# Optimizations for the Himeno Benchmark on Vector Computing System SX-Aurora TSUBASA

Akito Onodera[1)], Kazuhiko Komatsu[1)], Takumi Kishitani[1)2)], Masayuki Sato[1)], Yoko Isobe[2)], Hiroaki Kobayashi[1)]
1)Tohoku University, 2)NEC Corporation

# Background

## Importance of Vector Processing

- Modern processors improve computational performance by vector processing
  - ✓ NEC SX-Aurora TSUBASA, Intel Skylake, Fujitsu A64FX, etc.
- Many applications involve data-level parallelism best suited for vector processing
  - ✓ Large-scale numerical simulations, big data analysis, etc.

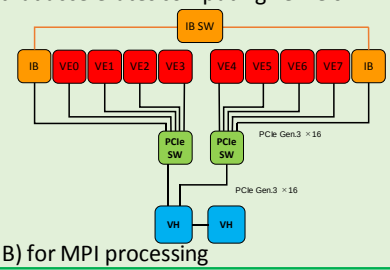
→ **The use of vector processing is necessary for accelerating applications**

## Vector Computing System SX-Aurora TSUBASA

- Composed of Vector Hosts (VHs) and Vector Engines (VEs) [1]
  - ✓ VH: Standard x86 processor that is responsible for executions of OS-related tasks
  - ✓ VE: NEC-developed processor that accelerates computing kernels
- Applications are executed on VEs
  - ✓ VEs request VHs to execute applications only when them need system calls

**◎ System Overview of A300-8**

- 2 VHs and 8 VEs
- One VH and 8 VEs are connected by PCI Express via 2 PCIe switches
- VEs are connected by Infiniband (IB) for MPI processing




**◎ Architecture of the VE**

- 8-vector cores
- Vector registers for 256-elements data

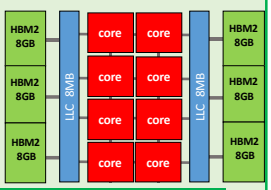
→ **Need to utilize the multiple cores and the long vector length**

- A high bandwidth Last-Level Cache (LLC)

→ **Need to utilize the LLC bandwidth**

- A main memory by 6 modules of High Bandwidth Memory (HBM2)

→ **Need to utilize the main memory bandwidth**

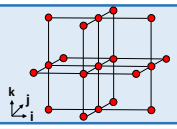



**Exploitation of the potentials of SX-Aurora TSUBASA by using the Himeno benchmark**

# Optimizations for the Himeno Benchmark

## Overview of the Himeno Benchmark

- The benchmark solves Poisson's equation using the Jacobi iteration method [2]
- The Jacobi iteration method in this benchmark
  - ✓ 19-point stencil calculation nested by i, j, and k
  - ✓ The memory-intensive kernel (Code B/F: 3.74)




## Three Optimizations for VE

### ① Store highly reusable data in the LLC

- ✓ Utilize the LLC whose bandwidth is higher than that of HBM2
- ✓ An array of the pressure variables is referenced 19 times per iteration
  - ➢ Force to place the array in the LLC using compiler directive "retain"

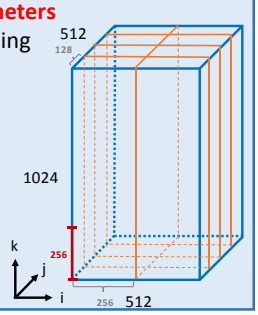
### ② Reduce the loop overheads

- ✓ Apply loop unrolling using compiler directive "outerloop_unroll"
  - ➢ Long loops with nested structures in the kernel
  - ➢ Use of plenty of vector registers of vector processors
    - Set the unroll time as large as possible without register spilling

### ③ Tune the domain decomposition parameters

- ✓ Important for efficient vector processing
  - ➢ Longer vector length
  - ➢ Higher LLC hit ratio

