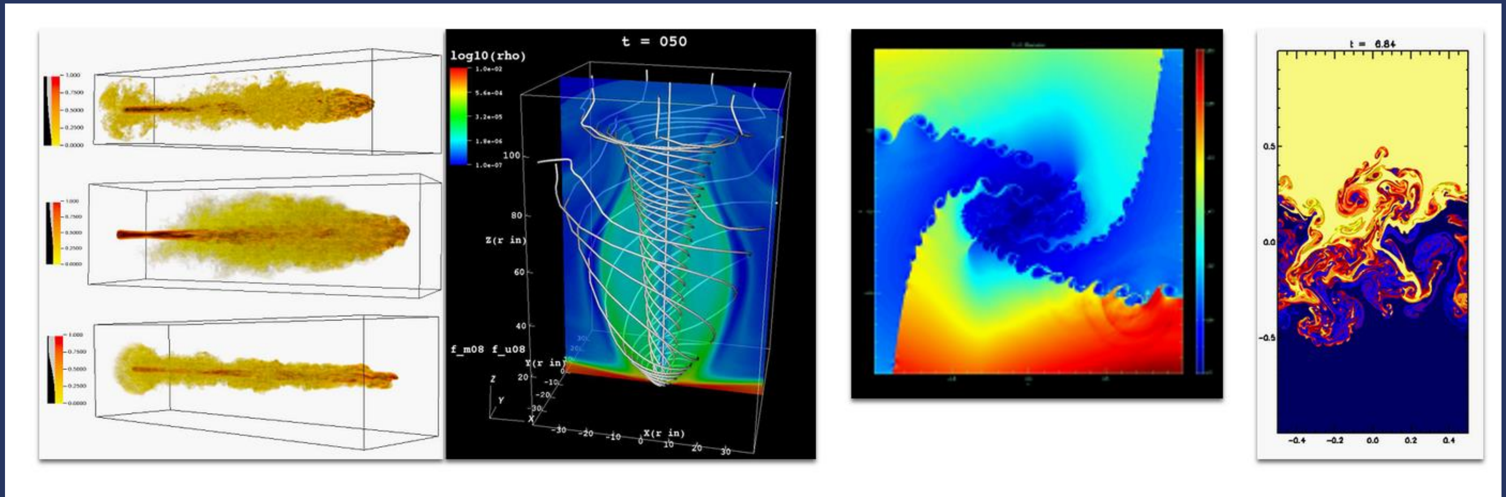
GPUs for PLUTO

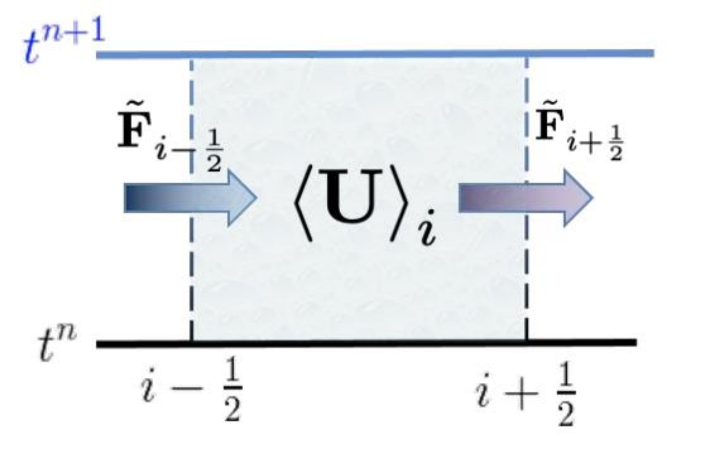
GPU-porting of the PLUTO code for Computational Plasma Physics

