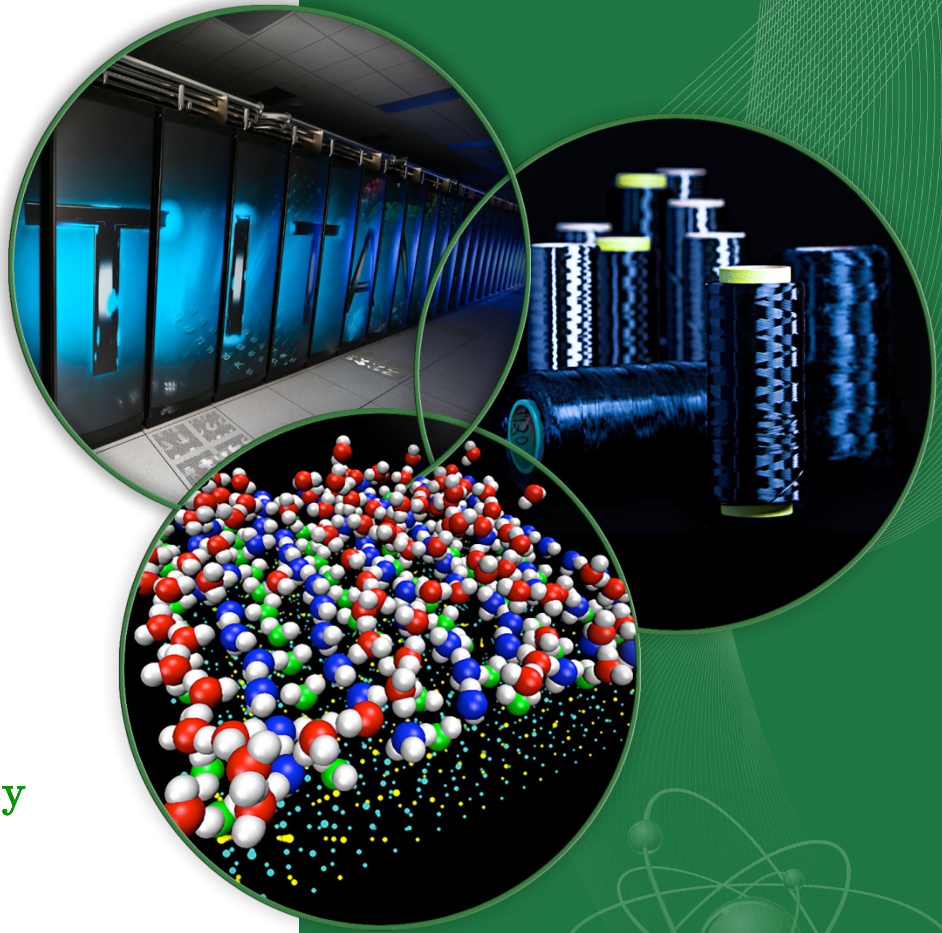
Portable Translation of Physical Models into High-Performance Software via Domain-Specific Virtualization: Quantum Many-Body Theory



Dmitry I. Lyakh (Liakh)

Scientific Computing

Oak Ridge Leadership Computing Facility

liakhdi@ornl.gov

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the US Department of Energy under contract No. DE-AC05-00OR22725.

ORNL is managed by UT-Battelle
for the US Department of Energy

OAK RIDGE National Laboratory | OAK RIDGE LEADERSHIP COMPUTING FACILITY

Quantum Many-Body Theory

$$|\Psi\rangle = \exp(\hat{T})|0\rangle = \left(1 + \hat{T} + \frac{1}{2!}\hat{T}^2 + \frac{1}{3!}\hat{T}^3 + \frac{1}{4!}\hat{T}^4 + \cdots\right)|0\rangle$$

$$|\Psi_{excited}\rangle = \hat{R}e^{\hat{T}}|0\rangle$$

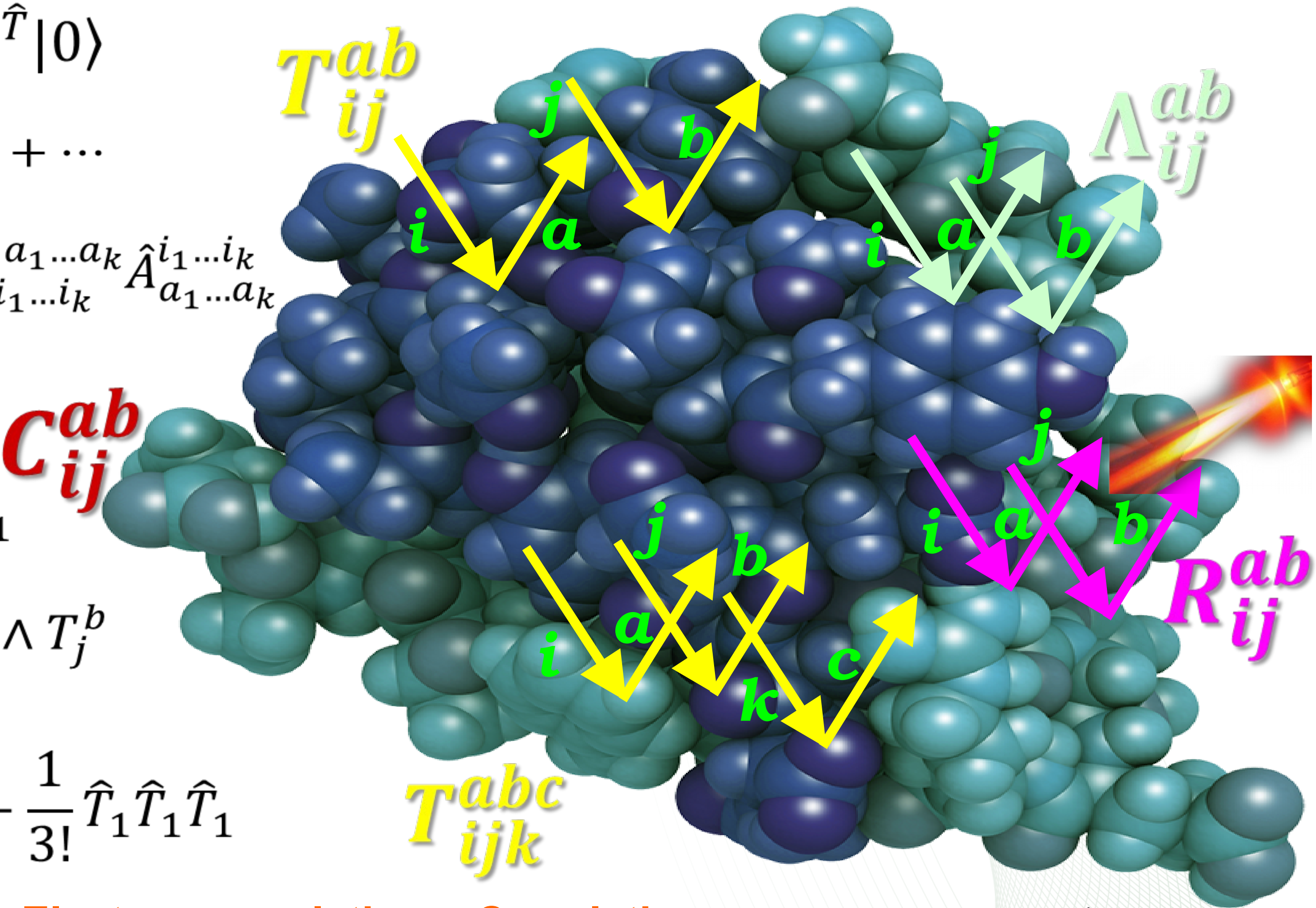
$$\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3 + \cdots$$

