Petascale Parallelization of the Gyrokinetic Toroidal Code

Stéphane Ethier[1], Mark Adams[2], Jonathan Carter[3], Leonid Oliker[3]

[1] Princeton Plasma Physics Laboratory, Princeton University, Princeton NJ 08453
[2] APAM Department, Columbia University, New York NY 10027
[3] NERSC/CRD Lawrence Berkeley National Laboratory, Berkeley CA, 94720
ethier@pppl.gov, mark.adams@columbia.edu, jtcarter@lbl.gov, loliker@lbl.gov

Abstract. The Gyrokinetic Toroidal Code (GTC) is a global, three-dimensional particle-in-cell application developed to study microturbulence in tokamak fusion devices. The global capability of GTC is unique, allowing researchers to systematically analyze important dynamics such as turbulence spreading. In this work we examine a new radial domain decomposition approach to allow scalability onto the latest generation of petascale systems. Extensive performance evaluation is conducted on three high performance computing systems: the IBM BG/P, the Cray XT4, and an Intel Xeon Cluster. Overall results show that the radial decomposition approach dramatically increases scalability, while reducing the memory footprint — allowing for fusion device simulations at an unprecedented scale.
Research Topics: Large scale simulations in CS&E, parallel and distributed computing, performance analysis

1 Introduction

After a decade where high-end computing (HEC) was dominated by the rapid pace of improvements to processor frequencies, the performance of next-generation supercomputers is increasingly differentiated by varying interconnect designs and levels of integration. Understanding the tradeoffs of these system designs is a key step towards making effective petascale computing a reality. In this work, we examine a new parallelization scheme for the Gyrokinetic Toroidal Code (GTC) [5] micro-turbulence fusion application. Extensive scalability results and analysis are presented on three HEC systems: the IBM BlueGene/P (BG/P) at Argonne National Laboratory, the Cray XT4 at Lawrence Berkeley National Laboratory, and an Intel Xeon cluster at Lawrence Livermore National Laboratory. Overall results indicate that the new radial decomposition approach successfully attains unprecedented scalability to 131,072 BG/P cores by overcoming the memory limitations of the previous approach. The new version is well suited to utilize emerging petascale resources to access new regimes of physical phenomena.

2 Architectural Testbed

We have chosen three architectures which are often deployed at high-performance computing installations, the Cray XT, IBM BG/P, and an Intel/Infiniband clus-

Core Architecture	**AMD Budapest**	**Intel Core2**	**IBM PowerPC 450**
Type	superscalar out-of-order	superscalar out-of-order	dual issue in-order
Clock (GHz)	2.30	2.66	0.85
DP Peak (GFlop/s)	9.20	10.66	3.40
Private L1 Data Cache	64 KB	32 KB	32 KB
Private L2 Data Cache	512 KB	—	—
Socket/Node Architecture	**Opteron 1356 Budapest**	**Xeon E5430 Harpertown**	**BG/P Chip**
Cores per Socket	4	4 (MCM)	4
Shared Cache	2 MB L3	2×6 MB L2	8 MB L2
Sockets per SMP	1	2	1
Node DP Peak (GFlop/s)	36.8	85.33	13.60
DRAM Bandwidth (GB/s)	12.80	21.33(read) 10.66(write)	13.60
Node DP Flop:Byte Ratio	2.88	2.66	1.00
System Architecture	**Cray XT4 Franklin**	**Intel Cluster Hyperion**	**IBM BG/P Intrepid**
Interconnect	Seastar2+/ 3D Torus	Infiniband 4xDDR	3D Torus/ Fat Tree
Total Nodes	9660	576	40960
MPI Bandwidth (GB/s)	1.67	0.37	1.27

Table 1. Highlights of CPU and node architectures for examined platforms. MPI bandwidth is measured using unidirection MPI benchmarks to exercise the fabric between nodes with 0.5MB messages.

ter. In selecting the systems to be benchmarked, we have attempted to cover a wide range of systems having different interconnects. The Cray XT is designed with tightly integrated node and interconnect fabric. Cray has opted to design a custom network ASIC and messaging protocol and couple this with a commodity AMD processor. In contrast, the Intel/IB cluster is assembled from off the shelf high-performance networking components and Intel server processors. The final system, BG/P, is custom designed for processor, node and interconnect with power efficiency as one of the primary goals. Together these represent the most common design tradeoffs in the high performance computing arena. Table 1 shows the CPU and node architectures, as well as the size and topology of the three system interconnects.

