Proceedings of the 9th bwHPC Symposium 2023

October 23, 2023
University of Mannheim

Mannheim Conference Series, 2025
https://majournals.bib.uni-mannheim.de/maconf/issue/view/80

Contents

Continuous Benchmarking of Numerical Algorithms Implemented in M++ via Gitlab CI/CD and Google Benchmark*

Niklas Baumgarten[1] and Daniele Corallo[2]

[1]Department of Mathematics, Heidelberg University
[2]Department of Mathematics, Karlsruhe Institute of Technology

Abstract

We present an automated framework for benchmarking numerical algorithms that solve partial differential equations under consistent and reproducible conditions using the parallel finite element software M++. This framework integrates GitLab CI/CD, Google Benchmark, and the HoreKa supercomputing system to enable continuous integration and benchmarking. By incorporating ongoing software development, the framework supports improving performance and reliability, which are vital for various scientific computing applications, including wave propagation, cardiovascular simulations, dislocation dynamics, and uncertainty quantification. These applications motivate the two benchmarking examples presented in this text. We further outline the benchmarking workflow as well as the use of a research database storing comprehensive performance data, facilitating reproducibility for future studies.

1 Introduction

Advancements in computational science and engineering nowadays often depend on the sustainable and robust development of open-source software. It has become a major carrier of knowledge and scientific methodologies, playing a key role in understanding physical, biological, technical, or economical phenomena. As complexity increases in the modeling of such phenomena, the supporting software systems must also grow in complexity, creating a multi-target optimization problem where usability, flexibility, performance, reliability, and maintainability must be balanced. Consequently, careful software engineering, supported by performance data collection and automated testing, has become a crucial part of the development process.

Here, we present our approach to this engineering task: an automated benchmarking system within the release cycle for M++, a high-performance computing (HPC) finite element method (FEM) software [8], developed over two decades by numerous researchers based on the initial works [26, 27]. Today, M++ is applied across a wide range of scientific fields, such as geoscience, where it is used for full waveform inversion in seismic imaging [10, 15, 22] and for simulating gas dynamics in carbon capturing [21]. It is also employed in material science to model nonlinear solid mechanics [9] and dislocation dynamics [23, 24, 25], as well as in life science to simulate cardiovascular processes [17, 18, 19].

M++ implements a wide range of numerical methods, e.g. standard Lagrange FEM, discontinuous Galerkin (DG) and Raviart-Thomas discretizations as well as a Message Passing Interface (MPI) based parallel linear algebra providing several preconditioners and Krylov subspace solvers [8]. Further, more recent features include implicit space-time discretizations for wave [11, 13, 14] and transport [28, 12] equations, interval arithmetic methods for computer-assisted existence proofs [31], and efficient time integration schemes combined with DG methods [20] for simulating time-dependent wave problems. The software also provides methodologies for large-scale uncertainty quantification (UQ), such as multilevel Monte Carlo, stochastic collocation and stochastic gradient descent methods [4, 5, 6, 7], to cope with the inherent uncertainties in the physical models and to impose an optimal control to the aforementioned applications.

*Proceedings of the 9th bwHPC-Symposium 2023. Mannheim Conference Series (MaConf). 2025.

Supporting such a wide range of applications and algorithms results in different requirements on the software architecture and the development process. It raises the questions on, when, how and where performance improvements should be considered and which new features might be pursued to further expand the application and decrease the runtime of the software. The discussion in [1] and the development process of *Ginkgo* [2] and *OpenCarp* [3] serve as inspiration to address these questions by including benchmarking and automated performance evaluation to the continuous integration/continuous deployment (CI/CD) pipeline of M++. As a result, we collect comprehensive data to make informed decisions on how to proceed in the development process, while ensuring that already achieved results are maintained or even improved.

In this text, we describe the details of this benchmarking approach, its integration within the CI/CD pipeline, and the HoreKa supercomputing system to enable fair comparisons using Google Benchmark as the evaluation tool. We present two benchmarking examples, each solving a PDE to outline the simplicity of the new set-up for the software.

2 Methodology

Researchers at different universities develop M++, e.g. at the Karlsruhe Institute of Technology (KIT) or Heidelberg University, all with different research projects and different requirements to the software. To streamline the development and to be able to always move forward towards a common, new and improved version of the software, we work within release cycles as described in [16]. We refer to Figure 1 at hand of which we explain the procedure and the involved technologies used to create a new version of the software.

The starting point for a new software release (left side of Figure 1) typically involves programming a new algorithm, designing a computational model, or enhancing existing code. Researchers with access to the Git repository[1] can make changes, which are automatically built by the Gitlab CI pipeline with various configurations within Docker containers. These containers run on a Kubernetes cluster at the Department of Mathematics at KIT, executing a rigorous and large set of tests to verify the correctness of the code (cf. top left of Figure 1). Typical tests check for consistency of the implementation based on mathematical theory, execute convergence experiments with respect to a certain discretization parameter, but also verify the integration of the core M++ library within projects depending upon it.

Release candidates of M++ which have passed the build and verification stage can be deployed in a next step on the HoreKa supercomputing system (cf. top right of Figure 1). Here, the pipeline executes HPC-experiments and benchmarks parallelized with Open MPI, typically in the range of 64 to 16384 processing units, to create the simulation data of interest in the form of HDF5[2] or VTU files, but also to collect performance and configuration records in JSON files. All benchmark data is collected within the Google benchmark framework and archived.

Subsequently, the data collected within the benchmarking stage is analyzed and post-processed via Python tools (cf. bottom right of Figure 1). This involves an automated comparison with previous release data, and preparation for a final publication with a persistent identifier via the RADAR[3] service. (cf. bottom left of Figure 1). With this final publication, new benchmarking results are available, serving as the reference for future developments and a new release cycle.

At the heart of this cycle is Gitlab CI/CD as the automation tool and the driver of this procedure. It takes over the configuration and execution of the aforementioned workflow by defining the utilized infrastructure and stages in YML files. As a result, all components are easy to maintain, replace or extend simply by changing the Gitlab CI/CD configuration. Under this development workflow, we currently create between three and five new releases every year depending on the implemented features.

With over 30 releases ([30] as the latest), each developed with increasing levels of scrutiny, this workflow has driven major improvements in usability, reliability and ensuring reproducibility of results, which is crucial for the scientific method. To highlight the continuous advancements in quality control, we present two benchmarking examples in the following section.

[1] `https://gitlab.kit.edu/kit/mpp/mpp/`
[2] availbale in upcoming release
[3] `https://radar.products.fiz-karlsruhe.de/en`

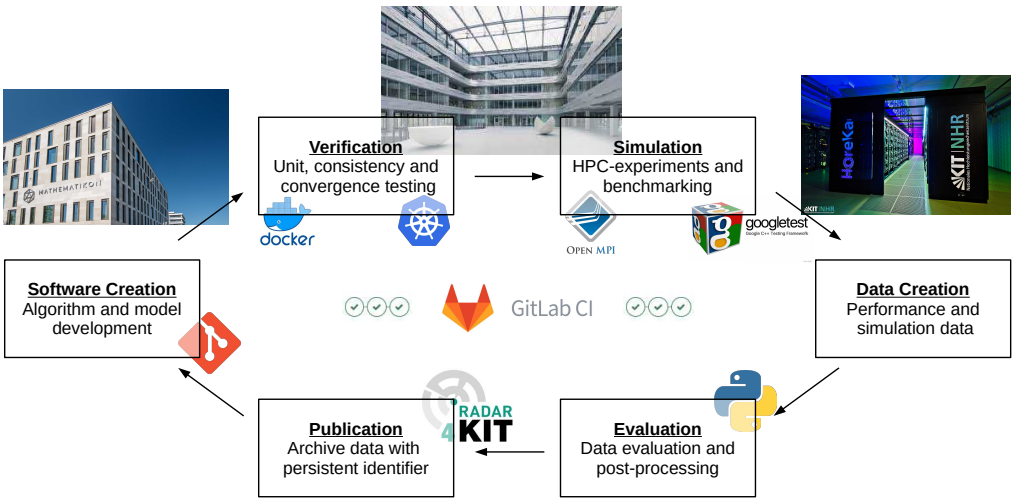

Figure 1: Release cycle of M++ with automated testing, benchmarking and performance data collection. Figure inspired by [16] and adapted from [4].

3 Examples

Defining a new benchmark is as simple as defining a new C++ function and registering it withing the Google benchmark framework (cf. Listing 1). The following two subsections will introduce the definition of the `EllipticSubsurfaceDiffusion3D` and the `AcousticGaussHatAndRicker2D` benchmark, which run among many others on the HoreKa supercomputing system and serve here as examples. While research software quite often favors specification over generalization, and therefore, tight integration of features over usability, we want to emphasize with these examples the user accessibility for running PDE related applications.

```
BENCHMARK(EllipticSubsurfaceDiffusion3D);                    // Registers 3D diffusion problem as benchmark

BENCHMARK(AcousticGaussHatAndRicker2D);                      // Registers 2D acoustic problem as benchmark

// ...                                                       // Register further benchmarks

int main(int argc, char **argv) {
 MppTest mppTest = MppTestBuilder(argc, argv).WithPPM(); // MppTest is wrapper around Google benchmark
 return mppTest.RUN_ALL_MPP_BENCHMARKS(argc, argv);      // Runs all benchmarks
}
```

Listing 1: Mockup main file collecting several PDE solver benchmarks

3.1 An Elliptic Subsurface Diffusion Benchmark in 3D

Firstly, we consider a 3D elliptic subsurface diffusion problem on a unit cube $\mathcal{D} = (0,1)^3$. Given a spatially dependent permeability $\boldsymbol{\kappa}\colon \mathcal{D} \to \mathbb{R}$, homogeneous Dirichlet boundary conditions on the bottom face $\Gamma_{\mathrm{D}} = \{\mathbf{x} \in \partial\mathcal{D}\colon x_2 = 0\}$ and an inflow Neumann boundary condition on the top face $\Gamma_{\mathrm{N}} = \{\mathbf{x} \in \partial\mathcal{D}\colon x_2 = 1\}$, the computational task is to find the pressure head $\mathbf{u}\colon \mathcal{D} \to \mathbb{R}$, such that

$$\begin{cases} -\operatorname{div}(\boldsymbol{\kappa}(\mathbf{x})\nabla\mathbf{u}(\mathbf{x})) &= 0 \quad \text{on} \quad \mathcal{D} \\ \mathbf{n}\cdot\nabla\mathbf{u}(\mathbf{x}) &= -1 \quad \text{on} \quad \Gamma_{\mathrm{N}} \\ \mathbf{u}(\mathbf{x}) &= 0 \quad \text{on} \quad \Gamma_{\mathrm{D}} \end{cases}.$$

The permeability function $\boldsymbol{\kappa}$ takes either the value one or one-hundred at 16 distinct locations (cf. [29] for definition). In M++ this problem can be implemented as a benchmark as shown in Listing 2. The

`EllipticPDESolver` class can be configured in various ways to solve the problem including different ansatz spaces for the FEM and various preconditioned linear solvers. In its default configuration, the benchmark uses a GMRES solver with an incomplete LU preconditioner for solving the linear system arising by using a standard Lagrange FEM of degree one. The benchmarking functionality in Listing 2 solves the PDE several times to collect statistically robust timing data by using the wrapper macros `BENCH_START_TIMING()` and `BENCH_END_TIMING(state)`, accounting for the run time imbalance of different MPI ranks.

```
static void EllipticSubsurfaceDiffusion3D(benchmark::State &state) {
 const int level = 4;                                          // Discretization level
 EllipticPDESolver pdeSolver();                                // Creates default solver

 auto pdeProblem = EllipticPDEProblemBuilder("Cube")           // Defines problem on unit cube
  .WithDirichletBoundary([](const Point &x) { return (x[1] == 0); }) // Defines Dirichlet boundary
  .WithNeumannBoundary([](const Point &x){ return (x[1] == 1); })    // Defines Neumann boundary
  .WithPermeability([](const Point &x) { return Permeability(x); })  // Confer repository for def.
  .WithSolution([](const Point &x) { return 0.0; })            // Solution on Dirichlet boundary
  .WithFlux([](const Point &x) { return {0.0, -1.0, 0.0}; })   // Flux on Neumann boundary
  .WithLoad([](const Point &x) { return 0.0; })                // Definition of right-hand-side
  .WithName("EllipticSubsurfaceDiffusion3D")                   // Name of problem
  .BuildShared();                                              // Creates problem shared pointer

 for(auto _ : state) {                                         // Multiple benchmark runs
  BENCH_START_TIMING();                                        // Starts timing for benchmark
  benchmark::DoNotOptimize(pdeSolver.Run(pdeProblem, level));  // Solves the problem on level
  BENCH_END_TIMING(state);                                     // Ends timing for benchmark
 }
}
```

Listing 2: Mockup implementation of a 3D elliptic subsurface diffusion benchmark in M++

3.2 An Acoustic Wave Propagation Benchmark in 2D

The PDE for the second benchmark example is an acoustic wave equation where we search a pressure and velocity component $(p, \mathbf{v})$: $[0,T] \times \mathcal{D} \to \mathbb{R} \times \mathbb{R}^2$ on a unit square $\mathcal{D} = (0,1)^2$ and with end time $T = 1$, such that we satisfy

$$\begin{cases} \partial_t \mathbf{v}(t,\mathbf{x}) - \nabla p(t,\mathbf{x}) &= \mathbf{f}(t,\mathbf{x}) & \text{on} & (0,T] \times \mathcal{D} \\ \partial_t p(t,\mathbf{x}) - \operatorname{div}(\mathbf{v}(t,\mathbf{x})) &= g(t,\mathbf{x}) & \text{on} & (0,T] \times \mathcal{D} \\ \mathbf{n} \cdot \mathbf{v}(t,\mathbf{x}) &= 0 & \text{on} & (0,T] \times \partial\mathcal{D} \\ \mathbf{v}(0,\mathbf{x}) &= 0 & \text{on} & \mathcal{D} \\ p(0,\mathbf{x}) &= 0 & \text{on} & \mathcal{D} \end{cases}$$

with homogeneous boundary and initial conditions. The right-hand side is given by a constant $\mathbf{f} \equiv \mathbf{0}$ and a separated $g(t,\mathbf{x}) = g_1(t)\, g_2(\mathbf{x})$. Here, $g_1(t)$ is a Ricker wavelet,

$$g_1(t) = 10\left(1 - \left(\tfrac{t}{a}\right)^2\right)\exp\left(-\tfrac{t^2}{2a^2}\right) \quad \text{with} \quad a = \tfrac{\pi}{10}, \quad t \in [0,1].$$

while $g_2(\mathbf{x})$ is a nascent delta function centered at $\mathbf{c} = (0.5, 0.75)^\top$ and with a diameter of $w = 0.1$. The nascent delta is multiplied with $\overline{g}_2$ such that $\|g_2\|_{\mathrm{L}^1(\mathcal{D})} = 1$, thereby it is given by

$$g_2(\mathbf{x}) = \begin{cases} \overline{g}_2 \exp\left(-\left(1 - \left\|\frac{\mathbf{x}-\mathbf{c}}{w}\right\|_2^2\right)^{-1}\right), & \|\mathbf{x}-\mathbf{c}\|_2 < w \\ 0, & \|\mathbf{x}-\mathbf{c}\|_2 \geq w \end{cases} \qquad \mathbf{x} \in \mathcal{D}.$$

A similar example is also used in [4, 5] where this wave propagates through randomly distributed material. As a result, this benchmark ensures the reproducibility of these studies and scaling results.

