Adapting STAR Code for Non-Volatile Memory Systems

Rui Chen, Lihao Xu

Wayne State University

Outline

- Motivation
- What is STAR Code?
- Why STAR Code?
- How to adapt STAR Code?
- Conclusion

Motivation

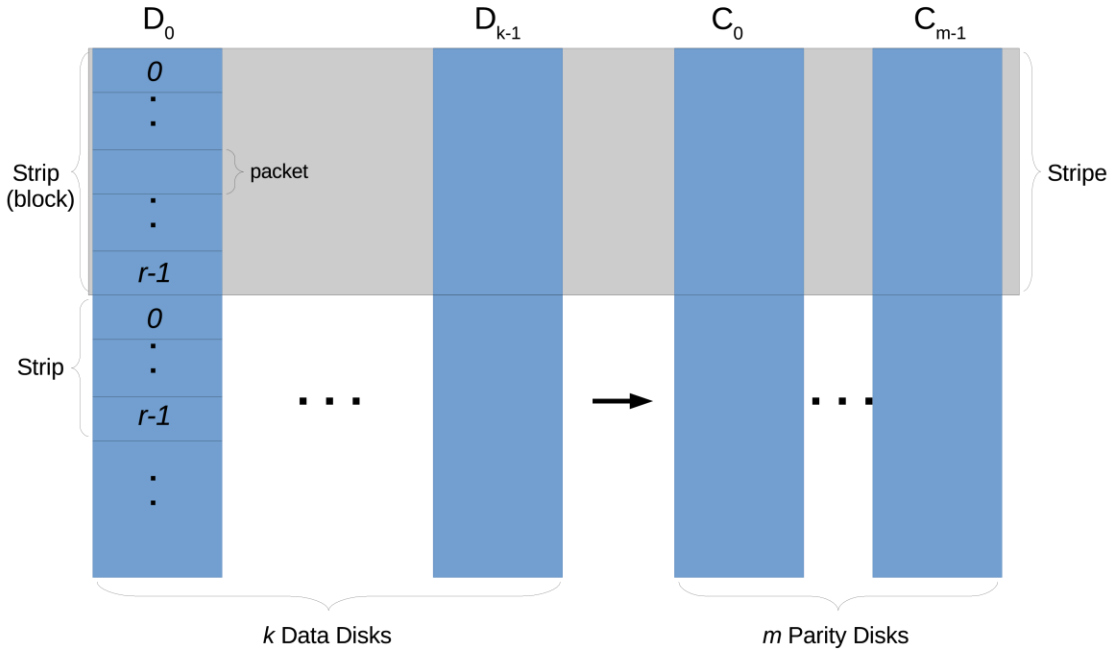
- With NVM emergence, I/O throughput can reach 10s of GB/s instead of GB/s
- Requires higher performance in erasure codes, rather than traditional RS-Code

What is STAR Code?

- One kind of Erasure Codes

Erasure Code

- Structure for a typical (k, m) erasure code
- One storage unit here can be one NVM chip/disk/node



Erasure Code

- Matrix-based erasure codes
  - Reed-Solomon Code

- Array-based erasure codes
  - STAR Code

Reed-Solomon Code

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ x_{0,0} & x_{0,1} & x_{0,2} & x_{0,3} \\ x_{1,0} & x_{1,1} & x_{1,2} & x_{1,3} \end{bmatrix}}_{G^T} * \underbrace{\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix}}_{\text{Data}} = \underbrace{\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ c_0 \\ c_1 \end{bmatrix}}_{\text{Codeword}} \begin{matrix} \} \text{Data} \\ \\ \\ \\ \} \text{Parity} \\ \\ \end{matrix}$$

- An example for (4, 2) RS Code
- Multiplications done by discrete logarithm tables --- computationally expensive

STAR Code

	Data columns					Parity column III		
	A	E	D	C	B	X	X	A
	B	A	E	D	C	X	X	B
	C	B	A	E	D	X	X	C
	D	C	B	A	E	X	X	D
imaginary row	E	D	C	B	A	X	X	E

- Uses only XOR operations
- Can tolerate up to 3 erasures
- A example of (5, 3) STAR Code

Why STAR Code?

