Multi-GPU Parallelization on SOPHIA Code for Large Scale Simulation and Preliminary Analysis

So-Hyun Park [a], Young Beom Jo [a], Eung Soo Kim [a*], Yong-Woo Sim [b], Dong Jo Ha [b], Dong-Hak Lee [b]
[a]Dept. of Nuclear Engineering. Seoul National Univ. 1 Gwanak-ro, Gwanak-gu, Seoul, Korea
[b]CoCoLink Co., 122 Gwanak-ro, Gwanak-gu, Seoul, Korea
[*]Corresponding author: kes7741@snu.ac.kr

1. Introduction

In the field of nuclear safety, there are many physically complicated phenomena such as internal molten fuel phenomena during severe accident, intricate corium behavior in the reactor vessel, and molten corium concrete interaction. Since these phenomena are difficult to be investigated separately or integrally by experiment, numerical simulation is preferred for a multifaceted analysis.

With the development of Computational Fluid Dynamics (CFD), Eulerian based fixed-grid methods are able to solve most of the thermal hydraulic and hydrodynamic problems. However, the problems including irregular boundaries require additional treatments, such as mesh re-construction and Volume of Fluid (VOF) consuming a high computational cost. Otherwise, Lagrangian based meshless methods have the advantage of solving problems with free surface flow, interfacial flow, or large deformation because the moving mesh or material point representing the fluid moves along the flow stream. One of the Lagrangian-based meshless methods, Smoothed Particle Hydrodynamics (SPH) method represents the fluid as moving particles and solves the system using particle summation.

Using the SPH method, SOPHIA Plus code, the multi-dimensional and multi-physics code, was developed in Seoul national university in order to simulate specific nuclear safety-related phenomena. SOPHIA Plus was based on Weakly Compressible SPH (WCSPH) method that allows a slight compressibility of fluid and the code was implemented on a single GPU. In addition, SPH-formulated models for viscous force, surface tension, multi-phase, heat transfer, turbulence, and elastic solid mechanics were applied [1]. Besides, Incompressible SPH (ISPH) model was also applied, enabling to simulate fully incompressible fluid flow [2]. These models were verified/validated through the simulations of various nuclear safety-related phenomena [1, 2].

However, from the recent simulation experience, the limitation of single GPU was confirmed and the need for multi-GPU had emerged. To improve the accuracy of simulation requires high-resolution simulation and to solve comprehensive phenomena demands large-scale simulation. Nevertheless, the computational resource of a single GPU is insufficient to handle a number of particles of high-resolution and large-scale simulation. Multi-GPU system where the multiple GPUs are executed in parallel, 1) allows memory usage more than single GPU's memory and 2) distribute computational load to multiple GPUs' resources. Therefore, it is necessary to utilize multiple GPUs to overcome the limitation of particle numbers.

The objective of this paper is development of multi-GPU based SOPHIA Plus code for high-resolution and large-scale simulation. First, the previous single-GPU based SOPHIA Plus code was extended to multi-GPU parallel computation code by applying concurrency programming technique. Then, the code acquired capability of handling more than 10 millions particles with six GPUs equipped on PCIe fabric subsystem. Finally, in order to verify the computational performance, three-dimensional dam break simulation using 10 million particles was performed.

2. Multi-GPU Based SOPHIA Code

This section briefly reviews the basic concept of SPH method and SPH-formulated physical models. Next, explains the implementation of multi-GPU concurrent computation and the algorithm of multi-GPU based SOPHIA Plus.

2.1. Basic concept of SPH method

The SPH method is formulated by particle summation approximation using Gaussian-shaped weighting function and discretized fluid particles in a finite computational domain [3]. The weighting function is defined as Smoothing kernel and the finite domain is defined by Smoothing length. Fig.1 shows a schematic diagram of SPH formulation. This formulation can approximate a scalar/vector distribution over the space, but also approximate differential form of them. The following equations present SPH particle-based approximation for the scalar f and gradient of scalar f respectively [3].

$$f(\boldsymbol{r}_i) = \sum_j f_j W(\boldsymbol{r}_i - \boldsymbol{r}_j, h) \frac{m_j}{\rho_j} \tag{1}$$

$$\nabla f(\boldsymbol{r}_i) = \sum_j f_j \frac{m_j}{\rho_j} \nabla W(\boldsymbol{r}_i - \boldsymbol{r}_j, h) \tag{2}$$

Where $m, \rho, \boldsymbol{r}, and\ h$ denote the mass, density, position vector, and smoothing length. The subscript

i *and* j denote center particle and neighbor particle. W in the equation presents smoothing kernel.

