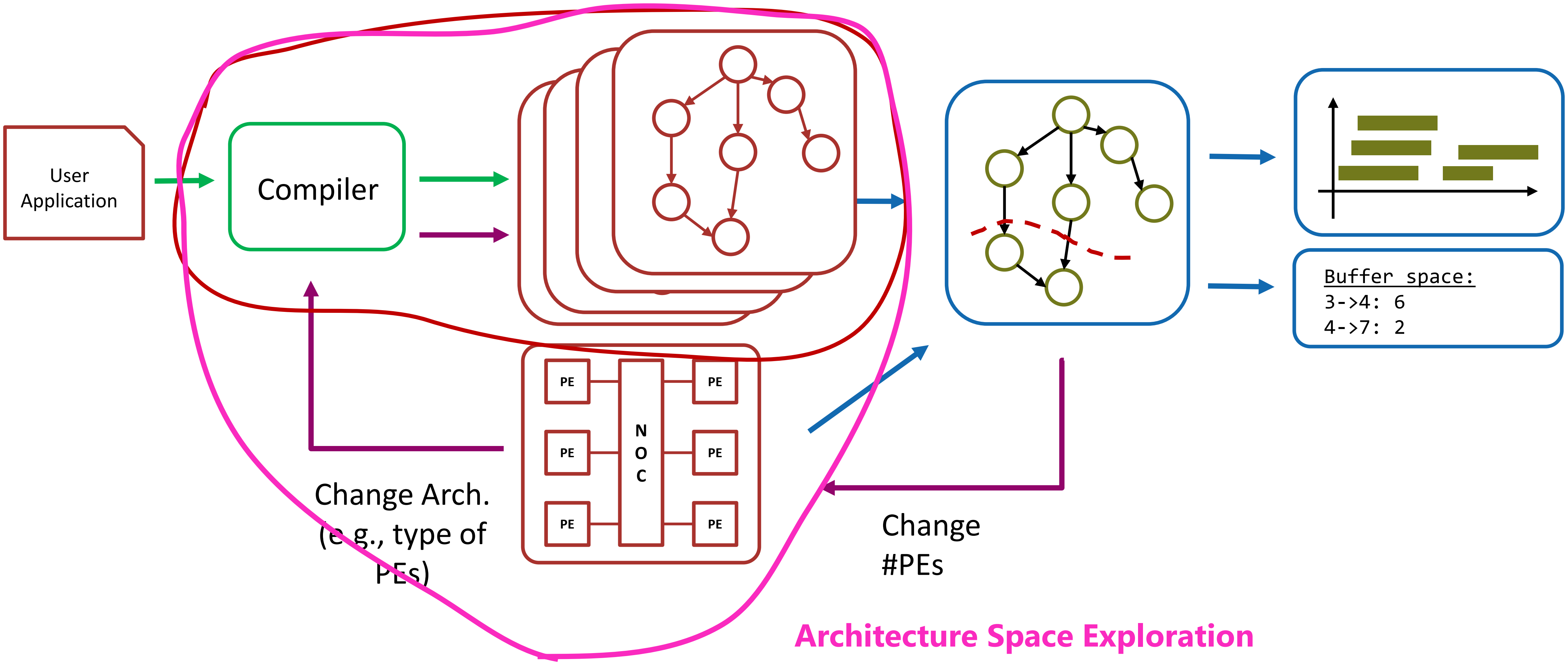
ASA: Enabling DSE

Different types of exploration

Application Space Exploration

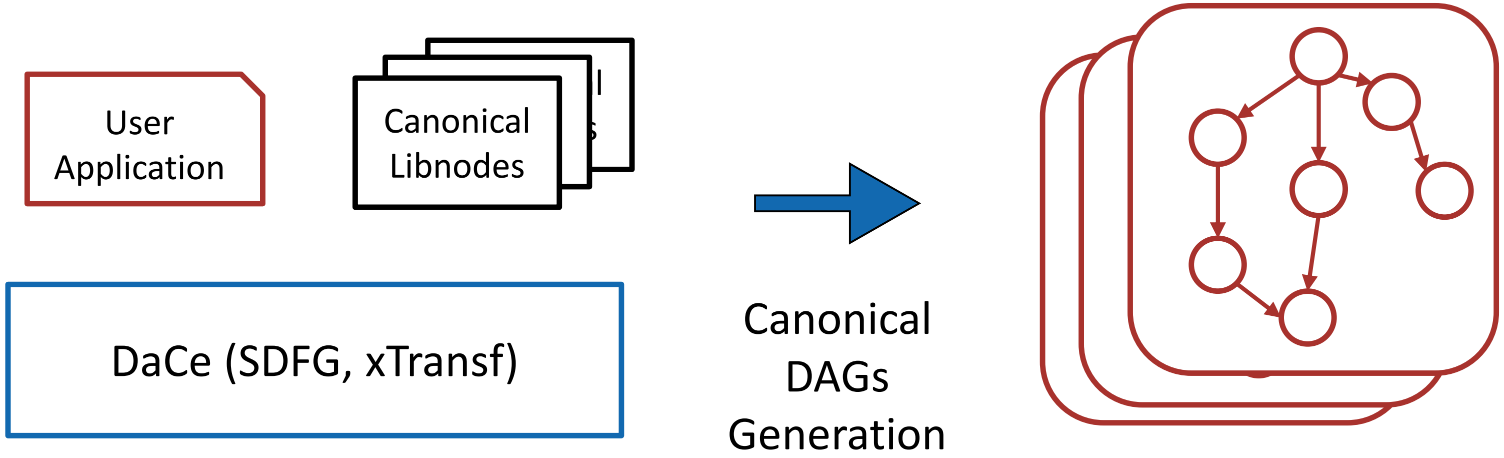


Architecture Space Exploration

Goal: Find application representation and architecture that give max performance

Approach

We want to use DaCe (IR, LibNode, Transformations) to enable all of this (“data-centric and compiler approach”)



