Fast Dictionary-based Compression for Inverted Indexes

Giulio Ermanno Pibiri
The University of Pisa
and ISTI-CNR
Pisa, Italy

Matthias Petri
The University of Melbourne
Melbourne, Australia

Alistair Moffat
The University of Melbourne
Melbourne, Australia

12/02/2019

Context — Inverted Indexes

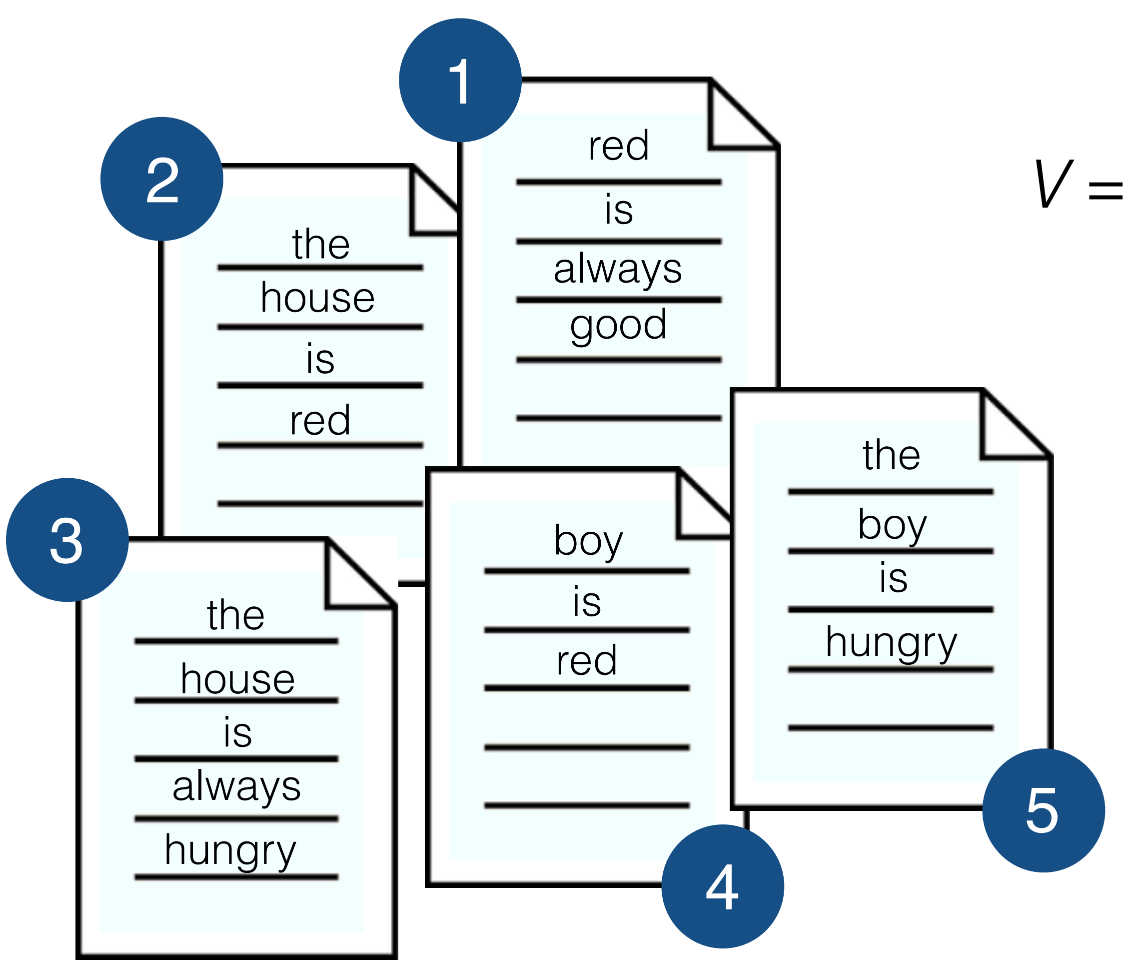
We focus on compression effectiveness
and decoding speed for **inverted indexes**.

The inverted index is the *de-facto* data structure
at the basis of every large-scale retrieval system.

Context — Inverted Indexes

We focus on compression effectiveness
and decoding speed for **inverted indexes**.

The inverted index is the *de-facto* data structure
at the basis of every large-scale retrieval system.



$$V = \{\overset{t_1}{\text{always}}, \overset{t_2}{\text{boy}}, \overset{t_3}{\text{good}}, \overset{t_4}{\text{house}}, \overset{t_5}{\text{hungry}}, \overset{t_6}{\text{is}}, \overset{t_7}{\text{red}}, \overset{t_8}{\text{the}}\}$$

$L_{t_1}=[1, 3]$
$L_{t_2}=[4, 5]$
$L_{t_3}=[1]$
$L_{t_4}=[2, 3]$
$L_{t_5}=[3, 5]$
$L_{t_6}=[1, 2, 3, 4, 5]$
$L_{t_7}=[1, 2, 4]$
$L_{t_8}=[2, 3, 5]$

Many solutions

Huge research corpora describing different **space/time** trade-offs.

- Elias gamma/delta
- Variable-Byte family
- Binary Interpolative Coding
- Simple family
- PForDelta
- Optimized PForDelta
- Elias-Fano
- Partitioned Elias-Fano
- Clustered Elias-Fano
- Asymmetric Numeral Systems

Many solutions

Huge research corpora describing different **space/time** trade-offs.

