TECHNISCHE UNIVERSITÄT DRESDEN

Dresden Database Systems Group

Conflict Detection-based Run-Length Encoding – AVX-512 CD Instruction Set in Action

Annett Ungethüm, Johannes Pietrzyk, Patrick Damme, Dirk Habich, *Wolfgang Lehner*

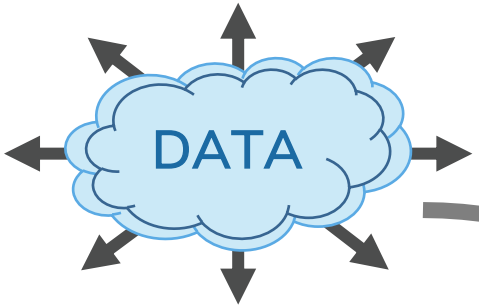
HardBD & Active'18 Workshop in Paris, France on April 16, 2018

Challenges for Data Processing Nowadays

Dresden Database Systems Group

Application Side

Multiple application areas



Rapidly growing data volumes
to be processed efficiently

System Side

Increasingly fast processors
for processing the data

Problem

Growing gap
between processor speed
and main memory bandwidth

Main bottleneck for efficient
data processing nowadays

Solution

Lightweight Compression
Access less bytes for the same
logical information

- Reduced transfer times
- Better cache utilization
- Less TLB misses

Growing main memory
to store the data

Lightweight Compression Techniques

Dresden Database Systems Group

Technique = abstract idea of how compression works

RLE

Run Length Encoding
Replace run by value & length

1200	1200
1200	4
1200	300
1200	2
300	
300	

DELTA

Differential Coding
Replace data elem. by difference to predecessor

1000	1000
1100	100
1150	50
1350	200
1355	5

FOR

Frame-of-Reference
Replace data elem. by difference to reference value

1200	200
1100	100
1000	0
1050	50

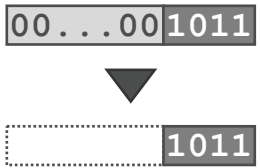
DICT

Dictionary Coding
Replace data elem. by 0-based key in dictionary

1000	0
1200	1
1000	0
1050	2
1050	2

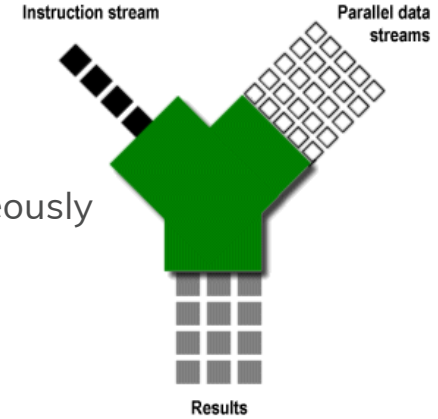
NS

Null Suppression
Eliminate leading zeroes in binary representation



Vectorization is crucial from performance perspective

Vectorization using SIMD



Dresden Database Systems Group

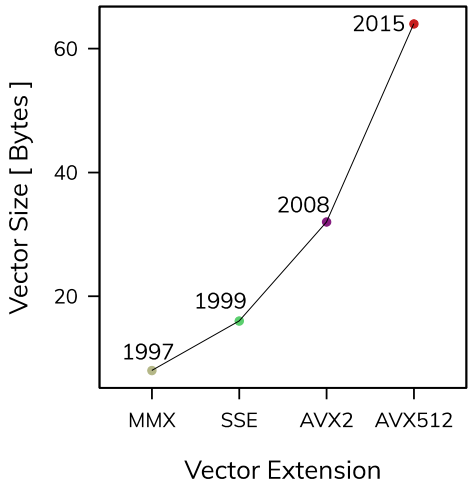
Single Instruction Multiple Data (SIMD)

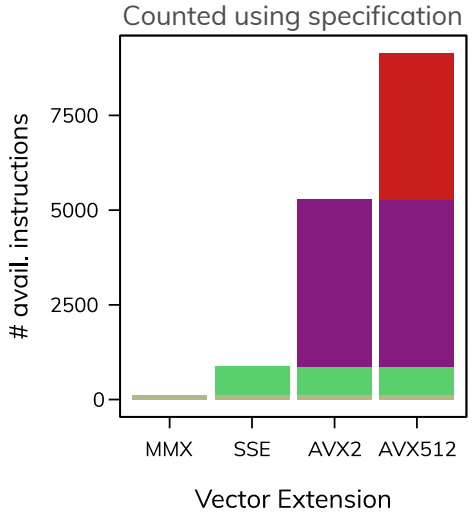
- same instruction on multiple data points simultaneously

Development of Intel's SIMD Extension

- Trend to larger vector registers
  - 128-bit (SSE)
  - 256-bit (AVX and AVX2)
  - 512-bit (AVX-512)

