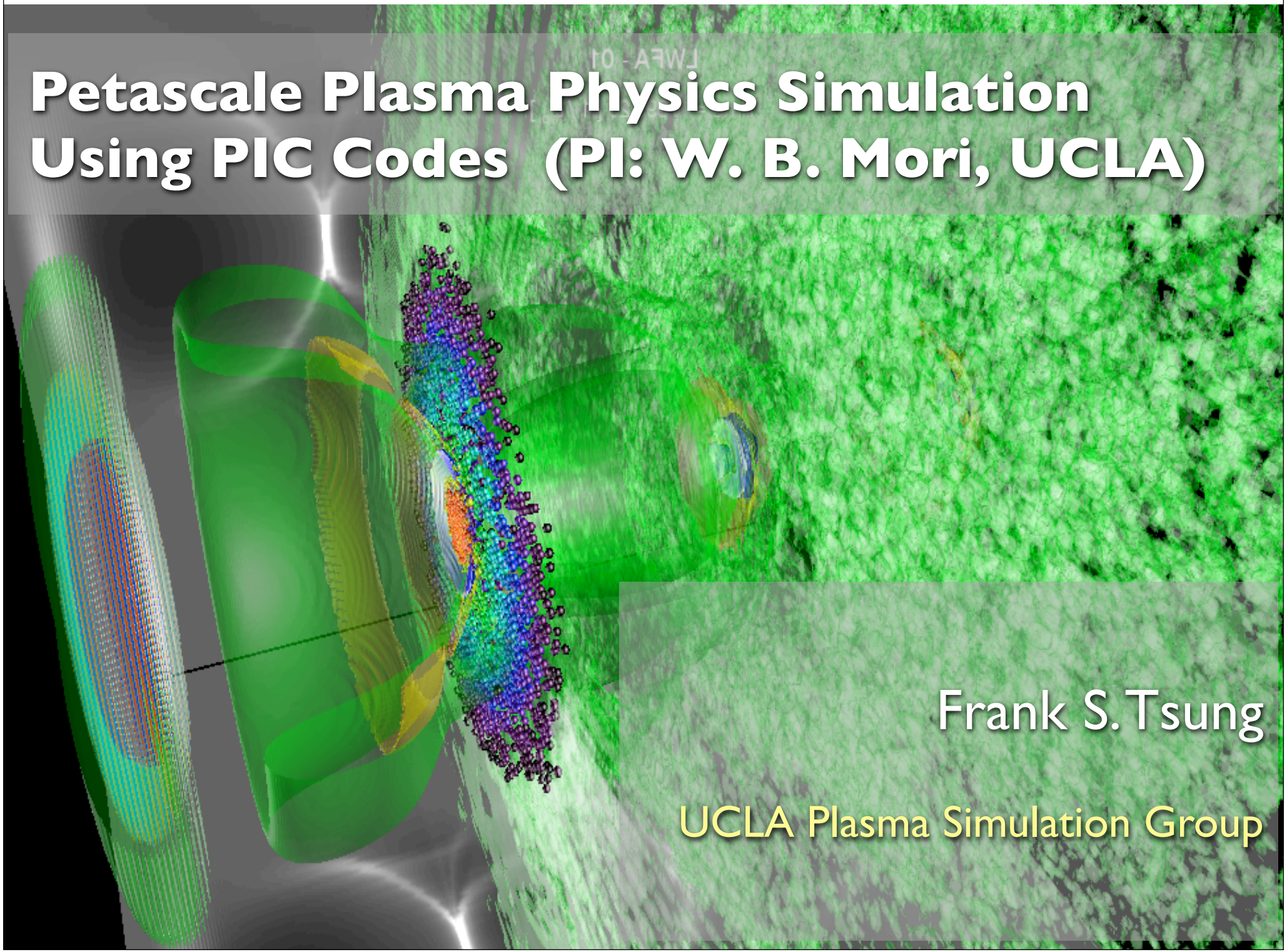
Petascale Plasma Physics Simulation Using PIC Codes (PI: W. B. Mori, UCLA)



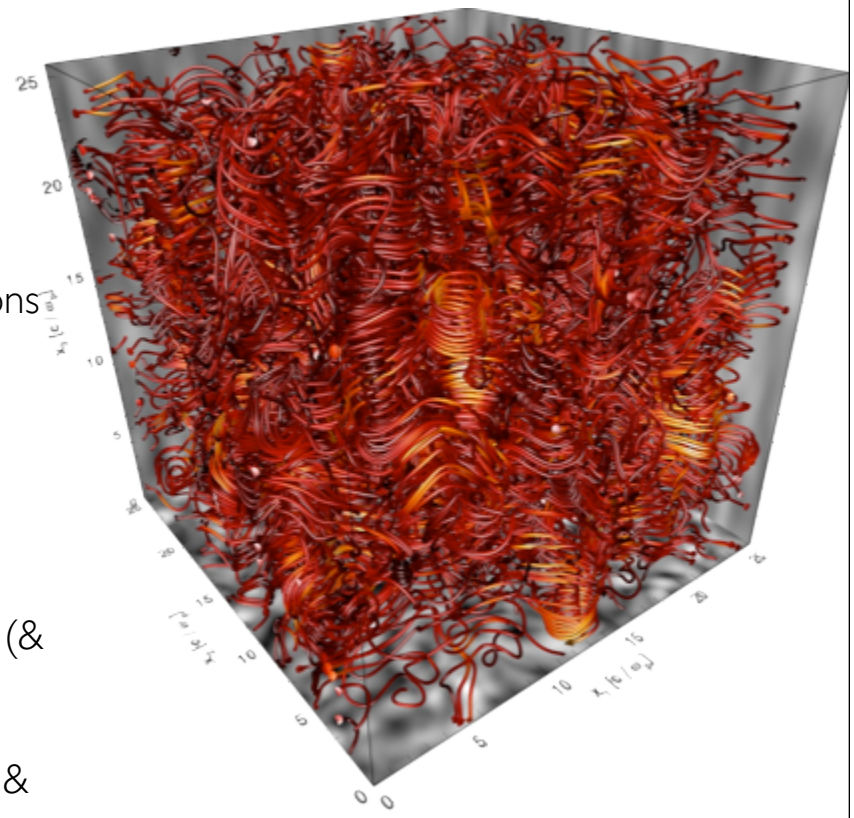
Frank S. Tsung

UCLA Plasma Simulation Group

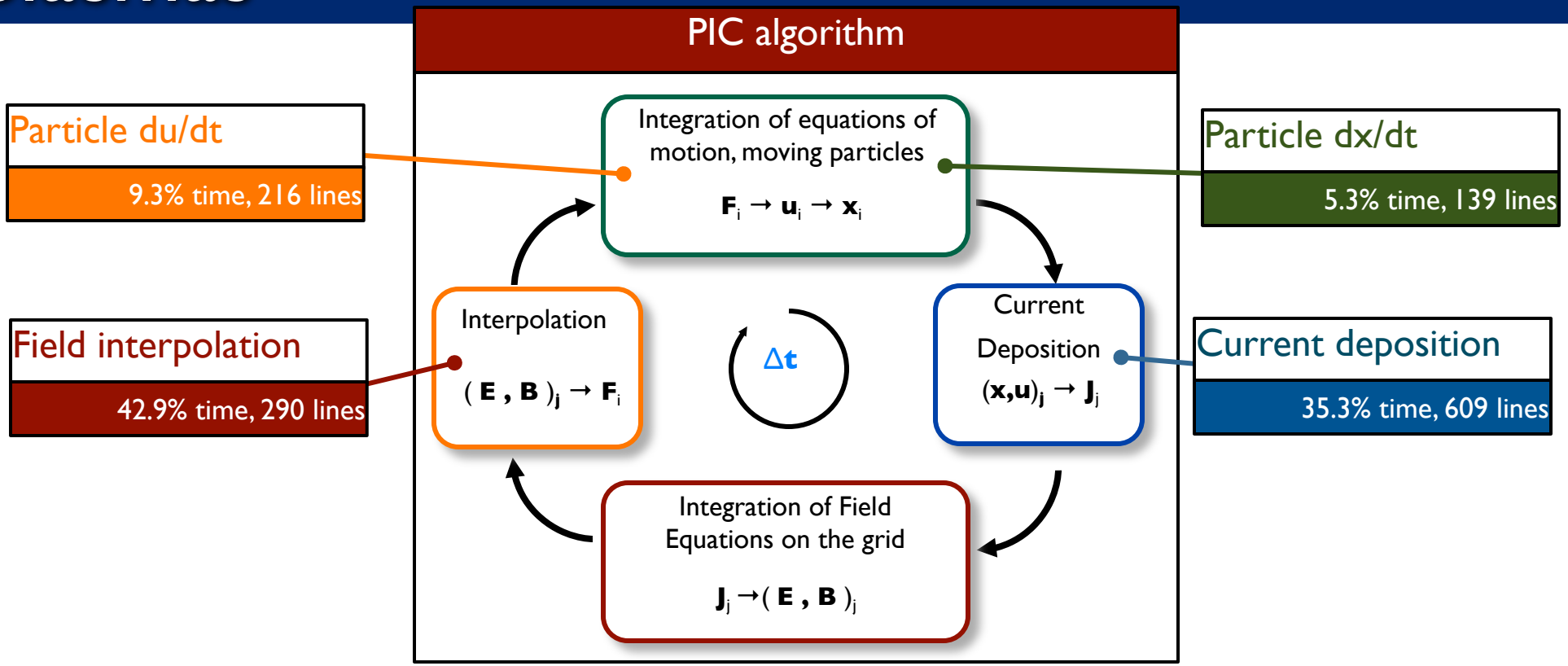
Summary and Outline

OUTLINE

- Overview of the project
  - Particle-in-cell codes
  - OSIRIS and recent >2PFlops benchmark on Blue Waters
- Application of OSIRIS to plasma based accelerators:
  - full scale, one-to-one 3D simulations of plasma based accelerators and quantitative agreements between simulations and experiments.
- Applications of OSIRIS to LPI's Relevant to IFE
  - SRS in indirect drive IFE targets.
  - SRS and TPD for direct drive and shock ignition relevant conditions.
  - Estimates of large (full) scale LPI simulations in 2D and 3D (& justify the need for much larger supercomputers)
- Development works for Blue Waters and beyond --- our GPU & GPU + MPI PIC code



