Lucent Technologies
Bell Labs Innovations

StatiX: Making XML Count

Juliana Freire*
Prasan Roy
Jerome Simeon
Bell Labs - Lucent Technologies

Jayant Haritsa
Maya Ramanath
Indian Institute of Science

Motivation

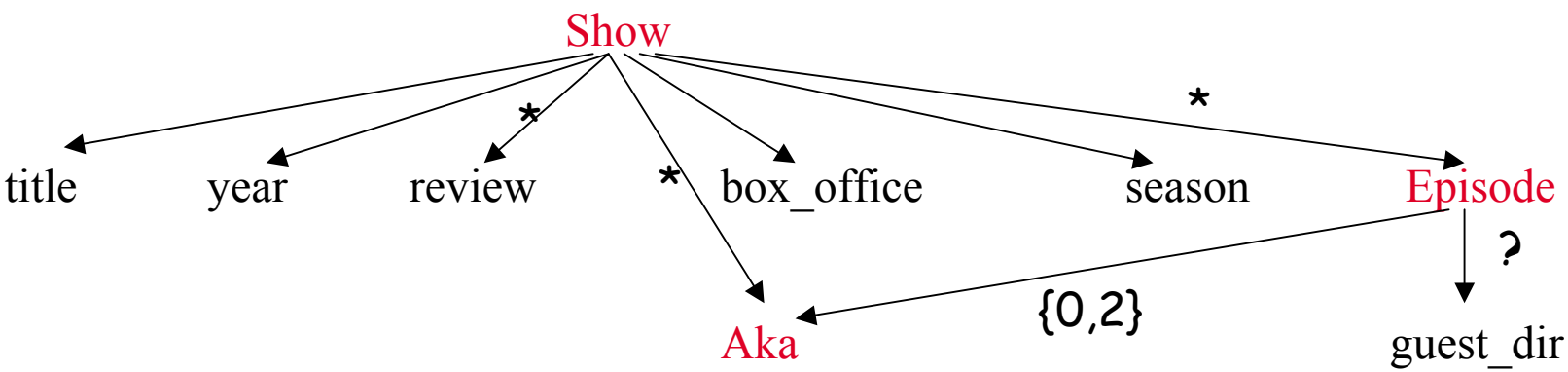
Lucent Technologies
Bell Labs Innovations

- Statistics to estimate cardinality of queries
  - Query optimization
  - Exploratory queries
  - Cost-based storage design for XML data
- Relational
  - Fixed and flat structure
  - Need information about distribution of values
- XML
  - Flexible and nested structure
  - Need information about both values and structure

Example: IMDB Schema

Lucent Technologies
Bell Labs Innovations

```
type Show =
    show [ title [String], year [Integer], review[String]*, Aka *,
           ( box_office [Integer] |
            ( season [Integer], Episode * ) ) ]
type Aka = aka [String]
type Episode = episode [ Aka{0,2}, guest_dir [String]? ]
```



IMDB Sample Data

Lucent Technologies
Bell Labs Innovations

```
<show>
  <title> Fugitive, The </title>
  <year> 1993 </year>
  <review>
    best action movie of the decade...
  </review>
  <review>
    Ford and Jones at their best...
  </review>
  <review>
    top notch action thriller...
  </review>
  <review>
     Solid action and great suspense...
  </review>
  <aka> Auf der Flucht </aka>
  <box_office> 183752965 </box_office>
</show>
```

```
<show>
  <title> Seinfeld </title>
  <year> 1990 </year>
  <review>
    The best comedy series ever!
  </review>
  <seasons> 9 </season>
  <episode>
     <aka> The Soup Nazi </aka>
     <aka> The Soup </aka>
  </episode>
  <episode>
     <aka> Good Samaritan, The
     </aka>
     <guest_director>
       Alexander, Jason
     </guest_director>
  </episode>
  <episode>
     <aka> Gum, The </aka>
  </episode>
</show>
```

Selectivity of XML Queries

Lucent Technologies
Bell Labs Innovations

List all akas for show

```
/show/aka
```

List all akas for episodes

```
/show/episode/aka
```

Capture structure of tags

List all akas and reviews for movies released after 1991

```
FOR $s in
    document ("imdb.xml")/sho
$a in $s/aka, $r in $s/review
WHERE $s/year > 1991
RETURN $s/title, $s/box_office, $a, $r
```

Capture distribution of values over structure

- How to concisely capture structural and value skew?
- How to generate accurate estimates?

