Pushing the Frontier in the Design of Laser-Based Electron Accelerators with Groundbreaking Mesh-Refined Particle-In-Cell Simulations on Exascale-Class Supercomputers

Luca Fedeli*, Axel Huebl†, France Boillod-Cerneux*, Thomas Clark*, Kevin Gott†, Conrad Hillairet‖,
Stephan Jaure**, Adrien Leblanc‡, Rémi Lehe†, Andrew Myers†, Christelle Piechurski§, Mitsuhisa Sato¶,
Neil Zaïm*,ac Weiqun Zhang†, Jean-Luc Vay†, and Henri Vincenti*
*LIDYL, CEA-Université Paris-Saclay, CEA Saclay, 91191 Gif-sur-Yvette, France
†Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
‡Laboratoire d'Optique Appliquée, ENSTA Paris, CNRS, Ecole polytechnique, 91762 Palaiseau, France
§GENCI, 6 bis rue Auguste Vitu 75015 Paris, France
¶RIKEN, 7-1-26, Minatojima-minami-machi, Chuo-ku, Kobe, Hyogo, 650-0047, Japan
‖Arm, 738 Avenue de Roumanille 06410 Biot - Sophia Antipolis, France
**ATOS, 1 Rue de Provence 38130 Echirolles, France
(Email: luca.fedeli@cea.fr, axelhuebl@lbl.gov, france.boillod-cerneux@cea.fr, thomas.clark@cea.fr,
kngott@lbl.gov, conrad.hillairet@arm.com, stephan.jaure@atos.net, adrien.leblanc@ensta-paris.fr, rlehe@lbl.gov,
atmyers2@gmail.com, christelle.piechurski@genci.fr, msato@riken.jp, neil.zaim@cea.fr,
weiqunzhang@lbl.gov, jlvay@lbl.gov, henri.vincenti@cea.fr)

***Abstract*—*(150 word max)* We present a first-of-kind mesh-refined (MR) massively parallel Particle-In-Cell (PIC) code for kinetic plasma simulations optimized on the Frontier, Fugaku, Summit, and Perlmutter supercomputers. Major innovations, implemented in the WarpX PIC code, include: (i) a three level parallelization strategy that demonstrated performance portability and scaling on millions of A64FX cores and tens of thousands of AMD and Nvidia GPUs (ii) a groundbreaking mesh refinement capability that provides between $1.5\times$ to $4\times$ savings in computing requirements on the science case reported in this paper, (iii) an efficient load balancing strategy between multiple MR levels. The MR PIC code enabled 3D simulations of laser-matter interactions on Frontier, Fugaku, and Summit, which have so far been out of the reach of standard codes. These simulations helped remove a major limitation of compact laser-based electron accelerators, which are promising candidates for next generation high-energy physics experiments and ultra-high dose rate FLASH radiotherapy.**

***Index Terms*—high-field science, laser-matter interaction, Plasma accelerators, Particle-In-Cell method, Adaptive mesh refinement, High performance computing, Exascale computing**

I. Justification for ACM Gordon Bell Prize

First-of-a-kind Particle-In-Cell modeling of novel integrated laser-plasma injection and acceleration scheme on millions of A64FX cores and 10-thousands of AMD/Nvidia GPUs. Time-to-solution was reduced by up to a factor of four thanks to groundbreaking mesh refinement capability, opening a new era in large-scale modeling of laser-plasma interactions at Exascale.

II. Performance Attributes

Performance attributes	Our submission
Category of achievement	Scalability, time-to-solution
Type of Method used	Explicit
Results reported	Whole application including I/O
Precision reported	Single, double precision
System scale	Full-scale system
Measurement mechanism	Timers, FLOP count

III. Overview of the Problem

A. The Physical Problem

During the past decade, in-vivo experiments have demonstrated a strong sensitivity difference between healthy and unhealthy tissues to ionizing radiations when delivered within short and bright pulses [1]. This so-called ultra-high dose rate radiotherapy (RT), also known as FLASH-RT, has the potential to revolutionize medical treatments. In ultra-high dose RT, a therapeutic dose is delivered in very few seconds at a much higher dose-rate ($>$ 30 Gy/s) than in conventional treatment protocols (0.01 Gy/s). To date, mechanisms behind the benefits of ultra-high dose rate RT have not been elucidated. This understanding requires a deeper insight into the basis of radiation toxicity on biological samples at disparate timescales ranging from femtoseconds (1 fs $= 10^{-15}$ s, molecule excitation timescale) to the hour (cellular response) and beyond. Conventional accelerators cannot provide irradiation on the fs timescale. Only laser-based particle sources [2], [3] can open this route [4] as they leverage on the high power and ultrashort duration of fs laser pulses

SC22, November 13-18, 2022, Dallas, Texas, USA
978-1-6654-5444-5/22/$31.00 ©2022 IEEE

based on the Chirped Pulse Amplification (CPA) technique [5]. By providing electron or ion bunches over very short durations (< 10 fs), currently out of the reach of conventional means, laser-based accelerators could help elucidate the mechanisms behind FLASH-RT and deliver therapeutic doses at yet unexplored dose rates ($> 10^7$ Gy/s) [6].

At present, although laser-based accelerators offer great promise for these studies, they still suffer from major limitations for their use as efficient sources for RT. In the case of laser-based electron accelerators, which are at the core of this study, the main limitation comes from the difficulty to level up the electron charge per bunch from tens of pC (picocoulomb) up to a few-nC (nanocoulomb) while maintaining a high beam quality. This has so far strongly hindered the dose one can deposit in a few seconds at realistic laser repetition rates ($<$kHz), as required in ultra-high dose rate RT.

B. Our Physical Approach

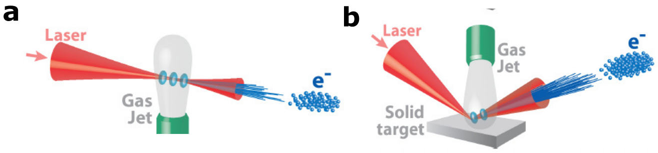

Fig. 1: Sketches showing the focusing of a high-power femtosecond laser (a) into a gas jet (b) onto a hybrid solid-gas target.

A laser-based electron accelerator, also called Laser Wakefield electron Accelerator (LWFA) [3] can be obtained by focusing a high-power fs laser on a gas target (see Fig. 1 (a)). At laser focus, the gas is quasi-instantly ionized by the ultra-intense laser field and forms a transparent plasma in which the laser pulse can propagate. During its propagation, the laser pulse violently expels electrons from its path and forms a "bubble" devoided of electrons in its wake (see Fig. 1 (a) or Fig. 2 (a) yellow scale), which can sustain huge ($\approx$100 GV/m) accelerating fields. LWFAs already demonstrated the production of GeV electron beams on a cm scale [7], making them promising candidates to build the next generation of compact particle accelerators [8]. Achieving a high-beam quality with these accelerators is a real challenge as it requires a highly localized injection of electron bunches in time (fs scale) and space (μm scale) to ensure that all electrons follow the same acceleration in the LWFA. This has been successfully demonstrated for low amounts of charge ($\approx$100 pC/bunch at 100 MeV-level down to $<$ 10 pC/bunch at GeV-level) using techniques relying on electron injection directly from the ionized gas medium [9]–[15]. Going to a higher charge/bunch at high energy with these techniques has proven extremely difficult so far as it inevitably requires longer injection times resulting in a lower beam quality (energy spread).

To break this barrier, we propose a novel scheme relying on a hybrid solid-gas target design (see Fig. 1 (b)), which can level up the electron charge per bunch of laser-based electron accelerators by almost two orders of magnitude while preserving a high beam quality. In this scheme, the ionization of the solid target by the intense laser creates a plasma orders of magnitude denser than gas and reflective for the incident light: a so-called plasma mirror (PM) [16]–[20]. When illuminated by an intense laser, this new "hybrid solid-gas" target allows both a highly-localized injection and acceleration of high-charge electron bunches as follows (see Fig. 2 (a)-(c)). First, the reflection of the laser pulse on the PM leads to the production of high charge electron bunches (injection stage). Then, the reflected laser drives a LWFA in the gas part, which traps and further accelerates the high-charge bunches (acceleration stage).

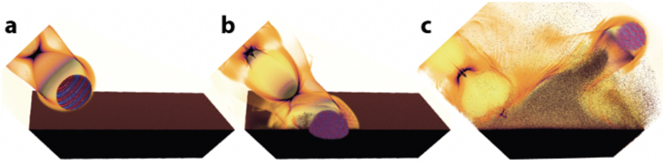

Fig. 2: Snapshots from a 3D simulation - performed with WarpX on Summit - before, during and after the reflection of the laser onto an initially solid target with gas at its front. Visualized are: in orange the electron density from the gas, in blue & red the laser field, in brown the plasma target and the electrons extracted by the laser.

