Heterogeneous Programming and Optimization of Gyrokinetic Toroidal Code Using Directives

GTC CAAR project:

Wenlu Zhang[1,2], Wayne Joubert[3], Peng Wang[4], Matthew Niemerg[5], Bei Wang[6], William Tang[6], Sam Taimourzadeh[1], Lei Shi[1], Jian Bao[1], Zhihong Lin[1]

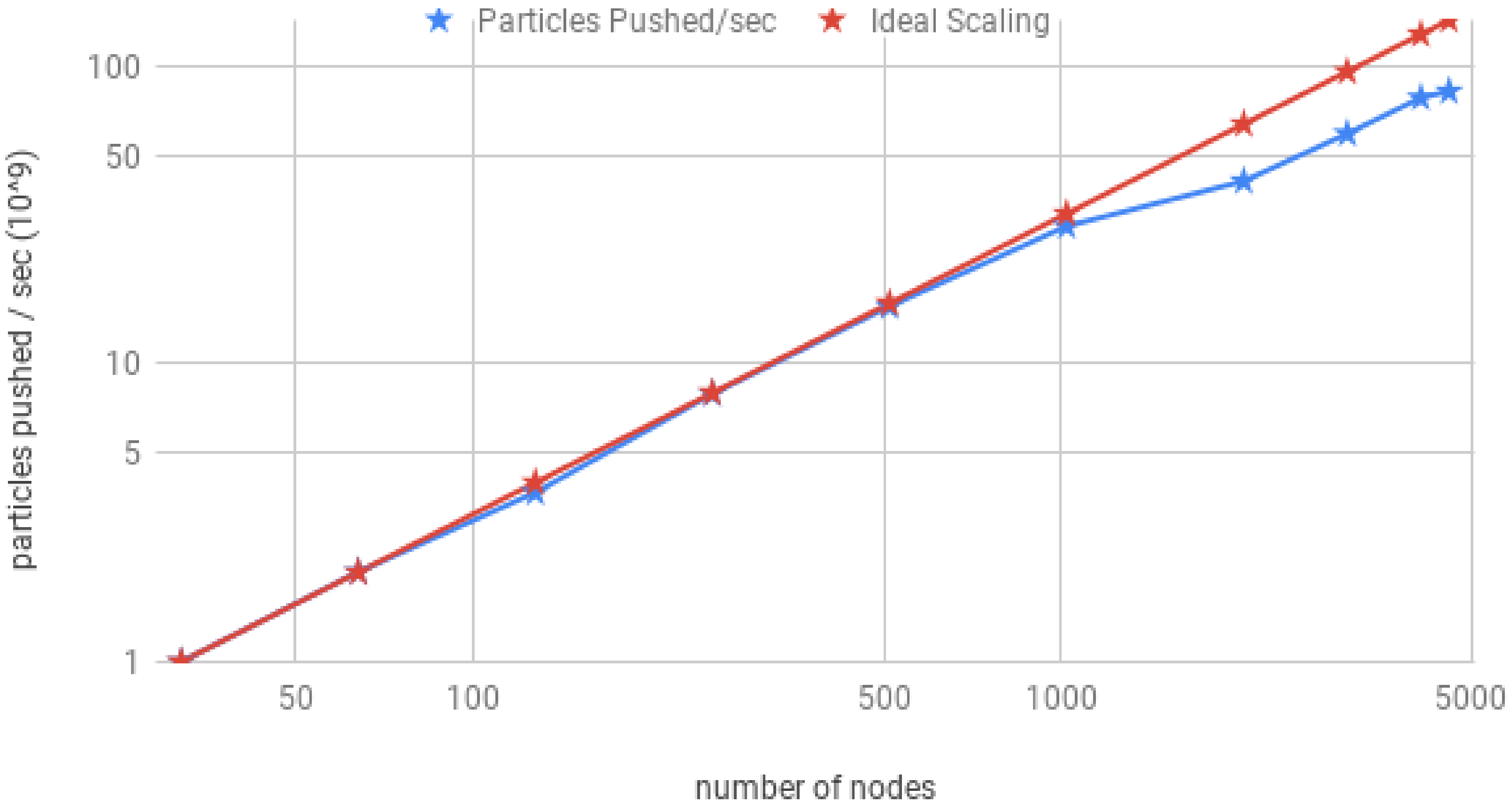
[1] University of California, Irvine, California, USA
[2] Institute of Physics, Chinese Academy of Sciences, Beijing, China
[3] Oak Ridge National Lab, Oak Ridge, TN, USA
[4] NVidia, USA
[5] IBM, USA
[6] Princeton University, Princeton, NJ, USA

Presenter: Zhihong Lin, University of California, Irvine

SciDAC ISEP Center and GTC team

Outlines

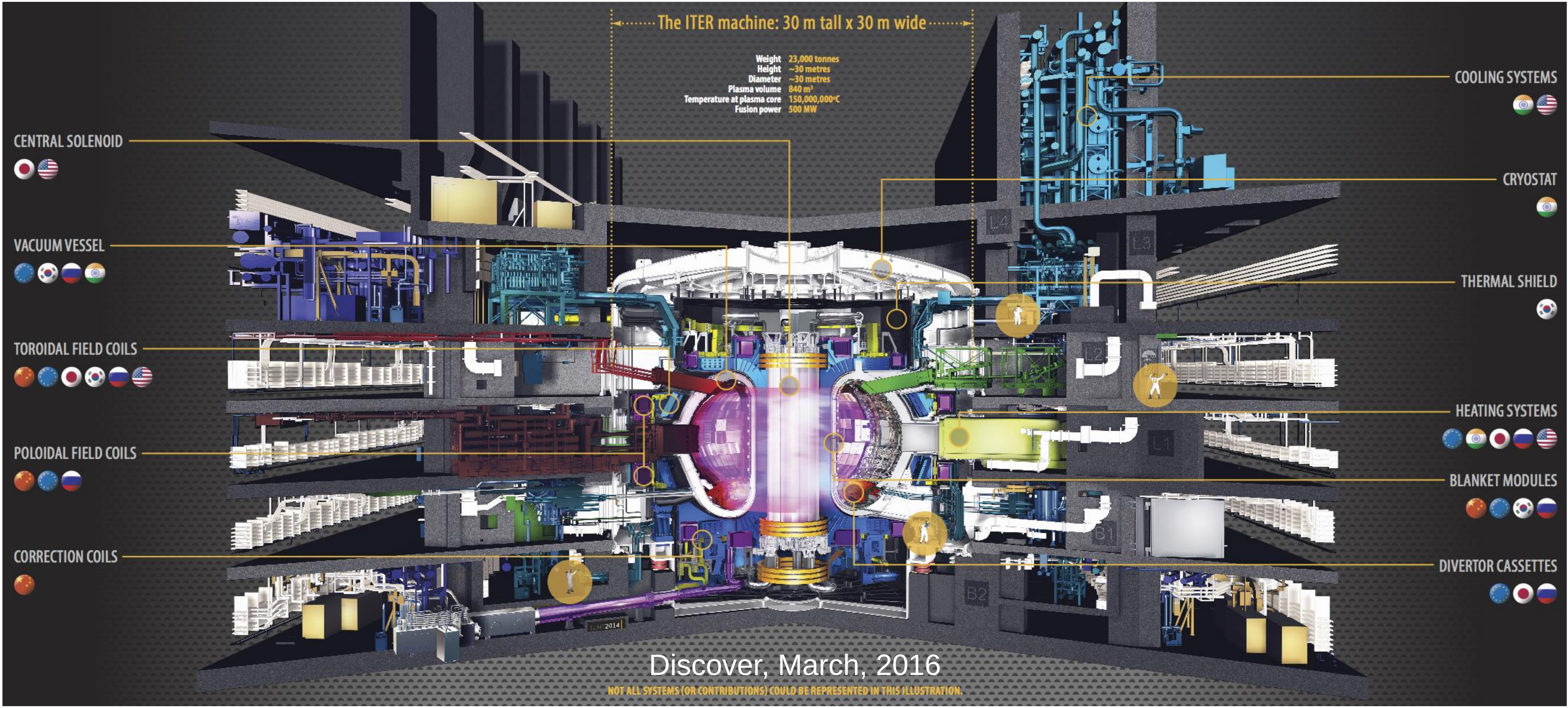
- Introduction to GTC
- Porting and optimization on Titan using OpenACC
- Optimization on Summit
- Conclusions



