TensorKMC: Kinetic Monte Carlo Simulation of 50 Trillion Atoms Driven by Deep Learning on a New Generation of Sunway Supercomputer

Honghui Shang*
SKL of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
shanghonghui@ict.ac.cn

Xin Chen*
Institute of Applied Physics and Computational Mathematics
Beijing, China
chen_xin@iapcm.ac.cn

Xingyu Gao
Institute of Applied Physics and Computational Mathematics
Beijing, China
gao_xingyu@iapcm.ac.cn

Rongfen Lin†
Department of Computer Science and Technology, Tsinghua University
Wuxin, China
lrf21@mails.tsinghua.edu.cn

Lifang Wang
Institute of Applied Physics and Computational Mathematics
Beijing, China
lifang_wang@iapcm.ac.cn

Fang Li
National Supercomputer Center in Wuxi
Wuxi, China
38349735@qq.com

Qian Xiao
National Supercomputer Center in Wuxi
Wuxi, China
413476601@qq.com

Lei Xu
SKL of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
xulei19b@ict.ac.cn

Qiang Sun
National Supercomputer Center in Wuxi
Wuxi, China
18638602662@163.com

Leilei Zhu
SKL of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
zhull18@mail.ustc.edu.cn

Fei Wang
Department of Computer Science and Technology, Tsinghua University
Beijing, China
f-wang20@mails.tsinghua.edu.cn

Yunquan Zhang†
SKL of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
zyq@ict.ac.cn

Haifeng Song†
Institute of Applied Physics and Computational Mathematics
Beijing, China
song_haifeng@iapcm.ac.cn

ABSTRACT

The atomic kinetic Monte Carlo method plays an important role in multi-scale physical simulations because it bridges the micro and macro worlds. However, its accuracy is limited by empirical potentials. We therefore propose herein a triple-encoding algorithm and vacancy-cache mechanism to efficiently integrate *ab initio* neural network potentials (NNPs) with AKMC and implement them in our TensorKMC codes. We port our program to SW26010-pro and innovate a fast feature operator and a big fusion operator for the NNPs for fully utilizing the powerful heterogeneous computing units of the new-generation Sunway supercomputer. We further optimize memory usage. With these improvements, TensorKMC can simulate up to 54 trillions of atoms and achieve excellent strong and weak scaling performance up to 27,456,000 cores.

*Both authors contributed equally to this research.
†Corresponding authors

SC '21, November 14–19, 2021, St. Louis, MO, USA

ACM ISBN 978-1-4503-8442-1/21/11…$15.00
https://doi.org/10.1145/3458817.3476174

KEYWORDS

Kinetic Monte Carlo, Neural Network Potentials , Many-core processor, Scalability

ACM Reference Format:

Honghui Shang, Xin Chen, Xingyu Gao, Rongfen Lin, Lifang Wang, Fang Li, Qian Xiao, Lei Xu, Qiang Sun, Leilei Zhu, Fei Wang, Yunquan Zhang, and Haifeng Song. 2021. TensorKMC: Kinetic Monte Carlo Simulation of 50 Trillion Atoms Driven by Deep Learning on a New Generation of Sunway Supercomputer . In *The International Conference for High Performance Computing, Networking, Storage and Analysis (SC '21), November 14–19, 2021, St. Louis, MO, USA.* ACM, Denver, CO, USA, 13 pages. https://doi.org/10.1145/3458817.3476174

1 INTRODUCTION

Macroscopic failures of materials under operational conditions, such as cracking, wearing, and corrosion, usually originate from the evolution of microstructures at mesoscales, which is challenging to study experimentally. However, kinetic Monte Carlo (KMC), one of the most popular computational methods for mesoscale simulations, provides tools for understanding failure mechanisms and estimating material lifetimes[1]. By tracking microstructural evolution up to experimentally relevant temporal and spatial scales, KMC simulation sheds light on mechanisms of various kinetic processes, including irradiation, defect diffusion, cracking, and solute precipitation, in a wide variety of materials. [2–18].

Given its versatility, various KMC models have been developed, including objective kinetic Monte Carlo (OKMC), atomistic kinetic Monte Carlo (AKMC), and event kinetic Monte Carlo (EKMC). Of these models, the AKMC method, which maps hop events to changes in the states of lattice sites, provides feasibility for simulating systems at mesoscopic with atomic resolution. However, it remains problematic to widely apply the AKMC model to reproduce mesoscale phenomena because of the complex interaction and kinetics of materials. Two typical approaches are available to construct interaction and kinetics in AKMC models: the first approach is to establish microkinetic models before the AKMC simulations; this allows interaction and kinetic parameters to be employed as tabulates during simulations. Although this method could readily expand AKMC simulations to experimentally relevant mesoscales, physical reliability is limited as the microkinetic mechanism is simplified in models. The second approach calculates interaction and kinetic processes based on DFT or empirical interatomic potentials simultaneously during the AKMC simulations. Although this method enhances the fidelity of AKMC parameterization, the attainable spatial-temporal scales are reduced by the expensive overhead of *ab initio* calculations, and the applicability of empirical interatomic potentials would also limit the physical reliability. A physically reliable AKMC that attains experimentally relevant mesoscales is highly demanding.

Machine learning techniques, specifically artificial neural networks (ANN), have been applied to improve the physical parameterization of AKMC simulations [4–6]. These ANNs are designed to reproduce and predict thermodynamic and kinetic processes in materials. The incorporation with density functional theory (DFT) based neural network potentials (NNPs) is more and more common and important in various fields of science, as indicated by the award of the 2020 Gordon Bell Prize to molecular dynamics (MD) with NNP [19], which pushed the limit of molecular dynamics with *ab initio* accuracy to nanosecond-long simulations of 100 million atoms.

However, the main disadvantage of MD simulation is that the MD time step is typically on the order of 10^{-15} s, which limits the total duration of the simulation. This impedes reproducing the microstructural evolution of materials because many kinetic processes occur at mesoscales, which involve microsecond-long trajectories of trillions of atoms. Thus, to simulate larger timescales up to the formation of experimentally resolvable damage features such as point-defect precipitates, computational tools such as kinetic Monte Carlo (KMC) methods must be adopted [20, 21].

The application of NNPs to AKMC simulations allows for the transfer of electronic-scale properties to even larger scales than MD, as suggested in Ref. [22] and achieved in Ref. [23]. Very recently, an improved AKMC model implemented in OpenKMC[24] is used to investigate the dynamical evolution of vacancy defects in actinide alloys. In the present work, we incorporate NNPs trained with *ab initio* calculations to improve the fidelity of AKMC models in simulating the evolution of defects in chemically complex alloys. We efficiently couple two open-sourced programs, OpenKMC with TensorAlloy [25], to simulate the microstructural evolution of alloys in nuclear reactor pressure vessels (RPVs). In such simulations, the activation or migration energy of the vacancy is essential, but elementary properties at the atomic level are often not known experimentally. They can be obtained by the *ab initio* calculation software FHI-aims from the Fritz Haber Institute [26]. Such calculations have been performed to investigate the interaction between point defects and the Cu, Ni, Mn, and Si solute atoms. More importantly, to bridge the gap between the microscopic and macroscopic worlds, we must consider using advanced supercomputers to achieve a real-world mesoscopic simulation at the micron and microsecond levels and beyond.

However, several challenges stand between us and this goal: (1) NNPs were primarily developed for MD, and current implementation approaches may not be efficient for AKMC. (2) emory consumption may be too large upon increasing the size of the systems. (3) Heterogeneous computing units on the new Sunway supercomputer are powerful, but harness such power requires a specific design.

To overcome these challenges and achieve high efficiency, we propose several optimization strategies. Specifically, the following contributions are made:

- We proposed a triple-encoding algorithm and a vacancy-cache mechanism to efficiently integrate OpenKMC (AKMC) and TensorAlloy (NNPs). Then, TensorKMC was born. Memory usage was further optimized for mesoscale simulation.
- We developed a fast feature operator and a big-fusion operator to improve the computational efficiency by fully utilizing heterogeneous computing units.
- We achieved strong, near-linear scaling up to 24 960 000 cores and good weak scaling up to 27 456 000 cores with the new-generation of Sunway supercomputers. It enables large-scale, highly reliable AKMC simulations enhanced by NNPs, leading to an unprecedented microsecond evolution of vacancies interacting with 50 trillion alloying atoms.

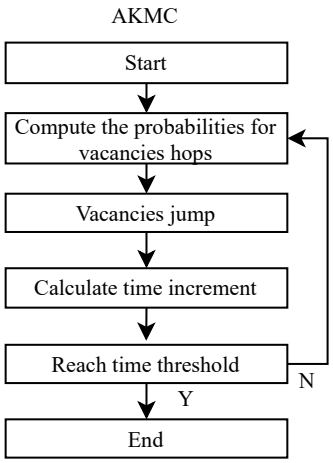


Figure 1: The flowchart for the serial AKMC framework.

2 BACKGROUND

2.1 Serial AKMC

The AKMC method can simulate the evolution of microstructures at the atomic scale. The first work to present kinetic Monte Carlo approach was published by Young and Elcock [27] for the kinetic of the vacancy diffusion. The basic assumption in this model is that vacancy diffuses via a series of first nearest neighbor (1NN) jumps. In the body-centred-cubic (bcc) lattice, the vacancy is exchanged with one of its eight nearest neighbor atoms. The flowchart of the AKMC method is described in Fig. 1.

