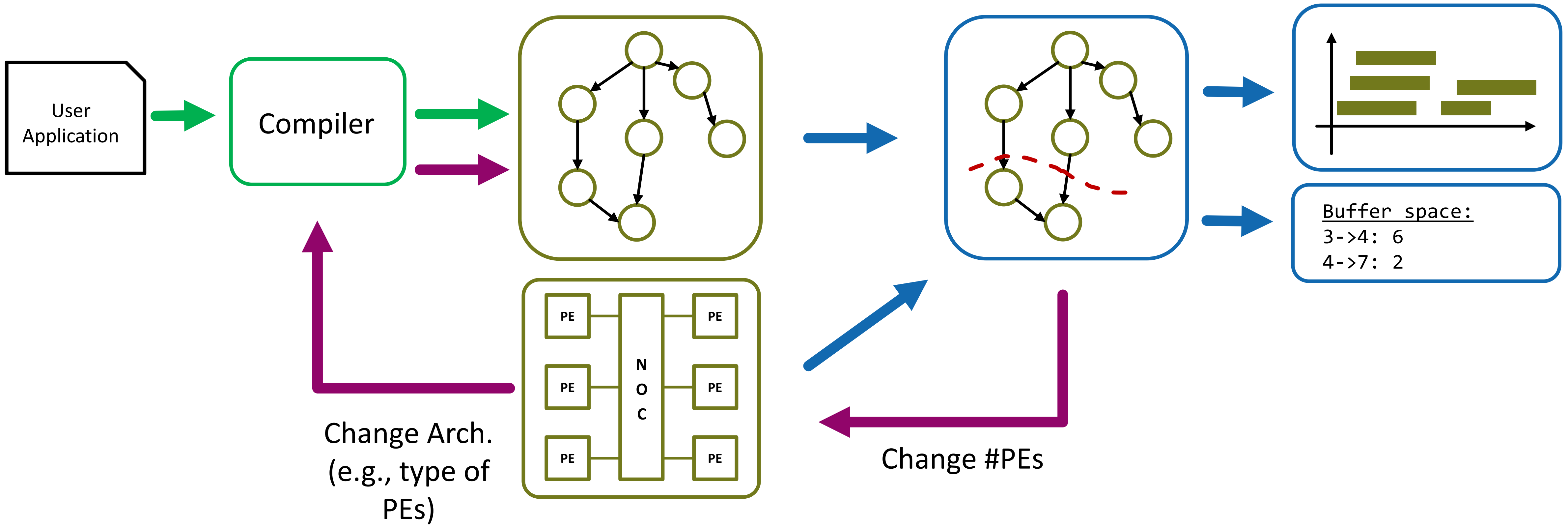
ASA: Moving beyond Scheduling

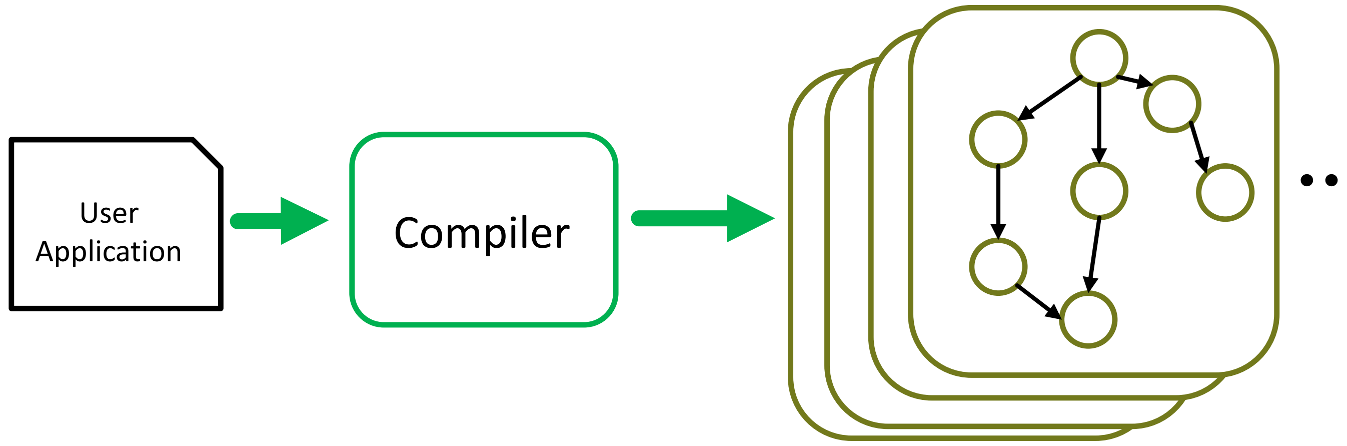
The need for a “compiler”



```
Buffer space:
3->4: 6
4->7: 2
```

🟫	Input
🟦	Scheduling
🟪	DSE
🟩	Other

The need for a “compiler”



For the compiler, we want:

- Generate all the canonical task graphs
- Evaluate each of them: scheduling or other approaches?

🟩	Input
🟦	Scheduling
🟪	DSE
🟩	Other

DaCe Overview

Domain Scientist

Problem Formulation

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

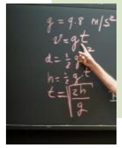
python

NumPy

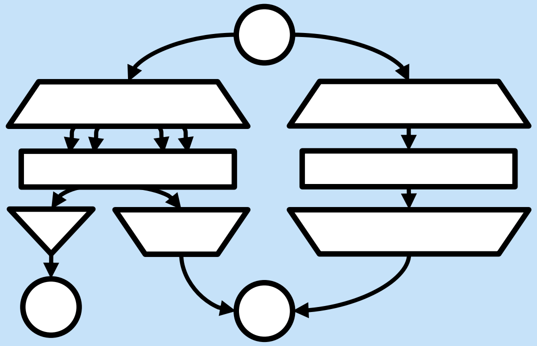
ONNX

...

