최소 중복을 이용한 Hotspot 시간 데이터의 관리

강지형* 윤홍원

*신라대학교

Management Strategy of Hotspot Temporal Data using Minimum Overlap

Ji-hyung Kang* Hong-won Yun

Silla University
E-mail : hwyun@silla.ac.kr

요 약

이 논문에서는 과학적 응용에서 발생하는 중복을 이용하는 시간 데이터를 관리하는 방법을 제안한다. 먼저, 시간 데이터를 구분하는 경계값 LB와 RB를 정의하고 과거, 현재, 미래 세그먼트에 각각 저장되는 개체 비전을 정의하였다. 또한 Hotspot 통로를 가지는 시간 데이터에 대하여 각 세그먼트 사이에 이동하는 알고리즘을 제안하였다. 이 논문에서 제안하는 최소중복을 이용한 이동 방법과 기존 방법에 대하여 성능을 비교하였다. 정의에 대한 평균 응답 시간에서는 기존의 방법과 비교해 결과를 보였다. 제안한 이동 방법은 세그먼트 사이에 중복에서 저장되는 데이터 수를 적게 하므로 공간을 사용하는 기존의 이동 방법보다 효율적이었다.

ABSTRACT

We propose a strategy to manage temporal data which are occurred on scientific applications. Firstly, We define LB and RB to separate temporal data, and entity versions to be stored in past, current, future segments. Also, We describe an algorithm to migrate temporal data with hotspot distribution among segments. The performance evaluation of average response time and space utilization is conducted. Average response time between two methods is similar, and space is saved in proposed method.

키워드

Temporal data management, Data migration method

1. 서 론

시간 데이터의 보기는 없는 응용 비즈니스 응용과 과학적 응용으로 분류할 수 있다. 기존의 데이터 이동 방법들은 일반적인 비즈니스 응용의 데이터를 바탕으로 이루어졌다. 과학적 응용에서 보기는 되는 전반적인 시간 데이터의 특성은 물리 성질을 둔 수 있는데, 이러한 성질에 대한 반응으로 나타난 시간 데이터는 일정한 시간 동안 고정을 이루고 이산적인 특성을 가지고 있다.[1,2]. 이 논문에서는 과학적 응용에서 발생하는 시간 데이터의 특성을 고려해 시간적으로 연관성이 많은 개체 비전을 관리하는 방법에 대해서 살펴본다. 먼저, 시간 데이터를 구분하고, 현재, 그리고 미래 데이터로 구분하는데 사용하는 경계값을 정의한다. 그리고, 각 세그먼트에 저장되는 개체 비전을 정의하고 데이터를 이동하는 알고리즘을 나타낸다. 제안하는 시간 데이터 이동 방법과 기존의 이동 방법 사이에 성능을 비교하고 평가한다.

II. Hotspot 시간 데이터

개체 비전이 중복을 이루는 데이터에 대해서는 개체 비전 사이에 연관성을 고려할 필요가 있다[1,4,5]. 중복은 이루어 데이터에서 임의의 시점 집단이 주어질 때 임의의 개체 비전 집단을 만족하려면 그 개체 비전의 유효 시간 간격이 집단의 시점을 포함해야 한다. 임의의 시점 집단에 대해서 임의의 두 개체 비전이 동시에 선택되려면 두 개체 비전의 유효 시간이 겹치야 한다. 두 개체 비전의 겹치는 정도와 두 개체
비전이 동시에 선택될 확률은 일정한 관리온이 있다. 직관적으로 두 개의 비전의 유 효 시간이 많이 겹칠수록 동시에 선택될 확률은 높을 수 있다. 그림 1에서는 개체 비전 P1과 P2가 겹치는 시간을 A1이라고 하고 P2와 P3이 겹치는 시간을 A2라고 하자. A1시간이 A2시간보다 길다고 가정하면 임의의 시점 2의 대체된 P1과 P2가 동시에 선택될 확률이 P2와 P3가 동시에 선택될 확률보다 큼을 알 수 있다. 임의의 두 개체 비전이 동시에 선택될 시간은 다음과 같이 나타낼 수 있다.

\[ R(p) = \bigcup \{Vs, Ve\}, R(lc) = \bigcup \{Vs, Ve\}, R(lf) = \{i.e, l\} \]

따라서 과거 세그먼트와 현재 세그먼트 그리고 미래 세그먼트가 포함하는 모든 개체 비전의 집합은 다음과 같이 정의할 수 있다.