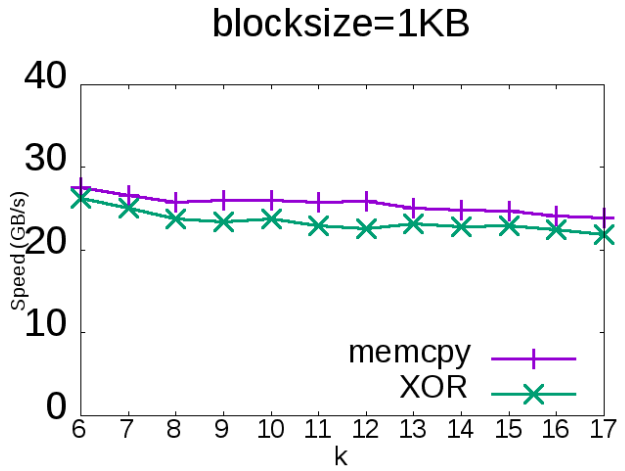
- Performance evaluation through experiments

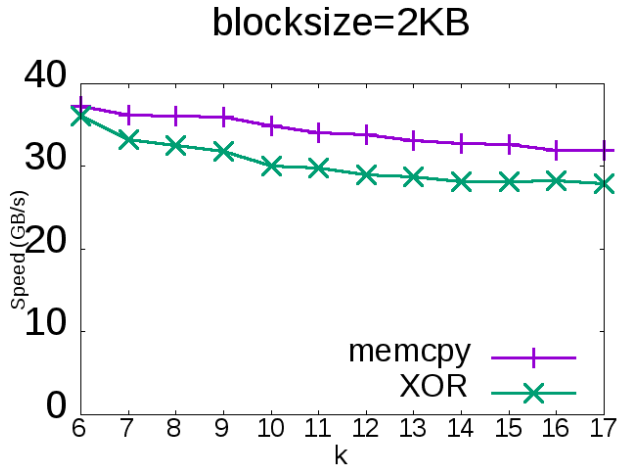
Experiment Setup

- Lenovo ThinkCentre M900
  - i5-6500（L1: 256KB, L2: 1MB, L3: 6MB)
  - Ubuntu 16.04.3
  - gcc 5.4.0 (-O3)

Baseline

- memcpy & XOR
- blocksize=1KB/2KB
- 1000 stripes (codewords)
- k: 6 – 17





Testing Libraries

- RS-Code
  - Jerasure 2.0 (C)
  - Intel's ISA-L 2.17 (C & asm)
  - both includes Intel's SSE
- STAR Code
  - own implementation (C)
  - using Intel's SSE

How to adapt STAR Code?

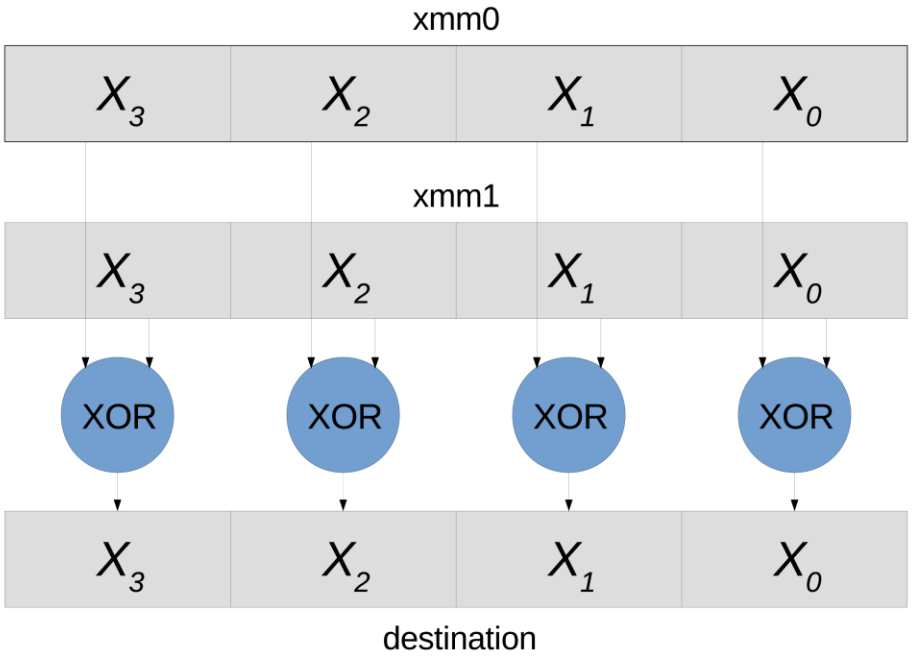
- Intel's SSE
- Parameter Configuration
  - p=17
  - adapting blocksize & k

