Andor Menczer

# MASSIVELY PARALLEL TENSOR NETWORK ALGORITHMS

Supervisors: Dr. Tamás Kozsik (ELTE), Dr. Örs Legeza (Wigner)
Note: these slides only function as an outline, therefore they are not to be used without giving further explanation

ELTE
EÖTVÖS LORÁND
TUDOMÁNYEGYETEM

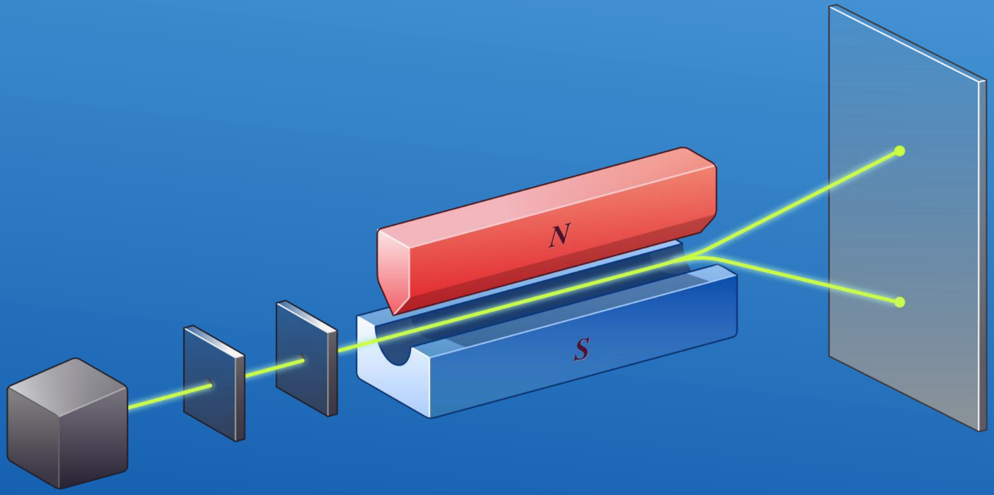
# ELECTRON SPIN

Spinning electrons? Whaaat? How do they do that?

# ELECTRONS ARE SPINNING BALLS

… except they are not spinning

… and they aren’t balls either



# ENTANGLEMENT OF ELECTRONS

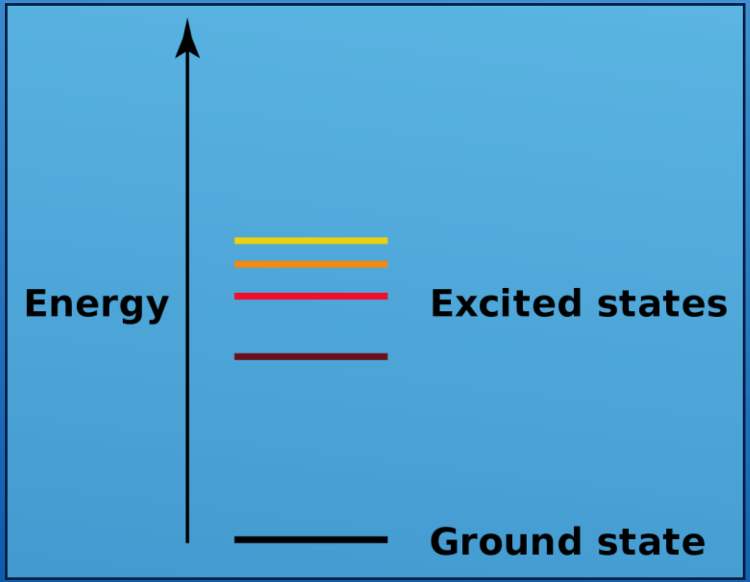
Subsystems cannot be described independently

Superposition and mixed states are common

# HAMILTONIAN

Description of the total energy level of the system

Ground state is defined as the eigen vector corresponding to the smallest possible eigenvalue



