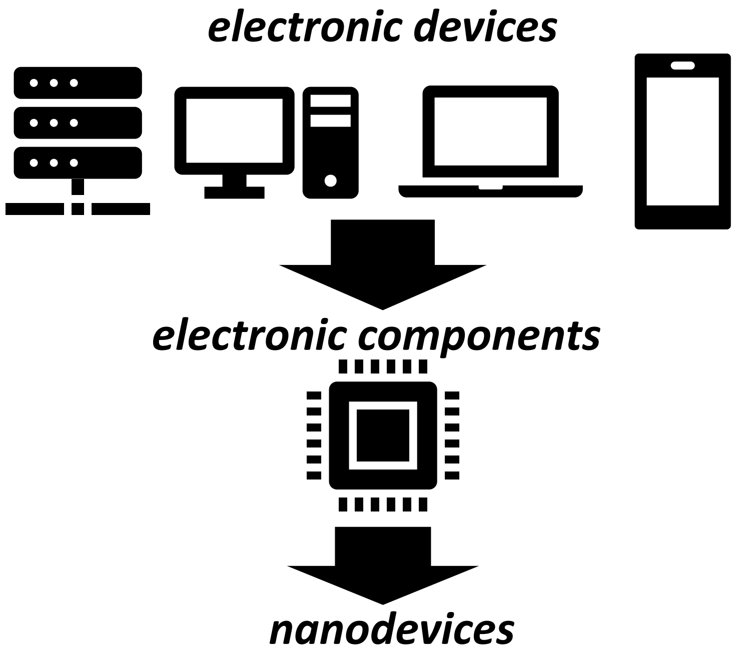
A. N. Ziogas, Tal Ben-Nun, G. Fernandéz, T. Schneider, M. Luisier, T. Hoefler

A Data-Centric Approach to Quantum Transport Simulations

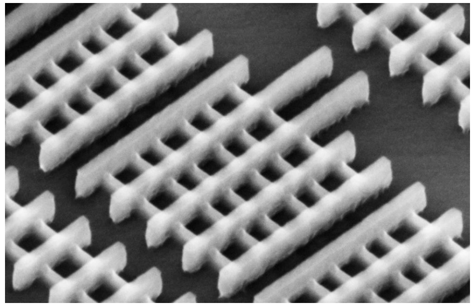
Motivation



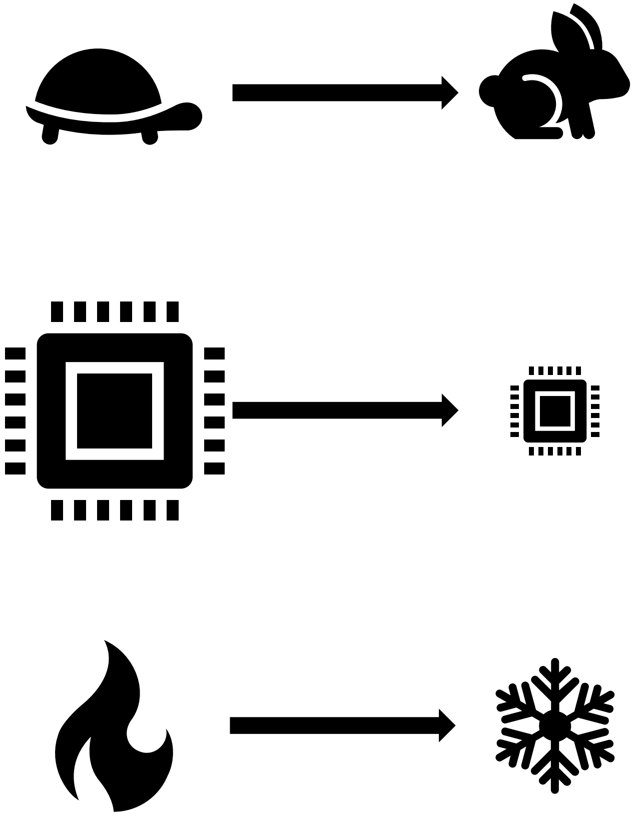
electronic devices

electronic components

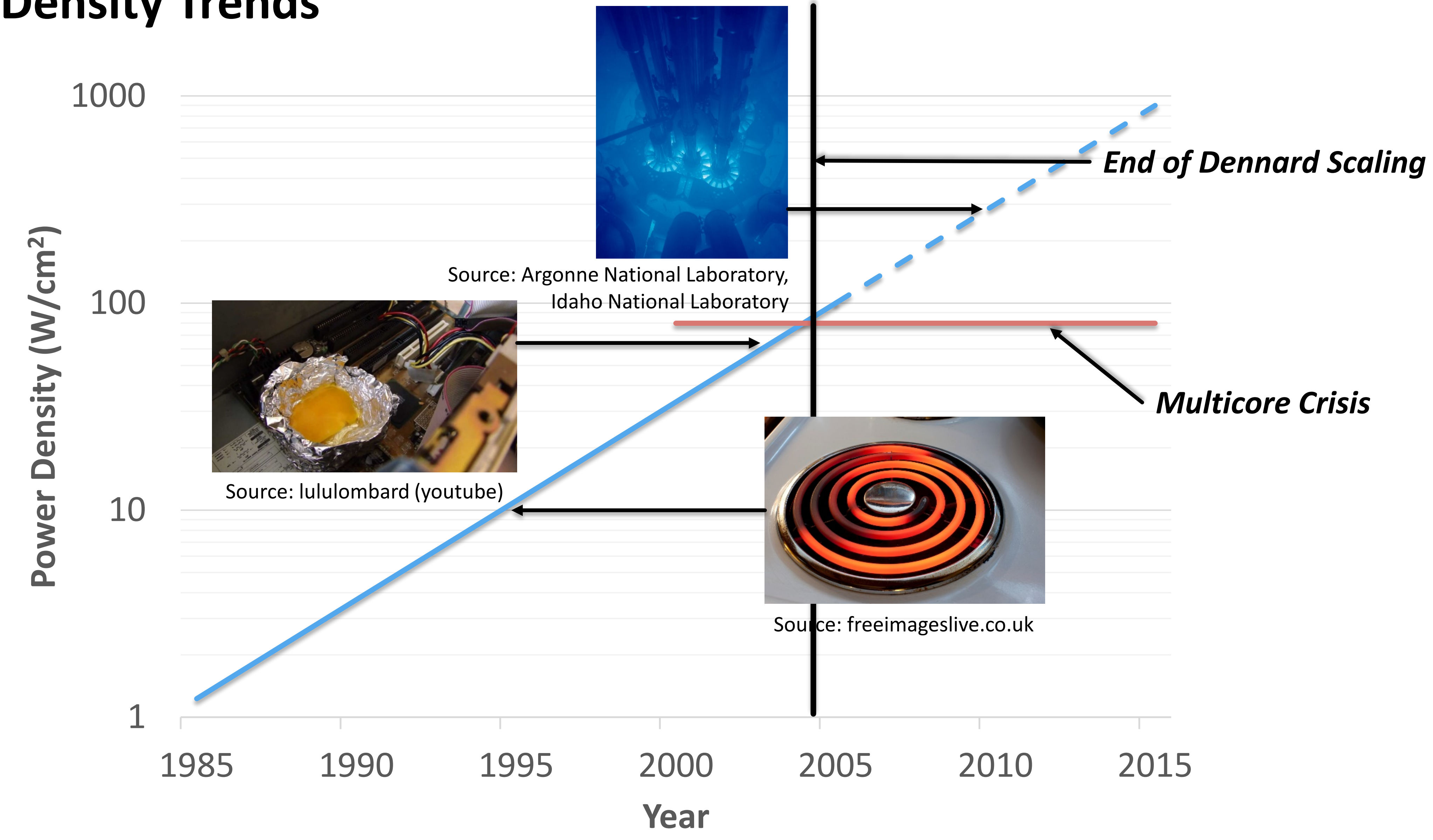
nanodevices



Source: Intel



CPU Power Density Trends



End of Dennard Scaling

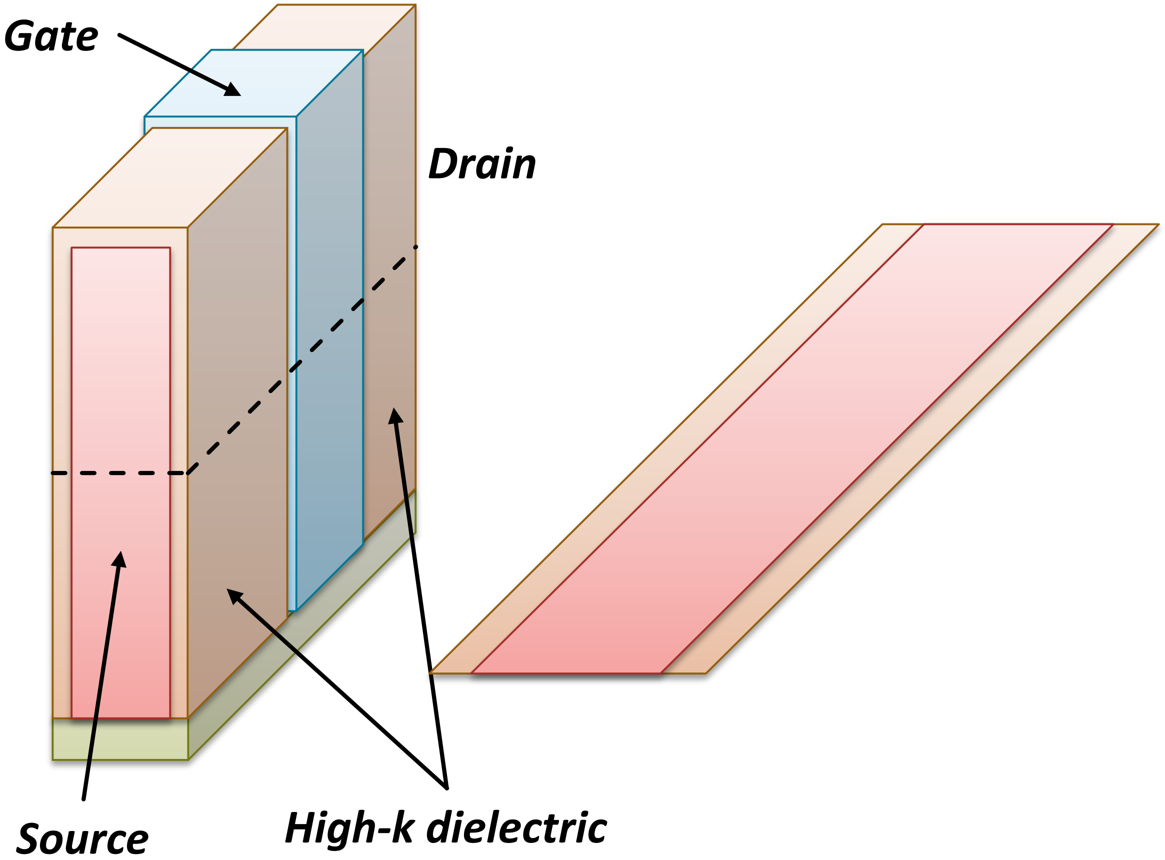
Multicore Crisis

Source: Argonne National Laboratory, Idaho National Laboratory

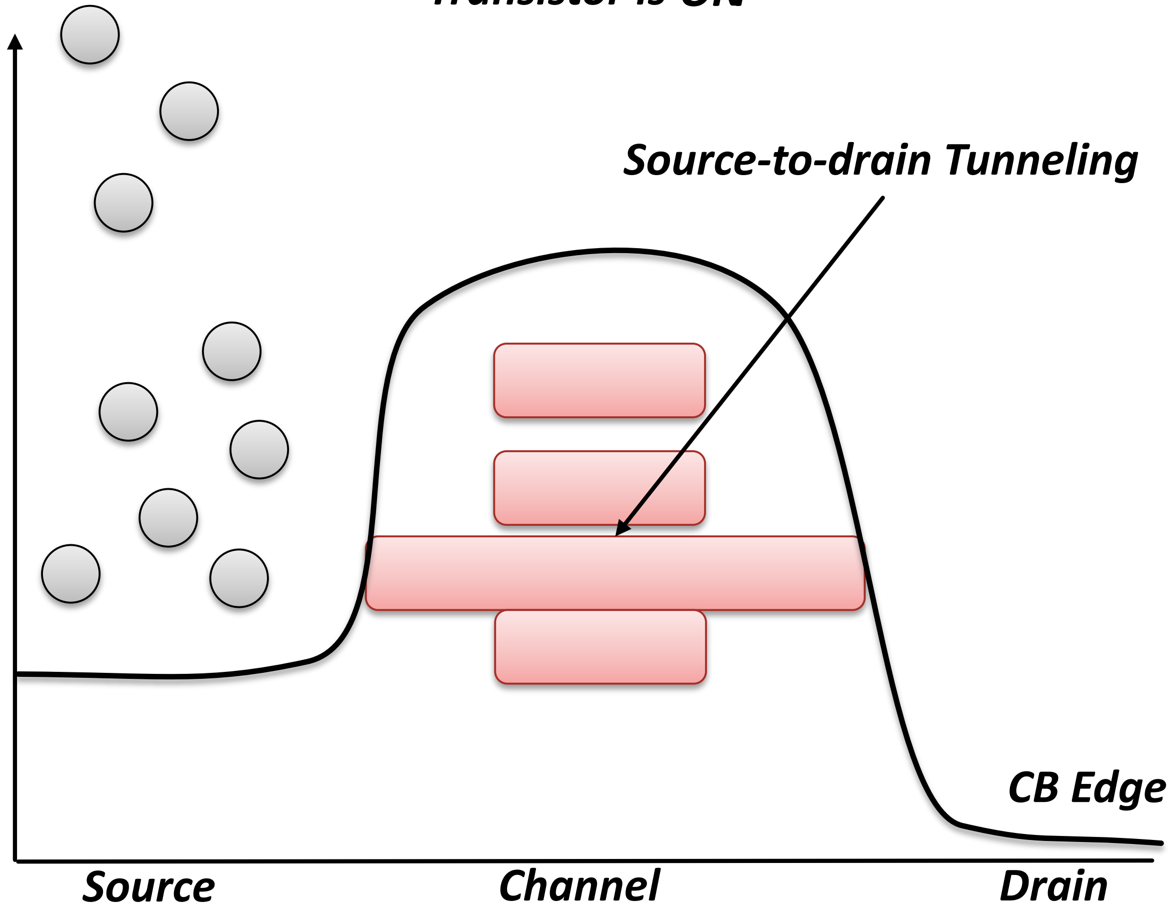
Source: lululombard (youtube)

