Efficient RegEx Queries on Compressed Data

Anurag Khandelwal, Rachit Agarwal, Ion Stoica

Regular Expressions

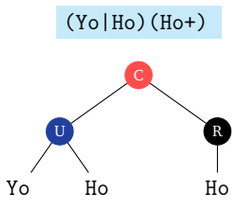
- **Example:** `1-\d\d\d-\d\d\d-\d\d\d\d` (US Phones)
- **Wide range of applications**: text and document stores, bioinformatics, data mining, etc.

Existing Techniques

- Full Data Scans (NFA, DFA)
  - Do not scale with data size
  - E.g., MongoDB, Oracle, MySQL, etc
- m-gram Indexes with Partial Data Scans
  - Index token of length m for multiple or all values of m
  - Avoid scans for indexed tokens.
  - Partial scans for tokens not indexed.
  - Suffer from large memory footprint
  - E.g., ElasticSearch
- Succinct (Search on compressed data)
  - Memory-efficient search for arbitrary length tokens
  - Asymptotic search complexity similar to m-gram indexes
  - Can be used as a *black box*

Black-box RegEx

- Represent RegEx as an `RTree`
  - Leaves are tokens; Interior nodes are operators
- Search for leaves
- Traverse the `RTree` bottom-up, combining intermediate results at each node



Operator	Complexity	Optimal?
`Union`	$O(s)$	Yes
`Repeat`	$O(s)$	Yes
`Wildcard`	$O(s \log n)$	Yes
`Concat`	$O(n+m)$	No

n, m: input sizes (intermediate results)
s: output size

Output cardinality for `Concat` operator can be arbitrarily smaller than cardinality of intermediate results

- Unnecessary operations if the gap is large

Complexity of `Union`, `Repeat`, and `Wildcard` operators depend on output cardinality

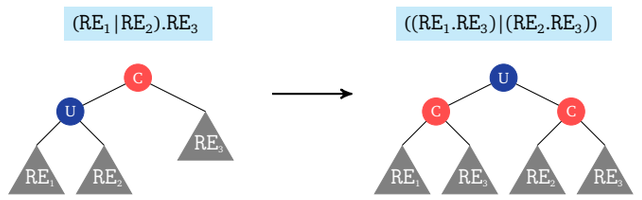
- Output cardinality smaller up the RTree

Succinct

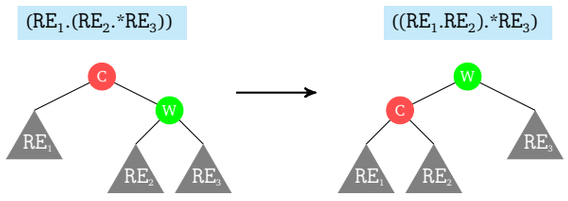
Main Idea: Transform the `RTree` for the RegEx such that:

- the black-box approach can be avoided for the `Concat` operator altogether
- `Union`, `Repeat` and `Wildcard` operators are pushed up the `RTree`

`Pull-Up Union` Transformation:



`Pull-Up Wildcard` Transformation:



`Pull-Out Repeat` Transformation:

Replace `Repeat` operator by `Unions` of `Concats`:

$$\texttt{RE+} = (\texttt{RE}^1|\texttt{RE}^2|\texttt{RE}^3|...|\texttt{RE}^k)$$

k: smallest integer for which $\texttt{RE}^{k+1}$ has 0 occurrences.

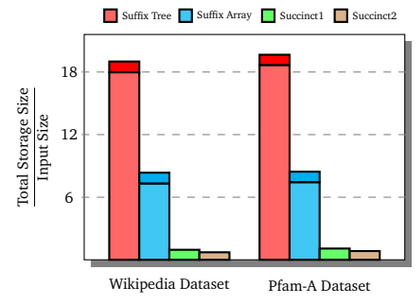
- Use heuristic to upper bound value of k
- k can be large when RegEx contains character classes
- Use partial scans beyond threshold

`Pull-Out Concat` Transformation:

- Find `Concat` nodes whose children are tokens ($\texttt{T}_1$, $\texttt{T}_2$)
- Replace with new token $\texttt{T}_1\texttt{T}_2$

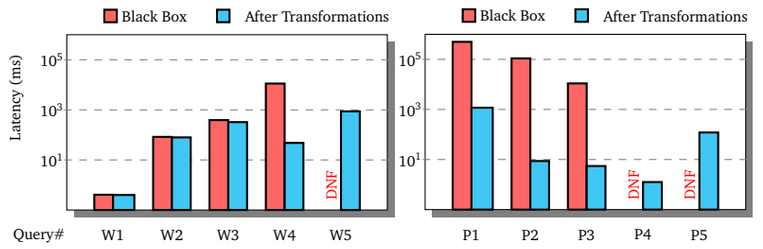
Transformations incorporated within Succinct data structures.

Results

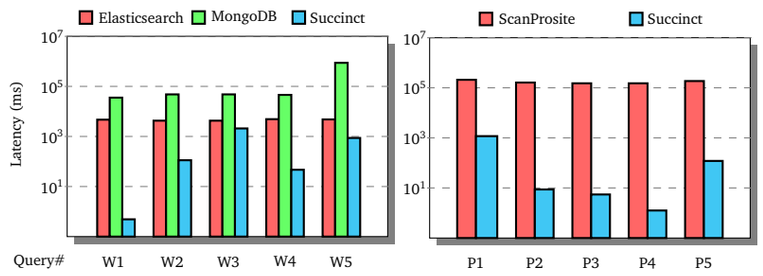


Succinct has 8× smaller storage footprint than uncompressed data structures

	Query ID	Query	Description
Wikipedia	Query#W1	<script>.*</script>	HTML Scripts
	Query#W2	Motorola.*(XPC|MPC)([0-9])+([0-9a-z])*	Motorola PowerPC Chip#
	Query#W3	William [A-Z]([a-z])+ Clinton	Bill Clinton's middle name
	Query#W4	1-\d\d\d-\d\d\d-\d\d\d\d	US Phone Numbers
	Query#W5	([a-z0-9_\.])+(([a-z0-9])+\.)*stanford\.edu	Stanford domain URLs.
Pfam-A	Query#P1	[DE]GSW.[GE].W[GA][LIVM].[FY].Y[GA]	TERPENE_SYNTHASES
	Query#P2	[AC]GL.FPV	HISTONE_H2A
	Query#P3	CKPCLK.TC	CLUSTERIN_1
	Query#P4	G[MV]ALFCGCGH	MYELIN_PLP_1
	Query#P5	[FYW]P[GS]N[LIVM]R[EQ]L.[NHAT]	SIGMA54_INTERACT_3



Black-box, with and without transformations



Succinct versus {ElasticSearch, MongoDB, ScanProsite}

Open Source Release

- `https://github.com/amplab/succinct-cpp`
- For questions & feedback, contact us at:

{anuragk, ragarwal, istoica}@berkeley.edu

Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · Princeton University · UC Berkeley · University of Washington