C. The Computational Challenge

Owing to the high light intensities on target as well as the extremely short spatial ($< \mu$m) and temporal scales ($<$ fs) involved, the understanding of the basic physics of electron injection and acceleration cannot be directly captured from experiments and requires a full 3D first-principles kinetic in-silico modeling with the Particle-In-Cell (PIC) method [21]. The main challenge in the modeling of the hybrid target design comes from the huge interval of spatial (0.01 μm to a few mm) and temporal scales (0.01 fs to tens of ps) that need to be resolved in simulations. This both mandates the largest supercomputing machines available to date coupled with a mesh refinement strategy to apply the highest resolution where needed in the simulation. For the very first time, our team managed to implement and optimize for Exascale a massively parallel mesh-refined PIC algorithm in the WarpX code, which enabled the modeling of the hybrid target scheme. 3D mesh-refined simulations performed with WarpX on Frontier, Fugaku, and Summit supercomputers demonstrate that our scheme can resolve major limitations of current laser-based electron accelerators and provide, for the first time, 100 MeV-GeV nC electron bunches with unprecedented beam quality ($< 10\%$ energy spread). This should enable the use of LWFAs in high-impact medical and physical research studies previously out of the reach of conventional accelerators.

IV. Current State of the Art

Owing to the highly non-linear and kinetic effects involved in the interaction of an intense laser with a very dense plasma, a fully kinetic description is required, for which the PIC method has become the method of choice [22]. The governing system of equations are the coupled Vlasov-Maxwell equations, in which charged particles of the plasma interact self-consistently with electromagnetic fields. The electric and magnetic fields are discretized on a mesh

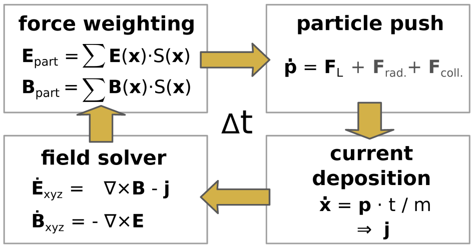
force weighting
$\mathbf{E}_{part} = \sum \mathbf{E}(\mathbf{x}) \cdot S(\mathbf{x})$
$\mathbf{B}_{part} = \sum \mathbf{B}(\mathbf{x}) \cdot S(\mathbf{x})$
particle push
$\dot{\mathbf{p}} = \mathbf{F}_L + \mathbf{F}_{rad.} + \mathbf{F}_{coll.}$
Δt
field solver
$\dot{\mathbf{E}}_{xyz} = \nabla \times \mathbf{B} - \mathbf{j}$
$\dot{\mathbf{B}}_{xyz} = - \nabla \times \mathbf{E}$
current deposition
$\dot{\mathbf{x}} = \mathbf{p} \cdot t / m$
$\Rightarrow \mathbf{j}$

Fig. 3: Explicit Particle-In-Cell cycle for an electromagnetic PIC code

Capability	Code	Epoch	Osiris	PICADOR	PIConGPU	Smilei	VPIC	WarpX
High-order particle shape*		✓	✓	✓	✓	✓		✓
Moving window*		✓	✓	✓	✓	✓		✓
Single-Source CPU & GPU*				✓	✓		✓	✓
Dyn. LB for CPU & GPU*								✓
Mesh refinement*								✓
Boosted frame			✓					✓
PSATD Maxwell field solver								✓

Table I: Advanced capabilities available in leading parallel electromagnetic Particle-In-Cell codes. All listed capabilities marked with * are essential for the science case presented here. The last two capabilities are key to extensions of this work, as discussed in the last section.

(Eulerian description) while plasma particles are treated using a Lagrangian description. In the PIC method, particles are modeled as samples of the distribution function in phase space. They are advanced in 3D space with explicit time stepping. These samples, also called macroparticles, are assigned a central position, have a B-spline charge assignment function (particle shape factor), a delta-distribution in momentum, and possibly other physical quantities such as ionization state. Fig. 3 illustrates a typical timestep of the PIC cycle, which consists of several sub-steps. First, according to the particle shape factor, the electromagnetic field is weighted to each particle to calculate the change of momentum in the particle push ("field gathering"). Then, particle positions and momenta are updated over interleaved time points (leap-frog), and the current density generated by charged particles motion is computed on the simulation grid by interpolating the product of the particles charge and velocity onto the grid cells. Usually, the computationally most demanding part of the PIC cycle is the evaluation of the current from particles to the computational field grid ("current deposition"). Finally, the generated current is fed self-consistently back into the field solver to update the electromagnetic fields.

Many electromagnetic PIC codes have been developed, mostly implementing the same core *recipe* that was established decades ago [21]: (i) second-order *leapfrog* Finite-Difference Time-Domain (FDTD) Maxwell field solver discretized on a staggered *Yee* Grid [23], (ii) second-order *leapfrog* relativistic *Boris* particle pusher [24], (iii) linear macroparticles shape factor for interpolations between the particles and fields data. The default method is based on second-order finite-difference in space and time, which makes it very easy to parallelize using domain decomposition across MPI ranks. Advanced PIC codes also usually include some OpenMP threading for additional parallelization within an MPI rank, vectorization of key loops, and dynamic load balancing on CPU-based systems.

Giving quantitative comparative estimates of where the various codes stand in performance for the science problem at hand - or even for the simple problem of a uniform neutral plasma - is very difficult, because of the wide parameter range in physical conditions (e.g., plasma densities, initial temperature of electron and ions), numerical settings (e.g., order of particle shape factor, type of field solver), implementation details (e.g., existence and strategies used for vectorization on CPUs and dynamic load balancing) and specificities of the software (e.g., compiler vendor, version and settings) and hardware. Instead, we highlight the features that differentiate our code from other leading codes and provide some quantitative estimates of their impact.

From the tens of codes that exist, we will focus on a subselection of full 3D PIC codes that are used for large-scale modeling of laser-plasma interactions: Epoch [22], Osiris [25], PICADOR [26], PIConGPU [27], Smilei [28], VPIC [29], and WarpX [30] (the code used in this study). All these codes implement the standard *recipe* of the PIC method described above. Other codes exist for the modeling of laser-plasma interaction (e.g., VLPL, V-Sim) that do not differ appreciably from some of the codes considered here and have been omitted for brevity.

While the modeling of laser-plasma interaction is possible with the standard PIC method, it is quickly becoming very expensive when the target has a very high density. Table I lists additional capabilities to standard PIC that are needed for efficient modeling of (a) a laser with a high-density target, (b) the subsequent acceleration of the electron beam that is extracted from the target, and (c) is portable across platforms:

a) High-order particle shapes: implemented in all the codes but VPIC. Use of quadratic or cubic splines to enable the modeling of high-density plasmas while mitigating the *finite grid instability* to an acceptable level, which otherwise demands higher resolution and number of macroparticles to achieve the same result at a much higher computational cost [31].

b) Moving window: implemented in all the codes but VPIC. Enables the grid to follow the laser (and extracted electron bunch) over long distance after its interaction with the solid target. For the science problem studied here, without a moving window, the simulation region would need to be nearly 7 times longer than the one used to cover the 150 microns of the laser-target interaction, in order to cover 1 mm of laser propagation distance.

c) Single-source on CPU & GPU: implemented in PIConGPU, VPIC, PICADOR, and WarpX. To our knowledge, at the time of writing, only these four codes offer a single-source implementation that runs efficiently on CPU and GPU-based machines. More details are given in the next section.

d) Dynamic load balancing for CPUs and GPUs: implemented in WarpX. While it is common to implement dynamic load balancing in PIC codes on CPUs, to the best of our knowledge, WarpX is the only one to have implemented it on GPUs, leading to demonstrated 3.8× speedup on simulations of laser interactions with high-density plasmas over simulations without load balancing [32].

e) Mesh refinement: implemented in WarpX only. This feature, which is definitely among the most advanced and unique features of WarpX is described in detail in the next section.

The other two capabilities listed in Table I were not used for the simulations reported in this paper but are important to the discussion in the last section.

V. Innovations Realized

A. Performance Portability

WarpX is a C++17 code based on the AMReX library [33]. Relevant to this paper, functionality used and extended from AMReX in WarpX are: a) parallel data containers for block-structured meshes and particles, b) domain decomposition & communication functions (including load balancing, halo exchange, and particle redistribution), c) and on-node accelerated performance portability primitives, such as `ParallelFor` and data reductions. The latter are aware of the AMReX data blocks and index space to enable single-source development. At compile-time, the corresponding accelerated backend can be chosen for GPU and CPU compute kernels: Nvidia CUDA 11+, AMD ROCm HIP 5.0+, block-wise OpenMP 3.1+ threading, and Intel OneAPI DPC++ 2022+/SYCL are production-ready choices in WarpX.

