

A Real-Time Tsunami Inundation Forecast System for Tsunami Disaster Prevention and Mitigation

Akihiro Musa^{a),b)}, Hiroshi Matsuoka^{a),b)}, Osamu Watanabe^{a),b)}, Yoichi Murashima^{c),d)}, Shunichi Koshimura^{c)}, Ryota Hino^{e)}, Yusaku Ohta^{e)}, and Hiroaki Kobayashi^{a)}

^{a)}Cyberscience Center, Tohoku University, ^{b)}NEC Corporation, ^{c)}IRIDeS Tohoku University, ^{d)}Kokusai Kogyo Co. ^{e)}AOB Tohoku University

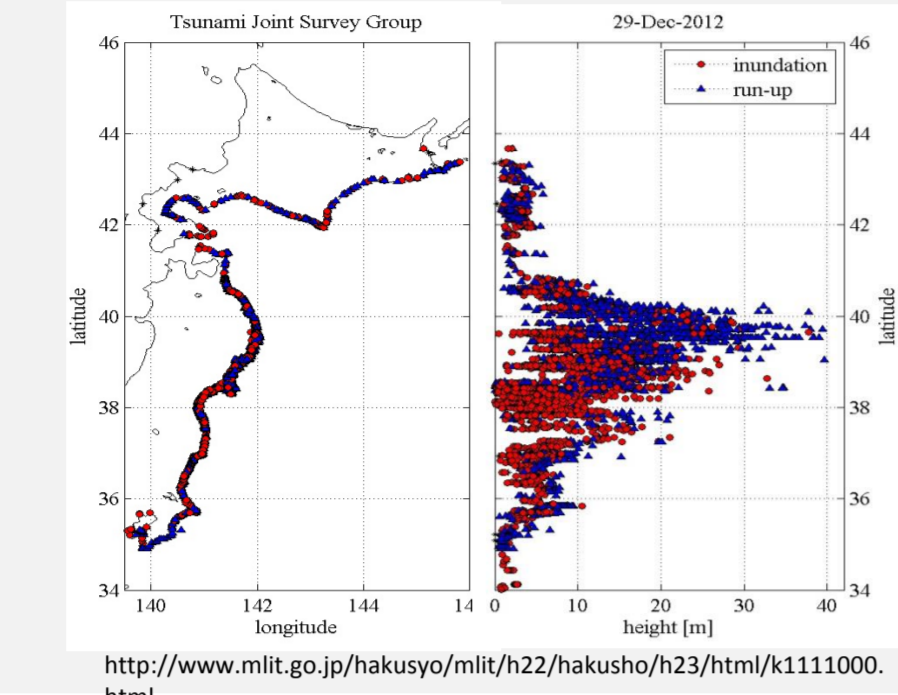
Background and Motivation

□ 2011 Great East Japan Earthquake of magnitude 9.0

□ A large tsunami caused by the earthquake struck Japanese coasts.

- Tsunami wave height: over 10m
- Highest run-up: 40m
- Inundation of 561 Km²
- 18,549 fatalities
- 120,000 buildings destroyed

Tsunami inundation and run-up height



The Great East Japan Earthquake



□ The most important lessons learned from the Great East Japan Earthquake.

- Require quick understanding of disaster situation for mitigation

□ What necessary therefore ??