Related Work

Lucent Technologies
Bell Labs Innovations

- McHugh & Widom, VLDB 1999: all subpaths of length up to k
  - For "longer" queries, combine info about subpaths
- Chen et al, ICDE 2001: capture correlation between paths
  - Use set hash signatures to represent paths
  - Compute selectivity by combining signatures
- Aboulnaga et al, VLDB 2001: compress path information
  - Focus: small summaries
  - Coalesce/delete elements with low frequency
- Polyzotis & Garofalakis, SIGMOD 2002: estimates for graph-structured data
- Wu et al, EDBT 2002: use histograms to capture ancestor/descendent relations

Related Work

Lucent Technologies
Bell Labs Innovations

	Summary	Stats	Supported Queries	Estimate computation
Chen et al ICDE 2001	Data	Paths + correlations	Tree pattern, no values	Specialized
Aboulnaga et al VLDB 2001	Data	Paths	Simple path, no values	Specialized
StatiX	Schema + Data	Schema types	Tree pattern, values	Classical histogram multiplication

Outline

Lucent Technologies
Bell Labs Innovations

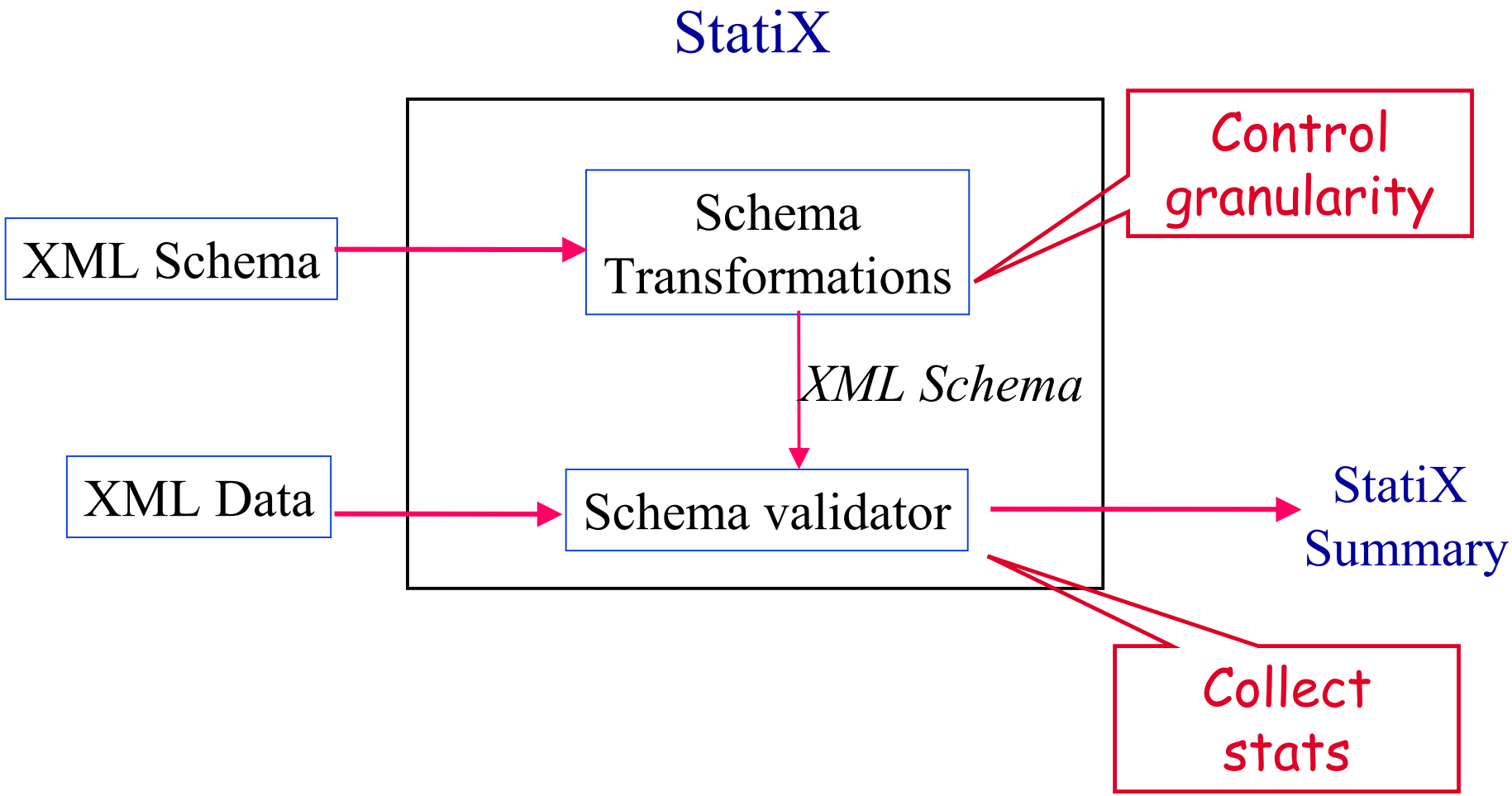
- Motivation
- Related Work
- Overview
- Architecture
- Gathering statistics
- Transformations: flexible granularity
- Applying StatiX to XML storage
- Experiments
- Summary and Future Work

