통합 XQuery 질의의 병렬처리와 순차처리 성능분석

강순종[0] 박종현 강지훈
충남대학교 컴퓨터공학과
{sjgains[0], jhpark, jhkang}@cnu.ac.kr

Analysis of Parallel and Sequential processing for integrated XQuery query

Soon-jong Kang[0], Jong-hyun Park, Ji-hoon Kang
Dept. of Computer Engineering, Chungnam National University

요 약

XML 문서의 검색을 위한 질의 언어인 XQuery는 다양한 데이터 소스로부터 가져온 고유한 구조를 가진 질의 결과로 구성할 수 있도록 설계되어 XML질의 언어의 표준이 되었다. XQuery를 이용해 특별히, 분산 환경에서 다중 XML문서 를 대상으로 하는 통합 질의의 경우, 질의 처리 계획을 결정하는 것은 처리 효율과 직결된다. 따라서 질의 처리 계획을 결정하는 요소 중 하나인 조인 처리 방법의 연구는 중요하다. 그러나 통합 질의에서 조인구조를 기준으로 단일 XML문서에 대한 질의 처리방법을 결정하는 것은 쉽지 않다. 본 논문에서는 분산환경에서 다중 XML문서를 대상으로 하는 조인을 포함한 다양한 통합 질의를 대상으로 실험을 통해 병렬처리 방법과 순차처리 방법 그리고 두 가지 처리방법을 조합한 하이브리드 방법을 적용하여 처리 시간을 비교 분석하고, 다중 문서에 대한 효율적인 조인방법과 순서를 모색한다.

1. 서 론

XQuery[1]는 W3C의 XML Query Working Group에서 권고하는 질의 언어로서 모든 형식의 XML 데이터에 질의할 수 있게 최적화된 지능적이고 뛰어난 언어다. 사용자는 XML 데이터 형식과 연결된 메서드를 사용하여 XML 데이터 형식의 변수와 열에 대한 질의를 실행할 수 있다. 이러한 XQuery를 이용해 사용자는 단일 및 다중 XML문서에 조인질의를 할 수 있다.

특별히, 분산환경에서 다중 XML문서를 대상으로 하는 질의의 경우, 질의 처리 계획은 처리 효율과 직결된다. 다시 말하면, 어떤 질의 처리 계획을 결정하는가는 분산 XQuery 질의의 효율적인 처리를 위해서 반드시 필요한 요소이다. 질의 처리 계획을 결정하기 위한 중요한 고려사항 가운데 하는 XQuery에 존재하는 조인이다. 즉, 조인을 포함한 다중 XML문서에 대한 통합 질의의 경우 질의처리 속도를 향상시키기 위해 조인방법과 순서를 결정하는 것은 매우 중요하다. 그러나 통합 질의에서 조인구조를 기준으로 각각의 단일 XML문서에 대한 질의 처리 방법과 순서를 결정하는 것은 쉽지 않다.

본 논문에서 우리는 다중 XML문서를 대상으로 기술된 XQuery질의의 효율적인 처리 방법을 모색하기 위하여 통합 질의의 순차처리 방법과 병렬처리 방법의 성능을 비교 분석한다. 또한, 두 가지 질의처리 방법의 성능향상을 위해 질의 구조를 중심으로 지역 질의들의 조인 횟수와 지역 시스템의 접근 횟수 그리고 독립 적으로 처리 가능한 지역 질의 선택을 고려한 방법을 연구한다.

본 논문의 구성은 다음과 같다. 2절은 관련연구를 알아보고, 3절에서는 병렬처리 방법, 순차처리 방법 그리고 하이브리드 처리방법을 예제를 통해 알아본다. 4절에서는 각각의 방법들을 실험을 통해 성능평가를 하고, 5절에서 향후 연구계획과 결론을 내린다.

