Importance of Selecting Data Layouts in the Tsunami Simulation Code

TAKUMI KISHITANI[1], KAZUHIKO KOMATSU[1], MASAYUKI SATO[1],
AKIHIRO MUSA[1,2] ,HIROAKI KOBAYASHI[1]

[1]TOHOKU UNIVERSITY, [2]NEC CORPORATION

Contents

TOHOKU UNIVERSITY

- Background
  - The real-time tsunami inundation forecast system
  - The tsunami simulation
- Objective & Approach
- Data layout in the tsunami simulation
  - The characteristics of DA, AoS, SoA data layouts
  - Implementation of the tsunami simulation
- Evaluation
  - Experimental environments
  - Evaluation of execution time
  - Analysis of results
- Conclusion

Backgrounds

TOHOKU UNIVERSITY

HPC applications

- Playing important roles in various fields
- Tending to require high data transfer capability
  - E.g. Simulation applications
  - Demands for dealing with much data for calculations

Memory-intensive applications

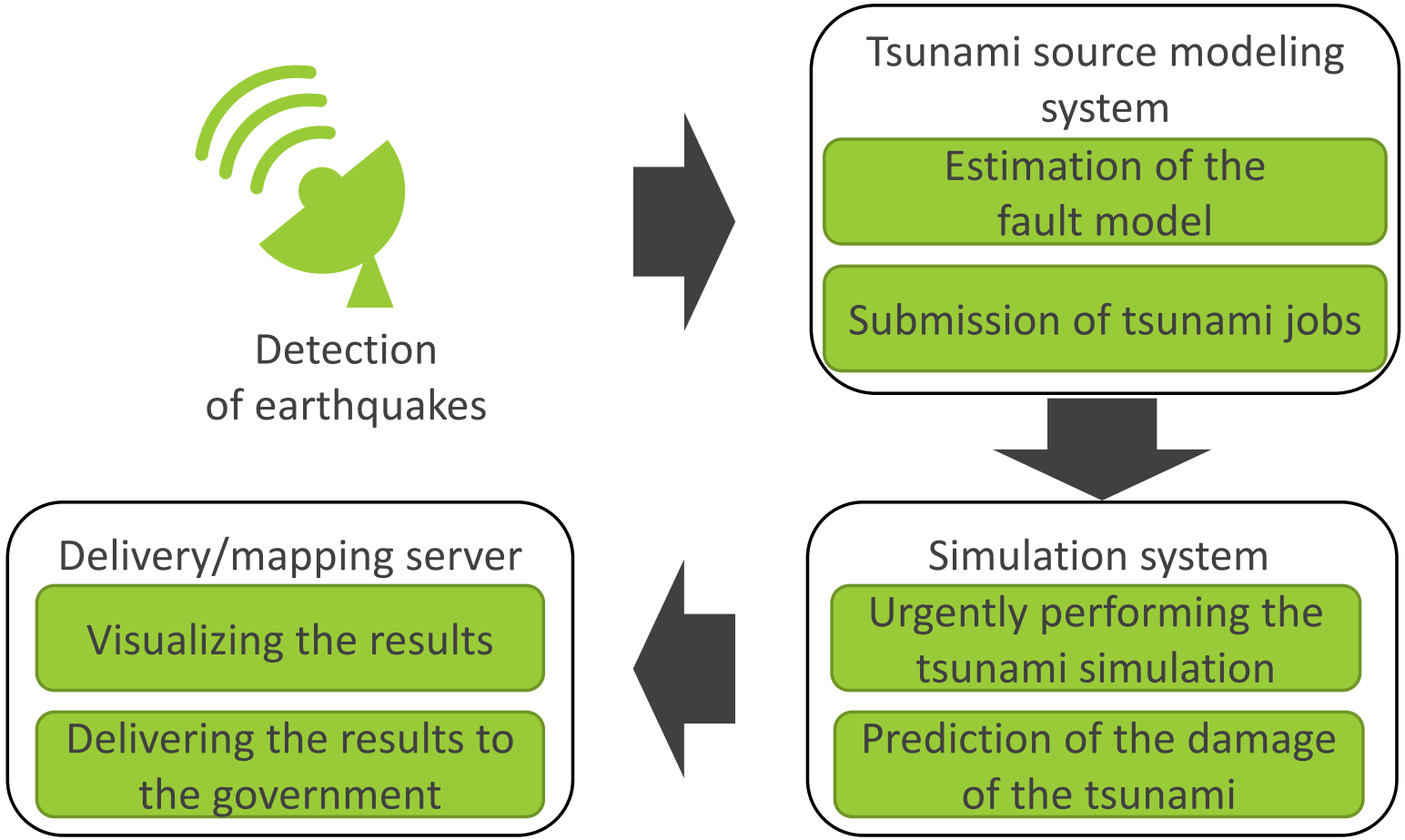
- A sustained performance is limited by the memory performance of computing systems
  - The wide performance gap between processor and memory
- One of the challenge to accelerate memory-intensive applications is full exploitation of the memory performance
- A tsunami simulation as the target application in this research
  - A part of the real-time tsunami inundation forecast system

