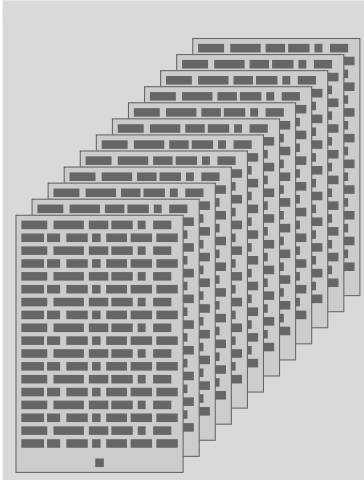
Karlsruhe Institute of Technology

Massive Text Indices

Final Meeting SPP Big Data

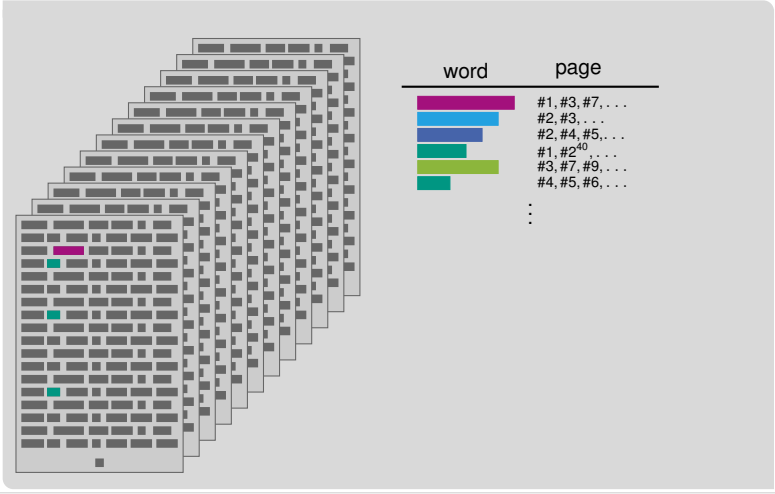
Florian Kurpicz

Motivation



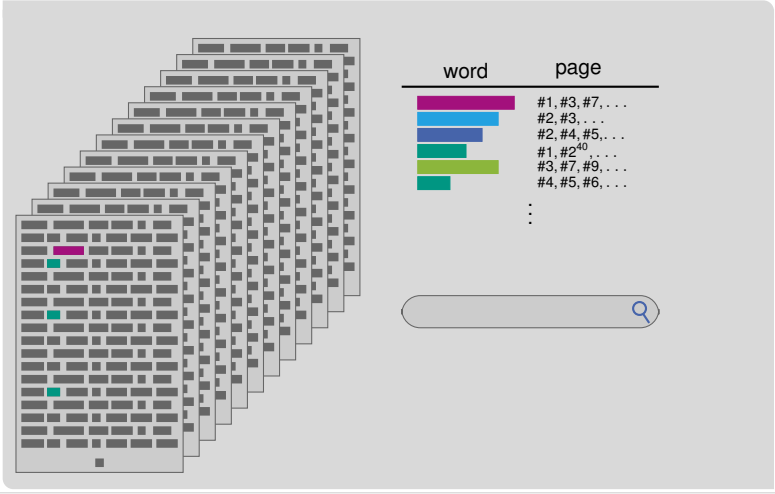
- texts are everywhere
- collection of texts
- scanning not feasible

Motivation



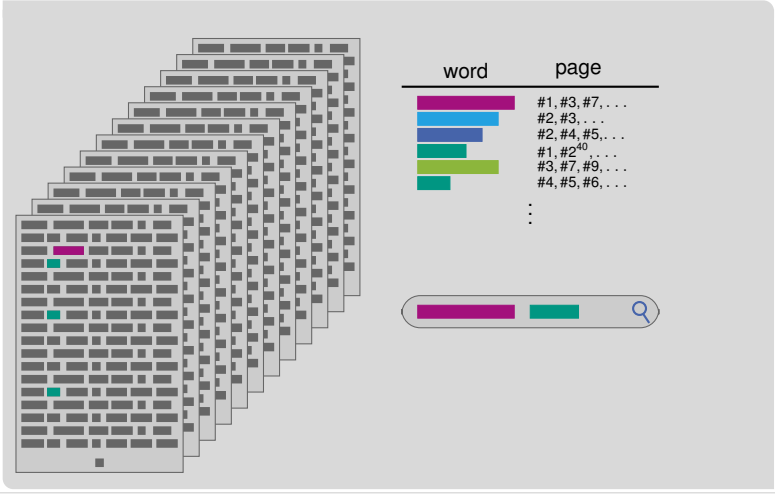
- texts are everywhere
- collection of texts
- scanning not feasible
- inverted index (word based)

Motivation



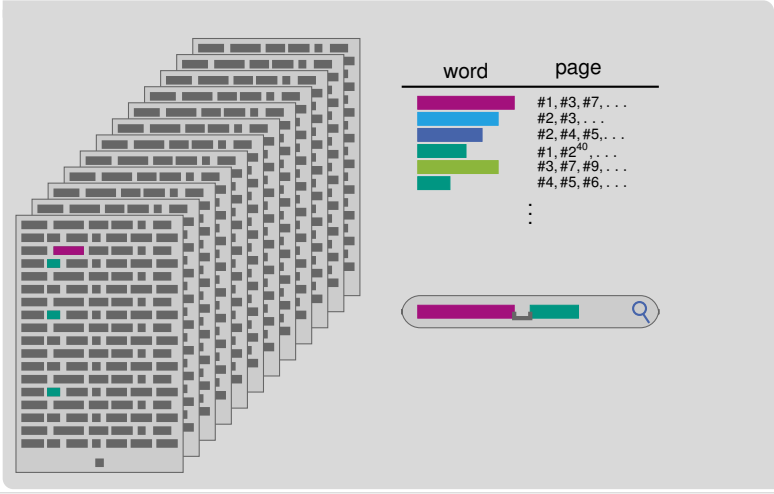
- texts are everywhere
- collection of texts
- scanning not feasible
- inverted index (word based)

Motivation



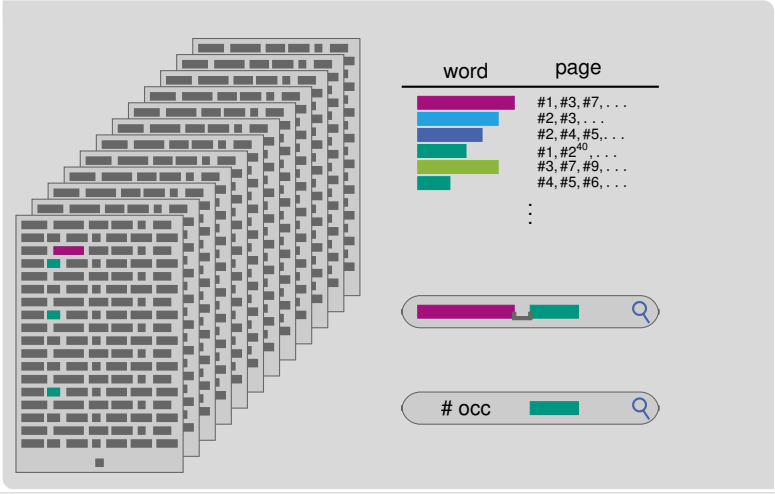
- texts are everywhere
- collection of texts
- scanning not feasible
- inverted index (word based)

Motivation



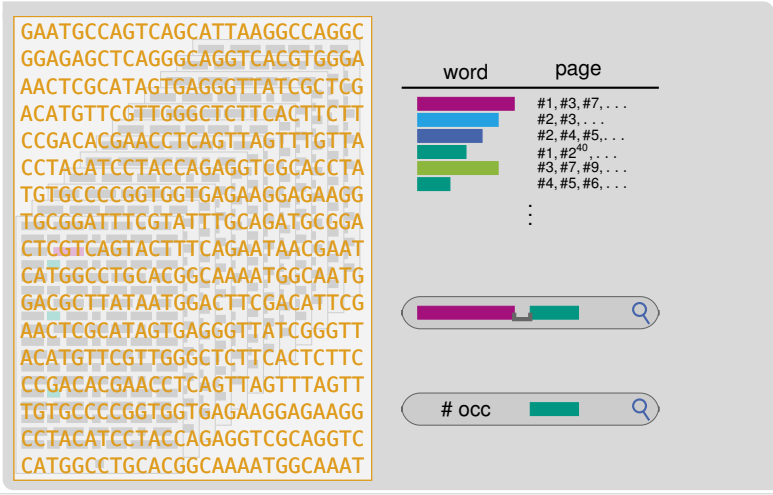
- texts are everywhere
- collection of texts
- scanning not feasible
- inverted index (word based)
- phrase search

Motivation



- texts are everywhere
- collection of texts
- scanning not feasible
- inverted index (word based)
- phrase search
- counting queries