<table>
<thead>
<tr>
<th>세그먼트</th>
<th>저장되는 개체비전 집합</th>
</tr>
</thead>
<tbody>
<tr>
<td>과거 세그먼트</td>
<td>( P = {i.e, Ejy, Ve \leq LB} )</td>
</tr>
<tr>
<td>현재 세그먼트</td>
<td>( C = {Eij</td>
</tr>
<tr>
<td>미래 세그먼트</td>
<td>( F = {Eij, Ve \geq RB} )</td>
</tr>
<tr>
<td>과거와현재 세그먼트</td>
<td>( PC = {Eij</td>
</tr>
<tr>
<td>현재와미래 세그먼트</td>
<td>( CF = {Eij</td>
</tr>
</tbody>
</table>

따라서 과거 세그먼트와 현재 세그먼트와 미래 세그먼트가 포함하는 모든 개체 비전의 집합은 다음과 같이 정의할 수 있다.
\( \text{LB} = \min \{ \text{stil} | \text{Ol(t1)}, \text{Ol(t2)}, \text{Ol(tn)} \} \)

\( \text{If} \)의 임의의 시작에서 오른쪽으로 가장 가까운 유효 끝 시작을 \( \text{nirget}(\text{If}) \)라고 하고, 다음과 같이 정의한다. \( \text{nirget}(\text{If}) = \min \{ \text{If}, \text{Ve} \in \text{E}(\text{If}) \} \)

\( \text{If} \)의 임의의 시작에서 오른쪽으로 가장 가까운 유효 끝 시작에서 걸치는 유효 시간 간격의 개수는 \( \text{Of}(\text{nirget}(\text{If}))) \)로 나타낼 수 있다. 새로운 \( \text{RB} \)를 구하는 시점을 \( t \)라고 하면 새로운 \( \text{RB} \)를 구하는데는 해당 되는 시간 영역은 \( \text{시점} \text{t} \)에서 \( \text{LB} \)까지의 시간 길이를 \( t \)를 중심으로 잡았을 때 \( \text{LB} \)가 만나는 시작점, \( \text{RB} \)를 구하기 위한 구간의 길이 시점으로 한다. \( \text{시점} \text{t} \)에서 \( \text{LB} \)까지의 길이 \( t - \text{LB} \)가 된다. 새로운 \( \text{RB} \)를 구하는데는 해당되는 영역은 시점 \( \text{t} \)에서 \( \text{t} + 1 \)인, 이를 \( \text{S}(\text{If}) \)라고 하고 다음과 같이 정의한다.

\( \text{S}(\text{If}) = \{ \text{t} + 1, \text{t} + 1 \} \)

새로운 \( \text{RB} \)를 구하는 시점을 \( t \)라고 하면, \( \text{t} + 1 \)에서 오른쪽으로 가장 가까운 유효 끝 시작은 \( \text{nirget}(\text{If}) \).

\( \text{S}(\text{If}) \)에서 나오는 모든 유효 끝 시작은 \( \text{nirget}(\text{If}) \)가 된다. \( \text{nirget}(\text{If}) \)를 통해서 나오는 바로 앞의 유효 끝 시작을 차례대로 \( \text{etq} \).

\( \text{etq} \)라고 하면, 값의 범위는 \( t + 1 < \text{etq} \).

\( \text{etq} \)라고 하면, 값의 범위는 \( t + 1 < \text{etq} \).

이 유효 끝 시작 간격의 시작에서 걸치는 시간 간격의 개수는 \( \text{Of}(\text{etq}) \).

\( \text{Of}(\text{etq}) \), \( \text{Of}(\text{etq}) \)로 나타낼 수 있다. 이 중에서 가장 최소값이 되는 유효 끝 시작은 현재 세그먼트와 미래 세그먼트를 구간하는 경계 값 \( \text{RB}(\text{If}) \)가 되고, 다음과 같은 다음과 같다.

\( \text{RB} = \min \{ \text{etq} | \text{Of}(\text{etq}), \text{Of}(\text{etq}) \} \)

\( \text{Get LB} \)

begin

" for all \text{If} in Future Segment and Current Segment

\( \text{LBpre} = \text{LB}; \quad \text{RBpre} = \text{RB}; \quad \text{LB}, \text{RB}, \text{TABLE}[] \) = null;

while (not eof (StartTime < (TABLE[k++]))) do

if ( \( \text{LBpre} < \text{EntityVersion}, \text{Vs} < \text{RBpre} \)) then

Append (TABLE[k++], EntityVersion, Vs);

end if

end while

while (not eof (EntityVersion < (TABLE[k++]))) do

if ( EntityVersion < read (FunSeg))) do

Append (TABLE[k++], EntityVersion, Vs);

end if

end while

while (not eof (StartTime < (TABLE[k++]))) do

end if

end while

while (not eof (EntityVersion < (TABLE[k++]))) do

end if

end while

end while

Overlap < Overlap + 1;

end if

end while

Overlap < Overlap + 1;

end if

end while

end while

LB = \min \{ \text{EntityVersion}, \text{Vs} | \text{TABLE}[k], Overlap \};

return \text{LB}

지연관계상 \( \text{RB} \)를 구하는 알고리즘은 생략한다.

