Computational advances in gyrokinetic particle-in-cell simulations

Stéphane Ethier

Princeton Plasma Physics Laboratory

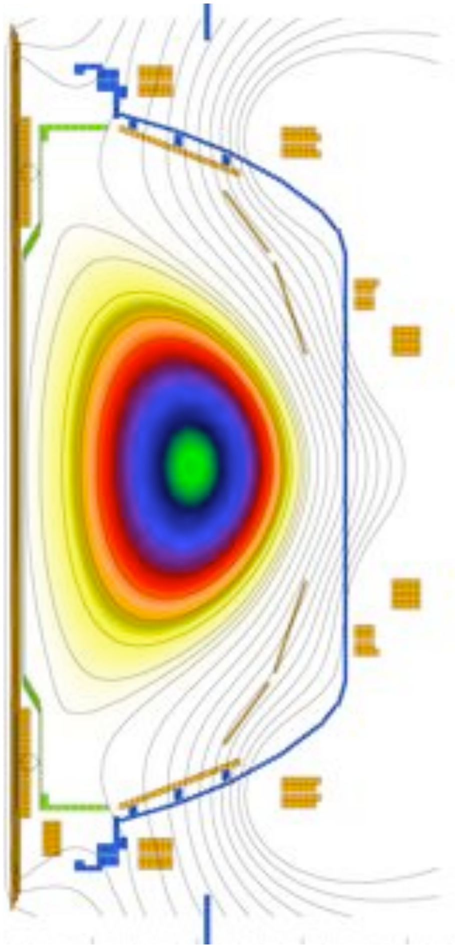
Accelerating Computational Science Symposium
March 29, 2012, Washington DC

Work supported by the DOE contract DE-AC02-09CH11466

PPPL
PRINCETON
PLASMA PHYSICS
LABORATORY

Gyrokinetic simulations: for studying turbulent transport (and more?)

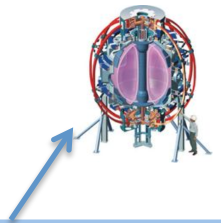
- Macroscopic Stability
  - What limits the pressure in plasmas?
- Wave-particle Interactions
  - How do particles and plasma waves interact?
- Microturbulence & Transport
  - What causes plasma transport?
- Plasma-material Interactions
  - How can high-temperature plasma and material surfaces co-exist?



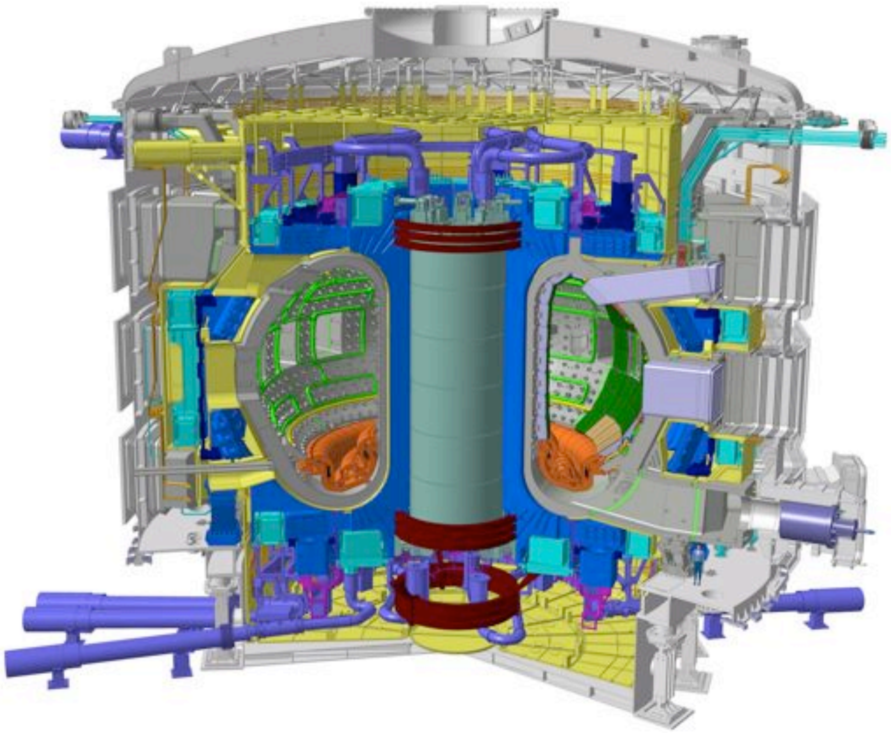
PPPL

The challenge: Global simulation of ITER

- ITER is extremely large compared to current experiments
- We need advanced simulations to predict its performance
- VERY large smulations

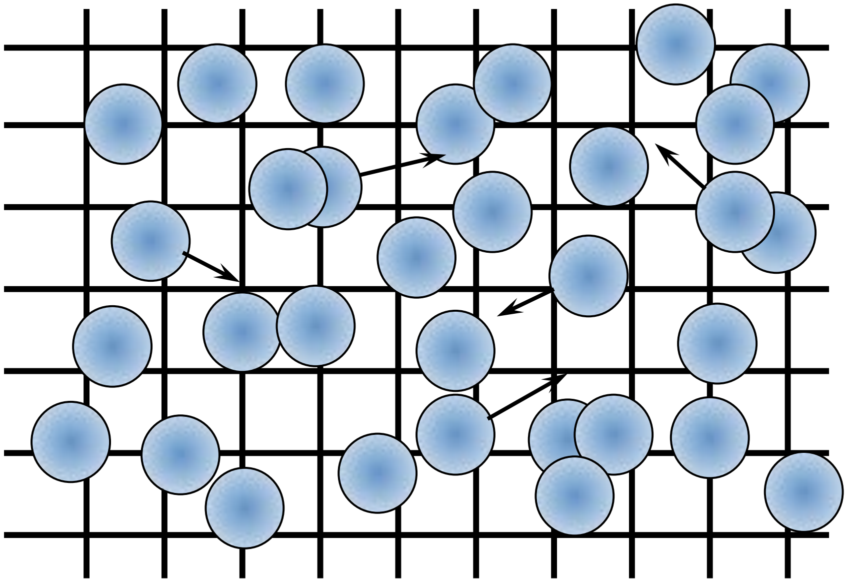


Current experiment at PPPL



The Particle-in-Cell method in a nutshell

- Particles sample distribution function
- Interactions via the grid, on which the potential is calculated (from deposited charges).
- Grid resolution dictated by Debye length or gyroradius



The PIC Steps

- **"SCATTER", or deposit, charges on the grid (nearest neighbors)**
- **Solve Poisson equation**
- **"GATHER" forces on each particle from potential**
- **Move particles (PUSH)**
- **Repeat…**

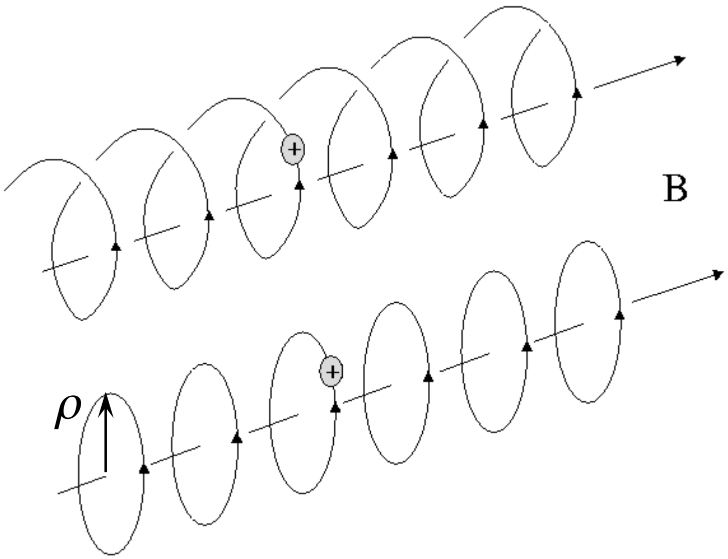
Gyrokinetic approximation for low frequency modes