For an illustration of the pressure component p and its temporal development, we refer to Figure 2, for the implemented benchmark to Listing 3. Here, a DG discretization of degree two is used in combination with an implicit midpoint rule. The linear system in each time step is solved with a GMRES solver preconditioned by a point-block Jacobi iteration by default. The relation between the mesh width h and the time-step size τ is chosen as $\tau = C_{\mathrm{CFL}} h$ with $C_{\mathrm{CFL}} = 0.25$.

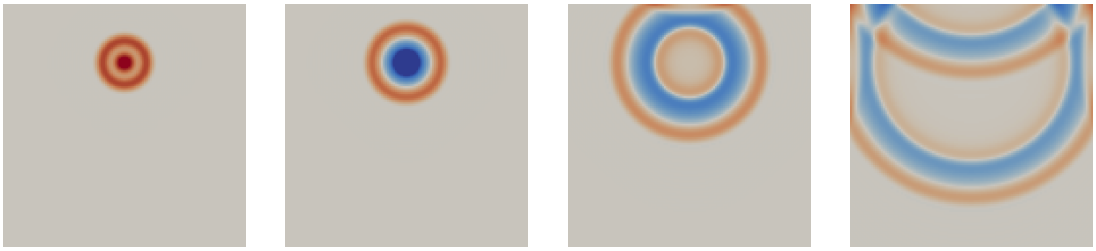

Figure 2: Pressure component p at different time steps

```
static void AcousticGaussHatAndRicker2D(benchmark::State &state) {
 const int level = 7;                                          // Discretization level
 auto pdeSolver = AcousticPDESolver(PDESolverConfig()          // Creates PDE solver with implicit
  .WithRkOrder(-2).WithDegree(2));                             // midpoint rule and DG degree two

 const double startTime = 0.0;                                 // Start time of problem
 const double gaussWidth = 0.1;                                // Width of GaussHat
 const double rickerDuration = 0.1;                            // Duration of Ricker wavelet
 const Point shotLocation = {0.5, 0.75};                       // Location of shot

 auto pdeProblem = AcousticPDEProblemBuilder("Square")         // Defines problem on unit square
  .WithForce([&](double t, const Point &x, COMPONENT comp) {   // Mockup for force term
    if (comp != COMPONENT::P0) { return 0.0; }                 // Homo. velocity right-hand-side
    return 10.0 * Ricker(t - startTime, rickerDuration)        // Defines right-hand-side for
       * GaussHat(dist(x, shotLocation), gaussWidth);          // Preassure component
  })
  .WithName("GaussHatAndRicker2D")                             // Name of problem
  .WithEndTime(1.0)                                            // Defines end time of problem
  .WithCFL(0.25)                                               // Defines CFL number of problem
  .BuildShared();                                              // Creates problem shared pointer

 for(auto _ : state) {                                         // Multiple benchmark runs
  BENCH_START_TIMING();                                        // Starts timing for benchmark
  benchmark::DoNotOptimize(pdeSolver.Run(pdeProblem, level));  // Solves the problem on level
  BENCH_END_TIMING(state);                                     // Ends timing for benchmark
 }
}
```

Listing 3: Mockup implementation of a 2D acoustic wave propagation benchmark in M++

4 Evaluation and Conclusion

The final evaluation of the benchmarking results, e.g. of the examples presented in Section 3, is part of the automated workflow as outlined in Section 2 and includes three levels. The first two levels are executed on a weekly basis, and prior to each release of M++. This includes over 60 benchmarks testing critical functionalities (like Krylov subspace methods or mesh generation) running on a single node using 64 CPUs for up to 3 hours in total. The results are evaluated using Python tools that generate historical performance plots for comparison, such that any performance change is noticeable in the deviations of these plots.

These functionality benchmarks are accompanied by larger integration benchmarks, based on applications from prior work [8, 10, 4, 20], and cover complete 2D and 3D simulations finishing within 10 minutes each and requiring up to 128 GB of memory. Previously observed performance is then imposed with some tolerance to the slurm job scheduler, such that if a job exceeds its limits, it is terminated and marked as failed.

The last level is only executed on demand. This includes all job scripts used in [11, 5, 6, 7] which are stored in the Git repository for reproducibility. Although these large-scale HPC experiments—using up to 16,384 CPUs over 12 hours—are not run regularly due to their cost, they remain executable via a simple pipeline trigger. We refer to the pipeline configurations for a detailed overview of which job has which CPU-time and memory requirements in order to pass[1].

We view software development, maintenance, documentation, and performance analysis as integral to the scientific method, enhancing research robustness and credibility. This paper outlines our approach to

[1] https://gitlab.kit.edu/kit/mpp/mpp/

sustainable research software development, focusing on new benchmarking capabilities and automation. We present two simple benchmarking examples to highlight the workflow's flexibility and ease. With this framework we support future software development, ensuring result reproducibility and code reliability.

5 Acknowledgements

The associated software development to the benchmarking framework has started during the student project "Praktikum Softwareentwicklung" at the Karlsruhe Institute of Technology (KIT). Nico Fröhlich, Noah Hackenjos, Clara Maier, Tobias Merkel and Simon Wülker have implemented new ways for in- and output file handling, while the latter two extended their work on the outlined benchmarking framework. Their work was funded by the CRC 1173 on wave phenomena and by the research group of Christian Wieners.

We further acknowledge the technical support by Mathias Reichardt as well as by the National High-Performance Computing Center (NHR) at KIT. All benchmarks are performed on the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research.

References

[1] H. Anzt, F. Bach, S. Druskat, F. Löffler, A. Loewe, et al. An environment for sustainable research software in germany and beyond - current state, open challenges, and call for action. F1000Research, 9, 2020.

[2] H. Anzt, T. Cojean, G. Flegar, F. Göbel, T. Grützmacher, P. Nayak, T. Ribizel, Y. M. Tsai, and E. S. Quintana-Ortí. Ginkgo: a modern linear operator algebra framework for high performance computing. ACM Trans. Math. Software, 48(1):Art. 2, 33, 2022.

[3] F. Bach, J. Klar, A. Loewe, J. Sánchez, G. Seemann, Y.-L. Huang, and R. Ulrich. The openCARP CDE–concept for and implementation of a sustainable collaborative development environment for research software. arXiv preprint arXiv:2201.04434, 2022.

[4] N. Baumgarten. A Fully Parallelized and Budgeted Multi-level Monte Carlo Framework for Partial Differential Equations. PhD thesis, Karlsruher Institut für Technologie (KIT), 2023.

[5] N. Baumgarten, S. Krumscheid, and C. Wieners. A fully parallelized and budgeted multilevel monte carlo method and the application to acoustic waves. SIAM/ASA Journal on Uncertainty Quantification, 12(3):901–931, 2024.

[6] N. Baumgarten, R. Kutri, and R. Scheichl. A budgeted multi-level monte carlo method for full field estimates of multi-pde problems, 2025.

[7] N. Baumgarten and D. Schneiderhan. Multilevel stochastic gradient descent for optimal control under uncertainty. arXiv preprint arXiv:2506.02647, 2025.

[8] N. Baumgarten and C. Wieners. The parallel finite element system M++ with integrated multilevel preconditioning and multilevel Monte Carlo methods. Comput. Math. Appl., 81:391–406, 2021.

[9] H. R. Bayat, J. Krämer, L. Wunderlich, S. Wulfinghoff, S. Reese, B. Wohlmuth, and C. Wieners. Numerical evaluation of discontinuous and nonconforming finite element methods in nonlinear solid mechanics. Comput. Mech., 62(6):1413–1427, 2018.

[10] T. Bohlen, M. R. Fernandez, J. Ernesti, C. Rheinbay, A. Rieder, and C. Wieners. Visco-acoustic full waveform inversion: from a DG forward solver to a Newton-CG inverse solver, 2021.

[11] D. Corallo, W. Dörfler, and C. Wieners. Space-time discontinuous Galerkin methods for weak solutions of hyperbolic linear symmetric Friedrichs systems. Technical Report 1, 2023.

[12] D. Corallo and C. Wieners. A parallel adaptive space-time discontinuousgalerkin method for transport in porous media. 2025.

[13] W. Dörfler, S. Findeisen, and C. Wieners. Space-time discontinuous Galerkin discretizations for linear first-order hyperbolic evolution systems. Comput. Methods Appl. Math., 16(3):409–428, 2016.

[14] W. Dörfler, S. Findeisen, C. Wieners, and D. Ziegler. Parallel adaptive discontinuous Galerkin discretizations in space and time for linear elastic and acoustic waves. 25:61–88, [2019] ©2019.

[15] J. Ernesti and C. Wieners. A space-time discontinuous Petrov-Galerkin method for acoustic waves. 25:89–115, [2019] ©2019.

[16] D. Farley. Continuous Delivery Pipelines - How to Build Better Software Faster. Pearson Education, 2020.

[17] J. Fröhlich, T. Gerach, J. Krauß, A. Loewe, L. Stengel, and C. Wieners. Numerical evaluation of elasto-mechanical and visco-elastic electro-mechanical models of the human heart. GAMM-Mitteilungen, 46(3-4):e202370010, 2023.

[18] J. Fröhlich. A segregated finite element method for cardiac elastodynamics in a fully coupled human heart model. PhD thesis, Karlsruher Institut für Technologie (KIT), 2022.

[19] T. Gerach, S. Schuler, J. Fröhlich, L. Lindner, C. Wieners, and A. Loewe. Electro-mechanical whole-heart digital twins a fully coupled multi-physics approach. Mathematics, 9(11):1247, 2021.

[20] M. Hochbruck, T. Pažur, A. Schulz, E. Thawinan, and C. Wieners. Efficient time integration for discontinuous Galerkin approximations of linear wave equations [Plenary lecture presented at the 83rd Annual GAMM Conference, Darmstadt, 26th–30th March, 2012]. ZAMM Z. Angew. Math. Mech., 95(3):237–259, 2015.

[21] M. M. Knodel, S. Kräutle, and P. Knabner. Global implicit solver for multiphase multicomponent flow in porous media with multiple gas components and general reactions: global implicit solver for multiple gas components. Comput. Geosci., 26(3):697–724, 2022.

[22] C. Rheinbay. All-At-Once and Reduced Solvers for Visco-Acoustic Full Waveform Inversion. PhD thesis, Dissertation, Karlsruhe, Karlsruher Institut für Technologie (KIT), 2023, 2023.

[23] K. Schulz, L. Wagner, and C. Wieners. A mesoscale continuum approach of dislocation dynamics and the approximation by a Runge-Kutta discontinuous Galerkin method. International Journal of Plasticity, 120:248–261, 2019.

[24] E. Thawinan. Numerical approximation of higher-dimensional Continuum Dislocation Dynamics theory in single crystal plasticity. PhD thesis, 2015.

[25] L. Wagner. A discontinuous Galerkin method for continuum dislocation dynamics in a fully-coupled elastoplasticity model. PhD thesis, Karlsruher Institut für Technologie (KIT), 2019.

[26] C. Wieners. Distributed point objects. A new concept for parallel finite elements. In Domain decomposition methods in science and engineering, volume 40 of Lect. Notes Comput. Sci. Eng., pages 175–182. Springer, Berlin, 2005.

[27] C. Wieners. A geometric data structure for parallel finite elements and the application to multigrid methods with block smoothing. Comput. Vis. Sci., 13(4):161–175, 2010.

[28] C. Wieners. A space-time discontinuous galerkin discretization for the linear transport equation. Computers & Mathematics with Applications, 152:294–307, 2023.

[29] C. Wieners, D. Corallo, D. Schneiderhan, L. Stengel, H. D. N. Pham, and N. Baumgarten. Mpp 3.4.1, 2024.

[30] C. Wieners, D. Corallo, D. Schneiderhan, L. Stengel, H. D. N. Pham, and N. Baumgarten. Mpp 3.5.0, 2025.

[31] J. M. Wunderlich. Computer-assisted Existence Proofs for Navier-Stokes Equations on an Unbounded Strip with Obstacle. PhD thesis, Karlsruher Institut für Technologie (KIT), 2022.

End-user data analysis at the LHC*

Nils Faltermann Manuel Giffels Günter Quast

Karlsruhe Institute of Technology (KIT)

Abstract

The Large Hadron Collider (LHC), located at CERN in Geneva, stands as one of the most monumental scientific experiments in human history. This remarkable machine facilitates approximately 40 million particle collisions every second, generating an astronomical amount of data. Even with rigorous filtering of collision events, the data retained for subsequent analysis remains staggering in scale. In addition to the recorded data, conducting a successful physics analysis demands an extensive set of simulations that can be compared to the recorded events. In this presentation, we will delve into our approach to incorporating external resources, such as the NEMO cluster in Freiburg, into our local batch system. This integration greatly enhances accessibility for the complex workflows required for physics data analyses.

1 Introduction

The Large Hadron Collider (LHC) is the world's largest and most powerful particle accelerator. It is located at the European Organization for Nuclear Research (CERN) near Geneva, Switzerland. The LHC is a multinational scientific endeavor involving thousands of physicists and engineers from over 100 countries around the world. The LHC accelerates beams of protons or heavy ions to extremely high energies. These particles are guided around the ring by powerful magnetic fields generated by the superconducting magnets. The beams travel in opposite directions and are made to collide at four main detector sites (ATLAS, CMS, ALICE, and LHCb), where the outcome of the particle collisions is recorded. The primary purpose of the LHC is to explore the fundamental properties of particles and the forces that govern them. By colliding high-energy particles together at nearly the speed of light, scientists aim to recreate conditions similar to those just nanoseconds after the Big Bang, providing insights into the origins of the Universe.

2 Experimental Setup

Particle detectors are designed to measure various properties of particles produced in high-energy collisions. These detectors are complex instruments consisting of multiple subsystems designed to track the paths of charged particles, measure their energies, and identify different types of particles. They collect vast amounts of data from collision events, which include information such as particle trajectories, energy deposits, and timing signals. The data collected by the detectors provide insights into the behavior of fundamental particles and the interactions between them.

The CMS (Compact Muon Solenoid) detector is one of the two large general-purpose detectors located at the LHC. It is a cylindrical apparatus with a mass of 14,000 tonnes, about 15 meters in diameter and 21 meters in length, designed to observe a wide range of particles and phenomena produced by high-energy proton-proton collisions. It consists of multiple layers of sub-detectors arranged concentrically around the collision point.

The LHC operates proton-proton collisions at very high rates. The nominal collision rate at the LHC is around 40 million proton-proton collisions per second per experiment. With around 55 million readout channels of the CMS detector, this results in approximately a data rate of $1\,\mathrm{PB/s}$, too much for any storage system to cope with. For this reason, the first step of the data acquisition process is the trigger system, which is designed

*Proceedings of the 9th bwHPC-Symposium 2023. Mannheim Conference Series (MaConf). 2025.

to select collision events of interest for further analysis and drop the rest. The CMS trigger system consists of two levels: the Level-1 trigger (L1) and the High-Level Trigger (HLT). The L1 trigger is implemented in hardware and makes rapid decisions based on information from the calorimeters and muon detectors. It reduces the event rate from the LHC's collision rate of 40 million events per second to around 100,000 events per second. The events passing the L1 trigger are further processed by the HLT, which is implemented in software running on a large farm of computers. The HLT performs more sophisticated event reconstruction and selects events for storage based on specific physics criteria. It further reduces the event rate to a manageable level for offline analysis, around 2,000 events per second. Despite the significant reduction in event rates, the total amount of data in a year that needs to be recorded and processed amounts to at least $20\,\mathrm{PB}$ for each of the detectors at the LHC.