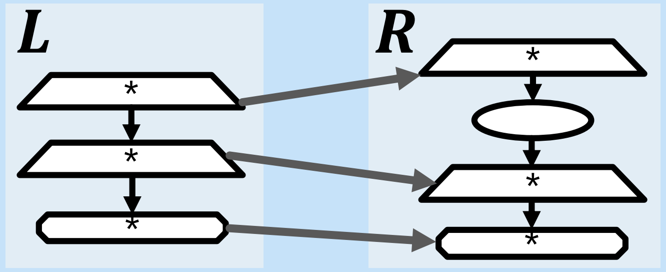
Scientific Frontend



Performance Engineer



Data-Centric Intermediate Representation (SDFG)



Graph Transformations

Transformed Dataflow

Performance Results

System

Hardware Information

Compiler

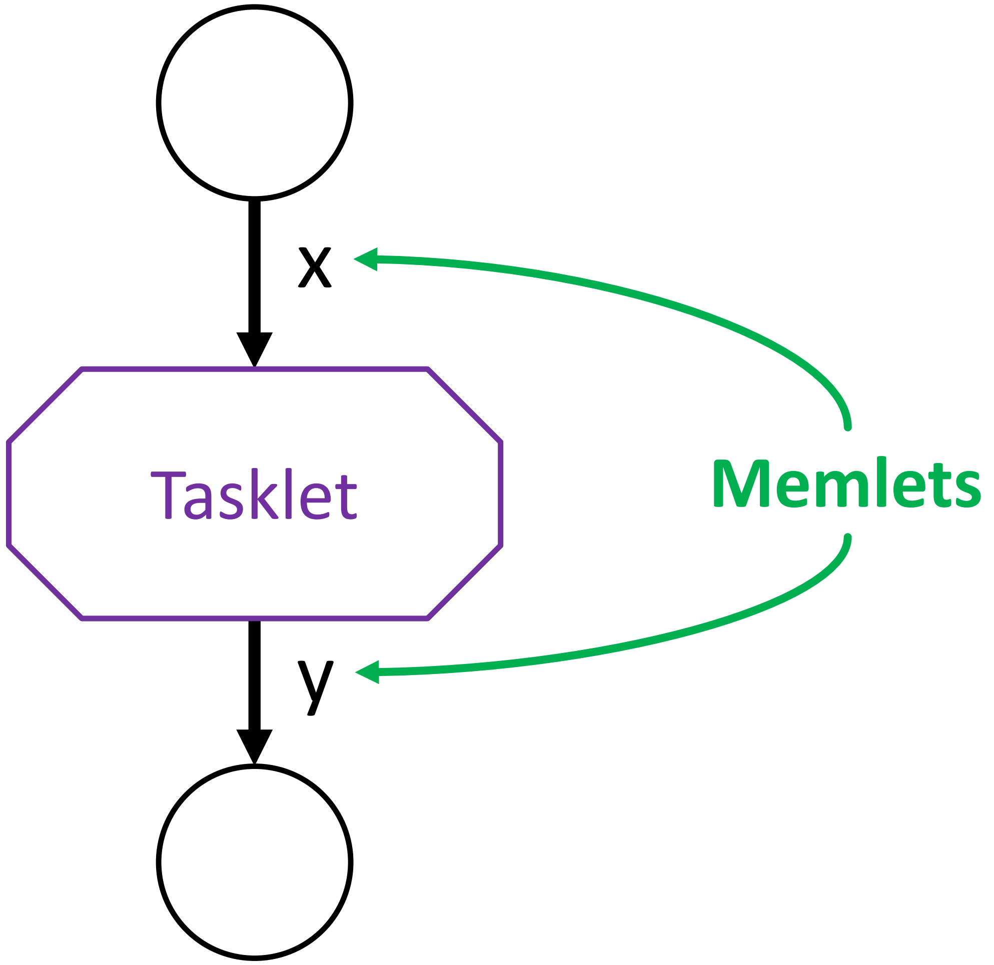
