Performance Portability of HPC Discovery Science Software: Fusion Energy Turbulence Simulations at Extreme Scale

***W. Tang*[1], *B. Wang*[1], *S. Ethier* [2], *Z. Lin* [3]**

As HPC R&D moves forward on a variety of "path to exascale" architectures today, an associated objective is to demonstrate performance portability of discovery-science-capable software. Important application domains, such as Magnetic Fusion Energy (MFE), have improved modeling of increasingly complex physical systems – especially with respect to reducing "time-to-solution" as well as "energy to solution." The emergence of new insights on confinement scaling in MFE systems has been aided significantly by efficient software capable of harnessing powerful supercomputers to carry out simulations with unprecedented resolution and temporal duration to address increasing problem sizes. Specifically, highly scalable particle-in-cell (PIC) programming methodology is used in this paper to demonstrate how modern scientific applications can achieve efficient architecture-dependent optimizations of performance scaling and code portability for path-to-exascale platforms. *Keywords: Turbulence Simulations, Particle-In-Cell, Portability, HPC.*

Introduction

A major challenge for supercomputing today is to demonstrate how advances in HPC technology translate to accelerated progress in key grand challenge application domains. This is the focus of an exciting new program in the US called the "National Strategic Computing Initiative (NSCI)" that was announced on July 29, 2015 involving all research & development (R&D) programs in the country to "enhance strategic advantage in HPC for security, competitiveness, and discovery." A strong focus in such prominent application domains is to accelerate progress in modern codes capable of modeling complex physical systems – especially with respect to reduction in "time-to-solution" as well as "energy to solution." In general, the demise of Dennard Scaling coupled with the desire for processor and application performance to continue to track Moore's Law [reference] has necessitated the switch from traditional superscalar processors to increasingly efficient processors built from lightweight cores and a hierarchical memory architecture. It is understood that if properly validated against experimental measurements/observational data and verified with mathematical tests and computational benchmarks, advanced codes can greatly improve much-needed predictive capability in many strategically important areas of interest.

As an illustrative example, computational advances in Magnetic Fusion Energy – a key scientific application area that was identified by the 2015 CNN "Moonshots for the 21st Century" series [Reference] as one of five such prominent grand challenges – have produced particle-in-cell (PIC) simulations of turbulent kinetic dynamics for which computer run-time and problem size scale very well with the number of processors on massively parallel many-core supercomputers. For example, the GTC-Princeton (GTC-P) code, which has been developed with a "co-design" focus, has demonstrated the effective usage of the full power of current leadership class computational platforms worldwide at the petascale and beyond to produce efficient nonlinear PIC simulations that have advanced progress in understanding the complex nature of plasma turbulence

[1]Princeton Institute for Computational Science and Engineering, Princeton University
[2]Princeton Plasma Physics Laboratory
[3]University of California Irvine

and confinement in fusion systems for the largest problem sizes. Unlike fluid-like computational fluid dynamics (CFD) codes, PIC codes are characterized by having less than 10 key operations which can then be the focus of advanced computer science performance optimization methods. Results from these truly cross-disciplinary investigations have provided strong encouragement for being able to include increasingly realistic dynamics in extreme-scale computing campaigns with the goal of enabling predictive simulations characterized by unprecedented physics resolution/realism needed to help accelerate progress in delivering fusion energy.

1. Background

As the global energy economy makes the transition from fossil fuels toward cleaner alternatives, fusion becomes an attractive potential solution for satisfying the growing needs. Fusion energy, which is the power source for the sun, can be generated on earth, for example, in magnetically-confined laboratory plasma experiments (called "tokamaks") when the isotopes of hydrogen (e.g., deuterium and tritium) combine to produce an energetic helium "alpha" particle and a fast neutron, with an overall energy multiplication factor of 450:1. Building the scientific foundations needed to develop fusion power demands high-physics-fidelity predictive simulation capability for magnetically-confined fusion energy (MFE) plasmas. To do so in a timely way requires utilizing the power of modern supercomputers to simulate the complex dynamics governing MFE systems, including ITER, a multi-billion dollar international burning plasma experiment supported by 7 governments representing over half of the world's population. Currently under construction in France, ITER will be the world's largest tokamak system, a device that uses strong magnetic fields to contain the burning plasma in a doughnut-shaped vacuum vessel. In tokamaks, unavoidable variations in the plasma's ion temperature profile drive microturbulence, fluctuating electromagnetic fields, which can grow to levels that can significantly increase the transport rate of heat, particles, and momentum across the confining magnetic field. Since the balance between these energy losses and the self-heating rates of the actual fusion reactions will ultimately determine the size and cost of an actual fusion reactor, understanding and possibly controlling the underlying physical processes is key to achieving the efficiency needed to help ensure the practicality of future fusion reactors. The associated motivation drives the pursuit of sufficiently realistic calculations of turbulent transport that can only be achieved through advanced simulations. The present paper on advanced Particle-in-Cell (PIC) global simulations of plasma microturbulence at the extreme scale is accordingly associated with this fusion energy science (FES) grand challenge [1, 12].

