

Massive Text Indices

Final Meeting SPP Big Data







Florian Kurpicz

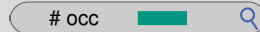
The slides are licensed under a Creative Commons Attribution-ShareAlike 4.0 International License © ⓘ ⓘ: www.creativecommons.org/licenses/by-sa/4.0 | commit 52f896c compiled at 2022-06-30-14:10

Motivation

```

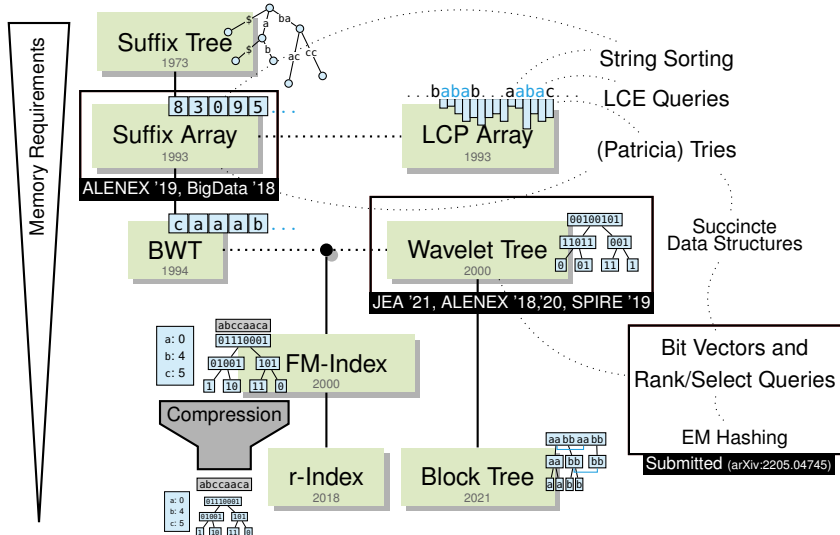
GAATGCCAGTCAGCATTAAAGGCCAGGC
GGAGAGCTCAGGGCAGGTCACGTGGGA
AACTCGCATAGTGAGGGTTATCGCTCG
ACATGTTTCGTTGGGCTCTTCACTTCTT
CCGACACGAACCTCAGTTAGTTTGTTA
CCTACATCCTACCAGAGGTCGCACCTA
TGTGCCCCGGTGGTGAGAAGGAGAAGG
TGCGGATTTCGTATTTGCAGATGCGGA
CTCGTCAGTACTTTCAGAATAACGAAT
CATGGCCTGCACGGCAAATGGCAATG
GACGCTTATAATGGACTTCGACATTTCG
AACTCGCATAGTGAGGGTTATCGGGTT
ACATGTTTCGTTGGGCTCTTCACTCTTC
CCGACACGAACCTCAGTTAGTTTAGTT
TGTGCCCCGGTGGTGAGAAGGAGAAGG
CCTACATCCTACCAGAGGTCGCAGGTC
CATGGCCTGCACGGCAAATGGCAAAT
  
```

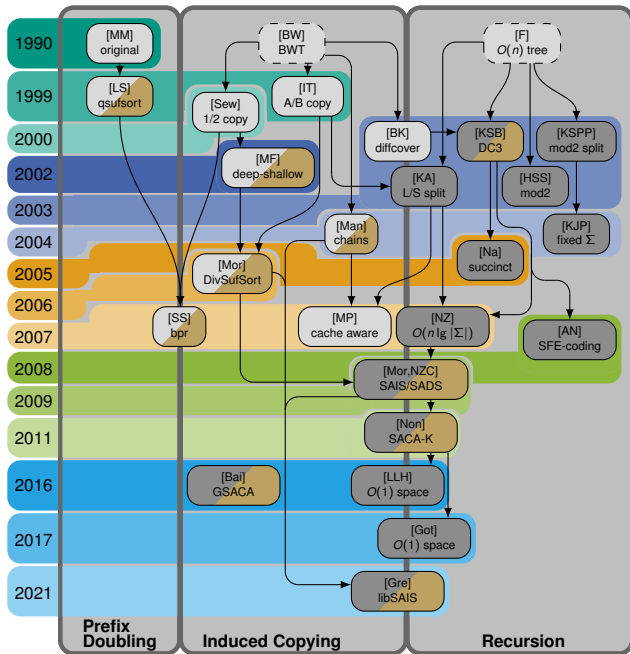
word	page
	#1, #3, #7, ...
	#2, #3, ...
	#2, #4, #5, ...
	#1, #2 ⁴⁰ , ...
	#3, #7, #9, ...
	#4, #5, #6, ...
	⋮



