YAKUP KORAY BUDANAZ

SoftHier July 14

Overiew of the Topics:

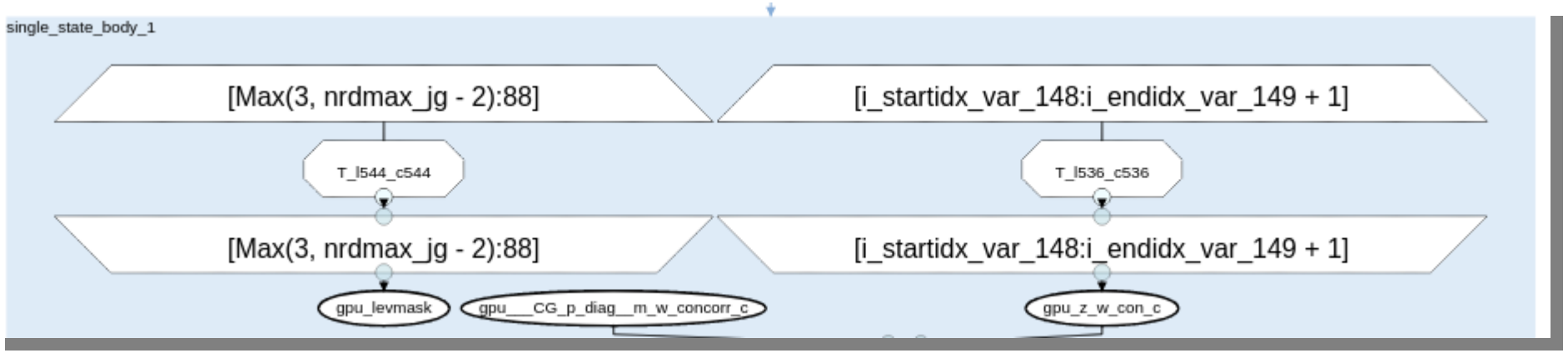
- **New GPU Transformations:**
  - *Copy / Assignment Kernel to Memcpy / Memset*
  - *Move If Inside Kernel*
  - *Eager Transformation: Array-Value-to-Constant Replacement*
  - *Eager Transformation: Bitwidth Lowering Transformations*
  - *Manual Profiling SDFG Generation*
- **GPU Codegen Re-design:** *Stream Management in DaCe*
- **New Student Projects:** *SoftHier Backend & BlockedFP Formats in DaCe*

- Still no response from the legal team.

- **New GPU Transformations:** *Copy / Memset Map to Tasklet*

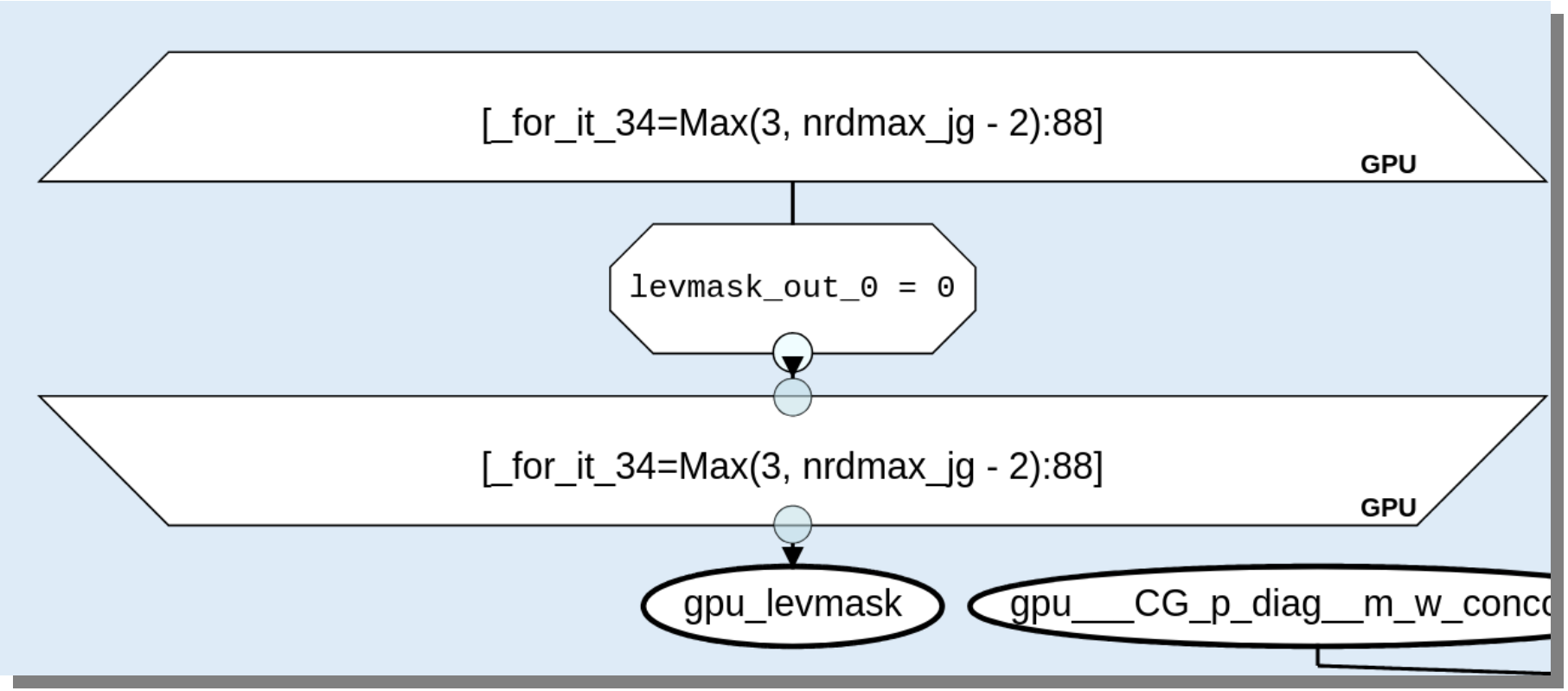
Copy/Memset Map To Copy/Memset Tasklet

DaCe's Frontend usually generate SDFGs where certain element-wise operations are generated as maps but could be exchanged with optimized implementations. For example here:

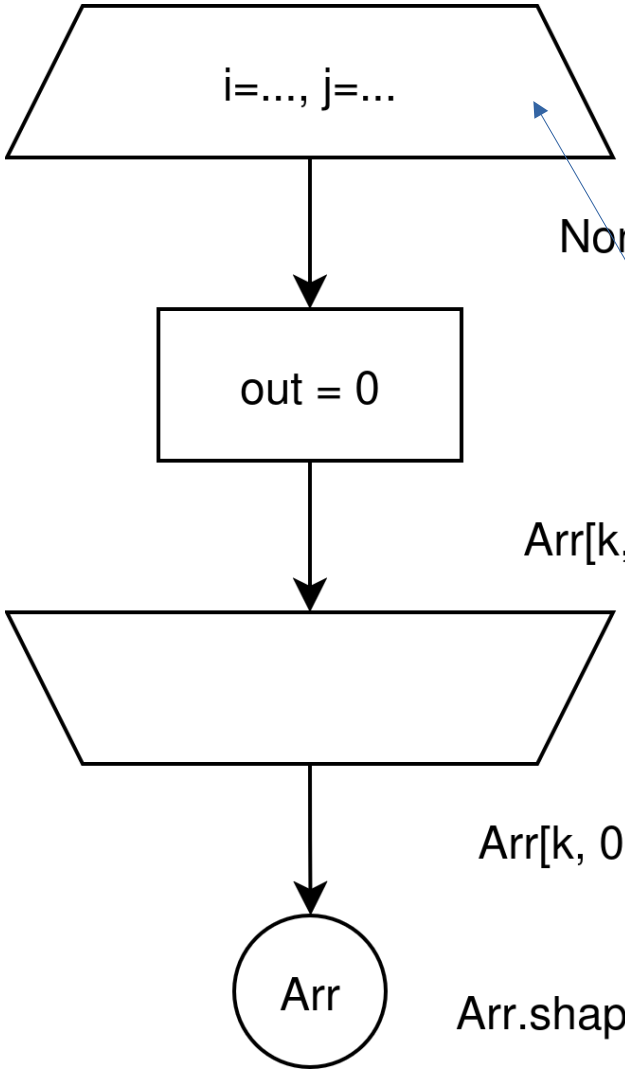


Copy/Memset Map To Copy/Memset Tasklet

This map only assigns 0 to the output array:



Copy/Memset Map To Copy/Memset Tasklet



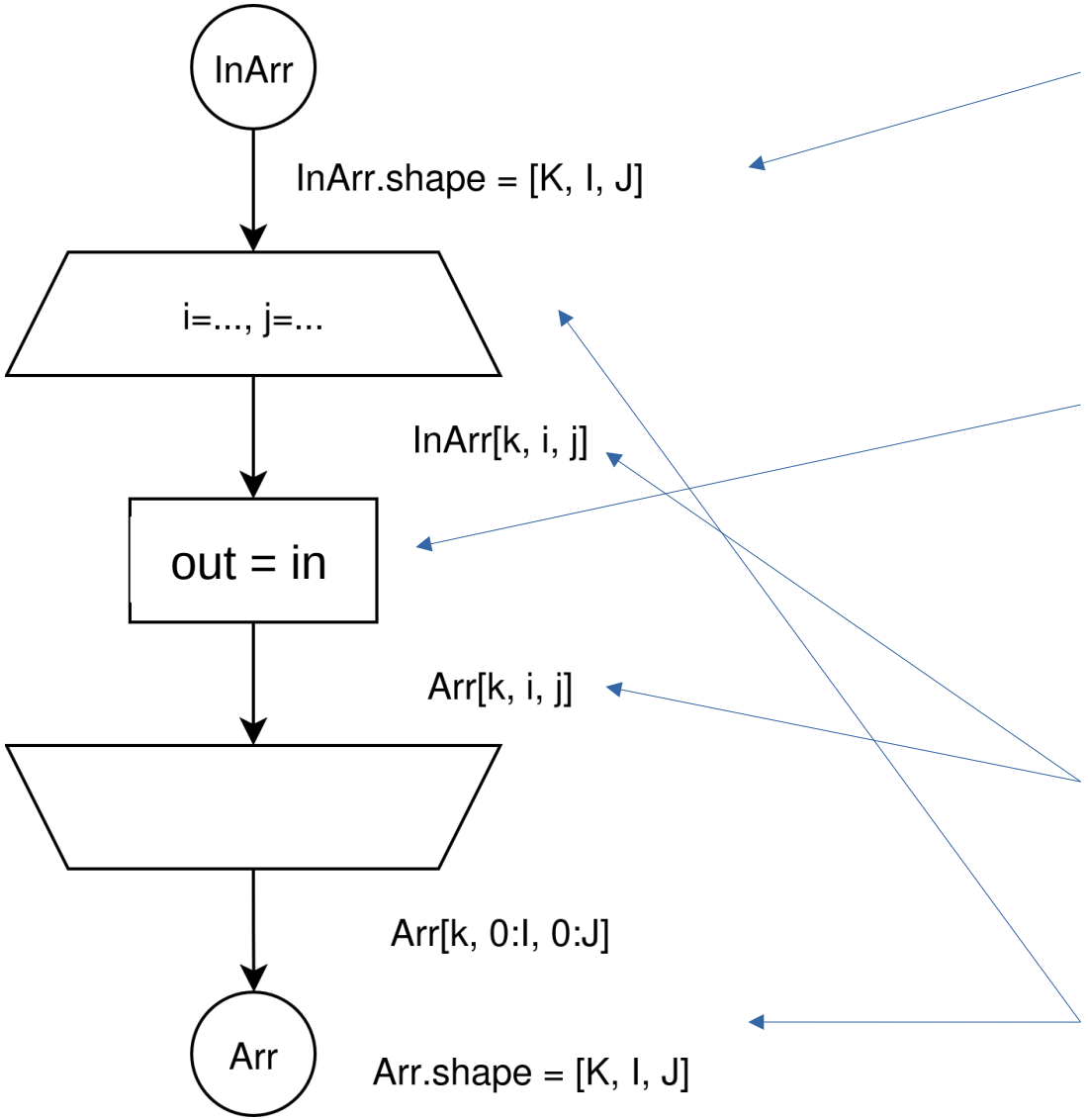
If exists path from map entry to map exit

If tasklet is a 0 assignment (transformation can be improved to support different repeating int values)

If subset within the map results in a contiguous access*

If map schedule and data types are supported

Copy/Memset Map To Copy/Memset Tasklet



If exists path from map entry to map exit

If tasklet is a simple copy without type casts.