Source: freeimageslive.co.uk

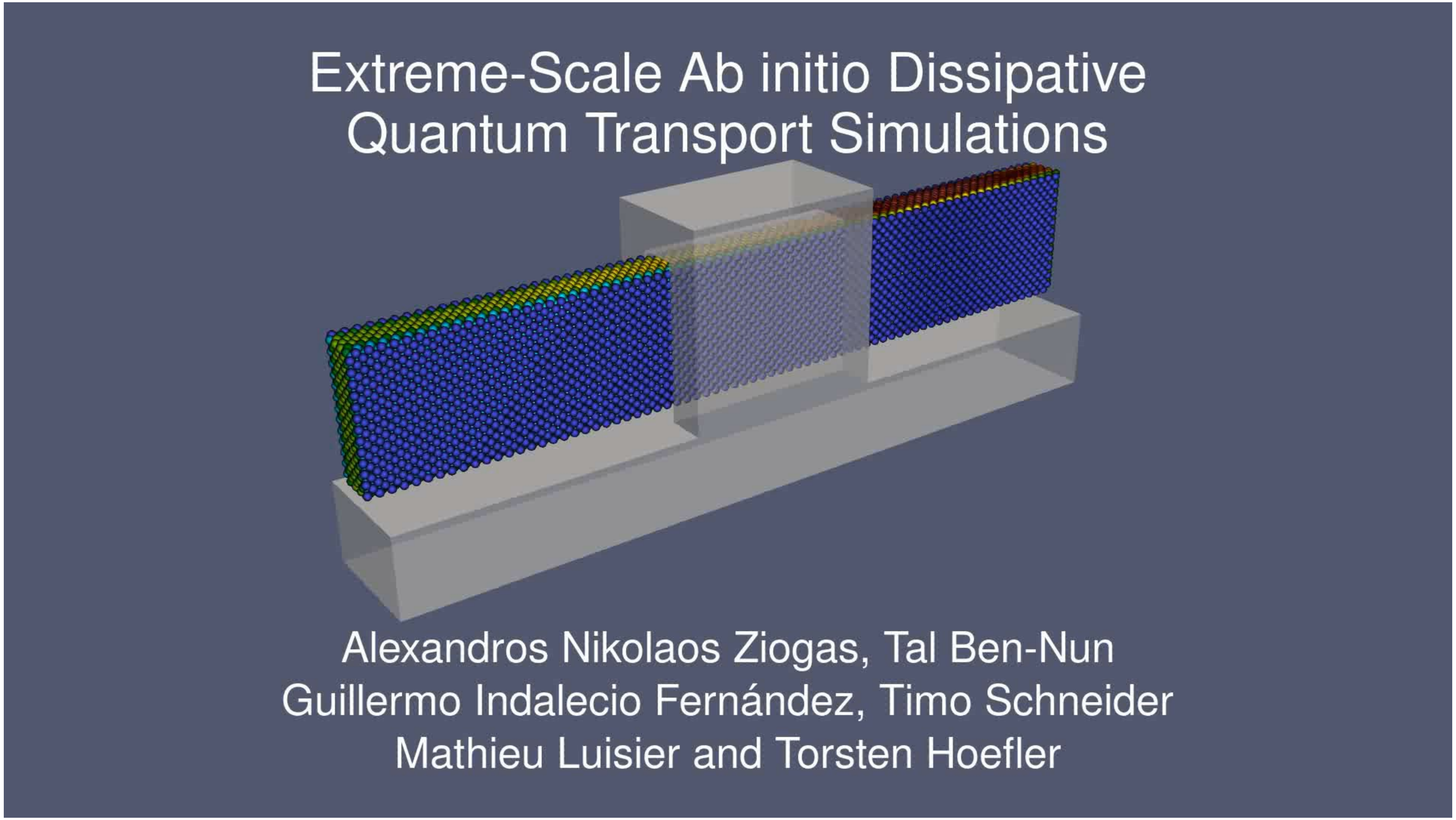
Quantum Mechanical Phenomena



Transistor is OFF



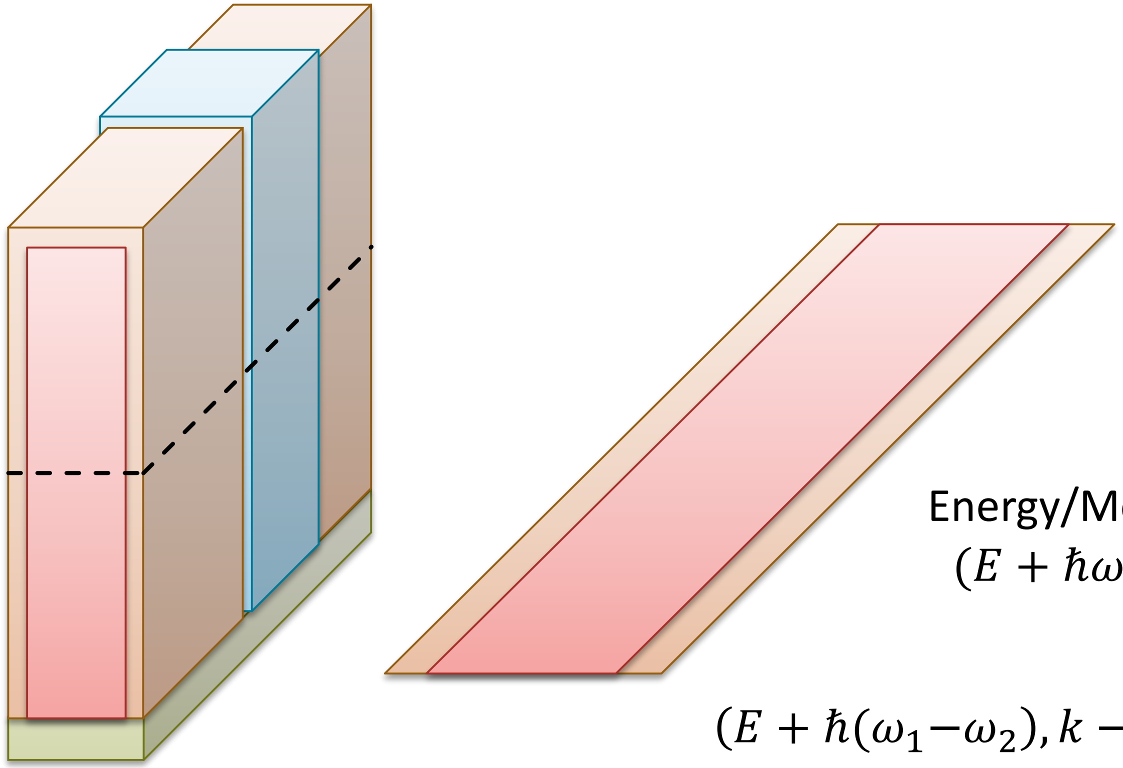
Characteristics of Nanotransistors



Extreme-Scale Ab initio Dissipative
Quantum Transport Simulations

Alexandros Nikolaos Ziogas, Tal Ben-Nun
Guillermo Indalecio Fernández, Timo Schneider
Mathieu Luisier and Torsten Hoefler

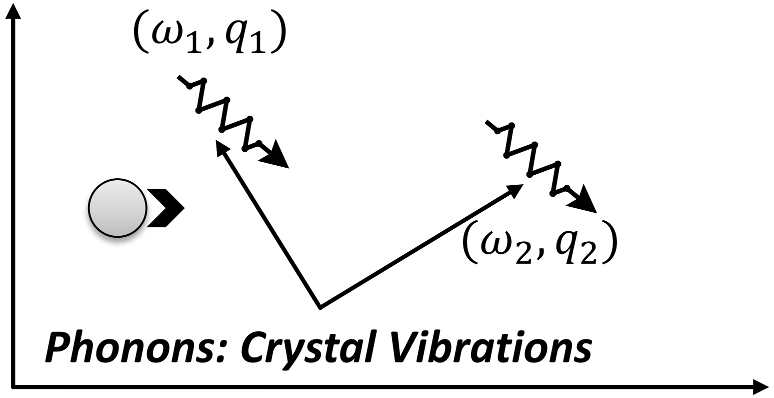
Quantum Transport Simulation



Ballistic Transport

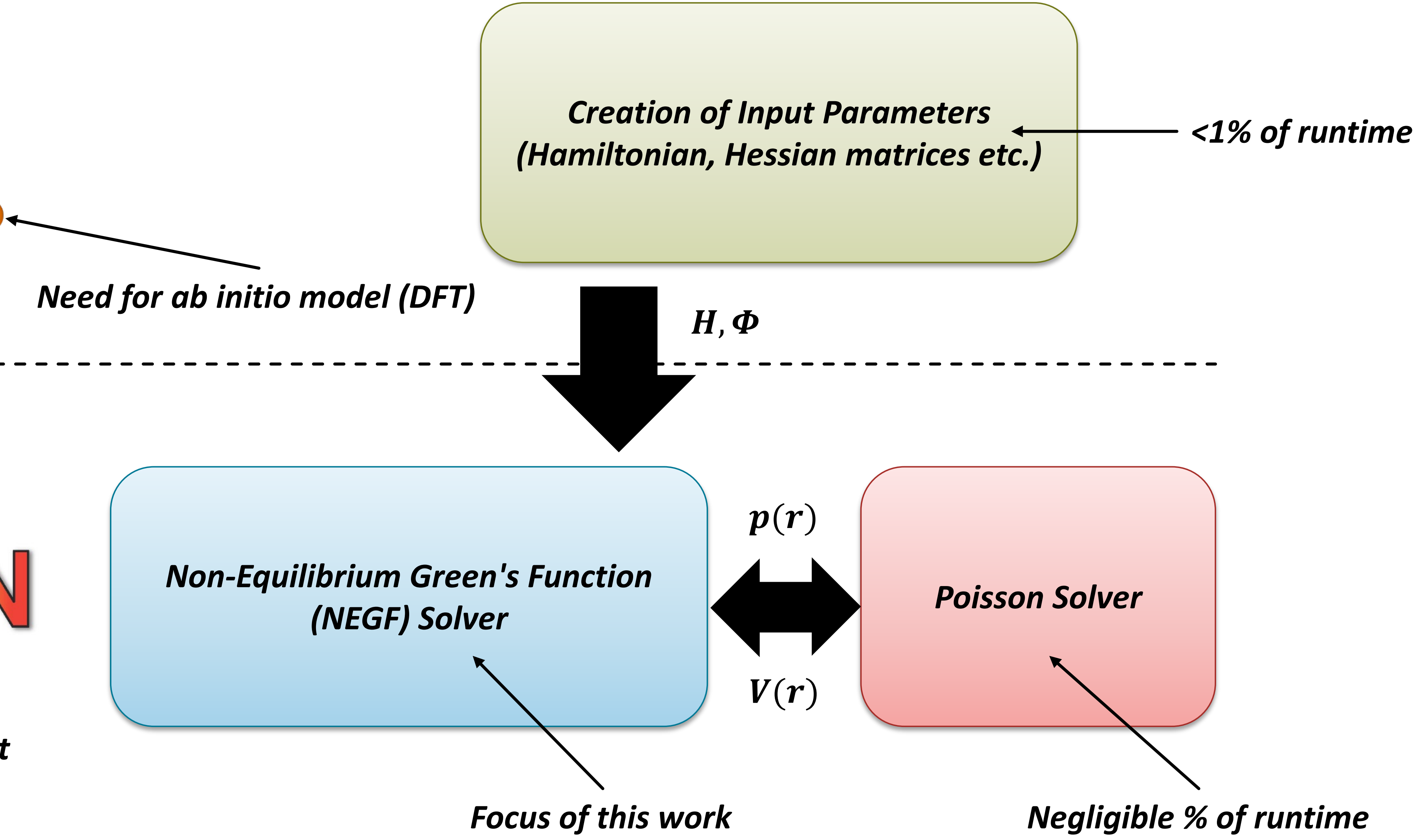
Energy/Momentum
(E, k)
Channel

Incoherent Transport


Energy/Momentum
$(E + \hbar\omega_1, k - q_1)$
(E, k)
$(E + \hbar(\omega_1 - \omega_2), k - q_1 - q_2)$
(ω_1, q_1)
(ω_2, q_2)
Channel

Phonons: Crystal Vibrations

Workflow of Quantum Transport Simulation



Need for ab initio model (DFT)

Creation of Input Parameters (Hamiltonian, Hessian matrices etc.)