Profile of OSIRIS/Introduction to PIC for plasmas



- The particle-in-cell method treats plasma as a collection of computer particles. The interactions does not scale as N^2 due to the fact the particle quantities are deposited on a grid and the fields are solved on these grids only. Because (# of particles) >> (# of grids), the timing is dominated by the orbit calculations
- The code spends over 90 % of execution time in only 4 routines
- These routines correspond to less than 2 % of the code, so optimization is fairly straight forward

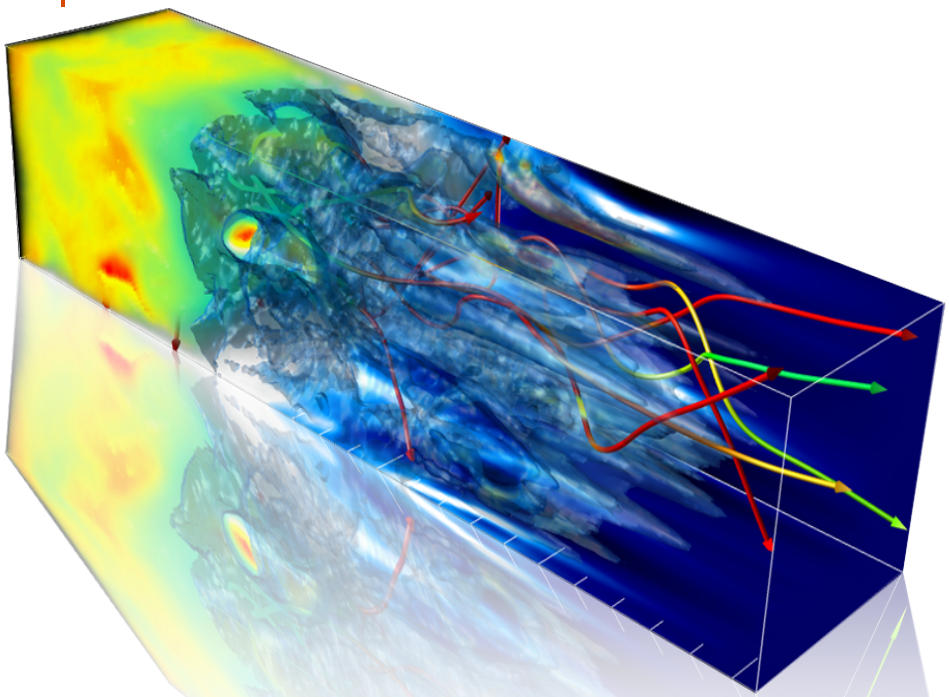
osiris 2.0

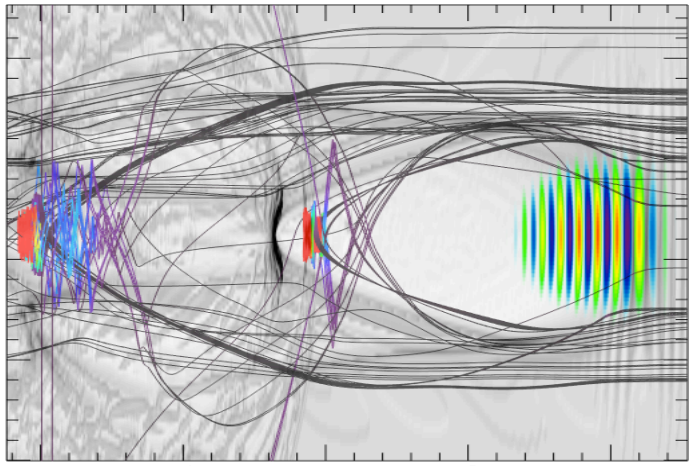
osiris v2.0

osiris framework

- Massively Parallel, Fully Relativistic Particle-in-Cell (PIC) Code
- Visualization and Data Analysis Infrastructure
- Developed by the osiris.consortium

⇒ UCLA + IST





New Features in v2.0

- Bessel Beams
- Binary Collision Module (to study plasmas which behave more like fluids)
- **Energy Conserving Algorithm**
- **Multi-dimensional Dynamic Load Balancing**
- **OpenMP/MPI hybrid parallelism**
- PML absorbing BC
- **Higher order splines**
- Parallel I/O (HDF5)
- Boosted frame in 1/2/3D

Ricardo Fonseca: ricardo.fonseca@ist.utl.pt
Frank Tsung: tsung@physics.ucla.edu

http://cfp.ist.utl.pt/golp/epp/
http://exodus.physics.ucla.edu/

On Blue Waters OSIRIS recently achieved >2 Pflops on > 750,000 CPU cores.

OSIRIS is a very mature code and is optimized on a variety of platforms, it use the Intel/AMD SSE instruction set which can carry out multiple FP operations per cycle. We've recently performed timing studies on Blue Waters using this version of OSIRIS. Using this version, we achieved sustained 2.2PFlops

Simulation Parameters:

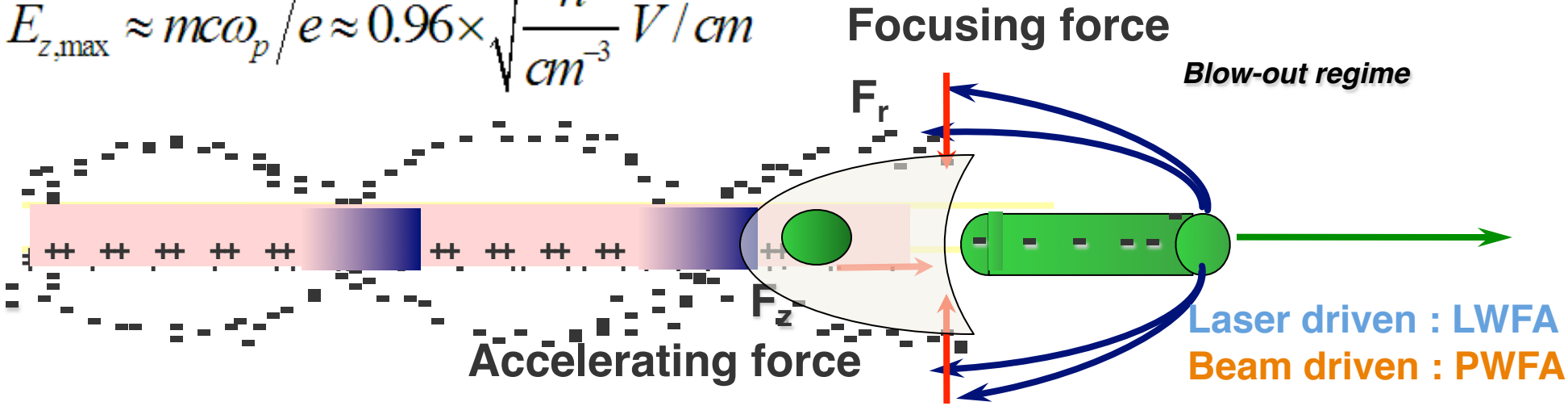
- Using latest OSIRIS with SSE SIMD instructions
- ~25 billion cells (38624 x 1024 x 640), 772,480 cores
- 10 trillion particles (400 particles/cell), quadratic spline (~1,000 floating point operations per particle per step)
- In 2154 seconds, the code ran for 437 steps (4.4 quadrillion "particle steps", and performed 4.78 exa-FLOPs) (>90% of the time is spent on orbit calculations), the code sustained ~2.2 Pflops over this period.
- Thermal plasma, very balanced benchmark simulation, a real 3D science simulation is being designed right now which should easily reach 70-80% of the results shown here.

Iterations = 437

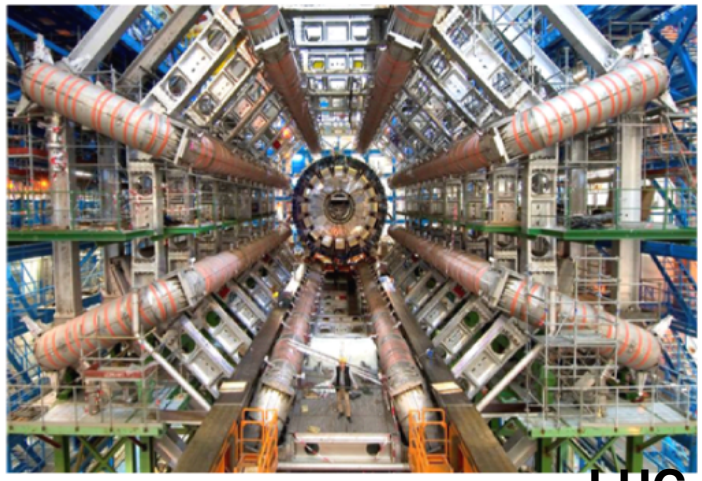
Event	real cycles	real usec	user cycles	user usec	TOT_INS	TOT_FP	L2_DCM
loop	4954593207843	2154170955	4954798000000	2154260000	4449812271244012886	4786273632293261082	204326910869145
dynamic load balance (total)	0	0	0	0	0	0	0
.....							
update emf boundary	154917350426	67355613	154721000000	67270000	29765232664499274	0	243341635909
EMF diagnostics	76837641590	33407591	74313000000	32310000	110322817116961	569557228930	3975881539
field solver	408381740	177567	805000000	350000	232002388284043	121928819470080	95130060633
field smooth	0	0	0	0	0	0	0
psi calculation	0	0	0	0	0	0	0
electric current diagnostics	69102262	30041	115000000	50000	162563793935	0	372873121
current smooth	67669394	29434	115000000	50000	402273823248	0	1099013532
update current boundary	156530320230	68057009	156538000000	68060000	35132905334305375	3836846361600	446063370870
advance deposit	4737931907733	2059971330	4738092000000	2060040000	4263093646158535748	4785236041863962481	96448370972603
reduce current	0	0	0	0	0	0	0
.....							
particle sort (total)	184297555705	80129399	184207000000	80090000	14290027249867813	0	60385010318484
particle sort, gen. idx	60531859839	26318346	59846000000	26020000	4648944915971532	0	1489627450363
particle sort, rearrange particles	171072965685	74379603	171074000000	74380000	9640978044909390	0	58894151509372

