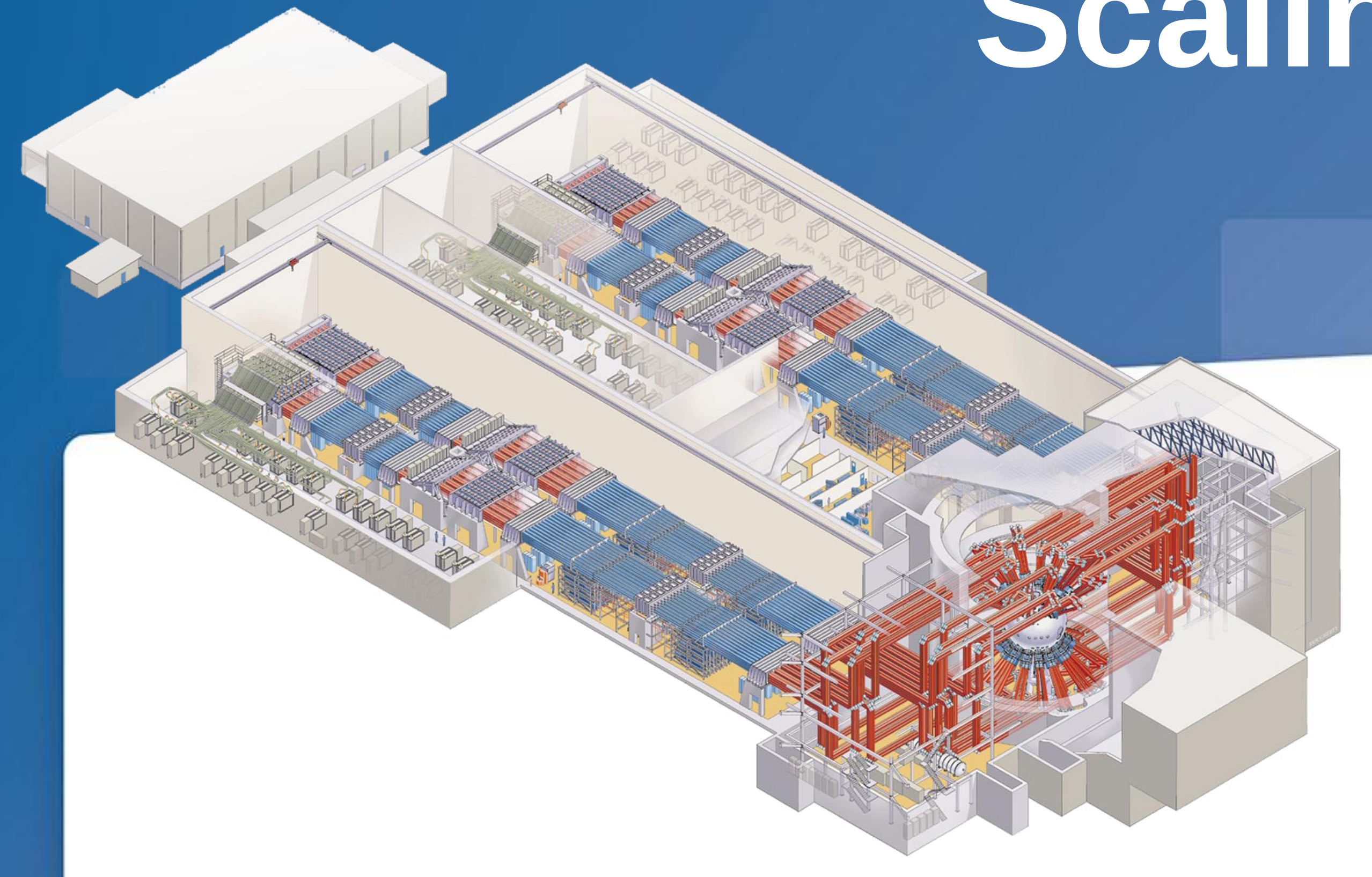
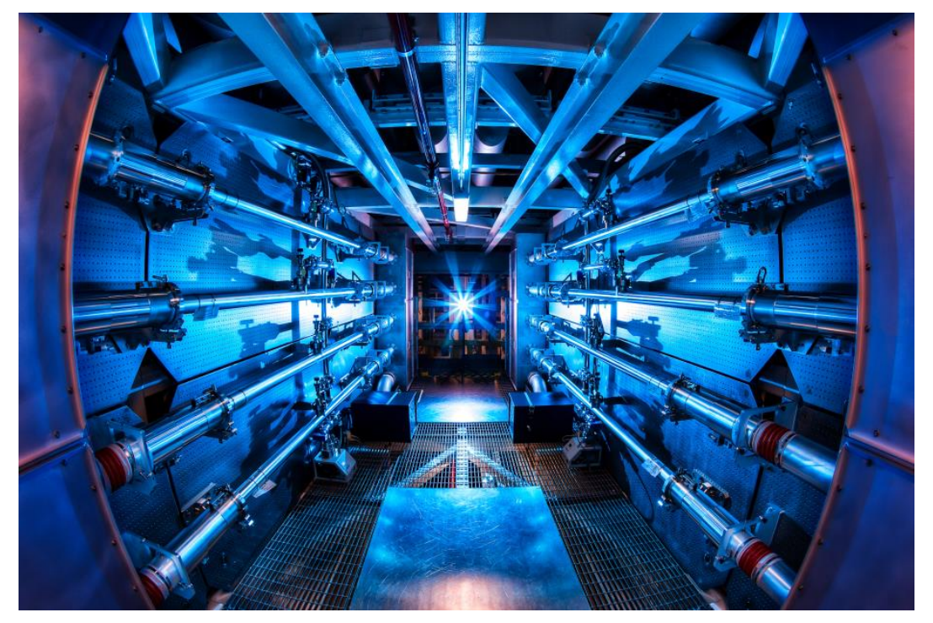
# Scaling a high energy laser application (VBL) using MPI and RAJA

