

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/310594667>

Extreme-Scale Phase Field Simulations of Coarsening Dynamics on the Sunway TaihuLight Supercomputer

Conference Paper · November 2016

DOI: 10.1109/SC.2016.3

CITATIONS

15

READS

514

10 authors, including:



Jian Zhang

Tianjin University

514 PUBLICATIONS 8,175 CITATIONS

SEE PROFILE



Yangang Wang

Chinese Academy of Sciences

11 PUBLICATIONS 192 CITATIONS

SEE PROFILE



Lili Ju

University of South Carolina

105 PUBLICATIONS 1,910 CITATIONS

SEE PROFILE



Qiang Du

xbentury network tech.,Ltd

287 PUBLICATIONS 8,501 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Nickel-based single crystal superalloy [View project](#)



Domain decomposition-based exponential integrators [View project](#)

Extreme-Scale Phase Field Simulations of Coarsening Dynamics on the Sunway TaihuLight Supercomputer

Jian Zhang

Computer Network Information Center,
Chinese Academy of Sciences
Beijing 100190
China
Email: zhangjian@sccas.cn

Chunbao Zhou

Computer Network Information Center,
Chinese Academy of Sciences
Beijing 100190
China
Email: zhoubc@sccas.cn

Yangang Wang

Computer Network Information Center,
Chinese Academy of Sciences
Beijing 100190
China
Email: wangyg@sccas.cn

Lili Ju

Department of Mathematics,
University of South Carolina
Columbia, SC 29208
U.S.A.
Email: ju@math.sc.edu

Qiang Du

Department of Applied Physics
and Applied Mathematics,
and the Data Science Institute,
Columbia University
New York, NY 10027
U.S.A.
Email: qd2125@columbia.edu

Xuebin Chi

Computer Network Information Center,
Chinese Academy of Sciences
Beijing 100190
China
Email: chi@sccas.cn

Dongsheng Xu

Institute of Metal Research,
Chinese Academy of Sciences
Shenyang, Liaoning 110016
China
Email: dsxu@imr.ac.cn

Dexun Chen and Yong Liu

National Research Center of
Parallel Computer Engineering and Technology
Beijing 100190
China
Email: adch@263.net and liuy_99@163.com

Zhao Liu

National Supercomputing Center in Wuxi
Wuxi, Jiangsu 214072
China
Email: kimilionheart@qq.com

Abstract—Many important properties of materials such as strength, ductility, hardness and conductivity are determined by the microstructures of the material. During the formation of these microstructures, grain coarsening plays an important role. The Cahn-Hilliard equation has been applied extensively to simulate the coarsening kinetics of a two-phase microstructure. It is well accepted that the limited capabilities in conducting large scale, long time simulations constitute bottlenecks in predicting microstructure evolution based on the phase field approach. We present here a scalable time integration algorithm with large step-sizes and its efficient implementation on the Sunway TaihuLight supercomputer. The highly nonlinear and severely stiff Cahn-Hilliard equations with degenerate mobility for microstructure evolution are solved at extreme scale, demonstrating that the latest advent of high performance computing platform and the new advances in algorithm design are now offering us the possibility to simulate the coarsening dynamics accurately at unprecedented spatial and time scales.

precision and demonstrated good weak and strong scalability up to 10.6 million cores of the full system.

Performance Attributes

Category of achievement	Peak performance: 50,579 TFlops Scalability: up to 10.6 million cores Time-to-solution: coarsening dynamics governed by Cahn-Hilliard equation on 300 billion grid points at 0.133 s/step, full simulation took 150,000 steps.
Type of method used	Scalable compact localized exponential time differencing
Results reported on the basis of	Whole application except I/O
Precision reported	Double precision
System scale	Results measured on full-scale system
Measurement mechanism	Flop count

Justification for ACM Gordon Bell Prize

Severely nonlinear and stiff phase field equations were accurately solved at unprecedented spatial and time scales. Our simulations on the Sunway TaihuLight supercomputer achieved a peak performance of 50,579 TFlops in double

I. INTRODUCTION

In materials science, coarsening refers to changes in spatial scales over time that are associated with mesoscale morphological patterns. Such patterns are often referred to as the microstructure of the material. The microstructure has a critical

role in determining many important material properties, including strength, hardness, ductility, and conductivity. To predict and optimize these microstructure-sensitive macroscopic material properties, it is important to obtain a deeper insight into this microscopic structure evolution. Diffuse-interface or phase-field based modeling and simulations have become widely used methodologies for describing the microstructure formation in materials. These methodologies can be viewed as physically motivated level-set approaches on account of their energy-based variational formalism. In phase field modeling, an inhomogeneous system is described by a set of conserved (c_1, c_2, \dots) and nonconserved (η_1, η_2, \dots) field variables. The total free energy of the system is then given by

$$\begin{aligned} \mathcal{E} = & \int_{\Omega} [f(c_1, c_2, \dots; \eta_1, \eta_2, \dots) + \sum_i \alpha_i |\nabla c_i|^2 \\ & + \sum_{i,j,k} \beta_{ij} \nabla_i \eta_k \nabla_j \eta_k] d\mathbf{x}. \end{aligned}$$

where f is the local free-energy density and α_i and β_{ij} are the gradient energy coefficients [1]. The evolution of the field variables can be obtained by solving the following Cahn-Hilliard [2] and Allen-Cahn [3] equations,

$$\begin{aligned} \frac{\partial c_i}{\partial t} &= \nabla \cdot (M_{ij} \nabla \frac{\delta F}{\delta c_j}) \\ \frac{\partial \eta_p}{\partial t} &= -L_{pq} \frac{\delta F}{\delta \eta_q}, \end{aligned}$$

where M_{ij} and L_{pq} correspond to atomic or interface mobility [1]. The conservative Cahn-Hilliard equation is a key component of typical phase field models, along with the nonconservative Allen-Cahn or Ginzburg-Landau equation. The Cahn-Hilliard equation has been extensively applied to model coarsening kinetics of two-phase microstructures. For example, asymptotic analysis of coarsening rates has been given in [4], [5], [6], [7] and numerical simulations can be found in [8], [9], [10], [11], [12], [13], [14]. Further theoretical justifications have been given in [15]. Owing to the presence of strong nonlinear effects and varying spatial and temporal scales in the solutions, numerical simulations have posed significant challenges to computational material scientists and the community of high performance computing. Most of the existing simulations are performed in two spatial dimensions; very few large-scale three-dimensional (3D) simulations have been conducted. (A widely cited work [16] reported simulations on the order of 128^3 , which is also the resolution given in one of the latest works [17].) However, most material processes exhibit essential 3D features. It is well accepted that two-dimensional (2D) studies often produce faster dynamics than experimental observations. Moreover, the limited resolution in 3D simulations constitutes computational bottlenecks in validating predictions based on the phase field approach [18]. Thus, enhanced capability in carrying out large-scale and long-time-scale phase field simulations in 3D space is greatly needed to provide significantly more accurate microstructure

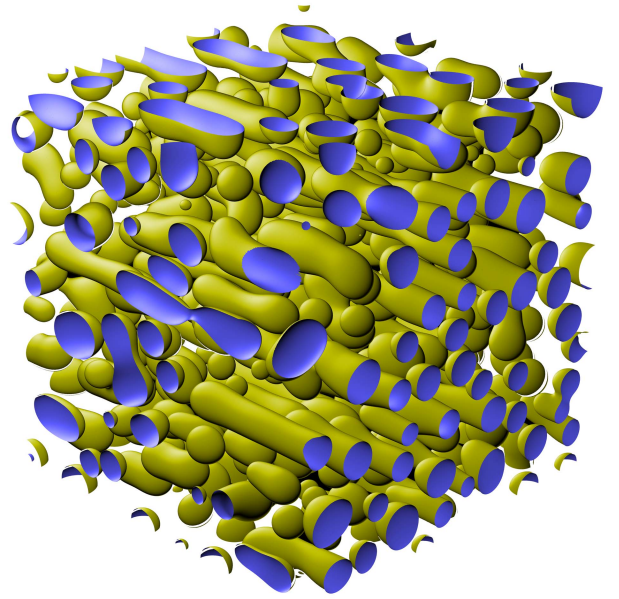


Fig. 1. Grains in various sizes and shapes coexist, revealing transitions into columnar like morphologies with unprecedented precision and details. (Previously reported experiments have been largely limited to resolutions with a few cylinder-like structures, at most, in the computational domain.)

morphologies and to offer true insight into realistic material processes [19]. Traditionally, a predominant approach used by computational materials scientists for numerically integrating the phase-field equation has been the explicit forward Euler method. To maintain the stability, the time step must be very small. This becomes prohibitive for long time simulations based on the Cahn-Hilliard equations that involve high-order spatial differential operators. To overcome this major obstacle, large time step schemes are essential for successful simulation of coarsening dynamics. However, the widely used semi-implicit and implicit schemes based on fast Fourier transform (FFT) or iterative linear system solvers encounter scalability and effective performance issues at extreme scales. Moreover, our most recent findings have revealed that high-order in-time discretization is particularly effective and almost necessary for modeling coarsening kinetics on account for the need for accuracy over long time integration [20], [21]. This presents additional challenges to the design of numerical algorithms.

