

Performance Evaluation of Tsunami Inundation Simulation on SX-Aurora TSUBASA

Akihiro Musa^{1,2}, Takashi Abe¹, Takumi Kishitani¹, Takuya Inoue^{1,3}, Masayuki Sato¹, Kazuhiko Komatsu¹, Yoichi Murashima^{1,3}, Shunichi Koshimura¹, and Hiroaki Kobayashi¹

¹ Tohoku University, Sendai, Miyagi 908-8578, Japan
{musa@, t-abe@irides., takumi.kishitani.p6@dc., masa@, komatsu@, koshimura@irides., koba@}tohoku.ac.jp

² NEC Corporation, Minato-ku, Tokyo 108-8001, Japan

³ Kokusai Kogyo Co.,LTD., Fuchu-shi, Tokyo 183-0057, Japan
{takuya.inoue, yoichi.murashima}@kk-grp.jp

Abstract. As tsunamis may cause damage in wide area, it is difficult to immediately understand the whole damage. To quickly estimate the damages of and respond to the disaster, we have developed a real-time tsunami inundation forecast system that utilizes the vector supercomputer SX-ACE for simulating tsunami inundation phenomena. The forecast system can complete a tsunami inundation and damage forecast for the southwestern part of the Pacific coast of Japan at the level of a 30-m grid size in less than 30 minutes. The forecast system requires higher-performance supercomputers to increase resolutions and expand forecast areas. In this paper, we compare the performance of the tsunami inundation simulation on SX-Aurora TSUBASA, which is a new vector supercomputer released in 2018, with those on Xeon Gold and SX-ACE. We clarify that SX-Aurora TSUBASA achieves the highest performance among the three systems and has a high potential for increasing resolutions as well as expanding forecast areas.

Keywords: system performance · supercomputer · tsunami simulation.

1 Introduction

Tsunamis are a natural disaster that may cause many fatalities and extensive damage of infrastructures in coastal areas. In 2011, the 2011 Tohoku earthquake tsunami, which was generated by a 9.0-magnitude earthquake, struck the northeastern part of the Pacific coast of Japan. Its inundated area ranged up to 2000 km along the Japanese coastlines and spread out over 500 km² [4, 13]. The number of fatalities was more than 19,000, and many houses and infrastructures were destroyed [4, 18]. Since the damaged area was widespread, the Japanese government was not able to grasp the entire picture of the damage shortly after the tsunami. Under this situation, it was difficult to promptly cope with the disaster [11]. Therefore, a forecast system that can quickly estimate the damage

of a tsunami inundation after an earthquake is required worldwide in order to provide an early understanding of damage situations.

We have developed a real-time tsunami inundation forecast system using the vector supercomputer SX-ACE [15] to quickly estimate tsunami inundations and infrastructure damages [9]. The forecast system is automatically activated by a large earthquake that may generate a tsunami, simulates the tsunami inundation, and finally visualizes the results of the damage estimations on a Web site. The total processing time is within 20 minutes for a coastal city region at the resolution of a 10-m grid size [7, 15]. The forecast system has also been implemented as the disaster information system of the Cabinet Office of Japan, however, the forecast area is needed to be expanded from a city region to the southwestern part of the Pacific coast of Japan, resulting in the coastal length of 5,500 km. Therefore, in the implementation for the Cabinet Office, the grid size and the total processing time are changed from 10 m to 30 m and from 20 minutes to 30 minutes due to the processing time of the SX-ACE system. In addition, since the entire Pacific coast of Japan has the high probability of damages by large tsunamis, it is necessary to expand the forecast areas. Moreover, the resolution of a 10-m grid size is required for accurate forecasting. Therefore, we need performance improvements of the tsunami inundation simulation in order to improve disaster prevention and mitigation by expanding forecast areas and increasing resolutions.

Modern supercomputers consist of multi-nodes, each of which is equipped with multi-core processors. These processors have recently employed vector (SIMD) instructions to further exploit loop-level parallelism. Intel Corporation released a Xeon Phi processor (*Knights Landing*) in 2016 and a Xeon processor (*Skylake*) in 2017, and NEC Corporation developed an SX-Aurora TSUBASA processor in 2018. We evaluated the performance of our tsunami inundation on the Xeon Phi processor, and found that Xeon Phi needs optimizations for vectorization to achieve a high performance [14]. However, since the performance of Xeon Phi did not exceed that of SX-ACE, the forecast system cannot increase resolutions and expand forecast areas by Xeon Phi. In this paper, we discuss the performance of the tsunami inundation simulation on the latest vector supercomputer SX-Aurora TSUBASA [20] and Xeon processor (*Skylake*) and the possibilities to increase the forecast resolutions and areas.

The rest of this paper is organized as follows. In Section 2, we present related work on performance evaluations using supercomputers. Section 3 explains the tsunami inundation simulation. Section 4 describes the experimental setup of the performance evaluation, which includes an overview of SX-Aurora TSUBASA. Section 5 presents the results of the performance evaluations. We conclude this paper in Section 6 with a brief summary and mention of future work.

