Memory Access Analysis

Andes Technology

Outline

- Tiling Memory Access – Recap
- Improvement of Matrix Multiplication
  ① Transpose Instruction Proposal
- Suggest 3 Metrics for
  ① Compute Intensity (MACs**/data size**)
  ② Compute Cost (MACs**/u-Arch Cost(E.g. VRF R/W ports)**)
  ③ Compute Per-$-Locality (term to revise) (MACs**/# of $-DRAM Access**)
- Revised Profiles for Option A~E

Tile-based Memory Access - Recap

- "**Minimizing memory transactions** while maximizing parallelism and **MAC utilization** are central to any effective solution"
- "In the figure, slices are represented as light/dark gray areas, and each **red dot represents a (128B) memory burst access** to the DRAM"

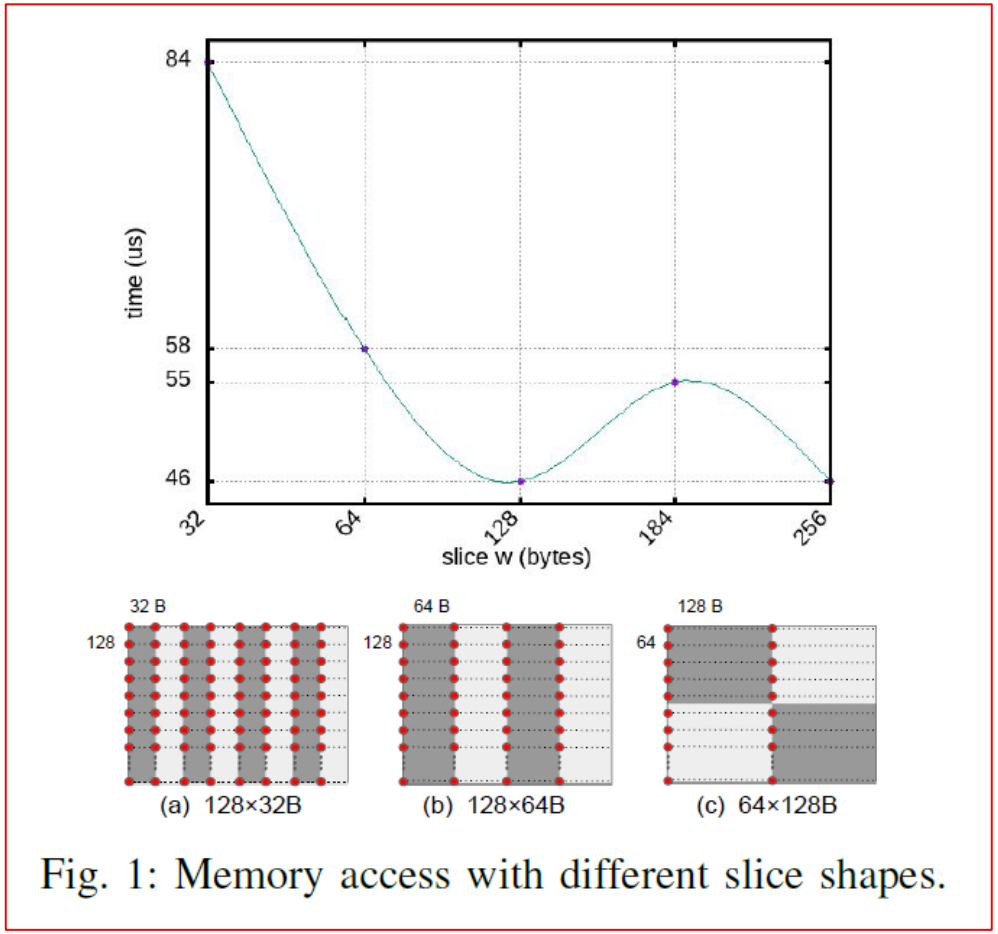


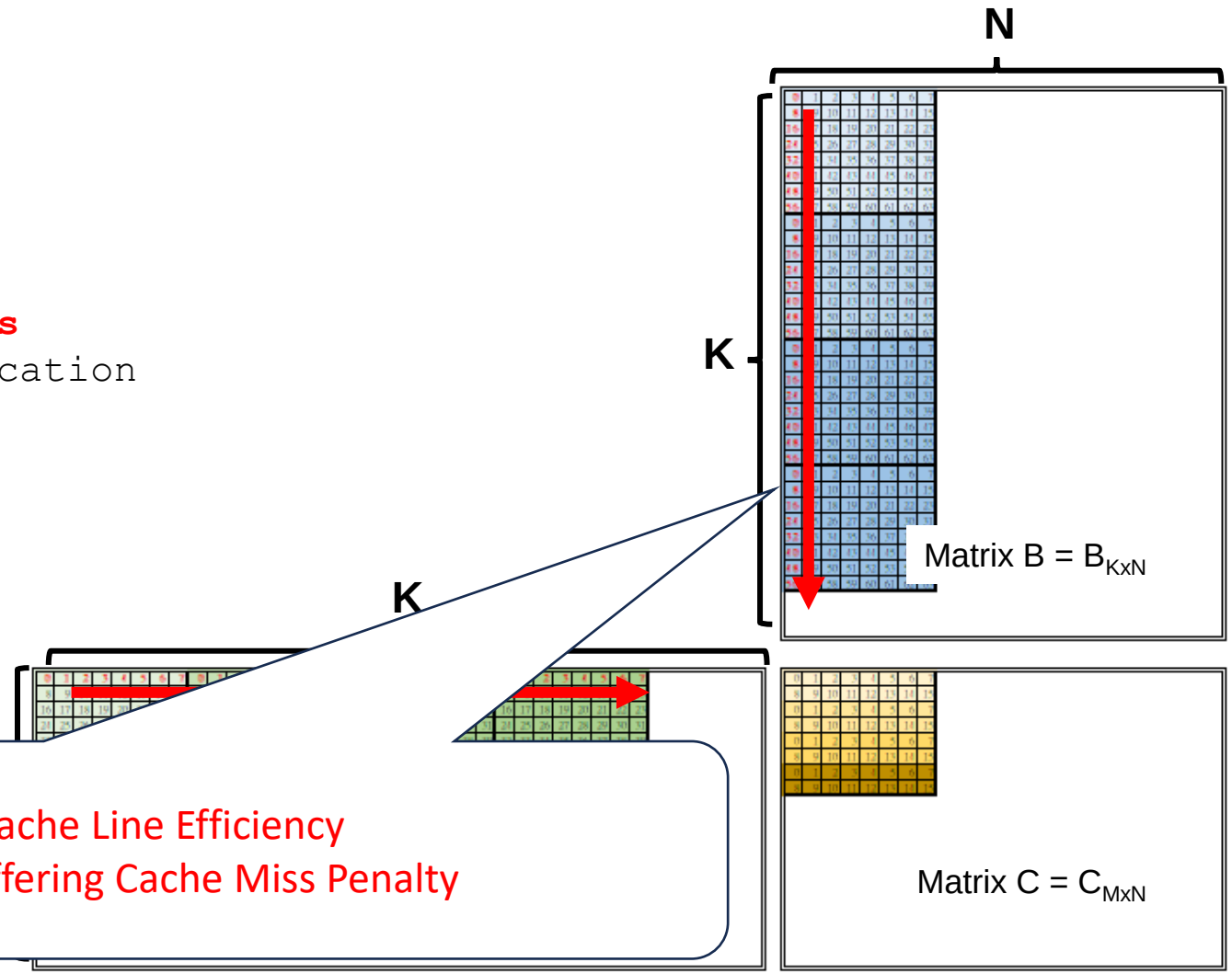
Fig. 1: Memory access with different slice shapes.

Courtesy of "Efficient Tensor Slicing for Multicore NPUs using Memory Burst Modeling"
https://drive.google.com/open?id=1ODJuxwnJHrYOCE6EXeeZauNqLPyz3Lcc&usp=drive_fs

Naive Tiles-based Matrix Multiplication (Option E, int32 += int8*int8)

$C_{MxN}=A_{MxK}*B_{KxN}$

```
void kernel() {
…
 while(k>0){
  vld.2d va,[mem_a];//Row Major Access
  vld.2d vb,[mem_b];//Columm Major Access
  amm vc, vb, va; //Tile matrix multiplication
  k-=Ktile;
 }
…
}
```

