Chapter 8
Multi-scale Continuum-Particle Simulation on CPU–GPU Hybrid Supercomputer

Wei Ge, Ji Xu, Qingang Xiong, Xiaowei Wang, Feiguo Chen, Limin Wang, Chaofeng Hou, Ming Xu and Jinghai Li

Abstract This chapter serves as an introduction to the supercomputing works carried out at CAS-IPE following the strategy of structural consistency among the physics in the simulated systems, mathematical model, computational software expressing the numerical methods and algorithms, and finally architecture of the computer hardware (Li et al., From multiscale modeling to Meso-science—a chemical engineering perspective, 2012; Li et al., Meso-scale phenomena from compromise—a common challenge, not only for chemical engineering, 2009; Ge et al., Chem Eng Sci 66:4426–4458, 2011). Multi-scale simulation of gas-solid flow in continuum-discrete approaches and molecular dynamics simulation of crystalline silicon are taken as examples, both making full use of CPU-GPU hybrid supercomputers. This strategy is demonstrated to be effective and critical for achieving good scalability and efficiency in such simulations. The software and hardware systems thus designed have found wide applications in process engineering.

8.1 Background

Process engineering is a collective term covering a wide range of industries and disciplines, from traditional chemical, metallurgical and mineral domains, to the fast-growing material, biological, pharmaceutical and cosmetic areas. Despites their apparent diversity, they do share some general activities such as the transformation and utilization of energies and resources, which are fundamental and critical for the whole society. A more intrinsic similarity underlying these activities is the vast scale gap between the products and production equipments in these industries and the multi-scale dynamic structures spanning this gap. For example, the properties and

W. Ge (✉) · J. Xu, Q. Xiong · X. Wang · F. Chen · L. Wang · C. Hou · M. Xu · J. Li
Institute of Process Engineering (IPE), Chinese Academy of Sciences (CAS),
100190 Beijing, China
e-mail: wge@home.ipe.ac.cn

D. A. Yuen et al. (eds.), *GPU Solutions to Multi-scale Problems in Science and Engineering*, 143
Lecture Notes in Earth System Sciences, DOI: 10.1007/978-3-642-16405-7_8,

quality of the gasoline we use for our cars is determined by the molecular structures and fractions of its compositions, which is at the scale of 10^{-10}–10^{-9} m, while the reactors for refining gasoline from crude oil, such as the Fluid Catalytic Cracking (FCC) facilities, are typically 50–80 m high.

Therefore, it is not surprising that simulation of such processes has become one of the most demanding area for high performance computing. However, the actual performance of traditional simulation softwares on general purpose supercomputers is, as a whole, not impressive, and sometimes even very frustrating. In some computational fluid dynamics (CFD) simulations on commercial multi-phase reactors, the scalability is limited to dozens of CPU cores albeit more than a quarter million cores are available in modern high-end supercomputers. Even for these cores, the sustainable performance is about 10–20 % of the corresponding peak values.

In principle, this situation is not ascribed to the status of the technology for elemental components at the hardware level, but to the lack of coordination among the models, algorithms and hardwares involved in the simulations. In short, the physical world features multi-scale structures and the computer hardwares are most easily and efficiently organized in a multi-scale manner (at least in terms of their logical architecture). However, the mathematical model and numerical algorithms in traditional simulations only discretize and partition the physical system at a single scale, which incurs excessive long-range and global correlations in the model, and hence data dependence in the algorithm and communications among hardware components in execution. This is the main reason for the low efficiency and poor scalability of traditional simulation softwares in process engineering.

Based on this understanding, systematic multi-scale simulation approaches, from mathematical model to computer hardware, are implemented for gas-solid flow and crystalline silicon. All implementations have reflected the consistency among the physics, model, algorithm and hardware, which are summarized, in a more general sense, by the so-called *EMMS Paradigm* (Li et al. 2009, 2013; Ge et al. 2011).

Currently, the mainstream simulation method for gas-solid flow is the two-fluid model (TFM, Anderson and Jackson 1967; Gidaspow 1994), which treats both the gas and solid phases as continuum. It is considered advantageous for industrial simulations as its computational cost is not necessarily linked to the scale of the system, but to the number of numerical cells which is determined flexibly by the desired resolution. However, due to the intrinsic discrete nature of the solid phase, its constitutive laws as a continuum are not easily obtained, and may not exist at all. Especially, the meso-scale heterogeneity presents below the numerical grid scale proposed great challenges to quantify its statistical behavior and hence the constitutive laws. Therefore, the accuracy of TFM is not satisfactory for engineering purpose in general. On the other hand, direct discrete presentation of the solid phase, though more reasonable and simple, is far beyond the capability of current computing technology, just imaging that an industrial gas-solid reactor may contain trillions of interacting particles and advancing one particle for one time step, typically below milliseconds, may cost hundreds to thousands of flops.