Motivation

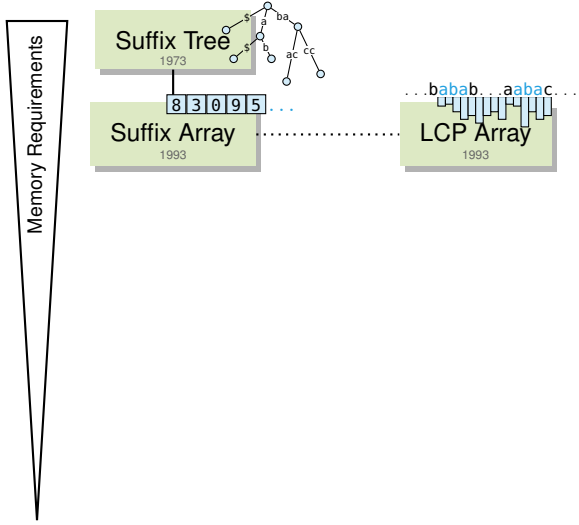


- texts are everywhere
- collection of texts
- scanning not feasible
- inverted index (word based)
- phrase search
- counting queries
- what if there are no “words”

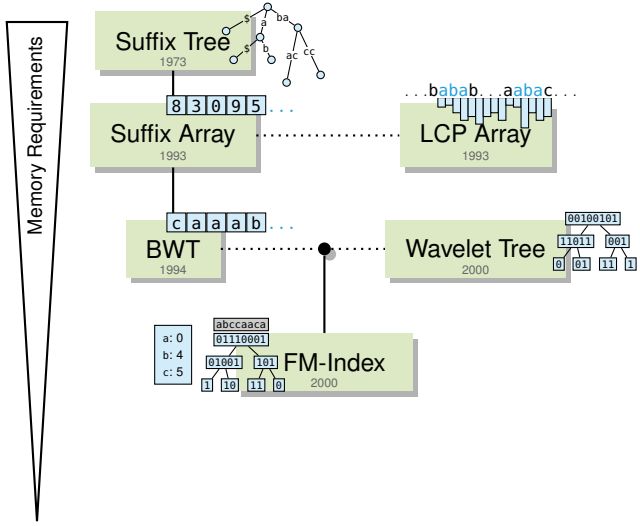
Overview: Text Indices and Main Results

Suffix Tree
1973

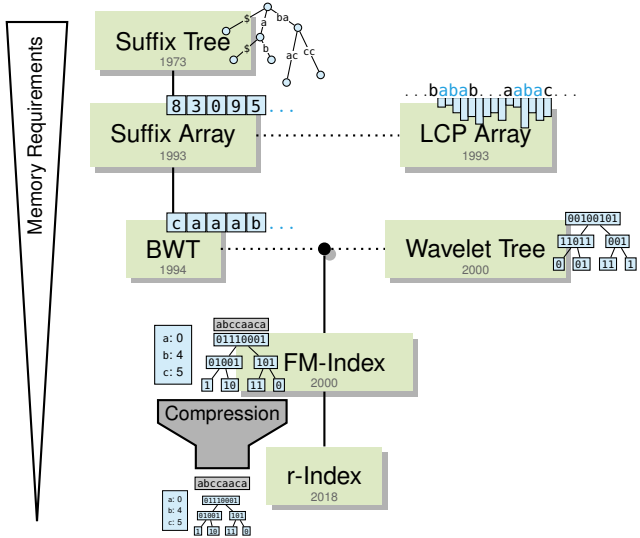
Overview: Text Indices and Main Results



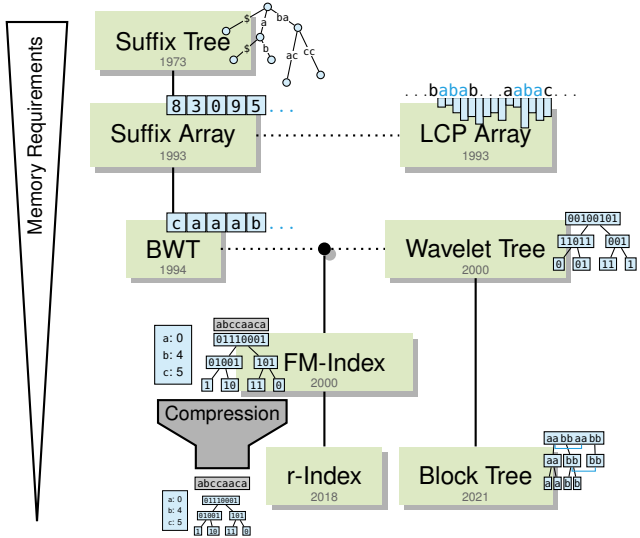
Overview: Text Indices and Main Results



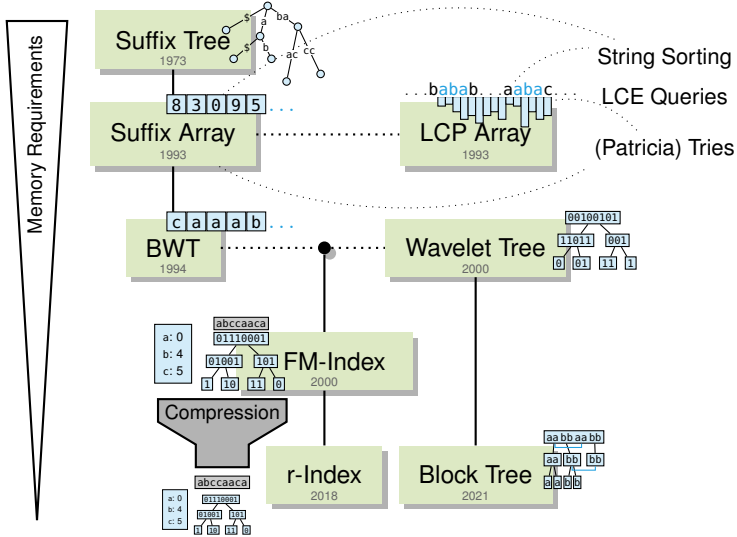
Overview: Text Indices and Main Results



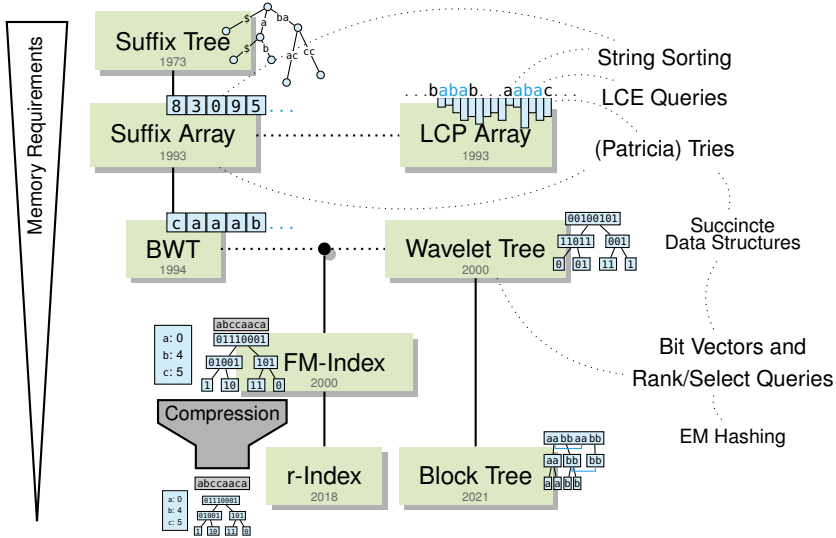
Overview: Text Indices and Main Results



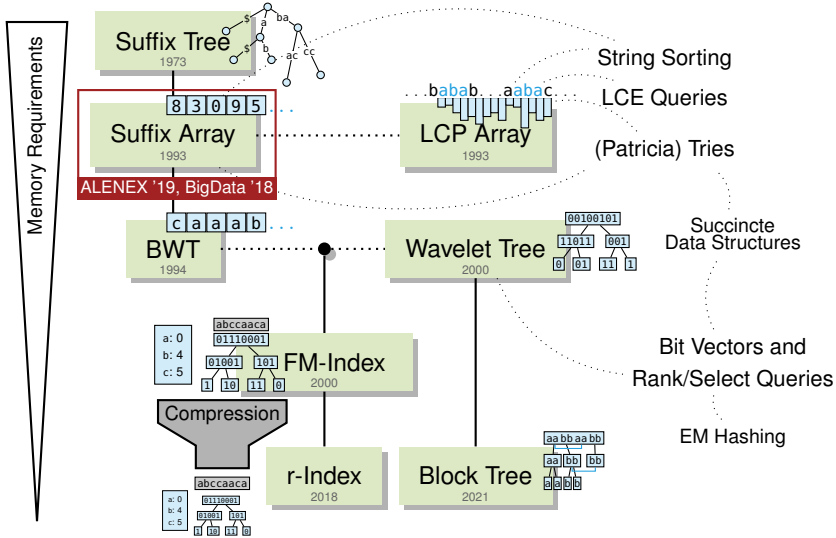
Overview: Text Indices and Main Results



Overview: Text Indices and Main Results

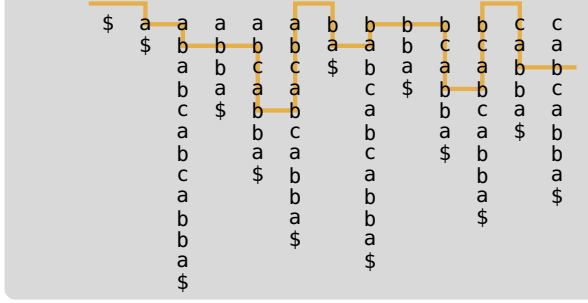


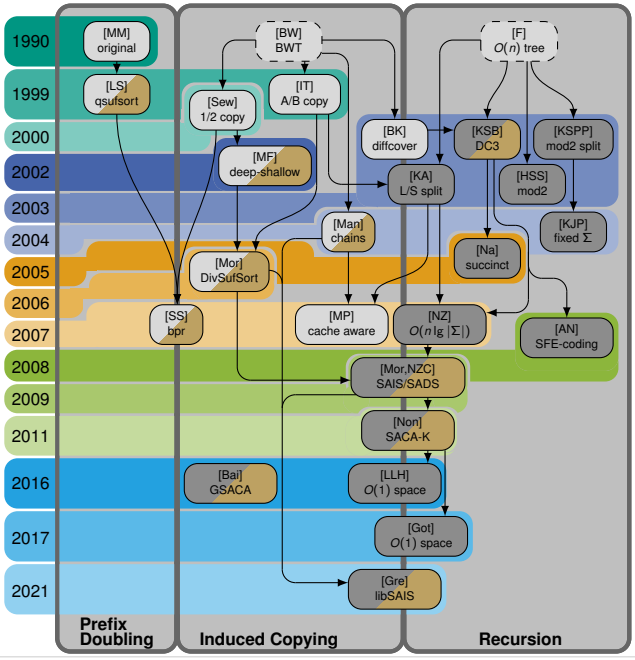
Overview: Text Indices and Main Results



Suffix Array and Longest Common Prefix Array

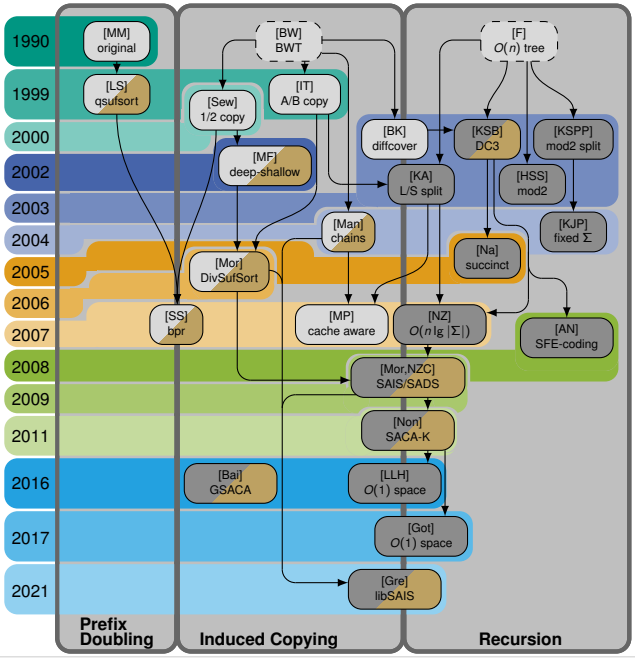
	1	2	3	4	5	6	7	8	9	10	11	12	13
T	a	b	a	b	c	a	b	c	a	b	b	a	$