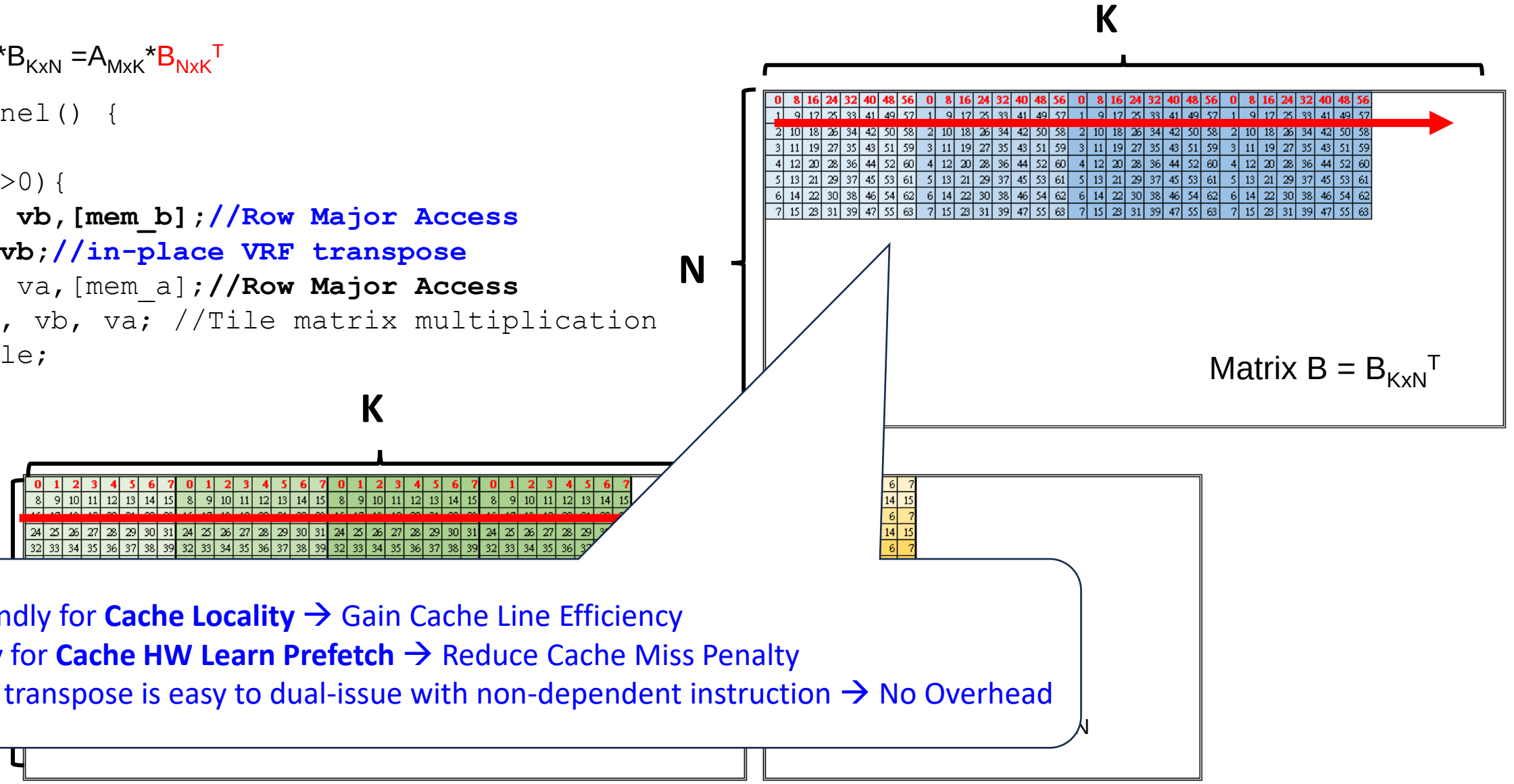


(1) Non-friendly for **Cache Locality** → Suffering Cache Line Efficiency
(2) Non-Easy for **Cache HW Learn Prefetch** → Suffering Cache Miss Penalty

Efficient Tiles-based Matrix Multiplication (Option E, int32 += int8*in8)

$C_{MxN}=A_{MxK}*B_{KxN}=A_{MxK}*B_{NxK}^{T}$

```
void kernel() {
…
 while(k>0){
  vld.2d vb,[mem_b];//Row Major Access
  vt vb vb;//in-place VRF transpose
  vld.2d va,[mem_a];//Row Major Access
  amm vc, vb, va; //Tile matrix multiplication
  k-=Ktile;
 }
…
}
```



Proposed In-place VRF Transpose Instruction (Option E, VLEN1024, sew = 0)

```
#Destination(vd) = Transpose of Input(vs)
```

$vd_{mxn} = vs_{mxn}^{T}$

```
#Destination(vd),source(vs), m row element number(r1)
vte8.vv vd, vs, r1 # r1 for m elements
vte16.vv vd, vs, r1 # r1 for m elements
vte32.vv vd, vs, r1 # r1 for m elements
```