- Gyrokinetic ordering

$$\frac{\omega}{\Omega_i} \sim \frac{\rho_i}{L} \sim \frac{e\phi}{T} \sim k_{//}\rho_i << 1$$

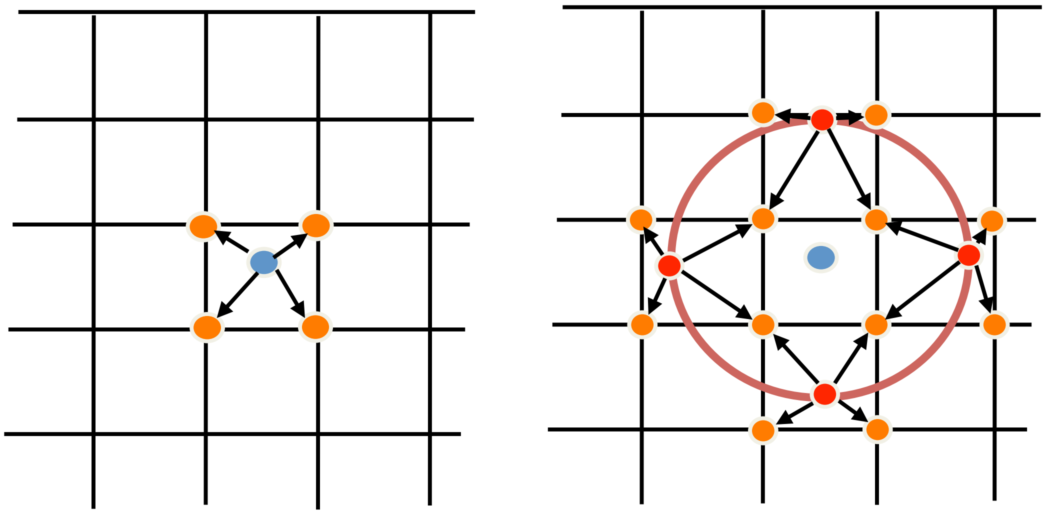
$$k_\perp \rho_i \sim 1$$

- Gyro-motion: guiding center drifts + charged ring
  - Parallel to B: mirror force, magnetically trapped
  - Perpendicular: E x B, polarization, gradient, and curvature drifts
- Gyrophase-averaged **5D** gyrokinetic equation
  - Suppress plasma oscillation and gyro-motion
  - Larger time step and grid size, smaller number of particles



Charge Deposition:
4-point average method

Charge Deposition Step (SCATTER operation)

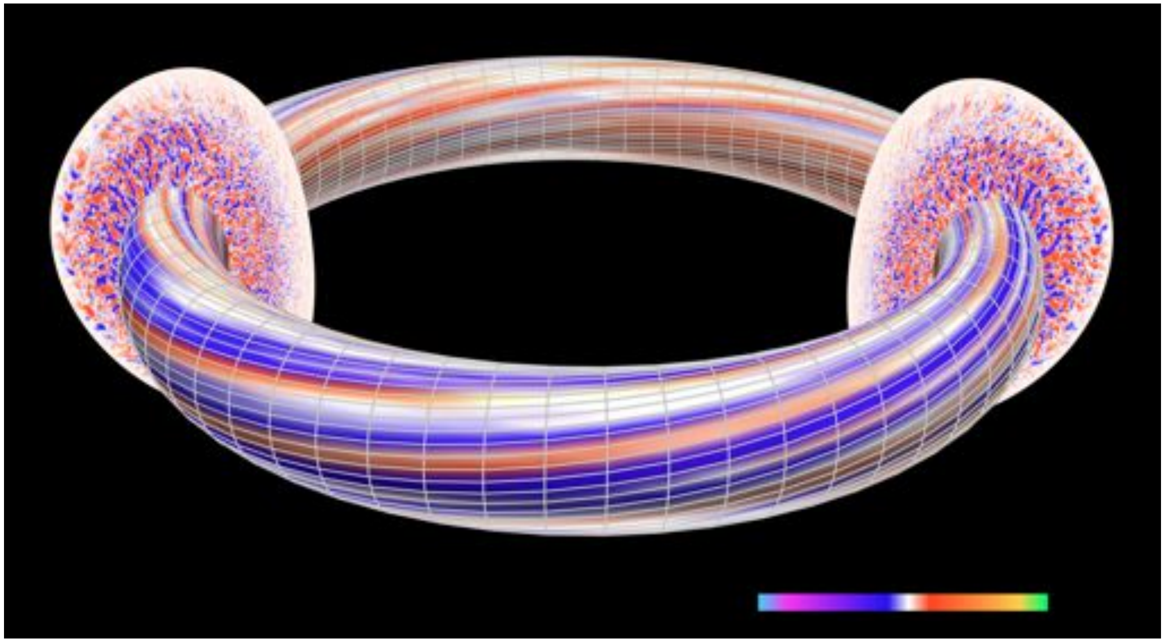


Classic PIC

4-Point Average GK
(W.W. Lee, JCP 1987)

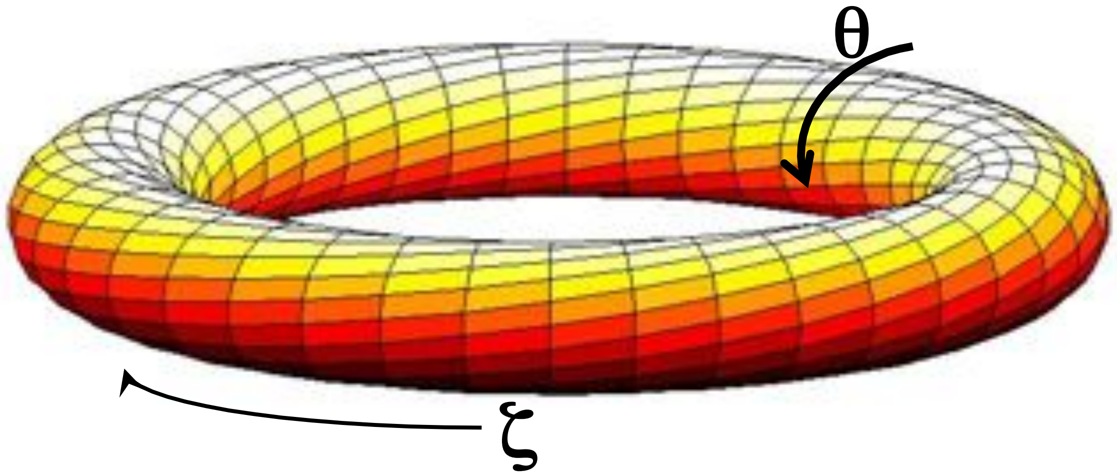
Quasi-2D electrostatic potential

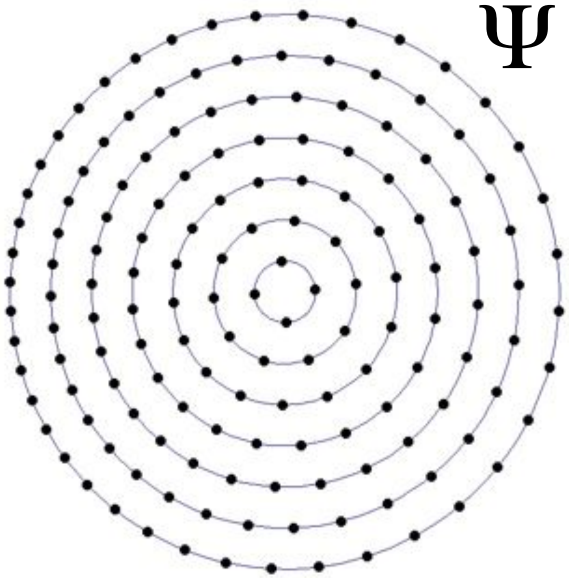
- Ions and electrons move fast along the magnetic field lines ($k_{||} << 1$)
- Slow diffusion across magnetic surfaces (radially) mainly due to ExB field (usually larger than neoclassical diffusion)