Kathleen McCandless, Tom Epperly, Jean Michel Di Nicola, Katie Lewis,
Gabriel Mennerat, Jarom Nelson, Samuel Schrauth, Paul Wegner

National Ignition Facility, Lawrence Livermore National Laboratory

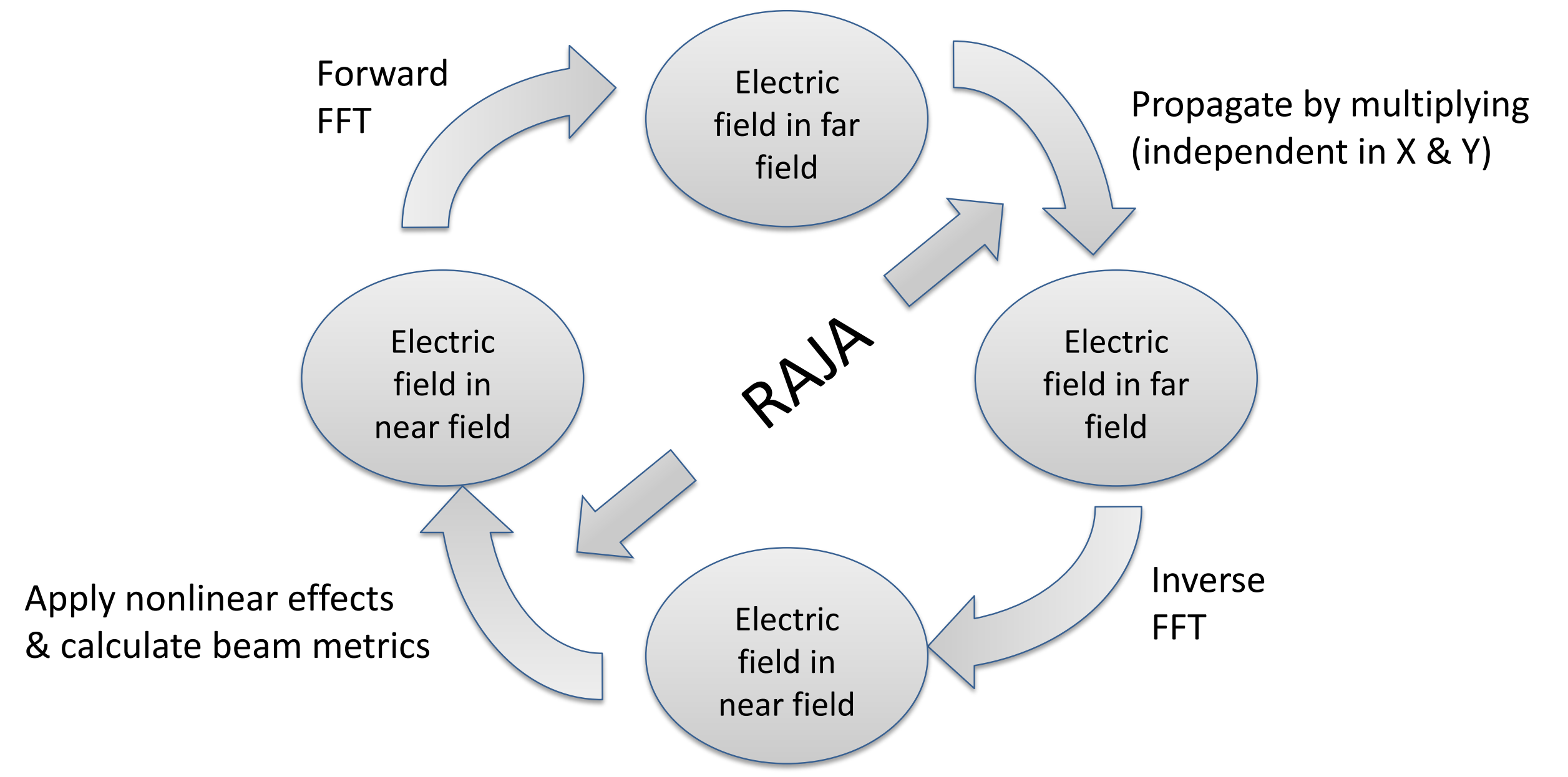




## Abstract

LLNL is a world leader in designing and maintaining high energy lasers, built upon decades of leadership in the modeling of high energy laser systems. Here we present initial results for a parallel mini-app based on the National Ignition Facility's (NIF) Virtual Beamline (VBL)[1,2] code, a single-node laser physics modeling engine. Recent advances in ultra-intense short-pulse laser systems are driving us to develop massively parallel laser physics capabilities similar to the laser physics code Miró[3] (an MPI-only implementation) to support the multi-order increase in time/space resolution needed for these types of broadband, chirped-pulse amplification lasers. Here we present a demonstration of our new scalable simulation