

Lightweight compression algorithms and interfaces for flexible storage architectures

Master thesis defense – Paul Peschel – 17.12.2015

Motivation - Big Data

Volume SCALE OF DATA

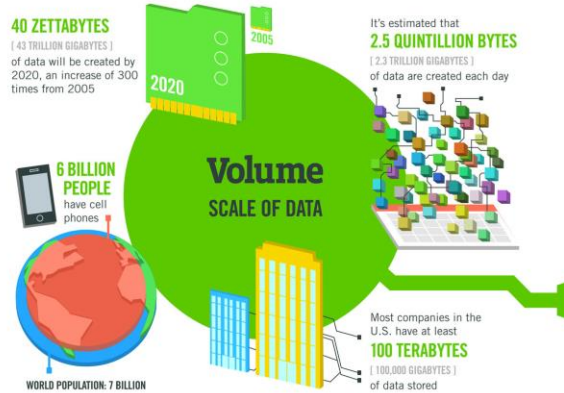
40 ZETTABYTES
[43 TRILLION GIGABYTES]
of data will be created by 2020, an increase of 300 times from 2005

6 BILLION PEOPLE have cell phones

WORLD POPULATION: 7 BILLION

It's estimated that **2.5 QUINTILLION BYTES** [2.3 TRILLION GIGABYTES] of data are created each day

Most companies in the U.S. have at least **100 TERABYTES** [100,000 GIGABYTES] of data stored



The FOUR V's of Big Data

From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can these massive amounts of data be used?

As a leader in the sector, IBM data scientists break big data into four dimensions: **Volume, Velocity, Variety and Veracity**

Depending on the industry and organization, big data encompasses information from multiple internal and external sources such as transactions, social media, enterprise content, sensors and mobile devices. Companies can leverage data to adapt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.

By 2015, **4.4 MILLION IT JOBS** will be created globally to support big data, with 1.9 million in the United States.



Variety DIFFERENT FORMS OF DATA

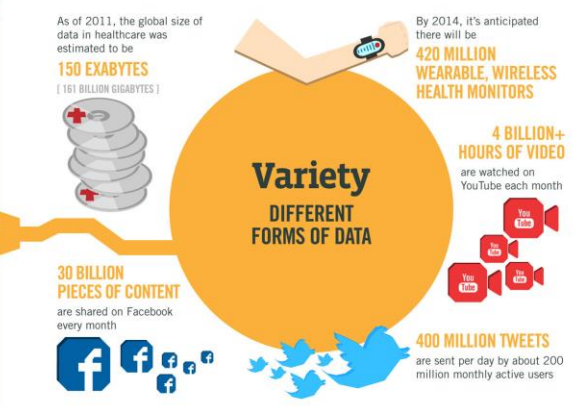
As of 2011, the global size of data in healthcare was estimated to be **150 EXABYTES** [161 BILLION GIGABYTES]

By 2014, it's anticipated there will be **420 MILLION WEARABLE, WIRELESS HEALTH MONITORS**

4 BILLION+ HOURS OF VIDEO are watched on YouTube each month

30 BILLION PIECES OF CONTENT are shared on Facebook every month

400 MILLION TWEETS are sent per day by about 200 million monthly active users

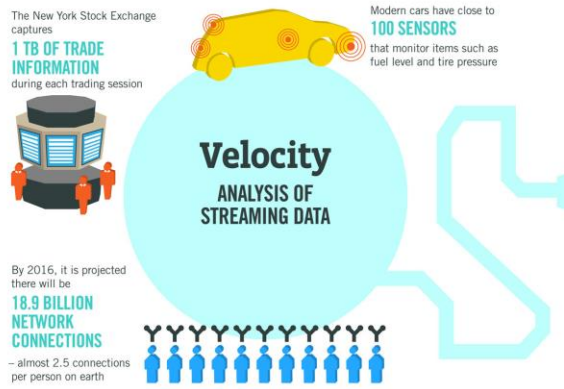


Velocity ANALYSIS OF STREAMING DATA

The New York Stock Exchange captures **1 TB OF TRADE INFORMATION** during each trading session

Modern cars have close to **100 SENSORS** that monitor items such as fuel level and tire pressure

By 2016, it is projected there will be **18.9 BILLION NETWORK CONNECTIONS** - almost 2.5 connections per person on earth

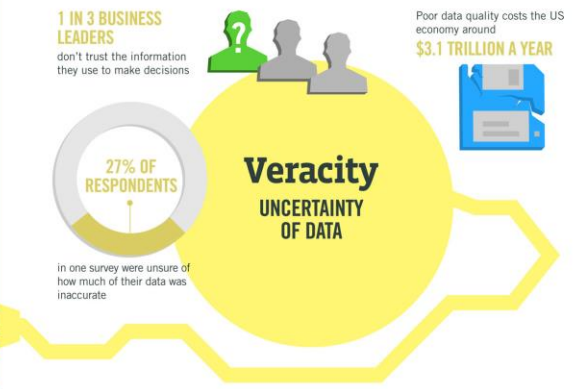


Veracity UNCERTAINTY OF DATA

1 IN 3 BUSINESS LEADERS don't trust the information they use to make decisions

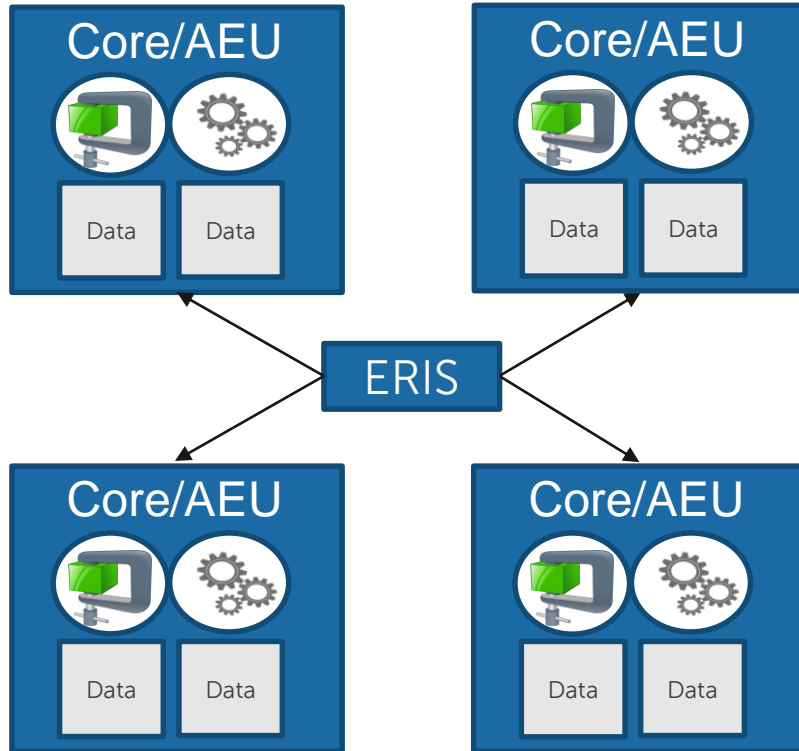
Poor data quality costs the US economy around **\$3.1 TRILLION A YEAR**

27% OF RESPONDENTS in one survey were unsure of how much of their data was inaccurate



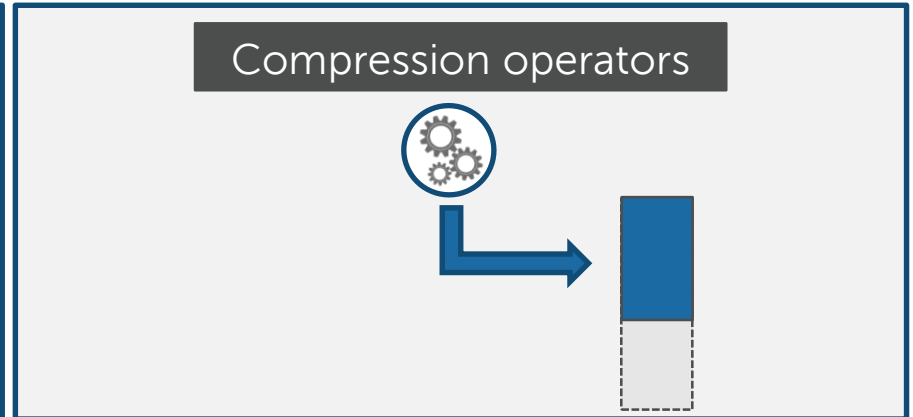
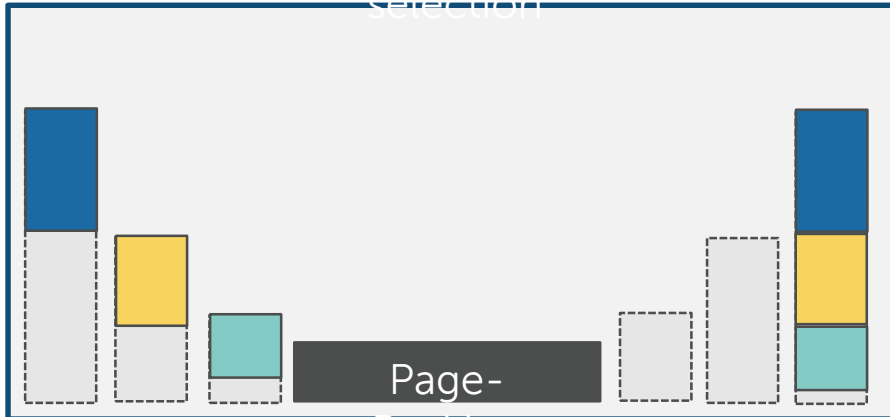
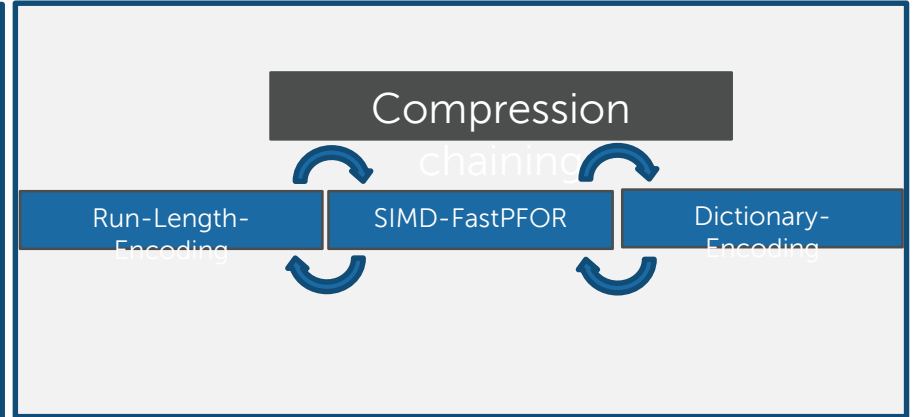
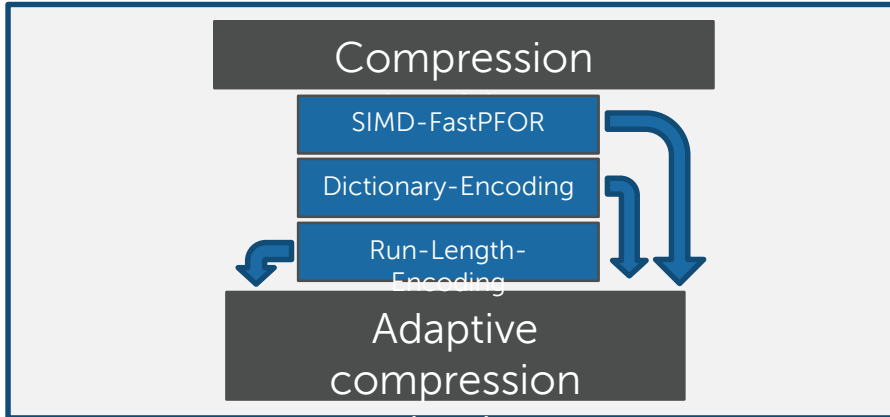
Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, QAS