SA	13	12	1	9	6	3	11	2	10	7	4	8	5
LCP	0	0	1	2	2	5	0	2	1	1	4	0	3





Timeline Sequential Suffix Sorting

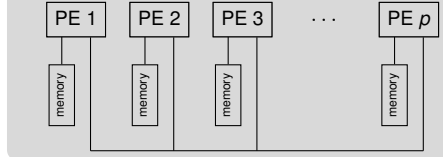
- **darker grey**: linear running time
- **brown**: available implementation



Timeline Sequential Suffix Sorting

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

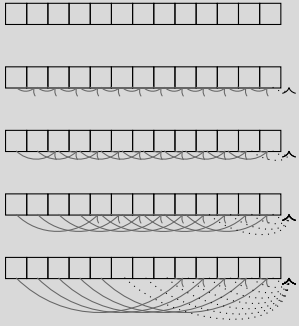
Distributed Memory



- not whole text available on every PE

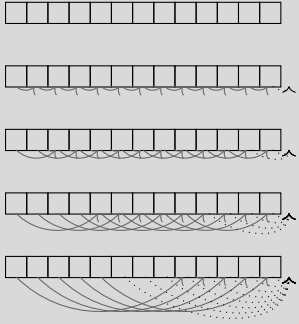
Suffix Sorting Techniques

Prefix Doubling

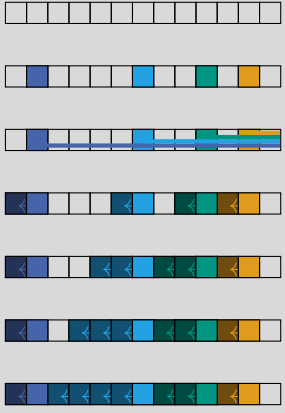


Suffix Sorting Techniques

Prefix Doubling

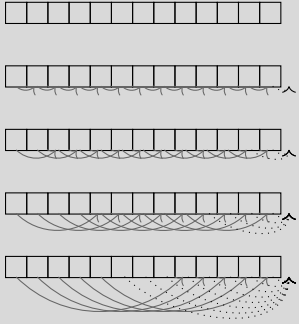


Induced Copying

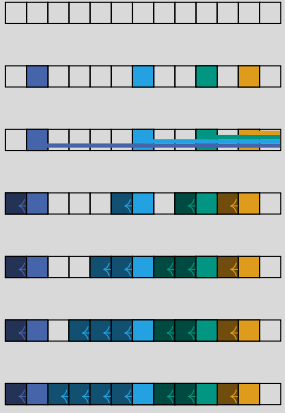


Suffix Sorting Techniques

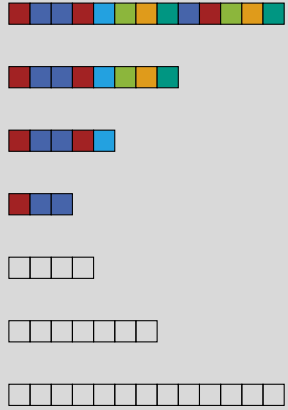
Prefix Doubling



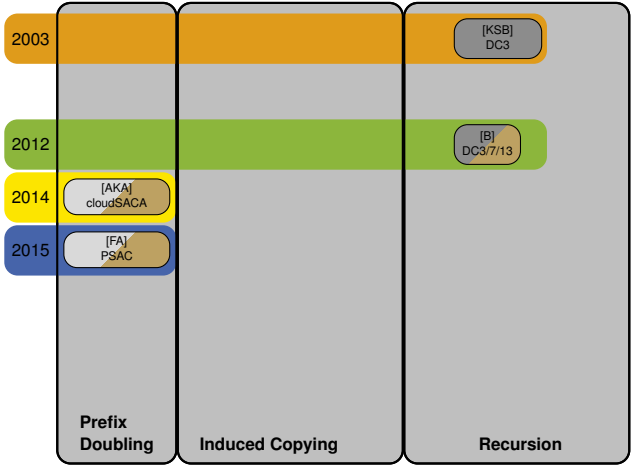
Induced Copying



Recursion

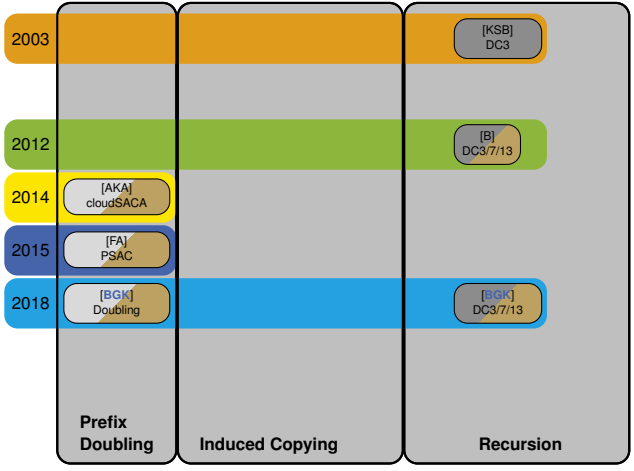


Distributed Memory Suffix Sorting



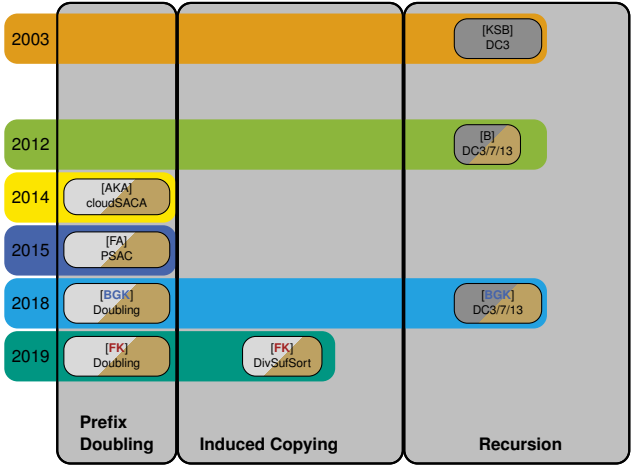
- suffix sorting using Thrill
- suffix sorting using MPI

