HHS Public Access

Author manuscript

Comput Phys Commun. Author manuscript; available in PMC 2017 July 01.

Published in final edited form as:

Comput Phys Commun. 2016 July ; 204: 132–140. doi:10.1016/j.cpc.2016.03.019.

Scalability Test of Multiscale Fluid-Platelet Model for Three Top Supercomputers

Peng Zhang[1,*], **Na Zhang**[2], **Chao Gao**[1], **Li Zhang**[2], **Yuxiang Gao**[3], **Yuefan Deng**[2], and **Danny Bluestein**[1,*]

[1] Biomedical Engineering Department, Stony Brook University, Stony Brook, NY 11794-3600, United States

[2] Applied Mathematics Department, Stony Brook University, Stony Brook, NY 11794-8151, United States

[3] Cluster Solution Department, Cray Inc., San Jose, CA 95112, United States

Abstract

We have tested the scalability of three supercomputers: the Tianhe-2, Stampede and CS-Storm with multiscale fluid-platelet simulations, in which a highly-resolved and efficient numerical model for nanoscale biophysics of platelets in microscale viscous biofluids is considered. Three experiments involving varying problem sizes were performed: Exp-S: 680,718-particle single-platelet; Exp-M: 2,722,872-particle 4-platelet; and Exp-L: 10,891,488-particle 16-platelet. Our implementations of multiple time-stepping (MTS) algorithm improved the performance of single time-stepping (STS) in all experiments. Using MTS, our model achieved the following simulation rates: 12.5, 25.0, 35.5 μs/day for Exp-S and 9.09, 6.25, 14.29 μs/day for Exp-M on Tianhe-2, CS-Storm 16-K80 and Stampede K20. The best rate for Exp-L was 6.25 μs/day for Stampede. Utilizing current advanced HPC resources, the simulation rates achieved by our algorithms bring within reach performing complex multiscale simulations for solving vexing problems at the interface of biology and engineering, such as thrombosis in blood flow which combines millisecond-scale hematology with microscale blood flow at resolutions of micro-to-nanoscale cellular components of platelets. This study of testing the performance characteristics of supercomputers with advanced computational algorithms that offer optimal trade-off to achieve enhanced computational performance serves to demonstrate that such simulations are feasible with currently available HPC resources.

Keywords

Performance analysis; heterogeneous multicore and multi-GPU architecture; multiscale simulation; computational bioengineering

[*] Corresponding authors at Biofluids Research Group, Biomedical Engineer Department, Stony Brook University, NY 11794, United States. peng.zhang@stonybrook.edu (P. Zhang) danny.bluestein@stonybrook.edu (D. Bluestein).

1. INTRODUCTION

Numerical simulations revolutionize many fields of engineering and science [1, 2] by extending traditional theoretical studies and laboratory experiments to explore multiscale phenomena that are hardly observable or measurable in laboratory settings. Such simulations also require overcoming computational challenges including resolving the diverse spatial-temporal scales [3-5] for understanding multi-component biological and behavioral systems [3-5]. For instance, cardiovascular diseases account for nearly 30% of all deaths globally and 35% of all U.S. deaths annually. Whether due to acute thrombosis associated with myocardial infarction, or progressive intermittent atherothrombotic events, significant ventricular dysfunction may result, leading to hear failure. Presently over 5.5 million patients suffer from heart failure in US and their number is expected to grow by 50%. Of these patients, a significant proportion will become candidates for mechanical circulatory support and prosthetic cardiovascular devices, also burdened with thromboembolic risk and complications. The coagulation cascade of blood may be initiated by flow-induced platelet activation, which prompts clot formation in prosthetic cardiovascular devices and in arterial disease processes. Upon activation, platelets undergo complex biophysical and morphological changes. Activated platelets polymerize fibrinogen into a fibrin network that enmeshes red blood cells. Continuum methods fail to capture the small-scale molecular mechanisms such as filopodia formation upon platelet activation. Utilizing molecular dynamics (MD) that can capture and model the molecular mechanisms is computationally prohibitive for large-scale hematologic problems [5, 6]. Innovative numerical approaches are essential for elucidating such vexing problems and enhance the linkage to applications for solving complex problems at the interface of biology and engineering. We propose a multiscale numerical approach for modeling the multiscale fluid-platelet phenomena. The proposed approach utilizes a multiscale model where dissipative particle dynamics is coupled with molecular dynamics to describe flow induced mechanotransduction processes and biochemical events spanning the vast range of spatial and temporal scales characterizing blood flow-induced clotting and thrombosis [6-9].

However, the challenge of multiscale modeling cannot be simply resolved by leveraging on the raw computing speeds of hardware systems. The multiscale nature of such problems mandates developing efficient computational algorithms Hence, we develop a multiple time-stepping (MTS) algorithm in which time integration is carried out at multiple temporal scales [8] and we implement this algorithm on multi-CPU and multi-GPU supercomputers. Our MTS algorithm helps to bridge the temporal gap between macroscale flow regime and cellular-scale events. As a result, it avoids a massive number of redundant computations that are inherent in single time-stepping (STS) algorithms, thus significantly boosting the computing efficiency.

In terms of the scalability tests, we build a problem-specific benchmark for three benchmark systems. The smallest system contains 680,718 particles, forming a single platelet that flows in a viscous flow. In this system, the platelet and flow models contain 140,015 (21%) and 540,703 (79%) particles, respectively. This system is typically used for parameterization and observation of morphological changes for a single human platelet. The largest system contains 10,891,488 particles and it includes 16 platelets. This system helps to study the

initial stage of clot processes involving multiple platelets. The middle system contains 2,722,872 particles forming 4 platelets. This system is suitable for validating the inter-platelet and platelet-vessel interactions. From a practical point of view, our results provide the references for evaluating the computational capabilities for given computers. These benchmarks are carried out on two main categories of supercomputer architectures: homogeneous multi-processor and heterogeneous multi-accelerator architectures. This selection allows us to establish the relationship between diverse supercomputer architectures and diverse multiscale simulations.

In terms of engineering efforts, we implement our novel multiscale model and 4-level multiple time-stepping (MTS) algorithm on three supercomputers of different architectures and we further implement this MTS algorithm on GPUs. These implementations enabled us to: provide for the first time computational insights for solving multiscale simulations; and demonstrate the computational capabilities for simulating *millisecond*-scale multiscale phenomena of multi-component biological systems. The substantial speedup resulting from aggregated efforts of physical models, mathematical algorithms and sophisticated implementation on supercomputers of different architectures offer the community of computational science and biomedical engineering new insights and enhanced ability for incorporating and advancing multiscale modeling in their research. Our work is the first examination of the performance of multiscale platelet-mediated simulations on supercomputers. It clearly demonstrates the feasibility and affordability for conducting millisecond-order micro-to-nanoscale simulations of multiple platelets in viscous blood flows on top supercomputers- , an essential step for numerical studies of the thrombogenicity in clinical problems. The integration of multiple model and MTS algorithm is a cost-effective approach for various classes of specific problems in biomedical engineering and biomedical sciences.

The computational research presented in this paper is organized as follows: a multiscale model and multiple time-stepping (MTS) algorithm is presented in Section 2. The key features of three supercomputers are described in Section 3. In Section 4, the experiment setup is presented, followed by results. In Section 4.3, we characterize and analyze the performance of individual supercomputers, followed by conclusions in Section 5.

2. MULTISCALE SIMULATIONS

The multiscale model of a platelet is introduced by examining the anatomy of the platelet. The numerical algorithm is designed to model the platelet with sufficient spatio-temporal resolution using minimal computing resources that would typically require far larger ones. The key element of the algorithm is the multiple time-stepping integration of the equation of motion. In this section, we first describe the multiscale model where multiple spatial-temporal scales are formulated and then employ the multiple time-stepping (MTS) algorithm as an efficient numeric solver for this model.