Franklin Cray XT4: Franklin, a 9660 node Cray XT4 supercomputer, is located at Lawrence Berkeley National Laboratory (LBNL). Each XT4 node contains a quad-core 2.3 GHz AMD Opteron processor, which is tightly integrated to the XT4 interconnect via a Cray SeaStar2+ ASIC through a HyperTransport 2 interface capable of capable of 6.4 GB/s. All the SeaStar routing chips are interconnected in a 3D torus topology with each link is capable of 7.6 GB/s peak bidirectional bandwidth, where each node has a direct link to its six nearest neighbors. Typical MPI latencies will range from 4-8μs, depending on the size of the system and the job placement.

Intrepid IBM BG/P: Intrepid is a BG/P system located at Argonne National Laboratory (ANL) with 40 *racks* of 1024 nodes each. Each BG/P node has 4 PowerPC 450 CPUs (0.85 GHz) and 2GB of memory. BG/P implements three high-performance networks: a 3D torus with a peak bandwidth of 0.4 GB/s per link (6 links per node) for point to point messaging; a collectives network for broadcast and reductions with 0.85 GB/s per link (3 links per node); and a network for a low-latency global barrier. Typical MPI latencies will range from 3-10μs, depending on the size of the system and the job placement.

Hyperion Intel Xeon Cluster: The Hyperion cluster, located at Lawrence Livermore National Laboratory (LLNL), is composed of four scalable units, each consisting of 134 dual-socket nodes utilizing 2.5 GHz quad-core Intel Harpertown processors. The nodes within a scalable unit are fully connected via a 4$\times$ IB DDR network with an peak bidirectional bandwidth of 2.0 GB/s. The scalable units are connected together via spine switches providing full bisection bandwidth between scalable units. Typical MPI latencies will range from 2-5μs, depending on the size of the system and the job placement.

3 GTC: Turbulent Transport in Magnetic Fusion

GTC is a 3D particle-in-cell (PIC) code developed to study the turbulent transport properties of tokamak fusion devices (see Figure 1) from first principles [2,5]. The current production version of GTC scales well with the number of particles on the largest systems available. It achieves this by using multiple levels of parallelism: a 1D domain decomposition in the toroidal dimension (long way around the torus geometry), a multi-process particle distribution within each toroidal domain, and shared memory multitasking at the loop level via OpenMP directives. The 1D domain decomposition and particle distribution are implemented with MPI using 2 different communicators: a toroidal communicator to move particles from one domain to another, and an intra-domain communicator to gather the contribution of all the particles located in the same domain. Communication involving all the processes is kept to a minimum.

In the PIC method, a grid-based field is used to calculate the interaction between the charged particles instead of evaluating the N^2 direct binary Coulomb interactions. This field is evaluated by solving the gyrokinetic Poisson equation [4] using the particles' charge density accumulated on the grid. The basic steps of the PIC method are: *(i)* Accumulate the charge density on the grid from each particle to its nearest grid points. *(ii)* Solve the Poisson equation to evaluate the field. *(iii)* Gather the field values at the position of the particles. *(iv)* Advance particles one time step using equations of motion ("push" step). The most expensive steps are the charge accumulation and particle "push", which account for about 80% to 85% of the time for most realistic experiments. The memory usage depends strongly on the problem size, essentially the number of grid points and particles. In a typical simulation there are anywhere between 2 to 20 particles per grid point per MPI process, and the same factor roughly applies to the amount of memory used by each.

