\Orchestrating a brighter world NEC

The 4th

Aurora Forum

ISC2019 in Frankfurt

June 17th, 2019

NEC Corporation

Orchestrating a brighter world

NEC brings together and integrates technology and expertise to create the ICT-enabled society of tomorrow.
We collaborate closely with partners and customers around the world, orchestrating each project to ensure all its parts are fine-tuned to local needs.

Every day, our innovative solutions for society contribute to greater safety, security, efficiency and equality, and enable people to live brighter lives.

Agenda

[9:00 - 9:05] Opening Greetings

[9:05 - 9:45] Introduction of SX-Aurora TSUBASA

How SX-Aurora TSUBASA can contribute to your R&D

-HPC and AI case studies-

Masashi Ikuta (NEC Corporation)

[9:45 - 10:05] Partner session

NEC & HPE join forces

Aurora Vector Engine meets global HPC leader

Mr. Bill Mannel (Hewlett Packard Enterprise)

[10:05～10:35] In-Memory Query Processing on NEC Aurora TSUBASA

Prof. Wolfgang Lehner (Technische Universität Dresden)

[10:35～10:50] Ease of code tuning on NEC SX-Aurora TSUBASA

Raghunandan Mathur (NEC Corporation)

[10:50～11:00] Introduction to the Aurora Forum Website

Takenori Mishio (NEC Corporation)

[11:00] Closing


Orchestrating a brighter world
NEC

\Orchestrating a brighter world NEC

Aurora Forum

Jun 17, 2019

NEC Corporation

Aurora Forum

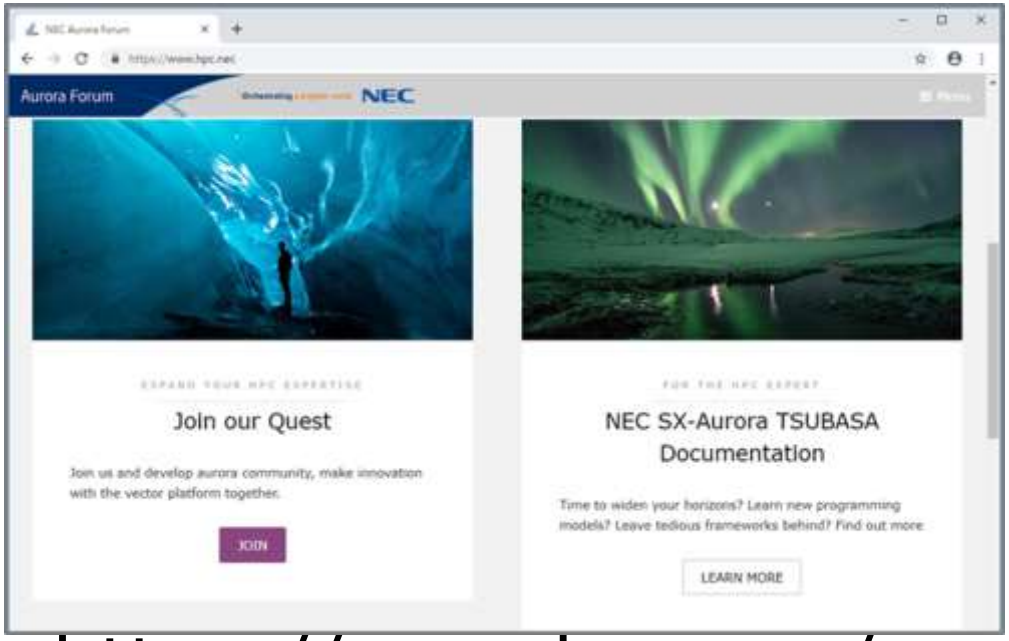
- Thank you for supporting SX-Aurora TSUBASA
- You are always welcome to join our quest

"Supercomputer close to everyone"

Aurora Forum Web

Find more information in Aurora Web Forum

- Latest documents
- Announcements
- Tuning guide
- TIPS
- Exchange information with colleagues



https://www.hpc.nec/

Orchestrating a brighter world

NEC

Aurora Forum

Introduction of SX-Aurora TSUBASA

How SX-Aurora TSUBASA can contribute to your R&D
- HPC and AI case studies -

Jun 17, 2019

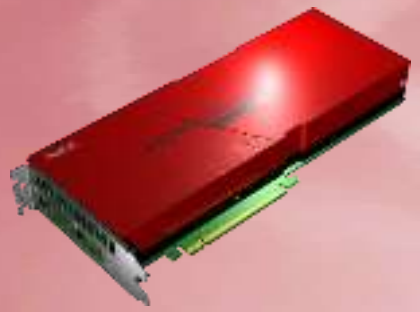
NEC Corporation

Your image of supercomputers

Fast, but

- Expensive
- Large and heavy
- Power consuming
- Something special (only for special people)

You need to change your supercomputer image



Vector Engine

SX-Aurora TSUBASA Tower model

SX-Aurora TSUBASA Rack mount model

Fast, and

- Economical
- Compact (small and light)
- Energy saving
- Nothing special (for everyone)

Comparison

No.1 system 20 years ago

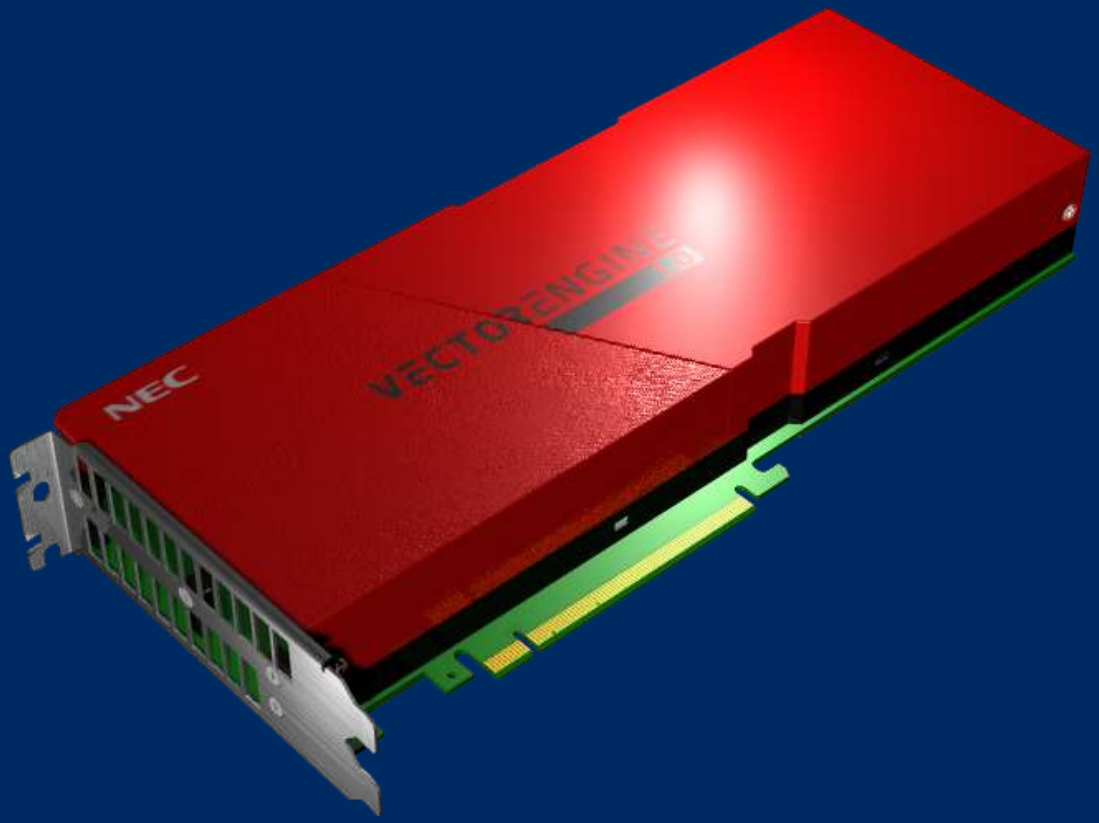
- Linpack : 2.12 Tflops
- Peak performance: 3.15 Tflops
- Power : 850kW
- Machine room : 250m^2 (swimming pool size)

Today's Vector Engine Card

- Linpack : 1.86 Tflops
- Peak performance: 2.45 Tflops
- Power : <300W
- ~~Machine room~~ : desk side

https://www.top500.org/system/168752

Vector Processor on The Card



- New Developed Vector Processor
- PCIe Card Implementation
- 8 cores / processor
- 2.45TF performance
- 1.22TB/s memory bandwidth
- Normal programing with Fortran/C/C++

Yet another accelerator?