- texts are everywhere
- collection of texts
- scanning not feasible
- inverted index (word based)
- phrase search
- counting queries
- what if there are no “words”

Thank You!

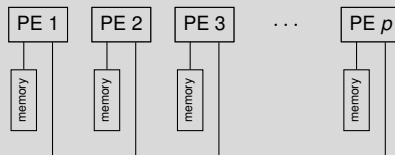




Timeline Sequential Suffix Sorting

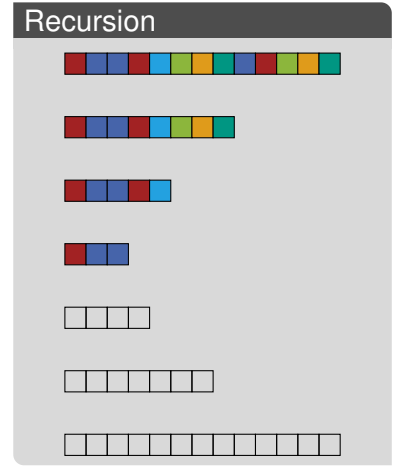
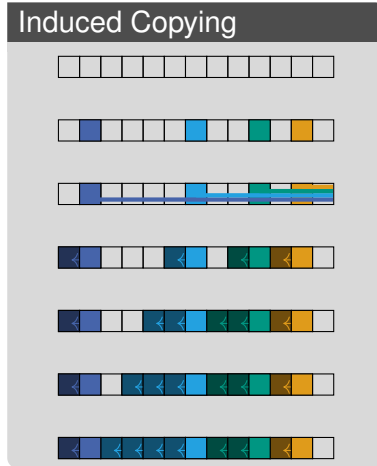
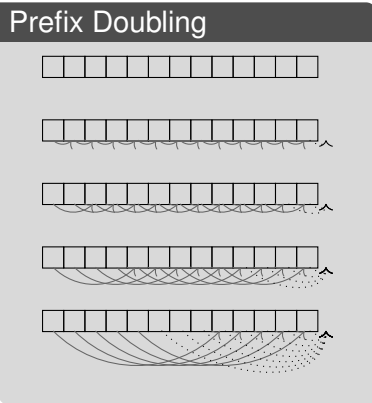
- darker grey: linear running time
- brown: available implementation