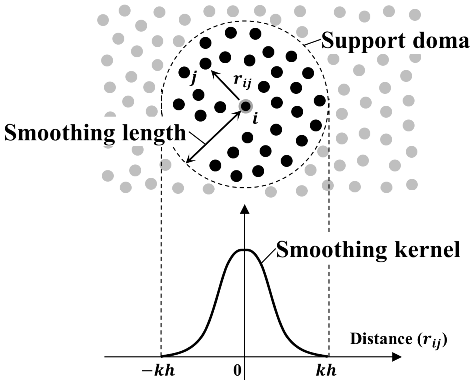


Figure 1. Schematic diagram of SPH formulation

The governing equations for fluid hydrodynamics are mass conservation and momentum conservation.

$$\frac{d\rho}{dt} = -\rho\nabla\cdot\boldsymbol{u} \tag{3}$$

$$\frac{d\boldsymbol{u}}{dt} = -\frac{\nabla p}{\rho} + \nu\nabla^2\boldsymbol{u} + \rho\boldsymbol{g} \tag{4}$$

Where $\boldsymbol{u}, p, \nu,$ *and* $\boldsymbol{g}$ denote velocity vector, pressure, kinematic viscosity, and gravitational acceleration.

Table I presents the SPH-formulated models for each term of above governing equations. The mass conservation is satisfied by 1) mass summation or 2) continuity equation. Especially, SOPHIA Plus applied new density equations using normalized density formulation which is suitable for multi-phase/multi-component fluid flow. These new equations are constructed by substituting density term with normalized density in the conventional SPH density equations. As a result, the new equations not only prevents non-physical pressure noise at fluid interfaces with large density gradient, but also calculates exact density change of the phenomena where the density or mass of fluid changes continuously. Finally, the governing equations are closed by Equation of State (EOS), which is based on Tait's equation. For a detailed description, refer to [1].

Table I. SPH formulated models of SOPHIA Plus code

Model	Formulation
Mass summation	$\left(\frac{\rho}{\rho_0}\right)_i = \sum_j \frac{m_j}{\rho_{0,j}} W_{ij} \qquad (5)$
Continuity	$\frac{d}{dt}\left(\frac{\rho}{\rho_0}\right)_i = -\left(\frac{\rho_i}{\rho_{0,i}}\right)\sum_j \frac{m_j}{\rho_j}\boldsymbol{u}_{ij} \cdot \nabla W_{ij} \qquad (6)$
Pressure force	$\left(\frac{d\boldsymbol{u}}{dt}\right)_i^{fp} = -\sum_j m_j \left(\frac{p_i + p_j}{\rho_i\rho_j}\right)\nabla W_{ij} \qquad (7)$
Laminar viscous force	$\left(\frac{d\boldsymbol{u}}{dt}\right)_i^{fv} = \sum_j \left[\frac{4\mu_i\mu_j}{m_i(\mu_i+\mu_j)} \times \left(\left(\frac{m}{\rho}\right)_i^2 + \left(\frac{m}{\rho}\right)_j^2\right) \times \boldsymbol{u}_{ij}\frac{\boldsymbol{r}_{ij}\cdot\nabla W_{ij}}{\lvert\boldsymbol{r}_{ij}\rvert + \eta^2}\right] \qquad (8)$
EOS	$p_i = \frac{{c_0}^2\rho_{0,i}}{\gamma}\left[\left(\frac{\rho_i}{\rho_{0,i}}\right)^\gamma - 1\right] \qquad (9)$

2.2. Multi-GPU Implementation

Concurrent computation based on Multi-GPU was implemented in two perspectives; a programming perspective and a hardware perspective. In the perspective of programming, the parallelized concurrency was achieved by multi-threading technique (Fig.2.(a)). Dissimilar to processors, threads share code, data, heap memory, but no stack memory [4]. This feature facilitates efficient data transfer without special technique like Message Passing Interface (MPI). In addition, multiple threads are executed independently with separated stack, resulting in straightforward programming. Therefore, multi-threading is suitable for multi-GPU based parallel computing. In the perspective of hardware, the multi-GPU system was equipped with a 'PCIe Fabric Subsystem' (Fig.2.(b)) developed by CoCoLink Corporation (http://www.cocolink.co.kr/). Generally, multi-GPU requires multi-CPU system because the limited CPU's PCIe memory restricts the number of GPU connection to the CPU. However, CoCoLink Corp. developed the PCIe fabric subsystem where up to 20 GPUs share the same PCIe Gen 3.0 domain. This fabric subsystem not only allows multiple GPUs system with a single CPU, but also provides constant Peer-to-Peer (P2P) memory bandwidth for all GPUs in maximum. That is, multi-GPU concurrent programming based on the PCIe fabric subsystem, is possible by multi-threading only without multi-processing or MPI. Fig.2 shows a schematic diagram of multi-thread and multi-GPU system with PCIe fabric subsystem.