First, the probabilities of the jumps from the current state of vacancies are computed. Here X refers to the index of the jump direction,

$$\Gamma^X = \Gamma_0 \cdot \exp(-\frac{E_a}{k_B T}) \ . \tag{1}$$

k_B is the Boltzmann's constant, T is the absolute temperature, Γ_0 is the attempt frequency which is set to 6×10^{12} s^{-1} and E_a represents the migration energy:

$$E_a = E_a^0 + \frac{1}{2} \cdot (E_f - E_i) \ . \tag{2}$$

The reference activation energy E_a^0 (Fe:0.65 eV, Cu:0.56 eV used in this work) is assumed to depend only on the chemical nature of the migrating atom that exchanges position with the vacancy. E_i and E_f are energies of the entire system before and after the vacancy hops [6, 28]. Second, the vacancy jumps randomly with a weight corresponding to its probability. Third, the residence time algorithm [27] is employed to calculate a time increment (Δt),

$$\Delta t = \frac{-\ln r}{\sum_{X=1,8} \Gamma_X} \tag{3}$$

which is proportional to the inverse of the sum of all possible event frequencies with a random number (r) according to its probability. Finally, the AKMC method repeats the steps above until the simulation time has reached a preset threshold.

2.2 Parallel AKMC method

The parallel AKMC algorithm is needed to carry out realistic computations over extended spatial and temporal scales. In the parallel AKMC algorithm, the spatial decomposition method is firstly performed to distribute the large systems into different MPI processes labeled with different colors as shown in Fig. 2(a). Then ghost regions (labeled with grey) are created to synchronize atomic position information on different processes with point-to-point commutation. Such domain decomposition method has been applied to MD simulations, where atoms move continuously. The positions of two atoms will not overlap after the MD evolution due to repulsive interatomic interaction, so there is no boundary conflict problem. However, in the AKMC method, atoms always stays at lattice sites, so the atoms belong to different MPI processes can move to an identical site, leading to boundary conflicts. To address this problem, Shim and Amar [29] proposed a synchronous sublattice algorithm to solve the boundary inconsistencies and avoid global communications. In this algorithm, the simulation domain of one process are further divided into eight sectors, and all processes execute events in sequence according to sector number on their own subdomain independently for a period of time. When moving to the computation of the next sector in a process, sites in the boundary region must be updated in advance. As shown in Fig. 2(b), in this way, the conflicting hops can be avoided.

2.3 HPC Platforms and Software Environment

The new-generation Sunway supercomputer is used for performance assessment in this work, which is the successor of the Sunway TaihuLight supercomputer. Like the Sunway TaihuLight system, the new Sunway supercomputer adopts a new-generation domestic high-performance heterogeneous many-core processors and interconnection network chips in China.

The new Sunway many-cores processor (SW26010-pro) is designed for massive thread and data parallelism and delivers high performance on parallel workloads. The architecture of the this new many-cores processor is shown in Fig. 3. Each processor contains 6 core-groups (CGs), with 65 cores in each CG, and in total 390 cores. Each CG has one management processing element (MPE), one cluster of computing processing elements (CPEs), and one memory controller (MC). The MPE within each CG is used for computations, management, and communication. The CPEs are organized as an 8×8 mesh (64 cores) and are designed to maximize the aggregated computing power and minimize the micro-architecture complexity. The CPEs are organized with a mesh network to achieve high-bandwidth data communication (P2P and collective communications) among the CPEs in one CG, which is called remote scratchpad memory access (RMA).

2.4 Current state of the art and Related Work and Challenges of Parallel TensorKMC

Even though capability of AKMC with efficient parallelization can reach mesoscales, fidelity of interatomic potentials employed in the models limits its physical reliability. However, the emergence of machine-learning (ML) based interatomic potentials trained with *ab initio* calculations provides an innovative way to construct AKMC

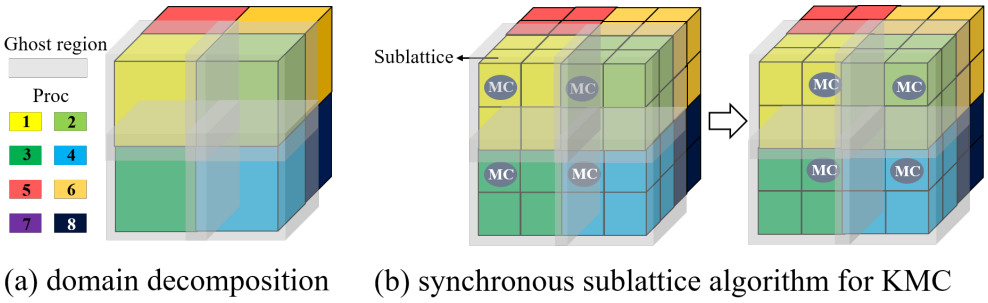


Figure 2: The parallel AKMC framework. (a) The spatial domain decomposition method to distribute the large systems into different MPI processes. (b) The synchronous sublattice algorithm to avoid conflicting hops near the shared boundary. The surrounding grey blocks represent the ghost region for synchronizing atomic positions of different processes.

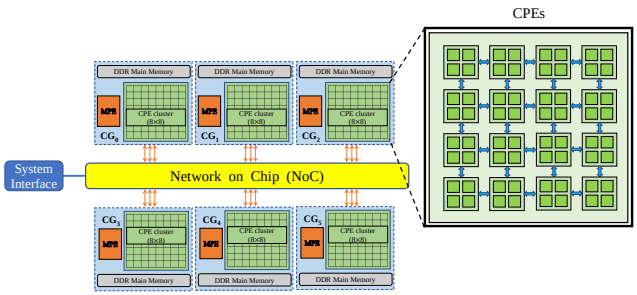


Figure 3: The architecture of the many-core SW26010-pro CPU.

model without loss of accuracy. The successful application of ML-based potential in MD simulation has proven its physical reliability and versatility [30–35], yet AKMC framework employing ML-based potentials is still very few. Recently, Castin et al. has developed a framework of a hybrid KMC model which is powered by an artificial neural network (ANN) [4, 6], and successfully applied it in simulations of Fe alloys. It shows the feasibility of coupling ML-based potential and AKMC models. However, this KMC framework only has the serial version, its space scale is limited to nanometers, far from the mesoscales. Up to now, the capabilities of AKMC models, with high physical reliability and expansibility in space scales, have never reached a spatial scale where trillions of atoms could be simulated. However, by employing a machine-learning interatomic potential with DFT accuracy and efficient implementation on the new-generation Sunway supercomputer, a new AKMC framework, TensorKMC, could simulate trillions of atoms, which truly realize the experiment relevant scales.

Previously, NNPs were mainly designed for MD simulations. In order to efficiently integrate NNPs with AKMC, several challenges and opportunities arise because AKMC has significant differences with MD:

- Atoms always locate on lattice sites in AKMC, while they do not have fixed locations in MD. Hence, interatomic distances in AKMC are discretely distributed.
- For AKMC simulations, energy is the driven power for evolution, while in MD runs, atomic forces (the first-order derivative of energy with respect to positions) play the central role.
- In AKMC, the atoms and vacancies jump directly between lattice sites, so the traditional parallel domain decomposition algorithm using in MD could execute in conflicting hops when the two or more processes perform events simultaneously near the shared boundary.
- A single AKMC step (a site jump) generally only affects a very small portion of atoms in the process. On the contrary, all atoms will be involved at each MD step.

AKMC has a much larger simulation box, but fewer atoms are invoked at each step, and the calculation is simpler. However, OpenKMC is developed with the principle of MD. OpenKMC stores properties and computes energies for all atoms in the domain. This strategy works for small systems with cheap energy routines (for example the extremely short-ranged pair potential or embedded-atom method (EAM)). But for accurate mesoscale simulation, severe problems occur:

- Storing essential properties for all atoms requires a vast amount of memory. OpenKMC is limited to approximately 11 million atoms per process for largescale simulation.
- Computing energies of all atoms in the domain are too expensive and unnecessary.
- CPEs are difficult to play a role because the energy calculation in OpenKMC requires a lot of direct memory access (DMA) operations on the *lattice* array. But this array is too large to place on local device memory (LDM) of CPEs. *OpenKMC gains no benefit from CPEs when the vacancy concentration has physical meaning.*

With several innovations and optimizations, TensorKMC successfully solves these challenges.

3 INNOVATION AND OPTIMIZATION

In this section we discuss innovations and optimizations in TensorKMC. Briefly, Sec. 3.1 and 3.2 are major AKMC innovations. Sec. 3.3 introduces our memory optimizaton strategies for mesoscale simulation. Sec. 3.4 and 3.5 describe details of innovated algorithms

for efficiently computing energies on CPEs. Sec. 3.6 discusses the portability of our innovations.

3.1 Triple-Encoding

In AKMC, vacancy plays a crucial role in dynamical evolution by causing random diffusion, even though its concentration is usually low. A basic kinetic process, a vacancy hop, only involves one vacancy and one of its first nearest neighbors. The probability of a vacancy transition depends on the energy difference resulting from the change of state, as introduced in Sec. 2.1. As interatomic interaction has a limited range (the cutoff radius r_{cut}), only sites near the vacancy are affected by a vacancy transition. The concentration of vacancy in real systems is typically low (1×10^{-2}% to 1×10^{-4}%), only a small portion of atoms should be invoked for computing the total energy of an AKMC system. Hence, a large simulation domain can be decomposed to discrete vacancy systems, as illustrated in Fig. 4a. Therefore, the problem arises: how to effectively and efficiently describe those vacancy systems.

In this work, we propose a triple-encoding tabulation algorithm (TET) to overcome this challenge. The body-centered cubic (bcc) Fe-Cu alloy system is used in this work. For bcc lattice, there are eight possible final states, and all lattice sites are geometrically equivalent. TET consists of three encoding tabulations: the relative coordinates encoding tabulation (CET), the neighbor list encoding tabulation (NET), and the vacancy encoding tabulation (VET).

