EXTREME SCALE PLASMA TURBULENCE SIMULATIONS ON TOP SUPERCOMPUTERS WORLDWIDE

William M. Tang*
Princeton Institute for Computational Science & Engineering (PICSciE)
and Intel Parallel Computing Center (IPCC)
Princeton University, Princeton, New Jersey

Supercomputing 2016 Conference (SC'16)
Paper 399s-4

November 16, 2016

***Co-authors**: *Bei Wang (PU), S. Ethier (PPPL), G. Kwasniewski (ETH-Zurich), T. Hoefler (ETH-Zurich), K. Ibrahim (LBNL), K. Madduri (Penn State U), S. Williams (LBNL), L. Oliker (LBNL), C. Rosales-Fernandez (TACC), T. Williams (ANL)*

OUTLINE

I. Introduction: FES as Grand Scientific Challenge with Societal Impact

II. Computational Approach: Particle-in-Cell (PIC)

III. Performance Models for the Key Kernels of the PIC code

IV. Optimizations and Performance Results

V. Large-scale (i.e., JET and ITER) Physics Simulations enabled by Software Advances

VI. Future Implications & Summary

INTRODUCTION

Presentation Emphasis:

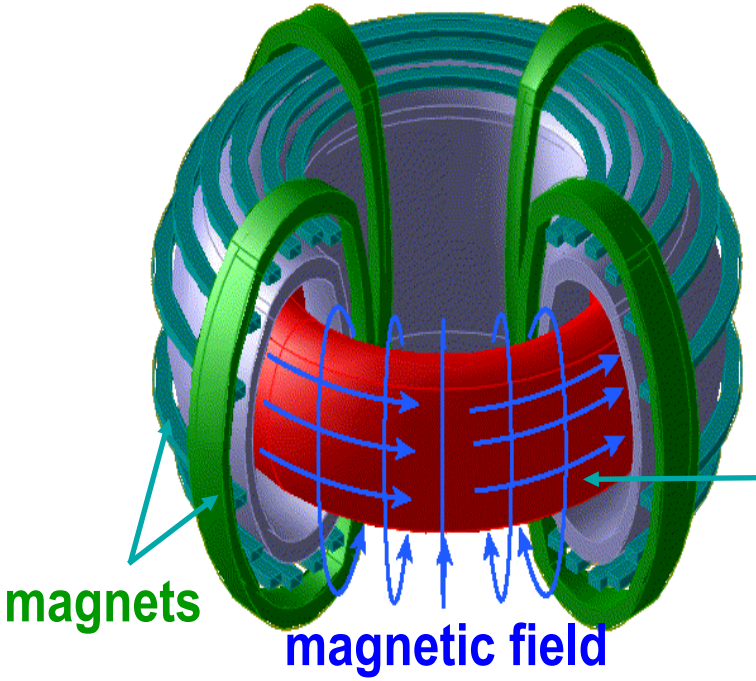
HPC Performance Scalability and Portability in an exascale-relevant grand challenge application domain (Fusion Energy Science)

• **Goal** → *delivery of discovery-science-capable software with good performance scaling, while demonstrating viable metrics on top supercomputing systems worldwide including "portability," "time to solution," & associated "energy to solution"*

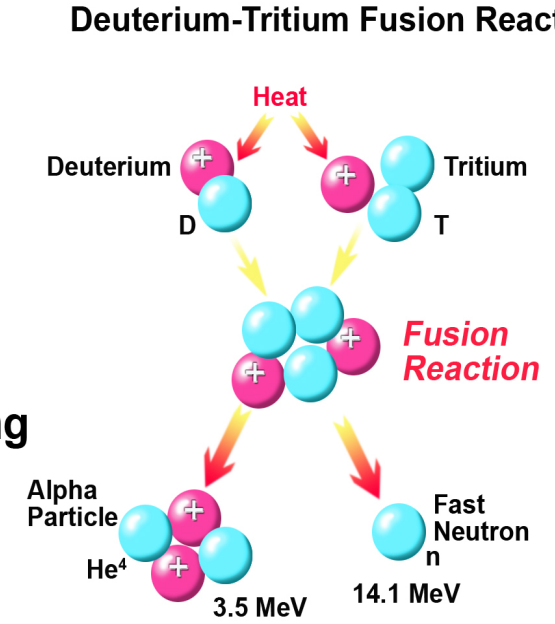
• **Task** → *Deployment of innovative algorithms utilizing MPI & OpenMP, CUDA, and OpenACC within modern code that delivers new scientific insights on world-class systems → currently: Mira; Sequoia; K-Computer; Titan; Piz Daint; Blue Waters; Stampede;TH-2 & in near future on: Summit (via CAAR), Cori, Stampede-II, Tsubame 3.0, -----*

• **Focus** → *Performance Modeling of Particle-in-Cell operations via scalable scientific software for extreme scale applications with FES as illustrative application domain*

HPC SCIENCE APPLICATION DOMAIN: MAGNETIC FUSION ENERGY (MFE)



Plasma self-heating



Energy Multiplication About 450:1

"Tokamak" Device

$D^+ + T^+ \rightarrow {}^4He^{++}$ (3.5 MeV) + n^0 (14.1 MeV)

ITER *~$25B facility located in France & involving 7 governments representing over half of world's population*

→ *dramatic next-step for Magnetic Fusion Energy (MFE) producing a sustained burning plasma*

