Evolving the GTC plasma PIC simulation for Mira

Bei Wang[1] Stephane Ethier[2] William Tang[1,2]

[1]Princeton Institute for Computational Science and Engineering, Princeton University

[2]Princeton Plasma Physics Laboratory, Princeton University

Mira Community Conference
Argonne Leadership Computing Facility, Chicago

Mar 6, 2013

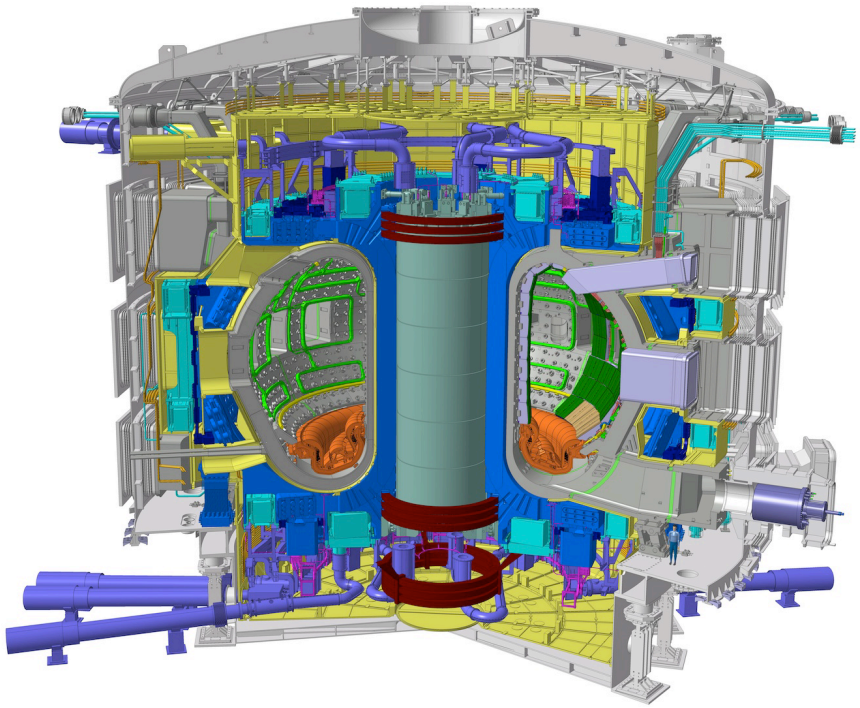
Outline

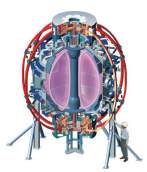
Collaborators

- Tim Williams, Catalyst in Argonne Leadership Computing Facility
- Khaled Ibrahim, Kamesh Madduri (now at Penn State Univer.), Sam Williams, Leonid Oliker of the Future Technologies Group at Lawrence Berkeley National Laboratory

The Challenge: Global Simulation of ITER

- ✓ This is extremely large compared to current experiments
- ✓ We need advanced computational method, mathematical model and high performance computing to predict its performance
- ✓ Very large simulation





Gyrokinetic Simulations: for studying turbulent transport

- Macroscopic Stability

  What limits the pressure in plasmas?

- Wave-particle interactions

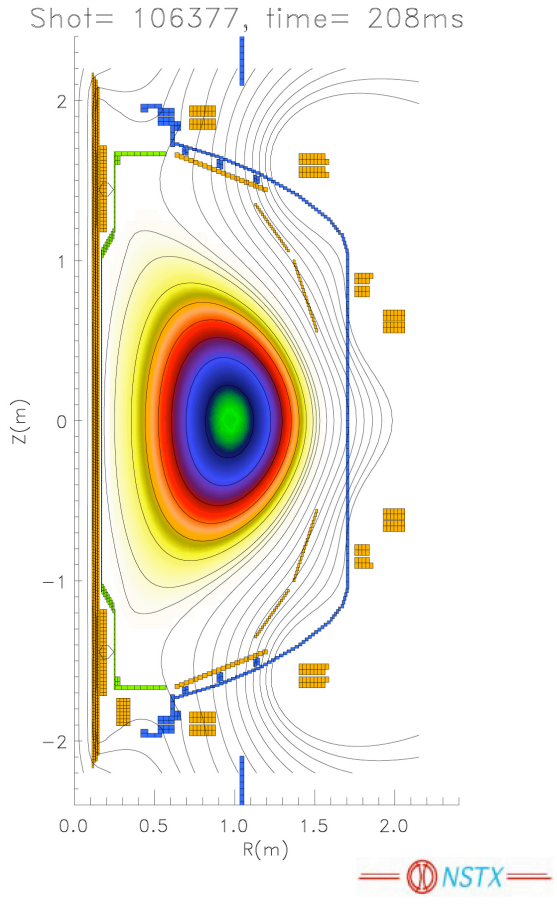
  How do particles and plasma waves interact?