Distributed Memory

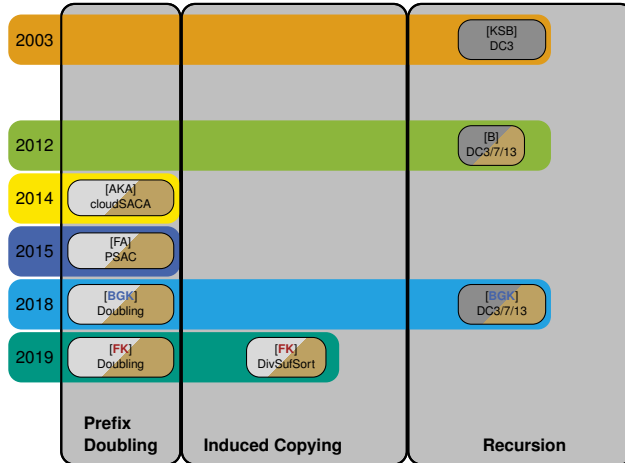


- not whole text available on every PE

Suffix Sorting Techniques

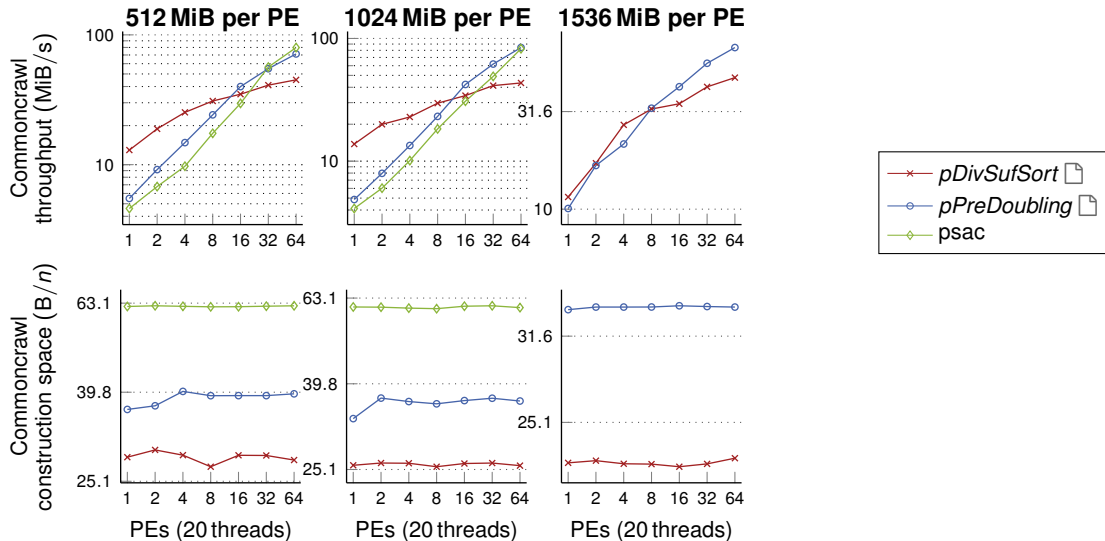


Distributed Memory Suffix Sorting

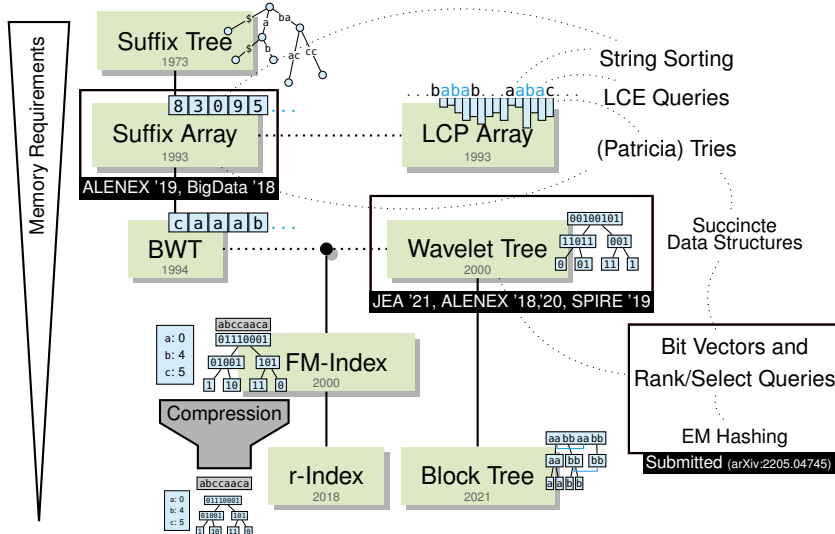


- suffix sorting using Thrill
- suffix sorting using MPI