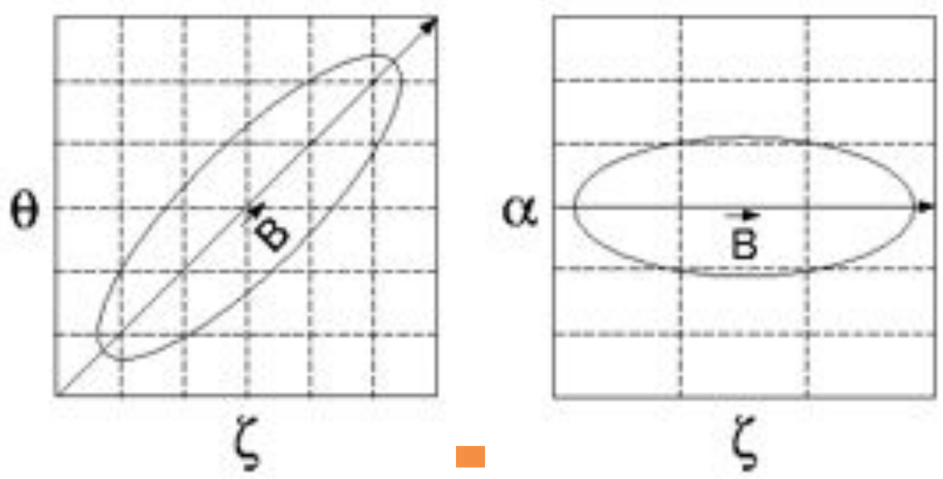


Global Field-aligned Mesh (magnetic coord.)

$$(\Psi,\alpha,\zeta) \;\Rightarrow\; \alpha = \theta - \zeta/q$$



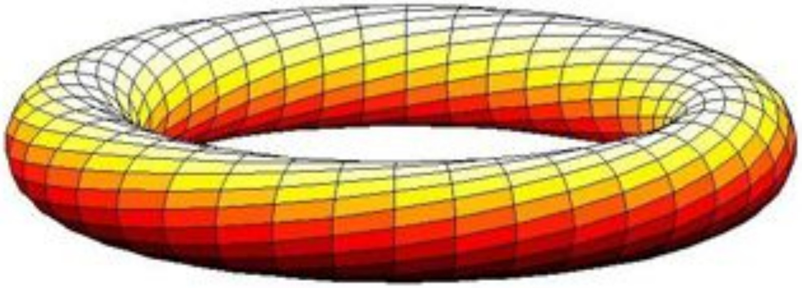




Saves a factor of about 100 in CPU time

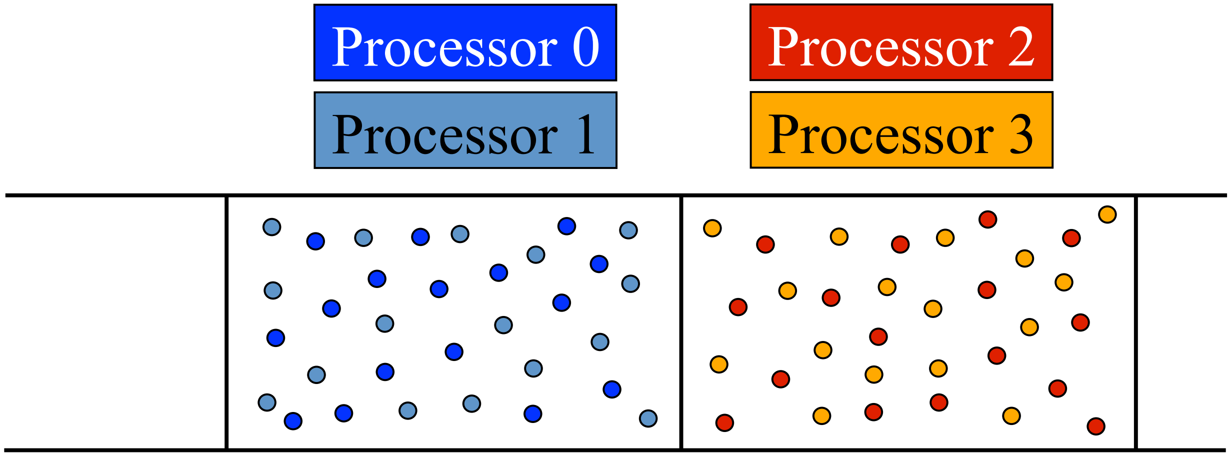
Original domain decomposition in the toroidal direction

- Domain decomposition:
  - each MPI process holds a toroidal section
  - each particle is assigned to a processor according to its position
- Initial memory allocation is done locally on each processor to maximize efficiency
- Communication between domains is done with MPI calls (in a ring-like fashion for point-to-point particle motion)