2 Related Work

A real-time tsunami forecast system requires reduction in execution time in order to provide adequate disaster prevention and mitigation. To this end, there

have been studies on reducing the execution time of tsunami simulations using high-performance computing. Christgau et al. [3] evaluated the performance of Intel Xeon Phi (*Knights Corner*), Xeon (*Ivy Bridge*), and NVIDIA Tesla K40m GPGPU (general-purpose computing on graphics processing units) using the tsunami simulation EasyWave, which is a TUNAMI-F1 model [6]. They optimized the program to increase the reusability of cache data. Specifically, the computer domain was divided into smaller blocks that fit into the last level cache on Xeon and Xeon Phi. The program was also vectorized by the Intel compiler. They parallelized it using OpenMP on Xeon and MPI on XeonPhi, and the program has a CUDA version for GPGPU. They compared the performances of the program with an $18,001 \times 16,301$ grid on Xeon, Xeon Phi, and Tesla K40m. The performance of the single Xeon processor (10 cores) was 1.23 and 1.8 times higher than those of the single Xeon Phi processor (61 cores) and the single Tesla K40m GPGPU (2880 cores), respectively. They found that the tsunami simulation EasyWave was memory intensive and required the load reduction of the memory to achieve higher performance, and the vector processing on Xeon and Xeon Phi was useful for accelerating the tsunami simulation. We will discuss the potential of vector processing and high memory bandwidth by comparing the performances of vector supercomputers, SX-Aurora TSUBASA and SX-ACE, as well as Xeon Gold in this paper.

Nagasu et al. [16] developed a field-programmable gate array (FPGA)-based tsunami simulation system using the method of splitting tsunami (MOST) model, which solves shallow water equations for ocean-wide tsunami propagation. The machine contained an ALTERA Arria 10 FPGA, two DDR3 PC12800 SDRAMs, and a PCI-Express Gen2 interface. They evaluated the performance and power consumption of the tsunami propagation simulation with a 2581×2879 grid in the Pacific Ocean on the FPGA system, NVIDIA Tesla K20c, and AMD Radeon R9 280X GPGPU. The sustained performances were 383.0 Gflop/s on the FPGA system, 44.6 Gflop/s on Tesla K20c and 219.2 Gflop/s on Radeon R9 280X, respectively. The performance per power on the system was 17.2 and 7.1 times higher than those on Tesla K20c and Radeon R9 280x, respectively. The research demonstrated the significance of power efficiency. However, their resolutions were several kilometers and the tsunami inundation behavior was not simulated. This paper conducts the tsunami inundation simulation with higher resolutions that is necessary for the forecast system.

Acuña et al. [1] developed a non-linear spherical shallow water model to compute tsunami propagation across an ocean on GPGPU. The model utilized the tree-based mesh refinement method to generate a computational mesh and was used to replicate the 2004 Indonesia tsunami off the Indian Ocean ($1,500 \text{ km} \times 500 \text{ km}$) with grid sizes from 1852 m to 50 m. The execution time for ten hours of tsunami behaviors was 8.19 minutes on eight NVIDIA Tesla P100 GPGPUs. These results indicated that their model had a potential for real-time tsunami forecasting using GPGPU. However, the area of the 50-m grid size was a city region, and the total number of grid points was 230,000. Meanwhile, we

will evaluate the performance of our tsunami inundation simulation using the grid points of 3.5×10^7 : it is 150 times larger than that of their model.

3 Tsunami Inundation Simulation

Our tsunami inundation simulation is based on the TUNAMI model [10] that solves non-linear shallow water equations with the staggered leap-frog finite difference method. The governing equations are

$$\frac{\partial \eta}{\partial t} + \frac{\partial M}{\partial x} + \frac{\partial N}{\partial y} = 0, \quad (1)$$

$$\begin{aligned} \frac{\partial M}{\partial t} + \frac{\partial}{\partial x} \left(\frac{M^2}{D} \right) + \frac{\partial}{\partial y} \left(\frac{MN}{D} \right) + gD \frac{\partial \eta}{\partial x} \\ + \frac{gn^2}{D^{7/3}} M \sqrt{M^2 + N^2} = 0, \end{aligned} \quad (2)$$

$$\begin{aligned} \frac{\partial N}{\partial t} + \frac{\partial}{\partial x} \left(\frac{MN}{D} \right) + \frac{\partial}{\partial y} \left(\frac{N^2}{D} \right) + gD \frac{\partial \eta}{\partial y} \\ + \frac{gn^2}{D^{7/3}} N \sqrt{M^2 + N^2} = 0, \end{aligned} \quad (3)$$

where η is the vertical displacement of the water surface, D is the total water depth, g is the gravitational acceleration, n is Manning's roughness coefficient, x and y are the horizontal directions. M and N are the discharge fluxes in the x and y directions, respectively. M and N are given by

$$M = \int_{-h}^{\eta} u dz = \bar{u}(\eta + h), \quad (4)$$

$$N = \int_{-h}^{\eta} v dz = \bar{v}(\eta + h), \quad (5)$$

where h is the still water depth. u and v are the water velocities in the x and y directions, respectively. \bar{u} and \bar{v} are the average water velocities in the x and y directions, respectively. The tsunami source is calculated using Okada's equation [19], and the damage estimation is based on a study by Koshimura et al. [10].