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



Alternating

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

42 128 ... 42 128

HalfRange

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

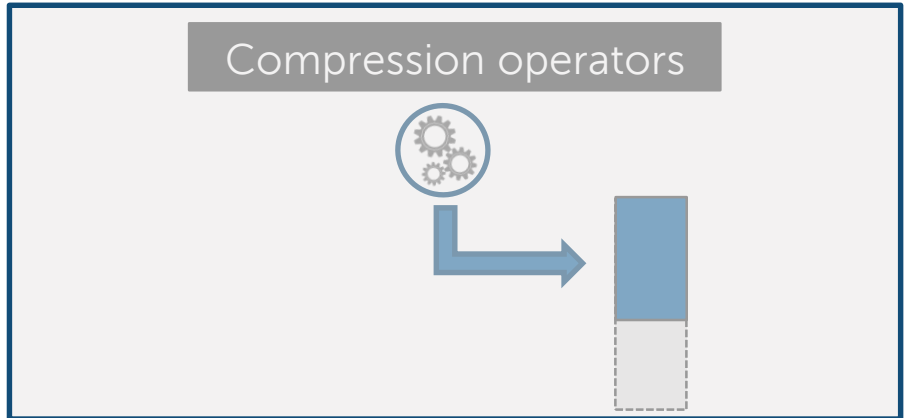
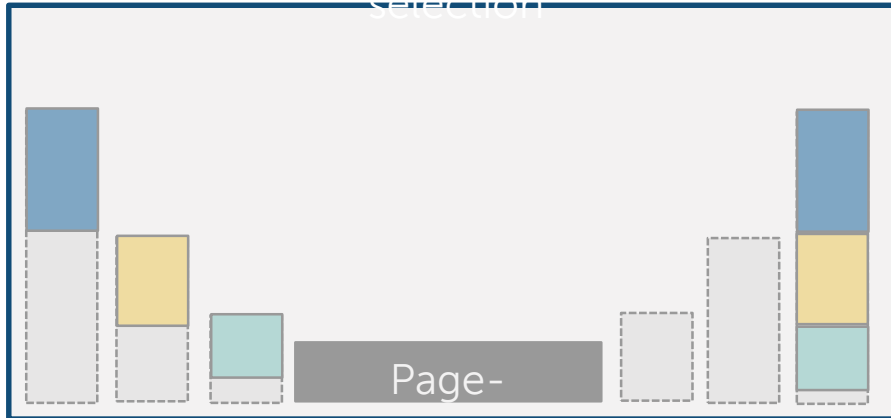
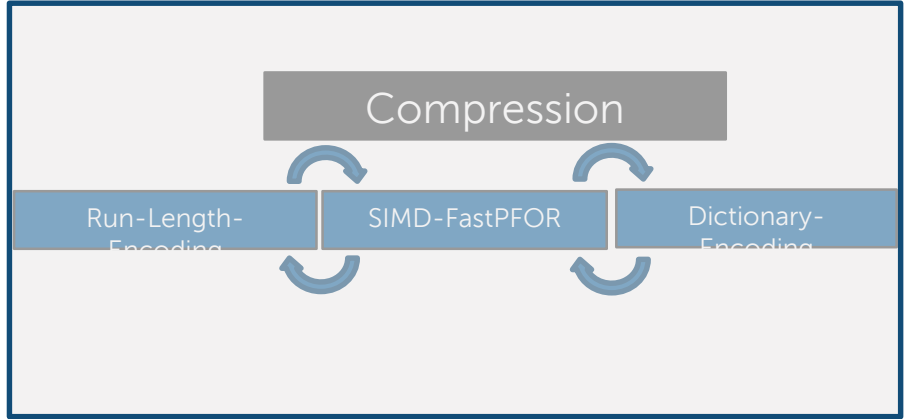
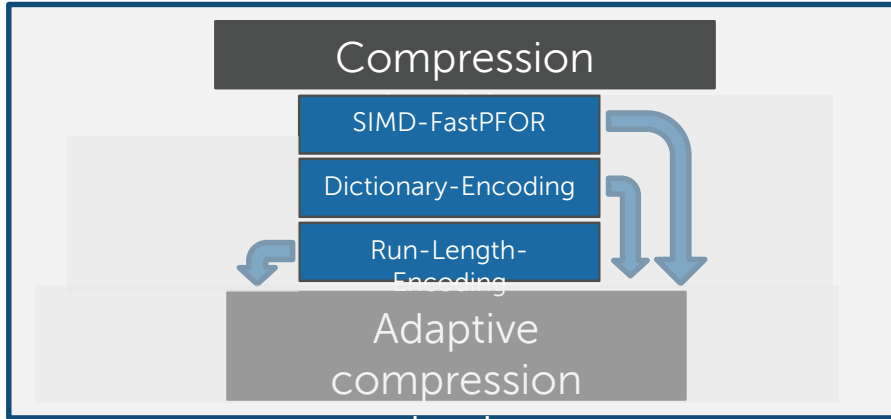
42 ... 42 128 ... 128

Ordered

- A_1, A_2, \dots, A_n

1 2 ...

Overview



Compression Interface

Run-
Length-
Encoding

Dictionary-Encoding

SIMD-FastPFOR

Encodes runs of equal values to (value;run-length)-pairs; Coherent RLE: value and runlength sections separated

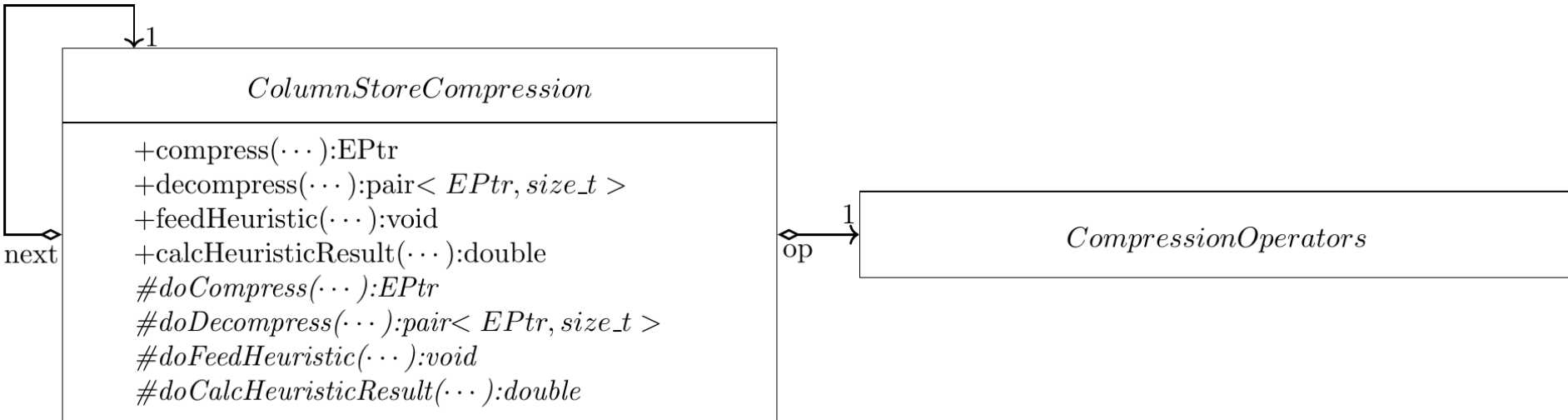
Creates dictionary of distinct values

Replaces values by dictionary index

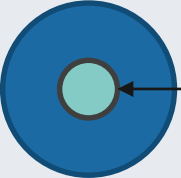
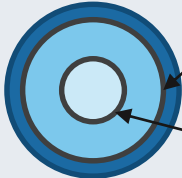
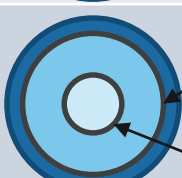
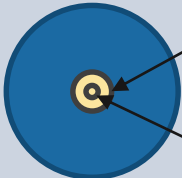
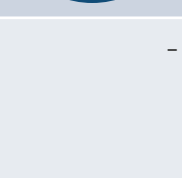
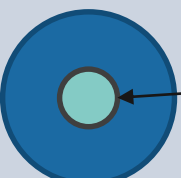
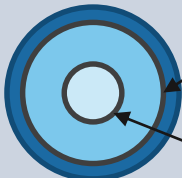
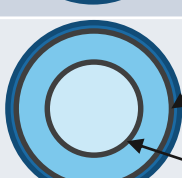
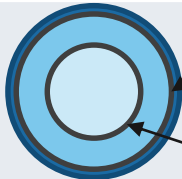

Calculates bit width b using a cost function

Removes bit widths greater b from values and saves the removed bit widths as exceptions