1) Optimization of the PIC algorithm on A64FX: The two main hotspots of WarpX, namely the field gather and current/charge deposition routines, consist in interpolating data between particle positions (in continuous physical space xyz) and the corresponding discrete grid mesh indices (ijk). For order 3 interpolation, the gather and deposition routines may require up to 64 sampling points per particle. In this context, strategies to optimize the memory accesses and computations had to be implemented in WarpX. For memory access optimizations, grid tiling and particle sorting are used to improve data locality. Regarding the optimization of the computational cost, enabling vectorization was key for increased performances, especially on the A64FX processor. We notably performed a tuning experiment with A64FX-specific optimizations to take advantage of its large vector registries. To that end, we implemented a multiple particle SIMD computation approach. In the following, the interpolation coefficients for particle p are noted W_{ijkp} with $(i, j, k) \in [1, 4]^3$ (the leftmost index being the contiguous index). Trying to vectorize the interpolation coefficient computation for a single particle (vectorizing over ijk with p fixed) leads to inefficient code, in particular due to very small loops. On the other hand, vectorizing the computation of the coefficient ijk for multiple particles (vectorizing over p with ijk fixed) requires some data reorganization but allows extending loops to arbitrary sizes which is ideal for vectorization. This data reorganization is done via a transposition and comes at a cost. To lower this cost, we have chosen to implement our method on small groups of cells of size N^{grp}. N^{grp} should be chosen to be large enough to favor vectorization whilst maintaining temporary arrays in the CPU cache (typical values should be powers of 2 such as 32, 64 or 128). For single precision floats, transposition operations have been heavily optimized for the *ARM Neon* instruction set. This instruction set can natively handle load/stores of vectors of 4 floats and can perform matrix transposes of 4 × 4 matrices directly in registers. We implemented these operations with *ARM Neon* intrinsics, which was our most efficient solution.

In the following table we present the speed-up obtained with the optimized routines (vs baseline) on a representative single A64FX node benchmark:

Routine	**Reference (s)**	**Optimized (s)**	**Speed up**
Gather	270.6	102.7	2.63X
Deposition	246.2	53.51	4.60X

Despite the important amount of memory accesses in these routines, our *A64FX* optimized routines still provide a significant speed up.

2) Optimization of the PIC algorithm on GPUs: Whenever possible, WarpX operates in a mode where all the core simulation data (mesh variables and particles) resides in GPU device memory persistently, to avoid the overhead associated with frequent host-device transfers. This includes the communication buffers used to store particle data that needs to be transferred to other MPI ranks. However, particularly in the presence of dynamic load balancing and/or dynamic mesh refinement, the memory needed by these buffers on a single GPU can spike. To handle this case, we have implemented a fall-back strategy where particle communication buffers are written in pinned memory instead of device memory if there is not enough space left on the device. This option, while slower, is needed to avoid out-of-memory errors during large load balancing steps. Since load balancing and/or the removal of the mesh refinement patch is only performed occasionally, this trade-off has a negligible effect on overall performance. Similar space/time tradeoffs are made, for example, when sorting particles if the fastest approach results in too much memory use.

B. Mesh-Refinement in Electromagnetic PIC

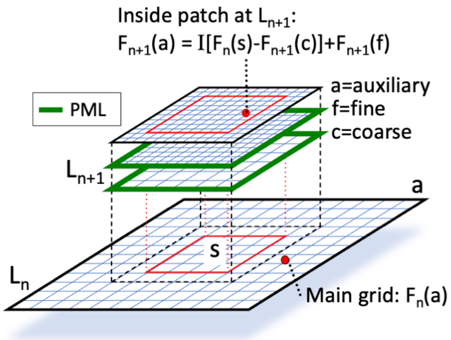
Inside patch at L_{n+1}:
$F_{n+1}(a) = I[F_n(s)-F_{n+1}(c)]+F_{n+1}(f)$
PML
a=auxiliary
f=fine
c=coarse
L_{n+1}
a
s
L_n
Main grid: $F_n(a)$

Fig. 4: Sketch of the implementation of mesh refinement in WarpX.

One of WarpX's unique features for a fully electromagnetic Particle-In-Cell code is its implementation of mesh refinement. The mesh refinement technique that is used in the electromagnetic mode of WarpX (see Fig. 4) is quite different [34], [35], and more complex, than the ones used traditionally in, e.g., shock hydrodynamics or combustion [36], where the connection between the solutions at various levels of refinement relies largely on interpolations at the boundary, and sometimes interior, values. Such an approach is not directly applicable to solving the Maxwell equations, due to the generation of spurious reflections of electromagnetic waves from interpolations, and more complex algorithms are necessary [34], [35], [37]. In the algorithm implemented in WarpX, additional grids are collocated with the refinement patch and are terminated by absorbing layers (e.g., Perfectly Matched Layers [38] *a.k.a.* PMLs) to prevent the reflection of electromagnetic waves. The linearity of Maxwell's equations is used to reconstruct the field to push particles inside a refined region based on the underlying set of grids construct (see [35] for details). An additional coarse grid c is added to the fine grid patch f so that the full solution is obtained on an auxiliary grid a by *substitution*.

The detailed steps of WarpX's PIC loop with mesh refinement are as follows (given without subcycling of the refined levels steps for clarity). The current density resulting from the motion of charged macroparticles within the refined region is accumulated on the fine patch, interpolated onto the coarse patch and added onto the parent grid. The process is repeated recursively from the finest level down to the coarsest. When not using subcycling, the Maxwell equations are then solved for one time interval on the entire set of grids. The field on the coarse and fine patches only contains the contributions from the particles that have evolved within the refined area but not from the current sources outside the area. The total contribution of the field from sources within and outside the refined area is obtained by adding the field from the refined grid $F(r)$, and adding an interpolation I of the difference between the relevant subset s of the field in the parent grid $F(s)$ and the field of the coarse grid $F(c)$, on an auxiliary grid a, i.e., $F_{n+1}(a) = F_{n+1}(r) + I[F_n(s) - F_{n+1}(c)]$. The field on the parent grid subset $F(s)$ contains contributions from sources from both within and outside of the refined area. Thus, in effect, there is substitution of the coarse field resulting from sources within the patch area by its fine resolution counterpart. The operation is carried out recursively starting at the coarsest level up to the finest.

Such a complex construct is needed because of the singular nature of the coupling of electromagnetic fields and macroparticles that can create spurious wave reflections and unphysical forces and require special techniques for mitigation [34], [35]. The field solution at a given level L_n is unaffected by the solution at higher levels L_{n+1} and up, allowing for mitigation of some spurious effects [35] by providing a transition zone via extensions of the patches by a few cells beyond the desired refined area (red rectangle) in which the field is interpolated onto particles from the coarser parent level only.

In order to increase efficiency, an option has been implemented to subcycle the operations at the refined levels, in which various grid levels are pushed with different time steps, given as a fixed fraction of the individual grid Courant condition (assuming same cell aspect ratio for all grids and refinement by integer factors). In this case, the fields from the coarse levels, which are advanced less often, are interpolated in time. The description of the subcycling algorithm is omitted for brevity and is described here [39].

Finally, the implementation of mesh refinement in WarpX is compatible with the use of a moving window, where the grid follows a physical phenomenon of interest, here the laser pulse after interaction with the solid target, which is not possible with any other PIC code that we are aware of.

C. *Multi-Level, Dynamic Load Balancing for GPUs*

In WarpX, each level of refinement consists of a union of rectangular grid patches called *boxes*. Domain decomposition is accomplished by assigning these boxes to MPI ranks according to some load balancing strategy; currently supported strategies are *round robin*, which simply loops over the boxes in order, giving one to each rank until all the boxes have been assigned, *space-filling curve*, which attempts to place spatially close boxes on the same MPI rank by putting the boxes in Z-sorted order, and *knapsack*, which attempts to evenly distribute the load with no consideration of locality by heuristically solving the knapsack problem for a given set of costs. By default, WarpX uses the space-filling curve strategy with no consideration on the number of particles in each box. This approach works well to minimize communication costs when exchanging guard cell information but results in sub-optimal performance with largely load-imbalanced particle distributions. Additionally, the load balancing changes as the simulation evolves and when dynamic mesh refinement is employed, such as when the refinement patch is removed. Thus, WarpX also has the ability to dynamically redistribute the boxes on-the-fly based on a number of heuristics and on measured runtime cost information. Another optimization was, to the extent possible, to co-locate the PML patches that surround both the problem domain and the refined region with the parent grid they are spatially closest to. Because the PMLs must frequently exchange data with these grids, placing them when possible on the same MPI rank led to significant (25%) performance increases in WarpX simulations that use PMLs.