Distributed Memory Suffix Sorting



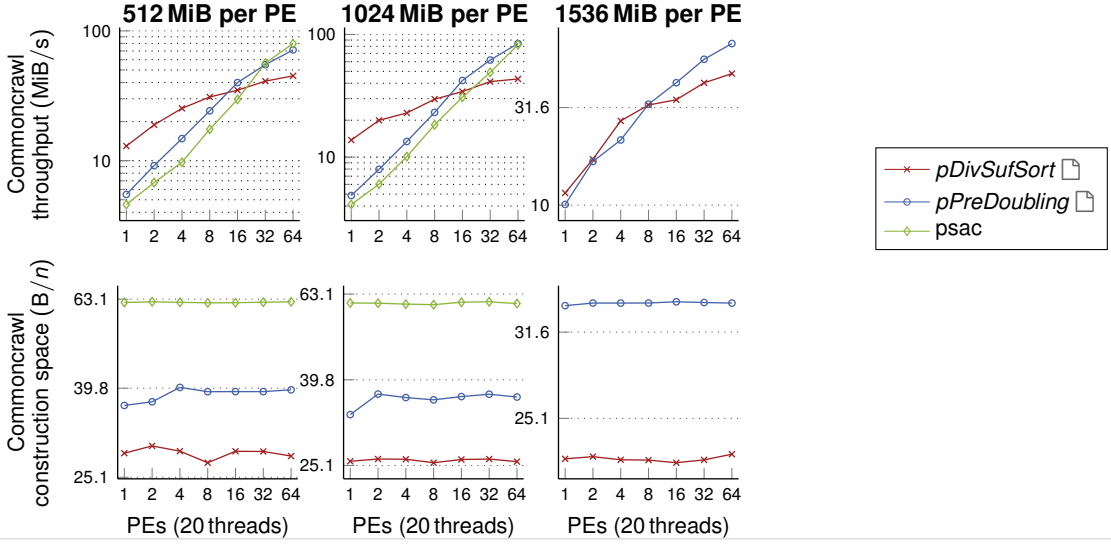
- suffix sorting using Thrill
- suffix sorting using MPI

Distributed Memory Suffix Sorting

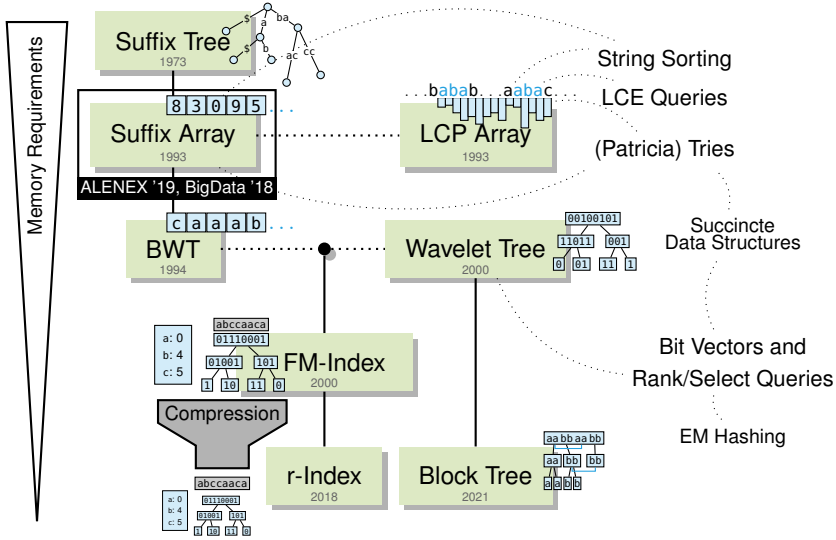


- suffix sorting using Thrill
- suffix sorting using MPI

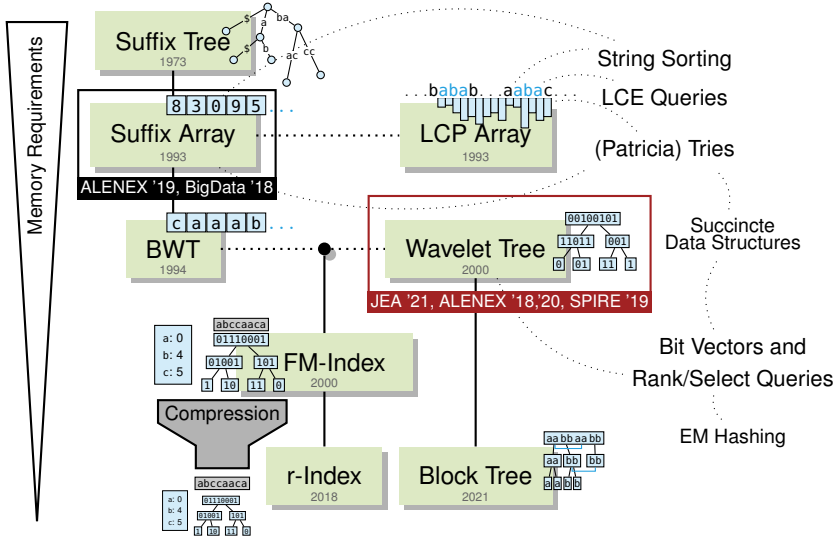
Distributed Memory Suffix Sorting: Experiments (MPI)