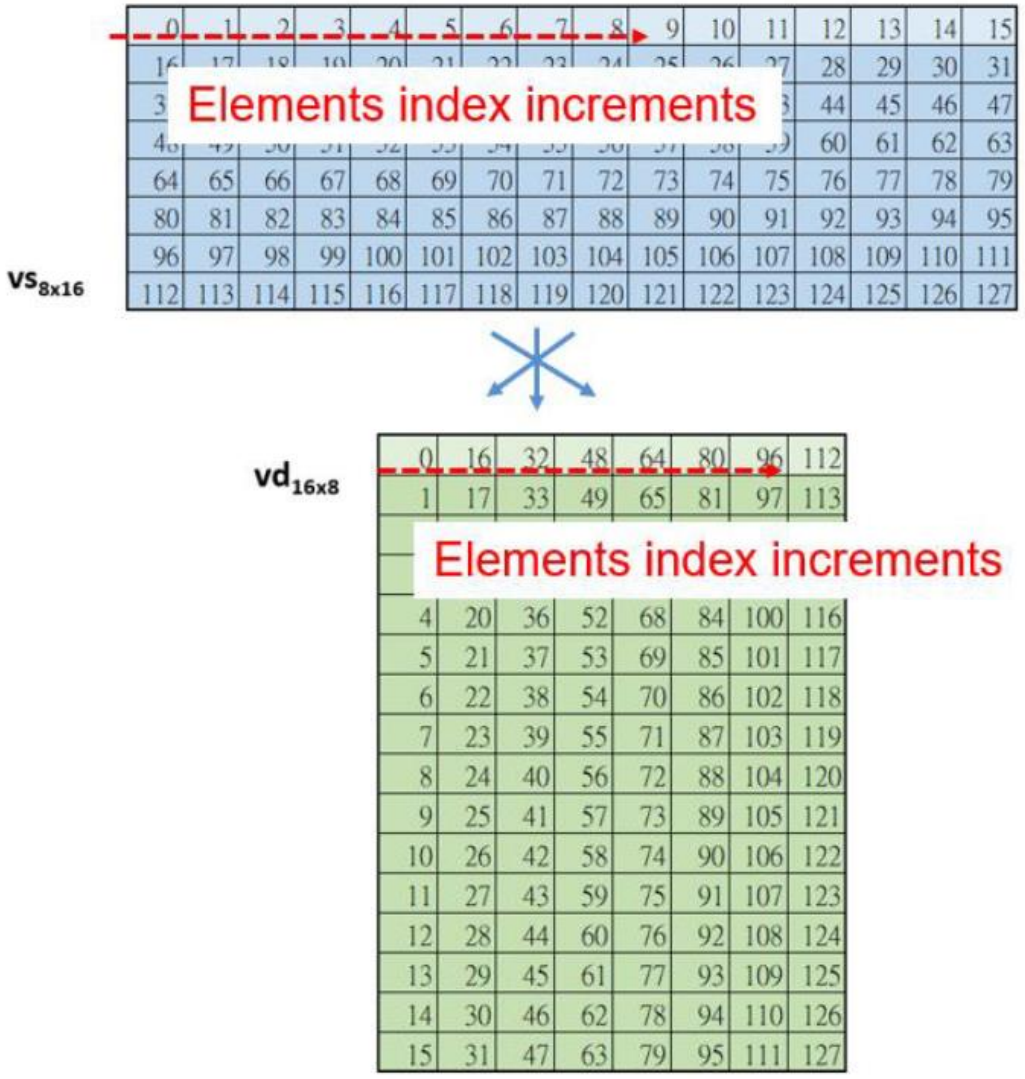
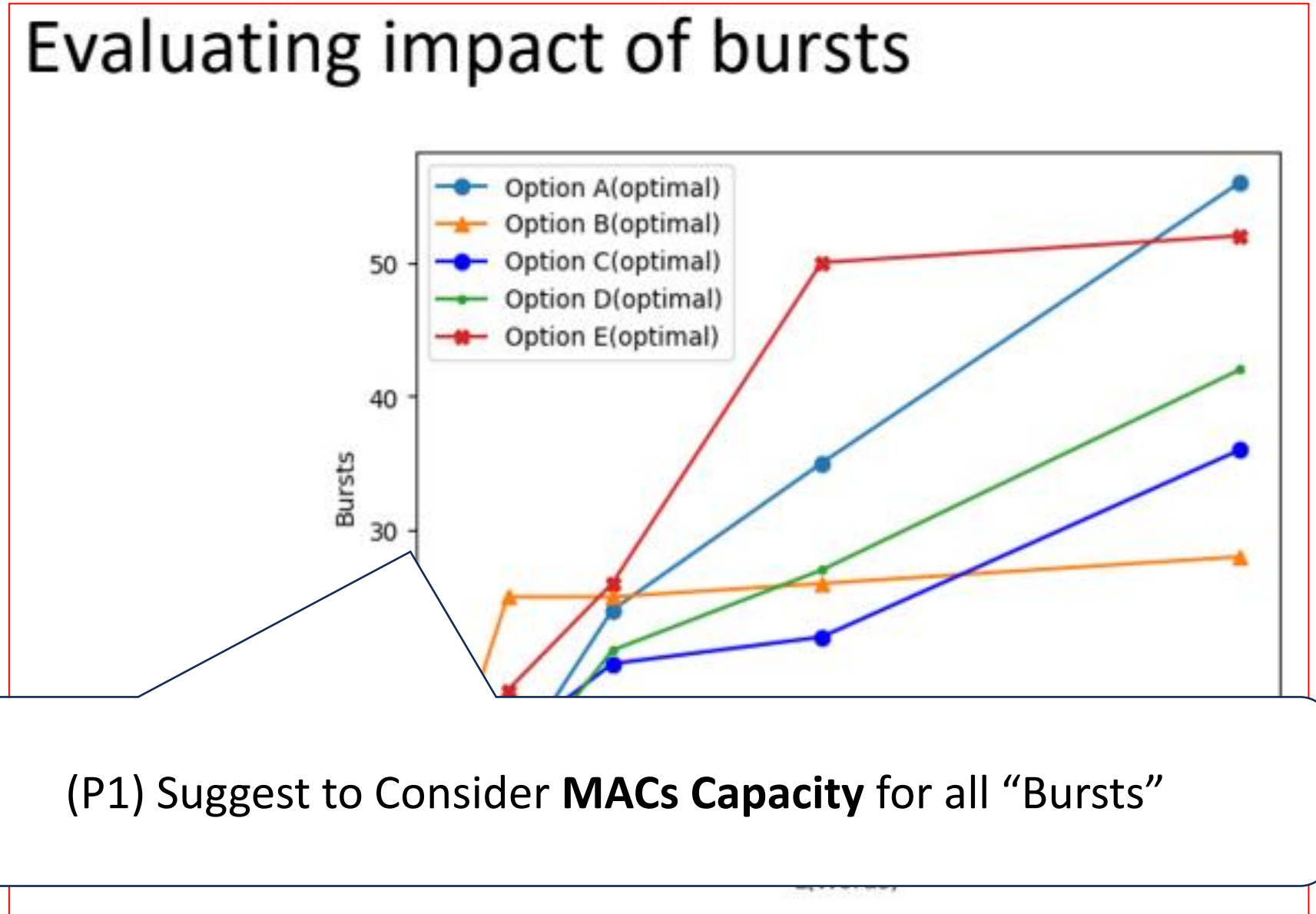


Figure 25. Example illustrates for VLEN 1024, sew 8, m = 16 row elements

Burst Analysis – Rcap (1/)

