MEEP

MareNostrum Experimental
Exascale Platform

www.meep-project.eu

MEDEA: Improved Memory-Level Parallelism in a decoupled execute/access vector accelerator

(work in progress)

Umair Riaz

Luis A. Plana

Peter Wilson

John D. Davis

Barcelona Supercomputing Center

Overview

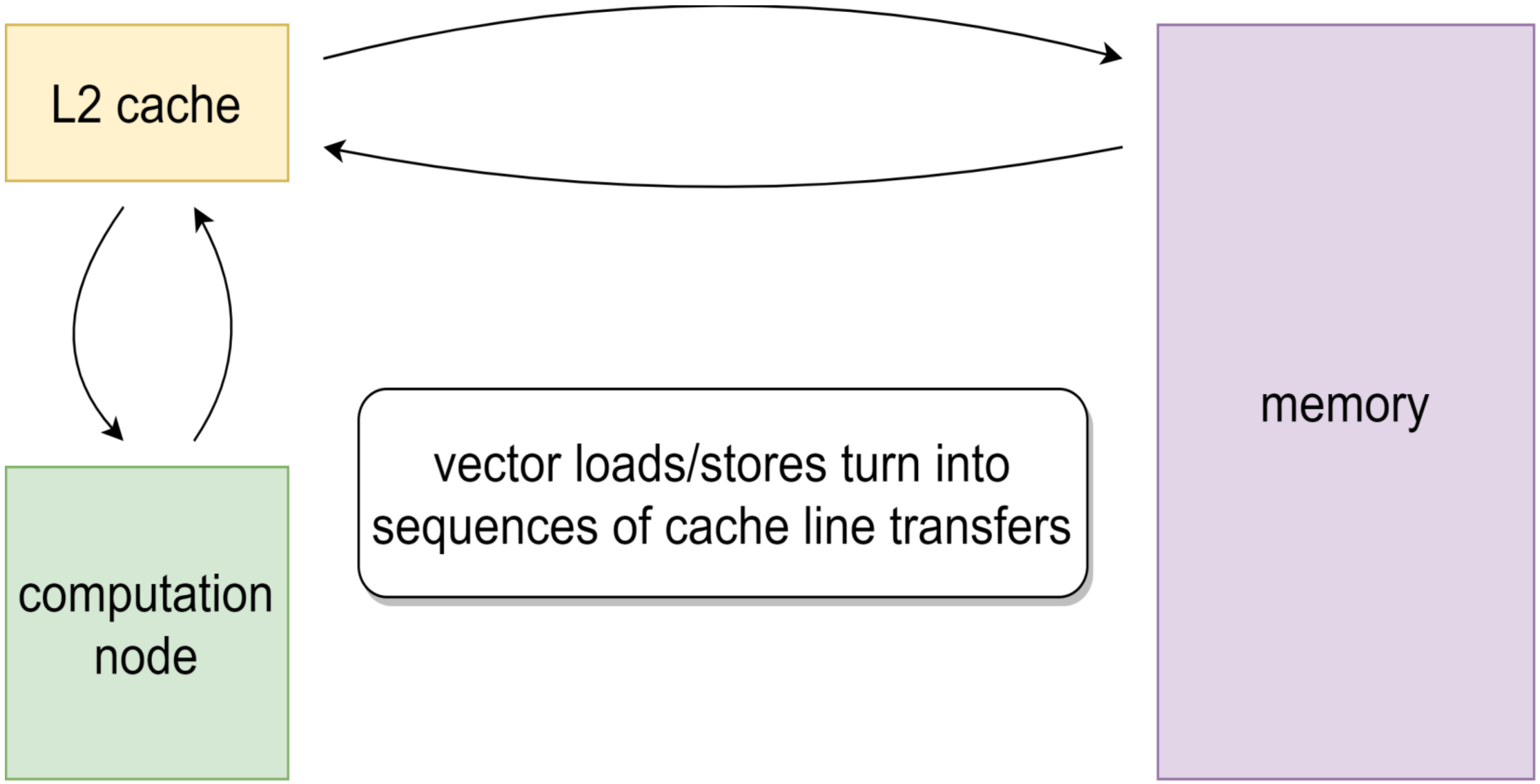
- Motivation
- Introduction to MEDEA
- Microarchitecture
  - Interfaces
  - Supported types of requests
  - Building blocks of MEDEA
- Discussion

Motivation

- Efficient use of memory bandwidth for sparse access patterns
- Reducing data movements between compute node and memory
- Reducing NoC traffic
- Efficient vector data processing
- Improve memory-level parallelism (MLP)

