S62388
Achieving Higher Performance From Data Center & Cloud Applications

Daniel Horowitz, NVIDIA, Senior Director of Engineering
Ankur Srivastava, Amazon, AWS Senior Solutions Architect

Agenda

Achieving Higher Performance From Data Center & Cloud Applications

Builds on GTC 2023 S51421 - Optimizing at Scale

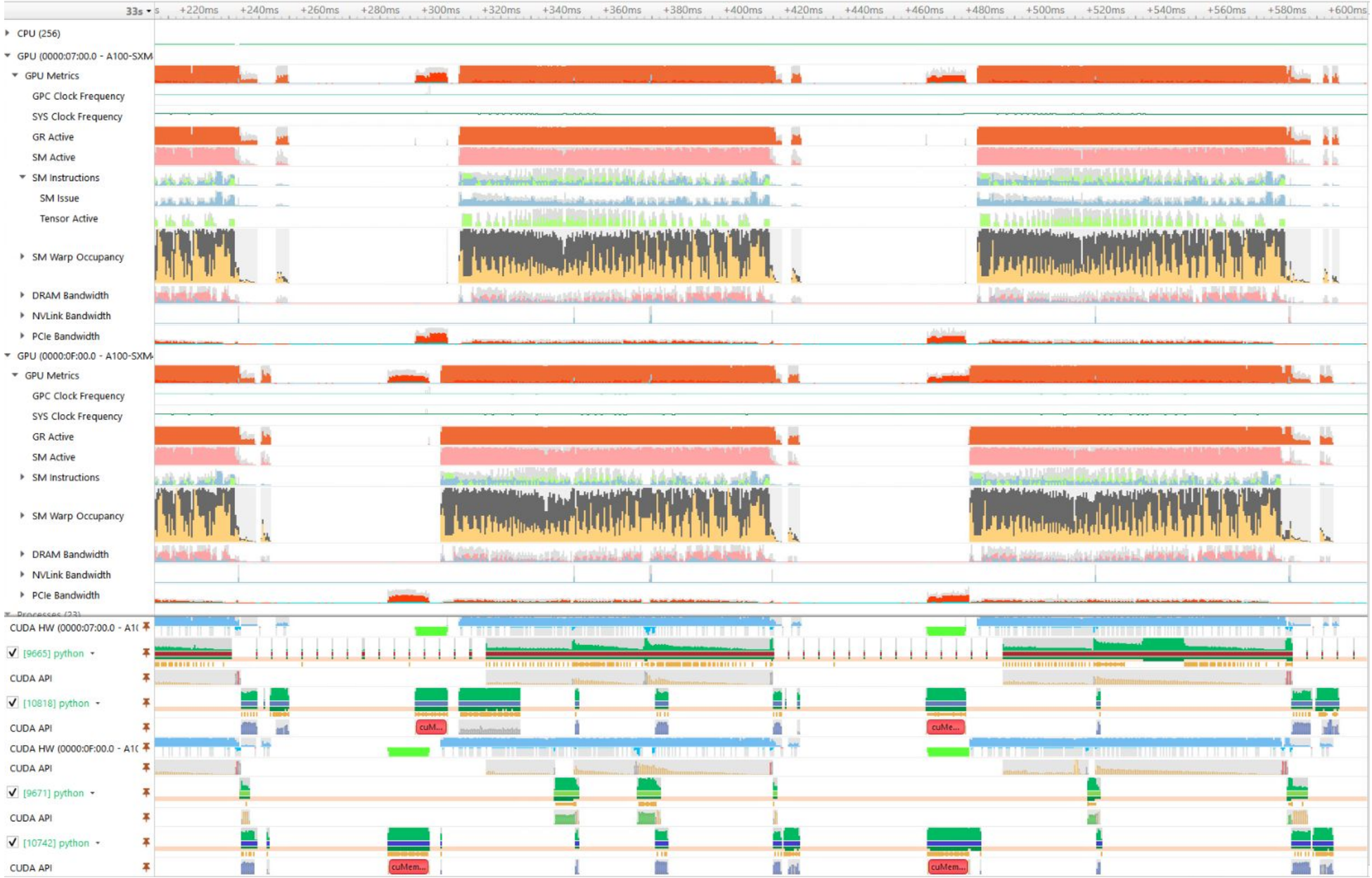
Investigating Hidden Bottlenecks for Multi-Node Workloads

- Intro/Recap
- Data Centers
- Cloud & AWS
- Remote Servers

For those not familiar with…

Deep Learning Training Example

Zoomed Out - Graph Mode



CPU cores

GPU metrics samples

SMs active

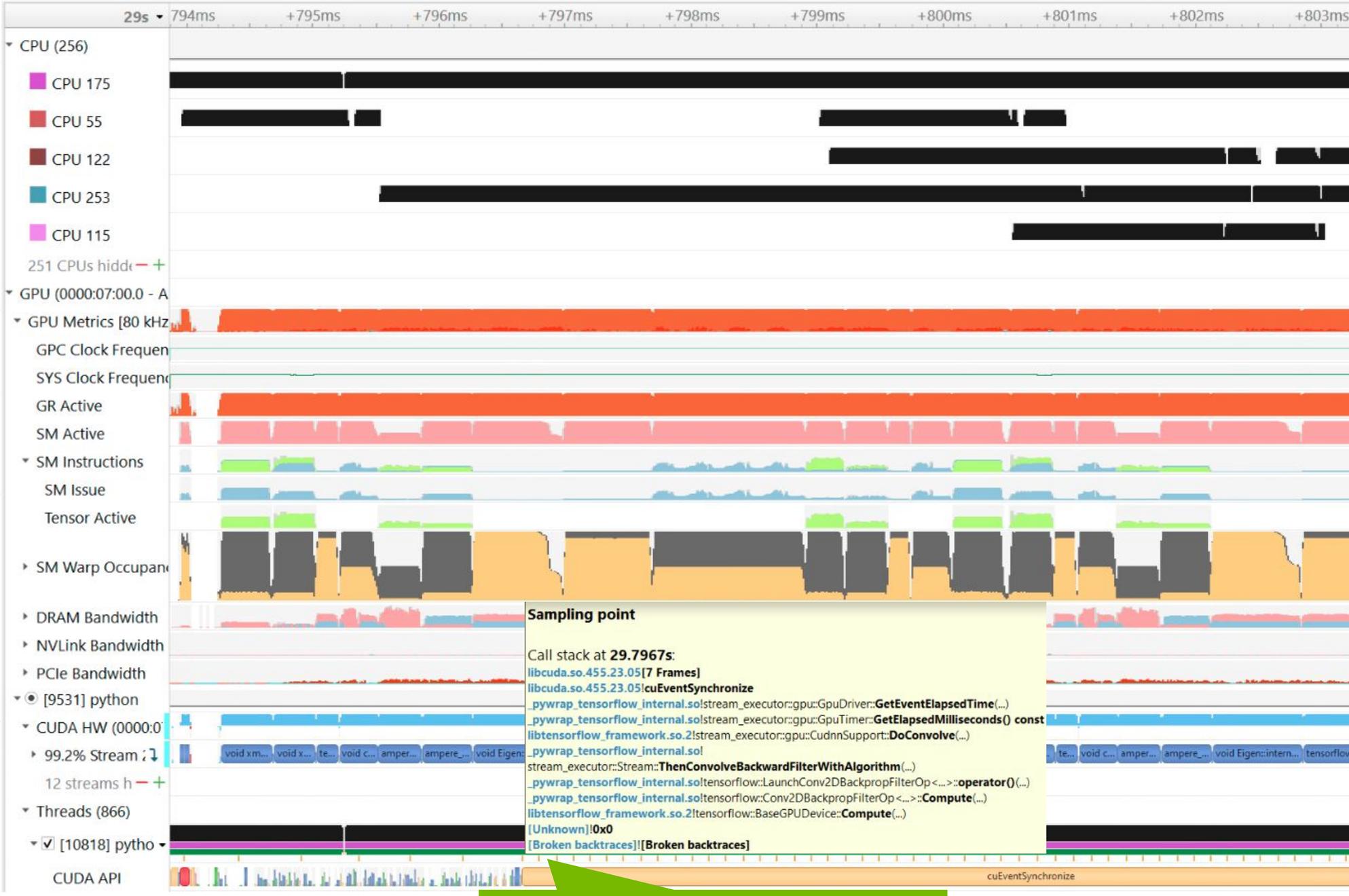
SM instructions TensorCores

GPU bandwidths

GPU CUDA kernel & memory activities

CPU processes & thread states

CUDA API trace



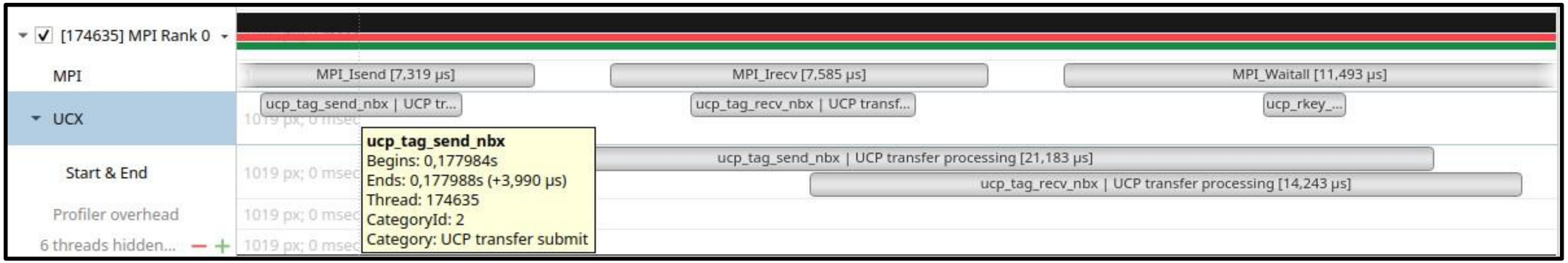
CPU call-stack samples

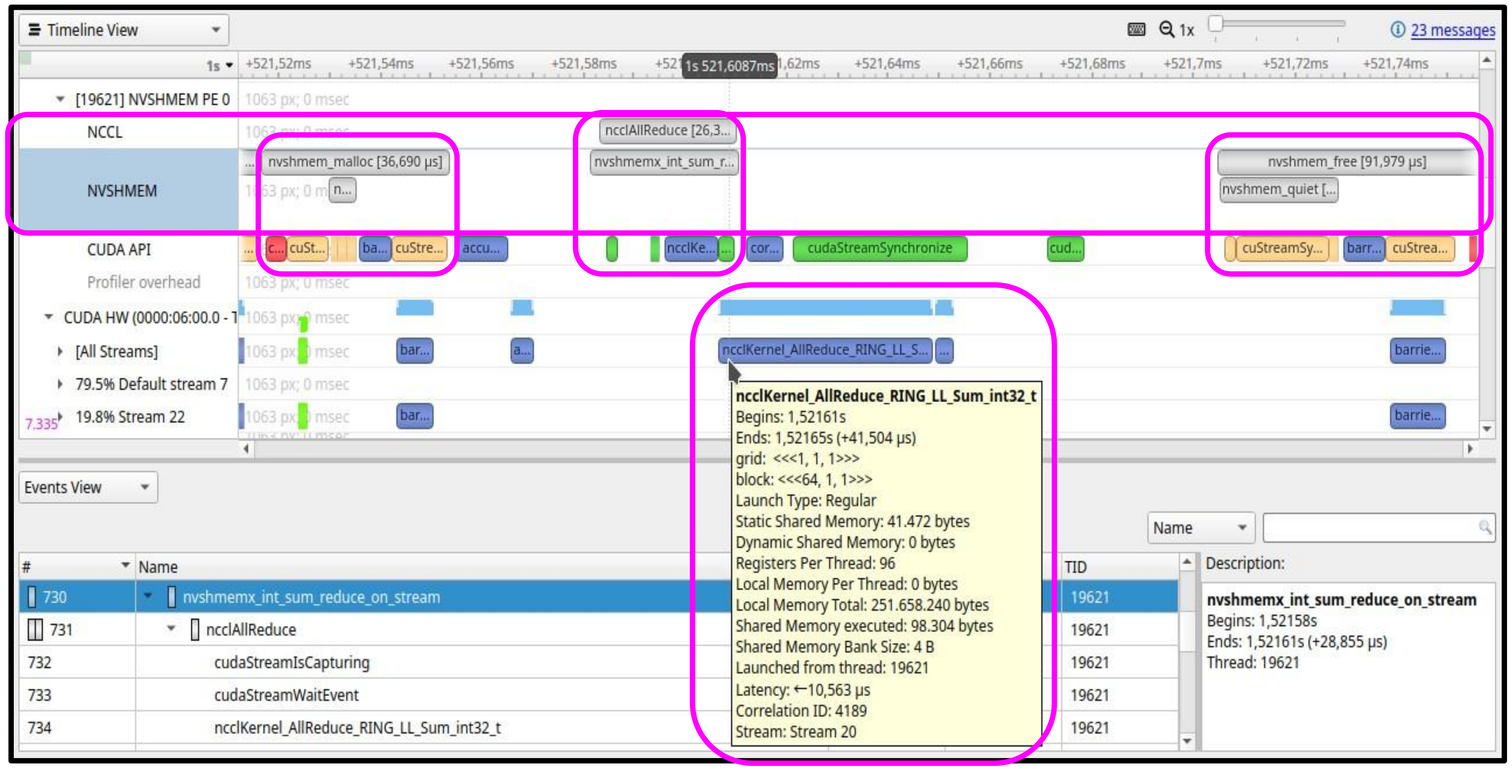
NVIDIA

Networking

Communications Trace - LIBC, UCX, MPI, NCCL, & NVSHMEM

APIs, Destinations, Sizes, Async Completion, & Related GPU Events

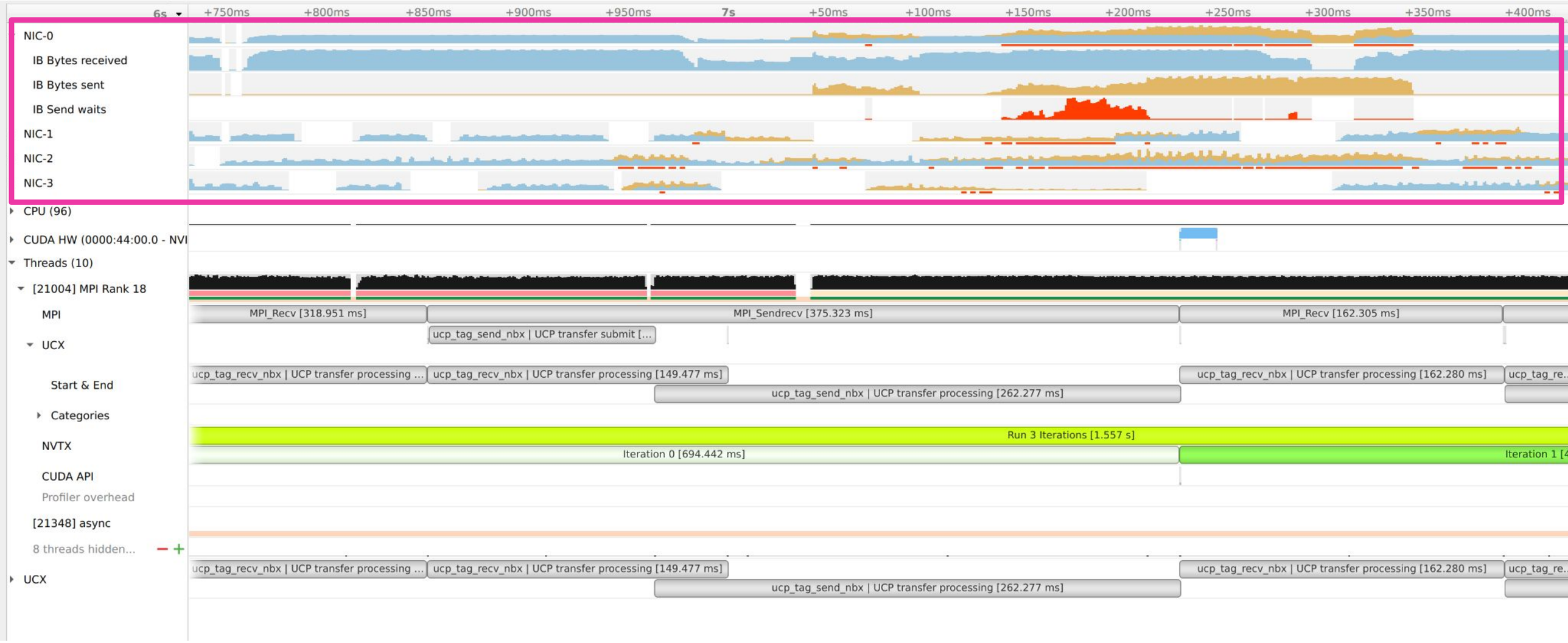




Description:

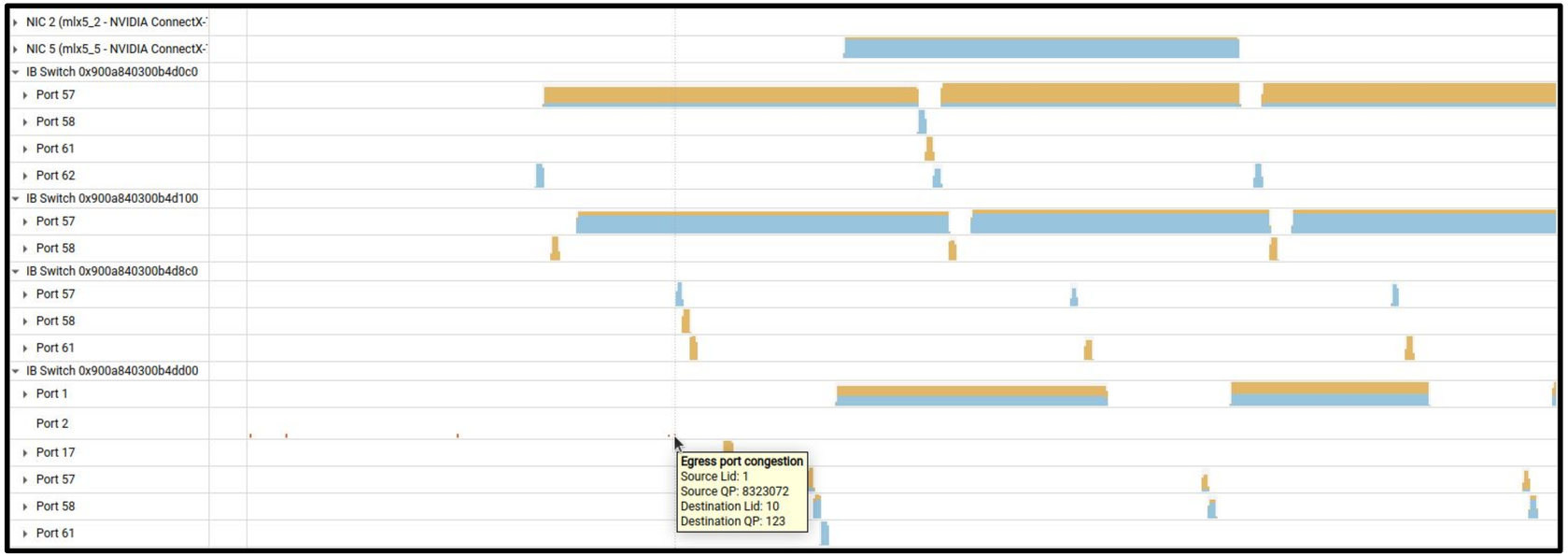
MPI_Bcast
Begins: 0,164935s
Ends: 0,165045s (+109,210 µs)
Thread: 1342434
Bytes sent: 0
Bytes received: 4
Root: 0
MPI_COMM_WORLD

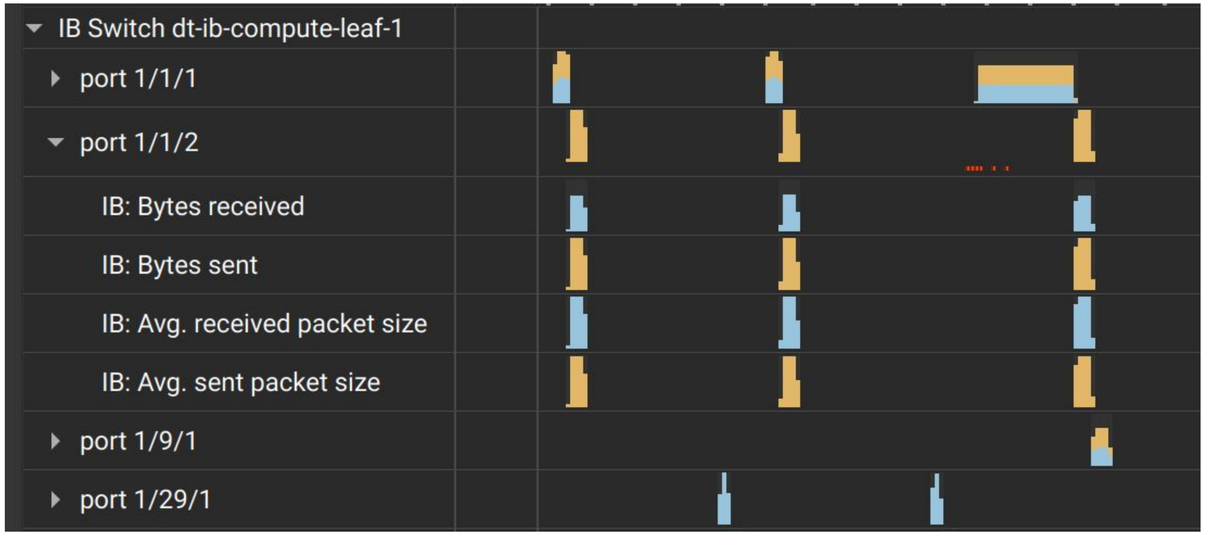
NVIDIA NIC/HCA Metrics Sampling (InfiniBand & RoCE)

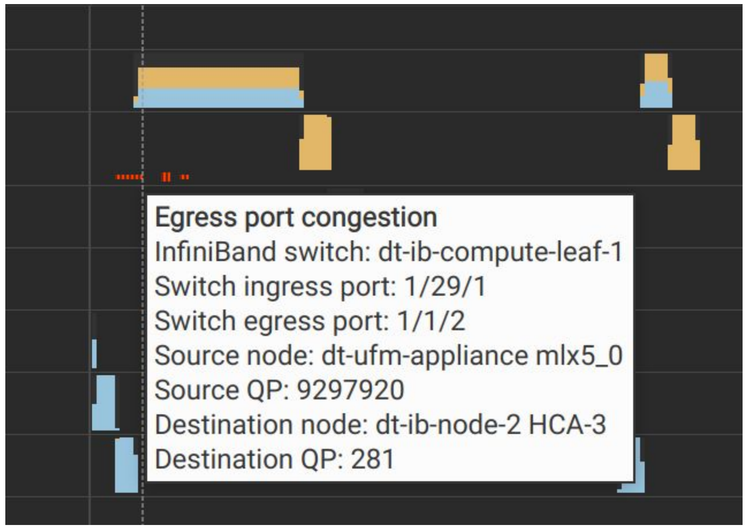


NVIDIA Quantum-2 Switch Sampling

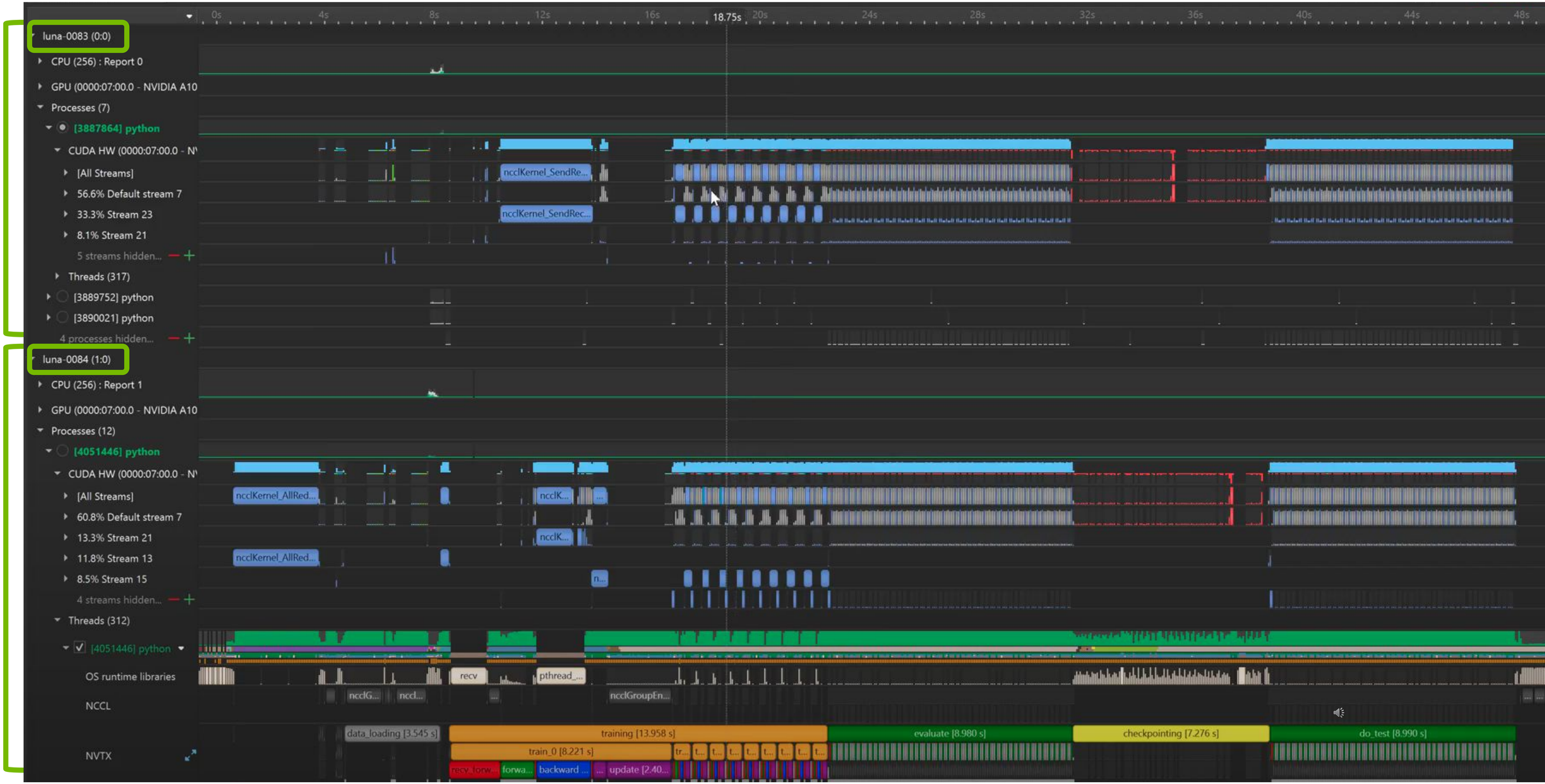
Send, Receive, Congestion, & Now Per-Port Sampling







Combining Reports Dynamically - But Does That Scale?



Scaling Out
Multi-Report Analysis

Flexible Scale-Out Profiling

- Infrastructure diversity
  - On-premise or cloud service providers
  - Node hardware and networking fabrics
  - Single or multi-tenancy
  - Bare metal, virtual machines, containers
  - Job schedulers(Slurm), task scheduler(Dask), orchestrators(Kubernetes)
    - Micro-services vs homogenous tasks
- Often isolation with communication limitations
- No “single point of contact” required to setup & profile
- Pick and choose for your needs & evolving ecosystems
  - Many profiles taken independently

Multi-Node / Multi-Report Analysis

Reduce “getting started” time

Post-process instead of complicated realtime setups

- Which might not be possible in your environment

Analyze thousands of reports

- No manual measure & search
- Avoid “guess which file”
- Cluster-wide conclusion
- Scale-out processing

Library of recipes → customizable

Results presentations (jupyter, graphs, …)

Collaborate / Share

Analysis Recipes Library <= 2024.1

CLI Usage: nsys recipe [<args>] <recipe name> [<recipe args>]

cuda_api_sum -- CUDA API Summary

cuda_api_sync -- CUDA Synchronization APIs

cuda_gpu_kern_pace -- CUDA GPU Kernel Pacing

cuda_gpu_kern_sum -- CUDA GPU Kernel Summary

cuda_gpu_mem_size_sum -- CUDA GPU MemOps Summary (by Size)

cuda_gpu_mem_time_sum -- CUDA GPU MemOps Summary (by Time)

cuda_gpu_time_util_map -- CUDA GPU Time Utilization Heatmap

cuda_memcpy_async -- CUDA Async Memcpy with Pageable Memory

cuda_memcpy_sync -- CUDA Synchronous Memcpy

cuda_memset_sync -- CUDA Synchronous Memset

diff -- Statistics Diff

dx12_mem_ops -- DX12 Memory Operations

gpu_gaps -- GPU Gaps

gpu_metric_util_map -- GPU Metric Utilization Heatmap

gpu_time_util -- GPU Time Utilization

mpi_gpu_time_util_map -- MPI and GPU Time Utilization Heatmap

mpi_sum -- MPI Summary

nccl_gpu_proj_sum -- NCCL GPU Projection Summary

nccl_sum -- NCCL Summary

nvtx_gpu_proj_pace -- NVTX GPU Projection Pacing

nvtx_gpu_proj_sum -- NVTX GPU Projection Summary

nvtx_gpu_proj_trace -- NVTX GPU Projection Trace

nvtx_pace -- NVTX Pacing

nvtx_sum -- NVTX Range Summary

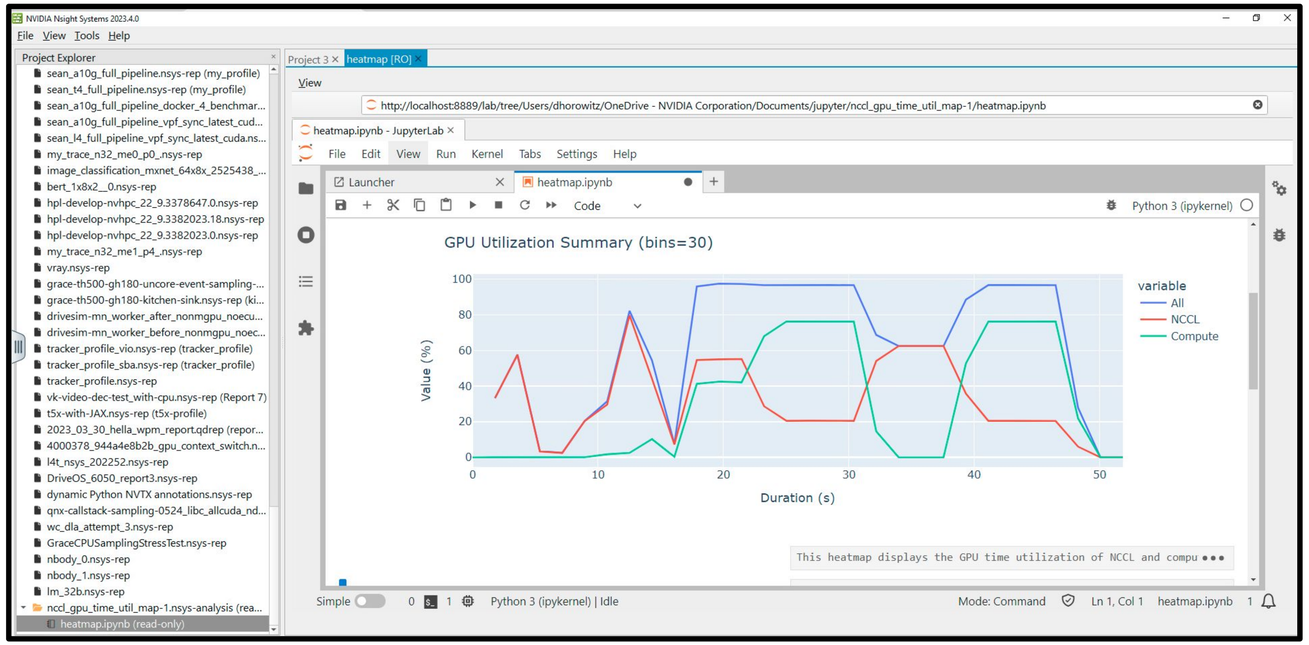
osrt_sum -- OS Runtime Summary

and growing...

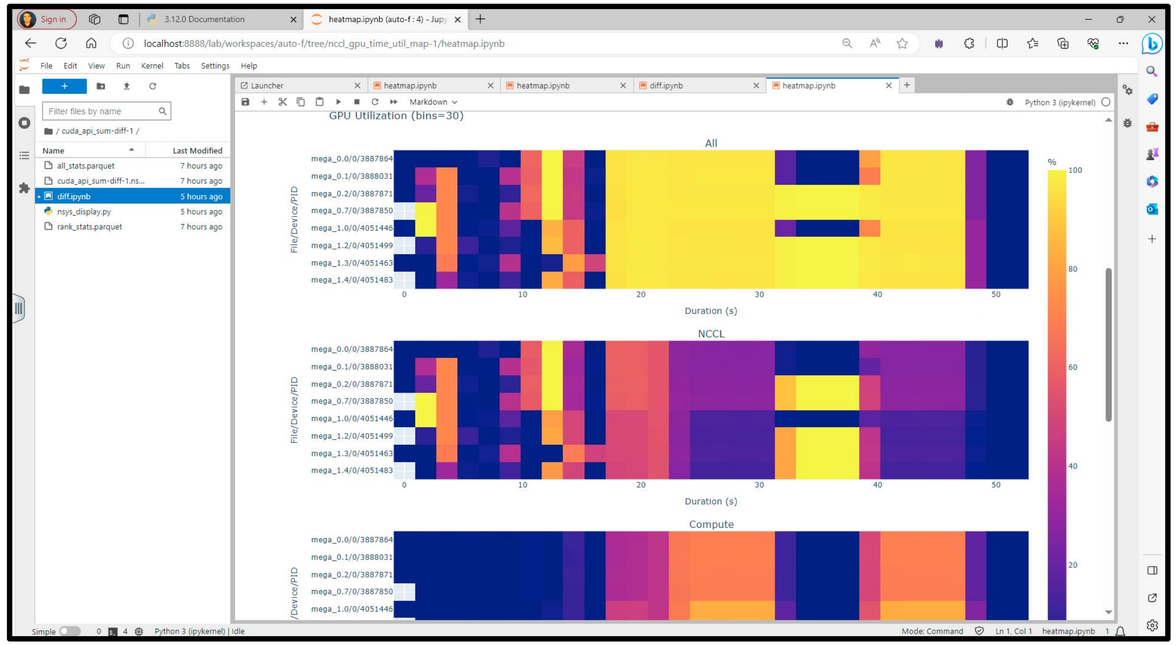
Jupyter Notebooks As Data Presentation

Loadable in...