2. 관련연구

다중 XML문서에서 XQuery 질의의 구조정보를 이용한 조인 질의처리 순서결정에 관한 연구들은 이미 수행되었다.[2] 그러나 이러한 방법들은 질의의 구조정보를 이용해 질의의 실행 순서만을 결정할 뿐 질의처리 방법은 다루고 있지는 않다. 반면, 관계형 데이터베이스 시스템에서는 분산 데이터베이스 상의 SQL 질의 처리와 조인 질의 처리위해 병렬 처리방법을 이용한 연구들이 활발히 이루어 져왔다.[3],[4],[5],[6]

위의 연구들은 질의의 구조정보를 이용해 통합 질의를 지역 질의로 분할하고, 분할된 질의들을 병렬처리 함으로서 순차처리의 단점인 속도의 한계를 극복하기 위한 방법을 제시하였다. 그러나 조인 질의를 포함하고 있는 통합 질의에서는 병렬처리의 결과로 얻어진 지역 질의의 모든 중간결과에 대해 질의처리기 혹은 지역 데이터베이스 시스템이 다시 조인 연산을 수행해야 하는 단점이 있다. 이와 마찬가지로 XQuery 질의에서 병렬처리 방법만을 이용할 경우 조인 질의 처리에 있어 중간질의 처리기 혹은 지역 질의 처리기에 너무 많은 부담을 준다.

3. 통합질의 처리 방법

<그림 1>의 통합 XQuery 질의는 Where 절에서 4개의 다중 XML 문서가 모두 조인되는 질의 구조로 되어 있다. 첫 번째로 예제 질의를 처리하기 위해 For와 Let절에 해당하는 문서를 대상으로 지역 질의로 분할한다. 이때, 통합 질의의 분산질의 처리를 위해 질의 처리 방법과는 무관하게 분할된 지역질의가 필요하다. 두 번째는 각각의 단일 XML 문서에 대한 질의의 선택치가 같거나 알 수 없을 때, 효율적인 질의 처리를 위해 조인 횟수와 지역 시스템의 접근 횟수를 줄이는 질의 처리 방법을 택한다. 많은 경우, XQuery 질의는 그 특성 상 질의를 처리할 때, 지역질의들의 일부는 병렬로 처리해야 되고, 또 다른 지역 질의들의 일부는 순차처리를 해야 하는 경우가 발생 한다.

```
(:For:)
for $people in doc("people.xml")/person
for $open in doc("open_auctions.xml")/open_auction
for $closed in doc("closed_auctions.xml")/closed_auction
for $itemAsia in doc("ItemsOfAsia.xml")/item

(:Let:)
let $pID := $people/@id
let $oID := $open/seller/@person
let $cID := $closed/seller/@person
let $citem := $closed/itemref/@item
let $aItem := $itemAsia/@id

(:Where:)
where $pID=$oID and $oID=$cID and $citem=$Item

(:Return:)
return
<result>{ $people/name, $open/interval,
$closed/buyer, $itemAsia/name }</result>
```

<그림 1> 통합 XQuery 질의

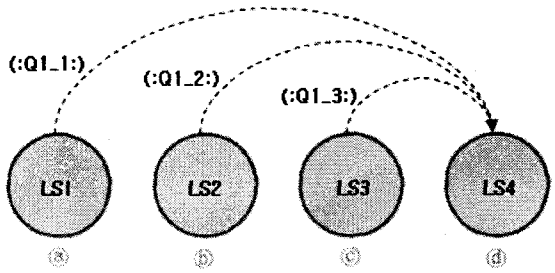
<그림 2>는 통합 XQuery 질의에서 Where 절의 조인을 처리하기 위해 통합 질의로부터 분할한 지역 질의의 예를 나타낸다. 4개의 지역질의는 XML 문서 혹은 XML 데이터베이스가 위치한 지역 시스템으로부터 Return 절에 해당되는 데이터를 추출한다. 이때, 각각의 지역 질의의 선택치를 알고 있다면, 선택치가 높은 지역 질의를 우선 처리할 수 있다.

```
①
for $people in doc("people.xml")/person
let $pID := $people/@id
return <Subresult1>{ $pID }</Subresult1>
②
for $open in doc("open_auctions.xml")/open_auction
let $oID := $open/seller/@person
return <Subresult2>{ $oID }</Subresult2>
③
for $closed in doc("closed_auctions.xml")/closed_auction
let $cID := $closed/seller/@person
let $citem := $closed/itemref/@item
return <Subresult3>{ $cID, $citem }</Subresult3>
④
for $itemAsia in doc("ItemsOfAsia.xml")/item
let $aItem := $itemAsia/@id
return <Subresult4>{ $aItem }</Subresult4>
```

<그림 2> 지역 XQuery 질의

3.1 병렬처리

병렬처리는 통합 질의 처리에 있어 분할된 지역 질의나 조인을 동시에 처리하는 방법이다. <그림 3>은 <그림 2>의 지역 질의를 병렬처리 방법을 이용해 질의 ⓐ, ⓑ,ⓒ,ⓓ를 동시 실행하고, ⓓ에서 조인 연산을 수행 하는 질의 구조를 나타낸다.