Runtime

CPU Binary

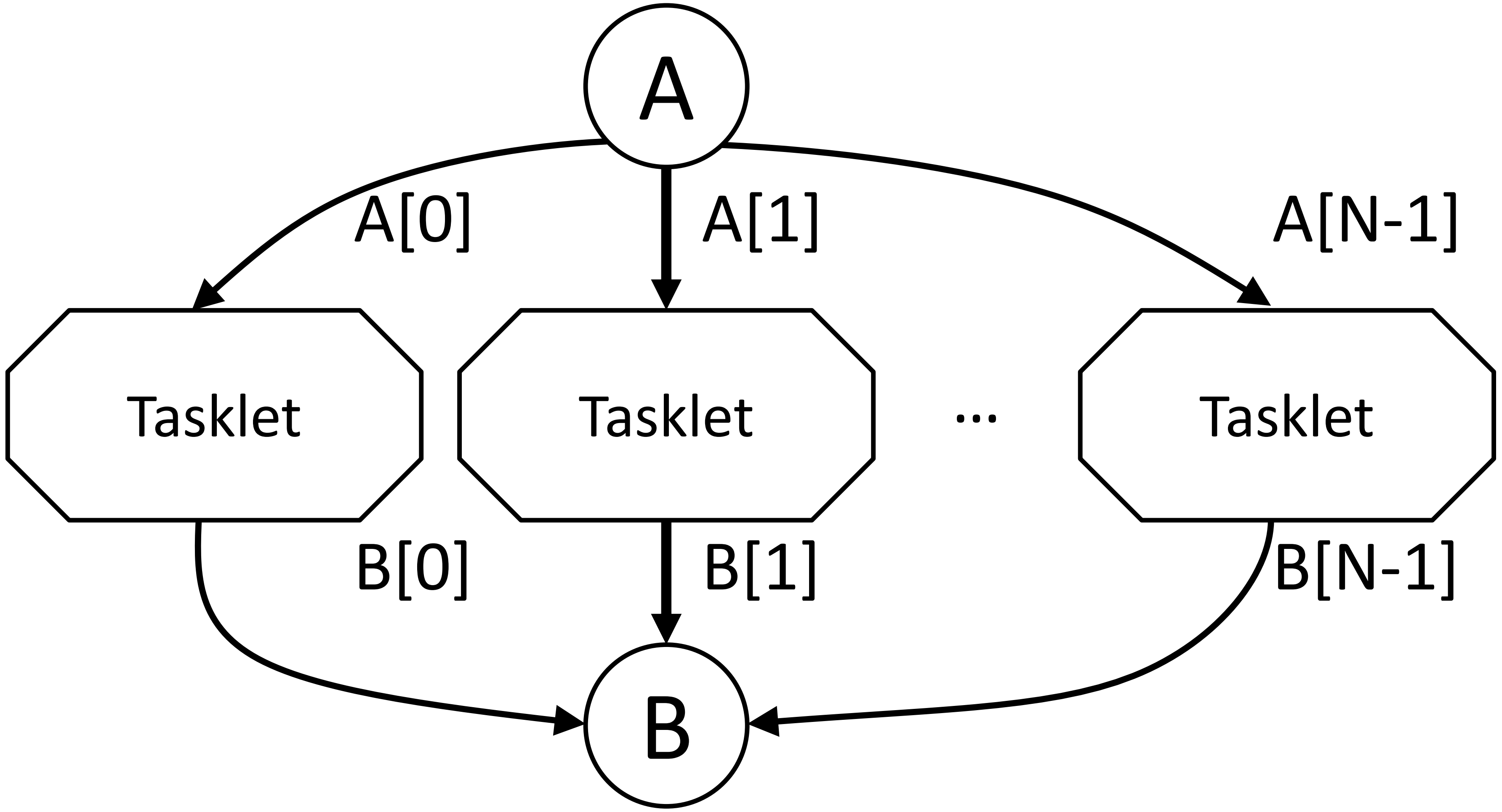
GPU Binary

FPGA Modules

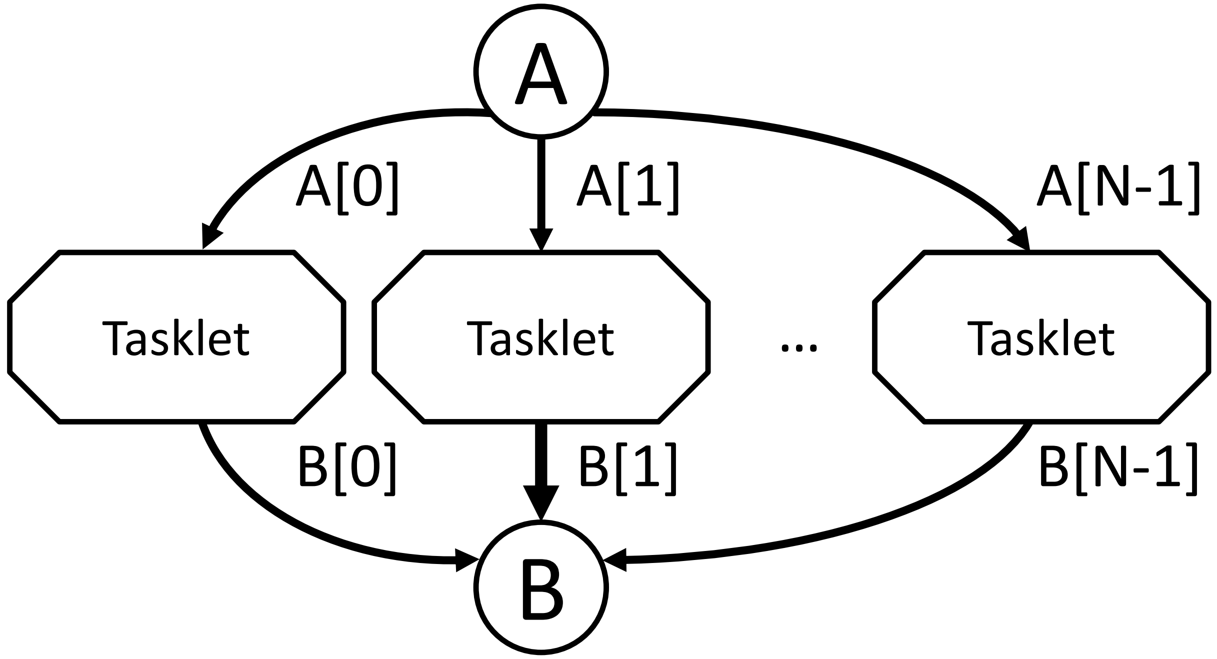
Dataflow Programming in DaCe

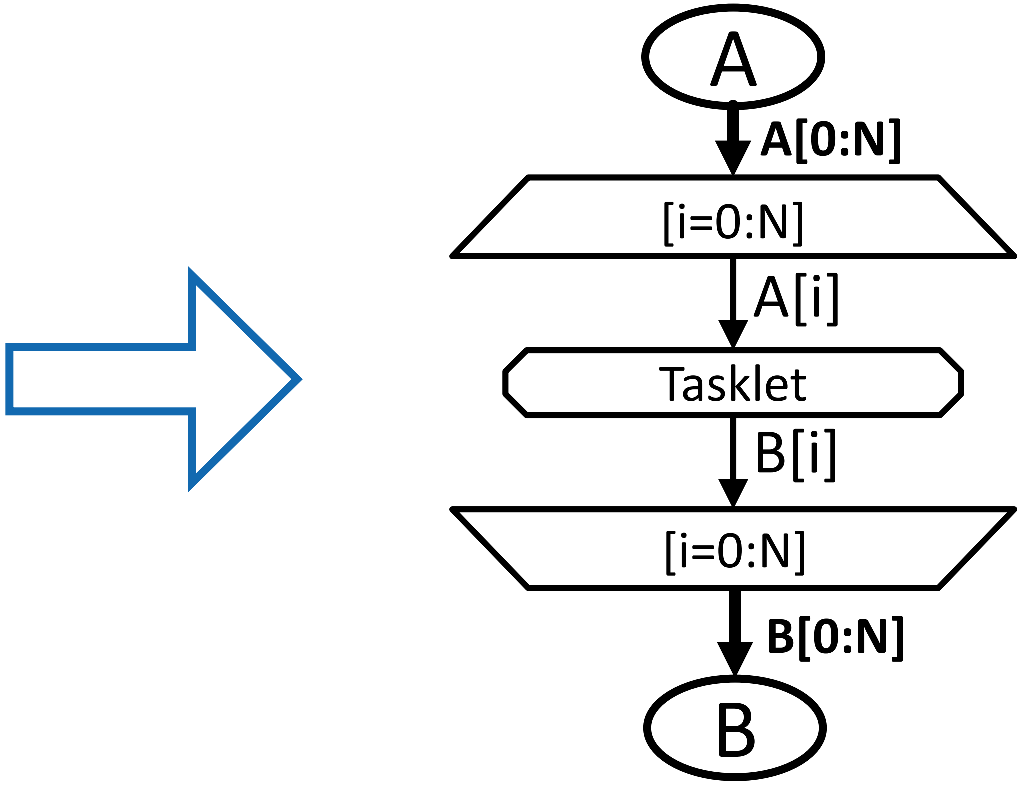


Parallel Dataflow Programming

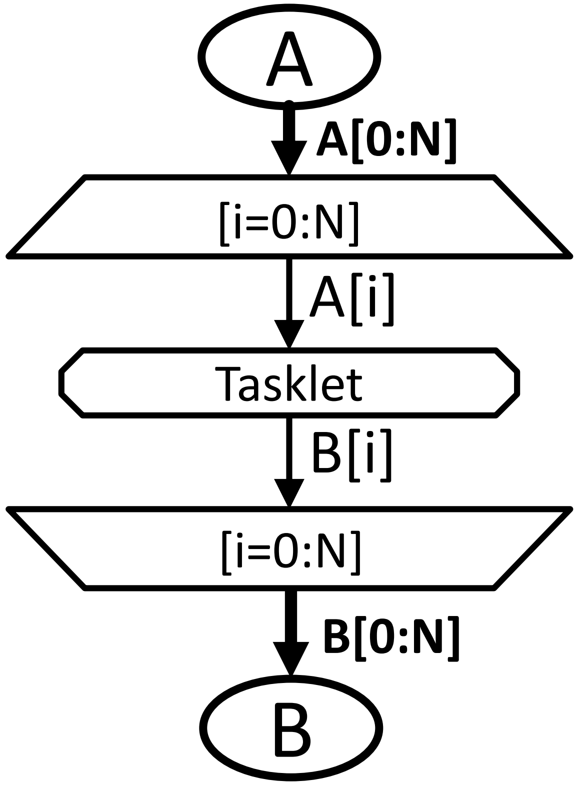


