

# GPU Performance Analysis

Dominik Ernst



# Example: 2D Jacobi

Get the Source:

```
git clone https://github.com/te42kyfo/omp_jacobi
```

Load the compiler module

```
module load nvhpc
```

Build and run the CPU base line

```
make main1  
./main1
```

Run with likwid-pin

```
module load likwid  
likwid-pin ./main1
```

# Analysis: 2D Jacobi

3 ADDs, 1 MUL per iteration:

$$A[o] = 0.25 * (B[^] + B[v] + B[<] + B[>])$$

Read entire grid A and B once each → on average: one value per iteration

$$= 2 \times 8B / \text{iteration}$$

Code Intensity:

$$4 \text{ Flop} / 16 \text{ B} = 0.25 \text{ Flop/B}$$

A100 Machine Intensity:

$$9.7 \text{ Tflop/s} / 1555 \text{ GB/s} = 6.2 \text{ Flop/B}$$

# Build/Run/Profile

Build and run the Nth version

```
make main<N>  
./main<N>
```

Create a profile

```
nsys profile main<N>
```

Launch the profiling GUI

```
nsys-ui
```

# Kernel Profiling

## Kernel profiling

```
ncu <application>
```

## List metric sections

```
ncu --list-sections
```

## Collect all sections

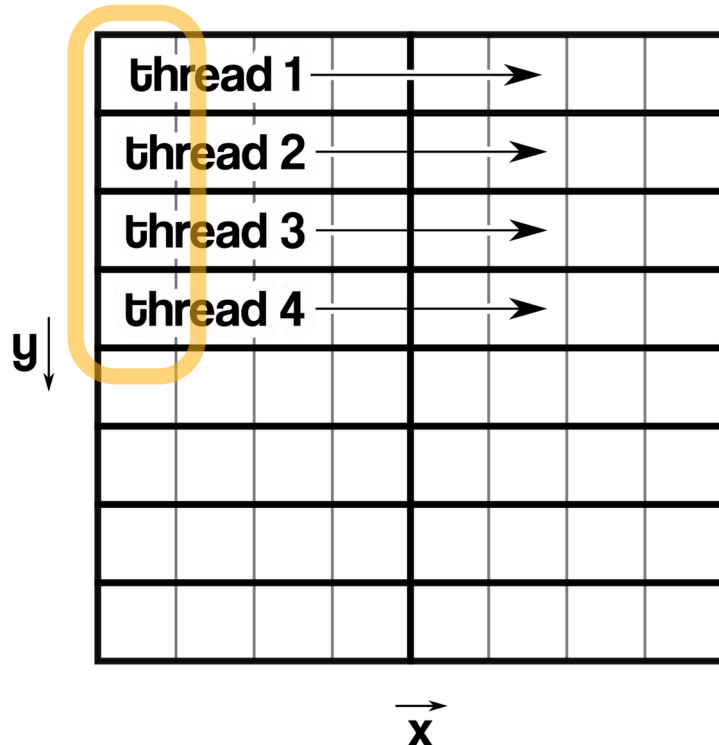
```
ncu --set full -f -o <output file> <application>
```

## Launch the ncu profiling GUI

```
ncu-ui
```

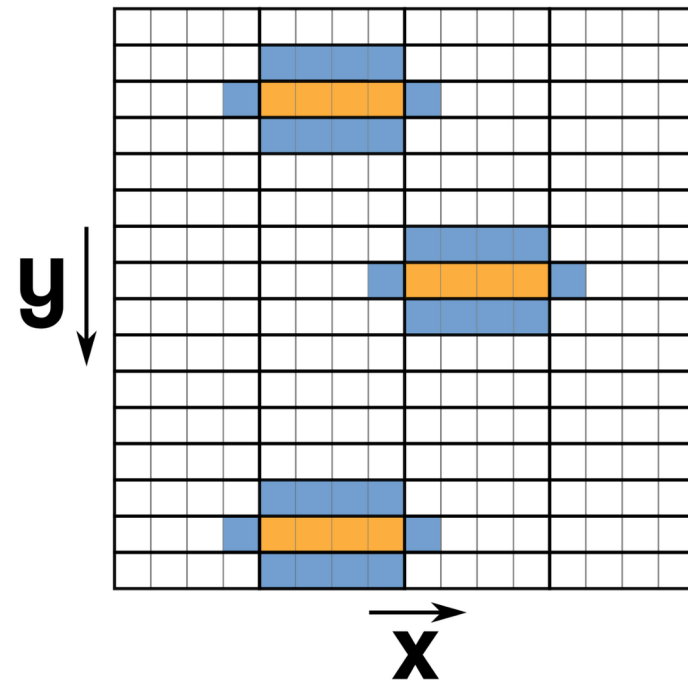
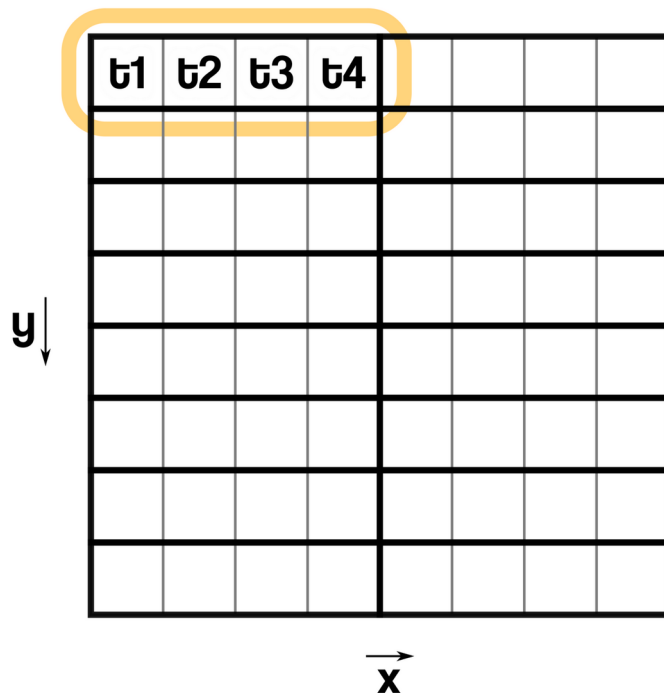
# OpenMP Loop Parallelization #1

```
#pragma omp target parallel for  
for (int y = 1; y < height - 1; y++)  
    for (int x = 1; x < width - 1; x++)  
        ...
```



# OpenMP Loop Parallelization #2

```
#pragma omp target parallel for collapse(2)
for (int y = 1; y < height - 1; y++)
    for (int x = 1; x < width - 1; x++)
        ...
```



# OpenMP Loop Parallelization #3

```
#pragma omp target parallel for collapse(2)
for (int oy = 1; oy < height - 1; oy += 4)
  for (int x = 1; x < width - 1; x++)
    for (int iy = 0; iy < 4; iy++) {
      int y = oy + iy;
```