- Elias gamma/delta
- Variable-Byte family
- Binary Interpolative Coding
- Simple family
- PForDelta
- Optimized PForDelta
- Elias-Fano
- Partitioned Elias-Fano
- Clustered Elias-Fano
- Asymmetric Numeral Systems

Space

Interpolative

~**3X** smaller

← Spectrum →

Time

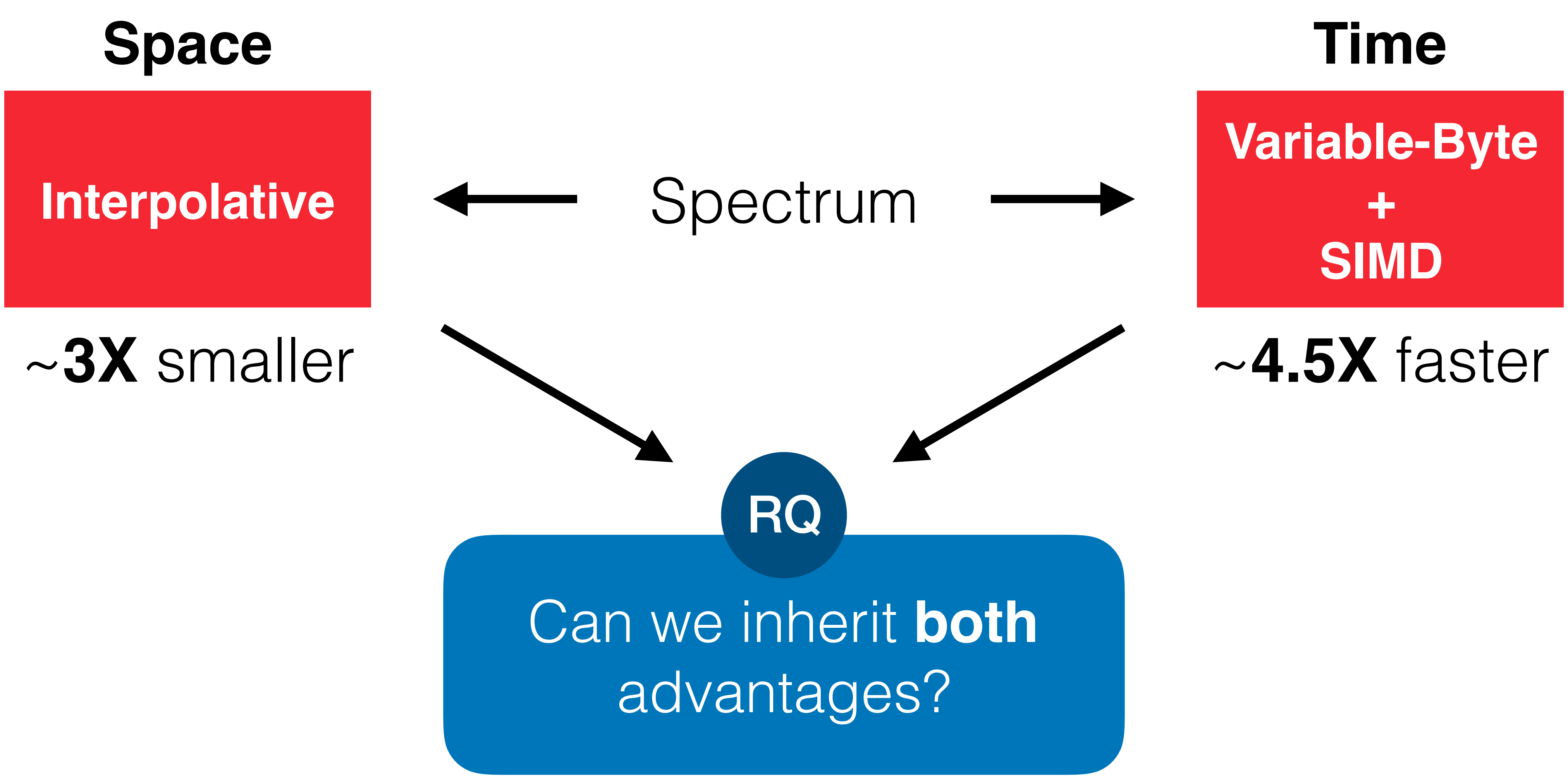
Variable-Byte
+
SIMD

~**4.5X** faster