Introduction – A classical system

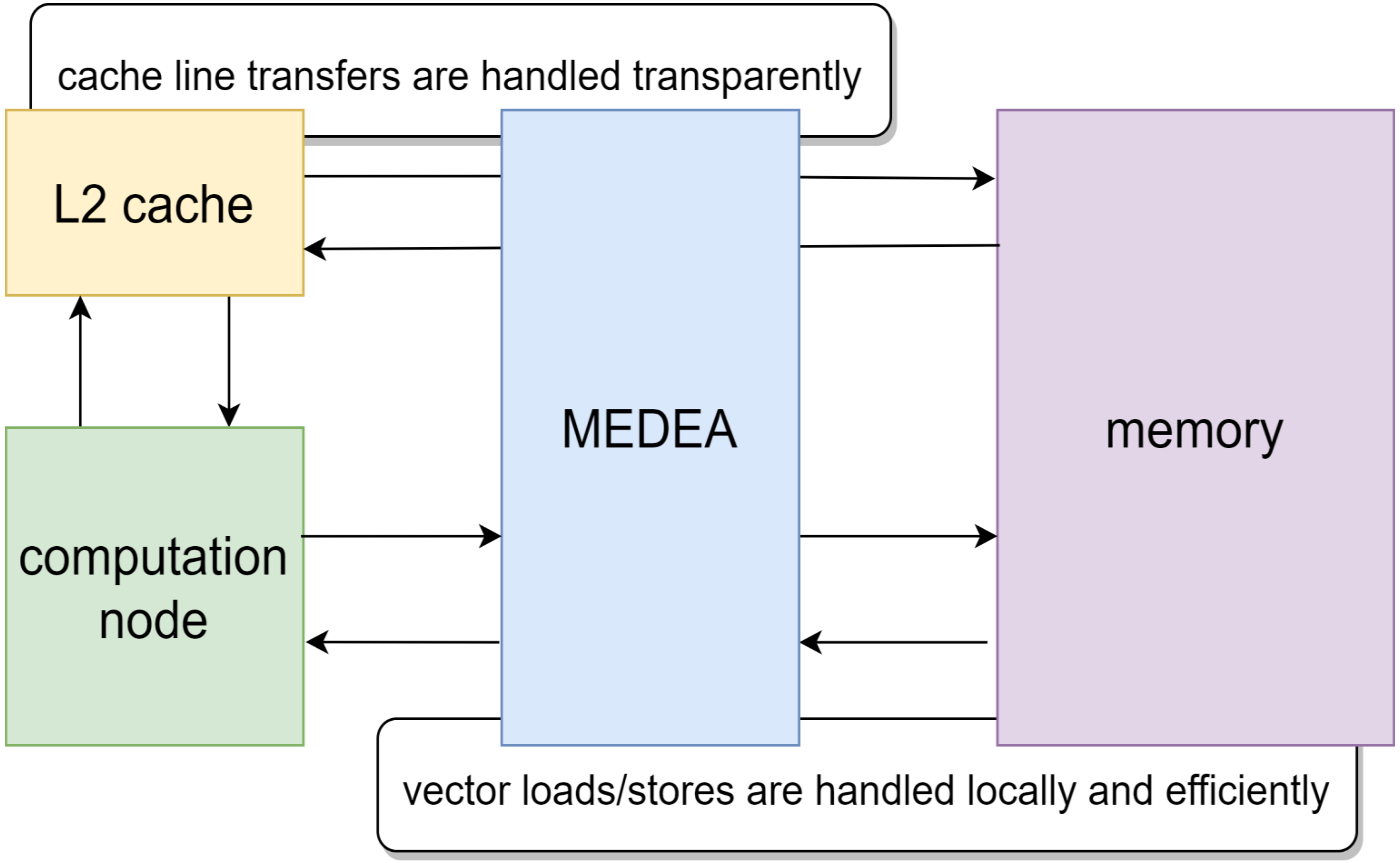
- A classical system's representation



vector loads/stores turn into sequences of cache line transfers

Introduction – MEDEA in a system