In this paper, we therefore employ the Cahn-Hilliard equation with the standard double-well potential $f = f(u)$ in the free energy and a variable mobility,

$$u_t = \nabla \cdot (M(u) \nabla \mu), \quad \mu = -\gamma \Delta u + f'(u), \quad (1)$$

where $\sqrt{\gamma}$ gives the width of the phase transition regions. The variable mobility $M = M(u)$ produces significant effects on the coarsening kinetics [11], in which bulk and interface diffusion play different roles. Meanwhile, it makes the model even more nonlinear than the constant mobility case, which thereby makes it more difficult to design stable and high-order

numerical discretization. Effective simulations of the equation (1) illuminate the quantitative understanding of a fully 3D microstructure evolution in a binary system. We demonstrate here that the latest computing platform and advances in algorithm design have enabled the possibility of accurately simulating Cahn-Hilliard dynamics at unprecedented spatial and time scales. This capability provides useful benchmarks and guidelines to help validate and further enhance the predictive capability of phase field simulations of more complex materials systems.

II. CURRENT STATE OF THE ART

Numerical simulations of phase field equations are conventionally performed by explicit finite difference time marching with stencil computation, which is a sub-optimal but easily parallelizable algorithm. The main drawback to this type of algorithm is that the time step size is severely restricted by the stability issue raised by the high-order spatial derivatives of the solution variables. The higher the order is, the worse the problem is. For Allen-Cahn-type equations that involve second-order spatial derivatives of the solution variables, the above restriction is not prohibitive, and successful large-scale simulations have been conducted. The optimal exemplar may be the 3D phase-field-model-based dendritic solidification process of Shimokawabe et al. [22], which won the Gordon Bell Prize for Special Achievements in Scalability and Time-to-Solution in 2011. In this esteemed work, a standard second-order finite difference discretization in space and first-order explicit forward Euler discretization in time were used. This approach achieved 1.017 petaflops (PFlops) in single precision and 259.2 TFlops in double precision using 4,000 GPUs along with 16,000 CPU cores on the TSUBAME2.0 supercomputer (peak 2.4 PFlops in double precision and 4.8 in single precision). In the largest configuration, 277 billion grid points were used and 94% parallel efficiency (weak scaling) was achieved. Further investigations of binary dendritic growth were performed by Yamanaka et al. [23] and Takaki et al. [24], [25] on the TSUBAME 2.0 and TSUBAME 2.5 GPU supercomputers, respectively. Furthermore, Bauer et al. [26] recently simulated the ternary eutectic directional solidification process with an Allen-Cahn-type phase field model based on the grand potential approach. The phase field equations were solved using finite difference methods and an explicit Euler scheme for time discretization. Their codes scaled up to 262,144 cores using 1,048,576 message passing interface (MPI) processes on the Blue Gene/Q machine JUQUEEN, and a 25% peak performance on the node level was reported.

For phase field models involving higher-order spatial derivatives, such as Cahn-Hilliard models (fourth order) and phase field crystal models (sixth order or higher), explicit time marching schemes are ineffective in large-scale long-time-scale simulations on account of severe restrictions in the time step size, which must be orders of magnitude smaller (the order can easily reach four or above) than those for Allen-Cahn-type models to ensure numerical stability. Semi-implicit or implicit time marching schemes are usually employed, such

as in [27] and [28]. The main challenge for these approaches is to solve a highly ill-conditioned massive sparse linear or nonlinear system. The ill-conditioning stems from the severe stiffness and nonlinearity of the governing partial differential equations (PDEs). To overcome this obstacle, a sophisticated multi-level preconditioner-solver must be designed and efficiently implemented.

A successful extreme-scale simulation based on the above approach has yet to be presented. Nevertheless, illuminating this topic is the work of Rudi et al. [29], which won the Gordon Bell Prize for Special Achievements in Scalability in 2015. Although this work addressed a different problem, it solved a highly nonlinear and ill-conditioned problem arising from mantle convection using a hybrid multi-level preconditioner-solver that scaled up to 1.5 million cores on the Blue Gene/Q machine Sequoia. The solver achieved 97% and 33% parallel efficiency over a 96-fold increase in parallelism in weak and strong scaling, respectively. On the other hand, iterative solvers based on matrix-vector production face the difficulty of achieving high performance. Despite improvements engendered by certain techniques, such as matrix-free and structuring, the node performance remains low compared with other applications.

Other than conventional explicit and implicit finite difference time marching schemes, various approaches for integrating PDEs exist. Among these approaches, exponential time differencing (ETD) methods are popular for stiff temporal differential equations. Various studies and applications relating to this subject have been undertaken, such as in [30], [31], [32], [33], [34], [35]. In their respective methods, the linear operators of the high-order derivative are precisely handled. As a result, the corresponding stability constraint is completely removed, and large time steps can be used. Furthermore, a sound review and additional references on exponential-integrator-based methods are provided in [36]. The computation of matrix exponentials has been a popular subject of numerical analysis for years [37]. A key for successful applications of ETD schemes to numerical PDEs is the effective computation for matrices associated with the discretization of differential operators. We refer to further discussions in the review [36] and additional references on exponential-integrator based methods provided there. Meanwhile, we designed compact exponential time differencing (cETD) schemes for phase field equations [20], [21], where the compact representation of finite difference spatial discretization, operator splitting, and the ETD method are combined to produce stable, efficient, and high-order accurate numerical schemes. The stability is a consequence of the operator splitting and the fact that ETD schemes are essentially semi-implicit schemes where the use of matrix exponential eliminates the need to solve linear systems. This approach enabled us to numerically study the coarsening kinetics through 2D simulations with much more refined spatial resolution and higher time accuracy than those in existing literature. However, it requires advances in both the algorithm and implementation to conduct realistic extreme scale 3D simulation.

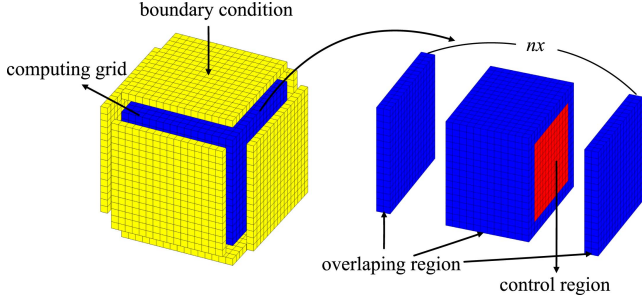


Fig. 2. Inner boundary conditions and overlapping domain decomposition. *Left:* Subdomain computing grid (blue). The boundary conditions for u and μ (equation 1) are given by the adjacent subdomains and separately saved as 2D arrays (yellow). *Right:* Computing grid is divided into the control region (red) and a two-layer thick overlapping region (blue). The grid points in the control regions of all subdomains have a one-to-one mapping on the physical grid. The values of solution variable u on the overlapping grid points are replaced by the corresponding values in the control regions of the adjacent subdomains during the integration process.

III. INNOVATIONS REALIZED

A. Algorithmic innovation

1) *Algorithm overview:* We designed the scalable compact localized exponential time differencing (ScLETDT) algorithm which combines the original second-order cETD-RK2 scheme [20], [38] with domain decomposition and localization of matrix exponentials. This innovation is essential for the successful simulation of extreme-scale coarsening dynamics on an extreme-scale supercomputer. The computational cost of the cETD is $O(N^{(d+1)/d})$ per time step where d denotes the space dimension and N the total number of grid points. Thus, it is not highly scalable. Instead of using the global exponential matrices to exactly integrate, we now use the localized exponential matrices to perform the integration approximately. The subdomains respectively integrate with appropriate boundary conditions for u and μ (equation 1) given by the adjacent subdomains (Figure 2, left). Two-layer-thick overlapping regions between adjacent subdomains are introduced (Figure 2, right) to enhance the numerical stability of the ScLETDT algorithm and reduce the errors caused by the inexact localized exponential time integration. Recall that the original cETD is essentially a stable time-marching algorithm based on finite difference spatial discretization. Although this coupling is similar to domain decomposition for conventional finite difference schemes, it is important to demonstrate the particular effectiveness when the localization is applied to the Cahn-Hilliard equations. The resulting ScLETDT scheme remains non-iterative and still allows large time-steps to be taken without producing instability nor degrading accuracy.

