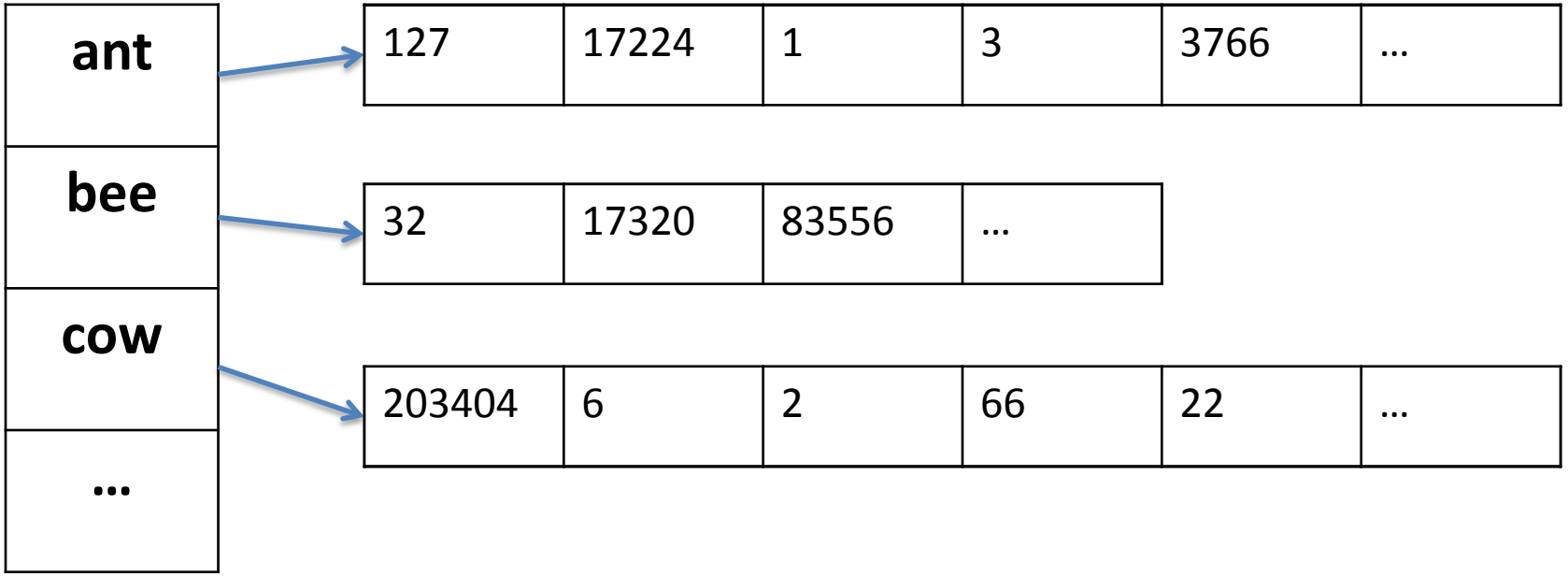
CIKM 2011
Glasgow, UK

# SIMD-Based Decoding of Posting Lists

Alexander A. Stepanov, Anil R. Gangolli, Daniel E. Rose,
Ryan J. Ernst, Paramjit Oberoi

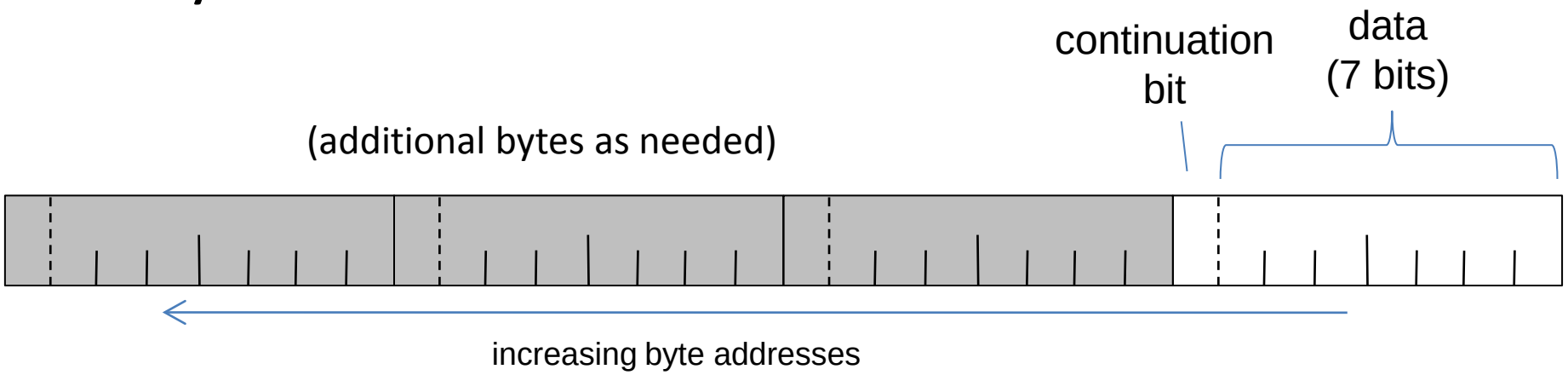
A9.com
130 Lytton Ave.
Palo Alto, CA 94301
USA

