EPFL

EMBEDDED SYSTEMS LABORATORY

Accelerator-driven Data Arrangement to Minimize Transformers Run-time on Multi-core Architectures

Alireza Amirshahi, Giovanni Ansaloni, David Atienza

Embedded Systems Laboratory, EPFL, Switzerland

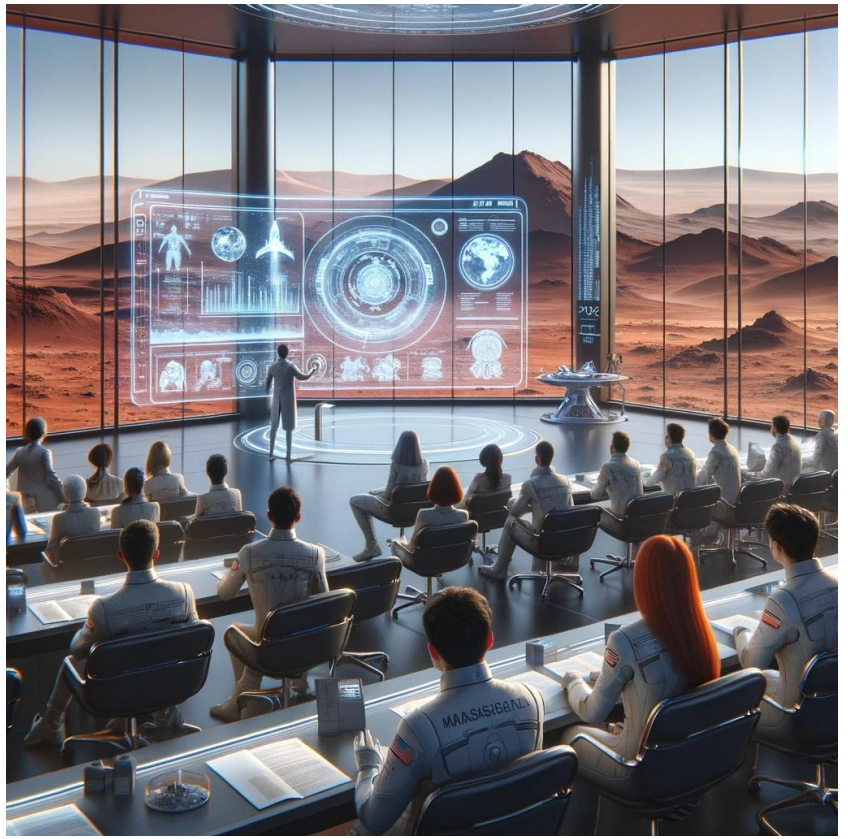
Applications of transformers

Input: *“A Ph.D. student is talking at a conference on the planet of Mars in the year 2124.”*

Text Generation:

The year is 2124, and humanity has made incredible strides in space exploration, establishing a presence on Mars. The conference is being held in one of the advanced research facilities …

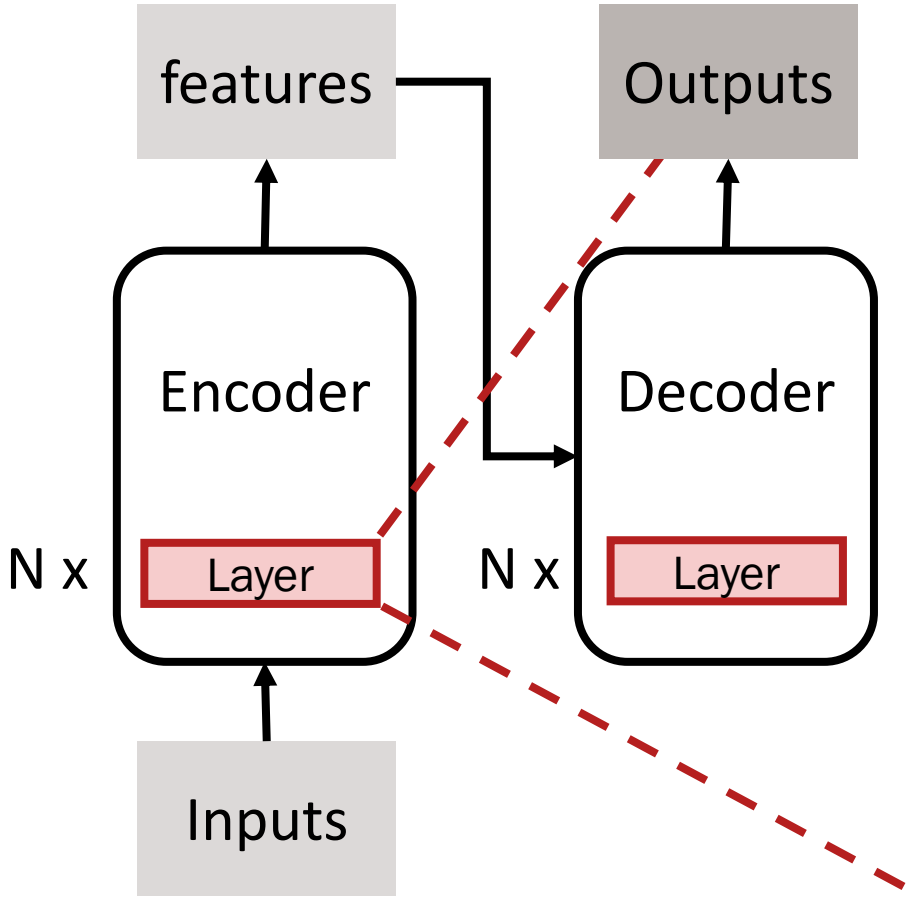
Image Generation [1]:

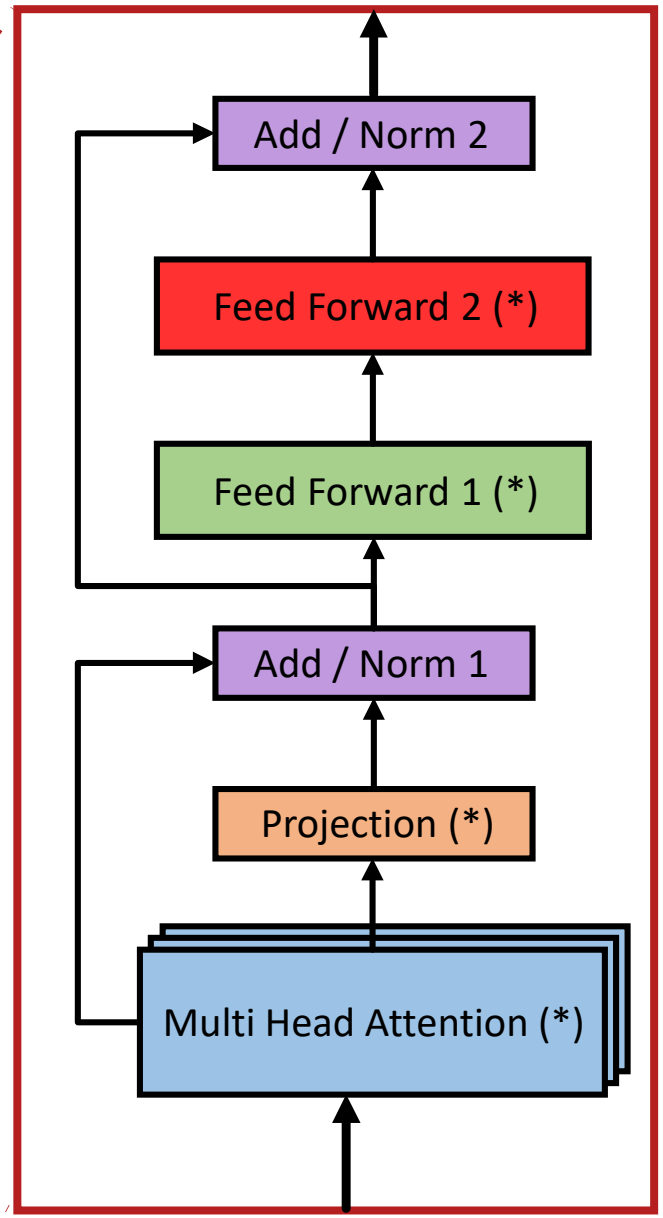


[1] A. Ramesh et al. ArXiv 2022

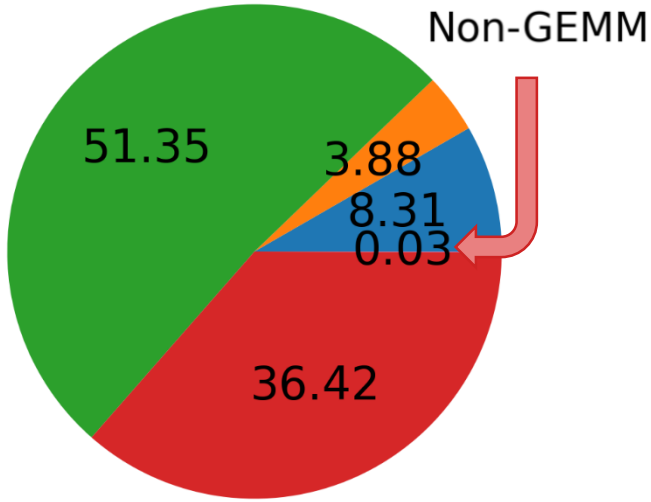
What is a transformer?

- An encoder-decoder structure





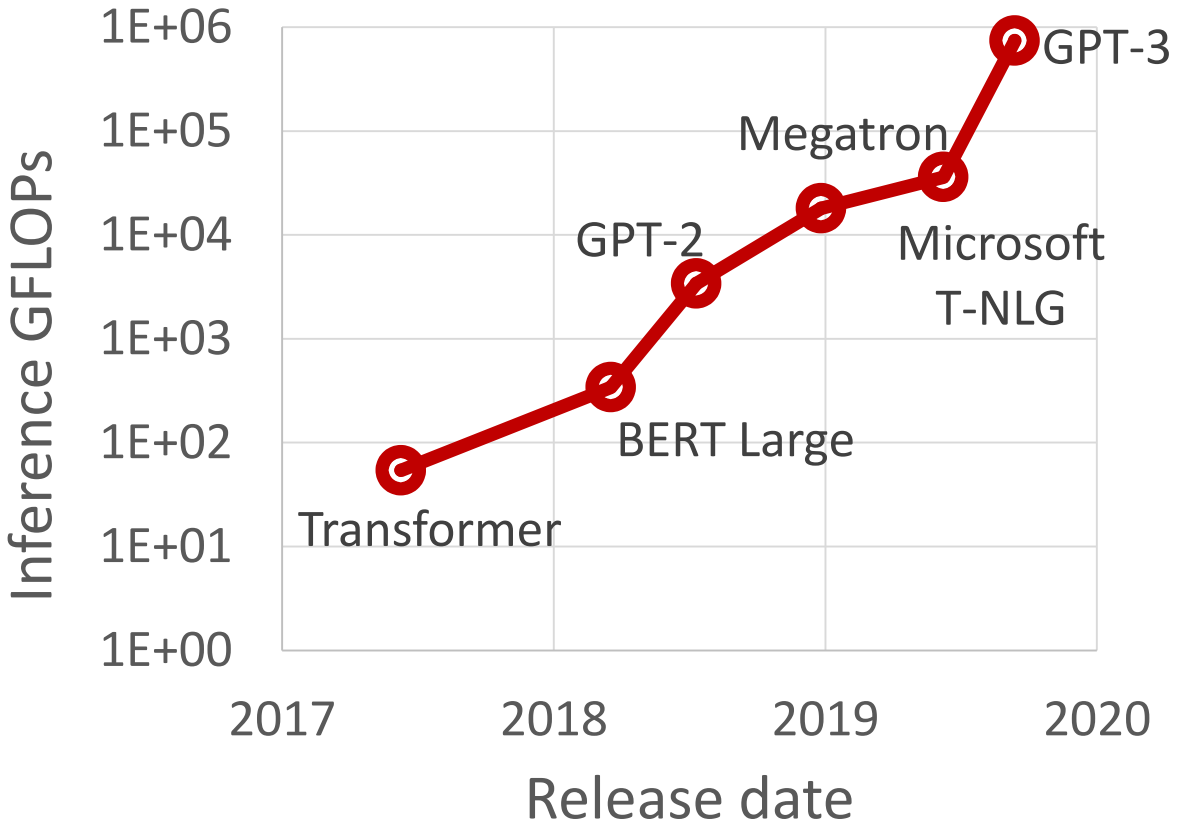
- Mostly General Matrix to Matrix Multiplication (GEMM)