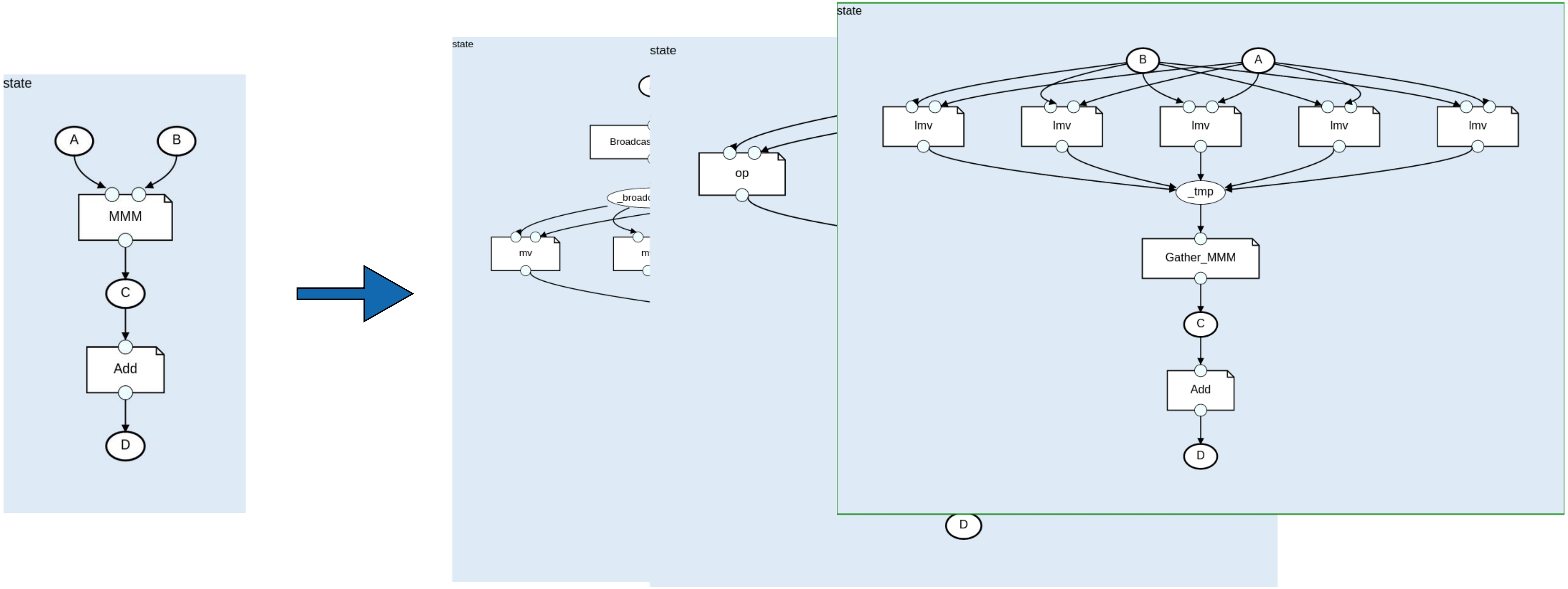
User Application

Canonical Libnodes

DaCe (SDFG, xTransf)