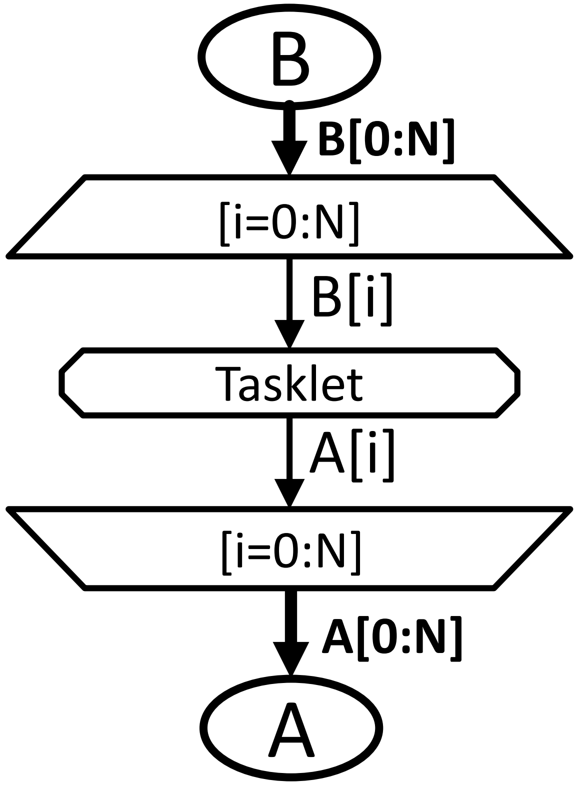
Parallel Dataflow Programming



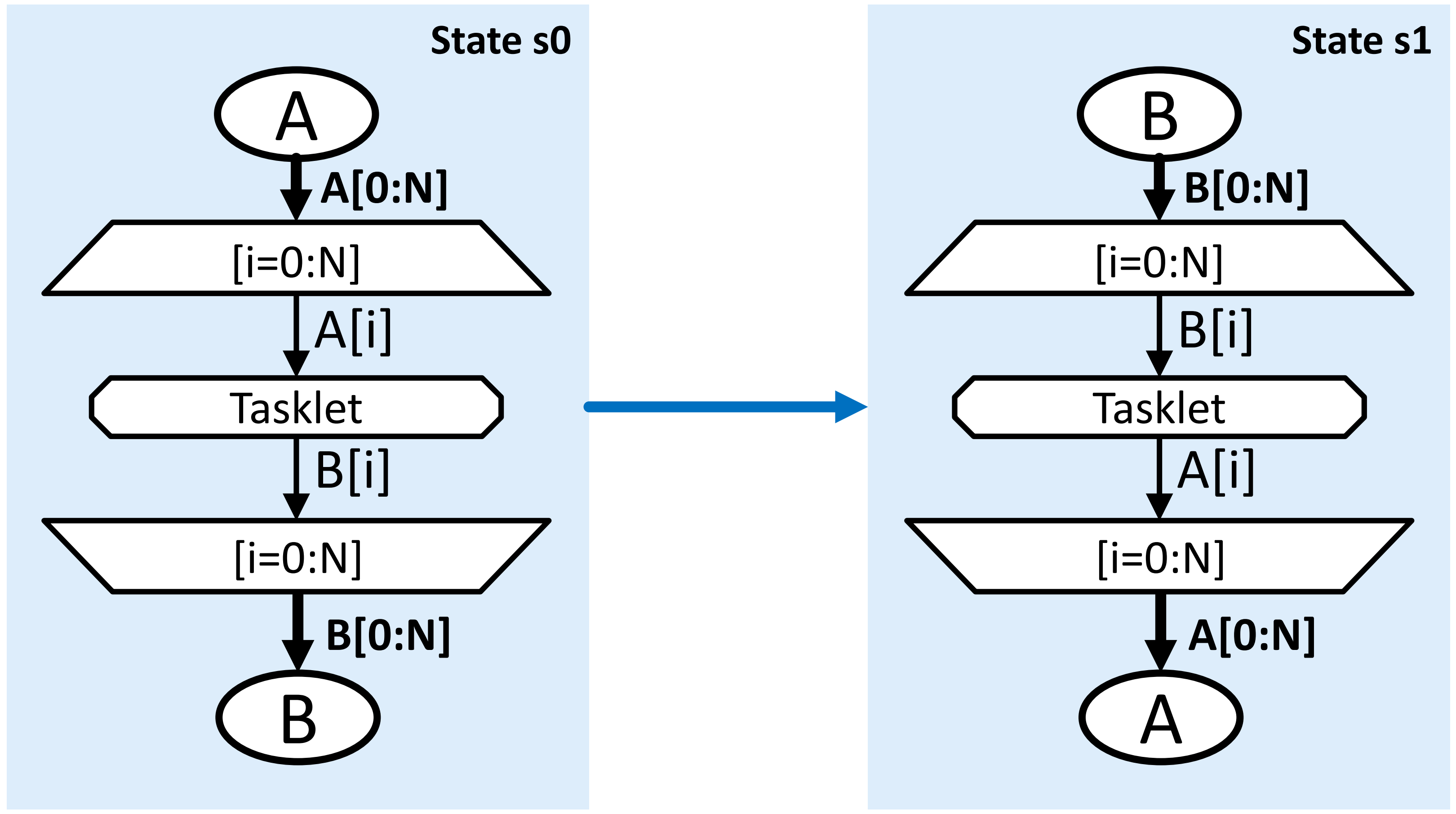


Stateful Dataflow Parallel Programming in DaCe

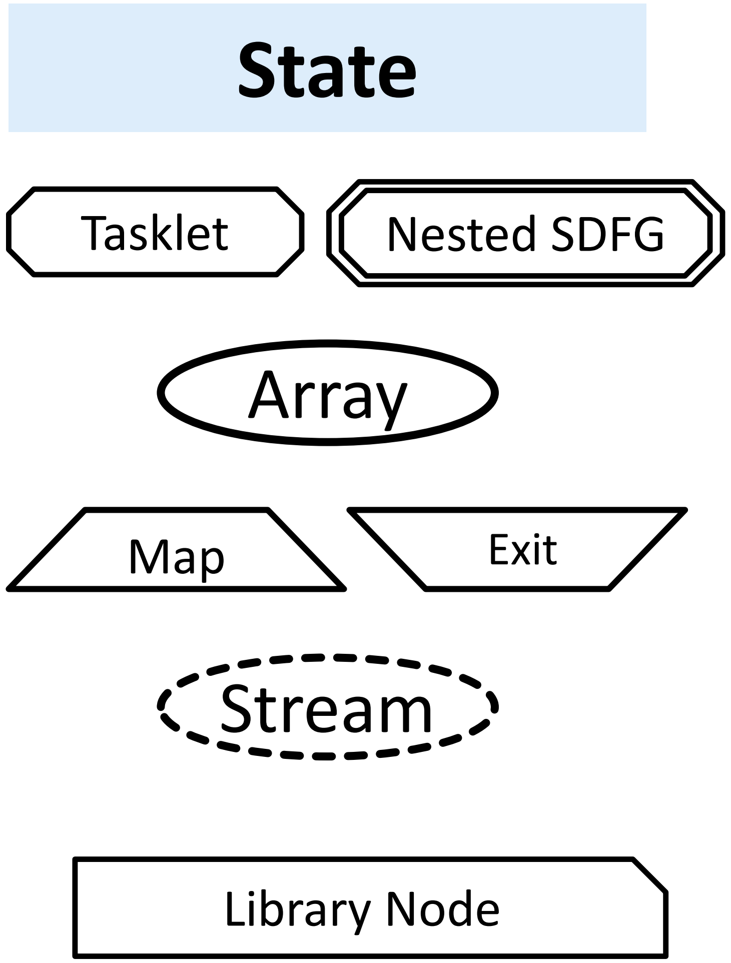




Stateful Dataflow Parallel Programming in DaCe



Meet the Nodes



State machine element