표 2 이동 및 복사 대상 개체비전

<table>
<thead>
<tr>
<th>작업 세그먼트</th>
<th>해당 개체비전</th>
</tr>
</thead>
<tbody>
<tr>
<td>이동 미래 → 현재 (Ei, Vs ≥ LB) ∨ (Ei, Vs &lt; RB)</td>
<td></td>
</tr>
<tr>
<td>현재 → 과거 Ei, Vs ≤ LB</td>
<td></td>
</tr>
<tr>
<td>복사 미래 → 현재 (Lb &lt; Ei, Vs &lt; RB) ∨ (Ei, Vs &lt; RB)</td>
<td></td>
</tr>
<tr>
<td>현재 → 과거 Ei, Vs ≤ LB ∨ (Ei, Vs &lt; RB)</td>
<td></td>
</tr>
</tbody>
</table>

데이터를 이동하는 과정에서 발생하는 작업은 이동과 복사로 나눌 수 있다. 표 2는 최소중복에 의한 데이터 이동 방법에서 데이터를 이동할 때 수행하는 이동, 복사의 대상이 되는 개체 비전을 나타내고 있다.

VI. 성능평가

상성 평가 모델은 질의 생성기, 질의 처리기, 테이터 이동 프로세스로 구성된다. 질의 생성기는 질의와 병렬로 질의를 수신 본포에 의해서 만들어내고 만들어진 질의는 질의 대기 큐에 들어간다. 질의 처리기는 질의 대기 큐에서 하나의 질의를 거내며 질의 점의 인가, 병렬질의인가를 구분하고 검색할 세그먼트를 결정한 뒤에 해당 세그먼트를 임니다.

데이터 이동 프로세스는 테이터 이동 방법에 따라 현재 세그먼트와 미래 세그먼트를 검색해서 다른 세그먼트로 송가나 복사한다. 데이터 이동 프로세스가 현재 세그먼트와 미래 세그먼트를 검색하고 이동하는 중에 발생한 모든 질의는 데이터 이동이 완료될 때까지 질의 대기 큐에서 기타리고 데이터 이동이 끝날 때 까지 질의를 계속 처리한다.

최소 중복에 의한 이동 방법은 일정한 시간 동안 군집을 이루는 시간 데이터에 적합한 이동 방법으로, 모의 실험 데이터는 추적적으로 군집을 이루는 핫스팟 분포를 이용해서 생성하였다. 군집의 군집에는 1% 내지 2% 정도 유형이 들어 가도록 하였고, LTL은 1% 와 5%로 각각 만들었고 폐사의 군집이 균등 분포로 생성하였다. 이 모의 실험 데이터는 LTL의 비율을 1% 와 5%로 각각 생성하였다.

![그림 2: 평균응답시간](attachment:image2)
그림 3. 중복 저장되는 데이터의 수

그림 2는 본리 저장 구조에서 LLT가 1%이고 시간 절이가 10%, 30%, 그리고 50%인 경우에 평균에 의한 이동 방법과 최소 중복에 의한 이동 방법에 대해 사용자 절이에 대한 응답 시간을 측정한 것이다. 두 가지 이동 방법은 시간 절이의 비율이 같은 경우에 사용자 절이에 대한 응답 시간이 거의 비슷하게 나타나고 있다. 이것은 평균에 의한 이동 방법과 최소 중복에 의한 이동 방법은 현재 세그먼트의 크기가 비슷하고 데이터 이동 반도가 비슷하기 때문이다.

사용자 절이에 대한 응답 시간에서 보면, 평균에 의한 이동 방법과 최소 중복에 의한 이동 방법에서 비슷한 성능을 보이지만, 최소 중복에 의한 방법은 두 세그먼트에 중복되는 개체 비전의 수를 작게 하므로 공간 이용율의 측면에서 효율적이다. 그림 3은 현재 세그먼트에 대해서 중복 저장되는 개체 비전의 수를 백분율로 나타낸 것이다. 최소 중복에 의한 이동 방법은 사용하는 실체의 데이터를 한 세그먼트에 저장함으로써 서로 시간적으로 연관성이 있는 개체 비전을 묶을 수 있기 때문에 실제 사용의 특성을 그대로 반영하는 특징을 가지고 있다.

V. 결론

과학적 응용에서 발생하는 구조를 이루는 시간 데이터에 적합한 이동 방법으로서 최소 중복에 의한 이동 방법을 제안하였다. 최소 중복에 의한 이동 방법에서 시간 데이터를 구분하는 경계값 LB와 RB를 정의하고 각 세그먼트에 저장되는 개체 비전을 정의하였다. LB와 RB를 구하는 방법을 보이고, 세그먼트 사이에 데이터를 알고리즘을 보였다. 과학적 응용에서 발생하는 구조를 이루는 시간 데이터에 적합한 이동 방법으로 제안한 최소 중복에 의한 이동 방법은, 절이에 대한 평균 응답 시간에서 평균에 의한 이동 방법과 비슷한 결과를 보였다. 최소 중복에 의한 이동 방법은 세그먼트 사이에 중복해서 저장되는 데이터 수를 적게 하므로 공간 이용율 측면에서는 평균에 의한 이동 방법보다 효율적이다.

참고문헌