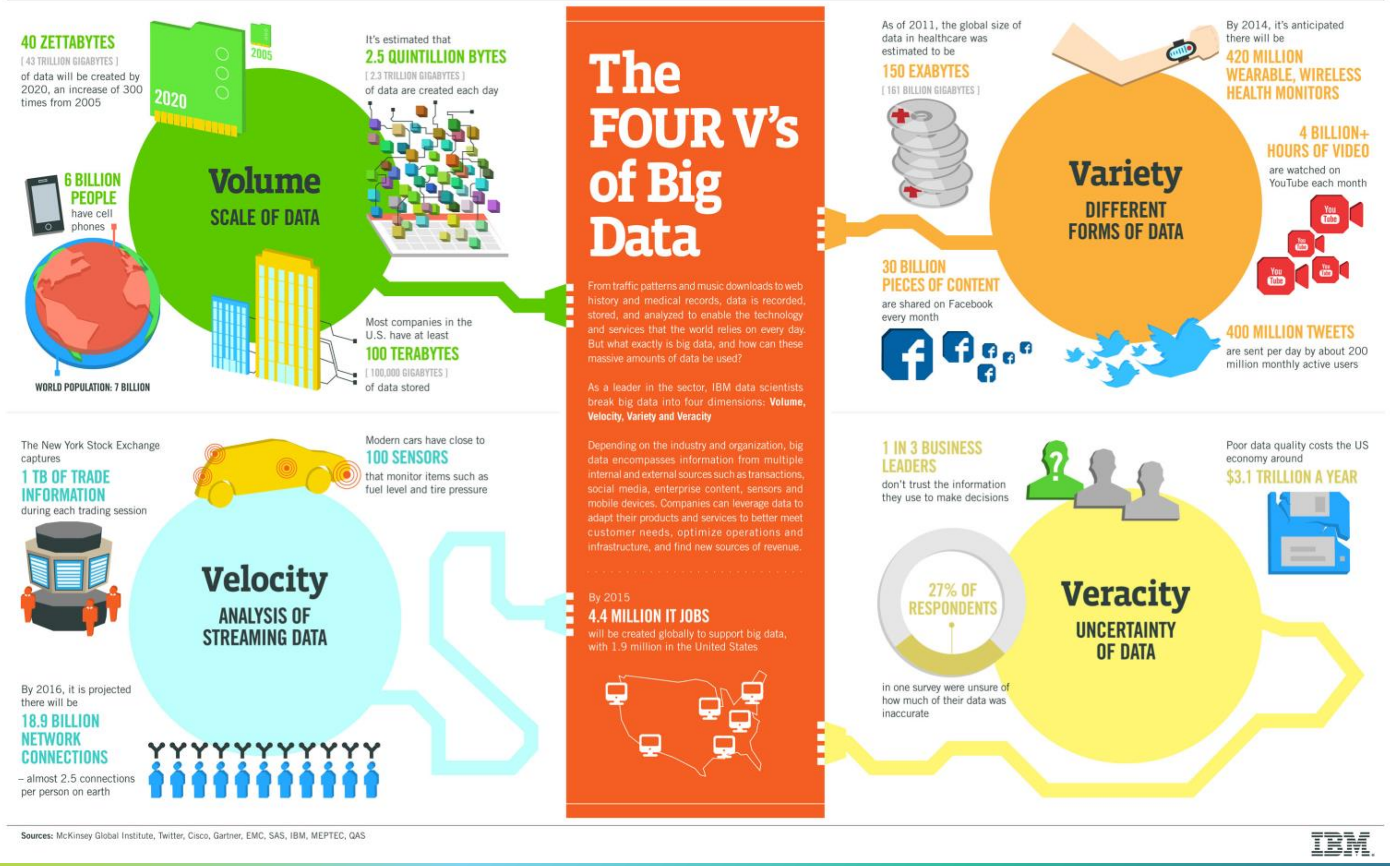
Dresden Database Systems Group

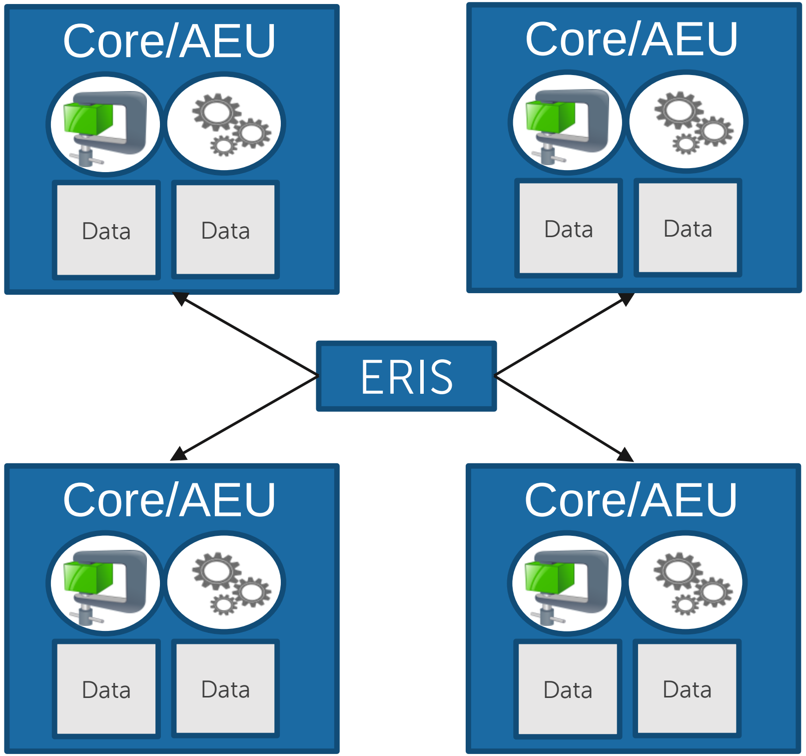
Lightweight compression algorithms and interfaces for flexible storage architectures

Master thesis defense – Paul Peschel – 17.12.2015

Motivation - Big Data

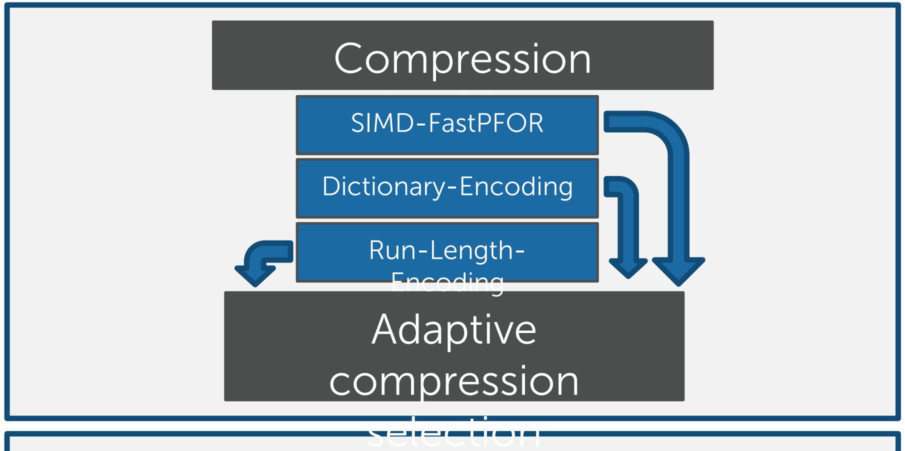


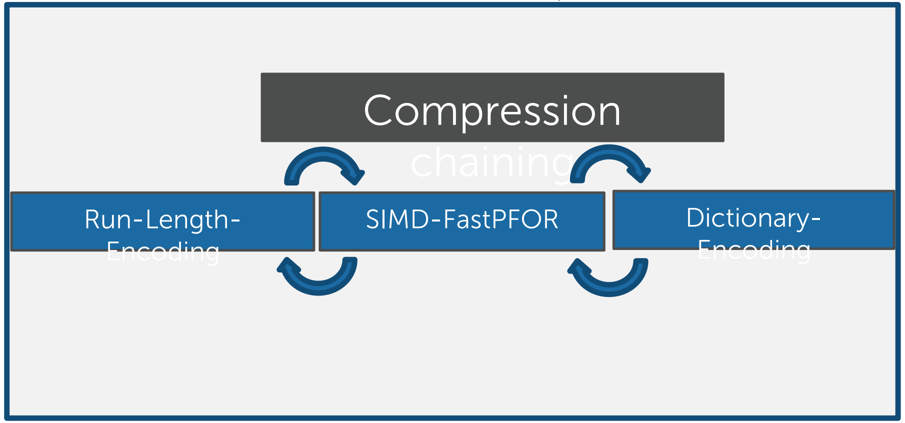
Motivation - solution: Eris & compression

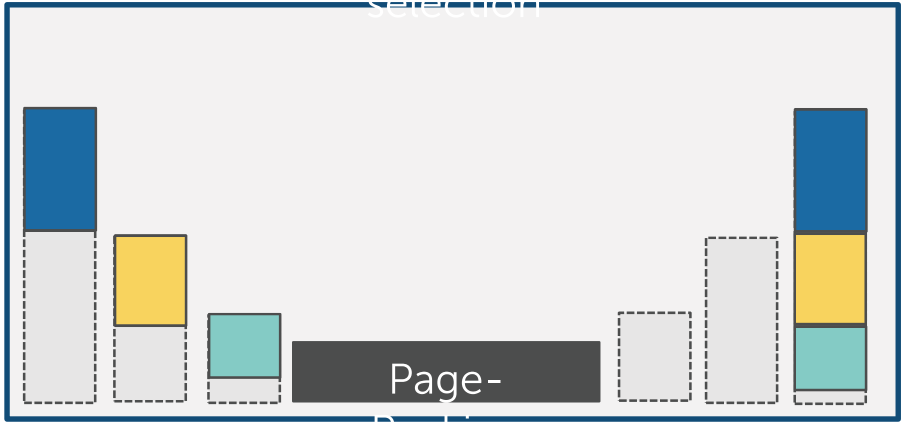


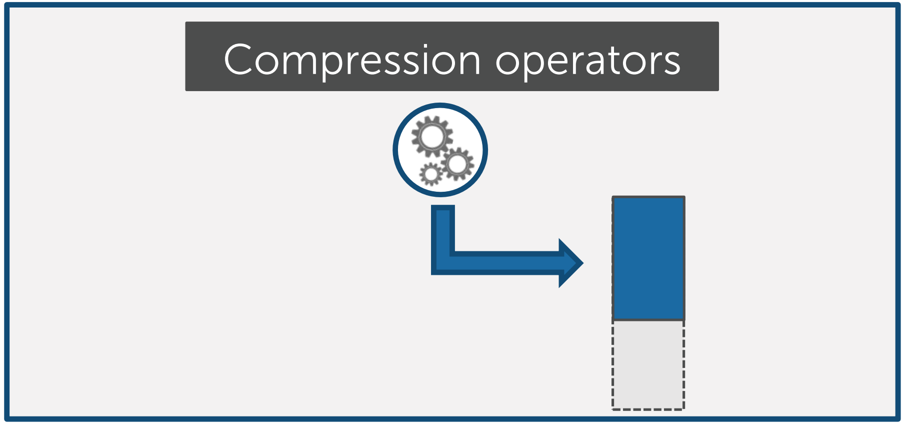
- Volume: data partitioning and compression
- Variety: different compression algorithms for different data properties
- Compression and storage should be flexible with respect to the data
- →Adaptive compression selects almost an appropriate compression algorithm for the given data
- →Simple integration of new compression algorithms

Overview


Compression
SIMD-FastPFOR
Dictionary-Encoding
Run-Length-
Encoding
Adaptive
compression
selection