- Trend to more instructions





Vectorization and Lightweight Data Compression

Dresden Database Systems Group

Most algorithms have been proposed for 128-bit SIMD registers

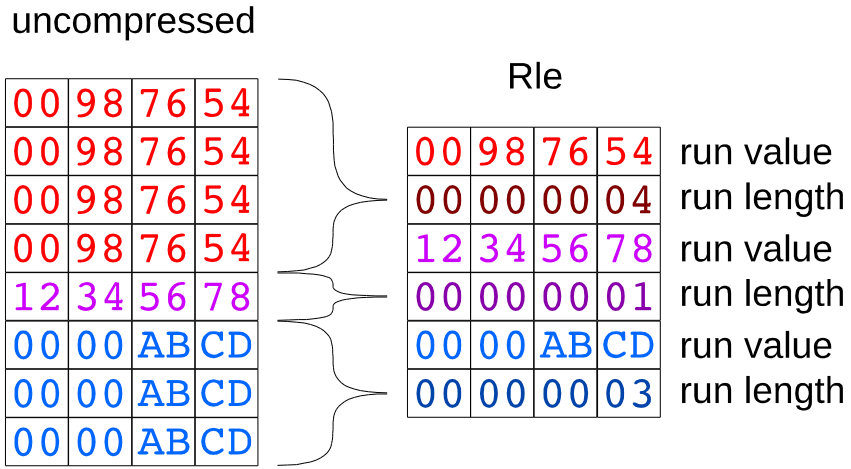
- Processing 4 elements (32 bit integers) at one

Example Run-Length Encoding

- View subsequent occurrences of the same value as a run
- Each run representable by its value and length → just two integers