<1% of runtime

H, Φ

2 times GB finalist

Non-Equilibrium Green's Function (NEGF) Solver

$p(r)$

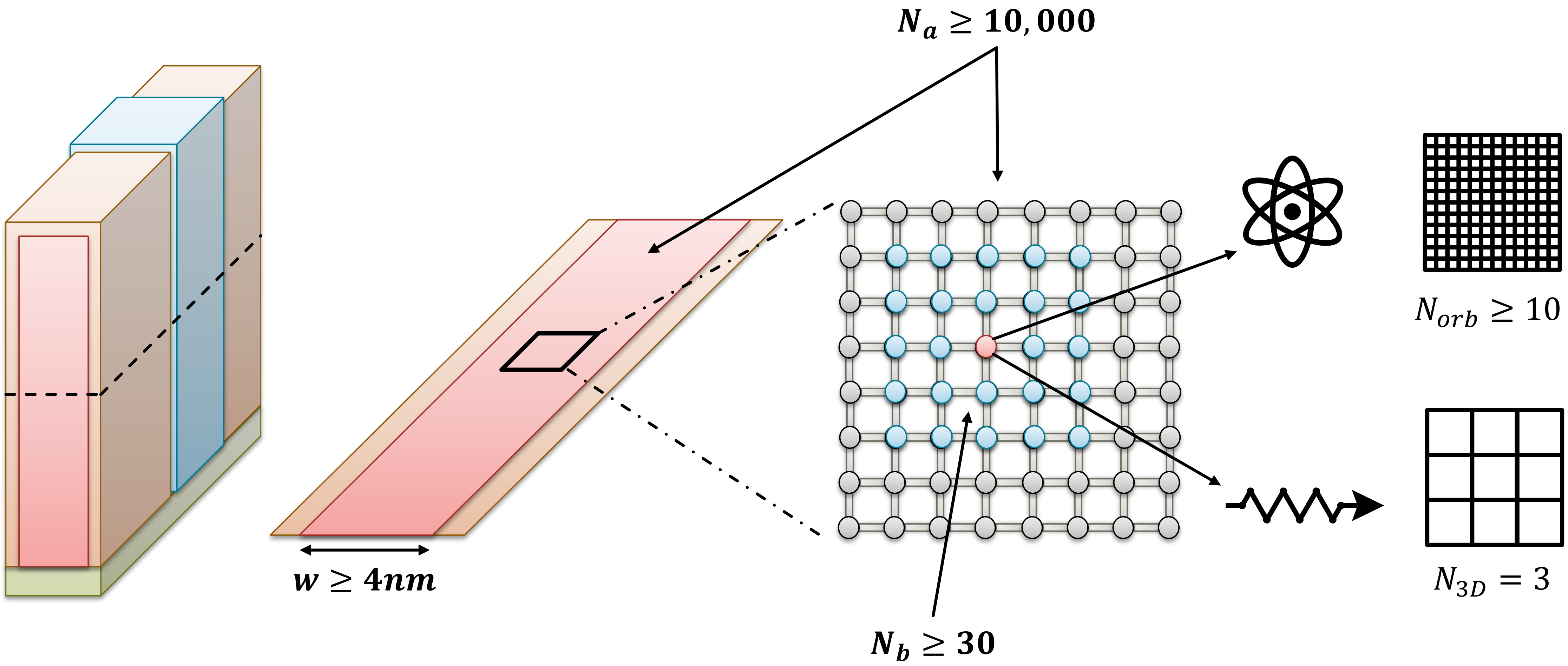
$V(r)$

Poisson Solver

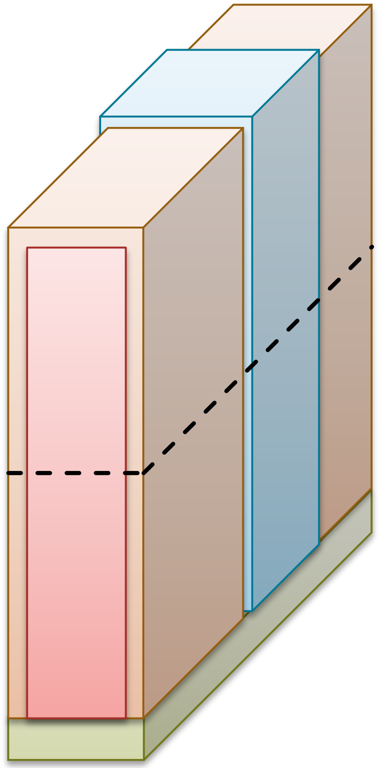
Focus of this work

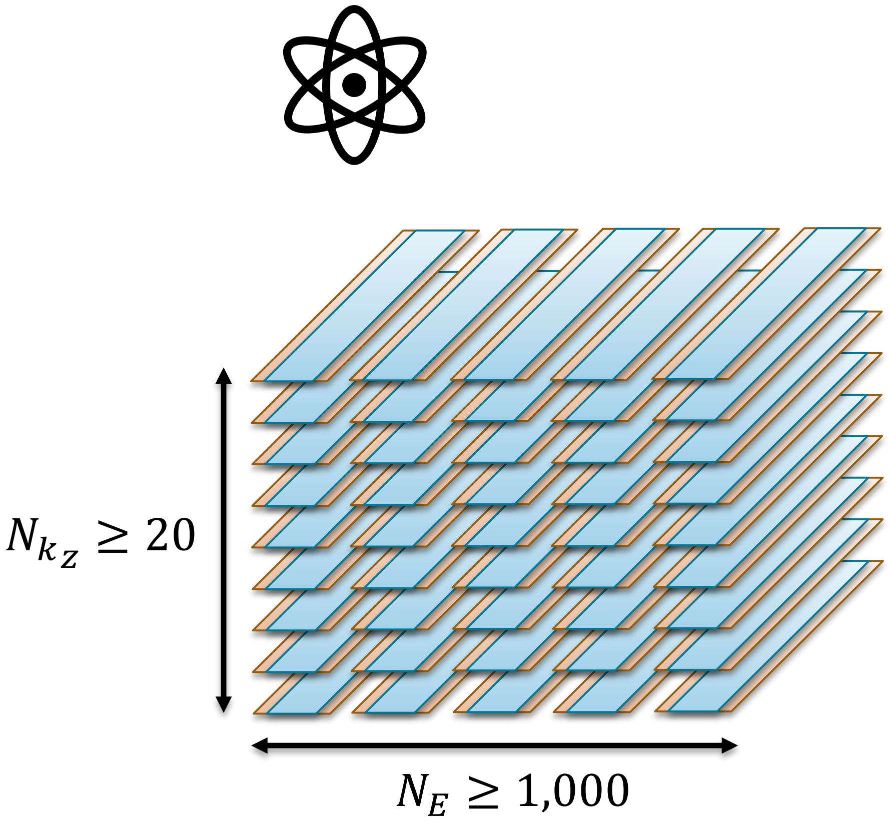
Negligible % of runtime

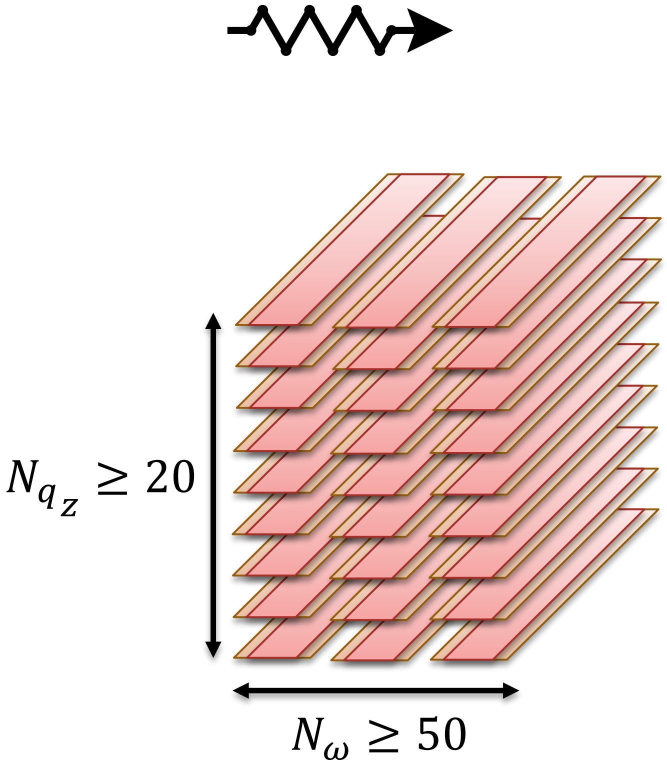
OMEN Model



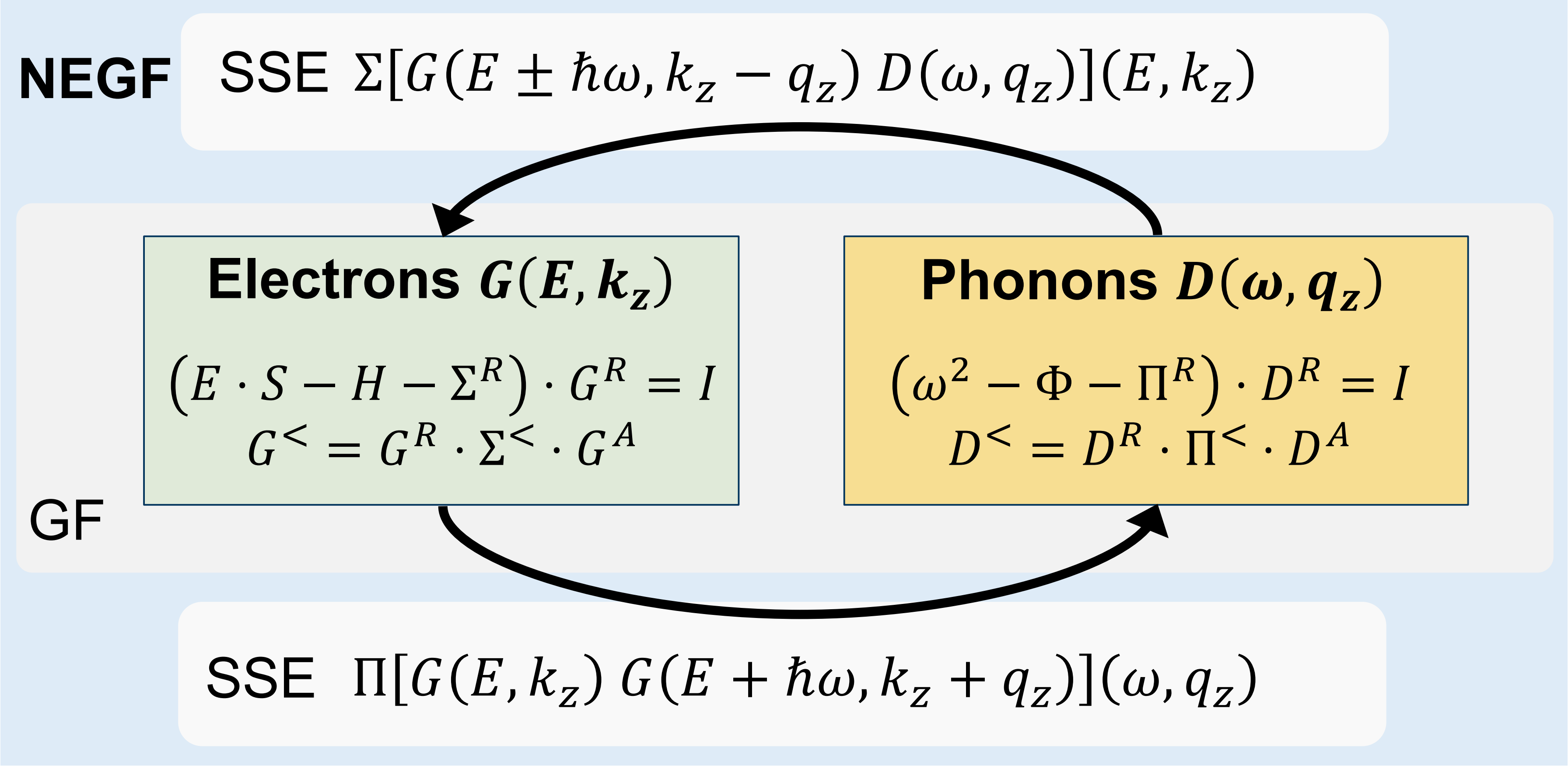
OMEN Model







NEGF Mathematical Formulation



NEGF

SSE $\Sigma[G(E \pm \hbar\omega, k_z - q_z)\, D(\omega, q_z)](E, k_z)$

Electrons $\boldsymbol{G(E, k_z)}$

$$\left(E \cdot S - H - \Sigma^R\right) \cdot G^R = I$$
$$G^< = G^R \cdot \Sigma^< \cdot G^A$$

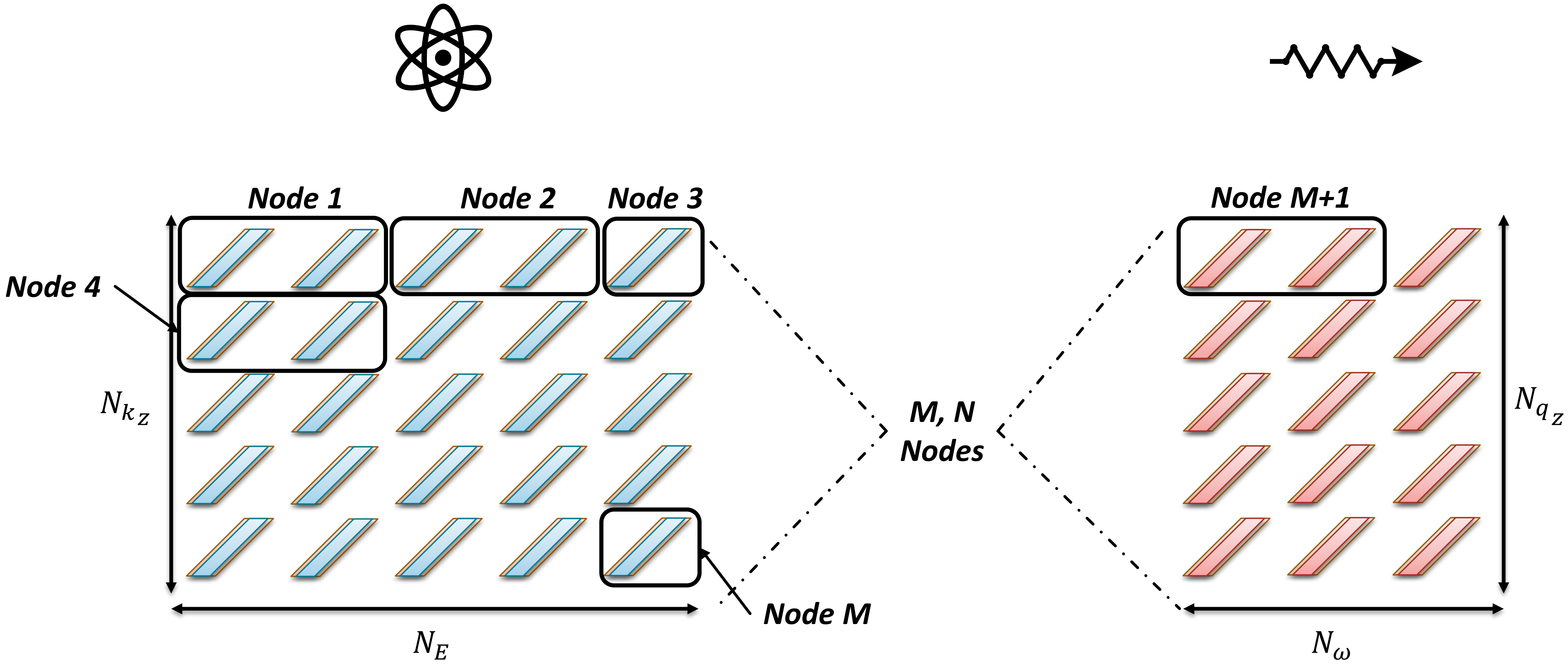
Phonons $\boldsymbol{D(\omega, q_z)}$

$$\left(\omega^2 - \Phi - \Pi^R\right) \cdot D^R = I$$
$$D^< = D^R \cdot \Pi^< \cdot D^A$$

GF

SSE $\Pi[G(E, k_z)\, G(E + \hbar\omega, k_z + q_z)](\omega, q_z)$

Assignment to Compute Resources



Green's Functions Phase

OMEN

Electrons $G(E,k_z)$

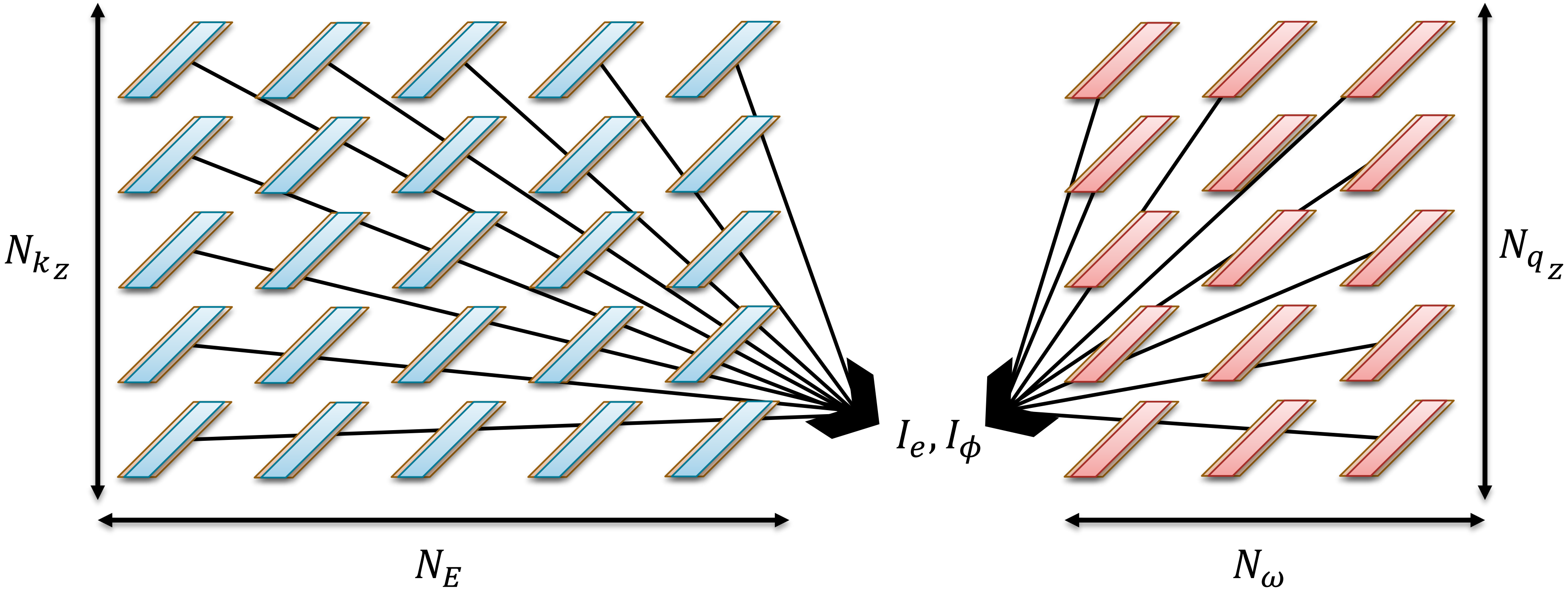
$$\left(E\cdot S-H-\Sigma^R\right)\cdot G^R=I$$
$$G^<=G^R\cdot\Sigma^<\cdot G^A$$