code architecture using MPI and the RAJA Portability Layer[4]. This hybrid parallelization approach promises to bridge the gap in resolution allowing us to deliver future simulations with the requisite physics fidelity at an unprecedented scale.

## Split-step Algorithm Overview



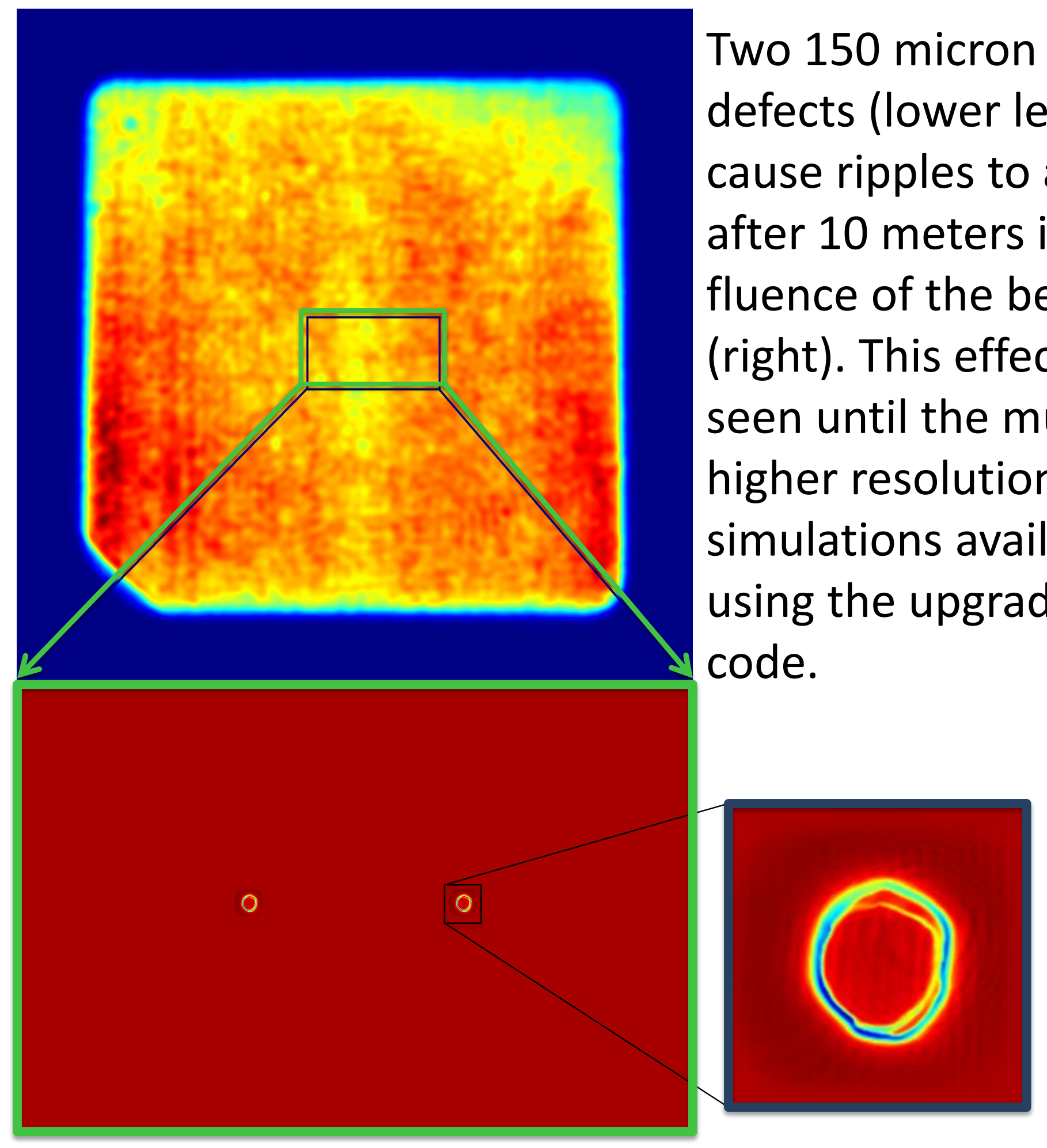
### RAJA Diffractive Propagation Loop

```
const std::int32_t numX = this->nx( );
const std::int32_t numY = this->ny( );
const Real::BaseType scale = d_FField.nearFieldScaleFactor( );
const Real::BaseType dx = vbl::TWO_PI * kx_max / numX;
const Real::BaseType dy = vbl::TWO_PI * ky_max / numY;

lfieldx->template forallN< vbl::fine >([=]VBL_DEVICE( vbl::TimeInd t, vbl::YInd y, vbl::XInd x )
{
   const Real::BaseType ky = spatialFrequency( y_global_off + *y, numY, dy );
   const Real::BaseType kx = spatialFrequency( x_global_off + *x, numX, dx );
   lfieldx->value( t, y, x ) *= ( scale * exp( - leadingConstant * ( kx * kx + ky * ky ) * dz) );
});
```