Compression
chaining
Run-Length-
Encoding
SIMD-FastPFOR
Dictionary-
Encoding


Page-


Compression operators

Numerical series

Alternating

- $A_1, B_2, \cdots, A_{n-1}, B_n$

42 128 … 42 128

HalfRange

- $A_1, A_2, \cdots, A_{n/2}, B_{n/2+1}, \cdots, B_n$

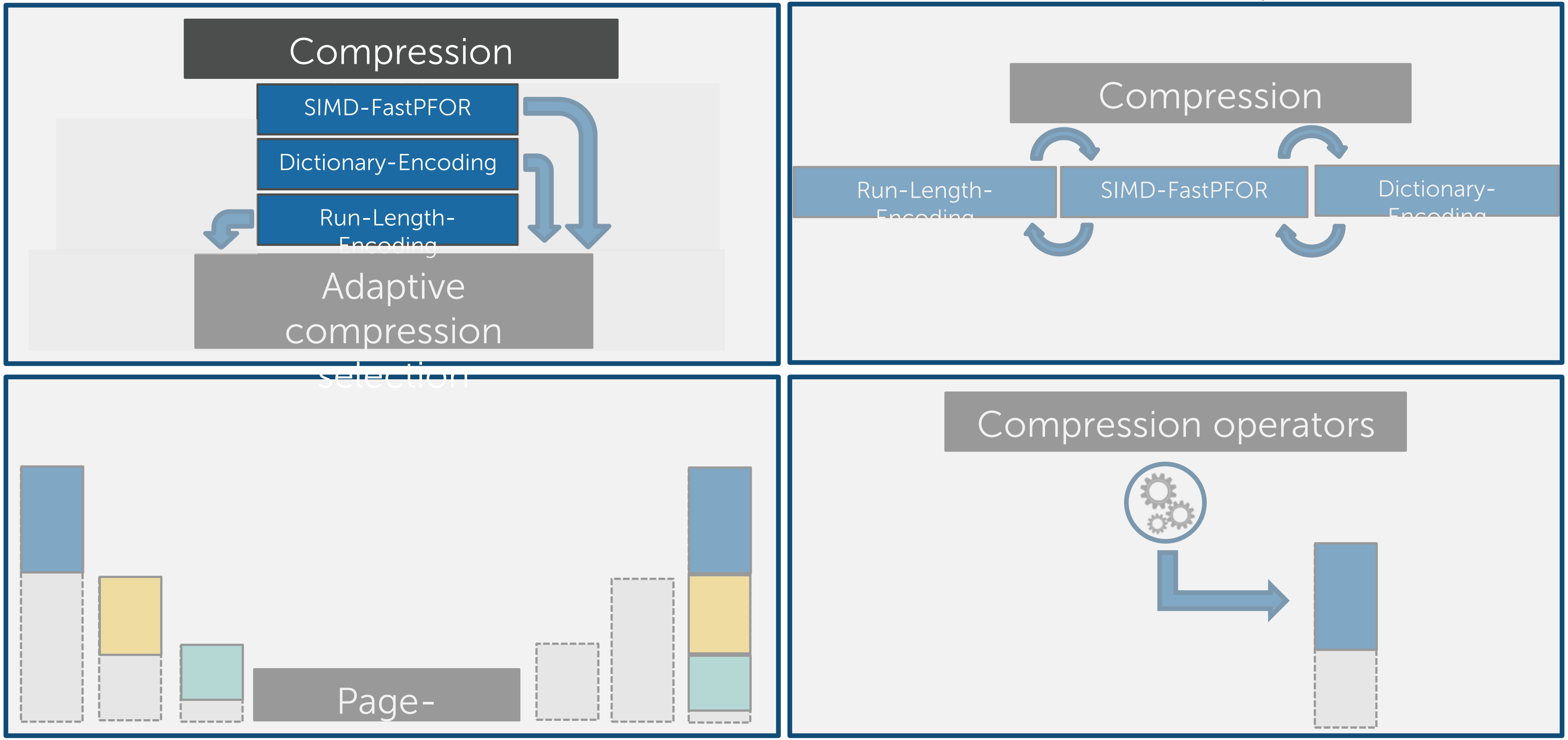
42 … 42 128 … 128

Ordered

- $A_1, A_2, \cdots, A_n$

1 2 …

Overview


Compression
SIMD-FastPFOR
Dictionary-Encoding
Run-Length-Encoding
Adaptive compression selection
Compression
Run-Length-Encoding
SIMD-FastPFOR
Dictionary-Encoding
Page-
Compression operators

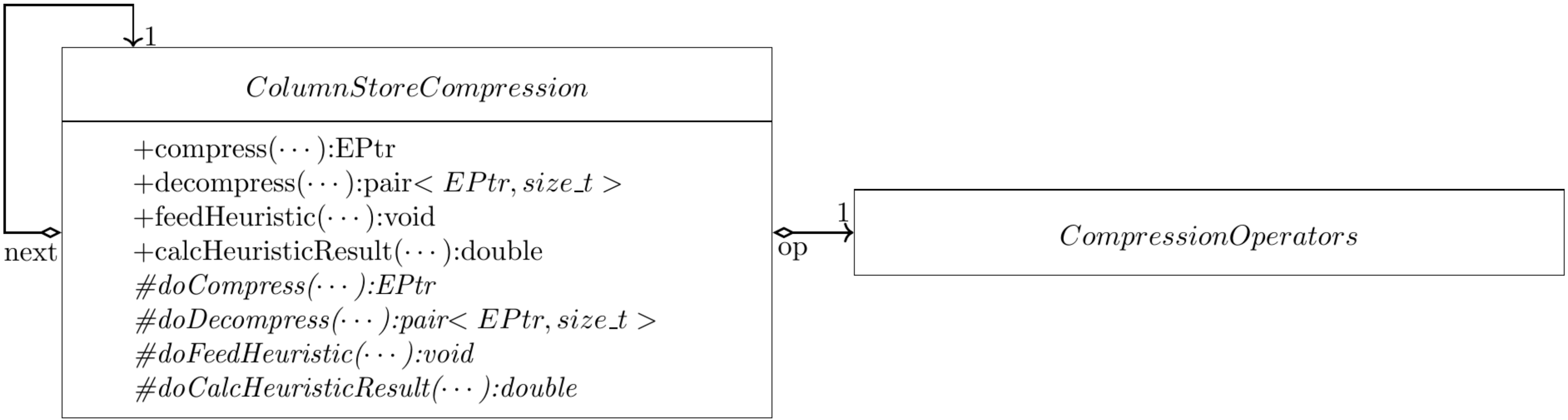
Compression algorithms

Dresden Database
Systems Group

Compression Interface

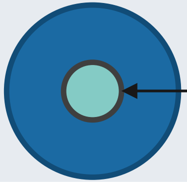
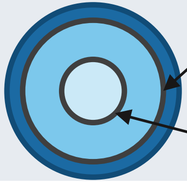
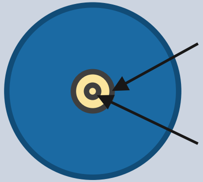
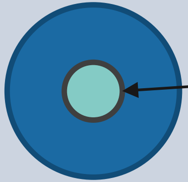
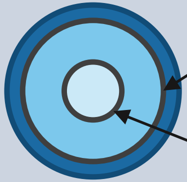
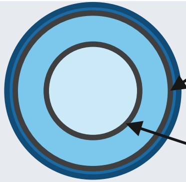
- **Run-Length-Encoding**
  - Encodes runs of equal values to (value;run-length)-pairs; Coherent RLE: value and runlength sections separated
- **Dictionary-Encoding**
  - Creates dictionary of distinct values
  - Replaces values by dictionary index
- **SIMD-FastPFOR**
  - Calculates bit width b using a cost function
  - Removes bit widths greater b from values and saves the removed bit widths as exceptions

Compression interface

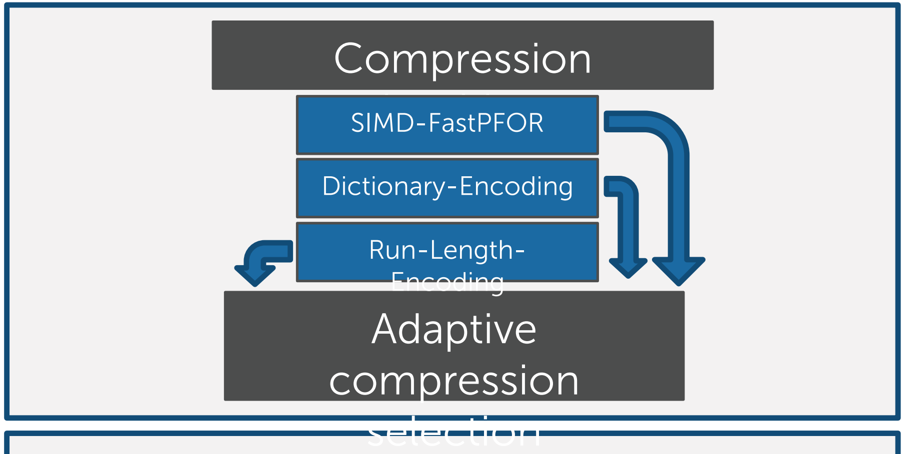


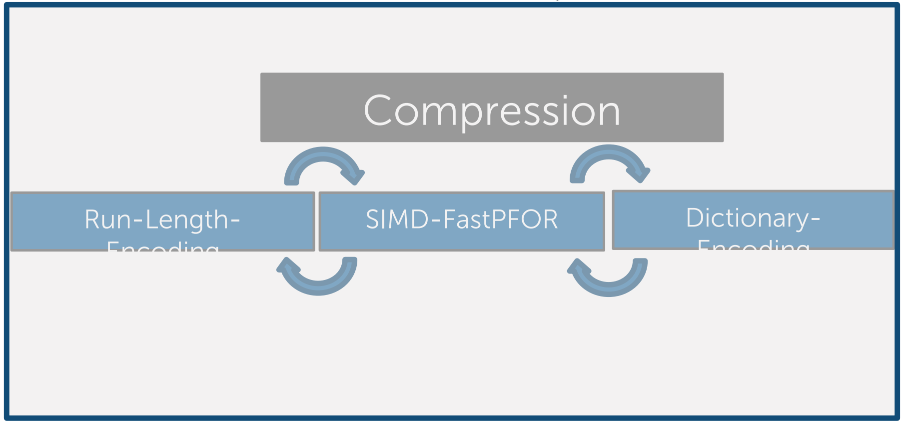
Compression algorithms – compression rates

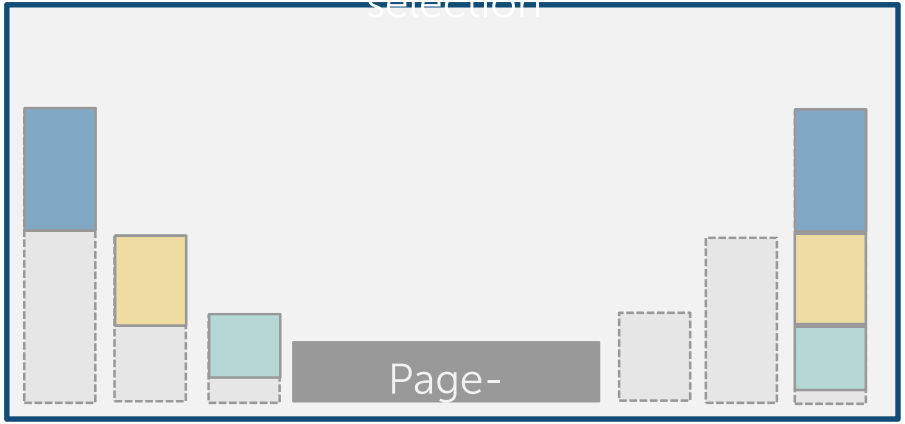
Dresden Database Systems Group

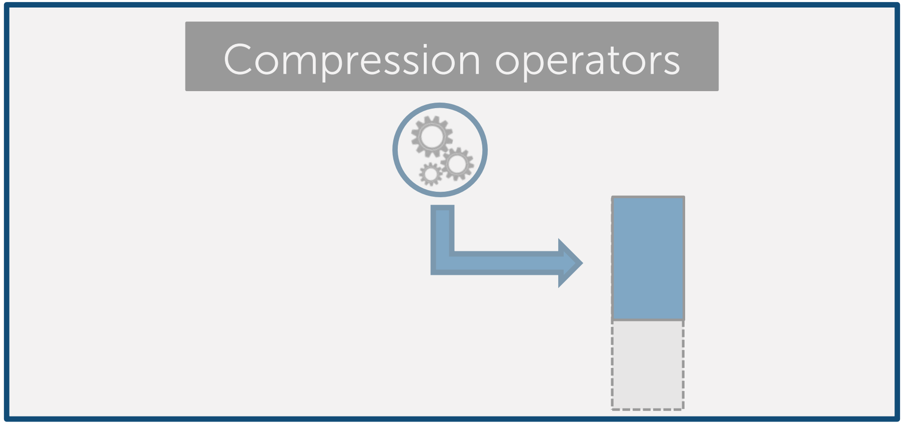
Compression rates			
	Run-Length-Encoding	Dictionary-Encoding	SIMD-FastPFOR
Alternating $A_1, B_2, \cdots, A_{n-1}, B_n$	-		
HalfRange $A_1, A_2, \cdots, A_{n/2}, B_{n/2+1}, \cdots, B_n$			
Ordered $A_1, A_2, \cdots, A_n$	-	-	