Marco Rossazza

supervisor: Andrea Mignone

PLUTO



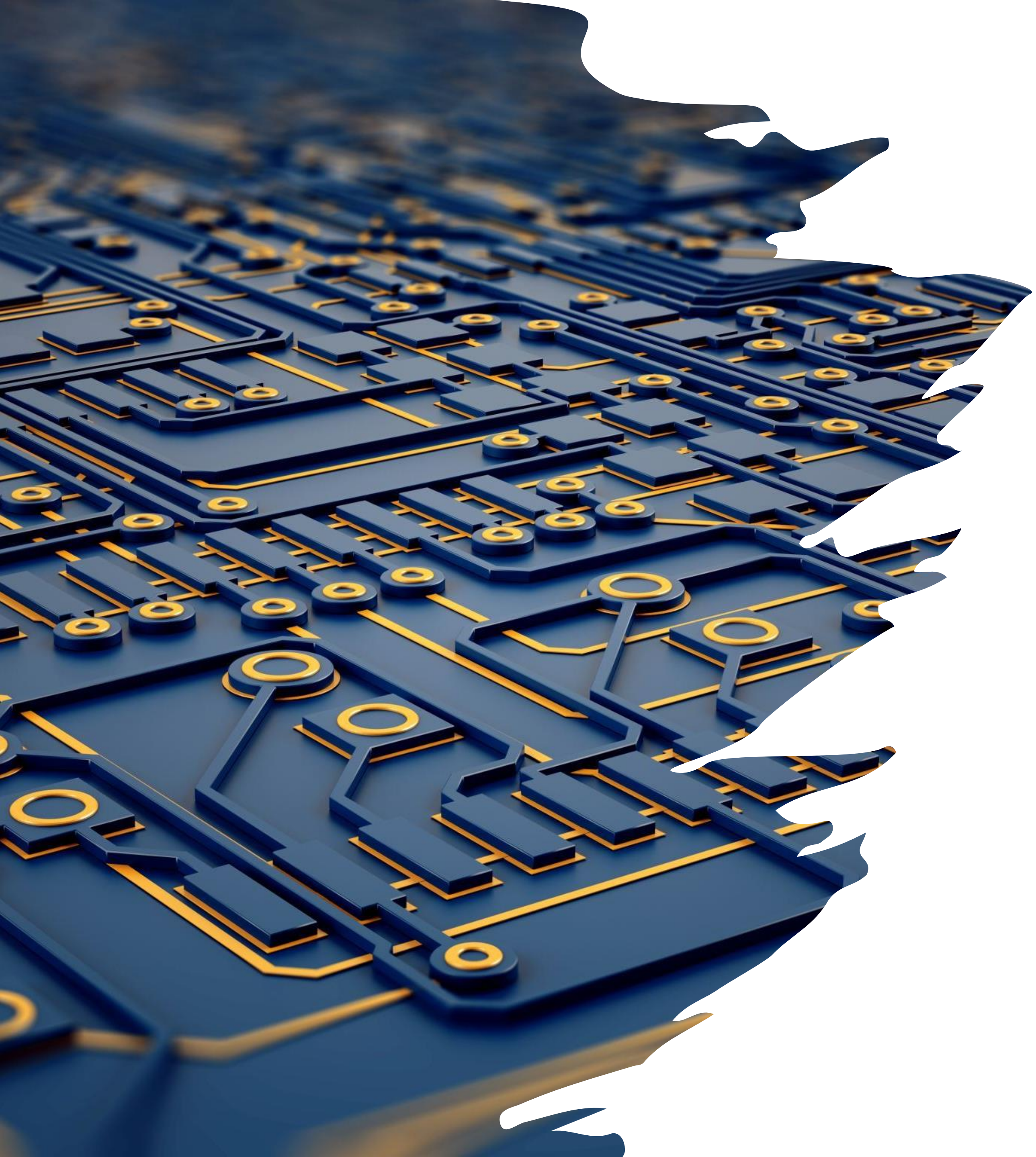


t^{n+1}
$\tilde{\mathbf{F}}_{i-\frac{1}{2}}$
$\langle \mathbf{U} \rangle_i$
$\tilde{\mathbf{F}}_{i+\frac{1}{2}}$
t^n
$i-\frac{1}{2}$
$i+\frac{1}{2}$

PLUTO is a modular parallel code providing a multi-physics and multi-algorithm framework for solving the equations of gas and plasma dynamics in astrophysics.

Grid-based, finite volume code relying on a Reconstruct-Solve-Average (RSA) strategy.

http://plutocode.ph.unito.it/



GPU-PORTING

GPU-porting refers to the process of adapting or optimizing a software application to run efficiently on GPUs.

GPUs typically offer higher computational efficiency compared to CPUs, meaning they can achieve more computations per power consumed.

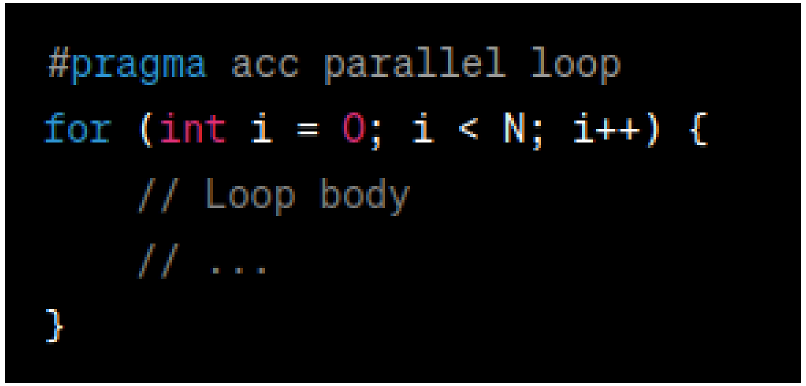
The ultimate reason to recur to GPUs is that they can lead to significantly faster computations.

OPENACC MORE SCIENCE, LESS PROGRAMMING

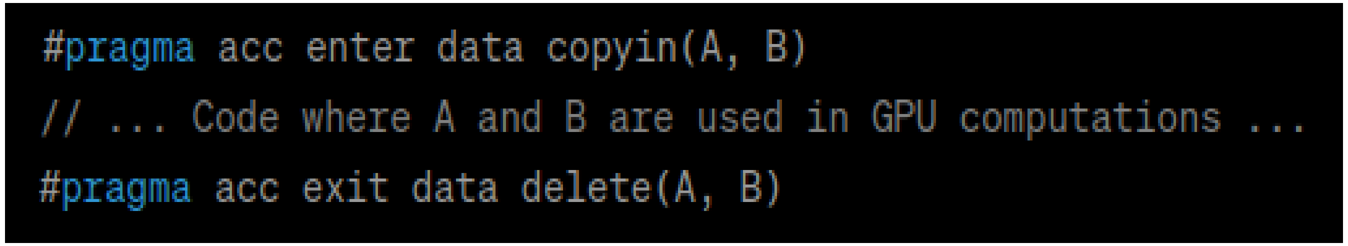
OPENACC is the programming model chosen for the GPU-porting.

- High-level
- Requires few changes to the code
- Directive-based

The pragma acc parallel loop is a directive used in OpenACC to indicate that a loop can be parallelized and executed in parallel on the GPU.