Table 8.1 Specifications of the Mole-8.5 system (Wang et al. 2010, 2012; Ge et al. 2011) (adapted from Li et al. (2013), Dubitzky et al. (2012), Ge et al. (2011))

Peak performance in single precision	2.206 Petaflop/s
Peak performance in double precision	1.103 Petaflop/s
Linpack sustained performance	496.5 Teraflop/s (on 320 nodes)
Megaflop/s per Watt	963.7 (Linpack)
Number of nodes/Number of GPU's (Type)	362/2088 (Tesla C2050)
Top layer	2/0
Middle layer	18/36 (Tesla C2050)
Bottom layer	342/2052 (Tesla C2050)
Total RAM	17.8 Terabyte
Total VRAM	6.5 Terabyte
Total hard disk space	720 Terabyte
Management communication	H3C Gigabit Ethernet
Message passing communication	Mellanox infiniband quad data rate
Occupied area	150 sq.m.
Weight	12.6 ton
Max power	600 kW (computing) + 200 kW (cooling)
Operating system	CentOS 5.4, PBS
Monitor	Ganglia, GPU monitoring
Programming languages	C, C++, CUDA

In recent years, however, developments in many-core computing and coarse-grained discrete modeling begin to show the feasibility of industrial scale discrete solid phase simulation (Xu et al. 2012). Similar to pseudo-particle modeling (Ge and Li 1996, 2003), real solid particles can be presented by much less number of computational particles, whose properties can be measured in simulations and mapped physically to the solid phase (Zhou et al. 2010), which expresses the consistency among the simulated system, the physical model and the numerical method. Evolution of the computational particles features additive and localized operations which are best carried out by many-core processors, such as GPUs, in the highly parallel mode of single-instruction multi-data (SIMD). The gas flow can be solved either by traditional finite difference (FD) or finite volume (FV) methods, or by LBM methods, at scales either above or below the particle scale, which are suitable for CPUs or GPUs, respectively. Thus, the consistency among the *Four Elements* is presented, as summarized in Table 8.5, and the *EMMS Paradigm* can thus be implemented, with a preliminary version found in Ge et al. (2011).

8.2 Physical Model

Although we will focus on the algorithmic and computational aspects of the *EMMS Paradigm*, it is helpful to briefly revisit its physical background and models first. Most gas-solid systems in industries are confined in certain geometries, usually equipment walls, and are operated under steady conditions. The time-averaged steady

state distribution of the flow variables, such as gas and fluid flow velocities and solids concentration, can be predicted with reasonable accuracy by some macro-scale models, such as the global EMMS model (at the reactor level) with some empiric correlations (Ge et al. 2011; Liu et al. 2011). These distributions are then served as the initial conditions for simulating the spatio-temporal evolution of the flow structures in the systems, which basically constitutes the descriptions for the gas phase, the solid phase and their interactions, as introduced below.

The gas phase model below the particle scale is similar to single phase flow, which can be well described by the classical Navier-Stokes (N-S) equation except additional boundary conditions at particle surfaces. Above the particle scale, however, the flow structure induced by the embedded particle may cause the deviation of its effective properties (e.g., viscosity and pressure), from pure gas, and significant nonlinearity is found. Correlations for these properties can be obtained in direct numerical simulations (DNS) based on the N-S equation or Boltzmann equation. Coarse-grained LBM may provide another basis for the modeling of the gas phase, where the partial occupation of the solid phase and different permeabilities are allowed (Wang et al. 2012). With the introduction of multi-relaxation time (MRT) and large-eddy simulation (LES), and proper smoothing of the boundary configuration, the method may sustain high velocity and pressure different for the lab-scale reactor simulation (Yu et al. 2006). In all these attempts, the compressibility of gas phase can be increased to facilitate the numerical methods without affecting the accuracy very much.

The solid phase can be described either as a continuum or a discrete material. For higher resolution, the discrete description is preferred, and in order to reduce computational cost, coarse-graining of the real solid particles or description of their collective behavior is desirable. Several approaches are followed for this purpose:

Coarse-grained particles: In this approach, we try to simulate a much smaller number of elements to present the same statistical behavior of a huge number of real particles. To achieve this equivalence, the simulated particles will be, in general, more dissipative (with lower restitution) as compared to real particles, so as to maintain the energy balance, and more elastic to accommodate deformability, and less frictional to keep fluidity. The time step for these coarse-grained particles can be much larger than real solids, which further improve its efficiency. Usually, number dependence of the constitutive laws sets in when the particle number is small enough, which caps the extent of such coarse-graining.

Particle parcels: On the other hand, we may try to approximate the behavior of a swarm of particles as a single one, vividly called a parcel. Such parcels have continuous interactions with their neighbors, in a manner much more complicated than single particles, so as to account for the deformation and the exchanges of mass and momentum between the parcels. Smoothed particle hydrodynamics for the solid phase (Xiong et al. 2011) may present a framework model for the parcels with rational basis, but adjustments to its particle properties are necessary.

Particle clusters: In gas-solid systems, the particle distribution is very heterogeneous. Most particles aggregate to form islands in the gas flow field with few particles (the

so-called dilute phase). Such particle clusters can be taken as natural discrete entities for simulation purpose, and it can be larger than the coarse-grained particles or particle parcels discussed above. However, the shapes of clusters are usually very complicated and deformable, which have to be simplified drastically. The energy-minimization multi-scale (EMMS) model (Li et al. 1988; Li and Kwauk 1994), from which the EMMS paradigm is developed, can be employed as a rational basis for determining the effective size of the clusters.

Grid based approaches: Some (partially) grid-based approaches also possess particulate nature and can be used for the simulation of the solid phase. Particle in cell (PIC, Harlow 1988) methods is a hybrid Euler-Lagrange description of fluid flow, where fluid is tracked as a collection of mass carriers, statistics on these carriers are then performed via a Eulerian grid, and the continuum equations are solved on the grid numerically with the statistical data, which give the flow field. The velocities of the mass carriers are then interpolated from the grid values and their positions are updated individually, and so forth. As the solid phase is intrinsically discrete, PIC for the solids may be proven to be more reasonable (Li et al. 2012). In fact, PIC is similar to SPH expect it is partly grid-based. That means, similar difficulties will be faced, such as the collapse of particles at high concentration gradient. Insertion of a DEM core may also be helpful for this method, or otherwise, the method can be switched to DEM or parcel based methods when certain concentration or concentration gradient limits are met.