Plasma-based accelerators and applications

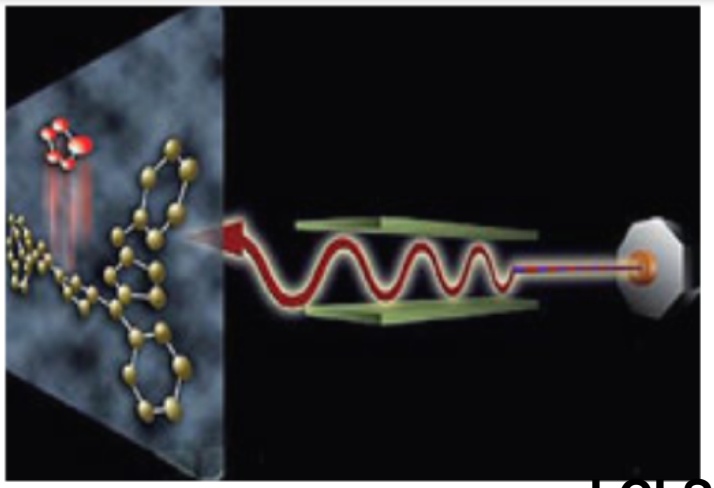
$$E_{z,\max} \approx mc\omega_p / e \approx 0.96 \times \sqrt{\frac{n}{cm^{-3}}}\, V/cm$$



TeV e^-/e^+ beams for High energy physics



10s GeV e^- beams for Photon science (e.g., SLAC LCLS (Linac Coherent Light Source) -- plasma will allow for table-top light sources in the near future



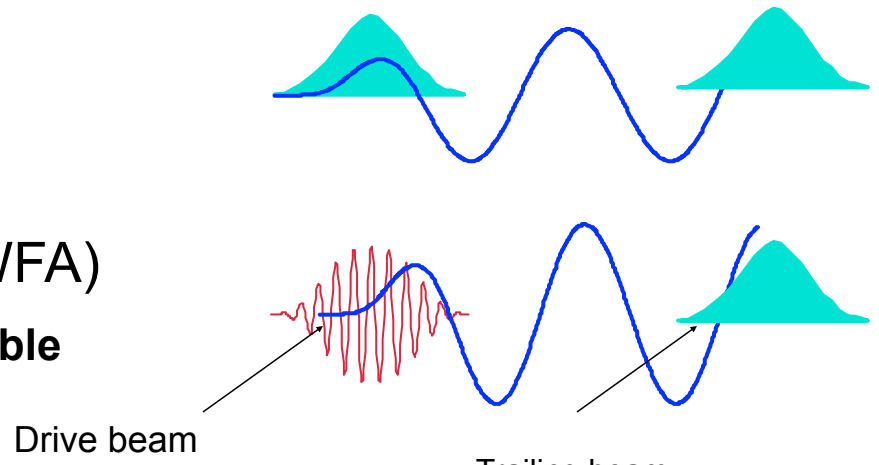
Livingston Curve for Accelerators --- Why plasmas?

Plasma Wake Field Accelerator(PWFA)

A high energy electron bunch

Laser Wake Field Accelerator(LWFA, SMLWFA)

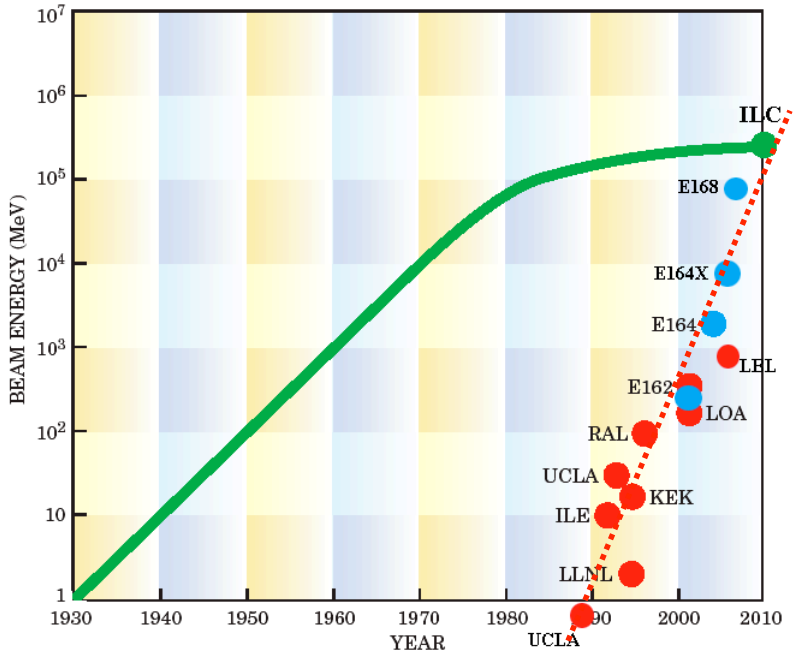
A single short-pulse of photons -- wasn't available in 1979



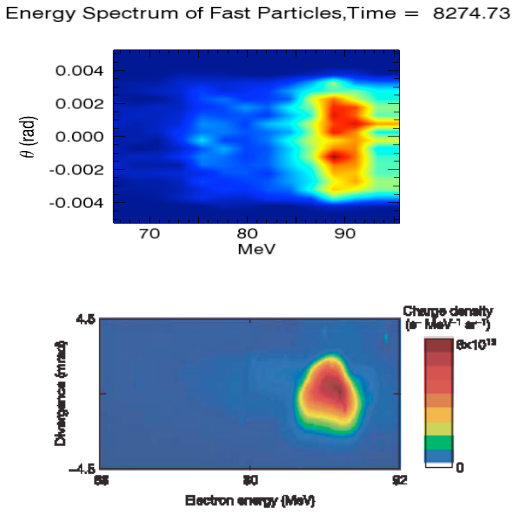
The Livingston curve traces the history of electron accelerators from Lawrence's cyclotron to present day technology.

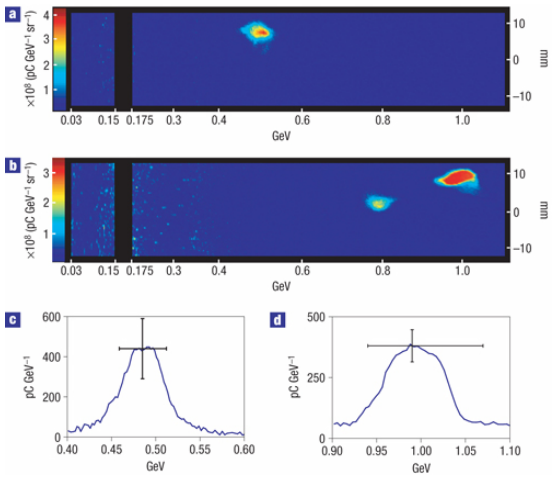
Conventional accelerator uses metals which has a upper limit for fields due to metal breakdown (i.e., sparking). Since 1980, the only way to increase energy is to build bigger accelerators. Plasma is pre-ionized and do not have this problem.

When energies from plasma based accelerators are plotted in the same curve, it shows the exciting trend that within a few years it is will surpass conventional accelerators in terms of energy.



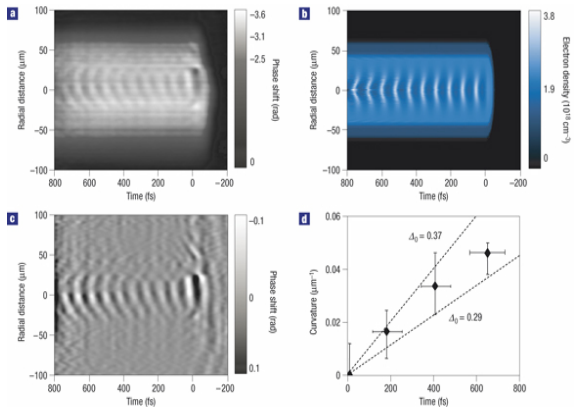
Recent Highlights in Plasma Based Acceleration (< Last 5 years) -- Simulations play a big role in all of these discoveries!!!



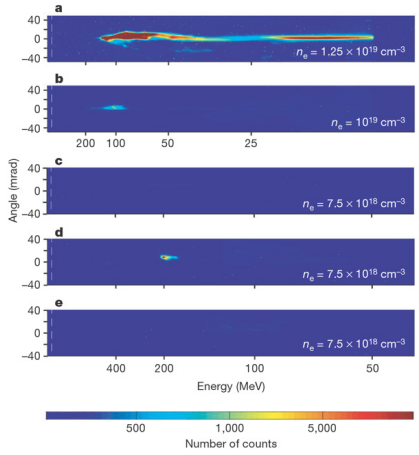


"Dream Beam" (Nature, 2004) -- 3 groups observed monoenergetic bunches using short (< 100fs) pulse lasers -- 3D simulations produced qnantitative agreements!!

