Enabling Efficient Runtime Data Analysis to a Crystal Deformation Simulation

PhD Student: Arthur Jaquard,[1] **Advisors:** Gabriel Antoniu,[1] Laurent Colombet,[2] Silvina Caino-Lores,[1] Julien Bigot[3]
Contact: arthur.jaquard@inria.fr
1. University of Rennes, Inria, CNRS, IRISA, Rennes, France
2. CEA, DAM, DIF, Bruyères-le-Châtel, France
3. Université Paris-Saclay, UVSQ, CNRS, CEA, Maison de la Simulation, Gif-sur-Yvette, France

PROGRAMME DE RECHERCHE — NUMÉRIQUE POUR L'EXASCALE — Exa-DoST

Context and Objectives

Coddex

Code for Discontinuous Deformation Evolution in Xstals

- **MPI-based simulation (+ OpenMPI)**
- **Hundreds of variables per data point**
- **Multiple data-analysis pipelines**

Coddex : a simulation code that solves the equations of continuum mechanics in dynamic hyperelasticity (shocks, and rapid loading).

Element-free Galerkin solver, that works at the smallest scale at which differential equations can be used to model matter.

Chosen as a demonstrator to be used on the upcoming Alice Recoque supercomputer from the Jules Vernes Consortium.

- Direct output of some variables for later analysis (e.g. Paraview)
- Projection output of a given variable as Postscript image
- Compute and render X-ray diffraction within the crystal
- …

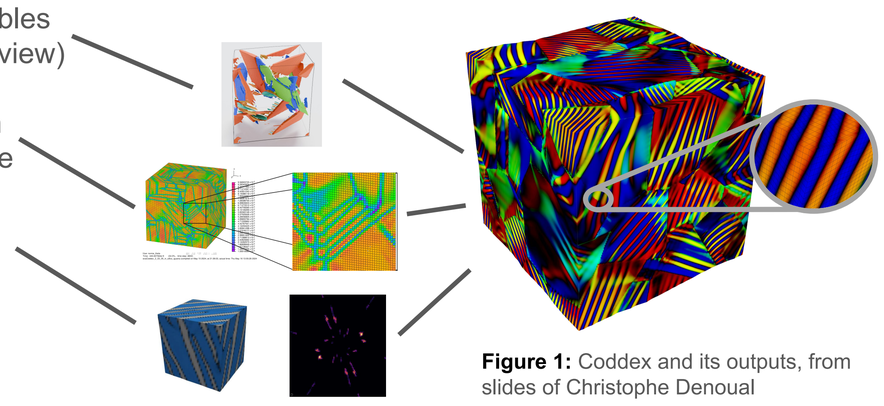

Figure 1: Coddex and its outputs, from slides of Christophe Denoual

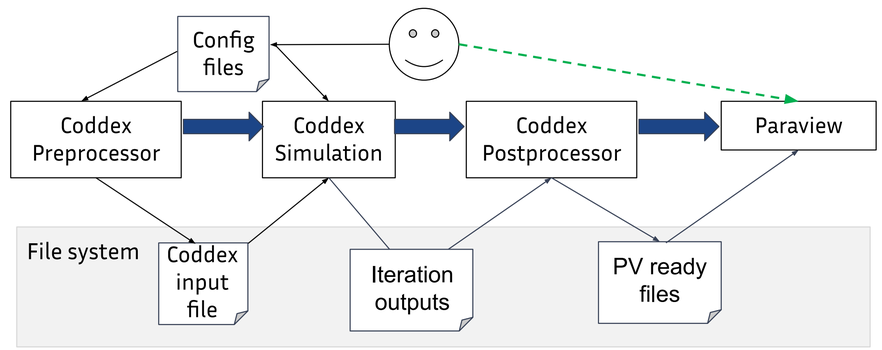

Figure 2: Existing Coddex workflow for post-hoc viz / analysis with Paraview

Limitations of the current workflow in operation

- All simulation data is written to disk and post-processed before analysis and visualization in ParaView (PV)
  - Impossible to conduct these analyses at runtime
  - This introduces limitations to the temporal resolution and number of variables that can be efficiently written at runtime in Coddex

Goals and challenges

- Enable runtime analysis while minimizing overhead
- Increase frequency of analysis

In situ processing

- Perform data analysis at simulation time
- Exchange data through memory and interconnect
- Within a single HPC system

In situ methods appear fitting for Coddex

Damaris

Middleware for in situ processing in large scale, MPI-based HPC application **[1]**