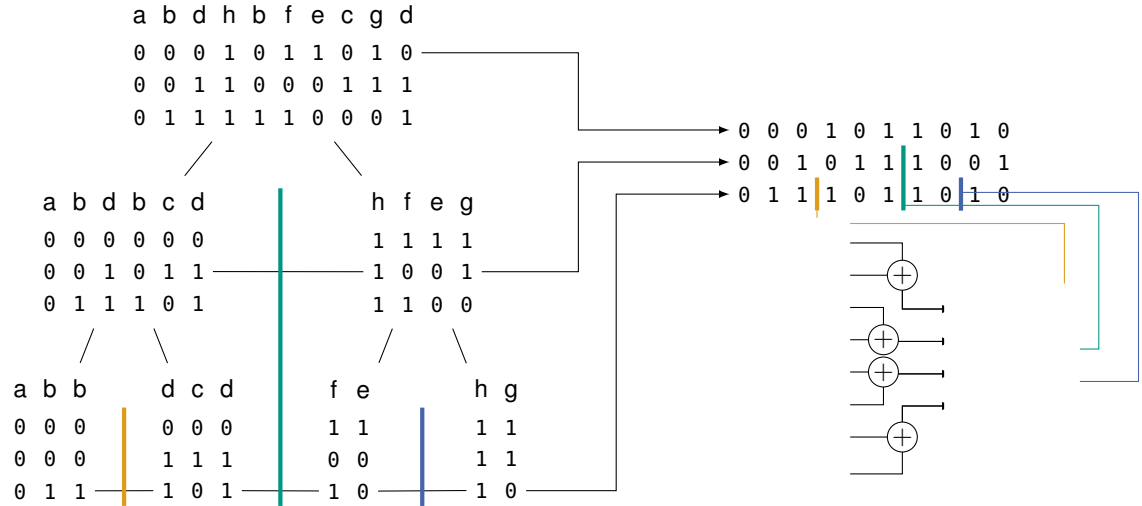
Distributed Memory Suffix Sorting: Experiments (MPI)



Thank You!



The Wavelet Tree Bottom-Up



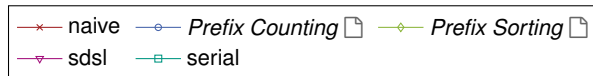
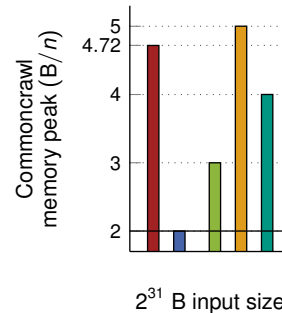
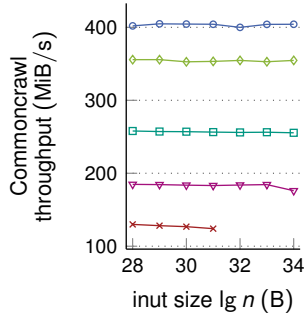
Sequential Wavelet Tree Construction

