SCALABLE SCIENTIFIC SOFTWARE FOR EXTREME SCALE APPLICATIONS: FUSION ENERGY SCIENCE

William M. Tang*
Princeton University, Princeton, NJ USA

ARGONNE TRAINING PROGRAM ON EXTREME SCALE COMPUTING (ATPESC 2015)

St. Charles, Illinois

August 10, 2015

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

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

   → Illustration of domain application that delivers discovery science, 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


III. CURRENT PROGRESS: Deployment of innovative algorithms 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, -----  

IV. COMMENTS ON FUTURE PROGRESS: need algorithmic & solver advances enabled by Applied Mathematics – in an interdisciplinary “Co-Design” type environment together with Computer Science & Extreme-Scale HPC Domain Applications
Performance Development of HPC over the Last 22 Years from the Top 500 (J. Dongarra)

- 59.7 GFlop/s
- 400 MFlop/s
- 1.17 TFlop/s
- 33.9 PFlop/s
- 153 TFlop
- 309 PFlop/s

My Laptop 70 Gflop/s
My iPhone 4 Gflop/s

SUM N=1
N=500
Applications Impact ➔ Actual value of extreme Scale HPC to scientific domain applications & industry

Context: recent White House announcement of NATIONAL STRATEGIC COMPUTING INITIATIVE

• Practical Considerations: “Better Buy-in” from Science & Industry requires:
  - Moving beyond “voracious” (more of same - just bigger & faster) to “transformational” (achievement of major new levels of scientific understanding)
  - Improving experimental validation, verification & uncertainty quantification to enhance realistic predictive capability of both hypothesis-driven and big-data-driven statistical approaches
  - Deliver software engineering tools to improve “time to solution” and “energy to solution”
  - David Keyes: Billions of $ of scientific software worldwide hangs in the balance until better algorithms arrive to span the “architecture-applications gap.”

• Associated Challenges:
  - Hardware complexity: Heterogeneous multicore; gpu+cpu ➔ Summit; mic+cpu ➔ Aurora
  - Software challenges: Rewriting code focused on data locality

• Applications Imperative: “Accountability” aspect
  ➔ Need to provide specific examples of impactful scientific and mission advances enabled by progress from terascale to petascale to today’s multi-petascale HPC capabilities
HPC SCIENCE APPLICATION DOMAIN: MAGNETIC FUSION ENERGY (MFE)

- Extremely hot plasma (several hundred million degree) confined by strong magnetic field

- **Turbulence** → *Physics mechanism for energy leakage from magnetic confinement system*
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
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)
  \[
  \frac{dx_j}{dt} = v_j, \quad \frac{dv_j}{dt} = \frac{q}{m} \left( \mathbf{E} + \frac{1}{c} \mathbf{v} \times \mathbf{B} \right)_{x_j}.
  \]

- Klimontovich-Dupree representation,
  \[
  F = \sum_{j=1}^{N} \delta(x - x_j) \delta(v - v_j),
  \]

- Poisson’s Equation: (Linear PDE in Eulerian coordinates (lab frame))
  \[
  \nabla^2 \phi = -4\pi \sum_{\alpha} q_{\alpha} \sum_{j=1}^{N} \delta(x - x_{\alpha j})
  \]

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

• **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…
BASIC STRUCTURE OF PIC METHOD

• System represented by set of particles
• Each particle carries components: position, velocity and weight ($x, v, w$)
• Particles interact with each other through long range electromagnetic forces
• Forces evaluated on grid and then interpolated to the particle
  ~ $O(N+M\log M)$
• PIC approach involves two different data structures and two types of operations
  – **Charge**: Particle to grid interpolation (**SCATTER**)
  – **Poisson/Field**: Poisson solve and field calculation
  – **Push**: Grid to particle interpolation (**GATHER**)

Microturbulence in Fusion Plasmas – Mission Importance:  
*Fusion reactor size & cost determined by balance between loss processes & self-heating rates*

- "Scientific Discovery" - Transition to favorable scaling of confinement produced in simulations for ITER-size plasmas
  - $a/\rho_i = 400$ (JET, largest present lab experiment)
  - $a/\rho_i = 1000$ (ITER, ignition experiment)