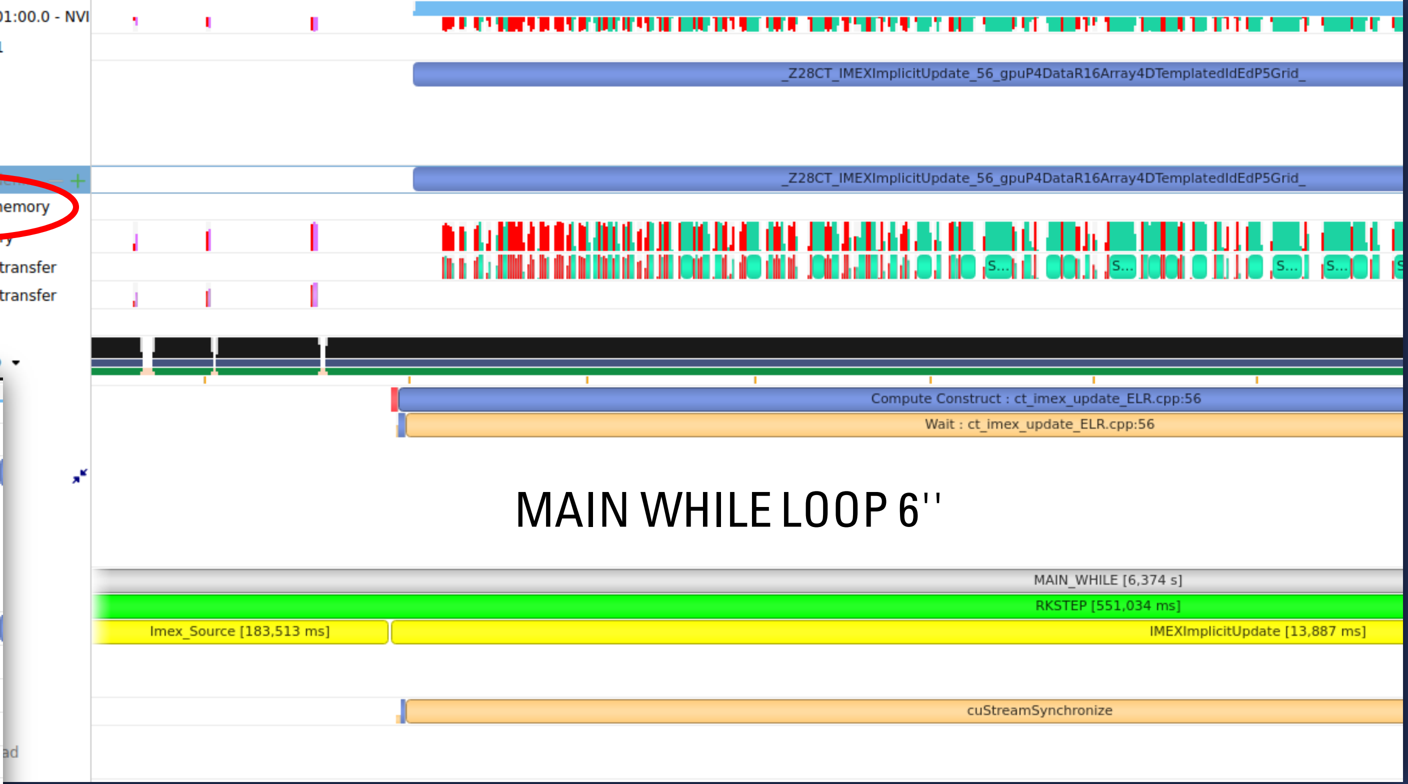
```
#pragma acc parallel loop
for (int i = 0; i < N; i++) {
    // Loop body
    // ...
}
```



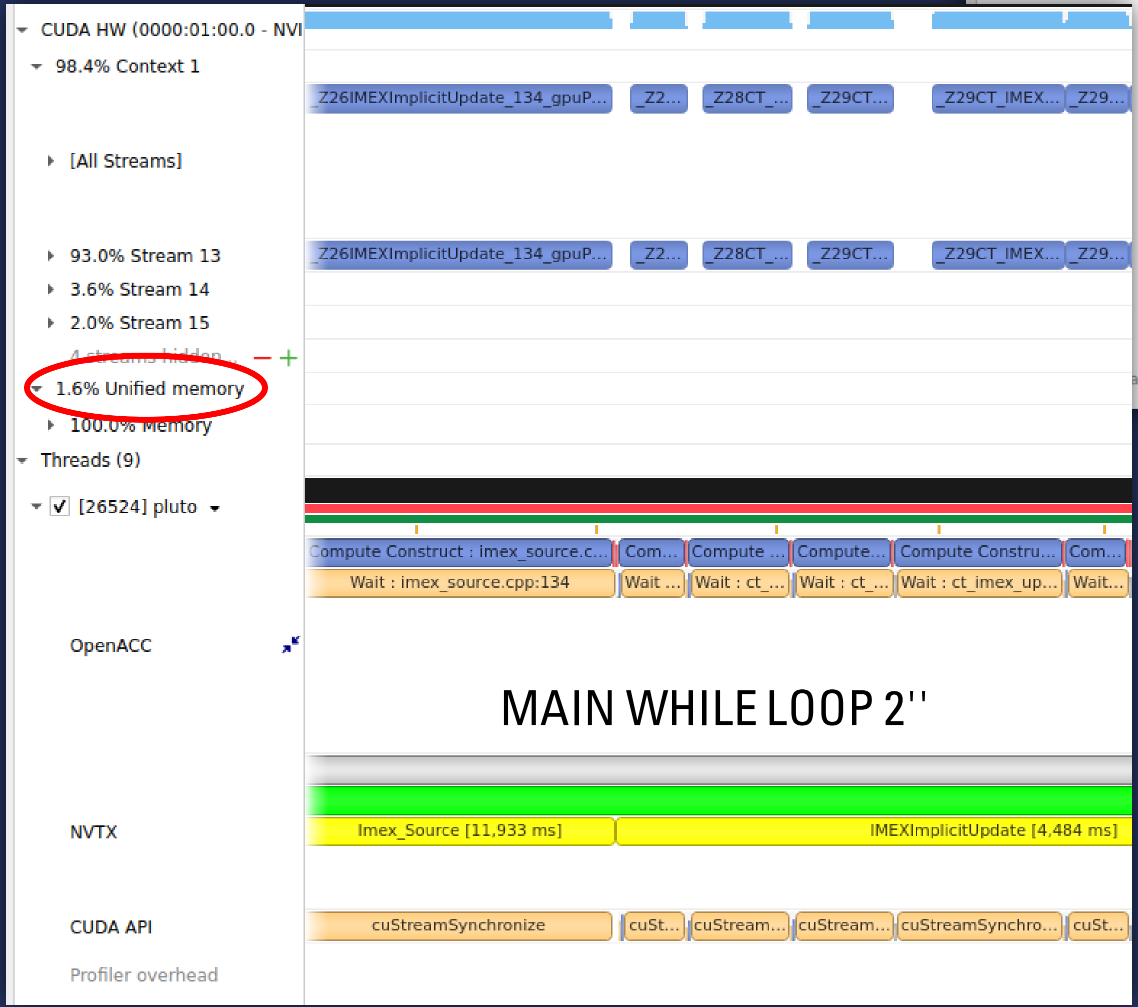
```
#pragma acc enter data copyin(A, B)
// ... Code where A and B are used in GPU computations ...
#pragma acc exit data delete(A, B)
```

The pragma enter data copyin directive is used to explicitly transfer data from the CPU memory to the GPU memory.

