

Generating **Efficient** Execution Plans for Vertically Partitioned XML Databases

Research paper review by

QING Pei, Edward 11500811g
LO Wing Yi, Wing 11523479g
SHAO Shuai, Philip 11552402g

April 10, 2012

What ?

Why ?

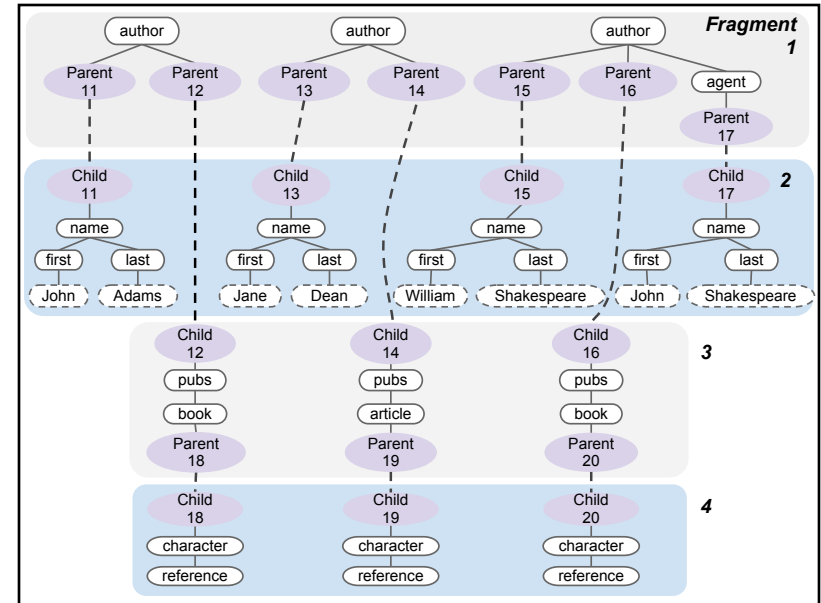
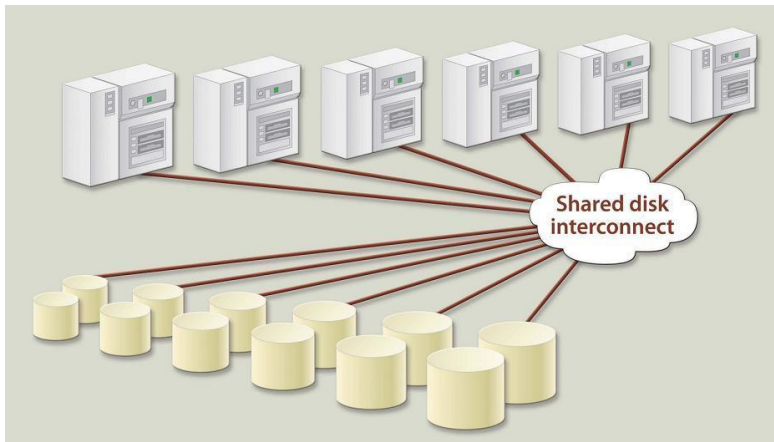
How ?

What ?

Query Processing

	Centralized	Distributed
RDBMS	✓	✓
XML	✓	<i>This paper</i>

XML in the Cloud



Why ?

Distributed architecture
leads to
Different execution plans

For a single query, the **order** in which *joins* are performed results in various time consumed.