- A classical system with MEDEA

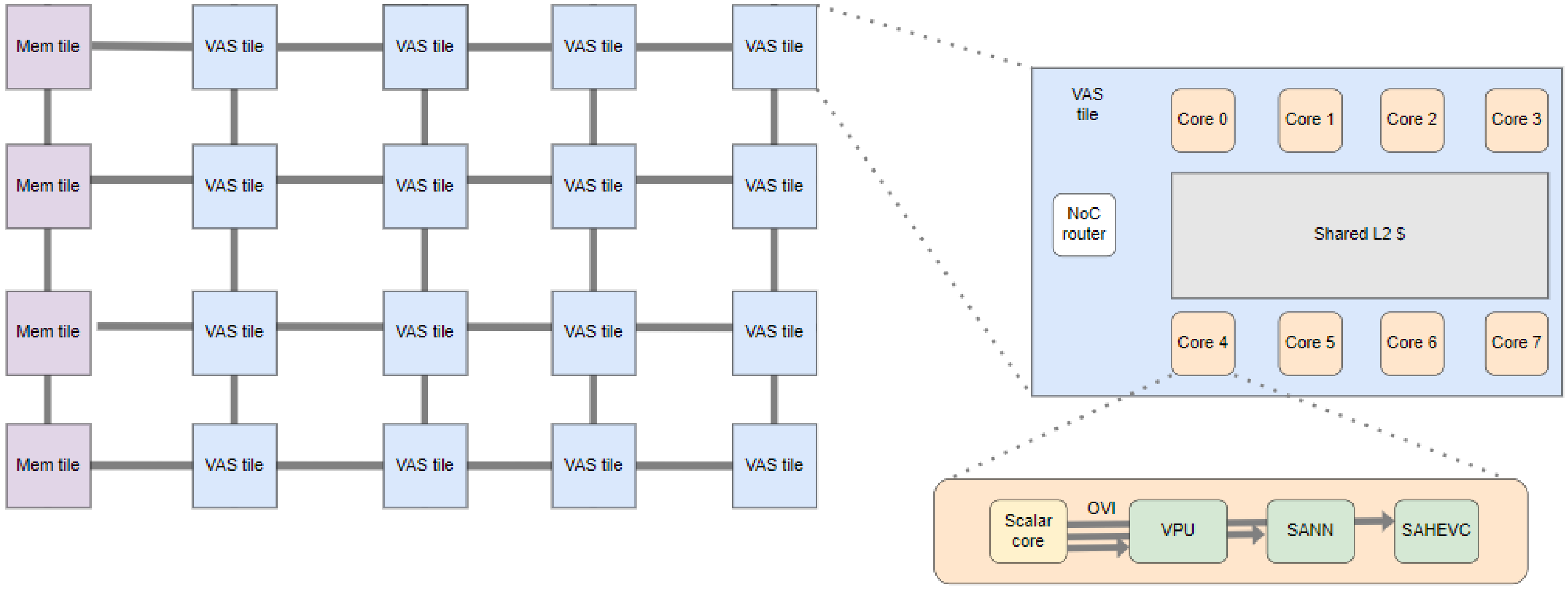


cache line transfers are handled transparently

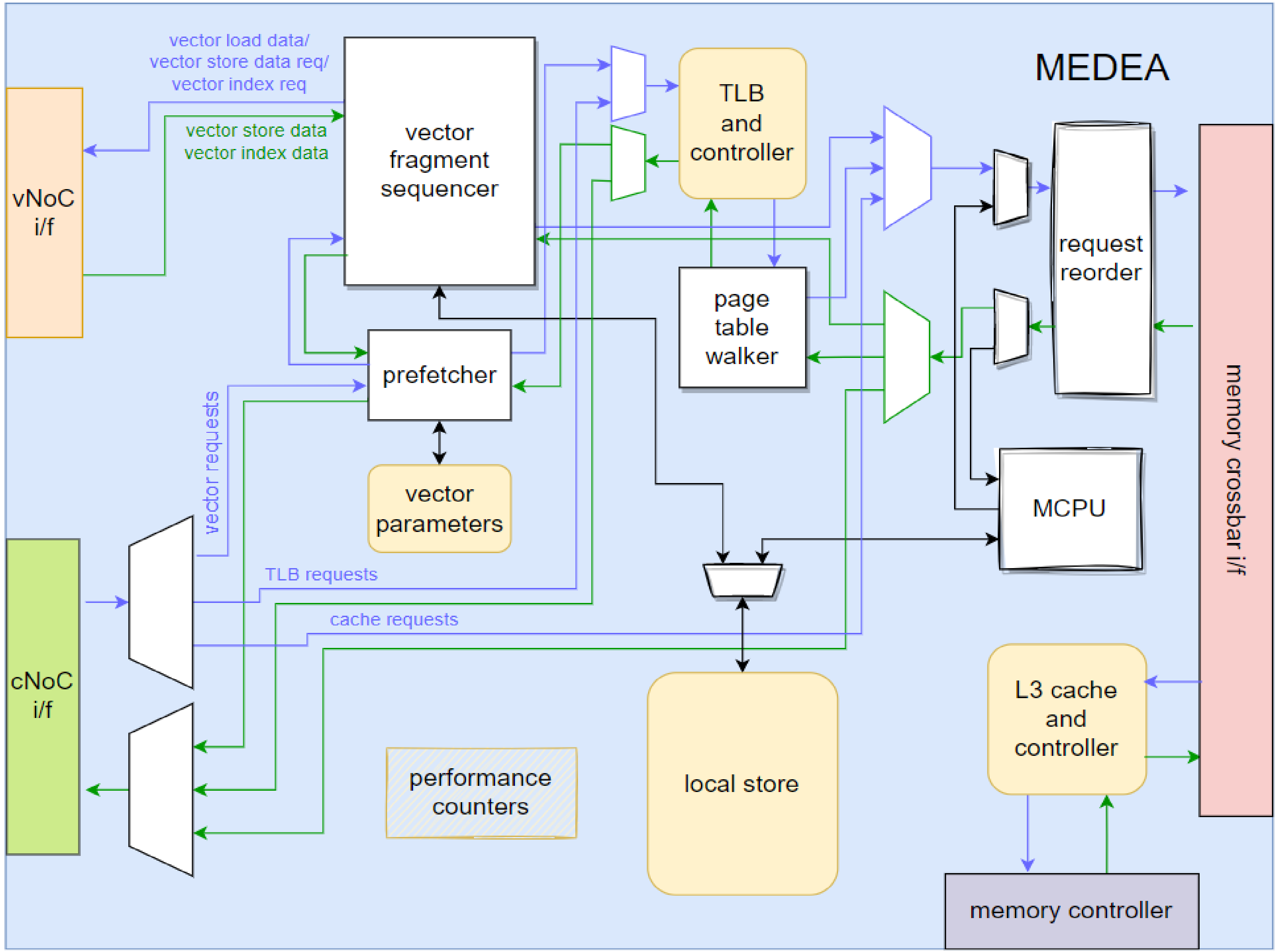
vector loads/stores are handled locally and efficiently