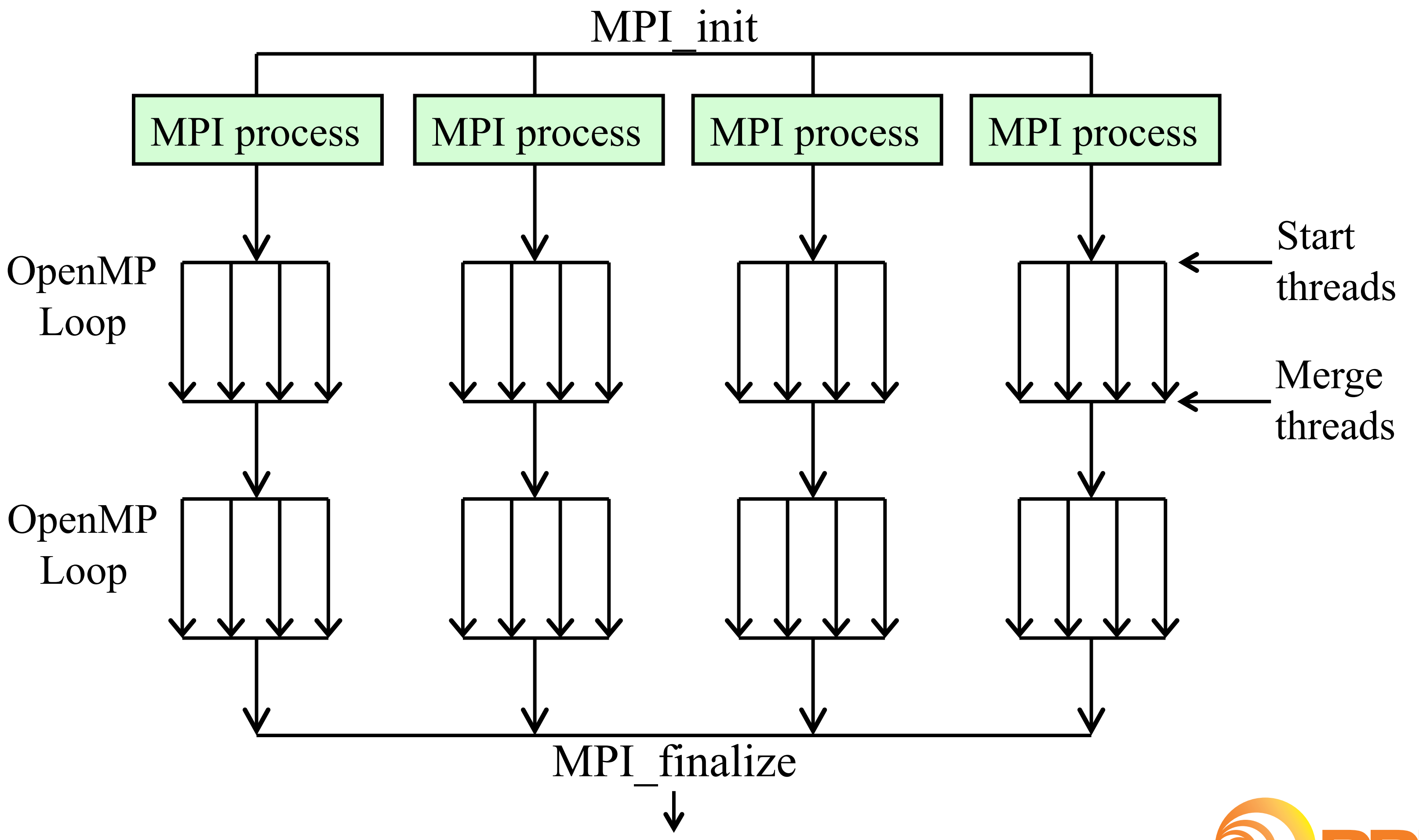


Increasing concurrency further with MPI-based particle distribution

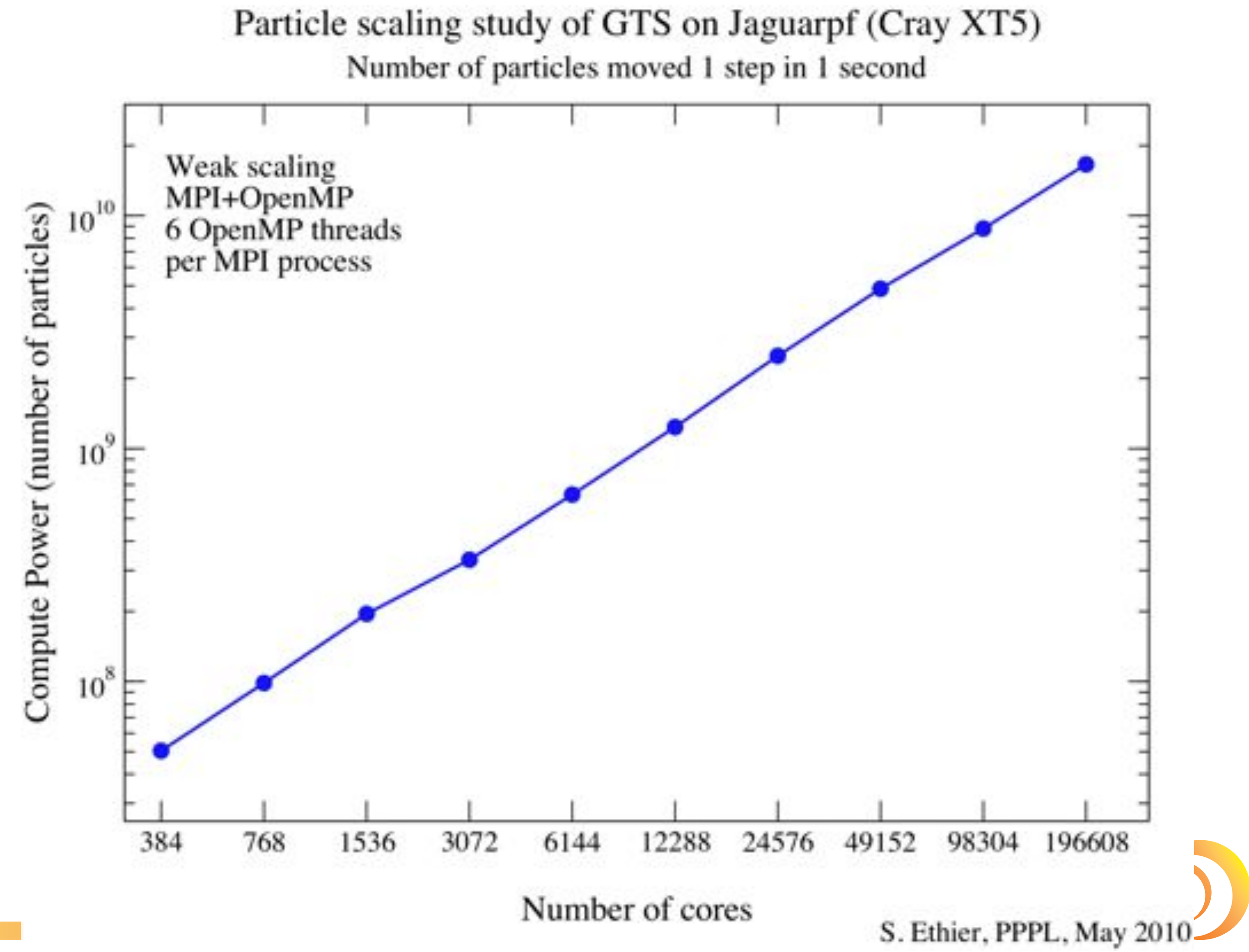
- Each domain in the 1D domain decomposition can have more than 1 processor associated with it.
- Each processor holds a fraction of the total number of particles in that domain.
- Scales perfectly with the number of particles
- Original version did not improve grid-based solver though. Solution: PETSc parallel solver