VI. How Performance Was Measured

The following section summarizes the benchmark steps performed to characterize the performance of WarpX on the world's largest supercomputers. Performance is measured and where projections are applied, this is explicitly highlighted. This section introduces the tested precision/compilation modes, weak and strong scaling, flop measurements, and figure-of-merit analyses.

WarpX can be compiled either in single-precision (SP) or double-precision (DP). Currently, WarpX's single precision mode is a mixed-precision (MP) mode, in which some numerically sensitive, particle-related operations are cast to double precision. For the setups for which it is applicable, users prefer the MP mode since it offers a faster time-to-solution and it was consequently used in the science run as well. WarpX measurements were performed in both modes to allow comparisons with published High-Performance Conjugate Gradient (HPCG) results, which are DP.

Exact WarpX and AMReX commits, compilation options, additional compilation flags, and compiler versions on respective machines are part of the data artifact [40]. For ARM A64FX CPUs, the Fujitsu compiler version 4.8.0 with Clang frontend was used, since the traditional frontend produced code that led to runtime errors. For Nvidia GPUs, `nvcc` v11.3 & v11.5 were used and for AMD GPUs, ROCm/HIP v5.1. `-O3` optimization flags and fast-math were enabled for each run.

A. Weak- and Strong-Scaling

Weak- and strong-scaling results, as well as FOM numbers, were obtained by measuring the time-to-solution on the presented number of ranks, up to the full machine. WarpX was run for 100 (Fugaku) or 1 000 (all other machines) time steps with average steps taking about 1-2 seconds (Fugaku) or 0.5-1 seconds (all the other machines). Results are full application runtime after MPI initialization: simulation data initialization, compute, and light self-diagnostics ; timings are obtained once. Fugaku and Perlmutter benchmarks were performed in reservations, avoiding any influence on performance from runs of other users. Frontier and Summit runs were performed in shared operation hours and showed no unusual variation compared to periodically performed measurements of WarpX on the same machines.

For weak-scaling, the numerical size of the simulation domain was increased linearly with the available number of compute resources. The number of time steps computed was kept the same for the scaling, which keeps the data per node and Flop per node in WarpX constant. In strong-scaling, a multi-node scenario with maximally filled GPU memory was picked as the basis. Then the number of compute resources was increased until the level of granularity in AMReX was reached, which is one block of cells per device. The following block sizes where used: Frontier=256^3, Fugaku=64^3-96^3, Summit=128^3, Perlmutter=128^3. Presented numbers in Figure 5 are from WarpX DP runs for all graphs.

B. Sustained Flop/s

Floating Point operations (Flop) on OLCF Summit GPUs were measured using Nsight Compute 2022.1.0 and include the following SASS opcodes: `FADD`, `FFMA`, `FMUL` (SP) and `DADD`, `DFMA`, `DMUL` (DP). FMA (fused multiply-add) operations are counted as two floating point operations, all other operations as one.

Special care was taken to count operations per thread, respecting instruction predicates via the `thread_inst_executed_true` metric. This is more detailed than the usual statistics collected on the warp-level from `--section InstructionStats`, which relies on the metric `inst_executed` per warp and is independent of the number of participating threads within each warp. Thus, only GPU Flop counts of active GPU threads are used to report average Flop/s for WarpX execution.

Since Nvidia SASS instructions have no direct support for floating point divisions, such operations - which occur regularly in WarpX's numerical algorithms - are emulated on Nvidia GPUs. Besides others, we see this as an increased number of `IMAD` (integer multiply-add) operations. The full count details of all SASS operations are part of the data artifact [40].

We measured Flop/s from weak-scaling runs, scaled to two nodes. WarpX uses 1 MPI rank to control one GPU. Since all ranks have the same work in our uniform plasma tests, the last MPI rank was picked for Flop counting over 1 (Perlmutter) to 3 (Summit) steps. To arrive at Flop/s per GPU for the application, the measured counts were divided by the uninstrumented runtime.

Flops on OLCF Frontier GPUs were measured using the AMD ROCm Profiler. As for Nvidia GPUs, the `ADD`, `MUL`, and `FMA` instructions are counted for both double and single precision operations. We performed two single GPU runs with one running one more step than the other. The difference in the flop counts gives the total number of operations in one step. The Flop/s is calculated by combining it with the wall time for that step in an uninstrumented run.

For ARM A64FX CPUs on Fugaku, we measured the floating point operation count on a single node via the `fipp/fapp` (Fujitsu Instant/Advanced Performance Profiler) tools. `fipp` provides undifferentiated "Floating-point operation" counters for both SP and DP. However, it provides insights into the fraction of the SIMD vectorized operations and of the SVE operations, showing that in the version of our code specifically optimized for A64FX the SIMD instantaneous rate increases from $\sim 2.3\%$ to $\sim 24\%$, and the SVE operation rate increases from $\sim 3.6\%$ to $\sim 70\%$. The `fapp` tools provides more detailed insights, including separate counters for SP and DP floating point operations.

The achieved Flop/s at system-scale are based on the measured few-node Flop counts and scaled with the measured weak-scaling efficiency.

C. Figure-of-Merit

In the U.S. Department of Energy Exascale Computing Project (ECP), participating applications track performance improvement over time as they port to GPU-based Exascale systems. Each application's Figure of Merit (FOM) is different, based on their respective domain science goals. As WarpX's target is runtime improvement for system-scale simulations, the FOM was designed to measure the time-to-solution of a selected laser-plasma modeling problem, scalable on the size of the problem and the percentage of the system on which it was run. WarpX's chosen FOM is given by Equation (1):

$$\text{FOM} = \frac{\alpha N_c + \beta N_p}{\text{avg. time per step} \cdot \text{percent of system used}} \quad (1)$$

with $\alpha = 0.1$, $\beta = 0.9$, N_c the number of cells, N_p the number of particles in the simulation. α and β were chosen

Table II: Machines used in this study, central computing hardware, vendor-specified maximum TFlop/s per device [42]–[45], and published 2021/11 HPCG benchmark results on the full machine [46], [47].

Machine	Compute	TFlop/s	TByte/s	HPCG
	Hardware	per device	per device	PFlop/s
Frontier	MI250X	DP: 47.9 SP: 95.7	3.3	not yet available
Fugaku	A64FX	DP: 3.38 SP: 6.76	1.0	16.0 (158 976 nodes)
Summit	V100 SXM2 (16GB)	DP: 7.5 SP: 15	0.9	2.93 (4 608 nodes)
Perlmutter	A100 SXM2 (40GB)	DP: 9.7 SP: 19.5	1.6	1.91 (1 424 nodes)

based on the initial measured ratio of mesh-to-particle work at the beginning of the study and have been maintained for consistency. When collecting data for our FOM, we measure as close to full-system size as possible and then extrapolate to the full machine.

VII. Performance Results

Benchmarks were performed on some of the fastest supercomputers today with respect to the TOP500 HPL & HPCG benchmarks [41]: TOP1/TBA Frontier (OLCF), TOP2/1 Fugaku (RIKEN), TOP4/2 Summit (OLCF), and TOP7/4 Perlmutter (NERSC). Frontier is the first reported machine above 1 ExaFlop/s in HPL. Experiments were run in two phases: between March-April and during July, 2022. Perlmutter and Frontier tests were run while the systems were in early, pre-production/pre-acceptance state. Reported Perlmutter tests are on a Slingshot 10 network; first tests on Perlmutter with Slingshot 11 showed performance improvements of about 5 % up to 128 nodes in time-to-solution. A brief summary of computing architectures of these machines is given in Table II.

A. Weak- and Strong-Scaling

Scalability tests were performed using uniform plasma simulations. As of July 2022, the following system sizes were available: Frontier: 9 316 nodes (out of 9 472 total nodes), Fugaku: 152 064 nodes (out of 158 976 total nodes), Summit: 4 608 nodes, Perlmutter: 1 100 (out of 1 500) nodes. Collected results are presented in Figure 5.

Weak-scaling tests were performed on the following range of nodes: Frontier: 1 to 8 576 nodes; Fugaku: 1 - 152 064 nodes; Summit: 2 - 4 263 nodes; Perlmutter: 2 - 1 088 nodes.

Weak-scaling results are on the left of Figure 5. Ideal scaling is marked as a horizontal, dotted grey line at 100 %. The scaling efficiency is lowered by any relative increase in runtime compared to the smallest run of each measurement. Note that the node numbers in weak scaling are shown over 4 orders of magnitude. For Frontier and Fugaku, the results stay very close to the ideal scaling. Efficiency barely decreases over 4 orders of magnitude and ultimately, at full-system size, reaches 80 and 84 % efficiency, respectively.