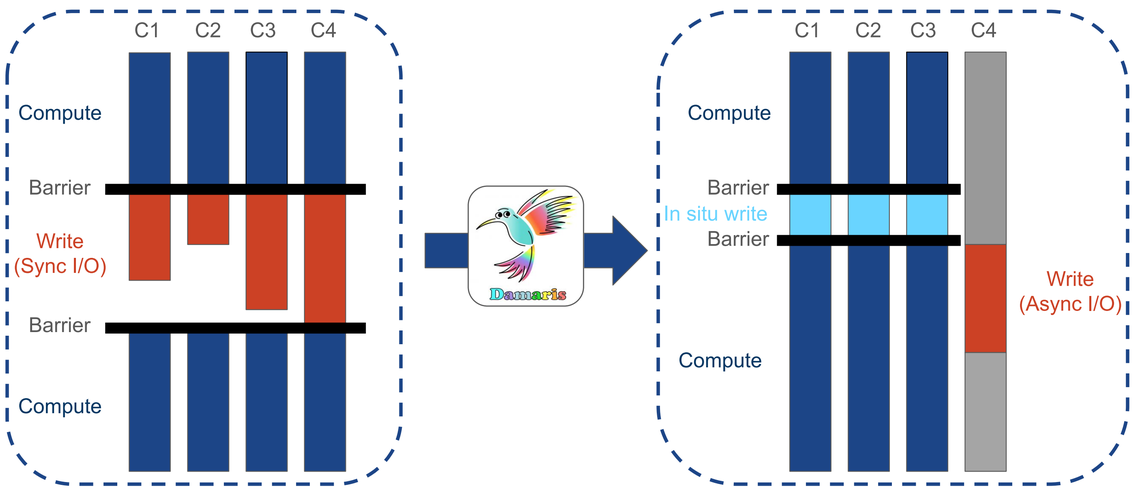

Figure 3: Damaris on dedicated cores for an I/O task. One or many core(s) are dedicated to in situ processing

Why use the Damaris framework with Coddex

- Demonstrated scalability with MPI-based simulations on top supercomputers
- Non-intrusive support of Paraview - Catalyst for in-situ data analysis (+ other plugins)

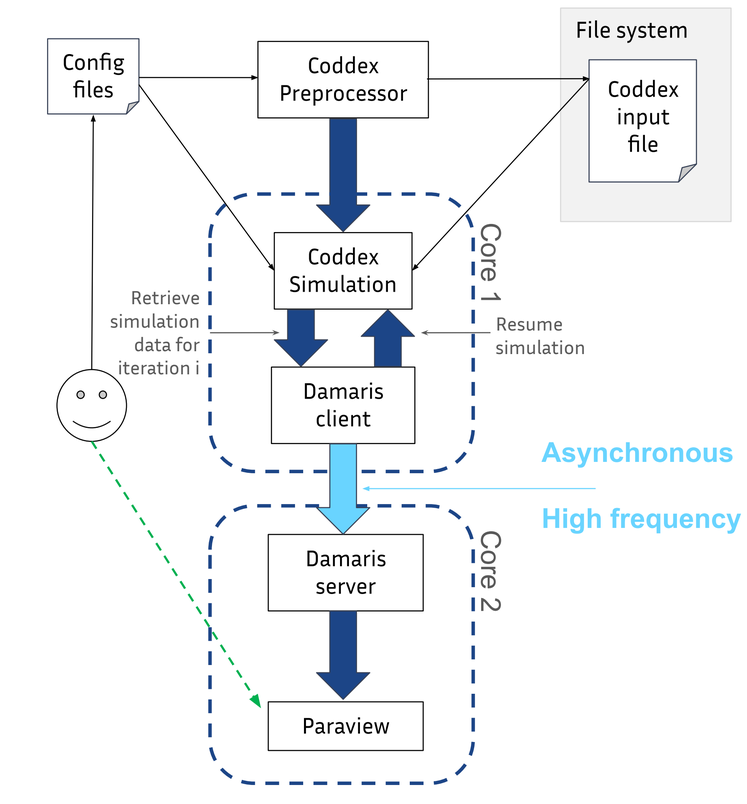

Figure 4: Coddex workflow integrated with Damaris for Paraview in situ visualization / analysis

Our approach

Evaluation and Takeaways

Experimental setting

Testbed INTI (at CEA)

- CPU: AMD Milan (64 cores)
- Nodes: 2 CPU (128 cores)
- Interconnect: Atos BXIv2 100GB/s
- File system: 300GB/s Lustre

Configuration

- Tin hysteresis simulation (of size $n^3 = \{1, 8, 27\}$ for $n = \{1, 2, 3\}$)
- Output data every 10 iterations
  - **I/O scenario:** write raw variable values to disk
  - **In situ scenario:** create PV visualization image in situ, then write image to disk
- #output variables = {5, 50}

Deployment settings

I/O scenario

Simulation of size n^3

- n^3 MPI processes for Coddex
- 64 cores per Coddex process (for 64 OpenMPI threads)
- CPU -> Coddex*64

In situ scenario

Simulation of size n^3

- n^3 MPI processes for Coddex
- 63 cores per Coddex process (for 63 OpenMPI threads)
- CPU -> Coddex*63 + Damaris*1