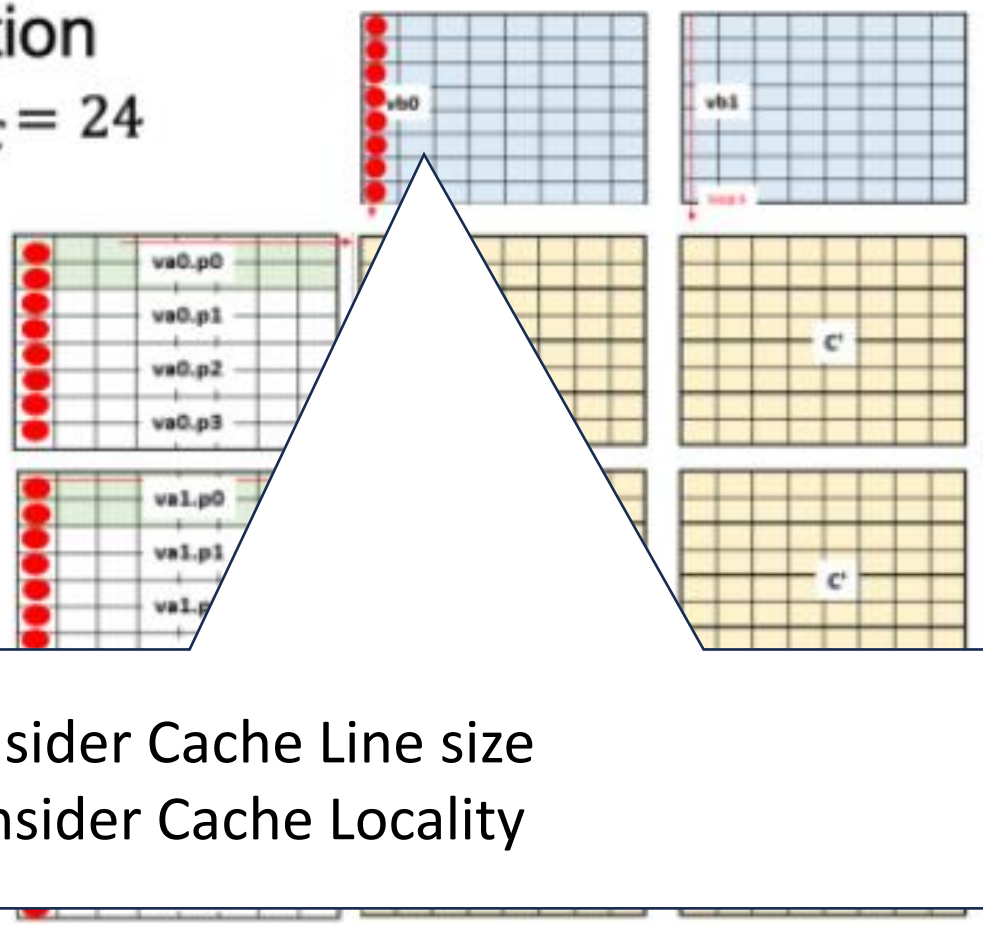


(P1) Suggest to Consider **MACs Capacity** for all “Bursts”

Burst Analysis – Rcap (2/)

Tiling and Memory Bursts: Option E

- Option E: Variable matrix representation
  - Configuration: $SRC_A = 3, SRC_B = 2, ACC_C = 24$
  - $L_{min} = 4$
- Bursts
  - A: $SRC_A \cdot \frac{L}{\sqrt{L_{min}}}$
  - B: $\sqrt{L_{min}}$
  - For this configuration
    -



(P2) Suggest to Consider Cache Line size
(P3) Suggest to Consider Cache Locality

Suggested 3 Metrics for Performance Profiling

① Compute Intensity (MACs/load size)

② Compute Cost (MACs/u-Arch Cost(E.g. VRF R/W ports)

③ Compute Per-$-Locality (term to revise) (MACs/# of $-DRAM Access)

Where # of $ Access is analyzed according to :

1. Assume Cache line Size 32~128 Bytes (general real case 64Bytes)
2. Assume No Cache Conflict (simplicity analysis as fully associated cash u-arch)
