The Heterogeneous Exascale Particle-In-Cell (HEXAPIC)

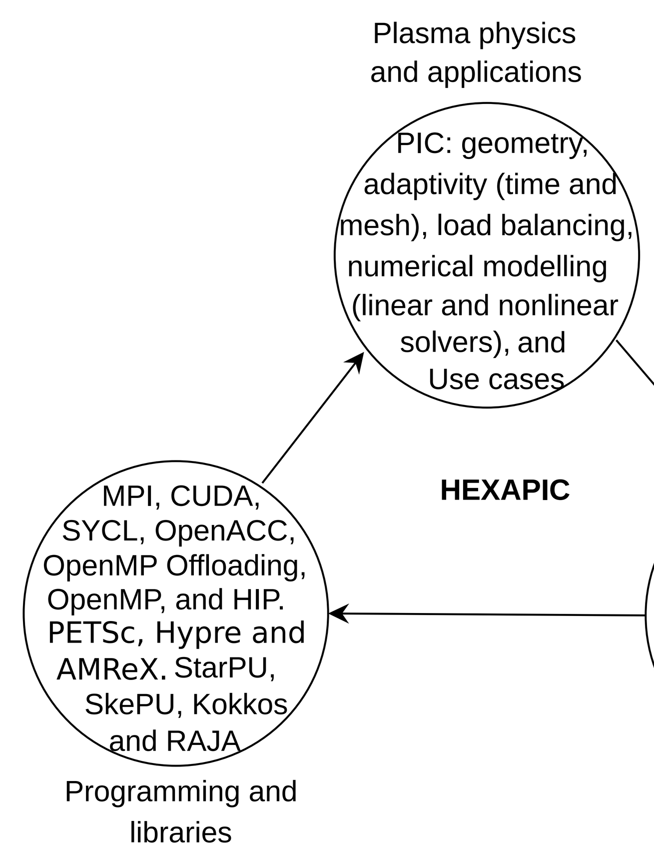
Ezhilmathi Krishnasamy[1,2] Shan-Chang Lin[1] Ayushi Agrawal[1] Pascal Bouvry[1] Stefan Costea[2] Jernej Kovačič[2] Leon Kos[2]
[1]University of Luxembourg, Luxembourg
[2]University of Ljubljana, Slovenia
ezhilmathi.krishnasamy@uni.lu

Abstract

This research investigates the potential enhancements of particle-in-cell (PIC) codes through modern heterogeneous computer architectures, focusing on two key hypotheses. First, we propose that implementing advanced algorithms designed for complex physical mechanisms will significantly improve PIC code algorithms by optimizing workload distribution, efficiently utilizing computing accelerators, and enhancing communication and memory management, thereby achieving substantial performance gains. Second, we emphasize the critical need for fully-kinetic PIC codes to accurately model plasma technological processes. Our aim is to optimize our PIC code using state-of-the-art techniques that facilitate the integration of realistic geometries and comprehensive plasma dynamics, ultimately enhancing predictive and optimization capabilities and achieving unprecedented precision in plasma simulations.

Objectives

- Develop a novel Particle-In-Cell (PIC) code architecture that fully exploits the heterogeneous capabilities of modern high-performance computing systems, ensuring scalability to exascale computing levels.
- Establish a formalized PIC code engineered for exascale performance, facilitating the creation of realistic plasma simulations that accurately represent complex geometries and plasma processes.



Schematic overview of the workflow of Components in the HEXAPIC Project.