CET is an ordered matrix describing IDs and relative coordinates of sites in a vacancy system (Fig. 4b). The center of this system, locating at (0,0,0), is a vacancy. Given a Euclidean cutoff radius r_{cut}, positions of its neighbors can be determined, and they are colored as green in Fig. 4a. Hence, a $(N_{local}, 3)$ matrix can be constructed. As all sites of bcc lattice are geometrically equivalent, these relative positions can be applied to the arbitrary site to fetch its neighbors within r_{cut}. In bcc lattice, a site has eight first nearest neighbors (1NN sites). Neighbors of these 1NN sites are all affected by a site jump. These N_{region} sites (red, green and light blue sites in Fig. 4a) together forms the jumping region. The N_{out} The outer deep blue sites in Fig. 4a) act as neighbors of sites in the region. Their energies will not change. Therefore, a vacancy system consists of total $N_{all} = N_{region} + N_{out}$ sites.

Based on CET, the three-dimensional array NET, describing neighbors (index and distance) of each site in the jumping region, can be constructed. NET describes neighbor relations of sites in the region. NET[i] is a $(N_{local}, 2)$ matrix storing neighbor IDs (the first column) and distances (the second column) of site i. CET and NET only depend on lattice constant a and cutoff radius r_{cut}. They are shared across all vacancy systems.

The last tabulation is VET, which encodes real environments of a vacancy system. VET is a vector of length N_{all}. VET[id] is the atom type of the site corresponding to site id. Assuming a vacancy locates at (i, j, k), coordinates of sites belonged to this vacancy system can be easily obtained by translating CET from $(0, 0, 0)$ to (i, j, k). Then IDs of these sites can be calculated (Section 3.3). Finally, atom types of these sites can be accessed from the *lattice* array.

The triple-encoding algorithm builds the foundation of TensorKMC. A huge simulation domain can be simplified to several small dense vacancy systems. Only during the initialization of a VET should we access the large *lattice* array. Thus, further optimizations, including significant memory reduction and distributing calculations on CPEs, become possible.

3.2 Vacancy Cache Mechanism

In OpenKMC, a cache all strategy is used. Per-atom properties used for computing energies are stored in memory and updated during evolution. For small and simple systems where the number of atoms is rather low and the interaction between atoms is localized, this strategy can perform very well as energy calculation can be extremely fast. When system scales are considerably large and interaction among atoms is long-range and highly complex, storing essential properties for all atoms is impossible.

Benefit from the domain decomposition introduced in Section 3.1, only vacancy systems are cached in TensorKMC. For each vacancy system, the corresponding VET vector and global IDs of its sites are cached. After a vacancy hop or a sublattice synchronization, euclidean distances between active site(s) and centers of vacancy systems in the simulation domain are calculated. Vacancy systems close to the active site(s) will be updated during the next propensity calculation.

3.3 Memory Usage

In OpenKMC, accessing properties of a lattice site with spatial coordinates (i, j, k), needs two steps: 1) get id from the POS_ID array with spatial coordinates; 2) get site property from a one-dimensional array with id. Fig. 5 demonstrates an example. For the target location (0, 0) in the ghost area of the two-dimensional simulation domain, POS_ID[0][0] should be visited first to get id = 2, then the atom type can be obtained by reading *lattice*[2] from main memory. The POS_ID array requires significant amount of memory. Getting id from the two or three-dimensional POS_ID array is also a memory-intensive operation.

In TensorKMC, a direct computation method is adopted. Each process marks a local ID for its own local and ghost lattice by traversing the cell. The order of the Lattice array is based on the local ID, which stores all the local grid points (red points) in the front of the array (shaded area in C), and the ghost area grid points (blue points) in the back (non-shaded area in C). Thus, to get the requisite index, we first calculate the number of ghost atoms nghost(x, y, z) whose local ID is less than the grid point.

$$\text{index} = \begin{cases} N + \text{nghost}(x, y, z), & \text{if } (x, y, z)\text{in ghost} \\ \text{ID}(x, y, z) - \text{nghost}(x, y, z) & \text{else} \end{cases} \tag{4}$$

where ID(x,y,z) is the local ID number calculated according to 3D coordinates, and N is the number of local lattice sites in the process.

Another significant memory reduction comes from the removal of per-atom property arrays (E_V, E_R, etc.). These arrays grow linearly with the simulation size. Because of the vacancy-cache mechanism, these arrays are no longer needed. CPEs can efficiently compute features of the atoms interested on-demand.

3.4 Fast Feature Operator

Until now, numerous choices of atomic feature descriptors have been proposed. In this work, we use an exponential-style feature

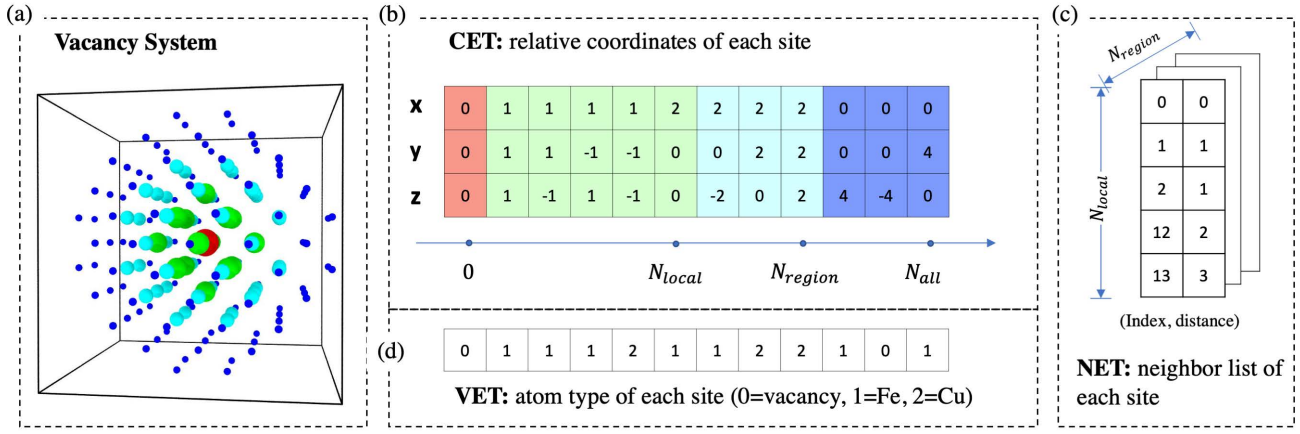


Figure 4: AKMC Domain decomposition and the triple-encoding tabulation algorithm innovated by TensorKMC. (a) is the visualization of a vacancy system with cutoff radius $r_{cut} = a$, where a is the lattice constant. (b) is the coordinates encoding tabulation (CET) describing the relative coordinates of each site in the vacancy system. (c) is the neighbor encoding tabulation (NET) storing neighbor list (Indices and distances) of each site. (d) is the vacancy encoding tabulation (VET) representing the atom type of each site.

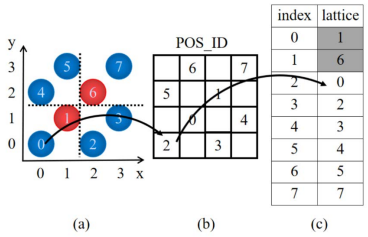


Figure 5: Schematic diagram of getting property of a lattice site in a two-dimensional simulation box (a) where red circles are local sites in this domain and blue circles represent ghost sites. (b) is the corresponding POS_ID array for the two-dimensional system. The blank grids represent wasted memory cells. (c) demonstrates the *lattice* array. Shaded area are atom types of local sites, others correspond to the ghost sites.

descriptor published by Oganov et al.:

$$f(r|p,q) = \sum_{j}^{N_{local}} \exp\{-\left(\frac{r}{p}\right)^{q}\} \tag{5}$$

where r is the interatomic distance, p and q are hyperparameters and $f(r|p,q)$ represents the corresponding feature value of r for (p,q). This feature algorithm has been proven useful for a variaty of metals. Assume N_{dim} is the total number of (p,q) sets and N_{el} is the number of unique elements in the system, each atom will be described by a $N_{dim} \times N_{el}$ feature vector. In AKMC atoms always locates on lattice sites, interatomic distances are discretely distributed. Hence, Equation 5 can be further simplified to a tabulated form:

$$f(r|p,q) = \sum_{j}^{N_{local}} \text{TABLE(r, p, q)} \tag{6}$$

where TABLE is a precomputed table for (r,p,q). Given the NET and a VET, *theoretically* one can easily compute features for all sites in a vacancy system with *for* loops. As a result, computing features in AKMC is changed to a memory-intensive task. To handle this challenge, we treat CPEs of each core group as *normal* processors and a micro parallel system is formed. Atomic features are calculated in parallel on CPEs:.

- The N_{region} sites in a vacancy system are assigned to CPEs circularly. For each CPE, the NET array, a copy of the VET vector, and the precomputed TABLE are stored in LDM.
- A total of $1 + N^f$ states will be performed on each CPE. The first loop computes the features for the initial state E_i. The following N^f loops compute features for N^f possible final states. For each final state k, the corresponding vacancy hop is simulated by swapping VET[0] with VET[k] and reorganizing neighbor lists.
- The generated features for $1 + N^f$ states are all kept in LDM until all done. These data are then sent back to the main memory with a direct memory access operation.

3.5 Big-Fusion Operator

In this work, multiple convolutional layers with 1x1 filters are used to build up atomistic neural network potentials [25, 36]. Each batch of the input corresponds to an AKMC state, and outputs are energies of the states.