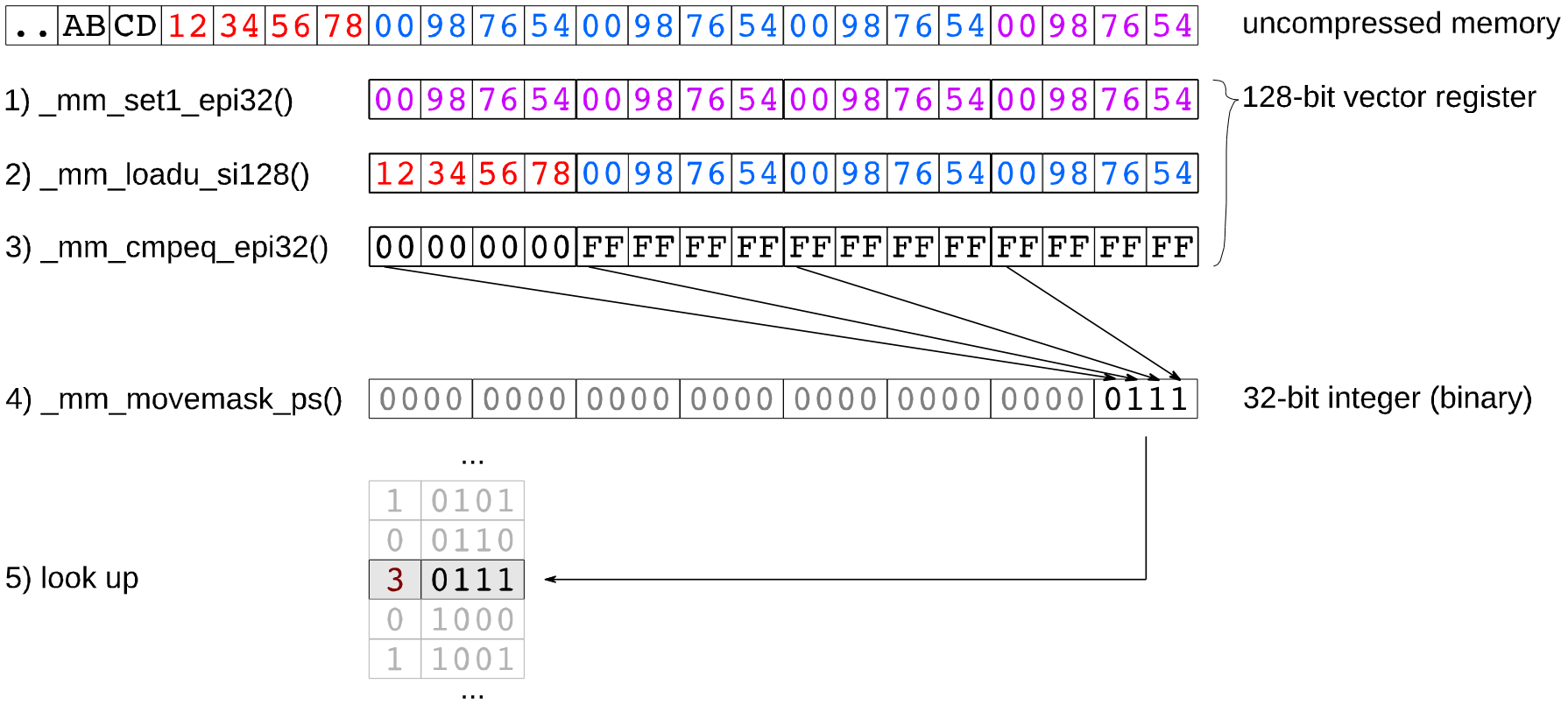
RLE-SIMD

- Uses SIMD instructions to parallelize comparisons



RLE-SIMD: Compression

Dresden Database Systems Group



Evaluation using Different Vector Sizes

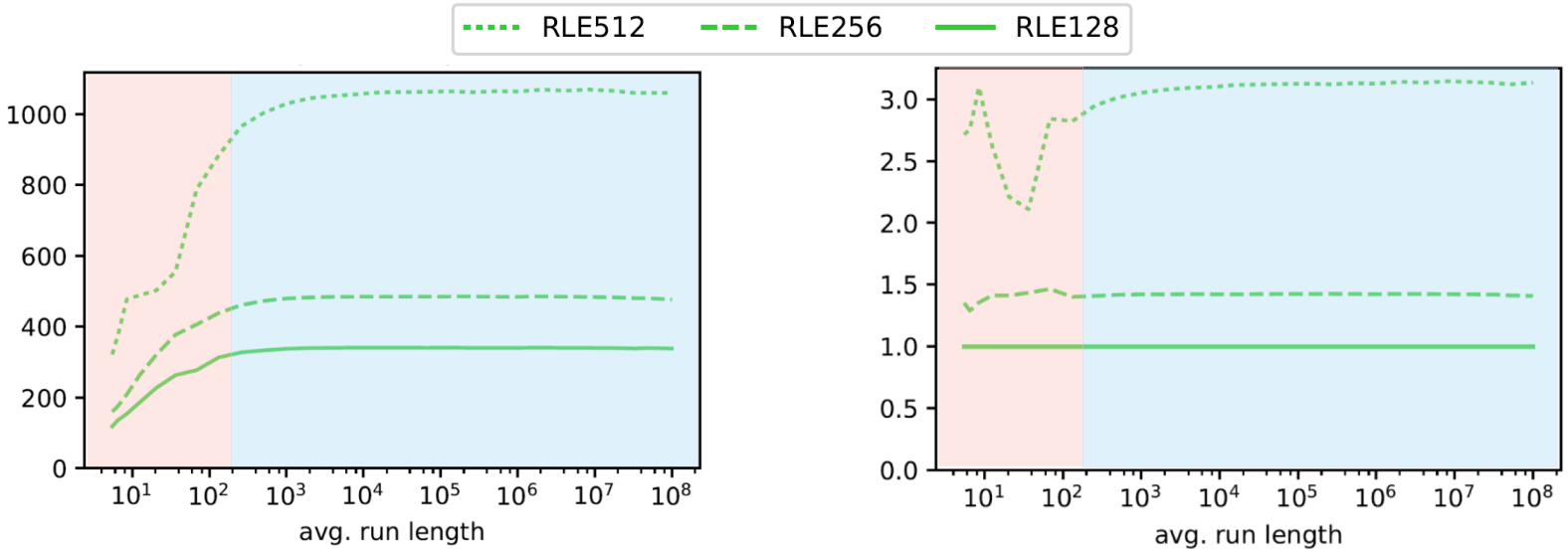
Dresden Database Systems Group

Compression Speed

- Measured in million integers per second (mis)

