ETH zürich

spcl.inf.ethz.ch
@spcl_eth

D INFK

Alexandru Calotoiu

Slides courtesy of Tal Ben-Nun, Johannes de Fine Licht, Alexandros-Nikolaos Ziogas, Timo Schneider, Torsten Hoefler, Jan Kleine, Philipp Schaad and others at SPCL

Stateful Dataflow Multigraphs: A Data-Centric Model for Performance Portability on Heterogeneous Architectures

SPCL

This project has received funding from the European Research Council (ERC) under grant agreement "DAPP (PI: T. Hoefler)".

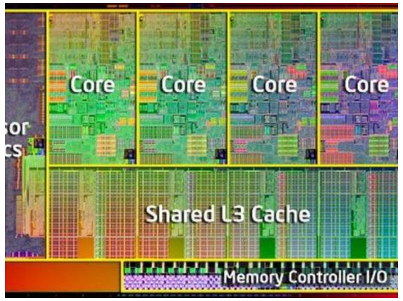
erc

Motivation

Cray-1 Vector Processor

CPU

Multi-Core



GPU Computing
Heterogeneous Systems

NVIDIA CUDA

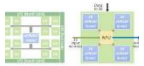
OpenCL

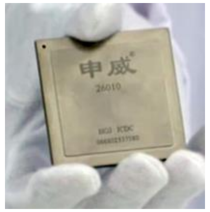
OpenACC
More Science, Less Programming

AMD RADEON GRAPHICS

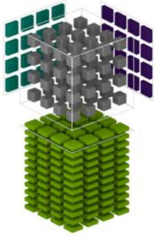
NVIDIA GEFORCE GTX

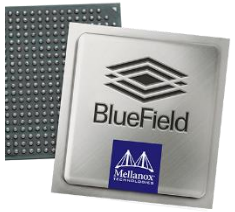
Specialization





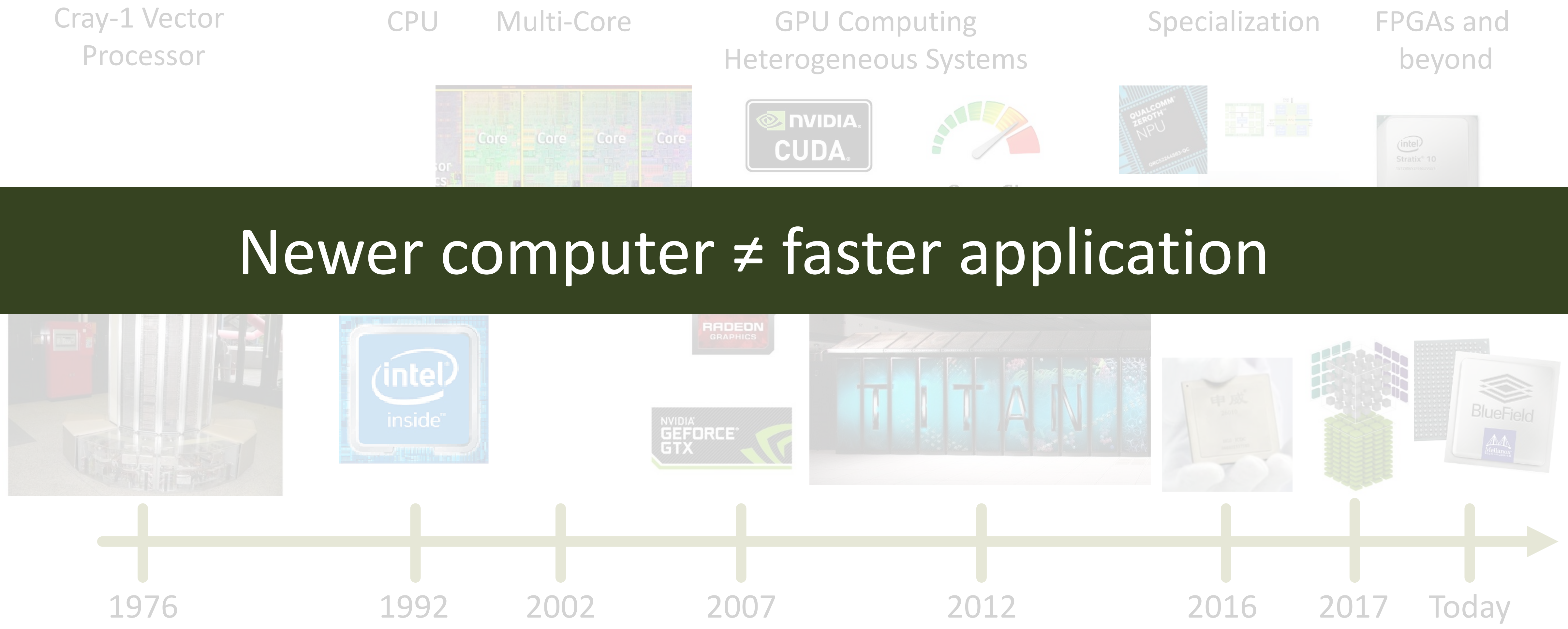
FPGAs and beyond





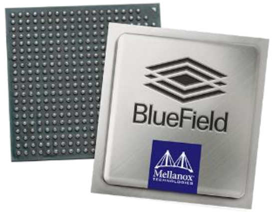
1976 | 1992 | 2002 | 2007 | 2012 | 2016 | 2017 | Today