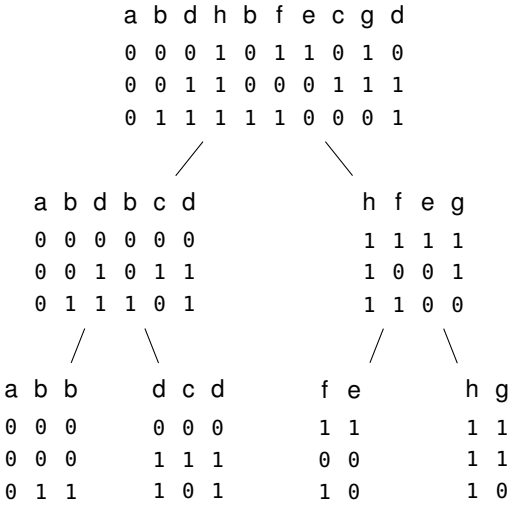
Overview: Text Indices and Main Results



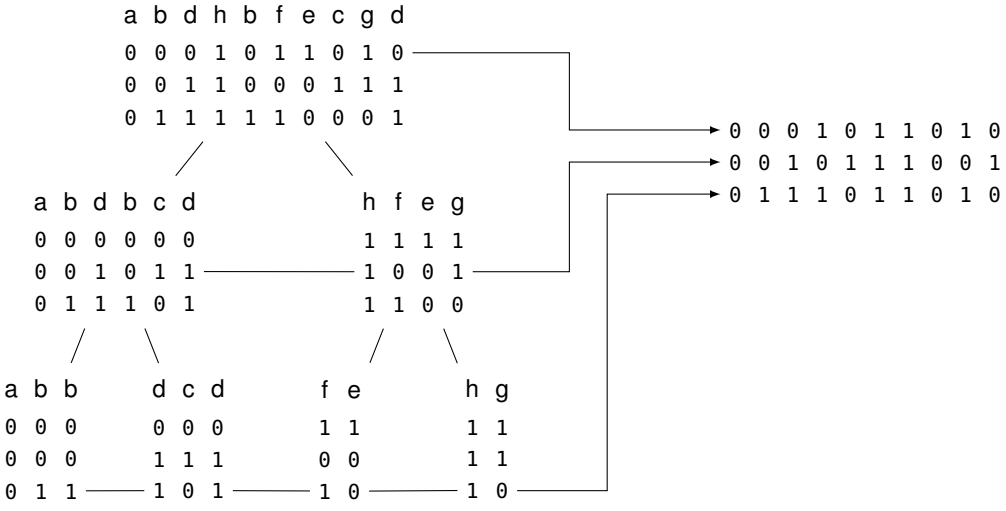
Overview: Text Indices and Main Results



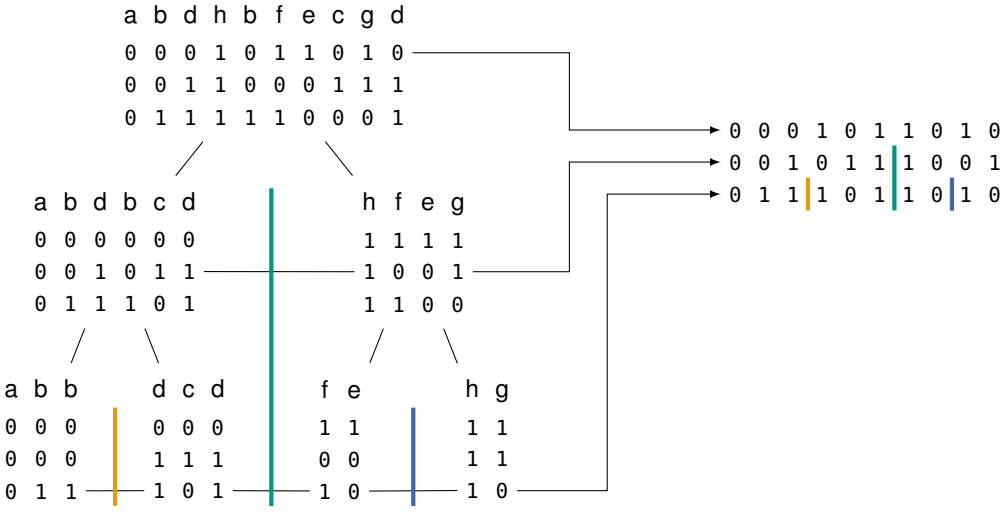
The Wavelet Tree Bottom-Up



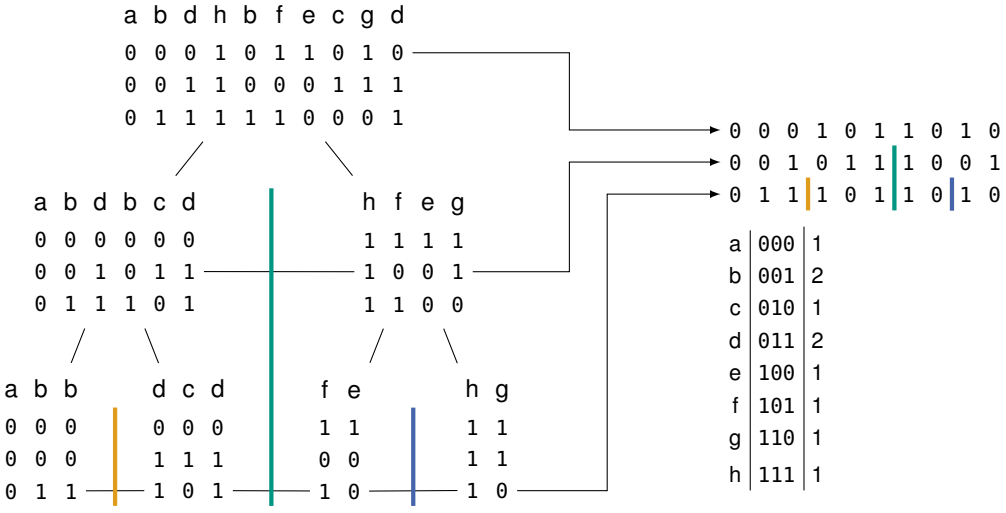
The Wavelet Tree Bottom-Up



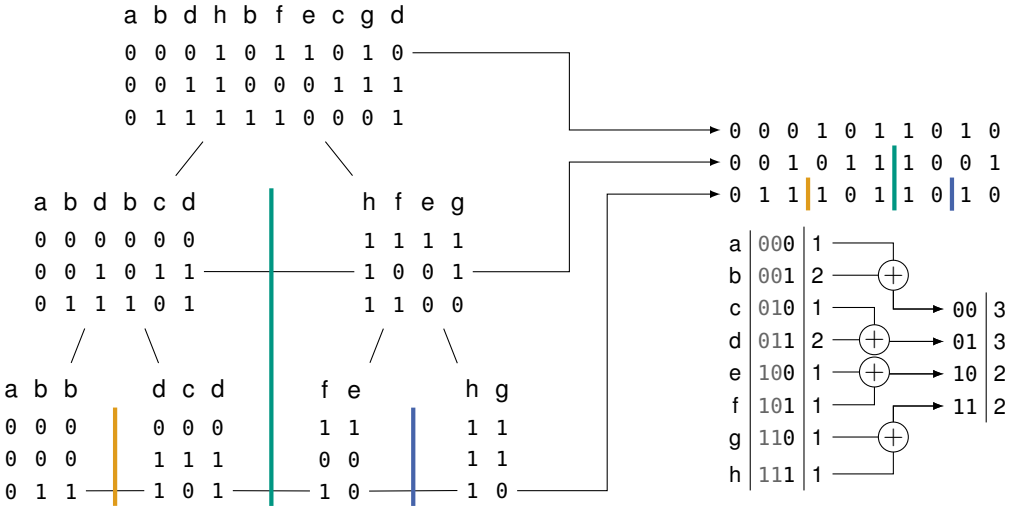
The Wavelet Tree Bottom-Up



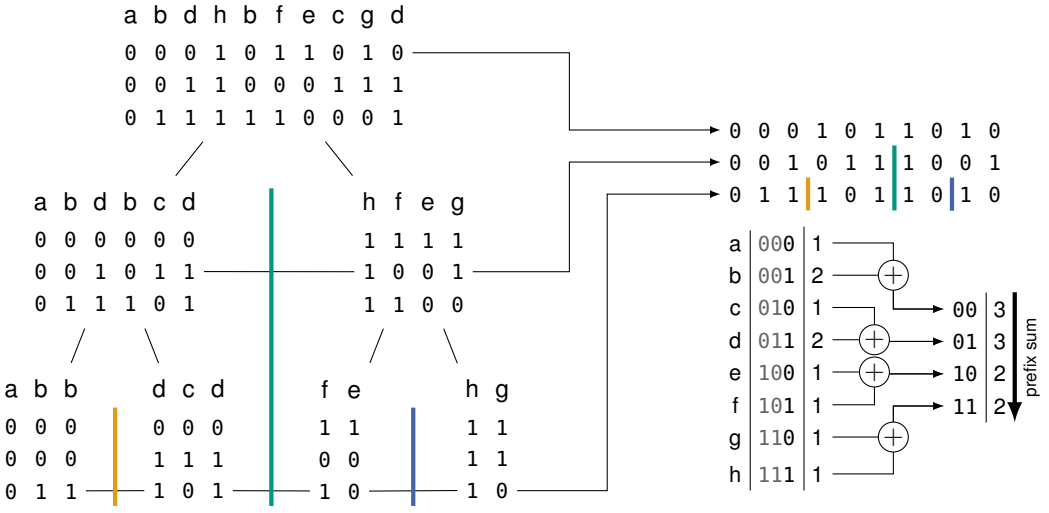
The Wavelet Tree Bottom-Up



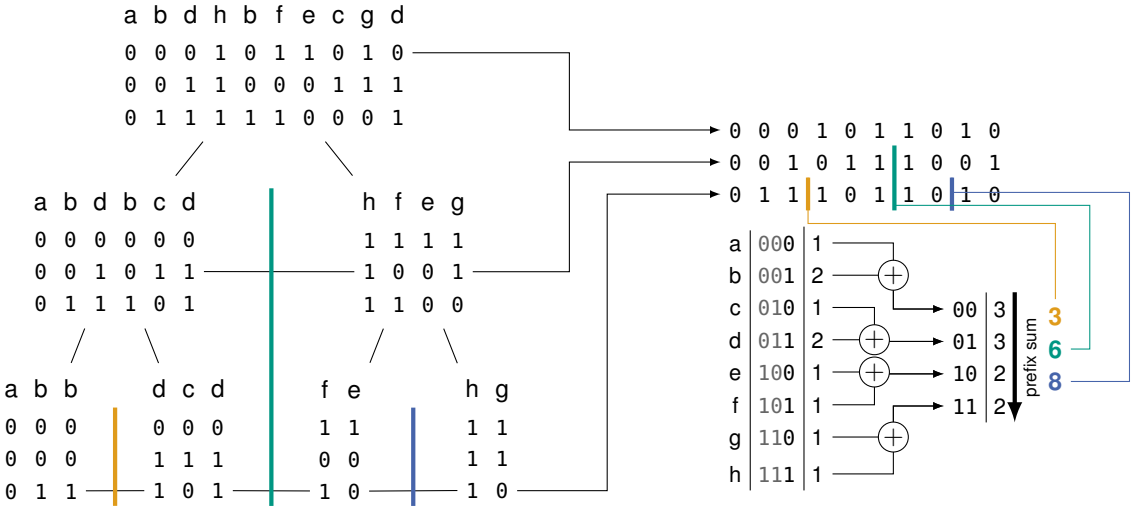
The Wavelet Tree Bottom-Up



The Wavelet Tree Bottom-Up



The Wavelet Tree Bottom-Up



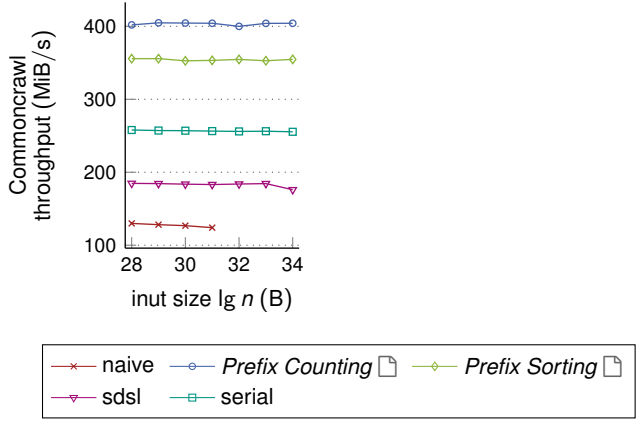
Sequential Wavelet Tree Construction

Prefix Counting

- insert bits based on borders

Prefix Sorting

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



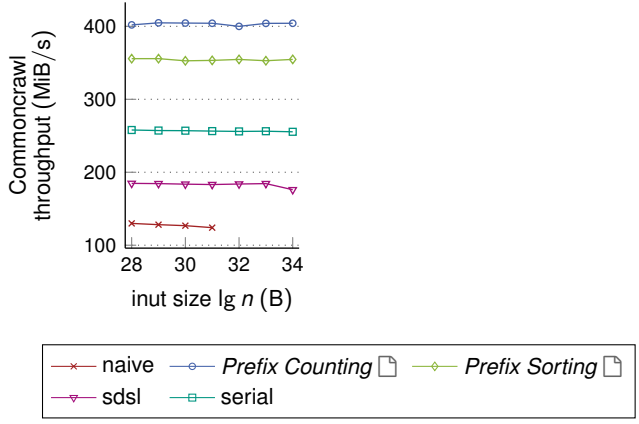