$$\begin{aligned} \text{Response time} \\ = \\ \text{local execution time} \\ + \\ \text{joining time} \end{aligned}$$

local execution time

snip(i): the number of document subtrees accessed by the local plan at *fragment i*

smaller *snip(i)* preferred

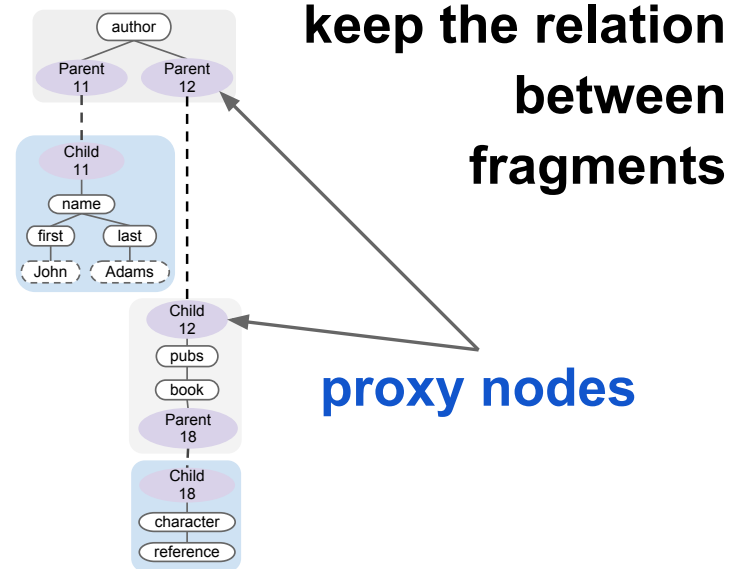
joining time

card(i): the number of tuples that are returned by the local plan when evaluated at *fragment i*

smaller *card(i)* preferred

***Which* plan has the
minimum response time?**

How ?