Phonons $D(\omega,q_z)$

$$\left(\omega^2-\Phi-\Pi^R\right)\cdot D^R=I$$
$$D^<=D^R\cdot\Pi^<\cdot D^A$$

GF

Embarrassingly parallel computation of G/D + Reduction for I



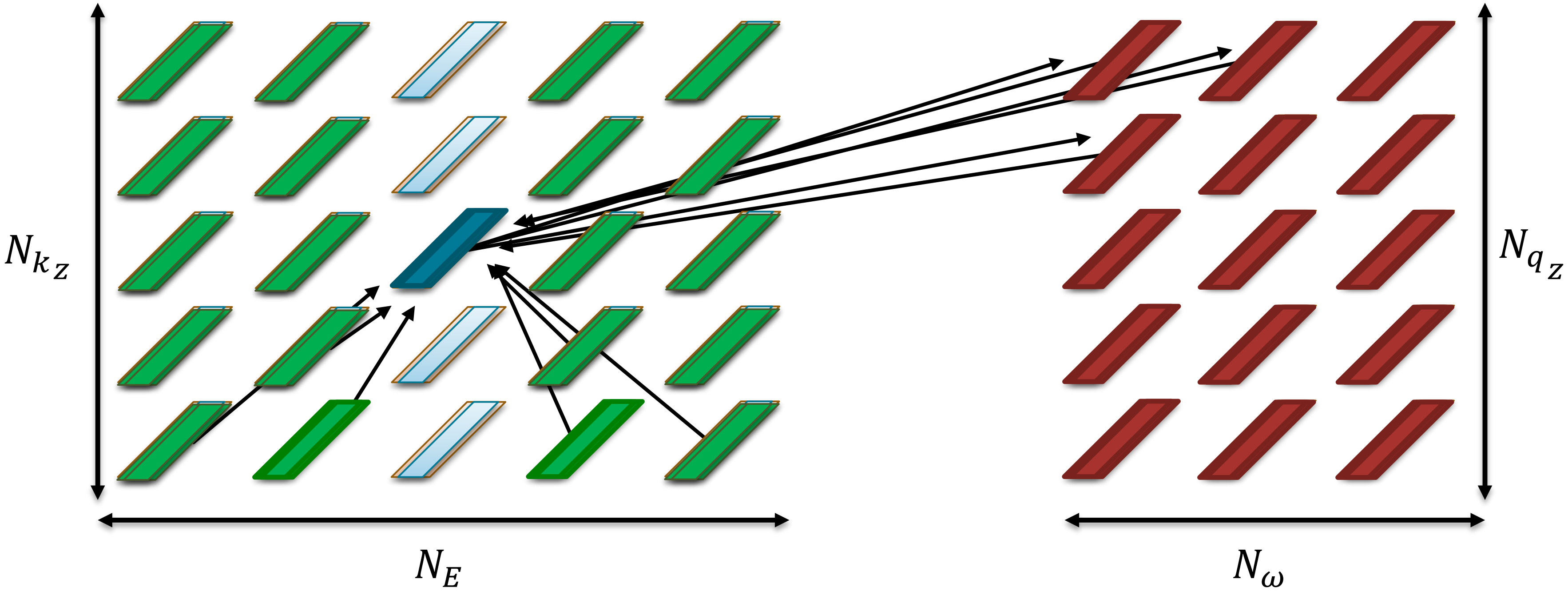
Scattering Self-Energies Phase

SSE $\Sigma[G(E \pm \hbar\omega, k_z - q_z)\ D(\omega, q_z)](E, k_z)$

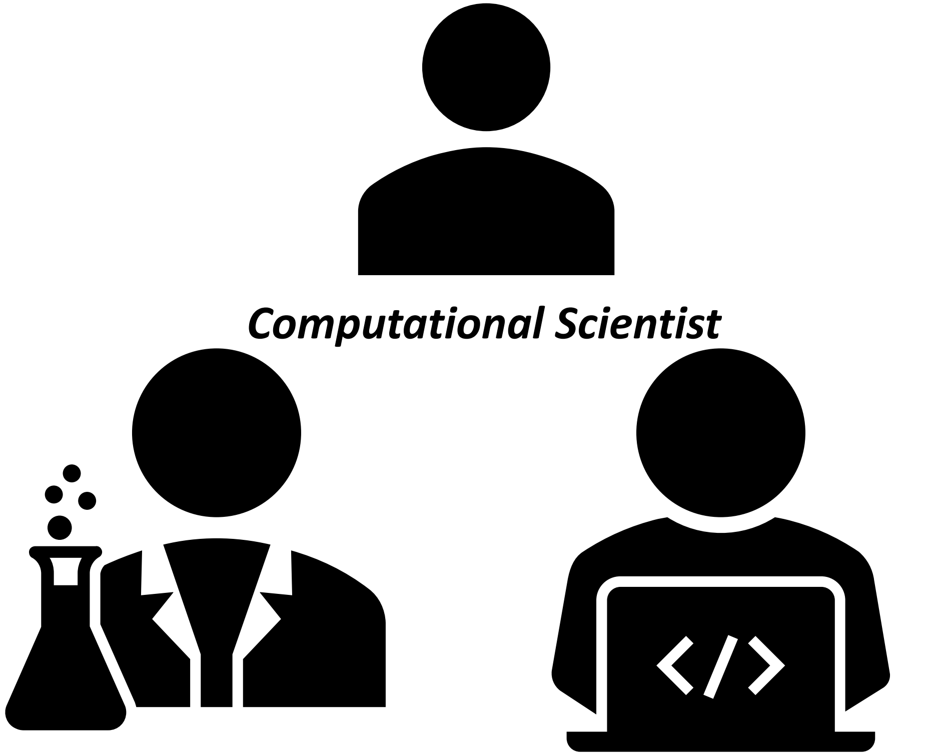
Stencil-like computation for Σ/Π

$\boldsymbol{2N_{q_z}N_\omega}$

SSE $\Pi[G(E, k_z)\ G(E + \hbar\omega, k_z + q_z)](\omega, q_z)$



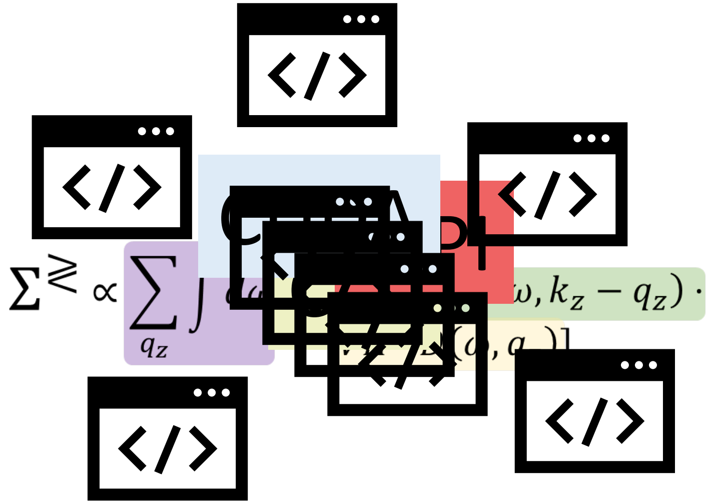
Motivation for a Data-Centric Framework



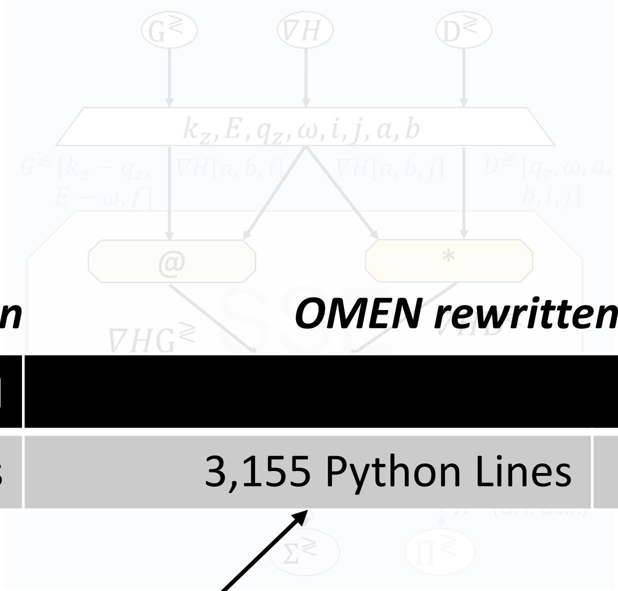
Computational Scientist

Domain Scientist

Performance Engineer



Data-Centric Framework



Data-Centric Intermediate Representation (SDFG)

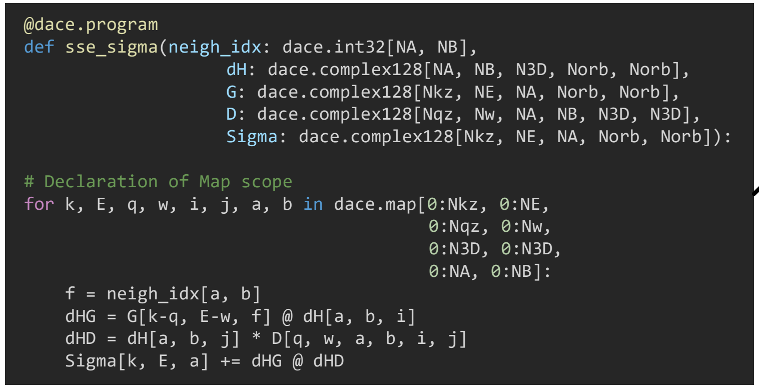
$\Sigma^{\gtrless} \propto \sum_{k_z,E,q_z} \int d\omega[\nabla H \cdot G(E - \hbar\omega, k_z - q_z) \cdot$
$\nabla H \cdot D(\omega$

Problem F

Original Application

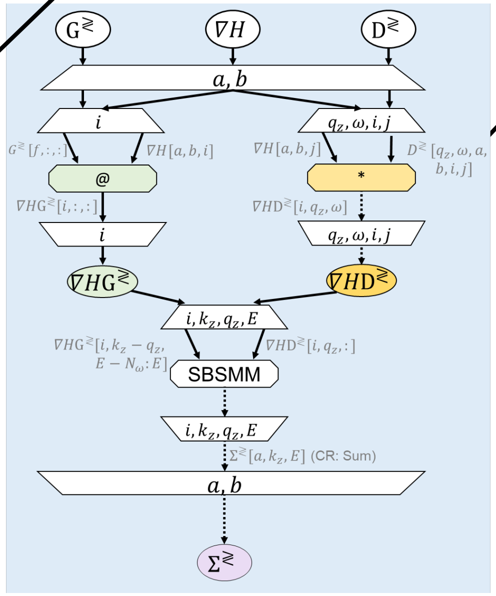
OMEN rewritten with the DaCe framework

OMEN	DaCe OMEN	
15,798 C++ Lines	3,155 Python Lines	2,015 SDFG Nodes



```
@dace.program
def sse_sigma(neigh_idx: dace.int32[NA, NB],
              dH: dace.complex128[NA, NB, N3D, Norb, Norb],
              G: dace.complex128[Nkz, NE, NA, Norb, Norb],
              D: dace.complex128[Nqz, Nw, NA, NB, N3D, N3D],
              Sigma: dace.complex128[Nkz, NE, NA, Norb, Norb]):

# Declaration of Map scope
for k, E, q, w, i, j, a, b in dace.map[0:Nkz, 0:NE,
                                        0:Nqz, 0:Nw,
                                        0:N3D, 0:N3D,
                                        0:NA, 0:NB]:

    f = neigh_idx[a, b]
    dHG = G[k-q, E-w, f] @ dH[a, b, i]
    dHD = dH[a, b, j] * D[q, w, a, b, i, j]
    Sigma[k, E, a] += dHG @ dHD
```

High-Level Program


G≷
∇H
D≷
a, b
i
q_z, ω, i, j
$G^{\gtrless}[f, :, :]$
$\nabla H[a, b, i]$
$\nabla H[a, b, j]$
$D^{\gtrless}[q_z, \omega, a, b, i, j]$
@
*
$\nabla HG^{\gtrless}[i, :, :]$
$\nabla HD^{\gtrless}[i, q_z, \omega]$
i
q_z, ω, i, j
∇HG≷
∇HD≷
i, k_z, q_z, E
$\nabla HG^{\gtrless}[i, k_z - q_z, E - N_\omega : E]$
$\nabla HD^{\gtrless}[i, q_z, :]$
SBSMM
i, k_z, q_z, E
$\Sigma^{\gtrless}[a, k_z, E]$ (CR: Sum)
a, b
Σ≷

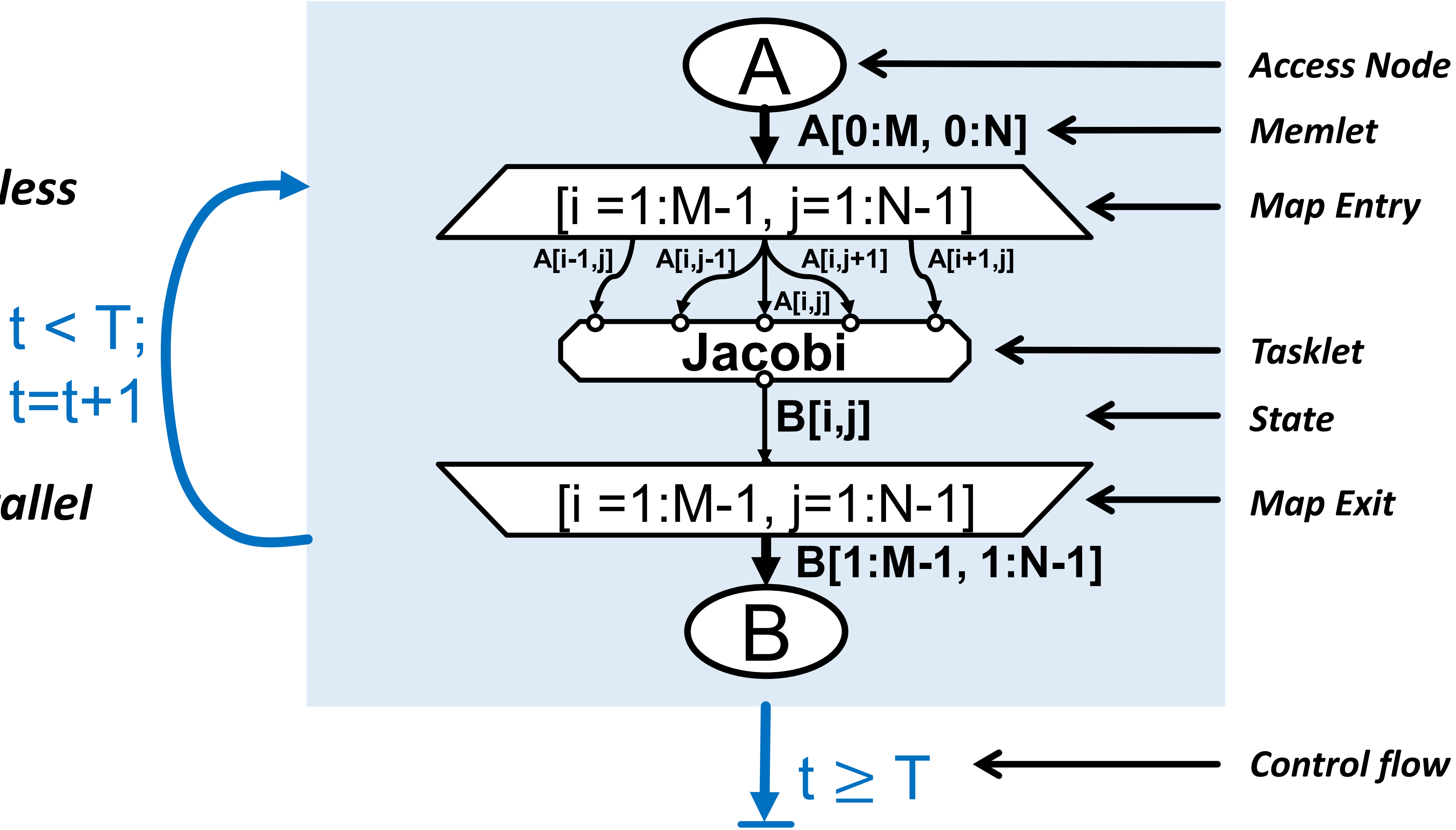
```
mplex<double> * __restrict__ A,
 complex<double> * __restrict__ B,
x<double> * __restrict__ C,
 Nmat, int Nbatch, int Csize) {

    int i = threadIdx.x*ILP;
    int j = threadIdx.y;
    int klen = (Norb * Nmat / TOTALK);
    int kstart = blockIdx.y;

    A += t * Norb * Norb + i + kstart*klen*Norb;
    B += j + kstart*klen*Norb;

    thrust::complex<double> res[ILP] = {0};

    for(int k = 0; k < klen; ++k) {
        auto b = B[0];
        #pragma unroll
        for(int l = 0; l < ILP; ++l)
            res[l] += A[l] * b;
        A += Norb;
        B += Norb;
    }

    C += kstart * Csize + t * Norb * Norb + j * Norb + i;
    for(int l = 0; l < ILP; ++l)
        C[l] = res[l];
}
```

Code Generation

Graph Transformations (API, Interactive)