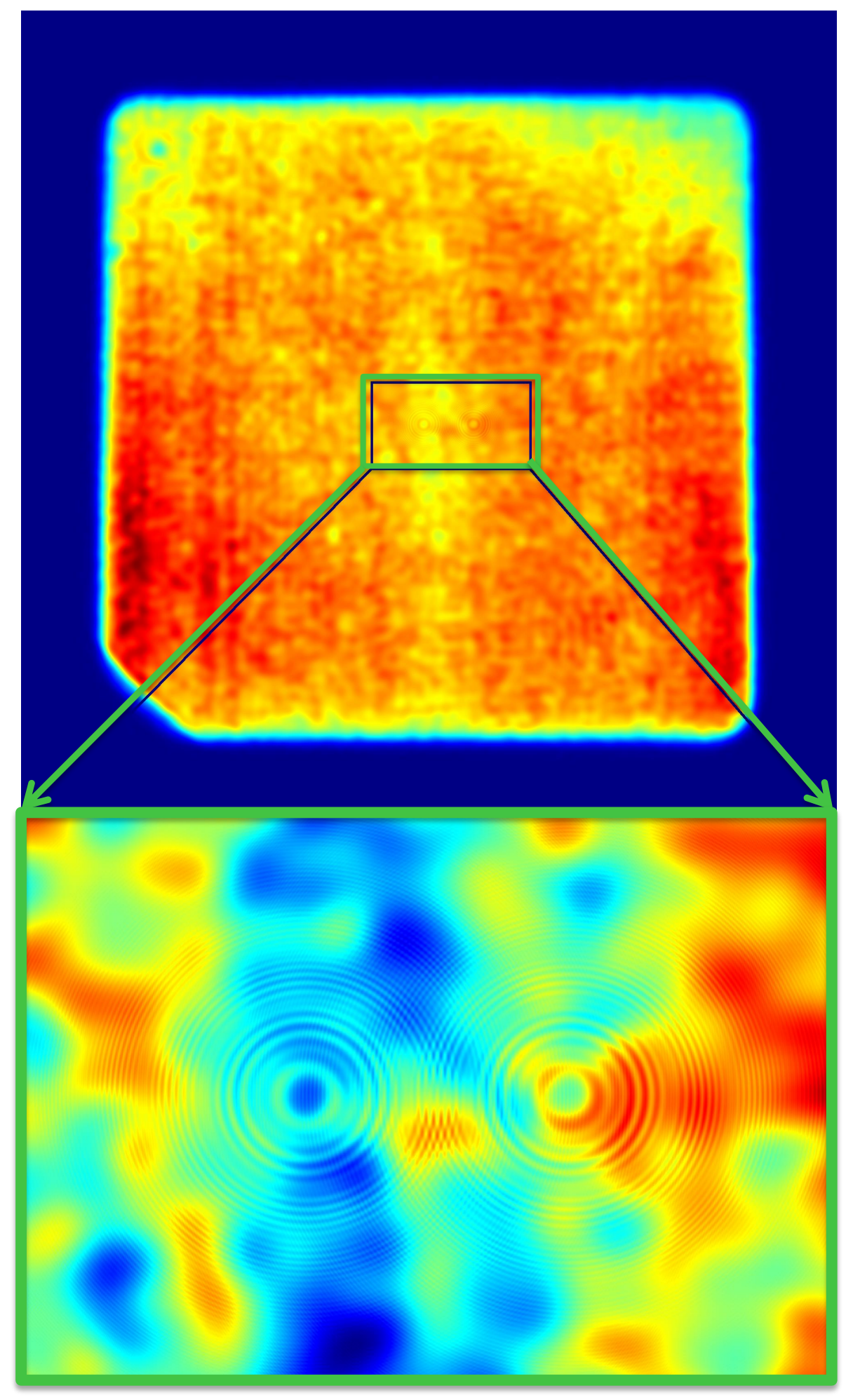
### RAJA Apply Nonlinear Effects Loop

```
const Real::Aligned nfScaleFactor = d_FField.nearFieldScaleFactor( );
const Real::BaseType nonlinearphase = gamma * vbl::TWO_PI * dz / d_wavelength;

lfieldx->template forallN< vbl::fine > ([=]VBL_DEVICE( vbl::TimeInd t, vbl::YInd y, vbl::XInd x )
{
  const Complex::Aligned fieldValue( nfScaleFactor * lfieldx->value( t,y,x ) );
  const Real::Aligned selfPhaseModulation = nonlinearphase * COMPLEX_NS::norm( fieldValue );
  const Complex::Aligned operand( cos( selfPhaseModulation ), sin( selfPhaseModulation ) );
      lfieldx->value( t, y, x ) = fieldValue * operand;
});
```