$$\hat{T}_k = \frac{1}{k!\,k!}\sum_{\substack{a_1\ldots a_k\\ i_1\ldots i_k}} T^{a_1\ldots a_k}_{i_1\ldots i_k}\hat{A}^{i_1\ldots i_k}_{a_1\ldots a_k}$$

$$\hat{C}_2 = \hat{T}_2 + \frac{1}{2!}\hat{T}_1\hat{T}_1$$

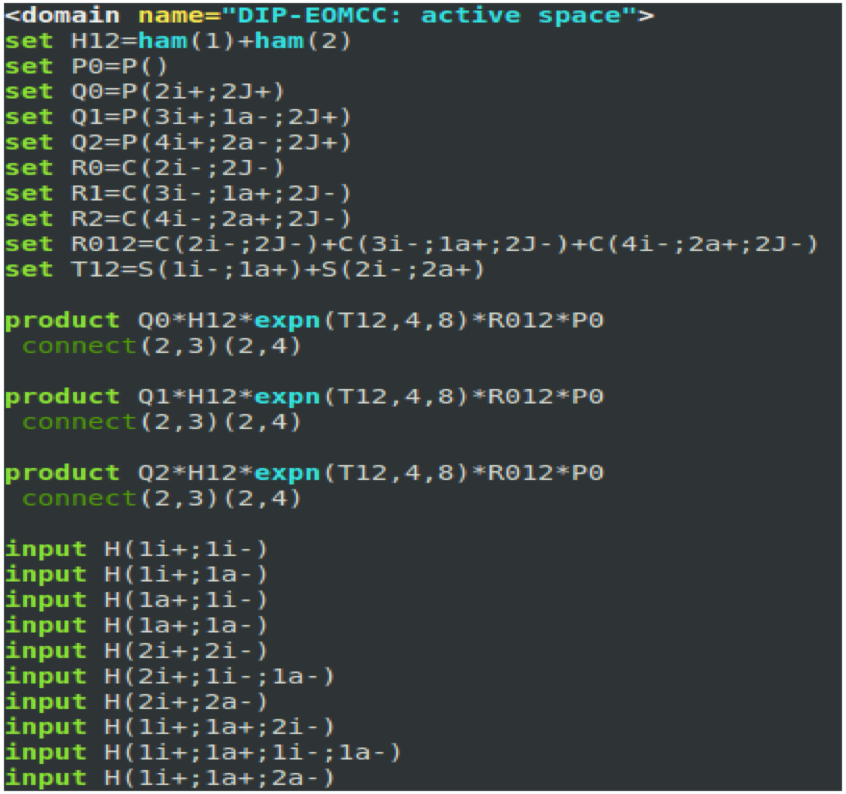
$$C^{ab}_{ij} = T^{ab}_{ij} + T^a_i \wedge T^b_j$$

$$\hat{C}_3 = \hat{T}_3 + \hat{T}_2\hat{T}_1 + \frac{1}{3!}\hat{T}_1\hat{T}_1\hat{T}_1$$



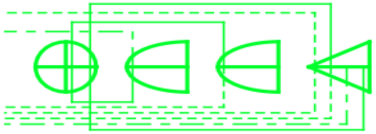
Electron correlation = Correlation between hole-particle excitations

DiaGen: Automated Design and Implementation

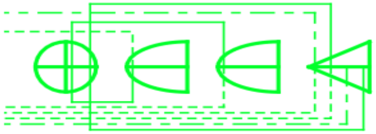


```
<domain name="DIP-EOMCC: active space">
set H12=ham(1)+ham(2)
set P0=P()
set Q0=P(2i+;2J+)
set Q1=P(3i+;1a-;2J+)
set Q2=P(4i+;2a-;2J+)
set R0=C(2i-;2J-)
set R1=C(3i-;1a+;2J-)
set R2=C(4i-;2a+;2J-)
set R012=C(2i-;2J-)+C(3i-;1a+;2J-)+C(4i-;2a+;2J-)
set T12=S(1i-;1a+)+S(2i-;2a+)

product Q0*H12*expn(T12,4,8)*R012*P0
 connect(2,3)(2,4)

product Q1*H12*expn(T12,4,8)*R012*P0
 connect(2,3)(2,4)

product Q2*H12*expn(T12,4,8)*R012*P0
 connect(2,3)(2,4)

input H(1i+;1i-)
input H(1i+;1a-)
input H(1a+;1i-)
input H(1a+;1a-)
input H(2i+;2i-)
input H(2i+;1i-;1a-)
input H(2i+;2a-)
input H(1i+;1a+;2i-)
input H(1i+;1a+;1i-;1a-)
input H(1i+;1a+;2a-)
```

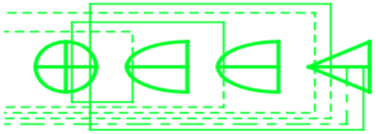
960



962



964



$$192.3.896 : Z^{A_1^b}_{I_1^a I_2^a I_1^b} + = H^{l_1^a K_1^a}_{d_1^a, d_2^a} S^{d_1^a}_{I_1^a} S^{d_2^a}_{I_2^a} C^{A_1^b}_{I_1^b, l_1^a K_1^a} \cdot +1/2 \tag{285}$$

$$198.1.932 : Z^{A_1^b}_{I_1^a I_2^a I_1^b} + = H^{l_1^b, l_2^b}_{d_1^b, d_2^b} S^{d_1^b}_{I_1^b} S^{d_2^b}_{l_1^b} C^{A_1^b}_{I_1^a I_2^a, l_2^b} \tag{286}$$

$$198.2.933 : Z^{A_1^b}_{I_1^a I_2^a I_1^b} + = H^{l_1^b, l_1^a}_{d_1^a, d_1^b} S^{d_1^a}_{I_1^a} S^{d_1^b}_{l_1^b} C^{A_1^b}_{I_2^a I_1^b, l_1^a} \tag{287}$$

$$198.4.935 : Z^{A_1^b}_{I_1^a I_2^a I_1^b} + = H^{l_1^a, l_1^b}_{d_1^b, d_1^a} S^{d_1^b}_{I_1^b} S^{d_1^a}_{l_1^a} C^{A_1^b}_{I_1^a I_2^a, l_1^b} \tag{288}$$

$$198.5.936 : Z^{A_1^b}_{I_1^a I_2^a I_1^b} + = H^{l_1^a, l_2^a}_{d_1^a, d_2^a} S^{d_1^a}_{I_1^a} S^{d_2^a}_{l_1^a} C^{A_1^b}_{I_2^a I_1^b, l_2^a} \tag{289}$$

$$202.1.946 : Z^{A_1^b}_{I_1^a I_2^a I_1^b} + = H^{l_1^b, K_1^a}_{d_1^a d_1^b} S^{A_1^b}_{l_1^b} S^{d_1^a d_1^b}_{I_1^a I_1^b} C_{I_2^a, K_1^a} \tag{290}$$

$$324.85.1.1.3.1.0.20333376.09 : Z^{l_1^b}_{I_1^a I_2^a i_1^b} + = H^{l_1^b, l_2^b}_{i_1^b, d_1^b} C^{d_1^b}_{I_1^a I_2^a, l_2^b} \cdot -1. \tag{447}$$