Yes

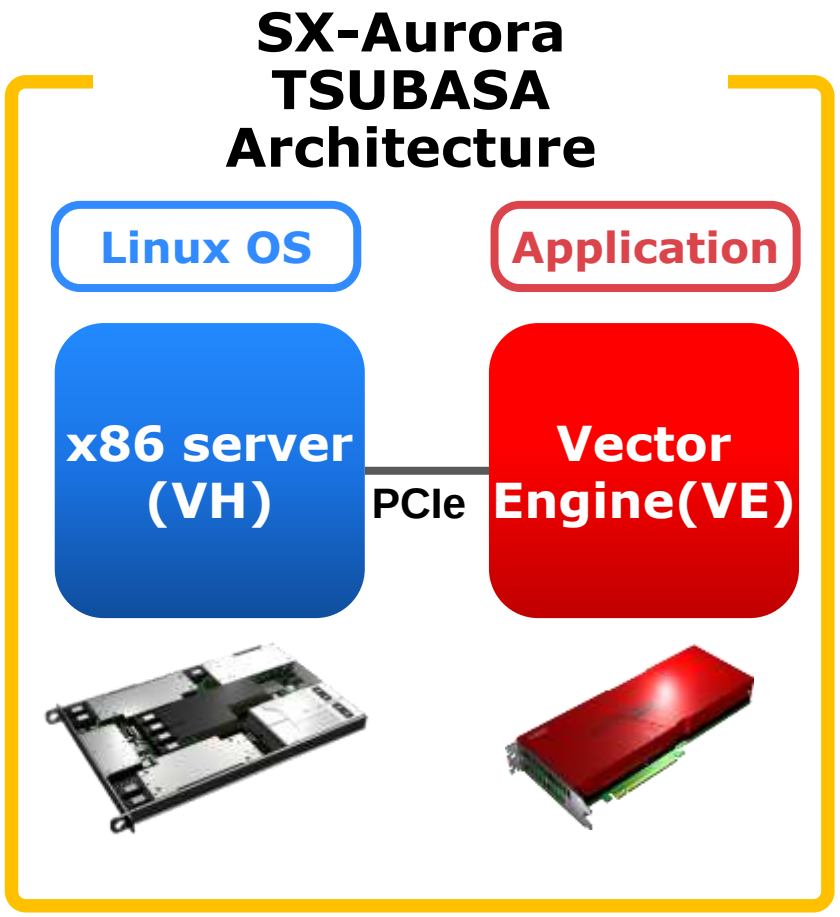
- Works with x86 host server (not standalone)
- PCI card (same form factor as GPGPU card)
- Accelerates computation
- Offload programing model

No

- Entire program can run on VE
  (not only offload model)
- No special language

New Architecture

- SX-Aurora TSUBASA = Standard x86 + Vector Engine
- Linux + standard language (Fortran/C/C++)
- Enjoy high performance with easy programming



Hardware

- Standard x86 server + Vector Engine

Software

- Linux OS
- Automatic vectorization compiler
- Fortran/C/C++
  → No special programming like CUDA

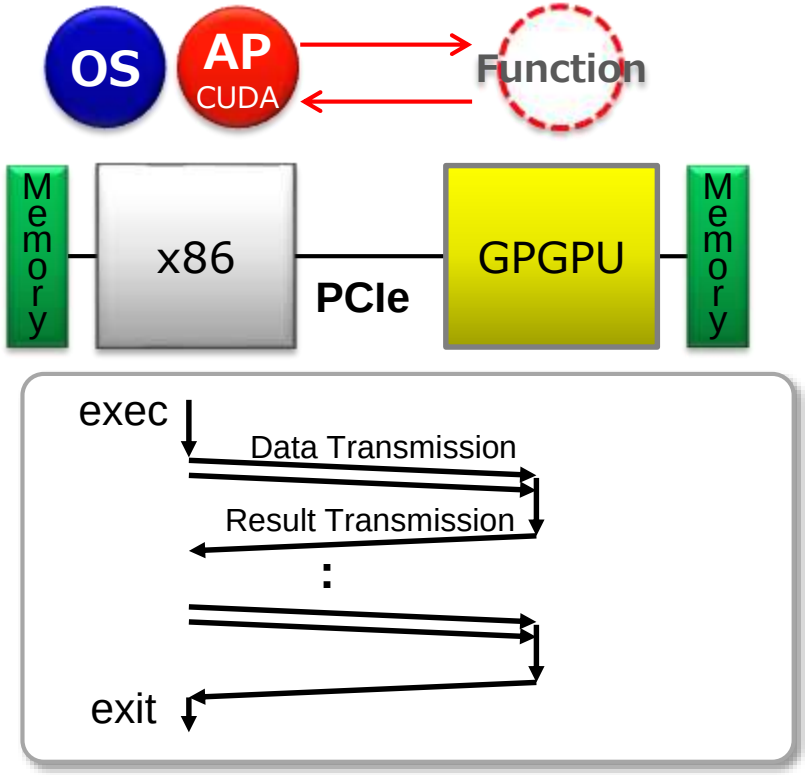
Interconnect

- InfiniBand for MPI
- VE-VE direct communication support

Easy programming (standard language) ▶ **Automatic vectorization compiler** ▶ **Enjoy high performance**

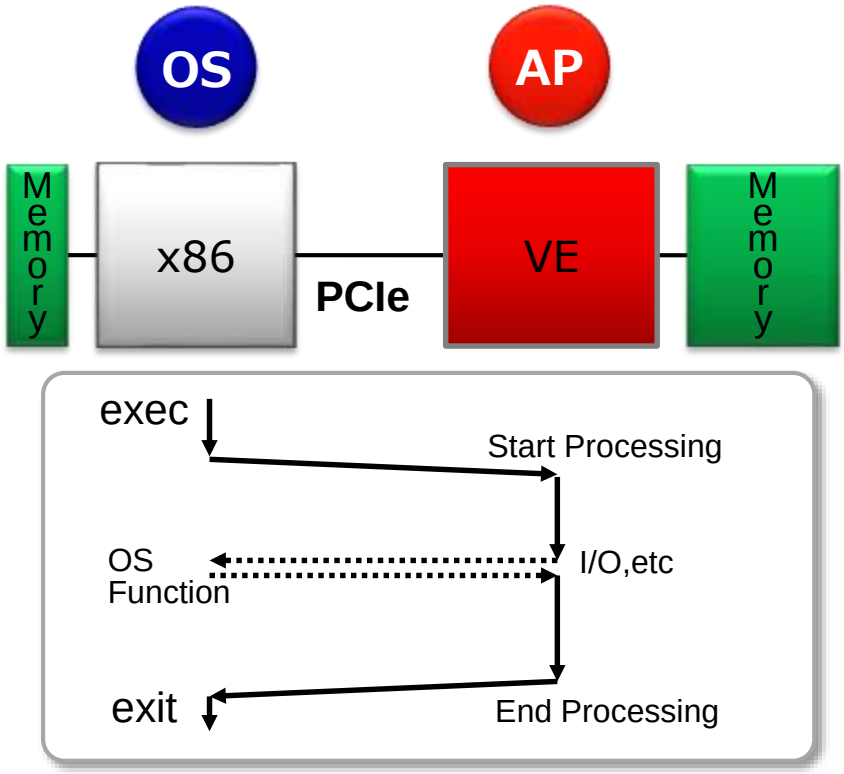
GPGPU and VE

GPGPU Architecture (Function offload model)



Frequent PCIe transmission

Aurora Architecture (OS offload model)