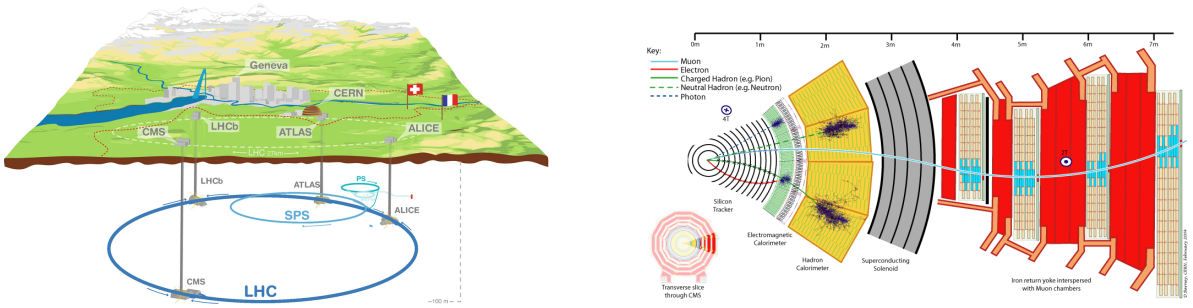

Figure 1: Illustration of the LHC tunnel and its surrounding area with the location of the main particle detectors (left) [1] and a vertical slice of the CMS detector (right) [2], demonstrating the potential path of particles produced in the proton-proton collisions and their signatures.

3 Computing Model

The Worldwide LHC Computing Grid (WLCG) is a global computing infrastructure established to handle the massive amounts of data produced by the different LHC experiments. The WLCG represents a collaboration among thousands of scientists, engineers, and computing experts from research institutions and universities around the world. It involves more than 170 computing centers distributed across more than 40 countries, forming a global network of computing resources for storing, processing, and analyzing data.

The WLCG is organized in a hierarchical, tiered structure: The Tier-0 center is located at CERN itself. It serves as the initial data collection point where raw data from the LHC experiments is stored and distributed to Tier-1 centers for further processing and analysis. Tier-1 centers are major computing facilities located around the world, such as GridKa in Karlsruhe. They are responsible for storing and processing a significant portion of the LHC data, providing computing resources, storage capacity, and networking infrastructure to support the needs of the global particle physics community. Tier-2 centers are regional computing facilities located at universities and research institutions. They serve as hubs for data analysis and simulation tasks, providing computing resources and storage capacity to physicists and researchers within their respective regions. The final physics data analysis is usually carried out at so-called unofficial Tier-3 centers, which consist of local institute clusters.

While there is a constant workload in terms of processing and simulation jobs on higher-level tiers due to centrally organized workflow management systems, the situation at Tier-3 clusters is usually different: The end-user working on physics analyses has different and diverse requirements:

- Processing of data and simulation, deriving new quantities and condensing the information into histograms. Medium to high CPU usage, but usually very high I/O requirements. Typical use case for data-intensive computing.
- Tasks that require processing data multiple times, for example statistical inference and training of machine learning methods. Very high CPU usage, but moderate I/O requirements. Referred to as

CPU-intensive computing.

- The demand for resources is not constant over time. Because of the limited amount of users and workflow of analyses, i.e., first developing and implementing algorithms that are then applied to data, resources are usually requested in bursts. This results in a situation where either not enough resources are available for the demand or resources are idling.

The infrastructure hosted at institute clusters is typically built homogeneously, similar to a higher-tier site in the WLCG, with a tendency towards CPU-intensive computing to support also many other applications outside of high-energy physics. On top of that it cannot be justified to spend the already sparse funding in research for the purchase of additional resources that would not be utilized to the full extent. Given the user requirements and the reality at local institutes, this situation is a prime example of a use case where so-called opportunistic resources can provide the solution. Opportunistic resources are resources that are not permanently dedicated to but temporarily available for a specific task, user or group. With the integration of opportunistic resources, the resource pool can also be made more heterogeneous, thus providing better-suited resources for individual demand, e.g., external GPUs for specialized machine learning applications. As the name already implies, opportunistic resources are also only allocated when there is demand for them, which for instance can also be realized by booking resources from cloud providers.

4 Dynamic Resource Integration

To run user jobs on WLCG resources, placeholder jobs, called pilots, are employed, which interact with an overlay batch system (OBS) to dynamically register available resources to a resource pool. The user only interacts with this OBS and is therefore transparent to the underlying site. This concept works fine in the homogeneous environment of the WLCG sites, where the hardware and software are tailored to the needs of the community. Naturally, the situation for external resources is different. For this reason, the pilot concept is extended to the so-called drone. Depending on the resource, the drone can run natively on the resource as a batch job, or can provide an additional layer of abstraction in terms of a virtual machine or a container to provide the required software environment. The demand and request for additional resources at an external site is managed by COBalD - the Opportunistic Balancing Daemon (COBalD) [3, 4] by aggregating similar drones into an abstract pool. The decision to increase or decrease the resource demand in a pool involves also the utilization of the current resources in this pool. Drones themselves are managed via the Transparent Adaptive Resource Dynamic Integration System (TARDIS) [5, 6], which directly interacts with COBalD. Each site can run an individual TARDIS instance to facilitate specialized configurations and react independently on site-specific circumstances. The TARDIS workflow is illustrated in Fig. 2.

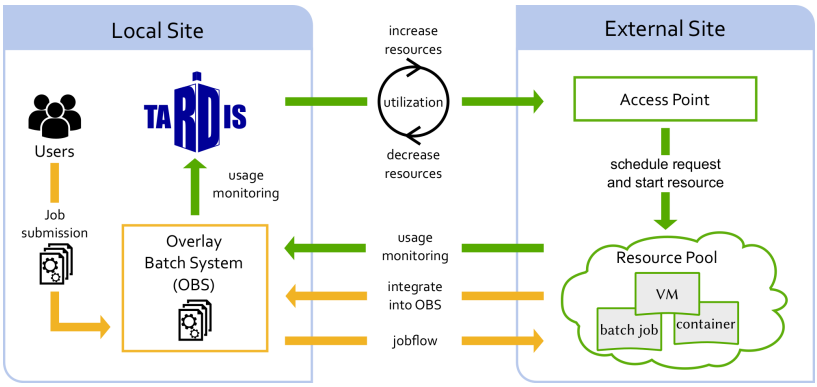

Figure 2: Illustration of *The Transparent Adaptive Resource Dynamic Integration System* (TARDIS) and its working principle to incorporate opportunistic resources.

TARDIS is successfully deployed to the OBS instance (HTCondor) at the Institute of Experimental Particle Physics (ETP), which is part of the Karlsruhe Institute of Technology (KIT). Resources that have been inte-

grated so far consist of research clusters such as the bwForCluster (NEMO), TOpAS (*Throughput Optimized Analysis System*, GridKa), ForHLRII (KIT), HoreKa (KIT), as well as well as commercial cloud providers such as Open Telekom Cloud, Exoscale and 1&1. In Fig. 3 the batch system utilization at the ETP is shown over the time of a week. Local resources were always used in this case but only made up a small fraction of all used resources. During this particular time, many jobs were submitted to the batch system and thus opportunistic resources were allocated to serve this demand, as can be seen by the rising amount of opportunistic resources whenever new jobs were queued in the batch system. Notably at some point during the week, the utilization exceeded the total share of the CMS experiment at the Tier-1 site GridKa.

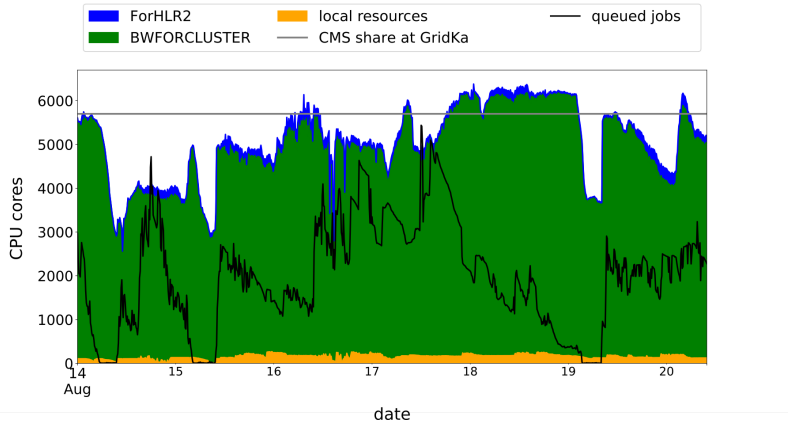

Figure 3: Example usage of the local batch system at the Institute of Experimental Particle Physics (ETP). Shown are the static local resources in yellow and the dynamically integrated opportunistic resources in blue (ForHLR II) and green (bwForCluster NEMO).

5 Scientific Results

During recent years the integration of opportunistic resources enhanced the computing capabilities of the ETP institute cluster significantly. In total, there are more than 20 journal publications up to this day that directly benefited from the additional computing power. The most recent ones involving the CMS experiment are outlined shortly in the following.

In light of the currently ongoing Run 3 of the LHC at $\sqrt{s} = 13.6\,\mathrm{TeV}$ it is important to first look into well-known physics processes to understand and calibrate the detector, particularly after a longer shutdown and maintenance period. This also needs to happen quickly to identify potential problems with the data-taking to resolve them as soon as possible. One of these so-called standard candles is the production of a Z boson that decays into a pair of muons. This process has been measured successfully with high precision [7], paving the way for future Run 3 precision measurements. The result of the measurement is also highlighted in Fig. 4 (left). Especially in the context of an early measurement, where frequent reprocessing of data happens, the addition of opportunistic resources helps for a fast turnaround.

In contrast to the previous example, the measurement of ttH and tH production with H $\rightarrow$ bb decays [8] is targeting an already completed data-taking campaign to probe the coupling of the Higgs boson to the top quark. With the complex final state consisting of many jets, fully reconstructing events becomes a combinatorial task, for which dedicated machine learning algorithms are employed. To extract the results elaborate and computing-intensive statistical calculations are necessary, e.g., maximum-likelihood estimations and multidimensional likelihood scans for various parameters of interest. One of these results is exemplary shown in Fig. 4 (right), where the coupling of the Higgs boson to the top quark and vector bosons is probed simultaneously to derive exclusion limits.

The search for additional Higgs bosons and leptoquarks in $\tau\tau$ final states [9] directly searches for possible extensions to the standard model of physics. To model the already-known background process of a Z boson

decaying into a pair of τ leptons, the analysis utilizes a refined method called τ-embedding. This method is a data-simulation hybrid approach, where data events with a Z boson decaying into a pair of muons are cleaned of the muon signature, which is then replaced with simulated τ decays and their interaction with the detector material. The latter part is a highly computationally intensive task and the bottleneck for simulated events in general, but the use of opportunistic resources significantly speeds up the task.

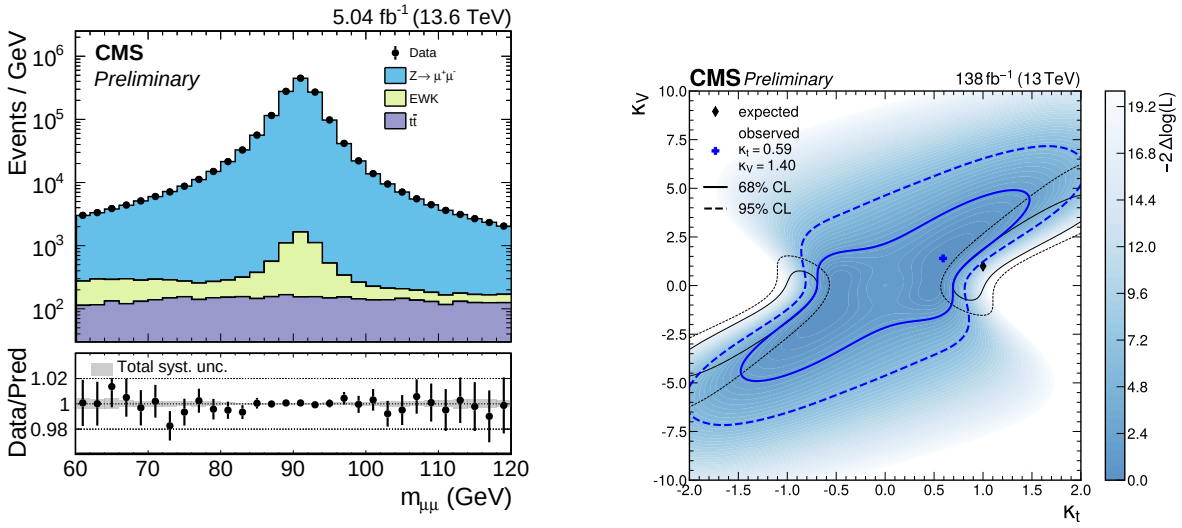

Figure 4: Exemplary scientific results that have been obtained with the help of opportunistic resources. Shown is the invariant mass distribution of the dimuon system in the measurement of the Z boson cross section at $\sqrt{s} = 13.6\,\mathrm{TeV}$ [7] (left) and the two-dimensional likelihood scan of the coupling modifiers for the top quark κ_t and for vector bosons κ_V in the combined measurement of ttH and tH production at $\sqrt{s} = 13\,\mathrm{TeV}$ [8] (right).

6 Conclusions

The enormous data recorded at the Large Hadron Collider (LHC) is a computational challenge on many fronts. While data processing and simulation are mostly automated on dedicated sites, the physics analysis as an end-user is mostly done on conventional computing clusters. To provide additional resources that are transparent to use the COBalD/TARDIS system was developed and successfully deployed to our local batch system at the Institute of Experimental Particle Physics (ETP). The demand for computing resources at the LHC will increase greatly during the next decade when the LHC will be upgraded to the High-Luminosity LHC, where each collision event will contain significantly more particles to analyze. The integration of opportunistic resources can be one of the cornerstones to overcome this upcoming obstacle.

7 Acknowledgements

The authors acknowledge support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 39/963-1 FUGG (bwForCluster NEMO).

References

[1] Overall view of the LHC, Service graphique, CERN, OPEN-PHO-CHART-2014-006

[2] Interactive Slice of the CMS detector, Siona Ruth Davis, CMS Collaboration, CMS-OUTREACH-2016-027

[3] Lightweight dynamic integration of opportunistic resources, M. Fischer et al., EPJ Web Conf. 245 (2020) 07040

[4] MatterMiners/cobald: v0.14.0, M. Fischer et al., Zenodo, doi: 10.5281/zenodo.8199049

[5] Effective Dynamic Integration and Utilization of Heterogenous Compute Resources, M. Fischer et al., EPJ Web Conf. 245 (2020) 07038

[6] MatterMiners/tardis: 0.8.1, M. Giffels et al., Zenodo, doi: 10.5281/zenodo.10411024

[7] Measurement of the inclusive cross section of Z boson production in pp collisions at $\sqrt{s} = 13.6\,\mathrm{TeV}$, CMS Collaboration, CMS-PAS-SMP-22-017, to be submitted to PLB

[8] Measurement of the ttH and tH production rates in the H $\rightarrow$ bb decay channel with $138\,\mathrm{fb}^{-1}$ of proton-proton collision data at $\sqrt{s} = 13\,\mathrm{TeV}$, CMS Collaboration, CMS-PAS-HIG-19-011, to be submitted to JHEP

[9] Searches for additional Higgs bosons and for vector leptoquarks in $\tau\tau$ final states in proton-proton collisions at $\sqrt{s} = 13\,\mathrm{TeV}$, CMS Collaboration, JHEP 07 (2023) 073

bwFDM – The Federal State Initiative for Research Data Management in Baden-Württemberg*

Cora F. Krömer
KIT Library
Karlsruhe Institute of Technology

Fabian Schubö
KIT Library
Karlsruhe Institute of Technology

Abstract

This paper provides an overview of the federal state initiative for research data management in Baden-Württemberg (bwFDM) and its multi-faceted approach to addressing the challenges and opportunities presented by research data. bwFDM's activities include the establishment of networks and collaborations, the dissemination of information and knowledge, the expansion of information services, the provision of training and consultation opportunities, and the organisation of the conference series E-Science-Tage. Overall, bwFDM's activities emphasise the importance of collaboration, information dissemination, and training in shaping the present and future of research data management within and beyond Baden-Württemberg.