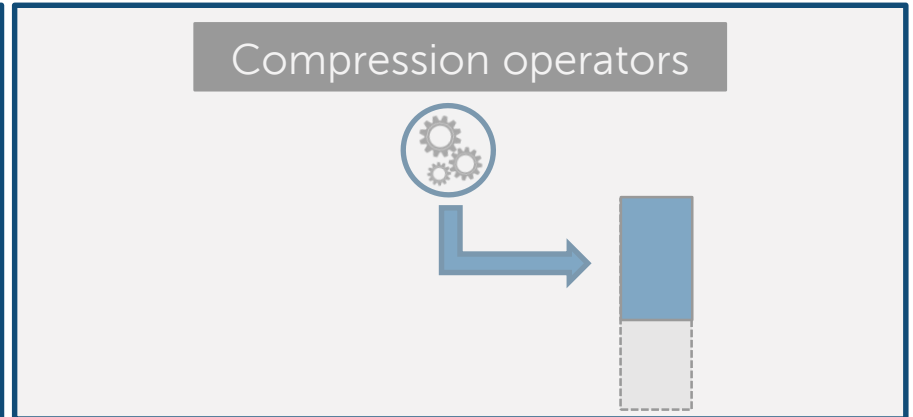
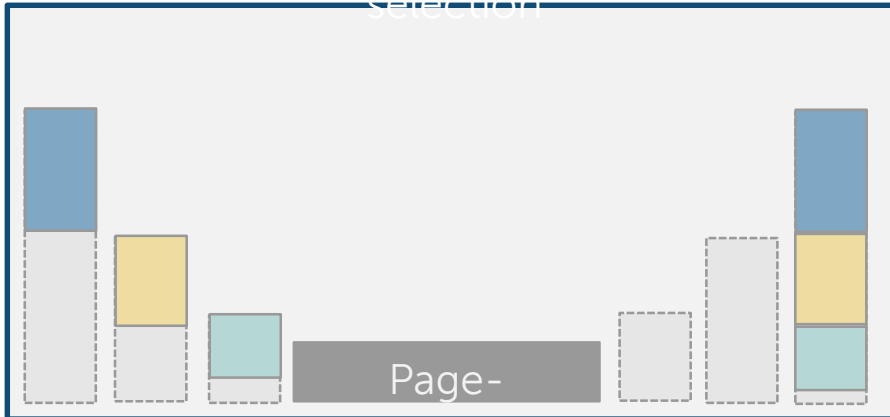
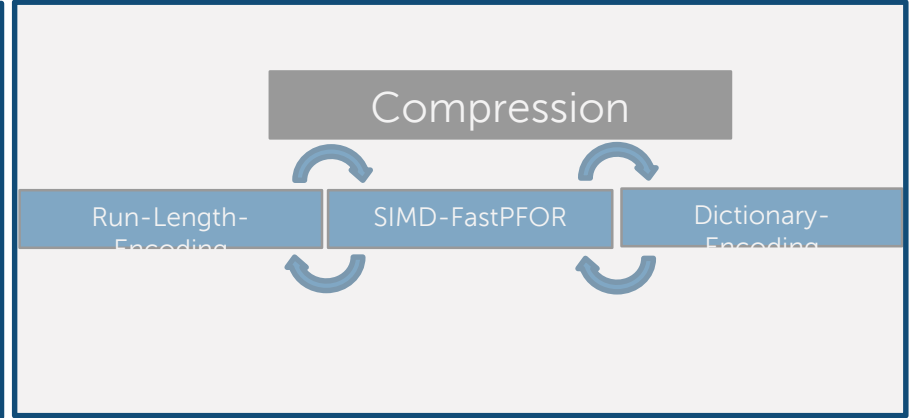
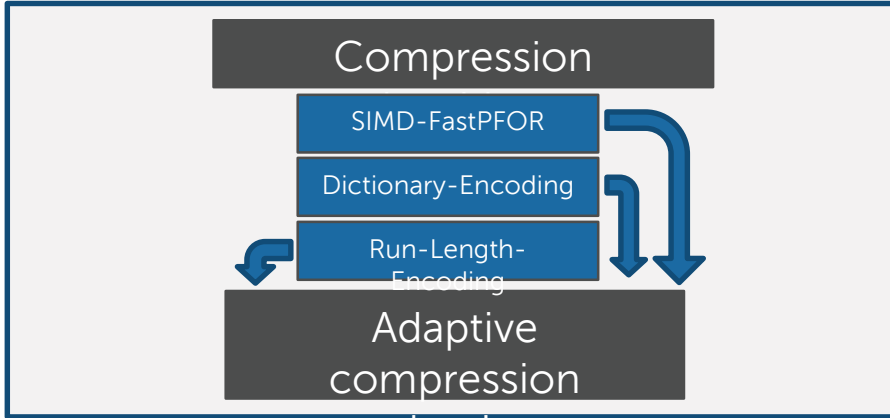
Compression interface



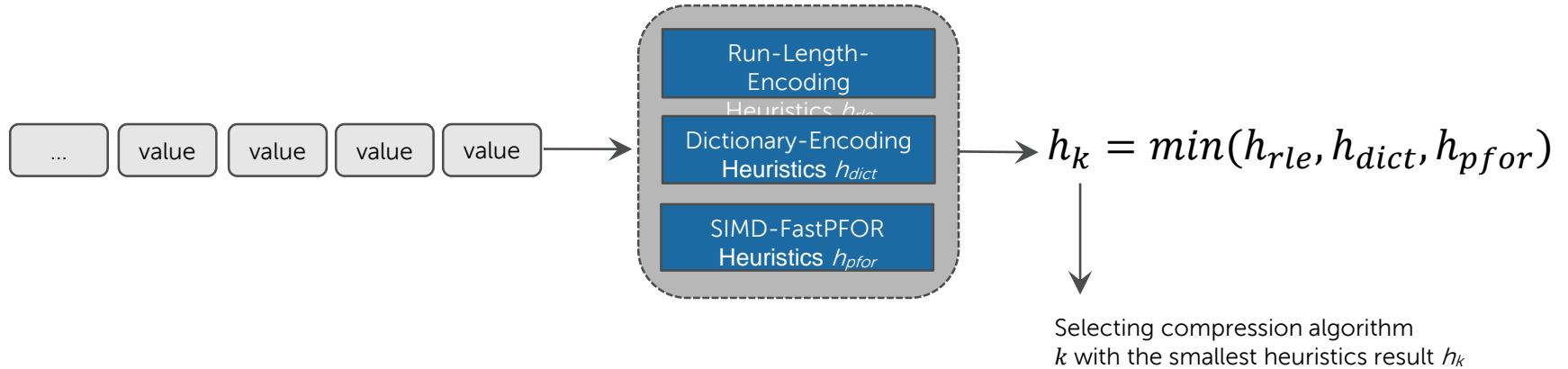
Compression algorithms – compression rates

Compression rates			
	Run-Length-Encoding	Dictionary-Encoding	SIMD-FastPFOR
Alternating $A_1, B_2, \dots, A_{n-1}, B_n$	-	 25%	 83%  27%
HalfRange $A_1, A_2, \dots, A_{n/2}, B_{n/2+1}, \dots, B_n$	 2.7%  0.1%	 25%	 83%  25%
Ordered A_1, A_2, \dots, A_n	-	-	 92%  50%

Overview



Adaptive compression selection



- 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

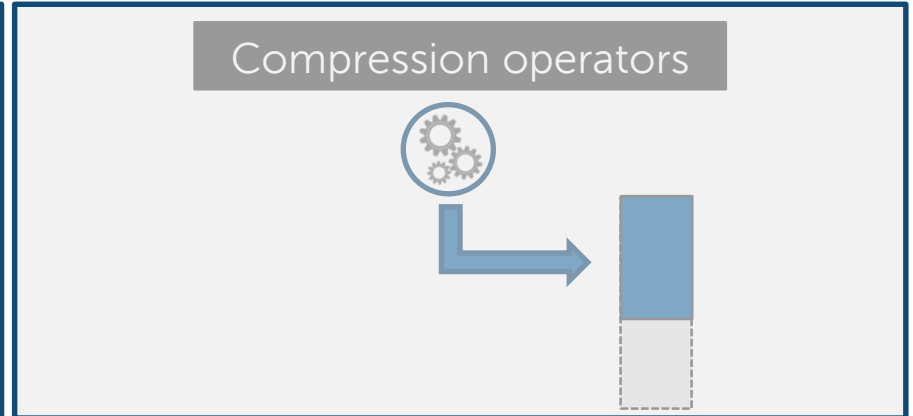
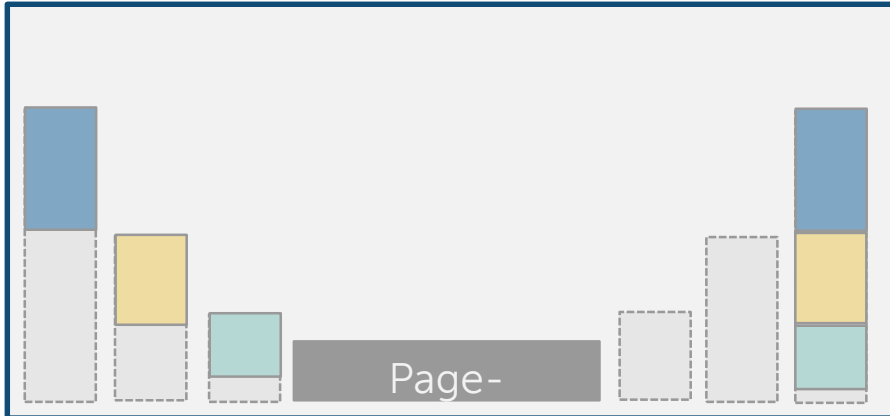
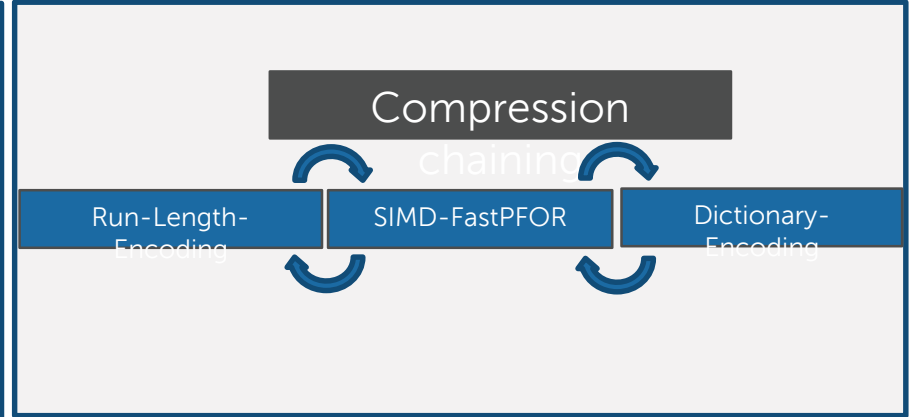
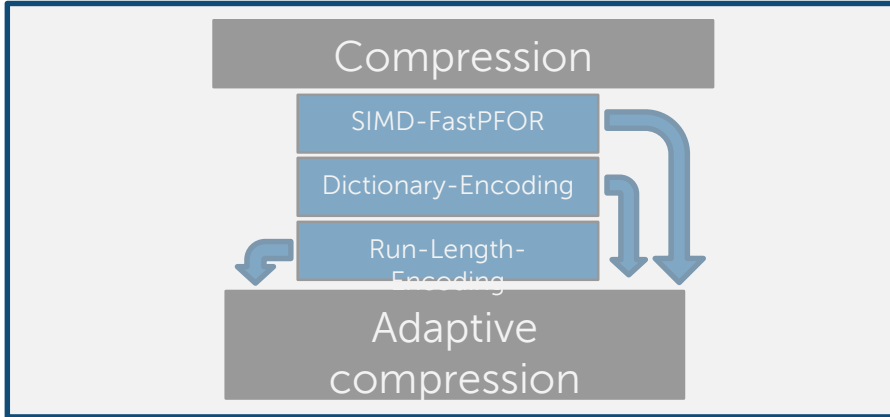
$valueChange = 1 \quad n = 4$