Nsight Systems



Jupyter Lab

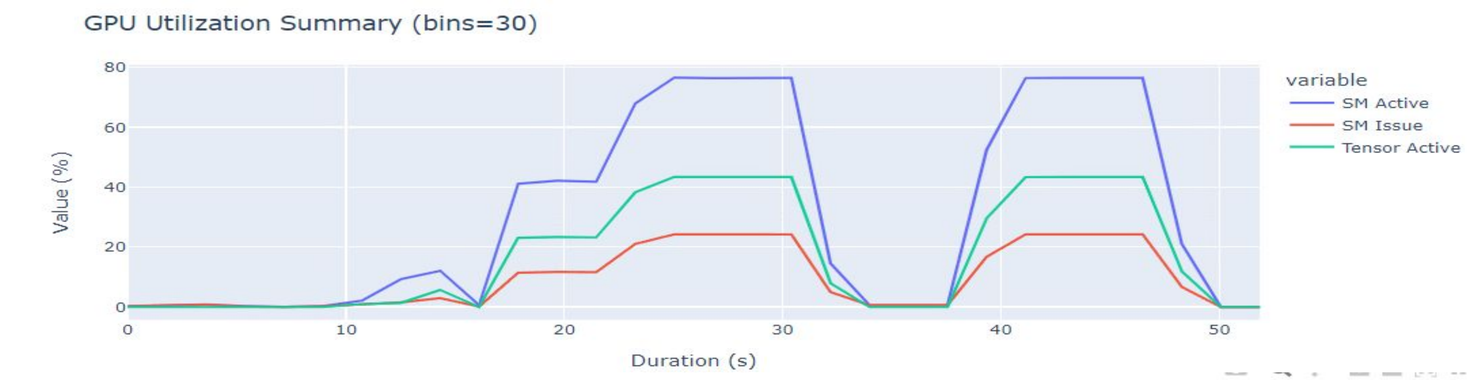


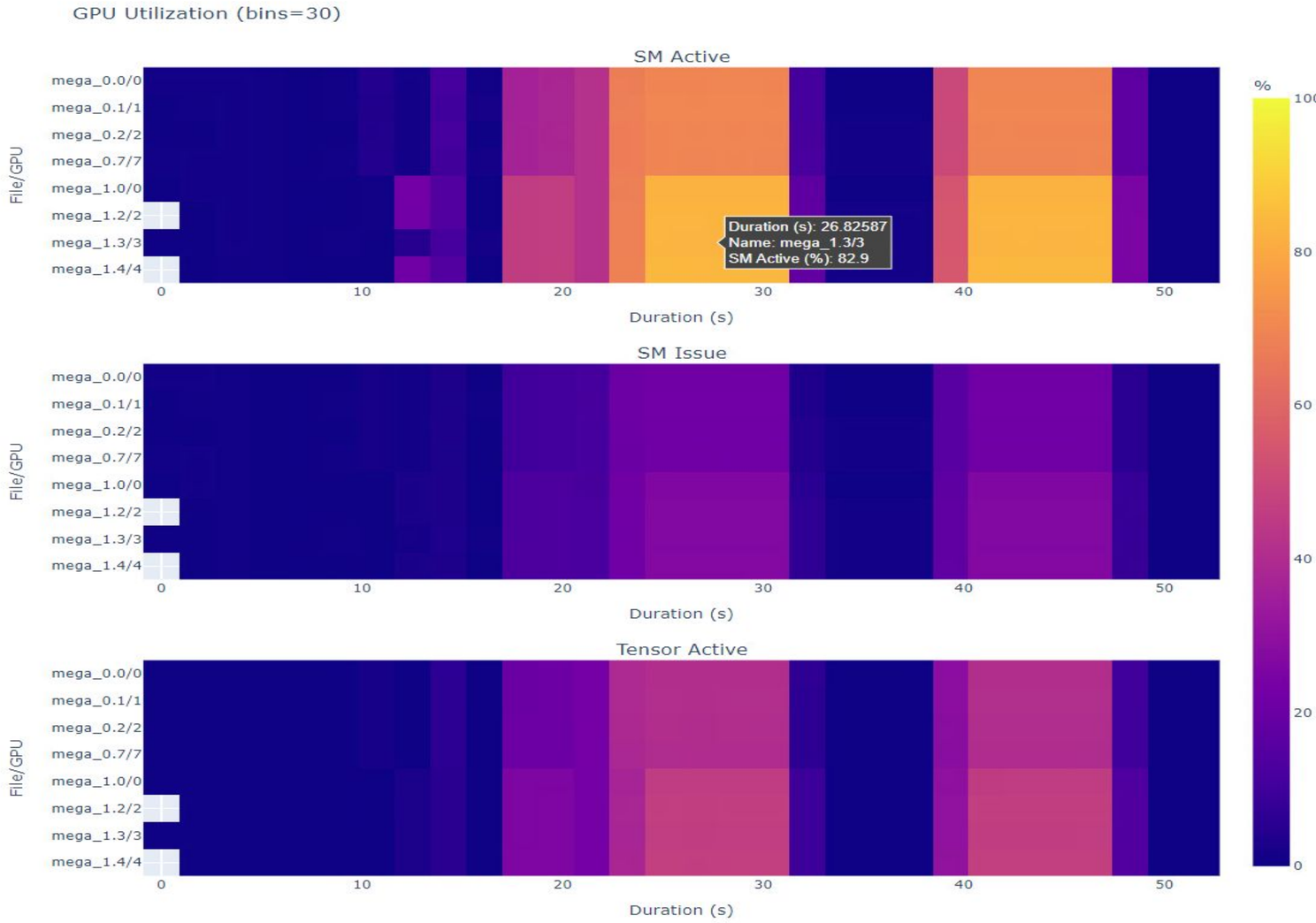
Example From GTC 2023 - DL Training Barrier Time

Heatmaps Overview

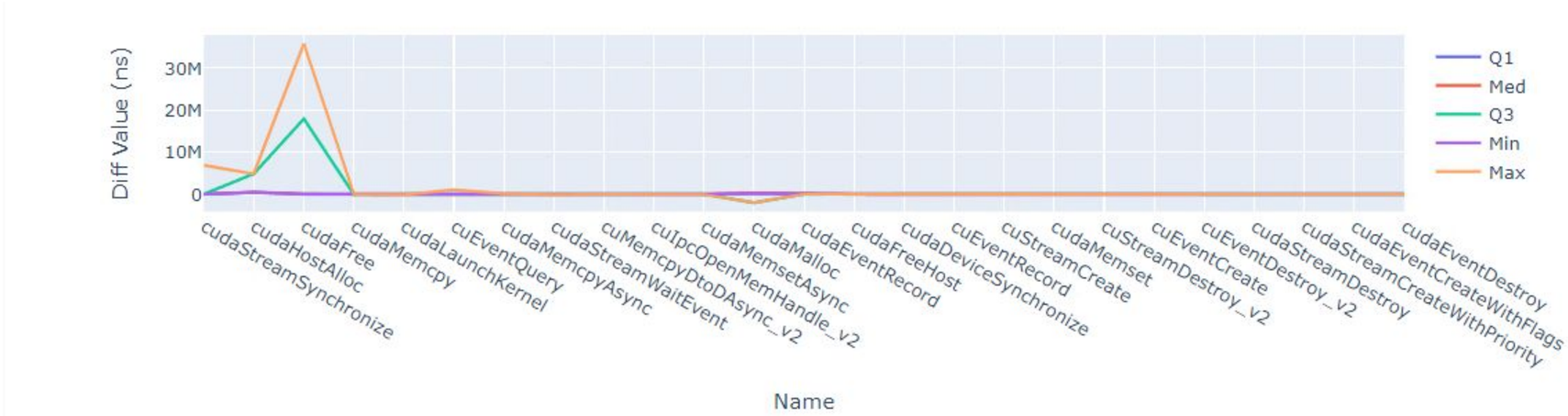
Ex: GPU metrics

- SM Active
- SM Issue
- Tensor Active





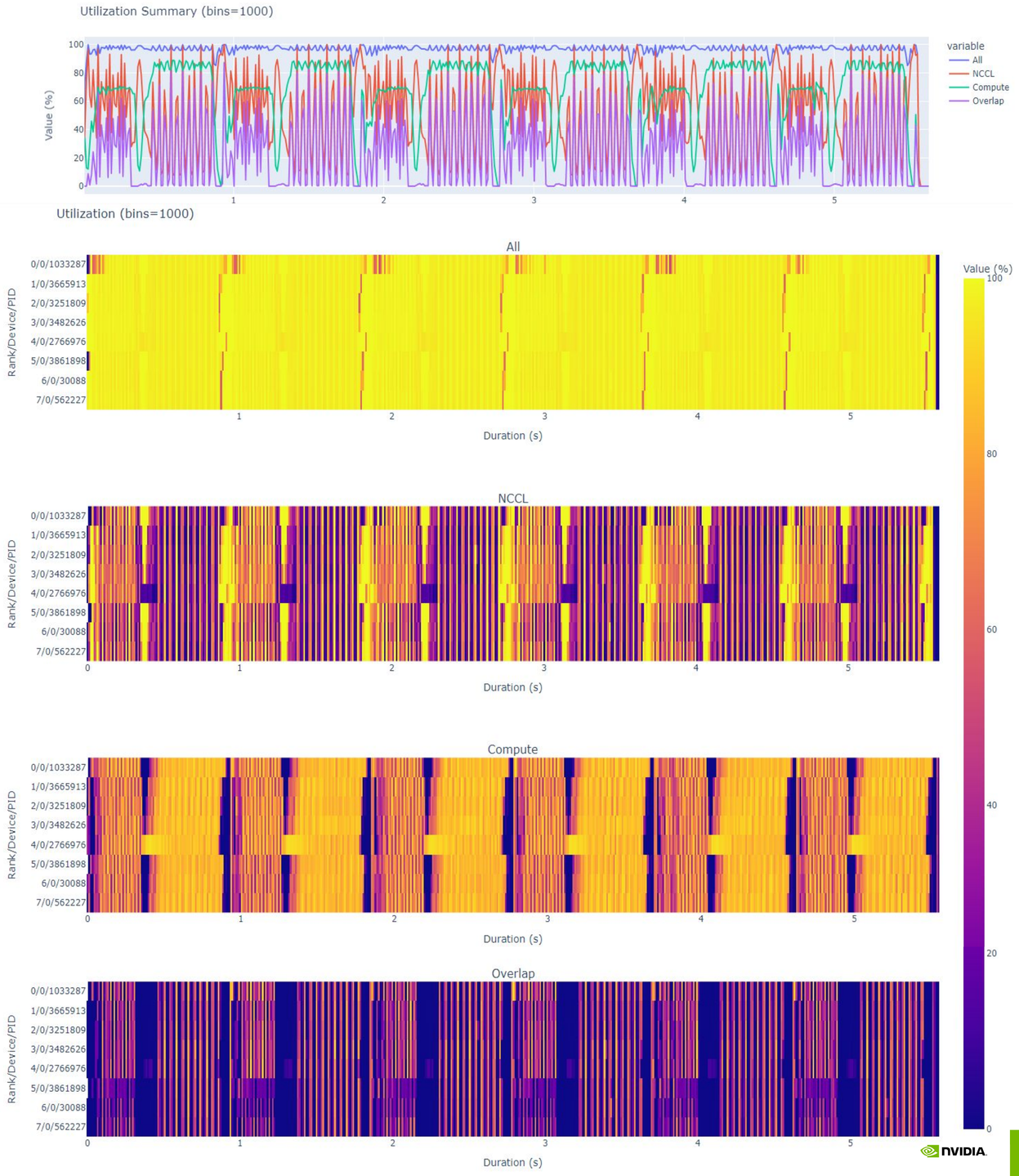
Recipe Report Differencing w/ Graphs



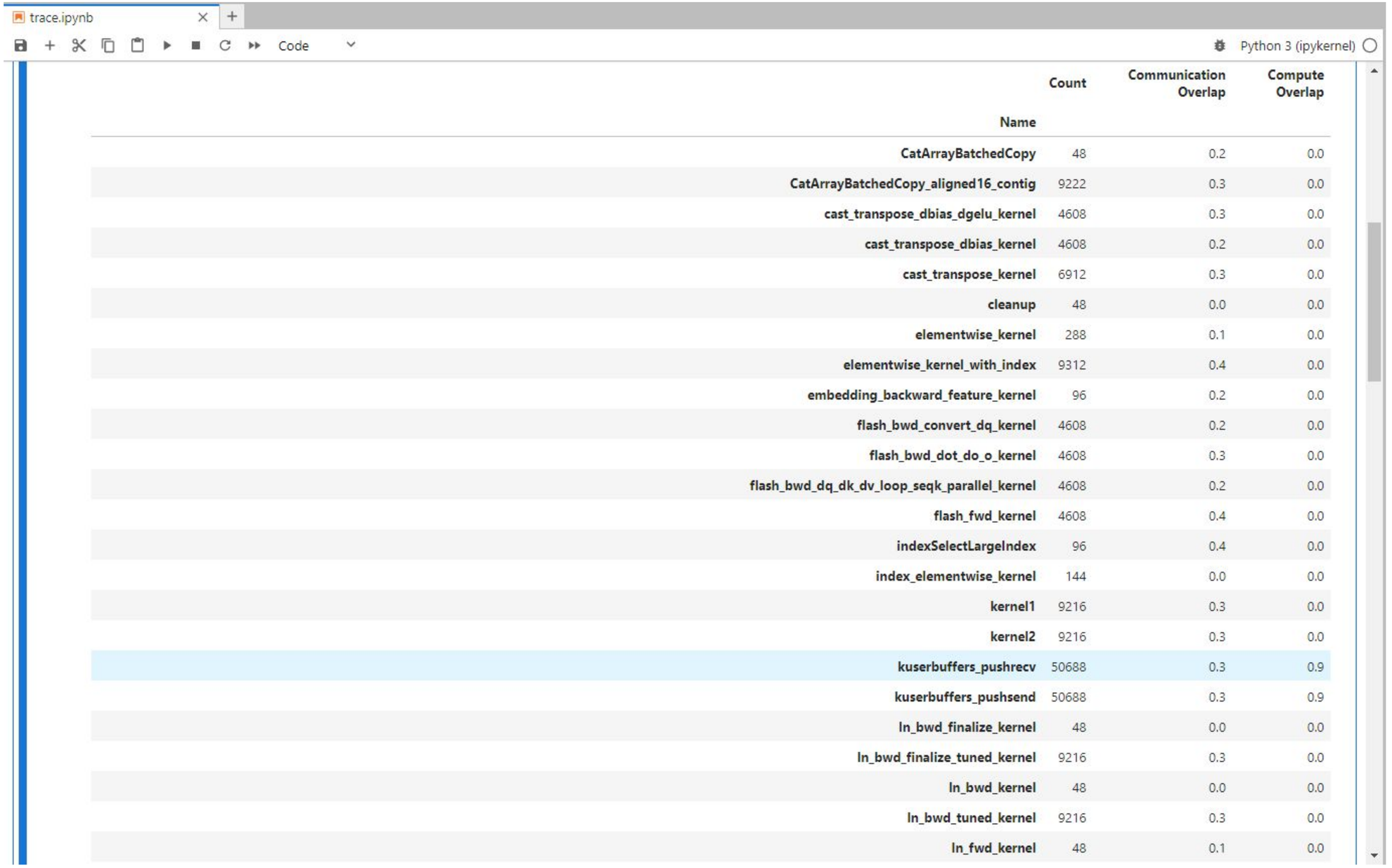
Name	Time	Total Time	Num Calls	Avg	Q1	Med	Q3	Min	Max	StdDev
cudaStreamSynchronize	-0.3	55548526	0	6172.1	0.0	-40.0	2756.5	60	6879563	18388.9
cudaHostAlloc	-0.8	831411	0	138568.5	424233.0	475441.0	4861555.0	424233	4861555	-354439.0
cudaFree	1.1	124139003	0	10344917.0	12897.5	42314.5	17929245.0	9210	35779248	18757998.8
cudaMemcpy	0.0	-171010	0	-57003.3	-18380.0	-76804.0	-75826.0	-18380	-75826	-32734.4
cudaLaunchKernel	-0.1	-712243	0	-79.1	10.0	10.0	-400.0	-10	-133221	-845.9
cuEventQuery	0.0	1349876	1469	9.5	-10.0	20.0	-10.0	0	1034738	3768.8
cudaMemcpyAsync	0.0	-81256	0	-27.1	-50.0	-60.0	30.0	-90	149851	2315.2
cudaStreamWaitEvent	-0.1	-484673	0	-40.4	0.0	-10.0	-50.0	-20	-99931	-851.4
cuMemcpyDtoDAsync_v2	-0.1	-199439	0	-65.9	-130.0	-60.0	20.0	40	-1020	51.1
cuIpcOpenMemHandle_v2	0.0	-486229	0	-81038.2	960.0	-103223.0	-117344.0	960	-117344	-32562.1
cudaMemsetAsync	0.0	-545582	0	-181.9	-130.0	-160.0	-65.0	-170	-81550	-1086.6
cudaMalloc	0.0	-962676	0	-106964.0	167189.0	273113.0	-2032764.0	167189	-2032764	-604037.8
cudaEventRecord	0.0	-74148	0	-8.2	-20.0	-20.0	60.0	-10	460	-50.1