If subset within the map results in a contiguous access for both input and output*

If map schedule and data types are supported

Copy/Memset Map To Copy/Memset Tasklet

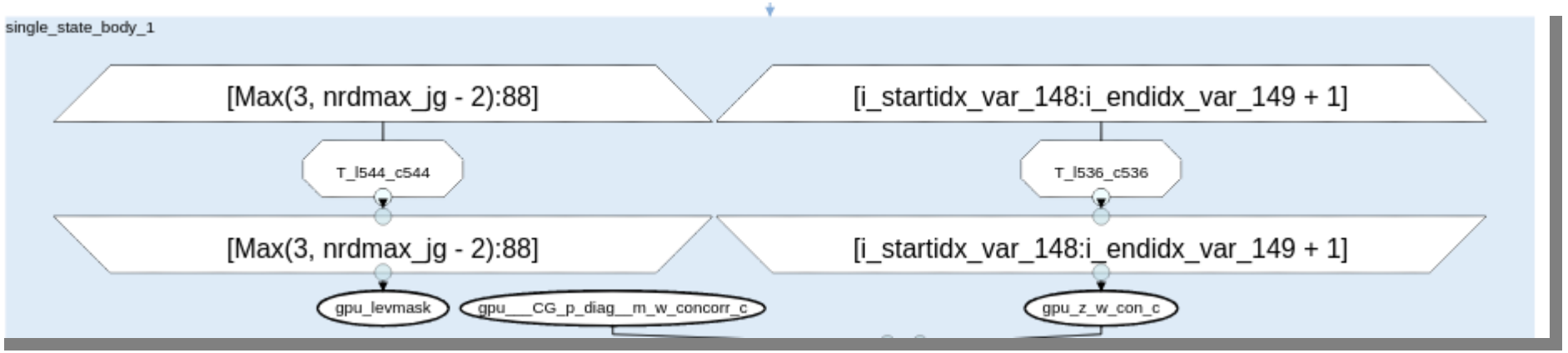
If subset within the map results in a contiguous access for both input and output*

There is no existing utility to detect if an array is packed an in column-major / row-major format.

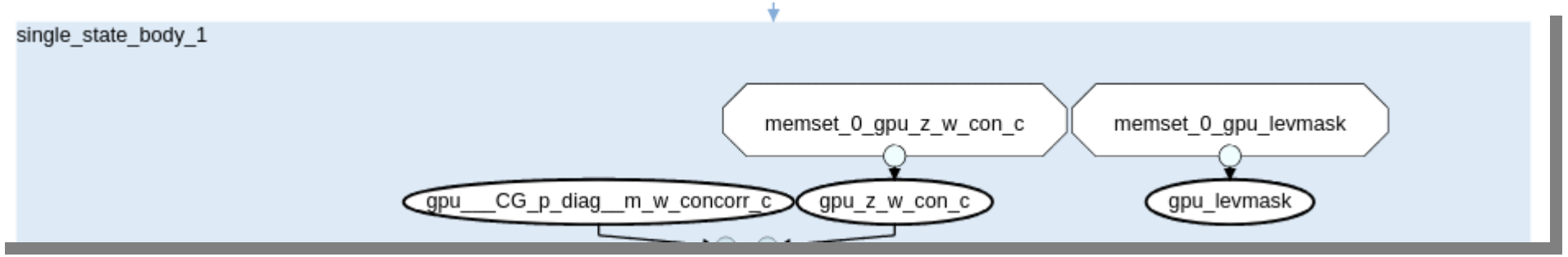
There is also no existing utility to check if a subset contiguous.

I am working on PR to merge the extensions and the pass to upstream DaCe

Copy/Memset Map To Copy/Memset Tasklet

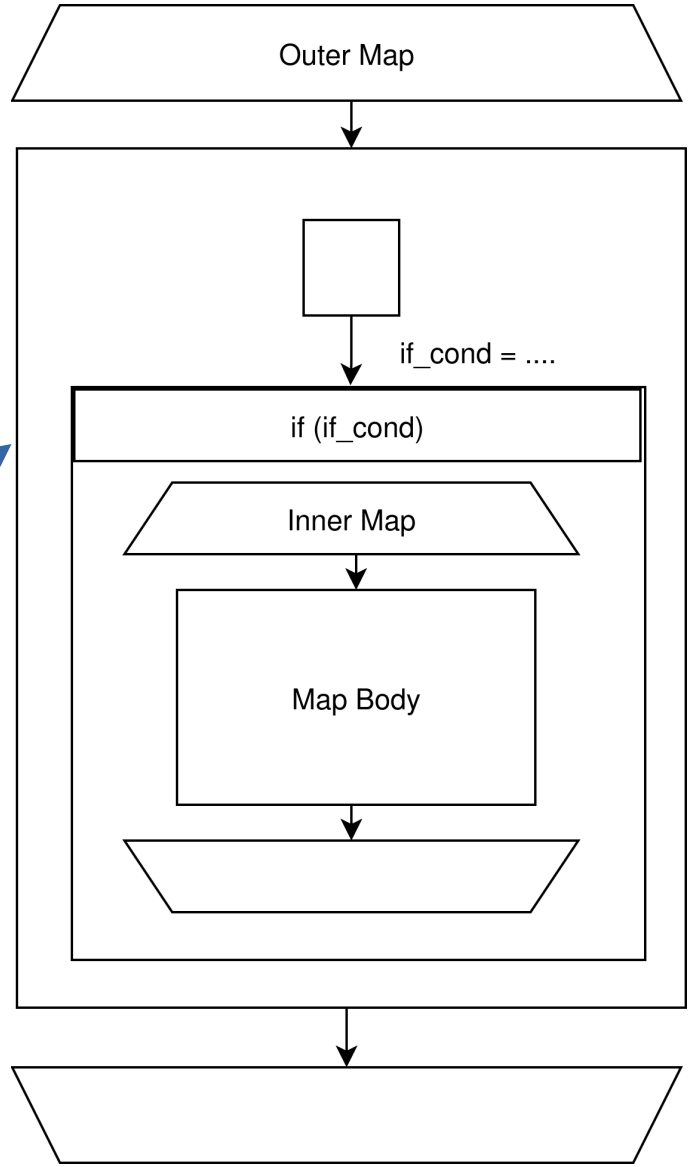



```
AssignmentAndCopyKernelToMemcpyAndMemset().apply_pass(sdfg)
```