Heuristics results

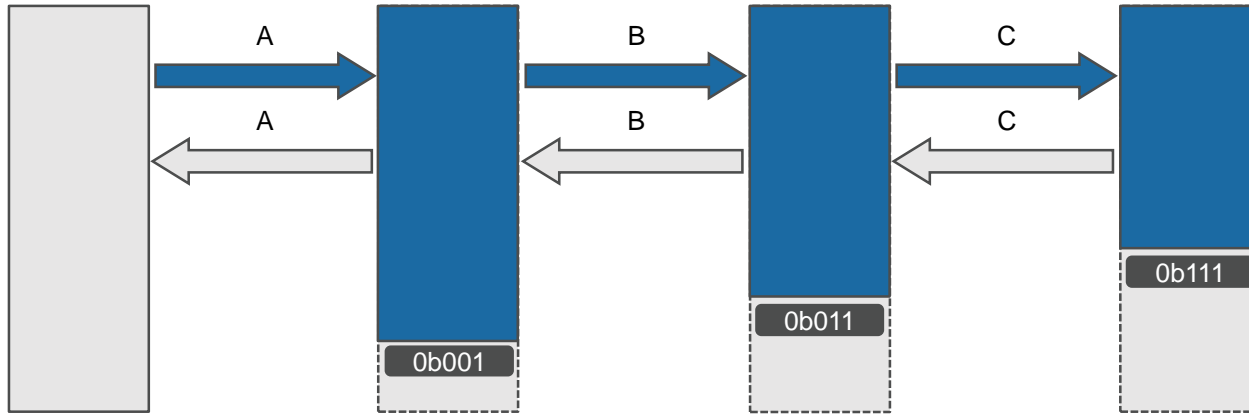
Heuristics results

	FastPFOR		RLE	Dictionary		Selected
Alternating $A_1, B_2, \dots, A_{n-1}, B_n$	27% 	22% 	200%	25% 	25% 	FastPFOR (better would be Dictionary-Encoding)
HalfRange $A_1, A_2, \dots, A_{n/2}, B_{n/2+1}, \dots, B_n$	25% 	24% 	0.1%  <0.01% 	25% 	25% 	RLE (RLE is the best selection)
Ordered A_1, A_2, \dots, 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



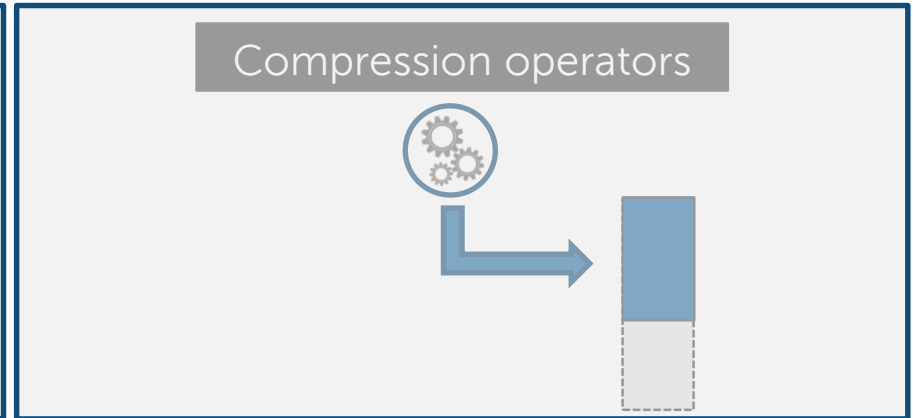
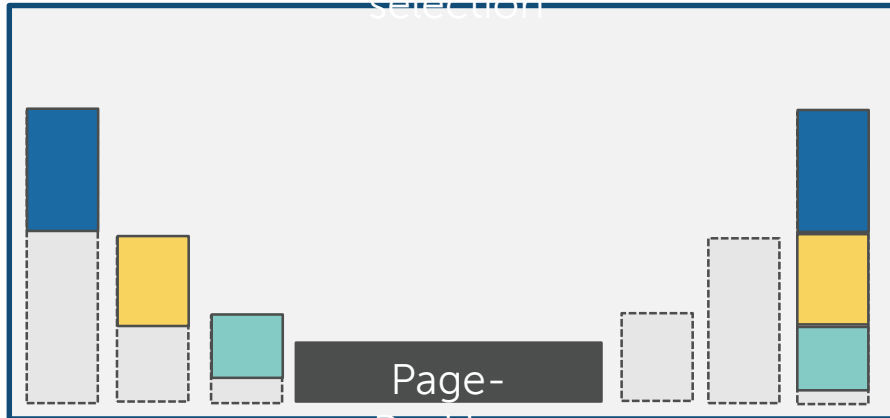
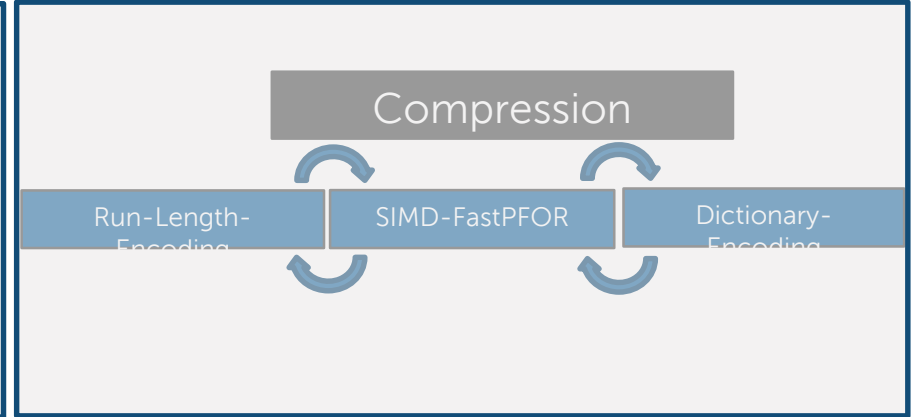
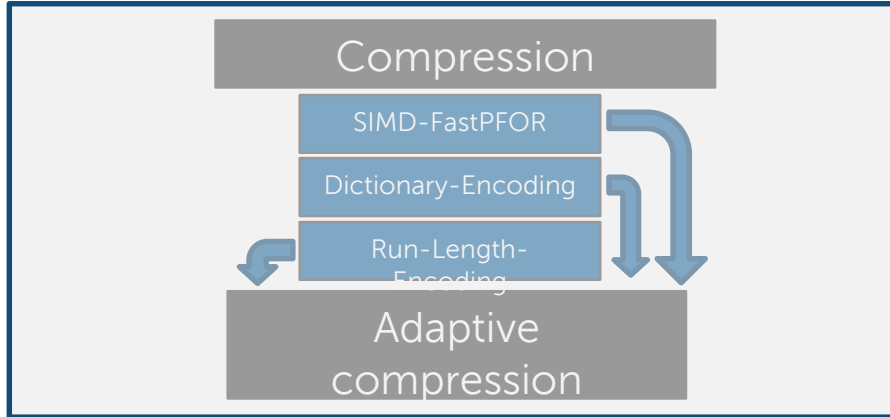
Compression chaining



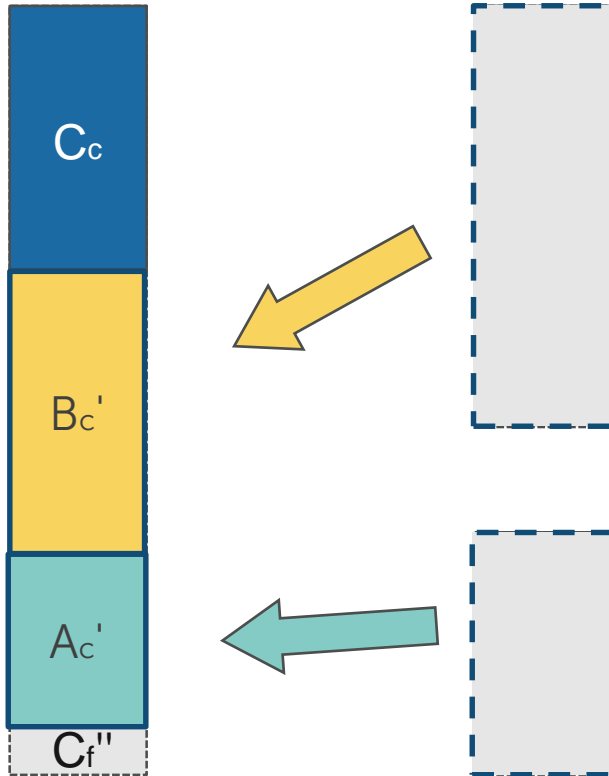
Compression rates

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

Overview



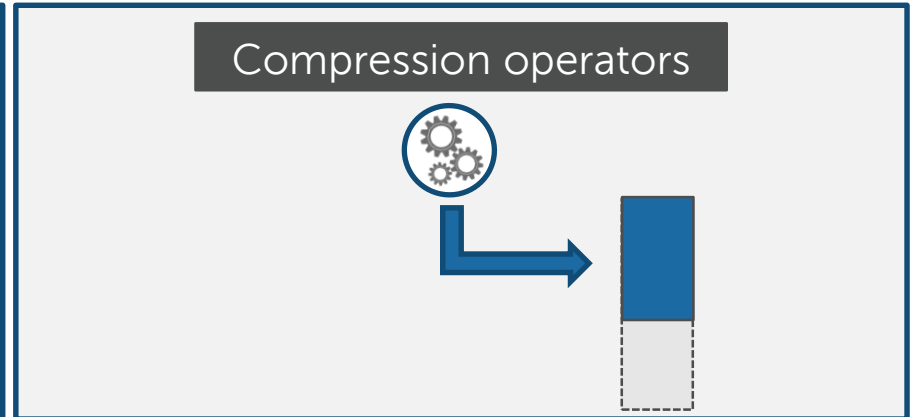
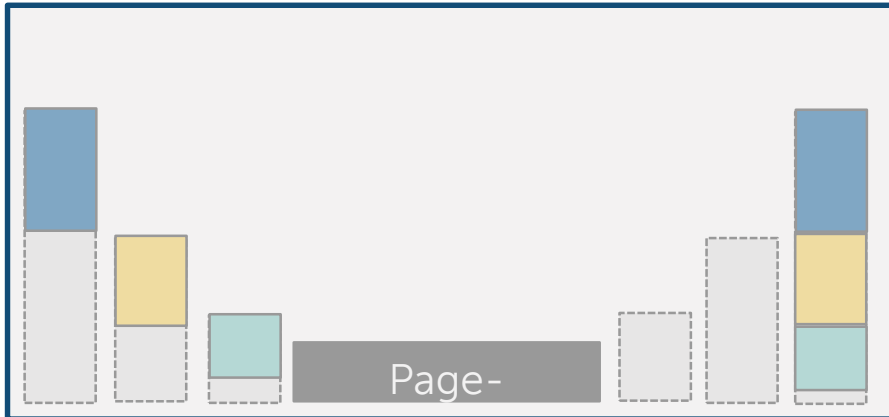
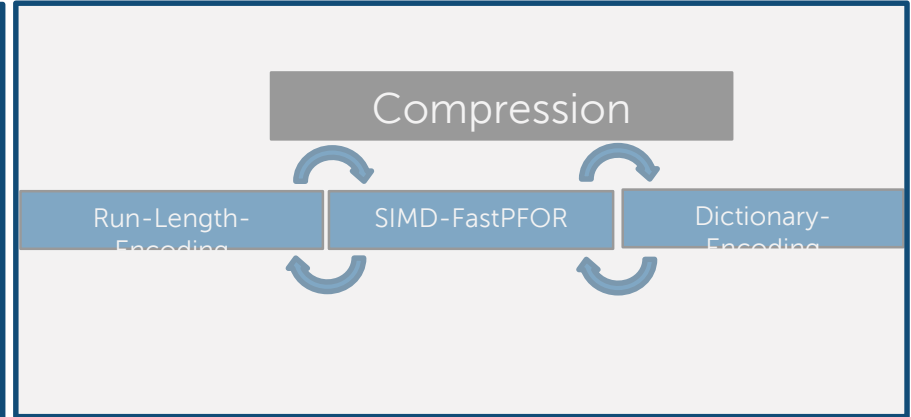
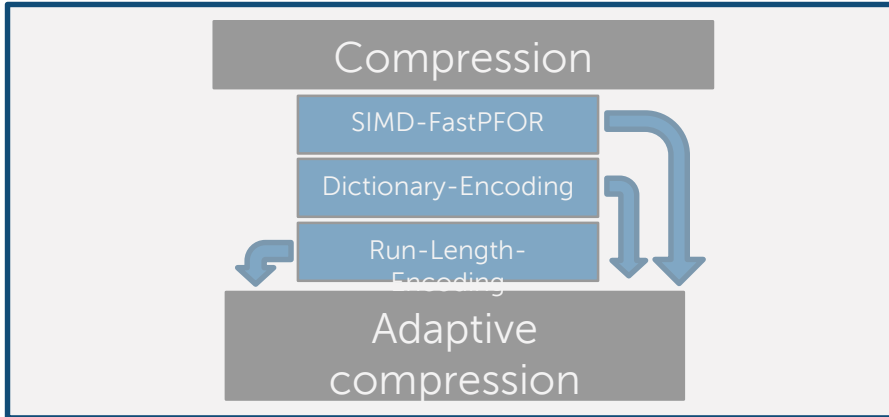
Page-Packing