Particle dynamics are well represented either by the 5D gyrokinetic (GK) equation (for low-frequency turbulence) or the 6D fully kinetic equation (for high frequency waves). The flagship GTC code and it's "co-design" partner GTC-P are massively parallel particle-in-cell (PIC) codes designed to carry out first principles, integrated simulations of thermonuclear plasmas, including the future burning plasma International Thermonuclear Experimental Reactor (ITER). These codes solve the 5D GK equation in full, global toroidal geometry to address kinetic turbulence issues in magnetically-confined fusion experimental facilities. GTC is the key production code for the fusion SciDAC Center "Gyrokinetic Simulation of Energetic Particle Turbulence and Transport (GSEP)" and for the Accelerated Application Readiness (CAAR) program at the Oak Ridge Leadership Computing Facility (OLCF). It is the only PIC code in the world fusion program capable of multiscale simulations of a variety of important physics processes in fusion-grade plasmas including microturbulence, energetic particle dynamics, collisional (neoclassical)

transport, kinetic magnetohydrodynamic (MHD) modes, and nonlinear radio-frequency (RF) waves. GTC interfaces with MHD equilibrium solvers for addressing realistic toroidal geometry features that include both axisymmetric tokamaks and non-axisymmetric stellarators. A recent upgrade enables this code to carry out global PIC simulations covering both the tokamak core and scrape-off layer (SOL) regions. It should also be noted that the current comprehensive version of GTC can carry out both perturbative (δ f) and non-perturbative (full-f) simulations with capability of dealing with kinetic electrons, electromagnetic fluctuations, multiple ion species, collisional (neoclassical) effects using Fokker-Planck collision operators, equilibrium current and radial electric field, plasma rotation, sources/sinks, and external antennae for auxiliary wave heating. Beyond the conventional application domain of gyrokinetic simulation of microturbulence, the GTC code has a long history in pioneering the development and application of gyrokinetic simulations of meso-scale electromagnetic Alfven eigenmodes excited by energetic particles (EP) in toroidal geometry. This is one of the most important scientific challenges that must be addressed in future burning plasma experiments such as ITER. Accordingly, the GTC work-scope has recently been extended to include simulation of macroscopic kinetic-MHD modes driven by equilibrium currents. The associated importance is that such efforts could ultimately lead to key knowledge needed to systematically analyze and possibly help avoid or mitigate highly dangerous reactor relevant thermonuclear disruptions.

The GTC-P code is a performance-optimized, highly portable modern PIC code that serves as a "co-design" proxy for the flagship GTC code. It serves to help accelerate progress on architecture-dependent optimization including scalability and portability for emerging exascale computers with heterogeneous architectures including both the GPU-accelerated Summit at ORNL and many-core system Aurora available in 2019 at ANL. The associated focus of GTC-P involves evaluation and implementation into GTC the emerging standards-based programming models that may enable performance portability across many-core and GPU-accelerated architectures.

The current transition from traditional superscalar processors to increasingly energy-efficient processors built from lightweight cores and a hierarchical memory architecture can be expected to be a trend that will persist into the next 5 years. Many-core processors (Intel KNL/KNH) and NVIDIA's GPU-accelerators provide a management technology that frees end users from needing to micromanage data movement and data locality. Over the past decade, strong collaborative interactions between Princeton and LBNL have delivered increasingly improved versions of GTC-P on some of the fastest supercomputers in the world including a series of GPU-accelerated systems and the first generation of Intel many-core coprocessors – culminating in the recent publication that also involved key contributions from ETH-Zurich [13]. The increased prominence of highly threaded architectures coupled with the desire to minimize memory usage has led to the need to implement novel domain decomposition, synchronization, and particle binning techniques. For example, the deployment of domain decomposition in the radial dimension has been demonstrated in GTC-P [13, 14] to dramatically reduce the memory footprint and the associated computational work for grid-based subroutines. This unique capability, which has not been implemented in most fusion codes, is targeted for implementation into GTC. More generally, the novel techniques developed in GTC-P including the use of floating-point atomics on GPUs, as well as the pragmatic approaches required for efficient vectorization on the Knights family will also be integrated into GTC. We will plan to continue our exploration of pipelined, one-sided communication (MPI or UPC++) in order to efficiently and productively implement

electron particle pushing and shifting. GTC-P is expected to have an increasingly enhanced role as the GTC co-design proxy for interaction with both the computational centers (OLCF at ORNL, ALCF at ANL, and NERSC at LBNL) and vendors (Intel, NVDIA, IBM), who can tweak their respective offerings for the computational requirements of GTC in the pre-exascale and exascale timeframes. This will all serve achieving the goal of efficiently carrying out architecture-dependent optimization of GTC kernels to help ensure performance portability across both large many-core and GPU systems.

2. Scientific Methodology

The GTC and GTC-P codes include all of the important physics and geometric features captured in numerous global PIC simulation studies of plasma size scaling over the years, extending from the seminal work in the Phys. Rev. Letter (PRL) by Z. Lin, et al. [5] up to the more recent PRL paper by B. F. McMillan, et al. [9] on system size effects on gyrokinetic turbulence. The current generally supported picture is that size-scaling follows an evolution from a "Bohm-like" trend where the confinement degrades with increasing system size, to a "Gyro-Bohm-like" trend where the confinement for JET-sized plasmas begins to "plateau" and then exhibits no further confinement degradation as the system size further increases toward ITER-sized plasmas. A number of physics papers over the past decade have proposed theories, such as turbulence spreading, to account for this transition to Gyro-Bohm scaling with plasma size for large systems. From a physics perspective, the main point in this paper is that this key decade-long fusion physics picture of the transition or "rollover" trend associated with toroidal ion temperature gradient micro-instabilities that are highly prevalent in tokamak systems, should be re-examined by modern supercomputing-enabled simulation studies which are now capable of being carried out with much higher phase-space resolution and duration. With a focused approach based on performance optimization of key functions within PIC codes in general, GTC-P, the "co-design" focus, has demonstrated the effective usage of the full power of current leadership class computational platforms worldwide at the petascale and beyond to produce efficient nonlinear PIC simulations that have advanced progress in understanding the complex nature of plasma turbulence and confinement in fusion systems for the largest problem sizes. Unlike fluid-like computational fluid dynamics (CFD) codes, GTC-P has concentrated on the fact that PIC codes are characterized by having less than 10 key operations which can then be an especially tractable target for advanced computer science performance optimization methods. As illustrated in [13], these efforts have resulted in accelerated progress in a discovery-science-capable global PIC code that models complex physical systems with unprecedented resolution and produces valuable new insights into reduction in "time-to-solution" as well as "energy to solution" on a large variety of leading supercomputing systems. In a sense, GTC-P is a "co-desing proxy" for the "flagship" electromagnetic GTC code which is the most comprehensive PIC code with respect to the complex physics included. GTC has delivered many scientific advances while using increasingly powerful supercomputing systems over the years. For example, it is the first large-scale fusion code to deliver production run stimulations at the terascale in 2002 [5] and on a petaflop system in 2009 [16]. Several key associated computational methodologies will be elaborated upon in the subsequent sections of this paper.