Computing matrix exponentials has been a problem of much interests to the scientific computing community, see for example [39], [40], [41] and [42], [43]. Our localization approach is new in the sense that we utilize the particular structure of the matrices that come from the discretization of PDEs and adopt a domain decomposition like approach to obtain fast and accurate evaluation while attaining scalability.

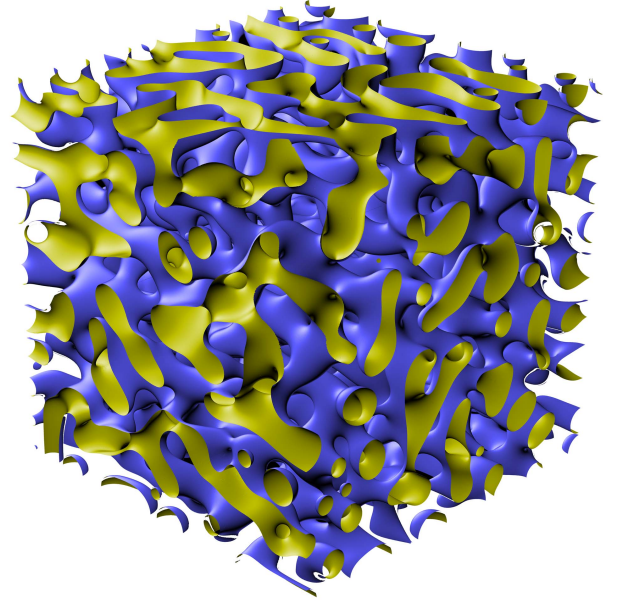


Fig. 3. Configuration of a two-phase system during coarsening process governed by the Cahn-Hilliard equation with constant mobility. Blue and yellow indicate the two sides of the interface between the two phases.

2) *Accuracy and efficiency:* In terms of strong scaling, as the number of subdomains increases, the approximation of the global exponential matrices degrades. It is expected that the error associated with the inexact integration increases. We thus conducted numerical experiments to address this issue. The benchmark solutions are obtained by running the second-order cETD-RK2 scheme (without domain decomposition) on a 512^3 grids for a long period of time with various time step sizes. A series of tests using the same initial configuration and time step sizes are then conducted on the 2nd order ScLETDT schemes with various domain decompositions. The

TABLE I
ERROR CAUSED BY LOCALIZED INTEGRATION

Time step-size=4,000 δt		
number of subdomains	relative L^2 error	relative energy error
$2 \times 2 \times 2$	0.0053%	0.0010%
$4 \times 4 \times 4$	0.0095%	0.0035%
$8 \times 8 \times 8$	0.0162%	0.0068%
Time step-size=8,000 δt		
number of subdomains	relative L^2 error	relative energy error
$2 \times 2 \times 2$	0.0339%	0.0023%
$4 \times 4 \times 4$	0.0568%	0.0106%
$8 \times 8 \times 8$	0.1089%	0.0282%
Time step-size=20,000 δt		
number of subdomains	relative L^2 error	relative energy error
$2 \times 2 \times 2$	0.2107%	0.0183%
$4 \times 4 \times 4$	0.3216%	0.0782%
$8 \times 8 \times 8$	0.5654%	0.1800%

* δt denotes the maximum time step size allowed for the explicit Euler scheme on account of the stability constraint.

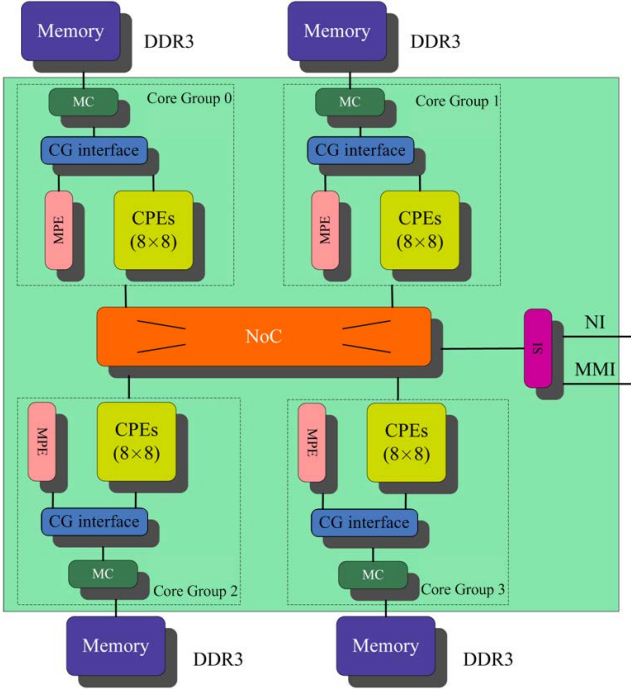


Fig. 4. The SW26010 many-core processor architecture. It consists of management processing elements (MPEs), computing processing element (CPE) clusters, memory controllers (MCs), and the system interface (SI). All modules are interconnected via network on chip (NoC).

differences between the resulting solutions and the corresponding benchmark solutions are then the errors caused by the localized integration. To illustrate the efficiency of the ScLETd scheme, we use the maximum allowed time step-size (owing to the stability constraint) of the conventional explicit Euler scheme, δt , as the time unit to present the results. The width of the testing time window is 40 million δt , and the time step sizes of the ScLETd scheme range from $4,000\delta t$ to $20,000\delta t$. The relative errors are reported in Table I. Over a 512-fold increase in parallelism, the error caused by localized integration remains negligible.

We next conduct large scale simulations to demonstrate the accuracy and efficiency of the ScLETd scheme in production simulations. In the coarsening dynamics governed by the Cahn-Hilliard equation (1), a characteristic length scale l grows similarly to a power of time

$$l(t)^\alpha = l(0)^\alpha + ct$$

for some positive constant c . The Cahn-Hilliard energy is proportional to the interface area that has a dimension of l^{-1} . Consequently, we expect that

$$\ln \mathcal{E} = -\frac{1}{\alpha} \ln t + C + o(1)$$

as t goes to infinity. When mobility $M(u)$ is a constant independent of u , the coarsening is determined by a diffusion field in both phases. In this case, it is well established that coarsening rate $1/\alpha$ is expected to be $1/3$. Because the theoretical coarsening rate is the limit as t goes to infinity,

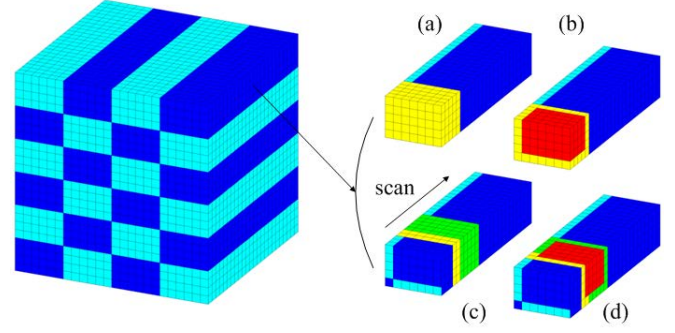


Fig. 5. Seven-point stencil computation using CPEs. *Left*: Computing grid is divided into sticks and distributed to the 64 CPEs. *Right*: Scanning process designed to calculate the target values on a stick on which the data are loaded to the scratch pad memory (SPM) via direct memory access (DMA) in an incremental pattern. (a): Block of data (yellow) is loaded at the beginning of each scan (in the ScLETd algorithm, the boundary condition is incorporated into the exponential integration, requiring no consideration here). (b): Target values (red) are computed and put to the main memory via DMA. (c): Second block of data (green) is loaded while the last two slices of data (yellow) are retained in SPM and reused in the second block calculation. (d): Target values (red) are computed and put to the main memory via DMA. The same process is applied to other blocks.

an accurate long time simulation must be conducted to obtain a good approximation of this asymptotic behavior. We thus employed $32 \times 32 \times 32$ domain decomposition with $128 \times 128 \times 128$ subdomain grids. The physical grid is of the size of $3,972 \times 3,972 \times 3,972$. Starting from random mixtures of the two components, we ran three simulations with time step-size $10,000 \delta t$ for 20,000 steps each. Figure 3 shows the final configuration of one simulation. The estimated coarsening rates of the three runs were 0.3301, 0.3375 and 0.3221, respectively. Compared with the theoretical prediction of $1/3$, the average error was 1%, indicating excellent spatial and temporal accuracy.