1 Introduction

In the contemporary research landscape, the exponential growth of data presents both opportunities and challenges. Effectively managing, preserving and sharing research data is essential to ensuring the verifiability, reproducibility and impact of academic research. Hence Research Data Management (RDM) has emerged to address these challenges. RDM encompasses a multitude of strategies, policies, and practices designed to ensure the findability, accessibility, interoperability and (re)usability of research data throughout its lifecycle. From the moment data is generated to its eventual archival and beyond, RDM seeks to provide a framework for organizing, storing, and disseminating data in ways that facilitate collaboration, reproducibility, and reusability.

Recognising the importance of RDM, the Ministry of Science, Research and Arts Baden-Württemberg has been funding state-wide projects to coordinate RDM activities since 2014. In 2023, bwFDM – the federal state initiative for research data management in Baden-Württemberg [1] – has received funding for four years to further promote practices and foster cooperation in RDM within the region of Baden-Württemberg. bwFDM is organised at three universities (Karlsruhe Institute of Technology, University of Konstanz, Heidelberg University).

This paper aims to provide a comprehensive overview of bwFDM's activities and their profound impact on research data management in Baden-Württemberg and beyond. From establishing networks and partnerships to providing information and disseminating knowledge, from expanding information services to offering training opportunities, bwFDM's multi-faceted approach to RDM aims to shape the research landscape and to empower stakeholders in their RDM activities. Several of the services offered by bwFDM are relevant to research groups, bwHPC service providers and HPC support centres. A central aim is to connect these stakeholders with RDM consultants and other stakeholders from the research data infrastructure in Baden-Württemberg.

2 Establishing Networks and Collaborations

At the core of bwFDM's mission lies the recognition that effective research data management (RDM) is not a solitary endeavour but a collaborative effort that thrives on synergies and partnerships.

*Proceedings of the 9th bwHPC-Symposium 2023. Mannheim Conference Series (MaConf). 2025.

To achieve this mission, bwFDM is committed to fostering networks among stakeholders involved in RDM. By bringing together researchers, high-performance computing (HPC) experts, support centres, and RDM consultants, bwFDM creates a platform for exchange and collaboration. Through strategic alliances and ongoing dialogue with key projects such as bwHPC-S5, bwFDM seeks to identify common challenges, share best practices, and foster a culture of collaboration focused on advancing RDM practices across diverse domains and disciplines.

Moreover, bwFDM recognises the importance of ongoing dialogue and knowledge exchange to drive continuous improvement in RDM practices. Through training activities, working groups, the information platform forschungsdaten.info [2], and the conference series E-Science-Tage [3], bwFDM provides opportunities for stakeholders to share their experiences, learn from each other, and collectively chart the course for the future of RDM in Baden-Württemberg and beyond. Of these offerings, mainly forschungsdaten.info and the conference series E-Science-Tage have a significant impact on the national and international RDM landscape. By fostering a culture of collaboration and knowledge sharing, bwFDM aims not only to address immediate RDM challenges, but also to lay the groundwork for sustainability in data management practices.

3 Providing Information and Knowledge Dissemination

One of the central bwFDM activities to promote RDM practices is forschungsdaten.info, a prestigious information platform, which is an extensive collection of knowledge, resources, and tools meticulously curated to cater to the diverse needs of the German-speaking RDM community. bwFDM is editor-in-chief of forschungsdaten.info, which brings together a board of 70 editors from 30 institutions. Thus, the platform fosters a culture of knowledge sharing and collaboration across the German-speaking countries.

With its user-friendly interface and large database, forschungsdaten.info offers RDM beginners and RDM experts a wealth of insights into RDM topics, services, and tools, tailored to specific academic fields and research disciplines. forschungsdaten.info serves as more than just a registry; it is a dynamic hub of information, constantly evolving to keep pace with the latest developments in the RDM landscape. Through curated articles, forschungsdaten.info equips stakeholders with the knowledge and tools they need to navigate the complexities of RDM effectively. Whether it is guidance on data management planning, insights into metadata standards, or tips for data publication, forschungsdaten.info provides invaluable resources to support stakeholders at every stage of the data lifecycle. Through regular newsletters [4], webinars (forschungsdaten.info live [5]), news [6] and event announcements [7], forschungsdaten.info ensures that stakeholders remain informed and engaged in shaping the present and the future of RDM.

4 Expanding Information Services

As the landscape of research data management (RDM) continues to evolve, bwFDM remains committed to anticipating emerging trends and evolving needs within Baden-Württemberg's research community. bwFDM is actively developing its information services to better serve its stakeholders and to foster collaboration across institutions and disciplines.

At the centre of this expansion effort is a comprehensive mapping initiative aimed at charting the RDM landscape in Baden-Württemberg. This involves identifying and cataloguing RDM stakeholders (all research and higher education institutions), services, and tools across the region, providing stakeholders with a detailed overview of the resources available to them. By creating a centralized registry of RDM resources, bwFDM aims to streamline access to essential tools and services, facilitating collaboration and knowledge exchange among researchers and support staff.

Through ongoing engagement with stakeholders and feedback mechanisms, bwFDM will ensure that the mapping results remain relevant and up to date, reflecting the latest developments in RDM services. This iterative approach allows bwFDM to identify gaps, address challenges, and tailor its information services to meet the diverse needs of the research community effectively.

Central to bwFDM's information services expansion is its website bwfdm.de [1]. This website will provide stakeholders with easy access to a number of different resources, including an interactive map of consulting

teams and RDM services [8], a helpdesk and information about our training offerings [9] designed to support their RDM activities.

5 Training and Consultation

Recognising the critical importance of equipping support personnel and researchers with the necessary skills and knowledge to effectively manage research data, bwFDM has developed the RDM training program Certificate of Advanced Studies Forschungsdatenmanagement (CAS FDM) [10], starting in October 2025. This program addresses the growing demand for structured RDM training and empowers individuals with the necessary expertise to navigate the complexities of data management within higher education and research institutions.

The CAS FDM primarily targets present or future infrastructure employees, including data stewards and RDM consultants. Its structure is intended to enable connectivity with existing qualification formats and opportunities already available.

The CAS FDM involves a comprehensive curriculum covering a diverse range of topics essential to proficient data management practices. Participants will delve into fundamental concepts such as data management planning, metadata standards, and data sharing protocols, gaining a thorough understanding of the principles underpinning effective data stewardship. In addition, the course explores ethical considerations surrounding data collection, storage, and dissemination, instilling a strong sense of responsibility and integrity in data management practices.

Furthermore, bwFDM provides consultation on research data management issues via its helpdesk [11]. The helpdesk's aim is to offer a first, easily accessible point of contact for questions relating to RDM in Baden-Württemberg and also to cover the consulting needs of researchers and support staff whose institutions do not yet have a dedicated RDM service team. In addition to answering questions directly, the helpdesk also serves as a mediator by directing enquirers to discipline-specific or service-specific helpdesks.

6 The E-Science-Tage Conference Series

A highlight of bwFDM's outreach efforts is the biennial conference series E-Science-Tage in Heidelberg, renowned for its interdisciplinary focus on RDM and Open Science. The conference series serves as an important platform for professional exchange, bringing together researchers, support staff and policymakers from across Baden-Württemberg and beyond. Through its engaging programme of keynote speeches, panel discussions and workshops, E-Science-Tage provides participants with a unique opportunity to explore emerging trends, share best practices, and showcase innovative solutions in RDM and Open Science.

By providing a forum for stakeholders to engage in dialogue, exchange ideas between academia and technology, and forge partnerships, E-Science-Tage facilitates the co-creation of strategies and initiatives aimed at advancing RDM practices and promoting Open Science principles. Submissions on topics related to HPC are most welcome, in particular those fostering collaborations between RDM and HPC initiatives.

Conference materials, including videos of keynote talks, presentation slides and posters are made available after each conference to facilitate the sharing, evaluation and preservation of information and research findings [12]. Furthermore, E-Science-Tage invites authors to submit their contributions for publication in a conference proceedings [13]. The submission and review process takes place after the conference, allowing authors to incorporate feedback received during the conference sessions. The submissions are reviewed by at least two reviewers from the field of RDM, which helps ensure a high quality standard. Altogether, the different formats of presentation, discussion, and publication provide an outstanding outreach for the contributions accepted to the conference.

7 Conclusions

In conclusion, the contemporary research landscape is characterized by the exponential growth of data, presenting both opportunities and challenges for research. Research data management has emerged as a critical discipline aimed at addressing these challenges and ensuring the integrity, accessibility, and impact of research data.

The Ministry of Science, Research and Arts Baden-Württemberg recognises the significance of RDM and has been funding state-wide RDM projects since 2014. bwFDM, the federal state initiative for research data management in Baden-Württemberg has received funding for four more years (2023-2027) to promote practices and foster collaboration in RDM within the region and beyond.

This paper has provided a comprehensive overview of bwFDM's activities, highlighting its role in establishing networks and collaborations, providing information and knowledge dissemination, expanding information services, offering training and education opportunities, and organizing the conference series E-Science-Tage. These endeavours are also meant to facilitate the advancement of collaboration between projects and stakeholders related to RDM and HPC.

8 Acknowledgements

The federal state initiative for research data management bwFDM is funded by the Ministry of Science, Research and Arts Baden-Württemberg.

References

[1] https://bwfdm.de/.

[2] https://forschungsdaten.info/.

[3] https://e-science-tage.de/.

[4] https://forschungsdaten.info/praxis-kompakt/forschungsdateninfo-aktuell/.

[5] https://forschungsdaten.info/praxis-kompakt/forschungsdateninfo-live/.

[6] https://forschungsdaten.info/nachrichten/.

[7] https://forschungsdaten.info/kalender-index/.

[8] https://bwfdm.de/fdm-in-bw/.

[9] https://bwfdm.de/beratung-service/.

[10] https://afww.uni-konstanz.de/de/fdm/certificate-advanced-studies-fdm/.

[11] Enquiries can be sent to support@bwfdm.de.

[12] https://e-science-tage.de/en/downloads.

[13] See the conference proceedings from the E-Science-Tage 2023: https://doi.org/10.11588/heibooks.1288.

cellular_raza - Novel Flexibility in Design of Agent-Based Models in Cellular Systems*

Jonas Pleyer, Christian Fleck

Freiburg Center for Data Analysis, Modeling and AI
University of Freiburg

21.11.2024

Abstract

This paper uses `cellular_raza` to develop a model with cell-type specific interactions whereby cells self-assemble into regions of similar species which is also known as cell-sorting. We use this model to asses the parallelization performance of the numerical backend at the core of `cellular_raza` and show that values of up to $p = 97.78 \pm 0.14\%$ parallelizable code can be achieved, which indicates a high level of parallelizability.

1 Introduction

`cellular_raza` is a cellular agent-based modeling framework [1] which allows researchers to construct models from a clean slate. In contrast to other agent-based modelling toolkits, it is free of assumptions about the underlying cellular representation. This enables researchers to build up complex models while retaining full control over every parameter and behaviour introduced.
At the 9th bwHPC Symposium 2023, we presented a very compacted variety of applications and analytics of `cellular_raza` [2]. Here, we focus on the cell-sorting example which was also presented but has been refined since then. We refer the interested reader to the documentation website cellular-raza.com where one can find the full documentation, remaining examples and information about the underlying assumptions and methods which we leveraged in the development of `cellular_raza`.

2 Cell Sorting

Cell Sorting is a naturally occuring phenomenon which drives many biological processes [3, 4]. While the biological reality can be quite complex, it is rather simple to describe such a system in its most basic form. The underlying principle is that interactions between cells are specific with respect to their type.

Mathematical Description We assume that our cells are spherical objects which interact via force potentials. The values R_i, R_j are the radii of the cells ($i \neq j$) interacting with each other. For simplification, we can assume that they are identical $R_i = R_j = R$. The two positions of cells are x_i, x_j and their distance is $r_{i,j} = ||x_i - x_j||$. Furthermore, we assume that the equation of motion is given by

$$\partial_t^2 \vec{x}_i = \vec{F}_i - \lambda \partial_t \vec{x}_i \tag{1}$$

where the first term is the usual force term $\vec{F}_i = -\sum_j \vec{\nabla} V_{i,j}$ obtained by differentiating the given potential and the second term is a damping term which arises due to the cells being immersed inside a viscuous fluid.

*Proceedings of the 9th bwHPC-Symposium 2023. Mannheim Conference Series (MaConf). 2025.

Parameter	Symbol	Value
Cell Radius	R_i	$6\,\mu\mathrm{m}$
Potential Strength	V_0	$2\,\mu\mathrm{m}^2/\mathrm{min}^2$
Damping Constant	λ	$2\,\mathrm{min}^{-1}$
Interaction Range	ξ	$1.5(R_i + R_j) = 3R_i$
Time Stepsize	Δt	$0.2\,\mathrm{min}$
Time Steps	N_t	$10'000$
Domain Size	L	$110\ \mu\mathrm{m}$
Cells Species 1	$N_{C,1}$	800
Cells Species 2	$N_{C,2}$	800

Table 1: This table shows two sections containing parameters and other variables respectively which to fully specify and initialize the system. In total, 1600 cells with random initial positions and zero velocity were placed inside the domain.

Note that we rescaled the units of our parameters in order to normalize our mass to $m = 1$. This means, every parameter can be expressed in units of length and time.
We can assume that interactions between cells are restricted to close ranges and thus enforce a cutoff $\xi \geq R_i + R_j$ for the interaction where the resulting force is identical to zero. We further assume that cells of different species do not attract each other but do repel. To describe this behaviour, we set the potential to zero when $r > R_i + R_j$ (i.e., $\kappa_{i,j} > 1$) and both cells have distinct species type s_i.

$$\kappa_{i,j} = \frac{r_{i,j}}{R_i + R_j} \tag{2}$$

$$U_{i,j} = V_0 \left(\frac{1}{3\kappa_{i,j}^3} - \frac{1}{\kappa_{i,j}} \right) \tag{3}$$

$$V_{i,j} = \begin{cases} 0 & \text{if } \kappa_{i,j} \geq \xi/(R_i + R_j) \\ 0 & \text{if } s_i \neq s_j \text{ and } \kappa_{i,j} \geq 1\ . \\ U(\kappa_{i,j}) & \text{else} \end{cases} \tag{4}$$

Parameters In total, we are left with only 4 cellular parameters to describe our system. In order to fully specify the system, we also need to define the time for which we are solving, the overall physical size and the initial number of agents which we put into the simulation. Their values are given in Table 1.