According to the roofline model of new Sunway many-cores processor, the multi-layers operation is also memory-intensive. The upper limit of its performance is bounded by the memory-access

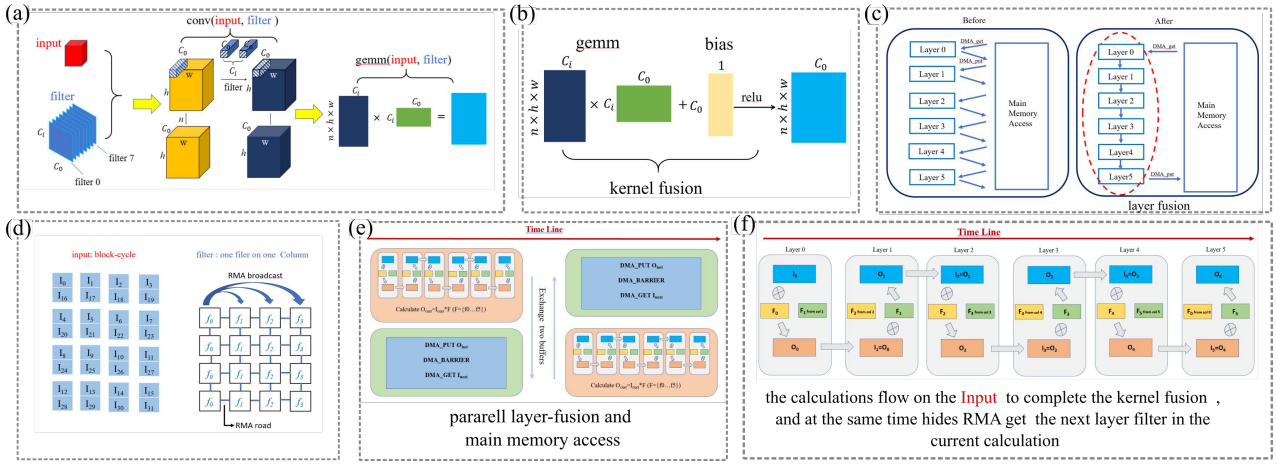


Figure 6: The big-fusion operator. (a) Convert convolution with 1x1 kernel to matrix multiplication. (b) Kernel fusion: matrix multiplication, bias and ReLU are fused to a single kernel. (c) Minimize the times of the main memory access. (d) Distribute the entire NNP model (all multi-layer fusion operators) across all CPEs. Each CPE shares a portion of model parameters through RMA. (e) DMA-based state flow: putting state $k-1$ back then fetching state $k+1$ in the same buffer through DMA and computing state k in sync (f) RMA based operator flow: the features are always kept in LDM until all operators passed through. Calculation of operator i and fetching of $i+1$ through RMA are executed in sync.

speed, which is not particularly fast (compared to the ultrahigh-speed HBM2 memory on Tesla GPUs). By analyzing the characteristics of the operations and the new Sunway many-cores processor, we build up a multi-level parallel fusion operator (Fig. 6) in which CPEs act as distributed computing and storing nodes. The ultimate goal is to minimize memory access and hide data exchange behind computation flows. This algorithm converts the block to a computation-intensive task.

Our strategy fully and cleverly utilizes features of Sunway many-core processor, such as software-controllable LDM, the asynchronous DMA, remote scratchpad memory access (RMA), single instruction multiple data (SIMD) and so on. Key innovations are as follows:

- Convert the convolution (1x1 kernel, stride 1) to the matrix multiplication, as shown in Fig. 6(a).
- Integrate the matrix multiplication with bias and ReLU activation into a fused operation acting as the basis of a calculation flow (Fig. 6(b)). This needs fewer memory operations. Furthermore, bias and ReLU can be done in the registers effectively through SIMD assembly.
- Distribute the entire model (filters and biases of all layers) across all CPEs by RMA, as shown in Fig. 6(d). In this work, each column of CPEs acts as one unit storing a layer fusion operation. Then they share operations to other CPEs of the same row through RMA upon requirement.
- Traditionally, to calculate a multi-layers neural network, each layer's input (output) data needs to get (put) from main memory. The overall speed is bottle-necked by the memory-access speed, which is not that fast on the new Sunway. To overcome this problem we develop a big-fusion strategy (Fig. 6(c)). Multiple fused layers are merged into one single kernel. Only two main memory accesses are required: 1) fetch input data of the first layer and 2) put back output data of the final layer.
- Fig. 6(e) demonstrates the DMA based state flow for computing energies of multiple states: putting state $k-1$ back then fetching state $k+1$ in the same buffer while computing state k in an other buffer.
- Algo. 1 and Fig. 6(f) shows the algorithm for computing energy of a state: features are always kept in LDM until all operators passed through. Calculation of operator i and fetching of $i+1$ through RMA are executed in sync. Operators flow through the features to complete the calculation of state k.

With the innovations listed above, the big-fusion operator can reach 76.64% of the peak (single-precision) at most. The algorithm is also scalable, and the current implementation (64 CPEs per MPE) can support up to eight layers of convolutional layers, which is large enough for materials modeling.

3.6 Portability

First, the domain decomposition strategy, the triple-encoding and vacancy cache algorithms are independent of architecture. They can be ported to other systems without obstacle. Second, porting

Algorithm 1 Big-fusion operator algorithm in TensorKMC.

$M \leftarrow n \times h \times w$
id : *index of cpes in one CG*
col_id : *column index of cpes in one CG*
$level$: *number of layers*
m_block : *the M − direct size of each cpe get once*
Init :
$i \leftarrow 0$
$DMA\_GET(x_{id})$ *and* $wait\_DMA\_GET$
$DMA\_GET(filter_{col\_id})$ *and* $wait\_DMA\_GET$
while $i < \frac{M}{64 \times m\_block}$ **do**
  $DMA\_GET(x_{i*64+id})$
  for all $j \leftarrow 0$, $level$ **do**
    $index \leftarrow (j+1)\ mod\ level$
    if $col\_id == index$ **then**
      $RMA\_ROW\_BCAST(filter_{index})$
    end if
    $y^j_{i+id} \leftarrow x^j_{i+id} \times f_j$
    $x^{j+1}_{i+id} \leftarrow y^j_{i+id}$
    $WAIT\_RMA\_ROW\_BCAST(filter_{index})$
  end for
  $DMA\_PUT(y_{id})$
  $DMA\_BARRIER(one\ buffer\ for\ y_{id}\ and\ x_{i*64+id})$
  $Wait\_DMA\_GET(x_{i*64+id})$
  $i \leftarrow i++$
end while
$Wait\_DMA\_PUT(y_{last})$

the feature operator and the big fusion operator to other many-core processors is also possible. In fact, a many-core processor can be viewed as a micro-parallel system. The specific mapping algorithm depends on the architecture. As an example, RMA is used to share the neural network potential parameters across CPEs on the new Sunway, while on Fugaku the large L2 share-cache (8 MB for 12 computing nodes) may be used to accomplish this task. Third, the motivation of our algorithms is to minimize data transfer so that arithmetic intensity can be significantly increased. This 'data centric' design principle can be adopted to any other architectures. Our work can be extended to other scientific problems, such as the helium bubble formation/diffusion in metals, or surface growth. Such investigations on other systems can further prove the power of our work but beyond the scope of this work.

4 EVALUATION

4.1 Validation

4.1.1 Potential. For the Fe-Cu alloy system, a cutoff radius of 6.5 Å is used. The N_{region}, N_{local} are 253 and 112, respectively. Total 32 different (p, q) sets are used, where p ranges from 4.2 to 1.1 with step -0.1 and q increases from 1.85 to 3.4 with step 0.05. The channels for the convolutional neural network are $(64, 128, 128, 128, 64, 1)$. ReLU activation function is used in our model. We generate a training dataset of 540 Fe-Cu structures, and the structure size ranges from 60 to 64. Energies are calculated using FHI-aims [26] with PBE exchange-correlation functional, *light numerous basis set* and 10x10x10 k-point grid. 400 structures are randomly selected as the training set. Our optimized potential can give a mean absolute error of 2.9 meV/atom (energy) and 0.04 eV/Å (force) on the test dataset.

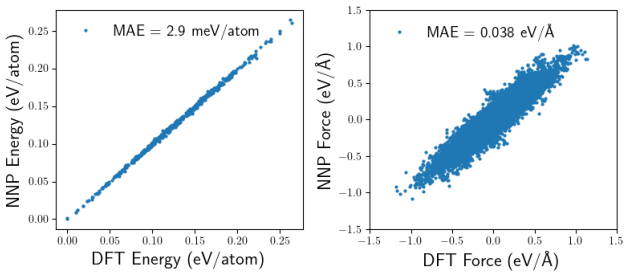


Figure 7: DFT and NNP predicted energies (eV/atom) and atomic forces (eV/Å. The corresponding R^2 scores are 0.998 and 0.880, respectively.

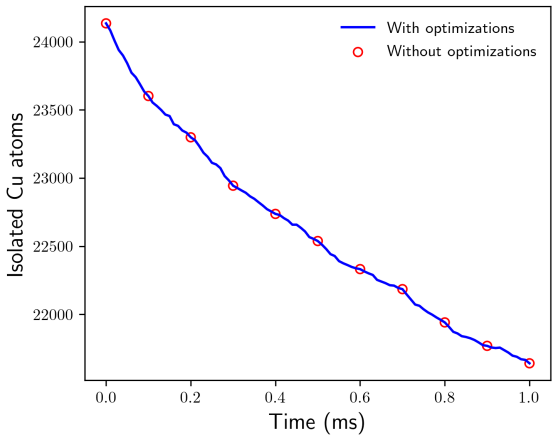


Figure 8: Validation of the domain decomposition and triple encoding.