Note that, we have also listed in Table 8.5 a continuum model for the solid phase, that is, considering the solid phase as highly compressible gas with collisional cooling. However, as the numerical method for simulating such gas is explicit and lattice-based, it is algorithmically similar to particle methods with fix neighborhood. Therefore, the whole framework of the implementation is still of the continuum-particle type. The high non-linearity of the state equation of the solid phase, that is, the dramatic increase of the solid phase stress near minimum fluidization voidage, may present a difficulty.

The gas and the solid phases are coupled by the interfacial forces, mainly the drag between them. For uniform suspension of the particles, the drag can be well predicted by semi-empirical correlations, such as the Wen and Yu (1966) equation linking the drag with local slip velocity and particle concentration. Under more general conditions, the EMMS model or similar approaches (Xu et al. 2007) should be used to account for the effect of non-uniformity in the gas and/or solid phases.

8.3 Numerical Methods and Algorithm

For the physical models described above, the corresponding numerical methods can be selected or developed, and then software algorithms are designed for these methods with considerations to the computing hardware available. We will discuss the

numerical methods for the gas and solid phases, respectively, and then the major types of algorithms they can share.

8.3.1 Gas Phase Simulation

Accurate numerical methods must reflect the nature of the physical model. The gas phase in most gas-solid systems is nearly incompressible, that means flow at one location is affected by other locations simultaneously. Implicit methods are, therefore, more accurate for the gas phase because it can reflect this global dependence. However, this dependence is also expressed in its algorithm, which is boiled down to the solving of linear equation sets featuring sparse matrixes. Low computation to data accessing rate, global data dependence and hence poor scalability are the major challenges for efficient implementation of this method on massive parallel computers. Multi-core CPUs with large shared memory coupled with message passing interface (MPI) is suitable for these algorithms as explicit data communication can be minimized. But as the communication inevitably increases non-linearly with the number of CPUs involved, it is desirable to use coarse grid for the gas phase, so as to reduce the computational cost. In this regard, meso-scale models considering the distribution of gas flow in the grids and the appropriate drag law is critical for maintaining reasonable accuracy.

When high resolution of the gas phase is required, explicit methods may become more favorable, since no global data dependence and iterations are involved, and updating of the data at each grid point requires only data in neighboring grids, which allows virtually unlimited weak scalability and hence spatial scale. But this is at the price of much finer grid and time step to recovery the physical global dependence at larger spatial-temporal scales. Weak compressibility is assumed in the model, which may introduce further errors to the model, especially for the pressure distribution. These prices are paid off only when the system is large enough. Therefore, LBM and explicit FD or FV methods are more suitable for resolving the gas phase at the scale comparable or smaller than the solid entities (particles, parcels or clusters). One get-around may be provided by a modification to the physical picture of flow. At relatively high particle concentration (e.g., above 1 %), the mass of the flow is mainly carried by the solid phase, and hence the actual density distribution of the gas phase becomes less important to the flow of the mixture, as long as they provides a similar flow distribution and drag force. In this case, the compressibility of the gas phase can be increased artificially, bring the Mach number to the range of about 0.3–0.5, to validate the use of explicit numerical schemes for compressible flow. Adjustments to the drag coefficients are required to maintain the same level of inter-phase frictions. These explicit methods are intrinsically suitable to GPUs or other single-instruction multi-data (SIMD) manycore processors, which are highly parallel in computation and largely localized in memory access.

For implementation of these methods, open source or commercial software, such as Fluent (http://www.ansys.com) can be used besides development from scratch. With its user interface, we can exchange particle data with the cells of the software through files. To speedup the process, we may start multiple Fluent processes in a domain-decomposition mode, which also communicate through files. To accelerate file reading and writing, virtual disks can be installed in the memory. And most importantly, the amount of data exchanged between the solid and gas phase should be minimized. In principle, only cell averaged voidages and velocities should be included.

8.3.2 Solid Phase Simulation

Particle methods can be employed for solid phase on different coarse-graining levels with similar numerical methods and algorithms. The interactions between the particles are processed as the numerical integration of the forces between neighboring particles, which is pairwise additive and explicit, and the interactions are organized through a neighbor detecting process and followed by the updating of the particle positions. Though interactions may present the most time-consuming part of the algorithm, neighbor detection is usually the most complicated part and is critical to the efficiency of the algorithm. Cell-list and neighbor-list algorithms are the two mainstream approaches for this part, which are suitable for fast changing and more stable neighborhood, respectively. All procedures of the particle methods can be implemented on GPUs with higher speed as compared to CPUs, but extensive optimizations is necessary to reach best performance.

Note that, explicit numerical methods for continuum models are computationally a simplified form of the particles methods, where the complicated neighbor detecting process is not needed anymore. Highest performance can be achieved on GPUs with these methods, if the operations on the grid data are computationally intensive. As the solid phase is highly compressible, continuum description solved by explicit FD or discrete kinetic method (DKM) can be most efficient, though not the most accurate in general, and it is still fit well into the continuum-particle implementation of the *EMMS Paradigm*. On the other hand, the PIC method presents a hybrid continuum-particle method, where particles do not interact pairwisely, but collectively via the grids, which is also applicable to this implementation.

