Code Performance and Scaling:

Data-Driven Multiscale Modeling of Liquid-Vapor Interface Dynamics During Two-Phase Cooling

Han Hu, PI
Department of Mechanical Engineering
University of Arkansas, Fayetteville, AR 72701

The performance and scaling of our machine learning (ML), density functional theory (DFT), and molecular dynamics (MD) models using TensorFlow, Quantum Espresso, and LAMMP, respectively, are benchmarked on Bridges2 GPU-AI, Bridges2 RM, and Stampede2. The cases used for the benchmark are summarized in Table 1.

Table 1. A summary of ML, DFT, and MD cases for benchmark

(a) ML cases using TensorFlow

ML cases	Data sets	Clusters
Convolutional long short-term memory (ConvLSTM) classification	Videos	Bridges2 GPU-AI
Convolutional neural networks (CNN) classification	Images	Bridges2 GPU-AI
Multilayer perceptron (MLP) classification	Images	Bridges2 GPU-AI

(b) DFT cases using Quantum Espresso

DFT cases	Pseudopotential	Number of atoms	Clusters
FeNi	Fe/Ni.pbe-n-kjpaw_psl.1.0.0.UPF	108	Bridges2 RM

(c) MD cases using LAMMPS

MD cases	Force fields	Number of atoms	Clusters
HAP minimization at 0 K	IFF	44,000 – 1,441,792	Bridges2 RM, Stampede2
HAP deformation at 300 K	IFF	120,736 – 2,816,000	Bridges2 RM
FeNi diffusion	MEAM	80,000 – 1,280,000	Bridges2 RM

Figure 1a compares the training speed of a convolutional long short-term memory (ConvLSTM) neural network model for image sequence (video)-based boiling regime classification and a convolutional neural networks (CNN) model on our local CPU (Xeon E2286M) and Bridges2 GPU-AI (NVIDIA Tesla V100 32 GB). The ConvLSTM neural networks have 260,268,035 trainable parameters which will be trained with 2,077 sequences of images, each of which consists of 50 frames of boiling images. A speedup of 34.2 on is observed for the ConvLSTM model and a speedup of 46.7 is observed for the CNN model on Bridges2 GPU-AI compared to the local CPU. It is evident that the ML code runs more efficiently on GPUs.

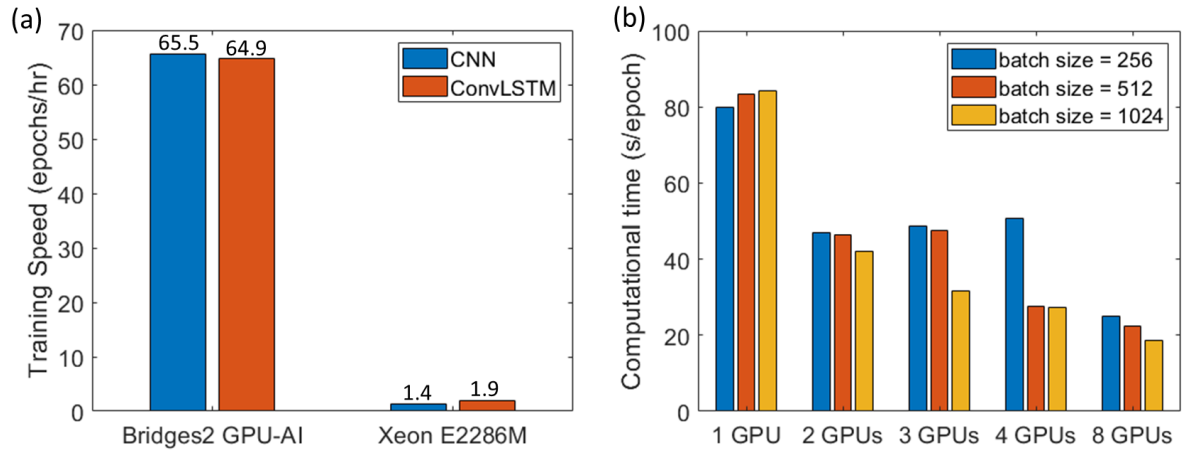

Figure 1. (a) Comparison of the training speed of ConvLSTM and CNN for image classification on PI's local CPU and Bridges2 GPU-AI. (b) Computational time per epoch of CNN for image classification on

Bridges2 GPU-AI using 1, 2, 3, 4, and 8 GPUs with different batch sizes (256, 512, 1024).