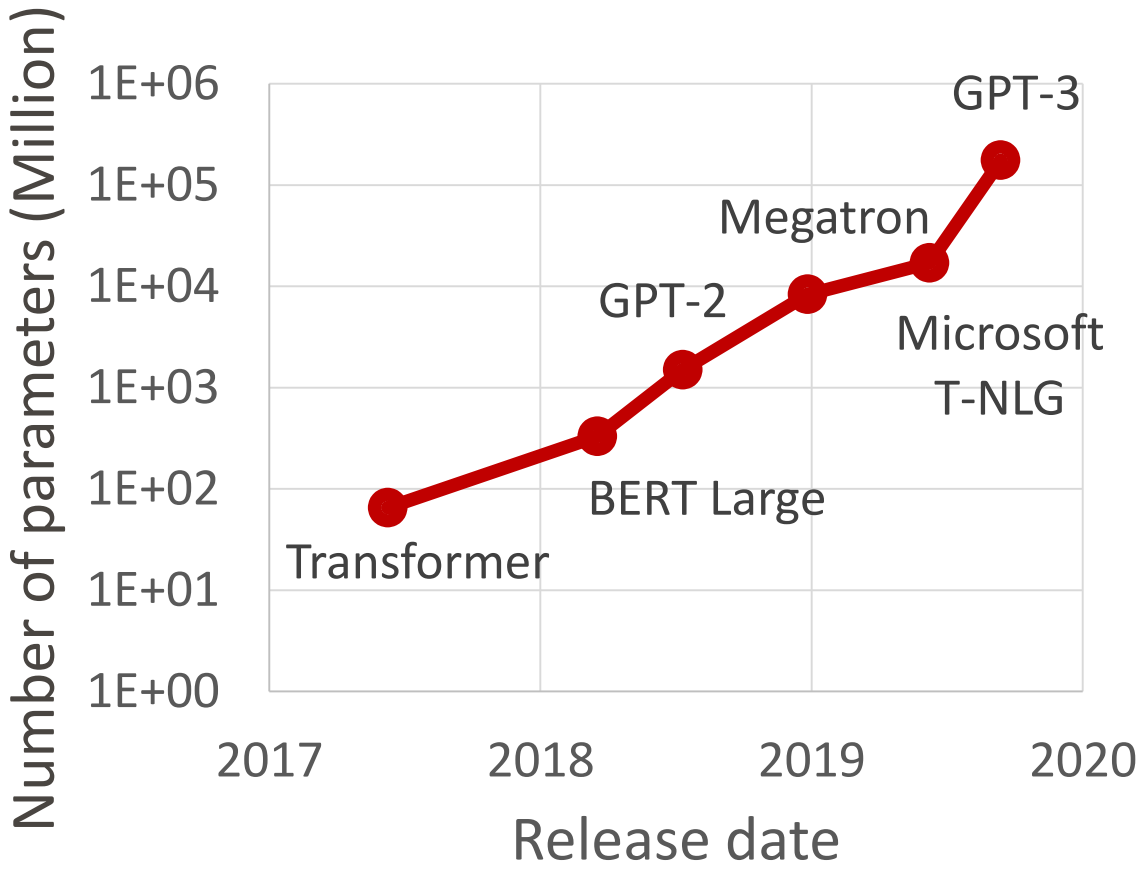


Execution time (%)

Problem definition

- Transformers are big [2]



