Benchmarking Leading Computational Methods on the Earth Simulator

Project Leaders

Horst D. Simon — National Energy Research Scientific Center at Lawrence Berkeley National Laboratory

Leonid Oliker — National Energy Research Scientific Center at Lawrence Berkeley National Laboratory

Shigemune Kitawaki — The Earth Simulator Center, Japan Agency for Marine-Earth Science and Technology

Authors

Horst Simon*1, Leonid Oliker*1, Jonathan Carter*1, Stephane Ethier*2, Shigemune Kitawaki*3 and Yoshinori Tsuda*3

*1 National Energy Research Scientific Center at Lawrence Berkeley National Laboratory

*2 Princeton Plasma Physics Laboratory

*3 Earth Simulator Center, Independent Administrative Institution, Japan Agency for Marine-Earth Science and Technology

This work explores two applications from leading scientific domains in the areas of magnetic fusion (GTC), and Navier-Stokes turbulent flow (ELBM3D). We compare performance between the vector-based Earth Simulator, and other leading supercomputer architectures. Overall results show that the ES attains unprecedented aggregate performance across our evaluated applications, demonstrating the tremendous potential of modern parallel vector systems.

Keywords: Performance evaluation, vectorization, scientific computing

1. Introduction

Despite their dominance of high-end computing (HEC) through the 1980's, vector systems have been progressively replaced by microprocessor based systems due to the lower costs afforded by mass-market commercialization and the relentless pace of clock frequency improvements for microprocessor cores. However, while peak performance of superscalar systems has grown exponentially, the gradual slide in sustained performance delivered to scientific applications has become a growing concern among HEC users. This trend has been widely attributed to the use of superscalar-based commodity components whose architectural designs offer a balance between memory performance, network capability, and execution rate that is poorly matched to the requirements of large-scale numerical computations. Furthermore, now that power dissipation is limiting the growth rate in clock frequency, the low sustained performance of superscalar systems has risen to the forefront of concerns. The latest generation of custom-built parallel vector systems have the potential to address these performance challenges for numerical algorithms amenable to vectorization.

This work build on our previous effort [1, 2] and compares performance of the cacheless vector Earth Simulator (ES) versus various other supercomputer platforms. Performance results are presented from two key scientific computing domains: computational fluid dynamics and magnetic fusion.

2. GTC

GTC is a 3D particle-in-cell code used for studying turbulent transport in magnetic fusion plasmas [3]. The simulation geometry is that of a torus, which is the natural configuration of all tokamak fusion devices. As the charged particles forming the plasma move within the externally-imposed magnetic field, they collectively create their own self-consistent electrostatic (and electromagnetic) field that quickly becomes turbulent under driving temperature and density gradients. Waves and particles interact self-consistently with each other, exchanging energy that grows or damps their motion or amplitude. The particle-in-cell (PIC) method describes this complex phenomenon by solving the 5D gyro-averaged kinetic equation coupled to the Poisson equation.

GTC was originally optimized for superscalar SMP-based architectures by utilizing two levels of parallelism: a one-dimensional MPI-based domain decomposition in the toroidal direction, and a loop-level work splitting method implemented with OpenMP. However, the mixed-mode GTC implementation is poorly suited for vector platforms due to memory constraints and the fact that vectorization and thread-based loop-level parallelism compete directly with each other. As a result, previous vector experiments [1] were limited to 64-way parallelism-the optimal number of domains in the 1D toroidal decomposition. Note that the

number of domains (64) is not limited by the scaling of the algorithm but rather by the physical properties of the system, which features a quasi two-dimensional electrostatic potential when put on a coordinate system that follows the magnetic field lines. GTC uses such a coordinate system and increasing the number of grid points in the toroidal direction does not change the results of the simulation.

To increase GTC's concurrency in pure MPI mode, a third level of parallelism was recently introduced. Since the computational work directly involving the particles accounts for almost 85% of the overhead, the updated algorithm splits the particles between several processors within each domain of the 1D spatial decomposition. Each processor then works on a subgroup of particles that span the whole volume of a given domain. This allows us to divide the particle-related work between several processor and, if needed, to considerably increase the number of particles in the simulation. The update approach maintains a good load balance due to the uniformity of the particle distribution.

Prior to our October 2005 visit to the ESC, further vector optimizations were explored in GTC for the newly upgraded Cray X1E platform at the Oak Ridge National Laboratory. The ES version of the code did not perform as well on the X1 and X1E as on the ES due to the much slower scalar processor on those Cray computers and the large impact of having sections of the codes that do not multistream nor vectorize. Such sections run at least 32 times slower than fully multistreamed and vectorized sections on the X1/X1E. The main optimizations consisted of vectorizing a smaller loop in the code and changing the order of the dimensions for several GTC grid arrays. This latter change improved the performance of the code by 20% on the X1E but only by up to 3% on the ES. This can be attributed to the faster memory subsystem on the ES. Although the X1E processor (MSP) has a higher peak performance than the original X1 (and ES processor), 2 MSPs now share the same memory bandwidth as was available to a single MSP on the X1. In spite of this, the new vector and multistream optimizations now allow GTC to run as fast on an X1E MSP as on an Earth Simulator processor, although not as efficiently in terms of percentage of peak performance (10% vs 23%).