The real-time tsunami inundation forecast system

TOHOKU UNIVERSITY

Overviews

- The system that helps to decide the appropriate actions for tsunamis' damage



The tsunami simulation

TOHOKU UNIVERSITY

Overviews

- Computational model :TUNAMI Model
  - Governing equations :Nonlinear shallow water equations
  - Numerical scheme :The staggered leap-flog finite difference method
- Outputs
  - The maximum inundation depth, the tsunami arrival time, the water level changes over time, and so on

Challenges

- Acceleration
  - Simulation of the damage in real time
- Achievement of high performance independent of target areas
  - The execution time varies depending on input data
  - It is impossible to predict the target areas where tsunami reaches

Objective & Approach

TOHOKU UNIVERSITY

Objective

- Acceleration of the tsunami simulation with various input data and computing systems
- Clarification of the necessity of the tuning strategy for selecting data structure depending on input data and computing systems

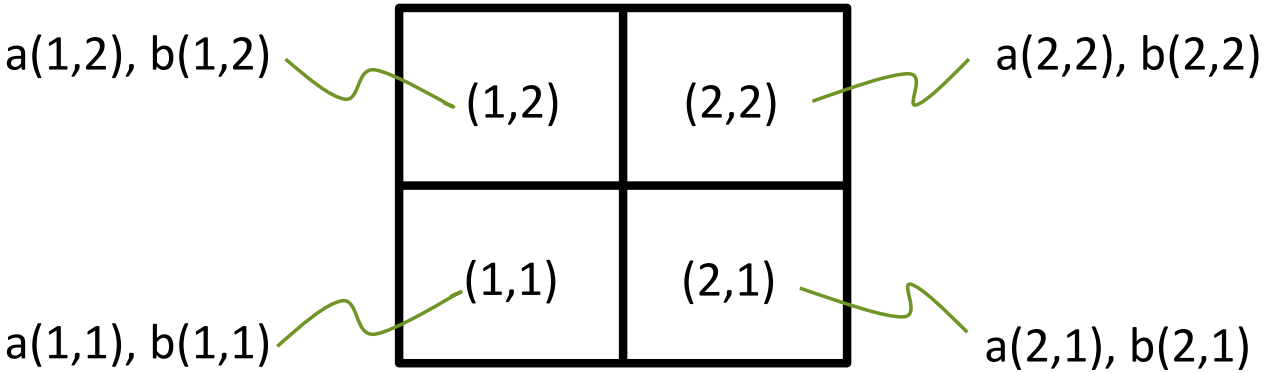
Approach

- Adapting typical types of memory-access-efficient data structures to the tsunami simulation
  - Discrete array
  - Array of structures
  - Structure of arrays
- Analyzing the performance variations