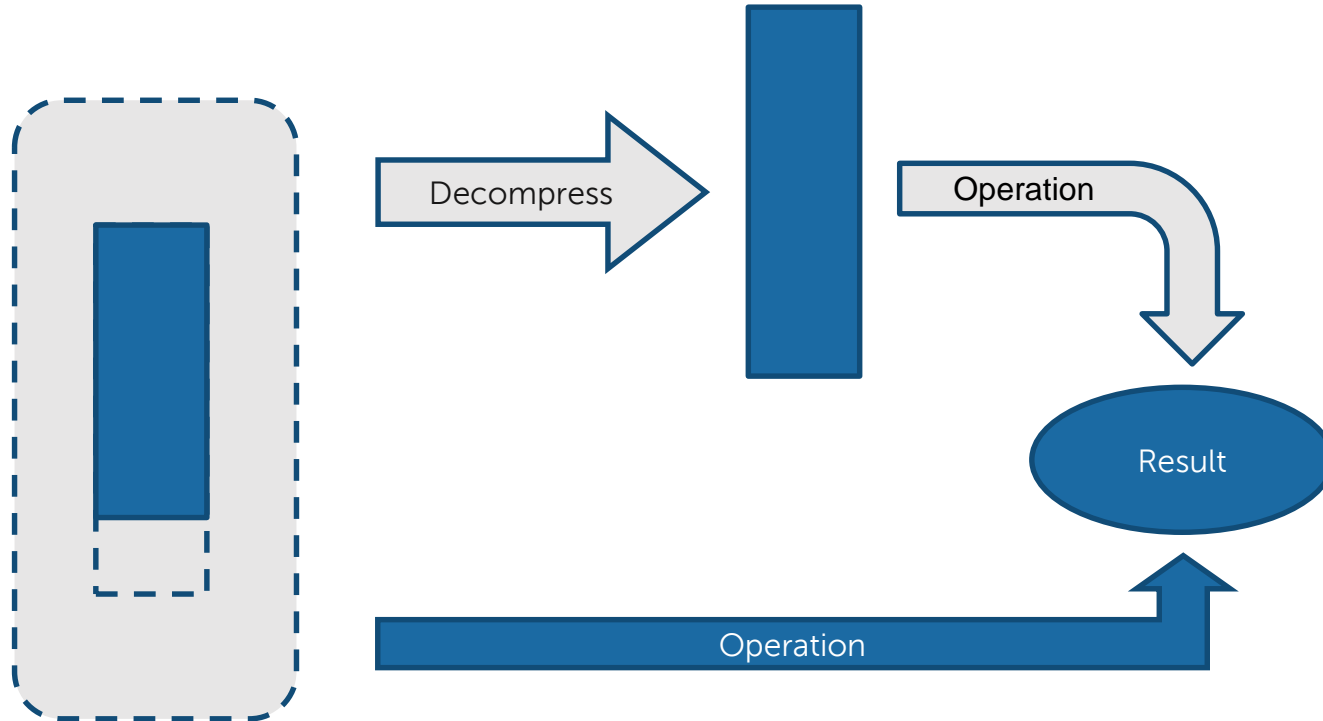
free	pos	originalPage
0	A_f	A
0	B_f	B
1581	C_f''	C

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

Overview



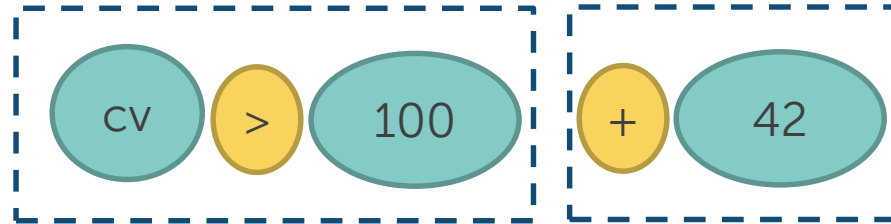
Compression operators - Motivation



Compression operators – Construction

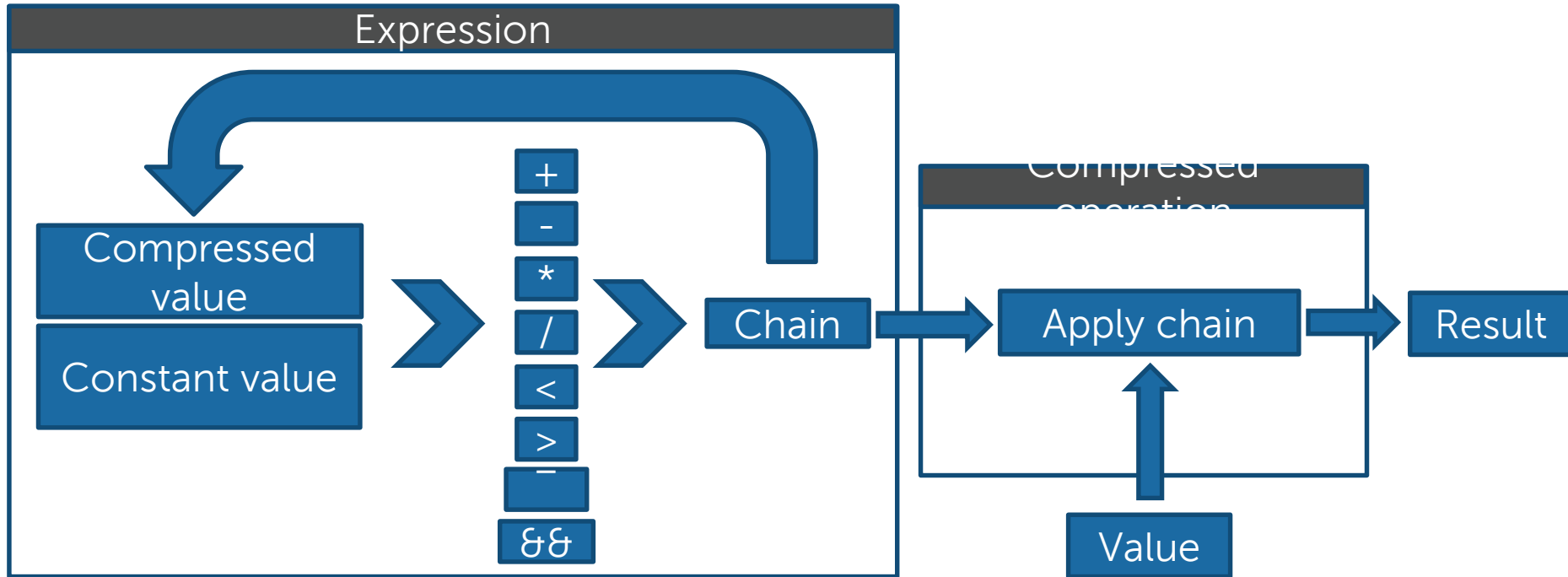
1

$$(x > 100) + 42$$

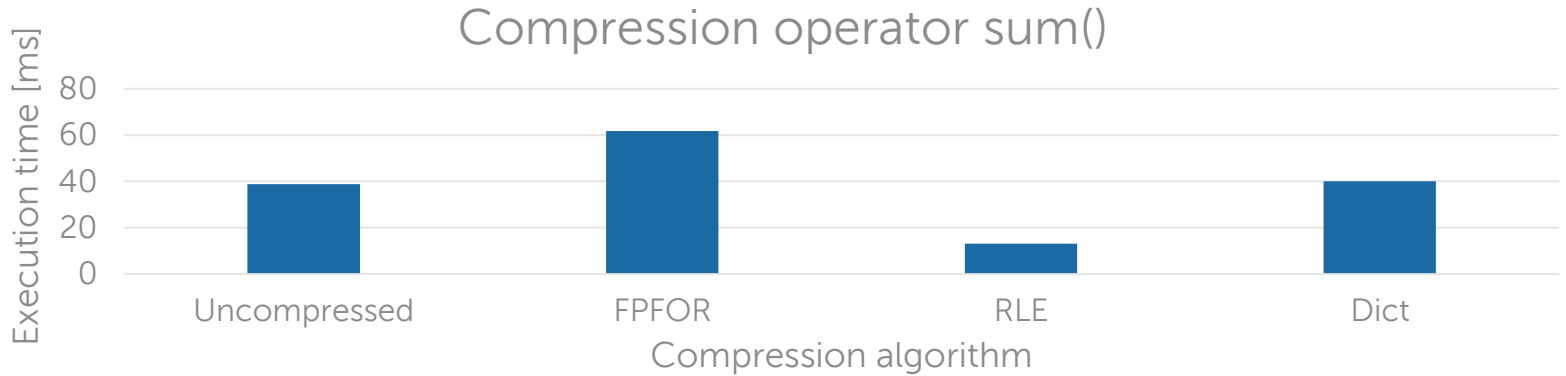
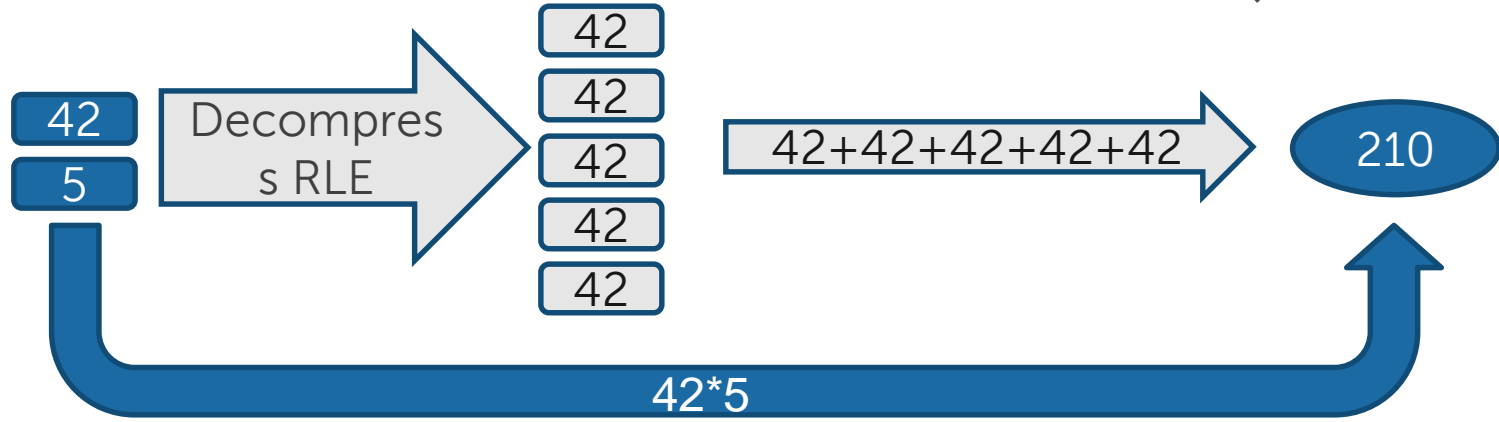