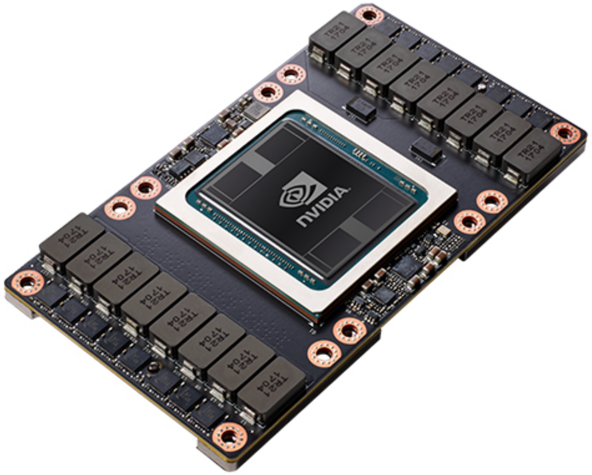
$$331.86.1.1.2.1.0.10042704.09 : Z^{l_1^b}_{I_1^a I_2^a i_1^b} + = H^{l_1^b, K_1^a}_{I_1^a, d_1^b} C^{d_1^b}_{I_2^a i_1^b, K_1^a} \cdot -1. \tag{448}$$

$$325.85.1.1.3.1.0.20333376.09 : Z^{l_1^b}_{I_1^a I_2^a i_1^b} + = H^{l_1^b, l_1^a}_{i_1^b, d_1^a} C^{d_1^a}_{I_1^a I_2^a, l_1^a} \cdot -1. \tag{449}$$

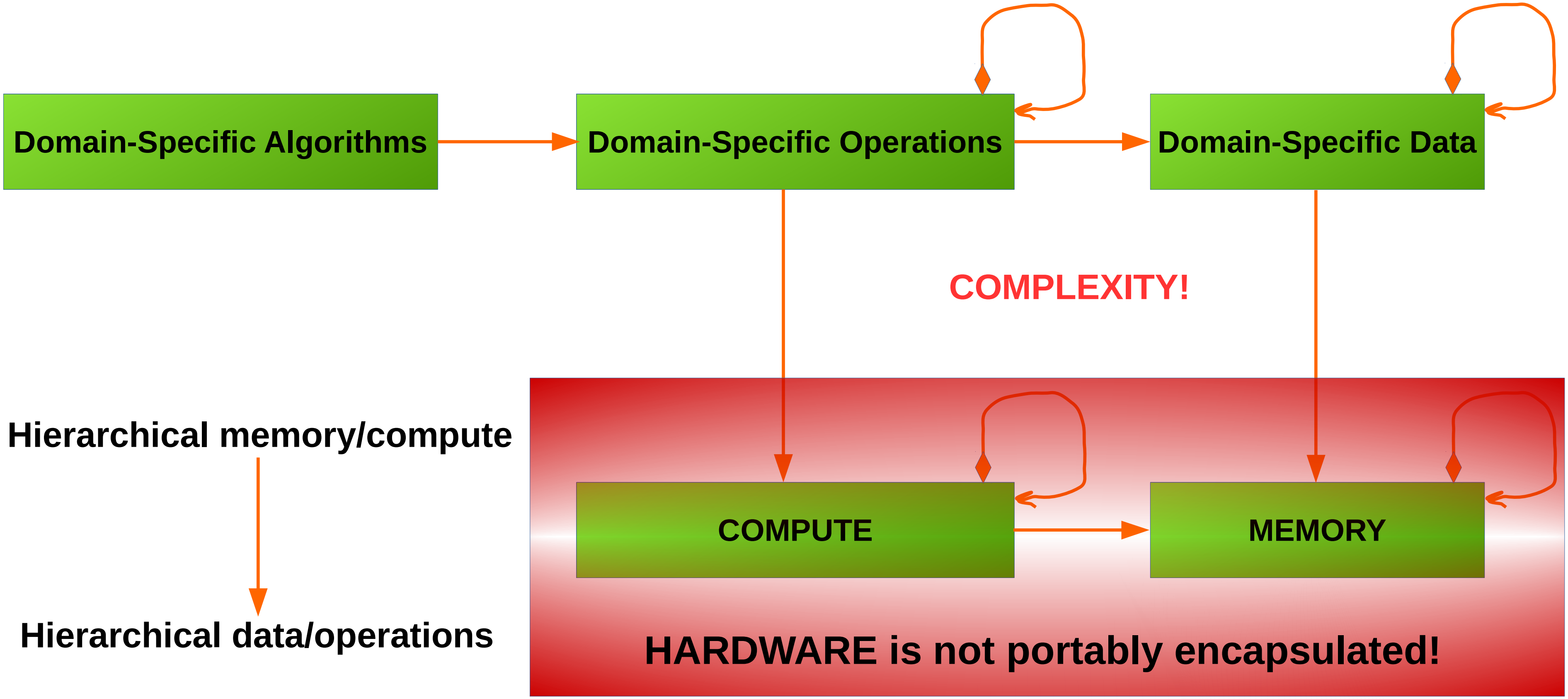
$$821.177.2.1.2.1.0.49593600.07 : Z^{l_1^b}_{I_1^a I_2^a i_1^b} + = S^{d_1^b}_{i_1^b} R^{l_1^b}_{I_1^a I_2^a, d_1^b} \cdot -1. \tag{450}$$

$$938.199.1.2.2.2.0.11716488.10 : R^{l_1^b, K_1^a}_{I_1^a i_1^b} + = H^{l_1^b, K_1^a}_{d_1^a d_1^b} S^{d_1^a d_1^b}_{I_1^a i_1^b} \tag{451}$$