2.1 Multiscale Model

By extending our previous efforts for modeling platelets under viscous shear flow conditions at multiple spatio-temporal scales [6-9], we develop a multiscale benchmark model for

assessing the performances of three supercomputers for understanding multiscale phenomena of complex biological systems. In this model, we cover two spatio-temporal scales: (i) the microscale flow regime using dissipative particle dynamics (DPD) to describe the bulk transfer of viscous blood flow [10]; and, (ii) the nanoscale platelet model using coarse-grained molecular dynamics (CGMD) to describe the cellular scale structural details of a platelet such as membranous morphology, cytoplasmic biorheology, cytoskeletal filaments and the flow-mediated cellular mechanotransduction of hemodynamic stresses across the platelet surface and through the cytoskeleton [6]. Both DPD and CGMD are coarse-grained particle-based methods Specifically DPD/CGMD is the coarse-grained stochastic molecular dynamics, respectively. In the formulation, DPD employs a special stochastic character that depends on the momentum to increase the scales of viscous flows it can formulate. CGMD employs a conservative force field to formulate the cellular-scale intra-platelet interactions. The interface of these two formulations is described by a hybrid force field; the local thermodynamics and exchange of momentum is governed by the DPD stochastic term while the incompressibility of the platelets, under the stress of the surrounding flow is governed by the CGMD conservative term [6]. Figure 1 is a schematic description of the multiple spatio-temporal scales in which different force fields capture the characteristics of the different scales. A wall-driven Couette benchmark flow is introduced to induce the characteristic flipping of the deformable platelets in such shear flows.

This model may be utilized to study similar phenomena of flowing blood cells such as erythrocytes and leukocytes that require understanding, a priori, the scalability and time-to-solution for such problems. **Figure 2** illustrates the hierarchy of multiple spatial-temporal scales correlated with biological phenomena at multiple resolutions.

2.2 Multiple Time Stepping Algorithm

The spatial interface between DPD and CGMD is the first step for performing such multiscale simulation and the corresponding time integration, multiple time stepping (MTS) algorithm, solving the DPD and CGMD at microseconds and nanoseconds temporal scales [8] is the following step. Far more sophisticated than single time stepping algorithms [11-13], MTS algorithms handle time integration of step sizes differing by 3~4 orders of magnitude, as those between DPD and CGMD time scales. Following [8], we use a four-step scheme model in which the fluid advances at the largest step size, the fluid-platelet interface at a middle step size and the nonbonded and bonded force fields within platelets at the two smallest step sizes. The relationship between accuracy and computational load, i.e., the MTS parameters, was studied in [8].

We customized the LAMMPS software suite to implement our new model and algorithm. We implemented our multiscale model and MTS algorithm on the Tianhe-2 and Stampede supercomputers, and additionally implemented the GPU codes on the Stampede and CS-Storm supercomputers. Specifically, the DPD-CGMD interface function is implemented on CPUs. The force evaluation for DPD and CGMD formulas is implemented on CPUs and GPUs. In the floating-point previsions, the force evaluation on GPU provides three floating-point precisions: single, double and mixed; and the force evaluation on CPU uses double

precision. Inter-process communication (IPC) is unaltered and the integrator uses the NVE integrator.

3. THREE TOP SUPERCOMPUTERS

As increasing computer performance simply by increasing CPU clock speeds reaches its limitations, supercomputer developers must investigate new architectures that can achieve enhanced performance. Interconnection networks are developed to support much larger number of processors at the expense of increased engineering complexities and programming difficulties [14-17]. Accelerators are used for boosting floating-point performance, though it further escalates the difficulty of programming that can fully utilize the novel accelerator or co-processor design [18-21]. In this work, we implement our models on multi-CPU and multi-GPU platforms and test our multiscale codes on three top supercomputers: Tianhe-2 (No. 1 in TOP500 from Jun 2013 through Nov 2014), Stampede (No. 7 in TOP500 from Nov 2012 through Nov 2014) and Cray CS-Storm, a high-density multi-accelerator system. The CS-Storm system is used in the No. 10 computer of the November 2014 release of TOP500 with NVIDIA K40. In this work, we test the CS-Storm system with K40 and K80, the latest NVIDIA GPU generation.

3.1 Tianhe-2 Supercomputer

Tianhe-2 supercomputer [22] (introduced in 2013) is a 54.9-PFlops system that is powered by 3.12 million cores from 16,000 compute nodes. Each node has two Intel Ivy Bridge processors (Xeon E5-2692 v2 12C 2.2GHz) with 64GB of memory, providing 422.4 GFlops per dual socket node. Each node also has three Intel Phi coprocessors, providing 3 TFlops per node. The processing nodes are connected by the TH Express-2 customized interconnect networks [23].

3.2 Stampede Supercomputer

Stampede supercomputer [24] (introduced in 2012) is an 8.5-PFlops system that is powered by 0.46 million cores from 6,400 compute nodes. Each node has two Intel Sandy Bridge processors (Xeon E5-2680 8C 2.7GHz) with 32GB of memory, providing 345.6 GFlops per dual socket node. Each node also has one Intel Phi coprocessor, providing 1 TFlops per node. These nodes use the FDR Infiniband network in a 2-level fat-tree topology. The Tianhe-2 single-node peak-performance is 2.54 times that of Stampede.

3.3 Cray CS-Storm Supercomputer

CS-Storm is a high-density multi-accelerator system and it can integrate up to eight GPU cards, e.g., incorporate 8 K40m or 16 K80 where K80 is a dual GPU accelerator. A single CS-Storm 8-K40m/16-K80 server provides the double-precision (single-precision) peak performance of 11.4 / 15 (34.3 / 44.8) TFlops respectively. The memory is 12 GB per K40/K80 GPU. The CS-Storm node is much more powerful than the node in Tianhe-2 and Stampede. We tested two configurations of the CS-Storm node: one with 8 NVIDIA K40m GPUs and 2 Intel Xeon E5-2670 v2 2.5GHz CPUs (264GB memory) and the other with 16 K80 GPUs and 2 Intel Xeon E5-2680 v3 2.5 GHz CPUs (132GB memory).

4. PERFORMANCE EXAMINATIONS

4.1 Experiment Setups

Three problem sizes of our multiscale simulations were selected to benchmark the three supercomputers as shown in Table 2. In this setup, we extend the system size only by extending the simulation domain in the x- and z-dimension. The shear stress is generated by moving the walls in y-dimension and the stress as well as the platelet density remains unchanged for all experiments.

Algorithmically, the MTS and STS algorithms are tested. The single step size in STS is 1×10^{-6}. The step sizes for the flow regime and platelet-flow interface in MTS are 5×10^{-4} and 5×10^{-5}, and the smallest step size for intra-platelet force fields is 5×10^{-6} [8]. The error propagation in terms of energy conservation and other measures are prevented in MTS [8]. The scaling of dimensionless units and physical units in simulations is one dimensionless time unit representing 2.083 microseconds in SI units. The wall-driven shear stress is 8.58 Pa and shear rate is 8000 s^{-1}. The mean velocity of flow is 3.2 cm/s and the viscosity of flow is 1.072 mPa·s which is close to the normal viscosity of blood plasma [25]. This model can simulate the flow conditions through the small gap clearances in crevices of cardiovascular devices, in which platelet activation is a major clinical problem [6, 26, 27].

In every step, the particle data are exchanged between processors and accelerators. Neighbor list building and force evaluation are accelerated. The software provides three precision options: *single*, *double* and *mixed*. It allows the GPUs to process data in *single* or *double* precision or in *mixed*-mode precision, where the pairwise forces are computed in single precision but they are accumulated into double-precision force vectors. For the sake of speed, we experiment the mixed precision for all tests. Though LAMMPS starts to support some simulations on Intel Xeon Phi, it is still unavailable to support all of necessary functions on coprocessors for our multiscale model.