Figure 1b plots the computational time per epoch of the CNN model on different numbers of GPUs, i.e., 1, 2, 3, 4 (using the GPU-shared queue), and 8 (using the GPU queue) using three different batch sizes (256, 512, and 1024). The tf.distribute.MirroredStrategy API is used for multi-GPU computation. For all three tested batch sizes, it is observed that the computational time decreases as more GPUs are used, with one exception from 3 to 4 CPUs for the batch size of 512. When one GPU is used, the computational time becomes slightly larger when larger batch sizes are used. Nevertheless, for 2, 3, 4, and 8 GPUs, the computational time always decreases with increasing batch size. As such, it is evident that larger batch sizes are beneficial for ML code scaling on multiple GPUs.

To quantitatively examine the scaling of the CNN model on multi-GPUs, **Figure 2** plots (a) the speedup and (b) the parallel efficiency of the CNN model on single and multi-GPUs. As consistent with **Figure 1b**, the speedup and parallel efficiency also show that the CNN model scales better for larger batch sizes. For the batch size of 1,024, the model shows good scaling and high parallel efficiency (> 75%) up to 4 GPUs. Even for 8 GPUs, the parallel efficiency is still approximately 60%, which is reasonable for urgent tasks that require results within a short time frame.

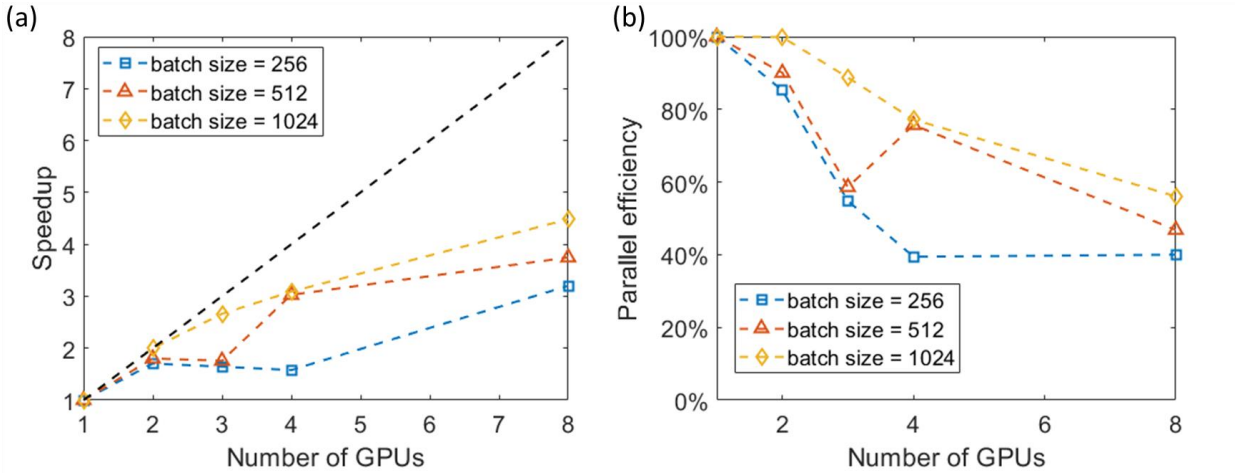

Figure 2. (a) Speedup and (b) parallel efficiency of the CNN image classification model on multiple GPUs of Bridges2 GPU-AI.

