Accelerating Toeplitz Hash with GFNI

Why Toeplitz Hash Matters

RSS (Receive Side Scaling) distributes packets across NIC queues using Toeplitz hash
Software needs the same hash for:

- Flow classification / steering decisions
- SNAT — pick source port so return traffic lands on the right queue
- MPLS / VXLan / IPSec — bind tunnel to a desired queue
- TCP stack — choose source port for outgoing connections

Problem: The scalar implementation is slow for high-throughput paths

Toeplitz hash pseudo-code

- Loop for each input bit
- Data-dependent
- Some techniques may slightly improve performance

```
// For hash-input input[] of length N bytes (8N bits)
// and random secret key K of X bits
ComputeHash(input[], N)
Result = 0;
For each bit b in input[] {
    if (b == 1) then
        Result ^= (left-most 32 bits of K);
        shift K left 1 bit position;
    }
    return Result;
```

Intel® 82599 10 GbE Controller Datasheet 7.1.2.8.1 RSS Hash Function

What the Toeplitz hash really is

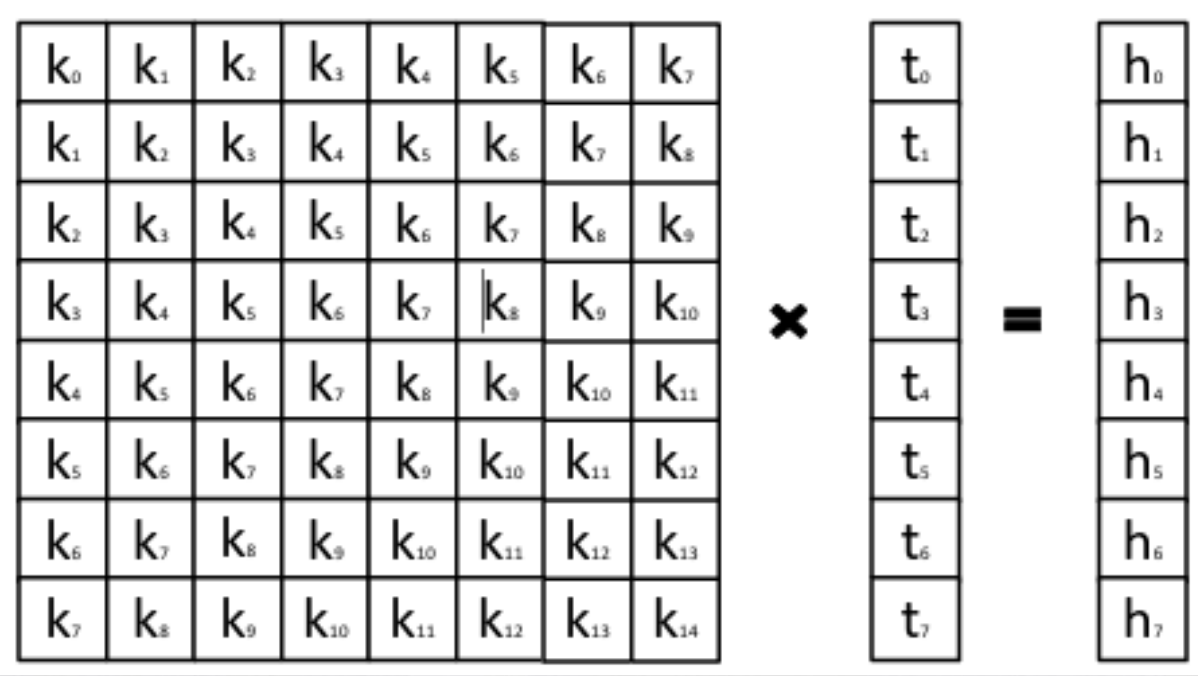
- Input tuple represented as a bit vector
- Multiplication by the matrix of a special form (Toeplitz matrix)
- Multiplication over GF(2)

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

https://en.wikipedia.org/wiki/Toeplitz_Hash_Algorithm

What is GFNI

- Galois Field New Instructions set
- 2 instructions:
  - two vector multiplication in GF2 field extension
  - affine transformation (matrix multiplication)
- Details could be found here: [galois-field-new-instructions]