DATA-LOCALITY


CUDA HW (0000:01:00.0 - NVI
72.3% Context 1
[All Streams]
27.7% Unified memory
100.0% Memory
56.3% HtoD transfer
43.7% DtoH transfer
Threads (9)
[25318] pluto
_Z28CT_IMEXImplicitUpdate_56_gpuP4DataR16Array4DTemplatedIdEdP5Grid_
_Z28CT_IMEXImplicitUpdate_56_gpuP4DataR16Array4DTemplatedIdEdP5Grid_
Compute Construct : ct_imex_update_ELR.cpp:56
Wait : ct_imex_update_ELR.cpp:56
MAIN_WHILE [6,374 s]
RKSTEP [551,034 ms]
Imex_Source [183,513 ms]
IMEXImplicitUpdate [13,887 ms]
cuStreamSynchronize

MAIN WHILE LOOP 6''



MAIN WHILE LOOP 2''

One point of the computation where quantities are copied back and forth between processors is enough to compromise performace.

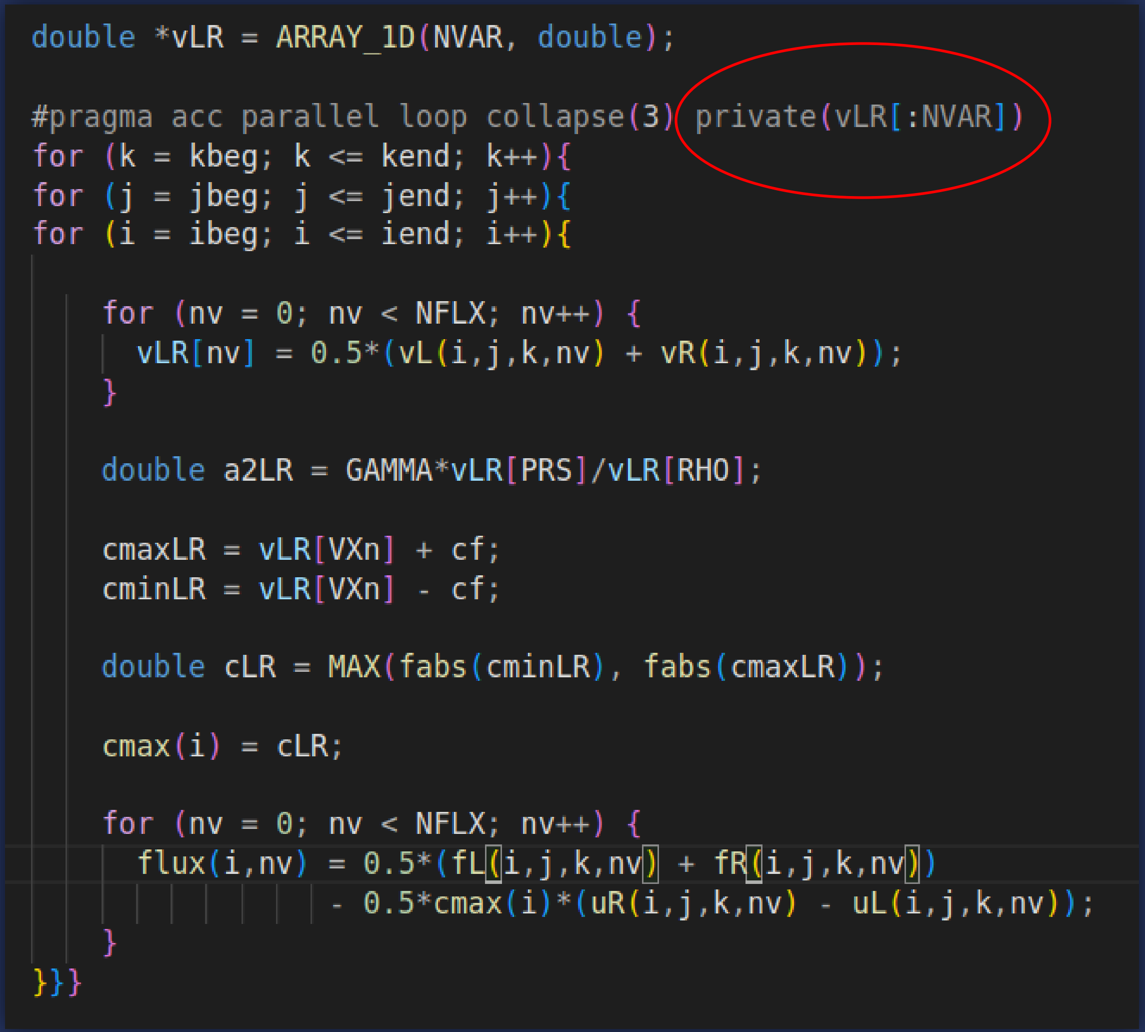
In PLUTO, data are copied on the GPUs memory at the begging of the computation and remains there until the end,

data movement between CPU and GPU is our enemy!

PRIVATE VARIABLES

Private variables are those that have local scope and are allocated individually for each thread or GPU core.

Multiple threads or GPU cores may be executing the same kernel concurrently. If a variable were shared among all threads, concurrent read/write operations could lead to data races and inconsistencies.

