
NVIDIA
Hands-on

必要なファイル、データをコピー

- 講習会で使うハンズオンコードは以下にあります
  - /work/gt00/share/20230621_GPU-bootcamp/hands-on.tar.gz
- Wisteria にログインした後
  - $ cd /work/gt00/tXXXXX *tXXXXX はアカウント名
  - $ cp /work/gt00/share/20230621_GPU-bootcamp/hands-on.tar.gz .
  - $ tar xvzf hands-on.tar.gz
- コンパイルおよび実行のための環境準備
  - $ module load nvidia/23.3

```
hands-on/
|-- cuda
|   |-- dcdread.h
|   |-- Makefile
|   |-- rdf.cu
|   |-- rdf.f90
|   |-- solution
|   |   |-- dcdread.h
|   |   |-- Makefile
|   |   |-- rdf_malloc.cu
|   |   |-- rdf_unified_memory.cu
|   |   |-- rdf_unified_memory.f90
|   |   `-- submit.sh
|   |-- vector_addition_gpu_block_only.cu
|   |-- vector_addition_gpu_thread_block.cu
|   `-- vector_addition_gpu_thread_only.cu
|-- input
|   `-- alk.traj.dcd
|-- iso
|   |-- dcdread.h
|   |-- Makefile
|   |-- rdf.cpp
|   |-- rdf.f90
|   `-- solution
|       |-- dcdread.h
|       |-- Makefile
|       |-- rdf.cpp
|       |-- rdf.f90
|       `-- submit.sh
`-- openacc
    |-- dcdread.h
    |-- Makefile
    |-- rdf.cpp
    |-- rdf.f90
    `-- solution
        |-- data
        |   |-- dcdread.h
        |   |-- Makefile
        |   |-- rdf_data_directive.cpp
        |   |-- rdf_data_directive.f90
        |   `-- submit.sh
        |-- kernel
        |   |-- dcdread.h
        |   |-- Makefile
        |   |-- rdf_kernel_directive.cpp
        |   |-- rdf_kernel_directive.f90
        |   `-- submit.sh
        |-- multicore
        |   |-- dcdread.h
        |   |-- Makefile
        |   |-- rdf.cpp
        |   |-- rdf.f90
        |   `-- submit.sh
        `-- parallel
            |-- dcdread.h
            |-- Makefile
            |-- rdf.cpp
            |-- rdf.f90
            `-- submit.sh
```

ハンズオン

- C++ or Fortran を選択し、以下の教材に沿ってハンズオンを実施
  - ISO standard: https://github.com/openhackathons-org/nways_accelerated_programming/blob/df3d3ceec17bd4b483e10c2ee93ab2a6d57ebbe5/_basic/iso/jupyter_notebook/nways_iso.ipynb
  - OpenACC: https://github.com/openhackathons-org/nways_accelerated_programming/blob/df3d3ceec17bd4b483e10c2ee93ab2a6d57ebbe5/_basic/openacc/jupyter_notebook/nways_openacc.ipynb
  - CUDA: https://github.com/openhackathons-org/nways_accelerated_programming/blob/df3d3ceec17bd4b483e10c2ee93ab2a6d57ebbe5/_basic/cuda/jupyter_notebook/nways_cuda.ipynb
- 教材を読み進める
- ハンズオンコードは、上記 jupyter-notebook 中のリンクを開くのではなく、Wisteria 上でコピーしたものを使用しビルド、実行
- 正しい結果を得られるようにコードを修正

Nsight Systems のローカル端末へのインストール

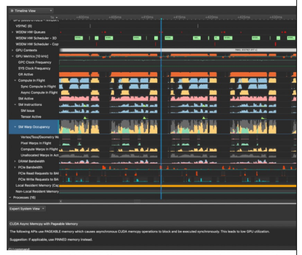
Win/Mac 用も

NVIDIA Nsight Systems

NVIDIA Nsight™ Systems is a system-wide performance analysis tool designed to visualize an application's algorithms, help you identify the largest opportunities to optimize, and tune to scale efficiently across any quantity or size of CPUs and GPUs, from large servers to our smallest system on a chip (SoC).

Get started

Nsight Systems 2022.5 is available



Learn how Nsight Graphics can be used to accelerate d
high-performance games with beautifu

Getting Started with Nsight Systems

Download
- Supported Platforms
- System Requirements

Latest version for:
- Windows & Linux
- Jetson
- DRIVE
- Apple macOS
- Release Notes
- Feature Table
- Archives

Resources
- Documentation
- Tutorial Sessions

Windows & Linux Servers, Workstations, and Gaming PCs: Download Nsight Systems 2023.1

These packages and documents are for local and remote profiling of Windows & Linux Servers, Workstations, and Gaming PCs. Profiling is supported on the following target architectures: x86-64 & arm-SBSA. See "Supported Platforms" for specifics about combinations of local, remote, and mixed-OS compatibilities.

NOTE1: Apple macOS is only supported
these devices, and report viewer.

NOTE2: NVIDIA Tegra-based devices suc
supported through the variants of Nsigh
respective SDK. See "Supported Platforn
packages are not guaranteed to be a co
generated on those platforms.