GTC weak scaling on Summit, W. Joubert

Fusion will provide clean, unlimited energy source
ITER is crucial next step in the quest of fusion energy

$25B collaboration: China, EU, India, Japan, Korea, Russia, USA

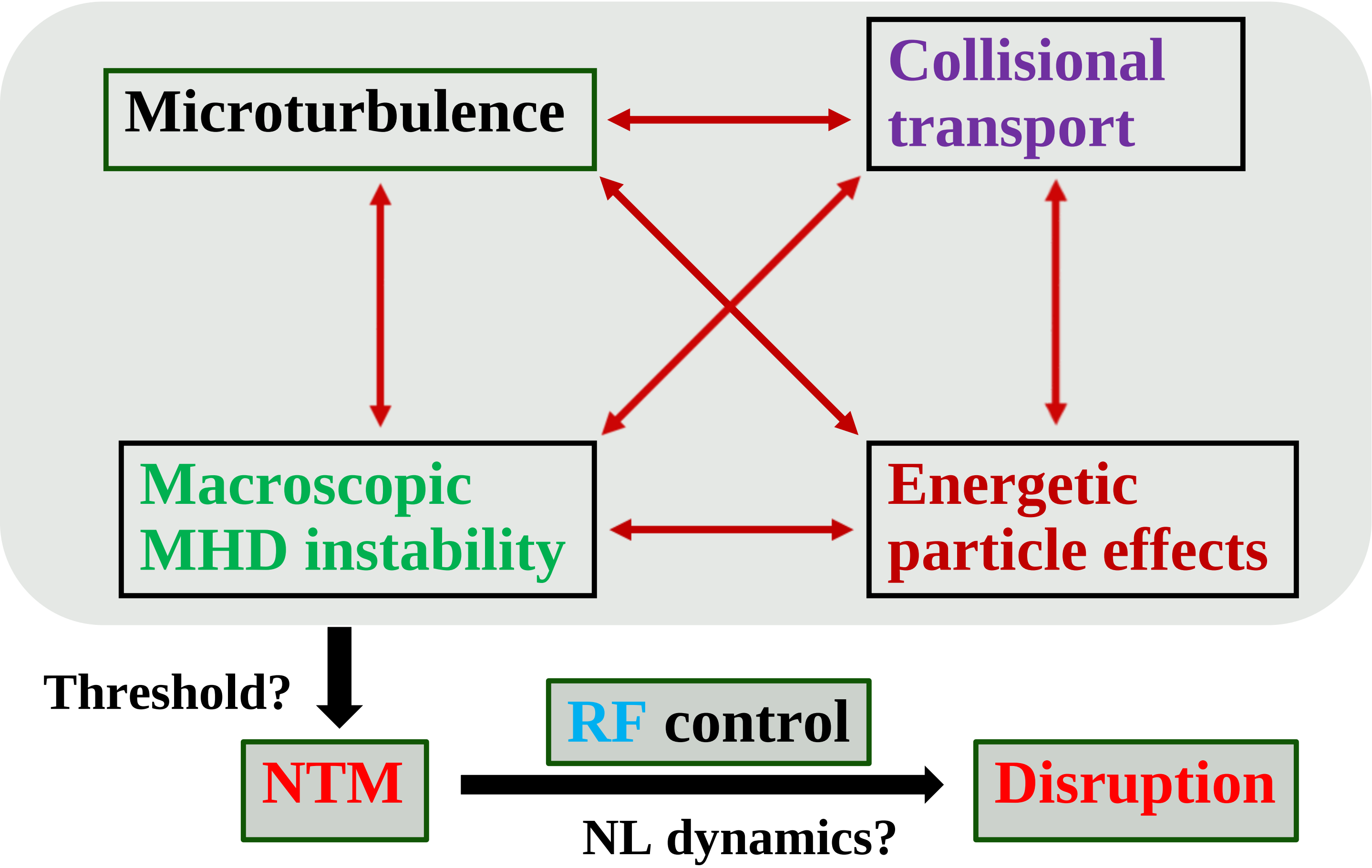


Discover, March, 2016

Predictive simulation is needed for ITER burning plasmas

- Simulation of plasma confinement and stability are required before each ITER experiment
- Since ignition in ITER relies on self-heating by energetic fusion products (α-particles), confinement of energetic particles (EP) is a critical issue for ITER
- Plasma confinement properties in the *new* ignition regime of self-heating by α-particles is one of the most uncertain issues when extrapolating from existing fusion devices to ITER
  - EP transport by meso-scale EP instabilities
  - Interaction between EP with microturbulence responsible for thermal plasma transport and macroscopic magnetohydrodynamic (MHD) instabilities potentially leading to disruptions
- SciDAC ISEP: integrated simulations of EP turbulence by treating relevant physical processes from micro to macro scales on same footing
  - ISEP Center: UCI, GA, PPPL, ORNL, LBNL, LLNL, PU, UCSD, UT