Data structures in simulation applications

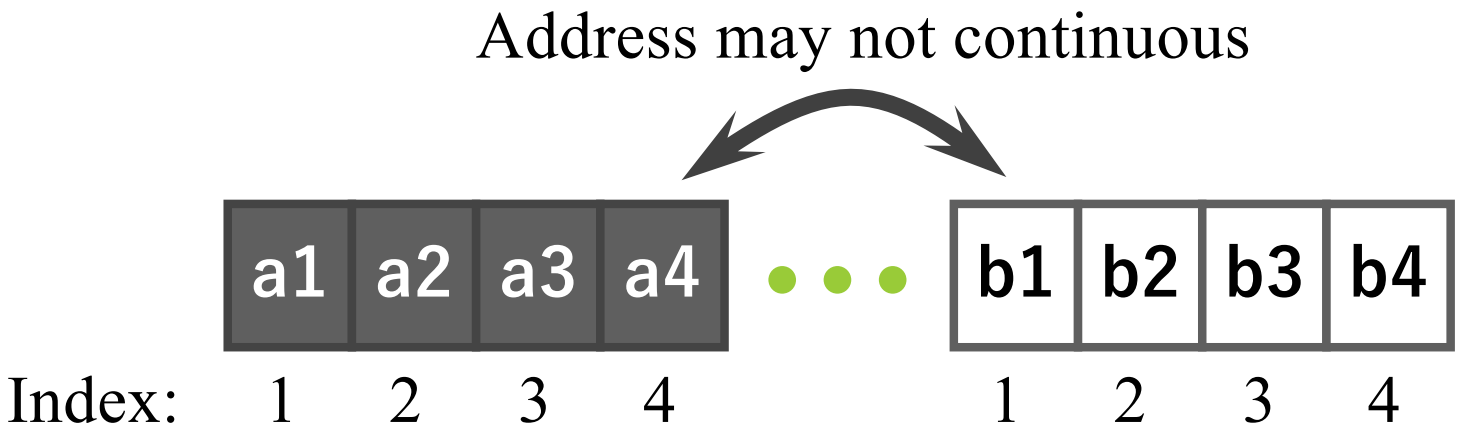
TOHOKU UNIVERSITY

- Data definition in the numerical study of dynamics
  - Multiple physical values are defined for each point of a computation grid



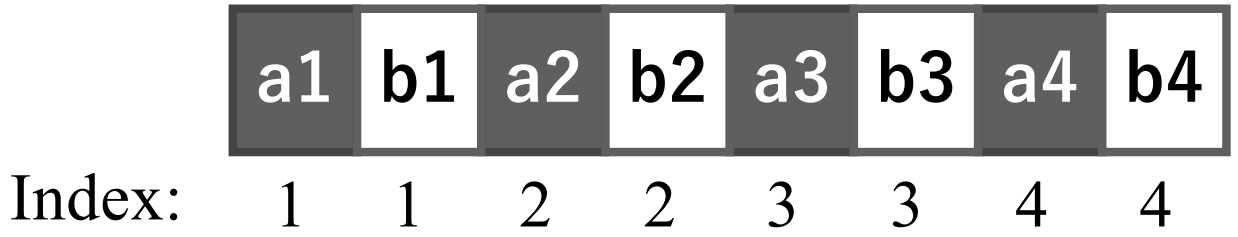
- Typical data structures where physical values are defined
  - Three types of typical data structures
    - Discrete arrays
    - Array of structures
    - Structure of arrays

Discrete arrays (**DA**)



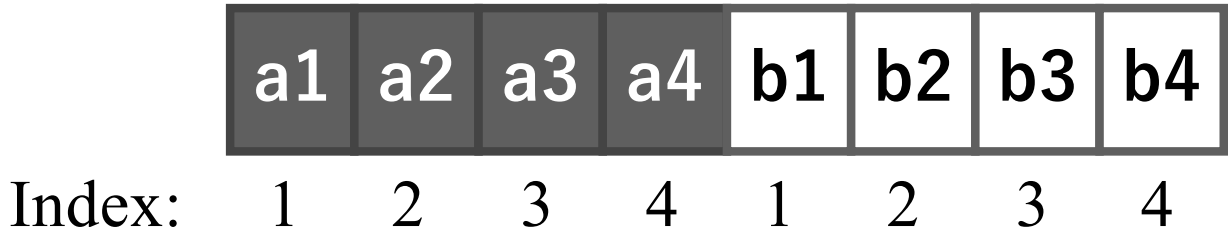
- Data layout on a memory
  - Physical values are stored in each array
  - Elements of physical values are stored in the order of index on a grid
    - ✓ Efficient memory accesses
  - Head addresses of arrays may be separated each other
    - × Inefficient memory accesses when accessing some arrays at once

Array of structures (**AoS**)



- Data layout on a memory
  - Data sets compose an array
  - Physical values of each grid are continuously aligned
    - ✓ Convenient to deal with physical values data of individual grid points
  - An element of a physical value is distant from another element
    - × Memory accesses may become inefficient