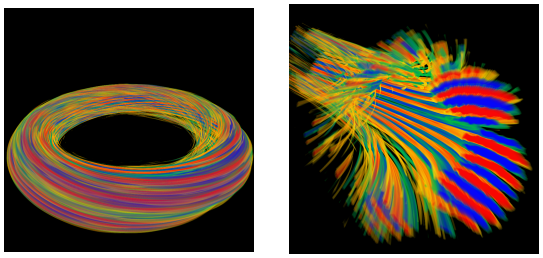

Fig. 1. Volume rendering of the electrostatic potential field created by the plasma particles in a GTC simulation; (left) shows the whole volume, and (right) a cross-section through a poloidal plane where elongated eddies of the turbulence can be seen. (Visualization by K.-L. Ma, UC Davis)

3.1 Radial Decomposition Implementation

In the described GTC version, the local grid within a toroidal domain is replicated on each MPI process within that domain and the particles are randomly distributed to cover that whole domain. The grid work, which comprises of the field solve and field smoothing [1], is performed redundantly on each MPI process in the domain. Only the particle-related work is fully divided between the processes. This has not been an issue until recently, since the grid work is small when using a large number of particles per cell. However, when simulating large fusion devices, such as the international experiment ITER [3], which will be 8 times larger in volume than the largest fusion device currently in existence, a much larger mesh must be used to fully resolve the microturbulence physics. All the replicated copies of the local grid on the processes within a toroidal domain make for a proportionally large memory footprint. With only a small amount of memory left on the system's nodes, only a relatively small number of particles per cell per process can fit. This problem is particularly severe on the IBM BG/P system where the amount of memory per core is relatively small. Eventually, the grid work starts dominating the calculation even if a very large number of processor cores is used.

The solution to our non-scalable grid work problem was to add another level of domain decomposition to the existing toroidal decomposition. Although one might think that a fully 3D domain decomposition is the ideal choice, the dynamics of magnetically confined charged particles in tokamaks tells us otherwise. The particle motion is very fast in both the toroidal and poloidal (short way around the torus) directions, but is fairly slow in the radial direction. In the toroidal direction, the domains are large enough that only 10% of the particles, on average, leave their domain per time step in spite of their high velocities. With about one thousand processes per toroidal domain dividing the local grid, the poloidal domains end up being much smaller, leading to a high level of communication due to a larger percentage of particles moving in and out of the domains at each step. Furthermore, the poloidal grid points are not aligned with each other in the radial direction, which precludes the use of a simple algebraic equation to locate the particles when they move between domains. The cross-sectional grid consists

of concentric circles with constant arc length between the grid point. The radial grid has the advantage of being regularly spaced and easy to split into several domains. The slow average velocity of the particles in that direction ensures that only a small percentage of them will move in and out of the domains, which is what we observe.

One disadvantage, however, is that the radial width of the domains needs to decrease with the radius in order to keep a uniform number of particles in each domain since the particles are uniformly distributed across the whole volume. This essentially means that each domain will have the same volume but a different number of grid points. For a small grid having a large number of radial domains, it is possible that a domain will fall between two radial grid points. Another disadvantage is that the domains require a fairly large number of ghost cells, from 3 to 8 on each side, depending on the maximum velocity of the particles. This is due to the fact that our particles are not point particles but rather charged "rings", where the radius of the ring corresponds to the Larmor radius of the particle in the magnetic field. We actually follow the guiding center of that ring as it moves about the plasma, and the radius of the ring changes according to the local value of the magnetic field. A particle with a guiding center sitting close to the boundary of its radial domain can have its ring extend several grid points outside of that boundary. We need to take that into account for the charge deposition step since we pick four points on that ring and split the charge between them [4]. As for the field solve for the grid quantities, it is now fully parallel and implemented with the Portable, Extensible Toolkit for Scientific Computation (PETSc) [8]. We refer to the new radial decomposition version of GTC as GTC-P below.