8.3.3 General-Purpose Particle Simulator

As we know from the discussions above, discrete particle simulation can be employed in different forms for both gas and solid phases. In a more general background, it also covers a variety of systems and processes, such as granular flow (Liu et al. 2008),

emulsions (Gao et al. 2005), polymers (Xu et al. 2010) and proteins (Ren et al. 2011), foams (Sun et al. 2007), micro-/nano-flows (Chen et al. 2008), crystals (Hou et al. 2012) and reaction-diffusion processes. The efficiency and scalability of discrete simulation was demonstrated repeatedly in these works, and the common nature of discrete methods that leads to these advantages, namely additivity and locality, is also recognized (Ge et al. 2011; Ge and Li 2000, 2002). Here additivity refers to the interactions between the particles which can be processed independently at the same time and then sum up to give the resultant force on a particle. It ensures that parallel computing can be carried out at a very fundamental level of the algorithm, that is, fine-grain parallelism. On the other hand, the locality refers to the fast decay of the strength of such interactions, so that only local interactions should be considered rigorously. It provides the parallelism at a larger scale and the weak scalability of the algorithm.

This common nature enables us to develop a general-purpose platform for particle methods at different coarse-graining levels (Ge and Li 2000, 2002; Tang et al. 2004; Wang et al. 2005), from atoms and molecules at micro-scale to boulders at macro-scale, and from real particles to more complicated discrete entities representing particle clusters. With these methods, the full range of phenomena in process engineering, from atoms to apparatus, can be simulated, the general structure, main modules and functions of the platform are summarized in Fig. 8.1.

8.3.4 GPU Implementation of the Particle Simulator

This platform for particle simulation was originally developed for CPU-based massive parallel systems. With the development of GPGPU and its programming environment, the time is ripe for transplanting the platform to CPU+GPU hybrid systems. Although other approaches, like the implicit PDE solver for the gas, have been tried with encouraging success (Wang et al. 2010), particle simulation is, in a broader sense, more suitable for GPU implementation. As detailed in Ge et al. (2011), the cell-list and neighbor-list schemes are combined in our GPU implementation, where cell list is employed to traverse all elements and find their interacting neighbors which are then put into their neighbor list. When putting the particles into cells, one thread is preferably assigned with one particle. Thanks to the atomic functions supported by Nvidia C2050 GPUs, one cell can contain several particles, but the write conflict, occurred when multiple threads write to the global memory can be avoided. The neighbor list thus generated for each particle is stored in a two dimensional array in the global memory of the GPU. In this way, although memory redundancy is unavoidable, coalesced global memory access is achieved. When generating the neighbor list, one block corresponds to one cell with each particle in it assigned to a different thread to speedup the computation. The particle information of the local and neighboring cells are buffered in the shared memory to reduce the global memory

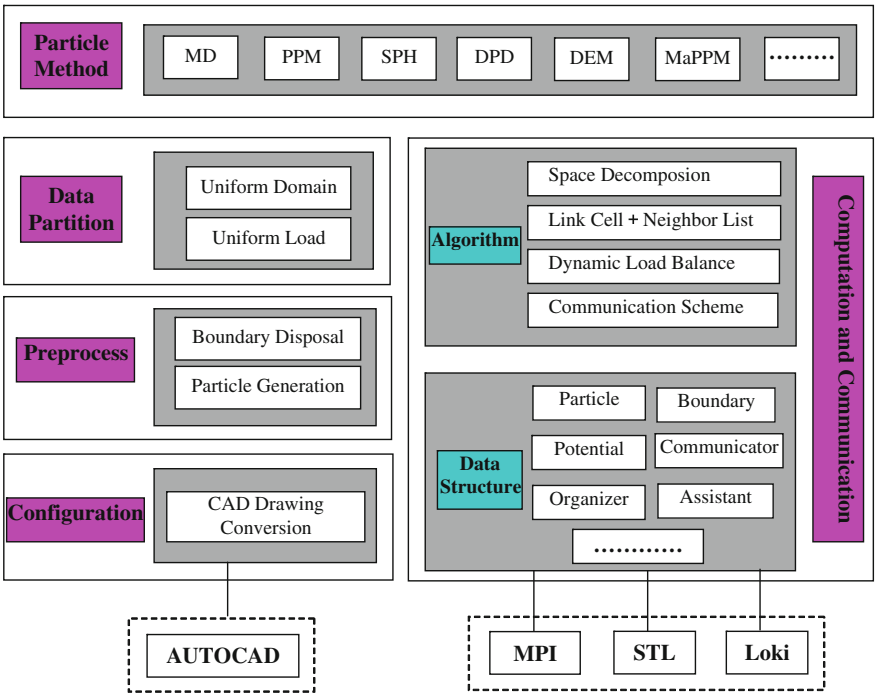

Fig. 8.1 General algorithmic platform for discrete simulation (Tang et al. 2004)

access. The overall flow chat for the general algorithm is show in Fig. 8.2 reproduced from Ge et al. (2011).

For simpler cases of fixed neighbors and for processing the interactions after the neighbors are listed, similar algorithms can be shared, also with explicit finite difference or finite volume methods, lattice-based methods and MD methods for condensed materials at low temperature. They are usually very efficient for GPUs, due to their spatial locality, natural parallelism and explicit schemes.

Though extensive optimizations are required to implement the various interactions between the discrete elements on GPUs, our emphasis has been on the effective use of the device memory bandwidth, since it is common to most methods, and it is especially important for methods with low ratios of computational operations to memory access. For best performance, the data in registers and local memories should be reused as much as possible, and storing and loading of the data to global memory should aligned and coalesces. LBM may serve as a typical example for memory bounded applications on GPUs and interesting readers are referred to our recent publications (Ge et al. 2011; Xiong et al. 2012; Li et al. 2012).