For all tests, we performed a lengthy process for the wall-driven Couette flow to become equilibrated. A no-slip boundary condition characteristic of viscous flows is applied to the y-dimension and a periodic boundary condition is applied to the x-/z-dimension. The timing output from the LAMMPS code is used as the final timing results. From these timing results, we compute the simulation speed in units of: (1) *days/μs*, i.e., the number of wallclock *days* it takes to complete 1-*μs* multiscale phenomena, for showing the days-per-microsecond performance rate; and (2) *μs/day*, i.e., the number of microseconds one physical day's simulation can achieve, for demonstrating the microseconds-per-day rate. The timing for simulation speed uses the wallclock time of a simulation, i.e., the *loop time* from the code timing output. Then we compute the performance improvement percentage of MTS over STS using the formula: $P(t_{MTS}, t_{STS}) = (t_{STS} - t_{MTS}) / t_{STS}$, in which t_{MTS} and t_{STS} are the timing results in *days/μs* for MTS and STS algorithms. For detailed analysis, we present the communication-to-computation ratio. The communication uses the *comm* timing and the computation is the sum of the *pair* and *bond* timing. In all tests, we notice that the *output* timing is almost zero since most outputs are disabled and thus the I/O operation with file systems is negligible. The *neigh* timing is also relatively small, less than 5%. This is then used to examine the performance results, as described below.

4.2 Performance Results

Performance results are presented in Figures 3-14. Every three successive figures present the performance metrics of one supercomputer, namely, the simulation speed in *days/μs*, the performance improvement for MTS over STS and the communication-to-computation ratio. The supercomputers are presented in the following order: Tianhe-2, Stampede, CS-Storm 8-K40m and 16-K80. For Tianhe-2 and Stampede, the horizontal axis is the number of processor cores in Figure 3 through Figure 8. For CS-Storm systems, the horizontal axis is the number of accelerators in Figure 9 through Figure 14. These results for speeds vs. number of cores/GPUs show the strong scaling of the multiscale simulations, in which the problem size is fixed while the number of processing elements (core/GPU) is increased. The weak scaling is not applicable since the problem size is dictated by the biological nature (i.e., number of platelet and platelet density) so it is hardly tailored to offer the fixed workload per processing element.

Constrained by limited memory of computer configuration, the multiscale simulations failed to run on a small group of computing nodes. This is common for large scale complex simulations. Thus, we begin our tests with 16 cores for Exp-S/Exp-M and 64 cores for Exp-L on Tianhe-2 and Stampede. Since the improvement of simulation speeds diminish with the rapid increase in the number of cores, we limited the tests to below 512 cores.

In the CS-Storm, we use 16 cores and vary the number of GPUs. Similarly, constrained by the limited memory of accelerators, the Exp-M failed to run on single GPU. The minimum numbers for K40m/K80 for the Exp-L with STS was eight. However, the minimum requirements for the Exp-L with MTS were 8 K40m and 16 K80 GPUs correspondingly, since the MTS needs more memory to store multi-level force vectors than the STS.

With the results from these runs obtained, the simulation speeds and the communication-to-computation ratios for Exp-L re compared between Tianhe-2 and Stampede in Figure 15 and Figure 16, respectively. The best performance in *days/μs* and *μs/day* are presented in Table 3, Table 4 and Table 5 for the Exp-M and Exp-L. Moreover, we extend the tests for Exp-S and Exp-M on Stampede K20-enabled nodes and present the results in Table 3 and Table 4, respectively. Exp-L failed to run on K20-enabled nodes due to memory deficiency.

4.3 Analysis and Summary

Based on the performance studies of our multiscale simulations on three top supercomputers, we arrive at the following conclusions: (1) MTS is an efficient algorithm that achieves significantly enhanced performance required by large multiscale simulations, and that developing problem-specific MTS algorithms may enable solving of various multiple spatio-temporal problems.

The results in Figs. 4, 7, 10 and 13 show that MTS is consistently superior over STS in all tests on all supercomputers in terms of the simulation speeds. The improvement for MTS over STS is larger on Tianhe-2/Stampede than that on CS-Storm. For example, for the Exp-L, the improvement that was achieved was 84% on Tianhe-2/Stampede and 66% and 45% on CS-Storm 16-K80 / 8-K40m, respectively.

MTS excels at reducing the computation costs while the communication-to-computation ratio remains unaltered across diversified supercomputers. In general, MTS has higher ratios on Tianhe-2/Stampede (Figs. 5 and 8) and lower ratios on CS-Storm 8-K40m/16-K80 (Figs. 11 and 14), in comparison with STS. These phenomena could be the result of widely different communication schemes: in the code, the inter-process communication employs the intra-node wires in CS-Storm and the inter-node cables in Tianhe-2 and Stampede.

Comparing speeds in Fig. 15, one can see that for Exp-L, Tianhe-2 is faster than Stampede for up to 128 cores, where then the Stampede is faster for 256 and 512 cores than the Tianhe-2. This could be the result of the node configuration. Tianhe-2 and Stampede have 24 and 16 cores per node, respectively. To provide the run with 128 cores, Stampede and Tianhe-2 needs 8 and 5 nodes respectively and thus Tianhe-2 would need less inter-node data transfers than Stampede. In addition, Tianhe-2 has more host memory (64GB per node) than Stampede (32GB per node), further boosting the node-level performance. Thus, when using a small group of cores, Tianhe-2 is better than Stampede. With the increase of cores, communication becomes more stressful than computation (Fig. 16) then the performance gap between Stampede and Tianhe-2 appears to diminish: 0.16~0.17 days/μs (MTS) and 0.98~1.09 days/μs (STS) for Stampede/Tianhe-2 using 512 cores.

(2) Multiscale simulations can provide nanoscale details of biomedical problems with affordable computing resources. The results demonstrate that the simulation rate of multiple microseconds of physical time per day for systems with multiple platelets and 10-million particles is achievable on these supercomputers.

Our multiscale approach offers nanoscale details for intracellular details while modeling the bulk transport of blood flows at the microscale [6]. The multiscale nature helps improve the scale of stepping sizes from femtosecond to nanosecond [8] for the deforming platelets. In addition, the MTS allows simulating viscous flow particles at the scale of 100-nanosecond stepping sizes. This algorithmic approach vastly improved the simulation speeds. For example, the CS-Storm 16-K80 server simulated 25 microseconds for a single-platelet system within a single day (Table 3), and 512-cores in Tianhe-2 and Stampede simulated 5.88~6.25 microseconds for a 16-platelet system (Table 5) within one day. Accordingly, the 1-ms nano/micro-composite simulation for a 10-million-particle system could be completed within 160 wallclock days.

The accelerator further boosts the performance of large scale computer systems. For example, the K20-enabled nodes improved the performance of CPU-only nodes by a factor of 2~3 for the Exp-M on Stampede (Table 4). In other words, the time it takes for 1-ms simulation of a 2.7-million-particle system is reduced from 140 days (32 CPU nodes) to 70 days (16 K20-enabled nodes) on Stampede.

The results demonstrated that the millisecond-scale nano/micro-composite simulations for million-particle systems are computationally feasible and affordable on supercomputers. As a comparison, Anton 2, a special-purpose molecular dynamics supercomputer achieved a simulation of 2.2-million-atom ribosome system at a rate of 3.6 μs/day [28]. Our efficient algorithms achieved a simulation of multiscale 10-million-particle 16-platelet system at a

rate of 6.25 and 5.88 μs/day on Stampede and Tianhe-2 (Table 5), and 2.7-million-particle 4-platelet system at a rate of 14.29 and 6.25 μs/day on Stampede/K20 and Tianhe-2 (Table 4).

The computationally affordable multiscale model presented here raise the bar by facilitating simulations of complex clinical problems such as thrombogenic and thrombolytic mechanisms involving multiple flowing blood elements in small arteries, which are rarely modeled utilizing atomic scale molecular dynamics multiscale simulations due to the prohibitive computational costs.

(3) upercomputers play a key role in development of multiscale models for large scale biological multi-component systems.