Many solutions

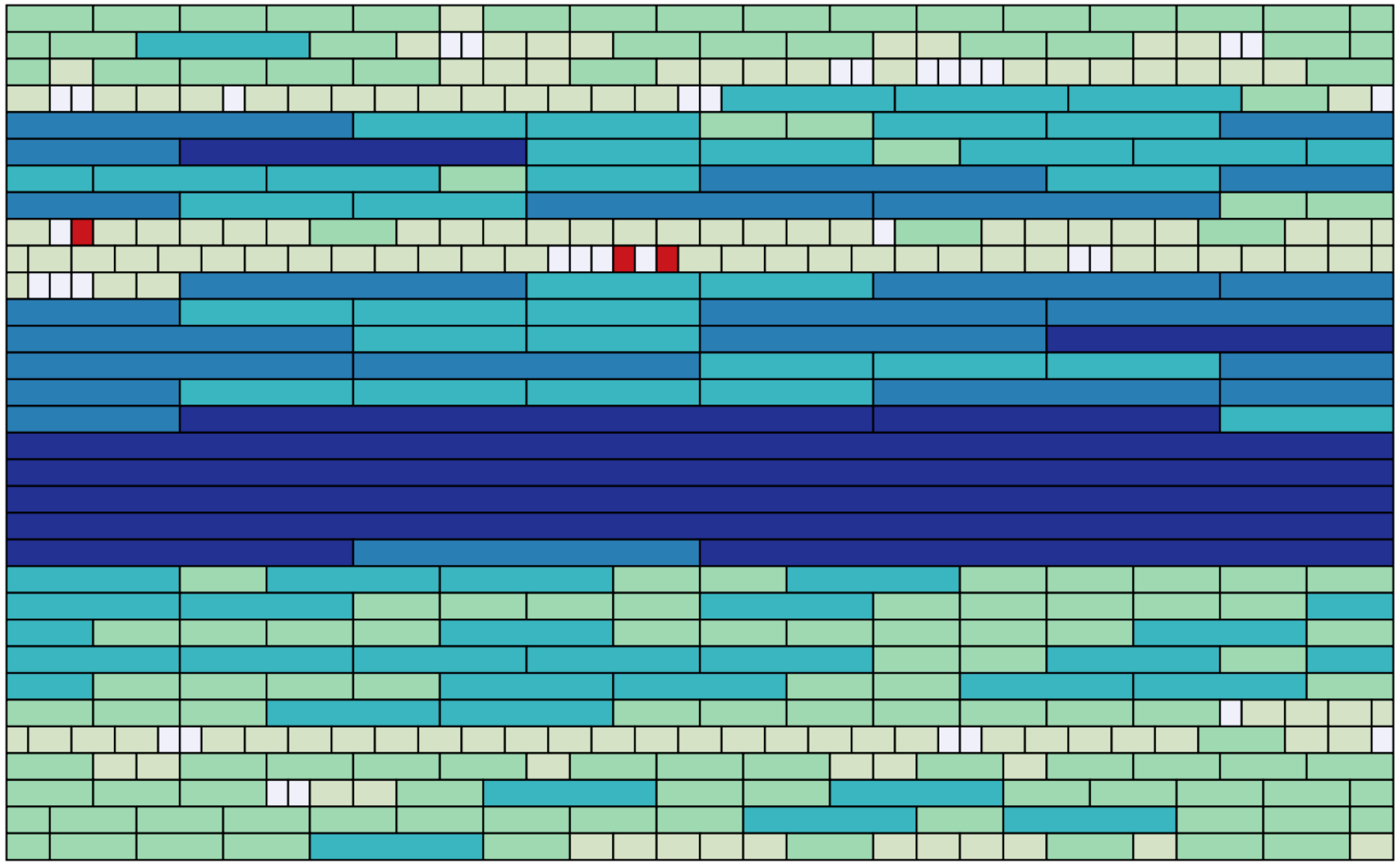
Huge research corpora describing different **space/time** trade-offs.

- Elias gamma/delta
- Variable-Byte family
- Binary Interpolative Coding
- Simple family
- PForDelta
- Optimized PForDelta
- Elias-Fano
- Partitioned Elias-Fano
- Clustered Elias-Fano
- Asymmetric Numeral Systems