The StatiX Framework

Lucent Technologies
Bell Labs Innovations

- Schema-based stats
  - Use XML Schema to identify sources of skew
- Gather stats about types
  - Piggyback on validation
  - Concise summaries
  - Flexible granularity
- Histograms for summarizing information about distribution of values and structure
  - Well-studied and effective
  - Scalable: adjust to memory budgets
  - Symmetric mechanism for capturing structural and value skews
  - Easy to migrate functionality into relational backend
- Queries
  - Tree patterns, value-based selection and joins

Architecture

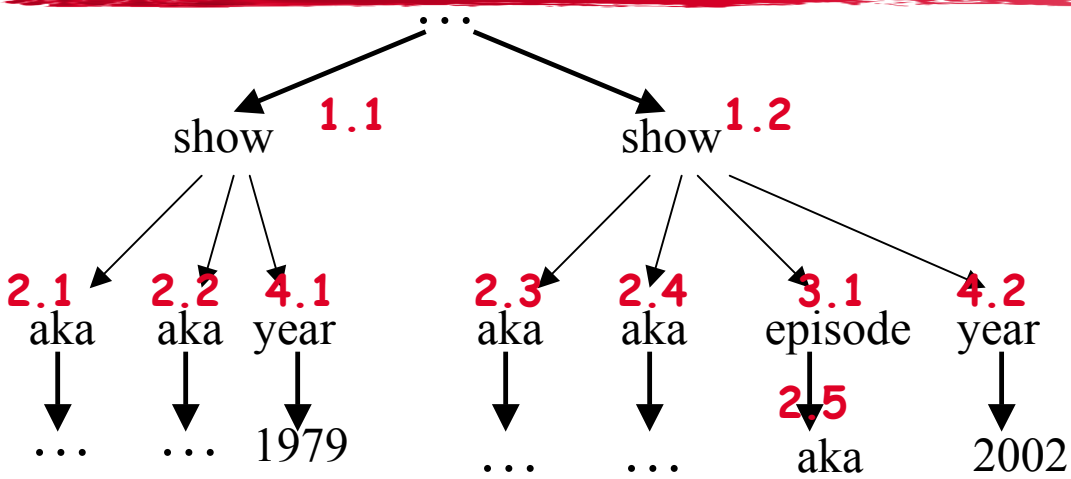


Validation and Statistics Collection

Lucent Technologies
Bell Labs Innovations

- Validation
  - Check document against schema
  - Result: assignment of types to nodes
- Extend validation for statistics collection
- Count occurrences of types and their parents
  - Assign a unique id to each type
  - Keep one counter and a parent set per type
  - For each type instance
    - » Assign a global id: id.counter
    - » Add the id of the parent to the parent set

Gathering Statistics



Type	ID	Parent Set
Show	1	...
Aka	2	1.1 {2} 1.2 {2} 3.1 {1}
Episode	3	1.2 {1}
Year	4	1.1 {1} 1.2 {1} 1979 {1} 2002 {1}

StatiX Summary

Show: card = 2, id =[11,12]

