Succinct and SuccinctStore

Fast Interactive Queries via "Homomorphic" Compression

Rachit Agarwal, Anurag Khandelwal, Ion Stoica

Succinct in a Nutshell

A data store that supports:

- Queries on primary and secondary attributes
- Input data stored in a compressed form
- Query responses on secondary attributes embedded within the compressed representation
- Read-optimized, append-only updates
- Low memory
  (input in compressed form; no secondary indexes)
- Low Latency (no scans, no decompression)

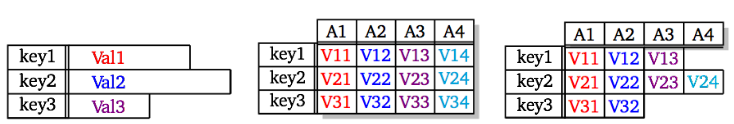
Succinct

- Supports queries on flat files
- Easily extensible to support queries on semi-structured data (key-value pairs, tables, etc.)
- Three basic queries:
  - `extract(offset, len)`
  - `count(string str)`
  - `search(string str)`
- Extended to support `range` and `wildcard` queries

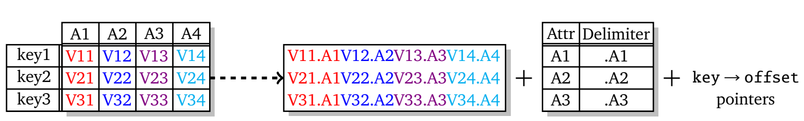
SuccinctStore

Succinct supports queries on flat files. It can be easily extended to support queries on semi-structured data (key-value pairs, tables, etc.)

- Logical collection of records



- Transformation to flat file:

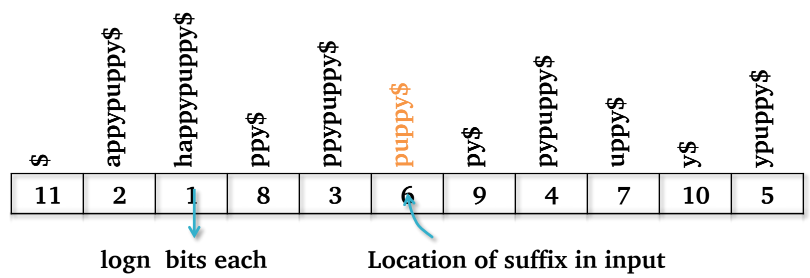


- Query translation:
  - `get(key)` → `extract(offset, delim)`
  - `count(A2, val)` → `count(val.A2)`
  - `search(A2, val)` → `search(val.A2)`

Technique Overview

A simplified problem:
Searching on a compressed representation of input data
happypuppy$

Step 1: Extract suffixes from input data.
Step 2: Sort suffixes lexicographically.
Step 3: Obtain locations of suffixes in the input data:

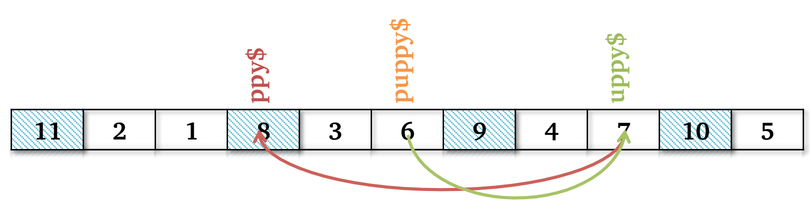


Problem: Total space to store the locations is nlogn.

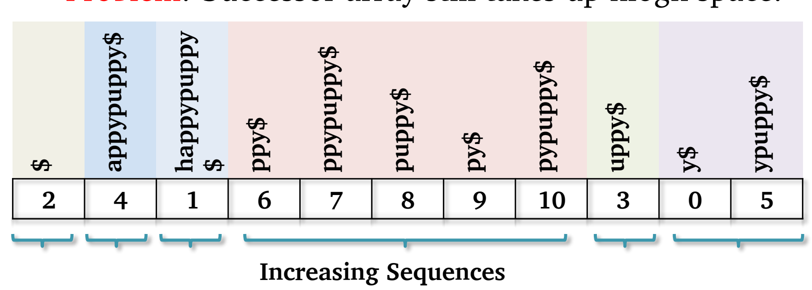
Step 4: Store sampled values only.

Problem: How to lookup unsampled values?

Step 5: Maintain pointers to index of successor.



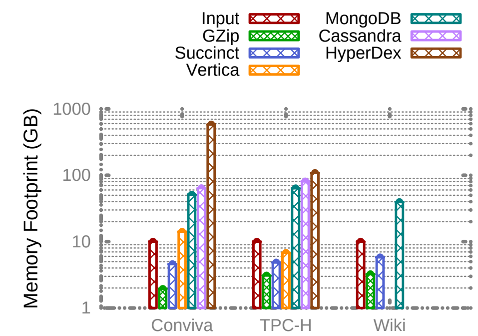
Problem: Successor array still takes up nlogn space.



Step 6: Compress the increasing sequences.

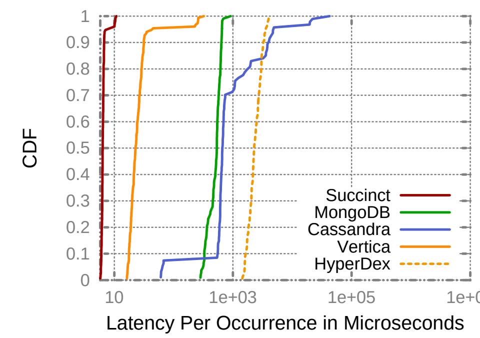
- More complex operations to convert these values into contiguous sequences, which are more compressible.

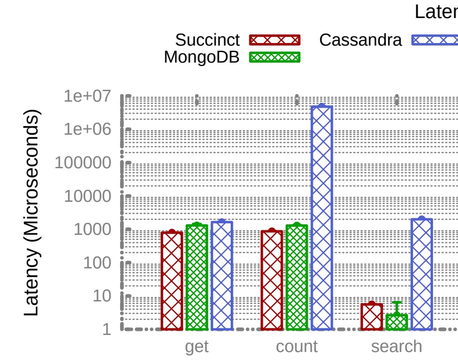
Preliminary Results



Succinct requires 16 − 126× lower memory than systems with similar functionality

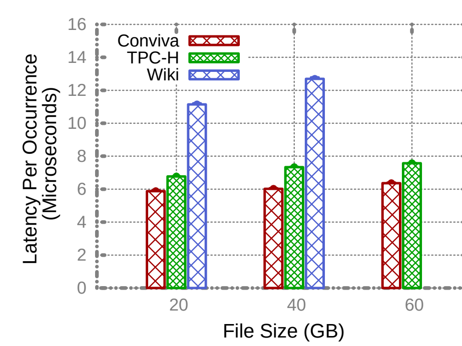
When datasets do not fit in memory, Succinct achieves 20 − 400× lower latency.





When datasets fit in memory, Succinct achieves same performance using 16× lower memory.

Succinct latency scales very well with file size.



Related Work

- **Traditional Compression:** GZip, LZ, Snappy, etc.
- **Sampling Data:** BlinkDB, for *approximate computations* in data analytics
- **Column-Stores:** Use of *opportunistic compression*
- **Full-Text Indexes:** Compressed Suffix Arrays, FM-Index, etc.

{anuragk, ragarwal, istoica}@eecs.berkeley.edu