Constantly Evolving HPC Hardware



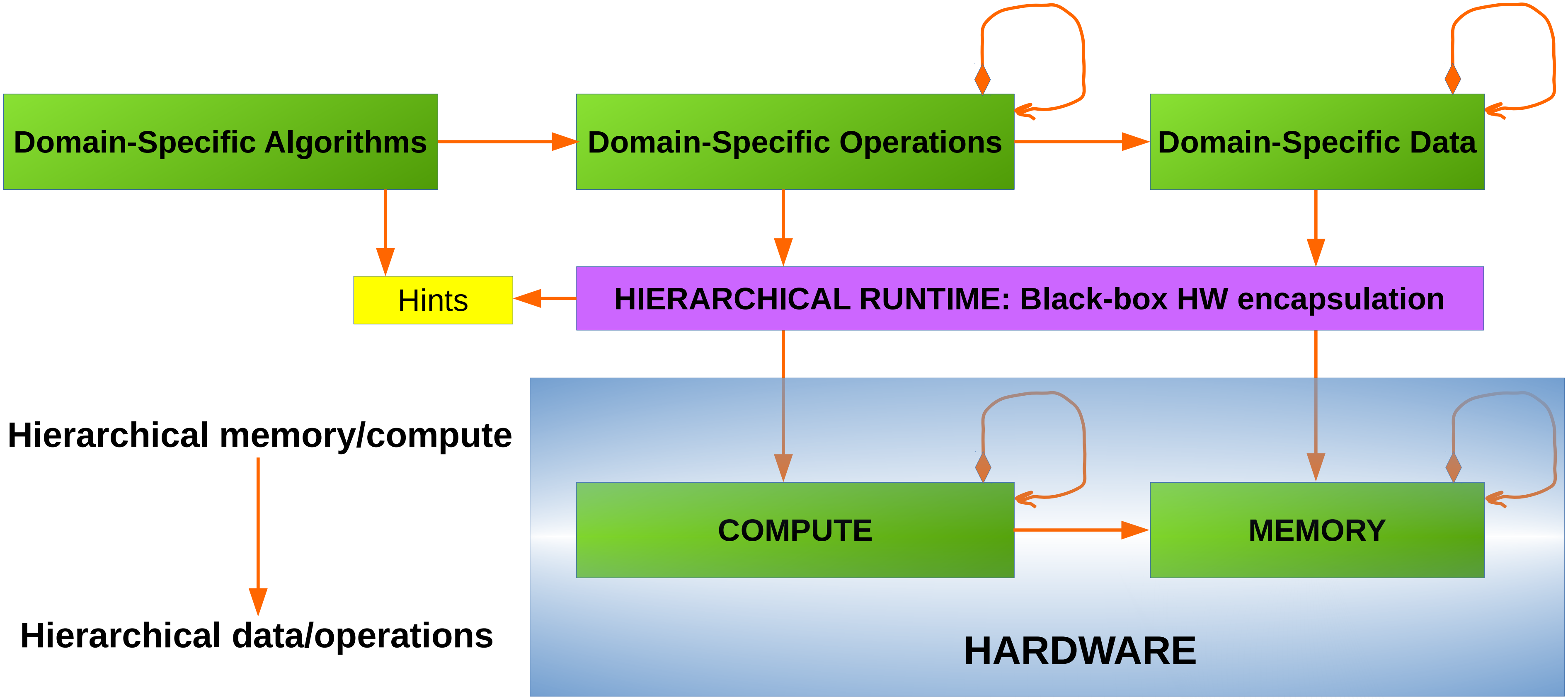
Lack of Portability


Domain-Specific Algorithms
Domain-Specific Operations
Domain-Specific Data
COMPLEXITY!
Hierarchical memory/compute
Hierarchical data/operations
COMPUTE
MEMORY
HARDWARE is not portably encapsulated!

PORTABILITY: Multiple targets, one code, maybe minor extension (not modification)

PERFORMANCE: Minimization/optimization of data movement to keep compute busy:
Optimal mapping of data and operations

Black-Box Portability

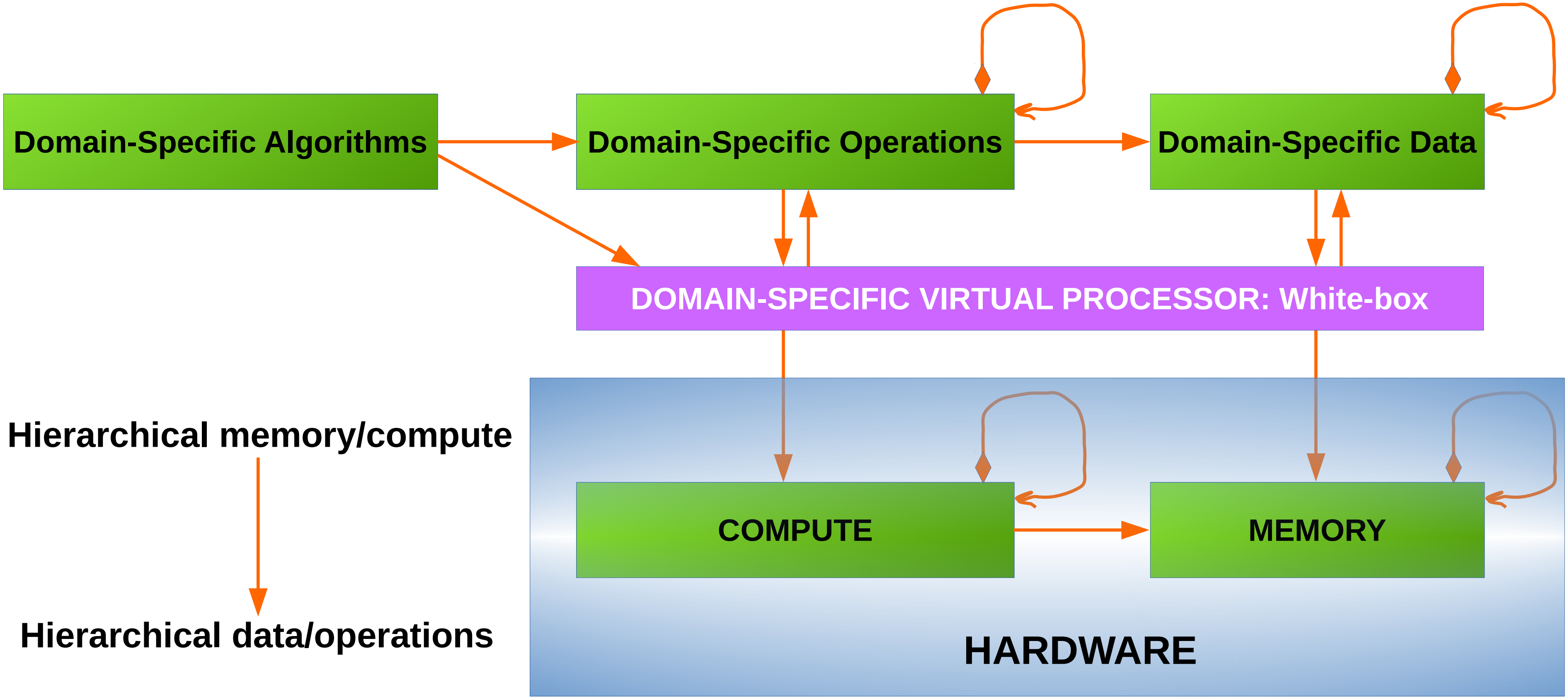

Domain-Specific Algorithms
Domain-Specific Operations
Domain-Specific Data
Hints
HIERARCHICAL RUNTIME: Black-box HW encapsulation
Hierarchical memory/compute
COMPUTE
MEMORY
Hierarchical data/operations
HARDWARE

PORTABILITY: Multiple targets, one code, maybe minor extension (not modification)

PERFORMANCE: Minimization/optimization of data movement to keep compute busy:
Optimal mapping of data and operations