-- Today: 10 MW(th) for 1 second with gain ~1

-- ITER: 500 MW(th) for >400 seconds with gain >10

CNN's "MOONSHOTS for 21st CENTURY" HOSTED by FAREED ZAKARIA

– *Five segments (broadcast in Spring, 2015 on CNN) exploring "exciting futuristic endeavors in science & technology" in the 21st century*

(1) Human Mission to Mars

(2) 3D Printing of a Human Heart

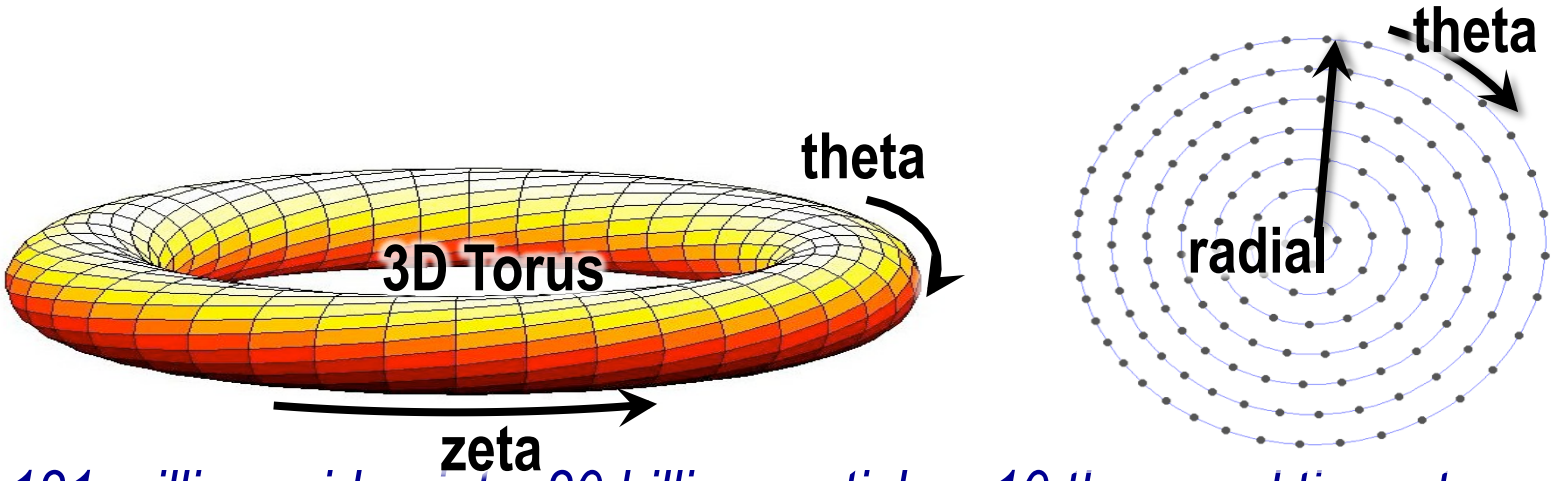
(3) Creating a Star on Earth: Quest for Fusion Energy

(4) Hypersonic Aviation

(5) Mapping the Human Brain

GPS (General Public Square) Moonshots Series: ***"Creating a Star on Earth"*** → ***"takes a fascinating look at how harnessing the energy of nuclear fusion reactions may create a virtually limitless source of clean energy."***

- Mathematics: 5D Gyrokinetic Vlasov-Poisson Equations
- Numerical Approach: Gyrokinetic Particle-in-Cell (PIC) Method

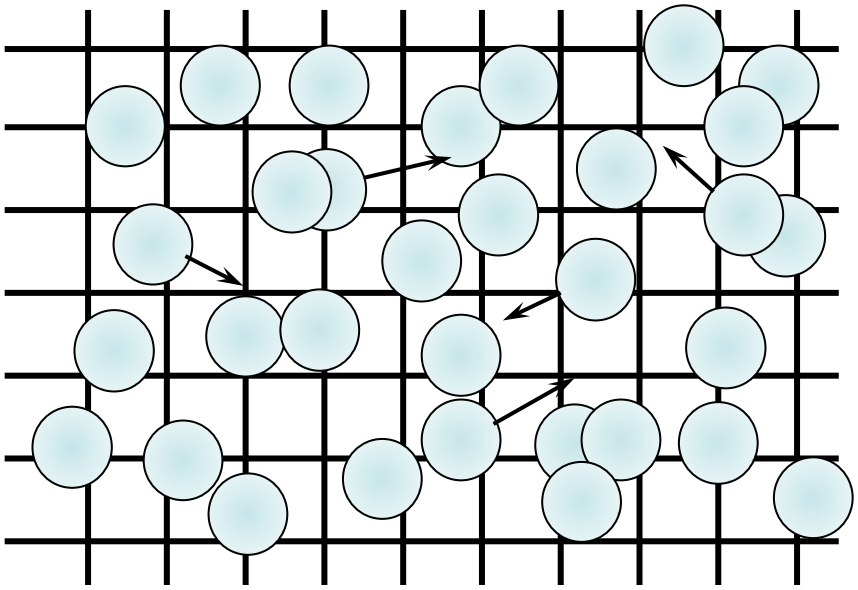


131 million grid points, 30 billion particles, 10 thousand time steps

- Domain Application Objective → *Develop efficient numerical tool to realistically simulate turbulence and associated transport in magnetically-confined plasmas (e.g., "tokamaks") using high end supercomputers*