Download | Documentation

Filter...

- GAMEWORKS
- TARGET
- PHYSX SDK
- GRAPHICS API
- DEVELOPMENT PLATFORM
- APPLICATION
- ACTIVITY
- ADDITIONAL

Showing 490 downloads.

Title	Version	Release Date
Nsight Systems	2023.1	2023/01/31

NVIDIA Nsight Systems is a system-wide performance analysis tool designed to visualize an application's algorithm, help you select the largest opportunities to optimize, and tune to scale efficiently across any quantity of CPUs and GPUs in your computer; from laptops to DGX servers.

Nsight Systems is part of the powerful debugging and profiling NVIDIA Nsight Tools Suite. A developer can start with Nsight Systems for an overall system view and avoid picking less efficient optimizations based on assumptions and false-positive indicators.

Nsight Systems 2023.1 highlights include:

- UX and performance improvements

It can be run from a Windows, Linux, or macOS host system to analyze a remote Linux or Windows x86_64 target. It can be run on a Windows or Linux x86_64 system to analyze the local system. The CLI can be used to generate results on Linux on x86_64, Arm Server or Power. Those results can be visualized in the Linux, Windows, or macOS host.

For more information on Nsight Systems and its requirements, please visit the Nsight Systems overview page.

More Information ›

DOWNLOADS
- Nsight Systems 2023.1.1 (Linux Host .deb Installer)
- Nsight Systems 2023.1.1 (Linux Host .rpm Installer)
- Nsight Systems 2023.1.1 (Linux CLI only .deb Installer)
- Nsight Systems 2023.1.1 (Linux CLI only .rpm Installer)
- Nsight Systems 2023.1.1 (Linux Host .run Installer)
- Nsight Systems 2023.1.1 (Windows Host)
- Nsight Systems 2023.1.1 (macOS Host)
- Nsight Systems 2023.1.1 (Power CLI only .deb Installer)
- Nsight Systems 2023.1.1 (P CLI only .rpm Installer)
- Nsight Systems 2023.1.1 (A Server Full .deb Installer)
- Nsight Systems 2023.1.1 (A Server CLI only .deb Installer)

Nsight Systems 2023.1.1 (Windows Host)

Nsight Systems 2023.1.1 (macOS Host)

NVIDIA

Welcome

Connect with **millions of like-minded developers, researchers, and innovators.**

Accelerate your apps with **the latest tools** and **150+ SDKs.**

Receive **technical training** and **expert help.**

Log in or sign up for an NVIDIA account

You need to sign in or sign up before continuing.

Email

Next

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

Nsight Systems 101

Quick start

- Nsight Systems CLI によるプロファイリング

```
$ nsys profile [options] <application> [application-arguments]
```

  - `-t <parameters>` : トレースする API を指定。デフォルトは、`cuda, opengl nvtx, osrt`
  - `--stats <true|false>` : `true` でプログラム実行時の統計情報を標準出力に表示
  - `-o <filename>` : 出力ファイル名を指定
  - `--force-overwrite <true|false>` : `true` で出力ファイルの上書きを許可。デフォルトは `false`

  など... 詳細は、`nsys --help or nsys [specific command] --help` で確認可能

  - `<filename>.nsys-rep` が出力される
- ローカルに `<filename>.nsys-rep` を転送し、Nsight Systems UI で可視化

Nsight Systems user guide:
https://docs.nvidia.com/nsight-systems/UserGuide/index.html

Nsight Systems 101

Example

- 実行コマンド例

```
$ nsys profile  --stats true --force-overwrite true -o my_report python solution.py
```