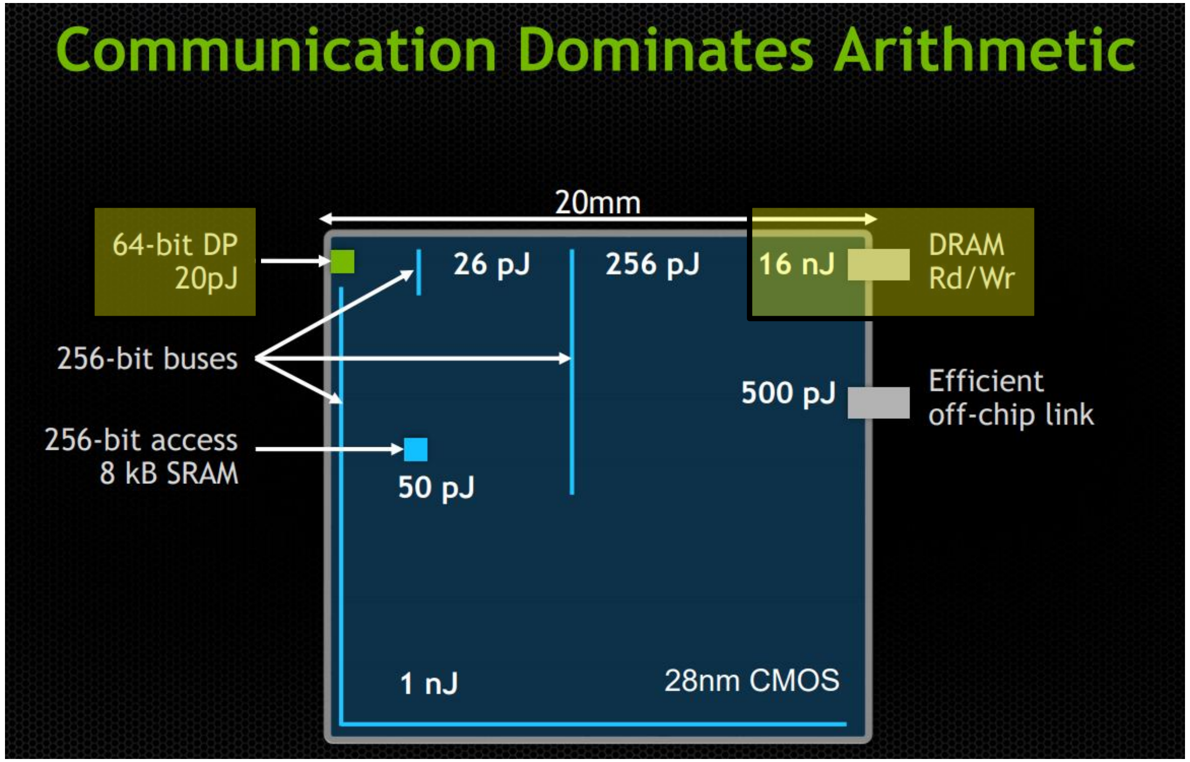
Motivation

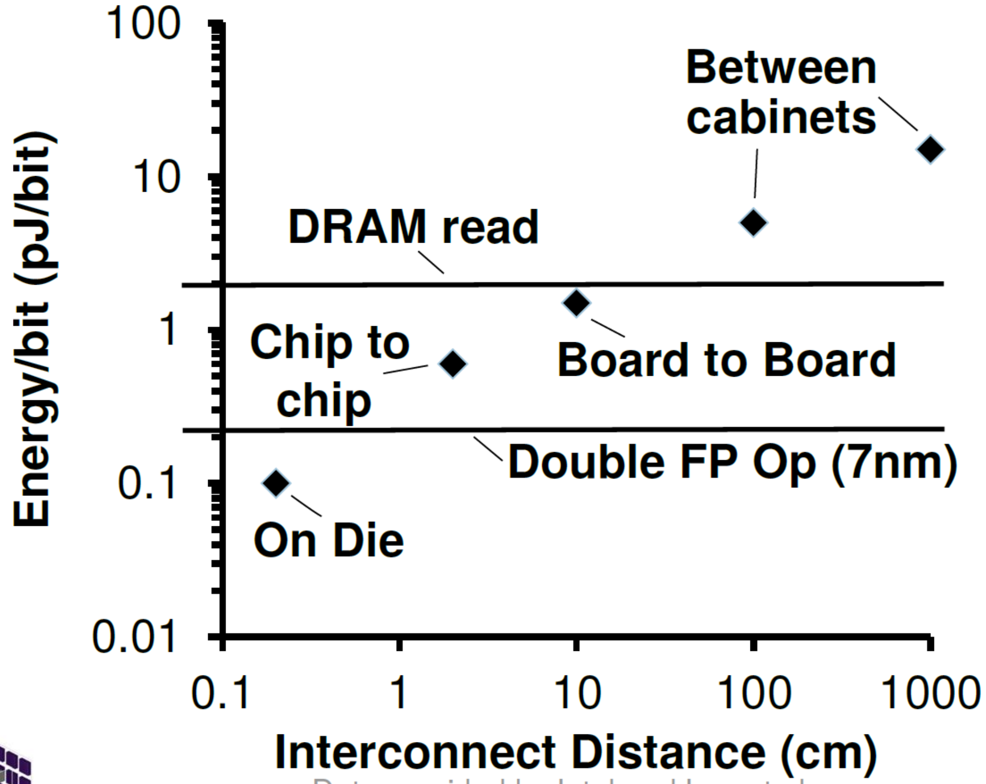


Newer computer ≠ faster application

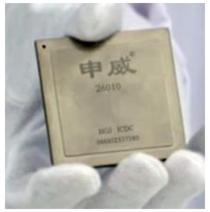
Motivation



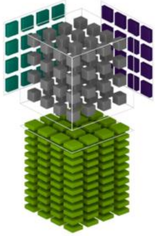




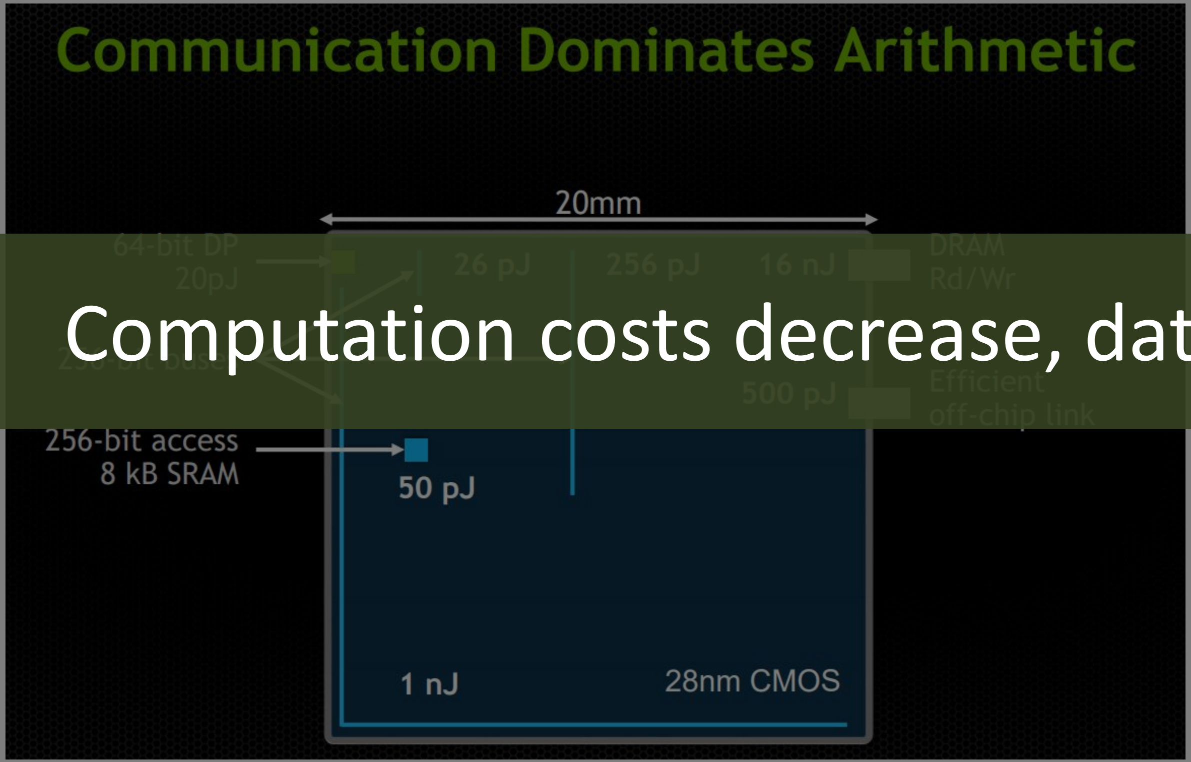
Data provided by Intel and Lee et al.