2.1. Global PIC Geometric Models

In plasma turbulence studies, the standard approach is to divide the physical quantities into an equilibrium part and a fluctuating part. The GTC code uses two set of meshes, one for the specification of the equilibrium and the other to represent fluctuating turbulent fields. In particular, the turbulence mesh is an unstructured field-aligned mesh for finite difference or finite element in 3D space.

The equilibrium quantities are governed by the Grad-Shafranov equation for toroidal geometry, while the fluctuating part is driven by various instabilities that lead to turbulent transport. Equilibrium magnetic configurations typically used in gyrokinetic simulations either come from: (i) analytic models such as the simple circular cross section or the Miller equilibrium; and (ii) numerical equilibrium codes such as EFIT or VMEC. For the rapidly evolving optimization studies that deliver very high resolution results from investigations of plasmas with increasing problem size on the most powerful supercomputing systems, the practical choice, as exemplified by the "co-design" GTC-P code – is the category (i) analytically-based equilibria. On the other hand, comprehensive production runs carried out by the flagship GTC code demand interfacing with the numerical equilibria of category (ii) that properly represent the actual experimental conditions.

The most accurate representation of the equilibrium in tokamaks is by using magnetic flux coordinates rather than Cartesian coordinates. This is due to the fact that most important equilibrium quantities, such as plasma temperature and density, can be shown to depend on the magnetic flux only. The flagship GTC code employs magnetic flux coordinates (Ψ, θ, ζ) to represent the electromagnetic fields and plasma profiles, where Ψ is the poloidal magnetic flux, θ is the poloidal angle, and ζ is the toroidal angle. Specifically, the inputs come from the numerical magnetic equilibrium and plasma profiles obtained from EFIT/VMEC by transforming the equilibrium quantities defined in the cylindrical coordinates (R, ϕ, Z) to those defined in the magnetic coordinates (Ψ, θ, ζ). The equilibrium data are provided by MHD equilibrium codes for the magnetic field strength B, and cylindrical coordinates (R, ϕ, Z) of points forming magnetic flux surfaces. Additionally, the flux functions representing poloidal g(Ψ) and toroidal I(Ψ) currents, magnetic safety factor q(Ψ), and minor radius r(Ψ) – defined as a distance from the magnetic axis along the outer mid-plane – are provided. First-order continuous B-splines are implemented for the 1D, 2D, and 3D functions to interpolate the complicated magnetic geometry and plasma profiles which provide a good compromise between high numerical confidence and reasonable computational efficiency.

The GTC capability to carry out simulations of problems with general toroidal geometry has recently been extended to also include non-axisymmetric configurations. For non-axisymmetric devices, the equilibrium data are presented on the uniform (Ψ, θ) grid for all n=(1, 2, $\cdots$, N) toroidal harmonics. To reduce the computational load and memory usage, the transformation of non-axisymmetric variables into spline functions of ζ is chosen for implementation in GTC, with spline coefficients associated with a particular grid point ζ_i being stored by processors with corresponding toroidal rank using message passing interface (MPI) parallelization.

The GTC-P code deploys the so-called large aspect ratio equilibrium, which is an analytical model describing a simplified toroidal magnetic field with a circular cross-section. The associated model takes into account the key geometric and physics properties needed to carry out a meaningful study of the influence of increasing plasma size on magnetically-confined fusion plasmas. Such an approach enables working with a sufficiently straightforward but nevertheless

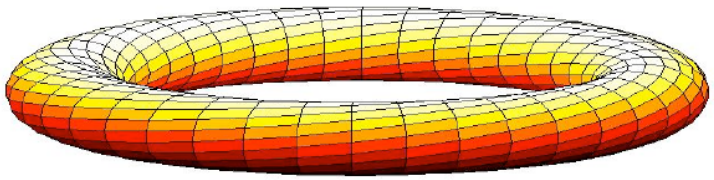

Figure 1. Illustrative Figure showing the grid structure of the GTC-P code on a 3D torus

discovery-science-capable physics [4,5] code that makes more tractable the formidable task of developing the algorithmic advances needed to take advantage of the rapidly evolving modern platforms featuring, for example, both homogenous and hybrid architectures. The associated physics approach is to deploy GTC-P plasma size-scaling studies because it is a fast streamlined modern code with the capability to efficiently carry out computations at extreme scales with unprecedented resolution and speed on present-day multi-petaflop computers [13]. The corresponding scientific goal is to accelerate progress toward capturing new physics insights into the key question of how turbulent transport and associated confinement characteristics scale from present generation laboratory plasmas to the much larger ITER-scale burning plasmas. This includes a systematic characterization of the spectral properties of the turbulent plasma as the confinement scaling evolves from a "Bohm-like" trend where the confinement degrades with increasing system size to a "Gyro-Bohm-like" trend where the confinement basically "plateaus", exhibiting no further confinement degradation as the system size further increases. "Lessons learned" achieved in a timely way from this co-design effort can be expected to expedite associated advances in the flagship GTC code in particular as well as to generally providing valuable information on PIC performance modeling advances to ongoing and future efforts in improving PIC code deployment on multi-petaflop supercomputers on the path to exascale and beyond.