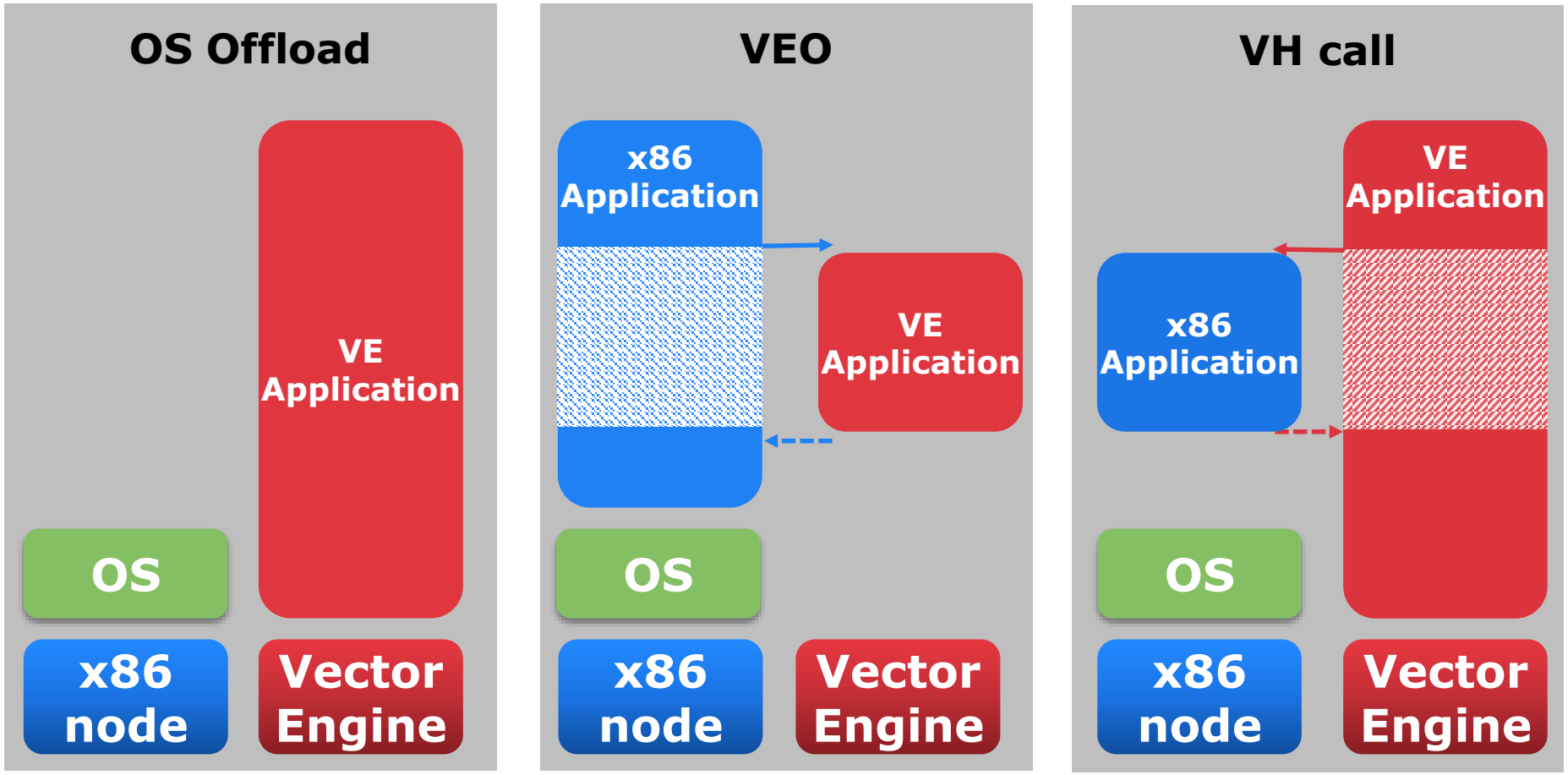
Whole AP is executed on VE

disadvantage
- PCIe bottleneck
- Programming difficulty

Advantage
- Avoiding PCIe bottleneck
- Standard language

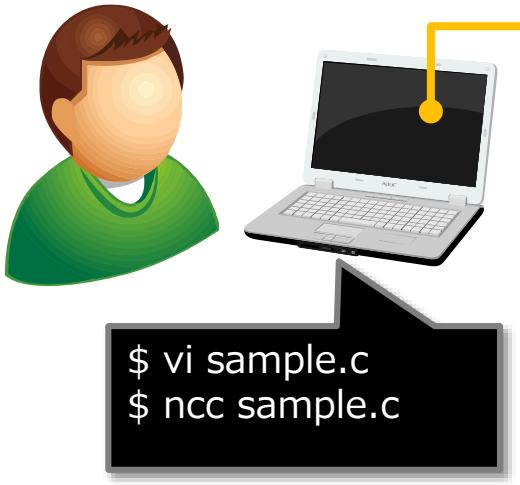
Offload models

Run the application in the right way



Usability

Programing Environment



Vector Cross Compiler

- automatic vectorization
- automatic parallelization

Fortran:	F2003, F2008
C/C++:	C11/C++14
OpenMP:	OpenMP4.5
Library:	MPI 3.1, libc, BLAS, Lapack, etc
Debugger:	gdb, Eclipse parallel tools platform
Tools:	PROGINF, FtraceViewer

Execution Environment

execution

Vector programming

- Standard C/Fortran programming (No need to change program)
- Automatic vectorization feature and various tools help vectorization

```
void matmul(float *A, float *B, float *C, int l, int m, int n){
    int i, j, k;
    for (i = 0; i < l; i++) {
        for (j = 0; j < n; j++) {
            float sum = 0.0;
            for (k = 0; k < m; k++)
                sum += A[i * m + k] * B[k * n + j];
            C[i*n+j] = sum;
        }
    }
}

void alloc_matrix(float **m_h, int h, int w){
    *m_h = (float *)malloc(sizeof(float) * h * w);
}

// other function definitions …

int main(int argc, char *argv[]){
    float *Ah, *Bh, *Ch;
    struct timeval t1, t2;

    // prepare matrix A
    alloc_matrix(&Ah, L, M);
    init_matrix(Ah, L, M);
    // do it again for matrix B
    alloc_matrix(&Bh, M, N);
    init_matrix(Bh, M, N);
    // allocate spaces for matrix
    alloc_matrix(&Ch, L, N);

    // call matmul function
    matmul(Ah, Bh, Ch, L, M, N);

    return 0;
}
```

No modification is necessary for vectorization

Just compile, and loops are vectorized automatically

[Compiler diagnostic message]

```
$ ncc sample.c -O4 -report-all -fdiag-vector=2
ncc: opt(1589): sample.c, line 11: Outer loop moved inside inner loop(s).: j
ncc: vec( 101): sample.c, line 11: Vectorized loop.
ncc: opt(1592): sample.c, line 13: Outer loop unrolled inside inner loop.: k
ncc: vec( 101): sample.c, line 13: Vectorized loop.
ncc: vec( 128): sample.c, line 14: Fused multiply-add operation applied.
ncc: vec( 101): sample.c, line 15: Vectorized loop.
```

[Format list]

```
 8:           void matmul(float *A, float *B, float *C, int l, int m, int n){
 9:               int i, j, k;
10: +-------->    for (i = 0; i < l; i++) {
11: |X------->        for (j = 0; j < n; j++) {
12: ||                    float sum = 0.0;
13: ||V------>            for (k = 0; k < m; k++)
14: ||V----- F                sum += A[i * m + k] * B[k * n + j];
15: ||                    C[i*n+j] = sum;
16: |X------          }
17: +-------      }
18:           }
```

NEC Library Collection

NLC has more mathematical functionalities than competitors, but no deep learning functionalities yet. ← under planning to support

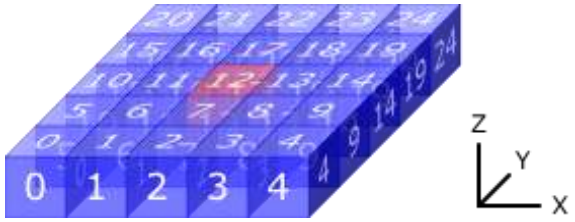
		NLC	MKL	CUDA
Linear Algebra	Dense Matrix basic operations, simultaneous equations, eigen-problems, etc.	✓	✓	✓
	Sparse Matrix basic operations, simultaneous equations, eigen-problems, etc.	✓	✓	✓
Function Transform	Fourier	✓	✓	✓
	Real-to-Real (DCT, etc.)	✓	✓	
	Laplace, Wavelet, etc.	✓		
Statis-tics	Random Number Gen.	✓	✓ w/o MPI	✓ w/o MPI
	Multivariate, Regression, etc.	✓		
Others	Sorting	✓		
	Special Functions	✓		
	Integrals, Derivatives, etc.	✓		
	Stencil Code	✓		
	Deep Learning		✓	✓

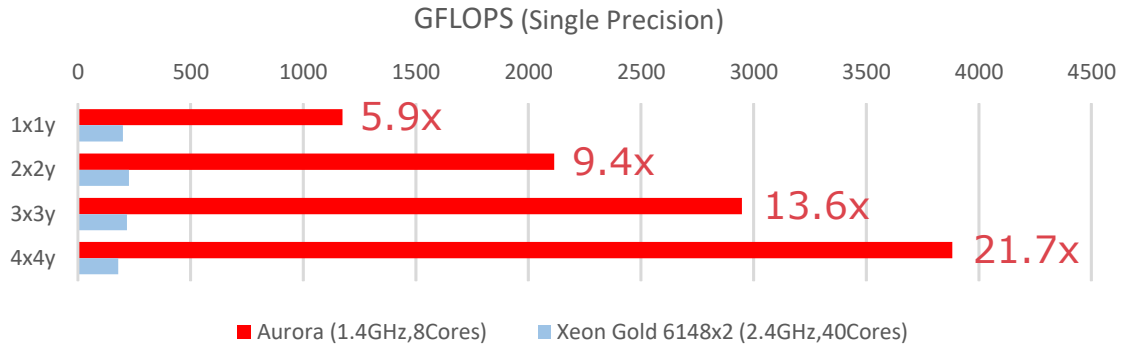
Stencil Code Accelerator: Performance (1)

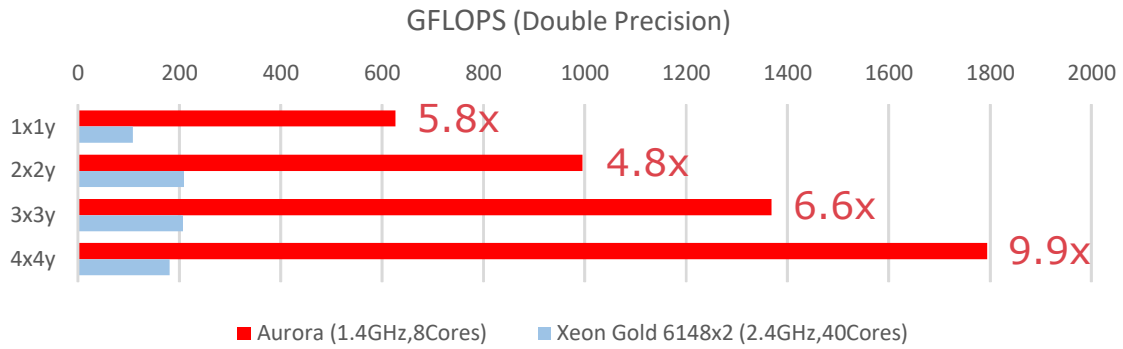
Aurora with SCA overwhelms Xeon Skylake 40 cores: 4.8x-21.7x.