B. Implementation

Node architecture: We first briefly introduce the SW26010 many-core processor [44] in the Sunway TaihuLight super-computer (see section IV for details). The architecture of SW26010 processor is depicted in Figure 4. There are 260 cores in the SW26010 processor, and they are divided into four core groups (CGs), each consisting of one management processing element (MPE) and 64 computing processing elements (CPEs). The MPEs are fully functional cores. The CPEs run only in the user mode. Each core group is assigned 8 GB of the shared main memory. Cross-group memory access causes significant bandwidth degradation. Thus, it is recommended to use MPI between the core groups. Each CPE has 64 KB scratch pad memory (SPM) and can access the main memory through direct memory access (DMA).

Subdomain routines: In our realization, each subdomain is handled by a CG. Field variables are considered dense 3D tensors in the ScLETd algorithm. The Laplace operator Δ in equation (1) is discretized using a second-order

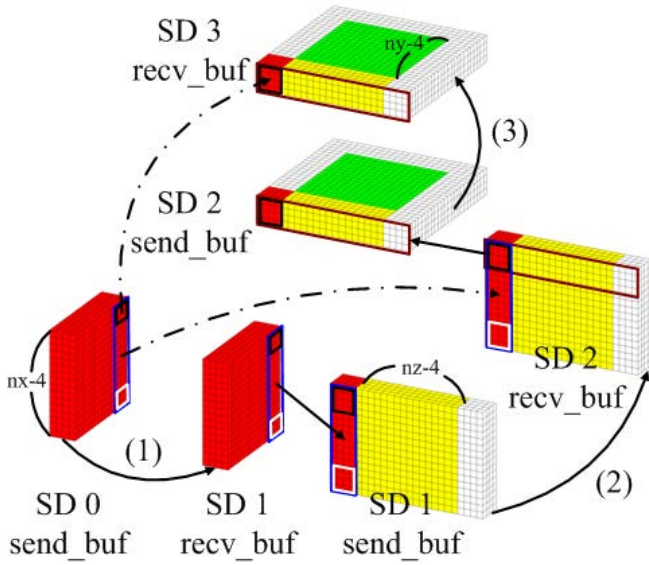


Fig. 6. Inter-subdomain communication. MPI calls are invoked by MPEs. Packing/unpacking is performed by CPEs via DMA. Edge and corner data are transferred together with face data using enlarged buffers. After three rounds of communication between face-adjacent subdomains in x, y, and z directions, face, edge, and corner data arrive to their destinations. (1) Round 1: face data (red) from control region are packed into send buffers and sent to adjacent subdomain. During communication, face data in second direction (yellow) are packed into send buffer. After receipt, edge/corner (red) data are copied to send buffer, where face data are already filled. (2) Round 2: during communication, face data in third direction (green) are packed into send buffer. (3) Round 3: data transfer in the third direction.

central finite scheme. The compact representation and eigen-decomposition of the discrete differential operator results in six dense matrices of the size of $N_x \times N_x$, $N_y \times N_y$, $N_z \times N_z$, respectively, where N_x, N_y, N_z are the dimensions of the subdomain grid. The cETD subdomain solver mainly involves the dot product of a 3D tensor (field variable) with a matrix, and stencil/pointwise computing for the nonlinear terms. Three types of tensor dot products exist in the cETD algorithm:

$$R_{ijk} = \sum_l P_{il}^x U_{ljk}, \quad (2)$$

$$R_{ijk} = \sum_l P_{jl}^y U_{ilk}, \quad (3)$$

$$R_{ijk} = \sum_l P_{kl}^z U_{ijl}. \quad (4)$$

They correspond to contractions in the x , y and z directions, respectively. A highly optimized parallel math kernel library that runs on one CG is available to the users. We thus designed

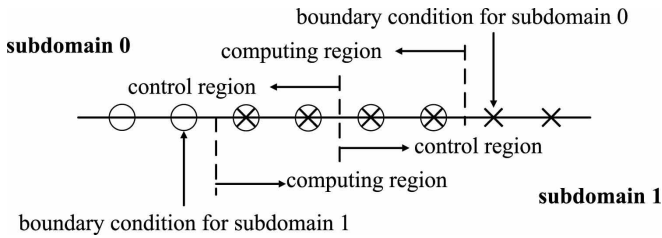


Fig. 7. Overlapping between adjacent subdomains and the positions where boundary conditions for u and μ are given.

two schemes using DGEMM to perform these tensor dot products. The contraction in the x direction (2) can be considered a matrix multiplication and performed by one DGEMM call with matrix size parameters of $m = N_x$, $k = N_x$, $n = N_y \times N_z$ (for matrices of size $m \times k$ and $k \times n$, respectively). The contraction in the y direction (3) is carried out either by transposing the field variable into the $y - z - x$ format, and then invoking one DGEMM call with matrix size parameters of $m = N_y$, $k = N_y$, $n = N_x \times N_z$ or by invoking N_z DGEMM calls with $m = N_x$, $k = N_y$, $n = N_y$,

$$(R_k)_{ij} = \sum_l P_{jl}^y (U_k)_{il} = (U_k (P^y)^T)_{ij},$$

to avoid tensor transposition. The contraction in the z direction (4) is performed by transposing the field variable into the $z - x - y$ format and invoke one DGEMM call with $m = N_z$, $k = N_z$, $n = N_x \times N_y$. The transposition of the resulting tensor R back to the $x - y - z$ format is incorporated in the pointwise computing subroutines. The transposition is realized using CPEs exploiting DMA. The stencil/pointwise computation is also performed by the CPEs, Figure 5 illustrates the seven-point stencil computing scheme. The performance tuning of the subdomain routines mainly involves choosing tensor dot product schemes for different subdomain dimensions and adjusting the block sizes in the transposition and stencil/pointwise subroutines.

Inter-subdomain: In the ScLETd algorithm, subdomains are coupled together through exchanging solution variable u in the overlapping region and obtaining boundary conditions for both u and μ from the adjacent subdomains. Figure 2 depicts the grids. Unlike in the standard stencil computation, we did not add a halo region outside the computing grid because efficient tensor dot production requires contiguous storage. Figure 7 illustrates the overlapping and boundary conditions in one dimension. The value of μ is calculated using u with the standard seven-point stencil. Hence, four layers of data from the face, edge, and corner adjacent subdomains must be transferred to a subdomain. Moreover, the data are used in different places. The inner two layers involved in the tensor dot product proceed to the overlapping region, while the outer three layers are needed when calculating the boundary conditions. A three-round MPI communication scheme is employed to transfer data between the subdomains, where the packing/unpacking is conducted by CPEs and is overlapped with the communication. This approach exploits the fast DMA while enabling incorporation of MPI process mapping optimization for stencil applications provided by the machine producer.

Asynchronous I/O: In our code, only the solution variable u requires output. In the largest configuration, each subdomain produces approximately 1 GB of solution data. At the checkpointing/outputting point, the solution is copied to a solution buffer via DMA, MPE then invokes asynchronous I/O.