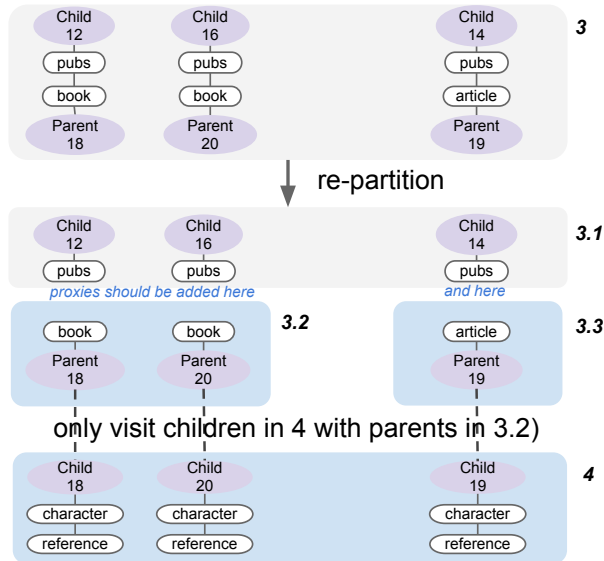


Optimizing distributed plans

Optimizing distributed plans

Pushing Cross-Fragment Joins

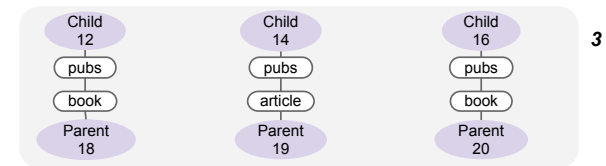
fully works on left-deep plans



Optimizing distributed plans

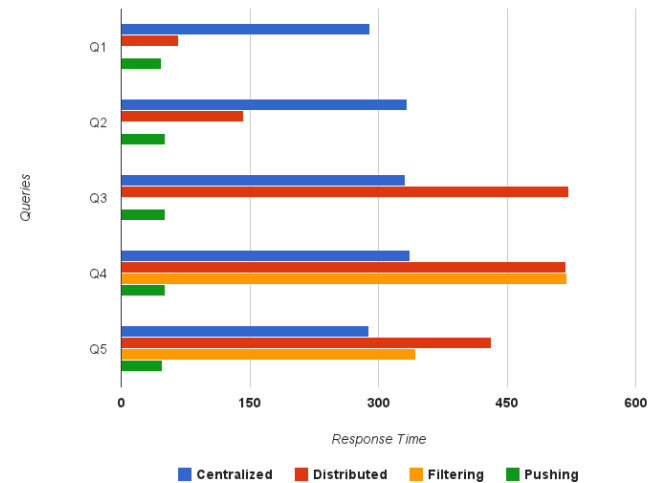
Label Path Filtering

//book//reference

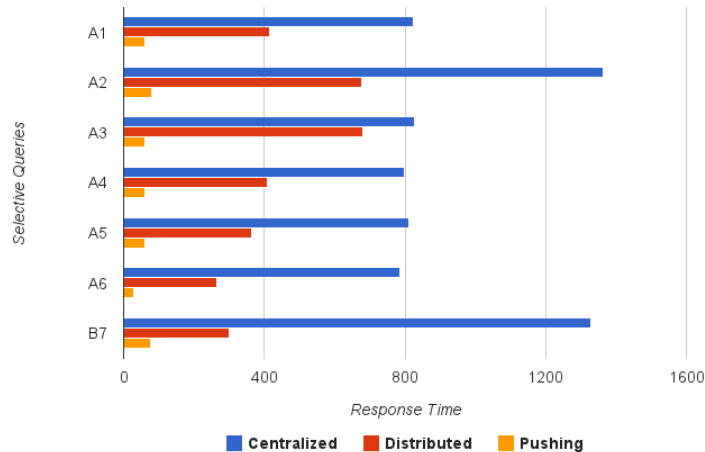


Evaluation

Centralized & Distributed Techniques Comparison (Collection 3.5GB)



**Selective XPathMark Performance Results
(Collection 12GB)**



Conclusion

Greatly improves response time of querying large XML collections.

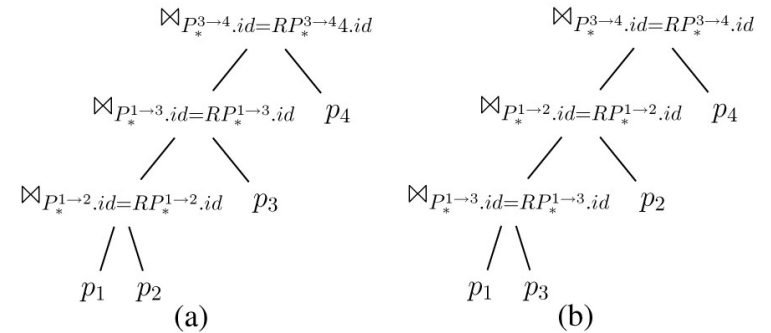
Small overhead. Choosing the fastest plan took **< 0.01** seconds.

Q & A

Merci beaucoup

Appendix

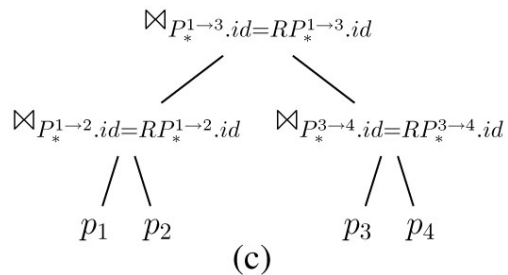
Distributed Execution Plans



left-deep execution plans

Appendix

Distributed Execution Plans



not a left-deep execution plan

Appendix

Queries used for evaluation

- Q1** /open auction[initial > 200]//item//mail/from
- Q2** /open auction[initial > 200][.//author/person/name[starts-with(., 'Ry')]]//item//mail/from
- Q3** /open auction[initial > 200][.//author/person/name[starts-with(., 'Ry')]]//item//category/id
- Q4** /open auction[initial > 200][.//author/person[profile/age > 30]/name[starts-with(., 'Ry')]]//item//category/id
- Q5** /open auction[initial > 200]//author/person[starts-with(name, 'Ry')]/profile/interest/category/description