- **New GPU Transformations:** *Move If Inside Map*

Move If Inside Map

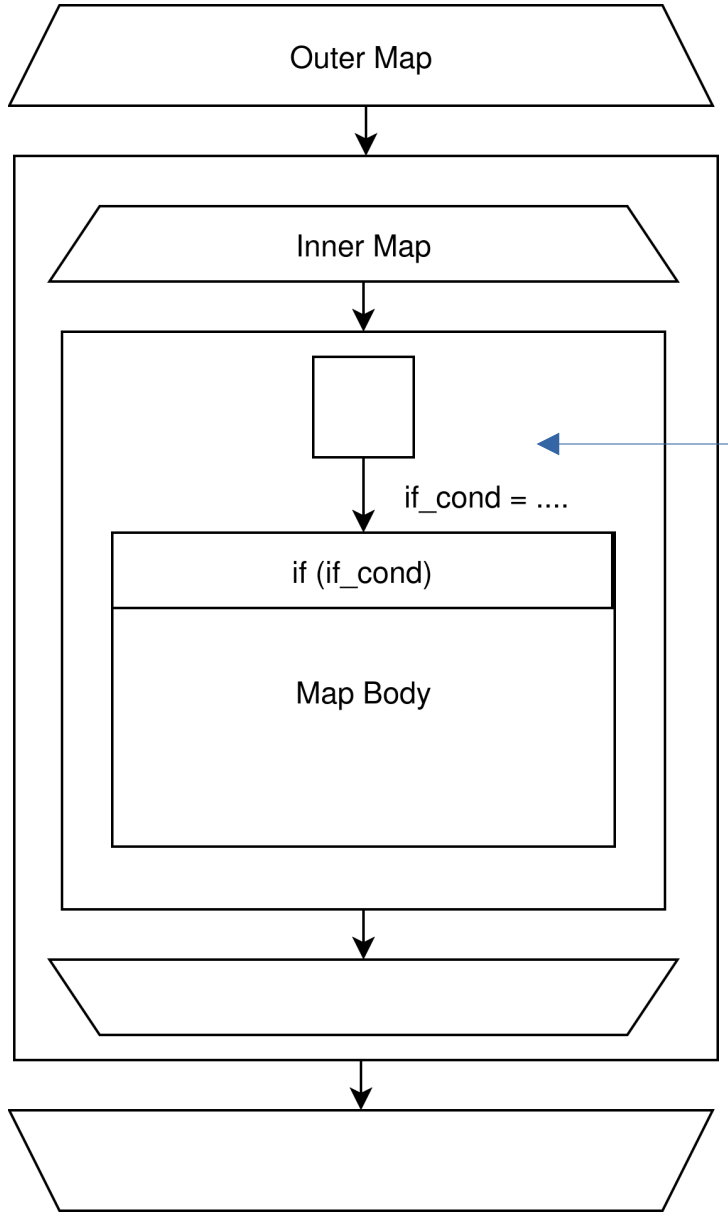


Outer Map can be offloaded to the accelerators

Inner computation depends on a conditions. Inner body has to be a sequential loop executed by a thread (only option due to how DaCe handles offloading).

Move If Inside Map

The outer and the inner map can be collapsed together now*



Every thread duplicates the computation of the if condition

*not always necessarily

Move If Inside Map

The PR is in Merge Queue

- **New GPU Transformations:** *Eager Transformation: Array-Value-to-Constant Replacement*

Array-Value-to-Constant

```
1  program = SDFG(...)
2
3  # Old Program
4  program(...)
```

Instrument the program to check the values of candidate arrays – and create two specialized programs.

```
1  # The new program
2  all_candidates_are_constant = all(is_constant(candidate_array) for candidate_array in candidate_constant_arrays)
3  if all_candidates_are_constant:
4      specialized_program = SDFG(...).replace({candidate_array: value(candidate_array) for candidate_array in candidate_constant_arrays})
5      specialized_program(...)
6  else:
7      program(...)
```

Array-Value-to-Constant

```python
# The new program
all_candidates_are_constant = all(is_constant(candidate_array) for candidate_array in candidate_constant_arrays)
if all_candidates_are_constant:
    specialized_program = SDFG(...).replace({candidate_array: value(candidate_array) for candidate_array in candidate_constant_arrays})
    specialized_program(...)
else:
    program(...)
```

When the SDFG is called for the first time, the check is performed.

The transformation currently assumes these arrays are constant across program invocations.

- **New GPU Transformations:** *Eager Transformation: Bitwidth Lowering Transformations*