- **Multi-TF simulations** using 3D PIC code [Z. Lin, et al, 2002] \( \rightarrow \) 1B particles, 100M spatial grid points; 7K time steps \( \rightarrow \) 1st ITER-scale simulation with ion gyroradius resolution

- BUT, **physics understanding** of problem size scaling demands **high resolution** requiring **modern LCF’s, new algorithms, & modern diagnostics for VV&UQ**

\( \rightarrow \) **Progress enabled by DOE INCITE Projects on LCF’s & G8 Fusion Exascale Project on major international facilities**

\( \rightarrow \) **Excellent Scalability of 3D PIC Codes on modern HPC platforms enables resolution/physics fidelity needed for physics understanding of large fusion systems**

\( \rightarrow \) **BUT – efficient usage of current LCF’s worldwide demands code re-write featuring modern CS/AM methods addressing locality & memory demands**
• 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
New Physics Results: Fusion system size-scaling study of “trapped-electron-mode” turbulence showing the “plateauing” of the radial electron heat flux as size of tokamak increases.
GTC-P: six major subroutines

- **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
Operational breakdown of time per step when using 80M grid points, 8B ions, and 8B kinetic electrons on 4K nodes of *Mira, Titan, and Piz Daint.*
“True weak scaling study” carried out on *increasing problem size* (four different sized plasmas labeled A to D) on a variety of leadership-class supercomputers worldwide

- Roughly 3.2M particles per process in these computations
- Both *1 MPI process per node* and *1 MPI process per NUMA* node are considered in these studies.

*for non-uniform-memory access [NUMA] issues*
## Performance Evaluation Platforms (1)

<table>
<thead>
<tr>
<th>Systems</th>
<th>IBM BG/Q (Mira)</th>
<th>Cray XK7 (Titan)</th>
<th>Cray XC 30 (Piz Daint)</th>
<th>NVIDIA Kepler</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPUs per node</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Interconnect</td>
<td>Custom 5D Torus</td>
<td>Gemini 3D Torus</td>
<td>Aries Dragonfly</td>
<td>-</td>
</tr>
<tr>
<td>Core</td>
<td>IBM A2</td>
<td>AMD Opteron 6274 (Interlagos)</td>
<td>Intel Xeon E5-2670 (Sandy Bridge)</td>
<td>K20x</td>
</tr>
<tr>
<td>Frequency (GHz)</td>
<td>1.6</td>
<td>2.2</td>
<td>2.6</td>
<td>0.732</td>
</tr>
<tr>
<td>Data cache per core (KB)</td>
<td>32</td>
<td>16+2048¹</td>
<td>32+256</td>
<td>64</td>
</tr>
<tr>
<td>Cores per CPU</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>14 (SMX’s)</td>
</tr>
<tr>
<td>Last-level cache per CPU (MB)</td>
<td>32</td>
<td>8</td>
<td>16</td>
<td>1.5</td>
</tr>
<tr>
<td>DP GFlop/s per node</td>
<td>204.8</td>
<td>140.8</td>
<td>166.4</td>
<td>1311</td>
</tr>
<tr>
<td>STREAM GB/s per node</td>
<td>28</td>
<td>31²</td>
<td>38</td>
<td>171</td>
</tr>
</tbody>
</table>

¹Each pair of cores shared 2048 KB L2 cache ²NUMA
# Performance Evaluation Platforms (2)