C. Extreme scale coarsening dynamics

Exploiting the computing power of the Sunway TaihuLight supercomputer and the high-order accuracy of the ScLETd

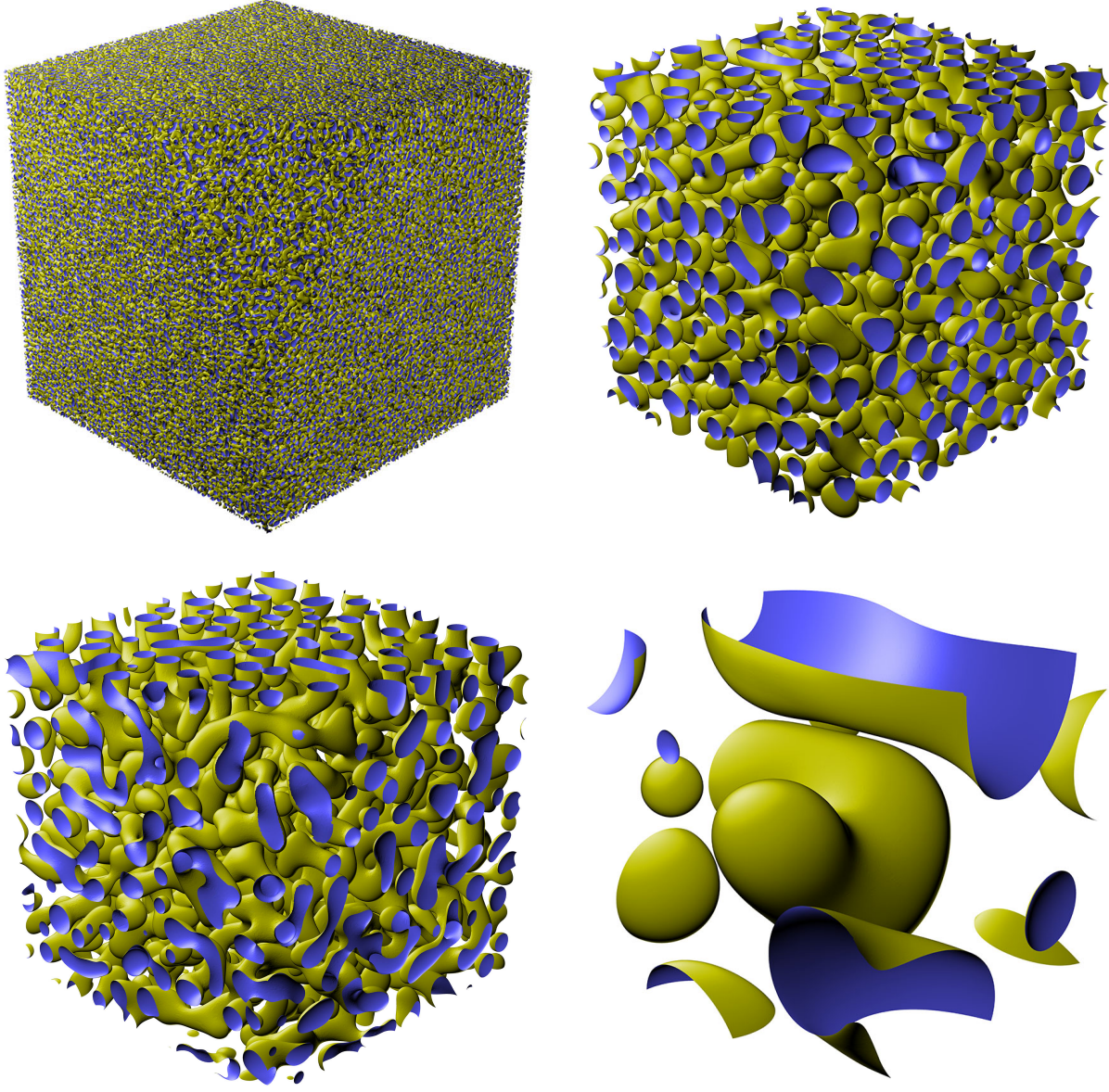


Fig. 8. Configurations of a two-phase system during the coarsening process governed by the Cahn-Hilliard equation with degenerate mobility. Complex morphological evolution, separation and coarsening occur simultaneously. The simulation grid size is $6,700^3$, which enables quantitative analysis of three dimensional coarsening dynamics. *Upper left*: early stage configuration. *Lower left*: enlargement of a subset of size $1,116^3$ in the early stage. *Upper right*: late stage configuration. *Lower right*: enlargement of a subset of size $1,116^3$ in the late stage.

scheme, we are able to simulate the coarsening dynamics of two-component systems at an unprecedented spatial resolution and time scale. This enhanced computational capability significantly increases integration accuracy over long periods of time and large spatial domains and reduces the effects due to a finite domain size. Hence, for the first time, a computational tool is provided that enables the conducting of quantitative analysis of 3D coarsening dynamics.

We conducted coarsening simulations, wherein mobility $M(u)$ was constant in one phase and degenerated to zero in another phase. This setup corresponded to the case in which

the mobilities of the two components of the alloy differ by orders of magnitude. In this case, the coarsening occurs at a slower rate and the governing equation becomes even more nonlinear and stiff. In other words, the time scale of the physical problem increases and the computational problem becomes more difficult to solve. The ScLET code successfully overcame the obstacles and produced interesting simulation results. Snapshots obtained from high-resolution simulations using the $3,972 \times 3,972 \times 3,972$ grid size are shown in Figure 1 and Figure 9.

Another simulation used $54 \times 54 \times 54$ domain decomposition

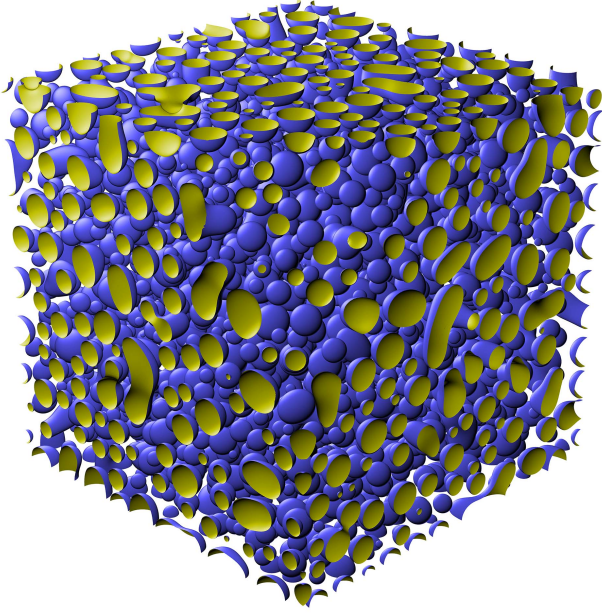


Fig. 9. Configuration of a two-phase system during the coarsening process governed by the Cahn-Hilliard equation with degenerate mobility. Spherical droplets form early and coarsening dominates.

with $128 \times 128 \times 128$ subdomain grids. The physical grid was $6,700 \times 6,700 \times 6,700$ with 300 billion grid points. Using 10,235,160 cores (157,464 core groups), the average speed of the simulation was 0.133 s per time step. The full simulation took 150,000 time steps with various step sizes. Figure 8 presents snapshots obtained from the early and late stages of the coarsening process. The extreme-scale simulation reveals a detailed 3D morphology evolution with accurate separations and collisions that simultaneously occurred.

IV. HOW PERFORMANCE WAS MEASURED

The Sunway TaihuLight supercomputer has 40 cabinets, which hold 40,960 SW26010 many-core processors. All 260 cores are integrated in one SW26010 processor and are divided into four core groups. Each core group consists of one MPE and an 8×8 cluster of CPEs. An MPE is a fully functional core, which integrates two levels of cache structures (a 32-KB L1 cache and a 256-KB L2 cache). It employs a single-instruction multiple-data (SIMD) vector extension. MPEs can run in both user and system mode, and support superscalar, out-of-order issue, out-of-order execution, and speculative execution. CPEs support dual-issue, out-of-order execution, and static branch prediction. They only run in the user mode, and support double-precision floating-point multiply-add and divide/square-root operations with a 32B SIMD vector extension. Both MPEs and CPEs work at 1.45GHz. The theoretical peak performance for a SW26010 processor is 3062.4 GFlops. Each core group is assigned 8 GB of main memory. Intergroup memory access is allowed; however, it will cause approximately 20% bandwidth degradation. Each

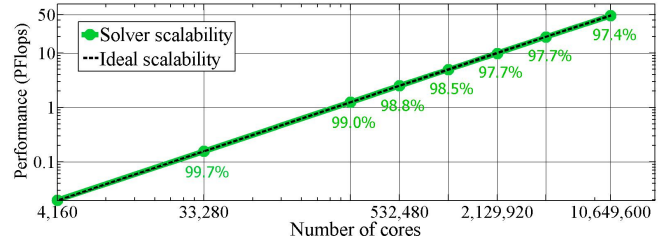


Fig. 10. Weak scalability results from 4,160 cores (64 core groups) to 10,649,600 cores (163,840 core groups). Numbers along the graph line indicate efficiency with respect to ideal scaling (the efficiency baseline is the 4,160 cores result). The largest configuration has 21.5 trillion grid points. The highest performance is 50,579 TFlops.