Picture of Particle-in-Cell Method

- Charged particles sample distribution function
- Interactions occur on a grid with the forces determined by gradient of electrostatic potential (calculated from deposited charges)
- *Grid resolution dictated by Debye length ("finite-sized" particles) up to gyro-radius scale*

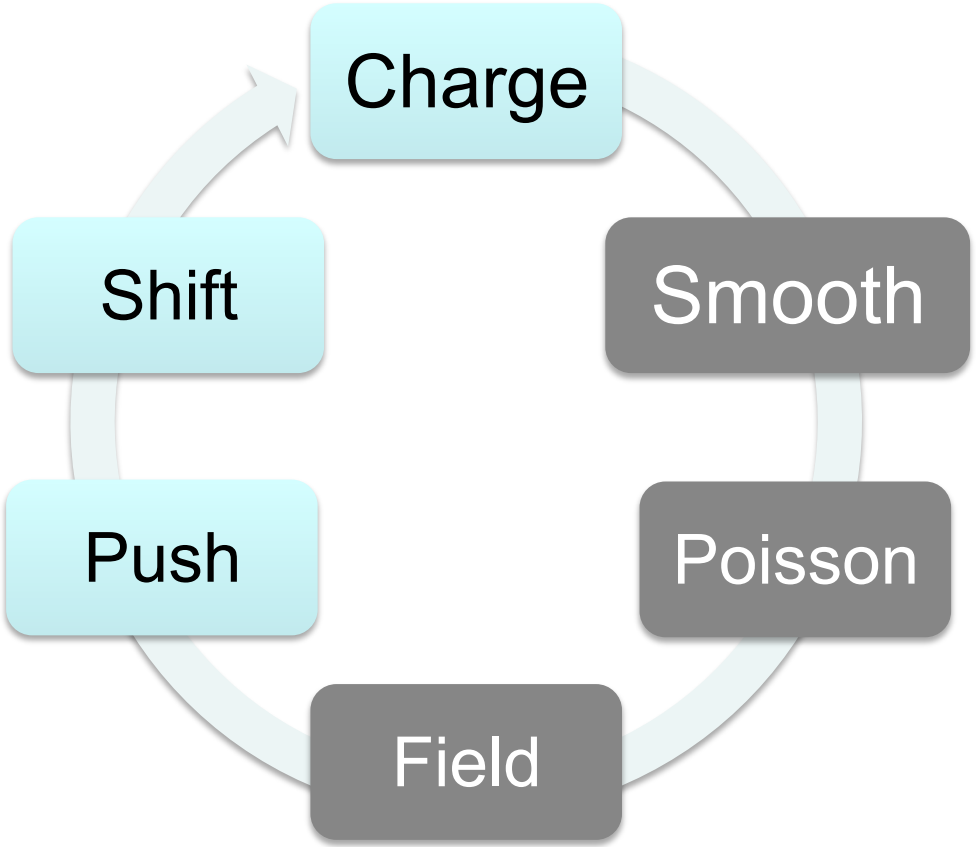


Specific PIC Operations:

- *"SCATTER", or deposit, charges as "nearest neighbors" on the grid*
- *Solve Poisson Equation for potential*
- *"GATHER" forces (gradient of potential) on each particle*
- *Move particles (PUSH)*
- *Repeat…*

Gyrokinetic PIC Code: six major subroutines

→ *provides focus for Computer Science performance modeling*



- **Charge**: particle to grid interpolation (SCATTER)
- **Smooth/Poisson/Field**: grid work (local stencil)
- **Push**:
  - grid to particle interpolation (GATHER)
  - update position and velocity
- **Shift**: in distributed memory environment, exchange particles among processors

Performance Models of Key Kernels of the PIC code (1)

PHYSICS MODEL: Full ion dynamics and electron dynamics with:

Adiabatic model (focus of previous optimization work)

- Biggest impediments to performance are *data hazards* and *data locality.*
- Time consuming kernels include charge deposition (*charge*) and field interpolation (*push*) operations.

Kinetic electron model

- Most time-consuming kernels now become field interpolation (*push*) and particle communication (*shift*) operations.
- In addition to data locality challenge, performance of code strongly influenced by network performance and specific implementation of the communication methodology.

Focus:

- **Develop performance model to evaluate the *push* and *shift* (along with particle *sort* to improve data locality).**
- **Implement the shift with MPI-3 One-sided communication to leverage capabilities of hardware-enabled Remote Direct Memory Access (RDMA).**

Performance Models of Key Kernels of the PIC code (2)

- *Analysis of data movement through caches crucial for assessing performance is far more challenging than comparing peak and achieved flop/s performance !*
- Properties of data movement investigated: *size, access pattern, source & destination*
  - Intra-node access: model the number of cache lines transferred between memory levels
  - Inter-node communication: analyze the amount of data transferred over the network

- This systematic approach yields very high accuracy execution time predictions → at least 93% achieved in the worst case

- Results show that modeling data movement can effectively predict performance on modern supercomputing platforms

Performance Models of Key Kernels of the PIC code (3)

TABLE IV

PERFORMANCE MODEL OF THE GTC-P APPLICATION. PRESENTED COMPARISON BETWEEN THE BEST FITTING MODEL (PIZ DAINT) AND THE WORST (TITAN). EXAMPLE SCENARIO SHOWN FOR 1024 NODES WEAK SCALING. B_{mc}, B_{mr}, B_c, B_{nt} AND B_{nr} REFER TO MEASURED BANDWIDTH PER CACHE LINE FOR CONTINUOUS MEMORY ACCESS, RANDOM MEMORY ACCESS, CACHE, NETWORK COMMUNICATION IN TOROIDAL AND RADIAL DIRECTION.