## New high resolution simulation results



Two 150 micron phase defects (lower left), cause ripples to appear after 10 meters in the fluence of the beam (right). This effect is not seen until the much higher resolution simulations available using the upgraded code.



### Example RAJA Policies

**Single Core Policy**

```
typedef RAJA::NestedPolicy<
                RAJA::ExecList< RAJA::seq_exec,
                                RAJA::seq_exec,
                                RAJA::simd_exec > > fine;
```

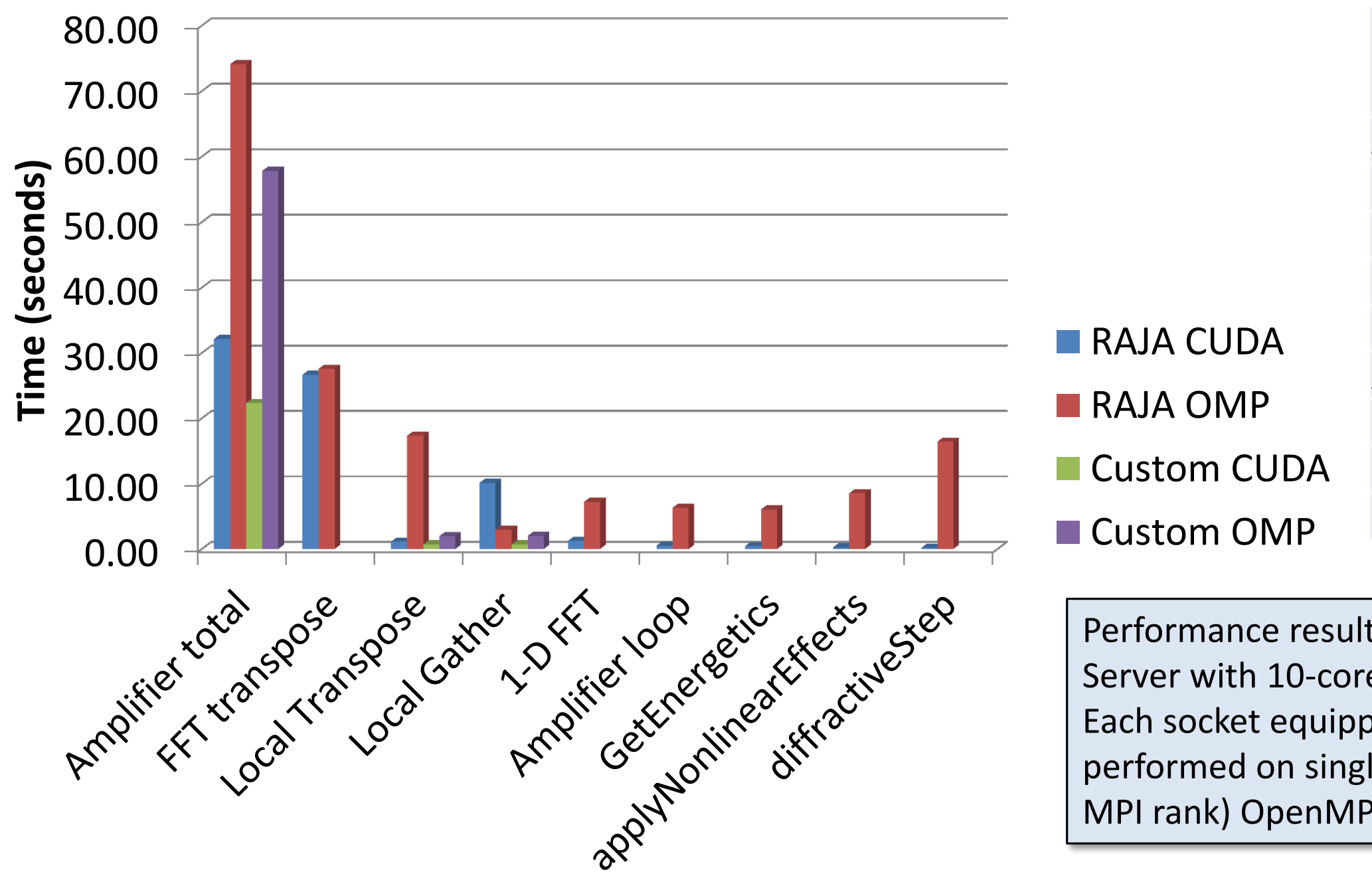
**OpenMP**

```
typedef RAJA::NestedPolicy<
                RAJA::ExecList< RAJA::omp_collapse_nowait_exec,
                                RAJA::omp_collapse_nowait_exec,
                                RAJA::simd_exec  >,
                                RAJA::OMP_Parallel< > > fine;
```