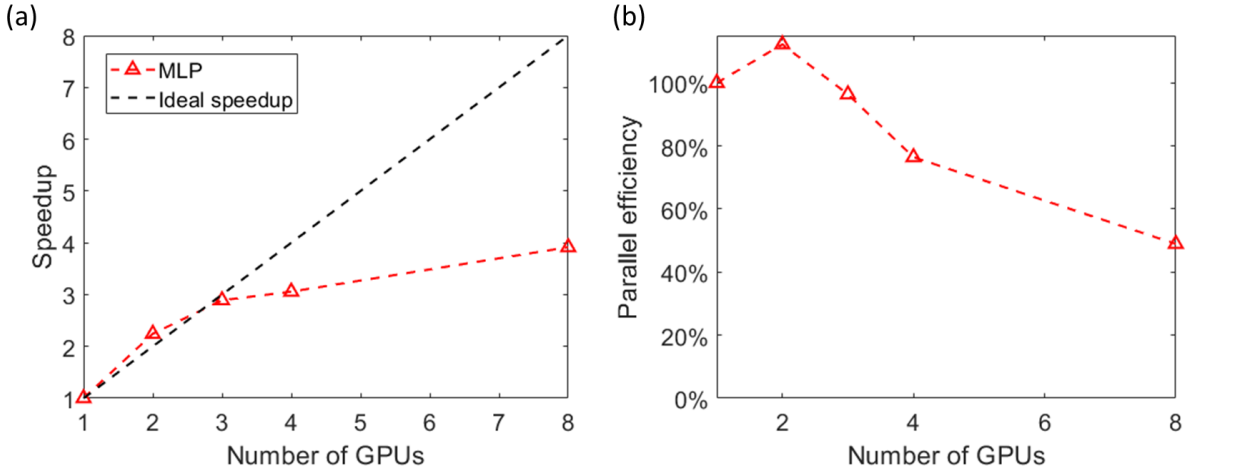

Figure 3. (a) Speedup and (b) parallel efficiency of the MLP image classification model on multiple GPUs of Bridges2 GPU-AI.

Figure 3 plots (a) the speedup and (b) the parallel efficiency of a multilayer perceptron (MLP) neural network for image classification against the number of GPUs on Bridges2 GPU-AI. The MLP model is

shown to scale very well on multiple GPUs up to 4 GPUs (efficiency > 70%). The efficiency drops to approximately 50% at 8 GPUs.

Quantum simulations using the density functional theory (DFT) are performed to model the FeNi system with the open-source package Quantum Espresso on Bridges2 RM. **Figure 4** shows (a) the speedup and (b) the fixed-size parallel efficiency for DFT simulations of the FeNi system using the pseudopotentials, Ni.pbe-n-kjpaw_psl.1.0.0.UPF and Fe.pbe-n-kjpaw_psl.1.0.0.UPF. Due to the relatively small system size (108 atoms), the fixed-size efficiency is kept high only for a small number of CPU cores. Nevertheless, 40% fixed-size parallel efficiency is kept for up to 16 cores, making parallel computation a more reasonable option than serial computation.

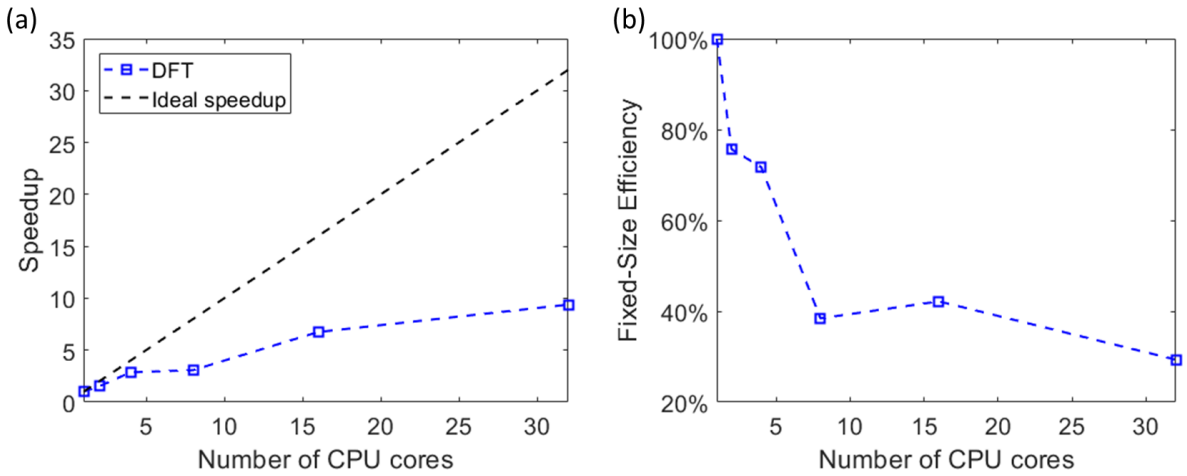

Figure 4. (a) Speedup and (b) fixed-size parallel efficiency against the number of CPU cores for DFT simulations of the FeNi system on Bridge2 RM.