Benchmark 1

- Stencil Shape: XY-planer, size 1-4 ← most commonly used in image processing
- Data Size: 1024 x 1024
- Processors:
  - Aurora 1.4 GHz, 8 cores
  - Xeon Gold 6148 x 2 sockets (Skylake 2.4 GHz, 40 cores)







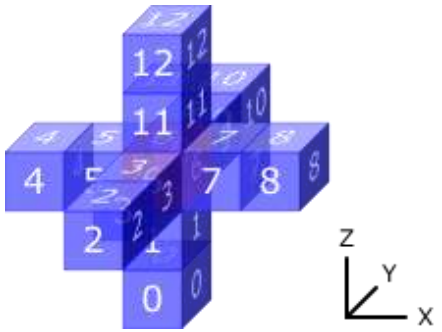
- The performance is brought by reducing memory load using vector registers like cache.

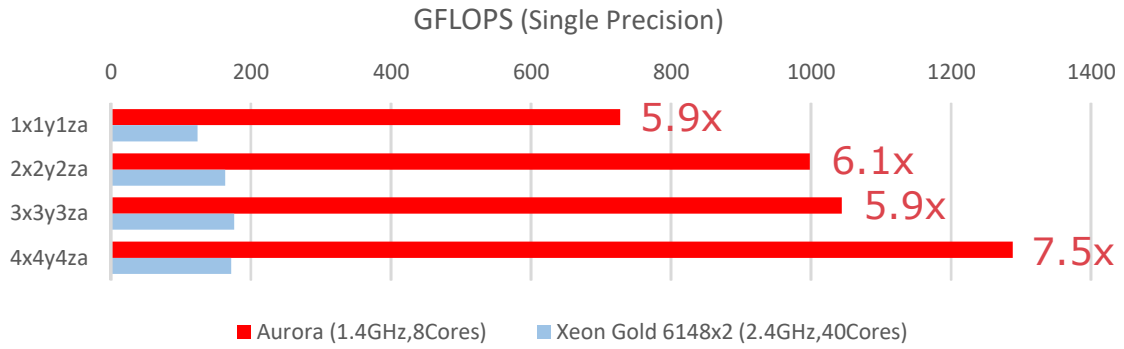
Stencil Code Accelerator: Performance (2)

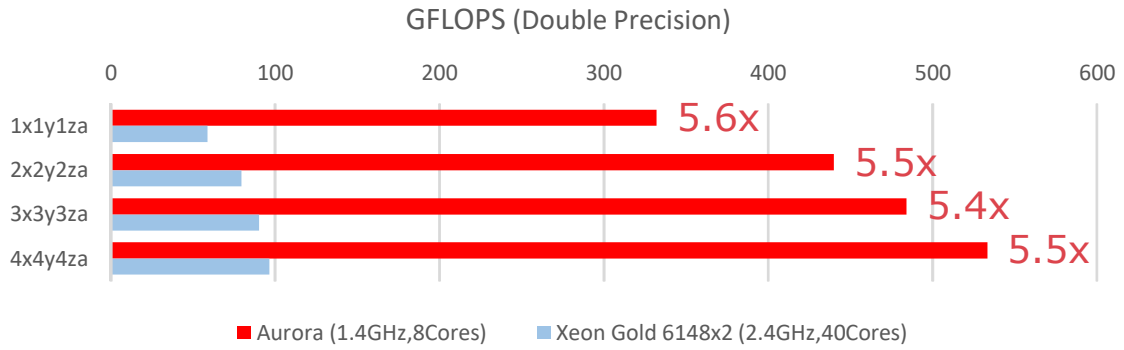
Aurora with SCA overwhelms Xeon Skylake 40 cores: 5.4x-7.5x.

Benchmark 2

- Stencil Shape: XYZ-axial, size 1-4 ← most commonly used in scientific simulations
- Data Size: 1024 x 1024 x 1024
- Processors:
  - Aurora 1.4 GHz, 8 cores
  - Xeon Gold 6148 x 2 sockets (Skylake 2.4 GHz, 40 cores)







- The performance is brought by reducing memory load using vector registers like cache.

HeteroSolver (1)

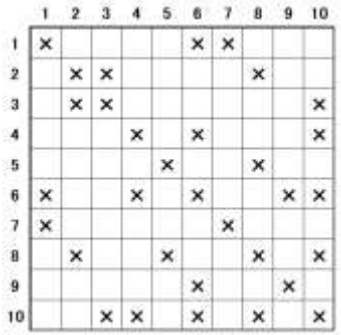
HeteroSolver solves simultaneous linear equations defined with a sparse matrix under VE-VH cooperation.

Process of Direct Sparse Solver

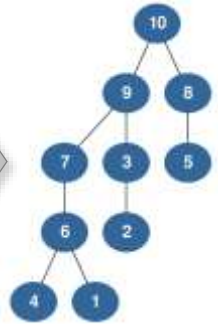
$$Ax = b$$

- Preprocessing Phase
  - Analyzes the sparsity of the matrix A using graph theory.

  ← Unable to be well vectorized.
- Numeric Factorization Phase
  - Factorizes the matrix (e.g. LU factorization).
- Solution Phase
  - Computes the solution x.



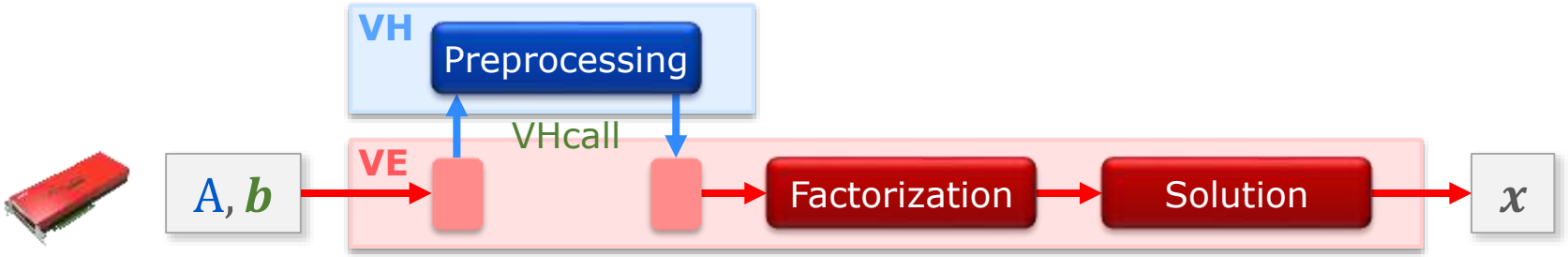
Matrix A



Graph

Optimization using VE-VH cooperation

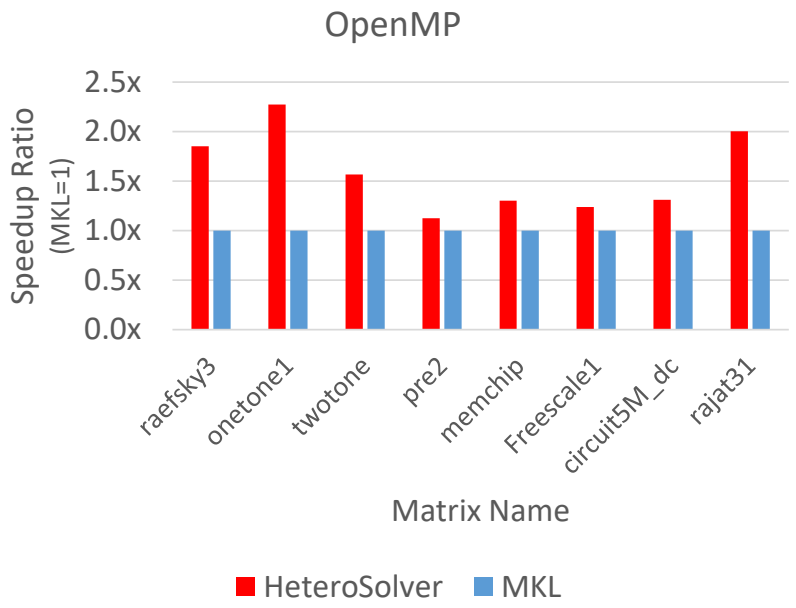
- The preprocessing phase is executed on VH automatically.