2.2. Global PIC Grid Considerations

To accurately track the key physics in magnetically-confined toroidal plasmas, the GTC and GTC-P codes utilize a highly specialized grid that follows the magnetic field lines as they twist around the torus (see Figure 1). This allows the code to retain the same accuracy while using fewer toroidal planes than a regular, non-field-aligned grid. From relevant physics considerations, since short wavelength waves parallel to the magnetic field are suppressed by Landau damping, increasing the grid resolution in the toroidal dimension will leave the results essentially unchanged. Consequently, a typical production simulation run usually consists of a constant number of poloidal planes (e.g., 32 or 64) wrapped around the torus. Each poloidal plane is represented by an unstructured grid, where the grid sizes in the radial and poloidal dimensions correspond approximately to the size of the gyro-radius of the particles. As we consider larger plasma sizes (e.g., 2x in major and minor radius), the number of grid points in each 2D plane increases 4x. The number of grid points for a 3D grid increases 4x as well since the number of planes in the toroidal dimension remains the same for all problem sizes. For a modest-sized fusion device (e.g, the DIII-D tokamak at General Atomics in San Diego, CA), the associated plasma simulation typically uses ∼128 thousand grid points in a 2D plane. As we move to the larger Joint European Torus (JET) device and then eventually to the ITER size plasmas, the number of grid points increases 4x and 16x, respectively. Using a fixed number of 64 toroidal planes, the total number of grid points for an ITER-sized plasma will be ∼131 million. With 100

particles per cell resolution, an ITER-sized simulation will accordingly involve ∼13 billion particles. Tracking the dynamics of this large number of particles would of course be an extremely daunting task without access to leadership-class supercomputers.

3. Programming Approach

The basic parallel programming approach for global PIC codes such as GTC and GTC-P includes: (i) explicit message passing using MPI; (ii) architecture-specific models such as CUDA for computing on GPUs; and (iii) directive-based compiler options such as OpenMP and OpenACC with possible promise of being more cross-machine portable between architectures.

A more detailed recent description of global PIC code characteristics/considerations with respect to scalability, performance, portability, modern computational platforms, and external libraries, associated discussions will touch on the rationale for the chosen programming approach, and the associated balance between performance and portability can be found in [13]. In future R&D, attention must of course be focused on the many specific challenges for global PIC applications in achieving efficiency on exascale architectures. A preview is given in the following sections.

3.1. PIC Scalability

In describing efforts to improve the performance scalability of the global PIC codes – well represented by GTC and GTC-P, key topics include: (i) on-node thread scaling; and (ii) between node scaling. The GTC/GTC-P codes have been designed with four levels of parallelism: (i) an inter-node distributed memory domain decomposition via MPI, (ii) an inter-node distributed memory particle decomposition via MPI, (iii) an intra-node shared memory work partition implemented with OpenMP; and (iv) a SIMD vectorization within each core. This approach was shown to lead to nearly-perfect scaling with respect to the number of particles [2].

In order to efficiently address large grid sizes and the associated significant memory increase, the domain decomposition in GTC-P is further extended in the radial dimension (beyond the toroidal dimension) [14]. This leads to a 2D domain decomposition and enables carrying out true weak scaling studies, where both particle and grid work are appropriately scaled. The multi-level particle and domain decompositions provide significant flexibility in distributed-memory task creation and layout. While the ranks in the toroidal dimension are usually fixed as 32 or 64 due to Landau damping physics, there is freedom to choose any combination of process partitioning along the radial and particle dimensions. For scaling with a fixed problem size, the procedure involves first partitioning along the radial direction and then switching to particle decomposition for additional scalability. The decompositions were implemented with three individual communicators in MPI (toroidal, radial, and particle communicator), and further tuning is made available via options to change the order of MPI rank placement.

It is important to note that gyrokinetic PIC simulations typically exhibit highly anisotropic behavior – with the velocity parallel to the magnetic field being an order of magnitude larger than that in the perpendicular direction. Consequently, the message sizes in the toroidal dimension can be 10 times larger than those in the radial dimension at each time step. On the IBM Blue Gene systems with explicit process mapping, it was found to be convenient and effective to group processes to favor the MPI communicator in the toroidal dimension. For other systems,

assigning consecutive ranks for processes within each toroidal communicator generally leads to improved performance.