# Posting Lists

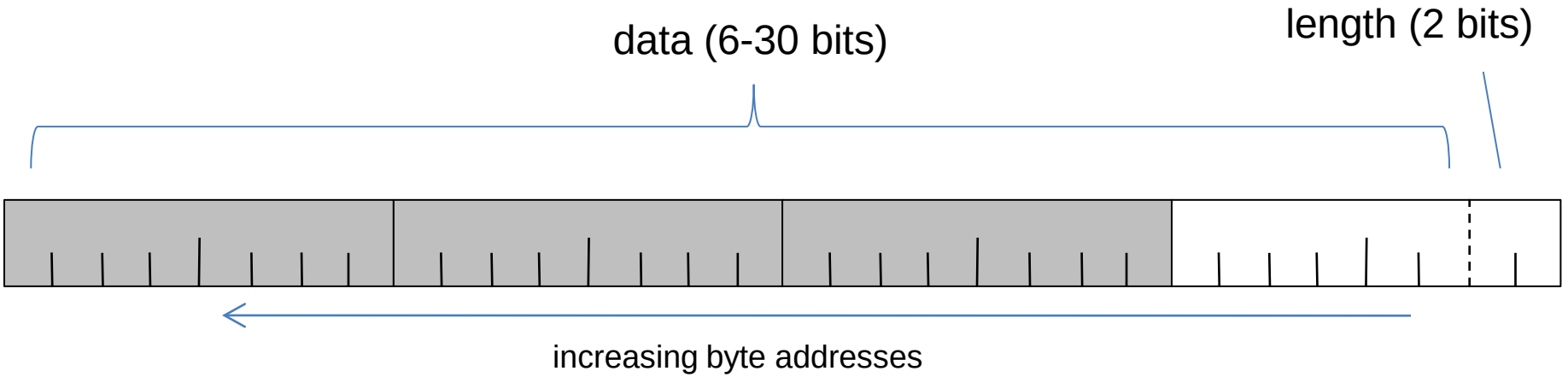



Replacing IDs with deltas gives smaller numbers,
which can be stored in less space given appropriate encoding.

# “vbyte” Format:




# Grossman’s “Byte Aligned (BA)” Format:



# Definition: *Byte-Oriented Encoding*

1. All significant bits of the natural binary representation are preserved.
2. Each byte contains bits from only one integer.
3. Data bits within a single byte of the encoding preserve the ordering they had in the original integer.
4. All data bits from a single integer precede all bits from the next integer.

# Descriptors

- When does an integer end?
- Equivalent to knowing its length
- Encodings use auxiliary *descriptor bits* to represent the length

# Dimensions of Encodings

- Descriptor can express length in ***binary*** or ***unary***.
- Descriptor bits can be stored adjacent to each single integer, or descriptors of several integers can be ***grouped*** so that each byte contains either descriptor or data.
- If length of a single integer is expressed in unary, the bits of the unary representation may be ***packed*** contiguously or ***split*** across several bytes (as in vbyte).

# A Taxonomy of Byte-Oriented Encodings

| Our Name | Arrangement | Length Encoding | Names in the Literature |
|---|---|---|---|
| varint-SU | Split | Unary | v-byte, vbyte, VB, varint, VInt |
| | Packed | Unary | |
| | Group | Unary | |
| | Split | Binary | |
| varint-PB | Packed | Binary | BA, varint30 |
| varint-GB | Group | Binary | group varint,<br>k-wise (k=4) null suppression |

# A Taxonomy of Byte-Oriented Encodings

| Our Name | Arrangement | Length Encoding | Names in the Literature |
|---|---|---|---|
| varint-SU | Split | Unary | v-byte, vbyte, VB, varint, VInt |
| varint-PU | Packed | Unary | none (introduced here) |
| varint-GU | Group | Unary | none (introduced here) |
| varint-SB | Split | Binary | none (not useful) |
| varint-PB | Packed | Binary | BA, varint30 |
| varint-GB | Group | Binary | group varint,<br>k-wise (k=4) null suppression |

