Tackling exascale systems for astrophysics

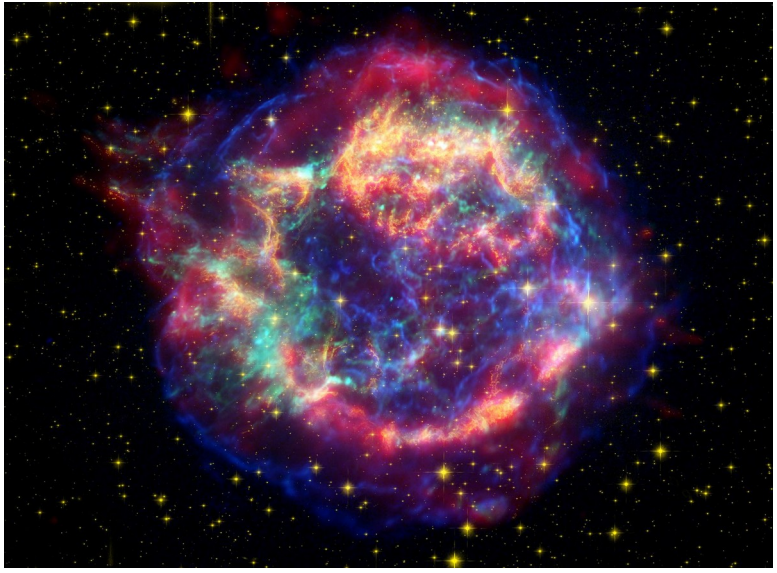
Thomas Padioleau[1], Lou Roussel-Hard[1], Edouard Audit[1], Luc Dessart[2], Yushan Wang[1]

1. Maison de la Simulation, CEA Paris-Saclay

2. Institut d’Astrophysique de Paris

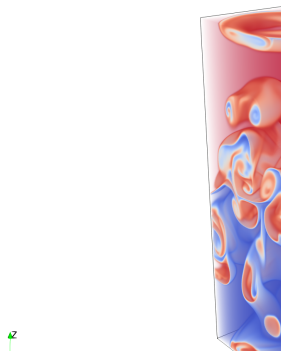
Application context

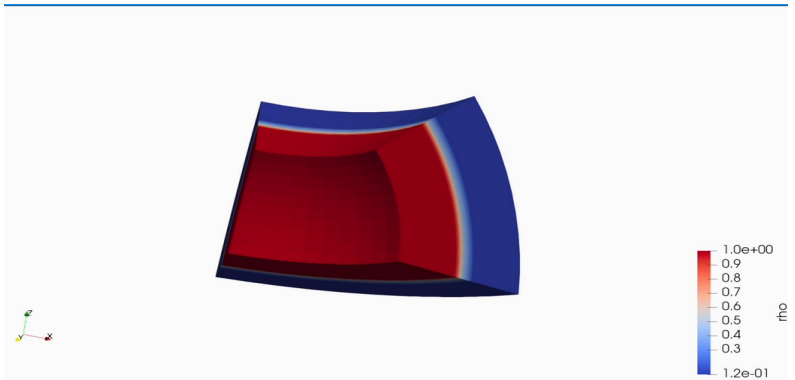
- Phd thesis of L. Roussel-Hard advised by E. Audit and L. Dessart
- Multidimensional radiative hydrodynamics for supernovae
  - Study of local instabilities
  - Evolution of the Nickel 56 bubbles
  - Spectral characteristics of the ejecta
- Existing code Heracles
  - Fortran
  - CPU
  - Makefile



Heracles++ features

- 1D/2D/3D structured mesh
  - Cartesian geometry
  - Spherical geometry
- Hydrodynamics
  - Explicit finite volumes schemes, Godunov-types
  - MUSCL-Hancock second order reconstruction
  - HLL, HLLC, low-Mach approximate Riemann solvers
  - Equations of state
    - Perfect gas
    - Radiative gas
- Radiative transfer [WIP]
  - Hydrodynamics equation of state
  - reduced speed of light
  - implicit M1 model
- Gravity
  - Uniform
  - Point mass
- Arbitrary passive scalars





Technical choices

- Build system
  - CMake
- Language
  - C++17
  - Dynamic and static polymorphism
- Parallelization
  - MPI
    - Domain decomposition
  - Kokkos
    - One Kokkos::View per physical variable
    - parallel_for, parallel_reduce
    - Kokkos::MDRangePolicy
- I/O
  - PDI with HDF5
  - Simple in-situ computation

```
Kokkos::parallel_for("Parallel_for_example",
Kokkos::MDRangePolicy<Kokkos::Rank<3>>({0, 0, 0}, {nx, ny, nz}),
KOKKOS_LAMBDA(int i, int j, int k) {
        auto F_L = compute_FL(i,j,k);
        auto F_R = compute_FR(i,j,k);
        auto dv = compute_dV(i,j,k);
        U_new(i, j, k) = U(i, j, k) + dt/dv * (F_L - F_R) + dt * S(i, j, k);
});
```

Compact code: 7kLOC

User experience

- MDRangePolicy
  - Easy to use
  - Tiling not always needed
  - Useful to get a thread-private allocator for MDRangePolicy
    - Static: experiment on using Kokkos::View<double[ndim][2]>
    - Dynamic: ?
- Difficulties to get vectorization
  - Need aggressive inlining
  - Need loop unrolling due to inner loops over the number of dimensions
- Usage of KOKKOS_CLASS_LAMBDA
  - Some classes are not copyable
  - Warnings about classes not device-compatible
- A Kokkos::View per physical variable
  - API can be verbose → possibility to have incorrect order
  - A lot of metadata is duplicated, can it impact performance ?
- MPI device-aware
  - Difficulty to know if the MPI implementation supports it

Thank you for your attention!