Compute-Comms Overlap

- GPU Utilization
- Communications
- Compute
- Overlap



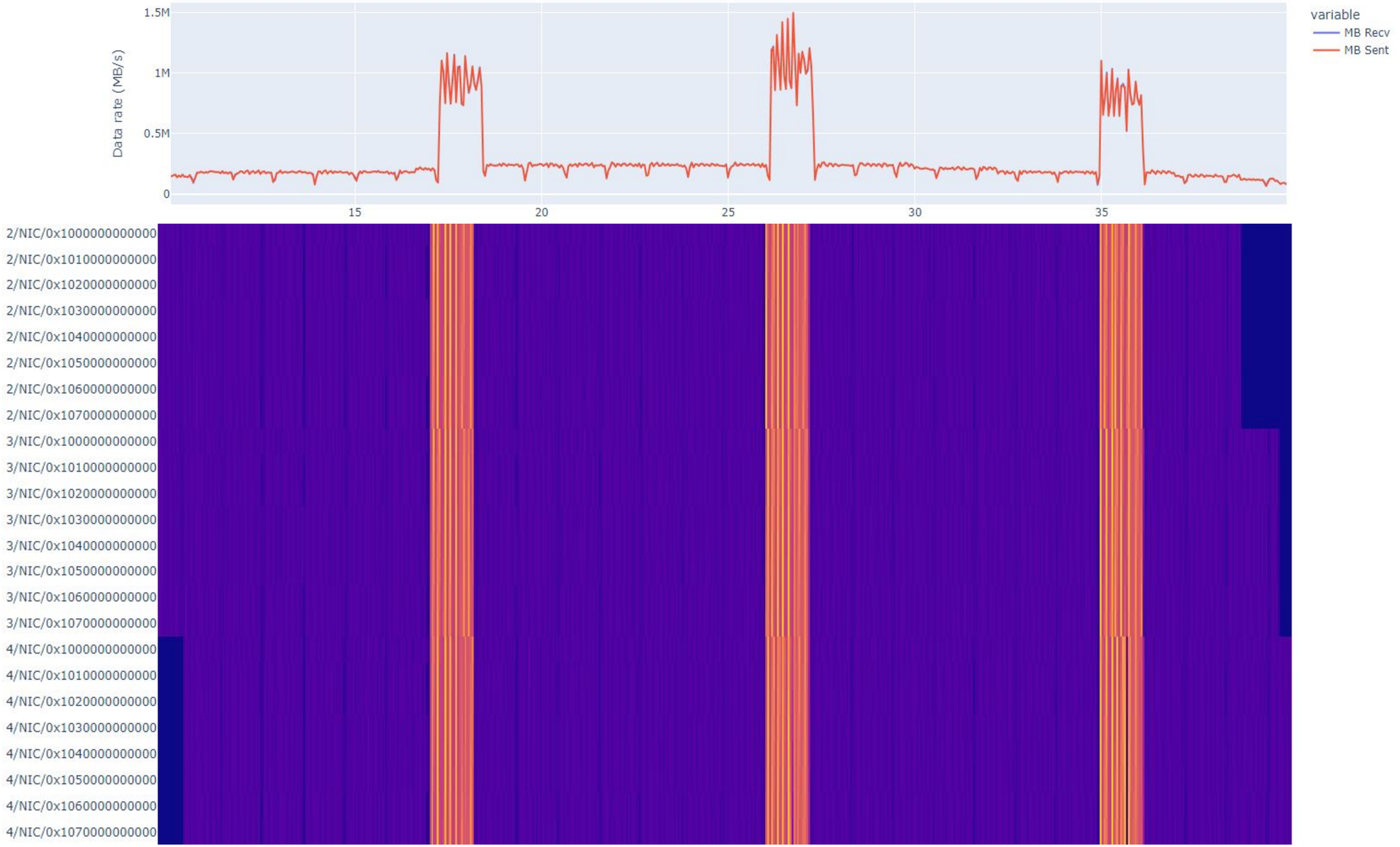
CUDA Kernel Overlap Statistics



Name	Count	Communication Overlap	Compute Overlap
CatArrayBatchedCopy	48	0.2	0.0
CatArrayBatchedCopy_aligned16_contig	9222	0.3	0.0
cast_transpose_dbias_dgelu_kernel	4608	0.3	0.0
cast_transpose_dbias_kernel	4608	0.2	0.0
cast_transpose_kernel	6912	0.3	0.0
cleanup	48	0.0	0.0
elementwise_kernel	288	0.1	0.0
elementwise_kernel_with_index	9312	0.4	0.0
embedding_backward_feature_kernel	96	0.2	0.0
flash_bwd_convert_dq_kernel	4608	0.2	0.0
flash_bwd_dot_do_o_kernel	4608	0.3	0.0
flash_bwd_dq_dk_dv_loop_seqk_parallel_kernel	4608	0.2	0.0
flash_fwd_kernel	4608	0.4	0.0
indexSelectLargeIndex	96	0.4	0.0
index_elementwise_kernel	144	0.0	0.0
kernel1	9216	0.3	0.0
kernel2	9216	0.3	0.0
kuserbuffers_pushrecv	50688	0.3	0.9
kuserbuffers_pushsend	50688	0.3	0.9
ln_bwd_finalize_kernel	48	0.0	0.0
ln_bwd_finalize_tuned_kernel	9216	0.3	0.0
ln_bwd_kernel	48	0.0	0.0
ln_bwd_tuned_kernel	9216	0.3	0.0
ln_fwd_kernel	48	0.1	0.0

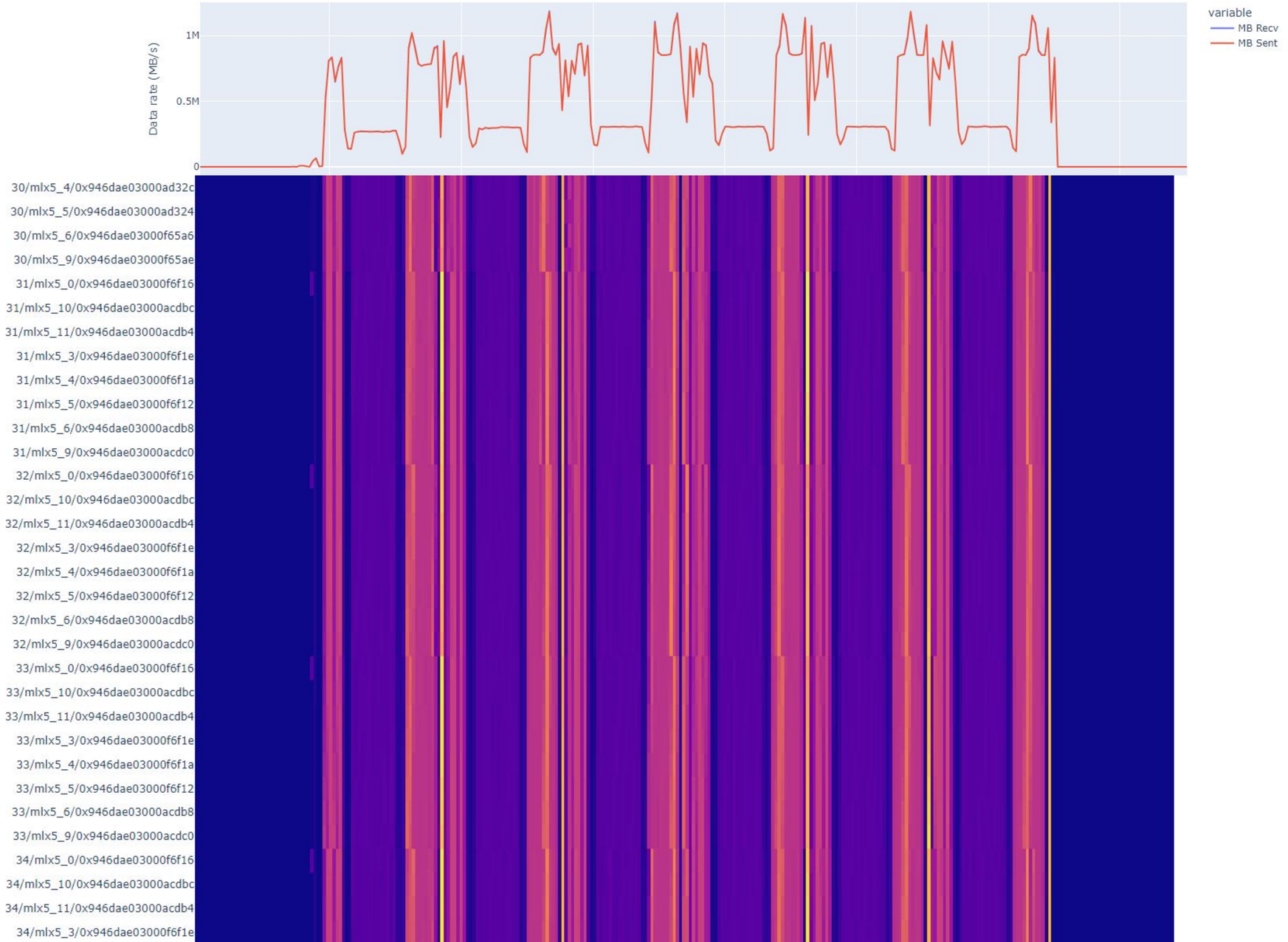
NVIDIA HCA/NIC Throughput Heatmaps

Example - Llamas training 64 HCAs - Many NICs off-screen



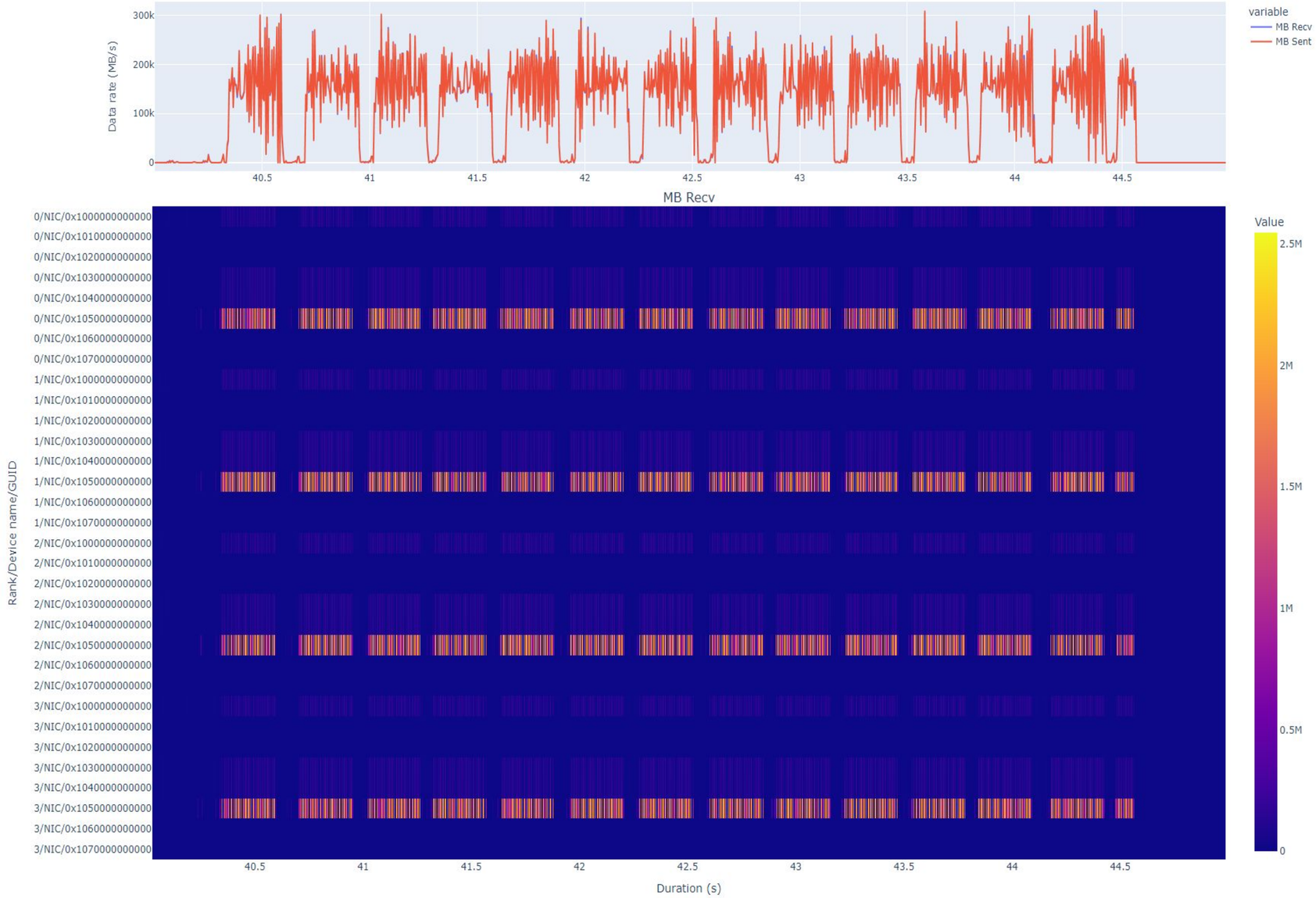
NVIDIA HCA/NIC Throughput Heatmaps

Example - LLM training 128 HCAs - Many NICs off-screen



NVIDIA HCA/NIC Throughput Heatmaps

Example - Gromacs 16 HCAs 4 nodes - Many NICs off-screen



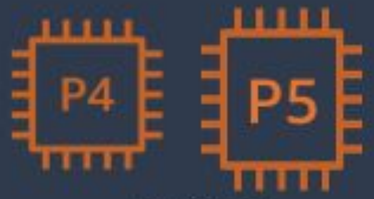
Per node, 1 GPU for long-range (PME) force calculations, which require all-to-all comms (1 NIC pinned per GPU for best perf). The remaining 3 GPUs per node are dedicated to short-range force calcs which have less intensive comm requirements. Massively Improved Multi-node NVIDIA GPU Scalability with GROMACS | NVIDIA Technical Blog

Cloud and Amazon AWS Collaborations

Ankur Srivastava, Amazon, AWS Senior Solutions Architect

Distributed training – HPC-ML cluster building blocks

Compute

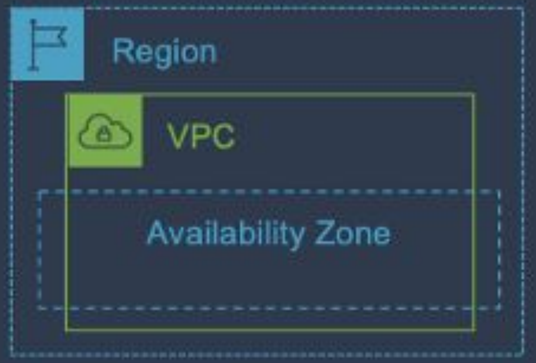


Instances

Containers

Network

Elastic Fabric Adapter



Storage

Amazon FSx for Lustre

Amazon EC2 Instance Store

Amazon Elastic Block Store (Amazon EBS)

Architecture & Orchestration

AWS ParallelCluster

Amazon EKS

AWS CloudFormation

Distributed Training on AWS

Sagemaker Jobs	Fully Managed ML Service
Sagemaker Hyperpod	Resilient and Persistent clusters dedicated for distributed training
AWS ParallelCluster	Slurm based HPC clusters
Amazon EKS	Kubernetes based clusters

Best Practices for Large Scale Distributed Training

Step by step guides to create clusters:

- One-click VPC Deployments
- Mount FSx for Lustre Filesystems
- EFA enabled clusters

Recipes to customize AMIs

AWS optimized Dockerfiles

EFA cheatsheet

Distributed Training Examples:

- Slurm scripts/K8 manifests
- Working with Pyxis/Enroot
- Nemo (MegatronLM, Multimodal, Bionemo)
- MosaicML
- DDP, FSDP
- SMDP, SMMP
- Tensorflow/Jax

Validation (NCCL Tests etc)

Observability (Prometheus-Grafana etc)

Profiling (Nsight Product Family)

Running Nsight Systems with SLURM

- Nsight Systems comes pre-installed with DL AMIs
- NIC Sampler to collect EFA metrics will be available in the next Nsight release.
  - WIP to merge the latest Nsight Systems comes pre-installed with DL AMIs
- **Supports P5, P4, P4de, G5 and G4dn instances**
- EFA metrics are shared across GPUs on the same node
- Compatible with container plugins like Pyxis
- **srun -l "${ARGS[@]}" /fsx/nsys-slurm-exec python3 <train.py>**

nsys-slurm-exec

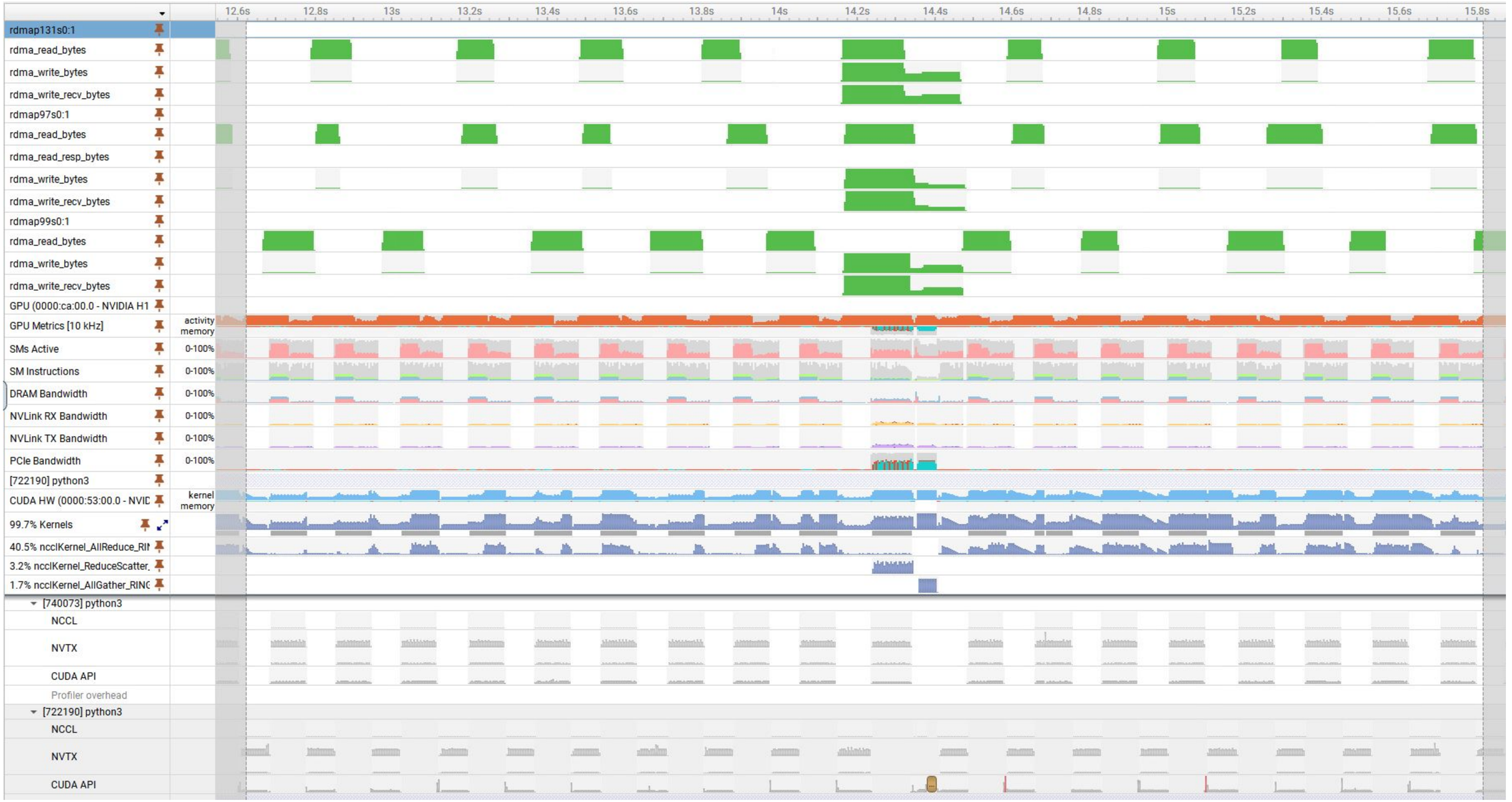
```
#! /bin/bash -x

NSYS_EXTRAS=""
if [ "$SLURM_LOCALID" == "0" ]; then
NSYS_EXTRAS="--enable nic_sampler,-mode:counters,-struct:true,-efa:true"
fi

/fsx/nsight-efa/target-linux-x64/nsys profile $NSYS_EXTRAS \
        --gpu-metrics-device all --sample none --delay 30 \
        --force-overwrite true \
        --output report_nccl_efa_job%q{SLURM_JOB_ID}_rank%q{SLURM_PROCID}_on_%q{HOSTNAME}.nsys-rep \
    "$@"
~
```