- **Miroturbulence and Transport**

  What causes plasma transport?

- Plasma-material Interactions

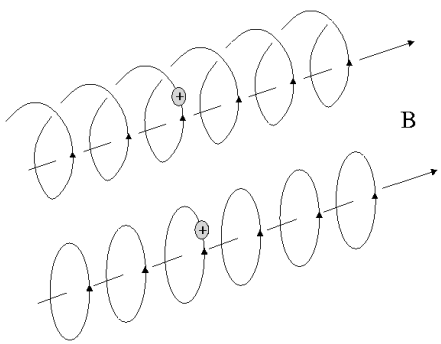
  How can high-temperature plasma and material surfaces co-exist?



Gyrokinetic Vlasov-Poisson Equation

We consider kinetic ions and adiabatic electrons. The dynamics of the ions is described by the 5D gyrophase-averaged Vlasov equation in toroidal geometry.

- gyro-motion: guiding center drifts + charged ring



- suppress plasma oscillation and motion
- larger time step and grid size

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{d\mathbf{R}}{dt} \cdot \frac{\partial f}{\partial \mathbf{R}} + \frac{dv_{\parallel}}{dt} \frac{\partial f}{\partial v_{\parallel}} = 0$$

$$\frac{d\mathbf{R}}{dt} = v_{\parallel} \hat{\mathbf{b}} + \mathbf{v}_d - \frac{\partial \bar{\phi}}{\partial \mathbf{R}} \times \mathbf{b}$$

$$\frac{dv_{\parallel}}{dt} = -\mathbf{b}^* \cdot \left(\frac{v_{\perp}^2}{2} \frac{\partial}{\partial \mathbf{R}} \ln B + \frac{\partial \bar{\phi}}{\partial \mathbf{R}} \right)$$

$$\mathbf{b}^* = \hat{\mathbf{b}} + v_{\parallel} \hat{\mathbf{b}} \times (\hat{\mathbf{b}} \cdot \frac{\partial}{\partial \mathbf{R}}) \hat{\mathbf{b}}$$

$$\nabla^2 \phi + \frac{\tau}{\lambda_D^2} (\phi - \tilde{\phi}) = 4\pi e (\delta \bar{n} - \delta n_e)$$

$$\delta \bar{n}(\mathbf{x}) = \frac{1}{2\pi} \int (f(\mathbf{R}) - f_{eq}(\mathbf{R})) \delta(\mathbf{R} - \mathbf{x} - \rho) d\mathbf{x} d\mu dv_{\parallel} d\varphi.$$

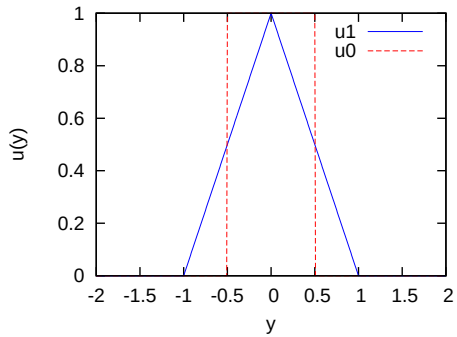
Particle in cell methods

- In particle methods, we approximate the distribution function by a collection of finite-size particles

$$f(\mathbf{x}, \mathbf{v}, t) \approx \sum_k q_k \delta_{\varepsilon_\mathbf{x}}(\mathbf{x} - \mathbf{X}_k(t)) \delta_{\varepsilon_\mathbf{v}}(\mathbf{v} - \mathbf{V}_k(t))$$

$$\int_{-\infty}^{\infty} \delta_\varepsilon(y) dy = 1$$

$$\delta_\varepsilon(y) = \frac{1}{\varepsilon} u(\frac{y}{\varepsilon})$$



- At each time step, particles are transported along trajectories described by the equation of motion

$$\frac{dq_k}{dt} = 0 \quad \frac{d\mathbf{X}_k}{dt} = \mathbf{V}_k \quad \frac{d\mathbf{V}_k}{dt} = \mathbf{F}_k$$