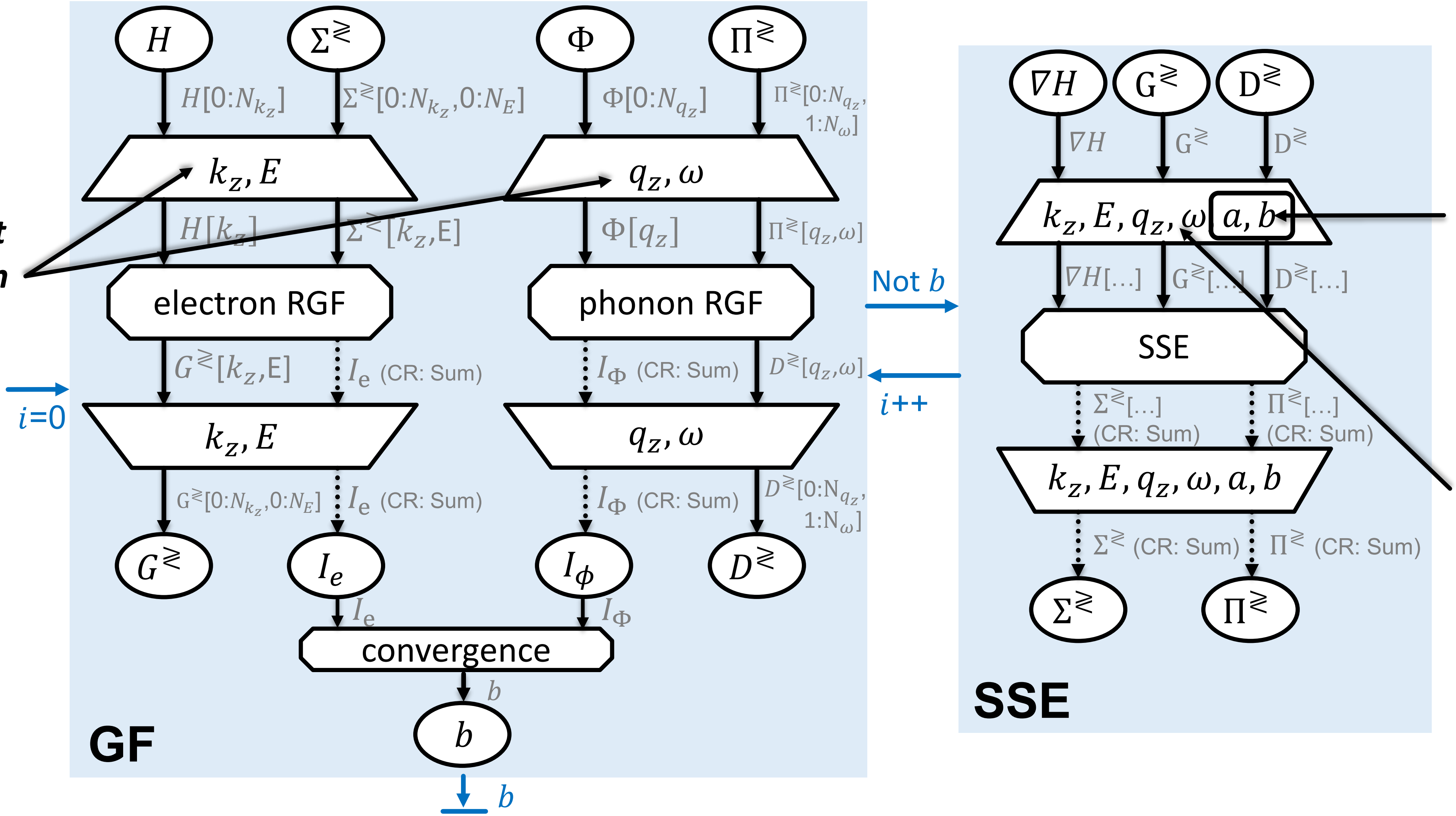
Basic Elements of the Data-Centric IR

Data
Fine-grained stateless computations
Dataflow
Control-flow
Abstraction of independently parallel computations



Ben-Nun et al. “Stateful Dataflow Multigraphs: A Data-Centric Model for Performance Portability on Heterogeneous Architectures”, SC19.

Data-Centric Representation of OMEN

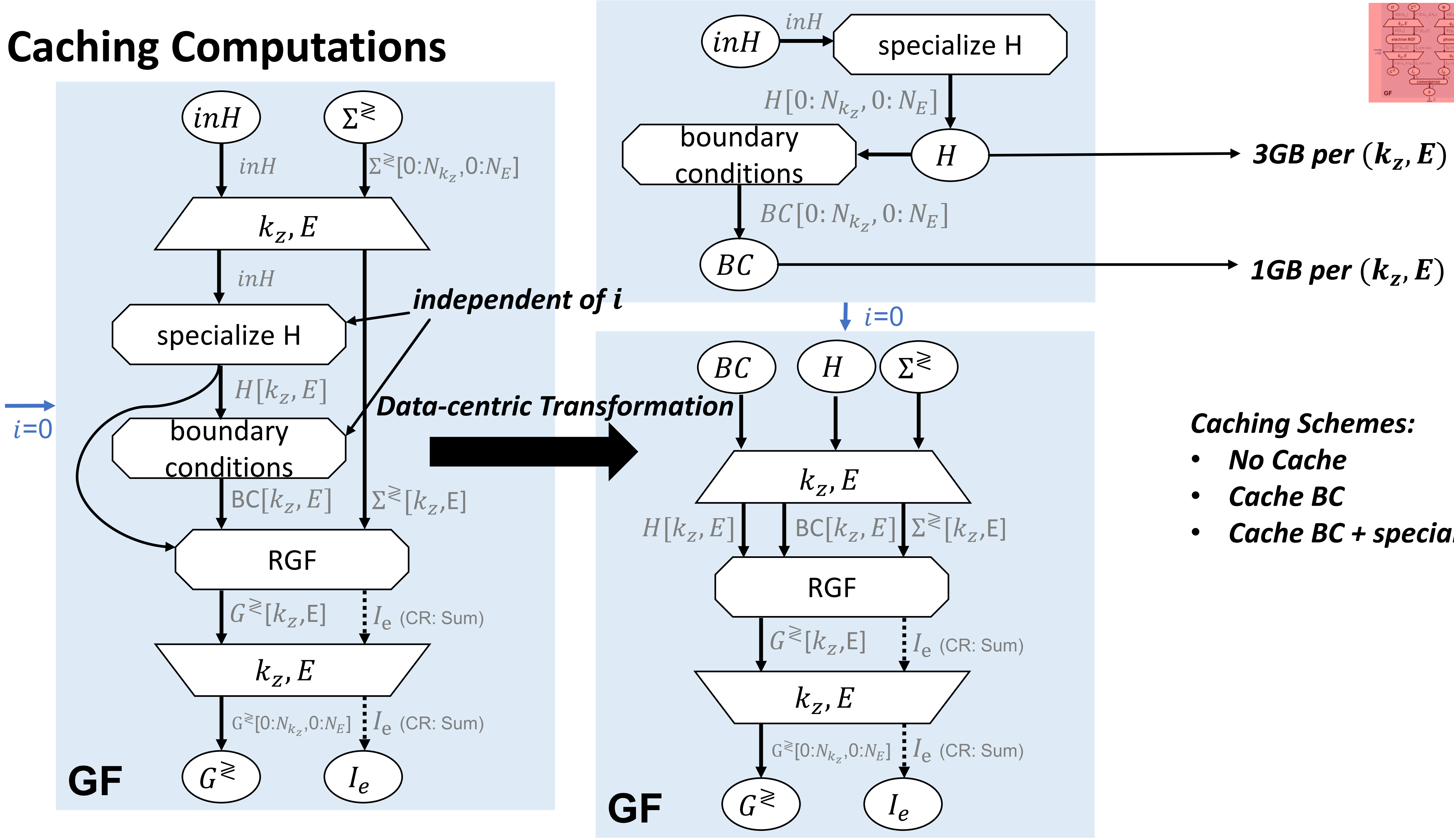


independent computation per slice

atom-neighbor dimensions

greater parallelization potential to exploit …

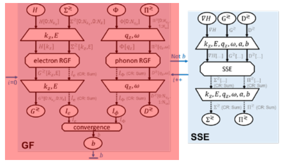
Caching Computations

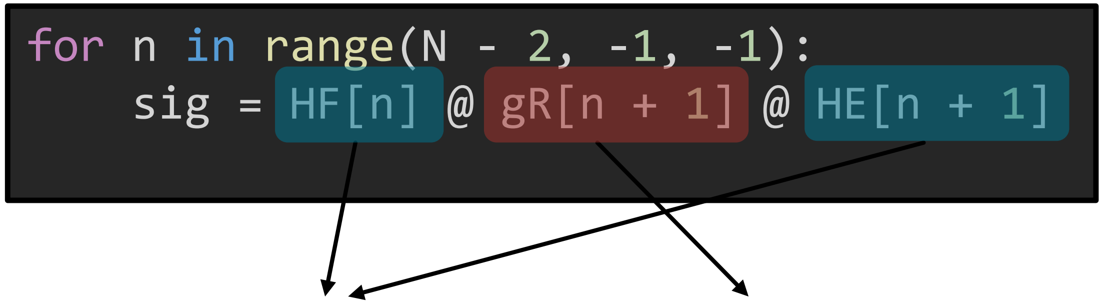


Caching Schemes:

- ***No Cache***
- ***Cache BC***
- ***Cache BC + specialize H***

Optimization of Sparse Operations





```
for n in range(N - 2, -1, -1):
    sig = HF[n] @ gR[n + 1] @ HE[n + 1]
```

Sparse (either CSR or CSC) *Dense*

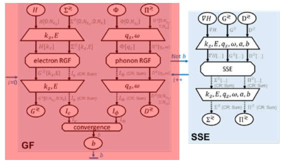
Method\Operation	NN	NT	TN	TT
cublasZgemm	58.382 ms	58.144 ms	58.666 ms	58.315 ms
cusparseZcsrmm2	8.202 ms	6.140 ms	52.722 ms	–
cusparseZgemmi	15.198 ms	–	–	–

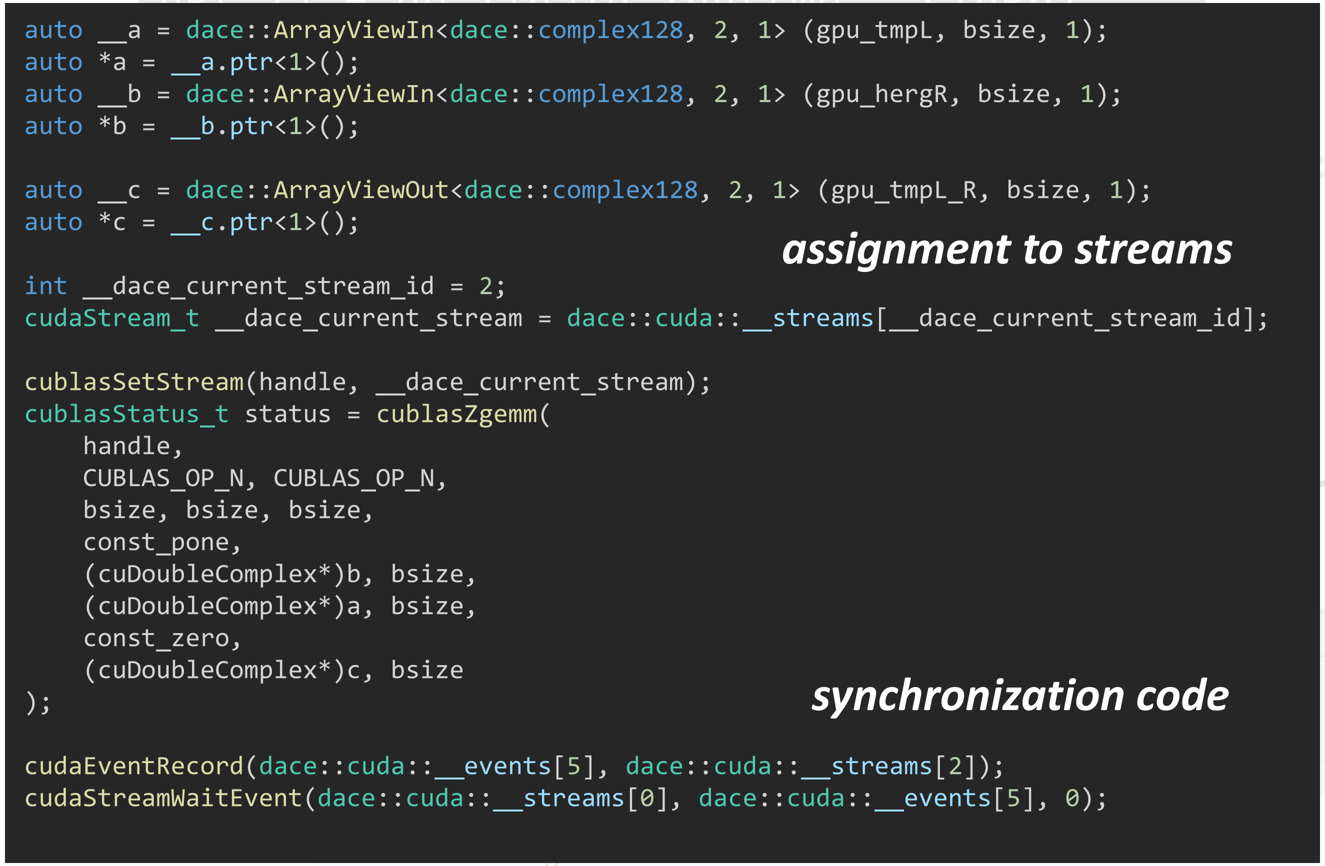
Approach	Time
gemm/gemm	116.881 ms
gemmi(csrmm2(TN, HE, gR), HF)	67.924 ms
csrmm2(NT, HE, csrmm2(NT, HF, gR))	11.994 ms

← *HF and HE are CSR*

← *HF is CSR, HE is CSC*

Extracting Parallelism

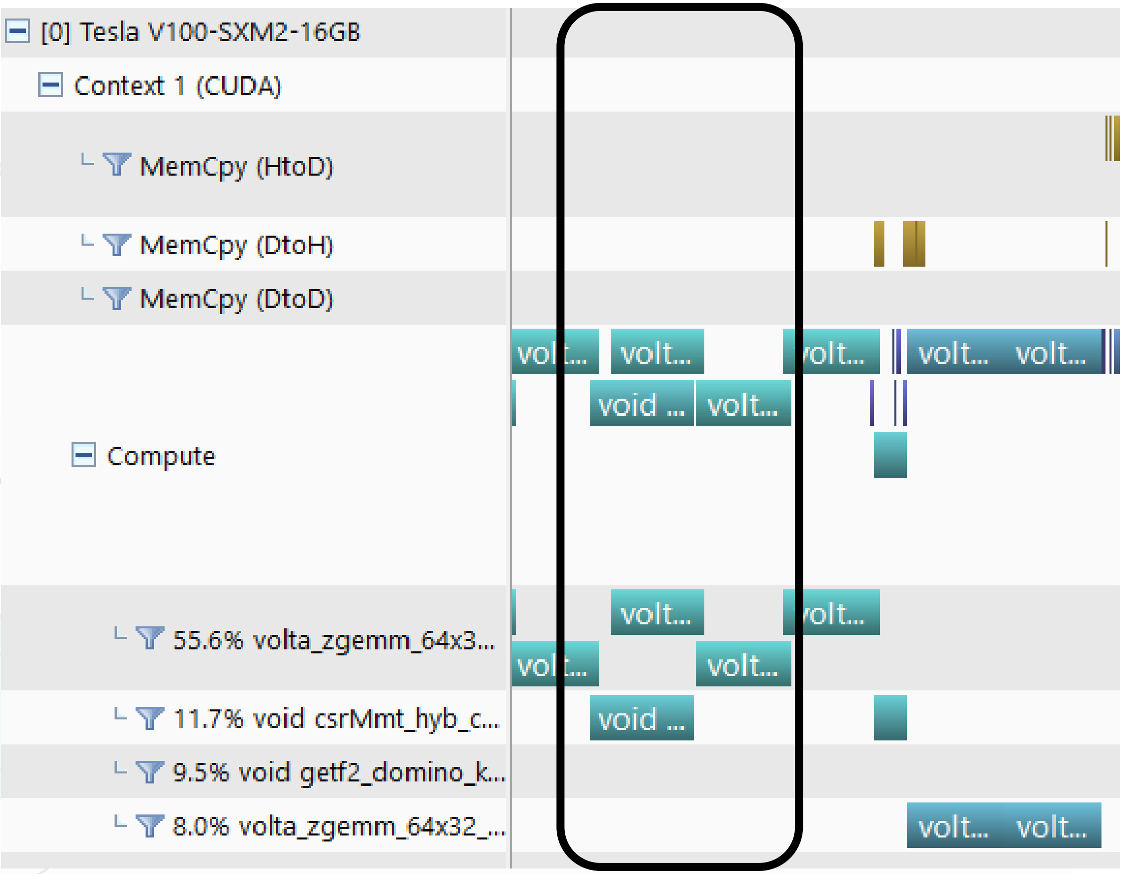




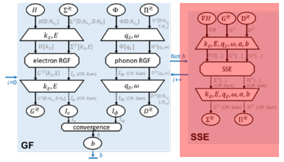
```
auto __a = dace::ArrayViewIn<dace::complex128, 2, 1> (gpu_tmpL, bsize, 1);
auto *a = __a.ptr<1>();
auto __b = dace::ArrayViewIn<dace::complex128, 2, 1> (gpu_hergR, bsize, 1);
auto *b = __b.ptr<1>();

auto __c = dace::ArrayViewOut<dace::complex128, 2, 1> (gpu_tmpL_R, bsize, 1);
auto *c = __c.ptr<1>();

int __dace_current_stream_id = 2;
cudaStream_t __dace_current_stream = dace::cuda::__streams[__dace_current_stream_id];

cublasSetStream(handle, __dace_current_stream);
cublasStatus_t status = cublasZgemm(
    handle,
    CUBLAS_OP_N, CUBLAS_OP_N,
    bsize, bsize, bsize,
    const_pone,
    (cuDoubleComplex*)b, bsize,
    (cuDoubleComplex*)a, bsize,
    const_zero,
    (cuDoubleComplex*)c, bsize
);

cudaEventRecord(dace::cuda::__events[5], dace::cuda::__streams[2]);
cudaStreamWaitEvent(dace::cuda::__streams[0], dace::cuda::__events[5], 0);
```