Intel's SSE

- Jerasure & ISA-L already include SSE
- 8 128-bit general-purpose registers

  -XMM0 to XMM7

  -each consists of 4 32-bit floating-point numbers

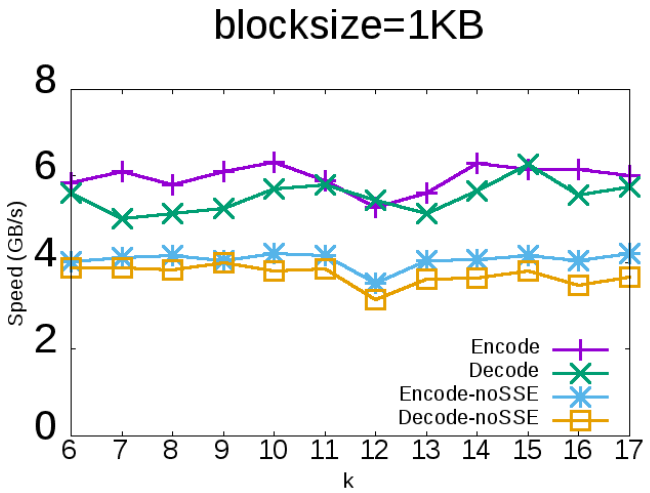


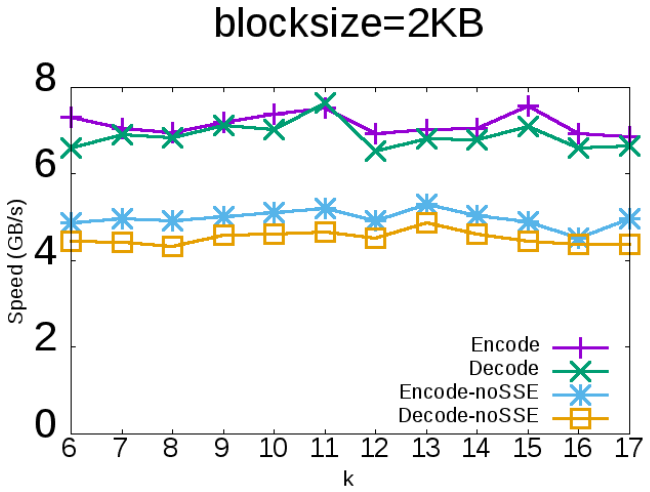
Results

- Adapting SSE
- Encoding