The GFNI Insight

Toeplitz hash = series of GF(2) affine transformations

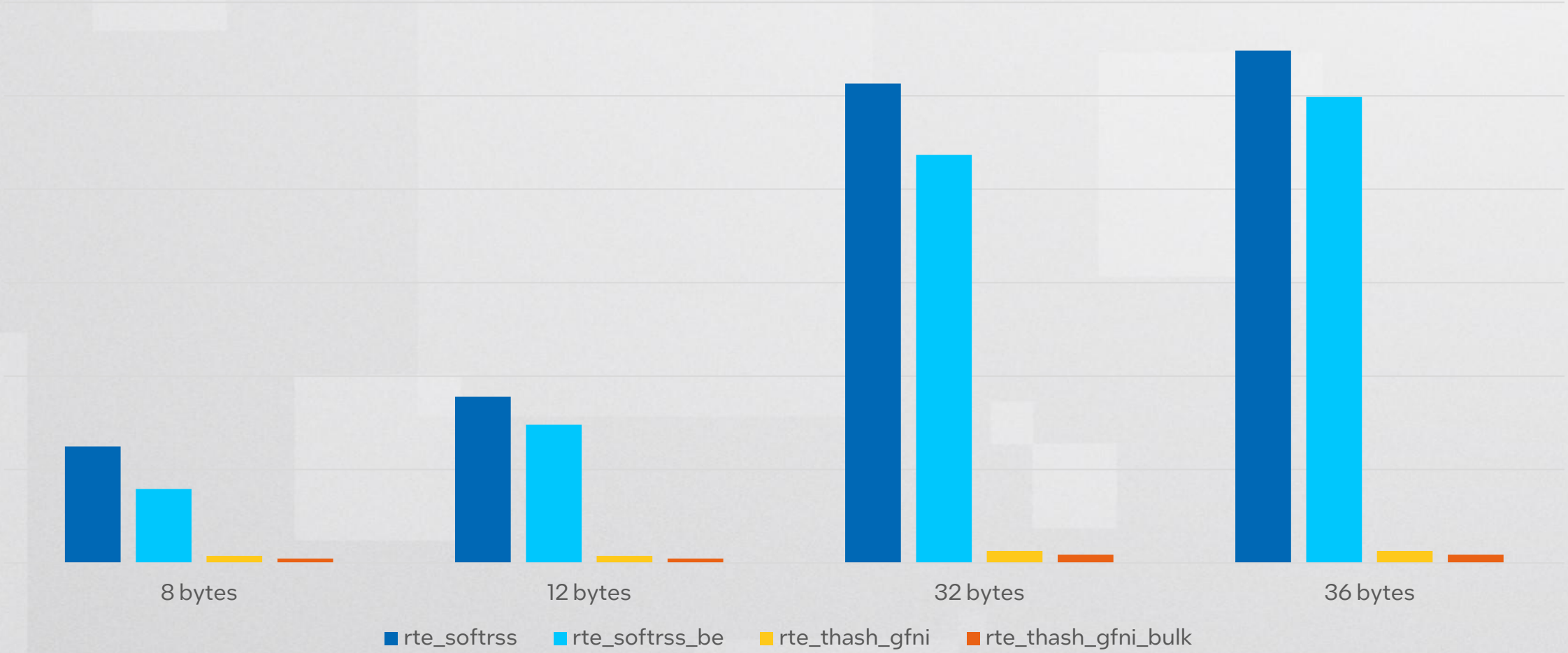
Concept	Mapping
RSS key	→ 8×8 binary matrices
Tuple bytes	→ input vectors in GF(2)
Hash computation	→ matrix × vector multiply over GF(2)

RSS hash key is converted into the set of matrices using rte_thash_complete_matrix()

Key instruction: GF2P8AFFINEQB (AVX-512 + GFNI)

- Performs 8 parallel affine transforms over GF(2) per 64-bit lane
- One 512-bit instruction processes 64 byte-transforms simultaneously
- Available since 3rd Gen Intel® Xeon® Scalable Processors (Ice Lake, 2019)

Performance



Thank You — Questions?

- **Code:** lib/hash/rte_thash_x86_gfni.h
  Docs: doc/guides/prog_guide/toeplitz_hash_lib.rst
  Benchmark: app/test/test_thash_perf.c
  Whitepaper: Intel GFNI Toeplitz Hash Technology Guide


intel