Results Figure 1 shows the inital placement of the cells and their final state. We can clearly see that the cells have assembled into connected regions of the same species. The size of these regions depends on the interaction range and its strength. In this example, we did not assume any stochastic motion of the cells. It is to be noted that this assumption is not required by `cellular_raza` but rather a free choice of the model. Depending on the desired complexity, users can substantially modify the cellular representation, which is the main reason for the development of `cellular_raza`. We provide two additional examples which implement mechanics with Brownian [5] and Langevin [5, 6] dynamics but even more complex cellular representations are possible as well (see eg. bacterial-rods).

3 Multithreading Performance (Amdahl's Law)

Theory One measure of multithreaded performance is to calculate the possible theoretical speedup given by Amdahl's law [7]. It provides an estimate for the speedup and assumes that the workload can be split into a parallelizable and non-parallelizable part which is quantified by $0 \leq p \leq 1$. A higher value means that the contribution coming from non-parallelizable algorithms is lower. The theoretical maximum $p = 1$ means that all of the executed code is parallelizable. Amdahl's law is given by

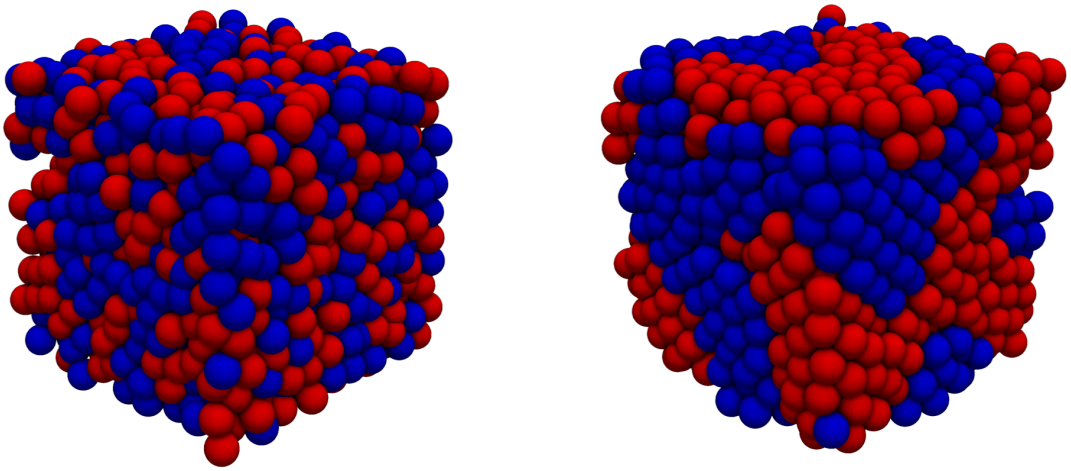

Figure 1: Cells are initially placed randomly inside the cuboid simulation domain. After the simulation has finished, the cells have assembled into connected regions of the same species.

$$T(n) = T_0 \frac{1}{(1-p) + \frac{p}{n}} \tag{5}$$

where $T(n)$ describes the throughput which can be achieved given n parallel threads and the variable p is the relative proportion of execution time which benefits from parallelization. The total latency of a program can be determined via the inverse of the throughput.

Simulation Setup Measuring the performance of any simulation will be highly dependent on the specific cellular properties and complexity. For this comparison, we chose the previously explained cell-sorting example which contains minimal complexity compared to other examples (see cellular-raza.com/showcase). Any computational overhead which is intrinsic to `cellular_raza` and not related to the chosen example would thus be more likely to manifest itself in performance results.
In order to produce reproducible results and simplify this overall process, we provide the cellular_raza-benchmarks crate. It is a command-line utility which can be used to run benchmarks with various configurations. Its arguments are displayed in Listing 1.

CPU	Fixed Clockspeed	Memory Frequency	TDP
AMD Ryzen 3700X [8]	$2200\,\mathrm{MHz}$	$3200\,\mathrm{MT/s}$	$65\,\mathrm{W}$
AMD Ryzen Threadripper 3960X [8]	$2000\,\mathrm{MHz}$	$3200\,\mathrm{MT/s}$	$280\,\mathrm{W}$
Intel Core i7-12700H [9]	$2000\,\mathrm{MHz}$	$4800\,\mathrm{MT/s}$	$45\,\mathrm{W}$

Table 2: List of tested hardware configurations.

```
# cd cellular_raza-benchmarks
# cargo run -- -h
cellular_raza benchmarks

Usage: cell_sorting [OPTIONS] <NAME> [COMMAND]

Commands:
  threads   Thread scaling benchmark
  sim-size  Simulation Size scaling benchmark
  help      Print this message or the help of the given subcommand(s)

Arguments:
  <NAME>  Name of the current runs such as name of the device to be benchmarked

Options:
  -o, --output-directory <OUTPUT_DIRECTORY>
          Output directory of benchmark results [default: benchmark_results]
  -s, --sample-size <SAMPLE_SIZE>
          Number of samples to be generated for each measurement [default: 5]
      --no-save
          Do not save results. This takes priority against the overwrite settings
      --overwrite
          Overwrite existing results
      --no-output
          Disables output
  -h, --help
          Print help
  -V, --version
          Print version
```

Listing 1: Usage of the benchmark CLI tool. We provide two benchmarks, one for increasing the number of agents and another for increasing the number of threads. The subcommands can be further customized and will automatically run the given simulation multiple times for the specified configurations.

Results generated in this way are stored inside the `benchmark_results` folder. In addition, we provide a python script `plotting/cell_sorting.py` to quickly visualize the obtained results.

Hardware This benchmark was run on three distinct hardware configurations. There exists a wide range of variables which could influence our measured runtime results. However, we expect that the biggest effects are due to power-limits and variable frequency of the central processing unit (CPU) (see Figure 2). Both of these effects can be circumvented by choosing an artificially fixed frequency which is low enough such that the total power limit of the CPU is never reached even when multiple cores are under load. While it is well known that other aspects such as cache-size and memory latency can have an impact on absolute performance, they should however not introduce any significant deviations in terms of relative performance scaling.

Results In figure 2, we fit Amdahl's law of equation 5 to our measured datapoints and obtain the parameter p from which the theoretical maximal speedup S can be calculated via

$$\lim_{n\to\infty} T(n) = S = \frac{1}{1-p} \tag{6}$$

The values for the maximum theoretical speedup are $S_{\text{3700X}} = 13.64 \pm 1.73$, $S_{\text{3960X}} = 44.99 \pm 2.80$ and $S_{\text{12700H}} = 34.74 \pm 5.05$. Their uncertainty $\sigma(S)$ can be calculated via the standard gaussian propagation

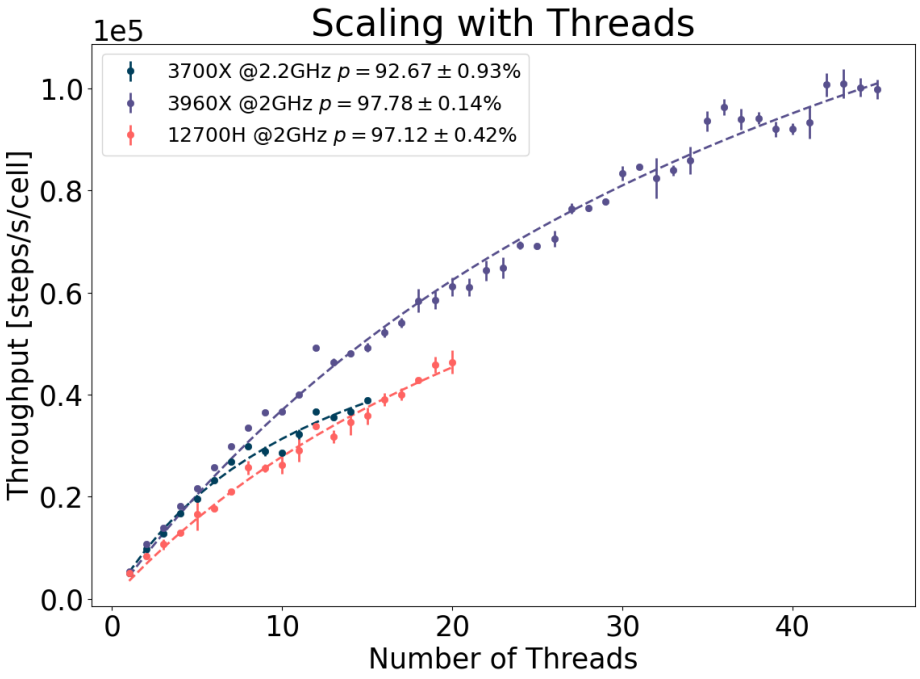

Figure 2: Performance of the throughput $T(n)$ for increasing number of utilized threads n.

$$\sigma(S) = \frac{\sigma(p)}{(1-p)^2} \tag{7}$$

where $\sigma(p)$ is the uncertainty of the parameter p obtained via the fit in figure 2.

Discussion The perfect score of a fully parallelizable system with $p = 1$ is considered almost unobtainable in a real-world scenario due to effects such as the workload of the underlying operating system and physical constraints. Our results showed that the measured value p does also depend on the respective hardware.
In addition to hardware-related effects, we also expect a portion of $1 - p$ of our simulation code to be fundamentally not parallelizable. This fraction can be made up of the initial setup of the simulation which necessarily has to start single-threaded and can only extend to multiple workers once all respective subdomains are generated. Furthermore, ending the simulation not only frees resources which requires computation but also locks the main routine until every individual thread has finished. Even more importantly, all threads are currently using a shared barrier [10] to sync with each other which means that even a single worker can block all others. This limitation could be fixed in the future with improved versions `cellular_raza` and is only an implementation detail and not a fundamental drawback.
However, the total speedup S is still very good for all configurations which can be directly attributed to a good implementation and the core assumption of `cellular_raza` that all interactions are strictly local and subdomains are only interacting along their borders without the need to construct long-ranging synchronization algorithms.

4 Conclusions

We showed along the example of cell-sorting how `cellular_raza` can be used to model cellular biological systems. The modeling process is flexible due to the variety of cellular representations which are supported. We chose to represent our cells with only 4 parameters, which specify the physical representation and interaction

of the cells. The numerical results showed the expected behaviour of cells assembling into connected regions of identical species.
Utilizing the cell-sorting simulation, we benchmarked the performance of the numerical backend. We picked a simulation which is large enough to fully saturate all processors and gradually increased the number of threads utilized. The fractional amount of parallelizable code was determined by fitting the values of runtime to Amdahl's law. It reached values up to $p = 97.78\%$ for a workstation setup which indicates a well-parallelizable implementation.

5 Acknowledgements

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. JP and CF received funding from FET-Open research and innovation actions grant under the European Union's Horizon 2020 (CyGenTiG; grant agreement 801041).

References

[1] J. Pleyer and C. Fleck, "Agent-based models in cellular systems," *Frontiers in Physics*, vol. 10, Jan. 2023. [Online]. Available: http://dx.doi.org/10.3389/fphy.2022.968409

[2] Jonas Pleyer. (2023) bwhpc symposium 2023. [Online]. Available: https://jonaspleyer.github.io/peace-of-posters/showcase/2023-10-23-bwhpc-symposium/

[3] M. S. Steinberg, "Reconstruction of tissues by dissociated cells: Some morphogenetic tissue movements and the sorting out of embryonic cells may have a common explanation." *Science*, vol. 141, no. 3579, p. 401–408, Aug. 1963. [Online]. Available: http://dx.doi.org/10.1126/science.141.3579.401

[4] F. Graner and J. A. Glazier, "Simulation of biological cell sorting using a two-dimensional extended potts model," *Physical Review Letters*, vol. 69, no. 13, p. 2013–2016, Sep. 1992. [Online]. Available: http://dx.doi.org/10.1103/physrevlett.69.2013

[5] T. Schlick, *Molecular Modeling and Simulation*. Springer New York, 2002. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-22464-0

[6] R. W. Pastor, *Techniques and Applications of Langevin Dynamics Simulations*. Springer Netherlands, 1994, p. 85–138. [Online]. Available: http://dx.doi.org/10.1007/978-94-011-1168-3_5

[7] D. P. Rodgers, "Improvements in multiprocessor system design," *ACM SIGARCH Computer Architecture News*, vol. 13, no. 3, p. 225–231, Jun. 1985. [Online]. Available: http://dx.doi.org/10.1145/327070.327215

[8] AMD Ryzen™ 7 3700X. (2024) Amd product specifications. [Online]. Available: https://www.amd.com/en/products/specifications.html

[9] Intel Corporation. (2024) Intel i7 12700h. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/132228/intel-core-i7-12700h-processor-24m-cache-up-to-4-70-ghz.html

[10] Jon Gjengset, "Hurdles." [Online]. Available: https://crates.io/crates/hurdles

Fair sharing of resources between clusters with AUDITOR*

Benjamin Rottler
Institute of Physics
Freiburg University

Michael Böhler
Institute of Physics
Freiburg University

Anton J. Gamel
Institute of Physics
Freiburg University

Dirk Sammel
Institute of Physics
Freiburg University

Markus Schumacher
Institute of Physics
Freiburg University

Raghuvar Vijayakumar
Institute of Physics
Freiburg University

Abstract

For several years, we have been dynamically and opportunistically integrating the computing resources of the HPC cluster NEMO into the HTC cluster ATLAS-BFG using the COBalD/TARDIS software. To increase usage efficiency, we allow the integrated resources to be shared between the various High Energy Physics (HEP) research groups in Freiburg. However, resource sharing also requires accounting. This is done with AUDITOR (AccoUnting DatahandlIng Toolbox for Opportunistic Resources), a flexible and extensible accounting ecosystem that can cover a wide range of use cases and infrastructures. Accounting data is recorded via so-called collectors and stored in a database. So-called plugins can access the data and take measures based on the accounted data. In this work, we present how NEMO resources can be fairly shared among contributing working groups when integrated into ATLAS-BFG using AUDITOR.

1 Introduction

High Energy Physics (HEP) studies fundamental particles and their interactions with the primary goal of discovering new elementary particles and precisely measuring the properties of existing ones. To achieve these goals, particles are accelerated to high energies and are brought to collision at the Large Hadron Collider (LHC) [1], the largest and most powerful particle accelerator in the world, located at CERN. It can collide protons up to 40 million times per second. The ATLAS detector [2], a general-purpose detector, records particle collisions at a rate of up to 3000 collisions per second, generating approximately $10\,\mathrm{PB}$ of data annually. This data is then analyzed by physicists around the world to gain more insight into the fundamental laws of nature.

After data acquisition, the recorded collision events undergo several phases of analysis. First, events are reconstructed to identify particles and to measure their properties. Subsequent steps filter out irrelevant events and aggregate low-level detector measurements into high-level physics observables. This analysis process starts on the Worldwide LHC Computing Grid (WLCG) [3], a global network of computing resources, and ends on local university clusters for final evaluation. To understand the complex processes that occur during particle collisions, the data analysis is supported by extensive simulations performed on the WLCG. In Freiburg, two compute clusters are available for HEP data analysis: ATLAS-BFG and NEMO.