- **Real-time tsunami analysis using HPC**

Tsunami Simulation Diagram

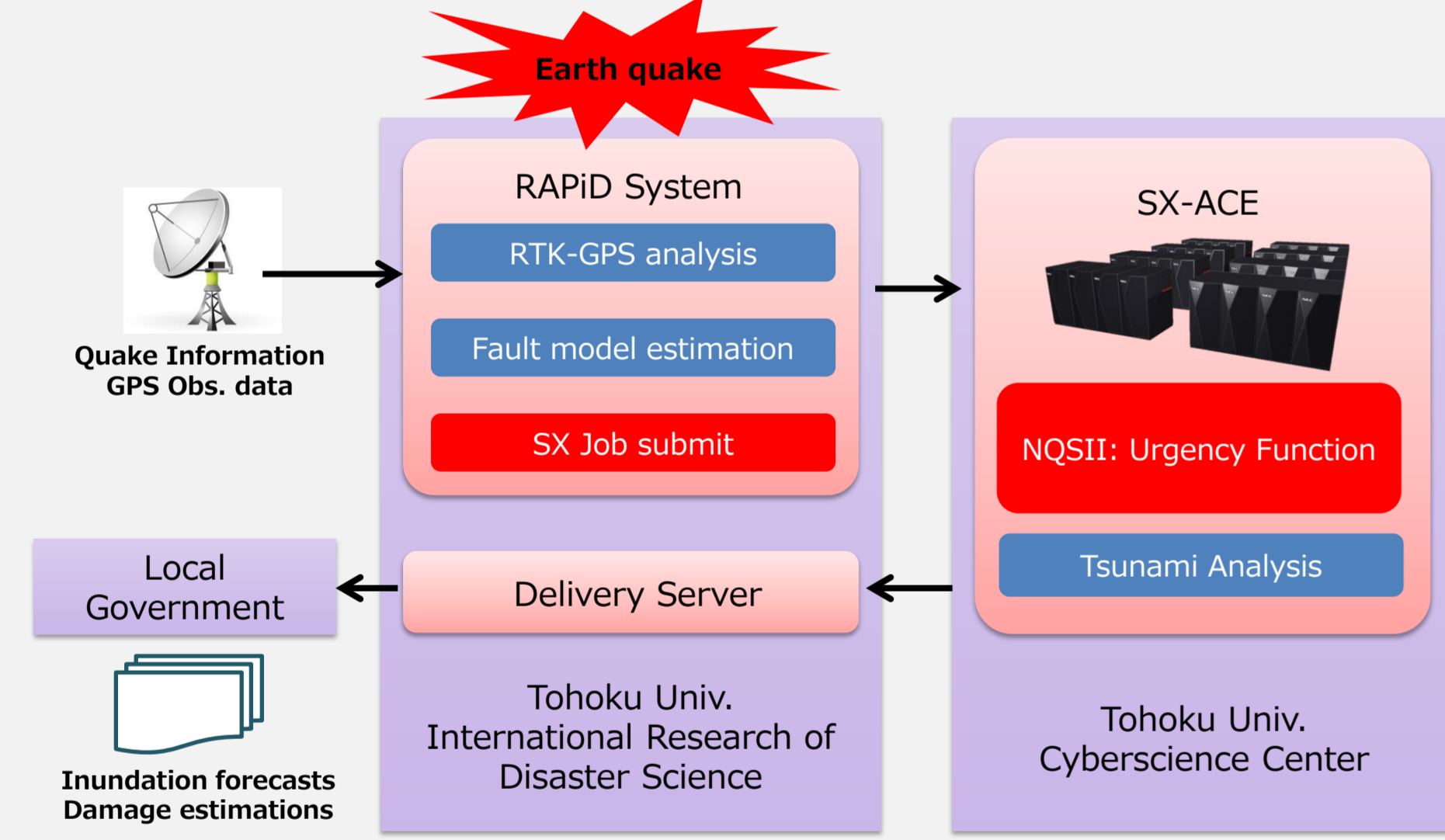
□ We have developed a real-time tsunami inundation forecast system that consists of

- Tsunami source modeling system (RAPID system),
- Simulation system (SX-ACE), and
- Delivery/mapping server.

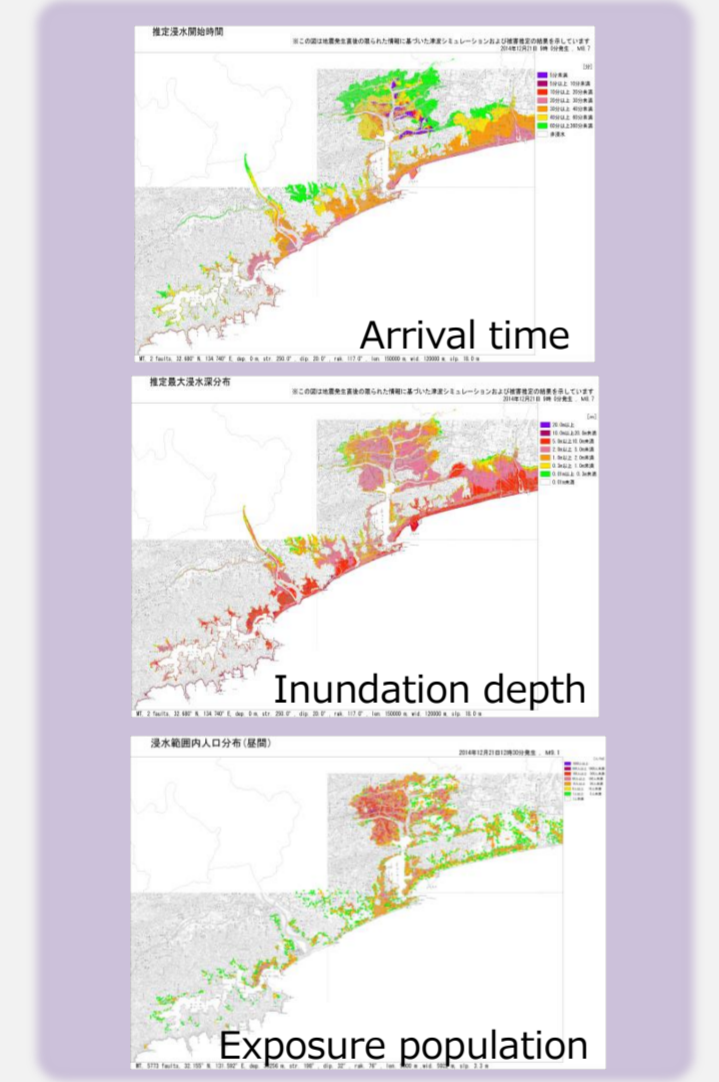
□ The simulation flow

1. Obtaining the GPS observation data with earthquake information
2. Estimating the coseismic fault by the RAPID system
3. Executing the tsunami inundation simulation on SX-ACE
4. Visualizing and providing the simulation results by the delivery/mapping server