4.1.2 Triple-encoding and vacancy cache. Fig. 8 validates the triple encoding and vacancy cache strategy by comparing the number of isolated Cu atoms. The domain size is $100 \times 100 \times 100a^3$ where $a = 2.87$ Å is the lattice constant. The simulation time is 1.0 ms. The concentrations of Cu and vacancy are 1.34 at. % and 0.0008%, respectively. Both runs give identical results, proving the correctness of our algorithms.

4.2 Bound Analysis and Performance Results

In this section, we analyze the performance and contribution of each optimization individually, discuss the coupling relations between these optimizations, and expose potentials for future optimizations.

Fig. 9 shows the Roofline model of energy kernels at the new Sunway. Upper panel of the table lists memory requirements, calculation volumes and intensities of the original fused operator (Conv2D + Bias + Relu) and the big-fusion operator. We use N, H, W of [32, 16, 16] as an example to show this algorithm. The operation-intensity of each layer increases from 0.48 to 21.3, which

Layer	In-c	Out-c	Memory (MB)	Volume (Gflop)	Intensity (flop/B)
1	64	128	10	0.125	12.8
2	128	128	12	0.25	21.3
3	128	128	12	0.25	21.3
4	128	128	12	0.25	21.3
5	128	64	8	0.25	21.3
6	64	1	2.06	0.001	0.48
Original			56.06	1.01	18.44
Big-fusion			2.03	1.01	509.05

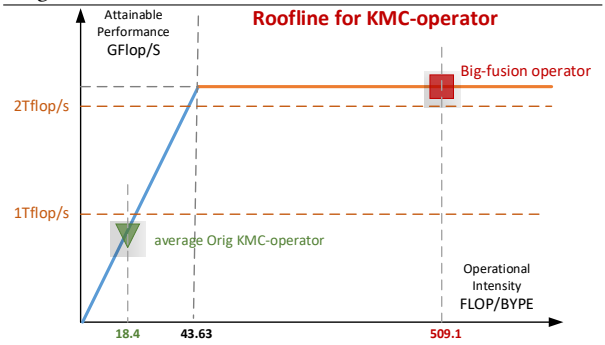


Figure 9: The Roofline analysis of the energy kernels at the new Sunway architecture. The Roofline analysis shows the interplay between computation capacity, memory bandwidth and data reuse. The arithmetic intensity of the big-fusion operator is 509.1 F (FLOPS/Byte). Therefore the attainable performance is much higher than the original version (each layer of the network is called separately).

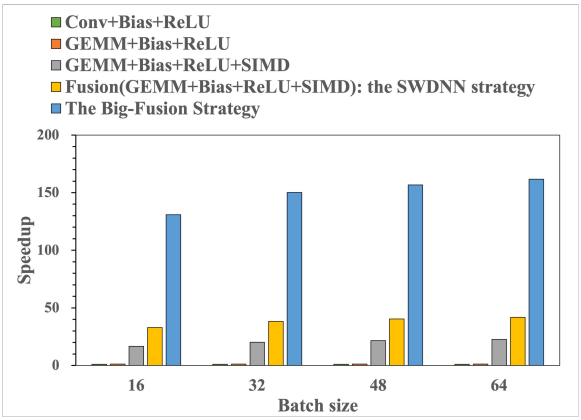
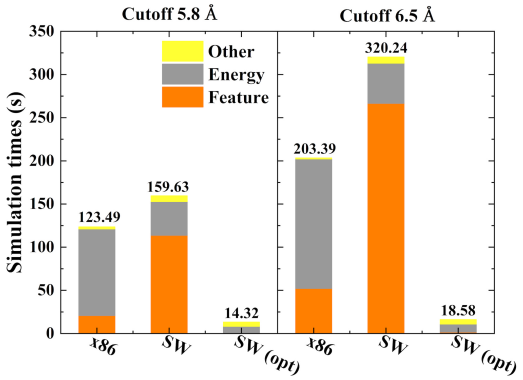


Figure 10: The performance comparison among different optimization steps for the TensorKMC-operator calculation.

is far less than the 43.63 of the machine, and the computing performance is limited by the access bandwidth of the system. Our big-fusion operator reduces the total memory access from 56 MB to 2 MB, which is equal to the memory size of the last layer. The arithmetic intensity is increased up to 509.1 (FLOPS/Byte). As a result, energy calculation becomes compuation-bounded but not memeory-bounded.



	x86	SW	SW(opt)
Feature	CPU	MPE	Fast Feature Op CPE
Energy	TensorFlow CPU FusedConv2D	TensorFlow SWDNN CPE FusedConv2D	Big-Fusion Op CPE

Figure 11: Serial performances of TensorKMC with different settings. Here x86 refers to AMD Ryzen EPYC.

Then we analyze the performance improvements for different levels of optimization methods. Fig. 10 gives detailed comparisons of adopted optimizations. First, the convolution is converted to matrix multiplication, increasing the performance from 1.0x to 1.23x. Second, applying SIMD vectorization improves the performance to 16x~22x. Third, the (Conv2D, Bias, ReLU) fusion (yellow bar) speedups up around 33x to 41x. Finally, the big-fusion strategy (blue bar in Fig. 10), which significantly reduces main memory access, are applied and we achieve a performance speedup around 131x to 161x compared with the base version.

4.3 Serial Comparisons

Serial performances of TensorKMC on x86 and the new Sunway are shown in Fig. 11. The benchmark simulates 1×10^{-7} seconds with 128 millions of atoms. Here:

- **x86** refers to AMD Ryzen EPYC with libtensorflow_cc. Atomic features are calculated sequentially. Energies are calculated with FusedConv2D operators (TensorFlow automatically converts Conv2D + Bias + ReLU to FusedConv2D).
- **SW** refers to the new Sunway with libtensorflow_cc and Sunway Deep Neural Network library (SWDNN). Atomic features are calculated on MPE. Energies are calculated with FusedConv2D operators on CPEs. The yellow bar in Fig 10 represents the performance of this fusion operator.
- **SW(opt)** refers to the new Sunway with customized operators innovated in this work. Atomic features are calculated in parallel on CPEs. Energies are calculated with the big-fusion operator.

The standard cutoff radius 6.5Åand a shorter 5.8 Å are used for comparison. Shorter r_{cut} indicates lesser atoms in a vacancy system and less computation and memory operations in total.

Table 1: Memory statistics for OpenKMC and TensorKMC on a single node in terms of the simulation box. The entry *Runtime* denotes the overall memory consumption during AKMC iterations. For OpenKMC, T and POS_ID are arrays for accessing *lattice* array while E_V and E_R are arrays for computing atomic energies with the EAM potential.

Millions of atoms		2	16	54	128
OpenKMC	T	68	515	1709	4014
	POS_ID	34	258	856	2009
	E_V	68	515	1709	4014
	E_R	68	515	1709	4014
	Runtime	467	3038	9964	-
TensorKMC	VAC Cache	0.09	1.50	2.53	6.00
	Runtime	133	1021	3594	8120

4.3.1 Feature. Computing atomic features of each state is a purely memory-intensive task. Bottle-necked by the main memory speed, the serial version is ~5 times slower on the new Sunway, compared with AMD EPYC. With our innovation in Sec 3.4, feature calculations can run in parallel on CPEs efficiently. As a result, the optimized version is ~60 times faster than the serial version on the new Sunway, or ~14 times faster than EPYC.

4.3.2 Energy. CPEs are very powerful vectorized computing units. Energy calculation on new Sunway is ~3 times faster than EPYC, even though there are a lot of main-memory accesses. With our big fusion algorithm, main memory accesses are successfully hidden behind computation. Thus, the time cost of computing energies is amazingly further reduced by almost 80% (or ~15 times faster than EPYC).

4.3.3 Overall speed. For the overall performance, TensorKMC with our fully optimized operators is no doubt the fastest version. It is approximately 11 times faster than the TensorFlow based version on EPYC and 17 times faster than the TensorFlow/SWDNN based version on the new Sunway.

4.3.4 Memory. Table 1 summarizes memory usages of OpenKMC and TensorKMC for different simulation settings. *Runtime* represents the real memory consumption during AKMC iterations. OpenKMC cannot simulate 128 million atoms with a single process. TensorKMC only needs ~ 1/3 memory of OpenKMC. The huge reduction mostly comes from two parts: 1) T and POS_ID arrays are removed (Sec 3.3) and the per-atom property arrays, E_V and E_R, are no longer needed. E_V, and E_R arrays can be viewed as the atomic features for the EAM potential, as the atomic energy E[i] can be directly calculated from them:

$$E(i) = \frac{1}{2}\text{E\_V[i]} + F_\rho(\text{E\_R[i]}) \tag{7}$$

where E_V[i] is the pair contribution, E_R[i] is the electron density and $F_\rho(\cdot)$ is a tabulated embedding function. These arrays are essential for OpenKMC but no longer needed for TensorKMC because of the vacancy cache mechanism.

In conclusion, both the speed and memory evaluations prove the advantages of our innovations adopted in TensorKMC.

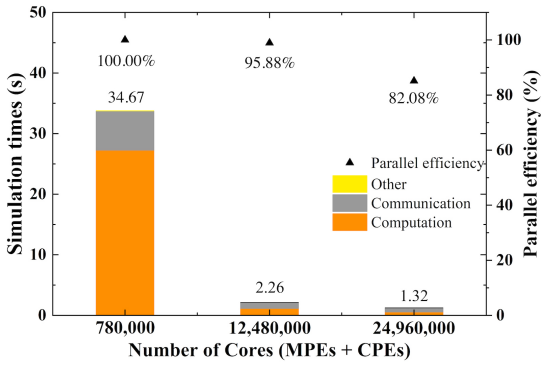


Figure 12: The strong scalability for the computation time of the 1.92 trillions of atoms. Simulation time and parallel efficiency values are annotated on the top of bars and in the parentheses respectively.