2nd Level of Parallelism: Loop-level with OpenMP

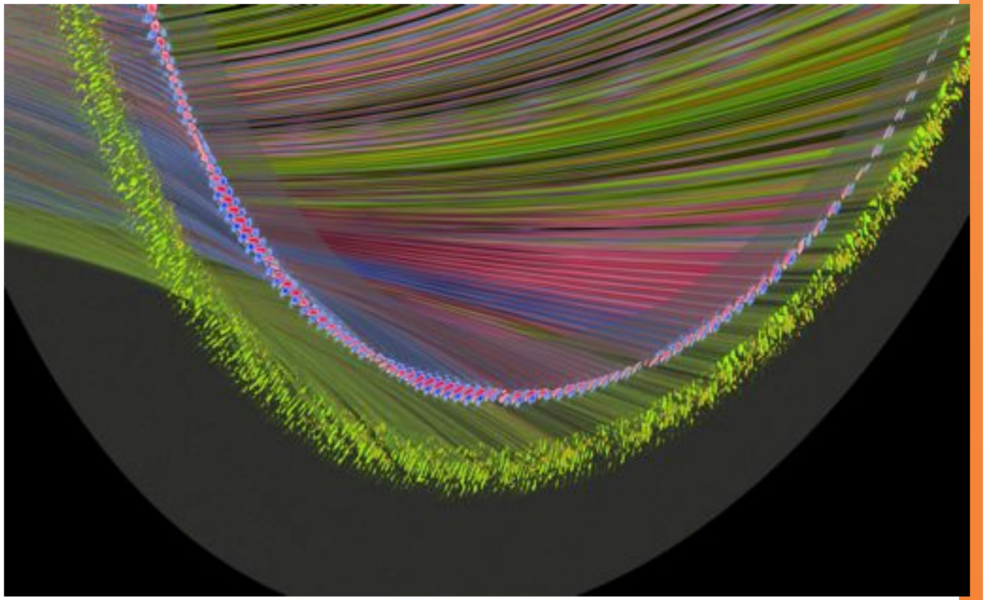


GTS particle scaling on Cray XT5: MPI+OpenMP on multi-core nodes



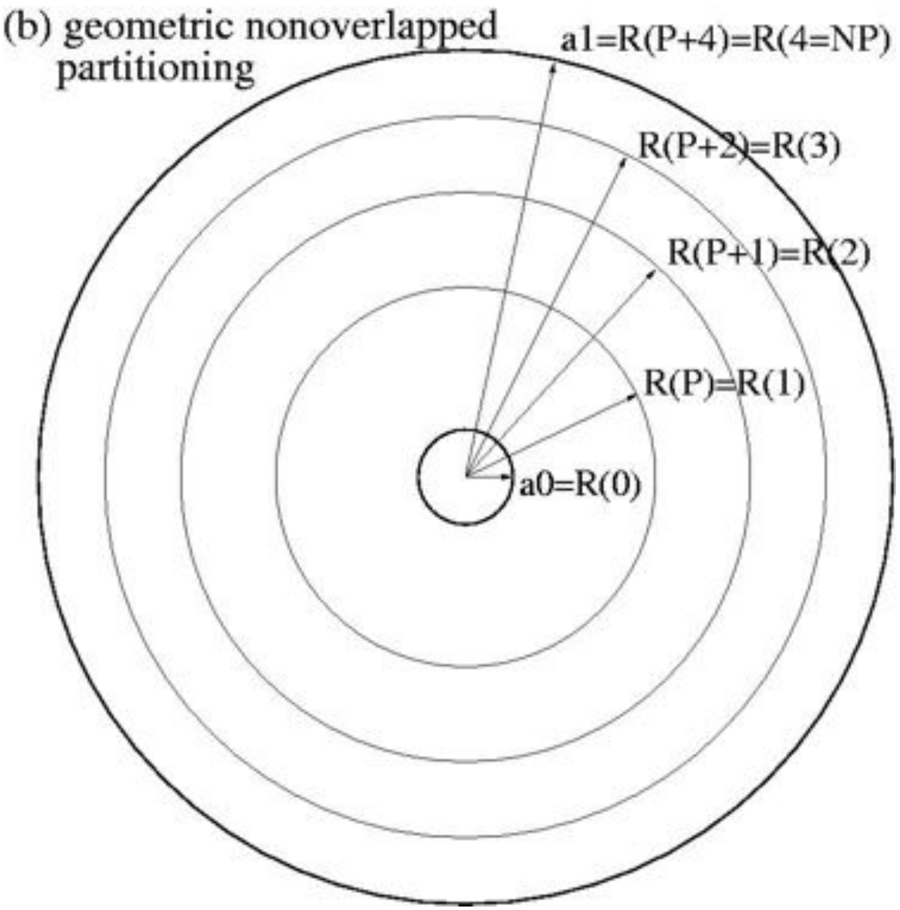
ETG simulations on Cray XT5:
Production-version of GTS with MPI+OpenMP

- ETG simulation of real-size tokamak → NSTX
- Direct comparison to experimental data
- Simulation uses 23 billion particles and 400 million grid points
- Production simulations on 98,304 cores

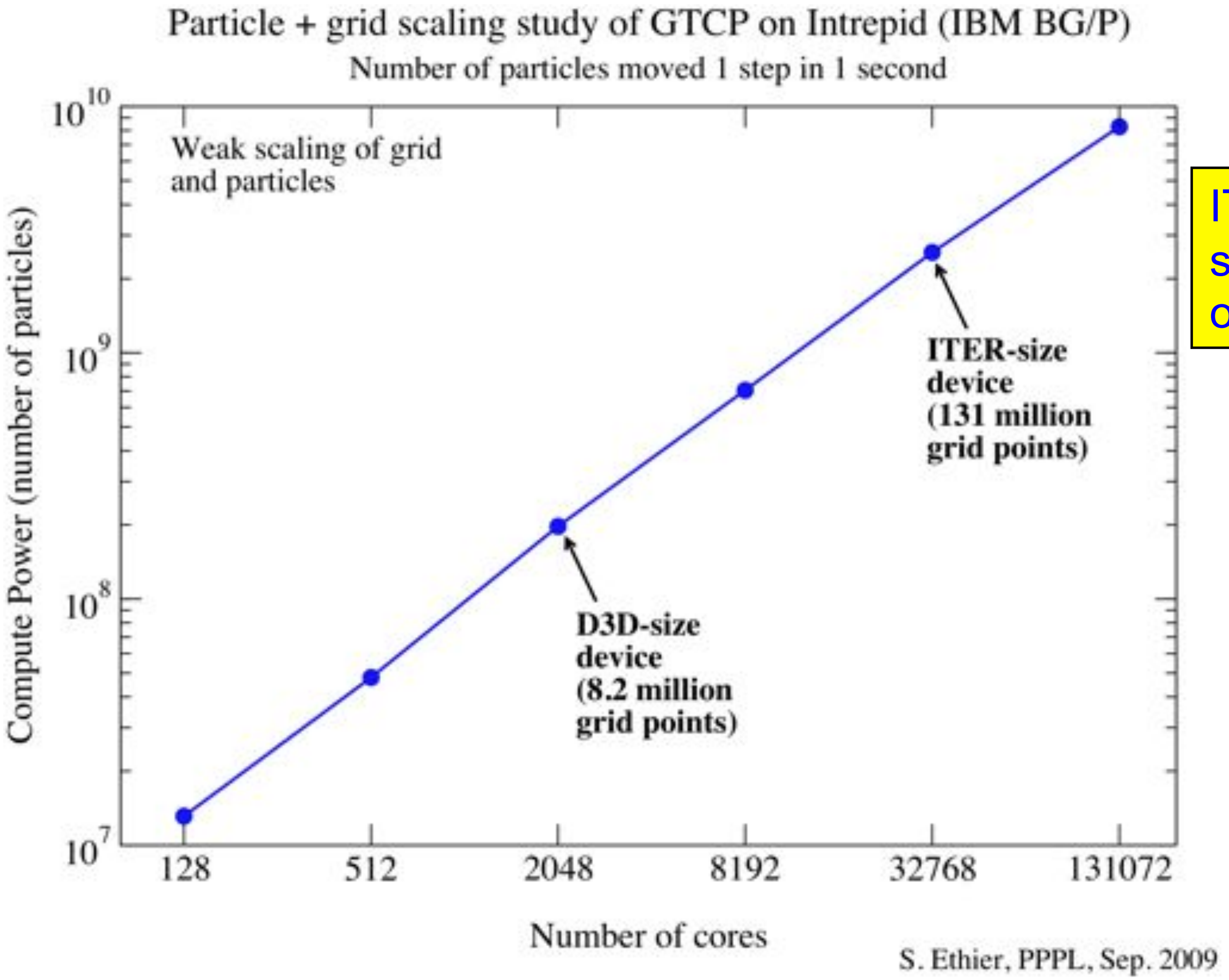


Addressing grid scalability with radial grid decomposition

- Non-overlapping geometric partitioning



Great scaling of 2D domain decomposition on Blue Gene/P system (flat MPI)



S. Ethier, PPPL, Sep. 2009

ITER-size simulation with only 512MB/core!

How do we move forward to be ready for Exascale systems?

- Weak scaling has its limits. How grid is getting too large to fit on a node for electron-scale simulations. We need to start focusing on strong scaling.
- The gather-scatter PIC algorithm wastes too much time accessing random numbers in a large grid array so we have to improve locality by sorting the particles according to their positions.
- We need to explore highly multi-threaded algorithms so that we can hide memory latency.
- Flops are cheap and memory is limited, and its access is expensive, so we can redo some calculations instead of storing results that are reused somewhere else in the code (e.g. particle-grid positions).
- Explore GPUs and emerging architectures.
- Close collaboration with computer scientists and applied mathematicians.