**CUDA**

```
typedef RAJA::NestedPolicy<
                RAJA::ExecList< RAJA::cuda_block_z_exec,
                                RAJA::cuda_threadblock_y_exec< 16 >,
                                RAJA::cuda_threadblock_x_exec< 8 > > > fine;
```

## OpenMP & CUDA performance

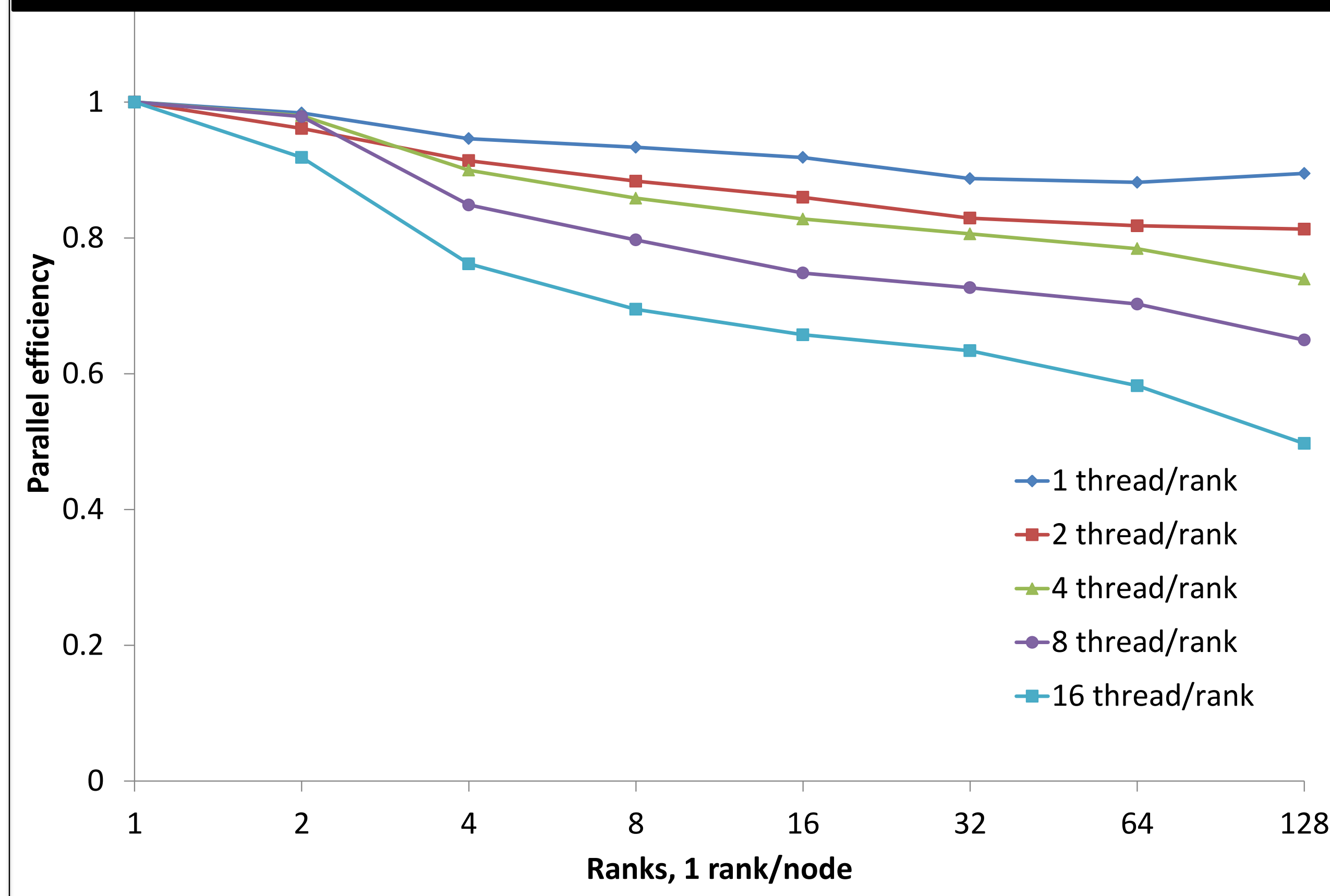


| Time in seconds | RAJA CUDA | RAJA OMP | Custom CUDA | Custom OMP |
|---|---|---|---|---|
| Amplifier total | 32.14 | 74.09 | 22.32 | 57.81 |
| FFT transpose | 26.65 | 27.56 | | |
| Local Transpose | 1.11 | 17.32 | 0.70 | 1.96 |
| Local Gather | 10.12 | 2.94 | 0.70 | 2.01 |
| 1-D FFT | 1.25 | 7.21 | | |
| Amplifier loop | 0.50 | 6.30 | | |
| GetEnergetics | 0.43 | 6.05 | | |
| applyNonlinearEffects | 0.27 | 8.52 | | |
| diffractiveStep | 0.18 | 16.41 | | |

Performance results from IBM Power System S822LC Dual-socket POWER8 Server with 10-core processors 8-way SMT per core running at 3.7GHz. Each socket equipped with one NVIDIA Tesla K80 (two K40) GPU. All tests performed on single node with 4 MPI ranks GPU results use 4 GPUs (one per MPI rank) OpenMP results use 4 threads per core. Results courtesy of IBM.