The tsunami inundation simulation leverages a nested grid system to simulate the multiscale nature of a tsunami. The computational domain is a polygonal domain [5], which can reduce the number of computations in the simulation [14]. Also, the system of the Cabinet Office simulates the coastal length of 5,500 km, and the simulated area is divided into fifteen subareas, and each subarea is individually simulated because the geographical coordinates of the subareas are different and a different tidal level is used for each region.

4 Experimental Setup for Performance Evaluation

We evaluate the performance of the tsunami inundation simulation by using SX-Aurora TSUBASA, SX-ACE, and Xeon Gold. Table 1 lists the specifications

Table 1: Specifications of systems used in the evaluation

		SX-Aurora TSUBASA	SX-ACE	Xeon Gold
Clock freq. (GHz)		1.4	1.0	2.6
Core	Performance (Gflop/s)	537.6 (SP) 268.8 (DP)	64 (SP/DP)	147.2 (SP) 73.6 (DP)
	Memory bandwidth (GB/s)	358	256	N/A
	B/F	0.67 (SP)	4 (SP/DP)	N/A
		1.33 (DP)		
Processor No. of cores		8	4	12
Node	Performance (Tflop/s)	4.3 (SP) 2.15 (DP)	0.256 (SP/DP)	1.77 (SP) 0.86 (DP)
	LLC (MB)	16 (shared)	1 (private)	19.5
	LLC bandwidth (GB/s)	2854	1000	N/A
	B/F of LLC	0.67 (SP)	4 (SP/DP)	N/A
		1.33 (DP)		
	Memory (GB)	48	64	96
	Memory bandwidth (GB/s)	1220	256	128
	B/F of memory	0.28 (SP)	1 (SP/DP)	0.07 (SP)
0.56 (DP)				
No. of processors		8	1	2
Memory (GB)		384	64	192
Memory bandwidth (GB/s)		9760	256	256
Network (GB/s)		25	8	12.5

of each system. Here, SP and DP indicate single precision and double precision floating-point operations, respectively.

4.1 SX-Aurora TSUBASA

SX-Aurora TSUBASA is a new vector supercomputer released in 2018 [8, 20]. It consists of one or more card-type vector engines (VEs) and a vector host (VH). The VE is the main part of SX-Aurora TSUBASA and contains a vector processor, a 16 MB shared last-level cache (LLC), and six HBM2 memory modules, as shown in Fig. 1. The vector processor has eight vector cores and two types of the clock frequency: 1.6 GHz and 1.4 GHz. In this evaluation, we use the VE of 1.4 GHz. The peak performances of single precision floating-point operations on a vector core and a vector processor are 537.6 Gflop/s and 4.3 Tflop/s, respectively. The memory bandwidths are 358 GB/s per vector core and 1.22 TB/s per vector processor. The ratios of the memory bandwidth to the floating-point operation per second, so called the Bytes/Flop ratio (B/F), on the vector core and the vector processor are 0.67 and 0.28, respectively. The VH is a standard x86 Linux server and executes an operating system (OS) of the VE. While the conventional SX series runs an OS on a vector processor, SX-Aurora TSUBASA offloads OS-related processing on the VH since the VH is suitable for scalar processing such as the OS-related processing. Figure 2 shows the maximum configuration of one VH that contains eight VEs, two Infiniband interfaces, and two

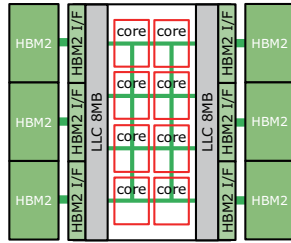


Fig. 1: Block diagram of a vector engine

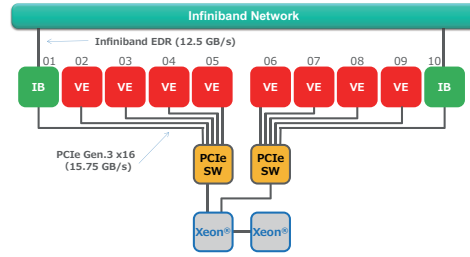


Fig. 2: Block diagram of SX-Aurora TSUBASA system

Xeon processors. Moreover, SX-Aurora TSUBASA can compose a large system by connecting the VHs via the Infiniband switch.