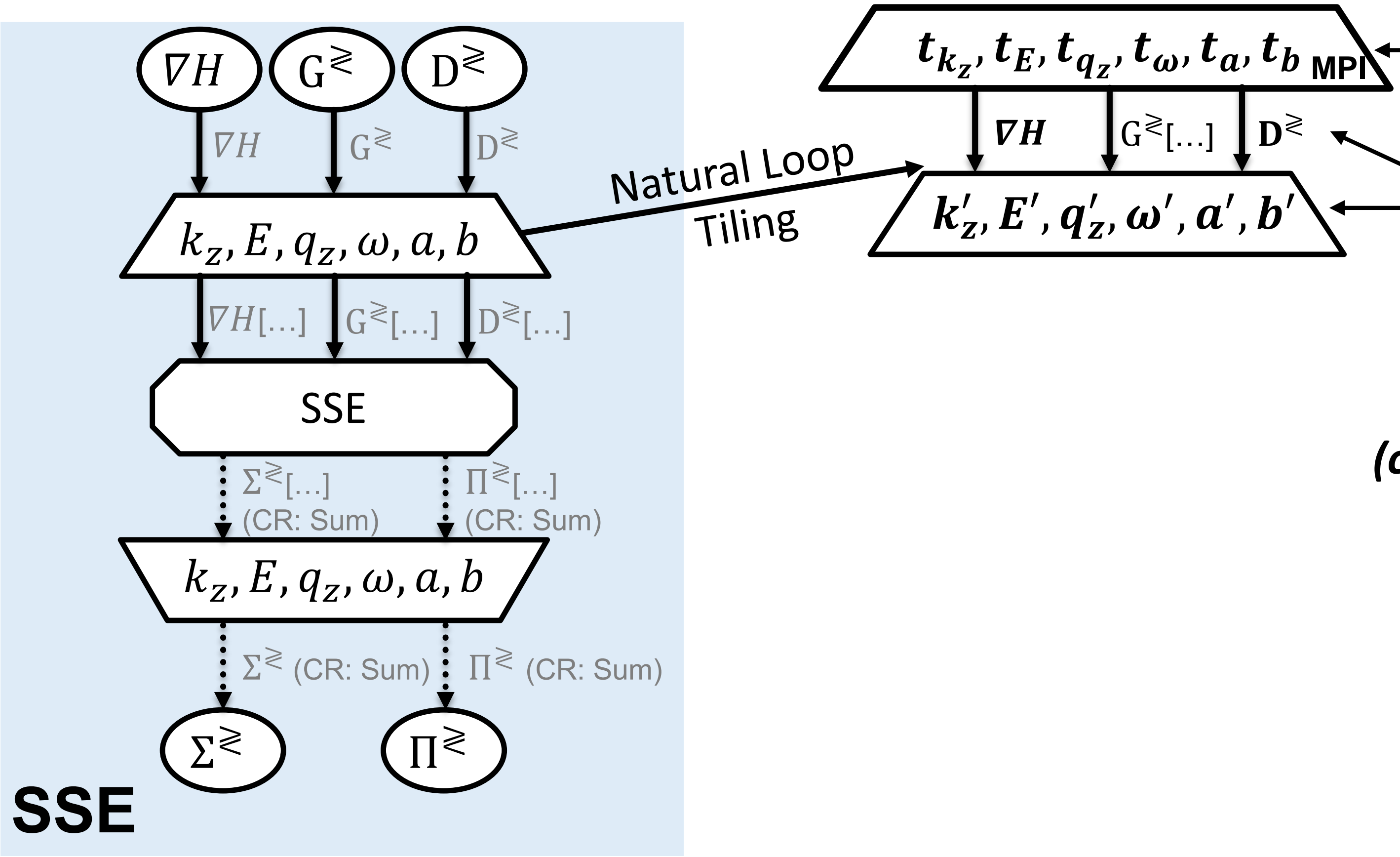
assignment to streams

synchronization code



Optimizing Coarse-Grained Data Movement



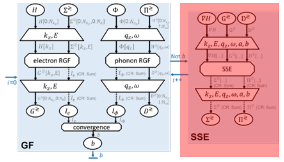

∇H
$G^{\gtrless}$
$D^{\gtrless}$
∇H
$G^{\gtrless}$
$D^{\gtrless}$
$k_z, E, q_z, \omega, a, b$
$\nabla H[\ldots]$
$G^{\gtrless}[\ldots]$
$D^{\gtrless}[\ldots]$
SSE
$\Sigma^{\gtrless}[\ldots]$ (CR: Sum)
$\Pi^{\gtrless}[\ldots]$ (CR: Sum)
$k_z, E, q_z, \omega, a, b$
$\Sigma^{\gtrless}$ (CR: Sum)
$\Pi^{\gtrless}$ (CR: Sum)
$\Sigma^{\gtrless}$
$\Pi^{\gtrless}$
SSE
Natural Loop Tiling
$t_{k_z}, t_E, t_{q_z}, t_\omega, t_a, t_b$ MPI
∇H
$G^{\gtrless}[\ldots]$
$D^{\gtrless}$
$k'_z, E', q'_z, \omega', a', b'$

Iterates over the tiles (nodes)

Iterates over a single tile (workload of a single node)

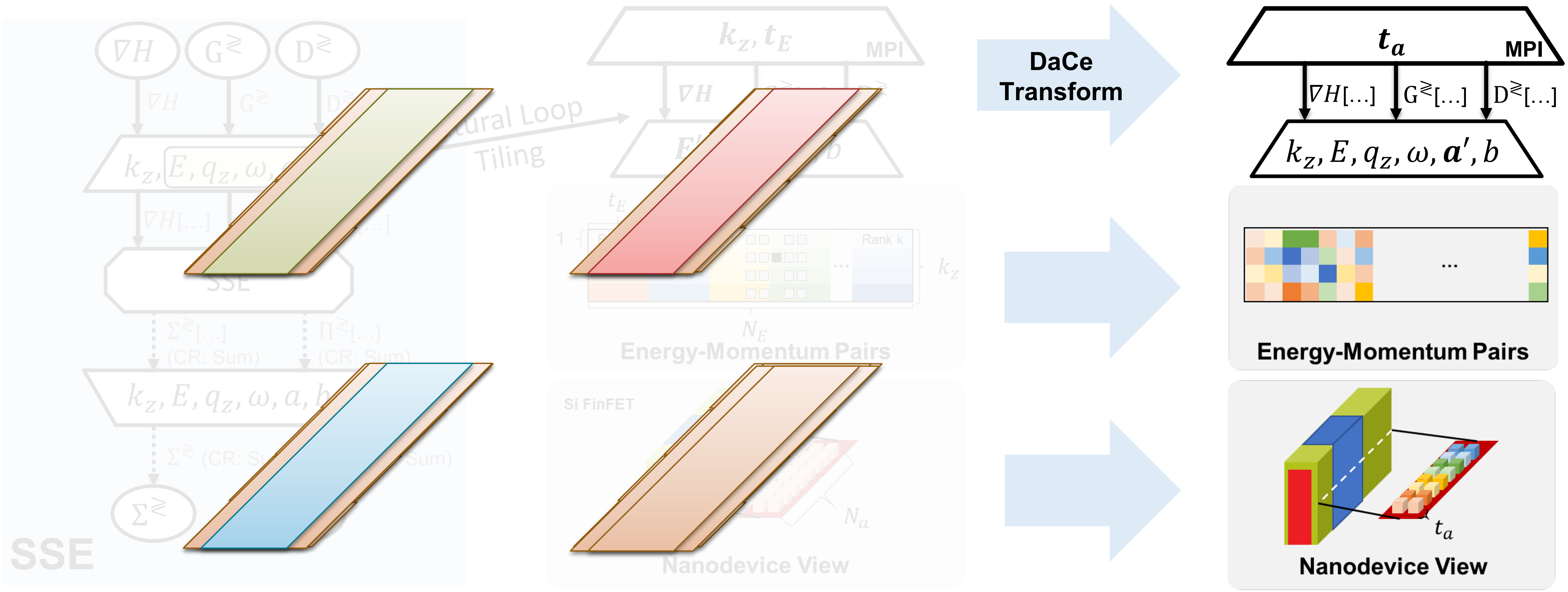
Memlets represent the surface of each tile (communication volume needed for each node)

Optimizing Coarse-Grained Data Movement



OMEN Decomposition

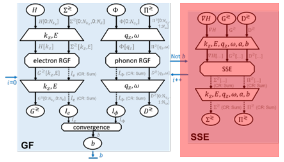
Data-Centric Decomposition

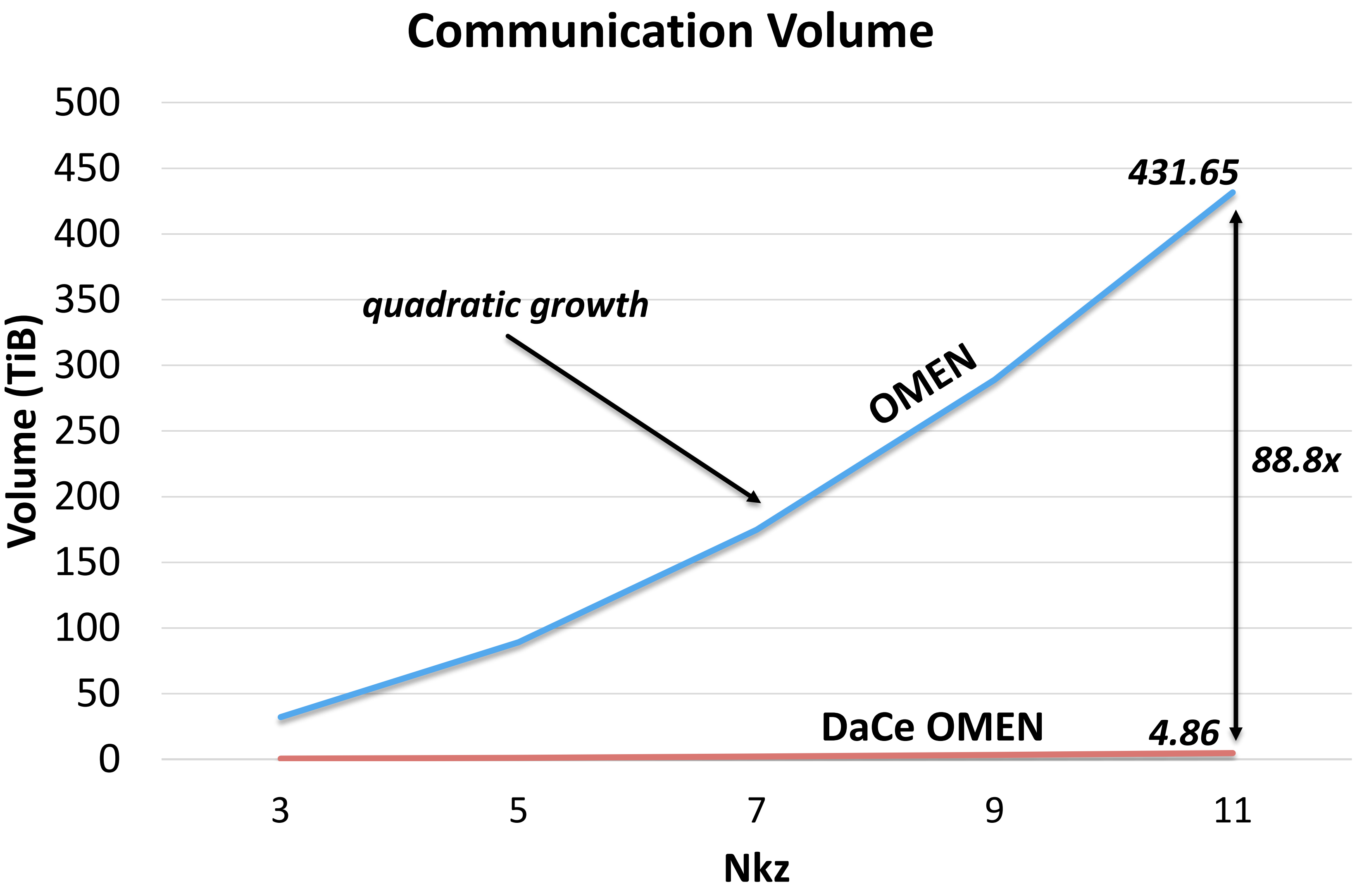


Method: Broadcast, Reduce, P2P
Volume: $\mathcal{O}(N_{k_z} N_E N_{q_z} N_\omega N_a N_{orb}^2)$
MPI Invocations: $9 N_\omega N_{q_z}$

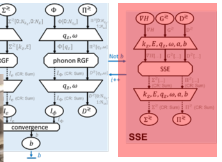
Method: Collective Alltoall
Volume: $\mathcal{O}(N_{k_z}(N_E + N_\omega)(N_a + N_b)\ N_{orb}^2)$
MPI Invocations: 4