Introduction - MEDEA in ACME

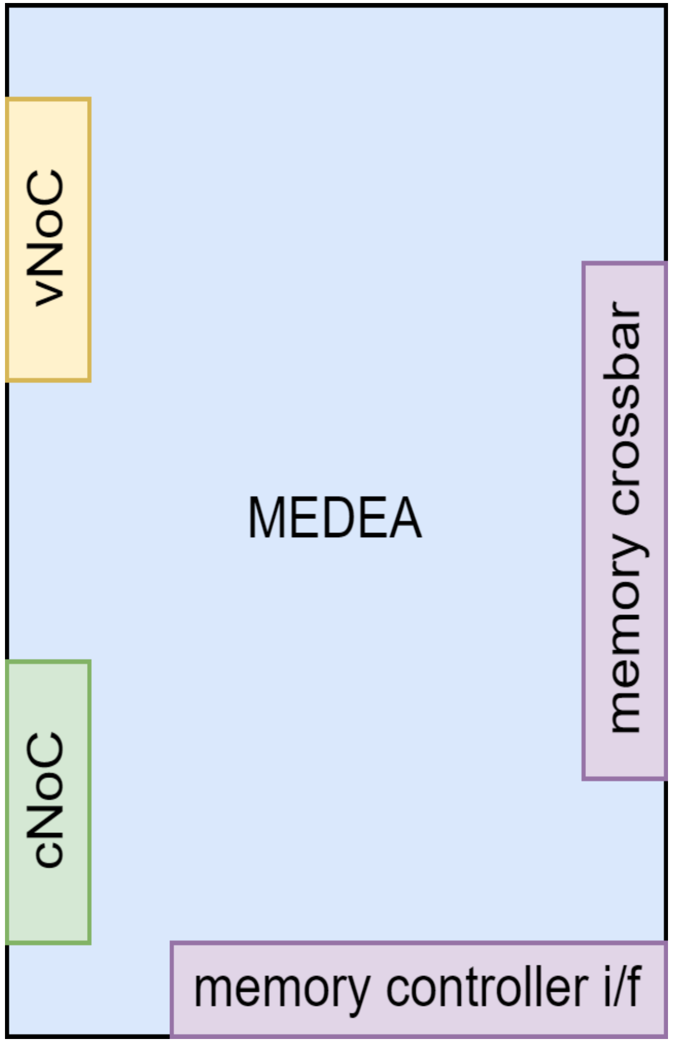
- ACME increases MLP by shifting memory-accessing responsibilities from compute tile to specialized **M**emory **E**ngine for **D**ecoupled **E**xecute/**A**ccess (MEDEA)
- VPU is commonly known to exploit data-level parallelism (DLP), but with the addition of MEDEA, it adds up capabilities for MLP