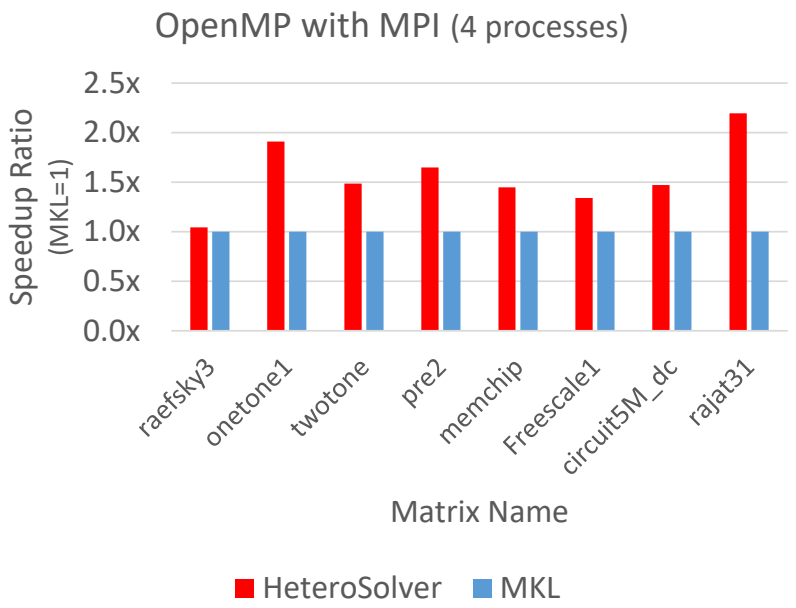


HeteroSolver (2)

Aurora with HeteroSolver beats Xeon Skylake 40 cores with MKL.

Benchmark



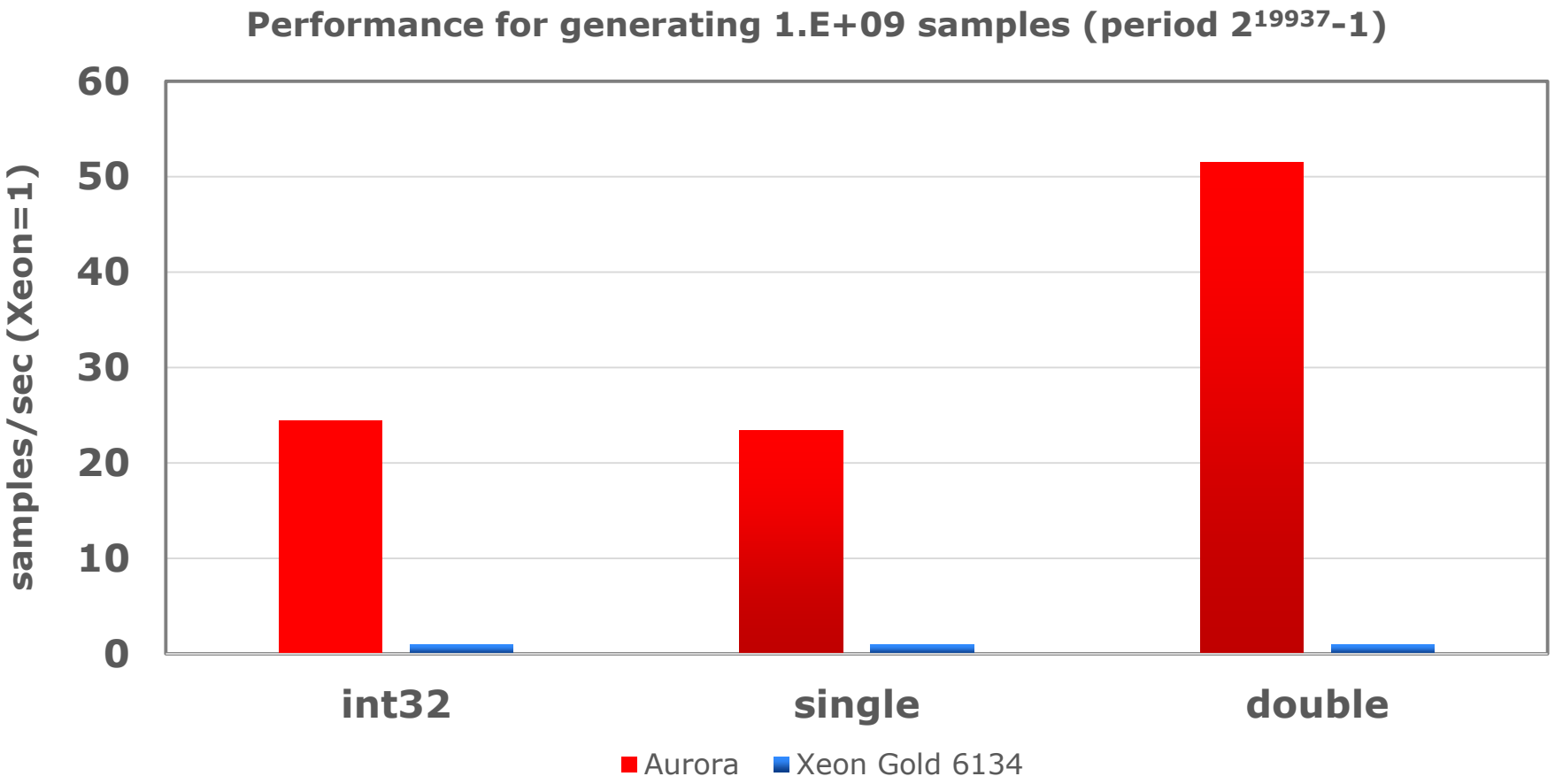


	Platform
HeteroSolver	SX-Aurora TSUBASA (1.4 GHz, 8 Cores)
MKL	Xeon Gold 6148 x 2 sockets (Skylake 2.4 GHz, 40 cores)

Matrices: SuiteSparse Matrix Collection (https://sparse.tamu.edu/)

Random Generation (MT)

Aurora shows over 50x performance compared to Xeon

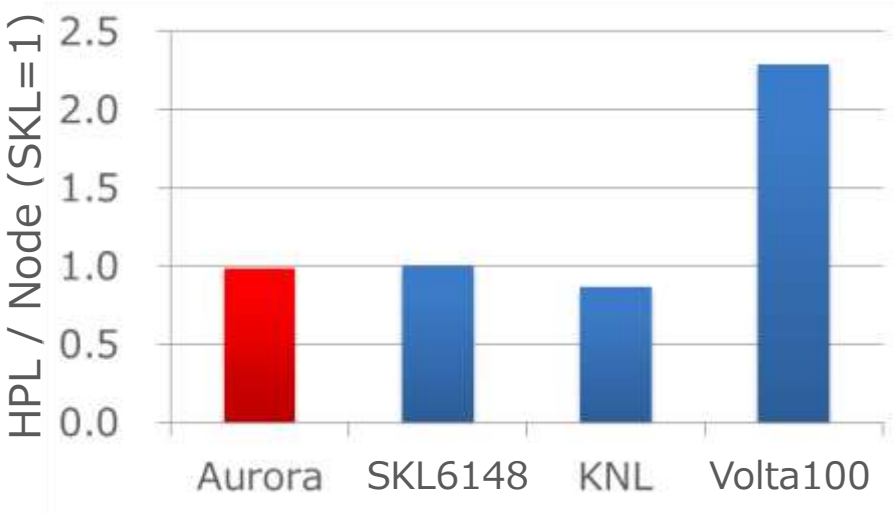


- Aurora: Vector Engine Type 10-B (1.4GHz, 8core) using MT19937-64 (period $2^{19937}-1$)
- Xeon 6134: Intel Xeon Gold 6134 3.20GHz using MKL, SFMT19937 (period $2^{19937}-1$)
- cuRAND does not support period $2^{19937}-1$

HPL and STREAM

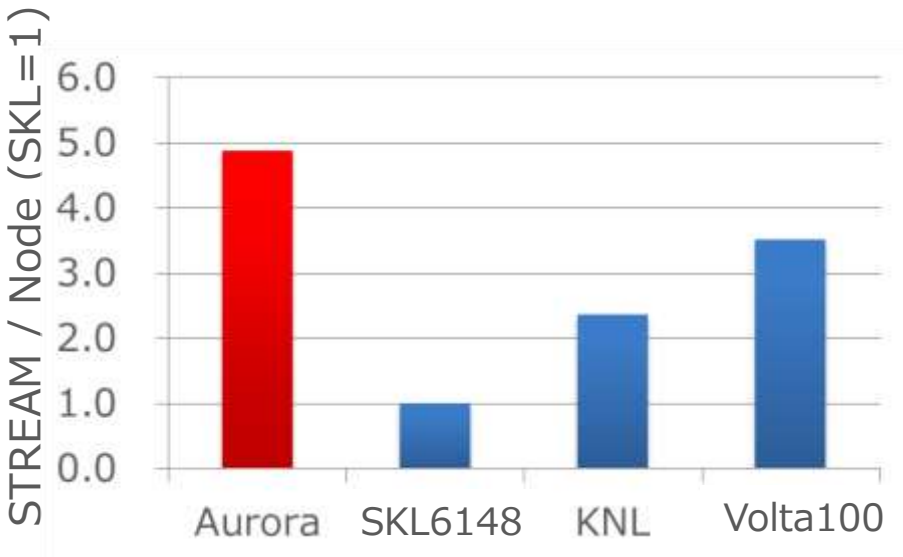
- HPL : Aurora provides competitive FLOPS capability
- STREAM : Aurora provides highest sustained memory bandwidth