2.3. Multi-GPU based algorithm

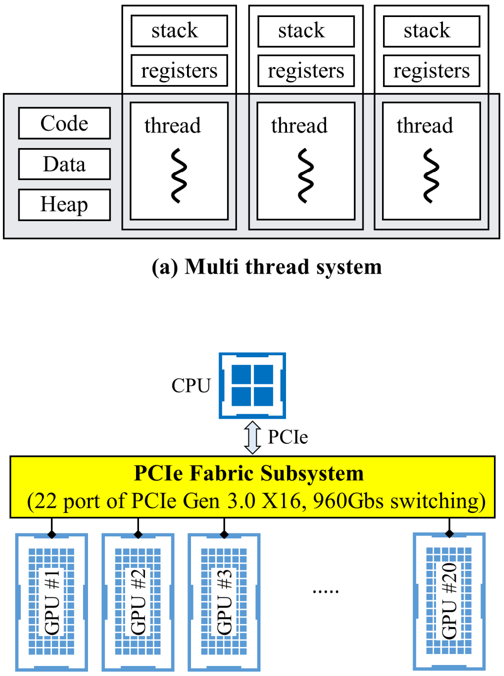
Fig.3 describes algorithm of multi-GPU based SOPHIA Plus and three essential computation steps. First, initialization and multi-GPU allocation steps are performed before entering main loop. Initialization step reads the input files and stores simulation variables through the serial computation of CPU. In multi-GPU allocation step (Fig.3.(b)), the threads are created as many as the number of GPUs using POSIX API, then the computation domain is decomposed and allocated to individual GPUs using threads. In this paper, the domain is divided in the x direction, having equal number of gird cells. This is because one-way partitioning benefits for efficient memory access of the GPUs and for simple definition of the interface between GPUs. After entering the main loop, each GPU performs Nearest Neighbor Particle Search (NNPS) within their subdomains. Then subdomain division step (Fig.3.(c)) is conducted. In this step, in order to efficient data exchange between neighbor GPUs, the subdomain is divided into several regions by the subdomain boundaries and interfaces, and the particles are distinguished by which region they are located in. Next, the governing equations are solved and the particle properties are updated for the next time step. At last, inter-GPU communication step transfers the updated outer particles' data to neighbor GPUs using P2P data copy. Through the inter communication, the particles entering or leaving the subdomain can be included in the proper GPUs respectively. Note that all computations of the algorithm are parallelized with regard to the number of particles as well as the number of GPUs by mapping as in the single GPU based code.

3. High-resolution Simulation

In this study, three dimensional dam break, referred to the experiment [5], was simulated in high resolution using multi-GPU based SOPHIA Plus. The simulation was executed on a server with 6 NVIDIA Tesla P100 and PCIe Fabric Subsystem.

3.1. Simulation model

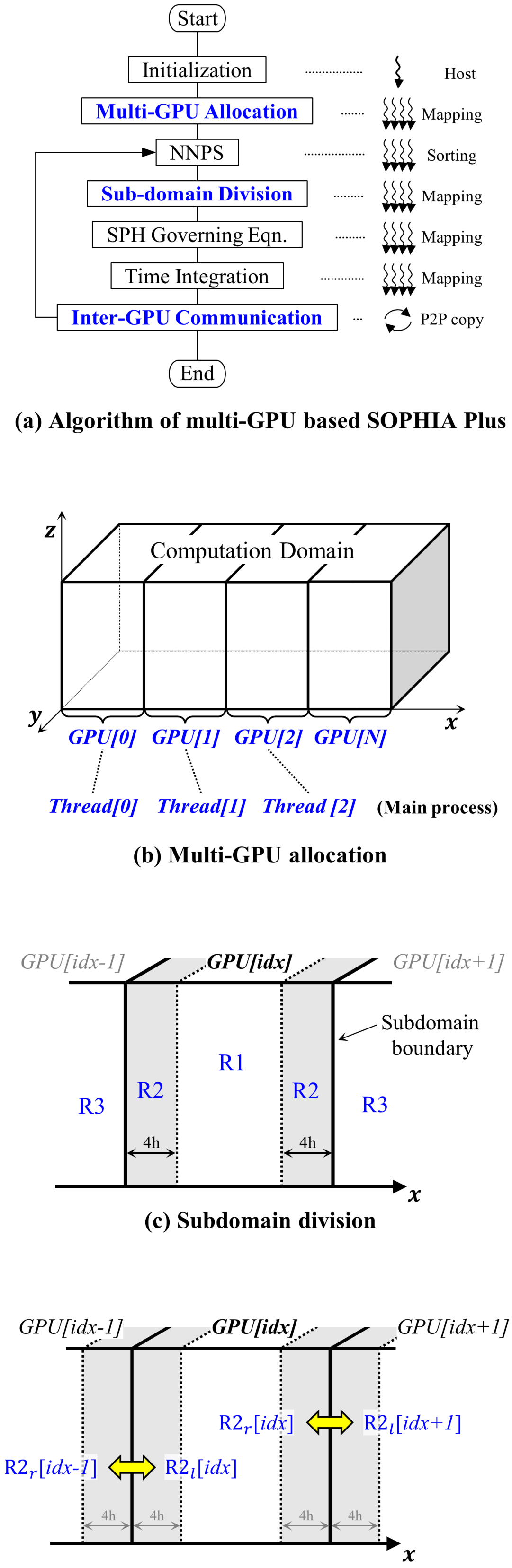
The simulation model consists of a tank, a square column, and water. The tank is 0.61m wide, 1.6m long, and 0.5m high. The tank is filled with the water layer of 10mm depth. The column is 0.21m wide, 0.21m long, and 0.5m high. The water initially trapped in the front side of tank is 0.61m wide, 0.4m long, and 0.3m high. The total 12,673,128 particles were generated using