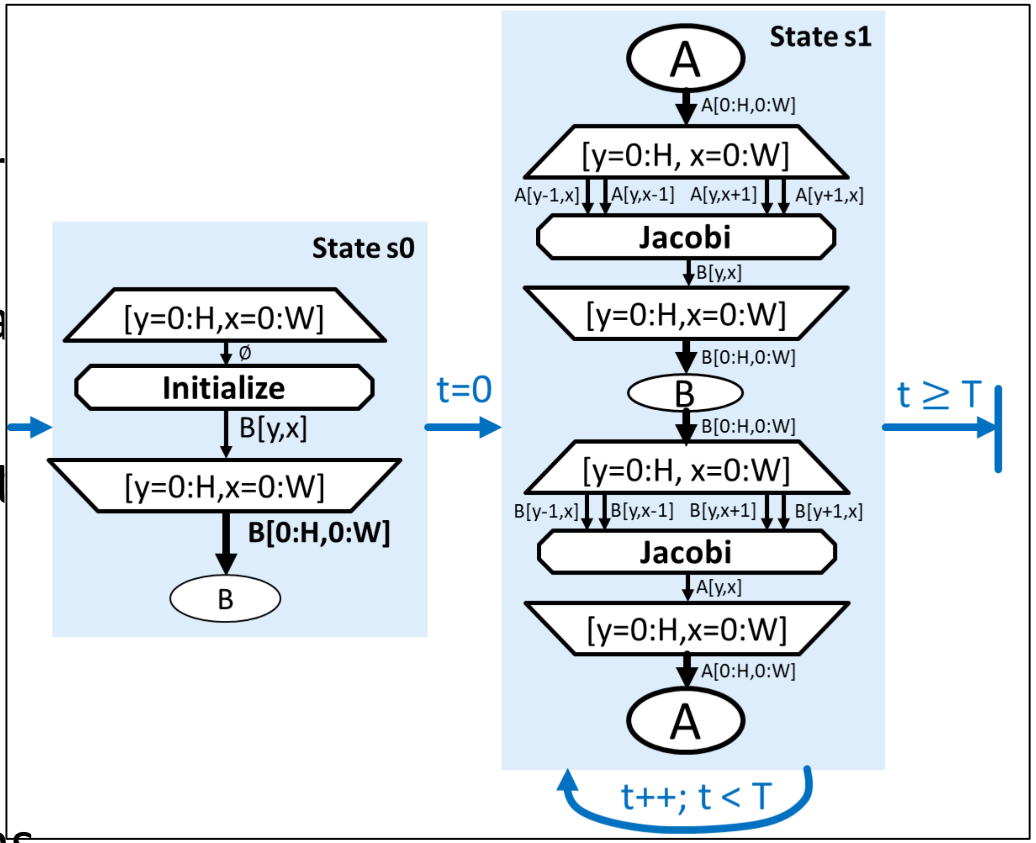
Fine-grained computation

N-dimensional data conta

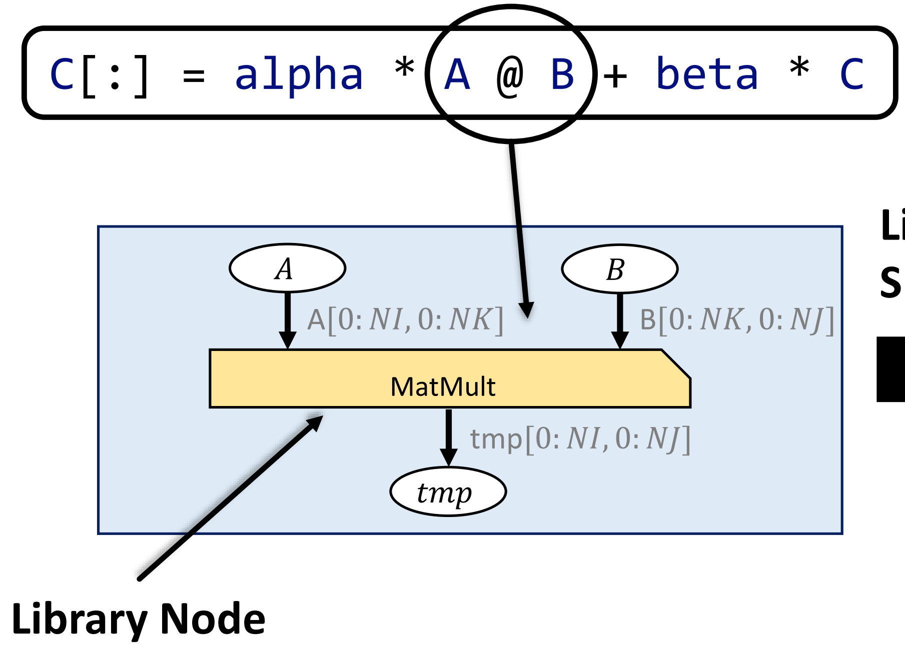
Parametric graph abstract

Streaming data container

Represent call to functions

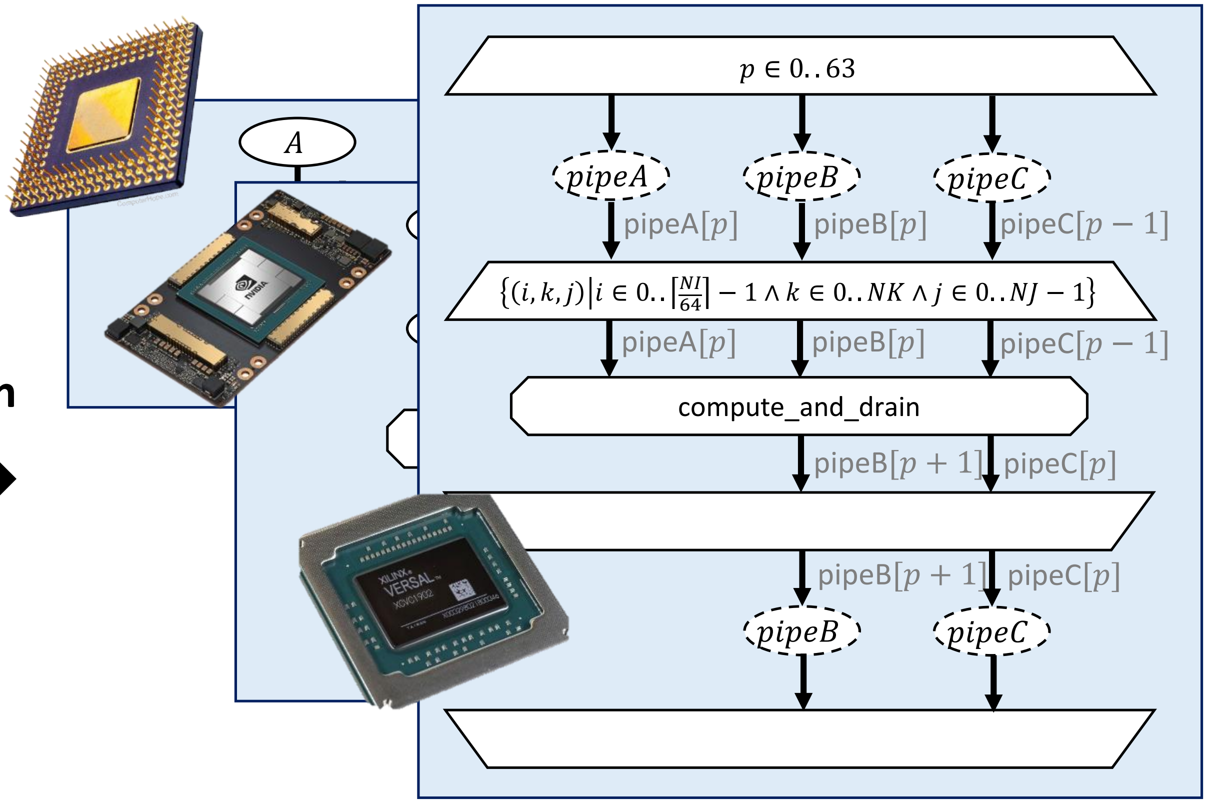


Library Nodes



```
C[:] = alpha * A @ B + beta * C