Looking toward the ongoing and future challenge of maximizing on-node performance and efficiency, it is already clear that modern processor architectures have evolved with more cores and wider vector units in a single node. In order to fully exploit the emerging architectures on the path to exascale, it is important that application scientists design their software such that the algorithms and the implementations map well on the hardware for maximum scalability. In GTC/GTC-P, this translates to multicore parallelism using shared-memory multi-threading and implementation changes to enable SIMD vectorization. For example, in an earlier version of GTC-P, "holes" were used to represent non-physical "invalid" particles, i.e., in a distributed environment, at every time step, the particles that are being moved to other processes are marked as "holes" and considered to be "invalid" in the local particle array. These invalid particles are then removed from the array periodically to empty memory space for new incoming particles. In this type of implementation, two particles in consecutive memory locations may have different operations in charge and push depending on if they belong to the same type of particles (valid or invalid) or not. This accordingly introduces difficulty for automatic vectorization. To maximize the usage of vector units, the latest version of GTC-P includes removing the holes completely for charge and push by filling the holes at the end of shift and using the new incoming particles sent from neighboring processors at every time step. If a process has sent more particles than received, then the remaining holes are filled with the last particles in the array. A similar strategy has been applied for the GPU implementation to remove the branch statement caused by the "holes".

3.2. PIC Performance Challenges

PIC algorithms are challenging to optimize on modern computer architectures due to issues such as data conflict and data locality. In GTC and GTC-P, parallel binning algorithms have been developed to improve data locality for charge and push. More specifically, several choices are provided to bin the particles, i.e., along the radial dimension and along the poloidal dimension. The best binning strategy will be used for production runs by first running a few benchmarks. In GTC-P, the additional use of intrinsics has helped improve the vectorization of the binning implementation. On GPU's, the CUDA version of the binning algorithm was implemented using the Thrust Library.

To address the data conflict issue in charge, optimization strategies have been explored via static replication of grid segments that are coupled with synchronization via atomics, where the size of the replica may be traded for increased performance [7, 8]. The best performance is often obtained by employing the full poloidal grid for each OpenMP thread. In GTC with only toroidal domain decomposition, the full poloidal grid replication dramatically increases the temporary grid-related storage for large size grid on manycore architectures such as the Intel Xeon Phi systems. As such, static replication of grid segments that are coupled with synchronization via atomics will likely be the best strategy. In GTC-P, the radial domain decomposition solves locality and memory pressure without resorting to costly atomics. In essence, since only a small segment of the full poloidal grid is required for a hazard-free charge deposition, the private grid replication strategy can be readily employed on a per thread basis for the best performance.

In dealing with heterogenous supercomputing platforms such as "Titan", the approach followed in the deployment of global PIC codes involves off-loading the computationally intensive

and highly scalable subroutines to GPU's,while the communication-dominant subroutines remain on CPU's [4]. Performance, however, is known to be impeded due to the synchronization of atomic operations and the unavoidable memory transpose associated with the structure-of-array to array-of-structure data layout. To address this issue, the time-consuming global memory atomic operations have been replaced with local shared memory atomic operation. This R&D activity falls generally in the category of advances and challenges involving heterogeneous architectures.

3.3. Portability

Global PIC codes such as GTC and GTC-P have demonstrated increasing capability for portability over the past few years across different architectures. In this section the associated techniques applied for doing so are discussed along with examples of success achieved. In general, a high a priority is being placed on portability in HPC because of the significant differences between quite different main-line approaches receiving heavy emphasis and by government investments, a prominent example being the major architectural differences between the upcoming 200 PF systems: the SUMMIT system at the OLCF and the AURORA system at the ALCF. Since both approaches have significant exciting potential for enabling accelerated performance at scale, most advanced applications, including prominent global PIC codes such as GTC/GTC-P, will continue to focus attention on achieving both performance enhancement as well as portability. For example, performance portability of these advanced codes helps ensure, in a risk mitigation sense, the capability to perform very well on whichever platform proves to provide the greater eventual computing at extreme scale advantage.

GTC-P has been particularly successful in porting modern optimized versions across a wide range of multi peta-flop platforms at full or near-to-full capability. Benefit is associated in part from the fact that GTC-P is not critically dependent on any third-party libraries. For example this effort was initiated with the implementation a highly-optimized Poisson solver with multithreading capability. Additional performance enhancement for both GTC and GTC-P has been obtained by utilizing a specialized damped Jacobi iterative solver [6]. In this iterative solver, the damping parameter was carefully chosen to favor the desired range of wavelengths for the fastest growing modes in plasma turbulence simulations. As a result, a small and fixed iteration count is sufficient to achieve the desired accuracy.

Although achieving the best performance on each explored architecture requires platform-specific optimization strategies, a "pluggable" software component approach in architecting the GTC/GTC-P application codes have proven to be a quite successful approach. Specifically, the interface is preserved across all implementations targeting CPU-based codes as well as GPU (or Xeon Phi) hybrid implementations. Components are chosen based on the target platform during the application build process. This enables having a unified code base with the best-possible performance, without sacrificing portability. Behind the unified interface, platform-specific optimization strategies are systematically investigated.

Some optimizations, such as sorting particles and vectorization, are common to all platforms, but implementation details differ. Other optimizations, such as handling NUMA issues and load imbalance, are specific to certain platforms. Designing routine interfaces is of crucial importance to allow portability without compromising performance-tuning opportunities.

GTC-P uses the MPI-3 standard for distributed-memory communication, including the exploration of explore one-sided communication. The motivation here is again to provide better portability for diverse architectures and programming models.

Significant advances in GTC-P on many-core processors with respect to portability and scalability have been recently achieved by porting the code to GPU systems with OpenACC 2.0 as a viable option instead of CUDA. This has led to the very recent success in porting and optimizing an OpenACC 2.0 version of GTC-P on the Sunway TaihuLight Supercomputer [15] – the new No. 1 system on the international Top500 as of June 2016. The only approach for achieving good performance on TaihuLight requires software compatibility with their SWACC compiler, a customized OpenACC 2.0 syntax supported software.