Computational characteristics of GKPIC algorithm

- Push ions (P,P-G)
  - Major computational kernel "Moves particles"
  - Large body loops; Gathers*; Lots of loop-level parallelism*
- Charge deposition (P-G)
  - Major computational kernel "Scatter"
  - *Pressure on cache* – unstructured access to grid – block grid
- Shift ions, communication (P)
  - Sorts out particles that move out of its domain and sends those to the "next" processor – large point-to-point communication
- Poisson Solver (G)
  - Solve Poisson Equation. Prior to 2007 the solve was redundantly executed on each processor. Current version uses the PETSc solver to efficiently distribute the work
- Smooth (G) and Field (G)
  - Smaller computational kernels
  - *Executed redundantly by MPI processes on local grid*

GTC on GPU effort

- Several projects focused on porting gyrokinetic PIC codes on GPUs
- Peng Wang, HPC Developer Technology, NVIDIA, and Xiangfei Meng of the National Supercomputer Center, NSCC-TJ
- UC Irvine Prof. Z. Lin's production version of GTC with kinetic electrons (http://phoenix.ps.uci.edu/GTC/)
- Port to Tianhe-1A and TitanDev
- Focus on electron push and shift as they account for 86% of the run time for the chosen test case (70% pushe, 16% shift)
- Best case achieves 3X speedup over CPU-only version, 1.6X for smaller test case
- On-going work to port all of GTC

GTC results on Tianhe-1A for large problem

	128 CPU	%	128 GPU	%	speedup
loop	521.43		167.46		3.1
field	0.63	0.12%	0.66	0.39%	
ion	54.4	10.43%	54.6	32.60%	
shifte	84.6	16.22%	51.8	30.93%	1.6
pushe	365.4	70.08%	44	26.27%	8.3
poisson	4.4	0.84%	4.4	2.63%	
electron other	12	2.30%	12	7.17%	

Platform: Tianhe-1A

- Compute node:
  - 2 Intel Xeon 5670 (6c, 2.93 GHz)
  - 1 NVIDIA Tesla M2050
  - 24 GB DRAM

Results on Tianhe-1A and TitanDev for small test

TitanDev

	32 XE6	%	32 XK6	%	speedup
loop	448		279.7		1.6
field	3.8	0.85%	5.9	2.11%	
ion	44.8	10.00%	81	28.96%	
shifte	74.9	16.72%	61.6	22.02%	1.2
pushe	254	56.70%	70	25.03%	3.6
poisson	10.5	2.34%	11.2	4.00%	
electron other	60	13.39%	50	17.88%	

TitanDev node:

- 16-core AMD 6200 Interlagos
- 1 NVIDIA Tesla X2080 6GB

Tianhe-1A

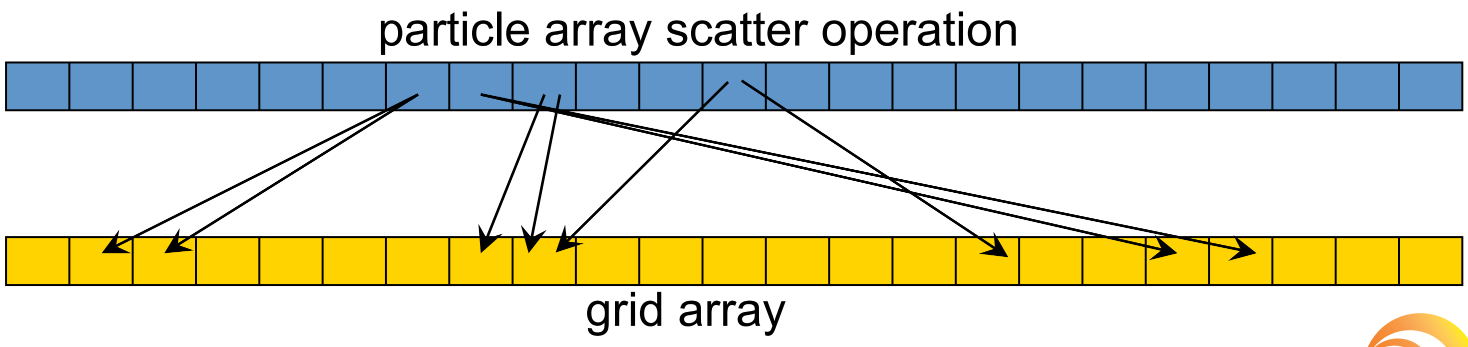
	64CPU	%	32GPU	%	speedup
loop	347.57		218.02		1.6
field	1.57	0.45%	1.32	0.61%	
ion	40	11.51%	64.8	29.72%	
shifte	57.5	16.54%	61.5	28.21%	0.9
pushe	210	60.42%	62	28.44%	3.4
poisson	10.5	3.02%	5.4	2.48%	
electron other	28	8.06%	23	10.55%	

Tianhe-1A Compute node:

- 2 Intel Xeon 5670
- 1 NVIDIA Tesla M2050
- 24 GB DRAM

Optimization Challenges for both CPU and GPU

- "Gather-Scatter" operation in PIC codes
  - The particles are randomly distributed in the simulation volume (grid).
  - Particle charge deposition on the grid leads to indirect addressing in memory
  - Not cache friendly.
  - Need to be tuned differently depending on the architecture.



Keeping up with hardware advances: Close collaboration with the computer scientists at LBL and PSU

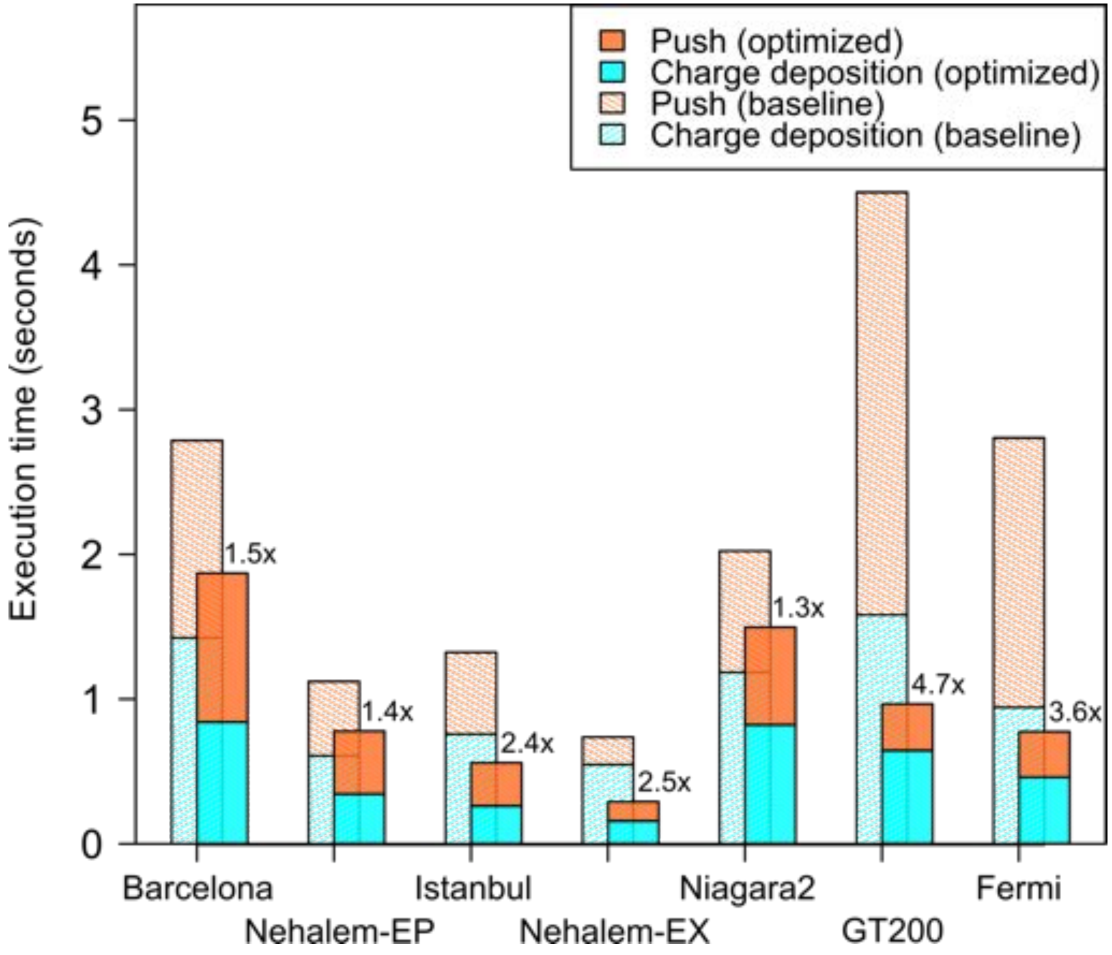
- K. Madduri (now at PSU), K.Z. Ibrahim, S.W. Williams, L. Oliker of the Future Technologies Group (LBNL)
- Advanced node-level optimizations for modern multi-core processors, including GPUs
  - NUMA-aware optimizations
  - Gather-scatter optimizations → locks *vs* atomic operations
  - Thread-base partitioned grid with and without replication
- The basic version of GTC (simple geometry, adiabatic electrons) was entirely re-written in C to allow the use of low level pthread calls and CUDA