```

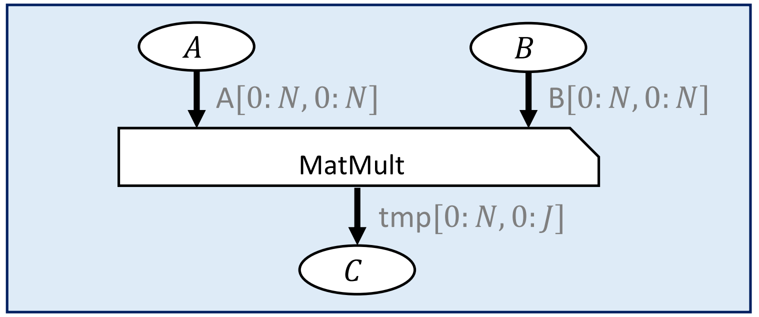
Library Specialization



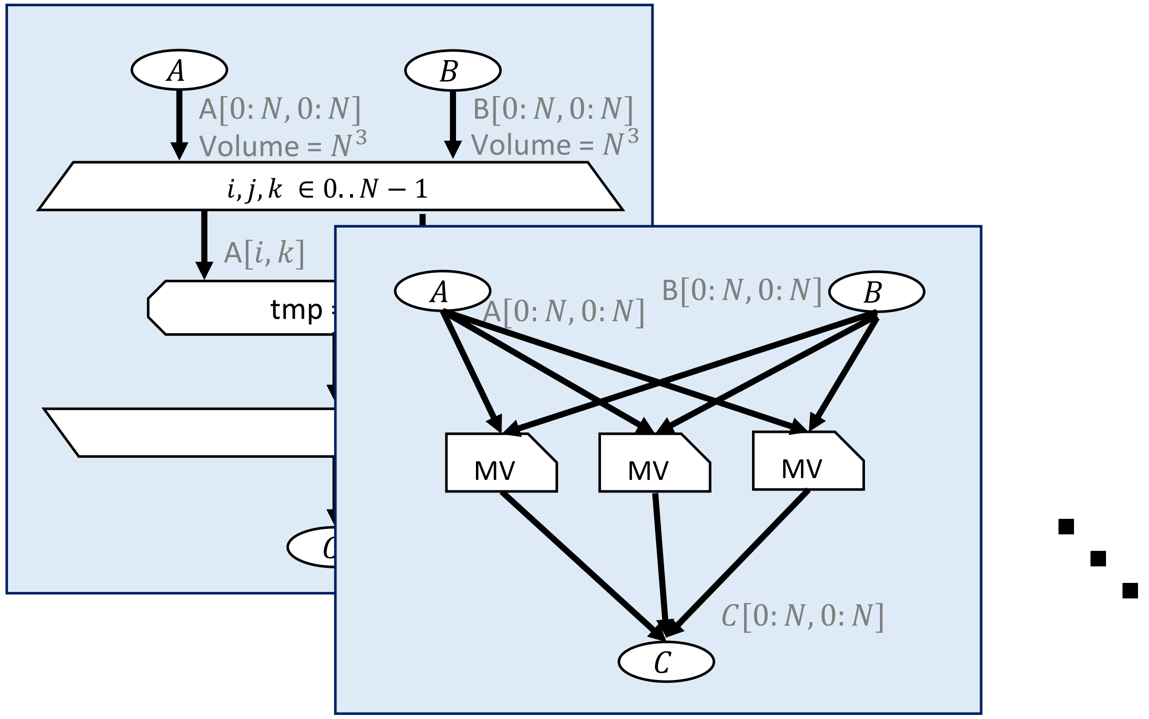
How can we use DaCe?

The idea is to leverage DaCe (frontend), intermediate representation and transformations to understand how to build a canonical task graph