Speedup

- Compared to baseline of 128-bit



non-well performing area

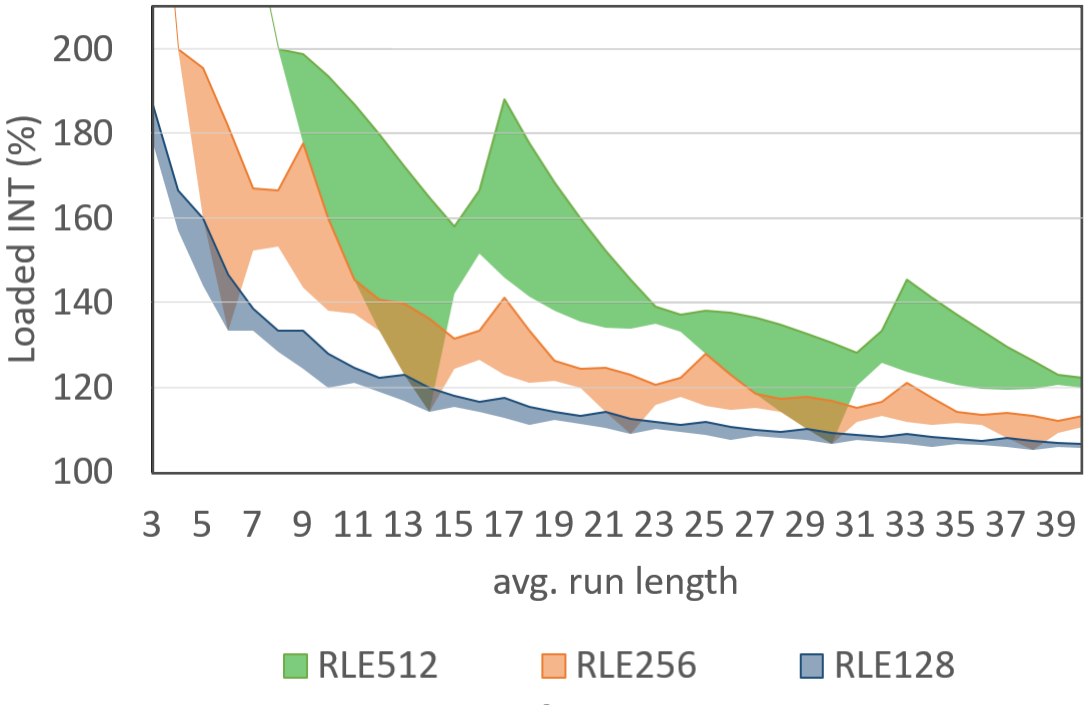
well-performing area

Non-Well Performing Area

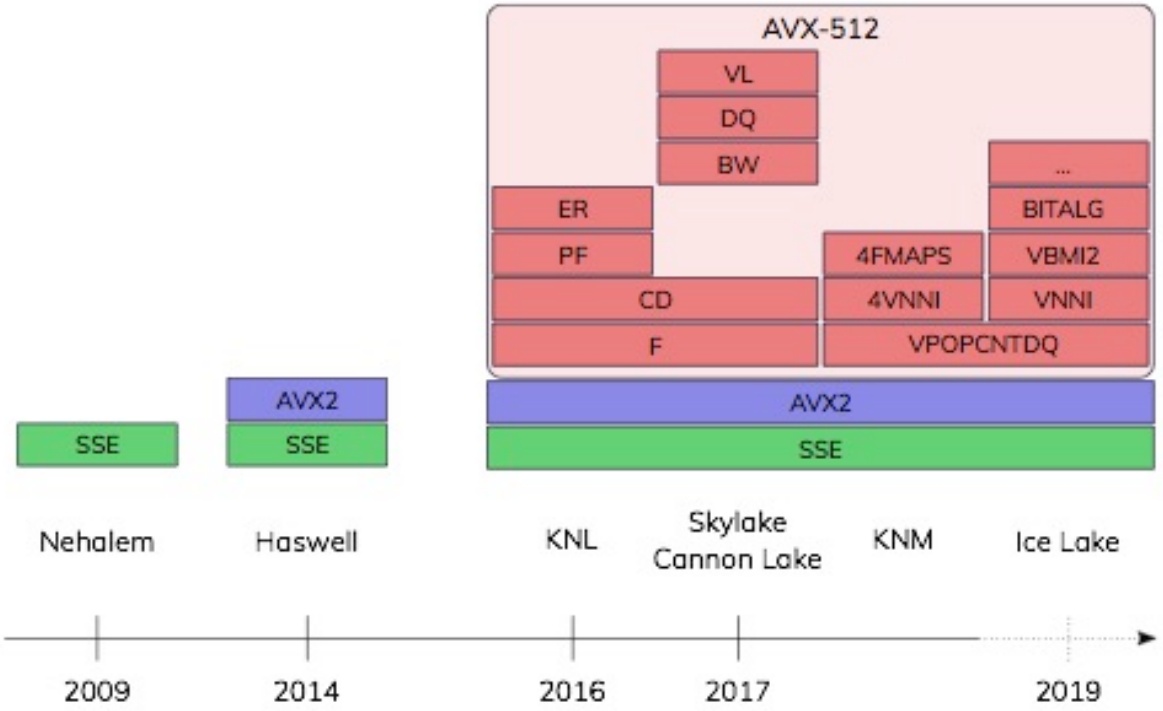
Dresden Database Systems Group

Reasons

- For large run lengths, the number of loaded integers approaches more or less 100%, i.e. every value is only processed once.
- RLE vectorization uses a significantly higher number of load operations for sequences with short runs.
- The redundant processing dramatically increases with increasing vector widths.