- the long range forces are usually solved on a grid

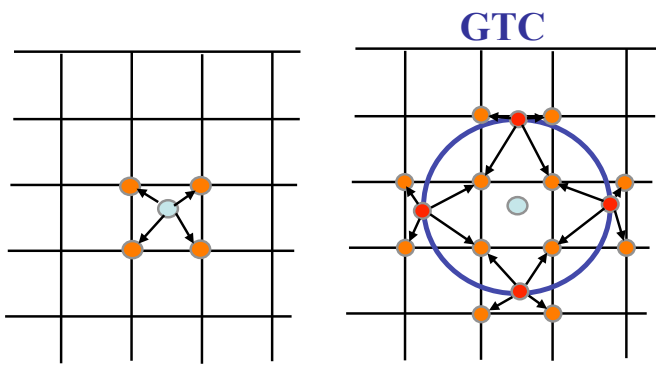
$$\rho_\mathbf{j} = \sum_k \frac{q_k}{\varepsilon_\mathbf{x}} \mathbf{u}_1(\frac{\mathbf{x_j} - \mathbf{X}_k}{\varepsilon_\mathbf{x}}) \quad \Delta\phi_\mathbf{j} = -\rho_\mathbf{j} \quad \mathbf{E_j} = -\nabla\phi_\mathbf{j} \quad \mathbf{E}(\mathbf{X}_k) = \sum_\mathbf{j} \mathbf{E_j} \mathbf{u}_1(\frac{\mathbf{x_j} - \mathbf{X}_k}{\varepsilon_\mathbf{x}})$$

Charge assignment and field interpolation in GKPIC method

$$\rho(\mathbf{x_j}) = \sum_k \sum_{p=1}^{p=4} \frac{q_k}{4\varepsilon_x} \mathbf{u}_1\left(\frac{\mathbf{x_j} - \mathbf{X}_p}{\varepsilon_x}\right) \delta(\mathbf{R}_k - \mathbf{X}_p + \rho_k)$$

$$\mathbf{E}(\mathbf{R}_k) = \sum_{p=1}^{p=4} \sum_{\mathbf{j}} \mathbf{E_j} \mathbf{u}_1\left(\frac{\mathbf{x_j} - \mathbf{X}_p}{\varepsilon_\mathbf{x}}\right) \delta(\mathbf{R}_k - \mathbf{X}_p + \rho_k))$$

Charge Deposition Step (SCATTER operation)

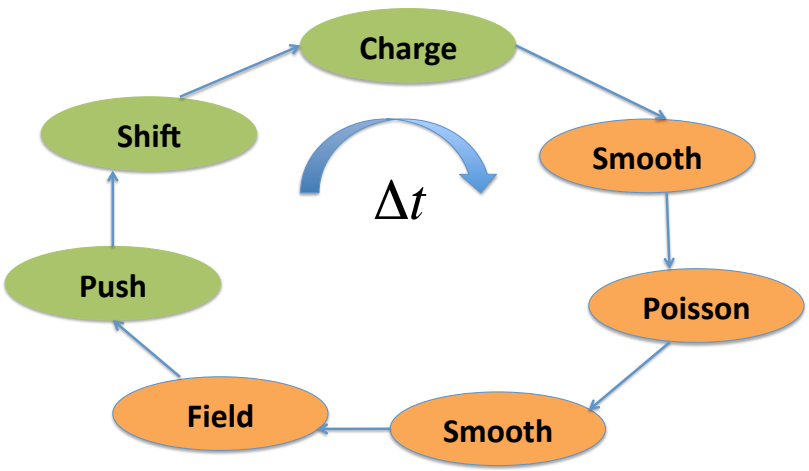


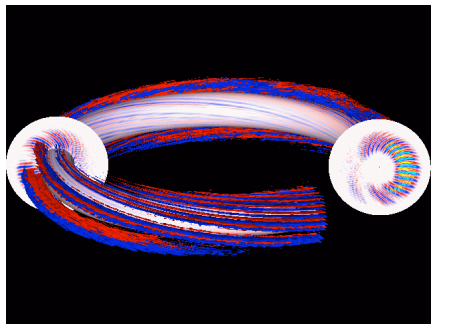
Classic PIC

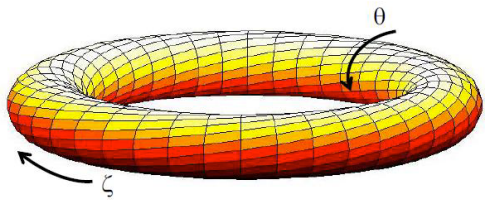
4-Point Average GK
(W.W. Lee)