OAK RIDGE National Laboratory | OAK RIDGE LEADERSHIP COMPUTING FACILITY

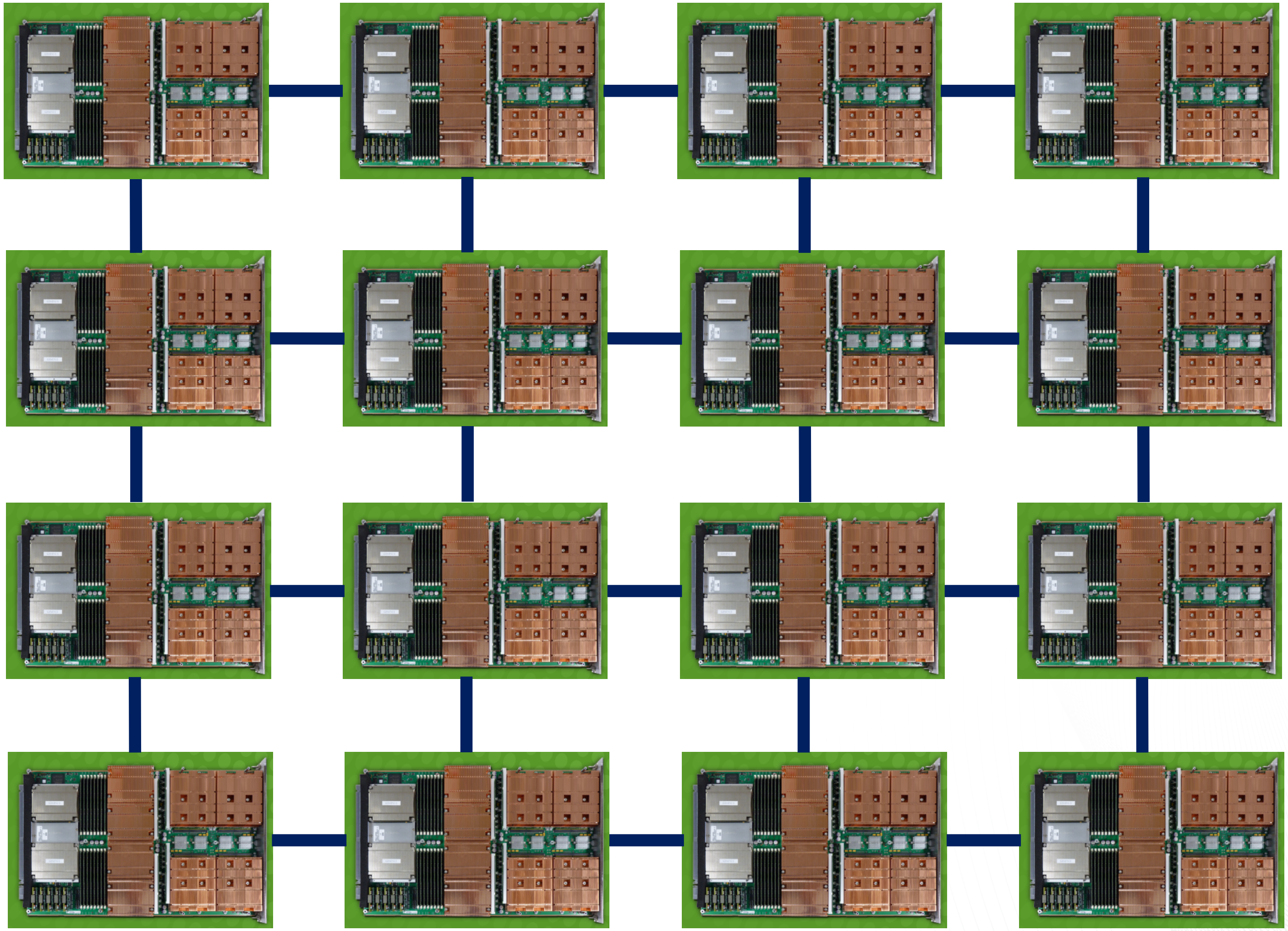
White-Box Portability


Domain-Specific Algorithms
Domain-Specific Operations
Domain-Specific Data
DOMAIN-SPECIFIC VIRTUAL PROCESSOR: White-box
Hierarchical memory/compute
Hierarchical data/operations
COMPUTE
MEMORY
HARDWARE

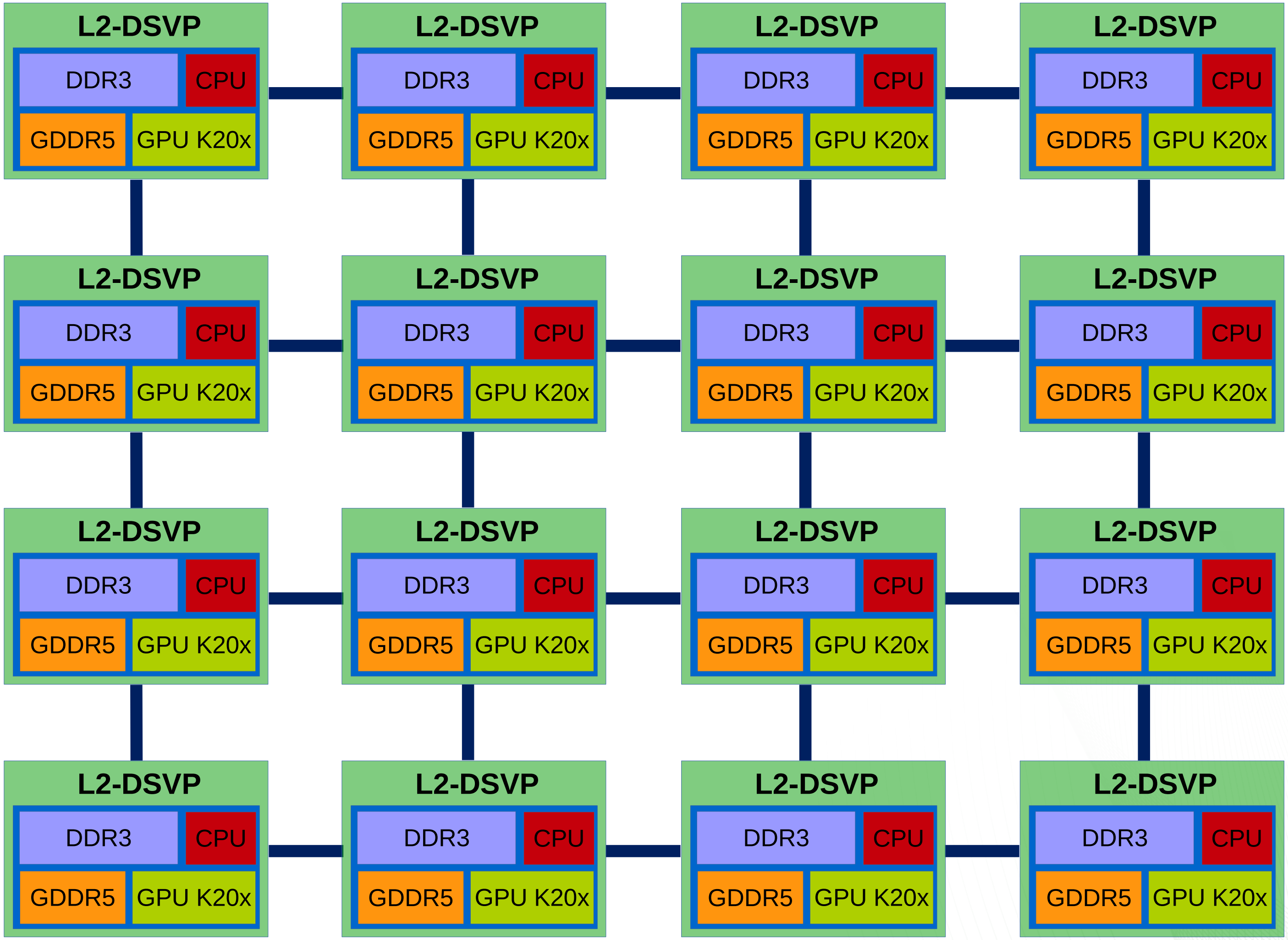
PORTABILITY: Multiple targets, one code, maybe minor extension (not modification)