The heterogeneous multi-accelerator architecture demonstrated the powerful acceleration potential for boosting the multiscale simulation speed, with the caveat that those are closely dependent on the complexities involved in algorithm development and coding. As the supercomputer architectures are becoming more diversified, there is a need to match of the architectural supply and application demand for achieving an optimal performance. The match can be optimized by exploring the characteristics of the supercomputers and the specifics of the models. For example, the simulation for parameterizing the multiscale models could be handled by a high-density multi-accelerator computer (Table 3). The simulation for validating the inter-platelet interactions could be handled by using a small group of computer nodes (Table 4). The simulation for studying large-scale multi-platelet and multi-component interactions in arteriole need use the large scale supercomputers with high performance interconnects (Table 5). Though novel supercomputers are emerging as dominant tools for computer simulations, it has to be judicially matched to the state-of-the-art supercomputer architecture and state-of-the-practical application demand for efficient use of invaluable computing resources [29].

Based on these observations, we realize that the triple alliance of the multiscale model, efficient algorithm and powerful supercomputer should be consolidated for facilitating the anticipated revolutionizing of engineering and science through simulations [1]. In summary, application (multiscale model), algorithm (efficient algorithm), and architecture (supercomputer) must be developed collectively for securing optimal performance and driving the forefront of HPC.

5. Conclusions

We devised a multiscale benchmark model to assess the performance of three supercomputers by testing the performance of multiscale fluid-platelet simulations. This study demonstrates that (1) the MTS algorithm is superior to traditional STS algorithm for speeding up computation without losing accuracy, on all three supercomputers. The speedups achieved with 10 million particles system simulations were 84% (Tianhe-2/Stampede), 66% and 45% for CS-Storm 16-K80 and 8-K40m, respectively. (2) The high-density multi-accelerator computer, the CS-Storm, is capable of supporting a single-platelet multiscale simulation, taking 40 days to complete a 1-*ms* multiscale simulation of a single-platelet system. (3) Tianhe-2 and Stampede are capable of supporting multiscale multi-

platelet simulations, taking 160 days to complete 1-ms simulation for 16-platelet system (10.9 million particles). To compare, a 16-core Tianhe-2 or Stampede will require 42 years to complete a 1-ms multiscale simulation of the 16-platelet system. In particular, our multiple spatio-temporal model has brought a large multiscale simulation such as the millisecond-scale simulations of platelet-mediated processes presented here within computational reach to a practical arena. This study of testing the performance characteristics of supercomputers with advanced computational algorithms that offer optimal trade-off to achieve enhanced computational performance serves to demonstrate that such simulations are feasible with currently available HPC resources.

These performance examinations support the assertion that the triple alliance of models, algorithms and powerful computers can be formed to advance the frontiers of high performance computing for domain-specific applications, such as those represented by complex problems at the interface of biology and engineering,

ACKNOWLEDGMENTS

The tests on Stampede used the research project award DMS140019 (PZ) on TACC Stampede by XSEDE (Extreme Science and Engineering Discovery Environment). We thank the TACC support team for technical support and assistance. The tests on Tianhe-2 used the award of 20K computing hours (PZ) from National Supercomputer Center in Guangzhou, China. We thank the NSCC-GZ support team for assistance (YD). This project of developing the multiscale model was funded by NIH NHLBI R21 HL096930-01 and NIBIB Quantum U01EB012487 (DB).

REFERENCES

[1]. Oden JT, Belytschko T, Fish J, Hughes T, Johnson C, Keyes D, Laub A, Petzold L, Srolovitz D, Yip S. Revolutionizing engineering science through simulation. National Science Foundation Blue Ribbon Panel Report. 2006; 65

[2]. Dongarra JJ, Meuer HW, Strohmaier E. TOP500 supercomputer sites. Supercomputer. 1997; 13:89–120.

[3]. Karplus M. Development of multiscale models for complex chemical systems: from H+H(2) to biomolecules (Nobel Lecture). Angew Chem Int Ed Engl. Sep.2014 5315:9992–10005. [PubMed: 25066036]

[4]. Hodak H. The Nobel Prize in chemistry 2013 for the development of multiscale models of complex chemical systems: a tribute to Martin Karplus, Michael Levitt and Arieh Warshel. J Mol Biol. Jan 9.2014 426:1–3. [PubMed: 24184197]

[5]. Yamaguchi, T.; Ishikawa, T.; Imai, Y.; Matsuki, N.; Xenos, M.; Deng, Y.; Bluestein, D. Particle-based methods for multiscale modeling of blood flow in the circulation and in devices: challenges and future directions. Ann Biomed Eng; Sixth International Bio-Fluid Mechanics Symposium and Workshop March 28-30, 2008; Pasadena, California. Mar. 2010 p. 1225-35.

[6]. Zhang P, Gao C, Zhang N, Slepian MJ, Deng Y, Bluestein D. Multiscale Particle-Based Modeling of Flowing Platelets in Blood Plasma Using Dissipative Particle Dynamics and Coarse Grained Molecular Dynamics. Cellular and Molecular Bioengineering. 2014; 7:552–574. 2014/12. [PubMed: 25530818]

[7]. Zhang, P.; Sheriff, J.; Soares, JS.; Gao, C.; Pothapragada, S.; Zhang, N.; Deng, Y.; Bluestein, D. Multiscale Modeling of Flow Induced Thrombogenicity Using Dissipative Particle Dynamics and Coarse Grained Molecular Dynamics; ASME 2013 Summer Bioengineering Conference; 2013. p. V01BT36A002-V01BT36A002.

[8]. Zhang P, Zhang N, Deng Y, Bluestein D. A multiple time stepping algorithm for efficient multiscale modeling of platelets flowing in blood plasma. Journal of Computational Physics. 2015; 284:668–686. [PubMed: 25641983]

[9]. Pothapragada S, Zhang P, Sheriff J, Livelli M, Slepian MJ, Deng Y, Bluestein D. A phenomenological particle-based platelet model for simulating filopodia formation during early activation. Int j numer method biomed eng. 2015; 31:1–16.

[10]. Groot R, Warren P. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. The Journal of Chemical Physics. 1997; 107:4423–4435.

[11]. Symeonidis V, Karniadakis GE. A family of time-staggered schemes for integrating hybrid DPD models for polymers: Algorithms and applications. Journal of Computational Physics. 2006; 218:82–101.

[12]. Tuckerman M, Berne BJ, Martyna GJ. Reversible multiple time scale molecular dynamics. The Journal of Chemical Physics. 1992; 97:1990.

[13]. Streett W, Tildesley D, Saville G. Multiple time-step methods in molecular dynamics. Molecular Physics. 1978; 35:639–648.

[14]. Deng Y, Zhang P, Marques C, Powell R, Zhang L. Analysis of Linpack and power efficiencies of the world's TOP500 supercomputers. Parallel Computing. 2013; 39:271–279.

[15]. Feng, R.; Zhang, P.; Deng, Y. Network Design Considerations for Exascale Supercomputers; Proceedings of the 24th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2012); 2012. p. 86-93.

[16]. Zhang P, Deng Y. Design and Analysis of Pipelined Broadcast Algorithms for the All-Port Interlaced Bypass Torus Networks. IEEE Transactions on Parallel and Distributed Systems. 2012; 23:2245–2253.

[17]. Zhang P, Powell R, Deng Y. Interlacing Bypass Rings to Torus Networks for More Efficient Networks. Parallel and Distributed Systems, IEEE Transactions on. 2011; 22:287–295.

[18]. Tang Y-H, Karniadakis GE. Accelerating dissipative particle dynamics simulations on GPUs: Algorithms, numerics and applications. Computer Physics Communications. 2014; 185:2809–2822.

[19]. Bernaschi M, Bisson M, Salvadore F. Multi-Kepler GPU vs. multi-Intel MIC for spin systems simulations. Computer Physics Communications. Oct.2014 185:2495–2503.

[20]. Zhang, P.; Gao, Y. Matrix Multiplication on High-Density Multi-GPU Architectures: Theoretical and Experimental Investigations. In: Kunkel, JM.; Ludwig, T., editors. High Performance Computing. Vol. 9137. Springer International Publishing; 2015. p. 17-30.

[21]. Zhang, P.; Gao, Y.; Qiu, M. A Data-Oriented Method for Scheduling Dependent Tasks on High-Density Multi-GPU Systems; High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conferen on Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on; 2015. p. 694-699.