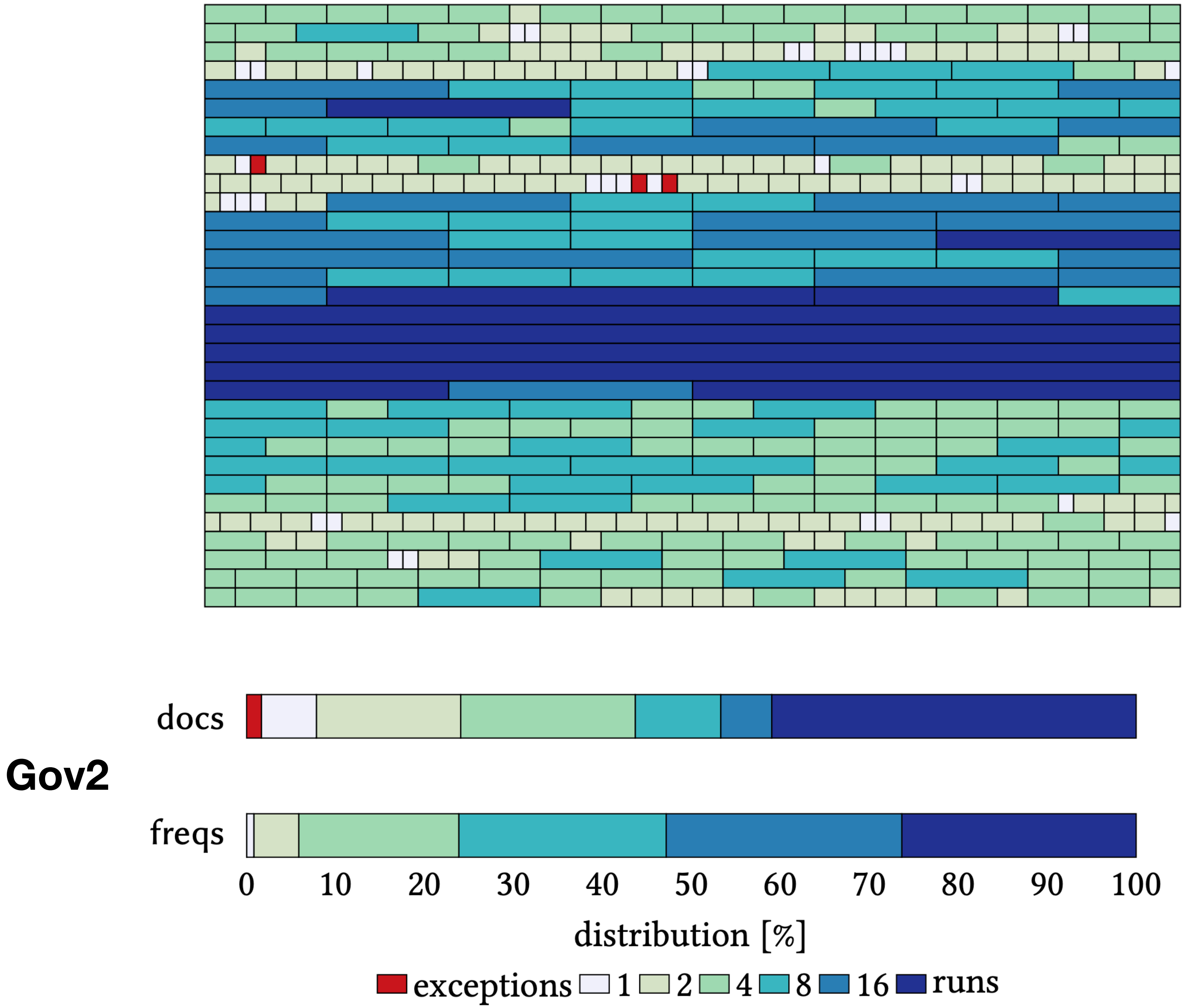
A crucial fact

Patterns of *d*-gaps are **repetitive**.

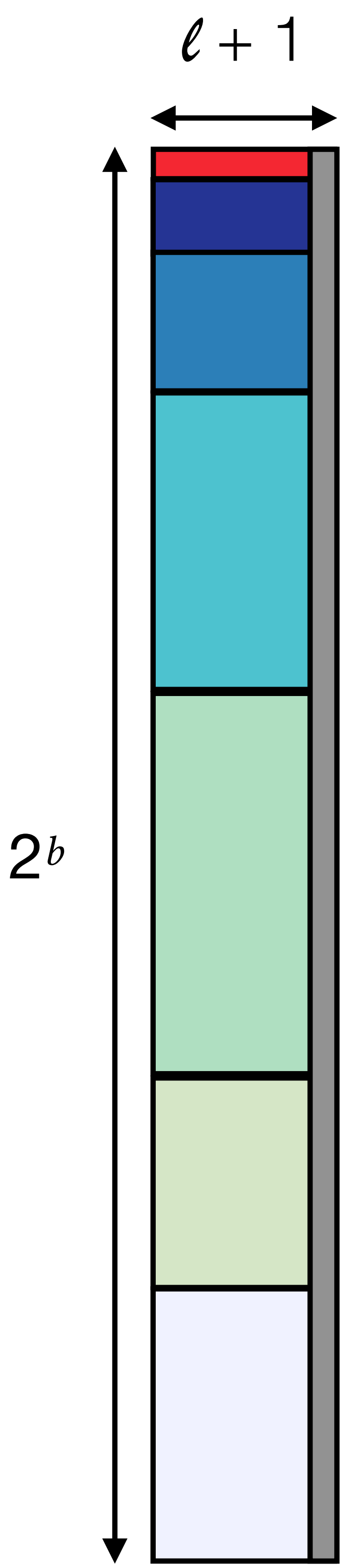


A crucial fact

Patterns of *d*-gaps are **repetitive**.