Aka: card = 4, id = [21,25]

parent_histogram:
[11,12] = 4, [31] = 1

Year: card = 2, id = [41,42]

value_histogram:
[1979,2002]= 2

parent_histogram:
[11,12] = 2

Summarization

Estimate Computation

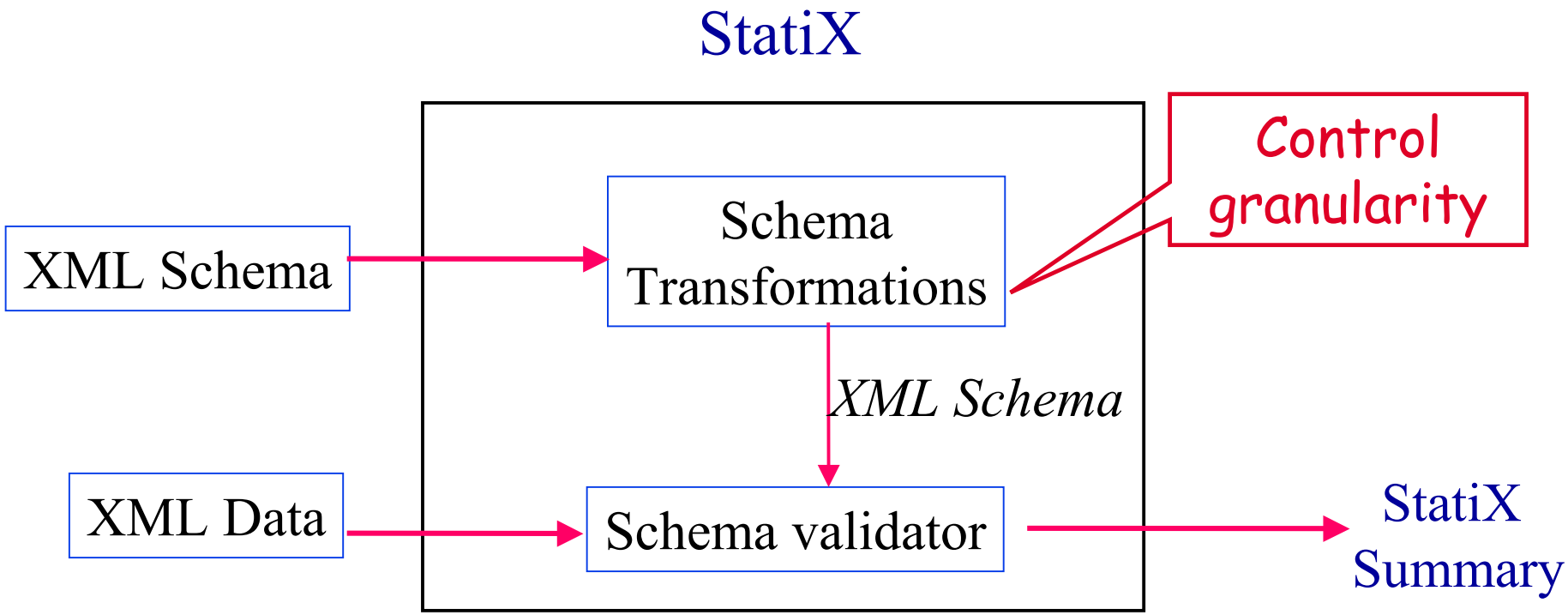
Lucent Technologies
Bell Labs Innovations

```
FOR $s in
   document ("imdb.xml")/show,
$a in $s/aka, $r in $s/review
WHERE $s/year > 1991
RETURN $s/title, $a, $r
```


Histogram multiplication

```
SELECT title, aka, review
FROM Show, Aka, Review
WHERE Show.year > 1991 AND
Show.ID = Aka.ParID AND
Show.ID = Review.ParID
```

Architecture



Tuning Statistics: Schema Rewriting

Lucent Technologies
Bell Labs Innovations

- A given document can be validated by different XML Schemas
  - Different but equivalent regular expressions can be used to define an element
  - The presence or absence of a type name does not change the semantics of an XML Schema
- Control stat collection by:
  - Creating type names for potential sources of skew, I.e., Unions, repetitions, optional elements
  - Additional transformations (LegoDB, ICDE 2002)

Schema Transformations

Lucent Technologies
Bell Labs Innovations

```
type Show = show [
 title [String],  year [Integer],
 review[String]*,  Aka *,
( box_office [Integer] |
( season [Integer], Episode * ) ) ]

type Aka = aka [String]

type Episode =  episode [ Aka{0,2},
   guest_dir [String]? ]
```

Distribution of reviews over shows

```
type Sh
 title [Stri    ], year [Integer],
 Review*, Aka *,
 (Movie|
  TVShow ) ]

type Aka = aka [String]

type Episode = episode [ Aka{0,2},
   GuestDirector?]

type Review = review[String]
...
```