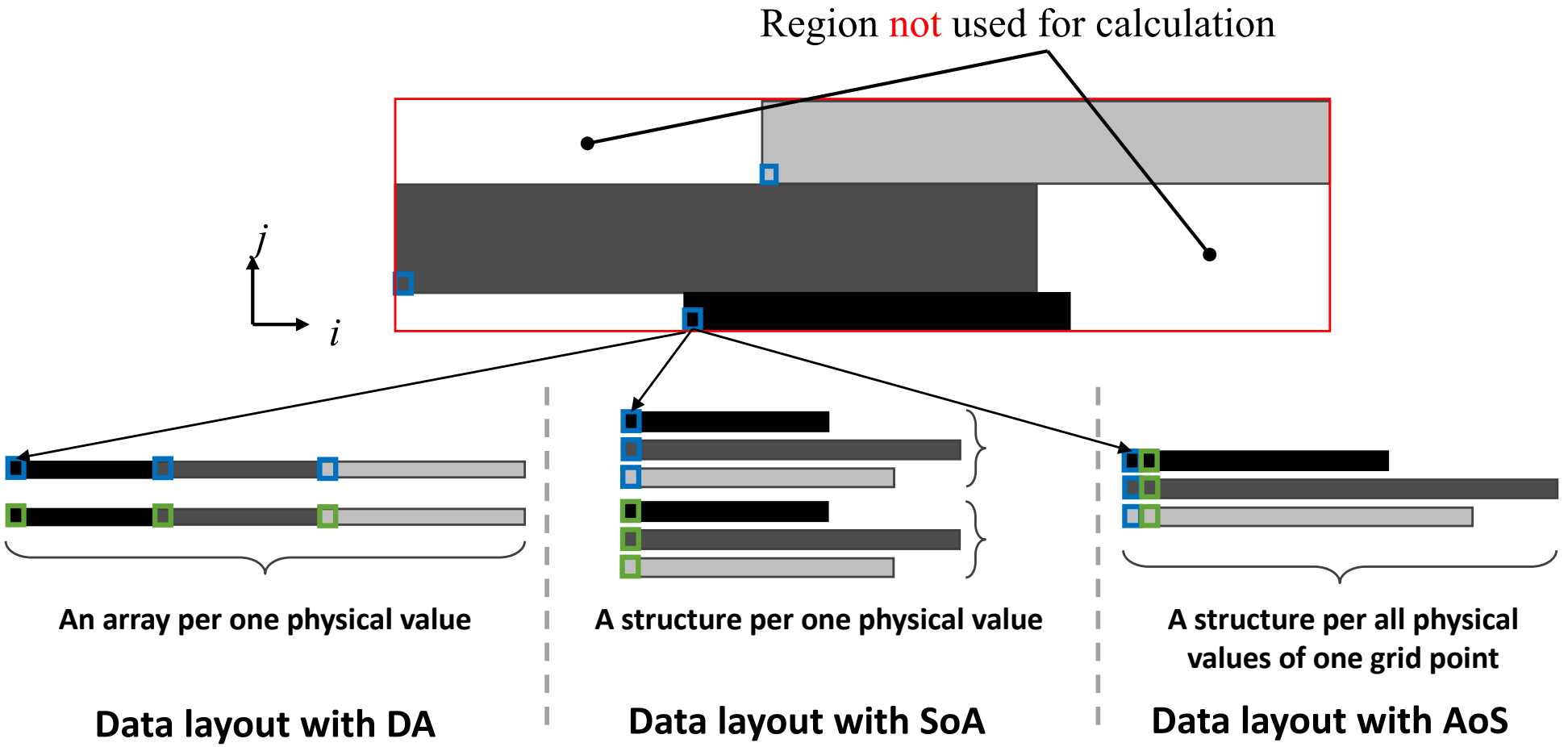
Structure of Arrays (**SoA**)



- Data layout on a memory
  - One data structure contains several arrays holding physical values
    - One array is provided for each physical value (same as the **DA** data layout)
  - Elements of the same physical values are placed closer on a memory
    - ✓ Efficient memory accesses
  - The arrays are also continuously aligned on a memory
    - ✓ Memory access may become more efficient than the **DA** data layout

Implementation of the tsunami simulation

- Concepts of three data layouts in the tsunami simulation



Implementation of the tsunami simulation

TOHOKU UNIVERSITY

Example codes

Code 4. The implementation of the *DA* data layout

```
 1:SUBROUTINE NLMNT2_DA
 2:···
 3:REAL,DIMENSION(BDLNG(N),2) :: ZL,ML,NL
 4:···
 5:DO K=1,NUMBLOCK(N)
 6:!$OMP DO PRIVATE(L,J)
 7:  DO J=1,JLNG(K,N)
 8:!$OMP SIMD
 9:    DO L=LJSTA(J,K,N),LJEND(J,K,N)
10:      ···
11:      XNNL=0.25*(NL(L,1)+NL(L+1,1)+NL(L-BIL,1)&
12:         +NL(L+1-BIL,1))
13:      FFXL=FN*SQRT(ML(L,1)*ML(L,1)+XNNL*XNNL)&
14:         /DDXL**(7.0/3.0)
15:      XML =(1.0-FFXL)*ML(L,1)&
16:         -GG*RX*DDXL*(ZL(L+1,2)-ZL(L,2))
17:      ···
18:    END DO
19:  END DO
20:END DO
21:···
22:END SUBROUTINE NLMNT2_DA
```

