氣象署-興大應數聯隊
CWAGFS-TCo

Team Members: Jen-Her Chen[1], Pang-Yen Liu[1], Ting-An Chen[2], Lu-Hung Chen[2], Chun-Hao Teng[2]

Mentors: Leo Chen[3], Jay Chen[3]

[1] Central Weather Administration.
[2] Department of Applied Mathematics, NCHU.
[3] NVIDIA.

CWAGFS-TCo

Numerical weather prediction model

- Motivation: Accelerate computing
- Programming language: Fortran
- Parallel computing method: MPI
- Application module/function: AdvH (horizontal tracer advection)
  - Piecewise Parabolic Method (Irregular grid interpolation)
- Method: OpenACC and CUDA
- Goals: Porting AdvH to the GPU through OpenACC, and other modules still using MPI parallel computing.

Hardware and Software

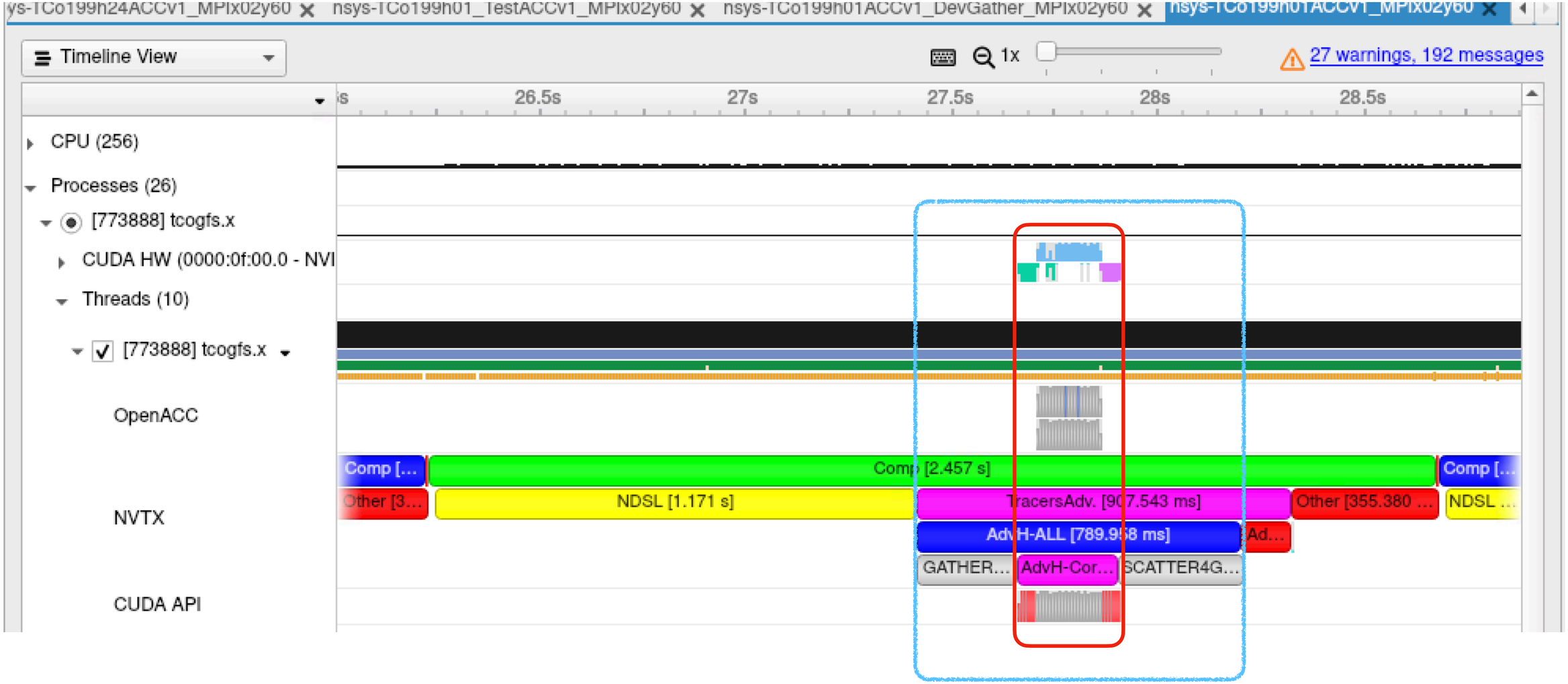
- CPU: Dual AMD EPYC 7742 64-Core (128 cores total)
- GPU: A100*8
- NVIDIA HPC SDK: v22.7
  - CUDA: v11.7
  - MPI: OpenMPI-3.1.5
- Use:
  - Processors: 40, 80, 120(cores)
  - GPU: 1(A100)

Version Description

- MPI
  - The original MPI parallel code
- GPU-MPI
  - Porting AdvH to the GPU through OpenACC.
  - Execute gather and scatter operations by using MPI on the CPU before and after AdvH calculations.
- GPU-CA
  - Based on GPU-MPI.
  - Execute gather and scatter operations before and after AdvH computations by using CUDA-Aware MPI on GPU.