CPE has a 64-KB SPM and can access the main memory through DMA. The computing nodes (one SW26010 processor per node) are connected by a hybrid network. A group of 256 nodes are fully-connected and form a super-node. Each super-node is connected to the central network via 64 optical fibers, which means that four nodes share a fiber. The I/O network connects 272 I/O nodes to the supernode network. Two optical fibers dedicated to I/O are assigned to each supernode. The link bandwidth of the network is 16 GB/s, the aggregate bandwidth is 462 TB/s, the bi-section bandwidth is 70 TB/s, and the I/O is 288 GB/s.

The programming language environment supports C, C++ (for MPE only), Fortran, and MPI, OpenMP, Pthread, and OpenACC for parallel programming. A parallel math kernel library that runs on one core group and an Athread interface (similar to Pthread) are also provided to help programming for the CPEs. Our code is written in C for both MPE and CPE. It adopts the Athread interface to spawn and fork threads on the CPEs and to conduct DMA operations. We employed PAPI [45] to measure the flops. A C code which performs the same computation using CPU only is compiled and run on CPU to obtain the Flop count.

V. PERFORMANCE RESULTS

In this section, we present peak performance and scaling results and conduct node performance analysis. The main results are shown in Figure 10. The code maintains 97.4% parallel efficiency (green curve) over a 2,560-fold increase in problem size, from 4,160 cores to more than 10.6 million cores of the full system. The largest problem has 21.5 trillion unknowns. The performance reached 50,579 TFlops in double precision, which is approximately 41.3% of the theoretical peak performance.

A. Weak scaling

In the weak scaling runs, each subdomain was set to be $512 \times 512 \times 512$ and was handled by one core group. The baseline run employed 64 core groups with a $4 \times 4 \times 4$ domain decomposition. Owing to the overlap between the subdomains, the size of the physical grid was $2,036 \times 2,036 \times 2,036$. The largest configuration had $64 \times 64 \times 40$ subdomains, which resulted in a $32,516 \times 32,516 \times 20,324$ physical grid

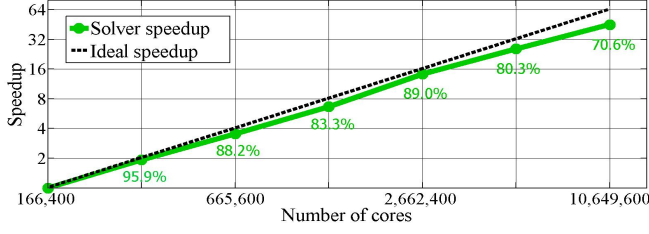


Fig. 11. Strong scalability results from 166,400 cores (2,560 core groups) to 10,649,600 cores (163,840 core groups). Numbers along the graph lines indicate efficiency with respect to the ideal speedup (the efficiency baseline is the 166,400 cores result).

with approximately 21.5 trillion grid points. The size of the physical problem increased 2,546-fold, and the size of the computational problem 2,560-fold. On account of the short time window for accessing the full system, 50 time steps were performed in the largest run, and no output was written. We measured the wall time elapsed for 30 consecutive time steps to calculate the performance. The results are shown in Table II. The parallel efficiencies for the weak scaling runs are shown in Figure 10. In the largest configuration, the size of the solution data was 160 TB, and the aggregate I/O bandwidth is 288 GB/s. Hence, output was expected to take 569 seconds for output. The average time consumed for a time step was approximately 6.09 s. The solution should be able to be outputted every few hundred time steps using the asynchronous I/O without interfering with the calculation, which is more than satisfactory for phase field simulations of this size.

TABLE II
WEAK SCALING RESULTS

number of core groups	performance in TFlops
$4 \times 4 \times 4$	20.2842
$8 \times 8 \times 8$	161.778
$16 \times 16 \times 16$	1,285.28
$32 \times 16 \times 16$	2,565.09
$32 \times 32 \times 16$	5,114.59
$32 \times 32 \times 32$	10,147.9
$64 \times 32 \times 32$	20,296.9
$64 \times 64 \times 32$	40,591.6
$64 \times 64 \times 40$	50,579.4

B. Strong scaling

In the strong scaling runs, the baseline run used 2,560 core groups with a $16 \times 16 \times 10$ domain decomposition. The computational grid size was $8,192 \times 8,192 \times 5,120$. Owing to the overlap between subdomains, the physical grid size varied slightly with the domain decomposition. For example, the initial domain decomposition is $16 \times 16 \times 10$ with a $512 \times 512 \times 512$ grid for each subdomain, and the physical grid size was $8,132 \times 8,132 \times 5,084$. When the domain decomposition was $16 \times 16 \times 20$, the subdomain grid became $512 \times 512 \times 256$ which resulted in a $8,132 \times 8,132 \times 5,044$ physical grid. We measured the elapsed wall time for 30 consecutive time steps to calculate the speedups. The results

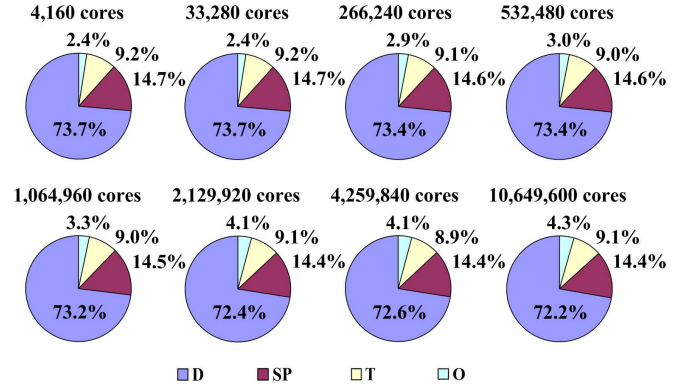


Fig. 12. Analysis of the kernel routines in the phase field solver of the weak scalability runs. The routines are divided into four categories, DGEMM for tensor dot products D, stencil and pointwise computing for inner boundary condition and nonlinear term SP, tensor transposition T, and other O which mainly involves MPI calls for inter-subdomain data transfer. The pie charts show fraction of time in each category.

are shown in Table III. Starting from 2,560 subdomains using 166,400 cores (2,560 core groups), The code achieved a 45-fold speedup on 10,649,600 cores (163,840 core groups), indicating 70.6% parallel efficiency in strong scaling. The parallel efficiencies for the strong scaling runs are shown in Figure 11.

C. Performance analysis

Figure 12 decomposes the overall runtime for each time step into its largest contributors in the weak scaling runs. The sub-routines are divided into four categories, the DGEMM (D) and tensor transposition (T) for the tensor dot product, the stencil/pointwise (SP) computing for computing nonlinear terms and inner boundary condition corrections to the exponential integrator, and other (O) which mainly involves MPI calls. The time for category O is calculated by taking the time consumed between the synchronization points of two consecutive time steps and subtracting off the time consumed by DGEMM, tensor transposition and SP computing subroutines, for which the differences among core groups can be neglected.

Scaling up from 64 subdomains to 163,840 subdomains, the overall minimum core group performance was in the range of 316.12 GFlops to 324.55 GFlops, which was approximately 41% to 42% of the peak performance. We can observe that the computation intensive DGEMM calls dominates with 72.2% to 73.7% of the overall runtime. Recall that there are two

TABLE III
STRONG SCALING RESULTS

number of core groups	time (30 time steps)	speedup
$16 \times 16 \times 10$	179.66s	1.00
$16 \times 16 \times 20$	93.633s	1.92
$32 \times 16 \times 20$	50.953s	3.53
$32 \times 32 \times 20$	26.961s	6.66
$32 \times 32 \times 40$	12.618s	14.24
$64 \times 32 \times 40$	6.9941s	25.69
$64 \times 64 \times 40$	3.9768s	45.18

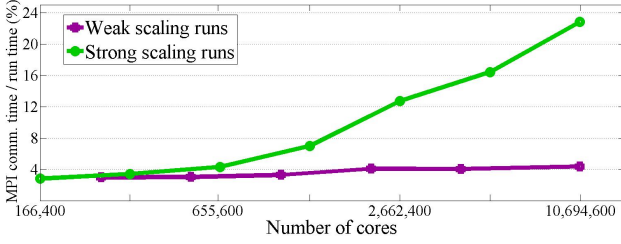


Fig. 13. MPI communication time relative to the overall runtime in the weak and strong scaling runs.

types of DGEMM calls in the code, as described in section III. One corresponds to contraction in the x and z directions, the other in the y direction. They steadily perform at 457.2 GFlops and 408.5 GFlops, which are 59.7% and 53.4% of the peak performance, respectively. These DGEMM calls contribute over 99% of the total computation. In terms of the tensor dot product, the contraction in the x direction achieved the highest performance at 457.2 GFlops. The contraction in the y direction performed at 408.5 GFlops on account of the smaller sizes of the matrices. The performance of contraction in the z direction was decreased to 344.1 GFlops by the tensor transposition.

