NiagaraCQ

Professor : Rachel Potinger
Discussion by Linda li
Presentation by Massih Khorvash

Department of Computer Sciences
University of British Columbia
Note: some of the slides are modified from J. Chen et al.

Overview

- Introduction
- Grouping queries
- NiagaraCQ grouping technique
  - Incremental grouping optimization
  - Query-split scheme
  - Memory Caching
  - Incremental evaluation technique

Introduction

- Continuous Queries
  - frequently changing environments
- Web queries
  - Similar structures

Writing continuous queries in NiagaraCQ language

- **CREATE** *CQ_name*
  XML-QL query
  DO *action*
  {**START** *start_time*} {**EVERY** *time_interval*} {**EXPIRE** *expiration_time*}

Grouping queries

- Share common computation
- Common execution plan
  - Saving I/O costs

NiagaraCQ grouping technique

- Incremental group optimization strategy
- Query-split scheme
- Change-based and timer-based queries
- To ensure scalability:
  - Incremental evaluation of continuous queries.
  - Use of both pull and push models
  - Memory caching.

Incremental Grouping Algorithm

- When a new query is submitted:
  - Match the expression signature with the signatures of existing groups.
  - Group optimizer breaks new query plan into two parts
  - Update constant table
  - If no match, then a new group

```
Where <Quotes>
<Quote>
    <Symbol>AOL</>
    </></>
  element_as $g
in
"http://www.cs.wisc.edu/
db/quotes.xml"
construct $g
```

Figure 3.6 XML-QL query examples

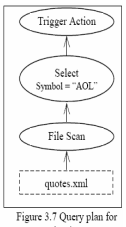


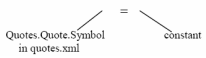
Figure 3.7 Query plan for query in Figure 3.6

Expression Signature

- Represent the same syntax structure, but possibly different constant values, in different queries.
- Expression signatures allow queries with the same syntactic structure to be grouped together to share computation

```
Where <Quotes> <Quote>
          <Symbol>INTC</>
      </> </> element_as $g
in "http://www.cs.wisc.edu/db/quotes.xml"
construct $g
```

```
Where <Quotes> <Quote>
          <Symbol>MSFT</>
      </> </> element_as $g
in "http://www.cs.wisc.edu/db/quotes.xml"
construct $g
```



Expression Signature (Fig. 3.2)

Components of Groups

- *Group signature*
  - in-memory hash table keyed by group signature
- *Group constant table*
  - stored on disk
- *Group plan (next slide)*

Constant value	Destination buffer
....	....
INTC	Dest. i
MSFT	Dest. j
....	....

Figure 3.4 an example of group constant table

Group plan

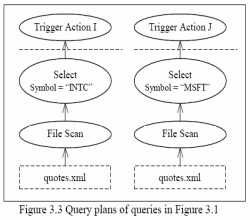


Figure 3.3 Query plans of queries in Figure 3.1

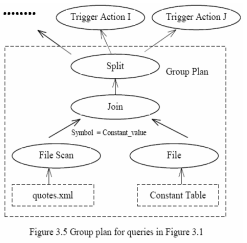


Figure 3.5 Group plan for queries in Figure 3.1

Discussion (1)

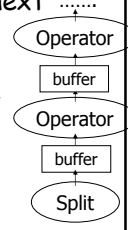
1. Is NiagaraCQ a better application for XML or for relational data? Why?

Query split buffer management

- The destination buffer for the split operator is needed
  - Pipelined scheme
  - Intermediate Files

Pipeline approach

- Tuples are pipelined from the output of one operator into the input of the next operator.
- Main disadvantages:
  - Doesn't work for grouping timer-based CQ's.
  - Bottle-neck



The use of Intermediate Files

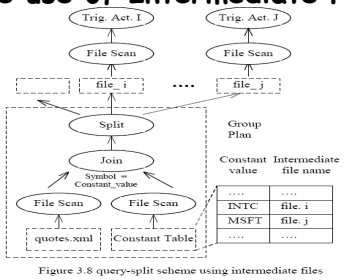


Figure 3.8 query-split scheme using intermediate files

Intermediate Files

- **Advantages**
  - Each query is scheduled independently.
  - The potential bottleneck problem of the pipelined approach is avoided.
- **Disadvantages**
  - Extra disk I/Os.
  - Split operator becomes a blocking operator

Incremental Evaluation technique

- Queries to be invoked only on the changed data.
- "delta file"
- A time stamp is added to each tuple in the delta file.

Issues with Timer-based Continuous Queries

- monitor the timer
- Sharing the common computation
- Intermediate files

Memory Caching

- The queries that didn't fit into any group
- Recently accessed delta files
- "time window"

Discussion (2)

- 1. The authors state "we assume that no more than thousands of groups will be generated for millions of user queries". What kinds of applications can you imagine about this size being used/needed for? Can you imagine extending these techniques to other work, e.g., caching?
- 2. Optional (if time permits) this paper has some conceptual/functional similarities with other systems, e.g. use of time concept, integration of information from various sources. Compare and contrast these things and what are the challenges for this system?