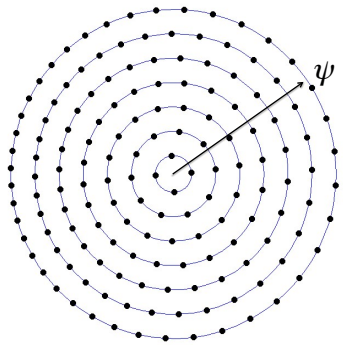
Gyrokinetic Toroidal Code (GTC, Z.Lin Science 98) for magnetically confined plasma simulations

- Solve 5D gyrokinetic Vlasov-Poisson equation using PIC methods
- 3D toroidal mesh
- Two data structures: grid-based array and particle-based array
- Six subroutines









GTC Parallelization

- Written in Fortran 90
- 3 levels of parallelism in original GTC:
  - 1d domain decomposition in toroidal dimensional
  - particle decomposition in each toroidal domain
  - loop-level parallelization with OpenMP
- Massive grid memory footprint: difficulty for simulating large scale plasmas such as ITER

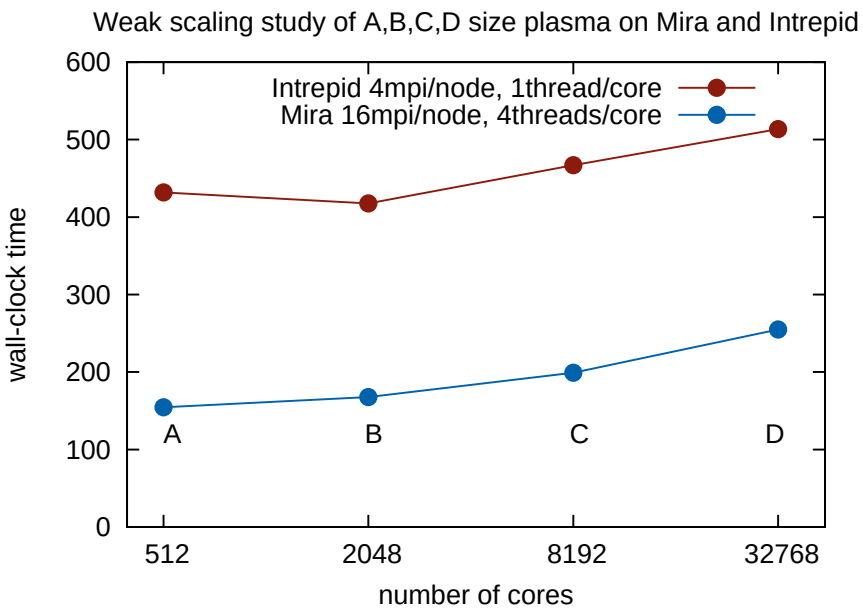
Problem Size	grid size	num. of particles in one toro. dom.
A ($a/\rho = 125$)	32,449 (1M)	3,235,896 (0.3G)
B ($a/\rho = 250$)	128,893 (4M)	12,943,584 (1.2G)
C ($a/\rho = 500$)	513,785 (16M)	51,774,336 (4.8G)
D ($a/\rho = 1000$)	2,051,567 (64M)	207,097,344 (19.2G)

using 100 particle per cell

- Introduce the key additional level of domain decomposition in the radial dimension-which is essential for efficiently carrying out ITER size simulations (Adams-Ethier 08), called GTCP
- Use PETSc 2.3.3 for Poisson solver