<table>
<thead>
<tr>
<th>Systems</th>
<th>Dell Cluster (Stampede)</th>
<th>Cray XE6 (Blue Waters)</th>
<th>Intel Xeon Phi (Stampede)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPUs per node</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Interconnect</td>
<td>InfiniBand Fat-Tree</td>
<td>Gemini 3D Torus</td>
<td>InfiniBand Fat-Tree</td>
</tr>
<tr>
<td>Core</td>
<td>Intel Xeon E5-2680</td>
<td>AMD Opteron 6276</td>
<td>Intel Xeon Phi SE10P</td>
</tr>
<tr>
<td></td>
<td>(Sandy Bridge)</td>
<td>(Interlagos)</td>
<td></td>
</tr>
<tr>
<td>Frequency (GHz)</td>
<td>2.7</td>
<td>2.45</td>
<td>1.1</td>
</tr>
<tr>
<td>Data cache per core (KB)</td>
<td>32+256</td>
<td>16+2048(^1)</td>
<td>32+512</td>
</tr>
<tr>
<td>Cores per CPU</td>
<td>8</td>
<td>8</td>
<td>61</td>
</tr>
<tr>
<td>Last-level cache per CPU (MB)</td>
<td>20</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>DP GFlop/s per node</td>
<td>345.6</td>
<td>313.6</td>
<td>1070</td>
</tr>
<tr>
<td>STREAM GB/s per node</td>
<td>78(^2)</td>
<td>62(^2)</td>
<td>160</td>
</tr>
</tbody>
</table>

\(^1\)Each pair of cores shared 2048 KB L2 cache  \(^2\)NUMA
Weak Scaling of GTC-P (GPU-version) on Heterogenous (GPU/CPU) “Titan” and “Piz Daint”

The number of particles per cell is 100

GTC-P GPU obtains 1.7x speed up

Same code for all cases → Performance difference solely due to hardware/system software
GTC-P Weak Scaling Results on Various Supercomputers
[Titan, Blue Waters, Mira, Piz Daint, Stampede: 1 MPI per NUMA node]
vertical scale = wall-clock time for 100 time-steps

A (MPI ranks: 64)

B (MPI ranks: 256)

C (MPI ranks: 1024)

D (MPI ranks: 4096)

PIC Operations
- smooth
- field
- poisson
- sort
- shift
- push
- charge
GTC-P Weak Scaling Results on Various Supercomputers
[Titan, Blue Waters, Mira, Piz Daint, Stampede: 1 MPI per node]
vertical scale = wall-clock time for 100 time-steps

A (MPI ranks: 64)

B (MPI ranks: 256)

C (MPI ranks: 1024)

D (MPI ranks: 4096)

PIC Operations:
- smooth
- field
- poisson
- sort
- shift
- push
- charge
GTC-P (adiabatic electron model) strong scaling for the 131M grid points, 13B particles case from 512 nodes on Titan (GPU), Mira and Piz Daint (GPU).

Note: plotted on log-log axes
GTC-P (kinetic electron model) strong scaling for the 80M grid points, 8B ion and 8B electron case on Titan (GPU), Mira and Piz Daint (GPU).

Note → plotted on log-log axes
Comparative Weak Scaling Time to Solution for 6 HPC Platforms

- GTC-P (adiabatic electron model) results for 4 problem sizes (2.1M, 8.2M, 32.8M, 131.3M grid points) each using 100 ions per grid point (with 200 on Sequoia);
- Problems ran at 12.5%, 25%, 50%, and 100% of maximum nodes used for each system.
GTC-P (kinetic electron) weak scaling performance using a fixed problem size per node across all systems allows comparisons of node performance.
Collaborative Studies with TH-2

- Measure MPI bandwidth between CPU to CPU (“host”), MIC to MIC (“native”) and CPU to MIC (“symmetric”) operation on TH-2 using the Intel MPI benchmark.
- “Offload” mode version of GTC-P developed to facilitate using many MICS on one compute node.
- Associated investigations include:
  - True weak scaling performance with increasing problem size and phase-space resolution.
    - starting from A100 problem size on 224 TH-2 nodes to D100 (ITER) problem size on 8192 nodes.
  - Deployment of 1MIC, 2MIC’s and 3MIC’s respectively for these weak scaling performance studies.
Collaborative Studies with “Stampede”

**Tasks:**

- Improve intra-node communication between the host and the MICs to reduce overhead in the MPI Scatter operation in GTC-P
- Improve inter-node communication between MIC’s (for particle shift operation)
- (Intel – R. Rahman): optimize particle loading for symmetric runs; explore KNC intrinsics
- Move actively into next phase of true weak scaling performance studies with increasing problem size – using up to 4K MIC nodes.
GTC-P (kinetic electron model) weak scaling time-to-solution results:

- 4 problems (5M, 20M, 80M, and 321M grid points) run on each system using 100 ions and 100 electrons per grid point
- 4 configurations are run at 12.5%, 25%, 50%, and 100% of the maximum nodes used for each system.
“ENERGY TO SOLUTION” ESTIMATES
(for Mira, Titan, and Piz Daint)

<table>
<thead>
<tr>
<th></th>
<th>CPU-Only</th>
<th>CPU+GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mira</td>
<td>Titan</td>
</tr>
<tr>
<td>Nodes</td>
<td>4096</td>
<td>4096</td>
</tr>
<tr>
<td>Power/node (W)</td>
<td>69.7</td>
<td>254.1</td>
</tr>
<tr>
<td>Time/step (s)</td>
<td>13.77</td>
<td>15.46</td>
</tr>
<tr>
<td>Energy (KWh)</td>
<td>1.09</td>
<td>4.47</td>
</tr>
</tbody>
</table>

- 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.
• Number of “Lines of Code (LOC)” modified provides quantitative measure of “Level of Effort” made to port and optimize GTC-P code to a specific architecture.
  -- considered “pushe” and “sorte” operations in GTC-P code
  -- speed-up measures:
    ➔ GPU: single-node Kepler vs. single Sandybridge node
    ➔ Xeon-Phi: single MIC vs. two Sandybridge nodes
Current Collaborative Studies for Intel MIC (TACC and ETH Zurich)

• LOCAL MEMORY ISSUES:
  “Holes Removal” → Moving particles out of a local domain creates "a hole" (no longer a valid particle location) in the associated memory space
→ efficient "particle removal algorithm” to avoid exhausting the existent local memory.

→ need to remove the hole periodically -- but best to remove holes completely

“Vectorization” → Improve "PUSH" & "CHARGE” operations: need to deal with two particles exhibiting different behavior at different consecutive memory locations.

→ This necessitates two separate instructions down to the computer level;
→ "Vectorization" means using a single instruction for multiple data;

“Latency”
implementation of one-side MPI communication →
  2 sided: synchronized; increases latency
  1 sided: unsynchronized; helps with reducing latency
APPLIED MATH LOCALITY CHALLENGE: GEOMETRIC HAMILTONIAN APPROACH TO SOLVING GENERALIZED VLASOV-MAXWELL EQUATIONS

Hamiltonian $\rightarrow$ Lagrangian $\rightarrow$ Action $\rightarrow$ Variational Optimization $\rightarrow$ Discretized Symplectic Orbits for Particle Motion

I. “Ultrahigh Performance 3-Dimensional Electromagnetic Relativistic Kinetic Plasma Simulation

$\rightarrow$ Basic foundation for symplectic integration of particle orbits in electromagnetic fields without frequency ordering constraints
$\rightarrow$ Foundational approach for present-day simulations of laser-plasma interactions on modern supercomputing systems
$\rightarrow$ Limited applicability with respect to size of simulation region and geometric complexity

II. “Geometric Gyrokinetic Theory for Edge Plasmas”

$\rightarrow$ Basic foundation for symplectic integration of particle orbits in electromagnetic low-frequency plasma following GK ordering
$\rightarrow$ Still outstanding challenge: Address reformulation of non-local Poisson Equations structure for electromagnetic field solve
Concluding Comments

• Presentation of a modern HPC domain application code capable of scientific discovery while providing good performance scaling and portability on top supercomputing systems worldwide – together with illustrating the key metrics of “time to solution” and associated “energy to solution”

• Illustrative HPC domain application considered: Fusion Energy Science


• Current progress achieved included deployment of innovative algorithms within a modern application code (GTC-P) that delivers new scientific insights on world-class systems → currently: Mira; Sequoia; K-Computer; Titan; Piz Daint; Blue Waters; Stampede;TH-2

w/ future targets: Summit (via CAAR), Cori, Aurora, Stampede-II, Tsubame 3.0, -----

• Future progress will require algorithmic & solver advances enabled by Applied Mathematics – in an interdisciplinary “Co-Design” type environment together with Computer Science & Extreme-Scale HPC Domain Applications