Enabling Efficient Runtime Data Analysis for a Crystal Deformation Simulation

Arthur Jaquard
arthur.jaquard@inria.fr
University of Rennes, Inria, CNRS, IRISA
Rennes, France

Silvina Caino-Lores (advisor)
silvina.caino-lores@inria.fr
University of Rennes, Inria, CNRS, IRISA
Rennes, France

Gabriel Antoniu (advisor)
gabriel.antoniu@inria.fr
University of Rennes, Inria, CNRS, IRISA
Rennes, France

Laurent Colombet (advisor)
laurent.colombet@cea.fr
CEA, DAM, DIF
Bruyères-le-Châtel, France

Julien Bigot (advisor)
julien.bigot@cea.fr
Université Paris-Saclay, UVSQ, CNRS, CEA, Maison de la Simulation
Gif-sur-Yvette, France

Abstract

Exascale simulations generate massive data volumes that strain I/O and post-hoc analysis. We integrate the Damaris in situ middleware into Coddex, a crystal deformation code, to offload data movement and analysis to dedicated processes, enabling runtime extraction of key diagnostics without writing intermediate files. We evaluate tin hysteresis cases on CEA's INTI cluster (with 14 nodes, 1,728 cores) and compare against a ParaView-based post-hoc pipeline. in situ analysis eliminates per-iteration I/O stalls and reduces output time by up to 5× while preserving overall iteration time, with benefits increasing with the number of tracked variables. We describe the integration design, process pinning, and data exchange, and outline forthcoming support for additional analyses. This work is conducted within the Exa-DoST project of the PEPR NumPEx program, which aims to build the software infrastructure for the first Exascale machine expected to be set up in France (Alice Recoque, Jules Verne project).

ACM Reference Format:
Arthur Jaquard, Silvina Caino-Lores (advisor), Gabriel Antoniu (advisor), Laurent Colombet (advisor), and Julien Bigot (advisor). 2025. Enabling Efficient Runtime Data Analysis for a Crystal Deformation Simulation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Supercomputing '25).* ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Coddex (Code for Discontinuous Deformation Evolution in Xstals) is an Element-Free Galerkin solver, developed at CEA, for the equations of dynamic hyperelasticity; it targets crystal plasticity and phase transitions. It uses MPI for inter-node communication and OpenMP for thread-level parallelism. In Coddex, hundreds of variables are tracked at each data point, and several analysis pipelines are available: (1) direct output for post-hoc analysis (e.g., ParaView [1]); (2) 2D projections exported as PostScript; (3) X-ray diffraction rendering; among others. For this work we concentrate on the ParaView pipeline (Figure 1)), where the simulation writes fields to the parallel file system for later processing. However, in the current Coddex workflow, analysis cannot be performed at runtime, which constrains temporal resolution and the number of variables that can be inspected efficiently due to I/O bottlenecks.

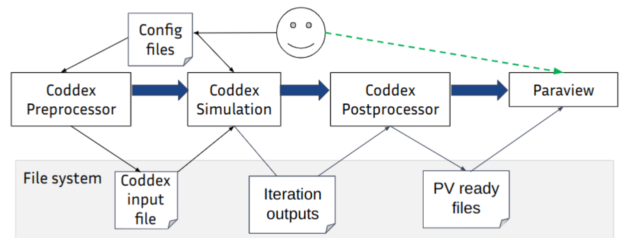

Figure 1: Existing Coddex workflow for post-hoc visualization or analysis with ParaView.

A widely adopted strategy to enable runtime analytics is in situ processing: executing analysis concurrently with the simulation on the same HPC system, exchanging data in memory or via the interconnect rather than files. Multiple frameworks exist with varying integration effort and performance trade-offs [3]. In this work we rely on Damaris, a middleware that enables in situ processing for large-scale MPI applications by dedicating cores and processes to data movement and analysis [2, 4, 5]. Using dedicated cores for in situ processing, it offers non-intrusive support for in situ data analysis with ParaView, making it suitable to use with Coddex (Figure 2).

Context. This research is conducted in Exa-DoST (Data-oriented Software and Tools for the Exascale), a project of the French NumPEx program ("Digital for Exascale") [6]. Exa-DoST develops operational, co-designed solutions to exascale data challenges—scalable storage and I/O, in situ processing, and smart analytics, aiming to contribute to the HPC software stack for future Exascale machine in France and Europe.