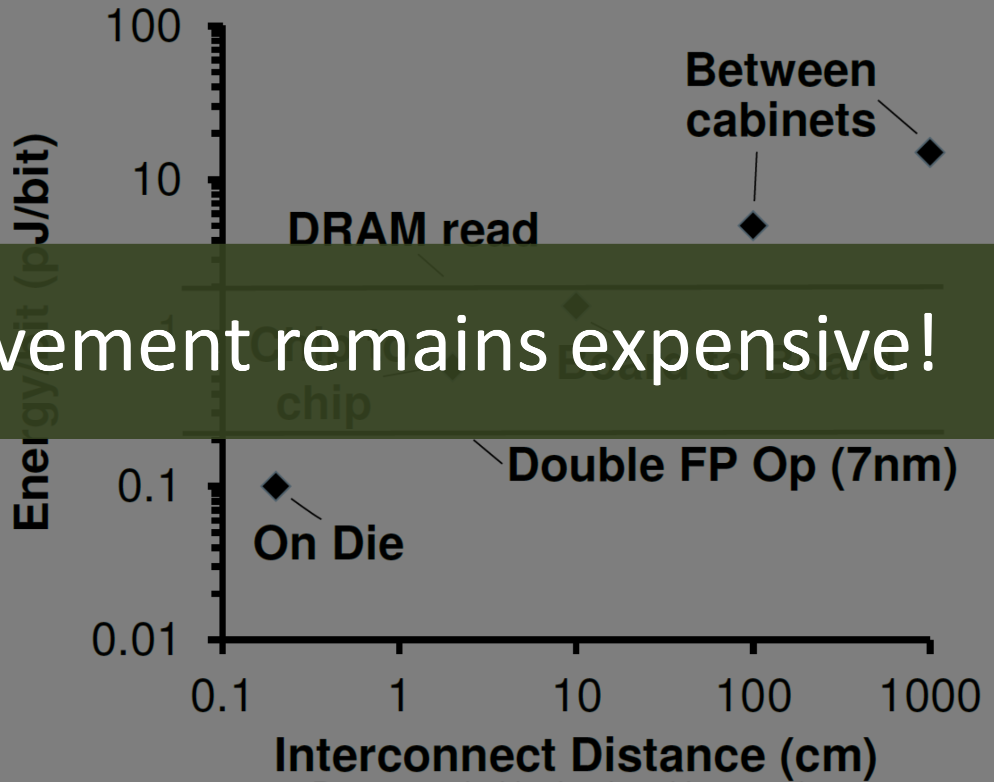
Slide courtesy of NVIDIA



Motivation

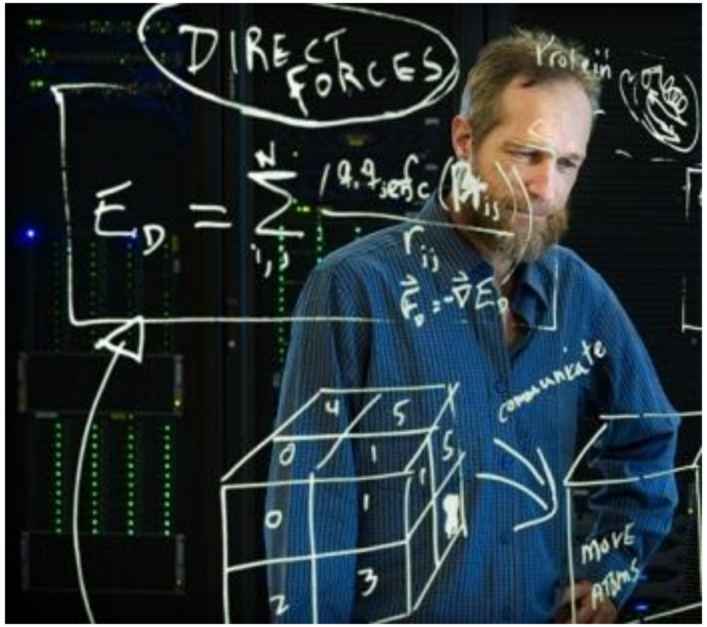


Slide courtesy of NVIDIA



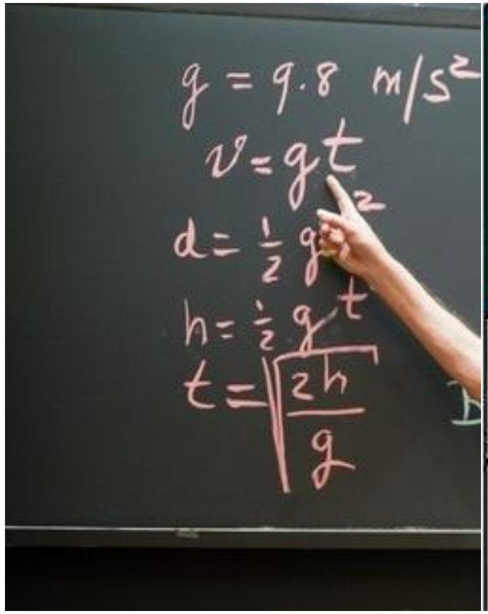
Data provided by Intel and Lee et al.

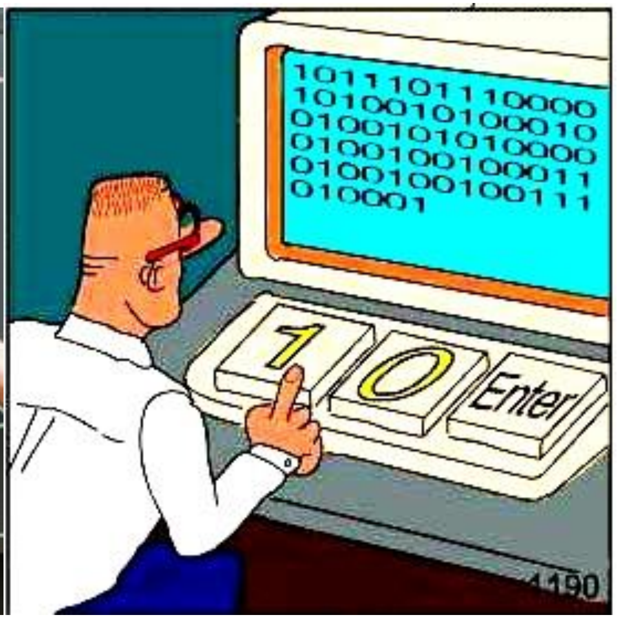
Computation costs decrease, data movement remains expensive!



Source: US DoE

Computational Scientist





Domain Scientist

Performance Engineer

Optimization Techniques

- **Multi-core CPU**
  - Tiling for complex cache hierarchies
  - Register optimizations
  - Vectorization

- **Many-core GPU**
  - Warp divergence minimization, register tiling
  - Task fusion

- **FPGA**
  - Maximize resource utilization (logic units, DSPs)
  - Streaming optimizations, pipelining
  - Explicit buffering (FIFO) and wiring

High-performance optimization = data movement reduction

DaCe Overview

Domain Scientist

Problem Formulation

$$\frac{\partial u}{\partial t} - \alpha \nabla^2 u = 0$$

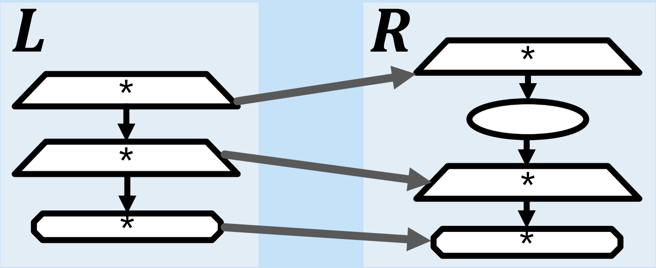
Python	DSLs
TensorFlow	MATLAB

...