Simulation Flow Diagram



Delivered Information



An Urgent Job Management Function of SX-ACE

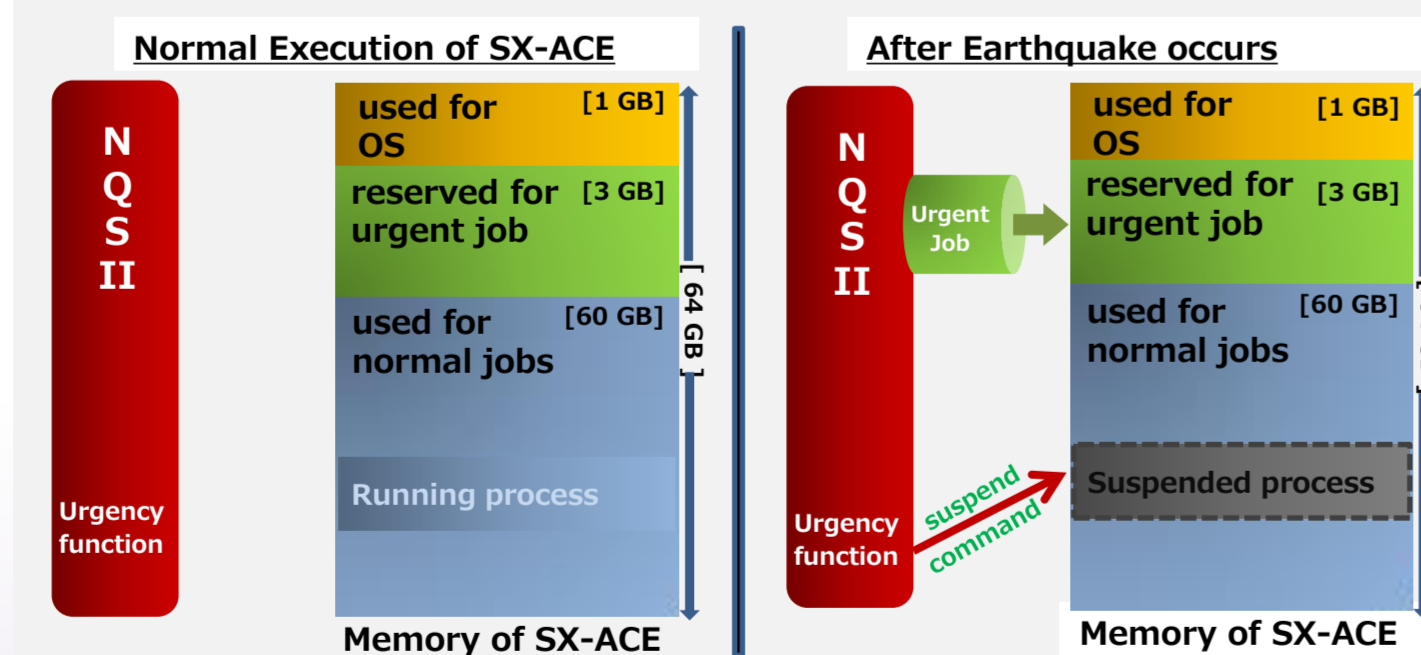
□ Developed based on SX-ACE job management system "NQS II

□ Running an urgent job on SX-ACE only when a tsunamigenic earthquake occurs in the normal operation

- Usually running academic and industry usage jobs on SX-ACE

In the emergency operation

- Switching from normal jobs to an urgent job immediately (within 30 sec) by the 'suspend' function
- Running the urgent job at the highest priority
- Resuming suspended normal jobs when the urgent job has finished