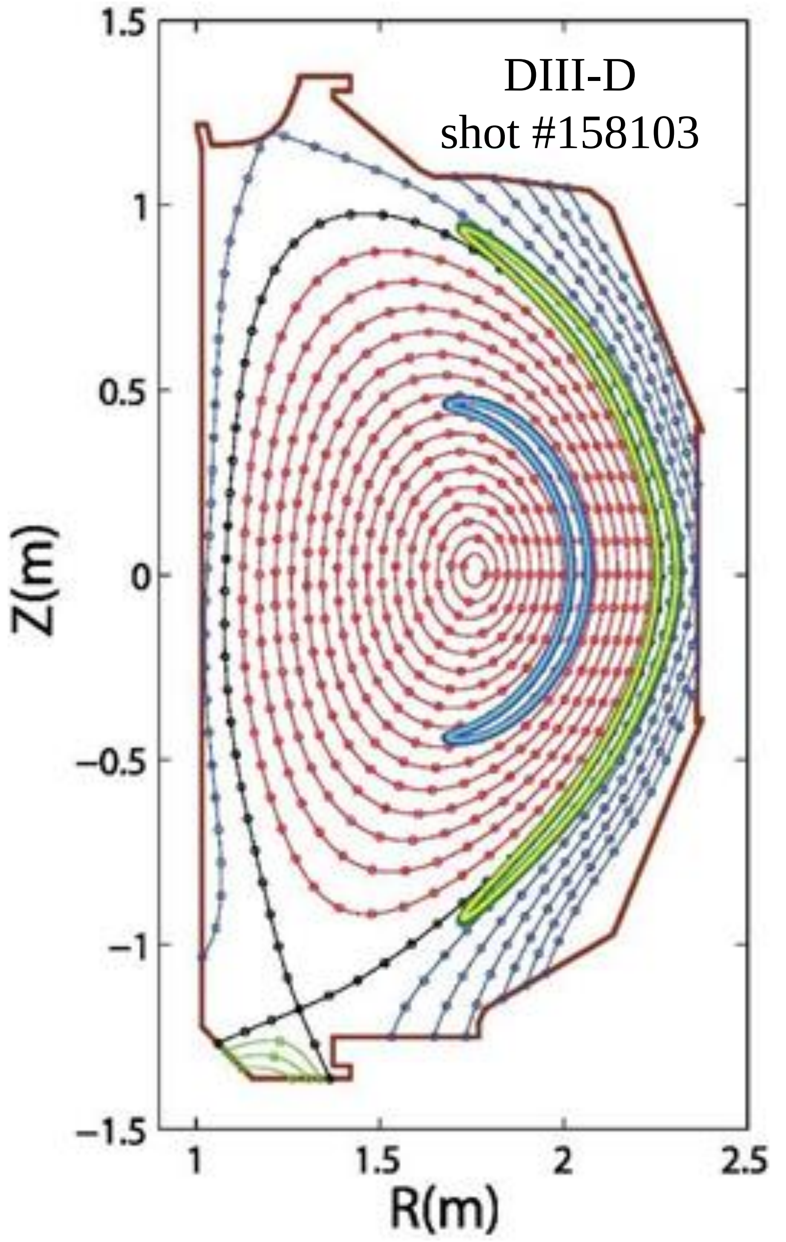
Predictive capability requires integrated simulation for nonlinear interactions of multiple kinetic-MHD processes



- Neoclassical tearing mode (NTM) is the most likely instability leading to disruption
- NTM excitation depends on nonlinear interaction of MHD instability, microturbulence, collisional transport, and EP effects. NTM control requires radio frequency (RF) waves

Gyrokinetic Toroidal Code (GTC)

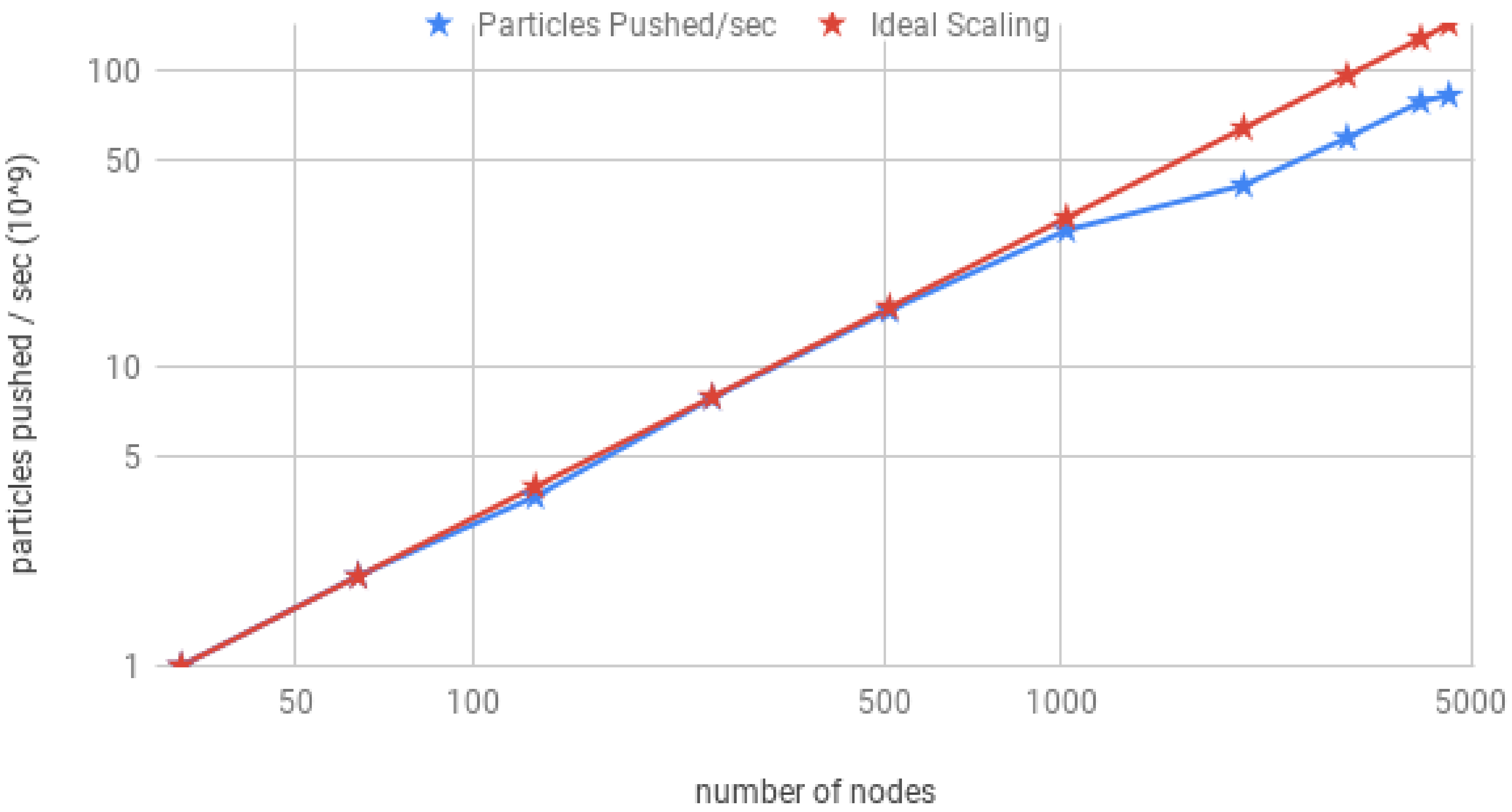
- First-principles, global, integrated simulation capability for nonlinear interactions of multiple kinetic-MHD processes
- Current physics capability
  - ✓ Global 3D toroidal geometry for tokamak, stellarator, FRC
  - ✓ Microturbulence: 5D gyrokinetic ions & electrons, electromagnetic compressible fluctuations, collisionless/collisional tearing modes
  - ✓ MHD and energetic particle (EP): Alfven eigenmodes, kink, resistive tearing modes
  - ✓ Neoclassical transport: Fokker-Planck collision operators
  - ✓ Radio frequency (RF) waves: 6D Vlasov ions
- Adapted as ISEP framework by SciDAC ISEP
- Key code in ITER-China fusion simulation project
- Large user community (>40 users/developers); Broad impacts to fusion (12 papers in *PRL*, *Science*, *Nature Comm.*)



Open source:
Phoenix.ps.uci.edu/GTC

Outlines

- Introduction to GTC
- Porting and optimization on Titan using OpenACC
- Optimization on Summit
- Conclusions