Amazon AWS Elastic Fabric Adapter(EFA) Sampling

Nemo training, in picture 3 of 32 EFA ports, 1 of 8 H100 GPUs …



Amazon AWS Elastic Kubernetes Service (EKS)

Previously profiling in Kubernetes required

- Building Nsight Systems into the container
- Or mount a volume it into the container
- Changing the run or exec commands
- K8/Helm spec edits for profile vs production

Now profiling in Kubernetes requires

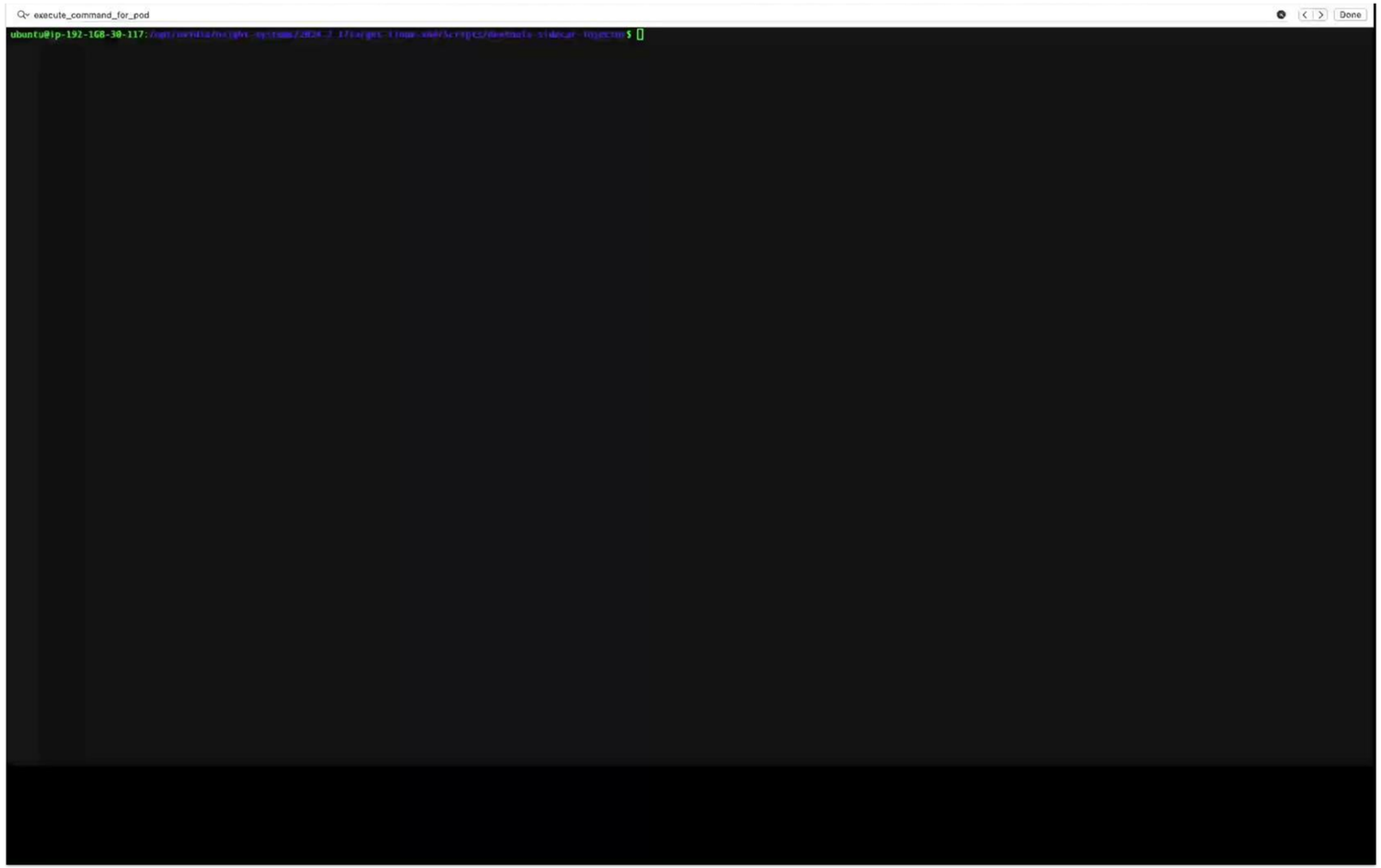
- Installing a helm chart & some custom values
- Labeling namespaces or pods of interest

NVIDIA’s controller takes care of the rest!

Collaboration to support Amazon AWS EKS

Managed Kubernetes have host restrictions

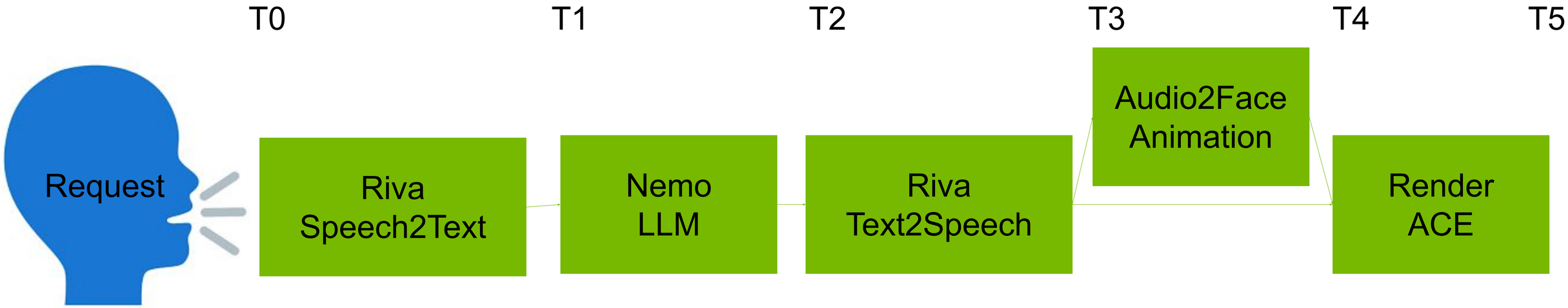
Iterating to make it more powerful & user friendly



Cloud Microservices
Kubernetes & Helm

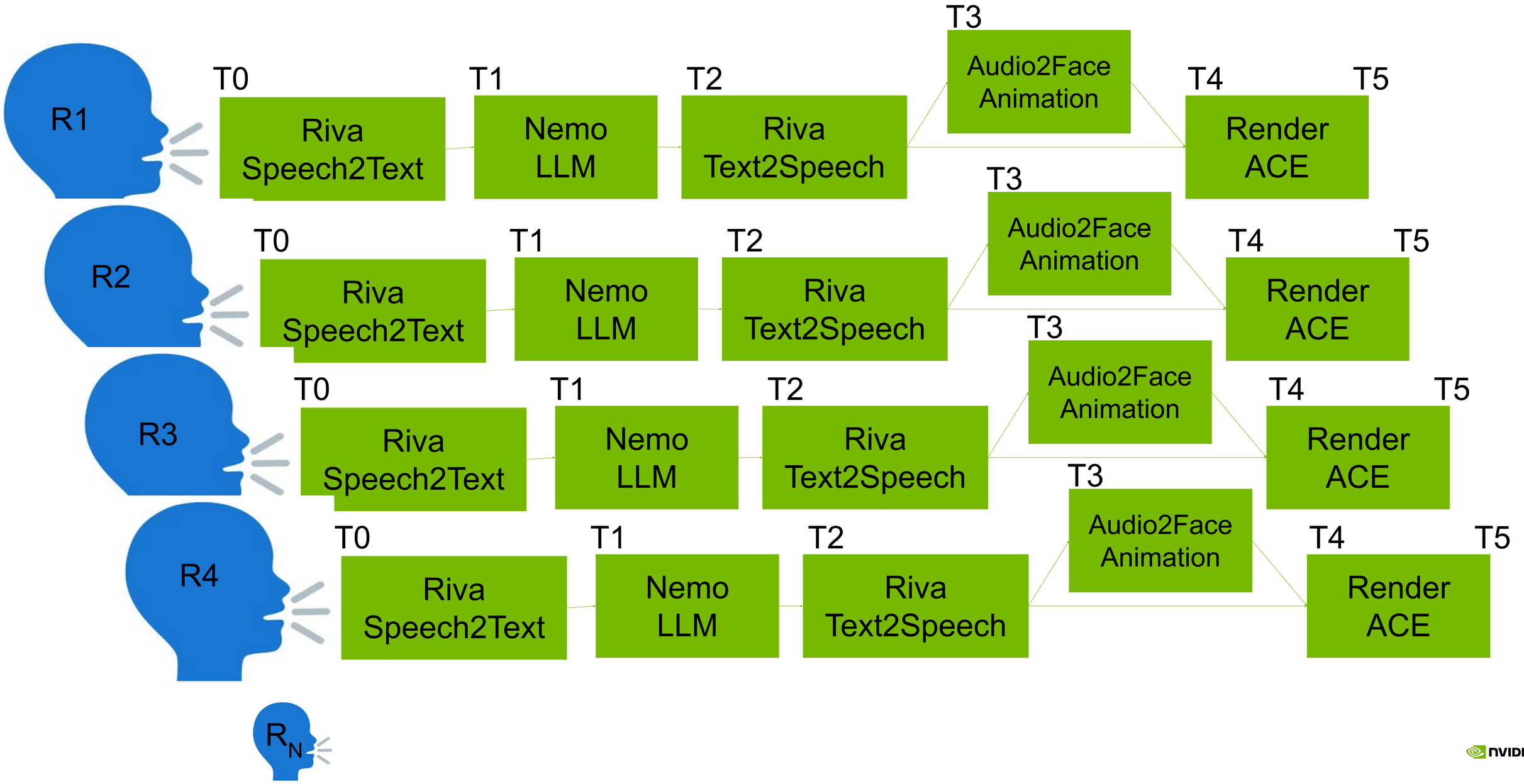
‘My’ cloud application with kubernetes & microservices!

Requests are a parallel pipelined across these stages/microservices



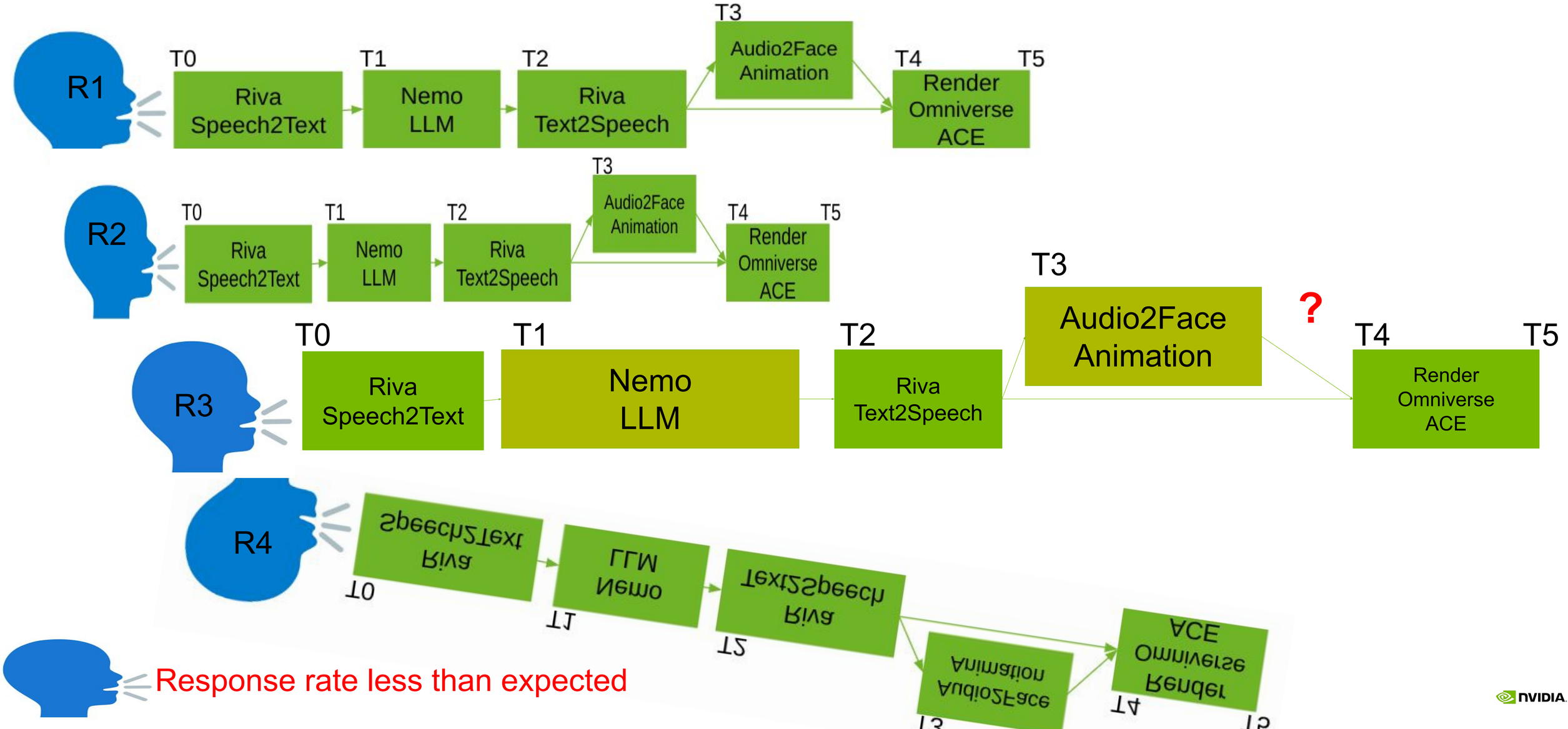
I want it to scale uniformly, consistent, & keep peek GPU performance

Ideally it behaves like this!!!



But what if I'm seeing this?!?! HELP

Find the needle in the haystack and investigate very deep!
Correlate to APIs, workloads, & source code algorithms, not just OpenTelementry



NVIDIA DevTools Kubernetes Services

Deep investigations of container performance

- Very high trace and sample rates vs OpenTelementry
- Injection into Pods/Container
- Without modification of containers & k8/helm specs

Filtering

- Kubernetes labels on namespaces or pods
- Within-container process command-line regex

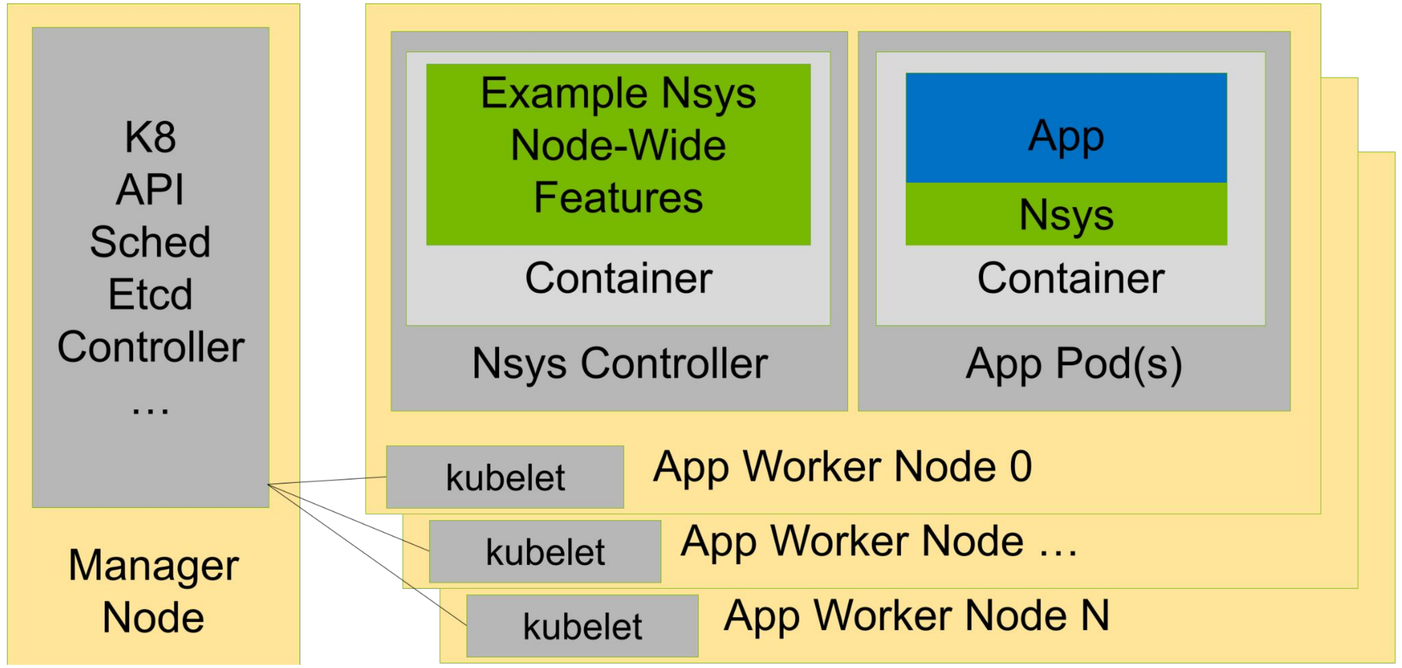
nsys_k8s for extra capabilities

- download without persistent volumes
- Interactive start & stop

Available for CSP managed kubernetes services

Where to get it:

https://catalog.ngc.nvidia.com/orgs/nvidia/teams/devtools/helm-charts/devtools-sidecar-injector