Work supported by the DOE Office of Advanced Scientific Computing Research under contract number DE-AC02-05- CH11231

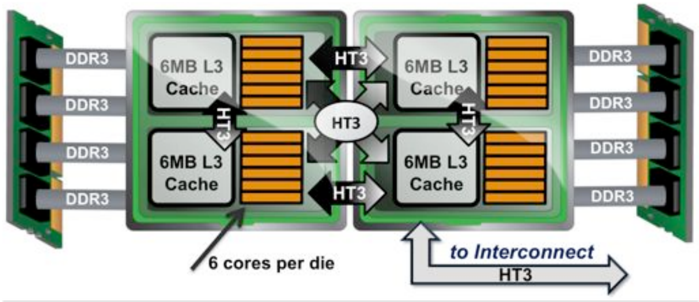
Porting GTC to NVIDIA GPU

- The charge deposition (scatter) step is the main challenge to achieve top performance on GPU (and CPU)
- We depend a lot on atomic updates and their efficiency
  - 64-bit atomic updates implemented with “compare-and-swap”
  - We also tried mixed-precision (32 bits) and fixed point atomics
  - Atomics are relatively slow on the TESLA
  - Will greatly improve on KEPLER!
- Sorting helps improve locality but at a cost
- DIRAC system at NERSC
  - NVIDIA Tesla C2050 (Fermi) 3GB of main memory
  - 2 Quad-core Intel Nehalem processors per node

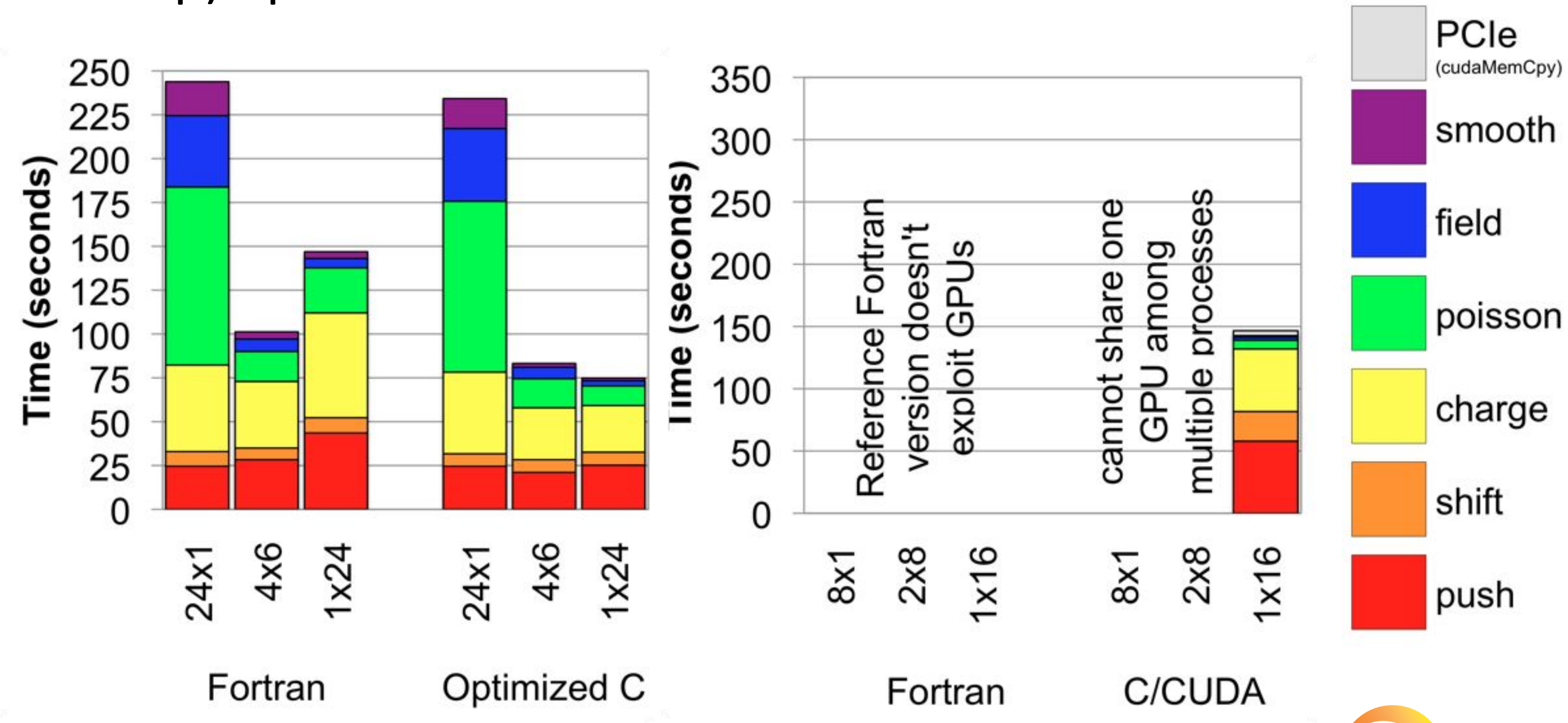
GPU vs. CPU optimizations



Latest work (SC’11)



- 64-bit atomics on GPU implemented with CAS (compare-and-swap) operations



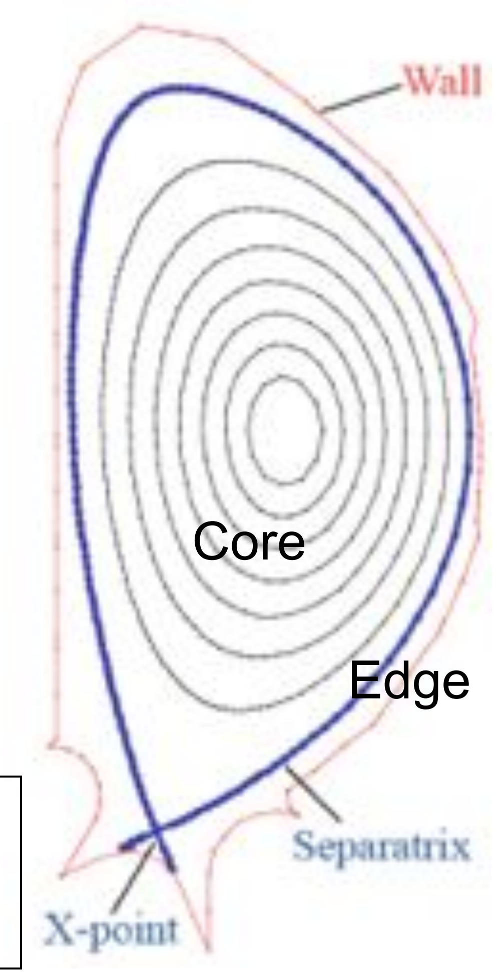
Edge plasma simulation in the fusion gyrokinetic code XGC1 requires extreme scale computing

- **Understanding edge plasma is critical for success of ITER**
  - **Fusion efficiency** is largely determined by plasma confinement at the edge
  - **Life time of material wall** is mostly determined by edge plasma property
- **Edge plasma requires extreme scale computing**
  - Perturbative "delta-f" approach cannot be used in edge
    - Edge plasma is in the **non-equilibrium thermodynamic** state with sources and sinks
    - Dominated by non-linear **self-organization of multi-scale multi-physics**

    → **"full-f" kinetic** simulation is required
  - The popular and economical "magnetic" coordinate system has singularity on separatrix