## Scaling Results

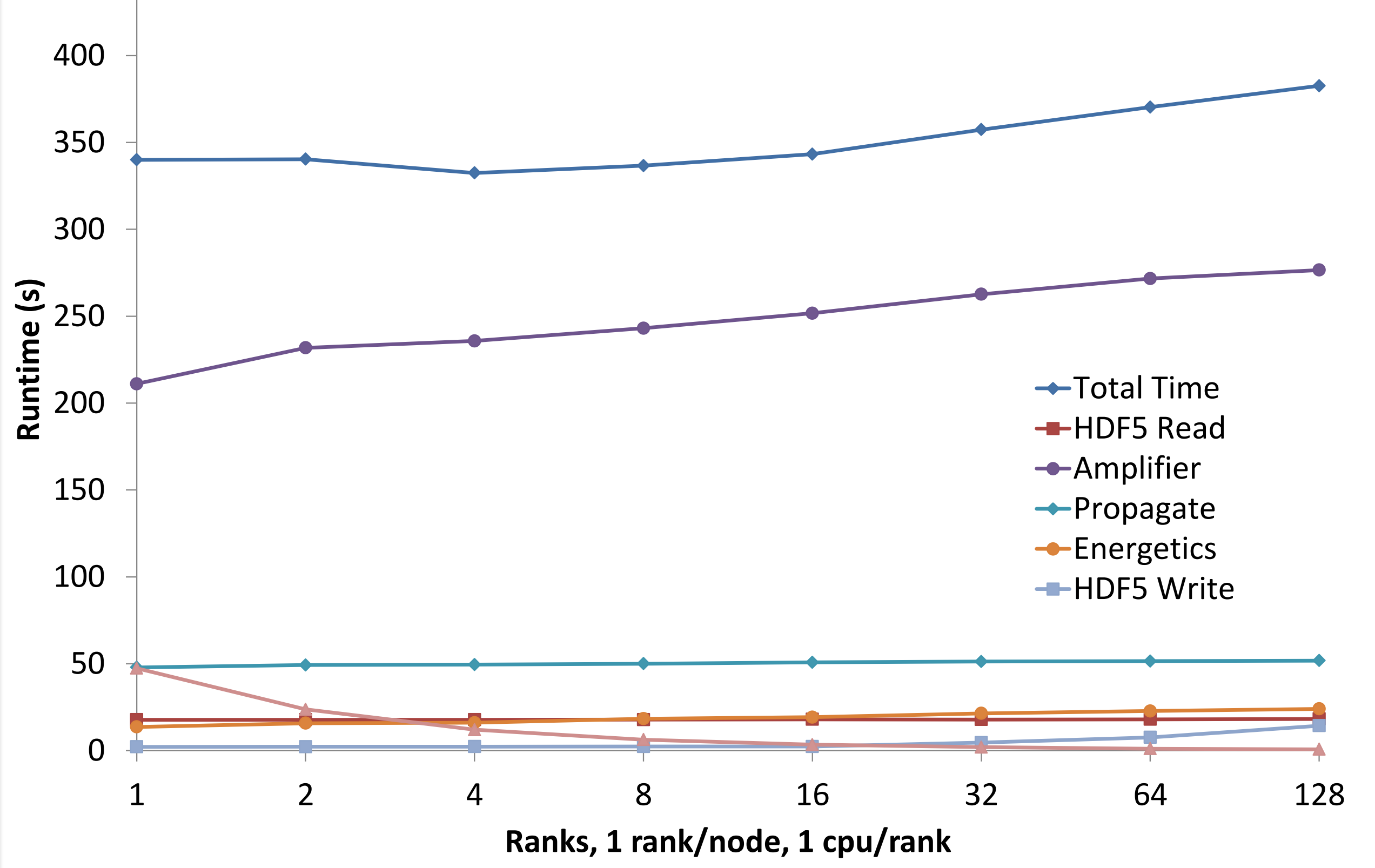
### Strong scaling parallel efficiency



Increasing ranks or thread/rank gives speedup in all cases. With one thread (MPI only), PE is at or above 90%. Speedup from 1 rank, 1 thread to 128 ranks, 16 threads is over 200x.

Performance results from Syrah
2.6 GHz Intel Xeon E5-2670
16 cores/node
InifiniBand QDR (QLogic)
$1024^3$ fixed global problem size

### Weak scaling sub-step breakdown



Split-step amplifier (FFT) is limiting factor in scaling. HDF output starts to become a problem with increasing ranks. Amplifier setup speedup is due to fixed size of amplifier with increasing ranks.

Performance results from Syrah
2.6 GHz Intel Xeon E5-2670
16 cores/node
InifiniBand QDR (QLogic)
$8192^3$ problem size per rank

## Future Work & Conclusions

We converted our 'mini-app' from an MPI-only application to a hybrid application using the RAJA portability framework which provides a common interface to heterogeneous compute resources. With a minimal code footprint, we are able to use RAJA to express traversals over the spatio-temporal grid.

Thanks to the RAJA team members, Jeff Keasler and Richard Hornung. Also thanks to Todd Gamblin for help with the Atlassian tool suite. Additional thanks to Xing Liu and Bob Walkup at IBM for assistance with the RAJA/CUDA results and algorithm improvements. Finally, thanks to the staff and machinery at the Livermore Computing Center. This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