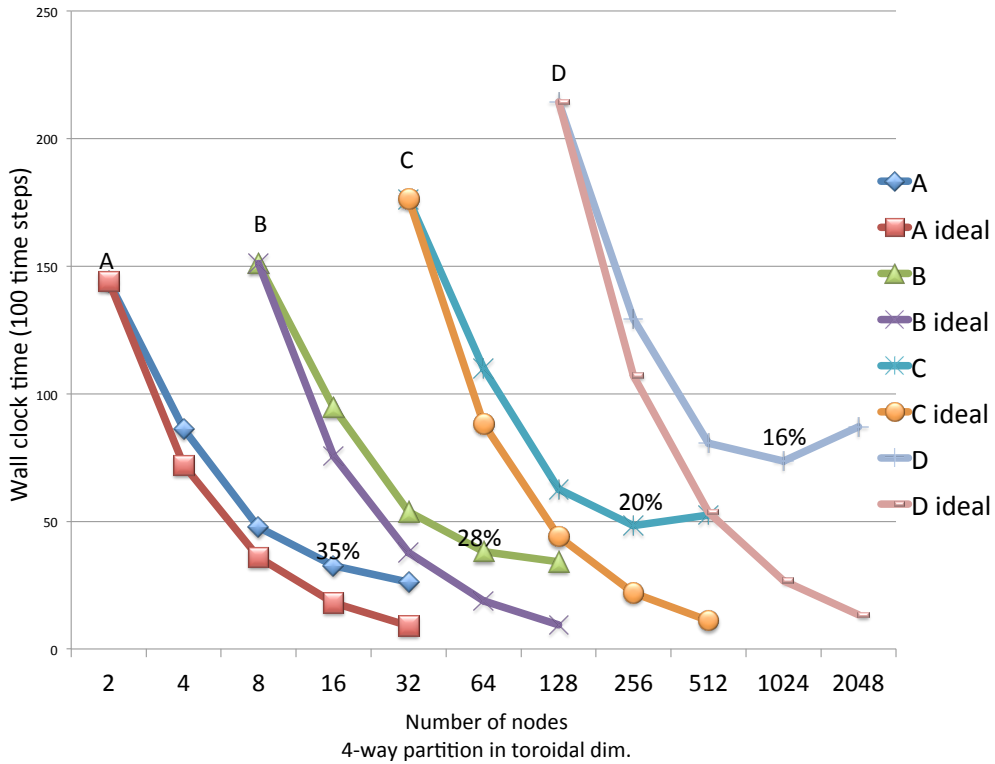
Porting GTCP from BG/P to BG/Q

- Changed the size of some OpenMP related array from 4 to 64
- Ported PETSc 2.3.3 version to BG/Q system $\longrightarrow$ Thanks to our catalyst Tim Williams
- Complier flags: -O3 -qnohot -qsmp=omp:noauto



* Q/P performance ratio per core is 2.02-2.79
* Q/P performance ratio per node is 8.08-11.16

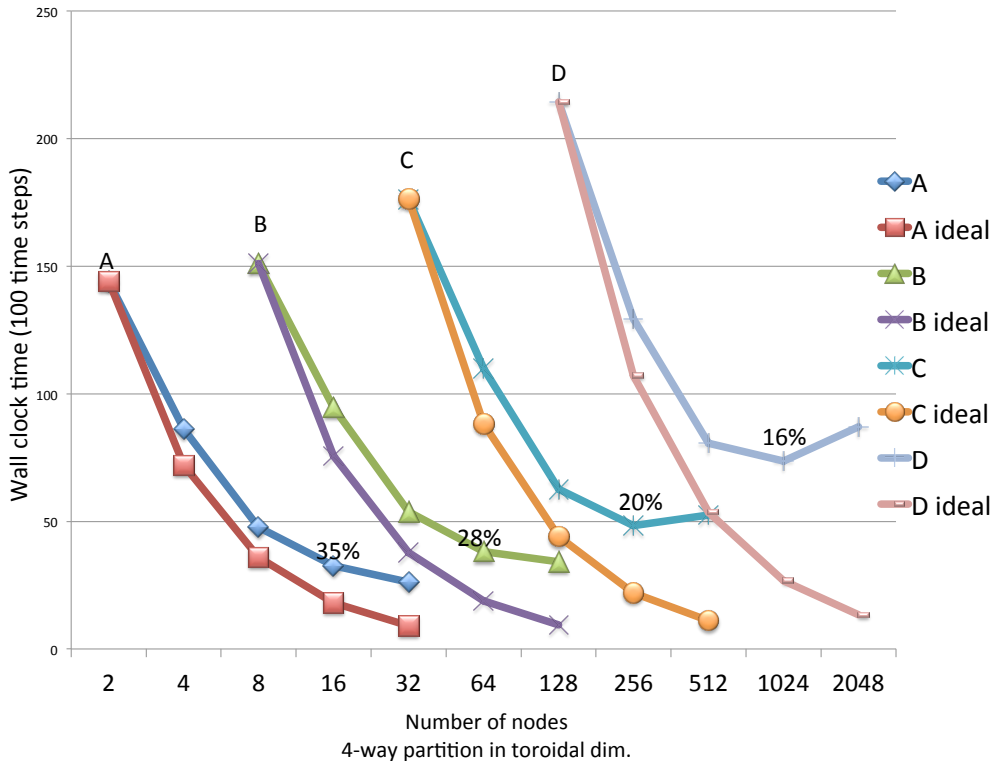
Strong Scaling Study of GTCP on BG/Q



Achieved only 16% – 35% maximal parallel efficiency (strong scaling)