Another platform of interest for particle- in-cell codes is the recent Cray XT3 computer, which uses the AMD Opteron processor. Due to its architecture the Opteron has a higher memory access speed than other super-scalar systems, a key requirement to achieve good performance for the numerous random memory accesses performed in PIC calculations.

2.1 Experimental Results

For this performance study, we keep the grid size constant but increase the total number of particles so as to maintain the same number of particles per processor, where each processor follows about 3.2 million particles. Table 2 shows the performance for the Cray XT3 and ES. The first difference from the previous GTC vector study [2], is the achievement of yet another increase in concurrency. The new particle decomposition algorithm allowed GTC to efficiently utilize 4,096 processors (compared with only 64 using the original approach), although this is not the limit of its scalability as Blue Gene/L benchmarks have shown good scaling past the 16,000 processor mark. With this new algorithm in place, GTC fulfilled the very strict scaling requirements of the ES and achieved an unprecedented 7.2 Tflop/s on 4,096 processors. This performance has not been achieved yet on any other platform. Additionally, the Earth Simulator sustains a significantly higher percentage of peak (23%) compared with other platforms. While GTC achieves only 10 to12% on the Cray X1E, the Opteron-based Cray XT3 gets up to 15% of peak, which is very good for a super-scalar machine. Although the XT3 is only 2.5 times slower than the Earth Simulator, we would still need a little more than 10,000 processors to achieve the top performance of GTC on the ES.

Table 1 GTC results on Cray XT3 and ES.

P	Part/ Cell	Cray XT3		ES		
		Gflp/ Proc	% Pk	Gflp/ Proc	% Pk	Spd up
64	100	0.66	13.7	1.85	23.1	2.8
128	200	0.69	14.3	1.78	22.3	2.6
256	400	0.71	14.7	1.77	22.1	2.5
512	800	0.72	15.0	1.78	22.2	2.5
1024	1600	0.73	15.2	1.77	22.0	2.4
2048	3200	0.74	15.3	1.75	21.9	2.4
4096	6400	0.73	15.2	1.76	22.0	2.4

3. ELBM3D

Lattice-Boltzmann methods (LBM) have proved a good alternative to conventional numerical approaches for simulating fluid flows and modeling physics in fluids [4]. The basic idea is to develop a simplified kinetic model that incorporates the essential physics, and reproduces correct macroscopic averaged properties. These algorithms have been used extensively over the past ten years for simulating Navier-Stokes flows. As can be expected from explicit algorithms, LBM are prone to numerical nonlinear instabilities as one pushes to higher Reynolds numbers. These numerical instabilities arise because there are no constraints imposed to enforce the distribution functions to remain non-negative. Such entropic LBM algorithms, which do preserve the non-negativity of the distribution functions-even in the limit of arbitrary small transport coefficients-have recently been developed for Navier-Stokes turbulence [5]. Our LBM application is representative of this active research area: the ELBM3D code uses the entropic LB algorithm to simulate

the behaviour of Navier-Stokes turbulence [6]. While LBM methods lend themselves to easy implementation of difficult boundary geometries, e.g., by the use of bounce-back to simulate no slip wall conditions, here we report on 3D simulations under periodic boundary conditions, with the spatial grid and phase space velocity lattice overlaying each other. Each lattice point is associated with a set of mesoscopic variables, whose values are stored in vectors proportional to the number of streaming directions. The lattice is partitioned onto a 3-dimensional Cartesian processor grid, and MPI is used for communication. As in most simulations of this nature, ghost cells are used to hold copies of the planes of data from neighboring processors.

In simple terms a LB simulation proceeds by a sequence of collision and stream steps. A collision step involves data local only to that spatial point, allowing concurrent, dependence-free point updates; the mesoscopic variables at each point are updated through a complex algebraic expression originally derived from appropriate conservation laws. A stream step evolves the mesoscopic variables along the streaming lattice. However, a key optimization is often implemented, saving on the work required by the stream step. The two phases of the simulation can be combined, so that either the newly calculated particle distribution function can be scattered to the correct neighbor as soon as it is calculated, or equivalently, data can be gathered from adjacent cells to calculate the updated value for the current cell.

For ELBM3D, a non-linear equation must be solved for each grid-point and at each time-step so that the collision process satisfies certain constraints. The equation is solved via Newton-Raphson iteration (5 iterations are usually enough to converge to within 10^{-8}), and as this equation involves taking the logarithm of each component of the distribution function at each iteration, the whole algorithm becomes heavily constrained by the performance of the log function.