# EXCITED QUANTUM SYSTEMS

Entanglement → combinatorial explosion

Exact value of ground state cannot be calculated

Whole matrix representation → Tensor network states

Operations on the network can be executed locally, thus a polynomial solution becomes available

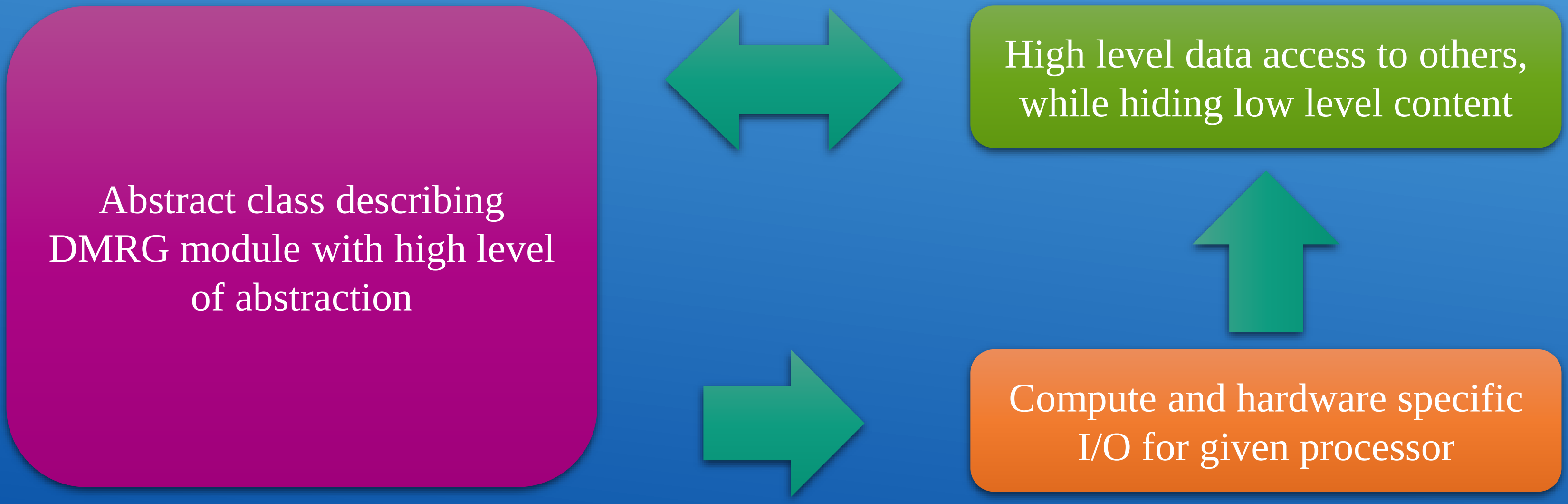
# GPU ACCELERATION

Restructuring monolithic CPU code for modular CPU / GPU execution

# MODULAR DESIGN

Base algorithm is decoupled from the way data is stored, processed and operated on

Data storage, access and their operations are universally designed without any specific algorithm attached to them