subroutine	kernel	model	Piz Daint			Titan		
			%time	pred.	measured	%time	pred.	measured
intra-node data transfers								
push	loop1	$l_1(\frac{c_1 m}{B_{mc}} + \frac{c_2 e + c_3 f}{B_{mr}} + \frac{c_2(m-e)+c_3(m-f)}{B_c})$	23	3.82	3.77 (1.01)	14	3.33	3.47 (0.95)
	loop2	$l_1(\frac{c_4 m}{B_{mc}} + \frac{c_5 p}{B_{mr}} + \frac{c_5(m-p)}{B_c})$	36	6.13	5.76 (1.06)	28	6.61	6.77 (0.97)
sort	loop1	$l_2(\frac{c_6 m}{B_{mc}} + \frac{c_7 p}{B_{mr}} + \frac{c_7(m-p)}{B_c})$	0.8	0.14	0.13 (1.04)	0.6	0.13	0.12 (1.13)
	loop2	$l_2(\frac{9m}{B_{mc}} + \frac{5m}{B_c})$	2.1	0.35	0.34 (1.03)	2	0.46	0.49 (0.95)
	loop3	$l_2(\frac{8m}{B_{mc}})$	1.9	0.31	0.32 (0.98)	1.8	0.41	0.41 (0.99)
shift	detect part.	$l_1(\frac{3m}{B_{mc}} + \frac{s_t+s_r}{B_{mr}})$	9.1	1.53	1.65 (0.94)	11	2.55	2.53 (1.01)
	pack part.	$l_1(\frac{4(s_t+s_r)}{B_{mc}})$	5.8	0.97	0.93 (1.03)	3.8	0.89	0.88 (1.01)
inter-node data transfers								
shift	toroidal	$l_1(\frac{s_t}{B_{nt}})$	10.6	1.86	2.14 (0.87)	20.3	4.83	5.2 (0.93)
	radial	$l_1(\frac{s_r}{B_{nr}} + c_8 log_2(r))$	7.3	1.04	0.99 (1.05)	9.5	2.27	2.83 (0.8)
total			96.6	16.9	16.8 (1.006)	91	22.4	23.7 (0.94)

New PIC Performance Optimizations

New efficient “holes removal” implementation ***to improve vectorization***

PROBLEM: (I) Moving particles out of a local domain creates "a hole" (no longer a valid particle location) in the associated memory space; AND (II) “Holes” may lead to different operations for two particles in consecutive memory locations, which brings difficulty for automatic vectorization

SOLUTION: Remove the holes completely (rather than periodically) at every time step to maximize the usage of vector units

APPLICATION: Implement on GPU and Intel Xeon Phi

Implementation of one-sided communication to ***reduce latency***

- Transfer the outgoing particles directly to the neighboring processes using MPI_Put()
- Source processes using MPI_Fetch_and_op() to reserve buffer space in the array
- Remote put operations are overlapped with packing the particles locally to transmission buffer
- We observe that on Mira, the One-sided version improved application performance between 5-7% for large runs. On other architectures, our measurements reports no significant difference.

We conclude that not all MPI libraries implement the relatively new specification in a high-performance manner

Experimental Setup: Performance Analysis (for most advanced current version of GTC-P)

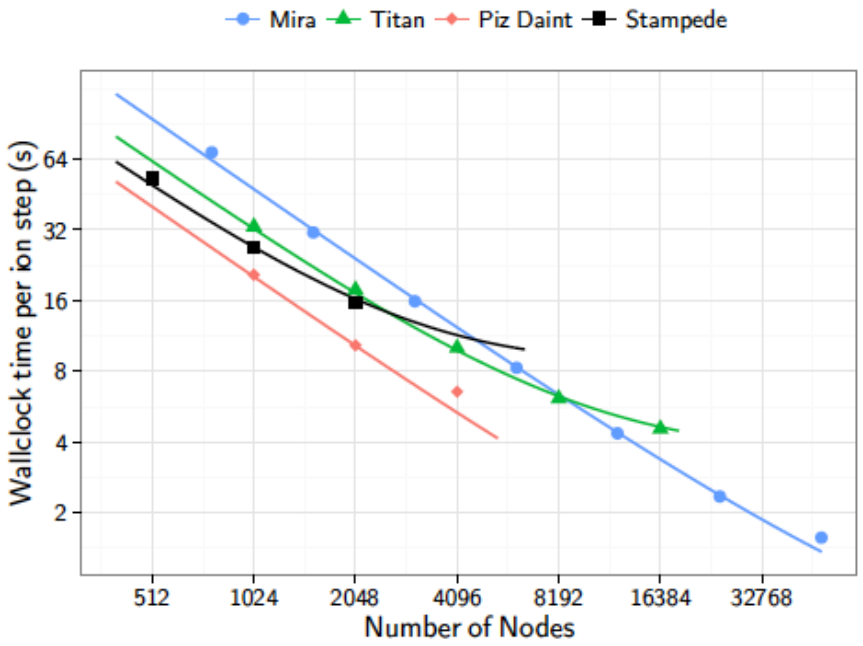
TABLE III
GTC-P NUMERICAL SETTINGS FOR KINETIC ELECTRON SIMULATIONS. *mpsi* IS THE NUMBER OF GRID POINTS IN THE RADIAL DIMENSION. *mthetamax* IS THE MAXIMUM NUMBER OF GRID POINTS IN THE POLOIDAL DIMENSION. *mgrid* IS THE NUMBER OF GRID POINTS PER TOROIDAL PLANE. *ntoroidal* IS THE TOTAL NUMBER OF TOROIDAL PLANES. *micell/mecell* IS THE NUMBER OF IONS OR ELECTRONS PER GRID POINT. *total ion/electron* = *mgrid*×*ntoroidal*×*micell/mecell*. FOR ALL SIMULATIONS, WE SET *micell* = *mecell* = 100.