More Transformations…

Lucent Technologies
Bell Labs Innovations

```
type Show = show [
 title [Str
 Review*, Aka *,
 (Movie|TVShow ) ]

type Aka = aka [String]

type Episode = episode [ Aka{0,2},
    GuestDirector?]

type Review = review[String]
...
```

Distribution of reviews over shows

```
type Show1
 title [String], ear [Integer],
 Review*, Aka *, Movie  ]

type Show2
 title [Strin
 Review*, Aka *, TVShow  ]

type Aka = aka [String]

type Episode = episode [ Aka{0,2},
    GuestDirector?]

type Review = review[String]
...
```

Distribution of reviews over shows that are movies

Distribution of reviews over shows that are tv shows

Union Distribution

Summary Sizes

Lucent Technologies
Bell Labs Innovations

$$|summary| = \sum_i (ctypes_i * nbuckets_i * bsize_i) + \sum_j (btypes_j * nbuckets_j * bsize_j)$$

- Size is proportional to:
  - Number of incoming edges into types (ctypes)
  - Number of basic types (btypes)
  - Number of buckets
- 100MB XMark, 30 buckets/hist
  - 125 types: 25KB
  - 1887 types: 200KB (finest grain)
- Limit the number of types during transformation
- Limit the number of buckets

Applying StatiX

StatiX and Cost-Based Storage Design

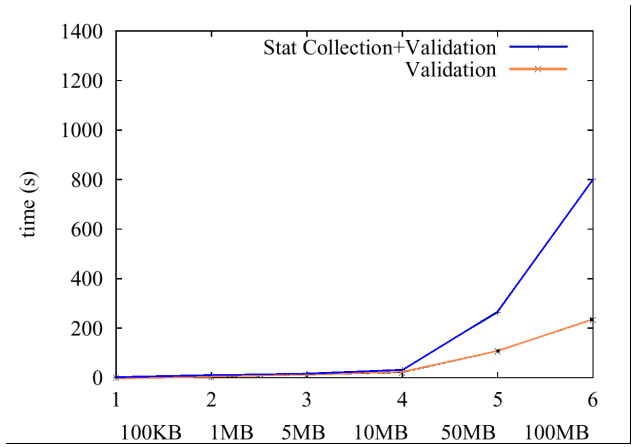
Lucent Technologies
Bell Labs Innovations

- LegoDB (ICDE2002): cost-based XML-to-relational mapping
- Given a schema, query workload, and data sample
  - Explore a set of alternative mappings
  - Select the mapping the leads to lowest cost
- Not practical to gather stats for each configuration
  - Need to derive precise stats for each configuration
- Use StatiX to generate fine-grained stats

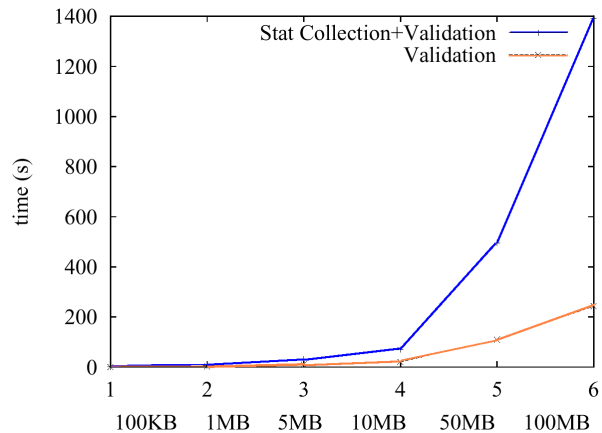
Lucent Technologies
Bell Labs Innovations