Prefix Counting

- insert bits based on borders

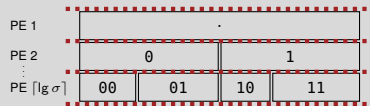
Prefix Sorting

- sort text based on borders
- insert bits from left to right



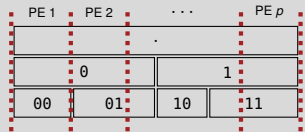
Parallel Wavelet Tree Construction

Parallel Prefix Counting

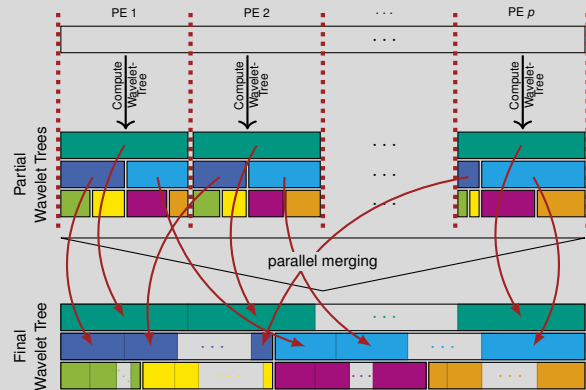


■ at most $\lceil \lg \sigma \rceil$ PEs

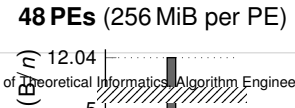
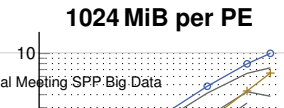
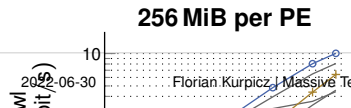
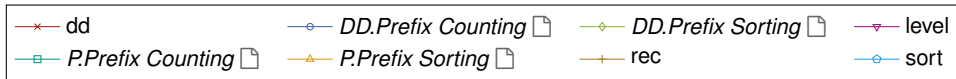
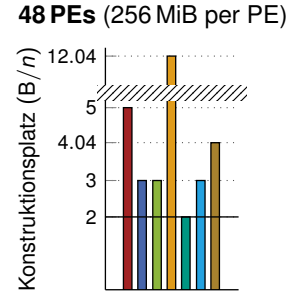
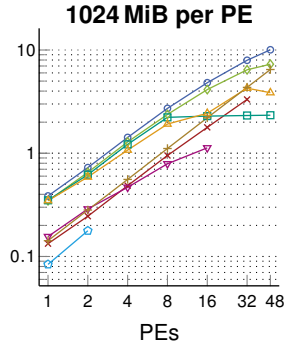
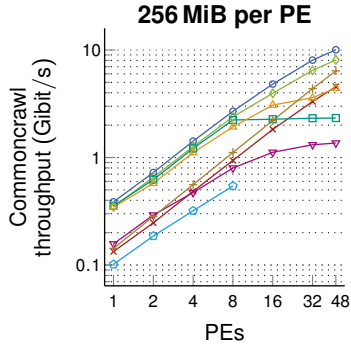
Parallel Prefix Sorting



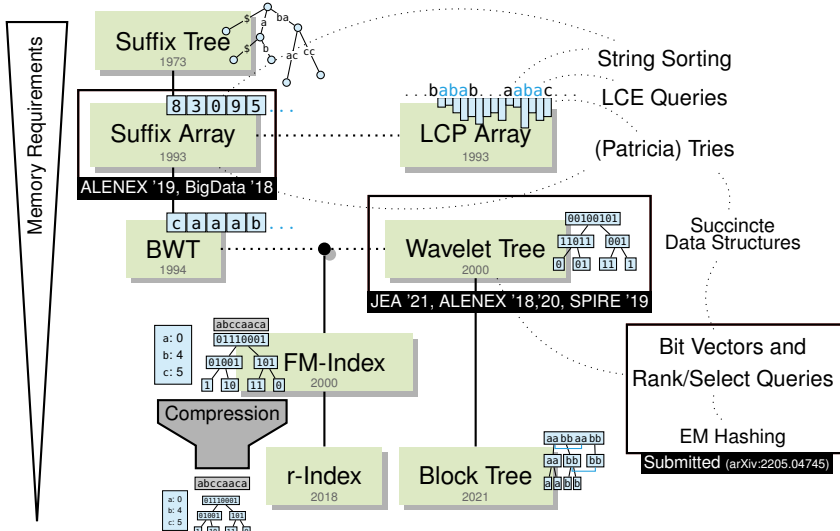
Domain Decomposition



Parallel Wavelet Tree Construction: Experimental Evaluation



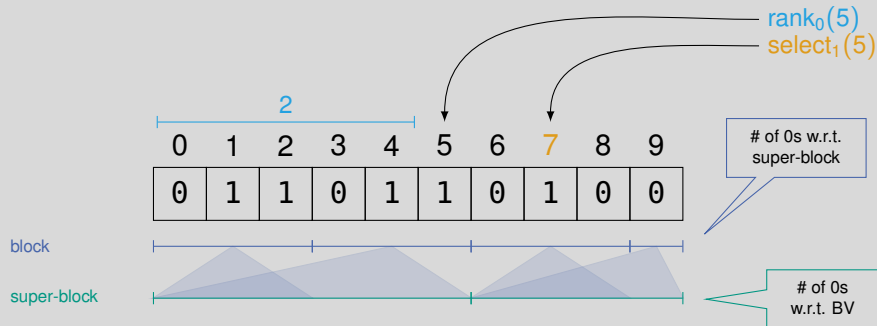
Thank You!