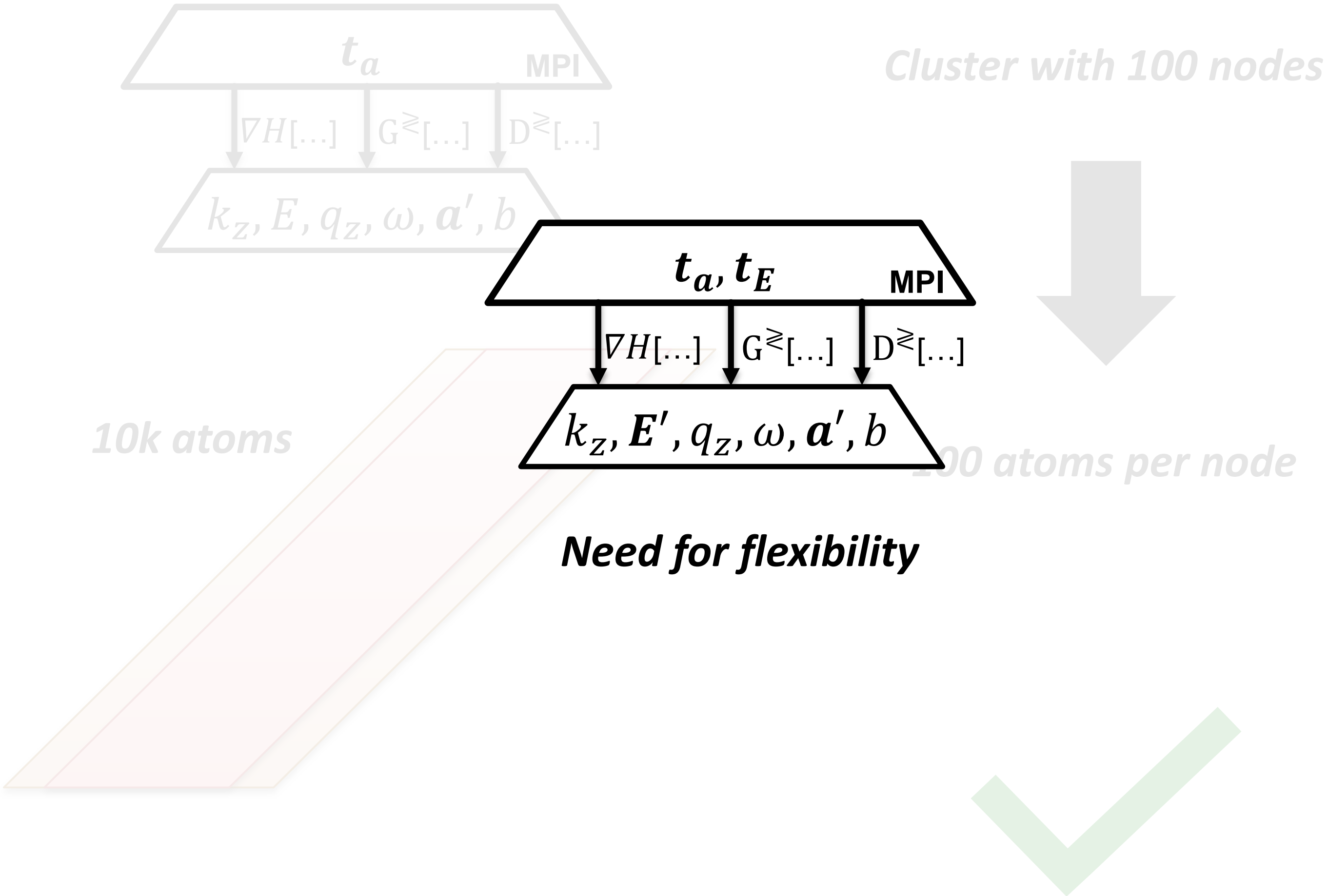
Optimizing Coarse-Grained Data Movement





Optimizing Coarse-Grained Data Movement





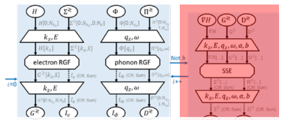
Need for flexibility

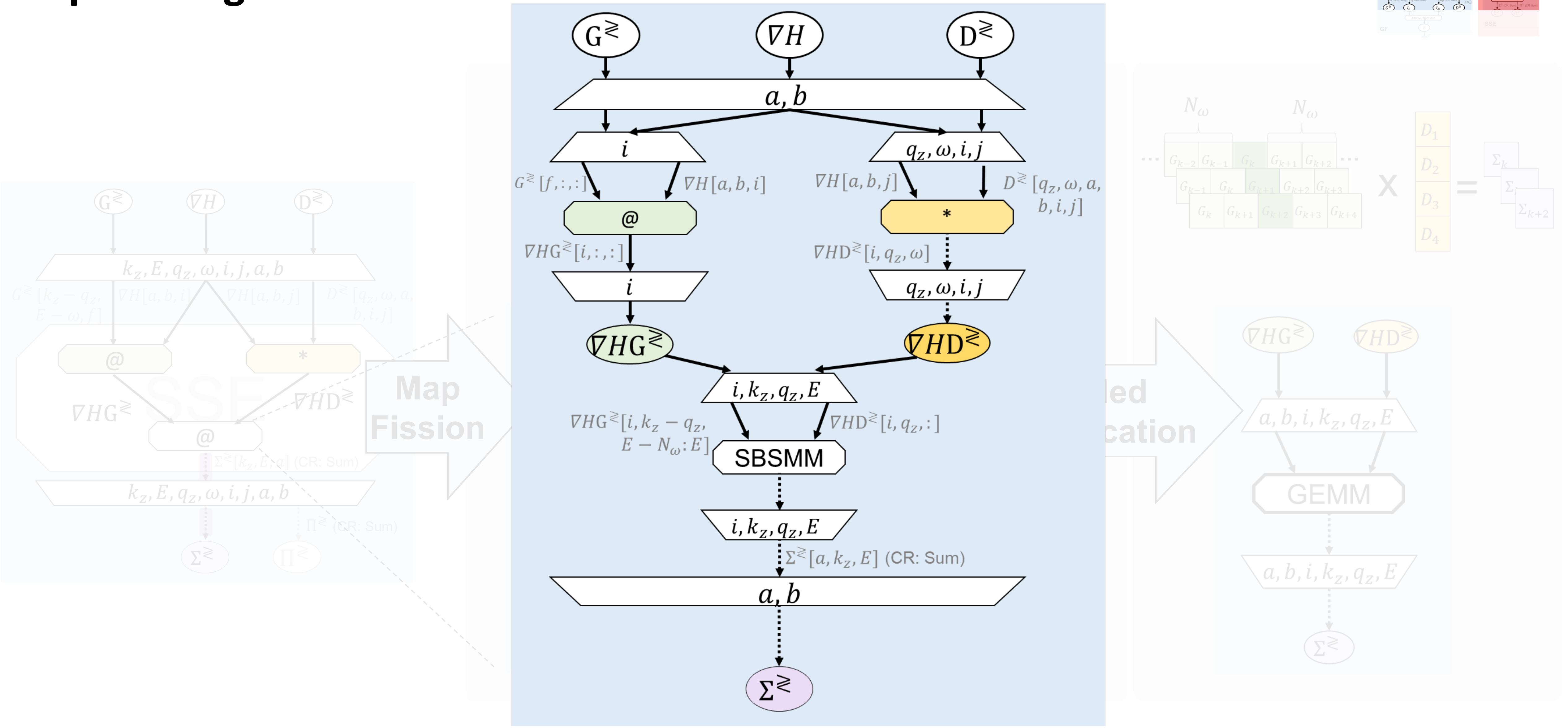
Source: Oak Ridge National Laboratory

***27k* X**

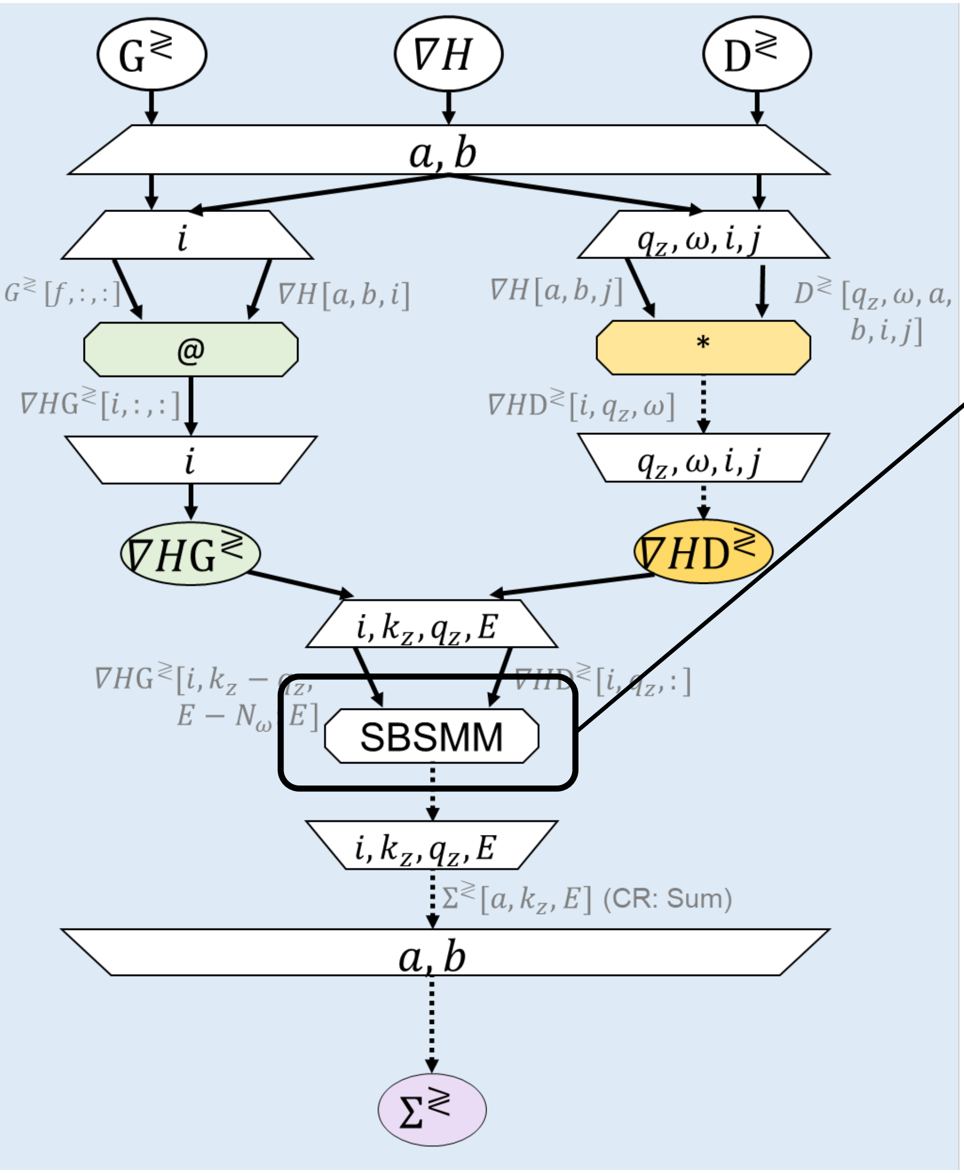
Source: NVIDIA

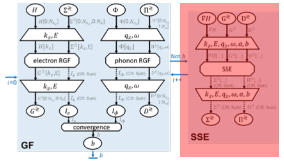
Optimizing Fine-Grained Data Movement





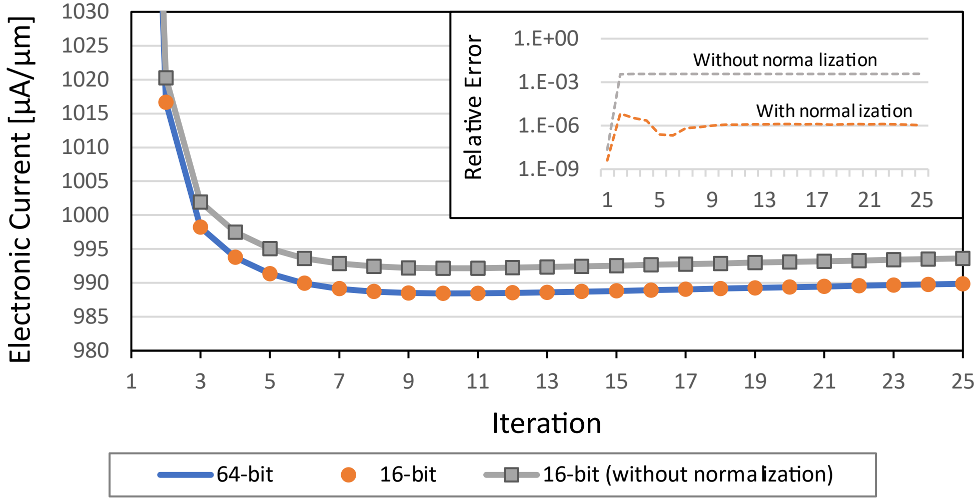
Mixed Precision



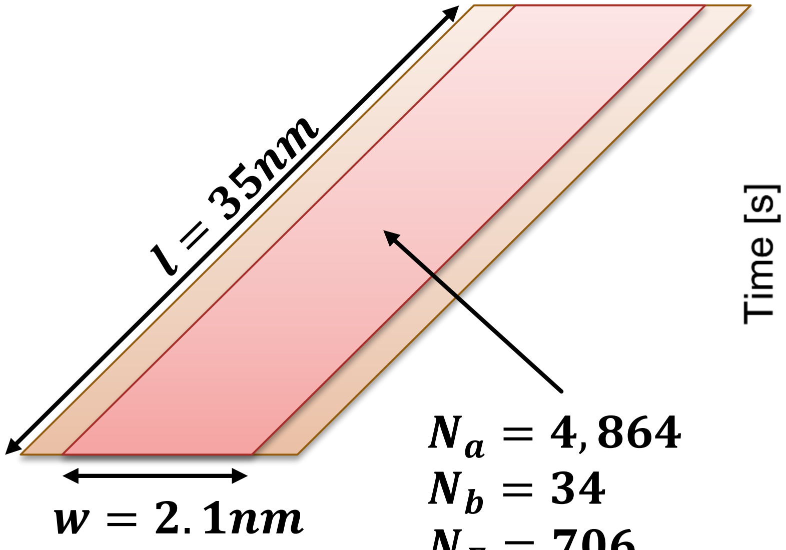


cublasGemmStridedBatchedEx

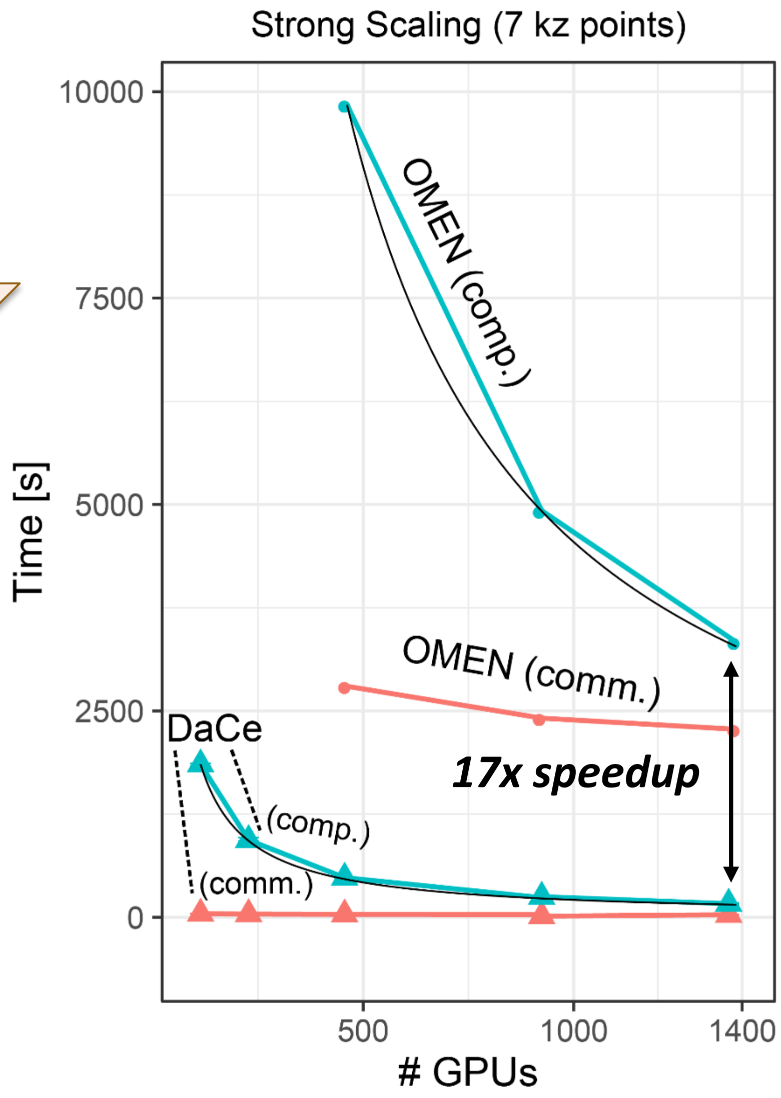
Precision\Implementation	cuBLAS	DaCe
Double	4.62 ms	0.70 ms
Mixed – Tensor Cores	N/A	0.13 ms