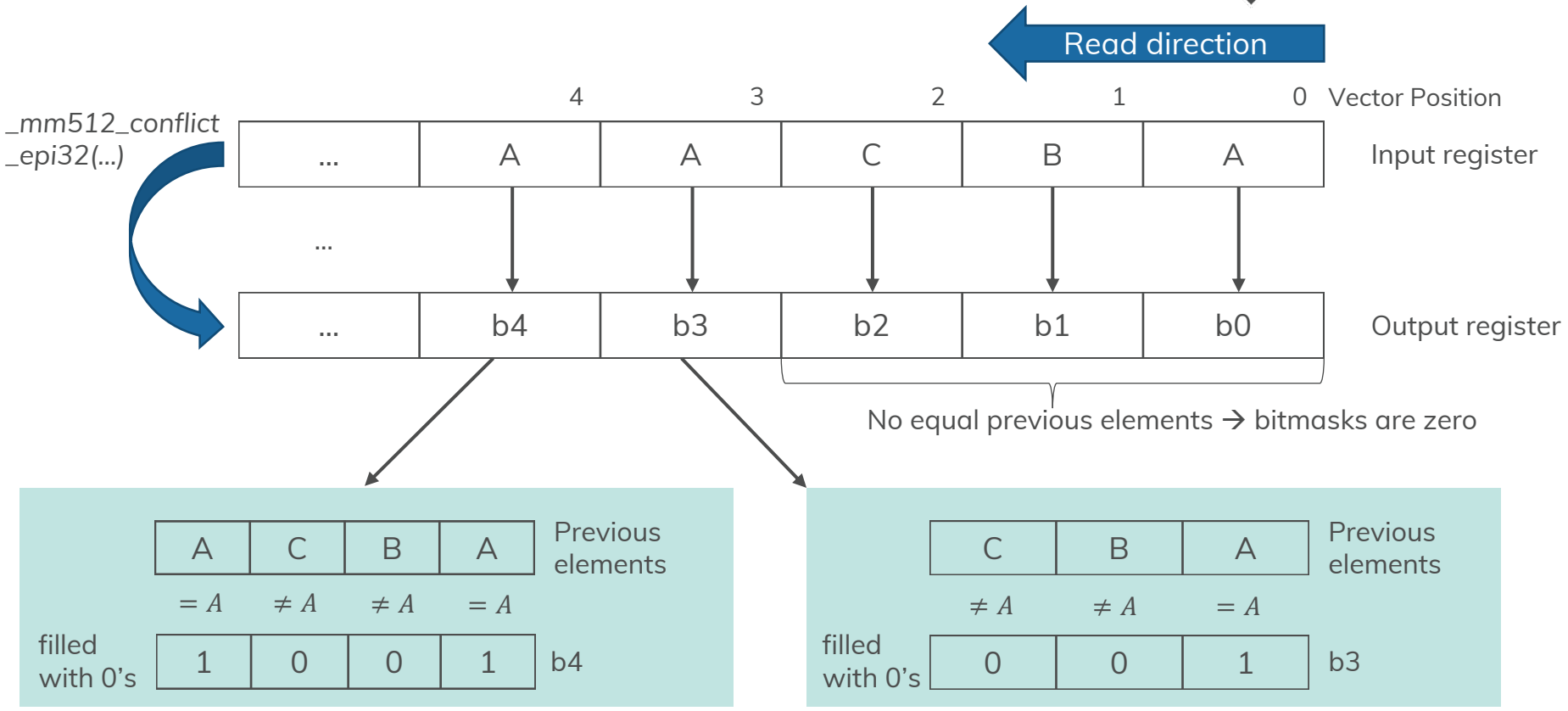


SIMD – New Instruction Sets

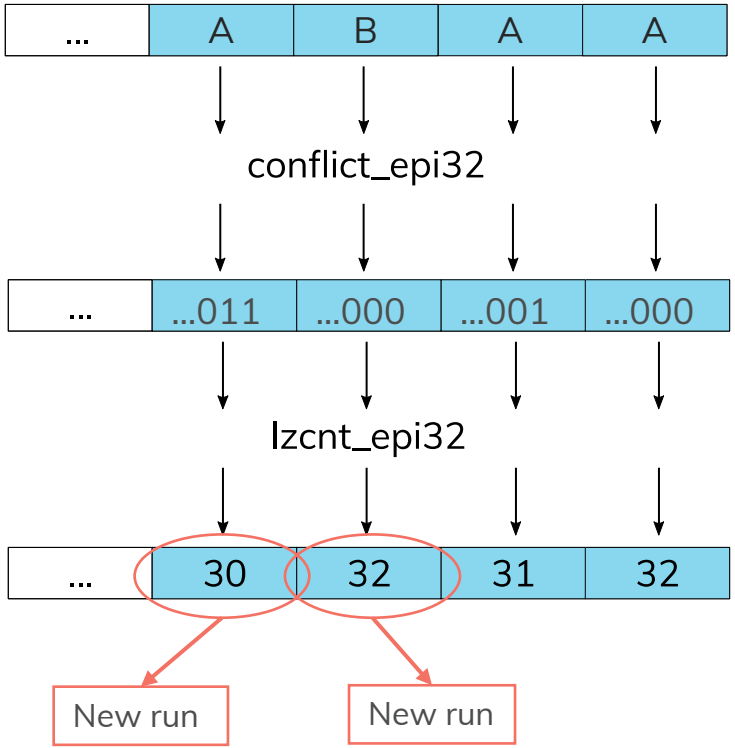


Conflict Detection using AVX512CD

Dresden Database Systems Group



Step 1: Run Detection



Input

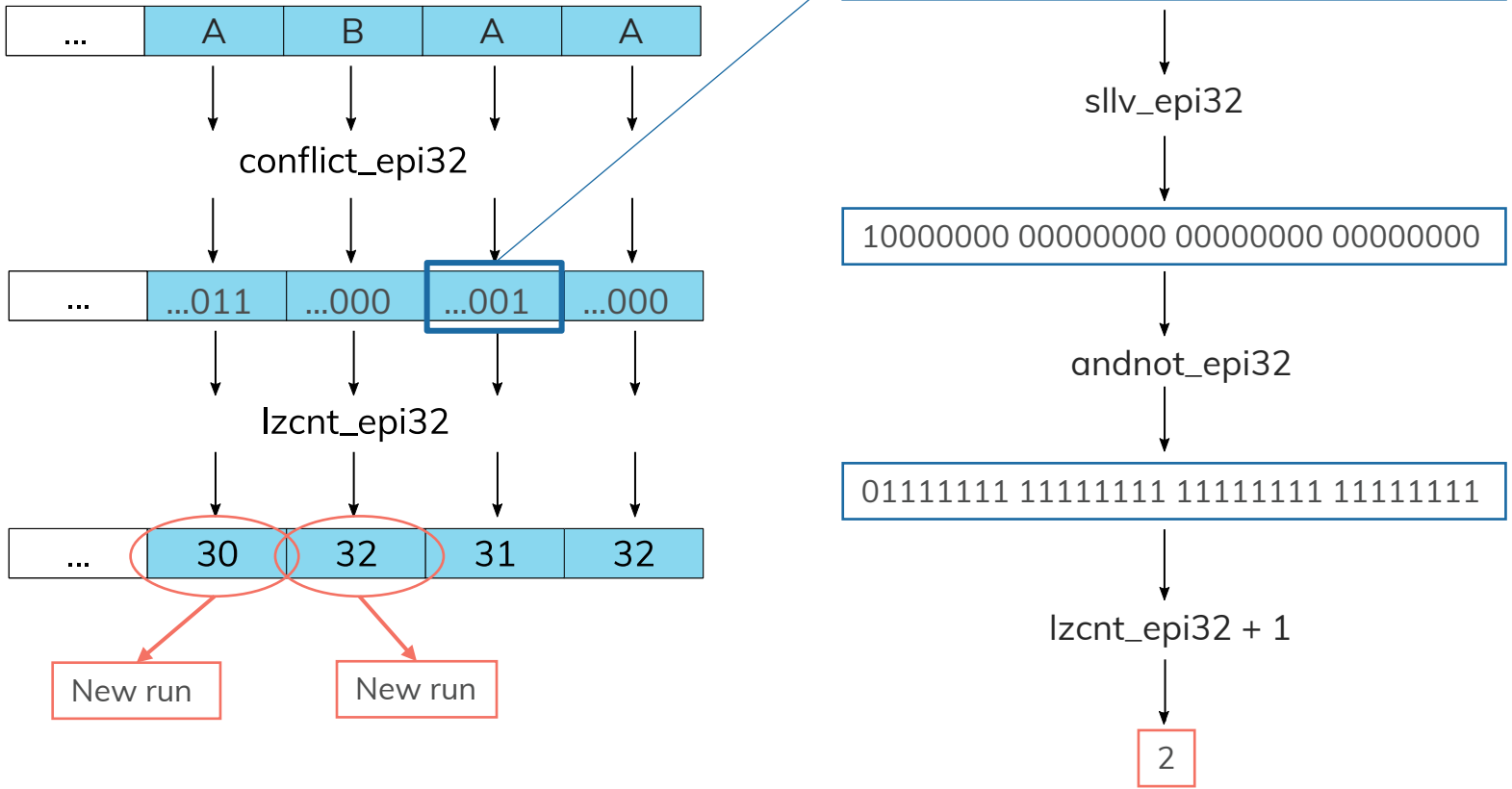