Execution time per iteration

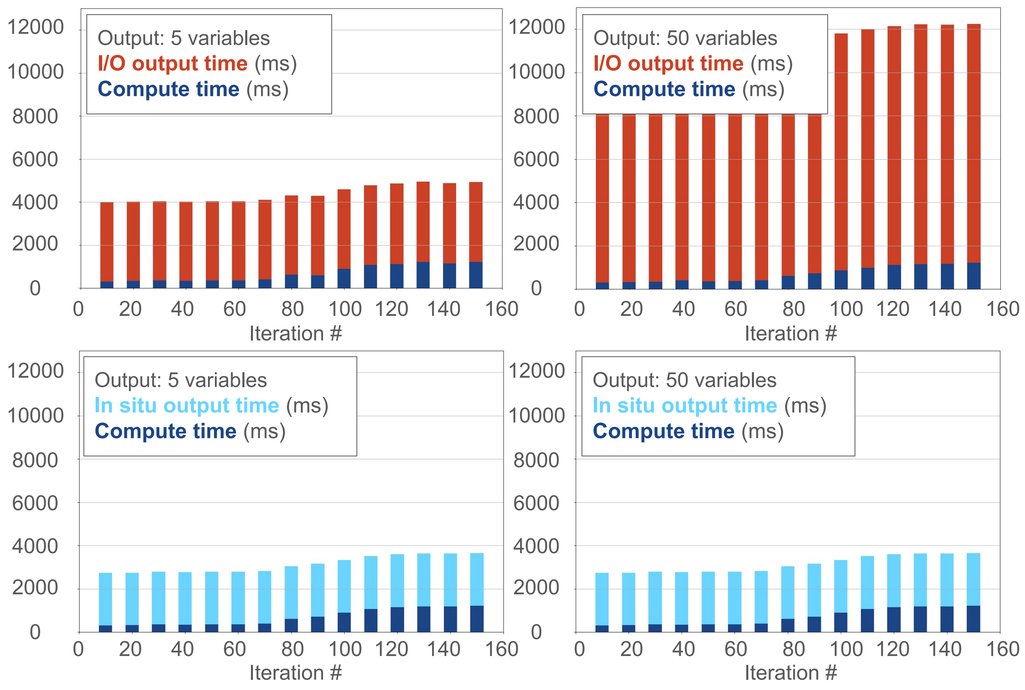

Figure 5: Execution time per iteration (in ms) for the **I/O scenario** (top) and **in situ scenario** (bottom), for respectively 5 output variables (left) and 50 output variables (right). 14 nodes ~ 1728 cores ($n^3 = 27$)

Average iteration output time

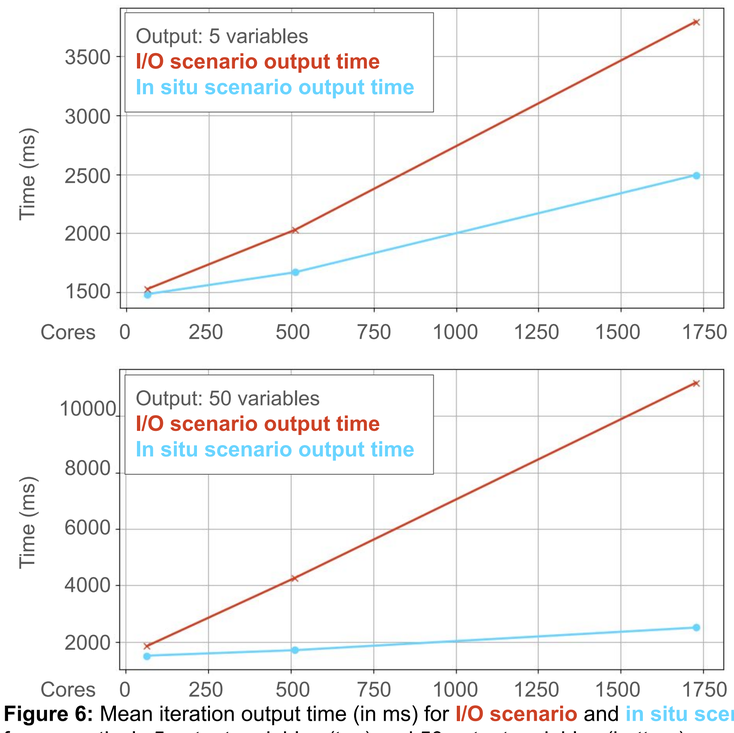

Figure 6: Mean iteration output time (in ms) for **I/O scenario** and **in situ scenario**, for respectively 5 output variables (top) and 50 output variables (bottom).

Takeaways

Using Damaris for in situ processing

- Does not introduce overhead in the measured stages (compared to I/O scenario)
- Reduces execution time at each iteration (by reducing I/O time)
- Becomes more beneficial as the number of variables increases (in situ output time up to 5 times shorter than I/O output time)

Future work

- Investigate the effect of different deployment modes on the scalability and performance of the in situ approach, such as dedicating a node to the Damaris server
- Investigate the effect of introducing Damaris on overall workflow performance

References

[1] Bowden, J. C., Tessier, F., Deltel, C., Bnà, S., Antoniu, G. (2021). *In-situ visualization using Damaris: the Code Saturne use case*. PRACE white paper.

Acknowledgements: Christophe Denoual, developer of Coddex.

RÉPUBLIQUE FRANÇAISE — Liberté Égalité Fraternité

PROGRAMME DE RECHERCHE — NUMÉRIQUE POUR L'EXASCALE