Scientific Frontend

Performance Engineer

Data-Centric Intermediate Representation (SDFG)



Graph Transformations

Transformed Dataflow

Performance Results

System

Hardware Information

Compiler

Runtime

CPU Binary

GPU Binary

FPGA Modules

Data-centric Parallel Programming for Python

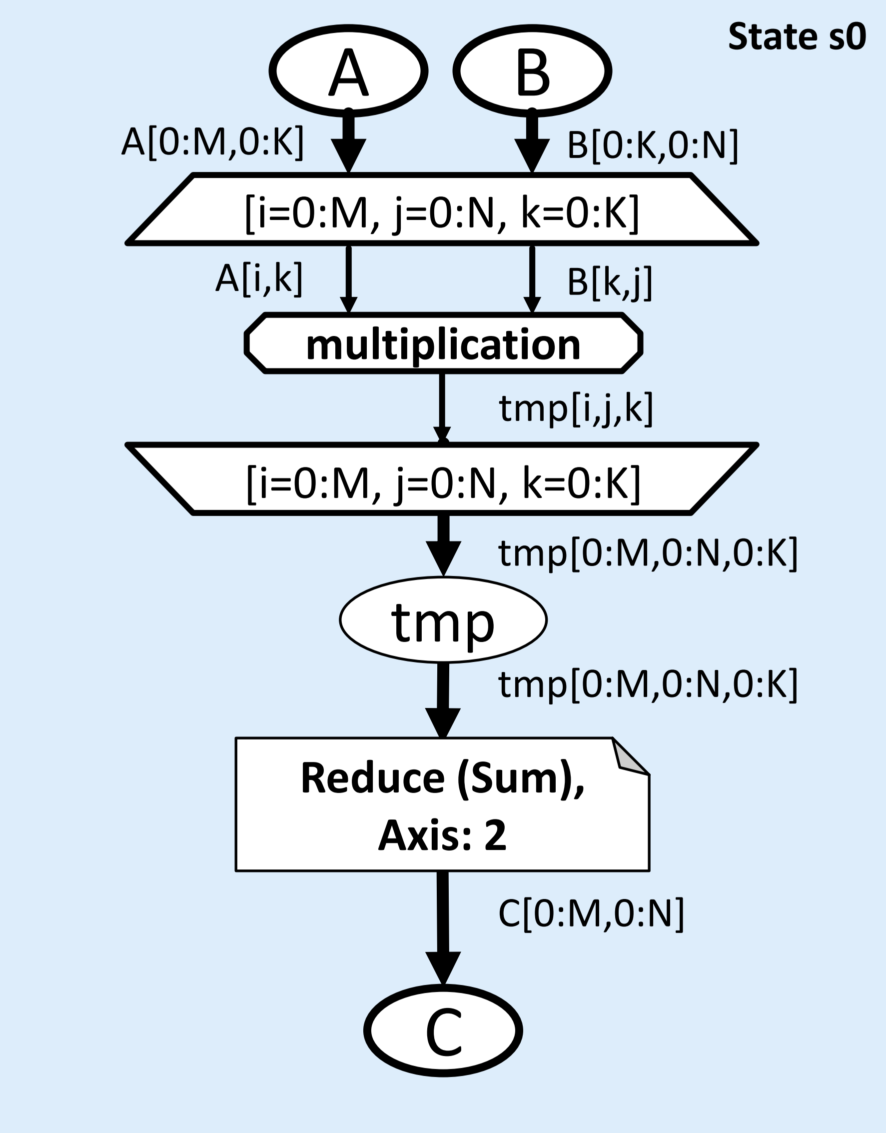
- **Programs are integrated within existing codes**
  - In Python, integrated functions in existing code
  - In MATLAB, separate .m files
  - In TensorFlow, takes existing graph
- **In Python: Implicit and Explicit Dataflow**
  - **Implicit:** numpy syntax
  - **Explicit:** Enforce memory access decoupling from computation
- **Output compatible with existing programs**
  - C-compatible SO/DLL file with autogenerated include file

```
@dace.program
def program_numpy(A, B):
  B[:] = np.transpose(A)
```

```
@dace.program
def program_explicit(A, B):

  @dace.map
  def transpose(i: _[0:N],
                j: _[0:M]):
    a << A[i,j]
    b >> B[j,i]

    b = a
```

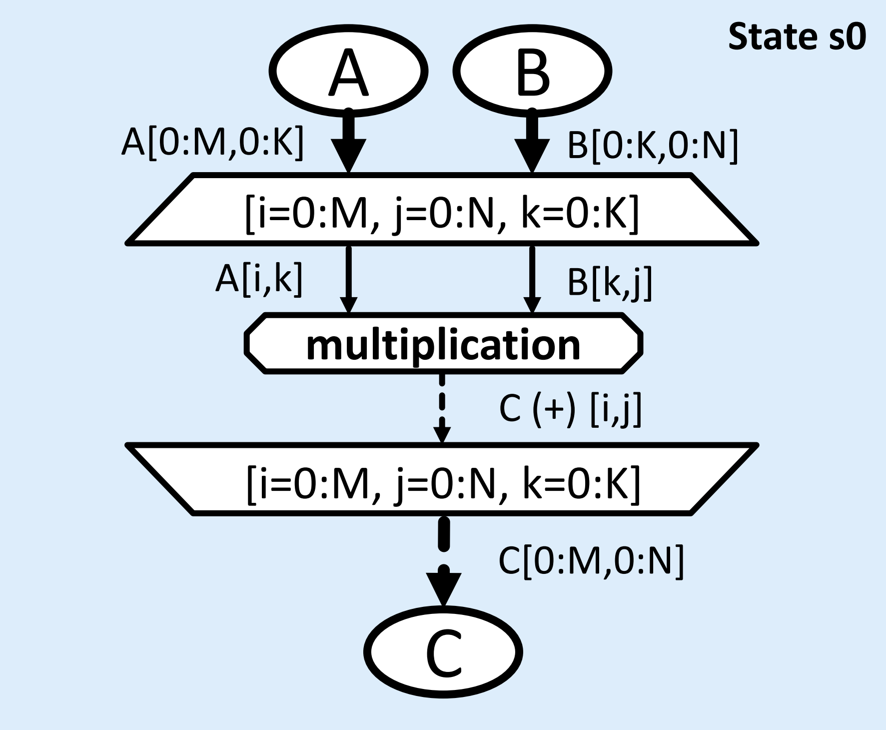
Matrix Multiplication SDFG

```
@dace.program
def gemm(A: dace.float64[M, K], B: dace.float64[K, N],
         C: dace.float64[M, N]):
    # Transient variable
    tmp = np.ndarray([M, N, K], dtype=A.dtype)

    @dace.map
    def multiplication(i: _[0:M], j: _[0:N], k: _[0:K]):
        in_A << A[i,k]
        in_B << B[k,j]
        out  >> tmp[i,j,k]

        out = in_A * in_B

    dace.reduce(lambda a, b: a + b, tmp, C, axis=2)
```