The SX-Aurora TSUBASA system supports Red Hat Enterprise Linux and CentOS as the OS, and the VH provides OS functions such as process scheduling and handling of system calls invoked by the application running on the VEs. The development environment is NEC MPI and NEC Software Development Kit for Vector Engine, which contains NEC Fortran compiler, NEC C/C++ compiler, NEC Numeric Library Collection, NEC Parallel Debugger, and NEC Ftrace Viewer. The Fortran and C/C++ compilers support functions of automatic optimization, automatic vectorization, automatic parallelization, and OpenMP. NEC MPI is implemented in accordance with MPI-3.1.

4.2 SX-ACE

SX-ACE, which is also a vector supercomputer, was released in 2014. It consists of one vector processor and a 64 GB memory [12]. The vector processor has four vector cores, each of which is comprised of a scalar processing unit, a vector processing unit, and an on-chip private cache with 1MB. The peak performances on a vector core and a vector processor are 64 Gflop/s and 256 Gflop/s, respectively. The memory bandwidth is 256 GB/s per vector core and per vector processor. The B/F ratios of a vector core and a vector processor are 4.0 and 1.0, respectively. The SX-ACE system is composed of 512 nodes connected via a custom interconnect network (called IXS) by an 8 GB/s bandwidth.

The OS of the SX-ACE is SUPER-UX, which is based on the UNIX System V with extensions for performance and functionality. The FORTRAN compiler for SX-ACE, FORTRAN90/SX, has functions of automatic optimization, automatic vectorization, automatic parallelization and OpenMP. The MPI Library, MPI2/SX, implements the Message Passing Interface specifications MPI-1.3 and MPI-2.1.

4.3 Xeon Gold

The Xeon Gold system consists of two Intel Xeon Gold 6126 processors (released in 2017) with a 192 GB memory and 128 GB/s memory bandwidth. The

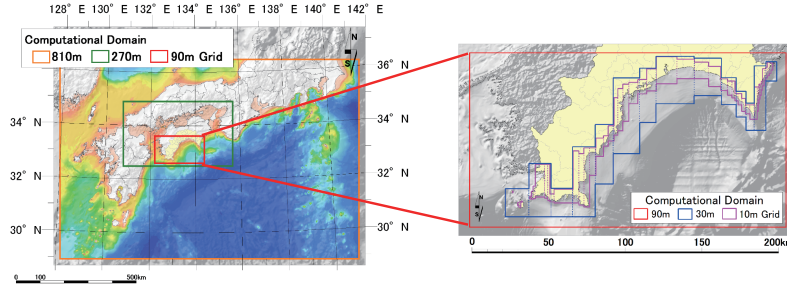


Fig. 3: Computational domains for Kochi prefecture

Table 2: Computational domain

Domain	Grid size (m)	Number of grid points
1	810	1.6×10^6
2	270	1.7×10^6
3	90	2.9×10^6
4	30	6.0×10^6
5	10	22.8×10^6
Total		3.5×10^7

processor contains 12 cores and a shared last-level cache of 19.5 MB. The peak performances of single precision floating-point operations on a core and a processor are 147.2 Gflop/s and 1766.4 Gflop/s, respectively. The B/F of a processor is 0.07. Each node is connected with an InfiniBand EDR at 12.5 GB/s.

The OS of the Xeon Gold system is Red Hat Enterprise Linux, and the programming environment is Intel Parallel Studio XE Cluster Edition, which contains Fortran and C/C++ compilers and an MPI library. The Fortran compiler has the functions of automatic optimization, automatic vectorization, automatic parallelization, and OpenMP.

4.4 Evaluated Model and Compiler Options

We evaluate the performances of a tsunami inundation simulation in the Kochi prefecture region in Japan with a 10-m grid size, as shown in Fig. 3. This coastal length is about 720 km, and this region is one of the fifteen subareas in the system of Cabinet Office of Japan. Table 2 lists the specifications of the model grid sizes and the number of grid points in each domains. *Domains 4 and 5* refer to the polygonal domains. The simulation case is the tsunami generated by the 9.0-magnitude Nankai Trough earthquake [2], and tsunami inundation behaviors are calculated with the time interval of 0.2 seconds for six hours after the earthquake. Here, the simulation is executed with a single precision floating-point operations. Figure 4 shows the simulation results of the maximum tsunami inundation depth and structural damage. In the traditional tsunami forecast, the items of prediction information were tsunami arrival times and tsunami heights

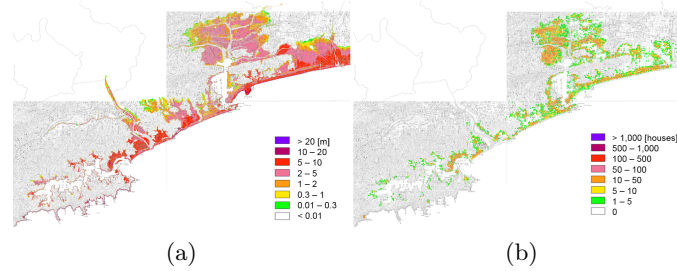


Fig. 4: Simulation results. *a* Maximum tsunami inundation depth. *b* Structural damage (houses)

Table 3: Fortran compiler’s options and directives on each system

System	Version	Options	Directives
SX-Aurora TSUBASA	Fortran compiler for	-O3 -msched-block	-
	Vector Engine 1.5.7	-finline-functions	
SX-ACE	FORTTRAN 90/SX Rev.535	-Chopt -pi	-
Xeon Gold	Intel Fortran 18.0.3.222	-O3 -ipo -xCORE-AVX512	vector always

on coastlines. But our forecast system can estimate tsunami inundations and infrastructure damages of coastal land areas.