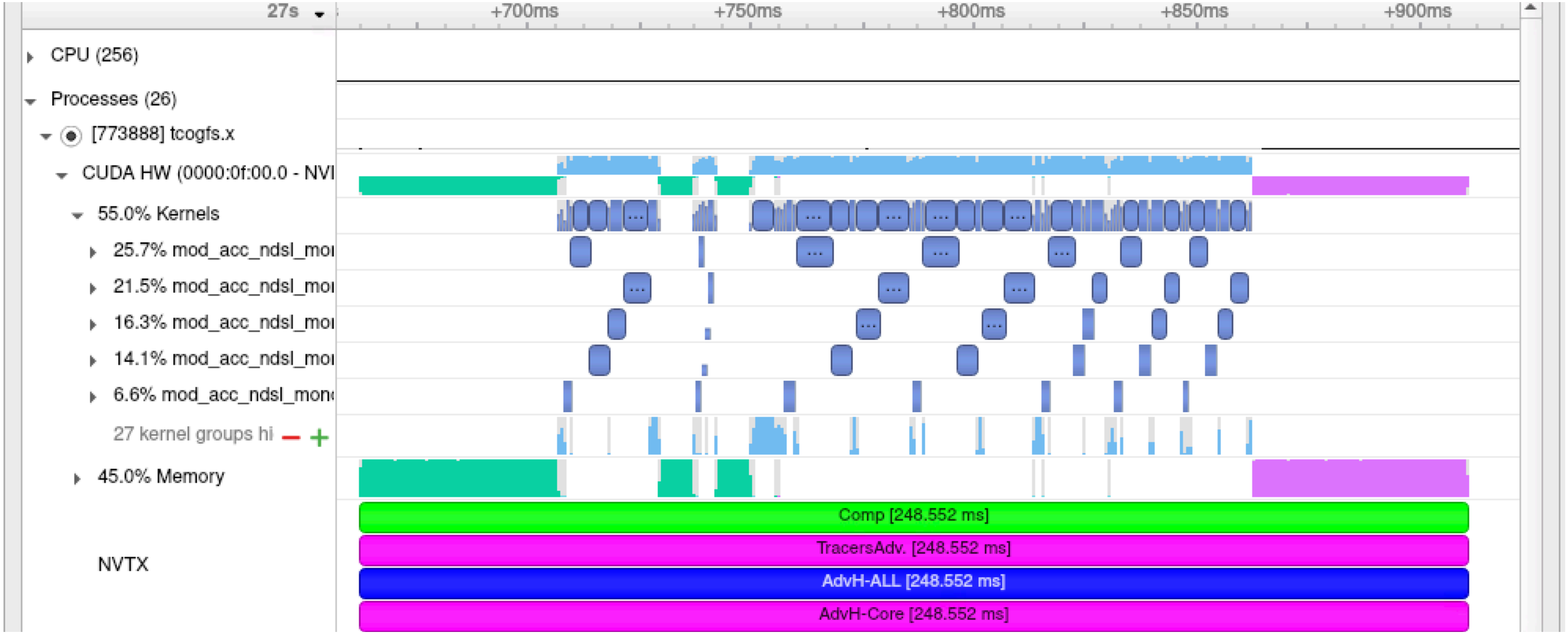
GPU-MPI

Porting AdvH to GPU.