3.1 Experimental Results

Table [3] presents ELBM3D performance on the Cray XT3 and ES. Observe that the vector architecture clearly outperform the scalar systems by a significant factor. The ES sustains the highest fraction of peak across all architectures tested to date—39% even at the highest 2048-processors concurrencies. Further experiments on the ES on 4096 processors attained an aggregate performance of over 13 Tflop/s. However, the Cray XT3 also does quite well with this application achieving a steady 22% of peak, or just over 1 GFlop/s. The relatively high performance is due to the log functions being computed via a call to the AMD ACML library, which enables pipelining of all log computations. In addition, the code is tuned slightly for cache reuse, obtaining a speedup of about 15% over the vector version.

Preliminary experiments on the Cray X1E show performance of about 4.5 GFlop/s per processor. This is somewhat faster than the ES, but a lower percentage of peak, 25%, than that delivered by the ES.

Table 2 ELBM3D results on Cray XT3 and ES.

P	Grid Size	XT3		ES		
		Gflp/ Proc	% Pk	Gflp/ Proc	% Pk	Spd up
64	512	1.11	23	3.36	42	3.0
256	512	1.07	22	3.35	42	3.1
512	1024	1.07	22	3.16	39	3.0
1024	1024	1.06	22	3.12	39	2.9
2048	2048	1.06	22	3.10	39	2.9

4. Summary

This study examined two scientific applications on the vector-based ES and superscalar Cray XT3 platforms. We presented a refinement of the decomposition parallelization for the GTC magnetic fusion simulation. This new approach allowed scalability to 4096 processors on the ES (compared to only 64 using the previous code version), opening the door to a new set of high-phase space-resolution simulations, that to date have not been possible. Next we presented ELBM3D, an entropic lattice Boltzmann application used to study the onset evolution of fluid flow turbulence. The ES showed very high ELBM3D performance, achieving over 39% of peak at the highest concurrencies.

Overall results show that the ES achieved the highest aggregate performance on any architecture tested to date across our pair of applications, demonstrating the tremendous potential of modern parallel vector systems.

Acknowledgements

The authors would like to gratefully thank: the staff of the Earth Simulator Center, especially Dr. T. Sato, for their assistance during our visit.

This research used the resources of the National Center for Computational Sciences at Oak Ridge National Laboratory supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

All authors from LBNL were supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231. Dr. Ethier was supported by the Department of Energy under contract number DE-AC020-76-CH03073.

References

1) L. Oliker, A. Canning, J. Carter, J. Shalf. And S. Ethier, "Scientific Computations on Modern Parallel Vector Systems", Proc. SC2004: High performance computing,

networking, and storage conference, 2004.
2) L. Oliker, J. Carter, M. Wehner, A. Canning, S. Ethier, A. Mirin, G. Bala, D. Parks, S. Kitawaki and Y. Tsuda, "Leading Computational Methods on Scalar and Vector HEC Platforms", Proc. SC2005: High performance computing, networking, and storage conference, 2005.
3) Z. Lin, S. Ethier, T. S. Hamh, and W. M. Tang, "Size Scaling of turbulent transport in magnetically confined plasmas", Phys. Rev. Lett., vol.88, pp.195004 (2002).
4) S. Succi, "The lattice Boltzmann equation for fluids and beyond". Oxford Science Publ. (2001)
5) S. Ansumali and I.V. Karlin, "Stabilization of the lattice Boltzmann method by the H theorem: A numerical test". Phys. Rev. E., vol.62, pp.7999 (2000).
6) G. Vahala, J. Yepez, L. Vahala, M. Soe, J. Carter, "3D entropic lattice Boltzmann simulations of 3D Navier-Stokes turbulence". Proc. of 47th Annual Meeting of the APS Division of Plasma Physics. (2005).

地球シミュレータを使用した先進的な計算手法に関するベンチマーク

プロジェクトリーダー
Horst Simon　ローレンス バクレー国立研究所　国立エネルギー研究科学コンピューティングセンター（米国）
Leonid Oliker　ローレンス バクレー国立研究所　国立エネルギー研究科学コンピューティングセンター（米国）
北脇　重宗　海洋研究開発機構　地球シミュレータセンター

著者
Horst Simon*1，Leonid Oliker*1，Jonathan Carter*1，Stephane Ethier*2，北脇　重宗*3，津田　義典*3
*1　ローレンス バクレー国立研究所　国立エネルギー研究科学コンピューティングセンター（米国）
*2　プリンストンプラズマ物理研究所（米国）
*3　海洋研究開発機構　地球シミュレータセンター

このプロジェクトでは先進科学分野から核融合（Gyrokinetic Toroidal Code: GTC）、格子ボルツマン法による乱流コード（Entropic lattice Boltzmann simulation of 3D Navier-Stokes turbulence: ELBM3D）の2種のコードを研究した。ベクトルプロセッサベースの地球シミュレータと他のスーパースカラプロセッサベースの先進的なスーパーコンピュータのアーキテクチャの性能を比較した。総合的な結果として地球シミュレータは評価した全てのアプリケーションで今までにない統合性能を達成し、最新の並列ベクトルプロセッサシステムの素晴らしい潜在能力を示した。

キーワード：性能評価，ベクトル化，科学技術計算