Compression operators – Construction

2

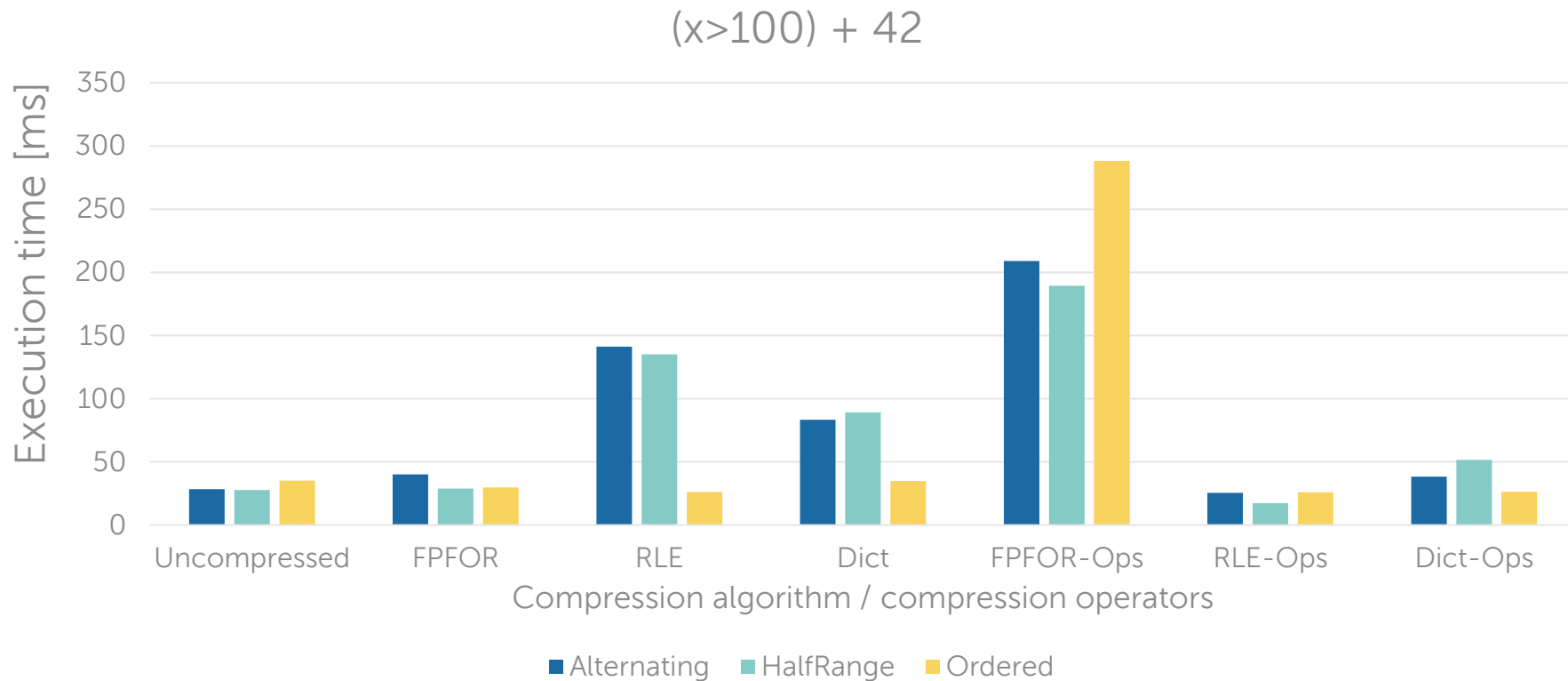


Compression operators – sum()



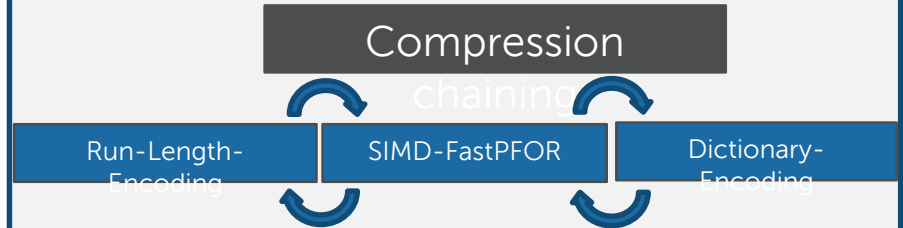
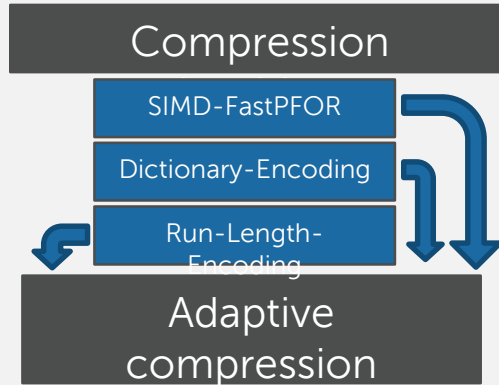
■ HalfRange