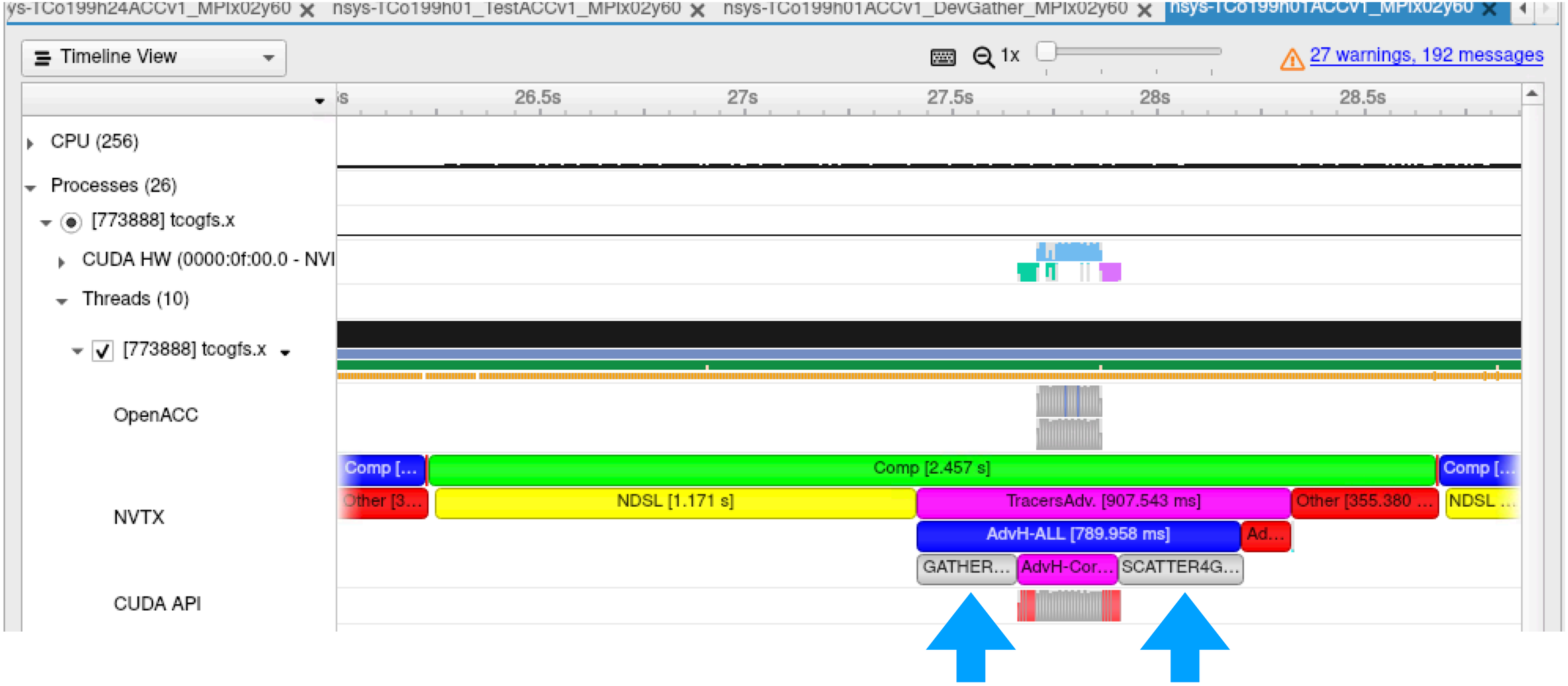
GPU-MPI

Place the OpenACC directives on the loops.



GPU-MPI

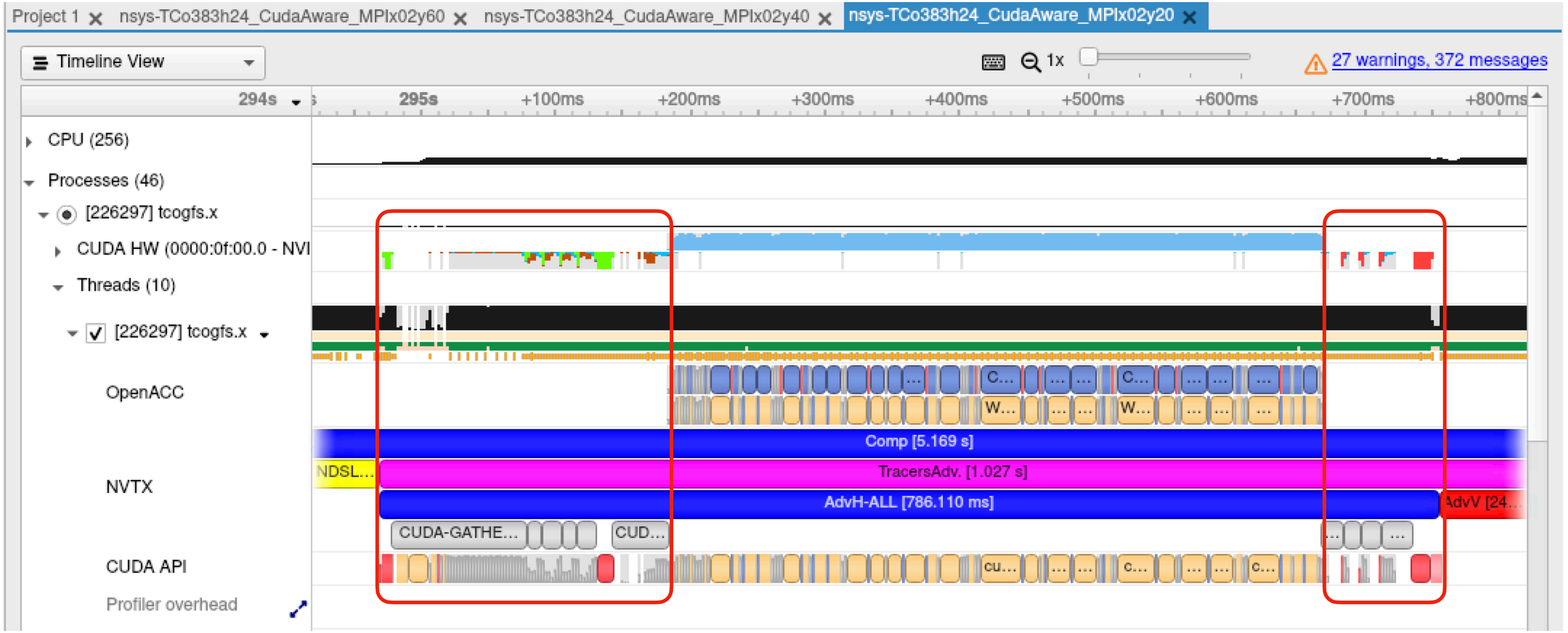
Spending too much time on gather and scatter operations.