(a) Multi thread system

(b) Multi-GPU with Fabric Subsystem

Figure 2. Schematic diagram of multi-thread and multi-GPU with PCIe fabric subsystem



(a) Algorithm of multi-GPU based SOPHIA Plus

(b) Multi-GPU allocation

(c) Subdomain division

(d) Inter-GPU communication

Figure 3. Algorithm of multi-GPU-based SOPHIA Plus and three essential computation steps

MATLAB. These particles have the identical properties: Reference density of 1000 kg/m^3, and dynamic viscosity of 0.001 Pas. The time step is 10^{-5} sec and simulation time is 4.0 sec.

3.2. Result

Fig.4 presents the surface flow motion of dam break with rendering. The fluid confined in the gate began to collapse toward the solid column and it reached the column at t=0.3sec and the opposite side wall at t=0.6sec. The fluid striking the solid column lost the momentum, hence the wall-reaching fluid showed relatively gentle uprush motion. The collision of returning fluid with column caused energy loss again, leading to rapid decay of wave propagation.

As a result, it was confirmed that multi-GPU based SOPHIA Plus handled the number of particles more than the capacity of single GPU. Furthermore, the uniform data transfer bandwidth was confirmed to be ensured by the hardware system, algorithm and performance optimization implemented in this paper.

4. Conclusion

This paper presents the development of multi-GPU based SOPHIA Plus code that can perform high-resolution and large-scale simulation requiring a large number of particles than single GPU's capacity in order to increase the accuracy of simulation and to deal with extensive nuclear safety-related phenomena. For an implementation of multi-GPU parallel computing, two techniques were introduced in the existing single-GPU-based code. One is concurrency programming using multi-threading, and the other is parallelized SPH algorithm for multi-GPU. First, concurrent computing resources were constructed by POSIX pthread in the PCIe fabric subsystem-equipped hardware system. Next, the multi-GPU-based parallelized algorithm was developed by adding three steps to the single-GPU based algorithm; 1) Multi-GPU allocation, 2) Subdomain