HPL / Node



- Aurora provides same range HPL sustained performance as SKL/KNL

STREAM / Node

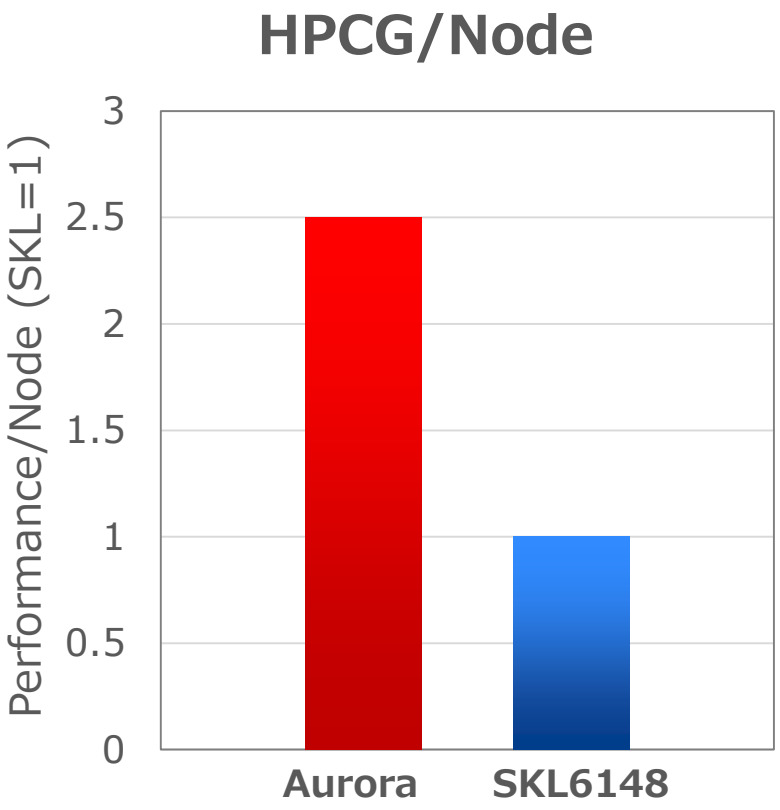


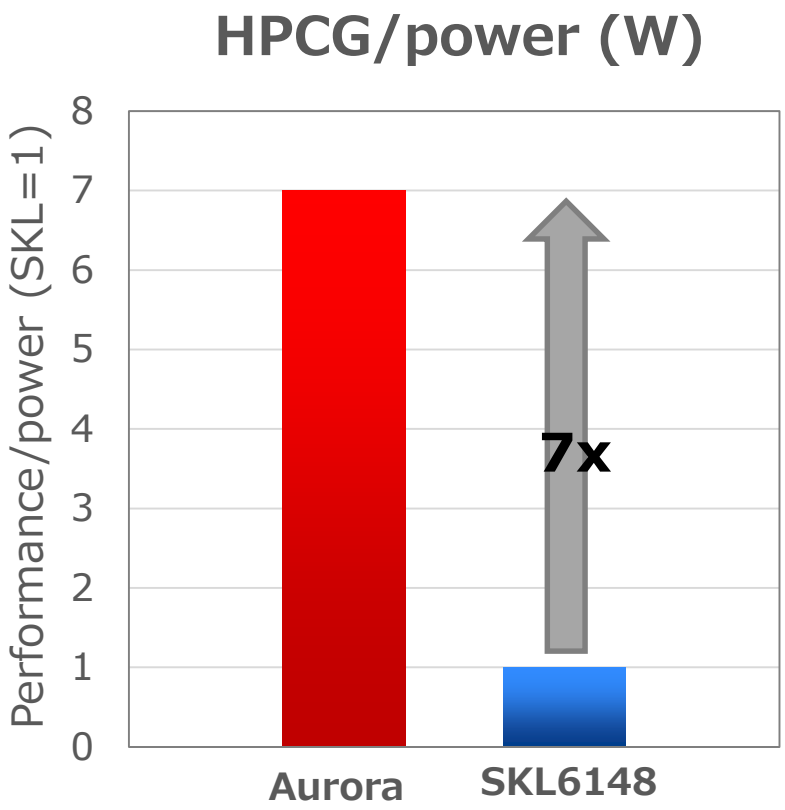
- Aurora provides the highest memory bandwidth

- Aurora is Vector Engine Type 10-B (1.4GHz, 8core)
- SKL is Intel Skylake 6148 Xeon x2/node
- KNL is Intel Knight Landing x1/node
- V100 is NVIDIA Tesla V100 x1/node

HPCG

Performance/power of Aurora shows 7 times better than SKL





- Aurora is Vector Engine Type 10-B (1.4GHz, 8core)
- SKL is Intel Skylake 6148 Xeon x2/node

Product Lineup

- Product: VEs + x86/Linux server
- 3 series, A100, A300, and A500 for various users

Supercomputer Model

- For large scale configuration
- **DLC with 40℃ water**

Rack Mount Model

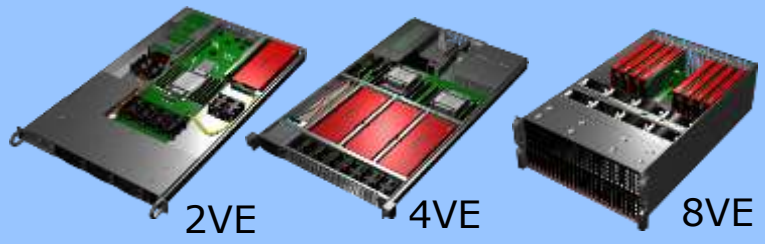
- **Flexible configuration**
- Air Cooled

