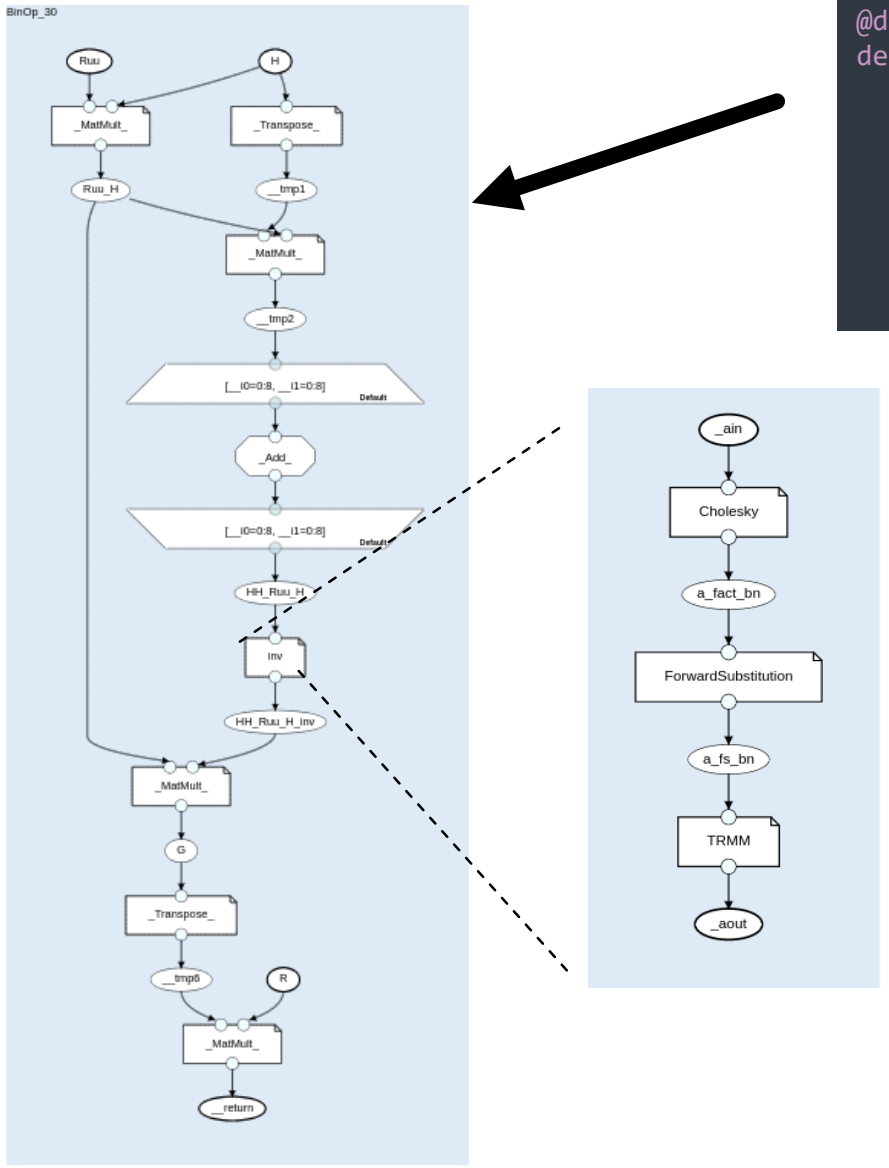
ASA: Enabling DSE

PUSCH-MIMO




```python
import numpy as np

@dace.program
def mimo(Ruu: dace.float32[64, 64], H: dace.float32[64, 8],
         R: dace.float32[64, 1]):
    Ruu_H = Ruu @ H
    HH_Ruu_H = np.transpose(H) @ Ruu_H + 1
    HH_Ruu_H_inv = np.linalg.inv(HH_Ruu_H)
    G = Ruu_H @ HH_Ruu_H_inv
    S = np.transpose(G) @ R
    return S
```

Right now these are place holders:

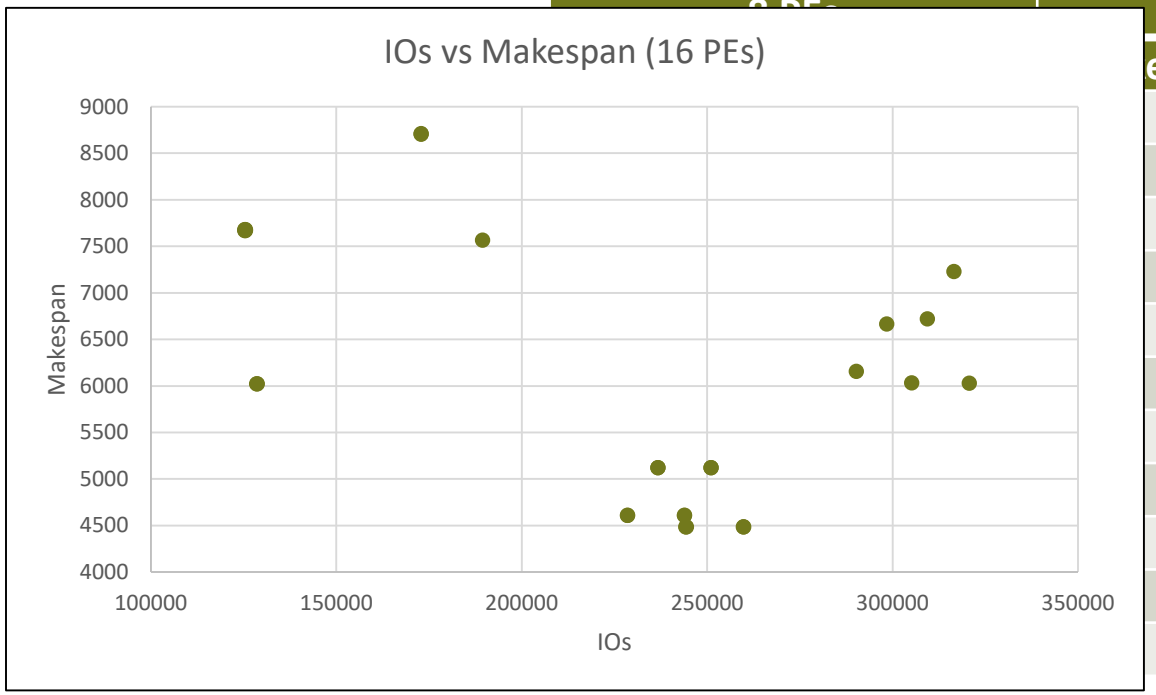
- They mimic computation time ($O(N^3)$)
- But they don't perform the actual computation

Thanks to Suleyman and Patroklos for the insight on the algorithm

Transpose (conj) is represented as an actual copy

PUSCH-MIMO

Performed Application Space Exploration + Scheduling. There are three MMM, this results in 27 different variations



16 PEs		32 PEs	
espan	#IOs	Makespan	#IOs
4486	259728	3334	259728
4486	259856	3334	259856
4487	244240	3335	227856
4487	244368	3335	227984
4609	243856	3585	235536
	...	...	
7671	125456	7177	151824
7680	125328	7187	120848
7680	125456	7296	151824
8704	172816	7745	113680
8713	172816	7784	113680

IOs counted by considering non-streaming edges and buffer nodes (we can decide how to count this)

Why no correlation?

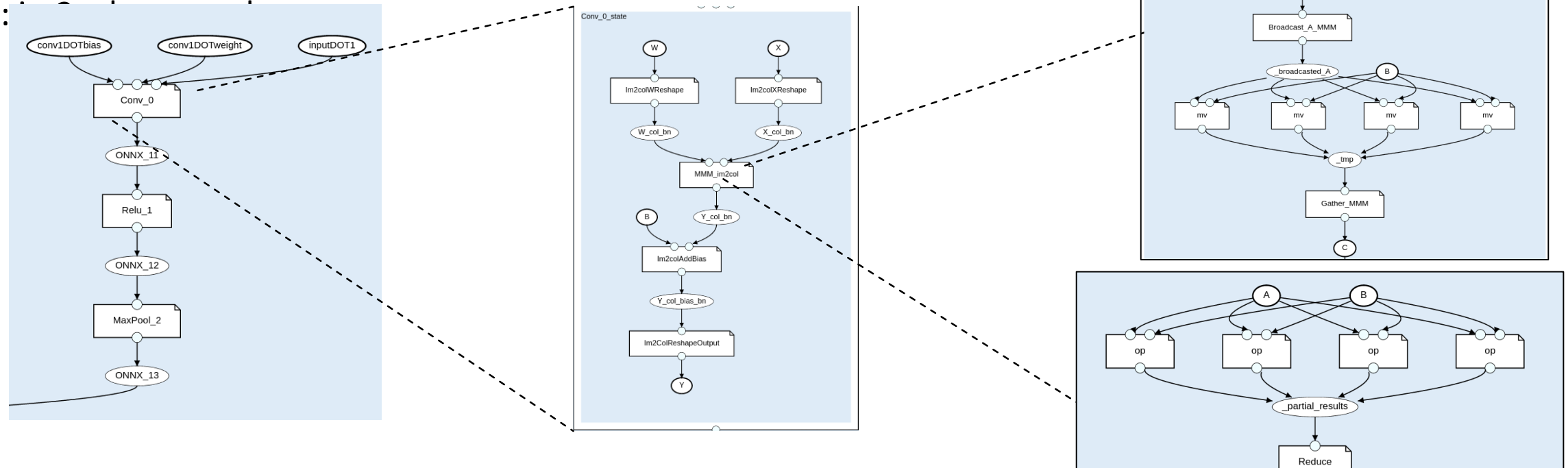
One of the reason is that certain variation allow more parallelism