- Decoding

Adapting SSE

- blocksize=1KB/2KB
- 1000 stripes
- k: 6-17
- 3 erasures for decoding



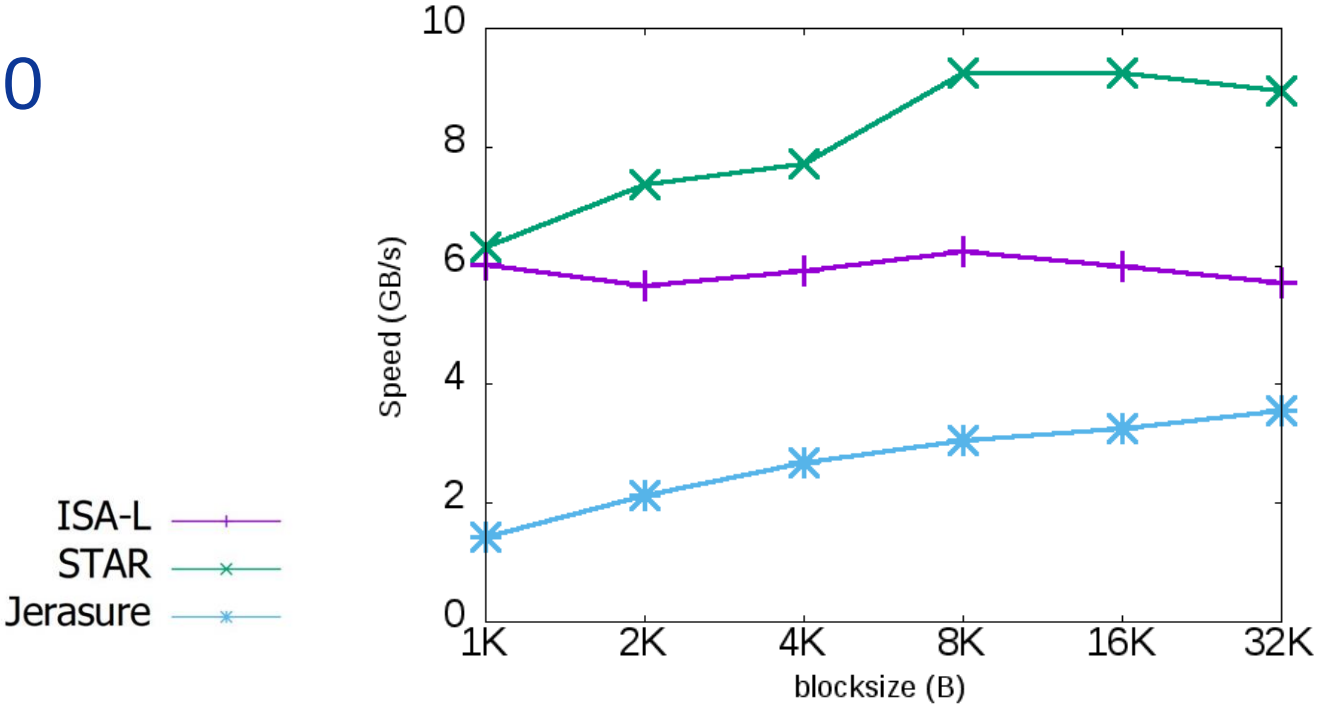


Encoding Performance

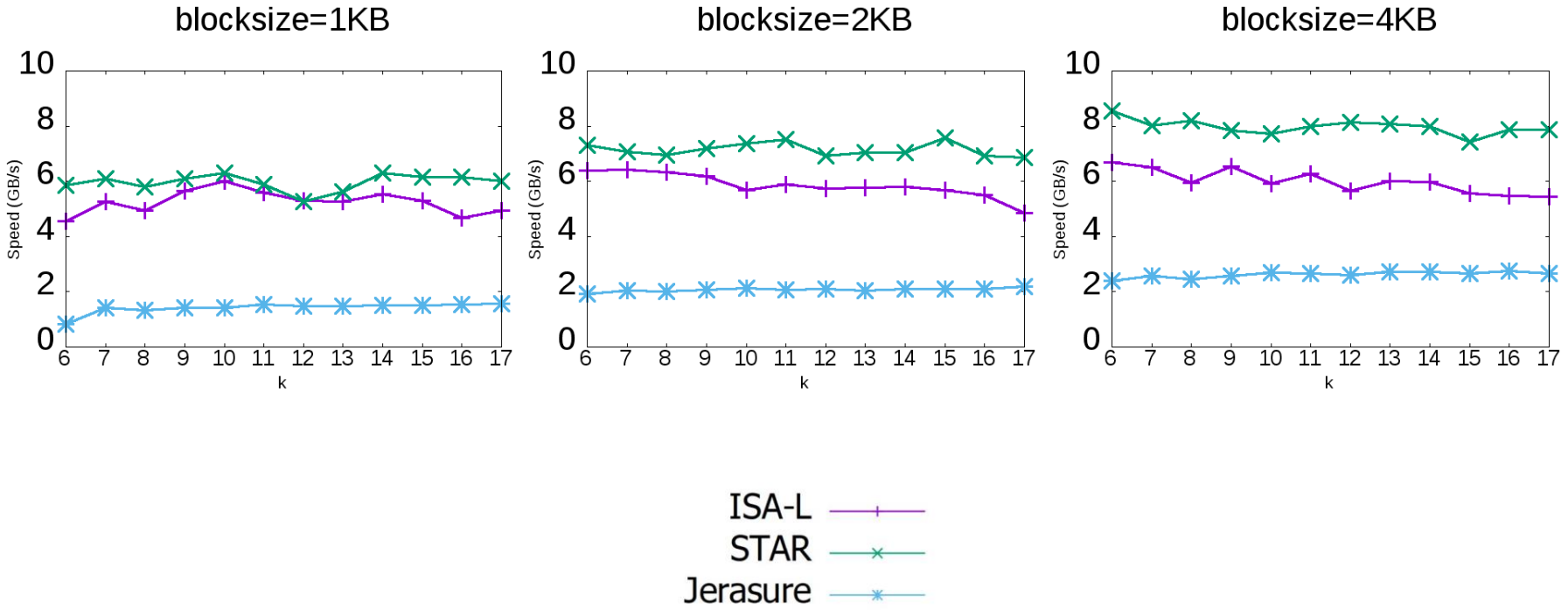
- 1000 stripes (codewords)
- m=3
- w=8 for RS-Code
- p=17 for STAR