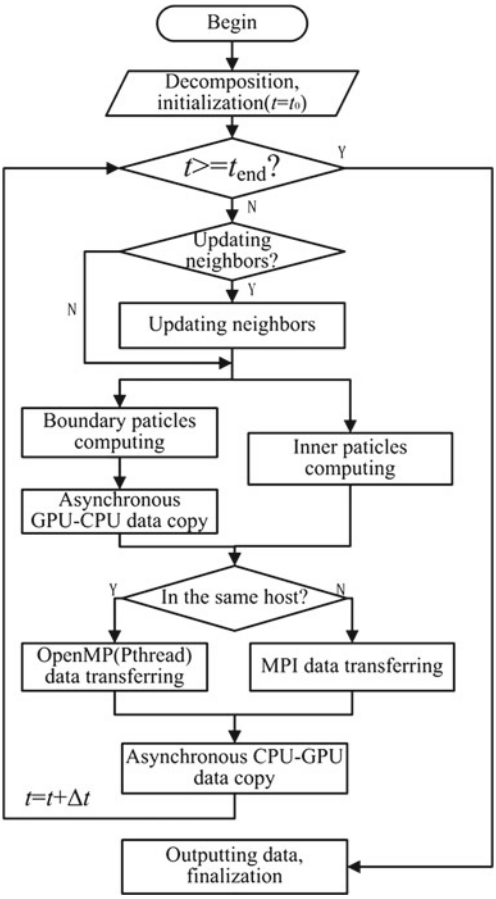

Fig. 8.2 General purpose particle simulation algorithm on multiple GPUs (adopted from Ge et al. 2011)

8.4 Hardware Development

With the development and extension of the EMMS model to different areas and the expression of the common nature of different discrete methods under the same algorithmic framework, a general multi-scale computing mode was established (Chen et al. 2009; Ge et al. 2011; Li et al. 2012) for typical complex system in process engineering. In this mode, the system is discretized on different levels. On the top and middle levels, long range interactions or correlations are treated by imposing stability conditions, which gives the global and local distribution of variables at the statistically steady state with relatively low computational cost; While on the middle and bottom levels, local interactions among the discrete elements are treated explicitly based on these distributions, reproducing the dynamic evolution of the system in detail. Taking advantage of the fast

Fig. 8.3 The Mole-8.5 system at IPE, CAS (Photo by Xianfeng He) (adopted from Li et al. 2013; Ge et al. 2011; Dubitzky et al. 2012)

distribution process, development of system behavior from the artificial initial condition to the steady state, which is of little interest to engineering practice, can be bypassed almost completely, and hence speedup the simulation considerably (Ge et al. 2011; Liu et al. 2011, 2012).

However, with traditional CPU-based supercomputers, no significant advantage of this computing mode can be demonstrated because the interactions and motion of the particles are processed with very limited parallelism as compared to its full potential. The advent of GPU computing, facilitated by CUDA^{TM}, introduced new means to implement this mode. As GPUs typically contain hundreds of relatively simple stream processors operated in the SIMD mode, they have a good balance, for discrete simulation, between the complexity of the arithmetic or logic operations that can be carried out by a stream processor and the number of parallel threads they can run. The communication among multiple GPUs may present an imperfection, as for the moment it has to resort to the PCIE bus and CPUs, or even the inter-node network, with limited bandwidth and considerable latency. However, weak scalability is still warranted for most discrete simulations.

The Mole-8.5 system (Wang et al. 2010, 2012; Ge et al. 2011; Li et al. 2013) at IPE, pictured in Fig. 8.3, is the first supercomputer using NVIDIA Tesla C2050 GPU boards in the world, reaching 1PFlops peak performance in double-precision. It was established to provide a customized hardware that can taking full advantage of the

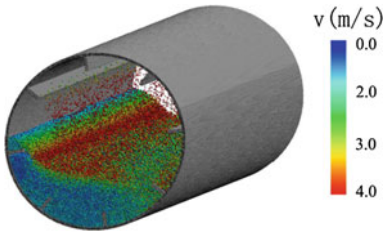

Fig. 8.4 Snapshot from the simulation of the industrial scale rotary drum (adapted from Xu et al. 2011)

CPU-GPU hybrid architecture to implement the multi-scale computing mode based on EMMS model and discrete simulations. It features a three-layer structure with increasing number of GPUs per node at lower layers, as specified in Table 8.1. We demonstrate that this design is economically profitable for most discrete simulations though it may not give good results for Linpack tests.

8.5 Applications

The multi-scale computing mode introduced above has been applied to a wide range of processes in chemical and metallurgical engineering, molecular biology and renewable energy, either for industrial designing and optimization, or for purely scientific exploration. Even a full H1N1 viron in vivo can be simulated on the molecular level at a speed of 0.77 ns per day (Xu et al. 2011). We will give some further examples below.

8.5.1 Quasi-Realtime Simulation of Rotating Drums

To demonstrate how discrete particle simulation can be accelerated by GPU or many-core computing, we carried out a DEM simulation on the granular flow in rotating drums which are widely used in process industries (Xu et al. 2011a). When a simple interaction model for smooth particles is used, each C2050 GPU can process at most about 90 million particle updates per second, about two orders faster than the serial code on CPUs. And when an industrial scale rotating drum, 13.5 m long and 1.5 m in diameter with nearly 10 million centimeter particles (a segment of the drum is show in Fig. 8.4) are simulated on 270 GPUs with message passing interface (MPI), nearly realtime speed can be achieved (Xu et al. 2011a) even when a more comprehensive tangential interaction model was added.