Sequential Wavelet Tree Construction

Prefix Counting

- insert bits based on borders

Prefix Sorting

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



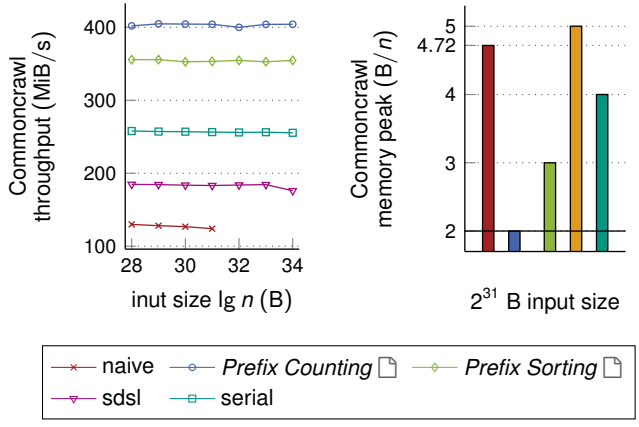
Sequential Wavelet Tree Construction

Prefix Counting

- insert bits based on borders

Prefix Sorting

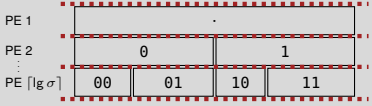
- sort text based on borders
- insert bits from left to right



Parallel Wavelet Tree Construction

Parallel Wavelet Tree Construction

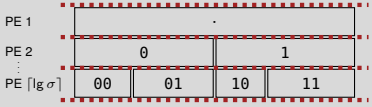
Parallel Prefix Counting



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

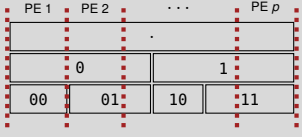
Parallel Wavelet Tree Construction

Parallel Prefix Counting



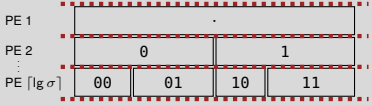
- at most $\lceil \lg \sigma \rceil$ PEs