Molecular dynamics (MD) simulations are performed with the open-source package LAMMPS. We have benchmarked LAMMPS using three representative MD systems, hydroxyapatite crystals at 0 K (ideal lattice), hydroxyapatite (HAP) crystals at 300 K (with lattice vibrations), and the FiNe alloy on Bridges2 RM (PSC) and Stampede2 (TACC). Three metrics are used to characterize the performance: i) the fixed-size parallel efficiency where the number of CPU cores is increased with the system size (number of atoms) kept constant, ii) the scaled-size parallel efficiency where the system size scales up as the number of CPU cores with a fixed number of atoms per CPU core, and iii) the scaled computational time per atom calculated by increasing the number of atoms in the system with a fixed number of CPU cores.

Figure 5 shows (a) speedup and (b) the fixed-size parallel efficiency versus the number of CPU cores for nonequilibrium moelcular dynamics (NEMD) simulations for calculating the diffusion coefficients of Fe and Ni in the FeNi alloy using the modified embedded atom method (MEAM) force field on Bridges2 RM. An MD system with 1,280,000 atoms is used for the calculation of the fixed-size parallel efficiency. As shown in **Figure 5**, the fixed-size parallel efficiency is kept beyond 70% for up to 256 CPU cores and beyond 60% for up to 512 CPU cores on Bridges2 RM. **Figure 6** shows (a) the speedup and (b) the fixed-size parallel efficiency versus the number of CPU cores for MD simulations for calculating the elastic moduli of hydroxyapatite crystals using the interface force field (IFF) on Bridges2 RM. MD systems with 35,200 atoms are used for the calculation of the fixed-size parallel efficiency for minimization at 0 K and deformation at 300 K. The fixed-size parallel efficiency is kept beyond 90% for up to 96 cores for the 300 K case and beyond 55% for the 0 K case. **Figure 7** shows the scaled-size parallel efficiency for (a) NEMD of FeNi interdiffusion and (**b**) MD simulations for calculating the mechanical properties of HAP at 0 K and 300 K on Bridges2 RM. For the calculation of the scaled-size parallel efficiency, the ratio of the number of atoms and the number of CPU cores is fixed at 2,500, 1,408, and 4,752 atoms/CPU-core for the FeNi interdiffusion, HAP minimization at 0 K, and HAP deformation at 300 K systems, respectively. For all three cases tested, the scaled-size parallel efficiency is kept beyond 80% for up to 256 CPU cores.

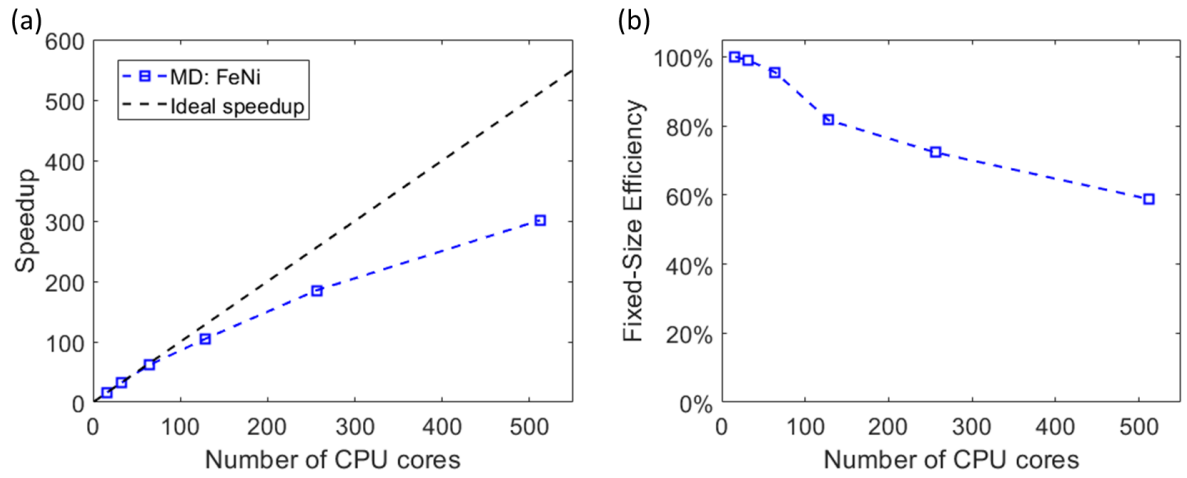

Figure 5. (a) Speedup and (b) fixed-size parallel efficiency against the number of CPU cores for NEMD simulations of the FeNi interdiffusion on Bridge2 RM.