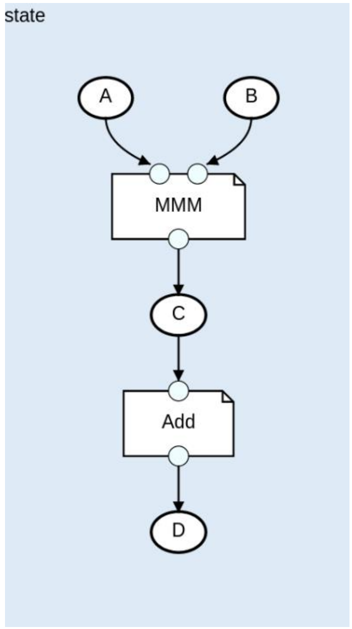
Canonical DAGs Generation

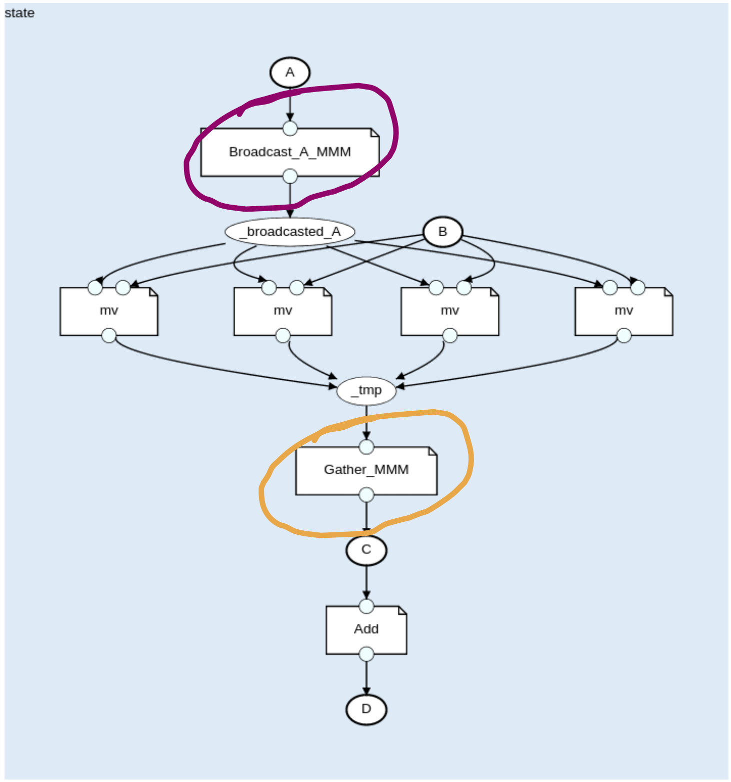
Canonical LibNodes



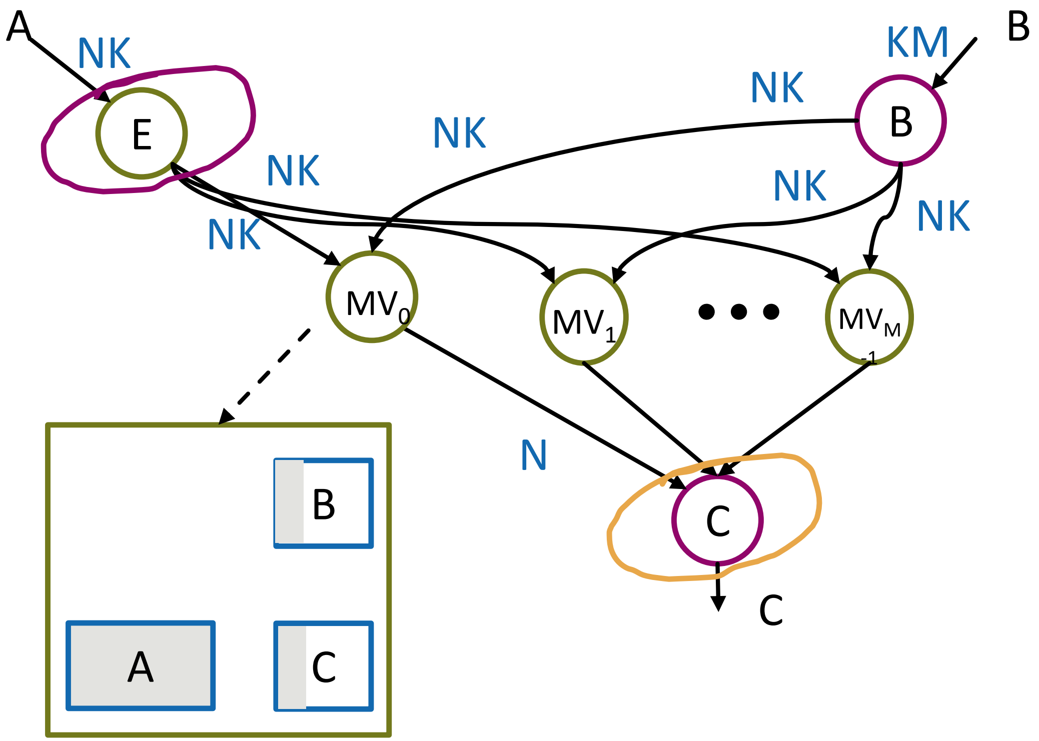
Current constraint: The application SDFG must be composed only by LibNodes and Access nodes