In common with the large majority of codes in the fusion energy science/plasma physics application domain, GTC-P was originally written in Fortran language. However, to better facilitate interdisciplinary collaborations with computer science and applied math colleagues, modern versions of this code have been developed in C language as well as a CUDA implementation for dealing with GPU's. As just noted, this capability has recently been further advanced with the development and implementation of an OpenACC 2.0 version of GTC-P. Although the original Fortran version of this code is still used for verification purposes in cross-checking and benchmarking results, the primary utilization has involved the C and CUDA versions for performance studies and physics production runs on supercomputing systems such as the ALCF's "Mira" and the OLCF's "Titan".

4. Scaling Results

Using resources from INCITE, previous early Science Projects (ESP) at the ALCF, and Director's Discretionary allocations from both the ALCF and OLCF in the past few years, GTC-P has demonstrated excellent scalability to more than 100,000 cores on leadership computing facilities at ANL and ORNL. It has been successfully deployed for major scientific production runs on the IBM BG/Q/"Mira", where the excellent weak scaling performance was carried over to much larger scale on LLNL's more powerful Sequoia system. These results are illustrated in Figure 2.

In addition, it is relevant to note that the GTC-P code was the featured U.S. code in the G8 international exascale project in nuclear fusion energy, "NuFuSE" (http://www.nu-fuse.com/) that was supported in the U.S. by the National Science Foundation (NSF) [10]. The G8 program helped provide unique access to a variety of international leadership class computational facilities such as the Fujitsu K Computer in Japan. Results from weak-scaling studies carried out on the K-computer are illustrated in Figure 3.

As seen from subsequent stimulating new results [11], substantive impact can be expected to help stimulate progress in preparing for actual research engagement on ITER. In order to do so in a timely way, it is critically important that new software for extreme concurrency systems that demand increasing data locality be developed to help accelerate progress toward the ultimate goal of computational fusion research, a predictive simulation capability that is properly validated against experiments in regimes relevant for practical fusion energy production.

Having demonstrated the ability to effectively utilize the most powerful homogeneous supercomputing platforms worldwide, GTC-P R&D efforts have also examined performance characteristic on heterogeneous architectures. More generally, these studies represent productive investigations of extreme scale science across advanced scientific computing basic research pro-

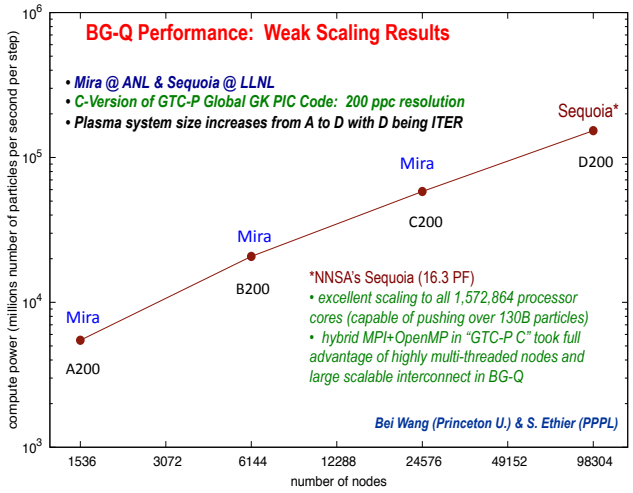

Figure 2. GTC-P Code Performance on World-Class IBM BG/Q Systems

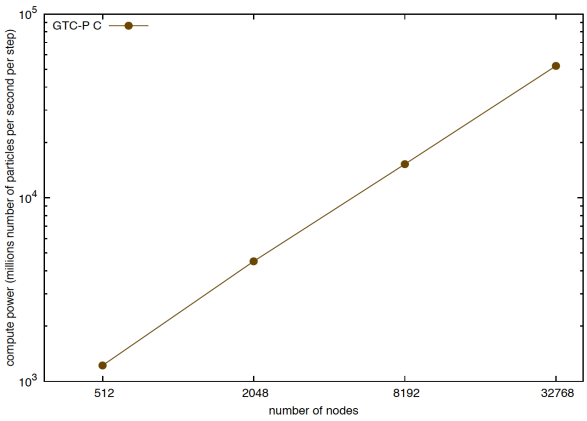

Figure 3. Weak Scaling of the GTC-P code Achieved on the Fujitsu-K Computer in Japan.

grams with fusion energy science as an illustrative application domain. Here the focus was on developing new algorithms for advanced heterogeneous supercomputing systems such as the GPU/CPU "Titan" at DOE's OLCF and the Intel Xeon Phi/Intel Xeon "Stampede" at NSF's TACC. In doing so, a new version of GTC-P code was developed that features algorithms which include new heterogeneous capabilities for deployment on hybrid GPU (Nvidia K20)/CPU as well as the Intel Xeon Phi/Intel Xeon systems such as Stampede and also TH-2 in China. From a verification perspective, this research effort also includes systematic comparison of new results against the successful work described in earlier in studies that featured high resolution, long temporal scale simulation results obtained on world-class homogeneous systems such as the IBM BG/Q Mira at the ALCF, Sequoia at LLNL, and the K-Computer in Kobe, Japan. A weak scaling performance of GTC-P across a wide-range of systems is shown in Figure 4. Interested readers can also find more details in [13].