Overview

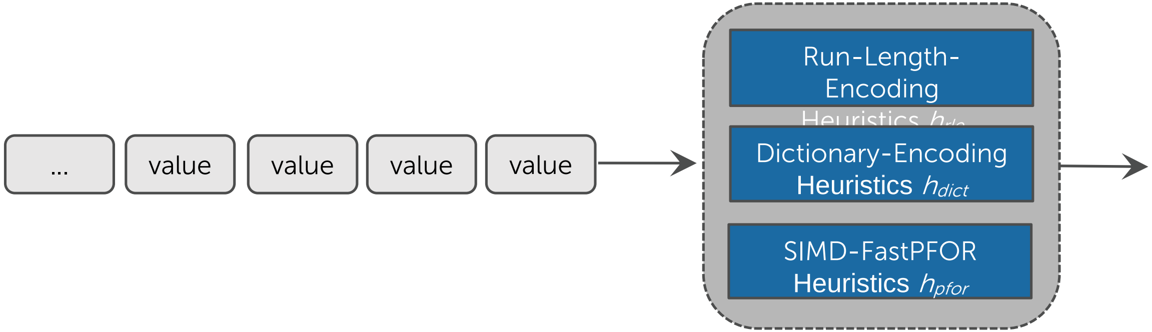

Compression
SIMD-FastPFOR
Dictionary-Encoding
Run-Length-Encoding
Adaptive compression selection


Compression
Run-Length-Encoding
SIMD-FastPFOR
Dictionary-Encoding


Page-


Compression operators

Adaptive compression selection



$$h_k = min(h_{rle}, h_{dict}, h_{pfor})$$

Selecting compression algorithm
k with the smallest heuristics result h_k

RLE heuristics

- Heuristics evaluate compression rates for given data

Run-Length-Encoding heuristics h_{rle}

- $h_{rle} = \frac{2*(valueChanges+1)}{n}$

Value A	Value B	Value B	Value B

valueChange = 1 n = 4

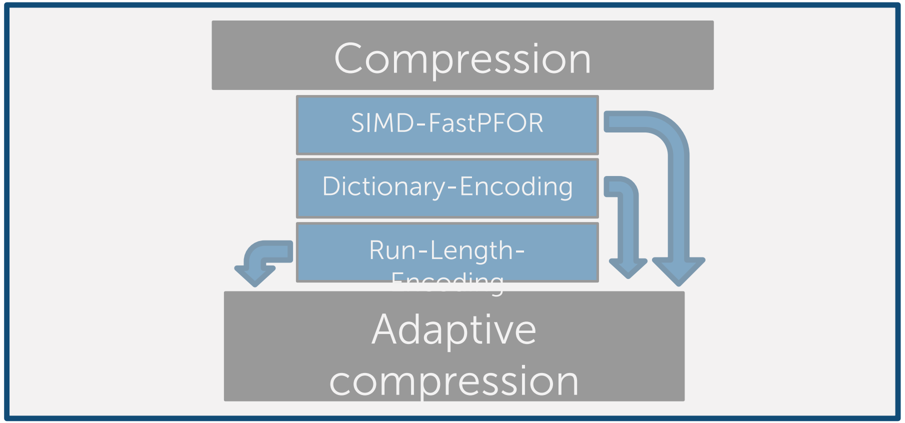
Heuristics results

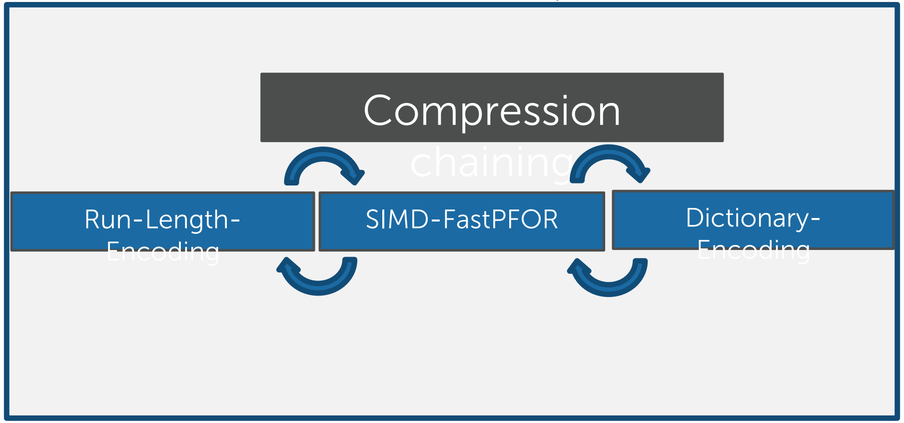
Dresden Database Systems Group

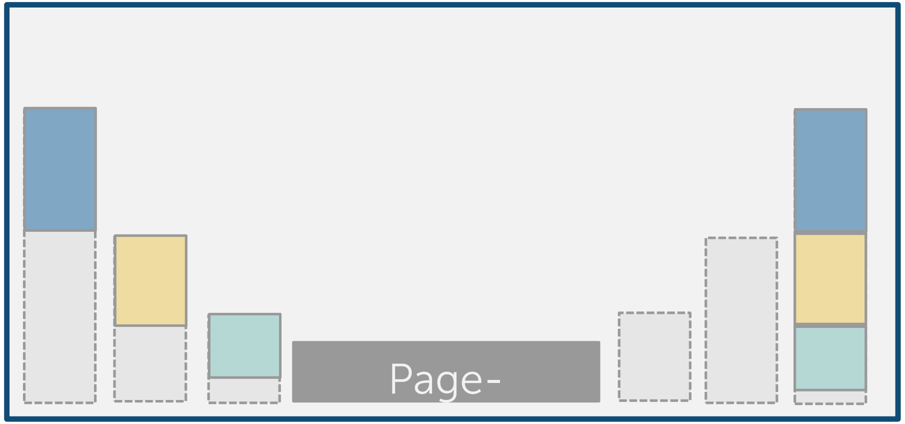
Heuristics results

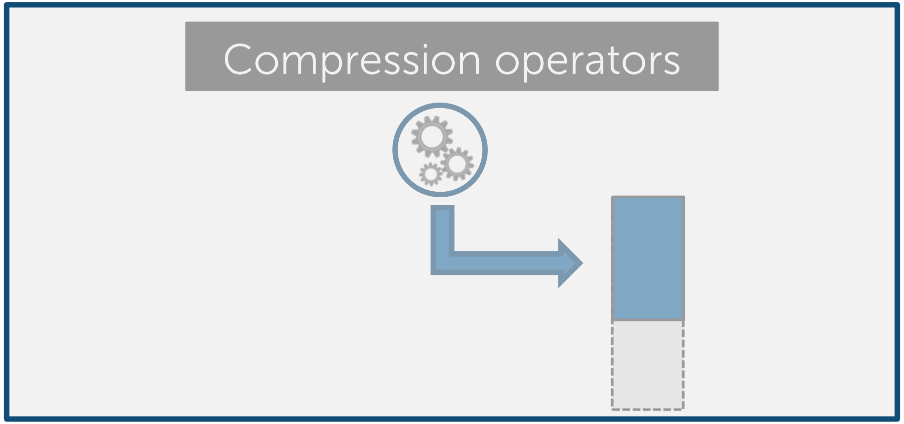
	FastPFOR	RLE	Dictionary	Selected
Alternating $A_1, B_2, \cdots, A_{n-1}, B_n$	27% 22%	200%	25% 25%	FastPFOR (better would be Dictionary-Encoding)
HalfRange $A_1, A_2, \cdots, A_{n/2}, B_{n/2+1}, \cdots, B_n$	25% 24%	0.1% <0.01%	25% 25%	RLE (RLE is the best selection)
Ordered $A_1, A_2, \cdots, A_n$	50% 43%	200%	150%	FastPFOR (FastPFOR is the best selection)
Alternating (30-Bit values)	98% 94%	200%	25% 25%	Dictionary (Dictionary is the best selection)

Overview

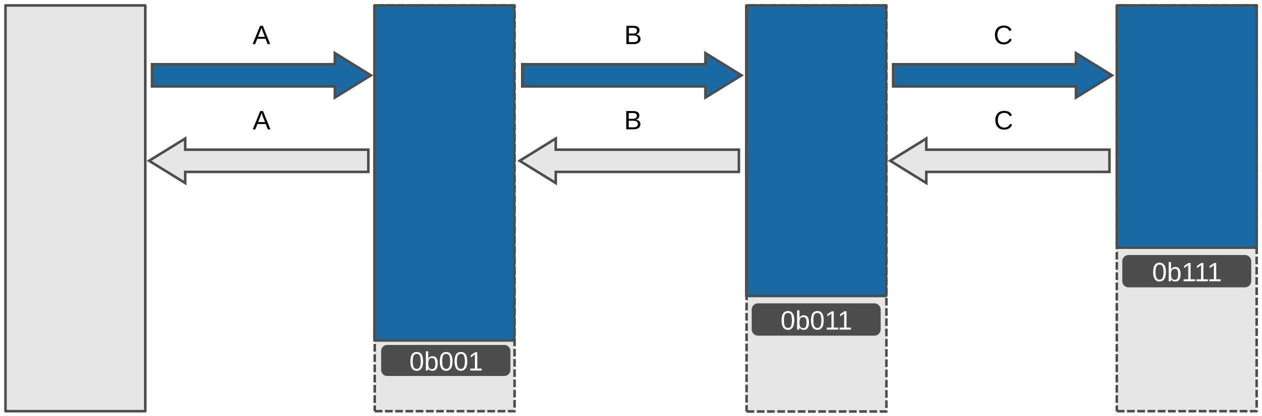

Compression
SIMD-FastPFOR
Dictionary-Encoding
Run-Length-Encoding
Adaptive compression