Microarchitecture - MEDEA



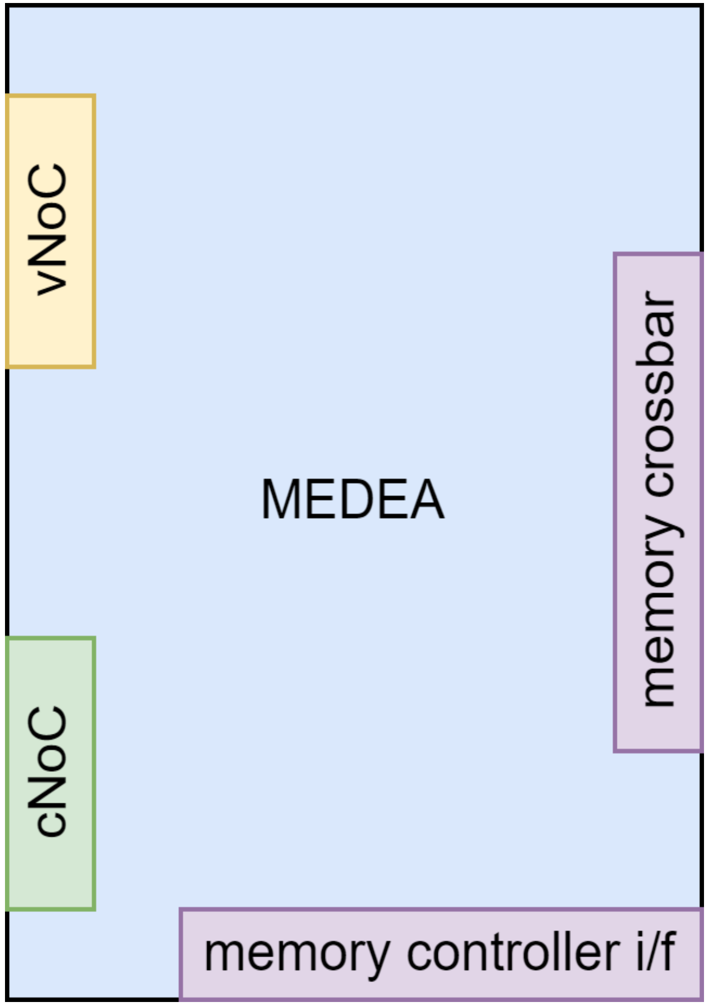
Microarchitecture - Interfaces



Microarchitecture – Interfaces (2)

- cNoC (compute NoC) ←→ compute node
- vNoC (vector NoC) ←→ LVRF
- Memory crossbar ←→ interconnecting all the MEDEA

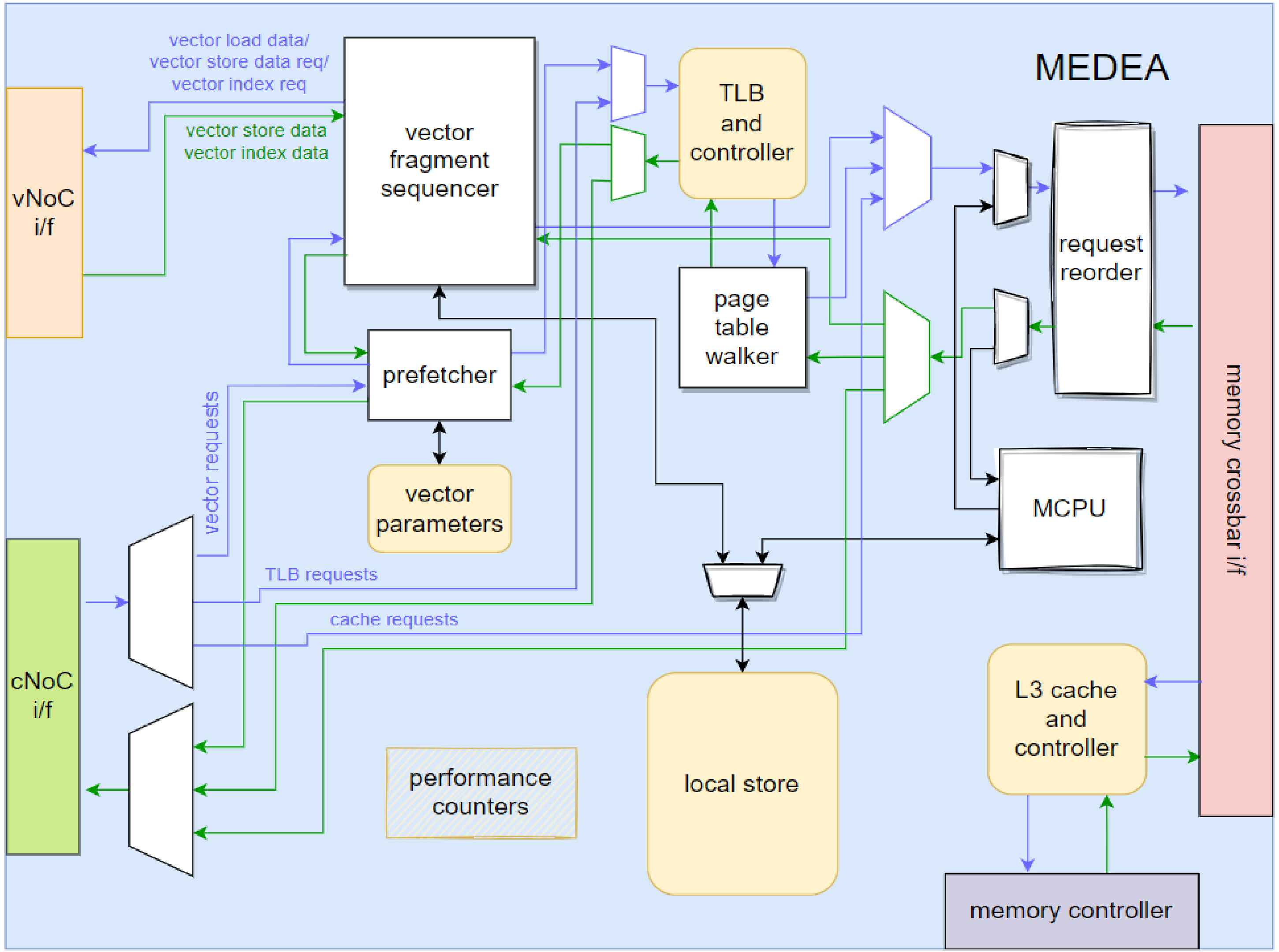
  . tiles and memories
- Memory controller i/f ←→ HBM and NVRAM