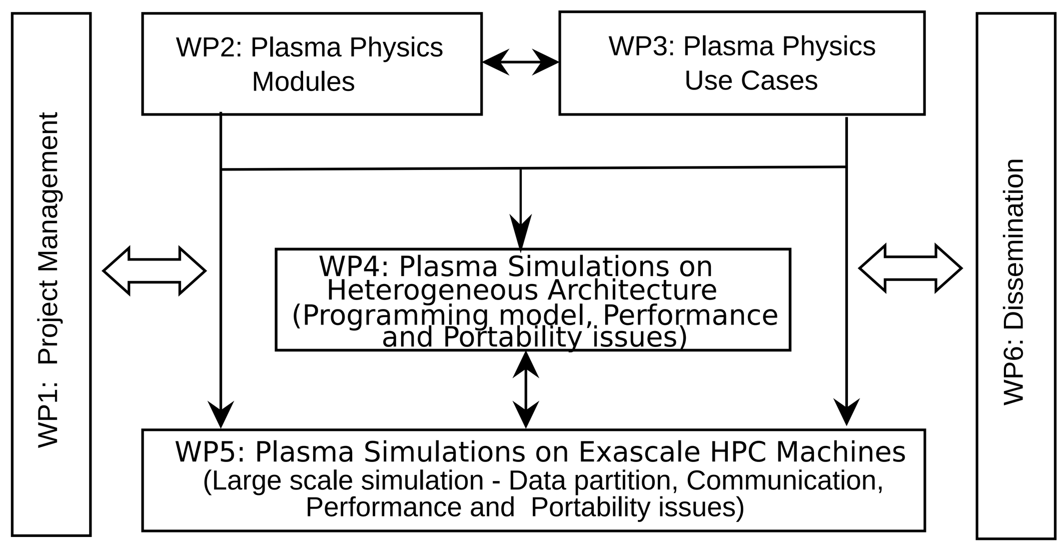
Contributions

- Introduction of a groundbreaking PIC code, optimized for exascale performance, built on an adaptable heterogeneous architecture that includes advanced memory management techniques and effective workload distribution.
- Presentation of practical use cases demonstrating the novel PIC code's capability to produce realistic plasma simulations, accounting for intricate geometrical configurations and various collisional and transport processes, ultimately leading to new insights into physical phenomena.

Project Overview

Runtime: 2024 September - 2027 August
Funding source: Programme / Grant No. C23/IS/18105668/HEXAPIC
Participating partners:

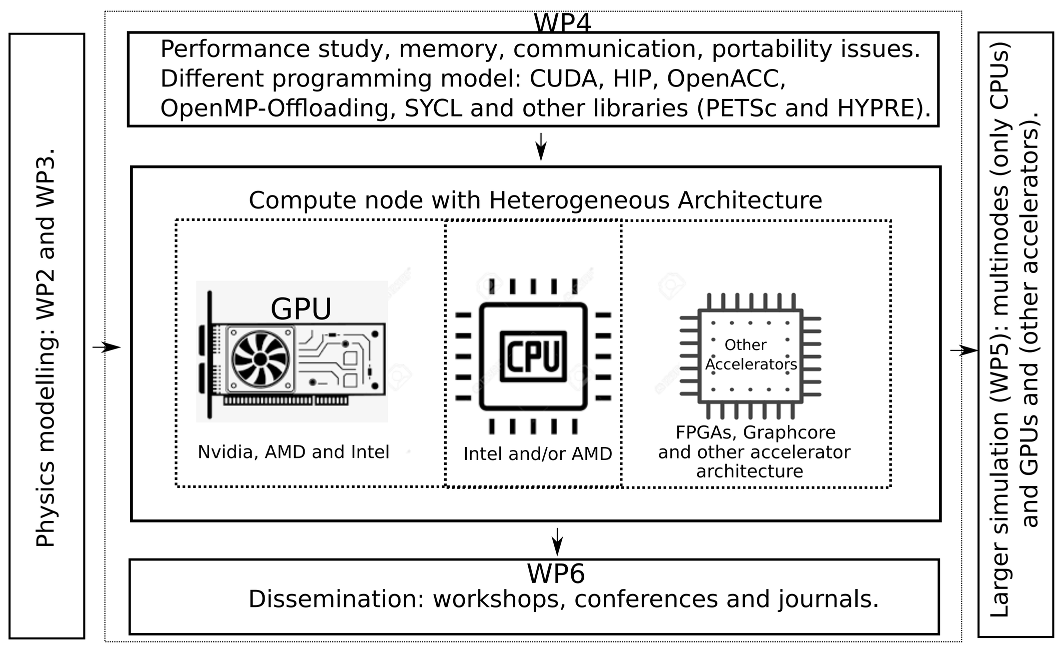
- University of Luxembourg (Lead)
- University of Ljubljana (Partner)
- University of Oslo (In-kind partner)
- Simula Research Laboratory (In-kind partner)
- Universidade de Santiago de Compostela (In-kind partner)
- University of Groningen (In-kind partner)
- FNR / ARIS (funding agencies)



Complete workplan of the HEXAPIC project, outlining key phases, tasks, and milestones.