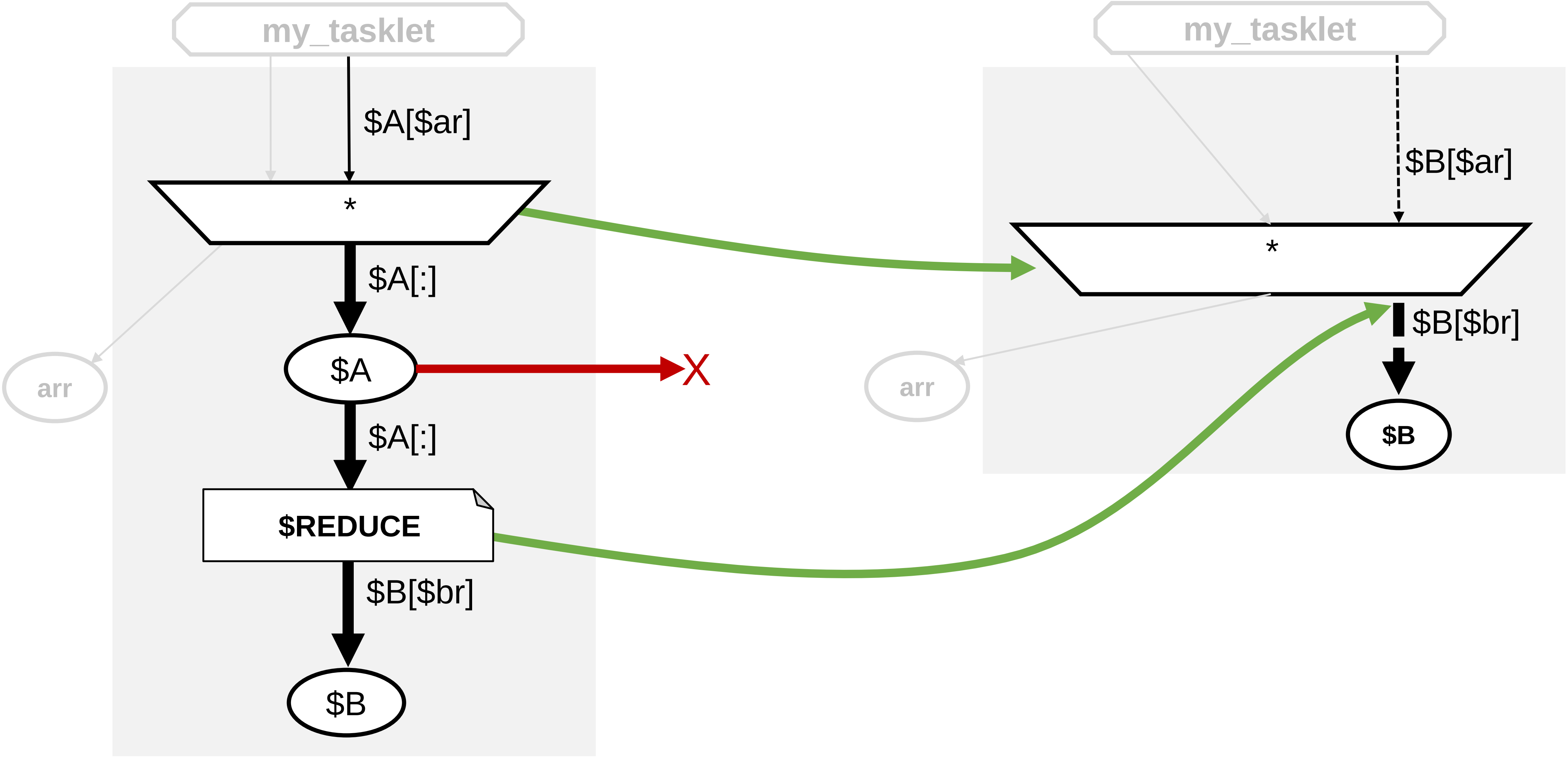


Matrix Multiplication SDFG

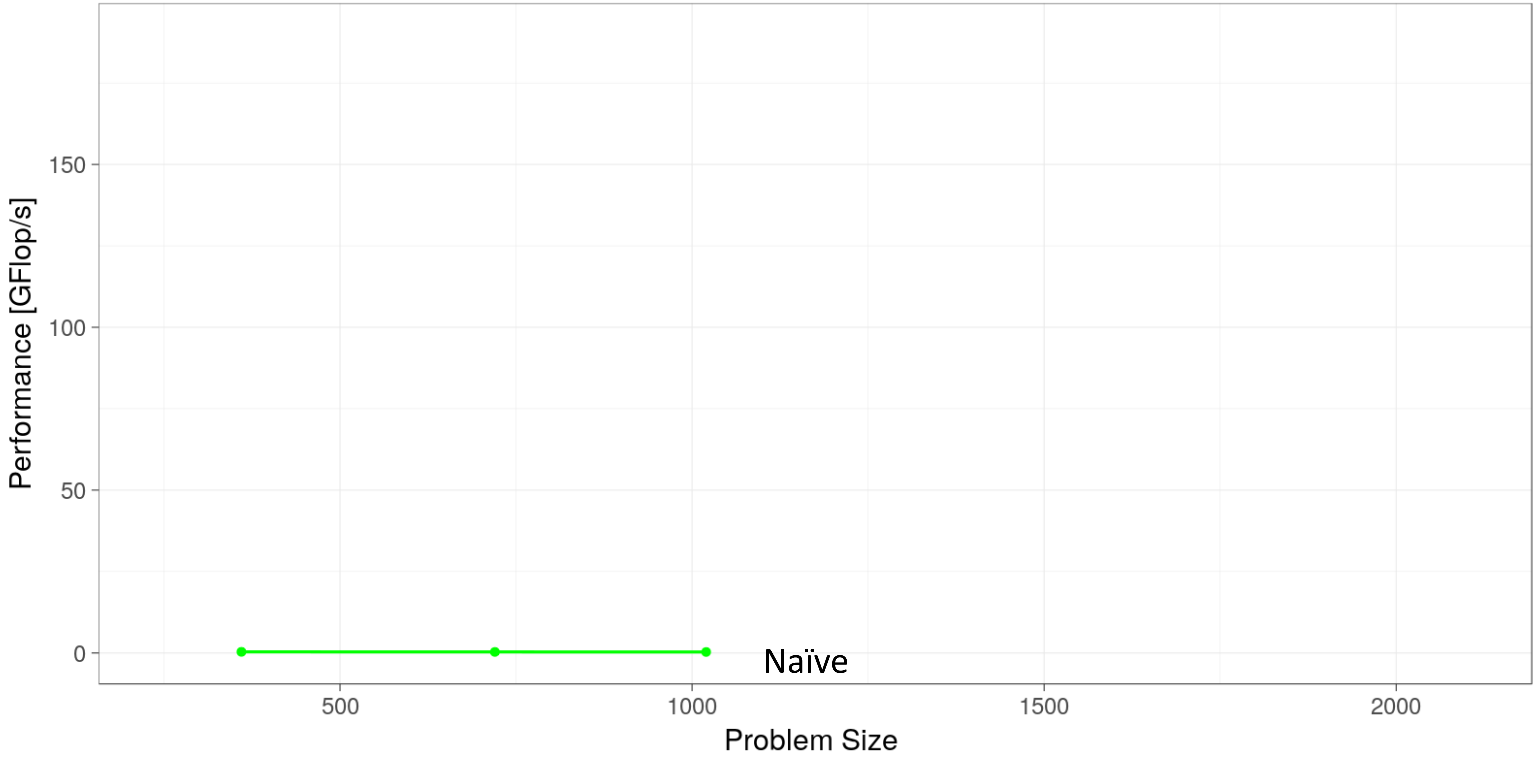
```
@dace.program
def gemm(A: dace.float64[M, K], B: dace.float64[K, N],
         C: dace.float64[M, N]):
    # Transient variable
    tmp = np.ndarray([M, N, K], dtype=A.dtype)

    @dace.map
    def multiplication(i: _[0:M], j: _[0:N], k: _[0:K]):
        in_A << A[i,k]
        in_B << B[k,j]
        out  >> tmp[i,j,k]

        out = in_A * in_B

    dace.reduce(lambda a, b: a + b, tmp, C, axis=2)
```