Table 8.2 Outline of the multi-scale approach to DNS of gas-solid suspension

System components	Physical model	Numerical method	Software algorithm	Hardware
Gas phase	Continuum (Boltzmann equation)	Lattice Boltzmann (fine grid)	Regular, explicit & local lattice operations	Linked many-core (e.g. GPU)
Solid phase	Discrete particles (Newton)	Integration of ordinary differential equations	List & arithmetic operations	Shared memory multi-core (e.g. CPU)

8.5.2 Direct Numerical Simulation of Gas-Solid Suspension

When gas-solid systems are simulated, the multi-scale computing mode can be fully exemplified (Ge et al. 2011; Xiong et al. 2012). For DNS, the consistency from the simulated system to computing hardware is detailed in Table 8.2. In this method, we have carried out the largest scale DNS of gas-solid systems so far (Ge et al. 2011; Xiong et al. 2012), which contains more than 1 million solid particles with 1 billion lattices for the gas phase in 2D, and 100 thousand particle with 500 million lattices in 3D. Some of the results are shown in Figs. 8.5 and 8.6. Some 20–60 folds speedup is obtained when comparing one GPU with one CPU core.

8.5.3 Euler-Lagrangian Simulation of Gas-Solid Fluidization

DNS of gas-solid flow has revealed unprecedented details of the flow field which is important for the establishing larger scale models for industrial applications (Xu et al. 2012). However, its direct application in industry is very limited. Most industrial simulations have employed TFM which treat both gas and solid phases as continuum and follows a Euler-Euler frame of description. This is certainly insufficient in terms of accuracy but was previously the only feasible method due to computational cost. Now with our multi-scale computing mode, a Euler-Lagranian method with less computation than DNS and higher resolution than TFM can be employed for industrial simulations (Xu et al. 2012). As detailed in Table 8.3, the solid particles (either real or coarse-grained) are still tracked one by one as in DNS, which is the Lagrangian part, but the gas flow is resolved at a scale much larger than the solid particles using continuum-based finite volume method, which constitutes the Eulerian part. With GPU computing for the Lagrangian part, its speed can be comparable to traditional TFM simulation on CPUs (Xu et al. 2012).

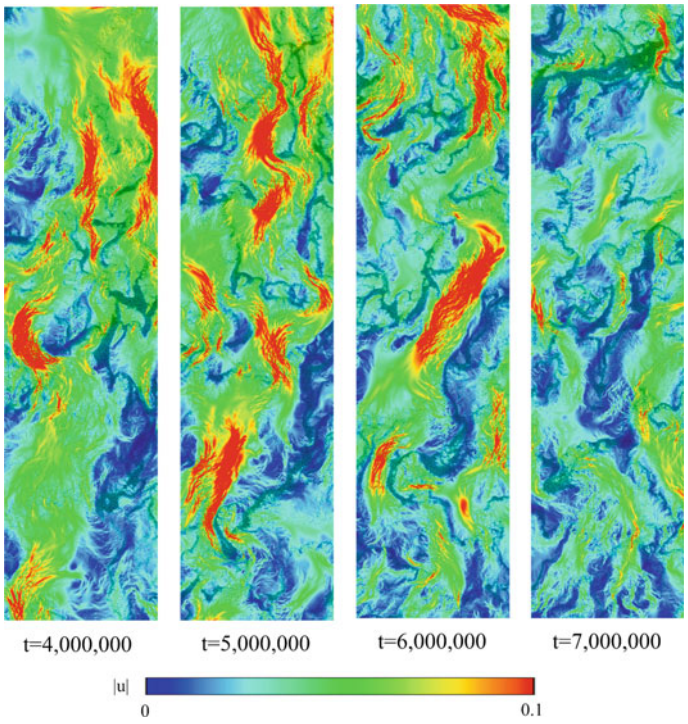

Fig. 8.5 Snapshot from 2D DNS of gas-solid suspension (adopted from Ge et al. 2011; Xiong et al. 2012)

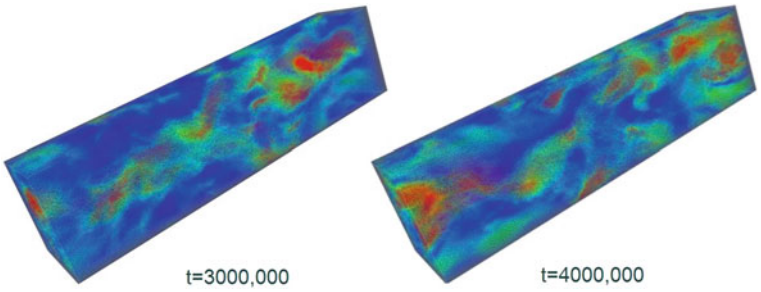

Fig. 8.6 Snapshot from 2D DNS of gas-solid suspension (adopted from Ge et al. 2011; Xiong et al. 2012)

Table 8.3 Outline of the multi-scale approach to Euler-Lagrangian simulation of gas-solid flow

System components	Physical model	Numerical method	Software algorithm	Hardware architecture
Solid phase	Particles (Newton)	ODE integration	List & arithmetic operations	Linked many-core (e.g., GPU)
Gas phase	Continuum (N-S)	PDE solver (Simple)	Sparse matrix operations	Shared memory multi-core (e.g., CPU)

Table 8.4 Outline of the multi-scale approach to atomistic simulation of crystalline silicon

System components	Physical model	Software algorithm	Hardware architecture
Bulk atoms (majority)	Regular lattices with fixed neighbors	Massive and intensive computing in SIMD-style, single precision allowed	Shared memory many-core (e.g., GPU)
Interface/defects/ dopants (minority)	Irregular lattices with dynamic neighbors	Less but more complicated computing in MIMD-style, double precision required	Linked multi-core (e.g., CPU)