Table 3 lists Fortran compiler’s versions, options, and directives on each system. The options, *-Chopt*, *-O3*, and *-msched-block* are high-level optimization options, and *-finline-functions*, *-pi*, and *-ipo* are inline options. As Xeon Gold is unable to vectorize two do-loops, which are time-consuming do-loops, just with the compiler options, we insert the *vector always* directive into the simulation code. *-xCORE-AVX512* also creates the object code including the AVX512 operations.

5 Performance Results

5.1 Single Core Performance

Figure 5 shows the execution times of a single core on SX-Aurora TSUBASA, SX-ACE and Xeon Gold. The CPU time is the running time of the CPU. The memory time indicates the stall time of the CPU due to data loads from the memory. *Directive* and *Non-directive* indicate the cases with and without the vectorization directive on Xeon Gold, respectively. The vectorization directive *vector always* can vectorize two time-consuming do-loops, which take up about 80 % of the total execution time even when the loops are vectorized. The CPU time of the *directive* case is less than half that of the *non-directive* case. This result shows that the vector processing is key in the achievement of a high performance on the Xeon Gold processor.

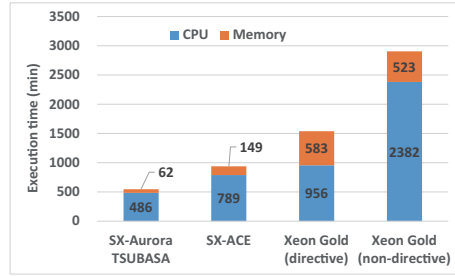


Fig. 5: Execution times on a single core. *CPU*: running time of CPU, *Memory*: Stall time of CPU due to data loads

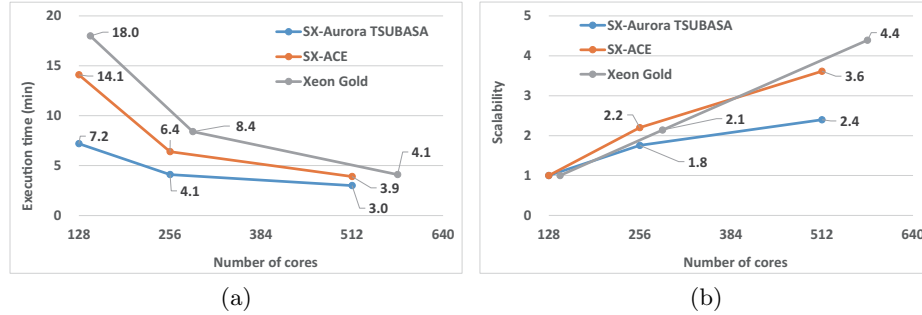
Table 4: Code B/F and actual B/F on SX-ACE and SX-Aurora TSUBASA

Code B/F	Actual B/F (SX-Aurora TSUBASA)	Actual B/F (SX-ACE)
1.8	0.48	0.66

The execution times of both the CPU and memory on SX-Aurora TSUBASA is the shortest among all the systems. SX-Aurora TSUBASA can calculate the tsunami inundation simulation in 58% and 36% of the execution times of SX-ACE and Xeon Gold (directive), respectively. The memory time on SX-Aurora TSUBASA is 40% and 11% of those on SX-ACE and Xeon Gold, respectively. This is because SX-Aurora TSUBASA has the highest LLC bandwidth among all the systems. Moreover, the memory accesses on SX-Aurora TSUBASA can be hidden by vector arithmetic operations since SX-Aurora TSUBASA has long vector pipelines whose vector length is 256. SX-ACE can also hide them and has 61% of the execution time of Xeon Gold (directive). The memory time on SX-ACE is 25% of that on Xeon Gold (directive).

To analyze the performance using the B/F ratios, we use two metrics: a code B/F and an actual B/F. The code B/F is defined as the necessary data in bytes divided by the floating-point operations in a code. The numbers of memory operations and floating-point operations are counted from the simulation code. Since a processor contains a cache memory, the amount of data from a memory is smaller than that of the necessary data. Thus, we utilize the actual B/F calculated from the numbers of block memory accesses and the floating-point operations in the actual behavior of a processor when the simulation is executed. These execution statistics on SX-Aurora TSUBASA and SX-ACE are obtained by using the NEC profiler tool *proginf* [17]. Table 4 shows the code B/F and actual B/F on SX-Aurora TSUBASA and SX-ACE. The code B/F of 1.8 is reduced to 0.48 and 0.66 of the actual B/F on SX-Aurora TSUBASA and SX-ACE, respectively. This result shows that the cache memory can reduce the memory loads of the tsunami inundation simulation on these processors.