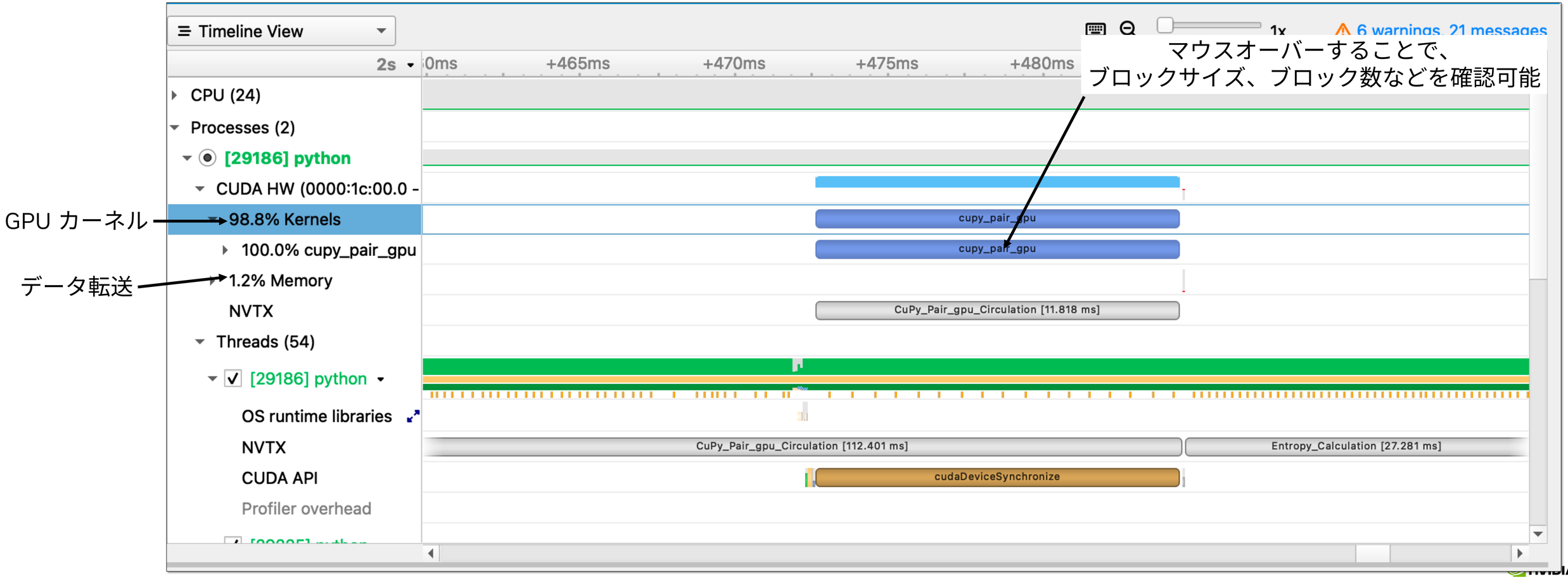
- 統計情報が標準出力に出力される

```
NVTX Range Statistics:

 Time (%)  Total Time (ns)  Instances     Avg (ns)         Med (ns)        Min (ns)       Max (ns)     StdDev (ns)   Style            Range
 --------  ---------------  ---------  ---------------  ---------------  -------------  -------------  -----------  -------  -------------------------
     87.9    1,015,559,769          1  1,015,559,769.0  1,015,559,769.0  1,015,559,769  1,015,559,769          0.0  PushPop  Read_File
      9.7      112,401,039          1    112,401,039.0    112,401,039.0    112,401,039    112,401,039          0.0  PushPop  CuPy_Pair_gpu_Circulation
      2.4       27,280,924          1     27,280,924.0     27,280,924.0     27,280,924     27,280,924          0.0  PushPop  Entropy_Calculation

...

[5/8] Executing 'cudaapisum' stats report

CUDA API Statistics:

 Time (%)  Total Time (ns)  Num Calls    Avg (ns)      Med (ns)     Min (ns)    Max (ns)     StdDev (ns)              Name
 --------  ---------------  ---------  ------------  ------------  ----------  ----------  -------------  ----------------------------
     94.3      219,106,094          4  54,776,523.5       8,807.5       6,978  219,081,501  109,536,651.7  cudaMalloc
      5.1       11,788,845          1  11,788,845.0  11,788,845.0  11,788,845   11,788,845            0.0  cudaDeviceSynchronize
      0.4          958,943          2     479,471.5     479,471.5       6,294      952,649      669,174.0  cudaHostAlloc
      0.0          106,326          1     106,326.0     106,326.0     106,326      106,326            0.0  cuModuleLoadData
      0.0           67,470          1      67,470.0      67,470.0      67,470       67,470            0.0  cuModuleUnload
...

[6/8] Executing 'gpukernsum' stats report

CUDA Kernel Statistics:

 Time (%)  Total Time (ns)  Instances    Avg (ns)      Med (ns)     Min (ns)    Max (ns)   StdDev (ns)      Name
 --------  ---------------  ---------  ------------  ------------  ----------  ----------  -----------  ------------
    100.0       11,817,865          1  11,817,865.0  11,817,865.0  11,817,865  11,817,865          0.0  cupy_pair_gpu

[7/8] Executing 'gpumemtimesum' stats report

CUDA Memory Operation Statistics (by time):

 Time (%)  Total Time (ns)  Count  Avg (ns)  Med (ns)  Min (ns)  Max (ns)  StdDev (ns)      Operation
 --------  ---------------  -----  --------  --------  --------  --------  -----------  ------------------
     98.1          143,295      4  35,823.8  46,207.5     3,232    47,648     21,738.5  [CUDA memcpy HtoD]
      1.9            2,784      1   2,784.0   2,784.0     2,784     2,784          0.0  [CUDA memcpy DtoH]

[8/8] Executing 'gpumemsizesum' stats report

CUDA Memory Operation Statistics (by size):

 Total (MB)  Count  Avg (MB)  Med (MB)  Min (MB)  Max (MB)  StdDev (MB)      Operation
 ----------  -----  --------  --------  --------  --------  -----------  ------------------
      1.629      4     0.407     0.538     0.016     0.538        0.261  [CUDA memcpy HtoD]
      0.016      1     0.016     0.016     0.016     0.016        0.000  [CUDA memcpy DtoH]
```

Nsight Systems 101

Example

- *.nsys-rep を手元の端末に転送
- 手元の端末で Nsight Systems を起動し、File->Open で *nsys-rep ファイルを選択




NVIDIA