Conflict Detection

Resulting bitmask

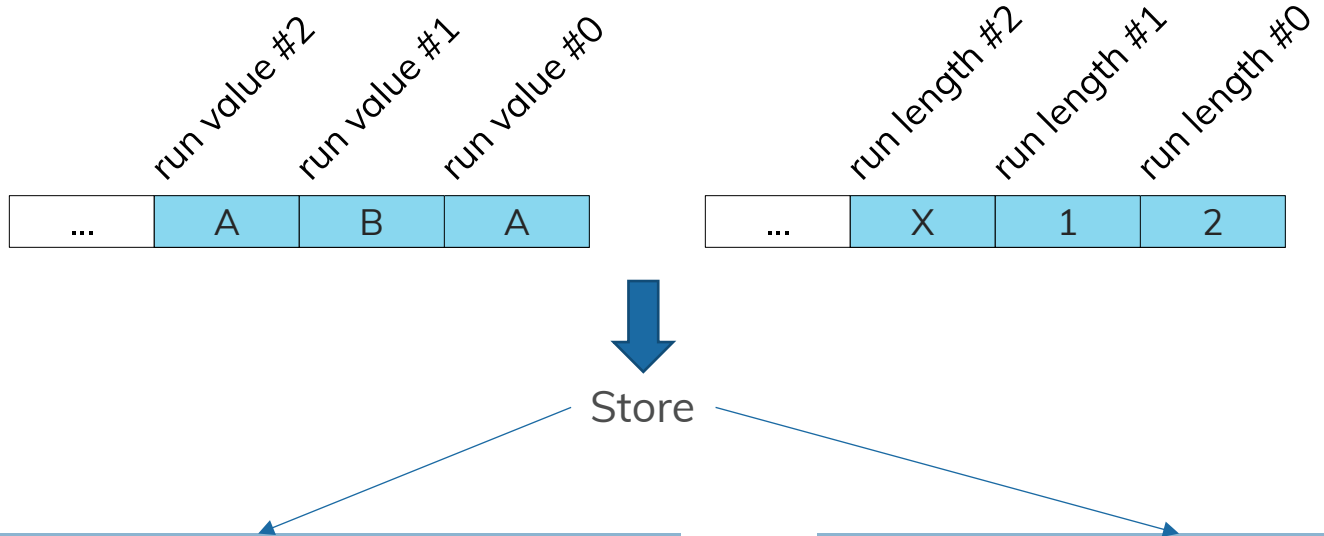
Count leading zeros

Are leading zeros descending?

Step 2: Run Length Detection



Step 3: Storing



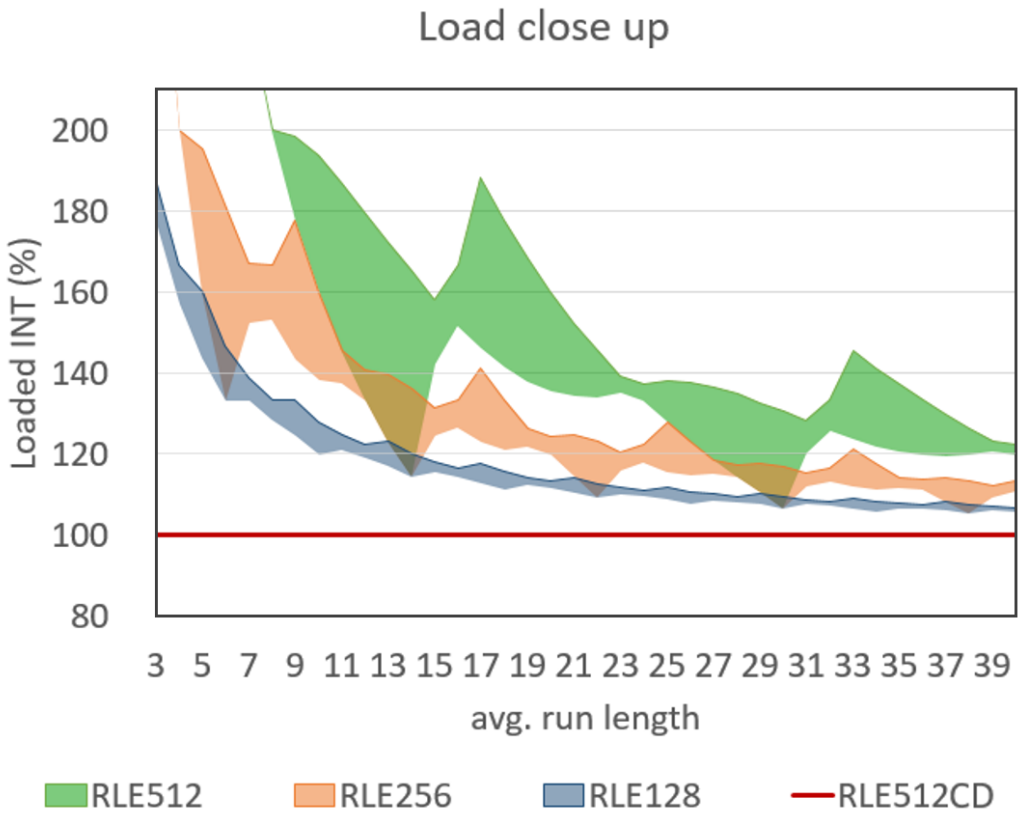
Scatter (RLE512CD)