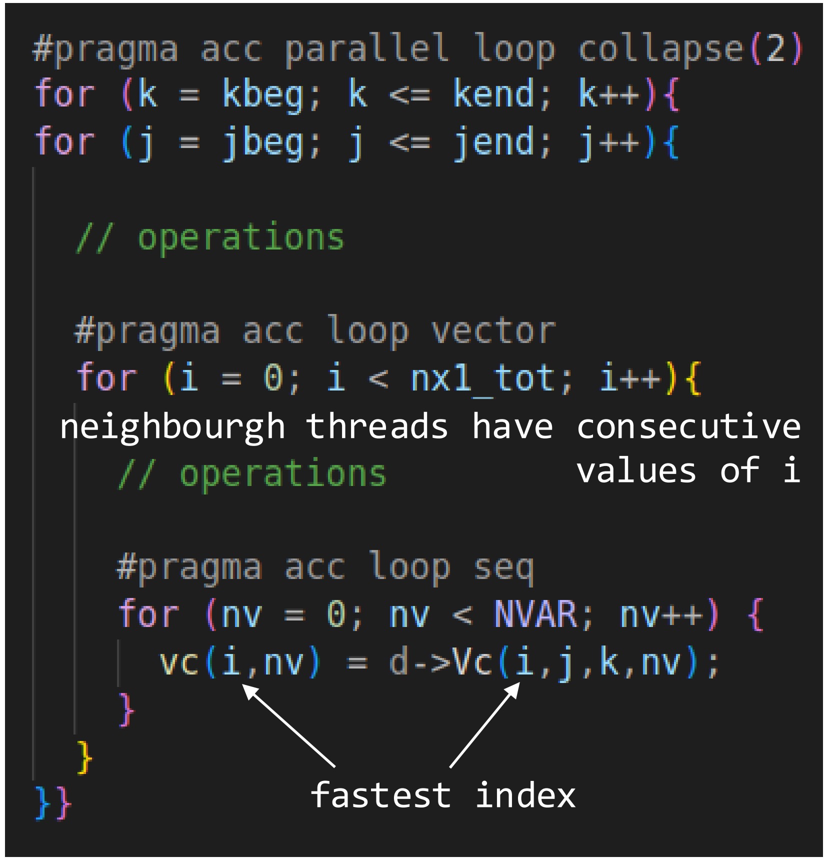


```
double *vLR = ARRAY_1D(NVAR, double);

#pragma acc parallel loop collapse(3) private(vLR[:NVAR])
for (k = kbeg; k <= kend; k++){
for (j = jbeg; j <= jend; j++){
for (i = ibeg; i <= iend; i++){

    for (nv = 0; nv < NFLX; nv++) {
      vLR[nv] = 0.5*(vL(i,j,k,nv) + vR(i,j,k,nv));
    }

    double a2LR = GAMMA*vLR[PRS]/vLR[RHO];

    cmaxLR = vLR[VXn] + cf;
    cminLR = vLR[VXn] - cf;

    double cLR = MAX(fabs(cminLR), fabs(cmaxLR));

    cmax(i) = cLR;

    for (nv = 0; nv < NFLX; nv++) {
      flux(i,nv) = 0.5*(fL(i,j,k,nv) + fR(i,j,k,nv))
                 - 0.5*cmax(i)*(uR(i,j,k,nv) - uL(i,j,k,nv));
    }
}}}
```

COALESCED MEMORY ACCESS

When neighboring threads access neighboring memory locations, it is considered coalesced memory access.

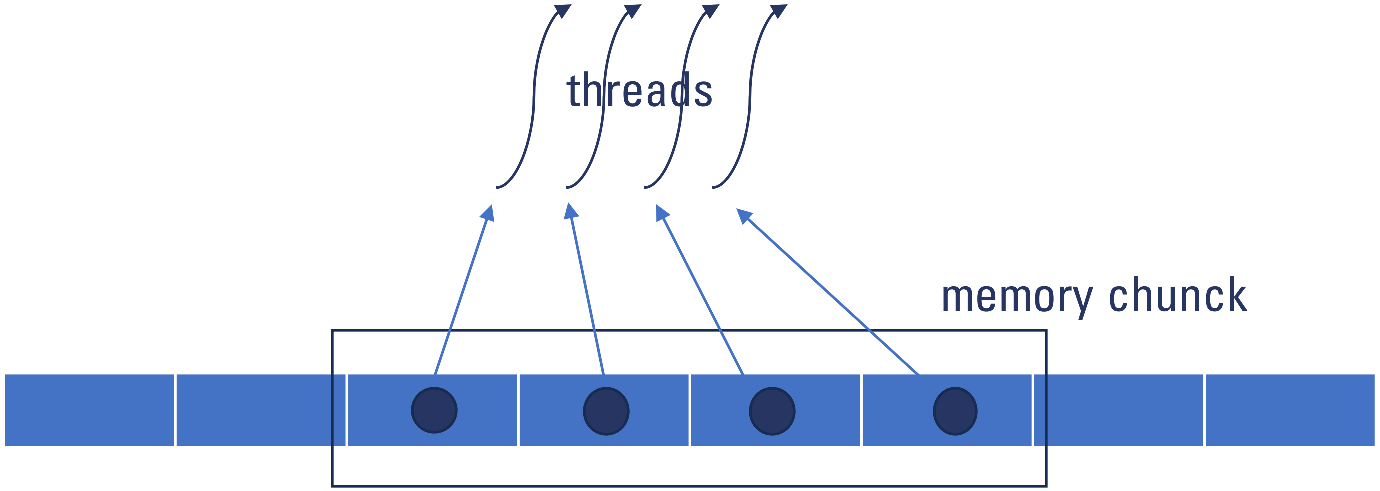
In this case the GPU can perform memory transactions more efficiently, reducing the overall memory access time and improving performance.



```
#pragma acc parallel loop collapse(2)
for (k = kbeg; k <= kend; k++){
for (j = jbeg; j <= jend; j++){

  // operations

  #pragma acc loop vector
  for (i = 0; i < nx1_tot; i++){

    // operations

    #pragma acc loop seq
    for (nv = 0; nv < NVAR; nv++) {
      vc(i,nv) = d->Vc(i,j,k,nv);
    }
  }
}}
```