Compression
chaining
Run-Length-Encoding
SIMD-FastPFOR
Dictionary-Encoding


Page-


Compression operators

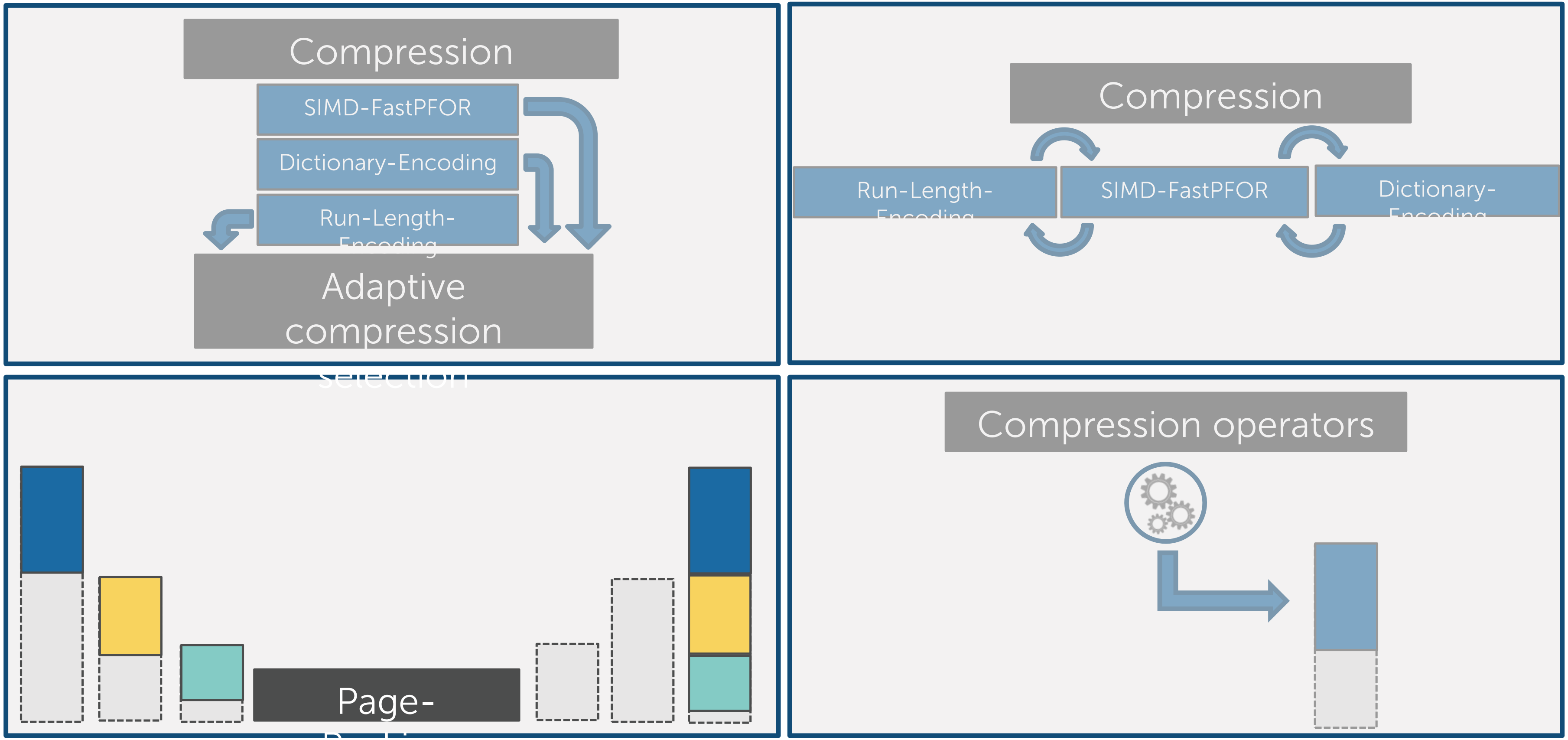
Compression chaining



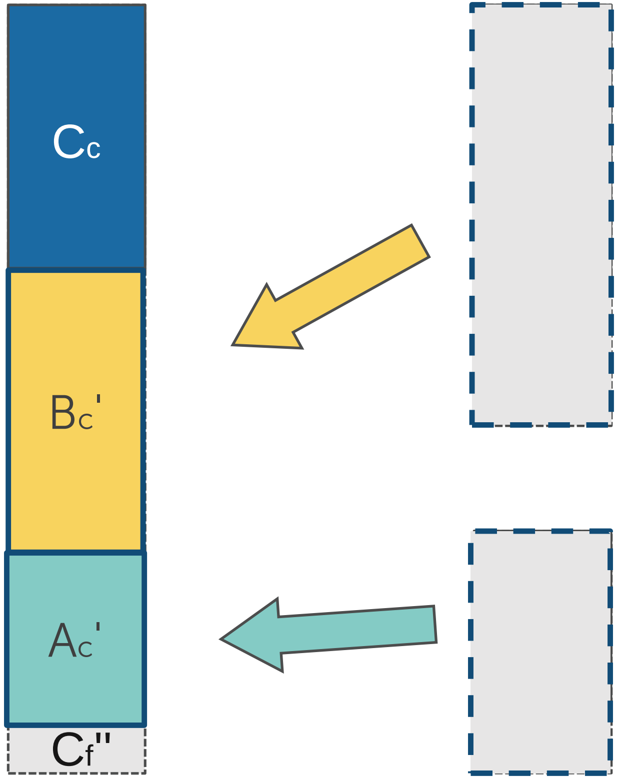
Compression rates

	Dictionary → Coherent RLE → SIMD-FastPFOR	Coherent RLE → SIMD-FastPFOR → Dictionary
Alternating $A_1, B_2, \cdots, A_{n-1}, B_n$	25% 0.05%	25% 6.9%
HalfRange $A_1, A_2, \cdots, A_{n/2}, B_{n/2+1}, \cdots, B_n$	0.1% 0.08%	0.1% 0.05%
Ordered $A_1, A_2, \cdots, A_n$	50% 50%	50% 50%

Overview


Compression
SIMD-FastPFOR
Dictionary-Encoding
Run-Length-Encoding
Adaptive compression selection
Compression
Run-Length-Encoding
SIMD-FastPFOR
Dictionary-Encoding
Page-
Compression operators

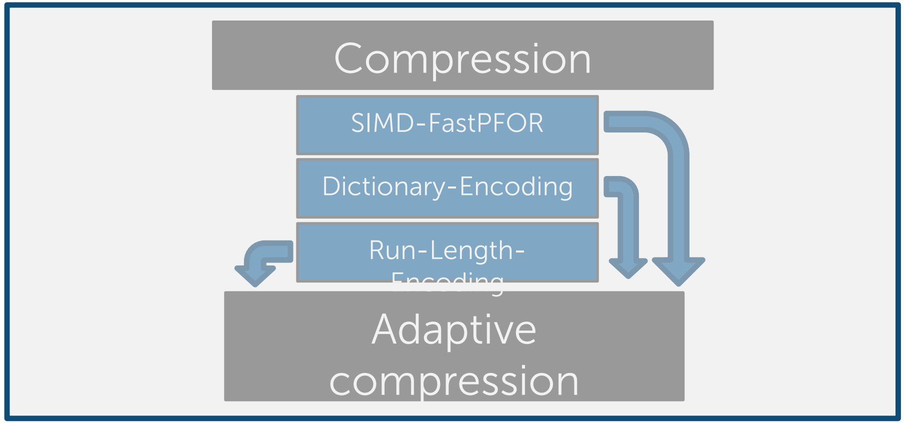
Page-Packing



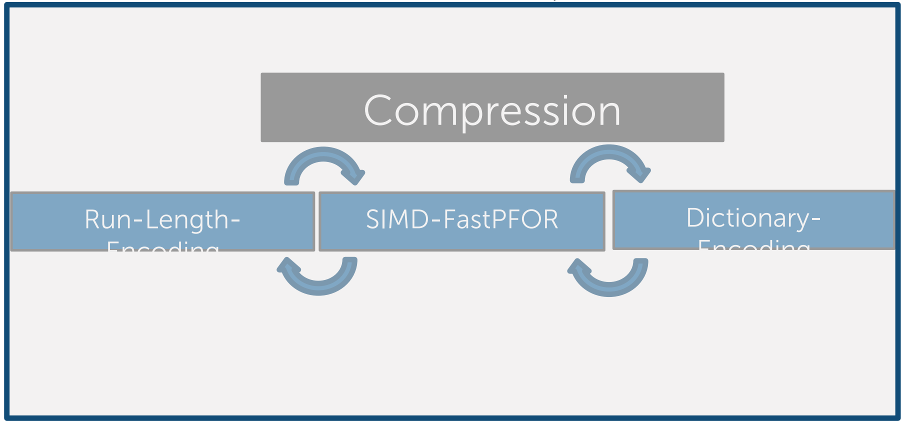
free	pos	originalPage
0	A_f	A
0	B_f	B
1581	C_f''	C

used	pos	parentPage
MOVED	B_f	A_c'
MOVED	A_f	0
14803	C_f''	B_c'

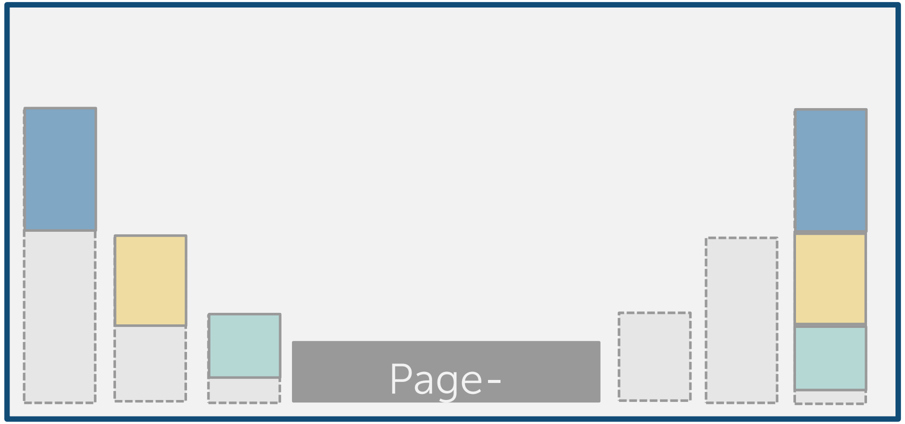
Overview



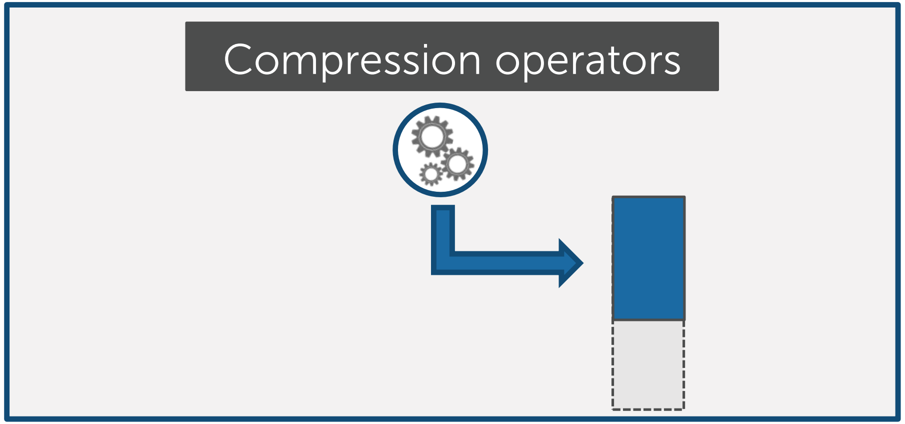
Compression
SIMD-FastPFOR
Dictionary-Encoding
Run-Length-
Adaptive compression



