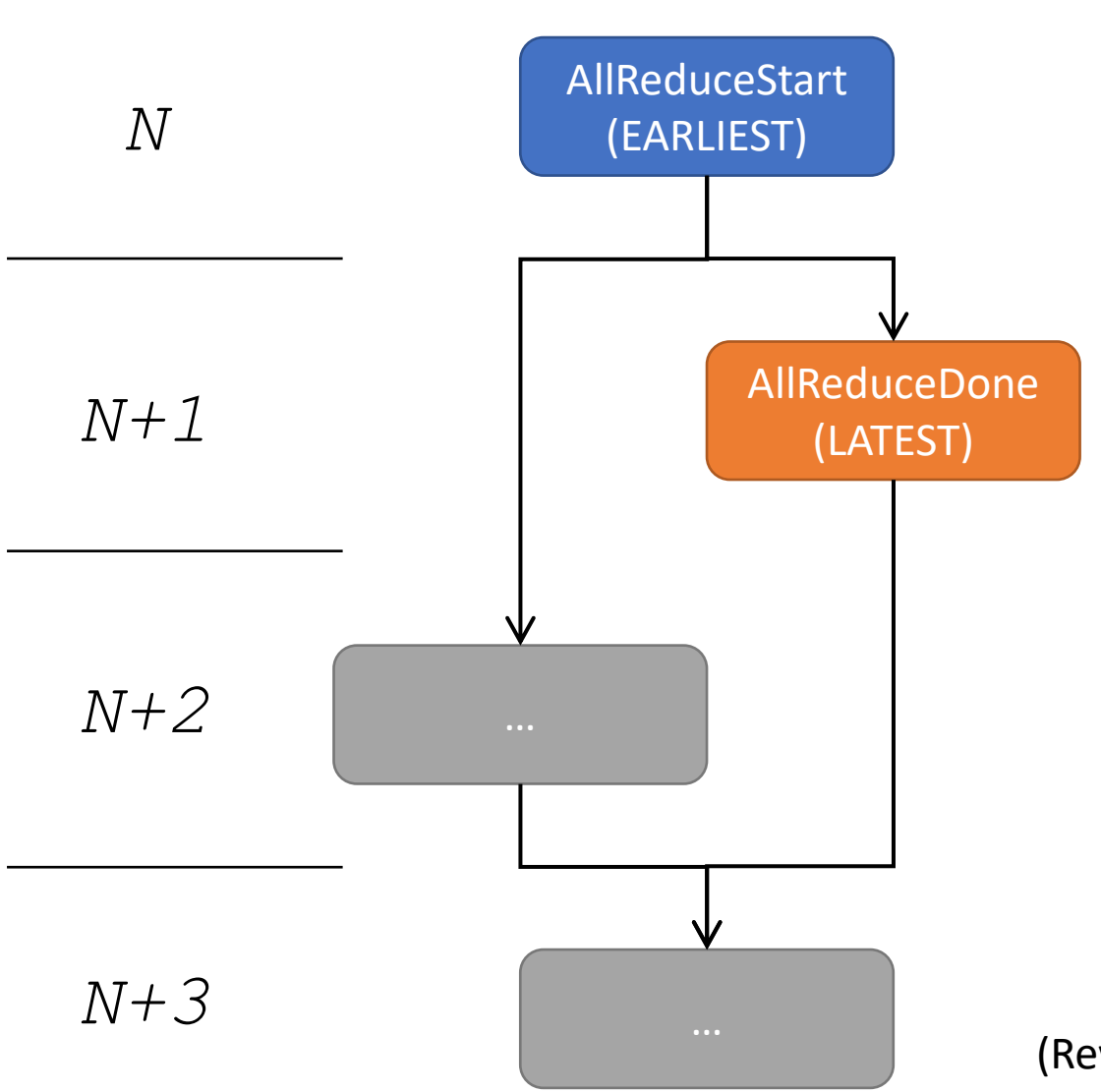
XLA Memory Scheduler Background

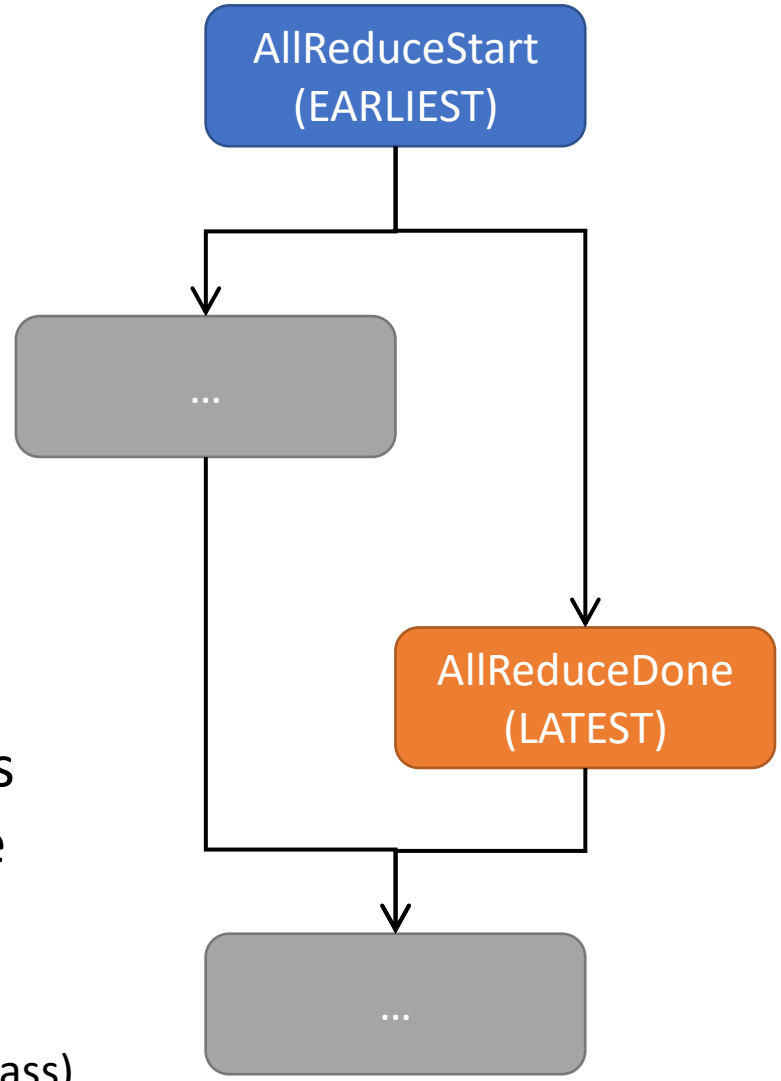
- Memory Scheduler `tensorflow/compilers/xla/service/hlo_memory_scheduler.cc`
  - Generates 3 different schedules (`dfs`, `list`, `postorder`)
  - Chooses schedule with the smallest (approximate) memory consumption
- Postprocessor (GPU only) `tensorflow/compilers/xla/service/gpu/gpu_hlo_schedule.cc`
  - Reorders instruction sequence
  - 3 categories of instructions:
    - `EARLIEST` (e.g. `AllReduceStart` or `CustomCall` with the `EARLIEST` attribute)
    - "Normal"
    - `LATEST` (e.g. `AllReduceDone` or `CustomCall` with the `LATEST` attribute)