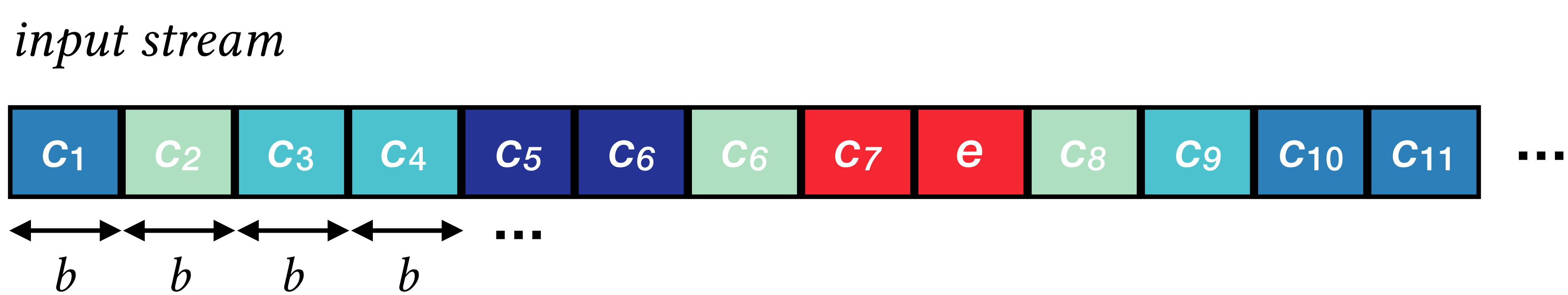


DINT — Dictionary of INTegers

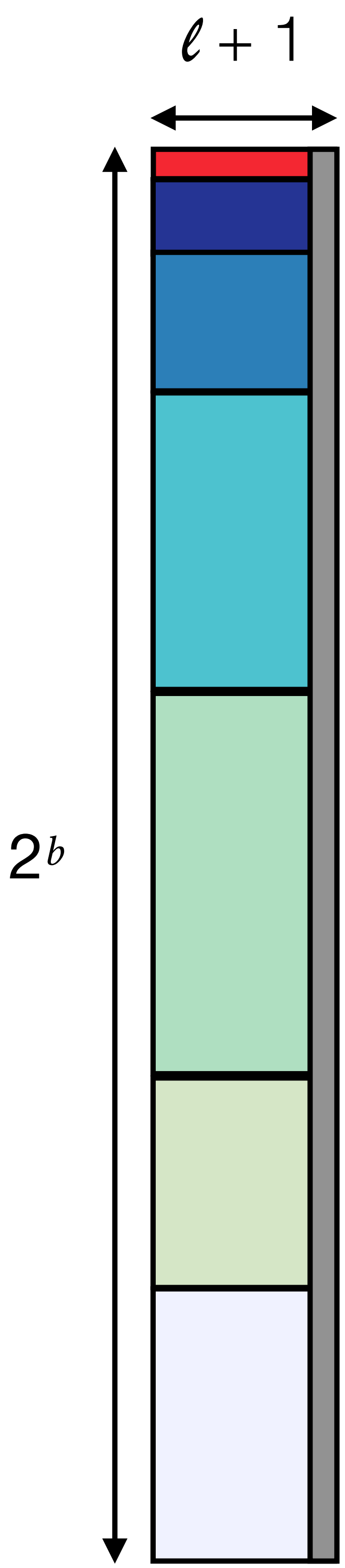


- Encode a whole pattern with a *single* dictionary reference of b bits
- Decode a whole pattern with a *single* dictionary access

***fixed-to-fixed* arrangement**



DINT — Dictionary of INTegers



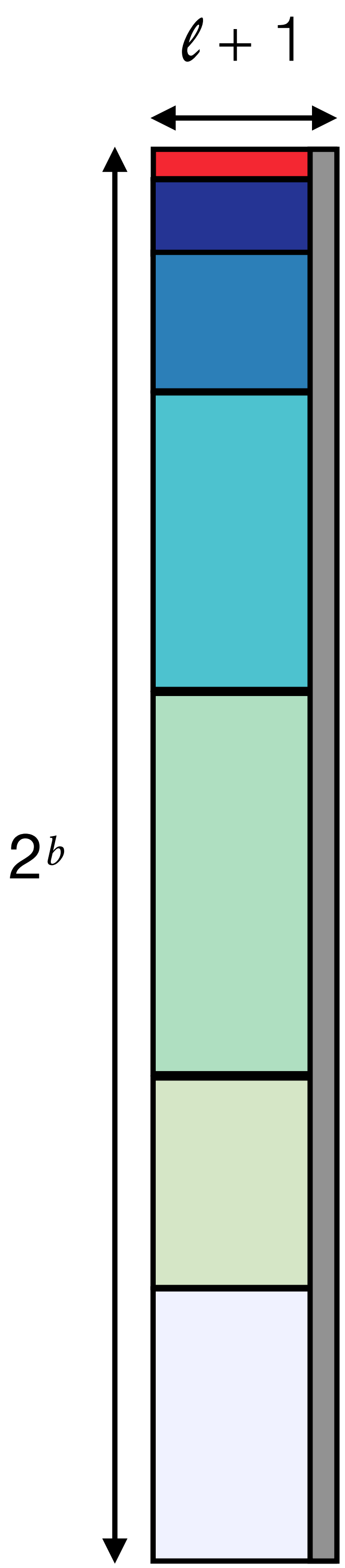
- Encode a whole pattern with a *single* dictionary reference of b bits
- Decode a whole pattern with a *single* dictionary access

***fixed-to-fixed* arrangement**

```
copy(D, c, output)
    begin = c × (ℓ + 1)
    copy 4 × ℓ bytes starting from D[begin] to output
    end = begin + ℓ
    size = D[end]
    return size
```

```
decode(D, input, output)
    for i = 0; i < B;
        c = get_16bits(input)
        size = 1
        if c > 2
            size = copy(D, c, output)
        else
            e = 0
            if c == 1
                e = get_32bits(input)
            else
                e = get_16bits(input)
            copy e to output
        i = i + size
```

DINT — Dictionary of INTegers



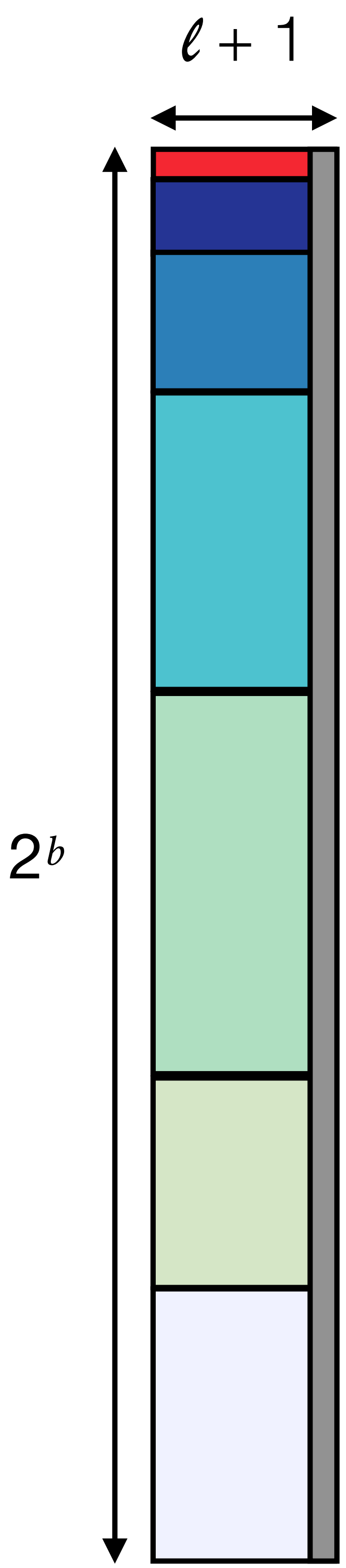
- Encode a whole pattern with a *single* dictionary reference of b bits
- Decode a whole pattern with a *single* dictionary access