Rank Queries in Bit Vectors

$\text{rank}_\alpha(i)$ # of α s before i

$\text{select}_\alpha(j)$ position of j -th α

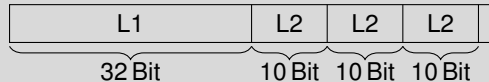


Bit Vectors with Rank and Select Support

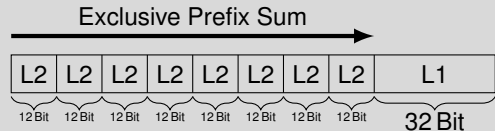
cs-poppy

- Blocks (L2): 512 bits
- Super-Blocks (L1): 2048 bits
- Super-Super-Blocks (L0): 2^{31} bits
- 3.51 % space overhead

L1+L2 Together in 64 Bits



New: L1+L2 Together in 128 Bits



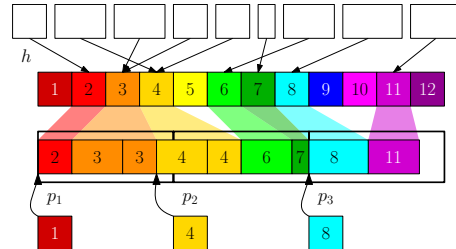
SIMD Access

- 3 consecutive bytes contain two 12 bit entries
- shuffle and shift entries in seven 16 bit blocks
- search all entries at the same time
- 8%–16 % faster than cs-poppy

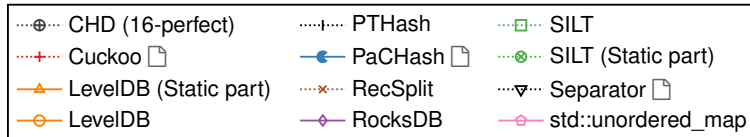
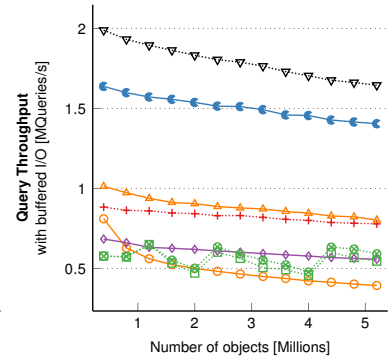
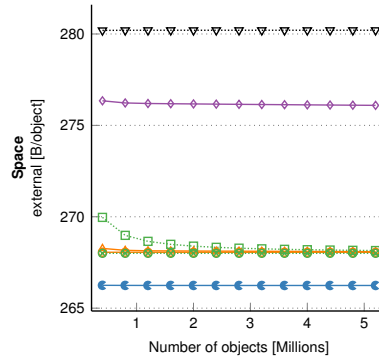
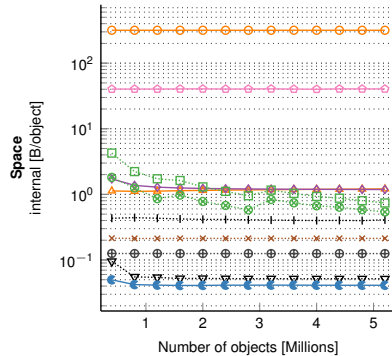
EM Hashing: PaCHash

General Idea

- small internal memory index
- Elias-Fano representation
- $2 + \log a + o(1)$ bits per EM block
- $1.4427 + \log(a + 1) + o(1)$ theoretically possible
- variable size objects contiguously in EM
- retrieving x loads $\leq 1 + |x|/B + 1/a$ blocks in expectation



PaCHash Experiments (Fixed Size Objects)



Thank You!