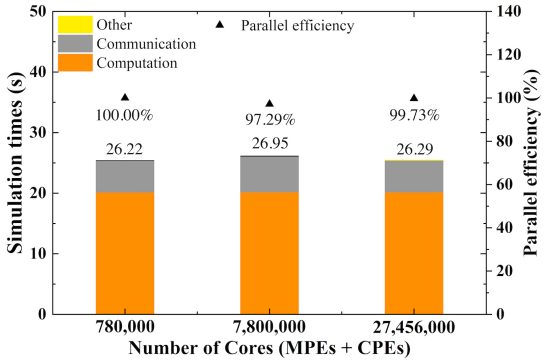


Figure 13: The weak scalability for the computation time up to the 54.067 trillions of atoms. Simulation time and parallel efficiency values are annotated on the top of bars and in the parentheses respectively.

4.4 Scalability Results

Scalability of the optimized TensorKMC on the New Sunway supercomputer for large-scale simulations is exhibited in this section. Mesoscale systems, where the number of atoms ranges from 1.536 to 54.067 trillion, are employed in the scalability tests, which are not available with the baseline implementation and MD schemes. In all scalability tests, we use the tree strategy for propensity update and a very strict synchronization interval $t_{stop} = 2 \times 10^{-8}$ even if the vacancy and Cu concentrations are relatively low. In practical simulations, one can adjust this variable to some larger values to significantly reduce communication between processes.

4.4.1 Strong Scaling. Strong scaling scalability of the optimized TensorKMC is measured with the simulation duration of 1×10^{-7} seconds ranging from 780,000 to 24,960,000 cores. A defective Fe-Cu alloy system is tested with thermal aging at 573 K. To achieve a balance between simulation size and the available memory (maximum

16GB per CG), we take the performance of 780,000 cores (12,000 CGs) as the baseline. For the baseline, each CG simulate 160 million atoms. Per-atom memory cost is reduced from 0.70 kB (OpenKMC) to 0.10 kB. The entire simulation system contains 1.92 trillion atoms, including 25.728 billion Cu atoms (1.34 at.%) and 15.36 million vacancies (8×10^{-4} at.%). Fig. 12 shows the performance of strong scalability tests in this case. Even with the strict synchronization strategy, TensorKMC shows good strong scalability in all test cases. The parallel efficiency is still 85% when scaling to 24,960,000 cores (384,000 CGs) on the new Sunway supercomputer.

4.4.2 Weak Scaling. Weak scalability of the optimized TensorKMC is also measured with a 1×10^{-7} second KMC simulation for the defective Fe-Cu systems with thermal aging 573 K, as shown in Fig. 13. In the weak scalability tests, each CG simulates 128 million atoms. The system exhibits excellent scaling with respect to the number of CGs. Among the testing systems, the largest size is 54.067 trillion atoms (422,400 CGs, 27,456,000 cores), which is more than two orders of magnitude larger compared with OpenKMC, reaching a true experimentally relevant time-spatial scale. The great weak scalability of the optimized TensorKMC indicates that the software is able to simulate much larger material or chemical systems on future exascale supercomputers with no intrinsic obstacles.

5 APPLICATIONS

Solute precipitation is one of the most notorious degeneration phenomena in metallic alloys, as they cause hardening embrittlement by impeding the movement of dislocations [37–40]. Specifically, in reactor pressure vessels, a key nuclear power plant component, the formation of the Cu cluster precipitation induced by neutron irradiation in Fe-Cu alloys could increase embrittlement and susceptibility to fracture over time. It limits the lifetime of reactor pressure vessels, which are prohibitively expensive to replace in commercial light water reactor nuclear power plants [38]. Therefore, it is imperative to have an explicit understanding of the kinetic mechanism of Cu cluster precipitation, including the Cu diffusing process, nucleation of Cu clusters, and long-term dynamical evolution of Cu precipitation. These processes usually happen and evolve at mesoscale, where millions of atoms and tens of microseconds are requested for a simulation. It is far beyond the capability of direct molecular dynamics based on ab initio calculations or empirical atomistic potentials. However, the Tensor KMC model, which evolves a system by sampling a sequence of energetically activated events derived from a machine learning force field, could extend the space-time scale up to larger orders while retaining full atomistic detail. It makes the Tensor KMC an ideal model to track the dynamical process of Cu cluster precipitation in Fe-Cu alloys. For the first time, micro-seconds evolution of defective FeCr system with nano-size could be simulated and displayed.

We show in Fig. 14, the Cu precipitation and void formation after one-second evolution in a 250,000,000 atom Fe-Cu alloy by TensorKMC simulation. The cell size is set to $500 \times 500 \times 500 a^3$, where a is the lattice constant. Typical reactor operating temperature, 573K, and typical Cu alloying concentration, 1.34 at.%, are applied to the simulation. As seen in Fig. 14, after a long-term evolution, considerable Cu cluster precipitations are observed while isolated Cu atoms are significantly reduced. By careful cluster analysis, the

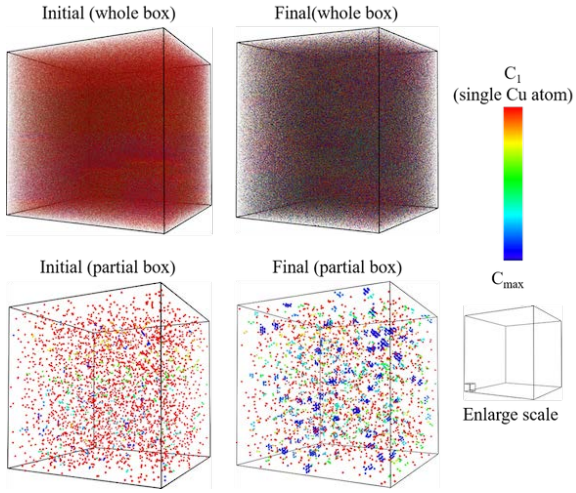


Figure 14: Cu precipitation is reproduced by TensorKMC simulation. To show details of the precipitation, simulation box, labeled as "whole box", is zoomed in at 10x scale, which is labeled as "partial box". Cu clusters are colored by size, where C_1 denotes Cu cluster with minimum size, i.e. single Cu atom, while C_max denotes Cu cluster with maximum size. Appearance of large blue Cu clusters indicates its precipitation.

max size of Cu precipitation is found around 40, and the number density of Cu cluster gradually stabilized at about $1.71 \times 10^{26}/m^3$. These results shed light on the mesoscale characters of defective Fe-Cu alloys, which is crucial for a reliable assessment of in-service safety nuclear power plant components.

Given the capacity of extending spatio-temporal scales to mesoscopic by new Sunway heterogeneous computing units and the accuracy of calculating atomic interaction by a neural network potential inherited from *ab initio* calculations, TensorKMC is able to simulate not only solute precipitation process in alloys, but also other non-equilibrium and kinetic phenomena, such as defect accumulation and diffusion [41–45], grain growth [46–51], surface morphology evolution [52–56], dislocation dynamics [57–63], and so on. It is for the first time that mesoscale simulations could be achieved with atomistic resolution. With a wide range of application scenarios, including material prediction, physical chemistry research, and biological engineering, practical multi-scale modeling, where micro-scale calculations and macro-scale numerical simulations are connected by reliable mesoscale modeling, is about to come true.

6 CONCLUSION

In the upcoming exascale machines, the innovations realized in TensorKMC will make it suitable for mesoscale simulation of the atomic system on the future exascale machines. To the best of our knowledge, this is first reported that the micron-long kinetic simulation of 50 trillion atoms with ab initio accuracy is achieved over twenty million cores, which open new perspectives for predicting

and investigating the microstructure evolution at experimental resolution. The optimization methods proposed in this work could be extended to other KMC codes with the same computational characteristics; hence, they will be broadly beneficial to the quantum chemistry, biological, and material science communities.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China under Grant No. 2016YFB0201203, the Science Challenge Project under Grant No. TZZT2019, the State Key Laboratory of Computer Architecture Foundation under Grant No. CARCH 4205, CARCH 4411, the Science Challenge Project under Grant No. TZ2018002 and the National Natural Science Foundation of China under Grant No. 12004046.

REFERENCES

[1] Steve Plimpton, Corbett Battaile, Mike Chandross, Liz Holm, Aidan Thompson, Veena Tikare, Greg Wagner, E Webb, X Zhou, C Garcia Cardona, et al. Crossing the mesoscale no-man's land via parallel kinetic monte carlo. *Sandia Report SAND2009-6226*, 2009.

[2] Eugene Ustinov. Kinetic Monte Carlo approach for molecular modeling of adsorption. *Current Opinion in Chemical Engineering*, 24:1–11, 2019.

[3] K. Nordlund. Historical review of computer simulation of radiation effects in materials. *Journal of Nuclear Materials*, 520:273–295, 2019.

[4] N Castin, MI Pascuet, Luca Messina, C Domain, Pär Olsson, RC Pasianot, and L Malerba. Advanced atomistic models for radiation damage in fe-based alloys: Contributions and future perspectives from artificial neural networks. *Computational Materials Science*, 148:116–130, 2018.

[5] Ignacio Martin-Bragado, Ricardo Borges, Juan Pablo Balbuena, and Martin Jaraiz. Kinetic Monte Carlo simulation for semiconductor processing: A review. *Progress in Materials Science*, 92:1–32, 2018.

[6] N Castin, L Messina, C Domain, RC Pasianot, and Pär Olsson. Improved atomistic monte carlo models based on ab-initio-trained neural networks: Application to fecu and fecr alloys. *Physical Review B*, 95(21):214117, 2017.

[7] N J van der Kaap and L Jan Anton Koster. Massively parallel kinetic Monte Carlo simulations of charge carrier transport in organic semiconductors. *Journal of Computational Physics*, 307:321–332, 2016.