Code 5. The implementation of the *SoA* data layout

```
 1:SUBROUTINE NLMNT2_SOA
 2:···
 3:TYPE ZLST
 4:  REAL,DIMENSION(:) :: ZLS
 5:END TYPE
 6:TYPE MLST
 7:  REAL,DIMENSION(:) :: MLS
 8:END TYPE
 9:TYPE NLST
10:  REAL,DIMENSION(:) :: NLS
11:END TYPE
12:TYPE ALLD
13:  TYPE(ZLST),DIMENSION(:,:) :: ZL
14:  TYPE(MLST),DIMENSION(:,:) :: ML
15:  TYPE(NLST),DIMENSION(:,:) :: NL
16:END TYPE
17:TYPE(ALLD) :: ALLV
18:···
19:DO K=1,NUMBLOCKS
20:!$OMP DO PRIVATE(L,J)
21:  DO J=1,JLNG(K)
22:!$OMP SIMD
23:    DO L=LJSTA(J,K),LJEND(J,K)
24:      ···
25:      XNNL=0.25*(ALLV%NL(K,1)%NLS(L) &
26:         +ALLV%NL(K,1)%NLS(L+1) &
27:         +ALLV%NL(K,1)%NLS(L-BIL) &
28:         +ALLV%NL(K,1)%NLS(L+1-BIL))
29:      FFXL=FN*SQRT(ALLV%ML(K,1)%MLS(L) &
30:         *ALLV%ML(K,1)%MLS(L) &
31:         +XNNL*XNNL)/DDXL**(7.0/3.0)
32:      XML =(1.0-FFXL)*ALLV%ML(K,1)%MLS(L) &
33:         -GG*RX*DDXL*(ALLV%ZL(K,1)%ZL(L+1) &
34:         -ALLV%ZL(K,2)%ZL(L))
35:      ···
36:    END DO
37:  END DO
38:END DO
39:···
40:END SUBROUTINE NLMNT2_SOA
```

Code 6. The implementation of the *AoS* data layout

```
 1:SUBROUTINE NLMNT2_AOS
 2:···
 3:TYPE ELEMENTS
 4:  REAL,DIMENSION(:) :: ZL
 5:  REAL,DIMENSION(:) :: ML
 6:  REAL,DIMENSION(:) :: NL
 7:END TYPE
 8:TYPE BLOCKS
 9:  TYPE(ELEMENTS),DIMENSION(:) :: EMLS
10:END TYPE
11:TYPE(BLOCKS),DIMENSION(NUMBLOCKS) :: BLKS
12:···
13:DO K=1,NUMBLOCKS
14:!$OMP DO PRIVATE(L,J)
15:  DO J=1,JLNG(K)
16:!$OMP SIMD
17:    DO L=LJSTA(J,K),LJEND(J,K)
18:      ···
19:      XNNL=0.25*(BLKS(K)%ELMS(L)%NL(1) &
20:         +BLKS(K)%ELMS(L+1)%NL(1) &
21:         +BLKS(K)%ELMS(L-BIL)%NL(1) &
22:         +BLKS(K)%ELMS(L+1-BIL)%NL(1))
23:      FFXL=FN*SQRT(BLKS(K)%ELM(L)S%ML(1) &
24:         *BLKS(K)%ELMS(L)%ML(1) &
25:         +XNNL*XNNL)/DDXL**(7.0/3.0)
26:      XML =(1.0-FFXL)*BLKS(K)%ELMS(L)%ML(1) &
27:         -GG*RX*DDXL*(BLKS(K)%ELMS(L+1)%ZL(2) &
28:         -BLKS(K)%ELMS(L)%ZL(2))
29:      ···
30:    END DO
31:  END DO
32:END DO
33:···
34:END SUBROUTINE NLMNT2_AOS
```

Experimental environments

TOHOKU UNIVERSITY

Computing systems

Processor	Intel Xeon Gold 6126	Vector Engine Type 10B
Performance	1996.8 Gflop/s (SP)	4.30 Tflop/s (SP)
Memory subsystems	DDR4-2666	HBM2
Memory bandwidth	128 GB/s	1.22 TB/s
Memory capacity	192 GB	48 GB

Software environments

Compiler	Intel Compiler 19.0.5.281	NEC Compiler for VE 2.5.1
Options	-O3 -mcmodel=large -xCORE-AVX512 -qopt-zmm-usage=high -qopemmp	-O3 -fopenmp