***fixed-to-fixed* arrangement**

```
copy(D, c, output)
    begin = c × (ℓ + 1)
    copy 4 × ℓ bytes starting from D[begin] to output
    end = begin + ℓ
    size = D[end]
    return size
```

	Variable-length		Constant-length	
	docs	freqs	docs	freqs
instructions ($\times 10^9$)	53.63	35.02	41.72	28.35
instructions/cycle	1.16	1.13	1.28	1.24
cache-misses ($\times 10^7$)	10.77	9.06	8.21	7.60
branch-misses (%)	3.40	2.79	2.24	0.35
nanoseconds/int	1.82	1.08	1.12	0.73

DINT — Dictionary of INTegers



- Encode a whole pattern with a *single* dictionary reference of b bits
- Decode a whole pattern with a *single* dictionary access

***fixed-to-fixed* arrangement**

```
copy(D, c, output)
    begin = c × (ℓ + 1)
    copy 4 × ℓ bytes starting from D[begin] to output
    end = begin + ℓ
    size = D[end]
    return size
```

	Variable-length		Constant-length	
	docs	freqs	docs	freqs
instructions ($\times 10^9$)	53.63	35.02	41.72	28.35
instructions/cycle	1.16	1.13	1.28	1.24
cache-misses ($\times 10^7$)	10.77	9.06	8.21	7.60
branch-misses (%)	3.40	2.79	2.24	0.35
nanoseconds/int	1.82	1.08	1.12	0.73

1/3 of the time is saved

Refinements

1. **Packed dictionary structure**
   Exploiting string overlap

2. **Optimal block parsing**

3. **Multiple dictionaries**

Experimental results: setting

Datasets

Collection	Lists	Postings	Documents
Gov2	39,180,840	5,880,709,591	25,205,179
CCNEWS	43,844,574	20,150,335,440	43,530,315

Machine

Intel Xeon 6144 processor, 512 GiB RAM, Linux 4.13.0

Compiler

`gcc` 7.2.0 (with all optimizations)

C++ code available at
https://github.com/jermp/dint

Experimental results: compression effectiveness

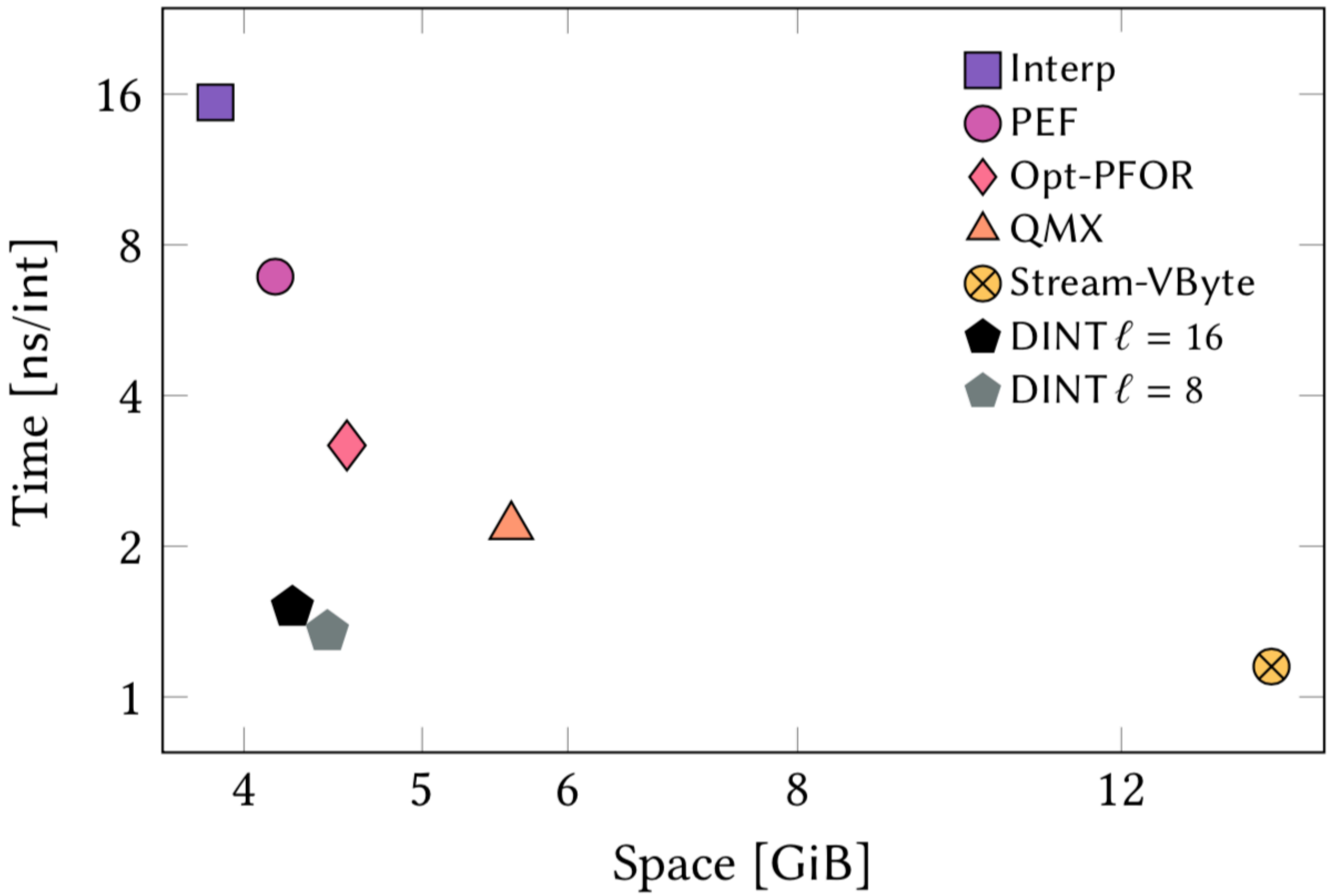
Method	Gov2			CCNEWS		
	GiB	docids	freqs	GiB	docids	freqs
Varint-GB	14.48	11.04	10.04	48.68	10.72	10.01
Varint-G8IU	12.77	9.90	8.69	43.87	9.75	8.93
VByte	11.85	9.22	8.02	39.65	8.88	8.00
QMX	5.59	4.99	3.11	19.20	5.12	3.04
Simple16	5.28	4.84	2.81	16.85	4.70	2.46
Opt-PFOR	4.55	4.33	2.26	15.50	4.49	2.10
DINT	4.29	4.22	1.98	15.09	4.48	1.92
PEF	4.16	3.85	2.23	13.75	3.89	1.97
Clust-EF	4.02	3.66	2.16	13.44	3.79	1.92
Interp	3.86	3.54	2.04	12.80	3.64	1.79
ANS, 2d	3.71	3.56	1.86	12.58	3.67	1.69