For Summit, weak scaling efficiency only degrades slightly at scale, besides a 15 % loss in efficiency from 2-8 nodes (12, 24, and 48 GPUs). Profiling details revealed that this can be attributed to halo exchange operations that include more buffer preparations on GPU than CPU, as average communication pairs for next-neighbor synchronizations in 3D decrease for runs smaller than $3 \times 3 \times 3 = 27$ ranks. The efficiency at 4 263 nodes is 74 %. Perlmutter efficiency, with Slingshot 10 network, drops gradually, by 11 % for 30 nodes and finally to 62 % at 1088 nodes.

Connecting these results to the need in laser-plasma interaction modeling underlines why efficient weak-scaling is essential: many scenarios in laser-plasma modeling are resolution-limited and require large machines to hold and advance the simulation state - and the results presented herein show that such immense scaling of many orders of magnitude is possible with WarpX.

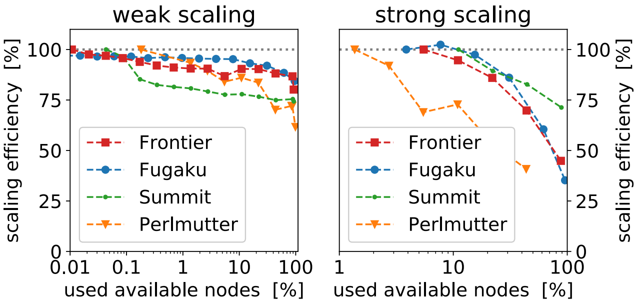

Fig. 5: Weak and strong scaling of WarpX. Perfect scaling is the grey dotted line at 100 % in both graphs.

Strong-scaling tests were performed on Frontier: 512 to 8 192 nodes, Fugaku: 6 144 to 152 064 nodes, Summit: 512 to 4 096 nodes and Perlmutter: 15 to 480 nodes. As WarpX performs domain decomposition on patches (AMReX blocks), there is a limit on how fine-grained one can distribute the computational domain before running out of blocks to work on per rank. Typically, one assigns 1-4 blocks per 10 GB device memory [30] to achieve high occupancy per device, while providing granularity for dynamic load balancing. For a given science case, one would usually fill GPUs memory as much as possible, since 3D laser-plasma modeling is, as mentioned earlier, limited by resolution. The strong-scaling on the right of Figure 5 shows comparable gradients for the measured machines as the same simulation setup is distributed over more compute nodes. Offsets on the horizontal axis are due to varying starting points for the different machine sizes and available nodes at the time of measuring. This demonstrates that reasonable WarpX strong-scaling can be achieved over a variety of architectures to improve time-to-solution, loosing only about 30 % efficiency over an order of magnitude scaling.

B. Sustained Flop/s

Electromagnetic Particle-In-Cell simulations are typically dominated by the deposition of current density from particles to the grid (a scatter operation) and the gathering of the electromagnetic fields from the grid onto the particles (a gather operation). Both operations involve many access to memory with relatively little computation, and PIC codes are in effect typically memory-bound. This leads to relatively low utilization of Flop/s compared to peak values. A landmark of what is considered a very high Flop/s for electromagnetic PIC code was achieved with VPIC on Roadrunner (2008) [48], where a

Table III: Flop/s measured on different architectures. † refers to the version of the code specifically optimized for A64FX CPUs. Achieved Flop/s is the largest weak-scaling run.

Machine	Mode	TFlop/s	Achieved PFlop/s
		per device	(% HPCG)
Frontier	DP	DP: 1.58 (3.3 %)	43.45
	MP	SP: 1.43 (1.5 %)	(HPCG not yet available)
		DP: 0.56 (1.2 %)	
Fugaku	DP	DP : 0.037 (1.1 %)	5.31 (34.7 %)
	MP	SP : 0.036 (0.53 %)	
		DP: 0.003 (0.01 %)	
	MP†	SP : 0.12 (1.78 %)	17.3
		DP: 0.0 (0.0 %)	
Summit	DP	DP: 0.62 (8.3 %)	11.785 (435 %)
	MP	SP: 0.64 (4.3 %)	
		DP: 0.22 (2.9 %)	
Perlmutter	DP	DP: 1.26 (12.9 %)	3.38 (223 %)
	MP	SP: 1.33 (6.8 %)	
		DP: 0.31 (3.2 %)	

sustained 0.374 PFlop/s were obtained out of the 2.82 PFlop/s peak of the machine, for a 13.2% peak utilization. What is considered a very high PFlop/s for a PIC code demands finely tuned optimizations that are usually not easily portable, if at all. Recently, VPIC was rewritten using Kokkos, leading to a single source version of the code (VPIC2.0) that runs on both CPUs and GPUs [29]. While no Flop/s measurements are available, comparisons between VPIC2.0 and VPIC1.2 show a degradation between 25% on Fujitsu A64FX to over 5× on Xeon 8280s [29]. Following these results and earlier research with PIConGPU on X86 & Power8 [49], one would expect sustained Flop/s to be in the range of 2 to 10% of the peak for PIC codes.

In effect, the operations that are dominant in electromagnetic PIC codes (in particular, current deposition and field gathering) have similar memory locality and arithmetic intensity as the ones that are used in the High-Performance Conjugate Gradient (HPCG) Benchmark suite (e.g. local stencil operation, sparse linear algebra). In general, these operations have much lower arithmetic intensity than the dense linear algebra used in the standard HPL benchmark. Therefore, here we use the HPCG benchmark (instead of the HPL benchmark) as a proxy to provide some ballpark reference.

Results on floating point operations per second for the individual precision modes of WarpX and A64FX patches are presented in Table III.

Nvidia V100 & A100 GPU measurements with Nsight Compute indicate a high but reasonable Flop/s result, given the arithmetic intensity of a PIC code [48], [49], relative to vendor peak numbers. 8.3-12.9 % DP peak performance for Nvidia GPUs is exactly in the aforementioned typical utilization range. The main difference between Summit (V100) and Perlmutter (A100) results in terms of relative floating point intensity to peak vendor TFlop/s per device can be explained primarily with the increased memory bandwidth on A100. The GPU memory bandwidth per peak Flop/s increased between V100 and A100 by 1.37×, which favors the memory-bound kernels in the PIC loop [30]. Looking at the full system-scale comparison between the two Nvidia GPU systems, Summit and Perlmutter, the main difference at scale besides system size are the WarpX peak flop rate (favors Perlmutter) and weak-scaling results (currently favoring Summit).

On Fugaku A64FX CPUs, the `fipp` tool showed an increase in SIMD vectorized operations when the version of the code specifically optimized for this architecture was used.

On Frontier, the ROCm Profiler was used for measuring Flop/s. The names of the counters and metrics used were still under a Non-Disclosure Agreement as of the writing of this paper. Comparing memory bandwidth ratio to Nvidia GPUs indicates that further optimizations to WarpX might be possible for AMD HIP kernels, i.e., to approach the ratio of 2x between MI250X and A100.

C. Figure-of-Merit

As part of the DOE Exascale Computing Project, the FOM was measured over time for WarpX, first on the Knights Landing based Cori supercomputer, then on the NVidia V100 based supercomputer Summit, and ultimately on Frontier in July 2022. A significant refactoring of the code was made in 2019, as the code was converted from CPU-optimized mix of C++ with Fortran to a C++-only single source code optimized for both CPUs and GPUs. Fortran was initially used for the central Particle-In-Cell operations (i.e. Maxwell solver stencil operations and particle updates) which have since all been converted to C++17 and GPU native languages via AMReX performance primitives.

As the writing of the new version of WarpX progressed, several features were introduced to improve performance on GPUs, e.g., reducing the amount of data stored per particle, fusing of communication kernel launches into a single kernel, restructuring of AMReX parallel communication routines to take advantage of GPU-aware MPI implementations and periodic sorting of particles for better cache performance [30]. These improvements are reflected in the steady improvement of the FOM of WarpX on Summit from 6/19 until now, as can be seen on Table IV.

As expected, the measured FOM is led by Frontier, followed by Fugaku, Summit and Perlmutter, demonstrating that WarpX is able to utilize both GPU-based and ARM-based CPU supercomputers at scale.