PERFORMANCE: Minimization/optimization of data movement to keep compute busy:
Optimal mapping of data and operations

Global Virtualization: Hiding HPC Platform



Global Virtualization: Hiding HPC Platform



L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

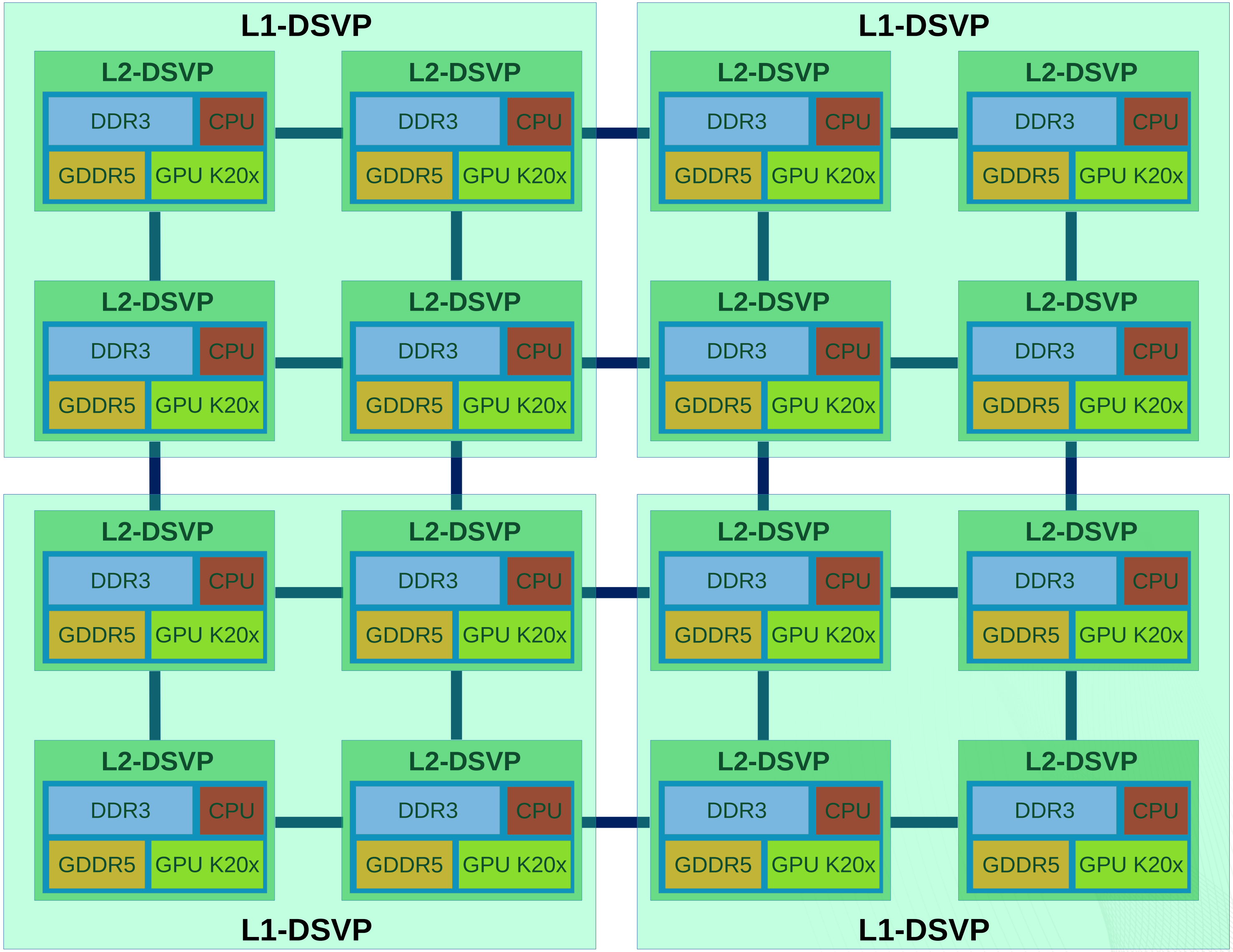
L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

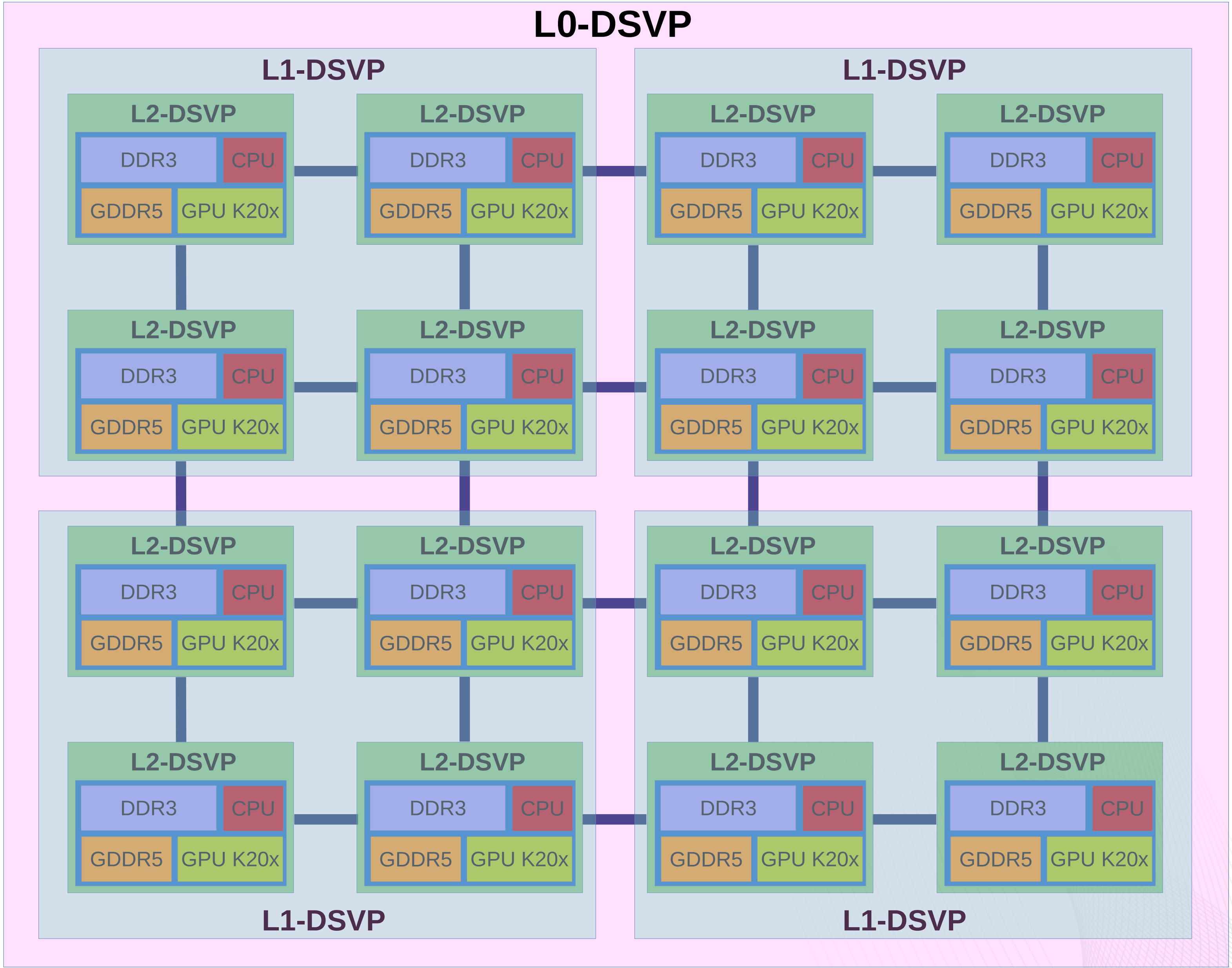
L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