Existing Scheduler Postprocessor

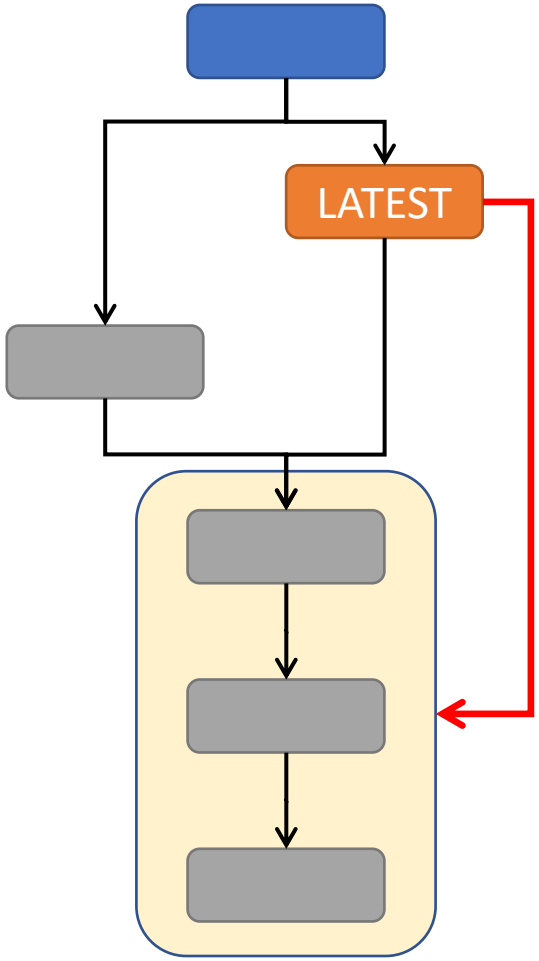


Move `LATEST` instructions as close as possible to immediate successor

(Reverse direction for EARLIEST pass)



Alternate Scheduler Postprocessor



"cluster" successor instructions together and schedule these as a group

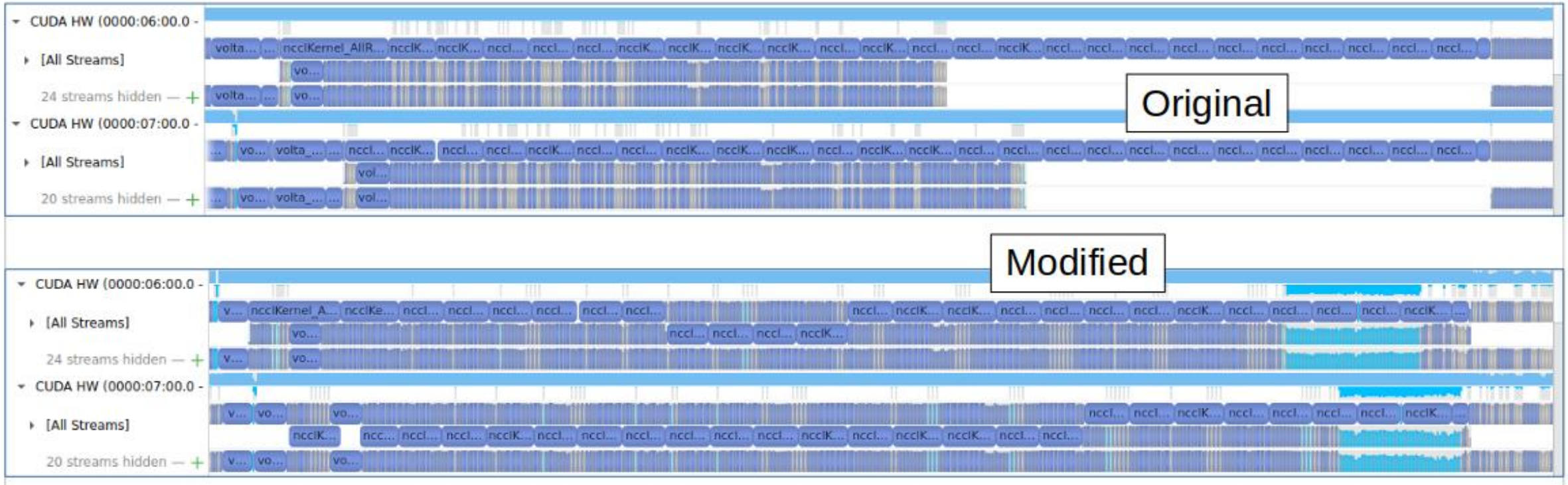
- Moves other instructions (not just `LATEST`), but retains some of the relative ordering of the original schedule
- Increases the distance between the EARLIEST and `LATEST` instructions by (potentially) using instructions from other `EARLIEST`/`LATEST` pairs

Example results for a JAX T5X network:

	Original Postprocessor			Modified		
	start	done	distance	start	done	distance
`all-reduce`	1930	1965	35	1838	2815	977
`all-reduce.1`	1958	2023	65	1949	4804	2855
`all-reduce.2`	1959	1960	1	1959	2802	843
`all-reduce.3`	6043	6044	1	1963	6064	4101

Preliminary Results

Exposed communication with the original scheduler postprocessor



Enabling the alternate scheduler in XLA

- Option 1
  - Flag for opt-in to alternate scheduler:
    - `bool xla_gpu_enable_alt_sched_postproc`
  - Allows experimentation and comparison with existing scheduler
  - Can eventually make the alternate scheduler the default, replacing the existing one
- Option 2
  - Replace the existing scheduling postprocessor