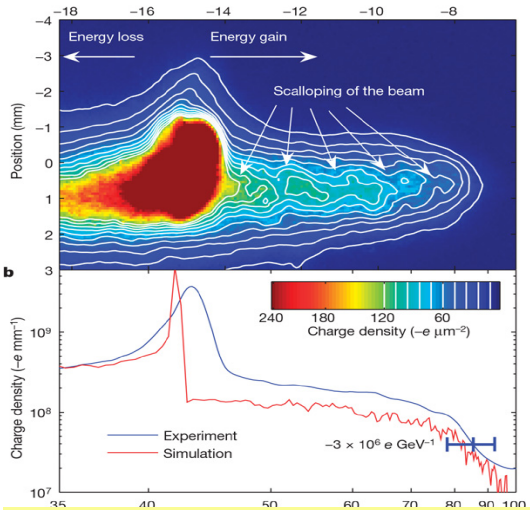
GeV LWFA in cm scale plasma



Laboratory snap shot of wakefield (using "laser holography")



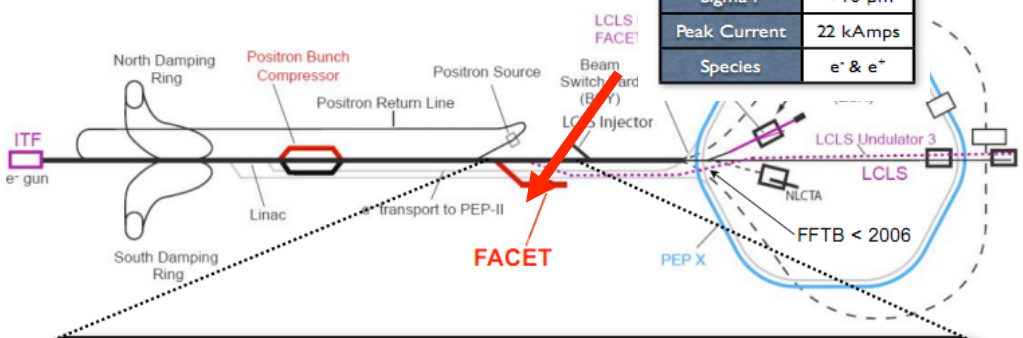
Controlled electron injection

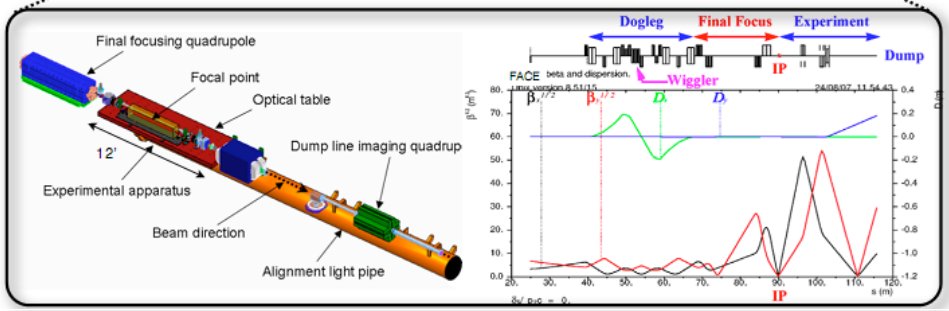


42 GeV in a PWFA in less than one meter! (i.e., 0-42 GeV in 3km, 42-85 GeV in 1m) Simulations also identified ionization induced erosion as the limiting mechanism

Currently OSIRIS is used to study several Experiments, including FACET ---- Facilities for ACcelerator science and Experimental Test Beams

Energy	24 GeV
Charge	3 nC
Sigma z	17 μm
Sigma r	< 10 μm
Peak Current	22 kAmps
Species	e^- & e^+





FACET is a new facility to provide high-energy, high peak current e^- & e^+ beams for PWFA experiments at SLAC, the goal is to achieve high efficiency, with low energy spread and low emittance. (In 2006 this facility demonstrates energy doubling in 1 meter using a long beam) Particle-in-cell simulations played a big role in understanding the detailed physics in that experiment.

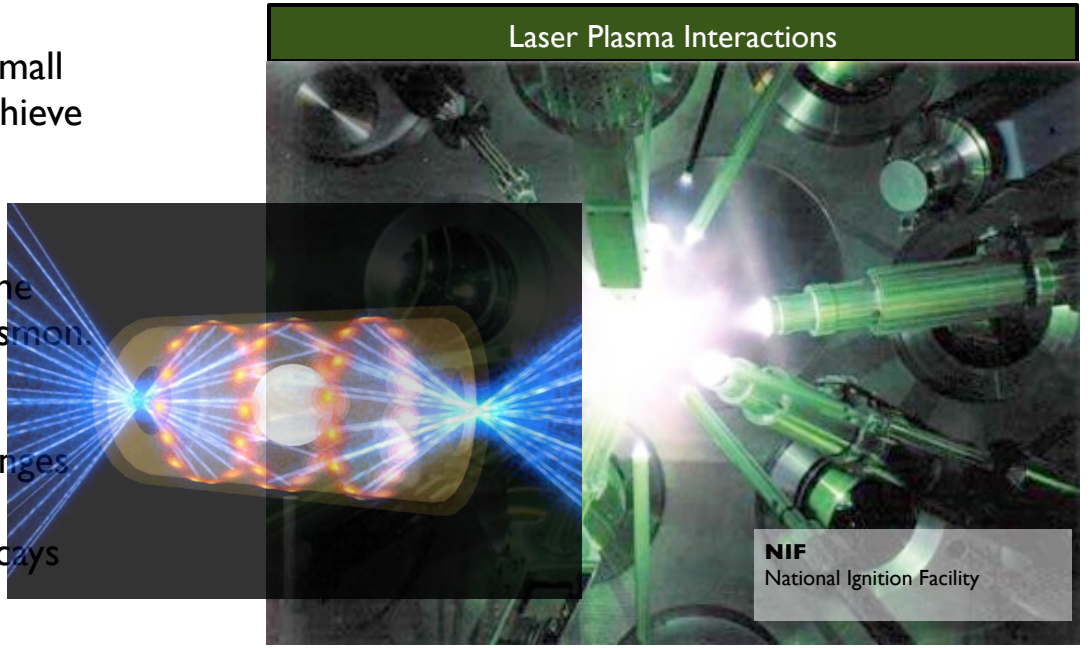
The PWFA-LC illustrates the key questions that must be answered:

- Producing high quality beams with both e- and e+. (the beam is made up of positrons but the plasma is not, so the plasma waves created by positrons is very diffrent from those created by electrons, especially in the nonlinear regime)
- Small energy spread (required to achieve luminosity and luminosity spectrum)
- **Small emittance (i.e., transverse profile)** and small emittance dilution (required to achieve luminosity). -- This requires very fine resolution in the transverse dimensions and requires simulations which are much larger than current studies. We feel that Blue Waters is well suited for this particular area study.
- Staging of multiple PWFA's
- Source of "dark current" in current FACET experiments

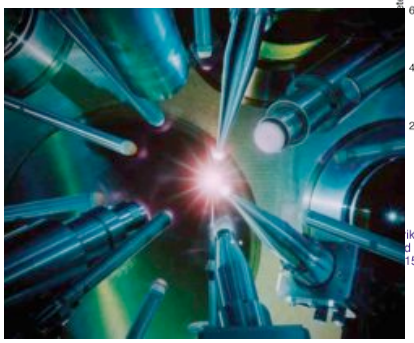
Laser Plasma Interactions relevant to IFE

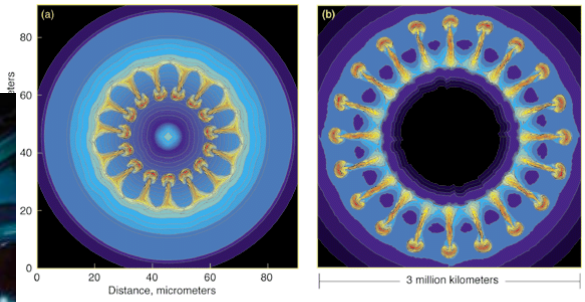
In Inertial Fusion (IFE), lasers are used to compress small pellets made of D-T to > solid density in order to achieve fusion. The two main IFE schemes are:

- Indirect Drive, e.g. NIF, is susceptible to SRS, where the laser decays to a backward going laser light and a plasmon.
- Direct Drive, e.g., OMEGA/LLE, where the laser impinges directly on the target. In this scenario, the laser is susceptible to both SRS and TPD, where the laser decays into two counter-propagating plasma waves.



- PIC codes are ideally suited for such problems:
  - Because PIC does not use any linear approximations, it allows for arbitrarily large plasma waves and other nonlinearities to develop
  - Do not assume the dominant physics before the start of the simulation, in direct drive experiments where SRS and TPD can both occur PIC is not biased toward a dominant absorption mechanism.





iking similarities exist between hydrodynamic instabilities in (a) inertial confinement fusion capsule implosio
d (b) core-collapse supernova explosions. [Image (a) is from Sakagami and Nishihara, *Physics of Fluids B* 2
15 (1990); image (b) is from Hachisu et al., *Astrophysical Journal* **368**, L27 (1991).]

LPI Example 1: NIF-Relevant Simulations of Stimulated Raman Scattering(in 1D)

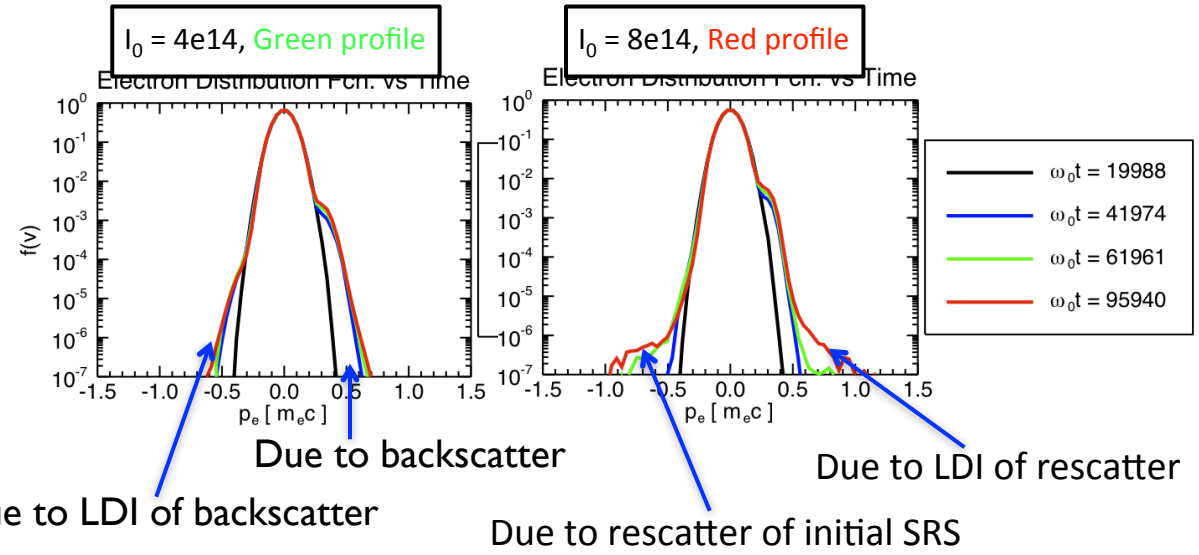
- We have simulated stimulated Raman scattering (SRS) in 1D and 2D over 0.5-1.5 mm lengths with NIF-relevant density profiles