Experimental environments

TOHOKU UNIVERSITY

Kernel code

- Extracted high-cost subroutine

Input data

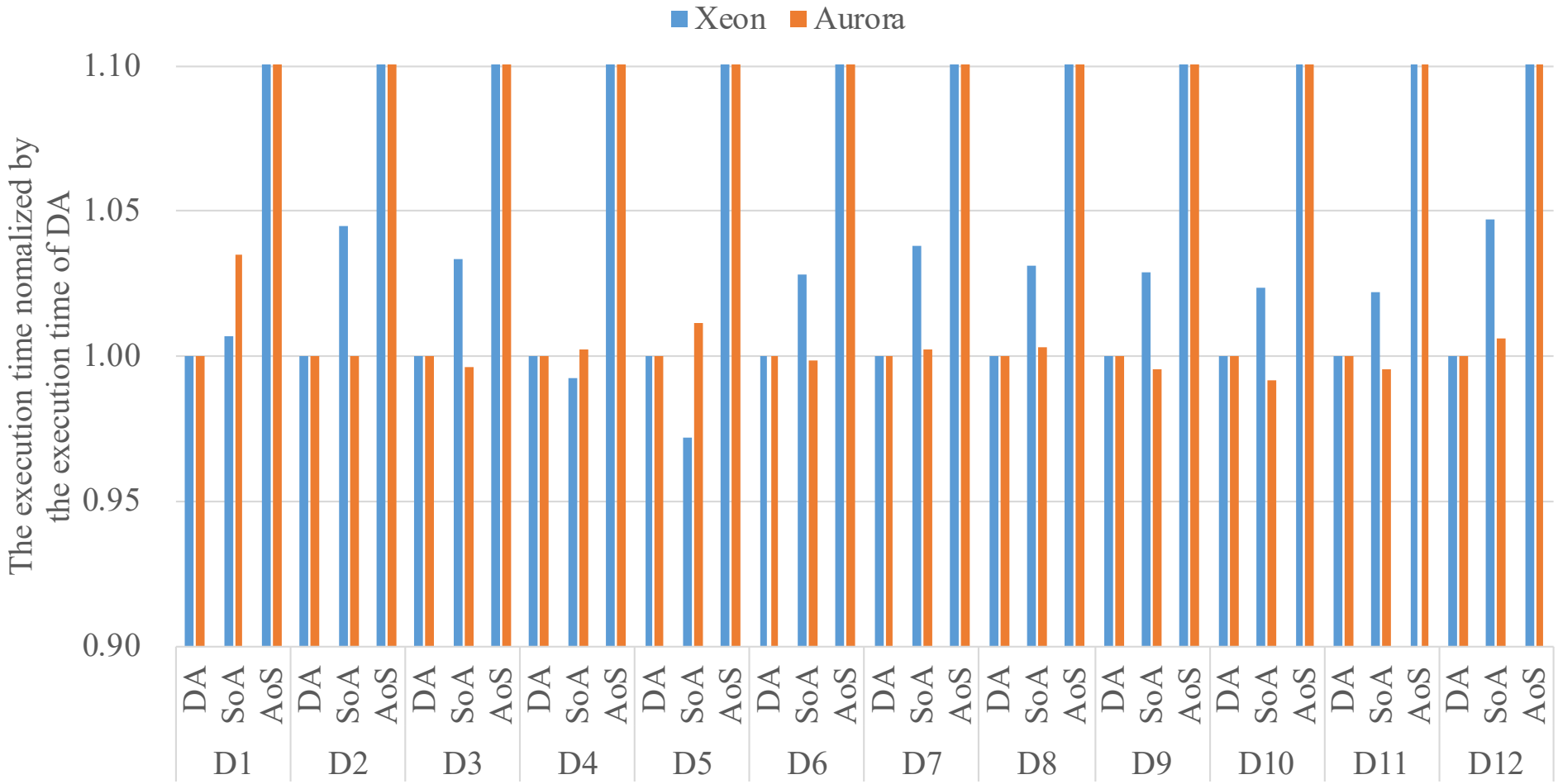
- 12 computational domains that have possibilities to be suffered from significant damage of tsunamis
  - Computational domain composed of five-level resolutions: D1
  - Computational domains composed of four-level resolutions: D2~D12

Domain	Grid size (m)	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
		Number of grids											
1	810	1566720	2012940	2118960	2162160	2146320	2146320	2146320	2012940	2012940	2118600	2118600	2118600
2	270	1663200	1703484	1395072	551448	556416	944748	1421856	2098980	1703484	2149200	1246716	1246716
3	90	2887920	1807560	2284092	1524456	1315260	2023920	2704248	2475180	1807560	3178800	1409760	1301544
4	30	5969016	4983372	6295104	5405220	6525900	5757768	4133700	3424248	4983372	1990728	4539600	2907468
5	10	22795020	-	-	-	-	-	-	-	-	-	-	-