8.5.4 Atomistic Simulation of Crystalline Silicon

The multi-scale computing mode can be used in areas other than multi-phase flow. One example is the atomistic simulation of crystalline silicon and its surface reconstruction (Hou and Ge 2011; Hou et al. 2012), which is of special interest to the photovoltaic and IC industrials (Hou et al. 2012). As explained in Table 8.4, the features of CPUs and GPUs, respectively, are best utilized in this mode. As a result, for bulk simulation, we have obtained 1.87 Petaflops (single precision) sustained performance on the Tianhe1A supercomputer (www.Top500.org/lists/2010/11), which has 7168 Nvidia M2050 GPUs. That is, the simulation using the multi-body Tersoff potential has reached 25.3 % of its peak performance. In fact, the instruction throughput and memory throughput on a single GPU approached 80 %. When coupled with 86016 CPU cores, the more complicated simulation on surface reconstruction also reached Petaflops sustainable performance (1.17 Petaflops in single precision plus 92 Teraflops in double precision). More than 1000 billion atoms were simulated in this case, which links atomistic behavior with macro-scale material properties.

Table 8.5 Simulation of gas-solid flow in the EMMS paradigm

Simulated system		Physical model		Numerical method		Software algorithm		Hardware architecture
Macro-scale distribution	Global Local	EMMS global EMMS local		Nonlinear equation set Multi-objective optimization		Iteration; GA/ANN/PSO		Single node: CPU+multi-GPU
Meso-scale evolution	Inter-phase	Drag/flow distribution		Interpolation/mapping		Reduce/broadcast		CPU+ inter-layer communication
	Gas phase	Continuum	Above particle scale (incompressible)	Partial differential equation: Implicit finite difference (volume)		Sparse matrix operations		Multi-CPU: Intra-node—shared memory; Inter-node—neighborhood communication
			Below particle scale	Explicit finite difference (volume)	LBM		Particle Interaction & updating	Multi-GPU Additive operation
			Above particle scale (compressible)		Discrete kinetics			Regular adjacent communication
Micro-scale evolution	Solid phase	Discrete	Coarse grain	Ordinary differential equation		Particle searching		Multi-GPU Additive operation
			Soft parcel Deformable cluster	Numerical integration				Irregular local communication
		Hybrid	PIC	Hybrid		Hybrid		Hybrid

8.6 Conclusions

In summary, structural consistency among the hardware, software, model and the system to be simulated is critical for the high efficiency of supercomputing. The continuum-discrete implementation of the so-called EMMS paradigm can take full advantage of the CPU-GPU hybrid computing mode and unprecedented simulation results on multi-phase systems or even beyond have been or can be obtained in this paradigm. The prospects of simulating industrial scale multi-phase systems at almost realtime with reasonable accuracy and resolution, or in short, virtual process engineering is not remote considering the dramatic development of both the hybrid computing mode and its hardware developments.

Acknowledgments We thank all members of the EMMS group at IPE for their long term collaboration and support on this work. This work is sponsored by National Natural Science Foundation of China under the Grant no. 20821092, Ministry of Finance under the Grant no. ZDYZ2008-2, Chinese Academy of Sciences under the Grants nos. KGCX2-YW-124 and KGCX2-YW-222. We also thank Nvidia for sponsoring the CUDA Center of Excellence (CCOE) at IPE.

References

Anderson TB, Jackson R (1967) A fluid mechanical description of fluidized beds: equations of motion. Ind Eng Chem Fundam 6:527–539

Chen F, Ge W, Wang L, Li J (2008) Numerical study on gas-liquid nano-flows with pseudo-particle modeling and soft-particle molecular dynamics simulation. Microfluid Nanofluid 5:639–653

Chen F, Ge W, Guo L, He X, Li B, Li J, Li X, Wang X, Yuan X (2009) Multi-scale HPC system for multi-scale discrete simulation-development and application of a supercomputer with 1 Petaflops peak performance in single precision. Particuology 7:332–335

Dubitzky (2012) Large-scale computing techniques for complex system simulations. Wiley

Gao J, Ge W, Hu G, Li J (2005) From homogeneous dispersion to MicellesA molecular dynamics simulation on the compromise of the hydrophilic and hydrophobic effects of sodium dodecyl sulfate in aqueous solution. Langmuir 21:5223–5229

Ge W, Li J (1996) Pseudo-particle approach to hydrodynamics of gas-solid two-phase flow. In: Kwauk M, Li J (eds) Proceedings of the 5th international conference on circulating fluidized bed. Science Press, Beijing, pp 260–265

Ge W, Li J (2000) Conceptual model for massive parallel computing of discrete systems with local interactions. Comput Appl Chem 17(5): 385–388. (Chinese)

Ge W, Li J (2002) General approach for discrete simulation of complex systems. Chin Sci Bull 47(14):1172–1175

Ge W, Li J (2003) Macro-scale phenomena reproduced in microscopic systems: pseudo particle modeling of fluidization. Chem Eng Sci 58:1565–1585

Ge W, Wang W, Yang N, Li J, Kwauk M, Chen F, Chen J, Fang X, Guo L, He X, Liu X, Liu Y, Lu B, Wang J, Wang J, Wang L, Wang X, Xiong Q, Xu M, Deng L, Han Y, Hou C, Hua L, Huang W, Li B, Li C, Li F, Ren Y, Xu J, Zhang N, Zhang Y, Zhou G, Zhou G (2011) Meso-scale oriented simulation towards virtual process engineering (VPE)-The EMMS paradigm. Chem Eng Sci 66(19):4426–4458

Gidaspow D (1994) Multiphase flow and fluidization: continuum and kinetic theory description. Academic Press, Boston