SX-ACE Architecture

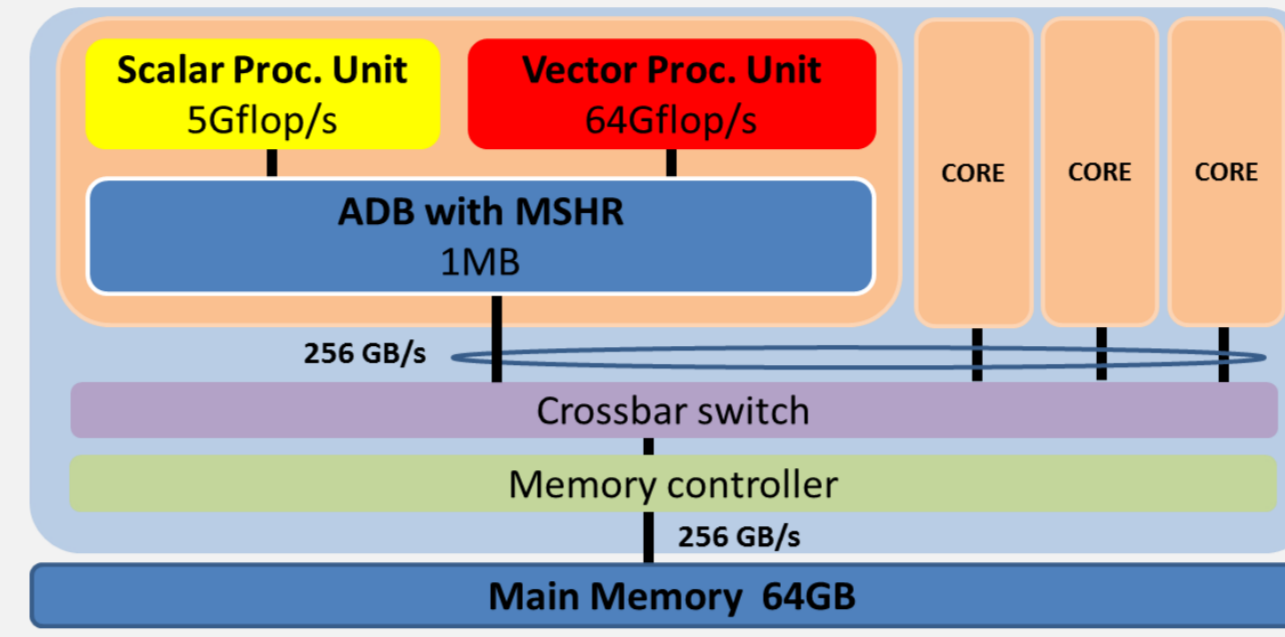
□ SX-ACE major architectural features for a high memory bandwidth with a high computational performance

- A multi vector-cores architecture
- An ADB with MSHR mechanisms to keep a high sustained memory bandwidth
- An Out-of-order vector memory access mechanism to improve the performance of indirect/stride memory accesses
- A shorter memory access latency and a shorter vector pipe latency in chaining to realize efficient short-vector processing

□ ADB & MSHR

- ADB (Assignable Data Buffer) Specification
 - 1MB private on-chip cache memory per core (4MB per socket)
 - 4-Way set-associative
 - 256GB/s to/from Vec. Reg.
 - Software controllable
 - ✓ Compiler/User control
 - ✓ A bypass flag each instruction
 - ✓ Avoiding cache pollution
 - Customized for fast random access
- MSHR (Miss Status Handling Registers) reduces redundant memory access by holding the information of in-flight memory accesses.

SX-ACE Processor Diagram



Core Specification	
Peak Performance	69.0 Gflop/s
Vector Performance	64.0 Gflop/s
ADB Capacity	1.0 MB
Memory Bandwidth	64 GB/s (4 cores) - 256 GB/s (1 core)
B/F	1.0 (4 cores) - 4.0 (1 core)

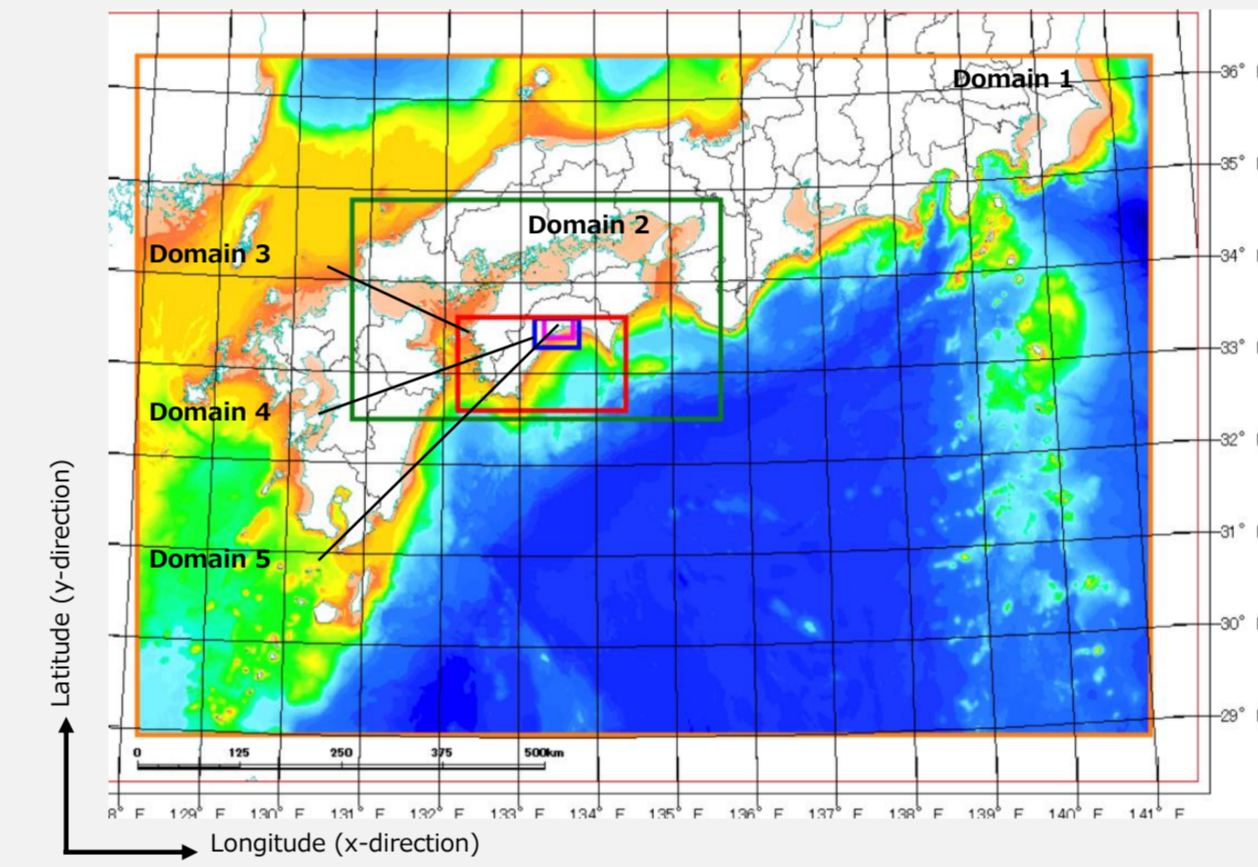
CPU Specification	
Number of Cores	4
Peak Performance	276.0 Gflop/s
Vector Performance	256.0 Gflop/s
Memory Capacity	64 GB
Memory Bandwidth	256 GB/s
B/F	1.0