Experimental results: compression effectiveness

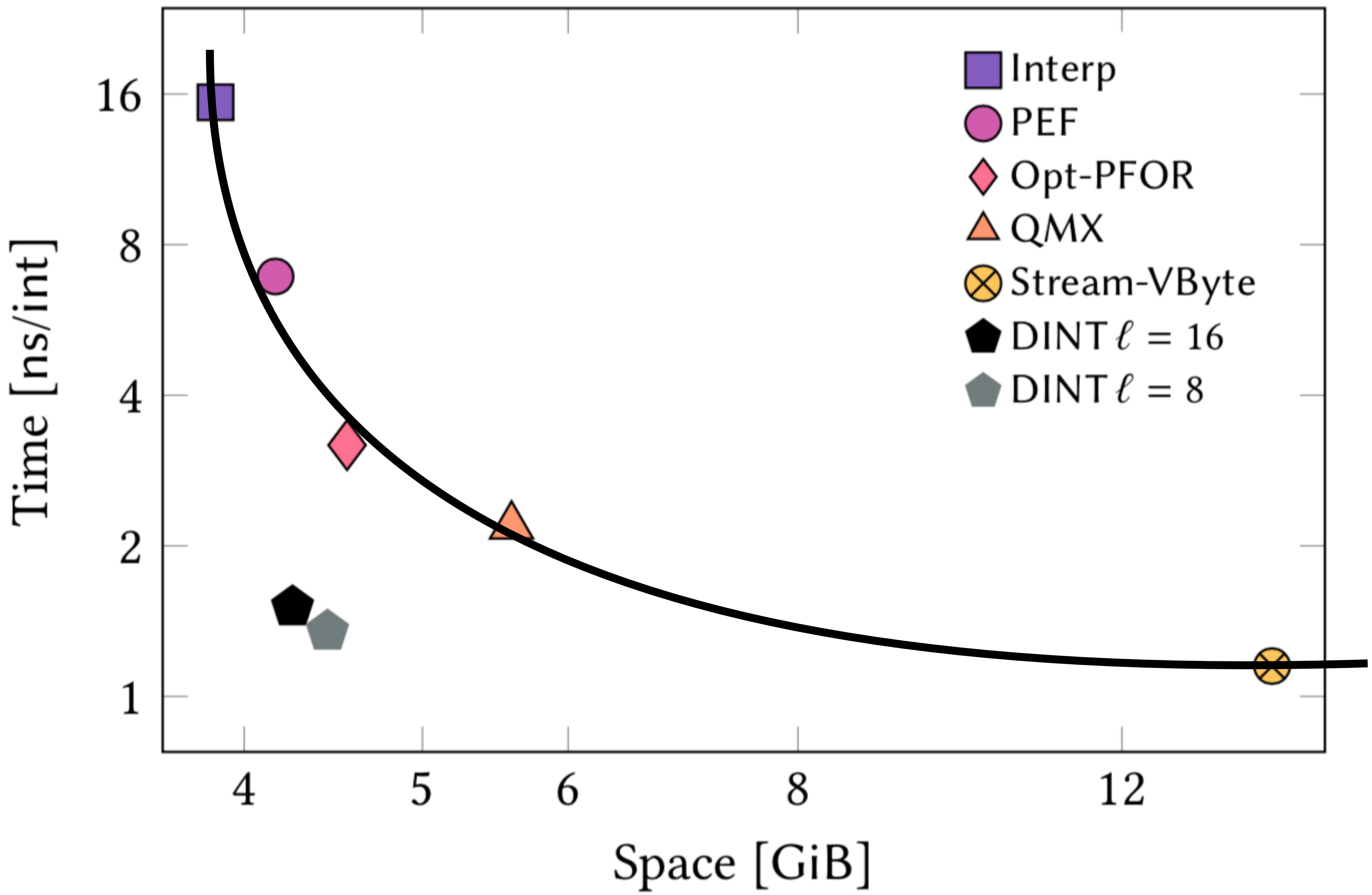
Method	Gov2			CCNEWS		
	GiB	docids	freqs	GiB	docids	freqs
Varint-GB	14.48	11.04	10.04	48.68	10.72	10.01
Varint-G8IU	12.77	9.90	8.69	43.87	9.75	8.93
VByte	11.85	9.22	8.02	39.65	8.88	8.00
QMX	5.59	4.99	3.11	19.20	5.12	3.04
Simple16	5.28	4.84	2.81	16.85	4.70	2.46
Opt-PFOR	4.55	4.33	2.26	15.50	4.49	2.10
DINT	4.29	4.22	1.98	15.09	4.48	1.92
PEF	4.16	3.85	2.23	13.75	3.89	1.97
Clust-EF	4.02	3.66	2.16	13.44	3.79	1.92
Interp	3.86	3.54	2.04	12.80	3.64	1.79
ANS, 2d	3.71	3.56	1.86	12.58	3.67	1.69