- 1D simulations are quick and allow for methodical parameter scans and comparisons with linear theory
  - Hydro conditions ⟶ NIF uses 1D fluid postprocessing tools such as SLIP/NEWLIP:

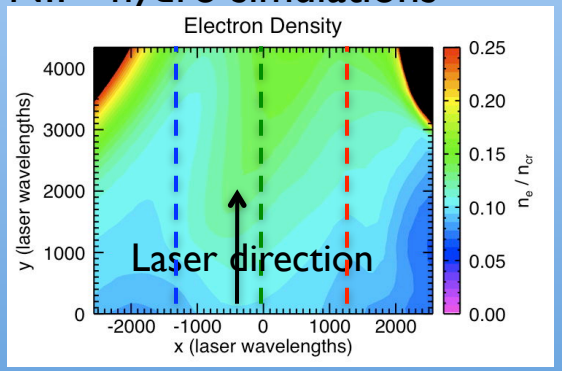
    Predict the frequency and reflectivity of the most unstable LPI
  - Hydro conditions ⟶ 1D OSIRIS simulations:

    Similar capabilities + detailed information about energy partition, backscattered light, and energetic electrons, and identify the mechanisms for hot electron generation.

    We hope small 1D PIC simulations can replace the SLIP/ NEWLIP calculations



$I_{laser} = 2 - 8 \times 10^{14}$ W/cm^2
$\lambda_{laser} = 351$nm,
$T_e = 2.75$ keV,
$T_i = 1$ keV, Z=1,
t_{max} up to 20 ps
Length = 1.5 mm
Density profiles from NIF hydro simulations



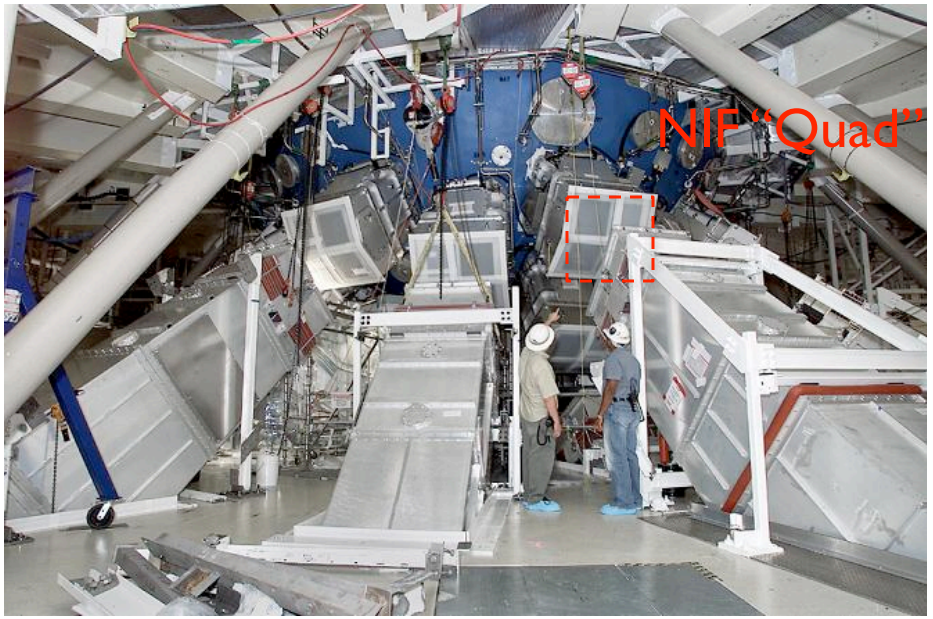
14 million particles
~400 CPU hours per run
~1 hr on modest size supercomputer

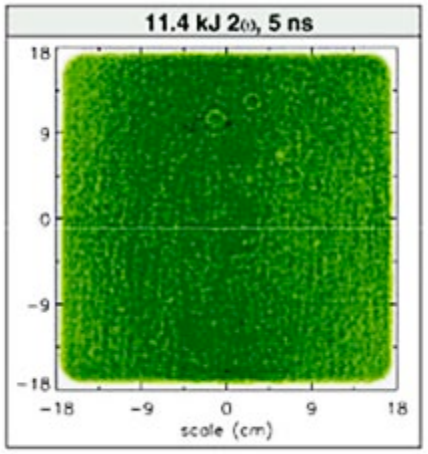
We have simulated stimulated Raman scattering in multi-speckle scenarios (in 2D)

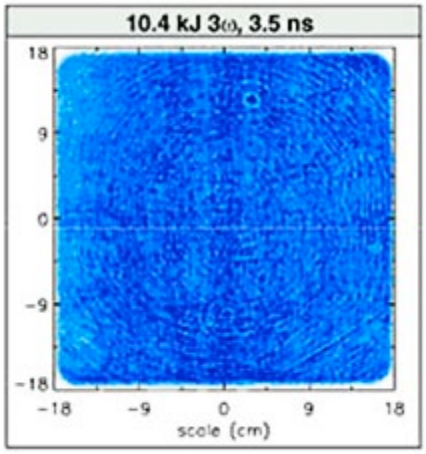
- The SRS problem is not strictly 1D -- each "beam" (right) is made up of 4 lasers, called a NIF "quad," and each laser is not a plane wave but contains "speckles," each one a few microns in diameter. These hotspots can produce highly nonlinear LPI and trigger LPI's in neighbor speckles

- We have been using OSIRIS to look at SRS in multi-speckle scenarios. In our simulations using two controlled speckles (one above threshold and one below threshold) we observed the excitation of SRS in under-threshold speckles via:
  - "seeding" from backscatter light from neighboring speckles
  - "seeding" from plasma wave seeds from a neighboring speckle.
  - "inflation" where hot electrons from a neighboring speckle flatten the distribution function and reduce plasma wave damping.

- The interaction of multiple speckles is a highly complex process and is ideally suited for PIC simulations







Multispeckle SRS Simulation using NIF density profiles --- 2D OSIRIS Sim on BW

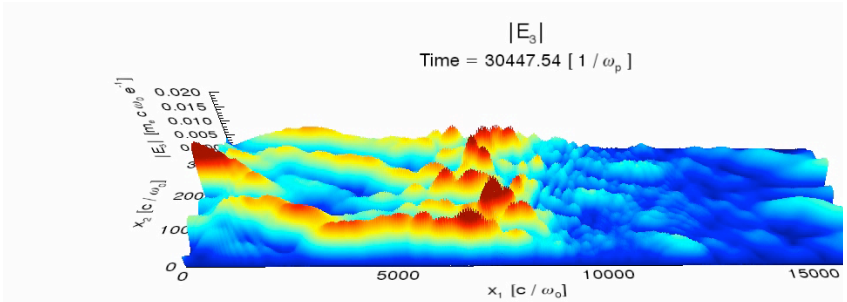
UCLA

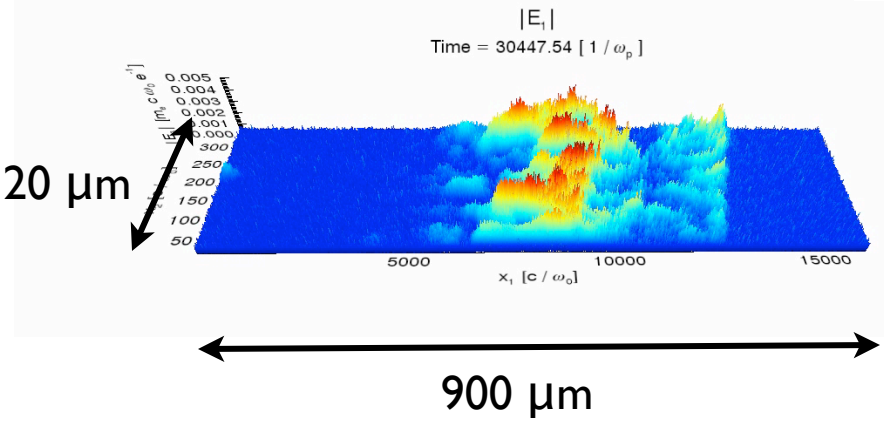
Speckled Laser Beam:

$\lambda = 351\text{nm}$

$I_{avg} = 10^{15}\,W/cm^2$

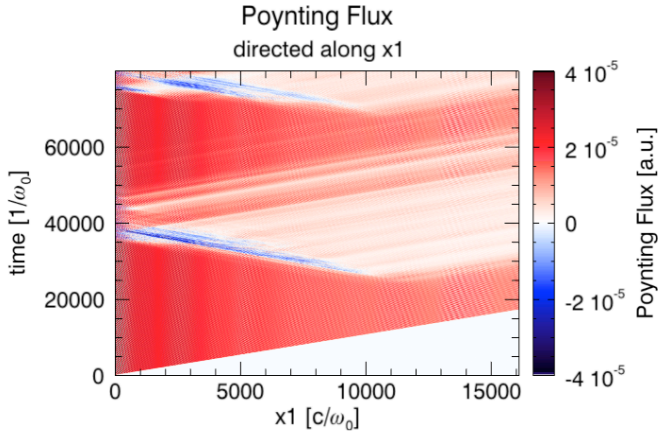
5 speckles long x 7 speckles wide

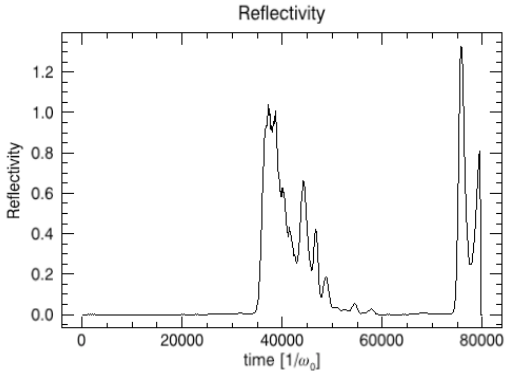




20 μm

900 μm





Domain:

6.4 million cells

16 billion particles

450,000 time steps

Computational:

32768 processors on Blue Waters

880,000 CPU hours

This simulation is ongoing, we will compare simulation results from the speckled simulations and compare that to those from our previous 1D (plane wave) simulations, The simulation is ongoing and will use ~1.5 million CPU hours. These simulations will quantify the effects speckles have on laser reflectivities in NIF experiments.