3. Assume Cache line aligned with tile boundary (simplicity analysis before simulator ready)
4. Assume Scenarios for cache prefetch/hit cases

Analysis Tool

- Available on https://github.com/CN-Ke/IME_Evaluation/blob/main/compute_locality.py
- You can config the cache line size ranges from 32,64,128 and check the results

Option A: Compute Per-$-Locality Illustration (VLEN512 fp32 ← fp32*fp32)

- $\widetilde{M}, \widetilde{N}$ Denotes Reasonable Unrolling Parameters

$\widetilde{N}$ =4 (L=16)

$\widetilde{M}$ =5 (L=16)

Effective N per-$-Locality†

Effective K per-$-Locality†

†: Cache Line Size 32Bytes, 8Words

Option B: Compute Per-$-Locality Illustration (VLEN512 fp32 ← fp32*fp32)

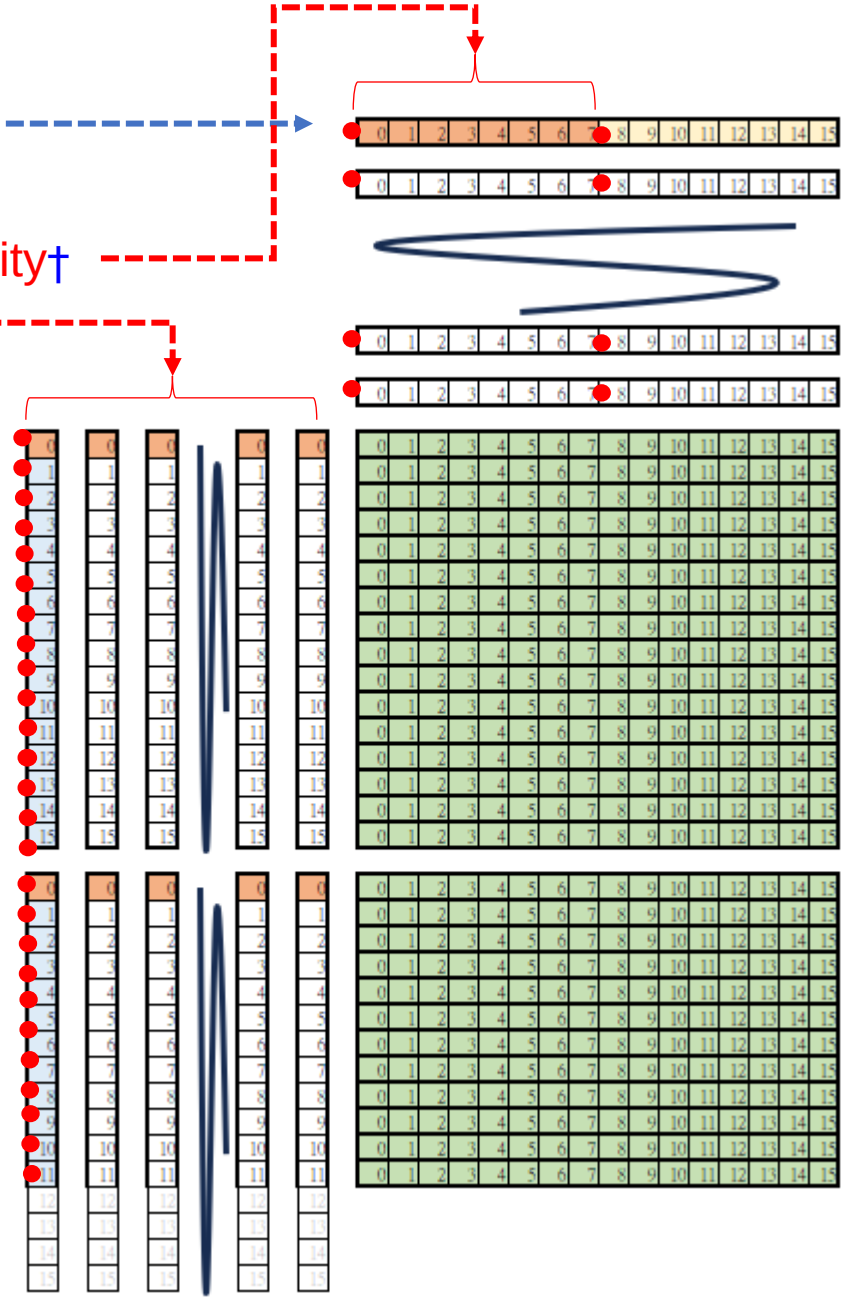
- $\widetilde{M}, \widetilde{N}$ Denotes Reasonable Unrolling Parameters