Microarchitecture – Types of Requests

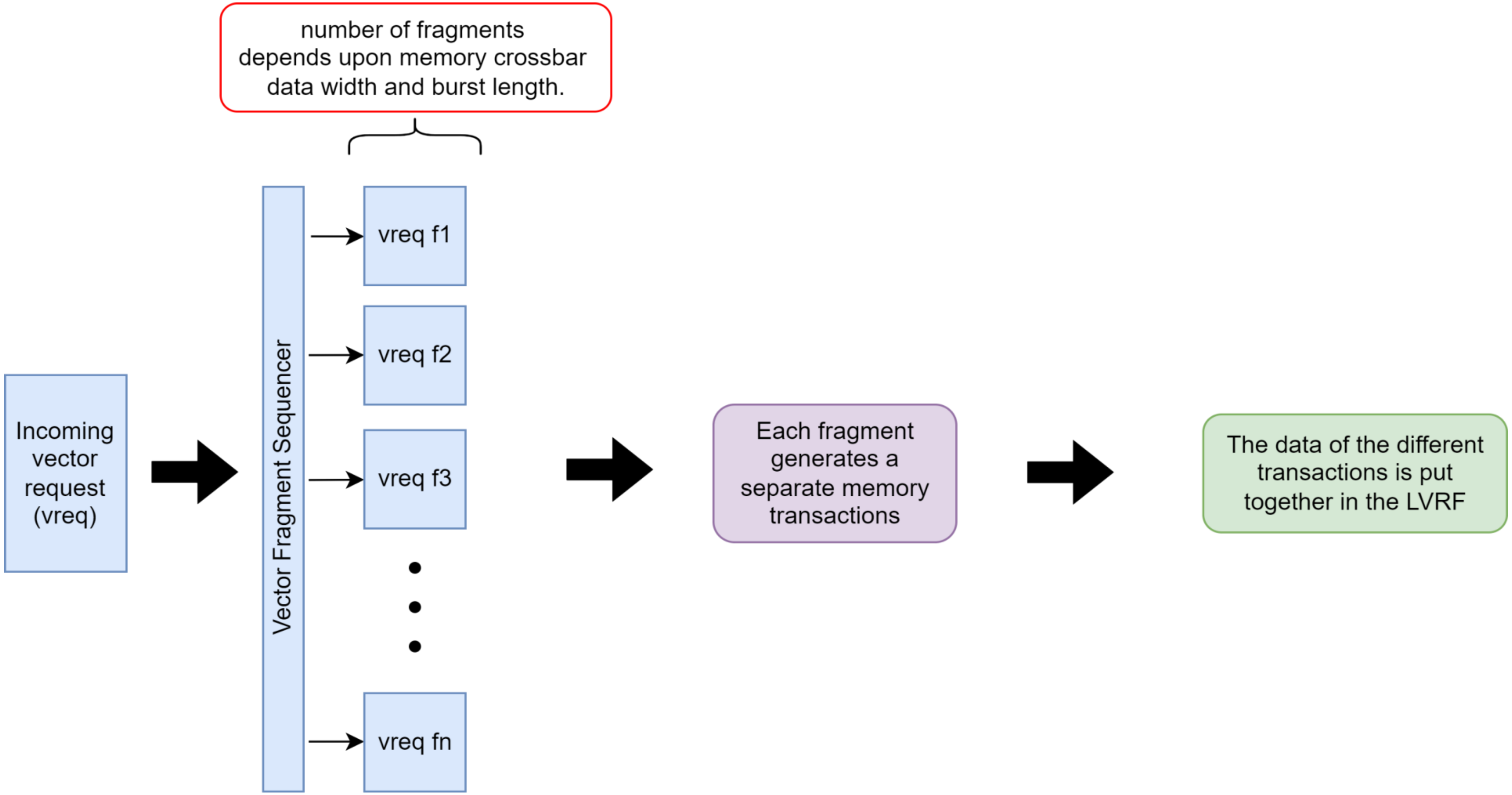
Request	Request Parameters	Reply
cache miss – read	physical address, length	memory data
cache miss - write	physical address, length, data	completion acknowledge
virtual-to-physical address translation	virtual address	physical address
vector parameter set	application-requested vector length	granted vector length
vector load	virtual address, addressing mode, vector register, *renamed* vector register, *(mode-dependent parameters: stride, index* vector*)*	(densified) memory data
vector store	virtual address, addressing mode, vector register, *renamed* vector register, *(mode-dependent parameters: stride, index* vector*)*	completion acknowledge
atomic memory operations	TBD	completion acknowledge

Microarchitecture - MEDEA



MICROARCHITECTURE – BUILDING BLOCKS

Vector Fragment Sequencer



number of fragments depends upon memory crossbar data width and burst length.