PIC simulations of LPI's is still a challenge, and requires exa-scale supercomputers, this will require code developments

	2D multi-speckle along NIF beam path	3D, 2 speckles	3D, multi-speckle along NIF beam path
Speckle scale	50 x 8	2 x 1	10 x 10 x 5
Size (microns)	150 x 1500	18 x 9 x 120	28 x 28 x 900
Grids	9,000 x 134,000	1,000 x 500 x 11,000	1,700 x 1,700 x 80,000
Particles	300 billion	620 billion	22 trillion
Steps	470,000 (15 ps)	180,000 (5 ps)	540,000 (15 ps)
Memory Usage*	7 TB	18 TB	1.6 PB
CPU-Hours	8 million	9 million	1 billion (2 months on the full BW)

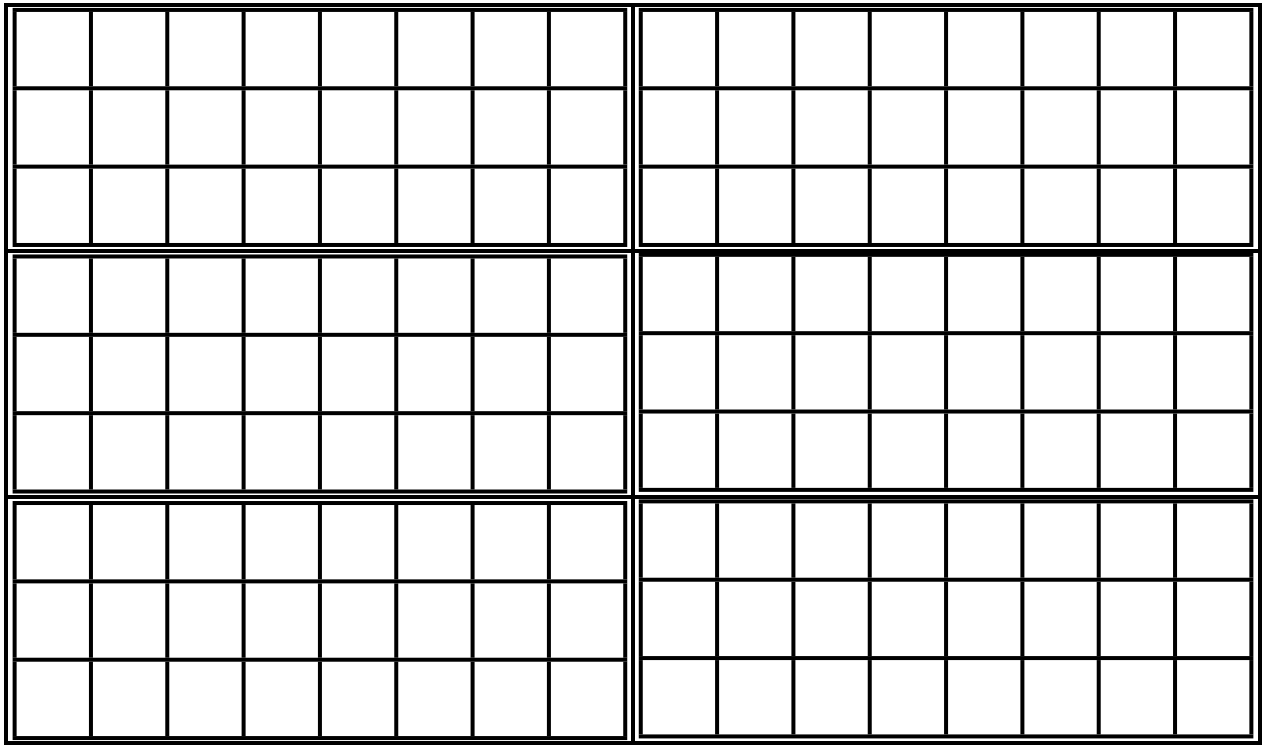
*memory usage can be reduced by the use of higher order particle shapes to reduce noise

Particles ordered by tiles, varying from 2 x 2 to 16 x 16 grid points
On Fermi M2090:

• Associate a **thread block** with each tile (analogous to a domain in domain decomposition) and particles located in that tile

We created a new data structure for particles, partitioned among threads blocks (i.e., particles are sorted according to its tile id, and there is a local domain decomposition within the GPU), within the tile the particles and the particle data are fairly aligned and the loops can be easily parallelized

dimension part(npmax,idimp,num_blocks)

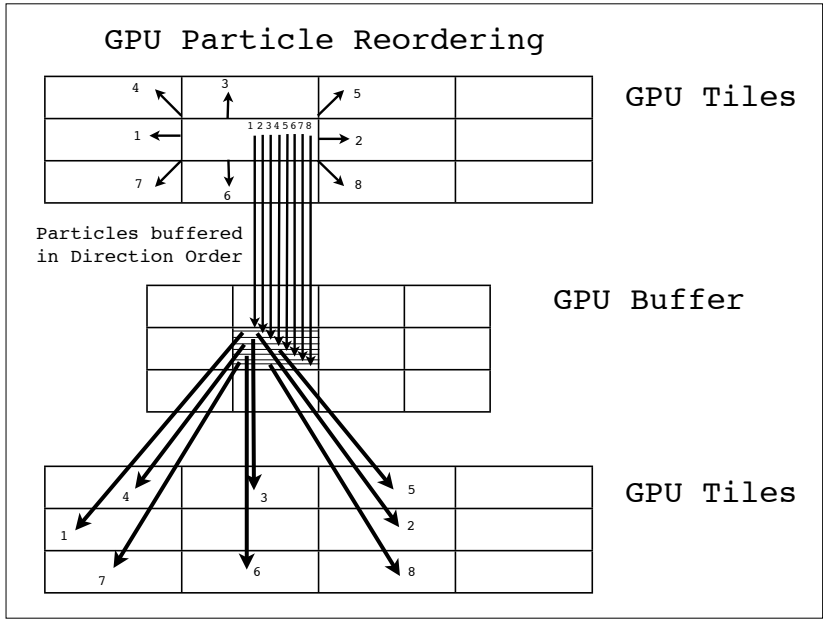


Designing New Particle-in-Cell (PIC) Algorithms:
Maintaining Particle Order

Three steps:
1. Particle Push creates a list of particles which are leaving a tile (much like a MPI send/receive buffer)
2. Using list, each thread places outgoing particles into an ordered buffer it controls (a la MPI_send)
3. Using lists, each tile copies incoming particles from buffers into particle array (a la MPI_receive)

- Less than a full sort, **low overhead** if particles already in correct tile
- Essentially message-passing, except buffer contains multiple destinations (and we know how to program this)

In the end, the particle array belonging to a tile has no gaps
- Particles are moved to any existing holes created by departing particles
- If holes still remain, they are filled with particles from the end of the array



Evaluating New Particle-in-Cell (PIC) Algorithms on GPU: **Electrostatic Case**
2D ES Benchmark with 2048x2048 grid, 150,994,944 particles, 36 particles/cell
optimal block size = 128, optimal tile size = 16x16

GPU algorithm also implemented in OpenMP where the message passing within a node is handled using the scheme described earlier (on a 12 core intel i7 where it achieved close to 12x speedup on 12 cores)

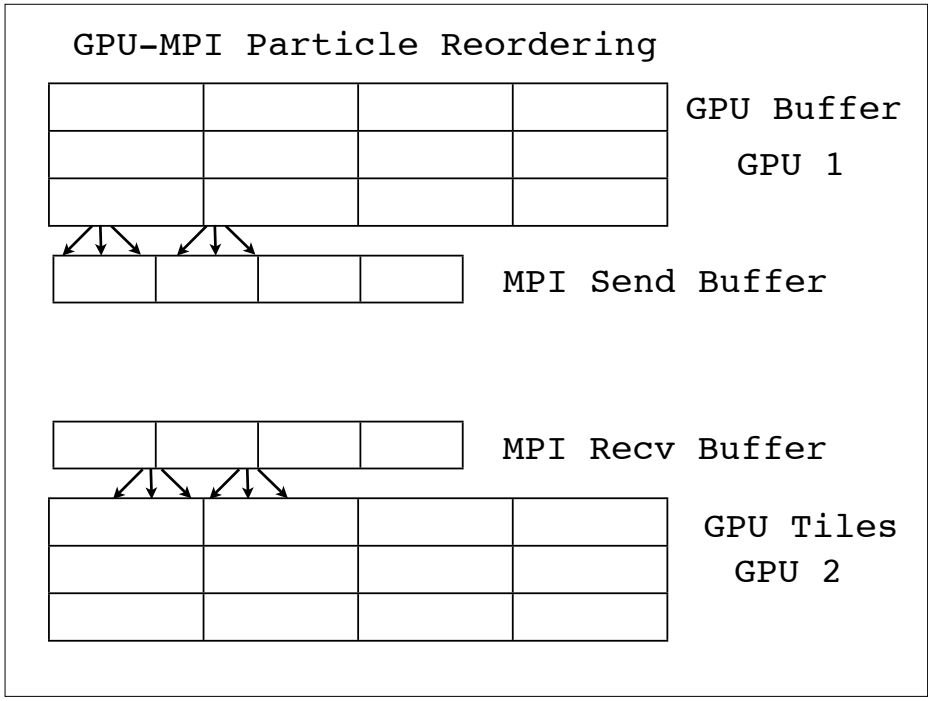
Hot Plasma results with dt = 0.1

	CPU:Intel i7	GPU:Fermi M2090	OpenMP(12 CPUs)
Push	22.1 ns.	532 ps.	1.678 ns.
Deposit	8.5 ns.	227 ps.	0.818 ns.
Reorder	0.4 ns.	115 ps.	0.113 ns.
Total Particle	31.0 ns.	874 ps.	2.608 ns.