[8] Da Jiang Liu, Andres Garcia, Jing Wang, David M. Ackerman, Chi Jen Wang, and James W. Evans. Kinetic Monte Carlo Simulation of Statistical Mechanical Models and Coarse-Grained Mesoscale Descriptions of Catalytic Reaction-Diffusion Processes: 1D Nanoporous and 2D Surface Systems. *Chemical Reviews*, 115(12):5979–6050, 2015.

[9] H. M. Cuppen, L. J. Karssemeijer, and T. Lamberts. The kinetic Monte Carlo method as a way to solve the master equation for interstellar grain chemistry. *Chemical Reviews*, 113(12):8840–8871, 2013.

[10] Michail Stamatakis and Dionisios G. Vlachos. Unraveling the complexity of catalytic reactions via kinetic monte carlo simulation: Current status and frontiers. *ACS Catalysis*, 2(12):2648–2663, 2012.

[11] J. Pezoldt, D. V. Kulikov, V. S. Kharlamov, M. N. Lubov, and Yu V. Trushin. Multiscale simulation of nucleation and growth of nanoscale SiC on Si. *Journal of Computational and Theoretical Nanoscience*, 9(11):1941–1966, 2012.

[12] C. S. Becquart and C. Domain. Introducing chemistry in atomistic kinetic Monte Carlo simulations of Fe alloys under irradiation. *Phys. status solidi*, 247(1):9–22, jan 2010.

[13] Charlotte S Becquart, Christophe Domain, Utpal Sarkar, Andrée Debacker, and Marc Hou. Microstructural evolution of irradiated tungsten: Ab initio parameterisation of an OKMC model. *Journal of nuclear materials*, 403(1-3):75–88, 2010.

[14] F. Soisson, C.S. Becquart, N. Castin, C. Domain, L. Malerba, and E. Vincent. Atomistic Kinetic Monte Carlo studies of microchemical evolutions driven by diffusion processes under irradiation. *J. Nucl. Mater.*, 406(1):55–67, nov 2010.

[15] E. Vincent, C. S. Becquart, C. Pareige, P. Pareige, and C. Domain. Precipitation of the FeCu system: A critical review of atomic kinetic Monte Carlo simulations. *J. Nucl. Mater.*, 373(1-3):387–401, 2008.

[16] Alexander Slepoy, Aidan P Thompson, and Steven J Plimpton. A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks. *The journal of chemical physics*, 128(20):05B618, 2008.

[17] Abhijit Chatterjee and Dionisios G. Vlachos. An overview of spatial microscopic and accelerated kinetic Monte Carlo methods. *Journal of Computer-Aided Materials Design*, 14(2):253–308, 2007.

[18] Corbett C. Battaile and David J. Srolovitz. Kinetic Monte Carlo simulation of chemical vapor deposition. *Annual Review of Materials Science*, 32:297–319, 2002.

[19] W. Jia, H. Wang, M. Chen, D. Lu, J. Liu, L. Lin, R. Car, . Weinan, E., and L. Zhang. Pushing the limit of molecular dynamics with ab initio accuracy to 100 million atoms with machine learning. 2020.

[20] Mikael Leetmaa and Natalia V Skorodumova. Kmclib: A general framework for lattice kinetic monte carlo (kmc) simulations. *Computer Physics Communications*, 185(9):2340–2349, 2014.

[21] JA Blackman and PA Mulheran. Growth of clusters on surfaces: Monte carlo simulations and scaling properties. *Computer physics communications*, 137(1):195–205, 2001.

[22] J. Behler. Representing potential energy surfaces by high-dimensional neural network potentials. *Journal of Physics Condensed Matter*, 26(18), 2014.

[23] Nongnuch Artrith and Alexie M. Kolpak. Understanding the composition and activity of electrocatalytic nanoalloys in aqueous solvents: A combination of DFT and accurate neural network potentials. *Nano Letters*, 14(5):2670–2676, 2014.

[24] Kun Li, Honghui Shang, Yunquan Zhang, Shigang Li, Baodong Wu, Dong Wang, Libo Zhang, Fang Li, Dexun Chen, and Zhiqiang Wei. OpenKMC: a KMC design for hundred-billion-atom simulation using millions of cores on Sunway Taihulight. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, page 68. ACM, 2019.

[25] Xin Chen, Xing Yu Gao, Ya Fan Zhao, De Ye Lin, Wei Dong Chu, and Hai Feng Song. TensorAlloy: An automatic atomistic neural network program for alloys. *Computer Physics Communications*, 250:107057, 2020.

[26] Volker Blum, Ralf Gehrke, Felix Hanke, Paula Havu, Ville Havu, Xinguo Ren, Karsten Reuter, and Matthias Scheffler. Ab initio molecular simulations with numeric atom-centered orbitals. *Comput. Phys. Commun.*, 180(11):2175–2196, November 2009.

[27] W M Young and E W Elcock. Monte carlo studies of vacancy migration in binary ordered alloys: I. *Proceedings of the Physical Society*, 89(3):735–746, nov 1966.

[28] Pär Olsson, TPC Klaver, and C Domain. Ab initio study of solute transition-metal interactions with point defects in bcc fe. *Physical Review B*, 81(5):054102, 2010.

[29] Yunsic Shim and Jacques G. Amar. Semirigorous synchronous sublattice algorithm for parallel kinetic monte carlo simulations of thin film growth. *Phys. Rev. B*, 71:125432, Mar 2005.

[30] Jonathan Schmidt, Mario R. G. Marques, Silvana Botti, and Miguel A. L. Marques. Recent advances and applications of machine learning in solid-state materials science. *NPJ COMPUTATIONAL MATERIALS*, 5, AUG 8 2019.

[31] Volker L. Deringer and Gabor Csanyi. Machine learning based interatomic potential for amorphous carbon. *PHYSICAL REVIEW B*, 95(9), MAR 3 2017.

[32] Rampi Ramprasad, Rohit Batra, Ghanshyam Pilania, Arun Mannodi-Kanakkithodi, and Chiho Kim. Machine learning in materials informatics: recent applications and prospects. *NPJ COMPUTATIONAL MATERIALS*, 3, DEC 13 2017.

[33] Joerg Behler. Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations. *PHYSICAL CHEMISTRY CHEMICAL PHYSICS*, 13(40):17930–17955, 2011.

[34] Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS CENTRAL SCIENCE*, 4(1):120–131, JAN 24 2018.

[35] Albert P. Bartok, Sandip De, Carl Poelking, Noam Bernstein, James R. Kermode, Gabor Csanyi, and Michele Ceriotti. Machine learning unifies the modeling of materials and molecules. *SCIENCE ADVANCES*, 3(12), DEC 2017.

[36] X. Chen, M. S. Jorgensen, J. Li, and B. Hammer. Atomic energies from a convolutional neural network. *J. Chem. Theory Comput.*, 14(7):3933–3942, 2018.

[37] Alan J. Ardell and Pascal Bellon. Radiation-induced solute segregation in metallic alloys. *Current Opinion in Solid State and Materials Science*, 20(3):115–139, 2016.

[38] J. E. Zelenty. Understanding thermally induced embrittlement in low copper RPV steels utilising atom probe tomography. *Materials Science and Technology (United Kingdom)*, 31(8):981–988, 2015.

[39] F. Carre, C. Renault, P. Anzieu, P. Brossard, and P. Yvon. *Outlook on generation IV nuclear systems and related materials RD challenges.* 2008.

[40] Sigmund J. Andersen, Calin D. Marioara, Jesper Friis, Sigurd Wenner, and Randi Holmestad. Precipitates in aluminium alloys. *Advances in Physics: X*, 3(1):790–814, 2018.

[41] Xian-Ming Bai, Arthur F. Voter, Richard G. Hoagland, Michael Nastasi, and Blas P. Uberuaga. Efficient Annealing of Radiation Damage Near Grain Boundaries via Interstitial Emission. *SCIENCE*, 327(5973):1631–1634, MAR 26 2010.

[42] Nicholas Aristidou, Christopher Eames, Irene Sanchez-Molina, Xiangnan Bu, Jan Kosco, M. Saiful Islam, and Saif A. Haque. Fast oxygen diffusion and iodide defects mediate oxygen-induced degradation of perovskite solar cells. *NATURE COMMUNICATIONS*, 8, MAY 11 2017.

[43] Chenyang Lu, Liangliang Niu, Nanjun Chen, Ke Jin, Taini Yang, Pengyuan Xiu, Yanwen Zhang, Fei Gao, Hongbin Bei, Shi Shi, Mo-Rigen He, Ian M. Robertson, William J. Weber, and Lumin Wang. Enhancing radiation tolerance by controlling defect mobility and migration pathways in multicomponent single-phase alloys. *NATURE COMMUNICATIONS*, 7, DEC 15 2016.

[44] Timofey Frolov, David L. Olmsted, Mark Asta, and Yuri Mishin. Structural phase transformations in metallic grain boundaries. *NATURE COMMUNICATIONS*, 4,

MAY 2013.

[45] Jihan Zhou, Yongsoo Yang, Yao Yang, Dennis S. Kim, Andrew Yuan, Xuezeng Tian, Colin Ophus, Fan Sun, Andreas K. Schmid, Michael Nathanson, Hendrik Heinz, Qi An, Hao Zeng, Peter Ercius, and Jianwei Miao. Observing crystal nucleation in four dimensions using atomic electron tomography. *NATURE*, 570(7762):500+, JUN 27 2019.

[46] Y. T. Zhu, X. Z. Liao, and X. L. Wu. Deformation twinning in nanocrystalline materials. *PROGRESS IN MATERIALS SCIENCE*, 57(1):1–62, JAN 2012.