Experiments

Statistics Collection: Overheads



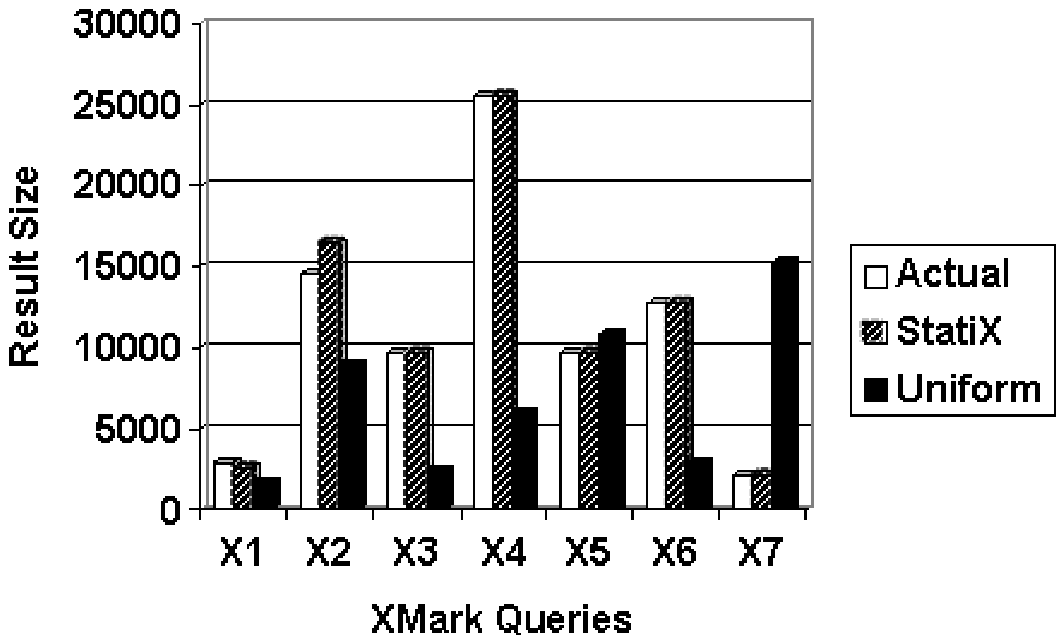
125 types



1887 types

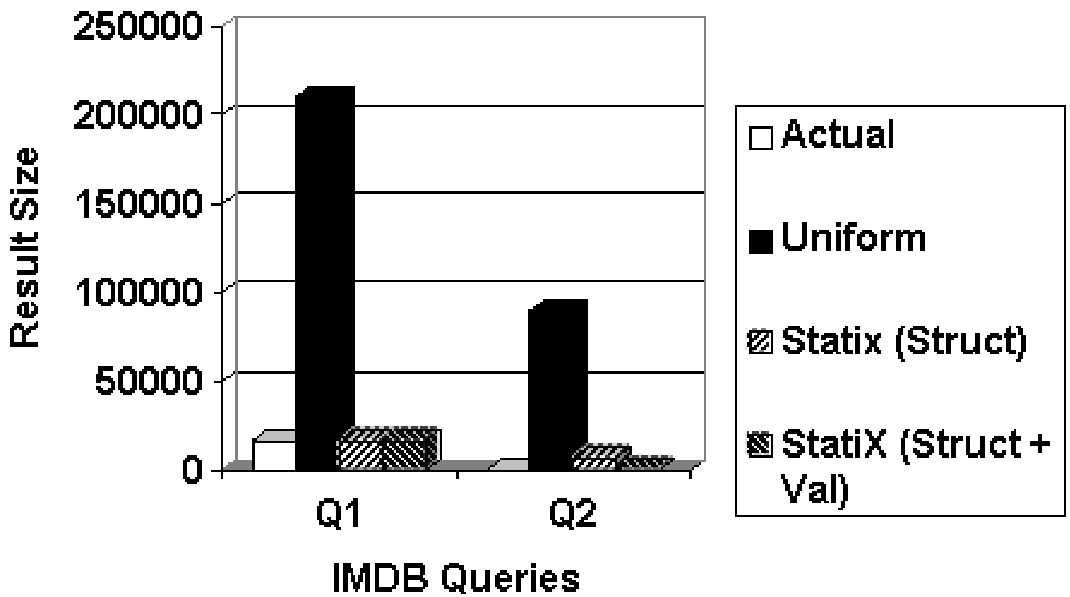
- Stat gathering overhead depends on number of types
- Overheads can be reduced
  - Tune code + Sampling
- One-time only procedure

Accuracy: XMark



- Queries involving value-based and structure-based joins (2-8 joins)
- Uniform distribution leads to large errors
- Chen et al using a summary twice as big lead to 53% error for X7

Accuracy: IMDB



Summary and Future Work

Lucent Technologies
Bell Labs Innovations

- StatiX: leverage XML Schema data model, type transformations, and histograms to generate concise, flexible, and accurate summaries of XML documents
- Differentiators:
  - Handles larger class of queries
  - Localized impact of updates
  - Constructive approach
  - Easy to integrate into relational backend
- Future:
  - Support for recursion and aggregates
  - Ambiguity in validation (union distribution)