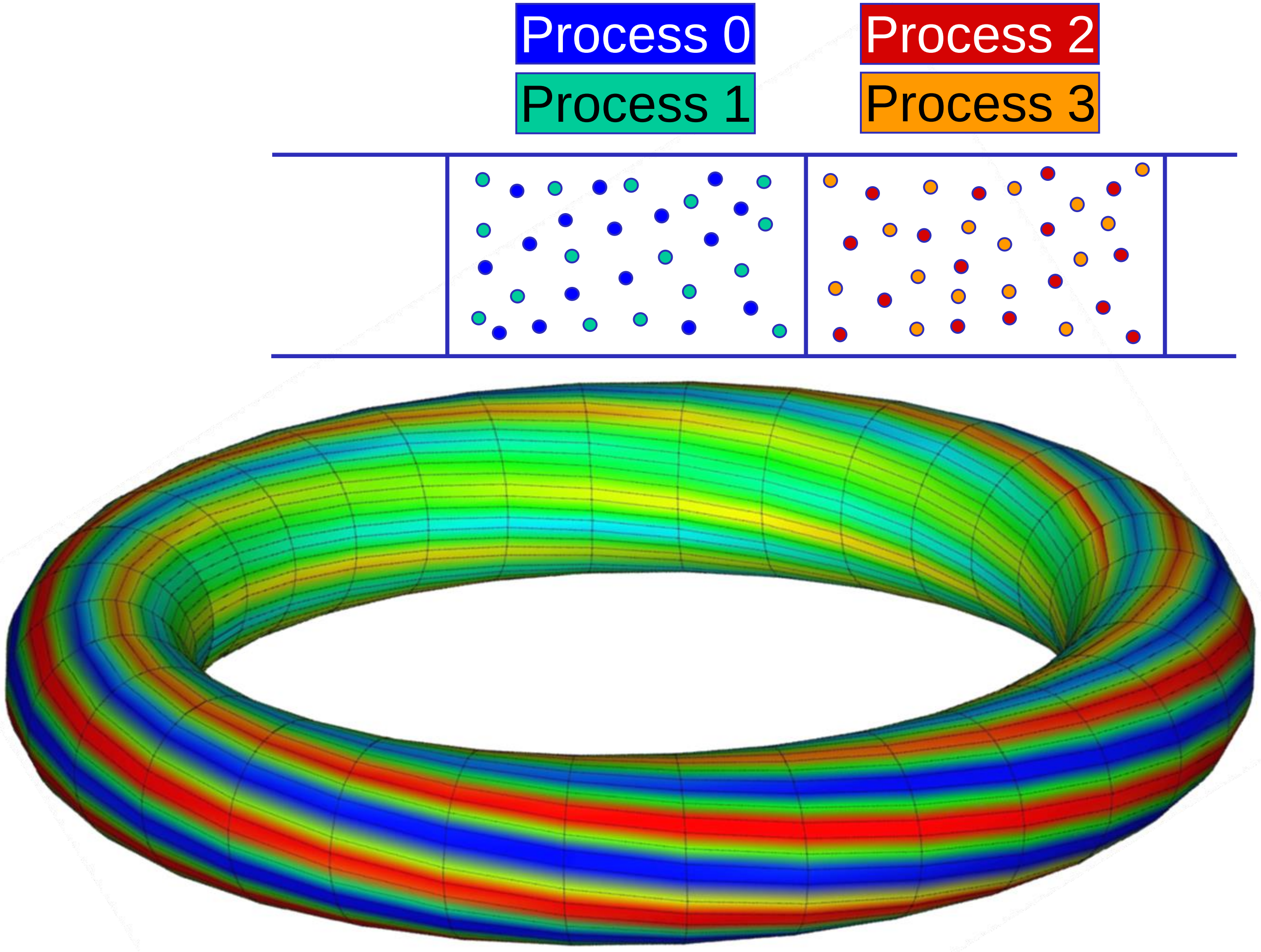


GTC weak scaling on Summit, W. Joubert

GTC Multi-level Parallelization on CPU

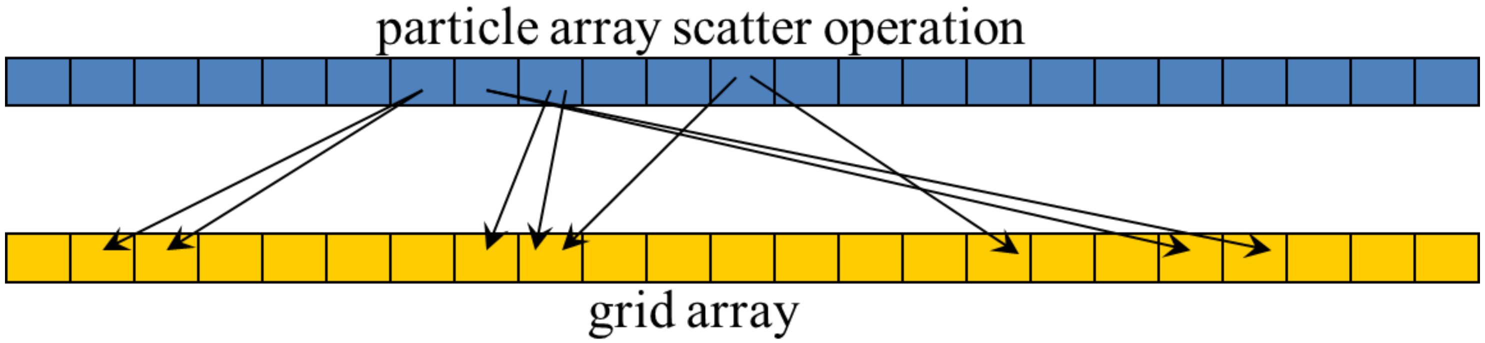
- Multi-physics model: particle-in-cell (PIC) and fluid models solved together
- PIC Simulation: particles in 5D/6D phase space; Electromagnetic fields and fluids on 3D fixed grids
- Multi-level parallelization
  1. MPI domain-decomposition (1D) for particle-field interactions
  2. MPI particle-decomposition with series/parallel fields/fluids solvers (typically 100-1000 particles per grid)
  3. Loop level parallelism using OpenMP
- Porting to GPU with same MPI parallelization; replace OpenMP with OpenACC

Particle-tight MPI mapping:
good for gather-scatter operations



Porting GTC to Titan GPU using OpenACC

- Use one MPI per GPU, move most computing-intensive particle and field data to GPU
- Restructured for unified subroutines for particles
  - push(species_name, and other parameters): gather fields on particles from grids
  - charge(species_name): scatter particle charge and current to grids
  - shift(species_name): send particles to their MPI domain
- Charge subroutine: scatter operations on GPU use $acc atomic update
  - CPU version use work-vector method: each register has a private copy of local grids
- Shift subroutine rarely changed: use optimized CUDA version
  - CPU version used a sequential implementation. So cannot apply OpenACC directly.
  - Algorithm redesign using CUDA



Particle optimization: binning, texture cache

- Particle binning: rearrange particle arrays every several time steps (not much overhead in sorting), so physically close particles stay close in memory => depends on same/similar section of field array significantly enhance data locality in particle pushes => maximize GPU “cache” reuse
- Enable texture cache on Titan Kepler GPU for grid arrays in gather operation leads to 3X speedup; Summitdev Pascal GPU & Summit Volta GPU unify texture/L1 cache
- Both Array of Structure (AoS) and Structure of Array (SoA) data layouts for particles have been implemented on GTC-P. Performance analysis using NVPROF on Titan shows no significant speedup is obtained with SoA. We thus decide to continue using AoS layout for all particle species

Push subroutine: careful mapping for local memory

Most time consuming loop in gather operations in push subroutine is related to 2D/3D spline function

eqdata.F90:

```
spdim=27 or 9

push.F90:
real dx(27)
!$acc parallel loop private(dx)
do m=1,me
   do ii = 1, spdim
      dx(ii)=zpart(1,m)...
      …
   enddo
   …
enddo
!$acc end parallel
```

Compiler doesn't know the index of dx.
Private storage in global memory for each thread.
Leads to uncoealesced access