[47] Emily B. Moore and Valeria Molinero. Structural transformation in supercooled water controls the crystallization rate of ice. *NATURE*, 479(7374):506–U226, NOV 24 2011.

[48] Patrick R. Cantwell, Ming Tang, Shen J. Dillon, Jian Luo, Gregory S. Rohrer, and Martin P. Harmer. Grain boundary complexions. *ACTA MATERIALIA*, 62:1–48, JAN 2014.

[49] Adam Nahum, Jonathan Ruhman, Sagar Vijay, and Jeongwan Haah. Quantum Entanglement Growth under Random Unitary Dynamics. *PHYSICAL REVIEW X*, 7(3), JUL 24 2017.

[50] J. G. Nuttall, G. J. O'Leary, J. F. Panozzo, C. K. Walker, K. M. Barlow, and G. J. Fitzgerald. Models of grain quality in wheat-A review. *FIELD CROPS RESEARCH*, 202(SI):136–145, FEB 15 2017.

[51] Miao Song, Gang Zhou, Ning Lu, Jaewon Lee, Elias Nakouzi, Hao Wang, and Dongsheng Li. Oriented attachment induces fivefold twins by forming and decomposing high-energy grain boundaries. *SCIENCE*, 367(6473):40+, JAN 3 2020.

[52] Donghua Dai and Dongdong Gu. Tailoring surface quality through mass and momentum transfer modeling using a volume of fluid method in selective laser melting of TiC/AlSi10Mg powder. *INTERNATIONAL JOURNAL OF MACHINE TOOLS & MANUFACTURE*, 88:95–107, JAN 2015.

[53] Nenad Miljkovic, Ryan Enright, and Evelyn N. Wang. Effect of Droplet Morphology on Growth Dynamics and Heat Transfer during Condensation on Superhydrophobic Nanostructured Surfaces. *ACS NANO*, 6(2):1776–1785, FEB 2012.

[54] Marco Bieri, Manh-Thuong Nguyen, Oliver Groening, Jinming Cai, Matthias Treier, Kamel Ait-Mansour, Pascal Ruffieux, Carlo A. Pignedoli, Daniele Passerone, Marcel Kastler, Klaus Muellen, and Roman Fasel. Two-Dimensional Polymer Formation on Surfaces: Insight into the Roles of Precursor Mobility and Reactivity. *JOURNAL OF THE AMERICAN CHEMICAL SOCIETY*, 132(46):16669–16676, NOV 24 2010.

[55] Xin-Bing Cheng, Meng-Qiang Zhao, Chi Chen, Amanda Pentecost, Kathleen Maleski, Tyler Mathis, Xue-Qiang Zhang, Qiang Zhang, Jianjun Jiang, and Yury Gogotsi. Nanodiamonds suppress the growth of lithium dendrites. *NATURE COMMUNICATIONS*, 8, AUG 25 2017.

[56] Xiao-Yu Yang, Li-Hua Chen, Yu Li, Joanna Claire Rooke, Clement Sanchez, and Bao-Lian Su. Hierarchically porous materials: synthesis strategies and structure design. *CHEMICAL SOCIETY REVIEWS*, 46(2):481–558, JAN 21 2017.

[57] Yang Cao, Song Ni, Xiaozhou Liao, Min Song, and Yuntian Zhu. Structural evolutions of metallic materials processed by severe plastic deformation. *MATERIALS SCIENCE & ENGINEERING R-REPORTS*, 133:1–59, NOV 2018.

[58] I. J. Beyerlein, M. J. Demkowicz, A. Misra, and B. P. Uberuaga. Defect-interface interactions. *PROGRESS IN MATERIALS SCIENCE*, 74:125–210, OCT 2015.

[59] Stephane Gorsse, Christopher Hutchinson, Mohamed Goune, and Rajarshi Banerjee. Additive manufacturing of metals: a brief review of the characteristic microstructures and properties of steels, Ti-6Al-4V and high-entropy alloys. *SCIENCE AND TECHNOLOGY OF ADVANCED MATERIALS*, 18(1):584–610, AUG 25 2017.

[60] F. Roters, P. Eisenlohr, L. Hantcherli, D. D. Tjahjanto, T. R. Bieler, and D. Raabe. Overview of constitutive laws, kinematics, homogenization and multiscale methods in crystal plasticity finite-element modeling: Theory, experiments, applications. *ACTA MATERIALIA*, 58(4):1152–1211, FEB 2010.

[61] I. A. Ovid'ko, R. Z. Valiev, and Y. T. Zhu. Review on superior strength and enhanced ductility of metallic nanomaterials. *PROGRESS IN MATERIALS SCIENCE*, 94:462–540, MAY 2018.

[62] Ting Zhu and Ju Li. Ultra-strength materials. *PROGRESS IN MATERIALS SCIENCE*, 55(7):710–757, SEP 2010.

[63] Taku Sakai, Andrey Belyakov, Rustam Kaibyshev, Hiromi Miura, and John J. Jonas. Dynamic and post-dynamic recrystallization under hot, cold and severe plastic deformation conditions. *PROGRESS IN MATERIALS SCIENCE*, 60:130–207, MAR 2014.

Appendix: Artifact Description/Artifact Evaluation

SUMMARY OF THE EXPERIMENTS REPORTED

Appendix: Artifact Description / Artifact Evaluation

SUMMARY OF THE EXPERIMENTS REPORTED

All calculations and scalability tests were run on the new Sunway supercomputer. The code was built with swg++, swgcc, swmpi and libswdnn_kmc. Each node on the new Sunway supercomputer has 6 core groups (CGs), each core group has 1 master processing element (MPE) and 64 computing processing elements (CPEs). For each CG maximum 16 GB memory can be used. Logs of the weak and strong benchmark tests can be obtaind from https://github.com/mainAims/sc21-tensorkmc.

Submit command: bsub -b -share_size 12000 -cache_size 0 -n $nprocs -cgsp 64 tensorkmc -in input

We ran strong benchmark tests from 24,000 to 384,000 processes with one process per CG. We ran weak benchmark tests from 12,000 to 422,400 processes with one process per CG. The x86 serial benchmark test was performed on AMD EPYC 7452 CPU and libtensorflow_cc-1.15.3. The code was compiled with g++-7.4.1 For the memory tests, runtime memory costs were measured with the ‘ps‘ command.

ARTIFACT AVAILABILITY

Software Artifact Availability: Some author-created software artifacts are NOT maintained in a public repository or are NOT available under an OSI-approved licence.

Hardware Artifact Availability: There are no author-created hardware artifacts.

Data Artifact Availability: There are no author-created data artifacts.

Proprietary Artifact Availability: None of the associated artifacts, author-created or otherwise, are proprietary.

Author-Created or Modified Artifacts:

Persistent ID:

BASELINE EXPERIMENTAL SETUP, AND MODIFICATIONS MADE FOR THE PAPER

Relevant hardware details: the new-generation Sunway supercomputer, SW26010pro many-cores processor.

The new-generation Sunway supercomputer is used for performance assessment in this work, which is the successor of the Sunway TaihuLight supercomputer. Like the Sunway TaihuLight system, the new Sunway supercomputer adopts a new-generation domestic high-performance heterogeneous many-core processors and interconnection network chips in China.

The new Sunway many-cores processor (SW26010-pro) is designed for massive thread and data parallelism and delivers high performance on parallel workloads. Each processor contains 6 core-groups (CGs), with 65 cores in each CG, and in total 390 cores. Each CG has one management processing element (MPE), one cluster of computing processing elements (CPEs), and one memory controller (MC). The MPE within each CG is used for computations, management, and communication. The CPEs are organized as an 8*8 mesh (64 cores) and are designed to maximize the aggregated computing power and minimize the micro-architecture complexity. The CPEs are organized with a mesh network to achieve high-bandwidth data communication (P2P and collective communications) among the CPEs in one CG, which is called remote scratchpad memory access (RMA).

Operating systems and versions: Sunway customized Linux with kernel version 3.10.0

Compilers and versions: swg++ (-std=c++14), swgcc

Applications and versions: TensorKMC/1.0

Libraries and versions: libswdnn_kmc, swmpi, cmake-3.17

Author-Created or Modified Artifacts:

```
Persistent ID:
↪  https://github.com/mainaims/sc21-tensorkmc
Artifact name: TensorKMC
```

BASELINE EXPERIMENTAL SETUP, AND MODIFICATIONS MADE FOR THE PAPER

Relevant hardware details: The new-generation Sunway Supercomputer, SW26010-pro many-cores processor, The new-generation Sunway supercomputer is used for performance assessment in this work, which is the successor of the Sunway TaihuLight supercomputer. Like the Sunway TaihuLight system, the new Sunway supercomputer adopts a new-generation domestic high-performance heterogeneous many-core processors and interconnection network chips in China. The new Sunway many-cores processor (SW26010-pro) is designed for massive thread and data parallelism and delivers high performance on parallel workloads. Each processor contains 6 core-groups (CGs), with 65 cores in each CG, and in total 390 cores. Each CG has one management processing element (MPE), one cluster of computing processing elements (CPEs), and one memory controller (MC). The MPE within each CG is used for computations, management, and communication. The CPEs are organized as an 8*8 mesh (64 cores) and are designed to maximize the aggregated computing power and minimize the micro-architecture complexity. The CPEs are organized with a mesh network to achieve high-bandwidth data communication (P2P and collective communications) among the CPEs in one CG, which is called remote scratchpad memory access (RMA)

Operating systems and versions: Sunway customized Linux with kernel version 3.10.0

Compilers and versions: swg++, swmpi

Applications and versions: TensorKMC/1.0

Libraries and versions: libswdnn_kmc