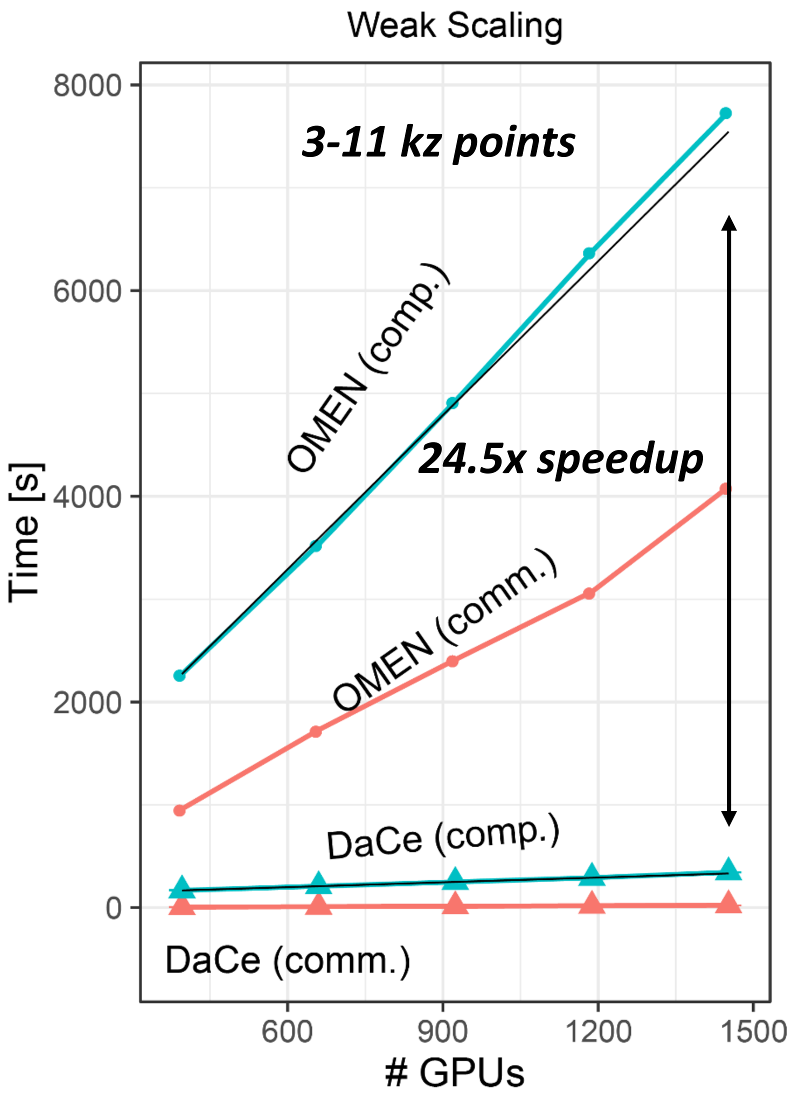


OMEN vs DaCe OMEN: Performance

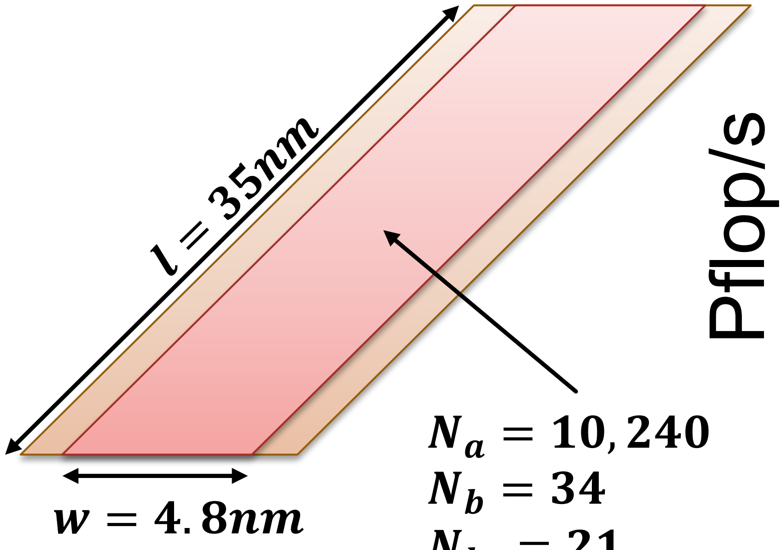


$N_a = 4,864$
$N_b = 34$
$N_E = 706$
$N_\omega = 70$

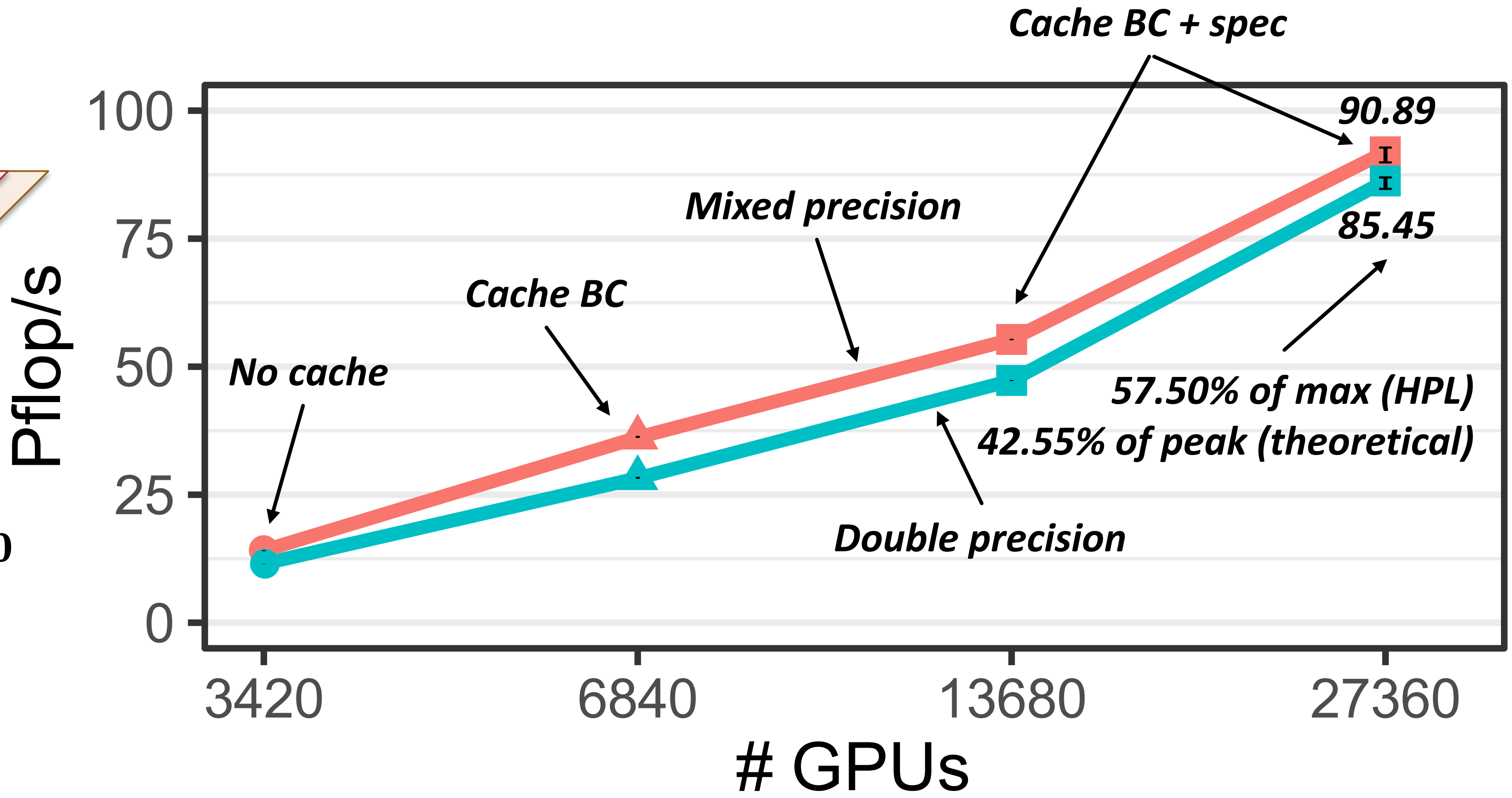




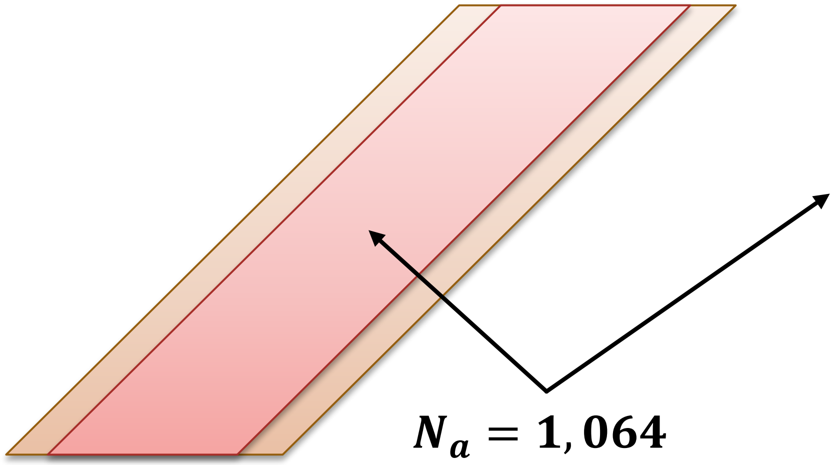
Beyond OMEN



$N_a = 10,240$
$N_b = 34$
$N_{k_z} = 21$
$N_E = 1220$
$N_\omega = 70$



Beyond OMEN



$N_a = 1,064$
$N_b = 34$
$N_{k_z} = 21$
$N_E = 1220$
$N_\omega = 70$

6,840 GPUs

Variant	Atom No.	Time [s]	Time/Atom [s]	Speedup
OMEN	1,064	4,695.70	4.413	
DaCe OMEN	10,240	333.36	0.033	140.9x

Conclusions

Heat Dissipation Issue

QT Simulation

OMEN Application

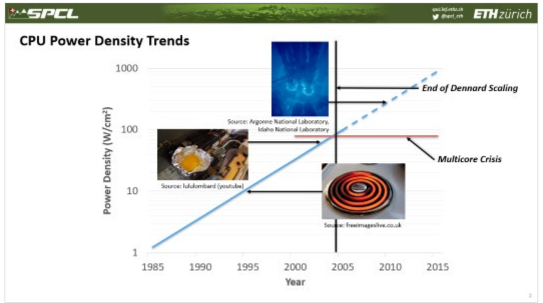
Domain Scientists' View

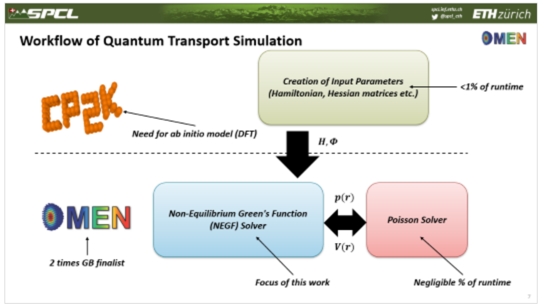
Data-Centric View

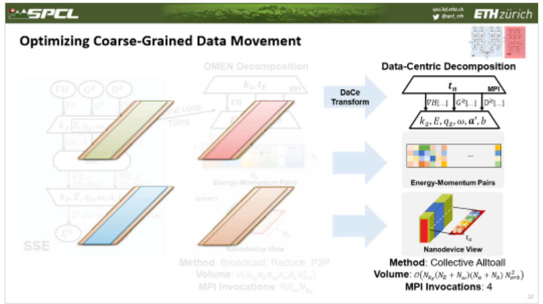
Optimizing Coarse-Grained and Fine-Grained Dataflow

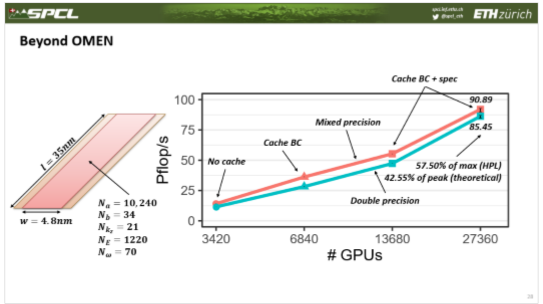
Extracting Parallelism

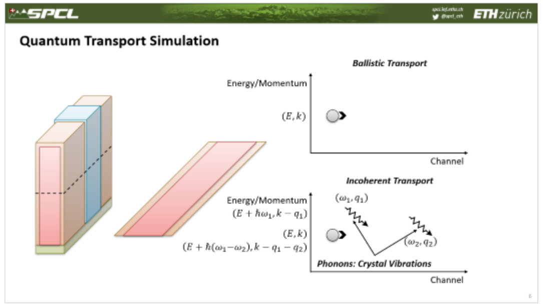
Performance

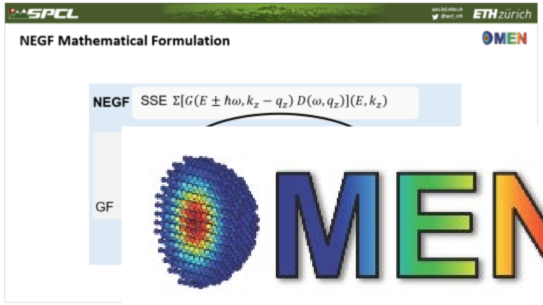


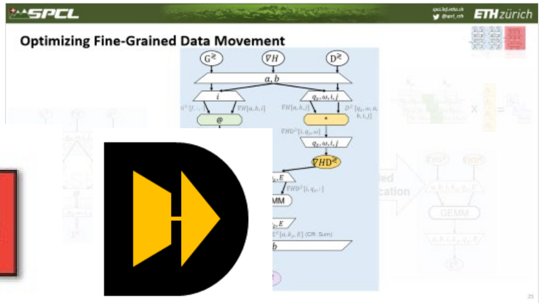

Workflow of Quantum Transport Simulation
OMEN
CP2K
Need for ab initio model (DFT)
Creation of Input Parameters (Hamiltonian, Hessian matrices etc.)
<1% of runtime
H, Φ
Non-Equilibrium Green's Function (NEGF) Solver
Poisson Solver
p(r)
V(r)
2 times GB finalist
Focus of this work
Negligible % of runtime


Optimizing Coarse-Grained Data Movement
OMEN Decomposition
DaCe Transform
Data-Centric Decomposition
Energy-Momentum Pairs
Nanodevice View
SSE
Method: Collective Alltoall
MPI Invocations: 4



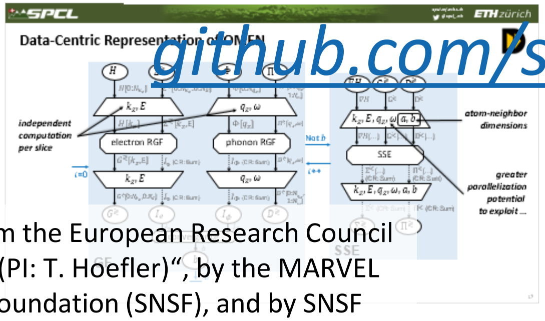

Quantum Transport Simulation
Ballistic Transport
Energy/Momentum
(E, k)
Channel
Incoherent Transport
(ω_1, q_1)
$(E + \hbar\omega_1, k - q_1)$
(ω_2, q_2)
$(E + \hbar(\omega_1 - \omega_2), k - q_1 - q_2)$
Phonons: Crystal Vibrations
Channel


NEGF Mathematical Formulation
OMEN
NEGF
SSE $\Sigma[G(E \pm \hbar\omega, k_z - q_z)\, D(\omega, q_z)](E, k_z)$
GF



MEN

github.com/spcl/dace


Data-Centric Representation of OMEN
independent computation per slice
electron RGF
phonon RGF
SSE
atom-neighbor dimensions
greater parallelization potential to exploit ...

This project has received funding from the European Research Council (ERC) under grant agreement "DAPP (PI: T. Hoefler)", by the MARVEL NCCR of the Swiss National Science Foundation (SNSF), and by SNSF grant 175479 (ABIME).