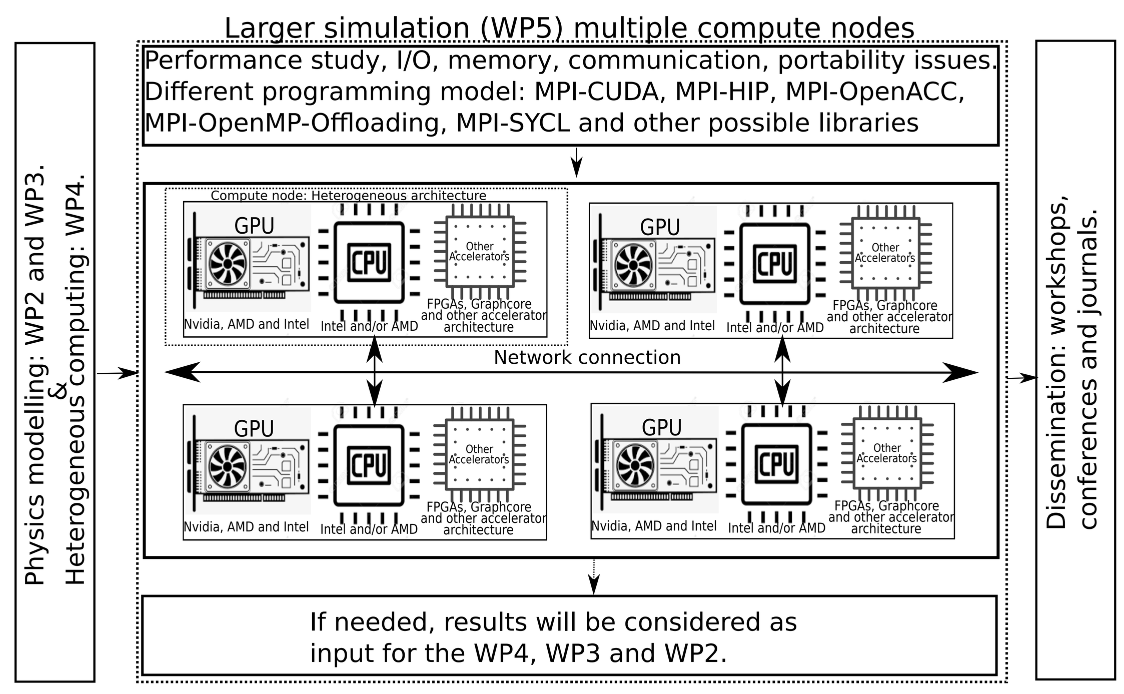
Approach & Methodology

Stable Sequential Version: A stable version of the software captures all physical phenomena along with material properties, boundary, and initial conditions, while also introducing an MPI version that includes support for scientific libraries such as PETSc and HYPRE. We also utilize profiling tools to optimize the code for both CPU and GPU.



Scientific workflow outlining the proposed optimization strategy for a single compute node.

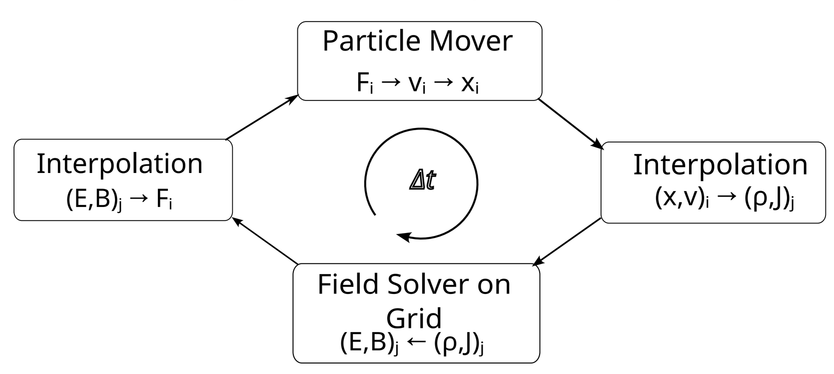
On a multiple compute node: On a multiple compute node setup, it is important to check the efficiency of multigrid solvers with HYPRE for GPU applications, if applicable. Additionally, implementing custom solvers should be considered using a suitable programming model or library that targets heterogeneous large HPC machines.



Scientific workflow outlining the proposed optimization strategy for multiple compute nodes in HEXAPIC.

Current Status

The 2D MPI code of the PIC model has been released with HYPRE support for both single-node and multi-node CPU environments, available at **github.com/LeCAD-PEG/hexapic**. The GPU status has also been updated to include the HYPRE solver, focusing on both single-node and multi-node GPU setups. Additionally, various parallel programming models have been studied for GPUs across both single and multiple compute nodes.



Explicit PIC algorithm: i and j indicate particles and grids; v: velocity, x: position, F: force, ρ: charge density, J: current density, E: electric field, B: magnetic flux, Δt: time step.