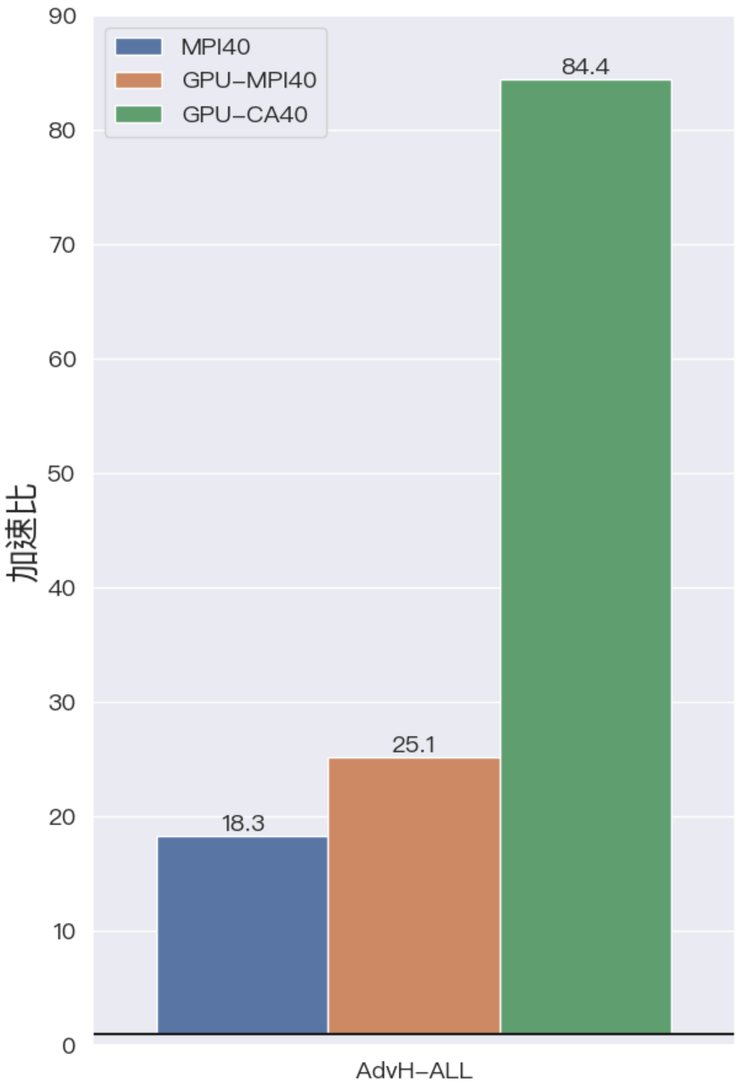
Gather and Scatter Operators

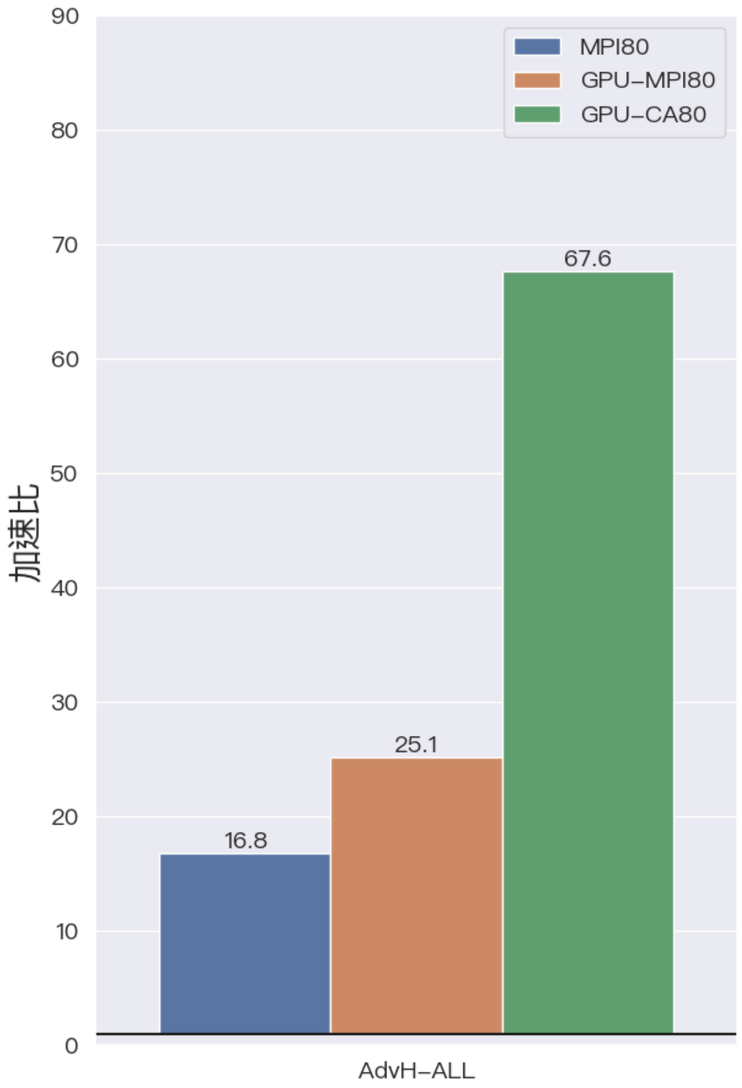
- GPU-MPI
  - Gather data from others CPU to rank0 CPU via MPI.
  - Transfer data from rank0 CPU to GPU.
  - AdvH calculation.
  - Transfer data from GPU to rank0 CPU.
  - Scatter data from rank0 CPU to others CPU via MPI.
- GPU-CA
  - Each CPU core transfers data to the GPU on its own CPU.
  - Gather data from others GPU to the GPU on rank0 CPU via CUDA-Aware MPI.
  - AdvH calculation.
  - Scatter data the from GPU on rank0 CPU to others GPU via CUDA-Aware MPI.
  - Each CPU core gets data from the GPU on its own CPU.