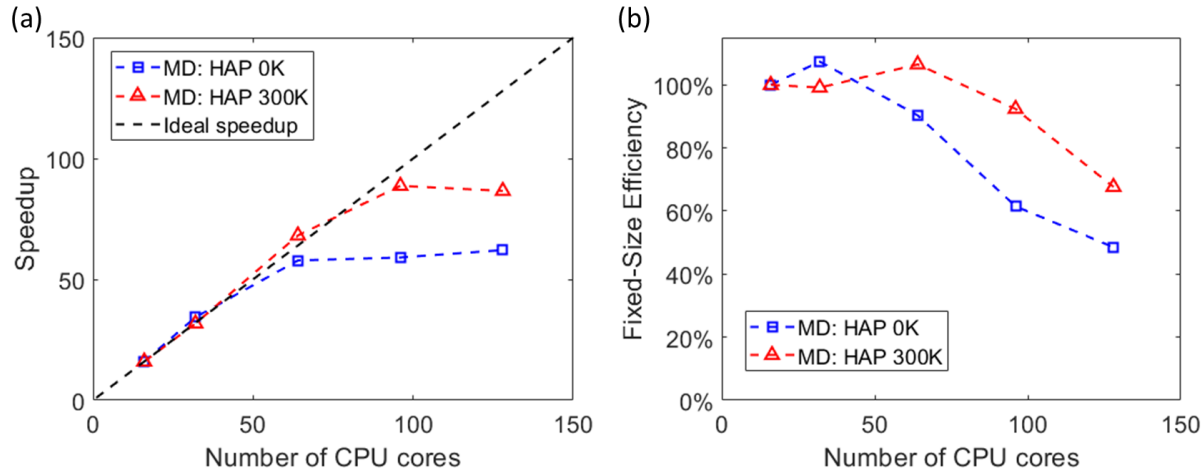

Figure 6. (a) Speedup and (b) fixed-size parallel efficiency against the number of CPU cores for calculating the mechanical properties of HAP at 0 K and 300 K on Bridge2 RM.

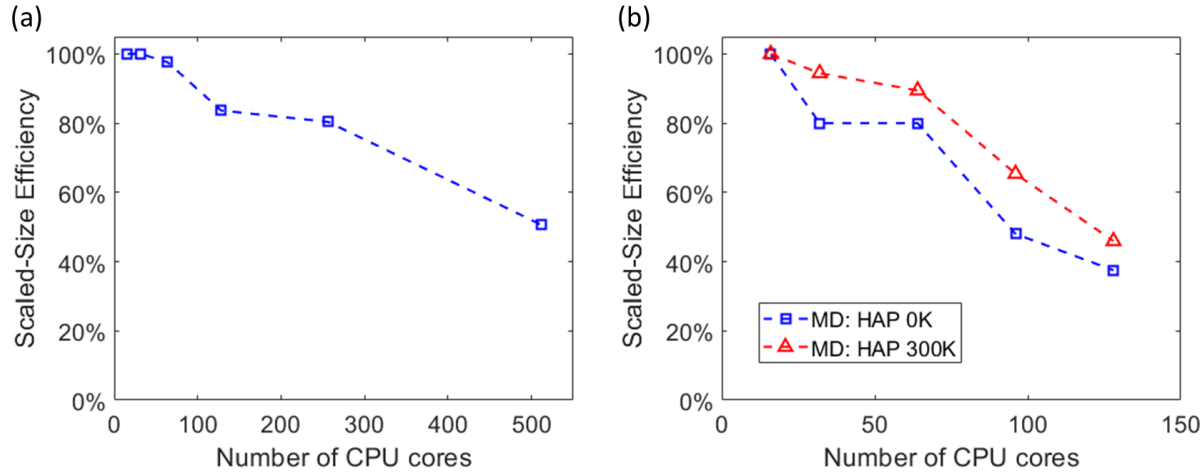

Figure 7. Scaled-size parallel efficiency against the number of CPU cores for (a) NEMD of FeNi interdiffusion and (b) calculation of the mechanical properties of HAP at 0 K and 300 K on Bridge2 RM.