D. Mesh Refinement

Mesh refinement can lead to a substantial reduction of the time-to-solution in several physical scenarios where a high spatial resolution is required either in a small region of space or for a limited amount of time. Figure 6 shows a comparison between different numerical simulations of the same physical scenario, with and without mesh refinement. The simulated physical scenario requires a high spatial resolution for a limited time in a relatively large portion of the simulation domain. Initially, the computational cost of the case with mesh refinement is similar to that of a simulation without mesh refinement but with twice the coarse resolution and having the same total number of macro-particles. However, as soon as the high-resolution patch is removed, the simulation becomes substantially faster. The case with mesh refinement has a clear advantage with respect to a simulation performed with twice

Table IV: Progress in the FOM measurement over time. Date: the date when the measurement was taken. Machine: which computer was used to make the measurement. N_c/Node: the problem size in number of cells per node. Nodes: how many nodes the measurement was performed on; there are 9 668 KNL nodes on Cori, 4 608 nodes on Summit, 158 976 nodes on Fugaku, 1 526 nodes on Perlmutter (June 2022: up to 1100 available), and 9 472 nodes on Frontier (July 2022: up to 9 316 available). FOM: the figure of merit, extrapolated from the number of nodes the measurement was taken on to the full machine. All numbers are in DP mode, unless marked as † for MP.

Date	Machine	N_c/Node	Nodes	FOM
3/19	Cori	0.4e7	6 625	1.0e11
6/19	Summit	2.8e7	1 000	7.8e11
9/19	Summit	2.3e7	2 560	6.8e11
1/20	Summit	2.3e7	2 560	1.0e12
2/20	Summit	2.5e7	4 263	1.2e12
6/20	Summit	2.0e7	4 263	1.4e12
7/20	Summit	2.0e8	4 263	2.5e12
3/21	Summit	2.0e8	4 263	2.9e12
6/21	Summit	2.0e8	4 263	2.7e12
7/21	Perlmutter	2.7e8	960	1.1e12
12/21	Summit	2.0e8	4 263	3.3e12
4/22	Perlmutter	4.0e8	928	1.0e12
4/22	Perlmutter†	4.0e8	928	1.4e12
4/22	Summit	2.0e8	4 263	3.4e12
4/22	Fugaku†	3.1e6	98 304	8.1e12
6/22	Perlmutter	4.4e8	1 088	1.0e12
7/22	Fugaku	3.1e6	98 304	2.2e12
7/22	Fugaku†	3.1e6	152 064	9.3e12
7/22	Frontier	8.1e8	8 576	1.1e13

the coarse resolution, being essentially between 1.5 and 4 times faster after the removal of the high-resolution patch, depending on the assumptions on particles-per-cell in the simulations without mesh refinement. The figure also shows that mesh refinement can be used in combination with other advanced features of WarpX, such as the moving window.
The mesh refinement science-case run was tested on Frontier, Fugaku, and Summit. In particular, we successfully performed a 3D simulation with mesh refinement on 8 192 Frontier nodes (i.e., almost the full machine).

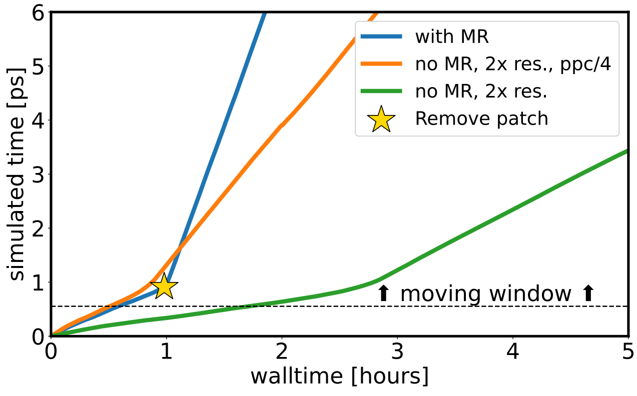

Fig. 6: Comparison of the time-to-solution for three 2D simulation cases: a) "with MR", a reduced two-dimensional version of the case simulated on 4K Summit nodes (see Fig. 7). b) "no MR, 2× res., ppc/4" the same physical scenario simulated without mesh refinement but with twice the spatial resolution and with a reduced number of particles-per-cell in order to match the total number of macro-particles of case a). c) "no MR, 2× res." same as b) but with as many particles-per-cell as a). Simulations in this figure were performed on 50 Fugaku nodes. The star marks the point where the fine patch is removed, while the dashed line marks the starting time of the moving window.

VIII. Implications

A. *Physics Impact*

The highly-scalable MR-PIC code enabled the 3D modeling of electron injection and acceleration with the hybrid solid-gas target introduced in Fig. 1 (b). Fig. 7 displays results from a 3D simulation performed with WarpX at full scale on Summit supercomputer (4K nodes, with MR enabled), compared to a 2D simulation with MR on Summit and a 3D simulation without MR on Fugaku. Simulation parameters are detailed in the caption of Fig. 7. The target density was varied slightly between the Summit and Fugaku simulations, to explore the sensitivity of the electron beam extraction to this parameter. In those simulations, a moving window follows the laser pulse during its propagation. In the simulations with MR, the MR patch follows the moving window while still covering the high-density target, which needs higher resolution. As soon as the high-density target is completely out of the simulation domain, the refinement patch is removed from the simulation.

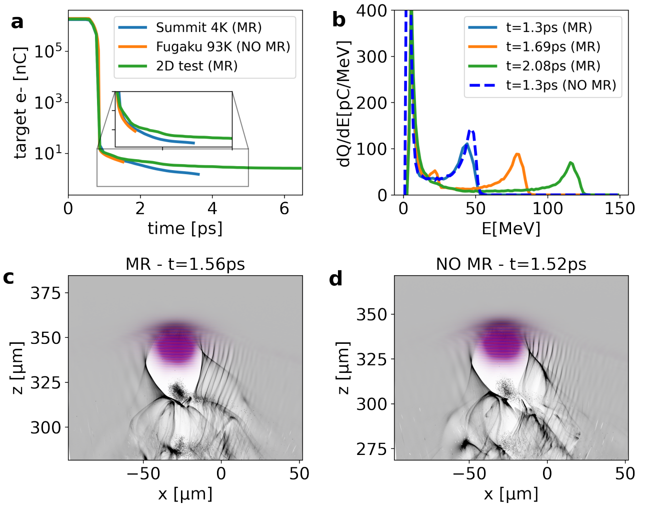

Fig. 7: In-silico 3D modeling with WarpX of laser interaction with a hybrid target. In those simulations, the gas density is $2.34 \times 10^{18} cm^{-3}$ and the solid density is 50 n_c for simulations on Summit and 55 n_c for the simulations on Fugaku, where n_c is the critical density associated to the laser wavelength $\lambda = 0.8\ \mu m$. The laser is a PW-class laser with a waist of 19.5 μm and a duration of 30.8 fs. The laser impinges at oblique incidence (θ =45°) on the solid target. Two simulations have been performed in those conditions: (i) a first simulation with MR performed at full-scale on Summit (4K nodes). This simulation involved two MR levels: a first level with $3840 \times 1920 \times 3840$ grid nodes and a second level with $7680 \times 2151 \times 5120$ grid nodes. $3 \times 2 \times 3$ macroparticles per cell were used for solid electrons whereas for the gas electrons two species were used, the first one for the initial part of the simulation (interaction with the solid) with : $2 \times 2 \times 2$ macroparticles per cell and the second one for the long-distance propagation part with $1 \times 1 \times 2$ macroparticles per cell. (ii) a second simulation performed without MR on 60% of Fugaku (93k nodes). This simulation involved $4608 \times 2304 \times 4608$ grid nodes and $2 \times 2 \times 2$ macroparticles per cell were used for solid electrons and $1 \times 1 \times 1$ for both species of gas electrons. (a) Beam charge in the simulation window. At times > 1 ps where the solid target leaves the simulation window, this charge roughly corresponds to the injected charge in the LWFA. (b) Electron beam spectrum (computed from electrons in the simulation window). (c) Snapshot of plasma electron density (gray) and laser amplitude (color scale) in the x-z plane - with MR. (d) Same as (c) but without MR.

MR simulations show that a substantial amount of charge ($\approx$ nC) can be injected from the solid part of the target and accelerated by the LWFA generated in the gaseous part (see blue curve in Fig. 7 (a)). Electron injection is highly localized (see Fig. 7 (c)) leading to peaked electron spectra with an energy spread below 10% above 100 MeV (cf. Fig. 7 (b)). This cutting-edge 3D simulation establishes the hybrid solid-gas target as a promising scheme to provide e- beam with high charge and excellent beam quality.

The validity of the MR-PIC simulation was established through a comparison with a second simulation performed on 60 % of Fugaku (93K nodes) without MR and at an intermediate resolution between the highest and lowest used in the MR simulation. This has been illustrated in Fig. 7 (a) and (b) showing that the amount of injected charge and the associated electron beam spectra agree well with or without MR. Fig. 7 (c-d), displaying snapshots of the laser amplitude (color) and plasma density (grayscale) after reflection on the solid target, also shows a good agreement between the two cases. The slight differences between the two snapshots might indicate variations from the target density, but it is also likely that simulations are not yet fully converged at this resolution, pointing to the need for exascale computing power to reach a high degree of convergence on the modeling of this problem.