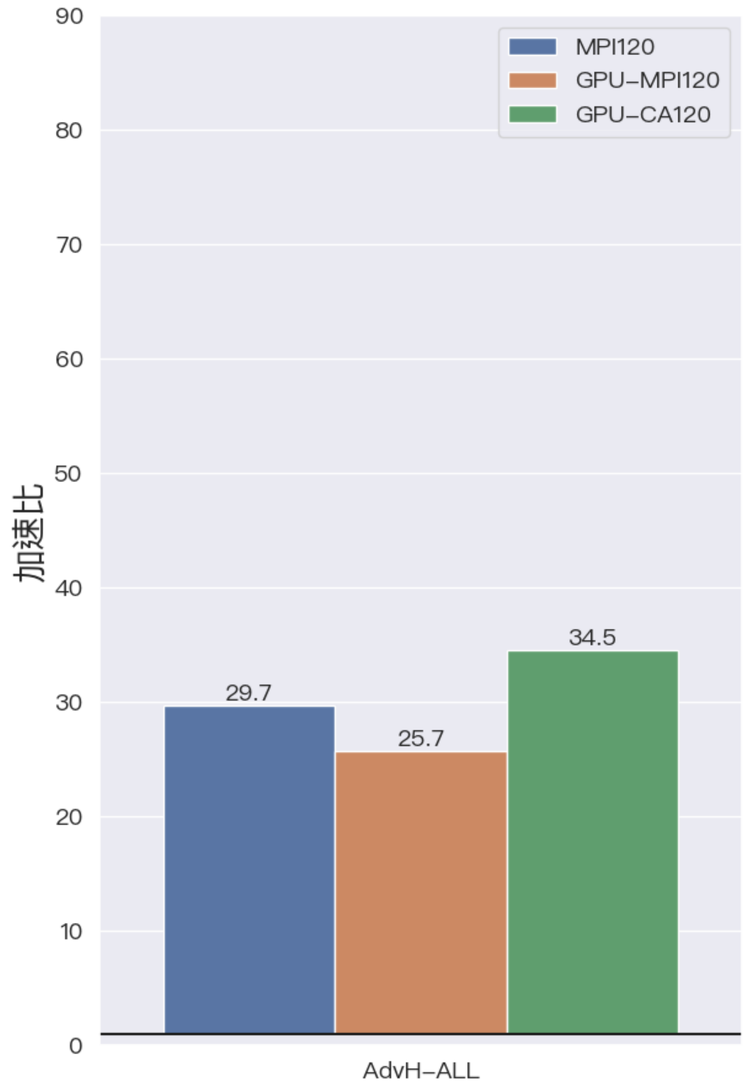
GPU-CA


Project 1
nsys-TCo383h24_CudaAware_MPIx02y60
nsys-TCo383h24_CudaAware_MPIx02y40
nsys-TCo383h24_CudaAware_MPIx02y20
Timeline View
1x
27 warnings, 372 messages
294s
295s
+100ms
+200ms
+300ms
+400ms
+500ms
+600ms
+700ms
+800ms
CPU (256)
Processes (46)
[226297] tcogfs.x
CUDA HW (0000:0f:00.0 - NVI
Threads (10)
[226297] tcogfs.x
OpenACC
NVTX
CUDA API
Profiler overhead
Comp [5.169 s]
NDSL...
TracersAdv. [1.027 s]
AdvH-ALL [786.110 ms]
CUDA-GATHE...
CUD...