Parallel Prefix Sorting



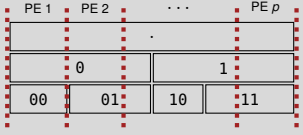
Parallel Wavelet Tree Construction

Parallel Prefix Counting

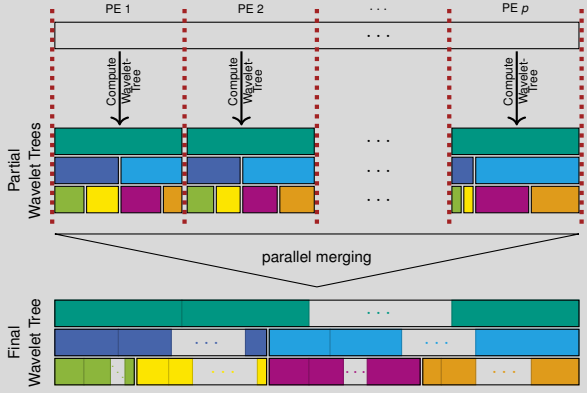


- at most $\lceil \lg \sigma \rceil$ PEs

Parallel Prefix Sorting

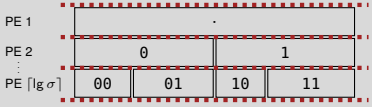


Domain Decomposition



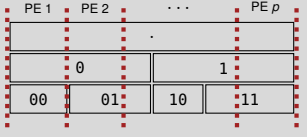
Parallel Wavelet Tree Construction

Parallel Prefix Counting

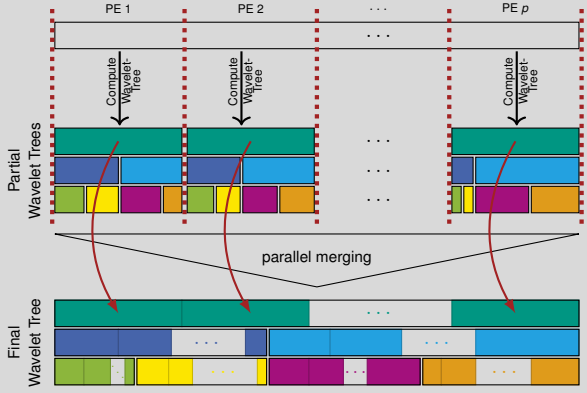


- at most $\lceil \lg \sigma \rceil$ PEs

Parallel Prefix Sorting

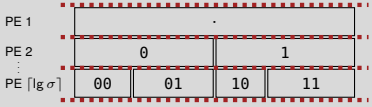


Domain Decomposition



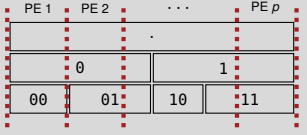
Parallel Wavelet Tree Construction

Parallel Prefix Counting

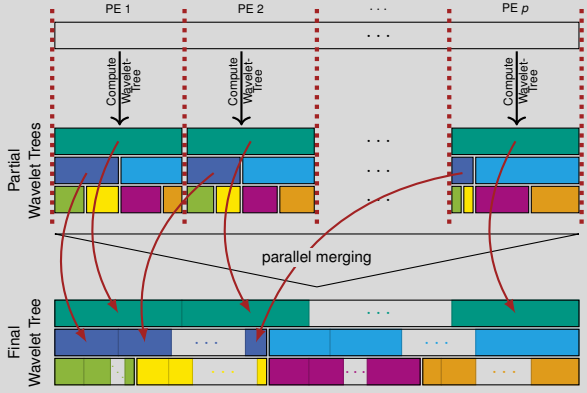


- at most $\lceil \lg \sigma \rceil$ PEs

Parallel Prefix Sorting

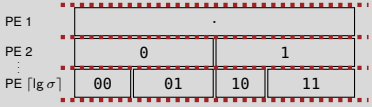


Domain Decomposition



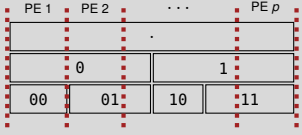
Parallel Wavelet Tree Construction

Parallel Prefix Counting

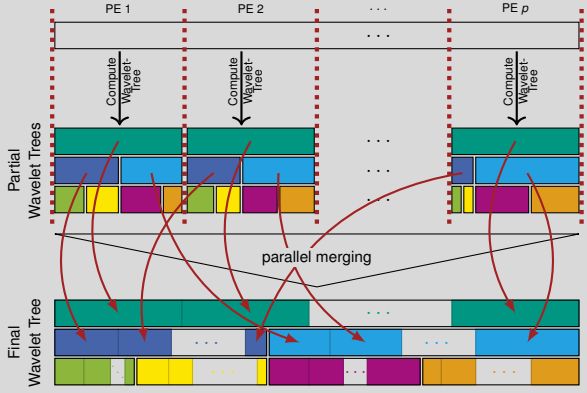


- at most $\lceil \lg \sigma \rceil$ PEs

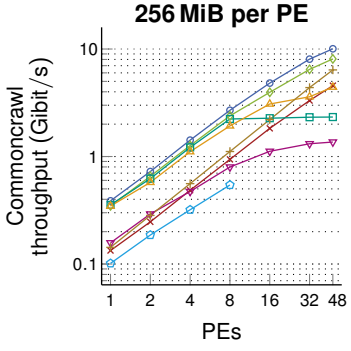
Parallel Prefix Sorting

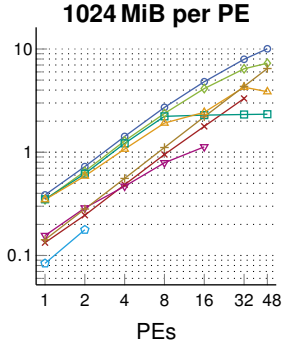


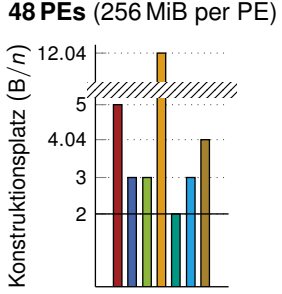
Domain Decomposition

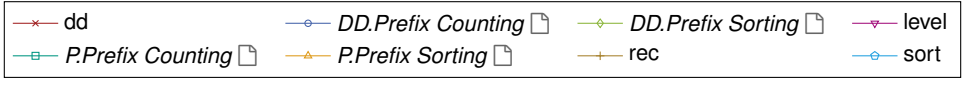


Parallel Wavelet Tree Construction: Experimental Evaluation

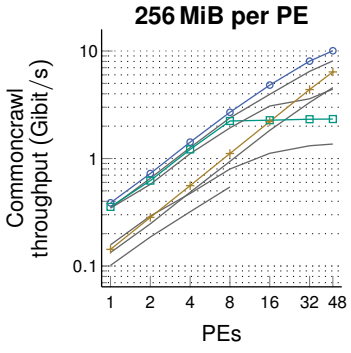


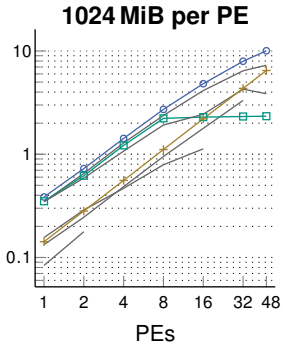


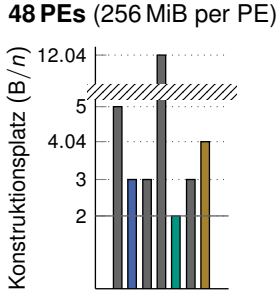


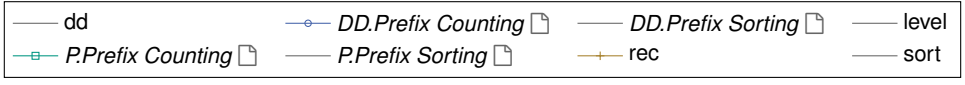


Parallel Wavelet Tree Construction: Experimental Evaluation

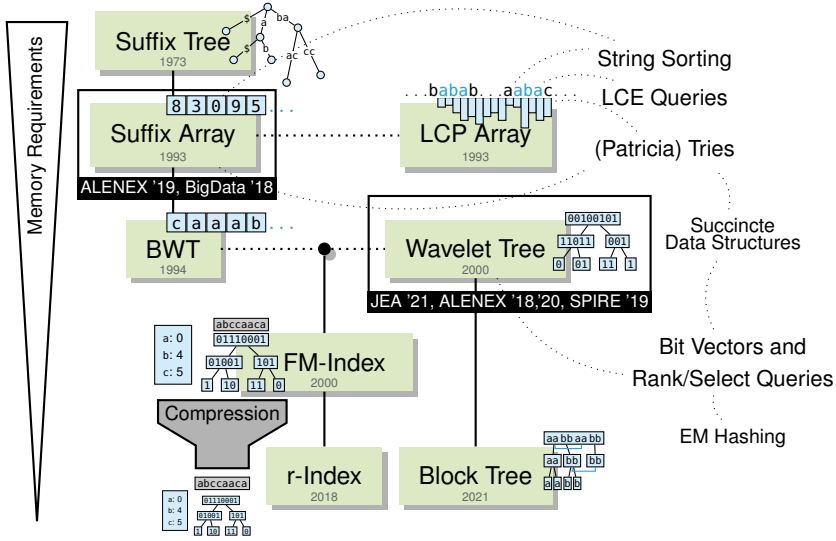




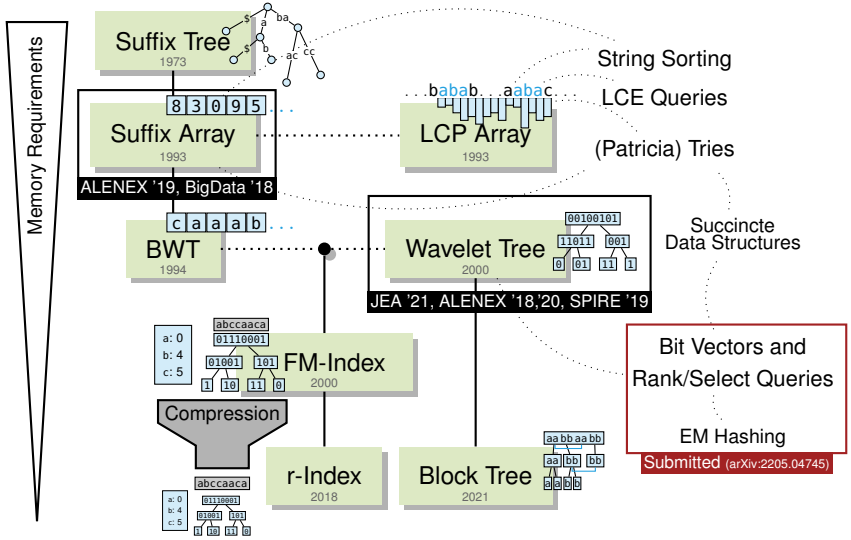




Overview: Text Indices and Main Results



Overview: Text Indices and Main Results



Rank Queries in Bit Vectors

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

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

0	1	2	3	4	5	6	7	8	9
0	1	1	0	1	1	0	1	0	0

Rank Queries in Bit Vectors

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

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

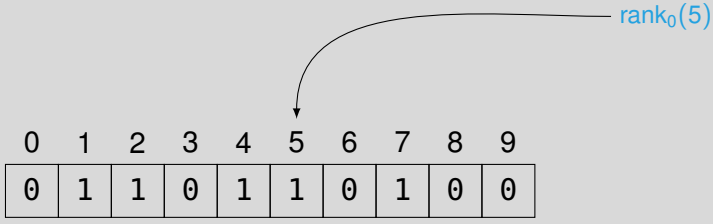
$\text{rank}_0(5)$

0	1	2	3	4	5	6	7	8	9
0	1	1	0	1	1	0	1	0	0

Rank Queries in Bit Vectors

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

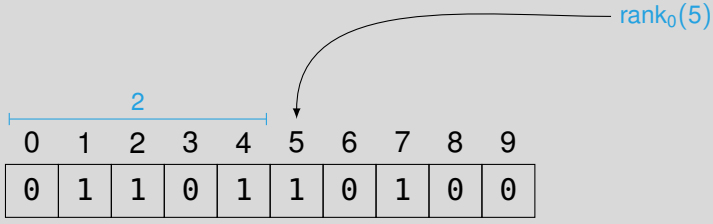
$\text{select}_\alpha(j)$ position of j-th α



Rank Queries in Bit Vectors

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

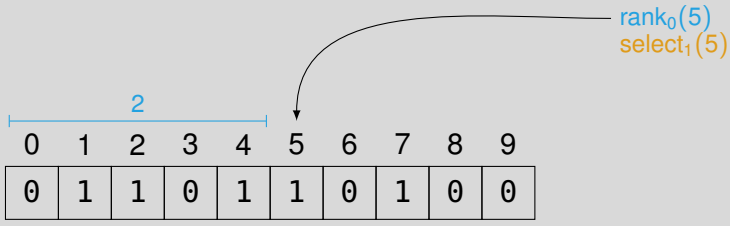
$\text{select}_\alpha(j)$ position of j-th α



Rank Queries in Bit Vectors

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

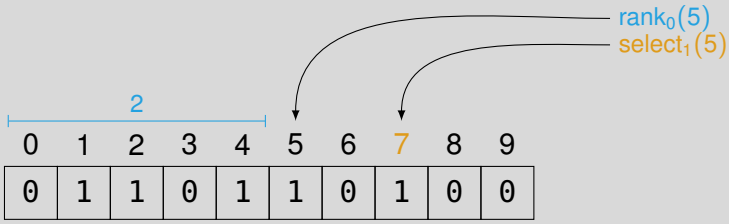
$\text{select}_\alpha(j)$ position of j-th α



Rank Queries in Bit Vectors

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

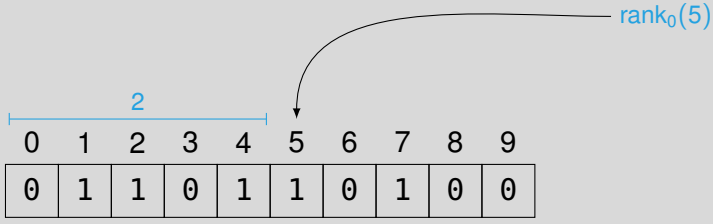
$\text{select}_\alpha(j)$ position of j-th α