[22]. Liao X, Xiao L, Yang C, Lu Y. MilkyWay-2 supercomputer: system and application. Frontiers of Computer Science. 2014; 8:345–356. 2014/06/01.

[23]. Pang Z, Xie M, Zhang J, Zheng Y, Wang G, Dong D, Suo G. The TH Express high performance interconnect networks. Frontiers of Computer Science. 2014; 8:357–366. 2014/06/01.

[24]. Dubrow A. What Got Done in One Year at NSF's Stampede Supercomputer. Computing in Science & Engineering. 2015; 17:83–88.

[25]. Kesmarky G, Kenyeres P, Rabai M, Toth K. Plasma viscosity: A forgotten variable. Clin Hemorheol Microcirc. 2008; 39:243–246. [PubMed: 18503132]

[26]. Alemu Y, Girdhar G, Xenos M, Sheriff J, Jesty J, Einav S, Bluestein D. Design optimization of a mechanical heart valve for reducing valve thrombogenicity-A case study with ATS valve. ASAIO J. Sep-Oct;2010 56:389–96. [PubMed: 20613492]

[27]. Yun BM, Wu J, Simon HA, Arjunon S, Sotiropoulos F, Aidun CK, Yoganathan AP. A numerical investigation of blood damage in the hinge area of aortic bileaflet mechanical heart valves during the leakage phase. Annals of Biomedical Engineering. 2012; 40:1468–1485. [PubMed: 22215278]

[28]. Shaw, DE.; Grossman, J.; Bank, JA.; Batson, B.; Butts, JA.; Chao, JC.; Deneroff, MM.; Dror, RO.; Even, A.; Fenton, CH. Anton 2: raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer; Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis; 2014. p. 41-53.

[29]. Zhang, P. Ph.D. Dissertation. Applied Mathematics, State University of New York; Stony Brook, ProQuest: 2012. A Methodology for Design and Applications of Parallel Computers,. document ID: 1039264152

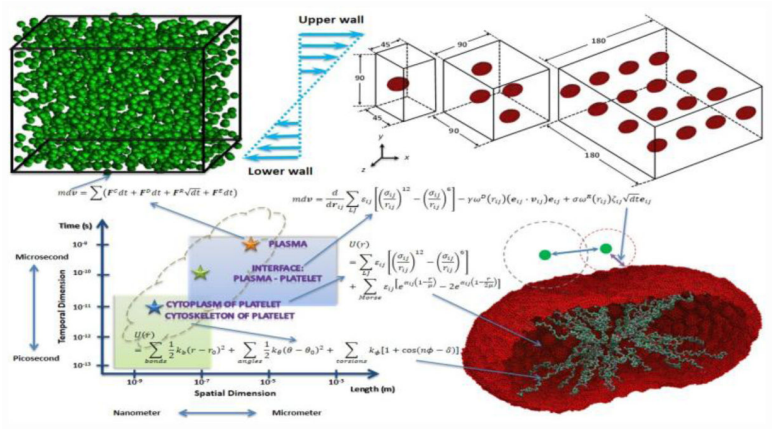

Figure 1.
Multiple spatial-temporal scales for modeling platelets under shear flow conditions

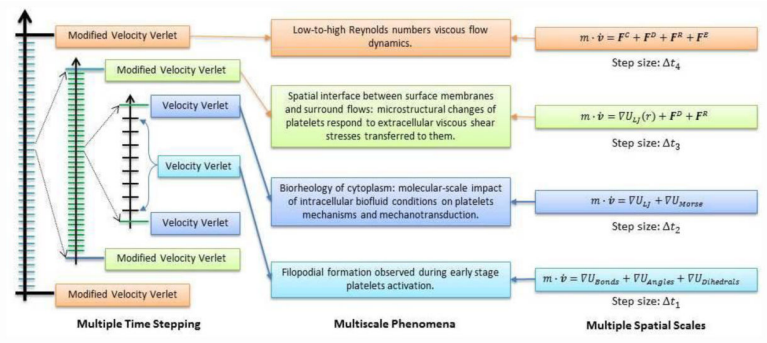

Figure 2.

The hierarchy of multiple spatial-temporal scales correlated with biological phenomena at multiple resolutions