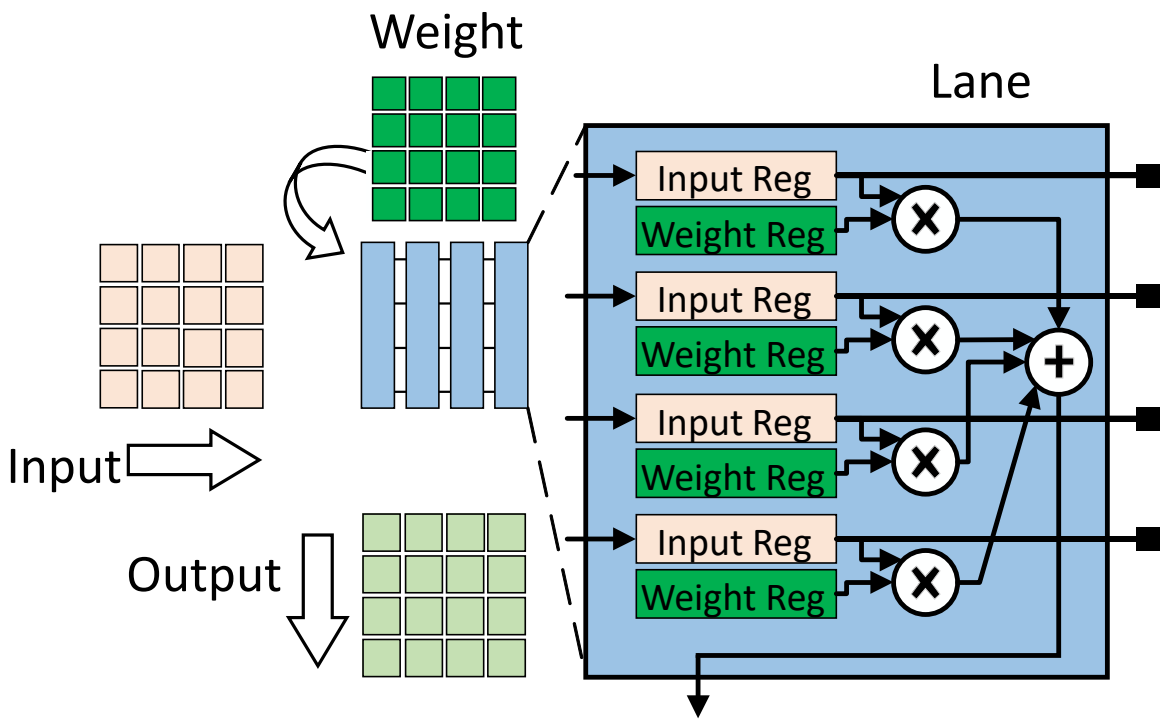


[2] A. Gholami et al. Riselab medium post 2021

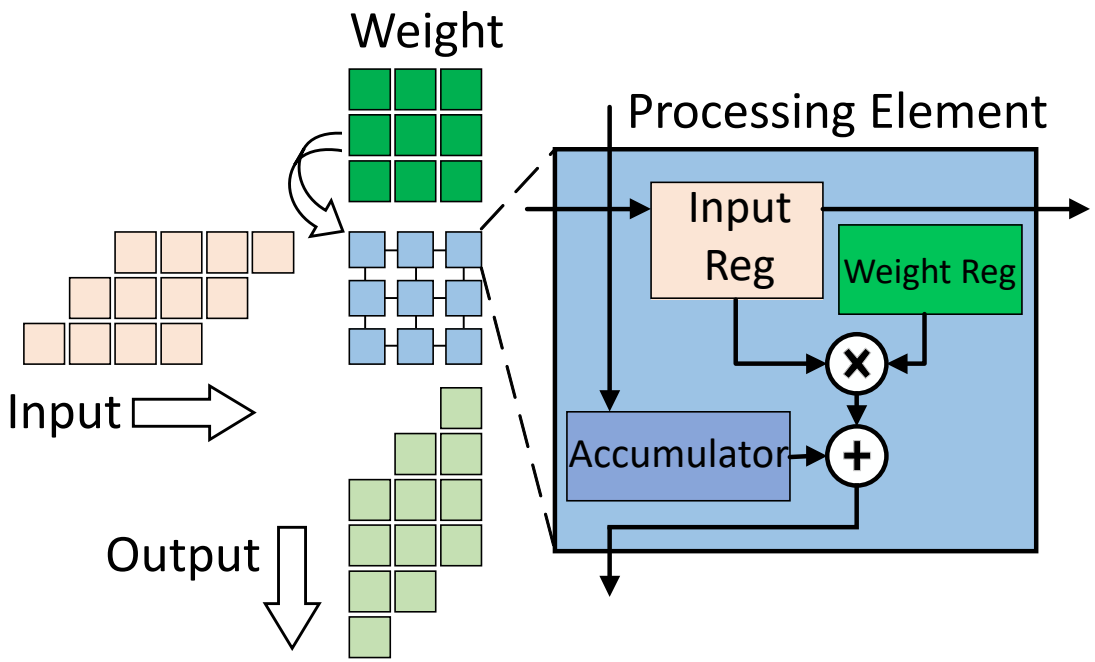
Need for GEMM accelerators and memory management for transformers

GEMM accelerators

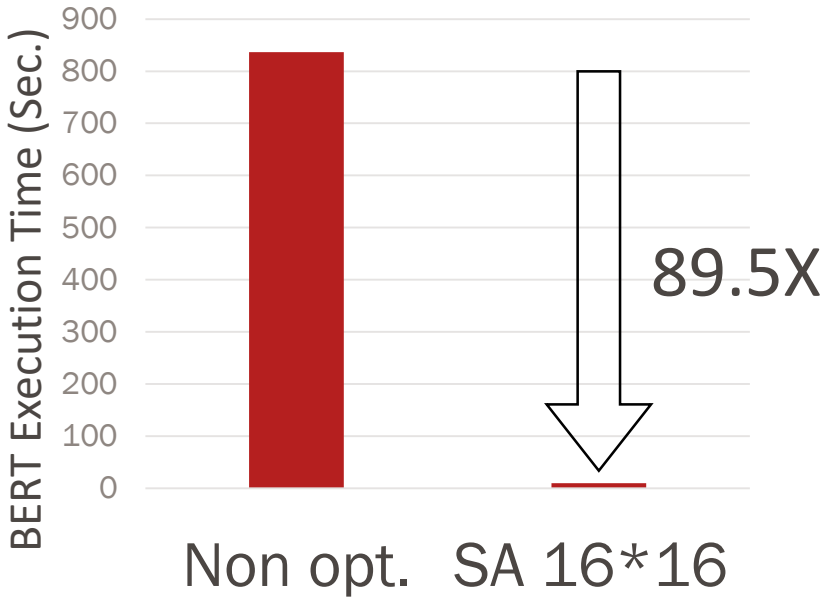
Single-Instruction-Multiple-Data (SIMD)

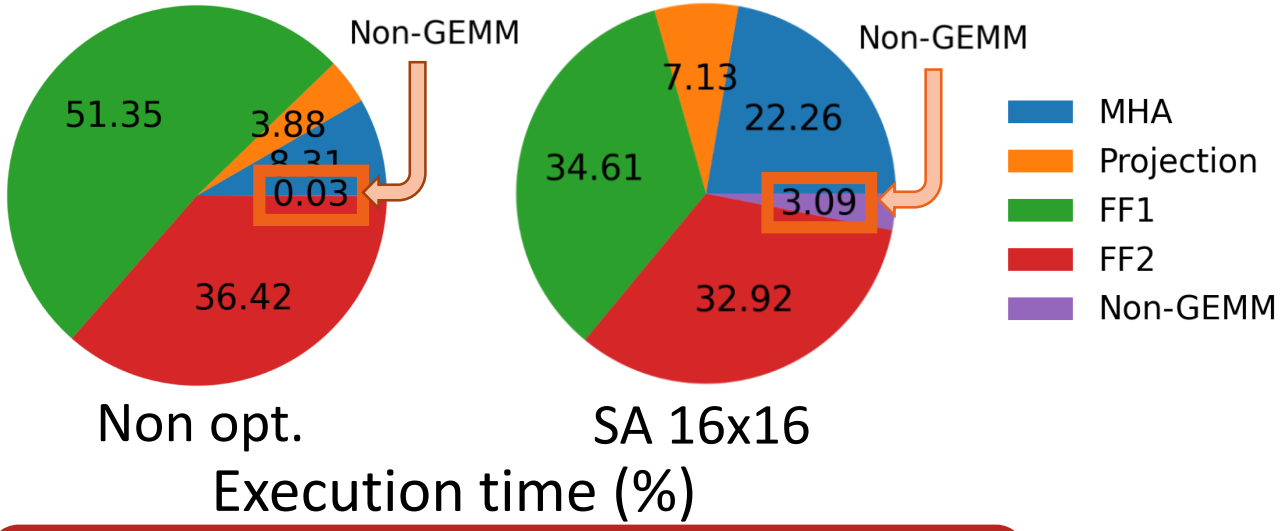


Systolic Array (SA)