ATLAS-BFG, a High Throughput Computing (HTC) cluster, is equipped with 3400 CPU cores and offers $4\,\mathrm{PB}$ of object storage. It is integrated into the WLCG and provides a specialized environment tailored for ATLAS data analysis. Job scheduling is facilitated by the Slurm [4] scheduler. The cluster is used to execute ATLAS simulation and event reconstruction tasks, as well as analysis jobs originating from local researchers.

NEMO[1], a High Performance Computing (HPC) cluster, is part of the bwHPC initiative serving the broader scientific community in Baden-Württemberg. Designed to support multiple scientific fields, NEMO is equipped with $18\,000$ CPU cores and $800\,\mathrm{TB}$ of parallel storage. Computing resources on NEMO are allocated via a batch system managed by the Moab [5] scheduler, adhering to a single-user node policy, where each node is only accessible to one user at a time. This cluster can be used by local researchers for their user analysis jobs.

*Proceedings of the 9th bwHPC-Symposium 2023. Mannheim Conference Series (MaConf). 2025.

[1] Neuroscience, Elementary Particle Physics, Microsystems Engineering and Material Science

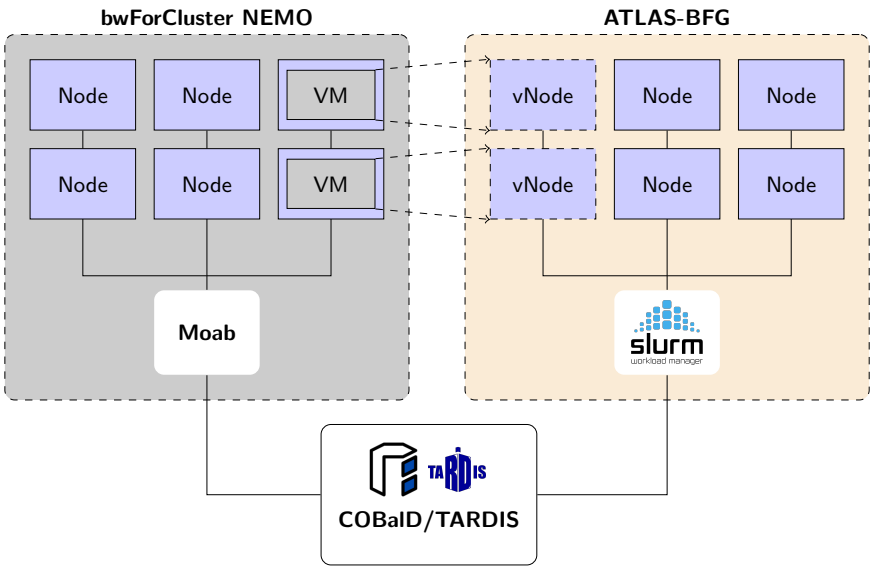

Figure 1: Integrating resources from the NEMO cluster (based on the Moab scheduler) into the ATLAS-BFG cluster (using the Slurm scheduler). NEMO acts as the underlying batch system (UBS) that provides resources for the overlay batch system (OBS), ATLAS-BFG. The resources are provided in the form of virtual machines (VMs) that appear as virtual nodes (vNodes) in the OBS. COBalD/TARDIS monitors the utilization of the vNodes and dynamically increases/decreases the number of VMs.

Since 2019, we have been dynamically and opportunistically integrating the computing resources of NEMO into ATLAS-BFG using the software tools COBalD [6] and TARDIS [7]. The COBalD/TARDIS system allows the orchestration of virtual machines (VMs) in one batch system, allowing them to be utilized as opportunistic resources in another batch system. The first batch system serves as an underlying batch system (UBS), while the second is classified as an overlay batch system (OBS). In our case, as shown in Figure 1, NEMO acts as the UBS and ATLAS-BFG acts as the OBS. When a new VM starts on NEMO, it connects as a virtual node (vNode) to the batch system managed by Slurm on ATLAS-BFG. COBalD/TARDIS monitors the CPU and memory usage of these vNodes and queues new VMs on NEMO when a predefined upper threshold for resource usage is reached. Furthermore, when no new VMs are required, idle VMs can be shut down, freeing up the previously allocated resources.

In Freiburg, four different HEP groups have a share in NEMO and are therefore allowed to use the cluster. Each group is allocated its own group fair share, which is shared among the group members. The fair share is a resource allocation policy that is utilised in computing environments, particularly in job schedulers such as Slurm, PBS and HTCondor, with the objective of ensuring the equitable distribution of computing resources. We operate a COBalD/TARDIS instance for each HEP group, which uses the corresponding group fair share in NEMO. On ATLAS-BFG, there is a Slurm queue for each group, to which users of that group can submit compute jobs. To efficiently utilize the opportunistic resources, we allow jobs from one group to be executed on a VM that was started by another group. This effectively unifies and distributes each group's fair share among the four HEP groups. Figure 2 shows an example of this setup with two groups.

Due to several factors, the amount of resources provided by each group may differ. For instance, users can still directly connect to NEMO and submit jobs there, which reduces the corresponding group fair share on NEMO. This can result in unfair situations, where one group provides fewer resources on NEMO but uses a larger portion of the integrated resources on ATLAS-BFG than the other groups. To restore fairness, the job priority for the groups on ATLAS-BFG should match the amount of resources provided on NEMO. This means that jobs from a group that provides more resources will be scheduled with a higher priority than jobs from the other groups. The AUDITOR [8] ecosystem, which will be introduced in the next section, can be used to achieve this.

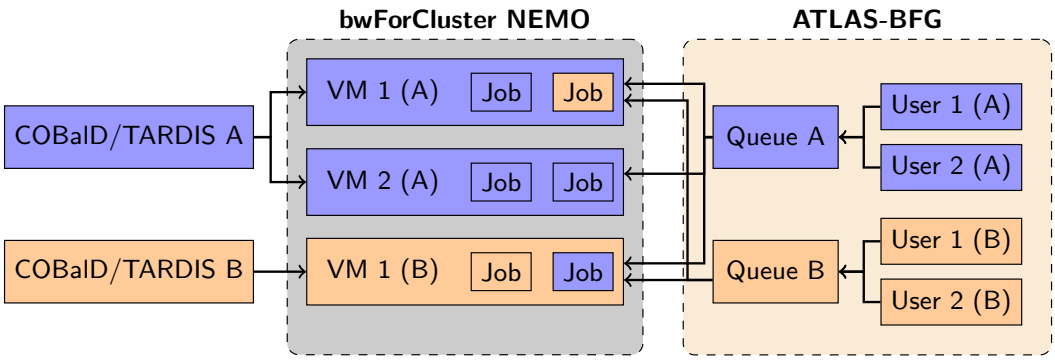

Figure 2: Example of two HEP groups A (blue) and B (orange) sharing resources on NEMO. Both groups have a share on NEMO and use COBalD/TARDIS to start VMs on NEMO. On the ATLAS-BFG cluster, users from both groups submit their jobs to separate queues. Because the VMs are set up to be shared between groups, jobs from users in group A can be assigned to a VM started by group B, and vice versa.

2 The AUDITOR accounting ecosystem

The AUDITOR ecosystem comprises three types of components, as shown in Figure 3. The *core component* stores *records* in a PostgreSQL [9] database. *Collectors* gather data and send it as records to the core component, while *plugins* perform specific actions based on the stored information. Both collectors and plugins interact with the core component through a REST API. To enable the extension of the AUDITOR ecosystem, client libraries that abstract the REST API are available for both the Rust [10] and Python [11] programming languages. The Python client library is a lightweight Python layer built on top of the Rust client library.

2.1 The AUDITOR core component

The core component of AUDITOR is written in the Rust programming language and stores all data in a PostgreSQL database. It is designed to be stateless, making it more resilient against data loss and suitable for high-availability setups with multiple instances distributed behind a load balancer. The core component is available as an RPM package and a Docker container for easy installation.

2.2 Collectors

The collectors are responsible for gathering relevant accounting data from various sources and transmitting it to the core component in the form of records. Four different collectors are available: the TARDIS Collector, the Slurm Collector, the Slurm Epilog Collector, and the HTCondor Collector.

As stated in the introduction, COBalD/TARDIS manages VMs in an UBS. These VMs are referred to as *drones* in the context of COBalD/TARDIS. Each drone can be in one of several states, such as *booting*, *running*, or *stopped*. TARDIS tracks transitions between these states and forwards each transition event to a plugin interface. The TARDIS Collector connects to this plugin interface. Because only state changes are transmitted, a complete record can only be created after the drone has been terminated. To avoid the collector having to track each drone from creation to termination, the TARDIS Collector first sends an incomplete record of a drone to the core component and updates the record later with the stop time of the drone once the drone is terminated. This simplifies the collector but increases the number of interactions with AUDITOR, as each record requires creation and updating. The collector is distributed with the TARDIS [7] software.

The Slurm Collector and Slurm Epilog Collector can be used to collect accounting information from a Slurm batch system. The Slurm Epilog Collector utilizes the epilog functionality of the Slurm scheduler to execute scripts at the end of each batch job. It collects job information using the Slurm command line interface (CLI) and sends it to the core component. However, its usage may affect job runtime due to processing delays. Additionally, the information available from Slurm at the time of the epilog may be limited and may not cover

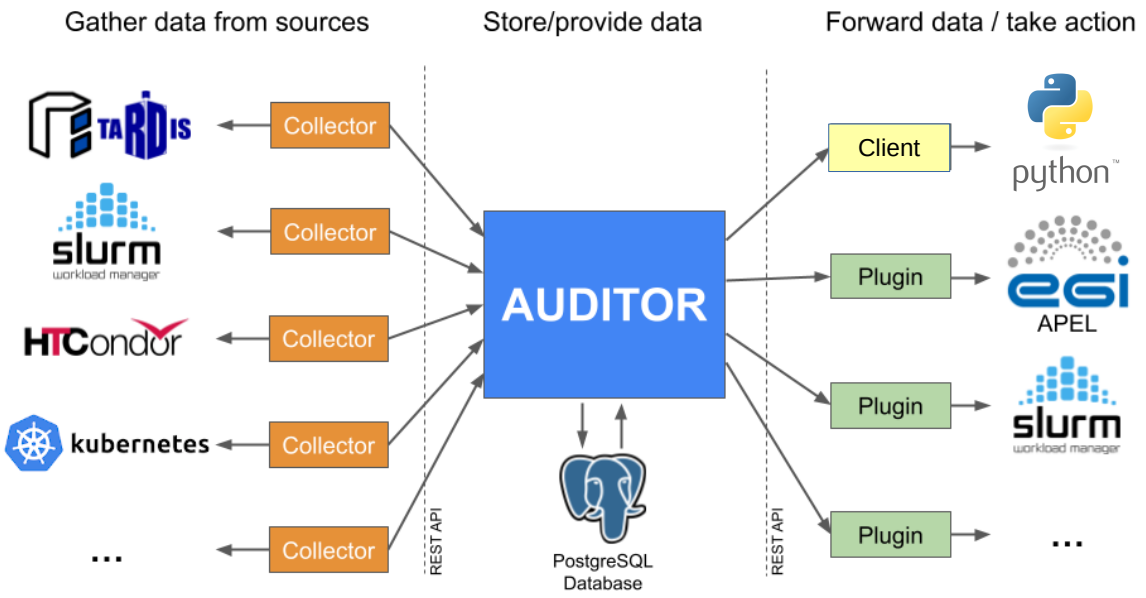

Figure 3: Overview of the AUDITOR ecosystem. AUDITOR accepts records from collectors and stores them in a PostgreSQL database. It grants access to these records to the plugins, which perform actions based on the stored information.

all use cases. To address these issues, the Slurm Collector operates independently and periodically queries the Slurm accounting database using the Slurm CLI. The records are then transmitted to the core component at regular, configurable time intervals. Both collectors are compiled into portable, statically linked binary files with no system dependencies. They are highly configurable and require only the installation of the Slurm client software.

The HTCondor Collector provides functionality similar to that of the Slurm Collector but for the HTCondor [12] scheduler. It queries accountable data from HTCondor using the HTCondor CLI. The key difference between the HTCondor Collector and the Slurm Collector is that the former is written in Python and therefore requires a compatible Python distribution on the host where it is running.

The Kubernetes collector gets its info from two places: the Kubernetes API and a Prometheus instance. This is a necessary process since Kubernetes does not provide its resource metrics, such as CPU time, via its API. Consequently, the collector must possess the capability to access both the API and Prometheus.

2.3 Plugins

Plugins can perform various tasks based on the information stored as records in the AUDITOR database. When two clusters are connected via COBalD/TARDIS, the Priority Plugin can adjust the group priority in the OBS based on the resources provided in the UBS. The APEL Accounting Plugin forwards the accounting information gathered in AUDITOR to the EGI APEL accounting platform [13]. Another potential use case is a utilization reporting tool that analyzes requested versus the consumed resources per user job and provides a weekly overview of potential savings and corresponding CO_2 footprint to raise user awareness. Please note that the last plugin is not yet available at the time of writing. The following section presents the priority plugin in detail.

3 Fair sharing of resources between clusters

As described in the introduction, resources from the NEMO cluster are opportunistically integrated into the ATLAS-BFG cluster using COBalD/TARDIS and shared among the four HEP groups. To ensure fair allocation

of resources among the these groups, the job priority for the groups on ATLAS-BFG should correspond to the amount of resources provided on NEMO. This can be achieved using a combination of the TARDIS Collector, AUDITOR, and the priority plugin. The TARDIS Collector gathers accounting data from COBalD/TARDIS and stores it in the AUDITOR database. The Priority Plugin calculates the priority for each group based on the resources allocated on NEMO by COBalD/TARDIS and adjusts the priority in the Slurm scheduler on ATLAS-BFG accordingly.

To calculate the priority, the plugin first computes the resources provided by each group $i \in [A, \ldots, D]$ in vCore hours during the previous 14 days, denoted as $c_i(t)$:

$$c_i = \int_{t_{\text{now}}-14\,\text{d}}^{t_{\text{now}}} N_i(t)\,\text{d}t\,. \tag{1}$$

Here, $N_i(t)$ represents the number of vCores that group i provided at time t. The priority p_i for group i is then defined as the ratio of resources provided by a single group c_i to the total amount of vCore hours by all HEP groups. The fraction is then scaled to the range between the minimum and maximum priority values, p_{min} and p_{max}:

$$p_i = \frac{c_i}{\sum_j c_j} \cdot (p_{\text{max}} - p_{\text{min}}) + p_{\text{min}}\,. \tag{2}$$

The group priorities p_i are adjusted hourly on the OBS, which in this case is the Slurm scheduler. The minimum and maximum priority values for Slurm have been chosen as $p_{\text{min}} = 1$ and $p_{\text{max}} = 2^{16} - 1 = 65535$, respectively.

Figures 4 and 5 show the integrated vCore hours calculated according to Eq. 1 and the updated group priorities determined using Eq. 2, respectively. It is evident that Group A provides fewer resources than the other groups, which is reflected in their lower priority. As a result, members of Group A experience significantly longer waiting times for job submissions.

The Priority Plugin makes the data from Figures 4 and 5 available through a Prometheus [14] exporter. When connected to a Grafana [15] instance, the data can be visualized in real time.