Adapting blocksize

- k=10



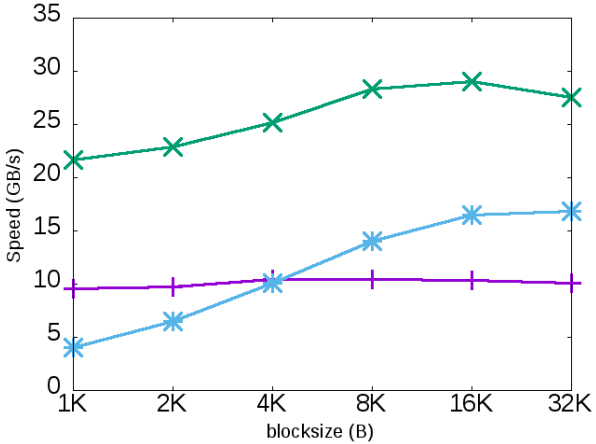
Adapting k


blocksize=1KB
blocksize=2KB
blocksize=4KB
Speed (GB/s)
k
0
2
4
6
8
10
6
7
8
9
10
11
12
13
14
15
16
17
ISA-L
STAR
Jerasure

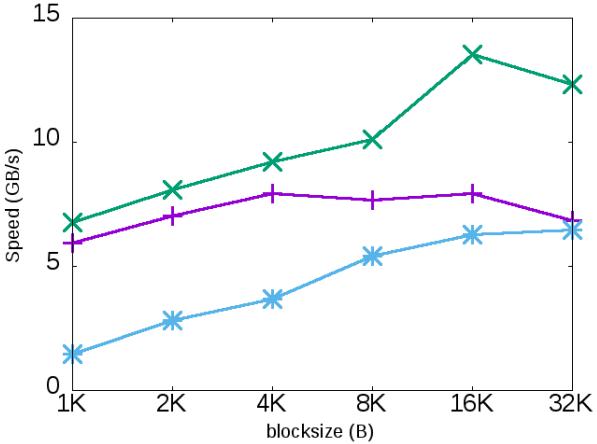
Decoding Performance

- 1000 stripes (codewords)
- m=1/2/3 respectively
- w=8 for RS-Code
- p=17 for STAR