neighbourgh threads have consecutive values of i

fastest index

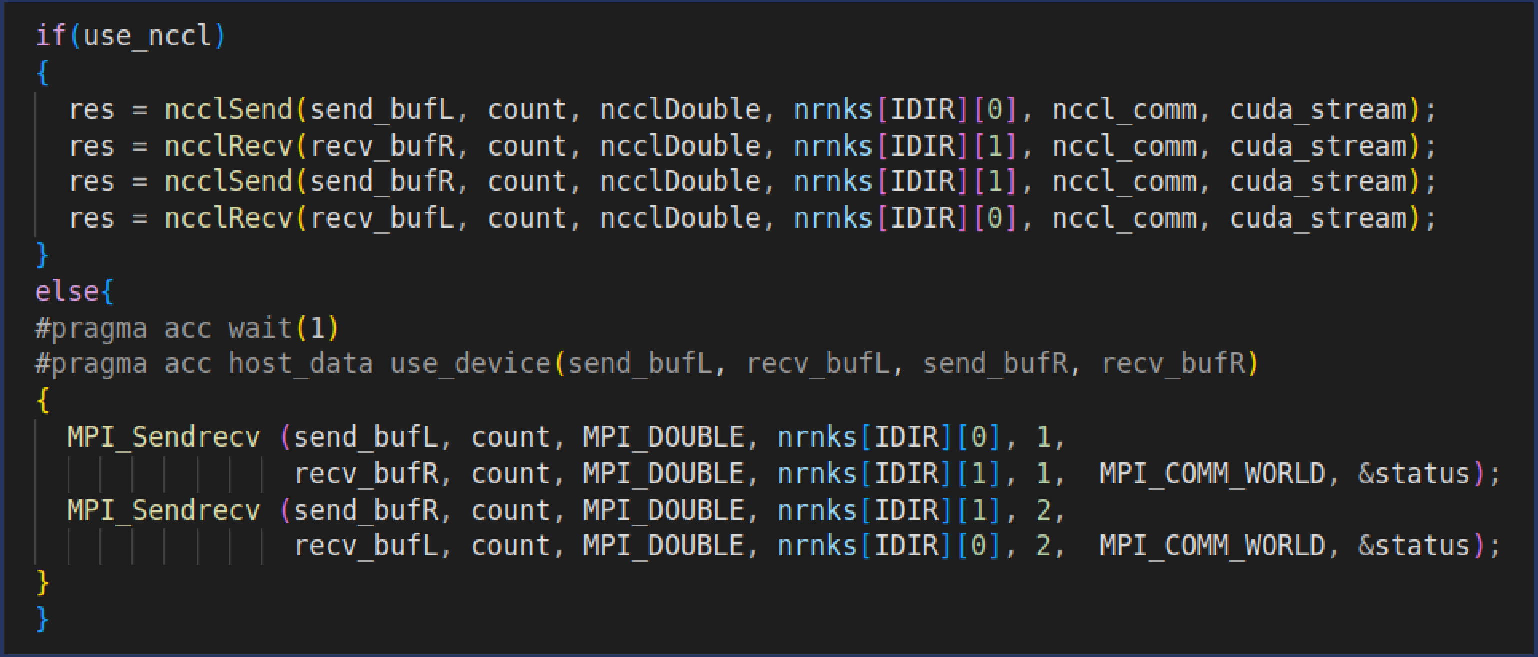


NCCL

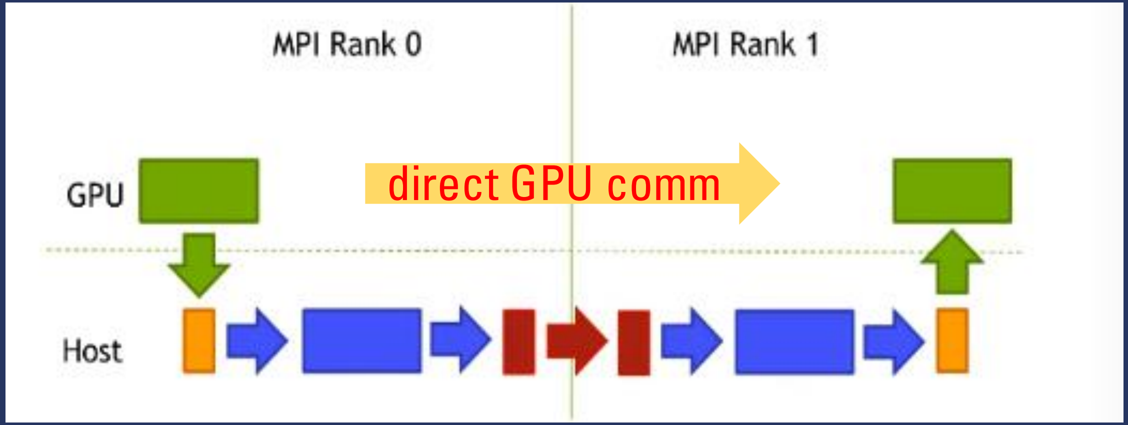
NCCL stands for NVIDIA Collective Communications Library.

It is a software library developed by NVIDIA that provides high-performance inter-GPU communication primitives for parallel computing applications.

In particular NCCL assures direct GPU communication.