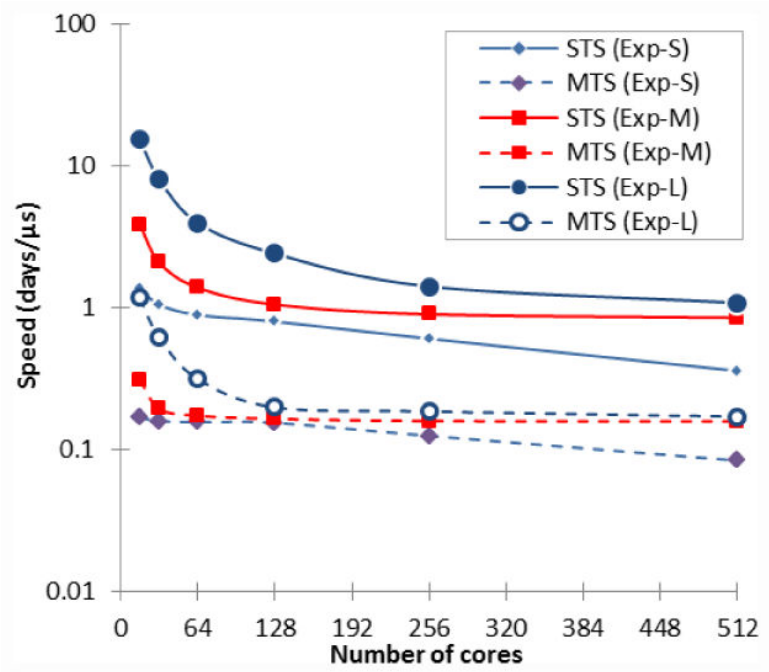

Figure 3.
Speeds of STS and MTS algorithms on Tianhe-2

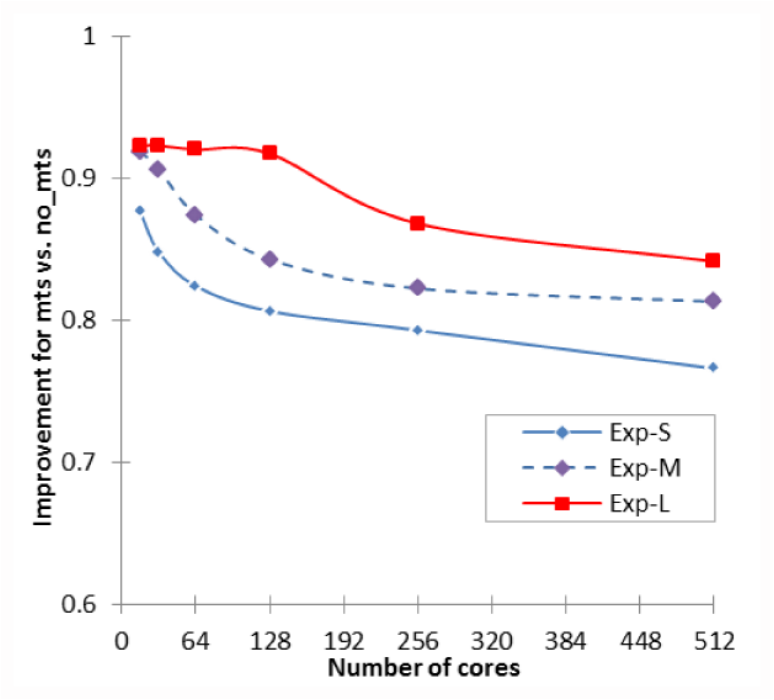

Figure 4.
Perf. Improvement of MTS over STS on Tianhe-2

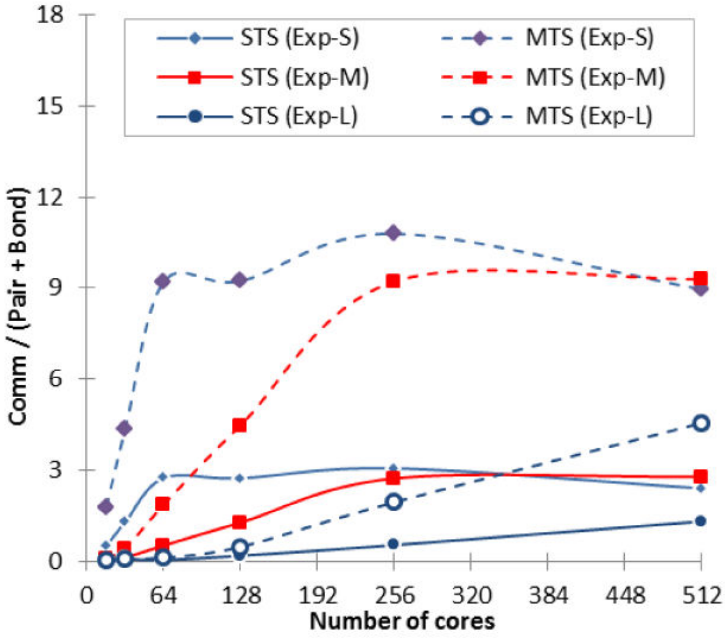

Figure 5.
Ratio of communication over computation on Tianhe-2

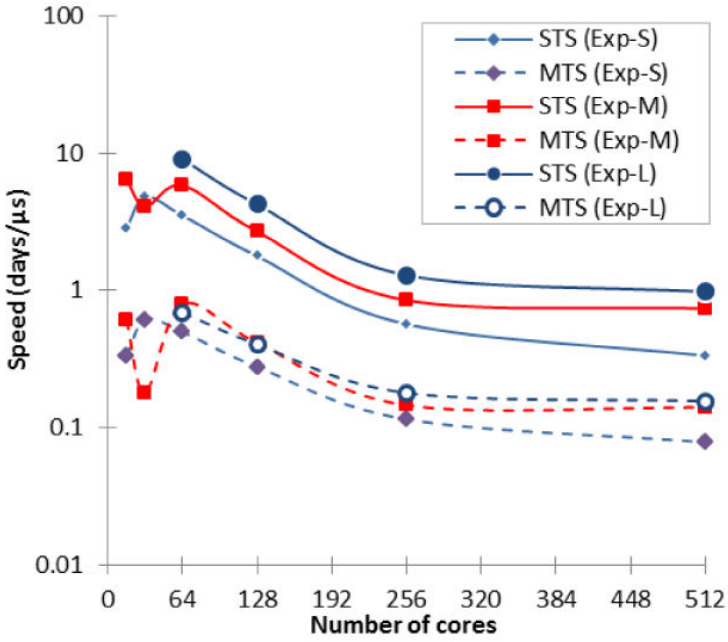

Figure 6.
Speeds of STS and MTS algorithms on Stampede

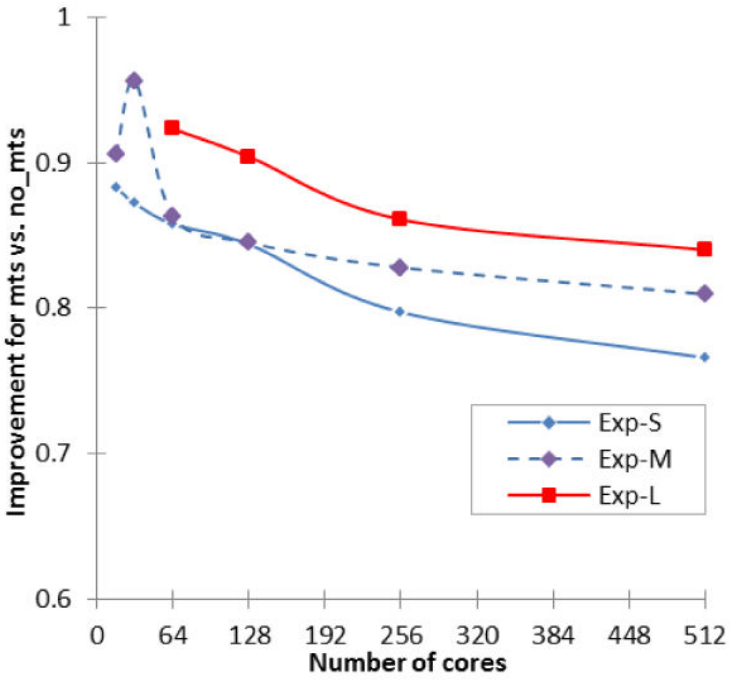

Figure 7.
Perf. Improvement of MTS over STS on Stampede

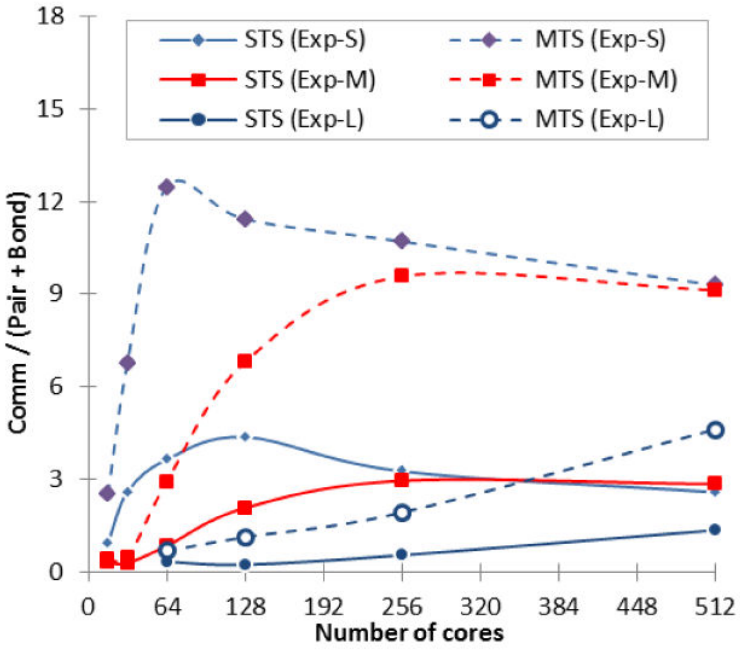

Figure 8.
Ratio of communication over computation on Stampede

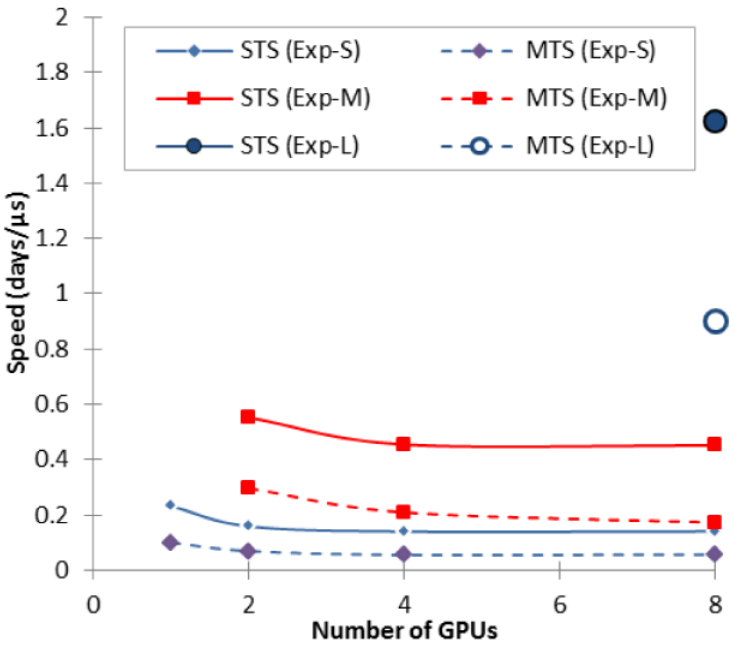

Figure 9.
Speeds of STS and MTS algorithms on 8-K40m CS-Storm

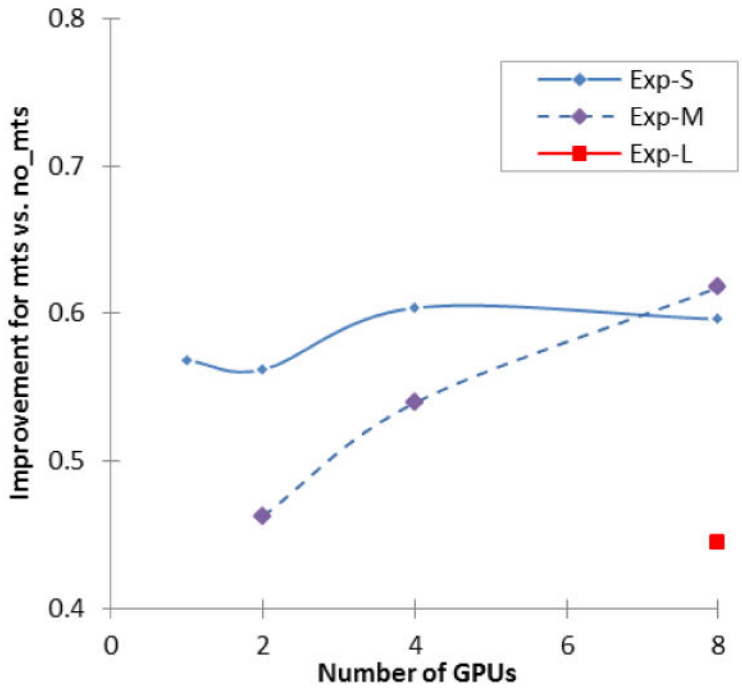

Figure 10.
Perf. Improvement of MTS over STS on 8-K40m CS-Storm

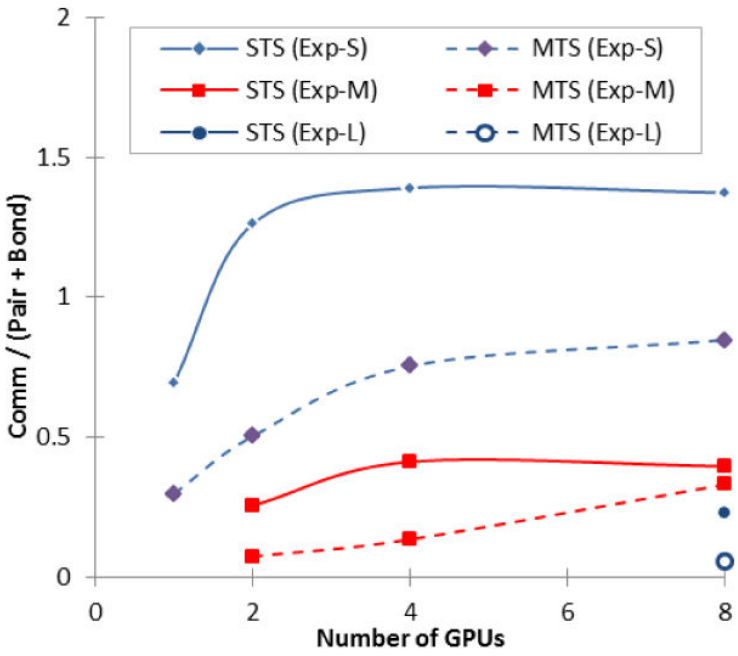

Figure 11.
Ratio of communication over computation on 8-K40m CS-Storm

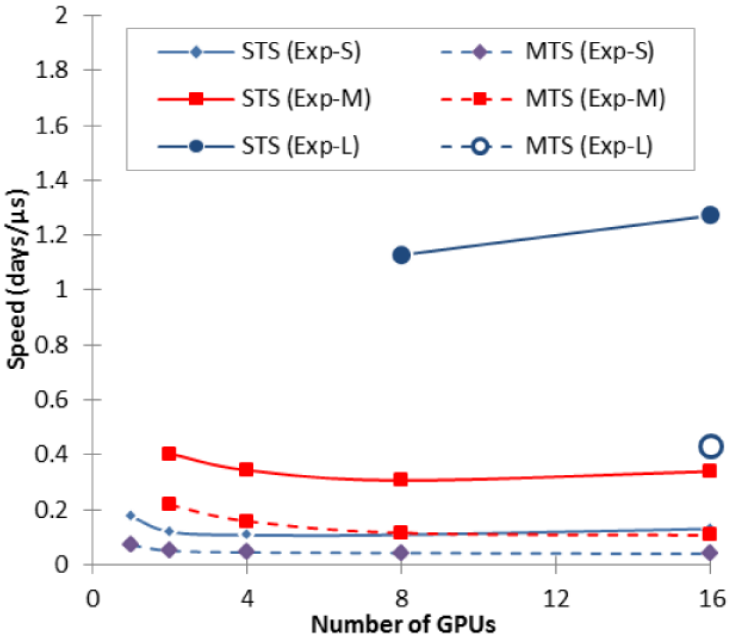

Figure 12.
Speeds of STS and MTS algorithms on 16-K80 CS-Storm

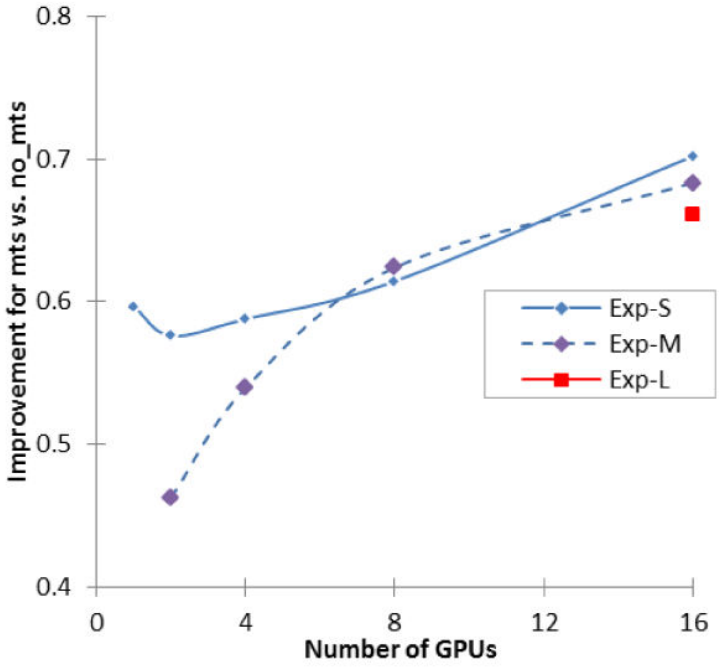

Figure 13.
Perf. Improvement of MTS over STS on 16-K80 CS-Storm

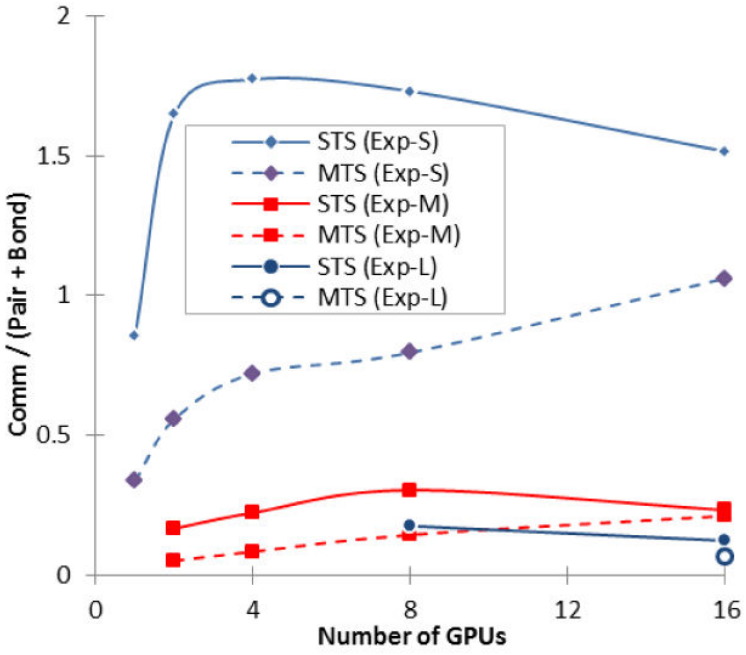

Figure 14.
Ratio of communication over computation on 16-K80 CS-Storm

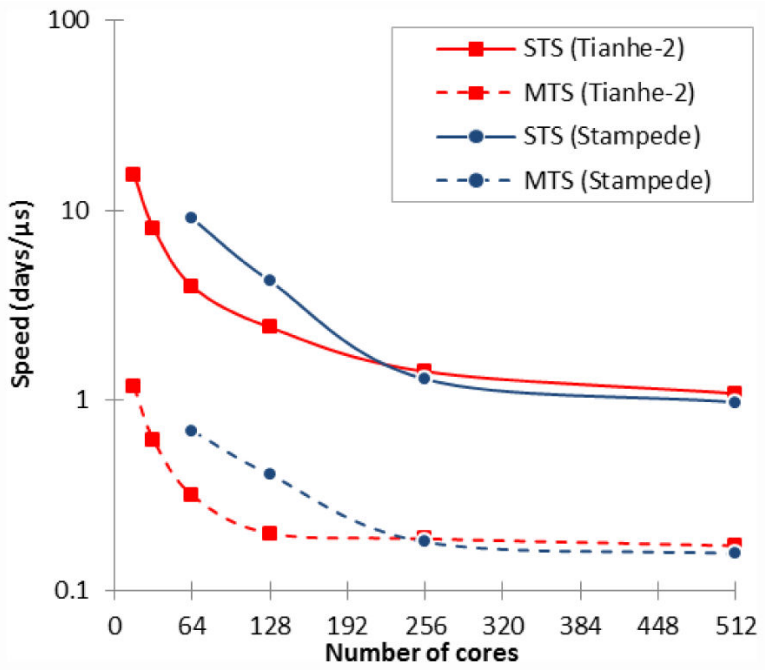

Figure 15.
Speed of STS and MTS algorithms for Exp-L on Tianhe-2 and Stampede

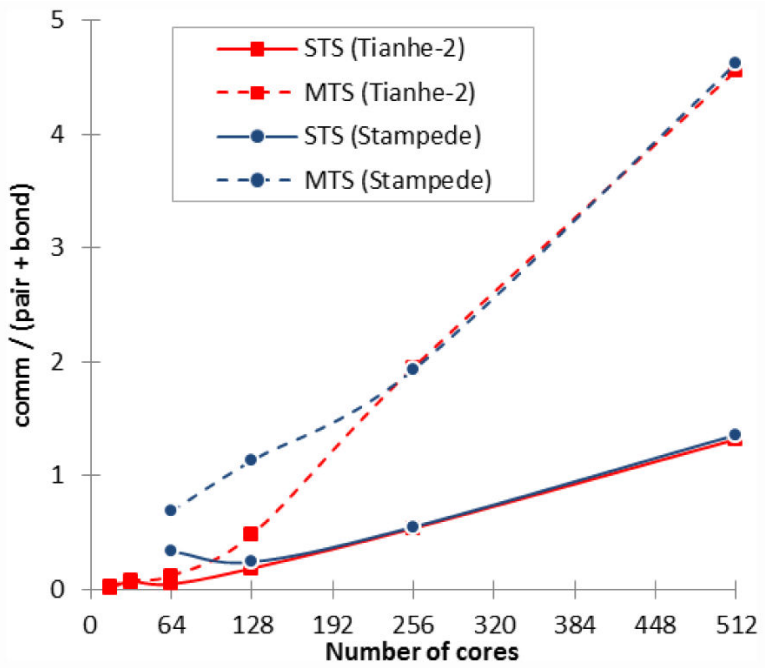