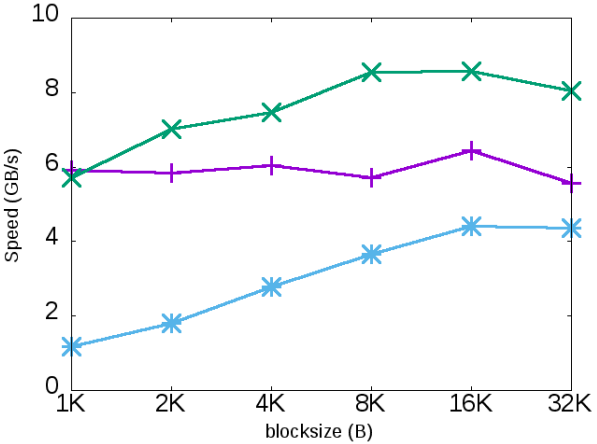
Adapting blocksize with k=10



m=1



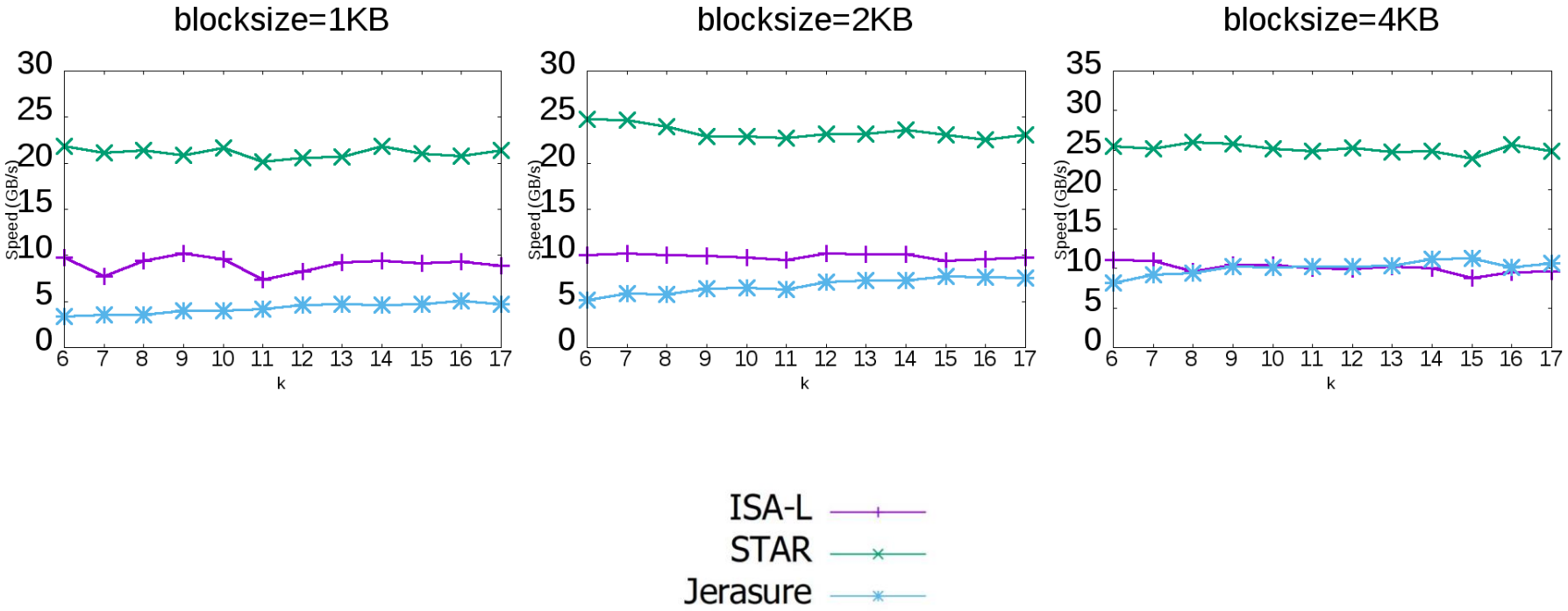
m=2



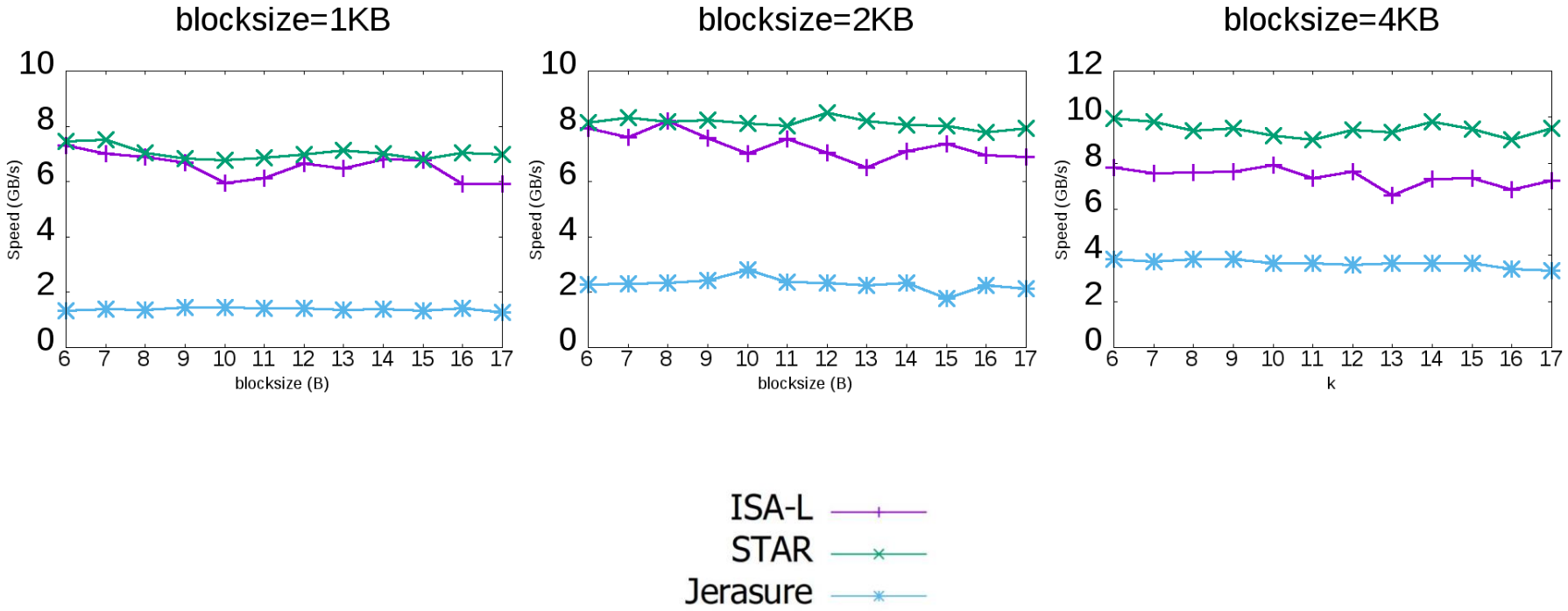
m=3

ISA-L
STAR
Jerasure