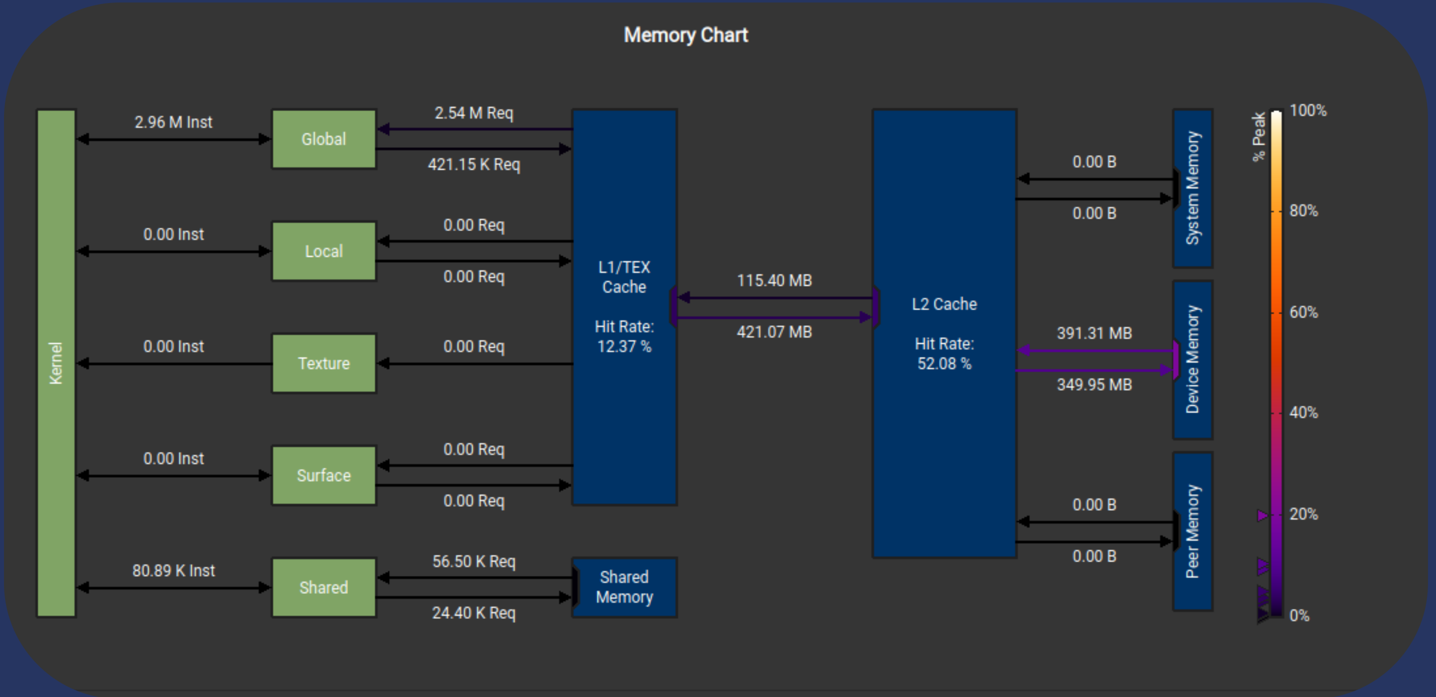
```
if(use_nccl)
{
  res = ncclSend(send_bufL, count, ncclDouble, nrnks[IDIR][0], nccl_comm, cuda_stream);
  res = ncclRecv(recv_bufR, count, ncclDouble, nrnks[IDIR][1], nccl_comm, cuda_stream);
  res = ncclSend(send_bufR, count, ncclDouble, nrnks[IDIR][1], nccl_comm, cuda_stream);
  res = ncclRecv(recv_bufL, count, ncclDouble, nrnks[IDIR][0], nccl_comm, cuda_stream);
}
else{
#pragma acc wait(1)
#pragma acc host_data use_device(send_bufL, recv_bufL, send_bufR, recv_bufR)
{
  MPI_Sendrecv (send_bufL, count, MPI_DOUBLE, nrnks[IDIR][0], 1,
                recv_bufR, count, MPI_DOUBLE, nrnks[IDIR][1], 1,  MPI_COMM_WORLD, &status);
  MPI_Sendrecv (send_bufR, count, MPI_DOUBLE, nrnks[IDIR][1], 2,
                recv_bufL, count, MPI_DOUBLE, nrnks[IDIR][0], 2,  MPI_COMM_WORLD, &status);
}
}
```



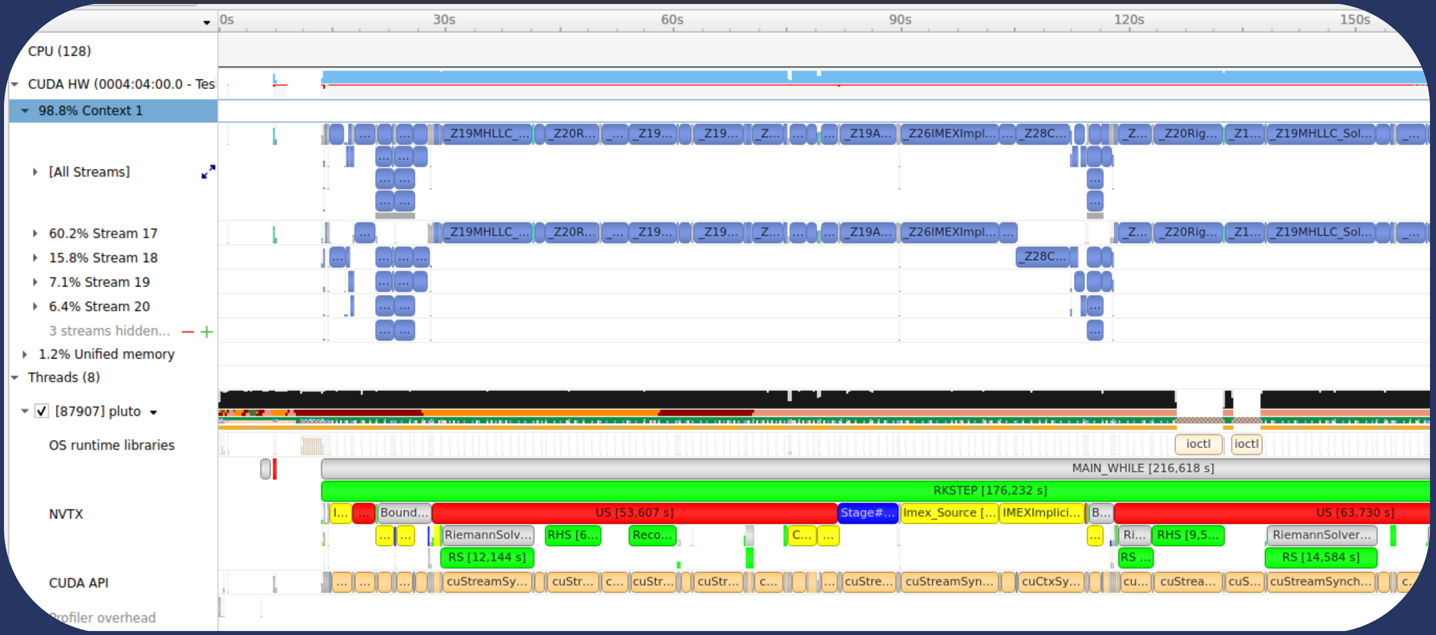
Multi-GPU Programming with CUDA, GPUDirect, NCCL, NVSHMEM, and MPI (CWES1084)