- Classical storage layout: (run value, run length)-pair
- Independent of vector width

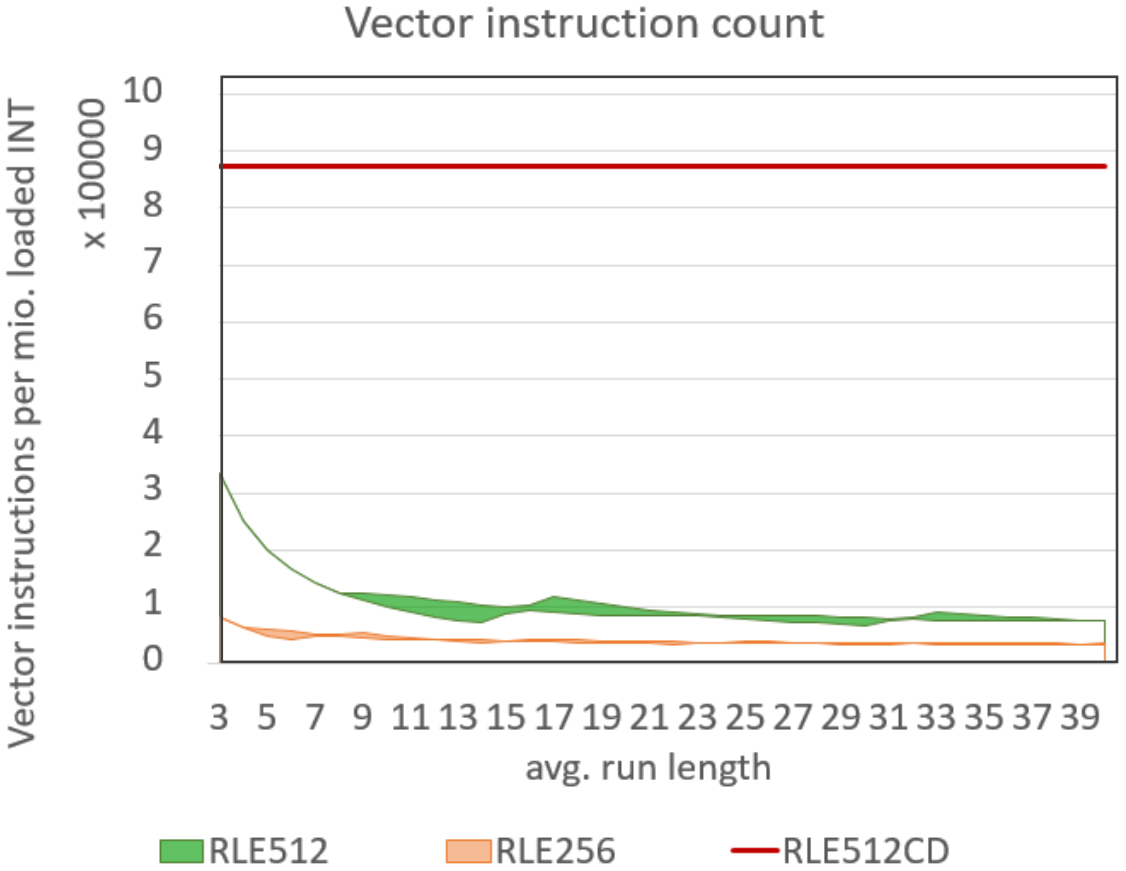
Continuous (RLE512CDAligned)