Compression operators evaluation



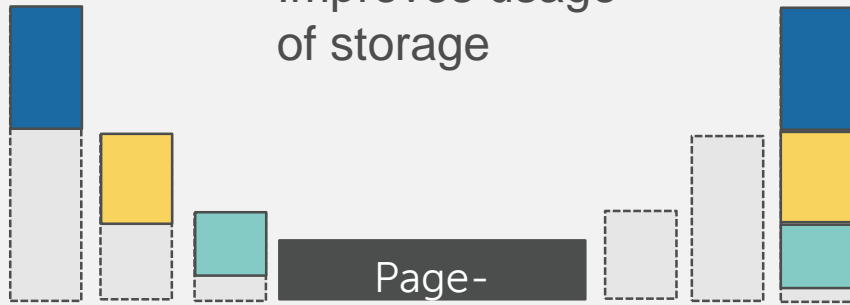
Conclusions

- Easy integration
- Selects appropriate compression algorithm



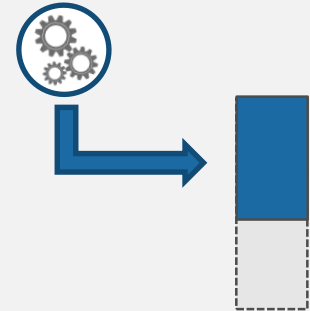
- Improves compression rates

- Improves usage of storage

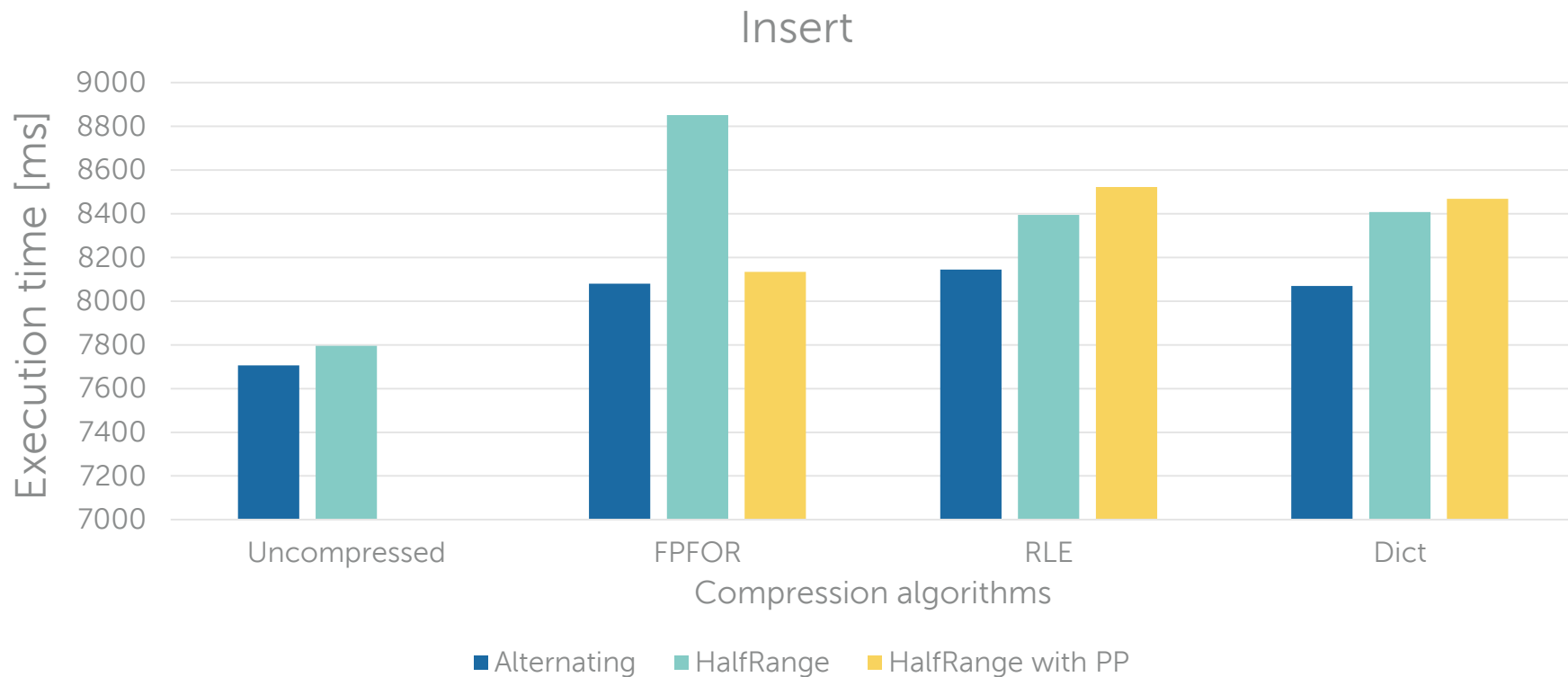


Compression operators

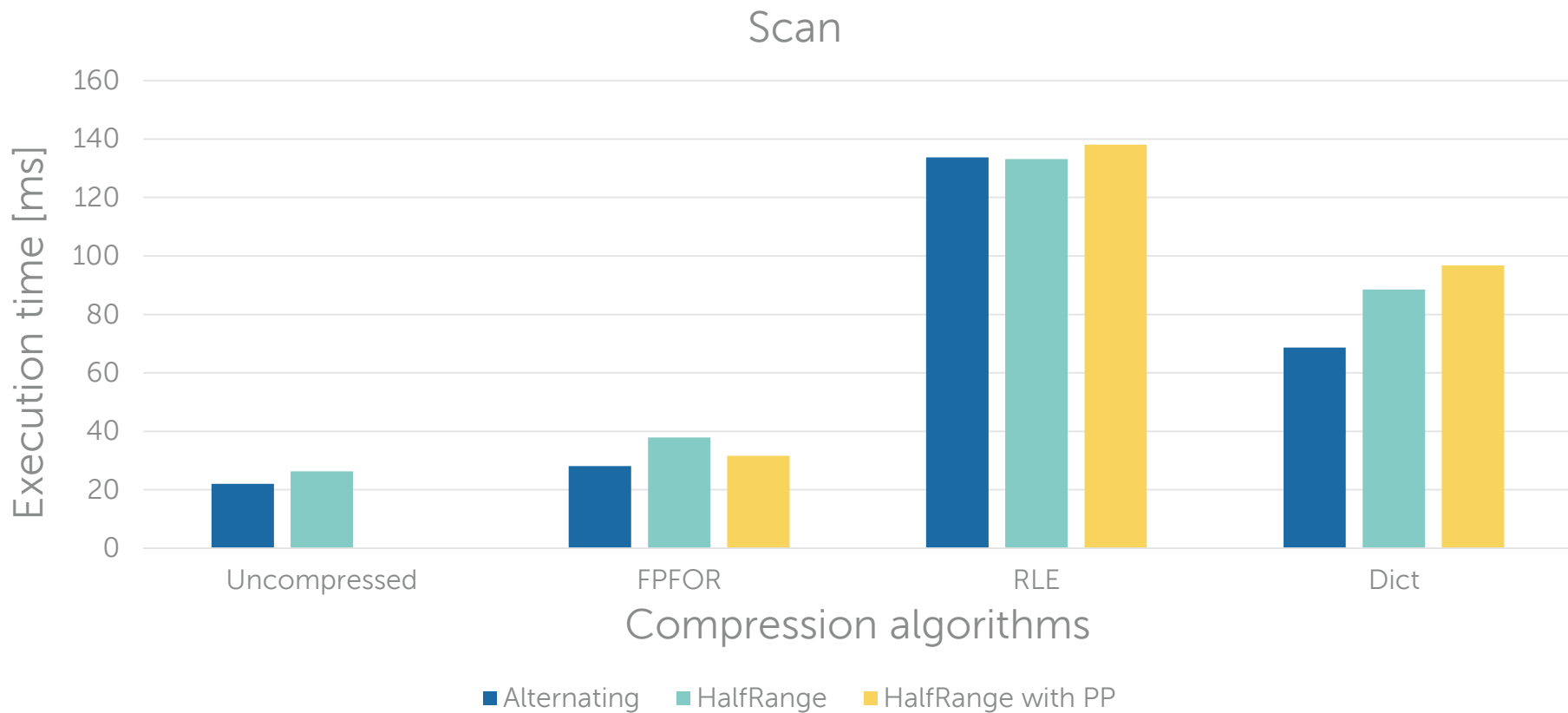
- Faster analytical operations with RLE and Dict
- But not with SIMD-FastPFOR