Figure 16.
Ratio of communication over computation for STS and MTS algorithms for Exp-L on Tianhe-2 and Stampede

Table 1

Nodal configurations and peak performance (GFlops) for Tianhe-2, Stampede and CS-Storm

Systems	Configuration of a Single Compute Node*	Performance per Node (GFlops)
Tianhe-2	24 cores	422.4
Stampede	16 cores	345.6
Stampede (K20)	16 cores + K20	1,515.6
CS-Storm (K40)	20 cores + 8 K40	11,840.0
CS-Storm (k80)	24 cores + 16 K80	15,440.0

Note:

* We consider CPUs and GPUs here.

Table 2

Problem sizes and dimensions of multiscale simulations for performance examinations

Experiments	# of Platelets	# of Particles	Dimensions
Exp-S	1	680,718	45×90×45
Exp-M	4	2,722,872	90×90×90
Exp-L	16	10,891,488	180×90×180

Table 3

Best performance for STS and MTS algorithms on three supercomputers (problem size: 0.7M particles)

Systems	*Resources*		*Speed (days/μs)*		*Speed (μs/day)*		*Speedup for MTS vs. STS*
	# of GPUs	**# of Cores**	**STS**	**MTS**	**STS**	**MTS**	
K40m	8	16	0.14	0.06	7.14	16.67	2.33
K80	16	16	0.13	0.04	7.69	25.00	3.25
Tianhe-2		256	0.61	0.13	1.64	7.69	4.69
Tianhe-2		512	0.36	0.08	2.78	12.50	4.50
Stampede		256	0.57	0.12	1.75	8.33	4.75
Stampede		512	0.34	0.08	2.94	12.50	4.25
Stampede/K20	16	256	0.14	0.04	7.14	25.00	3.50
Stampede/K20	32	512	0.62	0.03	1.61	35.50	20.67

Table 4

Best performance for STS and MTS algorithms on three supercomputers (problem size: 2.7M particles)