Compression
Run-Length-
SIMD-FastPFOR
Dictionary-



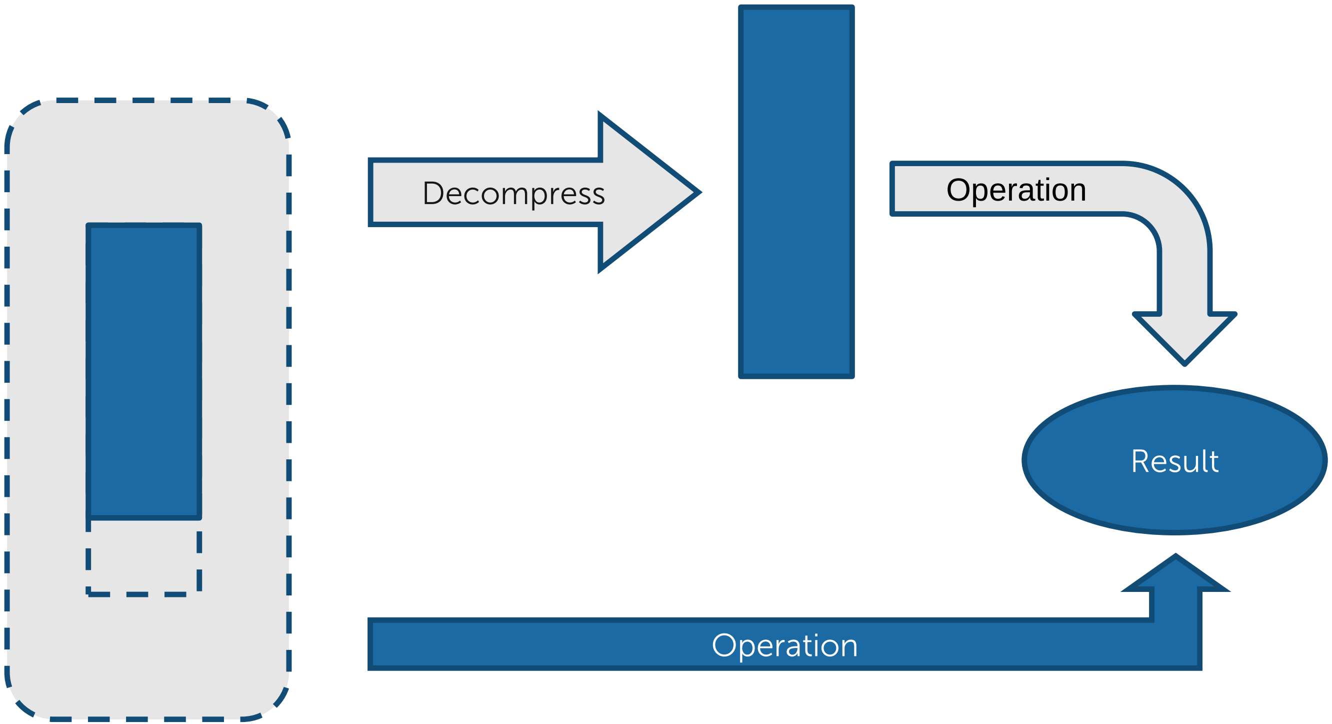
Page-



Compression operators

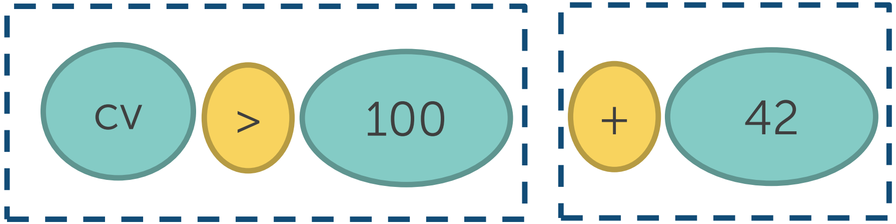
Compression operators - Motivation

Dresden Database Systems Group



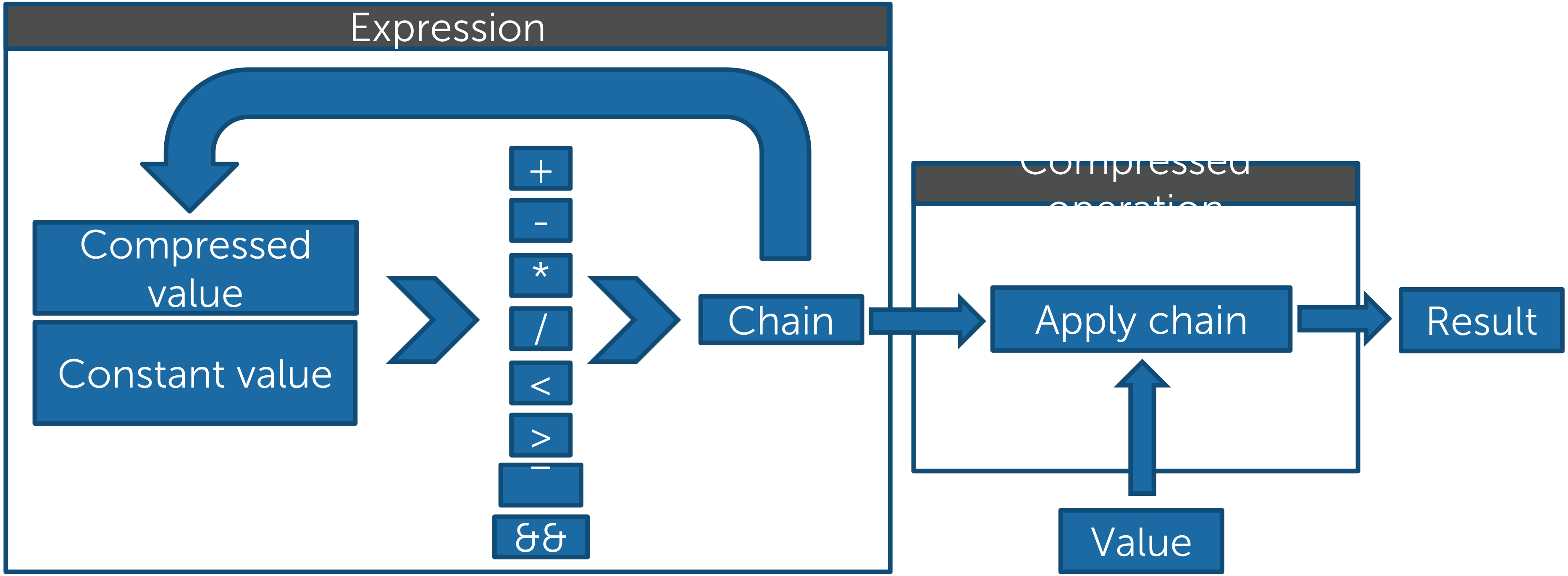
Compression operators – Construction 1

$$(x > 100) + 42$$

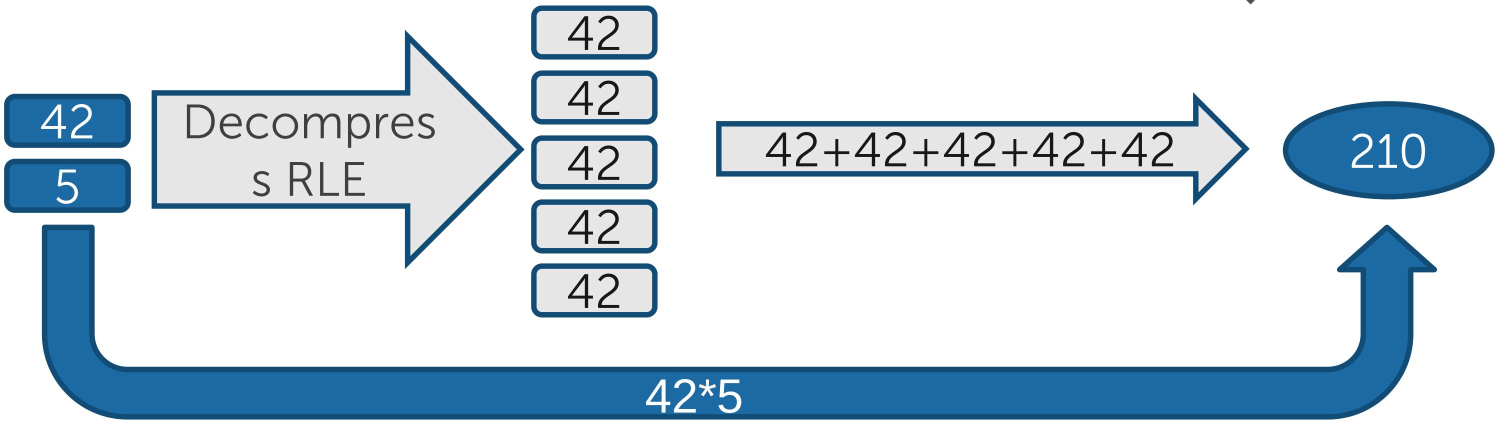


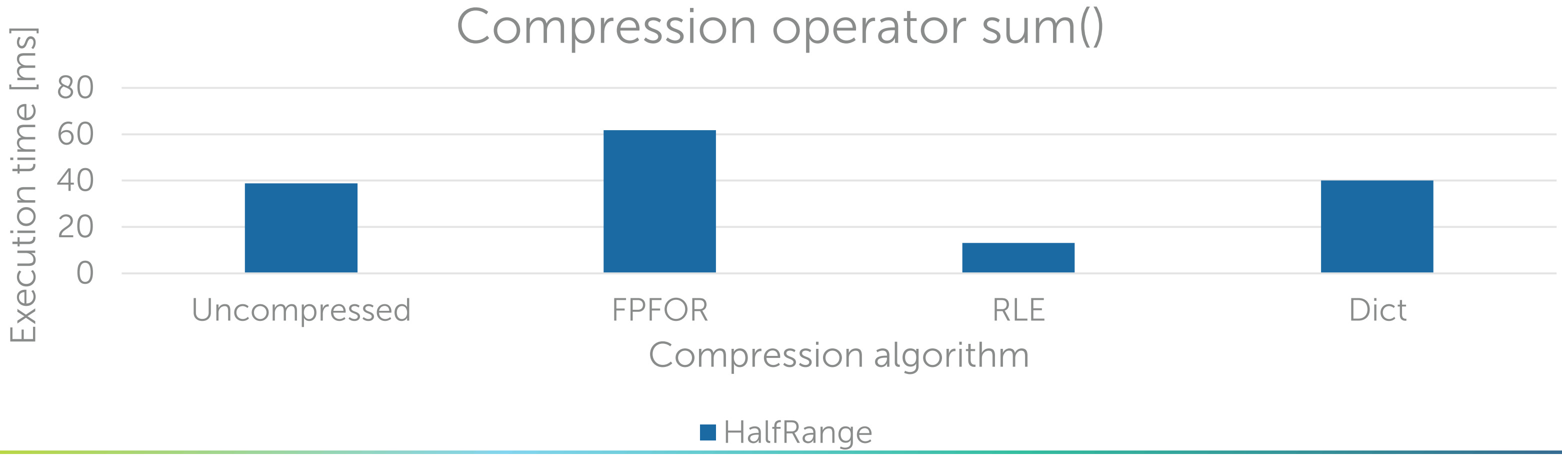
Compression operators – Construction 2

Dresden Database Systems Group

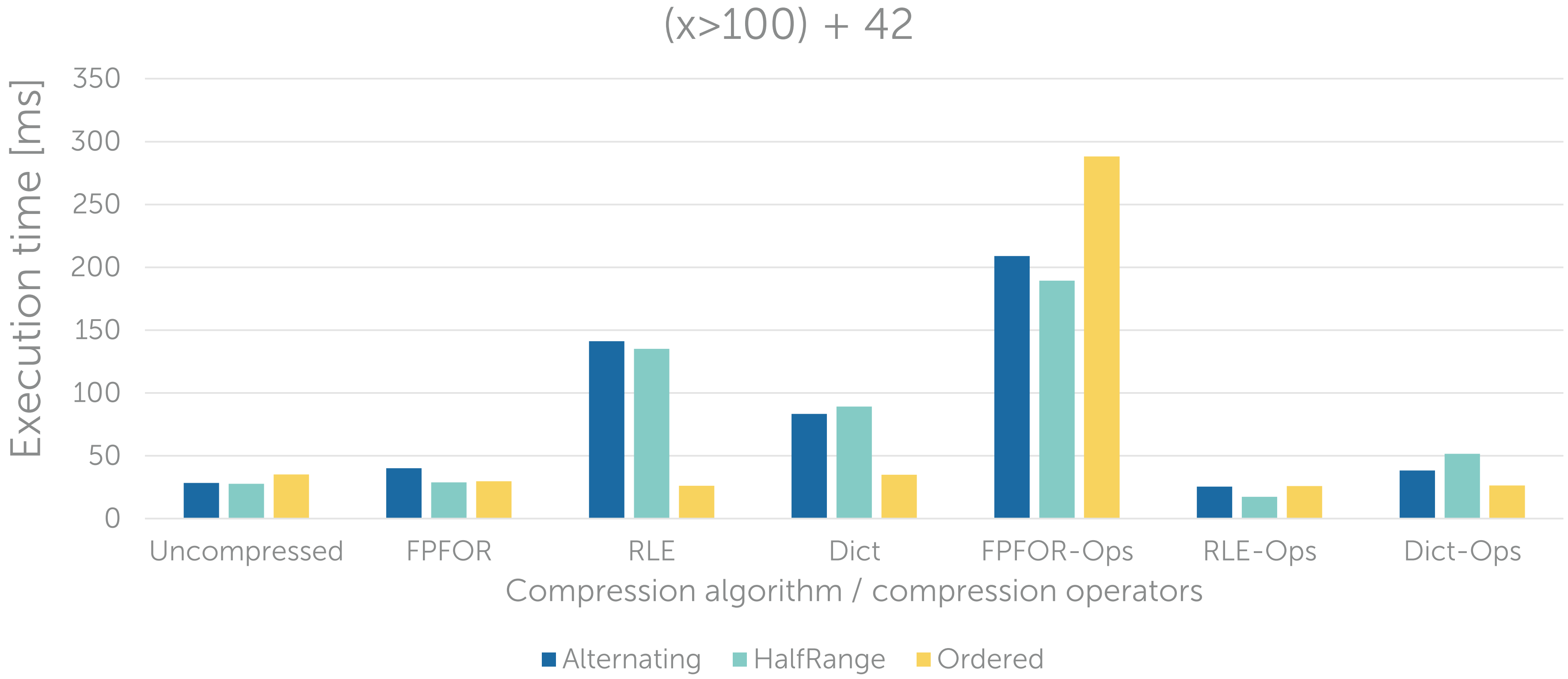


Compression operators – sum()