Canonical LibNodes



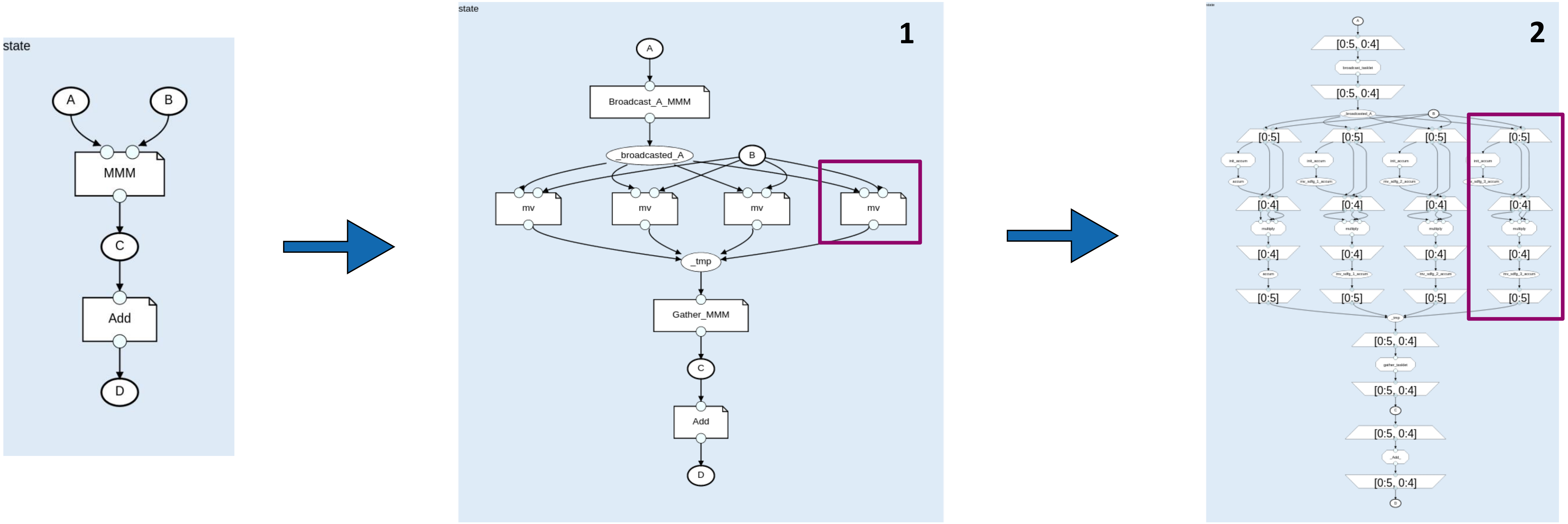


A is an NxK matrix, B is KxM and C is NxM



Why the gather/broadcast nodes: keep track of these operations (they do not come for free)

