높을때의 이동 호스트 접근률과 접근비용

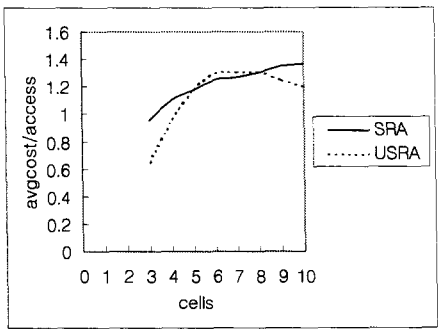
(Fig.7) mobile host access rate and access cost when high to mobility(0.75)

4.3 셀 수에 의한 접근비용

(그림 8과 9)는 네트워크 위치가 변화될 때

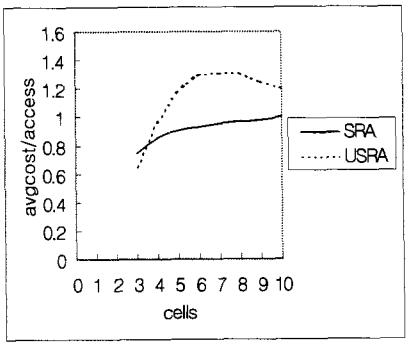
SRA기법과 USRA기법의 접근 비용을 보여주는데, 이것은 셀의 수에 대한 변화를 나타내고 있다. 여기서 USRA기법에서 접근 단가는 SRA기법과 비교하여 네트워크 변화에 민감함을 알 수 있다. 이것은 다음과 같이 설명할 수 있다. 셀의 수가 증가될 때, 복제 셀도 증가하기 때문에 이런 현상이 나타난다고 할 수 있다. 복제 셀로부터 복제이후에 셀간의 연결을 필요로 하는데, 이것은 이 위치에서 접근 단가의 증가가 나타남을 의미한다. 그러나, USRA기법에서 SRA기법과 비교하여 접근 비율이 높을때는 좋지 않게 나타나는데, 그곳은 사용자의 이동이 복재 재배치에 의존하기 때문에 높게 나타나게 된다.

이것은 복제 셀 에서, 사용자 증가와 다른 셀 에서도 다른 사용자 수의 증가에 의해서 이런 증가되는 현상이 나타나고 있다. 그래서, 비용은 복재 재배치의 증가에 있다고 볼 수 있다. 실제적으로 매개변수가 주어지면, 이 셀의 수가 6개나 5개일 때 3명의 이동사용자가 있을 때, 높게 나타나게 된다. 여기서, USRA기법의 비용은 셀의 수가 6일때가 가장 나쁘게 나타났다. 그러나, 셀의 수가 6개 이상일때에 비용은 이 위치에서 보다는, 점점 낮아지고 있음을 알 수 있다.



(그림 8) 데이터에 접근비율(5%)이 낮을때에 셀의 수에따른 민감도

(Fig.8) A sensitivity on the number of cell when low of access rate(5%)



(그림 9) 데이터 접근 비율(50%)이 높을 때의 셀에 수에 따른 민감도

(Fig.9) A sensitivity on the number of cell when high to access rate(50%)

5. 결론

본 논문에서는 이동 데이터베이스에 대한 복제 재배치 기법을 제안하였다. SRA 기법은 이동 호스트가 이동한 셀에 복제되어 사용되는데, 이것은 쉽게 복제해서 사용할수 있다는 장점은 있다. 그러나 이동한 셀에 이동 사용자나 이동 호스트가 존재하지 않을 경우에는 공유되지 못하는 데이터가 되므로 이동률이 높아질 가능성이 있다. 그러나 본 논문에서 제안된 USRA기법은 이동호스트의 이동성에 관계없이 이동 사용자가 많은 셀을 선택하여 복제되므로 많은 이동사용자의 이동률을 감소시킬 수가 있었다. 이 기법은 주로 이동 컴퓨팅 환경에서 적용해야 되고, 이동 사용자가 다른 이동 사용자의 데이터에 접근을 허락하는 곳에서 이루어져야 한다.

본 논문에서는 각각의 기법에 대한 모델을 분석하였고, 이 모델을 기반으로 하여 비용을 비교하여 성능평가를 하였다. SRA와 USRA기법에서 비용을 비교한 결과, 다음과 같은 결론을 얻을 수 있었다.

본 논문에서 제시한 복제 재배치 기법은 이동성이 낮을 때, USRA기법은 5%정도 낮게 나타났음을 알 수가 있었다. 그 이유는, USRA와 SRA기법과 비교해서, 이동 호스트의 이동성에 대한 효과로서, 접근비율이 10%정도일 때에는

이동성에 대한 접근 비용이 2% 감소가 되었고, 접근 비율이 100%정도일때는 이동성에 대한 접근 비용이 2.5%정도 감소되었음을 알 수 있었다.

마지막으로, 셀의 수에 따른 민감도는, USRA기법이 확장성은 셀의 수에 따라서, 1.7%정도 떨어짐을 알 수 있었다. 본 논문에서는 이동 컴퓨팅 환경에 대한 데이터베이스 시스템에서 트랜잭션 경영을 포함해서 연구하였다. 향후 이동 컴퓨팅 시스템에 적용시키는 것이 연구방향이다.

참 고 문 헌

[1]. Ada Fu et al., "Dynamic Policies in Selecting a Caching Set for a Distributed Mobile Computing Environment", May 13, 1995.

[2]. Bernstein P. A., V. Hadzilacos and N. Goodman, "Concurrency Control and Recovery in database Systems", Addison Wesley, Reading, Massachusetts, 1987.

[3]. B.R.Badrinath and T.Lmielinski, "Replication and mobility" Proc. of the 2th Workshop on the Management of Replicated Data, 1992.

[4]. C.D.Tait and D.Duchamp, "Service interface and replica management algotithm for mobile file system clients" Proc. of the Parallel and Distributed Information Systems Conference, 1991.

[5]. Evaggelia Pitoura et al., "Revising Transaction Concepts for Mobile Computing", proc. IEEE Workshop on Mobile Systems and Applications, Dec 1994.

[6]. Stefano Ceri, Giuseppe Pelagatti, "Distributed Databases Principles and Systems", McGraw-Hill, 1984.

[7]. T.Lmielinski and B.R.Badrinath, "Data Management for mobile computing," the ACM SIGMOD RECORD. vol.22, no.1, pp.34-39, 1993.

[8]. T.Lmielinski and B.R.Badrinath, "Mobile wireless computing, solutions and challenges in data management." Technical Report DCS-TR-296. Department of Computer Science, Rutgers University, U.S.A. 1993.

[9].Vivek R. Narasayya, "Distributed Transactions in a Mobile Computing System", CSE552, pp. 1-13, March 9, 1994.

최강희

1988.3-1993.2 관동대학교 정보처리과 이학사

1993.3-1995.2 관동대학교 전자계산공학과 공학석사

1996.9-2000.2 관동대학교 전자계산공학과 공학박사

2001.3-2006.4 현재 대원과학대학 컴퓨터정보계열 교수

관심분야 :Mobile DB, 전자상거래