Comparing to 2D and 3D simulations, Fig. 7 (a) confirms in 2D the amount of injected charge (green curve) at the beginning of the interaction but gives wrong results at later times (the diffraction of the laser beam being different in 2D and 3D). This highlights the importance of full 3D simulations in the design and optimization of laser-based accelerators.

As demonstrated by this first full-scale MR simulation, the hybrid solid-gas scheme has the potential to remove a major limitation of laser-based accelerators and enable their use in applications requiring a high-charge and a high-beam quality such as the building of compact high-energy particle colliders based on LWFA stages, the building of miniature X-FEL for ultra-fast science or ultra-high dose rate RT.

B. Exascale Impact

As shown by the results presented in this paper, reaching convergence for design and tolerance studies of laser-based beam injection and acceleration systems is requiring Exascale-class computing capabilities that will offer much larger computational power than what is available today. The latest and ongoing developments in WarpX are putting the code in an ideal and strong position to use the full range of upcoming Exascale supercomputers efficiently, whether they will be GPU/accelerator or CPU based. ARM tuning as a special-platform tuning experiment identified opportunities for generalization of AMReX parallelism primitives and data structures. The use of mesh refinement that is being pioneered in WarpX is already allowing the combined modeling of the injection and acceleration stages with unprecedented efficiency. Additional developments that will allow the use of adaptive collections of refinement patches that conform more closely to the high-density target and on a higher number of refinement levels, and couple to adaptive particle splitting and merging, will provide even higher opportunities for increased efficiency for adjusting local grid and particle statistic resolution. These will also offer an increased level of granularity that will be key to enabling efficient dynamic load balancing, which will be essential at Exascale.

To our knowledge, the use of mesh refinement in large-scale electromagnetic PIC simulations is a first and represents a paradigm shift. The successful modeling with savings between $1.5\times$ and $4\times$ with mesh refinement that is reported in this paper is a landmark stepping stone toward a new era in the modeling of laser-plasma interactions.

The WarpX code and simulations reported in this paper are setting the stage for the integrated modeling of the injection stage followed by chains of meter-long plasma accelerator stages for acceleration to high energy, up to the modeling of chains of hundreds to thousands of stages for the design of future plasma-based high-energy physics colliders. WarpX will be ideal for these simulations on Exascale supercomputers, thanks to unique features such as the modeling in Lorentz boosted frame, which gives several orders of magnitude speedups over standard laboratory-frame modeling [50], and unique algorithms for control of the *numerical Cherenkov instability* using properties of the Pseudo-Spectral Analytical Time-Domain Maxwell solver [51], [52] (the last two capabilities in Table I).

ACKNOWLEDGMENTS

WarpX is supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. Primary WarpX contributors are with LBNL, LLNL, CEA-LIDYL, SLAC, DESY, CERN, and Modern Electron. We acknowledge all the contributors and users of the WarpX community who participate to the code quality with valuable code improvements and important feedback. CEA-LIDYL actively contributes to the co-development of WarpX. As part of this initiative, WarpX also receives funding from the French National Research Agency (ANR - Plasm-On-Chip), the Horizon H2020 program and CEA. We thank the French National Technology Watch group (Cellule de Veille technologique - CVT) coordinated by GENCI for providing us an early access to the Joliot-Curie A64FX (CEA/TGCC) as supporting the team on code portability and optimization through the support of Atos, Arm and Fujitsu. We also thank IDRIS and its teams for providing us access to the accelerated partition of the Jean Zay machine. We thank F. Schmitt and M. Strengert (both Nvidia) for support in measuring Flop counts on Nvidia GPUs. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. We are grateful for support by OLCF leadership and system support in executing our runs. This research used resources of the National Energy

Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 using NERSC award ASCR-ERCAP0022112. The authors would like to thank Stony Brook Research Computing and Cyberinfrastructure, and the Institute for Advanced Computational Science at Stony Brook University for access to the innovative high-performance Ookami computing system, which was made possible by a $5M National Science Foundation grant (#1927880). This work used computational resources of the supercomputer Fugaku provided by RIKEN through the HPCI System Research Project (Project ID: ra010013).

References

[1] V. Favaudon *et al.*, "Ultrahigh dose-rate flash irradiation increases the differential response between normal and tumor tissue in mice," *Sci. Transl. Med.*, vol. 6, no. 245, 2014. [Online]. Available: https://doi.org/10.1126/scitranslmed.3008973

[2] A. Macchi, M. Borghesi, and M. Passoni, "Ion acceleration by superintense laser-plasma interaction," *Rev. Mod. Phys.*, vol. 85, pp. 751–793, May 2013. [Online]. Available: https://doi.org/10.1103/RevModPhys.85.751

[3] E. Esarey, C. B. Schroeder, and W. P. Leemans, "Physics of laser-driven plasma-based electron accelerators," *Rev. Mod. Phys.*, vol. 81, pp. 1229–1285, Aug 2009. [Online]. Available: https://doi.org/10.1103/RevModPhys.81.1229

[4] J. Bourhis *et al.*, "Treatment of a first patient with flash-radiotherapy," *Radiother. Oncol.*, vol. 139, pp. 18–22, 2019. [Online]. Available: https://doi.org/10.1016/j.radonc.2019.06.019

[5] D. Strickland and G. Mourou, "Compression of amplified chirped optical pulses," *Opt. Commun.*, vol. 55, no. 6, pp. 447–449, 1985. [Online]. Available: https://doi.org/10.1016/0030-4018(85)90120-8

[6] L. Pommarel *et al.*, "Spectral and spatial shaping of a laser-produced ion beam for radiation-biology experiments," *Phys. Rev. Accel. Beams*, vol. 20, p. 032801, Mar 2017. [Online]. Available: https://doi.org//10.1103/PhysRevAccelBeams.20.032801

[7] A. J. Gonsalves *et al.*, "Petawatt laser guiding and electron beam acceleration to 8 gev in a laser-heated capillary discharge waveguide," *Phys. Rev. Lett.*, vol. 122, p. 084801, Feb 2019. [Online]. Available: https://doi.org/10.1103/PhysRevLett.122.084801

[8] W. Leemans and E. Esarey, "Laser-driven plasma-wave electron accelerators," *Phys. Today*, vol. 62, no. 3, pp. 44–49, 2009. [Online]. Available: https://doi.org/10.1063/1.3099645

[9] J. Faure, C. Rechatin, A. Norlin, A. Lifschitz, Y. Glinec, and V. Malka, "Controlled injection and acceleration of electrons in plasma wakefields by colliding laser pulses," *Nature*, vol. 444, no. 7120, pp. 737–739, 2006. [Online]. Available: https://doi.org/10.1038/nature05393

[10] C. Geddes *et al.*, "Plasma-density-gradient injection of low absolute-momentum-spread electron bunches," *Phys. Rev. Lett.*, vol. 100, no. 21, p. 215004, 2008. [Online]. Available: https://doi.org/10.1103/PhysRevLett.100.215004

[11] C. E. Clayton *et al.*, "Self-guided laser wakefield acceleration beyond 1 GeV using ionization-induced injection," *Phys. Rev. Lett.*, vol. 105, p. 105003, Sep 2010. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.105.105003

[12] A. Pak, K. Marsh, S. Martins, W. Lu, W. Mori, and C. Joshi, "Injection and trapping of tunnel-ionized electrons into laser-produced wakes," *Phys. Rev. Lett.*, vol. 104, no. 2, p. 025003, 2010. [Online]. Available: https://doi.org/10.1103/PhysRevLett.104.025003

[13] L.-L. Yu *et al.*, "Two-color laser-ionization injection," *Phys. Rev. Lett.*, vol. 112, no. 12, p. 125001, 2014. [Online]. Available: https://doi.org/10.1103/PhysRevLett.112.125001

[14] C. Thaury *et al.*, "Shock assisted ionization injection in laser-plasma accelerators," *Sci. Rep.*, vol. 5, no. 1, pp. 1–7, 2015. [Online]. Available: https://doi.org/10.1038/srep16310

[15] J. Couperus *et al.*, "Demonstration of a beam loaded nanocoulomb-class laser wakefield accelerator," *Nat. Commun.*, vol. 8, no. 1, pp. 1–7, 2017. [Online]. Available: https://doi.org/10.1038/s41467-017-00592-7

[16] B. Dromey, S. Kar, M. Zepf, and P. Foster, "The plasma mirror—a subpicosecond optical switch for ultrahigh power lasers," *Rev. Sci. Instrum*, vol. 75, no. 3, pp. 645–649, 2004. [Online]. Available: https://doi.org/10.1063/1.1646737

[17] G. Doumy *et al.*, "Complete characterization of a plasma mirror for the production of high-contrast ultraintense laser pulses," *Phys. Rev. E*, vol. 69, no. 2, p. 026402, 2004. [Online]. Available: https://doi.org/10.1103/PhysRevE.69.026402

[18] C. Thaury *et al.*, "Plasma mirrors for ultrahigh-intensity optics," *Nat. Phys.*, vol. 3, no. 6, pp. 424–429, 2007. [Online]. Available: https://doi.org/10.1038/nphys595

[19] M. Thévenet *et al.*, "Vacuum laser acceleration of relativistic electrons using plasma mirror injectors," *Nat. Phys.*, vol. 12, no. 4, pp. 355–360, 2016. [Online]. Available: https://doi.org/10.1038/nphys3597

[20] L. Chopineau *et al.*, "Identification of coupling mechanisms between ultraintense laser light and dense plasmas," *Phys. Rev. X*, vol. 9, no. 1, p. 011050, 2019. [Online]. Available: https://doi.org/10.1103/PhysRevX.9.011050

[21] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation*. Boca Raton, FL: CRC press, 2004.