Harlow FH (1988) PIC and its progeny. Comput Phys Comm 48:1–10

Hou C, Xu J, Wang P, Huang W, Wang X, Shen G, Ge W, He X, Guo L, Li J (2012) Petaflops molecular dynamics simulation of crystalline silicon on Tianhe-1A. Int J High Perform Comput (In print) Doi:10-1177/1094342012456047

Hou C, Xu J, Ge W, Wang P, Huang W, Wang X (2012) Efficient GPU-accelerated molecular dynamics simulation of solid covalent crystals. Comput Phys Comm Accepted

Hou C, Ge W (2011) GPU-accelerated molecular dynamics simulation of solid covalent crystals. Mol Simul 38(1):8–15

Li J, Kwauk M (1994) Particle-fluid two-phase flow: the energy-minimization multi-scale method. Metallurgical Industry Press, Beijing, P. R. China

Li J, Tung Y, Kwauk M (1988) Multi-scale modeling and method of energy minimization in particle-fluid two-phase flow. In: Basu P, Large JF (eds) Circulating fluidized bed technology II. Pergamon Press, Toronto, pp 89–103

Li F, Song F, Benyahia S, Wang W, Li J (2012) MP-PIC simulation of CFB riser with EMMS- based drag model. Chem Eng Sci 82(12):104–113

Li J, Ge W, Kwauk M (2009) Meso-scale phenomena from compromise - a common challenge, not only for, chemical engineering arXiv:0912.5407

Li J, Ge W, Wang W, Yang N, Liu X, Wang L, He X, Wang X, Wang J, Kwauk M (2013) From multiscale modeling to Meso-science - a chemical engineering perspective. Springer (In print), Berlin

Liu X, Ge W, Li J (2008) Non-equilibrium phase transitions in suspensions of oppositely driven inertial particles. Powder Technol 184:224–231

Liu Y, Chen J, Ge W, Wang J, Wang W (2011) Acceleration of CFD simulation of gas-solid flow by coupling macro-/meso-scale EMMS model. Powder Technol 212:289–295

Liu X, Guo L, Xia Z, Lu B, Zhao M, Meng F, Li Z, Li J (2012) Harnessing the power of virtual reality. Chem Eng Prog 108(7):28–33

Ren Y, Gao J, Xu J, Ge Wei, Li Jinghai (2011) Key factors in chaperonin-assisted protein folding. Particuology 10(1):105–116

Sun Q, Ge W, Huang J (2007) Influence of gravity on narrow input forced drainage in 2D liquid foams. Chin Sci Bull 52:423–427

Tang D, Ge W, Wang X, Ma J, Guo L, Li J (2004) Parallelizing of macro-scale pseudo-particle modeling for particle-fluid systems. Sci China Ser B Chem 47(5):434–442

Wang X, Ge W, He X (2010) Development and application of a HPC system for multi-scale discrete simulation-Mole-8.5. In: International supercomputing conference. Hamburg, Germany

Wang X, Ge W (2012) The Mole-8.5 supercomputing system. In: Vetter JS (ed) Contemporary high performance computing: from petascale toward exascale. Taylor and Francis, Boca Raton

Wang X, Guo L, Ge W, Tang D, Ma J, Yang Z, Li J (2005) Parallel implementation of macro-scale pseudo-particle simulation for particle-fluid systems. Comput Chem Eng 29:1543–1553

Wang J, Xu M, Ge W, Li J (2010) GPU accelerated direct numerical simulation with SIMPLE arithmetic for single-phase flow. Chin Sci Bulletin 55:1979–1986

Wang L, Zhang B, Wang X, Ge W, Li J (2012) Lattice Boltzmann based discrete simulation of gas-solid fluidization. Chin Sci Bull, Accepted

Wen CY, Yu YH (1966) Mechanics of fluidization. Chem Eng Progr Symp Ser 62:100–111

Xiong Q, Deng L, Wang W, Ge W (2011) SPH method for two-fluid modeling of particle-fluid fluidization. Chem Eng Sci 66:1859–1865

Xiong Q, Li B, Zhou G, Fang X, Xu J, Wang J, He X, Wang X, Wang L, Li J (2012) Large-scale DNS of gas-solid flows on Mole-8.5. Chem Eng Sci 71:422–430

Xu M, Ge W, Li J (2007) A discrete particle model for particle-fluid flows with considerations of sub-grid structures. Chem Eng Sci 62:2302–2308

Xu J, Ren Y, Ge W, Yu X, Yang X, Li J (2010) Molecular dynamics simulation of macromolecules using graphics processing unit. Mol Simul 36:1131–1140

Xu J, Wang X, He X, Ren Y, Ge W, Li J (2011) Application of the Mole-8.5 supercomputer: probing the whole influenza virion at the atomic level. Chin Sci Bull 56(20):2114–2118

Xu J, Qi H, Fang X, Lu L, Ge W, Wang X, Xu M, Chen F, He X, Li J (2011a) Quasi-real-time simulation of rotating drum using discrete element method with parallel GPU computing. Particuology 9:446–450

Xu M, Chen F, Liu X, Ge W, Li J (2012) Discrete particle simulation of gas-solid two-phase flows with multi-scale CPU-GPU hybrid computation. Chem Eng J 207–208:746–757

Yu H, Luo L-S, Girimaji SS (2006) LES of turbulent square jet flow using an MRT lattice Boltzmann model. Comput Fluids 35:957–965

Zhou G, Ge W, Li J (2010) Smoothed particles as a non-Newtonian fluid: a case study in Couette flow. Chem Eng Sci 65:2258–2262