Other Cloud and Remote Server Enhancements

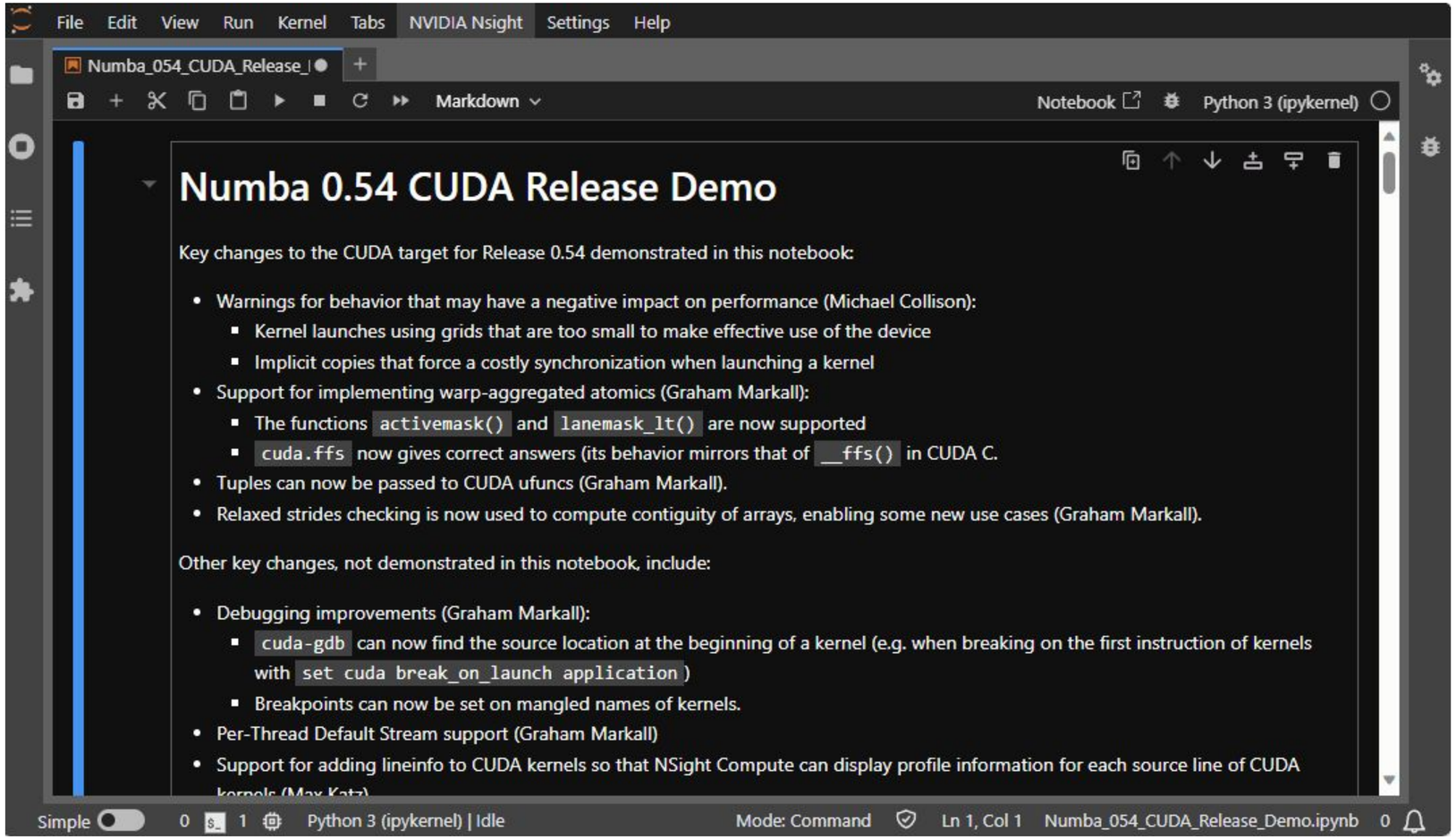
Jupyter Lab

Jupyter Lab Extension

Cell profiling

Launch remote streaming

Change options restarts kernel



Get it:

pip install jupyterlab-nvidia-nsight

https://pypi.org/project/jupyterlab-nvidia-nsight/

Remote UI Streaming

No need to download files

No need to match nsys versions

Access to more RAM vs laptop

Container with X, WM, & WebRTC server

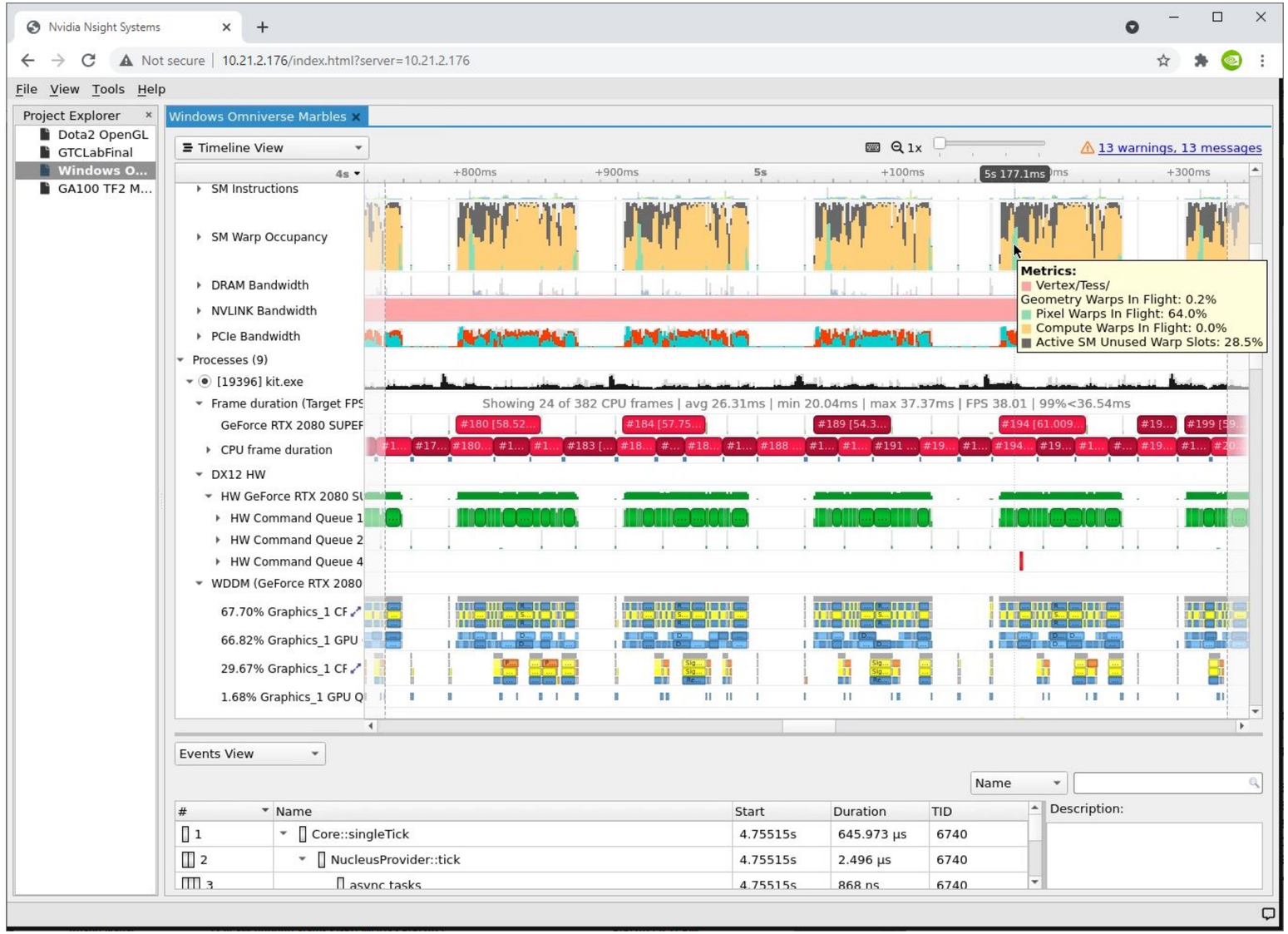
So your server doesn't have to!

Jupyter Lab extension can launch this!

Get it:

Dockerfile inside Nsight Systems Install dir

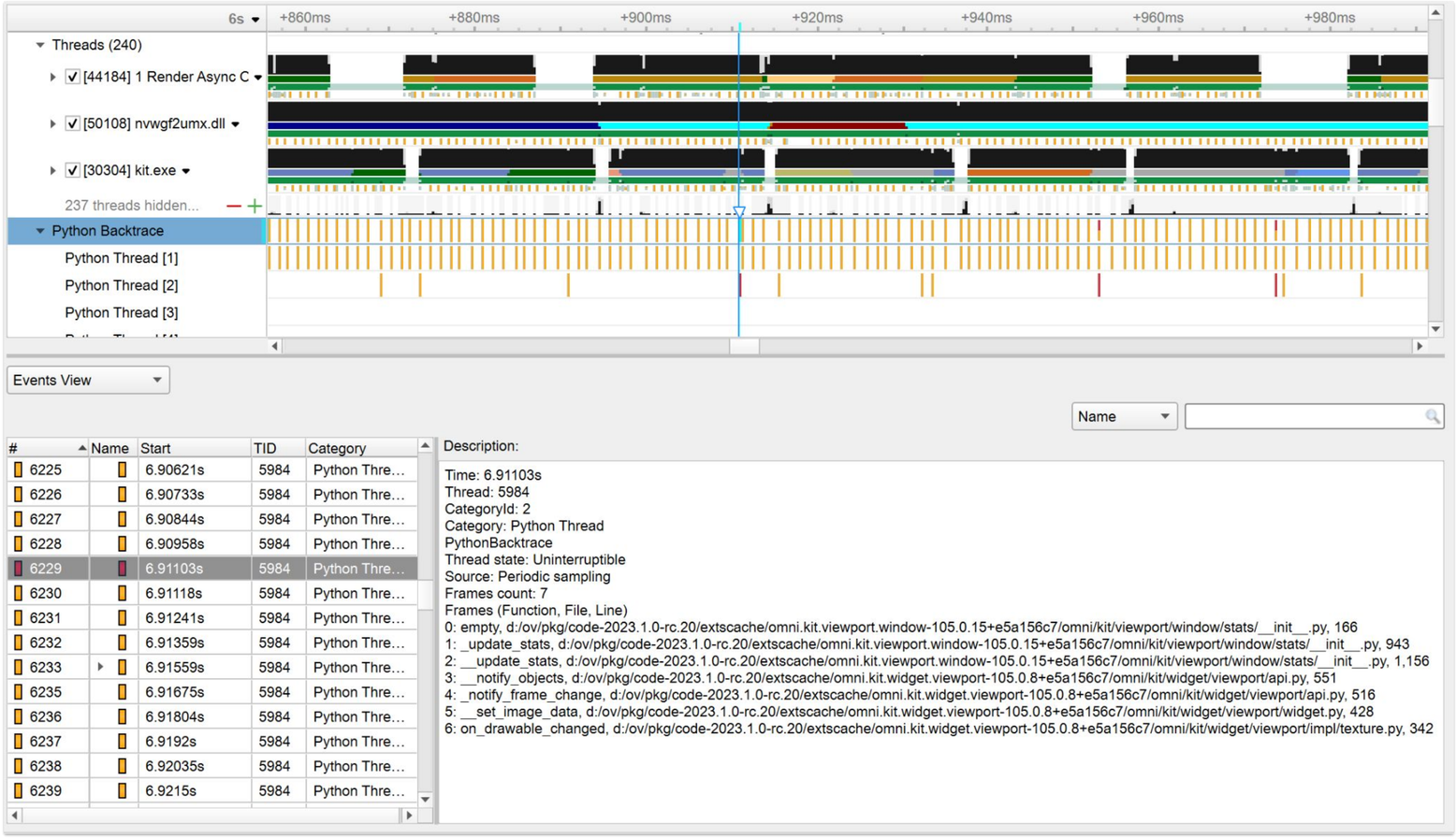
NVIDIA NCG Catalog soon



Python

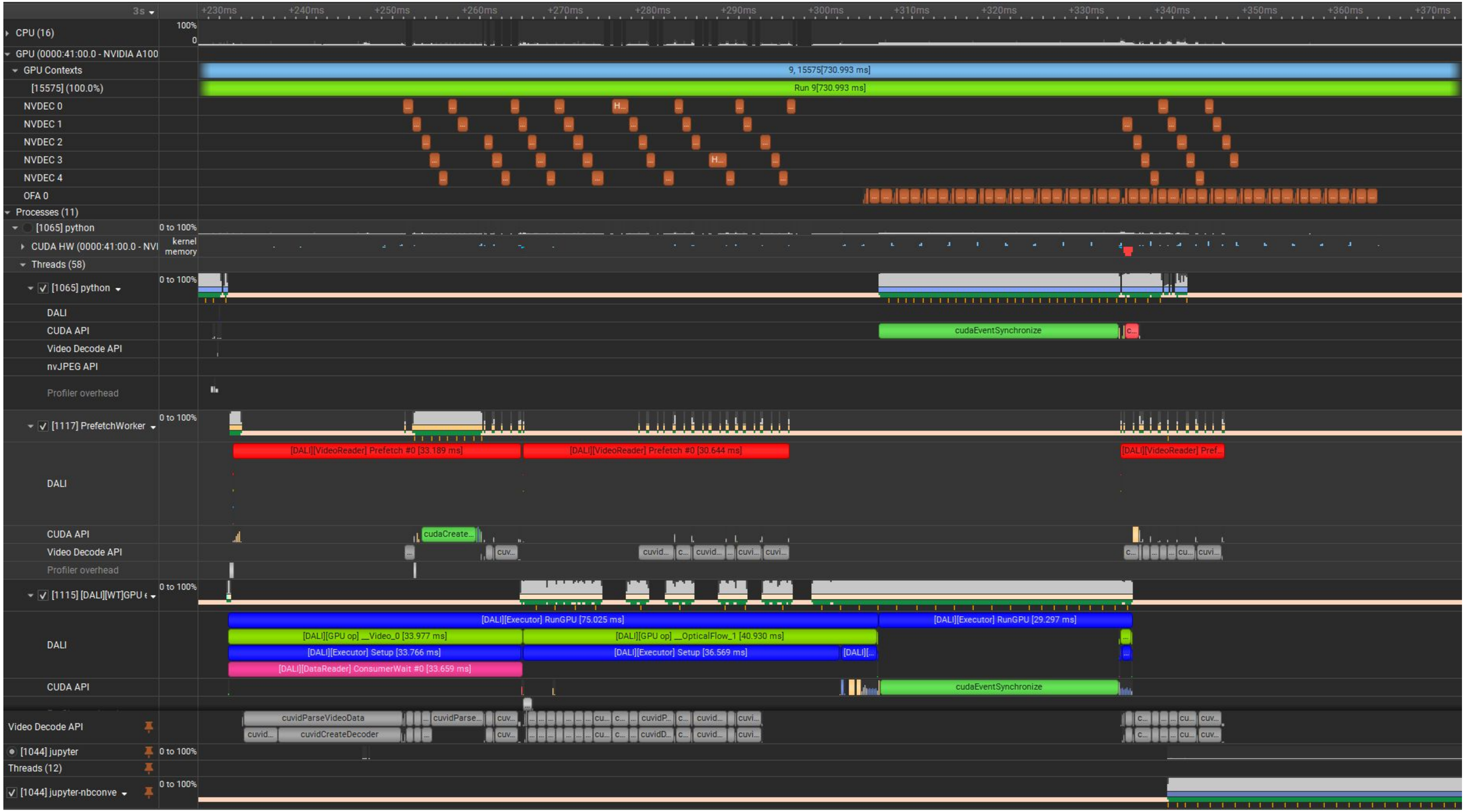
Modern artificial intelligence and data analytics applications

- Call-stack sampling
- GIL trace
- Code annotation library (NVTX)
  - @ decorators
  - ‘With’ statement
  - Download: ‘pip install nvtx’
- Dynamic function trace
  - Config file to target functions



ENC, DEC, OFA, & JPG Hardware Trace

Example NVIDIA Data Loading Library (DALI)