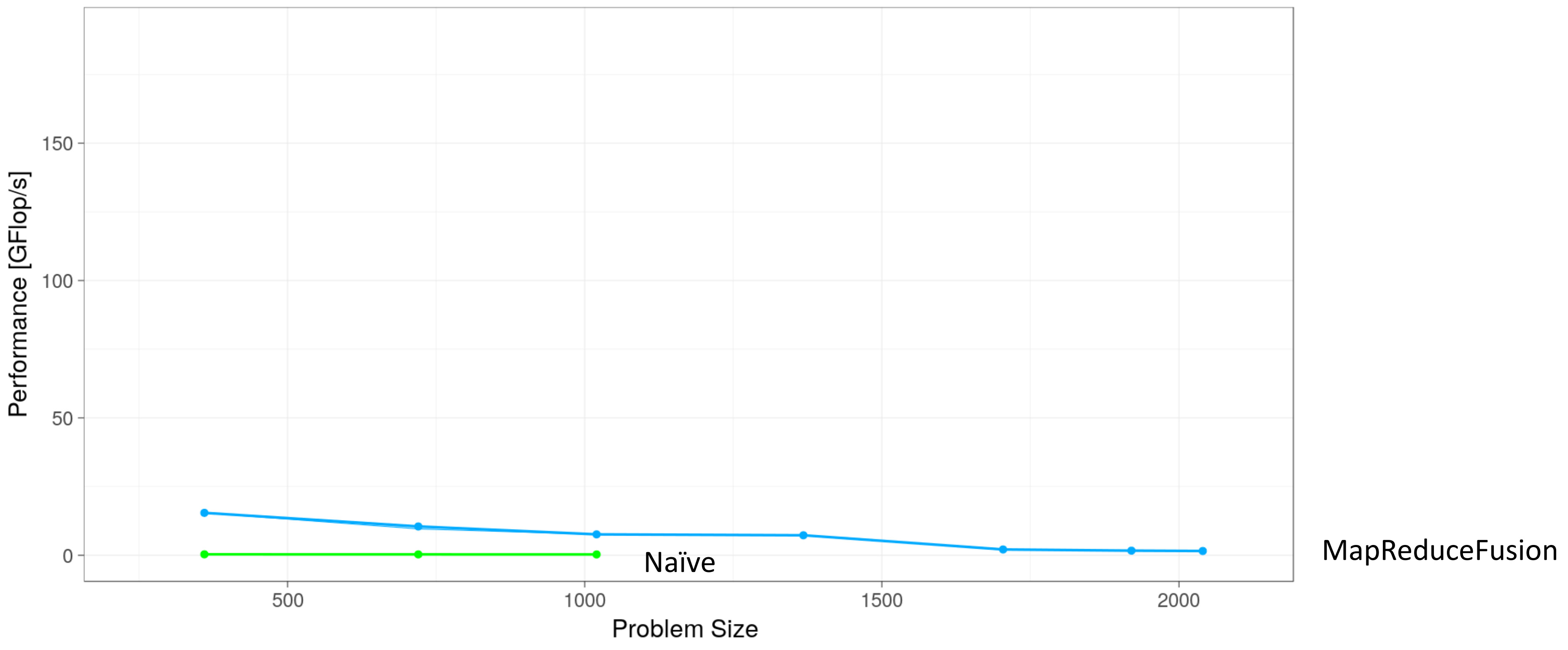
MapReduceFusion Transformation



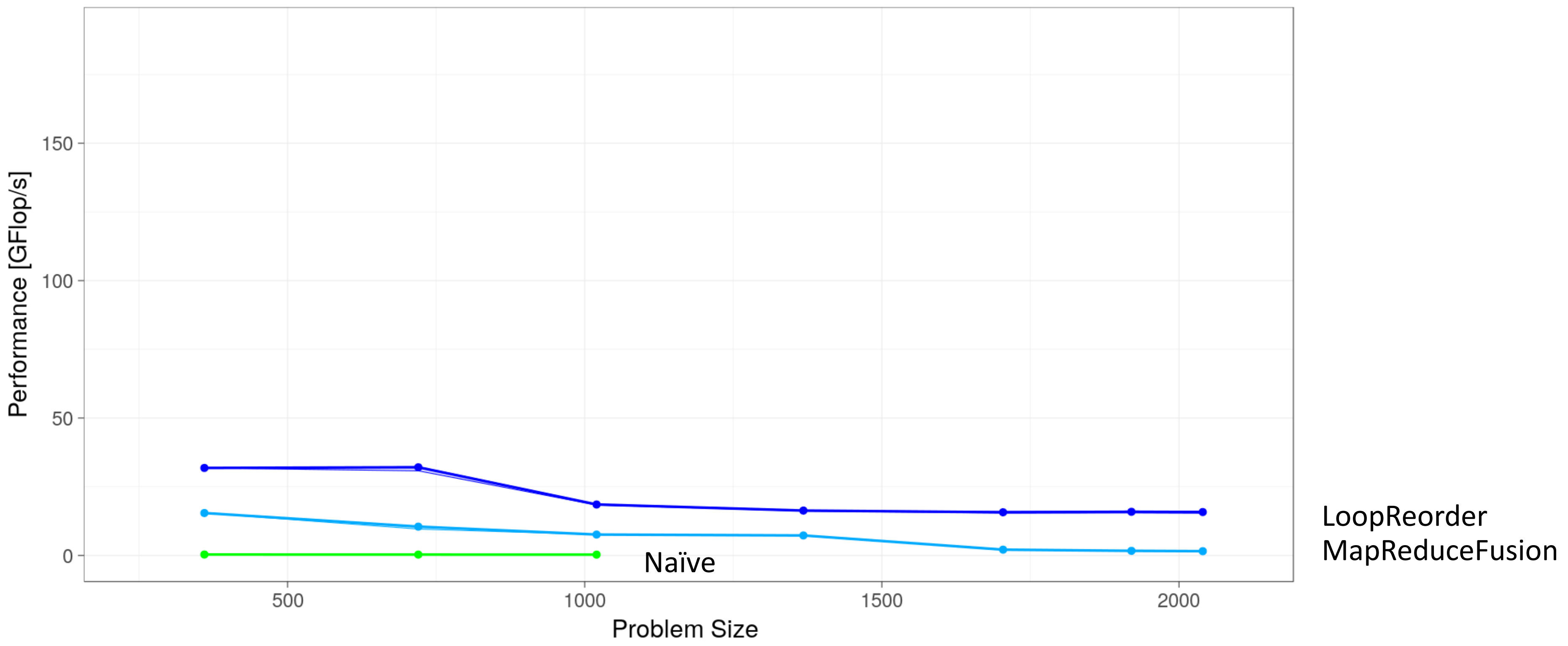
Performance



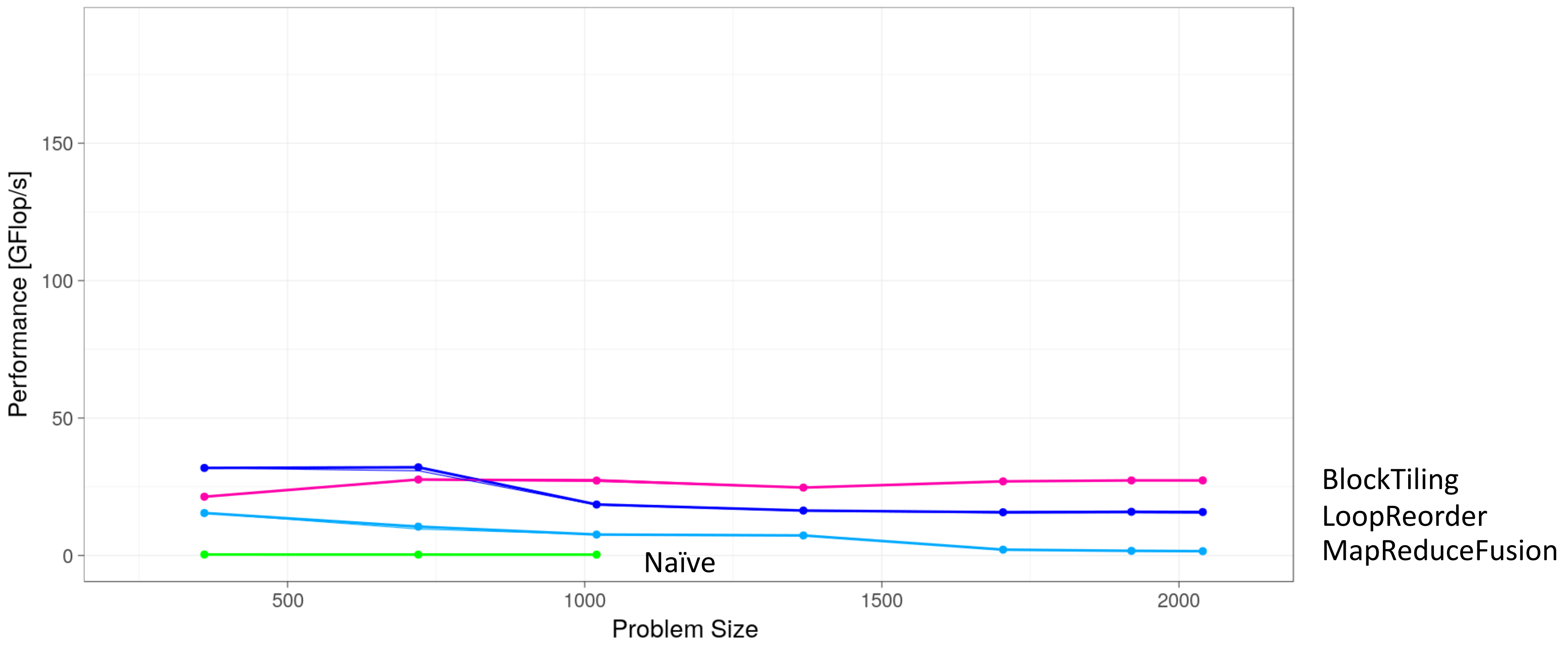
Performance



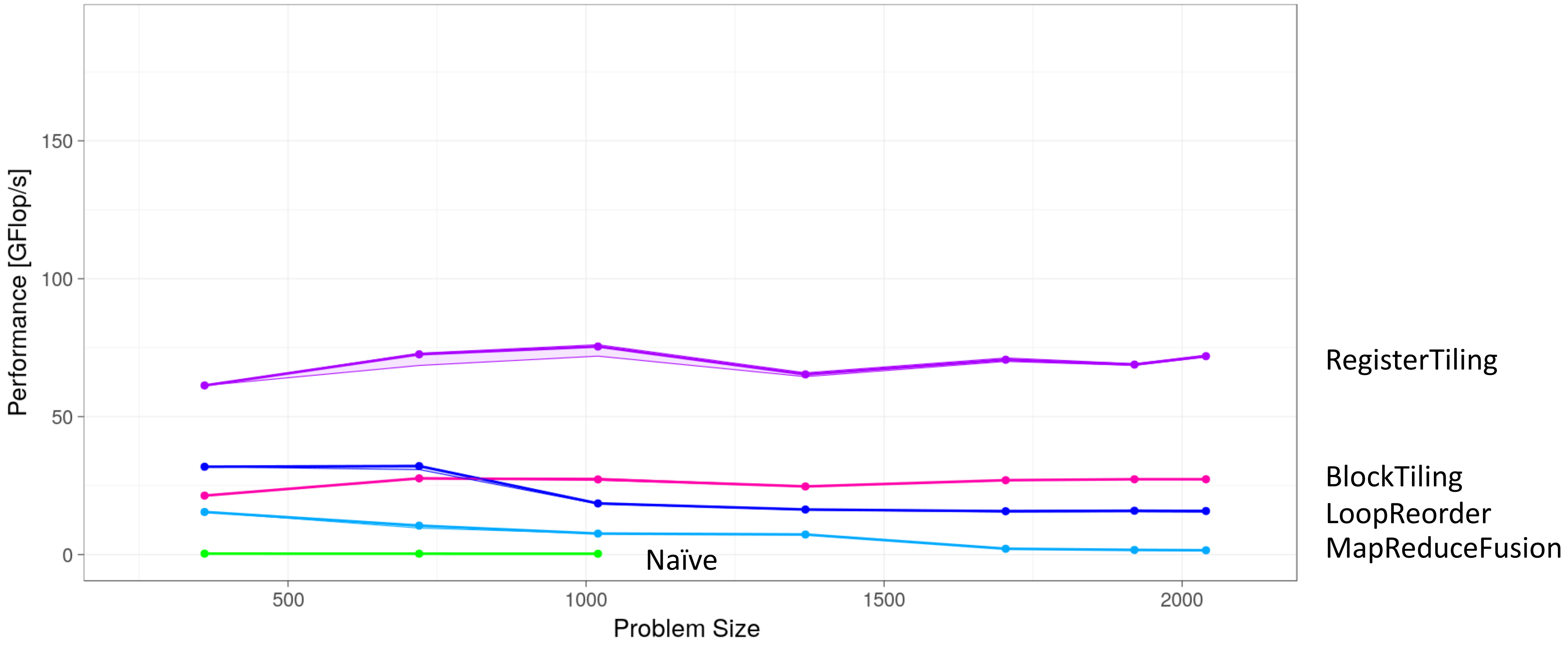
Performance