Computational Domain (Kochi City and Adjacent Sea Area)

□ Kochi area as a simulation example

- A case of Nankai trough earthquake
- 1244 km x 826 km (size of Domain 1)
- 6 hours simulation
- Delta t = 0.1 sec
- 5 nested grids

Computational Domains of Kochi



Domain	Grid Size (m)	Number of Grid Points	
		x-direction	y-direction
1	810	1536	1020
2	270	1680	990
3	90	2292	1260
4	30	1782	1188
5	10	3504	2364

TUNAMI-Code – Tsunami Simulation Model

□ TUNAMI: Tohoku University's Numerical Analysis Model for Investigating Tsunami

- Governing equations: Non-linear shallow water equations
- Numerical scheme: Staggered leap-frog difference method

Governing Equations

$$\frac{\partial \eta}{\partial t} + \frac{\partial M}{\partial x} + \frac{\partial N}{\partial y} = 0$$

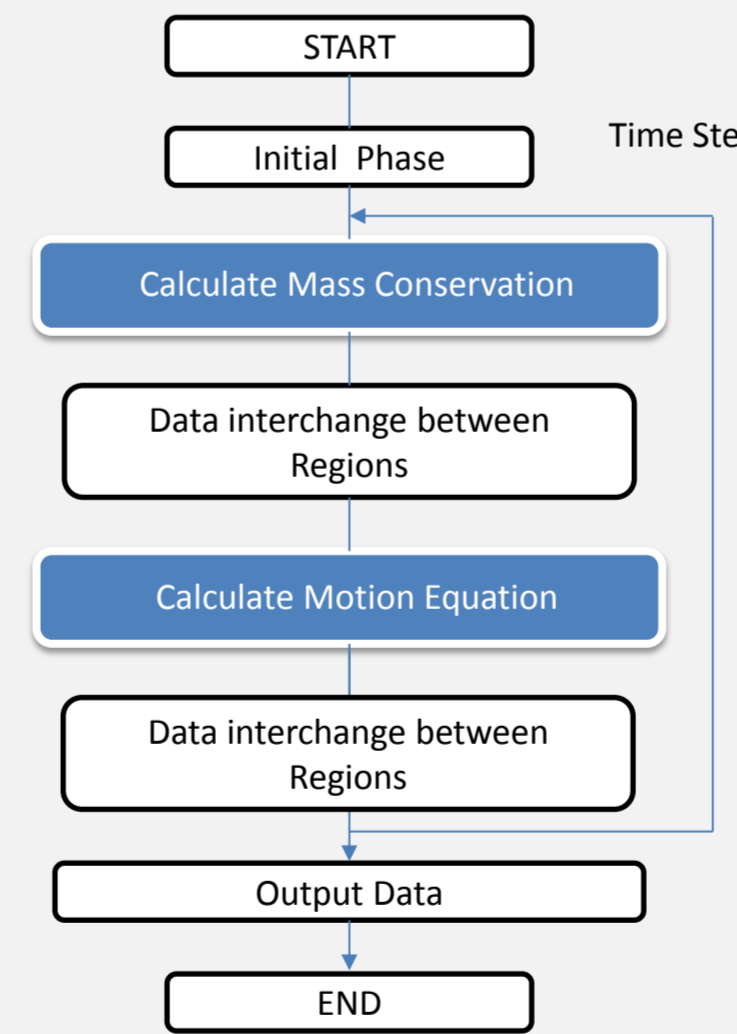
$$M = \int_{-h}^{\eta} u dz = \bar{u}(\eta + h)$$