	Grid Size			
	A2	B2	C2	D2
mpsi	200	400	800	1600
mthetamax	784	1568	3136	6272
mgrid	157785	629169	2512737	10043073
ntoroidal	32	32	32	32
total ion	504271872	2012060672	8038195200	32132710400
total electron	504271872	2012060672	8038195200	32132710400

- As we increase problem size each time, the number of grid points and particles increase by 4x
- Current largest problem size involves 320 million grid points and 64 billion particles

Performance Results (1)

- Strong scaling of large (C2) problem on Titan (GPU), Mira, Piz Daint and Stampede
- Solid line indicates *model-predicted running time*

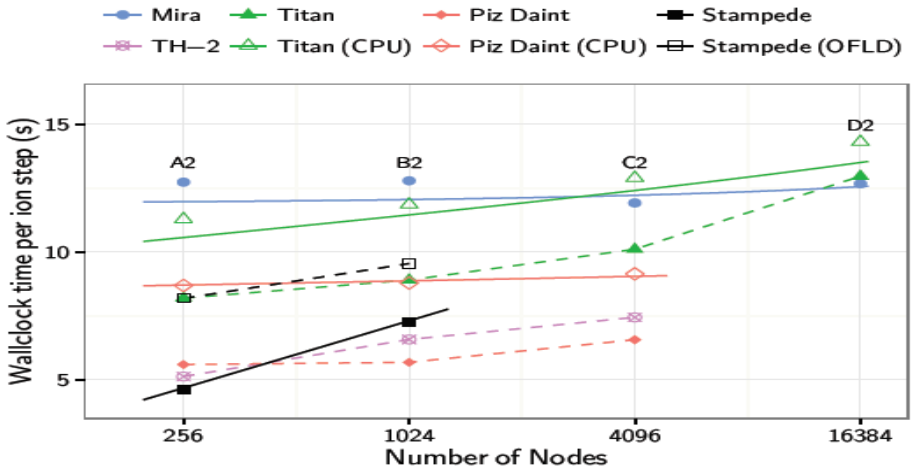


- Scalability of Titan is moderately impaired beyond 8K nodes, while scalability is impaired on Mira scaling only beyond 24K nodes
- Overall, Mira delivers twice the application performance of Titan despite having less than half of the peak performance

Performance Results (2)

- Weak scaling of four problem sizes (A2-D2) on TH-2, Titan, Mira, Piz Daint and Stampede using a fixed problem size per node on all systems

