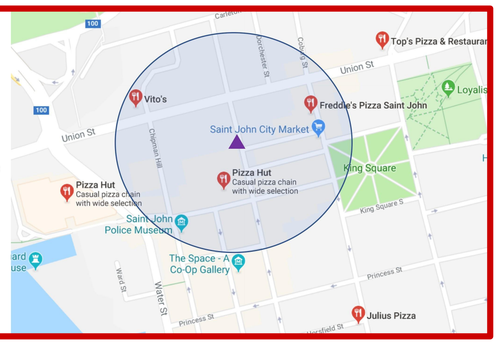
Memory Efficient Spatio-Textual Search with Concurrent Updates

Yoann S.M. Arseneau, Suprio Ray, Bradford G. Nickerson
Faculty of Computer Science, University of New Brunswick, Fredericton, New Brunswick, Canada

Question

How can we perform efficient spatio-textual search on big data?

Example query:
Find pizza parlors within walking distance



Common Solutions

Most solutions separate search components resulting in potentially slower searches

For example: acquire all pizza-related results, then filter them by distance. All work done retrieving spatially irrelevant results is effectively wasted.

Our Solution

The Spatio-Textual InterLeaved Tree (STILT)

- Binary tree similar to a trie
  - Path to a leaf produced by interleaving components (example below)
  - Single child paths are compressed
- Does not contain data, only document IDs
  - Backed by I/O-efficient LSM-tree store [1]
- Each path represents 63 bits; 21 bits of text and 42 bits of location

How to index a document

An index is generated for each unique word of a document
2:(46, 79; Bar, Alcohol) → (46, 79; Alcohol), (46, 79; Bar)

The components are converted to binary strings
(46, 79; Alcohol) → (4, 7, a) → (0100, 0111, 0001)

The binary strings are combined with interleaving
(0100, 0111; 0001) → 000 110 010 011

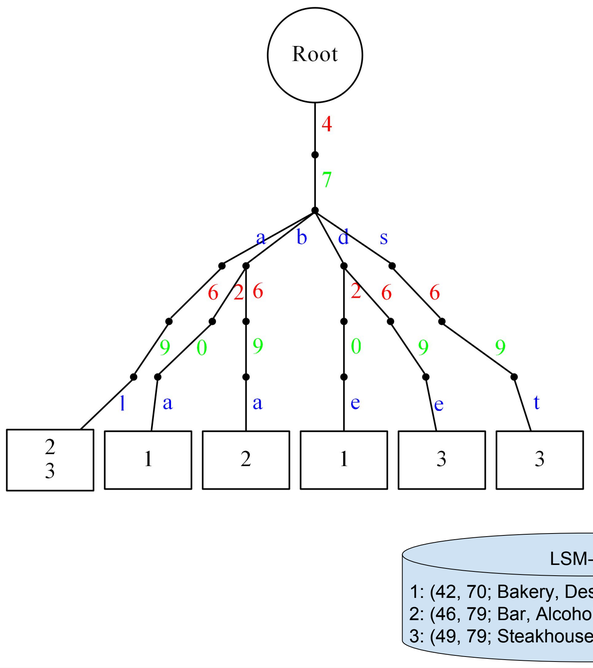
Interpreting the binary digits as left (0) and right (1), the document id is stored at the corresponding leaf

[1] O'Neil, Patrick and Cheng, Edward and Gawlick, Dieter and O'Neil, Elizabeth. 1996. The Log-structured Merge-tree (LSM-tree). Acta Informatica 33, 4 (June 1996).
[2] Dongxiang Zhang, Kian-Lee Tan, and Anthony K. H. Tung. 2013. Scalable Top-k Spatial Keyword Search. In 16th International Conference on Extending Database Technology (EDBT). 359–370
[3] Tuan-Anh Hoang-Vu, Huy T. Vo, and Juliana Freire. 2016. A Unified Index for Spatio-Temporal Keyword Queries. In CIKM (International Conference on Information and Knowledge Management). 135–144.

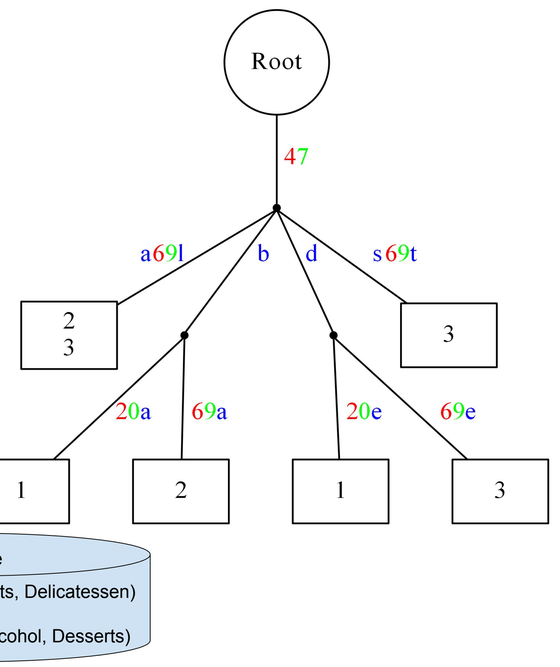
Example

Keys are kept as decimal digits and letters to simplify the visualisation
1: (42, 70; Bakery, Desserts, Delicatessen)
2: (46, 79; Bar, Alcohol)
3: (46, 79; Steakhouse, Alcohol, Desserts)

Uncompressed



Compressed

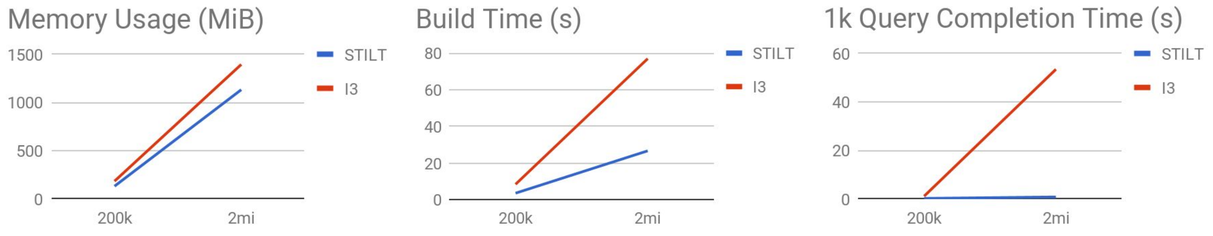


Preliminary Results

Compared to I3 [2]

- Uses ~20% less RAM to build
- Can be built ~4x faster
- Query throughput is ~10x higher (average over 1,000 queries)
- Supports concurrent searching and insertion
  - Concurrent deletion feasible but not implemented
  - I3 does not support dynamic operations (insertion, deletion)

Experimental results for two spatially referenced Twitter datasets



Sponsored by: UNB Faculty of COMPUTER SCIENCE · NSERC CRSNG