Evaluation results

TOHOKU UNIVERSITY

Execution times



Summary of the evaluation result

TOHOKU UNIVERSITY

Performances

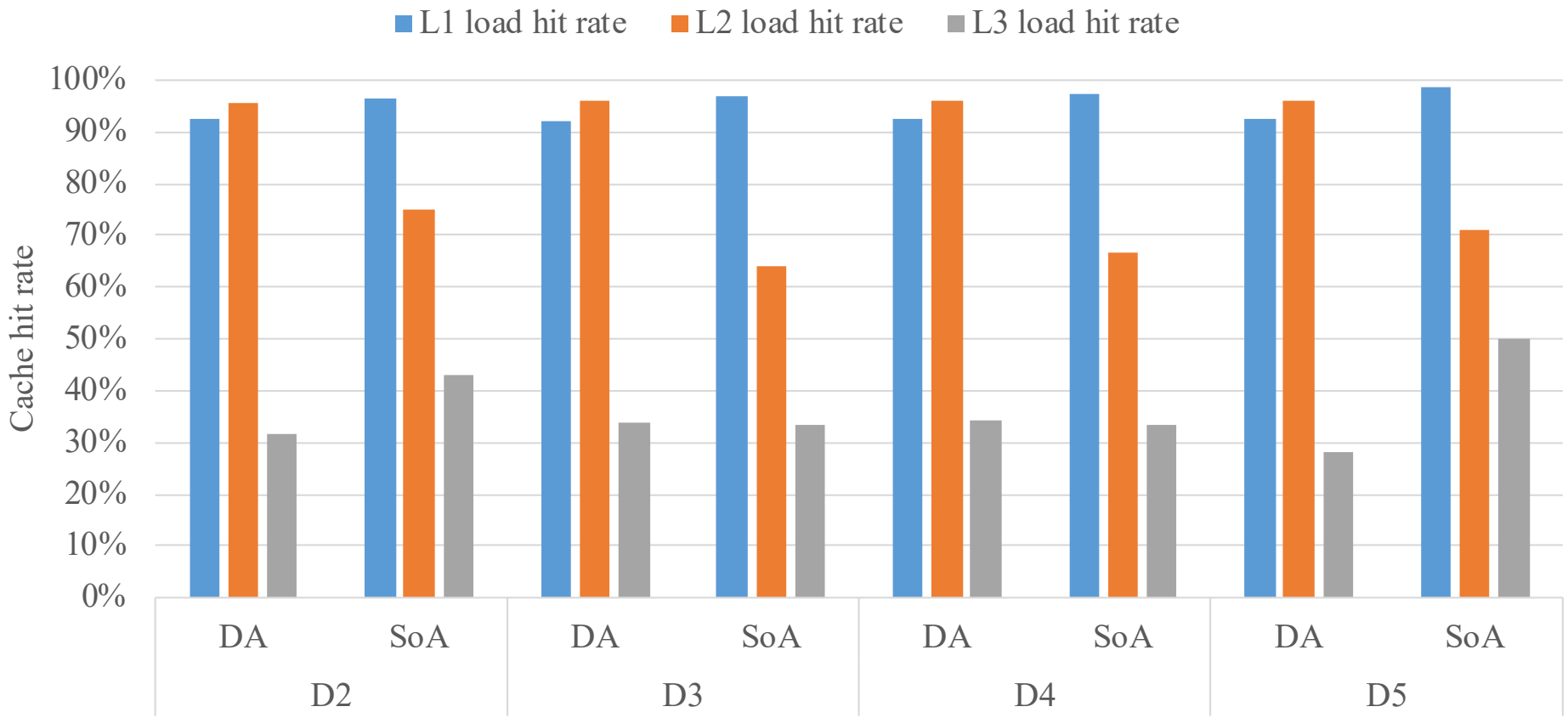
- DA and SoA achieves high performance
- AoS is always lowest performance
  - The overhead of accesses to scatteredly placed data elements
  - The requirement of much clock cycles for vector gather instructions

The variation of the performance

- The performance of the tsunami simulation is changed by
  - Computing systems
  - Data structure used in the code
  - Input data (target areas)
- Results indicates the necessity of appropriate selection of the data structure based on computing systems and input data