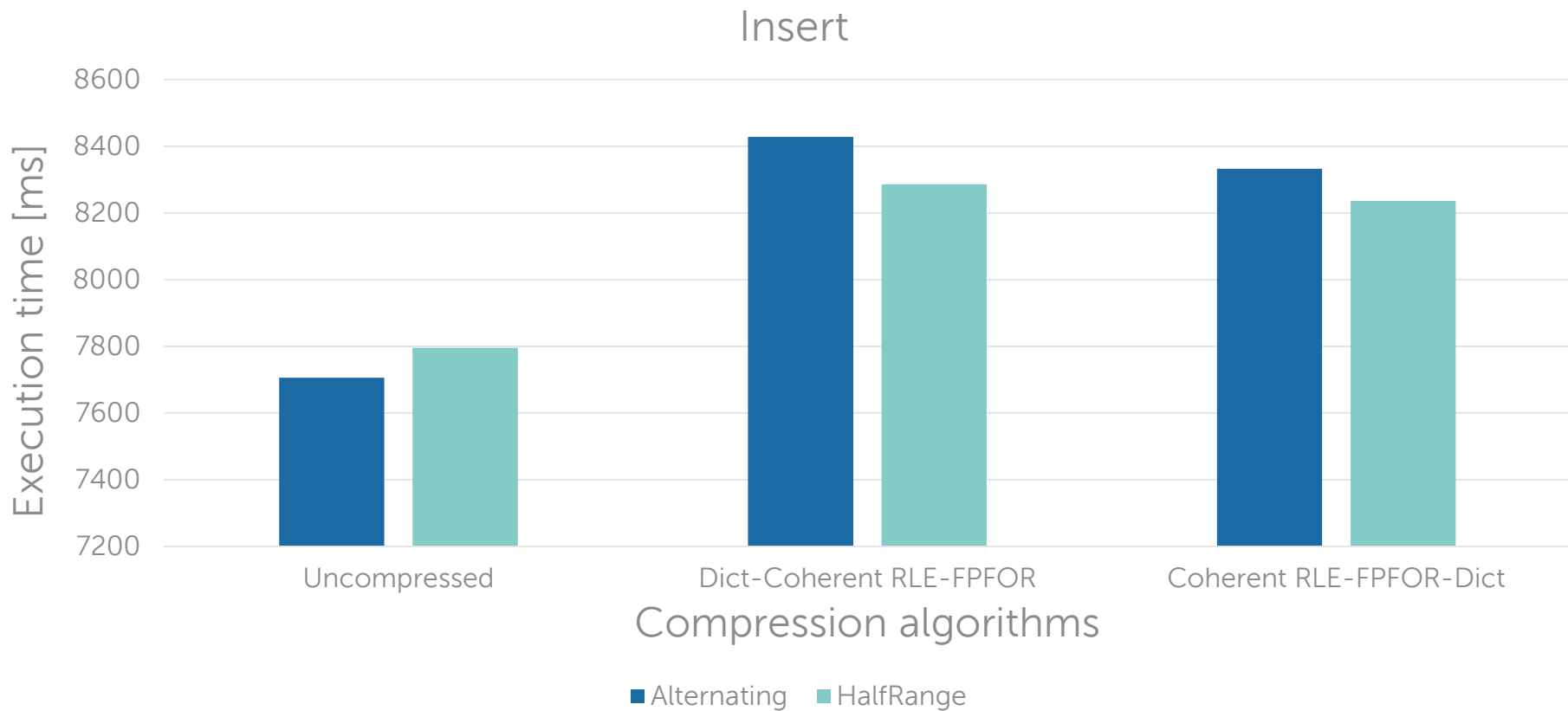
Backup - Evaluation



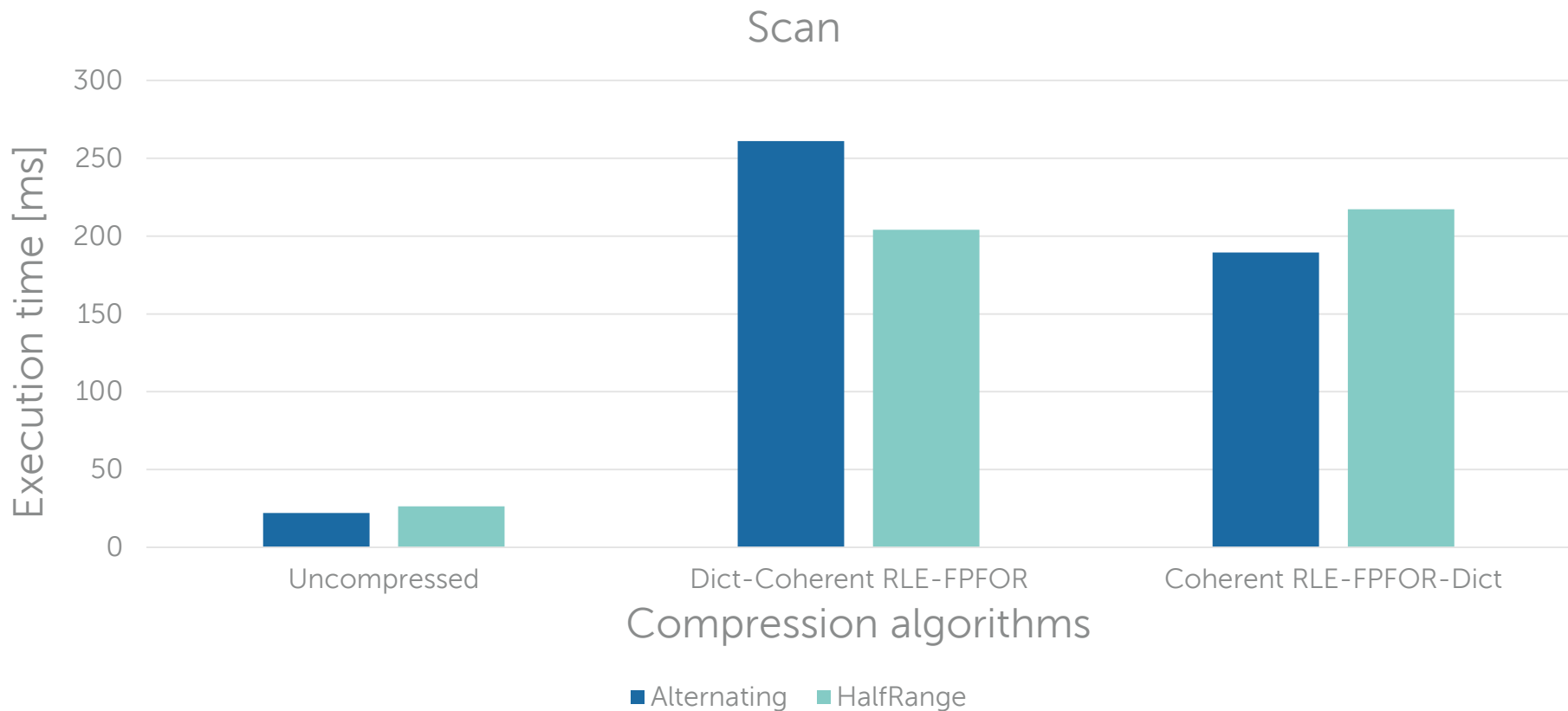
Backup - Evaluation



Backup - Evaluation



Backup - Evaluation



Adaptive compression selection - Part 2

Run-Length-Encoding
Heuristics h_{rle}

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



$valueChange = 1$

Dictionary-Encoding
Heuristics h_{dict}

$$h_{dict} = \frac{n * effectiveBytes + 4 * n_{dist}}{4 * n}$$



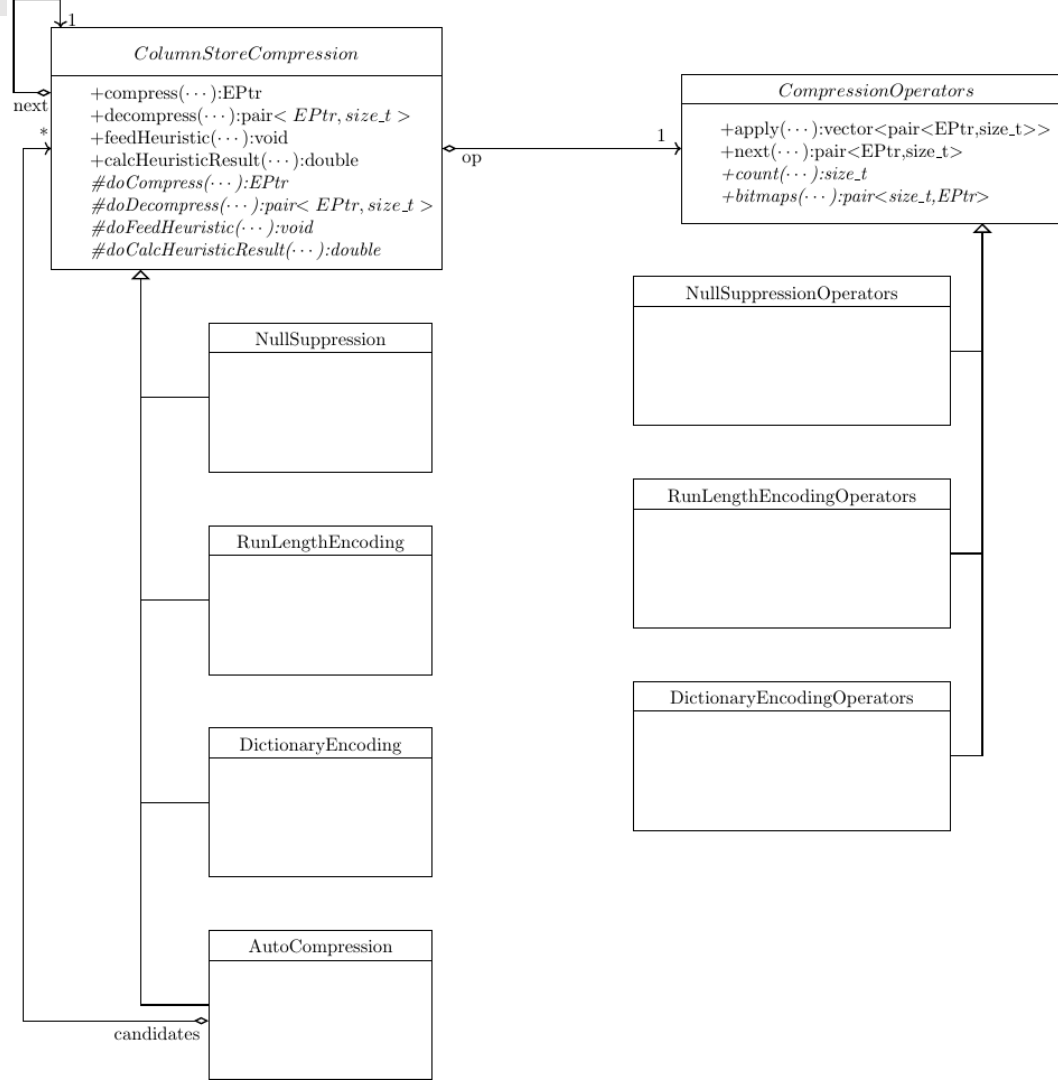
$n_{dist} = 3, effectiveBytes = 1$

SIMD-FastPFOR
Heuristics h_{pfor}

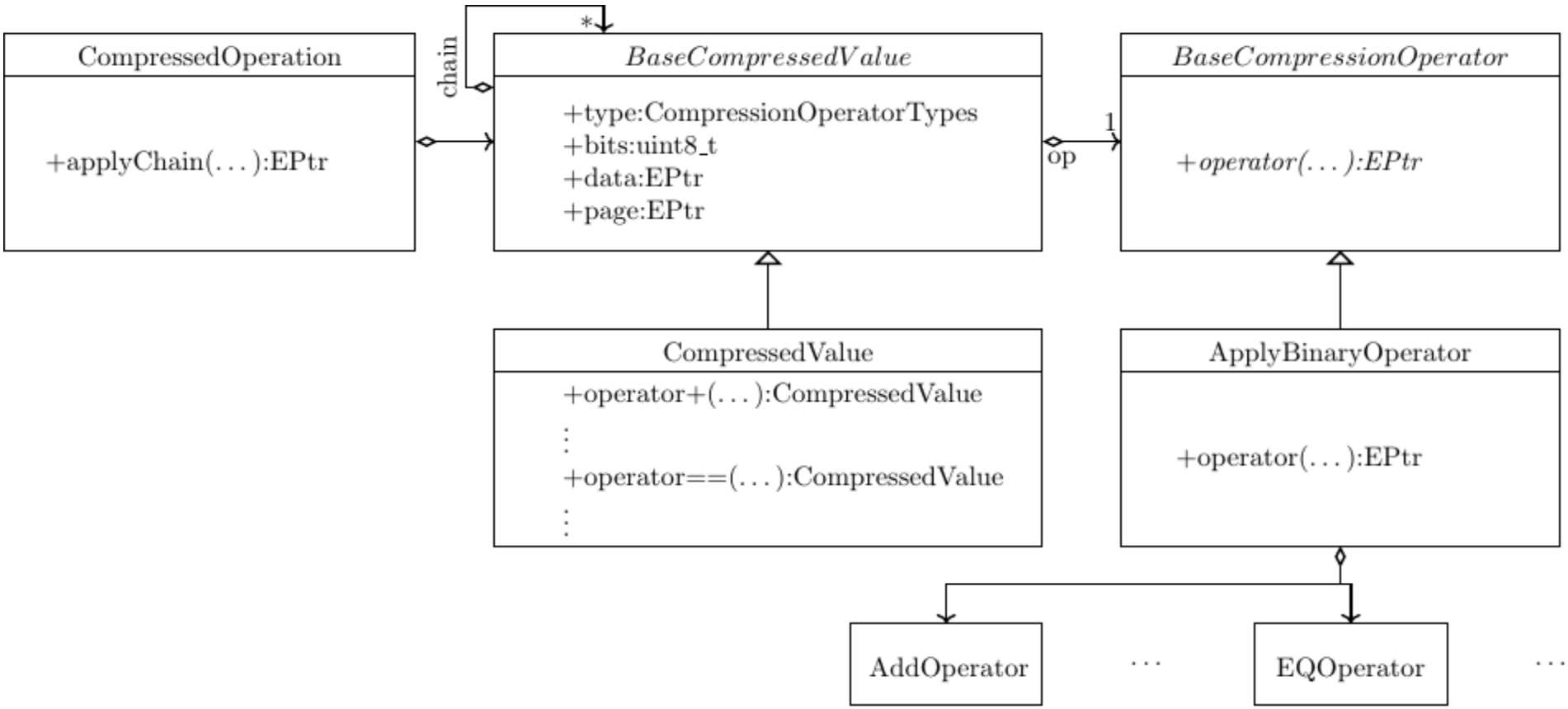
$$h_{pfor} = \frac{\log_2(avgValue) + 1}{32}$$



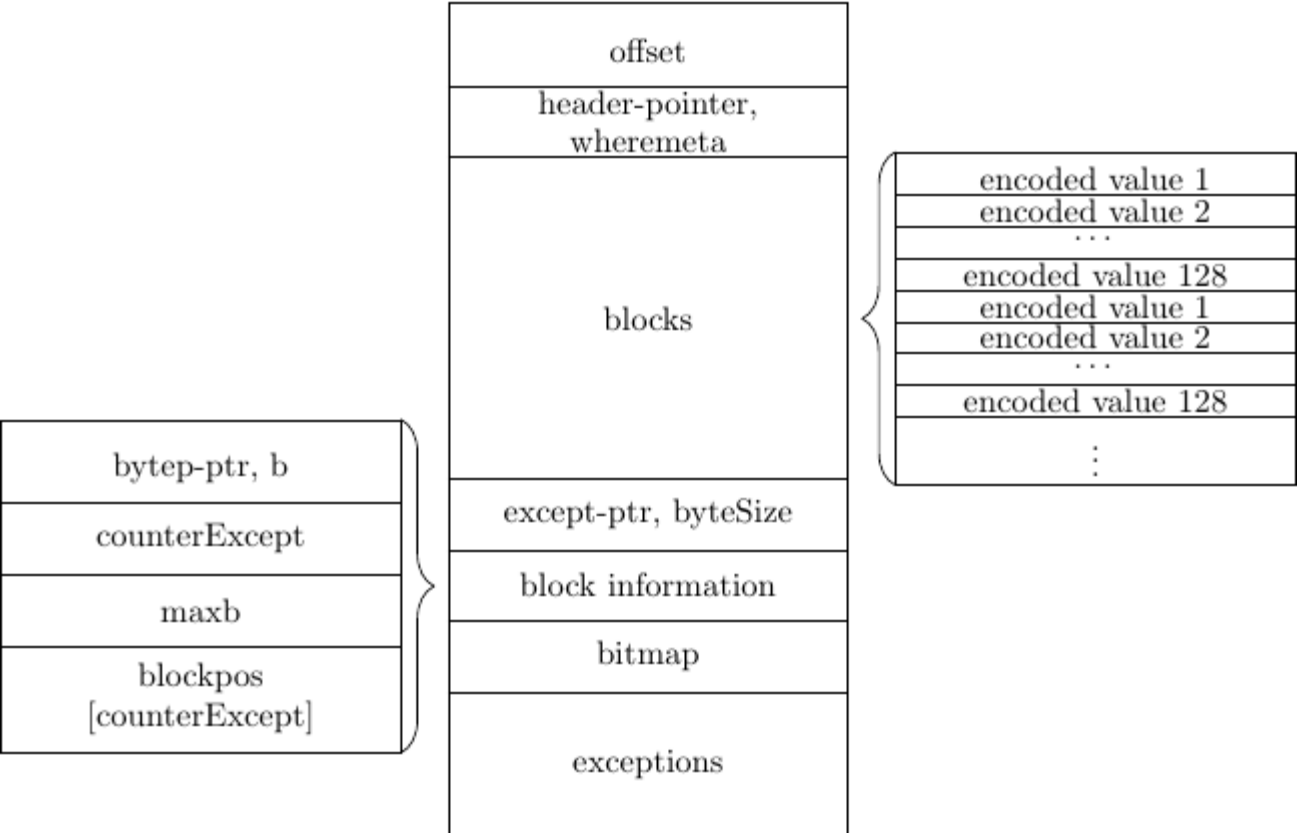
$avgValue = 296 \log_2(avgValue) = 8,21$



Compression operators structure



Structure of a FPFOR-Page



Structure of a RLE-Page

value 1
runlength 1
value 2
runlength 2
⋮
value n
runlength n

runlength 1
runlength 2
⋮
runlength n
value 1
value 2
⋮
value n

Overview