Canonical LibNodes

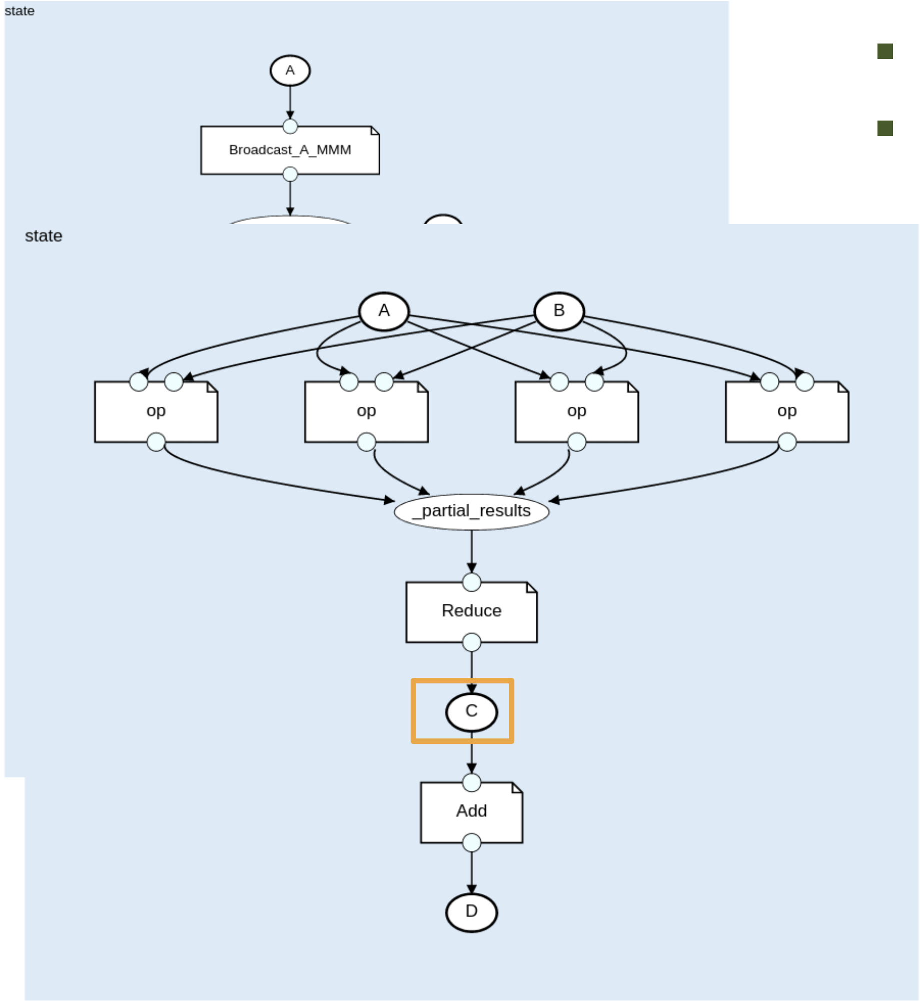


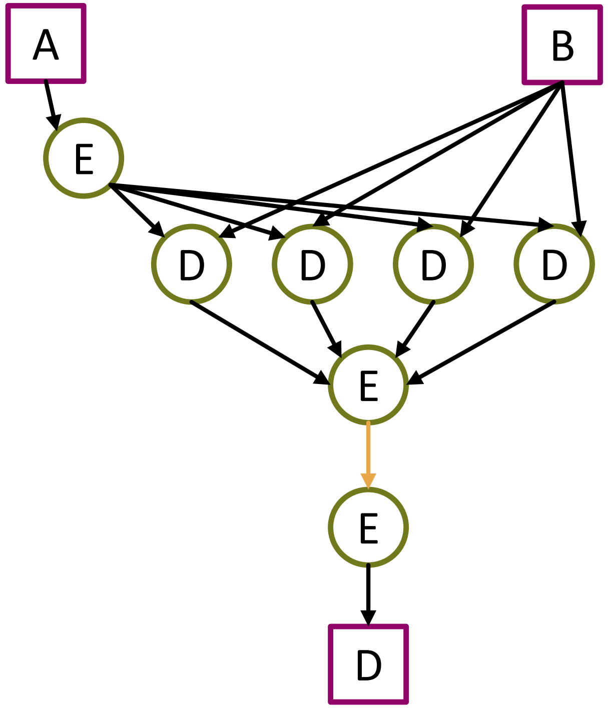
We need to “pull out” the Canonical DAG from one of these two representations

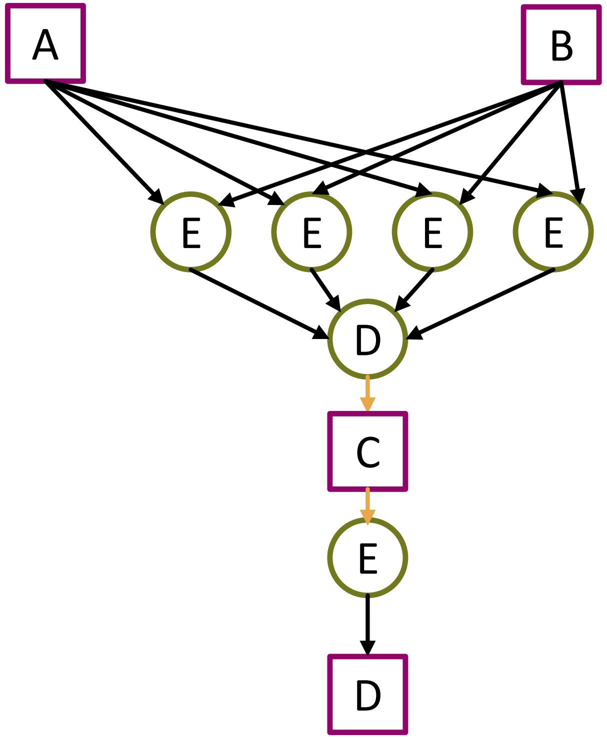
- 1 is more straightforward, but we will need anyway the fully expanded SDFG to analyze data movements