Overall, the implementation of the radial decomposition in GTC resulted in a dramatic increase in scalability for the grid work and decrease in the memory footprint of each MPI process. We are now capable of carrying out an ITER-size simulation of 130 million grid points and 13 billion particles using 32,768 cores on the BG/P system, with as little as 512 Mbytes per core. This would not have been possible with the old algorithm due to the replication of the local poloidal grid (2 million points).

4 GTC-P Performance

Figure 2 shows a weak scaling study of GTC-P on Franklin, Intrepid, and Hyperion. In contrast with previous scaling studies that were carried out with the production version of GTC and where the computational grid was kept fixed [6, 7], the new radial domain decomposition in GTC-P allows us to perform a true weak scaling study where both the grid resolution and the particle number are increased proportionally to the number of processor cores. In this study, the 128-core benchmark uses 0.52 million grid points and 52 million particles while the 131,072-core case uses 525 million grid points and 52 billion particles. This spans 3 orders of magnitude in computational problem size and a range of fusion toroidal devices from a small laboratory experiment of 0.17 m in minor radius to

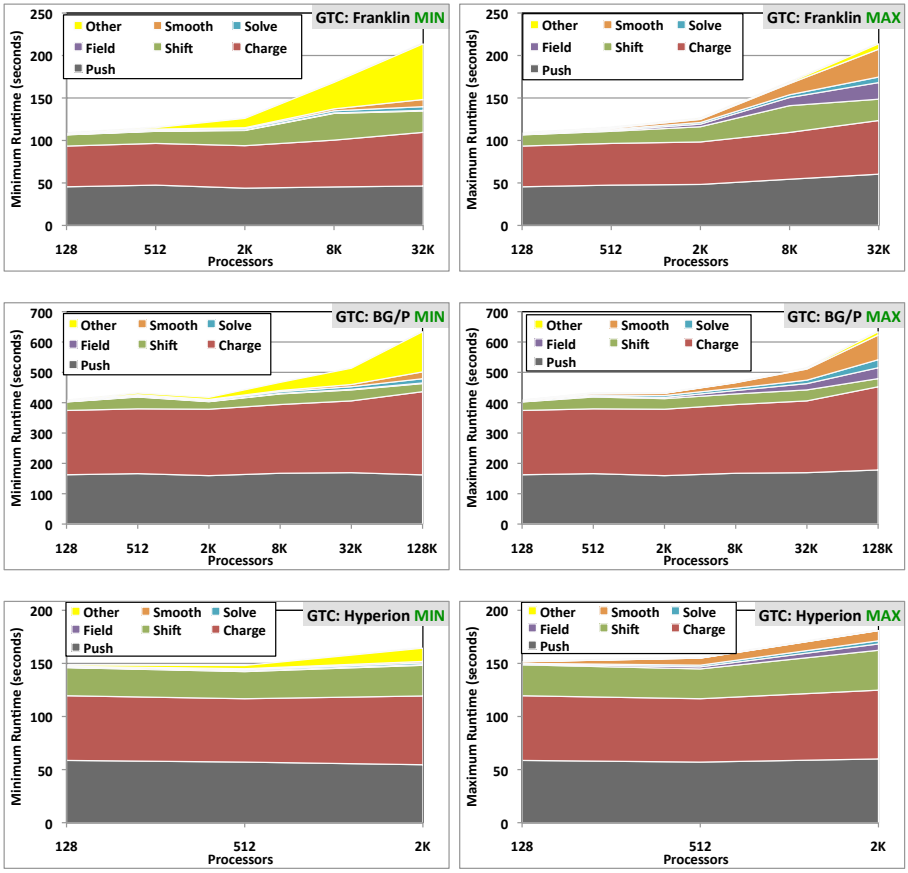

Fig. 2. GTC-P weak scaling, showing minimum and maximum times, on the (top) Franklin XT4, (middle) Intrepid BG/P, and (bottom) Hyperion Xeon systems. The total number of particles and grid points are increased proportionally to the number of cores, describing a fusion device the size of ITER at 32,768 cores.