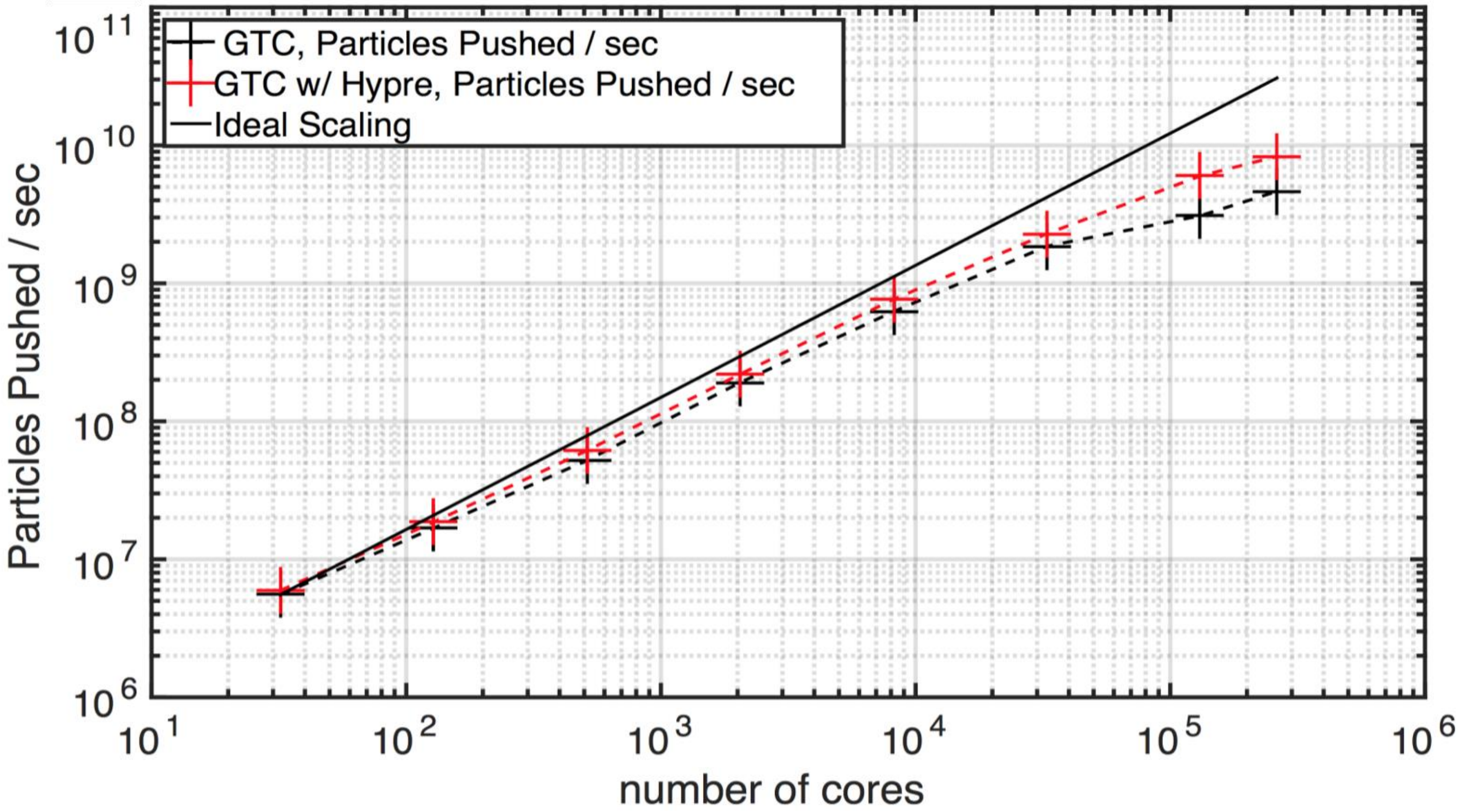
module.F90:

```
spdim=27 or 9

push.F90:
real dx(spdim)
!$acc parallel loop private(dx)
do m=1,me
   do ii = 1, spdim
      dx(ii)=zpart(1,m)...
      …
   enddo
   …
enddo
!$acc end parallel
```

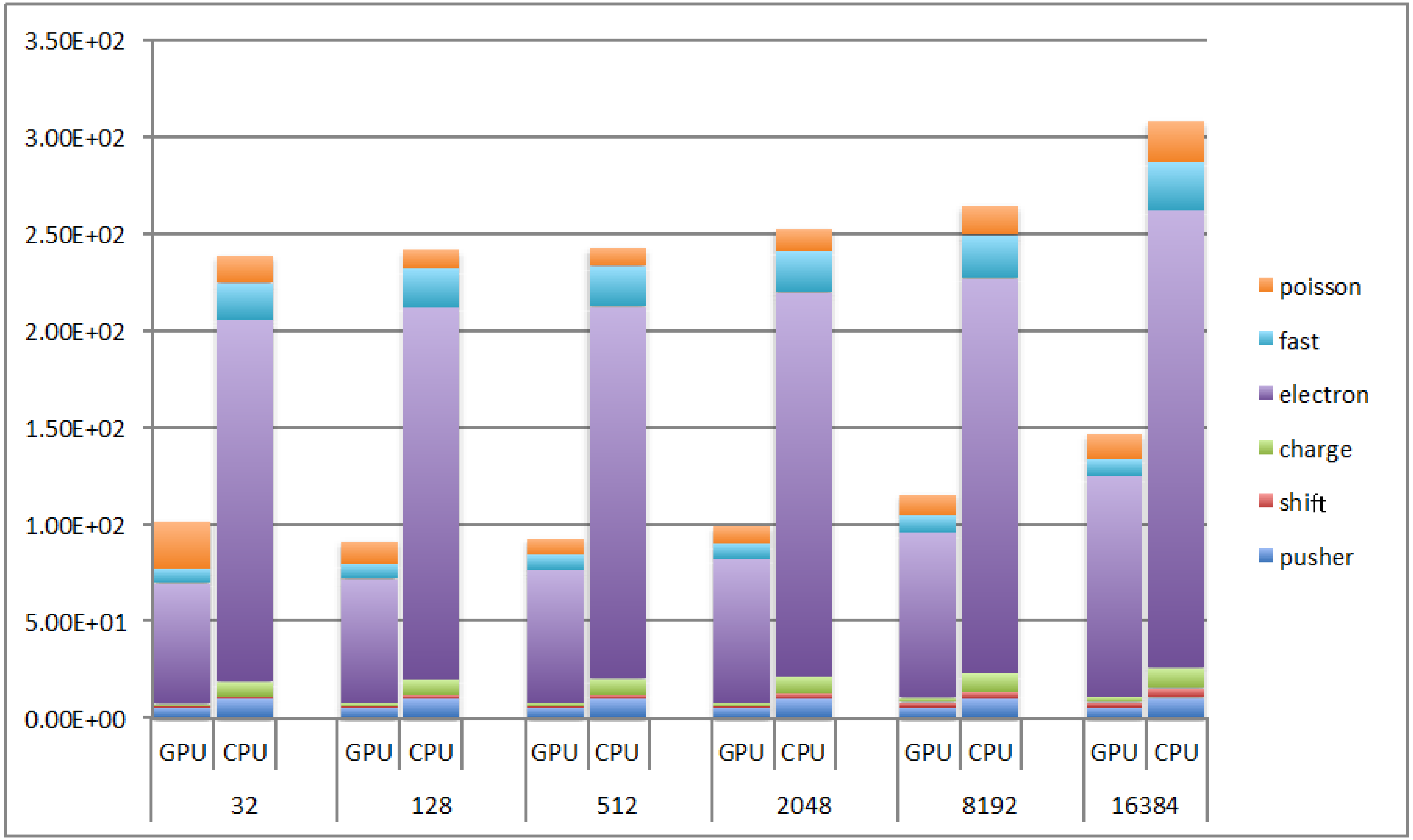
Compiler knows the index.
Can put into register or local memory.
4X speedup in this loop