The tensor transpositions are bandwidth bound operations. They employed approximately 9% of the runtime. The transposition subroutine works in two modes, one for forward-cycling of the indices (e.g. $x - y - z$ to $z - x - y$) and the other backward-cycling (e.g. $z - x - y$ to $x - y - z$). In the forward-cycling mode, the memory access pattern is loading chunks of data in the first dimension distributed along the third dimension and writing blocks of data in the first and second dimensions. The CPEs invoke DMA through C interface to access the main memory. The results of benchmark tests on the aggregate bandwidth of the corresponding vector DMA operations are presented in Table IV. We use the harmonic mean of the peak bandwidths,

$$\frac{2}{\frac{1}{dma_get_{peak}} + \frac{1}{dma_put_{peak}}} = 22.0 \text{ GB/s},$$

to predict the processing speed. The actual processing speed (calculated using the average execution time of the transposition subroutine) is 21.3 GB/s. Similar analysis is applied to the backward-cycling mode, in which the data loading and writing patterns are swapped. The predicted speed is 22.6 GB/s while the actual speed is 22.1 GB/s.

TABLE IV
AGGREGATE BANDWIDTH FOR VECTOR DMA OPERATIONS IN THE FORWARD-CYCLING MODE

message length	dma_get (GB/s)	dma_put (GB/s)
256B	22.2	21.8
512B	21.8	21.9
768B	20.3	20.7

The pointwise computation is integrated with a backward-cycling tensor transposition. Hence, their memory access pattern is the same, except that multi input variables are being handled simultaneously in pointwise computation. The peak dma_get bandwidth is 24.2 GB/s and dma_put 21.7 GB/s. The averaged actual processing speed of all pointwise computing subroutines is 22.1 GB/s. The stencil computation in the cETD subdomain solver is slightly different from those in finite difference schemes. Boundary values are used to calculate corrections to the nonlinear term [21]. The scanning process illustrated in Figure 5 is realized to perform stencil computation using the CPEs. The averaged processing speed is 1.19G stencils per second, corresponding to 19.0 GB/s effective processing speed. The degradation in the processing speed is mainly caused by the overlap between blocks (see Figure 5).

The MPI communication time relative to the overall runtime in the weak and strong scaling runs are summarized in Figure 13. Considering that the communication pattern in our application is similar to those of stencil computations and the subdomain solver is computationally intensive, it is expected that the data transfer would take a small portion of the total runtime. Indeed, we can clearly observe that the percentage of time spent in MPI communication remains very low in the weak scaling runs. This contributes to the nearly perfect scalability results presented in Figure 10. In the strong scaling runs, the granularity decreases from 2M grid points per core to 32K grid points per core. The relative MPI communication time gradually increases, as is expected.

VI. CONCLUSION

We designed a highly scalable, large time-step integrating algorithm for solving the highly nonlinear and severely stiff Cahn-Hilliard equations. The proposed algorithm was implemented on a leading class supercomputer. We demonstrated the capability of conducting phase field simulations for coarsening dynamics at extreme spatial and time scales. The overall performance of 50,579 TFlops (approximately 41.3% of the peak performance) was achieved in the weak scaling runs on the Sunway TaihuLight supercomputer. In the largest configuration, the physical grid had 21.5 trillion nodes. The code scaled up to 10.6 million cores with a 97.4% weak scalability over a 2,560-fold increase in parallelism. In the strong scaling sense, the parallel efficiency remained at 71%, over a 64-fold increase in parallelism, up to 10.6 million cores. In terms of the time-to-solution, the coarsening process of the two-component system (governed by the highly nonlinear and severely stiff Cahn-Hilliard equation with degenerate mobility) in the size of 300 billion grid points could be simulated at a speed of 0.133 s/step, where the step-size could be four orders of magnitude larger than the standard explicit Euler scheme. cETD is a stable, high-order-accurate, large time-step algorithm. By incorporating the inexact localized exponential integration and the subdomain coupling techniques, the proposed ScLET algorithm is greatly improved in the sense of scalability. Furthermore, its computation intensive nature

makes it a promising algorithm for a variety of architectures. Indeed, we have implemented it on clusters consisting of heterogeneous computing nodes equipped with Intel Xeon and Xeon Phi processors and achieved good performance as well as time-to-solution efficiency.

For applications, an important component of computational materials science—constituting fundamental issues and challenges of computational materials scientists—is the ability to effectively and quantitatively model and simulate microstructures and link them to material structures and properties. The large-scale 3D numerical simulations discussed here can infer the dependence of coarsening kinetics in a binary system on the system parameters such as volume fractions and diffusional mobilities. While the present study was focused on simulations of a scalar Cahn-Hilliard model, the algorithmic advancement and presented implementation strategy can be extended to more general phase field models that account for temperature dependence and elastic and electrostatic effects. By utilizing leading-class computational platforms and the latest algorithmic advances, the computational capability demonstrated herein enables analysis of complex morphological microstructure patterns on unprecedented time and spatial scales. It can be regarded as a welcome contribution to the materials science community, which has been embracing the deployment of many national and international initiatives (such as the Materials Genome Initiative in the US). Indeed, these initiatives are calling for enhanced computational capabilities for development of new materials. These capabilities are becoming top priorities for addressing the needs of new and demanding applications that are critical for continued improvements in our quality of life.

ACKNOWLEDGMENT

The authors would like to thank Jun Liu and Yang Gao from the Computer Network Information Center, Chinese Academy of Sciences for providing visualization tools to depict the phase field simulations. We also thank Professor Chao Yang from the Institute of Software, Chinese Academy of Sciences for providing us optimized DGEMM kernels for narrow matrix multiplication in a pre-release version of the xMath library and Dr. Xin Liu and Dr. Xiyang Wang for their valuable assistance in conducting large scale runs on the Sunway TaihuLight supercomputer.

Jian Zhang's research is supported in part by the National Science Foundation of China under grant number 11271350. Lili Ju's research is supported in part by the US National Science Foundation DMS-1521965. Qiang Du's research is supported in part by the US National Science Foundation DMS-1534910 and DMS-1558744. Xuebin Chi's research is supported in part by the National Science Foundation of China under grant number 91530324. Dongsheng Xu acknowledges the support by the national major R & D project (2016YFB0701304).

REFERENCES

- [1] L.-Q. Chen, "Phase-field models for microstructure evolution," *Annu. Rev. Mater. Res.*, vol. 32, pp. 113–140, 2002.