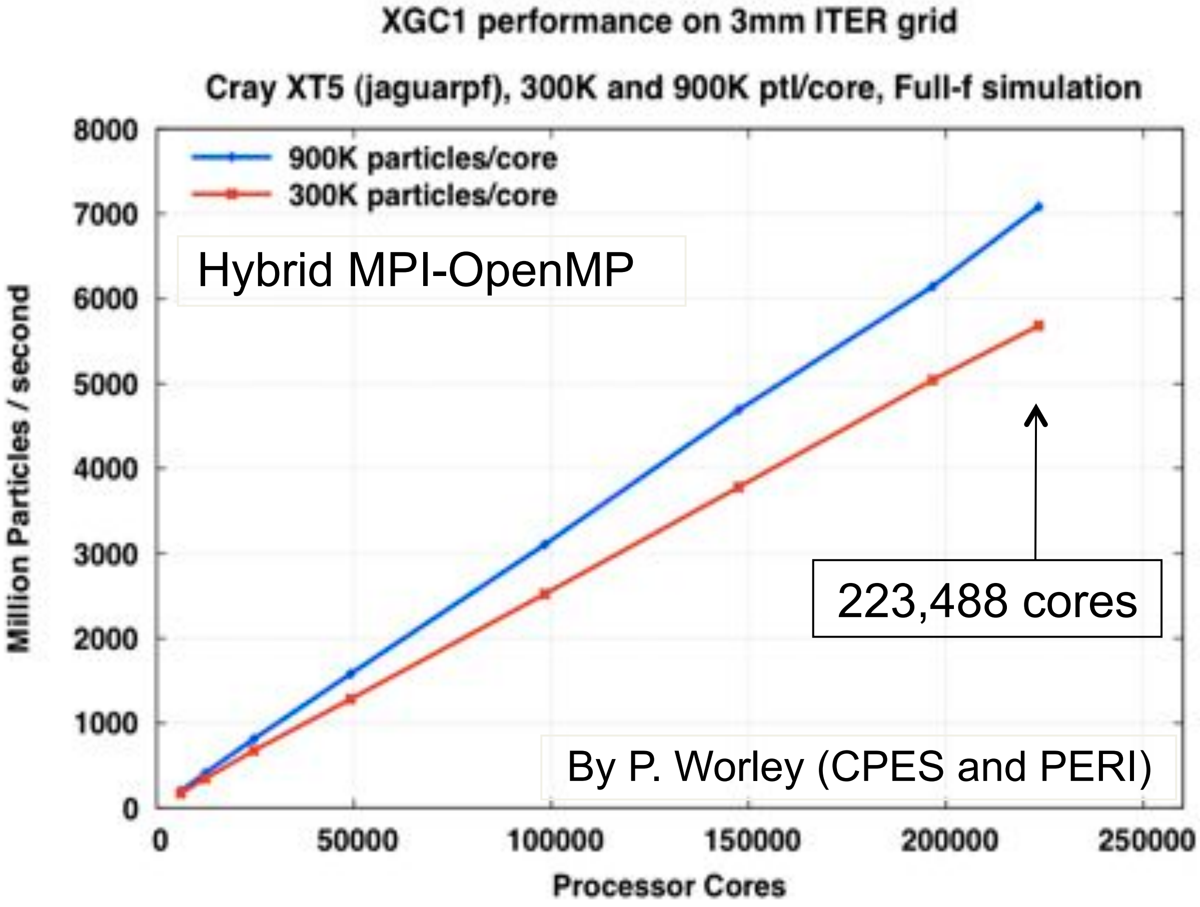
    → **Cylindrical coordinate system** is used
  - Complicated edge geometry demands **unstructured triangular grid**

Presently, XGC1 is the only gyrokinetic code in the world fusion program to include the magnetic separatrix surface.



CPES
Center for Plasma Edge Simulation

A fusion gyrokineic code XGC1 scales efficiently to the maximal number of Jaguar XT5 cores



Most of the XGC1 productions runs have been >70% Jaguarpf capability.

- **Utilized strong collaboration with past ASCR Centers and Institutes** (TOPS, SDM, PERI, ULTRAVIS, Courant Math and Computing Lab)
- **Strong collaboration with the new ASCR Centers is essential to continue the scalability to hybrid extreme computing** (FastMath, SUPER, QUEST, SDAV)

Extreme scale computing of XGC1 enables realistic tokamak edge physics simulation

Study Multi-scale multi-physics self-organization of edge physics and its impact on core confinement

•Large scale background profile evolution

•Small scale microturbulence

•Meso scale zonal flow dynamics

•Neutral fuel particles and impurity particles with atomic physics

▪Coarse grained mesh is needed all the way to the core to study edge effect on core confinement, and to avoid artificial core-edge boundary.

✧**Validation on present devices**

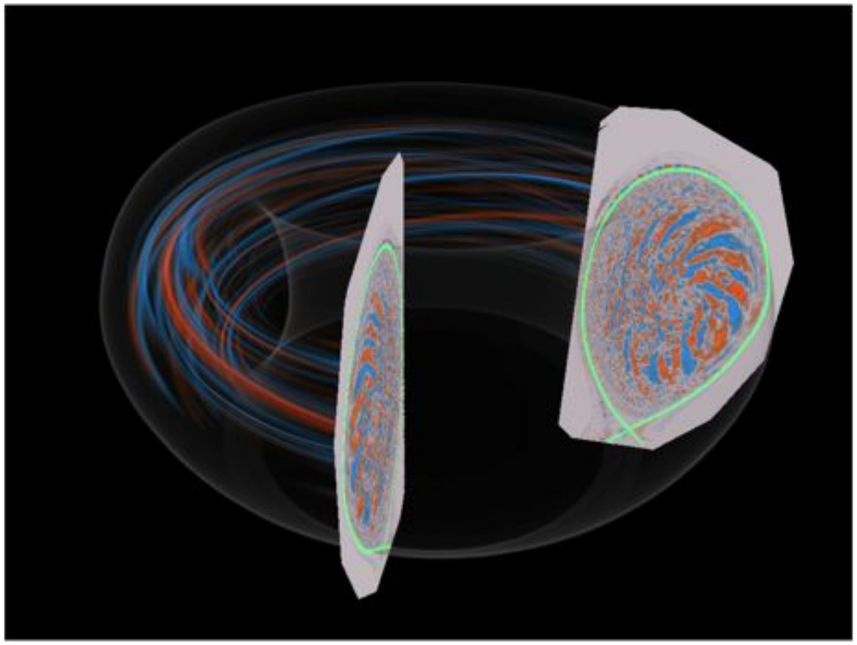
DIII-D & NSTX (US) **JET (EU)**
~ pFlop Jaguar and Hopper **Early Titan**

✧**Prediction for ITER**

>10 peta flops, Titan and beyond

✧**More physics capability with stronger computing power**

✧**Developed GPU optimization tech.**



XGC1 simulation in CPES (vis by K. Ma): Spreading of edge turbulence into DIII-D core, shedding light on the 30 year old mystery on how the edge plasma affects the core confinement so strongly. This experimental fact, without theoretical understanding, became the basis for ITER performance prediction. 170K Jaguar cores used for one day.

XGC1 GPU porting effort

- Preliminary study led by Sanjay Ranka from U. of Florida
- Current effort led by J. Cummings of Caltech
- What makes XGC1 different?
  - Full-f code → ~ 5000 particles per cell!
  - New algorithm for charge deposition using a velocity-space grid to reduce the number of 4-point averages from 5000 per cell to ~ 10

Conclusion

- Gyrokinetic PIC codes are a powerful tool to study wave-particle interactions in fusion plasmas
- The simulations are very demanding so it is important to keep up-to-date with the latest hardware and software optimizations
- Hybrid architecture can have a significant impact on improving performance of GKPIC codes
- In order to achieve this performance it is highly beneficial to collaborate with computer scientists and applied mathematicians!