The CAAR program has enabled GTC architecture-dependent optimization including scalability and portability for heterogeneous architectures. Figure 5 has shown the weak scaling of GTC on Titan up to 16384 nodes. Compared with CPU (16 cores AMD 6274), GPU (NVIDIA

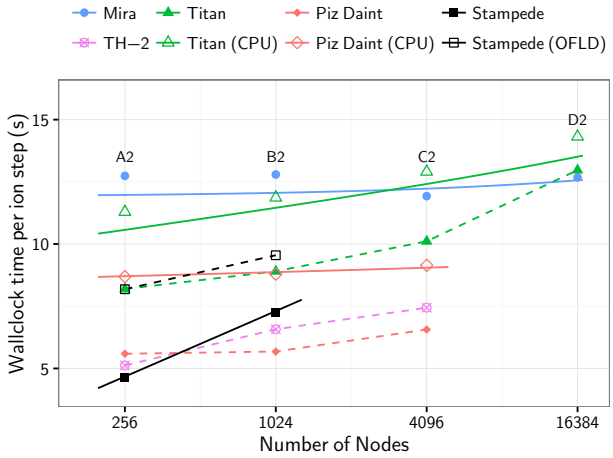

Figure 4. GTC-P weak scaling performance using a fixed problem size per node across all systems allows comparisons of node performance. Solid lines indicate model-predicted running times (shown for Mira, Titan (CPU), Piz Daint (CPU), Stampede), dashed line joins actual running times.

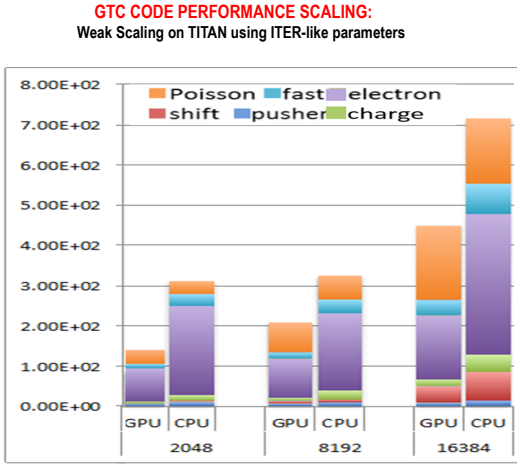

Figure 5. GTC hybrid weak scaling study where we scale both the grid size and total number of particles on Titan up to 16384 nodes

K20x) has boosted the overall performance by 1.5-2.8x. The decrease of the performance speed up in large processor counts is due to the increased portion of non-GPU accelerated subroutines.

5. Time-to-Solution and Energy-to-Solution Comparative Studies

As evident from the increasingly strong attention paid to "Green 500" in addition to the traditional "Top 500" supercomputer rankings, energy is clearly being recognized as a prominent impediment to advances in HPC development. Since the interplay between performance and power is highly dependent on algorithm and architecture, assessing the net energy efficiency of large scientific simulations can be particularly non-intuitive as one moves from one processor or network architecture to the next. In Table 1 (also shown in Table V of [13]), the energy per time step was illustrated for 4K nodes of Mira, Titan, and Piz Daint when using 80M

Table 1. Energy per ion time step (KWh) by platform for the weak-scaled, kinetic electron configuration at 4096 nodes. Power is obtained via system instrumentation including compute node, network, blades, AC to DC conversion.

	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

grid points, 8B ions, and 8B electrons. The power measured under actual load via system instrumentation was used. Attention should be paid to the fact that although Mira required the most wall clock time per time step, it also required the least power per node. Taken as whole, the conclusion was that Mira required the least energy per time step of all platforms considered. Conversely, using the host-only configurations on Titan and Piz Daint required between 2x and 4x the energy with the difference largely attributable to the relative lack of scalability on Titan. It is especially interesting to highlight that while code acceleration on these platforms significantly reduced the wall clock time per time step, the associated power expended was only slightly increased. Consequently, the energy required for the GPU-accelerated systems was reduced nearly proportionally with run time. It was also pointed out in [13] that if the CPU frequency is scaled down, the energy-to-solution estimates here could be at odds with time-to-solution when we use technologies such as DVFS [3]. When inter-node communication dominates the execution time, we observed up to 25% reduction in energy consumption together with an increase in the execution time by 28%. In particular, an experiment on an Intel Haswell-based Cray XC40 system was carried out. Although supported by much hardware, DVFS control was not actually enabled on most of the systems studied. As such, it was not possible to explore such optimization on all systems. It is important to highlight the fact that another impediment to adopting such technology is the policy adopted for resource allocation by HPC compute facilities – that is based on cpu-hours rather than energy consumption. Accordingly, such a policy of course makes the most performant solution the best from the user perspective.

With regard to energy-efficient scientific computing, instrumenting scientific applications to measure energy when running on large supercomputing installations today can be cumbersome and obtrusive – requiring significant interaction with experts at each center. As such, most applications have little or no information on energy-to-solution across the architecture design space spectrum. In order to affect energy-efficient co-design of supercomputers, energy measurement must always-on by default with, at a minimum, total energy and average power reported to the user at the end of an application. By reporting energy by component (memory, processor, network, storage, etc...), scientists and vendors could co-design their applications and systems to avoid energy hotspots and produce extremely energy-efficient computing systems.