Tower Model

- For developer/programmer
- **Office room use**

A500 series

A300 series



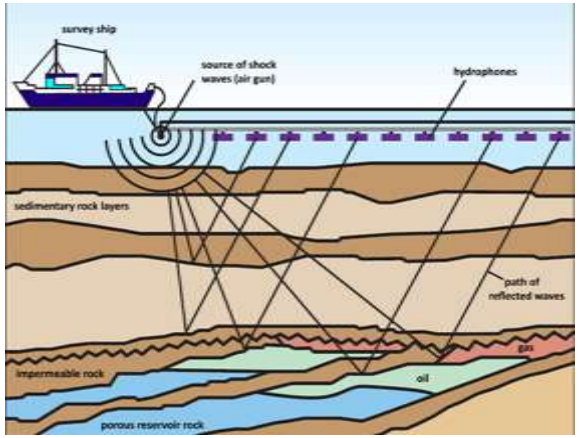
A100 series

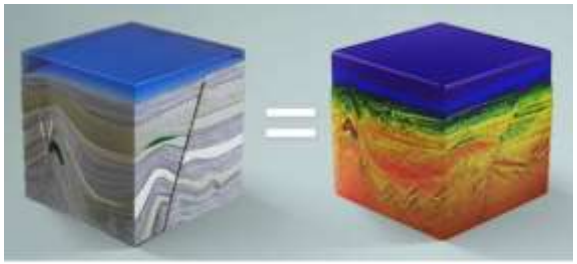
Case Study

Oil & Gas

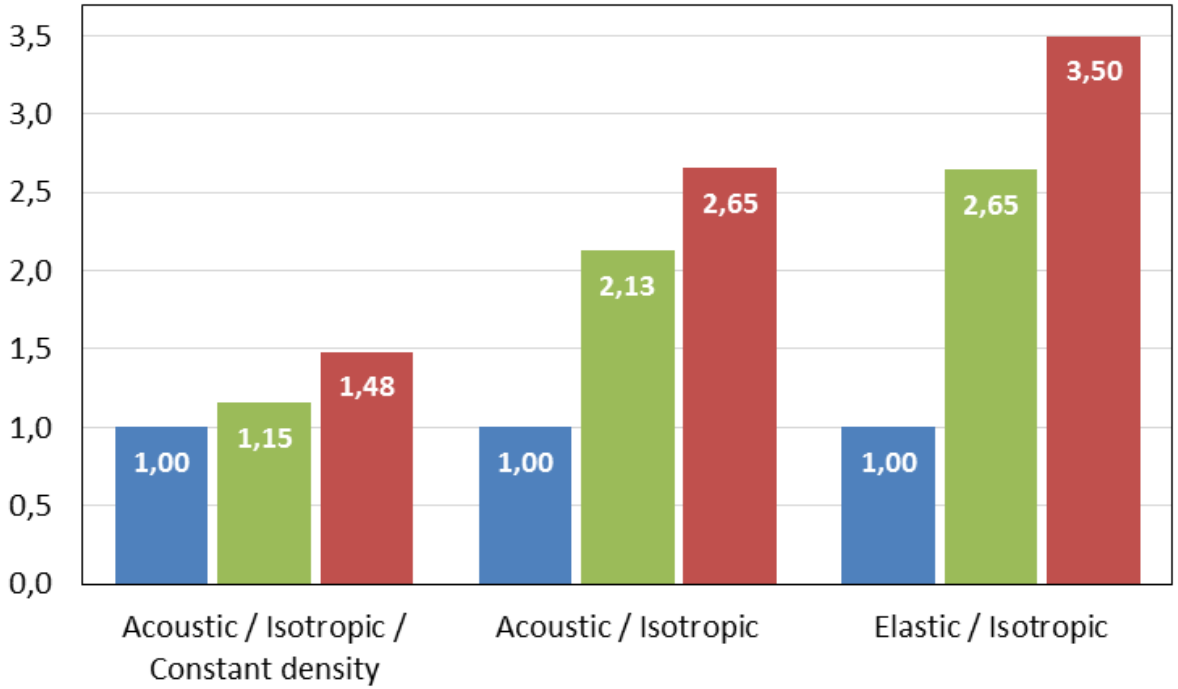
- RTM (Reverse Time Migration) on SX-Aurora TSUBASA shows good performance for memory bandwidth kernel
- Benefit from single precision packed vector and NEC stencil library

Send thousands of shockwave using air guns and analyze reflected waves by running propagation simulation





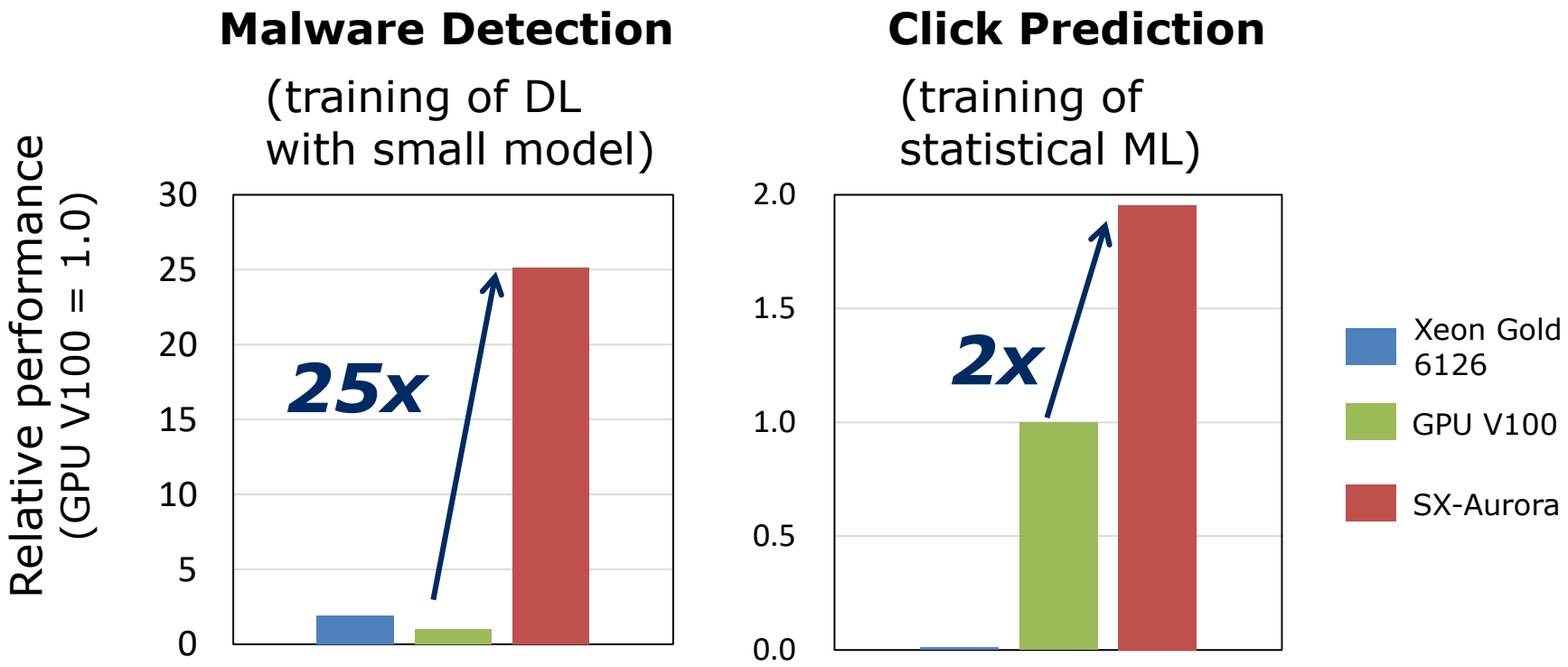
https://www.exxonmobil.eu/en-eu/energy/oil/technology



X86: dual socket Skylake 6148 (2x 20 cores @ 2.4GHz)
NVIDIA V100 PCIe 32GB – OpenACC
SX-Aurora TSUBASA Vector Engine (8 cores @ 1.4GHz)

TensorFlow

- NEC ported TensorFlow to SX-Aurora TSUBASA
- TF on SX-Aurora TSUBASA shows good performance in training of DL/ML

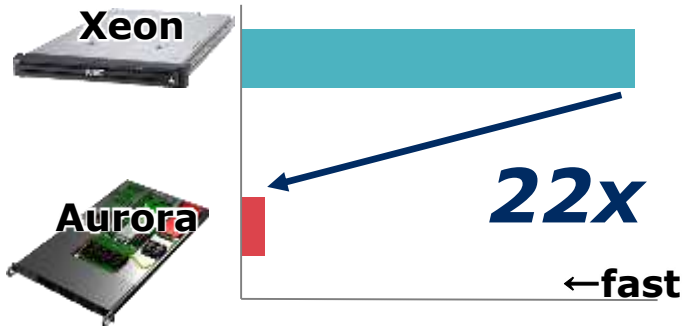


https://github.com/sx-aurora-dev/tensorflow
https://sx-aurora-dev.github.io/blog/post/20190605-tf-release/

Recommendation

Provide recommendation service with Aurora utilizing buying/browsing history at other shop (cross recommendation)
→ Sales increased & Customer satisfaction increased