- 2 more rich, but we need to track down node-task association (no such mechanism in DaCe currently)

Building the canonical DAG

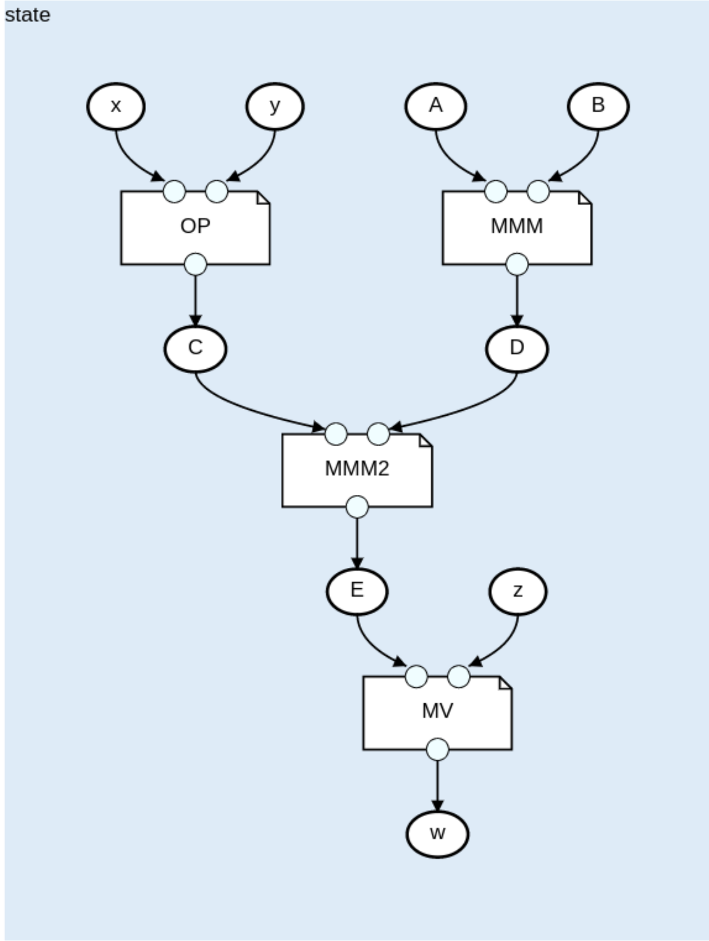
- Traverse the Canonical SDFG in topo order
- Add a node in the Canonical DAG for any LibNode in the SDFG
- When we are at the "boundaries" between two LibNodes, check if StreamingComposition can be applied (need the fully expanded SDFG)