6. Concluding Comments

Portability across many-core and GPU-accelerated architectures is important to the success of this illustrative PIC GTC framework highlighted in this paper. This discovery science software will have a large user community and needs to efficiently utilize all computational resources of the next generation computers. Further studies will continue to explore the efficacy of using

the OpenMP 4.5 offload model and the use of OpenACC to provide performance portability across architectures in lieu of using both CPU-specific OpenMP implementations and GPU-specific OpenACC implementations. We note that the performance portability of OpenACC has not been studied on Intel's KNL/KNH processors and the OpenMP offload model is not fully supported on many compilers. Moreover, given the offload model has been rendered irrelevant by KNL's self-hosted nature coupled with its use of MCDRAM as a hardware-managed cache, there is a distinct possibility that standards-based single-source performance portability will prove elusive and some specialization will remain essential. We will continue to leverage GTC-P as GTC's co-design proxy for interaction with both the computational centers (OLCF at ORNL, ALCF at ANL, and NERSC at LBNL) and vendors (Intel, NVDIA, IBM), who can tweak their respective offerings for the computational requirements of GTC in the pre-exascale and exascale timeframes.

As a final comment, it is appropriate to note that a broader impact of the work presented in this paper is the delivery of benefits to particle-in-cell codes in general since the associated codes share a common algorithmic foundation.For example, the continuing developments targeted in the global PIC GTC project can be expected to have a strong impact in dealing with the multi-threading challenges for efficient deployment of a large number of processors on modern homogeneous and heterogeneous systems advances that should prove beneficial to any particle-mesh algorithm.

Acknowledgements: The authors are very grateful to our many collaborative colleagues. We are especially indebted to Sam Williams and Khaled Ibrahim from LBNL.

References

1. R. Rosner etc. Opportunities and challenges of exascale computing - doe advanced scientific computing advisory committee report, 2010. `https://science.energy.gov/~/media/ascr/ascac/pdf/reports/Exascale_subcommittee_report.pdf`.

2. S. Ethier, W. M. Tang, and Z. Lin. Gyrokinetic particle-in-cell simulations of plasma microturbulence on advanced computing platforms. *Journal of Physics: Conference Series*, 16:1–15, 2005.

3. D. Hackenberg, R. Schne, T. Ilsche, D. Molka, J. Schuchart, and R. Geyer. An energy efficiency feature survey of the intel haswell processor. *The IEEE International Parallel and Distributed Processing Symposium Workshop (IPDPSW)*, pages 896–904, May 2015.

4. Khaled Z Ibrahim, Kamesh Madduri, Samuel Williams, Bei Wang, Stephane Ethier, and Leonid Oliker. Analysis and optimization of gyrokinetic toroidal simulations on homogenous and heterogenous platforms. *International Journal of High Performance Computing Applications*, 2013.

5. Z. Lin, S. Ethier, T. S. Hahm, and W. M. Tang. Size scaling of turbulent transport in magnetically confined plasmas. *Phys. Rev. Lett.*, 88:195004, Apr 2002.

6. Z. Lin and W. W. Lee. Method for solving the gyrokinetic poisson equation in general geometry. *Phys. Rev. E*, 52:5646–5652, Nov 1995.

7. K. Madduri, Khaled Z. Ibrahim, Samuel Williams, Eun-Jin Im, Stephane Ethier, John Shalf, and Leonid Oliker. Gyrokinetic toroidal simulations on leading multi- and manycore HPC systems. In *Proc. Int'l. Conf. for High Performance Computing, Networking, Storage and Analysis (SC '11)*, pages 23:1–23:12, New York, NY, USA, 2011. ACM.

8. K. Madduri, S. Williams, S. Ethier, L. Oliker, J. Shalf, E. Strohmaier, and K. Yelick. Memory-efficient optimization of gyrokinetic particle-to-grid interpolation for multicore processors. In *Proc. ACM/IEEE Conf. on Supercomputing (SC 2009)*, pages 48:1–48:12, November 2009.

9. B. F. McMillan, X. Lapillonne, S. Brunner, L. Villard, S. Jolliet, A. Bottino, T. Görler, and F. Jenko. System size effects on gyrokinetic turbulence. *Phys. Rev. Lett.*, 105:155001, Oct 2010.

10. NuFuSE. `http://www.nu-fuse.com/`.

11. W. Tang, Bei Wang, and S. Ethier. Scientific discovery in fusion plasma turbulence simulations at extreme scale. *Computing in Science Engineering*, 16(5):44–52, Sept 2014.

12. William Tang and David Keyes. Scientific grand challenges: Fusion energy science and the role of computing at the extreme scale. In *PNNL-19404*, page 212, 2009.

13. William Tang, Bei Wang, Stephane Ethier, Grzegorz Kwasniewski, Torsten Hoefler, Khaled Z. Ibrahim, Kamesh Madduri, Samuel Williams, Leonid Oliker, Carlos Rosales-Fernandez, and Tim Williams. Extreme scale plasma turbulence simulations on top supercomputers worldwide. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '16, pages 43:1–43:12, Piscataway, NJ, USA, 2016. IEEE Press.

14. Bei Wang, Stephane Ethier, William Tang, Timothy Williams, Khaled Z. Ibrahim, Kamesh Madduri, Samuel Williams, and Leonid Oliker. Kinetic turbulence simulations at extreme scale on leadership-class systems. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, pages 82:1–82:12, New York, NY, USA, 2013. ACM.

15. Yichao Wang, James Lin, Linjin Cai, William Tang, Stephane Ethier, Bei Wang, Simon See, and Satoshi Matsuoka. Porting and optimizing gtc-p on taihulight supercomputer with sunway openacc. In *HPC China*, 2016.

16. Yong Xiao and Zhihong Lin. Turbulent transport of trapped-electron modes in collisionless plasmas. *Phys. Rev. Lett.*, 103:085004, Aug 2009.