Preliminary Results (excerpt)

Single node GPU:

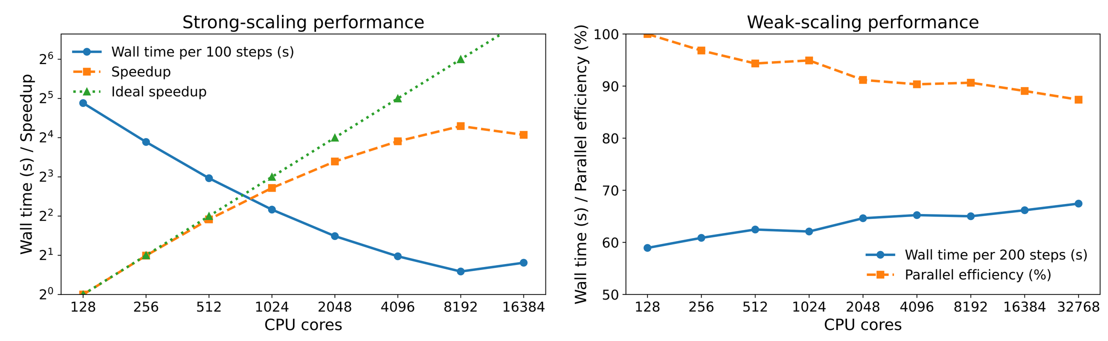
- Single GPU optimization on various programming models concluded [1, 2, 3, 4].

Multi node GPU:

- Various programming models tested with MPI+X, where X refers to various programming models such as CUDA, HIP, SYCL, OpenACC, and OpenMP Offloading for multiple GPUs [5].

HEXAPIC code on CPU and GPU

- We tested a 2D PIC code on multi-node CPUs using PETSc and HYPRE (PFMG) solvers.



Scalings of HEXAPIC on VEGA HPC, CPU multi-node.

- HYPRE has been introduced to accelerate support for NVIDIA and AMD GPUs.

Roadmap / Next Steps

HPC perspective: The project aims to enhance performance by incorporating HYPRE/PETSc GPU libraries and developing custom solvers using OpenACC and OpenMP offloading, as well as low-level models like CUDA, HIP, and SYCL. Additionally, well-tested libraries such as StatPU and Alpaka will be used to optimize performance on heterogeneous HPC machines, with a performance analysis comparing custom solvers to GPU library-supported ones.
Mathematical modeling: To enhance the stability of the existing solver, proposing a computationally stable alternative may be necessary to address performance discrepancies. Furthermore, introducing machine learning concepts to solve the entire/partial PIC algorithm could optimize the computational process, leading to more efficient and accurate results in simulations. This integration of advanced techniques with traditional methods can significantly improve the handling of complex problems within the PIC framework.
Use cases/ Physics modeling: To enhance code stability, it's crucial to test with diverse use cases using realistic data sets. Moreover, introducing complex plasma simulations with intricate geometry and physics parameters will lead to more accurate results.

Summary

HEXAPIC targets PIC simulations on large heterogeneous HPC systems. The design emphasizes performance portability across CPU and GPU architectures, allowing for more versatile deployment across various hardware. Early experiments have indicated improved scalability and efficiency over baseline approaches, highlighting the effectiveness of the design. Looking ahead, additional use cases will be introduced and validated within the HEXAPIC code to further enhance the accuracy of capturing physical behavior.

References

1. E. Krishnasamy and P. Bouvry, "Reproducible performance study of OpenACC vs. CUDA," 2024.
2. E. Krishnasamy and P. Bouvry, "OpenACC vs. CUDA for linear algebra computations," EECT, 2025.
3. E. Krisshnasamy and P. Bouvry, "CUDA vs. OpenMP offloading for BLAS operations on GPUs," Int. Conf. on ICT for Intelligent Systems, 2025.
4. E. Krishnasamy and P. Bouvry, "OpenMP offloading on AMD and NVIDIA GPUs," Int. Conf. on High Performance Compilation, Computing and Communications, 2025.
5. E. Krisshnasamy, J. Trotter, X. Cai, D. Pleiter, L. Kos, L. Saavedra, and P. Bouvry, "MPI+X offloading performance for dense matrix–vector multiplication," HPCAsia Workshops (HPCAsiaWS), 2026.

This work was supported by the Fonds National de la Recherche (FNR), Luxembourg, under the **FNR CORE** project grant number **C23/IS/18105668/HEXAPIC**, and the Slovenian Research and Innovation Agency **(ARIS)**, Slovenia.

Luxembourg National Research Fund
Slovenian Research and Innovation Agency
UNIVERZA V LJUBLJANI
project website