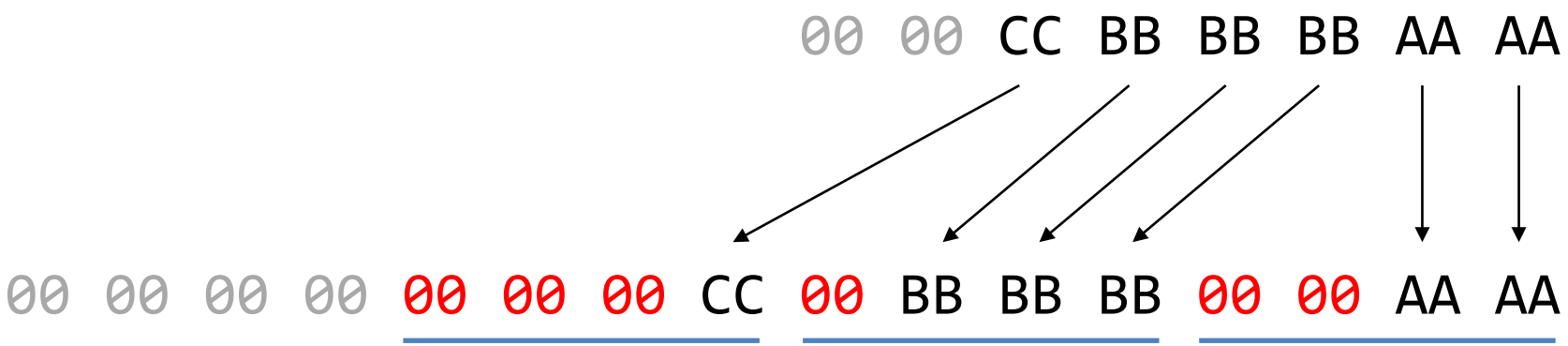
# Definition: *Byte-Preserving Encoding*

We call a format ***byte-preserving*** if each byte containing significant bits in the original integer appears without modification in the encoded form.

Observe:

- Encoding omits leading 0-bytes
- Decoding reinserts them

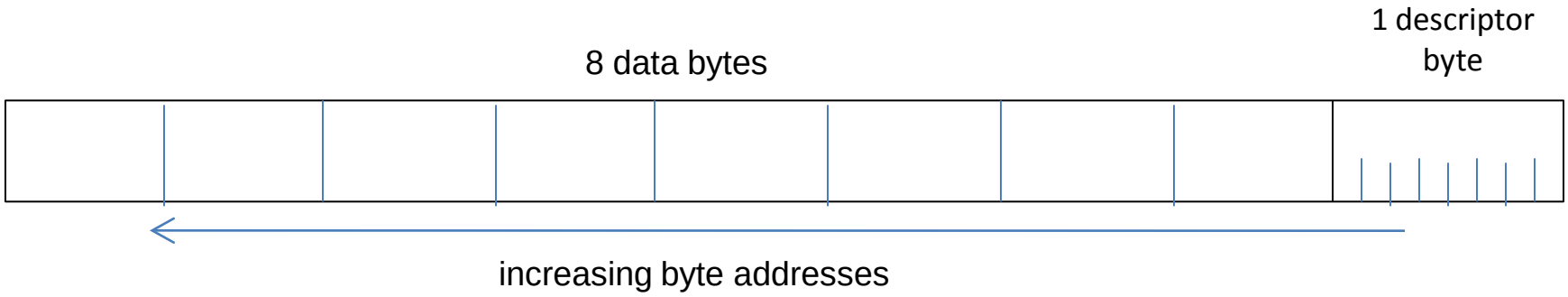
# Re-Inserting 0-bytes in Parallel



# Format for SIMD Decoding

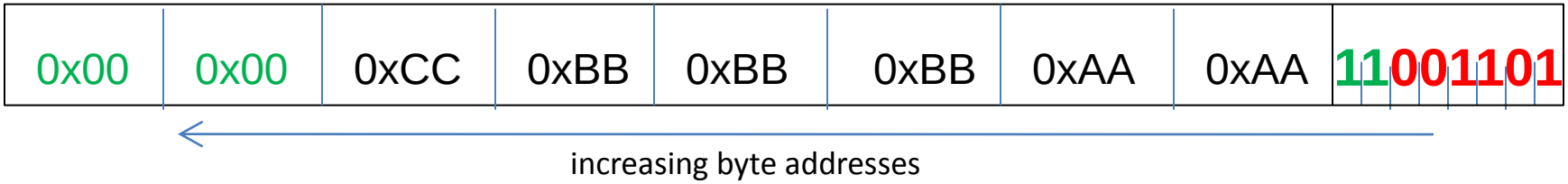
- Group descriptor bits from several encoded integers into a separate *descriptor byte*
- Group data bytes into k-byte blocks
- Decode *however many integers fit in this block*

# varint-GU




- Represent up to 8 variable-sized integers (as many as fit in 8 bytes)
- For each integer *i*, descriptor contains length(*i*)-1 in unary, separated by 0s
- Number of integers in block is number of zero bits in descriptor

Example: Encode 4 integers 0xAAAA, 0xBBBBBB, 0xCC, 0xDDDDDDDD.
Byte counts are 2, 3, 1, 4. Last integer doesn't fit in this block; pad with 0s.