- Vector wise
- Run length and run value to different memory locations

Evaluation – Load Instructions



Evaluation- Vector Instructions



Evaluation

Dresden Database Systems Group

Runtime Comparison

- Intel Xeon Phi Knights Landing Processor

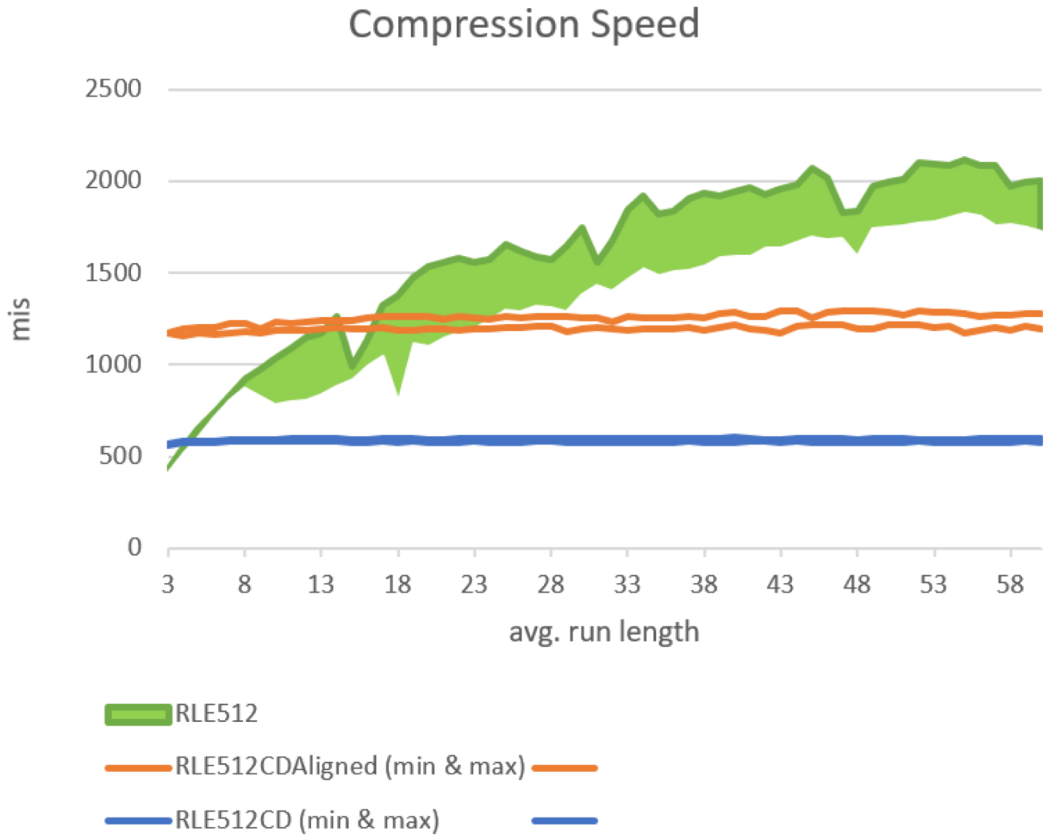
- RLE512CD (Aligned) outperforms state-of-the-art for small average run lengths

Evaluation

Dresden Database Systems Group

Runtime Comparison

- Intel Xeon 6130 Processor
- Similar results



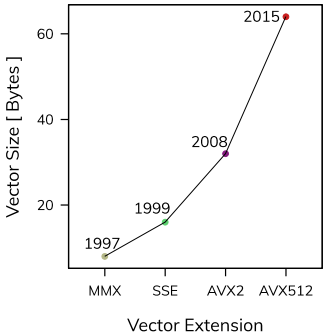
Summary

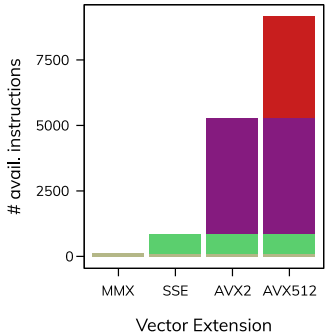
Dresden Database Systems Group

Development of Intel's SIMD Extension

- Trend to larger vector registers
  - 128-bit (SSE)
  - 256-bit (AVX and AVX2)
  - 512-bit (AVX-512)

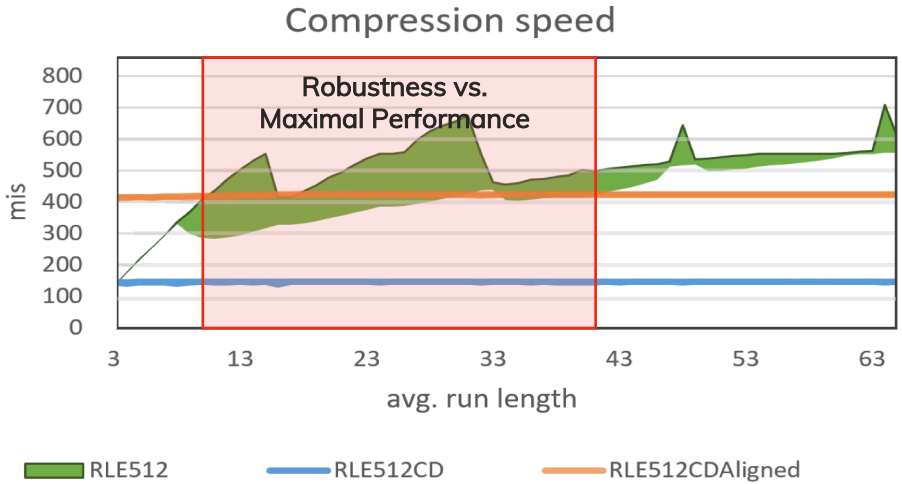
- Trend to more instructions





Run Length Encoding

- Proposed novel implementation using AVX512-CD functionality



TECHNISCHE UNIVERSITÄT DRESDEN

Dresden Database Systems Group

Conflict Detection-based Run-Length Encoding – AVX-512 CD Instruction Set in Action

Annett Ungethüm, Johannes Pietrzyk, Patrick Damme, Dirk Habich, *Wolfgang Lehner*

HardBD & Active'18 Workshop in Paris, France on April 16, 2018