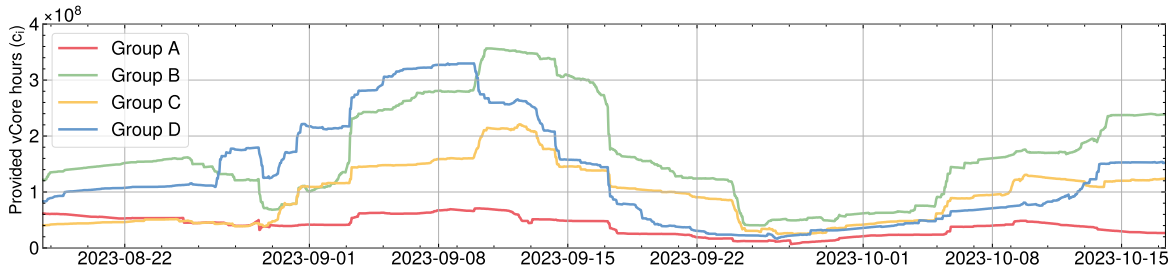

Figure 4: Integral over the vCore hours $c_i(t)$ of the previous 14 days per group i according to Eq. 1 recorded on the NEMO cluster in the period August - Oktober 2023.

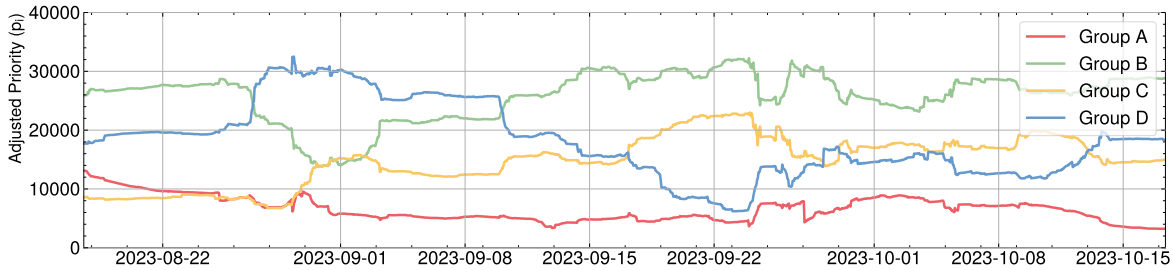

Figure 5: Calculated priority $p_i(t)$ per group i using Eq. 2 based on resources used on NEMO.

4 Conclusions

AUDITOR is a modular accounting ecosystem designed with flexibility in mind. Its REST interface and client libraries in Rust and Python enable a broad community of users to contribute to the expansion of the ecosystem and to implement their own collectors and plugins quickly and easily. The existing collectors and plugins can be combined in various ways to implement different use cases.

Resources of the NEMO cluster have been integrated into the ATLAS-BFG cluster using COBaID/TARDIS in Freiburg for several years. To ensure fair resource sharing among the four HEP groups in Freiburg, AUDITOR, together with the TARDIS Collector and the Priority Plugin, is utilized to adjust the job priority on ATLAS-BFG based on the resources provided on NEMO.

5 Acknowledgements

This work was supported by the Federal Ministry of Education and Research (BMBF) within the project 05H21VFRC2 "Entwicklung, Integration und Optimierung von digitalen Infrastrukturen für ErUM" in the context of the collaborative research centre "Föderierte Digitale Infrastrukturen für die Erforschung von Universum und Materie (FIDIUM)".

The HPC-cluster NEMO in Freiburg is supported by the Ministry of Science, Research and the Arts Baden-Württemberg through the bwHPC grant and by the German Research Foundation (DFG) through grant no INST 39/963-1 FUGG.

References

[1] L. Evans and P. Bryant, *LHC Machine*, JINST 3 (2008) S08001

[2] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST 3 (2008) S08003

[3] I. Bird, K. Bos, N. Brook, et al., *LHC computing Grid: Technical design report*, CERN-LHCC-2005-024

[4] A. Yoo, M. Jette and M. Grondona, *SLURM: Simple Linux Utility for Resource Management*, Job Scheduling Strategies For Parallel Processing, 44-60 (2003)

[5] Adaptive Computing, Inc., *Introduction to Cloud for HPC.*, White Paper (2011)

[6] M. Fischer, E. Kuehn, M. Giffels, et al., *Lightweight dynamic integration of opportunistic resources*, EPJ Web of Conferences **245**, 07040 (2020)

[7] M. Fischer, M. Giffels, A. Haas, et al., *Effective Dynamic Integration and Utilization of Heterogenous Compute Resources*, EPJ Web of Conferences **245**, 07038 (2020)

[8] Boehler, M., von Cube, R., Fischer, M., et al., AUDITOR: Accounting data handling toolbox for opportunistic resources. *The European Physical Journal C*. **85** (2025,3)

[9] PostgreSQL Global Development Group, *PostgreSQL*, https://www.postgresql.org, accessed 8th February 2024

[10] N. Matsakis and F. Klock, *The Rust language*, ACM SIGAda Ada Letters **34**, 103-104 (2014)

[11] G. Van Rossum and F. Drake, *Python 3 Reference Manual*, CreateSpace (2009)

[12] D. Thain, T. Tannenbaum and M. Livny, *Distributed computing in practice: the Condor experience.*, Concurrency and Computation: Practice and Experience **17**, 323-356 (2005)

[13] M. Jiang, C. Del Cano Novales, G. Mathieu, J. Casson, et al., *An APEL Tool Based CPU Usage Accounting Infrastructure for Large Scale Computing Grids*, Data Driven e-Science, 175-186, Springer (2011)

[14] B. Rabenstein and J. Volz, *Prometheus: A Next-Generation Monitoring System (Talk)*, USENIX Association, 2015

[15] Grafana Labs, *Grafana*, https://grafana.com, accessed 9th February 2024

Numerical Simulation of Plastic Pyrolysis*

Feichi Zhang, Salar Tavakkol, Dieter Stapf
Institute for Technical Chemistry (ITC)
Karlsruhe Institute of Technology (KIT), Kaiserstr. 12, 76131, Karlsruhe, Germany

Thorsten Zirwes
Institute for Reactive Flows (IRF)
University of Stuttgart, Pfaffenwaldring 31, 70569 Stuttgart, Germany

Abstract

Pyrolysis of plastic waste for base chemical production is a pivotal technology for closing the carbon cycle and advancing sustainable energy transition. This work presents an overview of research activities conducted at the Institute for Technical Chemistry (ITC) at KIT, focusing on the simulation of plastic pyrolysis to develop efficient and cost-effective recycling technologies. A key objective of this research is to elucidate the physicochemical behavior during pyrolysis through high-fidelity numerical simulations, leveraging high-performance computing resources. These simulations provide fundamental insights essential for process design and optimization. Several ongoing studies are discussed, systematically spanning from fundamental single-particle analyses to fully resolved laboratory-scale fluidized bed reactors. These investigations evaluate the impact of key operating conditions and reactor design parameters on pyrolysis performance, demonstrating how numerical simulations can streamline and optimize the design process.

1 Introduction

Currently, only about 9% of global plastic waste—approximately 350 million tons annually—is recycled. The remaining 90% is either landfilled, incinerated, or released into the environment, leading to severe pollution problems [1, 2]. The low recycling rate stems from the limitations of conventional mechanical recycling, which can only process highly pure plastic waste. In contrast, chemical recycling via pyrolysis reverses the polymerization process used in plastic production, enabling the recycling of mixed and contaminated plastics. During pyrolysis, plastic polymers are thermally broken down into petrochemical feedstocks, which can then be used to manufacture new plastics, creating a closed-loop recycling system. As shown in Fig.1, plastic waste is converted into pyrolysis oil, which can be further refined into plastic pellets and new plastic products.

The pyrolysis process is governed by numerous parameters, including reactor temperature, flow and heating conditions, residence time, particle size, and catalyst use. Additionally, the reactor type, geometry, and dimensions significantly influence pyrolysis performance. Given this complexity, experimental investigations covering the full range of operating and design parameters are highly challenging and time-intensive. Consequently, prior research on plastic pyrolysis has primarily relied on small-scale experiments using milligram to gram quantities of plastic. These studies have provided critical data on reaction kinetics, pyrolysis product distribution, and material thermodynamics, demonstrating the technology's potential for plastic recycling at the laboratory scale. However, the interplay of multi-scale and multi-physical phenomena—coupled with the vast parameter space—makes precise control over product yield difficult. Furthermore, findings from lab-scale experiments often fail to translate directly to industrial-scale pyrolysis plants due to substantial differences in reaction and flow conditions. Thus, further optimization is essential to enhance scalability, energy efficiency, product yield, and economic viability for real-world applications.

To address this challenge, we employ numerical simulations to investigate multiphase reactive flows, enabling a fundamental understanding of plastic pyrolysis at unprecedented resolution. Unlike experimental

*Proceedings of the 9th bwHPC-Symposium 2023. Mannheim Conference Series (MaConf). 2025.

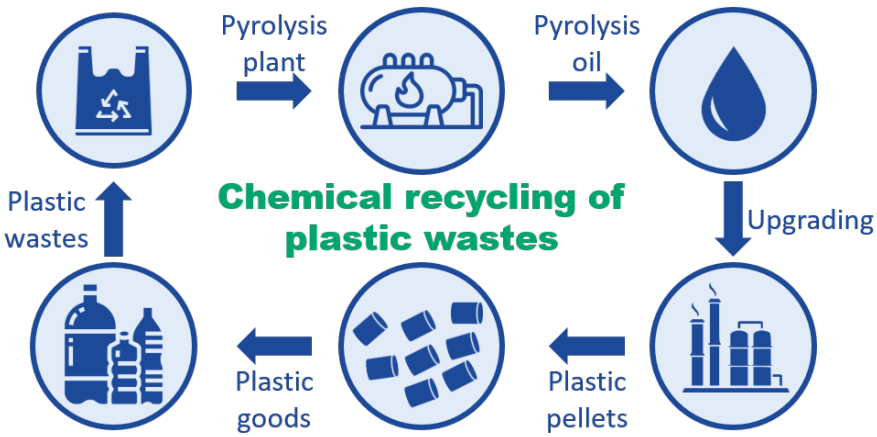

Figure 1: Chemical recycling of plastics via pyrolysis (icons taken from "https://www.drugplastics.com/an-introduction-to-advanced-plastic-recycling/ and modified").

approaches, these simulations provide temporally and spatially resolved flow fields, offering critical insights for pyrolysis reactor design. This work presents two case studies of high-performance computing (HPC) simulations for plastic pyrolysis, accompanied by key research findings that advance process optimization.

2 Simulation methods

Plastic pyrolysis involves heating shredded plastic particles to high temperatures, causing polymer degradation into short-chain and cyclic hydrocarbons (product vapors). This process constitutes a multiphase reactive flow system. For large-scale modeling of such flows, two fundamental approaches exist: The Euler-Euler approach treats both gas and solid phases as interpenetrating continua, with phase interactions (mass, momentum, and energy exchange) modeled using the kinetic theory of granular flow (KTGF) [3, 4]. While computationally efficient—solving two sets of Eulerian equations—it lacks resolution of individual particle dynamics and ignores particle size effects, which may significantly impact pyrolysis accuracy.

The Euler-Lagrange method [5] adopts a fundamentally different approach: the gas phase remains continuous, while the solid phase is treated as discrete Lagrangian parcels (LPs). Each LP represents a group of spherical particles sharing identical diameter and velocity characteristics. These parcels are tracked throughout the computational domain, with bidirectional coupling between phases achieved through source terms accounting for interphase mass, momentum, and heat transfer. Key advantages over the Euler-Euler approach include: Explicit resolution of individual particle trajectories and velocities; Incorporation of particle size distribution effects; Higher physical fidelity in multiphase interactions. However, this enhanced accuracy comes at substantially greater computational cost, making the method less practical for certain large-scale applications where Euler-Euler's approximations may suffice.

3 Results and discussions

3.1 Case I: Particle-resolved simulation

The Euler-Lagrange method assumes Lagrangian particles to be perfectly spherical and homogeneous. However, real shredded plastic particles typically exhibit irregular shapes, and the homogeneity assumption only holds for small or thermally-thin particles. Additionally, the method requires particles to be significantly smaller than the computational grid cells, preventing resolution of thermal and flow boundary layers around particles. To evaluate these limitations, we employed the Euler-Euler approach to simulate pyrolysis of a single high-density polyethylene (HDPE) particle in hot nitrogen flow. This alternative methodology enables direct resolution of both the particle-internal and its surrounding boundary layers.

As depicted in Figure 2, the simulation configuration [6] (left) and meshed computational domain (right) feature a vertical cylindrical reactor (base dimensions: diameter x length = 5.3 mm × 15 mm) with 0.35 mm quasi-equidistant grid resolution. Simulations encompassed particle diameters (2-6 mm) and geometries (sphere/cylinder/shell) across temperature regimes, with domain diameters proportionally scaled (5.3, 10.6, 21.2 mm) to maintain adequate clearance for 3, 4, 6 mm particles respectively.

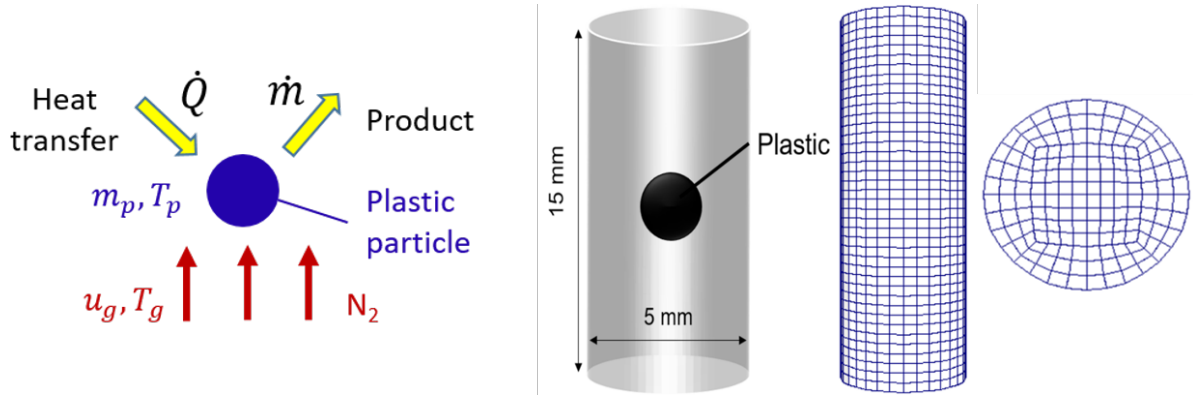

Figure 2: Setup and computational grid used for the particle-resolved simulation of plastic pyrolysis.

Figure 3 (left) displays the pyrolysis progress X (where $X = 0$ and $X = 1$ represent unpyrolyzed and fully pyrolyzed states, respectively) for plastic particles of varying diameters d_P. The results demonstrate a significant increase in pyrolysis time with d_P, attributable to greater particle mass and slower heating rates. For comparison, dotted lines show Euler-Lagrange simulations assuming ideal spherical, homogeneous particles under identical conditions. As particle size increases, particle-internal heat transfer effects become more pronounced, amplifying discrepancies between the Euler-Euler (solid lines) and Euler-Lagrange (dotted lines) methods. This is particularly evident for the 6 mm particle case in Fig. 3: the Euler-Lagrange simulation shows delayed reaction onset ($X > 0$) due to the time required to achieve uniform particle temperature – a limitation not present in the particle-resolved Euler-Euler approach, where pyrolysis initiates immediately at the hotter particle surface. The divergence between methods grows with d_P, as the Euler-Euler method better captures the thermal gradient-driven surface-initiated pyrolysis that dominates real particle behavior.