- Poisson solver doesn't scale with the number of threads
- Small number of iteration in the outer level loop in smooth and field subroutine
- Atomic operation in charge subroutine
- Memory copy without multithreading in shift subroutine

Strong Scaling Study of GTCP on BG/Q



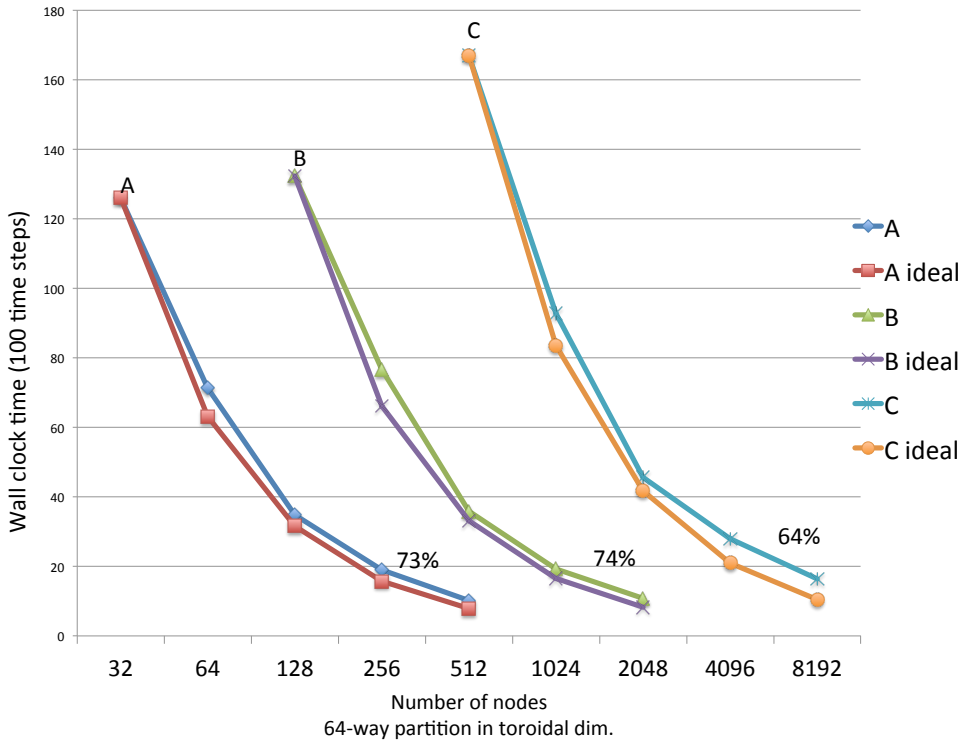
Achieved only 16% – 35% maximal parallel efficiency (strong scaling)

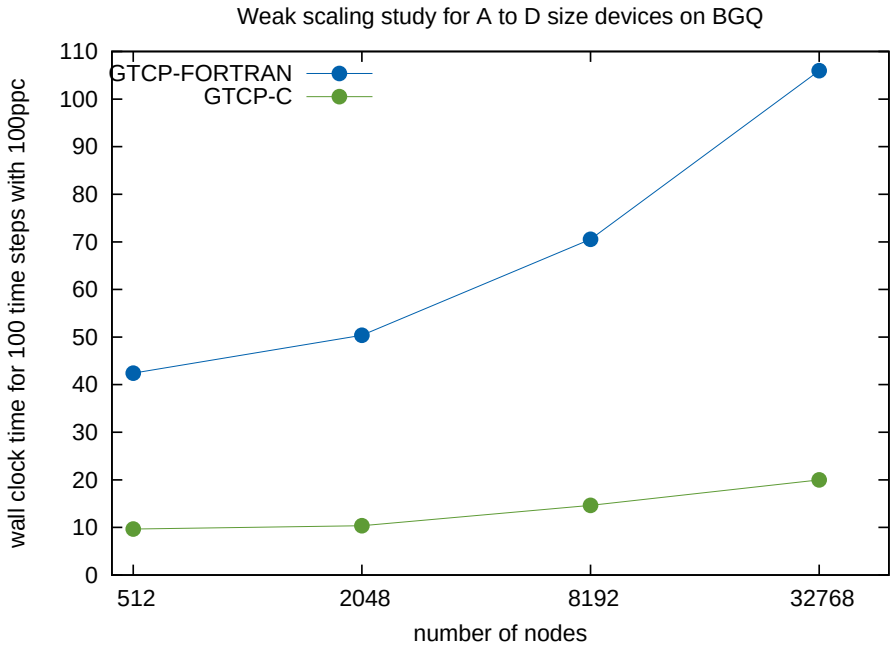
- Poisson solver doesn't scale with the number of threads
- Small number of iteration in the outer level loop in smooth and field subroutine
- Atomic operation in charge subroutine
- Memory copy without multithreading in shift subroutine