The B/F of a vector core on SX-Aurora TSUBASA is 0.67, which is smaller than the code B/F. This indicates that the tsunami inundation simulation can be considered a memory-bandwidth-limited program for SX-Aurora TSUBASA.

Fig. 6: Multi core performance. *a* Execution times. *b* Scalability

In addition, since the actual B/F of 0.48 is smaller than the B/F of 0.67, the bandwidth between the LLC and the memory should be sufficient. Therefore, the bandwidth between the vector core and the LLC (LLC bandwidth) poses a performance bottleneck, and the performance of the simulation is limited by the performance of the LLC. Although the LLC bandwidth becomes a bottleneck on SX-Aurora TSUBASA, the execution time is the shortest among all the systems. This is because the bandwidth of the LCC on SX-Aurora TSUBASA is the highest.

On SX-ACE, the B/F of the vector core is 4.0, which is greater than the code B/F. Therefore, the tsunami inundation simulation can be considered a computation-bound program for SX-ACE and does not require the full memory bandwidth. Although the peak performance of the core is a mere 64 Gflop/s, the CPU time is shorter than that of Xeon Gold (directive). Here, the calculation performance is 12.8 Gflop/s and the computational efficiency reaches 20%. This is because the B/F of the vector core on SX-ACE is high and the vector core can efficiently execute the simulation.

In the case of Xeon Gold, as the bandwidth of core is not officially announced, we assume that one core can utilize the full memory bandwidth of 128 GB/s. Its B/F of the core is 0.87 and smaller than the code B/F. Therefore, the tsunami inundation simulation can be considered a memory-bandwidth-limited program for Xeon Gold, and the bandwidth of the core or memory causes a performance bottleneck. As the memory system of Xeon has the lowest performance among three systems, the execution time on Xeon becomes the longest.

From these evaluations and analyses, we clarify that SX-Aurora TSUBASA has the shortest execution time among all the systems due to its high LLC bandwidth. This performance has a possibility to improve the functions of the forecast system.

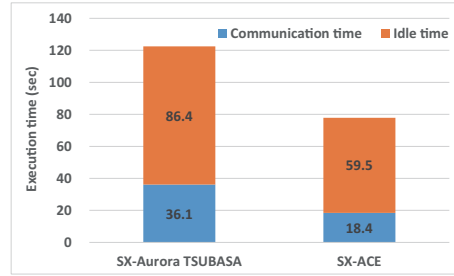


Fig. 7: MPI communication and idle times with 512 cores on SX-Aurora TSUBASA and SX-ACE

5.2 Multi-Core Performance

We evaluate the parallel performances of the tsunami inundation simulation using 512 cores of SX-Aurora TSUBASA and SX-ACE, and 576 cores of Xeon Gold. The peak system performances of single precision floating-point operations on the SX-Aurora TSUBASA, SX-ACE, and Xeon Gold systems are 275.3 Tflop/s, 32.8 Tflop/s, and 84.8 Gflop/s, respectively. Figure 6a shows the execution times of the tsunami inundation simulation. The execution time of SX-Aurora TSUBASA is the shortest time among all the systems due to the highest-core sustained performance. Specifically, it takes three minutes on 512 cores, which means 64 VEs and eight nodes. Figure 6b depicts the scalability of each system. SX-ACE and Xeon Gold achieve super-scalability when the numbers of cores are 256 on SX-ACE, and 288 and 576 on Xeon Gold. This is due to a load imbalance of the tsunami inundation simulation that utilizes the polygonal domain. The polygonal domain has the characteristic of variation in the number of grid points in each MPI process. Specifically, since the variation in the number of grid points per MPI process at small parallel number is larger than that at large parallel number, the load imbalance at small parallel number is large. Therefore, the performances at 128 and 144 cores are low and the super-scalability is generated. Figure 6b also shows that the scalability on SX-Aurora TSUBASA is the lowest among all the systems. The performance of 512 cores on SX-Aurora TSUBASA is 2.4 times higher than that on 128 cores. Meanwhile, the 512 cores on SX-ACE has a 3.6 times higher performance of 128 cores, and the performances of 576 cores on Xeon Gold is 4.4 times higher than that of 144 cores. This is because the network bandwidth per core between the nodes on SX-Aurora TSUBASA is lower than those on SX-ACE and Xeon Gold. The peak network bandwidths per core are 0.4 GB/s on SX-Aurora TSUBASA, 2 GB/s on SX-ACE, and 0.5 GB/s on Xeon Gold. Figure 7 shows the MPI communication times and idle times of communications in the cases of 512 cores on SX-Aurora TSUBASA and SX-ACE. The communication time on SX-Aurora TSUBASA is 1.96 times longer than that on SX-ACE, and the idle time with SX-Aurora TSUBASA has a longer idle time than with SX-ACE.