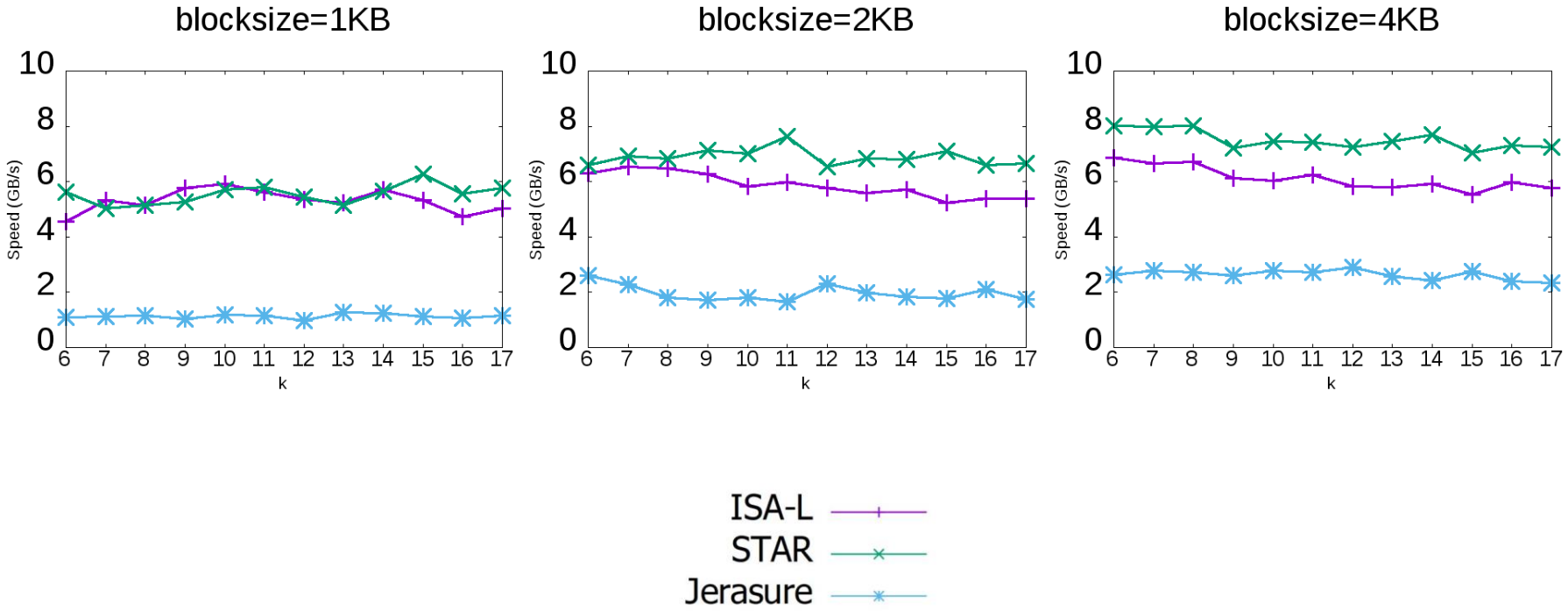
Adapting k with m=1



Adapting k with m=2



Adapting k with m=3



Conclusion

- Intel's SSE
- Jerasure vs. ISA-L
- STAR vs. RS-Code
- Parameter configuration guideline

Questions?

Thank you!