Speedup of AdvH

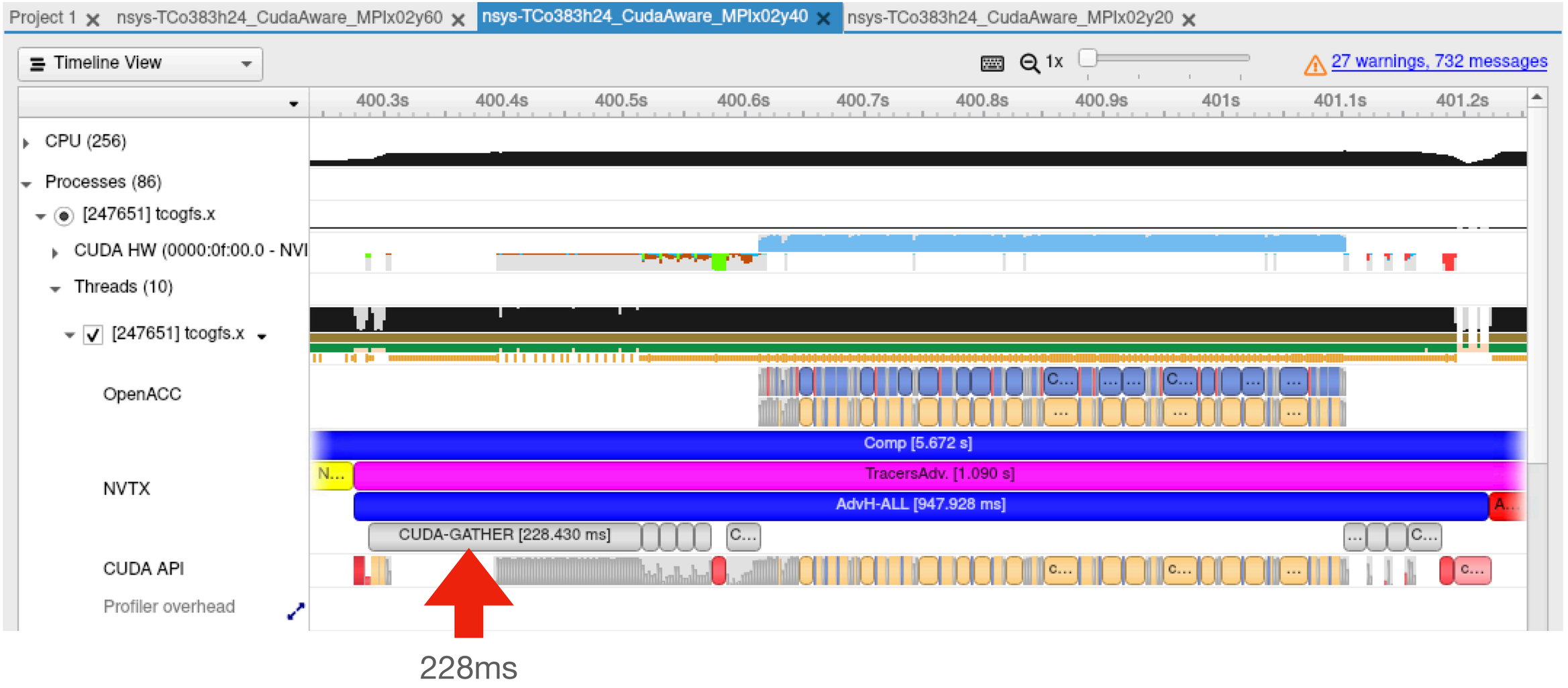






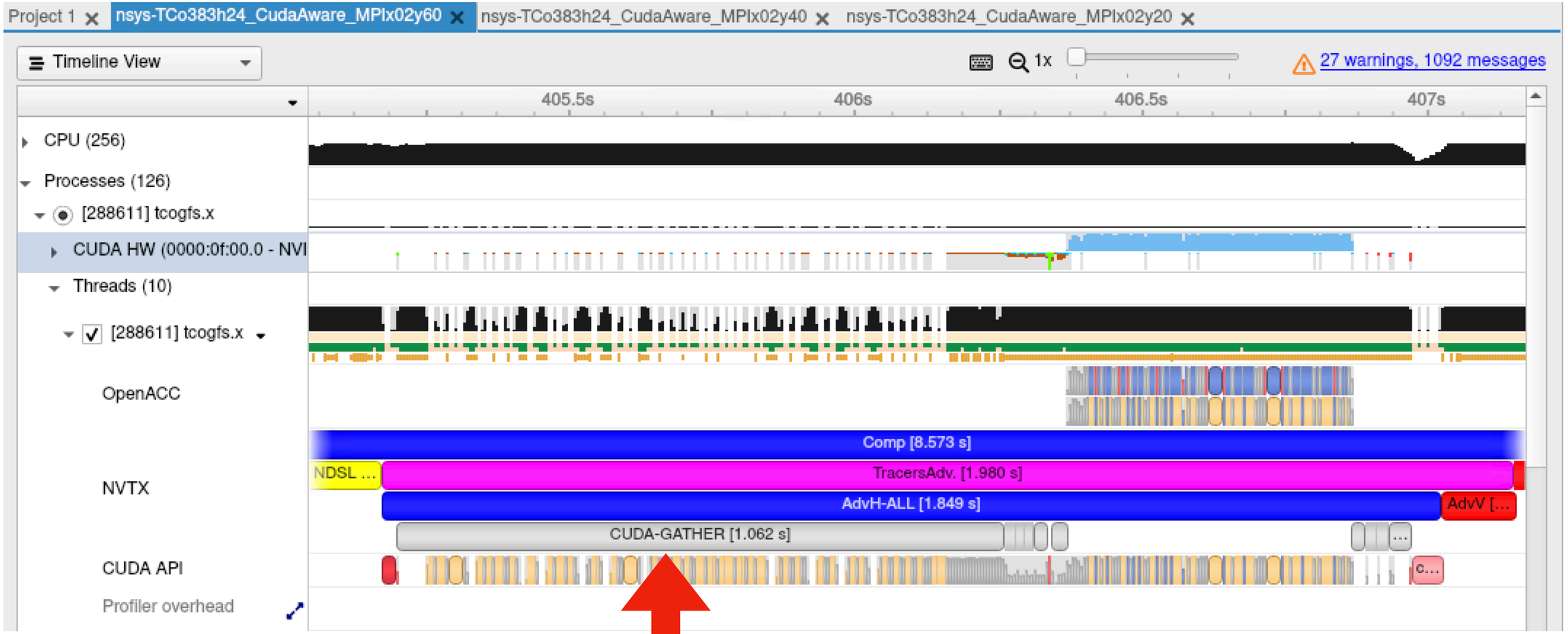
GPU-CA80

The first execution of gather operations is unusually slow.