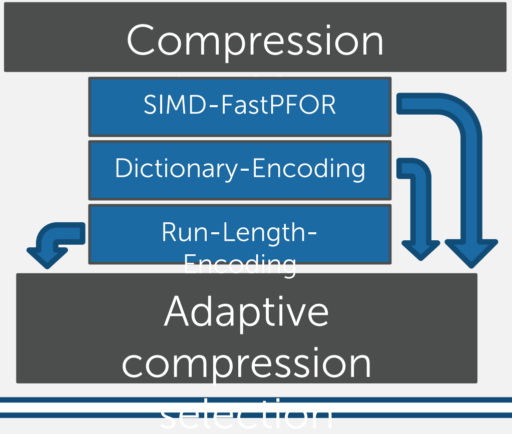


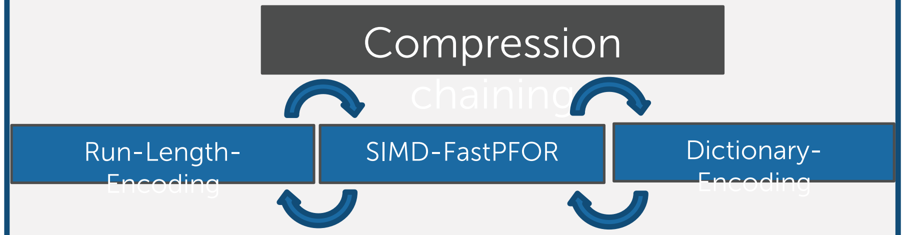
Compression operators evaluation



Conclusions

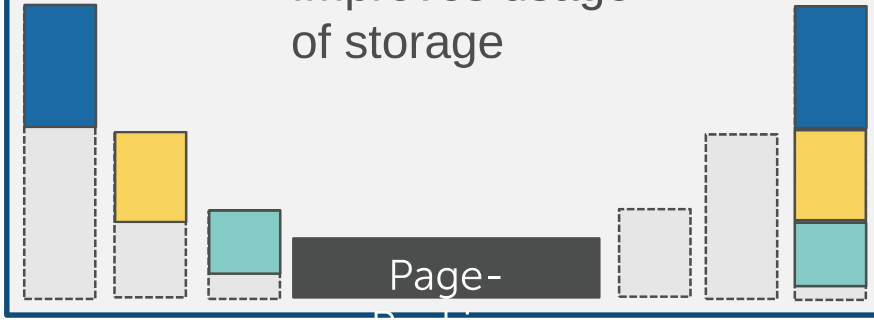
- Easy integration
- Selects appropriate compression algorithm





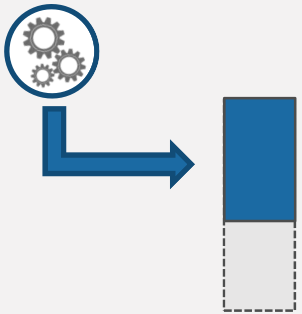
- Improves compression rates

- Improves usage of storage

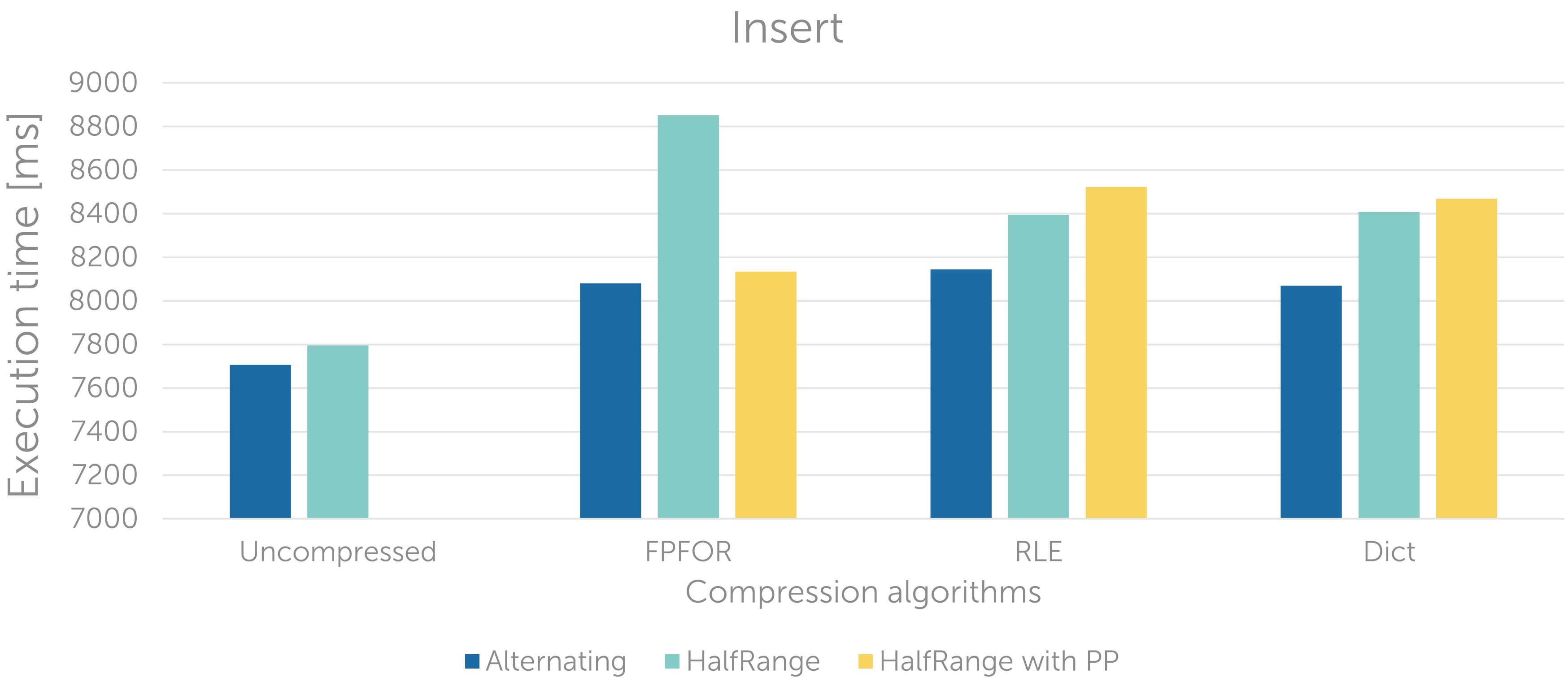


Compression operators

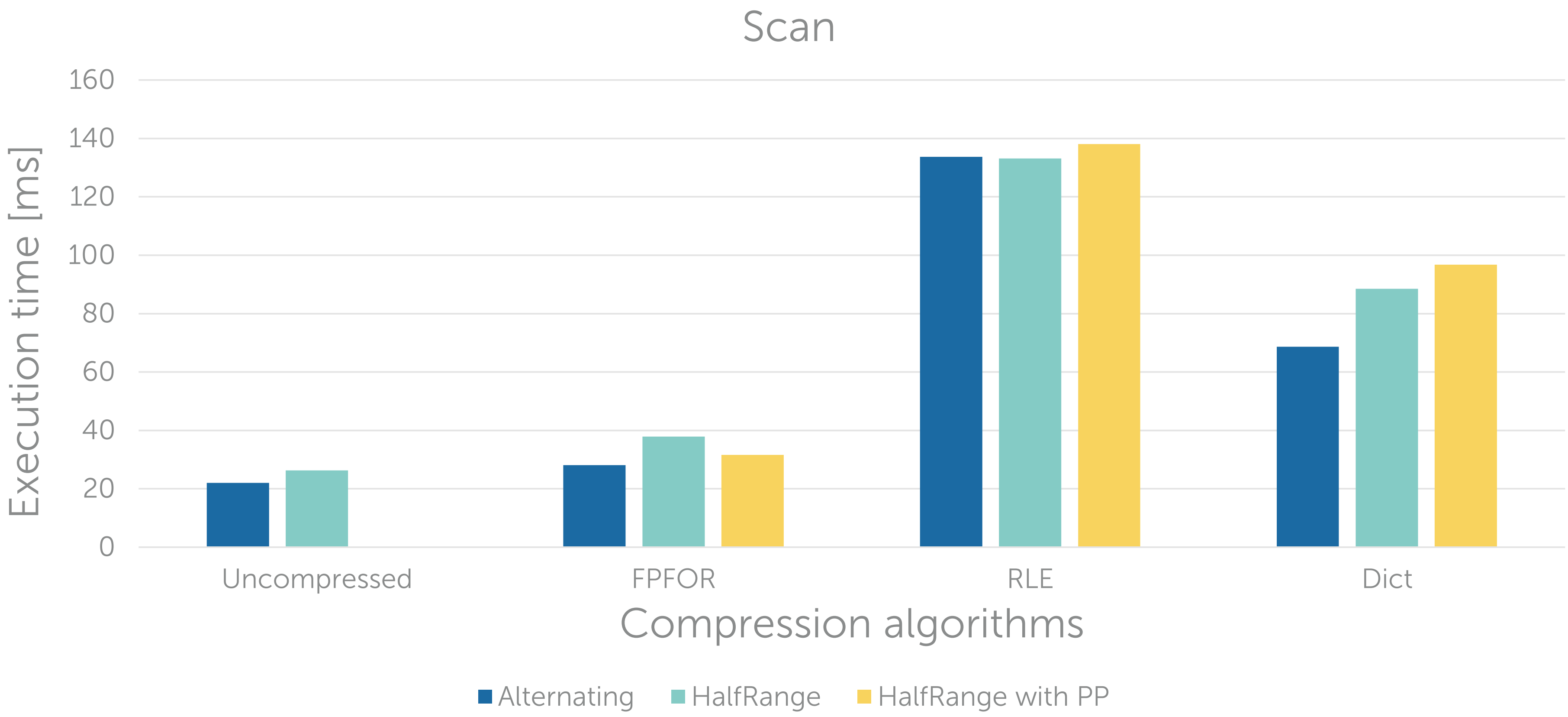
- Faster analytical operations with RLE and Dict
- But not with SIMD-FastPFOR