Node to node performance comparison across systems

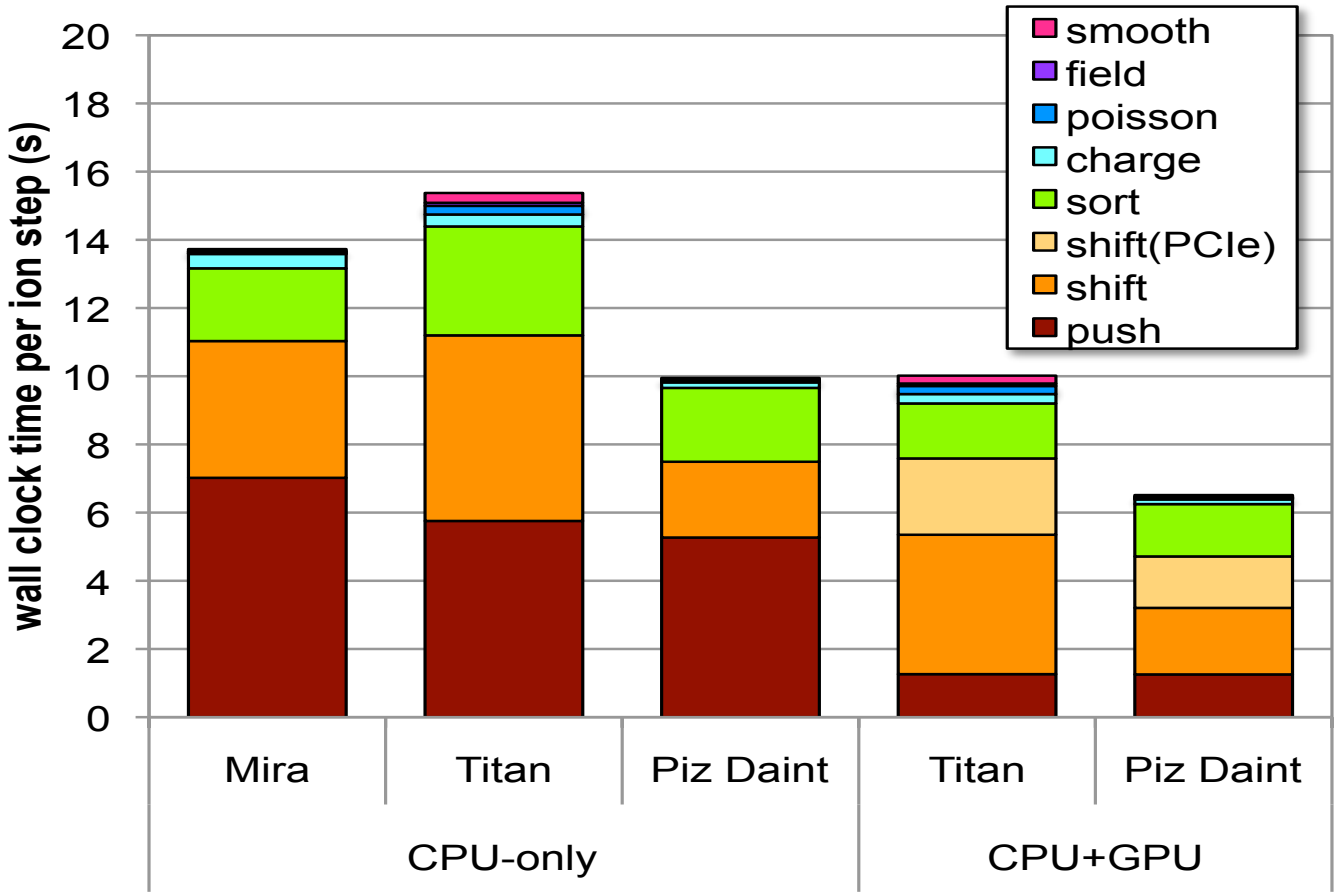


Node-to-Node Results

- Piz Daint attains 60% performance improvement over Titan despite the same GPU, and more than 2.2x improvement over BGQ on node-to-node basis
- TH2(Ivy Bridge only) and Stampede (Sandy Bridge only) deliver similar performance
- Stampede has 1.8x performance penalty when attempting to offload work to Xeon Phi
- At small scale, 3D torus, Dragonfly and Fat Tree deliver similar scalability, but networks are differentiated at large scale (beyond 4K nodes)

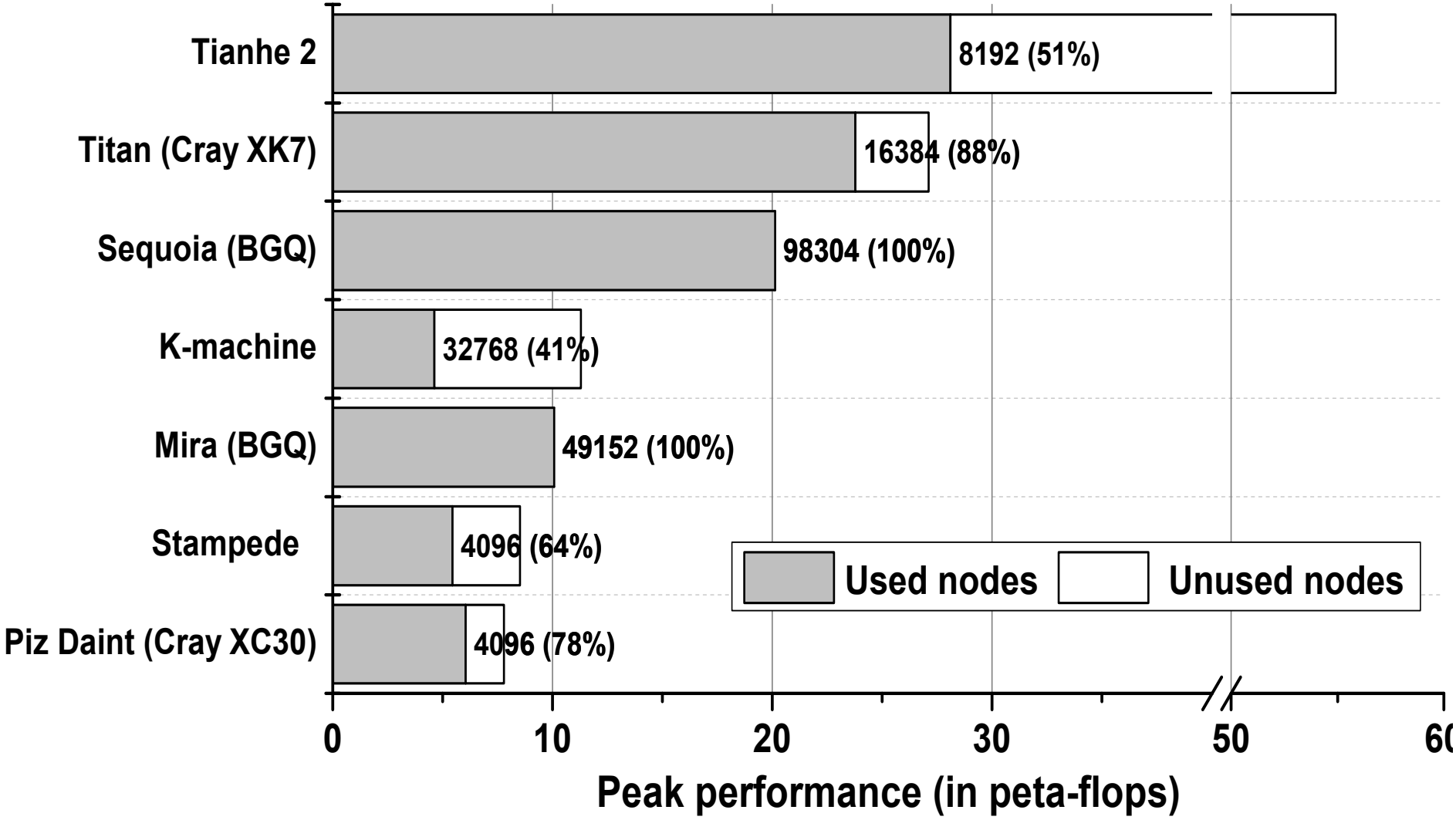
NOTE: Paper in preparation on "Modeling Data Movement Analytically: A Case Study with GTC-P" by Grzegorz Kwasniewski, Torsten Hoefler, et al. (ETH Zurich),