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

$$\frac{\partial M}{\partial t} + \frac{\partial}{\partial x} \left(\frac{M^2}{D} \right) + \frac{\partial}{\partial y} \left(\frac{MN}{D} \right) = -gD \frac{\partial \eta}{\partial x} - \frac{\tau_{xx}}{\rho} - \frac{\tau_{xy}}{\rho} = \frac{gn^2}{D^{7/3}} M \sqrt{M^2 + N^2}$$

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial x} \left(\frac{MN}{D} \right) + \frac{\partial}{\partial y} \left(\frac{N^2}{D} \right) = -gD \frac{\partial \eta}{\partial y} - \frac{\tau_{xy}}{\rho} - \frac{\tau_{yy}}{\rho} = \frac{gn^2}{D^{7/3}} N \sqrt{M^2 + N^2}$$

Program Structure



□ High cost routines have doubly nested loops

- Calculations of mass conservation and motion equation are dominant

□ High vectorized code

- **Vector Operation Ratio: 99.6 %**
- **Average Vector Length: 235**

□ High memory-intensive code

- **Program B/F: 1.85**

□ Main computations of TUNAMI-Code

Mass Conservation Calculation

```
DO J=2, JF ... Longitude direction loop <- Parallelized
DO I=2, IF ... Latitude direction loop <- Vectorized
...
ZZ=Z(I,J,1)-RX*(M(I,J,1)-M(I-1,J,1))
1 -RY*(M(I,J,1)-M(I,J-1,1))
...
END DO
END DO
```

Motion Equation Calculation

```
DO J=1, JF ... Longitude direction loop <- Parallelized
DO I=1, IF ... Latitude direction loop <- Vectorized
...
XNN=0.25*(N(I,J,1)+N(I+1,J,1)+N(I,J-1,1)+N(I+1,J-1,1))
FF=FN*SQRT(M(I,J,1)**2+XNN**2)/DD**(7.0/3.0)
XM=(1.0-FF)*M(I,J,1)-GG*RX*DD*(Z(I+1,J,2)-Z(I,J,2))
...
END DO
END DO
```

□ ADB realizes a reduction in memory access effectively.

- Arrays **M**, **N**, and **Z** are cached on ADB
- Maximum size of 1st x 2nd dimension of these arrays is 244KB < ADB Capacity (1MB)
- (*) 244KB = 3504 * (2364/133) * 4B, 133 is number of mpi processes assigned domain5 on 256 mpi execution.
- **ADB Hit Ratio: 67 %**

□ MSHR contributes to reduction in redundant memory request efficiently.

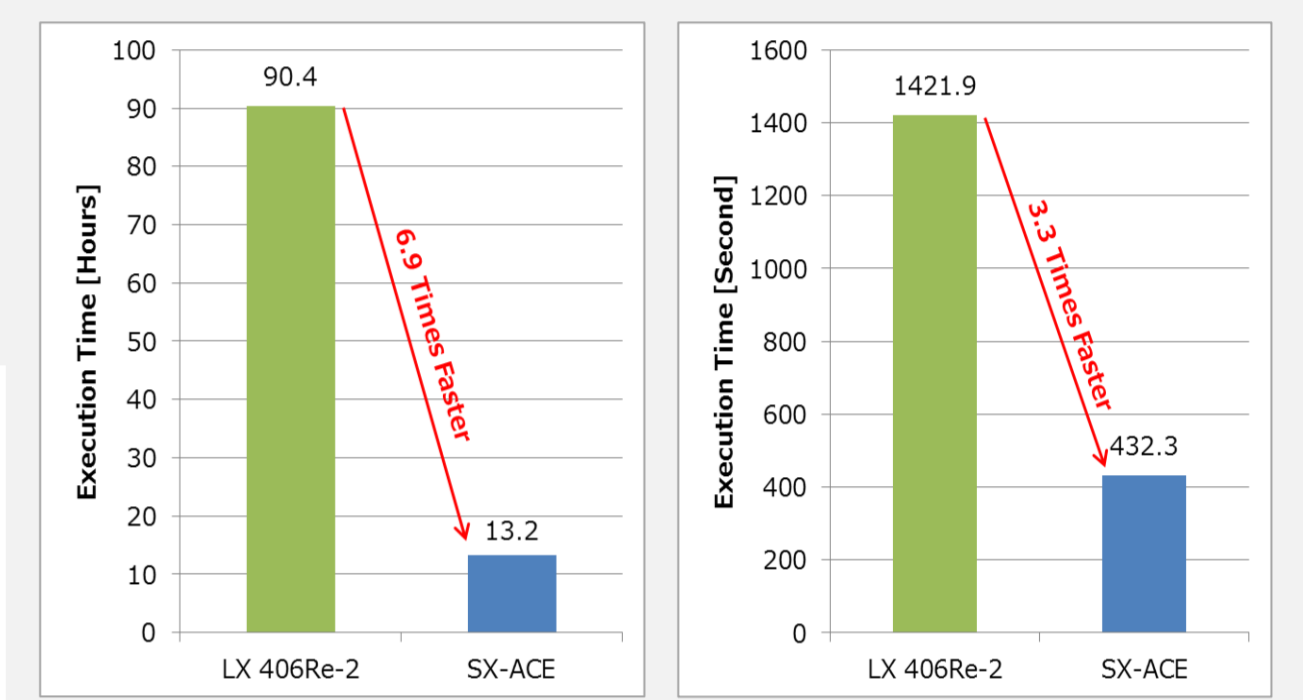
- Memory access of array **M**, **N**, and **Z** is adjacent along I-direction.
- MSHR holds information on in-flight load data (Instruction address and memory address)
- When a subsequent instruction loads the in-flight load data, the subsequent instruction loads the data from ADB instead of the memory.