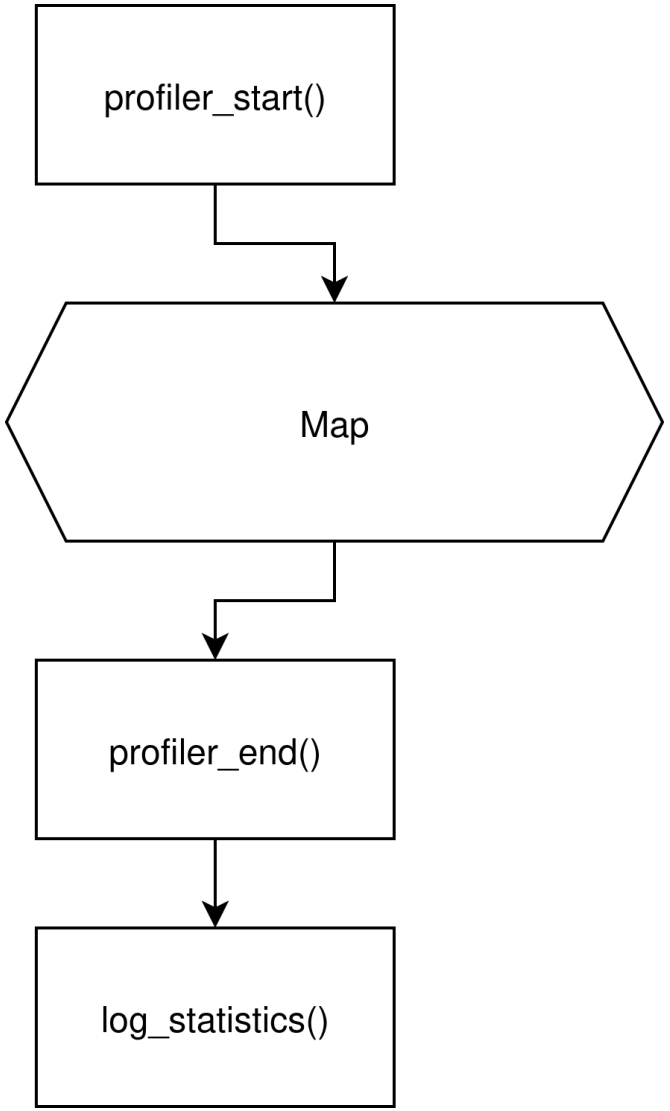
Bitwidth Lowering Transformation

```
# The new program
all_candidates_fit_x_bit = all(required_bits(candidate_array) < x for candidate_array in candidate_arrays)
if all_candidates_fit_x_bit:
    specialized_program = SDFG(...).replace({candidate_array.dtype: x for candidate_array in candidate_arrays})
    specialized_program(...)
else:
    program(...)

```

$

Manual Profiling SDFG Generation

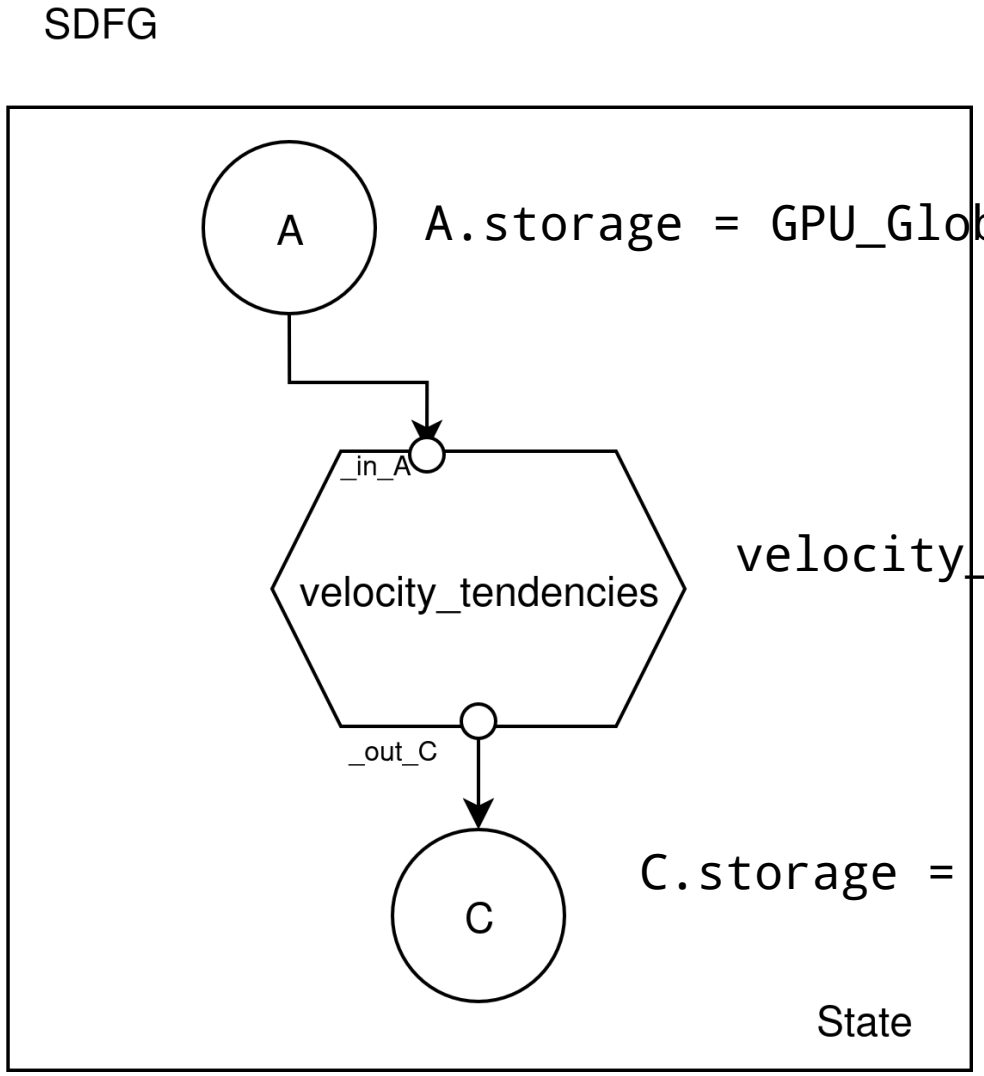


Do this on every map schedule type that one wants to analyze.

The pass triggered 3 bugs in the GPU codegen.