Learning time of customer's buying/browsing history

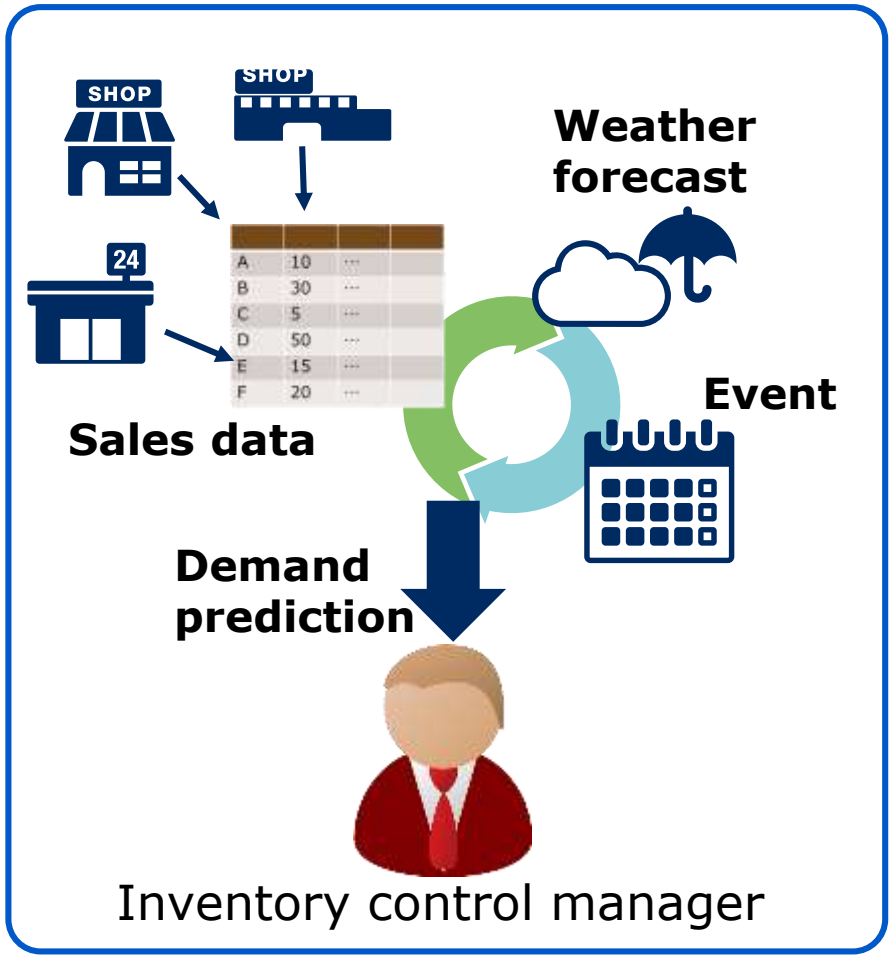


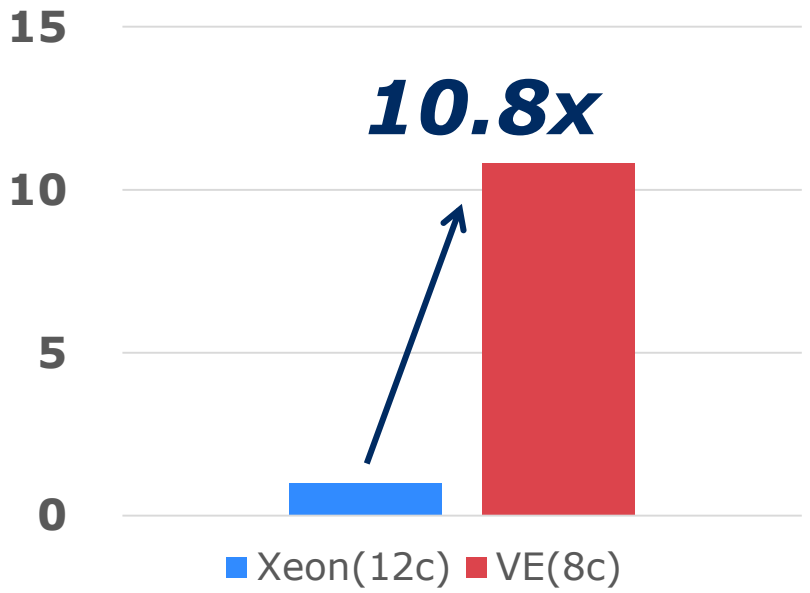
Aurora : 1VE/8core
Xeon : Gold 6126/1socket/12core

More frequent learning
(e.g. daily learning → hourly learning)
and enhance recommendation accuracy

Demand prediction

Over 10x faster demand prediction taking into account weather, event, sales record, etc



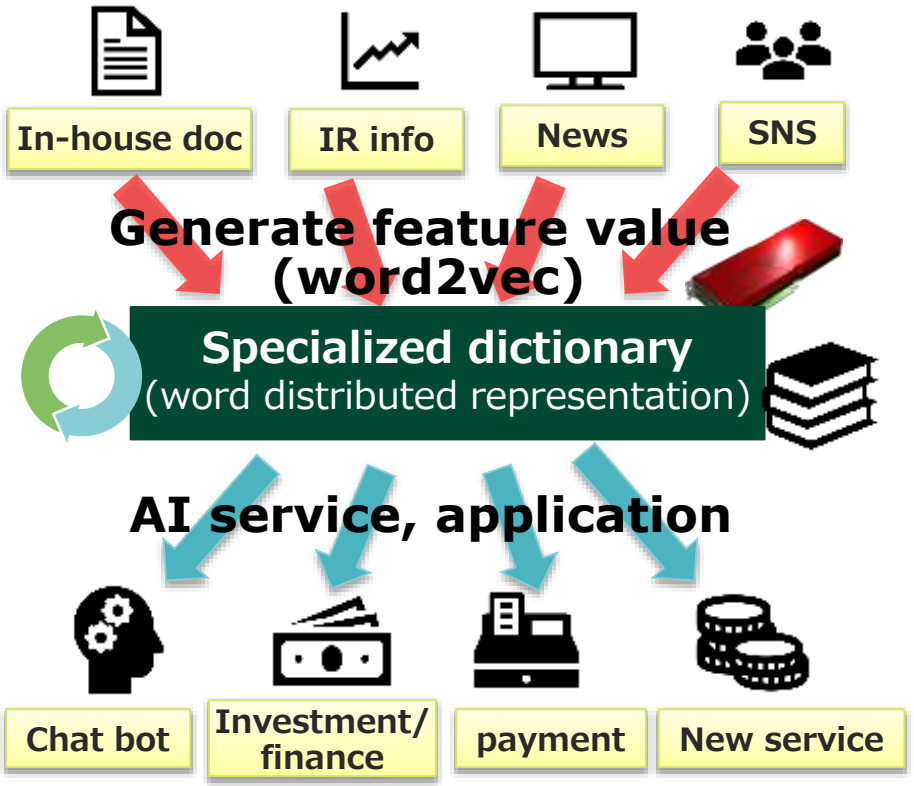


method : regression tree
Aurora : 1VE 8core
x86 : Xeon Gold 6126 1socket 12core

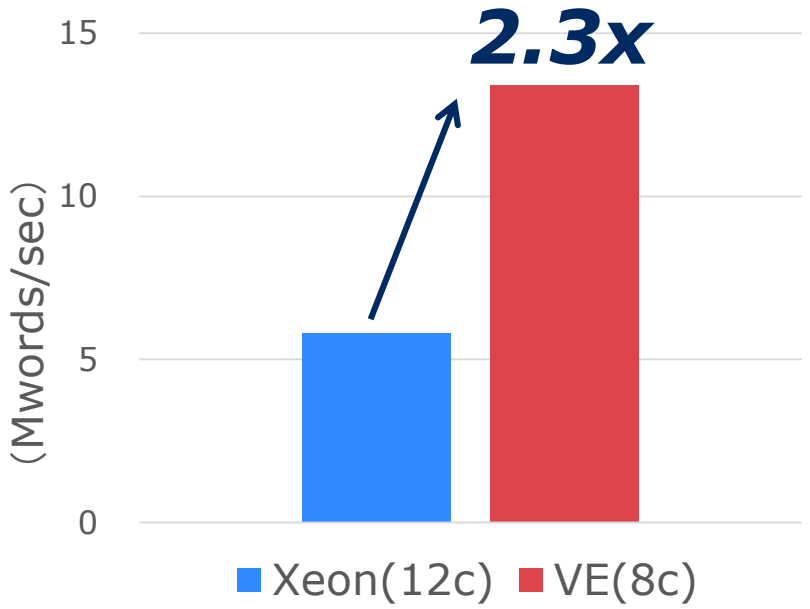
Text analysis : word2vec

Generate feature value for text analysis

Word2vec: word distributed representation in natural language processing task



- Word distributed representation is used as a "dictionary" so that AI can understand human language
- Quality of service can be improved if dictionary can be created/updated over a short time



method : word2vec
Aurora : 1VE 8core
x86 : Xeon Gold 6126 1socket 12core

Conclusion: How we can contribute to your R&D

10 reasons why you want to use SX-Aurora TSUBASA

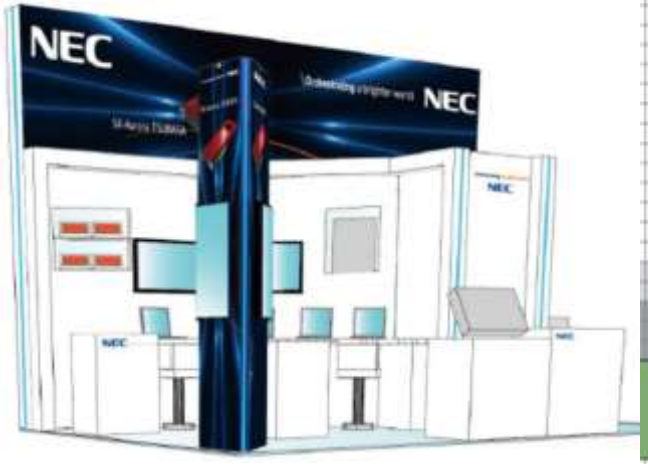
1. SX series is a proven product. It has been in the market for 35 years
2. Because it's vector. Vector technology is back into the spotlight
3. Try your existing codes, especially old vector codes
4. Use your time for your R&D. Do not use your time for programming
5. Unlimited machine time and no queue
6. It is suited for AI / ML
7. It is suited for HPC simulations. HPC and AI is merging
8. Hybrid architecture is a good research subject
9. It's affordable for personal usage
10. It's fun !!

NEC booth

Come to our booth !

NEC

Booth No: E-1020



NEC booth

Come to NEC booth and Touch SX-Aurora TSUBASA.
Experience the performance of vector computing.

Hands-On session (using sample program)

- Learn how to compile and run programs
- Learn how to see performance
- Learn how to vectorize/optimize

Talk to our engineers

- Run sample programs
- Bring your program and run it on SX-Aurora TSUBASA

Join tuning contest

- Our tuning advisor can help you
- Win a prize

Hands-On Schedule

Monday Jun 17
17:00-17:45

Tuesday Jun 18
11:00-11:45
14:00-14:45
16:00-16:45

Wednesday Jun 19
11:00-11:45
14:00-14:45

Bring your Laptop and try Vector !
Bring your code and try it on SX-Aurora TSUBASA !
Or just watching will be fun !!


Orchestrating a brighter world
NEC