Example – Application Space Exploration



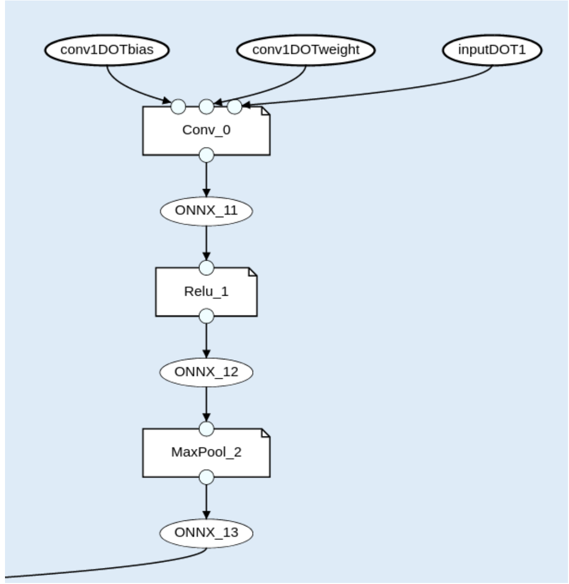
Given the application SDFG, we can enumerate all Canonical SDFG/DAGs that we can obtaining using the various available expansions.

(Then we will study more clever approaches)

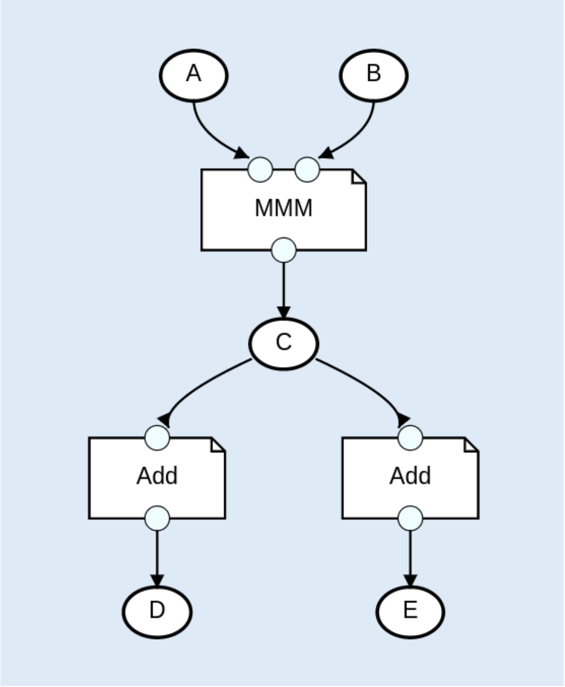
DAG #	Makespan
1	36
2	99
3	67
4	67
5	99
6	114
…	…

Scheduled with 8 PEs, 4x4 matrices

Discussion



This approach require us to have a collection of library node expansions that respects Canonical DAG rules. The same will be needed for dealing with ML applications (WIP): restricting to a limited set of supported operation can be helpful



Current Streamability analysis needs to be expanded

Discussion

This part needs more work, but it is important to start planning for the *Architecture Space Exploration*

- Varying number of PEs, can be done by scheduling with different parameters
- Varying the type of PEs (e.g. some operations are not supported or need different expansions)
- Vectorization, can be done by changing the DAG (leveraging again data-centric transformations)
- …

What about defining a full methodology?

Knowing what is the type of architectures to which you are interested in is important: for example

- a CGRA with a set of homogenous PEs and reconfigurable interconnection
- a SoC, with a set of heterogenous PEs
- …?