Supercomputing '25, St. Louis, MO, USA

ACM ISBN 978-X-XXXX-XXXX-X/YYYY/MM
https://doi.org/XXXXXXX.XXXXXXX

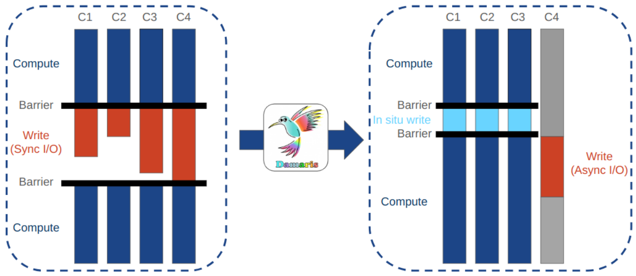

Figure 2: Damaris deployment with dedicated cores: simulation publishes data to Damaris servers, which marshal them to the analysis back-end (e.g., ParaView)

Goal. Our objective is to integrate Damaris into Coddex to enable runtime extraction of key diagnostics without intermediate files, and to quantitatively compare this in situ pipeline against the existing ParaView-based post-hoc one. We evaluate scalability and overhead on a modern HPC cluster at CEA, and report the conditions under which in situ analysis improves execution time for crystal deformation workloads.

Approach. To fulfill our objective to support efficient runtime analysis pipelines in Coddex, we integrate Damaris into the existing Coddex workflow. The integration is done by modifying the Coddex code to use the Damaris client-side API, which allows Coddex to make its variable fields available to the Damaris client running on dedicated cores. This way, we can extract data from Coddex at runtime without writing it to disk, and pass it directly to the analysis backend –in this case, ParaView– through the Damaris server. The Damaris server is also run on dedicated resources, which allows it to handle the data extraction and analysis without interfering with the Coddex simulation.

2 Evaluation

We perform our experiments on the INTI experimental computing center, hosted at CEA. Each node has two 64-core AMD Milan CPUs, and the file system is a 300 GB/s Lustre system. The nodes are connected by an Atos BXIv2 100GB/s interconnect.

We run multiple tin hysteresis iterative simulations with variable problem domain size (i.e., $\propto n^3 \in \{1, 8, 27\}$, for $n \in \{1, 2, 3\}$). Coddex is configured to output $v \in \{5, 50\}$ variables every 10 iterations. To evaluate the impact of our in situ approach, we run two different scenarios:

(1) **Baseline I/O scenario.** The output is directly written to disk at every 10th iteration. We use n^3 MPI processes for Coddex, 64 cores per Coddex process (for 64 OpenMPI threads), and each Coddex process is pinned to exactly one CPU.

(2) **Enhanced in situ scenario.** The output is extracted from Coddex into the Damaris in situ framework at every 10th iteration. We use n^3 MPI processes for Coddex, 63 cores per Coddex process (for 63 OpenMPI threads), and each Coddex process is pinned to exactly one CPU leaving one dedicated core for the Damaris server process.

For both scenarios we measure the execution time at every 10th iteration. This execution time is divided into time spent running the

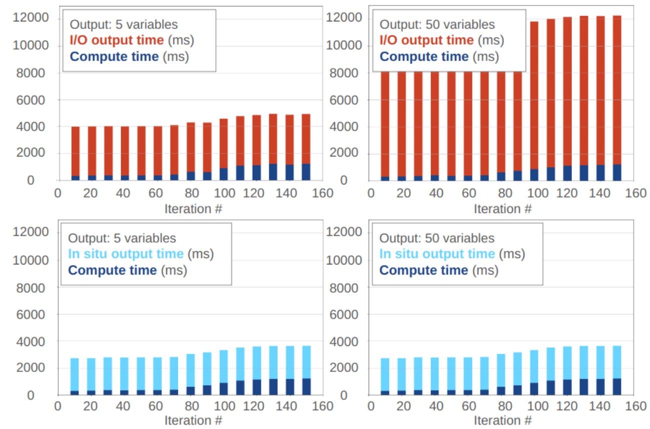

Figure 3: Execution time per iteration (in ms) for the I/O scenario (top) and in situ scenario (bottom), for respectively $v = 5$ (left) and $v = 50$ (right). Experiment run on 14 nodes, 1,728 cores ($n^3 = 27$).