L2-DSVP
DDR3 CPU
GDDR5 GPU K20x

Global Virtualization: Hiding HPC Platform



Global Virtualization: Hiding HPC Platform



L0-DSVP

L1-DSVP

L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

L1-DSVP

L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

L1-DSVP

L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

L1-DSVP

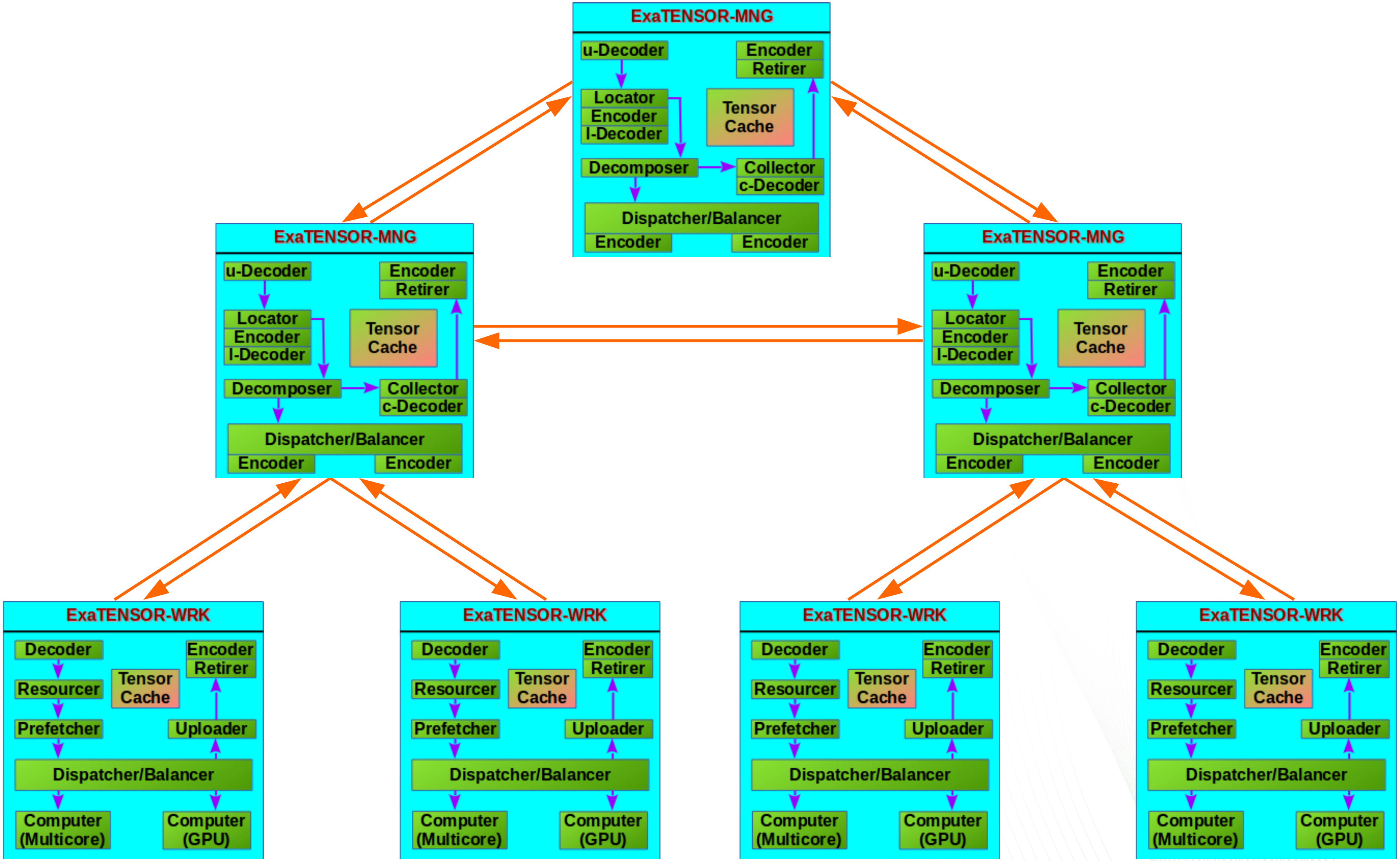
L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