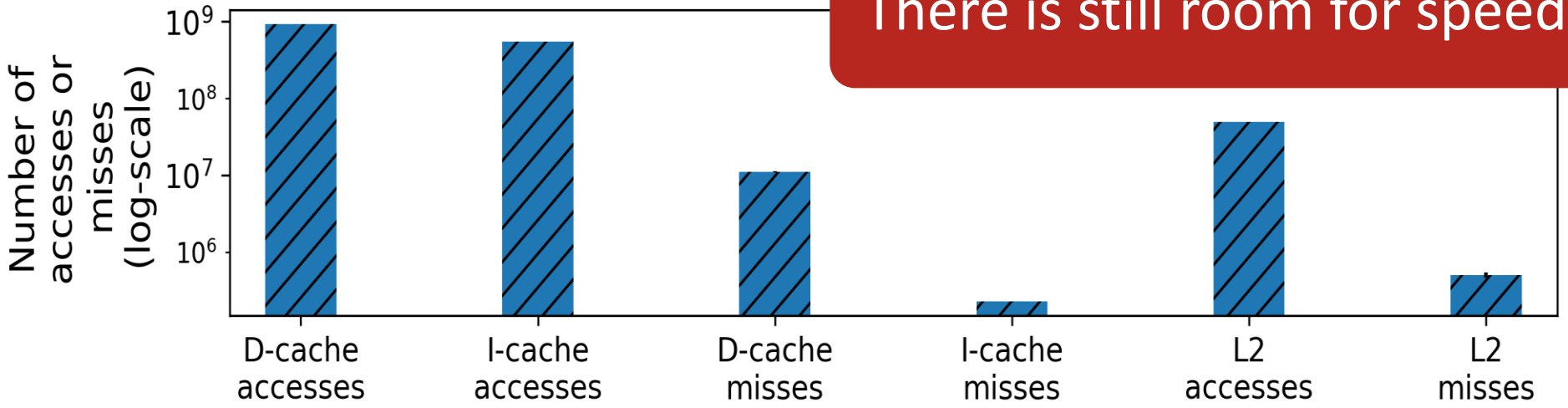


Transformer acceleration





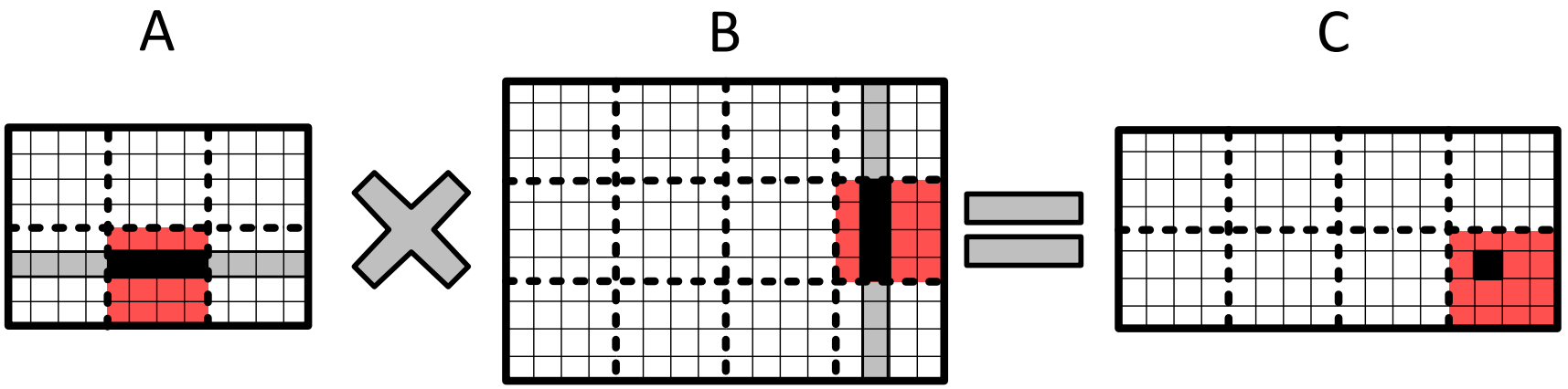
There is still room for speed up.



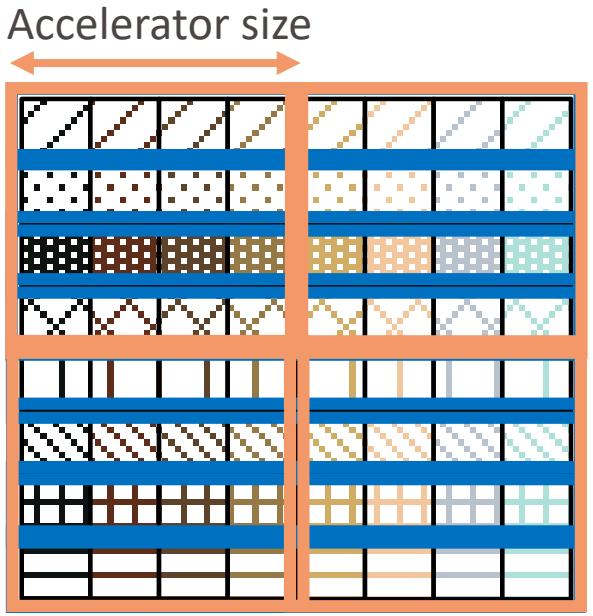
Tiling in GEMM operations

- Matrices are typically larger than the accelerator size
- Matrices must be tiled → One tile processed at a time
- Partial results are aggregated with element-wise additions.

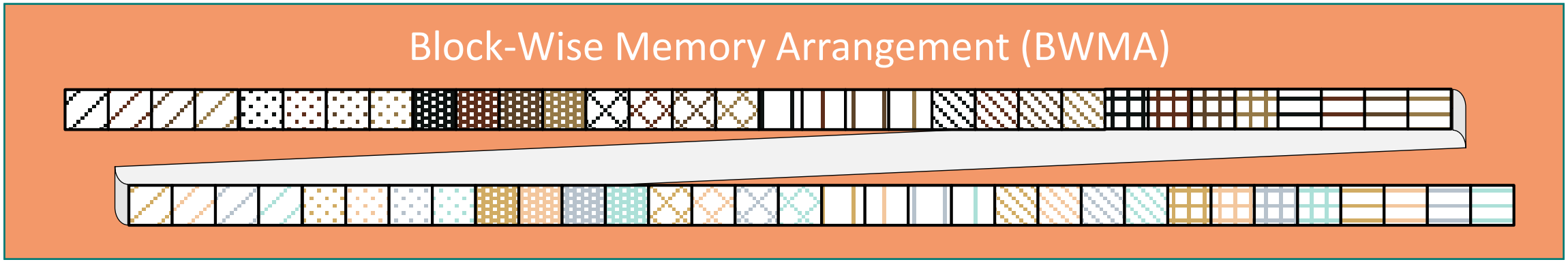
BUT, how do we store a matrix in the memory?