Performance Results (3)



Operational breakdown of time per step when using C2 problem (80M grid points, 8B ions, and 8B kinetic electrons) on 4K nodes of Mira, Titan, and Piz Daint.

GTC-P CODE PORTABILITY



- Broad range of leading multi-PF supercomputers worldwide
- Percentage indicates fraction of overall nodes currently utilized for GTC-P experiments
- NOTE: Results in this figure are only for CPU nodes on Stampede and TH-2

"ENERGY TO SOLUTION" RESULTS
(for Mira, Titan, and Piz Daint)

	CPU-Only			CPU+GPU	
	Mira	Titan	Piz Daint	Titan	Piz Daint
Nodes	4096	4096	4096	4096	4096
Power/node (W)	69.7	254.1	204.9	269.4	246.5
Time/step (s)	13.77	15.46	10	10.11	**6.56**
Energy (KWh)	**1.09**	4.47	2.33	3.10	1.84

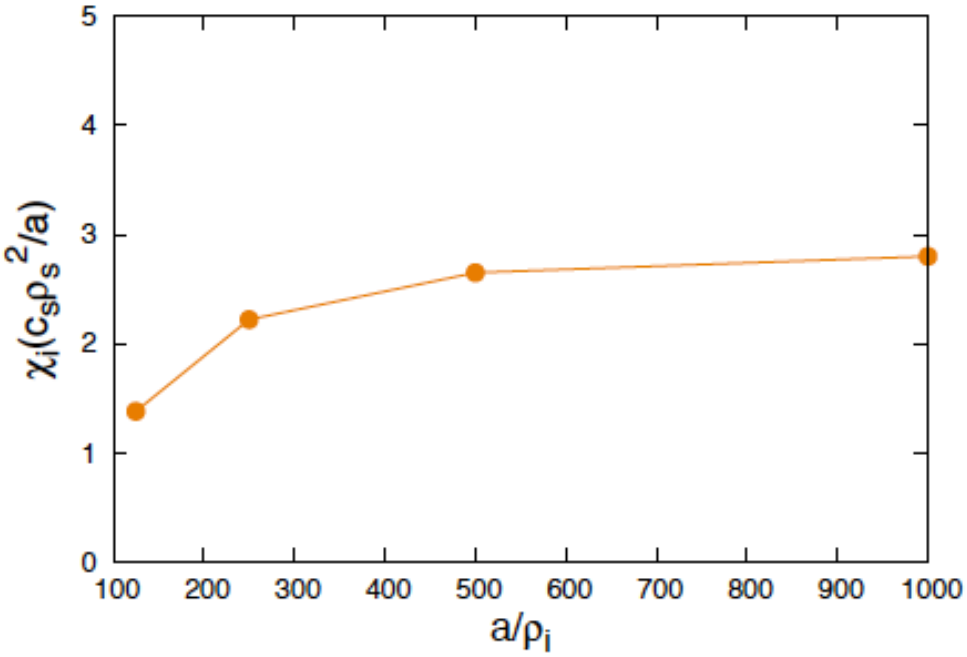
- **Energy per ion time step (KWh) by each system/platform for the weak-scaling, kinetic electron studies using 4K nodes.**

(Watts/node) * (#nodes) * (seconds per step) * (1KW/1000W) * (1hr/3600s)

- **Power/Energy estimates obtained from system instrumentation including compute nodes, network, blades, AC to DC conversion, etc.**

PHYSICS RESULTS: *Unprecedented high-resolution ITER scale (largest problem size) physics results enabled by new software advances*

- For the first time, we carry out size-scaling studies up to an ITER-size plasma for the trapped-electron instability at sufficient phase-space resolution
- Global takamak size-scaling study of trapped-electron-mode turbulence showing the plateauing of the radial electron heat flux as the size of tokamak increases



Future Implications

- Demands for increased ***physics fidelity***:
  - ITER-scale runs at the spatial resolution and temporal duration required, including complete electron dynamics
  - Capabilities to encompass electromagnetic physics, including faster and more portable multi-grid Poisson solvers (e.g.,
- Challenges & Promise for ***performance modeling optimizations for PIC codes in general***
  - Asymptotically decreasing local memory with highly localized communication networks
  - Addressing OpenMP4.5 (IPCC focus) & Open ACC2.0 (TaihuLight !) challenges
- Node and Network architecture
  - PIC simulations with flop:byte ~ 1 *require high on-node memory bandwidth*
  - For kinetic electron dynamics, inter-node communication begin dominating execution time. → *network performance and software implications (e.g., MPI libraries) play significant role for the overall PIC code performance*.
- **Energy-efficient scientific computing**
  - Today, most computer centers provide little or no information on energy and power to the users at the end of an application
  - Reporting energy by components (memory, processor, network, storage, etc) would enable scientists and vendors to help co-design their application to avoid energy hotspots and produce more energy-efficient systems.