<그림 3> 병렬 질의 처리 구조

병렬처리는 지역 질의를 동시에 처리하기 때문에 지역 질의의 처리 시간을 줄일 수 있다. ⓐ,ⓑ,ⓒ,ⓓ의 처리시간은 이들 중 처리시간이 가장 긴 것과 같다. 그러나 지역 질의로부터 얻은 결과를 하나의 질의 처리기에서 조인 연산을 해야 하는 단점도 있다. 이러한 병렬처리의 특징 때문에 조인의 횟수가 많아질수록 성능은 크게 저하 된다.

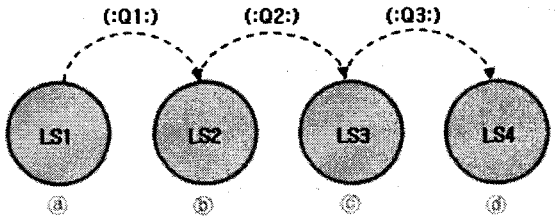
아래 질의는 ⓓ에서 수행되는 조인 질의를 나타낸다.

```
ⓓ
for $pID in (matched Subresult1)
for $oID in (matched Subresult2)
for $cID in (matched Subresult3)
for $aItem in (matched Subresult4)
let $citem := (matched Subresult3)
where  $pID=$oID and $oID=$cID and $citem=$aItem
return <Subresult5>{ $pID, citem }</Subresult5>
```

3.2 순차처리

순차처리는 통합 질의 처리에 있어 분할된 지역 질의나 조인을 순서대로 처리하는 방법이다. <그림 4>는

<그림 2>의 지역 질의를 순차처리 방법을 이용해 질의 ⓑ,ⓒ,ⓓ 각각의 지역 시스템에서 조인 연산을 수행 하는 질의 구조를 나타낸다.



<그림 4> 순차 질의 처리 구조

순차처리에서 조인 연산을 거듭 할수록 조인 처리 속도는 빨라진다. 왜냐하면, 첫 번째 조인으로 축약된 결과가 다음 조인 질의의 For나 Let 절에 쓰이기 때문이다. 또는, 최악의 경우 위의 질의 처리 구조에서 ⓑ,ⓒ,ⓓ의 조인 결과가 축약되지 않는다고 할지라도 하나의 지역 시스템에서 처리하는 조인의 횟수는 병렬처리 방법의 지역 시스템 ⓓ에서 처리하는 조인의 횟수 보다 작다. 그러나 조인 질의 처리의 분산으로 하나의 지역 질의 처리기에 가해지는 부담은 줄었지만, ⓒ의 조인 질의 처리를 위해 ⓑ의 조인 결과가 나올 때까지 대기해야 되는 단점이 있다. 마찬가지로 ⓓ의 조인 질의 처리를 위해 ⓒ의 결과가 나올 때까지 대기해야 된다. 아래 질의는 ⓑ,ⓒ,ⓓ에서 수행되는 조인 질의를 나타낸다.

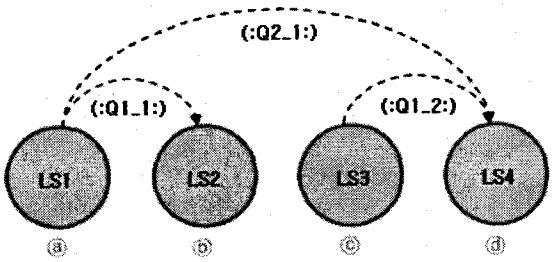
ⓑ
```
for $open in doc("open_auctions.xml")/open_auction
let $oID := $open/seller/@person
for $people in (matched person/@id)
let $pID := $people/@id
where $oID=$pID
return <Subresult2>{ $oID }</Subresult2>
```