Aligning memory arrangement with the accelerator

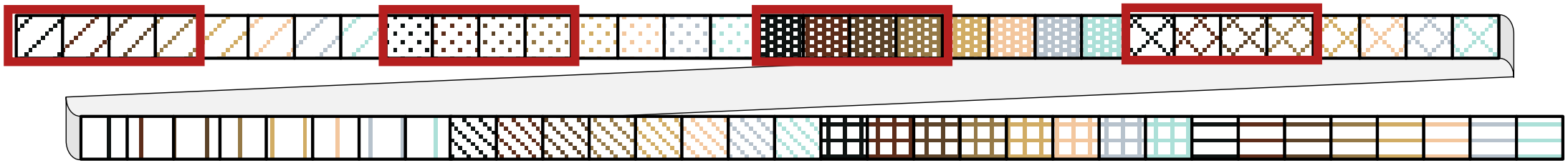


1-Dimensional array

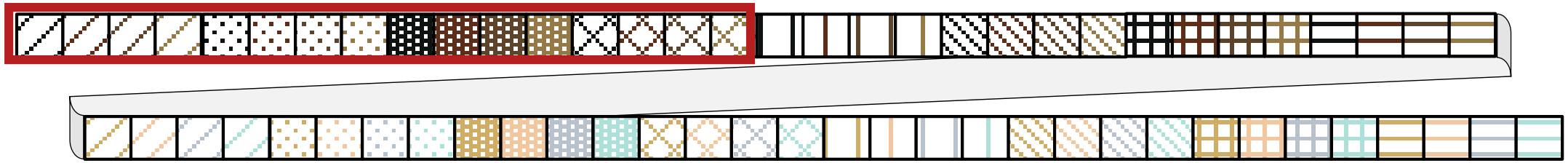


BWMA for GEMM operations

RWMA



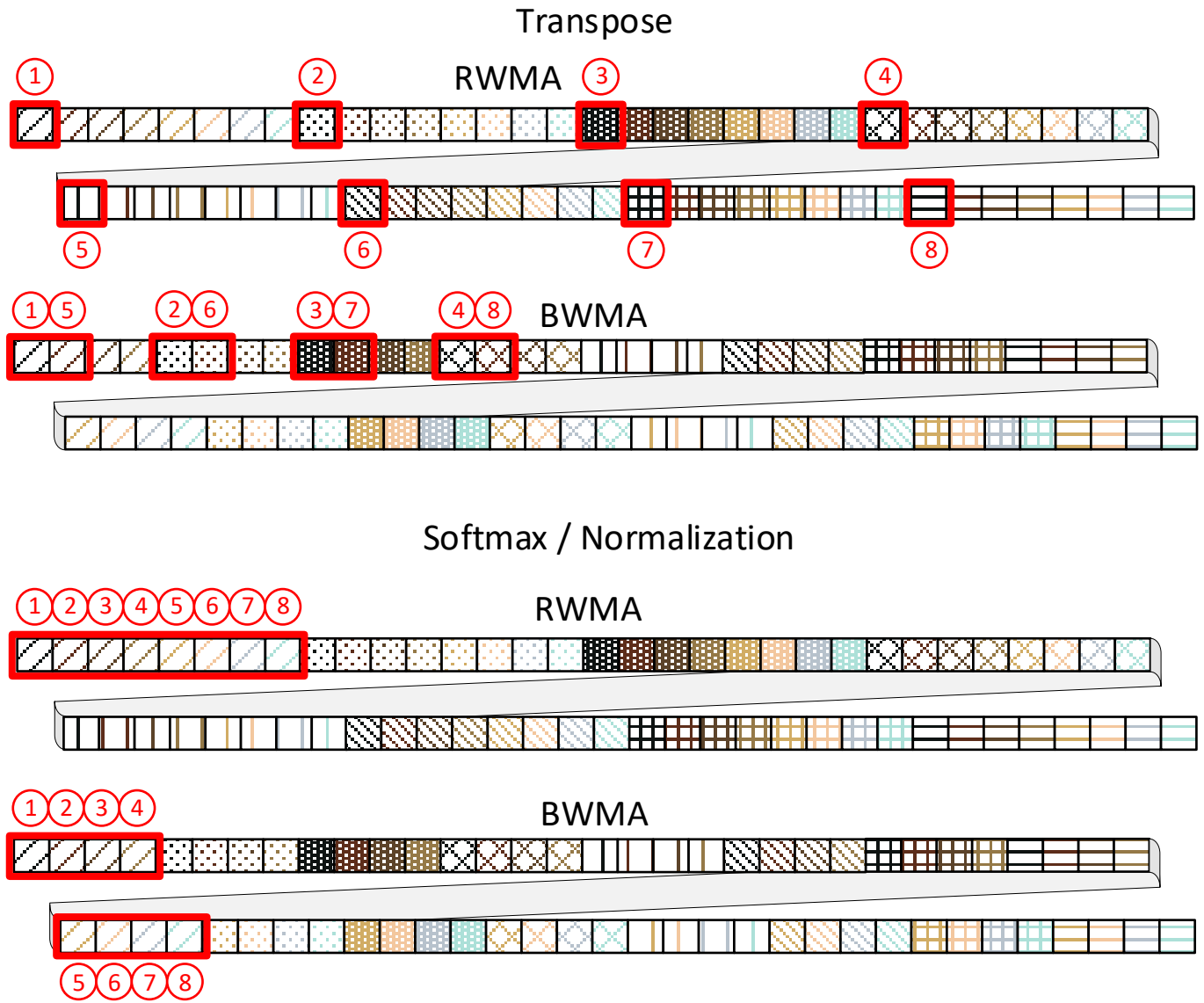
BWMA



Consider that 97% of the transformer runtime is GEMM

BWMA for non-GEMM operations

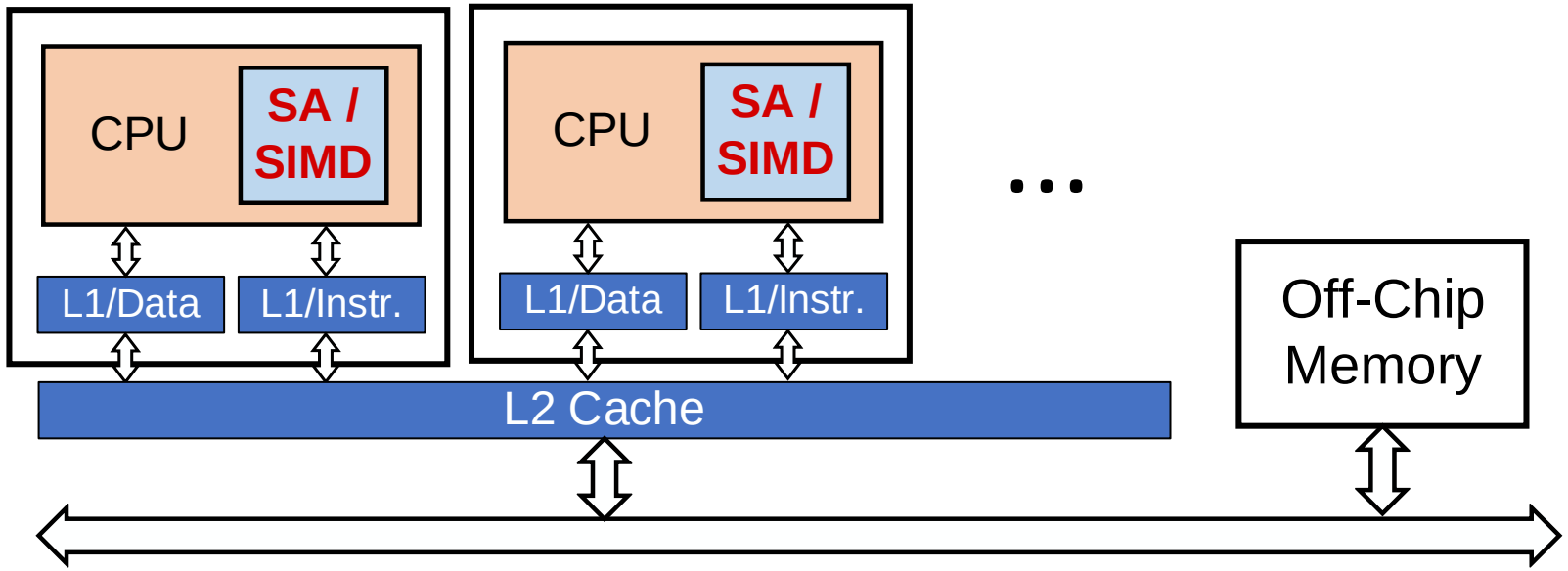
- Non-GEMM functions in a transformer model
  - Activation -> element-wise
  - Transpose
  - Softmax
  - Normalization
- Only 3% of the whole transformer runtime
- Negligible overhead in BWMA w.r.t RWMA