- ☐ Keeping the length in the k direction larger than 256, while
- ☐ Increasing the decomposition of the j direction compared with i direction to achieve the high LLC hit ratio

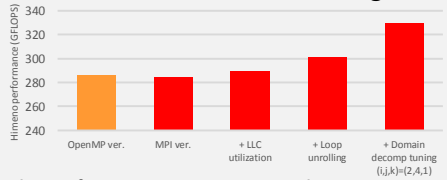



# Performance Evaluation

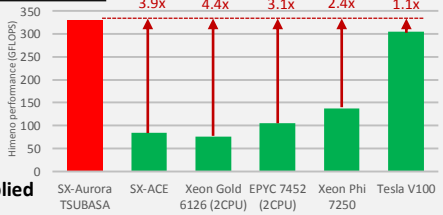
## Experimental Environment

- SX-Aurora TSUBASA A300-8
  - ✓ VH: Intel Xeon Gold 6126 x 2
    - ➢ Peak performance / socket: 1.0 TFLOPS
    - ➢ Peak memory bandwidth: 0.13 TB/s
  - ✓ VE: Type 10B x 8
    - ➢ Peak performance / node: 2.15 TFLOPS
    - ➢ Peak memory bandwidth / node: 1.2 TB/s
    - ➢ Peak LLC bandwidth / node: 3.2 TB/s
  - ✓ Infiniband FDR
- CentOS 7.5.1804
- VEOS 2.4.0
- NEC C/C++ Compiler 3.0.1
- NEC MPI 2.5.0
- Himeno benchmark version 3.0
  - ✓ C language
  - ✓ MPI static allocate version
  - ✓ Problem size: XL (512x512x1024)
  - ✓ Initial decomposition: (i,j,k)=(2,2,2)

## Single-Node Evaluation

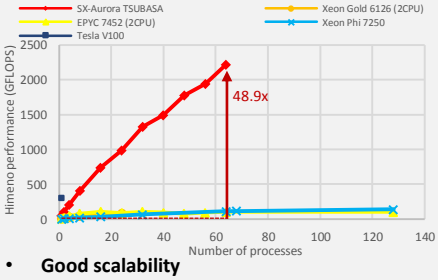



- **The performance improves as each optimization is applied**
  - ✓ The LLC hit ratio increases from 44.3% to 49.6%
  - ✓ The loop overhead decreases greatly by loop unrolling
  - ✓ The average vector length is 255.18 by tuning the decomposition
- **1.15x performance improvement compared with that of OpenMP version [3]**
- **7.7% of the peak performance in the MPI version**




- **SX-Aurora TSUBASA achieves the highest performance**
  - ✓ Utilize HBM2 and LLC bandwidth efficiently

## Multi-Node Evaluation




- **Good scalability**
  - ✓ 48.9x speedup and 76% parallel efficiency with 64 processes
- **6.5% of the peak performance with 8VEs**

# Conclusions & Future Work

- SX-Aurora TSUBASA has achieved high performance of the Himeno benchmark by the optimizations
  - ✓ Important to execute efficient vector processing with effective use of hardware resources
- The sustained performance of the Himeno benchmark on SX-Aurora TSUBASA can be further accelerated by introducing hybrid execution with VHs and VEs

**References**
[1] Y. Yamada et al., "Vector Engine Processor of NEC 's Brand-New Supercomputer SX-Aurora TSUBASA", Hot Chips 30, Aug 2018
[2] Himeno benchmark, http://i.riken.jp/en/supercom/documents/himenobmt/
[3] K. Komatsu et al., "Performance Evaluation of a Vector Supercomputer SX-Aurora TSUBASA", SC18, Nov 2018

**Acknowledgments**
This research was partially supported by MEXT Next Generation High-Performance Computing Infrastructures and Applications R&D Program, entitled "R&D of A Quantum-Annealing-Assisted Next Generation HPC Infrastructure and its Applications."