[22] T. D. Arber *et al.*, "Contemporary particle-in-cell approach to laser-plasma modelling," *Plasma Phys. Control. Fusion*, vol. 57, no. 11, p. 113001, sep 2015. [Online]. Available: https://doi.org/10.1088/0741-3335/57/11/113001

[23] K. Yee, "Numerical Solution Of Initial Boundary Value Problems Involving Maxwells Equations In Isotropic Media," *IEEE Trans. Antennas Propag.*, vol. Ap14, no. 3, pp. 302–307, 1966. [Online]. Available: https://doi.org/10.1109/TAP.1966.1138693

[24] J. Boris, "Relativistic Plasma Simulation-Optimization of a Hybrid Code," in *Proc. Fourth Conf. Num. Sim. Plasmas*, Naval Res. Lab., Wash., D. C., 1970, pp. 3–67.

[25] R. A. Fonseca *et al.*, "Osiris: A three-dimensional, fully relativistic particle in cell code for modeling plasma based accelerators," in *Computational Science — ICCS 2002*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 342–351.

[26] S. Bastrakov *et al.*, "Particle-in-cell plasma simulation on heterogeneous cluster systems," *J. Comput. Sci.*, vol. 3, no. 6, pp. 474–479, 2012. [Online]. Available: https://doi.org/10.1016/j.jocs.2012.08.012

[27] M. Bussmann *et al.*, "Radiative signatures of the relativistic kelvin-helmholtz instability," in *Proc. Conf. High Perform. Comput. Netw. Storage Anal.*, ser. SC '13. New York, NY, USA: ACM, 2013, pp. 5:1–5:12. [Online]. Available: http://doi.org/10.1145/2503210.2504564

[28] J. Derouillat *et al.*, "Smilei : A collaborative, open-source, multi-purpose particle-in-cell code for plasma simulation," *Comput. Phys. Commun.*, vol. 222, pp. 351–373, jan 2018. [Online]. Available: https://doi.org/10.1016/j.cpc.2017.09.024

[29] R. Bird, N. Tan, S. V. Luedtke, S. L. Harrell, M. Taufer, and B. Albright, "VPIC 2.0: next generation Particle-in-Cell simulations," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 4, 4 2022. [Online]. Available: https://doi.org/10.1109/TPDS.2021.3084795

[30] A. Myers *et al.*, "Porting WarpX to GPU-accelerated platforms," *Parallel Comput.*, vol. 108, p. 102833, 2021. [Online]. Available: https://doi.org/10.1016/j.parco.2021.102833

[31] H. Abe, N. Sakairi, R. Itatani, and H. Okuda, "High-Order Spline Interpolations In The Particle Simulation," *J. Comput. Phys.*, vol. 63, no. 2, pp. 247–267, apr 1986. [Online]. Available: https://doi.org/10.1016/0021-9991(86)90193-2

[32] M. Rowan *et al.*, "In-situ assessment of device-side compute work for dynamic load balancing in a GPU-accelerated PIC code," *Proceedings of the Platform for Advanced Scientific Computing Conference, PASC 2021*, vol. 11, jul 2021. [Online]. Available: https://doi.org/10.1145/3468267.3470614

[33] "AMReX Source and Documentation," https://amrex-codes.github.io, 2022.

[34] J.-L. Vay *et al.*, "Application of adaptive mesh refinement to particle-in-cell simulations of plasmas and beams," *Phys. Plasmas*, vol. 11, no. 5 PART 2, 2004. [Online]. Available: https://doi.org/10.1063/1.1689669

[35] J.-L. Vay, D. P. Grote, R. H. Cohen, and A. Friedman, "Novel methods in the particle-in-cell accelerator code-framework warp," *Comput. Sci. Discov.*, vol. 5, no. 1, p. 014019 (20 pp.), 2012. [Online]. Available: https://doi.org/10.1088/1749-4699/5/1/014019

[36] M. J. Berger and P. Colella, "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *J. Comput. Phys.*, vol. 82, p. 6484, 1989. [Online]. Available: https://doi.org/10.1016/0021-9991(89)90035-1

[37] B. Lo and P. Colella, "An adaptive local discrete convolution method for the numerical solution of Maxwell's equations," *Comm. App. Math. Comp. Sci.*, vol. 14, no. 1, pp. 105–119, apr 2019. [Online]. Available: https://doi.org/10.2140/camcos.2019.14.105

[38] J.-P. Berenger, "A Perfectly Matched Layer For The Absorption Of Electromagnetic-Waves," *J. Comput. Phys.*, vol. 114, no. 2, pp. 185–200, oct 1994. [Online]. Available: https://doi.org/10.1006/jcph.1994.1159

[39] J. L. Vay *et al.*, "Toward plasma wakefield simulations at exascale," in *2018 IEEE Advanced Accelerator Concepts Workshop, ACC 2018 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., mar 2019. [Online]. Available: https://doi.org/10.1109/AAC.2018.8659392

[40] "Data artifact: https://doi.org/10.5281/zenodo.6464240."

[41] TOP500 website. (Accessed Aug. 25, 2022). [Online]. Available: https://top500.org/

[42] Specifications - Supercomputer Fugaku : Fujitsu Global. (Accessed Aug. 25, 2022). [Online]. Available: https://www.fujitsu.com/global/about/innovation/fugaku/specifications/

[43] Nvidia TESLA V100 SXM2 16 GB. (Accessed Aug. 25, 2022). [Online]. Available: https://images.nvidia.com/content/technologies/volta/pdf/437317-Volta-V100-DS-NV-US-WEB.pdf

[44] Nvidia A100 40GB. (Accessed Aug. 25, 2022). [Online]. Available: https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet.pdf

[45] AMD MI250X 128GB. (Accessed Aug. 25, 2022). [Online]. Available: https://www.amd.com/en/products/server-accelerators/instinct-mi250x

[46] J. Dongarra, M. Heroux, and P. Luszczek, "High-performance conjugate-gradient benchmark: A new metric for ranking high-performance computing systems," *Int. J. High Perform. Comput. Appl.*, vol. 30, no. 1, pp. 3–10, 2016. [Online]. Available: https://doi.org/10.1177/1094342015593158

[47] HPCG List 11/2021. (Accessed Aug. 25, 2022). [Online]. Available: https://top500.org/lists/hpcg/2021/11/

[48] K. J. Bowers, B. J. Albright, B. Bergen *et al.*, "0.374 Pflop/s trillion-particle kinetic modeling of laser plasma interaction on roadrunner," in *SC '08: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, 2008, pp. 1–11. [Online]. Available: https://doi.org/10.1109/SC.2008.5222734

[49] E. Zenker *et al.*, "Performance-Portable Many-Core Plasma Simulations: Porting PIConGPU to OpenPower and Beyond," in *High Performance Computing*. Cham: Springer International Publishing, 2016, pp. 293–301. [Online]. Available: https://doi.org/10.1007/978-3-319-46079-6_21

[50] J.-L. Vay, "Noninvariance Of Space- And Time-Scale Ranges Under A Lorentz Transformation And The Implications For The Study Of Relativistic Interactions," *Phys. Rev. Lett.*, vol. 98, no. 13, pp. 130 405/1–4, 2007. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.98.130405

[51] R. Lehe *et al.*, "Elimination of numerical Cherenkov instability in flowing-plasma particle-in-cell simulations by using Galilean coordinates," *Phys. Rev. E*, vol. 94, no. 5, 2016. [Online]. Available: https://doi.org//10.1103/PhysRevE.94.053305

[52] O. Shapoval *et al.*, "Overcoming timestep limitations in boosted-frame particle-in-cell simulations of plasma-based acceleration," *Phys. Rev. E*, vol. 104, no. 5, p. 055311, nov 2021. [Online]. Available: https://doi.org/10.1103/PhysRevE.104.055311