Coddex simulation at a given iteration, and the time spent writing the output to disk or extracting it into Damaris. Figure 3 shows the execution time per iteration for both scenarios running on the largest experimental setup (i.e., mesh size of $n^3 = 27$, corresponding to 1,728 cores in 14 nodes). The top row corresponds to the baseline I/O scenario, while the bottom row shows the enhanced in situ scenario. For each scenario, results are presented for $v = 5$ (left) and $v = 50$ (right) output variables. We observe that not only the in situ approach does not introduce overhead, but it consistently reduces the output time at every iteration, especially as the number of variables increases. Figure 4 shows the average output time across iterations for both scenarios as the number of cores and variables increases. We observe a similar trend as in Fig. 3, with the in situ approach consistently reducing the output time compared to the baseline I/O scenario, and becoming more efficient as the number of variables increases.

3 Conclusions and future work

By dedicating cores and using asynchrony, Damaris enables runtime analysis in Coddex without increasing iteration time, while reducing output time—by up to 5× faster (≈ 80% less time) when many variables are tracked. These results align with prior Damaris integrations and indicate a straightforward path to scaling Coddex workflows that require runtime insight. Beyond per-iteration timing, Damaris's asynchronous output management suggests that end-to-end workflow time can drop further; we will quantify this in future experiments. We will also study alternative deployment modes (e.g., dedicating full nodes to Damaris servers), and extend the in situ path with additional analyses to broaden Coddex's capabilities.

Acknowledgments

Christophe Denoual, developer of Coddex.

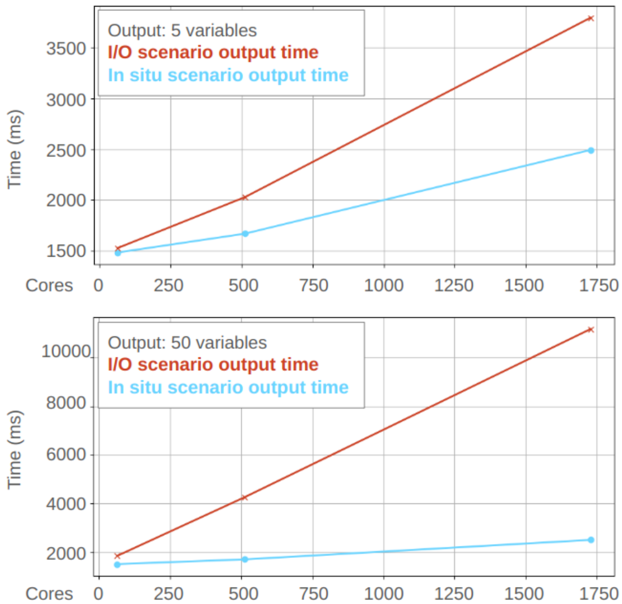

Figure 4: Average iteration output time (in ms) for I/O scenario and in situ scenario, for respectively $v = 5$ (top) and $v = 50$ (bottom).

References

[1] Utkarsh Ayachit, Andrew Bauer, Berk Geveci, Patrick O'Leary, Kenneth Moreland, Nathan Fabian, and Jeffrey Mauldin. 2015. Paraview catalyst: Enabling in situ data analysis and visualization. In *Proceedings of the first workshop on in situ infrastructures for enabling extreme-scale analysis and visualization*. 25–29.

[2] Joshua C Bowden, François Tessier, Charles Deltel, Simone Bnà, and Gabriel Antoniu. 2021. In-situ visualization using Damaris: the Code Saturne use case. https://inria.hal.science/hal-03354035 PRACE White Paper.

[3] Hank Childs, Sean D Ahern, James Ahrens, Andrew C Bauer, Janine Bennett, E Wes Bethel, Peer-Timo Bremer, Eric Brugger, Joseph Cottam, Matthieu Dorier, et al. 2020. A terminology for in situ visualization and analysis systems. *The International Journal of High Performance Computing Applications* 34, 6 (2020), 676–691.

[4] Damaris 2025. In Situ Data Analysis and Visualization in Support of Large-Scale HPC Simulations. https://project.inria.fr/damaris/

[5] Matthieu Dorier, Gabriel Antoniu, Franck Cappello, Marc Snir, Robert Sisneros, Orcun Yildiz, Shadi Ibrahim, Tom Peterka, and Leigh Orf. 2016. Damaris: Addressing Performance Variability in Data Management for Post-Petascale Simulations. *ACM Transactions on Parallel Computing* 3, 3 (2016), 15. doi:10.1145/2987371

[6] Exa-DoST. 2025. Data-oriented Software and Tools for the Exascale. https://numpex.org/exadost-data-oriented-software-and-tools-for-the-exascale/