## References

[1] R. A. Sacks ; K. P. McCandless ; E. Feigenbaum ; J. M. G. Di Nicola ; K. J. Luke, et al. "The virtual beamline (VBL) laser simulation code", *Proc. SPIE* 9345, High Power Lasers for Fusion Research III, 93450M (Feb 2015)
[2] M. L. Spaeth, et al. "Description of the NIF Laser", Fusion Science and Technology, Volume 69, 25-145 (Jan/Feb 2016)
[3] Morice O; "Miro´: complete modeling and software for pulse amplification and propagation in high-power laser systems", *Opt. Eng.* 0001;42(6):1530-1541 *(Miró is a laser physics code developed by CEA in France)*
[4] J.M. Di Nicola, et al. "The commissioning of the advanced radiographic capability laser system: experimental and modeling results at the main laser output", *Proc SPIE* 9345, High Power Lasers For Fusion Research III, 93450I (Feb 2015)
[5]Hornung, R., Keasler, J.: The RAJA portability layer: overview and status. Technical report, Lawrence Livermore National Laboratory (LLNL), Livermore, CA (2014) https://github.com/LLNL/RAJA
[6] Frantz, L.M., Nodvik, J.S.: Theory of pulse propagation in a laser amplifier. J. Appl. Phys. 34, 2346–2349 (1963)

National Ignition Facility • Lawrence Livermore National Laboratory • Operated by the U.S. Department of Energy
This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

LLNL-POST-704471