(Needs to be investigated further)

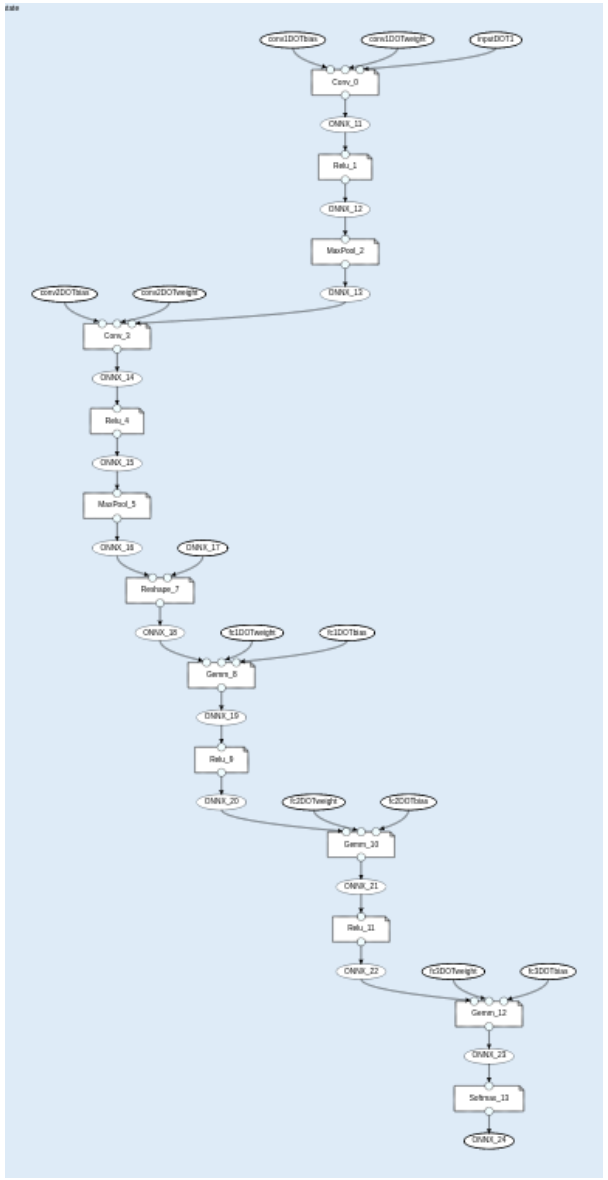
ML Workload

Implemented various canonical expansions for CNN operators

- For Elementwise operators (Relu, Add, Sub,) straightforward
- MaxPool, is represented as a downsampler
- Reshape, currently implemented as elemwise nodes (can be also interpreted as buffer node)
- Softmax: decomposed in multiple stages
- Gemm re-uses MatMul + Bia saddition
- Conv:

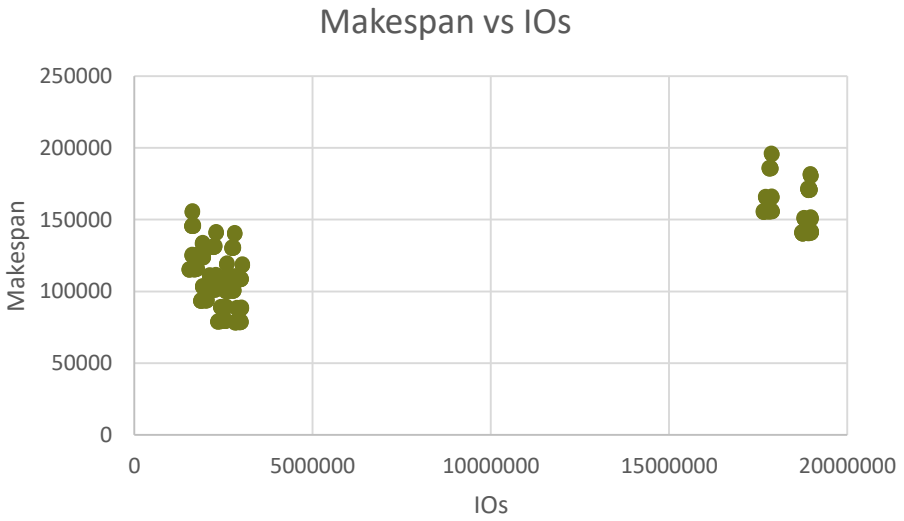


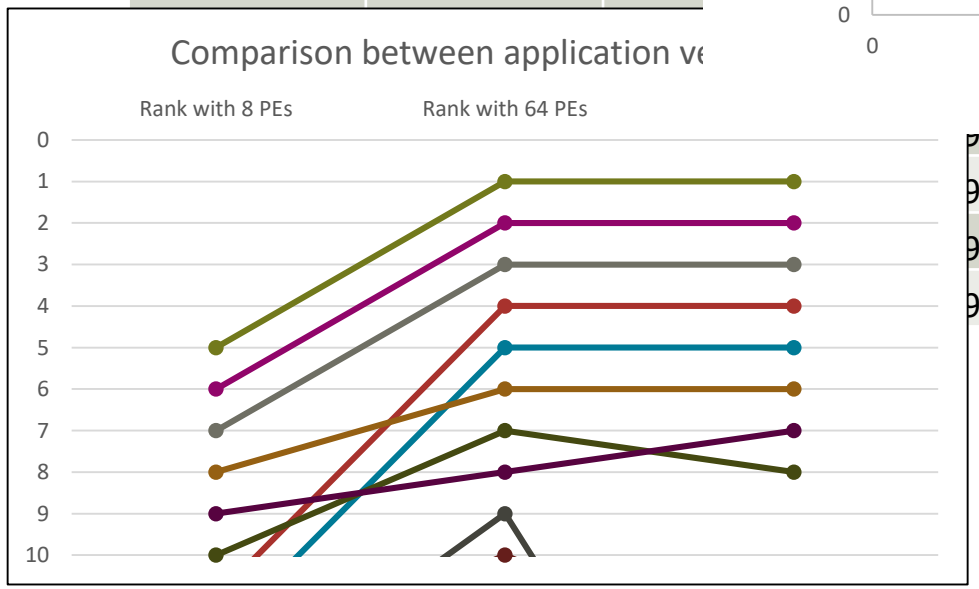
Lenet Example



244 different application representations

8 PEs		
Makespan	# IOs	Mak
145993	3440848	
146159	3381488	
146278	3394352	
146278	3456544	
150515	3462080	





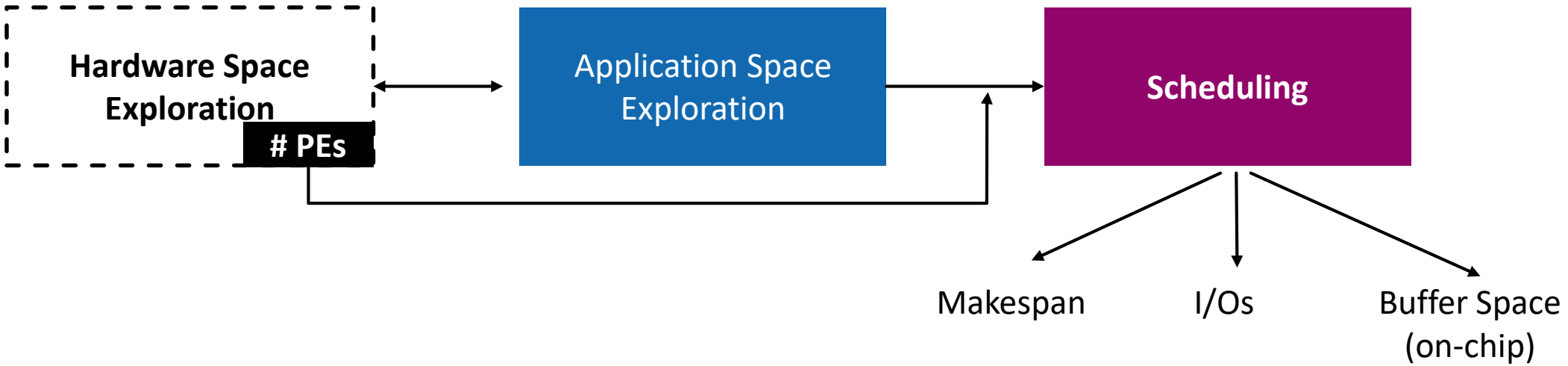
989572	180431	17848104
966052	195506	17881428
971572	195523	17884492
955888	196343	17874288

The best solution for a given #PEs, may be not as good for a different # PEs

Next steps

- Task granularity: implement new approach
- MIMO: implement inversion/have a more accurate analysis
- MIMO: consider the full amount of received signals (at least two approaches)
- Understand the results
- Start adding “intelligence” to Application Space Exploration

Space Exploration



- ASE can still be expanded, by considering other operation implementations
- Some pruning can be done before the scheduling (by looking at the streaming depth)
- ASE space is already large on its own and generating the Canonical DAG may require some time (complete data analysis on large graphs). We need to prune its space … ideally we would like to stay on the Pareto Frontier, but how do we understand it?