Improve single node performance with multicore and manycore optimizations(collaborated with Future Tech. Group at LBL)

- The whole code is converted to C
- A structure of array (SOA) data structure for particle array; aligned memory allocation to facilitate use of SIMD intrinsics
- Loop fusion to increase computational intensity
- Particle binning to improve locality (Kamesh et al SC11)
- A multilevel binning algorithm to improve locality and reduce data conflict: Binning the gyro-center of the particles periodically; Binning the four points of the gyro-particles at every time step (Wang et al SC12 poster)
- Atomic operation is avoided by providing a copy of the local grid for each thread
- Use a hand-coded iterative solver $\Rightarrow$ Enable multithreading
- Flat some 2D iteration to 1D

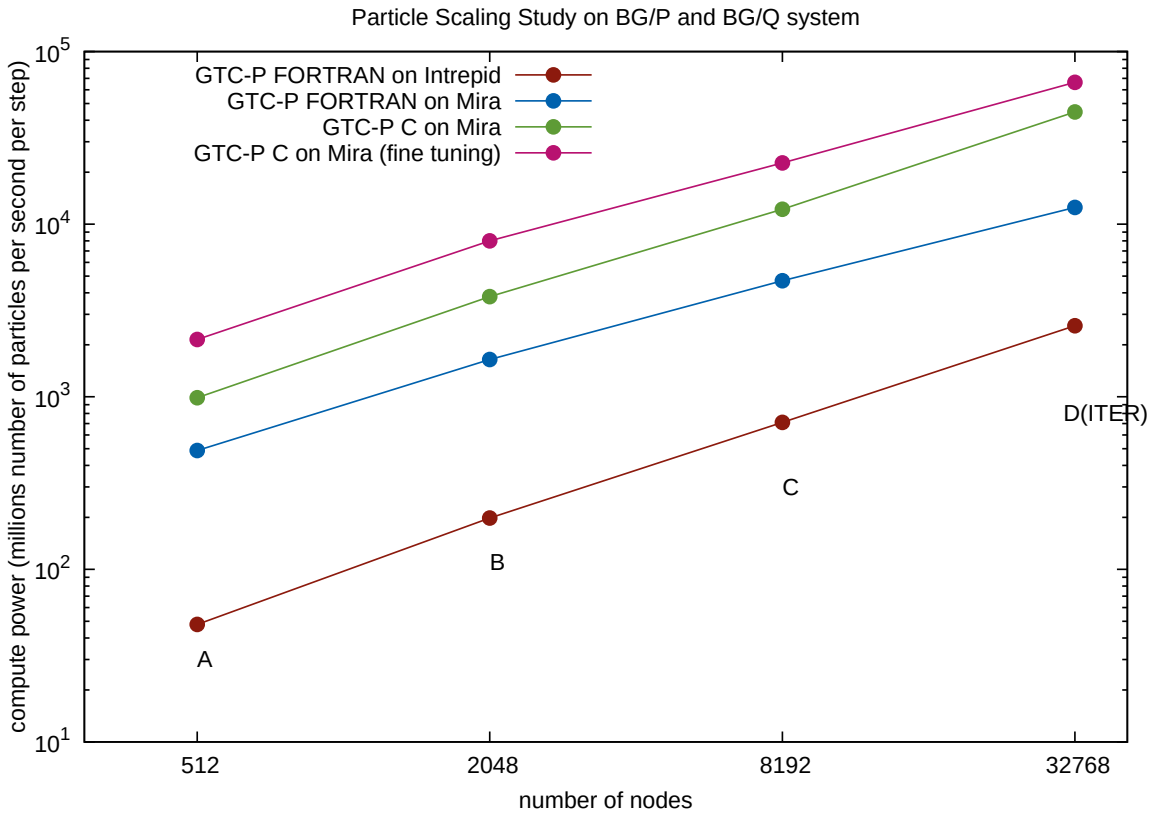
Scaling Study of GTCP-C on BG/Q





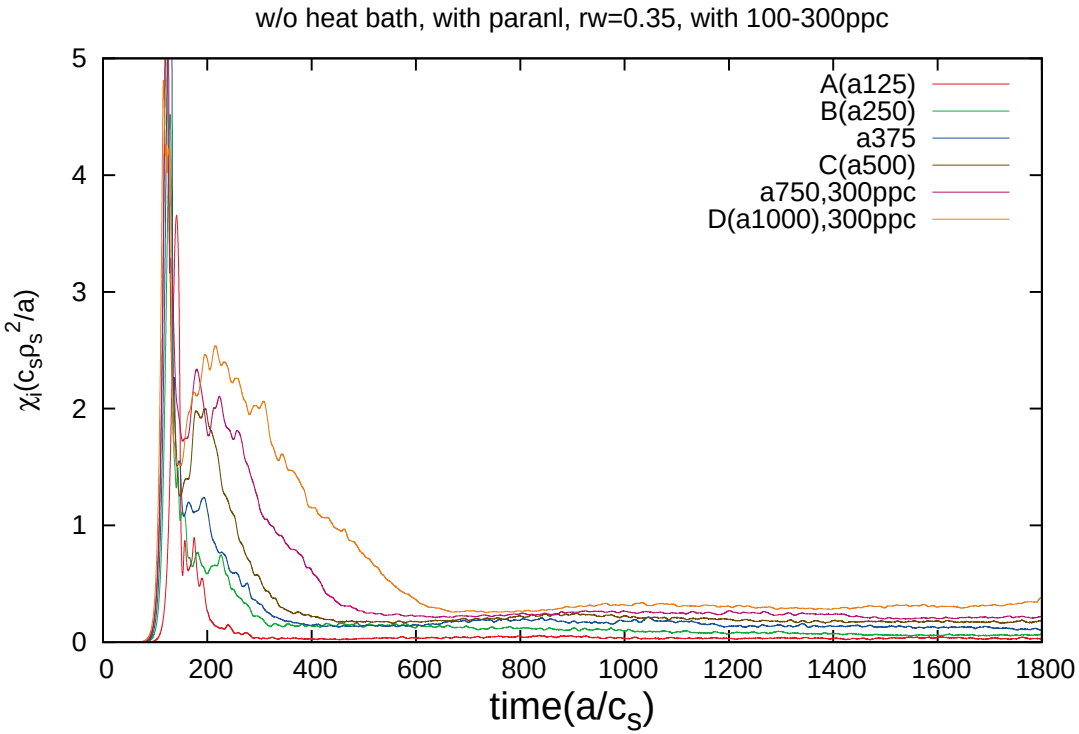
- For weak scaling plot on the right, we use one MPI in a single node with 64 OpenMP threads.
- Achieved 64% – 73% of maximal parallel efficiency (strong scaling)
- Obtained 4-5x speed up compared with the original FORTRAN code

Computer power of GTCP on BG/P and BG/Q



- red $\rightarrow$ blue: BG/P to BG/Q
- blue $\rightarrow$ green: multicore optimizations
- green $\rightarrow$ red purple: compiler flags and processing mapping
  $BG\_SMP\_FAST\_WAKEUP = YES$: $OMP\_WAIT\_POLICY = active$;
  $RUNJOB\_MAPPING$
- The "time to solution" is 50x compared with using GTCP FORTRAN code on BG/P with the same number of nodes

Study ITG driven turbulence spreading with the ultrafast GTCP-C code



- 2x longer, 7.5x more number of particles than the previous run
- Now we can simulate ITER size plasma with 300 ppc (total 40 billion particles) for 30k time steps $<$ 4 hours by using 2/3 of Mira

Conclusion

- Full application multicore and manycore optimizations
- Introduce radial domain decomposition
- Excellent scaling on BG/Q up to 524,288 cores
- Simulate ITER size plasma with 300 ppc (total 40 billion particles) for 30k time steps $<$ 4 hours by using 2/3 of Mira
- 50x "time to solution" speed up compared with GTCP FORTRAN on BG/P system

Current and future work

- Study turbulence spreading and size scaling up to ITER size with high resolution simulations
- Study phase space remapping for long time simulations with gyrokinetic δf particle in cell method
- Develop full-f capability the code

Thank you!
Questions?