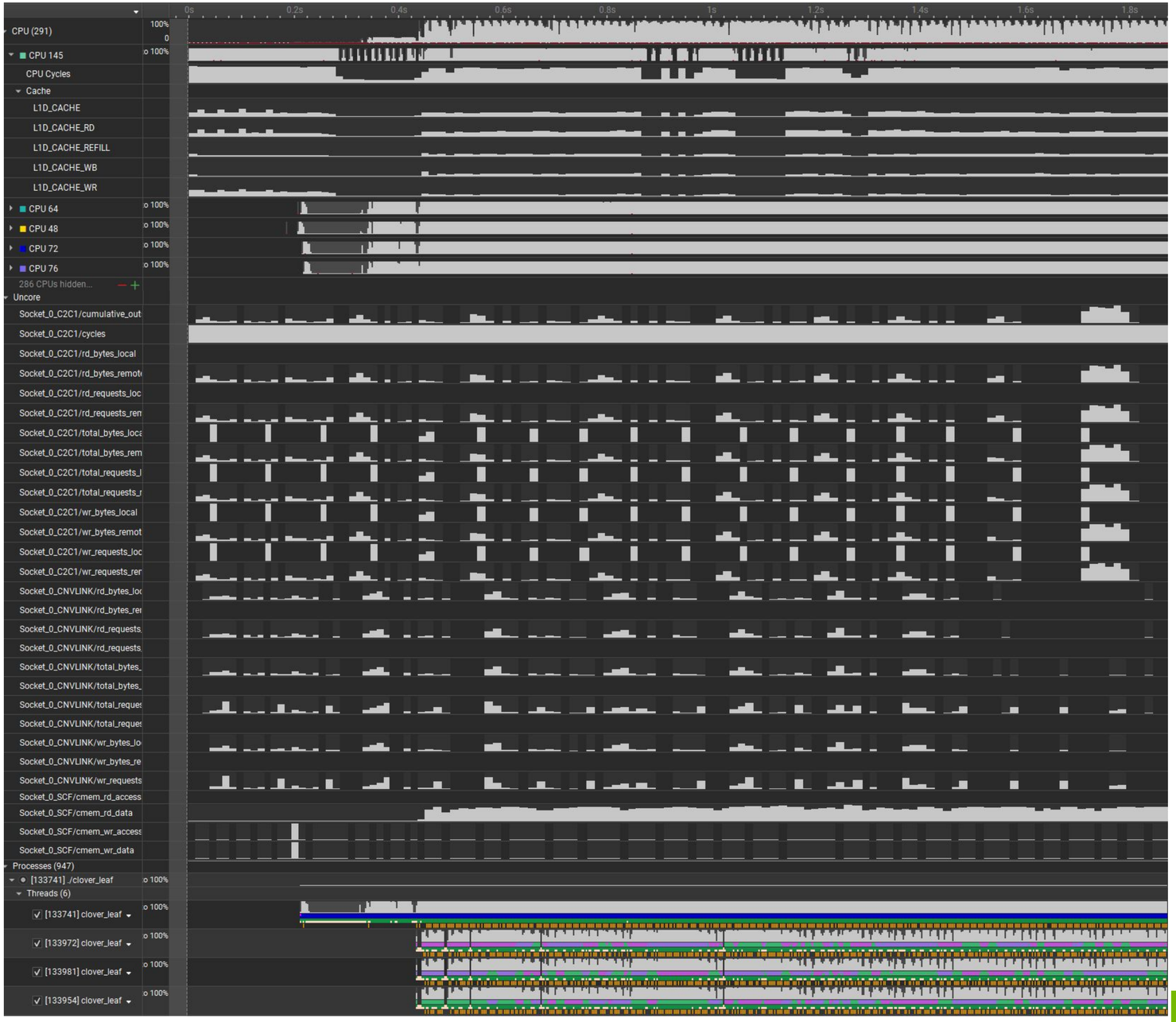


NVIDIA Grace CPU Performance Counters Sampling

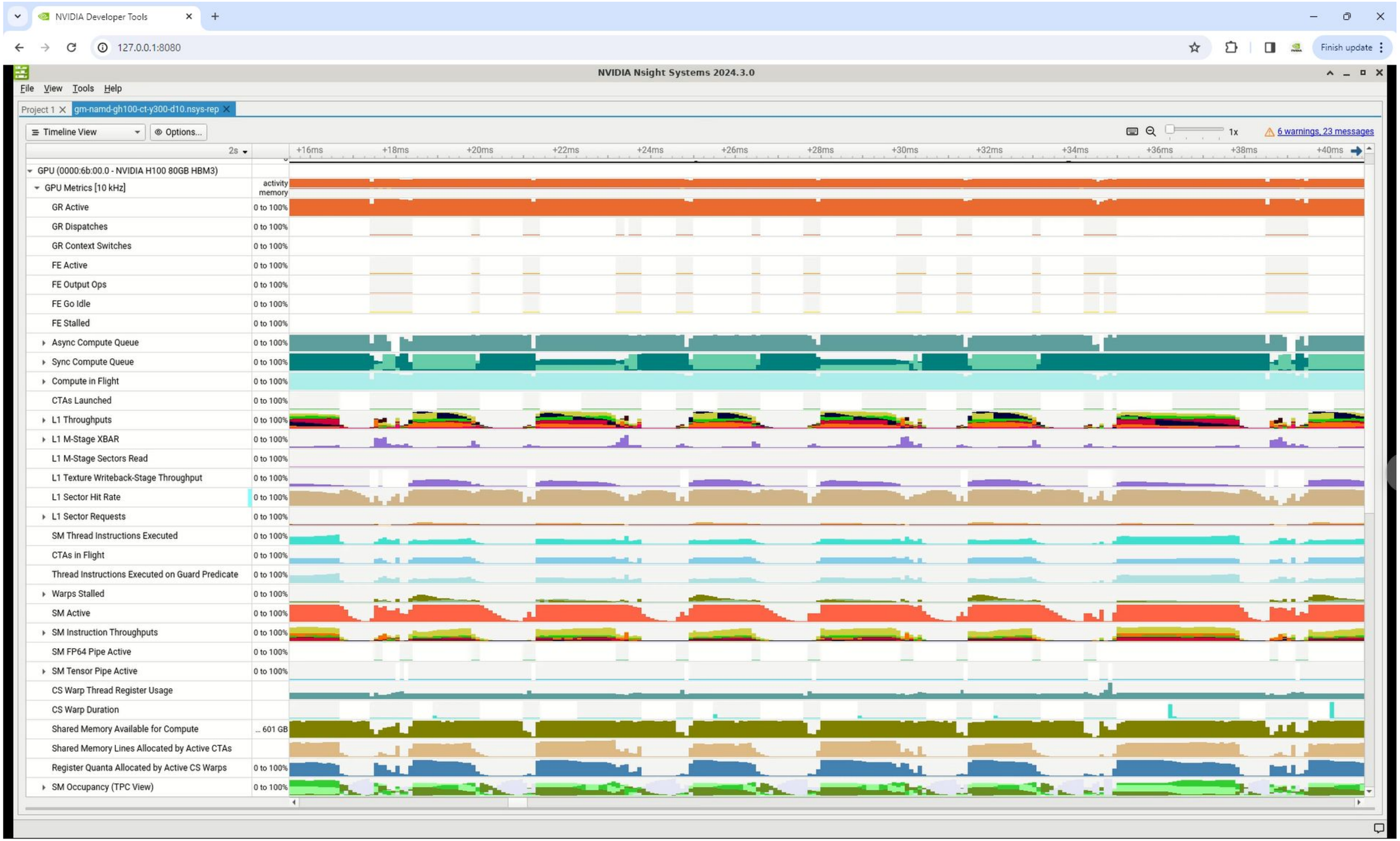
Sampling Grace counters & metrics

See data from

- Cores
- Uncore
- Coherency
- Caches
- Buses
  - Memory
  - NVLink-C2C
- …



H100 Compute Metrics-Set (Coming Soon)



Wrapping up

- Meeting developers where they are developing
- More visibility into critical path & utilization
- Paint the pictures to:
  - Swiftly guide your investigation
  - Ultimately get the most from your hardware

DEVELOPER TOOLS ACROSS GTC

Sessions

- S62388: Achieving Higher Performance From Your Data Center and Cloud Application
- S62256: Demystify CUDA debugging and performance with powerful developer tools
- SE62128: Exploring AI-Assisted Developer Tools for Accelerated Computing
- S62398: Advances in Ray Tracing Developer Tools

Labs

- DLIT61667: Profilers, Python, and Performance: Nsight Tools for Optimizing Modern CUDA Workloads

Connect with the Experts

- CWE61532: What's in Your CUDA Toolbox? CUDA Profiling, Optimization, and Debugging Tools
- CWE61581: Using Nsight Graphics Tools to Transform Your Application to a Next-Gen Powerhouse
- CWE61231: Connect With the Experts: GPU Compute Performance Analysis and Optimizations

Demos

Come and visit the Developer Tools pod during show floor hours!

The latest version of Nsight Systems is available at:

https://developer.nvidia.com/nsight-systems/get-started

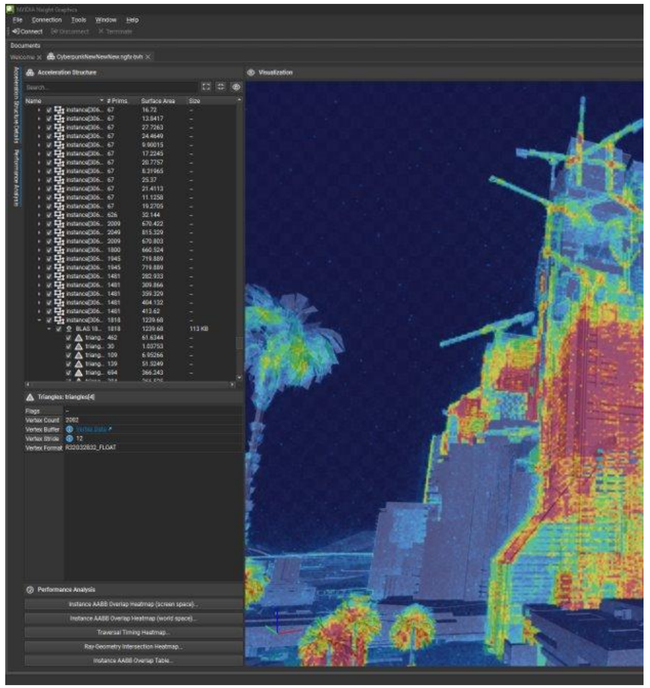
More developer tools at:

https://developer.nvidia.com/tools-overview

Support is available at:

https://forums.developer.nvidia.com/c/developer-tools

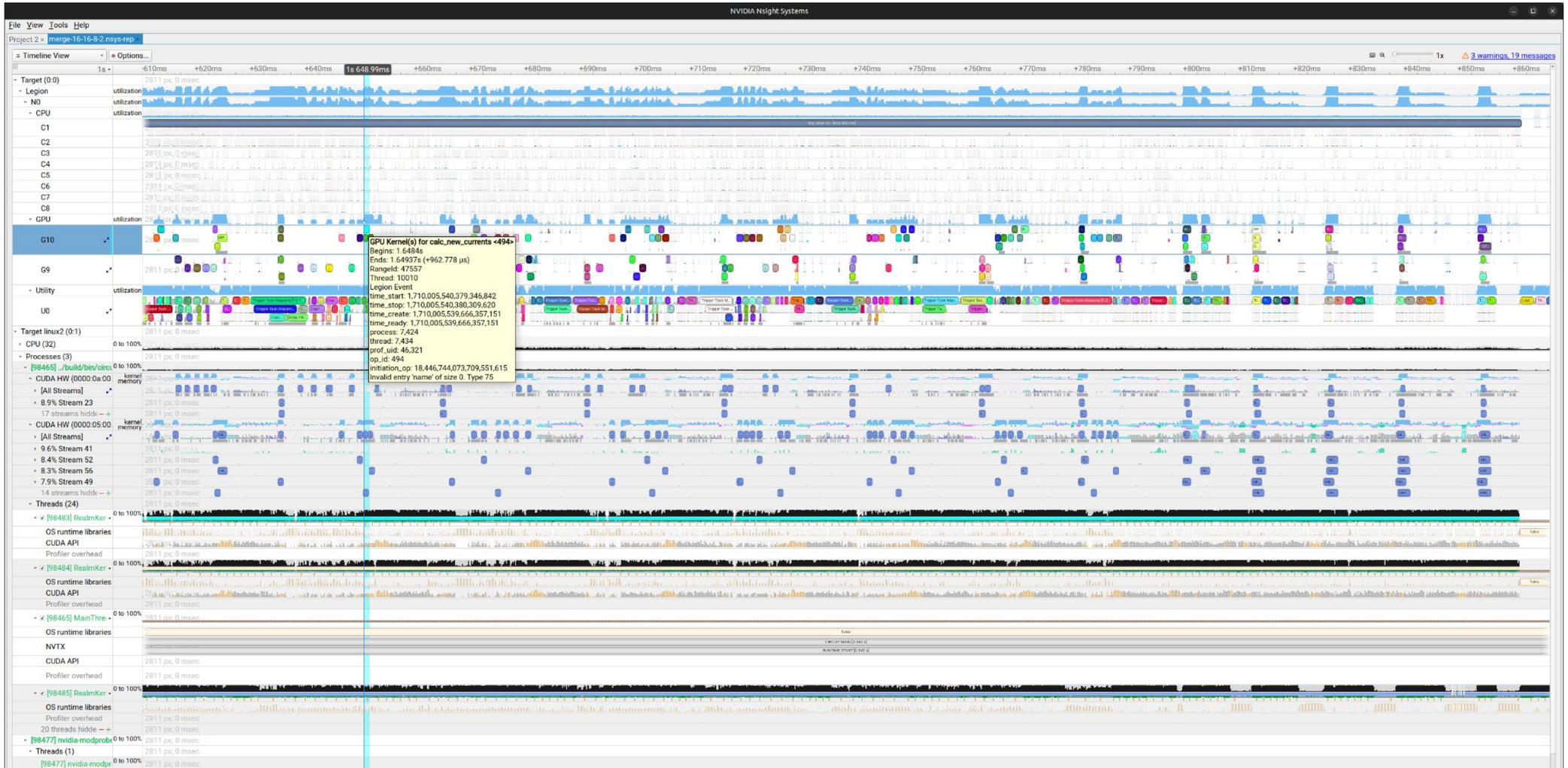
Interested in working on Developer Tools? We are hiring! Scan the QR code 👉



NVIDIA

Legate

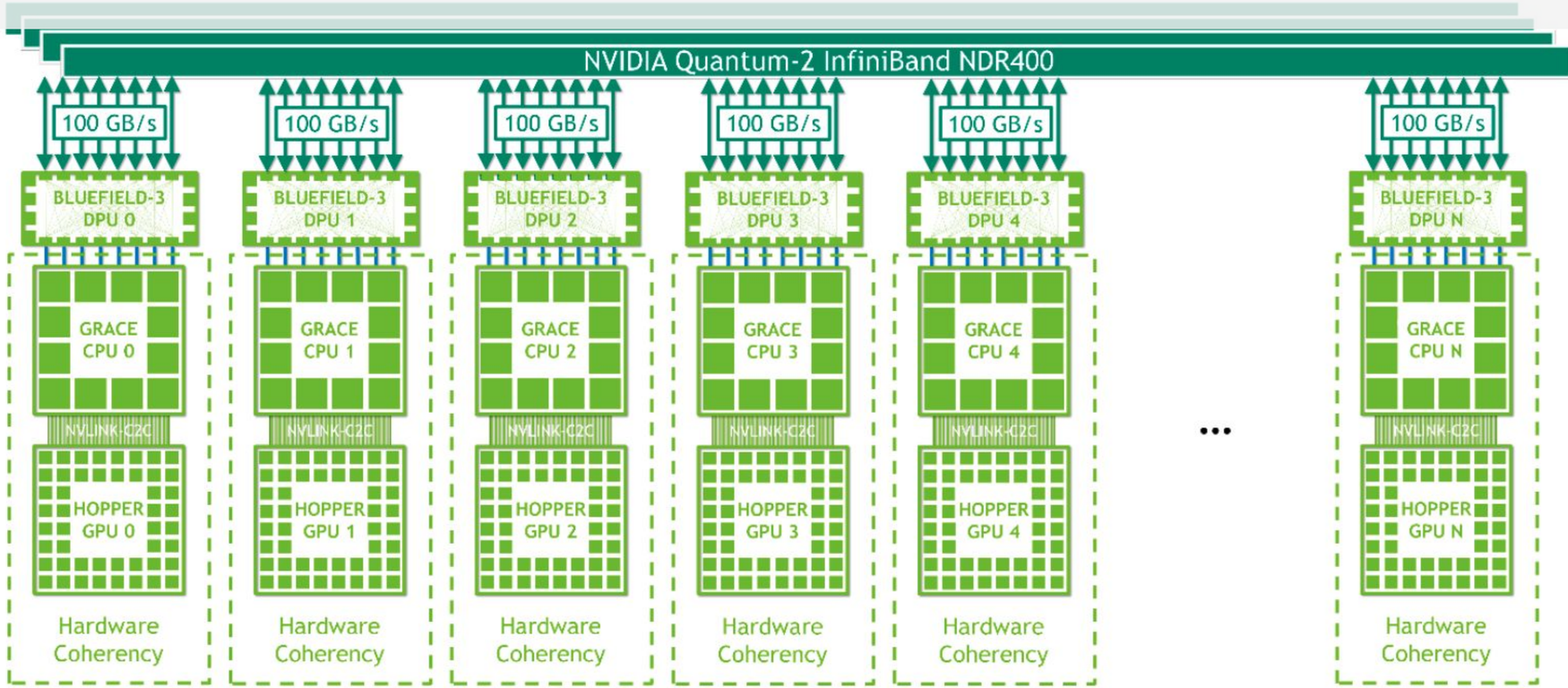
Unified timeline of Legate high-level logical task decomposition alongside low-level hardware, OS, and software details