- **GPU Codegen Re-design:** *Stream Management in DaCe*

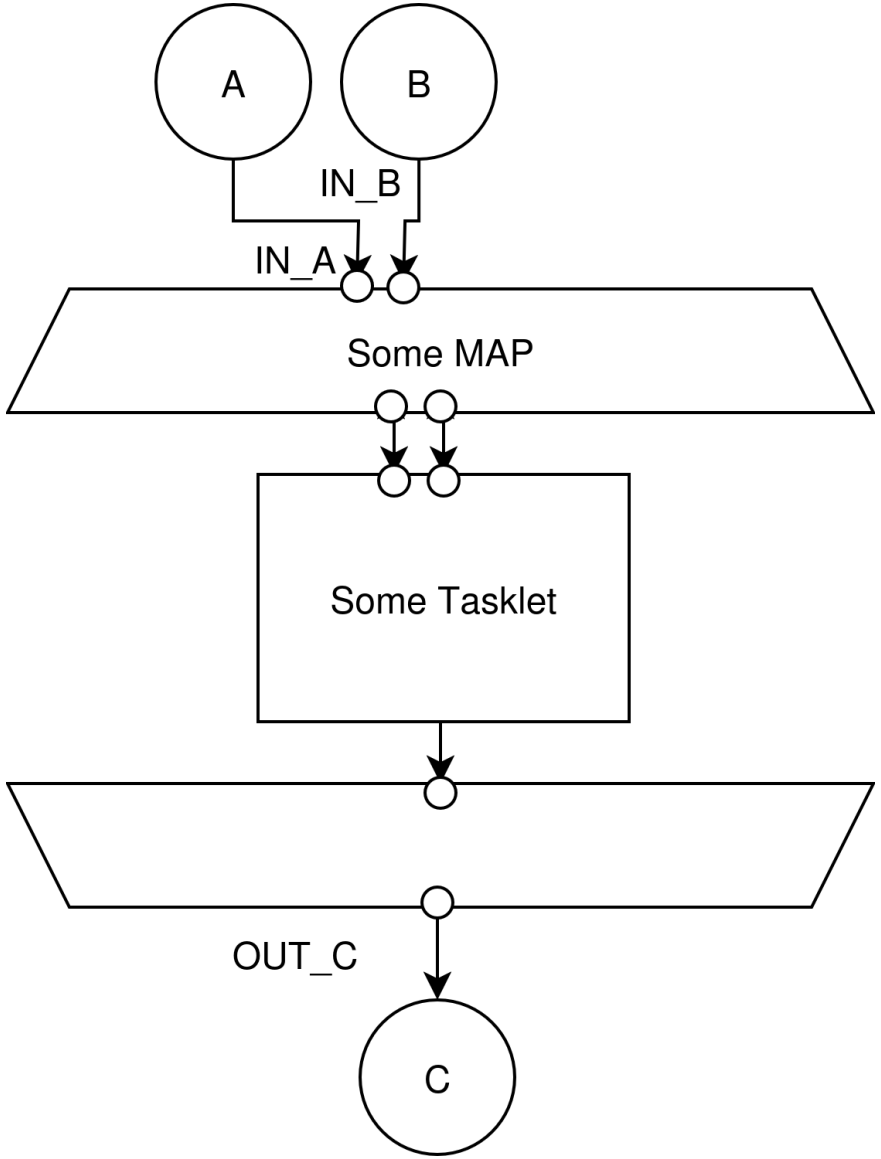
CUDA Streams and DaCe



This tasklet does not have a parent `map` scope or a parent `LoopRegion`.

SDFG passes validation, it generates valid SDFG and compiles but **HOW?** On which stream this kernel runs?

CUDA Streams and DaCe



How to choose the stream for this map? By assigning the private `_cuda_stream` field of a map node.

```
Some Map.node._cuda_stream = 0 # (int)
```

Or

```
Some Map.node._cuda_stream = “nullptr” # (str)
```

Backend checks map entries with a lot of `hasattr`, `setattr` usage.

CUDA Streams and DaCe

How to handle stream management of maps in DaCe?

1. `gpu_stream` field for a `MapNode`.

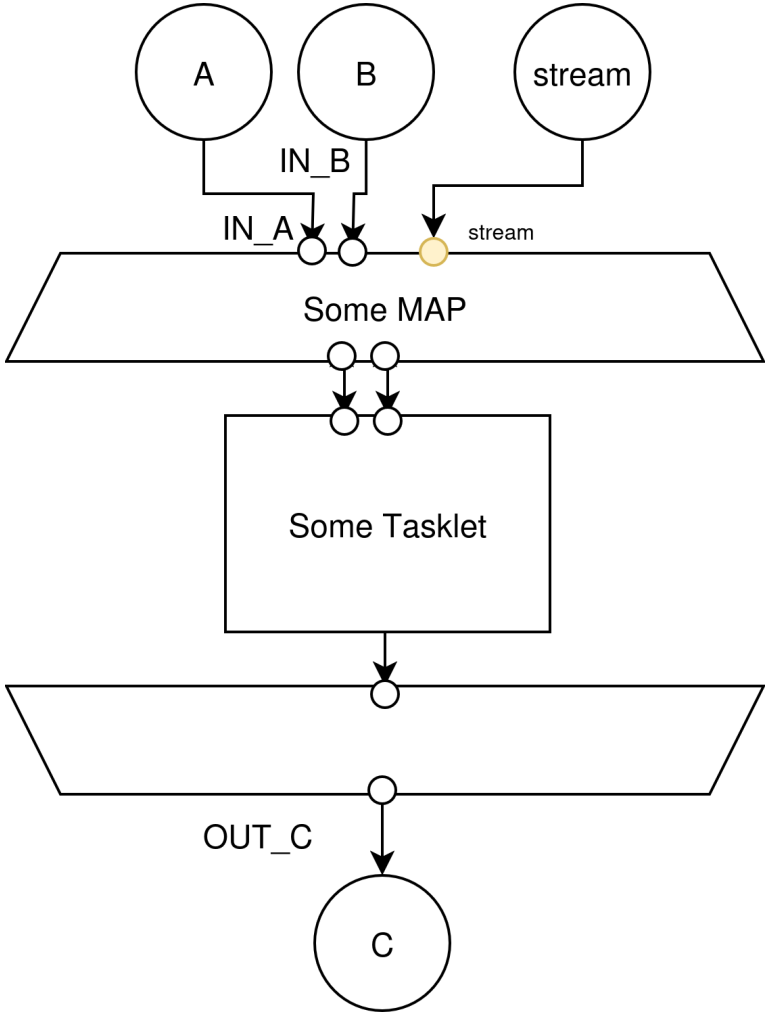
   - Default value `None` maps to `nullptr`. A schedule pass can assign a `symexpr`.
   - Synchronization is done by a tasklet which reads the stream id from the map.

2. A stream object.

   - I will show the details in the upcoming slides.