$\widetilde{N}$ =1
(L=16)

$\widetilde{M}$ =1.75
(L=16)

Effective N per-$-Locality†

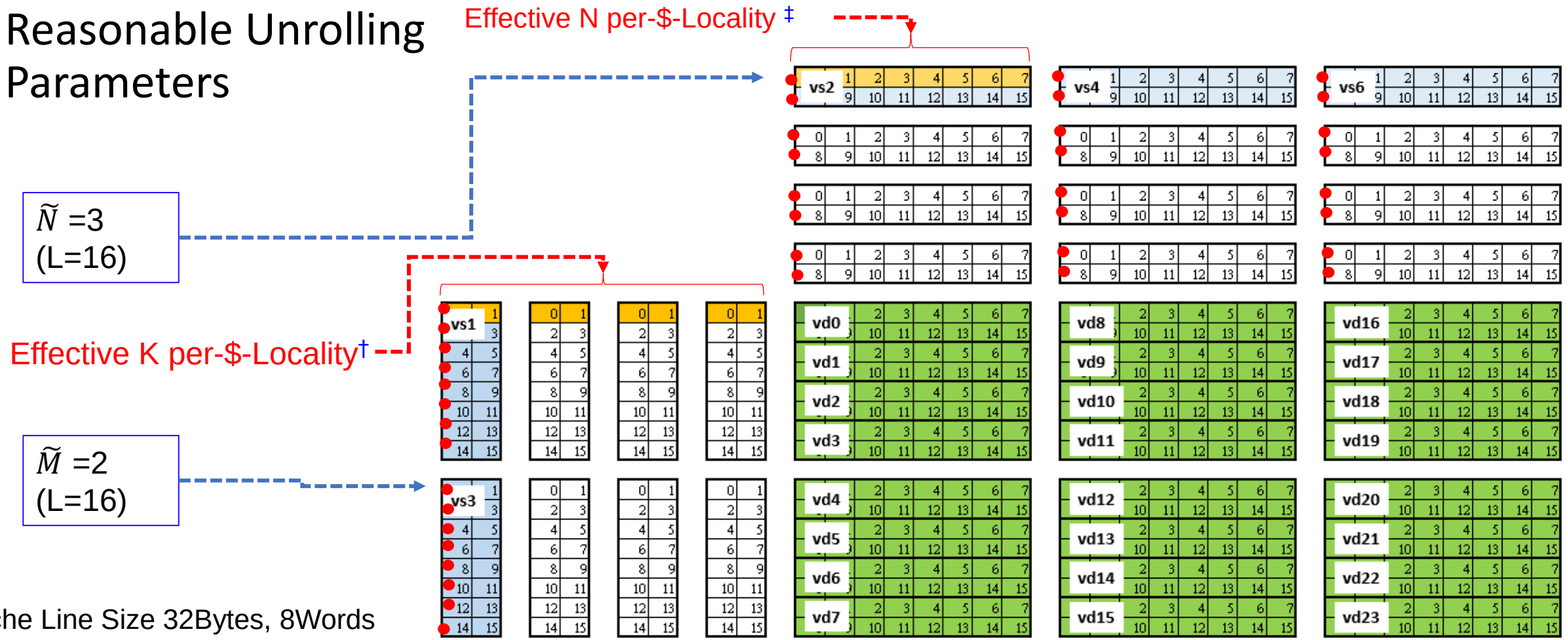
Effective K per-$-Locality†



†: Cache Line Size 32Bytes, 8Words

Option C: Compute Per-$-Locality Illustration (VLEN512 fp32 ← fp32*fp32)

- $\widetilde{M}, \widetilde{N}$ Denotes Reasonable Unrolling Parameters



†: Cache Line Size 32Bytes, 8Words

Option E: Compute Per-$-Locality Illustration (VLEN512 fp32 ← fp32*fp32)