1. Create a **collection** of library nodes and their own ***canonical*** expansions

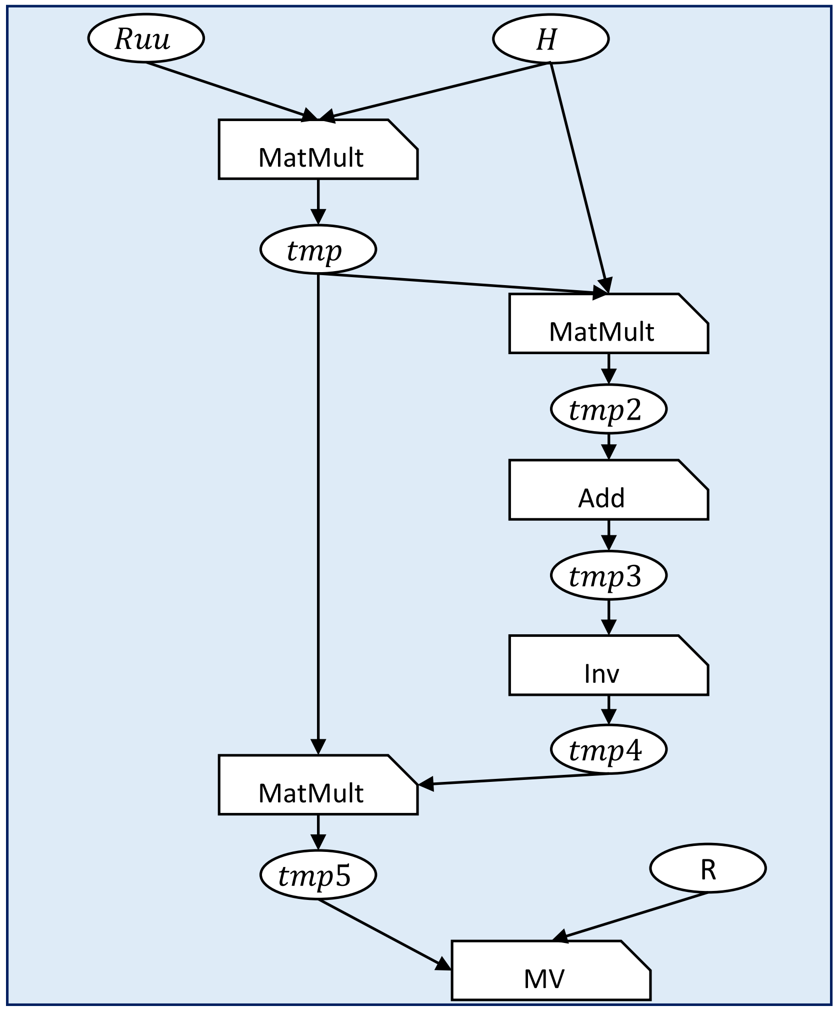


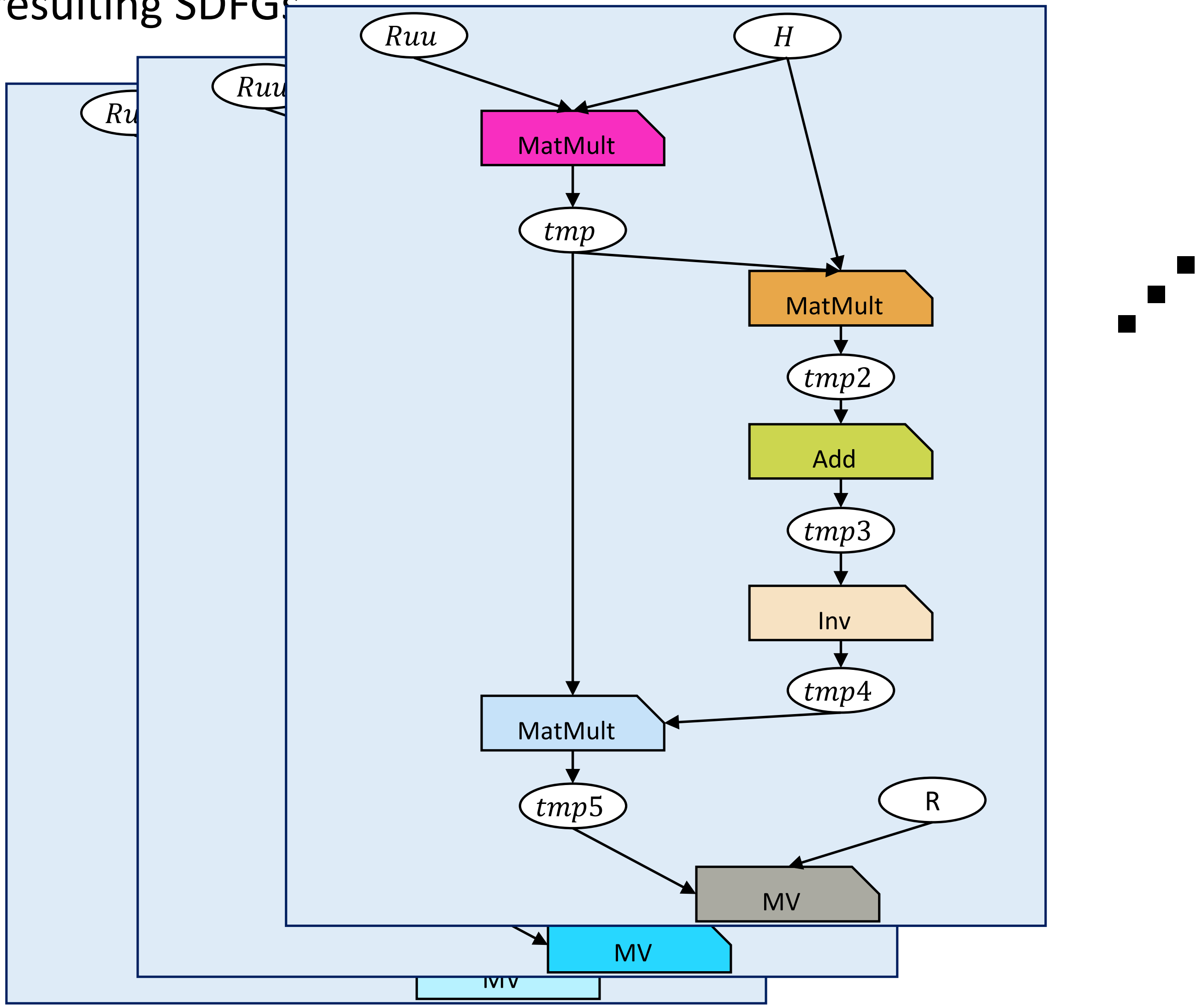
Expanded as



How can we use DaCe?

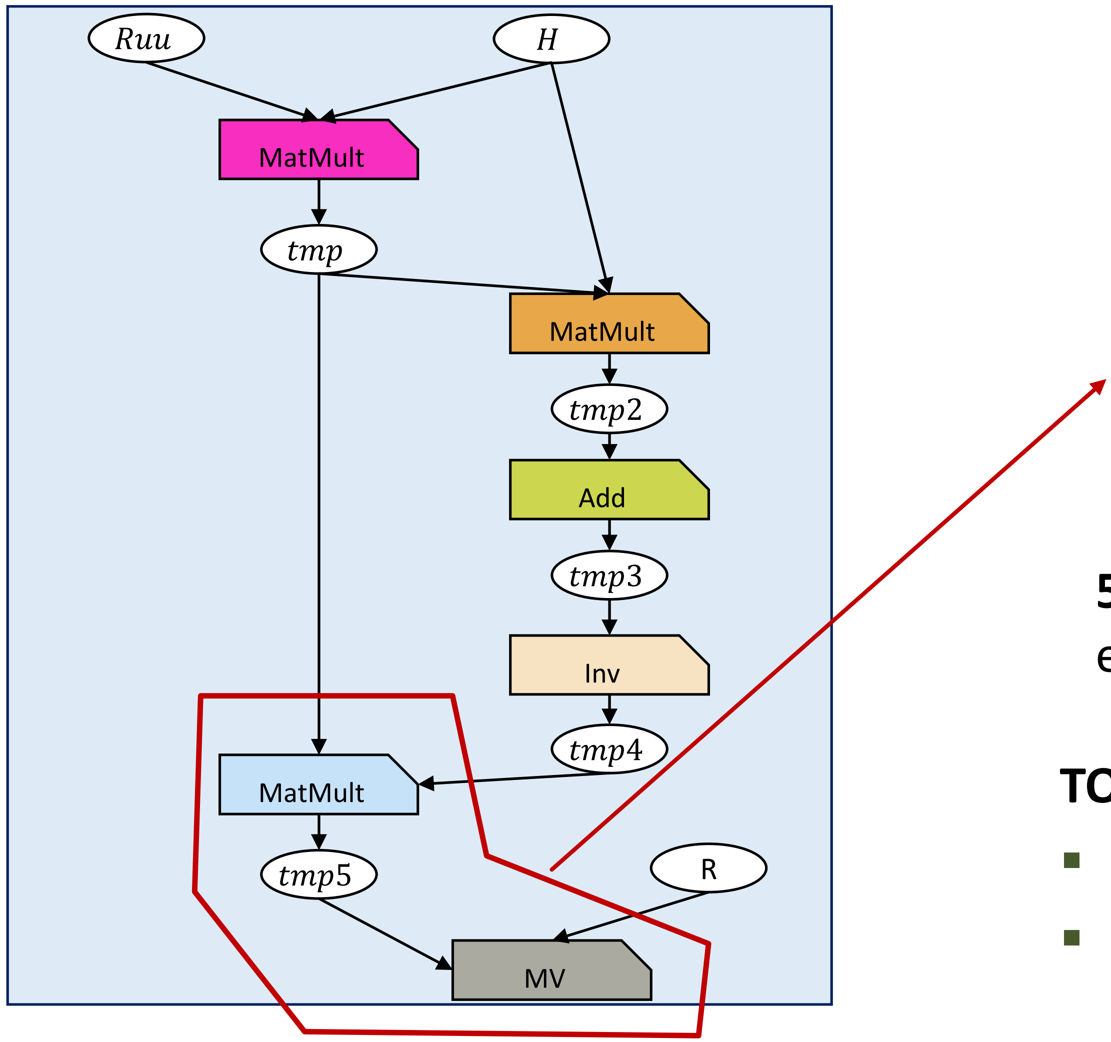
3. Expand all the library nodes. Enumerate all the resulting SDFGs





How can we use DaCe?

4. Understand whether we should introduce some buffer nodes



For example, let's assume that MV reads the input by row:

- If the MatMult produces the data by row, we can (potentially) stream between the two
- If MatMul produces the data by column, we need to store the result in a buffer node

5. Rebuild the starting task graph ("undo" the expansion)

TODO:

- Understand engineering effort
- Understand how extensible is this approach

Misc

- We would like to understand if there are common computational patterns in 5G/Radio appl. **Do you have any reference?**