Detailed analysis

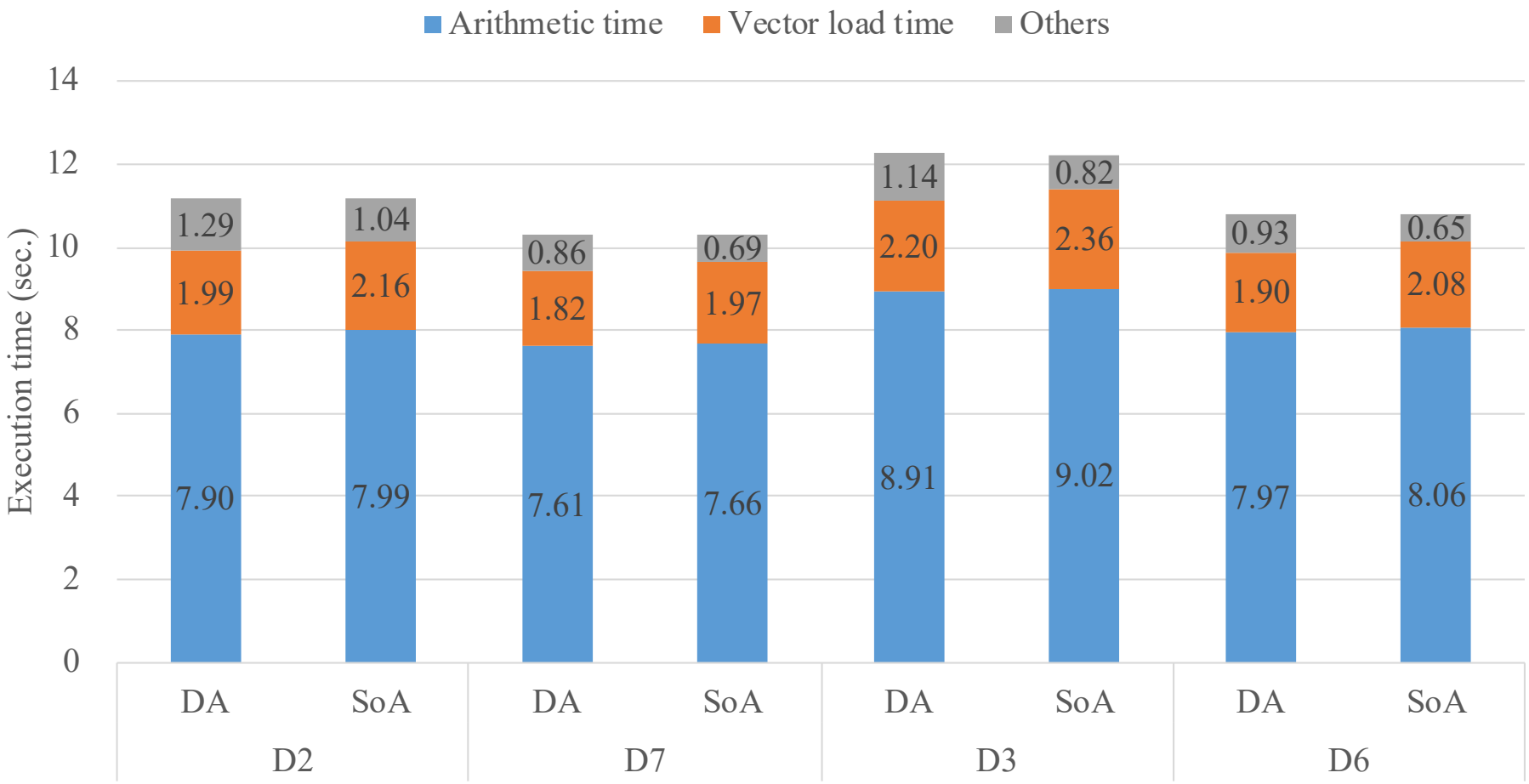
Cache load hit ratios on Xeon



Detailed analysis

TOHOKU UNIVERSITY

Breakdowns of the execution time on SX-Aurora TSUBASA



Summary of the detailed analysis

TOHOKU UNIVERSITY

Uncertainty of performance variation factor

- Xeon-based system
  - L1 cache load hit ratios are improved by using structure of arrays with all the inputs
  - L3 cache load hit ratios vary independent of data structure and inputs
- SX-Aurora TSUBASA
  - The arithmetic time and vector load time increase in DA and SoA with all the inputs
  - The execution time other than the arithmetic time and vector load time decreases with all the inputs

Necessity of more detailed analysis

- The complicated factors may affect the performance variation
- It is important to incorporate the selection of data structure into tuning techniques

Conclusions

- Tsunami simulation needs to be accelerated for real-time executions regardless of target areas

- This research focuses on selecting the data layout used in the tsunami simulation for efficient memory accesses
  - DA, AoS, and SoA

- From evaluation results,
  - Combinations of the computing systems, data layouts, and input data change the performance of the tsunami simulation
  - The factor of performance variation is unclear from the simple analysis
  - More detailed analysis is necessary for establishing a technique to tune the data layout

Future work

TOHOKU UNIVERSITY

- More detailed analysis
  - Factors affecting the performance variation
  - Clarification of the relationship among performance, input data, data layout, and computing systems
- Establishment of tuning technique
  - Selection of data layout based on;
    - Computing systems
    - Target applications
    - Input data

Contacts

- Takumi Kishitani
  - Mail: t-kishitani@nec.com
- Kazuhiko Komatsu
  - Mail: komatsu@tohoku.ac.jp
- Masayuki Sato
  - Mail: masa@tohoku.ac.jp
- Akihiro Musa
  - Mail: a-musa@nec.com
- Hiroaki Kobayashi
  - Mail: koba@tohoku.ac.jp