Backup - Evaluation

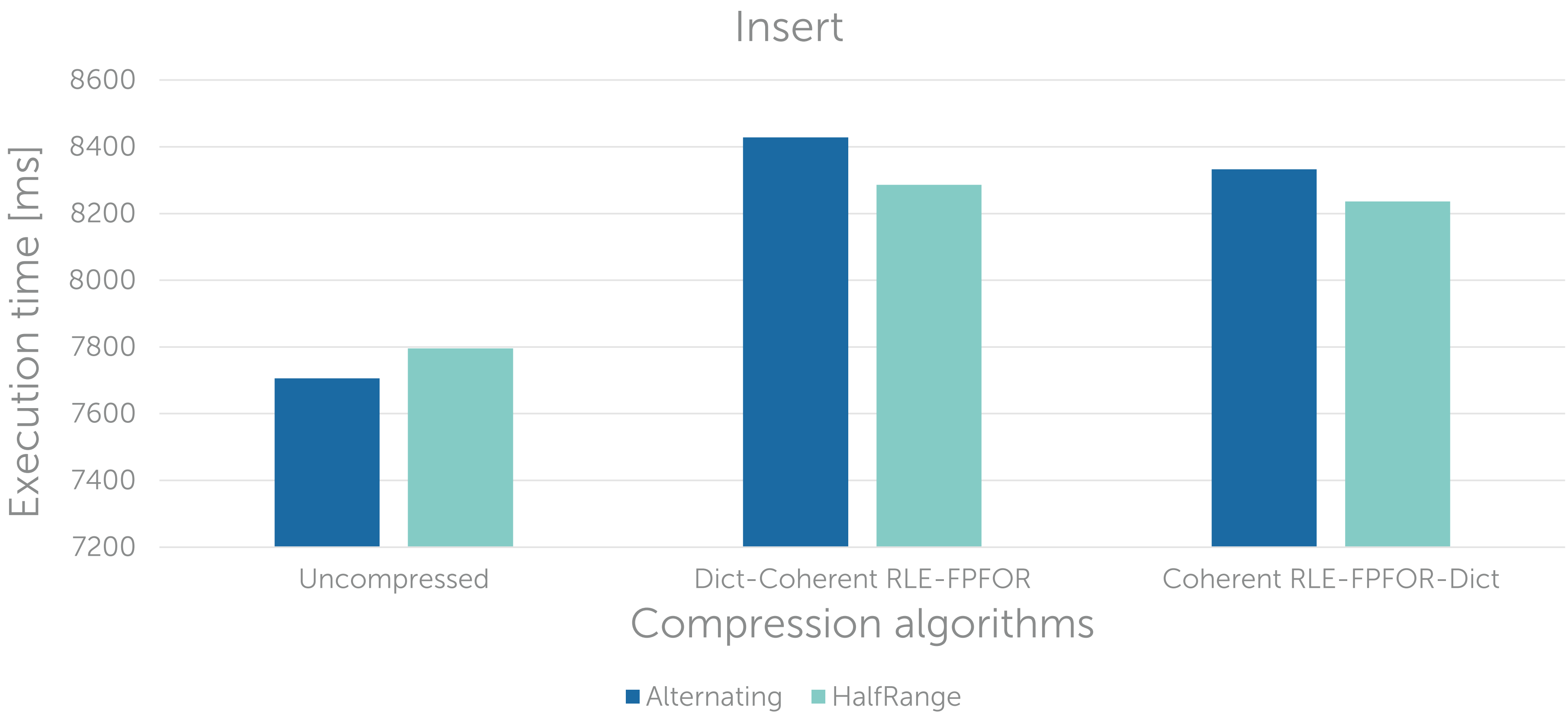


Backup - Evaluation

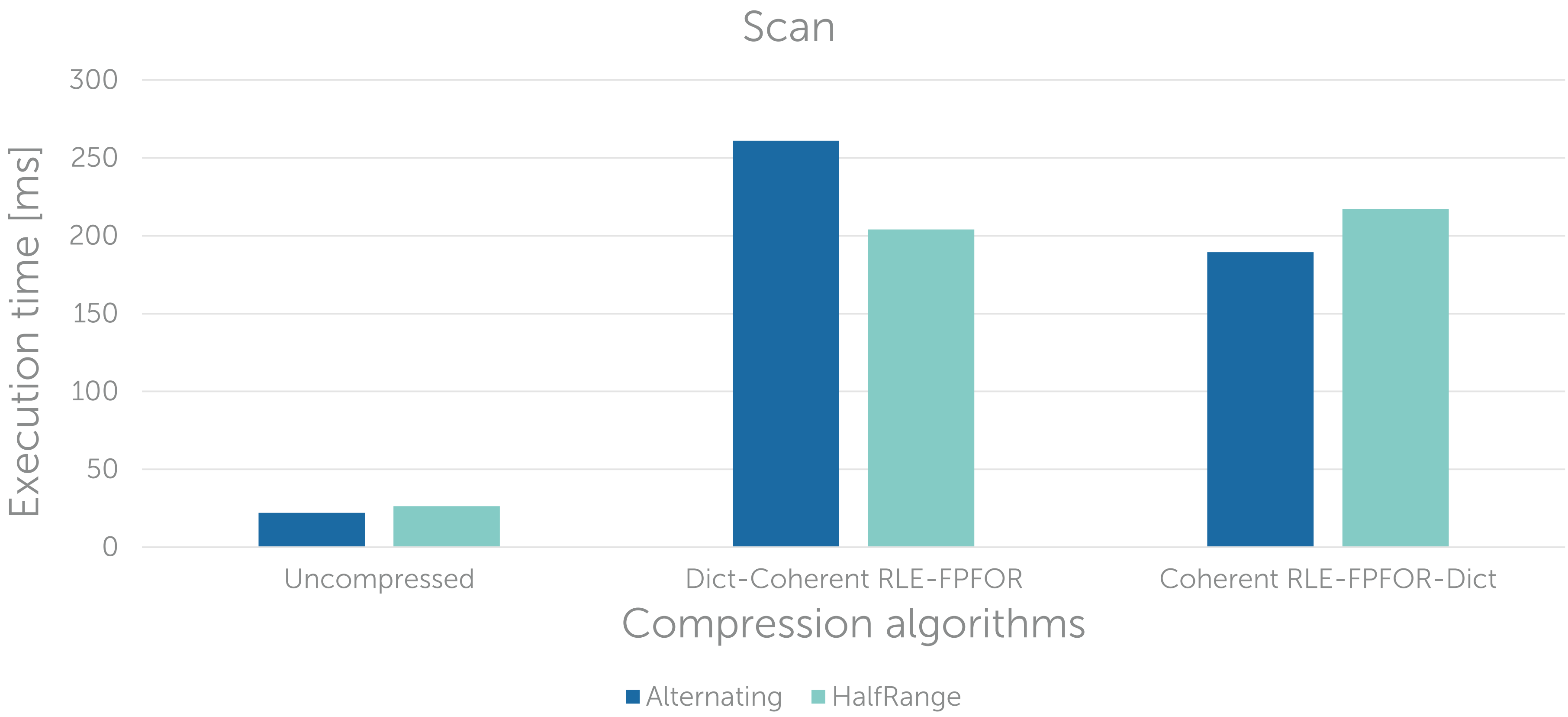


Backup - Evaluation

Dresden Database Systems Group



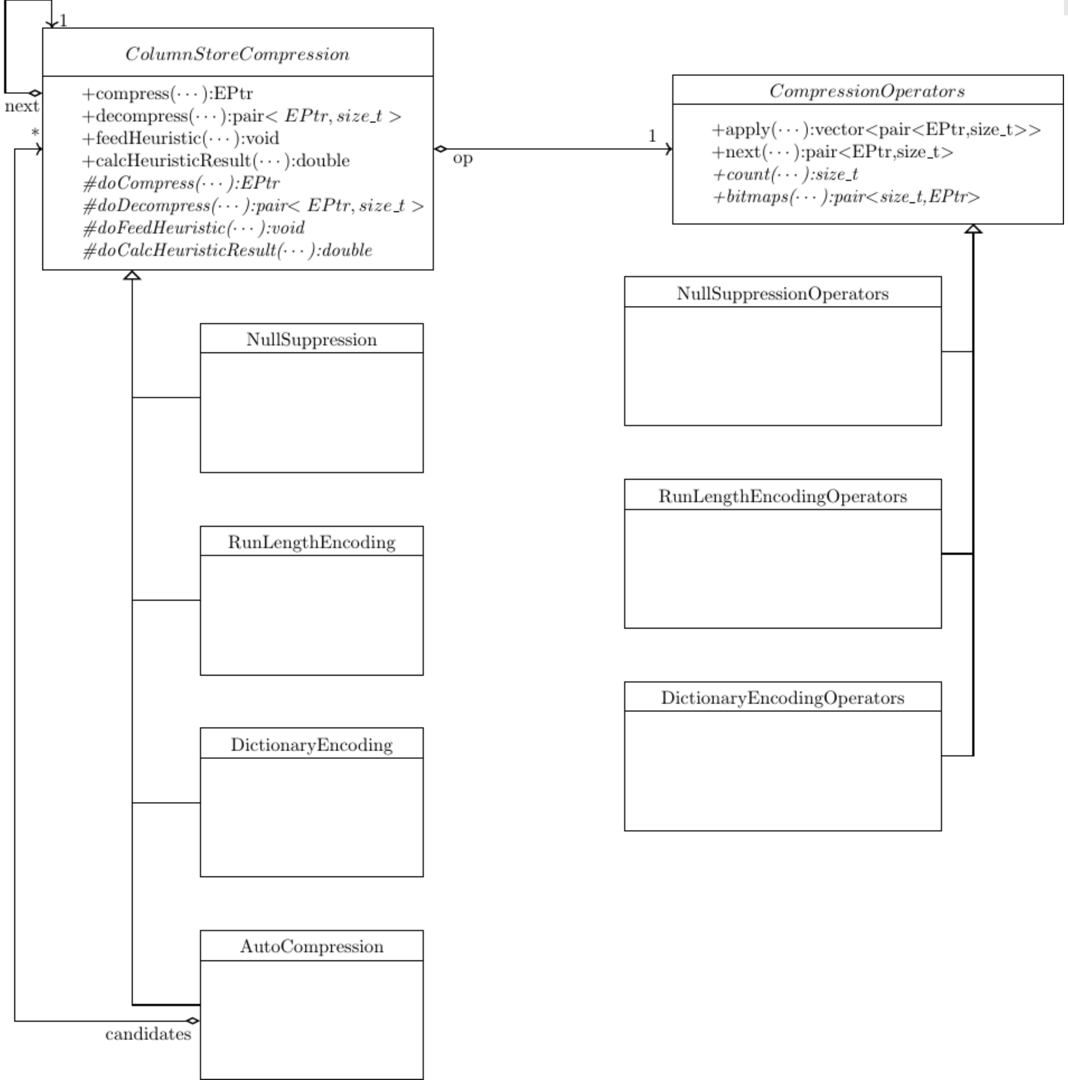
Backup - Evaluation



Adaptive compression selection - Part 2

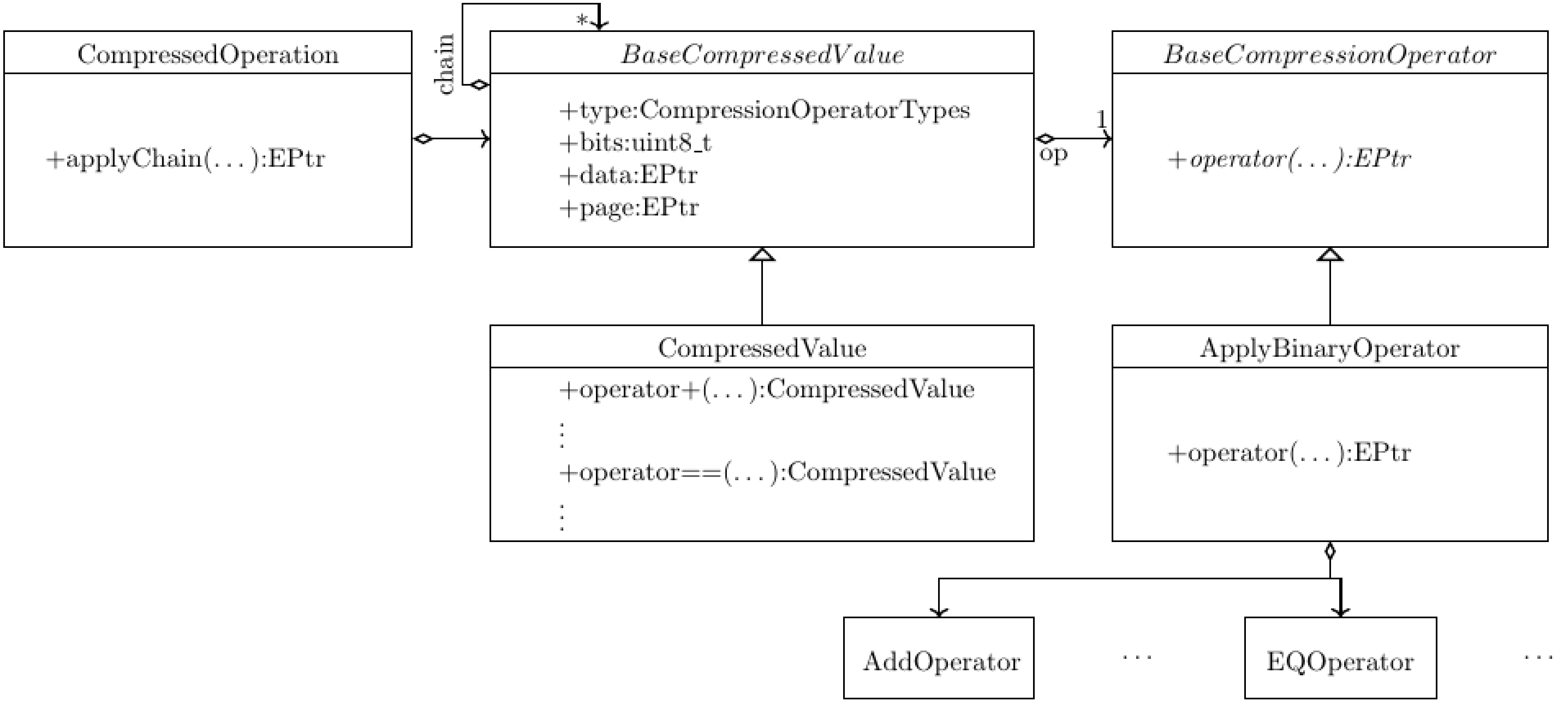
Run-Length-Encoding Heuristics h_{rle}	$$h_{rle} = \frac{2 * (valueChanges + 1)}{n}$$	A	B	B	B
		valueChange = 1			
Dictionary-Encoding Heuristics h_{dict}	$$h_{dict} = \frac{n * effectiveBytes + 4 * n_{dist}}{4 * n}$$	A	B	C	C