PROFILING TOOLS

Nsight-Compute



Nsight-System combined with NVTX



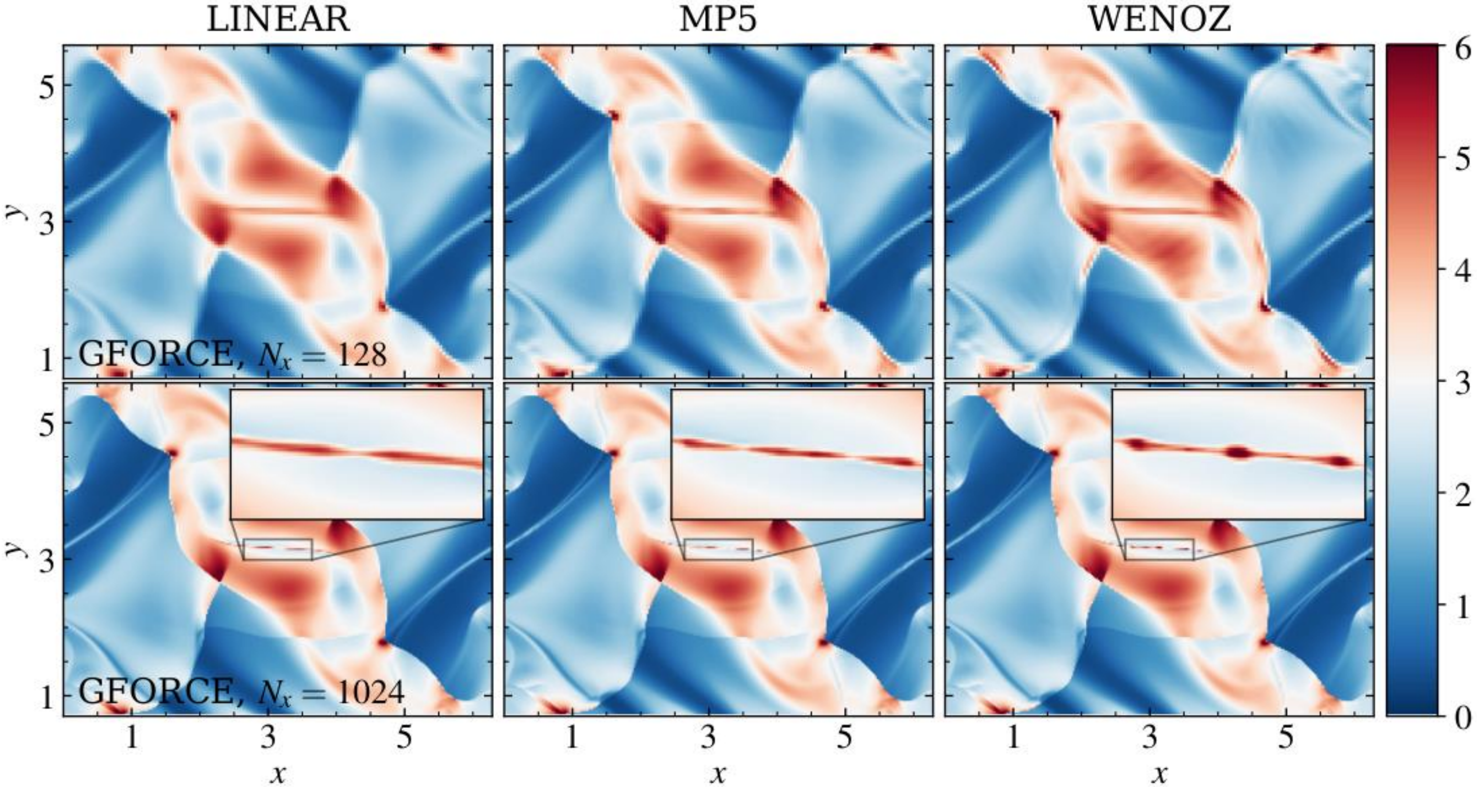
PERFORMANCE

	Marconi100 **ResRMHD Blast** 304^3 points
PLUTO CPU only 32(*4) proc	3106'' /12.2
gPLUTO CPU only 32(*4) proc	3914'' /15.5
gPLUTO GPU 4 proc	253''

	Leonardo **ResRMHD Blast** 512^3 points
gPLUTO CPU only 32 proc	>9h /30
gPLUTO GPU 4 proc	1036'' good scaling /1.8
gPLUTO GPU 8 proc (2 nodes)	572''

NEXT STEPS

- Optimization
- Porting of the Particle Module
- Implementation of the High-Order method for the ResRMHD module



A 4th -order accurate finite volume method for ideal MHD and RMHD based on pointwise reconstructions
V. Berta , A. Mignone , M. Bugli , G. Mattia

SUPPORT

https://www.openacc.org/events

NVIDIA

https://www.cineca.it

CINECA

https://www.space-coe.eu

SPACE

Thank you!

UNIVERSITAS STUDII TAURINENSIS 1404