Configurations: Full system simulation

- System simulated in gem5-X [4]:

CPU Core	Single- or multi-core in-order CPU @ 2.3 GHz
L1/L2 Cache Size	L1-Instruction: 32 KB, L1-Data: 32 KB, L2: 1 MB
Instruction Set Architecture (ISA)	ARMv8 (AArch64)
Main Memory	4 GB DDR4 @ 2400 MHz
Operating System	Ubuntu 16.04 LTS
Accelerators	**SA 16x16 / SA 8x8 / SIMD 16**

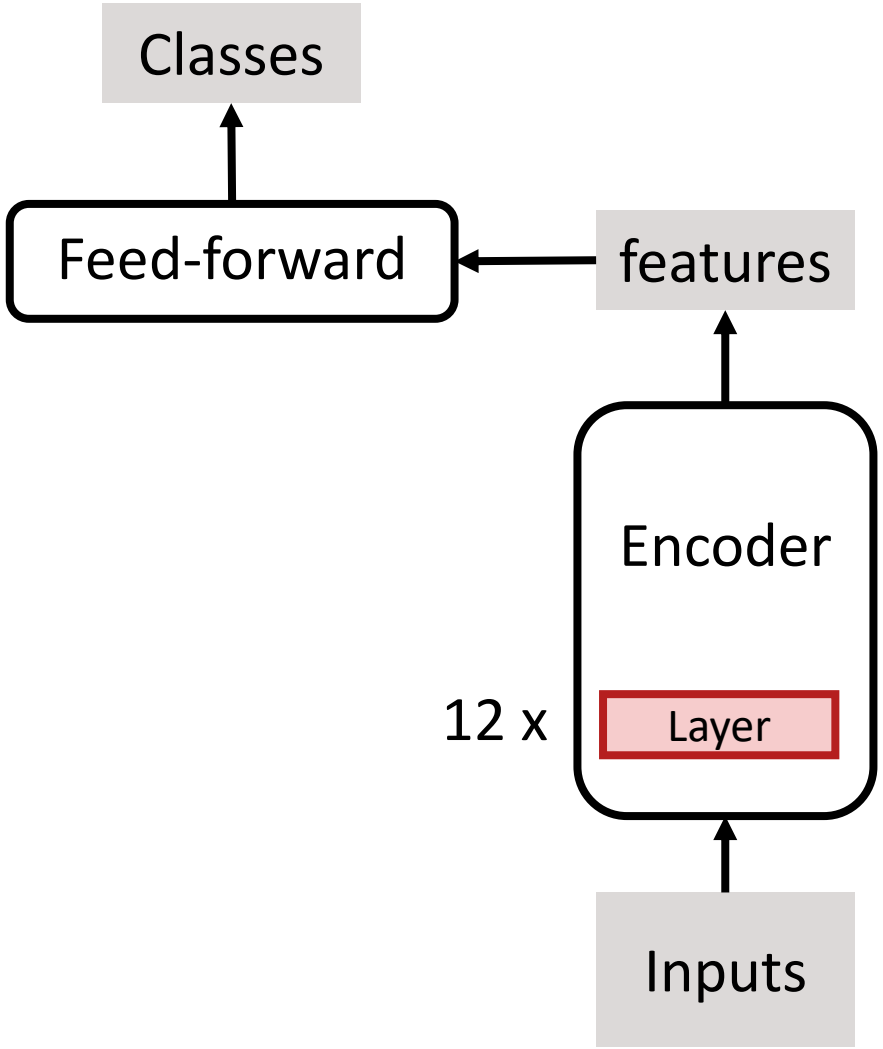


[4] Y. Qureshi et al., HPC 2019

Configurations: Application

- Application: BERT-base [5]

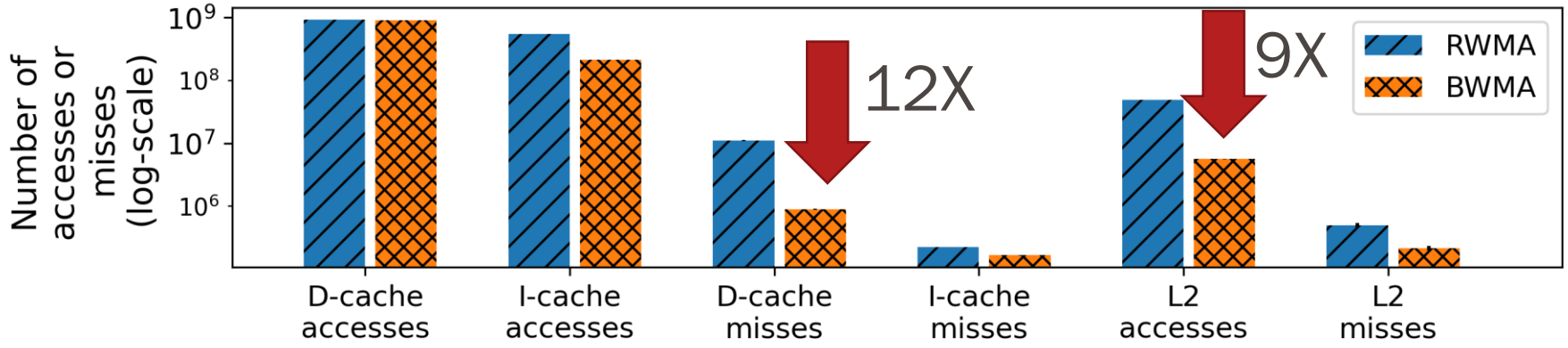
Sequence Length	512
Model Dimension	768
Feed-forward Dimension	3072
Number of Heads	12
Total Number of Parameters	110M