$\ell = 16$
$b = 16$

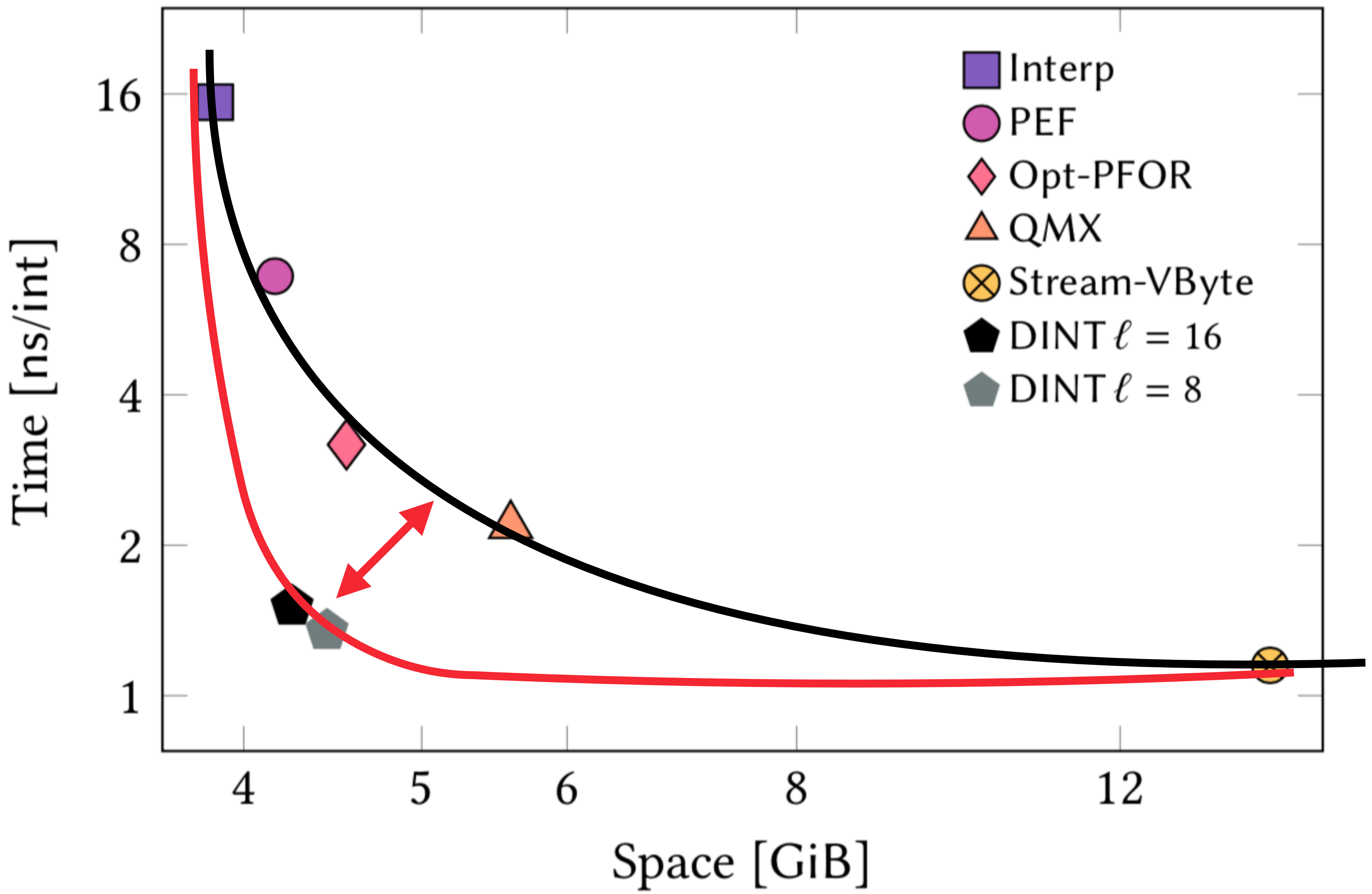
Experimental results: effectiveness/efficiency plot



Experimental results: effectiveness/efficiency plot



Experimental results: effectiveness/efficiency plot



Further readings

Chapter **6** and **7** of my Ph.D. thesis.

(more datasets, comparisons, query timings)

http://pages.di.unipi.it/pibiri/papers/phd_thesis.pdf

Thanks for your attention, time, patience!

Any questions?