an ITER-size device of 2.7 m, and to even twice that number for the 131,072-core test parameters. A doubling of the minor radius of the torus increases its volume by 8 if the aspect ratio is kept fixed. The Franklin Cray XT4 numbers stop at the ITER-size case on 32,768 cores due to the number of processors available on the system although the same number of cores can easily handle the largest case since the amount of memory per core is much larger than on BG/P. The concurrency on Hyperion stops at 2048, again due to the limited number of cores on this system. It is worth mentioning that we did not use the shared-memory OpenMP parallelism in this study although it is available in GTC-P.

The results of the weak scaling study are presented as area plots of the wall clock times for the main steps in the time-advanced loop as the number of cores increases from 128 to 32,768 in the case of the XT4, from 128 to 131,072 for the BG/P, and from 128 to 2048 for Hyperion. The main steps of the time loop are:

accumulating the particles' charge density on the grid ("charge" step, memory scatter), solving the Poisson equation on the grid ("solve" step), smoothing the potential ("smooth" step), evaluating the electric field on the grid ("field" step), advancing the particles by one time step ("push" phase including field gather), and finally, moving the particles between processes ("shift"). Notice that the XT4 is faster than the other two systems for the same number of cores, approximately 30% faster than Hyperion up to the maximum of 2048 cores available on that system. Compared to BG/P, Franklin is 4 times faster at low core count but that gap decreases to 2.4 times faster at 32,768 cores. This clearly indicates that GTC-P scales better on BG/P than Franklin, a conclusion that can be readily inferred visually from the area plots.

The scaling on BG/P is impressive and shows the good balance between the processor speed and the network speed. Both the "charge" and "push" steps have excellent scaling on all three systems as can be seen from the nearly constant width of their respective areas on the plots although the "charge" step starts to increase at large processor count. The "shift" step also has very good scaling but the "smooth" and "field" steps account for the largest degradation in the scaling at high processor counts. They also account for the largest differences between the minimum and maximum times spent by the MPI tasks in the main loop as can be seen by comparing the left (minimum times) and right (maximum times) plots for each system. These two steps hardly show up on the plots for the minimum times while they grow steadily on the plots for the maximum times. They make up for most of the unaccounted time on the minimum time plots, which shows up as "Other". This indicates a growing load imbalance as the number of processor-cores increases. We note that the "push", "charge", and "shift" steps involve almost exclusively particle-related work while "smooth" and "field" involve only grid-related work.

One might conclude that heavy communication is responsible for most of the load imbalance but we think otherwise due to the fact that grid work seems to be the most affected. We believe that the imbalance is due to a large disparity in the number of grid points handled by the different processes at high core count. It is virtually impossible to have the same number of particles and grid points on each core due to the toroidal geometry of the computational volume and the radially decomposed domains. Since we require a uniform density of grid points on the cross-sectional planes, this translates to a constant arc length (and also radial length) separating adjacent grid points, resulting in less points on the radial surfaces near the center of the circular plane compared to the ones near the outer boundary. Furthermore, the four-point average method used for the charge accumulation requires 6 to 8 radial ghost surfaces on each side of the radial zones to accommodate particles with large Larmor radii. For large device sizes, this leads to large differences in the total number of grid points that the processes near the outer boundary have to handle compared to the processes near the center. Since the particle work accounts for 80%–90% of the computational work, as shown by the sum of the "push" "charge" and "shift" steps in the area plots, it is more important to have the same number of particles in each radial

domain rather than the same number of grid points. The domain decomposition in its current implementation thus targets a constant average number of particles during the simulation rather than a constant number of grid points since both cannot be achieved simultaneously. This also implies that the amount of memory per MPI process slowly increases with the number of cores in our weak scaling study. This increase is negligible compared to the old GTC algorithm with the replicated local grids for which the 2048-core case does not fit in memory. It should be said, however, that this decomposition has allowed GTC to simulate dramatically larger fusion devices on BG/P and that the scaling still remains impressive.