Appendix

Queries used for XPathMark

- A1** /site/closed auctions/closed auction/annotation/description/text/keyword
- A2** //closed auction//keyword
- A3** /site/closed auctions/closed auction//keyword
- A4** /site/closed auctions/closed auction [annotation/description/text/keyword]/date
- A5** /site/closed auctions/closed auction[descendant::keyword]/date
- A6** /site/people/person[profile/gender and profile/age]/name
- B7** //person[profile/@income]/name

Appendix

Queries used for Selective XPathMark

- A1S** /site/closed auctions/closed auction[price > 600] /annotation/description/text/keyword
- A2S** //closed auction[price > 600]//keyword
- A3S** /site/closed auctions/closed auction[price > 600] //keyword
- A4S** /site/closed auctions/closed auction[price > 600] [annotation/description/text/keyword]/date
- A5S** /site/closed auctions/closed auction[price > 600] [descendant::keyword]/date
- A6S** /site/people/person[starts-with(name, 'Ry')] [profile/gender and profile/age]/name
- B7S** //person[starts-with(name, 'Ry')][profile/@income]/name

Appendix

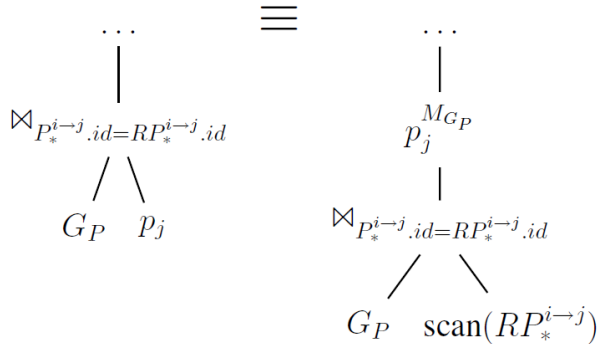


Figure 11: Cross-fragment join pushing rewrite

Appendix

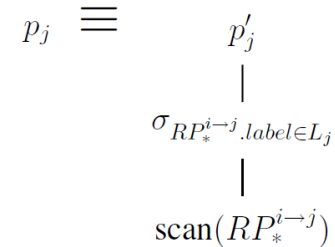


Figure 12: Label path rewrite

Appendix

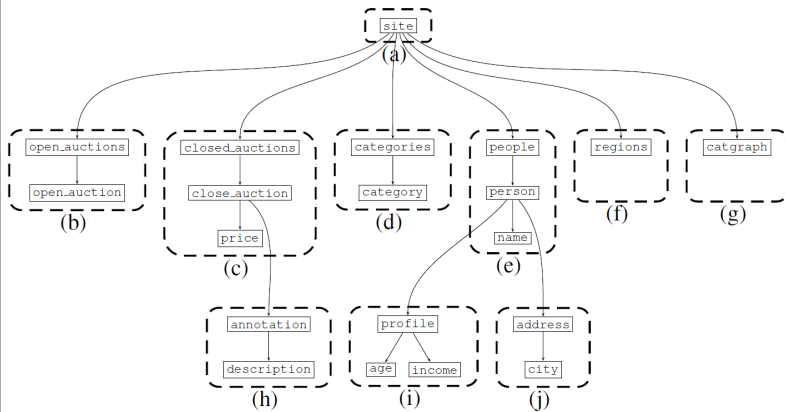


Figure 13: Fragmentation schema used in second experiment

Appendix

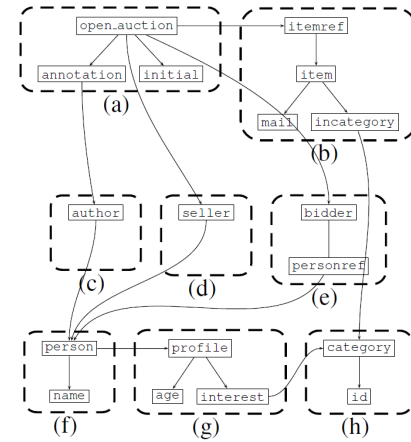


Figure 14: Fragmentation schema used in first experiment