# Intel SIMD PSHUFB Instruction

- Permutes data bytes in parallel, with optional insertion of 0-bytes.
- Operation specified by a “shuffle sequence”
- Both data and shuffle sequences are stored in special registers (currently 16 bytes)

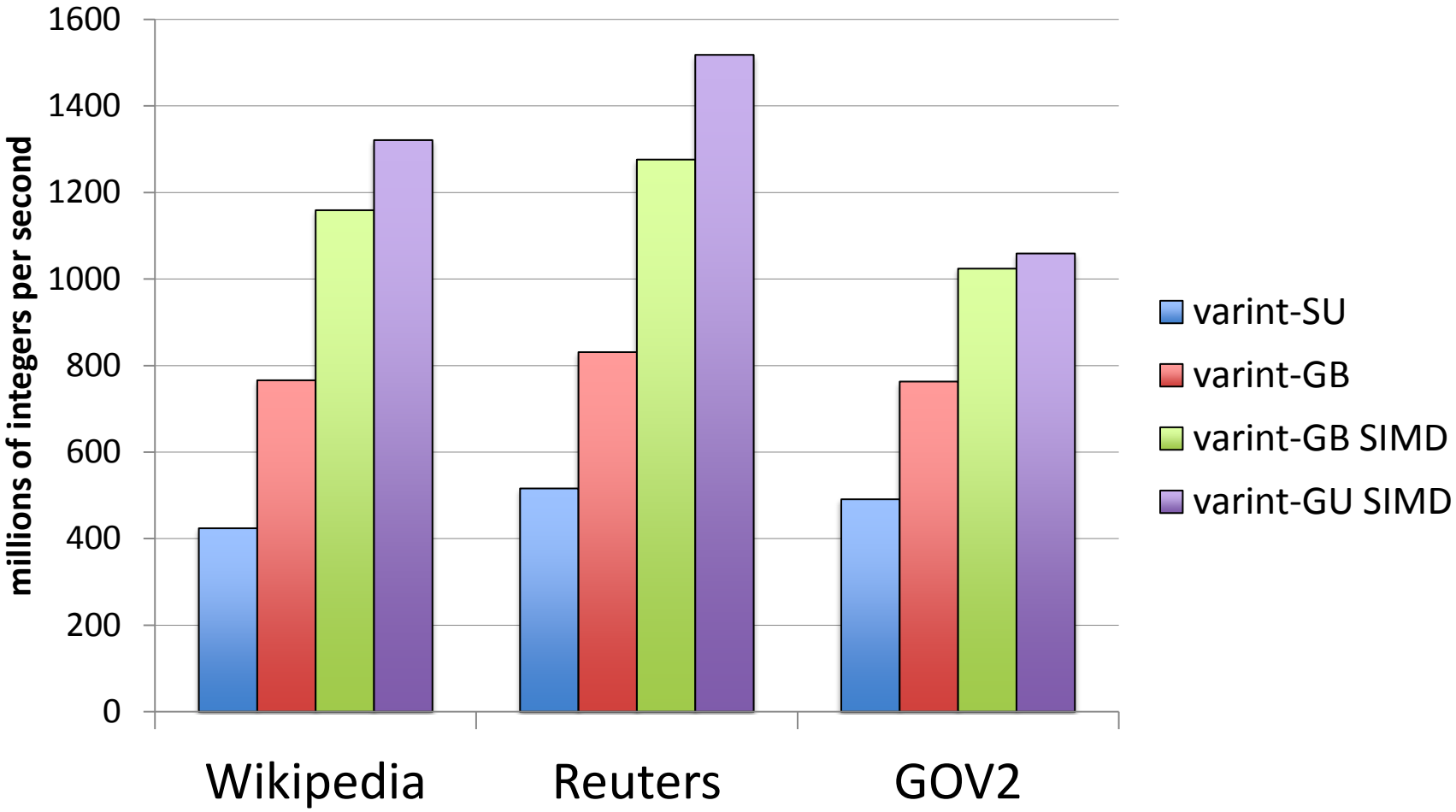
# Decoding Using PSHUFB

- We pre-compute a table of 256 possible shuffle sequences
- Each descriptor uniquely identifies the arrangement and lengths of the integers
- So, we use descriptor to index into table

# Generic Decoding Algorithm

1. Read a chunk of data and its corresponding descriptor.
2. Look up the appropriate shuffle sequence and offset from the table.
3. Perform the shuffle.
4. Write the result.
5. Advance the input and output pointers.

# Results: Decoding Speed



# Conclusions

- Taxonomy of byte-oriented formats clarifies relationships of existing formats and reveals new ones.
- SIMD provides significant performance gains for integer decoding.
- New format (varint-GU) outperforms others.