ⓒ
```
for $closed in doc("closed_auctions.xml")/closed_auction
let $cID := $closed/seller/@person
let $cItem := $closed/itemref/@item
for $open in (matched Subresult2)
let $oID := $open/@person
where $cID=$oID
return <Subresult3>{ $cID, $cItem }</Subresult3>
```

ⓓ
```
for $itemAsia in doc("ItemsOfAsia.xml")/item
let $aItem := $itemAsia/@id
for $closed in (matched Subresult3)
let $cItem := $closed/@item
where $aItem=$cItem
return <Subresult4>{ $aItem }</Subresult4>
```

3.3 하이브리드처리

하이브리드처리는 통합 질의 처리에 있어 병렬처리와 순차처리를 혼합한 방법이다. <그림 5>는 ⓐ와 ⓒ의 지역 질의를 동시에 수행하고, 그 결과를 ⓑ와 ⓓ에서 동시에 조인 연산을 수행하는 질의 구조를 나타낸다.



<그림 5> 하이브리드 질의 처리 구조

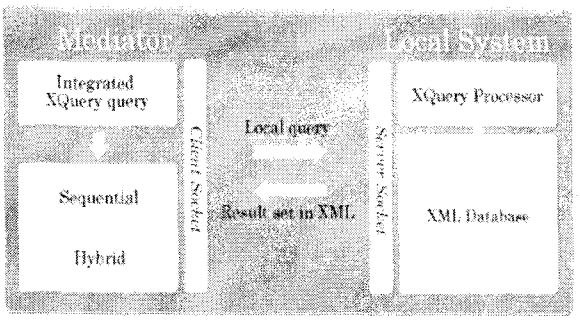
병렬처리의 장점은 지역질의와 조인을 동시에 실행함으로서 질의의 처리 시간을 줄일 수 있는 것이었고, 순차처리의 장점은 조인 연산을 분산함으로서 질의 처리 속도를 향상 시키는 것이었다. 위의 질의 처리 구조에서 지역 질의 처리(ⓐ,ⓒ)와 조인(ⓑ,ⓓ)을 동시에 처리함으로서 지역 시스템의 접근 횟수와 조인 횟수를 줄일 수 있다. 아래 질의는 ⓑ,ⓓ에서 수행되는 조인 질의를 나타낸다.

ⓑ
```
for $open in doc("open_auctions.xml")/open_auction
let $oID := $open/seller/@person
for $people in (matched person/@id)
let $pID := $people/@id
where $oID=$pID
return <Subresult2>{ $oID }</Subresult2>
```

ⓓ
```
for $itemAsia in doc("ItemsOfAsia.xml")/item
let $aItem := $itemAsia/@id
for $closed in (matched closed_auction)
let $cItem := $closed/@item
let $cID := $closed/seller/@person
where $aItem=$cItem
return <Subresult4>{ $cID, $aItem }</Subresult4>
```

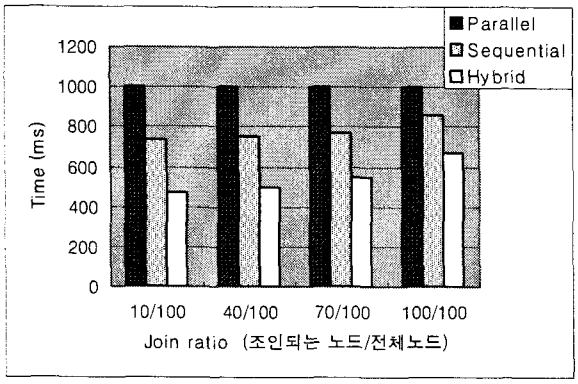
ⓓ
```
for $Id in (matched Subresult2)
for $IdAndItem in (matched Subresult4)
let $pID := $Id/@id
let $cID := $IdAndItem/@person
let $aItem := $IdAndItem/@item
where $pID=$cID
return <Subresult5>{ $pID, $aItem }</Subresult5>
```