GTC Weak Scaling on Titan



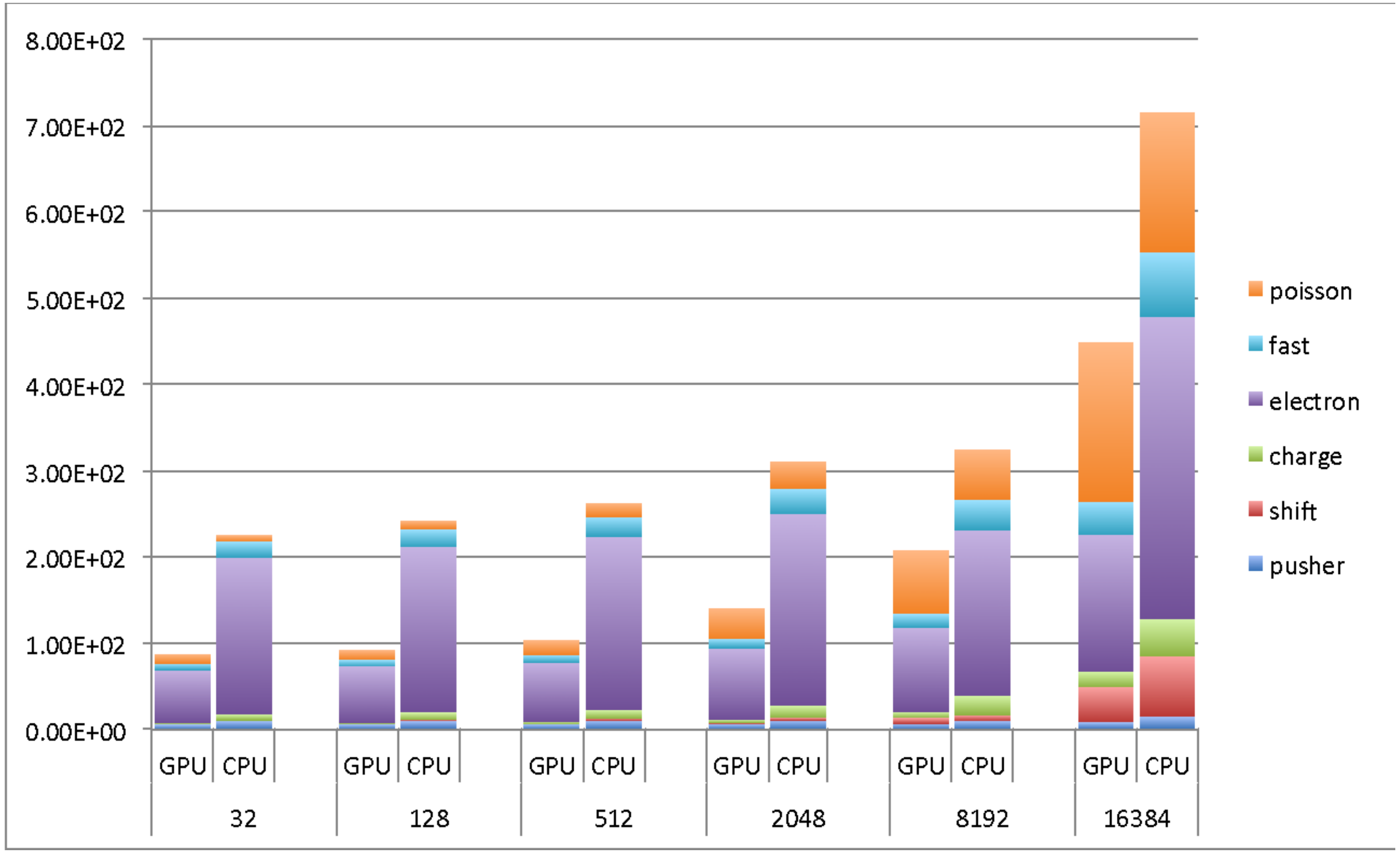
- EP physics simulation [Z. X. Wang et al, PRL2013] of most advanced US fusion device DIII-D using 3 species: electron, thermal & fast ions
- Both grid and particle numbers per core remain constant
- GPU (NVIDIA K20x) achieves 3x speedup from CPU (16 cores AMD 6274)

Timing Breakdown for Weak Scaling Test



- Electron is most compute intensive
- Decrease of performance in large processor counts is mainly due to increased portion of non-GPU accelerated subroutines as well as MPI time

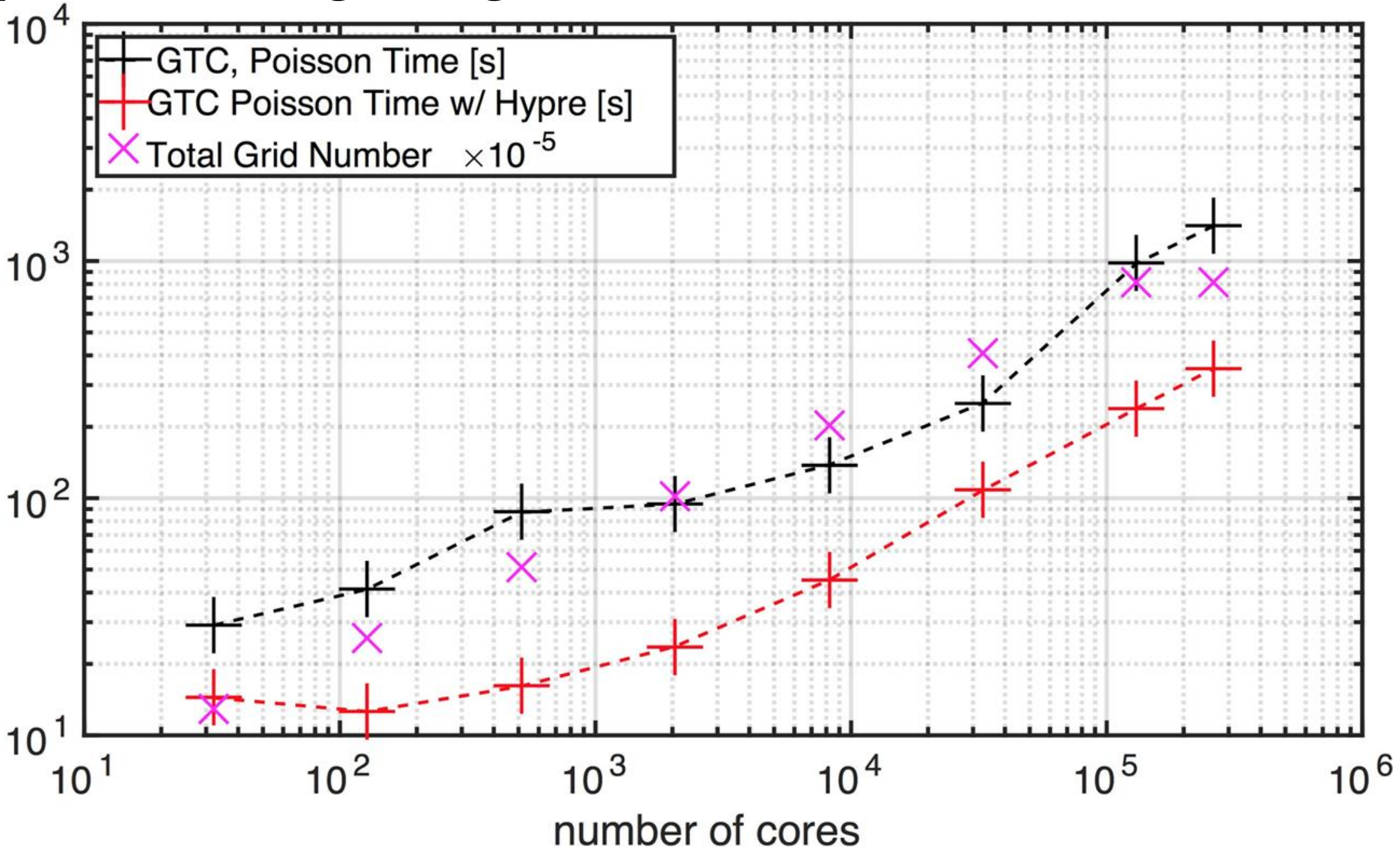
Timing Breakdown for Hybrid Weak Scaling on Titan