Each fragment generates a separate memory transactions

The data of the different transactions is put together in the LVRF

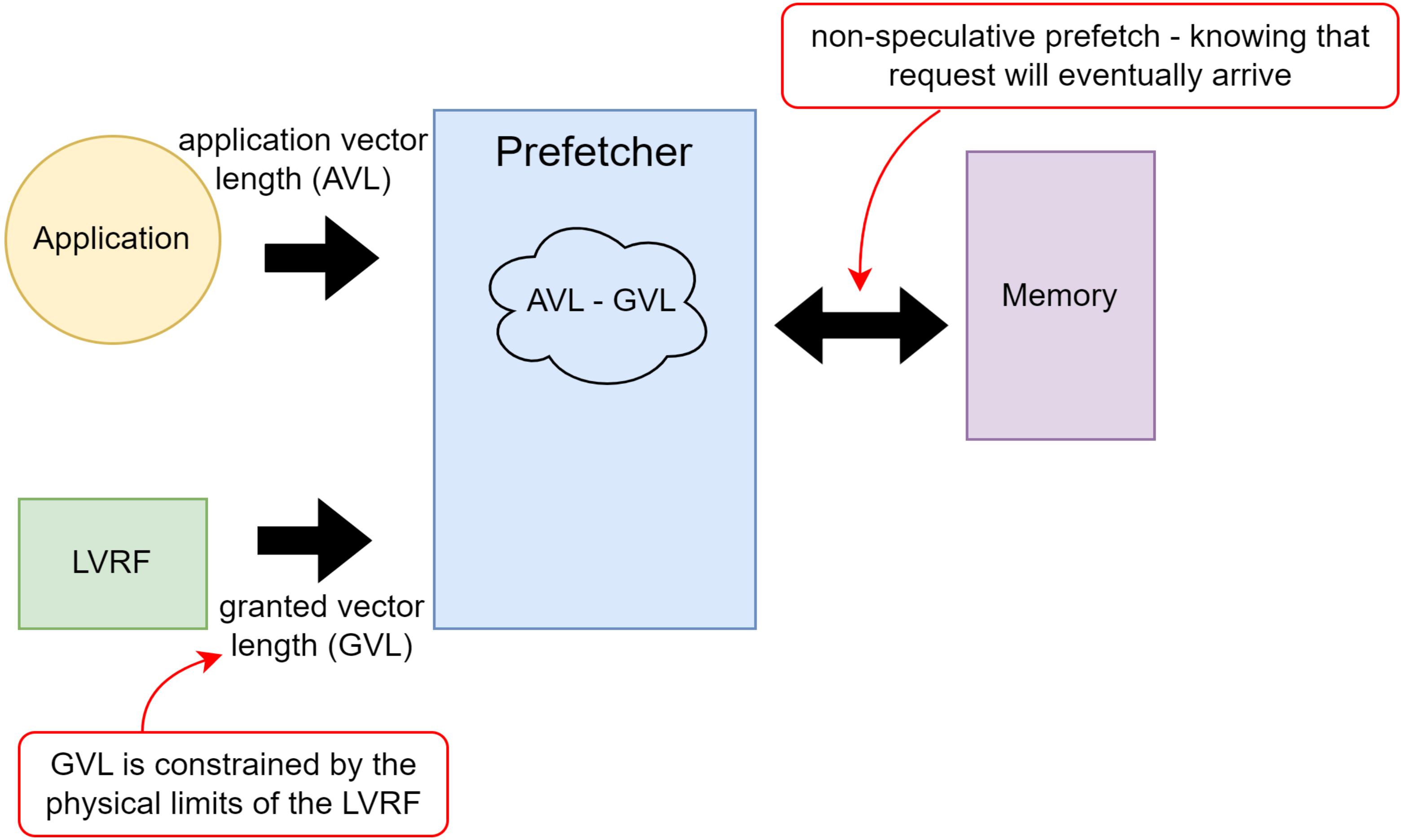
MICROARCHITECTURE – BUILDING BLOCKS (2)

