XML Indexing I

CPS 216
Advanced Database Systems

Announcements (April 12)

- ❖ Homework #3 due today
  - ▪ Office hours 3-4pm and after 6pm
- ❖ Reading assignment due next Monday
  - ▪ The Selinger paper on query optimization

XML indexing overview

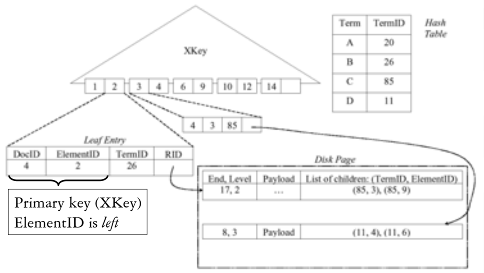
- ❖ It is a jungle out there
  - ▪ Different representation scheme lead to different indexes
  - ▪ Will we ever find the "One Tree" that rules them all?
- ❖ Building blocks: B^+-trees, inverted lists, tries, etc.
- ❖ Indexes for node/edge-based representations (graph)
- ❖ Indexes for interval-based representations (tree)
- ❖ Indexes for path-based representations (tree)
- ❖ Indexes for sequence-based representations (tree)
- ❖ Structural indexes (graph)

Warm-up: indexes in Lore (review)

- ❖ Label index: (child, label) → parent
  - ▪ B^+-tree
- ❖ Edge index: label → (parent, child)
  - ▪ B^+-tree
- ❖ Value index: (value, label) → Node
  - ▪ B^+-tree
- ❖ Path index: path expression → node
  - ▪ Structural index: DataGuide (more in next lecture)

Niagara: data manager index

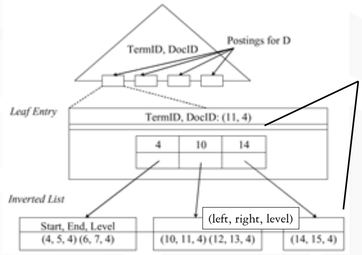
- ❖ A combination of node/edge-based and interval-based representations using B^+-tree



Primary key (XKey)
ElementID is *left*

Niagara: index manager index

- ❖ Essentially an inverted-list index for tag names with entries in each list sorted by XKey



(left, right, level)

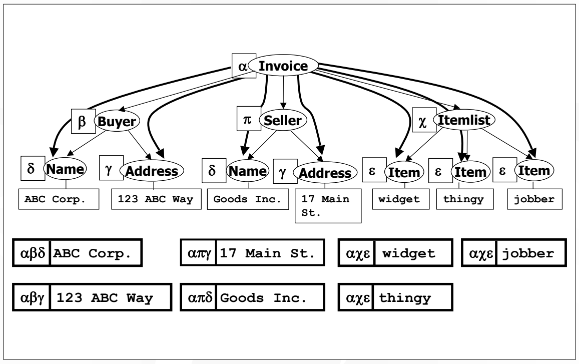
A sorted list,
or a sorted list
with a sparse index,
or a B^+-tree

Index Fabric: a path-based index

Cooper et al. “A Fast Index for Semistructured Data.” *VLDB* 2001

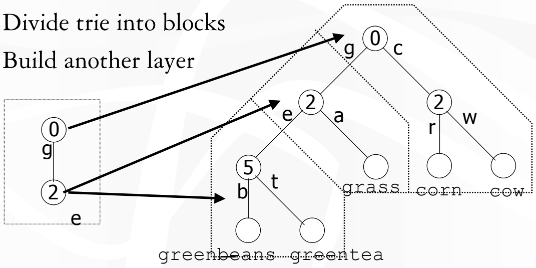
- ❖ Use a label-path encoding for XML
  - ▪ Each element is associated with a sequence of labels on the path from the root (e.g., `/Invoice/Buyer/Name/ABC Corp.`)
  - ▪ Encode the label path as a string (e.g., `/Invoice/Buyer/Name` → $\alpha\beta\delta$)
- ❖ Index all label paths in a Patricia trie
  - ▪ And try to make the trie balanced and I/O-efficient

Example of label paths in Index Fabric



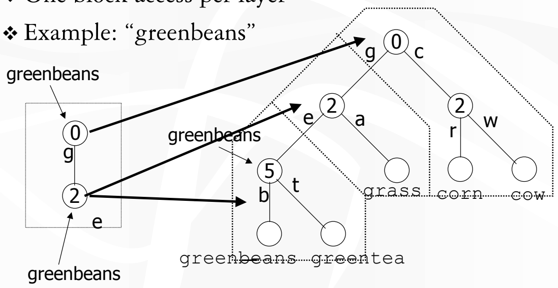
Balancing Patricia trie in Index Fabric

- ❖ Recall that Patricia trie indexes first point of difference between keys
- ❖ Divide trie into blocks
- ❖ Build another layer



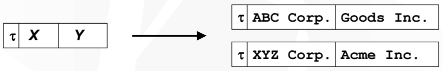
Searching Patricia trie in Index Fabric

- ❖ Start searching in the root layer
- ❖ One block access per layer
- ❖ Example: “greenbeans”



Refined paths in Index Fabric

- ❖ Queries supported by Index Fabric so far:
  - ▪ Label paths from the root (e.g., `/Invoice/Buyer/Name/`)
  - ▪ How about `//Buyer/Name`, or `//Buyer/Name|Address`?
- ❖ Refined paths: frequent queries
  - ▪ Just invent labels for these queries and index them in the same Patricia trie
  - ▪ Example: find invoices where *X* sold to *Y*



☞Extra refined paths → more space required