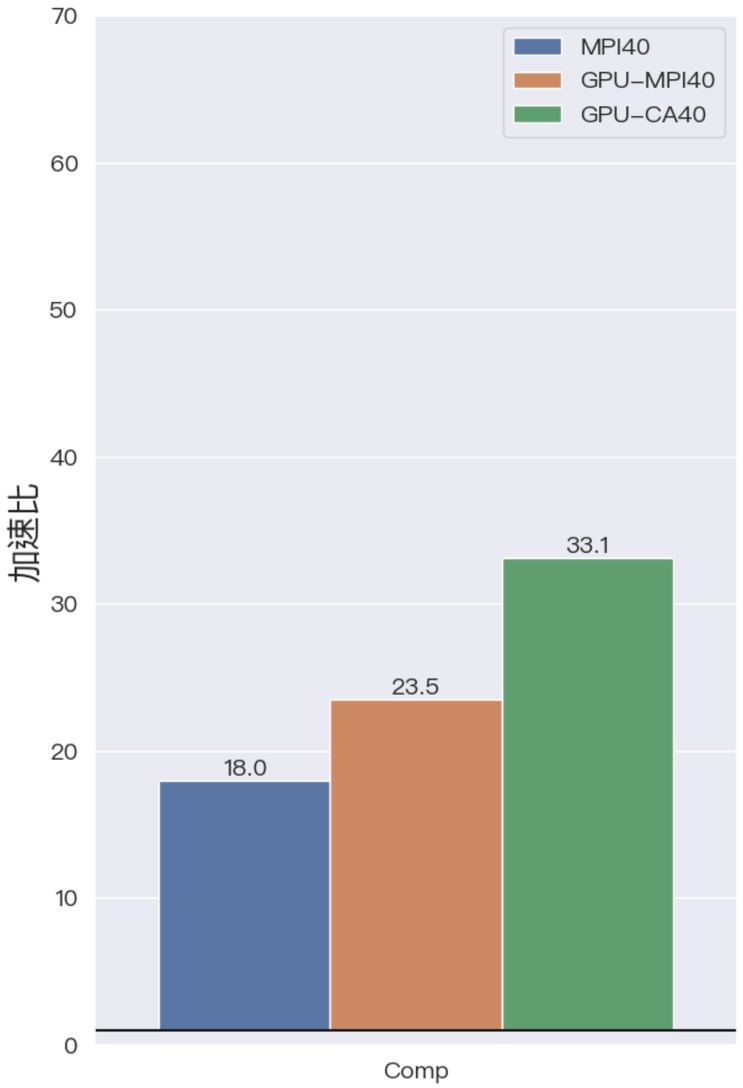
GPU-CA120

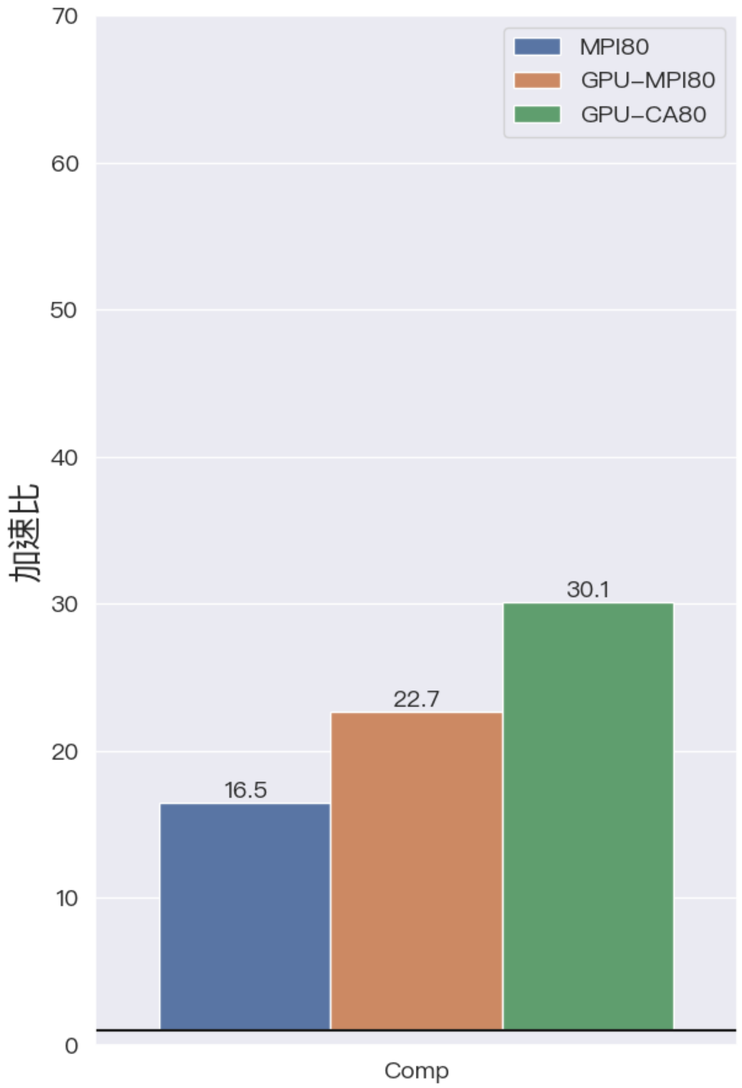
Increasing the number of processors, the first execution time of gather operations is even slower.