In addition to Bridges2 RM, MD simulations are also performed on Stampede 2. **Figure 8** shows (a)

the speedup and (b) the fixed-size parallel efficiency of MD simulations for calculating the elastic moduli of hydroxyapatite crystals at 0 K on Stampede2. The interface force field is used for the MD simulations and an MD system with 35,200 atoms is used for the calculation of the fixed-size parallel efficiency. As shown in **Figure 8**, the fixed-size parallel efficiency is kept beyond 90% up to 384 cores and beyond 60% up to 768 cores. **Figure 9** shows the scaled-size parallel efficiency as a function of the number of CPU cores for MD simulations for calculating the mechanical properties of HAP at 0 K using the IFF. For the calculation of the scaled-size parallel efficiency, the ratio of the number of atoms and the number of CPU cores is fixed at 1,408 atoms/CPU-core and the number of atoms in the MD systems ranges from 44,000 – 1,441,792. It is observed that the scaled-size parallel efficiency is kept beyond 80% with up to 192 CPU cores and beyond 60% up to 768 cores. In the proposed study, 96 cores will be used for the computation of the infiltration cases to keep high parallel efficiency.

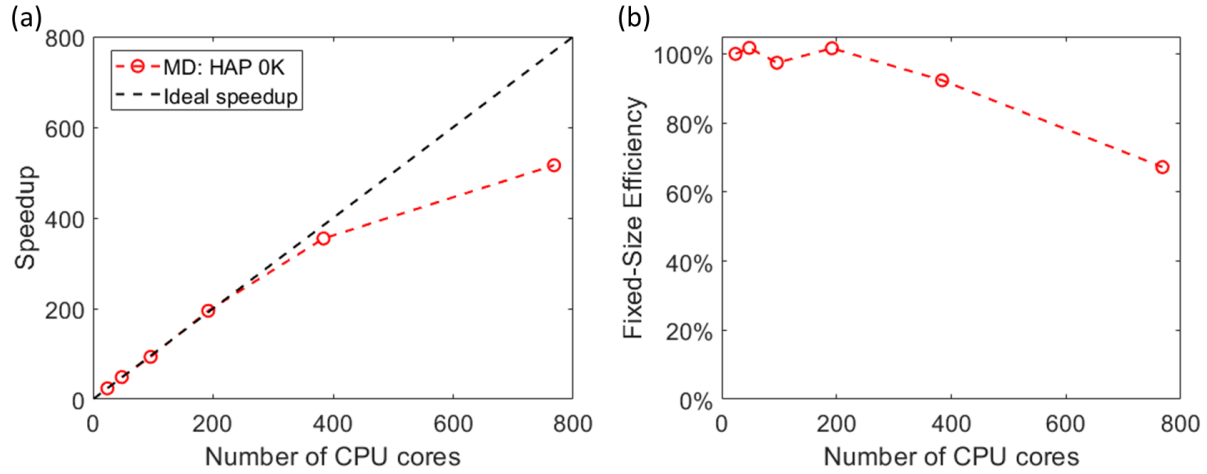

Figure 8. (a) Speedup and (b) fixed-size parallel efficiency against the number of CPU cores for calculating the mechanical properties of HAP at 0 K on Stampede2.

To summarize, our machine learning codes run very efficiently on the NVIDIA Tesla V100 34 GB GPUs of Bridges2 GPU-AI, with a more than 30 speedup compared to our local CPU (Xeon E2286M). The ML codes are also shown to scale reasonably well on multi-GPUs, particularly with the number of GPUs below 4. The DFT and MD simulations, listed in Table 1b and 1c, scale well on Bridges2 RM and Stampede2. The computational plan detailed in Section 5 of the main document is developed based on the results and conclusions of code benchmarks.

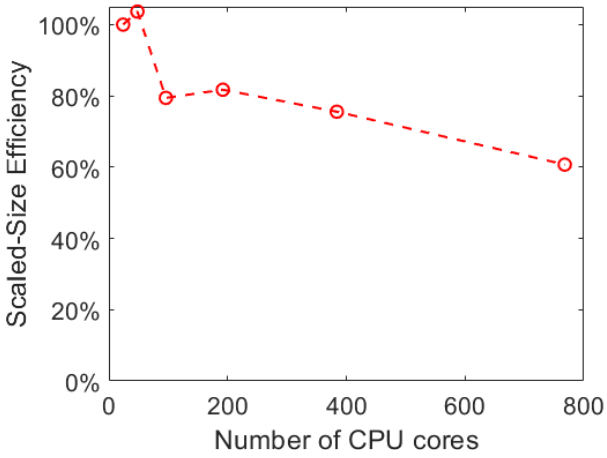

Figure 9. Scaled-size efficiency against the number of CPU cores for calculating the mechanical properties of HAP at 0 K on Stampede2.