CUDA Streams and DaCe

How to handle stream management of maps in DaCe? Through A stream object?



A stream object (type: `dace_stream_t` which is lowered to `cudaStream_t` for CUDA)

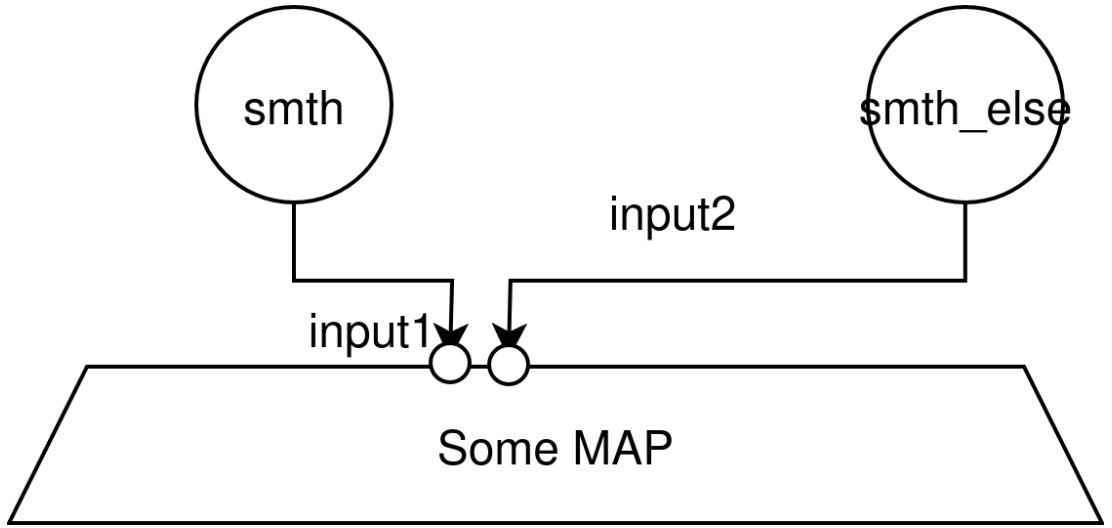
or

A stream id (type: `int`)

CUDA Streams and DaCe

How to handle stream management of maps in DaCe? Through A stream object?

Consider this map, it has 2 dynamic inputs:



CUDA Streams and DaCe

How to handle stream management of maps in DaCe? Through A stream object?

DaCe will generate code like this:
It is possible to implement using a dynamic in connector if we ensure that stream object always returns an integer id, or `stream_t` type.

In both cases, codegen needs to treat stream in connector differently than other dynamic in connectors.

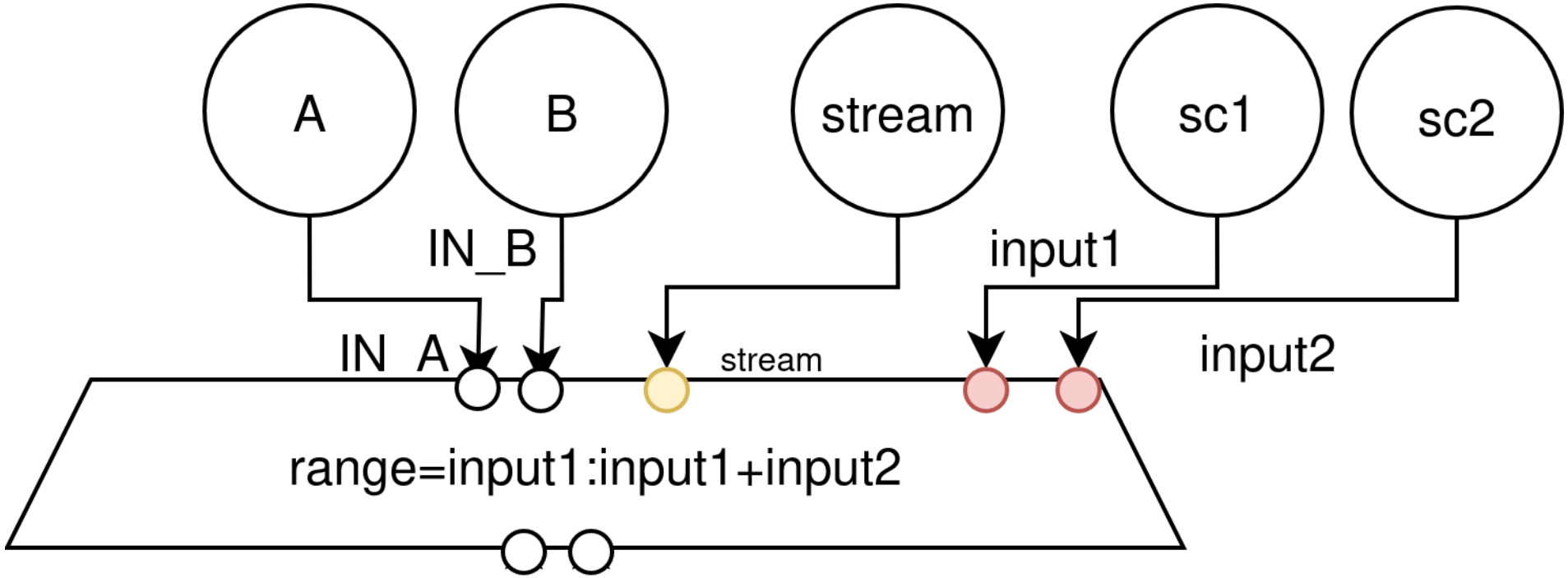
```
DACE_EXPORTED void __dace_runkernel_single_state_body_map_12_0_1(solve_nh_predictor_pre_state_t *__state, ...)
{
    // Dynamic Map Inputs
    int input1 = smth;
    int input2 = smth_else;

    if (empty_grid) {
        return;
    }

    void  *args[] = { (void *)&... };
    gpuError_t __err = cudaLaunchKernel((void*)kernel, grid_dim, block_dim, shr_mem, stream_id);
    DACE_KERNEL_LAUNCH_CHECK(__err, meta_information);
}
```

CUDA Streams and DaCe

How to handle stream management of maps in DaCe? Through A stream object?