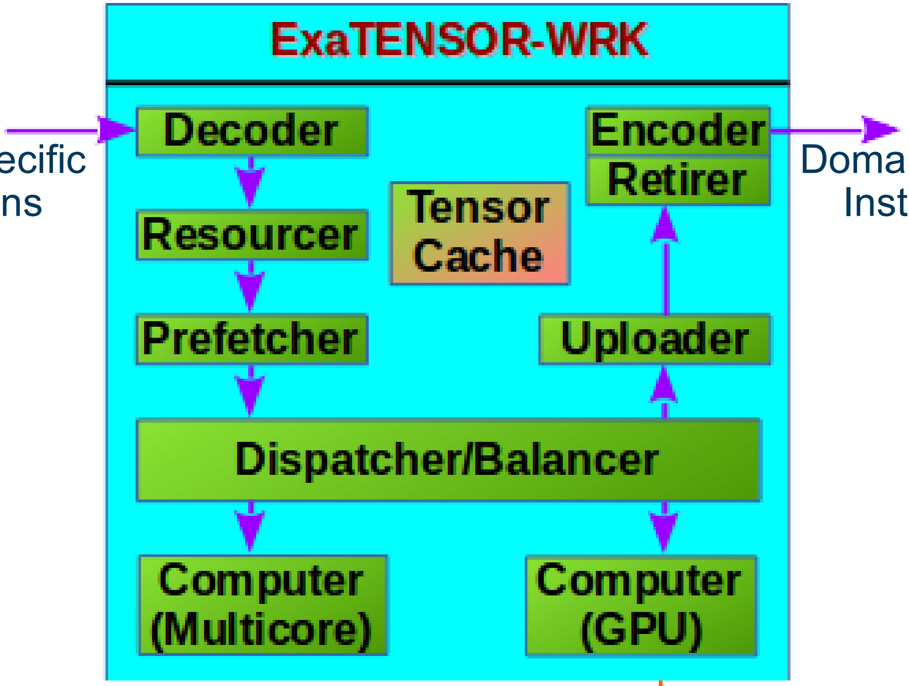
L2-DSVP: DDR3, CPU, GDDR5, GPU K20x

Hierarchical Virtualized HPC Platform

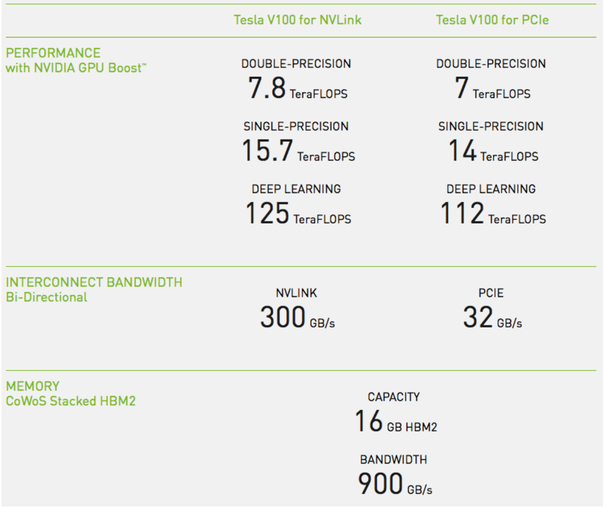


Node-Level Virtualization: Hiding Hardware

Domain-Specific Instructions



Domain-Specific Instructions



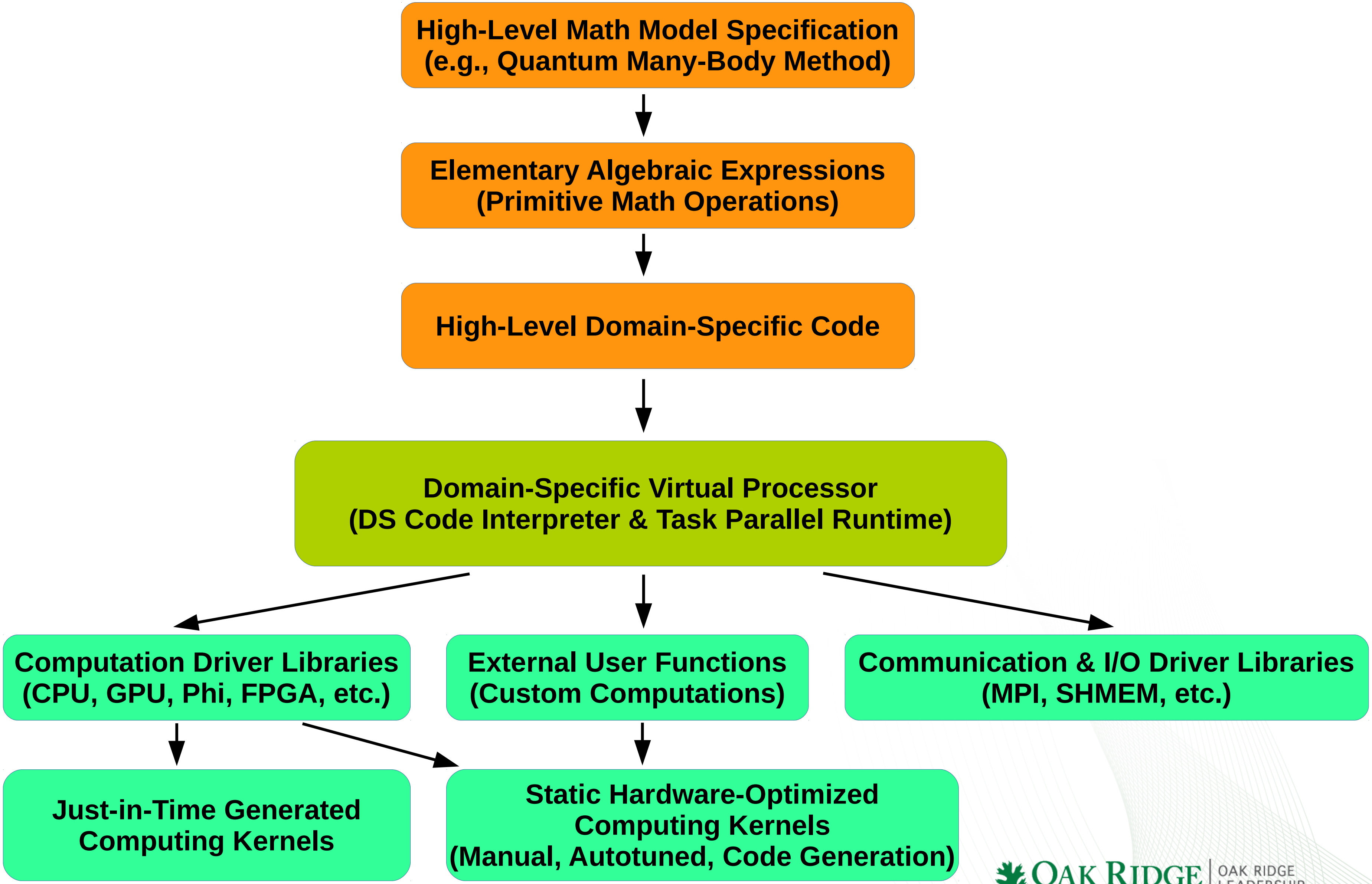
	Tesla V100 for NVLink	Tesla V100 for PCIe
PERFORMANCE with NVIDIA GPU Boost™	DOUBLE-PRECISION 7.8 TeraFLOPS	DOUBLE-PRECISION 7 TeraFLOPS
	SINGLE-PRECISION 15.7 TeraFLOPS	SINGLE-PRECISION 14 TeraFLOPS
	DEEP LEARNING 125 TeraFLOPS	DEEP LEARNING 112 TeraFLOPS
INTERCONNECT BANDWIDTH Bi-Directional	NVLINK 300 GB/s	PCIE 32 GB/s
MEMORY CoWoS Stacked HBM2	CAPACITY 16 GB HBM2	
	BANDWIDTH 900 GB/s	

TENSOR ALGEBRA DRIVER for Multicore CPU and NVIDIA GPU: TAL-SH library: (tensor algebra primitives = domain-specific microcode)

https://github.com/DmitryLyakh/TAL_SH.git

$$\forall p,q,r,s:\quad T^{pq}_{rs} = L^{pai}_{bcd} R^{qbcd}_{rsai}$$

Portable Scalable Scientific Computing



High-Level Math Model Specification
(e.g., Quantum Many-Body Method)

Elementary Algebraic Expressions
(Primitive Math Operations)

High-Level Domain-Specific Code

Domain-Specific Virtual Processor
(DS Code Interpreter & Task Parallel Runtime)

Computation Driver Libraries
(CPU, GPU, Phi, FPGA, etc.)

External User Functions
(Custom Computations)

Communication & I/O Driver Libraries
(MPI, SHMEM, etc.)

Just-in-Time Generated
Computing Kernels

Static Hardware-Optimized
Computing Kernels
(Manual, Autotuned, Code Generation)