The time reported is per particle/time step.
The total particle speedup on the Fermi M2090 was 35x compared to 1 CPU
Field solver takes an additional 6% on GPU, 11% on CPU.

OK, how about multiple GPU/CPU's?

This scheme can be easily parallelized since the particle manager that moves particles between tiles can be easily extended to move these particles into MPI buffers (as shown on the right, and in previous slides) For those tiles that sits on the processor edge, the particle managers simply place the outgoing particles into an MPI buffer.



Evaluating New Particle-in-Cell (PIC) Algorithms on GPU: Electrostatic Case
2D ES Benchmark with 512x512 grid, 9,437,184 particles, 36 particles/cell
(Field Solver not yet implemented on GPUs)

Hot Plasma results with dt = 0.1		
CPU:Intel i7	1 core	12 cores
Push	20.30 ns.	1.80 ns.
Deposit	8.34 ns.	0.75 ns.
Reorder	0.34 ns.	0.04 ns.
MPI Move	0.01 ns.	0.04 ns.
Total Particle	28.94 ns.	2.64 ns.

GPU:Fermi M2090	1 GPU	3 GPUs
Push	345 ps.	135 ps.
Deposit	266 ps.	97 ps.
Reorder	478 ps.	187 ps.
MPI Move	36 ps.	88 ps.
Total Particle	1125 ps.	506 ps.

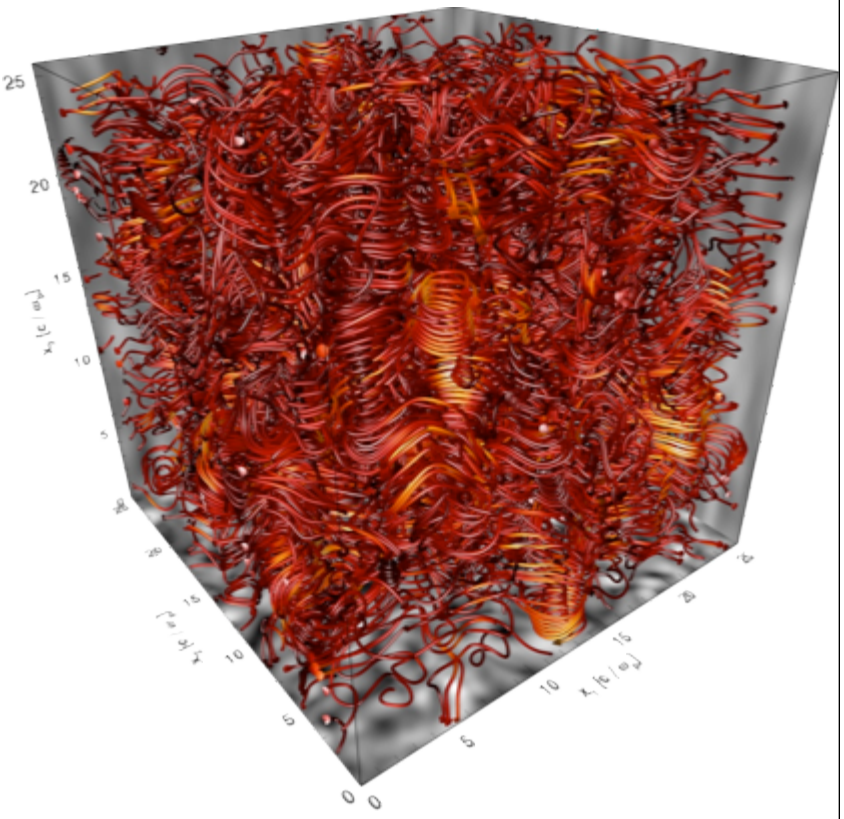
The time reported is per particle/time step.
The total speedup on the 3 Fermi M2090s compared to 12 cores was 5.2x (or roughly 63 intel cores), 2 GPU's should be equivalent to ~40-45 cores
Speedup on 3 M2090s compared to 1 M2090 was 2.2x

We will benchmark the parallel GPU code on Blue Waters **very** soon.

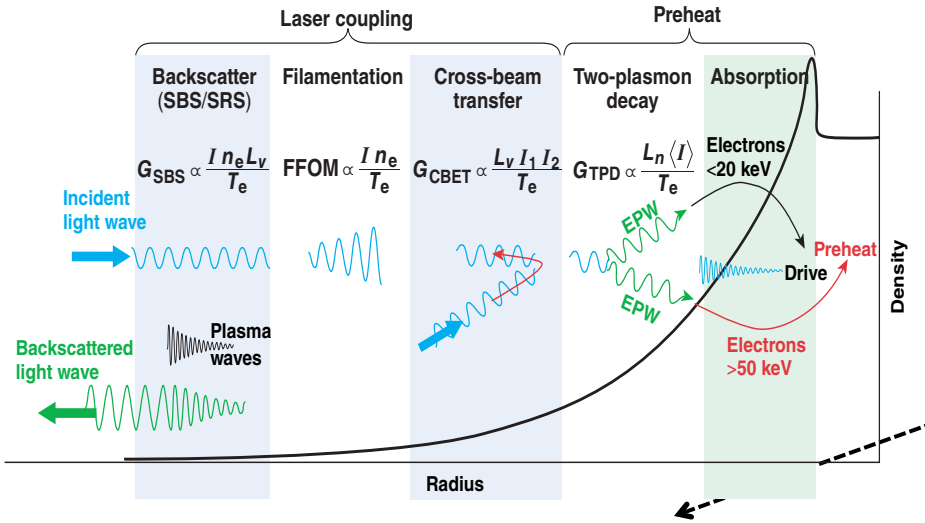
Summary and Outline

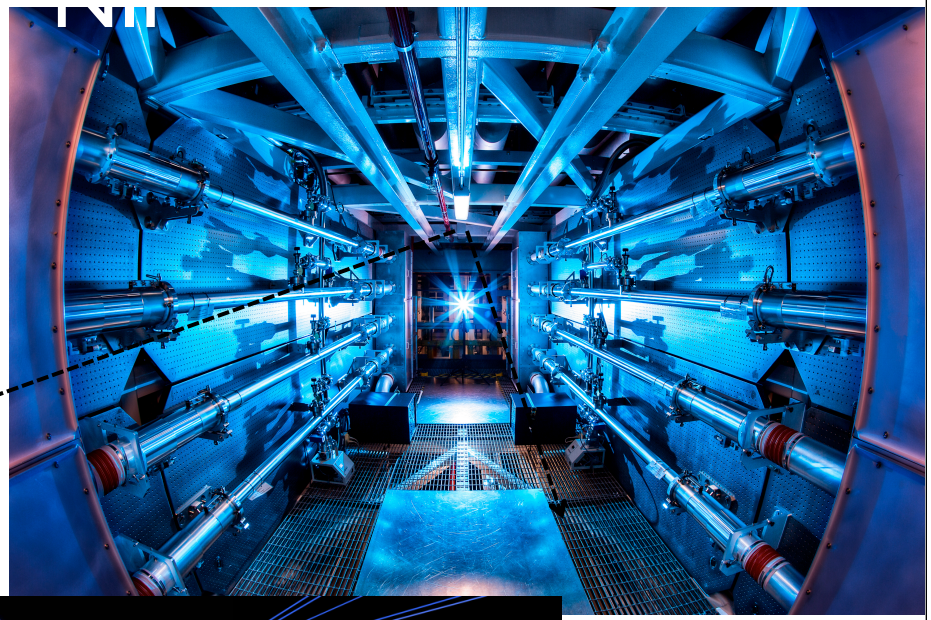
OUTLINE

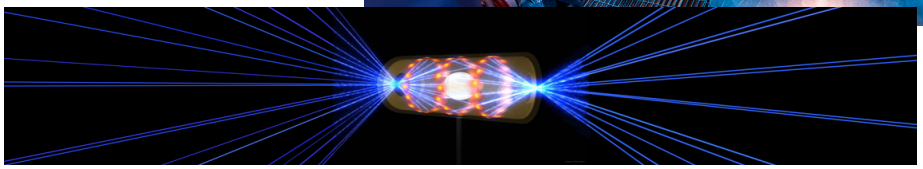
- Overview of the project
  - Particle-in-cell codes
  - OSIRIS and recent >2PFlops benchmark on Blue Waters
- Application of OSIRIS to plasma based accelerators:
  large, full scale 3D simulations of plasma based accelerators and quantitative agreements between simulations and experiments.
- Applications of OSIRIS to LPI's Relevant to IFE
  - SRS in indirect drive IFE targets.
  - LPI (including SRS and TPD) under shock ignition relevant conditions.
  - Estimates of large scale LPI simulations (& the need for exascale supercomputers)
- Development works for Blue Waters and beyond (including GPU's and other emerging architectures)



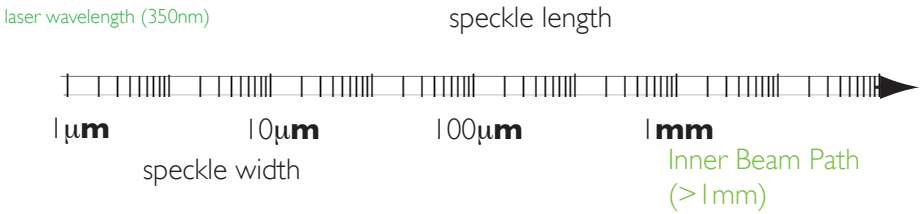
Overview of Laser Plasma Interactions in Inertial Fusion Energy (IFE) -- It is a complicated problem spanning many orders of magnitudes in time and space.