CUDA Streams and DaCe

How to handle stream management of maps in DaCe? Through A stream object?

If we use `stream_t` then we to pass the stream to kernel launch.

```
DACE_EXPORTED void __dace_runkernel_single_state_body_map_12_0_1(solve_nh_predictor_pre_state_t *__state, ...)
{
    // Dynamic Map Inputs
    int input1 = smth;
    int input2 = smth_else;
    stream_t stream = stream_smth;

    if (empty_grid) {
        return;
    }

    void  *args[] = { (void *)&... };
    gpuError_t __err = cudaLaunchKernel((void*)kernel,
     gridDim(ceil((input2 - input1) / block_dim)),
     block_dim, shr_mem, stream_id);
    DACE_KERNEL_LAUNCH_CHECK(__err, meta_information);
}
```

CUDA Streams and DaCe

How to handle stream management of maps in DaCe?

1. `gpu_stream` field for a `MapNode`.

   - Default value `None` maps to `nullptr`. A schedule pass can assign a `symexpr`.
   - Synchronization is done by a tasklet which reads the stream id from the map.

2. A stream object.

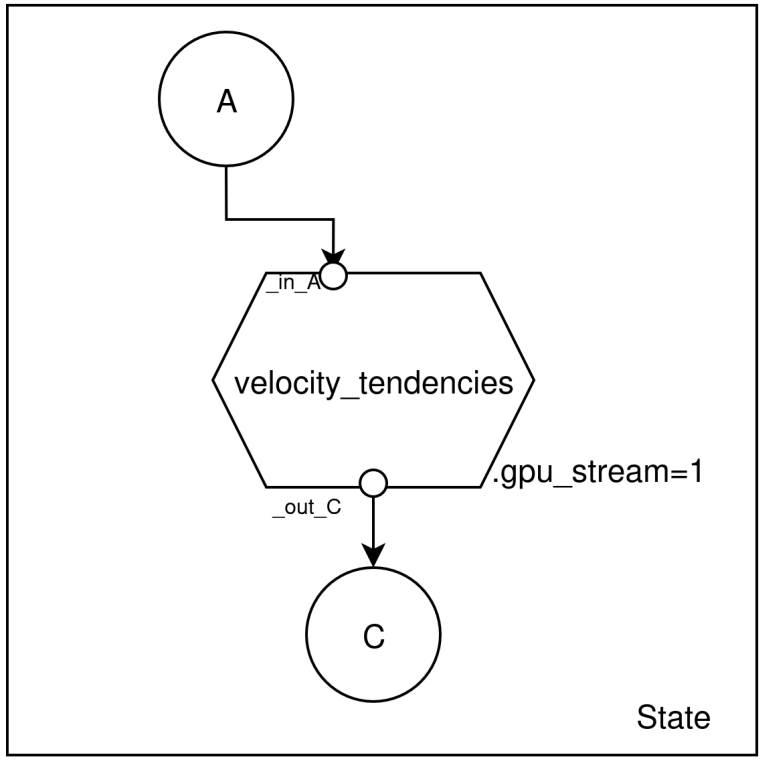
   - A stream id needs to be always passed to GPU maps and GPU Device scheduled tasklets. A schedule pass can be a `symexpr` if we support an array of streams.
   - Synchronization is done by a tasklet which takes the same stream as input and output.
   - A pass to specialize stream usage for accelerator based programming is necessary.

In both cases codegen needs to treat streams specially (whether in connector or field)

2. Aligns better with the design goal of making everything as explicit as possible

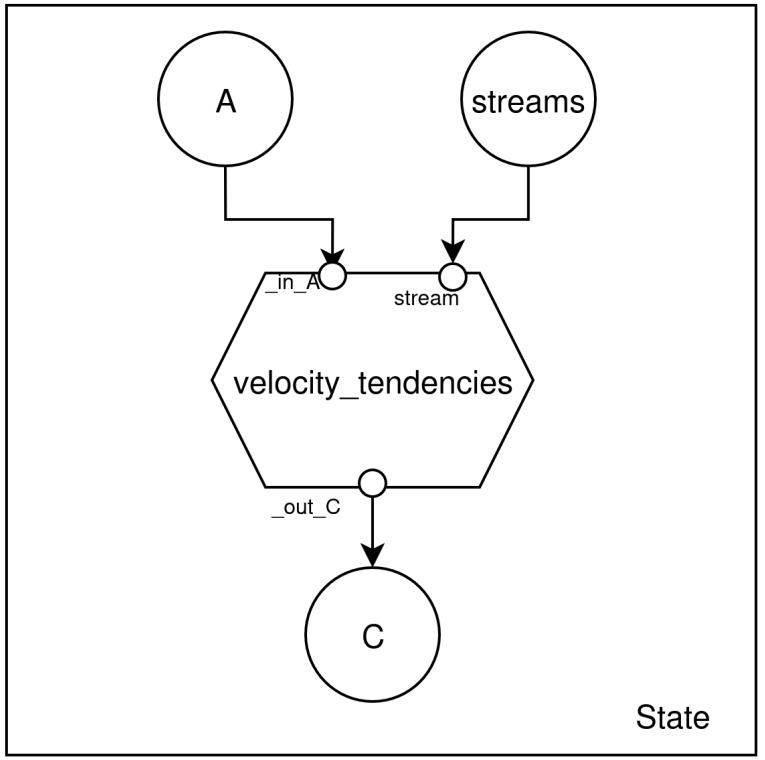
CUDA Streams and DaCe

SDFG



Option 1

SDFG



Option 2

- **Explicit Streams –** *Command Queues are part of the dataflow!*

- New Student Projects:
  - ***SoftHier Backend:***
    https://docs.google.com/document/d/1CfbUqIIaa4hYTY18cg64c-as2V6oiST5xcUUDqeP-gs/edit?usp=sharing
  - ***BlockedFP Formats:***
    https://docs.google.com/document/d/1qHMWNfJbAV8dLVXLs447M5Fompp2Bqxo9oJ048GNnHg/edit?usp=sharing
  - ***Modernizing NPBench:***
    https://docs.google.com/document/d/1sao_2bsDHQtiuMcJWTAd6Sf9MMWVw6R6Q0PPutM5DmI/edit?usp=sharing