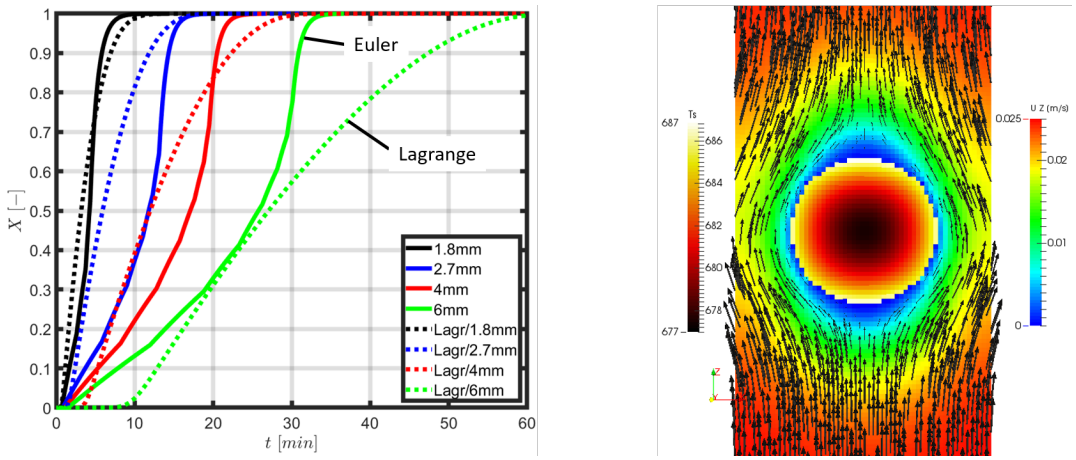

Figure 3: Time evolution of calculated conversion progress from Euler-Euler/-Lagrange approaches (left) and contours of particle temperature along with velocity vectors (right).

Figure 3 (right) presents instantaneous contours of particle temperature (T_P) and gas velocity on a meridional plane through the symmetry axis for the 6 mm particle case, with flow directions indicated by arrows. The visualization clearly reveals significant temperature gradients within the particle that are unresolvable by

the Euler-Lagrange approach, as well as boundary layer formation, where gas velocity approaches zero (no-slip condition) and temperature converges to T_P at the gas-solid interface. These findings demonstrate that the Euler-Lagrange method becomes unreliable for plastic particles exceeding 2 mm in diameter. Additional studies (not shown) further indicate substantial influences of particle shape and flow orientation on pyrolysis dynamics [7].

The simulations employed a computational grid of approximately 400,000 cells and simulated physical times up to 40 minutes. For the most computationally demanding case (6 mm particle requiring full conversion from $X = 0$ to $X = 1$), the calculations required 3 days of runtime utilizing 320 CPU cores on the bwUniCluster system at SCC/KIT.

3.2 Case II: Simulation of fluidized beds

Fluidized bed reactors offer significant advantages for plastic pyrolysis, including enhanced heat transfer, uniform temperature distribution, and continuous operation capability, making them particularly suitable for industrial-scale applications. In this study, we employ the Euler-Lagrange method to simulate a laboratory-scale fluidized bed reactor (developed at ITC/KIT) under isothermal conditions, using inert sand particles as the bed material and air as the fluidizing gas. The primary objective is to investigate the hydrodynamic behavior of the fluidized bed, focusing on particle fluidization, circulation and bubble formation dynamics, which critically influence reactor performance, such as the heat transfer.

The computational domain consists of a cylindrical reactor (60 cm length $\times$ 5 cm diameter) discretized with approximately 300,000 computational cells, yielding an average resolution of 1 mm. We examine various operating conditions by varying with bed inventory $m_S = 195 - 586$ g, superficial gas velocity $u_G = 14 - 30$ cm/s. The number of Lagrangian parcels (LPs) scales with bed inventory, reaching up to 6 million for the maximum m_S case (586 g).

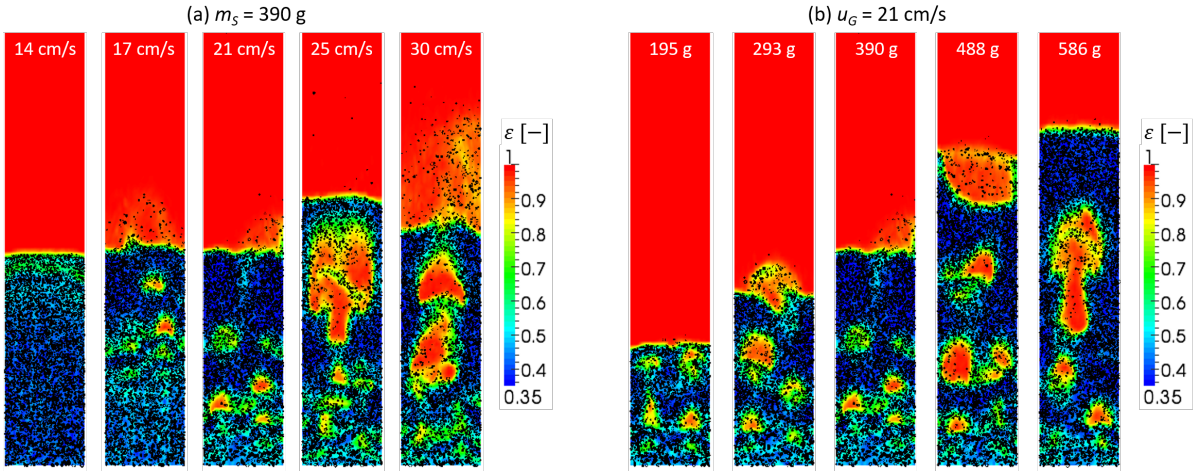

Figure 4: Snapshots of void fraction on a cutting plane passing through the centerline axis at increasing superficial gas flow velocity (a) and at increasing bed inventory (b).

Figure 4 depicts instantaneous contours of the void fraction ε (volume fraction of the gas phase) on a meridian cutting plane passing through the centerline axis for varied u_G at $m_S = 390$ g (left) and varied m_S at $u_G = 21$ cm/s (right). The solid points indicate the sand particles and the red zones surrounded by the sand particles denote the gas bubbles. The fluidization behavior observed in the experiment has been reproduced well, where an increased bubble size with u_G and an increased bed height with m_S can be detected. Although not shown here, the calculated pressure drop and bed height have shown a quantitatively good agreement with measured data, which validates the applicability of the simulations [8].

Figure 5(a) shows iso-contour of $\varepsilon = 0.66$ calculated from Euler-Lagrange simulations of fluidized beds with different reactor diameters d_R, which illustrate 3D structures of the bubbles. For the case with $d_R = 3$ cm, the hydrodynamics of the fluidized bed is dominated by large-scale bubbles with a size similar to the reactor

diameter. The formation of bubbles is significantly enhanced while up-scaling the fluidized bed, which is attributed to the increased volume for solid circulation. The same behavior with an enhanced bubble formation at larger d_R can also be detected in Fig.5(b) by the 2D contours of ε on a meridian cutting plane across the centerline axis. The results indicate an influence of up-scaling on the hydrodynamics of fluidized bed, which has a significant impact on the heat- and mass transfer behavior. This should be accounted for while designing industrial-scale pyrolysis plants using the fluidized bed technology. The reader is referred to [8] for a more detailed description of the results.

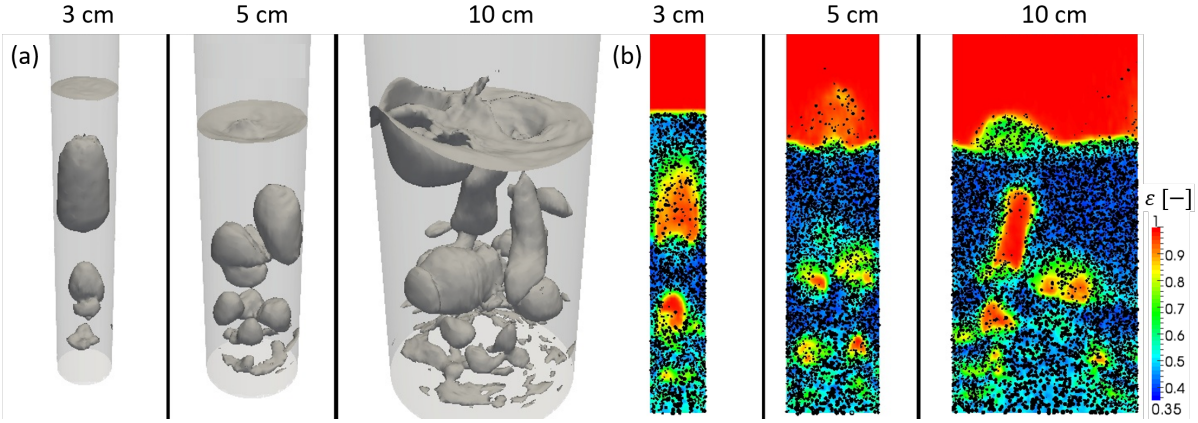

Figure 5: 3D bubble structures (a) and contours of void fraction ε on a cutting plane passing through the centerline axis (b) calculated with reactor diameters of $d_R = 3$, 5 and 10 cm.

3.3 Parallel performance

The open-source CFD software OpenFOAM has be used in this work for the particle-resolved simulations in Sec.3.1 and the simulations of the fluidized beds in Sec.3.2. It uses domain decomposition to allocate the total computational cells onto a number of sub-domains or processes, along with the message passing interface (MPI) for inter-process communication. Figure 6 on the left shows the parallel scalability and on the right the speed-up efficiency for the multiphase flow simulations (Euler-Lagrange) of the fluidized bed with $d_R = 10$ cm by using $32{\times}10^6$ and $16{\times}10^6$ LPs on the Horeka cluster at SCC/KIT, where each node has 76 CPU cores. The resolution used for the Eulerian/gas phase is 1 mm in both cases, leading to 0.56 million grid cells. The wall clock time per time step is 3.0 s for the case with $32{\times}10^6$ LPs and is 1.9 s for the case with $16{\times}10^6$ LPs, indicating that the computational effort of the Lagragian phase dominates over the computing time for the gaseous phase. Therefore, the parallel performance is better while using a larger number of LPs. The scale-up performance is good in case of using more than 100.000 LPs per core to achieve a speed-up efficiency larger than 0.8. A further increase of the number of CPU cores cannot efficiently accelerate the computing process. This is attributed to that fact that the Lagrangian parcels are located only in the lower half of the simulation domain, leading to an imbalance of computing load for each sub-domain or CPU core.

4 Conclusions

Euler-Euler and Euler-Lagrange simulations have been conducted to study the pyrolysis process of a single plastic particle with non-ideal shape and the hydrodynamic behavior of a laboratory-scale fluidized bed designed for plastic pyrolysis. The particle morphology (size, shape) has been found to play a vital role in the pyrolysis process, which is attributable to the modified condition of heat transfer as well as its impact on the pyrolysis reaction. In addition, the calculated particle circulation and bubble formation in the considered fluidized bed have shown a good agreement with corresponding experiments. Moreover, up-scaling of the fluidized beds has led to a transition from single- to multiple-column bubbling and internal particle circulation, indicating the strong influence of scale-up on the mixing and heat transfer process. In terms of an efficient utilization of HPC

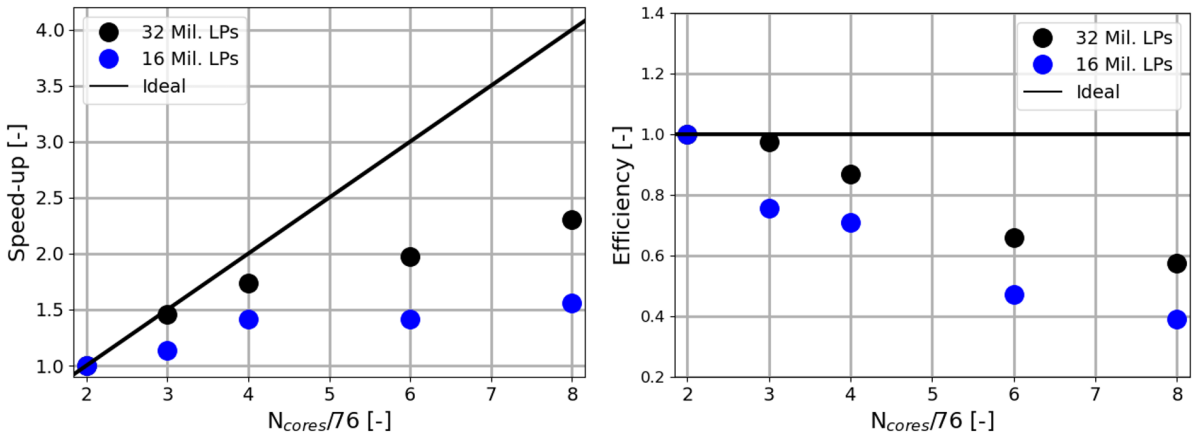

Figure 6: Speed-up performance (left) and efficiency (right) while running multiphase flow simulations (Euler-Lagrange) of fluidized beds by using 32 mil. 16 mil. Lagrange parcels (LPs) on the Horeka cluster at SCC.

resources, the usage of more than 100,000 Lagrange parcels are found to be beneficial for calculating the gas-solid flow in the current fluidized bed setup with the Euler-Lagrange method. The results reveal opportunities and challenges of numerical modeling for advancing the design process of plastic pyrolysis technology.

Acknowledgements

The authors gratefully acknowledge the financial support by the Helmholtz Association of German Research Centers (HGF), within the research program Materials and Technologies for the Energy Transition (MTET), topic Resource and Energy Efficiency. This work utilizes computing resources from the supercomputer HoreKa at the Scientific Centre for Computing (SCC) at KIT, Germany.

References

[1] OECD, "Plastic pollution is growing relentlessly as waste management and recycling fall short," 2022.

[2] R. Geyer, J. R. Jambeck, and K. L. Law, "Production, use, and fate of all plastics ever made," *Sci. Adv.*, vol. 3, p. e1700782, 2017.

[3] S. Elghobashi, "On predicting particle-laden turbulent flows," *Appl. Sci. Res.*, vol. 52, pp. 309–329, 1994.

[4] D. Gidaspow, "Multiphase flow and auidization: Continuum and kinetic theory description," *Academic Press*, p. 9780122824708, 1994.

[5] F. Alobaid, N. Almohammed, M. M. Farid, J. May, P. Rößger, A. Richter, and B. Epple, "Progress in CFD simulations of fluidized beds for chemical and energy process engineering," *Prog. Energy Combust. Sci.*, vol. 91, p. 109930, 2022.

[6] F. Zhang, J. Cao, T. Zirwes, N. Netsch, S. Tavakkol, R. Zhang, H. Bockhorn, and D. Stapf, "Numerical simulation of thermal decomposition of polyethylene with a single-particle model," in *International Conference on Computational Heat and Mass Transfer*, pp. 180–191, Springer, 2023.

[7] F. Zhang, S. Tavakkol, F. Galeazzo, and D. Stapf, "Particle-resolved simulation of pyrolysis process of a single plastic particle," *Heat Mass Transf.*, vol. 61, no. 12, 2025.

[8] F. Zhang, S. Tavakkol, S. Dercho, J. Zhou, T. Zirwes, M. Zeller, J. Vogt, R. Zhang, H. Bockhorn, and D. Stapf, "Assessment of dynamic characteristics of fluidized beds via numerical simulations," *Phys. Fluids*, vol. 36, no. 2, 2024.