		$n_{dist} = 3$, *effectiveBytes = 1*			
SIMD-FastPFOR Heuristics h_{pfor}	$$h_{pfor} = \frac{\log_2(avgValue) + 1}{32}$$	0	128	1024	32
		avgValue = 296 $\log_2(avgValue) = 8{,}21$			

Dresden Database Systems Group

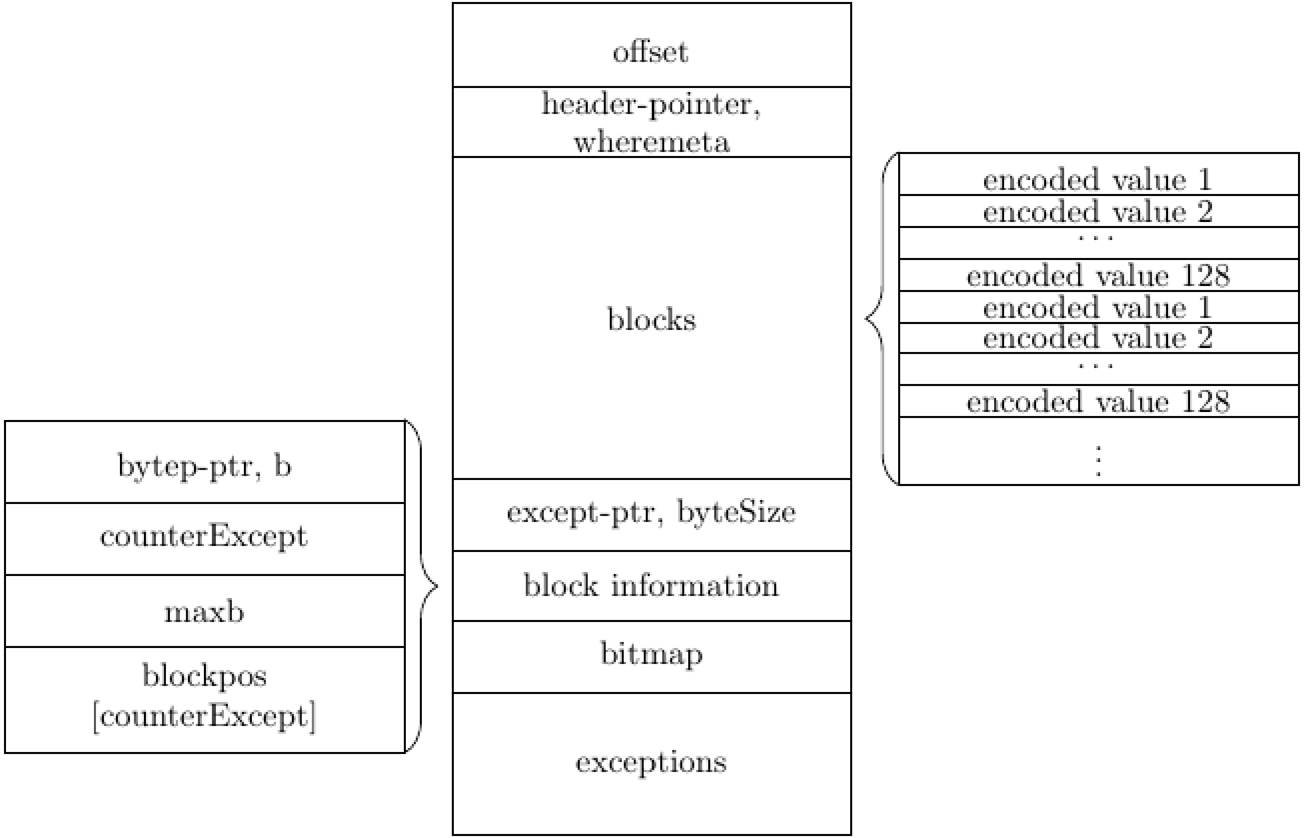


TECHNISCHE UNIVERSITÄT DRESDEN

Compression operators structure

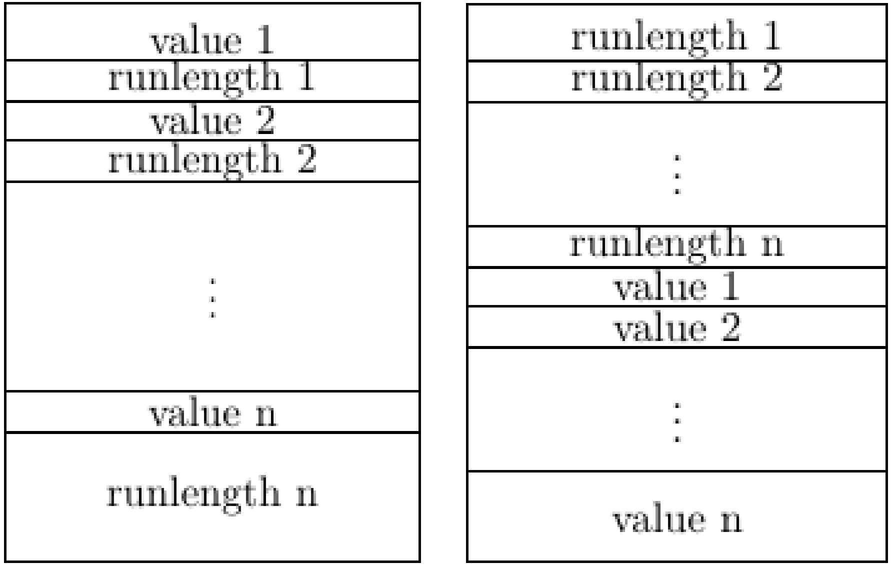


Structure of a FPFOR-Page

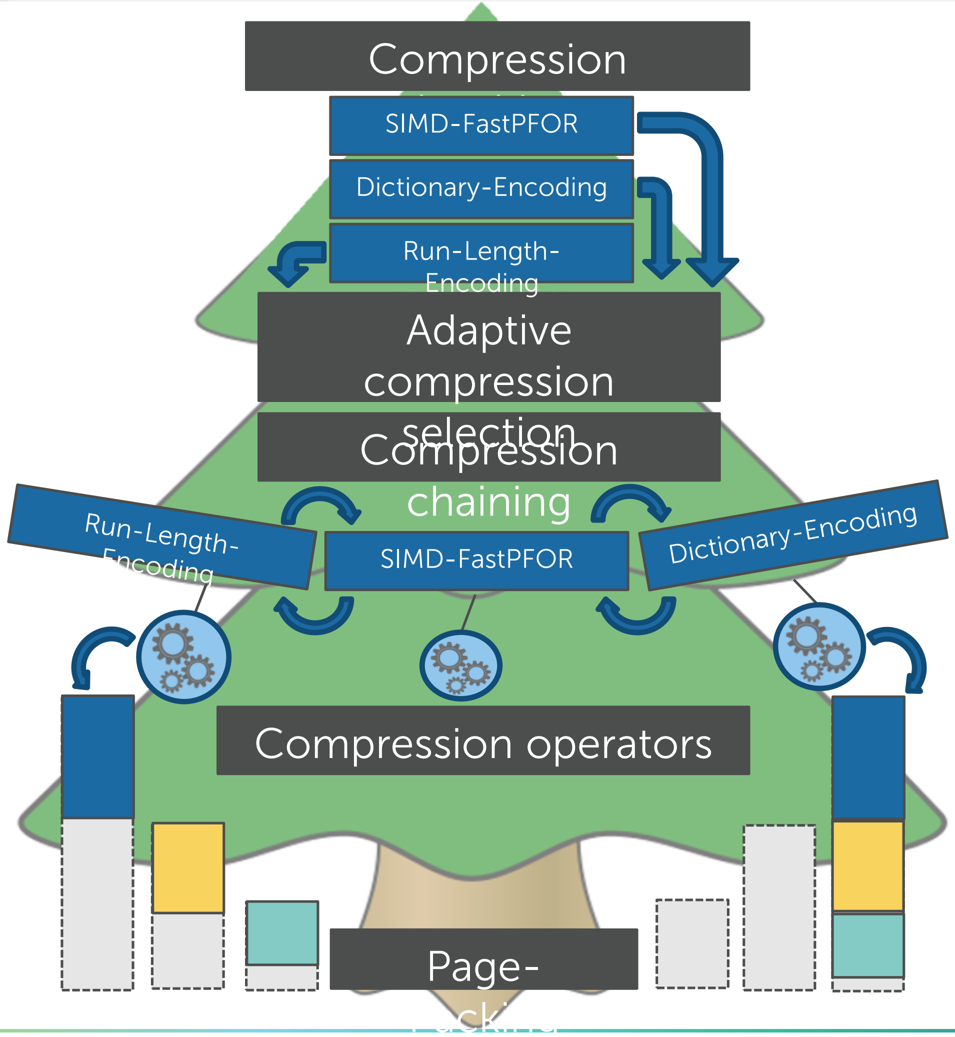


Structure of a RLE-Page

Dresden Database Systems Group



Overview



Dresden Database Systems Group