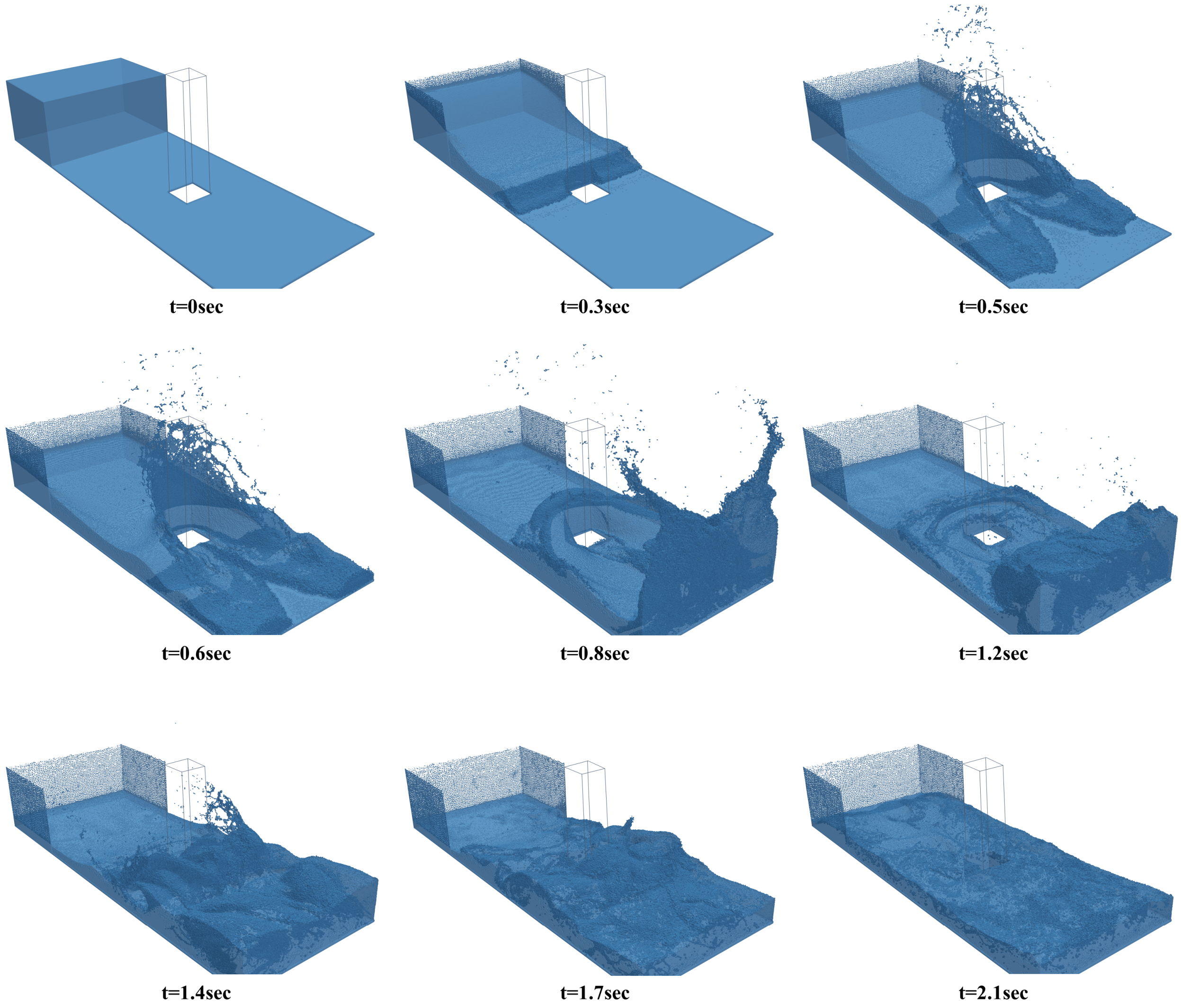


Figure 4. Surface flow motion of dam break with rendering

division, and 3)inter-GPU communication. These steps not only distribute the computational load to individual GPUs by decomposing the computation domain, but also achieve the high accuracy at the subdomain boundaries by satisfying compact support domain at the boundaries. In this study, high-resolution dam break simulation was performed with more than 10 million particles to verify the performance of multi-GPU based code. As a result, it was confirmed that the code stably handled a large number of particles more than single GPU' capacity and showed optimized computational performance for multi-GPU without exceptional bottleneck. Therefore, multi-GPU based SOPHIA Plus is expected to provide a detailed analysis on the nuclear safety-related phenomena through the simulations such as high-resolution LMR sloshing or large-scale tsunami propagation.

ACKNOWLEDGEMENT

This research was supported by National R&D Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No. 2016M2B2A9911846) and Nuclear Energy Technology Development Program (U.S.-ROK I-NERI Program) through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (2019M2A8A1000630).

REFERENCES

[1] Jo, Young Beom, et al. "SOPHIA: Development of Lagrangian-based CFD code for nuclear thermal-hydraulics and safety applications." Annals of Nuclear Energy 124 (2019): 132-149.

[2] Park, So-Hyun, et al. "Simulation of a laboratory-scale experiment for wave propagation and interaction with a structure of undersea topography near a nuclear power plant using a divergence-free SPH." *Annals of Nuclear Energy* 122 (2018): 340-351.

[3] Monaghan, Joe J. "Smoothed particle hydrodynamics." *Annual review of astronomy and astrophysics* 30.1 (1992): 543-574.

[4] Forbes, Elliot. "Learning Concurrency in Python: Build highly efficient, robust, and concurrent applications." (2017).

[5] Arnason, Halldor, Catherine Petroff, and Harry Yeh. "Tsunami bore impingement onto a vertical column." *Journal of Disaster Research* 4.6 (2009): 391-403.