- Fusion device size increases from existing DIII-D to future larger ITER
- Grid number is proportional to square root of node number
- Number of particles per MPI is fixed
- Fields solver became a bottleneck after all particles ported to GPU

Sparse Matrix Solver ported to Summitdev GPU

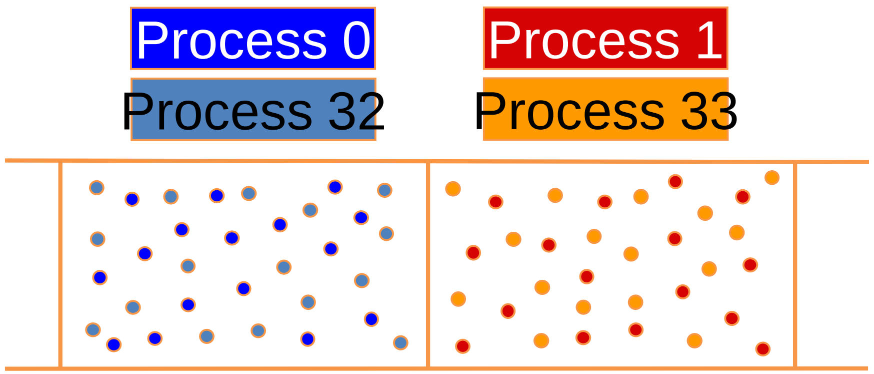
- Hypre algebraic multi-grid solver 11X faster than PETSc standard solver on Summitdev
- Field solver ported to GPU: NVIDIA AmgX solver 27X faster than PETSc
- Restrict GPU number used in AmgX: more GPU <==> more communications
- Fluid subroutines: advance field quantities => lightweight, done on CPU

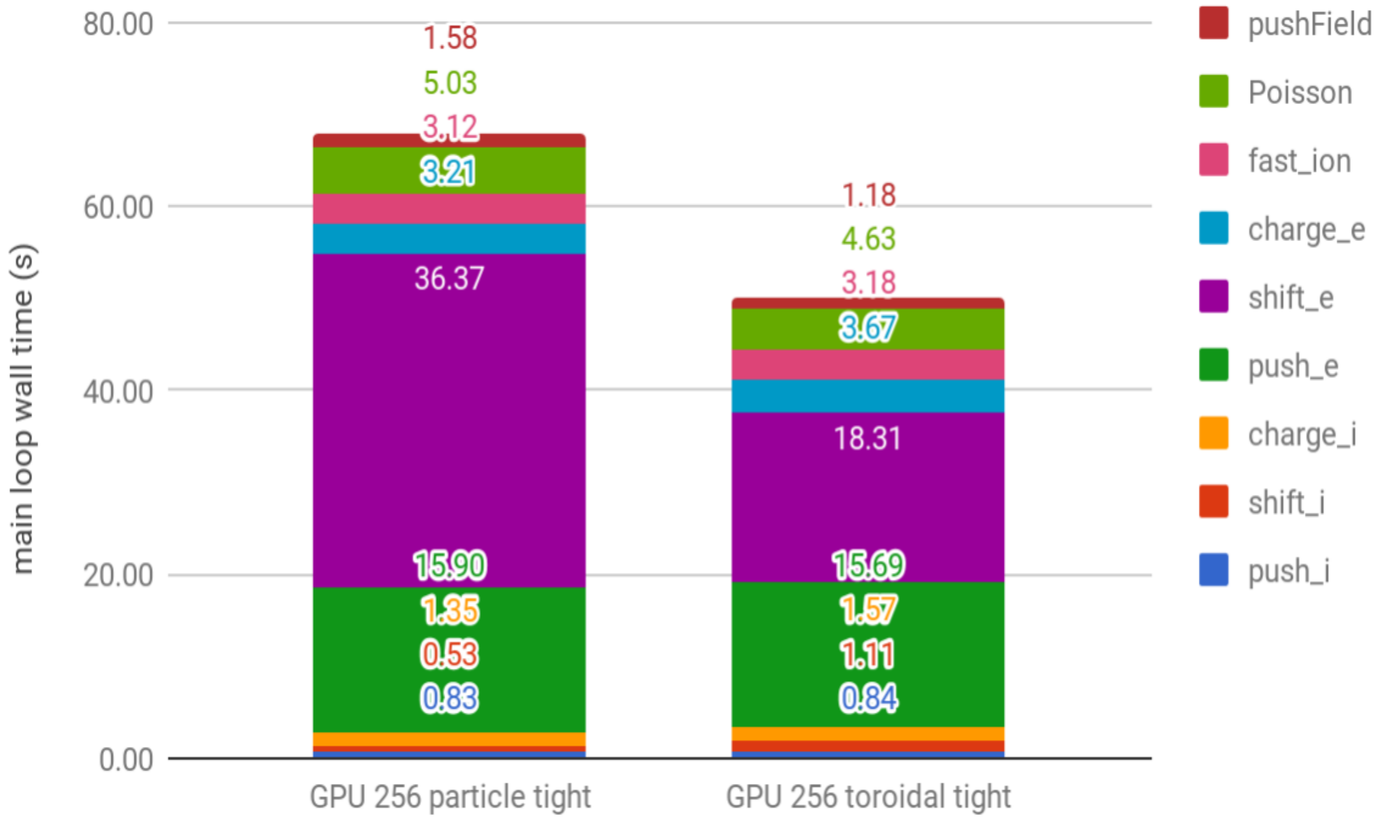


Optimize MPI mapping for shift on Summit

- Two MPI communicators in GTC:
  - ✓ Toroidal communicator: MPI ranks with the same particle domain ID, but different toroidal domain IDs
  - ✓ Particle communicator: MPI ranks with the same toroidal ID, but different particle domain ID
- Particle-tight mapping: not good for MPI_SendRecv in shift subroutine
- Change to toroidal-tight mapping: shift_e time is reduced by 2X