SUMMARY

I. PRESENTATION FOCUS: HPC Performance Scalability and Portability in a representative application domain

→ *Illustration of domain application that delivers discovery science with good performance scaling, while also helping provide viable metrics on top supercomputing systems such as "portability," "time to solution," & associated "energy to solution"*

II. HPC APPLICATION DOMAIN: Fusion Energy Science

References: (i) "Scientific Discovery in Fusion Plasma Turbulence Simulations @ Extreme Scale;" W. Tang, B. Wang, S. Ethier, Computing in Science and Engineering (CiSE), vol. 16. Issue 5, pp.44-52, 2014; *(ii) SC'16 Technical Paper*

III. CURRENT PROGRESS: *Deployment of innovative algorithms: MPI & OpenMP, CUDA, with active OpenACC and OpenMP4.5 R&D within modern code that delivers new scientific insights on world-class systems → currently: Mira; Sequoia; K-Computer; Titan; Piz Daint; Blue Waters; Stampede;TH-2; Tsubame 2.5; … & in future on:* ***Sunway TaihuLight,*** *Cori, Stampede-II, Tsubame 3.0, Summit (via CAAR), Aurora (via ESP),….*

IV. FUTURE CHALLENGES: *need algorithmic & solver advances further improving data-locality -- enabled by Applied Mathematics in an interdisciplinary "Co-Design" type environment together with Computer Science & Extreme-Scale HPC Domain Applications*

ADDITIONAL SLIDES

ITER Goal: *Demonstration of Scientific and Technological Feasibility of Fusion Power*

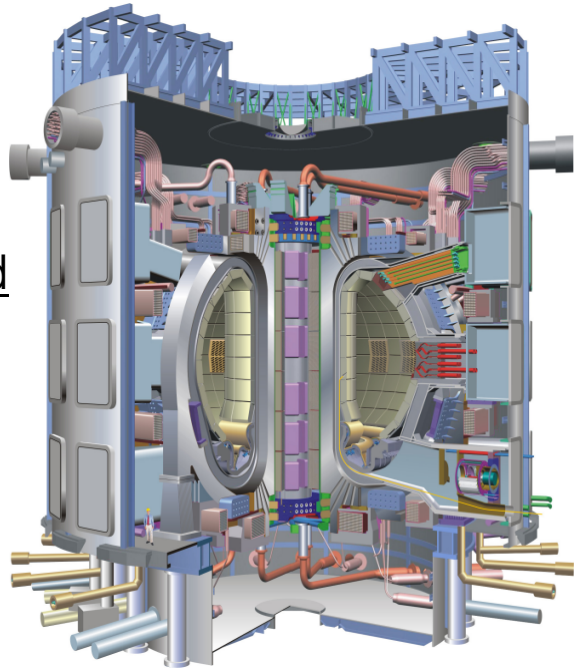
- **ITER** ***~$25B facility located in France & involving 7 governments representing over half of world's population***
  → dramatic next-step for Magnetic Fusion Energy (MFE) producing a sustained burning plasma
  -- Today: 10 MW(th) for 1 second with gain ~1
  -- ITER: 500 MW(th) for >400 seconds with gain >10

- **"DEMO"** ***demonstration fusion reactor after ITER***
  -- 2500 MW(th) continuous with gain >25, in a device of similar size and field as ITER

- **Ongoing R&D programs worldwide [experiments, theory, computation, and technology]** *essential to provide growing knowledge base for ITER operation targeted for ~ 2025*

→ Realistic HPC-enabled simulations required to cost-effectively plan, "steer," & harvest key information from expensive (~$1M/long-pulse) ITER shots

ITER



Boltzmann-Maxwell System of Equations

- The Boltzmann equation (Nonlinear PDE in Lagrangian coordinates):

$$\frac{dF}{dt} = \frac{\partial F}{\partial t} + \mathbf{v} \cdot \frac{\partial F}{\partial \mathbf{x}} + \left(\mathbf{E} + \frac{1}{c} \mathbf{v} \times \mathbf{B} \right) \cdot \frac{\partial F}{\partial \mathbf{v}} = C(F).$$

- "Particle Pushing" (Linear ODE's)

$$\frac{d\mathbf{x}_j}{dt} = \mathbf{v}_j, \qquad \frac{d\mathbf{v}_j}{dt} = \frac{q}{m} \left(\mathbf{E} + \frac{1}{c} \mathbf{v}_j \times \mathbf{B} \right)_{\mathbf{x_j}}.$$

- Klimontovich-Dupree representation,

$$F = \sum_{j=1}^{N} \delta(\mathbf{x} - \mathbf{x}_j) \delta(\mathbf{v} - \mathbf{v}_j),$$

- Poisson's Equation: (Linear PDE in Eulerian coordinates (lab frame)

$$\nabla^2 \phi = -4\pi \sum_{\alpha} q_\alpha \sum_{j=1}^{N} \delta(\mathbf{x} - \mathbf{x}_{\alpha j})$$

- Ampere's Law and Faraday's Law [Linear PDE's in Eulerian coordinates (lab frame)]