Logical view:
Tasks,regions,
scheduling,
dependencies

Physical view:
CPU, GPU, NIC,
OS, processes,
threads, libraries
…

Scale to 1000s of GPUs



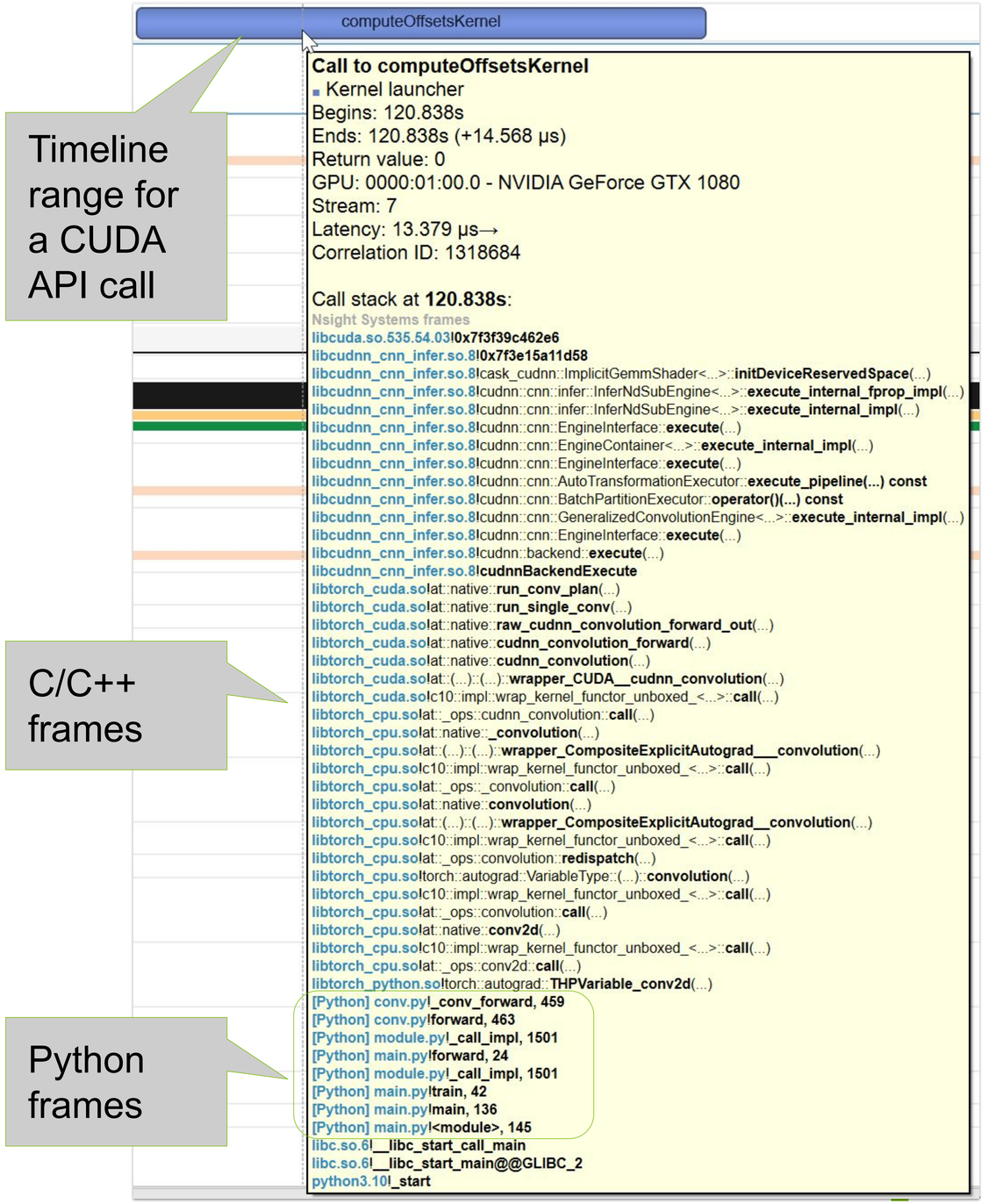
More Resources

- GTC Talks
  - 2023 - Optimizing at Scale: Investigating Hidden Bottlenecks for Multi-Node Workloads
  - 2022 - Optimizing Communication with Nsight Systems Network Profiling
  - 2021 - Tuning GPU Network and Memory Usage in Apache Spark
  - 2020 - Rebalancing the Load: Profile-Guided Optimization of the NAMD Molecular Dynamics Program for Modern GPUs using Nsight Systems
  - 2019 - Using Nsight Tools to Optimize the NAMD Molecular Dynamics Simulation Program
  - 2018 - Optimizing HPC Simulation and Visualization Codes Using NVIDIA Nsight Systems
  - 2018 - Boost DNN Training Performance using NVIDIA Tools
- DLI Labs
  - Fundamentals of Accelerated Computing with CUDA C/C++
  - Optimizing CUDA Machine Learning Codes With Nsight Profiling Tools
- More presentations, videos, blog posts, ...
  - https://docs.nvidia.com/nsight-systems/UserGuide/index.html#other_resources

Python CUDA API Backtrace

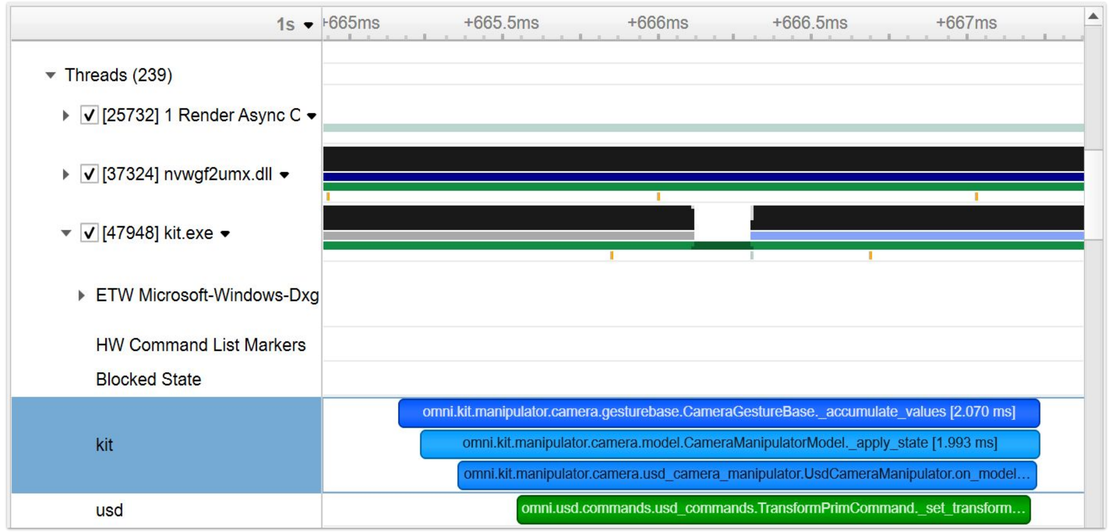
Combined backtrace of C/C++ and Python frames

- Extends the CUDA API Trace feature
- Time threshold
- Notes:
  - Requires CUDA trace
  - Target application must be launched by Nsys



Python Dynamic Annotations

- JSON config file
  - which functions
  - NVTX placement info
- No change to source code!
- Low overhead.
- Notes:
  - Target application must be launched by Nsys




```
[
    {
        "domain": "usd",
        "color": "0x008000",
        "module": "omni.usd.commands.usd_commands",
        "functions": ["TransformPrimCommand._set_transform_matrix"]
    },
    {
        "domain": "kit",
        "module": "omni.kit.manipulator.camera.usd_camera_manipulator",
        "color": "0x0080FF",
        "functions": ["UsdCameraManipulator.on_model_updated"]
    },
    {
        "domain": "kit",
        "module": "omni.kit.manipulator.camera.gesturebase",
        "color": "0x0055FF",
        "functions": ["CameraGestureBase._accumulate_values"]
    },
    {
        "domain": "kit",
        "module": "omni.kit.manipulator.camera.model",
        "color": "0x009CFF",
        "functions": ["CameraManipulatorModel._apply_state"]
    }
]
```

Traditional Job & Resource Scheduler

SLURM

Profile on every task/rank and keep each file unique in network storage

Add special features to one task/rank per node

```
NSYS_RANK_OPT=“ --trace cuda,nvtx … --output /netstorage/%q{SLURM_PROCID}.nsys-rep”
NSYS_NODE_OPT=$(if [ “$SLURM_LOCALID” ==  ‘’0” ]; then echo ”--nic-sample=true --gpu-metrics-device=all”; fi)
NSYS_SESSION=job%q{SLURM JOB_ID}_q{SLURM_LOCALID}

srun nsys profile --session-new $NSYS_SESSION $NSYS_RANK_OPT $NSYS_NODE_OPT <your-app> <args>
```

The yellow above is for **optional** interactive control, paired with features like --start-later, --delay, --duration, etc

```
srun --jobid <jobId> nsys start --session $NSYS_SESSION …
srun --jobid <jobId> nsys stop --session $NSYS_SESSION …
```

If pyxis plugin, slightly simpler session name

```
NSYS_SESSION=job%q{SLURM JOB_ID}
```

RAPIDS & Dask

DaskMPI (or custom SLURM or MPI scripts)

Start on every workers (optionally delayed until manually started)

Example with mpirun, easily translatable to SLURM srun/sbatch

```
mpirun|srun nsys profile <nsys-args> dask-mpi --scheduler-file <schedOnNet>.json
```

If interactive start/stop

Again, add `--start-later` & `--session-new` with SLURM_LOCALID or OMPI_COMM_WORLD_LOCAL_RANK

DaskSSH

Need a helper script to provide a local ID if more than one worker per OS at receiving IP address+port (ie not containerized)

Interactive Start/Stop

Options for coordinated start|stop

a) Dask API: client.run()

b) Tool like ‘parallel-ssh’ & address all local IDs

OpenMPI-Compatible Scheduler

Alternative Environment Variables

Profile on every rank and keep each file unique

```
srun nsys profile --output /netstorage/<user>/<jobname>/$OMPI_COMM_WORLD_RANK_$HOSTNAME.nsys-rep <app> <args>
```

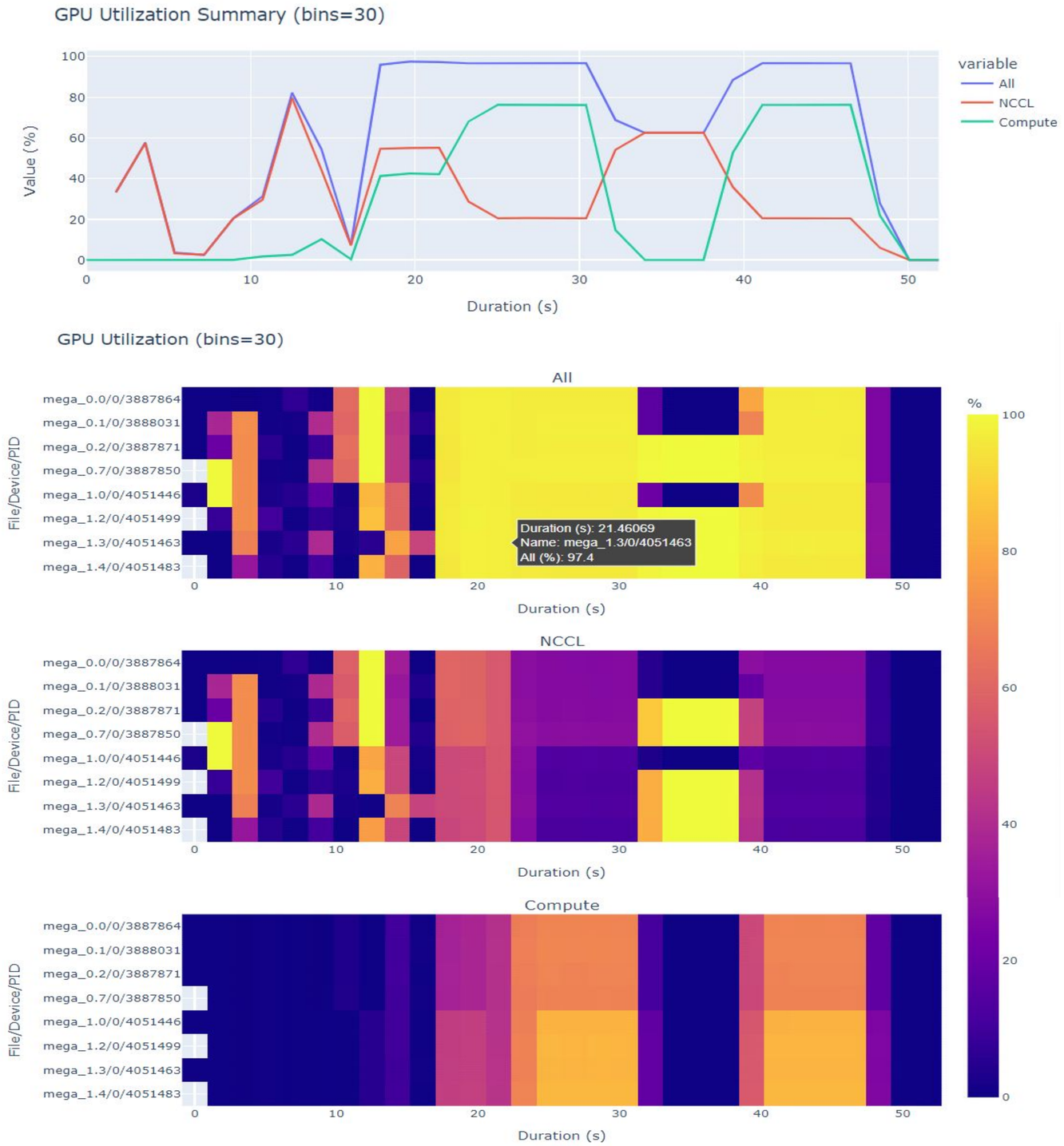
SLURMD_NODEID, SLURMD_NODENAME, SLURM_JOBID, SLURM_JOB_NAME

- Add special features to rank 0

```
if [ “$OMPI_COMM_WORLD_LOCAL_RANK” ==  ‘’0” ]
```

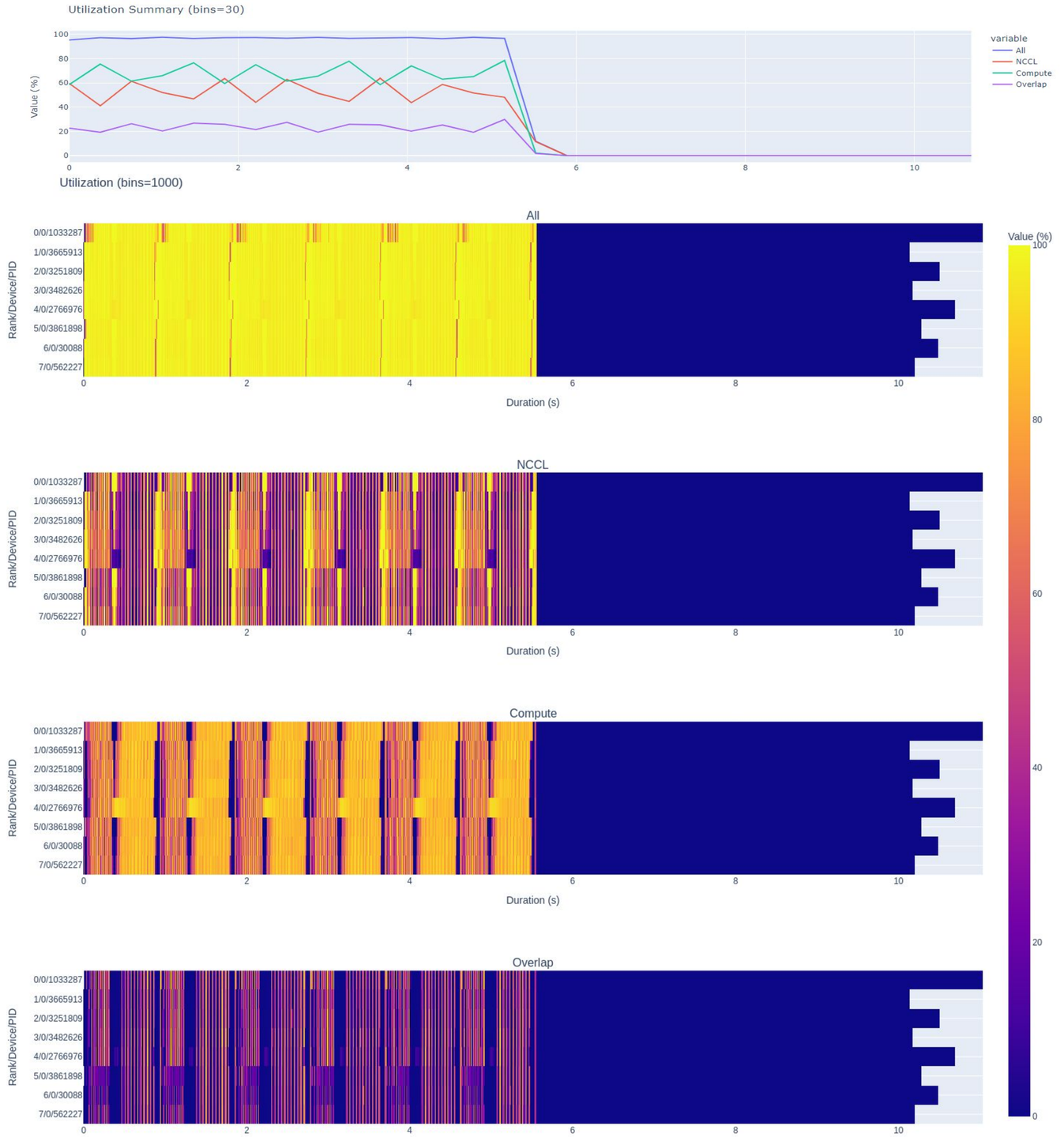
Compute-Only Zones

- GPU Utilization
- Communications
- Compute-Only



Compute-Comms Overlap

- GPU Utilization
- Communications
- Compute
- Overlap



Scaling Out
Multi-Report Analysis

Best Practices for Large Scale Distributed Training

Step by step guides to create clusters:

- One-click VPC Deployments
- Mount FSx for Lustre Filesystems
- EFA enabled clusters

Recipes to customize AMIs

AWS optimized Dockerfiles

EFA cheatsheet

Distributed Training Examples:

- Slurm scripts/K8 manifests
- Working with Pyxis/Enroot
- Nemo (MegatronLM, Multimodal, Bionemo)
- MosaicML
- DDP, FSDP
- SMDP, SMMP
- Tensorflow/Jax

Validation (NCCL Tests etc)

Observability (Prometheus-Grafana etc)

Profiling (Nsight Product Family)

Best Practices for Large Scale Distributed Training

Step by step guides to create clusters:

- One-click VPC Deployments
- Mount FSx for Lustre Filesystems
- EFA enabled clusters

Recipes to customize AMIs

AWS optimized Dockerfiles

EFA cheatsheet

Distributed Training Examples:

- Slurm scripts/K8 manifests
- Working with Pyxis/Enroot
- Nemo (MegatronLM, Multimodal, Bionemo)
- MosaicML
- DDP, FSDP
- SMDP, SMMP
- Tensorflow/Jax

Validation (NCCL Tests etc)

Observability (Prometheus-Grafana etc)

Profiling (Nsight Product Family)