We set a criterion of the execution time of the tsunami inundation simulation for the forecast system as five minutes in order to terminate the total process-

ing from system start-up to visualization of the results within 30 minutes. In this evaluation, SX-Aurora TSUBASA, SX-ACE, and Xeon Gold achieve this time on 200 cores (25 processors), 384 cores (96 processors), and 492 cores (41 processors), respectively. In the case of the system used by the Cabinet Office of Japan, the southwestern part of the Pacific coast of Japan is now simulated with the 30-m grid size. When the grid size is the 10-m size, the number of grid points increases by a factor of 9, and the time step is supposed to be one third of the 30-m size case, thus the overall calculation amount is 27 times larger than that of this evaluation case. Moreover, when the forecast system covers the entire Pacific coast of Japan, we assume that length of the coast becomes triple, and thus the calculation amount is also about 81 times larger than that of this evaluation case. Table 5 lists the number of processors required for the forecast system with the 10-m grid size within five minutes. SX-Aurora TSUBASA can build the system by using the smallest number of processors. The numbers of nodes are 85 and 254 nodes in the cases of the southwestern part of the Pacific coast and entire Pacific coast of Japan, respectively. The forecast system for the entire Pacific coast of Japan with the 10-m grid size is feasible by SX-Aurora TSUBASA. Therefore, SX-Aurora TSUBASA has a high potential for increasing resolution and expanding forecast areas for the forecast system.

Table 5: Number of processors required for the forecast system

Case	SX-Aurora TSUBASA	SX-ACE	Xeon Gold
Kochi prefecture	25	96	41
Southwestern part of the Pacific coast of Japan	675	2,592	1,107
Entire Pacific coast of Japan	2,025	7,776	3,321

6 Summary

We have developed a real-time tsunami inundation forecast system in order to provide an early understanding of tsunami damage situations. Our system has been operated as a Japanese disaster system on the southwestern part of the Pacific coast of Japan. In order to increase the resolutions of simulation and expand the forecast areas, higher-performance supercomputers are required. In this paper, we evaluate the performance of our tsunami inundation simulation using SX-Aurora TUBASA, SX-ACE, and Xeon Gold. SX-Aurora TSUBASA has the highest sustained performance among all the systems. Specifically, the core performance of SX-Aurora TSUBASA is 1.7 and 2.8 times higher than those of SX-ACE and Xeon Gold, respectively. SX-Aurora TSUBASA can simulate a tsunami inundation in Kochi prefecture with the 10-m grid within five minutes using 200 cores. In the same scenario, SX-ACE and Xeon Gold need 384 cores and 492 cores, respectively. We found that SX-Aurora TSUBASA has the high potential to simulate the entire Pacific coast of Japan with the required time and resolution constrains. However, the performance of SX-Aurora TSUBASA is limited by the LLC bandwidth. The network bandwidth on SX-Aurora TSUBASA is also lower than those on SX-ACE and Xeon Gold.

In future work, we plan to reevaluate the performance of the multi-core environment by changing the domain decomposition for enhancement of the parallel performance in the polygonal domain method [5]. We will also implement the tsunami inundation simulation in general-purpose computing on graphics processing units (GPGPU) and investigate the performance characteristics of the GPGPUs.

Acknowledgments

The authors would like to thank Mr. Takayuki Suzuki and Mr. Atsushi Tanobe of Kokusai Kogyo Co., LTD. for developing the tsunami model. This study was partially supported by JSPS KAKENHI Grant Numbers 16K12845, 17H06108, 18K11322, JST CREST Grant Number JP-MJCR1411, Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures in Japan (Project ID: jh180040-NAH), and Ministry of Education, Culture, Sports, Science and Technology of Japan (Next Generation High-Performance Computing Infrastructures and Applications R&D Program). In this work, we used SX-ACE of the Cyberscience Center at Tohoku University, and SX-ACE and Octopus of the Cybermedia Center at the Osaka University.

References

1. Arce Acuña, M., Aoki, T.: Tree-based mesh-refinement gpu-accelerated tsunami simulator for real-time operation. *Natural Hazards and Earth System Sciences* **18**(9), 2561–2602 (2018). <https://doi.org/10.5194/nhess-18-2561-2018>, <https://www.nat-hazards-earth-syst-sci.net/18/2561/2018/>
2. Cabinet Office: The investigative commission on massive earthquake models in nankai trough (2012), http://www.bousai.go.jp/jishin/nankai/model/data_teikyou.html, In Japanese
3. Christgau, S., Spazier, J., Schnor, B.: A performance and scalability analysis of the tsunami simulation easywave for different multi-core architectures and programming models. Tech. rep., Institute for Computer Science, University of Potsdam, (2015), <http://www.cs.uni-potsdam.de/bs/research/docs/techreports/2015/css15.pdf>
4. Imamura, F., Anawat, S.: Damage due to the 2011 Tohoku Earthquake Tsunami and its Lessons for Future Mitigation. In: *Proceedings of the International Symposium on Engineering Lessons Learned from the 2011 Great East Japan Earthquake*. Tokyo, Japan (March 2012)
5. Inoue, T., Abe, T., Koshimura, S., Musa, A., Murashima, Y., Kobayashi, H.: Development and validation of a tsunami numerical model with the polygonally nested grid system and its mpi-parallelization for real-time tsunami inundation forecast on a regional scale. *Journal of Disaster Research* **14**(3), 416–434 (2019). <https://doi.org/10.20965/jdr.2019.p0416>
6. IUGG/IOC Time Project: IUGG/IOC TIME PROJECT: Numerical method of Tsunami simulation with the leap-frog scheme. UNESCO (1997)