Abstract class describing DMRG module with high level of abstraction

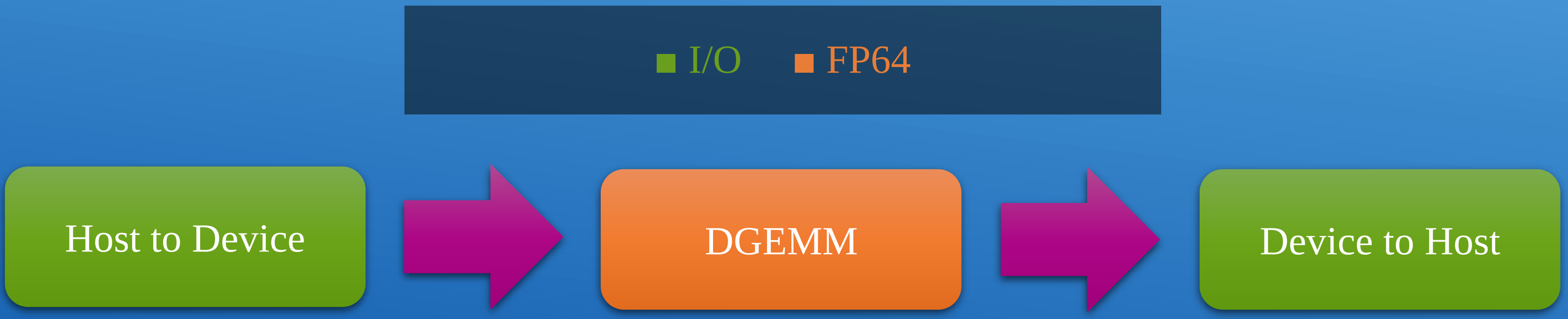
High level data access to others, while hiding low level content