- [2] J. Cahn and J. Hillard, "Free energy of a nonuniform system i. interfacial free energy," *J. Chem. Phys.*, vol. 28, pp. 258–267, 1958.
- [3] S. Allen and J. W. Cahn, "A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening," *Acta Metall.*, vol. 27, pp. 1084–1095, 1979.
- [4] J. Cahn, C. Elliott, and A. Novick-Cohen, "The Cahn-Hilliard equation with a concentration-dependent mobility: motion by minus the Laplacian of the mean curvature," *European J. Appl. Math.*, vol. 7, pp. 287–301, 1996.
- [5] S. Dai and Q. Du, "Motion of interfaces governed by the Cahn-Hilliard equation with highly disparate diffusion mobility," *SIAM J. Appl. Math.*, vol. 72, pp. 1818–1841, 2012.
- [6] —, "Coarsening mechanism for systems governed by the Cahn-Hilliard equation with degenerate diffusion mobility," *Multi. Model. Simul.*, vol. 12, pp. 1870–1889, 2014.
- [7] R. Pego, "Front migration in the nonlinear Cahn-Hilliard equation," *Proc. Royal Soc. London A*, vol. 442, pp. 261–278, 1989.
- [8] A. Bray and C. Emmott, "Lifshitz-slyozov scaling for late-stage coarsening with an order parameter-dependent mobility," *Phys. Rev. B*, vol. 52, pp. R685–R688, 1995.
- [9] D. Fan, S. Chen, L.-Q. Chen, and P. Voorhees, "Phase-field simulation of 2-d Ostwald ripening in the high volume fraction regime," *Acta Materialia*, vol. 50, pp. 1895–1907, 2002.
- [10] H. Garcke, B. Niethammer, M. Rumpf, and U. Weikard, "Transient coarsening behaviour in the Cahn-Hilliard model," *Acta Materialia*, vol. 51, pp. 2823–2830, 2003.
- [11] J. Zhu, L.-C. Chen, J. Shen and V. Tikare, "Coarsening kinetics from a variable-mobility Cahn-Hilliard equation: Application of a semi-implicit fourier spectral method," *Phys. Rev. E*, vol. 60, pp. 3564–3572, 1999.
- [12] T. Kupper and N. Masbaum, "Simulation of particle growth and ostwald ripening via the Cahn-Hilliard equation," *Acta Metall. Mater.*, vol. 42, pp. 1847–1858, 1994.
- [13] T. Rogers and R. Desai, "Numerical study of late-stage coarsening for off-critical quenches in the Cahn-Hilliard equation of phase separation," *Phys. Rev. B*, vol. 39, pp. 11 956–11 964, 1989.
- [14] T. Rogers, K. Elder, and R. Desai, "Numerical study of the late stages of spinodal decomposition," *Phys. Rev. B*, vol. 37, pp. 9638–9649, 1988.
- [15] R. Kohn and F. Otto, "Upper bounds on coarsening rates," *Comm. Math. Phys.*, vol. 229, pp. 375–395, 2002.
- [16] H. Gómez, V. M. Calo, Y. Bazilevs, and T. J. Hughes, "Isogeometric analysis of the cahn-hilliard phase-field model," *Computer methods in applied mechanics and engineering*, vol. 197, no. 49, pp. 4333–4352, 2008.
- [17] X. Zheng, C. Yang, X.-C. Cai, and D. Keyes, "A parallel domain decomposition-based implicit method for the Cahn-Hilliard-Cook phase-field equation in 3d," *Journal of Computational Physics*, vol. 285, pp. 55–70, 2015.
- [18] M. Mamivand, M. A. Zaeem, and H. El Kadiri, "A review on phase field modeling of martensitic phase transformation," *Computational Materials Science*, vol. 77, pp. 304–311, 2013.
- [19] M. Militzer, M. Mecozzi, J. Sietsma, and S. Van der Zwaag, "Three-dimensional phase field modelling of the austenite-to-ferrite transformation," *Acta materialia*, vol. 54, no. 15, pp. 3961–3972, 2006.
- [20] L. Ju, J. Zhang, and Q. Du, "Fast and accurate algorithms for simulating coarsening dynamics of Cahn-Hilliard equations," *Computational Materials Science*, vol. 108, pp. 272–282, 2015.
- [21] L. Ju, J. Zhang, L. Zhu, and Q. Du, "Fast explicit integration factor methods for semilinear parabolic equations," *J. Sci. Comput.*, vol. 62, pp. 431–455, 2015.
- [22] T. Shimokawabe, T. Aoki, T. Takaki, A. Yamanaka, A. Nukada, T. Endo, N. Maruyama, and S. M. Clark, "Peta-scale phase-field simulation for dendritic solidification on the tsubame 2.0 supercomputer," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC '11)*. Article No. 3, 2011.
- [23] A. Yamanaka, T. Aoki, S. Ogawa, and T. Takaki, "Gpu-accelerated phase-field simulation of dendritic solidification in a binary alloy," *European J. Appl. Math.*, vol. 381(1), pp. 40–45, 2011.
- [24] T. Takaki, M. Ohno, T. Shimokawabe, and T. Aoki, "Two-dimensional phase-field simulations of dendrite competitive growth during the directional solidification of a binary alloy bicrystal," *Acta Materialia*, vol. 81(0), pp. 272–283, 2014.
- [25] T. Takaki, T. Shimokawabe, M. Ohno, A. Yamanaka, and T. Aoki, "Unexpected selection of growing dendrites by very-large-scale phase-field simulation," *Journal of Crystal Growth*, vol. 382, pp. 21–25, 2013.

- [26] M. Bauer, J. Hötzer, M. Jainta, P. Steinmetz, M. Berghoff, F. Schornbaum, C. Godenschwager, H. Köstler, B. Nestler, and U. Rüde, "Massively parallel phase-field simulations for ternary eutectic directional solidification," *Proceedings of 2015 International Conference for High Performance Computing, Networking, Storage and Analysis (SC '15)*, 2015.
- [27] M. Dorr, J.-L. Fattebert, M. Wickett, J. Belak, and P. Turchi, "A numerical algorithm for the solution of a phase-field model of polycrystalline materials," *Journal of Computational Physics*, vol. 229, pp. 626–641, 2010.
- [28] O. Wodoa and B. Ganapathysubramanian, "Computationally efficient solution to the cahn-hilliard equation: adaptive implicit time schemes, mesh sensitivity analysis and the 3d isoperimetric problem," *Journal of Computational Physics*, vol. 230, pp. 6037–6060, 2011.
- [29] J. Rudi, A. C. I. Malossi, T. Isaac, G. Stadler, M. Gurnisz, P. W. J. Staary, Y. Ineicheny, C. Bekasy, A. Curioniy, and O. Ghattas, "An extreme-scale implicit solver for complex pdes: Highly heterogeneous flow in earth's mantle," *Proceedings of 2015 International Conference for High Performance Computing, Networking, Storage and Analysis (SC '15)*, 2015.
- [30] M. Hochbruck, C. Lubich, and H. Selhofer, "Exponential integrators for large systems of differential equations," *SIAM J. Sci. Comput.*, vol. 19, pp. 1552–1574, 1998.
- [31] S. Cox and P. Matthews, "Exponential time differencing for stiff systems," *J. Comput. Phys.*, vol. 176, pp. 430–455, 2002.
- [32] Q. Du and W. Zhu, "Stability analysis and applications of the exponential time differencing schemes and their contour integration modifications," *J. Comput. Math.*, vol. 22, pp. 200–209, 2004.
- [33] S. Krogstad, "Generalized integrating factor methods for stiff pdes," *J. Comput. Phys.*, vol. 203, pp. 72–88, 2005.
- [34] Q. Du and W. Zhu, "Analysis and applications of the exponential time differencing schemes," *BIT Numer. Math.*, vol. 45, pp. 307–328, 2005.
- [35] Q. Nie, Y.-T. Zhang, and R. Zhao, "Efficient semi-implicit schemes for stiff systems," *J. Comput. Phys.*, vol. 214, pp. 521–537, 2006.
- [36] M. Hochbruck and A. Ostermann, "Exponential integrators," *Acta Numerica*, vol. 19, pp. 209–286, 2010.
- [37] C. Moler and C. V. Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM REVIEW*, vol. 45, no. 1, pp. 3–49, 2003.
- [38] L. Zhu, L. Ju, and W. Zhao, "Fast high-order compact exponential time differencing Runge-Kutta methods for second-order semilinear parabolic equations," *Journal of Scientific Computing*, vol. 67, no. 3, pp. 1043–1065, 2016.
- [39] Y. Saad, "Analysis of some krylov subspace approximations to the matrix exponential operator," *SIAM Journal of Numerical Analysis*, vol. 29, pp. 209–228, 1992.
- [40] M. Hochbruck and C. Lubich, "On krylov subspace approximations to the matrix exponential," *SIAM J. Numer. Anal.*, vol. 34, pp. 1911–1925, 1997.
- [41] R. Sidje, "Expokit: A software package for computing matrix exponentials," *ACM Trans. Math. Software*, vol. 24, pp. 130–156, 1998.
- [42] I. Moret and P. Novati, "An interpolatory approximation of the matrix exponential based on faber polynomials," *J. Comput. Appl. Math.*, vol. 131, pp. 361–380, 2001.
- [43] A. Martinez, L. Bergamaschi, M. Caliar, and M. Vianello, "A massively parallel exponential integrator for advection-diffusion models," *Journal of Computational and Applied Mathematics*, vol. 231, pp. 82–91, 2009.
- [44] F. Zheng, H. Li, and H. L. et al., "Cooperative computing techniques for a deeply fused and heterogeneous many-core processor architecture," *JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY*, vol. 30(1), pp. 145–162, 2015.
- [45] S. Browne, J. Dongarra, N. Garner, G. Ho, and P. Mucci, "A portable programming interface for performance evaluation on modern processors," *Int. J. High Perform. Comput. Appl.*, vol. 14, no. 3, pp. 189–204, 2000.