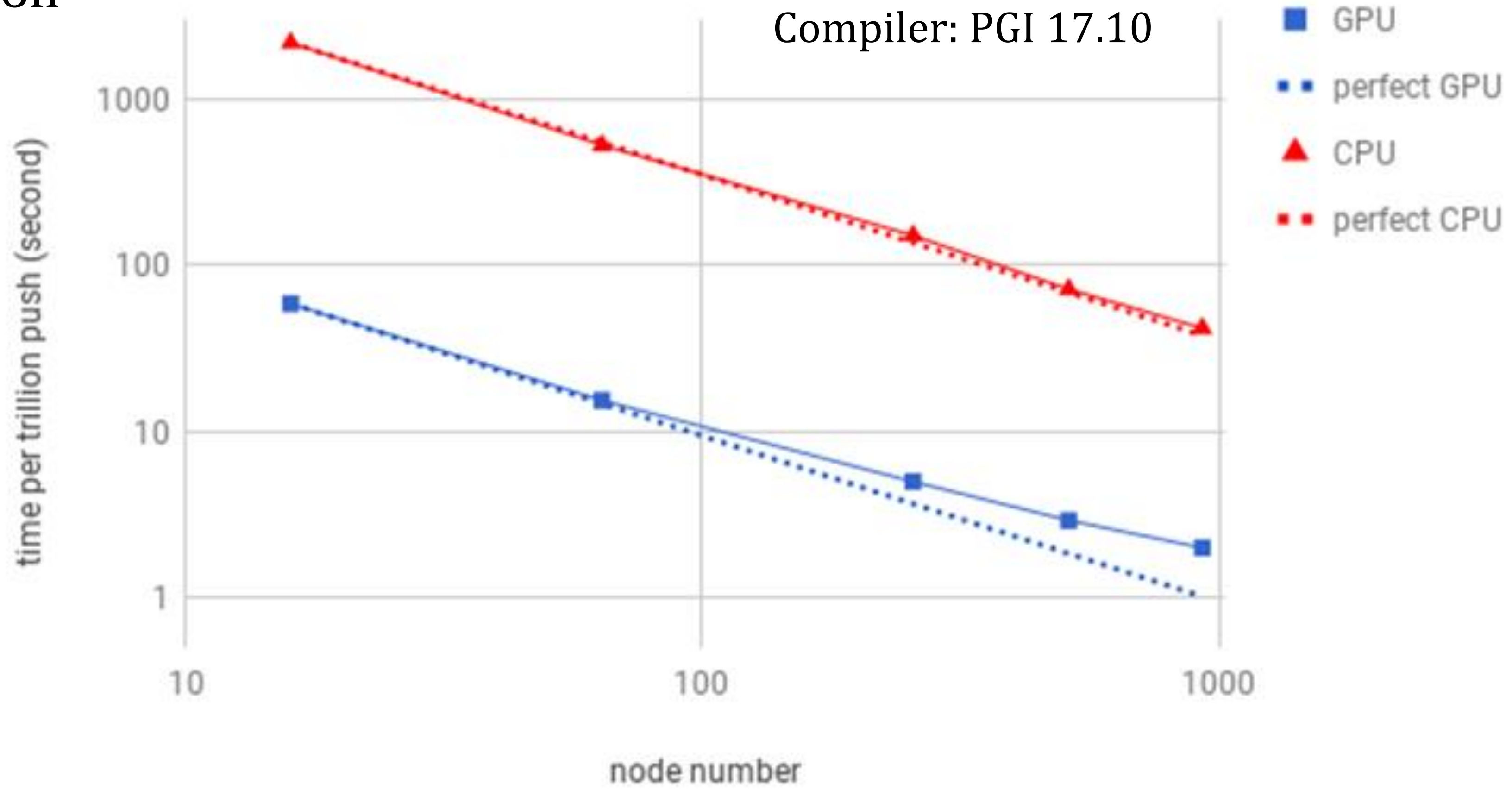
Toroidal-tight MPI mapping: good for MPI_SendRecv in shift





GTC Performance on Summit GPU

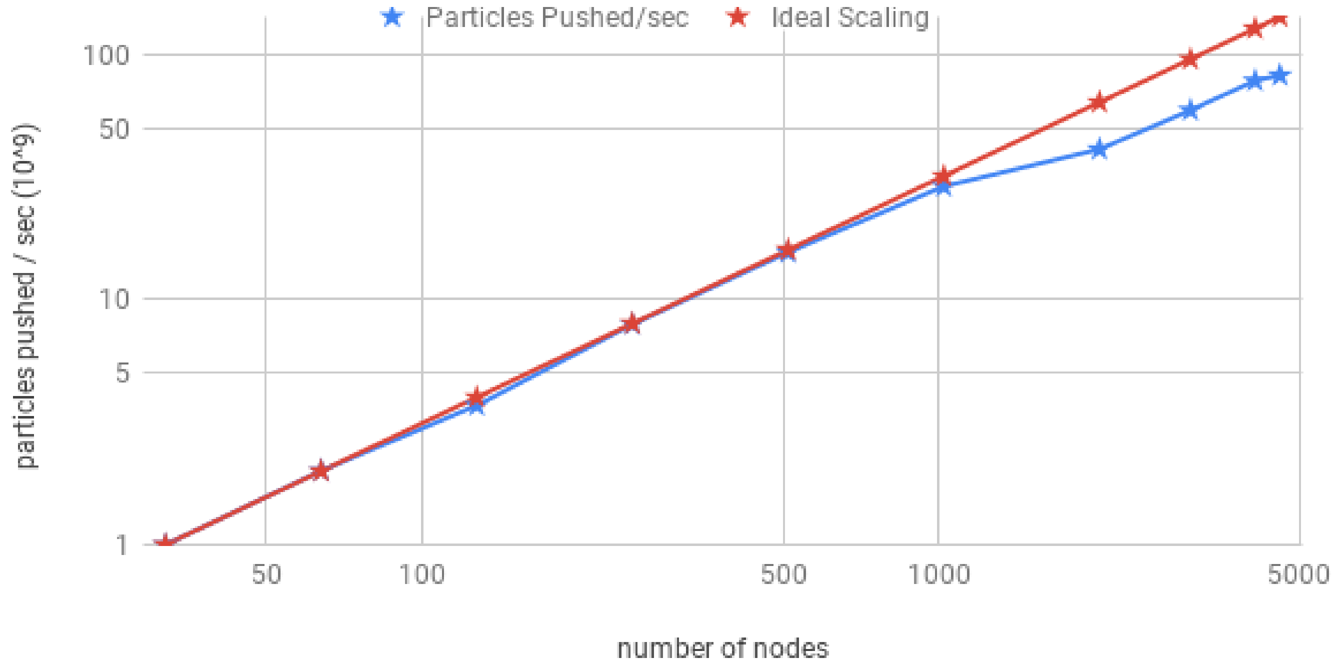
- Speeds up 37X from CPU to GPU on 384 GPUs; 20X on 5556 GPUs (1/5 of SUMMIT)
- Recently selected by *NVIDIA* as Top 15 App Worldwide
- Summit early science application



Wall-clock time for one trillion particle pushes in GTC hybrid weak scaling test on Summit

Conclusions and Plan

- GTC fully optimized and scaled up to whole Summit
- OpenACC directives: good performance on GPU and ease of maintenance by application users
- Memory management key to GPU performance
- Other completed work: FFT in fluid ported to GPU, ADIOS implemented in GTC
- GTC on Tianhe-3 prototype (Phytium ARM) using OpenMP 4.0 in GCC compiler
- SciDAC ISEP partnership (R. Falgout, S. Klasky, S. Williams, W. Tang): ISEP framework portability and optimization



GTC weak scaling on Summit, W. Joubert