Vector Fragment Sequencer

- RISC-V vector operations support following addressing modes
  - unit-stride: managed as dense memory accesses
  - strided, indexed: managed as sparse memory accesses
- In the case of strided or indexed mode, a fragment might end up having a single vector element
- All the elements from different fragments are collected and packed locally and transferred to LVRF as a dense vector
  - Less parasitic data movements
  - Consequently, saving energy and NoC traffic

Microarchitecture – Building Blocks (3)

Prefetcher



non-speculative prefetch - knowing that request will eventually arrive

GVL is constrained by the physical limits of the LVRF

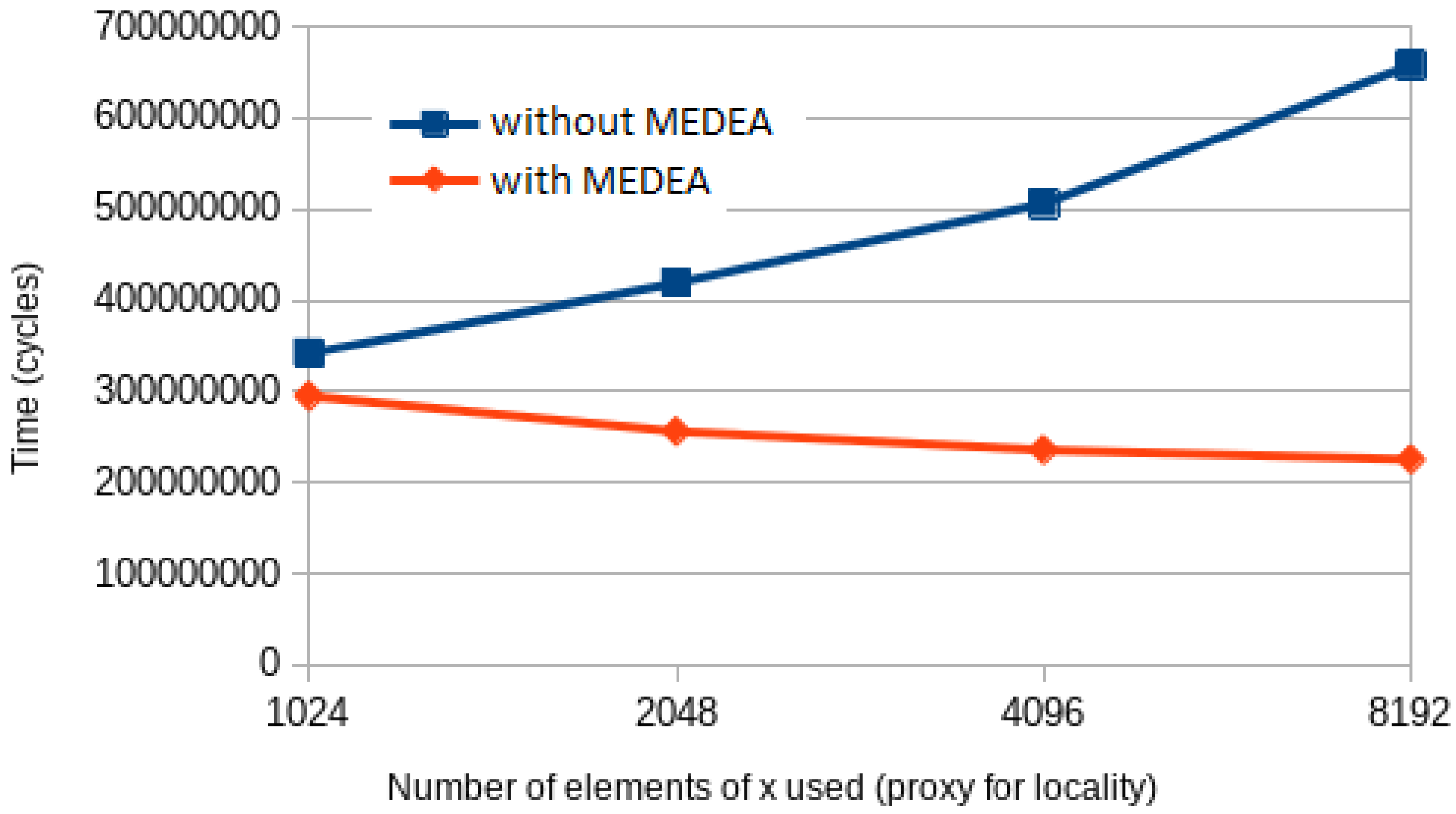
Microarchitecture – Building Blocks (4)

Memory CPU (MCPU)

- A scalar processor
- Tightly-coupled memory and a low-latency interface to the memory controller
- Provides a collection of memory-intensive functions that can be accessed by the compute tiles
- Executing the functions locally and close to memory improves:
  - Performance
  - Energy
  - NoC traffic

Discussion

- Sparse Matrix Vector (SpMV) benchmark simulation time comparison



MEEP

MareNostrum Experimental
Exascale Platform

www.meep-project.eu

Thank you!

umair.riaz@bsc.es