□ Actual B/F ⁽¹⁾ is reduced from program B/F ⁽¹⁾ dramatically by ADB with MSHR.

- **Program B/F: 1.85 -> Actual B/F: 0.48**

Performance Evaluation

	SX-ACE	LX 406 Re-2	K computer
Core Perf.	64.0 Gflop/s	19.2 Gflop/s	16.0 Gflop/s
CPU Perf.	256.0 Gflop/s	230.4 Gflop/s	128.0 Gflop/s
Mem-BW	256.0 GB/s	59.7 GB/s	64.0 GB/s
B/F	1.00	0.26	0.50



□ Single core performance

- Peak core performance of SX-ACE is 3.3 times higher than one of LX 406 Re-2.
- SX-ACE is 6.9 times faster than LX 406 Re-2.
- **Effective performance ratio is 2 times better than peak performance ratio.**

□ Single socket performance

- Peak socket performance of SX-ACE is almost the same as one of LX 406 Re-2.
- SX-ACE is 3.3 times faster than LX 406 Re-2.
- **Effective performance ratio is 3 times better than peak performance ratio.**
- ADB with MASH effectively reduces the actual B/F by storing reusable data in ADB and reducing redundant memory accesses.

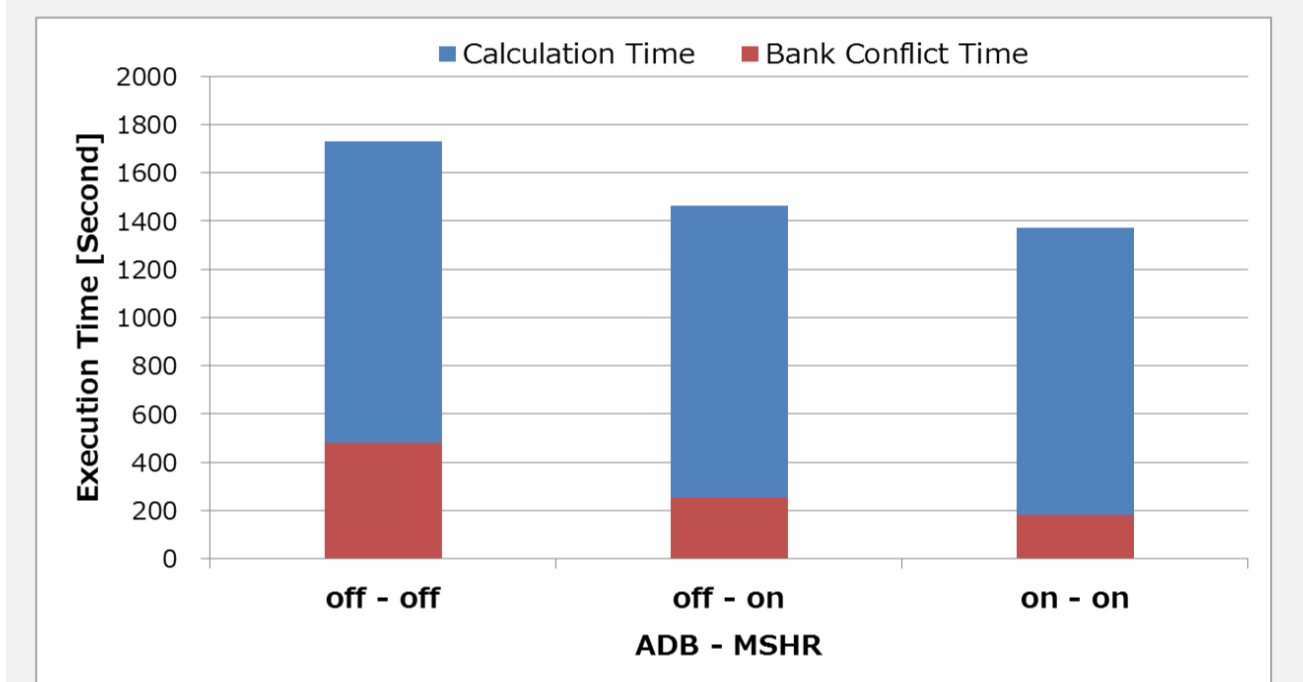
$$\text{Actual B/F: } 0.48 < \text{SX-ACE B/F: } 1.00 < \text{Program B/F: } 1.85$$