Rank Queries in Bit Vectors

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

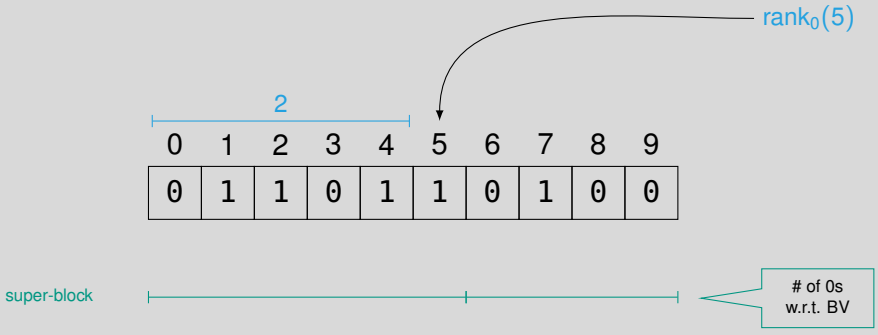
$\text{select}_\alpha(j)$ position of j-th α



Rank Queries in Bit Vectors

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

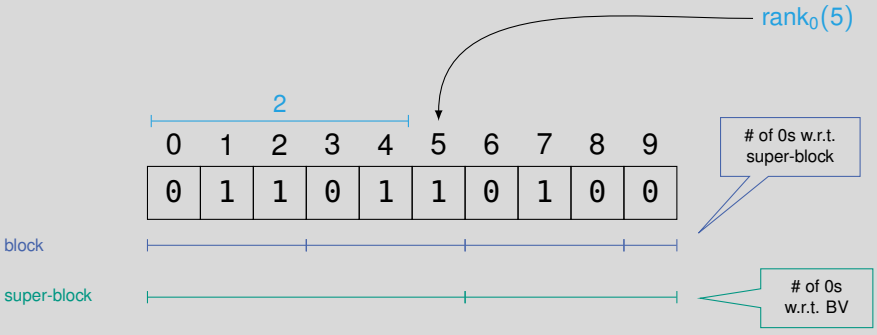
$\text{select}_\alpha(j)$ position of j-th α



Rank Queries in Bit Vectors

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

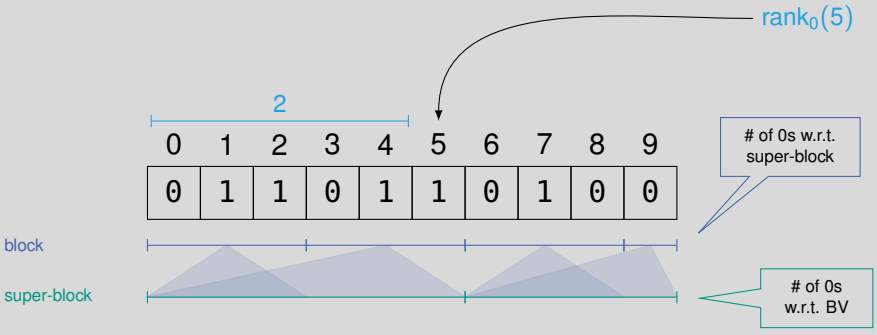
$\text{select}_\alpha(j)$ position of j-th α



Rank Queries in Bit Vectors

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

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



Bit Vectors with Rank and Select Support

cs-poppy

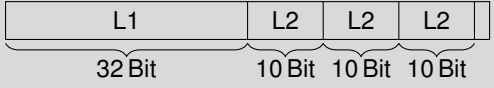
- Blocks (L2): 512 bits
- Super-Blocks (L1): 2048 bits
- Super-Super-Blocks (L0): 2^{31} bits
- 3.51 % space overhead

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

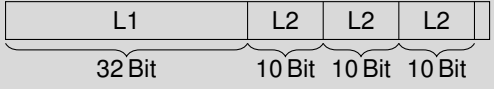


Bit Vectors with Rank and Select Support

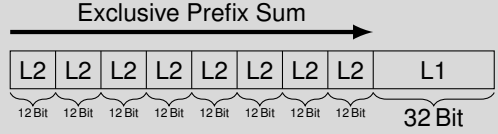
cs-poppy

- 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

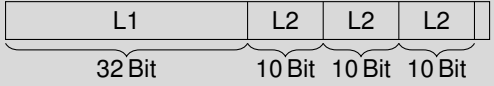


Bit Vectors with Rank and Select Support

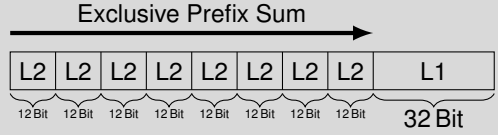
cs-poppy

- 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



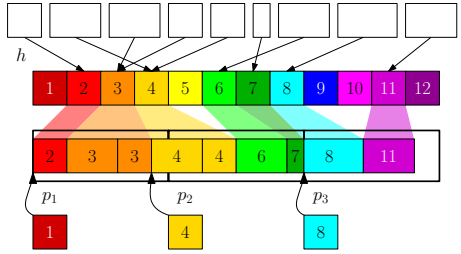
SIMD Access

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

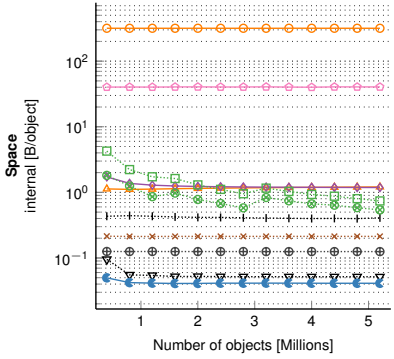
EM Hashing: PaCHash

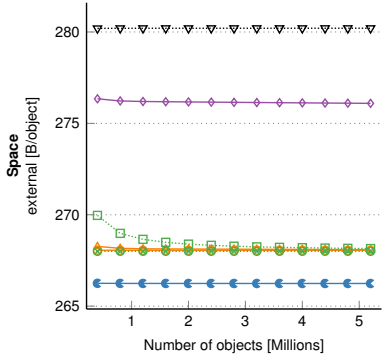
General Idea

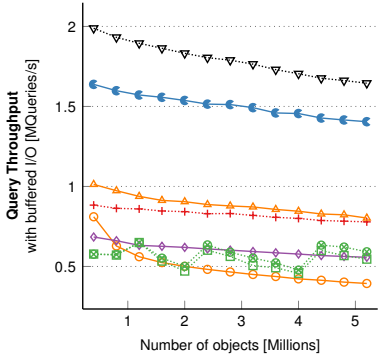
- small internal memory index
- Elias-Fano representataion
- $2 + \log a + o(1)$ bits per EM block
- $1.4427 + \log(a + 1) + o(1)$ theoretically possible
- variable size objects contiguously in EM
- retrieving x loads $\leq 1 + |x|/B + 1/a$ blocks in expectation

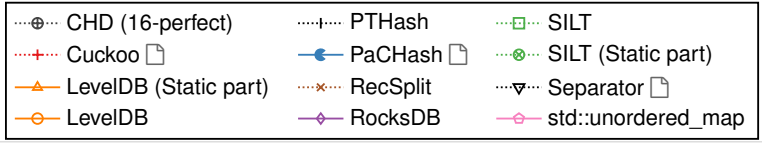


PaCHash Experiments (Fixed Size Objects)

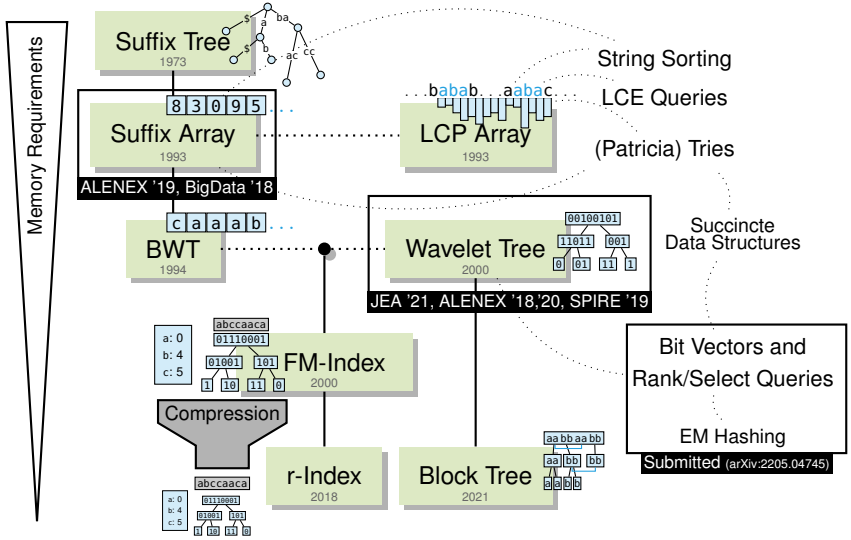








Thank You!


Memory Requirements
Suffix Tree
1973
Suffix Array
1993
ALENEX '19, BigData '18
BWT
1994
LCP Array
1993
Wavelet Tree
2000
JEA '21, ALENEX '18,'20, SPIRE '19
FM-Index
2000
Compression
r-Index
2018
Block Tree
2021
String Sorting
LCE Queries
(Patricia) Tries
Succincte
Data Structures
Bit Vectors and
Rank/Select Queries
EM Hashing
Submitted (arXiv:2205.04745)