Compute and hardware specific I/O for given processor

# LARGER MATRICES

On the fly synchronous I/O → Minimal VRAM requirement

$O(n^2)$ I/O time is masked by $O(n^3)$ FP64 time



🟩 I/O 🟧 FP64

Host to Device

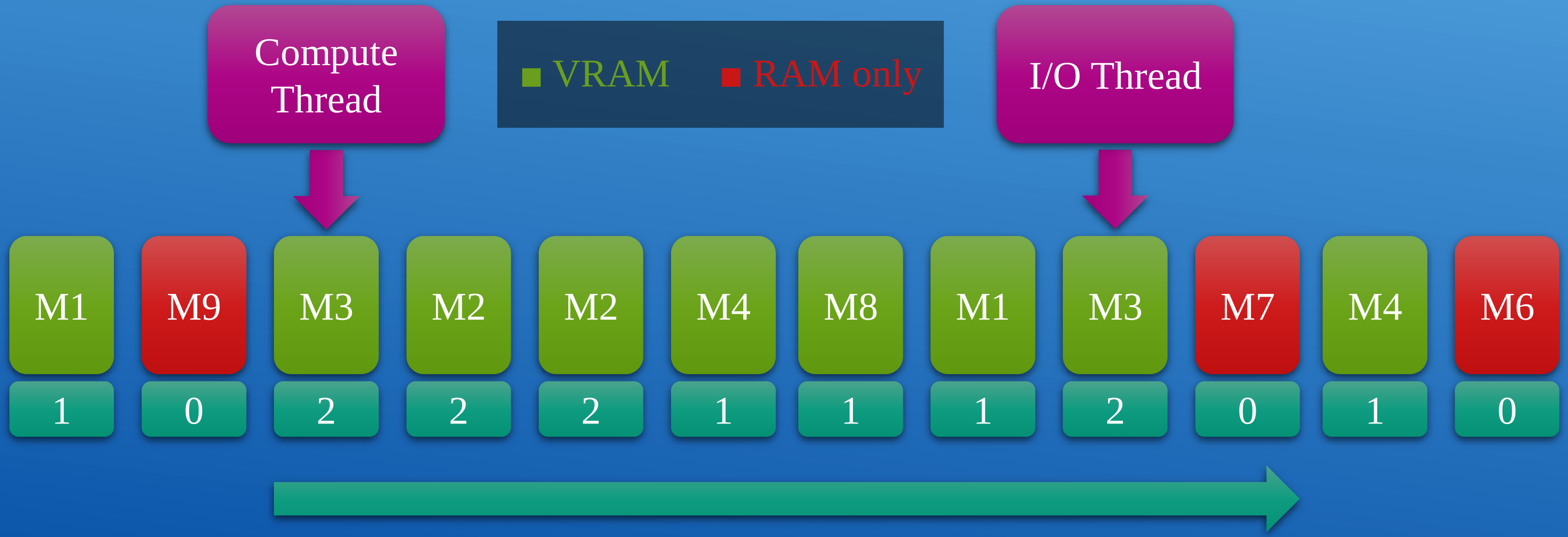
DGEMM

Device to Host

# I/O OF SMALLER MATRICES

Continuous asynchronous preloading on separate thread

I/O time is masked by running I/O and FP64 in parallel



Compute Thread

🟩 VRAM 🟥 RAM only

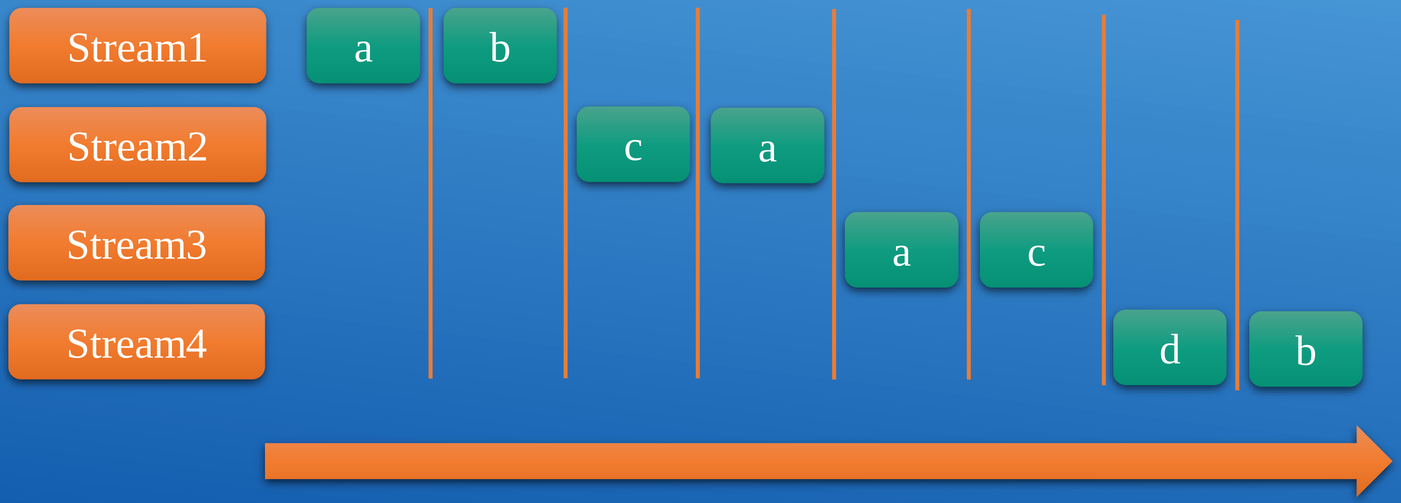
I/O Thread

|  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|
| M1 | M9 | M3 | M2 | M2 | M4 | M8 | M1 | M3 | M7 | M4 | M6 |
| 1 | 0 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 0 | 1 | 0 |