7. Kobayashi, H.: A Case Study of Urgent Computing on SX-ACE: Design and Development of a Real-Time Tsunami Inundation Analysis System for Disaster Prevention and Mitigation. In: Resch, M., et al. (eds.) *Sustained Simulation Performance 2016*. pp. 131–138. Springer-Verlag (2016)
8. Komatsu, K., Momose, S., Isobe, Y., Watanabe, O., Musa, A., Yokokawa, M., Aoyama, T., Sato, M., Kobayashi, H.: Performance Evaluation of a Vector Supercomputer SX-Aurora TSUBASA. In: *The International Conference on High Performance Computing, Networking, Storage and Analysis (SC18)*. pp. 54:1–54:12. IEEE Press, Dallas, USA (November 2018), <http://dl.acm.org/citation.cfm?id=3291656.3291728>
9. Koshimura, S.: Establishing the Advanced Disaster Reduction Management System by Fusion of Real-Time Disaster Simulation and Big Data Assimilation. *Journal of Disaster Research* **11**(2), 164–174 (2016)
10. Koshimura, S., Oie, T., Yanagisawa, H., Imamura, F.: Developing fragility functions for tsunami damage estimation using numerical model and post-tsunami data from Banda Aceh, Indonesia. *Coastal Engineering Journal, JSCE* **51**(3), 243–273 (2009)
11. Koshiyama, K.: Characteristics of Emergency Responcy at the Great East Japan Earthquake. In: *5th International Disaster and Risk Conference Davos 2014* [poster]. Davos, Switzzland (August 2014), https://idrc.info/fileadmin/user_upload/idrc/documents/IDRC14.PosterCollection.pdf
12. Momose, S.: SX-ACE, Brand-New Vector Supercomputer for Higher Sustained Performance I. In: Resch, M., et al. (eds.) *Sustained Simulation Performance 2014*. pp. 57–67. Springer-Verlag (2014)
13. Mori, N., Takahashi, T., Yanagisawa, H.: Survey of 2011 Tohoku earthquake tsunami inundation. *Geophysical Research Letters* **38** L00G14 (2016). <https://doi.org/10.1029/2011GL049210>
14. Musa, A., Abe, T., Inoue, T., Hokari, H., Murashima, Y., Kido, Y., Date, S., Shimojo, S., Koshimura, S., Kobayashi, H.: A real-time tsunami inundation forecast system using vector supercomputer sx-ace. *Journal of Disaster Research* **13**(2), 234–244 (2018). <https://doi.org/10.20965/jdr.2018.p0234>
15. Musa, A., Watanabe, O., Matsuoka, H., Hokari, H., Inoue, T., Murashima, Y., Ohta, Y., Hino, R., Koshimura, S., Kobayashi, H.: Real-time tsunami inundation forecast system for tsunami disaster prevention and mitigation. *The Journal of Supercomputing* **74**(7), 3093–3113 (Jul 2018). <https://doi.org/10.1007/s11227-018-2363-0>, <https://doi.org/10.1007/s11227-018-2363-0>
16. Nagasu, K., Sane, K., Kono, F., Nakasato, N.: FPGA-based tsunami simulation: Performance comparison with GPUs, and roofline model for scalability analysis. *Journal of Parallel and Distributed Computing* **106**, 153–169 (August 2017), <http://dx.doi.org/10.1016/j.jpdc.2016.12.015>
17. NEC Corporation: *NEC Fortran 2003 Programmer’s Guide* (2014)
18. Okada, N., Tao, Y., Yoshio, K., Peijun, S., Hirokazu, T.: The 2011 eastern japan great earthquake disaster: Overview and comments. *International Journal of Disaster Risk Science* **2**(1), 34–42 (Mar 2011). <https://doi.org/10.1007/s13753-011-0004-9>, <https://doi.org/10.1007/s13753-011-0004-9>
19. Okada, Y.: Internal Deformation due to Shear and Tensile Faults in a Half-Space. *Bulletin of the Seismological Society of America* **82**(2), 1018–1040 (1992)
20. Yamada, Y., Momose, S.: Vector Engine Processor of NEC’s Brand-New supercomputer SX-Aurora TSUBASA. In: *International symposium on High Performance Chips (Hot Chips)*. Cupertino, USA (August 2018)