The most communication intensive routine in GTC-P is the "shift" step, which moves the particles between the processes according to their new locations after the time advance step. By looking at the plots of wall clock times for the 3 studied architectures, we clearly see that BG/P has the smallest ratio of time spent in "shift" compared to the total loop time. This translates to the best compute to communication ratio, which is to be expected since BG/P has the slowest processor of the three evaluated platforms. Hyperion, on the other hand, delivered the highest ratio of time spent in "shift", indicating a network performance not as well balanced to its processor speed than the other two systems. In terms of raw communication performance, the time spent in "shift" on the XT4 is about half of that on the BG/P at low core count. At high processor count, the times are about the same. It is worth noting that on 131,072 cores on BG/P, process placement was used to optimize the communications while this was not yet attempted on Franklin at 32,768 cores.

5 Summary

Computational science is at the dawn of petascale computing capability, with the potential to achieve simulation scale and numerical fidelity at hitherto unattainable levels. However, harnessing such extreme computing power will require an unprecedented degree of parallelism both within the scientific applications and at all levels of the underlying architectural platforms. In this study, we examined extending the scalability properties of GTC, a key application for studying micro-turbulence in fusion devices. Performance results and analysis were presented on three leading HPC platforms: Franklin Cray XT4, Hyperion Intel Xeon cluster, and Intrepid IBM BG/P, representing some of the most common design tradeoffs in the high performance computing arena. To allow for GTC simulations of unprecedented size, we incorporated a new radial decomposition into the algorithm. This additional level of parallelization resulted in a dramatic increase in scalability for the grid work and decrease in the memory footprint of each MPI process. We are now capable of carrying out an ITER-size simulation of 130 million grid points and 13 billion particles using 32,768 cores on the BG/P system, with as little as 512 MB per core. This would not have been possible with the original algorithm due to the replication of the local poloidal grid. Future work will focus on reducing the impact of load imbalance at high concurrencies,

while continue exploring additional optimization strategies and performance on emerging high-end architectures.

Acknowledgments

Dr. Ethier is supported by the U. S. Department of Energy Office of Fusion Energy Sciences under contract number DE-ACO2-76CH03073. Dr. Oliker and Dr. Carter are supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231. This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research used resources of the Hyperion Project at Lawrence Livermore National Laboratory `http://www.hyperionproject.llnl.gov`.

References

1. C.K. Birdsall and A.B. Langdon. *Plasma Physics via Computer Simulation*, pages 437–441, Appendix C: Digital Filtering in One and Two Dimensions. Institute of Physics Publishing, 1991.
2. S. Ethier, W. M. Tang, R. Walkup, and L. Oliker. Large-scale gyrokinetic particle simulation of microturbulence in magnetically confined fusion plasmas. *IBM J. Res. and Dev.*, 52:105–115, 2008.
3. ITER: International thermonuclear experimental reactor. URL `http://www.iter.org/`.
4. W. W. Lee. Gyrokinetic particle simulation model. *J. Comp. Phys.*, 72:243–269, 1987.
5. Z. Lin, T. S. Hahm, W. W. Lee, et al. Turbulent transport reduction by zonal flows: Massively parallel simulations. *Science*, 281:1835–1837, 1998.
6. L. Oliker, A. Canning, J. Carter, et al. Scientific Application Performance on Candidate PetaScale Platforms. In *IPDPS:International Conference on Parallel and Distributed Computing Systems*, Long Beach, CA, 2007.
7. L. Oliker, A. Canning, J. Carter, and J. Shalf. Scientific computations on modern parallel vector systems. In *Proc. SC2004*, Pittsburgh, PA, 2004.
8. PETSc: Portable, extensible toolkit for scientific computation. URL `http://www.mcs.anl.gov/petsc/`.