# COMPUTE OF SMALLER MATRICES

Batched DGEMM for deeply stacked matrices

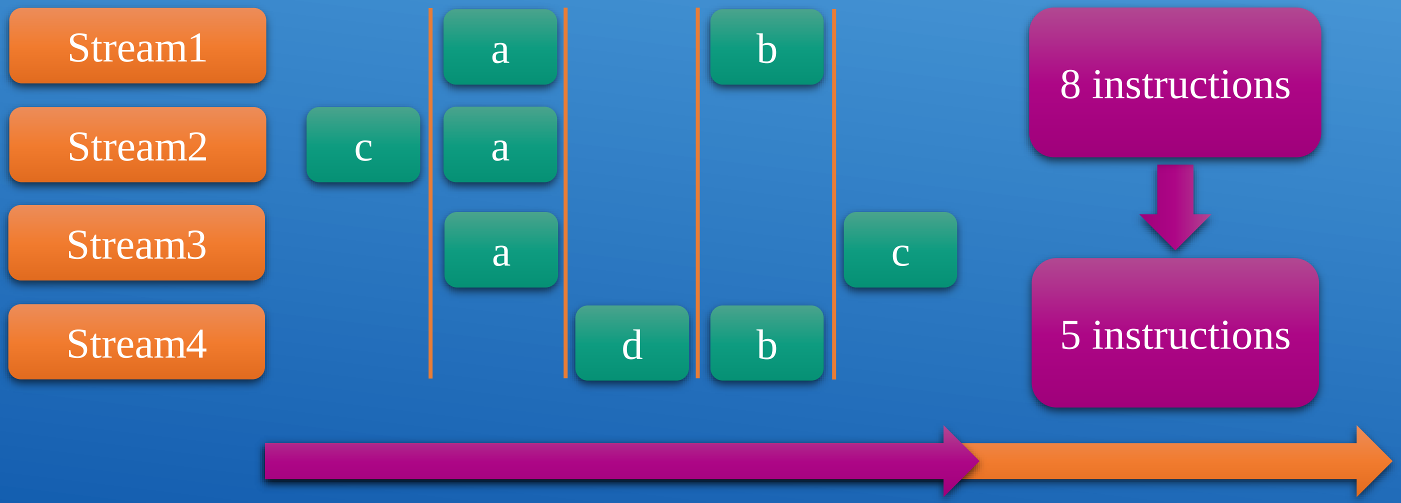
Interleaved streams for heterogenous matrix sizes