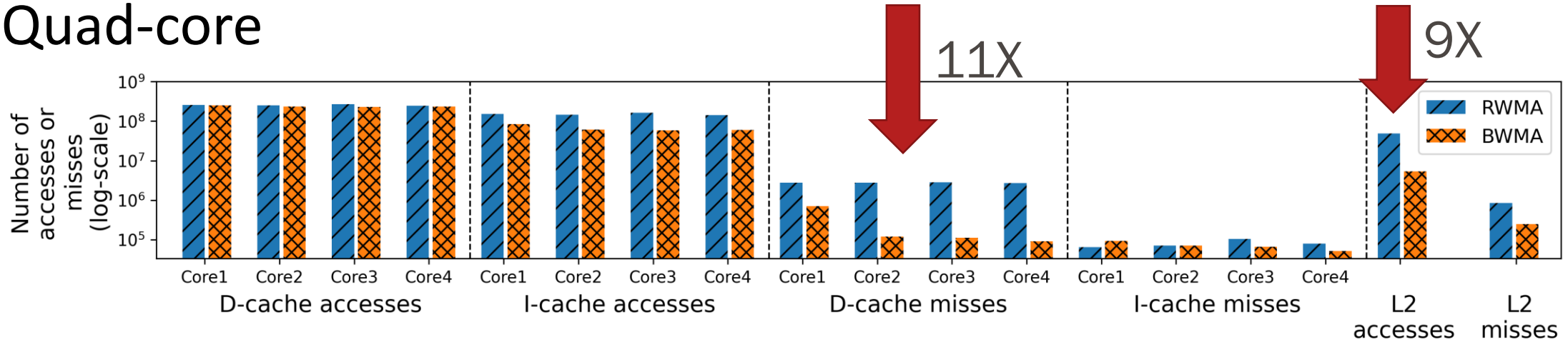
[5] J. Devlin et al., ArXiv 2019

Impact on memory accesses

Single-core



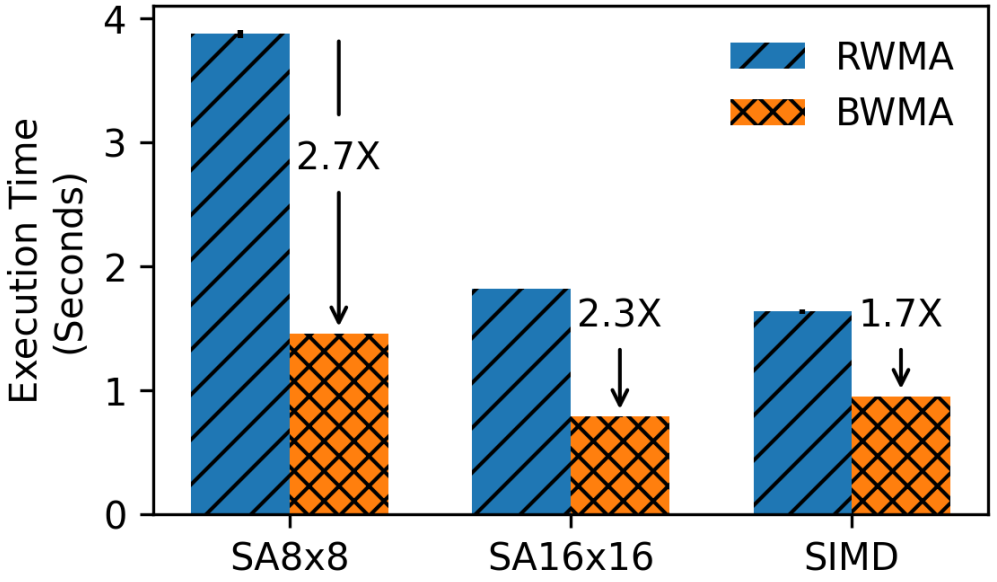
Quad-core



Impact of BWMA on inference run-time

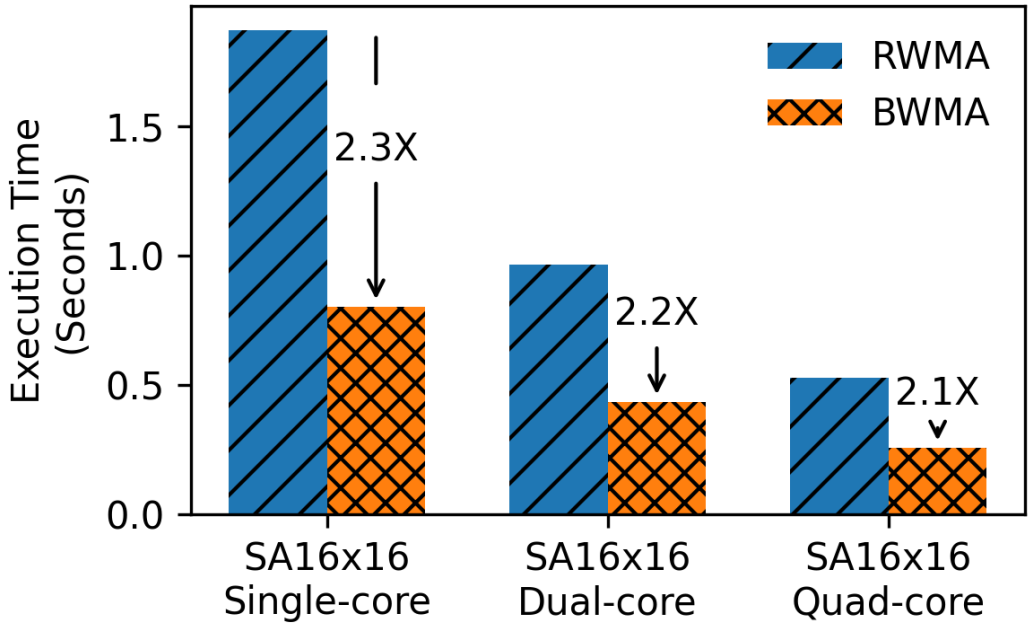
Single-core system with different accelerators

- Up to 2.7x speed up



Multi-core systems with SA16x16 accelerators

- BWMA is more effective than duplicating number of cores and accelerators.



Conclusion

- Aligning memory arrangements with accelerator size is key.
- Up to 2.7X speed up across different accelerator architectures
- Code available : https://github.com/gem5-X/TiC-SAT

Thank you!