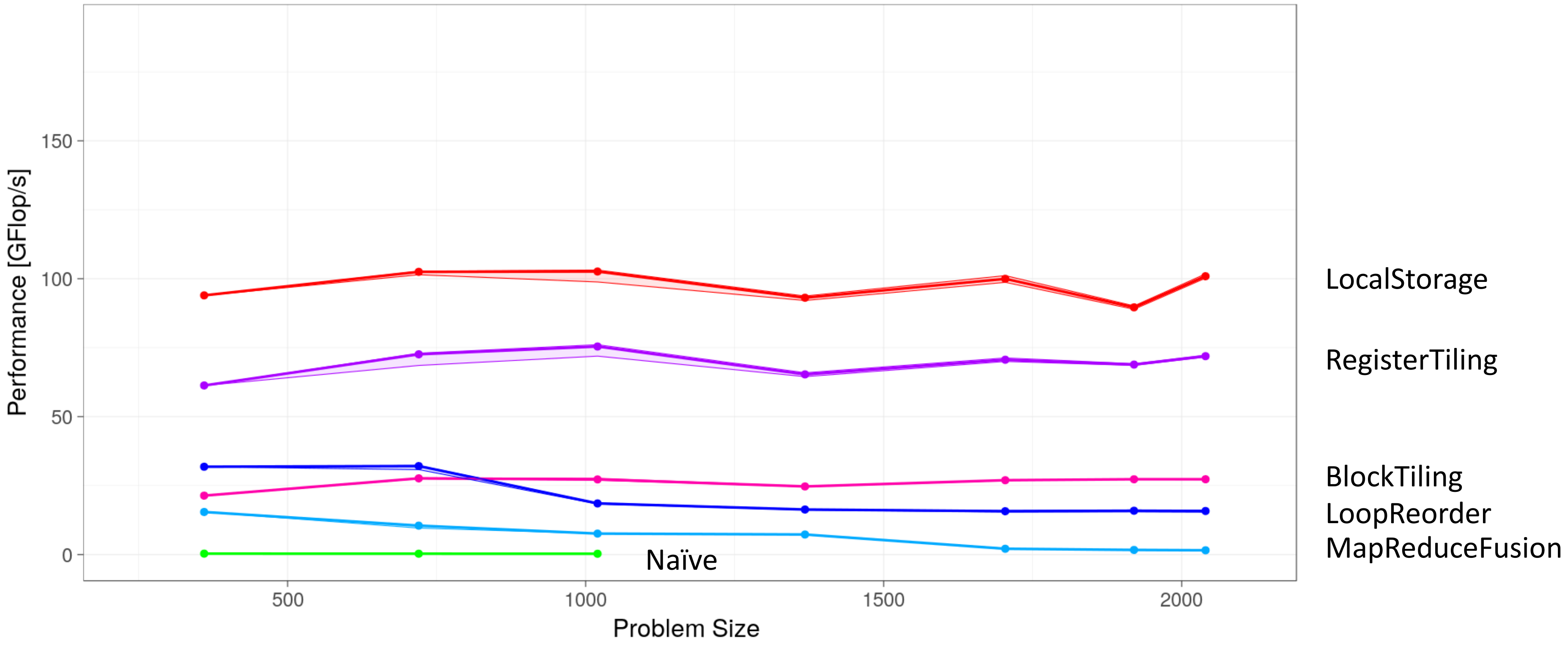
Performance



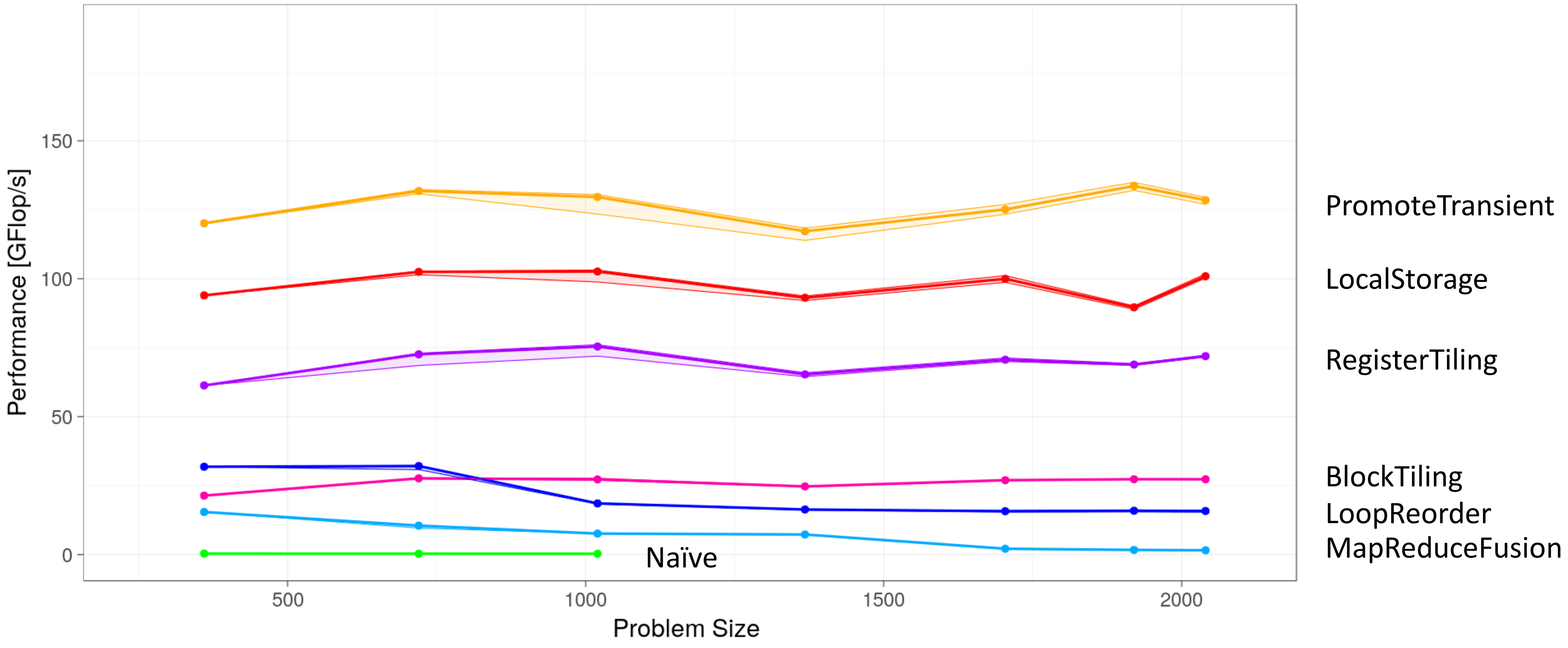
Performance



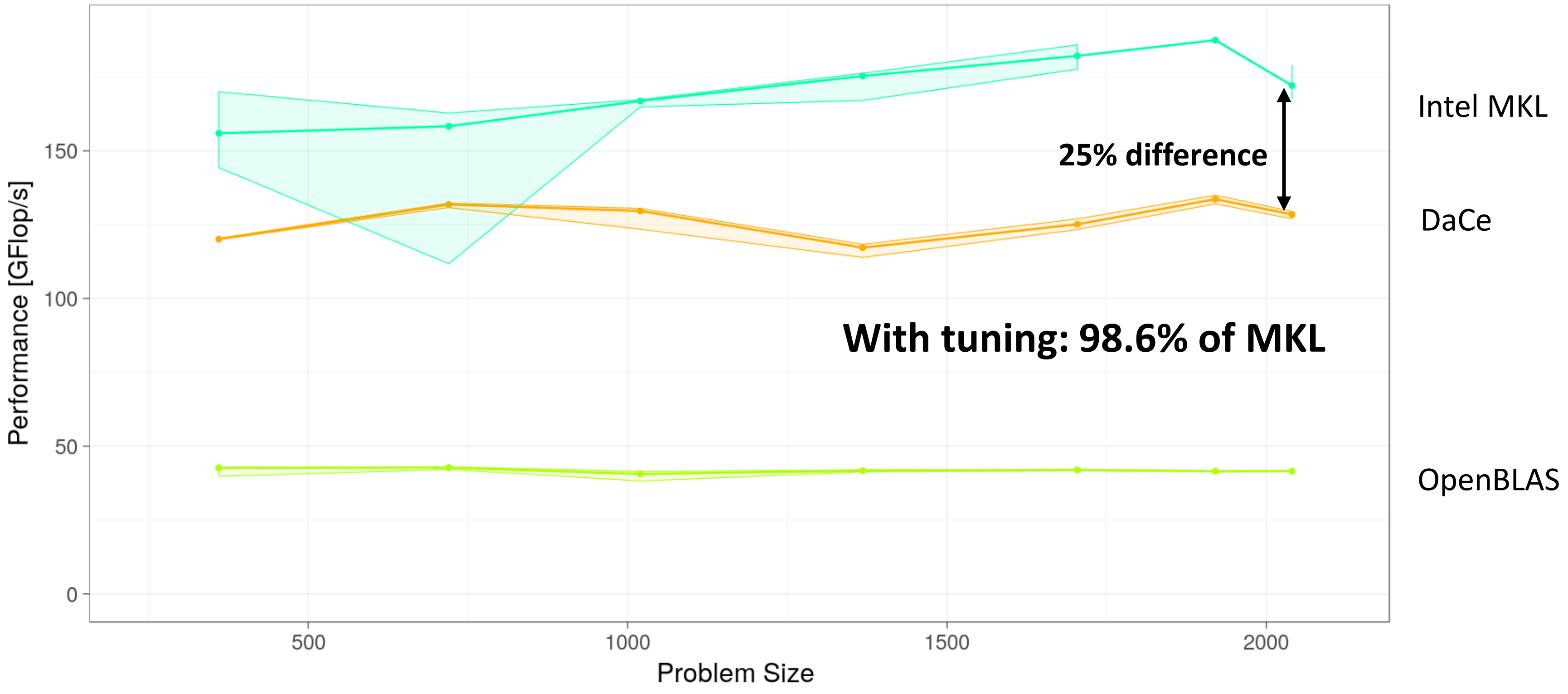
Performance



Performance

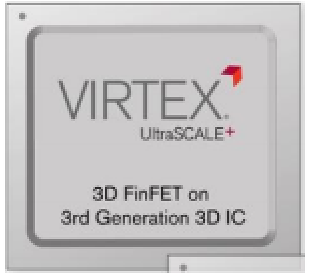


Performance



Intel Xeon E5-2650 v4

NVIDIA Tesla P100



Xilinx VU9P

SDFG

General Compilers

GCC 8, Clang 6, icc 18, NVCC 9.2, SDAccel

Polyhedral Optimizers

Polly 6, Pluto 0.11.4, PPCG 0.8

Frameworks & Libraries

HPX, Halide, Intel MKL, CUBLAS, CUSPARSE, CUTLASS, CUB

Related work