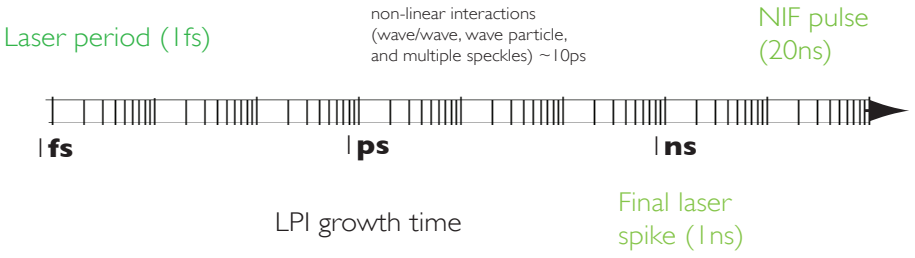




Lengthscales



Timescales



BW-2013

Conclusions

PIC Algorithms on GPU's are largely a hybrid combination of previous techniques

- Vector techniques from Cray
- Blocking techniques from cache-based architectures (such as Intel)
- Message-passing techniques from distributed memory architectures

Scheme should be portable to other architectures with similar hardware abstractions (we have shown that this technique is portable by developing an OpenMP version of this code)

2D Electrostatic code (where we get a 35x speedup compared to a single CPU) is a very simple code, with low computational intensity

- only 55 Flops per particle update

PIC codes with higher computational intensity should do better

- EM codes, 3D codes, higher order interpolations (OSIRIS is all of these)

Further information available at:
http://www.idre.ucla.edu/hpc/research/

Evaluating New Particle-in-Cell (PIC) Algorithms on GPU: **Electromagnetic Case**
2-1/2D EM Benchmark with 2048x2048 grid, 150,994,944 particles, 36 particles/cell
optimal block size = 128, optimal tile size = 16x16

GPU algorithm also implemented in OpenMP (on a 12 core i7 it achieved a >11x speedup)

Hot Plasma results with dt = 0.04, c/vth = 10, relativistic

	CPU:Intel i7	GPU:Fermi M2090	OpenMP(12 CPUs)
Push	66.5 ns.	0.426 ns.	5.645 ns.
Deposit	36.7 ns.	0.918 ns.	3.362 ns.
Reorder	0.4 ns.	0.698 ns.	0.056 ns.
Total Particle	103.6 ns.	2.042 ns.	**9.062 ns.**

The time reported is per particle/time step.
The total particle speedup on the Fermi M2090 was 51x compared to 1 CPU (or 1.6 x a 32 core Intel processor)

Field solver takes an additional 10% on GPU, 11% on CPU.

OK, so how about multiple CPU/GPU's?

Designing New Particle-in-Cell (PIC) Algorithms: **Multiple GPUs**

Multiple GPUs on one node can be controlled either with OpenMP or MPI

We started with an existing 2D Electrostatic MPI code from UPIC Framework
- Replacing MPI push/deposit with GPU version was no major challenge

With multiple GPUs, we need to integrate two different partitions
- MPI and GPU each have their own particle managers to maintain particle order

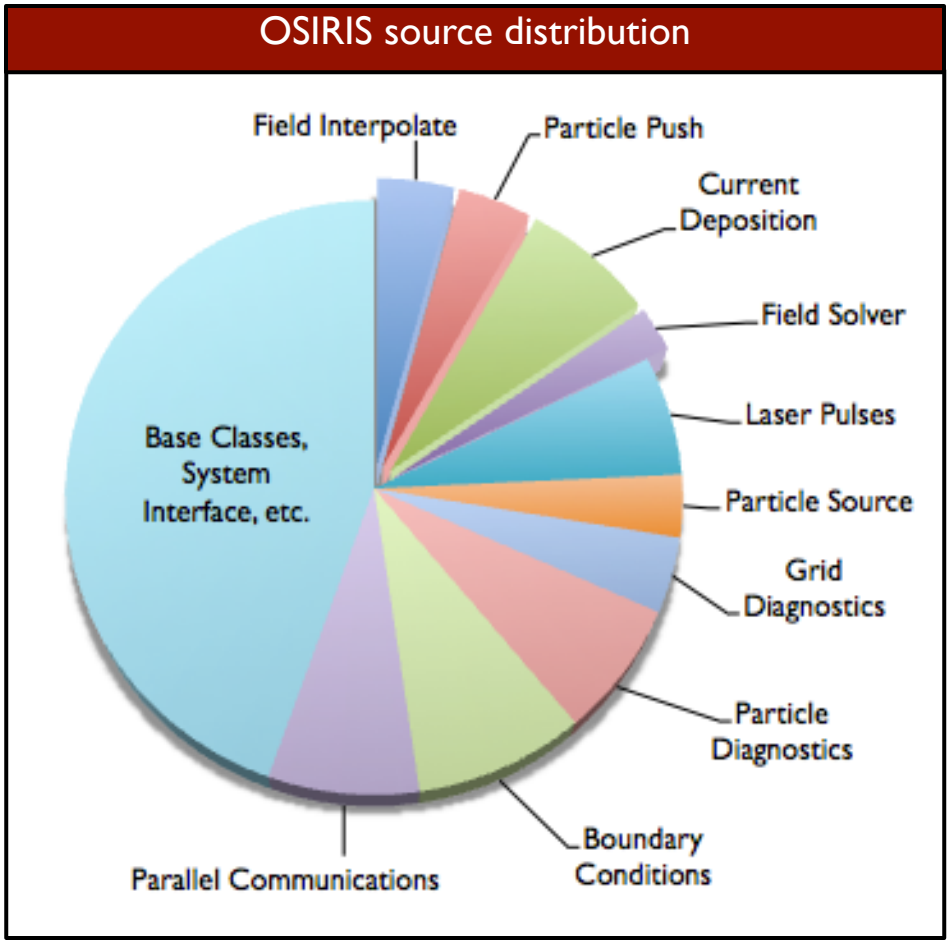
Only the first/last row or column of tiles on GPU interacts neighboring MPI node
- Particles in row/column of tiles collected in MPI send buffer
- Table of outgoing particles are also sent
- Table is used to determine where incoming particles must be placed

Field solver still performed on host

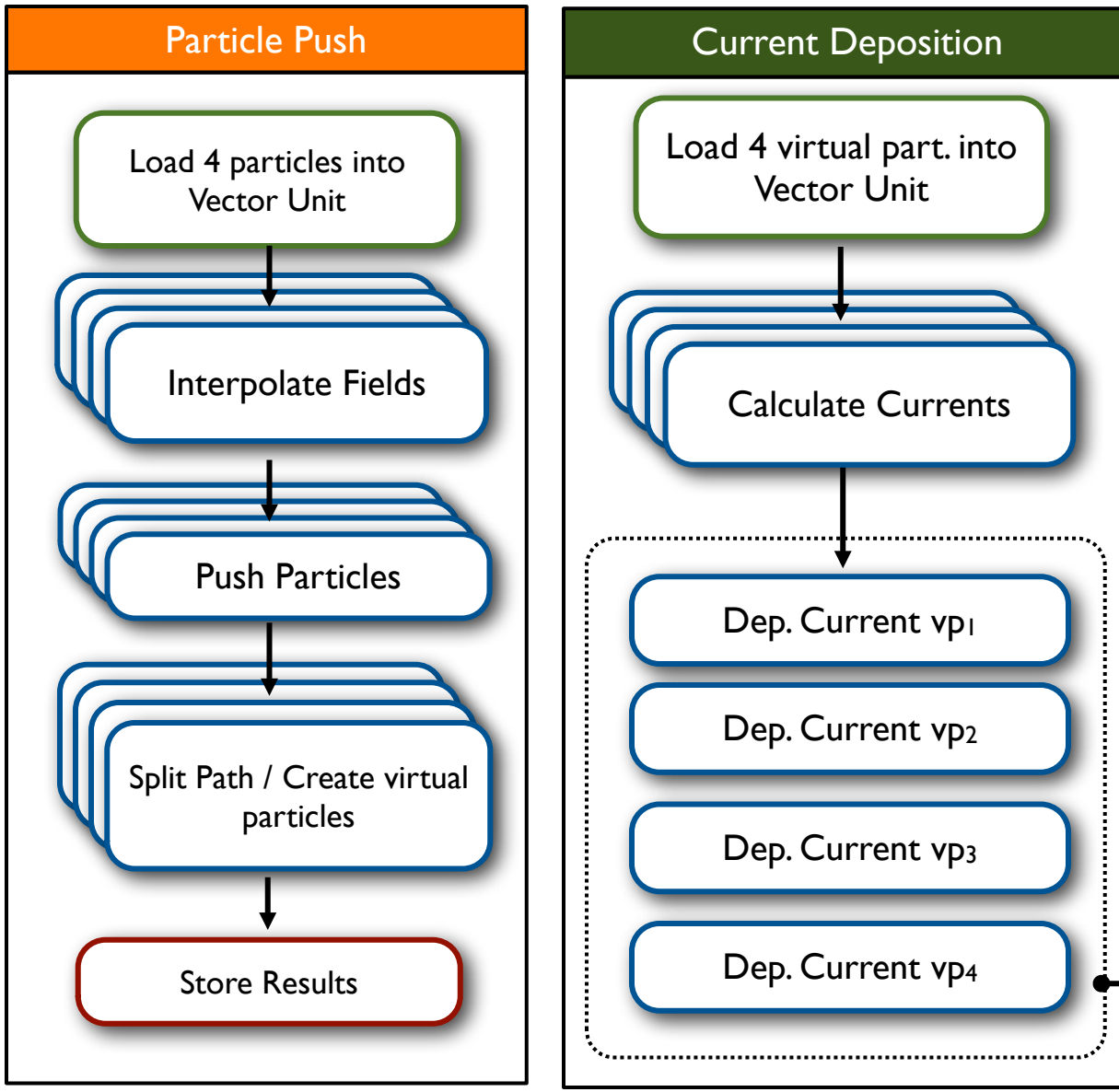
Source Code

- Basic PIC algorithm (with various implementations) accounts for only 18% of the code (but accounts for > 90% of the timing)
- Parallel communications correspond to less than 10% of the code

- Most of the code is devoted to:
  - Diagnostics
  - Additional physics
  - User Interface



SIMD Algorithm



- PIC codes are good candidates for optimization:
  - Operations on each particle independent from each other...
  - except for current deposition
  - For most cases work well in single precision
- Process 4 particles at a time
- Memory access much more expensive than calculation
  - Avoid temp buffers
  - Optimize for cache

- Particles may deposit to same cell
- Process each 4 particles sequentially