□ Multi node performance

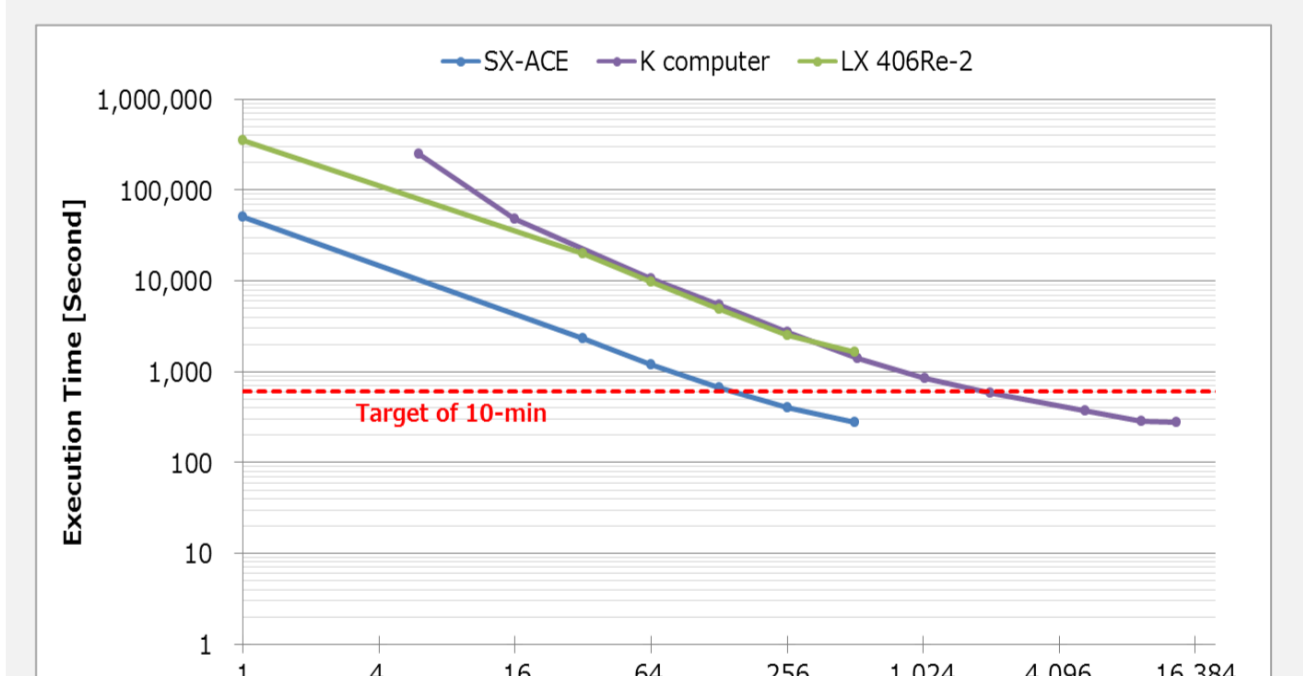
- SX-ACE with 256 cores achieves the time target of less than 10 minutes.
- LX 406 Re-2 and K computer ⁽¹⁾ need over 8000 cores to reach the same target ⁽²⁾.

□ SX-ACE's high memory bandwidth brings out a good performance of TUNAMI-Code.

- Key factor1: B/F: **TUNAMI-Code: 1.85 > SX-ACE: 1.00 > K computer: 0.50 > LX406Re-2: 0.26**
- Key factor2: **ADB with MSHR makes full use of the memory bandwidth further.** These mechanisms enhance superior advantages of memory performance over scalar architectures.



Effect of ADB and MSHR on Socket of SX-ACE



Performance on Multiple Cores (6hrs Simulation)

Conclusions and Future Work

□ HPC plays a key role in quick understanding of accurate tsunami disaster situations to enables prompt responses to the disaster.

- The tsunami forecasting systems for near-field tsunamis require increasing the accuracy and reducing the processing time.
- The real-time tsunami simulation system is developed using the vector supercomputer **NEC SX-ACE** and **achieves the high accuracy tsunami inundation forecasting with 10-meter high resolution grids in 10 minutes.**

□ The following future work is addressed:

- Parallel performance enhancement of the tsunami inundation simulation for cutting the execution time in half
- Building a regionally-distributed multiplexing system for redundant and spreading the target areas all over Japan
- Cooperation of our tsunami inundation forecast system with the Japanese disaster information sharing system for sharing the results with the public

Reference

[1] Yusuke Oishi, Fumihiko Imamura, and Daisuke Sugiyawa, Near-field tsunami inundation forecast using the parallel TUNAMI-N2 model: Application to the 2011 Tohoku-Oki earthquake combined with source inversions, Geophysical Research Letters, 42, doi:10.1002/2014GL062577, 2015

Note

(1) Program B/F is the ratio of the total amount of data (bytes) to the number of floating point operations when executing a program, and Actual B/F is the ratio of the actual data movement (bytes) between the processor and the memory of floating point operations when executing a program. Actual B/F is usually smaller than Program B/F because some of memory data can be provided by the on-chip cache without memory accesses.

(2) The data of K computer is obtained from [1]. We think the comparison is reasonable because these systems use the same TUNAMI code and the total number of grid points and the time step are almost the same in these systems.