4. 성능 평가



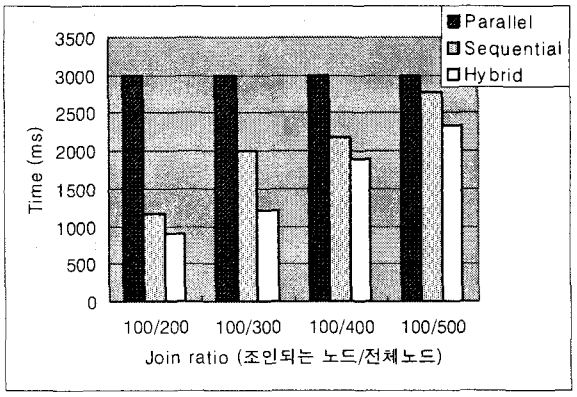
<그림 6> 지역 질의 처리 구조

지역 질의 처리를 위한 전반적인 구조는 <그림 6>과 같다. 정확한 성능평가를 위해 Mediator와 지역 시스템 간의 네트워크간의 신뢰성은 보장되고, 각각의 지역 시스템들의 성능은 동일하도록 구성하였다. <그림 8>은 <그림 1>의 통합 XQuery 질의를 병렬처리, 순차처리 그리고 하이브리드처리 방법을 이용해 실험한 결과이다. 단일 XML 문서의 노드 수는 100개이고 조인되는 노드수를 증가 시키면서 효율성을 측정하였다. 이때, 병렬처리 방법을 이용한 처리시간 측정은 비교 대상에서 제외하였다. 그 이유는 조인되는 노드수가 증가 할수록 처리 시간이 급격하게 증가해 순차처리와 하이브리드처리 방법과의 비교가 무의미 하였다. 실험결과 지역 질의의 병렬처리보다 조인을 분산처리 하는 것이 성능향상에 더 큰 영향을 주는 것을 확인할 수 있었다.



<그림 7> 조인되는 노드 증가에 따른 비교

<그림 8>은 조인되는 노드는 고정 시키고 전체노드를 증가 시켰을 때 질의 처리 방법에 따른 처리 시간을 비교 한 것이다. 노드 수가 증가하더라도 병렬처리와 순차처리를 혼합한 하이브리드처리 방법이 효율적인 것을 확인할 수 있었다. 또한, 두 개 단위의 단일 XML 문서를 대상으로 조인하는 것이 두 개 이상을 조인하는 것보다 더 효율적 이었다.



<그림 8> 전체 노드 증가에 따른 비교

5. 결 론

본 논문에서는 다중 XML문서를 대상으로 효율적인 조인 질의 처리를 위한 방법을 살펴보았다. 조인 질의 처리에 있어 병렬처리 방법만을 이용할 경우 조인 횟수에 의해 질의 처리 시간이 좌우되는 것을 확인할 수 있었고, 하이브리드 방법에서 조인 처리를 위해 두 개 단위의 단일 XML 문서를 대상으로 하는 것이 더 효율적이었음을 알 수 있었다. 또한, 병렬처리 방법에서 모든 조인 연산을 수행하는 지역 시스템이 하이브리드처리 방법을 이용한다면 성능이 더욱 향상 될 것으로 기대된다. 따라서 향후 지금의 하이브리드처리 방법을 발전 시켜 지역 질의는 병렬로 처리하고, 조인 연산은 순차처리와 병렬처리를 혼합한 방법으로 처리 해봄으로서 다중 XML문서를 대상으로 하는 질의 처리 시간을 살펴보고자 한다.

6. 참고논문

[1]. W3C, "XQuery 1.0 and XPath 2.0: An XML Query Language", http://www.w3.org/TR/xquery/.

[2]. 정성호, 정헌석, 임해철: 다중 문서에서 구조 정보를 이용한 XML 조인 질의 처리. 한국정보과학회 29(2):100-102 (2002).

[3]. C.T.Yu, C.C.Chang: Distributed Query Processing. ACM Comput. Surv. 16(4): 399-433 (1984) .

[4]. C.Wang, A.L.O.Chen, S.C.Shyu: A Parallel Execution Method for Minimizing Distributed Query Response Time. IEEE Press. 3(3): 325-333 (1992)

[5]. Yun Jiang: Dynamic Parallel Query Processing for Distributed Objects. IEEE Computer Society. Proceedings of the 9th International Workshop on Database and Expert Systems Applications. 699-(1998).

[6]. D.Kossmann: The State of the Art in Distributed Query Processing. ACM Comput. Surv. 32(4): 422-469 (2000)