Systems	*Resources*		*Speed (days/μs)*		*Speed (μs/day)*		*Speedup for MTS vs. STS*
	# of GPUs	**# of Cores**	**STS**	**MTS**	**STS**	**MTS**	
K40m	8	16	0.45	0.17	2.22	5.88	2.65
K80	16	16	0.34	0.11	2.94	9.09	3.09
Tianhe-2		256	0.90	0.16	1.11	6.25	5.63
Tianhe-2		512	0.85	0.16	1.18	6.25	5.31
Stampede		256	0.85	0.15	1.18	6.67	5.67
Stampede		512	0.74	0.14	1.35	7.14	5.29
Stampede/K20	16	256	0.15	0.07	6.67	14.29	2.14
Stampede/K20	32	512	0.14	0.08	7.14	12.50	1.75

Table 5

Best performance for STS and MTS algorithms on three supercomputers (problem size: 10.9M particles)

Systems	*Resources*		*Speed (days/μs)*		*Speed (μs/day)*		*Speedup for MTS vs. STS*
	# of GPUs	**# of Cores**	**STS**	**MTS**	**STS**	**MTS**	
K40m	8	16	1.63	0.90	0.61	1.11	1.81
K80	16	16	1.27	0.43	0.79	2.33	2.95
Tianhe-2		256	1.42	0.19	0.70	5.26	7.47
Tianhe-2		512	1.09	0.17	0.92	5.88	6.41
Stampede		256	1.29	0.18	0.78	5.56	7.17
Stampede		512	0.98	0.16	1.02	6.25	6.13
Stampede/K20	32	512	0.29	-	3.45	-	-

Table 6

Simulation speeds (μs/day) for multiscale model and MTS algorithm on Tianhe-2, Stampede and CS-Storm

Systems / Problems	Tianhe-2 (512 cores)	Stampede (512 cores)	Stampede (512 cores, 32 K20)	CS-Storm (20 cores + 8 K40)	CS-Storm (24 cores, 16 K80)
single platelet 680,718 particles	12.50	12.50	35.50	16.67	25.00
4 platelets 2,722,872 particles	6.25	7.14	12.50	5.88	9.09
16 platelets 10,891,488 particles	5.88	6.25	-	1.11	2.33