# INTERLEAVED STREAMS

Instruction sequence for each stream remains unchanged

Stacking the same instructions in each column → SIMD

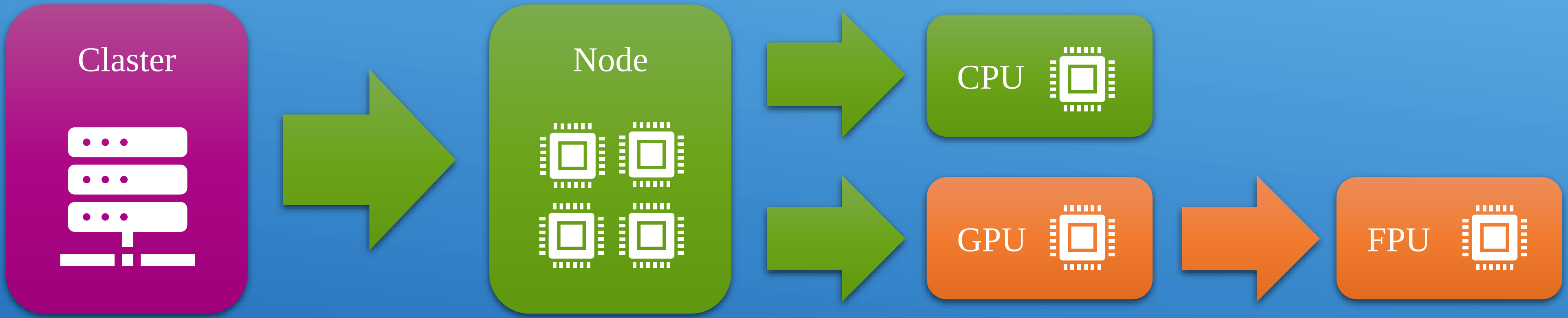


# HIGH LEVEL PARALLELIZATION

Optimizing TNS algorithms for HPC

# MULTI-STAGE PARALLELIZATION

🟪 Producer 🟩 Task parallelism 🟧 Data parallelism



Claster

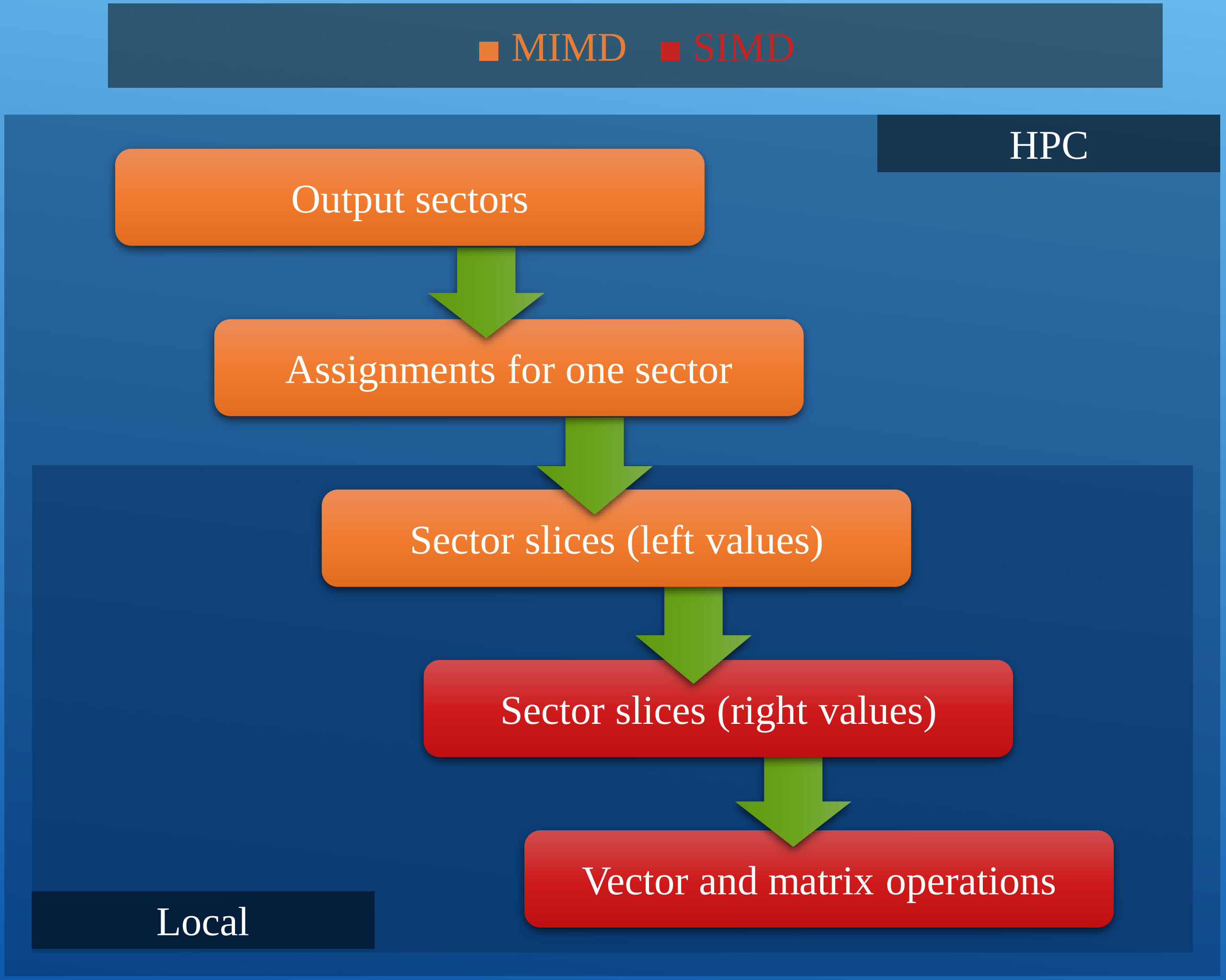
Node

CPU

GPU

FPU

# SOFTWARE LAYERS



🟧 MIMD 🟥 SIMD

HPC

Output sectors

Assignments for one sector

Sector slices (left values)

Sector slices (right values)

Vector and matrix operations

Local

THANK YOU FOR YOUR ATTENTION!