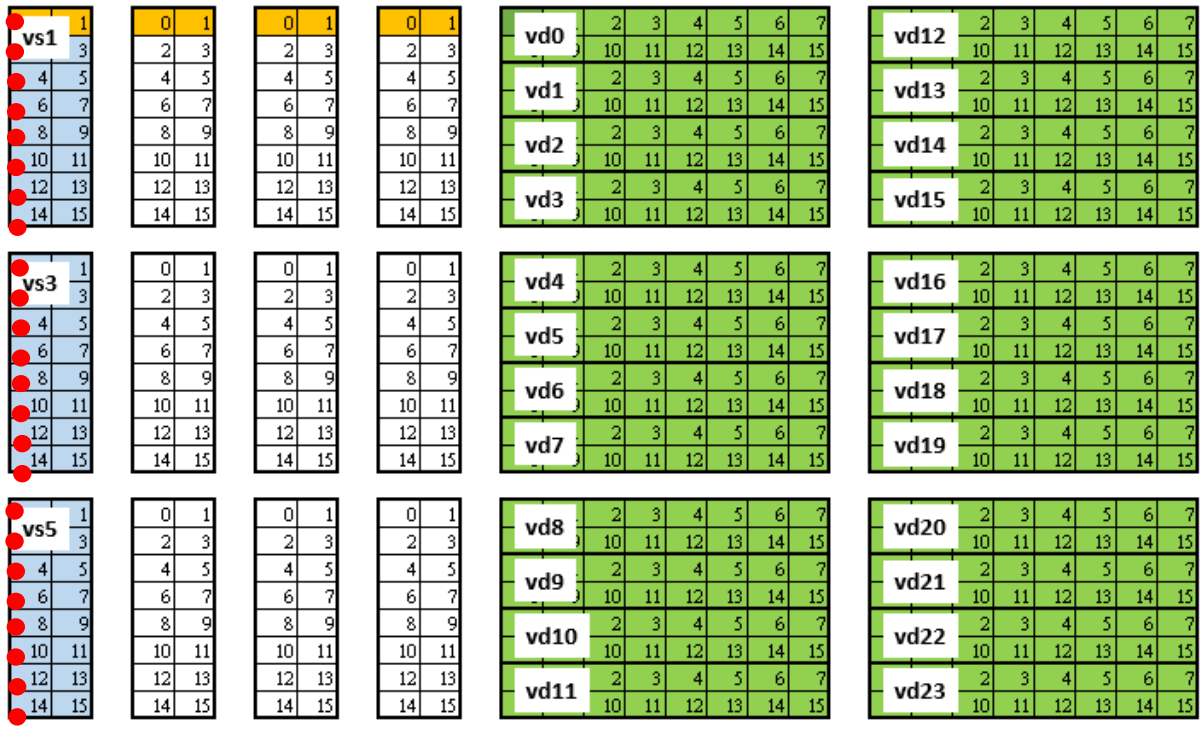
- $\widetilde{M}, \widetilde{N}$ Denotes Reasonable Unrolling Parameters

$\widetilde{N}$ =2
(L=16)

$\widetilde{M}$ =3
(L=16)