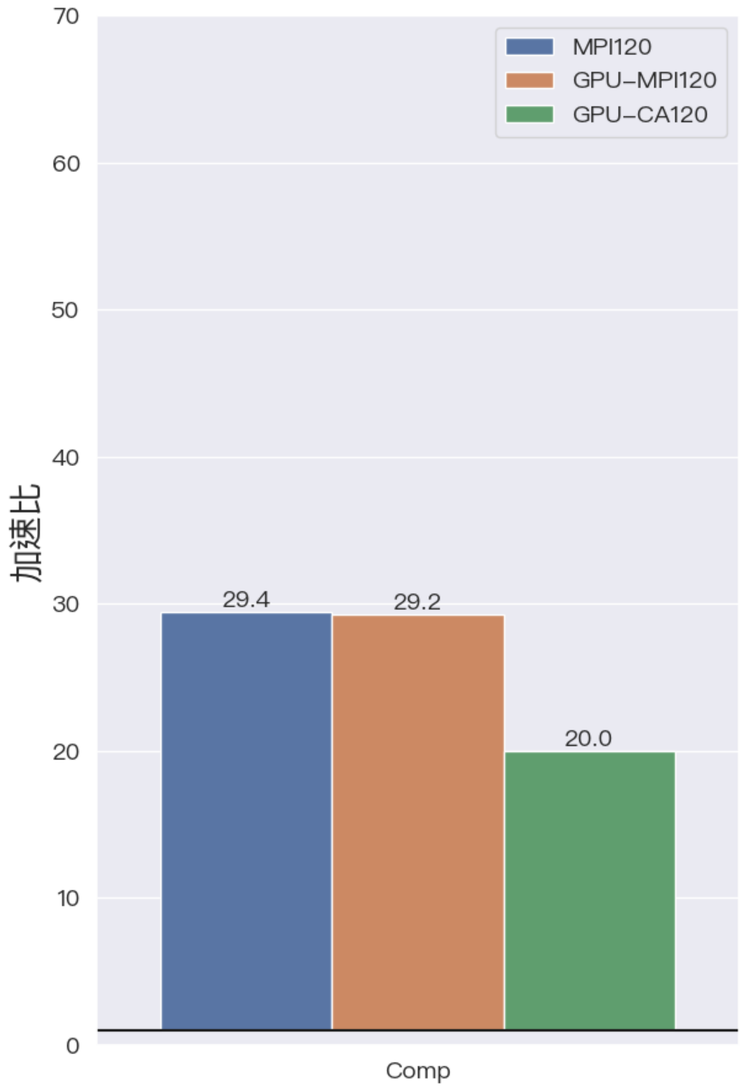


1062ms

Speedup of E2E wo I/O and init







Energy Efficiency

Compare energy consumption of a number of CPU only nodes with dual CPUs required to perform the same amount of work as 1 GPU node with 2 CPUs and 8 GPUs.

- E2E 1.8x (80 CPUs vs 1 GPU)
- Node replacement: 9.0x
- Node power efficiency: 1.6x
- Metric tons of CO_2 per year: 23 Tons

ASSUMPTIONS:

(1) The workload being input will run 24/7/365 on the node in question

(2) When the workload runs on a fraction of a CPU or GPU server, no other bottlenecks occur to stop it from scaling up to occupy the full server

(3) The calculations use TDP for both CPU and GPU. In reality, neither server will run full time at TDP. The comparison here is "worst case CPU" vs. "worst case GPU"

(4) Annual cost savings are operational for electricity only. Capital, personnel, etc are not included

(5) Perfect scaling of the workload to multiple nodes for CPUs

(6) Fractional workload scaling for both CPU and GPU nodes

(7) The GPU machine runs the CPUs a full speed, full power draw

Summary

- Porting AdvH to the GPU through OpenACC.
- Execute gather and scatter operators before and after AdvH computation using CUDA-Aware MPI on GPU.
- Before After Speedup
  - AdvH: 4.0x (80 CPUs vs 1 GPU)
  - E2E wo I/O, init: 1.8x (80 CPUs vs 1 GPU)
- Learn to use the performance analysis tool (Nsight Systems) to analyze program bottlenecks.
- Learn to use NVTX to annotate source code and provide information for the Nsight Systems.

Thank you for listening :)