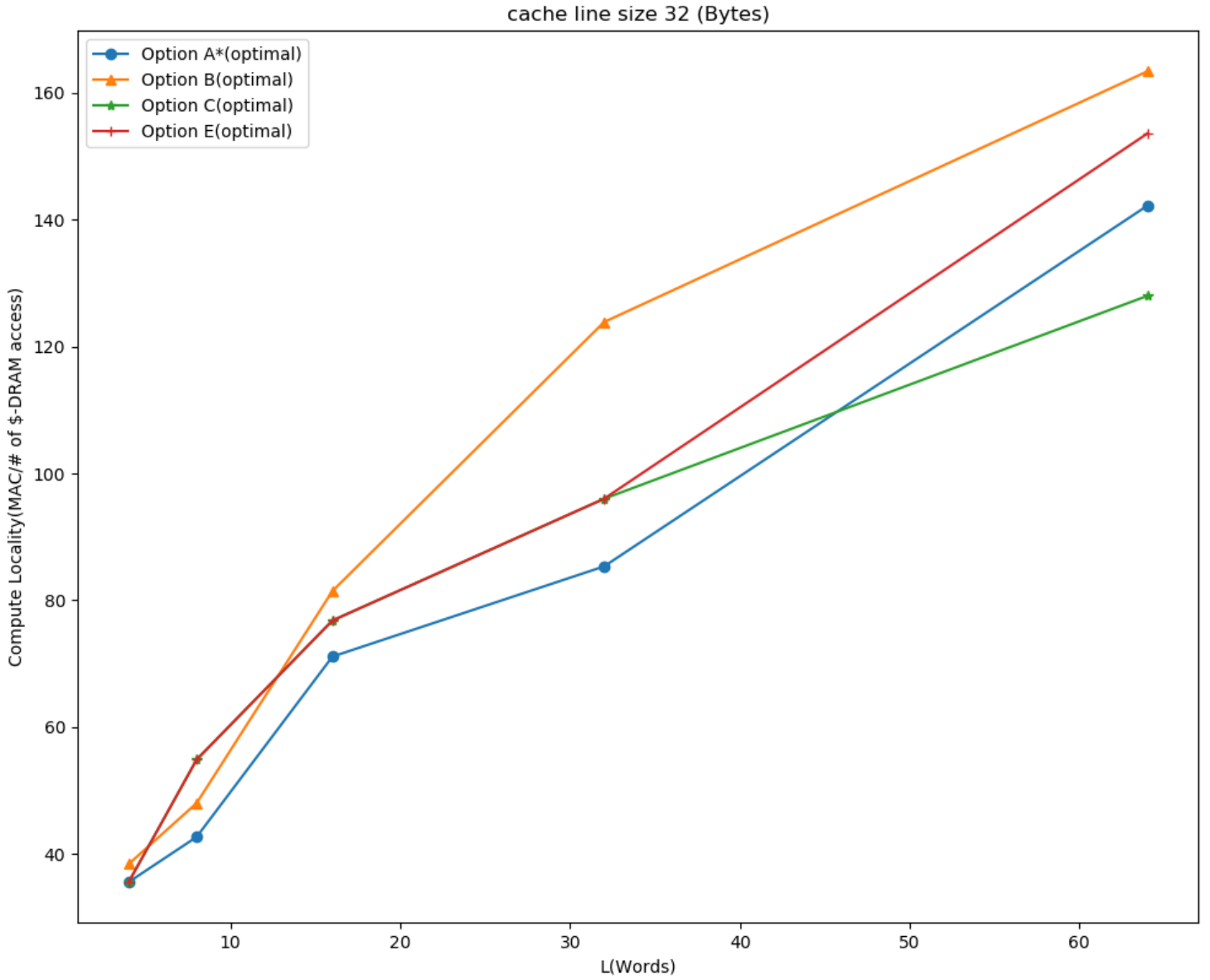
Effective K per-$-Locality ‡

Effective K per-$-Locality†

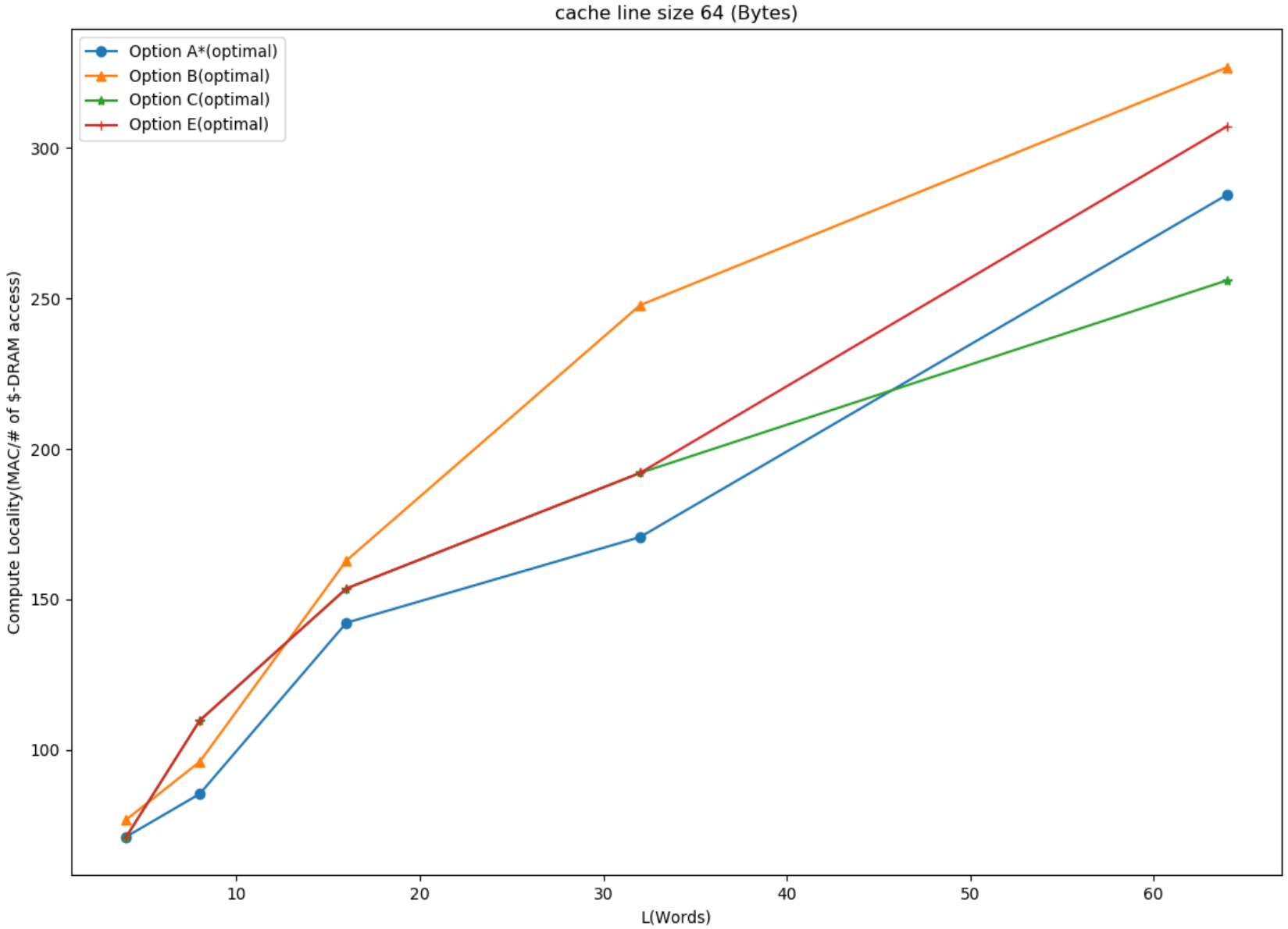


†: Cache Line Size 32Bytes, 8Words
‡: Consider $C_{MxN}=A_{MxK}*B_{KxN} =A_{MxK}*B_{NxK}^{T}$

Compute Per-$-Locality Comparison ($ line 32Bytes)



Compute Per-$-Locality Comparison ($ line 64Bytes)



Appendix